THE POWER TO KNOW.

SAS/STAT® 9.2

User's Guide

Second Edition



SAS® Documentation

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2009. *SAS/STAT® 9.2 User's Guide, Second Edition*. Cary, NC: SAS Institute Inc.

SAS/STAT® 9.2 User's Guide, Second Edition

Copyright © 2009, SAS Institute Inc., Cary, NC, USA

All rights reserved. Produced in the United States of America.

For a hard-copy book: No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, or otherwise, without the prior written permission of the publisher, SAS Institute Inc.

For a Web download or e-book: Your use of this publication shall be governed by the terms established by the vendor at the time you acquire this publication.

U.S. Government Restricted Rights Notice: Use, duplication, or disclosure of this software and related documentation by the U.S. government is subject to the Agreement with SAS Institute and the restrictions set forth in FAR 52.227-19, Commercial Computer Software-Restricted Rights (June 1987).

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st electronic book, September 2009

SAS® Publishing provides a complete selection of books and electronic products to help customers use SAS software to its fullest potential. For more information about our e-books, e-learning products, CDs, and hard-copy books, visit the SAS Publishing Web site at **support.sas.com/publishing** or call 1-800-727-3228.

SAS® and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are registered trademarks or trademarks of their respective companies.

Contents

Acknowledgments vii

Chapter 1. What's New in SAS/STAT 1

Chapter 2. Introduction 19

Chapter 3. Introduction to Statistical Modeling with SAS/STAT Software 27

Chapter 4. Introduction to Regression Procedures 75

Chapter 5. Introduction to Analysis of Variance Procedures 115

Chapter 6. Introduction to Mixed Modeling Procedures 127

Chapter 7. Introduction to Bayesian Analysis Procedures 141

Chapter 8. Introduction to Categorical Data Analysis Procedures 181

Chapter 9. Introduction to Multivariate Procedures 195

Chapter 10. Introduction to Discriminant Procedures 201

Chapter 11. Introduction to Clustering Procedures 209

Chapter 12. Introduction to Scoring, Standardization, and Ranking Procedures 251

Chapter 13. Introduction to Survival Analysis Procedures 253

Chapter 14. Introduction to Survey Procedures 259

Chapter 15. The Four Types of Estimable Functions 271

Chapter 16. Introduction to Nonparametric Analysis 291

Chapter 17. Introduction to Structural Equation Modeling with Latent Variables 299

Chapter 18. Shared Concepts and Topics 365

Chapter 19. Introduction to Power and Sample Size Analysis 413

Chapter 20. Using the Output Delivery System 433

Chapter 21. Statistical Graphics Using ODS 497

Chapter 22. The ACECLUS Procedure 659

Chapter 23. The ANOVA Procedure 691

Chapter 24. The BOXPLOT Procedure 749

Chapter 25. The CALIS Procedure 827

Chapter 26. The CANCORR Procedure 1029

Chapter 27. The CANDISC Procedure 1061

Chapter 28. The CATMOD Procedure 1091
Chapter 29. The CLUSTER Procedure 1229
Chapter 30. The CORRESP Procedure 1319
Chapter 31. The DISCRIM Procedure 1383
Chapter 32. The DISTANCE Procedure 1483
Chapter 33. The FACTOR Procedure 1533
Chapter 34. The FASTCLUS Procedure 1621
Chapter 35. The FREQ Procedure 1675
Chapter 36. The GAM Procedure 1833
Chapter 37. The GENMOD Procedure 1893
Chapter 38. The GLIMMIX Procedure 2077
Chapter 39. The GLM Procedure 2431
Chapter 40. The GLMMOD Procedure 2621
Chapter 41. The GLMPOWER Procedure 2641
Chapter 42. The GLMSELECT Procedure 2683
Chapter 43. The HPMIXED Procedure (Experimental) 2803
Chapter 44. The INBREED Procedure 2857
Chapter 45. The KDE Procedure 2885
Chapter 46. The KRIGE2D Procedure 2929
Chapter 47. The LATTICE Procedure 2979
Chapter 48. The LIFEREG Procedure 2991
Chapter 49. The LIFETEST Procedure 3099
Chapter 50. The LOESS Procedure 3187
Chapter 51. The LOGISTIC Procedure 3255
Chapter 52. The MCMC Procedure 3477
Chapter 53. The MDS Procedure 3703
Chapter 54. The MI Procedure 3743
Chapter 55. The MIANALYZE Procedure 3839
Chapter 56. The MIXED Procedure 3891
Chapter 57. The MODECLUS Procedure 4093

Chapter 58. The MULTTEST Procedure 4181
Chapter 59. The NESTED Procedure 4253
Chapter 60. The NLIN Procedure 4267
Chapter 61. The NLMIXED Procedure 4343
Chapter 62. The NPAR1WAY Procedure 4443
Chapter 63. The ORTHOREG Procedure 4505
Chapter 64. The PHREG Procedure 4521
Chapter 65. The PLAN Procedure 4731
Chapter 66. The PLS Procedure 4765
Chapter 67. The POWER Procedure 4815
Chapter 68. The Power and Sample Size Application 5047
Chapter 69. The PRINCOMP Procedure 5143
Chapter 70. The PRINQUAL Procedure 5195
Chapter 71. The PROBIT Procedure 5257
Chapter 72. The QUANTREG Procedure 5357
Chapter 73. The REG Procedure 5433
Chapter 74. The ROBUSTREG Procedure 5647
Chapter 75. The RSREG Procedure 5719
Chapter 76. The SCORE Procedure 5759
Chapter 77. The SEQDESIGN Procedure 5783
Chapter 78. The SEQTEST Procedure 5983
Chapter 79. The SIM2D Procedure 6155
Chapter 80. The SIMNORMAL Procedure 6193
Chapter 81. The STDIZE Procedure 6211
Chapter 82. The STEPDISC Procedure 6267
Chapter 83. The SURVEYFREQ Procedure 6293
Chapter 84. The SURVEYLOGISTIC Procedure 6373
Chapter 85. The SURVEYMEANS Procedure 6465
Chapter 86. The SURVEYREG Procedure 6533
Chapter 87. The SURVEYSELECT Procedure 6613

Chapter 88. The TCALIS Procedure (Experimental) 6679
Chapter 89. The TPSPLINE Procedure 7057
Chapter 90. The TRANSREG Procedure 7105
Chapter 91. The TREE Procedure 7355
Chapter 92. The TTEST Procedure 7391
Chapter 93. The VARCLUS Procedure 7461
Chapter 94. The VARCOMP Procedure 7493
Chapter 95. The VARIOGRAM Procedure 7517
Appendix A. Special SAS Data Sets 7605

Subject Index **7621**

Syntax Index **7745**

Acknowledgments

Credits

Documentation

Editing	Anne Jones, Ed Huddleston
Documentation Support	Tim Arnold

Software

The procedures and applications in SAS/STAT software were implemented by the following members of the development staff. Program development includes design, programming, debugging, support, and documentation. In the following list, the names of the developers who currently provide primary support are listed first; other developers and previous developers are also listed.

ACECLUS	Ann Kuo, Warren S. Sarle, Donna Lucas Watts
ANOVA	Randall D. Tobias, Yang C. Yuan
BOXPLOT	Bucky Ransdell, Robert N. Rodriguez
CALIS	Yiu-Fai Yung
CANCORR	Ann Kuo, Warren S. Sarle, Donna Lucas Watts
CANDISC	Warren F. Kuhfeld, Warren S. Sarle, Yang C. Yuan
CATMOD	Robert E. Derr, John P. Sall, Donna Lucas Watts
CLUSTER	Bart Killam, Warren S. Sarle
CORRESP	Warren F. Kuhfeld
DISCRIM	Warren F. Kuhfeld, Warren S. Sarle, Yang C. Yuan
DISTANCE	Ann Kuo
FACTOR	Yiu-Fai Yung, John P. Sall, Warren S. Sarle
FASTCLUS	Bart Killam, Warren S. Sarle, Donna Lucas Watts
FREQ	Donna Lucas Watts, John P. Sall
GAM	Weijie Cai, Randall D. Tobias, Robert Cohen

GENMOD	Gordon Johnston
GLIMMIX	Oliver Schabenberger
GLM	Randall D. Tobias, James H. Goodnight, John P. Sall, Warren S. Sarle, Yang C. Yuan
GLMMOD	Randall D. Tobias
GLMSELECT	Robert Cohen
HPMIXED	Tianlin Wang
INBREED	Wendy Czika, Anthony Baiching An
KDE	Bucky Ransdell, Russell D. Wolfinger
KRIGE2D	Alexander Kolovos, Bart Killam
LATTICE	Oliver Schabenberger, Russell D. Wolfinger
LIFEREG	Colin Chen, Gordon Johnston
LIFETEST	Ying So
LOESS	Robert Cohen
LOGISTIC	Robert E. Derr, Ying So
MCMC	Fang Chen
MDS	Warren S. Sarle, Warren F. Kuhfeld
MI	Yang C. Yuan
MIANALYZE	Yang C. Yuan
MIXED	Oliver Schabenberger, Russell D. Wolfinger
MODECLUS	Ann Kuo, Warren S. Sarle
MULTTEST	Robert E. Derr, Russell D. Wolfinger
NESTED	Randall D. Tobias, Leigh A. Ihnen
NLIN	Oliver Schabenberger, Don Erdman, James H. Goodnight, Leigh A. Ihnen
NLMIXED	Oliver Schabenberger, Russell D. Wolfinger
NPAR1WAY	Donna Lucas Watts, Jane Evans, John P. Sall
ORTHOREG	Randall D. Tobias, John P. Sall
PHREG	Ying So
PLAN	Pushpal K Mukhopadhyay, Randall D. Tobias, Leigh A. Ihnen
PLS	Randall D. Tobias
POWER	John Castelloe
PRINCOMP	Ann Kuo, Warren S. Sarle
PRINQUAL	Warren F. Kuhfeld
PROBIT	Colin Chen, Gordon Johnston
PSS	Wayne Watson
QUANTREG	Colin Chen
REG	Robert Cohen, Leigh A. Ihnen, John P. Sall
ROBUSTREG	Colin Chen
RSREG	Robert E. Derr, Randall D. Tobias, John P. Sall
SCORE	Ann Kuo, Donna Lucas Watts
SEQDESIGN	Yang C. Yuan
SEQTEST	Yang C. Yuan
SIM2D	Alexander Kolovos, Bart Killam
SIMNORMAL	Bart Killam
STDIZE	Ann Kuo
STEPDISC	Warren F. Kuhfeld, Warren S. Sarle, Yang C. Yuan
SURVEYFREQ	Donna Lucas Watts

SURVEYLOGISTIC	Anthony Baiching An, Pushpal K Mukhopadhyay
SURVEYMEANS	Anthony Baiching An
SURVEYREG	Anthony Baiching An, Pushpal K Mukhopadhyay
SURVEYSELECT	Donna Lucas Watts
TCALIS	Yiu-Fai Yung
TPSPLINE	Weijie Cai, Randall D. Tobias
TRANSREG	Warren F. Kuhfeld
TREE	Bucky Ransdell, Warren S. Sarle
TTEST	John Castelloe, James H. Goodnight, Padraic Neville, Warren S. Sarle,
VARCLUS	Warren S. Sarle
VARCOMP	Tianlin Wang, Oliver Schabenberger, James H. Goodnight, Randall D. Tobias, Russell D. Wolfinger
VARIOGRAM	Alexander Kolovos, Bart Killam
Probability Routines	Georges Guirguis
Numerical and Graphical Routines	Anthony Baiching An, Fang Chen, Robert Cohen, Mike Cybrynski, Robert E. Derr, Jane Evans, Georges Guirguis, Warren F. Kuhfeld, Warren S. Sarle, Oliver Schabenberger, Randall D. Tobias, Donna Lucas Watts, Yang C. Yuan, Leigh A. Ihnen, Richard D. Langston, Katherine Ng, John P. Sall, Tianlin Wang

The following people contribute to SAS/STAT software with their leadership and support: Robert Cohen, Warren F. Kuhfeld, Robert N. Rodriguez, Oliver Schabenberger, Maura E. Stokes, and Randall D. Tobias.

Testing

Shu An, Jack J. Berry, Ming-Chun Chang, Jared L. Dean, Bruce Elsheimer, Kelly M. Graham, Gregory D. Goodwin, Gerardo I. Hurtado, Cheryl LeSaint, Ming Liao, Yu Liang, Mark Traccarella, Greg Weier, Li Yang, Fouad G. Younan

Technical Support

Rob Agnelli, Craig DeVault, Phil Gibbs, Duane Hayes, Elizabeth S. Edwards, Kathleen Kiernan, Eddie Routten, Annette Sanders, Paul T. Savarese, David Schlotzhauer, Mike Stockstill

Acknowledgments

Many people make significant and continuing contributions to the development of SAS software products. The following are some of the people who have contributed significant amounts of their time to help us make improvements to SAS/STAT software. This includes research and consulting, testing, and reviewing documentation. We are grateful for the involvement of these members of the statistical community and the many others who are not mentioned here for their feedback, suggestions, and consulting.

Alan Agresti, University of Florida; Paul Allison, University of Pennsylvania; Douglas Bates, University of Wisconsin; John Barnard Jr., Cleveland Clinic Foundation; David Binder, David Binder Research; Suzette Blanchard, Frontier Science Technology Research Foundation; Mary Butler Moore, formerly of University of Florida at Gainesville; Wilbert P. Byrd, Clemson University; Vincent Carey, Harvard University; Sally Carson, RAND; Love Casanova, CSC-FSG; Helene Cavior, Abacus Concepts; Rao Chaganty, Old Dominion University; George Chao, DuPont Merek Pharmaceutical Company; Daniel M. Chilko, West Virginia University; Marc Cohen, Fair Isaac Corporation; Jan de Leeuw, University of California, Los Angeles; Dave DeLong, Duke University; Alex Dmitrienko, Eli Lilly; Sandra Donaghy, North Carolina State University; David B. Duncan, Johns Hopkins University; Paul Eilers, Leiden University; Scott Emerson, University of Washington; Michael Farrell, Oak Ridge National Laboratory; Stewart Fossceco, SLF Consulting; Michael Friendly, York University; Rudolf J. Freund, Texas A&M University; Wayne Fuller, Iowa State University; Andrzej Galecki, University of Michigan; A. Ronald Gallant, Duke University; Joseph Gardiner, Michigan State University; Charles Gates, Texas A&M University; Thomas M. Gerig, North Carolina State University; Francis Giesbrecht, North Carolina State University; Harvey J. Gold, North Carolina State University; Kenneth Goldberg, Centocor Inc; Donald Guthrie, University of California, Los Angeles; Gerald Hajian, Schering Plough Research Institute; Bob Hamer, University of North Carolina at Chapel Hill; Frank E. Harrell Jr., Vanderbilt University; Wolfgang M. Hartmann; Walter Harvey, Ohio State University; Douglas Hawkins, University of Minnesota; Xuming He, University of Illinois at Urbana-Champaign; Ronald W. Helms, Rho, Inc.; Joseph Hilbe, Arizona State University; Gerry Hobbs, West Virginia University; Ronald R. Hocking, Texas A & M University; Nick Horton, Smith College; Julian Horwich, Camp Conference Company; Jason C. Hsu, Ohio State University; David Hurst, University of Alabama at Birmingham; Joseph G. Ibrahim, University of North Carolina at Chapel Hill; Emilio A. Icaza, Louisiana State University; Joerg Kaufman, Bayer Schering Pharma AG; William Kennedy, Iowa State University; Gary Koch, University of North Carolina at Chapel Hill; Roger Koenker, University of Illinois at Urbana-Champaign; Kenneth L. Koonce, Louisiana State University; Rich La Valley, Strategic Technology Solutions; Russell V. Lenth, University of Iowa; Charles Lin, U.S. Census Bureau; Danyu Lin, University of North Carolina; Ardell C. Linnerud, North Carolina State University; Ramon C. Littel, University of Florida; George MacKenzie, University of Oregon; Brian Marx, Louisiana State University; J. Jack McArdle, University of Southern California; Roderick P. McDonald, Macquarie University; Alfio Marazzi, University of Lausanne; J. Philip Miller, Washington University Medical School; George Milliken, Kansas State University; Robert J. Monroe, North Carolina State University; Robert D. Morrison, Oklahoma State University; Keith Muller, University of Florida; Anupama Narayanan, Procter & Gamble Co; Meltem Narter; Ralph G. O'Brien, Cleveland Clinic Foundation; Kenneth Offord, Mayo Clinic; Christopher R. Olinger, d-Wise Technologies; Christopher J. Paciorek, Harvard University; Robert Parks, Washington University; Richard M. Patterson,

Auburn University; Virginia Patterson, University of Tennessee; Cliff Pereira, Oregon State University; Hans-Peter Piepho, Universität Hohenheim; Edward Pollak, Iowa State University; John Preisser, University of North Carolina at Chapel Hill; C. H. Proctor, North Carolina State University; Bahjat Qaqish, University of North Carolina at Chapel Hill; Dana Quade, University of North Carolina at Chapel Hill; Bill Raynor, Kimberly Clark; Georgia Roberts, Statistics Canada; James Roger, GlaxoSmithKline; Peter Rousseeuw, University of Antwerp; Donald Rubin, Harvard University; Joseph L. Schafer, Pennsylvania State University; Robert Schechter, AstraZeneca; Shayle Searle, Cornell University; Pat Hermes Smith, formerly of Ciba-Geigy; Roger Smith, formerly of USDA; Phil Spector, University of California, Berkeley; Michael Speed, Texas A&M University at College Station; William Stanish, Statistical Insight; Rodney Strand, Orion Enterprises, LLC; Walter Stroup, University of Nebraska; Robert Teichman, ICI Americas Inc.; Terry M. Therneau, Mayo Clinic; Edward Vonesh, Northwestern University; Grace Wahba, University of Wisconsin at Madison; Glenn Ware, University of Georgia; Peter H. Westfall, Texas Tech University; Edward W. Whitehorne, CI Partners, LLC; William Wigton, USDA; William Wilson, University of North Florida; Philip Whittall, Unilever (retired); Dong Xiang; Victor Yohai, University of Buenos Aires; Forrest W. Young (deceased), formerly of University of North Carolina at Chapel Hill; Ruben Zamar, University of British Columbia; Scott Zeger, Johns Hopkins University

The final responsibility for the SAS System lies with SAS alone. We hope that you will always let us know your opinions about the SAS System and its documentation. It is through your participation that SAS software is continuously improved.

Please see Feedback at `www.sas.com/statistics` for your comments.

Chapter 1

What's New in SAS/STAT

Contents

Overview **2**
ODS Statistical Graphics 2
New Related Software 3
New Procedures 3
Highlights of Enhancements 4
CALIS Procedure **5**
CLUSTER Procedure **5**
CORRESP Procedure **5**
FACTOR Procedure **6**
FREQ Procedure **6**
GAM Procedure **6**
GENMOD Procedure **6**
GLIMMIX Procedure **7**
GLM Procedure **7**
GLMPOWER Procedure **7**
GLMSELECT Procedure **8**
HPMIXED Procedure (Experimental) **8**
KRIGE2D Procedure **8**
LIFEREG Procedure **8**
LIFETEST Procedure **9**
LOGISTIC Procedure **9**
LOESS Procedure **9**
Macros **9**
MCMC Procedure **10**
MDS Procedure **10**
MIXED Procedure **10**
MULTTEST Procedure **10**
NLIN Procedure **11**
NLMIXED Procedure **11**
NPAR1WAY Procedure **11**
PHREG Procedure **11**
PLS Procedure **12**
POWER Procedure **12**
PRINCOMP Procedure **12**

PRINQUAL Procedure 12
PROBIT Procedure 12
PSS Application 13
QUANTREG Procedure 13
REG Procedure 13
RSREG Procedure 13
SEQDESIGN Procedure 14
SEQTEST Procedure 14
SIM2D Procedure 14
SIMNORMAL Procedure 14
STDIZE Procedure 15
SURVEYFREQ Procedure 15
SURVEYLOGISTIC Procedure 15
SURVEYMEANS Procedure 15
SURVEYREG Procedure 16
SURVEYSELECT Procedure 16
TCALIS Procedure (Experimental) 16
TRANSREG Procedure 16
TTEST Procedure 17
VARCOMP Procedure 17
VARIOGRAM Procedure 17

Overview

SAS 9.2 brings many new procedures and new enhancements to existing procedures to SAS/STAT software.

ODS Statistical Graphics

Prior to SAS 9.1, creating graphics with statistical procedures generally required additional programming. SAS 9.1 introduced an experimental extension to the Output Delivery System (ODS), which was used by over two dozen SAS/STAT and SAS/ETS procedures to create statistical graphics as automatically as they create tables. This new functionality, referred to as ODS Statistical Graphics (or ODS Graphics for short), requires minimal additional syntax, and it provides displays commonly needed for data analysis and statistical modeling, including scatter plots, histograms, and box-and-whisker plots.

In SAS 9.2, ODS Graphics is production, and over 50 procedures in SAS/STAT, SAS/ETS, SAS/QC, and Base SAS have been modified to use it. Many new plots are now produced by these procedures, either by default or with the specification of procedure options.

The functionality of ODS Graphics has been extended with the addition of new graph types, ODS styles designed for statistical work, and a point-and-click editor for enhancing titles, labels, and other graph features. You can also modify graphs by changing their underlying templates, which are supplied by SAS and are written in the Graph Template Language (GTL). The LISTING destination is now supported by ODS Graphics. A new family of SAS/GRAPH procedures uses ODS Graphics to create standalone plots, such as scatterplots overlaid with smoothers, which are particularly useful for exploratory data analysis. The new SGRENDER procedure provides a way to create customized displays by writing your own templates with the GTL.

See Chapter 21, "Statistical Graphics Using ODS," for an introduction to ODS Graphics and general information about ODS Graphics. The syntax for requesting plots with statistical procedures is described in the procedure chapters.

Note that a SAS/GRAPH license is now required to use ODS Graphics.

New Related Software

SAS/STAT users will be interested in SAS/IML® Studio, formerly known as SAS® Stat Studio, which is new software for data exploration and analysis. SAS/IML Studio provides a highly flexible programming environment in which you can run SAS/STAT or SAS/IML® analyses and display the results with dynamically linked graphics and data tables. SAS/IML Studio is intended for data analysts who write SAS programs to solve statistical problems but need more versatility for data exploration and model building. The programming language in SAS/IML Studio, which is called IMLPlus, is an enhanced version of the SAS/IML programming language. IMLPlus extends SAS/IML to provide new language features, including the ability to create and manipulate statistical graphics, call SAS procedures as functions, and call computational programs written in C, C++, Java, and Fortran. SAS/IML Studio runs on a PC in the Microsoft Windows operating environment.

SAS/IML Studio also includes an experimental interface to the R language. The IMLPlus language includes functions that transfer data between SAS data sets and R data frames, and between SAS/IML matrices and R matrices.

SAS/IML Studio is also the successor to the SAS/INSIGHT® product and provides the same interactive functionality. It is distributed with the SAS/IML product. For more information about SAS/IML Studio, see the *SAS/IML Studio User's Guide* and *SAS/IML Studio for SAS/STAT Users*.

New Procedures

SAS 9.2 brings a number of new procedures to SAS/STAT software. Several of these procedures have been previously available as Web downloads for SAS 9.1.3: GLIMMIX, GLMSELECT, and QUANTREG. The GLMSELECT procedure performs effect selection in the framework of general linear models. The QUANTREG procedure performs quantile regression. The GLIMMIX procedure analyzes generalized linear mixed models. All of these procedures are production with SAS 9.2 and are available on all platforms.

In addition, Bayesian capabilities were introduced to three procedures via Web downloads for SAS 9.1.3. The BGENMOD, BLIFEREG, and BPHREG procedures were experimental versions of the GENMOD, LIFEREG, and PHREG procedures that used the Gibbs sampler to produce posterior distributions while also providing trace plots and convergence diagnostics. These capabilities have been rolled into the GENMOD, LIFEREG, and PHREG procedures for SAS 9.2 and are now production software.

The MCMC, SEQDESIGN, and SEQTEST procedures were introduced as experimental procedures in SAS 9.2, and they became production in SAS 9.2M2.

The MCMC procedure is a general purpose Markov chain Monte Carlo (MCMC) simulation procedure that is designed to fit a variety of Bayesian models. You specify a likelihood function for the data and a prior distribution for the parameters. PROC MCMC obtains samples from the corresponding posterior distributions, produces summary and diagnostic statistics, and saves the posterior samples in an output data set.

The SEQDESIGN and SEQTEST procedures are tools for group sequential analysis. The SEQDESIGN procedure designs interim analyses for clinical trials, and the SEQTEST procedure performs interim analyses.

The experimental HPMIXED procedure uses a number of specialized high-performance techniques to fit linear mixed models with variance component structure. The HPMIXED procedure is specifically designed to cope with estimation problems that involve a large number of fixed effects, a large number of random effects, or a large number of observations. The models supported by the HPMIXED procedure are a subset of the models that you can fit with the MIXED procedure, and PROC HPMIXED can provide substantial performance improvements in terms of memory requirements and computational speed.

The experimental TCALIS procedure updates the CALIS procedure for structural equation modeling. It will become the CALIS procedure in the next release of SAS/STAT software.

Highlights of Enhancements

The Power and Sample Size application (PSS), previously available as a Web application, has been rewritten as a Java client. Its documentation is now included here; see Chapter 68, "The Power and Sample Size Application."

In addition, over two hundred enhancements have been added to existing procedure in SAS/STAT. For example,

- The TTEST procedure provides simple crossover analysis as well as equivalence tests.
- Jackknife and BRR variance estimation and domain analysis are now provided by all of the survey data analysis procedures.
- The POWER procedure now provides power for a number of additional analyses.

- The GENMOD procedure fits zero-inflated Poisson regression models. PROC GENMOD also provides deletion and diagnostics statistics for its GEE models and provides graphics for these statistics.
- The PHREG procedure adds a HAZARDRATIO statement for computing hazard ratios, including hazard ratios in the presence of interactions.
- The GLIMMIX procedure introduces the COVTEST statement for inference about covariance parameters. In addition, PROC GLIMMIX provides new estimation methods: Laplace and adaptive quadrature.
- An experimental EFFECT statement can be found in the GLIMMIX, GLMSELECT, and QUANTREG procedures. It enables you to construct special collections of columns for design matrices (for example, splines and multimember effects).

Finally, note that this documentation contains several new introductory chapters. See Chapter 3, "Introduction to Statistical Modeling with SAS/STAT Software," Chapter 6, "Introduction to Mixed Modeling Procedures," Chapter 19, "Introduction to Power and Sample Size Analysis," and Chapter 18, "Shared Concepts and Topics."

More information about the changes and enhancements follow. The details can be found in the documentation for the individual procedures.

CALIS Procedure

Standardized root mean square residuals (SRMSR) are now listed in the fit summary table, and PROC CALIS now offers residual plots. See below for more information on the TCALIS procedure.

CLUSTER Procedure

The PLOTS option in the PROC CLUSTER statement produces plots of the cubic clustering criterion (CCC), the pseudo F (PSF) statistic, and the pseudo t^2 (PST2) statistic, all plotted against the number of clusters.

CORRESP Procedure

The correspondence analysis plot is produced by default when ODS Graphics is enabled.

FACTOR Procedure

You can produce a number of graphs with the PLOTS= option in the PROC FACTOR statement. These include various factor pattern plots, reference structures, and scree and variance explained plots. You can now use the OUT= option in conjunction with a PARTIAL statement. The PARPREFIX= option in the PROC statement specifies the prefix for the residual variables in the output data sets.

FREQ Procedure

The FREQ procedure can now produce frequency plots, cumulative frequency plots, deviation plots, odds ratio plots, and kappa plots. You can now request equivalence and noninferiority tests for the binomial proportion and proportion difference. New confidence limits for the binomial proportion (such as Agresti-Coull, Jeffreys, and Wilson) are now available, as well as unconditional exact confidence limits for the proportion difference. You can request Zelen's exact test for equal odds ratios by specifying the EQOR option in the EXACT statement.

GAM Procedure

The GAM procedure is production with SAS 9.2. PROC GAM now produces graphs, including smoothing component plots and additive component plots. The target for an additive logistic model no longer has to be numeric; PROC GAM offers the same types of options for response and classification variables that are available in procedures such as PROC LOGISTIC and PROC GENMOD. The ANODEV=NOREFIT option in the MODEL statement enables a fast approximation analysis of deviance.

GENMOD Procedure

The BAYES statement produces Bayesian analysis via Gibbs sampling for most of the statistical analyses provided by the GENMOD procedure. This release also includes deletion diagnostics and plots for GLMs and GEEs, zero-inflated Poisson regression models, and AIC and QIC model fit statistics. Martingale residuals are now production. The LSMEANS statement now produces inverse link estimates.

GLIMMIX Procedure

The GLIMMIX procedure fits statistical models to data with correlations or nonconstant variability and where the response is not necessarily normally distributed. These generalized linear mixed models (GLMM), like linear mixed models, assume normal (Gaussian) random effects. Conditional on these random effects, data can have any distribution in the exponential family. The binary, binomial, Poisson, and negative binomial distributions, for example, are discrete members of this family. The normal, beta, gamma, and chi-square distributions are representatives of the continuous distributions in this family. The GLIMMIX procedure was first made available for SAS 9.1.3 as a Web download.

In SAS 9.2, the GLIMMIX procedure provides Laplace and adaptive quadrature estimation methods, and, with them, a likelihood-based empirical estimator. In addition, a new bias-corrected estimator is available. The experimental EFFECT statement provides for the creation of splines as well as other special effects. The COVTEST statement enables likelihood-based inference about the covariance parameters. A number of additional covariance structures have been added, including heterogeneous AR(1), heterogeneous compound symmetry, linear structures, heterogeneous Toeplitz, penalized B-spline, spatial anisotropic, and the Matérn covariance structure. Step-down multiplicity adjustments are now supported for all ADJUST= methods in the LSMEANS, ESTIMATE, and LSMESTIMATE statements, except for ADJUST=NELSON in the LSMEANS statement.

The DDFM=KR(FIRSTRORDER) option drops the second-derivative term in the KR calculations. The OUTDESIGN= option in the PROC GLIMMIX statement enables you to write the **X** and **Z** matrix to an output data set. New graphics include boxplots of data and/or residuals with respect to classification effects as well as plots of odds ratios and their confidence limits. The diffogram, meanplot, anomplot, and controlplot have been enhanced.

GLM Procedure

New graphics are now produced for means and for LS-means comparisons. The experimental EFFECTSIZE option in the MODEL statement adds measures of effect size to many analysis of variance tables. The PLOTS=DIAGNOSTICS and the PLOTS=RESIDUAL options in the PROC GLM statement produce summary diagnostics and residual plots, respectively.

GLMPOWER Procedure

The new ORDER= option in the PROC GLMPOWER statement specifies the sorting order for the levels of all of the classification variables specified in the CLASS statement. Continuous variables are now supported, and the noncentrality parameter is computed.

GLMSELECT Procedure

The GLMSELECT procedure performs effect selection in the framework of general linear models. A variety of model selection methods are available, including the LASSO method of Tibshirani (1996) and the related LAR method of Efron et al. (2004). The procedure offers extensive capabilities for customizing the selection with a wide variety of selection and stopping criteria, from traditional and computationally efficient significance-level-based criteria to more computationally intensive validation-based criteria. The procedure also provides graphical summaries of the selection search.

Enhancements in SAS 9.2 include an OUTDESIGN= option to obtain the design matrix, a PARMLABELSTYLE= option to control the style of the parameter labels, and an experimental EFFECT statement that you can use to create splines, polynomials, multimember, and collection effects.

HPMIXED Procedure (Experimental)

The experimental HPMIXED procedure uses a number of specialized high-performance techniques to fit linear mixed models with variance component structure. The HPMIXED procedure is specifically designed to cope with estimation problems that involve a large number of fixed effects, a large number of random effects, or a large number of observations. While the HPMIXED procedure fits only a subset of the models fit by the MIXED procedure and it does not provide the breadth of confirmatory inference that is available with the MIXED procedure, it can have considerably better performance in terms of memory requirements and computational speed.

KRIGE2D Procedure

ODS Graphics has been added to the KRIGE2D procedure, which now can produce scatter plots and prediction plots.

LIFEREG Procedure

The BAYES statement provides Bayesian analysis via Gibbs sampling.

LIFETEST Procedure

The LIFETEST procedure now produces the Nelson-Aalen estimates of the cumulative hazard function. The number of subjects at risk can be displayed for the Kaplan-Meier survival curves. Comparison methods are available for the k-sample test, and you can now specify a smoother hazard function using the kernel method.

LOGISTIC Procedure

The LOGISTIC procedure performs Firth's penalized maximum likelihood. The MULTIPASS option forces the procedure to reread the input data set as needed rather than requiring its storage in memory or in a temporary file on disk. Estimated culmulative probabilities have been added to the SCORE statement output. The CONTRAST statement now includes the inverse link. The ROC-CONTRAST statement compares different ROC models. Odds ratios in the presence of interactions are now computed, and odds ratio plots are provided. Note that the GRAPHICS statement has been replaced with the PLOTS= option in the PROC statement. The EFFECT plot can now handle multiple CLASS and continuous variables. Standard errors are now produced for the exact parameter estimates.

LOESS Procedure

The LOESS procedure includes a PRESEARCH option that uses a preliminary grid search to improve the chance of finding a global optimum of the selection criterion when a golden section search is used.

Macros

The %POWTABLE macro renders the output of the POWER and GLMPOWER procedures in rectangular form, and it optionally produces simplified results by using weighted means across chosen variables. The %ModStyle macro modifies the colors, line styles, and marker symbols displayed in ODS Graphics plots.

MCMC Procedure

The MCMC procedure is a flexible simulation-based procedure that is suitable for fitting a wide range of Bayesian models. To use the procedure, you need to specify a likelihood function for the data and a prior distribution for the parameters. You might also need to specify hyperprior distributions if you are fitting hierarchical models. PROC MCMC then obtains samples from the corresponding posterior distributions, produces summary and diagnostic statistics, and saves the posterior samples in an output data set that can be used for further analysis. You can analyze data that have any likelihood, prior, or hyperprior with PROC MCMC, as long as these functions are programmable using the SAS DATA step functions. The parameters can enter the model linearly or in any nonlinear functional form. The default algorithm that PROC MCMC uses is an adaptive blocked random-walk Metropolis algorithm that uses a normal proposal distribution.

MDS Procedure

ODS Graphics has been added to the MDS procedure, which now can produce fit plots, coefficient plots, and configuration plots.

MIXED Procedure

The RESIDUAL and INFLUENCE options in the MODEL statement are now production. The PLOTS= option in the PROC MIXED statement is now available to specify graphics.

MULTTEST Procedure

The MULTTEST procedure now provides the adaptive Holm, adaptive Hochberg, adaptive FDR, bootstrap FDR, pFDR, and permutation FDR p-value adjustments. ODS Graphics has been added to PROC MULTTEST, and plots of adjusted p-value plots, raw p-values by rank and histogram, and p-values by test are now available. Satterthwaite degrees of freedom are now provided for the t test. The EPSILON= option in the PROC MULTTEST statement specifies the comparison value.

NLIN Procedure

The ALPHA= option has been added to the PROC NLIN and OUTPUT statements. The PDATA= option in the PARAMETERS statement enables you to assign starting values for parameters through a SAS data set. The DER option in the OUTPUT statement saves the first derivatives of the model with respect to the parameters to the OUTPUT data set.

NLMIXED Procedure

The EMPIRICAL option in the PROC NLMIXED statement requests that the covariance matrix of the parameter estimates be computed as a likelihood-based empirical ("sandwich") estimator (White 1982). Subject-specific gradients can be added to a SAS data set with the SUBGRADIENT option in the PROC NLMIXED statement.

NPAR1WAY Procedure

ODS Graphics has been added to the NPAR1WAY procedure, and you can request boxplots, a median plot, and an empirical distribution plot with the PLOTS= option in the PROC NPAR1WAY statement. PROC NPAR1WAY now computes the Hodges-Lehmann estimate of location shift for two-sample data with the HL option. Confidence limits are provided, and you can request exact confidence limits by specifying the HL option in the EXACT statement. Tests based on Conover scores are now available, including exact tests.

PHREG Procedure

The CLASS statement, previously available only in the TPHREG procedure, is now included with the PHREG procedure. The BAYES statement provides Bayesian analysis via Gibbs sampling. PROC PHREG now fits the piecewise exponential model, which is specified in the BAYES statement. Bayesian baseline survival prediction becomes available with SAS 9.2 as well. The HAZARDRATIO statement provides a new facility for computing hazard ratios, including hazard ratios in the presence of interactions. The PLOTS option in the PROC PHREG statement produces baseline survival function plots. Profile-likelihood confidence limits are now available for hazard ratios produced in classical analyses. Firth's penalized likelihood method is provided as well.

PLS Procedure

The PLS procedure now produces more graphics, including a correlation loadings plot. The MISSING option for handling missing values with imputation is now production.

POWER Procedure

The new LOGISTIC statement performs power and sample size analyses for the likelihood ratio chi-square test of a single predictor in binary logistic regression, possibly in the presence of one or more covariates (where all predictors are independent of each other). The new TWOSAMPLEWILCOXON statement performs power and sample size analyses for the Wilcoxon-Mann-Whitney test for two independent groups. The ONESAMPLEFREQ statement now covers equivalence, noninferiority, and confidence interval precision for a proportion. The PAIREDFREQ statement offers new input parameterizations, including raw proportions and correlation.

PRINCOMP Procedure

The PRINCOMP procedure now produces more graphics, including an ellipse plot. It includes an ID statement and incorporates ID variables as tips in its scatter plots. The PARPREFIX= option in the PROC PRINCOMP statement specifies a prefix for naming the residual variables in the OUT= data set and the OUTSTAT= data set.

PRINQUAL Procedure

The PRINQUAL procedure now produces graphs. These include a multidimensional preference analysis plot and a variable transformation plot.

PROBIT Procedure

The PROBIT procedure now offers a predicted probability plot.

PSS Application

The PSS application has been converted to a Java client application and no longer requires a Web server. It now offers a relative risk parameterization for the two proportions analysis. New analyses covered include equivalence and noninferiority for proportions, confidence interval for one proportion, Wilcoxon-Mann-Whitney for two distributions, logistic regression, and GLM contrasts for interactions.

QUANTREG Procedure

Quantile regression extends the regression model to conditional quantiles of the response variable, such as the 90th percentile. Quantile regression is particularly useful when the rate of change in the conditional quantile, expressed by the regression coefficients, depends on the quantile. The main advantage of quantile regression over least squares regression is its flexibility for modeling data with heterogeneous conditional distributions. The QUANTREG procedure was first made available as a Web download for SAS 9.1.3.

With SAS 9.2, the QUANTREG procedure becomes production. In addition, it now includes the experimental EFFECT statement for generating splines and the ability to output results for multiple quantiles in the OUTPUT statement.

REG Procedure

The REG procedure now includes a lack-of-fit test. The PARTIAL option in the MODEL statement requests partial regression plots for each regressor; the PARTIALDATA option displays partial regression data. Heteroscedasticity-consistent (White) standard errors are now available, and you can obtain a heteroscedasticity-consistent covariance matrix for use with the ACOV, HCC, or WHITE option in the MODEL statement and for heteroscedasticity-consistent tests with the TEST statement.

RSREG Procedure

ODS Graphics has been added to the RSREG procedure. New graphs include diagnostic plots, ridge plots, and surface plots.

SEQDESIGN Procedure

The SEQDESIGN procedure designs interim analyses for clinical trials. PROC SEQDESIGN computes the boundary values and required sample sizes for the trial. The boundary values are derived in such a way that the overall Type I and Type II error probability levels are maintained at the levels specified in the design. Available methods include fixed boundary shape methods (which include unified family methods such as the O'Brien-Fleming method), Whitehead methods, and error spending methods. In addition to the boundary values, the SEQDESIGN procedure computes a variety of quantities such as average sample sizes and stopping probabilities.

SEQTEST Procedure

The SEQTEST procedure is used in conjunction with the SEQDESIGN procedure to carry out interim analyses for clinical trials. At each stage, you analyze all the data available at that point with a statistical procedure and compute a test statistic and its information level. You then use the SEQTEST procedure to compare the test statistic with the boundary values for that stage. If the information levels of the data do not match the levels specified in the design, the SEQTEST procedure modifies the boundary levels appropriately. In addition, the SEQTEST procedure computes quantities such as average sample sizes, stopping probabilities, and conditional power. At the conclusion of the trial, the SEQTEST procedure computes parameter estimates, *p*-values, and confidence limits.

SIM2D Procedure

ODS Graphics has been added to the SIM2D procedure. Means plots and scatter plots of the observed data are now available.

SIMNORMAL Procedure

The SIMNORMAL procedure becomes production with this release.

STDIZE Procedure

The NOTRUNCATE option in the FREQ statement specifies that frequency values are not truncated to integers. Quantile methods now accept noninteger frequencies and handle weights. In order to improve numerical precision, PROC STDIZE now creates double-precision values for output variables instead of inheriting the length of the variables in the analysis.

SURVEYFREQ Procedure

The SURVEYFREQ procedure now provides variance estimation by balanced repeated replication (BRR) and the jackknife methods, in addition to the Taylor series method. You can provide replicate weights for the new replication methods with a REPWEIGHTS statement, or the procedure can construct the replicate weights. PROC SURVEYFREQ now computes odds ratio and relative risk estimates. The new NOMCAR option in the PROC SURVEYFREQ statement requests a subpopulation analysis of the set of respondents for Taylor series variance estimation.

SURVEYLOGISTIC Procedure

The SURVEYLOGISTIC procedure now provides variance estimation by balanced repeated replication (BRR) and the jackknife methods, in addition to the Taylor series method. You can provide replicate weights for the new replication methods with a REPWEIGHTS statement, or the procedure can construct the replicate weights. The OUTPUT and DOMAIN statements are now available. The new NOMCAR option in the PROC SURVEYLOGISTIC statement requests a subpopulation analysis of the set of respondents for Taylor series variance estimation.

SURVEYMEANS Procedure

The SURVEYMEANS procedure now provides variance estimation by balanced repeated replication (BRR) and the jackknife methods, in addition to the Taylor series method. You can provide replicate weights for the new replication methods with a REPWEIGHTS statement, or the procedure can construct the replicate weights. The new NOMCAR option in the PROC SURVEYMEANS statement requests a subpopulation analysis of the set of respondents for Taylor series variance estimation. PROC SURVEYMEANS now computes percentiles (Woodruff variance estimation only).

SURVEYREG Procedure

The SURVEYREG procedure now provides variance estimation by balanced repeated replication (BRR) and the jackknife methods, in addition to the Taylor series method. You can provide replicate weights for the new replication methods with a REPWEIGHTS statement, or the procedure can construct the replicate weights. In addition, PROC SURVEYREG also includes a DOMAIN statement, for domain analysis. The OUTPUT statement enables you to produce predicted values and residuals and put them into a SAS data set. The ORDER= option has been added to the PROC SURVEYREG statement. The new NOMCAR option in the PROC SURVEYREG statement requests a subpopulation analysis of the set of respondents for Taylor series variance estimation.

SURVEYSELECT Procedure

The SURVEYSELECT procedure now provides methods to allocate the total sample size among the strata. Allocation methods include proportional, Neyman, and optimal allocation.

TCALIS Procedure (Experimental)

The TCALIS procedure is experimental in SAS 9.2. It enables you to perform the same kind of statistical analyses that you can do with PROC CALIS. In addition, PROC TCALIS provides functionality such as multiple-group analysis, enhanced mean structure analysis, path-like model specification, support of LISREL-type models, customizable effect analysis, general parametric function testing, customizable Lagrange multiplier tests, and so on. Currently, you can specify COSAN models only in PROC CALIS, but not in PROC TCALIS.

TRANSREG Procedure

The TRANSREG procedure include new options for existing splines to make exterior knot specification easier and more flexible. PROC TRANSREG now includes the penalized B-spline. A number of plots are now produced, including Box-Cox plots, preference mapping, regression residuals, and scatter plots.

TTEST Procedure

The TTEST procedure now performs TOST equivalence analyses, analyses of treatment and period in AB/BA crossover designs, weighted Satterthwaite tests and confidence intervals, analyses of ratios, and one-sided analyses. It supports both normal and lognormal data. Sasabuchi tests and Fieller confidence intervals are computed for normal ratios. PROC TTEST now provides graphs, including histograms, densities, box plots, profiles, agreement plots, Q-Q plots, and interval plots. The new ORDER= option in the PROC TTEST statement specifies the sorting order for the levels of classification variables (specified in the CLASS statement) and crossover treatment variables (specified in the CROSSOVER option in the VAR statement).

VARCOMP Procedure

The METHOD=GRR option has been added to provide gauge repeatability and reproducibility analysis. The CL option has been added to the MODEL statement to compute confidence limits for all of the parameters of interest. This applies to the balanced one-way or two-way designs for METHOD=TYPE1 or GRR. Autocorrelation statistics and tests are now available.

VARIOGRAM Procedure

Autocorrelation statistics are now available. In addition, PROC VARIOGRAM produces graphics, including a scatter plot of the observed data, histogram of the pairwise distance distribution, plots of the empirical classical and robust semivariograms, and panels of the empirical classical and robust semivariogram plots.

Chapter 2

Introduction

Contents

Overview of SAS/STAT Software **19**
Experimental Software **20**
About This Book **20**
Chapter Organization 20
Typographical Conventions 21
Options Used in Examples 22
Where to Turn for More Information **22**
Accessing the SAS/STAT Sample Library 22
Online Documentation 23
SAS Institute Technical Support Services 23
Related SAS Software **23**
Base SAS Software 23
SAS/ETS Software 24
SAS/GRAPH Software 24
SAS/IML Software 24
SAS/INSIGHT Software 25
SAS/OR Software 25
SAS/QC Software 25
SAS/IML Studio 26

Overview of SAS/STAT Software

SAS/STAT software provides comprehensive statistical tools for a wide range of statistical analyses, including analysis of variance, categorical data analysis, cluster analysis, multiple imputation, multivariate analysis, nonparametric analysis, power and sample size computations, psychometric analysis, regression, survey data analysis, and survival analysis. A few examples include nonlinear mixed models, generalized linear models, correspondence analysis, and robust regression. The software is constantly being updated to reflect new methodology.

In addition to over sixty procedures for statistical analysis, SAS/STAT software also includes the Market Research Application (MRA), a point-and-click interface to commonly used techniques in

market research. The Analyst Application provides convenient access to some of the more commonly used statistical analyses in SAS/STAT software including analysis of variance, regression, logistic regression, mixed models, survival analysis, and some multivariate techniques. Also, the Power and Sample Size Application (PSS) is an interface to power and sample size computations. The Analyst Application and MRA are documented separately.

Experimental Software

Experimental software is sometimes included as part of a production-release product. It is provided to (sometimes targeted) customers in order to obtain feedback. All experimental uses are marked Experimental in this document. Whenever an experimental procedure, statement, or option is used, a message is printed to the SAS log to indicate that it is experimental.

The design and syntax of experimental software might change before any production release. Experimental software has been tested prior to release, but it has not necessarily been tested to production-quality standards, and so should be used with care.

About This Book

Since SAS/STAT software is a part of the SAS System, this book assumes that you are familiar with Base SAS software and with the books *SAS Language Reference: Dictionary*, *SAS Language Reference: Concepts* and the *Base SAS Procedures Guide*. It also assumes that you are familiar with basic SAS System concepts such as creating SAS data sets with the DATA step and manipulating SAS data sets with the procedures in Base SAS software (for example, the PRINT and SORT procedures).

Chapter Organization

This book is organized as follows.

Chapter 1, "What's New in SAS/STAT," provides information about the changes and enhancements to SAS/STAT software in SAS 9.2. It describes several new procedures as well as numerous new features.

Chapter 2, this chapter, provides an overview of SAS/STAT software and summarizes related information, products, and services. The remaining introductory chapters provide some introduction to the broad areas covered by SAS/STAT software.

Chapter 18, "Shared Concepts and Topics," provides information on topics that are common to multiple procedures such as parameterization of model effects, the EFFECT statement, and the

NLOPTIONS statement. Chapter 19, "Introduction to Power and Sample Size Analysis," provides documentation for the Power and Sample Size Application (PSS).

Chapter 20, "Using the Output Delivery System," explains the fundamentals of using the Output Delivery System (ODS) to manage your SAS output. Chapter 21, "Statistical Graphics Using ODS," describes the extension to ODS that enables many statistical procedures to create statistical graphics as easily as tables.

Subsequent chapters describe the SAS procedures that make up SAS/STAT software. These chapters appear in alphabetical order by procedure name and are organized as follows:

- The "Overview" section provides a brief description of the analysis provided by the procedure.
- The "Getting Started" section provides a quick introduction to the procedure through a simple example.
- The "Syntax" section describes the SAS statements and options that control the procedure.
- The "Details" section discusses methodology and miscellaneous details, such as ODS tables and ODS graphics.
- The "Examples" section contains examples using the procedure.
- The "References" section contains references for the methodology and for examples of the procedure.

Following the chapters on the SAS/STAT procedures, Chapter A, "Special SAS Data Sets," documents the special SAS data sets associated with SAS/STAT procedures.

Typographical Conventions

This book uses several type styles for presenting information. The following list explains the meaning of the typographical conventions used in this book:

roman	is the standard type style used for most text.
UPPERCASE ROMAN	is used for SAS statements, options, and other SAS language elements when they appear in the text. However, you can enter these elements in your own SAS programs in lowercase, uppercase, or a mixture of the two.
UPPERCASE BOLD	is used in the "Syntax" sections' initial lists of SAS statements and options.
oblique	is used for user-supplied values for options in the syntax definitions. In the text, these values are written in *italic*.
helvetica	is used for the names of variables and data sets when they appear in the text.

bold	is used to refer to matrices and vectors.
italic	is used for terms that are defined in the text, for emphasis, and for references to publications.
`monospace`	is used for example code. In most cases, this book uses lowercase type for SAS code.

Options Used in Examples

Output of Examples

Most of the output shown in this book is produced with the following SAS System options:

```
options linesize=80 pagesize=500 nonumber nodate;
```

The STATISTICAL style is used to create the HTML output and graphs that appears in the online documentation. A style template controls stylistic elements such as colors, fonts, and presentation attributes. The style template is specified in the ODS HTML statement as follows:

```
ods html style=statistical;
```

See Chapter 21, "Statistical Graphics Using ODS," for more information about styles.

If you run the examples, you may get slightly different output. This is a function of the SAS System options used and the precision used by your computer for floating-point calculations.

Where to Turn for More Information

This section describes other sources of information about SAS/STAT software.

Accessing the SAS/STAT Sample Library

The SAS/STAT sample library includes many examples that illustrate the use of SAS/STAT software, including the examples used in this documentation. To access these sample programs, select the **Help** pull-down menu and then select **Getting Started with SAS Software**. From the **Contents** list, choose **Learning to Use SAS** and then **Sample SAS Programs**. Select the **SAS/STAT** product.

Online Documentation

This documentation is available online with the SAS System. If you are using the SAS windowing environment with pull-down menus, you can select **SAS Help and Documentation** from the **Help** menu. Under the **Contents** tab select **SAS/STAT** and then select *SAS/STAT User's Guide* from the list of available topics.

Alternatively, you can type **help STAT** in the command line. Note that you can use the online facility to search the documentation. You can also access the documentation from the SAS Web site. Go to **support.sas.com** and select **Documentation** for more detail.

SAS Institute Technical Support Services

As with all SAS Institute products, the SAS Institute Technical Support staff is available to respond to problems and answer technical questions regarding the use of SAS/STAT software.

Related SAS Software

Many features not found in SAS/STAT software are available in other parts of the SAS System. If you don't find something you need in SAS/STAT software, try looking for the feature in the following SAS software products.

Base SAS Software

The features provided by SAS/STAT software are in addition to the features provided by Base SAS software. Many data management and reporting capabilities you will need are part of Base SAS software. Refer to *SAS Language Reference: Concepts*, *SAS Language Reference: Dictionary*, and the *Base SAS Procedures Guide* for documentation of Base SAS software.

SAS DATA Step

The DATA step is your primary tool for reading and processing data in the SAS System. The DATA step provides a powerful general purpose programming language that enables you to perform all kinds of data processing tasks. The DATA step is documented in *SAS Language Reference: Concepts*.

Base SAS Procedures

Base SAS software includes many useful SAS procedures. Base SAS procedures are documented in the *Base SAS Procedures Guide*. The following is a list of Base SAS procedures you may find useful:

CORR	compute correlations
RANK	compute rankings or order statistics
STANDARD	standardize variables to a fixed mean and variance
MEANS	compute descriptive statistics and summarizing or collapsing data over cross sections
TABULATE	print descriptive statistics in tabular format
UNIVARIATE	compute descriptive statistics

SAS/ETS Software

SAS/ETS software provides SAS procedures for econometrics and time series analysis. It includes capabilities for forecasting, systems modeling and simulation, seasonal adjustment, and financial analysis and reporting. In addition, SAS/ETS software includes an interactive time series forecasting system.

SAS/GRAPH Software

SAS/GRAPH software includes procedures that create two- and three-dimensional high-resolution color graphics plots and charts. You can generate output that graphs the relationship of data values to one another, enhance existing graphs, or simply create graphics output that is not tied to data.

SAS/IML Software

SAS/IML software gives you access to a powerful and flexible programming language (Interactive Matrix Language) in a dynamic, interactive environment. The fundamental object of the language is a data matrix. You can use SAS/IML software interactively (at the statement level) to see results immediately, or you can store statements in a module and execute them later. The programming is dynamic because necessary activities such as memory allocation and dimensioning of matrices are done automatically. SAS/IML software is of interest to users of SAS/STAT software because it enables you to program your methods in the SAS System.

SAS/INSIGHT Software

SAS/INSIGHT software is a highly interactive tool for data analysis. You can explore data through a variety of interactive graphs including bar charts, scatter plots, box plots, and three-dimensional rotating plots. You can examine distributions and perform parametric and nonparametric regression, analyze general linear models and generalized linear models, examine correlation matrixes, and perform principal component analyses. Any changes you make to your data show immediately in all graphs and analyses. You can also configure SAS/INSIGHT software to produce graphs and analyses tailored to the way you work.

SAS/INSIGHT software may be of interest to users of SAS/STAT software for interactive graphical viewing of data, editing data, exploratory data analysis, and checking distributional assumptions.

SAS/OR Software

SAS/OR software provides SAS procedures for operations research and project planning and includes a point-and-click interface to project management. Its capabilities include the following:

- solving transportation problems
- linear, integer, and mixed-integer programming
- nonlinear programming
- scheduling projects
- plotting Gantt charts
- drawing network diagrams
- solving optimal assignment problems
- network flow programming

SAS/OR software may be of interest to users of SAS/STAT software for its mathematical programming features. In particular, the NLP procedure in SAS/OR software solves nonlinear programming problems, and it can be used for constrained and unconstrained maximization of user-defined likelihood functions.

SAS/QC Software

SAS/QC software provides a variety of procedures for statistical quality control and quality improvement. SAS/QC software includes procedures for

- Shewhart control charts
- cumulative sum control charts
- moving average control charts
- process capability analysis
- Ishikawa diagrams
- Pareto charts
- experimental design

SAS/QC software also includes the ADX interface for experimental design.

SAS/IML Studio

SAS/STAT users may be interested in the new SAS/IML Studio (formerly known as SAS Stat Studio). SAS/IML Studio is a tool for data exploration and analysis; it provides a highly flexible programming environment in which you can run SAS/STAT or SAS/IML analyses and display the results with dynamically linked graphics and data tables. You can also call SAS procedures from a SAS/IML program. SAS/IML Studio is intended for data analysts who write SAS programs to solve statistical problems but need more versatility for data exploration and model building.

The programming language in SAS/IML Studio, which is called *IMLPlus*, is an enhanced version of the SAS/IML programming language. IMLPlus extends SAS/IML to provide new language features such as the ability to create and manipulate statistical graphics and to call SAS procedures. SAS/IML Studio runs on a PC in the Microsoft Windows operating environment.

SAS/IML Studio also includes an experimental interface to the R language. The IMLPlus language includes functions that transfer data between SAS data sets and R data frames, and between SAS/IML matrices and R matrices.

SAS/IML Studio is distributed with the SAS/IML product and is also intended to be the successor to the SAS/INSIGHT product. For more information about SAS/IML Studio, see the *SAS/IML Studio User's Guide* and *SAS/IML Studio for SAS/STAT Users*.

Chapter 3

Introduction to Statistical Modeling with SAS/STAT Software

Contents

Overview: Statistical Modeling **28**
Statistical Models 28
Classes of Statistical Models 31
Linear and Nonlinear Models 31
Regression Models and Models with Classification Effects 33
Univariate and Multivariate Models 34
Fixed, Random, and Mixed Models 35
Generalized Linear Models 37
Latent Variable Models 38
Bayesian Models 40
Classical Estimation Principles 42
Least Squares 43
Likelihood 45
Inference Principles for Survey Data 47
Statistical Background **49**
Hypothesis Testing and Power 49
Important Linear Algebra Concepts 49
Expectations of Random Variables and Vectors 57
Mean Squared Error 60
Linear Model Theory 61
Finding the Least Squares Estimators 62
Analysis of Variance 63
Estimating the Error Variance 65
Maximum Likelihood Estimation 65
Estimable Functions 66
Test of Hypotheses 66
Residual Analysis 69
Sweep Operator 71
References **73**

Overview: Statistical Modeling

There are more than 70 procedures in SAS/STAT software, and the majority of them are dedicated to solving problems in statistical modeling. The goal of this chapter is to provide a roadmap to statistical models and to modeling tasks, enabling you to make informed choices about the appropriate modeling context and tool. This chapter also introduces important terminology, notation, and concepts used throughout this documentation. Subsequent introductory chapters discuss model families and related procedures.

It is difficult to capture the complexity of statistical models in a simple scheme, so the classification used here is necessarily incomplete. It is most practical to classify models in terms of simple criteria, such as the presence of random effects, the presence of nonlinearity, characteristics of the data, and so on. That is the approach used here. After a brief introduction to statistical modeling in general terms, the chapter describes a number of model classifications and relates them to modeling tools in SAS/STAT software.

Statistical Models

Deterministic and Stochastic Models

Purely mathematical models, in which the relationships between inputs and outputs are captured entirely in deterministic fashion, can be important theoretical tools but are impractical for describing observational, experimental, or survey data. For such phenomena, researchers usually allow the model to draw on stochastic as well as deterministic elements. When the uncertainty of realizations leads to the inclusion of random components, the resulting models are called *stochastic* models. A *statistical* model, finally, is a stochastic model that contains *parameters*, which are unknown constants that need to be estimated based on assumptions about the model and the observed data.

There are many reasons why statistical models are preferred over deterministic models. For example:

- Randomness is often introduced into a system in order to achieve a certain balance or representativeness. For example, random assignment of treatments to experimental units allows unbiased inferences about treatment effects. As another example, selecting individuals for a survey sample by random mechanisms ensures a representative sample.
- Even if a deterministic model can be formulated for the phenomenon under study, a stochastic model can provide a more parsimonious and more easily comprehended description. For example, it is possible in principle to capture the result of a coin toss with a deterministic model, taking into account the properties of the coin, the method of tossing, conditions of the medium through which the coin travels and of the surface on which it lands, and so on. A very complex model is required to describe the simple outcome—heads or tails. Alternatively, you can describe the outcome quite simply as the result of a stochastic process, a Bernoulli variable that results in heads with a certain probability.

- It is often sufficient to describe the average behavior of a process, rather than each particular realization. For example, a regression model might be developed to relate plant growth to nutrient availability. The explicit aim of the model might be to describe how the average growth changes with nutrient availability, not to predict the growth of an individual plant. The support for the notion of averaging in a model lies in the nature of expected values, describing typical behavior in the presence of randomness. This, in turn, requires that the model contain stochastic components.

The defining characteristic of statistical models is their dependence on parameters and the incorporation of stochastic terms. The properties of the model and the properties of quantities derived from it must be studied in a long-run, average sense through expectations, variances, and covariances. The fact that the parameters of the model must be estimated from the data introduces a stochastic element in applying a statistical model: because the model is not deterministic but includes randomness, parameters and related quantities derived from the model are likewise random. The properties of parameter estimators can often be described only in an asymptotic sense, imagining that some aspect of the data increases without bound (for example, the number of observations or the number of groups).

The process of estimating the parameters in a statistical model based on your data is called *fitting* the model. For many classes of statistical models there are a number of procedures in SAS/STAT software that can perform the fitting. In many cases, different procedures solve identical estimation problems—that is, their parameter estimates are identical. In some cases, the same model parameters are estimated by different statistical principles, such as least squares versus maximum likelihood estimation. Parameter estimates obtained by different methods typically have different statistical properties—distribution, variance, bias, and so on. The choice between competing estimation principles is often made on the basis of properties of the estimators. Distinguishing properties might include (but are not necessarily limited to) computational ease, interprative ease, bias, variance, mean squared error, and consistency.

Model-Based and Design-Based Randomness

A statistical model is a description of the data-generating mechanism, not a description of the specific data to which it is applied. The aim of a model is to capture those aspects of a phenomenon that are relevant to inquiry and to explain how the data could have come about as a realization of a random experiment. These relevant aspects might include the genesis of the randomness and the stochastic effects in the phenomenon under study. Different schools of thought can lead to different model formulations, different analytic strategies, and different results. Coarsely, you can distinguish between a viewpoint of *innate* randomness and one of *induced* randomness. This distinction leads to model-based and design-based inference approaches.

In a design-based inference framework, the random variation in the observed data is induced by random *selection* or random *assignment*. Consider the case of a survey sample from a finite population of size N; suppose that $\mathcal{F}_N = \{y_i : i \in U_N\}$ denotes the finite set of possible values and U_N is the index set $U_N = \{1, 2, \ldots, N\}$. Then a sample S, a subset of U_N, is selected by probability rules. The realization of the random experiment is the selection of a particular set S; the associated values selected from $\mathcal{F}_N$ are considered fixed. If properties of a design-based sampling estimator

are evaluated, such as bias, variance, and mean squared error, they are evaluated with respect to the distribution induced by the sampling mechanism.

Design-based approaches also play an important role in the analysis of data from controlled experiments by randomization tests. Suppose that k treatments are to be assigned to kr homogeneous experimental units. If you form k sets of r units with equal probability, and you assign the jth treatment to the tth set, a completely randomized experimental design (CRD) results. A design-based view treats the potential response of a particular treatment for a particular experimental unit as a constant. The stochastic nature of the error-control design is induced by randomly selecting one of the potential responses.

Statistical models are often used in the design-based framework. In a survey sample the model is used to motivate the choice of the finite population parameters and their sample-based estimators. In an experimental design, an assumption of additivity of the contributions from treatments, experimental units, observational errors, and experimental errors leads to a linear statistical model. The approach to statistical inference where statistical models are used to construct estimators and their properties are evaluated with respect to the distribution induced by the sample selection mechanism is known as *model-assisted inference* (Särndal, Swensson, and Wretman 1992).

In a purely model-based framework, the only source of random variation for inference comes from the unknown variation in the responses. Finite population values are thought of as a realization of a superpopulation model that describes random variables $Y_1, Y_2, \cdots$. The observed values $y_1, y_2, \cdots$ are realizations of these random variables. A model-based framework does not imply that there is only one source of random variation in the data. For example, mixed models might contain random terms that represent selection of effects from hierarchical (super-) populations at different granularity. The analysis takes into account the hierarchical structure of the random variation, but it continues to be model based.

A design-based approach is implicit in SAS/STAT procedures whose name commences with SURVEY, such as the SURVEYFREQ, SURVEYMEANS, SURVEYREG, and SURVEYLOGISTIC procedures. Inferential approaches are model based in other SAS/STAT procedures. For more information about analyzing survey data with SAS/STAT software, see Chapter 14, "Introduction to Survey Procedures."

Model Specification

If the model is accepted as a description of the data-generating mechanism, then its parameters are estimated using the data at hand. Once the parameter estimates are available, you can apply the model to answer questions of interest about the study population. In other words, the model becomes the lens through which you view the problem itself, in order to ask and answer questions of interest. For example, you might use the estimated model to derive new predictions or forecasts, to test hypotheses, to derive confidence intervals, and so on.

Obviously, the model must be "correct" to the extent that it sufficiently describes the data-generating mechanism. Model selection, diagnosis, and discrimination are important steps in the model-building process. This is typically an iterative process, starting with an initial model and refining it. The first important step is thus to formulate your knowledge about the data-generating process and to express the real observed phenomenon in terms of a statistical model. A statistical model describes the distributional properties of one or more variables, the *response* variables. The extent of

the required distributional specification depends on the model, estimation technique, and inferential goals. This description often takes the simple form of a model with additive error structure:

$$\text{response} = \text{mean} + \text{error}$$

In mathematical notation this simple model equation becomes

$$Y = f(x_1, \cdots, x_k; \beta_1, \cdots, \beta_p) + \epsilon$$

In this equation Y is the *response* variable, often also called the *dependent* variable or the *outcome* variable. The terms $x_1, \cdots, x_k$ denote the values of k regressor variables, often termed the *covariates* or the "independent" variables. The terms $\beta_1, \cdots, \beta_p$ denote parameters of the model, unknown constants that are to be estimated. The term ϵ denotes the random disturbance of the model; it is also called the residual term or the error term of the model.

In this simple model formulation, stochastic properties are usually associated only with the ϵ term. The covariates $x_1, \cdots, x_k$ are usually known values, not subject to random variation. Even if the covariates are measured with error, so that their values are in principle random, they are considered fixed in most models fit by SAS/STAT software. In other words, stochastic properties under the model are derived conditional on the xs. If ϵ is the only stochastic term in the model, and if the errors have a mean of zero, then the function $f(\cdot)$ is the *mean function* of the statistical model. More formally,

$$\mathrm{E}[Y] = f(x_1, \cdots, x_k; \beta_1, \cdots, \beta_p)$$

where $\mathrm{E}[\cdot]$ denotes the expectation operator.

In many applications, a simple model formulation is inadequate. It might be necessary to specify not only the stochastic properties of a single error term, but also how model errors associated with different observations relate to each other. A simple additive error model is typically inappropriate to describe the data-generating mechanism if the errors do not have zero mean or if the variance of observations depends on their means. For example, if Y is a Bernoulli random variable that takes on the values 0 and 1 only, a regression model with additive error is not meaningful. Models for such data require more elaborate formulations involving probability distributions.

Classes of Statistical Models

Linear and Nonlinear Models

A statistical estimation problem is nonlinear if the estimating equations—the equations whose solution yields the parameter estimates—depend on the parameters in a nonlinear fashion. Such estimation problems typically have no closed-form solution and must be solved by iterative, numerical techniques.

Nonlinearity in the mean function is often used to distinguish between linear and nonlinear models. A model has a nonlinear mean function if the derivative of the mean function with respect to the

parameters depends on at least one other parameter. Consider, for example, the following models that relate a response variable Y to a single regressor variable x:

$$\mathrm{E}[Y|x] = \beta_0 + \beta_1 x$$
$$\mathrm{E}[Y|x] = \beta_0 + \beta_1 x + \beta_2 x^2$$
$$\mathrm{E}[Y|x] = \beta + x/\alpha$$

In these expressions, $\mathrm{E}[Y|x]$ denotes the expected value of the response variable Y at the fixed value of x. (The conditioning on x simply indicates that the predictor variables are assumed to be non-random. Conditioning is often omitted for brevity in this and subsequent chapters.)

The first model in the previous list is a simple linear regression (SLR) model. It is linear in the parameters β_0 and β_1 since the model derivatives do not depend on unknowns:

$$\frac{\partial}{\beta_0}(\beta_0 + \beta_1 x) = 1$$
$$\frac{\partial}{\beta_1}(\beta_0 + \beta_1 x) = x$$

The model is also linear in its relationship with x (a straight line). The second model is also linear in the parameters, since

$$\frac{\partial}{\beta_0}\left(\beta_0 + \beta_1 x + \beta_2 x^2\right) = 1$$
$$\frac{\partial}{\beta_1}\left(\beta_0 + \beta_1 x + \beta_2 x^2\right) = x$$
$$\frac{\partial}{\beta_2}\left(\beta_0 + \beta_1 x + \beta_2 x^2\right) = x^2$$

However, this second model is *curvilinear*, since it exhibits a curved relationship when plotted against x. The third model, finally, is a nonlinear model since

$$\frac{\partial}{\beta}(\beta + x/\alpha) = 1$$
$$\frac{\partial}{\alpha}(\beta + x/\alpha) = -\frac{x}{\alpha^2}$$

The second of these derivatives depends on a parameter α. A model is nonlinear if it is not linear in at least one parameter. Only the third model is a nonlinear model. A graph of $\mathrm{E}[Y]$ versus the regressor variable thus does not indicate whether a model is nonlinear. A curvilinear relationship in this graph can be achieved by a model that is linear in the parameters.

Nonlinear mean functions lead to nonlinear estimation. It is important to note, however, that nonlinear estimation arises also because of the estimation principle or because the model structure contains nonlinearity in other parts, such as the covariance structure. For example, fitting a simple linear regression model by minimizing the sum of the absolute residuals leads to a nonlinear estimation problem despite the fact that the mean function is linear.

Regression Models and Models with Classification Effects

A linear regression model in the broad sense has the form

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\mathbf{Y}$ is the vector of response values, $\mathbf{X}$ is the matrix of regressor effects, $\boldsymbol{\beta}$ is the vector of regression parameters, and $\boldsymbol{\epsilon}$ is the vector of errors or residuals. A regression model in the narrow sense—as compared to a classification model—is a linear model in which all regressor effects are continuous variables. In other words, each effect in the model contributes a single column to the $\mathbf{X}$ matrix and a single parameter to the overall model. For example, a regression of subjects' weight (Y) on the regressors age (x_1) and body mass index (bmi, x_2) is a regression model in this narrow sense. In symbolic notation you can write this regression model as

$$\text{weight} = \text{age} + \text{bmi} + \text{error}$$

This symbolic notation expands into the statistical model

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \epsilon_i$$

Single parameters are used to model the effects of age (β_1) and bmi (β_2), respectively.

A classification effect, on the other hand, is associated with possibly more than one column of the $\mathbf{X}$ matrix. Classification with respect to a variable is the process by which each observation is associated with one of k levels; the process of determining these k levels is referred to as *levelization* of the variable. Classification variables are used in models to identify experimental conditions, group membership, treatments, and so on. The actual values of the classification variable are not important, and the variable can be a numeric or a character variable. What is important is the association of discrete values or levels of the classification variable with groups of observations. For example, in the previous illustration, if the regression also takes into account the subjects' gender, this can be incorporated in the model with a two-level classification variable. Suppose that the values of the gender variable are coded as "F" and "M," respectively. In symbolic notation the model

$$\text{weight} = \text{age} + \text{bmi} + \text{gender} + \text{error}$$

expands into the statistical model

$$Y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \tau_1 I(\text{gender} = \text{"F"}) + \tau_2 I(\text{gender} = \text{"M"}) + \epsilon_i$$

where $I(\text{gender} = \text{"F"})$ is the indicator function that returns 1 if the value of the gender variable is "F" and 0 otherwise. Parameters τ_1 and τ_2 are associated with the gender classification effect. This form of parameterizing the gender effect in the model is only one of several different methods of incorporating the levels of a classification variable in the model. This form, the so-called singular parameterization, is the most general approach, and it is used in the GLM, MIXED, and GLIMMIX procedures. Alternatively, classification effects with various forms of nonsingular parameterizations are available in such procedures as GENMOD and LOGISTIC. See the documentation for the individual SAS/STAT procedures on their respective facilities for parameterizing classification

variables and the section "Parameterization of Model Effects" on page 368 in Chapter 18, "Shared Concepts and Topics," for general details.

Models that contain only classification effects are often identified with *analysis of variance* (ANOVA) models, because ANOVA methods are frequently used in their analysis. This is particularly true for experimental data where the model effects comprise effects of the treatment and error-control design. However, classification effects appear more widely than in models to which analysis of variance methods are applied. For example, many mixed models, where parameters are estimated by restricted maximum likelihood, consist entirely of classification effects but do not permit the sum of squares decomposition typical for ANOVA techniques.

Many models contain both continuous and classification effects. For example, a continuous-by-class effect consists of at least one continuous variable and at least one classification variable. Such effects are convenient, for example, to vary slopes in a regression model by the levels of a classification variable. Also, recent enhancements to linear modeling syntax in some SAS/STAT procedures (including GLIMMIX and GLMSELECT) enable you to construct sets of columns in $\mathbf{X}$ matrices from a single continuous variable. An example is modeling with splines where the values of a continuous variable x are expanded into a spline basis that occupies multiple columns in the $\mathbf{X}$ matrix. For purposes of the analysis you can treat these columns as a single unit or as individual, unrelated columns. For more details, see the section "Constructed Effects and the EFFECT Statement (Experimental)" on page 377 in Chapter 18, "Shared Concepts and Topics."

Univariate and Multivariate Models

A multivariate statistical model is a model in which multiple response variables are modeled jointly. Suppose, for example, that your data consist of heights (h_i) and weights (w_i) of children, collected over several years (t_i). The following separate regressions represent two univariate models:

$$w_i = \beta_{w0} + \beta_{w1} t_i + \epsilon_{wi}$$
$$h_i = \beta_{h0} + \beta_{h1} t_i + \epsilon_{hi}$$

In the univariate setting, no information about the children's heights "flows" to the model about their weights and vice versa. In a multivariate setting, the heights and weights would be modeled jointly. For example:

$$\begin{aligned} \mathbf{Y}_i = \begin{bmatrix} w_i \\ h_i \end{bmatrix} &= \mathbf{X}\boldsymbol{\beta} + \begin{bmatrix} \epsilon_{wi} \\ \epsilon_{hi} \end{bmatrix} \\ &= \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}_i \\ \boldsymbol{\epsilon}_i &\sim \left(\mathbf{0}, \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{12} & \sigma_2^2 \end{bmatrix} \right) \end{aligned}$$

The vectors $\mathbf{Y}_i$ and $\boldsymbol{\epsilon}_i$ collect the responses and errors for the two observation that belong to the same subject. The errors from the same child now have the correlation

$$\text{Corr}[\epsilon_{wi}, \epsilon_{hi}] = \frac{\sigma_{12}}{\sqrt{\sigma_1^2 \sigma_2^2}}$$

and it is through this correlation that information about heights "flows" to the weights and vice versa. This simple example shows only one approach to modeling multivariate data, through the

use of covariance structures. Other techniques involve seemingly unrelated regressions, systems of linear equations, and so on.

Multivariate data can be coarsely classified into three types. The response vectors of *homogeneous multivariate data* consist of observations of the same attribute. Such data are common in repeated measures experiments and longitudinal studies, where the same attribute is measured repeatedly over time. Homogeneous multivariate data also arise in spatial statistics where a set of geostatistical data is the incomplete observation of a single realization of a random experiment that generates a two-dimensional surface. One hundred measurements of soil electrical conductivity collected in a forest stand compose a single observation of a 100-dimensional homogeneous multivariate vector. *Heterogeneous multivariate* observations arise when the responses that are modeled jointly refer to different attributes, such as in the previous example of children's weights and heights. There are two important subtypes of heterogeneous multivariate data. In *homocatanomic multivariate data* the observations come from the same distributional family. For example, the weights and heights might both be assumed to be normally distributed. With *heterocatanomic multivariate data* the observations can come from different distributional families. The following are examples of heterocatanomic multivariate data:

- For each patient you observe blood pressure (a continuous outcome), the number of prior episodes of an illness (a count variable), and whether the patient has a history of diabetes in the family (a binary outcome). A multivariate model that models the three attributes jointly might assume a lognormal distribution for the blood pressure measurements, a Poisson distribution for the count variable and a Bernoulli distribution for the family history.
- In a study of HIV/AIDS survival, you model jointly a patients CD4 cell count over time—itself a homogeneous multivariate outcome—and the survival of the patient (event-time data).

Fixed, Random, and Mixed Models

Each term in a statistical model represents either a *fixed effect* or a *random effect*. Models in which all effects are fixed are called fixed-effects models. Similarly, models in which all effects are random—apart from possibly an overall intercept term—are called random-effects models. Mixed models, then, are those models that have fixed-effects and random-effects terms. In matrix notation, the linear fixed, linear random, and linear mixed model are represented by the following model equations, respectively:

$$\begin{aligned}\mathbf{Y} &= \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \\ \mathbf{Y} &= \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon} \\ \mathbf{Y} &= \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}\end{aligned}$$

In these expressions, $\mathbf{X}$ and $\mathbf{Z}$ are design or regressor matrices associated with the fixed and random effects, respectively. The vector $\boldsymbol{\beta}$ is a vector of fixed-effects parameters, and the vector $\boldsymbol{\gamma}$ represents the random effects. The mixed modeling procedures in SAS/STAT software assume that the random effects $\boldsymbol{\gamma}$ follow a normal distribution with variance-covariance matrix $\mathbf{G}$ and, in most cases, that the random effects have mean zero.

Random effects are often associated with classification effects, but this is not necessary. As an example of random regression effects, you might want to model the slopes in a growth model

as consisting of two components: an overall (fixed-effects) slope that represents the slope of the average individual, and individual-specific random deviations from the overall slope. The $\mathbf{X}$ and $\mathbf{Z}$ matrix would then have column entries for the regressor variable associated with the slope. You are modeling fixed and randomly varying regression coefficients.

Having random effects in your model has a number of important consequences:

- Some observations are no longer uncorrelated but instead have a covariance that depends on the variance of the random effects.
- You can and should distinguish between the inference spaces; inferences can be drawn in a broad, intermediate, and narrow inference space. In the narrow inference space, conclusions are drawn about the particular values of the random effects selected in the study. The broad inference space applies if inferences are drawn with respect to all possible levels of the random effects. The intermediate inference space can be applied for effects consisting of more than one random term, when inferences are broad with respect to some factors and narrow with respect to others. In fixed-effects models, there is no corresponding concept to the broad and intermediate inference spaces.
- Depending on the structure of $\mathbf{G}$ and $\text{Var}[\boldsymbol{\epsilon}]$ and also subject to the balance in your data, there might be no closed-form solution for the parameter estimates. Although the model is linear in $\boldsymbol{\beta}$, iterative estimation methods might be required to estimate all parameters of the model.
- Certain concepts, such as least squares means and Type III estimable functions, are meaningful only for fixed effects.
- By using random effects, you are modeling variation through variance. Variation in data simply implies that things are not equal. Variance, on the other hand, describes a feature of a random variable. Random effects in your model are random variables: they model variation through variance.

It is important to properly determine the nature of the model effects as fixed or random. An effect is either fixed or random by its very nature; it is improper to consider it fixed in one analysis and random in another depending on what type of results you want to produce. If, for example, a treatment effect is random and you are interested in comparing treatment means, and only the levels selected in the study are of interest, then it is not appropriate to model the treatment effect as fixed so that you can draw on least squares mean analysis. The appropriate strategy is to model the treatment effect as random and to compare the solutions for the treatment effects in the narrow inference space.

In determining whether an effect is fixed or random, it is helpful to inquire about the *genesis* of the effect. If the levels of an effect are randomly sampled, then the effect is a random effect. The following are examples:

- In a large clinical trial, drugs A, B, and C are applied to patients in various clinical centers. If the clinical centers are selected at random from a population of possible clinics, their effect on the response is modeled with a random effect.
- In repeated measures experiments with people or animals as subjects, subjects are declared to be random because they are selected from the larger population to which you want to generalize.

- Fertilizers could be applied at a number of levels. Three levels are randomly selected for an experiment to represent the population of possible levels. The fertilizer effects are random effects.

Quite often it is not possible to select effects at random, or it is not known how the values in the data became part of the study. For example, suppose you are presented with a data set consisting of student scores in three school districts, with four to ten schools in each district and two to three classrooms in each school. How do you decide which effects are fixed and which are random? As another example, in an agricultural experiment conducted in successive years at two locations, how do you decide whether location and year effects are fixed or random? In these situations, the fixed or random nature of the effect might be debatable, bearing out the adage that "one modeler's fixed effect is another modeler's random effect." However, this fact does not constitute license to treat as random those effects that are clearly fixed, or vice versa.

When an effect cannot be randomized or it is not known whether its levels have been randomly selected, it can be a random effect if its impact on the outcome variable is of a stochastic nature—that is, if it is the realization of a random process. Again, this line of thinking relates to the genesis of the effect. A random year, location, or school district effect is a placeholder for different environments that cannot be selected at random but whose effects are the cumulative result of many individual random processes. Note that this argument does not imply that effects are random because the experimenter does not know much about them. The key notion is that effects represent something, whether or not that something is known to the modeler. Broadening the inference space beyond the observed levels is thus possible, although you might not be able to articulate what the realizations of the random effects represent.

A consequence of having random effects in your model is that some observations are no longer uncorrelated but instead have a covariance that depends on the variance of the random effect. In fact, in some modeling applications random effects might be used not only to model heterogeneity in the parameters of a model, but also to induce correlations among observations. The typical assumption about random effects in SAS/STAT software is that the effects are normally distributed.

For more information about mixed modeling tools in SAS/STAT software, see Chapter 6, "Introduction to Mixed Modeling Procedures."

Generalized Linear Models

A class of models that has gained increasing importance in the past several decades is the class of generalized linear models. The theory of generalized linear models originated with Nelder and Wedderburn (1972) and Wedderburn (1974), and was subsequently made popular in the monograph by McCullagh and Nelder (1989). This class of models extends the theory and methods of linear models to data with nonnormal responses. Before this theory was developed, modeling of nonnormal data typically relied on transformations of the data, and the transformations were chosen to improve symmetry, homogeneity of variance, or normality. Such transformations have to be performed with care because they also have implications for the error structure of the model, Also, back-transforming estimates or predicted values can introduce bias.

Generalized linear models also apply a transformation, known as the *link function*, but it is applied to a deterministic component, the mean of the data. Furthermore, generalized linear model take the

distribution of the data into account, rather than assuming that a transformation of the data leads to normally distributed data to which standard linear modeling techniques can be applied.

To put this generalization in place requires a slightly more sophisticated model setup than that required for linear models for normal data:

- The *systematic* component is a linear predictor similar to that in linear models, $\eta = \mathbf{x}'\boldsymbol{\beta}$. The linear predictor is a linear function in the parameters. In contrast to the linear model, η does not represent the mean function of the data.
- The *link function* $g(\cdot)$ relates the linear predictor to the mean, $g(\mu) = \eta$. The link function is a monotonic, invertible function. The mean can thus be expressed as the inversely linked linear predictor, $\mu = g^{-1}(\eta)$. For example, a common link function for binary and binomial data is the logit link, $g(t) = \log\{t/(1-t)\}$. The mean function of a generalized linear model with logit link and a single regressor can thus be written as

$$\log\left\{\frac{\mu}{1-\mu}\right\} = \beta_0 + \beta_1 x$$

$$\mu = \frac{1}{1+\exp\{-\beta_0 - \beta_1 x\}}$$

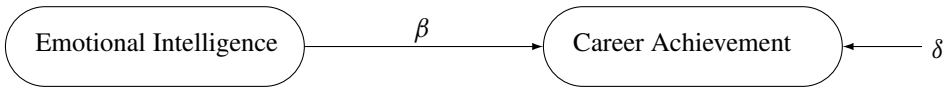
  This is known as a logistic regression model.
- The *random component* of a generalized linear model is the distribution of the data, assumed to be a member of the exponential family of distributions. Discrete members of this family include the Bernoulli (binary), binomial, Poisson, geometric, and negative binomial (for a given value of the scale parameter) distribution. Continuous members include the normal (Gaussian), beta, gamma, inverse gaussian, and exponential distribution.

The standard linear model with normally distributed error is a special case of a generalized linear model; the link function is the identity function and the distribution is normal.

Latent Variable Models

Latent variable modeling involves variables that are not observed directly in your research. It has a relatively long history, dating back from the measure of general intelligence by common factor analysis (Spearman 1904) to the emergence of modern-day structural equation modeling (Jöreskog 1973; Keesling, 1972; Wiley, 1973).

Latent variables are involved in almost all kinds of regression models. In a broad sense, all additive error terms in regression models are latent variables simply because they are not measured in research. Hereafter, however, a narrower sense of latent variables is used when referring to latent variable models. Latent variables are *systematic* unmeasured variables that are also referred to as *factors*. For example, in the following diagram a simple relation between Emotional Intelligence and Career Achievement is shown:



In the diagram, both Emotional Intelligence and Career Achievement are treated as latent factors. They are hypothetical constructs in your model. You hypothesize that Emotional Intelligence is a "causal factor" or predictor of Career Achievement. The symbol β represents the regression coefficient or the effect of Emotional Intelligence on Career Achievement. However, the "causal relationship" or prediction is not perfect. There is an error term δ, which accounts for the unsystematic part of the prediction. You can represent the preceding diagram by using the following linear equation:

$$\mathrm{CA} = \beta \mathrm{EI} + \delta$$

where CA represents Career Achievement and EI represents Emotional Intelligence. The means of the latent factors in the linear model are arbitrary, and so they are assumed to be zero. The error variable δ also has a zero mean with an unknown variance. This equation represents the so-called "structural model," where the "true" relationships among latent factors are theorized.

In order to model this theoretical model with latent factors, some observed variables must somehow relate to these factors. This calls for the measurement models for latent factors. For example, Emotional Intelligence could be measured by some established tests. In these tests, individuals are asked to respond to certain special situations that involve stressful decision making, personal confrontations, and so on. Their responses to these situations are then rated by experts or a standardized scoring system. Suppose there are three such tests and the test scores are labeled as X1, X2 and X3, respectively. The measurement model for the latent factor Emotional Intelligence is specified as follows:

$$\begin{aligned} \mathrm{X1} &= a_1 \mathrm{EI} + e_1 \\ \mathrm{X2} &= a_2 \mathrm{EI} + e_2 \\ \mathrm{X3} &= a_3 \mathrm{EI} + e_3 \end{aligned}$$

where a_1, a_2, and a_3 are regression coefficients and e_1, e_2, and e_3 are measurement errors. Measurement errors are assumed to be independent of the latent factors EI and CA. In the measurement model, X1, X2, and X3 are called the indicators of the latent variable EI. These observed variables are assumed to be centered in the model, and therefore no intercept terms are needed. Each of the indicators is a scaled measurement of the latent factor EI plus a unique error term.

Similarly, you need to have a measurement model for the latent factor CA. Suppose that there are four observed indicators Y1, Y2, Y3, and Y4 (for example, Job Status) for this latent factor. The measurement model for CA is specified as follows:

$$\begin{aligned} \mathrm{Y1} &= a_4 \mathrm{CA} + e_4 \\ \mathrm{Y2} &= a_5 \mathrm{CA} + e_5 \\ \mathrm{Y3} &= a_6 \mathrm{CA} + e_6 \\ \mathrm{Y4} &= a_7 \mathrm{CA} + e_7 \end{aligned}$$

where a_4, a_5, a_6, and a_7 are regression coefficients and e_4, e_5, e_6, and e_7 are error terms. Again, the error terms are assumed to be independent of the latent variables EI and CA, and Y1, Y2, Y3, and Y4 are centered in the equations.

Given the data for the measured variables, you analyze the structural and measurement models simultaneously by the structural equation modeling techniques. In other words, estimation of β, a_1–a_7, and other parameters in the model are carried out simultaneously in the modeling.

Modeling involving the use of latent factors is quite common in social and behavioral sciences, personality assessment, and marketing research. Hypothetical constructs, although not observable, are very important in building theories in these areas.

Another use of latent factors in modeling is to "purify" the predictors in regression analysis. A common assumption in linear regression models is that predictors are measured without errors. That is, in the following linear equation x is assumed to have been measured without errors:

$$y = \alpha + \beta x + \epsilon$$

However, if x has been contaminated with measurement errors that cannot be ignored, the estimate of β might be biased severely so that the true relationship between x and y would be masked.

A measurement model for x provides a solution to such a problem. Let F_x be a "purified" version of x. That is, F_x is the "true" measure of x without measurement errors, as described in the following equation:

$$x = F_x + \delta$$

where δ represents a random measurement error term. Now, the linear relationship of interest is specified in the following new linear regression equation:

$$y = \alpha + \beta F_x + \epsilon$$

In this equation, F_x, which is now free from measurement errors, replaces x in the original equation. With measurement errors taken into account in the simultaneous fitting of the measurement and the new regression equations, estimation of β is unbiased; hence it reflects the true relationship much better.

Certainly, introducing latent factors in models is not a "free lunch." You must pay attention to the identification issues induced by the latent variable methodology. That is, in order to estimate the parameters in structural equation models with latent variables, you must set some identification constraints in these models. There are some established rules or conventions that would lead to proper model identification and estimation. See Chapter 17, "Introduction to Structural Equation Modeling with Latent Variables," for examples and general details.

In addition, because of the nature of latent variables, estimation in structural equation modeling with latent variables does not follow the same form as that of linear regression analysis. Instead of defining the estimators in terms of the data matrices, most estimation methods in structural equation modeling use the fitting of the first- and second- order moments. Hence, estimation principles described in the section "Classical Estimation Principles" on page 42 do not apply to structural equation modeling. However, you can see the section "Estimation Criteria" on page 6888 in Chapter 88, "The TCALIS Procedure (Experimental)," for details about estimation in structural equation modeling with latent variables.

Bayesian Models

Statistical models based on the classical (or *frequentist*) paradigm treat the parameters of the model as fixed, unknown constants. They are not random variables, and the notion of probability is derived in an objective sense as a limiting relative frequency. The Bayesian paradigm takes a different

approach. Model parameters are random variables, and the probability of an event is defined in a subjective sense as the degree to which you believe that the event is true. This fundamental difference in philosophy leads to profound differences in the statistical content of estimation and inference. In the frequentist framework, you use the data to best estimate the unknown value of a parameter; you are trying to pinpoint a value in the parameter space as well as possible. In the Bayesian framework, you use the data to update your beliefs about the *behavior* of the parameter to assess its distributional properties as well as possible.

Suppose you are interested in estimating θ from data $\mathbf{Y} = [Y_1, \cdots, Y_n]$ by using a statistical model described by a density $p(\mathbf{y}|\theta)$. Bayesian philosophy states that θ cannot be determined exactly, and uncertainty about the parameter is expressed through probability statements and distributions. You can say, for example, that θ follows a normal distribution with mean 0 and variance 1, if you believe that this distribution best describes the uncertainty associated with the parameter.

The following steps describe the essential elements of Bayesian inference:

1. A probability distribution for θ is formulated as $\pi(\theta)$, which is known as the *prior* distribution, or just the prior. The prior distribution expresses your beliefs, for example, on the mean, the spread, the skewness, and so forth, about the parameter prior to examining the data.
2. Given the observed data $\mathbf{Y}$, you choose a statistical model $p(\mathbf{y}|\theta)$ to describe the distribution of $\mathbf{Y}$ given θ.
3. You update your beliefs about θ by combining information from the prior distribution and the data through the calculation of the *posterior* distribution, $p(\theta|\mathbf{y})$.

The third step is carried out by using Bayes' theorem, from which this branch of statistical philosophy derives its name. The theorem enables you to combine the prior distribution and the model in the following way:

$$p(\theta|\mathbf{y}) = \frac{p(\theta, \mathbf{y})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\theta)\pi(\theta)}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\theta)\pi(\theta)}{\int p(\mathbf{y}|\theta)\pi(\theta)d\theta}$$

The quantity $p(\mathbf{y}) = \int p(\mathbf{y}|\theta)\pi(\theta)\, d\theta$ is the normalizing constant of the posterior distribution. It is also the marginal distribution of $\mathbf{Y}$, and it is sometimes called the marginal distribution of the data.

The likelihood function of θ is any function proportional to $p(\mathbf{y}|\theta)$—that is, $L(\theta) \propto p(\mathbf{y}|\theta)$. Another way of writing Bayes' theorem is

$$p(\theta|\mathbf{y}) = \frac{L(\theta)\pi(\theta)}{\int L(\theta)\pi(\theta)\, d\theta}$$

The marginal distribution $p(\mathbf{y})$ is an integral; therefore, provided that it is finite, the particular value of the integral does not yield any additional information about the posterior distribution. Hence, $p(\theta|\mathbf{y})$ can be written up to an arbitrary constant, presented here in proportional form, as

$$p(\theta|\mathbf{y}) \propto L(\theta)\pi(\theta)$$

Bayes' theorem instructs you how to update existing knowledge with new information. You start from a prior belief $\pi(\theta)$, and, after learning information from data $\mathbf{y}$, you change or update the

belief on θ and obtain $p(\theta|\mathbf{y})$. These are the essential elements of the Bayesian approach to data analysis.

In theory, Bayesian methods offer a very simple alternative to statistical inference—all inferences follow from the posterior distribution $p(\theta|\mathbf{y})$. However, in practice, only the most elementary problems enable you to obtain the posterior distribution analytically. Most Bayesian analyses require sophisticated computations, including the use of simulation methods. You generate samples from the posterior distribution and use these samples to estimate the quantities of interest.

Both Bayesian and classical analysis methods have their advantages and disadvantages. Your choice of method might depend on the goals of your data analysis. If prior information is available, such as in the form of expert opinion or historical knowledge, and you want to incorporate this information into the analysis, then you might consider Bayesian methods. In addition, if you want to communicate your findings in terms of probability notions that can be more easily understood by nonstatisticians, Bayesian methods might be appropriate. The Bayesian paradigm can provide a framework for answering specific scientific questions that a single point estimate cannot sufficiently address. On the other hand, if you are interested in estimating parameters and in formulating inferences based on the properties of the parameter estimators, then there is no need to use Bayesian analysis. When the sample size is large, Bayesian inference often provides results for parametric models that are very similar to the results produced by classical, frequentist methods.

For more information, see Chapter 7, "Introduction to Bayesian Analysis Procedures."

Classical Estimation Principles

An estimation principle captures the set of rules and procedures by which parameter estimates are derived. When an estimation principle "meets" a statistical model, the result is an *estimation problem*, the solution of which are the parameter estimates. For example, if you apply the estimation principle of least squares to the SLR model $Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$, the estimation problem is to find those values $\widehat{\beta}_0$ and $\widehat{\beta}_1$ that minimize

$$\sum_{i=1}^{n} (y_i - \beta_0 - \beta_1 x_i)^2$$

The solutions are the least squares estimators.

The two most important classes of estimation principles in statistical modeling are the least squares principle and the likelihood principle. All principles have in common that they provide a metric by which you measure the distance between the data and the model. They differ in the nature of the metric; least squares relies on a geometric measure of distance, while likelihood inference is based on a distance that measures plausability.

Least Squares

The idea of the ordinary least squares (OLS) principle is to choose parameter estimates that minimize the squared distance between the data and the model. In terms of the general, additive model,

$$Y_i = f(x_{i1}, \cdots, x_{ik}; \beta_1, \cdots, \beta_p) + \epsilon_i$$

the OLS principle minimizes

$$\text{SSE} = \sum_{i=1}^{n} \left(y_i - f(x_{i1}, \cdots, x_{ik}; \beta_1, \cdots, \beta_p)\right)^2$$

The least squares principle is sometimes called "nonparametric" in the sense that it does not require the distributional specification of the response or the error term, but it might be better termed "distributionally agnostic." In an additive-error model it is only required that the model errors have zero mean. For example, the specification

$$\begin{aligned} Y_i &= \beta_0 + \beta_1 x_i + \epsilon_i \\ \text{E}[\epsilon_i] &= 0 \end{aligned}$$

is sufficient to derive ordinary least squares (OLS) estimators for β_0 and β_1 and to study a number of their properties. It is easy to show that the OLS estimators in this SLR model are

$$\begin{aligned} \widehat{\beta}_1 &= \left(\sum_{i=1}^{n} (Y_i - \overline{Y}) \sum_{i=1}^{n} (x_i - \overline{x})\right) \Big/ \sum_{i=1}^{n} (x_i - \overline{x})^2 \\ \widehat{\beta}_0 &= \overline{Y} - \widehat{\beta}_1 \overline{x} \end{aligned}$$

Based on the assumption of a zero mean of the model errors, you can show that these estimators are unbiased, $\text{E}[\widehat{\beta}_1] = \beta_1$, $\text{E}[\widehat{\beta}_0] = \beta_0$. However, without further assumptions about the distribution of the ϵ_i, you cannot derive the variability of the least squares estimators or perform statistical inferences such as hypothesis tests or confidence intervals. In addition, depending on the distribution of the ϵ_i, other forms of least squares estimation can be more efficient than OLS estimation.

The conditions for which ordinary least squares estimation is efficient are zero mean, homoscedastic, uncorrelated model errors. Mathematically,

$$\begin{aligned} \text{E}[\epsilon_i] &= 0 \\ \text{Var}[\epsilon_i] &= \sigma^2 \\ \text{Cov}[\epsilon_i, \epsilon_j] &= 0 \text{ if } i \neq j \end{aligned}$$

The second and third assumption are met if the errors have an *iid* distribution—that is, if they are independent and identically distributed. Note, however, that the notion of stochastic independence is stronger than that of absence of correlation. Only if the data are normally distributed does the latter implies the former.

The various other forms of the least squares principle are motivated by different extensions of these assumptions in order to find more efficient estimators.

Weighted Least Squares

The objective function in weighted least squares (WLS) estimation is

$$\text{SSE}_w = \sum_{i=1}^{n} w_i \left(Y_i - f(x_{i1}, \cdots, x_{ik}; \beta_1, \cdots, \beta_p)\right)^2$$

where w_i is a weight associated with the ith observation. A situation where WLS estimation is appropriate is when the errors are uncorrelated but not homoscedastic. If the weights for the observations are proportional to the reciprocals of the error variances, $\text{Var}[\epsilon_i] = \sigma^2/w_i$, then the weighted least squares estimates are best linear unbiased estimators (BLUE). Suppose that the weights w_i are collected in the diagonal matrix $\mathbf{W}$ and that the mean function has the form of a linear model. The weighted sum of squares criterion then can be written as

$$\text{SSE}_w = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})' \mathbf{W} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$$

which gives rise to the weighted normal equations

$$(\mathbf{X}'\mathbf{W}\mathbf{X})\boldsymbol{\beta} = \mathbf{X}'\mathbf{W}\mathbf{Y}$$

The resulting WLS estimator of $\boldsymbol{\beta}$ is

$$\widehat{\boldsymbol{\beta}}_w = \left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-} \mathbf{X}'\mathbf{W}\mathbf{Y}$$

Iteratively Reweighted Least Squares

If the weights in a least squares problem depend on the parameters, then a change in the parameters also changes the weight structure of the model. Iteratively reweighted least squares (IRLS) estimation is an iterative technique that solves a series of weighted least squares problems, where the weights are recomputed between iterations. IRLS estimation can be used, for example, to derive maximum likelihood estimates in generalized linear models.

Generalized Least Squares

The previously discussed least squares methods have in common that the observations are assumed to be uncorrelated—that is, $\text{Cov}[\epsilon_i, \epsilon_j] = 0$, whenever $i \neq j$. The weighted least squares estimation problem is a special case of a more general least squares problem, where the model errors have a general convariance matrix, $\text{Var}[\boldsymbol{\epsilon}] = \boldsymbol{\Sigma}$. Suppose again that the mean function is linear, so that the model becomes

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \qquad \boldsymbol{\epsilon} \sim (\mathbf{0}, \boldsymbol{\Sigma})$$

The generalized least squares (GLS) principle is to minimize the generalized error sum of squares

$$\text{SSE}_g = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})' \boldsymbol{\Sigma}^{-1} (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$$

This leads to the generalized normal equations

$$(\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{X})\boldsymbol{\beta} = \mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{Y}$$

and the GLS estimator

$$\widehat{\boldsymbol{\beta}}_g = \left(\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{X}\right)^{-}\mathbf{X}'\boldsymbol{\Sigma}^{-1}\mathbf{Y}$$

Obviously, WLS estimation is a special case of GLS estimation, where $\boldsymbol{\Sigma} = \sigma^2\mathbf{W}^{-1}$—that is, the model is

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \qquad \boldsymbol{\epsilon} \sim \left(\mathbf{0}, \sigma^2\mathbf{W}^{-1}\right)$$

Likelihood

There are several forms of likelihood estimation and a large number of offshoot principles derived from it, such as pseudo-likelihood, quasi-likelihood, composite likelihood, etc. The basic likelihood principle is *maximum likelihood*, which asks to estimate the model parameters by those quantities that maximize the likelihood function of the data. The likelihood function is the joint distribution of the data, but in contrast to a probability mass or density function, it is thought of as a function of the parameters, given the data. The heuristic appeal of the maximum likelihood estimates (MLE) is that these are the values that make the observed data "most likely." Especially for discrete response data, the value of the likelihood function is the ordinate of a probability mass function, even if the likelihood is not a probability function. Since a statistical model is thought of as a representation of the data-generating mechanism, what could be more preferable as parameter estimates than those values that make it most likely that the data at hand will be observed?

Maximum likelihood estimates, if they exist, have appealing statistical properties. Under fairly mild conditions, they are best-asymptotic-normal (BAN) estimates—that is, their asymptotic distribution is normal, and no other estimator has a smaller asymptotic variance. However, their statistical behavior in finite samples is often difficult to establish, and you have to appeal to the asymptotic results that hold as the sample size tends to infinity. For example, maximum likelihood estimates are often biased estimates and the bias disappears as the sample size grows. A famous example is random sampling from a normal distribution. The corresponding statistical model is

$$\begin{aligned} Y_i &= \mu + \epsilon_i \\ \epsilon_i &\sim \textit{iid}\ N(0, \sigma^2) \end{aligned}$$

where the symbol $\sim$ is read as "is distributed as" and *iid* is read as "independent and identically distributed." Under the normality assumption, the density function of y_i is

$$f(y_i; \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2}\left(\frac{y_i - \mu}{\sigma}\right)^2\right\}$$

and the likelihood for a random sample of size n is

$$L(\mu, \sigma^2; \mathbf{y}) = \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2}\left(\frac{y_i - \mu}{\sigma}\right)^2\right\}$$

Maximizing the likelihood function $L(\mu, \sigma^2; \mathbf{y})$ is equivalent to maximizing the log-likelihood function $\log L = l(\mu, \sigma^2; \mathbf{y})$,

$$\begin{aligned} l(\mu, \sigma^2; \mathbf{y}) &= \sum_{i=1}^{n} -\frac{1}{2}\left(\log\{2\pi\} + \frac{(y_i - \mu)^2}{\sigma^2} + \log\{\sigma^2\}\right) \\ &= -\frac{1}{2}\left(n\log\{2\pi\} + n\log\{\sigma^2\} + \sum_{i=1}^{n}(y_i - \mu)^2/\sigma^2\right) \end{aligned}$$

The maximum likelihood estimators of μ and σ^2 are thus

$$\widehat{\mu} = \frac{1}{n}\sum_{i=1}^{n} Y_i = \overline{Y} \qquad \widehat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n} (Y_i - \widehat{\mu})^2$$

The MLE of the mean μ is the sample mean, and it is an unbiased estimator of μ. However, the MLE of the variance σ^2 is not an unbiased estimator. It has bias

$$\mathrm{E}\left[\widehat{\sigma}^2 - \sigma^2\right] = -\frac{1}{n}\sigma^2$$

As the sample size n increases, the bias vanishes.

For certain classes of models, special forms of likelihood estimation have been developed to maintain the appeal of likelihood-based statistical inference and to address specific properties that are believed to be shortcomings:

- The bias in maximum likelihood parameter estimators of variances and covariances has led to the development of restricted (or residual) maximum likelihood (REML) estimators that play an important role in mixed models.
- Quasi-likelihood methods do not require that the joint distribution of the data be specified. These methods derive estimators based on only the first two moments (mean and variance) of the joint distributions and play an important role in the analysis of correlated data.
- The idea of composite likelihood is applied in situations where the likelihood of the vector of responses is intractable but the likelihood of components or functions of the full-data likelihood are tractable. For example, instead of the likelihood of $\mathbf{Y}$, you might consider the likelihood of pairwise differences $Y_i - Y_j$.
- The pseudo-likelihood concept is also applied when the likelihood function is intractable, but the likelihood of a related, simpler model is available. An important difference between quasi-likelihood and pseudo-likelihood techniques is that the latter make distributional assumptions to obtain a likelihood function in the pseudo-model. Quasi-likelihood methods do not specify the distributional family.
- The penalized likelihood principle is applied when additional constraints and conditions need to be imposed on the parameter estimates or the resulting model fit. For example, you might augment the likelihood with conditions that govern the smoothness of the predictions or that prevent overfitting of the model.

Least Squares or Likelihood

For many statistical modeling problems, you have a choice between a least squares principle and the maximum likelihood principle. Table 3.1 compares these two basic principles.

Table 3.1 Least Squares and Maximum Likelihood

Criterion	Least Squares	Maximum Likelihood
Requires specification of joint distribution of data	No, but in order to perform confirmatory inference (tests, confidence intervals), a distributional assumption is needed, or an appeal to asymptotics.	Yes, no progress can be made with the genuine likelihood principle without knowing the distribution of the data.
All parameters of the model are estimated	No. In the additive-error type models, least squares provides estimates of only the parameters in the mean function. The residual variance, for example, must be estimated by some other method—typically by using the mean squared error of the model.	Yes
Estimates always exist	Yes, but they might not be unique, such as when the $\mathbf{X}$ matrix is singular.	No, maximum likelihood estimates do not exist for all estimation problems.
Estimators are biased	Unbiased, provided that the model is correct—that is, the errors have zero mean.	Often biased, but asymptotically unbiased
Estimators are consistent	Not necessarily, but often true. Sometimes estimators are consistent even in a misspecified model, such as when misspecification is in the covariance structure.	Almost always
Estimators are best linear unbiased estimates (BLUE)	Typically, if the least squares assumptions are met.	Not necessarily: estimators are often nonlinear in the data and are often biased.
Asymptotically most efficient	Not necessarily	Typically
Easy to compute	Yes	No

Inference Principles for Survey Data

Design-based and model-assisted statistial inference for survey data requires that the randomness due to the selection mechanism be taken into account. This can require special estimation principles and techniques.

The SURVEYMEANS, SURVEYFREQ, SURVEYREG, and SURVEYLOGISTIC procedures support design-based and/or model-assisted inference for sample surveys. Suppose π_i is the selection probability for unit i in sample S. The inverse of the inclusion probability is known as sampling weight and is denoted by w_i. Briefly, the idea is to apply a relationship that exists in the population to the sample and to take into account the sampling weights. For example, to estimate the finite population total $T_N = \sum_{i \in U_N} y_i$ based on the sample S, you can accumulate the sampled values while properly weighting: $\widehat{T}_\pi = \sum_{i \in S} w_i y_i$. It is easy to verify that $\widehat{T}_\pi$ is design-unbiased in the sense that $\mathrm{E}[\widehat{T}_\pi | \mathcal{F}_N] = T_N$ (see Cochran 1997).

When a statistical model is present, similar ideas apply. For example, if β_{N0} and β_{N1} are finite population quantities for a simple linear regression working model that minimize the sum of squares

$$\sum_{i \in U_N} (y_i - \beta_{0N} - \beta_{1N} x_i)^2$$

in the population, then the sample-based estimators $\widehat{\beta}_{0S}$ and $\widehat{\beta}_{1S}$ are obtained by minimizing the weighted sum of squares

$$\sum_{i \in S} w_i (y_i - \hat{\beta}_{0S} - \hat{\beta}_{1S} x_i)^2$$

in the sample, taking into account the inclusion probabilities.

In model-assisted inference, weighted least squares or pseudo-maximum likelihood estimators are commonly used to solve such estimation problems. Maximum pseudo-likelihood or weighted maximum likelihood estimators for survey data maximize a sample-based estimator of the population likelihood. Assume a working model with uncorrelated responses such that the finite population log-likelihood is

$$\sum_{i \in U_N} l(\theta_{1N}, \ldots, \theta_{pN}; y_i),$$

where $\theta_{1N}, \ldots, \theta_{pN}$ are finite population quantities. For independent sampling, one possible sample-based estimator of the population log likelihood is

$$\sum_{i \in S} w_i l(\theta_{1N}, \ldots, \theta_{pN}; y_i)$$

Sample-based estimators $\widehat{\theta}_{1S}, \ldots, \widehat{\theta}_{pS}$ are obtained by maximizing this expression.

Design-based and model-based statistical analysis might employ the same statistical model (for example, a linear regression) and the same estimation principle (for example, weighted least squares), and arrive at the same estimates. The design-based estimation of the precision of the estimators differs from the model-based estimation, however. For complex surveys, design-based variance estimates are in general different from their model-based counterpart. The SAS/STAT procedures for survey data (SURVEYMEANS, SURVEYFREQ, SURVEYREG, and SURVEYLOGISTIC procedures) compute design-based variance estimates for complex survey data. See the section "Variance Estimation" on page 267, in Chapter 14, "Introduction to Survey Procedures," for details about design-based variance estimation.

Statistical Background

Hypothesis Testing and Power

In statistical hypothesis testing, you typically express the belief that some effect exists in a population by specifying an alternative hypothesis H_1. You state a null hypothesis H_0 as the assertion that the effect does *not* exist and attempt to gather evidence to reject H_0 in favor of H_1. Evidence is gathered in the form of sample data, and a statistical test is used to assess H_0. If H_0 is rejected but there really is *no* effect, this is called a *Type I error*. The probability of a Type I error is usually designated "alpha" or α, and statistical tests are designed to ensure that α is suitably small (for example, less than 0.05).

If there is an effect in the population but H_0 is *not* rejected in the statistical test, then a *Type II error* has been committed. The probability of a Type II error is usually designated "beta" or β. The probability $1-\beta$ of avoiding a Type II error—that is, correctly rejecting H_0 and achieving statistical significance, is called the *power* of the test.

An important goal in study planning is to ensure an acceptably high level of power. Sample size plays a prominent role in power computations because the focus is often on determining a sufficient sample size to achieve a certain power, or assessing the power for a range of different sample sizes.

There are several tools available in SAS/STAT software for power and sample size analysis. PROC POWER covers a variety of analyses such as t tests, equivalence tests, confidence intervals, binomial proportions, multiple regression, one-way ANOVA, survival analysis, logistic regression, and the Wilcoxon rank-sum test. PROC GLMPOWER supports more complex linear models. The Power and Sample Size application provides a user interface and implements many of the analyses supported in the procedures.

Important Linear Algebra Concepts

A *matrix* $\mathbf{A}$ is a rectangular array of numbers. The *order* of a matrix with n rows and k columns is $(n \times k)$. The element in row i, column j of $\mathbf{A}$ is denoted as a_{ij}, and the notation $\left[a_{ij}\right]$ is sometimes used to refer to the two-dimensional row-column array

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1k} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2k} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3k} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \cdots & a_{nk} \end{bmatrix} = \left[a_{ij}\right]$$

A *vector* is a one-dimensional array of numbers. A *column vector* has a single column ($k = 1$). A *row vector* has a single row ($n = 1$). A *scalar* is a matrix of order (1×1)—that is, a single number.

A *square* matrix has the same row and column order, $n = k$. A *diagonal* matrix is a square matrix where all off-diagonal elements are zero, $a_{ij} = 0$ if $i \neq j$. The *identity* matrix $\mathbf{I}$ is a diagonal matrix with $a_{ii} = 1$ for all i. The *unit vector* $\mathbf{1}$ is a vector where all elements are 1. The *unit matrix* $\mathbf{J}$ is a matrix of all 1s. Similarly, the elements of the null vector and the null matrix are all 0.

Basic matrix operations are as follows:

Addition — If $\mathbf{A}$ and $\mathbf{B}$ are of the same order, then $\mathbf{A} + \mathbf{B}$ is the matrix of elementwise sums,

$$\mathbf{A} + \mathbf{B} = \left[a_{ij} + b_{ij}\right]$$

Subtraction — If $\mathbf{A}$ and $\mathbf{B}$ are of the same order, then $\mathbf{A} - \mathbf{B}$ is the matrix of elementwise differences,

$$\mathbf{A} - \mathbf{B} = \left[a_{ij} - b_{ij}\right]$$

Dot product — The dot product of two n-vectors $\mathbf{a}$ and $\mathbf{b}$ is the sum of their elementwise products,

$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{n} a_i b_i$$

The dot product is also known as the *inner product* of $\mathbf{a}$ and $\mathbf{b}$. Two vectors are said to be *orthogonal* if their dot product is zero.

Multiplication — Matrices $\mathbf{A}$ and $\mathbf{B}$ are said to be conformable for $\mathbf{AB}$ multiplication if the number of columns in $\mathbf{A}$ equals the number of rows in $\mathbf{B}$. Suppose that $\mathbf{A}$ is of order $(n \times k)$ and that $\mathbf{B}$ is of order $(k \times p)$. The product $\mathbf{AB}$ is then defined as the $(n \times p)$ matrix of the dot products of the ith row of $\mathbf{A}$ and the jth column of $\mathbf{B}$,

$$\mathbf{AB} = \left[\mathbf{a_i} \cdot \mathbf{b_j}\right]_{n \times p}$$

Transposition — The transpose of the $(n \times k)$ matrix $\mathbf{A}$ is denoted as $\mathbf{A}'$ or $\mathbf{A}^T$ or $\mathbf{A}^{\mathrm{T}}$ and is obtained by interchanging the rows and columns,

$$\mathbf{A}' = \begin{bmatrix} a_{11} & a_{21} & a_{31} & \cdots & a_{n1} \\ a_{12} & a_{22} & a_{23} & \cdots & a_{n2} \\ a_{13} & a_{23} & a_{33} & \cdots & a_{n3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{1k} & a_{2k} & a_{3k} & \cdots & a_{nk} \end{bmatrix} = \left[a_{ji}\right]$$

A *symmetric* matrix is equal to its transpose, $\mathbf{A} = \mathbf{A}'$. The inner product of two $(n \times 1)$ column vectors $\mathbf{a}$ and $\mathbf{b}$ is $\mathbf{a} \cdot \mathbf{b} = \mathbf{a}'\mathbf{b}$.

Matrix Inversion

Regular Inverses

The right inverse of a matrix $\mathbf{A}$ is the matrix that yields the identity when $\mathbf{A}$ is postmultiplied by it. Similarly, the left inverse of $\mathbf{A}$ yields the identity if $\mathbf{A}$ is premultiplied by it. $\mathbf{A}$ is said to be

invertible and $\mathbf{B}$ is said to be the inverse of $\mathbf{A}$, if $\mathbf{B}$ is its right and left inverse, $\mathbf{BA} = \mathbf{AB} = \mathbf{I}$. This requires $\mathbf{A}$ to be square and nonsingular. The inverse of a matrix $\mathbf{A}$ is commonly denoted as $\mathbf{A}^{-1}$. The following results are useful in manipulating inverse matrices (assuming both $\mathbf{A}$ and $\mathbf{C}$ are invertible):

$$\begin{aligned}
\mathbf{A}\mathbf{A}^{-1} &= \mathbf{A}^{-1}\mathbf{A} = \mathbf{I} \\
\left(\mathbf{A}'\right)^{-1} &= \left(\mathbf{A}^{-1}\right)' \\
\left(\mathbf{A}^{-1}\right)^{-1} &= \mathbf{A} \\
(\mathbf{AC})^{-1} &= \mathbf{C}^{-1}\mathbf{A}^{-1} \\
\text{rank}(\mathbf{A}) &= \text{rank}\left(\mathbf{A}^{-1}\right)
\end{aligned}$$

If $\mathbf{D}$ is a diagonal matrix with nonzero entries on the diagonal—that is, $\mathbf{D} = \text{diag}\,(d_1, \cdots, d_n)$—then $\mathbf{D}^{-1} = \text{diag}\,(1/d_1, \cdots, 1/d_n)$. If $\mathbf{D}$ is a block-diagonal matrix whose blocks are invertible, then

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_3 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D}_n \end{bmatrix} \qquad \mathbf{D}^{-1} = \begin{bmatrix} \mathbf{D}_1^{-1} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{D}_2^{-1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{D}_3^{-1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{D}_n^{-1} \end{bmatrix}$$

In statistical applications the following two results are particularly important, because they can significantly reduce the computational burden in working with inverse matrices.

Partitioned Matrix Suppose $\mathbf{A}$ is a nonsingular matrix that is partitioned as

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}$$

Then, provided that all the inverses exist, the inverse of $\mathbf{A}$ is given by

$$\mathbf{A}^{-1} = \begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{12} \\ \mathbf{B}_{21} & \mathbf{B}_{22} \end{bmatrix}$$

where $\mathbf{B}_{11} = \left(\mathbf{A}_{11} - \mathbf{A}_{12}\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\right)^{-1}$, $\mathbf{B}_{12} = -\mathbf{B}_{11}\mathbf{A}_{12}\mathbf{A}_{22}^{-1}$, $\mathbf{B}_{21} = -\mathbf{A}_{22}^{-1}\mathbf{A}_{21}\mathbf{B}_{11}$, and $\mathbf{B}_{22} = \left(\mathbf{A}_{22} - \mathbf{A}_{21}\mathbf{A}_{11}^{-1}\mathbf{A}_{12}\right)^{-1}$.

Patterned Sum Suppose $\mathbf{R}$ is $(n \times n)$ nonsingular, $\mathbf{G}$ is $(k \times k)$ nonsingular, and $\mathbf{B}$ and $\mathbf{C}$ are $(n \times k)$ and $(k \times n)$ matrices, respectively. Then the inverse of $\mathbf{R} + \mathbf{BGC}$ is given by

$$(\mathbf{R} + \mathbf{BGC})^{-1} = \mathbf{R}^{-1} - \mathbf{R}^{-1}\mathbf{B}\left(\mathbf{G}^{-1} + \mathbf{C}\mathbf{R}^{-1}\mathbf{B}\right)^{-1}\mathbf{C}\mathbf{R}^{-1}$$

This formula is particularly useful if $k << n$ and $\mathbf{R}$ has a simple form that is easy to invert. This case arises, for example, in mixed models where $\mathbf{R}$ might be a diagonal or block-diagonal matrix, and $\mathbf{B} = \mathbf{C}'$.

Another situation where this formula plays a critical role is in the computation of regression diagnostics, such as in determining the effect of removing an observation from the analysis. Suppose that $\mathbf{A} = \mathbf{X}'\mathbf{X}$ represents the crossproduct

matrix in the linear model $\mathrm{E}[\mathbf{Y}] = \mathbf{X}\boldsymbol{\beta}$. If $\mathbf{x}_i'$ is the ith row of the $\mathbf{X}$ matrix, then $(\mathbf{X}'\mathbf{X} - \mathbf{x}_i\mathbf{x}_i')$ is the crossproduct matrix in the same model with the ith observation removed. Identifying $\mathbf{B} = -\mathbf{x}_i$, $\mathbf{C} = \mathbf{x}_i'$, and $\mathbf{G} = \mathbf{I}$ in the preceding inversion formula, you can obtain the expression for the inverse of the crossproduct matrix:

$$\left(\mathbf{X}'\mathbf{X} - \mathbf{x}_i\mathbf{x}_i'\right)^{-1} = \mathbf{X}'\mathbf{X} + \frac{\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{x}_i\mathbf{x}_i'\left(\mathbf{X}'\mathbf{X}\right)^{-1}}{1 - \mathbf{x}_i'\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{x}_i}$$

This expression for the inverse of the reduced data crossproduct matrix enables you to compute "leave-one-out" deletion diagnostics in linear models without refitting the model.

Generalized Inverse Matrices

If $\mathbf{A}$ is rectangular (not square) or singular, then it is not invertible and the matrix $\mathbf{A}^{-1}$ does not exist. Suppose you want to find a solution to simultaneous linear equations of the form

$$\mathbf{A}\mathbf{b} = \mathbf{c}$$

If $\mathbf{A}$ is square and nonsingular, then the unique solution is $\mathbf{b} = \mathbf{A}^{-1}\mathbf{c}$. In statistical applications, the case where $\mathbf{A}$ is $(n \times k)$ rectangular is less important than the case where $\mathbf{A}$ is a $(k \times k)$ square matrix of rank less than k. For example, the normal equations in ordinary least squares (OLS) estimation in the model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ are

$$\left(\mathbf{X}'\mathbf{X}\right)\boldsymbol{\beta} = \mathbf{X}'\mathbf{Y}$$

A *generalized inverse* matrix is a matrix $\mathbf{A}^-$ such that $\mathbf{A}^-\mathbf{c}$ is a solution to the linear system. In the OLS example, a solution can be found as $\left(\mathbf{X}'\mathbf{X}\right)^-\mathbf{X}'\mathbf{Y}$, where $\left(\mathbf{X}'\mathbf{X}\right)^-$ is a generalized inverse of $\mathbf{X}'\mathbf{X}$.

The following four conditions are often associated with generalized inverses. For the square or rectangular matrix $\mathbf{A}$ there exist matrices $\mathbf{G}$ that satisfy

$$\begin{array}{lrcl} \text{(i)} & \mathbf{AGA} & = & \mathbf{A} \\ \text{(ii)} & \mathbf{GAG} & = & \mathbf{G} \\ \text{(iii)} & (\mathbf{AG})' & = & \mathbf{AG} \\ \text{(iv)} & (\mathbf{GA})' & = & \mathbf{GA} \end{array}$$

The matrix $\mathbf{G}$ that satisfies all four conditions is unique and is called the Moore-Penrose inverse, after the first published work on generalized inverses by Moore (1920) and the subsequent definition by Penrose (1955). Only the first condition is required, however, to provide a solution to the linear system above.

Pringle and Rayner (1971) introduced a numbering system to distinguish between different types of generalized inverses. A matrix that satisfies only condition (i) is a g_1-inverse. The g_2-inverse satistifes conditions (i) and (ii). It is also called a *reflexive* generalized inverse. Matrices satisfying conditions (i)–(iii) or conditions (i), (ii), and (iv) are g_3-inverses. Note that a matrix that satisfies the first three conditions is a right generalized inverse, and a matrix that satisfies conditions (i), (ii), and (iv) is a left generalized inverse. For example, if $\mathbf{B}$ is $(n \times k)$ of rank k, then $\left(\mathbf{B}'\mathbf{B}\right)^{-1}\mathbf{B}'$ is

a left generalized inverse of **B**. The notation g_4-inverse for the Moore-Penrose inverse, satisfying conditions (i)–(iv), is often used by extension, but note that Pringle and Rayner (1971) do not use it; rather, they call such a matrix "the" generalized inverse.

If the $(n \times k)$ matrix **X** is rank-deficient—that is, $\text{rank}(\mathbf{X}) < \min\{n, k\}$—then the system of equations

$$\left(\mathbf{X}'\mathbf{X}\right)\boldsymbol{\beta} = \mathbf{X}'\mathbf{Y}$$

does not have a unique solution. A particular solution depends on the choice of the generalized inverse. However, some aspects of the statistical inference are invariant to the choice of the generalized inverse. If **G** is a generalized inverse of $\mathbf{X}'\mathbf{X}$, then $\mathbf{X}\mathbf{G}\mathbf{X}'$ is invariant to the choice of **G**. This result comes into play, for example, when you are computing predictions in an OLS model with a rank-deficient **X** matrix, since it implies that the predicted values

$$\mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-}\mathbf{X}'\mathbf{y}$$

are invariant to the choice of $\left(\mathbf{X}'\mathbf{X}\right)^{-}$.

Matrix Differentiation

Taking the derivative of expressions involving matrices is a frequent task in statistical estimation. Objective functions that are to be minimized or maximized are usually written in terms of model matrices and/or vectors whose elements depend on the unknowns of the estimation problem. Suppose that **A** and **B** are real matrices whose elements depend on the scalar quantities β and θ—that is, $\mathbf{A} = \left[a_{ij}(\beta, \theta)\right]$, and similarly for **B**.

The following are useful results in finding the derivative of elements of a matrix and of functions involving a matrix. For more in-depth discussion of matrix differentiation and matrix calculus, see, for example, Magnus and Neudecker (1999) and Harville (1997).

The derivative of **A** with respect to β is denoted $\dot{\mathbf{A}}_{\beta}$ and is the matrix of the first derivatives of the elements of **A**:

$$\dot{\mathbf{A}}_{\beta} = \frac{\partial}{\partial\beta}\mathbf{A} = \left[\frac{\partial a_{ij}(\beta, \theta)}{\partial\beta}\right]$$

Similarly, the second derivative of **A** with respect to β and θ is the matrix of the second derivatives

$$\ddot{\mathbf{A}}_{\beta\theta} = \frac{\partial^2}{\partial\beta\partial\theta}\mathbf{A} = \left[\frac{\partial^2 a_{ij}(\beta, \theta)}{\partial\beta\partial\theta}\right]$$

The following are some basic results involving sums, products, and traces of matrices:

$$\begin{aligned}
\frac{\partial}{\partial\beta}c_1\mathbf{A} &= c_1\dot{\mathbf{A}}_\beta \\
\frac{\partial}{\partial\beta}(\mathbf{A}+\mathbf{B}) &= \dot{\mathbf{A}}_\beta + \dot{\mathbf{B}}_\beta \\
\frac{\partial}{\partial\beta}(c_1\mathbf{A}+c_2\mathbf{B}) &= c_1\dot{\mathbf{A}}_\beta + c_2\dot{\mathbf{B}}_\beta \\
\frac{\partial}{\partial\beta}\mathbf{AB} &= \mathbf{A}\dot{\mathbf{B}}_\beta + \dot{\mathbf{A}}_\beta\mathbf{B} \\
\frac{\partial}{\partial\beta}\text{trace}(\mathbf{A}) &= \text{trace}\left(\dot{\mathbf{A}}_\beta\right) \\
\frac{\partial}{\partial\beta}\text{trace}(\mathbf{AB}) &= \text{trace}\left(\mathbf{A}\dot{\mathbf{B}}_\beta\right) + \text{trace}\left(\dot{\mathbf{A}}_\beta\mathbf{B}\right)
\end{aligned}$$

The next set of results is useful in finding the derivative of elements of $\mathbf{A}$ and of functions of $\mathbf{A}$, if $\mathbf{A}$ is a nonsingular matrix:

$$\begin{aligned}
\frac{\partial}{\partial\beta}\mathbf{x}'\mathbf{A}^{-1}\mathbf{x} &= -\mathbf{x}'\mathbf{A}^{-1}\dot{\mathbf{A}}_\beta\mathbf{A}^{-1}\mathbf{x} \\
\frac{\partial}{\partial\beta}\mathbf{A}^{-1} &= -\mathbf{A}^{-1}\dot{\mathbf{A}}_\beta\mathbf{A}^{-1} \\
\frac{\partial}{\partial\beta}|\mathbf{A}| &= |\mathbf{A}|\,\text{trace}\left(\mathbf{A}^{-1}\dot{\mathbf{A}}_\beta\right) \\
\frac{\partial}{\partial\beta}\log\{|\mathbf{A}|\} &= \frac{1}{|\mathbf{A}|}\frac{\partial}{\partial\beta}\mathbf{A} = \text{trace}\left(\mathbf{A}^{-1}\dot{\mathbf{A}}_\beta\right) \\
\frac{\partial^2}{\partial\beta\,\partial\theta}\mathbf{A}^{-1} &= -\mathbf{A}^{-1}\ddot{\mathbf{A}}_{\beta\theta}\mathbf{A}^{-1} + \mathbf{A}^{-1}\dot{\mathbf{A}}_\beta\mathbf{A}^{-1}\dot{\mathbf{A}}_\theta\mathbf{A}^{-1} + \mathbf{A}^{-1}\dot{\mathbf{A}}_\theta\mathbf{A}^{-1}\dot{\mathbf{A}}_\beta\mathbf{A}^{-1} \\
\frac{\partial^2}{\partial\beta\,\partial\theta}\log\{|\mathbf{A}|\} &= \text{trace}\left(\mathbf{A}^{-1}\ddot{\mathbf{A}}_{\beta\theta}\right) - \text{trace}\left(\mathbf{A}^{-1}\dot{\mathbf{A}}_\beta\mathbf{A}^{-1}\dot{\mathbf{A}}_\theta\right)
\end{aligned}$$

Now suppose that $\mathbf{a}$ and $\mathbf{b}$ are column vectors that depend on β and/or θ and that $\mathbf{x}$ is a vector of constants. The following results are useful for manipulating derivatives of linear and quadratic forms:

$$\begin{aligned}
\frac{\partial}{\partial\mathbf{x}}\mathbf{a}'\mathbf{x} &= \mathbf{a} \\
\frac{\partial}{\partial\mathbf{x}'}\mathbf{B}\mathbf{x} &= \mathbf{B} \\
\frac{\partial}{\partial\mathbf{x}}\mathbf{x}'\mathbf{B}\mathbf{x} &= \left(\mathbf{B}+\mathbf{B}'\right)\mathbf{x} \\
\frac{\partial^2}{\partial\mathbf{x}\partial\mathbf{x}'}\mathbf{x}'\mathbf{B}\mathbf{x} &= \mathbf{B}+\mathbf{B}'
\end{aligned}$$

Matrix Decompositions

To decompose a matrix is to express it as a function—typically a product—of other matrices that have particular properties such as orthogonality, diagonality, triangularity. For example, the Cholesky decomposition of a symmetric positive definite matrix $\mathbf{A}$ is $\mathbf{CC}' = \mathbf{A}$, where $\mathbf{C}$ is a lower-triangular matrix. The spectral decomposition of a symmetric matrix is $\mathbf{A} = \mathbf{PDP}'$, where $\mathbf{D}$ is a diagonal matrix and $\mathbf{P}$ is an orthogonal matrix.

Matrix decomposition play an important role in statistical theory as well as in statistical computations. Calculations in terms of decompositions can have greater numerical stability. Decompositions are often necessary to extract information about matrices, such as matrix rank, eigenvalues, or eigenvectors. Decompositions are also used to form special transformations of matrices, such as to form a "square-root" matrix. This section briefly mentions several decompositions that are particularly prevalent and important.

LDU, LU, and Cholesky Decomposition

Every square matrix $\mathbf{A}$, whether it is positive definite or not, can be expressed in the form $\mathbf{A} = \mathbf{LDU}$, where $\mathbf{L}$ is a unit lower-triangular matrix, $\mathbf{D}$ is a diagonal matrix, and $\mathbf{U}$ is a unit upper-triangular matrix. (The diagonal elements of a unit triangular matrix are 1.) Because of the arrangement of the matrices, the decomposition is called the LDU decomposition. Since you can absorb the diagonal matrix into the triangular matrices, the decomposition

$$\mathbf{A} = \mathbf{LD}^{1/2}\mathbf{D}^{1/2}\mathbf{U} = \mathbf{L}^*\mathbf{U}^*$$

is also referred to as the LU decomposition of $\mathbf{A}$.

If the matrix $\mathbf{A}$ is positive definite, then the diagonal elements of $\mathbf{D}$ are positive and the LDU decomposition is unique. Furthermore, we can add more specificity to this result in that for a symmetric, positive definite matrix, there is a unique decomposition $\mathbf{A} = \mathbf{U}'\mathbf{DU}$, where $\mathbf{U}$ is unit upper-triangular and $\mathbf{D}$ is diagonal with positive elements. Absorbing the square root of $\mathbf{D}$ into $\mathbf{U}$, $\mathbf{C} = \mathbf{D}^{1/2}\mathbf{U}$, the decomposition is known as the *Cholesky* decomposition of a positive-definite matrix:

$$\mathbf{B} = \mathbf{U}'\mathbf{D}^{1/2}\,\mathbf{D}^{1/2}\mathbf{U} = \mathbf{C}'\mathbf{C}$$

where $\mathbf{C}$ is upper triangular.

If $\mathbf{B}$ is $(n \times n)$ symmetric nonnegative definite of rank k, then we can extend the Cholesky decomposition as follows. Let $\mathbf{C}^*$ denote the lower-triangular matrix such that

$$\mathbf{C}^* = \begin{bmatrix} \mathbf{C}_{k\times k} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

Then $\mathbf{B} = \mathbf{CC}'$.

Spectral Decomposition

Suppose that $\mathbf{A}$ is an $(n \times n)$ symmetric matrix. Then there exists an orthogonal matrix $\mathbf{Q}$ and a diagonal matrix $\mathbf{D}$ such that $\mathbf{A} = \mathbf{QDQ}'$. Of particular importance is the case where the orthogonal

matrix is also orthonormal—that is, its column vectors have unit norm. Denote this orthonormal matrix as $\mathbf{P}$. Then the corresponding diagonal matrix—$\mathbf{\Lambda} = \text{diag}(\lambda_i, \cdots, \lambda_n)$, say—contains the eigenvalues of $\mathbf{A}$. The spectral decomposition of $\mathbf{A}$ can be written as

$$\mathbf{A} = \mathbf{P}\mathbf{\Lambda}\mathbf{P}' = \sum_{i=1}^{n} \lambda_i \mathbf{p}_i \mathbf{p}_i'$$

where $\mathbf{p}_i$ denotes the ith column vector of $\mathbf{P}$. The right-side expression decomposes $\mathbf{A}$ into a sum of rank-1 matrices, and the weight of each contribution is equal to the eigenvalue associated with the ith eigenvector. The sum furthermore emphasizes that the rank of $\mathbf{A}$ is equal to the number of nonzero eigenvalues.

Harville (1997, p. 538) refers to the spectral decomposition of $\mathbf{A}$ as the decomposition that takes the previous sum one step further and accumulates contributions associated with the distinct eigenvalues. If $\lambda_i^*, \cdots, \lambda_k^*$ are the distinct eigenvalues and $\mathbf{E}_j = \sum \mathbf{p}_i \mathbf{p}_i'$, where the sum is taken over the set of columns for which $\lambda_i = \lambda_j^*$, then

$$\mathbf{A} = \sum_{i=1}^{k} \lambda_j^* \mathbf{E}_j$$

You can employ the spectral decomposition of a nonnegative definite symmetric matrix to form a "square-root" matrix of $\mathbf{A}$. Suppose that $\mathbf{\Lambda}^{1/2}$ is the diagonal matrix containing the square roots of the λ_i. Then $\mathbf{B} = \mathbf{P}\mathbf{\Lambda}^{1/2}\mathbf{P}'$ is a square-root matrix of $\mathbf{A}$ in the sense that $\mathbf{BB} = \mathbf{A}$, because

$$\mathbf{BB} = \mathbf{P}\mathbf{\Lambda}^{1/2}\mathbf{P}'\mathbf{P}\mathbf{\Lambda}^{1/2}\mathbf{P}' = \mathbf{P}\mathbf{\Lambda}^{1/2}\mathbf{\Lambda}^{1/2}\mathbf{P}' = \mathbf{P}\mathbf{\Lambda}\mathbf{P}'$$

Generating the Moore-Penrose inverse of a matrix based on the spectral decomposition is also simple. Denote as $\mathbf{\Delta}$ the diagonal matrix with typical element

$$\delta_i = \begin{cases} 1/\lambda_i & \lambda_i \neq 0 \\ 0 & \lambda_i = 0 \end{cases}$$

Then the matrix $\mathbf{P}\mathbf{\Delta}\mathbf{P}' = \sum \delta_i \mathbf{p}_i \mathbf{p}_i'$ is the Moore-Penrose (g_4-generalized) inverse of $\mathbf{A}$.

Singular-Value Decomposition

The singular-value decomposition is related to the spectral decomposition of a matrix, but it is more general. The singular-value decomposition can be applied to any matrix. Let $\mathbf{B}$ be an $(n \times p)$ matrix of rank k. Then there exist orthogonal matrices P and Q of order $(n \times n)$ and $(p \times p)$, respectively, and a diagonal matrix $\mathbf{D}$ such that

$$\mathbf{P}'\mathbf{B}\mathbf{Q} = \mathbf{D} = \begin{bmatrix} \mathbf{D}_1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

where $\mathbf{D}_1$ is a diagonal matrix of order k. The diagonal elements of $\mathbf{D}_1$ are strictly positive. As with the spectral decomposition, this result can be written as a decomposition of $\mathbf{B}$ into a weighted sum of rank-1 matrices

$$\mathbf{B} = -\mathbf{P}\mathbf{D}\mathbf{Q}' = \sum_{i=1}^{n} d_i \mathbf{p}_i \mathbf{q}_i'$$

The scalars $d_1, \cdots, d_k$ are called the *singular values* of the matrix $\mathbf{B}$. They are the positive square roots of the nonzero eigenvalues of the matrix $\mathbf{B}'\mathbf{B}$. If the singular-value decomposition is applied to a symmetric, nonnegative definite matrix $\mathbf{A}$, then the singular values $d_1, \cdots, d_n$ are the nonzero eigenvalues of $\mathbf{A}$ and the singular-value decomposition is the same as the spectral decomposition.

As with the spectral decomposition, you can use the results of the singular-value decomposition to generate the Moore-Penrose inverse of a matrix. If $\mathbf{B}$ is $(n \times p)$ with singular-value decomposition $\mathbf{PDQ}'$, and if $\mathbf{\Delta}$ is a diagonal matrix with typical element

$$\delta_i = \begin{cases} 1/d_i & |d_i| \neq 0 \\ 0 & d_i = 0 \end{cases}$$

then $\mathbf{Q\Delta P}'$ is the g_4-generalized inverse of $\mathbf{B}$.

Expectations of Random Variables and Vectors

If Y is a discrete random variable with mass function $p(y)$ and support (possible values) $y_1, y_2, \cdots$, then the expectation (expected value) of Y is defined as

$$\mathrm{E}[Y] = \sum_{j=1}^{\infty} y_j \; p(y_j)$$

provided that $\sum |y_j| p(y_j) < \infty$, otherwise the sum in the definition is not well-defined. The expected value of a function $h(y)$ is similarly defined: provided that $\sum |h(y_j)| p(y_j) < \infty$,

$$\mathrm{E}[h(Y)] = \sum_{j=1}^{\infty} h(y_j) \; p(y_j)$$

For continuous random variables, similar definitions apply, but summation is replaced by integration over the support of the random variable. If X is a continuous random variable with density function $f(x)$, and $\int |x| f(x) dx < \infty$, then the expectation of X is defined as

$$\mathrm{E}[X] = \int_{-\infty}^{\infty} x f(x) \, dx$$

The expected value of a random variable is also called its *mean* or its first moment. A particularly important function of a random variable is $h(Y) = (Y - \mathrm{E}[Y])^2$. The expectation of $h(Y)$ is called the *variance* of Y or the second central moment of Y. When you study the properties of multiple random variables, then you might be interested in aspects of their joint distribution. The covariance between random variables Y and X is defined as the expected value of the function $(Y - \mathrm{E}[Y])(X - \mathrm{E}[X])$, where the expectation is taken under the bivariate joint distribution of Y and X:

$$\mathrm{Cov}[Y, X] = \mathrm{E}[(Y - \mathrm{E}[Y])(X - \mathrm{E}[X])] = \mathrm{E}[YX] - \mathrm{E}[Y]\mathrm{E}[X] = \int \int x \, y f(x, y) \; dxdy - \mathrm{E}[Y]\mathrm{E}[X]$$

The *covariance* between a random variable and itself is the variance, $\mathrm{Cov}[Y, Y] = \mathrm{Var}[Y]$.

In statistical applications and formulas, random variables are often collected into vectors. For example, a random sample of size n from the distribution of Y generates a random vector of order $(n \times 1)$,

$$\mathbf{Y} = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_n \end{bmatrix}$$

The expected value of the $(n \times 1)$ random vector $\mathbf{Y}$ is the vector of the means of the elements of $\mathbf{Y}$:

$$\mathbf{E}[\mathbf{Y}] = [\mathrm{E}[Y_i]] = \begin{bmatrix} \mathrm{E}[Y_1] \\ \mathrm{E}[Y_2] \\ \vdots \\ \mathrm{E}[Y_n] \end{bmatrix}$$

It is often useful to directly apply rules about working with means, variances, and covariances of random vectors. To develop these rules, suppose that $\mathbf{Y}$ and $\mathbf{U}$ denote two random vectors with typical elements $Y_1, \cdots, Y_n$ and $U_1, \cdots, U_k$. Further suppose that $\mathbf{A}$ and $\mathbf{B}$ are constant (nonstochastic) matrices, that $\mathbf{a}$ is a constant vector, and that the c_i are scalar constants.

The following rules enable you to derive the mean of a linear function of a random vector:

$$\begin{aligned} \mathrm{E}[\mathbf{A}] &= \mathbf{A} \\ \mathrm{E}[\mathbf{Y} + \mathbf{a}] &= \mathrm{E}[\mathbf{Y}] \\ \mathrm{E}[\mathbf{AY} + \mathbf{a}] &= \mathbf{A}\mathrm{E}[\mathbf{Y}] + \mathbf{a} \\ \mathrm{E}[\mathbf{Y} + \mathbf{U}] &= \mathrm{E}[\mathbf{Y}] + \mathrm{E}[\mathbf{U}] \end{aligned}$$

The *covariance matrix* of $\mathbf{Y}$ and $\mathbf{U}$ is the $(n \times k)$ matrix whose typical element in row i, column j is the covariance between Y_i and U_j. The covariance matrix between two random vectors is frequently denoted with the Cov "operator."

$$\begin{aligned} \mathrm{Cov}[\mathbf{Y}, \mathbf{U}] &= \left[\mathrm{Cov}[Y_i, U_j]\right] \\ &= \mathrm{E}\left[(\mathbf{Y} - \mathrm{E}[\mathbf{Y}])\,(\mathbf{U} - \mathrm{E}[\mathbf{U}])'\right] = \mathrm{E}\left[\mathbf{YU}'\right] - \mathrm{E}[\mathbf{Y}]\mathrm{E}[\mathbf{U}]' \\ &= \begin{bmatrix} \mathrm{Cov}[Y_1, U_1] & \mathrm{Cov}[Y_1, U_2] & \mathrm{Cov}[Y_1, U_3] & \cdots & \mathrm{Cov}[Y_1, U_k] \\ \mathrm{Cov}[Y_2, U_1] & \mathrm{Cov}[Y_2, U_2] & \mathrm{Cov}[Y_2, U_3] & \cdots & \mathrm{Cov}[Y_2, U_k] \\ \mathrm{Cov}[Y_3, U_1] & \mathrm{Cov}[Y_3, U_2] & \mathrm{Cov}[Y_3, U_3] & \cdots & \mathrm{Cov}[Y_3, U_k] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[Y_n, U_1] & \mathrm{Cov}[Y_n, U_2] & \mathrm{Cov}[Y_n, U_3] & \cdots & \mathrm{Cov}[Y_n, U_k] \end{bmatrix} \end{aligned}$$

The *variance matrix* of a random vector $\mathbf{Y}$ is the covariance matrix between $\mathbf{Y}$ and itself. The variance matrix is frequently denoted with the Var "operator."

$$\begin{aligned}\mathrm{Var}[\mathbf{Y}] &= \mathrm{Cov}[\mathbf{Y},\mathbf{Y}] = \left[\mathrm{Cov}[Y_i, Y_j]\right] \\ &= \mathrm{E}\left[(\mathbf{Y}-\mathrm{E}[\mathbf{Y}])\,(\mathbf{Y}-\mathrm{E}[\mathbf{Y}])'\right] = \mathrm{E}\left[\mathbf{Y}\mathbf{Y}'\right] - \mathrm{E}[\mathbf{Y}]\mathrm{E}[\mathbf{Y}]'\end{aligned}$$

$$= \begin{bmatrix} \mathrm{Cov}[Y_1,Y_1] & \mathrm{Cov}[Y_1,Y_2] & \mathrm{Cov}[Y_1,Y_3] & \cdots & \mathrm{Cov}[Y_1,Y_n] \\ \mathrm{Cov}[Y_2,Y_1] & \mathrm{Cov}[Y_2,Y_2] & \mathrm{Cov}[Y_2,Y_3] & \cdots & \mathrm{Cov}[Y_2,Y_n] \\ \mathrm{Cov}[Y_3,Y_1] & \mathrm{Cov}[Y_3,Y_2] & \mathrm{Cov}[Y_3,Y_3] & \cdots & \mathrm{Cov}[Y_3,Y_n] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[Y_n,Y_1] & \mathrm{Cov}[Y_n,Y_2] & \mathrm{Cov}[Y_n,Y_3] & \cdots & \mathrm{Cov}[Y_n,Y_n] \end{bmatrix}$$

$$= \begin{bmatrix} \mathrm{Var}[Y_1] & \mathrm{Cov}[Y_1,Y_2] & \mathrm{Cov}[Y_1,Y_3] & \cdots & \mathrm{Cov}[Y_1,Y_n] \\ \mathrm{Cov}[Y_2,Y_1] & \mathrm{Var}[Y_2] & \mathrm{Cov}[Y_2,Y_3] & \cdots & \mathrm{Cov}[Y_2,Y_n] \\ \mathrm{Cov}[Y_3,Y_1] & \mathrm{Cov}[Y_3,Y_2] & \mathrm{Var}[Y_3] & \cdots & \mathrm{Cov}[Y_3,Y_n] \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}[Y_n,Y_1] & \mathrm{Cov}[Y_n,Y_2] & \mathrm{Cov}[Y_n,Y_3] & \cdots & \mathrm{Var}[Y_n] \end{bmatrix}$$

Because the variance matrix contains variances on the diagonal and covariances in the off-diagonal positions, it is also referred to as the *variance-covariance matrix* of the random vector $\mathbf{Y}$.

If the elements of the covariance matrix $\mathrm{Cov}[\mathbf{Y},\mathbf{U}]$ are zero, the random vectors are uncorrelated. If $\mathbf{Y}$ and $\mathbf{U}$ are normally distributed, then a zero covariance matrix implies that the vectors are stochastically independent. If the off-diagonal elements of the variance matrix $\mathrm{Var}[\mathbf{Y}]$ are zero, the elements of the random vector $\mathbf{Y}$ are uncorrelated. If $\mathbf{Y}$ is normally distributed, then a diagonal variance matrix implies that its elements are stochastically independent.

Suppose that $\mathbf{A}$ and $\mathbf{B}$ are constant (nonstochastic) matrices and that c_i denotes a scalar constant. The following results are useful in manipulating covariance matrices:

$$\begin{aligned}\mathrm{Cov}[\mathbf{A}\mathbf{Y},\mathbf{U}] &= \mathbf{A}\mathrm{Cov}[\mathbf{Y},\mathbf{U}] \\ \mathrm{Cov}[\mathbf{Y},\mathbf{B}\mathbf{U}] &= \mathrm{Cov}[\mathbf{Y},\mathbf{U}]\mathbf{B}' \\ \mathrm{Cov}[\mathbf{A}\mathbf{Y},\mathbf{B}\mathbf{U}] &= \mathbf{A}\mathrm{Cov}[\mathbf{Y},\mathbf{U}]\mathbf{B}' \\ \mathrm{Cov}[c_1\mathbf{Y}_1 + c_2\mathbf{U}_1, c_3\mathbf{Y}_2 + c_4\mathbf{U}_2] &= c_1c_3\mathrm{Cov}[\mathbf{Y}_1,\mathbf{Y}_2] + c_1c_4\mathrm{Cov}[\mathbf{Y}_1,\mathbf{U}_2] \\ &\quad + c_2c_3\mathrm{Cov}[\mathbf{U}_1,\mathbf{Y}_2] + c_2c_4\mathrm{Cov}[\mathbf{U}_1,\mathbf{U}_2]\end{aligned}$$

Since $\mathrm{Cov}[\mathbf{Y},\mathbf{Y}] = \mathrm{Var}[\mathbf{Y}]$, these results can be applied to produce the following results, useful in manipulating variances of random vectors:

$$\begin{aligned}\mathrm{Var}[\mathbf{A}] &= \mathbf{0} \\ \mathrm{Var}[\mathbf{A}\mathbf{Y}] &= \mathbf{A}\mathrm{Var}[\mathbf{Y}]\mathbf{A}' \\ \mathrm{Var}[\mathbf{Y}+\mathbf{x}] &= \mathrm{Var}[\mathbf{Y}] \\ \mathrm{Var}[\mathbf{x}'\mathbf{Y}] &= \mathbf{x}'\mathrm{Var}[\mathbf{Y}]\mathbf{x} \\ \mathrm{Var}[c_1\mathbf{Y}] &= c_1^2\mathrm{Var}[\mathbf{Y}] \\ \mathrm{Var}[c_1\mathbf{Y} + c_2\mathbf{U}] &= c_1^2\mathrm{Var}[\mathbf{Y}] + c_2^2\mathrm{Var}[\mathbf{U}] + 2c_1c_2\mathrm{Cov}[\mathbf{Y},\mathbf{U}]\end{aligned}$$

Another area where expectation rules are helpful is quadratic forms in random variables. These forms arise particularly in the study of linear statistical models and in linear statistical inference. Linear inference is statistical inference about linear function of random variables, even if those random variables are defined through nonlinear models. For example, the parameter estimator $\widehat{\boldsymbol{\theta}}$ might be derived in a nonlinear model, but this does not prevent statistical questions from being raised that can be expressed through linear functions of $\boldsymbol{\theta}$; for example,

$$H_0: \begin{cases} \theta_1 - 2\theta_2 = 0 \\ \theta_2 - \theta_3 = 0 \end{cases}$$

if $\mathbf{A}$ is a matrix of constants and $\mathbf{Y}$ is a random vector, then

$$\mathrm{E}[\mathbf{Y}'\mathbf{A}\mathbf{Y}] = \mathrm{trace}(\mathbf{A}\mathrm{Var}[\mathbf{Y}]) + \mathrm{E}[\mathbf{Y}]'\mathbf{A}\mathrm{E}[\mathbf{Y}]$$

Mean Squared Error

The mean squared error is arguably the most important criterion used to evaluate the performance of a predictor or an estimator. (The subtle distinction between predictors and estimators is that random variables are predicted and constants are estimated.) The mean squared error is also useful to relay the concepts of bias, precision, and accuracy in statistical estimation. In order to examine a mean squared error, you need a target of estimation or prediction, and a predictor or estimator that is a function of the data. Suppose that the target, whether a constant or a random variable, is denoted as U. The mean squared error of the estimator or predictor $T(\mathbf{Y})$ for U is

$$\mathrm{MSE}\left[T(\mathbf{Y}); U\right] = \mathrm{E}\left[(T(\mathbf{Y}) - U)^2\right]$$

The reason for using a squared difference to measure the "loss" between $T(\mathbf{Y})$ and U is mostly convenience; properties of squared differences involving random variables are more easily examined than, say, absolute differences. The reason for taking an expectation is to remove the randomness of the squared difference by averaging over the distribution of the data.

Consider first the case where the target U is a constant—say, the parameter β—and denote the mean of the estimator $T(\mathbf{Y})$ as μ_T. The mean squared error can then be decomposed as

$$\begin{aligned} \mathrm{MSE}[T(\mathbf{Y}; \beta] &= \mathrm{E}\left[(T(\mathbf{Y}) - \beta)^2\right] \\ &= \mathrm{E}\left[(T(\mathbf{Y}) - \mu_T)^2\right] - \mathrm{E}\left[(\beta - \mu_T)^2\right] \\ &= \mathrm{Var}[T(\mathbf{Y})] + (\beta - \mu_T)^2 \end{aligned}$$

The mean squared error thus comprises the variance of the estimator and the squared bias. The two components can be associated with an estimator's precision (small variance) and its accuracy (small bias).

If $T(\mathbf{Y})$ is an unbiased estimator of β—that is, if $\mathrm{E}[T(\mathbf{Y})] = \beta$—then the mean squared error is simply the variance of the estimator. By choosing an estimator that has minimum variance, you also choose an estimator that has minimum mean squared error among all unbiased estimators. However, as you can see from the previous expression, bias is also an "average" property; it is defined as an

expectation. It is quite possible to find estimators in some statistical modeling problems that have smaller mean squared error than a minimum variance unbiased estimator; these are estimators that permit a certain amount of bias but improve on the variance. For example, in models where regressors are highly collinear, the ordinary least squares estimator continues to be unbiased. However, the presence of collinearity can induce poor precision and lead to an erratic estimator. Ridge regression stabilizes the regression estimates in this situation, and the coefficient estimates are somewhat biased, but the bias is more than offset by the gains in precision.

When the target U is a random variable, you need to carefully define what an unbiased prediction means. If the statistic and the target have the same expectation, $\mathrm{E}[U] = \mathrm{E}[T(\mathbf{Y})]$, then

$$\mathrm{MSE}\left[T(\mathbf{Y}); U\right] = \mathrm{Var}[T(\mathbf{Y})] + \mathrm{Var}[U] - 2\mathrm{Cov}[T(\mathbf{Y}), U]$$

In many instances the target U is a new observation that was not part of the analysis. If the data are uncorrelated, then it is reasonable to assume in that instance that the new observation is also not correlated with the data. The mean squared error then reduces to the sum of the two variances. For example, in a linear regression model where U is a new observation Y_0 and $T(\mathbf{Y})$ is the regression estimator

$$\widehat{Y}_0 = \mathbf{x}_0' \left(\mathbf{X}'\mathbf{X}\right)^{-1} \mathbf{X}'\mathbf{Y}$$

with variance $\mathrm{Var}[Y_0] = \sigma^2 \mathbf{x}_0' \left(\mathbf{X}'\mathbf{X}\right)^{-1} \mathbf{x}_0$, the mean squared prediction error for Y_0 is

$$\mathrm{MSE}\left[\widehat{Y}; Y_0\right] = \sigma^2 \left(\mathbf{x}_0' \left(\mathbf{X}'\mathbf{X}\right)^{-1} \mathbf{x}_0 + 1\right)$$

and the mean squared prediction error for predicting the mean $\mathrm{E}[Y_0]$ is

$$\mathrm{MSE}\left[\widehat{Y}; \mathrm{E}[Y_0]\right] = \sigma^2 \mathbf{x}_0' \left(\mathbf{X}'\mathbf{X}\right)^{-1} \mathbf{x}_0$$

Linear Model Theory

This section presents some basic statistical concepts and results for the linear model with homoscedastic, uncorrelated errors in which the parameters are estimated by ordinary least squares. The model can be written as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \qquad \boldsymbol{\epsilon} \sim (0, \sigma^2 \mathbf{I})$$

where $\mathbf{Y}$ is an $(n \times 1)$ vector and $\mathbf{X}$ is an $(n \times k)$ matrix of known constants. The model equation implies the following expected values:

$$\mathrm{E}[\mathbf{Y}] = \mathbf{X}\boldsymbol{\beta}$$

$$\mathrm{Var}[\mathbf{Y}] = \sigma^2 \mathbf{I} \Leftrightarrow \mathrm{Cov}[Y_i, Y_j] = \begin{cases} \sigma^2 & i = j \\ 0 & \text{otherwise} \end{cases}$$

Finding the Least Squares Estimators

Finding the least squares estimator of $\boldsymbol{\beta}$ can be motivated as a calculus problem or by considering the geometry of least squares. The former approach simply states that the OLS estimator is the vector $\widehat{\boldsymbol{\beta}}$ that minimizes the objective function

$$\text{SSE} = (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})' (\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})$$

Applying the differentiation rules from the section "Matrix Differentiation" on page 53 leads to

$$\begin{aligned}\frac{\partial}{\partial \boldsymbol{\beta}}\text{SSE} &= \frac{\partial}{\partial \boldsymbol{\beta}}(\mathbf{Y}'\mathbf{Y} - 2\mathbf{Y}'\mathbf{X}\boldsymbol{\beta} + \boldsymbol{\beta}'\mathbf{X}'\mathbf{X}\boldsymbol{\beta}) \\ &= \mathbf{0} - 2\mathbf{X}'\mathbf{Y} + 2\mathbf{X}'\mathbf{X}\boldsymbol{\beta} \\ \frac{\partial^2}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}}\text{SSE} &= \mathbf{X}'\mathbf{X}\end{aligned}$$

Consequently, the solution to the *normal equations*, $\mathbf{X}'\mathbf{X}\boldsymbol{\beta} = \mathbf{X}'\mathbf{Y}$, solves $\frac{\partial}{\partial \boldsymbol{\beta}}\text{SSE} = 0$, and the fact that the second derivative is nonnegative definite guarantees that this solution minimizes SSE. The geometric argument to motivate ordinary least squares estimation is as follows. Assume that $\mathbf{X}$ is of rank k. For any value of $\boldsymbol{\beta}$, such as $\tilde{\boldsymbol{\beta}}$, the following identity holds:

$$\mathbf{Y} = \mathbf{X}\tilde{\boldsymbol{\beta}} + \left(\mathbf{Y} - \mathbf{X}\tilde{\boldsymbol{\beta}}\right)$$

The vector $\mathbf{X}\tilde{\boldsymbol{\beta}}$ is a point in a k-dimensional subspace of R^n, and the residual $(\mathbf{Y} - \mathbf{X}\tilde{\boldsymbol{\beta}})$ is a point in an $(n-k)$-dimensional subspace. The OLS estimator is the value $\widehat{\boldsymbol{\beta}}$ that minimizes the distance of $\mathbf{X}\tilde{\boldsymbol{\beta}}$ from $\mathbf{Y}$, implying that $\mathbf{X}\widehat{\boldsymbol{\beta}}$ and $(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}})$ are orthogonal to each other; that is,

$(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}})'\mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{0}$. This in turn implies that $\widehat{\boldsymbol{\beta}}$ satisfies the normal equations, since

$$\widehat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{Y} = \widehat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{X}\widehat{\boldsymbol{\beta}} \Leftrightarrow \mathbf{X}'\mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{X}'\mathbf{Y}$$

Full-Rank Case

If $\mathbf{X}$ is of full column rank, the OLS estimator is unique and given by

$$\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y}$$

The OLS estimator is an unbiased estimator of $\boldsymbol{\beta}$—that is,

$$\begin{aligned}\text{E}[\widehat{\boldsymbol{\beta}}] &= \text{E}\left[\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y}\right] \\ &= \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\text{E}[\mathbf{Y}] = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{X}\boldsymbol{\beta} = \boldsymbol{\beta}\end{aligned}$$

Note that this result holds if $\text{E}[\mathbf{Y}] = \mathbf{X}\boldsymbol{\beta}$; in other words, the condition that the model errors have mean zero is sufficient for the OLS estimator to be unbiased. If the errors are homoscedastic and uncorrelated, the OLS estimator is indeed the *best* linear unbiased estimator (BLUE) of $\boldsymbol{\beta}$—that is, no other estimator that is a linear function of $\mathbf{Y}$ has a smaller mean squared error. The fact that the estimator is unbiased implies that no other linear estimator has a smaller variance. If, furthermore, the model errors are normally distributed, then the OLS estimator has minimum variance among all unbiased estimators of $\boldsymbol{\beta}$, whether they are linear or not. Such an estimator is called a *uniformly minimum variance unbiased estimator*, or UMVUE.

Rank-Deficient Case

In the case of a rank-deficient $\mathbf{X}$ matrix, a generalized inverse is used to solve the normal equations:

$$\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{X}\right)^{-}\mathbf{X}'\mathbf{Y}$$

Although a g_1-inverse is sufficient to solve a linear system, computational expedience and interpretation of the results often dictate the use of a generalized inverse with reflexive properties (that is, a g_2-inverse; see the section "Generalized Inverse Matrices" on page 52 for details). Suppose, for example, that the $\mathbf{X}$ matrix is partitioned as $\mathbf{X} = [\mathbf{X}_1 \ \mathbf{X}_2]$, where $\mathbf{X}_1$ is of full column rank and each column in $\mathbf{X}_2$ is a linear combination of the columns of $\mathbf{X}_1$. The matrix

$$\mathbf{G}_1 = \begin{bmatrix} \left(\mathbf{X}_1'\mathbf{X}_1\right)^{-1} & \left(\mathbf{X}_1'\mathbf{X}_1\right)^{-1}\mathbf{X}_1'\mathbf{X}_2 \\ -\mathbf{X}_2'\mathbf{X}_1\left(\mathbf{X}_1'\mathbf{X}_1\right)^{-1} & \mathbf{0} \end{bmatrix}$$

is a g_1-inverse of $\mathbf{X}'\mathbf{X}$ and

$$\mathbf{G}_2 = \begin{bmatrix} \left(\mathbf{X}_1'\mathbf{X}_1\right)^{-1} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}$$

is a g_2-inverse. If the least squares solution is computed with the g_1-inverse, then computing the variance of the estimator requires additional matrix operations and storage. On the other hand, the variance of the solution that uses a g_2-inverse is proportional to G_2.

$$\begin{aligned} \text{Var}\left[\mathbf{G}_1\mathbf{X}'\mathbf{Y}\right] &= \sigma^2\mathbf{G}_1\mathbf{X}'\mathbf{X}\mathbf{G}_1 \\ \text{Var}\left[\mathbf{G}_2\mathbf{X}'\mathbf{Y}\right] &= \sigma^2\mathbf{G}_2\mathbf{X}'\mathbf{X}\mathbf{G}_2 = \sigma^2\mathbf{G}_2 \end{aligned}$$

If a generalized inverse $\mathbf{G}$ of $\mathbf{X}'\mathbf{X}$ is used to solve the normal equations, then the resulting solution is a biased estimator of $\boldsymbol{\beta}$ (unless $\mathbf{X}'\mathbf{X}$ is of full rank, in which case the generalized inverse is "the" inverse), since $\text{E}\left[\widehat{\boldsymbol{\beta}}\right] = \mathbf{G}\mathbf{X}'\mathbf{X}\boldsymbol{\beta}$, which is not in general equal to $\boldsymbol{\beta}$.

If you think of estimation as "estimation without bias," then $\widehat{\boldsymbol{\beta}}$ is the estimator of something, namely $\mathbf{G}\mathbf{X}\boldsymbol{\beta}$. Since this is not a quantity of interest and since it is not unique—it depends on your choice of $\mathbf{G}$—Searle (1971, p. 169) cautions that in the less-than-full-rank case, $\widehat{\boldsymbol{\beta}}$ is a solution to the normal equations and "nothing more."

Analysis of Variance

The identity

$$\mathbf{Y} = \mathbf{X}\tilde{\boldsymbol{\beta}} + \left(\mathbf{Y} - \mathbf{X}\tilde{\boldsymbol{\beta}}\right)$$

holds for all vectors $\tilde{\boldsymbol{\beta}}$, but only for the least squares solution is the residual $(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}})$ orthogonal to the predicted value $\mathbf{X}\widehat{\boldsymbol{\beta}}$. Because of this orthogonality, the additive identity holds not only for the vectors themselves, but also for their lengths (Pythagorean theorem):

$$||\mathbf{Y}||^2 = ||\mathbf{X}\widehat{\boldsymbol{\beta}}||^2 + ||(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}})||^2$$

Note that $\mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y} = \mathbf{H}\mathbf{Y}$ and note that $\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}} = (\mathbf{I} - \mathbf{H})\mathbf{Y} = \mathbf{M}\mathbf{Y}$. The matrices $\mathbf{H}$ and $\mathbf{M} = \mathbf{I} - \mathbf{H}$ play an important role in the theory of linear models and in statistical computations. Both are *projection* matrices—that is, they are symmetric and idempotent. (An idempotent matrix $\mathbf{A}$ is a square matrix that satisfies $\mathbf{A}\mathbf{A} = \mathbf{A}$. The eigenvalues of an idempotent matrix take on the values 1 and 0 only.) The matrix $\mathbf{H}$ projects onto the subspace of R^n that is spanned by the columns of $\mathbf{X}$. The matrix $\mathbf{M}$ projects onto the orthogonal complement of that space. Because of these properties you have $\mathbf{H}' = \mathbf{H}$, $\mathbf{H}\mathbf{H} = \mathbf{H}$, $\mathbf{M}' = \mathbf{M}$, $\mathbf{M}\mathbf{M} = \mathbf{M}$, $\mathbf{H}\mathbf{M} = \mathbf{0}$.

The Pythagorean relationship now can be written in terms of $\mathbf{H}$ and $\mathbf{M}$ as follows:

$$||\mathbf{Y}||^2 = \mathbf{Y}'\mathbf{Y} = ||\mathbf{H}\mathbf{Y}||^2 + ||\mathbf{M}\mathbf{Y}||^2 = \mathbf{Y}'\mathbf{H}'\mathbf{H}\mathbf{Y} + \mathbf{Y}'\mathbf{M}'\mathbf{M}\mathbf{Y} = \mathbf{Y}'\mathbf{H}\mathbf{Y} + \mathbf{Y}'\mathbf{M}\mathbf{Y}$$

If $\mathbf{X}'\mathbf{X}$ is deficient in rank and a generalized inverse is used to solve the normal equations, then you work instead with the projection matrices $\mathbf{H} = \mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-}\mathbf{X}'$. Note that if $\mathbf{G}$ is a generalized inverse of $\mathbf{X}'\mathbf{X}$, then $\mathbf{X}\mathbf{G}\mathbf{X}'$, and hence also $\mathbf{H}$ and $\mathbf{M}$, are invariant to the choice of $\mathbf{G}$.

The matrix $\mathbf{H}$ is sometimes referred to as the "hat" matrix because when you premultiply the vector of observations with $\mathbf{H}$, you produce the fitted values, which are commonly denoted by placing a "hat" over the $\mathbf{Y}$ vector, $\widehat{\mathbf{Y}} = \mathbf{H}\mathbf{Y}$.

The term $\mathbf{Y}'\mathbf{Y}$ is the uncorrected total sum of squares (SST) of the linear model, $\mathbf{Y}'\mathbf{M}\mathbf{Y}$ is the error (residual) sum of squares (SSR), and $\mathbf{Y}'\mathbf{H}\mathbf{Y}$ is the uncorrected model sum of squares. This leads to the analysis of variance table shown in Table 3.2.

Table 3.2 Analysis of Variance with Uncorrected Sums of Squares

Source	df	Sum of Squares
Model	rank($\mathbf{X}$)	$\text{SSM} = \mathbf{Y}'\mathbf{H}\mathbf{Y} = \widehat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{Y}$
Residual	$n - \text{rank}(\mathbf{X})$	$\text{SSR} = \mathbf{Y}'\mathbf{M}\mathbf{Y} = \mathbf{Y}'\mathbf{Y} - \widehat{\boldsymbol{\beta}}\mathbf{X}'\mathbf{Y} = \sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2$
Uncorr. Total	n	$\text{SST} = \mathbf{Y}'\mathbf{Y} = \sum_{i=1}^{n} Y_i^2$

When the model contains an intercept term, then the analysis of variance is usually corrected for the mean, as shown in Table 3.3.

Table 3.3 Analysis of Variance with Corrected Sums of Squares

Source	df	Sum of Squares
Model	rank($\mathbf{X}$) $- 1$	$\text{SSM}_c = \widehat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{Y} - n\overline{Y}^2 = \sum_{i=1}^{n}\left(\widehat{Y}_i - \overline{Y}_i\right)^2$
Residual	$n - \text{rank}(\mathbf{X})$	$\text{SSR} = \mathbf{Y}'\mathbf{M}\mathbf{Y} = \mathbf{Y}'\mathbf{Y} - \widehat{\boldsymbol{\beta}}\mathbf{X}'\mathbf{Y} = \sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2$
Corrected Total	$n - 1$	$\text{SST}_c = \mathbf{Y}'\mathbf{Y} - n\overline{Y}^2 = \sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2$

The *coefficient of determination*, also called the R-square statistic, measures the proportion of the total variation explained by the linear model. In models with intercept, it is defined as the ratio

$$R^2 = 1 - \frac{\text{SSR}}{\text{SST}_c} = 1 - \frac{\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2}{\sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2}$$

In models without intercept, the R-square statistic is a ratio of the uncorrected sums of squares

$$R^2 = 1 - \frac{\text{SSR}}{\text{SST}} = 1 - \frac{\sum_{i=1}^{n}\left(Y_i - \widehat{Y}_i\right)^2}{\sum_{i=1}^{n} Y_i^2}$$

Estimating the Error Variance

The least squares principle does not provide for a parameter estimator for σ^2. The usual approach is to use a method-of-moments estimator that is based on the sum of squared residuals. If the model is correct, then the mean square for error, defined to be SSR divided by its degrees of freedom,

$$\begin{aligned}\widehat{\sigma}^2 &= \frac{1}{n - \text{rank}(\mathbf{X})}\left(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\right)'\left(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\right)\\ &= \text{SSR}/(n - \text{rank}(\mathbf{X}))\end{aligned}$$

is an unbiased estimator of σ^2.

Maximum Likelihood Estimation

To estimate the parameters in a linear model with mean function $\text{E}[\mathbf{Y}] = \mathbf{X}\boldsymbol{\beta}$ by maximum likelihood, you need to specify the distribution of the response vector $\mathbf{Y}$. In the linear model with a continuous response variable, it is commonly assumed that the response is normally distributed. In that case, the estimation problem is completely defined by specifying the mean and variance of $\mathbf{Y}$ in addition to the normality assumption. The model can be written as $\mathbf{Y} \sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I})$, where the notation $N(\mathbf{a}, \mathbf{V})$ indicates a multivariate normal distribution with mean vector $\mathbf{a}$ and variance matrix $\mathbf{V}$. The log likelihood for $\mathbf{Y}$ then can be written as

$$l(\boldsymbol{\beta}, \sigma^2; \mathbf{y}) = -\frac{n}{2}\log\{2\pi\} - \frac{n}{2}\log\{\sigma^2\} - \frac{1}{2\sigma^2}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

This function is maximized in $\boldsymbol{\beta}$ when the sum of squares $(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$ is minimized. The maximum likelihood estimator of $\boldsymbol{\beta}$ is thus identical to the ordinary least squares estimator. To maximize $l(\boldsymbol{\beta}, \sigma^2; \mathbf{y})$ with respect to σ^2, note that

$$\frac{\partial l(\boldsymbol{\beta}, \sigma^2; \mathbf{y})}{\partial \sigma^2} = -\frac{n}{2\sigma^2} + \frac{1}{2\sigma^4}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

Hence the MLE of σ^2 is the estimator

$$\begin{aligned}\widehat{\sigma}^2_M &= \frac{1}{n}\left(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\right)'\left(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\right)\\ &= \text{SSR}/n\end{aligned}$$

This is a biased estimator of σ^2, with a bias that decreases with n.

Estimable Functions

A function $\mathbf{L}\boldsymbol{\beta}$ is said to be estimable if there exists a linear combination of the expected value of $\mathbf{Y}$, such as $\mathbf{K}\mathrm{E}[\mathbf{Y}]$, that equals $\mathbf{L}\boldsymbol{\beta}$. Since $\mathrm{E}[\mathbf{Y}] = \mathbf{X}\boldsymbol{\beta}$, the definition of estimability implies that $\mathbf{L}\boldsymbol{\beta}$ is estimable if there is a matrix $\mathbf{K}$ such that $\mathbf{L} = \mathbf{K}\mathbf{X}$. Another way of looking at this result is that the rows of $\mathbf{X}$ form a generating set from which all estimable functions can be constructed.

The concept of estimability of functions is important in the theory and application of linear models because hypotheses of interest are often expressed as linear combinations of the parameter estimates (for example, hypotheses of equality between parameters, $\beta_1 = \beta_2 \Leftrightarrow \beta_1 - \beta_2 = 0$). Since estimability is not related to the particular value of the parameter estimate, but to the row space of $\mathbf{X}$, you can test only hypotheses that consist of estimable functions. Further, because estimability is not related to the value of $\boldsymbol{\beta}$ (Searle 1971, p. 181), the choice of the generalized inverse in a situation with rank-deficient $\mathbf{X}'\mathbf{X}$ matrix is immaterial, since

$$\mathbf{L}\widehat{\boldsymbol{\beta}} = \mathbf{K}\mathbf{X}\widehat{\boldsymbol{\beta}} = \mathbf{K}\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-}\mathbf{X}'\mathbf{Y}$$

where $\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-}\mathbf{X}$ is invariant to the choice of generalized inverse.

$\mathbf{L}\boldsymbol{\beta}$ is estimable if and only if $\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}(\mathbf{X}'\mathbf{X}) = \mathbf{L}$ (see, for example, Searle 1971, p. 185). If $\mathbf{X}$ is of full rank, then the *Hermite* matrix $(\mathbf{X}'\mathbf{X})^{-}(\mathbf{X}'\mathbf{X})$ is the identity, which implies that all linear functions are estimable in the full-rank case.

See Chapter 15, "The Four Types of Estimable Functions," for many details about the various forms of estimable functions in SAS/STAT.

Test of Hypotheses

Consider a general linear hypothesis of the form $H\colon \mathbf{L}\boldsymbol{\beta} = \mathbf{d}$, where $\mathbf{L}$ is a $(k \times p)$ matrix. It is assumed that $\mathbf{d}$ is such that this hypothesis is linearly consistent—that is, that there exists *some* $\boldsymbol{\beta}$ for which $\mathbf{L}\boldsymbol{\beta} = \mathbf{d}$. This is always the case if $\mathbf{d}$ is in the column space of $\mathbf{L}$, if $\mathbf{L}$ has full row rank, or if $\mathbf{d} = \mathbf{0}$; the latter is the most common case. Since many linear models have a rank-deficient $\mathbf{X}$ matrix, the question arises whether the hypothesis is testable. The idea of testability of a hypothesis is—not surprisingly—connected to the concept of estimability as introduced previously. The hypothesis $H\colon \mathbf{L}\boldsymbol{\beta} = \mathbf{d}$ is testable if it consists of estimable functions.

There are two important approaches to testing hypotheses in statistical applications—the reduction principle and the linear inference approach. The reduction principle states that the validity of the hypothesis can be inferred by comparing a suitably chosen summary statistic between the model at hand and a reduced model in which the constraint $\mathbf{L}\boldsymbol{\beta} = \mathbf{d}$ is imposed. The linear inference approach relies on the fact that $\widehat{\boldsymbol{\beta}}$ is an estimator of $\boldsymbol{\beta}$ and its stochastic properties are known, at least approximately. A test statistic can then be formed using $\widehat{\boldsymbol{\beta}}$, and its behavior under the restriction $\mathbf{L}\boldsymbol{\beta} = \mathbf{d}$ can be ascertained.

The two principles lead to identical results in certain—for example, least squares estimation in the classical linear model. In more complex situations the two approaches lead to similar but not identical results. This is the case, for example, when weights or unequal variances are involved, or when $\widehat{\boldsymbol{\beta}}$ is a nonlinear estimator.

Reduction Tests

The two main reduction principles are the sum of squares reduction test and the likelihood ratio test. The test statistic in the former is proportional to the difference of the residual sum of squares between the reduced model and the full model. The test statistic in the likelihood ratio test is proportional to the difference of the log likelihoods between the full and reduced models. To fix these ideas, suppose that you are fitting the model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$. Suppose that SSR denotes the residual sum of squares in this model and that SSR_H is the residual sum of squares in the model for which $\mathbf{L}\boldsymbol{\beta} = \mathbf{d}$ holds. Then under the hypothesis the ratio

$$(\text{SSR}_H - \text{SSR})/\sigma^2$$

follows a chi-square distribution with degrees of freedom equal to the rank of $\mathbf{L}$. Maybe surprisingly, the residual sum of squares in the full model is distributed independently of this quantity, so that under the hypothesis,

$$F = \frac{(\text{SSR}_H - \text{SSR})/\text{rank}(\mathbf{L})}{\text{SSR}/(n - \text{rank}(\mathbf{X}))}$$

follows an F distribution with $\text{rank}(\mathbf{L})$ numerator and $n - \text{rank}(\mathbf{X})$ denominator degrees of freedom. Note that the quantity in the denominator of the F statistic is a particular estimator of σ^2—namely, the unbiased moment-based estimator that is customarily associated with least squares estimation. It is also the restricted maximum likelihood estimator of σ^2 if $\mathbf{Y}$ is normally distributed.

In the case of the likelihood ratio test, suppose that $l(\widehat{\boldsymbol{\beta}}, \widehat{\sigma}^2; \mathbf{y})$ denotes the log likelihood evaluated at the ML estimators. Also suppose that $l(\widehat{\boldsymbol{\beta}}_H, \widehat{\sigma}^2_H; \mathbf{y})$ denotes the log likelihood in the model for which $\mathbf{L}\boldsymbol{\beta} = \mathbf{d}$ holds. Then under the hypothesis the statistic

$$\lambda = 2\left(l(\widehat{\boldsymbol{\beta}}, \widehat{\sigma}^2; \mathbf{y}) - l(\widehat{\boldsymbol{\beta}}_H, \widehat{\sigma}^2_H; \mathbf{y})\right)$$

follows approximately a chi-square distribution with degrees of freedom equal to the rank of $\mathbf{L}$. In the case of a normally distributed response, the log-likelihood function can be profiled with respect to $\boldsymbol{\beta}$. The resulting profile log likelihood is

$$l(\widehat{\sigma}^2; \mathbf{y}) = -\frac{n}{2}\log\{2\pi\} - \frac{n}{2}\left(\log\{\widehat{\sigma}^2\}\right)$$

and the likelihood ratio test statistic becomes

$$\lambda = n\left(\log\left\{\widehat{\sigma}^2_H\right\} - \log\left\{\widehat{\sigma}^2\right\}\right) = n\left(\log\{\text{SSR}_H\} - \log\{\text{SSR}\}\right) = n\left(\log\{\text{SSR}_H/\text{SSR}\}\right)$$

The preceding expressions show that, in the case of normally distributed data, both reduction principles lead to simple functions of the residual sums of squares in two models. As Pawitan (2001, p. 151) puts it, there is, however, an important difference not in the computations but in the statistical content. The least squares principle, where sum of squares reduction tests are widely used, does not require a distributional specification. Assumptions about the distribution of the data are added to provide a framework for confirmatory inferences, such as the testing of hypotheses. This framework stems directly from the assumption about the data's distribution, or from the sampling distribution of the least squares estimators. The likelihood principle, on the other hand, requires a distributional specification at the outset. Inference about the parameters is implicit in the model; it is the result of further *computations* following the estimation of the parameters. In the least squares framework, inference about the parameters is the result of further *assumptions*.

Linear Inference

The principle of linear inference is to formulate a test statistic for $H\colon \mathbf{L}\boldsymbol{\beta} = \mathbf{d}$ that builds on the linearity of the hypothesis about $\boldsymbol{\beta}$. For many models that have linear components, the estimator $\mathbf{L}\widehat{\boldsymbol{\beta}}$ is also linear in $\mathbf{Y}$. It is then simple to establish the distributional properties of $\mathbf{L}\widehat{\boldsymbol{\beta}}$ based on the distributional assumptions about $\mathbf{Y}$ or based on large-sample arguments. For example, $\widehat{\boldsymbol{\beta}}$ might be a nonlinear estimator, but it is known to asymptotically follow a normal distribution; this is the case in many nonlinear and generalized linear models.

If the sampling distribution or the asymptotic distribution of $\widehat{\boldsymbol{\beta}}$ is normal, then one can easily derive quadratic forms with known distributional properties. For example, if the random vector $\mathbf{U}$ is distributed as $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, then $\mathbf{U}'\mathbf{A}\mathbf{U}$ follows a chi-square distribution with rank($\mathbf{A}$) degrees of freedom and noncentrality parameter $1/2\boldsymbol{\mu}'\mathbf{A}\boldsymbol{\mu}$, provided that $\mathbf{A}\boldsymbol{\Sigma}\mathbf{A}\boldsymbol{\Sigma} = \mathbf{A}\boldsymbol{\Sigma}$.

In the classical linear model, suppose that $\mathbf{X}$ is deficient in rank and that $\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{X}\right)^{-}\mathbf{X}'\mathbf{Y}$ is a solution to the normal equations. Then, if the errors are normally distributed,

$$\widehat{\boldsymbol{\beta}} \sim N\left(\left(\mathbf{X}'\mathbf{X}\right)^{-}\mathbf{X}'\mathbf{X}\boldsymbol{\beta}, \sigma^2\left(\mathbf{X}'\mathbf{X}\right)^{-}\mathbf{X}'\mathbf{X}\left(\mathbf{X}'\mathbf{X}\right)^{-}\right)$$

Because $H\colon \mathbf{L}\boldsymbol{\beta} = \mathbf{d}$ is testable, $\mathbf{L}\boldsymbol{\beta}$ is estimable, and thus $\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X} = \mathbf{L}$, as established in the previous section. Hence,

$$\mathbf{L}\widehat{\boldsymbol{\beta}} \sim N\left(\mathbf{L}\boldsymbol{\beta}, \sigma^2\mathbf{L}\left(\mathbf{X}'\mathbf{X}\right)^{-}\mathbf{L}'\right)$$

The conditions for a chi-square distribution of the quadratic form

$$(\mathbf{L}\widehat{\boldsymbol{\beta}} - \mathbf{d})'\left(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}'\right)^{-}(\mathbf{L}\widehat{\boldsymbol{\beta}} - \mathbf{d})$$

are thus met, provided that

$$(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-}\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-}\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}' = (\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-}\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}'$$

This condition is obviously met if $\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}'$ is of full rank. The condition is also met if $\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}'^{-}$ is a reflexive inverse (a g_2-inverse) of $\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}$.

The test statistic to test the linear hypothesis $H\colon \mathbf{L}\boldsymbol{\beta} = \mathbf{d}$ is thus

$$F = \frac{(\mathbf{L}\widehat{\boldsymbol{\beta}} - \mathbf{d})'\left(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}'\right)^{-}(\mathbf{L}\widehat{\boldsymbol{\beta}} - \mathbf{d})/\text{rank}(\mathbf{L})}{SSR/(n - \text{rank}(\mathbf{X}))}$$

and it follows an F distribution with rank($\mathbf{L}$) numerator and $n - \text{rank}(\mathbf{X})$ denominator degrees of freedom under the hypothesis.

This test statistic looks very similar to the F statistic for the sum of squares reduction test. This is no accident. If the model is linear and parameters are estimated by ordinary least squares, then you can show that the quadratic form $(\mathbf{L}\widehat{\boldsymbol{\beta}} - \mathbf{d})'\left(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}'\right)^{-}(\mathbf{L}\widehat{\boldsymbol{\beta}} - \mathbf{d})$ equals the differences in the residual sum of squares, $\text{SSR}_H - \text{SSR}$, where SSR_H is obtained as the residual sum of squares from OLS estimation in a model that satisfies $\mathbf{L}\boldsymbol{\beta} = \mathbf{d}$. However, this correspondence between the two test formulations does not apply when a different estimation principle is used. For example, assume that $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \mathbf{V})$ and that $\boldsymbol{\beta}$ is estimated by generalized least squares:

$$\widehat{\boldsymbol{\beta}}_g = \left(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}\right)^{-}\mathbf{X}'\mathbf{V}^{-1}\mathbf{Y}$$

The construction of $\mathbf{L}$ matrices associated with hypotheses in SAS/STAT software is frequently based on the properties of the $\mathbf{X}$ matrix, not of $\mathbf{X}'\mathbf{V}^{-}\mathbf{X}$. In other words, the construction of the $\mathbf{L}$ matrix is governed only by the design. A sum of squares reduction test for $H: \mathbf{L}\boldsymbol{\beta} = \mathbf{0}$ that uses the generalized residual sum of squares $(\mathbf{Y} - \widehat{\boldsymbol{\beta}}_g)'\mathbf{V}^{-1}(\mathbf{Y} - \widehat{\boldsymbol{\beta}}_g)$ is not identical to a linear hypothesis test with the statistic

$$F^* = \frac{\widehat{\boldsymbol{\beta}}_g'\mathbf{L}'\left(\mathbf{L}\left(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}\right)^{-}\mathbf{L}'\right)^{-}\mathbf{L}\widehat{\boldsymbol{\beta}}_g}{\text{rank}(\mathbf{L})}$$

Furthermore, $\mathbf{V}$ is usually unknown and must be estimated as well. The estimate for $\mathbf{V}$ depends on the model, and imposing a constraint on the model would change the estimate. The asymptotic distribution of the statistic F^* is a chi-square distribution. However, in practical applications the F distribution with rank($\mathbf{L}$) numerator and ν denominator degrees of freedom is often used because it provides a better approximation to the sampling distribution of F^* in finite samples. The computation of the denominator degrees of freedom ν, however, is a matter of considerable discussion. A number of methods have been proposed and are implemented in various forms in SAS/STAT (see, for example, the degrees-of-freedom methods in the MIXED and GLIMMIX procedures).

Residual Analysis

The model errors $\boldsymbol{\epsilon} = \mathbf{Y} - \mathbf{X}\boldsymbol{\beta}$ are unobservable. Yet important features of the statistical model are connected to them, such as the distribution of the data, the correlation among observations, and the constancy of variance. It is customary to diagnose and investigate features of the model errors through the fitted residuals $\widehat{\boldsymbol{\epsilon}} = \mathbf{Y} - \widehat{\mathbf{Y}} = \mathbf{Y} - \mathbf{H}\mathbf{Y} = \mathbf{M}\mathbf{Y}$. These residuals are projections of the data onto the null space of $\mathbf{X}$ and are also referred to as the "raw" residuals to contrast them with other forms of residuals that are transformations of $\widehat{\boldsymbol{\epsilon}}$. For the classical linear model, the statistical properties of $\widehat{\boldsymbol{\epsilon}}$ are affected by the features of that projection and can be summarized as follows:

$$\begin{aligned} \text{E}[\widehat{\boldsymbol{\epsilon}}] &= \mathbf{0} \\ \text{Var}[\widehat{\boldsymbol{\epsilon}}] &= \sigma^2\mathbf{M} \\ \text{rank}(\mathbf{M}) &= n - \text{rank}(\mathbf{X}) \end{aligned}$$

Furthermore, if $\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$, then $\widehat{\boldsymbol{\epsilon}} \sim N(\mathbf{0}, \sigma^2\mathbf{M})$.

Because $\mathbf{M} = \mathbf{I} - \mathbf{H}$, and the "hat" matrix $\mathbf{H}$ satisfies $\partial\widehat{\mathbf{Y}}/\partial\mathbf{Y}$, the hat matrix is also the leverage matrix of the model. If h_{ii} denotes the ith diagonal element of $\mathbf{H}$ (the leverage of observation i), then the leverages are bounded in a model with intercept, $1/n \leq h_{ii} \leq 1$. Consequently, the variance of a raw residual is less than that of an observation: $\text{Var}[\widehat{\epsilon}_i] = \sigma^2(1 - h_{ii}) < \sigma^2$. In applications where the variability of the data is estimated from fitted residuals, the estimate is invariably biased low. An example is the computation of an empirical semivariogram based on fitted (detrended) residuals.

More important, the diagonal entries of $\mathbf{H}$ are not necessarily identical; the residuals are heteroscedastic. The "hat" matrix is also not a diagonal matrix; the residuals are correlated. In summary, the only property that the fitted residuals $\widehat{\boldsymbol{\epsilon}}$ share with the model errors is a zero mean. It is thus commonplace to use transformations of the fitted residuals for diagnostic purposes.

Raw and Studentized Residuals

A *standardized* residual is a raw residual that is divided by its standard deviation:

$$\widehat{\epsilon}_i^* = \frac{Y_i - \widehat{Y}_i}{\sqrt{\mathrm{Var}[Y_i - \widehat{Y}_i]}} = \frac{\widehat{\epsilon}_i}{\sqrt{\sigma^2(1-h_{ii})}}$$

Because σ^2 is unknown, residual standardization is usually not practical. A *studentized* residual is a raw residual that is divided by its estimated standard deviation. If the estimate of the standard deviation is based on the same data that were used in fitting the model, the residual is also called an *internally studentized* residual:

$$\widehat{\epsilon}_{is} = \frac{Y_i - \widehat{Y}_i}{\sqrt{\widehat{\mathrm{Var}}[Y_i - \widehat{Y}_i]}} = \frac{\widehat{\epsilon}_i}{\sqrt{\widehat{\sigma}^2(1-h_{ii})}}$$

If the estimate of the residual's variance does not involve the ith observation, it is called an *externally studentized* residual. Suppose that $\widehat{\sigma}^2_{-i}$ denotes the estimate of the residual variance obtained without the ith observation; then the externally studentized residual is

$$\widehat{\epsilon}_{ir} = \frac{\widehat{\epsilon}_i}{\sqrt{\widehat{\sigma}^2_{-i}(1-h_{ii})}}$$

Scaled Residuals

A scaled residual is simply a raw residual divided by a scalar quantity that is not an estimate of the variance of the residual. For example, residuals divided by the standard deviation of the response variable are scaled and referred to as Pearson or Pearson-type residuals:

$$\widehat{\epsilon}_{ic} = \frac{Y_i - \widehat{Y}_i}{\sqrt{\widehat{\mathrm{Var}}[Y_i]}}$$

In generalized linear models, where the variance of an observation is a function of the mean μ and possibly of an extra scale parameter, $\mathrm{Var}[Y] = a(\mu)\phi$, the Pearson residual is

$$\widehat{\epsilon}_{iP} = \frac{Y_i - \widehat{\mu}_i}{\sqrt{a(\widehat{\mu})}}$$

because the sum of the squared Pearson residuals equals the Pearson X^2 statistic:

$$X^2 = \sum_{i=1}^{n} \widehat{\epsilon}_{iP}^2$$

When the scale parameter ϕ participates in the scaling, the residual is also referred to as a Pearson-type residual:

$$\widehat{\epsilon}_{iP} = \frac{Y_i - \widehat{\mu}_i}{\sqrt{a(\widehat{\mu})\phi}}$$

Other Residuals

You might encounter other residuals in SAS/STAT software. A "leave-one-out" residual is the difference between the observed value and the residual obtained from fitting a model in which the observation in question did not participate. If $\widehat{Y}_i$ is the predicted value of the ith observation and $\widehat{Y}_{i,-i}$ is the predicted value if Y_i is removed from the analysis, then the "leave-one-out" residual is

$$\widehat{\epsilon}_{i,-i} = Y_i - \widehat{Y}_{i,-i}$$

Since the sum of the squared "leave-one-out" residuals is the PRESS statistic (prediction sum of squares; Allen 1974), $\widehat{\epsilon}_{i,-i}$ is also called the PRESS residual. The concept of the PRESS residual can be generalized if the deletion residual can be based on the removal of sets of observations. In the classical linear model, the PRESS residual for case deletion has a particularly simple form:

$$\widehat{\epsilon}_{i,-i} = Y_i - \widehat{Y}_{i,-i} = \frac{\widehat{\epsilon}_i}{1 - h_{ii}}$$

That is, the PRESS residual is simply a scaled form of the raw residual, where the scaling factor is a function of the leverage of the observation.

When data are correlated, $\mathrm{Var}[\mathbf{Y}] = \mathbf{V}$, you can scale the vector of residuals rather than scale each residual separately. This takes the covariances among the observations into account. This form of scaling is accomplished by forming the Cholesky root $\mathbf{C}'\mathbf{C} = \mathbf{V}$, where $\mathbf{C}'$ is a lower-triangular matrix. Then $\mathbf{C}'^{-1}\mathbf{Y}$ is a vector of uncorrelated variables with unit variance. The Cholesky residuals in the model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ are

$$\widehat{\boldsymbol{\epsilon}}_C = \mathbf{C}'^{-1}\left(\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\right)$$

In generalized linear models, the fit of a model can be measured by the scaled deviance statistic D^*. It measures the difference between the log likelihood under the model and the maximum log likelihood that is achievable. In models with a scale parameter ϕ, the deviance is $D = \phi \times D^* = \sum_{i=1}^{n} d_i$. The deviance residuals are the signed square roots of the contributions to the deviance statistic:

$$\widehat{\epsilon}_{id} = \mathrm{sign}\{y_i - \widehat{\mu}_i\}\sqrt{d_i}$$

Sweep Operator

The sweep operator (Goodnight 1979) is closely related to Gauss-Jordan elimination and the Forward Doolittle procedure. The fact that a sweep operation can produce a generalized inverse by in-place mapping with minimal storage and that its application invariably leads to some form of matrix inversion is important, but this observation does not do justice to the pervasive relevance of sweeping to statistical computing. In this section the sweep operator is discussed as a conceptual tool for further insight into linear model operations. Consider the nonnegative definite, symmetric, partitioned matrix

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}'_{12} & \mathbf{A}_{22} \end{bmatrix}$$

Sweeping a matrix consists of performing a series of row operations akin to Gauss-Jordan elimination. Basic row operations are the multiplication of a row by a constant and the addition of a

multiple of one row to another. The sweep operator restricts row operations to pivots on the diagonal elements of a matrix; further details about the elementary operations can be found in Goodnight (1979). The process of sweeping the matrix $\mathbf{A}$ on its leading partition is denoted as Sweep$(\mathbf{A}, \mathbf{A}_{11})$ and leads to

$$\text{Sweep}(\mathbf{A}, \mathbf{A}_{11}) = \begin{bmatrix} \mathbf{A}_{11}^{-} & \mathbf{A}_{11}^{-}\mathbf{A}_{12} \\ -\mathbf{A}_{12}'\mathbf{A}_{11}^{-} & \mathbf{A}_{22} - \mathbf{A}_{12}'\mathbf{A}_{11}^{-}\mathbf{A}_{12} \end{bmatrix}$$

If the kth row and column are set to zero when the pivot is zero (or in practice, less than some singularity tolerance), the generalized inverse in the leading position of the swept matrix is a reflexive, g_2-inverse. Suppose that the crossproduct matrix of the linear model is augmented with a "Y-border" as follows:

$$\mathbf{C} = \begin{bmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{Y} \\ \mathbf{Y}'\mathbf{X} & \mathbf{Y}'\mathbf{Y} \end{bmatrix}$$

Then the result of sweeping on the rows of $\mathbf{X}$ is

$$\begin{aligned} \text{Sweep}(\mathbf{C}, \mathbf{X}) &= \begin{bmatrix} (\mathbf{X}'\mathbf{X})^{-} & (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{Y} \\ -\mathbf{Y}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-} & \mathbf{Y}'\mathbf{Y} - \mathbf{Y}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{Y} \end{bmatrix} \\ &= \begin{bmatrix} (\mathbf{X}'\mathbf{X})^{-} & \widehat{\boldsymbol{\beta}} \\ -\widehat{\boldsymbol{\beta}} & \mathbf{Y}'\mathbf{M}\mathbf{Y} \end{bmatrix} = \begin{bmatrix} (\mathbf{X}'\mathbf{X})^{-} & \widehat{\boldsymbol{\beta}} \\ -\widehat{\boldsymbol{\beta}} & \text{SSR} \end{bmatrix} \end{aligned}$$

The "Y-border" has been transformed into the least squares solution and the residual sum of squares.

Partial sweeps are common in model selection. Suppose that the $\mathbf{X}$ matrix is partitioned as $[\mathbf{X}_1 \ \mathbf{X}_2]$, and consider the augmented crossproduct matrix

$$\mathbf{C} = \begin{bmatrix} \mathbf{X}_1'\mathbf{X}_1 & \mathbf{X}_1'\mathbf{X}_2 & \mathbf{X}_1'\mathbf{Y} \\ \mathbf{X}_2'\mathbf{X}_1 & \mathbf{X}_2'\mathbf{X}_2 & \mathbf{X}_2'\mathbf{Y} \\ \mathbf{Y}'\mathbf{X}_1 & \mathbf{Y}'\mathbf{X}_2 & \mathbf{Y}'\mathbf{Y} \end{bmatrix}$$

Sweeping on the $\mathbf{X}_1$ partition yields

$$\text{Sweep}(\mathbf{C}, \mathbf{X}_1) = \begin{bmatrix} (\mathbf{X}_1'\mathbf{X}_1)^{-} & (\mathbf{X}_1'\mathbf{X}_1)^{-}\mathbf{X}_1'\mathbf{X}_2 & (\mathbf{X}_1'\mathbf{X}_1)^{-}\mathbf{X}_1'\mathbf{Y} \\ -\mathbf{X}_2'\mathbf{X}_1(\mathbf{X}_1'\mathbf{X}_1)^{-} & \mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2 & \mathbf{X}_2'\mathbf{M}_1\mathbf{Y} \\ \mathbf{Y}'\mathbf{X}_1(\mathbf{X}_1'\mathbf{X}_1)^{-} & \mathbf{Y}'\mathbf{M}_1\mathbf{X}_2 & \mathbf{Y}'\mathbf{M}_1\mathbf{Y} \end{bmatrix}$$

where $\mathbf{M}_1 = \mathbf{I} - \mathbf{X}_1(\mathbf{X}_1'\mathbf{X}_1)^{-}\mathbf{X}_1'$. The entries in the first row of this partition are the generalized inverse of $\mathbf{X}'\mathbf{X}$, the coefficients for regressing $\mathbf{X}_2$ on $\mathbf{X}_1$, and the coefficients for regressing $\mathbf{Y}$ on $\mathbf{X}_1$. The diagonal entries $\mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2$ and $\mathbf{Y}'\mathbf{M}_1\mathbf{Y}$ are the sum of squares and crossproduct matrices for regressing $\mathbf{X}_2$ on $\mathbf{X}_1$ and for regressing $\mathbf{Y}$ on $\mathbf{X}_1$, respectively. As you continue to sweep the matrix, the last cell in the partition contains the residual sum of square of a model in which $\mathbf{Y}$ is regressed on all columns swept up to that point.

The sweep operator is not only useful to conceptualize the computation of least squares solutions, Type I and Type II sums of squares, and generalized inverses. It can also be used to obtain other statistical information. For example, adding the logarithms of the pivots of the rows that are swept yields the log determinant of the matrix.

References

Allen, D. M. (1974), "The Relationship between Variable Selection and Data Augmentation and a Method of Prediction," *Technometrics*, 16, 125–127.

Cochran, W. G. (1997), *Sampling Techniques*, Third Edition, New York: John Wiley & Sons.

Goodnight, J. H. (1979), "A Tutorial on the Sweep Operator," *The American Statistician*, 33, 149–158.

Harville, D. A. (1997), *Matrix Algebra from a Statistician's Perspective*, New York: Springer-Verlag

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning,* New York: Springer-Verlag.

Jöreskog, K. G. (1973), "A General Method for Estimating a Linear Structural Equation System," in *Structural Equation Models in the Social Sciences*, ed. A. S. Goldberger and O. D. Duncan, New York: Seminar Press.

Keesling, J. W. (1972), "Maximum Likelihood Approaches to Causal Analysis," Ph. D. dissertation, University of Chicago, 1972.

Magnus, J. R., and Neudecker, H. (1999), *Matrix Differential Calculus with Applications in Statistics and Econometrics*, Second Edition, New York: John Wiley & Sons.

McCullagh, P. and Nelder. J. A. (1989), *Generalized Linear Models*, Second Edition, London: Chapman & Hall.

Moore, E. H. (1920), "On the Reciprocal of the General Algebraic Matrix," *Bulletin of the American Mathematical Society*, 26, 394–395.

Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized Linear Models," *Journal of the Royal Statistical Society A*, 135, 370–384.

Pawitan, Y. (2001), *In All Likelihood: Statistical Modelling and Inference Using Likelihood*, Oxford: Clarendon Press.

Penrose, R. A. (1955), "A Generalized Inverse for Matrices," *Proceedings of the Cambridge Philosophical Society*, 51, 406–413.

Pringle, R. M. and Rayner, A. A. (1971), *Generalized Inverse Matrices with Applications to Statistics*, New York: Hafner Publishing.

Särndal, C. E., Swensson, B., and Wretman, J. (1992), *Model Assisted Survey Sampling*, New York: Springer-Verlag.

Searle, S. R. (1971), *Linear Models*, New York: John Wiley & Sons.

Spearman, C. (1904), "General Intelligence, Objectively Determined and Measured." *American Journal of Psychology*, 15, 201–293.

Wedderburn, R. W. M. (1974), "Quasilikelihood Functions, Generalized Linear Models and the Gauss-Newton Method," *Biometrika*, 61, 439–447.

Wiley, D. E. (1973), "The Identification Problem for Structural Equation Models with Unmeasured Variables," in *Structural Equation Models in the Social Sciences*, ed. A. S. Goldberger and O. D. Duncan, New York: Seminar Press, 69–83.

Chapter 4

Introduction to Regression Procedures

Contents

Overview: Regression Procedures **76**
Introduction 76
Introductory Example: Linear Regression 79
Linear Regression: The REG Procedure 85
Response Surface Regression: The RSREG Procedure 88
Partial Least Squares Regression: The PLS Procedure 88
Generalized Linear Regression 88
Logistic Regression 89
Other Generalized Linear Models 90
Regression for Ill-Conditioned Data: The ORTHOREG Procedure 90
Quantile Regression: The QUANTREG Procedure 91
Nonlinear Regression 91
Nonparametric Regression 92
Local Regression: The LOESS Procedure 93
Smooth Function Approximation: The TPSPLINE Procedure 93
Generalized Additive Models: The GAM Procedure 93
Robust Regression: The ROBUSTREG Procedure 93
Regression with Transformations: The TRANSREG Procedure 94
Interactive Features in the CATMOD, GLM, and REG Procedures 94
Statistical Background in Linear Regression **95**
Linear Regression Models 95
Parameter Estimates and Associated Statistics 96
Predicted and Residual Values 100
Testing Linear Hypotheses 102
Multivariate Tests 102
Comments on Interpreting Regression Statistics 107
References **111**

Overview: Regression Procedures

This chapter provides an overview of procedures in SAS/STAT software that perform regression analysis. The REG procedure provides the most general analysis capabilities for the linear regression model. But many other procedures can fit linear regression models, and many procedures are specifically designed for more general regression problems, such as robust regression, generalized linear regression, nonlinear regression, nonparametric regression, regression modeling of survey data, regression modeling of survival data, and regression modeling of transformed variables. The aim of this chapter is to provide a brief road map and delineation of the various SAS/STAT procedures that can fit regression models.

A sampling of the procedures that fall into this category are the CATMOD, GAM, GENMOD, GLIMMIX, GLM, LIFEREG, LOESS, LOGISTIC, MIXED, NLIN, NLMIXED, ORTHOREG, PHREG, PLS, PROBIT, REG, ROBUSTREG, RSREG, SURVEYLOGISTIC, SURVEYREG, and TRANSREG procedures.

This chapter also briefly mentions several procedures in SAS/ETS software.

Introduction

Recall from Chapter 3, "Introduction to Statistical Modeling with SAS/STAT Software," that the general regression problem is to model the mean of a random vector $\mathbf{Y}$ as a function of a parameters and covariates in a statistical model. The many forms of regression models have their origin in the characteristics of the response variable (discrete or continuous, normal or nonnormal distributed), assumptions about the form of the model (linear, nonlinear, or generalized linear), assumptions about the data generating mechanism (survey, observational, or experimental data), and estimation principles. The following procedures, listed in alphabetical order, perform at least one type of regression analysis.

CATMOD analyzes data that can be represented by a contingency table. PROC CATMOD fits linear models to functions of response frequencies, and it can be used for linear and logistic regression. See Chapter 8, "Introduction to Categorical Data Analysis Procedures," and Chapter 28, "The CATMOD Procedure," for more information.

GAM fits generalized additive models. The models fitted with the GAM procedure are nonparameteric in that the usual assumption of a linear predictor is relaxed. The name stems from the fact that the models consist of additive, smooth functions in the regression variables. The GAM procedure can fit additive models to nonnormal data. See Chapter 36, "The GAM Procedure," for more information.

GENMOD fits generalized linear models. PROC GENMOD is especially suited for responses with discrete outcomes, and it performs logistic regression and Poisson regression as well as fitting generalized estimating equations for repeated measures data. Bayesian analysis capabilities for generalized linear models are also

available with the GENMOD procedure. See Chapter 8, "Introduction to Categorical Data Analysis Procedures," and Chapter 37, "The GENMOD Procedure," for more information.

GLIMMIX — fits generalized linear mixed models by likelihood-based methods. In addition to many other analyses, PROC GLIMMIX can perform simple, multiple, polynomial, and weighted regression. The GLIMMIX procedure can also fit linear mixed models and models without random effects. See Chapter 38, "The GLIMMIX Procedure," for more information.

GLM — uses the method of least squares to fit general linear models. In addition to many other analyses, PROC GLM can perform simple, multiple, polynomial, and weighted regression. PROC GLM has many of the same input/output capabilities as PROC REG, but it does not provide as many diagnostic tools or allow interactive changes in the model or data. See Chapter 5, "Introduction to Analysis of Variance Procedures," and Chapter 39, "The GLM Procedure," for more information.

LIFEREG — fits parametric models to failure-time data that might be right censored. These types of models are commonly used in survival analysis. See Chapter 14, "Introduction to Survey Procedures," and Chapter 48, "The LIFEREG Procedure," for more information.

LOESS — fits nonparametric models using a local regression method. PROC LOESS is suitable for modeling regression surfaces where the underlying parametric form is unknown and where robustness in the presence of outliers is required. See Chapter 50, "The LOESS Procedure," for more information.

LOGISTIC — fits logistic models for binomial and ordinal outcomes. PROC LOGISTIC provides a wide variety of model-building methods and computes numerous regression diagnostics. See Chapter 8, "Introduction to Categorical Data Analysis Procedures," and Chapter 51, "The LOGISTIC Procedure," for more information.

MIXED — fits linear mixed models by likelihood-based techniques. In addition to many other analyses, PROC MIXED can fit models without random effects; hence, the procedure can perform simple, multiple, polynomial, and weighted regression. See Chapter 56, "The MIXED Procedure," for more information.

NLIN — fits general nonlinear regression models by the method of nonlinear least squares. Several different iterative methods are available. See Chapter 60, "The NLIN Procedure," for more information.

NLMIXED — fits general nonlinear mixed regression models by the method of maximum likelihood. With the NLMIXED procedure you can specify a custom objective function for parameter estimation and fit models with or without random effects. See Chapter 61, "The NLMIXED Procedure," for more information.

ORTHOREG — performs regression by using the Gentleman-Givens computational method. For ill-conditioned data, PROC ORTHOREG can produce more accurate parameter estimates than other procedures such as PROC GLM and PROC REG. See Chapter 63, "The ORTHOREG Procedure," for more information.

PHREG — fits Cox proportional hazards regression models to survival data. See Chapter 64, "The PHREG Procedure," for more information.

PLS performs partial least squares regression, principal components regression, and reduced rank regression, with cross validation for the number of components. See Chapter 66, "The PLS Procedure," for more information.

PROBIT performs probit regression as well as logistic regression and ordinal logistic regression. The PROBIT procedure is useful when the dependent variable is either dichotomous or polychotomous and the independent variables are continuous. See Chapter 71, "The PROBIT Procedure," for more information.

QUANTREG models the effects of covariates on the conditional quantiles of a response variable by means of quantile regression. See Chapter 72, "The QUANTREG Procedure," for more information.

REG performs linear regression with many diagnostic capabilities, selects models by using one of nine methods, produces scatter plots of raw data and statistics, highlights scatter plots to identify particular observations, and allows interactive changes in both the regression model and the data used to fit the model. See Chapter 73, "The REG Procedure," for more information.

ROBUSTREG performs robust regression by using Huber M estimation and high breakdown value estimation. PROC ROBUSTREG is suitable for detecting outliers and providing resistant (stable) results in the presence of outliers. See Chapter 74, "The ROBUSTREG Procedure," for more information.

RSREG builds quadratic response-surface regression models. PROC RSREG analyzes the fitted response surface to determine the factor levels of optimum response and performs a ridge analysis to search for the region of optimum response. See Chapter 75, "The RSREG Procedure," for more information.

SURVEYLOGISTIC fits logistic models for binary and ordinal outcomes to survey data by maximum likelihood. See Chapter 84, "The SURVEYLOGISTIC Procedure," for more information.

SURVEYREG fits linear regression models to survey data by generalized least squares by using elementwise regresssion. See Chapter 86, "The SURVEYREG Procedure," for more information.

TRANSREG fits univariate and multivariate linear models, optionally with spline and other nonlinear transformations. Models include ordinary regression and ANOVA, multiple and multivariate regression, metric and nonmetric conjoint analysis, metric and nonmetric vector and ideal point preference mapping, redundancy analysis, canonical correlation, and response surface regression. See Chapter 90, "The TRANSREG Procedure," for more information.

Several SAS/ETS procedures also perform regression. The following procedures are documented in the *SAS/ETS User's Guide*.

AUTOREG implements regression models that use time-series data where the errors are autocorrelated. See Chapter 8, "The AUTOREG Procedure" (*SAS/ETS User's Guide*), for more details.

COUNTREG analyzes regression models in which the dependent variable takes nonnegative integer or count values. See Chapter 10, "The COUNTREG Procedure" (*SAS/ETS User's Guide*), for more details.

MODEL	handles nonlinear simultaneous systems of equations, such as econometric models. See Chapter 18, "The MODEL Procedure" (*SAS/ETS User's Guide*), for more details.
PANEL	analyzes a class of linear econometric models that commonly arise when time series and cross-sectional data are combined. See Chapter 19, "The PANEL Procedure" (*SAS/ETS User's Guide*), for more details.
PDLREG	performs regression analysis with polynomial distributed lags. See Chapter 20, "The PDLREG Procedure" (*SAS/ETS User's Guide*), for more details.
SYSLIN	handles linear simultaneous systems of equations, such as econometric models. See Chapter 26, "The SYSLIN Procedure" (*SAS/ETS User's Guide*), for more details.

Introductory Example: Linear Regression

Regression analysis is the analysis of the relationship between a response or outcome variable and another set of variables. The relationship is expressed through a statistical model equation that predicts a *response variable* (also called a *dependent variable* or *criterion*) from a function of *regressor variables* (also called *independent variables, predictors, explanatory variables, factors,* or *carriers*) and *parameters*. In a linear regression model the predictor function is linear in the parameters (but not necessarily linear in the regressor variables). The parameters are estimated so that a measure of fit is optimized. For example, the equation for the ith observation might be

$$Y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

where Y_i is the response variable, x_i is a regressor variable, β_0 and β_1 are unknown parameters to be estimated, and ϵ_i is an error term. This model is termed the simple linear regression (SLR) model, because it is linear in β_0 and β_1 and contains only a single regressor variable.

Suppose you are using regression analysis to relate a child's weight to a child's height. One application of a regression model with the response variable Weight is to predict a child's weight for a known height. Suppose you collect data by measuring heights and weights of 19 randomly selected schoolchildren. A simple linear regression model with the response variable weight and the regressor variable height can be written as the following equation:

$$\text{Wght}_i = \beta_0 + \beta_1 \text{Height}_i + \epsilon_i$$

where

Wght_i	is the response variable for the ith child.
Height_i	is the regressor variable for the ith child.
β_0, β_1	are the unknown regression parameters.
ϵ_i	is the unobservable random error associated with the ith observation.

The following DATA step creates a SAS data set identifying the children and their observed heights (variable Height) and weights (variable Wght).

```
data class;
   input Name $ Height Wght Age;
   datalines;
Alfred  69.0 112.5 14
Alice   56.5  84.0 13
Barbara 65.3  98.0 13
Carol   62.8 102.5 14
Henry   63.5 102.5 14
James   57.3  83.0 12
Jane    59.8  84.5 12
Janet   62.5 112.5 15
Jeffrey 62.5  84.0 13
John    59.0  99.5 12
Joyce   51.3  50.5 11
Judy    64.3  90.0 14
Louise  56.3  77.0 12
Mary    66.5 112.0 15
Philip  72.0 150.0 16
Robert  64.8 128.0 12
Ronald  67.0 133.0 15
Thomas  57.5  85.0 11
William 66.5 112.0 15
;

ods graphics on;
proc reg data=class;
   model Wght = Height;
run;
```

Figure 4.1 displays the default tabular output of the REG procedure for this model. Nineteen observations are read from the data set and all observations are used in the analysis. The estimates of the two regression parameters are $\widehat{\beta}_0 = -143.02692$ and $\widehat{\beta}_1 = 3.89903$. These estimates are obtained by the least squares principle. See the sections "Classical Estimation Principles" and "Linear Model Theory" in Chapter 3, "Introduction to Statistical Modeling with SAS/STAT Software," for details about the principle of least squares estimation and its role in linear model analysis. For a general discussion of the theory of least squares estimation of linear models and its application to regression and analysis of variance, refer to one of the applied regression texts, including Draper and Smith (1981), Daniel and Wood (1980), Johnston (1972), and Weisberg (1985).

Figure 4.1 Regression for Weight and Height Data

```
                    The REG Procedure
                      Model: MODEL1
                Dependent Variable: Wght

          Number of Observations Read          19
          Number of Observations Used          19
```

Figure 4.1 *continued*

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	7193.24912	7193.24912	57.08	<.0001
Error	17	2142.48772	126.02869		
Corrected Total	18	9335.73684			

Root MSE	11.22625	R-Square	0.7705
Dependent Mean	100.02632	Adj R-Sq	0.7570
Coeff Var	11.22330		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-143.02692	32.27459	-4.43	0.0004
Height	1	3.89903	0.51609	7.55	<.0001

Based on the least squares estimates shown in Figure 4.1, the fitted regression line relating height to weight is described by the equation

$$\widehat{\text{Wght}} = -143.02692 + 3.89903 \times \text{Height}$$

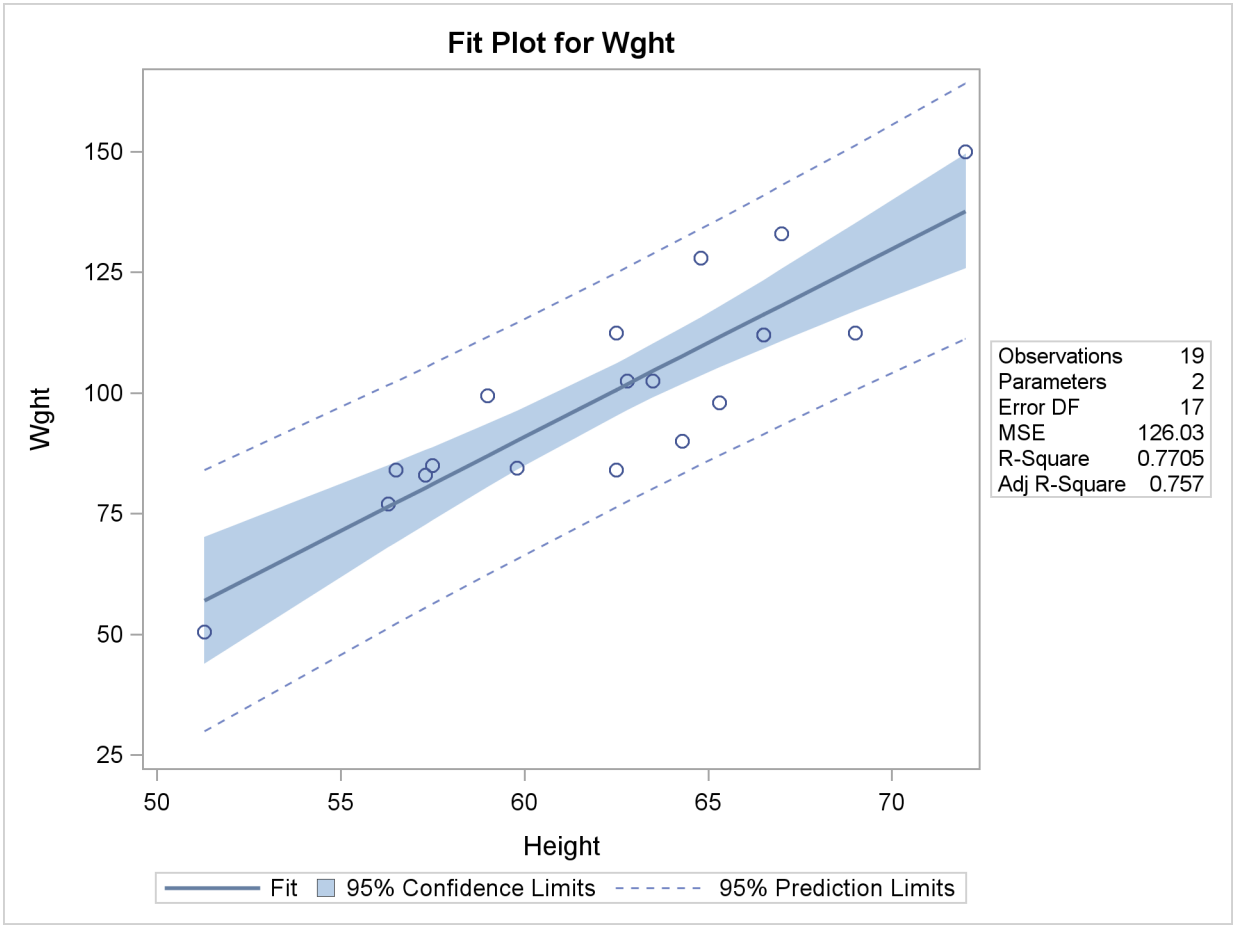
The "hat" notation is used to denote the predicted variable on the left side of the prediction equation to emphasize that $\widehat{\text{Wght}}$ is not one of the original observations but a value predicted under the regression model that has been fit to the data. At the least squares solution the residual sum of squares

$$\text{SSE} = \sum_{i=1}^{19} (\text{Wght}_i - \beta_0 - \beta_1 \text{Height}_i)^2$$

is minimized and the achieved criterion value is displayed in the analysis of variance table as the error sum of squares (2142.48772).

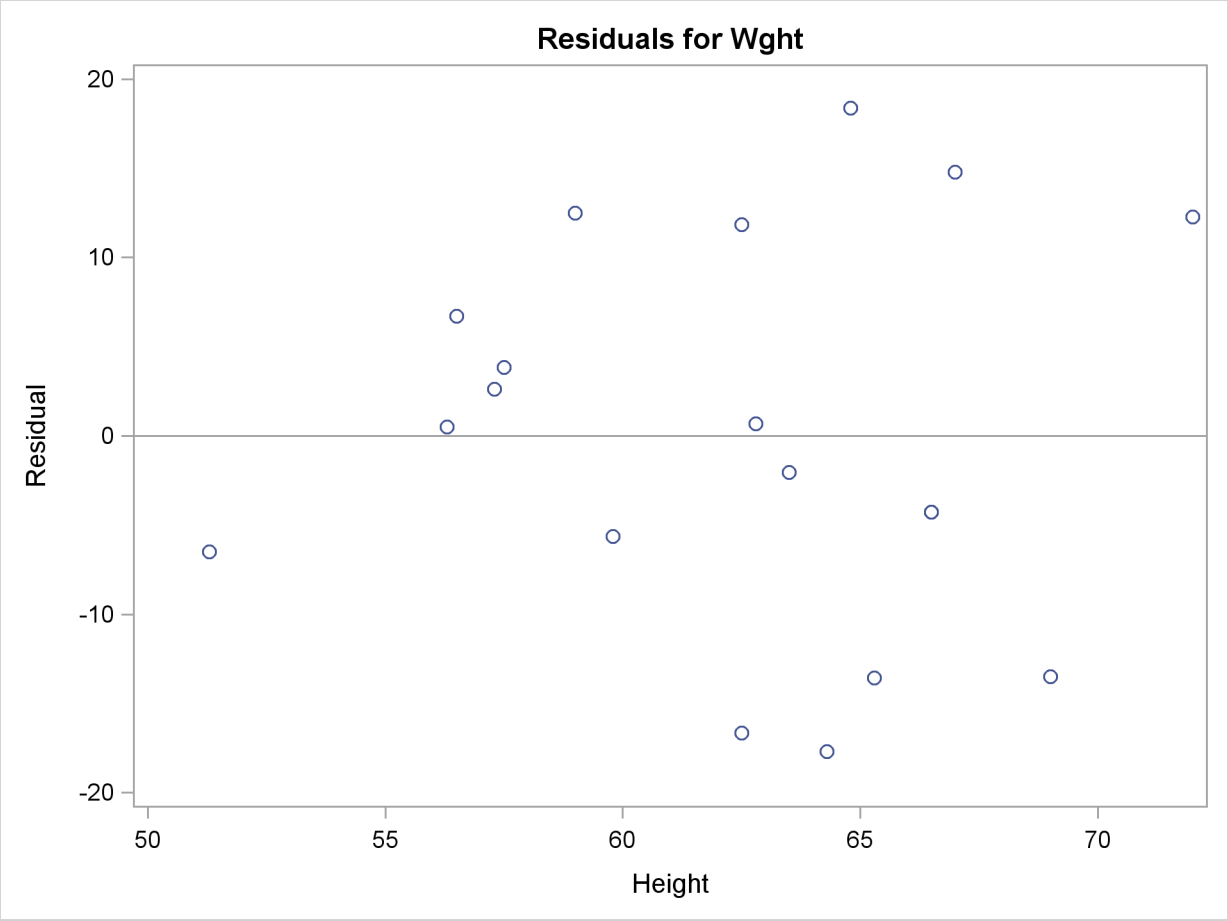
Figure 4.2 displays the fit plot induced by the `ODS GRAPHICS ON;` statement. The fit plot shows the positive slope of the fitted line. The average weight of a child changes by $\widehat{\beta}_1 = 3.89903$ units for each unit change in height. The 95% confidence limits in the fit plot are pointwise limits that cover the mean weight for a particular height with probability 0.95. The prediction limits, which are wider than the confidence limits, show the pointwise limits that cover a new observation for a given height with probability 0.95.

Figure 4.2 Fit Plot for Regression of Weight on Height



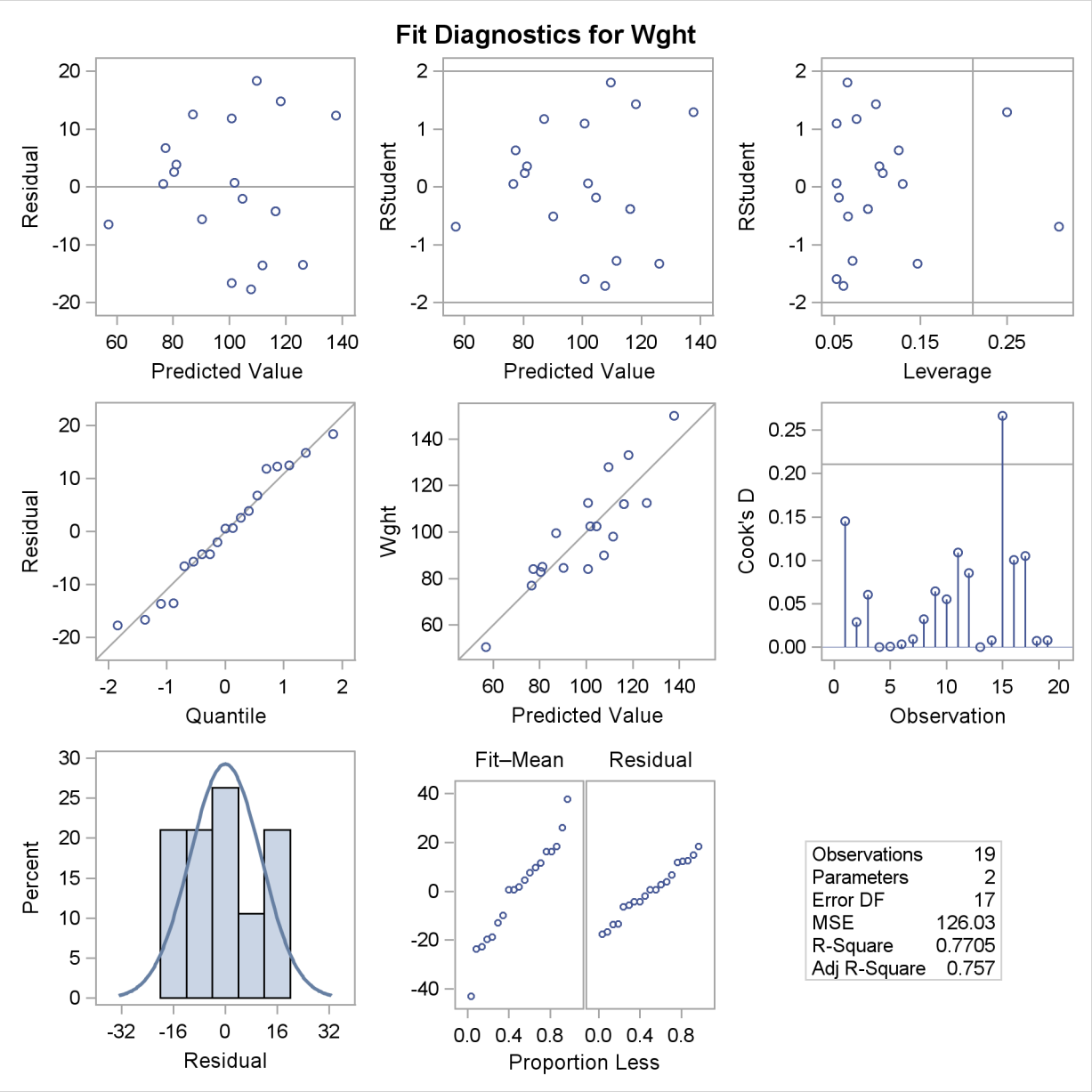
Regression is often used in an exploratory fashion to look for empirical relationships, such as the relationship between Height and Wght. In this example, Height is not the cause of Wght. You would need a controlled experiment to confirm the relationship scientifically. See the section "Comments on Interpreting Regression Statistics" on page 107 for more information. A separate question from a possible cause-and-effect relationship between the two variables involved in this regression is whether the simple linear regression model adequately describes the relationship in these data. If the usual assumptions about the model errors ϵ_i are met in the SLR model, then the errors should have zero mean and equal variance and be uncorrelated. Because the children were randomly selected, the observations from different children are not correlated. If the mean function of the model is correctly specified, the fitted residuals $\text{Wght}_i - \widehat{\text{Wght}}_i$ should scatter about the zero reference line without discernible structure. The residual plot in Figure 4.3 confirms this behavior.

Figure 4.3 Residual Plot for Regression of Weight on Height



An even more detailed look at the model-data agreement is gained with the panel of regression diagnostics in Figure 4.4. The graph in the upper-left panel repeats the raw residual plot in Figure 4.3. The plot of the RSTUDENT residuals shows externally studentized residuals that take into account heterogeneity in the variability of the residuals. RSTUDENT residuals that exceed the threshold values of ± 2 often indicate outlying observations. The residual-by-leverage plot shows that two observations have high leverage—that is, they are unusual in their height values relative to the other children. The normal-probability Q-Q plot in the second row of the panel shows that the normality assumption for the residuals is reasonable. The plot of the Cook's D statistic shows that observation 15 exceeds the threshold value; this indicates that the observation for this child is influential on the regression parameter estimates.

Figure 4.4 Panel of Regression Diagnostics



For detailed information about the interpretation of regression diagnostics and about ODS statistical graphics with the REG procedure, see Chapter 73, "The REG Procedure."

SAS/STAT regression procedures produce the following information for a typical regression analysis:

- parameter estimates derived by using the least squares criterion
- estimates of the variance of the error term
- estimates of the variance or standard deviation of the sampling distribution of the parameter estimates
- tests of hypotheses about the parameters

SAS/STAT regression procedures can produce many other specialized diagnostic statistics, including the following:

- collinearity diagnostics to measure how strongly regressors are related to other regressors and how this affects the stability and variance of the estimates (REG)
- influence diagnostics to measure how each individual observation contributes to determining the parameter estimates, the SSE, and the fitted values (LOGISTIC, MIXED, REG, RSREG)
- lack-of-fit diagnostics that measure the lack of fit of the regression model by comparing the error variance estimate to another pure error variance that is not dependent on the form of the model (CATMOD, PROBIT, RSREG)
- diagnostic scatter plots that check the fit of the model and highlighted scatter plots that identify particular observations or groups of observations (REG)
- predicted and residual values, and confidence intervals for the mean and for an individual value (GLM, LOGISTIC, REG)
- time-series diagnostics for equally spaced time-series data that measure how much errors might be related across neighboring observations. These diagnostics can also measure functional goodness of fit for data sorted by regressor or response variables (REG, SAS/ETS procedures).

Many SAS/STAT procedures produce general and specialized statistical graphics through ODS Graphics to diagnose the fit of the model and the model-data agreement, and to highlight observations that are influential on the analysis. Figure 4.2 and Figure 4.3, for example, are two of the ODS statistical graphs produced by the REG procedure by default for the simple linear regression model. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the ODS statistical graphs available with a SAS/STAT procedure, see the PLOTS option in the "Syntax" section for the PROC statement and the "ODS Table Names" section in the "Details" section of the individual procedure documentation.

Linear Regression: The REG Procedure

The REG procedure is a general-purpose procedure for linear regression that does the following:

- handles multiple regression models
- provides nine model-selection methods
- allows interactive changes both in the model and in the data used to fit the model
- allows linear equality restrictions on parameters
- tests linear hypotheses and multivariate hypotheses
- produces collinearity diagnostics, influence diagnostics, and partial regression leverage plots

- saves estimates, predicted values, residuals, confidence limits, and other diagnostic statistics in output SAS data sets
- generates plots of data and of various statistics
- "paints" or highlights scatter plots to identify particular observations or groups of observations
- uses, optionally, correlations or crossproducts for input

Model-Selection Methods in Linear Regression Models

An important step in building statistical models is to determine which effects and variables affect the response variable and to form a model that fits the data well without incurring the negative effects of overfitting the model. Models that are overfit—that is, contain too many regressor variables and unimportant regressor variables—have a tendency to be too closely molded to a a particular set of data, have unstable regression coefficients, and possibly have poor predictive precision. In situations where many potential regressor variables are available for inclusion in a regression model, guided, numerical variable-selection methods offer one approach to model building.

The model-selection techniques for linear regression models implemented in the REG procedure are as follows:

NONE — specifies that no selection be made. This method is the default and uses the full model given in the MODEL statement to fit the linear regression.

FORWARD — specifies that variables be selected based on a forward-selection algorithm. This method starts with no variables in the model and adds variables one by one to the model. At each step, the variable added is the one that most improves the fit of the model. You can also specify groups of variables to treat as a unit during the selection process. An option enables you to specify the criterion for inclusion.

BACKWARD — specifies that variables be selected based on a backward-elimination algorithm. This method starts with a full model and eliminates variables one by one from the model. At each step, the variable with the smallest contribution to the model is deleted. You can also specify groups of variables to treat as a unit during the selection process. An option enables you to specify the criterion for exclusion.

STEPWISE — specifies that variables be selected for the model based on a stepwise-regression algorithm, which combines forward-selection and backward-elimination steps. This method is a modification of the forward-selection method in that variables already in the model do not necessarily stay there. You can also specify groups of variables to treat as a unit during the selection process. Again, options enable you to specify criteria for entry into the model and for remaining in the model.

MAXR — specifies that model formation be based on the maximum R^2 improvement. This method tries to find the best one-variable model, the best two-variable model, and so on. The MAXR method differs from the STEPWISE method in that many more models are evaluated. The MAXR method considers all possible variable exchanges before making any exchange. The STEPWISE method might

remove the "worst" variable without considering what the "best" remaining variable might accomplish, whereas MAXR would consider what the "best" remaining variable might accomplish. Consequently, model building based on the maximum R^2 improvement typically takes much longer to run than stepwise model building.

MINR — specifies that model formation be based on the minimum R^2 improvement. This method closely resembles MAXR, but the switch chosen is the one that produces the smallest increase in R^2.

RSQUARE — finds a specified number of models having the highest R^2 in each of a range of model sizes.

CP — finds a specified number of models with the lowest C_p within a range of model sizes.

ADJRSQ — finds a specified number of models having the highest adjusted R^2 within a range of model sizes.

The GLMSELECT procedure has been specifically designed for the purpose of model building in linear models. In addition to having a wider array of selection methods and criteria compared to the REG procedure, the GLMSELECT procedure also supports classification variables. See Chapter 42, "The GLMSELECT Procedure," for more information.

Regression with the REG and GLM Procedures

In terms of the assumptions about the basic model and the estimation principles, the REG and GLM procedures are very closely related. Both procedures estimate parameters by ordinary or weighted least squares and assume homoscedastic, uncorrelated model errors with zero mean. An assumption of normality of the model errors is not necessary for parameter estimation, but it is implied in confirmatory inference based on the parameter estimates—that is, the computation of tests, p-values, and confidence and prediction intervals.

The GLM procedure supports a CLASS statement for the levelization of classification variables; see the section "Parameterization of Model Effects" on page 368 in Chapter 18, "Shared Concepts and Topics," on the parameterization of classification variables in statistical models. Classification variables are accommodated in the REG procedure by the inclusion of the necessary dummy regressor variables.

Most of the statistics based on predicted and residual values that are available in PROC REG are also available in PROC GLM. However, PROC GLM does not produce collinearity diagnostics, influence diagnostics, or scatter plots. In addition, PROC GLM allows only one model and fits the full model.

Both procedures are interactive, in that they do not stop after processing a RUN statement. The procedures accept statements until a QUIT statement is submitted.

Response Surface Regression: The RSREG Procedure

The RSREG procedure fits a quadratic response-surface model, which is useful in searching for factor values that optimize a response. The following features in PROC RSREG make it preferable to other regression procedures for analyzing response surfaces:

- automatic generation of quadratic effects
- a lack-of-fit test
- solutions for critical values of the surface
- eigenvalues of the associated quadratic form
- a ridge analysis to search for the direction of optimum response

Partial Least Squares Regression: The PLS Procedure

The PLS procedure fits models by using any of a number of linear predictive methods, including *partial least squares* (PLS). Ordinary least squares regression, as implemented in the GLM or REG procedure, has the single goal of minimizing sample response prediction error by seeking linear functions of the predictors that explain as much variation in each response as possible. The techniques implemented in the PLS procedure have the additional goal of accounting for variation in the predictors under the assumption that directions in the predictor space that are well sampled should provide better prediction for *new* observations when the predictors are highly correlated. All of the techniques implemented in the PLS procedure work by extracting successive linear combinations of the predictors, called *factors* (also called *components* or *latent vectors*), which optimally address one or both of these two goals—explaining response variation and explaining predictor variation. In particular, the method of partial least squares balances the two objectives, seeking factors that explain both response and predictor variation.

Generalized Linear Regression

As outlined in the section "Generalized Linear Models" on page 37 of Chapter 3, "Introduction to Statistical Modeling with SAS/STAT Software," the class of generalized linear model generalizes the linear regression model in two ways:

- by allowing the data to come from a distribution that is a member of the exponential family of distributions
- by introducing a link function $g(\cdot)$ that provides a mapping between the linear predictor $\eta = \mathbf{x}'\boldsymbol{\beta}$ and the mean of the data, $g(\mathrm{E}[Y]) = \eta$. The link function is monotonic, so that $\mathrm{E}[Y] = g^{-1}(\eta)$ and $g^{-1}(\cdot)$ is called the inverse link function.

One of the most commonly used generalized linear regression models is the logistic model for binary or binomial data. Suppose that Y denotes a binary outcome variable that takes on the values 1 and 0 with probabilities π and $1 - \pi$, respectively. The probability π is also referred to as the "success probability," supposing that the coding $Y = 1$ corresponds to a success in a Bernoulli experiment. The success probability is also the mean of Y, and one of the aims of logistic regression analysis is to study how regressor variables affect the outcome probabilities or functions thereof, such as odds ratios.

The logistic regression model for π is defined by a linear predictor $\eta = \mathbf{x}'\boldsymbol{\beta}$ and the logit link function:

$$\text{logit}\{\Pr(Y = 0)\} = \log\left\{\frac{\pi}{1-\pi}\right\} = \mathbf{x}'\boldsymbol{\beta}$$

The inversely linked linear predictor function in this model is

$$\Pr(Y = 0) = \frac{1}{1 + \exp\{-\eta\}}$$

An extension of the dichotomous logistic regression model is models for multinomial (polychotomous) data. Two classes of models for multinomial data can be fit with procedures in SAS/STAT software: models for ordinal data that rely on cumulative link functions and models for nominal (unordered) outcomes that rely on generalized logits. The next section briefly discusses SAS/STAT procedures for logistic regression. See Chapter 8, "Introduction to Categorical Data Analysis Procedures," for more information about the comparison of the procedures mentioned there with respect to analysis of categorical responses.

Logistic Regression

The SAS/STAT procedures CATMOD, GENMOD, GLIMMIX, LOGISTIC, and PROBIT can fit generalized linear models for binary, binomial, and multinomial outcomes.

CATMOD provides maximum likelihood estimation for logistic regression, including the analysis of logits for dichotomous outcomes and the analysis of generalized logits for polychotomous outcomes. The CATMOD procedure can analyze data represented by a contingency table.

GENMOD is a general modeling procedure for generalized linear models. It estimates parameters by maximum likelihood. Like the LOGISTIC procedure, it uses CLASS and MODEL statements in SAS/STAT procedures to form the statistical model and can fit models to binary and ordinal outcomes. The GENMOD procedure does not fit generalized logit models for nominal outcomes. However, the procedure also provides the capability of solving generalized estimating equations (GEE) to model correlated data and can perform a Bayesian analysis.

GLIMMIX is a general modeling procedure for generalized linear mixed models. If the model does not contain random effects, the GLIMMIX procedure fits generalized linear models by the method of maximum likelihood. In the class of logistic regression models, the procedure can fit models to binary, binomial, ordinal, and nominal outcomes.

LOGISTIC is specifically designed for logistic regression and estimates parameters by maximum likelihood. The procedure fits the usual logistic regression model for binary data as well as models with cumulative link function for ordinal data (such as the proportional odds model) and the generalized logit model for nominal data. The LOGISTIC procedure offers a number of variable selection methods and can perform conditional and exact conditional logistic regression analysis.

PROBIT calculates maximum likelihood estimates of regression parameters and the natural (or threshold) response rate for quantal response data from biological assays or other discrete event data. This includes probit, logit, ordinal logistic, and extreme value (or gompit) regression models.

SURVEYLOGISTIC is designed for logistic regression and estimates parameters by maximum likelihood. The procedure incorporates complex survey sample designs, including designs with stratification, clustering, and unequal weighting.

Other Generalized Linear Models

When a generalized linear model is formed with distributions other than the binary, binomial, or multinomial, you can use the GENMOD and GLIMMIX procedures for parameter estimation and inference.

Both procedures can accommodate correlated observations, but they use different techniques to accomplish this goal. The GENMOD procedure can fit correlated data models via generalized estimating equations that rely on a first- and second-moment specification for the response data and a working correlation assumption. With the GLIMMIX procedure, you can model correlations between the observations by (1) specifying random effects in the conditional distribution that induce a marginal correlation structure or (2) direct modeling of the marginal dependence. The GLIMMIX procedure employs likelihood-based techniques in parameter estimation.

The GENMOD procedure supports a Bayesian analysis through its BAYES statement.

With the GLIMMIX procedure you can vary the distribution or link function on an observation-by-observation basis.

To fit a generalized linear model with a distribution that is not available in the GENMOD or GLIMMIX procedure, you can use the NLMIXED procedure and code the log-likelihood function of an observation with SAS programming statements.

Regression for Ill-Conditioned Data: The ORTHOREG Procedure

The ORTHOREG procedure performs linear least squares regression by using the Gentleman-Givens computational method, and it can produce more accurate parameter estimates for ill-conditioned data. PROC GLM and PROC REG produce very accurate estimates for most problems. However, if you have very ill-conditioned data, consider using the ORTHOREG procedure. The collinearity diagnostics in PROC REG can help you to determine whether PROC ORTHOREG would be useful.

Quantile Regression: The QUANTREG Procedure

The QUANTREG procedure models the effects of covariates on the conditional quantiles of a response variable by means of quantile regression.

Ordinary least squares regression models the relationship between one or more covariates X and the conditional mean of the response variable $\mathrm{E}[Y|X = x]$. Quantile regression extends the regression model to conditional quantiles of the response variable, such as the 90th percentile. Quantile regression is particularly useful when the rate of change in the conditional quantile, expressed by the regression coefficients, depends on the quantile. An advantage of quantile regression over least squares regression is its flexibility for modeling data with heterogeneous conditional distributions. Data of this type occur in many fields, including biomedicine, econometrics, and ecology.

Features that you will find in the QUANTREG procedure include the following:

- simplex, interior point, and smoothing algorithms for estimation
- sparsity, rank, and resampling methods for confidence intervals
- asymptotic and bootstrap methods to estimate covariance and correlation matrices of the parameter estimates
- Wald and likelihood ratio tests for the regression parameter estimates
- regression quantile spline fits

Nonlinear Regression

Recall from Chapter 3, "Introduction to Statistical Modeling with SAS/STAT Software," that a nonlinear regression model is a statistical model in which the mean function depends on the model parameters in a nonlinear function. The SAS/STAT procedures that can fit general, nonlinear models are the NLIN and NLMIXED procedures. The procedures have the following in common:

- Nonlinear models are fit by iterative methods.
- You must provide an expression for the model through programming statements.
- Analytic derivatives of the objective function with respect to the parameters are calculated automatically.
- A grid search is available to select the best starting values for the parameters from a set of starting points that you provide.

The following items reflect some important differences between the NLIN and NLMIXED procedures:

- Parameters are estimated by nonlinear least squares with the NLIN procedure and by maximum likelihood with the NLMIXED procedure.
- The NLMIXED procedure enables you to construct nonlinear models that contain normally distributed random effects.
- The NLIN procedure requires that you declare all model parameters in the PARAMETERS statement and assign starting values. The NLMIXED procedure determines the parameters in your model based on the PARAMETER statement and the other modeling statements. It is not necessary to supply starting values for all parameters in the NLMIXED procedure, but it is highly recommended.
- The residual variance is not a parameter in models fit with the NLIN procedure, but it is in models fit with the NLMIXED procedure.
- The default iterative optimization method in the NLIN procedure is the Gauss-Newton method; the default method in the NLMIXED procedure is the quasi-Newton method. Other optimization techniques are available in both procedures.

Nonlinear models are fit with iterative techniques that begin from starting values and attempt to iteratively improve on the estimates by updating the estimates. There is no guarantee that the solution achieved when the iterative algorithm converges will correspond to a global optimum.

Nonparametric Regression

Regression models that suppose a parametric form express the mean of an observation as a function of regressor variables $x_1, \cdots, x_k$ and parameters $\beta_1, \cdots, \beta_p$:

$$\mathrm{E}[Y] = f(x_1, \cdots, x_k; \beta_1, \cdots, \beta_p)$$

Nonparametric regression techniques not only relax the assumption of linearity in the regression parameters, but they also do not require that you specify a precise functional form for the relationship between response and regressor variables. Consider a regression problem where the relationship between response Y and regressor X is to be modeled. It is assumed that $\mathrm{E}[Y_i] = g(x_i) + \epsilon_i$, where $g(\cdot)$ is an unspecified regression function. Two primary approaches in nonparametric regression modeling are as follows:

- approximate $g(x_i)$ locally by a parametric function constructed from information in a local neighborhood of x_i
- approximate the unknown function $g(x_i)$ by a smooth, flexible function and determine the necessary smoothness and continuity properties from the data

The SAS/STAT procedures LOESS, GAM, and TPSPLINE fit nonparametric regression models by one of these methods.

Local Regression: The LOESS Procedure

The LOESS procedure implements a local regression approach for estimating regression surfaces pioneered by Cleveland, Devlin, and Grosse (1988). No assumptions about the parametric form of the entire regression surface are made with the LOESS procedure. Only a parametric form of the local approximation is specified by the user. Furthermore, the LOESS procedure is suitable when there are outliers in the data and a robust fitting method is necessary.

Smooth Function Approximation: The TPSPLINE Procedure

The TPSPLINE procedure decomposes the regressor contributions to the mean function into parametric components and into smooth functional components. Suppose that the regressor variables are collected into the vector $\mathbf{x}$ and that this vector is partitioned as $\mathbf{x} = [\mathbf{x}_1' \ \mathbf{x}_2']'$. The relationship between Y and $\mathbf{x}_2$ is linear (parametric) and the relationship between Y and $\mathbf{x}_1$ is nonparametric. The TPSPLINE procedure fits models of the form

$$\mathrm{E}[Y] = g(\mathbf{x}_1) + \mathbf{x}_2'\boldsymbol{\beta}$$

The function $g(\cdot)$ can be represented as a sequence of spline basis functions.

The parameters are estimated by a penalized least squares method. The penalty is applied to the usual least squares criterion to obtain a regression estimate that fits the data well and to prevent the fit from attempting to interpolate the data (fit the data too closely).

Generalized Additive Models: The GAM Procedure

Generalized additive models are nonparametric models in which one or more regressor variables are present and can make different smooth contributions to the mean function. For example, if $\mathbf{x}_i = [x_{i1}, x_{i2}, \cdots, x_{ik}]$ is a vector of k regressor for the ith observation, then an additive model represents the mean function as

$$\mathrm{E}[Y] = f_0 + f_1(x_{i1}) + f_2(x_{i2}) + \cdots + f_3(x_{i3})$$

The individual functions f_j can have a parametric or nonparametric form. If all f_j are parametric, the GAM procedure fits a fully parametric model. If some f_j are nonparametric, the GAM procedure fits a semiparametric model. Otherwise, the models are fully nonparametric.

The generalization of additive models is akin to the generalization for linear models: nonnormal data are accommodated by explicitly modeling the distribution of the data as a member of the exponential family and by applying a monotonic link function that provides a mapping between the predictor and the mean of the data.

Robust Regression: The ROBUSTREG Procedure

The ROBUSTREG procedure implements algorithms to detect outliers and provide resistant (stable) results in the presence of outliers. The ROBUSTREG procedure provides four such methods: M estimation, LTS estimation, S estimation, and MM estimation.

- M estimation was introduced by Huber (1973), and it is the simplest approach both computationally and theoretically. Although it is not robust with respect to leverage points, it is still used extensively in analyzing data for which it can be assumed that the contamination is mainly in the response direction.
- Least trimmed squares (LTS) estimation is a high breakdown value method introduced by Rousseeuw (1984). The breakdown value is a measure of the proportion of contamination that an estimation method can withstand and still maintain its robustness.
- S estimation is a high breakdown value method introduced by Rousseeuw and Yohai (1984). With the same breakdown value, it has a higher statistical efficiency than LTS estimation.
- MM estimation, introduced by Yohai (1987), combines high breakdown value estimation and M estimation. It has both the high breakdown property and a higher statistical efficiency than S estimation.

Regression with Transformations: The TRANSREG Procedure

The TRANSREG procedure can fit many standard linear models. In addition, PROC TRANSREG can find nonlinear transformations of the data and fit a linear model to the transformed variables. This is in contrast to PROC REG and PROC GLM, which fit linear models to data, or PROC NLIN, which fits nonlinear models to data. The TRANSREG procedure fits many types of linear models, including the following:

- ordinary regression and ANOVA
- metric and nonmetric conjoint analysis
- metric and nonmetric vector and ideal point preference mapping
- simple, multiple, and multivariate regression with variable transformations
- redundancy analysis with variable transformations
- canonical correlation analysis with variable transformations
- response surface regression with variable transformations

Interactive Features in the CATMOD, GLM, and REG Procedures

The CATMOD, GLM, and REG procedures do not stop after processing a RUN statement. More statements can be submitted as a continuation of the previous statements. Many new features in these procedures are useful to request after you have reviewed the results from previous statements. The procedures stop if a DATA step or another procedure is requested or if a QUIT statement is submitted.

Statistical Background in Linear Regression

The remainder of this chapter outlines the way in which many SAS/STAT regression procedures calculate various regression quantities. The discussion focuses on the linear regression models. General statistical background about linear statistical models can be found in the section "Linear Model Theory" of Chapter 3, "Introduction to Statistical Modeling with SAS/STAT Software."

Exceptions and further details are documented with individual procedures.

Linear Regression Models

In matrix notation, a linear model is written as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\mathbf{X}$ is the $(n \times k)$ design matrix (rows are observations and columns are the regressors), $\boldsymbol{\beta}$ is the $(k \times 1)$ vector of unknown parameters, and $\boldsymbol{\epsilon}$ is the $(n \times 1)$ vector of unobservable model errors. The first column of $\mathbf{X}$ is usually a vector of 1s and is used to estimate the intercept term.

The statistical theory of linear models is based on strict classical assumptions. Ideally, the response is measured with all the factors controlled in an experimentally determined environment. If you cannot control the factors experimentally, some tests must be interpreted as being conditional on the observed values of the regressors.

Other assumptions are as follows:

- The form of the model is correct (all important explanatory variables have been included). This assumption is reflected mathematically in the assumption of a zero mean of the model errors, $\mathrm{E}[\boldsymbol{\epsilon}] = \mathbf{0}$.
- Regressor variables are measured without error.
- The expected value of the errors is zero.
- The variance of the error (and thus the dependent variable) for the ith observation is σ^2/w_i, where w_i is a known weight factor. Usually, $w_i = 1$ for all i and thus σ^2 is the common, constant variance.
- The errors are uncorrelated across observations.

When hypotheses are tested, or when confidence and prediction intervals are computed, an additional assumption is made that the errors are normally distributed.

Parameter Estimates and Associated Statistics

The Least Squares Estimators

Least squares estimators of the regression parameters are found by solving the normal equations

$$\left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)\boldsymbol{\beta} = \mathbf{X}'\mathbf{W}\mathbf{Y}$$

for the vector $\boldsymbol{\beta}$, where $\mathbf{W}$ is a diagonal matrix with the observed weights on the diagonal. The resulting estimator of the parameter vector is

$$\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}\mathbf{Y}$$

This is an unbiased estimator, since

$$\begin{aligned}\mathrm{E}\left[\widehat{\boldsymbol{\beta}}\right] &= \left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}\mathrm{E}[\mathbf{Y}] \\ &= \left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}\mathbf{X}\boldsymbol{\beta} = \boldsymbol{\beta}\end{aligned}$$

Notice that the only assumption necessary in order for the least squares estimators to be unbiased is that of a zero mean of the model errors. If the estimator is evaluated at the observed data, it is referred to Subidxleast squaresestimate (Introduction to Regression)as the least squares estimate,

$$\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{W}\mathbf{y}$$

If the standard classical assumptions are met, the least squares estimators of the regression parameters are the best linear unbiased estimators (BLUE). In other words, the estimators have minimum variance in the class of estimators that are unbiased and that are linear functions of the responses. If the additional assumption of normally distributed errors is satisfied, then the following are true:

- The statistics that are computed have the proper sampling distributions for hypothesis testing.
- Parameter estimators are normally distributed.
- Various sums of squares are distributed proportional to chi-square, at least under proper hypotheses.
- Ratios of estimators to standard errors follow the Student's t distribution under certain hypotheses.
- Appropriate ratios of sums of squares follow an F distribution for certain hypotheses.

When regression analysis is used to model data that do not meet the assumptions, the results should be interpreted in a cautious, exploratory fashion. The significance probabilities under these circumstances are unreliable.

Box (1966) and Mosteller and Tukey (1977, Chapters 12 and 13) discuss the problems that are encountered with regression data, especially when the data are not under experimental control.

Estimating the Precision

Assume for the present that $\mathbf{X}'\mathbf{W}\mathbf{X}$ has full column rank k (this assumption is relaxed later). The variance of the error terms, $\mathrm{Var}[\epsilon_i] = \sigma^2$, is then estimated by the mean square error

$$s^2 = \mathrm{MSE} = \frac{\mathrm{SSE}}{n-k} = \frac{1}{n-k}\sum_{i=1}^{n} w_i \left(y_i - \mathbf{x}_i'\boldsymbol{\beta}\right)^2$$

where $\mathbf{x}_i'$ is the ith row of the design matrix $\mathbf{X}$. The residual variance estimate is also unbiased: $\mathrm{E}[s^2] = \sigma^2$.

The covariance matrix of the least squares estimators is

$$\mathrm{Var}\left[\widehat{\boldsymbol{\beta}}\right] = \sigma^2 \left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}$$

An estimate of the covariance matrix is obtained by replacing σ^2 with its estimate, s^2 in the preceding formula. This estimate is often referred to as COVB in SAS/STAT modeling procedures:

$$\mathrm{COVB} = \widehat{\mathrm{Var}}\left[\widehat{\boldsymbol{\beta}}\right] = s^2 \left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}$$

The correlation matrix of the estimates, often referred to as CORRB, is derived by scaling the covariance matrix: Let $\mathbf{S} = \mathrm{diag}\left(\left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}\right)^{-\frac{1}{2}}$. Then the correlation matrix of the estimates is

$$\mathrm{CORRB} = \mathbf{S}\left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1}\mathbf{S}$$

The estimated standard error of the ith parameter estimator is obtained as the square root of the ith diagonal element of the COVB matrix. Formally,

$$\mathrm{STDERR}\left(\widehat{\boldsymbol{\beta}}_i\right) = \sqrt{\left[s^2(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\right]_{ii}}$$

The ratio

$$t = \frac{\widehat{\boldsymbol{\beta}}_i}{\mathrm{STDERR}\left(\widehat{\boldsymbol{\beta}}_i\right)}$$

follows a Student's t distribution with $(n-k)$ degrees of freedom under the hypothesis that β_i is zero and provided that the model errors are normally distributed.

Regression procedures display the t ratio and the significance probability, which is the probability under the hypothesis $H\colon \beta_i = 0$ of a larger absolute t value than was actually obtained. When the probability is less than some small level, the event is considered so unlikely that the hypothesis is rejected.

Type I SS and Type II SS measure the contribution of a variable to the reduction in SSE. Type I SS measure the reduction in SSE as that variable is entered into the model in sequence. Type II SS are the increment in SSE that results from removing the variable from the full model. Type II SS are equivalent to the Type III and Type IV SS reported in the GLM procedure. If Type II SS are used in the numerator of an F test, the test is equivalent to the t test for the hypothesis that the parameter is zero. In polynomial models, Type I SS measure the contribution of each polynomial term after it is orthogonalized to the previous terms in the model. The four types of SS are described in Chapter 15, "The Four Types of Estimable Functions."

Coefficient of Determination

The coefficient of determination in a regression model, also known as the R-square statistic (R^2), measures the proportion of variability in the response that is explained by the regressor variables. In a linear regression model with intercept, R^2 is defined as

$$R^2 = 1 - \frac{\text{SSE}}{\text{SST}}$$

where SSE is the residual (error) sum of squares and SST is the total sum of squares corrected for the mean. The adjusted R^2 statistic is an alternative to R^2 that takes into account the number of parameters in the model. This statistic is calculated as

$$\text{ADJRSQ} = 1 - \frac{n-i}{n-p}\left(1 - R^2\right)$$

where n is the number of observations used to fit the model, p is the number of parameters in the model (including the intercept), and i is 1 if the model includes an intercept term, and 0 otherwise.

R^2 statistics also play an important indirect role in regression calculations. For example, the proportion of variability explained by regressing all other variables in a model on a particular regressor can provide insights into the interrelationship among the regressors.

Tolerances and variance inflation factors measure the strength of interrelationships among the regressor variables in the model. If all variables are orthogonal to each other, both tolerance and variance inflation are 1. If a variable is very closely related to other variables, the tolerance approaches 0 and the variance inflation gets very large. Tolerance (TOL) is 1 minus the R^2 that results from the regression of the other variables in the model on that regressor. Variance inflation (VIF) is the diagonal of $(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}$, if $(\mathbf{X}'\mathbf{W}\mathbf{X})$ is scaled to correlation form. The statistics are related as

$$\text{VIF} = \frac{1}{\text{TOL}}$$

Explicit and Implicit Intercepts

A linear model contains an *explicit* intercept if the $\mathbf{X}$ matrix contains a column whose nonzero values do not vary, typically a column of ones. Many SAS/STAT procedures automatically add this column of ones as the first column in the $\mathbf{X}$ matrix. Procedures that support a NOINT option in the MODEL statement provide the capability to suppress the automatic addition of the intercept column.

In general, models without intercept should be the exception, especially if your model does not contain classification variables. An overall intercept is provided in many models to adjust for the grand total or overall mean in your data. A simple linear regression without intercept, such as

$$\text{E}[Y_i] = \beta_1 x_i + \epsilon_i$$

assumes that Y has mean zero if X takes on the value zero. This might not be a reasonable assumption.

If you explicitly suppress the intercept in a statistical model, the calculation and interpretation of your results can change. For example, the exclusion of the intercept in the following PROC REG

statements leads to a different calculation of the R-square statistic. It also affects the calculation of the sum of squares in the analysis of variance for the model. For example, the model and error sum of squares add up to the uncorrected total sum of squares in the absence of an intercept.

```
proc reg;
   model y = x / noint;
quit;
```

Many statistical models contain an *implicit* intercept. This occurs when a linear function of one or more columns in the $\mathbf{X}$ matrix produces a column of constant, nonzero values. For example, the presence of a CLASS variable in the GLM parameterization always implies an intercept in the model. If a model contains an implicit intercept, adding an intercept to the model does not alter the quality of the model fit, but it changes the interpretation (and number) of the parameter estimates.

The way in which the implicit intercept is detected and accounted for in the analysis depends on the procedure. For example, the following statements in the GLM procedure lead to an implied intercept:

```
proc glm;
   class a;
   model y = a / solution noint;
run;
```

Whereas the analysis of variance table uses the uncorrected total sum of squares (due to the NOINT option), the implied intercept does not lead to a redefinition or recalculation of the R-square statistic (compared to the model without the NOINT option). Also, because the intercept is implied by the presence of the CLASS variable a in the model, the same error sum of squares results whether the NOINT option is specified or not.

A different approach is taken, for example, by the TRANSREG procedure. The ZERO=NONE option in the CLASS parameterization of the following statements leads to an implicit intercept model:

```
proc transreg;
   model ide(y) = class(a / zero=none) / ss2;
run;
```

The analysis of variance table or the regression fit statistics are not affected in the TRANSREG procedure. Only the interpretation of the parameter estimates changes because of the way in which the intercept is accounted for in the model.

Implied intercepts not only occur when classification effects are present in the model. They also occur with B-splines and other sets of constructed columns.

Models Not of Full Rank

If the $\mathbf{X}$ matrix is not of full rank, then a generalized inverse can be used to solve the normal equations to minimize the SSE:

$$\widehat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-}\mathbf{X}'\mathbf{W}\mathbf{y}$$

However, these estimates are not unique since there are an infinite number of solutions corresponding to different generalized inverses. PROC REG and other regression procedures choose a nonzero solution for all variables that are linearly independent of previous variables and a zero solution for other variables. This corresponds to using a generalized inverse in the normal equations, and the expected values of the estimates are the Hermite normal form of $\mathbf{X}'\mathbf{W}\mathbf{X}$ multiplied by the true parameters:

$$\mathrm{E}\left[\widehat{\boldsymbol{\beta}}\right] = \left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-}\left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)\boldsymbol{\beta}$$

Degrees of freedom for the estimates that correspond to singularities are not counted (reported as zero). The hypotheses that are not testable have t tests displayed as missing. The message that the model is not of full rank includes a display of the relations that exist in the matrix.

See the sections "Generalized Inverse Matrices" and "Linear Model Theory" in Chapter 3, "Introduction to Statistical Modeling with SAS/STAT Software," on the nature and construction of generalized inverses and their importance for statistical inference in linear models.

Predicted and Residual Values

After the model has been fit, predicted and residual values are usually calculated, graphed, and output. The predicted values are calculated from the estimated regression equation; the raw residuals are calculated as the observed minus the predicted value. Often other forms of residuals are used for model diagnostics, such as studentized or cumulative residuals. Some procedures can calculate standard errors of residuals, predicted mean values, and individual predicted values.

Consider the ith observation where $\mathbf{x}_i'$ is the row of regressors, $\widehat{\boldsymbol{\beta}}$ is the vector of parameter estimates, and s^2 is the estimate of the residual variance (the mean squared error). The *leverage* value of the ith observation is defined as

$$h_i = w_i \mathbf{x}_i' \left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-1} \mathbf{x}_i$$

where $\mathbf{X}$ is the design matrix for the observed data, $\mathbf{x}_i'$ is an arbitrary regressor vector (possibly but not necessarily a row of $\mathbf{X}$), $\mathbf{W}$ is a diagonal matrix with the observed weights on the diagonal, and w_i is the weight corresponding to $\mathbf{x}_i'$.

Then the predicted mean and the standard error of the predicted mean are

$$\begin{aligned} \widehat{y}_i &= \mathbf{x}_i'\widehat{\boldsymbol{\beta}} \\ \mathrm{STDERR}(\widehat{y}_i) &= \sqrt{s^2 h_i / w_i} \end{aligned}$$

The standard error of the individual (future) predicted value y_i is

$$\mathrm{STDERR}(y_i) = \sqrt{s^2(1 + h_i)/w_i}$$

If the predictor vector $\mathbf{x}_i$ corresponds to an observation in the analysis data, then the raw residual for that observation and the standard error of the raw residual are defined as

$$\begin{aligned} \mathrm{RESID}_i &= y_i - \mathbf{x}_i'\widehat{\boldsymbol{\beta}} \\ \mathrm{STDERR}(\mathrm{RESID}_i) &= \sqrt{s^2(1 - h_i)/w_i} \end{aligned}$$

The *studentized residual* is the ratio of the raw residual and its estimated standard error. Symbolically,

$$\text{STUDENT}_i = \frac{\text{RESID}_i}{\text{STDERR}(\text{RESID}_i)}$$

There are two kinds of intervals involving predicted values that are associated with a measure of confidence: the *confidence* interval for the mean value of the response and the *prediction* (or *forecasting*) interval for an individual observation. As discussed in the section "Mean Squared Error" in Chapter 3, "Introduction to Statistical Modeling with SAS/STAT Software," both intervals are based on the mean squared error of predicting a target based on the result of the model fit. The difference in the expressions for the confidence interval and the prediction interval comes about because the target of estimation is a constant in the case of the confidence interval (the mean of an observation) and the target is a random variable in the case of the prediction interval (a new observation).

For example, you can construct a confidence interval for the ith observation that contains the true mean value of the response with probability $1-\alpha$. The upper and lower limits of the confidence interval for the mean value are

$$\begin{aligned}\text{LowerM} &= \mathbf{x}_i'\widehat{\boldsymbol{\beta}} - t_{\alpha/2,\nu}\sqrt{s^2 h_i / w_i} \\ \text{UpperM} &= \mathbf{x}_i'\widehat{\boldsymbol{\beta}} + t_{\alpha/2,\nu}\sqrt{s^2 h_i / w_i}\end{aligned}$$

where $t_{\alpha/2,\nu}$ is the tabulated t quantile with degrees of freedom equal to the degrees of freedom for the mean squared error, $\nu = n - \text{rank}(\mathbf{X})$.

The limits for the prediction interval for an individual response are

$$\begin{aligned}\text{LowerI} &= \mathbf{x}_i'\widehat{\boldsymbol{\beta}} - t_{\alpha/2,\nu}\sqrt{s^2(1+h_i) / w_i} \\ \text{UpperI} &= \mathbf{x}_i'\widehat{\boldsymbol{\beta}} + t_{\alpha/2,\nu}\sqrt{s^2(1+h_i) / w_i}\end{aligned}$$

Influential observations are those that, according to various criteria, appear to have a large influence on the analysis. One measure of influence, Cook's D, measures the change to the estimates that results from deleting an observation:

$$\text{COOKD}_i = \frac{1}{k}\text{STUDENT}_i^2\left(\frac{\text{STDERR}(\widehat{y}_i)}{\text{STDERR}(\text{RESID}_i)}\right)^2$$

where k is the number of parameters in the model (including the intercept). For more information, see Cook (1977, 1979).

The *predicted residual* for observation i is defined as the residual for the ith observation that results from dropping the ith observation from the parameter estimates. The sum of squares of predicted residual errors is called the *PRESS statistic*:

$$\begin{aligned}\text{PRESID}_i &= \frac{\text{RESID}_i}{1-h_i} \\ \text{PRESS} &= \sum_{i=1}^{n} w_i \text{PRESID}_i^2\end{aligned}$$

Testing Linear Hypotheses

Testing of linear hypothesis based on estimable functions is discussed in the section "Test of Hypotheses" on page 66 in Chapter 3, "Introduction to Statistical Modeling with SAS/STAT Software," and the construction of special sets of estimable functions corresponding to Type I–Type IV hypotheses is discussed in Chapter 15, "The Four Types of Estimable Functions." In linear regression models, testing of general linear hypotheses follows along the same lines. Test statistics are usually formed based on sums of squares associated with the hypothesis in question. Furthermore, when $\mathbf{X}$ is of full rank—as is the case in many regression models—the consistency of the linear hypothesis is guaranteed.

Recall from Chapter 3, "Introduction to Statistical Modeling with SAS/STAT Software," that the general form of a linear hypothesis for the parameters is $H\colon \mathbf{L}\boldsymbol{\beta} = \mathbf{d}$, where $\mathbf{L}$ is $(q \times k)$, $\boldsymbol{\beta}$ is $(k \times 1)$, and $\mathbf{d}$ is $(q \times 1)$. To test this hypothesis, the linear function is taken with respect to the parameter estimates: $\mathbf{L}\widehat{\boldsymbol{\beta}} - \mathbf{d}$. This linear function in $\widehat{\boldsymbol{\beta}}$ has variance

$$\mathrm{Var}\left[\mathbf{L}\widehat{\boldsymbol{\beta}}\right] = \mathbf{L}\mathrm{Var}[\widehat{\boldsymbol{\beta}}]\mathbf{L}' = \sigma^2\mathbf{L}\left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-}\mathbf{L}'$$

The *sum of squares due to the hypothesis* is a simple quadratic form:

$$\mathrm{SS}(H) = \mathrm{SS}(\mathbf{L}\widehat{\boldsymbol{\beta}} - \mathbf{d}) = \left(\mathbf{L}\widehat{\boldsymbol{\beta}} - \mathbf{d}\right)'\left(\mathbf{L}\left(\mathbf{X}'\mathbf{W}\mathbf{X}\right)^{-}\mathbf{L}'\right)^{-1}\left(\mathbf{L}\widehat{\boldsymbol{\beta}} - \mathbf{d}\right)$$

If this hypothesis is testable, then $\mathrm{SS}(H)$ can be used in the numerator of an F statistic:

$$F = \frac{\mathrm{SS}(H)/q}{s^2} = \frac{\mathrm{SS}(\mathbf{L}\mathbf{b} - \mathbf{d})/q}{s^2}$$

If $\widehat{\boldsymbol{\beta}}$ is normally distributed, which follows as a consequence of normally distributed model errors, then this statistic follows an F distribution with q numerator degrees of freedom and $n - \mathrm{rank}(\mathbf{X})$ denominator degrees of freedom. Note that it was assumed in this derivation that $\mathbf{L}$ is of full row rank q.

Multivariate Tests

Multivariate hypotheses involve several dependent variables in the form

$$H\colon \mathbf{L}\boldsymbol{\beta}\mathbf{M} = \mathbf{d}$$

where $\mathbf{L}$ is a linear function on the regressor side, $\boldsymbol{\beta}$ is a matrix of parameters, $\mathbf{M}$ is a linear function on the dependent side, and $\mathbf{d}$ is a matrix of constants. The special case (handled by PROC REG) in which the constants are the same for each dependent variable is expressed as

$$(\mathbf{L}\boldsymbol{\beta} - \mathbf{c}\mathbf{j})\mathbf{M} = \mathbf{0}$$

where $\mathbf{c}$ is a column vector of constants and $\mathbf{j}$ is a row vector of 1s. The special case in which the constants are 0 is then

$$\mathbf{L}\boldsymbol{\beta}\mathbf{M} = \mathbf{0}$$

These multivariate tests are covered in detail in Morrison (1976), Timm (1975), Mardia, Kent, and Bibby (1979), Bock (1975), and other works cited in Chapter 9, "Introduction to Multivariate Procedures."

Notice that in contrast to the tests discussed in the preceding section, $\boldsymbol{\beta}$ here is a matrix of parameter estimates. Suppose that the matrix of estimates is denoted as $\mathbf{B}$. To test the multivariate hypothesis, construct two matrices, $\mathbf{H}$ and $\mathbf{E}$, that correspond to the numerator and denominator of a univariate F test:

$$\begin{aligned}\mathbf{H} &= \mathbf{M}'(\mathbf{LB}-\mathbf{cj})'(\mathbf{L}(\mathbf{X}'\mathbf{WX})^{-}\mathbf{L}')^{-1}(\mathbf{LB}-\mathbf{cj})\mathbf{M} \\ \mathbf{E} &= \mathbf{M}'\left(\mathbf{Y}'\mathbf{WY}-\mathbf{B}'(\mathbf{X}'\mathbf{WX})\mathbf{B}\right)\mathbf{M}\end{aligned}$$

Four test statistics, based on the eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$ or $(\mathbf{E}+\mathbf{H})^{-1}\mathbf{H}$, are formed. Let λ_i be the ordered eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$ (if the inverse exists), and let ξ_i be the ordered eigenvalues of $(\mathbf{E}+\mathbf{H})^{-1}\mathbf{H}$. It happens that $\xi_i = \lambda_i/(1+\lambda_i)$ and $\lambda_i = \xi_i/(1-\xi_i)$, and it turns out that $\rho_i = \sqrt{\xi_i}$ is the ith canonical correlation.

Let p be the rank of $(\mathbf{H}+\mathbf{E})$, which is less than or equal to the number of columns of $\mathbf{M}$. Let q be the rank of $\mathbf{L}(\mathbf{X}'\mathbf{WX})^{-}\mathbf{L}'$. Let v be the error degrees of freedom and $s = \min(p,q)$. Let $m = (|p-q|-1)/2$, and let $n = (v-p-1)/2$. Then the following statistics test the multivariate hypothesis in various ways, and their p-values can be approximated by F distributions. Note that in the special case that the rank of $\mathbf{H}$ is 1, all of these F statistics will be the same and the corresponding p-values will in fact be exact, since in this case the hypothesis is really univariate.

Wilks' Lambda

If

$$\Lambda = \frac{\det(\mathbf{E})}{\det(\mathbf{H}+\mathbf{E})} = \prod_{i=1}^{n}\frac{1}{1+\lambda_i} = \prod_{i=1}^{n}(1-\xi_i)$$

then

$$F = \frac{1-\Lambda^{1/t}}{\Lambda^{1/t}}\cdot\frac{rt-2u}{pq}$$

is approximately F distributed, where

$$\begin{aligned} r &= v - \frac{p-q+1}{2} \\ u &= \frac{pq-2}{4} \\ t &= \begin{cases} \sqrt{\frac{p^2q^2-4}{p^2+q^2-5}} & \text{if } p^2+q^2-5>0 \\ 1 & \text{otherwise} \end{cases}\end{aligned}$$

The degrees of freedom are pq and $rt-2u$. The distribution is exact if $\min(p,q)\le 2$. (See Rao 1973, p. 556.)

Pillai's Trace

If

$$\mathbf{V} = \text{trace}\left(\mathbf{H}(\mathbf{H}+\mathbf{E})^{-1}\right) = \sum_{i=1}^{n} \frac{\lambda_i}{1+\lambda_i} = \sum_{i=1}^{n} \xi_i$$

then

$$F = \frac{2n+s+1}{2m+s+1} \cdot \frac{\mathbf{V}}{s-\mathbf{V}}$$

is approximately F distributed with $s(2m+s+1)$ and $s(2n+s+1)$ degrees of freedom.

Hotelling-Lawley Trace

If

$$U = \text{trace}\left(\mathbf{E}^{-1}\mathbf{H}\right) = \sum_{i=1}^{n} \lambda_i = \sum_{i=1}^{n} \frac{\xi_i}{1-\xi_i}$$

then for $n > 0$

$$F = (U/c)((4+(pq+2)/(b-1))/(pq))$$

is approximately F distributed with pq and $4+(pq+2)/(b-1)$ degrees of freedom, where $b = (p+2n)(q+2n)/(2(2n+1)(n-1))$ and $c = (2+(pq+2)/(b-1))/(2n)$; while for $n \leq 0$

$$F = \frac{2(sn+1)\mathbf{U}}{s^2(2m+s+1)}$$

is approximately F with $s(2m+s+1)$ and $2(sn+1)$ degrees of freedom.

Roy's Maximum Root

If $\Theta = \lambda_1$, then

$$F = \Theta\frac{v-r+q}{r}$$

where $r = \max(p,q)$ is an upper bound on F that yields a lower bound on the significance level. Degrees of freedom are r for the numerator and $v-r+q$ for the denominator.

Tables of critical values for these statistics are found in Pillai (1960).

Exact Multivariate Tests

Beginning with SAS 9, if you specify the MSTAT=EXACT option in the appropriate statement, p-values for three of the four tests are computed exactly (Wilks' lambda, the Hotelling-Lawley

trace, and Roy's greatest root), and the p-values for the fourth (Pillai's trace) are based on an F approximation that is more accurate (but occasionally slightly more liberal) than the default. The exact p-values for Roy's greatest root benefit the most, since in this case the F approximation provides only a lower bound for the p-value. If you use the F-based p-value for this test in the usual way, declaring a test significant if $p < 0.05$, then your decisions might be very liberal. For example, instead of the nominal 5% Type I error rate, such a procedure can easily have an actual Type I error rate in excess of 30%. By contrast, basing such a procedure on the exact p-values will result in the appropriate 5% Type I error rate, under the usual regression assumptions.

The MSTAT=EXACT option is supported in the ANOVA, CANCORR, CANDISC, GLM, and REG procedures.

The exact p-values are based on the following sources:

- **Wilks' lambda:** Lee (1972), Davis (1979)
- **Pillai's trace:** Muller (1998)
- **Hotelling-Lawley trace:** Davis (1970), Davis (1980)
- **Roy's greatest root:** Davis (1972), Pillai and Flury (1984)

Note that, although the MSTAT=EXACT p-value for Pillai's trace is still approximate, it has "substantially greater accuracy" than the default approximation (Muller 1998).

Since most of the MSTAT=EXACT p-values are not based on the F distribution, the columns in the multivariate tests table corresponding to this approximation—in particular, the F value and the numerator and denominator degrees of freedom—are no longer displayed, and the column containing the p-values is labeled "P Value" instead of "Pr > F." Suppose, for example, you use the following PROC ANOVA statements to perform a multivariate analysis of an archaeological data set:

```
data Skulls;
   input Loc $20. Basal Occ Max;
   datalines;
Minas Graes, Brazil  2.068 2.070 1.580
Minas Graes, Brazil  2.068 2.074 1.602
Minas Graes, Brazil  2.090 2.090 1.613
Minas Graes, Brazil  2.097 2.093 1.613
Minas Graes, Brazil  2.117 2.125 1.663
Minas Graes, Brazil  2.140 2.146 1.681
Matto Grosso, Brazil 2.045 2.054 1.580
Matto Grosso, Brazil 2.076 2.088 1.602
Matto Grosso, Brazil 2.090 2.093 1.643
Matto Grosso, Brazil 2.111 2.114 1.643
Santa Cruz, Bolivia  2.093 2.098 1.653
Santa Cruz, Bolivia  2.100 2.106 1.623
Santa Cruz, Bolivia  2.104 2.101 1.653
;
```

```
proc anova data=Skulls;
   class Loc;
   model Basal Occ Max = Loc / nouni;
   manova h=Loc;
   ods select MultStat;
run;
```

The default multivariate tests, based on the F approximations, are shown in Figure 4.5.

Figure 4.5 Default Multivariate Tests

```
                          The ANOVA Procedure
                    Multivariate Analysis of Variance

               MANOVA Test Criteria and F Approximations for
                  the Hypothesis of No Overall Loc Effect
                       H = Anova SSCP Matrix for Loc
                           E = Error SSCP Matrix

                             S=2    M=0    N=3

Statistic                      Value    F Value    Num DF    Den DF    Pr > F

Wilks' Lambda               0.60143661       0.77         6        16    0.6032
Pillai's Trace              0.44702843       0.86         6        18    0.5397
Hotelling-Lawley Trace      0.58210348       0.75         6    9.0909    0.6272
Roy's Greatest Root         0.35530890       1.07         3         9    0.4109

         NOTE: F Statistic for Roy's Greatest Root is an upper bound.
                NOTE: F Statistic for Wilks' Lambda is exact.
```

If you specify MSTAT=EXACT in the MANOVA statement, as in the following statements, then the displayed output is the much simpler table shown in Figure 4.6.

```
proc anova data=Skulls;
   class Loc;
   model Basal Occ Max = Loc / nouni;
   manova h=Loc / mstat=exact;
   ods select MultStat;
run;
```

Figure 4.6 Multivariate Tests with MSTAT=EXACT

The ANOVA Procedure
Multivariate Analysis of Variance

MANOVA Tests for the Hypothesis of No Overall Loc Effect
H = Anova SSCP Matrix for Loc
E = Error SSCP Matrix

S=2 M=0 N=3

Statistic	Value	P-Value
Wilks' Lambda	0.60143661	0.6032
Pillai's Trace	0.44702843	0.5521
Hotelling-Lawley Trace	0.58210348	0.6337
Roy's Greatest Root	0.35530890	0.7641

Notice that the p-value for Roy's greatest root is substantially larger in the new table, and correspondingly more in line with the p-values for the other tests.

If you reference the underlying ODS output object for the table of multivariate statistics, it is important to note that its structure does not depend on the value of the MSTAT= specification. In particular, it always contains columns corresponding to both the default MSTAT=FAPPROX and the MSTAT=EXACT tests. Moreover, since the MSTAT=FAPPROX tests are relatively cheap to compute, the columns corresponding to them are always filled in, even though they are not displayed when you specify MSTAT=EXACT. On the other hand, for MSTAT=FAPPROX (which is the default), the column of exact p-values contains missing values, and is not displayed.

Comments on Interpreting Regression Statistics

In most applications, regression models are merely useful approximations. Reality is often so complicated that you cannot know what the true model is. You might have to choose a model more on the basis of what variables can be measured and what kinds of models can be estimated than on a rigorous theory that explains how the universe really works. However, even in cases where theory is lacking, a regression model can be an excellent predictor of the response if the model is carefully formulated from a large sample. The interpretation of statistics such as parameter estimates might nevertheless be highly problematic.

Statisticians usually use the word "prediction" in a technical sense. *Prediction* in this sense does not refer to "predicting the future" (statisticians call that *forecasting*) but rather to guessing the response from the values of the regressors in an observation taken under the same circumstances as the sample from which the regression equation was estimated. If you developed a regression model for predicting consumer preferences in 1977, it might not give very good predictions in 2007 no matter how well it did in 1977. If it is the future you want to predict, your model must include whatever relevant factors might change over time. If the process you are studying does in fact change over time, you must take observations at several, perhaps many, different times. Analysis of such data is the province of SAS/STAT procedures such as MIXED and GLIMMIX and SAS/ETS

procedures such as AUTOREG and STATESPACE. See Chapter 38, "The GLIMMIX Procedure," and Chapter 56, "The MIXED Procedure," for more information about modeling serial correlation in longitudinal, repeated measures, or time series data with SAS/STAT mixed modeling procedures. See the *SAS/ETS User's Guide* for more information about the AUTOREG and STATESPACE procedures.

The comments in the rest of this section are directed toward linear least squares regression. For more detailed discussions of the interpretation of regression statistics, see Darlington (1968), Mosteller and Tukey (1977), Weisberg (1985), and Younger (1979).

Interpreting Parameter Estimates from a Controlled Experiment

Parameter estimates are easiest to interpret in a controlled experiment in which the regressors are manipulated independently of each other. In a well-designed experiment, such as a randomized factorial design with replications in each cell, you can use lack-of-fit tests and estimates of the standard error of prediction to determine whether the model describes the experimental process with adequate precision. If so, a regression coefficient estimates the amount by which the mean response changes when the regressor is changed by one unit while all the other regressors are unchanged. However, if the model involves interactions or polynomial terms, it might not be possible to interpret individual regression coefficients. For example, if the equation includes both linear and quadratic terms for a given variable, you cannot physically change the value of the linear term without also changing the value of the quadratic term. Sometimes it might be possible to recode the regressors, such as by using orthogonal polynomials, to simplify the interpretation.

If the nonstatistical aspects of the experiment are also treated with sufficient care (such as the use of placebos and double blinds), then you can state conclusions in causal terms; that is, this change in a regressor causes that change in the response. Causality can never be inferred from statistical results alone or from an observational study.

If the model you fit is not the true model, then the parameter estimates can depend strongly on the particular values of the regressors used in the experiment. For example, if the response is actually a quadratic function of a regressor but you fit a linear function, the estimated slope can be a large negative value if you use only small values of the regressor, a large positive value if you use only large values of the regressor, or near zero if you use both large and small regressor values. When you report the results of an experiment, it is important to include the values of the regressors. It is also important to avoid extrapolating the regression equation outside the range of regressors in the sample.

Interpreting Parameter Estimates from an Observational Study

In an observational study, parameter estimates can be interpreted as the expected difference in response of two observations that differ by one unit on the regressor in question and that have the same values for all other regressors. You cannot make inferences about "changes" in an observational study since you have not actually changed anything. It might not be possible even in principle to change one regressor independently of all the others. Neither can you draw conclusions about causality without experimental manipulation.

If you conduct an observational study and you do not know the true form of the model, interpretation of parameter estimates becomes even more convoluted. A coefficient must then be interpreted as an average over the sampled population of expected differences in response of observations that differ by one unit on only one regressor. The considerations that are discussed under controlled experiments for which the true model is not known also apply.

Comparing Parameter Estimates

Two coefficients in the same model can be directly compared only if the regressors are measured in the same units. You can make any coefficient large or small just by changing the units. If you convert a regressor from feet to miles, the parameter estimate is multiplied by 5280.

Sometimes standardized regression coefficients are used to compare the effects of regressors measured in different units. Standardized estimates are defined as the estimates that result when all variables are standardized to a mean of 0 and a variance of 1. Standardized estimates are computed by multiplying the original estimates by the sample standard deviation of the regressor variable and dividing by the sample standard deviation of the dependent variable.

Standardizing the variables effectively makes the standard deviation the unit of measurement. This makes sense only if the standard deviation is a meaningful quantity, which usually is the case only if the observations are sampled from a well-defined population. In a controlled experiment, the standard deviation of a regressor depends on the values of the regressor selected by the experimenter. Thus, you can make a standardized regression coefficient large by using a large range of values for the regressor.

In some applications you might be able to compare regression coefficients in terms of the practical range of variation of a regressor. Suppose that each independent variable in an industrial process can be set to values only within a certain range. You can rescale the variables so that the smallest possible value is zero and the largest possible value is one. Then the unit of measurement for each regressor is the maximum possible range of the regressor, and the parameter estimates are comparable in that sense. Another possibility is to scale the regressors in terms of the cost of setting a regressor to a particular value, so comparisons can be made in monetary terms.

Correlated Regressors

In an experiment, you can often select values for the regressors such that the regressors are orthogonal (not correlated with each other). Orthogonal designs have enormous advantages in interpretation. With orthogonal regressors, the parameter estimate for a given regressor does not depend on which other regressors are included in the model, although other statistics such as standard errors and p-values might change.

If the regressors are correlated, it becomes difficult to disentangle the effects of one regressor from another, and the parameter estimates can be highly dependent on which regressors are used in the model. Two correlated regressors might be nonsignificant when tested separately but highly significant when considered together. If two regressors have a correlation of 1.0, it is impossible to separate their effects.

It might be possible to recode correlated regressors to make interpretation easier. For example, if X and Y are highly correlated, they could be replaced in a linear regression by $X + Y$ and $X - Y$ without changing the fit of the model or statistics for other regressors.

Errors in the Regressors

If there is error in the measurements of the regressors, the parameter estimates must be interpreted with respect to the measured values of the regressors, not the true values. A regressor might be statistically nonsignificant when measured with error even though it would have been highly significant if measured accurately.

Probability Values (*p*-Values)

Probability values (p-values) do not necessarily measure the importance of a regressor. An important regressor can have a large (nonsignificant) p-value if the sample is small, if the regressor is measured over a narrow range, if there are large measurement errors, or if another closely related regressor is included in the equation. An unimportant regressor can have a very small p-value in a large sample. Computing a confidence interval for a parameter estimate gives you more useful information than just looking at the p-value, but confidence intervals do not solve problems of measurement errors in the regressors or highly correlated regressors.

Interpreting R^2

R^2 is usually defined as the proportion of variance of the response that is predictable from (can be explained by) the regressor variables. It might be easier to interpret $\sqrt{1 - R^2}$, which is approximately the factor by which the standard error of prediction is reduced by the introduction of the regressor variables.

R^2 is easiest to interpret when the observations, including the values of both the regressors and response, are randomly sampled from a well-defined population. Nonrandom sampling can greatly distort R^2. For example, excessively large values of R^2 can be obtained by omitting from the sample observations with regressor values near the mean.

In a controlled experiment, R^2 depends on the values chosen for the regressors. A wide range of regressor values generally yields a larger R^2 than a narrow range. In comparing the results of two experiments on the same variables but with different ranges for the regressors, you should look at the standard error of prediction (root mean square error) rather than R^2.

Whether a given R^2 value is considered to be large or small depends on the context of the particular study. A social scientist might consider an R^2 of 0.30 to be large, while a physicist might consider 0.98 to be small.

You can always get an R^2 arbitrarily close to 1.0 by including a large number of completely unrelated regressors in the equation. If the number of regressors is close to the sample size, R^2 is very biased. In such cases, the adjusted R^2 and related statistics discussed by Darlington (1968) are less misleading.

If you fit many different models and choose the model with the largest R^2, all the statistics are biased and the p-values for the parameter estimates are not valid. Caution must be taken with the interpretation of R^2 for models with no intercept term. As a general rule, no-intercept models should be fit only when theoretical justification exists and the data appear to fit a no-intercept framework. The R^2 in those cases is measuring something different (see Kvalseth 1985).

Incorrect Data Values

All regression statistics can be seriously distorted by a single incorrect data value. A decimal point in the wrong place can completely change the parameter estimates, R^2, and other statistics. It is important to check your data for outliers and influential observations. Residual and influence diagnostics are particularly useful in this regard.

When a data point is declared as influential or as outlying as measured by a particular model diagnostic, this does not imply that the case should be excluded from the analysis. The label "outlier" does not have a negative connotation. It means that a data point is unusual with respect to the model at hand. If your data follow a strong curved trend and you fit a linear regression, then many data points might be labeled as outliers not because they are "bad" or incorrect data values, but because your model is not appropriate.

References

Allen, D. M. (1971), "Mean Square Error of Prediction as a Criterion for Selecting Variables," *Technometrics*, 13, 469–475.

Allen, D. M. and Cady, F. B. (1982), *Analyzing Experimental Data by Regression*, Belmont, CA: Lifetime Learning Publications.

Belsley, D. A., Kuh, E., and Welsch, R .E. (1980), *Regression Diagnostics*, New York: John Wiley & Sons.

Bock, R. D. (1975), *Multivariate Statistical Methods in Behavioral Research*, New York: McGraw-Hill.

Box, G. E. P. (1966), "The Use and Abuse of Regression," *Technometrics*, 8, 625–629.

Cleveland, W. S., Devlin, S. J., and Grosse, E. (1988), "Regression by Local Fitting," *Journal of Econometrics*, 37, 87–114.

Cook, R. D. (1977), "Detection of Influential Observations in Linear Regression," *Technometrics*, 19, 15–18.

Cook, R. D. (1979), "Influential Observations in Linear Regression," *Journal of the American Statistical Association*, 74, 169–174.

Daniel, C. and Wood, F. (1980), *Fitting Equations to Data*, Revised Edition, New York: John Wiley & Sons.

Darlington, R. B. (1968), "Multiple Regression in Psychological Research and Practice," *Psychological Bulletin*, 69, 161–182.

Davis, A. W. (1970), "Differential Equation of Hotelling's Generalized T^2," *Annals of Statistics*, 39, 815–832.

Davis, A. W. (1972), "On the Marginal Distributions of the Latent Roots of the Multivariate Beta Matrix," *Biometrika*, 43, 1664–1670.

Davis, A. W. (1979), "On the Differential Equation for Meijer $G_{p,p}^{0,0}$ Function, and Further Wilks's Likelihood Ratio Criterion," *Biometrika*, 66, 519–531.

Davis, A. W. (1980), "Further Tabulation of Hotelling's Generalized T^2," *Communications in Statistics, Part B*, 9, 321–336.

Draper, N. and Smith, H. (1981), *Applied Regression Analysis*, Second Edition, New York: John Wiley & Sons.

Durbin, J. and Watson, G. S. (1951), "Testing for Serial Correlation in Least Squares Regression," *Biometrika*, 37, 409–428.

Freund, R. J., Littell, R. C., and Spector P. C. (1991), *SAS System for Linear Models*, Cary, NC: SAS Institute Inc.

Freund, R. J. and Littell, R. C. (1986), *SAS System for Regression, 1986 Edition*, Cary, NC: SAS Institute Inc.

Goodnight, J. H. (1979), "A Tutorial on the SWEEP Operator," *The American Statistician*, 33, 149−158. (Also available as SAS Technical Report R-106, *The Sweep Operator: Its Importance in Statistical Computing*, Cary, NC: SAS Institute Inc.)

Hawkins, D. M. (1980), "A Note on Fitting a Regression with No Intercept Term," *The American Statistician*, 34, 233.

Hosmer, D. W., Jr. and Lemeshow, S. (1989), *Applied Logistic Regression*, New York: John Wiley & Sons.

Huber, P. J. (1973), "Robust Regression: Asymptotics, Conjectures and Monte Carlo," *Annals of Statistics*, 1, 799-821.

Johnston, J. (1972), *Econometric Methods*, New York: McGraw-Hill.

Kennedy, W. J. and Gentle, J. E. (1980), *Statistical Computing*, New York: Marcel Dekker.

Kvalseth, T. O. (1985), "Cautionary Note about R^2," *The American Statistician*, 39, 279–285.

Lee, Y. (1972), "Some Results on the Distribution of Wilk's Likelihood Ratio Criterion," *Biometrika*, 95, 649.

Mallows, C. L. (1973), "Some Comments on C_p," *Technometrics*, 15, 661–75.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979), *Multivariate Analysis*, London: Academic Press.

Morrison, D. F. (1976), *Multivariate Statistical Methods*, Second Edition, New York: McGraw-Hill.

Mosteller, F. and Tukey, J. W. (1977), *Data Analysis and Regression*, Reading, MA: Addison-Wesley.

Muller, K. (1998), "A New F Approximation for the Pillai-Bartlett Trace Under H0," *Journal of Computational and Graphical Statistics*, 7, 131–137.

Neter, J. and Wasserman, W. (1974), *Applied Linear Statistical Models*, Homewood, IL: Irwin.

Pillai, K. C. S. (1960), *Statistical Table for Tests of Multivariate Hypotheses*, Manila: The Statistical Center, University of Philippines.

Pillai, K. C. S. and Flury, B. N. (1984), "Percentage Points of the Largest Characteristic Root of the Multivariate Beta Matrix," *Communications in Statistics, Part A*, 13, 2199–2237.

Pindyck, R. S. and Rubinfeld, D. L. (1981), *Econometric Models and Econometric Forecasts*, Second Edition, New York: McGraw-Hill.

Rao, C. R. (1973), *Linear Statistical Inference and Its Applications*, Second Edition, New York: John Wiley & Sons.

Rawlings, J. O. (1988), *Applied Regression Analysis: A Research Tool*, Pacific Grove, CA: Wadsworth & Brooks/Cole Advanced Books & Software.

Rousseeuw, P. J. (1984), "Least Median of Squares Regression," *Journal of the American Statistical Association*, 79, 871–880.

Rousseeuw, P. J. and Yohai, V. (1984), "Robust Regression by Means of S Estimators," in *Robust and Nonlinear Time Series Analysis*, ed. J. Franke, W. Härdle, and R. D. Martin, Lecture Notes in Statistics, 26, New York: Springer-Verlag, 256–274.

Timm, N. H. (1975), *Multivariate Analysis with Applications in Education and Psychology*, Monterey, CA: Brooks-Cole.

Weisberg, S. (1985), *Applied Linear Regression*, Second Edition. New York: John Wiley & Sons.

Yohai V. J. (1987), "High Breakdown Point and High Efficiency Robust Estimates for Regression," *Annals of Statistics*, 15, 642–656.

Younger, M. S. (1979), *Handbook for Linear Regression*, North Scituate, MA: Duxbury Press.

Chapter 5

Introduction to Analysis of Variance Procedures

Contents

Overview: Analysis of Variance Procedures **115**
Procedures That Perform Sum of Squares Analysis of Variance 117
Procedures That Perform General Analysis of Variance 118
Statistical Details for Analysis of Variance **119**
From Sums of Squares to Linear Hypotheses 119
Tests of Effects Based on Expected Mean Squares 120
Analysis of Variance for Fixed-Effect Models **121**
PROC GLM for General Linear Models 121
PROC ANOVA for Balanced Designs 122
Comparing Group Means 122
PROC TTEST for Comparing Two Groups 123
Analysis of Variance for Categorical Data and Generalized Linear Models **123**
Nonparametric Analysis of Variance **124**
Constructing Analysis of Variance Designs **124**
References **125**

Overview: Analysis of Variance Procedures

The statistical term "analyis of variance" is used in a variety of circumstances in statistical theory and applications. In the *narrowest* sense, and the original sense of the phrase, it signifies a decomposition of a variance into contributing components. This was the sense used by R. A. Fisher when he defined the term to mean the expression of genetic variance as a sum of variance components due to environment, heredity, and so forth:

$$\sigma^2 = \sigma_1^2 + \sigma_2^2 + \cdots + \sigma_p^2$$

In this sense of the term, the SAS/STAT procedures that fit variance component models, such as the GLIMMIX, HPMIXED, MIXED, NESTED, and VARCOMP procedures, are "true" analysis of variance procedures.

Analysis of variance methodology in a slightly broader sense—and the sense most frequently understood today—applies the idea of an additive decomposition of variance to an additive decomposition of *sums of squares*, whose expected values are functionally related to components of variation. A collection of sums of squares that measure and can be used for inference about meaningful features of a model is called a *sum of squares analysis of variance*, whether or not such a collection is an additive decomposition. In a linear model, the decomposition of sums of squares can be expressed in terms of projections onto orthogonal subspaces spanned by the columns of the design matrix $\mathbf{X}$. This is the general approach followed in the section "Analysis of Variance" on page 63 in Chapter 3, "Introduction to Statistical Modeling with SAS/STAT Software." Depending on the statistical question at hand, the projections can be formulated based on estimable functions, with different types of estimable functions giving rise to different types of sums of squares. Note that not all sum of squares analyses necessarily correspond to additive decompositions. For example, the Type III sums of squares often test hypotheses about the model that are more meaningful than those corresponding to the Type I sums of squares. But while the Type I sums of squares additively decompose the sum of squares due to all model contributions, the Type III sums of squares do not necessarily add up to any useful quantity. The four types of estimable functions in SAS/STAT software, their interpretation, and their construction are discussed in Chapter 15, "The Four Types of Estimable Functions." The application of sum of squares analyses is not necessarily limited to models with classification effects (factors). The methodology also applies to linear regression models that contain only continuous regressor variables.

An even broader sense of the term "analysis of variance" pertains to statistical models that contain classification effects (factors), and in particular, to models that contain *only* classification effects. Any statistical approach that measures features of such a model and can be used for inference is called a *general analysis of variance*. Thus the procedures for general analysis of variance in SAS/STAT are considered to be those that can fit statistical models containing factors, whether the data are experimental or observational. Some procedures for general analysis of variance have a statistical estimation principle that gives rise to a sum of squares analysis as discussed previously; others express a factor's contribution to the model fit in some other form. Note that this view of analysis of variance includes, for example, maximum likelihood estimation in generalized linear models with the GENMOD procedure, restricted maximum likelihood estimation in linear mixed models with the MIXED procedure, the estimation of variance components with the VARCOMP procedure, the comparison of means of groups with the TTEST procedure, and the nonparametric analysis of rank scores with the NPAR1WAY procedure, and so on.

In summary, analysis of variance in the contemporary sense of statistical modeling and analysis is more aptly described as *analysis of variation*, the study of the influences on the variation of a phenomenon. This can take, for example, the following forms:

- an analysis of variance table based on sums of squares followed by more specific inquiries into the relationship among factors and their levels
- a deviance decomposition in a generalized linear model
- a series of Type III tests followed by comparisons of least squares means in a mixed model

Procedures That Perform Sum of Squares Analysis of Variance

The flagship procedure in SAS/STAT software for linear modeling with sum of squares analysis techniques is the GLM procedure. It handles most standard analysis of variance problems. The following list provides descriptions of PROC GLM and other procedures that are used for more specialized situations:

ANOVA — performs analysis of variance, multivariate analysis of variance, and repeated measures analysis of variance for *balanced* designs. PROC ANOVA also performs multiple comparison tests on arithmetic means.

GLM — performs analysis of variance, regression, analysis of covariance, repeated measures analysis, and multivariate analysis of variance. PROC GLM produces several diagnostic measures, performs tests for random effects, provides contrasts and estimates for customized hypothesis tests, provides tests for means adjusted for covariates, and performs multiple-comparison tests on both arithmetic and adjusted means.

LATTICE — computes the analysis of variance and analysis of simple covariance for data from an experiment with a lattice design. PROC LATTICE analyzes balanced square lattices, partially balanced square lattices, and some rectangular lattices.

MIXED — performs mixed model analysis of variance and repeated measures analysis of variance via covariance structure modeling. When you choose one of the method-of-moment estimation techniques, the MIXED procedure produces an analysis of variance table with sums of squares, mean squares, and expected mean squares. PROC MIXED constructs statistical tests and intervals, allows customized contrasts and estimates, and computes empirical Bayes predictions.

NESTED — performs analysis of variance and analysis of covariance for purely nested random models.

ORTHOREG — performs regression by using the Gentleman-Givens computational method. For ill-conditioned data, PROC ORTHOREG can produce more accurate parameter estimates than other procedures, such as PROC GLM. See Chapter 63, "The ORTHOREG Procedure," for more information.

VARCOMP — estimates variance components for random or mixed models. If you choose the METHOD=TYPE1 or METHOD=GRR option, the VARCOMP procedure produces an analysis of variance table with sums of squares that correspond to the random effects in your models.

TRANSREG — fits univariate and multivariate linear models, optionally with spline and other nonlinear transformations. Models include ordinary regression and ANOVA, multiple and multivariate regression, metric and nonmetric conjoint analysis, metric and nonmetric vector and ideal point preference mapping, redundancy analysis, canonical correlation, and response surface regression. See Chapter 90, "The TRANSREG Procedure," for more information.

Procedures That Perform General Analysis of Variance

Many procedures in SAS/STAT enable you to incorporate classification effects into your model and to perform statistical inferences for experimental factors and their interactions. These procedures do not necessarily rely on sums of squares decompositions to perform these inferences.

CATMOD	fits linear models and performs analysis of variance and repeated measures analysis of variance for categorical responses. See Chapter 8, "Introduction to Categorical Data Analysis Procedures," and Chapter 28, "The CATMOD Procedure," for more information.
GENMOD	fits generalized linear models. PROC GENMOD is especially suited for responses with discrete outcomes, and it performs logistic regression and Poisson regression as well as fitting generalized estimating equations for repeated measures data. Bayesian analysis capabilities for generalized linear models are also available with the GENMOD procedure. See Chapter 8, "Introduction to Categorical Data Analysis Procedures," and Chapter 37, "The GENMOD Procedure," for more information.
GLIMMIX	fits generalized linear mixed models by likelihood-based methods. PROC GLIMMIX offers many facilities for analyzing and comparing classification effects and their levels, including multiplicity-adjusted linear estimates. See Chapter 38, "The GLIMMIX Procedure," for more information.
LOGISTIC	fits logistic models for binomial and ordinal outcomes. PROC LOGISTIC provides a wide variety of model-building methods and computes numerous regression diagnostics. See Chapter 8, "Introduction to Categorical Data Analysis Procedures," and Chapter 51, "The LOGISTIC Procedure," for more information.
NPAR1WAY	performs nonparametric one-way analysis of rank scores.
TTEST	compares the means of two groups of observations.

The following section discusses procedures in SAS/STAT that compute analysis of variance in models with classification factors in the narrow sense—that is, they produce analysis of variance tables and form F tests based on sums of squares, mean squares, and expected mean squares.

The subsequent sections discuss procedures that perform statistical inference in models with classification effects in the broader sense.

The following section also presents an overview of some of the fundamental features of analysis of variance. Subsequent sections describe how this analysis is performed with procedures in SAS/STAT software. For more detail, see the chapters for the individual procedures. Additional sources are described in the section "References" on page 125.

Statistical Details for Analysis of Variance

From Sums of Squares to Linear Hypotheses

Analysis of variance (ANOVA) is a technique for analyzing data in which one or more *response* (or *dependent* or simply Y) variables are measured under various conditions identified by one or more classification variables. The combinations of levels for the classification variables form the cells of the design for the data. This design can be the result of a controlled experiment or the result of an observational study in which you observe factors and factor level combinations in an uncontrolled environment. For example, an experiment might measure weight change (the dependent variable) for men and women who participated in three different weight-loss programs. The six cells of the design are formed by the six combinations of gender (men, women) and program (A, B, C).

In an analysis of variance, the variation in the response is separated into variation attributable to differences between the classification variables and variation attributable to random error. An analysis of variance constructs tests to determine the significance of the classification effects. A typical goal in such an analysis is to compare means of the response variable for various combinations of the classification variables.

The least squares principle is central to computing sums of squares in analysis of variance models. Suppose that you are fitting the linear model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ and that the error terms satisfy the usual assumptions (uncorrelated, zero mean, homogeneous variance). Further, suppose that $\mathbf{X}$ is partitioned according to several model effects, $\mathbf{X} = [\mathbf{X}_1\ \mathbf{X}_2 \cdots \mathbf{X}_k]$. If $\widehat{\boldsymbol{\beta}}$ denotes the ordinary least squares solution for this model, then the sum of squares attributable to the overall model can be written as

$$\text{SSM} = \widehat{\boldsymbol{\beta}}'\mathbf{X}'\mathbf{Y} = \mathbf{Y}'\mathbf{H}\mathbf{Y}$$

where $\mathbf{H}$ is the "hat" matrix $\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'$. (This model sum of squares is not yet corrected for the presence of an explicit or implied intercept. This adjustment would consist of subtracting $n\overline{Y}^2$ from SSM.) Because of the properties of the hat matrix $\mathbf{H}$, you can write $\mathbf{X}' = \mathbf{X}'\mathbf{H}$ and $\mathbf{H}\mathbf{X} = \mathbf{X}$. The (uncorrected) model sum of squares thus can also be written as

$$\text{SSM} = \widehat{\boldsymbol{\beta}}'(\mathbf{X}'\mathbf{X})\widehat{\boldsymbol{\beta}}$$

This step is significant, because it demonstrates that sums of squares can be identified with quadratic functions in the least squares coefficients. The generalization of this idea is to do the following:

- consider hypotheses of interest in an analysis of variance model
- express the hypotheses in terms of linear estimable functions of the parameters
- compute the sums of squares associated with the estimable function
- construct statistical tests based on the sums of squares

Decomposing a model sum of squares into sequential, additive components, testing the significance of experimental factors, comparing factor levels, and performing other statistical inferences fall within this generalization. Suppose that $\mathbf{L}\boldsymbol{\beta}$ is an estimable function (see the section "Estimable Functions" on page 66 in Chapter 3, "Introduction to Statistical Modeling with SAS/STAT Software," and Chapter 15, "The Four Types of Estimable Functions," for details). The sum of squares associated with the hypothesis $H\colon \mathbf{L}\boldsymbol{\beta} = \mathbf{0}$ is

$$\mathrm{SS}(H) = \mathrm{SS}(\mathbf{L}\boldsymbol{\beta} = \mathbf{0}) = \widehat{\boldsymbol{\beta}}'\mathbf{L}'\left(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}'\right)^{-1}\mathbf{L}\widehat{\boldsymbol{\beta}}$$

One application would be to form sums of squares associated with the different components of $\mathbf{X}$. For example, you can form a matrix $\mathbf{L}_2$ matrix such that $\mathbf{L}_2\boldsymbol{\beta} = \mathbf{0}$ tests the effect of adding the columns for $\mathbf{X}_2$ to an empty model or to test the effect of adding $\mathbf{X}_2$ to a model that already contains $\mathbf{X}_1$.

These sums of squares can also be expressed as the difference between two residual sums of squares, since $\mathbf{L}\boldsymbol{\beta} = \mathbf{0}$ can be thought of as a (linear) restriction on the parameter estimates in the model:

$$\mathrm{SS}(H) = \mathrm{SSR}(\text{constrained model}) - \mathrm{SSR}(\text{full model})$$

If, in addition to the usual assumptions mentioned previously, the model errors are assumed to be normally distributed, then $\mathrm{SS}(H)$ follows a distribution that is proportional to a chi-square distribution. This fact, and the independence of $\mathrm{SS}(H)$ from the residual sum of squares, enables you to construct F tests based on sums of squares in least squares models.

The extension of sum of squares analysis of variance to general analysis of variance for classification effects depends on the fact that the distributional properties of quadratic forms in normal random variables are well understood. It is not necessary to first formulate a sum of squares to arrive at an exact or even approximate F test. The generalization of the expression for $\mathrm{SS}(H)$ is to form test statistics based on quadratic forms

$$\widehat{\boldsymbol{\beta}}'\mathbf{L}'\,\mathrm{Var}\left[\mathbf{L}\widehat{\boldsymbol{\beta}}\right]^{-1}\mathbf{L}\widehat{\boldsymbol{\beta}}$$

that follow a chi-square distribution if $\widehat{\boldsymbol{\beta}}$ is normally distributed.

Tests of Effects Based on Expected Mean Squares

Statistical tests in analysis of variance models can be constructed by comparing independent mean squares. To test a particular null hypothesis, you compute the ratio of two mean squares that have the same expected value under that hypothesis; if the ratio is much larger than 1, then that constitutes significant evidence against the null. In particular, in an analysis of variance model with fixed effects only, the expected value of each mean square has two components: quadratic functions of fixed parameters and random variation. For example, for a fixed effect called A, the expected value of its mean square is

$$\mathrm{E}[\mathrm{MS(A)}] = \mathrm{Q}(\boldsymbol{\beta}) + \sigma^2$$

where σ^2 is the common variance of the ϵ_i.

Under the null hypothesis of no A effect, the fixed portion Q($\boldsymbol{\beta}$) of the expected mean square is zero. This mean square is then compared to another mean square—say, MS(E)—that is independent of the first and has the expected value σ^2. The ratio of the two mean squares

$$F = \frac{\text{MS(A)}}{\text{MS(E)}}$$

has an F distribution under the null hypothesis.

When the null hypothesis is false, the numerator term has a larger expected value, but the expected value of the denominator remains the same. Thus, large F values lead to rejection of the null hypothesis. The probability of getting an F value at least as large as the one observed given that the null hypothesis is true is called the *significance probability value* (or the p-value). A p-value of less than 0.05, for example, indicates that data with *no* A effect will yield F values as large as the one observed less than 5% of the time. This is usually considered moderate evidence that there *is* a real A effect. Smaller p-values constitute even stronger evidence. Larger p-values indicate that the effect of interest is less than random noise. In this case, you can conclude either that there is no effect at all or that you do not have enough data to detect the differences being tested.

The actual pattern in expected mean squares of terms related to fixed quantities (Q($\boldsymbol{\beta}$)) and functions of variance components depends on which terms in your model are fixed effects and which terms are random effects. This has bearing on how F statistics can be constructed. In some instances, exact tests are not available, such as when a linear combination of expected mean squares is necessary to form a proper denominator for an F test and a Satterthwaite approximation is used to determine the degrees of freedom of the approximation. The GLM and MIXED procedures can generate tables of expected mean squares and compute degrees of freedom by Satterthwaite's method. The MIXED and GLIMMIX procedures can apply Satterthwaite approximations and other degrees-of-freedom computations more widely than in analysis of variance models. See the section "Fixed, Random, and Mixed Models" on page 35 in Chapter 3, "Introduction to Statistical Modeling with SAS/STAT Software," for a discussion of fixed versus random effects in statistical models.

Analysis of Variance for Fixed-Effect Models

PROC GLM for General Linear Models

The GLM procedure is the flagship tool for classical analysis of variance in SAS/STAT software. It performs analysis of variance by using least squares regression to fit general linear models. Among the statistical methods available in PROC GLM are regression, analysis of variance, analysis of covariance, multivariate analysis of variance, repeated measures analysis, and partial correlation analysis.

While PROC GLM can handle most common analysis of variance problems, other procedures are more efficient or have more features than PROC GLM for certain specialized analyses, or they can handle specialized models that PROC GLM cannot. Much of the rest of this chapter is concerned with comparing PROC GLM to other procedures.

PROC ANOVA for Balanced Designs

When you design an experiment, you choose how many experimental units to assign to each combination of levels (or cells) in the classification. In order to achieve good statistical properties and simplify the computations, you typically attempt to assign the same number of units to every cell in the design. Such designs are called *balanced designs*.

In SAS/STAT software, you can use the ANOVA procedure to perform analysis of variance for balanced data. The ANOVA procedure performs computations for analysis of variance that assume the balanced nature of the data. These computations are simpler and more efficient than the corresponding general computations performed by PROC GLM. Note that PROC ANOVA can be applied to certain designs that are not balanced in the strict sense of equal numbers of observations for all cells. These additional designs include all one-way models, regardless of how unbalanced the cell counts are, as well as Latin squares, which do not have data in all cells. In general, however, the ANOVA procedure is recommended only for balanced data. **If you use ANOVA to analyze a design that is not balanced, you must assume responsibility for the validity of the output.** You are responsible for recognizing incorrect results, which might include negative values reported for the sums of squares. If you are not certain that your data fit into a balanced design, then you probably need the framework of general linear models in the GLM procedure.

Comparing Group Means

The *F* test for a classification factor that has more than two levels tells you whether the level effects are significantly different from each other, but it does not tell you which levels differ from which other levels.

If the level comparisons are expressed through differences of the arithmetic cell means, you can use the MEANS statement in the GLM and ANOVA procedure for comparison. If arithmetic means are not appropriate for comparison, for example, because your data are unbalanced or means need to be adjusted for other model effects, then you can use the LSMEANS statement in the GLIMMIX, GLM, and MIXED procedures for level comparisons.

If you have specific comparisons in mind, you can use the CONTRAST statement in these procedures to make these comparisons. However, if you make many comparisons that use some given significance level (0.05, for example), you are more likely to make a type 1 error (incorrectly rejecting a hypothesis that the means are equal) simply because you have more chances to make the error.

Multiple-comparison methods give you more detailed information about the differences among the means and enable you to control error rates for a multitude of comparisons. A variety of multiple-comparison methods are available with the MEANS statement in both the ANOVA and GLM procedures, as well as the LSMEANS statement in the GLIMMIX, GLM, and MIXED procedures. These are described in detail in the section "Multiple Comparisons" on page 2514 in Chapter 39, "The GLM Procedure," and in Chapter 38, "The GLIMMIX Procedure," and Chapter 56, "The MIXED Procedure."

PROC TTEST for Comparing Two Groups

If you want to perform an analysis of variance and have only one classification variable with two levels, you can use PROC TTEST. In this special case, the results generated by PROC TTEST are equivalent to the results generated by PROC ANOVA or PROC GLM.

You can use PROC TTEST with balanced or unbalanced groups. In addition to the test assuming equal variances, PROC TTEST also performs a Satterthwaite test assuming unequal variances.

The TTEST procedure also performs equivalence tests, computes confidence limits, and supports both normal and lognormal data. If you have an AB/BA crossover design with no carryover effects, then you can use the TTEST procedure to analyze the treatment and period effects.

The PROC NPAR1WAY procedure performs nonparametric analogues to t tests. See Chapter 16, "Introduction to Nonparametric Analysis," for an overview and Chapter 62, "The NPAR1WAY Procedure," for details on PROC NPAR1WAY.

Analysis of Variance for Categorical Data and Generalized Linear Models

A *categorical variable* is defined as one that can assume only a limited number of values. For example, a person's gender is a categorical variable that can assume one of two values. Variables with levels that simply name a group are said to be measured on a *nominal scale*. Categorical variables can also be measured using an *ordinal scale*, which means that the levels of the variable are ordered in some way. For example, responses to an opinion poll are usually measured on an ordinal scale, with levels ranging from "strongly disagree" to "no opinion" to "strongly agree."

For two categorical variables, one measured on an ordinal scale and one measured on a nominal scale, you can assign scores to the levels of the ordinal variable and test whether the mean scores for the different levels of the nominal variable are significantly different. This process is analogous to performing an analysis of variance on continuous data, which can be performed by PROC CATMOD. If there are n nominal variables, rather than 1, then PROC CATMOD can perform an n-way analysis of variance of the mean scores.

For two categorical variables measured on a nominal scale, you can test whether the distribution of the first variable is significantly different for the levels of the second variable. This process is an analysis of variance of proportions, rather than means, and can be performed by PROC CATMOD. The corresponding n-way analysis of variance can also be performed by PROC CATMOD.

See Chapter 8, "Introduction to Categorical Data Analysis Procedures," and Chapter 28, "The CATMOD Procedure," for more information.

The GENMOD procedure uses maximum likelihood estimation to fit generalized linear models. This family includes models for categorical data such as logistic, probit, and complementary log-log regression for binomial data and Poisson regression for count data, as well as continuous models

such as ordinary linear regression, gamma, and inverse gaussian regression models. PROC GENMOD performs analysis of variance through likelihood ratio and Wald tests of fixed effects in generalized linear models, and provides contrasts and estimates for customized hypothesis tests. It performs analysis of repeated measures data with generalized estimating equation (GEE) methods.

See Chapter 8, "Introduction to Categorical Data Analysis Procedures," and Chapter 37, "The GENMOD Procedure," for more information.

Nonparametric Analysis of Variance

Analysis of variance is sensitive to the distribution of the error term. If the error term is not normally distributed, the statistics based on normality can be misleading. The traditional test statistics are called *parametric tests* because they depend on the specification of a certain probability distribution except for a set of free parameters. Parametric tests are said to depend on distributional assumptions. Nonparametric methods perform the tests without making any strict distributional assumptions. Even if the data are distributed normally, nonparametric methods are often almost as powerful as parametric methods.

Most nonparametric methods are based on taking the ranks of a variable and analyzing these ranks (or transformations of them) instead of the original values. The NPAR1WAY procedure performs a nonparametric one-way analysis of variance. Other nonparametric tests can be performed by taking ranks of the data (using the RANK procedure) and using a regular parametric procedure (such as GLM or ANOVA) to perform the analysis. Some of these techniques are outlined in the description of PROC RANK in *SAS Language Reference: Concepts* and in Conover and Iman (1981).

Constructing Analysis of Variance Designs

Analysis of variance is most often used for data from designed experiments. You can use the PLAN procedure to construct designs for many experiments. For example, PROC PLAN constructs designs for completely randomized experiments, randomized blocks, Latin squares, factorial experiments, certain balanced incomplete block designs, and balanced crossover designs.

Randomization, or randomly assigning experimental units to cells in a design and to treatments within a cell, is another important aspect of experimental design. For either a new or an existing design, you can use PROC PLAN to randomize the experimental plan.

Additional features for design of experiments are available in SAS/QC software. The FACTEX and OPTEX procedures can construct a wide variety of designs, including factorials, fractional factorials, and D-optimal or A-optimal designs. These procedures, as well as the ADX Interface, provide features for randomizing and replicating designs; saving the design in an output data set; and interactively changing the design by changing its size, use of blocking, or the search strategies used. For more information, see the *SAS/QC User's Guide*.

References

Analysis of variance was pioneered by R. A. Fisher (1925). For a general introduction to analysis of variance, see an intermediate statistical methods textbook such as Steel and Torrie (1980), Snedecor and Cochran (1980), Milliken and Johnson (1984), Mendenhall (1968), John (1971), Ott (1977), or Kirk (1968). A classic source is Scheffé (1959). Freund, Littell, and Spector (1991) bring together a treatment of these statistical methods and SAS/STAT software procedures. Schlotzhauer and Littell (1997) cover how to perform *t* tests and one-way analysis of variance with SAS/STAT procedures. Texts on linear models include Searle (1971), Graybill (1976), and Hocking (1984). Kennedy and Gentle (1980) survey the computing aspects. Other references include the following:

Conover, W. J. and Iman, R. L. (1981), "Rank Transformations as a Bridge between Parametric and Nonparametric Statistics," *The American Statistician*, 35, 124–129.

Fisher, R. A. (1925), *Statistical Methods for Research Workers*, Edinburgh: Oliver & Boyd.

Freund, R. J., Littell, R. C., and Spector, P. C. (1991), *SAS System for Linear Models*, Cary, NC: SAS Institute Inc.

Graybill, F. A. (1976), *Theory and Applications of the Linear Model*, North Scituate, MA: Duxbury Press.

Hocking, R. R. (1984), *Analysis of Linear Models*, Monterey, CA: Brooks-Cole.

John, P. (1971), *Statistical Design and Analysis of Experiments*, New York: Macmillan.

Kennedy, W. J., Jr. and Gentle, J. E. (1980), *Statistical Computing*, New York: Marcel Dekker.

Kirk, R. E. (1968), *Experimental Design: Procedures for the Behavioral Sciences*, Monterey, CA: Brooks-Cole.

Mendenhall, W. (1968), *Introduction to Linear Models and the Design and Analysis of Experiments*, Belmont, CA: Duxbury Press.

Milliken, G. A. and Johnson, D. E. (1984), *Analysis of Messy Data Volume I: Designed Experiments*, Belmont, CA: Lifetime Learning Publications.

Ott, L. (1977), *Introduction to Statistical Methods and Data Analysis*, Second Edition, Belmont, CA: Duxbury Press.

Scheffé, H. (1959), *The Analysis of Variance*, New York: John Wiley & Sons.

Schlotzhauer, S. D. and Littell, R. C. (1997), *SAS System for Elementary Statistical Analysis*, Cary, NC: SAS Institute Inc.

Searle, S. R. (1971), *Linear Models*, New York: John Wiley & Sons.

Snedecor, G. W. and Cochran, W. G. (1980), *Statistical Methods*, Seventh Edition, Ames: Iowa State University Press.

Steel, R. G. D. and Torrie, J. H. (1980), *Principles and Procedures of Statistics*, Second Edition, New York: McGraw-Hill.

Chapter 6

Introduction to Mixed Modeling Procedures

Contents

Overview: Mixed Modeling Procedures **127**
Types of Mixed Models **129**
Linear, Generalized Linear, and Nonlinear Mixed Models 129
Linear Mixed Model 129
Generalized Linear Mixed Model 130
Nonlinear Mixed Model 132
Models for Clustered and Hierarchical Data 132
Models with Subjects and Groups 134
Linear Mixed Models **135**
Comparing the MIXED and GLM Procedures 135
Comparing the MIXED and HPMIXED Procedures 136
Generalized Linear Mixed Models **137**
Comparing the GENMOD and GLIMMIX Procedures 138
Nonlinear Mixed Models: The NLMIXED Procedure **138**
References **139**

Overview: Mixed Modeling Procedures

A mixed model is a model that contains fixed and random effects. Since all statistical models contain some stochastic component and many models contain a residual error term, the preceding sentence deserves some clarification. The classical linear model $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$ contains the parameters $\boldsymbol{\beta}$ and the random vector $\boldsymbol{\epsilon}$. The vector $\boldsymbol{\beta}$ is a vector of fixed-effects parameters; its elements are unknown constants to be estimated from the data. A mixed model in the narrow sense also contains random effects, which are unobservable random variables. If the vector of random effects is denoted by $\boldsymbol{\gamma}$, then a linear mixed model can be written as

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

In a broader sense, mixed modeling and mixed model software is applied to special cases and generalizations of this model. For example, a purely random effects model, $\mathbf{Y} = \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$, or a correlated-error model, $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, is subsumed by mixed modeling methodology.

Over the last few decades virtually every form of classical statistical model has been enhanced to accommodate random effecs. The linear model has been extended to the linear mixed model, generalized linear models have been extended to generalized linear mixed models, and so on. In parallel with this trend, SAS/STAT software offers a number of classical and contemporary mixed modeling tools. The aim of this chapter is to provide a brief introduction and comparison of the procedures for mixed model analysis (in the broad sense) in SAS/STAT software. The theory and application of mixed models are discussed at length in many monographs, including Milliken and Johnson (1992), Diggle, Liang, and Zeger (1994), Davidian and Giltinan (1995), Verbeke and Molenberghs (1997, 2000), Vonesh and Chinchilli (1997), Demidenko (2004), Molenberghs and Verbeke (2005), and Littell et al. (2006).

The following procedures in SAS/STAT software can perform mixed and random effects analysis to various degrees:

GLM — is primarily a tool for fitting linear models by least squares. The GLM procedure has some capabilities for including random effects in a statistical model and for performing statistical tests in mixed models. Repeated measures analysis is also possible with the GLM procedure, assuming unstructured covariance modeling. Estimation methods for covariance parameters in PROC GLM are based on the method of moments, and a portion of its output applies only to the fixed-effects model.

GLIMMIX — fits generalized linear mixed models by likelihood-based techniques. As in the MIXED procedure, covariance structures are modeled parametrically. The GLIMMIX procedure also has built-in capabilities for mixed model smoothing and joint modeling of heterocatanomic multivariate data.

HPMIXED — fits linear mixed models with simple covariance structures by sparse-matrix techniques. The experimental HPMIXED procedure is designed to handle large mixed model problems, such as the solution of mixed model equations with thousands of fixed-effects parameters and random-effects solutions.

LATTICE — computes the analysis of variance and analysis of simple covariance for data from an experiment with a lattice design. PROC LATTICE analyzes balanced square lattices, partially balanced square lattices, and some rectangular lattices. Analyses performed with the LATTICE procedure can also be performed as mixed models for complete or incomplete block designs with the MIXED procedure.

MIXED — performs mixed model analysis and repeated measures analysis by way of structured covariance models. The MIXED procedure estimates parameters by likelihood or moment-based techniques. You can compute mixed model diagnostics and influence analysis for observations and groups of observations. The default fitting method maximizes the restricted likelihood of the data under the assumption that the data are normally distributed and any missing data are missing at random. This general framework accommodates many common correlated-data methods, including variance component models and repeated measures analyses.

NESTED — performs analysis of variance and analysis of covariance for purely nested random-effects models. Because of its customized algorithms, PROC NESTED can be useful for large data sets with nested random effects.

NLMIXED fits mixed models in which the fixed or random effects enter nonlinearly. The NLMIXED procedure requires that you specify components of your mixed model via programming statements. Some built-in distributions enable you to easily specify the conditional distribution of the data, given the random effects.

VARCOMP estimates variance components for random or mixed models.

The focus in the remainder of this chapter is on procedures designed for random effects and mixed model analysis: the GLIMMIX, HPMIXED, MIXED, NESTED, NLMIXED, and VARCOMP procedures. The important distinction between fixed and random effects in statistical models is addressed in the section "Fixed, Random, and Mixed Models" on page 35, in Chapter 3, "Introduction to Statistical Modeling with SAS/STAT Software."

Types of Mixed Models

Linear, Generalized Linear, and Nonlinear Mixed Models

The linear model shown at the beginning of this chapter was incomplete because the distributional properties of the random variables and their relationship were not specified. In this section the specification of the models is completed and the three model classes, linear mixed models (LMM), generalized linear mixed models (GLMM), and nonlinear mixed models (NLMM), are delineated.

Linear Mixed Model

It is a defining characteristic of the class of linear mixed models (LMM), the class of generalized linear mixed models (GLMM), and the class of nonlinear mixed models (NLMM) that the random effects are normally distributed. In the linear mixed model, this also applies to the error term; furthermore, the errors and random effects are uncorrelated. The standard linear mixed model (LMM) is thus represented by the following assumptions:

$$\begin{aligned} \mathbf{Y} &= \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon} \\ \boldsymbol{\gamma} &\sim N(\mathbf{0}, \mathbf{G}) \\ \boldsymbol{\epsilon} &\sim N(\mathbf{0}, \mathbf{R}) \\ \mathrm{Cov}[\boldsymbol{\gamma}, \boldsymbol{\epsilon}] &= \mathbf{0} \end{aligned}$$

The matrices $\mathbf{G}$ and $\mathbf{R}$ are covariance matrices for the random effects and the random errors, respectively. A *G-side* random effect in a mixed model is an element of $\boldsymbol{\gamma}$, and its variance is expressed through an element in $\mathbf{G}$. An *R-side* random variable is an element of $\boldsymbol{\epsilon}$, and its variance is an element of $\mathbf{R}$. The GLIMMIX, HPMIXED, and MIXED procedures express the $\mathbf{G}$ and $\mathbf{R}$ matrix in parametric form—that is, you structure the covariance matrix, and its elements are expressed as functions of some parameters, known as the *covariance parameters* of the mixed models. The

NLMIXED procedure also parameterizes the covariance structure, but you accomplish this with programming statements rather than with predefined syntax.

Since the right side of the model equation contains multiple random variables, the stochastic properties of $\mathbf{Y}$ can be examined by conditioning on the random effects, or through the marginal distribution. Because of the linearity of the G-side random effects and the normality of the random variables, the conditional and the marginal distribution of the data are also normal with the following mean and variance matrices:

$$\begin{aligned}\mathbf{Y}|\boldsymbol{\gamma} &\sim N(\mathbf{X}\boldsymbol{\beta}+\mathbf{Z}\boldsymbol{\gamma},\mathbf{R})\\ \mathbf{Y} &\sim N(\mathbf{X}\boldsymbol{\beta},\mathbf{V})\\ \mathbf{V} &= \mathbf{Z}\mathbf{G}\mathbf{Z}'+\mathbf{R}\end{aligned}$$

Parameter estimation in linear mixed models is based on likelihood or method-of-moment techniques. The default estimation procedure in the MIXED procedure is restricted (residual) maximum likelihood, a form of likelihood estimation that accounts for the parameters in the fixed-effects structure of the model to reduce the bias in the covariance parameter estimates. Moment-based estimation of the covariance parameters is available in the MIXED procedure through the METHOD= option in the PROC MIXED statement. The moment-based estimators are associated with sums of squares, expected mean squares (EMS), and the solution of EMS equations.

Parameter estimation by likelihood-based techniques in linear mixed models maximizes the marginal (restricted) log likelihood of the data—that is, the log likelihood is formed from $\mathbf{Y} \sim N(\mathbf{X}\boldsymbol{\beta},\mathbf{V})$. This is a model for $\mathbf{Y}$ with mean $\mathbf{X}\boldsymbol{\beta}$ and covariance matrix $\mathbf{V}$, a correlated-error model. Such *marginal models* arise, for example, in the analysis of time series data, repeated measures, or spatial data, and are naturally subsumed into the linear mixed model family. Furthermore, some mixed models have an equivalent formulation as a correlated-error model, when both give rise to the same marginal mean and covariance matrix. For example, a mixed model with a single variance component is identical to a correlated-error model with compound-symmetric covariance structure, provided that the common correlation is positive.

Generalized Linear Mixed Model

In a generalized linear mixed model (GLMM) the G-side random effects are part of the linear predictor, $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta}+\mathbf{Z}\boldsymbol{\gamma}$, and the predictor is related nonlinearly to the conditional mean of the data

$$\mathrm{E}[\mathbf{Y}|\boldsymbol{\gamma}] = g^{-1}(\boldsymbol{\eta}) = g^{-1}(\mathbf{X}\boldsymbol{\beta}+\mathbf{Z}\boldsymbol{\gamma})$$

where $g^{-1}(\cdot)$ is the inverse link funtion. The conditional distribution of the data, given the random effects, is a member of the exponential family of distributions, such as the binary, binomial, Poisson, gamma, beta, or chi-square distribution. Because the normal distribution is also a member of the exponential family, the class of the linear mixed models is a subset of the generalized linear mixed models. In order to completely specify a GLMM, you need to do the following:

1. Formulate the linear predictor, including fixed and random effects.
2. Choose a link function.

3. Choose the distribution of the response, conditional on the random effects, from the exponential family.

As an example, suppose that s pairs of twins are randomly selected in a matched-pair design. One of the twins in each pair receives a treatment and the outcome variable is some binary measure. This is a study with s clusters (subjects) and each cluster is of size 2. If Y_{ij} denotes the binary response of twin $j = 1, 2$ in cluster i, then a linear predictor for this experiment could be

$$\eta_{ij} = \beta_0 + \tau x_{ij} + \gamma_i$$

where x_{ij} denotes a regressor variable that takes on the value 1 for the treated observation in each pair, and 0 otherwise. The γ_i are pair-specific random effects that model heterogeneity across sets of twins and that induce a correlation between the members of each pair. By virtue of random sampling the sets of twins, it is reasonable to assume that the γ_i are independent and have equal variance. This leads to a diagonal $\mathbf{G}$ matrix,

$$\text{Var}[\boldsymbol{\gamma}] = \text{Var}\begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \vdots \\ \gamma_s \end{bmatrix} = \begin{bmatrix} \sigma^2_\gamma & 0 & 0 & \cdots & 0 \\ 0 & \sigma^2_\gamma & 0 & \cdots & 0 \\ 0 & 0 & \sigma^2_\gamma & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \sigma^2_\gamma \end{bmatrix}$$

A common link function for binary data is the logit link, which leads in the second step of model formulation to

$$\begin{aligned} \text{E}\left[Y_{ij}|\gamma_i\right] = \mu_{ij}|\gamma_i &= \frac{1}{1+\exp\{-\eta_{ij}\}} \\ \text{logit}\left\{\frac{\mu_{ij}|\gamma_i}{1-\mu_{ij}|\gamma_i}\right\} &= \eta_{ij} \end{aligned}$$

The final step, choosing a distribution from the exponential family, is automatic in this example; only the binary distribution comes into play to model the distribution of $Y_{ij}|\gamma_i$.

As for the linear mixed model, there is a marginal model in the case of a generalized linear mixed model that results from integrating the joint distribution over the random effects. This marginal distribution is elusive for many GLMMs, and parameter estimation proceeds by either approximating the model or by approximating the marginal integral. Details of these approaches are described in the section "Generalized Linear Mixed Models Theory" on page 2218, in Chapter 38, "The GLIMMIX Procedure."

A marginal model, one that models correlation through the $\mathbf{R}$ matrix and does not involve G-side random effects, can also be formulated in the GLMM family; such models are the extension of the correlated-error models in the linear mixed model family. Because nonnormal distributions in the exponential family exhibit a functional mean-variance relationship, fully parametric estimation is not possible in such models. Instead, estimating equations are formed based on first-moment (mean) and second-moment (covariance) assumptions for the marginal data. The approaches for modeling correlated nonnormal data via generalized estimating equations (GEE) fall into this category (see, for example, Liang and Zeger 1986; Zeger and Liang 1986).

Nonlinear Mixed Model

In a nonlinear mixed model (NLMM), the fixed and/or random effects enter the conditional mean function nonlinearly. If the mean function is a general, nonlinear function, then it is customary to assume that the conditional distribution is normal, such as in modeling growth curves or pharmacokinetic response. This is not a requirement, however.

An example of a nonlinear mixed model is the following logistic growth curve model for the jth observation of the ith subject (cluster):

$$
\begin{aligned}
f(\boldsymbol{\beta}, \boldsymbol{\gamma}_i, x_{ij}) &= \frac{\beta_1 + \gamma_{i1}}{1 + \exp[-(x_{ij} - \beta_2)/(\beta_3 + \gamma_{i2})]} \\
Y_{ij} &= f(\boldsymbol{\beta}, \boldsymbol{\gamma}_i, x_{ij}) + \epsilon_{ij} \\
\begin{bmatrix} \gamma_{i1} \\ \gamma_{i2} \end{bmatrix} &\sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} \sigma_1^2 & \sigma_{12} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix} \right) \\
Y_{ij} | \gamma_{i1}, \gamma_{i2} &\sim N(0, \sigma_\epsilon^2)
\end{aligned}
$$

The inclusion of R-side covariance structures in GLMM and NLMM models is not as straightforward as in linear mixed models for the following reasons:

- The normality of the conditional distribution in the LMM enables straightforward modeling of the covariance structure because the mean structure and covariance structure are not functionally related.
- The linearity of the random effects in the LMM leads to a marginal distribution that incorporates the **R** matrix in a natural and meaningful way.

To incorporate R-side covariance structures when random effects enter nonlinearly or when the data are not normally distributed requires estimation approaches that rely on linearizations of the mixed model. Among such estimation methods are the pseudo-likelihood methods that are available with the GLIMMIX procedure. Generalized estimating equations also solve this marginal estimation problem for nonnormal data; these are available with the GENMOD procedure.

Models for Clustered and Hierarchical Data

Mixed models are often applied in situations where data are clustered, grouped, or otherwise hierarchically organized. For example, observations might be collected by randomly selecting schools in a school district, then randomly selecting classrooms within schools, followed by selecting students within the classroom. A longitudinal study might randomly select individuals and take repeatedly measurements on them. In the first example, a school is a cluster of observations, which consists of smaller clusters (classrooms) and so on. In the longitudinal example the observations for a particular individual form a cluster. Mixed models are popular analysis tools for hierarchically organized data for the following reasons:

- The selection of groups is often performed randomly, so that the associated effects are random effects.
- The data from different clusters are independent by virtue of the random selection or by assumption.
- The observations from the same cluster are often correlated, such as the repeated observations in a repeated measures or longitudinal study.
- It is often believed that there is heterogeneity in model parameters across subjects; for example, slopes and intercepts might differ across individuals in a longitudinal growth study. This heterogeneity, if due to stochastic sources, can be modeled with random effects.

A linear mixed models with clustered, hierarchical structure can be written as a special case of the general linear mixed model by introducing appropriate subscripts. For example, a mixed model with one type of clustering and s clusters can be written as

$$\mathbf{Y}_i = \mathbf{X}_i\boldsymbol{\beta} + \mathbf{Z}_i\boldsymbol{\gamma}_i + \boldsymbol{\epsilon}_i \quad i = 1, \cdots, s$$

In SAS/STAT software, the clusters are referred to as *subjects*, and the effects that define clusters in your data can be specified with the SUBJECT= option in the GLIMMIX, HPMIXED, MIXED, and NLMIXED procedures. The vector $\mathbf{Y}_i$ collects the n_i observations for the ith subject. In certain disciplines, the organization of a hierarchical model is viewed in a *bottom-up* form, where the measured observations represent the first level, these are collected into units at the second level, and so forth. In the school data example, the bottom-up approach considers a student's score as the level-1 observation, the classroom as the level-2 unit, and the school district as the level-3 unit (if these were also selected from a population of districts).

The following points are noteworthy about mixed models with SUBJECT= specification:

- A SUBJECT= option is available in the RANDOM statements of the GLIMMIX, HPMIXED, MIXED, and NLMIXED procedures and in the REPEATED statement of the MIXED procedure.
- A SUBJECT= specification is required with the NLMIXED procedure. It is not required with any other mixed modeling procedure in SAS/STAT software.
- Specifying models with subjects is usually more computationally efficient in the MIXED and GLIMMIX procedures, especially if the SUBJECT= effects are identical or contained within each other. The computational efficiency of the HPMIXED procedure is not dependent on SUBJECT= effects in the manner in which the MIXED and GLIMMIX procedures are affected.
- There is no limit to the number of SUBJECT= effects with the MIXED, HPMIXED, and GLIMMIX procedures—that is, you can achieve an arbitrary depth of the nesting.

Models with Subjects and Groups

The concept of a subject as a unit of clustering observations in a mixed model has been described in the preceding section. This concept is important for mixed modeling with the GLIMMIX, HPMIXED, MIXED, and NLMIXED procedures. Observations from two subjects are considered uncorrelated in the analysis. Observations from the same subject are potentially correlated, depending on your specification of the covariance structure. Random effects at the subject level always lead to correlation in the marginal distribution of the observations that belong to the subject.

The GLIMMIX, HPMIXED, and MIXED procedures also support the notion of a GROUP= effect in the specification of the covariance structure. Like a subject effect, a G-side group effect identifies independent random effects. In addition to a subject effect, the group effect assumes that the realizations of the random effects correspond to draws from different distributions; in other words, each level of the group effect is associated with a different set of covariance parameters. For example, the following statements in any of these procedures fit a random coefficient model with fixed intercept and slope and subject-specific random intercept and slope:

```
class id;
model y = x;
random intercept x / subject=id;
```

The interpretation of the RANDOM statement is that for each ID an independent draw is made from a bivariate normal distribution with zero mean and a diagonal covariance matrix. In the following statements (in any of these procedures) these independent draws come from different bivariate normal distributions depending on the value of the grp variable.

```
class id grp;
model y = x;
random intercept x / subject=id group=grp;
```

Adding GROUP= effects in your model increases the flexibility to model heterogeneity in the covariance parameters, but it can add numerical complexity to the estimation process.

Linear Mixed Models

You can fit linear mixed models in SAS/STAT software with the GLM, GLIMMIX, HPMIXED, LATTICE, MIXED, NESTED, and VARCOMP procedures.

The procedure specifically designed for statistical estimation in linear mixed models is the MIXED procedure. To fit the linear mixed model

$$\begin{aligned} \mathbf{Y} &= \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon} \\ \boldsymbol{\gamma} &\sim N(\mathbf{0}, \mathbf{G}) \\ \boldsymbol{\epsilon} &\sim N(\mathbf{0}, \mathbf{R}) \\ \mathrm{Cov}[\boldsymbol{\gamma}, \boldsymbol{\epsilon}] &= \mathbf{0} \end{aligned}$$

with the MIXED procedure, you specify the fixed-effects design matrix $\mathbf{X}$ in the MODEL statement, the random-effects design matrix $\mathbf{Z}$ in the RANDOM statement, the covariance matrix of the random effects $\mathbf{G}$ with options (SUBJECT=, GROUP=, TYPE=) in the RANDOM statement, and the $\mathbf{R}$ matrix in the REPEATED statement.

By default, covariance parameters are estimated by restricted (residual) maximum likelihood. In supported models, the METHOD=TYPE1, METHOD=TYPE2, and METHOD=TYPE3 options lead to method-of-moment-based estimators and analysis of variance. The MIXED procedure provides an extensive list of diagnostics for mixed models, from various residual graphics to observationwise and groupwise influence diagnostics.

The NESTED procdedure performs an analysis of variance in nested random effects models. The VARCOMP procedure can be used to estimate variance components associated with random effects in random and mixed models. The LATTICE procedure computes analysis of variance for balanced and partially balanced square lattices. You can fit the random and mixed models supported by these procedures with the MIXED procedure as well. Some specific analyses, such as the analysis of Gauge R & R studies in the VARCOMP procedure (Burdick, Borror, and Montgomery 2005), are unique to the specialized procedures.

The GLIMMIX procedure can fit most of the models that you can fit with the MIXED procedure, but it does not offer method-of-moment-based estimation and analysis of variance in the narrow sense. Also, PROC GLIMMIX does not support the same array of covariance structures as the MIXED procedure and does not support a sampling-based Bayesian analysis. An in-depth comparison of the GLIMMIX and MIXED procedures can be found in the section "Comparing the GLIMMIX and MIXED Procedures" on page 2268, in Chapter 38, "The GLIMMIX Procedure."

Comparing the MIXED and GLM Procedures

Random- and mixed-effects models can also be fit with the GLM procedure, but the *philosophy* of the two procedures is different. The following lists important differences between the GLM and MIXED procedures in fitting random and mixed models:

- The default estimation method for covariance parameters in the MIXED procedure is restricted maximum likelihood. Covariance parameters are estimated by the method of moments by solving expressions for expected mean squares.
- In the GLM procedure, fixed and random effects are listed in the MODEL statement. Only fixed effects are listed in the MODEL statement of the MIXED procedure. In the GLM procedure, random effects must be repeated in the RANDOM statement.
- You can request tests for model effects by adding the TEST option in the RANDOM statement of the GLM procedure. PROC GLM then constructs exact tests for random effects if possible and constructs approximate tests if exact tests are not possible. For details on how the GLM procedure constructs tests for random effects, see the section "Computation of Expected Mean Squares for Random Effects" on page 2540, in Chapter 39, "The GLM Procedure." Tests for fixed effects are constructed by the MIXED procedure as Wald-type *F* tests, and the degrees of freedom for these tests can be determined by a variety of methods.
- Some of the output of the GLM procedure applies only to the fixed effects part of the model, whether a RANDOM statement is specified or not.
- Variance components are independent in the GLM procedure and covariance matrices are generally unstructured. The default covariance structure for variance components in the MIXED procedure is also a variance component structure, but the procedure offers a large number of parametric structures to model covariation among random effects and observations.

Comparing the MIXED and HPMIXED Procedures

The experimental HPMIXED procedure is designed to solve large mixed model problems by using sparse matrix techniques. The largeness of a mixed model can take many forms: a large number of observations, large number of columns in the $\mathbf{X}$ matrix, a large number of random effects, or a large number of covariance parameters. The province of the HPMIXED procedure is parameter estimation, inference, and prediction in mixed models with large $\mathbf{X}$ and/or $\mathbf{Z}$ matrices, many observations, but relatively few covariance parameters.

The models that you can fit with the HPMIXED procedure and its postprocessing analyses are a subset of the models and analyses available with the MIXED procedure. With the experimental HPMIXED procedure in SAS 9.2, you can model only G-side random effects with variance component structure or an unstructured covariance matrix in a Cholesky parameterization. R-side random effects and direct modeling of their covariance structures are not supported. A high-performance computing tool has to make concessions with respect to the supported analyses to balance performance and generality.

To some extent, the generality of the MIXED procedure precludes it from serving as a high-performance computing tool for all the model-data scenarios that the procedure can potentially estimate parameters for. For example, although efficient sparse algorithms are available to estimate variance components in large mixed models, the computational configuration changes profoundly when, for example, standard error adjustments and degrees of freedom by the Kenward-Roger method are requested.

Generalized Linear Mixed Models

Generalized linear mixed models can be fit with the GLIMMIX and NLMIXED procedures in SAS/STAT software. The GLIMMIX procedure is specifically designed to fit this class of models and offers syntax very similar to the syntax of other linear modeling procedures, such as the MIXED procedure. Consider a generalized linear model with linear predictor and link function

$$\mathrm{E}[\mathbf{Y}|\boldsymbol{\gamma}] = g^{-1}(\boldsymbol{\eta}) = g^{-1}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma})$$

and distribution in the exponential family. The fixed-effects design matrix $\mathbf{X}$ is specified in the MODEL statement of the GLIMMIX procedure, and the random-effects design matrix $\mathbf{Z}$ is specified in the RANDOM statement, along with the covariance matrix of the random effects and the covariance matrix of R-side random variables. The link function and (conditional) distribution are determined by defaults or through options in the MODEL statement.

The GLIMMIX procedure can fit heterocatanomic multivariate data—that is, data that stem from different distributions. For example, one measurement taken on a patient might be a continuous, normally distributed outcome, whereas another measurement might be a binary indicator of medical history. The GLIMMIX procedure also provides capabilities for mixed model smoothing and mixed model splines.

The GLIMMIX procedure offers an extensive array of postprocessing features to produce output statistics and to perform linear inference. The ESTIMATE and LSMESTIMATE statements support multiplicity-adjusted p-values for the protection of the familywise Type-I error rate. The LSMEANS statement supports the slicing of interactions, simple effect differences, and ODS statistical graphs for group comparisons.

The default estimation technique in the GLIMMIX procedure depends on the class of models fit. For linear mixed models, the default technique is restricted maximum likelihood, as in the MIXED procedure. For generalized linear mixed models, the estimation is based on linearization methods (pseudo-likelihood) or on integral approximation by adaptive quadrature or Laplace methods.

The NLMIXED procedure facilitates the fitting of generalized linear mixed models through several built-in distributions from the exponential family (binary, binomial, gamma, negative binomial, Poisson). You have to code the linear predictor and link function with SAS programming statements and assign starting values to all parameters, including the covariance parameters. Although it is not required to specify starting values with the NLMIXED procedure, because the procedure assigns a default value of 1.0 to every parameter not explictly given a starting value, it is highly recommended that you specify good starting values. The default estimation technique of the NLMIXED procedure, an adaptive Gauss-Hermite quadrature, is also available in the GLIMMIX procedure through the METHOD=QUAD option in the PROC GLIMMIX statement. The Laplace approximation that is available in the NLMIXED procedure by setting QPOINTS=1 is available in the GLIMMIX procedure through the METHOD=LAPLACE option.

Comparing the GENMOD and GLIMMIX Procedures

The GENMOD and GLIMMIX procedures can fit generalized linear models and estimate the parameters by maximum likelihood. For multinomial data, the GENMOD procedure fits cumulative link models for ordinal data. The GLIMMIX procedure fits these models and generalized logit models for nominal data.

When data are correlated, you can use the REPEATED statement in the GENMOD procedure to fit marginal models via generalized estimating equations. A working covariance structure is assumed and the standard errors of the parameter estimates are computed according to an empirical ("sandwich") estimator that is robust to the misspecification of the covariance structure. Marginal generalized linear models for correlated data can also be fit with the GLIMMIX procedure by specifying the random effects as R-side effects. The empirical covariance estimators are available through the EMPIRICAL= option in the PROC GLIMMIX statement. The essential difference between the estimation approach taken by the GLIMMIX procedure, and generalized estimating equations, is that the latter estimate the covariance parameters by the method of moments, whereas the GLIMMIX procedure uses likelihood-based techniques.

The GENMOD procedure supports nonsingular parameterizations of classification variables through its CLASS statement. The GLIMMIX procedure supports only the standard, GLM-type singular parameterization of CLASS variables. For the differences between these parameterizations, see the section "Parameterization of Model Effects" on page 368, in Chapter 18, "Shared Concepts and Topics."

Nonlinear Mixed Models: The NLMIXED Procedure

PROC NLMIXED handles models in which the fixed or random effects enter nonlinearly. It requires that you specify a conditional distribution of the data given the random effects, with available distributions including the normal, binomial, and Poisson. You can alternatively code your own distribution with SAS programming statements. Under a normality assumption for the random effects, PROC NLMIXED performs maximum likelihood estimation via adaptive Gaussian quadrature and a dual quasi-Newton optimization algorithm. Besides standard maximum likelihood results, you can obtain empirical Bayes predictions of the random effects and estimates of arbitrary functions of the parameters with delta-method standard errors. PROC NLMIXED has a wide variety of applications, two of the most common being nonlinear growth curves and overdispersed binomial data.

References

Burdick, R. K., Borror, C. M., and Montgomery, D. C. (2005), *Design and Analysis of Gauge R& R Studies: Making Decisions with Confidence Intervals in Random and Mixed ANOVA Models*, Alexandria, VA: SIAM (Society for Industrial and Applied Mathematics).

Davidian, M. and Giltinan, D.M. (1995), *Nonlinear Models for Repeated Measurement Data,* New York: Chapman & Hall.

Demidenko, E. (2004), *Mixed Models: Theory and Applications*, New York: John Wiley.

Diggle, P. J., Liang, K.-Y., and Zeger, S. L. (1994), *Analysis of Longitudinal Data,* Oxford, UK: Oxford University Press.

Laird, N. M. and Ware, J. H. (1982), "Random-Effects Models for Longitudinal Data," *Biometrics*, 38, 963–974.

Liang, K. Y. and Zeger, S. L. (1986), "Longitudinal Data Analysis Using Generalized Linear Models," *Biometrika*, 73, 13–22.

Littell, R. C., Milliken, G. A., Stroup, W. W., Wolfinger, R. D., and Schabenberger, O. (2006), *SAS for Mixed Models,* Second Edition, Cary, NC: SAS Institute Inc.

Milliken, G. A. and Johnson, D. E. (1992), *Analysis of Messy Data, Volume 1: Designed Experiments*, New York: Chapman & Hall.

Molenberghs, G. and Verbeke, G. (2005), *Models for Discrete Longitudinal Data*, New York: Springer.

Verbeke, G. and Molenberghs, G., eds. (1997), *Linear Mixed Models in Practice: A SAS-Oriented Approach,* New York: Springer.

Verbeke, G. and Molenberghs, G. (2000), *Linear Mixed Models for Longitudinal Data,* New York: Springer.

Vonesh, E. F. and Chinchilli, V. M. (1997), *Linear and Nonlinear Models for the Analysis of Repeated Measurements*, New York: Marcel Dekker.

Zeger, S. L. and Liang, K.-Y. (1986), "Longitudinal Data Analysis for Discrete and Continuous Outcomes," *Biometrics*, 42, 121–130.

Chapter 7

Introduction to Bayesian Analysis Procedures

Contents

Overview **141**
Introduction **142**
Background in Bayesian Statistics **144**
Prior Distributions 144
Bayesian Inference 147
Bayesian Analysis: Advantages and Disadvantages 149
Markov Chain Monte Carlo Method 151
Assessing Markov Chain Convergence 156
Summary Statistics 170
A Bayesian Reading List **173**
Textbooks 173
Tutorial and Review Papers on MCMC 175
References **175**

Overview

SAS/STAT software provides Bayesian capabilities in four procedures: GENMOD, LIFEREG, MCMC, and PHREG. The GENMOD, LIFEREG, and PHREG procedures provide Bayesian analysis in addition to the standard frequentist analyses they have always performed. Thus, these procedures provide convenient access to Bayesian modeling and inference for generalized linear models, accelerated life failure models, Cox regression models, and piecewise constant baseline hazard models (also known as piecewise exponential models). The MCMC procedure is a general procedure that fits Bayesian models with arbitrary priors and likelihood functions.

This chapter provides an overview of Bayesian statistics; describes specific sampling algorithms used in these four procedures; and discusses posterior inference and convergence diagnostics computations. Sources that provide in-depth treatment of Bayesian statistics can be found at the end of this chapter, in the section "A Bayesian Reading List" on page 173. Additional chapters contain syntax, details, and examples for the individual procedures GENMOD (see Chapter 37, "The GENMOD Procedure"), LIFEREG (see Chapter 48, "The LIFEREG Procedure"), MCMC (see Chapter 52, "The MCMC Procedure"), and PHREG (see Chapter 64, "The PHREG Procedure").

Introduction

The most frequently used statistical methods are known as *frequentist* (or *classical*) methods. These methods assume that unknown parameters are fixed constants, and they define probability by using limiting relative frequencies. It follows from these assumptions that probabilities are objective and that you cannot make probabilistic statements about parameters because they are fixed. Bayesian methods offer an alternative approach; they treat parameters as random variables and define probability as "degrees of belief" (that is, the probability of an event is the degree to which you believe the event is true). It follows from these postulates that probabilities are subjective and that you can make probability statements about parameters. The term "Bayesian" comes from the prevalent usage of Bayes' theorem, which was named after the Reverend Thomas Bayes, an eighteenth century Presbyterian minister. Bayes was interested in solving the question of inverse probability: after observing a collection of events, what is the probability of one event?

Suppose you are interested in estimating θ from data $\mathbf{y} = \{y_1, \ldots, y_n\}$ by using a statistical model described by a density $p(\mathbf{y}|\theta)$. Bayesian philosophy states that θ cannot be determined exactly, and uncertainty about the parameter is expressed through probability statements and distributions. You can say that θ follows a normal distribution with mean 0 and variance 1, if it is believed that this distribution best describes the uncertainty associated with the parameter. The following steps describe the essential elements of Bayesian inference:

1. A probability distribution for θ is formulated as $\pi(\theta)$, which is known as the *prior* distribution, or just the prior. The prior distribution expresses your beliefs (for example, on the mean, the spread, the skewness, and so forth) about the parameter before you examine the data.
2. Given the observed data $\mathbf{y}$, you choose a statistical model $p(\mathbf{y}|\theta)$ to describe the distribution of $\mathbf{y}$ given θ.
3. You update your beliefs about θ by combining information from the prior distribution and the data through the calculation of the *posterior* distribution, $p(\theta|\mathbf{y})$.

The third step is carried out by using Bayes' theorem, which enables you to combine the prior distribution and the model in the following way:

$$p(\theta|\mathbf{y}) = \frac{p(\theta, \mathbf{y})}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\theta)\pi(\theta)}{p(\mathbf{y})} = \frac{p(\mathbf{y}|\theta)\pi(\theta)}{\int p(\mathbf{y}|\theta)\pi(\theta)d\theta}$$

The quantity

$$p(\mathbf{y}) = \int p(\mathbf{y}|\theta)\pi(\theta)d\theta$$

is the normalizing constant of the posterior distribution. This quantity $p(\mathbf{y})$ is also the marginal distribution of $\mathbf{y}$, and it is sometimes called the marginal distribution of the data. The likelihood function of θ is any function proportional to $p(\mathbf{y}|\theta)$; that is, $L(\theta) \propto p(\mathbf{y}|\theta)$. Another way of writing Bayes' theorem is as follows:

$$p(\theta|\mathbf{y}) = \frac{L(\theta)\pi(\theta)}{\int L(\theta)\pi(\theta)d\theta}$$

The marginal distribution $p(\mathbf{y})$ is an integral. As long as the integral is finite, the particular value of the integral does not provide any additional information about the posterior distribution. Hence, $p(\theta|\mathbf{y})$ can be written up to an arbitrary constant, presented here in proportional form as:

$$p(\theta|\mathbf{y}) \propto L(\theta)\pi(\theta)$$

Simply put, Bayes' theorem tells you how to update existing knowledge with new information. You begin with a prior belief $\pi(\theta)$, and after learning information from data $\mathbf{y}$, you change or update your belief about θ and obtain $p(\theta|\mathbf{y})$. These are the essential elements of the Bayesian approach to data analysis.

In theory, Bayesian methods offer simple alternatives to statistical inference—all inferences follow from the posterior distribution $p(\theta|\mathbf{y})$. In practice, however, you can obtain the posterior distribution with straightforward analytical solutions only in the most rudimentary problems. Most Bayesian analyses require sophisticated computations, including the use of simulation methods. You generate samples from the posterior distribution and use these samples to estimate the quantities of interest. PROC MCMC uses a self-tuning Metropolis algorithm (see the section "Metropolis and Metropolis-Hastings Algorithms" on page 152). The GENMOD, LIFEREG, and PHREG procedures use the Gibbs sampler (see the section "Gibbs Sampler" on page 154). An important aspect of any analysis is assessing the convergence of the Markov chains. Inferences based on nonconverged Markov chains can be both inaccurate and misleading.

Both Bayesian and classical methods have their advantages and disadvantages. From a practical point of view, your choice of method depends on what you want to accomplish with your data analysis. If you have prior information (either expert opinion or historical knowledge) that you want to incorporate into the analysis, then you should consider Bayesian methods. In addition, if you want to communicate your findings in terms of probability notions that can be more easily understood by nonstatisticians, Bayesian methods might be appropriate. The Bayesian paradigm can often provide a framework for answering specific scientific questions that a single point estimate cannot sufficiently address. Alternatively, if you are interested only in estimating parameters based on the likelihood, then numerical optimization methods, such as the Newton-Raphson method, can

give you very precise estimates and there is no need to use a Bayesian analysis. For further discussions of the relative advantages and disadvantages of Bayesian analysis, see the section "Bayesian Analysis: Advantages and Disadvantages" on page 149.

Background in Bayesian Statistics

Prior Distributions

A prior distribution of a parameter is the probability distribution that represents your uncertainty about the parameter before the current data are examined. Multiplying the prior distribution and the likelihood function together leads to the posterior distribution of the parameter. You use the posterior distribution to carry out all inferences. You cannot carry out any Bayesian inference or perform any modeling without using a prior distribution.

Objective Priors versus Subjective Priors

Bayesian probability measures the degree of belief that you have in a random event. By this definition, probability is highly subjective. It follows that all priors are *subjective priors*. Not everyone agrees with this notion of subjectivity when it comes to specifying prior distributions. There has long been a desire to obtain results that are objectively valid. Within the Bayesian paradigm, this can be somewhat achieved by using prior distributions that are "objective" (that is, that have a minimal impact on the posterior distribution). Such distributions are called *objective* or *noninformative* priors (see the next section). However, while noninformative priors are very popular in some applications, they are not always easy to construct. See DeGroot and Schervish (2002, Section 1.2) and Press (2003, Section 2.2) for more information about interpretations of probability. See Berger (2006) and Goldstein (2006) for discussions about objective Bayesian versus subjective Bayesian analysis.

Noninformative Priors

Roughly speaking, a prior distribution is noninformative if the prior is "flat" relative to the likelihood function. Thus, a prior $\pi(\theta)$ is noninformative if it has minimal impact on the posterior distribution of θ. Other names for the noninformative prior are *vague*, *diffuse*, and *flat* prior. Many statisticians favor noninformative priors because they appear to be more objective. However, it is unrealistic to expect that noninformative priors represent total ignorance about the parameter of interest. In some cases, noninformative priors can lead to *improper posteriors* (nonintegrable posterior density). You cannot make inferences with improper posterior distributions. In addition, noninformative priors are often not invariant under transformation; that is, a prior might be noninformative in one parameterization but not necessarily noninformative if a transformation is applied. A common choice for a noninformative prior is the flat prior, which is a prior distribution that assigns equal likelihood on all possible values of the parameter. Intuitively this makes sense, and in some cases,

such as linear regression, flat priors on the regression parameter are noninformative. However, this is not necessarily true in all cases. For example, suppose there is a binomial experiment with n Bernoulli trials where y 1s are observed. You want to make inferences about the unknown success probability p. A uniform prior on p,

$$\pi(p) \propto 1$$

might appear to be noninformative. However, using the uniform prior is actually equivalent to adding two observations to the data, one 1 and one 0. With small n and y, the added observations can be very influential to the parameter estimate of p.

To see this, note that the likelihood is this:

$$p^y(1-p)^{n-y}$$

The maximum likelihood estimator (MLE) of p is y/n. The uniform prior can be written as a beta distribution with both the shape (α) and scale (β) parameters being 1:

$$\pi(p) \propto p^{\alpha-1}(1-p)^{\beta-1}$$

The posterior distribution of p is proportional to the following:

$$p^{\alpha+y-1}(1-p)^{\beta+n-y-1}$$

which is beta($\alpha + y, \beta + n - y$). Therefore, the posterior mean is this:

$$\frac{\alpha + y}{\alpha + \beta + n} = \frac{1 + y}{2 + n}$$

and it can be quite different from the MLE if both n and y are small. See Box and Tiao (1973) for a more formal development of noninformative priors. See Kass and Wasserman (1996) for techniques for deriving noninformative priors.

Improper Priors

A prior $\pi(\theta)$ is said to be improper if

$$\int \pi(\theta) d\theta = \infty$$

For example, a uniform prior distribution on the real line, $\pi(\theta) \propto 1$, for $-\infty < \theta < \infty$, is an improper prior. Improper priors are often used in Bayesian inference since they usually yield noninformative priors and proper posterior distributions. Improper prior distributions can lead to posterior impropriety (improper posterior distribution). To determine whether a posterior distribution is proper, you need to make sure that the normalizing constant $\int p(\mathbf{y}|\theta)p(\theta)d\theta$ is finite for all $\mathbf{y}$. If an improper prior distribution leads to an improper posterior distribution, inference based on the improper posterior distribution is invalid.

The GENMOD, LIFEREG, and PHREG procedures allow the use of improper priors—that is, the flat prior on the real line—for regression coefficients. These improper priors do not lead to any improper posterior distributions in the models that these procedures fit. PROC MCMC allows the use of any prior, as long as the distribution is programmable using DATA step functions. However, the procedure does not verify whether the posterior distribution is integrable. You must ensure this yourself.

Informative Priors

An informative prior is a prior that is not dominated by the likelihood and that has an impact on the posterior distribution. If a prior distribution dominates the likelihood, it is clearly an informative prior. These types of distributions must be specified with care in actual practice. On the other hand, the proper use of prior distributions illustrates the power of the Bayesian method: information gathered from the previous study, past experience, or expert opinion can be combined with current information in a natural way. See the "Examples" sections of the GENMOD and PHREG procedure chapters for instructions about constructing informative prior distributions.

Conjugate Priors

A prior is said to be a conjugate prior for a family of distributions if the prior and posterior distributions are from the same family, which means that the form of the posterior has the same distributional form as the prior distribution. For example, if the likelihood is binomial, $y \sim \text{Bin}(n, \theta)$, a conjugate prior on θ is the beta distribution; it follows that the posterior distribution of θ is also a beta distribution. Other commonly used conjugate prior/likelihood combinations include the normal/normal, gamma/Poisson, gamma/gamma, and gamma/beta cases. The development of conjugate priors was partially driven by a desire for computational convenience—conjugacy provides a practical way to obtain the posterior distributions. The Bayesian procedures do not use conjugacy in posterior sampling.

Jeffreys' Prior

A very useful prior is Jeffreys' prior (Jeffreys 1961). It satisfies the local uniformity property: a prior that does not change much over the region in which the likelihood is significant and does not assume large values outside that range. It is based on the Fisher information matrix. Jeffreys' prior is defined as

$$\pi(\theta) \propto |I(\theta)|^{1/2}$$

where $|\ |$ denotes the determinant and $I(\theta)$ is the Fisher information matrix based on the likelihood function $p(\mathbf{y}|\theta)$:

$$I(\theta) = -E\left[\frac{\partial^2 \log p(\mathbf{y}|\theta)}{\partial \theta^2}\right]$$

Jeffreys' prior is locally uniform and hence noninformative. It provides an automated scheme for finding a noninformative prior for any parametric model $p(\mathbf{y}|\theta)$. Another appealing property of Jeffreys' prior is that it is invariant with respect to one-to-one transformations. The invariance property means that if you have a locally uniform prior on θ and $\phi(\theta)$ is a one-to-one function of θ, then $p(\phi(\theta)) = \pi(\theta) \cdot |\phi'(\theta)|^{-1}$ is a locally uniform prior for $\phi(\theta)$. This invariance principle carries through to multidimensional parameters as well. While Jeffreys' prior provides a general recipe for obtaining noninformative priors, it has some shortcomings: the prior is improper for many models, and it can lead to improper posterior in some cases; and the prior can be cumbersome to use in high dimensions. PROC GENMOD calculates Jeffreys' prior automatically for any generalized linear model. You can set it as your prior density for the coefficient parameters, and it does not lead to improper posteriors. You can construct Jeffreys' prior for a variety of statistical models in PROC MCMC. See the section "Logistic Regression Model with Jeffreys' Prior" on page 3598 for an example. PROC MCMC does not guarantee that the corresponding posterior distribution is proper, and you need to exercise extra caution in this case.

Bayesian Inference

Bayesian inference about θ is primarily based on the posterior distribution of θ. There are various ways in which you can summarize this distribution. For example, you can report your findings through point estimates. You can also use the posterior distribution to construct hypothesis tests or probability statements.

Point Estimation and Estimation Error

Classical methods often report the maximum likelihood estimator (MLE) or the method of moments estimator (MOME) of a parameter. In contrast, Bayesian approaches often use the posterior mean. The definition of the posterior mean is given by

$$E(\theta|\mathbf{y}) = \int \theta\ p(\theta|\mathbf{y})\ d\theta$$

Other commonly used posterior estimators include the posterior median, defined as

$$\theta\colon P(\theta \geq \text{median}|\mathbf{y}) = P(\text{median} \leq \theta|\mathbf{y}) = \frac{1}{2}$$

and the posterior mode, defined as the value of θ that maximizes $p(\theta|\mathbf{y})$.

The variance of the posterior density (simply referred to as the *posterior variance*) describes the uncertainty in the parameter, which is a random variable in the Bayesian paradigm. A Bayesian analysis typically uses the posterior variance, or the posterior standard deviation, to characterize the dispersion of the parameter. In multidimensional models, covariance or correlation matrices are used.

If you know the distributional form of the posterior density of interest, you can report the exact posterior point estimates. When models become too difficult to analyze analytically, you have to use simulation algorithms, such as the Markov chain Monte Carlo (MCMC) method to obtain posterior estimates (see the section "Markov Chain Monte Carlo Method" on page 151). All of the Bayesian procedures rely on MCMC to obtain all posterior estimates. Using only a finite number of samples, simulations introduce an additional level of uncertainty to the accuracy of the estimates. *Monte Carlo standard error (MCSE)*, which is the standard error of the posterior mean estimate, measures the simulation accuracy. See the section "Standard Error of the Mean Estimate" on page 170 for more information.

The posterior standard deviation and the MCSE are two completely different concepts: the posterior standard deviation describes the uncertainty in the parameter, while the MCSE describes only the uncertainty in the parameter estimate as a result of MCMC simulation. The posterior standard deviation is a function of the sample size in the data set, and the MCSE is a function of the number of iterations in the simulation.

Hypothesis Testing

Suppose you have the following null and alternative hypotheses: H_0 is $\theta \in \Theta_0$ and H_1 is $\theta \in \Theta_0^c$, where Θ_0 is a subset of the parameter space and Θ_0^c is its complement. Using the posterior distribution $\pi(\theta|\mathbf{y})$, you can compute the posterior probabilities $P(\theta \in \Theta_0|\mathbf{y})$ and $P(\theta \in \Theta_0^c|\mathbf{y})$, or the probabilities that H_0 and H_1 are true, respectively. One way to perform a Bayesian hypothesis test is to accept the null hypothesis if $P(\theta \in \Theta_0|\mathbf{y}) \geq P(\theta \in \Theta_0^c|\mathbf{y})$ and vice versa, or to accept the null hypothesis if $P(\theta \in \Theta_0|\mathbf{y})$ is greater than a predefined threshold, such as 0.75, to guard against falsely accepted null distribution.

It is more difficult to carry out a point null hypothesis test in a Bayesian analysis. A point null hypothesis is a test of $H_0\colon \theta = \theta_0$ versus $H_1\colon \theta \neq \theta_0$. If the prior distribution $\pi(\theta)$ is a continuous density, then the posterior probability of the null hypothesis being true is 0, and there is no point in carrying out the test. One alternative is to restate the null to be a small interval hypothesis: $\theta \in \Theta_0 = (\theta_0 - a, \theta_0 + a)$, where a is a very small constant. The Bayesian paradigm can deal with an interval hypothesis more easily. Another approach is to give a mixture prior distribution to θ with a positive probability of p_0 on θ_0 and the density $(1 - p_0)\pi(\theta)$ on $\theta \neq \theta_0$. This prior ensures a nonzero posterior probability on θ_0, and you can then make realistic probabilistic comparisons. For more detailed treatment of Bayesian hypothesis testing, see Berger (1985).

Interval Estimation

The Bayesian set estimates are called *credible sets*, which is also known as *credible intervals*. This is analogous to the concept of confidence intervals used in classical statistics. Given a posterior distribution $p(\theta|\mathbf{y})$, A is a credible set for θ if

$$P(\theta \in A|\mathbf{y}) = \int_A p(\theta|\mathbf{y})d\theta$$

For example, you can construct a 95% credible set for θ by finding an interval, A, over which $\int_A p(\theta|\mathbf{y}) = 0.95$.

You can construct credible sets that have equal tails. A $100(1-\alpha)\%$ equal-tail interval corresponds to the $100(\alpha/2)$th and $100(1-\alpha/2)$th percentiles of the posterior distribution. Some statisticians prefer this interval because it is invariant under transformations. Another frequently used Bayesian credible set is called the *highest posterior density* (HPD) interval.

A $100(1-\alpha)\%$ HPD interval is a region that satisfies the following two conditions:

1. The posterior probability of that region is $100(1-\alpha)\%$.
2. The minimum density of any point within that region is equal to or larger than the density of any point outside that region.

The HPD is an interval in which most of the distribution lies. Some statisticians prefer this interval because it is the smallest interval.

One major distinction between Bayesian and classical sets is their interpretation. The Bayesian probability reflects a person's subjective beliefs. Following this approach, a statistician can make the claim that θ is inside a credible interval with measurable probability. This property is appealing because it enables you to make a direct probability statement about parameters. Many people find this concept to be a more natural way of understanding a probability interval, which is also easier to explain to nonstatisticians. A confidence interval, on the other hand, enables you to make a claim that the interval covers the true parameter. The interpretation reflects the uncertainty in the sampling procedure; a confidence interval of $100(1-\alpha)\%$ asserts that, in the long run, $100(1-\alpha)\%$ of the realized confidence intervals cover the true parameter.

Bayesian Analysis: Advantages and Disadvantages

Bayesian methods and classical methods both have advantages and disadvantages, and there are some similarities. When the sample size is large, Bayesian inference often provides results for parametric models that are very similar to the results produced by frequentist methods. Some advantages to using Bayesian analysis include the following:

- It provides a natural and principled way of combining prior information with data, within a solid decision theoretical framework. You can incorporate past information about a parameter and form a prior distribution for future analysis. When new observations become available, the previous posterior distribution can be used as a prior. All inferences logically follow from Bayes' theorem.
- It provides inferences that are conditional on the data and are exact, without reliance on asymptotic approximation. Small sample inference proceeds in the same manner as if one had a large sample. Bayesian analysis also can estimate any functions of parameters directly, without using the "plug-in" method (a way to estimate functionals by plugging the estimated parameters in the functionals).
- It obeys the likelihood principle. If two distinct sampling designs yield proportional likelihood functions for θ, then all inferences about θ should be identical from these two designs. Classical inference does not in general obey the likelihood principle.
- It provides interpretable answers, such as "the true parameter θ has a probability of 0.95 of falling in a 95% credible interval."
- It provides a convenient setting for a wide range of models, such as hierarchical models and missing data problems. MCMC, along with other numerical methods, makes computations tractable for virtually all parametric models.

There are also disadvantages to using Bayesian analysis:

- It does not tell you how to select a prior. There is no correct way to choose a prior. Bayesian inferences require skills to translate subjective prior beliefs into a mathematically formulated prior. If you do not proceed with caution, you can generate misleading results.
- It can produce posterior distributions that are heavily influenced by the priors. From a practical point of view, it might sometimes be difficult to convince subject matter experts who do not agree with the validity of the chosen prior.
- It often comes with a high computational cost, especially in models with a large number of parameters. In addition, simulations provide slightly different answers unless the same random seed is used. Note that slight variations in simulation results do not contradict the early claim that Bayesian inferences are exact. The posterior distribution of a parameter is exact, given the likelihood function and the priors, while simulation-based estimates of posterior quantities can vary due to the random number generator used in the procedures.

For more in-depth treatments of the pros and cons of Bayesian analysis, see Berger (1985, Sections 4.1 and 4.12), Berger and Wolpert (1988), Bernardo and Smith (1994, with a new edition coming out), Carlin and Louis (2000, Section 1.4), Robert (2001, Chapter 11), and Wasserman (2004, Section 11.9).

The following sections provide detailed information about the Bayesian methods provided in SAS.

Markov Chain Monte Carlo Method

The Markov chain Monte Carlo (MCMC) method is a general simulation method for sampling from posterior distributions and computing posterior quantities of interest. MCMC methods sample successively from a target distribution. Each sample depends on the previous one, hence the notion of the Markov chain. A Markov chain is a sequence of random variables, $\theta^1, \theta^2, \cdots$, for which the random variable θ^t depends on all previous θs only through its immediate predecessor θ^{t-1}. You can think of a Markov chain applied to sampling as a mechanism that traverses randomly through a target distribution without having any memory of where it has been. Where it moves next is entirely dependent on where it is now.

Monte Carlo, as in Monte Carlo integration, is mainly used to approximate an expectation by using the Markov chain samples. In the simplest version

$$\int_S g(\theta)p(\theta)d\theta \cong \frac{1}{n}\sum_{t=1}^{n} g(\theta^t)$$

where $g(\cdot)$ is a function of interest and θ^t are samples from $p(\theta)$ on its support S. This approximates the expected value of $g(\theta)$. The earliest reference to MCMC simulation occurs in the physics literature. Metropolis and Ulam (1949) and Metropolis et al. (1953) describe what is known as the Metropolis algorithm (see the section "Metropolis and Metropolis-Hastings Algorithms" on page 152). The algorithm can be used to generate sequences of samples from the joint distribution of multiple variables, and it is the foundation of MCMC. Hastings (1970) generalized their work, resulting in the Metropolis-Hastings algorithm. Geman and Geman (1984) analyzed image data by using what is now called Gibbs sampling (see the section "Gibbs Sampler" on page 154). These MCMC methods first appeared in the mainstream statistical literature in Tanner and Wong (1987).

The Markov chain method has been quite successful in modern Bayesian computing. Only in the simplest Bayesian models can you recognize the analytical forms of the posterior distributions and summarize inferences directly. In moderately complex models, posterior densities are too difficult to work with directly. With the MCMC method, it is possible to generate samples from an arbitrary posterior density $p(\theta|\mathbf{y})$ and to use these samples to approximate expectations of quantities of interest. Several other aspects of the Markov chain method also contributed to its success. Most importantly, if the simulation algorithm is implemented correctly, the Markov chain is guaranteed to converge to the target distribution $p(\theta|\mathbf{y})$ under rather broad conditions, regardless of where the chain was initialized. In other words, a Markov chain is able to improve its approximation to the true distribution at each step in the simulation. Furthermore, if the chain is run for a very long time (often required), you can recover $p(\theta|\mathbf{y})$ to any precision. Also, the simulation algorithm is easily extensible to models with a large number of parameters or high complexity, although the "curse of dimensionality" often causes problems in practice.

Properties of Markov chains are discussed in Feller (1968), Breiman (1968), and Meyn and Tweedie (1993). Ross (1997) and Karlin and Taylor (1975) give a non-measure-theoretic treatment of stochastic processes, including Markov chains. For conditions that govern Markov chain convergence and rates of convergence, see Amit (1991), Applegate, Kannan, and Polson (1990), Chan (1993), Geman and Geman (1984), Liu, Wong, and Kong (1991a, b), Rosenthal (1991a, b), Tierney

(1994), and Schervish and Carlin (1992). Besag (1974) describes conditions under which a set of conditional distributions gives a unique joint distribution. Tanner (1993), Gilks, Richardson, and Spiegelhalter (1996), Chen, Shao, and Ibrahim (2000), Liu (2001), Gelman et al. (2004), Robert and Casella (2004), and Congdon (2001, 2003, 2005) provide both theoretical and applied treatments of MCMC methods. You can also see the section "A Bayesian Reading List" on page 173 for a list of books with varying levels of difficulty of treatment of the subject and its application to Bayesian statistics.

Metropolis and Metropolis-Hastings Algorithms

The Metropolis algorithm is named after its inventor, the American physicist and computer scientist Nicholas C. Metropolis. The algorithm is simple but practical, and it can be used to obtain random samples from any arbitrarily complicated target distribution of any dimension that is known up to a normalizing constant. The Bayesian procedures use a special case of the Metropolis algorithm called the Gibbs sampler to obtain posterior samplers.

Suppose you want to obtain T samples from a univariate distribution with probability density function $f(\theta|\mathbf{y})$. Suppose θ^t is the tth sample from f. To use the Metropolis algorithm, you need to have an initial value θ^0 and a symmetric *proposal* density $q(\theta^{t+1}|\theta^t)$. For the $(t+1)$th iteration, the algorithm generates a sample from $q(\cdot|\cdot)$ based on the current sample θ^t, and it makes a decision to either accept or reject the new sample. If the new sample is accepted, the algorithm repeats itself by starting at the new sample. If the new sample is rejected, the algorithm starts at the current point and repeats. The algorithm is self-repeating, so it can be carried out as long as required. In practice, you have to decide the total number of samples needed in advance and stop the sampler after that many iterations have been completed.

Suppose $q(\theta_{\text{new}}|\theta^t)$ is a symmetric distribution. The proposal distribution should be an easy distribution from which to sample, and it must be such that $q(\theta_{\text{new}}|\theta^t) = q(\theta^t|\theta_{\text{new}})$, meaning that the likelihood of jumping to θ_{new} from θ^t is the same as the likelihood of jumping back to θ^t from θ_{new}. The most common choice of the proposal distribution is the normal distribution $N(\theta^t, \sigma)$ with a fixed σ. The Metropolis algorithm can be summarized as follows:

1. Set $t = 0$. Choose a starting point θ^0. This can be an arbitrary point as long as $f(\theta^0|\mathbf{y}) > 0$.
2. Generate a new sample, θ_{new}, by using the proposal distribution $q(\cdot|\theta^t)$.
3. Calculate the following quantity:

$$r = \min\left\{\frac{f(\theta_{\text{new}}|\mathbf{y})}{f(\theta^t|\mathbf{y})}, 1\right\}$$

4. Sample u from the uniform distribution $U(0,1)$.
5. Set $\theta^{t+1} = \theta_{\text{new}}$ if $u < r$; otherwise set $\theta^{t+1} = \theta^t$.
6. Set $t = t + 1$. If $t < T$, the number of desired samples, return to step 2. Otherwise, stop.

Note that the number of iteration keeps increasing regardless of whether a proposed sample is accepted.

This algorithm defines a chain of random variates whose distribution will converge to the desired distribution $f(\theta|\mathbf{y})$, and so from some point forward, the chain of samples is a sample from the distribution of interest. In Markov chain terminology, this distribution is called the *stationary distribution* of the chain, and in Bayesian statistics, it is the posterior distribution of the model parameters. The reason that the Metropolis algorithm works is beyond the scope of this documentation, but you can find more detailed descriptions and proofs in many standard textbooks, including Roberts (1996) and Liu (2001). The random-walk Metropolis algorithm is used in PROC MCMC.

You are not limited to a symmetric random-walk proposal distribution in establishing a valid sampling algorithm. A more general form, the Metropolis-Hastings (MH) algorithm, was proposed by Hastings (1970). The MH algorithm uses an asymmetric proposal distribution: $q(\theta_{\text{new}}|\theta^t) \neq q(\theta^t|\theta_{\text{new}})$. The difference in its implementation comes in calculating the ratio of densities:

$$r = \min\left\{\frac{f(\theta_{\text{new}}|\mathbf{y})q(\theta^t|\theta_{\text{new}})}{f(\theta^t|\mathbf{y})q(\theta_{\text{new}}|\theta^t)}, 1\right\}$$

Other steps remain the same.

The extension of the Metropolis algorithm to a higher-dimensional $\boldsymbol{\theta}$ is straightforward. Suppose $\boldsymbol{\theta} = (\theta_1, \theta_2, \cdots, \theta_k)'$ is the parameter vector. To start the Metropolis algorithm, select an initial value for each θ_k and use a multivariate version of proposal distribution $q(\cdot|\cdot)$, such as a multivariate normal distribution, to select a k-dimensional new parameter. Other steps remain the same as those previously described, and this Markov chain eventually converges to the target distribution of $f(\boldsymbol{\theta}|\mathbf{y})$. Chib and Greenberg (1995) provide a useful tutorial on the algorithm.

Independence Sampler

Another type of Metropolis algorithm is the "independence" sampler. It is called the independence sampler because the proposal distribution in the algorithm does not depend on the current point as it does with the random-walk Metropolis algorithm. For this sampler to work well, you want to have a proposal distribution that mimics the target distribution and have the acceptance rate be as high as possible.

1. Set $t = 0$. Choose a starting point θ^0. This can be an arbitrary point as long as $f(\theta^0|\mathbf{y}) > 0$.
2. Generate a new sample, θ_{new}, by using the proposal distribution $q(\cdot)$. The proposal distribution does not depend on the current value of θ^t.
3. Calculate the following quantity:

$$r = \min\left\{\frac{f(\theta_{\text{new}}|\mathbf{y})/q(\theta_{\text{new}})}{f(\theta^t|\mathbf{y})/q(\theta^t)}, 1\right\}$$

4. Sample u from the uniform distribution $U(0, 1)$.

5. Set $\theta^{t+1} = \theta_{\text{new}}$ if $u < r$; otherwise set $\theta^{t+1} = \theta^t$.
6. Set $t = t + 1$. If $t < T$, the number of desired samples, return to step 2. Otherwise, stop.

A good proposal density should have thicker tails than those of the target distribution. This requirement sometimes can be difficult to satisfy especially in cases where you do not know what the target posterior distributions are like. In addition, this sampler does not produce independent samples as the name seems to imply, and sample chains from independence samplers can get stuck in the tails of the posterior distribution if the proposal distribution is not chosen carefully. The independence sampler is used in PROC MCMC.

Gibbs Sampler

The Gibbs sampler, named by Geman and Geman (1984) after the American physicist Josiah W. Gibbs, is a special case of the Metropolis sampler in which the proposal distributions exactly match the posterior conditional distributions and proposals are accepted 100% of the time. Gibbs sampling requires you to decompose the joint posterior distribution into full conditional distributions for each parameter in the model and then sample from them. The sampler can be efficient when the parameters are not highly dependent on each other and the full conditional distributions are easy to sample from. Some researchers favor this algorithm because it does not require an instrumental proposal distribution as Metropolis methods do. However, while deriving the conditional distributions can be relatively easy, it is not always possible to find an efficient way to sample from these conditional distributions.

Suppose $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)'$ is the parameter vector, $p(\mathbf{y}|\boldsymbol{\theta})$ is the likelihood, and $\pi(\boldsymbol{\theta})$ is the prior distribution. The full posterior conditional distribution of $\pi(\theta_i|\theta_j, i \neq j, \mathbf{y})$ is proportional to the joint posterior density; that is,

$$\pi(\theta_i|\theta_j, i \neq j, \mathbf{y}) \propto p(\mathbf{y}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})$$

For instance, the one-dimensional conditional distribution of θ_1 given $\theta_j = \theta_j^*, 2 \leq j \leq k$, is computed as the following:

$$\pi(\theta_1|\theta_j = \theta_j^*, 2 \leq j \leq k, \mathbf{y}) = p(\mathbf{y}|(\boldsymbol{\theta} = (\theta_1, \theta_2^*, \ldots, \theta_k^*)')\pi(\boldsymbol{\theta} = (\theta_1, \theta_2^*, \ldots, \theta_k^*)')$$

The Gibbs sampler works as follows:

1. Set $t = 0$, and choose an arbitrary initial value of $\boldsymbol{\theta}^{(0)} = \{\theta_1^{(0)}, \ldots, \theta_k^{(0)}\}$.
2. Generate each component of $\boldsymbol{\theta}$ as follows:
   - draw $\theta_1^{(t+1)}$ from $\pi(\theta_1|\theta_2^{(t)}, \ldots, \theta_k^{(t)}, \mathbf{y})$
   - draw $\theta_2^{(t+1)}$ from $\pi(\theta_2|\theta_1^{(t+1)}, \theta_3^{(t)}, \ldots, \theta_k^{(t)}, \mathbf{y})$

- ...
- draw $\theta_k^{(t+1)}$ from $\pi(\theta_k|\theta_1^{(t+1)}, \ldots, \theta_{k-1}^{(t+1)}, \mathbf{y})$

3. Set $t = t + 1$. If $t < T$, the number of desired samples, return to step 2. Otherwise, stop.

The name "Gibbs" was introduced by Geman and Geman (1984). Gelfand et al. (1990) first used Gibbs sampling to solve problems in Bayesian inference. See Casella and George (1992) for a tutorial on the sampler. The GENMOD, LIFEREG, and PHREG procedures update parameters using the Gibbs sampler.

Adaptive Rejection Sampling Algorithm

The GENMOD, LIFEREG, and PHREG procedures use the adaptive rejection sampling (ARS) algorithm to sample parameters sequentially from their univariate full conditional distributions. The ARS algorithm is a rejection algorithm that was originally proposed by Gilks and Wild (1992). Given a log-concave density (the log of the density is concave), you can construct an envelope to the density by using linear segments. You then use the linear segment envelope as a proposal density (it becomes a piecewise exponential density on the original scale and is easy to generate samplers from) in the rejection sampling. The log-concavity condition is met in some of the models fit by the procedures. For example, the posterior densities for the regression parameters in the generalized linear models are log-concave under flat priors. When this condition fails, the ARS algorithm calls for an additional Metropolis-Hasting step (Gilks, Best, and Tan 1995), and the modified algorithm becomes the adaptive rejection metropolis sampling (ARMS) algorithm. The GENMOD, LIFEREG, and PHREG procedures can recognize whether a model is log-concave and select the appropriate sampler for the problem at hand.

The GENMOD, LIFEREG, and PHREG procedures implement the ARMS algorithm based on code kindly provided by Walter R. Gilks, University of Leeds (Gilks 2003), to obtain posterior samples. For a detailed description and explanation of the algorithm, see Gilks and Wild (1992) and Gilks, Best, and Tan (1995).

Burn-in, Thinning, and Markov Chain Samples

Burn-in refers to the practice of discarding an initial portion of a Markov chain sample so that the effect of initial values on the posterior inference is minimized. For example, suppose the target distribution is $N(0, 1)$ and the Markov chain was started at the value 10^6. The chain might quickly travel to regions around 0 in a few iterations. However, including samples around the value 10^6 in the posterior mean calculation can produce substantial bias in the mean estimate. In theory, if the Markov chain is run for an infinite amount of time, the effect of the initial values decreases to zero. In practice, you do not have the luxury of infinite samples. In practice, you assume that after t iterations, the chain has reached its target distribution and you can throw away the early portion and use the good samples for posterior inference. The value of t is the burn-in number.

With some models you might experience poor mixing (or slow convergence) of the Markov chain. This can happen, for example, when parameters are highly correlated with each other. Poor mixing means that the Markov chain slowly traverses the parameter space (see the section "Visual Analysis

via Trace Plots" on page 156 for examples of poorly mixed chains) and the chain has high dependence. High sample autocorrelation can result in biased Monte Carlo standard errors. A common strategy is to *thin* the Markov chain in order to reduce sample autocorrelations. You thin a chain by keeping every kth simulated draw from each sequence. You can safely use a thinned Markov chain for posterior inference as long as the chain converges. It is important to note that thinning a Markov chain can be wasteful because you are throwing away a $\frac{k-1}{k}$ fraction of all the posterior samples generated. MacEachern and Berliner (1994) show that you always get more precise posterior estimates if the entire Markov chain is used. However, other factors, such as computer storage or plotting time, might prevent you from keeping all samples.

To use the GENMOD, LIFEREG, MCMC, and PHREG procedures, you need to determine the total number of samples to keep ahead of time. This number is not obvious and often depends on the type of inference you want to make. Mean estimates do not require nearly as many samples as small-tail percentile estimates. In most applications, you might find that keeping a few thousand iterations is sufficient for reasonably accurate posterior inference. In all four procedures, the relationship between the number of iterations requested, the number of iterations kept, and the amount of thinning is as follows:

$$\text{kept} = \left[\frac{\text{requested}}{\text{thinning}} \right]$$

where [ ] is the rounding operator.

Assessing Markov Chain Convergence

Simulation-based Bayesian inference requires using simulated draws to summarize the posterior distribution or calculate any relevant quantities of interest. You need to treat the simulation draws with care. There are usually two issues. First, you have to decide whether the Markov chain has reached its stationary, or the desired posterior, distribution. Second, you have to determine the number of iterations to keep after the Markov chain has reached stationarity. Convergence diagnostics help to resolve these issues. Note that many diagnostic tools are designed to verify a necessary but not sufficient condition for convergence. There are no conclusive tests that can tell you when the Markov chain has converged to its stationary distribution. You should proceed with caution. Also, note that you should check the convergence of *all* parameters, and not just those of interest, before proceeding to make any inference. With some models, certain parameters can appear to have very good convergence behavior, but that could be misleading due to the slow convergence of other parameters. If some of the parameters have bad mixing, you cannot get accurate posterior inference for parameters that appear to have good mixing. See Cowles and Carlin (1996) and Brooks and Roberts (1998) for discussions about convergence diagnostics.

Visual Analysis via Trace Plots

Trace plots of samples versus the simulation index can be very useful in assessing convergence. The trace tells you if the chain has not yet converged to its stationary distribution—that is, if it needs a

longer burn-in period. A trace can also tell you whether the chain is mixing well. A chain might have reached stationarity if the distribution of points is not changing as the chain progresses. The aspects of stationarity that are most recognizable from a trace plot are a relatively constant mean and variance. A chain that mixes well traverses its posterior space rapidly, and it can jump from one remote region of the posterior to another in relatively few steps. Figure 7.1 through Figure 7.4 display some typical features that you might see in trace plots. The traces are for a parameter called γ.

Figure 7.1 Essentially Perfect Trace for γ

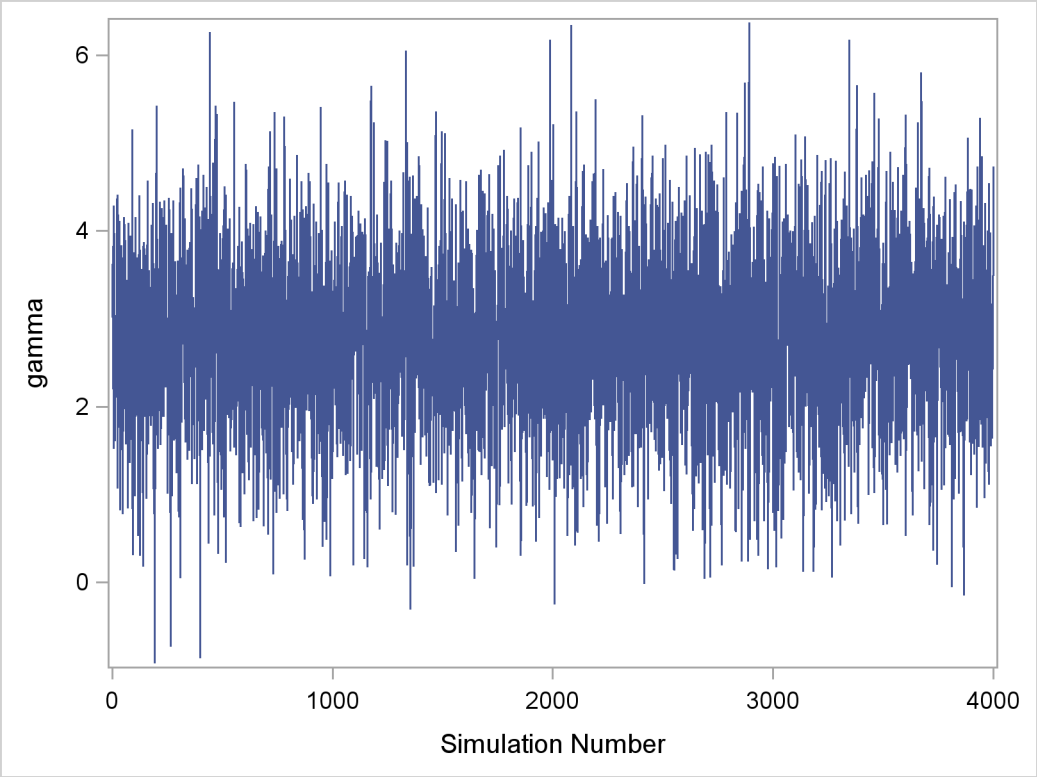


Figure 7.1 displays a "perfect" trace plot. Note that the center of the chain appears to be around the value 3, with very small fluctuations. This indicates that the chain could have reached the right distribution. The chain is mixing well; it is exploring the distribution by traversing to areas where its density is very low. You can conclude that the mixing is quite good here.

Figure 7.2 Nonconvergence of γ

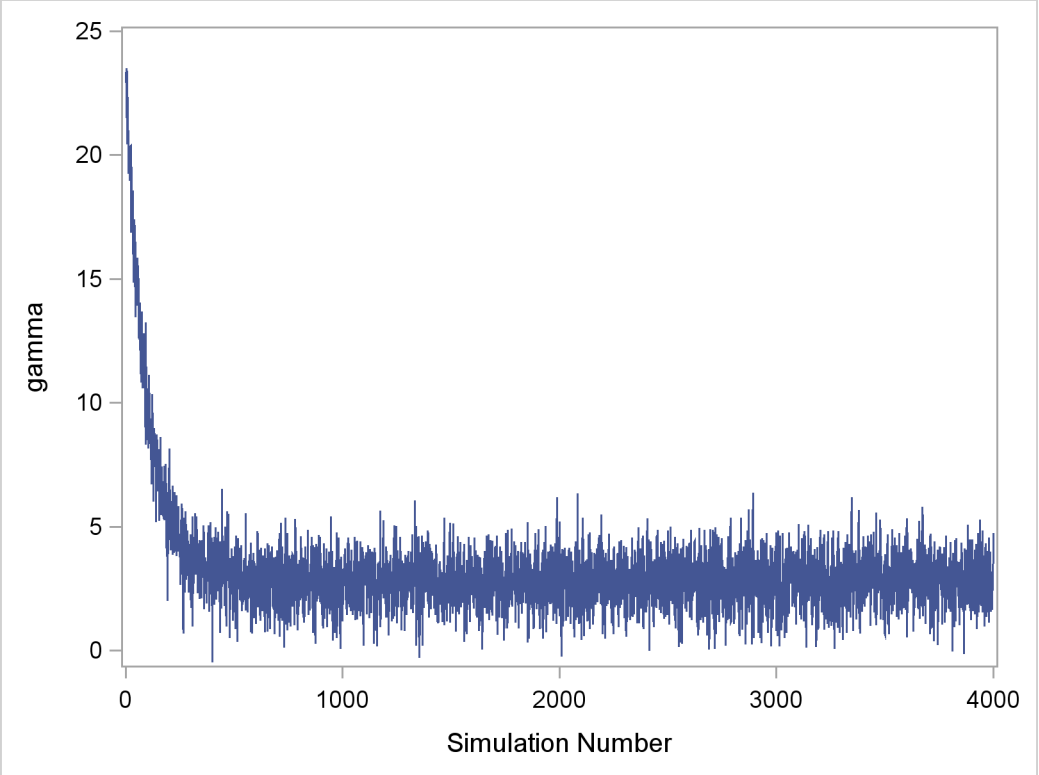


Figure 7.2 displays a trace plot for a chain that starts at a very remote initial value and makes its way to the targeting distribution. The first few hundred observations should be discarded. This chain appears to be mixing very well locally. It travels relatively quickly to the target distribution, reaching it in a few hundred iterations. If you have a chain that looks like this, you would want to increase the burn-in sample size. If you need to use this sample to make inferences, you would want to use only the samples toward the end of the chain.

Figure 7.3 Marginal Mixing for γ

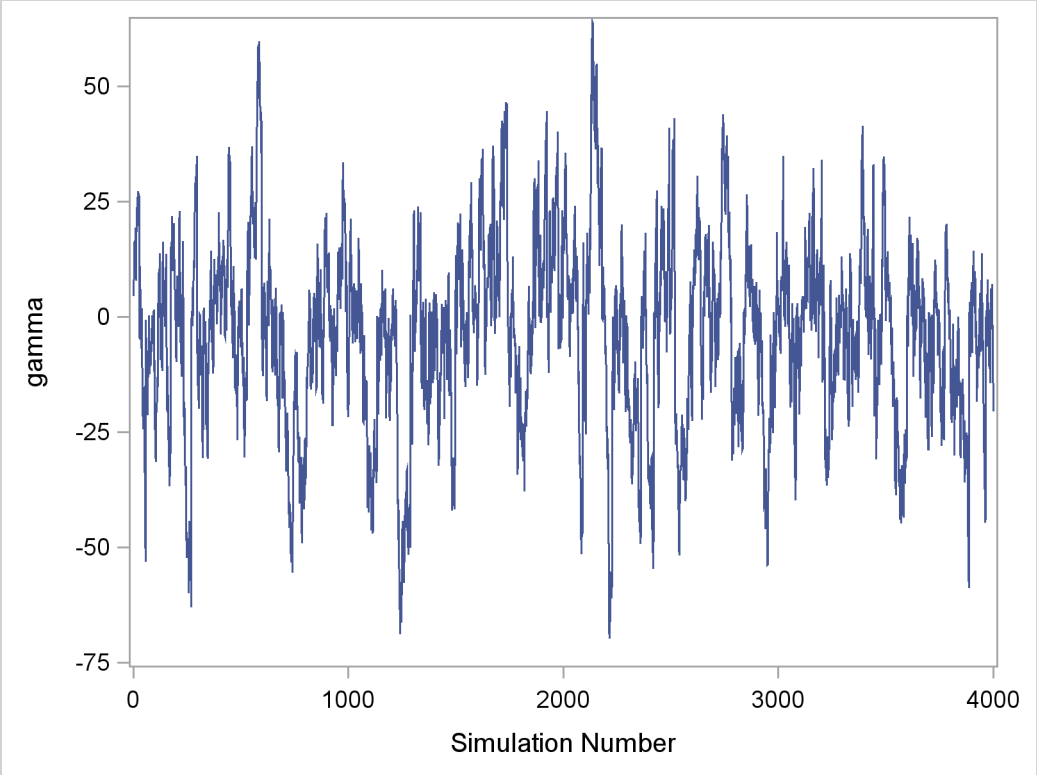
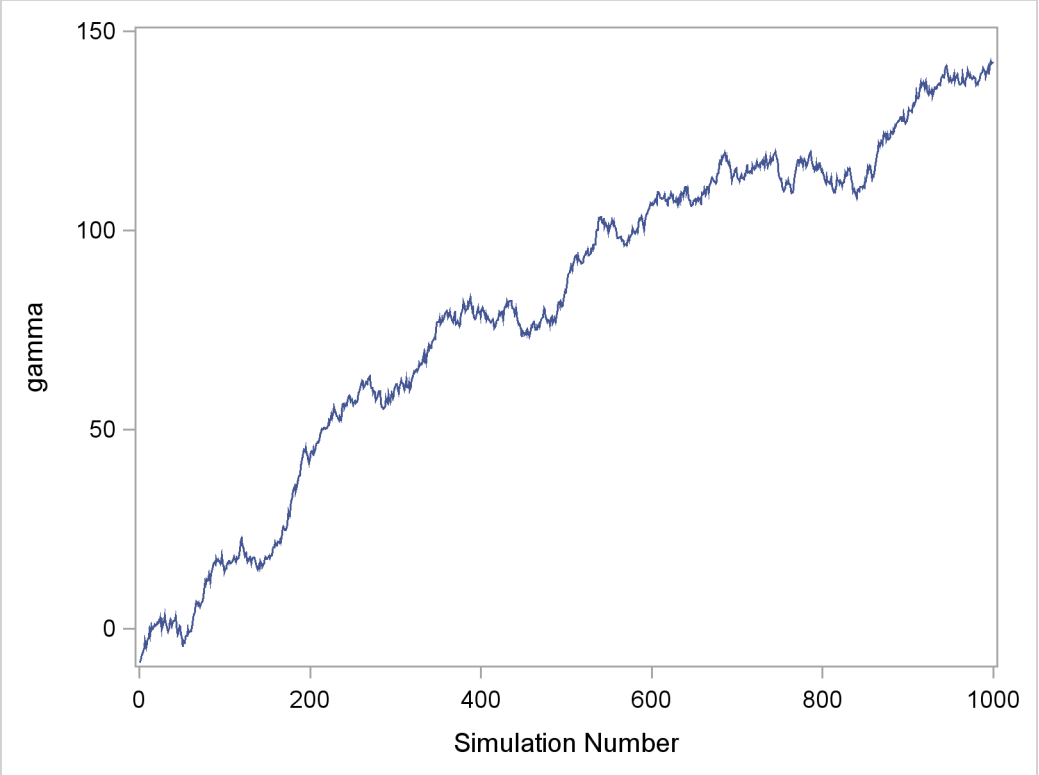


Figure 7.3 demonstrates marginal mixing. The chain is taking only small steps and does not traverse its distribution quickly. This type of trace plot is typically associated with high autocorrelation among the samples. To obtain a few thousand independent samples, you need to run the chain for much longer.

Figure 7.4 Bad Mixing, Nonconvergence of γ



The trace plot shown in Figure 7.4 depicts a chain with serious problems. It is mixing very slowly, and it offers no evidence of convergence. You would want to try to improve the mixing of this chain. For example, you might consider reparameterizing your model on the log scale. Run the Markov chain for a long time to see where it goes. This type of chain is entirely unsuitable for making parameter inferences.

Statistical Diagnostic Tests

The Bayesian procedures include several statistical diagnostic tests that can help you assess Markov chain convergence. For a detailed description of each of the diagnostic tests, see the following subsections. Table 7.1 provides a summary of the diagnostic tests and their interpretations.

Table 7.1 Convergence Diagnostic Tests Available in the Bayesian Procedures

Name	Description	Interpretation of the Test
Gelman-Rubin	Uses parallel chains with dispersed initial values to test whether they all converge to the same target distribution. Failure could indicate the presence of a multi-mode posterior distribution (different chains converge to different local modes) or the need to run a longer chain (burn-in is yet to be completed).	One-sided test based on a variance ratio test statistic. Large $\widehat{R}_c$ values indicate rejection.

Table 7.1 (continued)

Name	Description	Interpretation of the test
Geweke	Tests whether the mean estimates have converged by comparing means from the early and latter part of the Markov chain.	Two-sided test based on a z-score statistic. Large absolute z values indicate rejection.
Heidelberger-Welch (stationarity test)	Tests whether the Markov chain is a covariance (or weakly) stationary process. Failure could indicate that a longer Markov chain is needed.	One-sided test based on a Cramer–von Mises statistic. Small p-values indicate rejection.
Heidelberger-Welch (half-width test)	Reports whether the sample size is adequate to meet the required accuracy for the mean estimate. Failure could indicate that a longer Markov chain is needed.	If a relative half-width statistic is greater than a predetermined accuracy measure, this indicates rejection.
Raftery-Lewis	Evaluates the accuracy of the estimated (desired) percentiles by reporting the number of samples needed to reach the desired accuracy of the percentiles. Failure could indicate that a longer Markov chain is needed.	If the total samples needed are fewer than the Markov chain sample, this indicates rejection.
autocorrelation	Measures dependency among Markov chain samples.	High correlations between long lags indicate poor mixing.
effective sample size	Relates to autocorrelation; measures mixing of the Markov chain.	Large discrepancy between the effective sample size and the simulation sample size indicates poor mixing.

Gelman and Rubin Diagnostics

Gelman and Rubin diagnostics (Gelman and Rubin 1992; Brooks and Gelman 1997) are based on analyzing multiple simulated MCMC chains by comparing the variances within each chain and the variance between chains. Large deviation between these two variances indicates nonconvergence.

Define $\{\theta^t\}$, where $t = 1, \ldots, n$, to be the collection of a single Markov chain output. The parameter θ^t is the tth sample of the Markov chain. For notational simplicity, θ is assumed to be single dimensional in this section.

Suppose you have M parallel MCMC chains that were initialized from various parts of the target distribution. Each chain is of length n (after discarding the burn-in). For each θ^t, the simulations are labeled as θ_m^t, where $t = 1, \ldots, n$ and $m = 1, \ldots, M$. The between-chain variance B and the within-chain variance W are calculated as

$$
\begin{aligned}
B &= \frac{n}{M-1}\sum_{m=1}^{M}(\bar{\theta}_m^{\cdot}-\bar{\theta}_{\cdot}^{\cdot})^2, \quad \text{where } \bar{\theta}_m^{\cdot}=\frac{1}{n}\sum_{t=1}^{n}\theta_m^t, \quad \bar{\theta}_{\cdot}^{\cdot}=\frac{1}{M}\sum_{m=1}^{M}\bar{\theta}_m^{\cdot} \\
W &= \frac{1}{M}\sum_{m=1}^{M}s_m^2, \quad \text{where } s_m^2=\frac{1}{n-1}\sum_{t=1}^{n}(\theta_m^t-\bar{\theta}_m^{\cdot})^2
\end{aligned}
$$

The posterior marginal variance, $\text{var}(\theta|\mathbf{y})$, is a weighted average of W and B. The estimate of the variance is

$$
\widehat{V}=\frac{n-1}{n}W+\frac{M+1}{nM}B
$$

If all M chains have reached the target distribution, this posterior variance estimate should be very close to the within-chain variance W. Therefore, you would expect to see the ratio $\widehat{V}/W$ be close to 1. The square root of this ratio is referred to as the *potential scale reduction factor* (PSRF). A large PSRF indicates that the between-chain variance is substantially greater than the within-chain variance, so that longer simulation is needed. If the PSRF is close to 1, you can conclude that each of the M chains has stabilized, and they are likely to have reached the target distribution.

A refined version of PSRF is calculated, as suggested by Brooks and Gelman (1997), as

$$
\widehat{\mathrm{R}}_c=\sqrt{\frac{\hat{d}+3}{\hat{d}+1}\cdot\frac{\widehat{V}}{W}}=\sqrt{\frac{\hat{d}+3}{\hat{d}+1}\left(\frac{n-1}{n}+\frac{M+1}{nM}\frac{B}{W}\right)}
$$

where

$$
\hat{d}=\frac{2\widehat{V}^2}{\widehat{\text{Var}}(\widehat{V})}
$$

and

$$
\begin{aligned}
\widehat{\text{Var}}(\widehat{V}) &= \left(\frac{n-1}{n}\right)^2\frac{1}{M}\widehat{\text{Var}}(s_m^2)+\left(\frac{M+1}{nM}\right)^2\frac{2}{M-1}B^2 \\
&\quad +2\frac{(M+1)(n-1)}{n^2M}\frac{n}{M}\left(\widehat{\text{cov}}(s_m^2,(\bar{\theta}_m^{\cdot})^2)-2\bar{\theta}_{\cdot}^{\cdot}\widehat{\text{cov}}(s_m^2,\bar{\theta}_m^{\cdot})\right)
\end{aligned}
$$

All the Bayesian procedures also produce an upper $100(1-\alpha/2)\%$ confidence limit of $\widehat{\mathrm{R}}_c$. Gelman and Rubin (1992) showed that the ratio B/W in $\widehat{\mathrm{R}}_c$ has an F distribution with degrees of freedom

$M-1$ and $2W^2M/\widehat{\text{Var}}(s_m^2)$. Because you are concerned only if the scale is large, not small, only the upper $100(1-\alpha/2)\%$ confidence limit is reported. This is written as

$$\sqrt{\left(\frac{n-1}{n}+\frac{M+1}{nM}\cdot F_{1-\alpha/2}\left(M-1,\frac{2W^2}{\widehat{\text{Var}}(s_m^2)/M}\right)\right)\cdot\frac{\hat{d}+3}{\hat{d}+1}}$$

In the Bayesian procedures, you can specify the number of chains that you want to run. Typically three chains are sufficient. The first chain is used for posterior inference, such as mean and standard deviation; the other $M-1$ chains are used for computing the diagnostics and are discarded afterward. This test can be computationally costly, because it prolongs the simulation M-fold.

It is best to choose different initial values for all M chains. The initial values should be as dispersed from each other as possible so that the Markov chains can fully explore different parts of the distribution before they converge to the target. Similar initial values can be risky because all of the chains can get stuck in a local maximum; that is something this convergence test cannot detect. If you do not supply initial values for all the different chains, the procedures generate them for you.

Geweke Diagnostics

The Geweke test (Geweke 1992) compares values in the early part of the Markov chain to those in the latter part of the chain in order to detect failure of convergence. The statistic is constructed as follows. Two subsequences of the Markov chain $\{\theta^t\}$ are taken out, with $\{\theta_1^t : t=1,\ldots,n_1\}$ and $\{\theta_2^t : t=n_a,\ldots,n\}$, where $1<n_1<n_a<n$. Let $n_2=n-n_a+1$, and define

$$\bar{\theta}_1=\frac{1}{n_1}\sum_{t=1}^{n_1}\theta^t \quad\text{and}\quad \bar{\theta}_2=\frac{1}{n_2}\sum_{t=n_a}^{n}\theta^t$$

Let $\hat{s}_1(0)$ and $\hat{s}_2(0)$ denote consistent spectral density estimates at zero frequency (see the subsection "Spectral Density Estimate at Zero Frequency" on page 164 for estimation details) for the two MCMC chains, respectively. If the ratios n_1/n and n_2/n are fixed, $(n_1+n_2)/n<1$, and the chain is stationary, then the following statistic converges to a standard normal distribution as $n\to\infty$:

$$Z_n=\frac{\bar{\theta}_1-\bar{\theta}_2}{\sqrt{\frac{\hat{s}_1(0)}{n_1}+\frac{\hat{s}_2(0)}{n_2}}}$$

This is a two-sided test, and large absolute z-scores indicate rejection.

Spectral Density Estimate at Zero Frequency

For one sequence of the Markov chain $\{\theta_t\}$, the relationship between the h-lag covariance sequence of a time series and the spectral density, f, is

$$s_h = \frac{1}{2\pi}\int_{-\pi}^{\pi} \exp(\mathrm{i}\omega h) f(\omega) d\omega$$

where i indicates that ωh is the complex argument. Inverting this Fourier integral,

$$f(\omega) = \sum_{h=-\infty}^{\infty} s_h \exp(-\mathrm{i}\omega h) = s_0 \left(1 + 2\sum_{h=1}^{\infty} \rho_h \cos(\omega h)\right)$$

It follows that

$$f(0) = \sigma^2 \left(1 + 2\sum_{h=1}^{\infty} \rho_h\right)$$

which gives an autocorrelation adjusted estimate of the variance. In this equation, σ^2 is the naive variance estimate of the sequence $\{\theta_t\}$ and ρ_h is the lag h autocorrelation. Due to obvious computational difficulties, such as calculation of autocorrelation at infinity, you cannot effectively estimate $f(0)$ by using the preceding formula. The usual route is to first obtain the *periodogram* $p(\omega)$ of the sequence, and then estimate $f(0)$ by smoothing the estimated periodogram. The periodogram is defined to be

$$p(\omega) = \frac{1}{n}\left[\left(\sum_{t=1}^{n} \theta_t \sin(\omega t)\right)^2 + \left(\sum_{t=1}^{n} \theta_t \cos(\omega t)\right)^2\right]$$

The procedures use the following way to estimate $\hat{f}(0)$ from p (Heidelberger and Welch 1981). In $p(\omega)$, let $\omega = \omega_k = 2\pi k/n$ and $k = 1, \ldots, [\frac{n}{2}]$.[1] A smooth spectral density in the domain of $(0, \pi]$ is obtained by fitting a gamma model with the log link function, using $p(\omega_k)$ as response and $x_1(\omega_k) = \sqrt{3}(4\omega_k/(2\pi) - 1)$ as the only regressor. The predicted value $\hat{f}(0)$ is given by

$$\hat{f}(0) = \exp(\hat{\beta}_0 - \sqrt{3}\hat{\beta}_1)$$

where $\hat{\beta}_0$ and $\hat{\beta}_1$ are the estimates of the intercept and slope parameters, respectively.

[1] This is equivalent to the fast Fourier transformation of the original time series θ_t.

Heidelberger and Welch Diagnostics

The Heidelberger and Welch test (Heidelberger and Welch 1981, 1983) consists of two parts: a stationary portion test and a half-width test. The stationarity test assesses the stationarity of a Markov chain by testing the hypothesis that the chain comes from a covariance stationary process. The half-width test checks whether the Markov chain sample size is adequate to estimate the mean values accurately.

Given $\{\theta^t\}$, set $S_0 = 0$, $S_n = \sum_{t=1}^{n} \theta^t$, and $\bar{\theta} = (1/n)\sum_{t=1}^{n} \theta^t$. You can construct the following sequence with s coordinates on values from $\frac{1}{n}, \frac{2}{n}, \cdots, 1$:

$$B_n(s) = (S_{[ns]} - [ns]\bar{\theta})/(n\,\hat{p}(0))^{1/2}$$

where [ ] is the rounding operator, and $\hat{p}(0)$ is an estimate of the spectral density at zero frequency that uses the second half of the sequence (see the section "Spectral Density Estimate at Zero Frequency" on page 164 for estimation details). For large n, B_n converges in distribution to a Brownian bridge (Billingsley 1986). So you can construct a test statistic by using B_n. The statistic used in these procedures is the Cramer–von Mises statistic[2]; that is $\int_0^1 B_n(s)^2 ds = CVM(B_n)$. As $n \to \infty$, the statistic converges in distribution to a standard Cramer–von Mises distribution. The integral $\int_0^1 B_n(s)^2 ds$ is numerically approximated using Simpson's rule.

Let $y_i = B_n(s)^2$, where $s = 0, \frac{1}{n}, \cdots, \frac{n-1}{n}, 1$, and $i = ns = 0, 1, \cdots, n$. If n is even, let $m = n/2$; otherwise, let $m = (n-1)/2$. The Simpson's approximation to the integral is

$$\int_0^1 B_n(s)^2 ds \approx \frac{1}{3n}\left[y_0 + 4(y_1 + \cdots + y_{2m-1}) + 2(y_2 + \cdots + y_{2m-2}) + y_{2m}\right]$$

Note that Simpson's rule requires an even number of intervals. When n is odd, y_n is set to be 0 and the value does not contribute to the approximation.

This test can be performed repeatedly on the same chain, and it helps you identify a time t when the chain has reached stationarity. The whole chain, $\{\theta^t\}$, is first used to construct the Cramer–von Mises statistic. If it passes the test, you can conclude that the entire chain is stationary. If it fails the test, you drop the initial 10% of the chain and redo the test by using the remaining 90%. This process is repeated until either a time t is selected or it reaches a point where there are not enough data remaining to construct a confidence interval (the cutoff proportion is set to be 50%).

The part of the chain that is deemed stationary is put through a half-width test, which reports whether the sample size is adequate to meet certain accuracy requirements for the mean estimates. Running the simulation less than this length of time would not meet the requirement, while running it longer would not provide any additional information that is needed. The statistic calculated here is the *relative half-width* (RHW) of the confidence interval. The RHW for a confidence interval of level $1-\alpha$ is

[2] The von Mises distribution was first introduced by von Mises (1918). The density function is $p(\theta|\mu\kappa) \sim M(\mu,\kappa) = [2\pi I_0(\kappa)]^{-1}\exp(\kappa\cos(\theta-\mu))$ $(0 \le \theta \le 2\pi)$, where the function $I_0(\kappa)$ is the modified Bessel function of the first kind and order zero, defined by $I_0(\kappa) = (2\pi)^{-1}\int_0^{2\pi}\exp(\kappa\cos(\theta-\mu))d\theta$.

$$\text{RHW} = \frac{z_{(1-\alpha/2)} \cdot (\hat{s}_n/n)^{1/2}}{\hat{\theta}}$$

where $z_{(1-\alpha/2)}$ is the z-score of the $100(1-\alpha/2)$th percentile (for example, $z_{(1-\alpha/2)} = 1.96$ if $\alpha = 0.05$), $\hat{s}_n$ is the variance of the chain estimated using the spectral density method (see explanation in the section "Spectral Density Estimate at Zero Frequency" on page 164), n is the length, and $\hat{\theta}$ is the estimated mean. The RHW quantifies accuracy of the $1-\alpha$ level confidence interval of the mean estimate by measuring the ratio between the sample standard error of the mean and the mean itself. In other words, you can stop the Markov chain if the variability of the mean stabilizes with respect to the mean. An implicit assumption is that large means are often accompanied by large variances. If this assumption is not met, then this test can produce false rejections (such as a small mean around 0 and large standard deviation) or false acceptance (such as a very large mean with relative small variance). As with any other convergence diagnostics, you might want to exercise caution in interpreting the results.

The stationarity test is one-sided; rejection occurs when the p-value is greater than $1-\alpha$. To perform the half-width test, you need to select an α level (the default of which is 0.05) and a predetermined tolerance value ϵ (the default of which is 0.1). If the calculated RHW is greater than ϵ, you conclude that there are not enough data to accurately estimate the mean with $1-\alpha$ confidence under tolerance of ϵ.

Raftery and Lewis Diagnostics

If your interest lies in posterior percentiles, you want a diagnostic test that evaluates the accuracy of the estimated percentiles. The Raftery-Lewis test (Raftery and Lewis 1992, 1996) is designed for this purpose. Notation and deductions here closely resemble those in Raftery and Lewis (1996).

Suppose you are interested in a quantity θ_q such that $P(\theta \leq \theta_q|\mathbf{y}) = q$, where q can be an arbitrary cumulative probability, such as 0.025. This θ_q can be empirically estimated by finding the $[n \cdot 100 \cdot q]$th number of the sorted $\{\theta^t\}$. Let $\hat{\theta}_q$ denote the estimand, which corresponds to an estimated probability $P(\theta \leq \hat{\theta}_q) = \hat{P}_q$. Because the simulated posterior distribution converges to the true distribution as the simulation sample size grows, $\hat{\theta}_q$ can achieve any degree of accuracy if the simulator is run for a very long time. However, running too long a simulation can be wasteful. Alternatively, you can use coverage probability to measure accuracy and stop the chain when a certain accuracy is reached.

A stopping criterion is reached when the estimated probability is within $\pm r$ of the true cumulative probability q, with probability s, such as $P(\hat{P}_q \in (q-r, q+r)) = s$. For example, suppose you want the coverage probability s to be 0.95 and the amount of tolerance r to be 0.005. This corresponds to requiring that the estimate of the cumulative distribution function of the 2.5th percentile be estimated to within ± 0.5 percentage points with probability 0.95.

The Raftery-Lewis diagnostics test finds the number of iterations, M, that need to be discarded (burn-ins) and the number of iterations needed, N, to achieve a desired precision. Given a predefined cumulative probability q, these procedures first find $\hat{\theta}_q$, and then they construct a binary $0-1$ process $\{Z_t\}$ by setting $Z_t = 1$ if $\theta^t \leq \hat{\theta}_q$ and 0 otherwise for all t. The sequence $\{Z_t\}$ is itself not a Markov chain, but you can construct a subsequence of $\{Z_t\}$ that is approximately Markovian if it

is sufficiently k-thinned. When k becomes reasonably large, $\{Z_t^{(k)}\}$ starts to behave like a Markov chain.

Next, the procedures find this thinning parameter k. The number k is estimated by comparing the Bayesian information criterion (BIC) between two Markov models: a first-order and a second-order Markov model. A jth-order Markov model is one in which the current value of $\{Z_t^{(k)}\}$ depends on the previous j values. For example, in a second-order Markov model,

$$\begin{aligned} & p\left(Z_t^{(k)} = z_t | Z_{t-1}^{(k)} = z_{t-1}, Z_{t-2}^{(k)} = z_{t-2}, \cdots, Z_0^{(k)} = z_0\right) \\ = \quad & p\left(Z_t^{(k)} = z_t | Z_{t-1}^{(k)} = z_{t-1}, Z_{t-2}^{(k)} = z_{t-2}\right) \end{aligned}$$

where $z_i = \{0, 1\}, i = 0, \cdots, t$. Given $\{Z_t^{(k)}\}$, you can construct two transition count matrices for a second-order Markov model:

	$z_t = 0$		$z_t = 1$	
	$z_{t-1} = 0$	$z_{t-1} = 1$	$z_{t-1} = 0$	$z_{t-1} = 1$
$z_{t-2} = 0$	w_{000}	w_{010}	w_{001}	w_{011}
$z_{t-2} = 1$	w_{100}	w_{110}	w_{101}	w_{111}

For each k, the procedures calculate the BIC that compares the two Markov models. The BIC is based on a likelihood ratio test statistic that is defined as

$$G_k^2 = 2 \sum_{i=0}^{1} \sum_{j=0}^{1} \sum_{l=0}^{1} w_{ijl} \log \frac{w_{ijl}}{\hat{w}_{ijl}}$$

where $\hat{w}_{ijl}$ is the expected cell count of w_{ijl} under the null model, the first-order Markov model, where the assumption $(Z_t^{(k)} \perp Z_{t-2}^{(k)}) | Z_{t-1}^{(k)}$ holds. The formula for the expected cell count is

$$\hat{w}_{ijl} = \frac{\sum_i w_{ijl} \cdot \sum_l w_{ijl}}{\sum_i \sum_l w_{ijl}}$$

The BIC is $G_k^2 - 2\log(n_k - 2)$, where n_k is the k-thinned sample size (every kth sample starting with the first), with the last two data points discarded due to the construction of the second-order Markov model. The thinning parameter k is the smallest k for which the BIC is negative. When k is found, you can estimate a transition probability matrix between state 0 and state 1 for $\{Z_t^{(k)}\}$:

$$Q = \begin{pmatrix} 1-\alpha & \alpha \\ \beta & 1-\beta \end{pmatrix}$$

Because $\{Z_t^{(k)}\}$ is a Markov chain, its equilibrium distribution exists and is estimated by

$$\pi = (\pi_0, \pi_1) = \frac{(\beta, \alpha)}{\alpha + \beta}$$

where $\pi_0 = P(\theta \leq \theta_q | \mathbf{y})$ and $\pi_1 = 1 - \pi_0$. The goal is to find an iteration number m such that after m steps, the estimated transition probability $P(Z_m^{(k)} = i | Z_0^{(k)} = j)$ is within ϵ of equilibrium π_i for $i, j = 0, 1$. Let $e_0 = (1, 0)$ and $e_1 = 1 - e_0$. The estimated transition probability after step m is

$$P(Z_m^{(k)} = i | Z_0^{(k)} = j) = e_j \left[\begin{pmatrix} \pi_0 & \pi_1 \\ \pi_0 & \pi_1 \end{pmatrix} + \frac{(1-\alpha-\beta)^m}{\alpha+\beta} \begin{pmatrix} \alpha & -\alpha \\ -\beta & \beta \end{pmatrix} \right] e_j^\top$$

which holds when

$$m = \frac{\log\left(\frac{(\alpha+\beta)\epsilon}{\max(\alpha,\beta)}\right)}{\log(1-\alpha-\beta)}$$

assuming $1 - \alpha - \beta > 0$.

Therefore, by time m, $\{Z_t^{(k)}\}$ is sufficiently close to its equilibrium distribution, and you know that a total size of $M = mk$ should be discarded as the burn-in.

Next, the procedures estimate N, the number of simulations needed to achieve desired accuracy on percentile estimation. The estimate of $P(\theta \leq \theta_q | \mathbf{y})$ is $\bar{Z}_n^{(k)} = \frac{1}{n} \sum_{t=1}^{n} Z_t^{(k)}$. For large n, $\bar{Z}_n^{(k)}$ is normally distributed with mean q, the true cumulative probability, and variance

$$\frac{1}{n} \frac{(2-\alpha-\beta)\alpha\beta}{(\alpha+\beta)^3}$$

$P(q - r \leq \bar{Z}_n^{(k)} \leq q + r) = s$ is satisfied if

$$n = \frac{(2-\alpha-\beta)\alpha\beta}{(\alpha+\beta)^3} \left\{ \frac{\Phi^{-1}\left(\frac{s+1}{2}\right)}{r} \right\}^2$$

Therefore, $N = nk$.

By using similar reasoning, the procedures first calculate the minimal number of iterations needed to achieve the desired accuracy, assuming the samples are independent:

$$N_{min} = \left\{ \Phi^{-1} \left(\frac{s+1}{2} \right) \right\}^2 \frac{q(1-q)}{r^2}$$

If $\{\theta^t\}$ does not have that required sample size, the Raftery-Lewis test is not carried out. If you still want to carry out the test, increase the number of Markov chain iterations.

The ratio N/N_{min} is sometimes referred to as the *dependence factor*. It measures deviation from posterior sample independence: the closer it is to 1, the less correlated are the samples. There are a few things to keep in mind when you use this test. This diagnostic tool is specifically designed for the percentile of interest and does not provide information about convergence of the chain as a whole (Brooks and Roberts 1999). In addition, the test can be very sensitive to small changes. Both N and N_{min} are inversely proportional to r^2, so you can expect to see large variations in these numbers with small changes to input variables, such as the desired coverage probability or the cumulative probability of interest. Last, the time until convergence for a parameter can differ substantially for different cumulative probabilities.

Autocorrelations

The sample autocorrelation of lag h is defined in terms of the sample autocovariance function:

$$\hat{\rho}(h) = \frac{\hat{\gamma}(h)}{\hat{\gamma}(0)}, \quad |h| < n$$

The sample autocovariance function of lag h (of $\left\{\theta_i^t\right\}$) is defined by

$$\hat{\gamma}(h) = \frac{1}{n-h} \sum_{t=1}^{n-h} \left(\theta_i^{t+h} - \bar{\theta}_i \right) \left(\theta_i^t - \bar{\theta}_i \right), \quad 0 \le h < n$$

Effective Sample Size

You can use autocorrelation and trace plots to examine the mixing of a Markov chain. A closely related measure of mixing is the effective sample size (ESS) (Kass et al. 1998).

ESS is defined as follows:

$$\text{ESS} = \frac{n}{\tau} = \frac{n}{1 + 2\sum_{k=1}^{\infty} \rho_k(\theta)}$$

where n is the total sample size and $\rho_k(\theta)$ is the autocorrelation of lag k for θ. The quantity τ is referred to as the autocorrelation time. To estimate τ, the Bayesian procedures first find a cutoff point k after which the autocorrelations are very close to zero, and then sum all the ρ_k up to that point. The cutoff point k is such that $\rho_k < 0.01$ or $\rho_k < 2s_k$, where s_k is the estimated standard deviation:

$$s_k = 2\sqrt{\left(\frac{1}{n}\left(1 + 2\sum_{j=1}^{k-1}\rho_k^2(\theta)\right)\right)}$$

ESS and τ are inversely proportional to each other, and low ESS or high τ indicates bad mixing of the Markov chain.

Summary Statistics

Let $\boldsymbol{\theta}$ be a p-dimensional parameter vector of interest: $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_p\}$. For each $i \in \{1, \ldots, p\}$, there are n observations: $\theta_i = \{\theta_i^t,\ t = 1, \ldots, n\}$.

Mean

The posterior mean is calculated by using the following formula:

$$E(\theta_i|\mathbf{y}) \approx \bar{\theta}_i = \frac{1}{n}\sum_{t=1}^{n}\theta_i^t, \text{ for } i = 1, \ldots, n$$

Standard Deviation

Sample standard deviation (expressed in variance term) is calculated by using the following formula:

$$\mathrm{Var}(\theta_i|\mathbf{y}) \approx s_i^2 = \frac{1}{n-1}\sum_{t=1}^{n}\left(\theta_i^t - \bar{\theta}_i\right)^2$$

Standard Error of the Mean Estimate

Suppose you have n iid samples, the mean estimate is $\bar{\theta}_i$, and the sample standard deviation is s_i. The standard error of the estimate is $\hat{\sigma}_i/\sqrt{n}$. However, positive autocorrelation (see the section "Autocorrelations" on page 169 for a definition) in the MCMC samples makes this an underestimate. To take account of the autocorrelation, the Bayesian procedures correct the standard error by using effective sample size (see the section "Effective Sample Size" on page 169).

Given an effective sample size of m, the standard error for $\bar{\theta}_i$ is $\hat{\sigma}_i/\sqrt{m}$. The procedures use the following formula (expressed in variance term):

$$\widehat{\mathrm{Var}}(\bar{\theta}_i) = \frac{1 + 2\sum_{k=1}^{\infty}\rho_k(\theta_i)}{n} \cdot \frac{\sum_{t=1}^{n}\left(\theta_i^t - \bar{\theta}_i\right)^2}{(n-1)}$$

The standard error of the mean is also known as the Monte Carlo standard error (MCSE). The MCSE provides a measurement of the accuracy of the posterior estimates, and small values do not necessarily indicate that you have recovered the true posterior mean.

Percentiles

Sample percentiles are calculated using Definition 5 (see Chapter 4, "The UNIVARIATE Procedure" (*Base SAS Procedures Guide: Statistical Procedures*),).

Correlation

Correlation between θ_i and θ_j is calculated as

$$r_{ij} = \frac{\sum_{t=1}^{n} \left(\theta_i^t - \bar{\theta}_i\right)\left(\theta_j^t - \bar{\theta}_j\right)}{\sqrt{\sum_t \left(\theta_i^t - \bar{\theta}_i\right)^2 \sum_t \left(\theta_j^t - \bar{\theta}_j\right)^2}}$$

Covariance

Covariance θ_i and θ_j is calculated as

$$s_{ij} = \sum_{t=1}^{n} \left(\theta_i^t - \bar{\theta}_i\right)\left(\theta_j^t - \bar{\theta}_j\right)/(n-1)$$

Equal-Tail Credible Interval

Let $\pi\left(\theta_i|\mathbf{y}\right)$ denote the marginal posterior cumulative distribution function of θ_i. A $100\,(1-\alpha)\,\%$ Bayesian equal-tail credible interval for θ_i is $\left(\theta_i^{\alpha/2}, \theta_i^{1-\alpha/2}\right)$, where $\pi\left(\theta_i^{\alpha/2}|\mathbf{y}\right) = \frac{\alpha}{2}$, and $\pi\left(\theta_i^{1-\alpha/2}|\mathbf{y}\right) = 1 - \frac{\alpha}{2}$. The interval is obtained using the empirical $\frac{\alpha}{2}$th and $(1-\frac{\alpha}{2})$th percentiles of $\left\{\theta_i^t\right\}$.

Highest Posterior Density (HPD) Interval

For a definition of an HPD interval, see the section "Interval Estimation" on page 149. The procedures use the Chen-Shao algorithm (Chen and Shao 1999; Chen, Shao, and Ibrahim 2000) to estimate an empirical HPD interval of θ_i:

1. Sort $\left\{\theta_i^t\right\}$ to obtain the ordered values:

$$\theta_{i(1)} \leq \theta_{i(2)} \leq \cdots \leq \theta_{i(n)}$$

2. Compute the $100\,(1-\alpha)\,\%$ credible intervals:

$$R_j\,(n) = \left(\theta_{i(j)}, \theta_{i(j+[(1-\alpha)n])}\right)$$

for $j = 1, 2, \ldots, n - [(1-\alpha)\,n]$.

3. The $100\,(1-\alpha)\,\%$ HPD interval, denoted by $R_{j*}\,(n)$, is the one with the smallest interval width among all credible intervals.

Deviance Information Criterion (DIC)

The deviance information criterion (DIC) (Spiegelhalter et al. 2002) is a model assessment tool, and it is a Bayesian alternative to Akaike's information criterion (AIC) and the Bayesian information criterion (BIC, also known as the Schwarz criterion). The DIC uses the posterior densities, which means that it takes the prior information into account. The criterion can be applied to nonnested models and models that have non-iid data. Calculation of the DIC in MCMC is trivial—it does not require maximization over the parameter space, like the AIC and BIC. A smaller DIC indicates a better fit to the data set.

Letting $\boldsymbol{\theta}$ be the parameters of the model, the deviance information formula is

$$\text{DIC} = \overline{D(\boldsymbol{\theta})} + p_D = D(\bar{\boldsymbol{\theta}}) + 2p_D$$

where

$D(\boldsymbol{\theta}) = 2\,(\log(f(\mathbf{y})) - \log(p(\mathbf{y}|\boldsymbol{\theta})))$: deviance

where

$p(\mathbf{y}|\boldsymbol{\theta})$: likelihood function with the normalizing constants.

$f(\mathbf{y})$: a standardizing term that is a function of the data alone. This term is constant with respect to the parameter and is irrelevant when you compare different models that have the same likelihood function. Since the term cancels out in DIC comparisons, its calculation is often omitted.

NOTE: You can think of the deviance as the difference in twice the log likelihood between the saturated, $f(\mathbf{y})$, and fitted, $p(\mathbf{y}|\boldsymbol{\theta})$, models.

$\bar{\boldsymbol{\theta}}$: posterior mean, approximated by $\frac{1}{n}\sum_{t=1}^{n}\boldsymbol{\theta}^t$

$\overline{D(\boldsymbol{\theta})}$: posterior mean of the deviance, approximated by $\frac{1}{n}\sum_{t=1}^{n} D(\boldsymbol{\theta}^t)$. The expected deviation measures how well the model fits the data.

$D(\bar{\boldsymbol{\theta}})$: deviance evaluated at $\bar{\boldsymbol{\theta}}$, equal to $-2\log(p(\mathbf{y}|\bar{\boldsymbol{\theta}}))$. It is the deviance evaluated at your "best" posterior estimate.

p_D: effective number of parameters. It is the difference between the measure of fit and the deviance at the estimates: $\overline{D(\boldsymbol{\theta})} - D(\bar{\boldsymbol{\theta}})$. This term describes the complexity of the model, and it serves as a penalization term that corrects deviance's propensity toward models with more parameters.

A Bayesian Reading List

This section lists a number of Bayesian textbooks of varying difficulty degrees and a few tutorial/review papers.

Textbooks

Introductory Books

Berry, D. A. (1996), *Statistics: A Bayesian Perspective*, London: Duxbury Press.

Bolstad, W. M. (2007), *Introduction to Bayesian Statistics*, 2nd ed. New York: John Wiley & Sons.

DeGroot, M. H. and Schervish, M. J. (2002), *Probability and Statistics*, Reading, MA: Addison Wesley.

Gamerman, D. and Lopes, H. F. (2006), *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference,* 2nd ed. London: Chapman & Hall/CRC.

Ghosh, J. K., Delampady, M., and Samanta, T. (2006), *An Introduction to Bayesian Analysis,* New York: Springer-Verlag.

Lee, P. M. (2004), *Bayesian Statistics: An Introduction,* 3rd ed. London: Arnold.

Sivia, D. S. (1996), *Data Analysis: A Bayesian Tutorial,* Oxford: Oxford University Press.

Intermediate-Level Books

Box, G. E. P., and Tiao, G. C. (1992), *Bayesian Inference in Statistical Analysis,* New York: John Wiley & Sons.

Chen, M. H., Shao Q. M., and Ibrahim, J. G. (2000), *Monte Carlo Methods in Bayesian Computation*, New York: Springer-Verlag.

Gelman, A. and Hill, J. (2006), *Data Analysis Using Regression and Multilevel/Hierarchical Models*, Cambridge: Cambridge University Press.

Goldstein, M. and Woof, D. A. (2007), *Bayes Linear Statistics: Theory and Methods*, New York: John Wiley & Sons.

Harney, H. L. (2003), *Bayesian Inference: Parameter Estimation and Decisions*, New York: Springer-Verlag.

Leonard, T. and Hsu, J. S. (1999), *Bayesian Methods: An Analysis for Statisticians and Interdisciplinary Researchers*, Cambridge: Cambridge University Press.

Liu, J. S. (2001), *Monte Carlo Strategies in Scientific Computing*, New York: Springer-Verlag.

Marin, J. M. and Robert, C. P. (2007), *Bayesian Core: a Practical Approach to Computational Bayesian Statistics*, New York: Springer-Verlag.

Press, S. J. (2002), *Subjective and Objective Bayesian Statistics: Principles, Models, and Applications,* 2nd ed. New York: Wiley-Interscience.

Robert, C. P. (2001), *The Bayesian Choice*, 2nd ed. New York: Springer-Verlag.

Robert, C. P. and Casella, G. (2004), *Monte Carlo Statistical Methods*, 2nd ed. New York: Springer-Verlag.

Tanner, M. A. (1993), *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions*, New York: Springer-Verlag.

Advanced Titles

Berger, J. O. (1985), *Statistical Decision Theory and Bayesian Analysis*, New York: Springer-Verlag.

Bernardo, J. M. and Smith, A. F. M. (2007), *Bayesian Theory*, 2nd ed. New York: John Wiley & Sons.

de Finetti, B. (1992), *Theory of Probability*, New York: John Wiley & Sons.

Jeffreys, H. (1998), *Theory of Probability*, Oxford: Oxford University Press.

O'Hagan, A. (1994), *Bayesian Inference*, volume 2B of *Kendall's Advanced Theory of Statistics,* London: Arnold.

Savage, L. J. (1954), *The Foundations of Statistics*, New York: John Wiley & Sons.

Books Motivated by Statistical Applications and Data Analysis

Carlin, B. and Louris, T. A. (2000), *Bayes and Empirical Bayes Methods for Data Analysis*, 2nd ed. London: Chapman & Hall.

Congdon, P. (2006), *Bayesian Statistical Modeling*, 2nd ed. New York: John Wiley & Sons.

Congdon, P. (2003), *Applied Bayesian Modeling*, New York: John Wiley & Sons.

Congdon, P. (2005), *Bayesian Models for Categorical Data*, New York: John Wiley & Sons.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian Data Analysis*, 3rd ed. London: Chapman & Hall.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996), *Markov Chain Monte Carlo in Practice*, London: Chapman & Hall.

Tutorial and Review Papers on MCMC

Besag, J., Green, P., Higdon, D., and Mengersen, K. (1995), "Bayesian Computation and Stochastic Systems," *Statistical Science*, 10(1), 3–66.

Casella, G. and George, E. (1992), "Explaining the Gibbs Sampler," *The American Statistician*, 46, 167–174.

Chib, S. and Greenberg, E. (1995), "Understanding the Metropolis-Hastings Algorithm," *The American Statistician*, 49, 327–335.

Chib, S. and Greenberg, E. (1996), "Markov Chain Monte Carlo Simulation Methods in Econometrics," *Econometric Theory*, 12, 409–431.

Kass, R. E., Carlin, B. P., Gelman, A., and Neal, R. M. (1998), "Markov Chain Monte Carlo in Practice: A Roundtable Discussion," *Statistical Science*, 52(2), 93–100.

References

Amit, Y. (1991), "On Rates of Convergence of Stochastic Relaxation for Gaussian and Non-Gaussian Distributions," *Journal of Multivariate Analysis*, 38, 82–99.

Applegate, D., Kannan, R., and Polson, N. (1990), *Random Polynomial Time Algorithms for Sampling from Joint Distributions*, Technical report, School of Computer Science, Carnegie Mellon University.

Berger, J. O. (1985), *Statistical Decision Theory and Bayesian Analysis*, Second Edition, New York: Springer-Verlag.

Berger, J. O. (2006), "The Case for Objective Bayesian Analysis," *Bayesian Analysis*, 3, 385–402, http://ba.stat.cmu.edu/journal/2006/vol01/issue03/berger.pdf.

Berger, J. O. and Wolpert, R. (1988), *The Likelihood Principle*, 9, Second Edition, Hayward, California: Institute of Mathematical Statistics, monograph series.

Bernardo, J. M. and Smith, A. F. M. (1994), *Bayesian Theory*, New York: John Wiley & Sons.

Besag, J. (1974), "Spatial Interaction and the Statistical Analysis of Lattic Systems," *Journal of the Royal Statistical Society B*, 36, 192–326.

Billingsley, P. (1986), *Probability and Measure*, Second Edition, New York: John Wiley & Sons.

Box, G. E. P. and Tiao, G. C. (1973), *Bayesian Inference in Statistical Analysis*, Wiley Classics Library Edition, published 1992, New York: John Wiley & Sons.

Breiman, L. (1968), *Probability*, Reading, MA: Addison-Wesley.

Brooks, S. P. and Gelman, A. (1997), "General Methods for Monitoring Convergence of Iterative Simulations," *Journal of Computational and Graphical Statistics*, 7, 434–455.

Brooks, S. P. and Roberts, G. O. (1998), "Assessing Convergence of Markov Chain Monte Carlo Algorithms," *Statistics and Computing*, 8, 319–335.

Brooks, S. P. and Roberts, G. O. (1999), "On Quantile Estimation and Markov Chain Monte Carlo Convergence," *Biometrika*, 86(3), 710–717.

Carlin, B. P. and Louis, T. A. (2000), *Bayes and Empirical Bayes Methods for Data Analysis*, Second Edition, London: Chapman & Hall.

Casella, G. and George, E. I. (1992), "Explaining the Gibbs Sampler," *The American Statistician*, 46, 167–174.

Chan, K. S. (1993), "Asymptotic Behavior of the Gibbs Sampler," *Journal of the American Statistical Association*, 88, 320–326.

Chen, M. H. and Shao, Q. M. (1999), "Monte Carlo Estimation of Bayesian Credible and HPD Intervals," *Journal of Computational and Graphical Statistics*, 8, 69–92.

Chen, M. H., Shao, Q. M., and Ibrahim, J. G. (2000), *Monte Carlo Methods in Bayesian Computation*, New York: Springer-Verlag.

Chib, S. and Greenberg, E. (1995), "Understanding the Metropolis-Hastings Algorithm," *The American Statistician*, 49, 327–335.

Congdon, P. (2001), *Bayesian Statistical Modeling*, John Wiley & Sons.

Congdon, P. (2003), *Applied Bayesian Modeling*, John Wiley & Sons.

Congdon, P. (2005), *Bayesian Models for Categorical Data*, John Wiley & Sons.

Cowles, M. K. and Carlin, B. P. (1996), "Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review," *Journal of the American Statistical Association*, 883–904.

DeGroot, M. H. and Schervish, M. J. (2002), *Probability and Statistics*, 3rd Edition, Reading, MA: Addison-Wesley.

Feller, W. (1968), *An Introduction to Probability Theory and Its Applications*, Third Edition, New York: John Wiley & Sons.

Gelfand, A. E., Hills, S. E., Racine-Poon, A., and Smith, A. F. M. (1990), "Illustration of Bayesian Inference in Normal Data Models Using Gibbs Sampling," *Journal of the American Statistical Association*, 85, 972–985.

Gelman, A., Carlin, J., Stern, H., and Rubin, D. (2004), *Bayesian Data Analysis*, Second Edition, London: Chapman & Hall.

Gelman, A. and Rubin, D. B. (1992), "Inference from Iterative Simulation Using Multiple Sequences," *Statistical Science*, 7, 457–472.

Geman, S. and Geman, D. (1984), "Stochastic Relaxation, Gibbs Distribution, and the Bayesian Restoration of Images," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 6, 721–741.

Geweke, J. (1992), "Evaluating the Accuracy of Sampling-Based Approaches to Calculating Posterior Moments," in J. M. Bernardo, J. O. Berger, A. P. Dawiv, and A. F. M. Smith, eds., *Bayesian Statistics*, volume 4, Oxford, UK: Clarendon Press.

Gilks, W. (2003), "Adaptive Metropolis Rejection Sampling (ARMS)," software from MRC Biostatistics Unit, Cambridge, UK, http://www.maths.leeds.ac.uk/~wally.gilks/adaptive.rejection/web_page/Welcome.html.

Gilks, W. R., Best, N. G., and Tan, K. K. C. (1995), "Adaptive Rejection Metropolis Sampling with Gibbs Sampling," *Applied Statistics*, 44, 455–472.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996), *Markov Chain Monte Carlo in Practice*, London: Chapman & Hall.

Gilks, W. R. and Wild, P. (1992), "Adaptive Rejection Sampling for Gibbs Sampling," *Applied Statistics*, 41, 337–348.

Goldstein, M. (2006), "Subjective Bayesian Analysis: Principles and Practice," *Bayesian Analysis*, 3, 403–420, http://ba.stat.cmu.edu/journal/2006/vol01/issue03/goldstein.pdf.

Hastings, W. K. (1970), "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, 57, 97–109.

Heidelberger, P. and Welch, P. D. (1981), "A Spectral Method for Confidence Interval Generation and Run Length Control in Simulations," *Communication of the ACM*, 24, 233–245.

Heidelberger, P. and Welch, P. D. (1983), "Simulation Run Length Control in the Presence of an Initial Transient," *Operations Research*, 31, 1109–1144.

Jeffreys, H. (1961), *Theory of Probability*, third Edition, Oxford: Oxford University Press.

Karlin, S. and Taylor, H. (1975), *A First Course in Stochastic Processes*, Second Edition, Orlando, FL: Academic Press.

Kass, R. E., Carlin, B. P., Gelman, A., and Neal, R. (1998), "Markov Chain Monte Carlo in Practice: A Roundtable Discussion," *The American Statistician*, 52, 93–100.

Kass, R. E. and Wasserman, L. (1996), "Formal Rules of Selecting Prior Distributions: A Review and Annotated Bibliography," *Journal of the American Statistical Association*, 91, 343–370.

Liu, C., Wong, W. H., and Kong, A. (1991a), *Correlation Structure and Convergence Rate of the Gibbs Sampler (I): Application to the Comparison of Estimators and Augmentation Scheme*, Technical report, Department of Statistics, University of Chicago.

Liu, C., Wong, W. H., and Kong, A. (1991b), *Correlation Structure and Convergence Rate of the Gibbs Sampler (II): Applications to Various Scans*, Technical report, Department of Statistics, University of Chicago.

Liu, J. S. (2001), *Monte Carlo Strategies in Scientific Computing*, Springer-Verlag.

MacEachern, S. N. and Berliner, L. M. (1994), "Subsampling the Gibbs Sampler," *The American Statistician*, 48, 188–190.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E. (1953), "Equation of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, 21, 1087–1092.

Metropolis, N. and Ulam, S. (1949), "The Monte Carlo Method," *Journal of the American Statistical Association*, 44.

Meyn, S. P. and Tweedie, R. L. (1993), *Markov Chains and Stochastic Stability*, Berlin: Springer-Verlag.

Press, S. J. (2003), *Subjective and Objective Bayesian Statistics*, New York: John Wiley & Sons.

Raftery, A. E. and Lewis, S. M. (1992), "One Long Run with Diagnostics: Implementation Strategies for Markov Chain Monte Carlo," *Statistical Science*, 7, 493–497.

Raftery, A. E. and Lewis, S. M. (1996), "The Number of Iterations, Convergence Diagnostics and Generic Metropolis Algorithms," in W. R. Gilks, D. J. Spiegelhalter, and S. Richardson, eds., *Markov Chain Monte Carlo in Practice*, London, UK: Chapman & Hall.

Robert, C. P. (2001), *The Bayesian Choice*, Second Edition, New York: Springer-Verlag.

Robert, C. P. and Casella, G. (2004), *Monte Carlo Statistical Methods*, second Edition, New York: Springer-Verlag.

Roberts, G. O. (1996), "Markov Chain Concepts Related to Sampling Algorithms," in W. R. Gilks, D. J. Spiegelhalter, and S. Richardson, eds., *Markov Chain Monte Carlo in Practice*, 45–58, London: Chapman & Hall.

Rosenthal, J. S. (1991a), *Rates of Convergence for Data Augmentation on Finite Sample Spaces*, Technical report, Department of Mathematics, Harvard University.

Rosenthal, J. S. (1991b), *Rates of Convergence for Gibbs Sampling for Variance Component Models*, Technical report, Department of Mathematics, Harvard University.

Ross, S. M. (1997), *Simulation*, Second Edition, Orlando, FL: Academic Press.

Schervish, M. J. and Carlin, B. P. (1992), "On the Convergence of Successive Substitution Sampling," *Journal of Computational and Graphical Statistics*, 1, 111–127.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van der Linde, A. (2002), "Bayesian Measures of Model Complexity and Fit," *Journal of the Royal Statistical Society, Series B*, 64(4), 583–616, with discussion.

Tanner, M. A. (1993), *Tools for Statistical Inference: Methods for the Exploration of Posterior Distributions and Likelihood Functions*, New York: Springer-Verlag.

Tanner, M. A. and Wong, W. H. (1987), "The Calculation of Posterior Distributions by Data Augmentation," *Journal of the American Statistical Association*, 82, 528–550.

Tierney, L. (1994), "Markov Chains for Exploring Posterior Distributions," *Annals of Statistics*, 22(4), 1701–1762.

von Mises, R. (1918), "Über die 'Ganzzahligkeit' der Atomgewicht und verwandte Fragen," *Physikal. Z.*, 19, 490–500.

Wasserman, L. (2004), *All of Statistics: A Concise Course in Statistical Inference*, New York: Springer-Verlag.

Chapter 8

Introduction to Categorical Data Analysis Procedures

Contents

Overview: Categorical Data Analysis Procedures **181**
Introduction **183**
Sampling Frameworks and Distribution Assumptions **185**
Simple Random Sampling: One Population 185
Stratified Simple Random Sampling: Multiple Populations 186
Observational Data: Analyzing the Entire Population 187
Randomized Experiments 188
Relaxation of Sampling Assumptions 188
Comparison of PROC FREQ and the Modeling Procedures **189**
Comparison of Modeling Procedures **190**
Logistic Regression 191
References **193**

Overview: Categorical Data Analysis Procedures

There are two approaches to performing categorical data analyses. The first computes statistics based on tables defined by *categorical variables* (variables that assume only a limited number of discrete values), performs hypothesis tests about the association between these variables, and requires the assumption of a randomized process; following Stokes, Davis, and Koch (2000), call these methods *randomization procedures*. The other approach investigates the association by modeling a categorical response variable, regardless of whether the explanatory variables are continuous or categorical; call these methods *modeling procedures*. Several procedures in SAS/STAT software can be used for the analysis of categorical data.

The randomization procedures are:

FREQ builds frequency tables or contingency tables and can produce numerous statistics. For one-way frequency tables, it can perform tests for equal proportions, specified proportions, or the binomial proportion. For contingency tables, it can compute various tests and measures of association and agreement including chi-square statistics, odds ratios, correlation statistics, Fisher's exact test for any size

two-way table, kappa, and trend tests. In addition, it performs stratified analysis, computing Cochran-Mantel-Haenszel statistics and estimates of the common relative risk. Exact p-values and confidence intervals are available for various test statistics and measures. See Chapter 35, "The FREQ Procedure," for more information.

SURVEYFREQ incorporates complex sample designs to analyze one-way, two-way, and multiway crosstabulation tables. Estimates population totals and proportions and performs tests of goodness-of-fit and independence. See Chapter 14, "Introduction to Survey Procedures," and Chapter 83, "The SURVEYFREQ Procedure," for more information.

The modeling procedures, which require a categorical response variable, are:

CATMOD fits linear models to functions of categorical data, facilitating such analyses as regression, analysis of variance, linear modeling, log-linear modeling, logistic regression, and repeated measures analysis. Maximum likelihood estimation is used for the analysis of logits and generalized logits, and weighted least squares analysis is used for fitting models to other response functions. Iterative proportional fitting (IPF), which avoids the need for parameter estimation, is available for fitting hierarchical log-linear models when there is a single population. See Chapter 28, "The CATMOD Procedure," for more information.

GENMOD fits generalized linear models with maximum-likelihood methods. This family includes logistic, probit, and complementary log-log regression models for binomial data, Poisson and negative binomial regression models for count data, and multinomial models for ordinal response data. It performs likelihood ratio and Wald tests for Type I, Type III, and user-defined contrasts. It analyzes repeated measures data with generalized estimating equation (GEE) methods. Bayesian analysis capabilities for generalized linear models are also available. See Chapter 37, "The GENMOD Procedure," for more information.

GLIMMIX fits generalized linear mixed models with maximum-likelihood methods. If the model does not contain random effects, the GLIMMIX procedure fits generalized linear models by the method of maximum likelihood. This family includes logistic, probit, and complementary log-log regression models for binomial data, Poisson and negative binomial regression models for count data, and multinomial models for ordinal response data. See Chapter 38, "The GLIMMIX Procedure," for more information.

LOGISTIC fits linear logistic regression models for discrete response data with maximum-likelihood methods. It provides four variable selection methods, computes regression diagnostics, and compares and outputs receiver operating characteristic curves. It can also perform stratified conditional logistic regression analysis for binary response data and exact conditional regression analysis for binary and nominal response data. The logit link function in the logistic regression models can be replaced by the probit function or the complementary log-log function. See Chapter 51, "The LOGISTIC Procedure," for more information.

PROBIT fits models with probit, logit, or complementary log-log links for quantal assay or other discrete event data. It is mainly designed for dose-response analysis

with a natural response rate. It computes the fiducial limits for the dose variable and provides various graphical displays for the analysis. See Chapter 71, "The PROBIT Procedure," for more information.

SURVEYLOGISTIC fits logistic models for binary and ordinal outcomes to survey data by maximum likelihood, incorporating complex survey sample designs. See Chapter 84, "The SURVEYLOGISTIC Procedure," for more information.

Also see Chapter 3, "Introduction to Statistical Modeling with SAS/STAT Software," and Chapter 4, "Introduction to Regression Procedures," for more information about all the modeling and regression procedures.

Other procedures that can be used for categorical data analysis and modeling are:

CORRESP performs simple and multiple correspondence analyses, using a contingency table, Burt table, binary table, or raw categorical data as input. See Chapter 9, "Introduction to Multivariate Procedures," and Chapter 30, "The CORRESP Procedure," for more information.

PRINQUAL performs a principal component analysis of qualitative and/or quantitative data, and multidimensional preference analysis. See Chapter 9, "Introduction to Multivariate Procedures," and Chapter 70, "The PRINQUAL Procedure," for more information.

TRANSREG fits univariate and multivariate linear models, optionally with spline and other nonlinear transformations. Models include ordinary regression and ANOVA, multiple and multivariate regression, metric and nonmetric conjoint analysis, metric and nonmetric vector and ideal point preference mapping, redundancy analysis, canonical correlation, and response surface regression. See Chapter 4, "Introduction to Regression Procedures," and Chapter 90, "The TRANSREG Procedure," for more information.

Introduction

A *categorical variable* is a variable that assumes only a limited number of discrete values. The measurement scale for a categorical variable is unrestricted. It can be *nominal*, which means that the observed levels are not ordered. It can be *ordinal*, which means that the observed levels are ordered in some way. Or it can be *interval*, which means that the observed levels are ordered and numeric and that any interval of one unit on the scale of measurement represents the same amount, regardless of its location on the scale. One example of a categorical variable is litter size; another is the number of times a subject has been married. A variable that lies on a nominal scale is sometimes called a *qualitative* or *classification variable*.

Categorical data result from observations on multiple subjects where one or more categorical variables are observed for each subject. If there is only one categorical variable, then the data are generally represented by a *frequency table*, which lists each observed value of the variable and its frequency of occurrence.

If there are two or more categorical variables, then a subject's *profile* is defined as the subject's observed values for each of the variables. Such categorical data can be represented by a frequency table that lists each observed profile and its frequency of occurrence.

If there are exactly two categorical variables, then the data are often represented by a two-dimensional *contingency table*, which has one row for each level of variable 1 and one column for each level of variable 2. The intersections of rows and columns, called *cells*, correspond to variable profiles, and each cell contains the frequency of occurrence of the corresponding profile.

If there are more than two categorical variables, then the data can be represented by a *multidimensional contingency table*. There are two commonly used methods for displaying such tables, and both require that the variables be divided into two sets.

- In the first method, one set contains a row variable and a column variable for a two-dimensional contingency table, and the second set contains all of the other variables. The variables in the second set are used to form a set of profiles. Thus, the data are represented as a series of two-dimensional contingency tables, one for each profile. This is the data representation used by PROC FREQ. For example, if you request tables for RACE*SEX*AGE*INCOME, the FREQ procedure represents the data as a series of contingency tables: the row variable is AGE, the column variable is INCOME, and the combinations of levels of RACE and SEX form a set of profiles.
- In the second method, one set contains the independent variables, and the other set contains the dependent variables. Profiles based on the independent variables are called *population profiles*, whereas those based on the dependent variables are called *response profiles*. A two-dimensional contingency table is then formed, with one row for each population profile and one column for each response profile. Since any subject can have only one population profile and one response profile, the contingency table is uniquely defined. This is the data representation used by the modeling procedures.

NOTE: Modeling procedures for categorical data analysis only require that the response variable be categorical—the explanatory variables are allowed to be continuous or categorical. However, note that PROC CATMOD was designed to handle contingency table data, and it does not efficiently handle continuous covariates.

Sampling Frameworks and Distribution Assumptions

This section discusses the sampling frameworks and distribution assumptions for the modeling and randomization procedures.

Simple Random Sampling: One Population

Suppose you take a simple random sample of 100 people and ask each person the following question, "Of the three colors red, blue, and green, which is your favorite?" You then tabulate the results in a frequency table as shown in Table 8.1.

Table 8.1 One-Way Frequency Table

	Favorite Color			
	Red	**Blue**	**Green**	**Total**
Frequency	52	31	17	100
Proportion	0.52	0.31	0.17	1.00

In the population you are sampling, you assume there is an unknown probability that a population member, selected at random, would choose any given color. In order to estimate that probability, you use the sample proportion

$$p_j = \frac{n_j}{n}$$

where n_j is the frequency of the jth response and n is the total frequency.

Because of the random variation inherent in any random sample, the frequencies have a probability distribution representing their relative frequency of occurrence in a hypothetical series of samples. For a simple random sample, the distribution of frequencies for a frequency table with three levels is as follows. The probability that the first frequency is n_1, the second frequency is n_2, and the third is $n_3 = n - n_1 - n_2$, is given by

$$\Pr(n_1, n_2, n_3) = \frac{n!}{n_1! n_2! n_3!} \pi_1^{n_1} \pi_2^{n_2} \pi_3^{n_3}$$

where π_j is the true probability of observing the jth response level in the population.

This distribution, called the *multinomial distribution*, can be generalized to any number of response levels. The special case of two response levels is called the *binomial distribution*.

Simple random sampling is the type of sampling required by the (non-survey) modeling procedures when there is one population. The modeling procedures use the multinomial distribution to estimate a probability vector and its covariance matrix. If the sample size is sufficiently large, then the probability vector is approximately normally distributed as a result of central limit theory. This result is used to compute appropriate test statistics for the specified statistical model.

Stratified Simple Random Sampling: Multiple Populations

Suppose you take two simple random samples, 50 men and 50 women, and ask the same question as before. You are now sampling two different populations that may have different response probabilities. The data can be tabulated as shown in Table 8.2.

Table 8.2 Two-Way Contingency Table: Sex by Color

	Favorite Color			
Sex	**Red**	**Blue**	**Green**	**Total**
Male	30	10	10	50
Female	20	10	20	50
Total	50	20	30	100

Note that the row marginal totals (50, 50) of the contingency table are fixed by the sampling design, but the column marginal totals (50, 20, 30) are random. There are six probabilities of interest for this table, and they are estimated by the sample proportions

$$p_{ij} = \frac{n_{ij}}{n_i}$$

where n_{ij} denotes the frequency for the ith population and the jth response and n_i is the total frequency for the ith population. For this contingency table, the sample proportions are shown in Table 8.3.

Table 8.3 Table of Sample Proportions by Sex

	Favorite Color			
Sex	**Red**	**Blue**	**Green**	**Total**
Male	0.60	0.20	0.20	1.00
Female	0.40	0.20	0.40	1.00

The probability distribution of the six frequencies is the *product multinomial distribution*

$$\Pr(n_{11}, n_{12}, n_{13}, n_{21}, n_{22}, n_{23}) = \frac{n_1! n_2! \pi_{11}^{n_{11}} \pi_{12}^{n_{12}} \pi_{13}^{n_{13}} \pi_{21}^{n_{21}} \pi_{22}^{n_{22}} \pi_{23}^{n_{23}}}{n_{11}! n_{12}! n_{13}! n_{21}! n_{22}! n_{23}!}$$

where π_{ij} is the true probability of observing the jth response level in the ith population. The product multinomial distribution is simply the product of two or more individual multinomial distributions since the populations are independent. This distribution can be generalized to any number of populations and response levels.

Stratified simple random sampling is the type of sampling required by the modeling procedures when there is more than one population. The product multinomial distribution is used to estimate a probability vector and its covariance matrix. If the sample sizes are sufficiently large, then the probability vector is approximately normally distributed as a result of central limit theory, and this result is used to compute appropriate test statistics for the specified statistical model. The statistics

are known as Wald statistics, and they are approximately distributed as chi-square when the null hypothesis is true.

Observational Data: Analyzing the Entire Population

Sometimes the observed data do not come from a random sample but instead represent a complete set of observations on some population. For example, suppose a class of 100 students is classified according to sex and favorite color. The results are shown in Table 8.4.

In this case, you could argue that all of the frequencies are fixed since the entire population is observed; therefore, there is no sampling error. On the other hand, you could hypothesize that the observed table has only fixed marginals and that the cell frequencies represent one realization of a conceptual process of assigning color preferences to individuals. The assignment process is open to hypothesis, which means that you can hypothesize restrictions on the joint probabilities.

Table 8.4 Two-Way Contingency Table: Sex by Color

	Favorite Color			
Sex	**Red**	**Blue**	**Green**	**Total**
Male	16	21	20	57
Female	12	20	11	43
Total	28	41	31	100

The usual hypothesis (sometimes called *randomness*) is that the distribution of the column variable (Favorite Color) does not depend on the row variable (Sex). This implies that, for each row of the table, the assignment process corresponds to a simple random sample (without replacement) from the finite population represented by the column marginal totals (or by the column marginal subtotals that remain after sampling other rows). The hypothesis of randomness implies that the probability distribution on the frequencies in the table is the *hypergeometric distribution*.

If the same row and column variables are observed for each of several populations, then the probability distribution of all the frequencies can be called the *multiple hypergeometric distribution.* Each population is called a *stratum*, and an analysis that draws information from each stratum and then summarizes across them is called a *stratified analysis* (or a *blocked analysis* or a *matched analysis*). PROC FREQ does such a stratified analysis, computing test statistics and measures of association.

In general, the populations are formed on the basis of cross-classifications of independent variables. Stratified analysis is a method of adjusting for the effect of these variables without being forced to estimate parameters for them. Note that PROC LOGISTIC can perform analyses on stratified tables as well, using the usual modeling procedure assumptions, by using conditional or exact conditional logistic regression.

The multiple hypergeometric distribution is the one used by PROC FREQ for the computation of Cochran-Mantel-Haenszel statistics. These statistics are in the class of *randomization model test statistics*, which require minimal assumptions for their validity. PROC FREQ uses the multiple hypergeometric distribution to compute the mean and the covariance matrix of a function vector

in order to measure the deviation between the observed and expected frequencies with respect to a particular type of alternative hypothesis. If the cell frequencies are sufficiently large, then the function vector is approximately normally distributed as a result of central limit theory, and PROC FREQ uses this result to compute a quadratic form that has a chi-square distribution when the null hypothesis is true.

Randomized Experiments

Consider a *randomized experiment* in which patients are assigned to one of two treatment groups according to a randomization process that allocates 50 patients to each group. After a specified period of time, each patient's status (cured or uncured) is recorded. Suppose the data shown in Table 8.5 give the results of the experiment. The null hypothesis is that the two treatments are equally effective. Under this hypothesis, treatment is a randomly assigned label that has no effect on the cure rate of the patients. But this implies that each row of the table represents a simple random sample from the finite population whose cure rate is described by the column marginal totals. Therefore, the column marginals (58, 42) are fixed under the hypothesis. Since the row marginals (50, 50) are fixed by the allocation process, the hypergeometric distribution is induced on the cell frequencies. Randomized experiments can also be specified in a stratified framework, and Cochran-Mantel-Haenszel statistics can be computed relative to the corresponding multiple hypergeometric distribution.

Table 8.5 Two-Way Contingency Table: Treatment by Status

	Status		
Treatment	**Cured**	**Uncured**	**Total**
1	36	14	50
2	22	28	50
Total	58	42	100

Relaxation of Sampling Assumptions

As indicated previously, the modeling procedures assume that the data are from a stratified simple random sample, so they use the product multinomial distribution. If the data are not from such a sample, then in many cases it is still possible to use a modeling procedure by arguing that each row of the contingency table *does* represent a simple random sample from some hypothetical population. The extent to which the inferences are generalizable depends on the extent to which the hypothetical population is perceived to resemble the target population.

Similarly, the Cochran-Mantel-Haenszel statistics use the multiple hypergeometric distribution, which requires fixed row and column marginal totals in each contingency table. If the sampling process does not yield a table with fixed margins, then it is usually possible to fix the margins through conditioning arguments similar to the ones used by Fisher when he developed the Exact

Test for 2×2 tables. In other words, if you want fixed marginal totals, you can generally make your analysis conditional on those observed totals.

For more information on sampling models for categorical data, see Bishop, Fienberg, and Holland (1975, Chapter 13) and Agresti (2002, Chapter 1.2).

Comparison of PROC FREQ and the Modeling Procedures

PROC FREQ is used primarily to investigate the relationship between two variables; any confounding variables are taken into account by stratification rather than by parameter estimation. Modeling procedures are used to investigate the relationship among many variables, all of which are integrated into a parametric model.

When a modeling procedure estimates the covariance matrix of the frequencies, it assumes that the frequencies were obtained by a stratified simple random-sampling procedure. However, some modeling procedures can handle different sampling methods. PROC CATMOD can analyze input data that consists of a function vector and a covariance matrix, so you can estimate the covariance matrix of the frequencies in the appropriate manner before modeling the data. PROC SURVEYLOGISTIC can analyze data from a completely different, but known, sampling scheme.

For the FREQ procedure, Fisher's Exact Test and Cochran-Mantel-Haenszel (CMH) statistics are based on the hypergeometric distribution, which corresponds to fixed marginal totals. However, by conditioning arguments, these tests are generally applicable to a wide range of sampling procedures. Similarly, the Pearson and likelihood-ratio chi-square statistics can be derived under a variety of sampling situations.

PROC FREQ can do some traditional nonparametric analysis (such as the Kruskal-Wallis test and Spearman's correlation) since it can generate rank scores internally. Fisher's Exact Test and the CMH statistics are also inherently nonparametric. However, the main vehicle for nonparametric analyses in the SAS System is the NPAR1WAY procedure.

A large sample size is required for the validity of the chi-square distributions, the standard errors, and the covariance matrices for PROC FREQ and the modeling procedures. If sample size is a problem, then PROC FREQ has the advantage with its CMH statistics because it does not use any degrees of freedom to estimate parameters for confounding variables. In addition, PROC FREQ can compute exact p-values for any two-way table, provided that the sample size is sufficiently small in relation to the size of the table. It can also produce exact p-values for many tests, including the test of binomial proportions, the Cochran-Armitage test for trend, and the Jonckheere-Terpstra test for ordered differences among classes. PROC LOGISTIC can perform exact conditional logistic regression and Firth's penalized-likelihood regression to compensate for small sample sizes.

See the procedure chapters for more information. In addition, some well-known texts that deal with analyzing categorical data are listed in the "References" section of this chapter.

Comparison of Modeling Procedures

The CATMOD, GENMOD, GLIMMIX, LOGISTIC, PROBIT, and SURVEYLOGISTIC procedures can all be used for statistical modeling of categorical data.

The CATMOD procedure treats all explanatory (independent) variables as classification variables by default, and you specify continuous covariates in the DIRECT statement. The other procedures treat covariates as continuous by default, and you specify the classification variables in the CLASS statement.

The CATMOD procedure provides weighted least squares estimation of many response functions, such as means, cumulative logits, and proportions, and you can also compute and analyze other response functions that can be formed from the proportions corresponding to the rows of a contingency table. In addition, a user can input and analyze a set of response functions and user-supplied covariance matrix with weighted least squares. PROC CATMOD also provides maximum likelihood estimation for binary and polytomous logistic regression.

The GENMOD procedure is also a general statistical modeling tool which fits generalized linear models to data; it fits several useful models to categorical data including logistic regression, the proportional odds model, and Poisson and negative binomial regression for count data. The GENMOD procedure also provides a facility for fitting generalized estimating equations to correlated response data that are categorical, such as repeated dichotomous outcomes. The GENMOD procedure fits models using maximum likelihood estimation. PROC GENMOD can perform Type I and Type III tests, and it provides predicted values and residuals. Bayesian analysis capabilities for generalized linear models are also available.

The GLIMMIX procedure fits many of the same models as the GENMOD procedure but also allows the inclusion of random effects. The GLIMMIX procedure fits models using maximum likelihood estimation.

The LOGISTIC procedure is specifically designed for logistic regression. It performs the usual logistic regression analysis for dichotomous outcomes and it fits the proportional odds model and the generalized logit model for ordinal and nominal outcomes, respectively, by the method of maximum likelihood. This procedure has capabilities for a variety of model-building techniques, including stepwise, forward, and backward selection. It computes predicted values, the receiver operating characteristics (ROC) curve and the area beneath the curve, and a number of regression diagnostics. It can create output data sets containing these values and other statistics. PROC LOGISTIC can perform a conditional logistic regression analysis (matched-set and case-controlled) for binary response data. For small data sets, PROC LOGISTIC can perform exact conditional logistic regression. Firth's bias-reducing penalized-likelihood method can also be used in place of conditional and exact conditional logistic regression.

The PROBIT procedure is designed for quantal assay or other discrete event data. In additional to performing the logistic regression analysis, it can estimate the threshold response rate. PROC PROBIT can also estimate the values of independent variables that yield a desired response.

The SURVEYLOGISTIC procedure performs logistic regression for binary, ordinal, and nominal responses under a specified complex sampling scheme, instead of the usual stratified simple random sampling.

Stokes, Davis, and Koch (2000) provide substantial discussion of these procedures, particularly the use of the FREQ, LOGISTIC, GENMOD, and CATMOD procedures for statistical modeling.

Logistic Regression

Dichotomous Response

You have many choices of performing logistic regression in the SAS System. The CATMOD, GENMOD, GLIMMIX, LOGISTIC, PROBIT, and SURVEYLOGISTIC procedures fit the usual logistic regression model.

PROC CATMOD might not be efficient when there are continuous independent variables with large numbers of different values. For a continuous variable with a very limited number of values, PROC CATMOD might still be useful.

PROC GLIMMIX enables you to specify random effects in the models; in particular, you can fit a random-intercept logistic regression model.

PROC LOGISTIC provides the capability of model-building and performs conditional and exact conditional logistic regression. It can also use Firth's bias-reducing penalized likelihood method.

PROC PROBIT enables you to estimate the natural response rate and compute fiducial limits for the dose variable.

The LOGISTIC, GENMOD, GLIMMIX, PROBIT, and SURVEYLOGISTIC procedures can analyze summarized data by enabling you to input the numbers of events and trials; the ratio of events to trials must be between 0 and 1.

Ordinal Response

PROC LOGISTIC fits the proportional odds model to the ordinal response data by default, PROC PROBIT fits this model if you specify the logistic distribution, and PROC GENMOD and PROC GLIMMIX fit this model if you specify the CLOGIT link and the multinomial distribution. PROC CATMOD fits the cumulative logit or adjacent-category logit response functions.

Nominal Response

When the response variable is nominal, there is no concept of ordering of the response values. Response functions called *generalized logits* can be fit by the CATMOD, GLIMMIX, and LOGISTIC procedures. PROC CATMOD fits this model by default; PROC GLIMMIX and PROC LOGISTIC require you to specify the GLOGIT link.

Numerical Differences

Differences in the way the models are parameterized and fit might result in different parameter estimates if you perform logistic regression in each of these procedures.

- Parameter estimates from the procedures can differ in sign depending on the ordering of response levels, which you can change if you want.
- The parameter estimates associated with a categorical independent variable might differ among the procedures, since the estimates depend on the coding of the indicator variables in the design matrix. By default, the design matrix column produced by PROC CATMOD and PROC LOGISTIC for a binary independent variable is coded using the values 1 and −1 (deviation from the mean coding, which is a full-rank parameterization). The same column produced by the CLASS statement of PROC GENMOD, PROC GLIMMIX, and PROC PROBIT is coded using 1 and 0 (GLM coding, which is less-than-full-rank parameterization). As a result, the parameter estimate printed by PROC LOGISTIC is one-half of the estimate produced by PROC GENMOD. Both PROC GENMOD and PROC LOGISTIC allow you to select either a full-rank parameterization or the less-than-full-rank parameterization. The GLIMMIX and PROBIT procedures allow only the less-than-full-rank parameterization for the CLASS variables. The CATMOD procedure allows only full-rank parameterizations. See the "Details" sections in the chapters on the CATMOD, GENMOD, GLIMMIX, LOGISTIC, and PROBIT procedures for more information on the generation of the design matrices used by these procedures. See Chapter 18, "Shared Concepts and Topics," for a general discussion of the various parameterizations.
- The maximum-likelihood algorithm used differs among the procedures. PROC LOGISTIC uses the Fisher's scoring method by default, while PROC PROBIT, PROC GENMOD, PROC GLIMMIX, and PROC CATMOD use the Newton-Raphson method. The parameter estimates should be the same for all three procedures, and the standard errors should be the same for the logistic model. For the normal and extreme-value (Gompertz) distributions in PROC PROBIT, which correspond to the probit and cloglog links, respectively, in PROC GENMOD and PROC LOGISTIC, the standard errors might differ. In general, tests computed using the standard errors from the Newton-Raphson method are more conservative.
- The LOGISTIC, GENMOD, GLIMMIX, and PROBIT procedures can fit a cumulative regression model for ordinal response data by using maximum-likelihood estimation. PROC LOGISTIC and PROC GENMOD use a different parameterization from that of PROC PROBIT, which results in different intercept parameters. Estimates of the slope parameters, however, should be the same for both procedures. The estimated standard errors of the slope estimates are slightly different between the procedures because of the different computational algorithms used as default.

References

Agresti, A. (1984), *Analysis of Ordinal Categorical Data*, New York: John Wiley & Sons.

Agresti, A. (2002), *Categorical Data Analysis*, Second Edition, New York: John Wiley & Sons.

Bishop, Y. M. M., Fienberg, S. E., and Holland, P. W. (1975), *Discrete Multivariate Analysis: Theory and Practice*, Cambridge, MA: MIT Press.

Collett, D. (2003), *Modelling Binary Data*, Second Edition, London: Chapman & Hall.

Cox, D. R. and Snell, E. J. (1989), *The Analysis of Binary Data*, Second Edition, London: Chapman & Hall.

Dobson, A. (1990), *An Introduction to Generalized Linear Models*, London: Chapman & Hall.

Fleiss, J. L., Levin, B., and Paik, M. C. (2003), *Statistical Methods for Rates and Proportions*, Third Edition, Hoboken, NJ: John Wiley & Sons.

Freeman, D. H., Jr. (1987), *Applied Categorical Data Analysis*, New York: Marcel Dekker.

Grizzle, J. E., Starmer, C. F., and Koch, G. G. (1969), "Analysis of Categorical Data by Linear Models," *Biometrics*, 25, 489–504.

Hosmer, D. W., Jr. and Lemeshow, S. (2000), *Applied Logistic Regression*, Second Edition, New York: John Wiley & Sons.

McCullagh, P. and Nelder, J. A. (1989), *Generalized Linear Models*, Second Edition, London: Chapman & Hall.

Stokes, M. E., Davis, C. S., and Koch, G. G. (2000), *Categorical Data Analysis Using the SAS System*, Second Edition, Cary, NC: SAS Institute Inc.

Chapter 9

Introduction to Multivariate Procedures

Contents

Overview: Multivariate Procedures 195
Comparison of the PRINCOMP and FACTOR Procedures 197
Comparison of the PRINCOMP and PRINQUAL Procedures 197
Comparison of the PRINCOMP and CORRESP Procedures 198
Comparison of the PRINQUAL and CORRESP Procedures 198
Comparison of the TRANSREG and PRINQUAL Procedures 198
References 199

Overview: Multivariate Procedures

The procedures discussed in this chapter investigate relationships among variables without designating some as independent and others as dependent. Principal component analysis and common factor analysis examine relationships within a single set of variables, whereas canonical correlation looks at the relationship between two sets of variables. The following is a brief description of SAS/STAT multivariate procedures:

CORRESP performs simple and multiple correspondence analyses, with a contingency table, Burt table, binary table, or raw categorical data as input. Correspondence analysis is a weighted form of principal component analysis that is appropriate for frequency data. The results are displayed in plots and tables and are also available in output data sets.

PRINCOMP performs a principal component analysis and outputs standardized or unstandardized principal component scores. The results are displayed in plots and tables and are also available in output data sets.

PRINQUAL performs a principal component analysis of qualitative data and multidimensional preference analysis. The results are displayed in plots and are also available in output data sets.

FACTOR performs principal component and common factor analyses with rotations and outputs component scores or estimates of common factor scores. The results are displayed in plots and tables and are also available in output data sets.

CANCORR performs a canonical correlation analysis and outputs canonical variable scores. The results are displayed in tables and are also available in output data sets for plotting.

Many other SAS/STAT procedures can also analyze multivariate data—for example, the CATMOD, GLM, REG, CALIS, and TRANSREG procedures as well as the procedures for clustering and discriminant analysis.

The purpose of *principal component analysis* (Rao 1964) is to derive a small number of linear combinations (principal components) of a set of variables that retain as much of the information in the original variables as possible. Often a small number of principal components can be used in place of the original variables for plotting, regression, clustering, and so on. Principal component analysis can also be viewed as an attempt to uncover approximate linear dependencies among variables.

The purpose of *common factor analysis* (Mulaik 1972) is to explain the correlations or covariances among a set of variables in terms of a limited number of unobservable, latent variables. The latent variables are not generally computable as linear combinations of the original variables. In common factor analysis, it is assumed that the variables are linearly related if not for uncorrelated random error or *unique variation* in each variable; both the linear relations and the amount of unique variation can be estimated.

Principal component and common factor analysis are often followed by rotation of the components or factors. *Rotation* is the application of a nonsingular linear transformation to components or common factors to aid interpretation.

The purpose of *canonical correlation analysis* (Mardia, Kent, and Bibby 1979) is to explain or summarize the relationship between two sets of variables by finding a small number of linear combinations from each set of variables that have the highest possible between-set correlations. Plots of the canonical variables can be useful in examining multivariate dependencies. If one of the two sets of variables consists of dummy variables generated from a classification variable, the canonical correlation is equivalent to canonical discriminant analysis (see Chapter 27, "The CANDISC Procedure"). If both sets of variables are dummy variables, canonical correlation is equivalent to simple correspondence analysis.

The purpose of *correspondence analysis* (Lebart, Morineau, and Warwick 1984; Greenacre 1984; Nishisato 1980) is to summarize the associations between a set of categorical variables in a small number of dimensions. Correspondence analysis computes scores on each dimension for each row and column category in a contingency table. Plots of these scores show the relationships among the categories.

The PRINQUAL procedure obtains linear and nonlinear transformations of variables by using the method of alternating least squares (Young 1981) to optimize properties of the transformed variables' covariance or correlation matrix. PROC PRINQUAL nonlinearly transforms variables, improving their fit to a principal component model. The name, PRINQUAL, for principal components of qualitative data, comes from the special case analysis of fitting a principal component model to nominal and ordinal scale of measurement variables (Young, Takane, and de Leeuw 1978). However, PROC PRINQUAL also has facilities for smoothly transforming continuous variables. All of PROC PRINQUAL's transformations are also available in the TRANSREG procedure, which fits regression models with nonlinear transformations. PROC PRINQUAL can also perform metric and nonmetric multidimensional preference (MDPREF) analyses (Carroll 1972) and produce plots of the results.

Comparison of the PRINCOMP and FACTOR Procedures

Although PROC FACTOR can be used for common factor analysis, the default method is principal components. PROC FACTOR produces the same results as PROC PRINCOMP except that scoring coefficients from PROC FACTOR are normalized to give principal component scores with unit variance, whereas PROC PRINCOMP by default produces principal component scores with variance equal to the corresponding eigenvalue. PROC PRINCOMP can also compute scores standardized to unit variance. Both procedures produce graphical results through ODS Graphics.

PROC PRINCOMP has the following advantages over PROC FACTOR:

- PROC PRINCOMP is slightly faster if a small number of components is requested.
- PROC PRINCOMP can analyze somewhat larger problems in a fixed amount of memory.
- PROC PRINCOMP can output scores from an analysis of a partial correlation or covariance matrix.
- PROC PRINCOMP is simpler to use.

PROC FACTOR has the following advantages over PROC PRINCOMP for principal component analysis:

- PROC FACTOR produces more output.
- PROC FACTOR does rotations.

If you want to perform a common factor analysis, you must use PROC FACTOR instead of PROC PRINCOMP. Principal component analysis should never be used if a common factor solution is desired (Dziuban and Harris 1973; Lee and Comrey 1979).

Comparison of the PRINCOMP and PRINQUAL Procedures

The PRINCOMP procedure performs principal component analysis. The PRINQUAL procedure finds linear and nonlinear transformations of variables to optimize properties of the transformed variables' covariance or correlation matrix. One property is the sum of the first n eigenvalues, which is a measure of the fit of a principal component model with n components. Use PROC PRINQUAL to find nonlinear transformations of your variables or to perform a multidimensional preference analysis. Use PROC PRINCOMP to fit a principal component model to your data or to PROC PRINQUAL's output data set. PROC PRINCOMP produces a report of the principal component analysis, a number of graphical displays, and output data sets. PROC PRINQUAL produces only a few graphs and an output data set.

Comparison of the PRINCOMP and CORRESP Procedures

As summarized previously, PROC PRINCOMP performs a principal component analysis of interval-scaled data. PROC CORRESP performs correspondence analysis, which is a weighted form of principal component analysis that is appropriate for frequency data. If your data are categorical, use PROC CORRESP instead of PROC PRINCOMP. Both procedures produce graphical displays of the results with ODS Graphics. The plots produced by PROC CORRESP graphically show relationships among the categories of the categorical variables.

Comparison of the PRINQUAL and CORRESP Procedures

Both PROC PRINQUAL and PROC CORRESP can be used to summarize associations among variables measured on a nominal scale. PROC PRINQUAL searches for a single nonlinear transformation of the original scoring of each nominal variable that optimizes some aspect of the covariance matrix of the transformed variables. For example, PROC PRINQUAL could be used to find scorings that maximize the fit of a principal component model with one component. PROC CORRESP uses the crosstabulations of nominal variables, not covariances, and produces multiple scores for each category of each nominal variable. The main conceptual difference between PROC PRINQUAL and PROC CORRESP is that PROC PRINQUAL assumes that the categories of a nominal variable correspond to values of a single underlying interval variable, whereas PROC CORRESP assumes that there are multiple underlying interval variables and therefore uses different category scores for each dimension of the correspondence analysis. Scores from PROC CORRESP on the first dimension match the single set of PROC PRINQUAL scores (with appropriate standardizations for both analyses).

Comparison of the TRANSREG and PRINQUAL Procedures

Both the TRANSREG and PRINQUAL procedures are data transformation procedures that have many of the same transformations. These procedures can either directly perform the specified transformation (such as taking the logarithm of the variable) or search for an optimal transformation (such as a spline with a specified number of knots). Both procedures can use an iterative, alternating least squares analysis. Both procedures create an output data set that can be used as input to other procedures. PROC PRINQUAL displays relatively little output, whereas PROC TRANSREG displays many results. PROC TRANSREG has two sets of variables, usually dependent and independent, and it fits linear models such as ordinary regression and ANOVA, multiple and multivariate regression, metric and nonmetric conjoint analysis, metric and nonmetric vector and ideal point preference mapping, redundancy analysis, canonical correlation, and response surface regression. In contrast, PROC PRINQUAL has one set of variables, fits a principal component model or multidimensional preference analysis, and can also optimize other properties of a correlation or

covariance matrix. PROC TRANSREG performs hypothesis testing and can be used to code experimental designs prior to their use in other analyses. PROC TRANSREG can also perform Box-Cox transformations and fit models with smoothing spline and penalized B-spline transformations.

See Chapter 4, "Introduction to Regression Procedures," for comparisons of the TRANSREG and REG procedures.

References

Carroll, J. D. (1972), "Individual Differences and Multidimensional Scaling," in R. N. Shepard, A. K. Romney, and S. B. Nerlove, eds., *Multidimensional Scaling: Theory and Applications in the Behavioral Sciences (Volume 1)*, New York: Seminar Press.

Dziuban, C. D. and Harris, C. W. (1973), "On the Extraction of Components and the Applicability of the Factor Model," *American Educational Research Journal*, 10, 93–99.

Greenacre, M. J. (1984), *Theory and Applications of Correspondence Analysis*, London: Academic Press.

Lebart, L., Morineau, A., and Warwick, K. M. (1984), *Multivariate Descriptive Statistical Analysis: Correspondence Analysis and Related Techniques for Large Matrices*, New York: John Wiley & Sons.

Lee, H. B. and Comrey, A. L. (1979), "Distortions in a Commonly Used Factor Analytic Procedure," *Multivariate Behavioral Research*, 14, 301–321.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979), *Multivariate Analysis*, London: Academic Press.

Mulaik, S. A. (1972), *The Foundations of Factor Analysis*, New York: McGraw-Hill.

Nishisato, S. (1980), *Analysis of Categorical Data: Dual Scaling and Its Applications*, Toronto: University of Toronto Press.

Rao, C. R. (1964), "The Use and Interpretation of Principal Component Analysis in Applied Research," *Sankhya A*, 26, 329–358.

Young, F. W. (1981), "Quantitative Analysis of Qualitative Data," *Psychometrika*, 46, 357–388.

Young, F. W., Takane, Y., and de Leeuw, J. (1978), "The Principal Components of Mixed Measurement Level Multivariate Data: An Alternating Least Squares Method with Optimal Scaling Features," *Psychometrika*, 43, 279–281.

Chapter 10

Introduction to Discriminant Procedures

Contents

Overview: Discriminant Procedures 201
Background: Discriminant Procedures 202
Example: Contrasting Univariate and Multivariate Analyses 203
References 208

Overview: Discriminant Procedures

The SAS procedures for discriminant analysis fit data with one classification variable and several quantitative variables. The purpose of discriminant analysis can be to find one or more of the following:

- a mathematical rule, or *discriminant function*, for guessing to which class an observation belongs, based on knowledge of the quantitative variables only
- a set of linear combinations of the quantitative variables that best reveals the differences among the classes
- a subset of the quantitative variables that best reveals the differences among the classes

The SAS discriminant procedures are as follows:

DISCRIM	computes various discriminant functions for classifying observations. Linear or quadratic discriminant functions can be used for data with approximately multivariate normal within-class distributions. Nonparametric methods can be used without making any assumptions about these distributions.
CANDISC	performs a canonical analysis to find linear combinations of the quantitative variables that best summarize the differences among the classes.
STEPDISC	uses forward selection, backward elimination, or stepwise selection to try to find a subset of quantitative variables that best reveals differences among the classes.

Background: Discriminant Procedures

The term *discriminant analysis* (Fisher 1936; Cooley and Lohnes 1971; Tatsuoka 1971; Kshirsagar 1972; Lachenbruch 1975, 1979; Gnanadesikan 1977; Klecka 1980; Hand 1981, 1982; Silverman 1986) refers to several different types of analyses. Classificatory discriminant analysis is used to classify observations into two or more known groups on the basis of one or more quantitative variables. Classification can be done by either a parametric method or a nonparametric method in the DISCRIM procedure. A parametric method is appropriate only for approximately normal within-class distributions. The method generates either a linear discriminant function (the within-class covariance matrices are assumed to be equal) or a quadratic discriminant function (the within-class covariance matrices are assumed to be unequal).

When the distribution within each group is not assumed to have any specific distribution or is assumed to have a distribution different from the multivariate normal distribution, nonparametric methods can be used to derive classification criteria. These methods include the kernel method and nearest-neighbor methods. The kernel method uses uniform, normal, Epanechnikov, biweight, or triweight kernels in estimating the group-specific density at each observation. The within-group covariance matrices or the pooled covariance matrix can be used to scale the data.

The performance of a discriminant function can be evaluated by estimating error rates (probabilities of misclassification). Error count estimates and posterior probability error rate estimates can be evaluated with PROC DISCRIM. When the input data set is an ordinary SAS data set, the error rates can also be estimated by cross validation.

In multivariate statistical applications, the data collected are largely from distributions different from the normal distribution. Various forms of nonnormality can arise, such as qualitative variables or variables with underlying continuous but nonnormal distributions. If the multivariate normality assumption is violated, the use of parametric discriminant analysis might not be appropriate. When a parametric classification criterion (linear or quadratic discriminant function) is derived from a nonnormal population, the resulting error rate estimates might be biased.

If your quantitative variables are not normally distributed, or if you want to classify observations on the basis of categorical variables, you should consider using the CATMOD or LOGISTIC procedure to fit a categorical linear model with the classification variable as the dependent variable. Press and Wilson (1978) compare logistic regression and parametric discriminant analysis and conclude that logistic regression is preferable to parametric discriminant analysis in cases for which the variables do not have multivariate normal distributions within classes. However, if you do have normal within-class distributions, logistic regression is less efficient than parametric discriminant analysis. Efron (1975) shows that with two normal populations having a common covariance matrix, logistic regression is between one-half and two-thirds as effective as the linear discriminant function in achieving asymptotically the same error rate.

Do not confuse discriminant analysis with cluster analysis. All varieties of discriminant analysis require prior knowledge of the classes, usually in the form of a sample from each class. In cluster analysis, the data do not include information about class membership; the purpose is to construct a classification. See Chapter 11, "Introduction to Clustering Procedures."

Canonical discriminant analysis is a dimension-reduction technique related to principal components and canonical correlation, and it can be performed by both the CANDISC and DISCRIM procedures. A discriminant criterion is always derived in PROC DISCRIM. If you want canonical discriminant analysis without the use of a discriminant criterion, you should use PROC CANDISC. Stepwise discriminant analysis is a variable-selection technique implemented by the STEPDISC procedure. After selecting a subset of variables with PROC STEPDISC, use any of the other discriminant procedures to obtain more detailed analyses. PROC CANDISC and PROC STEPDISC perform hypothesis tests that require the within-class distributions to be approximately normal, but these procedures can be used descriptively with nonnormal data.

Another alternative to discriminant analysis is to perform a series of univariate one-way ANOVAs. All three discriminant procedures provide summaries of the univariate ANOVAs. The advantage of the multivariate approach is that two or more classes that overlap considerably when each variable is viewed separately might be more distinct when examined from a multivariate point of view.

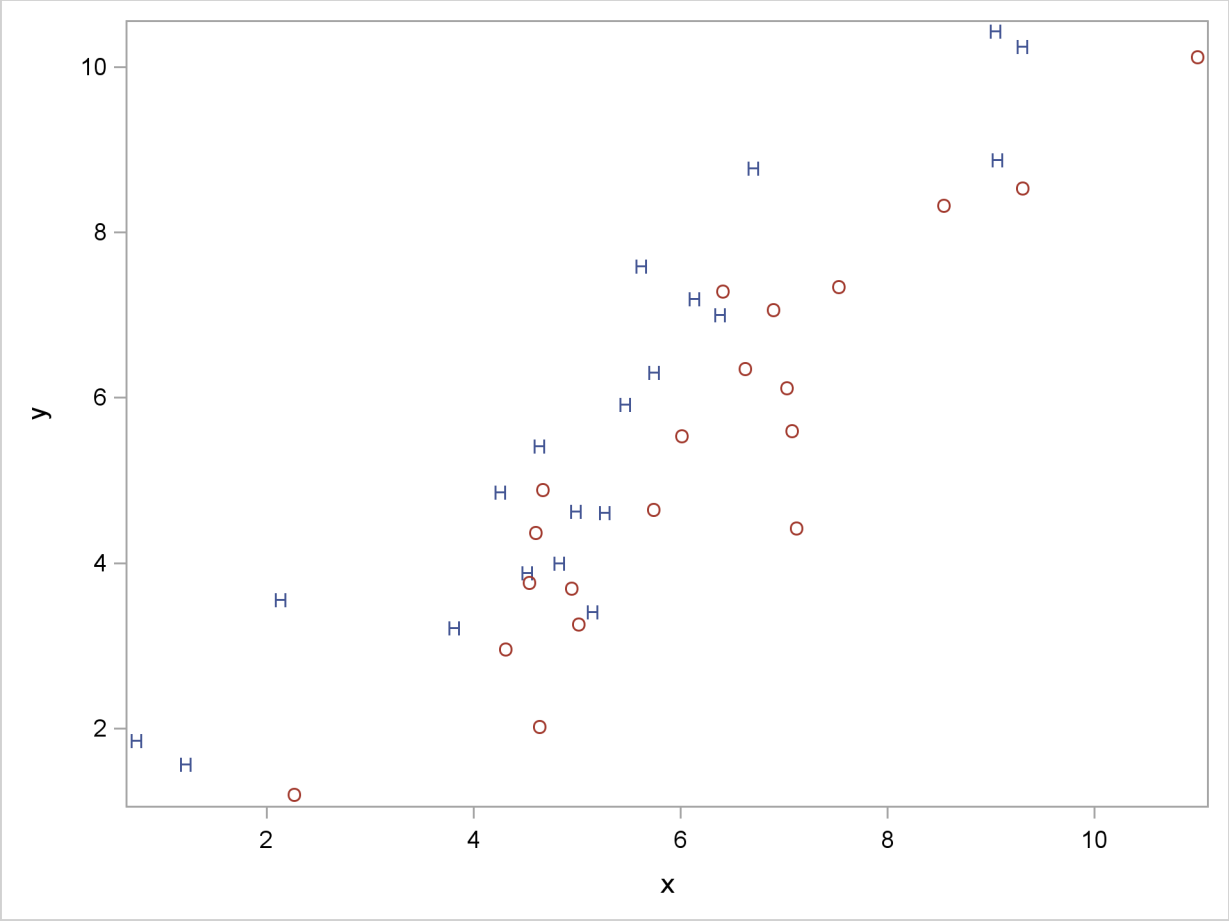
Example: Contrasting Univariate and Multivariate Analyses

Consider an artificial data set with two classes of observations indicated by 'H' and 'O'. The following statements generate and plot the data:

```
data random;
   drop n;

   Group = 'H';
   do n = 1 to 20;
      x = 4.5 + 2 * normal(57391);
      y = x + .5 + normal(57391);
      output;
   end;

   Group = 'O';
   do n = 1 to 20;
      x = 6.25 + 2 * normal(57391);
      y = x - 1 + normal(57391);
      output;
   end;

run;

proc sgplot noautolegend;
   scatter y=y x=x / markerchar=group group=group;
run;
```

The plot is shown in Figure 10.1.

Figure 10.1 Groups for Contrasting Univariate and Multivariate Analyses



The following statements perform a canonical discriminant analysis and display the results in Figure 10.2:

```
proc candisc anova;
   class Group;
   var x y;
run;
```

Figure 10.2 Contrasting Univariate and Multivariate Analyses

The CANDISC Procedure

Total Sample Size	40	DF Total	39
Variables	2	DF Within Classes	38
Classes	2	DF Between Classes	1

Number of Observations Read	40
Number of Observations Used	40

Figure 10.2 *continued*

Class Level Information

Group	Variable Name	Frequency	Weight	Proportion
H	H	20	20.0000	0.500000
O	O	20	20.0000	0.500000

The CANDISC Procedure

Univariate Test Statistics

F Statistics, Num DF=1, Den DF=38

Variable	Total Standard Deviation	Pooled Standard Deviation	Between Standard Deviation	R-Square	R-Square / (1-RSq)	F Value	Pr > F
x	2.1776	2.1498	0.6820	0.0503	0.0530	2.01	0.1641
y	2.4215	2.4486	0.2047	0.0037	0.0037	0.14	0.7105

Average R-Square

Unweighted	0.0269868
Weighted by Variance	0.0245201

Multivariate Statistics and Exact F Statistics

S=1 M=0 N=17.5

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.64203704	10.31	2	37	0.0003
Pillai's Trace	0.35796296	10.31	2	37	0.0003
Hotelling-Lawley Trace	0.55754252	10.31	2	37	0.0003
Roy's Greatest Root	0.55754252	10.31	2	37	0.0003

Figure 10.2 *continued*

```
                    The CANDISC Procedure

                          Adjusted    Approximate        Squared
          Canonical      Canonical       Standard      Canonical
        Correlation    Correlation          Error    Correlation

1          0.598300       0.589467       0.102808       0.357963

                        Eigenvalues of Inv(E)*H
                          = CanRsq/(1-CanRsq)

          Eigenvalue    Difference    Proportion    Cumulative

  1          0.5575                      1.0000        1.0000

        Test of H0: The canonical correlations in the
          current row and all that follow are zero

        Likelihood    Approximate
             Ratio        F Value    Num DF    Den DF    Pr > F

1       0.64203704          10.31         2        37    0.0003

              NOTE: The F statistic is exact.

                    The CANDISC Procedure

                  Total Canonical Structure

                  Variable              Can1

                  x                -0.374883
                  y                 0.101206

                  Between Canonical Structure

                  Variable              Can1

                  x                -1.000000
                  y                 1.000000

              Pooled Within Canonical Structure

                  Variable              Can1

                  x                -0.308237
                  y                 0.081243
```

Figure 10.2 *continued*

```
                    The CANDISC Procedure

          Total-Sample Standardized Canonical Coefficients

                Variable              Can1

                x             -2.625596855
                y              2.446680169

       Pooled Within-Class Standardized Canonical Coefficients

                Variable              Can1

                x             -2.592150014
                y              2.474116072

                Raw Canonical Coefficients

                Variable              Can1

                x             -1.205756217
                y              1.010412967

              Class Means on Canonical Variables

                 Group              Can1

                 H           0.7277811475
                 O           -.7277811475
```

The univariate R squares are very small, 0.0503 for x and 0.0037 for y, and neither variable shows a significant difference between the classes at the 0.10 level.

The multivariate test for differences between the classes is significant at the 0.0003 level. Thus, the multivariate analysis has found a highly significant difference, whereas the univariate analyses failed to achieve even the 0.10 level. The raw canonical coefficients for the first canonical variable, Can1, show that the classes differ most widely on the linear combination -1.205756217 x + 1.010412967 y or approximately y - 1.2 x. The R square between Can1 and the class variable is 0.357963 as given by the squared canonical correlation, which is much higher than either univariate R square.

In this example, the variables are highly correlated within classes. If the within-class correlation were smaller, there would be greater agreement between the univariate and multivariate analyses.

References

Cooley, W. W. and Lohnes, P. R. (1971), *Multivariate Data Analysis*, New York: John Wiley & Sons.

Efron, B. (1975), "The Efficiency of Logistic Regression Compared to Normal Discriminant Analysis," *Journal of the American Statistical Association*, 70, 892–898.

Fisher, R. A. (1936), "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, 7, 179–188.

Gnanadesikan, R. (1977), *Methods for Statistical Data Analysis of Multivariate Observations*, New York: John Wiley & Sons.

Hand, D. J. (1981), *Discrimination and Classification*, New York: John Wiley & Sons.

Hand, D. J. (1982), *Kernel Discriminant Analysis*, New York: Research Studies Press.

Klecka, W. R. (1980), *Discriminant Analysis*, Sage University Paper Series on Quantitative Applications in the Social Sciences, 07-019, Beverly Hills and London: Sage Publications.

Kshirsagar, A. M. (1972), *Multivariate Analysis*, New York: Marcel Dekker.

Lachenbruch, P. A. (1975), *Discriminant Analysis*, New York: Hafner Publishing.

Lachenbruch, P. A. (1979), "Discriminant Analysis," *Biometrics*.

Press, S. J. and Wilson, S. (1978), "Choosing between Logistic Regression and Discriminant Analysis," *Journal of the American Statistical Association*, 73, 699–705.

Silverman, B. W. (1986), *Density Estimation*, New York: Chapman & Hall.

Tatsuoka, M. M. (1971), *Multivariate Analysis*, New York: John Wiley & Sons.

Chapter 11

Introduction to Clustering Procedures

Contents

Overview: Clustering Procedures **209**
Clustering Variables **211**
Clustering Observations **212**
Methods for Clustering Observations **213**
Well-Separated Clusters 213
Poorly Separated Clusters 215
Multinormal Clusters of Unequal Size and Dispersion 223
Elongated Multinormal Clusters 233
Nonconvex Clusters 240
The Number of Clusters **243**
References **246**

Overview: Clustering Procedures

You can use SAS clustering procedures to cluster the observations or the variables in a SAS data set. Both hierarchical and disjoint clusters can be obtained. Only numeric variables can be analyzed directly by the procedures, although the DISTANCE procedure can compute a distance matrix that uses character or numeric variables.

The purpose of cluster analysis is to place objects into groups, or clusters, suggested by the data, not defined a priori, such that objects in a given cluster tend to be similar to each other in some sense, and objects in different clusters tend to be dissimilar. You can also use cluster analysis to summarize data rather than to find "natural" or "real" clusters; this use of clustering is sometimes called *dissection* (Everitt 1980).

Any generalization about cluster analysis must be vague because a vast number of clustering methods have been developed in several different fields, with different definitions of clusters and similarity among objects. The variety of clustering techniques is reflected by the variety of terms used for cluster analysis: botryology, classification, clumping, competitive learning, morphometrics, nosography, nosology, numerical taxonomy, partitioning, Q-analysis, systematics, taximetrics, taxonorics, typology, unsupervised pattern recognition, vector quantization, and winner-take-all learning. Good (1977) has also suggested aciniformics and agminatics.

Several types of clusters are possible:

- Disjoint clusters place each object in one and only one cluster.
- Hierarchical clusters are organized so that one cluster can be entirely contained within another cluster, but no other kind of overlap between clusters is allowed.
- Overlapping clusters can be constrained to limit the number of objects that belong simultaneously to two clusters, or they can be unconstrained, allowing any degree of overlap in cluster membership.
- Fuzzy clusters are defined by a probability or grade of membership of each object in each cluster. Fuzzy clusters can be disjoint, hierarchical, or overlapping.

The data representations of objects to be clustered also take many forms. The most common are as follows:

- a square distance or similarity matrix, in which both rows and columns correspond to the objects to be clustered. A correlation matrix is an example of a similarity matrix.
- a coordinate matrix, in which the rows are observations and the columns are variables, as in the usual SAS multivariate data set. The observations, the variables, or both can be clustered.

The SAS procedures for clustering are oriented toward disjoint or hierarchical clusters from coordinate data, distance data, or a correlation or covariance matrix. The following procedures are used for clustering:

CLUSTER	performs hierarchical clustering of observations by using eleven agglomerative methods applied to coordinate data or distance data.
FASTCLUS	finds disjoint clusters of observations by using a k-means method applied to coordinate data. PROC FASTCLUS is especially suitable for large data sets.
MODECLUS	finds disjoint clusters of observations with coordinate or distance data by using nonparametric density estimation. It can also perform approximate nonparametric significance tests for the number of clusters.
VARCLUS	performs both hierarchical and disjoint clustering of variables by using oblique multiple-group component analysis.
TREE	draws tree diagrams, also called *dendrograms* or *phenograms*, by using output from the CLUSTER or VARCLUS procedure. PROC TREE can also create a data set indicating cluster membership at any specified level of the cluster tree.

The following procedures are useful for processing data prior to the actual cluster analysis:

ACECLUS	attempts to estimate the pooled within-cluster covariance matrix from coordinate data without knowledge of the number or the membership of the clusters (Art, Gnanadesikan, and Kettenring 1982). PROC ACECLUS outputs a data set containing canonical variable scores to be used in the cluster analysis proper.

DISTANCE computes various measures of distance, dissimilarity, or similarity between the observations (rows) of a SAS data set. PROC DISTANCE also provides various nonparametric and parametric methods for standardizing variables. Different variables can be standardized with different methods.

PRINCOMP performs a principal component analysis and outputs principal component scores.

STDIZE standardizes variables by using any of a variety of location and scale measures, including mean and standard deviation, minimum and range, median and absolute deviation from the median, various M-estimators and A-estimators, and some scale estimators designed specifically for cluster analysis.

Massart and Kaufman (1983) is the best elementary introduction to cluster analysis. Other important texts are Anderberg (1973), Sneath and Sokal (1973), Duran and Odell (1974), Hartigan (1975), Titterington, Smith, and Makov (1985), McLachlan and Basford (1988), and Kaufmann and Rousseeuw (1990). Hartigan (1975) and Spath (1980) give numerous FORTRAN programs for clustering. Any prospective user of cluster analysis should study the Monte Carlo results of Milligan (1980), Milligan and Cooper (1985), and Cooper and Milligan (1988). Important references on the statistical aspects of clustering include MacQueen (1967), Wolfe (1970), Scott and Symons (1971), Hartigan (1977, 1978, 1981, 1985), Symons (1981), Everitt (1981), Sarle (1983), Bock (1985), and Thode, Mendell, and Finch (1988). Bayesian methods have important advantages over maximum likelihood; see Binder (1978, 1981), Banfield and Raftery (1993), and Bensmail et al. (1997). For fuzzy clustering, see Bezdek (1981) and Bezdek and Pal (1992). The signal-processing perspective is provided by Gersho and Gray (1992). See Blashfield and Aldenderfer (1978) for a discussion of the fragmented state of the literature on cluster analysis.

Clustering Variables

Factor rotation is often used to cluster variables, but the resulting clusters are fuzzy. It is preferable to use PROC VARCLUS if you want hard (nonfuzzy), disjoint clusters. Factor rotation is better if you want to be able to find overlapping clusters. It is often a good idea to try both PROC VARCLUS and PROC FACTOR with an oblique rotation, compare the amount of variance explained by each, and see how fuzzy the factor loadings are and whether there seem to be overlapping clusters.

You can use PROC VARCLUS to harden a fuzzy factor rotation; use PROC FACTOR to create an output data set containing scoring coefficients and initialize PROC VARCLUS with this data set as follows:

```
proc factor rotate=promax score outstat=fact;
run;

proc varclus initial=input proportion=0;
run;
```

You can use any rotation method instead of the PROMAX method. The SCORE and OUTSTAT= options are necessary in the PROC FACTOR statement. PROC VARCLUS reads the correlation

matrix from the data set created by PROC FACTOR. The INITIAL=INPUT option tells PROC VARCLUS to read initial scoring coefficients from the data set. The option PROPORTION=0 keeps PROC VARCLUS from splitting any of the clusters.

Clustering Observations

PROC CLUSTER is easier to use than PROC FASTCLUS because one run produces results from one cluster up to as many as you like. You must run PROC FASTCLUS once for each number of clusters.

The time required by PROC FASTCLUS is roughly proportional to the number of observations, whereas the time required by PROC CLUSTER with most methods varies with the square or cube of the number of observations. Therefore, you can use PROC FASTCLUS with much larger data sets than PROC CLUSTER.

If you want to hierarchically cluster a data set that is too large to use with PROC CLUSTER directly, you can have PROC FASTCLUS produce, for example, 50 clusters, and let PROC CLUSTER analyze these 50 clusters instead of the entire data set. The MEAN= data set produced by PROC FASTCLUS contains two special variables:

- The variable _FREQ_ gives the number of observations in the cluster.
- The variable _RMSSTD_ gives the root mean square across variables of the cluster standard deviations.

These variables are automatically used by PROC CLUSTER to give the correct results when clustering clusters. For example, you could specify Ward's minimum variance method (Ward 1963):

```
proc fastclus maxclusters=50 mean=temp;
   var x y z;
run;

proc cluster method=ward outtree=tree;
   var x y z;
run;
```

Or you could specify Wong's hybrid method (Wong 1982):

```
proc fastclus maxclusters=50 mean=temp;
   var x y z;
run;

proc cluster method=density hybrid outtree=tree;
   var x y z;
run;
```

More detailed examples are given in Chapter 29, "The CLUSTER Procedure."

Characteristics of Methods for Clustering Observations

Many simulation studies comparing various methods of cluster analysis have been performed. In these studies, artificial data sets containing known clusters are produced using pseudo-random-number generators. The data sets are analyzed by a variety of clustering methods, and the degree to which each clustering method recovers the known cluster structure is evaluated. See Milligan (1981) for a review of such studies. In most of these studies, the clustering method with the best overall performance has been either average linkage or Ward's minimum variance method. The method with the poorest overall performance has almost invariably been single linkage. However, in many respects, the results of simulation studies are inconsistent and confusing.

When you attempt to evaluate clustering methods, it is essential to realize that most methods are biased toward finding clusters possessing certain characteristics related to size (number of members), shape, or dispersion. Methods based on the least squares criterion (Sarle 1982), such as k-means and Ward's minimum variance method, tend to find clusters with roughly the same number of observations in each cluster. Average linkage is somewhat biased toward finding clusters of equal variance. Many clustering methods tend to produce compact, roughly hyperspherical clusters and are incapable of detecting clusters with highly elongated or irregular shapes. The methods with the least bias are those based on nonparametric density estimation such as single linkage and density linkage.

Most simulation studies have generated compact (often multivariate normal) clusters of roughly equal size or dispersion. Such studies naturally favor average linkage and Ward's method over most other hierarchical methods, especially single linkage. It would be easy, however, to design a study that uses elongated or irregular clusters in which single linkage would perform much better than average linkage or Ward's method (see some of the following examples). Even studies that compare clustering methods that use "realistic" data might unfairly favor particular methods. For example, in all the data sets used by Mezzich and Solomon (1980), the clusters established by field experts are of equal size. When interpreting simulation or other comparative studies, you must, therefore, decide whether the artificially generated clusters in the study resemble the clusters you suspect might exist in your data in terms of size, shape, and dispersion. If, like many people doing exploratory cluster analysis, you have no idea what kinds of clusters to expect, you should include at least one of the relatively unbiased methods, such as density linkage, in your analysis.

The rest of this section consists of a series of examples that illustrate the performance of various clustering methods under various conditions. The first, and simplest, example shows a case of well-separated clusters. The other examples show cases of poorly separated clusters, clusters of unequal size, parallel elongated clusters, and nonconvex clusters.

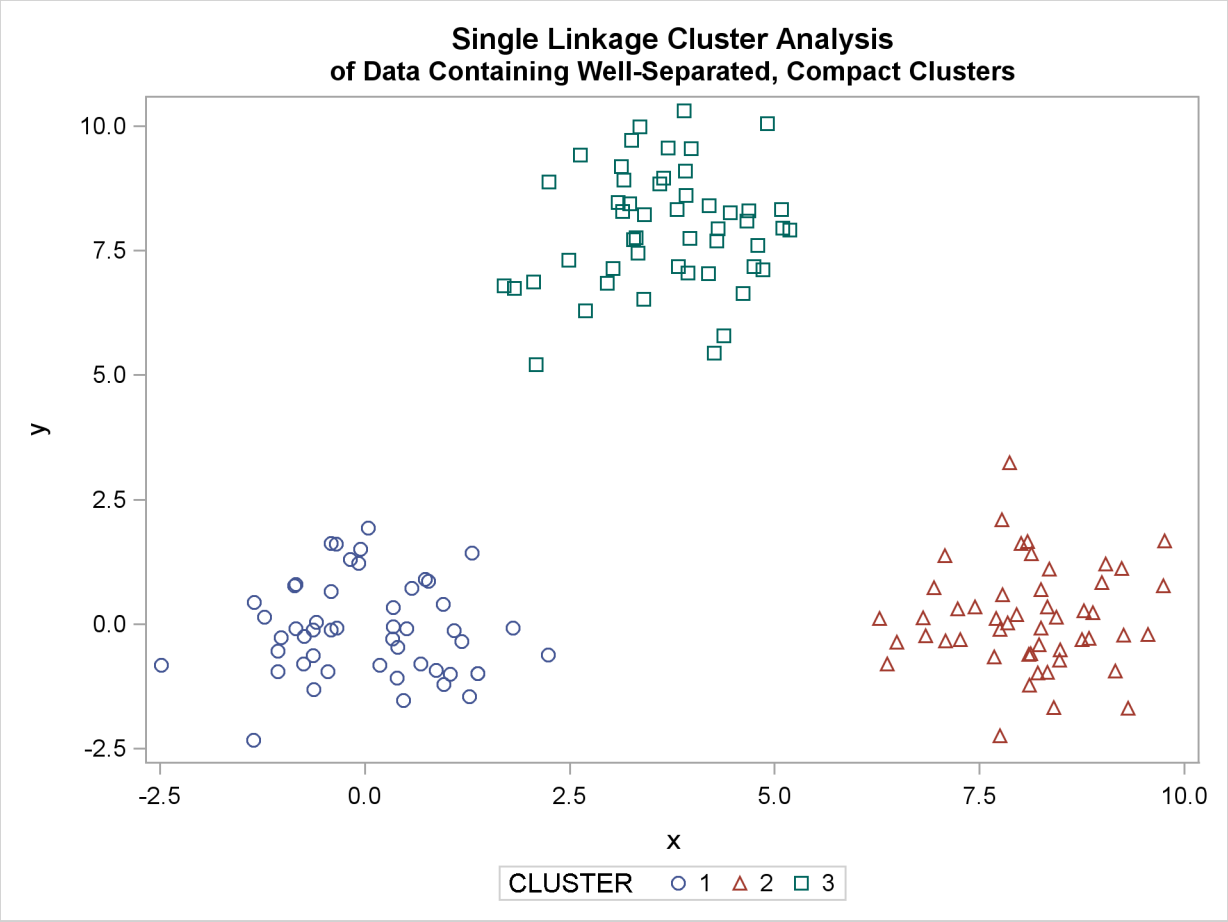
Well-Separated Clusters

If the population clusters are sufficiently well separated, almost any clustering method performs well, as demonstrated in the following example, which uses single linkage. In this and subsequent examples, the output from the clustering procedures is not shown, but cluster membership is dis-

played in scatter plots. The SAS autocall macro MODSTYLE is specified to change the default marker symbols for the plot. For more information about autocall libraries, see *SAS Macro Language: Reference*. The following SAS statements produce Figure 11.1:

```
data compact;
   keep x y;
   n=50; scale=1;
   mx=0; my=0; link generate;
   mx=8; my=0; link generate;
   mx=4; my=8; link generate;
   stop;
generate:
   do i=1 to n;
      x=rannor(1)*scale+mx;
      y=rannor(1)*scale+my;
      output;
   end;
   return;
   run;
proc cluster data=compact outtree=tree
             method=single noprint;
run;

proc tree noprint out=out n=3;
   copy x y;
run;

%modstyle(name=ClusterStyle,parent=Statistical,type=CLM,
markers=Circle Triangle Square circlefilled);
ods listing style=ClusterStyle;

proc sgplot;
   scatter y=y x=x / group=cluster;
   title 'Single Linkage Cluster Analysis';
   title2 'of Data Containing Well-Separated, Compact Clusters';
run;
```

Figure 11.1 Data Containing Well-Separated, Compact Clusters: PROC CLUSTER with METHOD=SINGLE and PROC SGPLOT



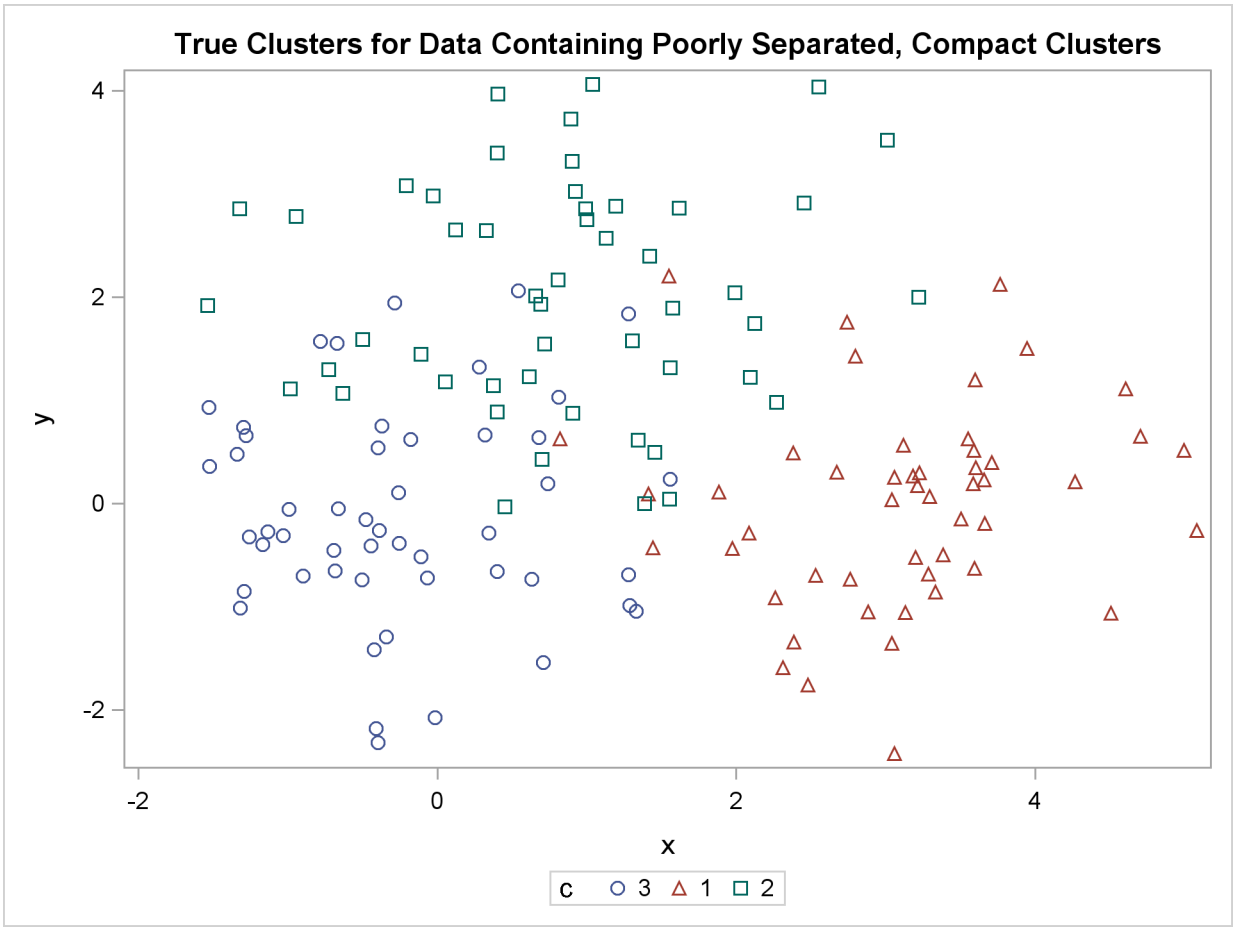
Poorly Separated Clusters

To see how various clustering methods differ, you must examine a more difficult problem than that of the previous example.

The following data set is similar to the first except that the three clusters are much closer together. This example demonstrates the use of PROC FASTCLUS and five hierarchical methods available in PROC CLUSTER. To help you compare methods, this example plots true, generated clusters. Also included is a bubble plot of the density estimates obtained in conjunction with two-stage density linkage in PROC CLUSTER. The following SAS statements produce Figure 11.2:

```
data closer;
   keep x y c;
   n=50; scale=1;
   mx=0; my=0; c=3; link generate;
   mx=3; my=0; c=1; link generate;
   mx=1; my=2; c=2; link generate;
   stop;
generate:
   do i=1 to n;
      x=rannor(9)*scale+mx;
      y=rannor(9)*scale+my;
      output;
   end;
   return;
   run;
title 'True Clusters for Data Containing Poorly Separated, Compact Clusters';
proc sgplot;
   scatter y=y x=x / group=c ;
run;
```

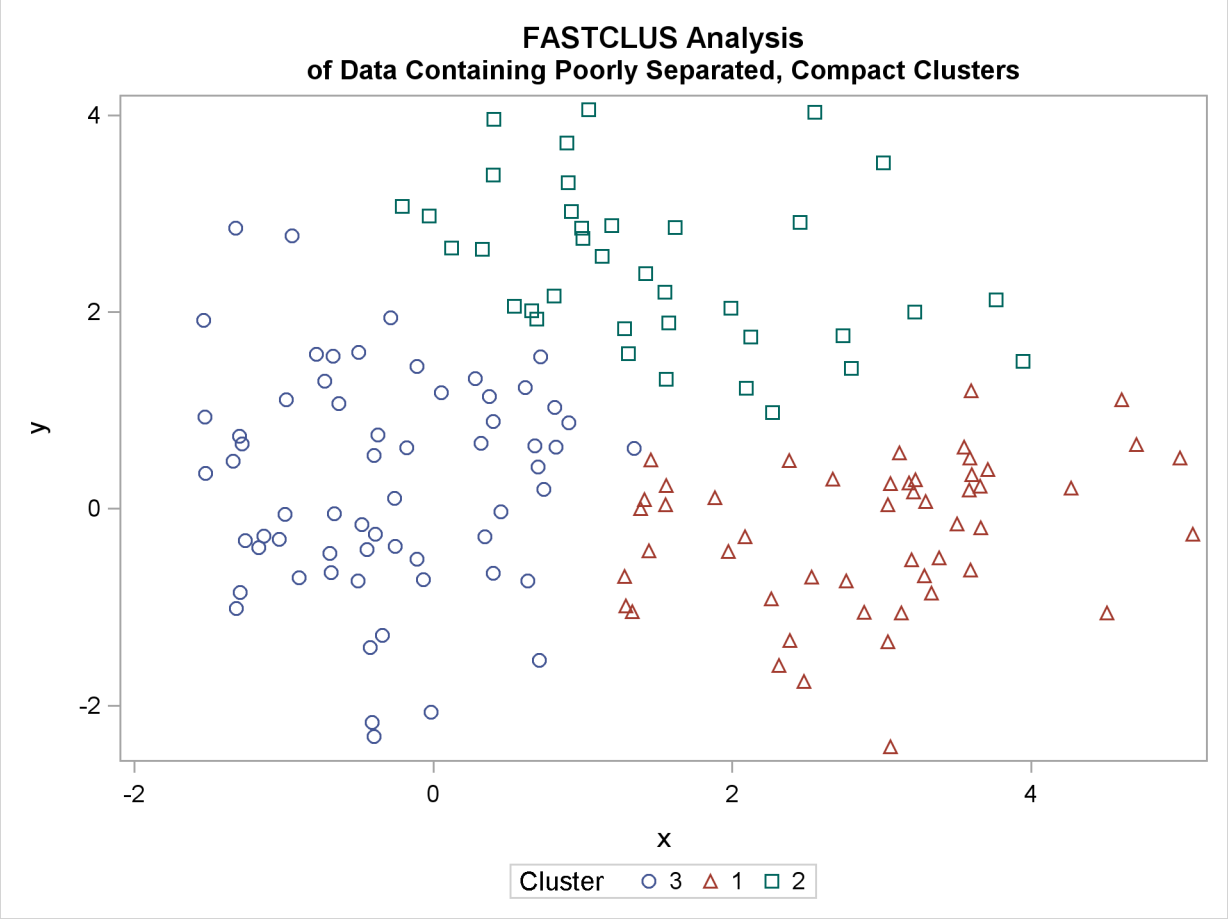
Figure 11.2 Data Containing Poorly Separated, Compact Clusters: Plot of True Clusters



The following statements use the FASTCLUS procedure to find three clusters and then use the SGPLOT procedure to plot the clusters. The following statements produce Figure 11.3:

```
proc fastclus data=closer out=out maxc=3 noprint;
   var x y;
   title 'FASTCLUS Analysis';
   title2 'of Data Containing Poorly Separated, Compact Clusters';
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
run;
```

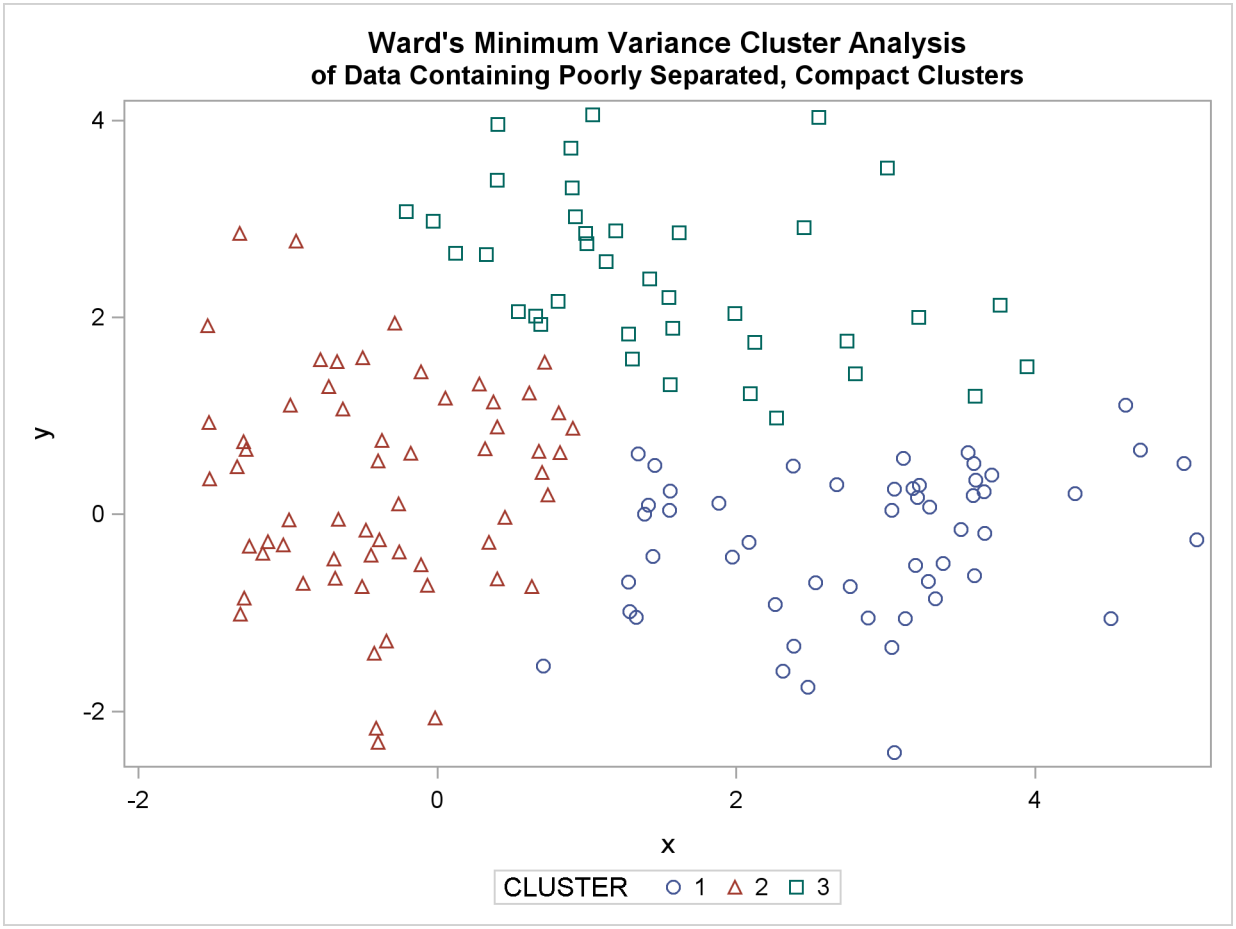
Figure 11.3 Data Containing Poorly Separated, Compact Clusters: PROC FASTCLUS



The following SAS statements produce Figure 11.4:

```
proc cluster data=closer outtree=tree method=ward noprint;
   var x y;
run;

proc tree noprint out=out n=3;
   copy x y;
   title 'Ward''s Minimum Variance Cluster Analysis';
   title2 'of Data Containing Poorly Separated, Compact Clusters';
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
run;
```

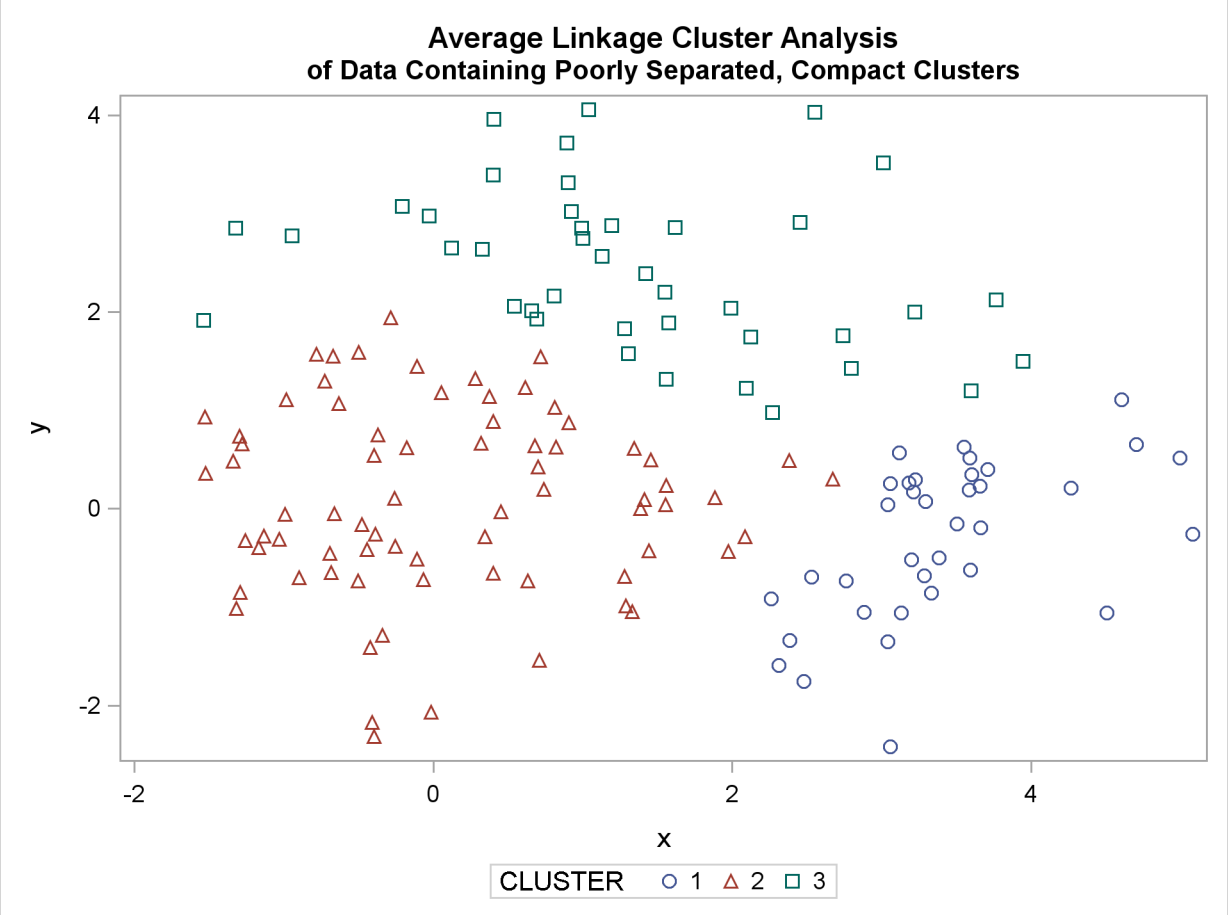
Figure 11.4 Data Containing Poorly Separated, Compact Clusters: PROC CLUSTER with METHOD=WARD



The following SAS statements produce Figure 11.5:

```
proc cluster data=closer outtree=tree method=average noprint;
   var x y;
run;

proc tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Average Linkage Cluster Analysis';
   title2 'of Data Containing Poorly Separated, Compact Clusters';
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
run;
```

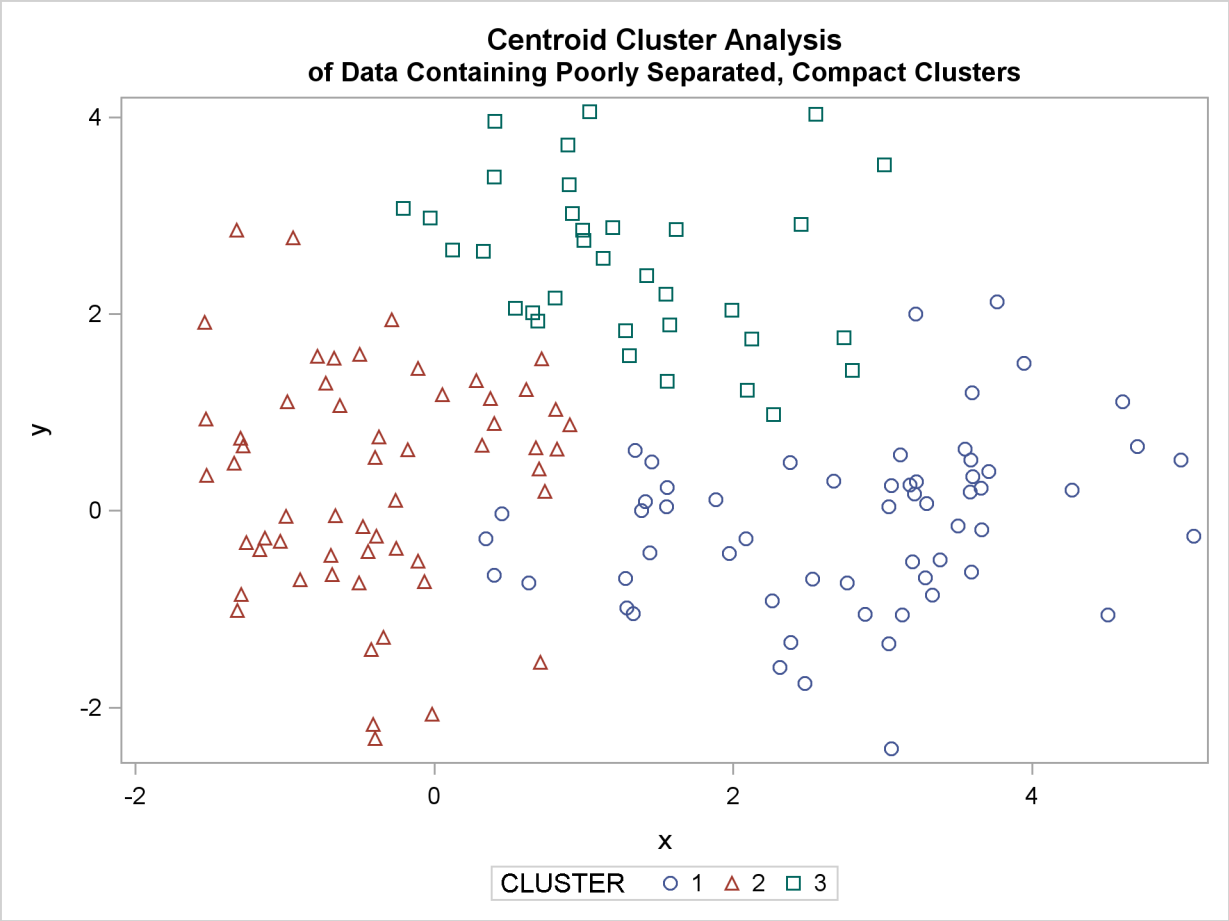
Figure 11.5 Data Containing Poorly Separated, Compact Clusters: PROC CLUSTER with METHOD=AVERAGE



The following SAS statements produce Figure 11.6:

```
proc cluster data=closer outtree=tree
             method=centroid noprint;
   var x y;
run;

proc tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Centroid Cluster Analysis';
   title2 'of Data Containing Poorly Separated, Compact Clusters';
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
run;
```

Figure 11.6 Data Containing Poorly Separated, Compact Clusters: PROC CLUSTER with METHOD=CENTROID



The following SAS statements produce Figure 11.7 and Figure 11.8:

```
proc cluster data=closer outtree=tree
             method=twostage k=10 noprint;
   var x y;
```

```
run;

proc tree noprint out=out n=3;
   copy x y _dens_;
   title 'Two-Stage Density Linkage Cluster Analysis';
   title2 'of Data Containing Poorly Separated, Compact Clusters';
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
run;

axis1 minor=none label=(angle=90 rotate=0);
axis2 minor=none;

proc gplot;
   bubble y*x=_dens_/frame vaxis=axis1 haxis=axis2 bsize=10;
   title h=1.2 'Estimated Densities';
   title2 h=1 'for Data Containing Poorly Separated, Compact Clusters';
run;
```

Figure 11.7 Data Containing Poorly Separated, Compact Clusters: PROC CLUSTER with METHOD=TWOSTAGE

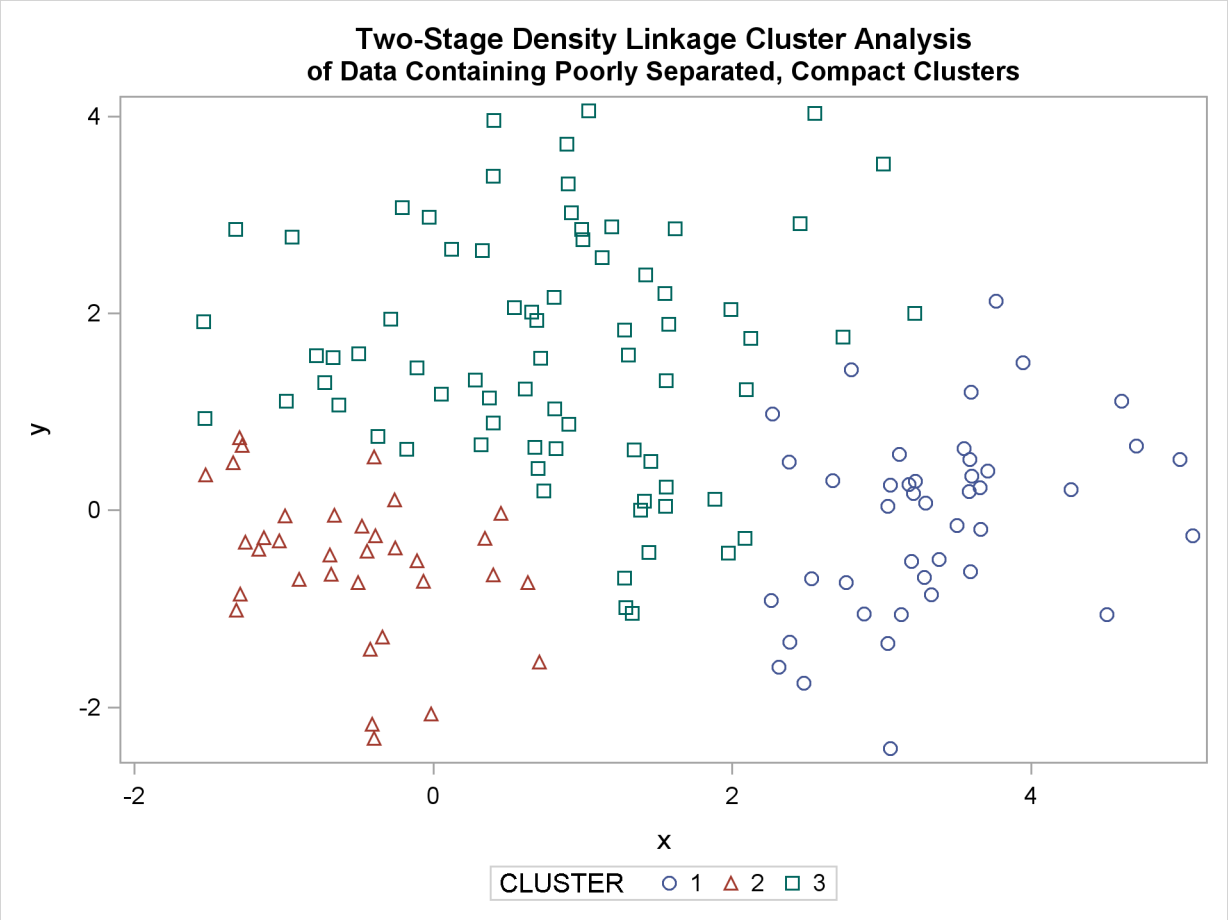
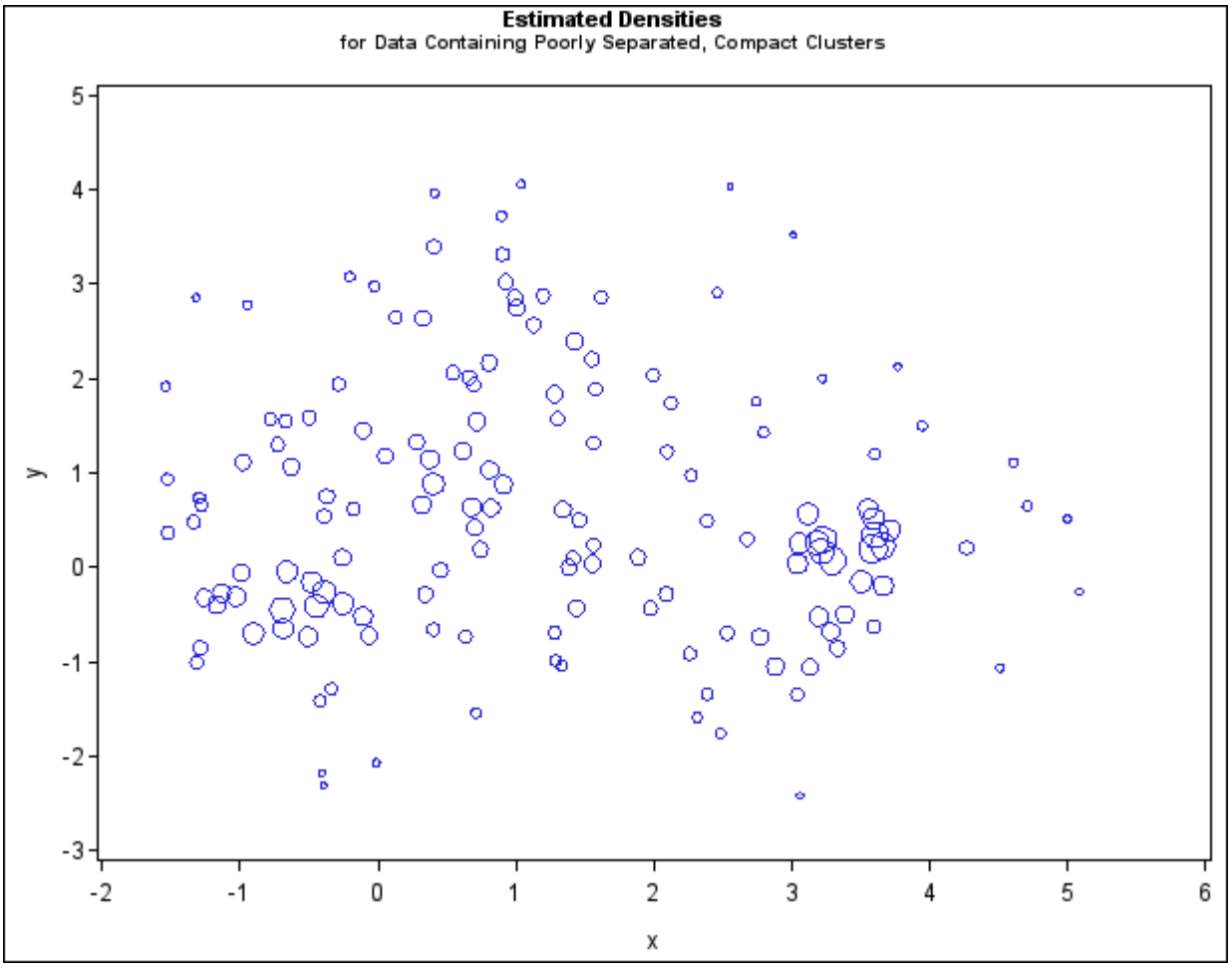


Figure 11.8 Data Containing Poorly Separated, Compact Clusters: PROC CLUSTER with METHOD=TWOSTAGE

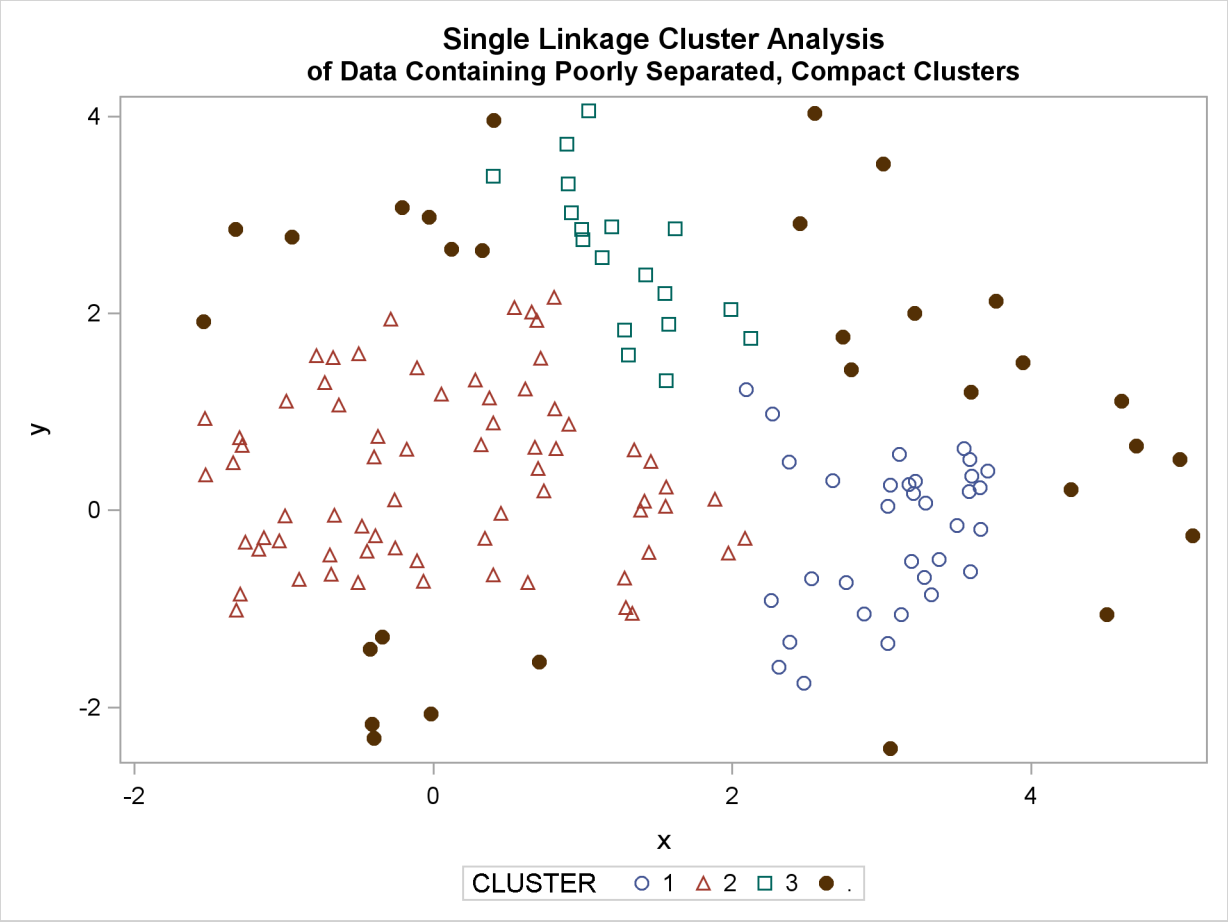


In two-stage density linkage, each cluster is a region surrounding a local maximum of the estimated probability density function. If you think of the estimated density function as a landscape with mountains and valleys, each mountain is a cluster, and the boundaries between clusters are placed near the bottoms of the valleys.

The following SAS statements produce Figure 11.9:

```
proc cluster data=closer outtree=tree
             method=single noprint;
   var x y;
run;

proc tree data=tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Single Linkage Cluster Analysis';
   title2 'of Data Containing Poorly Separated, Compact Clusters';
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
run;
```

Figure 11.9 Data Containing Poorly Separated, Compact Clusters: PROC CLUSTER with METHOD=SINGLE



The two least squares methods, PROC FASTCLUS and Ward's, yield the most uniform cluster sizes and the best recovery of the true clusters. This result is expected since these two methods are biased toward recovering compact clusters of equal size. With average linkage, the lower-left cluster is too large; with the centroid method, the lower-right cluster is too large; and with two-stage density linkage, the top cluster is too large. The single linkage analysis resembles average linkage except for the large number of outliers resulting from the DOCK= option in the PROC TREE statement; the outliers are plotted as filled circles (missing values).

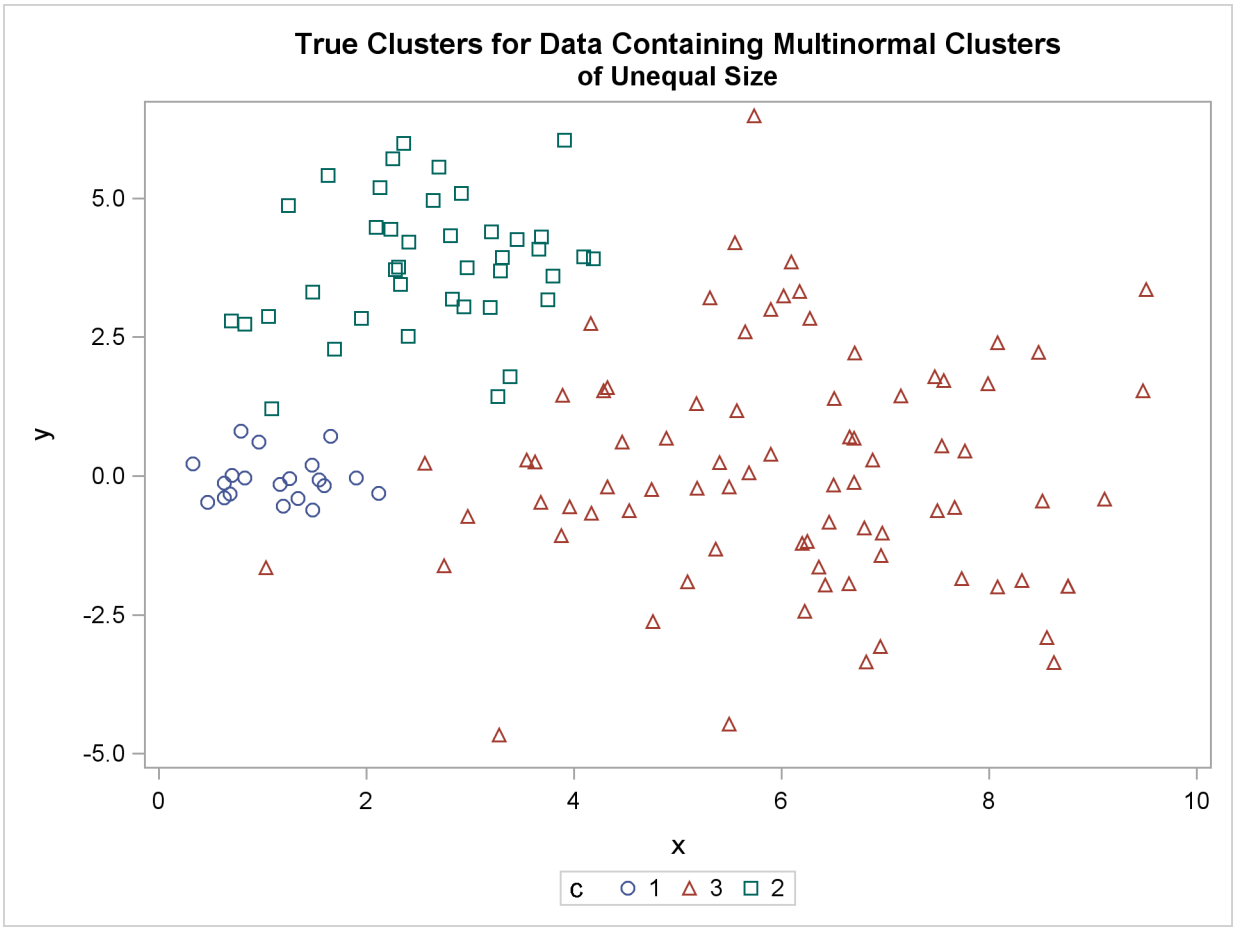
Multinormal Clusters of Unequal Size and Dispersion

In this example, there are three multinormal clusters that differ in size and dispersion. PROC FASTCLUS and five of the hierarchical methods available in PROC CLUSTER are used. To help you compare methods, the true, generated clusters are plotted.

The following SAS statements produce Figure 11.10:

```
data unequal;
   keep x y c;
   mx=1; my=0; n=20; scale=.5; c=1; link generate;
   mx=6; my=0; n=80; scale=2.; c=3; link generate;
   mx=3; my=4; n=40; scale=1.; c=2; link generate;
   stop;
generate:
   do i=1 to n;
      x=rannor(1)*scale+mx;
      y=rannor(1)*scale+my;
      output;
   end;
   return;
title 'True Clusters for Data Containing Multinormal Clusters';
title2 'of Unequal Size';
proc sgplot;
   scatter y=y x=x / group=c;
run;
```

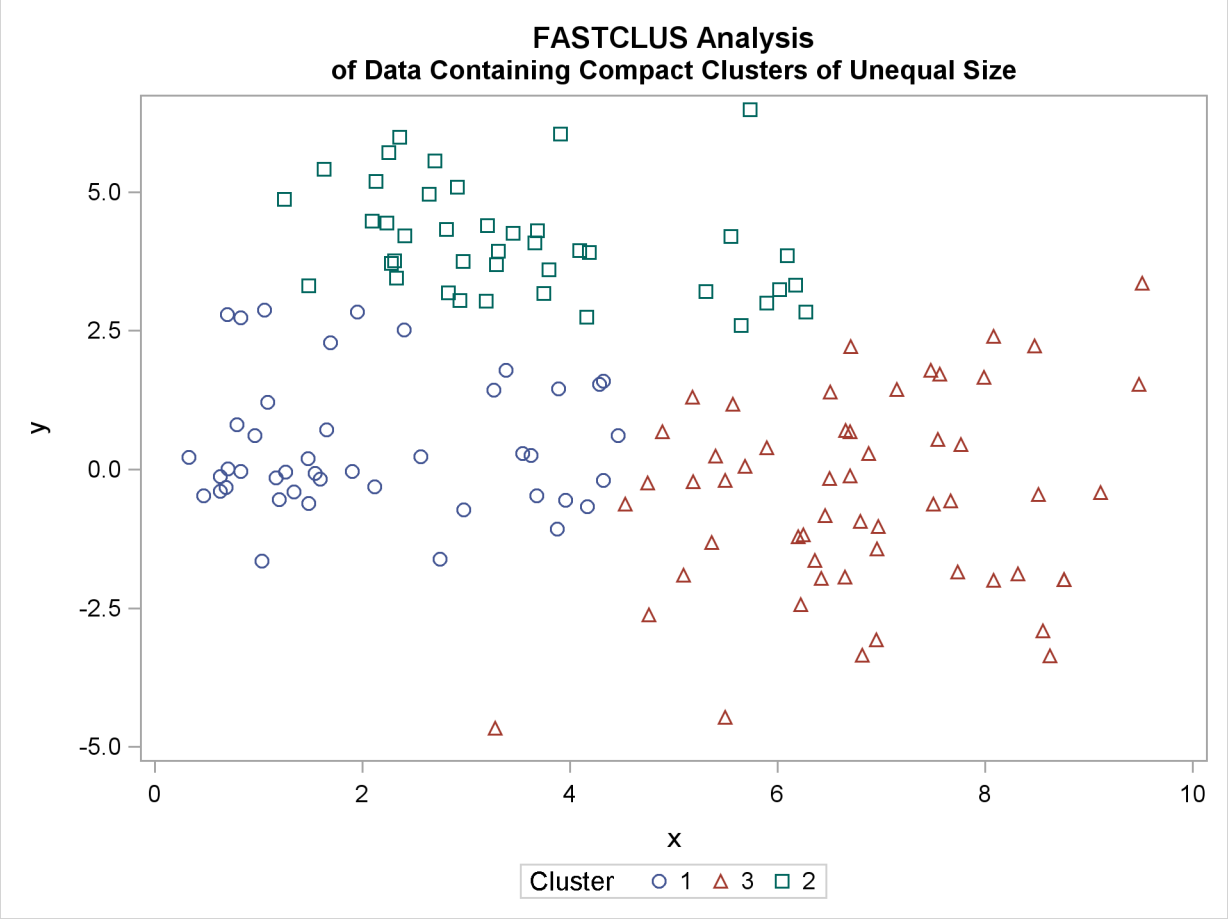
Figure 11.10 Data Containing Generated Clusters of Unequal Size



The following statements use the FASTCLUS procedure to find three clusters and then use the SGPLOT procedure to plot the clusters. The following statements produce Figure 11.11:

```
proc fastclus data=unequal out=out maxc=3 noprint;
   var x y;
   title 'FASTCLUS Analysis';
   title2 'of Data Containing Compact Clusters of Unequal Size';
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
run;
```

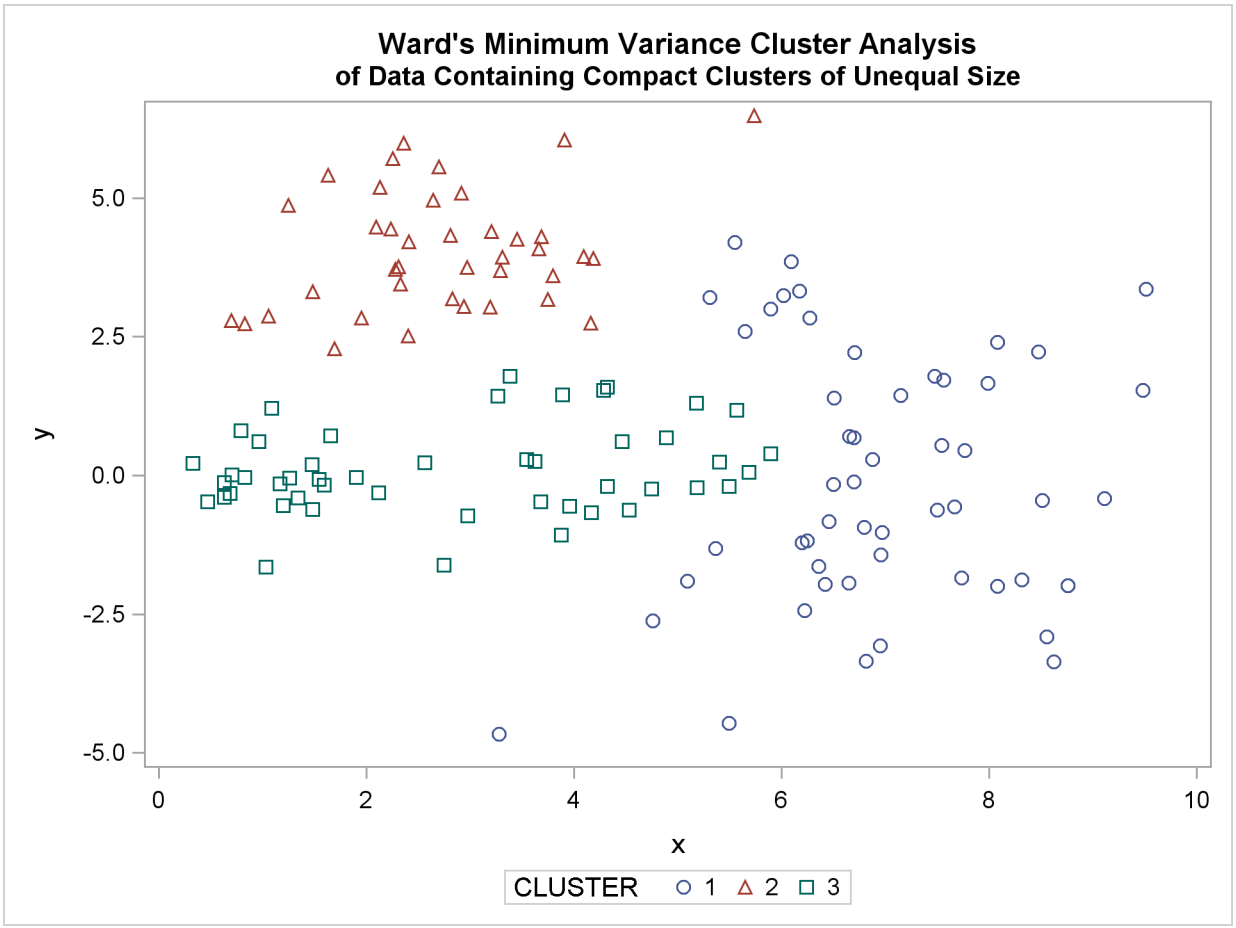
Figure 11.11 Data Containing Compact Clusters of Unequal Size: PROC FASTCLUS



The following SAS statements produce Figure 11.12:

```
proc cluster data=unequal outtree=tree
             method=ward noprint;
   var x y;
run;

proc tree noprint out=out n=3;
   copy x y;
   title 'Ward''s Minimum Variance Cluster Analysis';
   title2 'of Data Containing Compact Clusters of Unequal Size';
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
run;
```

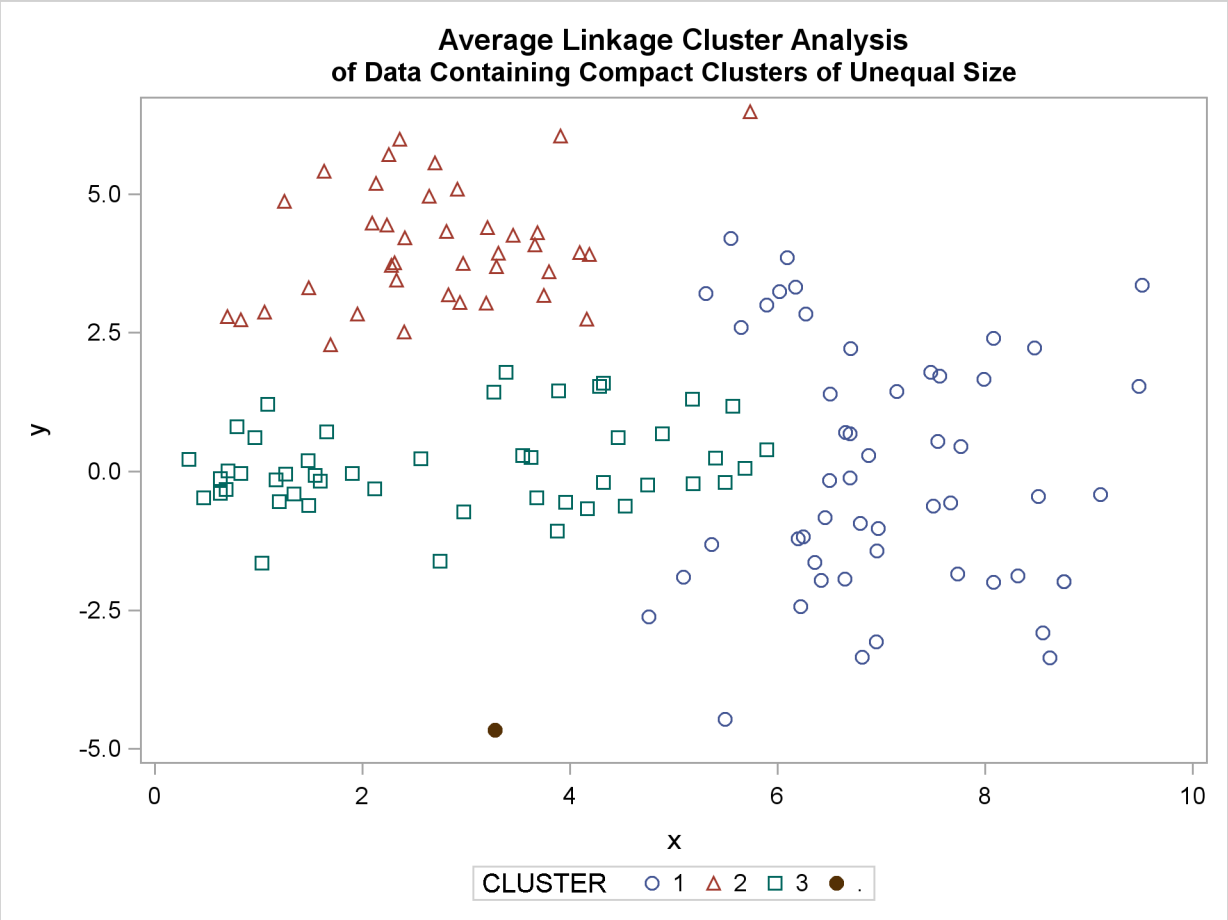
Figure 11.12 Data Containing Compact Clusters of Unequal Size: PROC CLUSTER with METHOD=WARD



The following SAS statements produce Figure 11.13:

```
proc cluster data=unequal outtree=tree method=average
             noprint;
   var x y;
run;

proc tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Average Linkage Cluster Analysis';
   title2 'of Data Containing Compact Clusters of Unequal Size';
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
run;
```

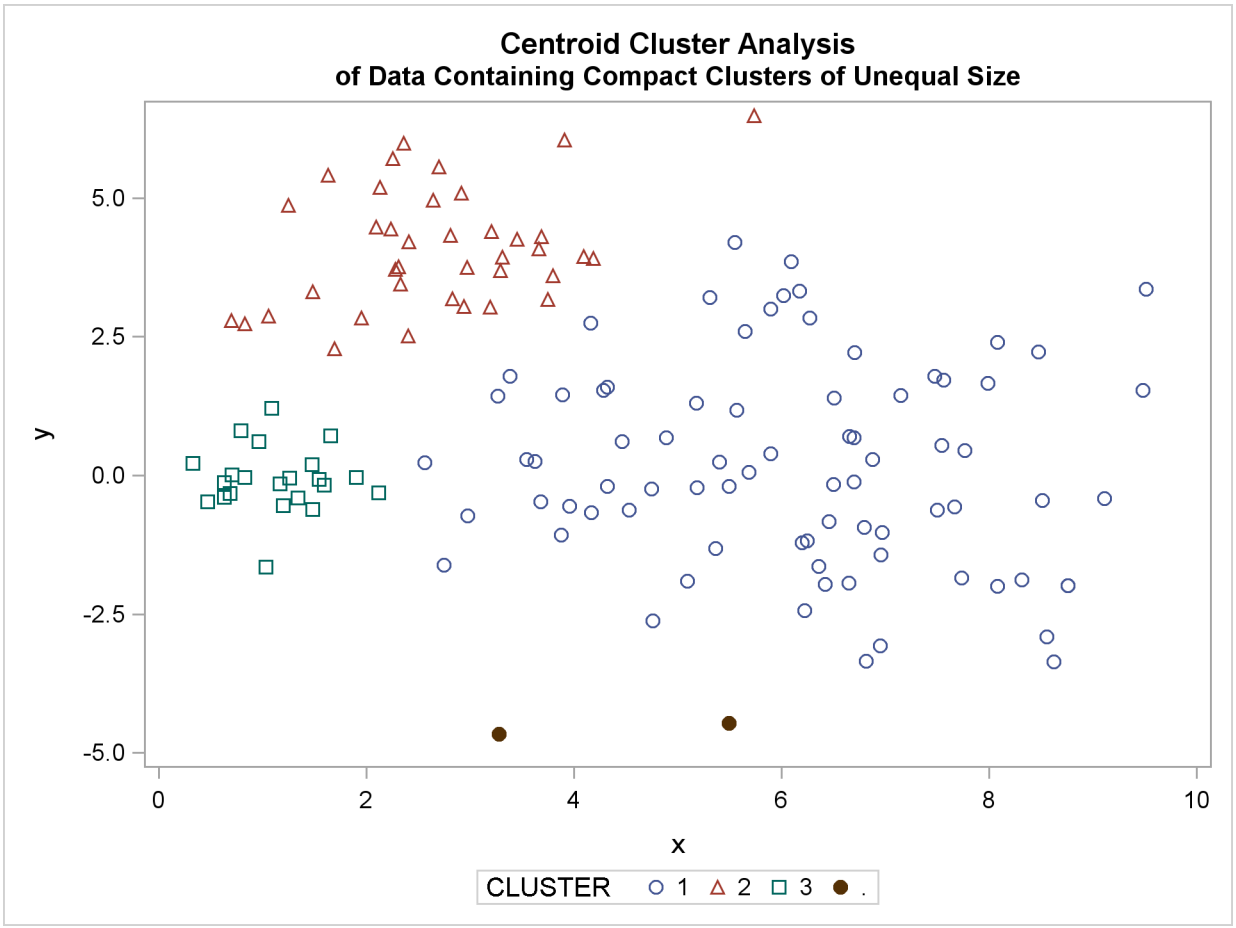
Figure 11.13 Data Containing Compact Clusters of Unequal Size: PROC CLUSTER with METHOD=AVERAGE



The following SAS statements produce Figure 11.14:

```
proc cluster data=unequal outtree=tree
             method=centroid noprint;
   var x y;
run;

proc tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Centroid Cluster Analysis';
   title2 'of Data Containing Compact Clusters of Unequal Size';
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
run;
```

Figure 11.14 Data Containing Compact Clusters of Unequal Size: PROC CLUSTER with METHOD=CENTROID



The following SAS statements produce Figure 11.15 and Figure 11.16:

```
proc cluster data=unequal outtree=tree method=twostage
             k=10 noprint;
   var x y;
run;

proc tree noprint out=out n=3;
   copy x y _dens_;
   title 'Two-Stage Density Linkage Cluster Analysis';
   title2 'of Data Containing Compact Clusters of Unequal Size';
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
run;

axis1 minor=none label=(angle=90 rotate=0);
axis2 minor=none;

proc gplot;
   bubble y*x=_dens_/frame vaxis=axis1 haxis=axis2 bsize=10;
   title h=1.2 'Estimated Densities';
   title2 h=1 'for Data Containing Compact Clusters of Unequal Size';
run;
```

Figure 11.15 Data Containing Compact Clusters of Unequal Size: PROC CLUSTER with METHOD=TWOSTAGE

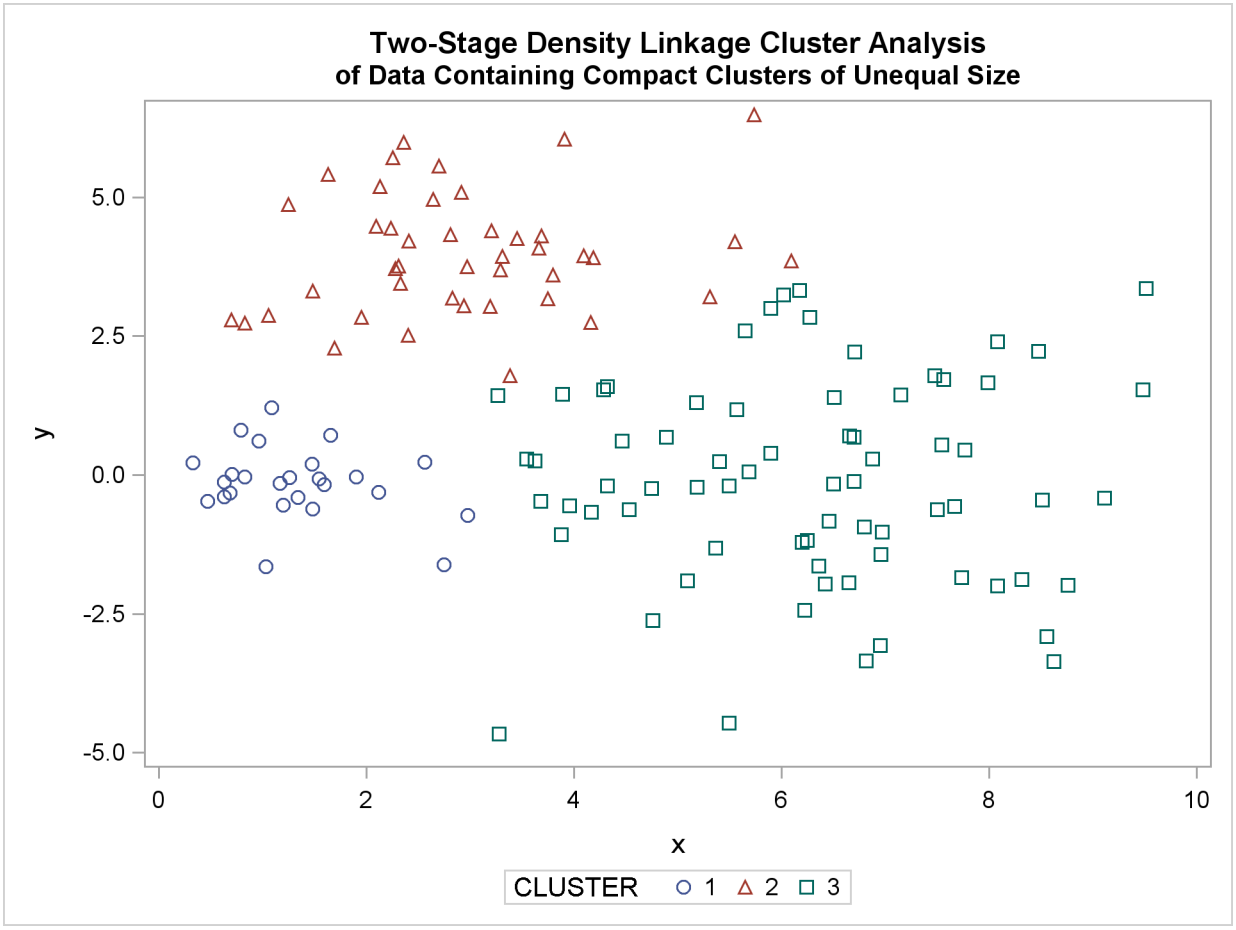
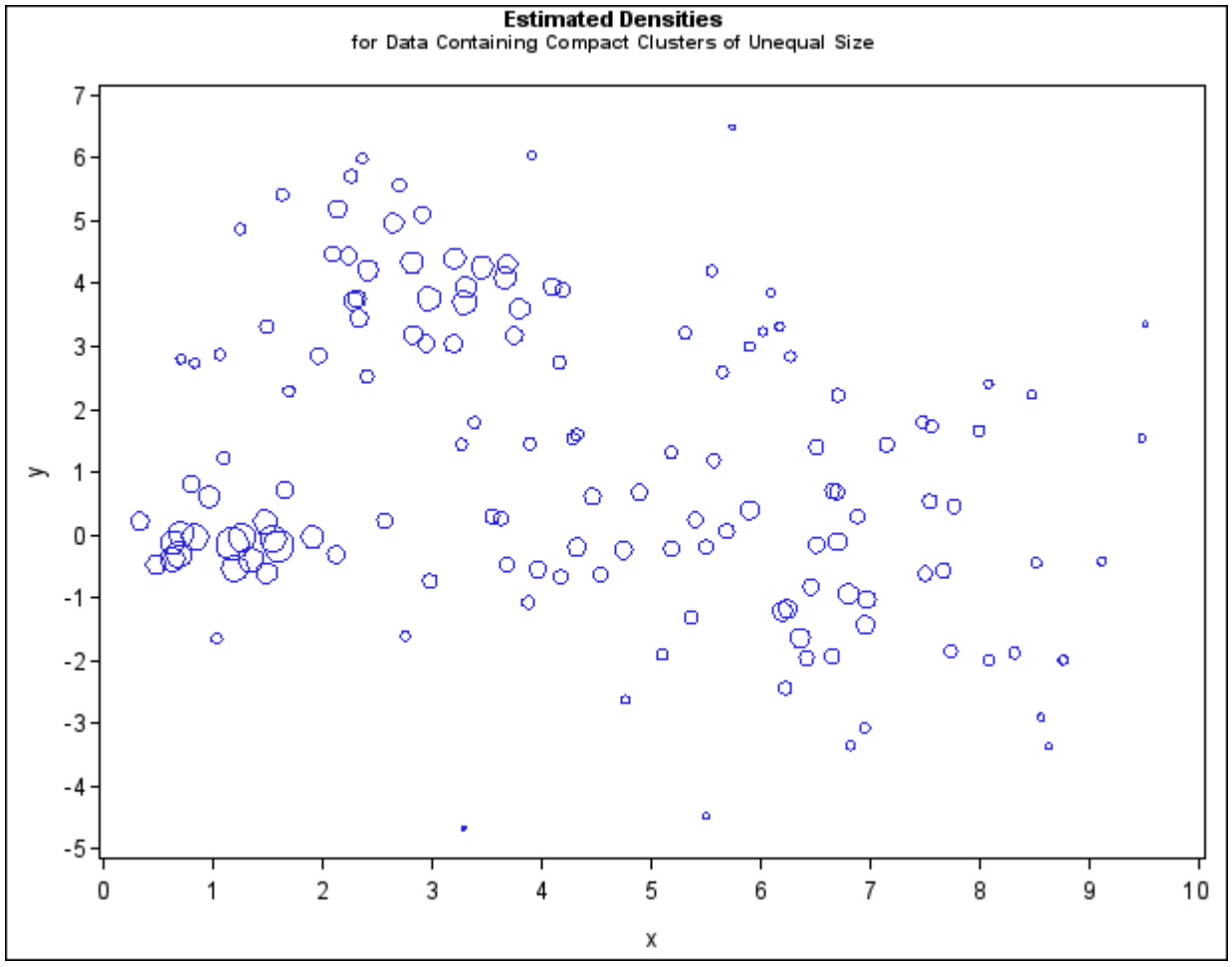


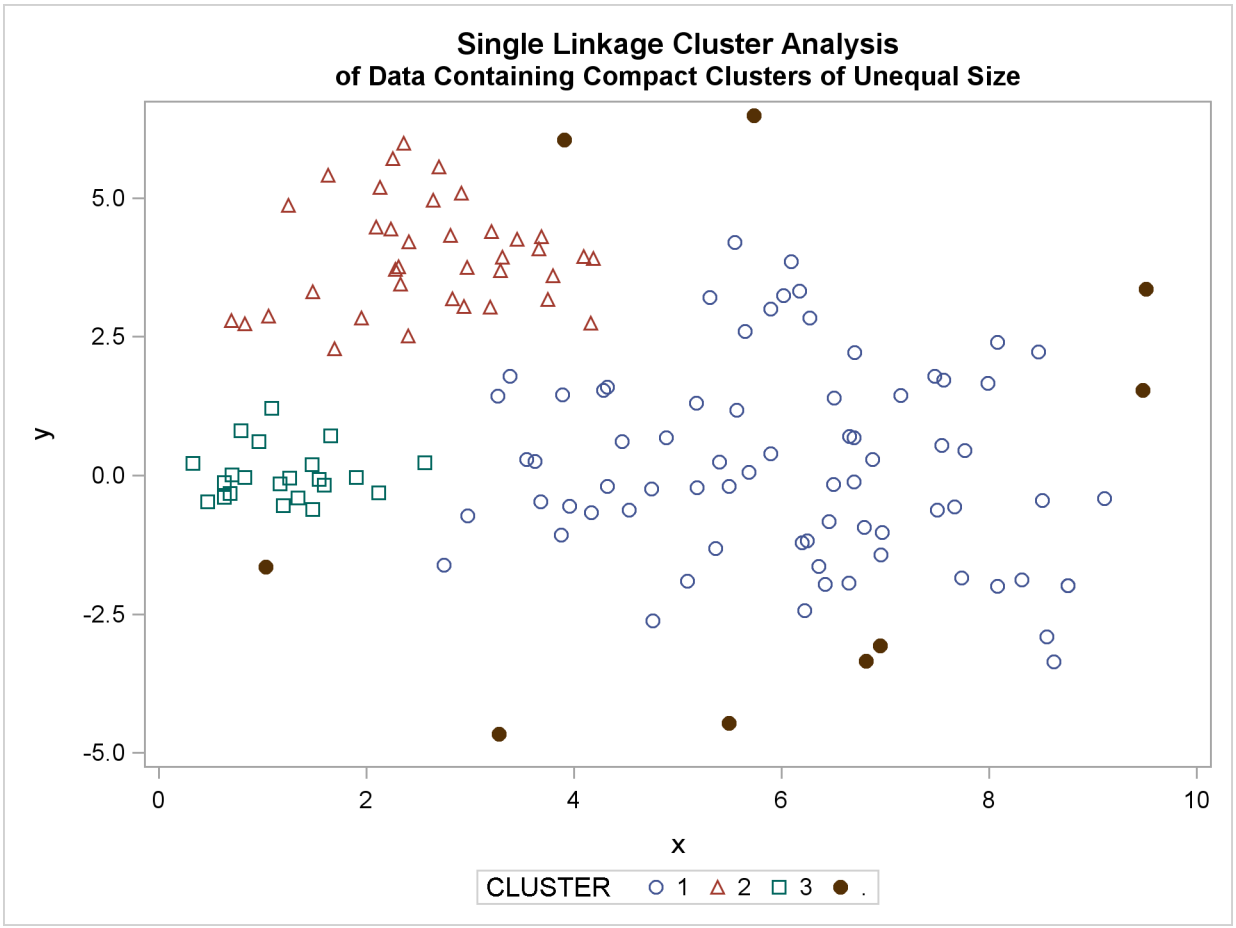
Figure 11.16 Data Containing Compact Clusters of Unequal Size: PROC CLUSTER with METHOD=TWOSTAGE



The following SAS statements produce Figure 11.17:

```
proc cluster data=unequal outtree=tree
             method=single noprint;
   var x y;
run;

proc tree data=tree noprint out=out n=3 dock=5;
   copy x y;
   title 'Single Linkage Cluster Analysis';
   title2 'of Data Containing Compact Clusters of Unequal Size';
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
run;
```

Figure 11.17 Data Containing Compact Clusters of Unequal Size: PROC CLUSTER with METHOD=SINGLE



In the PROC FASTCLUS analysis, the smallest cluster, in the bottom-left portion of the plot, has stolen members from the other two clusters, and the upper-left cluster has also acquired some observations that rightfully belong to the larger, lower-right cluster. With Ward's method, the upper-left

cluster is separated correctly, but the lower-left cluster has taken a large bite out of the lower-right cluster. For both of these methods, the clustering errors are in accord with the biases of the methods to produce clusters of equal size. In the average linkage analysis, both the upper-left and lower-left clusters have encroached on the lower-right cluster, thereby making the variances more nearly equal than in the true clusters. The centroid method, which lacks the size and dispersion biases of the previous methods, obtains an essentially correct partition.

Two-stage density linkage does almost as well, even though the compact shapes of these clusters favor the traditional methods. Single linkage also produces excellent results.

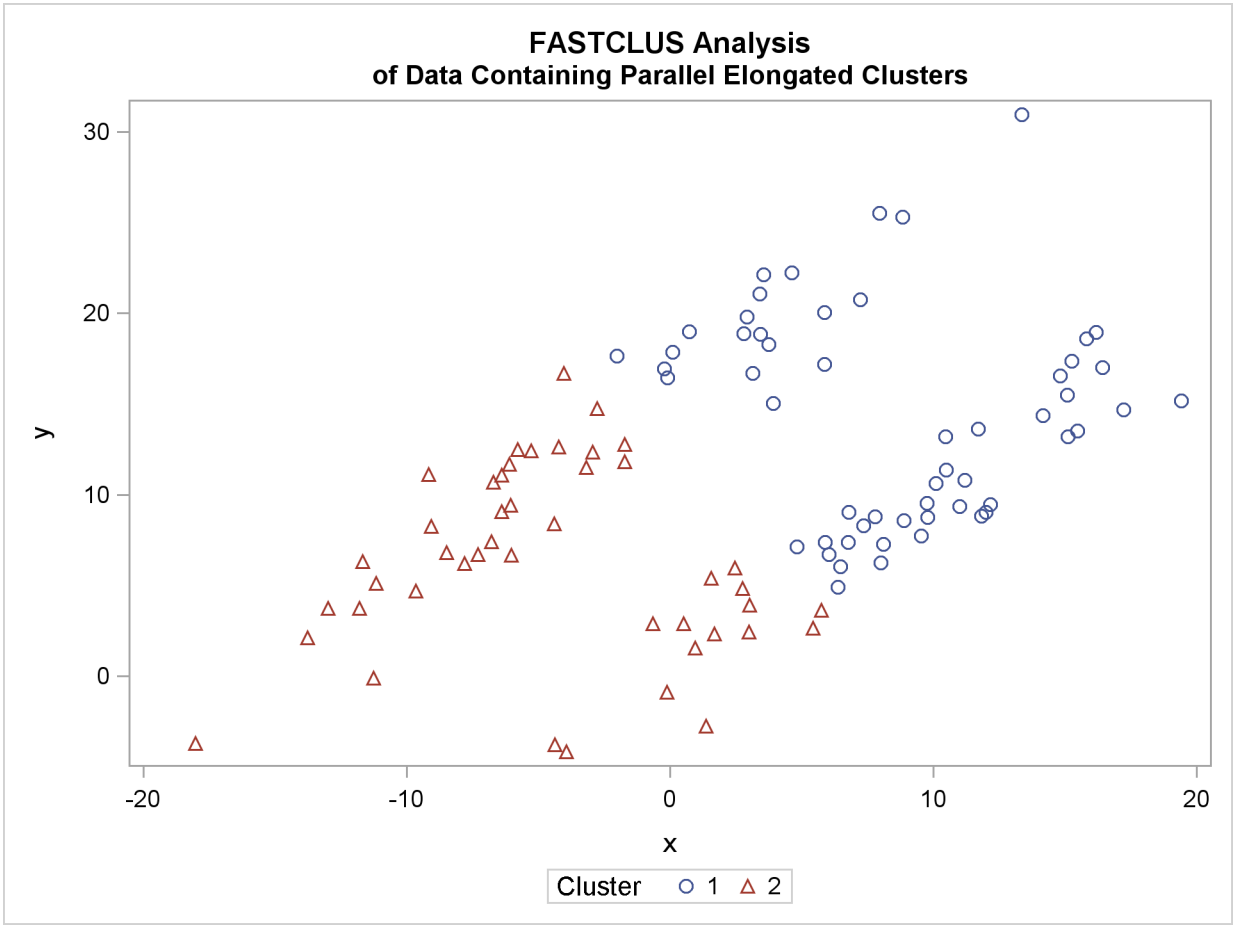
Elongated Multinormal Clusters

In this example, the data are sampled from two highly elongated multinormal distributions with equal covariance matrices. The following SAS statements produce Figure 11.18:

```
data elongate;
   keep x y;
   ma=8; mb=0; link generate;
   ma=6; mb=8; link generate;
   stop;
generate:
   do i=1 to 50;
      a=rannor(7)*6+ma;
      b=rannor(7)+mb;
      x=a-b;
      y=a+b;
      output;
   end;
   return;
proc fastclus data=elongate out=out maxc=2 noprint;
run;

%modstyle(name=ClusterStyle2,parent=Statistical,type=CLM,
markers=Circle Triangle circlefilled);
ods listing style=ClusterStyle;

proc sgplot;
   scatter y=y x=x / group=cluster;
   title 'FASTCLUS Analysis';
   title2 'of Data Containing Parallel Elongated Clusters';
run;
```

Notice that PROC FASTCLUS found two clusters, as requested by the MAXC= option. However, it attempted to form spherical clusters, which are obviously inappropriate for these data.

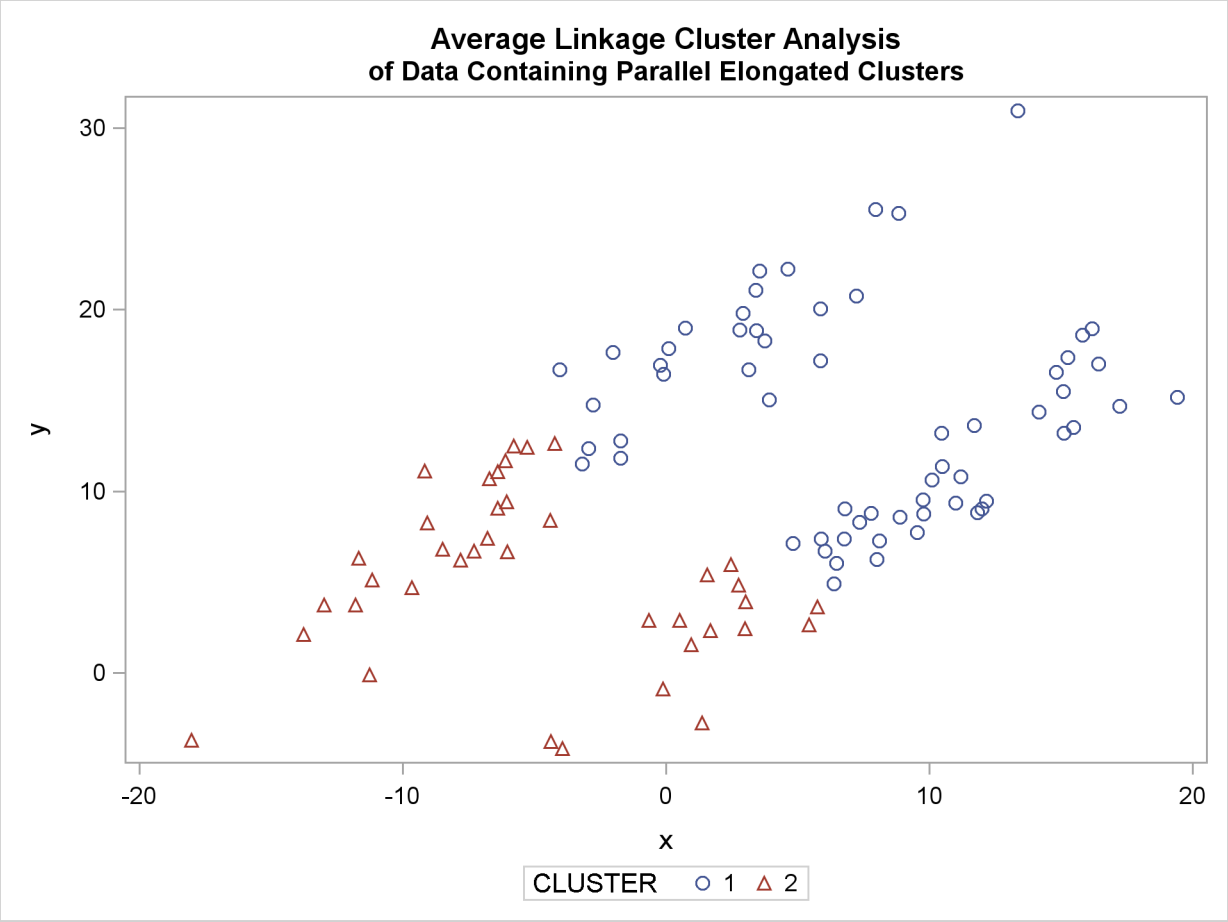
Figure 11.18 Data Containing Parallel Elongated Clusters: PROC FASTCLUS



The following SAS statements produce Figure 11.19:

```
proc cluster data=elongate outtree=tree
             method=average noprint;
run;

proc tree noprint out=out n=2 dock=5;
   copy x y;
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
   title 'Average Linkage Cluster Analysis';
   title2 'of Data Containing Parallel Elongated Clusters';
run;
```

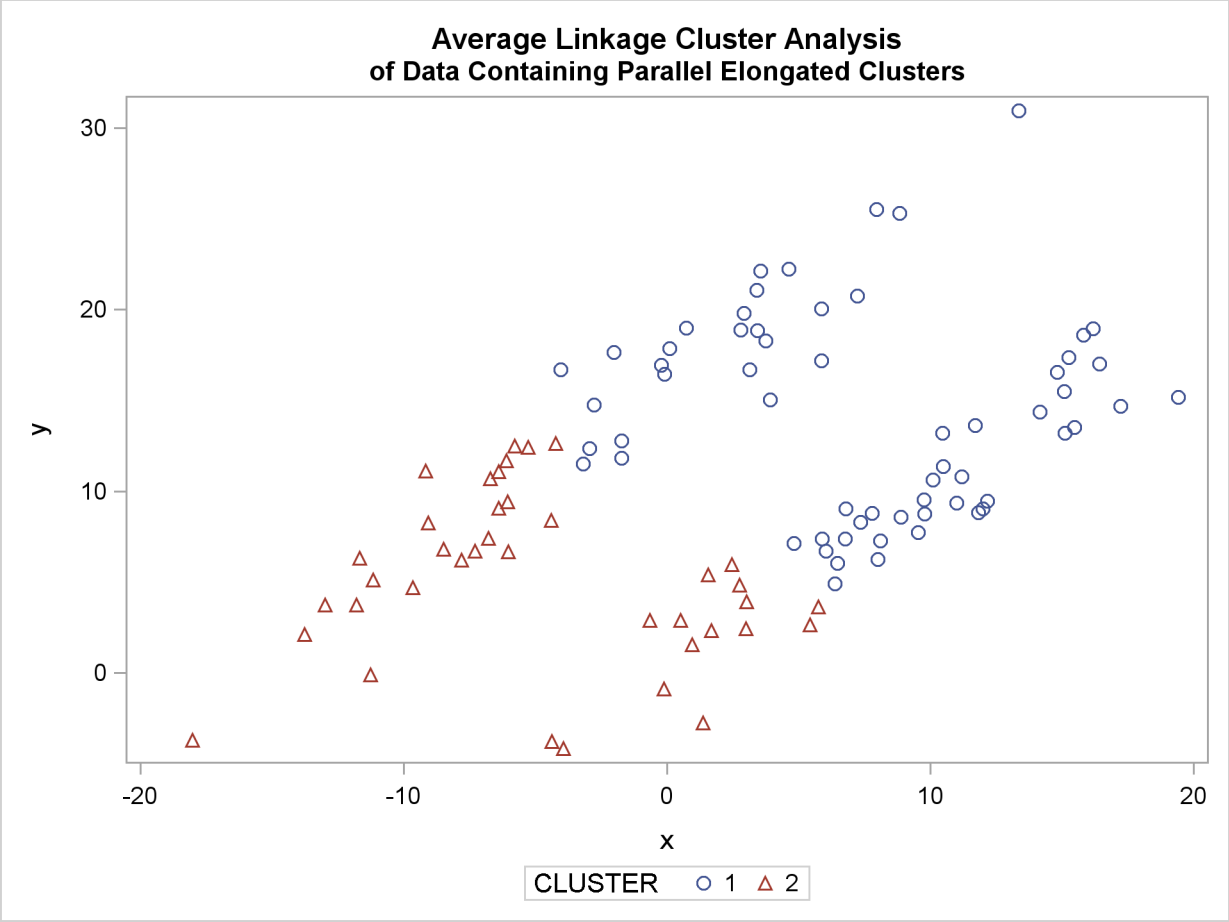
Figure 11.19 Data Containing Parallel Elongated Clusters: PROC CLUSTER with METHOD=AVERAGE



The following SAS statements produce Figure 11.20:

```
proc cluster data=elongate outtree=tree
             method=twostage k=10 noprint;
run;

proc tree noprint out=out n=2;
   copy x y;
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
   title 'Two-Stage Density Linkage Cluster Analysis';
   title2 'of Data Containing Parallel Elongated Clusters';
run;
```

Figure 11.20 Data Containing Parallel Elongated Clusters: PROC CLUSTER with METHOD=TWOSTAGE



PROC FASTCLUS and average linkage fail miserably. Ward's method and the centroid method (not shown) produce almost the same results. Two-stage density linkage, however, recovers the correct clusters. Single linkage (not shown) finds the same clusters as two-stage density linkage except for some outliers.

In this example, the population clusters have equal covariance matrices. If the within-cluster covariances are known, the data can be transformed to make the clusters spherical so that any of the clustering methods can find the correct clusters. But when you are doing a cluster analysis, you do not know what the true clusters are, so you cannot calculate the within-cluster covariance matrix. Nevertheless, it is sometimes possible to estimate the within-cluster covariance matrix without knowing the cluster membership or even the number of clusters, using an approach invented by Art, Gnanadesikan, and Kettenring (1982). A method for obtaining such an estimate is available in the ACECLUS procedure.

In the following analysis, PROC ACECLUS transforms the variables X and Y into the canonical variables CAN1 and CAN2. The latter are plotted and then used in a cluster analysis by Ward's method. The clusters are then plotted with the original variables X and Y.

The following SAS statements produce Figure 11.21 and Figure 11.22:

```
proc aceclus data=elongate out=ace p=.1;
   var x y;
   title 'ACECLUS Analysis';
   title2 'of Data Containing Parallel Elongated Clusters';
run;

proc sgplot;
   scatter y=can2 x=can1;
   title 'Data Containing Parallel Elongated Clusters';
   title2 'After Transformation by PROC ACECLUS';
run;
```

Figure 11.21 Data Containing Parallel Elongated Clusters: PROC ACECLUS

```
                              ACECLUS Analysis
                 of Data Containing Parallel Elongated Clusters

                            The ACECLUS Procedure

             Approximate Covariance Estimation for Cluster Analysis

               Observations         100    Proportion      0.1000
               Variables              2    Converge       0.00100

                         Means and Standard Deviations
                                                  Standard
                       Variable        Mean    Deviation

                       x             2.6406       8.3494
                       y            10.6488       6.8420

                         COV: Total Sample Covariances

                                       x                 y

                     x       69.71314819       24.24268934
                     y       24.24268934       46.81324861

      Initial Within-Cluster Covariance Estimate = Full Covariance Matrix


                           Threshold =    0.328478
```

Figure 11.21 *continued*

Iteration History

Iteration	RMS Distance	Distance Cutoff	Pairs Within Cutoff	Convergence Measure
1	2.000	0.657	672.0	0.673685
2	9.382	3.082	716.0	0.006963
3	9.339	3.068	760.0	0.008362
4	9.437	3.100	824.0	0.009656
5	9.359	3.074	889.0	0.010269
6	9.267	3.044	955.0	0.011276
7	9.208	3.025	999.0	0.009230
8	9.230	3.032	1052.0	0.011394
9	9.226	3.030	1091.0	0.007924
10	9.173	3.013	1121.0	0.007993

WARNING: Iteration limit exceeded.

ACE: Approximate Covariance Estimate Within Clusters

	x	y
x	9.299329632	8.215362614
y	8.215362614	8.937753936

Eigenvalues of Inv(ACE)*(COV-ACE)

	Eigenvalue	Difference	Proportion	Cumulative
1	36.7091	33.1672	0.9120	0.9120
2	3.5420		0.0880	1.0000

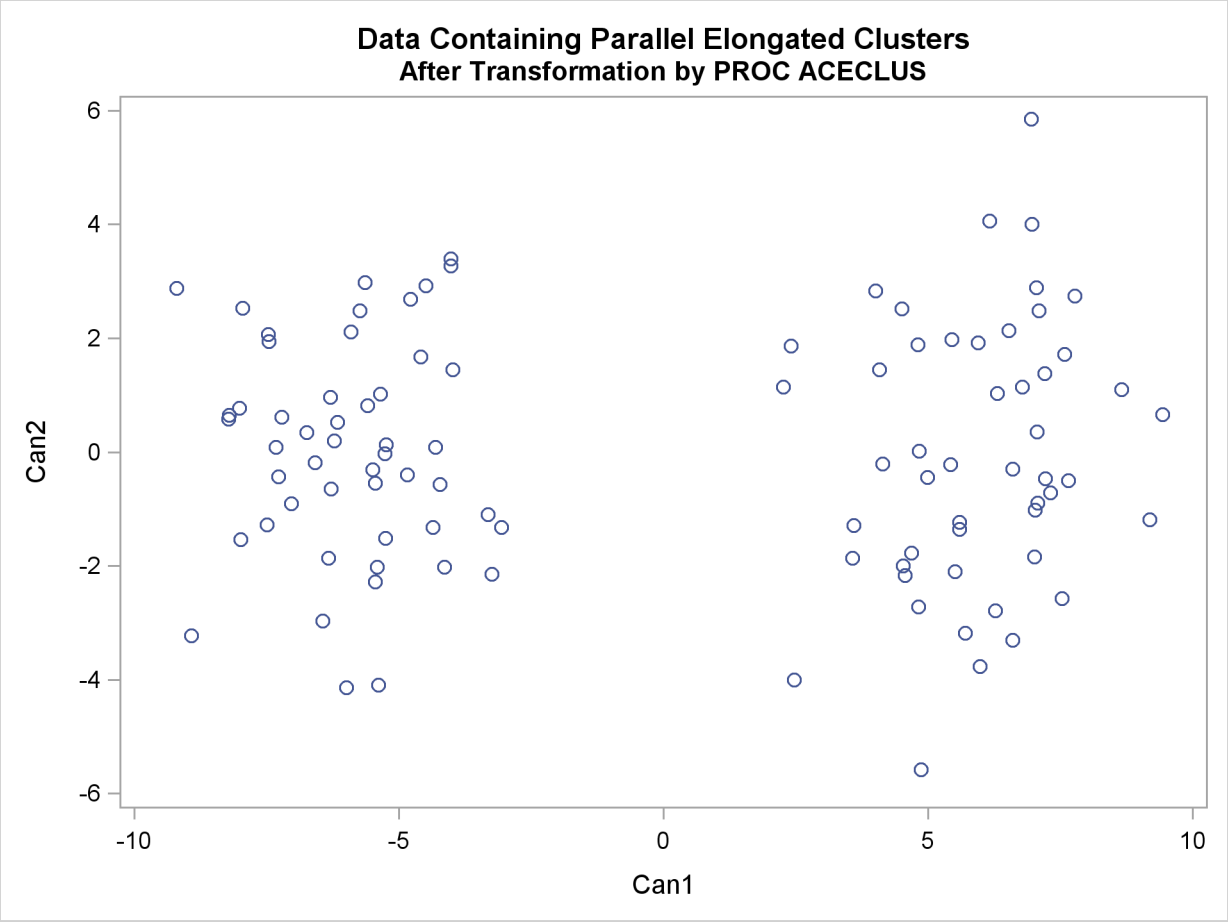
Eigenvectors (Raw Canonical Coefficients)

	Can1	Can2
x	-.748392	0.109547
y	0.736349	0.230272

Standardized Canonical Coefficients

	Can1	Can2
x	-6.24866	0.91466
y	5.03812	1.57553

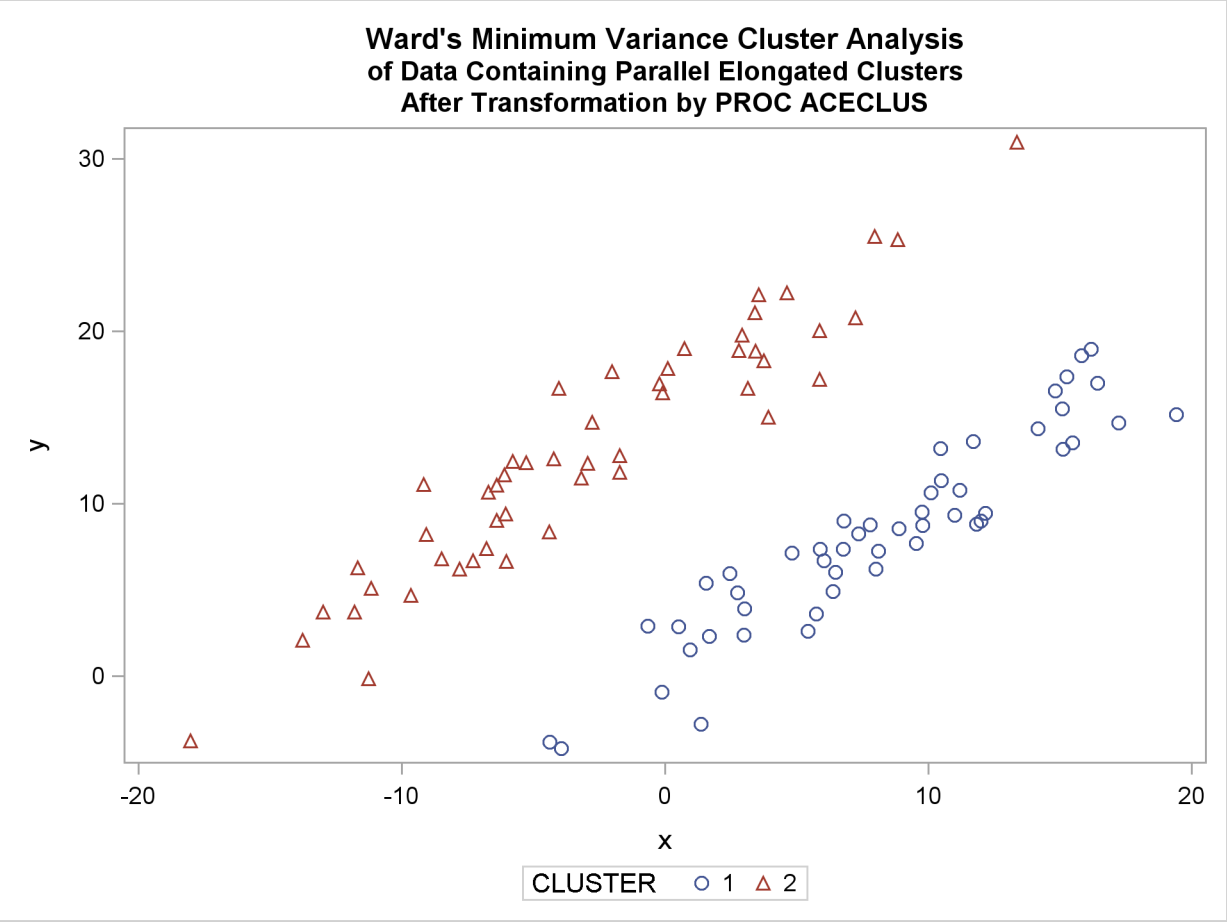
Figure 11.22 Data Containing Parallel Elongated Clusters after Transformation by PROC ACECLUS



The following SAS statements produce Figure 11.23:

```
proc cluster data=ace outtree=tree method=ward noprint;
   var can1 can2;
   copy x y;
run;

proc tree noprint out=out n=2;
   copy x y;
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
   title 'Ward''s Minimum Variance Cluster Analysis';
   title2 'of Data Containing Parallel Elongated Clusters';
   title3 'After Transformation by PROC ACECLUS';
run;
```

Figure 11.23 Transformed Data Containing Parallel Elongated Clusters: PROC CLUSTER with METHOD=WARD



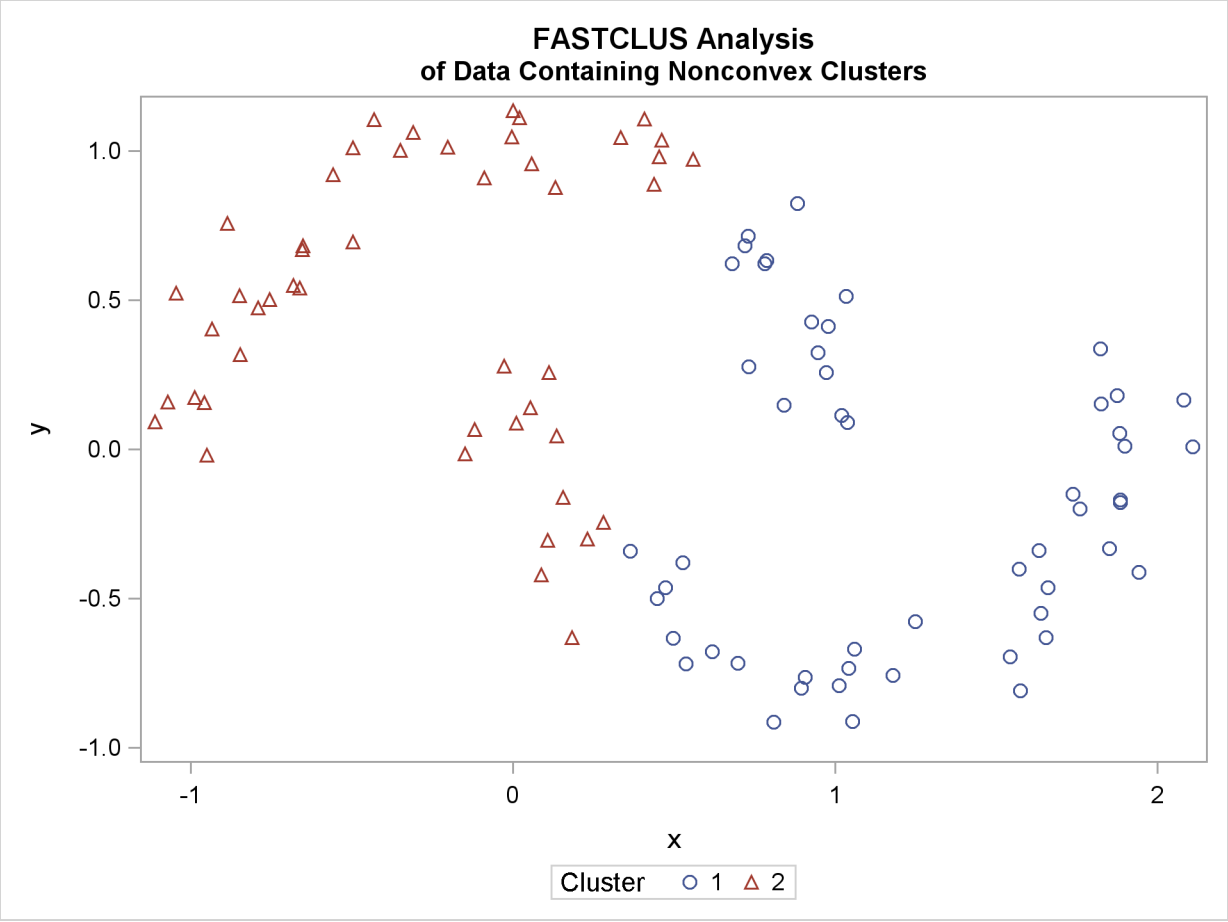
Nonconvex Clusters

If the population clusters have very different covariance matrices, using PROC ACECLUS is of no avail. Although methods exist for estimating multinormal clusters with unequal covariance matrices (Wolfe 1970; Symons 1981; Everitt and Hand 1981; Titterington, Smith, and Makov 1985; McLachlan and Basford 1988), these methods tend to have serious problems with initialization and might converge to degenerate solutions. For unequal covariance matrices or radically nonnormal distributions, the best approach to cluster analysis is through nonparametric density estimation, as in density linkage. The next example illustrates population clusters with nonconvex density contours. The following SAS statements produce Figure 11.24:

```
data noncon;
   keep x y;
   do i=1 to 100;
      a=i*.0628319;
      x=cos(a)+(i>50)+rannor(7)*.1;
      y=sin(a)+(i>50)*.3+rannor(7)*.1;
      output;
   end;
```

```
proc fastclus data=noncon out=out maxc=2 noprint;
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
   title 'FASTCLUS Analysis';
   title2 'of Data Containing Nonconvex Clusters';
run;
```

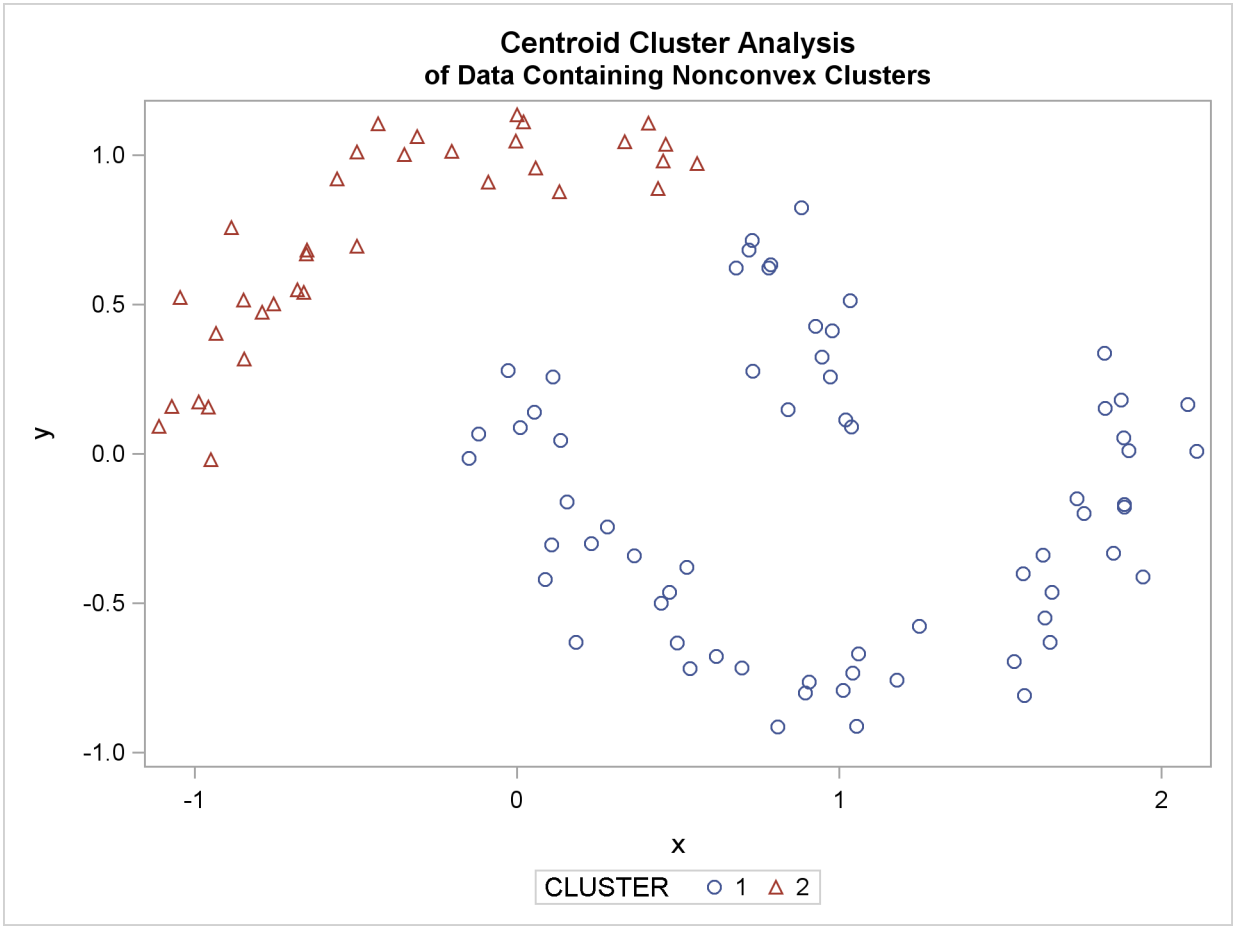
Figure 11.24 Data Containing Nonconvex Clusters: PROC FASTCLUS



The following SAS statements produce Figure 11.25:

```
proc cluster data=noncon outtree=tree
             method=centroid noprint;
run;

proc tree noprint out=out n=2 dock=5;
   copy x y;
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
   title 'Centroid Cluster Analysis';
   title2 'of Data Containing Nonconvex Clusters';
run;
```

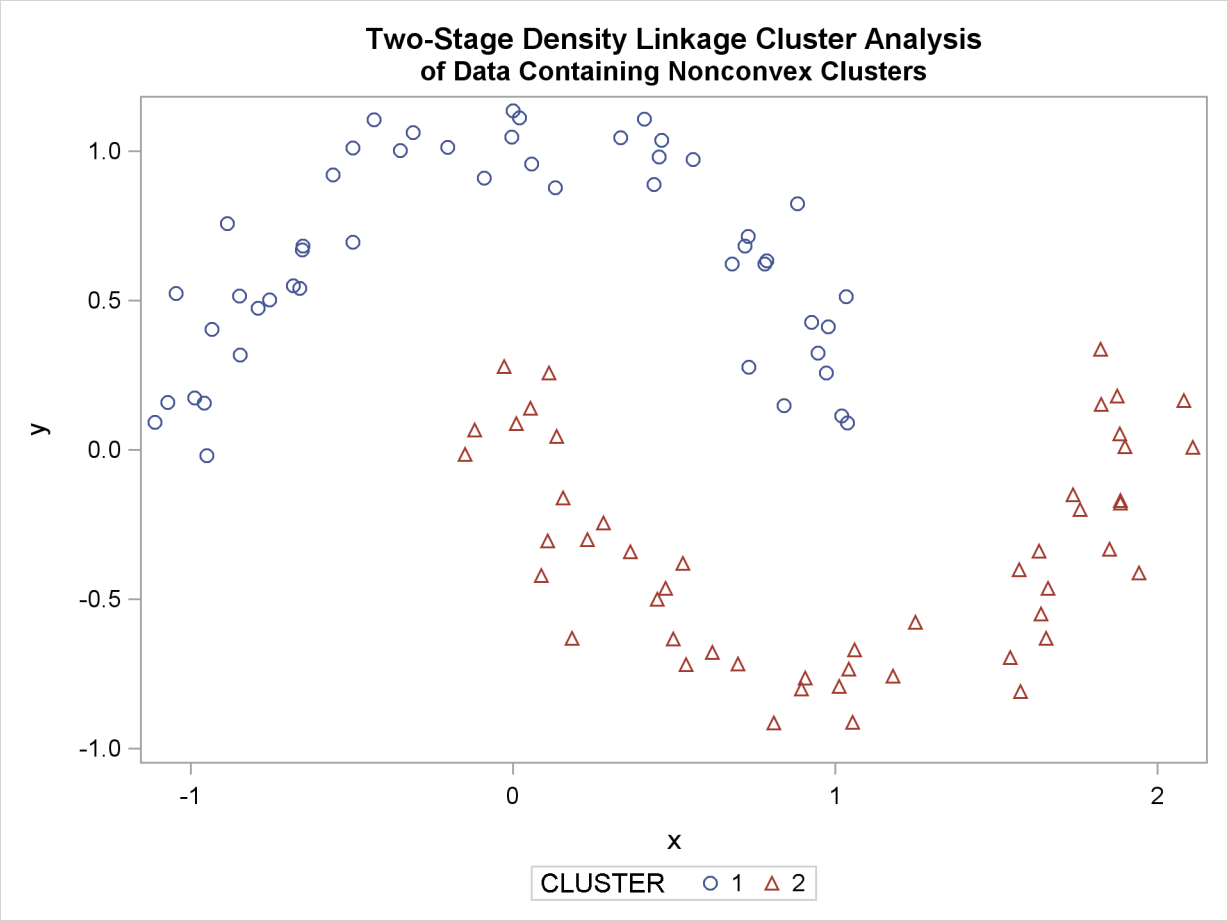
Figure 11.25 Data Containing Nonconvex Clusters: PROC CLUSTER with METHOD=CENTROID



The following SAS statements produce Figure 11.26:

```
proc cluster data=noncon outtree=tree
             method=twostage k=10 noprint;
run;

proc tree noprint out=out n=2;
   copy x y;
run;

proc sgplot;
   scatter y=y x=x / group=cluster;
   title 'Two-Stage Density Linkage Cluster Analysis';
   title2 'of Data Containing Nonconvex Clusters';
run;
```

Figure 11.26 Data Containing Nonconvex Clusters: PROC CLUSTER with METHOD=TWOSTAGE



Ward's method and average linkage (not shown) do better than PROC FASTCLUS but not as well as the centroid method. Two-stage density linkage recovers the correct clusters, as does single linkage (not shown).

The preceding examples are intended merely to illustrate some of the properties of clustering methods in common use. If you intend to perform a cluster analysis, you should consult more systematic and rigorous studies of the properties of clustering methods, such as Milligan (1980).

The Number of Clusters

There are no completely satisfactory methods that can be used for determining the number of population clusters for any type of cluster analysis (Everitt 1979; Hartigan, J. A. 1985; Bock 1985).

If your purpose in clustering is dissection—that is, to summarize the data without trying to uncover real clusters—it might suffice to look at R square for each variable and pooled over all variables. Plots of R square against the number of clusters are useful.

It is always a good idea to look at your data graphically. If you have only two or three variables, use PROC SGPLOT to make scatter plots identifying the clusters. With more variables, use PROC CANDISC to compute canonical variables for plotting.

Ordinary significance tests, such as analysis of variance F tests, are not valid for testing differences between clusters. Since clustering methods attempt to maximize the separation between clusters, the assumptions of the usual significance tests, parametric or nonparametric, are drastically violated. For example, if you take a sample of 100 observations from a single univariate normal distribution, have PROC FASTCLUS divide it into two clusters, and run a t test between the clusters, you usually obtain a p-value of less than 0.0001. For the same reason, methods that purport to test for clusters against the null hypothesis that objects are assigned randomly to clusters (such as McClain and Rao 1975; Klastorin 1983) are useless.

Most valid tests for clusters either have intractable sampling distributions or involve null hypotheses for which rejection is uninformative. For clustering methods based on distance matrices, a popular null hypothesis is that all permutations of the values in the distance matrix are equally likely (Ling 1973; Hubert 1974). Using this null hypothesis, you can do a permutation test or a rank test. The trouble with the permutation hypothesis is that, with any real data, the null hypothesis is implausible even if the data do not contain clusters. Rejecting the null hypothesis does not provide any useful information (Hubert and Baker 1977).

Another common null hypothesis is that the data are a random sample from a multivariate normal distribution (Wolfe 1970, 1978; Duda and Hart 1973; Lee 1979). The multivariate normal null hypothesis arises naturally in normal mixture models (Titterington, Smith, and Makov 1985; McLachlan and Basford 1988). Unfortunately, the likelihood ratio test statistic does not have the usual asymptotic χ^2 distribution because the regularity conditions do not hold. Approximations to the asymptotic distribution of the likelihood ratio have been suggested (Wolfe 1978), but the adequacy of these approximations is debatable (Everitt 1981; Thode, Mendell, and Finch 1988). For small samples, bootstrapping seems preferable (McLachlan and Basford 1988). Bayesian inference provides a promising alternative to likelihood ratio tests for the number of mixture components for both normal mixtures and other types of distributions (Binder 1978, 1981; Banfield and Raftery 1993; Bensmail et al. 1997).

The multivariate normal null hypothesis is better than the permutation null hypothesis, but it is not satisfactory because there is typically a high probability of rejection if the data are sampled from a distribution with lower kurtosis than a normal distribution, such as a uniform distribution. The tables in Englemann and Hartigan (1969), for example, generally lead to rejection of the null hypothesis when the data are sampled from a uniform distribution. Hawkins, Muller, and ten Krooden (1982, pp. 337–340) discuss a highly conservative Bonferroni method for the use of hypothesis testing. The conservativeness of this approach might compensate to some extent for the liberalness exhibited by tests based on normal distributions when the population is uniform.

Perhaps a better null hypothesis is that the data are sampled from a uniform distribution (Hartigan 1978; Arnold 1979; Sarle 1983). The uniform null hypothesis leads to conservative error rates when the data are sampled from a strongly unimodal distribution such as the normal. However, in two or more dimensions and depending on the test statistic, the results can be very sensitive to the shape of the region of support of the uniform distribution. Sarle (1983) suggests using a hyperbox with sides proportional in length to the singular values of the centered coordinate matrix.

Given that the uniform distribution provides an appropriate null hypothesis, there are still serious difficulties in obtaining sampling distributions. Some asymptotic results are available (Hartigan 1978, 1985; Pollard 1981; Bock 1985) for the within-cluster sum of squares, the criterion that PROC FASTCLUS and Ward's minimum variance method attempt to optimize. No distributional theory for finite sample sizes has yet appeared. Currently, the only practical way to obtain sampling distributions for realistic sample sizes is by computer simulation.

Arnold (1979) used simulation to derive tables of the distribution of a criterion based on the determinant of the within-cluster sum of squares matrix $|\mathbf{W}|$. Both normal and uniform null distributions were used. Having obtained clusters with either PROC FASTCLUS or PROC CLUSTER, you can compute Arnold's criterion with the ANOVA or CANDISC procedure. Arnold's tables provide a conservative test because PROC FASTCLUS and PROC CLUSTER attempt to minimize the trace of $\mathbf{W}$ rather than the determinant. Marriott (1971, 1975) also provides useful information about $|\mathbf{W}|$ as a criterion for the number of clusters.

Sarle (1983) used extensive simulations to develop the cubic clustering criterion (CCC), which can be used for crude hypothesis testing and estimating the number of population clusters. The CCC is based on the assumption that a uniform distribution on a hyperrectangle will be divided into clusters shaped roughly like hypercubes. In large samples that can be divided into the appropriate number of hypercubes, this assumption gives very accurate results. In other cases the approximation is generally conservative. For details about the interpretation of the CCC, consult Sarle (1983).

Milligan and Cooper (1985) and Cooper and Milligan (1988) compared 30 methods of estimating the number of population clusters by using four hierarchical clustering methods. The three criteria that performed best in these simulation studies with a high degree of error in the data were a pseudo F statistic developed by Calinski and Harabasz (1974), a statistic referred to as $J_e(2)/J_e(1)$ by Duda and Hart (1973) that can be transformed into a pseudo t^2 statistic, and the cubic clustering criterion. The pseudo F statistic and the CCC are displayed by PROC FASTCLUS; these two statistics and the pseudo t^2 statistic, which can be applied only to hierarchical methods, are displayed by PROC CLUSTER. It might be advisable to look for consensus among the three statistics—that is, local peaks of the CCC and pseudo F statistic combined with a small value of the pseudo t^2 statistic and a larger pseudo t^2 for the next cluster fusion. It must be emphasized that these criteria are appropriate only for compact or slightly elongated clusters, preferably clusters that are roughly multivariate normal.

Recent research has tended to deemphasize mixture models in favor of nonparametric models in which clusters correspond to modes in the probability density function. Hartigan and Hartigan (1985) and P. M. Hartigan (1985) developed a test of unimodality versus bimodality in the univariate case.

Nonparametric tests for the number of clusters can also be based on nonparametric density estimates. This approach requires much weaker assumptions than mixture models, namely, that the observations are sampled independently and that the distribution can be estimated nonparametrically. Silverman (1986) describes a bootstrap test for the number of modes using a Gaussian kernel density estimate, but problems have been reported with this method under the uniform null distribution. Further developments in nonparametric methods are given by Mueller and Sawitzki (1991), Minnotte (1992), and Polonik (1993). All of these methods suffer from heavy computational requirements.

One useful descriptive approach to the number-of-clusters problem is provided by Wong and Schaack (1982), based on a kth-nearest-neighbor density estimate. The kth-nearest-neighbor clustering method developed by Wong and Lane (1983) is applied with varying values of k. Each value of k yields an estimate of the number of modal clusters. If the estimated number of modal clusters is constant for a wide range of k values, there is strong evidence of at least that many modes in the population. A plot of the estimated number of modes against k can be highly informative. Attempts to derive a formal hypothesis test from this diagnostic plot have met with difficulties, but a simulation approach similar to Silverman's (1986) does seem to work (Girman 1994). The simulation, of course, requires considerable computer time.

Sarle and Kuo (1993) document a less expensive approximate nonparametric test for the number of clusters that has been implemented in the MODECLUS procedure. This test sacrifices statistical efficiency for computational efficiency. The method for conducting significance tests is described in the chapter on the MODECLUS procedure. This method has the following useful features:

- No distributional assumptions are required.
- The choice of smoothing parameter is not critical since you can try any number of different values.
- The data can be coordinates or distances.
- Time and space requirements for the significance tests are no worse than those for obtaining the clusters.
- The power is high enough to be useful for practical purposes.

The method for computing the p-values is based on a series of plausible approximations. There are as yet no rigorous proofs that the method is infallible. Neither are there any asymptotic results. However, simulations for sample sizes ranging from 20 to 2000 indicate that the p-values are almost always conservative. The only case discovered so far in which the p-values are liberal is a uniform distribution in one dimension for which the simulated error rates exceed the nominal significance level only slightly for a limited range of sample sizes.

References

Anderberg, M. R. (1973), *Cluster Analysis for Applications*, New York: Academic Press.

Arnold, S. J. (1979), "A Test for Clusters," *Journal of Marketing Research,* 16, 545–551.

Art, D., Gnanadesikan, R., and Kettenring, R. (1982), "Data-based Metrics for Cluster Analysis," *Utilitas Mathematica*, 21A, 75–99.

Banfield, J. D. and Raftery, A.E. (1993), "Model-Based Gaussian and Non-Gaussian Clustering," *Biometrics*, 49, 803–821.

Bensmail, H., Celeux, G., Raftery, A. E., and Robert, C. P. (1997), "Inference in Model-Based Cluster Analysis," *Statistics and Computing*, 7, 1–10.

Bezdek, J. C. (1981), *Pattern Recognition with Fuzzy Objective Function Algorithms*, New York: Plenum Press.

Bezdek, J. C. and Pal, S. K. (Editor) (1992), *Fuzzy Models for Pattern Recognition*, IEEE Press, New York.

Binder, D. A. (1978), "Bayesian Cluster Analysis," *Biometrika,* 65, 31–38.

Binder, D. A. (1981), "Approximations to Bayesian Clustering Rules," *Biometrika*, 68, 275–285.

Blashfield, R. K. and Aldenderfer, M. S. (1978), "The Literature on Cluster Analysis," *Multivariate Behavioral Research*, 13, 271–295.

Bock, H. H. (1985), "On Some Significance Tests in Cluster Analysis," *Journal of Classification*, 2, 77–108.

Calinski, T. and Harabasz, J. (1974), "A Dendrite Method for Cluster Analysis," *Communications in Statistics*, 3, 1–27.

Cooper, M. C. and Milligan, G. W. (1988), "The Effect of Error on Determining the Number of Clusters," *Proceedings of the International Workshop on Data Analysis, Decision Support and Expert Knowledge Representation in Marketing and Related Areas of Research,* 319–328.

Duda, R. O. and Hart, P. E. (1973), *Pattern Classification and Scene Analysis*, New York: John Wiley & Sons.

Duran, B. S. and Odell, P. L. (1974), *Cluster Analysis*, New York: Springer-Verlag.

Englemann, L. and Hartigan, J. A. (1969), "Percentage Points of a Test for Clusters," *Journal of the American Statistical Association,* 64, 1647–1648.

Everitt, B. S. (1979), "Unresolved Problems in Cluster Analysis," *Biometrics*, 35, 169–181.

Everitt, B. S. (1980), *Cluster Analysis*, Second Edition, London: Heineman Educational Books.

Everitt, B. S. (1981), "A Monte Carlo Investigation of the Likelihood Ratio Test for the Number of Components in a Mixture of Normal Distributions," *Multivariate Behavioral Research*, 16, 171–80.

Everitt, B. S. and Hand, D. J. (1981), *Finite Mixture Distributions*, New York: Chapman & Hall.

Gersho, A. and Gray, R. M. (1992), *Vector Quantization and Signal Compression*, Kluwer Academic Publishers.

Girman, C. J. (1994), "Cluster Analysis and Classification Tree Methodology as an Aid to Improve Understanding of Benign Prostatic Hyperplasia," Ph.D. diss., Department of Biostatistics, University of North Carolina.

Good, I. J. (1977), "The Botryology of Botryology," in *Classification and Clustering,* ed. J. Van Ryzin, New York: Academic Press.

Harman, H. H. (1976), *Modern Factor Analysis*, Third Edition, Chicago: University of Chicago Press.

Hartigan, J. A. (1975), *Clustering Algorithms*, New York: John Wiley & Sons.

Hartigan, J. A. (1977), "Distribution Problems in Clustering," in *Classification and Clustering*, ed. J. Van Ryzin, New York: Academic Press.

Hartigan, J. A. (1978), "Asymptotic Distributions for Clustering Criteria," *Annals of Statistics*, 6, 117–131.

Hartigan, J. A. (1981), "Consistency of Single Linkage for High-Density Clusters," *Journal of the American Statistical Association*, 76, 388–394.

Hartigan, J. A. (1985), "Statistical Theory in Clustering," *Journal of Classification*, 2, 63–76.

Hartigan, J. A. and Hartigan, P. M. (1985), "The Dip Test of Unimodality," *Annals of Statistics*, 13, 70–84.

Hartigan, P. M. (1985), "Computation of the Dip Statistic to Test for Unimodality," *Applied Statistics*, 34, 320–325.

Hawkins, D. M., Muller, M. W., and ten Krooden, J. A. (1982), "Cluster Analysis," in *Topics in Applied Multivariate Analysis*, ed. D. M. Hawkins, Cambridge: Cambridge University Press.

Hubert, L. (1974), "Approximate Evaluation Techniques for the Single-Link and Complete-Link Hierarchical Clustering Procedures," *Journal of the American Statistical Association*, 69, 698–704.

Hubert, L. J. and Baker, F. B. (1977), "An Empirical Comparison of Baseline Models for Goodness-of-Fit in r-Diameter Hierarchical Clustering," in *Classification and Clustering*, ed. J. Van Ryzin, New York: Academic Press.

Kaufman, L. and Rousseeuw, P. J. (1990), *Finding Groups in Data*, New York: John Wiley & Sons.

Klastorin, T. D. (1983), "Assessing Cluster Analysis Results," *Journal of Marketing Research*, 20, 92–98.

Lee, K. L. (1979), "Multivariate Tests for Clusters," *Journal of the American Statistical Association*, 74, 708–714.

Ling, R. F. (1973), "A Probability Theory of Cluster Analysis," *Journal of the American Statistical Association*, 68, 159–169.

MacQueen, J. B. (1967), "Some Methods for Classification and Analysis of Multivariate Observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1, 281–297.

Marriott, F. H. C. (1971), "Practical Problems in a Method of Cluster Analysis," *Biometrics*, 27, 501–514.

Marriott, F. H. C. (1975), "Separating Mixtures of Normal Distributions," *Biometrics,* 31, 767–769.

Massart, D. L. and Kaufman, L. (1983), *The Interpretation of Analytical Chemical Data by the Use of Cluster Analysis*, New York: John Wiley & Sons.

McClain, J. O. and Rao, V. R. (1975), "CLUSTISZ: A Program to Test for the Quality of Clustering of a Set of Objects," *Journal of Marketing Research*, 12, 456–460.

McLachlan, G. J. and Basford, K. E. (1988), *Mixture Models*, New York: Marcel Dekker.

Mezzich, J. E. and Solomon, H. (1980), *Taxonomy and Behavioral Science*, New York: Academic Press.

Milligan, G. W. (1980), "An Examination of the Effect of Six Types of Error Perturbation on Fifteen Clustering Algorithms," *Psychometrika*, 45, 325–342.

Milligan, G. W. (1981), "A Review of Monte Carlo Tests of Cluster Analysis," *Multivariate Behavioral Research*, 16, 379–407.

Milligan, G. W. and Cooper, M. C. (1985), "An Examination of Procedures for Determining the Number of Clusters in a Data Set," *Psychometrika*, 50, 159–179.

Minnotte, M. C. (1992), "A Test of Mode Existence with Applications to Multimodality," Ph.D. diss., Rice University, Department of Statistics.

Mueller, D. W. and Sawitzki, G. (1991), "Excess Mass Estimates and Tests for Multimodality," JASA 86, 738–746.

Pollard, D. (1981), "Strong Consistency of k-Means Clustering," *Annals of Statistics*, 9, 135–140.

Polonik, W. (1993), "Measuring Mass Concentrations and Estimating Density Contour Clusters—An Excess Mass Approach," Technical Report, Beitraege zur Statistik Nr. 7, University of Heidelberg.

Sarle, W. S. (1982), "Cluster Analysis by Least Squares," *Proceedings of the Seventh Annual SAS Users Group International Conference*, 651–653.

Sarle, W. S. (1983), *Cubic Clustering Criterion*, SAS Technical Report A-108, Cary, NC: SAS Institute Inc.

Sarle, W. S. and Kuo, An-Hsiang (1993), *The MODECLUS Procedure*, SAS Technical Report P-256, Cary, NC: SAS Institute Inc.

Scott, A. J. and Symons, M. J. (1971), "Clustering Methods Based on Likelihood Ratio Criteria," *Biometrics*, 27, 387–397.

Silverman, B. W. (1986), *Density Estimation*, New York: Chapman & Hall.

Sneath, P. H. A. and Sokal, R. R. (1973), *Numerical Taxonomy*, San Francisco: W. H. Freeman.

Spath, H. (1980), *Cluster Analysis Algorithms*, Chichester, England: Ellis Horwood.

Symons, M. J. (1981), "Clustering Criteria and Multivariate Normal Mixtures," *Biometrics*, 37, 35–43.

Thode, H. C., Jr., Mendell, N. R., and Finch, S. J. (1988), "Simulated Percentage Points for the Null Distribution of the Likelihood Ratio Test for a Mixture of Two Normals," *Biometrics*, 44,

1195–1201.

Titterington, D. M., Smith, A. F. M., and Makov, U. E. (1985), *Statistical Analysis of Finite Mixture Distributions*, New York: John Wiley & Sons.

Ward, J. H. (1963), "Hierarchical Grouping to Optimize an Objective Function," *Journal of the American Statistical Association*, 58, 236–244.

Wolfe, J. H. (1970), "Pattern Clustering by Multivariate Mixture Analysis," *Multivariate Behavioral Research*, 5, 329–350.

Wolfe, J. H. (1978), "Comparative Cluster Analysis of Patterns of Vocational Interest," *Multivariate Behavioral Research*, 13, 33–44.

Wong, M. A. (1982), "A Hybrid Clustering Method for Identifying High-Density Clusters," *Journal of the American Statistical Association*, 77, 841–847.

Wong, M. A. and Lane, T. (1983), "A kth Nearest Neighbor Clustering Procedure," *Journal of the Royal Statistical Society*, Series B, 45, 362–368.

Wong, M. A. and Schaack, C. (1982), "Using the kth Nearest Neighbor Clustering Procedure to Determine the Number of Subpopulations," *American Statistical Association 1982 Proceedings of the Statistical Computing Section*, 40–48.

Chapter 12

Introduction to Scoring, Standardization, and Ranking Procedures

Contents

Overview: Scoring, Standardization, and Ranking Procedures 251

Overview: Scoring, Standardization, and Ranking Procedures

Several SAS/STAT procedures are utilities that produce an output data set with new variables that are transformations of data in the input data set. SAS/STAT software includes four of these procedures. The RANK procedure produces rank scores across observations, the SCORE procedure constructs functions across the variables, and the STANDARD and STDIZE procedures transform each variable individually.

RANK	ranks the observations of each numeric variable and outputs ranks or rank scores. For a complete discussion of the RANK procedure, see the *Base SAS Procedures Guide: Statistical Procedures*.
SCORE	constructs new variables that are linear combinations of old variables according to a scoring data set. This procedure is used with the FACTOR procedure and other procedures that output scoring coefficients.
STANDARD	standardizes variables to a given mean and standard deviation. For a complete discussion of PROC STANDARD, see the *Base SAS Procedures Guide: Statistical Procedures*.
STDIZE	standardizes variables by subtracting a location measure and dividing by a scale measure. A variety of location and scale measures are provided. Such measures include the mean, median, Huber's estimate, Tukey's biweight estimate, and Andrew's wave estimate.

Chapter 13

Introduction to Survival Analysis Procedures

Contents

Overview **253**
Survival Analysis Procedures **254**
The LIFEREG Procedure 254
The LIFETEST Procedure 255
The PHREG Procedure 255
Survival Analysis with SAS/STAT Procedures **255**
Bayesian Survival Analysis with SAS/STAT Procedures **256**
References **257**

Overview

Data that measure lifetime or the length of time until the occurrence of an event are called *lifetime, failure time,* or *survival* data. For example, variables of interest might be the lifetime of diesel engines, the length of time a person stayed on a job, or the survival time for heart transplant patients. Such data have special considerations that must be incorporated into any analysis.

Survival data consist of a response (event time, failure time, or survival time) variable that measures the duration of time until a specified event occurs and possibly a set of independent variables thought to be associated with the failure time variable. These independent variables (concomitant variables, covariates, or prognostic factors) can be either discrete, such as sex or race, or continuous, such as age or temperature. The system that gives rise to the event of interest can be biological (as for most medical data) or physical (as for engineering data). The purpose of survival analysis is to model the underlying distribution of the failure time variable and to assess the dependence of the failure time variable on the independent variables.

An intrinsic characteristic of survival data is the possibility for censoring of observations (that is, the actual time until the event is not observed). Such censoring can arise from withdrawal from the experiment or termination of the experiment. Because the response is usually a duration, some of the possible events may not yet have occurred when the period for data collection has terminated. For example, clinical trials are conducted over a finite period of time with staggered entry of patients. That is, patients enter a clinical trial over time, and thus the length of follow-up varies by individuals; consequently, the time to the event may not be ascertained on all patients in the study. Additionally, some of the responses may be lost to follow-up (for example, a participant may move or refuse to

continue to participate) before termination of data collection. In either case, only a lower bound on the failure time of the censored observations is known. These observations are said to be *right censored*. Thus, an additional variable is incorporated into the analysis to indicate which failure times are observed event times and which are censored times. More generally, the failure time might only be known to be smaller than a given value (*left censored*) or known to be within a given interval (*interval censored*). There are numerous possible censoring schemes that arise in survival analysis. The monograph by Maddala (1983) discusses several related types of censoring situations, and the text by Kalbfleisch and Prentice (1980) also discusses several censoring schemes. Data with censored observations cannot be analyzed by ignoring the censored observations because, among other considerations, the longer-lived individuals are generally more likely to be right censored. The method of analysis must take the censoring into account and correctly use the censored observations as well as the uncensored observations.

Another characteristic of survival data is that the response cannot be negative. This suggests that a transformation of the survival time such as a log transformation might be necessary or that specialized methods might be more appropriate than those that assume a normal distribution for the error term. It is especially important to check any underlying assumptions as a part of the analysis because some of the models used are very sensitive to these assumptions.

Survival Analysis Procedures

There are three SAS procedures for analyzing survival data: LIFEREG, LIFETEST, and PHREG. PROC LIFEREG is a parametric regression procedure for modeling the distribution of survival time with a set of concomitant variables. PROC LIFETEST is a nonparametric procedure for estimating the survivor function, comparing the underlying survival curves of two or more samples, and testing the association of survival time with other variables. PROC PHREG is a semiparametric procedure that fits the Cox proportional hazards model and its extensions.

The LIFEREG Procedure

The LIFEREG procedure fits parametric accelerated failure time models to survival data that may be left, right, or interval censored. The parametric model is of the form

$$y = \mathbf{x}'\boldsymbol{\beta} + \sigma\epsilon$$

where y is usually the log of the failure time variable, $\mathbf{x}$ is a vector of covariate values, $\boldsymbol{\beta}$ is a vector of unknown regression parameters, σ is an unknown scale parameter, and ϵ is an error term. The baseline distribution of the error term can be specified as one of several possible distributions including (but not limited to) the log normal, log logistic, and Weibull distributions. Several texts that discuss these parametric models are Kalbfleisch and Prentice (1980), Lawless (1982), Nelson (1990), and Meeker and Escobar (1998). For more information about PROC LIFEREG, see Chapter 48, "The LIFEREG Procedure."

The LIFETEST Procedure

The LIFETEST procedure computes nonparametric estimates of the survival distribution function. You can request either the product-limit (Kaplan and Meier 1958) or the life-table (actuarial) estimate of the distribution. The texts by Cox and Oakes (1984) and Kalbfleisch and Prentice (1980) provide good discussions of the product-limit estimator, and the texts by Lee (1992) and Elandt-Johnson and Johnson (1980) include detailed discussions of the life-table estimator. PROC LIFETEST computes nonparametric tests to compare the survival curves of two or more groups. The procedure also computes rank tests of association of the survival time variable with other concomitant variables as given in Kalbfleisch and Prentice (1980, Chapter 6). For more information about PROC LIFETEST, see Chapter 49, "The LIFETEST Procedure."

The PHREG Procedure

The PHREG procedure fits the proportional hazards model of Cox (1972, 1975) to survival data that might be right censored. The Cox model is a semiparametric model in which the hazard function of the survival time is given by

$$\lambda(t;\mathbf{x}) = \lambda_0(t)e^{\boldsymbol{\beta}'\mathbf{x}(t)}$$

where $\lambda_0(t)$ is an unspecified baseline hazard function, $\mathbf{x}(t)$ is a vector of covariate values (possibly time-dependent), and $\boldsymbol{\beta}$ is a vector of unknown regression parameters. The model is referred to as a semiparametric model because part of the model involves the unspecified baseline function over time (which has an infinite dimension) and the other part involves a finite number of regression parameters. Several texts that discuss the Cox regression models are Collett (1994), Cox and Oakes (1984), Kalbfleisch and Prentice (1980), and Lawless (1982). Extensions of the Cox model are discussed in Therneau and Grambsch (2000), Andersen et al. (1992), and Fleming and Harrington (1991). For more information about PROC PHREG, see Chapter 64, "The PHREG Procedure."

Survival Analysis with SAS/STAT Procedures

The typical goal in survival analysis is to characterize the distribution of the survival time for a given population, to compare the survival distributions among different groups, or to study the relationship between the survival time and some concomitant variables.

A first step in the analysis of a set of survival data is to use PROC LIFETEST to compute and plot the estimate of the distribution of the survival time. In many applications, there are often several survival curves to compare. For example, you want to compare the survival experiences of patients who receive different treatments for their disease. The association between covariates and the survival time variable can be investigated by computing estimates of the survival distribution function within strata defined by the covariates. In particular, if the proportional hazards model is

appropriate, the estimates of the log(-log(SURVIVAL)) plotted against the log(TIME) variable should give approximately parallel lines, where SURVIVAL is the survival distribution estimate and TIME is the failure time variable. Additionally, these lines should be approximately straight if the Weibull model is appropriate.

Statistics that test for association between failure time and covariates can be used to select covariates for further investigation. The LIFETEST procedure computes linear rank statistics using either Wilcoxon or log-rank scores. These statistics and their estimated covariance matrix can be used with the REG procedure with the option METHOD=RSQUARE to find the subset of variables that produce the largest joint test statistic for association. An illustration of this methodology is given in Example 49.1 of Chapter 49, "The LIFETEST Procedure."

Another approach to examining the relationship between the concomitant variables and survival time is through a regression model in which the survival time has a distribution that depends on the concomitant variables. The regression coefficients can be interpreted as describing the direction and strength of the relationship of each explanatory variable on the effect of the survival time.

In many biological systems, the Cox model might be a reasonable description of the relationship between the distribution of the survival time and the prognostic factors. You use PROC PHREG to fit the Cox regression model. The regression coefficient is interpreted as the increase of the log-hazard ratio resulting in the increase of one unit in the covariate. However, the underlying hazard function is left unspecified and, as in any other model, the results can be misleading if the proportional hazards assumptions do not hold.

Accelerated failure time models are popular for survival data of physical systems. In many cases, the underlying survival distribution is known empirically. You use PROC LIFEREG to fit these parametric models. Also, PROC LIFEREG can accommodate data with left-censored or interval-censored observations, which are not allowed in PROC PHREG.

A common technique for checking the validity of a regression model is to embed it in a larger model and use the likelihood ratio test to check whether the reduction to the actual model is valid. Other techniques include examining the residuals. Both PROC LIFEREG and PROC PHREG produce predicted values, residuals, and other computed values that can be used to assess the model adequacy.

Bayesian Survival Analysis with SAS/STAT Procedures

Bayesian analysis of survival models can be requested in the LIFEREG and PHREG procedures. In addition to the Cox model, PROC PHREG also allow you to fit a piecewise exponential model. In Bayesian analysis, the model parameters are treated as random variables, and inference about parameters is based on the posterior distribution of the parameters. A posterior distribution is a weighted likelihood function of the data with a prior distribution of the parameters using the Bayes theorem. The prior distribution enables you to incorporate into the analysis knowledge or experience of the likely range of values of the parameters of interest. You can specify normal or uniform prior distributions for the model regression coefficients in both LIFEREG and PHREG procedures. In addition, you can specify a gamma or improper prior distribution for the scale or variance parameter

in PROC LIFEREG. For the piecewise exponential model in PROC PHREG, you can specify normal or uniform prior distributions for the log-hazard parameters; alternatively, you can specify gamma or improper prior distributions for the hazards parameters. If you have no prior knowledge of the parameter values, you can use a noninformative prior distribution, and the results of a Bayesian analysis will be very similar to a classical analysis based on maximum likelihood.

A closed form of the posterior distribution is often not feasible, and a Markov chain Monte Carlo method by Gibbs sampling is used to simulate samples from the posterior distribution. You can perform inference by using the simulated samples, for example, to estimate the probability that a function of the parameters of interest lies within a specified range of values.

See Chapter 7, "Introduction to Bayesian Analysis Procedures," for an introduction to the basic concepts of Bayesian statistics. Also see "Bayesian Analysis: Advantages and Disadvantages" on page 149 for a discussion of the advantages and disadvantages of Bayesian analysis. See Ibrahim, Chen, and Sinha (2001), Gelman et al. (2004), and Gilks, Richardson, and Spiegelhalter (1996) for more information about Bayesian analysis, including guidance about choosing prior distributions.

References

Andersen, P. K., Borgan, O., Gill, R. D., and Keiding, N. (1992), *Statistical Models Based on Counting Processes*, New York: Springer-Verlag.

Collett, D. (1994), *Modeling Survival Data in Medical Research*, London: Chapman & Hall.

Cox, D. R. (1972), "Regression Models and Life Tables," *Journal of the Royal Statistical Society, Series B*, 20, 187–220, with discussion.

Cox, D. R. (1975), "Partial Likelihood," *Biometrika*, 62, 269–276.

Cox, D. R. and Oakes, D. (1984), *Analysis of Survival Data*, London: Chapman & Hall.

Elandt-Johnson, R. C. and Johnson, N. L. (1980), *Survival Models and Data Analysis*, New York: John Wiley & Sons.

Fleming, T. R. and Harrington, D. (1991), *Counting Processes and Survival Analysis*, New York: John Wiley & Sons.

Gelman, A., Carlin, J., Stern, H., and Rubin, D. (2004), *Bayesian Data Analysis*, Second Edition, London: Chapman & Hall.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996), *Markov Chain Monte Carlo in Practice*, London: Chapman & Hall.

Ibrahim, J. G., Chen, M. H., and Sinha, D. (2001), *Bayesian Survival Analysis*, New York: Springer-Verlag.

Kalbfleisch, J. D. and Prentice, R. L. (1980), *The Statistical Analysis of Failure Time Data*, New York: John Wiley & Sons.

Kaplan, E. L. and Meier, P. (1958), "Nonparametric Estimation from Incomplete Observations," *Journal of the American Statistical Association*, 53, 457–481.

Lawless, J. F. (1982), *Statistical Methods and Methods for Lifetime Data*, New York: John Wiley & Sons.

Lee, E. T. (1992), *Statistical Methods for Survival Data Analysis*, Second Edition, New York: John Wiley & Sons.

Maddala, G. S. (1983), *Limited-Dependent and Qualitative Variables in Econometrics*, New York: Cambridge University Press.

Meeker, W. Q. and Escobar, L. A. (1998), *Statistical Methods for Reliability Data*, New York: John Wiley & Sons.

Nelson, W. (1990), *Accelerated Testing: Statistical Models, Test Plans, and Data Analyses*, New York: John Wiley & Sons.

Therneau, T. M. and Grambsch, P. M. (2000), *Modeling Survival Data: Extending the Cox Model*, New York: Springer-Verlag.

Chapter 14

Introduction to Survey Sampling and Analysis Procedures

Contents

Overview: Survey Sampling and Analysis Procedures **259**
The Survey Procedures **262**
PROC SURVEYSELECT 262
PROC SURVEYMEANS 262
PROC SURVEYFREQ 263
PROC SURVEYREG 263
PROC SURVEYLOGISTIC 264
Survey Design Specification **264**
Variance Estimation **267**
Example: Survey Sampling and Analysis Procedures **268**
References **270**

Overview: Survey Sampling and Analysis Procedures

This chapter introduces the SAS/STAT procedures for survey sampling and describes how you can use these procedures to analyze survey data.

Researchers often use sample survey methodology to obtain information about a large population by selecting and measuring a sample from that population. Due to variability among items, researchers apply scientific probability-based designs to select the sample. This reduces the risk of a distorted view of the population and enables statistically valid inferences to be made from the sample. See Lohr (1999), Kalton (1983), Cochran (1977), and Kish (1965) for more information about statistical sampling and analysis of complex survey data. To select probability-based random samples from a study population, you can use the SURVEYSELECT procedure, which provides a variety of methods for probability sampling. To analyze sample survey data, you can use the SURVEYMEANS, SURVEYFREQ, SURVEYREG, and SURVEYLOGISTIC procedures, which incorporate the sample design into the analyses.

Many SAS/STAT procedures, such as the MEANS, FREQ, GLM and LOGISTIC procedures, can compute sample means, produce crosstabulation tables, and estimate regression relationships. However, in most of these procedures, statistical inference is based on the assumption that the sample is

drawn from an infinite population by simple random sampling. If the sample is in fact selected from a finite population by using a complex survey design, these procedures generally do not calculate the estimates and their variances according to the design actually used. Using analyses that are not appropriate for your sample design can lead to incorrect statistical inferences.

The SURVEYMEANS, SURVEYFREQ, SURVEYREG, and SURVEYLOGISTIC procedures properly analyze complex survey data by taking into account the sample design. These procedures can be used for multistage or single-stage designs, with or without stratification, and with or without unequal weighting. The survey analysis procedures provide a choice of variance estimation methods, which include Taylor series linearization, balanced repeated replication (BRR), and the jackknife.

Table 14.1 briefly describes the sampling and analysis procedures in SAS/STAT software.

Table 14.1 Sampling and Analysis Procedures in SAS/STAT Software

PROC SURVEYSELECT	
Sampling Methods	simple random sampling unrestricted random sampling (with replacement) systematic sequential probability proportional to size (PPS) sampling with and without replacement PPS systematic PPS for two units per stratum PPS sequential with minimum replacement
Allocation Methods	proportional optimal Neyman
PROC SURVEYMEANS	
Statistics	estimates of population means and totals estimates of population proportions estimates of population quantiles ratio estimates standard errors confidence limits hypothesis tests domain analysis

Table 14.1 *continued*

PROC SURVEYFREQ	
Tables	one-way frequency tables two-way and multiway crosstabulation tables estimates of population totals and proportions standard errors confidence limits
Analyses	tests of goodness of fit tests of independence risks and risk differences odds ratios and relative risks
PROC SURVEYREG	
Analyses	linear regression model fitting regression coefficients covariance matrices confidence limits hypothesis tests estimable functions contrasts predicted values and residuals domain analysis
PROC SURVEYLOGISTIC	
Analyses	cumulative logit regression model fitting logit, probit, and complementary log-log link functions generalized logit regression model fitting regression coefficients covariance matrices confidence limits hypothesis tests odds ratios estimable functions contrasts model diagnostics domain analysis

The Survey Procedures

The SURVEYSELECT procedure provides methods for probability sample selection. The SURVEYMEANS, SURVEYFREQ, SURVEYREG, and SURVEYLOGISTIC procedures provide statistical analyses for sample survey data. The following sections contain brief descriptions of these procedures. See the chapters on these procedures for more detailed information.

PROC SURVEYSELECT

The SURVEYSELECT procedure provides a variety of methods for selecting probability-based random samples. The procedure can select a simple random sample or can sample according to a complex multistage sample design that includes stratification, clustering, and unequal probabilities of selection. With probability sampling, each unit in the survey population has a known, positive probability of selection. This property of probability sampling avoids selection bias and enables you to use statistical theory to make valid inferences from the sample to the survey population.

PROC SURVEYSELECT provides methods for both equal probability sampling and probability proportional to size (PPS) sampling. In PPS sampling, a unit's selection probability is proportional to its size measure. PPS sampling is often used in cluster sampling, where you select clusters (groups of sampling units) of varying size in the first stage of selection. Available PPS methods include without replacement, with replacement, systematic, and sequential with minimum replacement. The procedure can apply these methods for stratified and replicated sample designs.

For stratified sampling, PROC SURVEYSELECT provides survey design methods to allocate the total sample size among the strata. Available allocation methods include proportional, Neyman, and optimal allocation. Optimal allocation maximizes the estimation precision within the available resources, taking into account stratum sizes, costs, and variances.

See Chapter 87, "The SURVEYSELECT Procedure," for more information.

PROC SURVEYMEANS

The SURVEYMEANS procedure produces estimates of population means and totals from sample survey data. The procedure also computes estimates of proportions for categorical variables, estimates of quantiles for continuous variables, and ratio estimates of means and proportions. For all of these statistics, PROC SURVEYMEANS provides standard errors, confidence limits, and *t* tests.

PROC SURVEYMEANS provides domain analysis, which computes estimates for subpopulations or domains, in addition to analysis for the entire study population. Formation of subpopulations can be unrelated to the sample design, and so the domain sample sizes can actually be random variables.

Domain analysis takes this variability into account by using the entire sample in estimating the variance of domain estimates. Domain analysis is also known as subgroup analysis, subpopulation analysis, or subdomain analysis.

See Chapter 85, "The SURVEYMEANS Procedure," for more information.

PROC SURVEYFREQ

The SURVEYFREQ procedure produces one-way to *n*-way frequency and crosstabulation tables from sample survey data. These tables include estimates of population totals, population proportions (overall proportions, and also row and column proportions), and corresponding standard errors. Confidence limits, coefficients of variation, and design effects are also available. The procedure provides a variety of options to customize the table display.

For one-way frequency tables, PROC SURVEYFREQ provides Rao-Scott chi-square goodness-of-fit tests, which are adjusted for the sample design. You can test a null hypothesis of equal proportions for a one-way frequency table, or you can input custom null hypothesis proportions for the test. For two-way frequency tables, PROC SURVEYFREQ provides design-adjusted tests of independence, or no association, between the row and column variables. These tests include the Rao-Scott chi-square test, the Rao-Scott likelihood ratio test, the Wald chi-square test, and the Wald log-linear chi-square test.

For 2×2 tables, PROC SURVEYFREQ computes estimates and confidence limits for risks (or row proportions), the risk difference, the odds ratio, and relative risks.

See Chapter 83, "The SURVEYFREQ Procedure," for more information.

PROC SURVEYREG

The SURVEYREG procedure performs regression analysis for sample survey data. The procedure fits linear models and computes regression coefficients and their variance-covariance matrix. The procedure enables you to specify classification effects by using the same syntax as in the GLM procedure.

PROC SURVEYREG provides hypothesis tests for the model effects. The procedure also provides custom hypothesis tests for linear combinations of the regression parameters. The procedure computes confidence limits for the parameter estimates, as well as for any specified linear functions of the regression parameters. The procedure can produce an output data set that contains the predicted values from the linear regression, their standard errors and confidence limits, and the residuals.

PROC SURVEYREG also performs regression analysis for domains or population subgroups.

See Chapter 86, "The SURVEYREG Procedure," for more information.

PROC SURVEYLOGISTIC

The SURVEYLOGISTIC procedure provides logistic regression analysis for sample survey data. Logistic regression analysis investigates the relationship between discrete responses and a set of explanatory variables. PROC SURVEYLOGISTIC fits linear logistic regression models for discrete response survey data by the method of maximum likelihood and incorporates the sample design into the analysis. The SURVEYLOGISTIC procedure enables you to specify categorical classification variables (also known as CLASS variables) as explanatory variables in the model by using the same syntax for main effects and interactions as in the GLM and LOGISTIC procedures.

The following link functions are available for regression in PROC SURVEYLOGISTIC: the cumulative logit function (CLOGIT), the generalized logit function (GLOGIT), the probit function (PROBIT), and the complementary log-log function (CLOGLOG). The procedure performs maximum likelihood estimation of the regression coefficients with either the Fisher scoring algorithm or the Newton-Raphson algorithm.

PROC SURVEYLOGISTIC also performs logistic regression analysis for domains or subgroups.

See Chapter 84, "The SURVEYLOGISTIC Procedure," for more information.

Survey Design Specification

Survey sampling is the process of selecting a probability-based sample from a finite population according to a sample design. You then collect data from these selected units and use them to estimate characteristics of the entire population.

A *sample design* encompasses the rules and operations by which you select sampling units from the population and the computation of sample statistics, which are estimates of the population values of interest. The objective of your survey often determines appropriate sample designs and valid data collection methodology. A complex sample design can include stratification, clustering, multiple stages of selection, and unequal weighting. The survey procedures can be used for single-stage designs or for multistage designs, with or without stratification, and with or without unequal weighting.

To analyze your survey data with the SURVEYMEANS, SURVEYFREQ, SURVEYREG, and SURVEYLOGISTIC procedures, you need to specify sample design information for the procedures. This information can include design strata, clusters, and sampling weights. All the survey analysis procedures use the same syntax for specifying sample design information. You provide sample design information with the STRATA, CLUSTER, and WEIGHT statements, and with the RATE= or TOTAL= option in the PROC statement.

If you provide replicate weights for BRR or jackknife variance estimation, you do not need to specify a STRATA or CLUSTER statement. Otherwise, you should specify STRATA and CLUSTER statements whenever your design includes stratification and clustering.

When there are clusters, or PSUs, in the sample design, the procedures estimate variance by using the PSUs, as described in the section "Variance Estimation" on page 267. For a multistage sample design, the variance estimation depends only on the first stage of the sample design. So the required input includes only first-stage cluster (PSU) and first-stage stratum identification. You do not need to input design information about any additional stages of sampling.

The following sections provide brief descriptions of basic sample design concepts and terminology used in the survey procedures. See Lohr (1999), Kalton (1983), Cochran (1977), and Kish (1965) for more detailed information.

Population

Population refers to the target population or group of individuals of interest for study. Often, the primary objective is to estimate certain characteristics of this population, called *population values*. A *sampling unit* is an element or an individual in the target population. A *sample* is a subset of the population that is selected for the study.

Before you use the survey procedures, you should have a well-defined target population, sampling units, and an appropriate sample design.

In order to select a sample according to your sample design, you need to have a list of sampling units in the population. This is called a *sampling frame*. PROC SURVEYSELECT selects a sample by using this sampling frame.

Stratification

Stratified sampling involves selecting samples independently within strata, which are nonoverlapping subgroups of the survey population. Stratification controls the distribution of the sample size in the strata. It is widely used to meet a variety of survey objectives. For example, with stratification you can ensure adequate sample sizes for subgroups of interest, including small subgroups, or you can use stratification to improve the precision of overall estimates. To improve precision, units within strata should be as homogeneous as possible for the characteristics of interest.

Clustering

Cluster sampling involves selecting clusters, which are groups of sampling units. For example, clusters might be schools, hospitals, or geographical areas, and sampling units might be students, patients, or citizens. Cluster sampling can provide efficiency in frame construction and other survey operations. However, it can also result in a loss in precision of your estimates, compared to a nonclustered sample of the same size. To minimize this effect, units within clusters should be as heterogeneous as possible for the characteristics of interest.

Multistage Sampling

In *multistage sampling*, you select an initial or first-stage sample based on groups of elements in the population, called *primary sampling units* or *PSUs*.

Then you create a second-stage sample by drawing a subsample from each selected PSU in the first-stage sample. By repeating this operation, you can select a higher-stage sample. If you include all the elements from the selected primary sampling units, then the two-stage sample is a cluster sample.

Sampling Weights

Sampling weights, or *survey weights*, are positive values associated with the units in your sample. Ideally, the weight of a sampling unit should be the "frequency" that the sampling unit represents in the target population.

Often, sampling weights are the reciprocals of the selection probabilities for the sampling units. When you use PROC SURVEYSELECT, the procedure generates the sampling weight component for each stage of the design, and you can multiply these sampling weight components to obtain the final sampling weights. Sometimes, sampling weights also include nonresponse adjustments, postsampling stratification, or regression adjustments by using supplemental information.

When the sampling units have unequal weights, you must provide the weights to the survey analysis procedures. If you do not specify sampling weights, the procedures use equal weights in the analyses.

Population Totals and Sampling Rates

If you use Taylor series variance estimation, the survey procedures include a finite population correction factor in the analysis if you input either the sampling rate or the population total.

The sampling rate is the ratio of the sample size (the number of sampling units in the sample) n to the population size (the total number of sampling units in the target population) N, $f = n/N$. This ratio is also called the *sampling fraction*. If you select a sample without replacement, the extra efficiency compared to selecting a sample with replacement can be measured by the *finite population correction* (*fpc*) factor, $(1 - f)$.

If your analysis includes a finite population correction factor, you can input either the sampling rate or the population total. Otherwise, the procedures do not use the *fpc* in computing variance estimates. For fairly small sampling fractions, it is appropriate to ignore this correction. See Cochran (1977) and Kish (1965) for details.

As discussed in the section "Variance Estimation" on page 267, for a multistage sample design, the variance estimation depends only on the first stage of the sample design. Therefore, if you are specifying the sampling rate, you should input the *first-stage sampling rate*, which is the ratio of the number of PSUs in the sample to the total number of PSUs in the target population.

If you use BRR or jackknife variance estimate, the procedures do not include a finite population correction in the analysis, and so you do not need to input the sampling rate or the population total.

Variance Estimation

The survey analysis procedures provide a choice of variance estimation methods for complex survey designs. In addition to the Taylor series linearization method, the procedures offer two replication-based or resampling methods—balanced repeated replication (BRR) and the delete-1 jackknife. These variance estimation methods usually give similar, satisfactory results (Lohr 1999; Särndal, Swensson, and Wretman 1992; Wolter 1985). The choice of a variance estimation method can depend on the sample design used, the sample design information available, the parameters to be estimated, and computational issues. See Lohr (1999) for more details.

The Taylor series linearization method is appropriate for all designs where the first-stage sample is selected with replacement, or where the first-stage sampling fraction is small, as it often is in practice. The Taylor series method obtains a linear approximation for the estimator and then uses the variance estimate for this approximation to estimate the variance of the estimate itself (Fuller 1975; Woodruff 1971). When there are clusters, or primary sampling units (PSUs), in the sample design, the procedures estimate the variance from the variation among the PSUs. When the design is stratified, the procedures pool stratum variance estimates to compute the overall variance estimate.

For a multistage sample design, the Taylor series variance method depends only on the first stage of the sample design. So the required input includes only first-stage cluster (PSU) and first-stage stratum identification. You do not need to input design information about any additional stages of sampling.

Replication methods for variance estimation draw multiple replicates (or subsamples) from the full sample by following a specific resampling scheme. Commonly used resampling schemes include *balanced repeated replication* (BRR) and the *jackknife*. The parameter of interest is estimated from each replicate, and the variability among the replicate estimates is used to estimate the overall variance of the parameter estimate.

The BRR variance estimation method requires a stratified sample design with two PSUs in each stratum. Each replicate is obtained by deleting one PSU per stratum according to the corresponding Hadamard matrix and adjusting the original weights for the remaining PSUs. The adjusted weights are called *replicate weights*. The survey procedures also provide Fay's method, which is a modification of the BRR method.

The jackknife method deletes one PSU at a time from the full sample to create replicates, and modifies the original weights to obtain replicate weights. The total number of replicates equals the number of PSUs. If the sample design is stratified, each stratum must contain at least two PSUs, and the jackknife is applied separately within each stratum.

Instead of having the survey procedures generate replicate weights for the analysis, you can directly input your own replicate weights. This can be useful if you need to do multiple analyses with the same set of replicate weights, or if you have access to replicate weights without complete design information.

See the chapters on the survey procedures for complete details. For more information about variance estimation for sample survey data, see Lohr (1999); Wolter (1985); Särndal, Swensson, and Wretman (1992); Lee, Forthoffer, and Lorimor (1989); Cochran (1977); Kish (1965); and Hansen, Hurwitz, and Madow (1953).

Example: Survey Sampling and Analysis Procedures

This section demonstrates how you can use the survey procedures to select a probability-based sample and then analyze the survey data to make inferences about the population. The analyses include descriptive statistics and regression analysis. This example is a survey of income and expenditures for a group of households in North Carolina and South Carolina. The goals of the survey are as follows:

- to estimate total income and total living expenses
- to estimate the median income and the median living expenses
- to investigate the linear relationship between income and living expenses

Sample Selection

To select a sample with PROC SURVEYSELECT, you input a SAS data set that contains the sampling frame or list of units from which the sample is to be selected. You also specify the selection method, the desired sample size or sampling rate, and other selection parameters. PROC SURVEYSELECT selects the sample and produces an output data set that contains the selected units, their selection probabilities, and their sampling weights. See Chapter 87, "The SURVEYSELECT Procedure," for more details about PROC SURVEYSELECT.

In this example, the sample design is a stratified sample design, with households as the sampling units and selection by simple random sampling. The SAS data set HHFrame contains the sampling frame, which is the list of households in the survey population. The sampling frame is stratified by the variables State and Region. Within strata, households are selected by simple random sampling. The following PROC SURVEYSELECT statements select a probability sample of households according to this sample design:

```
proc surveyselect data=HHFrame out=HHSample
                  method=srs n=(3, 5, 3, 6, 2);
   strata State Region;
run;
```

The STRATA statement names the stratification variables State and Region. In the PROC SURVEYSELECT statement, the DATA= option names the SAS data set HHFrame as the input data set (or sampling frame) from which to select the sample. The OUT= option stores the sample in the SAS data set named HHSample. The METHOD=SRS option specifies simple random sampling as the sample selection method. The N= option specifies the stratum sample sizes.

The SURVEYSELECT procedure then selects a stratified random sample of households and produces the output data set HHSample, which contains the selected households together with their selection probabilities and sampling weights. The data set HHSample also contains the sampling unit identification variable Id and the stratification variables State and Region from the input data set HHFrame.

Survey Data Analysis

You can use the SURVEYMEANS and SURVEYREG procedures to estimate population values and perform regression analyses for survey data. The following example briefly shows the capabilities of these procedures. See Chapter 85, "The SURVEYMEANS Procedure," and Chapter 86, "The SURVEYREG Procedure," for more information.

The following PROC SURVEYMEANS statements estimate the total income and living expenses for the survey population based on the data from the stratified sample design:

```
proc surveymeans data=HHSample sum median;
   var Income Expense;
   strata State Region;
   weight Weight;
run;
```

The PROC SURVEYMEANS statement invokes the procedure, and the DATA= option names the SAS data set HHSample as the input data set to be analyzed. The keywords SUM and MEDIAN request estimates of population totals and medians.

The VAR statement specifies the two analysis variables Income and Expense. The STRATA statement names the stratification variables State and Region. The WEIGHT statement specifies the sampling weight variable Weight.

You can use PROC SURVEYREG to perform regression analysis for survey data. Suppose that, in order to explore the relationship between household income and living expenses in the survey population, you choose the following linear model:

$$\text{Expense} = \alpha + \beta * \text{Income} + \text{error}$$

The following PROC SURVEYREG statements fit this linear model for the survey population based on the data from the stratified sample design:

```
proc surveyreg data=HHSample;
   strata State Region ;
   model  Expense = Income;
   weight Weight;
run;
```

The STRATA statement names the stratification variables State and Region. The MODEL statement specifies the model, with Expense as the dependent variable and Income as the independent variable. The WEIGHT statement specifies the sampling weight variable Weight.

References

Cochran, W. G. (1977), *Sampling Techniques*, Third Edition, New York: John Wiley & Sons.

Fuller, W. A. (1975), "Regression Analysis for Sample Survey," *Sankhyā*, 37 (3), Series C, 117–132.

Hansen, M. H., Hurwitz, W. N., and Madow, W. G. (1953), *Sample Survey Methods and Theory*, Volumes I and II, New York: John Wiley & Sons.

Kalton, G. (1983), *Introduction to Survey Sampling*, Sage University Paper series on Quantitative Applications in the Social Sciences, series no. 07-035, Beverly Hills, CA, and London: Sage Publications.

Kish, L. (1965), *Survey Sampling*, New York: John Wiley & Sons.

Lee, E. S., Forthoffer, R. N., and Lorimor, R. J. (1989), *Analyzing Complex Survey Data*, Sage University Paper series on Quantitative Applications in the Social Sciences, series no. 07-071, Beverly Hills, CA, and London: Sage Publications.

Lohr, S. L. (1999), *Sampling: Design and Analysis*, Pacific Grove, CA: Duxbury Press.

Särndal, C. E., Swensson, B., and Wretman, J. (1992), *Model Assisted Survey Sampling*, New York: Springer-Verlag.

Wolter, K. M. (1985), *Introduction to Variance Estimation*, New York: Springer-Verlag.

Woodruff, R. S. (1971), "A Simple Method for Approximating the Variance of a Complicated Estimate," *Journal of the American Statistical Association*, 66, 411–414.

Chapter 15

The Four Types of Estimable Functions

Contents

Overview **271**
Estimability **271**
General Form of an Estimable Function 272
Introduction to Reduction Notation 274
Examples 275
Estimable Functions **278**
Type I SS and Estimable Functions 278
Type II SS and Estimable Functions 279
Type III and IV SS and Estimable Functions 283
References **288**

Overview

Many regression and analysis of variance procedures in SAS/STAT label tests for various effects in the model as Type I, Type II, Type III, or Type IV. These four types of hypotheses might not always be sufficient for a statistician to perform all desired inferences, but they should suffice for the vast majority of analyses. This chapter explains the hypotheses involved in each of the four test types. For additional discussion, see Freund, Littell, and Spector (1991) or Milliken and Johnson (1984).

The primary context of the discussion is testing linear hypotheses in least squares regression and analysis of variance, such as with PROC GLM. In this context, tests correspond to hypotheses about linear functions of the true parameters and are evaluated using sums of squares of the estimated parameters. Thus, there will be frequent references to Type I, II, III, and IV (estimable) functions and corresponding Type I, II, III, and IV sums of squares, or simply SS.

Estimability

Given a response or dependent variable $\mathbf{Y}$, predictors or independent variables $\mathbf{X}$, and a linear expectation model $\mathrm{E}[\mathbf{Y}] = \mathbf{X}\boldsymbol{\beta}$ relating the two, a primary analytical goal is to estimate or test for

the significance of certain linear combinations of the elements of $\boldsymbol{\beta}$. For least squares regression and analysis of variance, this is accomplished by computing linear combinations of the observed $\mathbf{Y}$s. An unbiased linear estimate of a specific linear function of the individual βs, say $\mathbf{L}\boldsymbol{\beta}$, is a linear combination of the $\mathbf{Y}$s that has an expected value of $\mathbf{L}\boldsymbol{\beta}$. Hence, the following definition:

> A linear combination of the parameters $\mathbf{L}\boldsymbol{\beta}$ is estimable if and only if a linear combination of the $\mathbf{Y}$s exists that has expected value $\mathbf{L}\boldsymbol{\beta}$.

Any linear combination of the $\mathbf{Y}$s, for instance $\mathbf{KY}$, will have expectation $\mathrm{E}[KY] = \mathbf{KX}\boldsymbol{\beta}$. Thus, the expected value of any linear combination of the $\mathbf{Y}$s is equal to that same linear combination of the rows of $\mathbf{X}$ multiplied by $\boldsymbol{\beta}$. Therefore,

> $\mathbf{L}\boldsymbol{\beta}$ is estimable if and only if there is a linear combination of the rows of $\mathbf{X}$ that is equal to $\mathbf{L}$—that is, if and only if there is a $\mathbf{K}$ such that $\mathbf{L} = \mathbf{KX}$.

Thus, the rows of $\mathbf{X}$ form a generating set from which any estimable $\mathbf{L}$ can be constructed. Since the row space of $\mathbf{X}$ is the same as the row space of $\mathbf{X}'\mathbf{X}$, the rows of $\mathbf{X}'\mathbf{X}$ also form a generating set from which all estimable $\mathbf{L}$s can be constructed. Similarly, the rows of $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$ also form a generating set for $\mathbf{L}$.

Therefore, if $\mathbf{L}$ can be written as a linear combination of the rows of $\mathbf{X}$, $\mathbf{X}'\mathbf{X}$, or $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$, then $\mathbf{L}\boldsymbol{\beta}$ is estimable.

In the context of least squares regression and analysis of variance, an estimable linear function $\mathbf{L}\boldsymbol{\beta}$ can be estimated by $\mathbf{L}\widehat{\boldsymbol{\beta}}$, where $\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{Y}$. From the general theory of linear models, the unbiased estimator $\mathbf{L}\widehat{\boldsymbol{\beta}}$ is, in fact, the *best* linear unbiased estimator of $\mathbf{L}\boldsymbol{\beta}$, in the sense of having minimum variance as well as maximum likelihood when the residuals are normal. To test the hypothesis that $\mathbf{L}\boldsymbol{\beta} = \mathbf{0}$, compute the sum of squares

$$\mathrm{SS}(H_0\colon\ \mathbf{L}\boldsymbol{\beta} = \mathbf{0}) = (\mathbf{L}\widehat{\boldsymbol{\beta}})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}\mathbf{L}\widehat{\boldsymbol{\beta}}$$

and form an F test with the appropriate error term. Note that in contexts more general than least squares regression (for example, generalized and/or mixed linear models), linear hypotheses are often tested by analogous sums of squares of the estimated linear parameters $(\mathbf{L}\widehat{\boldsymbol{\beta}})'(\mathrm{Var}[\mathbf{L}\widehat{\boldsymbol{\beta}}])^{-1}\mathbf{L}\widehat{\boldsymbol{\beta}}$.

General Form of an Estimable Function

This section demonstrates a shorthand technique for displaying the generating set for any estimable $\mathbf{L}$. Suppose

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix} \quad \text{and} \quad \boldsymbol{\beta} = \begin{bmatrix} \mu \\ A_1 \\ A_2 \\ A_3 \end{bmatrix}$$

$\mathbf{X}$ is a generating set for $\mathbf{L}$, but so is the smaller set

$$\mathbf{X}^* = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

$\mathbf{X}^*$ is formed from $\mathbf{X}$ by deleting duplicate rows.

Since all estimable $\mathbf{L}$s must be linear functions of the rows of $\mathbf{X}^*$ for $\mathbf{L}\boldsymbol{\beta}$ to be estimable, an $\mathbf{L}$ for a single-degree-of-freedom estimate can be represented symbolically as

$$L1 \times (1\ 1\ 0\ 0) + L2 \times (1\ 0\ 1\ 0) + L3 \times (1\ 0\ 0\ 1)$$

or

$$\mathbf{L} = (L1 + L2 + L3,\ L1,\ L2,\ L3)$$

For this example, $\mathbf{L}\boldsymbol{\beta}$ is estimable if and only if the first element of $\mathbf{L}$ is equal to the sum of the other elements of $\mathbf{L}$ or if

$$\mathbf{L}\boldsymbol{\beta} = (L1 + L2 + L3) \times \mu + L1 \times A_1 + L2 \times A_2 + L3 \times A_3$$

is estimable for any values of $L1$, $L2$, and $L3$.

If other generating sets for $\mathbf{L}$ are represented symbolically, the symbolic notation looks different. However, the inherent nature of the rules is the same. For example, if row operations are performed on $\mathbf{X}^*$ to produce an identity matrix in the first 3×3 submatrix of the resulting matrix

$$\mathbf{X}^{**} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

then $\mathbf{X}^{**}$ is also a generating set for $\mathbf{L}$. An estimable $\mathbf{L}$ generated from $\mathbf{X}^{**}$ can be represented symbolically as

$$\mathbf{L} = (L1,\ L2,\ L3,\ L1 - L2 - L3)$$

Note that, again, the first element of $\mathbf{L}$ is equal to the sum of the other elements.

With multiple generating sets available, the question arises as to which one is the best to represent $\mathbf{L}$ symbolically. Clearly, a generating set containing a minimum of rows (of full row rank) and a maximum of zero elements is desirable.

The generalized g_2-inverse $(\mathbf{X}'\mathbf{X})^-$ of $\mathbf{X}'\mathbf{X}$ computed by the modified sweep operation (Goodnight 1979) has the property that $(\mathbf{X}'\mathbf{X})^-\mathbf{X}'\mathbf{X}$ usually contains numerous zeros. For this reason, in PROC GLM the nonzero rows of $(\mathbf{X}'\mathbf{X})^-\mathbf{X}'\mathbf{X}$ are used to represent $\mathbf{L}$ symbolically.

If the generating set represented symbolically is of full row rank, the number of symbols $(L1, L2, \ldots)$ represents the maximum rank of any testable hypothesis (in other words, the maximum number of linearly independent rows for any $\mathbf{L}$ matrix that can be constructed). By letting each symbol in turn take on the value of 1 while the others are set to 0, the original generating set can be reconstructed.

Introduction to Reduction Notation

Reduction notation can be used to represent differences in sums of squares (SS) for two models. The notation $R(\mu, A, B, C)$ denotes the complete main-effects model for effects A, B, and C. The notation

$$R(A \mid \mu, B, C)$$

denotes the difference between the model SS for the complete main-effects model containing A, B, and C and the model SS for the reduced model containing only B and C.

In other words, this notation represents the differences in model SS produced by

```
proc glm;
   class a b c;
   model y = a b c;
run;
```

and

```
proc glm;
   class b c;
   model y = b c;
run;
```

As another example, consider a regression equation with four independent variables. The notation $R(\beta_3, \beta_4 \mid \beta_1, \beta_2)$ denotes the differences in model SS between

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4 + \epsilon$$

and

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

This is the difference in the model SS for the models produced, respectively, by

```
model y = x1 x2 x3 x4;
```

and

```
model y = x1 x2;
```

The following examples demonstrate the ability to manipulate the symbolic representation of a generating set. Note that any operations performed on the symbolic notation have corresponding row operations that are performed on the generating set itself.

Examples

A One-Way Classification Model

For the model

$$Y = \mu + A_i + \epsilon \qquad i = 1, 2, 3$$

the general form of estimable functions $\mathbf{L}\boldsymbol{\beta}$ is (from the previous example)

$$\mathbf{L}\boldsymbol{\beta} = L1 \times \mu + L2 \times A_1 + L3 \times A_2 + (L1 - L2 - L3) \times A_3$$

Thus,

$$\mathbf{L} = (L1, L2, L3, L1 - L2 - L3)$$

Tests involving only the parameters A_1, A_2, and A_3 must have an $\mathbf{L}$ of the form

$$\mathbf{L} = (0, L2, L3, -L2 - L3)$$

Since this $\mathbf{L}$ for the A parameters involves only two symbols, hypotheses with at most two degrees of freedom can be constructed. For example, letting $(L2, L3)$ be $(1, 0)$ and $(0, 1)$, respectively, yields

$$\mathbf{L} = \begin{bmatrix} 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

The preceding $\mathbf{L}$ can be used to test the hypothesis that $A_1 = A_2 = A_3$. For this example, any $\mathbf{L}$ with two linearly independent rows with column 1 equal to zero produces the same sum of squares. For example, a joint test for linear and quadratic effects of A

$$\mathbf{L} = \begin{bmatrix} 0 & 1 & 0 & -1 \\ 0 & 1 & -2 & 1 \end{bmatrix}$$

gives the same SS. In fact, for any $\mathbf{L}$ of full row rank and any nonsingular matrix $\mathbf{K}$ of conformable dimensions,

$$\text{SS}(H_0\colon\ \mathbf{L}\boldsymbol{\beta} = 0) = \text{SS}(H_0\colon\ \mathbf{KL}\boldsymbol{\beta} = 0)$$

A Three-Factor Main-Effects Model

Consider a three-factor main-effects model involving the CLASS variables A, B, and C, as shown in Table 15.1.

Table 15.1 Three-Factor Main-Effects Model

Obs	A	B	C
1	1	2	1
2	1	1	2
3	2	1	3
4	2	2	2
5	2	2	2

The general form of an estimable function is shown in Table 15.2.

Table 15.2 General Form of an Estimable Function for Three-Factor Main-Effects Model

Parameter	Coefficient
μ (Intercept)	$L1$
$A1$	$L2$
$A2$	$L1 - L2$
$B1$	$L4$
$B2$	$L1 - L4$
$C1$	$L6$
$C2$	$L1 + L2 - L4 - 2 \times L6$
$C3$	$-L2 + L4 + L6$

Since only four symbols ($L1$, $L2$, $L4$, and $L6$) are involved, any testable hypothesis will have at most four degrees of freedom. If you form an $\mathbf{L}$ matrix with four linearly independent rows according to the preceding rules, then testing $\mathbf{L}\boldsymbol{\beta} = \mathbf{0}$ is equivalent to testing that E[$\mathbf{Y}$] is uniformly 0. Symbolically,

$$\text{SS}(H_0\text{: } \mathbf{L}\boldsymbol{\beta} = 0) = R(\mu, A, B, C)$$

In a main-effects model, the usual hypothesis of interest for a main effect is the equality of all the parameters. In this example, it is not possible to unambiguously test such a hypothesis because of confounding: any test for the equality of the parameters for any one of A, B, or C will necessarily involve the parameters for the other two effects. One way to proceed is to construct a maximum rank hypothesis (MRH) involving only the parameters of the main effect in question. This can be done using the general form of estimable functions. Note the following:

- To get an MRH involving only the parameters of A, the coefficients of $\mathbf{L}$ associated with μ, $B1$, $B2$, $C1$, $C2$, and $C3$ must be equated to zero. Starting at the top of the general form, let $L1 = 0$, then $L4 = 0$, then $L6 = 0$. If $C2$ and $C3$ are not to be involved, then $L2$ must also be zero. Thus, $A1 - A2$ is not estimable; that is, the MRH involving only the A parameters has zero rank and $R(A \mid \mu, B, C) = 0$.
- To obtain the MRH involving only the B parameters, let $L1 = L2 = L6 = 0$. But then to remove $C2$ and $C3$ from the comparison, $L4$ must also be set to 0. Thus, $B1 - B2$ is not estimable and $R(B \mid \mu, A, C) = 0$.
- To obtain the MRH involving only the C parameters, let $L1 = L2 = L4 = 0$. Thus, the MRH involving only C parameters is

  $$C1 - 2 \times C2 + C3 = K \qquad \text{(for any } K\text{)}$$

  or any multiple of the left-hand side equal to K. Furthermore,

  $$\text{SS}(H_0\text{: } C1 - 2 \times C2 + C3 = 0) = R(C \mid \mu, A, B)$$

A Multiple Regression Model

Suppose

$$\mathrm{E}[Y] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

where the $\mathbf{X}'\mathbf{X}$ matrix has full rank. The general form of estimable functions is as shown in Table 15.3.

Table 15.3 General Form of Estimable Functions for a Multiple Regression Model When $\mathbf{X}'\mathbf{X}$ Matrix Is of Full Rank

Parameter	Coefficient
β_0	$L1$
β_1	$L2$
β_2	$L3$
β_3	$L4$

For example, to test the hypothesis that $\beta_2 = 0$, let $L1 = L2 = L4 = 0$ and let $L3 = 1$. Then $\mathrm{SS}(\mathbf{L}\boldsymbol{\beta} = \mathbf{0}) = R(\beta_2 \mid \beta_0, \beta_1, \beta_3)$. In this full-rank case, all parameters, as well as any linear combination of parameters, are estimable.

Suppose, however, that $X3 = 2x_1 + 3x_2$. The general form of estimable functions is shown in Table 15.4.

Table 15.4 General Form of Estimable Functions for a Multiple Regression Model When $\mathbf{X}'\mathbf{X}$ Matrix Is Not of Full Rank

Parameter	Coefficient
β_0	$L1$
β_1	$L2$
β_2	$L3$
β_3	$2 \times L2 + 3 \times L3$

For this example, it is possible to test $H_0: \beta_0 = 0$. However, β_1, β_2, and β_3 are not jointly estimable; that is,

$$\begin{aligned} R(\beta_1 \mid \beta_0, \beta_2, \beta_3) &= 0 \\ R(\beta_2 \mid \beta_0, \beta_1, \beta_3) &= 0 \\ R(\beta_3 \mid \beta_0, \beta_1, \beta_2) &= 0 \end{aligned}$$

Estimable Functions

Type I SS and Estimable Functions

In PROC GLM, the Type I SS and the associated hypotheses they test are byproducts of the modified sweep operator used to compute a generalized g_2-inverse of $\mathbf{X}'\mathbf{X}$ and a solution to the normal equations. For the model $\mathrm{E}[Y] = x_1\beta_1 + x_2\beta_2 + x_3\beta_3$, the Type I SS for each effect are as follows:

Effect	Type I SS
x_1	$R(\beta_1)$
x_2	$R(\beta_2 \mid \beta_1)$
x_3	$R(\beta_3 \mid \beta_1, \beta_2)$

Note that some other SAS/STAT procedures compute Type I hypotheses by sweeping $\mathbf{X}'\mathbf{X}$ (for example, PROC MIXED and PROC GLIMMIX), but their test statistics are not necessarily equivalent to the results of using those procedures to fit models that contain successively more effects.

The Type I SS are model-order dependent; each effect is adjusted only for the preceding effects in the model.

There are numerous ways to obtain a Type I hypothesis matrix $\mathbf{L}$ for each effect. One way is to form the $\mathbf{X}'\mathbf{X}$ matrix and then reduce $\mathbf{X}'\mathbf{X}$ to an upper triangular matrix by row operations, skipping over any rows with a zero diagonal. The nonzero rows of the resulting matrix associated with x_1 provide an $\mathbf{L}$ such that

$$\mathrm{SS}(H_0\colon\ \mathbf{L}\boldsymbol{\beta} = \mathbf{0}) = R(\beta_1)$$

The nonzero rows of the resulting matrix associated with x_2 provide an $\mathbf{L}$ such that

$$\mathrm{SS}(H_0\colon\ \mathbf{L}\boldsymbol{\beta} = \mathbf{0}) = R(\beta_2 \mid \beta_1)$$

The last set of nonzero rows (associated with x_3) provide an $\mathbf{L}$ such that

$$\mathrm{SS}(H_0\colon\ \mathbf{L}\boldsymbol{\beta} = \mathbf{0}) = R(\beta_3 \mid \beta_1, \beta_2)$$

Another more formalized representation of Type I generating sets for x_1, x_2, and x_3, respectively, is

$$\begin{aligned}
\mathbf{G}_1 &= (\ \mathbf{X}_1'\mathbf{X}_1 \ \mid \ \mathbf{X}_1'\mathbf{X}_2 \ \mid \ \mathbf{X}_1'\mathbf{X}_3 \ ) \\
\mathbf{G}_2 &= (\ 0 \ \mid \ \mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2 \ \mid \ \mathbf{X}_2'\mathbf{M}_1\mathbf{X}_3 \ ) \\
\mathbf{G}_3 &= (\ 0 \ \mid \ 0 \ \mid \ \mathbf{X}_3'\mathbf{M}_2\mathbf{X}_3 \ )
\end{aligned}$$

where

$$\mathbf{M}_1 = \mathbf{I} - \mathbf{X}_1(\mathbf{X}_1'\mathbf{X}_1)^{-}\mathbf{X}_1'$$

and

$$\mathbf{M}_2 = \mathbf{M}_1 - \mathbf{M}_1\mathbf{X}_2(\mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2)^-\mathbf{X}_2'\mathbf{M}_1$$

Using the Type I generating set $\mathbf{G}_2$ (for example), if an $\mathbf{L}$ is formed from linear combinations of the rows of $\mathbf{G}_2$ such that $\mathbf{L}$ is of full row rank and of the same row rank as $\mathbf{G}_2$, then SS(H_0: $\mathbf{L}\boldsymbol{\beta} = \mathbf{0}$) = $R(\beta_2 \mid \beta_1)$.

In the GLM procedure, the Type I estimable functions displayed symbolically when the E1 option is requested are

$$\begin{aligned}
\mathbf{G}_1^* &= (\mathbf{X}_1'\mathbf{X}_1)^-\mathbf{G}_1 \\
\mathbf{G}_2^* &= (\mathbf{X}_2'\mathbf{M}_1\mathbf{X}_2)^-\mathbf{G}_2 \\
\mathbf{G}_3^* &= (\mathbf{X}_3'\mathbf{M}_2\mathbf{X}_3)^-\mathbf{G}_3
\end{aligned}$$

As can be seen from the nature of the generating sets $\mathbf{G}_1$, $\mathbf{G}_2$, and $\mathbf{G}_3$, only the Type I estimable functions for β_3 are guaranteed not to involve the β_1 and β_2 parameters. The Type I hypothesis for β_2 can (and often does) involve β_3 parameters, and likewise the Type I hypothesis for β_1 often involves β_2 and β_3 parameters.

There are, however, a number of models for which the Type I hypotheses are considered appropriate. These are as follows:

- balanced ANOVA models specified in proper sequence (that is, interactions do not precede main effects in the MODEL statement and so forth)
- purely nested models (specified in the proper sequence)
- polynomial regression models (in the proper sequence)

Type II SS and Estimable Functions

For main-effects models and regression models, the general form of estimable functions can be manipulated to provide tests of hypotheses involving only the parameters of the effect in question. The same result can also be obtained by entering each effect in turn as the last effect in the model and obtaining the Type I SS for that effect. These are the *Type II SS*. Using a modified reversible sweep operator, it is possible to obtain the Type II SS without actually refitting the model.

Thus, the **Type II SS correspond to the R notation in which each effect is adjusted for all other appropriate effects**. For a regression model such as

$$\mathrm{E}[Y] = x_1\beta_1 + x_2\beta_2 + x_3\beta_3$$

the Type II SS correspond to

Effect	Type II SS
x_1	$R(\beta_1 \mid \beta_2, \beta_3)$
x_2	$R(\beta_2 \mid \beta_1, \beta_3)$
x_3	$R(\beta_3 \mid \beta_1, \beta_2)$

For a main-effects model (A, B, and C as classification variables), the Type II SS correspond to

Effect	Type II SS
A	$R(A \mid B, C)$
B	$R(B \mid A, C)$
C	$R(C \mid A, B)$

As the discussion in the section "A Three-Factor Main-Effects Model" on page 275 indicates, for regression and main-effects models the Type II SS provide an MRH for each effect that does not involve the parameters of the other effects.

In order to see what effects are appropriate to adjust for in computing Type II estimable functions, note that for models involving interactions and nested effects, in the absence of a priori parametric restrictions, it is not possible to obtain a test of a hypothesis for a main effect free of parameters of higher-level interactions effects with which the main effect is involved. It is reasonable to assume, then, that any test of a hypothesis concerning an effect should involve the parameters of that effect and only those other parameters with which that effect is involved. The concept of effect containment helps to define this involvement.

Contained Effect

Given two effects $F1$ and $F2$, $F1$ is said to be *contained in* $F2$ provided that the following two conditions are met:

- Both effects involve the same continuous variables (if any).
- $F2$ has more CLASS variables than $F1$ does, and if $F1$ has CLASS variables, they all appear in $F2$.

Note that the intercept effect μ is contained in all pure CLASS effects, but it is not contained in any effect involving a continuous variable. No effect is contained by μ.

Type II, Type III, and Type IV estimable functions rely on this definition, and they all have one thing in common: the estimable functions involving an effect $F1$ also involve the parameters of all effects that contain $F1$, and they do not involve the parameters of effects that do not contain $F1$ (other than $F1$).

Hypothesis Matrix for Type II Estimable Functions

The Type II estimable functions for an effect $F1$ have an $\mathbf{L}$ (before reduction to full row rank) of the following form:

- All columns of $\mathbf{L}$ associated with effects not containing $F1$ (except $F1$) are zero.
- The submatrix of $\mathbf{L}$ associated with effect $F1$ is $(\mathbf{X}_1'\mathbf{M}\mathbf{X}_1)^-(\mathbf{X}_1'\mathbf{M}\mathbf{X}_1)$.
- Each of the remaining submatrices of $\mathbf{L}$ associated with an effect $F2$ that contains $F1$ is $(\mathbf{X}_1'\mathbf{M}\mathbf{X}_1)^-(\mathbf{X}_1'\mathbf{M}\mathbf{X}_2)$.

In these submatrices,

$$\begin{aligned}
\mathbf{X}_0 &= \text{the columns of } \mathbf{X} \text{ whose associated effects do not contain } F1 \\
\mathbf{X}_1 &= \text{the columns of } \mathbf{X} \text{ associated with } F1 \\
\mathbf{X}_2 &= \text{the columns of } \mathbf{X} \text{ associated with an } F2 \text{ effect that contains } F1 \\
\mathbf{M} &= \mathbf{I} - \mathbf{X}_0(\mathbf{X}_0'\mathbf{X}_0)^-\mathbf{X}_0'
\end{aligned}$$

For the model

```
class A B;
model Y = A B A*B;
```

the Type II SS correspond to

$$R(A \mid \mu, B), \quad R(B \mid \mu, A), \quad R(A * B \mid \mu, A, B)$$

for effects A, B, and $A * B$, respectively. For the model

```
class A B C;
model Y =  A B(A) C(A B);
```

the Type II SS correspond to

$$R(A \mid \mu), \quad R(B(A) \mid \mu, A), \quad R(C(AB) \mid \mu, A, B(A))$$

for effects A, $B(A)$ and $C(A\,B)$, respectively. For the model

```
model Y = x x*x;
```

the Type II SS correspond to

$$R(X \mid \mu, X * X) \text{ and } R(X * X \mid \mu, X)$$

for x and $x * x$, respectively.

Note that, as in the situation for Type I tests, PROC MIXED and PROC GLIMMIX compute Type I hypotheses by sweeping $\mathbf{X}'\mathbf{X}$, but their test statistics are not necessarily equivalent to the results of sequentially fitting with those procedures models that contain successively more effects; while PROC TRANSREG computes tests labeled as being Type II by leaving out each effect in turn, but the specific linear hypotheses associated with these tests might not be precisely the same as the ones derived from successively sweeping $\mathbf{X}'\mathbf{X}$.

Example of Type II Estimable Functions

For a 2×2 factorial with w observations per cell, the general form of estimable functions is shown in Table 15.5. Any nonzero values for $L2$, $L4$, and $L6$ can be used to construct $\mathbf{L}$ vectors for computing the Type II SS for A, B, and $A * B$, respectively.

Table 15.5 General Form of Estimable Functions for 2×2 Factorial

Effect	Coefficient
μ	$L1$
$A1$	$L2$
$A2$	$L1 - L2$
$B1$	$L4$
$B2$	$L1 - L4$
$AB11$	$L6$
$AB12$	$L2 - L6$
$AB21$	$L4 - L6$
$AB22$	$L1 - L2 - L4 + L6$

For a balanced 2×2 factorial with the same number of observations in every cell, the Type II estimable functions are shown in Table 15.6.

Table 15.6 Type II Estimable Functions for Balanced 2×2 Factorial

	Coefficients for Effect		
Effect	A	B	$A * B$
μ	0	0	0
$A1$	$L2$	0	0
$A2$	$-L2$	0	0
$B1$	0	$L4$	0
$B2$	0	$-L4$	0
$AB11$	$0.5 * L2$	$0.5 * L4$	$L6$
$AB12$	$0.5 * L2$	$-0.5 * L4$	$-L6$
$AB21$	$-0.5 * L2$	$0.5 * L4$	$-L6$
$AB22$	$-0.5 * L2$	$-0.5 * L4$	$L6$

Now consider an unbalanced 2×2 factorial with two observations in every cell except the $AB22$ cell, which contains only one observation. The general form of estimable functions is the same as if it were balanced, since the same effects are still estimable. However, the Type II estimable functions for A and B are not the same as they were for the balanced design. The Type II estimable functions for this unbalanced 2×2 factorial are shown in Table 15.7.

Table 15.7 Type II Estimable Functions for Unbalanced 2×2 Factorial

	Coefficients for Effect		
Effect	A	B	$A * B$
μ	0	0	0
$A1$	$L2$	0	0
$A2$	$-L2$	0	0
$B1$	0	$L4$	0
$B2$	0	$-L4$	0
$AB11$	$0.6 * L2$	$0.6 * L4$	$L6$
$AB12$	$0.4 * L2$	$-0.6 * L4$	$-L6$
$AB21$	$-0.6 * L2$	$0.4 * L4$	$-L6$
$AB22$	$-0.4 * L2$	$-0.4 * L4$	$L6$

By comparing the hypothesis being tested in the balanced case to the hypothesis being tested in the unbalanced case for effects A and B, you can note that the Type II hypotheses for A and B are dependent on the cell frequencies in the design. For unbalanced designs in which the cell frequencies are not proportional to the background population, the Type II hypotheses for effects that are contained in other effects are of questionable value.

However, if an effect is not contained in any other effect, the Type II hypothesis for that effect is an MRH that does not involve any parameters except those associated with the effect in question.

Thus, Type II SS are appropriate for the following models:

- any balanced model
- any main-effects model
- any pure regression model
- an effect not contained in any other effect (regardless of the model)

In addition to the preceding models, Type II SS are generally accepted by most statisticians for purely nested models.

Type III and IV SS and Estimable Functions

When an effect is contained in another effect, the Type II hypotheses for that effect are dependent on the cell frequencies. The philosophy behind both the Type III and Type IV hypotheses is that the hypotheses tested for any given effect should be the same for all designs with the same general form of estimable functions.

To demonstrate this concept, recall the hypotheses being tested by the Type II SS in the balanced 2×2 factorial shown in Table 15.6. Those hypotheses are precisely the ones that the Type III

and Type IV hypotheses employ for all 2×2 factorials that have at least one observation per cell. The Type III and Type IV hypotheses for a design without missing cells usually differ from the hypothesis employed for the same design with missing cells since the general form of estimable functions usually differs.

Many SAS/STAT procedures can perform tests of Type III hypotheses, but only PROC GLM offers Type IV tests as well.

Type III Estimable Functions

Type III hypotheses are constructed by working directly with the general form of estimable functions. The following steps are used to construct a hypothesis for an effect $F1$:

1. For every effect in the model except $F1$ and those effects that contain $F1$, equate the coefficients in the general form of estimable functions to zero.

   If $F1$ is not contained in any other effect, this step defines the Type III hypothesis (as well as the Type II and Type IV hypotheses). If $F1$ is contained in other effects, go on to step 2. (See the section "Type II SS and Estimable Functions" on page 279 for a definition of when effect $F1$ is contained in another effect.)

2. If necessary, equate new symbols to compound expressions in the $F1$ block in order to obtain the simplest form for the $F1$ coefficients.

3. Equate all symbolic coefficients outside the $F1$ block to a linear function of the symbols in the $F1$ block in order to make the $F1$ hypothesis orthogonal to hypotheses associated with effects that contain $F1$.

By once again observing the Type II hypotheses being tested in the balanced 2×2 factorial, it is possible to verify that the A and $A * B$ hypotheses are orthogonal and also that the B and $A * B$ hypotheses are orthogonal. This principle of orthogonality between an effect and any effect that contains it holds for all balanced designs. Thus, construction of Type III hypotheses for any design is a logical extension of a process that is used for balanced designs.

The Type III hypotheses are precisely the hypotheses being tested by programs that reparameterize using the usual assumptions (for example, constraining all parameters for an effect to sum to zero). When no missing cells exist in a factorial model, Type III SS coincide with Yates' weighted squares-of-means technique. When cells are missing in factorial models, the Type III SS coincide with those discussed in Harvey (1960) and Henderson (1953).

The following discussion illustrates the construction of Type III estimable functions for a 2×2 factorial with no missing cells.

To obtain the $A * B$ interaction hypothesis, start with the general form and equate the coefficients for effects μ, A, and B to zero, as shown in Table 15.8.

Table 15.8 Type III Hypothesis for $A * B$ Interaction

Effect	**General Form**	$L1 = L2 = L4 = 0$
μ	$L1$	0
$A1$	$L2$	0
$A2$	$L1 - L2$	0
$B1$	$L4$	0
$B2$	$L1 - L4$	0
$AB11$	$L6$	$L6$
$AB12$	$L2 - L6$	$-L6$
$AB21$	$L4 - L6$	$-L6$
$AB22$	$L1 - L2 - L4 + L6$	$L6$

The last column in Table 15.8 represents the form of the MRH for $A * B$.

To obtain the Type III hypothesis for A, first start with the general form and equate the coefficients for effects μ and B to zero (let $L1 = L4 = 0$). Next let $L6 = K \times L2$, and find the value of K that makes the A hypothesis orthogonal to the A*B hypothesis. In this case, $K = 0.5$. Each of these steps is shown in Table 15.9.

In Table 15.9, the fourth column (under $L6 = K \times L2$) represents the form of all estimable functions not involving μ, $B1$, or $B2$. The prime difference between the Type II and Type III hypotheses for A is the way K is determined. Type II chooses K as a function of the cell frequencies, whereas Type III chooses K such that the estimable functions for A are orthogonal to the estimable functions for $A * B$.

Table 15.9 Type III Hypothesis for A

Effect	**General Form**	$L1 = L4 = 0$	$L6 = K \times L2$	$K = 0.5$
μ	$L1$	0	0	0
$A1$	$L2$	$L2$	$L2$	$L2$
$A2$	$L1 - L2$	$-L2$	$-L2$	$-L2$
$B1$	$L4$	0	0	0
$B2$	$L1 - L4$	0	0	0
$AB11$	$L6$	$L6$	$K * L2$	$0.5 * L2$
$AB12$	$L2 - L6$	$L2 - L6$	$(1 - K) * L2$	$0.5 * L2$
$AB21$	$L4 - L6$	$-L6$	$-K * L2$	$-0.5 * L2$
$AB22$	$L1 - L2 - L4 + L6$	$-L2 + L6$	$-(1 - K) * L2$	$-0.5 * L2$

An example of Type III estimable functions in a 3×3 factorial with unequal cell frequencies and missing diagonals is given in Table 15.10 (N_1 through N_6 represent the nonzero cell frequencies).

Table 15.10 3×3 Factorial Design with Unequal Cell Frequencies and Missing Diagonals

		B		
		1	2	3
A	1		N_1	N_2
	2	N_3		N_4
	3	N_5	N_6	

For any nonzero values of N_1 through N_6, the Type III estimable functions for each effect are shown in Table 15.11.

Table 15.11 Type III Estimable Functions for 3×3 Factorial Design with Unequal Cell Frequencies and Missing Diagonals

Effect	A	B	$A * B$
μ	0	0	0
$A1$	$L2$	0	0
$A2$	$L3$	0	0
$A3$	$-L2 - L3$	0	0
$B1$	0	$L5$	0
$B2$	0	$L6$	0
$B3$	0	$-L5 - L6$	0
$AB12$	$0.667 * L2 + 0.333 * L3$	$0.333 * L5 + 0.667 * L6$	$L8$
$AB13$	$0.333 * L2 - 0.333 * L3$	$-0.333 * L5 - 0.667 * L6$	$-L8$
$AB21$	$0.333 * L2 + 0.667 * L3$	$0.667 * L5 + 0.333 * L6$	$-L8$
$AB23$	$-0.333 * L2 + 0.333 * L3$	$-0.667 * L5 - 0.333 * L6$	$L8$
$AB31$	$-0.333 * L2 - 0.667 * L3$	$0.333 * L5 - 0.333 * L6$	$L8$
$AB32$	$-0.667 * L2 - 0.333 * L3$	$-0.333 * L5 + 0.333 * L6$	$-L8$

Type IV Estimable Functions

By once again looking at the Type II hypotheses being tested in the balanced 2×2 factorial (see Table 15.6), you can see another characteristic of the hypotheses employed for balanced designs: the coefficients of lower-order effects are averaged across each higher-level effect involving the same subscripts. For example, in the A hypothesis, the coefficients of $AB11$ and $AB12$ are equal to one-half the coefficient of $A1$, and the coefficients of $AB21$ and $AB22$ are equal to one-half the coefficient of $A2$. With this in mind, the basic concept used to construct Type IV hypotheses is that the coefficients of any effect, say $F1$, are distributed equitably across higher-level effects that contain $F1$. When missing cells occur, this same general philosophy is adhered to, but care must be taken in the way the distributive concept is applied.

Construction of Type IV hypotheses begins as does the construction of the Type III hypotheses. That is, for an effect $F1$, equate to zero all coefficients in the general form that do not belong to $F1$ or to any other effect containing $F1$. If $F1$ is not contained in any other effect, then the Type IV hypothesis (and Type II and III) has been found. If $F1$ is contained in other effects, then simplify,

if necessary, the coefficients associated with $F1$ so that they are all free coefficients or functions of other free coefficients in the $F1$ block.

To illustrate the method of resolving the free coefficients outside the $F1$ block, suppose that you are interested in the estimable functions for an effect A and that A is contained in AB, AC, and ABC. (In other words, the main effects in the model are A, B, and C.)

With missing cells, the coefficients of intermediate effects (here they are AB and AC) do not always have an equal distribution of the lower-order coefficients, so the coefficients of the highest-order effects are determined first (here it is ABC). Once the highest-order coefficients are determined, the coefficients of intermediate effects are automatically determined.

The following process is performed for each free coefficient of A in turn. The resulting symbolic vectors are then added together to give the Type IV estimable functions for A.

1. Select a free coefficient of A, and set all other free coefficients of A to zero.
2. If any of the levels of A have zero as a coefficient, equate all of the coefficients of higher-level effects involving that level of A to zero. This step alone usually resolves most of the free coefficients remaining.
3. Check to see if any higher-level coefficients are now zero when the coefficient of the associated level of A is not zero. If this situation occurs, the Type IV estimable functions for A are not unique.
4. For each level of A in turn, if the A coefficient for that level is nonzero, count the number of times that level occurs in the higher-level effect. Then equate each of the higher-level coefficients to the coefficient of that level of A divided by the count.

An example of a 3×3 factorial with four missing cells (N_1 through N_5 represent positive cell frequencies) is shown in Table 15.12.

Table 15.12 3×3 Factorial Design with Four Missing Cells

		B		
		1	2	3
A	1	N_1	N_2	
	2	N_3	N_4	
	3			N_5

The Type IV estimable functions are shown in Table 15.13.

Table 15.13 Type IV Estimable Functions for 3×3 Factorial Design with Four Missing Cells

Effect	A	B	$A*B$
μ	0	0	0
$A1$	$-L3$	0	0
$A2$	$L3$	0	0
$A3$	0	0	0
$B1$	0	$L5$	0
$B2$	0	$-L5$	0
$B3$	0	0	0
$AB11$	$-0.5*L3$	$0.5*L5$	$L8$
$AB12$	$-0.5*L3$	$-0.5*L5$	$-L8$
$AB21$	$0.5*L3$	$0.5*L5$	$-L8$
$AB22$	$0.5*L3$	$-0.5*L5$	$L8$
$AB33$	0	0	0

A Comparison of Type III and Type IV Hypotheses

For the vast majority of designs, Type III and Type IV hypotheses for a given effect are the same. Specifically, they are the same for any effect $F1$ that is not contained in other effects for any design (with or without missing cells). For factorial designs with no missing cells, the Type III and Type IV hypotheses coincide for all effects. When there are missing cells, the hypotheses can differ. By using the GLM procedure, you can study the differences in the hypotheses and then decide on the appropriateness of the hypotheses for a particular model.

The Type III hypotheses for three-factor and higher completely nested designs with unequal Ns in the lowest level differ from the Type II hypotheses; however, the Type IV hypotheses do correspond to the Type II hypotheses in this case.

When missing cells occur in a design, the Type IV hypotheses might not be unique. If this occurs in PROC GLM, you are notified, and you might need to consider defining your own specific comparisons.

References

Freund, R. J., Littell, R. C., and Spector, P. C. (1991), *SAS System for Linear Models*, Cary, NC: SAS Institute Inc.

Goodnight, J. H. (1978), *Tests of the Hypotheses in Fixed-Effects Linear Models*, Technical Report R-101, SAS Institute Inc, Cary, NC.

Goodnight, J. H. (1979), "A Tutorial on the Sweep Operator," *The American Statistician*, 33, 149–158.

Harvey, W. R. (1960), *Least-Squares Analysis of Data with Unequal Subclass Frequencies*, Technical Report ARS 20-8, USDA, Agriculture Research Service.

Henderson, C. R. (1953), "Estimation of Variance and Covariance Components," *Biometrics*, 9, 226–252.

Milliken, G. A. and Johnson, D. E. (1984), *Analysis of Messy Data, Volume I: Designed Experiments*, Belmont, CA: Lifetime Learning Publications.

Chapter 16

Introduction to Nonparametric Analysis

Contents

Overview: Nonparametric Analysis **291**
Testing for Normality 292
Comparing Distributions 292
One-Sample Tests **292**
Two-Sample Tests **293**
Comparing Two Independent Samples 293
Comparing Two Related Samples 294
Tests for *k* Samples **295**
Comparing *k* Independent Samples 295
Comparing *k* Dependent Samples 296
Measures of Correlation and Associated Tests **296**
Obtaining Ranks **297**
Kernel Density Estimation **297**
References **298**

Overview: Nonparametric Analysis

In statistical inference, or hypothesis testing, the traditional tests are called *parametric tests* because they depend on the specification of a probability distribution (such as the normal) except for a set of free parameters. Parametric tests are said to depend on distributional assumptions. *Nonparametric tests*, on the other hand, do not require any strict distributional assumptions. Even if the data are distributed normally, nonparametric methods are often almost as powerful as parametric methods.

Many nonparametric methods analyze the ranks of a variable rather than the original values. Procedures such as PROC NPAR1WAY calculate the ranks for you and then perform appropriate nonparametric tests. However, there are some situations in which you use a procedure such as PROC RANK to calculate ranks and then use another procedure to perform the appropriate test. See the section "Obtaining Ranks" on page 297 for details.

Although the NPAR1WAY procedure is specifically targeted for nonparametric analysis, many other procedures also perform nonparametric analyses. Some general references on nonparametrics include Hollander and Wolfe (1999), Conover (1999), Gibbons and Chakraborti (1992), Hettmansperger (1984), Randles and Wolfe (1979), and Lehmann (1975).

Testing for Normality

Many parametric tests assume an underlying normal distribution for the population. If your data do not meet this assumption, you might prefer to use a nonparametric analysis.

Base SAS software provides several tests for normality in the UNIVARIATE procedure. Depending on your sample size, PROC UNIVARIATE performs the Kolmogorov-Smirnov, Shapiro-Wilk, Anderson-Darling, and Cramér-von Mises tests. For more information, see the chapter "The UNIVARIATE Procedure" in the *Base SAS Procedures Guide*.

Comparing Distributions

To test the hypothesis that two or more groups of observations have identical distributions, use the NPAR1WAY procedure, which provides empirical distribution function (EDF) statistics. The procedure calculates the Kolmogorov-Smirnov test, the Cramér-von Mises test, and, when the data are classified into only two samples, the Kuiper test. Exact *p*-values are available for the two-sample Kolmogorov-Smirnov test. To obtain these tests, use the EDF option in the PROC NPAR1WAY statement. See Chapter 62, "The NPAR1WAY Procedure," for details.

One-Sample Tests

Base SAS software provides two one-sample tests in the UNIVARIATE procedure: a sign test and the Wilcoxon signed rank test. Both tests are designed for situations where you want to make an inference about the location (median) of a population. For example, suppose you want to test whether the median resting pulse rate of marathon runners differs from a specified value.

By default, both of these tests examine the hypothesis that the median of the population from which the sample is drawn is equal to a specified value, which is zero by default. The Wilcoxon signed rank test requires that the distribution be symmetric; the sign test does not require this assumption. These tests can also be used for the case of two related samples; see the section "Comparing Two Independent Samples" on page 293 for more information.

These two tests are automatically provided by the UNIVARIATE procedure. For details, formulas, and examples, see the chapter "The UNIVARIATE Procedure" in the *Base SAS Procedures Guide*.

Two-Sample Tests

This section describes tests appropriate for two independent samples (for example, two groups of subjects given different treatments) and for two related samples (for example, before-and-after measurements on a single group of subjects). Related samples are also referred to as paired samples or matched pairs.

Comparing Two Independent Samples

SAS/STAT software provides several nonparametric tests for location and scale differences for two independent samples.

When you perform these tests, your data should consist of a random sample of observations from two different populations. Your goal is to compare either the location parameters (medians) or the scale parameters of the two populations. For example, suppose your data consist of the number of days in the hospital for two groups of patients: those who received a standard surgical procedure and those who received a new, experimental surgical procedure. These patients are a random sample from the population of patients who have received the two types of surgery. Your goal is to decide whether the median hospital stays differ for the two populations.

Tests in the NPAR1WAY Procedure

The NPAR1WAY procedure provides the following location tests: Wilcoxon rank sum test (Mann-Whitney U test), median test, Savage test, and Van der Waerden (normal scores) test. Note that the Wilcoxon rank sum test can also be obtained from the FREQ procedure. PROC NPAR1WAY provides Hodges-Lehmann estimation of the location shift between two samples, including asymptotic (Moses) and exact confidence limits.

In addition, PROC NPAR1WAY produces the following tests for scale differences: Siegel-Tukey test, Ansari-Bradley test, Klotz test, and Mood test. PROC NPAR1WAY also provides the Conover test, which can be used to test for differences in both location and scale.

Additionally, PROC NPAR1WAY provides tests that use the input data observations as scores, enabling you to produce a wide variety of tests. You can construct any scores for your data with the DATA step, and then PROC NPAR1WAY computes the corresponding linear rank test. You can directly analyze the raw data this way, producing the permutation test known as Pitman's test.

When data are sparse, skewed, or heavily tied, the usual asymptotic tests might not be appropriate. In these situations, exact tests might be suitable for analyzing your data. The NPAR1WAY procedure can produce exact *p*-values for all of the two-sample tests for location and scale differences.

See Chapter 62, "The NPAR1WAY Procedure," for details, formulas, and examples of these tests.

Tests in the FREQ Procedure

The FREQ procedure provides nonparametric tests that compare the location of two groups and that test for independence between two variables.

The situation in which you want to compare the location of two groups of observations corresponds to a table with two rows. In this case, the asymptotic Wilcoxon rank sum test can be obtained by using SCORES=RANK in the TABLES statement and by looking at either of the following:

- the Mantel-Haenszel statistic in the list of tests for no association. This is labeled as "Mantel Haenszel Chi-Square," and PROC FREQ displays the statistic, the degrees of freedom, and the p-value. To obtain this statistic, specify the CHISQ option in the TABLES statement.
- the CMH statistic 2 in the section on Cochran-Mantel-Haenszel statistics. PROC FREQ displays the statistic, the degrees of freedom, and the p-value. To obtain this statistic, specify the CMH2 option in the TABLES statement.

When you test for independence, the question being answered is whether the two variables of interest are related in some way. For example, you might want to know if student scores on a standard test are related to whether students attended a public or private school. One way to think of this situation is to consider the data as a two-way table; the hypothesis of interest is whether the rows and columns are independent. In the preceding example, the groups of students would form the two rows, and the scores would form the columns. The special case of a two-category response (Pass/Fail) leads to a 2×2 table; the case of more than two categories for the response (A/B/C/D/F) leads to a $2 \times c$ table, where c is the number of response categories.

For testing whether two variables are independent, PROC FREQ provides Fisher's exact test. For a 2×2 table, PROC FREQ automatically provides Fisher's exact test when you specify the CHISQ option in the TABLES statement. For a $2 \times c$ table, use the FISHER option in the EXACT statement to obtain the test.

See Chapter 35, "The FREQ Procedure," for details, formulas, and examples of these tests.

Comparing Two Related Samples

SAS/STAT software provides the following nonparametric tests for comparing the locations of two related samples:

- Wilcoxon signed rank test
- sign test
- McNemar's test

The first two tests are available in the UNIVARIATE procedure, and the last test is available in the FREQ procedure. When you perform these tests, your data should consist of pairs of measurements for a random sample from a single population. For example, suppose your data consist of SAT

scores for students before and after attending a course on how to prepare for the SAT. The pairs of measurements are the scores before and after the course, and the students should be a random sample of students who attended the course. Your goal in analysis is to decide whether the median change in scores is significantly different from zero.

Tests in the UNIVARIATE Procedure

By default, PROC UNIVARIATE performs a Wilcoxon signed rank test and a sign test. To use these tests on two related samples, perform the following steps:

1. In the DATA step, create a new variable that contains the differences between the two related variables.
2. Run PROC UNIVARIATE, using the new variable in the VAR statement.

See the chapter "The UNIVARIATE Procedure" in the *Base SAS Procedures Guide* for details and examples of these tests.

Tests in the FREQ Procedure

The FREQ procedure can be used to obtain McNemar's test, which is simply another special case of a Cochran-Mantel-Haenszel statistic (and also of the sign test). The AGREE option in the TABLES statement produces this test for 2×2 tables, and exact p-values are also available for this test. See Chapter 35, "The FREQ Procedure," for more information.

Tests for *k* Samples

Comparing *k* Independent Samples

One goal in comparing k independent samples is to determine whether the location parameters (medians) of the populations are different. Another goal is to determine whether the scale parameters for the populations are different. For example, suppose new employees are randomly assigned to one of three training programs. At the end of the program, the employees are given a standard test that provides a rating score of their job ability. The goal of analysis is to compare the median scores for the three groups and decide whether the differences are real or due to chance alone.

To compare k independent samples, either the NPAR1WAY or the FREQ procedure provides a Kruskal-Wallis test. PROC NPAR1WAY also provides the Savage, median, and Van der Waerden (normal scores) tests. In addition, PROC NPAR1WAY produces the following tests for scale differences: Siegel-Tukey test, Ansari-Bradley test, Klotz test, and Mood test. PROC NPAR1WAY also

provides the Conover test, which can be used to test for differences in both location and scale. Note that you can obtain exact *p*-values for all of these tests.

Additionally, you can specify the SCORES=DATA option to use the input data observations as scores. This enables you to produce a very wide variety of tests. You can construct any scores for your data with the DATA step, and then PROC NPAR1WAY computes the corresponding linear rank and one-way ANOVA tests. You can also analyze the raw data with the SCORES=DATA option; for two-sample data, this permutation test is known as Pitman's test.

See Chapter 62, "The NPAR1WAY Procedure," for details, formulas, and examples.

To produce a Kruskal-Wallis test in the FREQ procedure, use SCORES=RANK and the CMH2 option in the TABLES statement. Then, look at the second Cochran-Mantel-Haenszel statistic (labeled "Row Mean Scores Differ") to obtain the Kruskal-Wallis test. The FREQ procedure also provides the Jonckheere-Terpstra test, which is more powerful than the Kruskal-Wallis test for comparing k samples against ordered alternatives. The exact test is also available. In addition, you can obtain a ridit analysis, developed by Bross (1958), by specifying SCORES=RIDIT or SCORES=MODRIDIT in the TABLES statement in the FREQ procedure. See Chapter 35, "The FREQ Procedure," for more information.

Comparing *k* Dependent Samples

Friedman's test enables you to compare the locations of three or more dependent samples. You can obtain Friedman's chi-square with the FREQ procedure by using the CMH2 option and SCORES=RANK and by looking at the second CMH statistic in the output. For an example, see Chapter 35, "The FREQ Procedure." This chapter also contains formulas and other details about the CMH statistics. For a discussion of how to use the RANK and GLM procedures to obtain Friedman's test, see Ipe (1987).

Measures of Correlation and Associated Tests

The CORR procedure in Base SAS software provides several nonparametric measures of association and associated tests. It computes Spearman's rank-order correlation, Kendall's tau-*b*, and Hoeffding's measure of dependence, and it provides tests for each of these statistics. PROC CORR also computes Spearman's partial rank-order correlation and Kendall's partial tau-*b*. Finally, PROC CORR computes Cronbach's coefficient alpha for raw and standardized variables. This statistic can be used to estimate the reliability coefficient. For a general discussion of correlations, formulas, interpretation, and examples, see the chapter "The CORR Procedure" in the *Base SAS Procedures Guide*.

The FREQ procedure also provides some nonparametric measures of association: gamma, Kendall's tau-b, Stuart's tau-c, Somer's D, and the Spearman rank correlation. The output includes the measure, the asymptotic standard error, confidence limits, and the asymptotic test that the measure equals zero. For the Spearman rank correlation, you can optionally request an exact p-value for the test that the correlation equals zero. For more information, see Chapter 35, "The FREQ Procedure."

Obtaining Ranks

The primary procedure for obtaining ranks is the RANK procedure in Base SAS software. Note that the PRINQUAL and TRANSREG procedures also provide rank transformations. With all three of these procedures, you can create an output data set and use it as input to another SAS/STAT procedure or to the IML procedure. For more information, see the chapter "The RANK Procedure" in the *Base SAS Procedures Guide*. Also see Chapter 70, "The PRINQUAL Procedure," and Chapter 90, "The TRANSREG Procedure."

In addition, you can specify SCORES=RANK in the TABLES statement in the FREQ procedure. PROC FREQ then uses ranks to perform the analyses requested and generates nonparametric analyses.

For more discussion of the rank transform, see Iman and Conover (1979); Conover and Iman (1981); Hora and Conover (1984); Iman, Hora, and Conover (1984); Hora and Iman (1988); and Iman (1988).

Kernel Density Estimation

The KDE procedure performs either univariate or bivariate kernel density estimation. Statistical *density estimation* involves approximating a hypothesized probability density function from observed data. *Kernel density estimation* is a nonparametric technique for density estimation in which a known density function (the kernel) is averaged across the observed data points to create a smooth approximation.

PROC KDE uses a Gaussian density as the kernel, and its assumed variance determines the smoothness of the resulting estimate. PROC KDE outputs the kernel density estimate to a SAS data set, which you can then use with other procedures for plotting or analysis. PROC KDE also computes a variety of common statistics, including estimates of the percentiles of the hypothesized probability density function.

For more information, see Chapter 45, "The KDE Procedure."

References

Agresti, A. (2007), *An Introduction to Categorical Data Analysis*, Second Edition, New York: John Wiley & Sons.

Bross, I. D. J. (1958), "How to Use Ridit Analysis," *Biometrics*, 14, 18–38.

Conover, W. J. (1999), *Practical Nonparametric Statistics*, Third Edition, New York: John Wiley & Sons.

Conover, W. J. and Iman, R. L. (1981), "Rank Transformations as a Bridge between Parametric and Nonparametric Statistics," *The American Statistician*, 35, 124–129.

Gibbons, J. D. and Chakraborti, S. (1992), *Nonparametric Statistical Inference*, Third Edition, New York: Marcel Dekker.

Hajek, J. (1969), *A Course in Nonparametric Statistics*, San Francisco: Holden-Day.

Hettmansperger, T. P. (1984), *Statistical Inference Based on Ranks*, New York: John Wiley & Sons.

Hollander, M. and Wolfe, D. A. (1999), *Nonparametric Statistical Methods*, Second Edition, New York: John Wiley & Sons.

Hora, S. C. and Conover, W. J. (1984), "The F Statistic in the Two-Way Layout with Rank-Score Transformed Data," *Journal of the American Statistical Association*, 79, 668–673.

Hora, S. C. and Iman, R. L. (1988), "Asymptotic Relative Efficiencies of the Rank-Transformation Procedure in Randomized Complete Block Designs," *Journal of the American Statistical Association*, 83, 462–470.

Iman, R. L. (1988), "The Analysis of Complete Blocks Using Methods Based on Ranks," *Proceedings of the Thirteenth Annual SAS Users Group International Conference*, 970–978.

Iman, R. L. and Conover, W. J. (1979), "The Use of the Rank Transform in Regression," *Technometrics*, 21, 499–509.

Iman, R. L., Hora, S. C., and Conover, W. J. (1984), "Comparison of Asymptotically Distribution-Free Procedures for the Analysis of Complete Blocks," *Journal of the American Statistical Association*, 79, 674–685.

Ipe, D. (1987), "Performing the Friedman Test and the Associated Multiple Comparison Test Using PROC GLM," *Proceedings of the Twelfth Annual SAS Users Group International Conference*, 1146–1148.

Lehmann, E. L. (1975), *Nonparametrics: Statistical Methods Based on Ranks*, San Francisco: Holden-Day.

Randles, R. H. and Wolfe, D. A. (1979), *Introduction to the Theory of Nonparametric Statistics*, New York: John Wiley & Sons.

Chapter 17

Introduction to Structural Equation Modeling with Latent Variables

Contents

Overview: Structural Equation Modeling with Latent Variables 299
TCALIS and CALIS Procedures 300
Comparison of the TCALIS and SYSLIN Procedures 300
Model Specification 301
Estimation Methods 302
Statistical Inference 303
Goodness-of-Fit Statistics 304
Optimization Methods 305
Structural Equation Models and the LINEQS Modeling Language 305
Identification of Models 311
Path Diagrams and the PATH Modeling Language 316
Some Measurement Models 320
A Combined Measurement-Structural Model with Reciprocal Influence and Correlated Residuals 334
References 363

Overview: Structural Equation Modeling with Latent Variables

You can use the CALIS or TCALIS procedure for analysis of covariance structures, fitting systems of linear structural equations, and path analysis. These terms are more or less interchangeable, but they emphasize different aspects of the analysis. The analysis of covariance structures refers to the formulation of a model for the variances and covariances among a set of variables and the fitting of the model to an observed covariance matrix. In linear structural equations, the model is formulated as a system of equations relating several random variables with assumptions about the variances and covariances of the random variables. In path analysis, the model is formulated as a path diagram, in which arrows connecting variables represent variances, covariances, and regression (or path) coefficients. Path models and linear structural equation models can be converted to models of the covariance matrix and can, therefore, be fitted by the methods of covariance structure analysis. All

of these methods support the use of hypothetical latent variables and measurement errors in the models.

Loehlin (1987) provides an excellent introduction to latent variable models by using path diagrams and structural equations. A more advanced treatment of structural equation models with latent variables is given by Bollen (1989). Fuller (1987) provides a highly technical statistical treatment of measurement-error models.

TCALIS and CALIS Procedures

In SAS 9.2, the newly developed TCALIS procedure serves as an enhanced version of the older CALIS procedure. The TCALIS procedure has most of the capabilities of the CALIS procedure. In addition, it has some new capabilities that are not available in PROC CALIS. These new capabilities include multiple-group analysis, mean structures analysis, and a whole spectrum of modeling languages, among others. See Chapter 88, "The TCALIS Procedure (Experimental)," for more details about the new features available in PROC TCALIS. In this chapter, statistical analyses in the examples are done using PROC TCALIS.

Comparison of the TCALIS and SYSLIN Procedures

The SYSLIN procedure in SAS/ETS software can also fit certain kinds of path models and linear structural equation models. PROC TCALIS differs from PROC SYSLIN in that PROC TCALIS is more general in the use of latent variables in the models. Latent variables are unobserved, hypothetical variables, as distinct from manifest variables, which are the observed data. PROC SYSLIN allows at most one latent variable, the error term, in each equation. PROC TCALIS allows several latent variables to appear in an equation—in fact, all the variables in an equation can be latent as long as there are other equations that relate the latent variables to manifest variables.

Both the TCALIS and SYSLIN procedures enable you to specify a model as a system of linear equations. When there are several equations, a given variable might be a dependent variable in one equation and an independent variable in other equations. Therefore, additional terminology is needed to describe unambiguously the roles of variables in the system. Variables with values that are determined jointly and simultaneously by the system of equations are called *endogenous variables*. Variables with values that are determined outside the system—that is, in a manner separate from the process described by the system of equations—are called exogenous variables. The purpose of the system of equations is to explain the variation of each endogenous variable in terms of exogenous variables or other endogenous variables or both. Refer to Loehlin (1987, p. 4) for further discussion of endogenous and exogenous variables. In the econometric literature, error and disturbance terms are usually distinguished from exogenous variables, but in systems with more than one latent variable in an equation, the distinction is not always clear.

In PROC SYSLIN, endogenous variables are identified by the ENDOGENOUS statement. In PROC TCALIS, endogenous variables are identified by the procedure automatically after you specify the model. With different modeling languages, the identification of endogenous variables by PROC TCALIS is done by different sets of rules. For example, when you specify structural equations by using the LINEQS modeling language in PROC TCALIS, endogenous variables are assumed to be those that appear on the left-hand sides of the equations; a given variable can appear on the left-hand side of at most one equation. When you specify your model by using the PATH modeling language in PROC TCALIS, endogenous variables are those variables pointed to by arrows at least once in the path specifications.

PROC SYSLIN provides many methods of estimation, some of which are applicable only in special cases. For example, ordinary least squares estimates are suitable in certain kinds of systems but might be statistically biased and inconsistent in other kinds. PROC TCALIS provides three major methods of estimation that can be used with most models. Both the TCALIS and SYSLIN procedures can do maximum likelihood estimation, which PROC TCALIS calls ML and PROC SYSLIN calls FIML. PROC SYSLIN can be much faster than PROC TCALIS in those special cases for which it provides computationally efficient estimation methods. However, PROC TCALIS has a variety of sophisticated algorithms for maximum likelihood estimation that might be much faster than FIML in PROC SYSLIN.

PROC TCALIS can impose a wider variety of constraints on the parameters, including nonlinear constraints, than can PROC SYSLIN. For example, PROC TCALIS can constrain error variances or covariances to equal specified constants, or it can constrain two error variances to have a specified ratio.

Model Specification

PROC TCALIS provides several modeling languages to specify a model. Different modeling languages in PROC TCALIS are signified by the main model specification statement used. In the TCALIS procedure, FACTOR, LINEQS, LISMOD, MSTRUCT, PATH, and RAM are the main modeling specification statements. Each of these statements invokes a specific modeling language. Depending on your modeling philosophy and the type of the model, you can choose a modeling language that is most suitable for your application. Models specified using structural equations can be transcribed directly into the LINEQS statement. Models that are hypothesized using path diagrams can be described easily in the PATH or RAM statement. First-order confirmatory or exploratory factor models are most conveniently specified using the FACTOR and MATRIX statements. Traditional LISREL models are supported through the LISMOD and MATRIX statements. Finally, patterned covariance and mean models can be specified directly by the MSTRUCT and MATRIX statements.

For most applications, the PATH and LINEQS statements are the easiest to use. In other cases, the FACTOR, LISMOD, MSTRUCT, or RAM statement might be more suitable. See the section "Which Modeling Language?" on page 6714 in Chapter 88, "The TCALIS Procedure (Experimental)," for a more detailed discussion.

You can save a model specification in an OUTMODEL= data set, which can then be used with the INMODEL= option to specify the model in a subsequent analysis.

Estimation Methods

The TCALIS procedure provides five methods of estimation specified by the METHOD= option:

DWLS	diagonally weighted least squares
ULS	unweighted least squares
GLS	normal theory generalized least squares
ML	maximum likelihood for multivariate normal distributions
WLS	weighted least squares for arbitrary distributions

Each estimation method is based on finding parameter estimates that minimize a badness-of-fit or discrepancy function, which measures the difference between the observed sample covariance matrix and the fitted (or predicted) covariance matrix, given the model and the parameter estimates. See the section "Estimation Criteria" on page 6888 in Chapter 88, "The TCALIS Procedure (Experimental)," for formulas, or refer to Loehlin (1987, pp. 54–62) and Bollen (1989, pp. 104–123) for further discussion.

The default is METHOD=ML, which is the most popular method for applications. The option METHOD=GLS usually produces very similar results to those produced by METHOD=ML. Asymptotically, ML and GLS are the same. Both methods assume a multivariate normal distribution in the population. The WLS method with default weight matrix, which is equivalent to the asymptotically distribution free (ADF) method, yields asymptotically normal estimates regardless of the distribution in the population. When the multivariate normal assumption is in doubt, especially if the variables have high kurtosis, you should seriously consider the WLS method. When a correlation matrix is analyzed, only the WLS can produce correct standard error estimates. However, in order to use the WLS method with the expected statistical properties, sample size must be large. Several thousands might be a minimum requirement.

The ULS and DWLS methods yield reasonable estimates under less restrictive assumptions. You can apply these methods to normal or nonnormal situations, or to covariance or correlation matrices. The drawback is that the statistical qualities of the estimates seem to be unknown. For this reason, PROC TCALIS does not provide standard errors or test statistics with these two methods.

You cannot use METHOD=ML or METHOD=GLS if the observed covariance matrix is singular. You could either remove variables involved in the linear dependencies or use less restrictive estimation methods like ULS. Specifying METHOD=ML assumes that the predicted covariance matrix is nonsingular. If ML fails because of a singular predicted covariance matrix, you need to examine whether the model specification leads to the singularity. If so, modify the model specification to eliminate the problem. If not, you probably need to use other estimation methods.

You should remove outliers and try to transform variables that are skewed or heavy-tailed. This applies to all estimation methods, since all the estimation methods depend on the sample covariance matrix, and the sample covariance matrix is a poor estimator for distributions with high kurtosis (Bollen 1989, pp. 415–418; Huber 1981; Hampel et al. 1986). PROC TCALIS displays estimates of univariate and multivariate kurtosis (Bollen 1989, pp. 418–425) if you specify the KURTOSIS option in the PROC TCALIS statement.

Statistical Inference

When you specify the ML, GLS, or WLS estimation with appropriate models, PROC TCALIS can compute the following:

- a chi-square goodness-of-fit test of the specified model versus the alternative that the data are from a population with unconstrained covariance matrix (Loehlin 1987, pp. 62–64; Bollen 1989, pp. 110, 115, 263–269)
- approximate standard errors of the parameter estimates (Bollen 1989, pp. 109, 114, 286), displayed with the STDERR option
- various modification indices, requested via the MODIFICATION or MOD option, that give the approximate change in the chi-square statistic that would result from removing constraints on the parameters or constraining additional parameters to zero (Bollen 1989, pp. 293–303)

If you have two models such that one model results from imposing constraints on the parameters of the other, you can test the constrained model against the more general model by fitting both models with PROC TCALIS. If the constrained model is correct, the difference between the chi-square goodness of fit statistics for the two models has an approximate chi-square distribution with degrees of freedom equal to the difference between the degrees of freedom for the two models (Loehlin 1987, pp. 62–67; Bollen 1989, pp. 291–292).

All of the test statistics and standard errors computed under ML and GLS depend on the assumption of multivariate normality. Normality is a much more important requirement for data with random independent variables than it is for fixed independent variables. If the independent variables are random, distributions with high kurtosis tend to give liberal tests and excessively small standard errors, while low kurtosis tends to produce the opposite effects (Bollen 1989, pp. 266–267, 415–432).

All test statistics and standard errors computed by PROC TCALIS are based on asymptotic theory and should not be trusted in small samples. There are no firm guidelines on how large a sample must be for the asymptotic theory to apply with reasonable accuracy. Some simulation studies have indicated that problems are likely to occur with sample sizes less than 100 (Loehlin 1987, pp. 60–61; Bollen 1989, pp. 267–268). Extrapolating from experience with multiple regression would suggest that the sample size should be at least 5 to 20 times the number of parameters to be estimated in order to get reliable and interpretable results. The WLS method might even require that the sample size be over several thousand.

The asymptotic theory requires that the parameter estimates be in the interior of the parameter space. If you do an analysis with inequality constraints and one or more constraints are active at the solution (for example, if you constrain a variance to be nonnegative and the estimate turns out to be zero), the chi-square test and standard errors might not provide good approximations to the actual sampling distributions.

For modeling correlation structures, the only theoretically correct method is the WLS method with the default ASYCOV=CORR option. For other methods, standard error estimates for modeling correlation structures might be inaccurate even for sample sizes as large as 400. The chi-square statistic

is generally the same regardless of which matrix is analyzed, provided that the model involves no scale-dependent constraints. However, if the purpose is to obtain reasonable parameter estimates for the correlation structures only, then you might find other estimation methods useful as well.

If you fit a model to a correlation matrix and the model constrains one or more elements of the predicted matrix to equal 1.0, the degrees of freedom of the chi-square statistic must be reduced by the number of such constraints. PROC TCALIS attempts to determine which diagonal elements of the predicted correlation matrix are constrained to a constant, but it might fail to detect such constraints in complicated models, particularly when programming statements are used. If this happens, you should add parameters to the model to release the constraints on the diagonal elements.

Goodness-of-Fit Statistics

In addition to the chi-square test, there are many other statistics for assessing the goodness of fit of the predicted correlation or covariance matrix to the observed matrix.

Akaike's (1987) information criterion (AIC) and Schwarz's (1978) Bayesian criterion (SBC) are useful for comparing models with different numbers of parameters—the model with the smallest value of AIC or SBC is considered best. Based on both theoretical considerations and various simulation studies, SBC seems to work better, since AIC tends to select models with too many parameters when the sample size is large.

There are many descriptive measures of goodness of fit that are scaled to range approximately from zero to one: the goodness-of-fit index (GFI) and GFI adjusted for degrees of freedom (AGFI) (Jöreskog and Sörbom 1988), centrality (McDonald 1989), and the parsimonious fit index (James, Mulaik, and Brett 1982). Bentler and Bonett (1980) and Bollen (1986) have proposed measures for comparing the goodness of fit of one model with another in a descriptive rather than inferential sense.

The root mean squared error approximation (RMSEA) proposed by Steiger and Lind (1980) does not assume a true model being fitted to the data. It measures the discrepancy between the fitted model and the covariance matrix in the population. For samples, RMSEA and confidence intervals can be estimated. Statistical tests for determining whether the population RMSEA's fall below certain specified values are available (Browne and Cudeck 1993). In the same vein, Browne and Cudeck (1993) propose the expected cross validation index (ECVI) that measures how good a model is for predicting future sample covariances. Point estimate and confidence intervals for ECVI are also developed.

None of these measures of goodness of fit are related to the goodness of prediction of the structural equations. Goodness of fit is assessed by comparing the observed correlation or covariance matrix with the matrix computed from the model and parameter estimates. Goodness of prediction is assessed by comparing the actual values of the endogenous variables with their predicted values, usually in terms of root mean squared error or proportion of variance accounted for (R^2). For latent endogenous variables, root mean squared error and R^2 can be estimated from the fitted model.

Optimization Methods

PROC TCALIS uses a variety of nonlinear optimization algorithms for computing parameter estimates. These algorithms are very complicated and do not always work. PROC TCALIS will generally inform you when the computations fail, usually by displaying an error message about the iteration limit being exceeded. When this happens, you might be able to correct the problem simply by increasing the iteration limit (MAXITER= and MAXFUNC=). However, it is often more effective to change the optimization method (OMETHOD=) or initial values. For more details, see the section "Use of Optimization Techniques" on page 6923 in Chapter 88, "The TCALIS Procedure (Experimental)," and refer to Bollen (1989, pp. 254–256).

PROC TCALIS might sometimes converge to a local optimum rather than the global optimum. To gain some protection against local optima, you can run the analysis several times with different initial estimates. The RANDOM= option in the PROC TCALIS statement is useful for generating a variety of initial estimates.

Structural Equation Models and the LINEQS Modeling Language

Consider fitting a linear equation to two observed variables, Y and X. Simple linear regression uses the model of a particular form, labeled for purposes of discussion, as model form A.

Model Form A

$$Y = \alpha + \beta X + E_Y$$

with the following assumption:

$$\mathrm{Cov}(X, E_Y) = 0$$

where α and β are coefficients to be estimated and E_Y is an error term. If the values of X are fixed, the values of E_Y are assumed to be independent and identically distributed realizations of a normally distributed random variable with mean zero and variance $\mathrm{Var}(E_Y)$. If X is a random variable, X and E_Y are assumed to have a bivariate normal distribution with zero correlation and variances $\mathrm{Var}(X)$ and $\mathrm{Var}(E_Y)$, respectively. Under either set of assumptions, the usual formulas hold for the estimates of the coefficients and their standard errors (see Chapter 4, "Introduction to Regression Procedures").

In the REG or SYSLIN procedure, you would fit a simple linear regression model with a MODEL statement listing only the names of the manifest variables, as shown in the following statements:

```
proc reg;
   model y = x;
run;
```

You can also fit this model with PROC TCALIS, but you must explicitly specify the error terms and the parameter name for the regression coefficient (except for the intercept, which is assumed to be present in each equation). The following specification in PROC TCALIS is equivalent to the preceding regression model:

```
proc tcalis;
   lineqs
      y = beta x + ey;
run;
```

where beta is the parameter name for the regression coefficient and ey is the error term of the equation. You do not need to type an "*" between beta and x to indicate the multiplication of the variable by the coefficient.

You might use other names for the parameters and the error terms, but there are rules to follow in the LINEQS model specification. The LINEQS statement uses the convention that the names of error terms begin with the letter E or e, disturbances (errors terms for latent variables) in equations begin with D or d, and other latent variables begin with F or f for "factor." Names of variables in the input SAS data set can, of course, begin with any letter.

Optionally, you can specify the variance parameters of exogenous variables explicitly by using the STD statement as follows:

```
proc tcalis;
   lineqs
      y  = beta x + ey;
   std
      x  = vx,
      ey = vey;
run;
```

where vx and vey represent the variance of x and ey, respectively. Explicitly giving these variance parameters names would be useful when you need to set constraints on these parameters by referring to their names. In this simple case without constraints, however, naming these variance parameters is not necessary.

Although naming variance parameters for exogenous variables is optional, naming the regression coefficients when they should be free parameters is not. Consider the following statements:

```
proc tcalis;
   lineqs
      y = x + ey;
run;
```

In this specification, you leave out the regression coefficient beta from the preceding LINEQS model. Instead of being a parameter to estimate, the regression coefficient in this specification is a fixed constant 1.

In certain special situations where you need to specify a regression equation without an error term, you can explicitly set the variance of the error term to zero. For example, the following statements, in effect, will fit an equation without an error term:

```
proc tcalis;
   lineqs
      y = beta x + ey;
   std
      ey = 0;
run;
```

In this specification, the mean of the error term ey is presumably zero, as all error terms in the PROC TCALIS are set to have fixed zero means. Together with the zero variance specification for ey in the STD statement, ey is essentially a zero constant in the equation.

By default, PROC TCALIS analyzes the covariance matrix. This is exactly the opposite of PROC CALIS or PROC FACTOR, which analyzes the correlation matrix by default. Most applications that use PROC TCALIS should employ the default covariance structure analysis.

Since the analysis of covariance structures is based on modeling the covariance matrix and the covariance matrix contains no information about means, PROC TCALIS neglects the intercept parameter by default. To estimate the intercept, you can add the intercept term explicitly into the LINEQS statement. For example, the following statements fit a regression model with the estimation of the intercept alpha:

```
proc tcalis;
   lineqs
      y = alpha intercept + beta x + ey;
run;
```

In the LINEQS statement, intercept represents a "variable" with a constant value of 1; hence, the coefficient alpha is the intercept parameter. Notice that with the simultaneous analysis of mean and covariance structures, you have to provide either the raw data or the means and covariances in your input data set.

Other commonly used options in the PROC TCALIS statement include the following:

- MODIFICATION to display model modification indices
- NOBS to specify the number of observations
- NOSE to suppress the display of approximate standard errors
- RESIDUAL to display residual correlations or covariances
- TOTEFF to display total and indirect effects

For ordinary unconstrained regression models, there is no reason to use PROC TCALIS instead of PROC REG. But suppose that the observed variables Y and X are contaminated by errors (especially measurement errors), and you want to estimate the linear relationship between their true, error-free scores. The model can be written in several forms. A model of form B is as follows.

Model Form B

$$\begin{aligned} Y &= \alpha + \beta F_X + E_Y \\ X &= F_X + E_X \end{aligned}$$

with the following assumption:

$$\mathrm{Cov}(F_X, E_X) = \mathrm{Cov}(F_X, E_Y) = \mathrm{Cov}(E_X, E_Y) = 0$$

This model has two error terms, E_Y and E_X, as well as another latent variable F_X representing the true value corresponding to the manifest variable X. The true value corresponding to Y does not appear explicitly in this form of the model.

The assumption in model form B that the error terms and the latent variable F_X are jointly uncorrelated is of critical importance. This assumption must be justified on substantive grounds such as the physical properties of the measurement process. If this assumption is violated, the estimators might be severely biased and inconsistent.

You can express model form B in the LINEQS statement as follows:

```
proc tcalis;
   lineqs
      y = beta fx + ey,
      x = fx + ex;
   std
      fx = vfx,
      ey = vey,
      ex = vex;
run;
```

In this specification, you specify a variance for each of the latent variables in this model by using the STD statement. You can specify either a name, in which case the variance is considered a parameter to be estimated, or a number, in which case the variance is constrained to equal that numeric value. In this model, vfx, vey, and vex are variance parameters to estimate.

The variances of endogenous variables are predicted from the model and hence are not parameters. Covariances involving latent exogenous variables are assumed to be zero by default.

Fuller (1987, pp. 18–19) analyzes a data set from Voss (1969) involving corn yields (Y) and available soil nitrogen (X) for which there is a prior estimate of the measurement error for soil nitrogen $\mathrm{Var}(E_X)$ of 57. You can fit model form B with this constraint to the data by using the following statements:

```
data corn(type=cov);
   input _type_ $ _name_ $ y x;
   datalines;
n    . 11       11
mean . 97.4545  70.6364
cov  y 87.6727  .
cov  x 104.8818 304.8545
;
```

```
proc tcalis data=corn;
   lineqs y = beta fx + ey,
          x = fx + ex;
   std ex = 57,
       fx = vfx,
       ey = vey;
run;
```

In the STD statement, the variance of ex is given as the constant value 57. PROC TCALIS produces the estimates shown in Figure 17.1.

Figure 17.1 Measurement Error Model for Corn Data

```
                          Linear Equations

                y       =    0.4232*fx   +  1.0000 ey
                Std Err      0.1658 beta
                t Value      2.5520
                x       =    1.0000 fx   +  1.0000 ex

             Estimates for Variances of Exogenous Variables

Variable                                                  Standard
Type          Variable    Parameter      Estimate          Error       t Value

Error         ex                          57.00000
Latent        fx          vfx           247.85450      136.33508       1.81798
Error         ey          vey            43.29105       23.92488       1.80946
```

PROC TCALIS also displays information about the initial estimates that can be useful if there are optimization problems. If there are no optimization problems, the initial estimates are usually not of interest; they are not reproduced in the examples in this chapter.

You can write an equivalent model (labeled here as model form C) by using a latent variable F_Y to represent the true value corresponding to Y.

Model Form C

$$\begin{aligned} Y &= F_Y + E_Y \\ X &= F_X + E_X \\ F_Y &= \alpha + \beta F_X \end{aligned}$$

with the following assumption:

$$\mathrm{Cov}(F_X, E_X) = \mathrm{Cov}(F_X, E_X) = \mathrm{Cov}(E_X, E_Y) = 0$$

The first two equations express the observed variables in terms of a true score plus error; these two equations are called the measurement model. The third equation, expressing the relationship

between the latent true-score variables, is called the structural or causal model. The decomposition of a model into a measurement model and a structural model (Keesling 1972; Wiley 1973; Jöreskog 1973) has been popularized by the program LISREL (Jöreskog and Sörbom 1988). The statements for fitting this model are shown in the following:

```
proc tcalis;
   lineqs
      y   = fy + ey,
      x   = fx + ex,
      fy  = beta fx + dfy;
   std
      fx  = vfx,
      ey  = vey,
      ex  = vex,
      dfy = 0;
run;
```

As a syntactic requirement, each equation in the LINEQS statement should have an error term. As discussed before, because dfy has a fixed variance 0 in the STD statement, in effect, there is no error term in the structural equation with the outcome variable fy.

You do not need to include the variance of F_Y in the STD statement because the variance of F_Y is determined by the structural model in terms of the variance of F_X—that is, $\mathrm{Var}(F_Y)=\beta^2\ \mathrm{Var}(F_X)$.

Correlations or covariances involving endogenous variables are derived from the model. For example, the structural equation in model form C implies that F_Y and F_X are correlated unless β is zero. In all of the models discussed so far, the latent exogenous variables are assumed to be jointly uncorrelated. For example, in model form C, E_Y, E_X, and F_X are assumed to be uncorrelated. If you want to specify a model in which E_Y and E_X, say, are correlated, you can use the COV statement to specify the numeric value of the covariance $\mathrm{Cov}(E_Y, E_X)$ between E_Y and E_X, or you can specify a name to make the covariance a parameter to be estimated. For example:

```
proc tcalis;
   lineqs
      y   = fy + ey,
      x   = fx + ex,
      fy  = beta fx + dfy;
   std
      fx  = vfx,
      ey  = vey,
      ex  = vex,
      dfy = 0;
   cov
      ey ex = ceyex;
run;
```

This COV statement specifies that the covariance between ey and ex is a parameter named ceyex. All covariances that are not listed in the COV statement and that are not determined by the model are assumed to be zero. If the model contained two or more manifest exogenous variables, their covariances would be set as free parameters by default.

Identification of Models

Unfortunately, if you try to fit models of form B or form C without additional constraints, you cannot obtain unique estimates of the parameters. These models have four parameters (one coefficient and three variances). The covariance matrix of the observed variables Y and X has only three elements that are free to vary, since Cov(Y,X)=Cov(X,Y). The covariance structure can, therefore, be expressed as three equations in four unknown parameters. Since there are fewer equations than unknowns, there are many different sets of values for the parameters that provide a solution for the equations. Such a model is said to be underidentified.

If the number of parameters equals the number of free elements in the covariance matrix, then there might exist a unique set of parameter estimates that exactly reproduce the observed covariance matrix. In this case, the model is said to be just identified or saturated.

If the number of parameters is less than the number of free elements in the covariance matrix, there might exist no set of parameter estimates that reproduces the observed covariance matrix. In this case, the model is said to be overidentified. Various statistical criteria, such as maximum likelihood, can be used to choose parameter estimates that approximately reproduce the observed covariance matrix. If you use ML, GLS, or WLS estimation, PROC TCALIS can perform a statistical test of the goodness of fit of the model under the certain statistical assumptions.

If the model is just identified or overidentified, it is said to be identified. If you use ML, GLS, or WLS estimation for an identified model, PROC TCALIS can compute approximate standard errors for the parameter estimates. For underidentified models, PROC TCALIS obtains approximate standard errors by imposing additional constraints resulting from the use of a generalized inverse of the Hessian matrix.

You cannot guarantee that a model is identified simply by counting the parameters. For example, for any latent variable, you must specify a numeric value for the variance, or for some covariance involving the variable, or for a coefficient of the variable in at least one equation. Otherwise, the scale of the latent variable is indeterminate, and the model will be underidentified regardless of the number of parameters and the size of the covariance matrix. As another example, an exploratory factor analysis with two or more common factors is always underidentified because you can rotate the common factors without affecting the fit of the model.

PROC TCALIS can usually detect an underidentified model by computing the approximate covariance matrix of the parameter estimates and checking whether any estimate is linearly related to other estimates (Bollen 1989, pp. 248–250), in which case PROC TCALIS displays equations showing the linear relationships among the estimates. Another way to obtain empirical evidence regarding the identification of a model is to run the analysis several times with different initial estimates to see if the same final estimates are obtained.

Bollen (1989) provides detailed discussions of conditions for identification in a variety of models.

The following example is inspired by Fuller (1987, pp. 40–41). The hypothetical data are counts of two types of cells, cells forming rosettes and nucleated cells, in spleen samples. It is reasonable to assume that counts have a Poisson distribution; hence, the square roots of the counts should have a constant error variance of 0.25.

You can use PROC TCALIS to fit a model of form C to the square roots of the counts without constraints on the parameters, as displayed in the following statements:

```
data spleen;
   input rosette nucleate;
   sqrtrose=sqrt(rosette);
   sqrtnucl=sqrt(nucleate);
   datalines;
4 62
5 87
5 117
6 142
8 212
9 120
12 254
13 179
15 125
19 182
28 301
51 357
;
proc tcalis data=spleen;
   lineqs sqrtrose = factrose + err_rose,
          sqrtnucl = factnucl + err_nucl,
          factrose = beta factnucl + disturb;
   std err_rose = v_rose,
       err_nucl = v_nucl,
       factnucl = v_factnu,
       disturb  = 0;
run;
```

This model is underidentified. PROC TCALIS displays the following warning:

```
WARNING: Estimation problem not identified: More parameters to
         estimate ( 4 ) than the total number of mean and
         covariance elements ( 3 ).
```

Then it diagnoses the indeterminacy as follows:

```
NOTE: Covariance matrix for the estimates is not full rank.
NOTE: The variance of some parameter estimates is zero or
      some parameter estimates are linearly related to other
      parameter estimates as shown in the following equations:

v_rose     =       0.207718   +       0.108978   *   beta
                              +       0.916873   *   v_nucl
                              -       0.916873   *   v_factnu
```

The constraint that the error variances equal 0.25 can be imposed by modifying the STD statement:

```
proc tcalis data=spleen;
   lineqs sqrtrose = factrose + err_rose,
          sqrtnucl = factnucl + err_nucl,
          factrose = beta factnucl + disturb;
   std err_rose = .25,
       err_nucl = .25,
       factnucl = v_factnu,
       disturb  = 0;
run;
```

This model is overidentified and the chi-square goodness-of-fit test yields a p-value of 0.0219, as displayed in Figure 17.2.

Figure 17.2 Spleen Data: Fit Statistics for Overidentified Model

Fit Summary

Modeling Info	N Observations	12
	N Variables	2
	N Moments	3
	N Parameters	2
	N Active Constraints	0
	Independence Model Chi-Square	13.2732
	Independence Model Chi-Square DF	1
Absolute Index	Fit Function	0.4775
	Chi-Square	5.2522
	Chi-Square DF	1
	Pr > Chi-Square	0.0219
	Z-Test of Wilson & Hilferty	2.0375
	Hoelter Critical N	10
	Root Mean Square Residual (RMSR)	0.1785
	Standardized RMSR (SRMSR)	0.0745
	Goodness of Fit Index (GFI)	0.7274
Parsimony Index	Adjusted GFI (AGFI)	0.1821
	Parsimonious GFI	0.7274
	RMSEA Estimate	0.6217
	RMSEA Lower 90% Confidence Limit	0.1899
	RMSEA Upper 90% Confidence Limit	1.1869
	Probability of Close Fit	0.0237
	ECVI Estimate	0.9775
	ECVI Lower 90% Confidence Limit	.
	ECVI Upper 90% Confidence Limit	2.2444
	Akaike Information Criterion	3.2522
	Bozdogan CAIC	1.7673
	Schwarz Bayesian Criterion	2.7673
	McDonald Centrality	0.8376
Incremental Index	Bentler Comparative Fit Index	0.6535
	Bentler-Bonett NFI	0.6043
	Bentler-Bonett Non-normed Index	0.6535
	Bollen Normed Index Rho1	0.6043
	Bollen Non-normed Index Delta2	0.6535
	James et al. Parsimonious NFI	0.6043

PROC TCALIS arranges the fit statistics according to their types: absolute, parsimony, and incremental. After displaying some important modeling information in the fit summary table, the absolute fit indices are printed. Absolute indices are those model fit statistics that assess the model fit without comparing to a "null" model. The most typical fit index of the this type is the chi-square fit statistic. Nonsignificance of the chi-square indicates good model fit. Other popular absolute fit indices include the root mean square residual (RMSR), the standardized RMSR (SRMSR), and the goodness-of-fit index (GFI). By convention, a good model should have an SRMSR smaller than 0.05 and a GFI larger than 0.90.

Parsimony indices are those fit indices that assess the model fit without comparing with a null model, but with the number of parameters in the model taking into account. These fit indices favor precise models. If two models have the same chi-square value for the same data set but have different number of parameters in the models, the model with fewer parameters will have a better parsimony fit statistic. The most popular parsimony index displayed in the table is perhaps the root mean squared error of approximation, or RMSEA (Steiger and Lind 1980). An RMSEA below 0.05 is recommended for a good model fit (Browne and Cudeck 1993). Another popular index in this category is the adjusted GFI (AGFI). By convention, an AGFI above 0.90 is required for a good model fit.

Finally, the incremental fit indices are those indices that measure model fit by comparing with a null model. A null model is usually the independence model that assumes the measured variables are all uncorrelated. The most popular incremental index is Bentler's CFI. By convention, a CFI above 0.90 is required for a good model fit.

After the model fit summary, the parameter estimates are displayed in Figure 17.3.

Figure 17.3 Spleen Data: Parameter Estimates for Overidentified Model

Linear Equations

```
sqrtrose =    1.0000 factrose +  1.0000 err_rose
sqrtnucl =    1.0000 factnucl +  1.0000 err_nucl
factrose =    0.4034*factnucl +  1.0000 disturb
Std Err       0.0508 beta
t Value       7.9439
```

Estimates for Variances of Exogenous Variables

Variable Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	err_rose		0.25000		
	err_nucl		0.25000		
Latent	factnucl	v_factnu	10.45846	4.56608	2.29047
Disturbance	disturb		0		

Overall, the model does not provide a good fit. The sample size is so small that the p-value of the chi-square test should not be taken to be accurate, but to get a small p-value with such a small sample indicates it is possible that the model is seriously deficient. The deficiency could be due to any of the following:

- The error variances are not both equal to 0.25.
- The error terms are correlated with each other or with the true scores.
- The observations are not independent.
- There is a nonzero disturbance in the linear relation between factrose and factnucl.
- The relation between factrose and factnucl is not linear.
- The actual distributions are not adequately approximated by the multivariate normal distribution.

A simple and plausible modification to the model is to make the "disturbance term" disturb a real random variable with nonzero variance in the structural model. This can be done by giving the variance of disturb a parameter name in the STD statement, as shown in the following statements:

```
proc tcalis data=spleen;
   lineqs sqrtrose = factrose + err_rose,
          sqrtnucl = factnucl + err_nucl,
          factrose = beta factnucl + disturb;
   std err_rose = .25,
       err_nucl = .25,
       factnucl = v_factnu,
       disturb  = v_dist;
run;
```

In the STD statement, v_dist is now specified as a free variance parameter to be estimated. The parameter estimates are produced in Figure 17.4.

Figure 17.4 Spleen Data: Parameter Estimated for Just Identified Model

```
                          Linear Equations

              sqrtrose =    1.0000 factrose +  1.0000 err_rose
              sqrtnucl =    1.0000 factnucl +  1.0000 err_nucl
              factrose =    0.3907*factnucl +  1.0000 disturb
              Std Err       0.0771 beta
              t Value       5.0692

               Estimates for Variances of Exogenous Variables

Variable                                                  Standard
Type          Variable    Parameter      Estimate          Error       t Value

Error         err_rose                    0.25000
              err_nucl                    0.25000
Latent        factnucl    v_factnu       10.50458        4.58577       2.29069
Disturbance   disturb     v_dist          0.38153        0.28556       1.33607
```

As shown in Figure 17.4, the variance of disturb is estimated at 0.382. Due to the inclusion of this new parameter, estimates for beta and v_factnu also shift a little bit from the previous analysis.

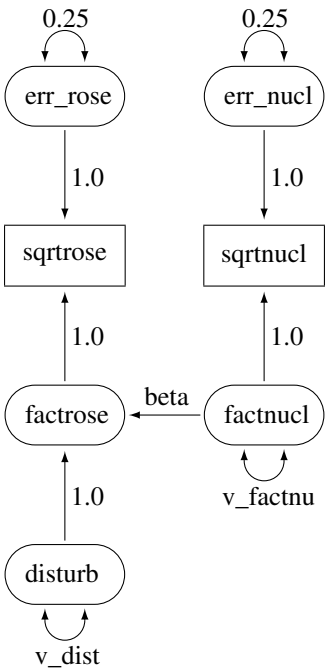
Because this model is just identified or saturated, there are no degrees of freedom for the chi-square goodness-of-fit test.

Path Diagrams and the PATH Modeling Language

Complicated models are often easier to understand when they are expressed as path diagrams. One advantage of path diagrams over equations is that variances and covariances can be shown directly in the path diagram. Loehlin (1987) provides a detailed discussion of path diagrams. Another advantage is that the path diagram can be translated easily into the PATH modeling language supported by PROC TCALIS.

It is customary to write the names of manifest variables in rectangles and the names of latent variables in ovals. The coefficients in each equation are indicated by drawing arrows from the independent variables to the dependent variable. Covariances between exogenous variables are drawn as two-headed arrows. The variance of an exogenous variable can be displayed as a two-headed arrow with both heads pointing to the exogenous variable, since the variance of a variable is the covariance of the variable with itself. Figure 17.5 displays a path diagram for the spleen data, explicitly showing all latent variables (including error terms) and variances of exogenous variables.

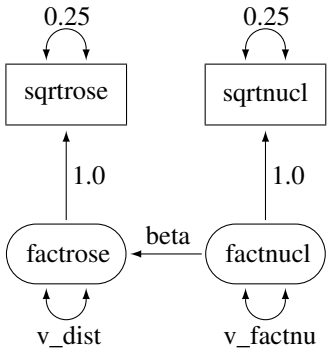
Figure 17.5 Path Diagram: Spleen



There is an easier way to draw the path diagram based on McArdle's reticular action model (RAM) (McArdle and McDonald 1984). McArdle uses the convention that a two-headed arrow that points to an endogenous variable actually refers to the error or disturbance term associated with that variable. A two-headed arrow with both heads pointing to the same endogenous variable represents the

error or disturbance variance for the equation that determines the endogenous variable; there is no need to draw a separate oval for the error or disturbance term. Similarly, a two-headed arrow connecting two endogenous variables represents the covariance between the error of disturbance terms associated with the endogenous variables. The RAM conventions enable the previous path diagram to be simplified, as shown in Figure 17.6.

Figure 17.6 Simplified Path Diagram: Spleen



The PATH modeling language in PROC TCALIS provides a simple way to transcribe a path diagram based on the reticular action model. In the PATH modeling languages, there are three statements to capture the specifications in path diagrams:

- The PATH statement enables you to specify each of the one-headed arrows (paths). The parameters specified in the PATH statement are the path (regression) coefficients.
- The PVAR statement enables you to specify each of the double-headed arrows with both heads pointing to the ***same*** variable. In general, you specify *partial* (or total) variance parameters in the PVAR statement. If the variable being pointed at is *exogenous*, a (total) variance parameter is specified. If the variable being pointed at is *endogenous*, a partial or an error variance parameter is specified.
- The PCOV statement enables you to specify each of the double-headed arrows with its heads pointing to ***different*** variables. In general, you specify (*partial*) covariance parameters in the PCOV statement. The two most common cases are as follows: (1) If the heads of a double-headed arrow are connecting two *exogenous* variables, a covariance parameter between the two variables is specified; and (2) If the heads of a double-headed arrow are connecting two *endogenous* variables, an error covariance parameter for the two variables is specified. This error covariance is also a partial covariance between the endogenous variables.

For example, the path diagram for the spleen data in Figure 17.6 can be specified with the PATH modeling language as follows:

```
proc tcalis data=spleen outmodel=splmod1;
   path
      sqrtrose <- factrose   1.0,
      sqrtnucl <- factnucl   1.0,
      factrose <- factnucl   beta;
   pvar
      sqrtrose = 0.25,      /* error variance for sqrtrose */
      sqrtnucl = 0.25,      /* error variance for sqrtnucl */
      factrose = v_dist,    /* disturbance variance for factrose */
      factnucl = v_factnu;  /* variance of factnucl */
run;
```

One notable item in the specification is that each of the single-headed or double-headed arrows in the path diagram is transcribed into an entry in either the PATH or PVAR statement:

- **PATH** statement:
  The paths "sqrtrose <– factrose" and "sqrtnucl <– factnucl" in the PATH statement are followed by the constant 1, indicating fixed path coefficients. The path "factrose <– factnucl" is followed by a parameter named beta, indicating a free path coefficient to estimate in the model.

- **PVAR** statement:
  A fixed value 0.25 is specified after the equal signs of sqrtrose and sqrtnucl in the PVAR statement. Because sqrtrose and sqrtnucl are endogenous in the model, you are fixing the error variances of sqrtrose and sqrtnucl to 0.25 in the specification.
  In the last two entries of the PVAR statement, you are putting parameter names after the equal signs. Because factrose and factnucl are exogenous in the model, v_dist and v_factnu are variance parameters of factrose and factnucl, respectively.

Because there are no double-headed arrows each pointing to different variables in the path diagram, the PCOV statement is not needed in the model specification. The resulting output of the PATH model is displayed in Figure 17.7.

Figure 17.7 Spleen Data: RAM Model

PATH List

Path			Parameter	Estimate	Standard Error	t Value
sqrtrose	<-	factrose		1.00000		
sqrtnucl	<-	factnucl		1.00000		
factrose	<-	factnucl	beta	0.39074	0.07708	5.06920

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	sqrtrose		0.25000		
	sqrtnucl		0.25000		
	factrose	v_dist	0.38153	0.28556	1.33607
Exogenous	factnucl	v_factnu	10.50458	4.58577	2.29069

In the PROC TCALIS statement, the OUTMODEL=SPLMOD1 option is used. This will save the model specification, together with final estimates in a SAS data set called SPLMOD1. This special type of SAS data set is called "CALISMDL." The following statements are used to display the contents of this OUTMODEL= data set:

```
proc print data=splmod1;
run;
```

As displayed in Figure 17.8, the first record saves the model type, which is the PATH model specification in this case. The next seven records save the information about the PATH model: 3 paths and 4 partial variances specifications.

Figure 17.8 Spleen Data: OUTMODEL= Data Set with Final Parameter Estimates

Obs	_TYPE_	_NAME_	_VAR1_	_VAR2_	_ESTIM_	_STDERR_
1	MDLTYPE	PATH			.	.
2	LEFT		sqrtrose	factrose	1.0000	.
3	LEFT		sqrtnucl	factnucl	1.0000	.
4	LEFT	beta	factrose	factnucl	0.3907	0.07708
5	PVAR		sqrtrose		0.2500	.
6	PVAR		sqrtnucl		0.2500	.
7	PVAR	v_dist	factrose		0.3815	0.28556
8	PVAR	v_factnu	factnucl		10.5046	4.58577

In each record, the variables involved, the parameter name, the final estimate, and the standard error estimate are stored. For records with fixed parameters, the parameter names entries are blanks and the standard error estimates are indicated by missing values. This data set can be used as input to another run of the TCALIS procedure with the INMODEL= option in the PROC TCALIS statement. For example, if the iteration limit is exceeded, you can use the CALISMODEL data set to start a new run that begins with the final estimates from the last run. Or you can change the data set to add or remove constraints or modify the model in various other ways. The easiest way to change a CALISMDL data set is to use the FSEDIT procedure, but you can also use a DATA step. For example, you could set the variance of the disturbance term to zero, effectively removing the disturbance from the equation, by removing the parameter name v_dist in the _NAME_ variable and setting the value of the estimate to zero in the _ESTIM_ variable:

```
data splmod2(type=calismdl);
   set splmod1;
   if _name_='v_dist' then
      do;
         _name_=' ';
         _estim_=0;
      end;
run;
```

Hence, due to the fixed zero error variance for factrose, a model with perfect prediction of factrose from factnucl is specified in the new CALISMDL data set SPLMOD2. This data set serves as the INMODEL= data set in the following statements for another PROC TCALIS run:

```
proc tcalis data=spleen inmodel=splmod2;
run;
```

The main estimation results are displayed in Figure 17.9.

Figure 17.9 Spleen Data: PATH Model Estimates with INMODEL= Data Set

PATH List

Path			Parameter	Estimate	Standard Error	t Value
sqrtrose	<-	factrose		1.00000		
sqrtnucl	<-	factnucl		1.00000		
factrose	<-	factnucl	beta	0.40340	0.05078	7.94391

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	sqrtrose		0.25000		
	sqrtnucl		0.25000		
	factrose		0		
Exogenous	factnucl	v_factnu	10.45846	4.56608	2.29047

As can be seen in Figure 17.9, error variance for factrose is a fixed zero in the final results. Because of this modification in the current model specified in the SPLMOD2 data set, the estimates of beta and v_factnu are different from those of the previous model results, which are stored in the SPLMOD1 data set.

Some Measurement Models

Psychometric test theory involves many kinds of models relating scores on psychological and educational tests to latent variables representing intelligence or various underlying abilities. The following example uses data on four vocabulary tests from Lord (1957). Tests W and X have 15 items each and are administered with very liberal time limits. Tests Y and Z have 75 items and are administered under time pressure. The covariance matrix is read by the following DATA step:

```
data lord(type=cov);
   input _type_ $ _name_ $ w x y z;
   datalines;
n    . 649       .       .       .
cov w 86.3979    .       .       .
cov x 57.7751 86.2632    .       .
cov y 56.8651 59.3177 97.2850    .
cov z 58.8986 59.6683 73.8201 97.8192
;
```

The psychometric model of interest states that W and X are determined by a single common factor F_{WX}, and Y and Z are determined by a single common factor F_{YZ}. The two common factors are expected to have a positive correlation, and it is desired to estimate this correlation. It is convenient to assume that the common factors have unit variance, so their correlation will be equal to their covariance. The error terms for all the manifest variables are assumed to be uncorrelated with each other and with the common factors. The model (labeled here as model form D) is as follows.

Model Form D

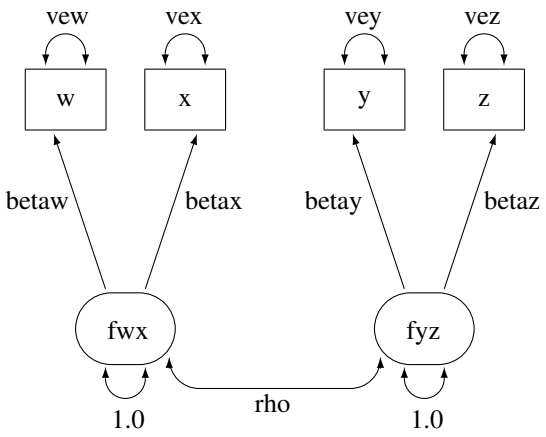
$$
\begin{aligned}
W &= \beta_W F_{WX} + E_W \\
X &= \beta_X F_{WX} + E_X \\
Y &= \beta_Y F_{YZ} + E_Y \\
Z &= \beta_Z F_{YZ} + E_Z
\end{aligned}
$$

with the following assumptions:

$$
\begin{aligned}
\text{Var}(F_{WX}) &= \text{Var}(F_{YZ}) = 1 \\
\text{Cov}(F_{WX}, F_{YZ}) &= \rho \\
\text{Cov}(E_W, E_X) &= \text{Cov}(E_W, E_Y) = \text{Cov}(E_W, E_Z) = \text{Cov}(E_X, E_Y) \\
&= \text{Cov}(E_X, E_Z) = \text{Cov}(E_Y, E_Z) = \text{Cov}(E_W, F_{WX}) \\
&= \text{Cov}(E_W, F_{YZ}) = \text{Cov}(E_X, F_{WX}) = \text{Cov}(E_X, F_{YZ}) \\
&= \text{Cov}(E_Y, F_{WX}) = \text{Cov}(E_Y, F_{YZ}) = \text{Cov}(E_Z, F_{WX}) \\
&= \text{Cov}(E_Z, F_{YZ}) = 0
\end{aligned}
$$

The corresponding path diagram is shown in Figure 17.10.

Figure 17.10 Path Diagram: Lord



With the following rules, the conversion from the path diagram to the PATH model specification is very straightforward:

- Each single-headed arrow in the path diagram is specified in the PATH statement.
- Each double-headed arrow that points to a single variable is specified in the PVAR statement.
- Each double-headed arrow that points to two distinct variables is specified in the PCOV statement.

Hence, this path diagram can be converted easily to a PATH model as follows:

```
title 'H4: Unconstrained';
proc tcalis data=lord outmodel=model4;
   path
      w <- fwx    betaw,
      x <- fwx    betax,
      y <- fyz    betay,
      z <- fyz    betaz;
   pvar
      fwx fyz = 2 * 1.0,
      w x y z = vew vex vey vez;
   pcov
      fwx fyz = rho;
run;
```

The major results are displayed in Figure 17.11 and Figure 17.12.

Figure 17.11 Lord Data: Fit Summary, Hypothesis H4

Fit Summary

Modeling Info	N Observations	649
	N Variables	4
	N Moments	10
	N Parameters	9
	N Active Constraints	0
	Independence Model Chi-Square	1466.5524
	Independence Model Chi-Square DF	6
Absolute Index	Fit Function	0.0011
	Chi-Square	0.7030
	Chi-Square DF	1
	Pr > Chi-Square	0.4018
	Z-Test of Wilson & Hilferty	0.2363
	Hoelter Critical N	3543
	Root Mean Square Residual (RMSR)	0.2720
	Standardized RMSR (SRMSR)	0.0030
	Goodness of Fit Index (GFI)	0.9995
Parsimony Index	Adjusted GFI (AGFI)	0.9946
	Parsimonious GFI	0.1666
	RMSEA Estimate	0.0000
	RMSEA Lower 90% Confidence Limit	.
	RMSEA Upper 90% Confidence Limit	0.0974
	Probability of Close Fit	0.6854
	ECVI Estimate	0.0291
	ECVI Lower 90% Confidence Limit	.
	ECVI Upper 90% Confidence Limit	0.0391
	Akaike Information Criterion	-1.2970
	Bozdogan CAIC	-6.7725
	Schwarz Bayesian Criterion	-5.7725
	McDonald Centrality	1.0002
Incremental Index	Bentler Comparative Fit Index	1.0000
	Bentler-Bonett NFI	0.9995
	Bentler-Bonett Non-normed Index	1.0012
	Bollen Normed Index Rho1	0.9971
	Bollen Non-normed Index Delta2	1.0002
	James et al. Parsimonious NFI	0.1666

Figure 17.12 Lord Data: Estimation Results, Hypothesis H4

PATH List

-------Path-------			Parameter	Estimate	Standard Error	t Value
w	<-	fwx	betaw	7.50066	0.32339	23.19390
x	<-	fwx	betax	7.70266	0.32063	24.02354
y	<-	fyz	betay	8.50947	0.32694	26.02730
z	<-	fyz	betaz	8.67505	0.32560	26.64301

Figure 17.12 *continued*

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous	fwx		1.00000		
	fyz		1.00000		
Error	w	vew	30.13796	2.47037	12.19979
	x	vex	26.93217	2.43065	11.08021
	y	vey	24.87396	2.35986	10.54044
	z	vez	22.56264	2.35028	9.60000

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
fwx	fyz	rho	0.89855	0.01865	48.17998

It is convenient to create the OUTMODEL= data set called model4 for use in fitting other models with additional constraints. The same analysis can be performed with the LINEQS statement, as specified in the following:

```
title 'H4: Unconstrained; LINEQS Specification';
proc tcalis data=lord;
   lineqs w = betaw fwx + ew,
          x = betax fwx + ex,
          y = betay fyz + ey,
          z = betaz fyz + ez;
   std fwx fyz = 2 * 1.,
       ew ex ey ez = vew vex vey vez;
   cov fwx fyz = rho;
run;
```

Unlike the PATH model specification, in the LINEQS specification you need to specify the error terms explicitly in the LINEQS statement. In the STD statement, you would need to specify the variance parameters for the exogenous variables, including both of the factors and the error terms. However, using the PATH model specification, no explicit names for error or disturbance terms are needed. As a result, the exogenous variance and error variance parameters are both specified in the PVAR statement. This treatment generalizes to the following useful rule about the PATH model specification:

- Each variable in the PATH model specification or path diagram should have a variance or partial variance parameter specified in the PVAR statement—as either an exogenous variance or a partial variance due to error.

The main results from the LINEQS model specification are displayed in Figure 17.13.

Figure 17.13 Lord Data: Using LINEQS Statement for RAM Model, Hypothesis H4

Linear Equations

w	=	7.5007*fwx	+ 1.0000 ew
Std Err		0.3234 betaw	
t Value		23.1939	
x	=	7.7027*fwx	+ 1.0000 ex
Std Err		0.3206 betax	
t Value		24.0235	
y	=	8.5095*fyz	+ 1.0000 ey
Std Err		0.3269 betay	
t Value		26.0273	
z	=	8.6751*fyz	+ 1.0000 ez
Std Err		0.3256 betaz	
t Value		26.6430	

Estimates for Variances of Exogenous Variables

Variable Type	Variable	Parameter	Estimate	Standard Error	t Value
Latent	fwx		1.00000		
	fyz		1.00000		
Error	ew	vew	30.13796	2.47037	12.19979
	ex	vex	26.93217	2.43065	11.08021
	ey	vey	24.87396	2.35986	10.54044
	ez	vez	22.56264	2.35028	9.60000

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
fwx	fyz	rho	0.89855	0.01865	48.17998

Aside from the output format, all estimates in the LINEQS model results in Figure 17.13 match those of the PATH model results in Figure 17.12. In some situations, the PATH and LINEQS statements might yield slightly different results due to the inexactness of the numerical optimization; the discrepancies can be reduced by specifying a more stringent convergence criterion such as GCONV=1E–4 or GCONV=1E–6.

Subsequent analyses are illustrated with the PATH statement rather than the LINEQS statement because it is easier to translate the path diagram to the PATH model specification.

In an analysis of these data by Jöreskog and Sörbom (1979, pp. 54–56; Loehlin 1987, pp. 84–87), four hypotheses are considered:

H_1: $\rho = 1$,
$\beta_W = \beta_X$, $\mathrm{Var}(E_W) = \mathrm{Var}(E_X) (or \mathrm{vew} = \mathrm{vex})$,
$\beta_Y = \beta_Z$, $\mathrm{Var}(E_Y) = \mathrm{Var}(E_Z) (or \mathrm{vey} = \mathrm{vez})$

H_2: same as H_1: except ρ is unconstrained

H_3: $\rho = 1$

H_4: model form D without any additional constraints

The hypothesis H_3 says that there is really just one common factor instead of two; in the terminology of test theory, W, X, Y, and Z are said to be congeneric. The hypothesis H_2 says that W and X have the same true scores and have equal error variance; such tests are said to be parallel. The hypothesis H_2 also requires Y and Z to be parallel. The hypothesis H_1 says that W and X are parallel tests, Y and Z are parallel tests, and all four tests are congeneric.

It is most convenient to fit the models in the opposite order from that in which they are numbered. The previous analysis fit the model for H_4 and created an OUTMODEL= data set called model4. The hypothesis H_3 can be fitted directly or by modifying the model4 data set. Since H_3 differs from H_4 only in that ρ is constrained to equal 1, the model4 data set can be modified by finding the observation for which _NAME_='rho' and changing the variable _NAME_ to a blank value (meaning that the observation represents a constant rather than a parameter to be fitted) and by setting the variable _ESTIM_ to the value 1. The following statements create a new model stored in the model3 data set that is modified from the model4 data set:

```
data model3(type=calismdl);
   set model4;
   if _name_='rho' then
      do;
         _name_=' ';
         _estim_=1;
      end;
run;
```

In other words, the model information stored in data set model3 is specified exactly as hypothesis H_3 requires. This data set is then read as an INMODEL= data set for the following PROC TCALIS run:

```
title 'H3: W, X, Y, and Z are congeneric';
proc tcalis data=lord inmodel=model3;
run;
```

Another way to specify the model under hypothesis H_3 is to specify the entire PATH model anew, such as in the following statements:

```
title 'H3: W, X, Y, and Z are congeneric';
proc tcalis data=lord;
   path  w <- f  betaw,
         x <- f  betax,
         y <- f  betay,
         z <- f  betaz;
   pvar
      f = 1,
      w x y z = vew vex vey vez;
run;
```

This would produce essentially the same results as those of the analysis based on the model stored in the data set model3. The main results from the analysis with the INMODEL=MODEL3 data set are displayed in Figure 17.14.

Figure 17.14 Lord Data: Major Results for Hypothesis H3

Fit Summary

Modeling Info	N Observations	649
	N Variables	4
	N Moments	10
	N Parameters	8
	N Active Constraints	0
	Independence Model Chi-Square	1466.5524
	Independence Model Chi-Square DF	6
Absolute Index	Fit Function	0.0559
	Chi-Square	36.2095
	Chi-Square DF	2
	Pr > Chi-Square	0.0000
	Z-Test of Wilson & Hilferty	5.2108
	Hoelter Critical N	109
	Root Mean Square Residual (RMSR)	2.4636
	Standardized RMSR (SRMSR)	0.0277
	Goodness of Fit Index (GFI)	0.9714
Parsimony Index	Adjusted GFI (AGFI)	0.8570
	Parsimonious GFI	0.3238
	RMSEA Estimate	0.1625
	RMSEA Lower 90% Confidence Limit	0.1187
	RMSEA Upper 90% Confidence Limit	0.2108
	Probability of Close Fit	0.0000
	ECVI Estimate	0.0808
	ECVI Lower 90% Confidence Limit	0.0561
	ECVI Upper 90% Confidence Limit	0.1170
	Akaike Information Criterion	32.2095
	Bozdogan CAIC	21.2586
	Schwarz Bayesian Criterion	23.2586
	McDonald Centrality	0.9740
Incremental Index	Bentler Comparative Fit Index	0.9766
	Bentler-Bonett NFI	0.9753
	Bentler-Bonett Non-normed Index	0.9297
	Bollen Normed Index Rho1	0.9259
	Bollen Non-normed Index Delta2	0.9766
	James et al. Parsimonious NFI	0.3251

Figure 17.14 *continued*

PATH List

Path			Parameter	Estimate	Standard Error	t Value
w	<-	fwx	betaw	7.10472	0.32177	22.08019
x	<-	fwx	betax	7.26906	0.31826	22.83965
y	<-	fyz	betay	8.37348	0.32542	25.73160
z	<-	fyz	betaz	8.51057	0.32409	26.25985

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous	fwx		1.00000		
	fyz		1.00000		
Error	w	vew	35.92087	2.41466	14.87615
	x	vex	33.42397	2.31038	14.46688
	y	vey	27.16980	2.24619	12.09595
	z	vez	25.38948	2.20839	11.49684

Covariances Among Exogenous Variables

Var1	Var2	Estimate	Standard Error	t Value
fwx	fyz	1.00000		

The hypothesis H_2 requires that several pairs of parameters be constrained to have equal estimates. With PROC TCALIS, you can impose this constraint by giving the same name to parameters that are constrained to be equal. This can be done directly in the PATH and PVAR statements or by using the DATA step to change the values in the model4 data set.

First, you can specify the model directly under the hypothesis H_2; the following PATH model is specified:

```
title 'H2: W and X parallel, Y and Z parallel';
proc tcalis data=lord;
   path
      w <- fwx   betawx,
      x <- fwx   betawx,
      y <- fyz   betayz,
      z <- fyz   betayz;
   pvar
      fwx fyz = 2 * 1.0,
      w x y z = vewx vewx veyz veyz;
   pcov
      fwx fyz = rho;
run;
```

Alternatively, if you use the DATA step to modify from the model4 data set, you would specify a new data set called model2 for storing the model information under the hypothesis H_2, as shown in the following statements:

```
data model2(type=calismdl);
   set model4;
   if _name_='betaw' then _name_='betawx';
   if _name_='betax' then _name_='betawx';
   if _name_='betay' then _name_='betayz';
   if _name_='betaz' then _name_='betayz';
   if _name_='vew' then _name_='vewx';
   if _name_='vex' then _name_='vewx';
   if _name_='vey' then _name_='veyz';
   if _name_='vez' then _name_='veyz';
run;
```

Then you would use model2 as the INMODEL= data set in the following PROC TCALIS run:

```
title 'H2: W and X parallel, Y and Z parallel';
proc tcalis data=lord inmodel=model2;
run;
```

The main results from either of these analyses are displayed in Figure 17.15.

Figure 17.15 Lord Data: Major Results for Hypothesis H2

Fit Summary

Modeling Info	N Observations	649
	N Variables	4
	N Moments	10
	N Parameters	5
	N Active Constraints	0
	Independence Model Chi-Square	1466.5524
	Independence Model Chi-Square DF	6
Absolute Index	Fit Function	0.0030
	Chi-Square	1.9335
	Chi-Square DF	5
	Pr > Chi-Square	0.8583
	Z-Test of Wilson & Hilferty	-1.0768
	Hoelter Critical N	3712
	Root Mean Square Residual (RMSR)	0.6983
	Standardized RMSR (SRMSR)	0.0076
	Goodness of Fit Index (GFI)	0.9985
Parsimony Index	Adjusted GFI (AGFI)	0.9970
	Parsimonious GFI	0.8321
	RMSEA Estimate	0.0000
	RMSEA Lower 90% Confidence Limit	.
	RMSEA Upper 90% Confidence Limit	0.0293
	Probability of Close Fit	0.9936
	ECVI Estimate	0.0185
	ECVI Lower 90% Confidence Limit	.
	ECVI Upper 90% Confidence Limit	0.0276
	Akaike Information Criterion	-8.0665
	Bozdogan CAIC	-35.4436
	Schwarz Bayesian Criterion	-30.4436
	McDonald Centrality	1.0024
Incremental Index	Bentler Comparative Fit Index	1.0000
	Bentler-Bonett NFI	0.9987
	Bentler-Bonett Non-normed Index	1.0025
	Bollen Normed Index Rho1	0.9984
	Bollen Non-normed Index Delta2	1.0021
	James et al. Parsimonious NFI	0.8322

PATH List

Path			Parameter	Estimate	Standard Error	t Value
w	<-	fwx	betawx	7.60099	0.26844	28.31580
x	<-	fwx	betawx	7.60099	0.26844	28.31580
y	<-	fyz	betayz	8.59186	0.27967	30.72146
z	<-	fyz	betayz	8.59186	0.27967	30.72146

Figure 17.15 *continued*

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous	fwx		1.00000		
	fyz		1.00000		
Error	w	vewx	28.55545	1.58641	18.00000
	x	vewx	28.55545	1.58641	18.00000
	y	veyz	23.73200	1.31844	18.00000
	z	veyz	23.73200	1.31844	18.00000

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
fwx	fyz	rho	0.89864	0.01865	48.18011

The hypothesis H_1 requires one more constraint in addition to those in H_2. Again, there are two ways to do this. First, a direct model specification is shown in the following statements:

```
title 'H1: W and X parallel, Y and Z parallel, all congeneric';
proc tcalis data=lord;
   path
      w <- f   betawx,
      x <- f   betawx,
      y <- f   betayz,
      z <- f   betayz;
   pvar
      f       = 1.0,
      w x y z = vewx vewx veyz veyz;
run;
```

Alternatively, you can modify the model2 data set to create a new data set model2 that stores the model information required by the hypothesis H_1, as shown in the following statements:

```
data model1(type=calismdl);
   set model2;
   if _name_='rho' then
      do;
         _name_=' ';
         _estim_=1;
      end;
run;
```

You can then pass the model information stored in model1 as an INMODEL= data set in the following PROC TCALIS run:

```
title 'H1: W and X parallel, Y and Z parallel, all congeneric';
proc tcalis data=lord inmodel=model1;
run;
```

The main results from either of these analyses are displayed in Figure 17.16.

Figure 17.16 Lord Data: Major Results for Hypothesis H1

Fit Summary

Modeling Info	N Observations	649
	N Variables	4
	N Moments	10
	N Parameters	4
	N Active Constraints	0
	Independence Model Chi-Square	1466.5524
	Independence Model Chi-Square DF	6
Absolute Index	Fit Function	0.0576
	Chi-Square	37.3337
	Chi-Square DF	6
	Pr > Chi-Square	0.0000
	Z-Test of Wilson & Hilferty	4.5535
	Hoelter Critical N	220
	Root Mean Square Residual (RMSR)	2.5430
	Standardized RMSR (SRMSR)	0.0286
	Goodness of Fit Index (GFI)	0.9705
Parsimony Index	Adjusted GFI (AGFI)	0.9509
	Parsimonious GFI	0.9705
	RMSEA Estimate	0.0898
	RMSEA Lower 90% Confidence Limit	0.0635
	RMSEA Upper 90% Confidence Limit	0.1184
	Probability of Close Fit	0.0076
	ECVI Estimate	0.0701
	ECVI Lower 90% Confidence Limit	0.0458
	ECVI Upper 90% Confidence Limit	0.1059
	Akaike Information Criterion	25.3337
	Bozdogan CAIC	-7.5189
	Schwarz Bayesian Criterion	-1.5189
	McDonald Centrality	0.9761
Incremental Index	Bentler Comparative Fit Index	0.9785
	Bentler-Bonett NFI	0.9745
	Bentler-Bonett Non-normed Index	0.9785
	Bollen Normed Index Rho1	0.9745
	Bollen Non-normed Index Delta2	0.9785
	James et al. Parsimonious NFI	0.9745

Figure 17.16 *continued*

PATH List

Path			Parameter	Estimate	Standard Error	t Value
w	<-	fwx	betawx	7.18622	0.26598	27.01798
x	<-	fwx	betawx	7.18622	0.26598	27.01798
y	<-	fyz	betayz	8.44198	0.28000	30.14946
z	<-	fyz	betayz	8.44198	0.28000	30.14946

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Exogenous	fwx		1.00000		
	fyz		1.00000		
Error	w	vewx	34.68878	1.64635	21.07013
	x	vewx	34.68878	1.64635	21.07013
	y	veyz	26.28501	1.39954	18.78118
	z	veyz	26.28501	1.39954	18.78118

Covariances Among Exogenous Variables

Var1	Var2	Estimate	Standard Error	t Value
fwx	fyz	1.00000		

The goodness-of-fit tests for the four hypotheses are summarized in the following table.

Hypothesis	Number of Parameters	χ^2	Degrees of Freedom	*p*-value	$\hat{\rho}$
H_1	4	37.33	6	0.0000	1.0
H_2	5	1.93	5	0.8583	0.8986
H_3	8	36.21	2	0.0000	1.0
H_4	9	0.70	1	0.4018	0.8986

The hypotheses H_1 and H_3, which posit $\rho = 1$, can be rejected. Hypotheses H_2 and H_4 seem to be consistent with the available data. Since H_2 is obtained by adding four constraints to H_4, you can test H_2 versus H_4 by computing the differences of the chi-square statistics and their degrees of freedom, yielding a chi-square of 1.23 with 4 degrees of freedom, which is obviously not significant. So hypothesis H_2 is consistent with the available data.

The estimates of ρ for H_2 and H_4 are almost identical, about 0.90, indicating that the speeded and unspeeded tests are measuring almost the same latent variable, even though the hypotheses that stated they measured exactly the same latent variable are rejected.

A Combined Measurement-Structural Model with Reciprocal Influence and Correlated Residuals

To illustrate a more complex model, this example uses some well-known data from Haller and Butterworth (1960). Various models and analyses of these data are given by Duncan, Haller, and Portes (1968), Jöreskog and Sörbom (1988), and Loehlin (1987).

The study is concerned with the career aspirations of high school students and how these aspirations are affected by close friends. The data are collected from 442 seventeen-year-old boys in Michigan. There are 329 boys in the sample who named another boy in the sample as a best friend. The observations to be analyzed consist of the data from these 329 boys paired with the data from their best friends.

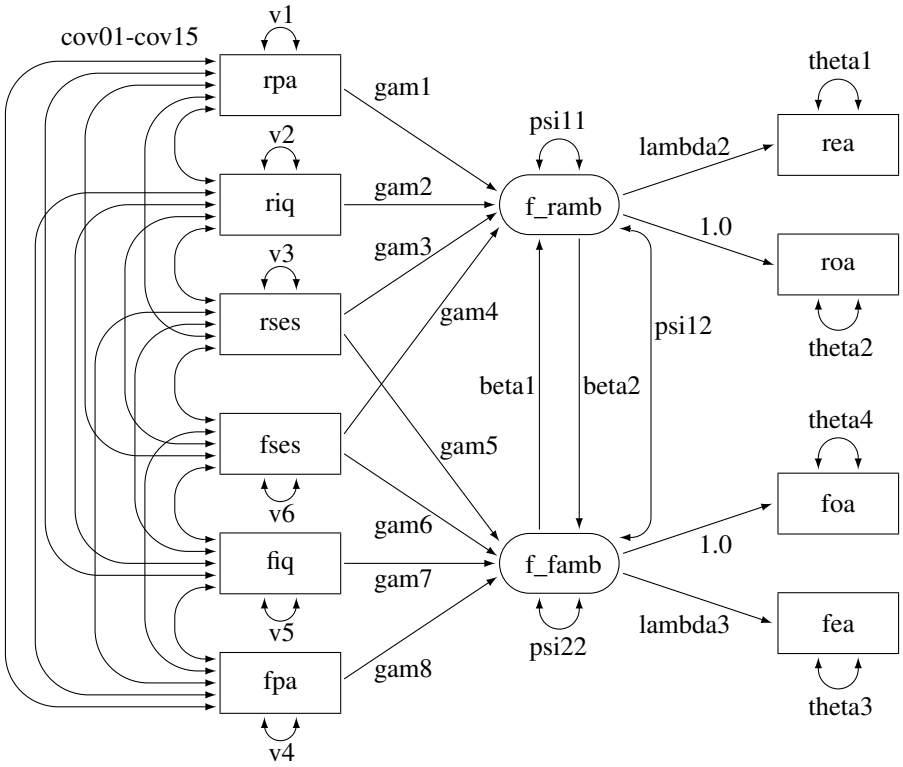
The method of data collection introduces two statistical problems. First, restricting the analysis to boys whose best friends are in the original sample causes the reduced sample to be biased. Second, since the data from a given boy might appear in two or more observations, the observations are not independent. Therefore, any statistical conclusions should be considered tentative. It is difficult to accurately assess the effects of the dependence of the observations on the analysis, but it could be argued on intuitive grounds that since each observation has data from two boys and since it seems likely that many of the boys will appear in the data set at least twice, the effective sample size might be as small as half of the reported 329 observations.

The correlation matrix, taken from Jöreskog and Sörbom (1988), is shown in the following DATA step:

```
title 'Peer Influences on Aspiration: Haller & Butterworth (1960)';
data aspire(type=corr);
   _type_='corr';
   input _name_ $ riq rpa rses roa rea fiq fpa fses foa fea;
   label riq='Respondent: Intelligence'
         rpa='Respondent: Parental Aspiration'
         rses='Respondent: Family SES'
         roa='Respondent: Occupational Aspiration'
         rea='Respondent: Educational Aspiration'
         fiq='Friend: Intelligence'
         fpa='Friend: Parental Aspiration'
         fses='Friend: Family SES'
         foa='Friend: Occupational Aspiration'
         fea='Friend: Educational Aspiration';
   datalines;
riq   1.      .      .      .      .      .      .      .      .      .
rpa   .1839  1.      .      .      .      .      .      .      .      .
rses  .2220  .0489  1.      .      .      .      .      .      .      .
roa   .4105  .2137  .3240  1.      .      .      .      .      .      .
rea   .4043  .2742  .4047  .6247  1.      .      .      .      .      .
fiq   .3355  .0782  .2302  .2995  .2863  1.      .      .      .      .
fpa   .1021  .1147  .0931  .0760  .0702  .2087   1.      .      .      .
fses  .1861  .0186  .2707  .2930  .2407  .2950  -.0438  1.      .      .
foa   .2598  .0839  .2786  .4216  .3275  .5007   .1988  .3607  1.      .
fea   .2903  .1124  .3054  .3269  .3669  .5191   .2784  .4105  .6404  1.
;
```

The model analyzed by Jöreskog and Sörbom (1988) is displayed in the path diagram in Figure 17.17.

Figure 17.17 Path Diagram: Career Aspiration – Jöreskog and Sörbom (1988)



Two latent variables, f_ramb and f_famb, represent the respondent's level of ambition and his best friend's level of ambition, respectively. The model states that the respondent's ambition is determined by his intelligence (riq) and socioeconomic status (rses), his perception of his parents' aspiration for him (rpa), and his friend's socioeconomic status (fses) and ambition (f_famb). It is assumed that his friend's intelligence (fiq) and parental aspiration (fpa) affect the respondent's ambition only indirectly through the friend's ambition (f_famb). Ambition is indexed by the manifest variables of occupational (roa) and educational aspiration (rea), which are assumed to have uncorrelated residuals. The path coefficient from ambition to occupational aspiration is set to 1.0 to determine the scale of the ambition latent variable.

This model can be analyzed with PROC TCALIS by using the PATH modeling language, as shown in the following statements:

```
proc tcalis corr data=aspire nobs=329;
   path
      /* measurement model for aspiration */
      rea <-  f_ramb    lambda2,
      roa <-  f_ramb    1.,
      foa <-  f_famb    1.,
      fea <-  f_famb    lambda3,
      /* structural model of influences */
      f_ramb <- rpa     gam1,
      f_ramb <- riq     gam2,
      f_ramb <- rses    gam3,
      f_ramb <- fses    gam4,
      f_famb <- rses    gam5,
      f_famb <- fses    gam6,
      f_famb <- fiq     gam7,
      f_famb <- fpa     gam8,
      f_ramb <- f_famb  beta1,
      f_famb <- f_ramb  beta2;
   pvar
      f_ramb = psi11,
      f_famb = psi22,
      rpa riq rses fpa fiq fses = v1-v6,
      rea roa fea foa = theta1-theta4;
   pcov
      f_ramb f_famb = psi12,
      rpa riq rses fpa fiq fses = 15 * cov__;
run;
```

In this specification, the names of the parameters correspond to those used by Jöreskog and Sörbom (1988). Since this TYPE=CORR data set does not contain an observation with _TYPE_='N' giving the sample size, it is necessary to specify the NOBS= option in the PROC TCALIS statement.

Specifying a name followed by double underscores is a quick way to generate unique parameter names. The double underscores are replaced with a unique number each time a new parameter name is generated. For example, in the COV statement, the specification

```
rpa riq rses fpa fiq fses = 15 * cov__;
```

is equivalent to

```
rpa riq rses fpa fiq fses = cov01-cov15;
```

In the PROC TCALIS statement, the CORR option is used to indicate that the correlation matrix is fitted by the model. Fitting correlation matrices by covariance structure modeling method is plagued with some statistical issues. For example, the chi-square statistic might not follow the theoretical distribution well, and the estimates of standard errors might not be accurate. Nonetheless, the correlation matrix is fitted here for illustration and comparison purposes.

The results from this analysis are displayed in Figure 17.18.

Figure 17.18 Career Aspiration Data: Fit Summary of Jöreskog and Sörbom (1988) Analysis 1

Fit Summary

Modeling Info	N Observations	329
	N Variables	10
	N Moments	55
	N Parameters	40
	N Active Constraints	0
	Independence Model Chi-Square	872.0008
	Independence Model Chi-Square DF	45
Absolute Index	Fit Function	0.0814
	Chi-Square	26.6972
	Chi-Square DF	15
	Pr > Chi-Square	0.0313
	Z-Test of Wilson & Hilferty	1.8625
	Hoelter Critical N	309
	Root Mean Square Residual (RMSR)	0.0202
	Standardized RMSR (SRMSR)	0.0202
	Goodness of Fit Index (GFI)	0.9844
Parsimony Index	Adjusted GFI (AGFI)	0.9428
	Parsimonious GFI	0.3281
	RMSEA Estimate	0.0488
	RMSEA Lower 90% Confidence Limit	0.0145
	RMSEA Upper 90% Confidence Limit	0.0783
	Probability of Close Fit	0.4876
	ECVI Estimate	0.3338
	ECVI Lower 90% Confidence Limit	0.3012
	ECVI Upper 90% Confidence Limit	0.3910
	Akaike Information Criterion	-3.3028
	Bozdogan CAIC	-75.2437
	Schwarz Bayesian Criterion	-60.2437
	McDonald Centrality	0.9824
Incremental Index	Bentler Comparative Fit Index	0.9859
	Bentler-Bonett NFI	0.9694
	Bentler-Bonett Non-normed Index	0.9576
	Bollen Normed Index Rho1	0.9082
	Bollen Non-normed Index Delta2	0.9864
	James et al. Parsimonious NFI	0.3231

Jöreskog and Sörbom (1988) present more detailed results from a second analysis in which two constraints are imposed:

- The coefficients connecting the latent ambition variables are equal (that is, beta1 = beta2).
- The covariance of the disturbances of the ambition variables is zero (that is, psi12 = 0).

This analysis can be performed by changing the names beta1 and beta2 to beta and omitting the line from the COV statement for psi12, as shown in the following statements:

```
proc tcalis corr data=aspire nobs=329;
   path
      /* measurement model for aspiration */
      rea <-  f_ramb    lambda2,
      roa <-  f_ramb    1.,
      foa <-  f_famb    1.,
      fea <-  f_famb    lambda3,
      /* structural model of influences */
      f_ramb <- rpa     gam1,
      f_ramb <- riq     gam2,
      f_ramb <- rses    gam3,
      f_ramb <- fses    gam4,
      f_famb <- rses    gam5,
      f_famb <- fses    gam6,
      f_famb <- fiq     gam7,
      f_famb <- fpa     gam8,
      f_ramb <- f_famb  beta,
      f_famb <- f_ramb  beta;
   pvar
      f_ramb = psi11,
      f_famb = psi22,
      rpa riq rses fpa fiq fses = v1-v6,
      rea roa fea foa = theta1-theta4;
   pcov
      rpa riq rses fpa fiq fses = 15 * cov__;
 run;
```

The fit summary is displayed in Figure 17.19, and the estimation results are displayed in Figure 17.20.

Figure 17.19 Career Aspiration Data: Fit Summary of Jöreskog and Sörbom (1988) Analysis 2

Fit Summary

Modeling Info	N Observations	329
	N Variables	10
	N Moments	55
	N Parameters	38
	N Active Constraints	0
	Independence Model Chi-Square	872.0008
	Independence Model Chi-Square DF	45
Absolute Index	Fit Function	0.0820
	Chi-Square	26.8987
	Chi-Square DF	17
	Pr > Chi-Square	0.0596
	Z-Test of Wilson & Hilferty	1.5599
	Hoelter Critical N	338
	Root Mean Square Residual (RMSR)	0.0203
	Standardized RMSR (SRMSR)	0.0203
	Goodness of Fit Index (GFI)	0.9843
Parsimony Index	Adjusted GFI (AGFI)	0.9492
	Parsimonious GFI	0.3718
	RMSEA Estimate	0.0421
	RMSEA Lower 90% Confidence Limit	.
	RMSEA Upper 90% Confidence Limit	0.0710
	Probability of Close Fit	0.6367
	ECVI Estimate	0.3218
	ECVI Lower 90% Confidence Limit	.
	ECVI Upper 90% Confidence Limit	0.3781
	Akaike Information Criterion	-7.1013
	Bozdogan CAIC	-88.6343
	Schwarz Bayesian Criterion	-71.6343
	McDonald Centrality	0.9851
Incremental Index	Bentler Comparative Fit Index	0.9880
	Bentler-Bonett NFI	0.9692
	Bentler-Bonett Non-normed Index	0.9683
	Bollen Normed Index Rho1	0.9183
	Bollen Non-normed Index Delta2	0.9884
	James et al. Parsimonious NFI	0.3661

Figure 17.20 Career Aspiration Data: Estimation Results of Jöreskog and Sörbom (1988) Analysis 2

PATH List

Path			Parameter	Estimate	Standard Error	t Value
rea	<-	f_ramb	lambda2	1.06097	0.08921	11.89233
roa	<-	f_ramb		1.00000		
foa	<-	f_famb		1.00000		
fea	<-	f_famb	lambda3	1.07359	0.08063	13.31498
f_ramb	<-	rpa	gam1	0.16367	0.03872	4.22740
f_ramb	<-	riq	gam2	0.25395	0.04186	6.06725
f_ramb	<-	rses	gam3	0.22115	0.04187	5.28219
f_ramb	<-	fses	gam4	0.07728	0.04149	1.86264
f_famb	<-	rses	gam5	0.06840	0.03868	1.76809
f_famb	<-	fses	gam6	0.21839	0.03948	5.53198
f_famb	<-	fiq	gam7	0.33063	0.04116	8.03314
f_famb	<-	fpa	gam8	0.15204	0.03636	4.18169
f_ramb	<-	f_famb	beta	0.18007	0.03912	4.60305
f_famb	<-	f_ramb	beta	0.18007	0.03912	4.60305

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	f_ramb	psi11	0.28113	0.04640	6.05867
	f_famb	psi22	0.22924	0.03889	5.89393
Exogenous	rpa	v1	1.00000	0.07809	12.80625
	riq	v2	1.00000	0.07809	12.80625
	rses	v3	1.00000	0.07809	12.80625
	fpa	v4	1.00000	0.07809	12.80625
	fiq	v5	1.00000	0.07809	12.80625
	fses	v6	1.00000	0.07809	12.80625
Error	rea	theta1	0.33764	0.05178	6.52039
	roa	theta2	0.41205	0.05103	8.07403
	fea	theta3	0.31337	0.04574	6.85165
	foa	theta4	0.40381	0.04608	8.76428

Figure 17.20 *continued*

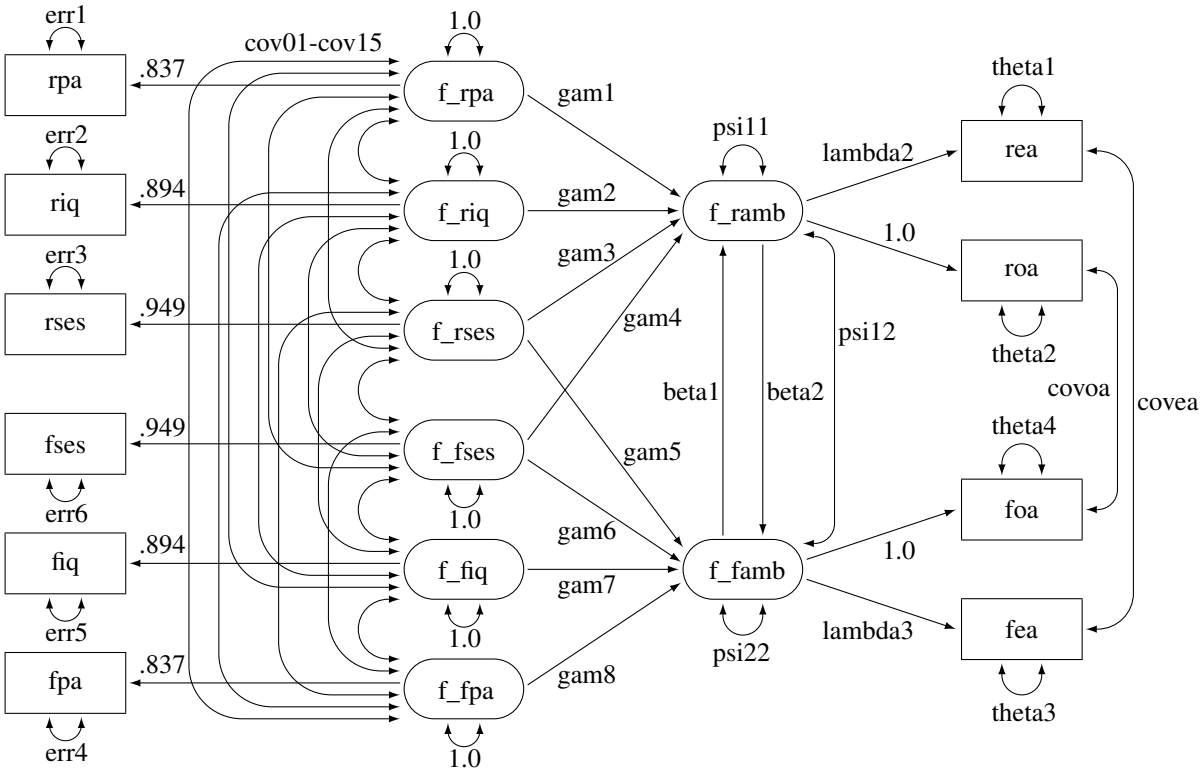
Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
rpa	riq	cov01	0.18390	0.05614	3.27564
rpa	rses	cov02	0.04890	0.05528	0.88456
riq	rses	cov03	0.22200	0.05656	3.92503
rpa	fpa	cov04	0.11470	0.05558	2.06377
riq	fpa	cov05	0.10210	0.05550	1.83955
rses	fpa	cov06	0.09310	0.05545	1.67885
rpa	fiq	cov07	0.07820	0.05538	1.41195
riq	fiq	cov08	0.33550	0.05824	5.76060
rses	fiq	cov09	0.23020	0.05666	4.06284
fpa	fiq	cov10	0.20870	0.05641	3.70000
rpa	fses	cov11	0.01860	0.05523	0.33680
riq	fses	cov12	0.18610	0.05616	3.31352
rses	fses	cov13	0.27070	0.05720	4.73226
fpa	fses	cov14	-0.04380	0.05527	-0.79249
fiq	fses	cov15	0.29500	0.05757	5.12435

The difference between the chi-square values for the two preceding models is $26.8987 - 26.6972 = 0.2015$ with 2 degrees of freedom, which is far from significant. This indicates that the restricted model (analysis 2) fits as well as the unrestricted model (analysis 1). However, the chi-square test of the restricted model against the alternative of a completely unrestricted covariance matrix yields a p-value of 0.0596, which indicates that the model might not be entirely satisfactory (p-values from these data are probably too small because of the dependence of the observations).

Loehlin (1987) points out that the models considered are unrealistic in at least two respects. First, the variables of parental aspiration, intelligence, and socioeconomic status are assumed to be measured without error. Loehlin adds uncorrelated measurement errors to the model and assumes, for illustrative purposes, that the reliabilities of these variables are known to be 0.7, 0.8, and 0.9, respectively. In practice, these reliabilities would need to be obtained from a separate study of the same or a very similar population. If these constraints are omitted, the model is not identified. However, constraining parameters to a constant in an analysis of a correlation matrix might make the chi-square goodness-of-fit test inaccurate, so there is more reason to be skeptical of the p-values. Second, the error terms for the respondent's aspiration are assumed to be uncorrelated with the corresponding terms for his friend. Loehlin introduces a correlation between the two educational aspiration error terms and between the two occupational aspiration error terms. These additions produce the path diagram for Loehlin's model 1 shown in Figure 17.21.

Figure 17.21 Path Diagram: Career Aspiration – Loehlin (1987)



In Figure 17.21, the observed variables rpa, riq, rses, fses, fiq, and fpa are all treated as measurements with errors. Their purified counterparts f_rpa, f_riq, f_rses, f_fses, f_fiq, and f_fpa are latent variables created in the model to represent measurements without errors. Path coefficients from these latent variables to the observed variables are fixed coefficients, indicating the square roots of the theoretical reliabilities in the model. These latent variables, rather than the observed counterparts, serve as predictors of the ambition variables f_ramb and f_famb. Correlated errors for the occupational aspiration variables and the educational aspiration variables are also shown in Figure 17.21. The error covariance for the educational aspiration variables rea and fea is indicated by the parameter covea, and the error covariance for the occupational aspiration variables roa and foa is indicated by the parameter covoa.

The statements for fitting this model by using the PATH modeling language are as follows:

```
proc tcalis corr data=aspire nobs=329;
   path
      /* measurement model for aspiration */
      rea  <-  f_ramb    lambda2,
      roa  <-  f_ramb    1.,
      foa  <-  f_famb    1.,
      fea  <-  f_famb    lambda3,

      /* measurement model for intelligence and environment */
      rpa  <-  f_rpa     0.837,
      riq  <-  f_riq     0.894,
      rses <-  f_rses    0.949,
      fses <-  f_fses    0.949,
      fiq  <-  f_fiq     0.894,
      fpa  <-  f_fpa     0.837,

      /* structural model of influences */
      f_ramb <- f_rpa     gam1,
      f_ramb <- f_riq     gam2,
      f_ramb <- f_rses    gam3,
      f_ramb <- f_fses    gam4,
      f_famb <- f_rses    gam5,
      f_famb <- f_fses    gam6,
      f_famb <- f_fiq     gam7,
      f_famb <- f_fpa     gam8,
      f_ramb <- f_famb    beta1,
      f_famb <- f_ramb    beta2;
   pvar
      f_ramb = psi11,
      f_famb = psi22,
      f_rpa f_riq f_rses f_fses f_fiq f_fpa = 6 * 1.0,
      rea roa fea foa                       = theta1-theta4,
      rpa riq rses fpa fiq fses             = err1-err6;
   pcov
      f_ramb f_famb   = psi12,
      rea  fea        = covea,
      roa  foa        = covoa,
      f_rpa f_riq f_rses f_fses f_fiq f_fpa = 15 * cov__;
run;
```

The fit summary is displayed in Figure 17.22, and the estimation results are displayed in Figure 17.23.

Figure 17.22 Career Aspiration Data: Fit Summary of Loehlin (1987) Model 1

Fit Summary

Modeling Info	N Observations	329
	N Variables	10
	N Moments	55
	N Parameters	42
	N Active Constraints	0
	Independence Model Chi-Square	872.0008
	Independence Model Chi-Square DF	45
Absolute Index	Fit Function	0.0366
	Chi-Square	12.0132
	Chi-Square DF	13
	Pr > Chi-Square	0.5266
	Z-Test of Wilson & Hilferty	-0.0679
	Hoelter Critical N	612
	Root Mean Square Residual (RMSR)	0.0149
	Standardized RMSR (SRMSR)	0.0149
	Goodness of Fit Index (GFI)	0.9927
Parsimony Index	Adjusted GFI (AGFI)	0.9692
	Parsimonious GFI	0.2868
	RMSEA Estimate	0.0000
	RMSEA Lower 90% Confidence Limit	.
	RMSEA Upper 90% Confidence Limit	0.0512
	Probability of Close Fit	0.9435
	ECVI Estimate	0.3016
	ECVI Lower 90% Confidence Limit	.
	ECVI Upper 90% Confidence Limit	0.3392
	Akaike Information Criterion	-13.9868
	Bozdogan CAIC	-76.3356
	Schwarz Bayesian Criterion	-63.3356
	McDonald Centrality	1.0015
Incremental Index	Bentler Comparative Fit Index	1.0000
	Bentler-Bonett NFI	0.9862
	Bentler-Bonett Non-normed Index	1.0041
	Bollen Normed Index Rho1	0.9523
	Bollen Non-normed Index Delta2	1.0011
	James et al. Parsimonious NFI	0.2849

Figure 17.23 Career Aspiration Data: Estimation Results of Loehlin (1987) Model 1

PATH List

Path			Parameter	Estimate	Standard Error	t Value
rea	<-	f_ramb	lambda2	1.08398	0.09417	11.51054
roa	<-	f_ramb		1.00000		
foa	<-	f_famb		1.00000		
fea	<-	f_famb	lambda3	1.11631	0.08627	12.93937
rpa	<-	f_rpa		0.83700		
riq	<-	f_riq		0.89400		
rses	<-	f_rses		0.94900		
fses	<-	f_fses		0.94900		
fiq	<-	f_fiq		0.89400		
fpa	<-	f_fpa		0.83700		
f_ramb	<-	f_rpa	gam1	0.18370	0.05044	3.64201
f_ramb	<-	f_riq	gam2	0.28004	0.06139	4.56183
f_ramb	<-	f_rses	gam3	0.22616	0.05223	4.33004
f_ramb	<-	f_fses	gam4	0.08698	0.05476	1.58836
f_famb	<-	f_rses	gam5	0.06327	0.05219	1.21240
f_famb	<-	f_fses	gam6	0.21539	0.05121	4.20600
f_famb	<-	f_fiq	gam7	0.35386	0.06741	5.24971
f_famb	<-	f_fpa	gam8	0.16877	0.04934	3.42051
f_ramb	<-	f_famb	beta1	0.11897	0.11396	1.04397
f_famb	<-	f_ramb	beta2	0.13022	0.12067	1.07921

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	f_ramb	psi11	0.25418	0.04469	5.68738
	f_famb	psi22	0.19698	0.03814	5.16533
Exogenous	f_rpa		1.00000		
	f_riq		1.00000		
	f_rses		1.00000		
	f_fses		1.00000		
	f_fiq		1.00000		
	f_fpa		1.00000		
Error	rea	theta1	0.32707	0.05452	5.99883
	roa	theta2	0.42307	0.05243	8.06948
	fea	theta3	0.28715	0.04804	5.97748
	foa	theta4	0.42240	0.04730	8.93103
	rpa	err1	0.29584	0.07774	3.80573
	riq	err2	0.20874	0.07832	2.66519
	rses	err3	0.09887	0.07803	1.26715
	fpa	err4	0.29987	0.07807	3.84089
	fiq	err5	0.19988	0.07674	2.60475
	fses	err6	0.10324	0.07824	1.31950

Figure 17.23 *continued*

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
f_rpa	f_riq	cov01	0.24677	0.07519	3.28204
f_rpa	f_rses	cov02	0.06184	0.06945	0.89034
f_riq	f_rses	cov03	0.26351	0.06687	3.94075
f_rpa	f_fses	cov04	0.02383	0.06952	0.34272
f_riq	f_fses	cov05	0.22135	0.06648	3.32976
f_rses	f_fses	cov06	0.30156	0.06359	4.74205
f_rpa	f_fiq	cov07	0.10853	0.07362	1.47419
f_riq	f_fiq	cov08	0.42476	0.07219	5.88373
f_rses	f_fiq	cov09	0.27250	0.06660	4.09143
f_fses	f_fiq	cov10	0.34922	0.06771	5.15755
f_rpa	f_fpa	cov11	0.15789	0.07873	2.00553
f_riq	f_fpa	cov12	0.13085	0.07418	1.76393
f_rses	f_fpa	cov13	0.11517	0.06978	1.65053
f_fses	f_fpa	cov14	-0.05623	0.06971	-0.80655
f_fiq	f_fpa	cov15	0.27867	0.07530	3.70083

Covariances Among Errors

Error of	Error of	Parameter	Estimate	Standard Error	t Value
f_ramb	f_famb	psi12	-0.00935	0.05010	-0.18669
rea	fea	covea	0.02308	0.03139	0.73543
roa	foa	covoa	0.11206	0.03258	3.43993

Since the p-value for the chi-square test is 0.5266, this model clearly cannot be rejected. However, Schwarz's Bayesian criterion for this model (SBC $= -63.3356$) is somewhat larger than for Jöreskog and Sörbom's (1988) analysis 2 (SBC $= -71.6343$), suggesting that a more parsimonious model would be desirable.

Since it is assumed that the same model applies to all the boys in the sample, the path diagram should be symmetric with respect to the respondent and his friend. In particular, the corresponding coefficients should be equal. By imposing equality constraints on the 15 pairs of corresponding coefficients, this example obtains Loehlin's (1987) model 2. The PATH model is as follows, where an OUTMODEL= data set is created to facilitate subsequent hypothesis tests:

```
proc tcalis corr data=aspire nobs=329 outmodel=model2;
   path
      /* measurement model for aspiration */
      rea  <-  f_ramb    lambda,              /* 1 ec! */
      roa  <-  f_ramb    1.,
      foa  <-  f_famb    1.,
      fea  <-  f_famb    lambda,

      /* measurement model for intelligence and environment */
      rpa  <-  f_rpa     0.837,
      riq  <-  f_riq     0.894,
      rses <-  f_rses    0.949,
      fses <-  f_fses    0.949,
      fiq  <-  f_fiq     0.894,
      fpa  <-  f_fpa     0.837,

      /* structural model of influences */
      f_ramb <- f_rpa     gam1,               /* 5 ec! */
      f_ramb <- f_riq     gam2,
      f_ramb <- f_rses    gam3,
      f_ramb <- f_fses    gam4,
      f_famb <- f_rses    gam4,
      f_famb <- f_fses    gam3,
      f_famb <- f_fiq     gam2,
      f_famb <- f_fpa     gam1,
      f_ramb <- f_famb    beta,
      f_famb <- f_ramb    beta;
   pvar
      f_ramb = psi,                           /* 1 ec! */
      f_famb = psi,
      f_rpa f_riq f_rses f_fpa f_fiq f_fses = 6 * 1.0,
      rea fea                 = 2 * thetaea,  /* 2 ec! */
      roa foa                 = 2 * thetaoa,
      rpa fpa                 = errpa1 errpa2,
      riq fiq                 = erriq1 erriq2,
      rses fses               = errses1 errses2;
   pcov
      f_ramb f_famb           = psi12,
      rea  fea                = covea,
      roa  foa                = covoa,
      f_rpa f_riq f_rses      = cov1-cov3,          /* 3 ec! */
      f_fpa f_fiq f_fses      = cov1-cov3,
      f_rpa f_riq f_rses * f_fpa f_fiq f_fses =     /* 3 ec! */
          cov4 cov5 cov6
          cov5 cov7 cov8
          cov6 cov8 cov9;
run;
```

The fit summary is displayed in Figure 17.24, and the estimation results are displayed in Figure 17.25.

Figure 17.24 Career Aspiration Data: Fit Summary of Loehlin (1987) Model 2

Fit Summary

Modeling Info	N Observations	329
	N Variables	10
	N Moments	55
	N Parameters	27
	N Active Constraints	0
	Independence Model Chi-Square	872.0008
	Independence Model Chi-Square DF	45
Absolute Index	Fit Function	0.0581
	Chi-Square	19.0697
	Chi-Square DF	28
	Pr > Chi-Square	0.8960
	Z-Test of Wilson & Hilferty	-1.2599
	Hoelter Critical N	713
	Root Mean Square Residual (RMSR)	0.0276
	Standardized RMSR (SRMSR)	0.0276
	Goodness of Fit Index (GFI)	0.9884
Parsimony Index	Adjusted GFI (AGFI)	0.9772
	Parsimonious GFI	0.6150
	RMSEA Estimate	0.0000
	RMSEA Lower 90% Confidence Limit	.
	RMSEA Upper 90% Confidence Limit	0.0194
	Probability of Close Fit	0.9996
	ECVI Estimate	0.2285
	ECVI Lower 90% Confidence Limit	.
	ECVI Upper 90% Confidence Limit	0.2664
	Akaike Information Criterion	-36.9303
	Bozdogan CAIC	-171.2200
	Schwarz Bayesian Criterion	-143.2200
	McDonald Centrality	1.0137
Incremental Index	Bentler Comparative Fit Index	1.0000
	Bentler-Bonett NFI	0.9781
	Bentler-Bonett Non-normed Index	1.0174
	Bollen Normed Index Rho1	0.9649
	Bollen Non-normed Index Delta2	1.0106
	James et al. Parsimonious NFI	0.6086

Figure 17.25 Career Aspiration Data: Estimation Results of Loehlin (1987) Model 2

PATH List

Path			Parameter	Estimate	Standard Error	t Value
rea	<-	f_ramb	lambda	1.10067	0.06842	16.08795
roa	<-	f_ramb		1.00000		
foa	<-	f_famb		1.00000		
fea	<-	f_famb	lambda	1.10067	0.06842	16.08795
rpa	<-	f_rpa		0.83700		
riq	<-	f_riq		0.89400		
rses	<-	f_rses		0.94900		
fses	<-	f_fses		0.94900		
fiq	<-	f_fiq		0.89400		
fpa	<-	f_fpa		0.83700		
f_ramb	<-	f_rpa	gam1	0.17585	0.03508	5.01299
f_ramb	<-	f_riq	gam2	0.32234	0.04702	6.85568
f_ramb	<-	f_rses	gam3	0.22273	0.03629	6.13725
f_ramb	<-	f_fses	gam4	0.07564	0.03750	2.01699
f_famb	<-	f_rses	gam4	0.07564	0.03750	2.01699
f_famb	<-	f_fses	gam3	0.22273	0.03629	6.13725
f_famb	<-	f_fiq	gam2	0.32234	0.04702	6.85568
f_famb	<-	f_fpa	gam1	0.17585	0.03508	5.01299
f_ramb	<-	f_famb	beta	0.11578	0.08390	1.38007
f_famb	<-	f_ramb	beta	0.11578	0.08390	1.38007

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	f_ramb	psi	0.22456	0.02971	7.55930
	f_famb	psi	0.22456	0.02971	7.55930
Exogenous	f_rpa		1.00000		
	f_riq		1.00000		
	f_rses		1.00000		
	f_fpa		1.00000		
	f_fiq		1.00000		
	f_fses		1.00000		
Error	rea	thetaea	0.30662	0.03726	8.22956
	fea	thetaea	0.30662	0.03726	8.22956
	roa	thetaoa	0.42295	0.03651	11.58311
	foa	thetaoa	0.42295	0.03651	11.58311
	rpa	errpa1	0.30758	0.07511	4.09498
	fpa	errpa2	0.28834	0.07369	3.91289
	riq	erriq1	0.26656	0.07389	3.60730
	fiq	erriq2	0.15573	0.06700	2.32445
	rses	errses1	0.11467	0.07267	1.57800
	fses	errses2	0.08814	0.07089	1.24333

Figure 17.25 *continued*

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
f_rpa	f_riq	cov1	0.26470	0.05442	4.86370
f_rpa	f_rses	cov2	0.00177	0.04996	0.03533
f_riq	f_rses	cov3	0.31129	0.05057	6.15553
f_fpa	f_fiq	cov1	0.26470	0.05442	4.86370
f_fpa	f_fses	cov2	0.00177	0.04996	0.03533
f_fiq	f_fses	cov3	0.31129	0.05057	6.15553
f_rpa	f_fpa	cov4	0.15784	0.07872	2.00521
f_rpa	f_fiq	cov5	0.11837	0.05447	2.17325
f_rpa	f_fses	cov6	0.06910	0.04996	1.38303
f_riq	f_fpa	cov5	0.11837	0.05447	2.17325
f_riq	f_fiq	cov7	0.43061	0.07258	5.93255
f_riq	f_fses	cov8	0.24967	0.05060	4.93420
f_rses	f_fpa	cov6	0.06910	0.04996	1.38303
f_rses	f_fiq	cov8	0.24967	0.05060	4.93420
f_rses	f_fses	cov9	0.30190	0.06362	4.74578

Covariances Among Errors

Error of	Error of	Parameter	Estimate	Standard Error	t Value
f_ramb	f_famb	psi12	-0.00344	0.04931	-0.06981
rea	fea	covea	0.02160	0.03144	0.68686
roa	foa	covoa	0.11208	0.03257	3.44076

The test of Loehlin's (1987) model 2 against model 1 yields a chi-square of $19.0697 - 12.0132 = 7.0565$ with 15 degrees of freedom, which is clearly not significant. This indicates the restricted model 2 fits at least as well as model 1. Schwarz's Bayesian criterion (SBC) is also much lower for model 2 (-143.2200) than for model 1 (-63.3356). Hence, model 2 seems preferable on both substantive and statistical grounds.

A question of substantive interest is whether the friend's socioeconomic status (SES) has a significant direct influence on a boy's ambition. This can be addressed by omitting the paths from f_fses to f_ramb and from f_rses to f_famb designated by the parameter name gam4, yielding Loehlin's (1987) model 3:

```
title2 'Loehlin (1987) analysis: Model 3';
data model3(type=calismdl);
   set model2;
   if _name_='gam4' then
      do;
         _name_=' ';
         _estim_=0;
      end;
run;
proc tcalis corr data=aspire nobs=329 inmodel=model3;
run;
```

The fit summary is displayed in Figure 17.26.

Figure 17.26 Career Aspiration Data: Fit Summary of Loehlin (1987) Model 3

Fit Summary

Modeling Info	N Observations	329
	N Variables	10
	N Moments	55
	N Parameters	26
	N Active Constraints	0
	Independence Model Chi-Square	872.0008
	Independence Model Chi-Square DF	45
Absolute Index	Fit Function	0.0702
	Chi-Square	23.0365
	Chi-Square DF	29
	Pr > Chi-Square	0.7749
	Z-Test of Wilson & Hilferty	-0.7563
	Hoelter Critical N	607
	Root Mean Square Residual (RMSR)	0.0304
	Standardized RMSR (SRMSR)	0.0304
	Goodness of Fit Index (GFI)	0.9858
Parsimony Index	Adjusted GFI (AGFI)	0.9731
	Parsimonious GFI	0.6353
	RMSEA Estimate	0.0000
	RMSEA Lower 90% Confidence Limit	.
	RMSEA Upper 90% Confidence Limit	0.0295
	Probability of Close Fit	0.9984
	ECVI Estimate	0.2343
	ECVI Lower 90% Confidence Limit	.
	ECVI Upper 90% Confidence Limit	0.2780
	Akaike Information Criterion	-34.9635
	Bozdogan CAIC	-174.0492
	Schwarz Bayesian Criterion	-145.0492
	McDonald Centrality	1.0091
Incremental Index	Bentler Comparative Fit Index	1.0000
	Bentler-Bonett NFI	0.9736
	Bentler-Bonett Non-normed Index	1.0112
	Bollen Normed Index Rho1	0.9590
	Bollen Non-normed Index Delta2	1.0071
	James et al. Parsimonious NFI	0.6274

The chi-square value for testing model 3 versus model 2 is $23.0365 - 19.0697 = 3.9668$ with 1 degree of freedom and a p-value of 0.0464. Although the parameter is of marginal significance, the estimate in model 2 (0.0756) is small compared to the other coefficients, and SBC indicates that model 3 is preferable to model 2.

Another important question is whether the reciprocal influences between the respondent's and friend's ambitions are needed in the model. To test whether these paths are zero, set the parameter beta for the paths linking f_ramb and f_famb to zero to obtain Loehlin's (1987) model 4:

```
title2 'Loehlin (1987) analysis: Model 4';
data model4(type=calismdl);
   set model2;
   if _name_='beta' then
      do;
         _name_=' ';
         _estim_=0;
      end;
run;

proc tcalis corr data=aspire nobs=329 inmodel=model4;
run;
```

The fit summary is displayed in Figure 17.27, and the estimation results are displayed in Figure 17.28.

Figure 17.27 Career Aspiration Data: Fit Summary of Loehlin (1987) Model 4

Fit Summary

Modeling Info	N Observations	329
	N Variables	10
	N Moments	55
	N Parameters	26
	N Active Constraints	0
	Independence Model Chi-Square	872.0008
	Independence Model Chi-Square DF	45
Absolute Index	Fit Function	0.0640
	Chi-Square	20.9981
	Chi-Square DF	29
	Pr > Chi-Square	0.8592
	Z-Test of Wilson & Hilferty	-1.0780
	Hoelter Critical N	666
	Root Mean Square Residual (RMSR)	0.0304
	Standardized RMSR (SRMSR)	0.0304
	Goodness of Fit Index (GFI)	0.9873
Parsimony Index	Adjusted GFI (AGFI)	0.9760
	Parsimonious GFI	0.6363
	RMSEA Estimate	0.0000
	RMSEA Lower 90% Confidence Limit	.
	RMSEA Upper 90% Confidence Limit	0.0234
	Probability of Close Fit	0.9994
	ECVI Estimate	0.2281
	ECVI Lower 90% Confidence Limit	.
	ECVI Upper 90% Confidence Limit	0.2685
	Akaike Information Criterion	-37.0019
	Bozdogan CAIC	-176.0876
	Schwarz Bayesian Criterion	-147.0876
	McDonald Centrality	1.0122
Incremental Index	Bentler Comparative Fit Index	1.0000
	Bentler-Bonett NFI	0.9759
	Bentler-Bonett Non-normed Index	1.0150
	Bollen Normed Index Rho1	0.9626
	Bollen Non-normed Index Delta2	1.0095
	James et al. Parsimonious NFI	0.6289

Figure 17.28 Career Aspiration Data: Estimation Results of Loehlin (1987) Model 4

PATH List

Path			Parameter	Estimate	Standard Error	t Value
rea	<-	f_ramb	lambda	1.10505	0.06804	16.24157
roa	<-	f_ramb		1.00000		
foa	<-	f_famb		1.00000		
fea	<-	f_famb	lambda	1.10505	0.06804	16.24157
rpa	<-	f_rpa		0.83700		
riq	<-	f_riq		0.89400		
rses	<-	f_rses		0.94900		
fses	<-	f_fses		0.94900		
fiq	<-	f_fiq		0.89400		
fpa	<-	f_fpa		0.83700		
f_ramb	<-	f_rpa	gam1	0.17760	0.03610	4.91945
f_ramb	<-	f_riq	gam2	0.34856	0.04625	7.53618
f_ramb	<-	f_rses	gam3	0.23834	0.03549	6.71576
f_ramb	<-	f_fses	gam4	0.10810	0.02992	3.61340
f_famb	<-	f_rses	gam4	0.10810	0.02992	3.61340
f_famb	<-	f_fses	gam3	0.23834	0.03549	6.71576
f_famb	<-	f_fiq	gam2	0.34856	0.04625	7.53618
f_famb	<-	f_fpa	gam1	0.17760	0.03610	4.91945
f_ramb	<-	f_famb		0		
f_famb	<-	f_ramb		0		

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	f_ramb	psi	0.22738	0.03140	7.24263
	f_famb	psi	0.22738	0.03140	7.24263
Exogenous	f_rpa		1.00000		
	f_riq		1.00000		
	f_rses		1.00000		
	f_fpa		1.00000		
	f_fiq		1.00000		
	f_fses		1.00000		
Error	rea	thetaea	0.30502	0.03728	8.18091
	fea	thetaea	0.30502	0.03728	8.18091
	roa	thetaoa	0.42429	0.03645	11.64071
	foa	thetaoa	0.42429	0.03645	11.64071
	rpa	errpa1	0.31354	0.07543	4.15664
	fpa	errpa2	0.29051	0.07374	3.93945
	riq	erriq1	0.29611	0.07299	4.05703
	fiq	erriq2	0.18181	0.06611	2.75034
	rses	errses1	0.12320	0.07273	1.69400
	fses	errses2	0.09873	0.07109	1.38881

Figure 17.28 *continued*

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
f_rpa	f_riq	cov1	0.27241	0.05520	4.93523
f_rpa	f_rses	cov2	0.00476	0.05032	0.09455
f_riq	f_rses	cov3	0.32463	0.05089	6.37870
f_fpa	f_fiq	cov1	0.27241	0.05520	4.93523
f_fpa	f_fses	cov2	0.00476	0.05032	0.09455
f_fiq	f_fses	cov3	0.32463	0.05089	6.37870
f_rpa	f_fpa	cov4	0.16949	0.07863	2.15559
f_rpa	f_fiq	cov5	0.13539	0.05407	2.50384
f_rpa	f_fses	cov6	0.07362	0.05027	1.46453
f_riq	f_fpa	cov5	0.13539	0.05407	2.50384
f_riq	f_fiq	cov7	0.46893	0.06980	6.71822
f_riq	f_fses	cov8	0.26289	0.05093	5.16164
f_rses	f_fpa	cov6	0.07362	0.05027	1.46453
f_rses	f_fiq	cov8	0.26289	0.05093	5.16164
f_rses	f_fses	cov9	0.30880	0.06409	4.81849

Covariances Among Errors

Error of	Error of	Parameter	Estimate	Standard Error	t Value
f_ramb	f_famb	psi12	0.05479	0.02699	2.03009
rea	fea	covea	0.02127	0.03150	0.67534
roa	foa	covoa	0.11245	0.03258	3.45136

The chi-square value for testing model 4 versus model 2 is $20.9981 - 19.0697 = 1.9284$ with 1 degree of freedom and a p-value of 0.1649. Hence, there is little evidence of reciprocal influence.

Loehlin's (1987) model 2 has not only the direct paths connecting the latent ambition variables f_ramb and f_famb but also a covariance between the disturbance terms d_ramb and d_famb to allow for other variables omitted from the model that might jointly influence the respondent and his friend. To test the hypothesis that this covariance is zero, set the parameter psi12 to zero, yielding Loehlin's (1987) model 5:

```
title2 'Loehlin (1987) analysis: Model 5';
data model5(type=calismdl);
   set model2;
   if _name_='psi12' then
      do;
         _name_=' ';
         _estim_=0;
      end;
run;

proc tcalis corr data=aspire nobs=329 inmodel=model5;
run;
```

The fit summary is displayed in Figure 17.29, and the estimation results are displayed in Figure 17.30.

Figure 17.29 Career Aspiration Data: Fit Summary of Loehlin (1987) Model 5

Fit Summary

Modeling Info	N Observations	329
	N Variables	10
	N Moments	55
	N Parameters	26
	N Active Constraints	0
	Independence Model Chi-Square	872.0008
	Independence Model Chi-Square DF	45
Absolute Index	Fit Function	0.0582
	Chi-Square	19.0745
	Chi-Square DF	29
	Pr > Chi-Square	0.9194
	Z-Test of Wilson & Hilferty	-1.4014
	Hoelter Critical N	733
	Root Mean Square Residual (RMSR)	0.0276
	Standardized RMSR (SRMSR)	0.0276
	Goodness of Fit Index (GFI)	0.9884
Parsimony Index	Adjusted GFI (AGFI)	0.9780
	Parsimonious GFI	0.6370
	RMSEA Estimate	0.0000
	RMSEA Lower 90% Confidence Limit	.
	RMSEA Upper 90% Confidence Limit	0.0152
	Probability of Close Fit	0.9998
	ECVI Estimate	0.2222
	ECVI Lower 90% Confidence Limit	.
	ECVI Upper 90% Confidence Limit	0.2592
	Akaike Information Criterion	-38.9255
	Bozdogan CAIC	-178.0111
	Schwarz Bayesian Criterion	-149.0111
	McDonald Centrality	1.0152
Incremental Index	Bentler Comparative Fit Index	1.0000
	Bentler-Bonett NFI	0.9781
	Bentler-Bonett Non-normed Index	1.0186
	Bollen Normed Index Rho1	0.9661
	Bollen Non-normed Index Delta2	1.0118
	James et al. Parsimonious NFI	0.6303

Figure 17.30 Career Aspiration Data: Estimation Results of Loehlin (1987) Model 5

PATH List

Path			Parameter	Estimate	Standard Error	t Value
rea	<-	f_ramb	lambda	1.10086	0.06836	16.10408
roa	<-	f_ramb		1.00000		
foa	<-	f_famb		1.00000		
fea	<-	f_famb	lambda	1.10086	0.06836	16.10408
rpa	<-	f_rpa		0.83700		
riq	<-	f_riq		0.89400		
rses	<-	f_rses		0.94900		
fses	<-	f_fses		0.94900		
fiq	<-	f_fiq		0.89400		
fpa	<-	f_fpa		0.83700		
f_ramb	<-	f_rpa	gam1	0.17618	0.03502	5.03081
f_ramb	<-	f_riq	gam2	0.32351	0.04346	7.44351
f_ramb	<-	f_rses	gam3	0.22334	0.03533	6.32150
f_ramb	<-	f_fses	gam4	0.07698	0.03225	2.38702
f_famb	<-	f_rses	gam4	0.07698	0.03225	2.38702
f_famb	<-	f_fses	gam3	0.22334	0.03533	6.32150
f_famb	<-	f_fiq	gam2	0.32351	0.04346	7.44351
f_famb	<-	f_fpa	gam1	0.17618	0.03502	5.03081
f_ramb	<-	f_famb	beta	0.11074	0.04283	2.58539
f_famb	<-	f_ramb	beta	0.11074	0.04283	2.58539

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	f_ramb	psi	0.22453	0.02973	7.55201
	f_famb	psi	0.22453	0.02973	7.55201
Exogenous	f_rpa		1.00000		
	f_riq		1.00000		
	f_rses		1.00000		
	f_fpa		1.00000		
	f_fiq		1.00000		
	f_fses		1.00000		
Error	rea	thetaea	0.30645	0.03721	8.23647
	fea	thetaea	0.30645	0.03721	8.23647
	roa	thetaoa	0.42304	0.03650	11.58877
	foa	thetaoa	0.42304	0.03650	11.58877
	rpa	errpa1	0.30781	0.07510	4.09880
	fpa	errpa2	0.28837	0.07366	3.91467
	riq	erriq1	0.26748	0.07295	3.66672
	fiq	erriq2	0.15653	0.06614	2.36682
	rses	errses1	0.11477	0.07265	1.57975
	fses	errses2	0.08832	0.07088	1.24608

Figure 17.30 *continued*

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
f_rpa	f_riq	cov1	0.26494	0.05436	4.87395
f_rpa	f_rses	cov2	0.00185	0.04995	0.03696
f_riq	f_rses	cov3	0.31164	0.05039	6.18460
f_fpa	f_fiq	cov1	0.26494	0.05436	4.87395
f_fpa	f_fses	cov2	0.00185	0.04995	0.03696
f_fiq	f_fses	cov3	0.31164	0.05039	6.18460
f_rpa	f_fpa	cov4	0.15828	0.07846	2.01729
f_rpa	f_fiq	cov5	0.11895	0.05383	2.20978
f_rpa	f_fses	cov6	0.06924	0.04993	1.38664
f_riq	f_fpa	cov5	0.11895	0.05383	2.20978
f_riq	f_fiq	cov7	0.43180	0.07084	6.09540
f_riq	f_fses	cov8	0.25004	0.05039	4.96207
f_rses	f_fpa	cov6	0.06924	0.04993	1.38664
f_rses	f_fiq	cov8	0.25004	0.05039	4.96207
f_rses	f_fses	cov9	0.30203	0.06360	4.74852

Covariances Among Errors

Error of	Error of	Parameter	Estimate	Standard Error	t Value
f_ramb	f_famb		0		
rea	fea	covea	0.02120	0.03094	0.68516
roa	foa	covoa	0.11197	0.03254	3.44068

The chi-square value for testing model 5 versus model 2 is $19.0745 - 19.0697 = 0.0048$ with 1 degree of freedom. This test statistic is insignificant. Omitting the covariance between the disturbance terms, therefore, causes hardly any deterioration in the fit of the model.

These data fail to provide evidence of direct reciprocal influence between the respondent's and friend's ambitions or of a covariance between the disturbance terms when these hypotheses are considered separately. Notice, however, that the covariance psi12 between the disturbance terms increases from -0.003344 for model 2 to 0.05479 for model 4. Before you conclude that all of these paths can be omitted from the model, it is important to test both hypotheses together by setting both beta and psi12 to zero as in Loehlin's (1987) model 7:

```
title2 'Loehlin (1987) analysis: Model 7';
data model7(type=calismdl);
   set model2;
   if _name_='psi12'|_name_='beta' then
      do;
         _name_=' ';
         _estim_=0;
      end;
run;

proc tcalis corr data=aspire nobs=329 inmodel=model7;
run;
```

The fit summary is displayed in Figure 17.31, and the estimation results are displayed in Figure 17.32.

Figure 17.31 Career Aspiration Data: Fit Summary of Loehlin (1987) Model 7

Fit Summary

Modeling Info	N Observations	329
	N Variables	10
	N Moments	55
	N Parameters	25
	N Active Constraints	0
	Independence Model Chi-Square	872.0008
	Independence Model Chi-Square DF	45
Absolute Index	Fit Function	0.0773
	Chi-Square	25.3466
	Chi-Square DF	30
	Pr > Chi-Square	0.7080
	Z-Test of Wilson & Hilferty	-0.5487
	Hoelter Critical N	568
	Root Mean Square Residual (RMSR)	0.0363
	Standardized RMSR (SRMSR)	0.0363
	Goodness of Fit Index (GFI)	0.9846
Parsimony Index	Adjusted GFI (AGFI)	0.9718
	Parsimonious GFI	0.6564
	RMSEA Estimate	0.0000
	RMSEA Lower 90% Confidence Limit	.
	RMSEA Upper 90% Confidence Limit	0.0326
	Probability of Close Fit	0.9975
	ECVI Estimate	0.2350
	ECVI Lower 90% Confidence Limit	.
	ECVI Upper 90% Confidence Limit	0.2815
	Akaike Information Criterion	-34.6534
	Bozdogan CAIC	-178.5351
	Schwarz Bayesian Criterion	-148.5351
	McDonald Centrality	1.0071
Incremental Index	Bentler Comparative Fit Index	1.0000
	Bentler-Bonett NFI	0.9709
	Bentler-Bonett Non-normed Index	1.0084
	Bollen Normed Index Rho1	0.9564
	Bollen Non-normed Index Delta2	1.0055
	James et al. Parsimonious NFI	0.6473

Figure 17.32 Career Aspiration Data: Estimation Results of Loehlin (1987) Model 7

PATH List

Path			Parameter	Estimate	Standard Error	t Value
rea	<-	f_ramb	lambda	1.10371	0.06784	16.27015
roa	<-	f_ramb		1.00000		
foa	<-	f_famb		1.00000		
fea	<-	f_famb	lambda	1.10371	0.06784	16.27015
rpa	<-	f_rpa		0.83700		
riq	<-	f_riq		0.89400		
rses	<-	f_rses		0.94900		
fses	<-	f_fses		0.94900		
fiq	<-	f_fiq		0.89400		
fpa	<-	f_fpa		0.83700		
f_ramb	<-	f_rpa	gam1	0.17653	0.03604	4.89810
f_ramb	<-	f_riq	gam2	0.35727	0.04609	7.75204
f_ramb	<-	f_rses	gam3	0.24187	0.03628	6.66710
f_ramb	<-	f_fses	gam4	0.11087	0.03056	3.62795
f_famb	<-	f_rses	gam4	0.11087	0.03056	3.62795
f_famb	<-	f_fses	gam3	0.24187	0.03628	6.66710
f_famb	<-	f_fiq	gam2	0.35727	0.04609	7.75204
f_famb	<-	f_fpa	gam1	0.17653	0.03604	4.89810
f_ramb	<-	f_famb		0		
f_famb	<-	f_ramb		0		

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	f_ramb	psi	0.21011	0.02940	7.14704
	f_famb	psi	0.21011	0.02940	7.14704
Exogenous	f_rpa		1.00000		
	f_riq		1.00000		
	f_rses		1.00000		
	f_fpa		1.00000		
	f_fiq		1.00000		
	f_fses		1.00000		
Error	rea	thetaea	0.31633	0.03648	8.67106
	fea	thetaea	0.31633	0.03648	8.67106
	roa	thetaoa	0.42656	0.03610	11.81508
	foa	thetaoa	0.42656	0.03610	11.81508
	rpa	errpa1	0.31329	0.07538	4.15589
	fpa	errpa2	0.29286	0.07389	3.96366
	riq	erriq1	0.30776	0.07307	4.21157
	fiq	erriq2	0.19193	0.06613	2.90250
	rses	errses1	0.14303	0.07313	1.95574
	fses	errses2	0.11804	0.07147	1.65171

Figure 17.32 *continued*

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
f_rpa	f_riq	cov1	0.27533	0.05552	4.95900
f_rpa	f_rses	cov2	0.00611	0.05085	0.12020
f_riq	f_rses	cov3	0.33510	0.05150	6.50648
f_fpa	f_fiq	cov1	0.27533	0.05552	4.95900
f_fpa	f_fses	cov2	0.00611	0.05085	0.12020
f_fiq	f_fses	cov3	0.33510	0.05150	6.50648
f_rpa	f_fpa	cov4	0.17099	0.07872	2.17210
f_rpa	f_fiq	cov5	0.13859	0.05431	2.55174
f_rpa	f_fses	cov6	0.07563	0.05077	1.48956
f_riq	f_fpa	cov5	0.13859	0.05431	2.55174
f_riq	f_fiq	cov7	0.48105	0.06993	6.87858
f_riq	f_fses	cov8	0.27235	0.05157	5.28154
f_rses	f_fpa	cov6	0.07563	0.05077	1.48956
f_rses	f_fiq	cov8	0.27235	0.05157	5.28154
f_rses	f_fses	cov9	0.32046	0.06517	4.91719

Covariances Among Errors

Error of	Error of	Parameter	Estimate	Standard Error	t Value
f_ramb	f_famb		0		
rea	fea	covea	0.04535	0.02918	1.55444
roa	foa	covoa	0.12085	0.03214	3.75976

When model 7 is tested against models 2, 4, and 5, the *p*-values are respectively 0.0433, 0.0370, and 0.0123, indicating that the combined effect of the reciprocal influence and the covariance of the disturbance terms is statistically significant. Thus, the hypothesis tests indicate that it is acceptable to omit either the reciprocal influences or the covariance of the disturbances, but not both.

It is also of interest to test the covariances between the error terms for educational (covea) and occupational aspiration (covoa), since these terms are omitted from Jöreskog and Sörbom's (1988) models. Constraining covea and covoa to zero produces Loehlin's (1987) model 6:

```
title2 'Loehlin (1987) analysis: Model 6';
data model6(type=calismdl);
   set model2;
   if _name_='covea'|_name_='covoa' then
      do;
         _name_=' ';
         _estim_=0;
      end;
run;

proc tcalis corr data=aspire nobs=329 inmodel=model6;
run;
```

The fit summary is displayed in Figure 17.33.

Figure 17.33 Career Aspiration Data: Loehlin (1987) Model 6

Fit Summary

Modeling Info	N Observations	329
	N Variables	10
	N Moments	55
	N Parameters	25
	N Active Constraints	0
	Independence Model Chi-Square	872.0008
	Independence Model Chi-Square DF	45
Absolute Index	Fit Function	0.1020
	Chi-Square	33.4475
	Chi-Square DF	30
	Pr > Chi-Square	0.3035
	Z-Test of Wilson & Hilferty	0.5151
	Hoelter Critical N	431
	Root Mean Square Residual (RMSR)	0.0306
	Standardized RMSR (SRMSR)	0.0306
	Goodness of Fit Index (GFI)	0.9802
Parsimony Index	Adjusted GFI (AGFI)	0.9638
	Parsimonious GFI	0.6535
	RMSEA Estimate	0.0187
	RMSEA Lower 90% Confidence Limit	.
	RMSEA Upper 90% Confidence Limit	0.0471
	Probability of Close Fit	0.9686
	ECVI Estimate	0.2597
	ECVI Lower 90% Confidence Limit	.
	ECVI Upper 90% Confidence Limit	0.3164
	Akaike Information Criterion	-26.5525
	Bozdogan CAIC	-170.4342
	Schwarz Bayesian Criterion	-140.4342
	McDonald Centrality	0.9948
Incremental Index	Bentler Comparative Fit Index	0.9958
	Bentler-Bonett NFI	0.9616
	Bentler-Bonett Non-normed Index	0.9937
	Bollen Normed Index Rho1	0.9425
	Bollen Non-normed Index Delta2	0.9959
	James et al. Parsimonious NFI	0.6411

The chi-square value for testing model 6 versus model 2 is $33.4476 - 19.0697 = 14.3779$ with 2 degrees of freedom and a p-value of 0.0008, indicating that there is considerable evidence of correlation between the error terms.

The following table summarizes the results from Loehlin's (1987) seven models.

Model	χ^2	df	*p*-value	SBC
1. Full model	12.0132	13	0.5266	−63.3356
2. Equality constraints	19.0697	28	0.8960	−143.2200
3. No SES path	23.0365	29	0.7749	−145.0492
4. No reciprocal influence	20.9981	29	0.8592	−147.0876
5. No disturbance correlation	19.0745	29	0.9194	−149.0111
6. No error correlation	33.4475	30	0.3035	−140.4342
7. Constraints from both 4 and 5	25.3466	30	0.7080	−148.5351

For comparing models, you can use a DATA step to compute the differences of the chi-square statistics and *p*-values:

```
data _null_;
   array achisq[7] _temporary_
      (12.0132 19.0697 23.0365 20.9981 19.0745 33.4475 25.3466);
   array adf[7] _temporary_
      (13 28 29 29 29 30 30);
   retain indent 16;
   file print;
   input ho ha @@;
   chisq = achisq[ho] - achisq[ha];
   df = adf[ho] - adf[ha];
   p = 1 - probchi( chisq, df);
   if _n_ = 1 then put
      / +indent 'model comparison   chi**2   df  p-value'
      / +indent '-------------------------------------';
   put +indent +3 ho ' versus ' ha @18 +indent chisq 8.4 df 5. p 9.4;
datalines;
2 1    3 2    4 2    5 2    7 2    7 4    7 5    6 2
;
```

The DATA step displays the table in Figure 17.34.

Figure 17.34 Career Aspiration Data: Model Comparisons

```
          Peer Influences on Aspiration: Haller & Butterworth (1960)
                       Loehlin (1987) analysis: Model 6

               model comparison   chi**2   df  p-value
               -------------------------------------
                  2  versus 1     7.0565   15   0.9561
                  3  versus 2     3.9668    1   0.0464
                  4  versus 2     1.9284    1   0.1649
                  5  versus 2     0.0048    1   0.9448
                  7  versus 2     6.2769    2   0.0433
                  7  versus 4     4.3485    1   0.0370
                  7  versus 5     6.2721    1   0.0123
                  6  versus 2    14.3778    2   0.0008
```

Although none of the seven models can be rejected when tested against the alternative of an unrestricted covariance matrix, the model comparisons make it clear that there are important differences

among the models. Schwarz's Bayesian criterion indicates model 5 as the model of choice. The constraints added to model 5 in model 7 can be rejected (p=0.0123), while model 5 cannot be rejected when tested against the less constrained model 2 (p=0.9448). Hence, among the small number of models considered, model 5 has strong statistical support. However, as Loehlin (1987, p. 106) points out, many other models for these data could be constructed. Further analysis should consider, in addition to simple modifications of the models, the possibility that more than one friend could influence a boy's aspirations, and that a boy's ambition might have some effect on his choice of friends. Pursuing such theories would be statistically challenging.

References

Akaike, H. (1987), "Factor Analysis and AIC," *Psychometrika*, 52, 317–332.

Bentler, P. M. and Bonett, D. G. (1980), "Significance Tests and Goodness of Fit in the Analysis of Covariance Structures," *Psychological Bulletin*, 88, 588–606.

Bollen, K. A. (1986), "Sample Size and Bentler and Bonett's Nonnormed Fit Index," *Psychometrika*, 51, 375–377.

Bollen, K. A. (1989), *Structural Equations with Latent Variables*, New York: John Wiley & Sons.

Browne, M. W. and Cudeck, R. (1993), "Alternative Ways of Assessing Model Fit," in *Testing Structural Equation Models*, ed. K. A. Bollen and S. Long, Newbury Park, CA: Sage Publications.

Duncan, O. D., Haller, A. O., and Portes, A. (1968), "Peer Influences on Aspirations: A Reinterpretation," *American Journal of Sociology*, 74, 119–137.

Fuller, W. A. (1987), *Measurement Error Models*, New York: John Wiley & Sons.

Haller, A. O., and Butterworth, C. E. (1960), "Peer Influences on Levels of Occupational and Educational Aspiration," *Social Forces*, 38, 289–295.

Hampel F. R., Ronchetti E. M., Rousseeuw P. J., and Stahel W. A. (1986), *Robust Statistics*, New York: John Wiley & Sons.

Huber, P. J. (1981), *Robust Statistics*, New York: John Wiley & Sons.

James, L. R., Mulaik, S. A., and Brett, J. M. (1982), *Causal Analysis*, Beverly Hills, CA: Sage Publications.

Jöreskog, K. G. (1973), "A General Method for Estimating a Linear Structural Equation System," in *Structural Equation Models in the Social Sciences*, ed. A. S. Goldberger and O. D. Duncan, New York: Seminar Press.

Jöreskog, K. G. and Sörbom, D. (1979), *Advances in Factor Analysis and Structural Equation Models*, Cambridge, MA: Abt Books.

Jöreskog, K. G. and Sörbom, D. (1988), *LISREL 7: A Guide to the Program and Applications*,

Chicago: SPSS.

Keesling, J. W. (1972), "Maximum Likelihood Approaches to Causal Analysis," Ph.D. dissertation, University of Chicago, 1972.

Loehlin, J. C. (1987), *Latent Variable Models*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Lord, F. M. (1957), "A Significance Test for the Hypothesis That Two Variables Measure the Same Trait Except for Errors of Measurement," *Psychometrika*, 22, 207–220.

McArdle, J. J. and McDonald, R. P. (1984), "Some Algebraic Properties of the Reticular Action Model," *British Journal of Mathematical and Statistical Psychology*, 37, 234–251.

McDonald, R. P. (1989), "An Index of Goodness-of-Fit Based on Noncentrality," *Journal of Classification*, 6, 97–103.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Steiger, J. H. and Lind, J. C. (1980), "Statistically Based Tests for the Number of Common Factors," paper presented at the annual meeting of the Psychometric Society, Iowa City, IA.

Voss, R. E. (1969), "Response by Corn to NPK Fertilization on Marshall and Monona Soils as Influenced by Management and Meteorological Factors," Ph.D. dissertation, Iowa State University.

Wiley, D. E. (1973), "The Identification Problem for Structural Equation Models with Unmeasured Variables," in *Structural Equation Models in the Social Sciences*, ed. A. S. Goldberger and O. D. Duncan, New York: Seminar Press, 69–83.

Chapter 18

Shared Concepts and Topics

Contents

Levelization of Classification Variables **366**
Parameterization of Model Effects **368**
GLM Parameterization of Classification Variables and Effects 369
Intercept 369
Regression Effects 369
Main Effects 369
Interaction Effects 370
Nested Effects 371
Continuous-Nesting-Class Effects 371
Continuous-by-Class Effects 372
General Effects 372
Other Parameterizations 373
Constructed Effects and the EFFECT Statement (Experimental) **377**
Collection Effects 378
Multimember Effects 378
Polynomial Effects 380
Spline Effects 384
Splines and Spline Bases 387
Truncated Power Function Basis 388
B-Spline Basis 389
Nonlinear Optimization: The NLOPTIONS Statement **391**
Syntax 391
Remote Monitoring 403
Choosing an Optimization Algorithm 405
First- or Second-Order Algorithms 405
Algorithm Descriptions 406
Programming Statements **410**
References **412**

This chapter introduces a number of concepts that are common to one or more procedures in SAS/STAT, such as the parameterization of effects, the use of the experimental EFFECT statement, and so on. The beginning of each major section displays a listing of the procedures for which the shared topic is relevant.

Levelization of Classification Variables

A classification variable is a variable that enters the statistical analysis or model not through its values, but through its levels. The process of associating values of a variable with levels is termed *levelization*.

A sufficient, but not necessary, condition for a procedure to perform levelization of classification variables is the presence of a CLASS statement. Hence, this section applies to all SAS/STAT procedures that support a CLASS statement. Some procedures use different syntax elements to request levelization of variables (for example, the TRANSREG procedure).

During the process of levelization, observations that share the same value are assigned to the same level. The manner in which values are grouped can be affected by the inclusion of formats. The sort order of the levels can be determined with the ORDER= option in the procedure statement. With the GENMOD, GLMSELECT, and LOGISTIC procedures, you can also control the sorting order separately for each variable in the CLASS statement.

Consider the data on nine observations in Table 18.1. The variable A is integer valued, and variable X is a continuous variable with a missing value for the fourth observations. The fourth and fifth columns of Table 18.1 apply two different formats to the variable X.

Table 18.1 Example Data for Levelization

Obs	A	x	format x 3.0	format x 3.1
1	1	1.09	1	1.1
2	1	1.13	1	1.1
3	1	1.27	1	1.3
4	2	.	.	.
5	2	2.26	2	2.3
6	2	2.48	2	2.5
7	3	3.34	3	3.3
8	3	3.34	3	3.3
9	3	3.14	3	3.1

By default, levelization of the variables groups observations by the formatted value of the variable, except for numerical variables where no explicit format is provided. These are sorted by their internal value. The levelization of the four columns in table Table 18.1 leads to the level assignment in Table 18.2.

Table 18.2 Values and Levels

	A		X		format x 3.0		format x 3.1	
Obs	**Value**	**Level**	**Value**	**Level**	**Value**	**Level**	**Value**	**Level**
1	2	1	1.09	1	1	1	1.1	1
2	2	1	1.13	2	1	1	1.1	1
3	2	1	1.27	3	1	1	1.3	2
4	3	2	.	.	.	.	.	.
5	3	2	2.26	4	2	2	2.3	3
6	3	2	2.48	5	2	2	2.5	4
7	4	3	3.34	7	3	3	3.3	6
8	4	3	3.34	7	3	3	3.3	6
9	4	3	3.14	6	3	3	3.1	5

The ORDER= option in the PROC statement specifies the sorting order for the levels of CLASS variables. When the default ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. To order numeric class levels with no explicit format by their BEST12. formatted values, you can specify this format explicitly for the CLASS variables.

The following table shows how values of the ORDER= option are interpreted.

Value of ORDER=	**Levels Sorted By**
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

For FORMATTED and INTERNAL values, the sort order is machine dependent. For more information about sort order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

The GLMSELECT, LOGISTIC, and GENMOD procedures support a MISSING option in the CLASS statement. When this option is in effect, missing values ('.' for a numeric variable and blanks for a character variable) are included in the levelization and are assigned a level. Table 18.3 displays the results of levelizing the values in Table 18.1 when the MISSING option is in effect.

Table 18.3 Values and Levels with MISSING Option

	A		X		format x 3.0		format x 3.1	
Obs	Value	Level	Value	Level	Value	Level	Value	Level
1	2	1	1.09	2	1	2	1.1	2
2	2	1	1.13	3	1	2	1.1	2
3	2	1	1.27	4	1	2	1.3	3
4	3	2	.	1	.	1	.	1
5	3	2	2.26	5	2	3	2.3	4
6	3	2	2.48	6	2	3	2.5	5
7	4	3	3.34	8	3	4	3.3	7
8	4	3	3.34	8	3	4	3.3	7
9	4	3	3.14	7	3	4	3.1	6

When the MISSING option is not specified, or for procedures whose CLASS statement does not support this option, it is important to understand the implications of missing values for your statistical analysis. When a SAS/STAT procedure levelizes the CLASS variables, an observation for which a CLASS variable has a missing value is excluded from the analysis. This is true regardless of whether the variable is used to form the statistical model. Consider, for example, the case where some observations contain missing values for variable A but the records for these observations are otherwise complete with respect to all other variables in the statistical models. The analysis results from the following statements do not include any observations for which variable A contains missing values, even though A is not specified in the MODEL statement:

```
class A B;
model y = B x B*x;
```

Many statistical procedures print a "Number of Observations" table that shows the number of observations read from the data set and the number of observations used in the analysis. You should pay careful attention to this table—especially when your data set contains missing values—to ensure that no observations are unintentionally excluded from the analysis.

Parameterization of Model Effects

Recall the general form of a linear regression model defined in Chapter 3, "Regression Models and Models with Classification Effects":

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

This section describes how matrices of regressor effects such as $\mathbf{X}$ are constructed in SAS/STAT. These constructions, or *parameterization* rules, apply to regression models, models with classification effects, generalized linear models, and mixed models. The simplest and most general parameterization rules are the ones used in the GLM procedure, and they are discussed first. Several procedures also support alternate parameterizations of classification variables, including the GENMOD,

GLMSELECT, LOGISTIC, SURVEYLOGISTIC, and PHREG procedures. These are discussed after the GLM parameterization of classification variables and model effects.

All modeling procedures that support classification variables and effects have a CLASS statement. Procedures that additionally support the supplemental parameterizations have a PARAM= option in the CLASS statement.

GLM Parameterization of Classification Variables and Effects

This section applies to the following procedures:
GAM, GENMOD, GLIMMIX, GLM, GLMPOWER, GLMSELECT, LIFEREG, LOGISTIC, MI, MIXED, MULLTEST, ORTHOREG, PHREG, PLS, QUANTREG, ROBUSTREG, and SURVEYLOGISTIC.

Intercept

By default, linear models in SAS/STAT automatically include a column of 1s in $\mathbf{X}$ corresponding to an intercept parameter. In many procedures you can use the NOINT option in the MODEL statement to suppress this intercept. The NOINT option is useful, for example, when the MODEL statement contains a classification effect and you want the parameter estimates to be in terms of the mean response for each level of that effect.

Regression Effects

Numeric variables, or polynomial terms involving them, can be included in the model as regression effects (covariates). The actual values of such terms are included as columns of the relevant model matrices. You can use the bar operator with a regression effect to generate polynomial effects. For instance, X|X|X expands to X X*X X*X*X, a cubic model.

Main Effects

If a classification variable has m levels, the GLM parameterization generates m columns for its main effect in the model matrix. Each column is an indicator variable for a given level. The order of the columns is the sort order of the values of their levels and frequently can be controlled with the ORDER= option in the procedure or CLASS statement.

Table 18.4 is an example where β_0 denotes the intercept and A and B are classification variables with two and three levels, respectively.

Table 18.4 Example of Main Effects

Data		I	A		B		
A	B	β_0	A1	A2	B1	B2	B3
1	1	1	1	0	1	0	0
1	2	1	1	0	0	1	0
1	3	1	1	0	0	0	1
2	1	1	0	1	1	0	0
2	2	1	0	1	0	1	0
2	3	1	0	1	0	0	1

Typically, there are more columns for these effects than there are degrees of freedom to estimate them. In other words, the GLM parameterization of main effects is *singular*.

Interaction Effects

Often a model includes interaction (crossed) effects to account for how the effect of a variable changes with the values of other variables. With an interaction, the terms are first reordered to correspond to the order of the variables in the CLASS statement. Thus, B*A becomes A*B if A precedes B in the CLASS statement. Then, the GLM parameterization generates columns for all combinations of levels that occur in the data. The order of the columns is such that the rightmost variables in the interaction change faster than the leftmost variables (Table 18.5). Note that in the MIXED and GLIMMIX procedures, which support both fixed- and random-effects models, empty columns (that is, columns that would contain all 0s) are not generated for fixed effects, but they are generated for random effects.

Table 18.5 Example of Interaction Effects

Data		I	A		B			A*B					
A	B	β_0	A1	A2	B1	B2	B3	A1B1	A1B2	A1B3	A2B1	A2B2	A2B3
1	1	1	1	0	1	0	0	1	0	0	0	0	0
1	2	1	1	0	0	1	0	0	1	0	0	0	0
1	3	1	1	0	0	0	1	0	0	1	0	0	0
2	1	1	0	1	1	0	0	0	0	0	1	0	0
2	2	1	0	1	0	1	0	0	0	0	0	1	0
2	3	1	0	1	0	0	1	0	0	0	0	0	1

In the preceding matrix, main-effects columns are not linearly independent of crossed-effects columns; in fact, the column space for the crossed effects contains the space of the main effect.

When your model contains many interaction effects, you might be able to code them more parsimoniously by using the bar operator (|). The bar operator generates all possible interaction effects. For example, A|B|C expands to A B A*B C A*C B*C A*B*C. To eliminate higher-order interaction effects, use the at sign (@) in conjunction with the bar operator. For instance, A|B|C|D@2 expands to A B A*B C A*C B*C D A*D B*D C*D.

Nested Effects

Nested effects are generated in the same manner as crossed effects. Hence, the design columns generated by the following two statements are the same (but the ordering of the columns is different):

```
model Y=A B(A);

model Y=A A*B;
```

The nesting operator in SAS/STAT software is more of a notational convenience than an operation distinct from crossing. Nested effects are typically characterized by the property that the nested variables never appear as main effects. The order of the variables within nesting parentheses is made to correspond to the order of these variables in the CLASS statement. The order of the columns is such that variables outside the parentheses index faster than those inside the parentheses, and the rightmost nested variables index faster than the leftmost variables (Table 18.6).

Table 18.6 Example of Nested Effects

Data		I	A		B(A)					
A	B	β_0	A1	A2	B1A1	B2A1	B3A1	B1A2	B2A2	B3A2
1	1	1	1	0	1	0	0	0	0	0
1	2	1	1	0	0	1	0	0	0	0
1	3	1	1	0	0	0	1	0	0	0
2	1	1	0	1	0	0	0	1	0	0
2	2	1	0	1	0	0	0	0	1	0
2	3	1	0	1	0	0	0	0	0	1

Continuous-Nesting-Class Effects

When a continuous variable nests or crosses with a classification variable, the design columns are constructed by multiplying the continuous values into the design columns for the classification effect (Table 18.7).

Table 18.7 Example of Continuous-Nesting-Class Effects

Data		I	A		X(A)	
X	A	β_0	A1	A2	X(A1)	X(A2)
21	1	1	1	0	21	0
24	1	1	1	0	24	0
22	1	1	1	0	22	0
28	2	1	0	1	0	28
19	2	1	0	1	0	19
23	2	1	0	1	0	23

This model estimates a separate intercept and a separate slope for X within each level of A.

Continuous-by-Class Effects

Continuous-by-class effects generate the same design columns as continuous-nesting-class effects. Table 18.8 shows the construction of the X*A effect. The two columns for this effect are the same as the columns for the X(A) effect in Table 18.7.

Table 18.8 Example of Continuous-by-Class Effects

Data		**I**	**X**	A		X*A	
X	A	β_0	X	A1	A2	X*A1	X*A2
21	1	1	21	1	0	21	0
24	1	1	24	1	0	24	0
22	1	1	22	1	0	22	0
28	2	1	28	0	1	0	28
19	2	1	19	0	1	0	19
23	2	1	23	0	1	0	23

You can use continuous-by-class effects together with pure continuous effects to test for homogeneity of slopes.

General Effects

An example that combines all the effects is X1*X2*A*B*C(D E). The continuous list comes first, followed by the crossed list, followed by the nested list in parentheses. You should be aware of the sequencing of parameters when you use statements that depend on the ordering of parameters, such as the CONTRAST or ESTIMATE statements in a number of procedures, used to estimate and test functions of the parameter.

Effects might be renamed by the procedure to correspond to ordering rules. For example, B*A(E D) might be renamed A*B(D E) to satisfy the following:

- Classification variables that occur outside parentheses (crossed effects) are sorted in the order in which they appear in the CLASS statement.
- Variables within parentheses (nested effects) are sorted in the order in which they appear in the CLASS statement.

The sequencing of the parameters generated by an effect can be described by which variables have their levels indexed faster:

- Variables in the crossed list index faster than variables in the nested list.
- Within a crossed or nested list, variables to the right index faster than variables to the left.

For example, suppose a model includes four effects—A, B, C, and D—each having two levels, 1 and 2. If the CLASS statement is

```
class A B C D;
```

then the order of the parameters for the effect B*A(C D), which is renamed A*B(C D), is

$$\begin{array}{llll} A_1B_1C_1D_1 \rightarrow & A_1B_2C_1D_1 \rightarrow & A_2B_1C_1D_1 \rightarrow & A_2B_2C_1D_1 \rightarrow \\ A_1B_1C_1D_2 \rightarrow & A_1B_2C_1D_2 \rightarrow & A_2B_1C_1D_2 \rightarrow & A_2B_2C_1D_2 \rightarrow \\ A_1B_1C_2D_1 \rightarrow & A_1B_2C_2D_1 \rightarrow & A_2B_1C_2D_1 \rightarrow & A_2B_2C_2D_1 \rightarrow \\ A_1B_1C_2D_2 \rightarrow & A_1B_2C_2D_2 \rightarrow & A_2B_1C_2D_2 \rightarrow & A_2B_2C_2D_2 \end{array}$$

Note that first the crossed effects B and A are sorted in the order in which they appear in the CLASS statement so that A precedes B in the parameter list. Then, for each combination of the nested effects in turn, combinations of A and B appear. The B effect changes fastest because it is rightmost in the cross list. Then A changes next fastest, and D changes next fastest. The C effect changes slowest because it is leftmost in the nested list.

Other Parameterizations

This section applies to the following procedures:
GENMOD, GLMSELECT, LOGISTIC, and PHREG.

Some SAS/STAT procedures, including GENMOD, GLMSELECT, and LOGISTIC, support nonsingular parameterizations for classification effects. A variety of these nonsingular parameterizations are available. In these procedures you use the PARAM= option in the CLASS statement to specify the parameterization.

Consider a model with one CLASS variable A that has four levels, 1, 2, 5, and 7. Details of the possible choices for the PARAM= option follow.

EFFECT

Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of −1. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

Effect Coding

	Design Matrix		
A	**A1**	**A2**	**A5**
1	1	0	0
2	0	1	0
5	0	0	1
7	−1	−1	−1

Parameter estimates of CLASS main effects that use the effect coding scheme estimate the difference in the effect of each nonreference level compared to the average effect over all four levels.

Note that the EFFECT parameterization is the default parameterization in the CATMOD procedure. See the section "Generation of the Design Matrix" on page 1153, in Chapter 28, "The CATMOD Procedure," for further details about parameterization of model effects with the CATMOD procedure.

GLM

As in the GLM procedure, four columns are created to indicate group membership. The design matrix columns for A are as follows.

GLM Coding				
	Design Matrix			
A	**A1**	**A2**	**A5**	**A7**
1	1	0	0	0
2	0	1	0	0
5	0	0	1	0
7	0	0	0	1

Parameter estimates of CLASS main effects that use the GLM coding scheme estimate the difference in the effects of each level compared to the last level. See the previous section for details about the GLM parameterization of model effects.

ORDINAL

THERMOMETER

Three columns are created to indicate group membership of the higher levels of the effect. For the first level of the effect (which for A is 1), all three dummy variables have a value of 0. The design matrix columns for A are as follows.

Ordinal Coding			
	Design Matrix		
A	**A2**	**A5**	**A7**
1	0	0	0
2	1	0	0
5	1	1	0
7	1	1	1

The first level of the effect is a control or baseline level. Parameter estimates of CLASS main effects, using the ORDINAL coding scheme, estimate the differences between effects of successive levels. When the parameters have the same sign, the effect is monotonic across the levels.

POLYNOMIAL

POLY

Three columns are created. The first represents the linear term (x), the second represents the quadratic term $\left(x^2\right)$, and the third represents the cubic term $\left(x^3\right)$, where x is the level value. If the CLASS levels are not numeric, they are translated into 1, 2, 3, ... according to their sorting order. The design matrix columns for A are as follows.

Polynomial Coding			
	Design Matrix		
A	**APOLY1**	**APOLY2**	**APOLY3**
1	1	1	1
2	2	4	8
5	5	25	125
7	7	49	343

REFERENCE

REF

Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of 0. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

Reference Coding			
	Design Matrix		
A	**A1**	**A2**	**A5**
1	1	0	0
2	0	1	0
5	0	0	1
7	0	0	0

Parameter estimates of CLASS main effects that use the reference coding scheme estimate the difference in the effect of each nonreference level compared to the effect of the reference level.

The REFERENCE parameterization is also available through the MODEL statement in the CATMOD procedure. See the section "Generation of the Design Matrix" on page 1153, in Chapter 28, "The CATMOD Procedure," for further details about parameterization of model effects with the CATMOD procedure.

ORTHEFFECT

The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=EFFECT. The design matrix columns for A are as follows.

Orthogonal Effect Coding			
	Design Matrix		
A	**AOEFF1**	**AOEFF2**	**AOEFF3**
1	1.41421	−0.81650	−0.57735
2	0	1.63299	−0.57735
5	0	0	1.73205
7	−1.41421	−0.81649	−0.57735

ORTHORDINAL

ORTHOTHERM The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=ORDINAL. The design matrix columns for A are as follows.

Orthogonal Ordinal Coding			
	Design Matrix		
A	**AOORD1**	**AOORD2**	**AOORD3**
1	−1.73205	0	0
2	0.57735	−1.63299	0
5	0.57735	0.81650	−1.41421
7	0.57735	0.81650	1.41421

ORTHPOLY The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=POLY. The design matrix columns for A are as follows.

Orthogonal Polynomial Coding			
	Design Matrix		
A	**AOPOLY1**	**AOPOLY2**	**AOPOLY5**
1	−1.153	0.907	−0.921
2	−0.734	−0.540	1.473
5	0.524	−1.370	−0.921
7	1.363	1.004	0.368

ORTHREF The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=REFERENCE. The design matrix columns for A are as follows.

Orthogonal Reference Coding			
	Design Matrix		
A	**AOREF1**	**AOREF2**	**AOREF3**
1	1.73205	0	0
2	−0.57735	1.63299	0
5	−0.57735	−0.81650	1.41421
7	−0.57735	−0.81650	−1.41421

Constructed Effects and the EFFECT Statement (Experimental)

This section applies to the following procedures:
GLIMMIX, GLMSELECT, and QUANTREG.

The experimental EFFECT statement in SAS 9.2 is supported by the GLIMMIX, GLMSELECT, and QUANTREG procedures. The EFFECT statement enables you to construct special collections of columns for design matrices. These collections are referred to as *constructed effects* to distinguish them from the usual model effects formed from continuous or classification variables, as discussed in the section "GLM Parameterization of Classification Variables and Effects" on page 369. For example, the terms A, B, x, A*x, A*B, and sub in the following statements define fixed, random, and subject effects of the usual type in a mixed model, respectively:

```
proc glimmix;
   class A B sub;
   model y = A B x A*x;
   random A*B / subject=sub;
run;
```

A constructed effect, on the other hand, is assigned through the EFFECT statement. For example, in the following program, the EFFECT statement defines a constructed effect named spl:

```
proc glimmix;
   class A B SUB;
   effect spl = spline(x);
   model y = A B A*spl;
   random A*B / subject=sub;
run;
```

The columns of spl are formed from the data set variable x as a cubic B-spline basis with three equally spaced interior knots.

Each constructed effect corresponds to a collection of columns that are referred to by using the name you supply. You can specify multiple EFFECT statements, and all EFFECT statements must precede the MODEL statement.

The general syntax for the EFFECT statement with *effect-specification* is

EFFECT *effect-name* = **effect-type** (*var-list* < / *effect-options* >) **;**

The name of the effect is specified after the EFFECT keyword. This name can appear in only one EFFECT statement and cannot be the name of a variable in the input data set. The effect type is specified after an equal sign, followed by a list of variables used in constructing the effect within parentheses. Effect-type specific options can be specified after a slash (/) following the variable list.

The following *effect-types* are available and subsequently discussed.

COLLECTION	is a collection effect defining one or more variables as a single effect with multiple degrees of freedom. The variables in a collection are considered as a unit for estimation and inference.
MULTIMEMBER | MM	is a multimember classification effect whose levels are determined by one or more variables that appear in a CLASS statement.
POLYNOMIAL | POLY	is a multivariate polynomial effect in the specified numeric variables.
SPLINE	is a regression spline effect whose columns are univariate spline expansions of one or more variables. A spline expansion replaces the original variable with an expanded or larger set of new variables.

Collection Effects

EFFECT *name*=**COLLECTION(***var-list* <**/ DETAILS**>**) ;**

You use a collection effect to define a set of variables that are treated as a single effect with multiple degrees of freedom. The variables in *var-list* can be continuous or classification variables. The columns in the design matrix contributed by a collection effect are the design columns of its constituent variables in the order in which they appear in the definition of the collection effect. If you specify the DETAILS option, then a table showing the constituents of the collection effect is displayed.

Multimember Effects

EFFECT *name*=**MULTIMEMBER(***var-list* <**/** *mm-options*>**) ;**

EFFECT *name*=**MM(***var-list* <**/** *mm-options*>**) ;**

A multimember effect is formed from one or more classification variables in such a way that each observation can be associated with one or more levels of the union of the levels of the classification variables. In other words, a multimember effect is a classification-type effect with possibly more than one nonzero column entry for each observation. Multimember effects are useful, for example, in modeling the following:

- nurses' effects on patient recovery in hospitals
- teachers' effects on student scores
- lineage effects in genetic studies (see Example 38.16 in Chapter 38, "The GLIMMIX Procedure," for an application with random multimember effects in a genetic diallel experiment)

The levels of a multimember effect consist of the union of formatted values of the variables defining this effect. Each such level contributes one column to the design matrix. For each observation, the

value corresponding to each level of the multimember effect in the design matrix is the number of times that this level occurs for the observation.

For example, the following data provide teacher information and end-of-year test scores for students after two semesters:

Student	Score	Teacher1	Teacher2
Mary	87	Tobias	Cohen
Tom	89	Rodriguez	Tobias
Fred	82	Cohen	Cohen
Jane	88	Tobias	.
Jack	99	.	.

For example, Mary had different teachers in the two semesters, Fred had the same teacher in both semesters, and Jane received instruction only in the first semester.

You can model the effect of the teachers on student performance by using a multimember effect specified as follows:

```
CLASS teacher1 teacher2;
EFFECT teacher = MM(teacher1 teacher2);
```

The levels of the teacher effect are {“Cohen”, “Rodriguez”, “Tobias”} and the associated design matrix columns are as follows:

Student	Cohen	Rodriguez	Tobias
Mary	1	0	1
Tom	0	1	1
Fred	2	0	0
Jane	0	0	1
Jack	.	.	.

You can specify the following *mm-options* after a slash (*/*):

DETAILS
requests a table showing the levels of the multimember effect.

NOEFFECT
specifies that, for observations with all missing levels of the multimember variables, the values in the corresponding design matrix columns are set to zero. If, in the preceding example, the teacher effect is defined by

```
EFFECT teacher = MM(teacher1 teacher2 / noeffect);
```

then the associated design matrix columns values for Jack are all zero. This enables you to include Jack in the analysis even though there is no effect of teachers on his performance.

A situation where it is important to designate observations as having no effect due to a classification variable is the analysis of crossover designs, where lagged treatment levels are used to

model the carryover effects of treatments between periods. Since there is no carryover effect for the first period, the treatment lag effect in a crossover design can be modeled with a multimember effect that consists of a single classification variable and the NOEFFECT option, as in the following statements:

```
CLASS Treatment lagTreatment;
EFFECT Carryover = MM(lagTreatment / noeffect);
```

The lagTreatment variable contains a missing value for the first period. Otherwise, it contains the value of the treatment variable for the preceding period.

STDIZE
specifies that for each observation, the entries in the design matrix corresponding to the multimember effect are scaled to have a sum of one.

WEIGHT=(*wght-list***)**
specifies numeric variables used to weigh the contributions of each of the classification effects that define the constructed multimember effect. The number of variables in the WEIGHT= list must match the number of classification variables defining the effect.

Polynomial Effects

This section discusses the construction of multivariate polynomial effects through the experimental EFFECT statement in the GLIMMIX and GLMSELECT procedures. You request a polynomial effect with the syntax

EFFECT *name*=**POLYNOMIAL(***var-list* < / *polynomial-options* >**) ;**

EFFECT *name*=**POLY(***var-list* < / *polynomial-options* >**) ;**

The variables in *var-list* must be numeric. A design matrix column is generated for each term of the specified polynomial. By default, each of these terms is treated as a separate effect for the purpose of model building. For example, the statements

```
proc glmselect;
   effect MyPoly = polynomial(x1-x3/degree=2);
   model y = MyPoly;
run;
```

yield the identical analysis to the statements

```
proc glmselect;
   model y = x1 x2 x3 x1*x1 x1*x2 x1*x3 x2*x2 x2*x3 x3*x3;
run;
```

You can specify the following *polynomial-options* after a slash (/):

DEGREE=*n*
specifies the degree of the polynomial. The degree must be a positive integer. The degree is typically a small integer, such as 1, 2, or 3. The default is DEGREE=1.

DETAILS

requests a table showing the details of the specified polynomial, including the number of terms generated. If you specify the STANDARDIZE option, then a table showing the standardization details is also produced.

LABELSTYLE=(*style-opts***)**

LABELSTYLE=*style-opt*

specifies how the terms in the polynomial are labeled. By default, powers are shown with ^ as the exponentiation operator and ∗ as the multiplication operator. For example, a polynomial term such as $x_1^3 x_2 x_3^2$ is labeled x1^3*x2*x3^2. You can change the style of the label by using the following *style-opts* within parentheses. If you specify a single *style-opt*, then you can omit the enclosing parentheses.

EXPAND

specifies that each variable with an exponent greater than one is written as products of that variable. For example, the term $x_1^3 x_2 x_3^2$ receives the label x1*x1*x1*x2*x3*x3.

EXPONENT < =*quoted string* **>**

specifies that each variable with an exponent greater than one is written using exponential notation. By default, the symbol ^ is used as the exponentiation operator. If you supply the optional quoted string after an equal sign, then that string is used as the exponentiation operator. For example, if you specify

```
LABELSTYLE=(EXPONENT="**")
```

then the term $x_1^3 x_2 x_3^2$ receives the label x1**3*x2*x3**2.

INCLUDENAME

specifies that the name of the effect followed by an underscore is used as a prefix for term labels. For example, the statement

```
EFFECT MyPoly=POLYNOMIAL(x1/degree=2 labelstyle=INCLUDENAME)
```

generates terms with labels MyPoly_x1 and MyPoly_x1^2. The INCLUDENAME option is ignored if you specify the NOSEPARATE option in the EFFECT=POLYNOMIAL statement.

PRODUCTSYMBOL =NONE | *quoted string*

specifies that the supplied string be used as the product symbol. For example, the statement

```
EFFECT MyPoly=POLYNOMIAL(x1 x2 / degree=2 mdegree=1
                                 labelstyle=(PRODUCTSYMBOL=" "))
```

generates terms with labels x1, x2, and x1 x2.

If you specify PRODUCTSYMBOL=NONE, then the labels are formed by juxtaposing the constituent variable names.

MDEGREE=*n*

specifies the maximum degree of any variable in a term of the polynomial. This degree must

be a positive integer. The default is the degree of the specified polynomial. For example, the statement

```
EFFECT MyPoly=POLYNOMIAL(x1 x2/degree=4 MDEGREE=2);
```

generates the terms x_1, x_2, x_1^2, x_1x_2, x_2^2,$x_1^2x_2$, and $x_1x_2^2$.

NOSEPARATE

specifies that the polynomial is treated as a single effect with multiple degrees of freedom. The effect name that you specify is used as the constructed effect name and the labels of the terms are used as labels of the corresponding parameters.

STANDARDIZE < (*centerscale-opts*) > < = *standardize-opt* >

specifies that the variables defining the polynomial are standardized. By default, the standardized variables receive prefix "s_" in the variable names.

You can use the following *centerscale-opts* to specify how the center and scale are estimated:

METHOD=MOMENTS

specifies that the center is estimated by the variable mean and the scale is estimated by the standard deviation. Note that if a weight variable is specified using a WEIGHT statement, the observations with invalid weights are ignored when forming the mean and standard deviation, but the weights are otherwise not used. Note that only observations that are used in performing the analysis are used for the standardization.

METHOD=RANGE

specifies that the center is estimated by the midpoint of the variable range and the scale is estimated as half the variable range. Any observation that has a missing value for any regressor used in the model is ignored when computing the range of variables in a polynomial effect. Observations with valid regressor values but missing or invalid values of frequency variables, weight variables, or dependent variables are used in computing variable ranges. The default (if you do not specify the METHOD= suboption) is METHOD=RANGE.

METHOD=WMOMENTS

is the same as METHOD=MOMENTS except that weighted means and weighted standard deviations are used.

Let

$$
\begin{aligned}
n &= \text{number of observations used in the analysis} \\
w &= \text{weight variable} \\
f &= \text{frequency variable} \\
x &= \text{variable to be standardized} \\
x_{(n)} &= \text{Max}_{i=1}^{n}(x_i) \\
x_{(1)} &= \text{Min}_{i=1}^{n}(x_i) \\
F &= \text{sum of frequencies} \\
&= \Sigma_{i=1}^{n} f_i \\
WF &= \text{sum of weighted frequencies} \\
&= \Sigma_{i=1}^{n} w_i f_i
\end{aligned}
$$

Table 18.9 shows how the center and scale are computed for each of the supported methods:

Table 18.9 Center and Scale Estimates by Method

Method	Center	Scale
Range	$(x_{(n)} - x_{(1)})/2$	$(x_{(n)} + x_{(1)})/2$
Moments	$\bar{x} = \Sigma_{i=1}^{n} f_i x_i / F$	$\sqrt{\Sigma_{i=1}^{n} f_i (x_i - \bar{x})^2/(F-1)}$
WMoments	$\bar{x}_w = \Sigma_{i=1}^{n} w_i f_i x_i / WF$	$\sqrt{\Sigma_{i=1}^{n} w_i f_i (x_i - \bar{x}_w)^2/(F-1)}$

PREFIX=NONE | *quoted string*

specifies the prefix that is appended to standardized variables when forming the term labels. If you omit this option, the default prefix is "s_". If you specify PREFIX=NONE, then standardized variables are not prefixed.

You can control whether the standardization is to center, scale, or both center and scale by specifying a *standardize-opt*:

CENTER

specifies that variables are centered but not scaled. For a variable x,

$$\text{s\_x} = x - \text{center}$$

CENTERSCALE

specifies that variables are centered and scaled. This is the default if you do not specify a *standardization-opt*. For a variable x,

$$\text{s\_x} = \frac{x - \text{center}}{\text{scale}}$$

NONE

specifies that no standardization is performed.

SCALE

specifies that variables are scaled but not centered. For a variable x,

$$\text{s\_x} = \frac{x}{\text{scale}}$$

Spline Effects

This section discusses the construction of spline effects through the experimental EFFECT statement in the GLIMMIX, GLMSELECT, and QUANTREG procedures. You can also include spline effects in statistical models by other means. The TRANSREG procedure has dedicated facilities for including regression splines in your model and controlling the construction of the splines. For example, you can use the TRANSREG procedure to fit a spline function but restrict the function to be always increasing or decreasing (monotone). See the section "Using Splines and Knots" on page 7192 in Chapter 90, "The TRANSREG Procedure," for more information about using splines with the TRANSREG procedure. The GAM and TPSPLINE procedures also can model the effects of regressor variables in terms of smooth functions that are generated from spline bases. For more information see Chapter 36, "The GAM Procedure," and Chapter 89, "The TPSPLINE Procedure."

A spline effect expands variables into spline bases whose form depends on the options that you specify. You can find details about regression splines and spline bases in the section "Splines and Spline Bases" on page 387. You request a spline effect with the syntax

EFFECT *name*=**SPLINE(***var-list* < / *spline-options* >**) ;**

The variables in *var-list* must be numeric. Design matrix columns are generated separately for each of these variables and the set of columns is collectively referred to with the specified name. By default, the spline basis generated for each variable is a cubic B-spline basis with three equally spaced knots positioned between the minimum and maximum values of that variable. This yields by default seven design matrix columns for each of the variables in the SPLINE effect.

You can specify the following *spline-options* after a slash (/):

BASIS=BSPLINE

specifies a B-spline basis for the spline expansion. For splines of degree d defined with n knots, this basis consists of $n + d + 1$ columns. In order to completely specify the B-spline basis, d left-side boundary knots and $\max\{d, 1\}$ right-side boundary knots are also required. See the suboptions KNOTMETHOD=, DATABOUNDARY=, KNOTMIN=, and KNOTMAX= for details about how to specify the positions of both the internal and boundary knots. This is the default if you do not specify the BASIS= suboption.

BASIS=TPF(*options***)**

specifies a truncated power function basis for the spline expansion. For splines of degree d defined with n knots for a variable x, this basis consists of an intercept, polynomials x, $x^2,\ldots,x^d$ and one truncated power function for each of the n knots. Note that unlike the B-spline basis no boundary knots are required. See the suboption KNOTMETHOD= for details about how you can specify the position of the internal knots.

You can modify the number of columns when you request BASIS=TPF with the following suboptions:

NOINT
excludes the intercept column.

NOPOWERS
excludes the intercept and polynomial columns.

DATABOUNDARY
specifies that the extremes of the data be used as boundary knots when building a B-spline basis.

DEGREE=*n*
specifies the degree of the spline transformation. The degree must be a nonnegative integer. The degree is typically a small integer, such as 0, 1, 2, or 3. The default is DEGREE=3.

DETAILS
requests tables showing the knot locations and the knots associated with each spline basis function.

KNOTMAX=*value*
specifies that, for each variable in the EFFECT statement, the right-side boundary knots be equally spaced starting at the maximum of the variable and ending at the specified value. This option is ignored for variables whose maximum value is greater than the specified value or if the DATABOUNDARY option is also specified.

KNOTMETHOD=*knot-method***<(***knot-options***)>**
specifies how the knots for spline effects are constructed. You can choose from the following *knot-methods* and affect the knot construction further with the type-specific *knot-options*:

EQUAL<(*n***)>**
specifies that n equally spaced knots be positioned between the extremes of the data. The default is $n = 3$. For a B-spline basis, any needed boundary knots continue to be equally spaced unless the DATABOUNDARY option has also been specified. KNOTMETHOD=EQUAL is the default if no *knot-method* is specified.

LIST(*number-list***)**
specifies the list of internal knots to be used in forming the spline basis columns. For a B-spline basis, the data extremes are used as boundary knots.

LISTWITHBOUNDARY(*number-list***)**
specifies the list of all knots in forming the spline basis columns. When you use a truncated power function basis, this list is interpreted as the list of internal knots. When you use a B-spline basis of degree d, then the first d entries are used as left-side boundary knots and the last MAX(d, 1) entries in the list are used as right-side boundary knots.

MULTISCALE<(*multiscale-options***)>**
specifies that multiple B-spline bases be generated, corresponding to sets with an increasing number of internal knots. As you increase the number of internal knots, the spline basis you generate is able to approximate features of the data at finer scales. So, by generating bases at multiple scales, you facilitate the modeling of both coarse- and fine-grained features of the data. For scale i, the spline basis corresponds to 2^i equally

spaced internal knots. By default, the bases for scales 0–7 are generated. For each scale, a separate spline effect is generated. The name of the constructed spline effect at scale *i* is formed by appending _S*i* to the effect name you specify in the EFFECT statement. If you specify multiple variables in the EFFECT statement, then spline bases are generated separately for each variable at each scale and the name of the corresponding effect is obtained by appending the variable name followed by _S*i* to the name in the EFFECT statement. For example, the following statement generates effects named spl_x1_S0, spl_x1_S1, spl_x1_S2, . . ., spl_x1_S7 and spl_x2_S1, spl_x2_S2, . . ., spl_x2_S7:

```
EFFECT spl = spline(x1 x2 / knotmethod=multiscale);
```

This option is ignored if you specify the BASIS=TPF suboption. It is not available for spline effects specified in the RANDOM statement of the GLIMMIX procedure.

You can control which scales are included with the following *multiscale-options*:

STARTSCALE=*n*
where *n* is a positive integer. The default is STARTSCALE=0.

ENDSCALE=*n*
where *n* is a positive integer. The default is ENDSCALE=7.

PERCENTILES(*n***)**
requests that internal knots be placed at *n* equally spaced percentiles of the variable or variables named in the EFFECT statement. For example, the following statement positions internal knots at the deciles of the variable x. For a B-spline basis, the extremes of the data are used as boundary knots:

```
EFFECT spl = spline(x / knotmethod=percentiles(9));
```

RANGEFRACTIONS(*fraction-list***)**
requests that internal knots be placed at each fraction of the ranges of the variables in the EFFECT statement. For example, if variable x1 ranges between 1 and 3, and variable x2 ranges between 0 and 20, then the following EFFECT statement uses internal knots 1.2, 2, and 2.5 for variable x1 and internal knots 2, 10, and 15 for variable x2:

```
EFFECT spl = spline(x1 x2 / knotmethod=rangefractions(.1 .5 .75));
```

For a B-spline basis, the data extremes are used as boundary knots.

KNOTMIN=*value*
specifies that for each variable in the EFFECT statement, the left-side boundary knots be equally spaced starting at the specified value and ending at the minimum of the variable. This option is ignored for variables whose minimum value is less than the specified value or if the DATABOUNDARY option is also specified.

SEPARATE
specifies that when multiple variables are specified in the EFFECT statement, the spline basis for each variable is treated as a separate effect. The names of these separated effects are formed by appending an underscore followed by the name of the variable to the name that you specify in the EFFECT statement. For example, the effect names generated with the following statement are spl_x1 and spl_x2:

```
EFFECT spl = spline(x1 x2 / separate);
```

In procedures that support variable selection, such as the GLMSELECT procedure, these two effects can enter or leave the model independently during the selection process. Separated effects are not supported in the RANDOM statement of the GLIMMIX procedure

SPLIT

specifies that each individual column in the design matrix corresponding to the spline effect is treated as a separate effect that can enter or leave the model independently. Names for these split effects are generated by appending the variable name and an index for each column to the name that you specify in the EFFECT statement. For example, the effects generated for the spline effect in the following statement are spl_x1:1, spl_x1:2, ...,spl_x1:7, spl_x2:1, spl_x2:2, ...,spl_x2:7:

```
EFFECT spl = spline(x1 x2 / split);
```

The SPLIT option is not supported in the GLIMMIX procedure.

Splines and Spline Bases

This section provides details about the construction of spline bases with the experimental EFFECT statement. A spline function is a piecewise polynomial function where the individual polynomials have the same degree and connect smoothly at join points whose abscissa values, referred to as knots, are prespecified. You can use spline functions to fit curves to a wide variety of data.

A spline of degree 0 is a step function with steps located at the knots. A spline of degree 1 is a piecewise linear function where the lines connect at the knots. A spline of degree 2 is a piecewise quadratic curve whose values and slopes coincide at the knots. A spline of degree 3 is a piecewise cubic curve whose values, slopes, and curvature coincide at the knots. Visually, a cubic spline is a smooth curve, and it is the most commonly used spline when a smooth fit is desired. Note that when no knots are used, splines of degree d are simply polynomials of degree d.

More formally, suppose you specify knots $k_1 < k_2 < k_3 < \cdots < k_n$. Then a spline of degree $d \geq 0$ is a function $S(x)$ with $d-1$ continuous derivatives such that

$$S(x) = \begin{cases} P_0(x) & x < k_1 \\ P_i(x) & k_i \leq x < k_{i+1};\ i = 1, 2, \ldots, n-1 \\ P_n(x) & x \geq k_n \end{cases}$$

where each $P_i(x)$ is a polynomial of degree d. The requirement that $S(x)$ has $d-1$ continuous derivatives is satisfied by requiring that the function values and all derivatives up to order $d-1$ of the adjacent polynomials at each knot match.

A counting argument yields the number of parameters that define a spline with n knots. There are $n+1$ polynomials of degree d, giving $(n+1)(d+1)$ coefficients. However, there are d restrictions at each of the n knots, so the number of free parameters is $(n+1)(d+1) - nd = n + d + 1$. In mathematical terminology this says that the dimension of the vector space of splines of degree d on n distinct knots is $n + d + 1$. If you have $n + d + 1$ basis vectors, then you can fit a curve to your

data by regressing your dependent variable by using this basis for the corresponding design matrix columns. In this context, such a spline is known as a regression spline. The EFFECT statement provides a simple mechanism for obtaining such a basis.

If you remove the restriction that the knots of a spline must be distinct and allow repeated knots, then you can obtain functions with less smoothness and even discontinuities at the repeated knot location. For a spline of degree d and a repeated knot with multiplicity $m \leq d$, the piecewise polynomials that join such a knot are required to have only $d - m$ matching derivatives. Note that this increases the number of free parameters by $m - 1$ but also decreases the number of distinct knots by $m - 1$. Hence the dimension of the vector space of splines of degree d with n knots is still $n + d + 1$, provided that any repeated knot has a multiplicity less than or equal to d.

The EFFECT statement provides support for the commonly used *truncated power function* basis and *B-spline* basis. With exact arithmetic and by using the complete basis, you will obtain the same fit with either of these bases. The following sections provide details about constructing spline bases for the space of splines of degree d with n knots satisfying $k_1 \leq k_2 \leq k_3 < \cdots \leq k_n$.

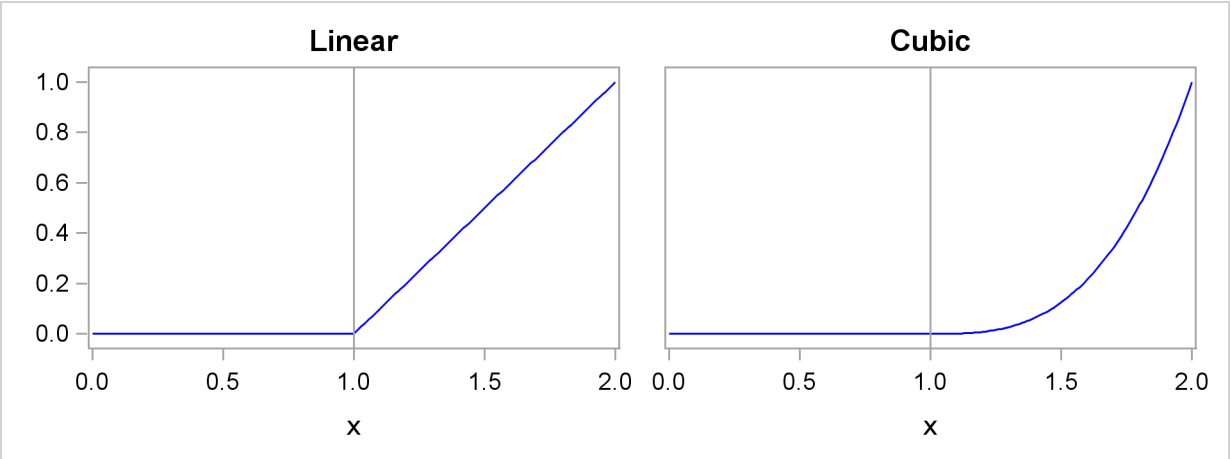
Truncated Power Function Basis

A truncated power function for a knot k_i is a function defined by

$$t_i(x) = \begin{cases} 0 & x < k_i \\ (x - k_i)^d & x \geq k_i \end{cases}$$

Figure 18.1 shows such functions for $d = 1$ and $d = 3$ with a knot at $x = 1$.

Figure 18.1 Truncated Power Functions with Knot at $x = 1$



The name is derived from the fact that these functions are shifted power functions that get truncated to zero to the left of the knot. These functions are piecewise polynomial functions with two pieces whose function values and derivatives of all orders up to $d - 1$ are zero at the defining knot. Hence these functions are splines of degree d. It is easy to see that these n functions are linearly independent. However, they do not form a basis, because such a basis requires $n + d + 1$ functions. The usual way to add $d + 1$ additional basis functions is to use the polynomials $1, x, x^2, \ldots, x^d$.

These $d + 1$ functions together with the n truncated power functions $t_i(x), i = 1, 2, \ldots, n$ form the truncated power basis.

Note that each time a knot is repeated, the associated exponent used in the corresponding basis function is reduced by 1. For example, for splines of degree d with 3 repeated knots $k_i = k_{i+1} = k_{i+2}$ the corresponding basis functions are $t_i(x) = ((x - k_i)_+^d$, $t_{i+1}(x) = ((x - k_i)_+^{d-1}$, and $t_{i+2}(x) = ((x - k_i)_+^{d-2}$. Provided that the multiplicity of each repeated knot is less than or equal to the degree, this construction continues to yield a basis for the associated space of splines.

The main advantage of the truncated power function basis is the simplicity of its construction and the ease of interpreting the parameters in a model corresponding to these basis functions. However, there are two weaknesses when you use this basis for regression. These functions grow rapidly without bound as x increases, resulting in numerical precision problems when the x data span a wide range. Furthermore, many or even all of these basis functions can be nonzero when evaluated at some x value, resulting in a design matrix with few zeros that precludes the use of sparse matrix technology to speed up computation. This weakness can be addressed by using a B-spline basis.

B-Spline Basis

A B-spline basis can be built by starting with a set of Haar basis functions, which are functions that are 1 between adjacent knots and zero elsewhere, and then applying a simple linear recursion relationship d times, yielding the $n + d + 1$ needed basis functions. For the purpose of building the B-spline basis, the n prespecified knots are referred to as internal knots. This construction requires d additional knots, known as boundary knots, to be positioned to the left of the internal knots, and MAX$(d, 1)$ boundary knots to be positioned to the right of the internal knots. The actual values of these boundary knots can be arbitrary. The EFFECT statement provides several methods for placing the needed boundary knots, including the common method of using repeated values of the data extremes as the boundary knots. The boundary knot placement affects the precise form of the basis functions generated but not the following two desirable properties:

1. The B-spline basis functions are nonzero over an interval spanning at most $d + 2$ knots. This yields design matrix columns each of whose rows contain at most $d + 2$ adjacent nonzero entries.

2. The computation of the basis functions at any x value is numerically stable and does not require evaluating powers of this value.

The following figures show the B-spline bases defined on [0, 1] with 4 equally spaced internal knots at 0.2, 0.4, 0.6, and 0.8.

Figure 18.2 shows a linear B-spline basis. Note that this basis consists of 6 functions each of which is nonzero over an interval spanning at most 3 knots.

Figure 18.2 Linear B-Spline Basis with Four Equally Spaced Interior Knots

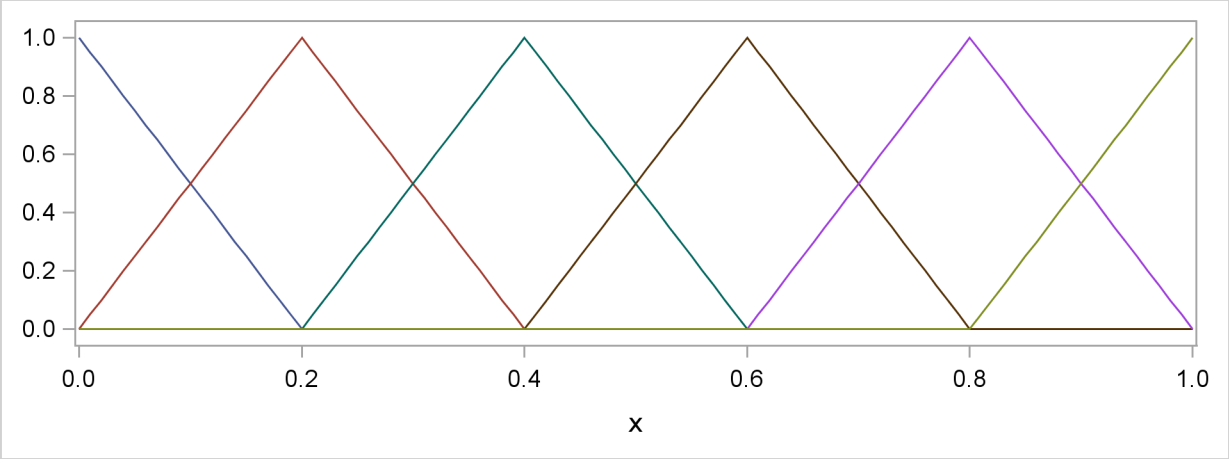


Figure 18.3 shows a cubic B-spline basis where the needed boundary knots are positioned at $x = 0$ and $x = 1$. Note that this basis consists of 8 functions, each of which is nonzero over an interval spanning at most 5 knots.

Figure 18.3 Cubic B-Spline Basis with Four Equally Spaced Interior Knots

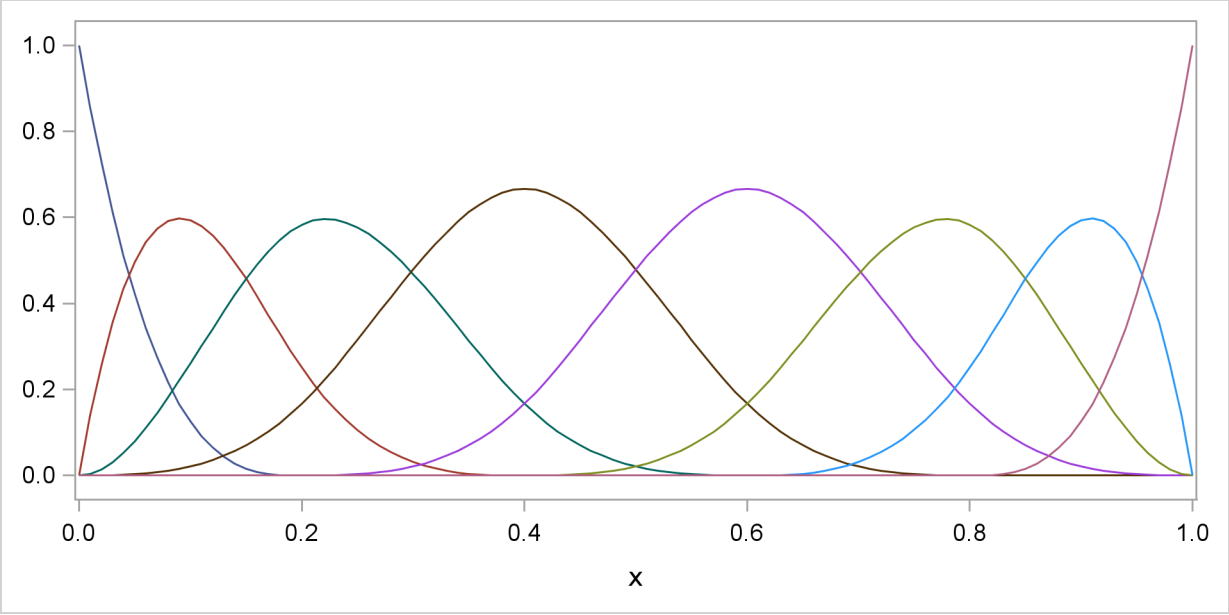
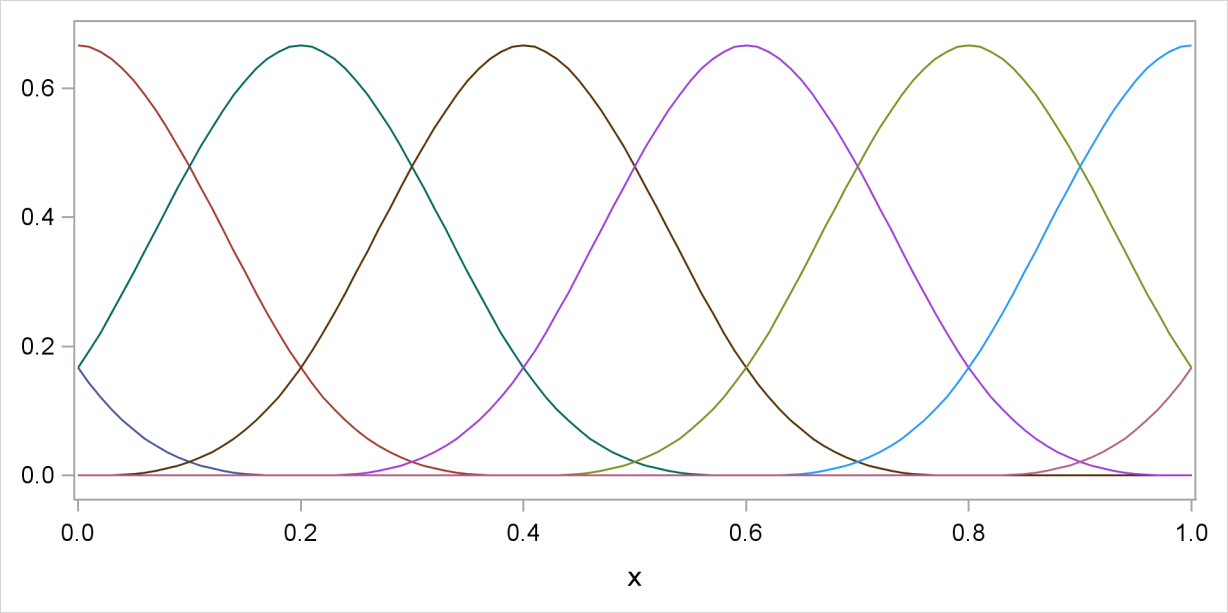


Figure 18.4 shows a different cubic B-spline basis where the needed left-side boundary knots are positioned at −0.6, −0.4, −0.2, and 0. The right-side boundary knots are positioned at 1, 1.2, 1.4, and 1.6. Note that, as in the basis shown in Figure 18.3, this basis consists of 8 functions, each of which is nonzero over an interval spanning at most 5 knots. The different positioning of the boundary knots has merely changed the shape of the individual basis functions.

Figure 18.4 Cubic B-Spline Basis with Equally Spaced Boundary and Interior Knots



You can find details about this construction in Hastie, Tibshirani, and Friedman (2001).

Nonlinear Optimization: The NLOPTIONS Statement

This section applies to the following procedures:
GLIMMIX, HPMIXED, and TCALIS. See the individual procedure chapters on deviations from the common syntax and defaults shown here.

Syntax

The NLOPTIONS statement provides you with syntax to control aspects of the nonlinear optimizations in the GLIMMIX, HPMIXED, and TCALIS procedures.

NLOPTIONS < *options* > **;**

The nonlinear optimization options are described in alphabetical order after Table 18.10, which summarizes the options by category. The notation used in describing the options is generic in the sense that $\boldsymbol{\psi}$ denotes the $p \times 1$ vector of parameters for the optimization and ψ_i is its ith element. The objective function being minimized, its $p \times 1$ gradient vector, and its $p \times p$ Hessian matrix are denoted as $f(\boldsymbol{\psi})$, $\mathbf{g}(\boldsymbol{\psi})$, and $\mathbf{H}(\boldsymbol{\psi})$, respectively. The gradient with respect to the ith parameter is denoted as $g_i(\boldsymbol{\psi})$. Superscripts in parentheses denote the iteration count; for example, $f(\boldsymbol{\psi})^{(k)}$ is the value of the objective function at iteration k. In the mixed model procedures, the

parameter vector ψ might consist of fixed effects only, covariance parameters only, or fixed effects and covariance parameters. In the TCALIS procedure, ψ consists of all independent parameters defined in the models and in the PARAMETERS statement.

Table 18.10 Options to Control Aspects of the Optimization

Option	Description
Optimization	
HESCAL=	determines the type of Hessian scaling
INHESSIAN=	specifies the start for approximated Hessian
LINESEARCH=	specifies the line-search method
LSPRECISION=	specifies the line-search precision
RESTART=	specifies the iteration number for update restart
TECHNIQUE=	determines the minimization technique
UPDATE=	determines the update technique
Termination Criteria	
ABSCONV=	tunes an absolute function convergence criterion
ABSFCONV=	tunes an absolute function convergence criterion
ABSGCONV=	tunes the absolute gradient convergence criterion
ABSXCONV=	tunes the absolute parameter convergence criterion
FCONV=	tunes the relative function convergence criterion
FCONV2=	tunes another relative function convergence criterion
FSIZE=	specifies the value used in FCONV, GCONV criterion
GCONV=	tunes the relative gradient convergence criterion
GCONV2=	tunes another relative gradient convergence criterion
MAXFUNC=	specifies the maximum number of function calls
MAXITER=	specifies the maximum number of iterations
MAXTIME=	specifies the upper limit seconds of CPU time
MINITER=	specifies the minimum number of iterations
XCONV=	specifies the relative parameter convergence criterion
XSIZE=	specifies the value used in XCONV criterion
Step Length	
DAMPSTEP=	dampens steps in line search
INSTEP=	specifies the initial trust region radius
MAXSTEP=	specifies the maximum trust region radius
Printed Output	
PALL	displays (almost) all printed output
PHISTORY	displays optimization history
NOPRINT	suppresses all printed output
Covariance Matrix Tolerances	
ASINGULAR=	absolute singularity for inertia
MSINGULAR=	relative M singularity for inertia
VSINGULAR=	relative V singularity for inertia

Table 18.10 *continued*

Option	Description
Constraint Specifications	
LCEPSILON=	range for active constraints
LCDEACT=	LM tolerance for deactivating
LCSINGULAR=	tolerance for dependent constraints
Remote Monitoring	
SOCKET=	specifies the fileref for remote monitoring

ABSCONV=*r*

ABSTOL=*r*

specifies an absolute function convergence criterion. For minimization, termination requires $f(\boldsymbol{\psi}^{(k)}) \leq r$. The default value of r is the negative square root of the largest double-precision value, which serves only as a protection against overflows.

ABSFCONV=*r* **<** *n* **>**

ABSFTOL=*r* **<** *n* **>**

specifies an absolute function convergence criterion. For all techniques except NMSIMP, termination requires a small change of the function value in successive iterations:

$$|f(\boldsymbol{\psi}^{(k-1)}) - f(\boldsymbol{\psi}^{(k)})| \leq r$$

The same formula is used for the NMSIMP technique, but $\boldsymbol{\psi}(k)$ is defined as the vertex with the lowest function value, and $\boldsymbol{\psi}^{(k-1)}$ is defined as the vertex with the highest function value in the simplex. The default value is $r = 0$. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

ABSGCONV=*r* **<** *n* **>**

ABSGTOL=*r* **<** *n* **>**

specifies an absolute gradient convergence criterion. Termination requires the maximum absolute gradient element to be small:

$$\max_j |g_j(\boldsymbol{\psi}^{(k)})| \leq r$$

This criterion is not used by the NMSIMP technique. The default value is r =1E−5. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

ABSXCONV=*r* **<** *n* **>**

ABSXTOL=*r* **<** *n* **>**

specifies an absolute parameter convergence criterion. For all techniques except NMSIMP, termination requires a small Euclidean distance between successive parameter vectors,

$$\| \boldsymbol{\psi}^{(k)} - \boldsymbol{\psi}^{(k-1)} \|_2 \leq r$$

For the NMSIMP technique, termination requires either a small length $\alpha^{(k)}$ of the vertices of a restart simplex,

$$\alpha^{(k)} \leq r$$

or a small simplex size,

$$\delta^{(k)} \leq r$$

where the simplex size $\delta^{(k)}$ is defined as the L1 distance from the simplex vertex $\boldsymbol{\xi}^{(k)}$ with the smallest function value to the other p simplex points $\boldsymbol{\psi}_l^{(k)} \neq \boldsymbol{\xi}^{(k)}$:

$$\delta^{(k)} = \sum_{\boldsymbol{\psi}_l \neq y} \| \boldsymbol{\psi}_l^{(k)} - \boldsymbol{\xi}^{(k)} \|_1$$

The default is r =1E−8 for the NMSIMP technique and $r = 0$ otherwise. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

ASINGULAR=*r*

ASING=*r*

specifies an absolute singularity criterion for the computation of the inertia (number of positive, negative, and zero eigenvalues) of the Hessian and its projected forms. The default value is the square root of the smallest positive double-precision value.

DAMPSTEP<=*r***>**

specifies that the initial step length value $\alpha^{(0)}$ for each line search (used by the QUANEW, CONGRA, or NEWRAP technique) cannot be larger than r times the step length value used in the former iteration. If the DAMPSTEP option is specified but r is not specified, the default is $r = 2$. The DAMPSTEP=*r* option can prevent the line-search algorithm from repeatedly stepping into regions where some objective functions are difficult to compute or where they could lead to floating-point overflows during the computation of objective functions and their derivatives. The DAMPSTEP=*r* option can save time-consuming function calls during the line searches of objective functions that result in very small steps.

FCONV=*r***<** *n***>**

FTOL=*r***<** *n***>**

specifies a relative function convergence criterion. For all techniques except NMSIMP, termination requires a small relative change of the function value in successive iterations,

$$\frac{|f(\boldsymbol{\psi}^{(k)}) - f(\boldsymbol{\psi}^{(k-1)})|}{\max(|f(\boldsymbol{\psi}^{(k-1)})|, \text{FSIZE})} \leq r$$

where FSIZE is defined by the FSIZE= option. The same formula is used for the NMSIMP technique, but $\boldsymbol{\psi}^{(k)}$ is defined as the vertex with the lowest function value, and $\boldsymbol{\psi}^{(k-1)}$ is defined as the vertex with the highest function value in the simplex.

The default is $r = 10^{-\text{FDIGITS}}$, where FDIGITS is by default $-\log_{10}\{\epsilon\}$ and ϵ is the machine precision. Some procedures, such as the GLIMMIX procedure, enable you to change the value with the FDIGITS= option in the PROC statement. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

FCONV2=*r*< *n* >

FTOL2=*r*< *n* >

specifies a second function convergence criterion. For all techniques except NMSIMP, termination requires a small predicted reduction,

$$df^{(k)} \approx f(\boldsymbol{\psi}^{(k)}) - f(\boldsymbol{\psi}^{(k)} + \mathbf{s}^{(k)})$$

of the objective function. The predicted reduction

$$\begin{aligned} df^{(k)} &= -\mathbf{g}^{(k)'}\mathbf{s}^{(k)} - \frac{1}{2}\mathbf{s}^{(k)'}\mathbf{H}^{(k)}\mathbf{s}^{(k)} \\ &= -\frac{1}{2}\mathbf{s}^{(k)'}\mathbf{g}^{(k)} \leq r \end{aligned}$$

is computed by approximating the objective function f by the first two terms of the Taylor series and substituting the Newton step,

$$\mathbf{s}^{(k)} = -[\mathbf{H}^{(k)}]^{-1}\mathbf{g}^{(k)}$$

For the NMSIMP technique, termination requires a small standard deviation of the function values of the $p+1$ simplex vertices $\boldsymbol{\psi}_l^{(k)}$, $l = 0, \ldots, p$,

$$\sqrt{\frac{1}{n+1}\sum_l \left[f(\boldsymbol{\psi}_l^{(k)}) - \overline{f}(\boldsymbol{\psi}^{(k)})\right]^2} \leq r$$

where $\overline{f}(\boldsymbol{\psi}^{(k)}) = \frac{1}{p+1}\sum_l f(\boldsymbol{\psi}_l^{(k)})$. If there are p_{act} boundary constraints active at $\boldsymbol{\psi}^{(k)}$, the mean and standard deviation are computed only for the $n+1-p_{act}$ unconstrained vertices. The default value is $r =$1E−6 for the NMSIMP technique and $r = 0$ otherwise. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

FSIZE=*r*

specifies the FSIZE parameter of the relative function and relative gradient termination criteria. The default value is $r = 0$. For more details, see the FCONV= and GCONV= options.

GCONV=*r*< *n* >

GTOL=*r*< *n* >

specifies a relative gradient convergence criterion. For all techniques except CONGRA and NMSIMP, termination requires that the normalized predicted function reduction be small,

$$\frac{\mathbf{g}(\boldsymbol{\psi}^{(k)})'[\mathbf{H}^{(k)}]^{-1}\mathbf{g}(\boldsymbol{\psi}^{(k)})}{\max(|f(\boldsymbol{\psi}^{(k)})|, \mathrm{FSIZE})} \leq r$$

where FSIZE is defined by the FSIZE= option. For the CONGRA technique (where a reliable Hessian estimate $\mathbf{H}$ is not available), the following criterion is used:

$$\frac{\| \mathbf{g}(\boldsymbol{\psi}^{(k)}) \|_2^2 \quad \| \mathbf{s}(\boldsymbol{\psi}^{(k)}) \|_2}{\| \mathbf{g}(\boldsymbol{\psi}^{(k)}) - \mathbf{g}(\boldsymbol{\psi}^{(k-1)}) \|_2 \max(|f(\boldsymbol{\psi}^{(k)})|, \mathrm{FSIZE})} \leq r$$

This criterion is not used by the NMSIMP technique. The default value is $r =$1E−8. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

GCONV2=*r*< *n* >

GTOL2=*r*< *n* >

specifies another relative gradient convergence criterion. For least-squares problems and the TRUREG, LEVMAR, NRRIDG, and NEWRAP techniques, the following criterion of Browne (1982) is used:

$$\max_j \frac{|\mathbf{g}_j(\boldsymbol{\psi}^{(k)})|}{\sqrt{f(\boldsymbol{\psi}^{(k)})\mathbf{H}^{(k)}_{j,j}}} \leq r$$

This criterion is not used by the other techniques. The default value is $r = 0$. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

HESCAL=0 | 1 | 2 | 3

HS=0 | 1 | 2 | 3

specifies the scaling version of the Hessian (or crossproduct Jacobian) matrix used in NRRIDG, TRUREG, LEVMAR, NEWRAP, or DBLDOG optimization.

If HS is not equal to 0, the first iteration and each restart iteration set the diagonal scaling matrix $D^{(0)} = \text{diag}(d_i^{(0)})$:

$$d_i^{(0)} = \sqrt{\max(|H_{i,i}^{(0)}|, \epsilon)}$$

where $H_{i,i}^{(0)}$ are the diagonal elements of the Hessian (or crossproduct Jacobian). In every other iteration, the diagonal scaling matrix $D^{(0)} = \text{diag}(d_i^{(0)})$ is updated depending on the HS option:

HS=0 specifies that no scaling be done.

HS=1 specifies the Moré (1978) scaling update:

$$d_i^{(k+1)} = \max\left[d_i^{(k)}, \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)}\right]$$

HS=2 specifies the Dennis, Gay, and Welsch (1981) scaling update:

$$d_i^{(k+1)} = \max\left[0.6 * d_i^{(k)}, \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)}\right]$$

HS=3 specifies that d_i be reset in each iteration:

$$d_i^{(k+1)} = \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)}$$

In each scaling update, ϵ is the relative machine precision. The default value is HS=0. Scaling of the Hessian can be time-consuming in the case where general linear constraints are active.

INHESSIAN< =*r* >

INHESS< =*r* >

specifies how the initial estimate of the approximate Hessian is defined for the quasi-Newton techniques QUANEW and DBLDOG. There are two alternatives:

- If you do not use the r specification, the initial estimate of the approximate Hessian is set to the Hessian at $\psi^{(0)}$.
- If you do use the r specification, the initial estimate of the approximate Hessian is set to the multiple of the identity matrix $r\mathbf{I}$.

By default, if you do not specify the option INHESSIAN=r, the initial estimate of the approximate Hessian is set to the multiple of the identity matrix $r\mathbf{I}$, where the scalar r is computed from the magnitude of the initial gradient.

INSTEP=*r*

SALPHA=*r*

RADIUS=*r*

reduces the length of the first trial step during the line search of the first iterations. For highly nonlinear objective functions, such as the EXP function, the default initial radius of the trust-region algorithm TRUREG or DBLDOG or the default step length of the line-search algorithms can result in arithmetic overflows. If this occurs, you should specify decreasing values of $0 < r < 1$ such as INSTEP=1E−1, INSTEP=1E−2, INSTEP=1E−4, and so on, until the iteration starts successfully.

- For trust-region algorithms (TRUREG, DBLDOG), the INSTEP= option specifies a factor $r > 0$ for the initial radius $\Delta^{(0)}$ of the trust region. The default initial trust-region radius is the length of the scaled gradient. This step corresponds to the default radius factor of $r = 1$.
- For line-search algorithms (NEWRAP, CONGRA, QUANEW), the INSTEP= option specifies an upper bound for the initial step length for the line search during the first five iterations. The default initial step length is $r = 1$.
- For the Nelder-Mead simplex algorithm, by using TECH=NMSIMP, the INSTEP=r option defines the size of the start simplex.

LCDEACT=*r*

LCD=*r*

specifies a threshold r for the Lagrange multiplier that determines whether an active inequality constraint remains active or can be deactivated. For maximization, r must be greater than zero; for minimization, r must be smaller than zero. An active inequality constraint can be deactivated only if its Lagrange multiplier is less than the threshold value. The default value is

$$r = \pm \min(0.01, \max(0.1 \times \text{ABSGCONV}, 0.001 \times \text{gmax}^{(k)}))$$

where "+" stands for maximization, "-" stands for minimization, ABSGCONV is the value of the absolute gradient criterion, and $\text{gmax}^{(k)}$ is the maximum absolute element of the gradient or the projected gradient.

LCEPSILON=*r* ≥ 0

LCEPS=*r* ≥ 0

LCE=*r* ≥ 0

specifies the range for active and violated boundary constraints. If the point $\boldsymbol{\psi}^{(k)}$ satisfies the condition

$$|\sum_{j=1}^{k} a_{ij}\psi_j^{(k)} - b_i| \leq r \times (|b_i| + 1)$$

the constraint i is recognized as an active constraint. Otherwise, the constraint i is either an inactive inequality or a violated inequality or equality constraint. The default value is r =1E−8. During the optimization process, the introduction of rounding errors can force the optimization to increase the value of r by a factor of $10, 100, \ldots$. If this happens, it is indicated by a message displayed in the log.

LCSINGULAR=*r* ≥ 0

LCSING=*r* ≥ 0

LCS=*r* $\geq$

specifies a criterion r, used in the update of the QR decomposition, that determines whether an active constraint is linearly dependent on a set of other active constraints. The default value is r =1E−8. The larger r becomes, the more the active constraints are recognized as being linearly dependent. If the value of r is larger than 0.1, it is reset to 0.1.

LINESEARCH=*i*

LIS=*i*

specifies the line-search method for the CONGRA, QUANEW, and NEWRAP optimization techniques. See Fletcher (1987) for an introduction to line-search techniques. The value of i can be $1, \ldots, 8$. For CONGRA, QUANEW, and NEWRAP, the default value is $i = 2$.

LIS=1 specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is similar to one used by the Harwell subroutine library.

LIS=2 specifies a line-search method that needs more function than gradient calls for quadratic and cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the LSPRECISION= option.

LIS=3 specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the LSPRECISION= option.

LIS=4 specifies a line-search method that needs the same number of function and gradient calls for stepwise extrapolation and cubic interpolation.

LIS=5 specifies a line-search method that is a modified version of LIS=4.

LIS=6 specifies golden section line search (Polak 1971), which uses only function values for linear approximation.

LIS=7 specifies bisection line search (Polak 1971), which uses only function values for linear approximation.

LIS=8 specifies the Armijo line-search technique (Polak 1971), which uses only function values for linear approximation.

LSPRECISION=*r*

LSP=*r*

specifies the degree of accuracy that should be obtained by the line-search algorithms LIS=2 and LIS=3. Usually an imprecise line search is inexpensive and successful. For more difficult optimization problems, a more precise and expensive line search might be necessary (Fletcher 1987). The second line-search method (which is the default for the NEWRAP, QUANEW, and CONGRA techniques) and the third line-search method approach exact line search for small LSPRECISION= values. If you have numerical problems, you should try to decrease the LSPRECISION= value to obtain a more precise line search. The default values are shown in Table 18.11.

Table 18.11 Default Values for Line-Search Precision

TECH=	UPDATE=	LSP Default
QUANEW	DBFGS, BFGS	$r = 0.4$
QUANEW	DDFP, DFP	$r = 0.06$
CONGRA	all	$r = 0.1$
NEWRAP	no update	$r = 0.9$

For more details, see Fletcher (1987).

MAXFUNC=*i*

MAXFU=*i*

specifies the maximum number i of function calls in the optimization process. The default values are as follows:

- TRUREG, NRRIDG, NEWRAP, LEVMAR: 125
- QUANEW, DBLDOG: 500
- CONGRA: 1000
- NMSIMP: 3000

Note that the optimization can terminate only after completing a full iteration. Therefore, the number of function calls that are actually performed can exceed the number that is specified by the MAXFUNC= option.

MAXITER=*i*

MAXIT=*i*

specifies the maximum number i of iterations in the optimization process. The default values are as follows:

- TRUREG, NRRIDG, NEWRAP, LEVMAR: 50

- QUANEW, DBLDOG: 200
- CONGRA: 400
- NMSIMP: 1000

These default values are also valid when *i* is specified as a missing value.

MAXSTEP=*r*<*n*>
specifies an upper bound for the step length of the line-search algorithms during the first *n* iterations. By default, *r* is the largest double-precision value and *n* is the largest integer available. Setting this option can improve the speed of convergence for the CONGRA, QUANEW, and NEWRAP techniques.

MAXTIME=*r*
specifies an upper limit of *r* seconds of CPU time for the optimization process. The default value is the largest floating-point double representation of your computer. Note that the time specified by the MAXTIME= option is checked only once at the end of each iteration. Therefore, the actual running time can be much longer than that specified by the MAXTIME= option. The actual running time includes the rest of the time needed to finish the iteration and the time needed to generate the output of the results.

MINITER=*i*
MINIT=*i*
specifies the minimum number of iterations. The default value is 0. If you request more iterations than are actually needed for convergence to a stationary point, the optimization algorithms can behave strangely. For example, the effect of rounding errors can prevent the algorithm from continuing for the required number of iterations.

MSINGULAR=*r* > 0
MSING=*r* > 0
specifies a relative singularity criterion for the computation of the inertia (number of positive, negative, and zero eigenvalues) of the Hessian and its projected forms. The default value is 1E−12.

NOPRINT
suppresses output related to optimization such as the iteration history. The GLIMMIX and HPMIXED procedures do not support printing options in the NLOPTIONS statement.

PALL
displays all optional output for optimization. This option is supported only by the TCALIS procedure.

PHISTORY
PHIST
displays the optimization history. The PHISTORY option is implied if the PALL option is specified. The PHISTORY option is supported only by the TCALIS procedure.

RESTART=*i > 0*

REST=*i > 0*

specifies that the QUANEW or CONGRA algorithm is restarted with a steepest descent/ascent search direction after, at most, i iterations. Default values are as follows:

- CONGRA: UPDATE=PB: restart is performed automatically, i is not used.
- CONGRA: UPDATE≠PB: $i = \min(10p, 80)$, where p is the number of parameters.
- QUANEW: i is the largest integer available.

SINGULAR=*r*

SING=*r*

specifies the singularity criterion r, $r0 \leq r \leq 1$, that is used for the inversion of the Hessian matrix. The default value is 1E−8.

SOCKET=*fileref*

specifies the fileref that contains the information needed for remote monitoring. See the section "Remote Monitoring" on page 403 for more details.

TECHNIQUE=*value*

TECH=*value*

OMETHOD=*value*

OM=*value*

specifies the optimization technique. You can find additional information about choosing an optimization technique in the section "Choosing an Optimization Algorithm" on page 405. Valid values for the TECHNIQUE= option are as follows:

- CONGRA
  performs a conjugate-gradient optimization, which can be more precisely specified with the UPDATE= option and modified with the LINESEARCH= option. When you specify this option, UPDATE=PB by default.
- DBLDOG
  performs a version of double-dogleg optimization, which can be more precisely specified with the UPDATE= option. When you specify this option, UPDATE=DBFGS by default.
- LEVMAR
  performs a highly stable but, for large problems, memory- and time-consuming Levenberg-Marquardt optimization technique, a slightly improved variant of the Moré (1978) implementation. You can also specify this technique with the alias LM or MARQUARDT. In the TCALIS procedure, this is the default optimization technique if there are fewer than 40 parameters to estimate. The GLIMMIX and HPMIXED procedures do not support this optimization technique.
- NMSIMP
  performs a Nelder-Mead simplex optimization. The TCALIS procedure does not support this optimization technique.
- NONE
  does not perform any optimization. This option can be used for the following:

  - to perform a grid search without optimization
  - to compute estimates and predictions that cannot be obtained efficiently with any of the optimization techniques
  - to obtain inferences for known values of the covariance parameters
- NEWRAP
  performs a Newton-Raphson optimization combining a line-search algorithm with ridging. The line-search algorithm LIS=2 is the default method.
- NRRIDG
  performs a Newton-Raphson optimization with ridging.
- QUANEW
  performs a quasi-Newton optimization, which can be defined more precisely with the UPDATE= option and modified with the LINESEARCH= option.
- TRUREG
  performs a trust-region optimization.

UPDATE=*method*
UPD=*method*

specifies the update method for the quasi-Newton, double-dogleg, or conjugate-gradient optimization technique. Not every update method can be used with each optimizer.

The following are the valid methods for the UPDATE= option:

- BFGS
  performs the original Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update of the inverse Hessian matrix.
- DBFGS
  performs the dual BFGS update of the Cholesky factor of the Hessian matrix. This is the default update method.
- DDFP
  performs the dual Davidon, Fletcher, and Powell (DFP) update of the Cholesky factor of the Hessian matrix.
- DFP
  performs the original DFP update of the inverse Hessian matrix.
- PB
  performs the automatic restart update method of Powell (1977) and Beale (1972).
- FR
  performs the Fletcher-Reeves update (Fletcher 1987).
- PR
  performs the Polak-Ribiere update (Fletcher 1987).
- CD
  performs a conjugate-descent update of Fletcher (1987).

VERSION=1 | 2
VS=1 | 2

specifies the version of the quasi-Newton optimization technique with nonlinear constraints.

VS=1 specifies the update of the μ vector as in Powell (1978a, 1978b) (update like VF02AD).

VS=2 specifies the update of the μ vector as in Powell (1982a, 1982b) (update like VMCWD).

The default is VERSION=2.

VSINGULAR=*r* > 0

VSING=*r* > 0

specifies a relative singularity criterion for the computation of the inertia (number of positive, negative, and zero eigenvalues) of the Hessian and its projected forms. The default value is $r = 1\text{E}{-8}$.

XCONV=*r***<** *n***>**

XTOL=*r***<** *n***>**

specifies the relative parameter convergence criterion. For all techniques except NMSIMP, termination requires a small relative parameter change in subsequent iterations:

$$\frac{\max_j |\psi_j^{(k)} - \psi_j^{(k-1)}|}{\max(|\psi_j^{(k)}|, |\psi_j^{(k-1)}|, \text{XSIZE})} \leq r$$

For the NMSIMP technique, the same formula is used, but $\psi_j^{(k)}$ is defined as the vertex with the lowest function value and $\psi_j^{(k-1)}$ is defined as the vertex with the highest function value in the simplex.

The default value is $r = 1\text{E}{-8}$ for the NMSIMP technique and $r = 0$ otherwise. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

XSIZE=*r*

specifies the XSIZE parameter of the relative parameter termination criterion, $r \geq 0$. The default value is $r = 0$. For more details, see the XCONV= option.

Remote Monitoring

SAS/EmMonitor is an application for Windows that enables you to monitor a CPU-intensive application running on a remote server. Remote monitoring is support through the NLOPTIONS statement.

On the server side, a FILENAME statement assigns a fileref to a SOCKET-type device that defines the IP address of the client and the port number for listening. The fileref is then specified in the SOCKET= option in the NLOPTIONS statement to control the SAS/EmMonitor application. The following statements show an example of server-side code:

```
filename sock socket 'your.pc.address.com:6943';

proc glimmix data=bigdataset;
   class year;
   model y1 = x1 x2 x3 / dist=gamma link=log;
   random x1 x3 / type=rsmooth knotmethod=kdtree(bucket=8)
                  subject=year;
   nloptions tech=nrridg gconv=2.e-5 socket=sock;
run;
```

On the client side, the SAS/EmMonitor application is started with the following syntax:

EmMonitor *options*

The options are as follows:

-p port_number	defines the port number.
-t title	defines the title of the SAS/EmMonitor window.
-k	keeps the monitor alive when the iteration is completed.

The default port number is 6943.

By default, the PC client displays on its **Graph** tab the objective function of the optimization process as the first Plot Group. The largest absolute gradient can be monitored as the next Plot Group. You can intervene in the optimization process through the **Stop Current** and **Stop All** buttons. Signaling the server by clicking these buttons effectively terminates the optimization. In a singly iterative optimization, clicking either of the two buttons halts the optimization process when the signal is received on the server side. The current parameter estimates are accepted, and postoptimization processing is based on these values. Note that this is different from a condition where the usual termination criterion is not met, such as when the maximum number of iterations is exceeded. If an optimization does not meet the termination criterion, no postprocessing occurs.

In doubly iterative processes, the **Stop All** button terminates the overall optimization process and accepts the current estimates. The **Stop Current** button stops the current optimization and uses the current parameter estimates to start the next optimization.

The server does not need to be running when you start the SAS/EmMonitor application, and you can start or stop it at any time during the iteration process. You need to remember only the port number.

If you do not start the PC client, or if you close it prematurely, it will have no effect on the server side. In other words, the iteration process will continue until one of the criteria for termination is met.

Choosing an Optimization Algorithm

First- or Second-Order Algorithms

The factors that go into choosing a particular optimization technique for a particular problem are complex. Trial and error can be involved.

For many optimization problems, computing the gradient takes more computer time than computing the function value. Computing the Hessian sometimes takes *much* more computer time and memory than computing the gradient, especially when there are many decision variables. Unfortunately, optimization techniques that do not use some kind of Hessian approximation usually require many more iterations than techniques that do use a Hessian matrix, and, as a result, the total run time of these techniques is often longer. Techniques that do not use the Hessian also tend to be less reliable. For example, they can terminate more easily at stationary points than at global optima.

Table 18.12 shows which derivatives are required for each optimization technique (FOD: first-order derivatives (=gradient evaluation); SOD: second-order derivatives (=Hessian evaluation)).

Table 18.12 First-Order and Second-Order Derivatives

Algorithm	FOD	SOD
LEVMAR	x	x
TRUREG	x	x
NEWRAP	x	x
NRRIDG	x	x
QUANEW	x	-
DBLDOG	x	-
CONGRA	x	-
NMSIMP	-	-

The second-derivative methods TRUREG, NEWRAP, and NRRIDG are best for small problems where the Hessian matrix is not expensive to compute. Sometimes the NRRIDG algorithm can be faster than the TRUREG algorithm, but TRUREG can be more stable. The NRRIDG algorithm requires only one matrix with $p(p+1)/2$ double words; TRUREG and NEWRAP require two such matrices. Here, p denotes the number of parameters in the optimization.

The first-derivative methods QUANEW and DBLDOG are best for medium-sized problems where the objective function and the gradient are much faster to evaluate than the Hessian. The QUANEW and DBLDOG algorithms, in general, require more iterations than TRUREG, NRRIDG, and NEWRAP, but each iteration can be much faster. The QUANEW and DBLDOG algorithms require only the gradient to update an approximate Hessian, and they require slightly less memory than TRUREG or NEWRAP (essentially one matrix with $p(p+1)/2$ double words).

The first-derivative method CONGRA is best for large problems where the objective function and the gradient can be computed much faster than the Hessian and where too much memory is required to store the (approximate) Hessian. The CONGRA algorithm, in general, requires more iterations than QUANEW or DBLDOG, but each iteration can be much faster. Because CONGRA requires

only a factor of p double-word memory, many large applications can be solved only by CONGRA.

The no-derivative method NMSIMP is best for small problems where derivatives are not continuous or are very difficult to compute.

Each optimization method employs one or more convergence criteria that determine when it has converged. An algorithm is considered to have converged when any one of the convergence criteria is satisfied. For example, under the default settings, the QUANEW algorithm will converge if ABSGCONV <1E−5, FCONV $< 10^{-FDIGITS}$, or GCONV <1E−8.

Algorithm Descriptions

Trust Region Optimization (TRUREG)

The trust region method uses the gradient $\mathbf{g}(\boldsymbol{\psi}^{(k)})$ and the Hessian matrix $\mathbf{H}(\boldsymbol{\psi}^{(k)})$; thus, it requires that the objective function $f(\boldsymbol{\psi})$ have continuous first- and second-order derivatives inside the feasible region.

The trust region method iteratively optimizes a quadratic approximation to the nonlinear objective function within a hyperelliptic trust region with radius Δ that constrains the step size corresponding to the quality of the quadratic approximation. The trust region method is implemented based on Dennis, Gay, and Welsch (1981), Gay (1983), and Moré and Sorensen (1983).

The trust region method performs well for small to medium-sized problems, and it does not need many function, gradient, and Hessian calls. However, if the computation of the Hessian matrix is computationally expensive, one of the (dual) quasi-Newton or conjugate gradient algorithms might be more efficient.

Newton-Raphson Optimization with Line Search (NEWRAP)

The NEWRAP technique uses the gradient $\mathbf{g}(\boldsymbol{\psi}^{(k)})$ and the Hessian matrix $\mathbf{H}(\boldsymbol{\psi}^{(k)})$; thus, it requires that the objective function have continuous first- and second-order derivatives inside the feasible region. If second-order derivatives are computed efficiently and precisely, the NEWRAP method can perform well for medium-sized to large problems, and it does not need many function, gradient, and Hessian calls.

This algorithm uses a pure Newton step when the Hessian is positive definite and when the Newton step reduces the value of the objective function successfully. Otherwise, a combination of ridging and line search is performed to compute successful steps. If the Hessian is not positive definite, a multiple of the identity matrix is added to the Hessian matrix to make it positive definite (Eskow and Schnabel 1991).

In each iteration, a line search is performed along the search direction to find an approximate optimum of the objective function. The default line-search method uses quadratic interpolation and cubic extrapolation (LIS=2).

Newton-Raphson Ridge Optimization (NRRIDG)

The NRRIDG technique uses the gradient $\mathbf{g}(\boldsymbol{\psi}^{(k)})$ and the Hessian matrix $\mathbf{H}(\boldsymbol{\psi}^{(k)})$; thus, it requires that the objective function have continuous first- and second-order derivatives inside the feasible region.

This algorithm uses a pure Newton step when the Hessian is positive definite and when the Newton step reduces the value of the objective function successfully. If at least one of these two conditions is not satisfied, a multiple of the identity matrix is added to the Hessian matrix.

The NRRIDG method performs well for small to medium-sized problems, and it does not require many function, gradient, and Hessian calls. However, if the computation of the Hessian matrix is computationally expensive, one of the (dual) quasi-Newton or conjugate gradient algorithms might be more efficient.

Because the NRRIDG technique uses an orthogonal decomposition of the approximate Hessian, each iteration of NRRIDG can be slower than that of the NEWRAP technique, which works with a Cholesky decomposition. Usually, however, NRRIDG requires fewer iterations than NEWRAP.

Quasi-Newton Optimization (QUANEW)

The (dual) quasi-Newton method uses the gradient $\mathbf{g}(\boldsymbol{\psi}^{(k)})$, and it does not need to compute second-order derivatives because they are approximated. It works well for medium-sized to moderately large optimization problems, where the objective function and the gradient are much faster to compute than the Hessian. However, in general, it requires more iterations than the TRUREG, NEWRAP, and NRRIDG techniques, which compute second-order derivatives. QUANEW is the default optimization algorithm because it provides an appropriate balance between the speed and stability required for most nonlinear mixed model applications.

The QUANEW technique is one of the following, depending upon the value of the UPDATE= option:

- the original quasi-Newton algorithm, which updates an approximation of the inverse Hessian
- the dual quasi-Newton algorithm, which updates the Cholesky factor of an approximate Hessian (default)

You can specify four update formulas with the UPDATE= option:

- DBFGS performs the dual Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update of the Cholesky factor of the Hessian matrix. This is the default.
- DDFP performs the dual Davidon, Fletcher, and Powell (DFP) update of the Cholesky factor of the Hessian matrix.
- BFGS performs the original BFGS update of the inverse Hessian matrix.
- DFP performs the original DFP update of the inverse Hessian matrix.

In each iteration, a line search is performed along the search direction to find an approximate optimum. The default line-search method uses quadratic interpolation and cubic extrapolation to obtain a step size α satisfying the Goldstein conditions. One of the Goldstein conditions can be violated if the feasible region defines an upper limit of the step size. Violating the left-side Goldstein condition can affect the positive definiteness of the quasi-Newton update. In that case, either the update is skipped or the iterations are restarted with an identity matrix, resulting in the steepest descent or ascent search direction. You can specify line-search algorithms other than the default with the LIS= option.

The QUANEW algorithm uses its own line-search technique. Not all options and parameters (except the INSTEP= option) controlling the line search in the other algorithms apply here. In several applications, large steps in the first iterations are troublesome. You can use the INSTEP= option to impose an upper bound for the step size α during the first five iterations. You can also use the INHESSIAN[=r] option to specify a different starting approximation for the Hessian. If you specify only the INHESSIAN option, the Cholesky factor of a (possibly ridged) finite-difference approximation of the Hessian is used to initialize the quasi-Newton update process.

Double-Dogleg Optimization (DBLDOG)

The double-dogleg optimization method combines the ideas of the quasi-Newton and trust region methods. In each iteration, the double-dogleg algorithm computes the step $\mathbf{s}^{(k)}$ as the linear combination of the steepest descent or ascent search direction $\mathbf{s}_1^{(k)}$ and a quasi-Newton search direction $\mathbf{s}_2^{(k)}$,

$$\mathbf{s}^{(k)} = \alpha_1 \mathbf{s}_1^{(k)} + \alpha_2 \mathbf{s}_2^{(k)}$$

The step is requested to remain within a prespecified trust region radius; see Fletcher (1987, p. 107). Thus, the DBLDOG subroutine uses the dual quasi-Newton update but does not perform a line search. You can specify two update formulas with the UPDATE= option:

- DBFGS performs the dual Broyden, Fletcher, Goldfarb, and Shanno update of the Cholesky factor of the Hessian matrix. This is the default.
- DDFP performs the dual Davidon, Fletcher, and Powell update of the Cholesky factor of the Hessian matrix.

The double-dogleg optimization technique works well for medium-sized to moderately large optimization problems, where the objective function and the gradient are much faster to compute than the Hessian. The implementation is based on Dennis and Mei (1979) and Gay (1983), but it is extended for dealing with boundary and linear constraints. The DBLDOG technique generally requires more iterations than the TRUREG, NEWRAP, or NRRIDG technique, which requires second-order derivatives; however, each of the DBLDOG iterations is computationally cheap. Furthermore, the DBLDOG technique requires only gradient calls for the update of the Cholesky factor of an approximate Hessian.

Conjugate Gradient Optimization (CONGRA)

Second-order derivatives are not required by the CONGRA algorithm and are not even approximated. The CONGRA algorithm can be expensive in function and gradient calls, but it requires only

$O(p)$ memory for unconstrained optimization. In general, many iterations are required to obtain a precise solution, but each of the CONGRA iterations is computationally cheap. You can specify four different update formulas for generating the conjugate directions by using the UPDATE= option:

- PB performs the automatic restart update method of Powell (1977) and Beale (1972). This is the default.
- FR performs the Fletcher-Reeves update (Fletcher 1987).
- PR performs the Polak-Ribiere update (Fletcher 1987).
- CD performs a conjugate-descent update of Fletcher (1987).

The default often behaves best for typical examples, whereas UPDATE=CD can perform poorly.

The CONGRA subroutine should be used for optimization problems with large p. For the unconstrained or boundary-constrained case, CONGRA requires only $O(p)$ bytes of working memory, whereas all other optimization methods require order $O(p^2)$ bytes of working memory. During p successive iterations, uninterrupted by restarts or changes in the working set, the conjugate gradient algorithm computes a cycle of p conjugate search directions. In each iteration, a line search is performed along the search direction to find an approximate optimum of the objective function. The default line-search method uses quadratic interpolation and cubic extrapolation to obtain a step size α satisfying the Goldstein conditions. One of the Goldstein conditions can be violated if the feasible region defines an upper limit for the step size. Other line-search algorithms can be specified with the LIS= option.

Nelder-Mead Simplex Optimization (NMSIMP)

The Nelder-Mead simplex method does not use any derivatives and does not assume that the objective function has continuous derivatives. The objective function itself needs to be continuous. This technique is quite expensive in the number of function calls, and it might be unable to generate precise results for $p \gg 40$.

The original Nelder-Mead simplex algorithm is implemented and extended to boundary constraints. This algorithm does not compute the objective for infeasible points, but it changes the shape of the simplex adapting to the nonlinearities of the objective function, which contributes to an increased speed of convergence. It uses a special termination criterion.

Programming Statements

This section applies to the following procedures:
CALIS, GLIMMIX, MCMC, NLIN, NLMIXED, and TCALIS.

The majority of the modeling procedures in SAS/STAT can take advantage of the fact that the statistical model can easily be translated into programming syntax (statements and options). Several procedures, however, require additional flexibility in specifying models—for example, when the model contains general nonlinear functions, when it is necessary to specify complicated restrictions, or when user-supplied expressions need to be evaluated. These procedures, listed at the beginning of the section, support—in addition to the usual procedure statements and options—programming statements that can be used in the SAS DATA step.

The following are valid statements:

ABORT;
CALL *name* < **(** *expression* < **,** *expression . . .* > **)** >**;**
DELETE;
DO < *variable* **=** *expression*
< **TO** *expression*> < **BY** *expression*>
< **,** *expression* < **TO** *expression*> < **BY** *expression* > . . . >
>
< **WHILE** *expression* > < **UNTIL** *expression* >**;**
END;
GOTO *statement-label*;
IF *expression*;
IF *expression* **THEN** *program-statement*;
ELSE *program-statement*;
variable **=** *expression*;
variable **+** *expression*;
LINK *statement-label*;
PUT < *variable*> < **=** > < *...* >**;**
RETURN;
SELECT < **(** *expression* **)** >**;**
STOP;
SUBSTR(*variable, index, length* **) =** *expression*;
WHEN (*expression*) *program-statement*;
OTHERWISE *program-statement*;

For the most part, these programming statements work the same as they do in the SAS DATA step, as documented in *SAS Language Reference: Concepts*. However, there are several differences:

- The ABORT statement does not allow any arguments.
- The DO statement does not allow a character index variable. Thus

```
do i = 1,2,3;
```

is supported, whereas the following statement is not supported:

```
do i = 'A','B','C';
```

- Not all procedures support LAG functionality. For example, the GLIMMIX procedure does not support lags.
- The PUT statement, used mostly for program debugging, supports only some of the features of the DATA step PUT statement, and it has some features not available with the DATA step PUT statement:
  - The PUT statement does not support line pointers, factored lists, iteration factors, overprinting, _INFILE_, the colon (:) format modifier, or "$".
  - The PUT statement does support expressions, but the expression must be enclosed in parentheses. For example, the following statement displays the square root of x:

    ```
    put (sqrt(x));
    ```

  - The PUT statement supports the item _PDV_ to display a formatted listing of all variables in the program. For example:

    ```
    put _pdv_;
    ```

- The WHEN and OTHERWISE statements enable you to specify more than one target statement. That is, DO/END groups are not necessary for multiple-statement WHENs. For example, the following syntax is valid:

```
select;
   when (exp1) stmt1;
               stmt2;
   when (exp2) stmt3;
               stmt4;
end;
```

Please consult the individual chapters for other, procedure-specific differences between programming statements and the SAS DATA step and for procedure-specific details, limitations, and rules.

The LINK statement is used in a program to jump immediately to the label *statement_label* and to continue program execution at that point. It is not used to specify a link function in a generalized linear model.

When coding your programming statements, you should avoid defining variables that begin with an underscore (_), because they might conflict with internal variables created by procedures that support programming statements.

References

Beale, E. M. L. (1972), "A Derivation of Conjugate Gradients," in *Numerical Methods for Nonlinear Optimization*, ed. F. A. Lootsma, London: Academic Press.

Browne, M. W. (1982), "Covariance Structures," in *Topics in Multivariate Analyses*, ed. D. M. Hawkins, New York: Cambridge University Press.

Dennis, J. E., Gay, D. M., and Welsch, R. E. (1981), "An Adaptive Nonlinear Least-Squares Algorithm," *ACM Transactions on Mathematical Software*, 7, 348–368.

Dennis, J. E. and Mei, H. H. W. (1979), "Two New Unconstrained Optimization Algorithms Which Use Function and Gradient Values," *J. Optim. Theory Appl.*, 28, 453–482.

Eskow, E. and Schnabel, R. B. (1991), "Algorithm 695: Software for a New Modified Cholesky Factorization," *Transactions on Mathematical Software*, 17(3), 306–312.

Fletcher, R. (1987), *Practical Methods of Optimization,* Second Edition, Chichester: John Wiley & Sons.

Gay, D. M. (1983), "Subroutines for Unconstrained Minimization," *ACM Transactions on Mathematical Software*, 9, 503–524.

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning,* New York: Springer-Verlag.

Moré, J. J. (1978), "The Levenberg-Marquardt Algorithm: Implementation and Theory," in *Lecture Notes in Mathematics 630*, ed. G.A. Watson, Berlin-Heidelberg-New York: Springer Verlag.

Moré, J. J. and Sorensen, D. C. (1983), "Computing a Trust-Region Step," *SIAM Journal on Scientific and Statistical Computing*, 4, 553–572.

Polak, E. (1971), *Computational Methods in Optimization*, New York: Academic Press.

Powell, J. M. D. (1977), "Restart Procedures for the Conjugate Gradient Method," *Math. Prog.*, 12, 241–254.

Powell, J. M. D. (1978a), "A Fast Algorithm for Nonlinearly Constraint Optimization Calculations," in *Numerical Analysis, Dundee 1977, Lecture Notes in Mathematics 630*, ed. G. A. Watson, Berlin: Springer-Verlag, 144–175.

Powell, J. M. D. (1978b), "Algorithms for Nonlinear Constraints That Use Lagrangian Functions," *Mathematical Programming*, 14, 224–248.

Powell, J. M. D. (1982a), "Extensions to Subroutine VF02AD," in *Systems Modeling and Optimization, Lecture Notes in Control and Information Sciences 38*, ed. R. F. Drenick and F. Kozin, Berlin: Springer-Verlag, 529–538.

Powell, J. M. D. (1982b), "VMCWD: A Fortran Subroutine for Constrained Optimization," *DAMTP 1982/NA4*, Cambridge, England.

Chapter 19

Introduction to Power and Sample Size Analysis

Contents

Overview **413**
Coverage of Statistical Analyses 414
Statistical Background **415**
Hypothesis Testing, Power, and Confidence Interval Precision 415
Standard Hypothesis Tests 415
Equivalence and Noninferiority 416
Confidence Interval Precision 416
Computing Power and Sample Size 417
Power and Study Planning **418**
Components of Study Planning 418
Effect Size 419
Uncertainty and Sensitivity Analysis 419
SAS/STAT Tools for Power and Sample Size Analysis **419**
Basic Graphs (POWER, GLMPOWER, Power and Sample Size Application) 421
Highly Customized Graphs (POWER, GLMPOWER) 423
Formatted Tables (%POWTABLE Macro) 424
Narratives and Graphical User Interface (Power and Sample Size Application) 425
Customized Power Formulas (DATA Step) 428
Empirical Power Simulation (DATA Step, SAS/STAT Software) 429
References **431**

Overview

Power and sample size analysis optimizes the resource usage and design of a study, improving chances of conclusive results with maximum efficiency. The standard statistical testing paradigm implicitly assumes that Type I errors (mistakenly concluding significance when there is no true effect) are more costly than Type II errors (missing a truly significant result). This may be appropriate for your situation, or the relative costs of the two types of error may be reversed. For example, in screening experiments for drug development, it is often less damaging to carry a few false positives

forward for follow-up testing than to miss potential leads. Power and sample size analysis can help you achieve your desired balance between Type I and Type II errors. With optimal designs and sample sizes, you can improve your chances of detecting effects that might otherwise have been ignored, save money and time, and perhaps minimize risks to subjects.

Relevant tools in SAS/STAT software for power and sample size analysis include the following:

- the GLMPOWER procedure
- the POWER procedure
- the Power and Sample Size application
- the %POWTABLE macro
- various procedures, statements, and functions in Base SAS and SAS/STAT for developing customized formulas and simulations

These tools, discussed in detail in the section "SAS/STAT Tools for Power and Sample Size Analysis" on page 419, deal exclusively with *prospective* analysis—that is, planning for a future study. This is in contrast to *retrospective* analysis for a past study, which is not supported by the main tools. Although retrospective analysis is more convenient to perform, it is often uninformative or misleading, especially when power is computed directly based on observed data.

The goals of prospective power and sample size analysis include the following:

- determining the sample size required to get a significant result with adequate probability (power)
- characterizing the power of a study to detect a meaningful effect
- computing the probability of achieving the desired precision of a confidence interval, or the sample size required to ensure this probability
- conducting what-if analyses to assess how sensitive the power or required sample size is to other factors

The phrase *power analysis* is used for the remainder of this document as a shorthand to represent any or all of these goals. For more information about the GLMPOWER procedure, see Chapter 41, "The GLMPOWER Procedure." For more information about the POWER procedure, see Chapter 67, "The POWER Procedure." For more information about the Power and Sample Size application, see Chapter 68, "The Power and Sample Size Application."

Coverage of Statistical Analyses

The GLMPOWER procedure covers power analysis for Type III tests and contrasts of fixed effects in univariate linear models, optionally with covariates. The covariates can be continuous or categorical. Tests and contrasts involving random effects are not supported.

The POWER procedure covers power analysis for the following:

- t tests, equivalence tests, and confidence intervals for means
- tests, equivalence tests, and confidence intervals for binomial proportions
- multiple regression
- tests of correlation and partial correlation
- one-way analysis of variance
- rank tests for comparing two survival curves
- logistic regression with binary response
- Wilcoxon Mann-Whitney rank-sum test

The Power and Sample Size application covers a large subset of the analyses in the GLMPOWER and POWER procedures.

Statistical Background

Hypothesis Testing, Power, and Confidence Interval Precision

Standard Hypothesis Tests

In statistical hypothesis testing, you typically express the belief that some effect exists in a population by specifying an alternative hypothesis H_1. You state a null hypothesis H_0 as the assertion that the effect does *not* exist and attempt to gather evidence to reject H_0 in favor of H_1. Evidence is gathered in the form of sample data, and a statistical test is used to assess H_0. If H_0 is rejected but there really is *no* effect, this is called a *Type I error*. The probability of a Type I error is usually designated "alpha" or α, and statistical tests are designed to ensure that α is suitably small (for example, less than 0.05).

If there is an effect in the population but H_0 is *not* rejected in the statistical test, then a *Type II error* has been committed. The probability of a Type II error is usually designated "beta" or β. The probability $1-\beta$ of avoiding a Type II error (that is, correctly rejecting H_0 and achieving statistical significance) is called the *power* of the test.

Most, but not all, of the power analyses in the GLMPOWER and POWER procedures are based on such standard hypothesis tests.

Equivalence and Noninferiority

Whereas the standard two-sided hypothesis test for a parameter μ (such as a mean difference) aims to demonstrate that it is significantly different than a null value μ_0:

$$H_0{:}\mu = \mu_0$$
$$H_1{:}\mu \neq \mu_0$$

an equivalence test instead aims to demonstrate that it is *significantly similar* to some value, expressed in terms of a range θ_L, θ_U around that value:

$$H_0{:}\mu < \theta_L \quad \text{or} \quad \mu > \theta_U$$
$$H_1{:}\theta_L \leq \mu \leq \theta_U$$

Whereas the standard one-sided hypothesis test for μ (say, the upper one-sided test) aims to demonstrate that it is significantly greater than μ_0:

$$H_0{:}\mu \leq \mu_0$$
$$H_1{:}\mu > \mu_0$$

a corresponding noninferiority test aims to demonstrate that it is *not significantly less* than μ_0, expressed in terms of a margin $\delta > 0$:

$$H_0{:}\mu \leq \mu_0 - \delta$$
$$H_1{:}\mu > \mu_0 - \delta$$

Corresponding forms of these hypotheses with the inequalities reversed apply to lower one-sided noninferiority tests (sometimes called *nonsuperiority* tests).

The POWER procedure performs power analyses for equivalence tests for one-sample, paired, and two-sample tests of normal and lognormal mean differences and ratios. It also supports noninferiority tests for a variety of analyses of means, proportions, and correlation, both directly (with a MARGIN= option representing δ) and indirectly (with an option for a custom null value representing the sum or difference of μ_0 and δ).

Confidence Interval Precision

An analysis of confidence interval precision is analogous to a traditional power analysis, with *CI Half-Width* taking the place of effect size and *Prob(Width)* taking the place of power. The *CI Half-Width* is the margin of error associated with the confidence interval, the distance between the point estimate and an endpoint. The *Prob(Width)* is the probability of obtaining a confidence interval with *at most* a target half-width.

The POWER procedure performs confidence interval precision analyses for *t*-based confidence intervals for one-sample, paired, and two-sample designs, and for several varieties of confidence intervals for a binomial proportion.

Computing Power and Sample Size

For some statistical models and tests, power analysis calculations are exact—that is, they are based on a mathematically accurate formula that expresses power in terms of the other components. Such formulas typically involve either enumeration or noncentral versions of the distribution of the test statistic.

When a power computation is based on a noncentral t, F, or chi-square distribution, the noncentrality parameter generally has the same form as the test statistic, with the conjectured population parameters in place of their corresponding estimators.

For example, the test statistic for a two-sample t test is computed as follows:

$$t = N^{\frac{1}{2}}(w_1 w_2)^{\frac{1}{2}}\left(\frac{\bar{x}_2 - \bar{x}_1 - \mu_0}{s_p}\right)$$

where N is the total sample size, w_1 and w_2 are the group allocation weights, $\bar{x}_1$ and $\bar{x}_2$ are the sample means, μ_0 is the null mean difference, and s_p is the pooled standard deviation. Under the null hypothesis, the statistic $F = t^2$ is distributed as $F(1, N-2)$. In general, F has a noncentral F distribution $F(1, N-2, \delta^2)$ where

$$\delta = N^{\frac{1}{2}}(w_1 w_2)^{\frac{1}{2}}\left(\frac{\mu_{\text{diff}} - \mu_0}{\sigma}\right)$$

and μ_{diff} and σ are the (unknown) true mean difference and common group standard deviation, respectively. Note that the square-root noncentrality δ is exactly the same as the t statistic except that the estimators of mean difference and standard deviation are replaced by their correponding true population values.

The power for the two-sided two-sample t test with significance level α is computed as

$$P\left(F \geq F_{1-\alpha}(1, N-2)\right)$$

where F is distributed as $F(1, N-2, \delta^2)$ and $F_{1-\alpha}(1, N-2)$ is the $100(1-\alpha)\%$ quantile of the central F distribution with 1 and $N-2$ degrees of freedom. See the section "Customized Power Formulas (DATA Step)" on page 428 for an example of the implementation of this formula in the DATA step.

In the absence of exact mathematical results, approximate formulas can sometimes be used. When neither exact power computations nor reasonable approximations are possible, simulation provides an increasingly viable alternative. You specify values for model parameters and use them to randomly generate a large number of hypothetical data sets. Applying the statistical test to each data set, you estimate power with the percentage of times the null hypothesis is rejected. While the simulation approach is computationally intensive, faster computing makes this less of an issue. A simulation-based power analysis is always a valid option, and, with a large number of data set replications, it can often be more accurate than approximations. See the section "Empirical Power Simulation (DATA Step, SAS/STAT Software)" on page 429 for an example of an empirical power simulation.

Sample size is usually computed by iterative numerical methods because it often cannot be expressed in closed form as a function of the other parameters. Sample size tends to appear in both a noncentrality parameter and a degrees of freedom term for the critical value.

Power and Study Planning

Power analysis is most effective when performed at the study planning stage, and as such it encourages early collaboration between researcher and statistician. It also focuses attention on effect sizes and variability in the underlying scientific process, concepts that both researcher and statistician should consider carefully at this stage.

There are many factors involved in a power analysis, such as the research objective, design, data analysis method, power, sample size, Type I error, variability, and effect size. By performing a power analysis, you can learn about the relationships between these factors, optimizing those that are under your control and exploring the implications of those that are fixed or unknown.

Components of Study Planning

Even when the research questions and study design seem straightforward, the ensuing power analysis can seem technically daunting. It is often helpful to break the process down into five components:

- **Study Design**: What is the structure of the planned design? This must be clearly and completely specified. What groups and treatments ("cells" and "factors" of the design) are going to be assessed, and what will be the relative sizes of those cells? How is each case going to be studied—that is, what is the primary outcome measure ("dependent variable")? Will covariates be measured and included in the statistical model?
- **Scenario Model**: What are your beliefs about patterns in the data? Imagine that you had unlimited time and resources to execute the study design, so that you could gather an "infinite data set." Characterize that infinite data set as best you can using a mathematical model, realizing that it will be a simplification of reality. Alternatively, as is common with complex linear models, you may decide to construct an "exemplary" data set that mimics the infinite data set. However you do this, your scenario model should capture the key features of the study design and the main relationships among the primary outcome variables and study factors.
- **Effects and Variability**: What exactly are the "signals and noises" in the patterns you suspect? Set specific values for the parameters of your scenario model, keeping at most one unspecified. It is often enlightening to consider a variety of realistic possibilities for the key values by performing a sensitivity analysis, to explore the consequences of competing views on what the infinite data set might look like.
- **Statistical Method**: How will you cast your model in statistical terms and conduct the eventual data analysis? Define the statistical models and procedures that will be used to embody the study design and estimate and/or test the effects central to the research question. What significance levels will be used? Will one- or two-sided tests be used?
- **Aim of Assessment**: Finally, what needs to be determined in the power analysis? Most often you want to examine the statistical powers obtained across the various scenarios for

the effects, variability, alternative varieties of the statistical procedures to be used, and the feasible total sample sizes. Sometimes the goal is to find sample size values that provide given levels of power, say 85%, 90%, or 95%.

Effect Size

There is some confusion in practice about how to postulate the effect size. One alternative is to specify the effect size that represents minimal clinical significance; then the result of the power analysis reveals the chances of detecting a minimally meaningful effect size. Often this minimal effect size is so small that it requires excessive resources to detect. Another alternative is to make an educated guess of the true underlying effect size. Then the power analysis determines the chance of detecting the effect size that is believed to be true. The choice is ultimately determined by the research goals. Finally, you can specify a collection of possible values, perhaps spanning the range between minimally meaningful effects and larger surmised effects.

You can arrive at values for required quantities in a power analysis, such as effect sizes and measures of variability, in many different ways. For example, you can use pilot data, results of previous studies reported in literature, educated guesses derived from theory, or educated guesses derived from partial data (a small sample or even just quantiles).

Uncertainty and Sensitivity Analysis

Uncertainty is a fact of life in any power analysis, because at least some of the numbers used are best guesses of unknown values. The result of a power calculation, whether it be achieved power or required sample size or something else, serves only as a point estimate, conditional on the conjectured values of the other components. It is not feasible in general to quantify the variability involved in using educated guesses or undocumented results to specify these components. If observed data are used, relevant adjustments for variability in the data tend to be problematic in the sense of producing confidence intervals for power that are too wide for practical use. But there is a useful way for you to characterize the uncertainty in your power analysis, and also discover the extent to which statistical power is affected by each component. You can posit a reasonable range for each input component, vary each one within its range, and observe the variety of results in the form of tables or graphs.

SAS/STAT Tools for Power and Sample Size Analysis

This section demonstrates how you can use the different SAS power analysis tools mentioned in the section “Overview” on page 413 to generate graphs, tables, and narratives; implement your own power formulas; and simulate empirical power.

Suppose you want to compute the power of a two-sample *t* test. You conjecture that the mean difference is between 5 and 6 and that the common group standard deviation is between 12 and 18. You plan to use a significance level between 0.05 and 0.1 and a sample size between 100 and 200. The following SAS statements use the POWER procedure to compute the power for these scenarios:

```
proc power;
   twosamplemeans test=diff
      meandiff = 5 6
      stddev = 12 18
      alpha = 0.05 0.1
      ntotal = 100 200
      power = .;
run;
```

Figure 19.1 shows the results. Depending on the plausibility of the various combinations of input parameter values, the power ranges between 0.379 and 0.970.

Figure 19.1 PROC POWER Tabular Output

The POWER Procedure
Two-sample t Test for Mean Difference

Computed Power

Index	Alpha	Mean Diff	Std Dev	N Total	Power
1	0.05	5	12	100	0.541
2	0.05	5	12	200	0.834
3	0.05	5	18	100	0.280
4	0.05	5	18	200	0.498
5	0.05	6	12	100	0.697
6	0.05	6	12	200	0.940
7	0.05	6	18	100	0.379
8	0.05	6	18	200	0.650
9	0.10	5	12	100	0.664
10	0.10	5	12	200	0.902
11	0.10	5	18	100	0.397
12	0.10	5	18	200	0.623
13	0.10	6	12	100	0.799
14	0.10	6	12	200	0.970
15	0.10	6	18	100	0.505
16	0.10	6	18	200	0.759

The following seven sections illustrate additional ways of displaying these results using the different SAS tools.

Basic Graphs (POWER, GLMPOWER, Power and Sample Size Application)

If you include a PLOT statement, the GLMPOWER and POWER procedures produce standard power curves, representing any multi-valued input parameters with varying line styles, symbols, colors, and/or panels. The Power and Sample Size application also has an option to produce power curves.

To display default power curves for the preceding PROC POWER call, simply add the PLOT statement with no arguments as follows:

```
proc power plotonly;
   twosamplemeans test=diff
      meandiff = 5 6
      stddev = 12 18
      alpha = 0.05 0.1
      ntotal = 100 200
      power = .;
   plot;
run;
```

Figure 19.2 shows the results. Note that the line style varies by the significance level α, the symbol varies by the mean difference, and the panel varies by standard deviation.

Figure 19.2 PROC POWER Default Graphical Output

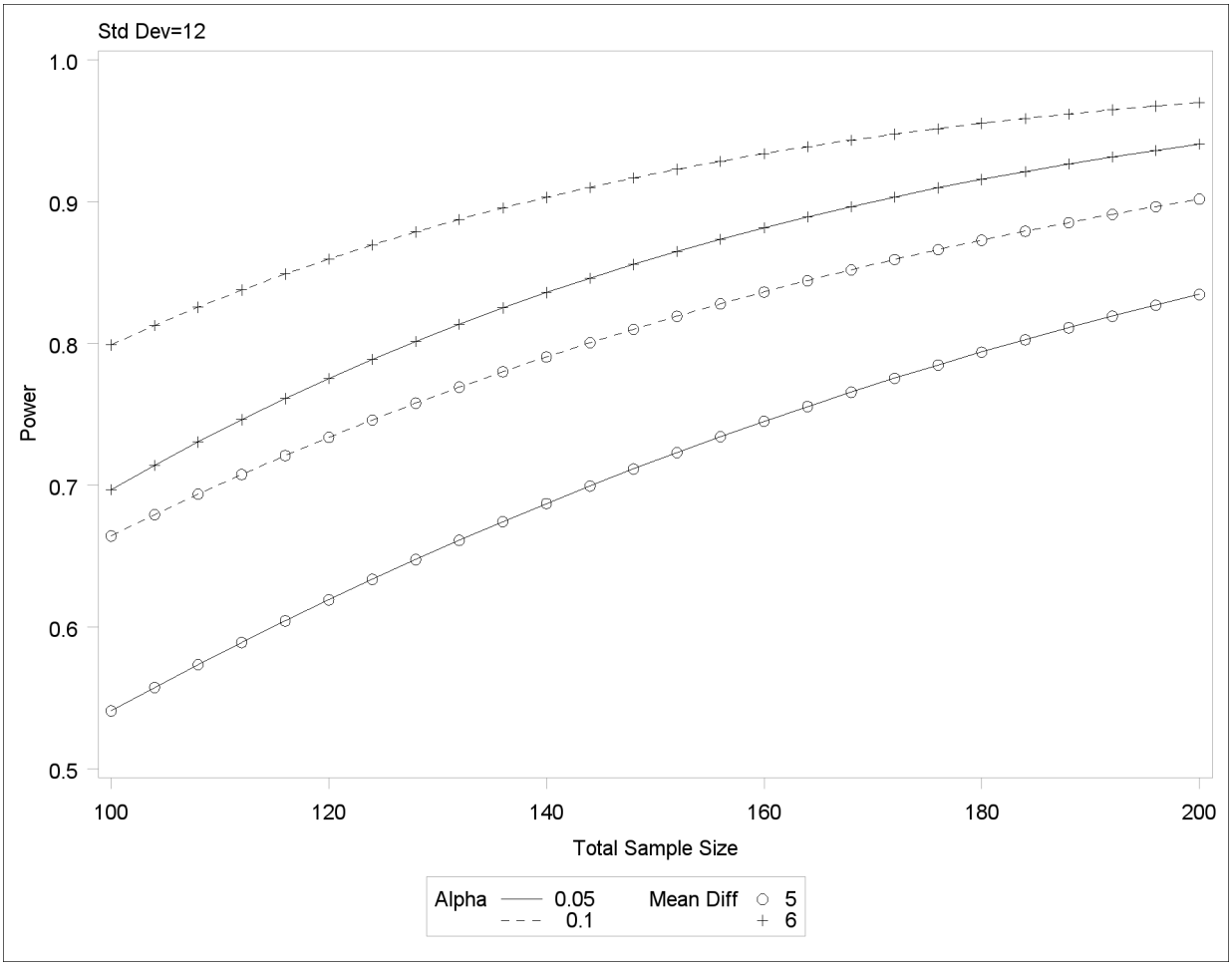
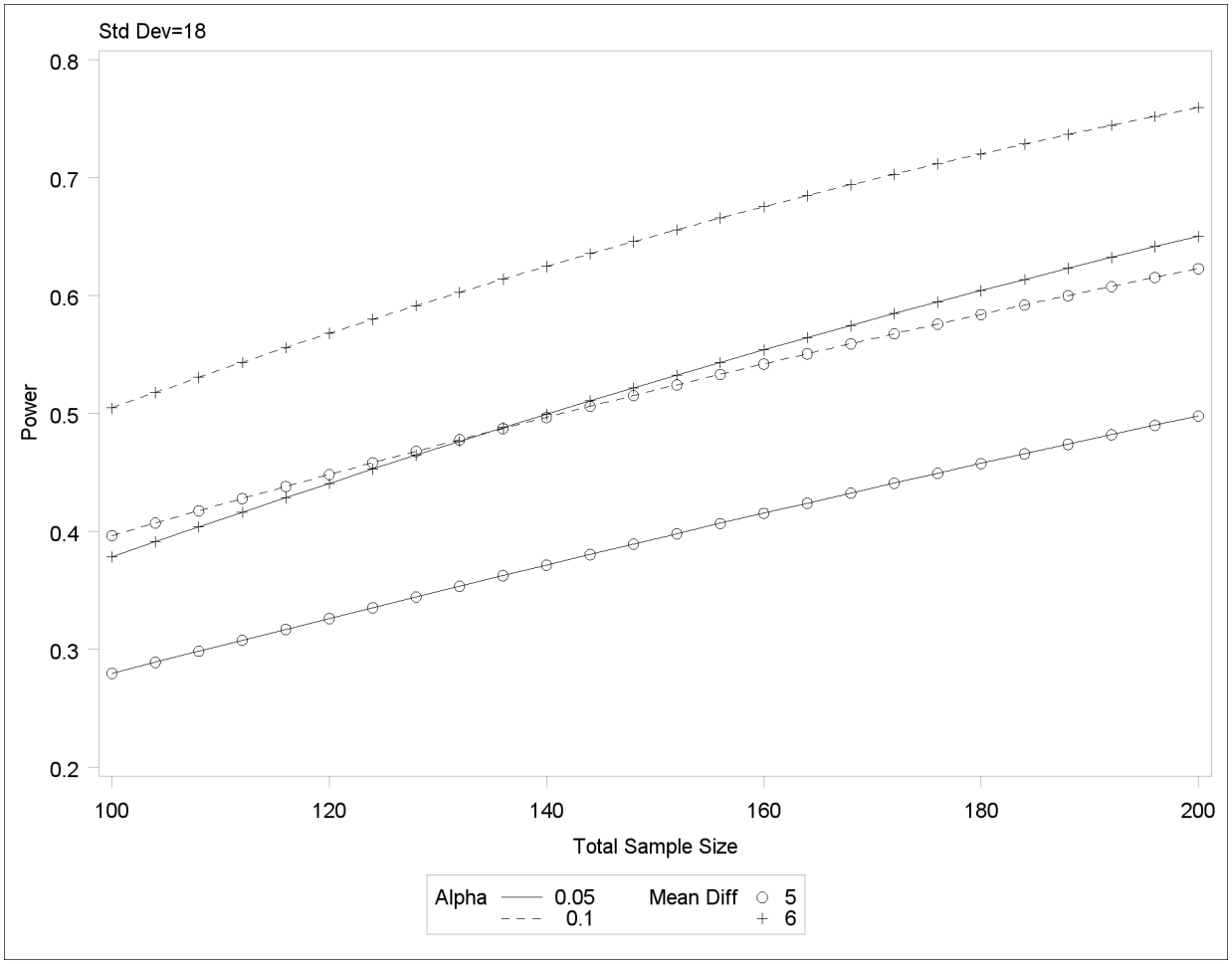


Figure 19.2 *continued*



Highly Customized Graphs (POWER, GLMPOWER)

Example 67.8 of Chapter 67, "The POWER Procedure," demonstrates various ways you can modify and enhance plots created in the GLMPOWER or POWER procedures:

- assigning analysis parameters to axes
- fine-tuning a sample size axis
- adding reference lines
- linking plot features to analysis parameters
- choosing key (legend) styles
- modifying symbol locations

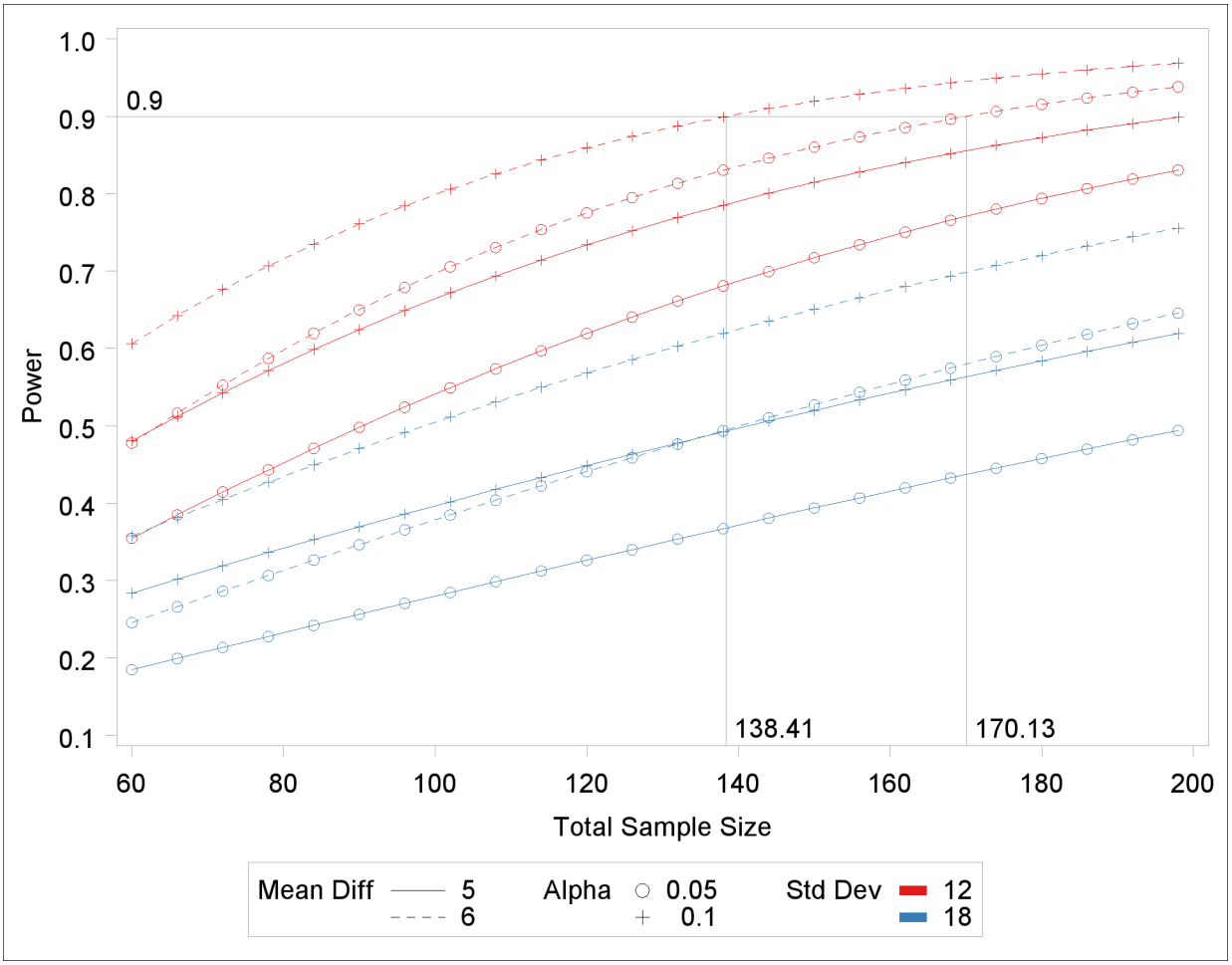
For example, replace the default PLOT statement with the following statement to modify the graphical results in Figure 19.2 to lower the minimum sample size to 60, show a reference line at

power=0.9 with corresponding sample size values, distinguish standard deviation by color instead of panel, and swap the roles of α and mean difference:

```
plot
   min=60
   yopts=(ref=0.9 crossref=yes)
   vary(color by stddev, linestyle by meandiff, symbol by alpha);
```

Figure 19.3 shows the results. The plot reveals that only the scenarios with the largest mean difference and smallest standard deviation achieve a power of at least 0.9 for this sample size range.

Figure 19.3 PROC POWER Customized Graphical Output



Formatted Tables (%POWTABLE Macro)

The %POWTABLE macro renders the output of the POWER and GLMPOWER procedures in rectangular form, and it optionally produces simplified results using weighted means across chosen variables. PROC REPORT and the Output Delivery System (ODS) are used to generate the tables. Base SAS and SAS/STAT 9.1 or higher versions are required.

You can run the %POWTABLE macro for the output in Figure 19.1 to display the results in a form more suitable for quickly discerning relationships among parameters. First use the ODS OUTPUT statement to assign the "Output" table produced by the POWER procedure to a data set as follows:

```
ods output output=powdata;
```

Next, specify the same PROC POWER statements that generate Figure 19.1. Finally, use the %POWTABLE macro to assign analysis parameters to table dimensions. To create a table of computed power values with mean difference assigned to rows, sample size and α assigned to columns, and standard deviation assigned to "panels" (rendered by default as rows separated by blank lines), specify the following statements:

```
%powtable ( Data = powdata,
            Entries = power,
            Rows = meandiff,
            Cols = ntotal alpha,
            Panels = stddev )
```

Figure 19.4 shows the results.

Figure 19.4 %POWTABLE Macro Output

The POWTABLE Macro

Entries are Power

		N Total			
		100		200	
Std Dev	Mean Diff	Alpha 0.05	Alpha 0.10	Alpha 0.05	Alpha 0.10
12	5	0.541	0.664	0.834	0.902
	6	0.697	0.799	0.940	0.970
18	5	0.280	0.397	0.498	0.623
	6	0.379	0.505	0.650	0.759

Narratives and Graphical User Interface (Power and Sample Size Application)

The Power and Sample Size application produces narratives for the results. Narratives are descriptions of the input parameters and a statement about the computed power or sample size.

For example, the Power and Sample Size application creates the following narrative for the scenario corresponding to the first row in Figure 19.1:

"For a two-sample pooled t test of a normal mean difference with a two-sided significance level of 0.05, assuming a common standard deviation of 12, a total sample size of 100 assuming a balanced

design has a power of 0.541 to detect a mean difference of 5."

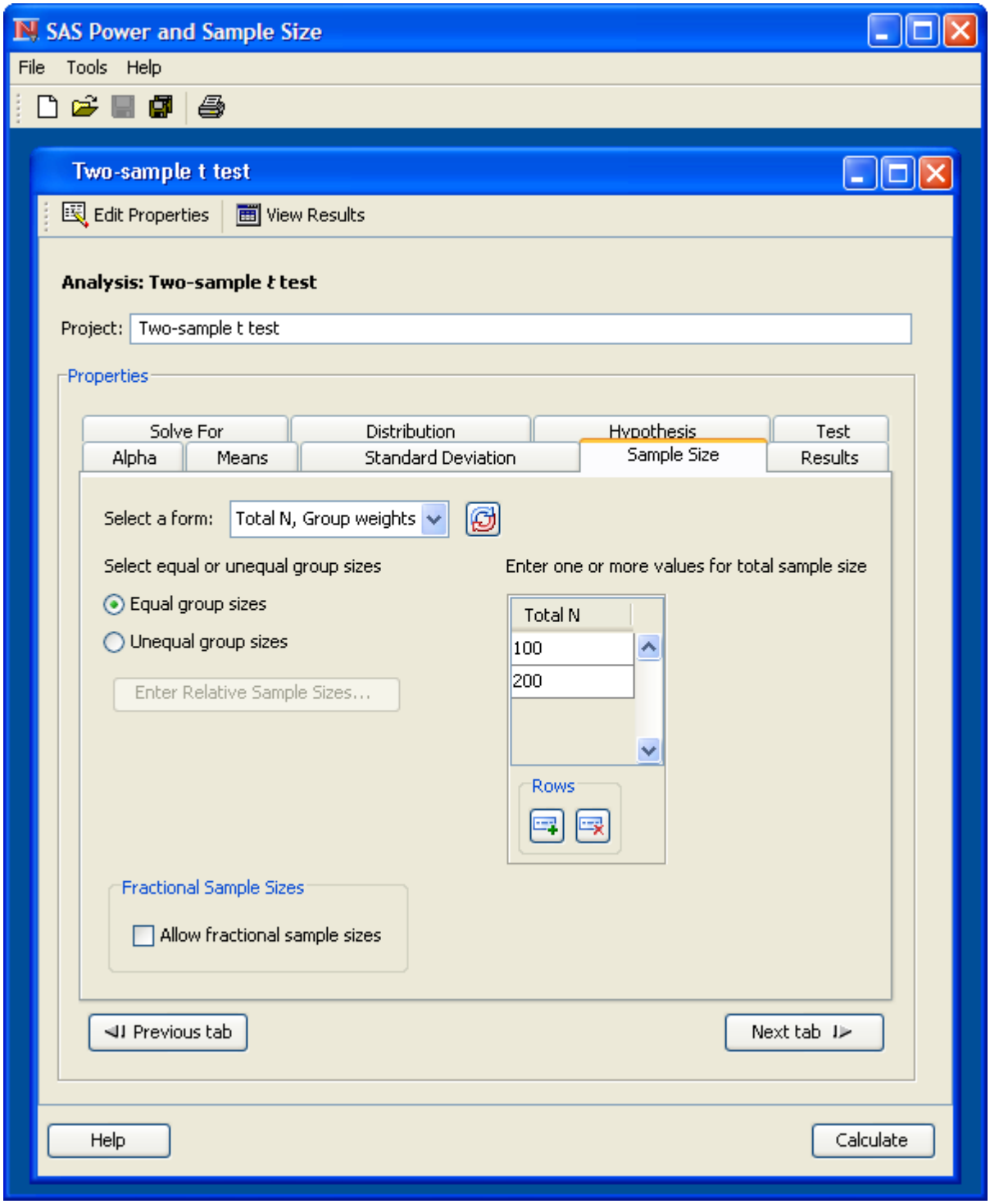
The Power and Sample Size application also provides multiple input parameter options, stores the results in a project format, displays power curves, and shows the SAS log and SAS code. You can access each project to review the results or to edit your input parameters and produce another analysis.

Where appropriate, several alternate ways of entering values for certain parameters are offered. For example, in the two-sample *t* test analysis, sample sizes can be entered in any of several parameterizations:

- total sample size in a balanced design
- sample size per group in a balanced design
- total sample size and group allocation weights
- groupwise sample sizes

See Figure 19.5 for an illustration of the application, showing the sample size input page for a two-sample *t* test.

Figure 19.5 Power and Sample Size Application



Customized Power Formulas (DATA Step)

If you want to perform a power compuation for an analysis that is not currently supported directly in SAS/STAT tools, and you have a power formula, then you can program the formula in the DATA step.

For purposes of illustration, here is the power formula in the section "Computing Power and Sample Size" on page 417 implemented in the DATA step to compute power for the *t* test example:

```
data tpow;
   do meandiff = 5, 6;
      do stddev = 12, 18;
         do alpha = 0.05, 0.1;
            do ntotal = 100, 200;
               ncp = ntotal * 0.5 * 0.5 * meandiff**2 / stddev**2;
               critval = finv(1-alpha, 1, ntotal-2, 0);
               power = sdf('f', critval, 1, ntotal-2, ncp);
               output;
            end;
         end;
      end;
   end;
run;

proc print data=tpow;
run;
```

The output is shown in Figure 19.6.

Figure 19.6 Customized Power Formula (DATA Step)

Obs	meandiff	stddev	alpha	ntotal	ncp	critval	power
1	5	12	0.05	100	4.3403	3.93811	0.54102
2	5	12	0.05	200	8.6806	3.88885	0.83447
3	5	12	0.10	100	4.3403	2.75743	0.66434
4	5	12	0.10	200	8.6806	2.73104	0.90171
5	5	18	0.05	100	1.9290	3.93811	0.27981
6	5	18	0.05	200	3.8580	3.88885	0.49793
7	5	18	0.10	100	1.9290	2.75743	0.39654
8	5	18	0.10	200	3.8580	2.73104	0.62287
9	6	12	0.05	100	6.2500	3.93811	0.69689
10	6	12	0.05	200	12.5000	3.88885	0.94043
11	6	12	0.10	100	6.2500	2.75743	0.79895
12	6	12	0.10	200	12.5000	2.73104	0.96985
13	6	18	0.05	100	2.7778	3.93811	0.37857
14	6	18	0.05	200	5.5556	3.88885	0.65012
15	6	18	0.10	100	2.7778	2.75743	0.50459
16	6	18	0.10	200	5.5556	2.73104	0.75935

Empirical Power Simulation (DATA Step, SAS/STAT Software)

You can obtain a highly accurate power estimate by simulating the power empirically. You need to use this approach for analyses that are not supported directly in SAS/STAT tools and for which you lack a power formula. But the simulation approach is also a viable alternative to existing power approximations. A high number of simulations will yield a more accurate estimate than a non-exact power approximation.

Although exact power computations for the two-sample t test are supported in several of the SAS/STAT tools, suppose for purposes of illustration that you want to simulate power for the continuing t test example. This section describes how you can use the DATA step and SAS/STAT software to do this.

The simulation involves generating a large number of data sets according to the distributions defined by the power analysis input parameters, computing the relevant p-value for each data set, and then estimating the power as the proportion of times that the p-value is significant.

The following statements compute a power estimate along with a 95% confidence interval for power for the first scenario in the two-sample t test example, with 10,000 simulations:

```
%let meandiff =     5;
%let stddev   =    12;
%let alpha    =  0.05;
%let ntotal   =   100;
%let nsim     = 10000;

data simdata;
   call streaminit(123);
   do isim = 1 to &nsim;
      do i = 1 to floor(&ntotal/2);
         group = 1;
         y = rand('normal', 0        , &stddev);
         output;
         group = 2;
         y = rand('normal', &meandiff, &stddev);
         output;
      end;
   end;
run;

ods listing close;
proc ttest data=simdata;
   ods output ttests=tests;
   by isim;
   class group;
   var y;
run;
ods listing;
```

```
data tests;
   set tests;
   where method="Pooled";
   issig = probt < &alpha;
run;

proc freq data=tests;
   ods select binomialprop;
   tables issig / binomial(level='1');
run;
```

First the DATA step is used to randomly generate *nsim*=10,000 data sets based on the *meandiff*, *stddev*, and *ntotal* parameters and the normal distribution, consistent with the assumptions underlying the two-sample *t* test. These data sets are contained in a large SAS data set called simdata indexed by the variable isim.

The CALL STREAMINIT(123) statement initializes the random number generator with a specific sequence and ensures repeatable results for purposes of this example. (**NOTE:** Skip this step when you are performing actual power simulations.)

The TTEST procedure is run using isim as a BY variable, with the ODS LISTING CLOSE statement to suppress output. The ODS OUTPUT statement saves the "TTests" table to a data set called tests. The *p*-values are contained in a column called probt.

The subsequent DATA step defines a variable called issig to flag the significant *p*-values.

Finally, the FREQ procedure computes the empirical power estimate as the estimate of $P(\text{issig} = 1)$ and provides approximate and exact confidence intervals for this estimate.

Figure 19.7 shows the results. The estimated power is 0.5388 with 95% confidence interval (0.5290, 0.5486). Note that the exact power of 0.541 shown in the first row in Figure 19.1 is contained within this tight confidence interval.

Figure 19.7 Simulated Power (DATA Step, SAS/STAT Software)

The FREQ Procedure

Binomial Proportion for issig = 1

Proportion	0.5388
ASE	0.0050
95% Lower Conf Limit	0.5290
95% Upper Conf Limit	0.5486
Exact Conf Limits	
95% Lower Conf Limit	0.5290
95% Upper Conf Limit	0.5486

References

Castelloe, J. M. (2000), "Sample Size Computations and Power Analysis with the SAS System," *Proceedings of the Twenty-fifth Annual SAS Users Group International Conference,* Paper 265-25, Cary, NC: SAS Institute Inc.

Castelloe, J. M. and O'Brien, R. G. (2001), "Power and Sample Size Determination for Linear Models," *Proceedings of the Twenty-sixth Annual SAS Users Group International Conference,* Paper 240-26. Cary, NC: SAS Institute Inc.

Muller, K. E. and Benignus, V. A. (1992), "Increasing Scientific Power with Statistical Power," *Neurotoxicology and Teratology*, 14, 211–219.

O'Brien, R. G. and Castelloe, J. (2007), "Sample-Size Analysis for Traditional Hypothesis Testing: Concepts and Issues," in *Pharmaceutical Statistics Using SAS: A Practical Guide*, ed. A. Dmitrienko, C. Chuang-Stein and R. D'Agostino, Cary, NC: SAS Institute Inc., Chapter 10, 237–271.

O'Brien, R. G. and Muller, K. E. (1993), "Unified Power Analysis for *t*-Tests through Multivariate Hypotheses," in *Applied Analysis of Variance in Behavioral Science*, ed. L. K. Edwards, New York: Marcel Dekker, Chapter 8, 297–344.

Lenth, R. V. (2001), "Some Practical Guidelines for Effective Sample Size Determination," *The American Statistician*, 55, 187–193.

Chapter 20

Using the Output Delivery System

Contents

Overview: Using the Output Delivery System **433**
Output Objects and ODS Destinations 434
Paths and Selection 435
ODS and the SAS Results Window 440
Controlling Output Appearance with Templates 441
ODS and the NOPRINT Option 447
Examples: Using the Output Delivery System **448**
Example 20.1: Creating HTML Output with ODS 448
Example 20.2: Selecting ODS Tables for Display 451
Example 20.3: Excluding ODS Tables from Display 454
Example 20.4: Creating an Output Data Set from an ODS Table 457
Example 20.5: Creating an Output Data Set: Subsetting the Data 460
Example 20.6: RUN-Group Processing 461
Example 20.7: ODS Output Data Sets and Using PROC TEMPLATE to Customize Output 464
Example 20.8: HTML Output with Hyperlinks between Tables 477
Example 20.9: HTML Output with Graphics and Hyperlinks 482
Example 20.10: Correlation and Covariance Matrices 486
References **495**

Overview: Using the Output Delivery System

Most SAS procedures use the Output Delivery System (ODS) to manage their output. ODS enables you to do the following:

- display your output in hypertext markup language (HTML), Rich Text Format (RTF), PDF, PostScript, SAS listing (the default), or other formats
- create SAS data sets directly from tables or plots
- select or exclude individual pieces of output

- customize the layout, format, headers, and style of your output
- produce graphs with ODS Graphics in many procedures (see Chapter 21, "Statistical Graphics Using ODS")

This chapter discusses some typical applications of ODS with SAS software. For complete documentation on the Output Delivery System, see the *SAS Output Delivery System: User's Guide*.

Output Objects and ODS Destinations

All SAS procedures produce *output objects* that ODS delivers to various *ODS destinations*, according to the default specifications for the procedure or according to your own specifications. Typically, you see the output objects displayed as tables, data sets, or graphs. Underlying all output (for example, a table of parameter estimates) are two component parts:

- the data component, which consists of the results computed by a SAS procedure
- the template, which contains the instructions for formatting and displaying the results

You define the form that the output should take when you specify an ODS destination. Some supported destinations are as follows:

- LISTING destination (the standard SAS listing), which is open by default
- HTML destination, for viewing in a browser
- RTF, for inclusion in Microsoft Word
- PDF, PostScript, and PCL, for high-fidelity printers
- OUTPUT destination, for saving results to SAS data sets
- DOCUMENT destination, for saving, modifying and replaying your output

You can open multiple ODS destinations at the same time so that a single procedure step can produce output for multiple destinations. If you do not supply any ODS statements, ODS delivers all output to the LISTING destination. You can specify an output style for each ODS destination. The style controls the foreground, background, colors, lines, fonts, and so on.

Each output object has an associated template, provided by SAS, that defines its presentation format. You can use the TEMPLATE procedure to view or alter these templates or to create new templates, changing the headers, formats, column order, and so on. For more information, see the chapter titled "The Template Procedure" in the *SAS Output Delivery System: User's Guide*.

The following statements provide an example of temporarily closing the LISTING destination and making an output data set, Parm, from the parameter estimates table from PROC REG. Closing the

LISTING destination is not required, but it is done frequently in the examples in this chapter for efficiency. Closing the LISTING suppresses the generation of output that is not needed or used. This is particularly beneficial with graphics. This example uses the SASHELP.Class data set, one of the sample data sets in the SASHELP library that are automatically available for your use. The following statements produce Figure 20.1:

```
title 'Getting Started with ODS';

ods listing close;

proc reg data=sashelp.class;
   model height=weight;
   ods output ParameterEstimates=parms;
run;

ods listing;

proc print noobs data=parms;
run;
```

Figure 20.1 PROC REG Parameter Estimates Table

Getting Started with ODS

Model	Dependent	Variable	DF	Estimate	StdErr	tValue	Probt
MODEL1	Height	Intercept	1	42.57014	2.67989	15.89	<.0001
MODEL1	Height	Weight	1	0.19761	0.02616	7.55	<.0001

The ODS OUTPUT statement contains a table name, an equal sign, and the name of the output SAS data set to create. The table names can be found using the ODS TRACE statement, which is described in the next section. Also see Example 20.4 for more information.

Paths and Selection

You use the ODS statement to provide instructions to ODS. You can use ODS statements to specify options for different destinations, specify the output style, and select and exclude output. Here are some examples:

```
/* open the HTML destination with the statistical style */
ods html style=Statistical;

/* select only the parameter estimates table */
ods select ParameterEstimates;

/* output the parameter estimates table to a SAS data set*/
ods output ParameterEstimates=Parms;
```

```
/* exclude the number of observations, ANOVA, and fit statistics tables */
ods exclude NObs ANOVA FitStatistics;
```

In order to select, exclude, or modify a table, you must first know its name. You can obtain the table names in several ways:

- You can obtain table names from the individual procedure chapter or from the individual procedure section of the SAS online Help system. See the "ODS Table Names" section within the "Details" section of the procedure documentation chapter.
- You can use the SAS Results window to view the names of the tables created in your SAS session (see the section "ODS and the SAS Results Window" on page 440 for more information).
- You can use the ODS TRACE statement to find the names of tables created in your SAS session. The ODS TRACE statement writes identifying information to the SAS log or listing for each generated output table.

If you are working interactively with reasonably small data sets, then the ODS TRACE statement is usually the most convenient way to find the names. Specify the ODS TRACE ON statement prior to the procedure statements that create the output for which you want information. For example, the following statements write the trace record for the specific tables created in the REG procedure step:

```
ods trace on;
proc reg;
   model y=x;
   model z=x;
run;
ods trace off;
```

By default, the trace record is written to the SAS log. The output from this step is as follows:

```
Output Added:
-------------
Name:        NObs
Label:       Number of Observations
Template:    Stat.Reg.NObs
Path:        Reg.MODEL1.Fit.y.NObs
-------------

Output Added:
-------------
Name:        ANOVA
Label:       Analysis of Variance
Template:    Stat.REG.ANOVA
Path:        Reg.MODEL1.Fit.y.ANOVA
-------------
```

```
Output Added:
-------------
Name:       FitStatistics
Label:      Fit Statistics
Template:   Stat.REG.FitStatistics
Path:       Reg.MODEL1.Fit.y.FitStatistics
-------------

Output Added:
-------------
Name:       ParameterEstimates
Label:      Parameter Estimates
Template:   Stat.REG.ParameterEstimates
Path:       Reg.MODEL1.Fit.y.ParameterEstimates
-------------

Output Added:
-------------
Name:       NObs
Label:      Number of Observations
Template:   Stat.Reg.NObs
Path:       Reg.MODEL2.Fit.z.NObs
-------------

Output Added:
-------------
Name:       ANOVA
Label:      Analysis of Variance
Template:   Stat.REG.ANOVA
Path:       Reg.MODEL2.Fit.z.ANOVA
-------------

Output Added:
-------------
Name:       FitStatistics
Label:      Fit Statistics
Template:   Stat.REG.FitStatistics
Path:       Reg.MODEL2.Fit.z.FitStatistics
-------------

Output Added:
-------------
Name:       ParameterEstimates
Label:      Parameter Estimates
Template:   Stat.REG.ParameterEstimates
Path:       Reg.MODEL2.Fit.z.ParameterEstimates
-------------
```

Alternatively, you can specify the LISTING option (**`ods trace on / listing;`**), which writes the trace record, interleaved with the procedure output, to the LISTING destination.

The trace record contains the name of each created table and its associated label, template, and path. The label provides a description of the table. The path shows the output hierarchy for the table (see,

for example, Figure 20.2). In this example, the hierarchy has a level for the REG procedure, a level for the model (MODEL1 or MODEL2), a level for the fit results, a level for the dependent variable (y or z), and a level for the table name (**NObs**, **ANOVA**, **FitStatistics**, **ParameterEstimates**, **NObs**, **ANOVA**, **FitStatistics**, **ParameterEstimates**). The path shown in the trace record is fully qualified. Often, you can omit levels and instead use a partially qualified path. A partially qualified path consists of any part of the full path that begins immediately after a period and continues to the end of the full path. For example, the full path for the parameter estimates for the first model in the preceding regression analysis is **Reg.Model1.Fit.y.ParameterEstimates**. This table can be referenced in any of the following ways:

ParameterEstimates	name
y.ParameterEstimates	partially qualified path
fit.y.ParameterEstimates	partially qualified path
Model1.fit.y.ParameterEstimates	partially qualified path
Reg.Model1.fit.y.ParameterEstimates	fully qualified path

When a procedure creates multiple tables that have the same name, as in the preceding example, you have several selection options. You can specify the name, a full path, or a partially qualified path when you refer to a table (for example, when you select it or exclude it from display). You can also specify a WHERE clause. For example, you can specify any of the following statements to display both tables of parameter estimates:

```
ods select ParameterEstimates;

ods select y.ParameterEstimates z.ParameterEstimates;

ods select Reg.Model1.Fit.y.ParameterEstimates
           Reg.Model2.Fit.z.ParameterEstimates;

ods select where = (_path_ ? 'Parameter');
```

There are other possibilities as well. The first ODS SELECT statement specifies the single name, which is shared by both tables. The second statement specifies a partially qualified path for both tables. The third statement specifies the full path for both tables. The fourth statement selects every object (table or graph) that contains the string **Parameter** anywhere in its path. Note that in the first three statements, selection is case insensitive. Any combination of uppercase and lowercase letters will work. This is not true in the fourth statement, which uses an ordinary SAS comparison of character strings. The following WHERE clause selection is case insensitive:

```
ods select where = (lowcase(_path_) ? 'parameter');
```

You can also select objects based on a WHERE clause and the label path. The following statements turn on the trace record, display a label path in addition to the name path, and select all tables that have the string 'var' in the label:

```
ods trace on / label;
ods select where = (lowcase(_label_) ? 'var');
```

A subset of the trace record for PROC REG with this select list, showing just the name and label paths, is as follows:

```
Path:       Reg.MODEL1.Fit.y.ANOVA
Label Path: 'The Reg Procedure'.'MODEL1'.'Fit'.y.'Analysis of Variance'
Path:       Reg.MODEL2.Fit.z.ANOVA
Label Path: 'The Reg Procedure'.'MODEL2'.'Fit'.z.'Analysis of Variance'
```

The ODS SELECT statement selects the ANOVA tables, because they have the string 'Analysis of Variance' (which when lowercased contains 'var') in their labels. WHERE clause selection is also useful for selecting all of the objects within a group or level of the path hierarchy (the group 'MODEL2' or 'Fit'). You can specify any part of the path or label path—for example, '.z.', the variable z ignoring any 'z' that might be in the middle of a word; '2.F', model 2 fit tables and any other table that has the string '2.F' in its path; and so on.

ODS records the specified table names in its internal selection or exclusion list, and then it processes the output it receives. Note that ODS maintains an overall selection or exclusion list that pertains to all ODS destinations, and it maintains a separate selection or exclusion list for each ODS destination. The list for a specific destination provides the primary filtering step. The restrictions that you specify in the overall list are added to the destination-specific lists.

Suppose, for example, that your LISTING exclusion list (that is, the list of tables you want to exclude from the LISTING destination) contains the **FitStatistics** table, which you specify with the following statement:

```
ods listing exclude FitStatistics;
```

Suppose also that your overall selection list (that is, the list of tables you want to select for all destinations) contains the tables **ParameterEstimates** and **FitStatistics**, which you specify with the following statement:

```
ods select ParameterEstimates FitStatistics;
```

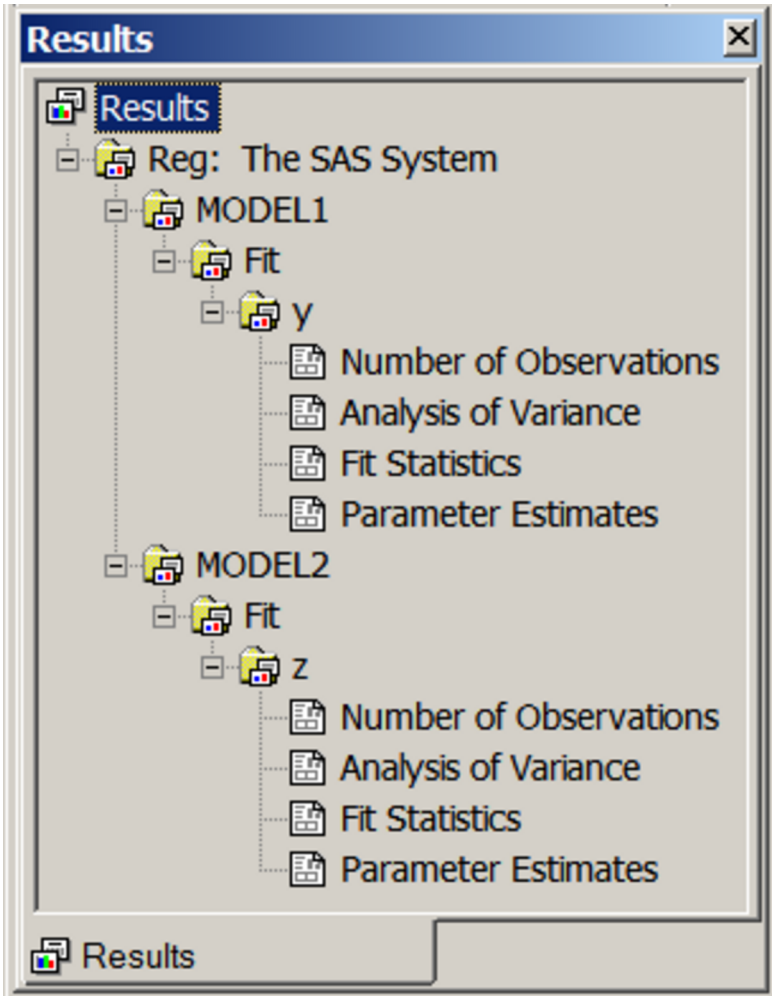
ODS then sends only the **ParameterEstimates** and **FitStatistics** tables to all open destinations except the LISTING destination. It sends only the **ParameterEstimates** table to the LISTING destination because the table **FitStatistics** is excluded from that destination.

Some SAS procedures, such as PROC REG and PROC GLM, support RUN-group processing, which means that a RUN statement does not end the procedure. A QUIT statement explicitly ends such procedures. If you omit the QUIT statement, a PROC or a DATA statement implicitly ends such procedures. When you use ODS with procedures that support RUN-group processing, it is good programming practice to specify a QUIT statement at the end of the procedure. This causes ODS to clear the selection or exclusion list, and you are less likely to encounter unexpected results.

ODS and the SAS Results Window

The SAS Results window contains a running record of the output from your SAS session. In the display manager, select **View ► Results** to open the Results window. Figure 20.2 displays the Results window from the PROC REG step shown previously.

Figure 20.2 The Results Window from the SAS Explorer



When you click on the output table names in the Results window, you link directly to the output in the output window (or HTML browser if you specify the HTML destination). The Results window contains a row for each level of the path and for each table. You can also use the Results window to determine the names of the templates associated with each output table. Right-click on the table name and then select **Properties.** You can see all of the templates from the Results window by selecting **View ► Templates ► Sashelp.Tmplmst**. Then click on a product such as **`Stat`**, a procedure such as **`REG`**, and a template such as **`ParameterEstimates`**.

Controlling Output Appearance with Templates

A template is a description of how output should appear when it is formatted. Templates describe several characteristics of the output, including headers, column ordering, style information, justification, and formats. Each table in the output has a template, and all SAS templates are stored in the SASHELP library. You can find the template associated with a particular output table or column by using the ODS TRACE statement or the SAS Results window. You can create or modify a template with the TEMPLATE procedure. For example, you can specify different column headings or different orders of columns in a table.

There are a number of different types of templates including column and table templates, graphical templates, and style templates. A column or table template applies to the specific columns or tables that reference the template. Graphical templates are discussed in more detail in Chapter 21, "Statistical Graphics Using ODS." A style template applies to an entire SAS job, including all tables and graphs, and can be specified with the STYLE= option in a valid ODS destination, such as HTML, RTF, or PDF. You can specify a style as follows:

```
ods html style=Statistical;
```

A style template controls stylistic elements such as colors, fonts, and presentation attributes. You can change the style to give your output different looks and color schemes. You can also reference style information in table templates for individual headers and data cells. You can modify all types of templates with PROC TEMPLATE. For information about creating your own styles, see the *SAS Output Delivery System: User's Guide*.

You can display the contents of a template by running PROC TEMPLATE with a SOURCE statement and a template name, as in the following example:

```
proc template;
   source Stat.REG.ANOVA;
   source Stat.GLM.OverallANOVA;
run;
```

Note that in many cases, a template definition is at least in part based on another template. When you see the PARENT=*template* option in a template definition, you need to look at the specified template to see more about the rest of the template definition. To illustrate, consider the following PROC GLM step:

```
proc glm data=x;
   model y = x;
run; quit;
```

The ANOVA table from this step is displayed in Figure 20.3.

Figure 20.3 PROC GLM ANOVA Table with the Default Template

Getting Started with ODS

The GLM Procedure

Dependent Variable: y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	77.05005975	77.05005975	190.53	<.0001
Error	8	3.23526554	0.40440819		
Corrected Total	9	80.28532529			

The sums of squares and mean squares are presented with eight decimal places. You can change the templates to change the formats of those columns to use fewer decimal places. First, you can use the ODS TRACE statement when you run PROC GLM to determine the name of the template:

```
ods trace output;
proc glm data=x;
   model y = x;
   run; quit;
ods trace off;
```

The trace output results include the following:

```
Output Added:
-------------
Name:        OverallANOVA
Label:       Overall ANOVA
Template:    stat.GLM.OverallANOVA
Path:        GLM.ANOVA.y.OverallANOVA
-------------
```

From this, you can see that the template for the overall ANOVA table is **stat.GLM.OverallANOVA**. You can submit the following statements to see the overall ANOVA table template:

```
proc template;
   source stat.glm.overallanova;
run;
```

Here are the results:

```
define table Stat.GLM.Overallanova;
   notes "Over-all ANOVA";
   top_space = 1;
   parent = Stat.GLM.ANOVA;
   double_space;
end;
```

The results show that this template inherits its definition from a parent template named **Stat.GLM.ANOVA**. Submit the following statements to see the parent template:

```
proc template;
   source stat.glm.anova;
run;
```

Some of the results are as follows:

```
.
.
.

define SS;
   parent = Stat.GLM.SS;
end;

define MS;
   parent = Stat.GLM.MS;
end;

.
.
.
```

These columns inherit there definitions from the parent columns named **Stat.GLM.SS** and **Stat.GLM.MS**. This is all of the information that you need to redefine these columns, but you can run PROC TEMPLATE again as follows to see more information about how these templates are defined:

```
proc template;
   source Stat.GLM.SS;
   source Stat.GLM.MS;
run;
```

The results are as follows:

```
define column Stat.GLM.Ss;
   notes "Parent for GLM ANOVA Sums of Squares columns";
   parent = Common.ANOVA.SS;
end;
define column Stat.GLM.Ms;
   notes "Parent for GLM ANOVA Mean Squares columns";
   parent = Common.ANOVA.MS;
end;
```

These columns inherit there definitions from the columns named **Common.ANOVA.SS** and **Common.ANOVA.MS**. You can run PROC TEMPLATE as follows to see their definitions:

```
proc template;
   source Common.ANOVA.SS;
   source Common.ANOVA.MS;
run;
```

The results are as follows:

```
define column Common.ANOVA.Ss;
   notes "Default ANOVA Sum of squares column";
   header = "Sum of Squares";
   translate _val_=._ into "";
end;
define column Common.ANOVA.Ms;
   notes "Default ANOVA Mean square column";
   header = "Mean Square";
   translate _val_=._ into "";
end;
```

You can redefine **Common.ANOVA.SS** and **Common.ANOVA.MS** to change all **SS** and **MS** columns in ANOVA tables. This would be the most general redefinition. More specifically, you can redefine **Stat.GLM.SS** and **Stat.GLM.MS** to change **SS** and **MS** columns in ANOVA tables produced by PROC GLM. Finally, and most specifically, you can change the **SS** and **MS** columns in just the overall ANOVA table template. In this example, the **Stat.GLM.SS** and **Stat.GLM.MS** columns are redefined as follows, so that results are displayed with fewer decimal places:

```
proc template;
   edit Stat.GLM.SS;
      choose_format=max format_width=8;
   end;
   edit Stat.GLM.MS;
      choose_format=max format_width=8;
   end;
run;
```

The CHOOSE_FORMAT=MAX option along with FORMAT_WIDTH=8 chooses the format for each column based on the maximum value in that column and an overall width of 8. Note that you are editing and not replacing the definition, so column header and other information in the definition is not lost. The following step uses the new templates:

```
proc glm data=x;
   model y = x;
run; quit;
```

The new ANOVA results, using the edited templates, are shown in Figure 20.4. You can see that the original results in Figure 20.3 have eight decimal places, whereas the new results in Figure 20.4 have only five decimal places and an overall format width of eight.

Figure 20.4 PROC GLM ANOVA Table after Template Customization

Getting Started with ODS

The GLM Procedure

Dependent Variable: y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	77.05006	77.05006	190.53	<.0001
Error	8	3.23527	0.40441		
Corrected Total	9	80.28533			

The preceding PROC TEMPLATE step produces the following notes:

```
NOTE: Overwriting existing template/link: Stat.GLM.Ss
NOTE: COLUMN 'Stat.GLM.Ss' has been saved to: SASUSER.TEMPLAT
NOTE: Overwriting existing template/link: Stat.GLM.Ms
NOTE: COLUMN 'Stat.GLM.Ms' has been saved to: SASUSER.TEMPLAT
```

When you run PROC TEMPLATE to modify or edit a template, the template is by default stored in your SASUSER library. You can then modify the path that ODS uses to look up templates with the ODS PATH statement—for example, to access these new templates in a later SAS session. This enables you to create a new default set of templates to modify the display format for all of your SAS output. You can specify the SHOW option in the ODS PATH statement to determine the current path. The following statements illustrate the ODS path:

```
ods path show;
libname mytpls '.';
ods path (prepend) mytpls.template(update);
ods path show;

proc template;
   edit Stat.GLM.SS;
      choose_format=max format_width=8;
   end;
   edit Stat.GLM.MS;
      choose_format=max format_width=8;
   end;
run;
```

The results of the first statement are as follows:

```
Current ODS PATH list is:

1. SASUSER.TEMPLAT(UPDATE)
2. SASHELP.TMPLMST(READ)
```

This shows that the SASUSER.TEMPLAT file is open for storing new templates and retrieving templates for use. After that, the SASHELP.TMPLMST file is used, but it is open only for Read access. You cannot modify templates in SASHELP. The second and third statements add a template library to the front of this list in the current directory. The fourth statement shows the new path list, which is as follows:

```
Current ODS PATH list is:

1. MYTPLS.TEMPLATE(UPDATE)
2. SASUSER.TEMPLAT(UPDATE)
3. SASHELP.TMPLMST(READ)
```

The PROC TEMPLATE step produces the following notes showing that the templates are now stored in MYTPLS.TEMPLATE:

```
NOTE: Overwriting existing template/link: Stat.GLM.Ss
NOTE: COLUMN 'Stat.GLM.Ss' has been saved to: MYTPLS.TEMPLATE
NOTE: Overwriting existing template/link: Stat.GLM.Ms
NOTE: COLUMN 'Stat.GLM.Ms' has been saved to: MYTPLS.TEMPLATE
```

In all cases, the original template definitions in SASHELP.TMPLMST are not changed. You can delete your custom template and restore the default template as follows:

```
proc template;
   delete Stat.GLM.SS;
   delete Stat.GLM.MS;
run;
```

It is good practice to delete any template redefinitions that you do not want to be permanent, because otherwise they will persist beyond the duration of your SAS session.

ODS and the NOPRINT Option

Many SAS procedures support a NOPRINT option that you can use when you want to create an output data set but without displaying any output. You use an option (such as OUTEST= or an OUTPUT statement with an OUT= option) in addition to the procedure's NOPRINT option to create a data set and suppress displayed output.

You can also use ODS to create output data sets by using the ODS OUTPUT statement. However, if you specify the NOPRINT option, the procedure might not send any output to ODS. In most procedures that support a NOPRINT option, NOPRINT means no ODS. (However, there are a few procedures that for historical reasons still might produce some output even when NOPRINT is specified.) When you want to create output data sets through the ODS OUTPUT statement, and you want to suppress the display of all output, specify the following statement instead of using the NOPRINT option:

```
ods select none;
```

Alternatively, you can close the active ODS destinations, for example, like this:

```
ods html close;
ods listing close;
```

Note that ODS statements do not instruct a procedure to generate output. Instead, they specify how ODS should manage output once it is created. You must ensure that the proper procedure options are in effect, or the output will not be generated. For example, the following statements do not create the requested data set Parms:

```
proc glm;
   ods output ParameterEstimates=Parms;
   class x;
   model y=x;
run; quit;
```

This is because the SOLUTION option was not specified in the MODEL statement. Since PROC GLM did not create the table, ODS cannot make the output data set. When you execute these statements, the following message is displayed in the log:

```
WARNING: Output 'ParameterEstimates' was not created.
```

The following step creates the output data set:

```
proc glm;
   ods output ParameterEstimates=Parms;
   class x;
   model y=x / solution;
run; quit;
```

Examples: Using the Output Delivery System

This section provides examples of making HTML output, selecting and excluding output, tracing ODS output, using the Results window, creating ODS output data sets, modifying templates, creating hyperlinks, and using ODS Graphics.

Example 20.1: Creating HTML Output with ODS

This example demonstrates how you can use the ODS HTML statement to display your output in HTML. The following statements create the data set Scores, which contains the golf scores of boys and girls in a physical education class:

```
title 'Comparing Group Means';

data Scores;
   input Gender $ Score @@;
   datalines;
f 75  f 76  f 80  f 77  f 80  f 77  f 73
m 82  m 80  m 85  m 85  m 78  m 87  m 82
;
```

The TTEST procedure is used to compare the scores. The ODS HTML statement specifies the name of the file to contain the HTML output. The following statements create the HTML file *ttest.htm*:

```
ods html body='ttest.htm' style=statistical;

proc ttest;
   class Gender;
   var Score;
run;

ods html close;
```

By default, the LISTING destination receives all output generated during your SAS session. In this example, the ODS HTML statement opens the HTML destination as well, and both destinations receive the generated output. Note that you must specify the following statement before you can view your output in a browser:

```
ods html close;
```

If you do not close the HTML destination, your HTML file might contain no output or incomplete output, or you might experience other unexpected results.

The following statements use ODS to display the output in HTML with a table of contents by using the STATISTICAL style of output, but this time closing the LISTING destination:

```
ods listing close;
ods html body='ttest.htm' contents='ttestc.htm' frame='ttestf.htm'
         style=statistical;

proc ttest;
   class Gender;
   var Score;
run;

ods html close;
ods listing;
```

The ODS HTML statement specifies three files. The BODY= option specifies the file that contains the SAS output. The CONTENTS= option specifies the file that contains the table of contents. The FRAME= option specifies the file that displays both the table of contents and the output. You can open the FRAME= file (*ttestf.htm*) in your browser to view the table of contents together with the generated output (see Output 20.1.1). By default, the HTML files are generated in your current working directory. You can instead specify paths, such as `frame='html/ttestf.htm'`, to put a file in a subdirectory.

Note that if you specify the ODS HTML statement with only the BODY= argument, no table of contents is created. The table of contents contains the descriptive label for each output table produced in the PROC TTEST step. You can select any label in the table of contents and the corresponding output will be displayed on the right side of the browser window.

Output 20.1.1 HTML Output with a Table of Contents, the FRAME='ttestf.htm' File

Table of Contents

1. Ttest
   - Score
     - Statistics
     - Confidence Limits
     - T-Tests
     - Equality of Variances

Comparing Group Means

The TTEST Procedure

Variable: Score

Gender	N	Mean	Std Dev	Std Err	Minimum	Maximum
f	7	76.8571	2.5448	0.9619	73.0000	80.0000
m	7	82.7143	3.1472	1.1895	78.0000	87.0000
Diff (1-2)		-5.8571	2.8619	1.5298		

Gender	Method	Mean	95% CL Mean		Std Dev	95% CL Std Dev	
f		76.8571	74.5036	79.2107	2.5448	1.6399	5.6039
m		82.7143	79.8036	85.6249	3.1472	2.0280	6.9303
Diff (1-2)	Pooled	-5.8571	-9.1902	-2.5241	2.8619	2.0522	4.7242
Diff (1-2)	Satterthwaite	-5.8571	-9.2064	-2.5078			

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	12	-3.83	0.0024
Satterthwaite	Unequal	11.496	-3.83	0.0026

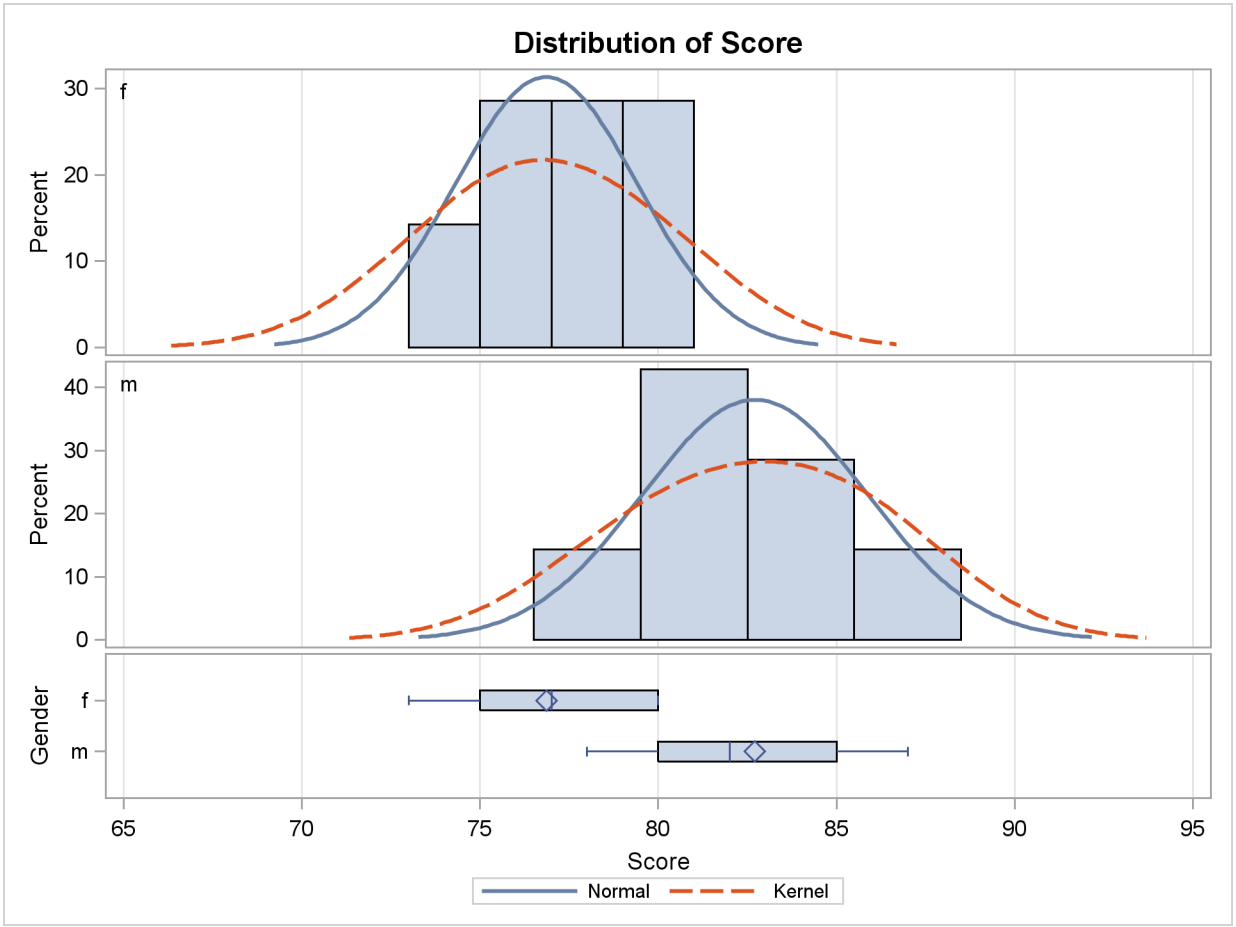
Equality of Variances				
Method	Num DF	Den DF	F Value	Pr > F
Folded F	6	6	1.53	0.6189

You could enable ODS Graphics by adding an ODS GRAPHICS ON statement as follows:

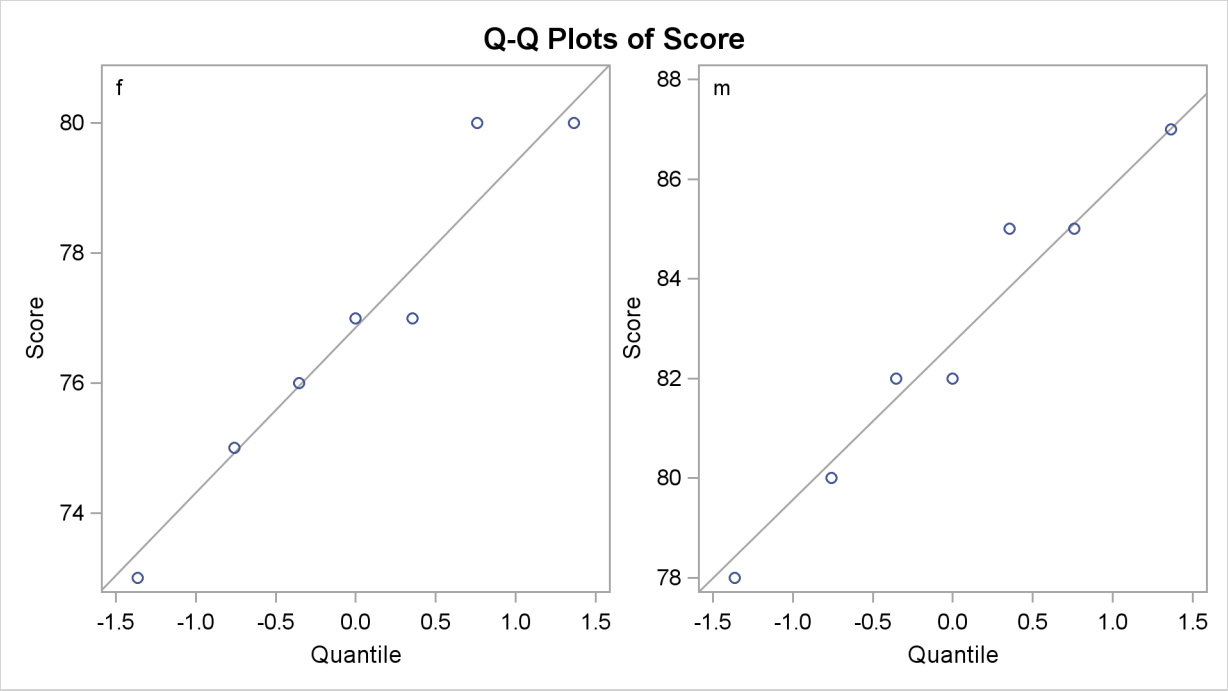
```
ods graphics on;
ods listing close;
ods html body='ttest.htm' contents='ttestc.htm' frame='ttestf.htm'
         style=statistical;

proc ttest;
   class Gender;
   var Score;
run;

ods html close;
ods listing;
ods graphics off;
```

Then in addition to the tables, you would also get the graphs shown in Figure 20.1.2. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

Output 20.1.2 PROC TTEST Graphs



Output 20.1.2 *continued*



Example 20.2: Selecting ODS Tables for Display

You can use the ODS SELECT statement to deliver only a subset of the tables or graphs to ODS destinations. The following statements create an input SAS data set and use PROC GLM to perform an analysis of an unbalanced two-way experimental design:

```
title 'Unbalanced Two-way Design';

data twoway;
   input Treatment Block y @@;
   datalines;
1 1 17   1 1 28   1 1 19   1 1 21   1 1 19
1 2 43   1 2 30   1 2 39   1 2 44   1 2 44
1 3 16
2 1 21   2 1 21   2 1 24   2 1 25
2 2 39   2 2 45   2 2 42   2 2 47
2 3 19   2 3 22   2 3 16
3 1 22   3 1 30   3 1 33   3 1 31
3 2 46
3 3 26   3 3 31   3 3 26   3 3 33   3 3 29   3 3 25
;
```

```
proc glm data=twoway;
   class Treatment Block;
   model y = Treatment | Block;
   means Treatment;
   lsmeans Treatment;

   ods select ModelANOVA Means;
   ods trace on;
   ods show;
run;
```

The ODS SELECT statement specifies that only the two tables **ModelANOVA** and **Means** are to be delivered to the ODS destinations. In this example, no ODS destinations are explicitly opened. Therefore, only the default LISTING destination receives the procedure output. The ODS SHOW statement displays the current overall selection list in the SAS log. The ODS SHOW statement is not required; it is used here simply to show the effects of the ODS SELECT statement. The results of the ODS SHOW statement are as follows:

```
Current OVERALL select list is:
1. ModelANOVA
2. Means
```

The ODS TRACE statement writes the trace record of the ODS output objects to the SAS log. The trace record is as follows:

```
Output Added:
-------------
Name:        ModelANOVA
Label:       Type I Model ANOVA
Template:    stat.GLM.Tests
Path:        GLM.ANOVA.y.ModelANOVA
-------------

Output Added:
-------------
Name:        ModelANOVA
Label:       Type III Model ANOVA
Template:    stat.GLM.Tests
Path:        GLM.ANOVA.y.ModelANOVA
-------------

Output Added:
-------------
Name:        Means
Label:       Means
Template:    stat.GLM.Means
Path:        GLM.Means.Treatment.Means
-------------
```

Note that there are two tables with the name **ModelANOVA**. One contains the "Type I Model ANOVA" table, and the other contains the "Type III Model ANOVA" table. If you wanted to select only one of them, you could specify either of the labels in the ODS SELECT statement instead of the name. You would specify one of the following:

```
ods select 'Type I Model ANOVA' Means;
ods select 'Type III Model ANOVA' Means;
```

In the following statements, the ODS SHOW statement writes the current overall selection list to the SAS log, the QUIT statement ends the PROC GLM step, and the second ODS SHOW statement writes the selection list to the log after PROC GLM terminates:

```
ods show;
quit;
ods show;
```

The results of these statements are as follows:

```
ods show;


Current OVERALL select list is:
1. ModelANOVA
2. Means

quit;
ods show;


Current OVERALL select list is: ALL
```

PROC GLM supports interactive RUN-group processing. Before the QUIT statement is executed, PROC GLM is active and the ODS selection list remains at its previous setting. The list includes only the two tables, **ModelANOVA** and **Means**. After the QUIT statement, when PROC GLM is no longer active, the selection list is reset to ALL. The displayed output, shown in Output 20.2.1, consists of the three selected tables (two **ModelANOVA** tables and the **Means** table). Note that the LS-means results are not displayed even though an LSMEANS statement was specified. This is because the LS-means table, named **LSMeans**, is not specified in the ODS SELECT statement. Other tables are suppressed as well.

Output 20.2.1 The Selected Tables from PROC GLM

Unbalanced Two-way Design

The GLM Procedure

Dependent Variable: y

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Treatment	2	8.060606	4.030303	0.24	0.7888
Block	2	2621.864124	1310.932062	77.95	<.0001
Treatment*Block	4	32.684361	8.171090	0.49	0.7460

Output 20.2.1 *continued*

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Treatment	2	266.130682	133.065341	7.91	0.0023
Block	2	1883.729465	941.864732	56.00	<.0001
Treatment*Block	4	32.684361	8.171090	0.49	0.7460

Unbalanced Two-way Design

The GLM Procedure

Level of Treatment	N	y Mean	y Std Dev
1	11	29.0909091	11.5104695
2	11	29.1818182	11.5569735
3	11	30.1818182	6.3058414

For more information about ODS exclusion and selection lists, see the section "Paths and Selection" on page 435.

Example 20.3: Excluding ODS Tables from Display

The following example demonstrates how you can use the ODS EXCLUDE statement to exclude particular tables from ODS destinations. This example also creates a SAS data set from the excluded table and uses it to create a specialized plot.

The data are from Hemmerle and Hartley (1973). The response variable consists of measurements from an oven experiment, and the model contains a fixed effect a and random effects b and a*b. The following statements create the input SAS data set:

```
title 'Oven Measurements';

data hh;
   input a b y @@;
   datalines;
1 1 237   1 1 254    1 1 246
1 2 178   1 2 179
2 1 208   2 1 178    2 1 187
2 2 146   2 2 145    2 2 141
3 1 186   3 1 183
3 2 142   3 2 125    3 2 136
;
```

The following ODS statements are submitted before the analysis, which will be done with the MIXED procedure:

```
ods listing close;
ods html body='mixed.htm' contents='mixedc.htm' frame='mixedf.htm'
         style=statistical;

ods exclude ParmSearch(persist);
ods show;
```

The ODS HTML statement specifies the filenames to contain the output generated from the statements that follow. The ODS EXCLUDE statement excludes the table **ParmSearch** from display. Although the table is excluded from the displayed output, the information contained in the **ParmSearch** table is graphically summarized in a later step.

The PERSIST option in the ODS EXCLUDE statement excludes the table for the entire SAS session or until you execute an ODS SELECT statement or an ODS EXCLUDE NONE statement. If you omit the PERSIST option, the exclusion list is cleared when the procedure terminates. The resulting exclusion list is displayed next:

```
Current OVERALL exclude list is:
1. ParmSearch(PERSIST)
```

The MIXED procedure is run to fit the model:

```
proc mixed data=hh;
   class a b;
   model y = a;
   random b a*b;
   parms (17 to 20 by 0.1) (.3 to .4 by .005) (1.0);
   ods output ParmSearch=parms;
run;

ods show;
```

All output from PROC MIXED, except the **ParmSearch** table, is delivered to the HTML and LISTING destinations. The ODS OUTPUT statement outputs the table **ParmSearch** to a SAS data set called Parms.

The ODS SHOW statement again displays the overall current exclusion list after PROC MIXED has terminated. The results of the ODS SHOW statement are displayed next:

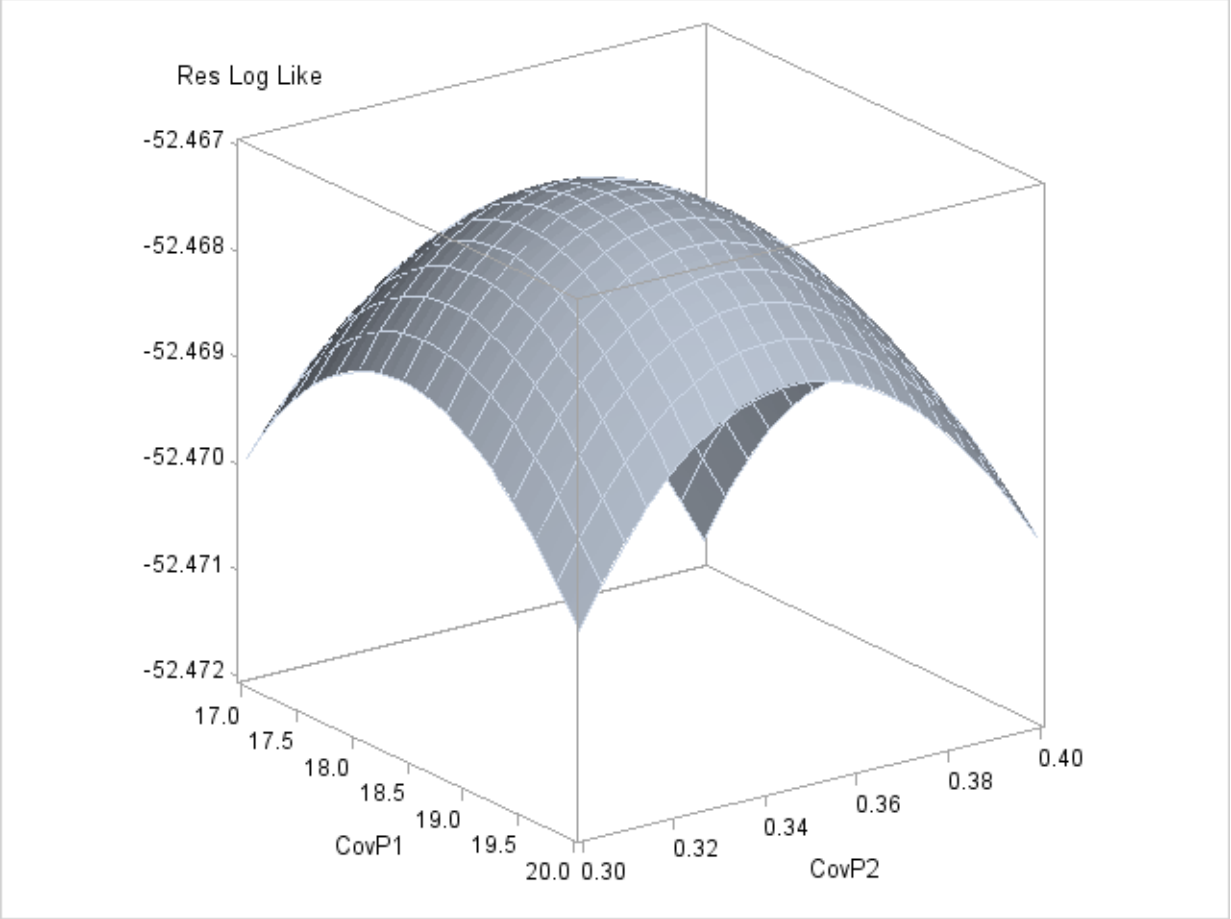
```
Current OVERALL exclude list is:
1. ParmSearch(PERSIST)
```

The **ParmSearch** table is saved in the Parms data set (as specified in the ODS OUTPUT statement). The following steps plot the surface of the residual log likelihood as a function of the covariance parameters and produce Output 20.3.1:

```
proc template;
   define statgraph surface;
      begingraph;
         layout overlay3d;
            surfaceplotparm x=CovP1 y=CovP2 z=ResLogLike;
         endlayout;
      endgraph;
   end;
run;

proc sgrender data=parms template=surface;
run;
ods html close;
```

PROC TEMPLATE is used to create a template for displaying the data as a three-dimensional surface plot. The plot is displayed with the ODS Graphics procedure SGRENDER. For more information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

Output 20.3.1 HTML Output from PROC MIXED



Example 20.4: Creating an Output Data Set from an ODS Table

In this example, the GENMOD procedure is used to perform Poisson regression, and part of the resulting procedure output is written to a SAS data set with the ODS OUTPUT statement. Insurance claims data are classified by two factors: age group (with two levels) and car type (with three levels). The following statements create the data set Insure:

```
title 'Insurance Claims';

data Insure;
   input n c Car $ Age;
   ln = log(n);
   datalines;
 500   42  Small  1
1200   37  Medium 1
 100    1  Large  1
 400  101  Small  2
 500   73  Medium 2
 300   14  Large  2
;
```

The variable n represents the number of insurance policyholders, and the variable c represents the number of insurance claims. The variable Car represents the type of car involved (classified into three groups), and the variable Age is the age of a policyholder (classified into two groups).

You can use PROC GENMOD to perform a Poisson regression analysis of these data with a log link function. Assume that the number of claims variable c has a Poisson probability distribution and the log of its mean, μ_i, is related to the factors Car and Age.

The following statements obtain the names of the tables produced by this PROC GENMOD run. The ODS TRACE statement lists the trace record. If you already know the names, such as by looking them up in the procedure documentation, you do not have to run this step. The following step displays the trace information:

```
ods trace on;

proc genmod data=insure;
   class car age;
   model c = car age / dist   = poisson
                       link   = log
                       offset = ln
                       obstats;
run;
ods trace off;
```

The trace record from the SAS log is displayed next:

```
Output Added:
-------------
Name:        ModelInfo
Label:       Model Information
Template:    Stat.Genmod.ModelInfo
```

```
Path:       Genmod.ModelInfo
-------------

Output Added:
-------------
Name:       NObs
Label:      Number of observations summary
Template:   Stat.Genmod.NObs
Path:       Genmod.NObs
-------------

Output Added:
-------------
Name:       ClassLevels
Label:      Class Level Information
Template:   Stat.Genmod.Classlevels
Path:       Genmod.ClassLevels
-------------

Output Added:
-------------
Name:       ParmInfo
Label:      Parameter Information
Template:   Stat.Genmod.Parminfo
Path:       Genmod.ParmInfo
-------------

Output Added:
-------------
Name:       ModelFit
Label:      Criteria For Assessing Goodness Of Fit
Template:   stat.genmod.ModelFit
Path:       Genmod.ModelFit
-------------

Output Added:
-------------
Name:       ConvergenceStatus
Label:      Convergence Status
Template:   Stat.Genmod.ConvergenceStatus
Path:       Genmod.ConvergenceStatus
-------------

Output Added:
-------------
Name:       ParameterEstimates
Label:      Analysis Of Parameter Estimates
Template:   stat.genmod.parameterestimates
Path:       Genmod.ParameterEstimates
-------------

Output Added:
-------------
Name:       ObStats
```

```
Label:       Observation Statistics
Template:    Stat.Genmod.Obstats
Path:        Genmod.ObStats
-------------
```

In the following step, the ODS OUTPUT statement writes the ODS table **ObStats** to a SAS data set named myObStats. The LISTING destination is closed so that no output is displayed. All of the usual data set options, such as the KEEP= or RENAME= option, can be used in the ODS OUTPUT statement. Thus, to create the myObStats data set so that it contains only certain columns from the **ObStats** table, you can use the data set options as follows:

```
ods listing close;
proc genmod data=insure;
   class car age;
   model c = car age / dist=poisson link=log offset=ln obstats;
   ods output ObStats=myObStats(keep=car age pred
                                rename=(pred=PredictedValue));
run;
```

The KEEP= option in the ODS OUTPUT statement specifies that only the variables Car, Age, and Pred are written to the data set, and the Pred variable is renamed PredictedValue. The following statements sort the output data set myObStats, reopen the LISTING destination for output, and produce Output 20.4.1:

```
proc sort data=myObStats;
   by descending PredictedValue;
run;

ods listing;
proc print data=myObStats noobs;
   title2 'Values of Car, Age, and the Predicted Values';
run;
```

When a destination is closed, it remains closed until it is explicitly reopened.

Output 20.4.1 The ObStats Table Created as a SAS Data Set

Insurance Claims
Values of Car, Age, and the Predicted Values

Car	Age	Predicted Value
Small	2	107.2011
Medium	2	67.025444
Medium	1	42.974556
Small	1	35.798902
Large	2	13.773459
Large	1	1.2265414

Example 20.5: Creating an Output Data Set: Subsetting the Data

This example demonstrates how you can create an output data set with the ODS OUTPUT statement and also use data set selection keywords to limit the output that ODS writes to a SAS data set. The data set, called Color, contains the eye color and hair color of children from two different regions of Europe. The data are recorded as cell counts, where the variable Count contains the number of children exhibiting each of the 15 combinations of eye and hair color. The following statements create the SAS data set:

```
title 'Hair Color of European Children';

data Color;
   input Region Eyes $ Hair $ Count @@;
   label Eyes  ='Eye Color'
         Hair  ='Hair Color'
         Region='Geographic Region';
   datalines;
1 blue  fair   23  1 blue  red     7  1 blue  medium 24
1 blue  dark   11  1 green fair   19  1 green red     7
1 green medium 18  1 green dark   14  1 brown fair   34
1 brown red     5  1 brown medium 41  1 brown dark   40
1 brown black   3  2 blue  fair   46  2 blue  red    21
2 blue  medium 44  2 blue  dark   40  2 blue  black   6
2 green fair   50  2 green red    31  2 green medium 37
2 green dark   23  2 brown fair   56  2 brown red    42
2 brown medium 53  2 brown dark   54  2 brown black  13
;
```

The following statements close the LISTING destination and sort the observations in the Color data set by the Region variable:

```
ods listing close;

proc sort data=Color;
   by Region;
run;
```

The following ODS OUTPUT statement creates the **ChiSq** table as a SAS data set named myStats:

```
ods output ChiSq=myStats
           (drop=Table
            where=(Statistic =: 'Chi' or
                   Statistic =: 'Like'));
```

Note that you can obtain the names of the tables created by any procedure in the individual procedure chapter or from the individual procedure section of the SAS online Help system. (See the "ODS Table Names" section in the "Details" section of the documentation.) You can also determine the names of tables with the ODS TRACE statement (see Example 20.4 and Example 20.2). The DROP= data set option excludes variables from the new data set. The WHERE= data set option se-

lects observations for output to the new data set myStats—specifically, those that begin with 'Chi' or 'Like'.

The following statements create Output 20.5.1:

```
proc freq data=Color order=data;
   weight Count;
   tables Eyes*Hair / testp=(30 12 30 25 3);
   by Region;
run;

ods listing;
proc print data=myStats noobs;
run;
```

The FREQ procedure is used to create and analyze a crosstabulation table from the two categorical variables Eyes and Hair, for each value of the variable Region. No ODS destinations are open until the ODS LISTING statement is encountered just prior to running the PRINT procedure.

Output 20.5.1 Output Data Set from PROC FREQ and ODS

Hair Color of European Children

Region	Statistic	DF	Value	Prob
1	Chi-Square	8	12.6331	0.1251
1	Likelihood Ratio Chi-Square	8	14.1503	0.0779
2	Chi-Square	8	18.2839	0.0192
2	Likelihood Ratio Chi-Square	8	23.3021	0.0030

Example 20.6: RUN-Group Processing

Some SAS procedures, such as PROC REG and PROC GLM, permit you to submit statements, followed by a RUN statement, followed by more statements and more RUN statements. Each group of statements, followed by a RUN statement, is called a RUN group. These procedures can produce several blocks of output for each of several RUN groups. The procedure stays active until a QUIT statement, a DATA statement, another PROC statement, or the end of the SAS session is encountered. However, ODS settings are by default cleared at RUN-group boundaries. In the following analysis, PROC REG is used to compute the covariance matrix of the estimates for two different models, and the covariance matrices are saved in a single SAS data set. The PERSIST= option in the ODS OUTPUT statement is required to make this happen. The PERSIST= option maintains ODS settings across RUN statements for procedures that support RUN-group processing.

Consider the following population growth trends. The population of the United States from 1790 to 1970 is fit to linear and quadratic functions of time. Note that the quadratic term, YearSq, is created in the DATA step; this is necessary since polynomial effects such as Year*Year cannot be specified in the MODEL statement in PROC REG. The data are as follows:

```
title1 'US Population Study';
title2 'Concatenating Two Tables into One Data Set';

data USPopulation;
   input Population @@;
   retain Year 1780;
   Year=Year+10;
   YearSq=Year*Year;
   Population=Population/1000;
   datalines;
3929 5308 7239 9638 12866 17069 23191 31443 39818 50155
62947 75994 91972 105710 122775 131669 151325 179323 203211
;
```

In the following statements, PROC REG is used, and the ODS OUTPUT statement with the PERSIST= option creates a data set with the **CovB** table (the covariance matrix of the estimates):

```
proc reg data=USPopulation;
   ods output covb(persist=run)=Bmatrix;
   var YearSq;
   model Population = Year / covb;
run;
```

The MODEL statement defines the regression model, and the COVB matrix is requested. The RUN statement executes PROC REG and the model is fit, producing a covariance matrix of the estimates with two rows and two columns. The results are displayed in Output 20.6.1 and Output 20.6.2.

Output 20.6.1 Regression Results for the Model Population

```
                          US Population Study
                 Concatenating Two Tables into One Data Set

                           The REG Procedure
                             Model: MODEL1
                     Dependent Variable: Population

                  Number of Observations Read          19
                  Number of Observations Used          19

                          Analysis of Variance

                                 Sum of           Mean
 Source                   DF       Squares         Square    F Value    Pr > F

 Model                     1         66336          66336     201.87    <.0001
 Error                    17    5586.29253      328.60544
 Corrected Total          18         71923

             Root MSE             18.12748    R-Square     0.9223
             Dependent Mean       69.76747    Adj R-Sq     0.9178
             Coeff Var            25.98271
```

Output 20.6.1 *continued*

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-1958.36630	142.80455	-13.71	<.0001
Year	1	1.07879	0.07593	14.21	<.0001

Output 20.6.2 CovB Matrix for the Model Population

Covariance of Estimates

Variable	Intercept	Year
Intercept	20393.138485	-10.83821461
Year	-10.83821461	0.0057650078

In the next step, the YearSq variable is added to the model and the model is again fit, producing a covariance matrix of the estimates with three rows and three columns:

```
add YearSq;
print;
run; quit;
```

The new COVB matrix is displayed in Output 20.6.3.

Output 20.6.3 CovB Matrix for the Model Population

US Population Study
Concatenating Two Tables into One Data Set

The REG Procedure
Model: MODEL1.1
Dependent Variable: Population

Covariance of Estimates

Variable	Intercept	Year	YearSq
Intercept	711450.62602	-757.2493826	0.2013282694
Year	-757.2493826	0.8061328943	-0.000214361
YearSq	0.2013282694	-0.000214361	5.7010894E-8

The PERSIST=RUN option maintains the ODS selection list across RUN statements for procedures that support RUN-group processing. If the PERSIST=RUN option is omitted, the selection list is cleared when the RUN statement is encountered and only the first COVB matrix is selected. Because the PERSIST=RUN option is specified, the selection list remains in effect throughout the PROC REG step. This ensures that each of the COVB matrices is selected and output. The following

statements display the ODS OUTPUT SAS data set and create Output 20.6.4:

```
proc print;
run;
```

Output 20.6.4 Results of the ODS OUTPUT Statement: Specifying the PERSIST Option

US Population Study
Concatenating Two Tables into One Data Set

Obs	_Run_	Model	Dependent	Variable	Intercept	Year	YearSq
1	1	MODEL1	Population	Intercept	20393.138485	-10.83821461	.
2	1	MODEL1	Population	Year	-10.83821461	0.0057650078	.
3	2	MODEL1.1	Population	Intercept	711450.62602	-757.2493826	0.2013282694
4	2	MODEL1.1	Population	Year	-757.2493826	0.8061328943	-0.000214361
5	2	MODEL1.1	Population	YearSq	0.2013282694	-0.000214361	5.7010894E-8

Note that even though the two COVB matrices do not have the same rows or columns, ODS automatically combines the two tables into one data set.

Example 20.7: ODS Output Data Sets and Using PROC TEMPLATE to Customize Output

You can use ODS statements, the DATA step, and PROC TEMPLATE to modify the appearance of your displayed tables or to display results in forms not directly produced by any procedure. The following example, similar to that given in Olinger and Tobias (1998), runs an analysis with PROC GLM. This example has several parts. It creates output data sets with the ODS OUTPUT statement, combines and manipulates those data sets, displays the results by using a standard SAS template, modifies a template by using PROC TEMPLATE, and displays the output data sets by using the modified template. Each step works toward the final goal of taking multiple tables and creating a custom display of those tables in a way that cannot be directly done by PROC GLM.

The following statements create a SAS data set named Histamine that contains the experimental data:

```
title1 'Histamine Study';

data Histamine;
   input Drug $12. Depleted $ hist0 hist1 hist3 hist5;
   logHist0 = log(hist0); logHist1 = log(Hist1);
   logHist3 = log(hist3); logHist5 = log(Hist5);
   datalines;
Morphine      N  .04  .20  .10  .08
Morphine      N  .02  .06  .02  .02
Morphine      N  .07 1.40  .48  .24
Morphine      N  .17  .57  .35  .24
```

```
Morphine      Y  .10  .09  .13  .14
Morphine      Y  .07  .07  .06  .07
Morphine      Y  .05  .07  .06  .07
Trimethaphan  N  .03  .62  .31  .22
Trimethaphan  N  .03 1.05  .73  .60
Trimethaphan  N  .07  .83 1.07  .80
Trimethaphan  N  .09 3.13 2.06 1.23
Trimethaphan  Y  .10  .09  .09  .08
Trimethaphan  Y  .08  .09  .09  .10
Trimethaphan  Y  .13  .10  .12  .12
Trimethaphan  Y  .06  .05  .05  .05
;
```

The data set comes from a preclinical drug experiment (Cole and Grizzle 1966). In order to study the effect of two different drugs on histamine levels in the blood, researchers administer the drugs to 13 animals, and the levels of histamine in the animals' blood is measured after 0, 1, 3, and 5 minutes. The response variable is the logarithm of the histamine level.

In the analysis that follows, PROC GLM is used to perform a repeated measures analysis, naming the drug and depletion status as between-subject factors in the MODEL statement and naming post-administration measurement time as the within-subject factor. For more information about this study and its analysis, see Example 39.7 in Chapter 39, "The GLM Procedure." Here are the PROC GLM statements that you would run to perform the analysis:

```
ods trace output;

proc glm data=Histamine;
   class Drug Depleted;
   model LogHist0--LogHist5 = Drug Depleted Drug*Depleted / nouni;
   repeated Time 4 (0 1 3 5) polynomial / summary printe;
run; quit;
```

The portion of the trace output containing the fully qualified paths is shown next:

```
Path:       GLM.Data.ClassLevels
Path:       GLM.Data.NObs
Path:       GLM.Repeated.RepeatedLevelInfo
Path:       GLM.Repeated.PartialCorr
Path:       GLM.Repeated.MANOVA.Model.Error.ErrorSSCP
Path:       GLM.Repeated.MANOVA.Model.Error.PartialCorr
Path:       GLM.Repeated.MANOVA.Model.Error.Sphericity
Path:       GLM.Repeated.MANOVA.Model.Time.MultStat
Path:       GLM.Repeated.MANOVA.Model.Time_Drug.MultStat
Path:       GLM.Repeated.MANOVA.Model.Time_Depleted.MultStat
Path:       GLM.Repeated.MANOVA.Model.Time_Drug_Depleted.MultStat
Path:       GLM.Repeated.BetweenSubjects.ModelANOVA
Path:       GLM.Repeated.WithinSubject.ModelANOVA
Path:       GLM.Repeated.WithinSubject.Epsilons
Path:       GLM.Repeated.Summary.Time_1.ModelANOVA
Path:       GLM.Repeated.Summary.Time_2.ModelANOVA
Path:       GLM.Repeated.Summary.Time_3.ModelANOVA
```

The goal here is to output the within-subjects multivariate statistics and the between-subjects ANOVA table to SAS data sets for use in subsequent steps. The following statements run the analysis and save the desired results to output data sets:

```
ods listing close;

proc glm data=Histamine;
   class Drug Depleted;
   model LogHist0--LogHist5 = Drug Depleted Drug*Depleted / nouni;
   repeated Time 4 (0 1 3 5) polynomial / summary printe;
   ods output MultStat                    = HistWithin
              BetweenSubjects.ModelANOVA = HistBetween;
run; quit;

ods listing;
```

The LISTING destination is closed so that no output is displayed. The ODS OUTPUT statement creates two SAS data sets, named HistWithin and HistBetween, from the two ODS tables. This analysis creates the following tables:

```
Path:       GLM.Repeated.MANOVA.Model.Time.MultStat
Path:       GLM.Repeated.MANOVA.Model.Time_Drug.MultStat
Path:       GLM.Repeated.MANOVA.Model.Time_Depleted.MultStat
Path:       GLM.Repeated.MANOVA.Model.Time_Drug_Depleted.MultStat
Path:       GLM.Repeated.BetweenSubjects.ModelANOVA
```

Here is the full trace output for the model ANOVA table:

```
Output Added:
-------------
Name:       ModelANOVA
Label:      Type III Model ANOVA
Template:   stat.GLM.Tests
Path:       GLM.Repeated.BetweenSubjects.ModelANOVA
-------------
```

All of the multivariate test results are routed to the HistWithin data set. This is because all multivariate test tables are named **MultStat**, even though they occur in different directories in the output directory hierarchy. Only the between-subject ANOVA table appears in the HistBetween data set, even though there are also other tables named **ModelANOVA**. ODS selects just the one specific table for the HistBetween data set because of the partial path (**BetweenSubjects.ModelANOVA**) in the second specification. For more information about names and qualified path names, see the discussion in the section "Paths and Selection" on page 435.

The following statements show the names and the variable labels for the two data sets and produce Output 20.7.1:

```
proc contents data=HistBetween p;
   ods select position;
run;
```

```
proc contents data=HistWithin p;
   ods select position;
run;
```

Output 20.7.1 The Variable Names and Labels for the Two Data Sets

Histamine Study

The CONTENTS Procedure

Variables in Creation Order

#	Variable	Type	Len	Format	Label
1	Dependent	Char	15		
2	HypothesisType	Num	8	BEST8.	
3	Source	Char	20		
4	DF	Num	8	BEST6.	
5	SS	Num	8	12.8	Type III SS
6	MS	Num	8	12.8	Mean Square
7	FValue	Num	8	7.2	F Value
8	ProbF	Num	8	PVALUE6.4	Pr > F

Histamine Study

The CONTENTS Procedure

Variables in Creation Order

#	Variable	Type	Len	Format	Label
1	Hypothesis	Char	32		
2	Error	Char	55		
3	Statistic	Char	22		
4	Value	Num	8	12.8	
5	FValue	Num	8	7.2	F Value
6	NumDF	Num	8	BEST6.	Num DF
7	DenDF	Num	8	BEST6.	Den DF
8	ProbF	Num	8	PVALUE6.4	Pr > F
9	PValue	Num	8	PVALUE6.4	P-Value

The following statements create a new data set that contains the two data sets created in the preceding PROC GLM step and display the results in Output 20.7.2:

```
ods listing;
title2 'The Combined Data Set';

data temp1;
   set HistBetween HistWithin;
run;

proc print label;
run;
```

Output 20.7.2 Listing of the Combined Data Set: Histamine Study

Histamine Study
The Combined Data Set

Obs	Dependent	Hypothesis Type	Source	DF
1	BetweenSubjects	3	Drug	1
2	BetweenSubjects	3	Depleted	1
3	BetweenSubjects	3	Drug*Depleted	1
4	BetweenSubjects	3	Error	11
5		.		.
6		.		.
7		.		.
8		.		.
9		.		.
10		.		.
11		.		.
12		.		.
13		.		.
14		.		.
15		.		.
16		.		.
17		.		.
18		.		.
19		.		.
20		.		.

Output 20.7.2 *continued*

Histamine Study
The Combined Data Set

Obs	Type III SS	Mean Square	F Value	Pr > F
1	5.99336243	5.99336243	2.71	0.1281
2	15.44840703	15.44840703	6.98	0.0229
3	4.69087508	4.69087508	2.12	0.1734
4	24.34683348	2.21334850	_	_
5	.	.	24.03	0.0001
6	.	.	24.03	0.0001
7	.	.	24.03	0.0001
8	.	.	24.03	0.0001
9	.	.	5.78	0.0175
10	.	.	5.78	0.0175
11	.	.	5.78	0.0175
12	.	.	5.78	0.0175
13	.	.	21.31	0.0002
14	.	.	21.31	0.0002
15	.	.	21.31	0.0002
16	.	.	21.31	0.0002
17	.	.	12.48	0.0015
18	.	.	12.48	0.0015
19	.	.	12.48	0.0015
20	.	.	12.48	0.0015

Obs	Hypothesis	Error
1		
2		
3		
4		
5	Time	Error SSCP Matrix
6	Time	Error SSCP Matrix
7	Time	Error SSCP Matrix
8	Time	Error SSCP Matrix
9	Time_Drug	Error SSCP Matrix
10	Time_Drug	Error SSCP Matrix
11	Time_Drug	Error SSCP Matrix
12	Time_Drug	Error SSCP Matrix
13	Time_Depleted	Error SSCP Matrix
14	Time_Depleted	Error SSCP Matrix
15	Time_Depleted	Error SSCP Matrix
16	Time_Depleted	Error SSCP Matrix
17	Time_Drug_Depleted	Error SSCP Matrix
18	Time_Drug_Depleted	Error SSCP Matrix
19	Time_Drug_Depleted	Error SSCP Matrix
20	Time_Drug_Depleted	Error SSCP Matrix

Output 20.7.2 *continued*

Histamine Study
The Combined Data Set

Obs	Statistic	Value	Num DF	Den DF	P-Value
1		.	.	.	.
2		.	.	.	.
3		.	.	.	.
4		.	.	.	.
5	Wilks' Lambda	0.11097706	3	9	.
6	Pillai's Trace	0.88902294	3	9	.
7	Hotelling-Lawley Trace	8.01087137	3	9	.
8	Roy's Greatest Root	8.01087137	3	9	.
9	Wilks' Lambda	0.34155984	3	9	.
10	Pillai's Trace	0.65844016	3	9	.
11	Hotelling-Lawley Trace	1.92774470	3	9	.
12	Roy's Greatest Root	1.92774470	3	9	.
13	Wilks' Lambda	0.12339988	3	9	.
14	Pillai's Trace	0.87660012	3	9	.
15	Hotelling-Lawley Trace	7.10373567	3	9	.
16	Roy's Greatest Root	7.10373567	3	9	.
17	Wilks' Lambda	0.19383010	3	9	.
18	Pillai's Trace	0.80616990	3	9	.
19	Hotelling-Lawley Trace	4.15915732	3	9	.
20	Roy's Greatest Root	4.15915732	3	9	.

The next steps are designed to produce a more parsimonious display of the most important information in Output 20.7.2. The next step creates a data set named HistTests. Only the observations from the input data sets that are needed for interpretation are included. The variable Hypothesis in the HistWithin data set is renamed Source, and the NumDF variable is renamed DF. The renamed variables correspond to the variable names found in the HistBetween data set. These names are chosen since the template for the **ModelANOVA** table will be used in subsequent steps. An explicit length for the new variable Source is provided since the input variables, Hypothesis and Source, have different lengths. The following statements produce Output 20.7.3:

```
data HistTests;
   length Source $ 20;
   set HistBetween(where =(Source    ^= 'Error'))
       HistWithin (rename=(Hypothesis =  Source NumDF=DF)
                   where =(Statistic  = 'Hotelling-Lawley Trace'));
run;

proc print label;
   title2 'Listing of the Combined Data Set';
run;
```

Output 20.7.3 Listing of the HistTests Data Set: Histamine Study

Histamine Study
Listing of the Combined Data Set

Obs	Source	Dependent	Hypothesis Type	Num DF	Type III SS
1	Drug	BetweenSubjects	3	1	5.99336243
2	Depleted	BetweenSubjects	3	1	15.44840703
3	Drug*Depleted	BetweenSubjects	3	1	4.69087508
4	Time		.	3	.
5	Time_Drug		.	3	.
6	Time_Depleted		.	3	.
7	Time_Drug_Depleted		.	3	.

Obs	Mean Square	F Value	Pr > F	Error	Statistic
1	5.99336243	2.71	0.1281		
2	15.44840703	6.98	0.0229		
3	4.69087508	2.12	0.1734		
4	.	24.03	0.0001	Error SSCP Matrix	Hotelling-Lawley Trace
5	.	5.78	0.0175	Error SSCP Matrix	Hotelling-Lawley Trace
6	.	21.31	0.0002	Error SSCP Matrix	Hotelling-Lawley Trace
7	.	12.48	0.0015	Error SSCP Matrix	Hotelling-Lawley Trace

Obs	Value	Den DF	P-Value
1	.	.	.
2	.	.	.
3	.	.	.
4	8.01087137	9	.
5	1.92774470	9	.
6	7.10373567	9	.
7	4.15915732	9	.

The amount of information contained in the HistTests data set is appropriate for interpreting the analysis; however, there is still extra information, and the information of interest is not being displayed in a compact or useful form. This data set consists of multiple tables, an ANOVA table with between-subjects information, and multivariate statistics tables with the variables renamed to match the names in the ANOVA table. This form was chosen so that the data set could be displayed using PROC GLM's ANOVA template. A template specifies how the data set should be displayed and which columns should be displayed. The output from the ODS TRACE statements shows that the template associated with PROC GLM's ANOVA table is named **`Stat.GLM.Tests`**. You can use the **`Stat.GLM.Tests`** template to display the SAS data set HistTests, as follows:

```
title2 'Listing of the Selections, Using a Standard Template';
proc sgrender data=histtests template=Stat.GLM.Tests;
run;
```

The SGRENDER procedure displays the DATA= data set with the specified TEMPLATE= template. (Note that you can use PROC SGRENDER to display both graphs and tables.) The results are displayed in Output 20.7.4.

Output 20.7.4 Listing of the Data Set Using a Standard PROC GLM ANOVA Template

Histamine Study
Listing of the Selections, Using a Standard Template

Source	DF	SS	Mean Square	F Value	Pr > F
Drug	1	5.99336243	5.99336243	2.71	0.1281
Depleted	1	15.44840703	15.44840703	6.98	0.0229
Drug*Depleted	1	4.69087508	4.69087508	2.12	0.1734
Time	3	.	.	24.03	0.0001
Time_Drug	3	.	.	5.78	0.0175
Time_Depleted	3	.	.	21.31	0.0002
Time_Drug_Depleted	3	.	.	12.48	0.0015

Alternatively and equivalently, you could display the results by using a DATA step as follows:

```
title2 'Listing of the Selections, Using a Standard Template';

data _null_;
   set histtests;
   file print ods=(template='Stat.GLM.Tests');
   put _ods_;
run;
```

The next steps will create a final display of these results, this time by using a custom template. This example shows you how to use PROC TEMPLATE to do the following:

- redefine the format for the SS and Mean Square columns
- include the table title and footnote in the body of the table
- translate the missing values for SS and Mean Square in the rows corresponding to multivariate tests to asterisks (to see the footnote)
- add a column depicting the level of significance of each effect

The following statements create a custom template:

```
proc template;
   define table CombinedTests;
      parent=Stat.GLM.Tests;

      header '#Histamine Study##';
      footer '#* - Test computed using Hotelling-Lawley trace';

      column Source DF SS MS FValue ProbF Star;

      define Source; width=20; end;
      define DF; format=bestd3.; end;
      define SS;
         parent=Stat.GLM.SS
         choose_format=max format_width=7;
         translate _val_ = . into '  *';
      end;
      define MS;
         parent=Stat.GLM.MS
         choose_format=max format_width=7;
         translate _val_ = . into '  *';
      end;
      define Star;
         compute as ProbF;
         translate _val_ <= 0.001 into 'xxx',
                   _val_ <= 0.01  into 'xx',
                   _val_ <= 0.05  into 'x',
                   _val_ >  0.05  into '';
         pre_space=1 width=3 just=l;
      end;
   end;
run;
```

The CHOOSE_FORMAT=MAX option along with FORMAT_WIDTH=7 chooses the format for each column based on the maximum value and an overall width of 7. Alternatively, you could have specified a format directly by specifying, for example, FORMAT=7.2 or FORMAT=D8.3. The TRANSLATE statements provide values to display in place of the original values. The first two TRANSLATE statements display missing values as an asterisk with leading blanks added to ensure alignment with the decimal place. The third TRANSLATE statement displays *p*-values greater than 0.05 as a blank, values greater than 0.01 but less than or equal to 0.05 as a single 'x', and so on. Note that the ProbF column is printed twice—once in the usual way as a numeric column with a PVALUE format and once with a column of blanks or x's. For detailed information about PROC TEMPLATE, see "The Template Procedure" in the *SAS Output Delivery System: User's Guide*. The following statements display the HistTests data set by using the customized template:

```
title2 'Listing of the Selections, Using a Customized Template';

proc sgrender data=HistTests template=CombinedTests;
run;
```

The results are displayed in Output 20.7.5.

Output 20.7.5 Display of the Data Sets Using a Customized Template: Histamine Study

Histamine Study
Listing of the Selections, Using a Customized Template

Histamine Study

	Num DF	Sum of Squares	Mean Square	F Value	Pr > F	
Drug	1	5.9934	5.9934	2.71	0.1281	
Depleted	1	15.4484	15.4484	6.98	0.0229	x
Drug*Depleted	1	4.6909	4.6909	2.12	0.1734	
Time	3	*	*	24.03	0.0001	xxx
Time_Drug	3	*	*	5.78	0.0175	x
Time_Depleted	3	*	*	21.31	0.0002	xxx
Time_Drug_Depleted	3	*	*	12.48	0.0015	xx

* - Test computed using Hotelling-Lawley trace

These next steps display the same table, but this time changing the background color for the entire row to highlight effects with p-values < 0.001 and also those with p-values < 0.01. The table is displayed three times. Output 20.7.6 displays the results by using bold green and yellow backgrounds and a bold font. Output 20.7.7 displays the results by using much subtler cyan and yellow backgrounds and a bold font. Output 20.7.8 displays the results by using very subtle cyan and gray backgrounds and a normal font. This control is provided by the CELLSTYLE statement in PROC TEMPLATE. There are many things you can do with the CELLSTYLE statement to enhance your output. Several more will be shown in other examples in this chapter. These next steps create the custom template with varying colors and fonts, and display the results by using PROC SGRENDER:

```
%macro hilight(c1,c2);
   proc template;
        define table CombinedTests;
           parent=Stat.GLM.Tests;

           header '#Histamine Study##';
           footer '#* - Test computed using Hotelling-Lawley trace';

           column Source DF SS MS FValue ProbF;

           cellstyle probf <= 0.001 as {background=&c1},
                     probf <= 0.01  as {background=&c2};

           define DF; format=bestd3.; end;
           define SS;
              parent=Stat.GLM.SS
              choose_format=max format_width=7;
              translate _val_ = . into '  *';
           end;
           define MS;
              parent=Stat.GLM.MS
              choose_format=max format_width=7;
```

```
            translate _val_ = . into '   *';
         end;
      end;
   run;

   proc sgrender data=HistTests template=CombinedTests;
   run;
%mend;

ods html style=statistical;

%hilight(CX22FF22 fontweight=bold, CXFFFF22 fontweight=bold)
%hilight(CXAAFFFF fontweight=bold, CXFFFFDD fontweight=bold)
%hilight(CXEEFAFA, CXEEEEEE)

ods html close;
```

Output 20.7.6 Rows Boldly Highlighted: Histamine Study

Histamine Study

Histamine Study					
Source	Num DF	Sum of Squares	Mean Square	F Value	Pr > F
Drug	1	5.9934	5.9934	2.71	0.1281
Depleted	1	15.4484	15.4484	6.98	0.0229
Drug*Depleted	1	4.6909	4.6909	2.12	0.1734
Time	**3**	*	*	**24.03**	**0.0001**
Time_Drug	3	*	*	5.78	0.0175
Time_Depleted	**3**	*	*	**21.31**	**0.0002**
Time_Drug_Depleted	**3**	*	*	**12.48**	**0.0015**
* - Test computed using Hotelling-Lawley trace					

Output 20.7.7 Rows Subtly Highlighted: Histamine Study

Histamine Study

Histamine Study					
Source	Num DF	Sum of Squares	Mean Square	F Value	Pr > F
Drug	1	5.9934	5.9934	2.71	0.1281
Depleted	1	15.4484	15.4484	6.98	0.0229
Drug*Depleted	1	4.6909	4.6909	2.12	0.1734
Time	**3**	*	*	**24.03**	**0.0001**
Time_Drug	3	*	*	5.78	0.0175
Time_Depleted	**3**	*	*	**21.31**	**0.0002**
Time_Drug_Depleted	**3**	*	*	**12.48**	**0.0015**
* - Test computed using Hotelling-Lawley trace					

Output 20.7.8 Rows Very Subtly Highlighted: Histamine Study

Histamine Study

Histamine Study					
Source	Num DF	Sum of Squares	Mean Square	F Value	Pr > F
Drug	1	5.9934	5.9934	2.71	0.1281
Depleted	1	15.4484	15.4484	6.98	0.0229
Drug*Depleted	1	4.6909	4.6909	2.12	0.1734
Time	3	*	*	24.03	0.0001
Time_Drug	3	*	*	5.78	0.0175
Time_Depleted	3	*	*	21.31	0.0002
Time_Drug_Depleted	3	*	*	12.48	0.0015
* - Test computed using Hotelling-Lawley trace					

SAS color values can be expressed as RGB (red, green, blue) values of the form CXRRGGBB. RR, GG, and BB are all integers in the range 0 to 255 but expressed in a HEX2. format (00 to FF) that specify the red, green, and blue values, respectively. You can run the following steps to see the correspondence between the integer and HEX formatting of values in the range 0 to 255:

```
data _null_;
   do color = 0 to 255;
      put color 3. +1 color hex2.;
      end;
   run;
```

The results of this step are not shown. Hexadecimal values 0 through F represent the numbers 0 to 15. A hex value xy can be converted to an integer as follows: $16x + y$. For example, BC is $16 \times 11 + 12 = 188$. Common colors are CXFF0000 (red), CX00FF00 (green), CX0000FF (blue), CXFFFF00 (yellow, a mix of red and green), CXFF00FF (magenta, a mix of red and blue), CX00FFFF (cyan, a mix of green and blue), CXFFFFFF (white, a mix of red, green, and blue), CX000000 (black, no color), CXDDDDDD (very light gray), CX222222 (very dark gray), and so on. Colors become lighter as the RGB values increase and darker as they decrease. For example, cyan (CX00FFFF) can be lightened by increasing the red component from 00 to FF until eventually it becomes indistinguishable from white. It can be darkened by jointly decreasing the green and blue values until it becomes indistinguishable from black.

The three CELLSTYLE statements that set the colors after the macro variables are substituted are as follows:

```
cellstyle probf <= 0.001 as {background=CX22FF22 fontweight=bold},
          probf <= 0.01  as {background=CXFFFF22 fontweight=bold};
cellstyle probf <= 0.001 as {background=CXAAFFFF fontweight=bold},
          probf <= 0.01  as {background=CXFFFFDD fontweight=bold};
cellstyle probf <= 0.001 as {background=CXEEFAFA},
          probf <= 0.01  as {background=CXEEEEEE};
```

The first color, CX22FF22, for the smallest p-values in the first table is a bold green color. The first table uses almost pure green and pure yellow, but a little red and blue are added to slightly lighten the colors. The second table uses a cyan and yellow that are very light due to the addition of AA

(170) red and DD (221) blue, respectively. The third table uses a cyan that is not much different from light gray, and a light gray that is not much different from white.

Example 20.8: HTML Output with Hyperlinks between Tables

This example demonstrates how you can use ODS to provide links between different parts of your HTML procedure output. In this example, a table is created where each row contains a link to another table with more information about that row.

Suppose that you are analyzing a 4×4 factorial experiment for an industrial process, testing for differences in the number of defective products manufactured by different machines, using different sources of raw material. The data set Experiment is created as follows:

```
title 'Product Defects Experiment';

data Experiment;
   do Supplier = 'A', 'B', 'C', 'D';
      do Machine = 1 to 4;
         do rep = 1 to 5;
            input Defects @@;
            output;
            end;
         end;
      end;
   datalines;
 2  6  3  3  6  8  6  6  4  4  4  2  4  0  4  5  5  7  8  5
13 12 12 11 12 16 15 14 14 13 11 10 12 12 10 13 13 14 15 12
 2  6  3  6  6  6  4  4  6  6  0  3  2  0  2  4  6  7  6  4
20 19 18 21 22 22 24 23 20 20 17 19 18 16 17 23 20 20 22 21
;
```

Suppose that you are interested in fitting a model to determine the effect that the supplier of raw material and machine type have on the number of defects in the products. If the F test for a factor is significant, you might like to follow up with a multiple-comparison test for the levels of that factor. The tables of interest are the model ANOVA and the multiple-comparison output. Since this is a balanced experiment, the ANOVA procedure computes the appropriate analysis. The following statements produce these tables and Figure 20.8.1:

```
ods listing close;
ods html body='anovab.htm' style=statistical anchor='anova1';
ods trace output;

proc anova data=Experiment;
   ods select ModelANOVA MCLines;
   class Supplier Machine;
   model Defects = Supplier Machine;
   means Supplier Machine / tukey;
run; quit;
```

```
ods html close;
ods listing;
```

The LISTING destination is closed to avoid generating the output twice. ODS writes the HTML output to the file *anovab.htm*. The ANCHOR= option specifies *anova1* as the root name for the HTML anchor tags. This means that within the HTML document, the URL for the first table will be *anova1*, the URL for the second table will be *anova2*, and so on.

Output 20.8.1 ANOVA and Multiple-Comparison Results: Histamine Study

```
                          Product Defects Experiment

                             The ANOVA Procedure

Dependent Variable: Defects

 Source                      DF       Anova SS    Mean Square   F Value   Pr > F

 Supplier                     3    3441.637500    1147.212500    580.72   <.0001
 Machine                      3     163.137500      54.379167     27.53   <.0001

                          Product Defects Experiment

                             The ANOVA Procedure

               Tukey's Studentized Range (HSD) Test for Defects

           Means with the same letter are not significantly different.


          Tukey Grouping          Mean      N    Supplier

                       A       20.1000     20    D

                       B       12.7000     20    B

                       C        4.6000     20    A
                       C
                       C        4.1500     20    C
```

Output 20.8.1 *continued*

Product Defects Experiment

The ANOVA Procedure

Tukey's Studentized Range (HSD) Test for Defects

Means with the same letter are not significantly different.

Tukey Grouping	Mean	N	Machine
A	11.7500	20	2
A			
A	11.5000	20	4
B	10.1500	20	1
C	8.1500	20	3

The ODS trace output (not shown) shows that PROC ANOVA uses the **Stat.GLM.Tests** template to format the ANOVA table. The following statements demonstrate how you can link a row of the ANOVA table to the corresponding multiple-comparison table by modifying the table template, using the original values and the URLs for the second and third tables (*anova2* and *anova3*):

```
proc template;
   edit Stat.GLM.Tests;
      edit Source;
         cellstyle _val_ = 'Supplier' as {url="#ANOVA2"},
                   _val_ = 'Machine'  as {url="#ANOVA3"};
      end;
   end;
run;
```

This template alters the values in the **Source** column ('Supplier' and 'Machine') of the ANOVA tests table by using the CELLSTYLE statement. The values of 'Supplier' and 'Machine' are displayed as hyperlinks in the HTML, and clicking them takes you to the links *anova2* and *anova3*, which are the multiple-comparison tables.

You can see the value to use in the URL by viewing the HTML source file, *anovab.htm*. You can either open the HTML file in a text editor or view it in a browser window and select **View ► Source**. Search for '<a name=' to find the URL names. The first table is *anova1*, the second is *anova2*, the third is *anova3*, and so on. If the ANCHOR= option had not been used in the ODS HTML statement, the names would have been *IDX*, *IDX1*, *IDX2*, and so on. Note that if you do not use the ODS SELECT statement, or if you do anything to change the tables that come out, the names will be different. The statements create the *Supplier* label as a link that enables you to open the table of means from the "Tukey's Studentized Range Test for Defects" associated with the Supplier variable. Similarly, *Machine* provides a link to the table of means from the "Tukey's Studentized Range Test for Defects" associated with the Machine variable.

Next, the analysis is run again, this time by using the modified template. The following statements produce the results:

```
ods listing close;
ods html body='anovab.htm' style=statistical anchor='anova1';

proc anova data=Experiment;
   ods select ModelANOVA MCLines;
   class Supplier Machine;
   model Defects = Supplier Machine;
   means Supplier Machine / tukey;
run; quit;

ods html close;
ods listing;
```

The ANOVA table is displayed in Output 20.8.2.

Output 20.8.2 HTML Output from PROC ANOVA: Linked Output

The ANOVA Procedure

Dependent Variable: Defects

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	3604.775000	600.795833	304.12	<.0001
Error	73	144.212500	1.975514		
Corrected Total	79	3748.987500			

R-Square	Coeff Var	Root MSE	Defects Mean
0.961533	13.53097	1.405530	10.38750

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Supplier	3	3441.637500	1147.212500	580.72	<.0001
Machine	3	163.137500	54.379167	27.53	<.0001

The underlined text displayed in Output 20.8.2 shows the links, *Supplier* and *Machine*, that you created with the modified template. When you click a link, the appropriate multiple-comparison table opens in your browser. Output 20.8.3 shows the table from the *Supplier* link.

Output 20.8.3 Linked Output: Multiple-Comparison Table from PROC ANOVA

Means with the same letter are not significantly different.			
Tukey Grouping	**Mean**	**N**	**Supplier**
A	20.1000	20	D
B	12.7000	20	B
C	4.6000	20	A
C			
C	4.1500	20	C

When you run the PROC TEMPLATE step shown previously, the following note is printed in the SAS log:

```
NOTE: TABLE 'Stat.GLM.Tests' has been saved to: SASUSER.TEMPLAT
```

You can see that there are now two versions of the template by running the following statements:

```
proc template;
   list Stat.GLM.Tests;
run;
```

These statements produce Output 20.8.4.

Output 20.8.4 The Templates

```
                    Product Defects Experiment

                Listing of: SASUSER.TEMPLAT
                Path Filter is: Stat.GLM.Tests
                Sort by: PATH/ASCENDING

                Obs    Path                Type
                -------------------------------
                 1     Stat.GLM.Tests      Table

                Listing of: SASHELP.TMPLMST
                Path Filter is: Stat.GLM.Tests
                Sort by: PATH/ASCENDING

                Obs    Path                Type
                -------------------------------
                 1     Stat.GLM.Tests      Table
```

You can delete your custom template and restore the default template as follows:

```
proc template;
   delete Stat.GLM.Tests;
run;
```

The following note is printed in the SAS log:

```
NOTE: 'Stat.GLM.Tests' has been deleted from: SASUSER.TEMPLAT
```

Example 20.9: HTML Output with Graphics and Hyperlinks

This example demonstrates how you can use ODS to create links between each bar in a bar chart (Output 20.9.1) and other parts of the analysis (Output 20.9.2). The data in this example are selected from a larger experiment on the use of drugs in the treatment of leprosy (Snedecor and Cochran 1967, p. 422). Variables in the study are as follows:

Drug	two antibiotics ('a' and 'd') and a control ('f')
PreTreatment	a pretreatment score of leprosy bacilli
PostTreatment	a posttreatment score of leprosy bacilli

The data set is created as follows:

```
title 'Treatment of Leprosy';

data drugtest;
   input Drug $ PreTreatment PostTreatment @@;
   datalines;
a 11  6  a  8  0  a  5  2  a 14  8  a 19 11
a  6  4  a 10 13  a  6  1  a 11  8  a  3  0
d  6  0  d  6  2  d  7  3  d  8  1  d 18 18
d  8  4  d 19 14  d  8  9  d  5  1  d 15  9
f 16 13  f 13 10  f 11 18  f  9  5  f 21 23
f 16 12  f 12  5  f 12 16  f  7  1  f 12 20
;
```

The following statement opens the HTML destination:

```
ods html body='glmb.htm' contents='glmc.htm' frame='glmf.htm'
         style=statistical;
```

The ODS HTML statement specifies the body filename, generates a table of contents for the output, and generates a frame to contain the body and table of contents. The following statements perform the analysis:

```
proc glm data=drugtest;
   class drug;
   model PostTreatment = drug | PreTreatment / solution;
   lsmeans drug / stderr pdiff;
   ods output LSMeans=lsmeans;
run; quit;
```

The ODS OUTPUT statement writes the table of LS-means to the data set named lsmeans. PROC GLM performs an analysis of covariance and computes LS-means for the variable Drug.

The following steps demonstrate how you can create links to connect the results of different analyses. In this example, the table of LS-means is graphically summarized in a horizontal bar chart. Each bar is linked to a plot that displays the relationship between the PostTreatment response variable and the PreTreatment variable for the drug corresponding to the bar. Note that PROC GLM can use ODS Graphics to create LS-means graphs that are different from the one constructed here. You do not have to run the following steps to get PROC GLM's standard LS-means plots. The following DATA step creates a new variable named DrugClick that matches each drug value with an HTML file:

```
data lsmeans;
   set lsmeans;
   if drug='a' then DrugClick='drug1.htm';
   if drug='d' then DrugClick='drug2.htm';
   if drug='f' then DrugClick='drug3.htm';
run;
```

The variable DrugClick is used in the chart. The variable provides the connection information for linking the two parts of the analysis together. The files referred to in these statements are created in a later step. The following statements create the chart:

```
ods graphics / imagemap=yes height=2in width=6.4in;

proc sgplot data=lsmeans;
   title 'Chart of LS-Means for Drug Type';
   hbar drug / response=lsmean stat=mean
               url=drugclick;
   footnote j=l 'Click on the bar to see a plot of PostTreatment '
                'versus PreTreatment for the corresponding drug.';
   format lsmean 6.3;
run;

ods graphics off;
footnote;
ods html close;
```

The chart is created with the ODS Graphics procedure SGPLOT. For more information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." The ODS GRAPHICS statement is not required before you run SG procedures. However, in this case, it is necessary to specify IMAGEMAP=YES so that the URL= option will work properly. The size of the graph is also specified with the HEIGHT= and WIDTH= options. PROC SGPLOT is used, and the HBAR statement requests a horizontal bar chart for the variable Drug. The lengths of the bars represent the values of the

LSMean variable. The URL= option specifies the variable DrugClick as the HTML linking variable. The FOOTNOTE statement provides text that indicates how to use the links in the graph.

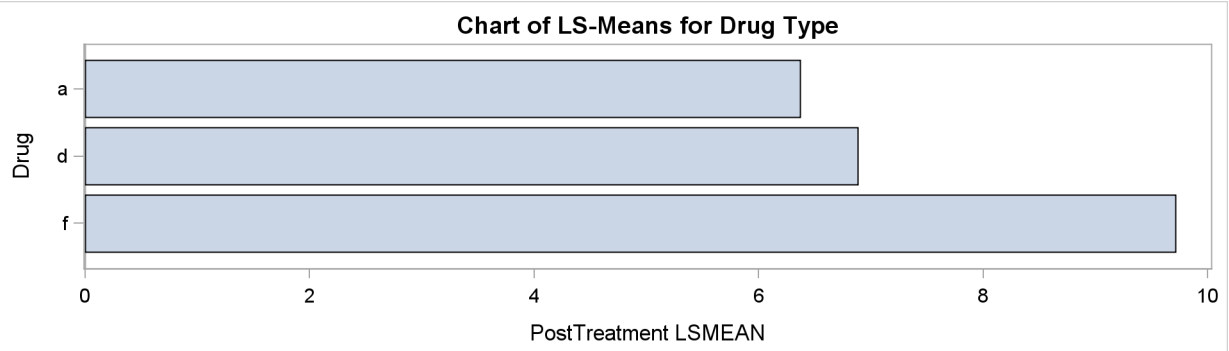
The following statements provide that second analysis. The three files referred to by the DrugClick variable are created as follows:

```
ods html body='drug1.htm' newfile=page style=statistical;

proc sgplot data=drugtest;
   title 'Plot of PostTreatment versus PreTreatment';
   scatter y=PostTreatment x=PreTreatment;
   by drug notsorted;
run;
ods html close;
```

The NEWFILE= option in the ODS HTML statement creates a new HTML file for each page of output. Note that page breaks occur only when a procedure explicitly starts a new page. The NEWFILE= option also increments the filename numeric suffix for each new HTML file created, with the first filename corresponding to that given in the BODY= option, *drug1.htm*.

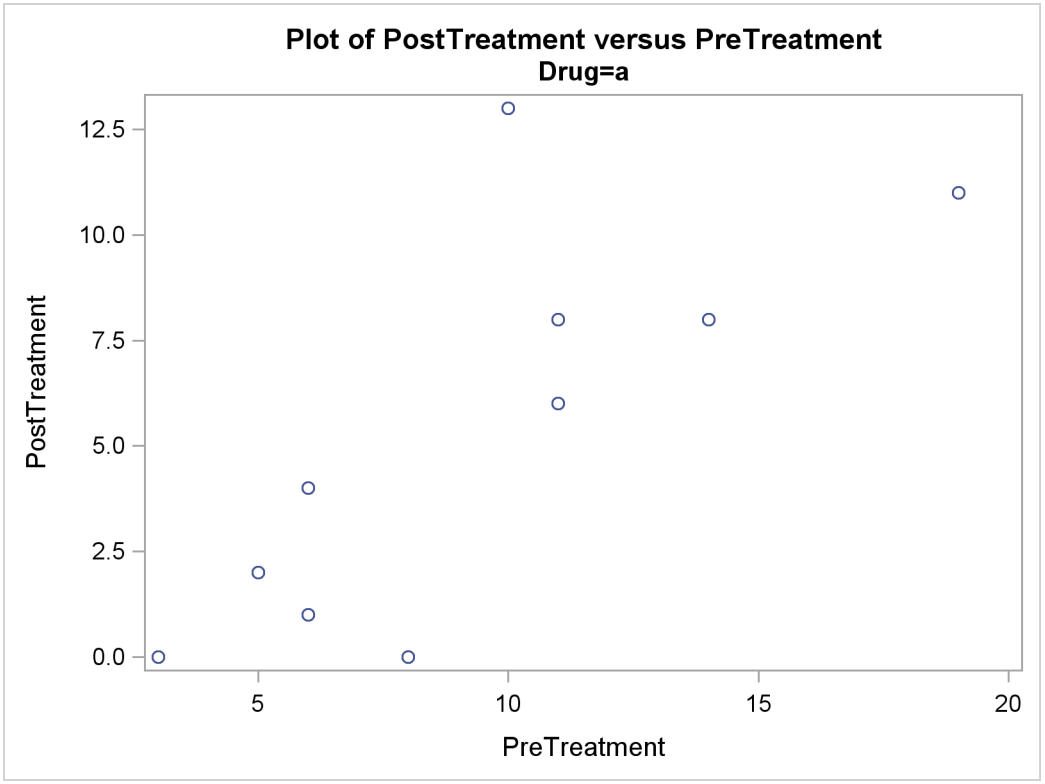
PROC SGPLOT is used, producing a plot of the variable PostTreatment versus the variable PreTreatment for each value of the Drug variable. Three plots are created, and each plot is contained in a separate HTML file. The files are named *drug1.htm*, *drug2.htm*, and *drug3.htm*. The filenames match those filenames specified as values of the DrugClick variable. By default, the HTML files are generated in your current working directory. You can instead specify a path, such as `frame='html/drug2.htm'`, to put a file in a subdirectory. The chart in Output 20.9.1 displays the difference in LS-means for each drug type. When you click on a bar that represents a value of the variable Drug, the browser opens the plot of PostTreatment versus PostTreatment variables that corresponds to that value of the variable Drug. Output 20.9.2 displays the plots for each drug type.

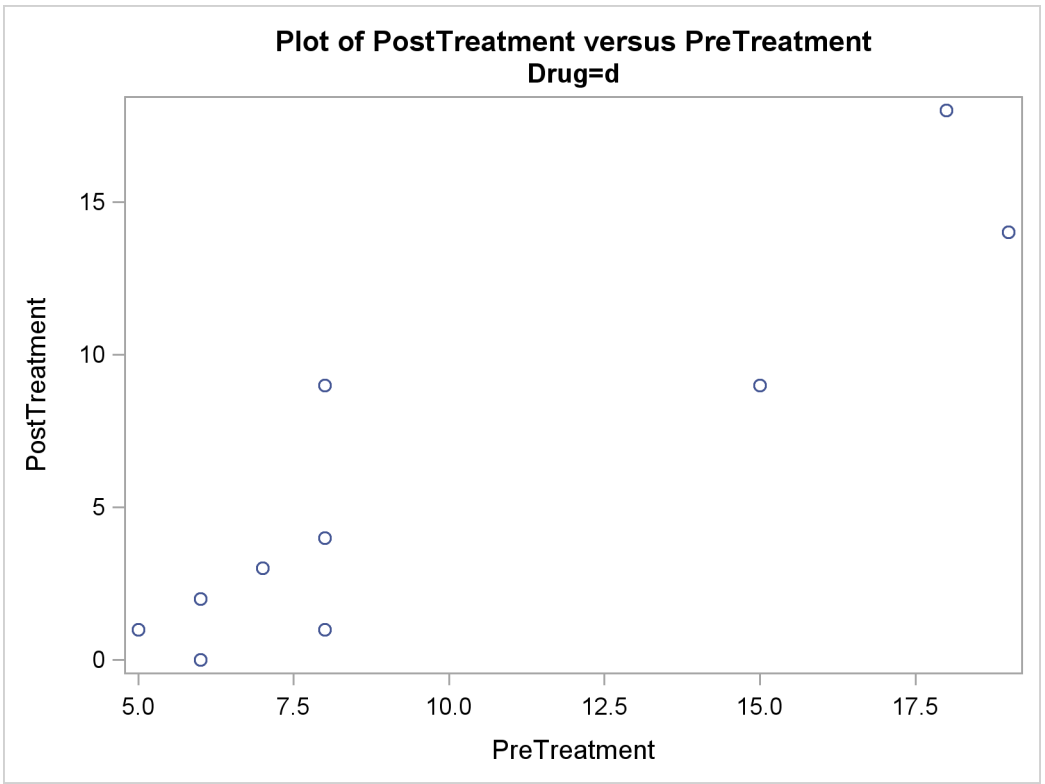
Output 20.9.1 Bar Chart of LS-Means by Drug Type with Links to Plots



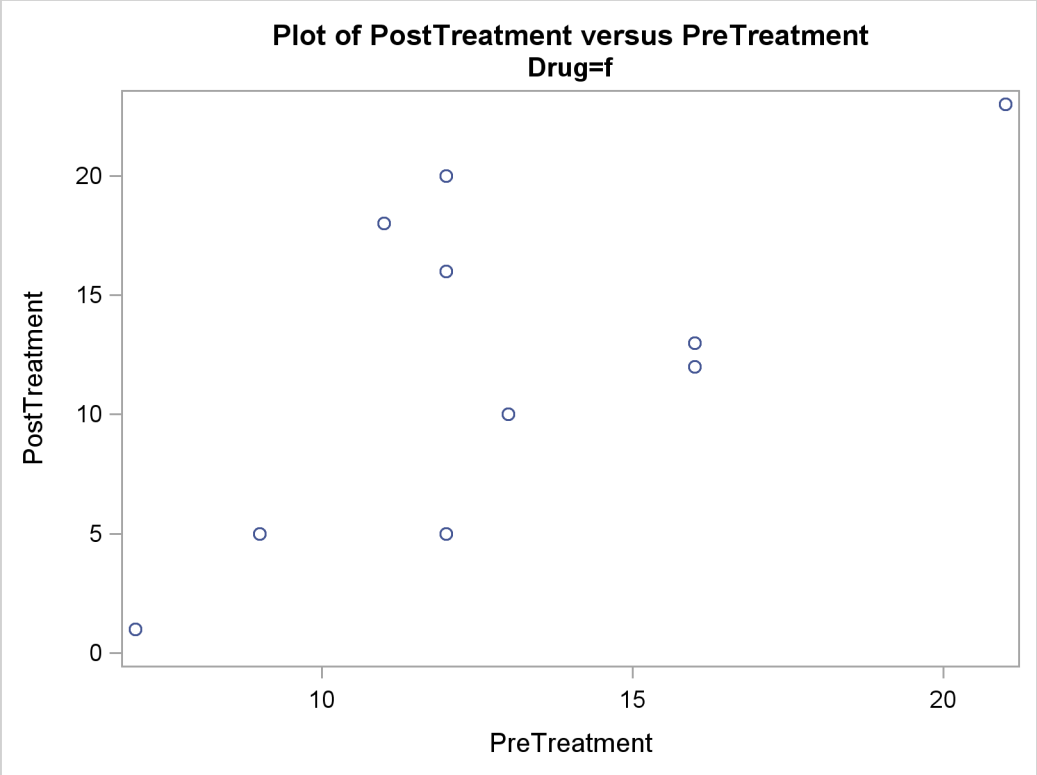
Click on the bar to see a plot of PostTreatment versus PreTreatment for the corresponding drug.

Output 20.9.2 Plots by Drug Type





Output 20.9.2 *continued*



Example 20.10: Correlation and Covariance Matrices

This example demonstrates how you can use ODS to set the background color of individual cells in a table. The color is set to reflect the magnitude of the value in the cell. You can use color to call attention to larger values and to see the pattern in the data in a way that is hard to visualize just by looking at the numbers. This is illustrated with correlation and covariance matrices. The data for this first part of this example are ratings of automobiles. The following statements create the data set:

```
title 'Rating of Automobiles';

data cars;
   input Origin $ 1-8 Make $ 10-19 Model $ 21-36
         (MPG Reliability Acceleration Braking Handling Ride
          Visibility Comfort Quiet Cargo) (1.);
   datalines;
GMC       Buick      Century          3334444544
GMC       Buick      Electra          2434453555

... more lines ...

GMC       Pontiac    Sunbird          3134533234
;
```

The following steps edit the template that PROC CORR uses to display the correlation matrix. The CELLSTYLE statement sets the background color to light gray for correlations equal to 1 or –1. Values less than –0.75 or greater than 0.75 are set to red. Values less than –0.50 or greater than 0.50 are set to blue. Values less than –0.25 or greater than 0.25 are set to cyan. Values in the range –0.25 to 0.25 are set to white. PROC CORR is then run using the custom template. Finally, the default template is restored. The following statements produce Output 20.10.1:

```
proc template;
   edit Base.Corr.StackedMatrix;
      column (RowName RowLabel) (Matrix) * (Matrix2);
      edit matrix;
         cellstyle _val_  = -1.00 as {backgroundcolor=CXEEEEEE},
                   _val_ <= -0.75 as {backgroundcolor=red},
                   _val_ <= -0.50 as {backgroundcolor=blue},
                   _val_ <= -0.25 as {backgroundcolor=cyan},
                   _val_ <=  0.25 as {backgroundcolor=white},
                   _val_ <=  0.50 as {backgroundcolor=cyan},
                   _val_ <=  0.75 as {backgroundcolor=blue},
                   _val_ <   1.00 as {backgroundcolor=red},
                   _val_  =  1.00 as {backgroundcolor=CXEEEEEE};
         end;
      end;
   run;

ods html body='corr.html' style=statistical;
ods listing close;
proc corr data=cars noprob;
   ods select PearsonCorr;
run;
ods listing;
ods html close;

proc template;
   delete Base.Corr.StackedMatrix;
run;
```

Output 20.10.1 Correlation Matrix from PROC CORR

Rating of Automobiles

The CORR Procedure

Pearson Correlation Coefficients, N = 50

	MPG	Reliability	Acceleration	Braking	Handling	Ride	Visibility	Comfort	Quiet	Cargo
MPG	1.00000	0.22003	-0.19454	0.41475	0.25594	-0.23705	0.67924	-0.06567	-0.49128	-0.03075
Reliability	0.22003	1.00000	-0.08512	0.25881	-0.09443	0.27406	0.33356	0.36607	0.45302	0.35261
Acceleration	-0.19454	-0.08512	1.00000	0.06688	0.07119	0.33888	-0.13280	0.06369	0.00934	-0.12112
Braking	0.41475	0.25881	0.06688	1.00000	0.22335	0.30309	0.44938	0.26165	0.00164	0.20880
Handling	0.25594	-0.09443	0.07119	0.22335	1.00000	0.12435	0.12599	-0.07516	-0.02418	-0.14274
Ride	-0.23705	0.27406	0.33888	0.30309	0.12435	1.00000	0.16114	0.75173	0.48498	0.39108
Visibility	0.67924	0.33356	-0.13280	0.44938	0.12599	0.16114	1.00000	0.29830	-0.18347	0.35585
Comfort	-0.06567	0.36607	0.06369	0.26165	-0.07516	0.75173	0.29830	1.00000	0.44917	0.53836
Quiet	-0.49128	0.45302	0.00934	0.00164	-0.02418	0.48498	-0.18347	0.44917	1.00000	0.33846
Cargo	-0.03075	0.35261	-0.12112	0.20880	-0.14274	0.39108	0.35585	0.53836	0.33846	1.00000

The preceding statements used a small number of discrete colors to show the range of values. In contrast, the following statements use a color gradient. The SAS autocall macro **Paint** is available for generating the CELLSTYLE colors list with a list of interpolated colors. If your site has installed the autocall libraries supplied by SAS and uses the standard configuration of software supplied by SAS, you need to ensure that the SAS system option MAUTOSOURCE is in effect to begin using autocall macros. The macros do not have to be included (for example, with a %INCLUDE statement). They can be called directly once they are properly installed. For more information about autocall libraries, see *SAS Macro Language: Reference*.

Usually, you can use the **Paint** macro by specifying a list of values and a list of colors. Here is an example for values ranging from 0 to 10:

```
%paint(values=0 to 10 by 0.5,
       colors=white cyan blue magenta red)

proc print data=colors;
run;
```

The **Paint** macro prints the following information to the SAS log:

```
Legend:
         0 = White
       2.5 = Cyan
         5 = Blue
       7.5 = Magenta
        10 = Red
```

A value of 0 maps to white, a value of 2.5 maps to cyan, values in the range 0 to 2.5 map to colors in the range from white to cyan, and so on. The **Paint** macro for this step creates an output data set, Colors, which is shown in Output 20.10.2.

Output 20.10.2 Color Interpolation

Rating of Automobiles

Obs	Start	_RGB_
1	0.0	CXFFFFFF
2	0.5	CXCBFFFF
3	1.0	CX97FFFF
4	1.5	CX63FFFF
5	2.0	CX2FFFFF
6	2.5	CX05FFFF
7	3.0	CX00D1FF
8	3.5	CX009CFF
9	4.0	CX0068FF
10	4.5	CX0034FF
11	5.0	CX0000FF
12	5.5	CX3400FF
13	6.0	CX6800FF
14	6.5	CX9C00FF
15	7.0	CXD100FF
16	7.5	CXFA00FF
17	8.0	CXFF00D1
18	8.5	CXFF009C
19	9.0	CXFF0068
20	9.5	CXFF0034
21	10.0	CXFF0000

This shows the color interpolation for a series of points. You could use a smaller BY value in the **Paint** macro to get more points along the color gradient. Note, however, that a few dozen colors are usually sufficient for most purposes.

The following steps use the **Paint** macro to create a color gradient for a correlation matrix, edit the template, display the results, and restore the default template:

```
%paint(values=-1 to 1 by 0.05, macro=setstyle,
       colors=CXEEEEEE red magenta blue cyan white
              cyan blue magenta red CXEEEEEE
              -1 -0.99 -0.75 -0.5 -0.25 0 0.25 0.5 0.75 0.99 1)

proc template;
   edit Base.Corr.StackedMatrix;
      column (RowName RowLabel) (Matrix) * (Matrix2);
      edit matrix;
         %setstyle(backgroundcolor)
         end;
      end;
   run;

ods html body='corr.html' style=statistical;
ods listing close;
proc corr data=cars noprob;
   ods select PearsonCorr;
run;
```

```
ods listing;
ods html close;

proc template;
   delete Base.Corr.StackedMatrix;
run;
```

The VALUES= option creates a range of values from –1 to 1 with an increment of 0.05. The **Paint** macro generates a CELLSTYLE **_val_ <=** *value* **as {backgroundcolor=** *color***},** line for each value in the list. Specifically, it generates a macro named SETSTYLE, from the MACRO= option, that contains the entire CELLSTYLE statement for use in PROC TEMPLATE. The argument to the macro is the option that you want to set. In this case, it is the background color. You could specify **foreground** instead to set the color of the numbers themselves. The first part of the generated statement is as follows:

```
cellstyle _val_<=-1 as {backgroundcolor=CXEFEEEE},
          _val_<=-0.95 as {backgroundcolor=CXFF0020},
          _val_<=-0.9 as {backgroundcolor=CXFF0062},
          _val_<=-0.85 as {backgroundcolor=CXFF008D},
          _val_<=-0.8 as {backgroundcolor=CXFF00CF},
```

The color mapping for a correlation matrix can be a bit more involved than it is for most tables. This is because you might want the maximum correlations, 1 and –1, to be displayed using colors outside the gradient used for other values. Usually, you specify the color list, and the **Paint** macro maps the first color to the minimum value, the last color to the maximum value, and colors in between using equal increments and values based on the minimum and maximum. Alternatively, you can provide these values, and that is what is done in this example. The legend, displayed in the SAS log, is as follows for the **Paint** macro step:

```
Legend:
      -1 = CXEEEEEE
   -0.99 = Red
   -0.75 = Magenta
    -0.5 = Blue
   -0.25 = Cyan
       0 = White
    0.25 = Cyan
     0.5 = Blue
    0.75 = Magenta
    0.99 = Red
       1 = CXEEEEEE
```

Values in the range –0.99 to 0.99 follow the interpolation red to magenta to blue to cyan to white to cyan to blue to magenta to red. Of course the actual correlations for these data do not span this entire range, so a pure red background will not appear in the matrix. Correlations of 1 and –1 are displayed as light gray. The resulting correlation matrix is displayed in Output 20.10.3. Notice that there are now a number of shades of colors, particularly shades of blues, not just a few discrete colors. The largest values are displayed in shades of purple and magenta.

Output 20.10.3 Correlation Matrix from PROC CORR with a Color Gradient

Rating of Automobiles

The CORR Procedure

Pearson Correlation Coefficients, N = 50

	MPG	Reliability	Acceleration	Braking	Handling	Ride	Visibility	Comfort	Quiet	Cargo
MPG	1.00000	0.22003	-0.19454	0.41475	0.25594	-0.23705	0.67924	-0.06567	-0.49128	-0.03075
Reliability	0.22003	1.00000	-0.08512	0.25881	-0.09443	0.27406	0.33356	0.36607	0.45302	0.35261
Acceleration	-0.19454	-0.08512	1.00000	0.06688	0.07119	0.33888	-0.13280	0.06369	0.00934	-0.12112
Braking	0.41475	0.25881	0.06688	1.00000	0.22335	0.30309	0.44938	0.26165	0.00164	0.20880
Handling	0.25594	-0.09443	0.07119	0.22335	1.00000	0.12435	0.12599	-0.07516	-0.02418	-0.14274
Ride	-0.23705	0.27406	0.33888	0.30309	0.12435	1.00000	0.16114	0.75173	0.48498	0.39108
Visibility	0.67924	0.33356	-0.13280	0.44938	0.12599	0.16114	1.00000	0.29830	-0.18347	0.35585
Comfort	-0.06567	0.36607	0.06369	0.26165	-0.07516	0.75173	0.29830	1.00000	0.44917	0.53836
Quiet	-0.49128	0.45302	0.00934	0.00164	-0.02418	0.48498	-0.18347	0.44917	1.00000	0.33846
Cargo	-0.03075	0.35261	-0.12112	0.20880	-0.14274	0.39108	0.35585	0.53836	0.33846	1.00000

Next, the same technique is used to display the covariance and correlation matrices of a heteroscedastic autoregressive model. The data are based on the famous growth measurement data of Pothoff and Roy (1964), but are modified here to illustrate the technique of painting the entries of a matrix. The original data are used, for example, in the second example of the PROC MIXED documentation. The data consist of four repeated growth measurements of 11 girls and 16 boys. The measurements from two adjacent children in the original data were combined and rearranged here to emulate a repeated measures sequence with 8 observations. The following statements create the data set:

```
title 'Analysis of Repeated Growth Measures';

data pr;
   input Person Gender $ y1 y2 y3 y4 y5 y6 y7 y8;
   array y{8};
   do time=5,7,8,4,3,2,1;
      Response = y{time};
      Age      = time+7;
      output;
   end;
   datalines;
 1   F   21.0  20.0  21.5  23.0  21.0  21.5  24.0  25.5
 2   F   20.5  24.0  24.5  26.0  23.5  24.5  25.0  26.5
 3   F   21.5  23.0  22.5  23.5  20.0  21.0  21.0  22.5
 4   F   21.5  22.5  23.0  25.0  23.0  23.0  23.5  24.0
 5   F   20.0  21.0  22.0  21.5  16.5  19.0  19.0  19.5
 6   F   24.5  25.0  28.0  28.0  26.0  25.0  29.0  31.0
 7   M   21.5  22.5  23.0  26.5  23.0  22.5  24.0  27.5
 8   M   25.5  27.5  26.5  27.0  20.0  23.5  22.5  26.0
 9   M   24.5  25.5  27.0  28.5  22.0  22.0  24.5  26.5
10   M   24.0  21.5  24.5  25.5  23.0  20.5  31.0  26.0
11   M   27.5  28.0  31.0  31.5  23.0  23.0  23.5  25.0
```

```
12   M   21.5  23.5  24.0  28.0  17.0  24.5  26.0  29.5
13   M   22.5  25.5  25.5  26.0  23.0  24.5  26.0  30.0
;
```

The following statements create a macro that sets colors for the covariance matrix, SETSTYLE1, create a macro that sets colors for the correlation matrix, SETSTYLE2, edit the templates, run the analysis with PROC GLIMMIX, and restore the default templates:

```
* You need to run the analysis once to know that 20 is a good maximum;
%paint(values=0 to 20 by 0.25,
       colors=cyan blue magenta red, macro=setstyle1)

%paint(values=0 to 1 by 0.05,
       colors=cyan blue magenta red, macro=setstyle2)

proc template;
   edit Stat.Glimmix.V;
      column Subject Index Row Col;
      edit Col;
         %setstyle1(backgroundcolor)
      end;
   end;
   edit Stat.Glimmix.VCorr;
      column Subject Index Row Col;
      edit Col;
         %setstyle2(backgroundcolor)
      end;
   end;
run;

ods html body='ar1.html' style=statistical;
ods listing close;
proc glimmix data=pr;
   class person gender time;
   model response = gender age gender*age;
   random _residual_ / sub=person type=arh(1) v residual vcorr;
   ods select v vcorr;
run;
ods listing;
ods html close;

proc template;
   delete Stat.Glimmix.V;
   delete Stat.Glimmix.VCorr;
run;
```

The results are displayed in Output 20.10.4 and Output 20.10.5. Both the covariance and correlation matrices have a structure that is more obvious when colors are added to the display. In particular, the colors clearly show the banded structure of the correlation matrix.

Output 20.10.4 Heteroscedastic AR(1) Covariance Matrix

Analysis of Repeated Growth Measures

The GLIMMIX Procedure

Estimated V Matrix for Person 1

Row	Col1	Col2	Col3	Col4	Col5	Col6	Col7
1	19.1973	10.5505	7.3707	4.5756	2.4983	1.4814	0.9697
2	10.5505	11.8104	8.2508	5.1220	2.7966	1.6582	1.0855
3	7.3707	8.2508	11.7407	7.2885	3.9795	2.3596	1.5446
4	4.5756	5.1220	7.2885	9.2159	5.0318	2.9836	1.9530
5	2.4983	2.7966	3.9795	5.0318	5.5959	3.3181	2.1720
6	1.4814	1.6582	2.3596	2.9836	3.3181	4.0075	2.6232
7	0.9697	1.0855	1.5446	1.9530	2.1720	2.6232	3.4976

Output 20.10.5 Heteroscedastic AR(1) Correlation Matrix

Estimated V Correlation Matrix for Person 1

Row	Col1	Col2	Col3	Col4	Col5	Col6	Col7
1	1.0000	0.7007	0.4910	0.3440	0.2410	0.1689	0.1183
2	0.7007	1.0000	0.7007	0.4910	0.3440	0.2410	0.1689
3	0.4910	0.7007	1.0000	0.7007	0.4910	0.3440	0.2410
4	0.3440	0.4910	0.7007	1.0000	0.7007	0.4910	0.3440
5	0.2410	0.3440	0.4910	0.7007	1.0000	0.7007	0.4910
6	0.1689	0.2410	0.3440	0.4910	0.7007	1.0000	0.7007
7	0.1183	0.1689	0.2410	0.3440	0.4910	0.7007	1.0000

Alternatively, you could just use the `Paint` macro to do the color interpolation, and use its output data set to create other types of style effects. The following statements show one way to set the font to bold and set the foreground color based on the values of the covariances:

```
%let inc = 0.25;

%paint(values=0 to 20 by &inc, colors=blue magenta red)

data cntlin;
   set colors;
   fmtname = 'paintfmt';
   label = _rgb_;
   end = start + &inc;
   keep start end label fmtname;
   run;

proc format cntlin=cntlin;
   run;
```

```
proc template;
   edit Stat.Glimmix.V;
      column Subject Index Row Col;
      edit Col;
         style = {foreground=paintfmt8. font_weight=bold};
      end;
   end;
run;

ods html body='ar1.html' style=statistical;
ods listing close;
proc glimmix data=pr;
   class person gender time;
   model response = gender age gender*age;
   random _residual_ / sub=person type=arh(1) v residual;
   ods select v;
run;
ods listing;
ods html close;

proc template;
   delete Stat.Glimmix.V;
run;
```

The **Paint** macro creates the SAS data set Colors with the result of the interpolation. This data set can be processed to create a format. The DATA step creates a range of values from Start to End and assigns a color to Label based on the color computed by the **Paint** macro. This data set is input to PROC FORMAT to create the format PAINTFMT. PROC TEMPLATE uses this format to set the color of the values in the table. The cell value is evaluated using the specified FOREGROUND= format for every cell in the table, and the appropriate color is assigned. PROC GLIMMIX does the analysis, and the results are displayed in Output 20.10.6.

Output 20.10.6 Heteroscedastic AR(1) Covariance Matrix

Analysis of Repeated Growth Measures

The GLIMMIX Procedure

Estimated V Matrix for Person 1

Row	Col1	Col2	Col3	Col4	Col5	Col6	Col7
1	19.1973	10.5505	7.3707	4.5756	2.4983	1.4814	0.9697
2	10.5505	11.8104	8.2508	5.1220	2.7966	1.6582	1.0855
3	7.3707	8.2508	11.7407	7.2885	3.9795	2.3596	1.5446
4	4.5756	5.1220	7.2885	9.2159	5.0318	2.9836	1.9530
5	2.4983	2.7966	3.9795	5.0318	5.5959	3.3181	2.1720
6	1.4814	1.6582	2.3596	2.9836	3.3181	4.0075	2.6232
7	0.9697	1.0855	1.5446	1.9530	2.1720	2.6232	3.4976

Many other effects could be achieved by using this approach and different options in the STYLE= specification.

References

Cole, J. W. L. and Grizzle, J. E. (1966), "Applications of Multivariate Analysis of Variance to Repeated Measures Experiments," *Biometrics*, 22, 810–828.

Hemmerle, W. J. and Hartley, H. O. (1973), "Computing Maximum Likelihood Estimates for the Mixed AOV Model Using the W-Transformation," *Technometrics*, 15, 819–831.

Olinger, C. R. and Tobias, R. D. (1998), "It Chops, It Dices, It Makes Julienne Slices! ODS for Data Analysis Output As-You-Like-It in Version 7," in *Proceedings of the Twenty-third Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Pothoff, R. F. and Roy, S. N. (1964), "A Generalized Multivariate Analysis of Variance Model Useful Especially for Growth Curve Problems," *Biometrika*, 51, 313–326.

Snedecor, G. W. and Cochran, W. G. (1967), *Statistical Methods*, Sixth Edition, Ames: Iowa State University Press.

Chapter 21

Statistical Graphics Using ODS

Contents

Introduction **499**
Getting Started with ODS Statistical Graphics **501**
Default Plots for Simple Linear Regression with PROC REG 501
Survival Estimate Plot with PROC LIFETEST 504
Contour and Surface Plots with PROC KDE 505
Contour Plots with PROC KRIGE2D 507
Partial Least Squares Plots with PROC PLS 510
Box-Cox Transformation Plot with PROC TRANSREG 512
LS-Means Diffogram with PROC GLIMMIX 513
Principal Component Analysis Plots with PROC PRINCOMP 515
Grouped Scatter Plot with PROC SGPLOT 517
A Primer on ODS Statistical Graphics **519**
Graph Styles 520
ODS Destinations 523
Accessing Individual Graphs 524
Specifying the Size and Resolution of Graphs 525
Modifying Your Graphs 525
Procedures That Support ODS Graphics 528
Procedures That Support ODS Graphics and Traditional Graphics 528
Syntax **529**
ODS GRAPHICS Statement 529
ODS Destination Statements 532
PLOTS= Option 533
Selecting and Viewing Graphs **535**
Specifying an ODS Destination for Graphics 535
Viewing Your Graphs in the SAS Windowing Environment 537
Determining Graph Names and Labels 538
Selecting and Excluding Graphs 540
Graphics Image Files **541**
Image File Types 541
Naming Graphics Image Files 542
Saving Graphics Image Files 544
Creating Graphs in Multiple Destinations 546
Graph Size and Resolution **547**

ODS Graphics Editor **548**
Enabling the Creation of Editable Graphs 549
Editing a Graph with the ODS Graphics Editor 550
The Default Template Libraries and the ODS PATH **553**
Styles **554**
An Overview of Styles 554
Style Elements and Attributes 555
Style Definitions and Colors 557
Some Common Style Elements 558
Creating an All-Color Style by Using the ModStyle Macro 564
Changing the Default Markers and Lines 567
Changing the Default Style 575
Graph Templates **577**
The Graph Template Language 577
Locating Templates 582
Displaying Templates 583
Editing Templates 585
Saving Customized Templates 587
Using Customized Templates 587
Reverting to the Default Templates 588
Statistical Graphics Procedures **589**
The SGPLOT Procedure 589
The SGSCATTER Procedure 591
The SGPANEL Procedure 593
The SGRENDER Procedure 596
Examples of ODS Statistical Graphics **601**
Example 21.1: Creating Graphs with Tool Tips in HTML 601
Example 21.2: Creating Graphs for a Presentation 602
Example 21.3: Creating Graphs in PostScript Files 604
Example 21.4: Displaying Graphs Using the DOCUMENT Procedure . . . 606
Example 21.5: Customizing Graphs Through Template Changes 610
Modifying Graph Titles and Axis Labels 610
Modifying Colors, Line Styles, and Markers 615
Modifying Tick Marks and Grid Lines 618
Modifying the Style to Show Grid Lines 619
Example 21.6: Customizing Survival Plots 622
Modifying the Plot Title 623
Modifying the Axes, Legend, and Inset Table 625
Modifying the Layout and Adding a New Inset Table 627
Example 21.7: Customizing Panels 633
Example 21.8: Customizing Axes and Reference Lines 637
Example 21.9: Customizing the Style for Box Plots 644
Example 21.10: Adding Text to Every Graph 647
Adding a Date and Project Stamp to a Few Graphs 648

Adding Data Set Information to a Graph 651
Adding a Date and Project Stamp to All Graphs 652
Example 21.11: PROC TEMPLATE Statement Order and Primary Plots . . 653

Introduction

Effective graphics are indispensable for modern statistical analysis. They reveal patterns, differences, and uncertainty that are not readily apparent in tabular output. Graphics provoke questions that stimulate deeper investigation, and they add visual clarity and rich content to reports and presentations.

In earlier SAS releases, creating graphs with statistical procedures typically required additional programming steps such as creating output data sets with the values to plot, modifying these data sets with a DATA step program, and using traditional SAS/GRAPH procedures to produce the plots.

SAS 9.2 eliminates the need for additional programming by providing new functionality, referred to as ODS Statistical Graphics (or ODS Graphics for short). ODS Graphics is an extension of ODS (the Output Delivery System). ODS manages procedure output and lets you display it in a variety of destinations, such as HTML and RTF. With ODS Graphics, statistical procedures now produce graphs as automatically as they produce tables, and graphs are now integrated with tables in the ODS output. ODS Graphics is available in procedures in SAS/STAT, Base SAS, SAS/ETS, SAS/QC, SAS/GRAPH, and other products (see the section "Procedures That Support ODS Graphics" on page 528). Note that SAS/GRAPH software is required for ODS Graphics functionality.

ODS Graphics is enabled when you specify the following statement:

```
ods graphics on;
```

When ODS Graphics is enabled, procedures that support ODS Graphics create appropriate graphs, either by default or when you specify procedure options for requesting specific graphs. These options are documented in the "Syntax" section of each procedure chapter, and the "Details" section of each chapter provides an "ODS Graphics" subsection that lists the graphs that are available. Once ODS Graphics is enabled, it stays enabled for the duration of your SAS session. Alternatively, you can turn it off as follows:

```
ods graphics off;
```

For example, you might consider disabling ODS Graphics if your goal is solely to produce computational results. Often though, you can specify ODS GRAPHICS ON and then leave it on. Throughout this chapter, ODS Graphics is enabled only once per section.

This chapter provides a basic introduction to ODS Graphics along with more detailed information. The following list provides a guide to reading this chapter:

- If you want to see a few of the many graphs that are produced by statistical procedures by using ODS Graphics, see the section "Getting Started with ODS Statistical Graphics" on page 501.
- If you are using ODS Graphics for the first time, read the section "A Primer on ODS Statistical Graphics" on page 519, which provides the minimum information that you need to get started.
- If you need to create plots of raw data or your own customized plots of statistical results, see the section "Statistical Graphics Procedures" on page 589, which describes new SAS/GRAPH procedures that use ODS Graphics.
- If you need information about specialized topics such as accessing your graphs, making changes to your graphs, and working with ODS styles, see the detailed discussions starting with the section "Syntax" on page 529 and including the section "Examples of ODS Statistical Graphics" on page 601.

If you are unfamiliar with ODS, see Chapter 20, "Using the Output Delivery System." For complete documentation about the Output Delivery System, see the *SAS Output Delivery System: User's Guide*. For complete documentation about ODS graph templates, see the *SAS/GRAPH: Graph Template Language User's Guide* and the *SAS/GRAPH: Graph Template Language Reference*. For complete documentation about the Graphics Editor, see the *Getting Started with the SAS/GRAPH Statistical Graphics Editor*. Also see the *SAS/GRAPH: Statistical Graphics Procedures Guide* for information about the statistical graphics procedures.

Getting Started with ODS Statistical Graphics

This section provides examples that illustrate the most basic uses of ODS Graphics with a few of the many plots that are produced by statistical procedures. These examples also illustrate several ODS styles that are useful for statistical analysis.

Default Plots for Simple Linear Regression with PROC REG

This example is taken from the section "Getting Started: REG Procedure" on page 5436 of Chapter 73, "The REG Procedure." It uses the following data from a study of 19 children:

```
data Class;
   input Name $ Height Weight Age @@;
   datalines;
Alfred  69.0 112.5 14  Alice  56.5  84.0 13  Barbara 65.3  98.0 13

... more lines ...

;
```

A larger version of this data set is available in the SASHELP library, and later examples use this data set by specifying **data=sashelp.class**.

The following statements use PROC REG to fit a simple linear regression model in which Weight is the response variable and Height is the independent variable:

```
ods graphics on;

proc reg data=Class;
   model Weight = Height;
run; quit;
```

The ODS GRAPHICS ON statement is specified to request ODS Graphics in addition to the usual tabular output. The statement ODS GRAPHICS OFF is not used here, but it can be specified to disable ODS Graphics.

The graphical output consists of a fit diagnostics panel, a residual plot, and a fit plot. These plots are integrated with the tabular output and are shown in Figure 21.1, Figure 21.2, and Figure 21.3, respectively. The results are displayed in the STATISTICAL style.

Figure 21.1 Fit Diagnostics Panel

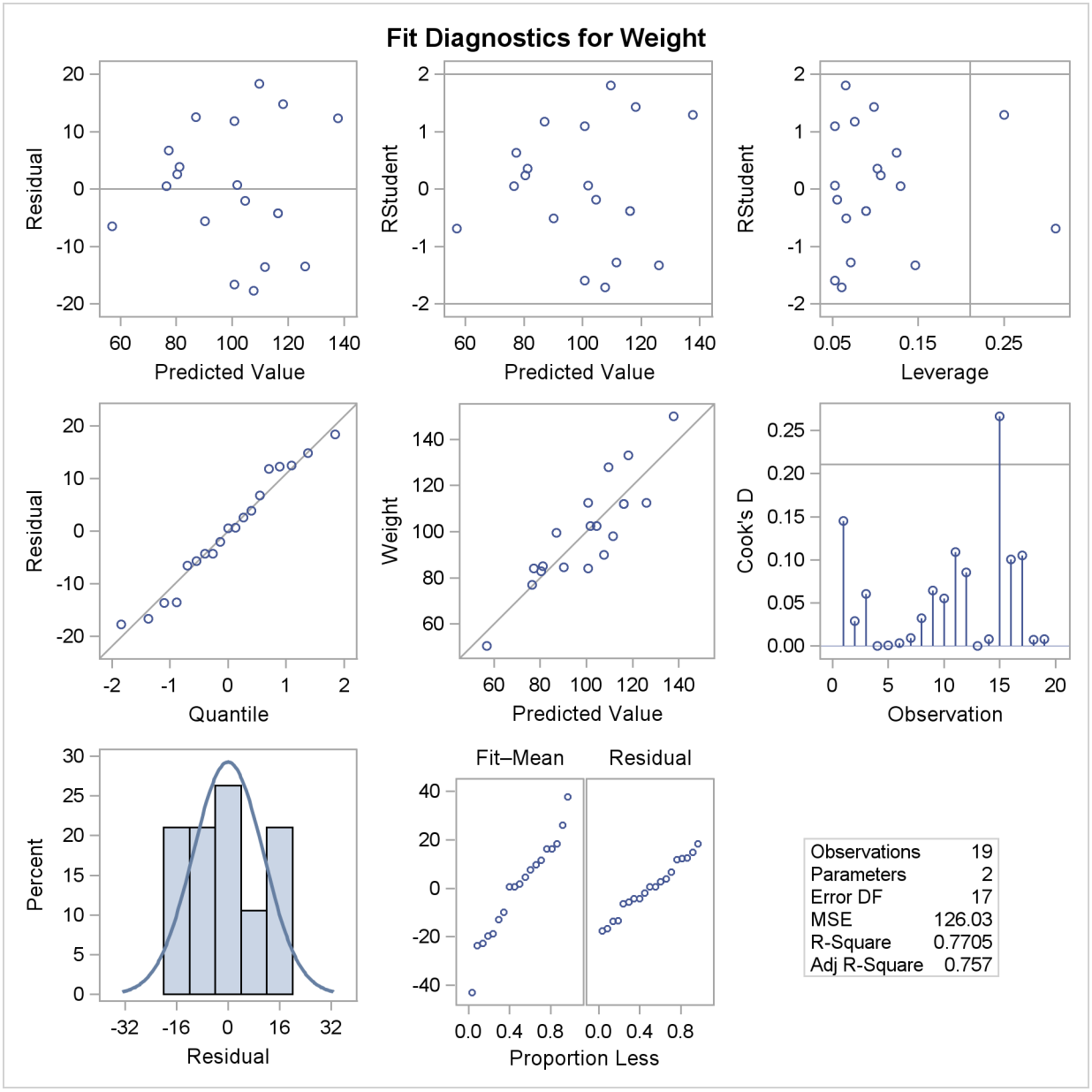


Figure 21.2 Residual Plot

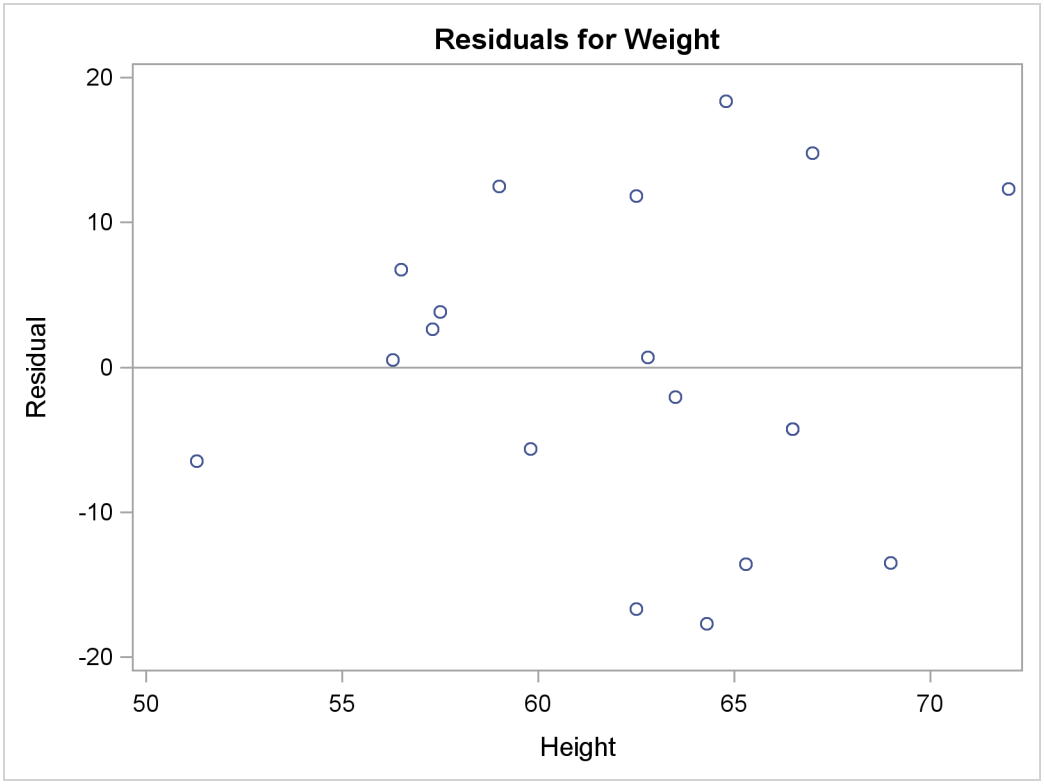
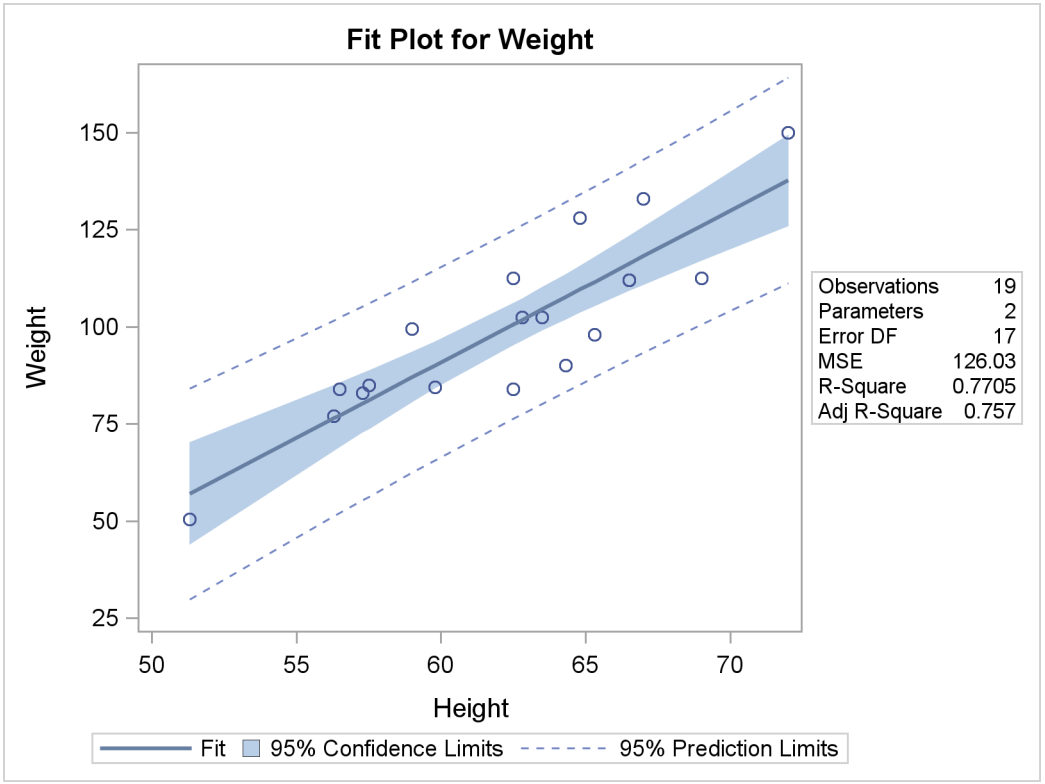


Figure 21.3 Fit Plot



ODS styles control the colors and general appearance of all graphs and tables, and SAS provides several styles that are recommended for use with statistical graphics. The STATISTICAL style is the default style in SAS/STAT documentation and is used unless another style is explicitly specified. The default style that you see when you run SAS depends on the ODS destination. Specifically, the default style for the LISTING destination is LISTING, the default style for the HTML destination is DEFAULT, and the default style for the RTF destination is RTF. These and other styles are shown in this chapter. For more information about styles, see the section "Graph Styles" on page 520 and the section "Styles" on page 554.

Survival Estimate Plot with PROC LIFETEST

This example is taken from Example 49.2 of Chapter 49, "The LIFETEST Procedure." It shows how to construct a product-limit survival estimate plot. Both the ODS GRAPHICS statement and procedure options are used to request the plot.

The following statements create a SAS data set with disease-free times for three risk categories:

```
proc format;
   value risk 1='ALL' 2='AML-Low Risk' 3='AML-High Risk';
run;

data BMT;
   input Group T Status @@;
   format Group risk.;
   label T='Disease Free Time';
   datalines;
1 2081 0 1 1602 0 1 1496 0 1 1462 0 1 1433 0

... more lines ...

;
```

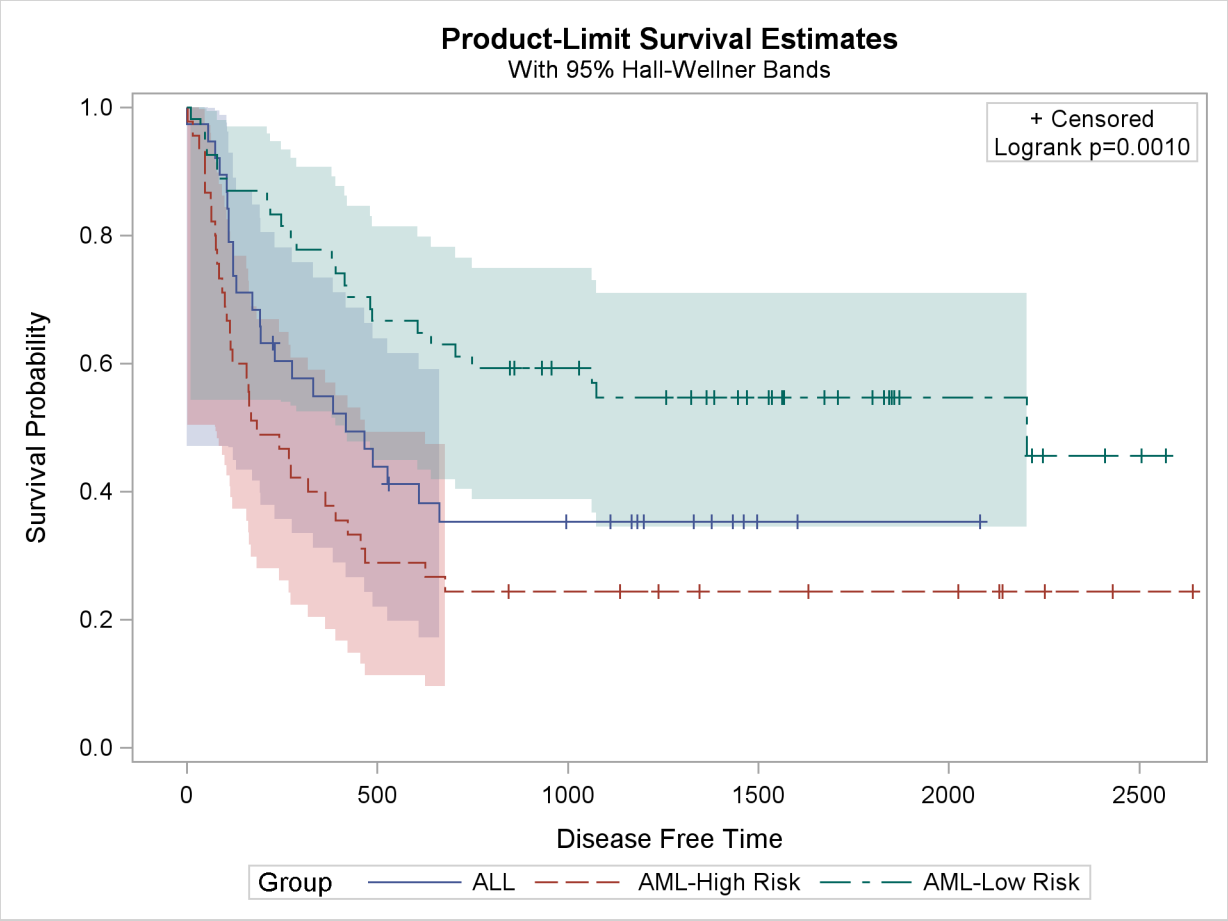
The following statements use PROC LIFETEST to compute the product-limit estimate of the survivor function for each risk category:

```
ods graphics on;

proc lifetest data=BMT plots=survival(cb=hw test);
   time T * Status(0);
   strata Group / test=logrank;
run;
```

The ODS GRAPHICS ON statement requests ODS Graphics functionality, and the PLOTS=SURVIVAL option requests a plot of the estimated survival curves. The CB=HW suboption requests Hall-Wellner confidence bands, and the TEST suboption displays the *p*-value for the log-rank test in a plot inset.

Figure 21.4 displays the plot; note that tabular output is not shown. The results are displayed in the STATISTICAL style. Patients in the AML-Low Risk group are disease-free longer than those in the ALL group, who in turn fare better than those in the AML-High Risk group.

Figure 21.4 Survival Plot



Contour and Surface Plots with PROC KDE

This example is taken from the section "Getting Started: KDE Procedure" on page 2886 in Chapter 45, "The KDE Procedure." Here, in addition to the ODS GRAPHICS statement, procedure options are used to request plots. The following statements simulate 1,000 observations from a bivariate normal density with means (0,0), variances (10,10), and covariance 9:

```
data bivnormal;
   do i = 1 to 1000;
      z1 = rannor(104);
      z2 = rannor(104);
      z3 = rannor(104);
      x  = 3*z1+z2;
      y  = 3*z1+z3;
      output;
   end;
run;
```

The following statements request a bivariate kernel density estimate for the variables x and y:

```
ods graphics on;

proc kde data=bivnormal;
   bivar x y / plots=contour surface;
run;
```

The PLOTS= option requests a contour plot and a surface plot of the estimate (displayed in Figure 21.5 and Figure 21.6, respectively). The results are displayed in the STATISTICAL style. For more information about the graphics available in PROC KDE, see the section "ODS Graphics" on page 2904 of Chapter 45, "The KDE Procedure."

Figure 21.5 Contour Plot of Estimated Density

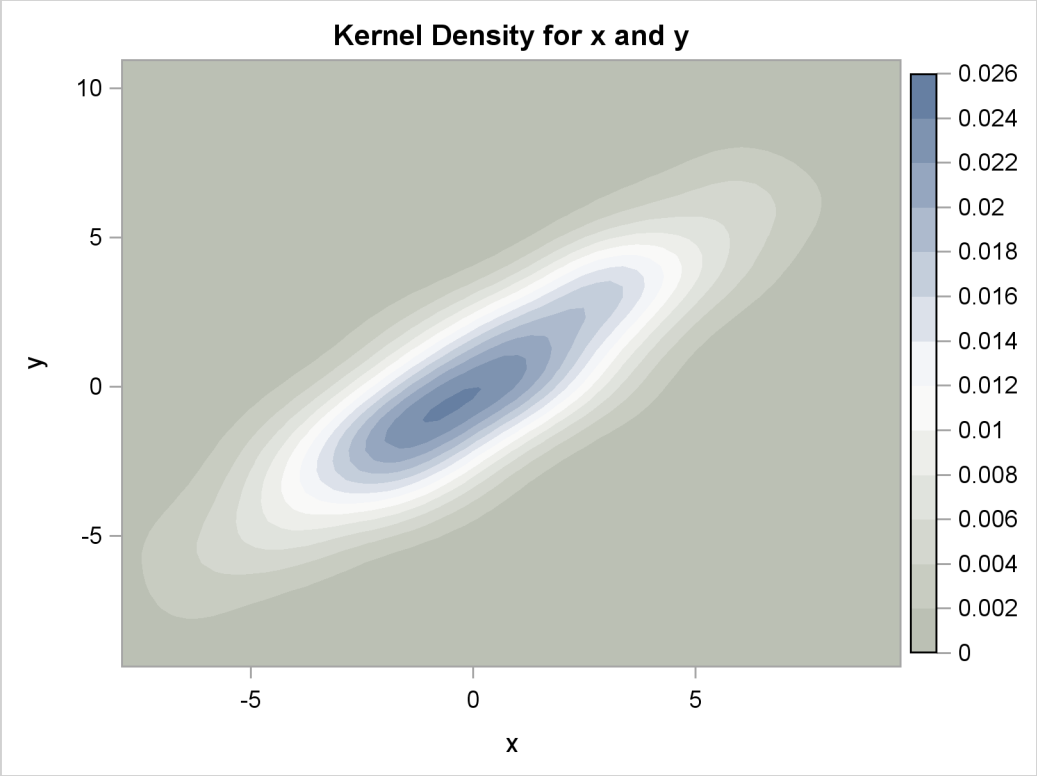
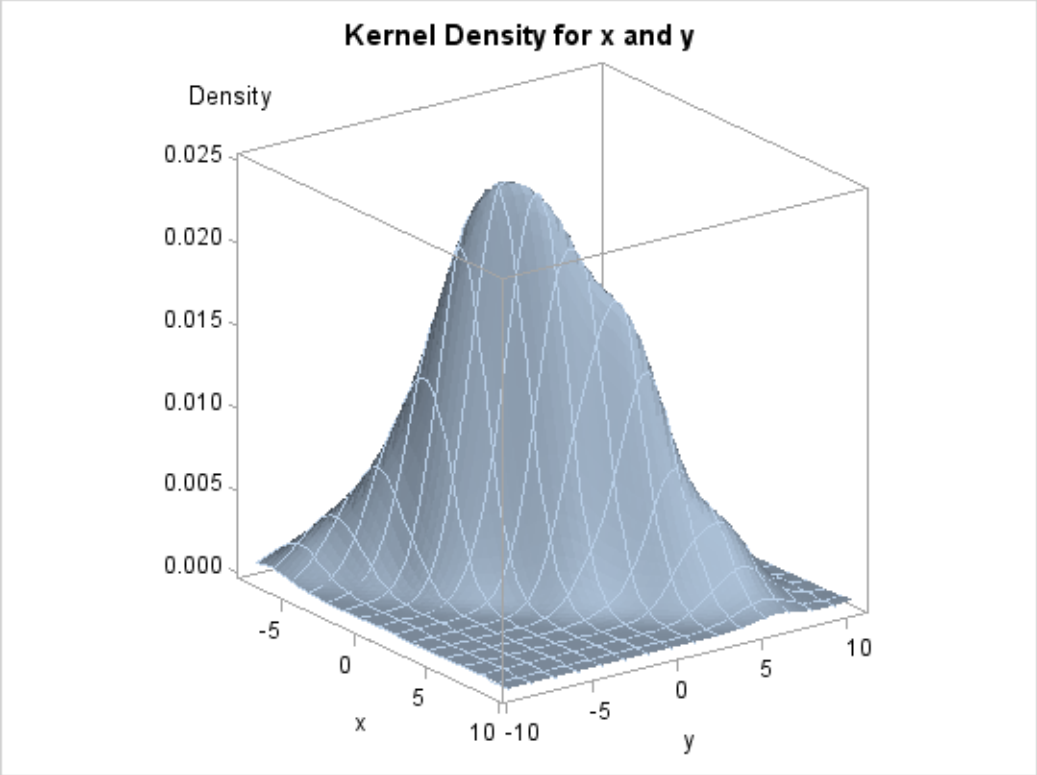


Figure 21.6 Surface Plot of Estimated Density



Contour Plots with PROC KRIGE2D

This example is taken from Example 46.2 of Chapter 46, "The KRIGE2D Procedure." The following statements create a SAS data set that contains measurements of coal seam thickness:

```
data thick;
   input East North Thick @@;
   label Thick='Coal Seam Thickness';
   datalines;
 0.7  59.6  34.1   2.1  82.7  42.2   4.7  75.1  39.5

... more lines ...

;
```

The following statements set the output style to DEFAULT and run PROC KRIGE2D:

```
ods listing style=default;
ods graphics on;

proc krige2d data=thick outest=predictions
             plots=(observ(showmissing)
                    pred(fill=pred line=pred obs=linegrad)
                    pred(fill=se line=se obs=linegrad));
   coordinates xc=East yc=North;
   predict var=Thick r=60;
   model scale=7.2881 range=30.6239 form=gauss;
   grid x=0 to 100 by 2.5 y=0 to 100 by 2.5;
run;
```

The PLOTS=OBSERV(SHOWMISSING) option produces a scatter plot of the data along with the locations of any missing data. The PLOTS=PRED option produces maps of the kriging predictions and standard errors. Two instances of the PLOTS=PRED option are specified with suboptions that customize the plots. The results are shown in Figure 21.7.

Figure 21.7 PROC KRIGE2D Results Using the DEFAULT Style

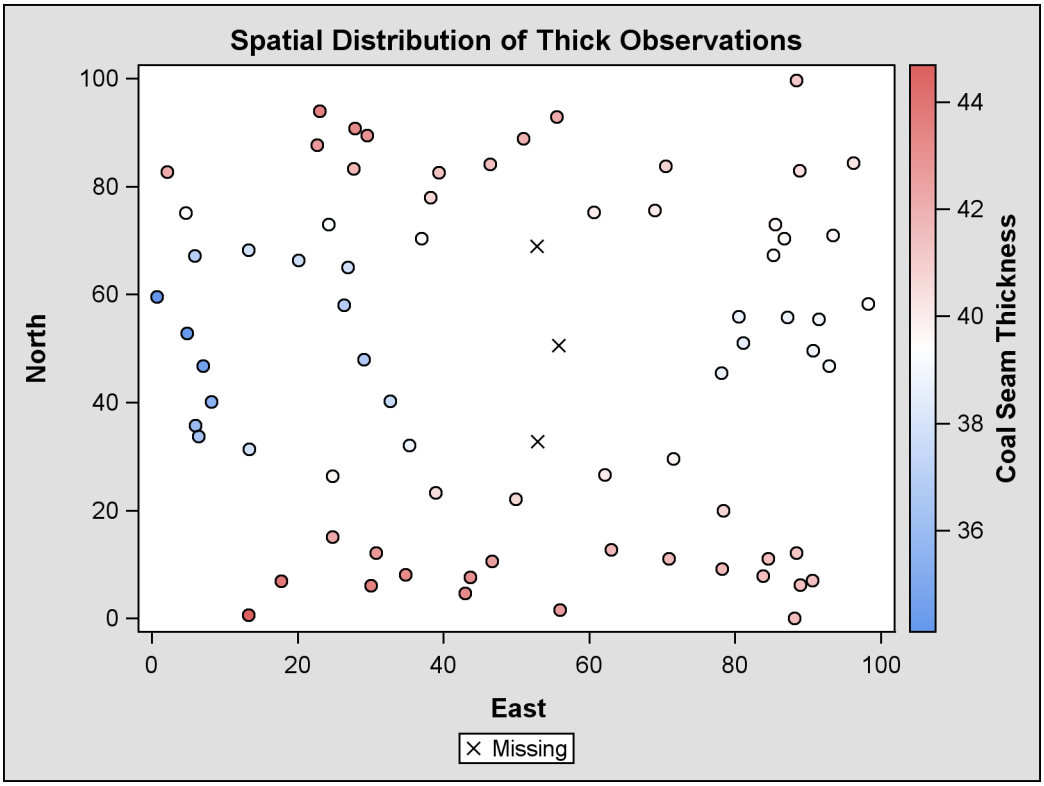
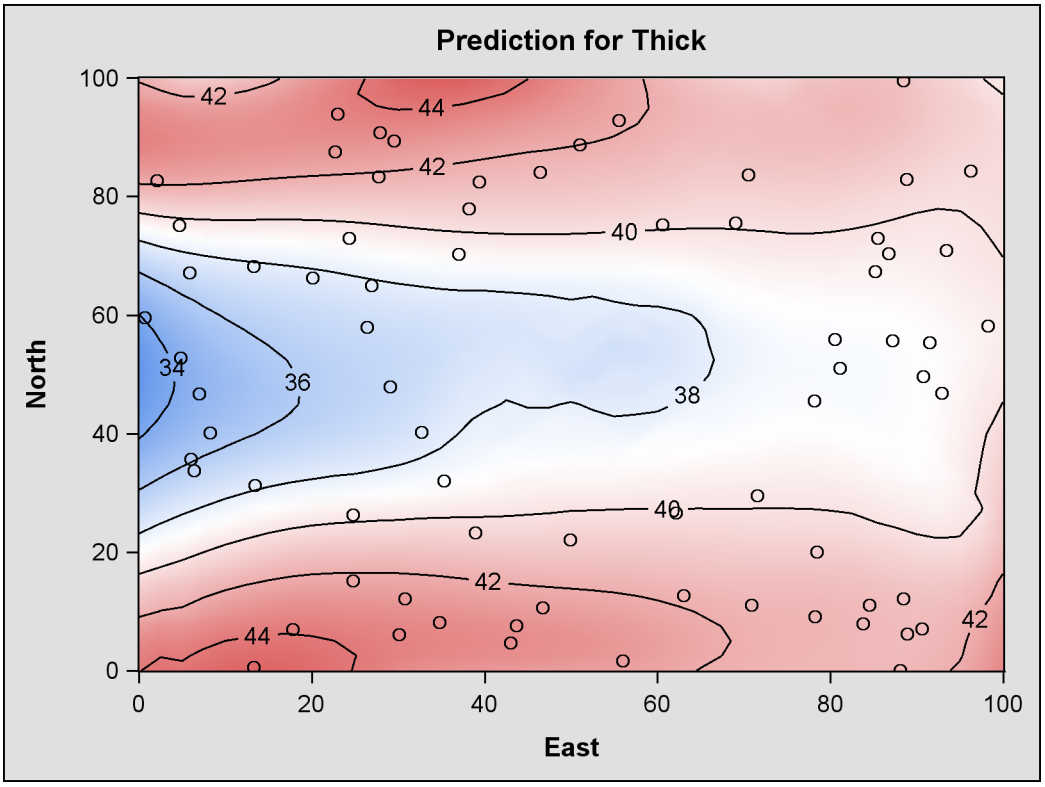
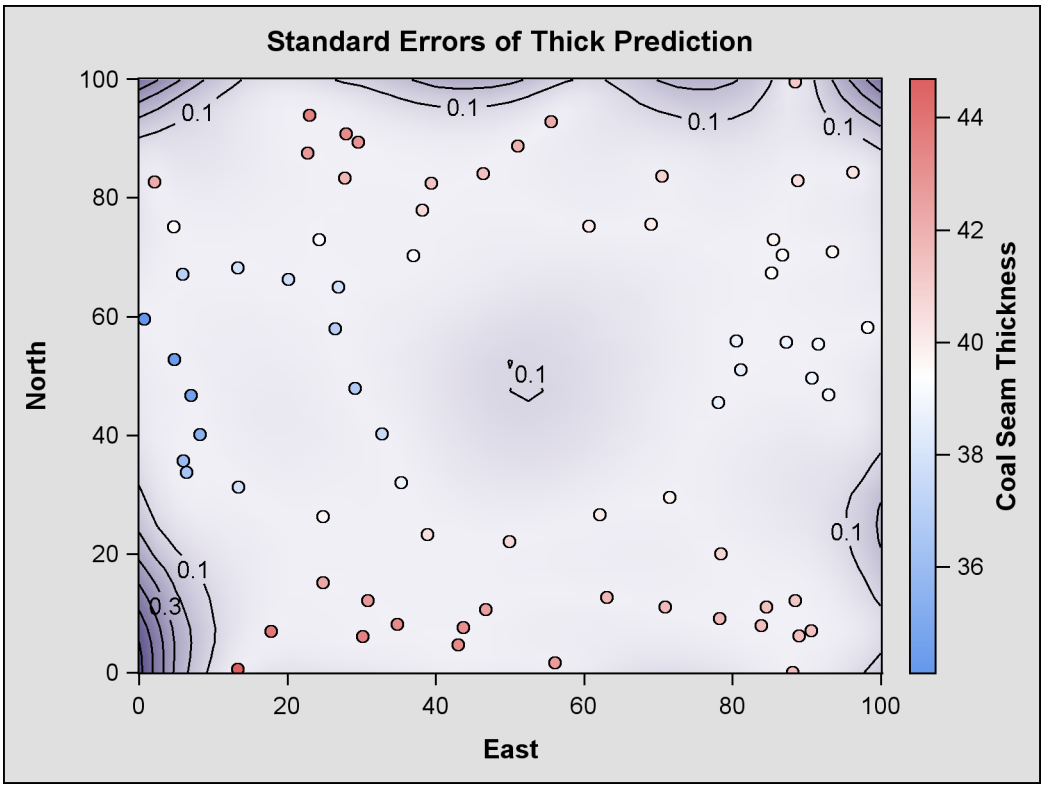


Figure 21.7 *continued*





Partial Least Squares Plots with PROC PLS

This example is taken from the section "Getting Started: PLS Procedure" on page 4767 of Chapter 66, "The PLS Procedure." The following statements create a SAS data set that contains measurements of biological activity in the Baltic Sea:

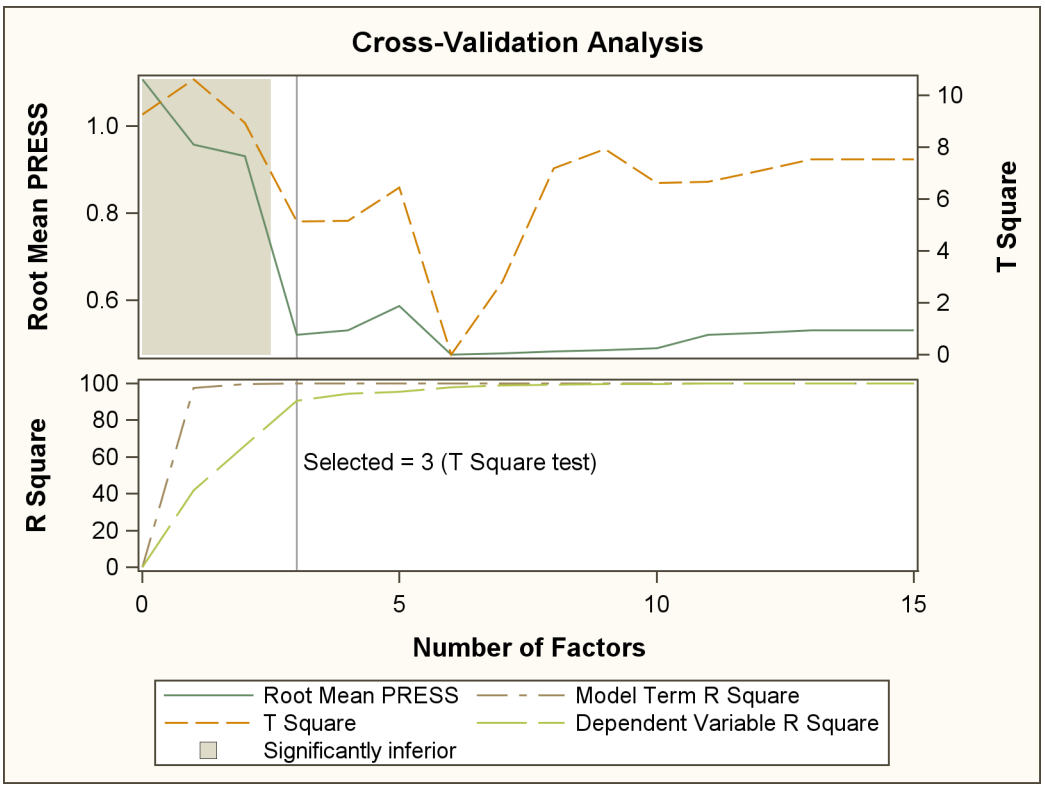
```
data Sample;
   input obsnam $ v1-v27 ls ha dt @@;
   datalines;
EM1   2766 2610 3306 3630 3600 3438 3213 3051 2907 2844 2796
      2787 2760 2754 2670 2520 2310 2100 1917 1755 1602 1467
      1353 1260 1167 1101 1017         3.0110  0.0000   0.00

... more lines ...

;
```

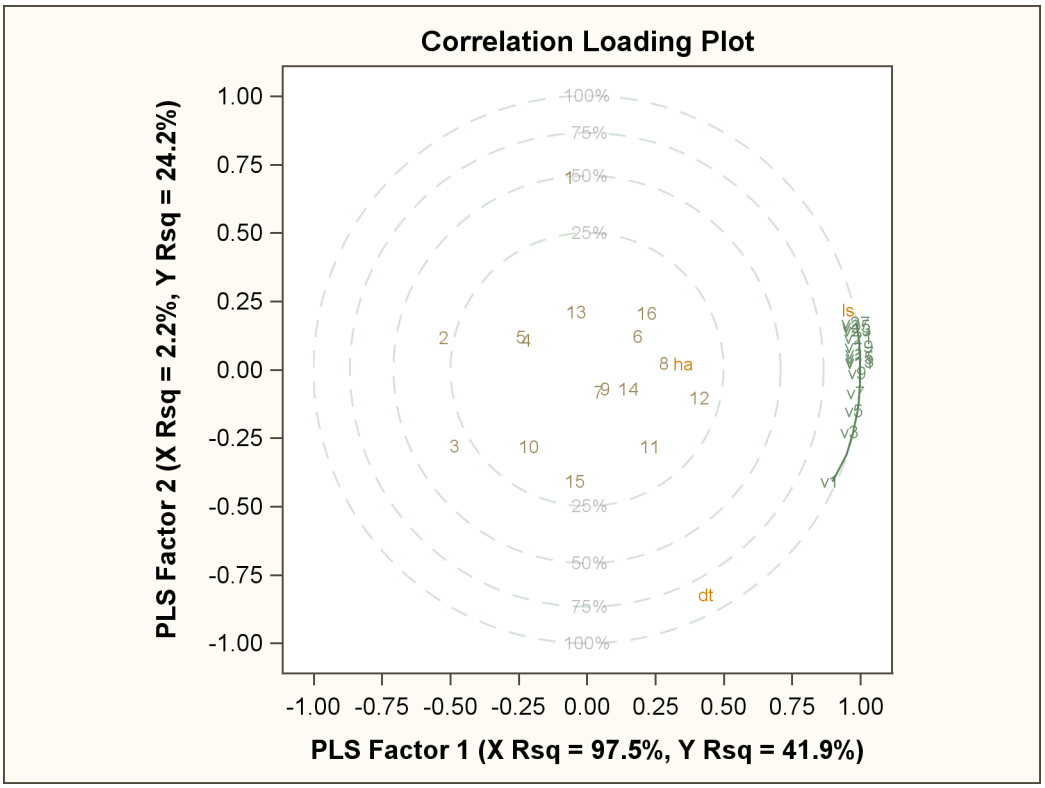
The following statements set the output style to ANALYSIS and run PROC PLS:

```
ods listing style=analysis;
ods graphics on;

proc pls data=sample cv=split cvtest(seed=104);
   model ls ha dt = v1-v27;
run;
```

By default, the procedure produces a plot for the cross validation analysis and a correlation loading plot (see Figure 21.8).

Figure 21.8 Partial Least Squares Results Using the ANALYSIS Style





Box-Cox Transformation Plot with PROC TRANSREG

This example is taken from Example 90.2 of Chapter 90, "The TRANSREG Procedure." The following statements create a SAS data set that contains failure times for yarn:

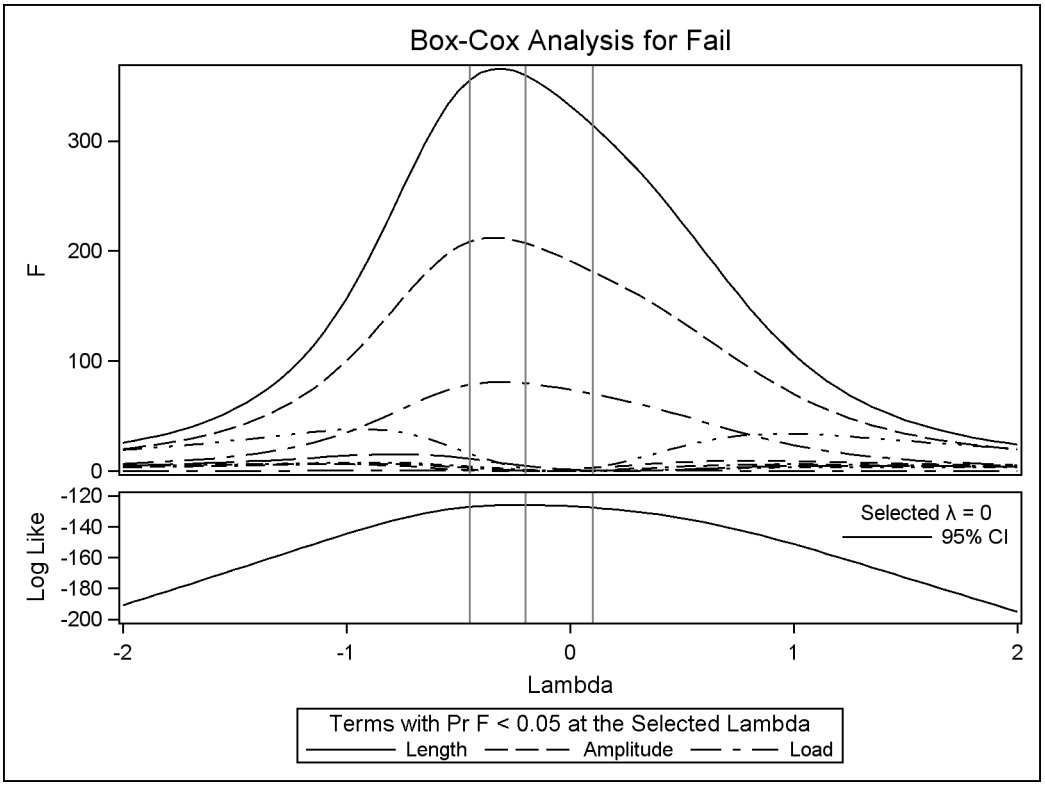
```
proc format;
   value a -1 =   8 0 =   9 1 =  10;
   value l -1 = 250 0 = 300 1 = 350;
   value o -1 =  40 0 =  45 1 =  50;
run;

data yarn;
   input Fail Amplitude Length Load @@;
   format amplitude a. length l. load o.;
   label fail = 'Time in Cycles until Failure';
   datalines;
 674 -1 -1 -1    370 -1 -1  0    292 -1 -1  1    338  0 -1 -1

... more lines ...

;
```

The following statements set the output style to JOURNAL2 and run PROC TRANSREG:

```
ods listing style=journal2;
ods graphics on;

proc transreg data=yarn;
   model BoxCox(fail / convenient lambda=-2 to 2 by 0.05) =
         qpoint(length amplitude load);
run;
```

The log-likelihood plot in Figure 21.9 suggests a Box-Cox transformation with $\lambda = 0$.

Figure 21.9 Box-Cox "Significant Effects" Using the JOURNAL2 Style



LS-Means Diffogram with PROC GLIMMIX

This example is taken from the section "Graphics for LS-Mean Comparisons" on page 2288 of Chapter 38, "The GLIMMIX Procedure." The following statements create a SAS data set that contains measurements from an experiment that investigates how snapdragons grow in various soils:

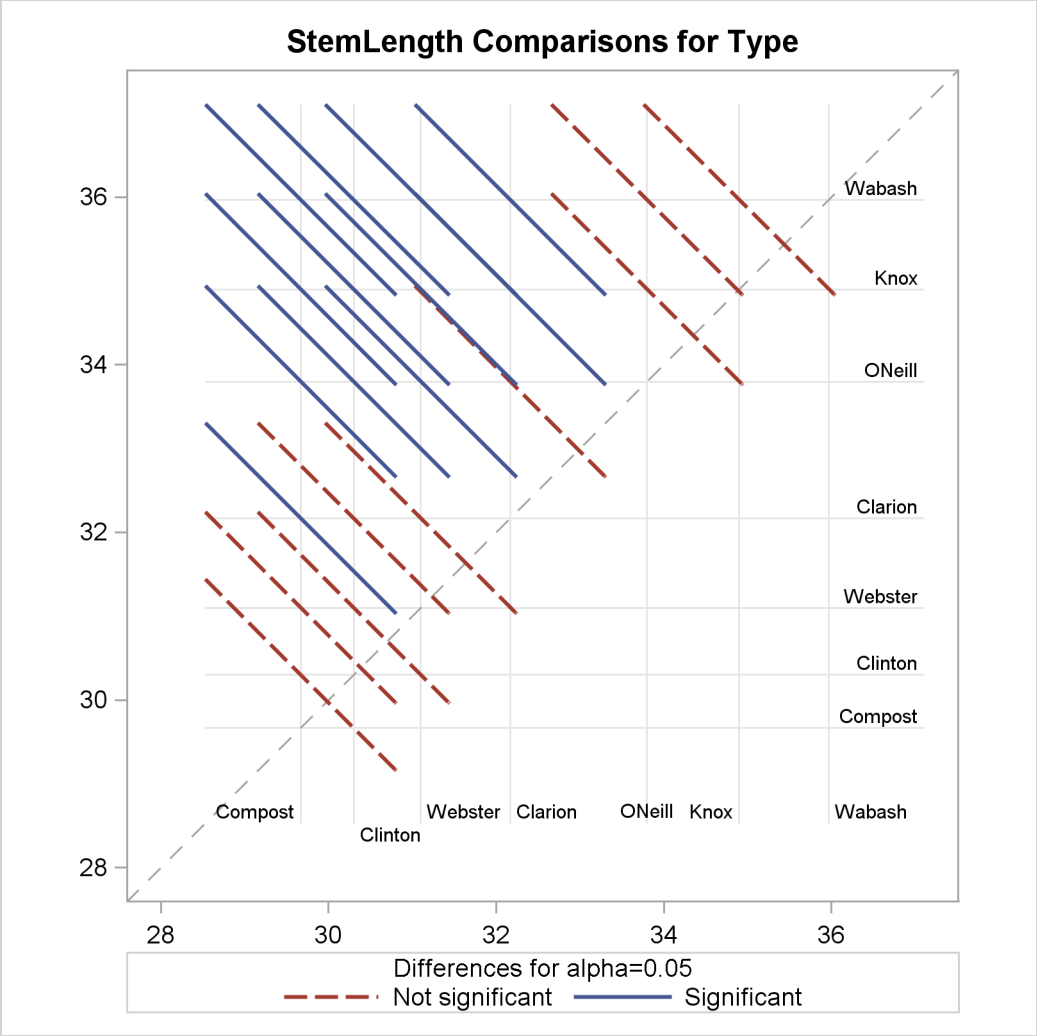
```
data plants;
   input Type $ @;
   do Block = 1 to 3;
      input StemLength @;
      output;
   end;
   datalines;
Clarion   32.7 32.3 31.5

... more lines ...

;
```

The following statements set the output style to STATISTICAL and run PROC GLIMMIX:

```
ods listing style=statistical;
ods graphics on;

proc glimmix data=plants order=data plots=diffogram;
   class Block Type;
   model StemLength = Block Type;
   lsmeans Type;
run;
```

The PLOTS=DIFFOGRAM option produces a diffogram, shown in Figure 21.10, that displays all of the pairwise least squares mean differences and indicates which are significant.

Figure 21.10 LS-Means Diffogram Using the STATISTICAL Style



Principal Component Analysis Plots with PROC PRINCOMP

This example is taken from Example 69.3 of Chapter 69, "The PRINCOMP Procedure." The following statements create a SAS data set that contains ratings of job performance of police officers:

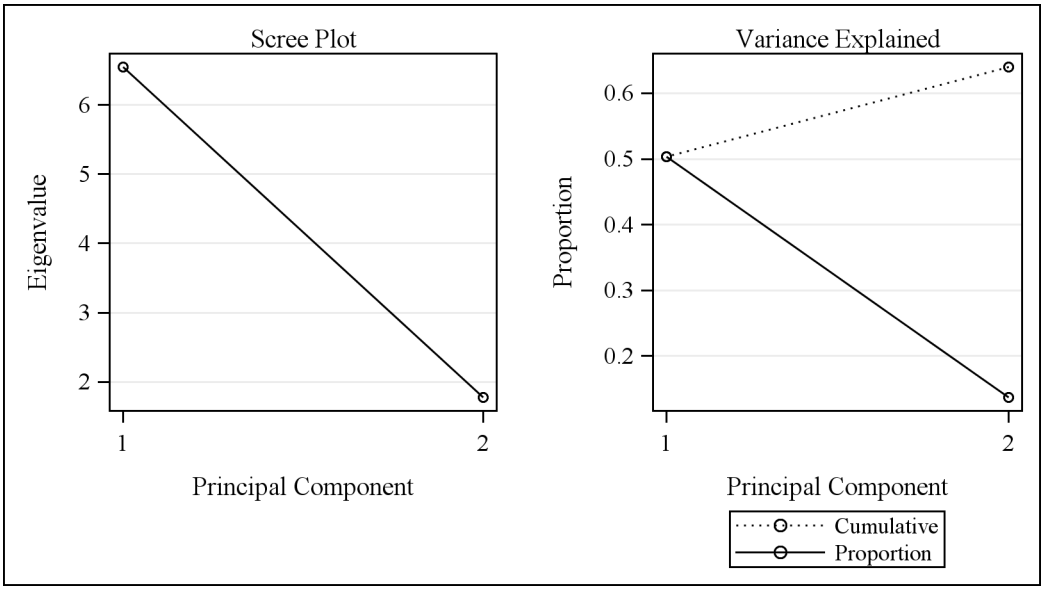
```
options validvarname=any;

data Jobratings;
   input ('Communication Skills'n
          'Problem Solving'n
          'Learning Ability'n
          'Judgment Under Pressure'n
          'Observational Skills'n
          'Willingness to Confront Problems'n
          'Interest in People'n
          'Interpersonal Sensitivity'n
          'Desire for Self-Improvement'n
          'Appearance'n
          'Dependability'n
          'Physical Ability'n
          'Integrity'n
          'Overall Rating'n) (1.);
   datalines;
26838853879867

... more lines ...

;
```

The following statements set the output style to RTF and run PROC PRINCOMP:

```
ods listing style=rtf;
ods graphics on;

proc princomp data=Jobratings(drop='Overall Rating'n) n=2
              plots=(Matrix PatternProfile);
run;
```

The plots are requested by the PLOTS=(MATRIX PATTERNPROFILE) option. The results, shown in Figure 21.11, contain the default scree and variance-explained plots, along with a scatter plot matrix of component scores and a pattern profile plot.

Figure 21.11 Principal Components Using the RTF Style



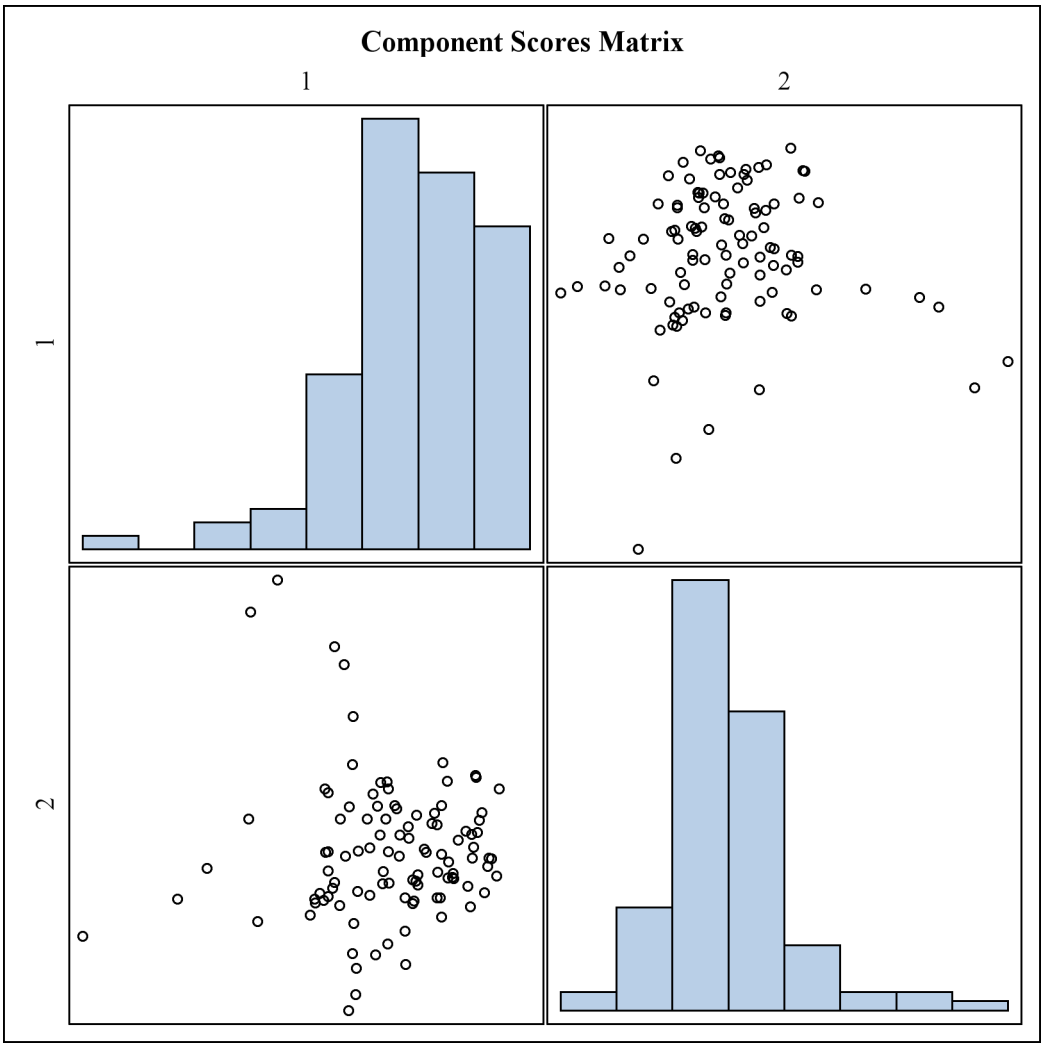
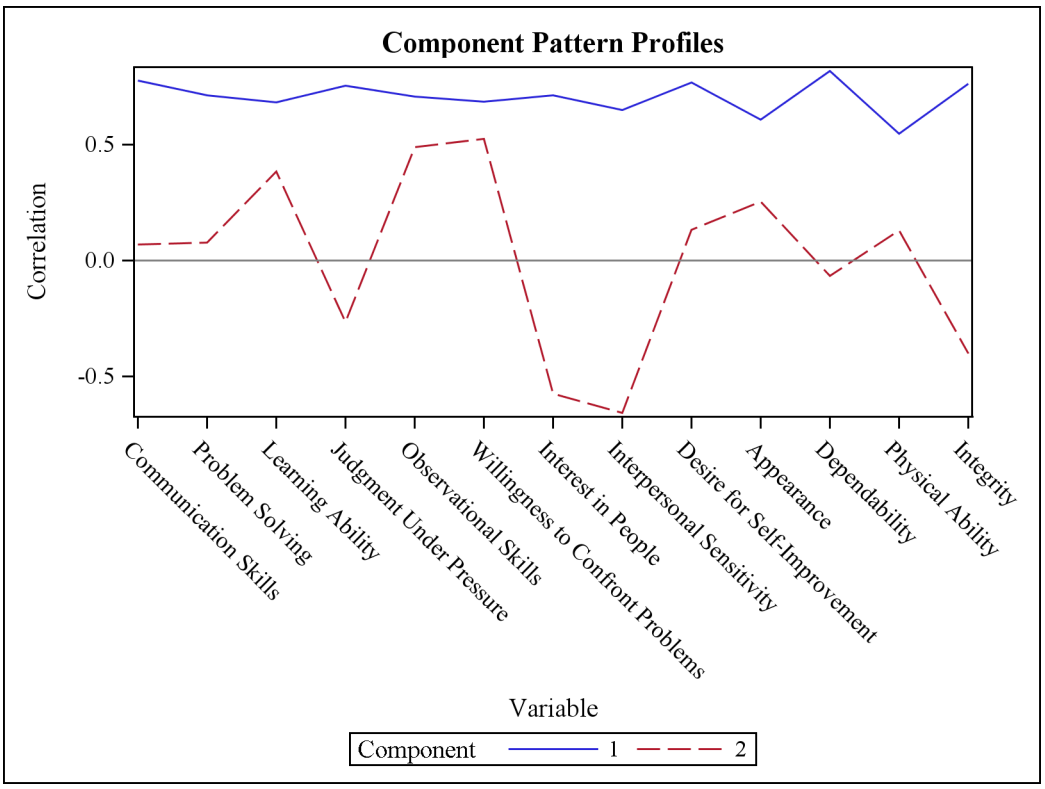


Figure 21.11 *continued*



Grouped Scatter Plot with PROC SGPLOT

This example is taken from Example 31.1 of Chapter 31, "The DISCRIM Procedure." The following statements create a SAS data set from the iris data:

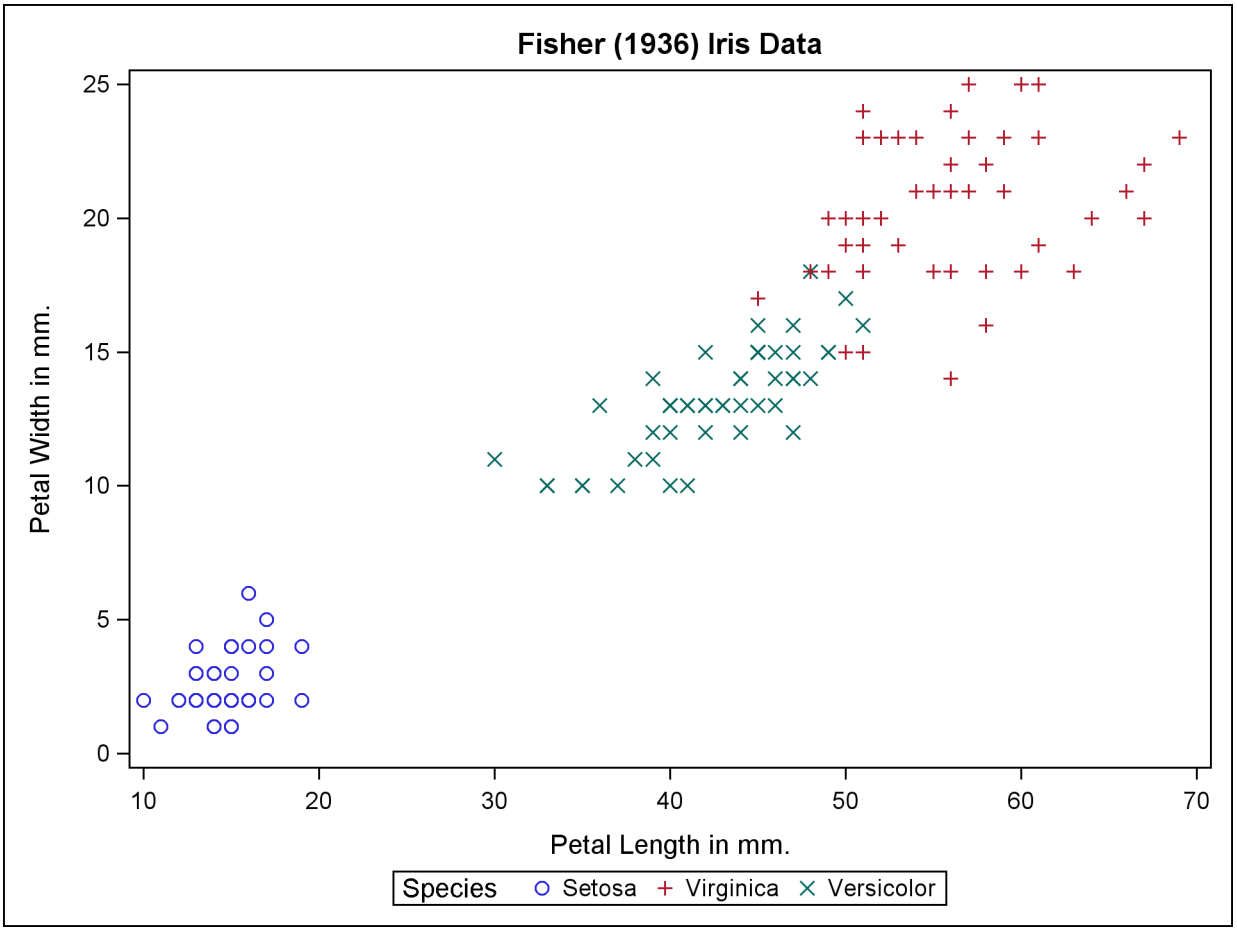
```
proc format;
   value specname
      1='Setosa    '
      2='Versicolor'
      3='Virginica ';
run;

data iris;
   input SepalLength SepalWidth PetalLength PetalWidth
         Species @@;
   format Species specname.;
   label SepalLength='Sepal Length in mm.'
         SepalWidth ='Sepal Width in mm.'
         PetalLength='Petal Length in mm.'
         PetalWidth ='Petal Width in mm.';
   datalines;
50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3

... more lines ...

;
```

The following statements set the output style to LISTING and run PROC SGPLOT to make a scatter plot, grouped by iris species:

```
ods listing style=listing;

proc sgplot data=iris;
   title 'Fisher (1936) Iris Data';
   scatter x=petallength y=petalwidth / group=species;
run;
```

The results are shown in Figure 21.12.

Figure 21.12 Scatter Plot That Uses the LISTING Style



See the section "Statistical Graphics Procedures" on page 589 and the *SAS/GRAPH: Statistical Graphics Procedures Guide* for more information about PROC SGPLOT (statistical graphics plot) and other SG procedures. You do not need to enable ODS Graphics in order to use SG procedures (because making plots with ODS Graphics is their sole function).

A Primer on ODS Statistical Graphics

You invoke ODS Graphics by specifying the following statement:

```
ods graphics on;
```

ODS Graphics remains in effect for all procedure steps until you turn it off with the following statement:

```
ods graphics off;
```

See the section "Syntax" on page 529 for details about the more commonly used ODS GRAPHICS statement options. Once you have invoked ODS Graphics, creating graphical output with procedures is as simple as creating tabular output. You can use control your output in the following ways:

- ODS destination statements (such as ODS HTML or ODS RTF) specify where you want your graphs displayed. See Figure 21.13 for an example of HTML output. See the section "ODS Destination Statements" on page 532 for a list of the supported destinations. See the section "Syntax" on page 529 for details about the more commonly used ODS destination statement options.
- ODS SELECT and ODS EXCLUDE statements select and exclude graphs from your output. See the section "Selecting and Excluding Graphs" on page 540 for an example of how to select graphs.
- ODS OUTPUT statements create SAS data sets from the data object used to make the plot. See the section "Specifying an ODS Destination for Graphics" on page 535 for an example.
- Procedure options specify which graphs to create. For each procedure, these options are described in the "Syntax" section of the procedure chapter. Typically, you use the PLOTS= option to control all graphs. The available graphs are listed in the "ODS Graphics" section, which is found in the "Details" section of each procedure chapter. Many graphs are produced by default.
- ODS styles control the general appearance and consistency of all graphs and tables. See the sections "Graph Styles" on page 520 and "Styles" on page 554 for more information about styles.
- ODS templates modify the layout and details of each graph. See the section "Graph Templates" on page 577 for more information about templates.

  NOTE: A default template is provided by SAS for each graph, so you do not need to know anything about templates to create statistical graphics.

You can also access individual graphs, control the resolution and size of graphs, and modify your graphs (as explained in the sections beginning with "Selecting and Viewing Graphs" on page 535). Alternatively, you can use special statistical graphics procedures to create custom graphs directly (see the section "Statistical Graphics Procedures" on page 589).

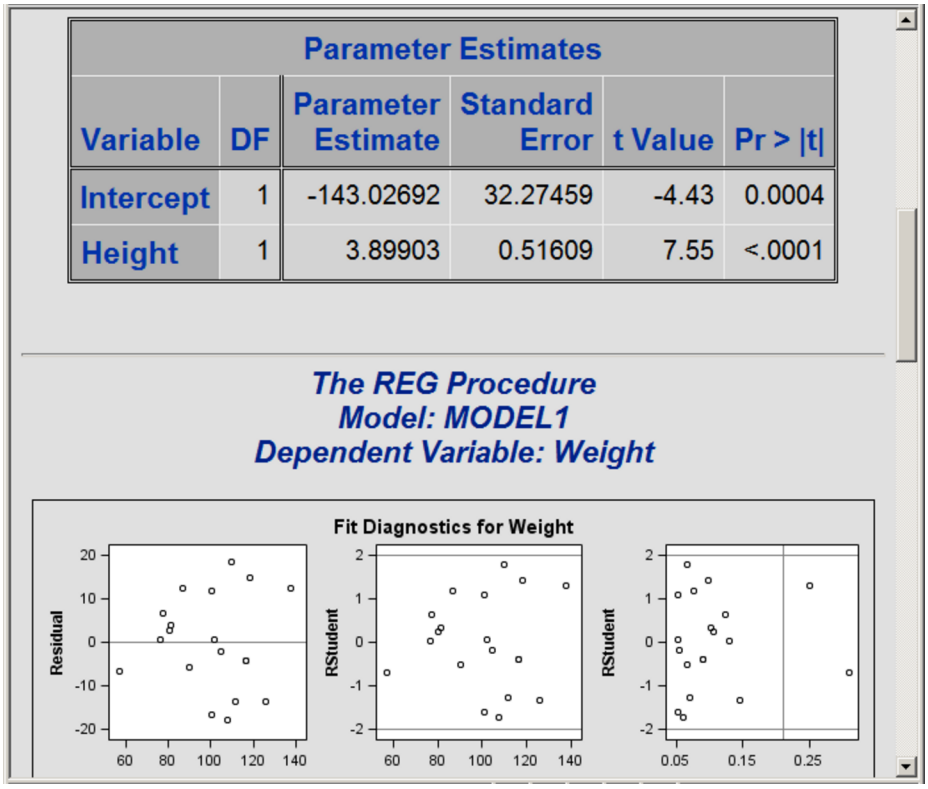
Graph Styles

ODS styles control the overall appearance of graphs and tables. They specify colors, fonts, line styles, and other attributes of graph elements. The following styles are recommended for statistical work:

- The DEFAULT style is a color style intended for general-purpose work. See Figure 21.13 for an example of the DEFAULT style, which is the default style for the HTML destination. Most other styles inherit some of their elements from this style.
- The STATISTICAL style is a color style recommended for output in Web pages or color print media. The STATISTICAL style might not necessarily print well on black-and-white devices. See Figure 21.14 for an example. This is the default style for SAS/STAT documentation.
- The ANALYSIS style is a color style with a somewhat different appearance from the STATISTICAL style. See Figure 21.8 for an example.
- The JOURNAL family of styles (JOURNAL and JOURNAL2) consists of black-and-white or gray-scale styles that are recommended for graphs that appear in journals and in other black-and-white publications. See Figure 21.15 for an example of the JOURNAL style, see Figure 21.9 for an example of the JOURNAL2 style, and see Example 21.3 for a comparison of the two styles.
- The RTF style is used to produce graphs to insert into a Microsoft Word document or a Microsoft PowerPoint slide. See Figure 21.11 for an example of the RTF style, which is the default style for the RTF destination.

There are many other styles including the LISTING style, which is the default style for the LISTING destination.

Figure 21.13 HTML Output from PROC REG with DEFAULT Style

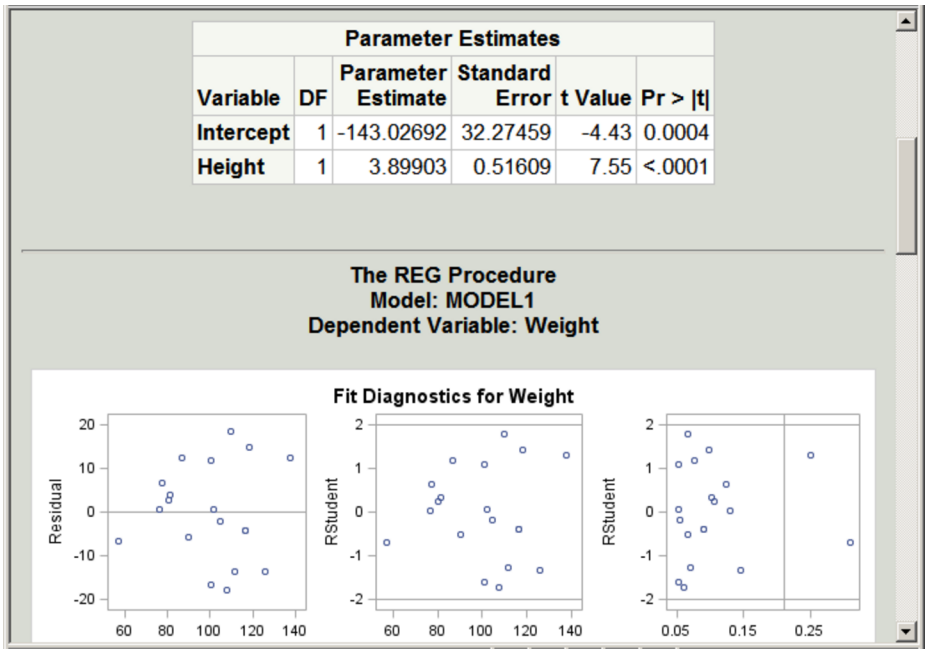


Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-143.02692	32.27459	-4.43	0.0004
Height	1	3.89903	0.51609	7.55	<.0001

The REG Procedure
Model: MODEL1
Dependent Variable: Weight

Figure 21.14 HTML Output from PROC REG with the Statistical Style

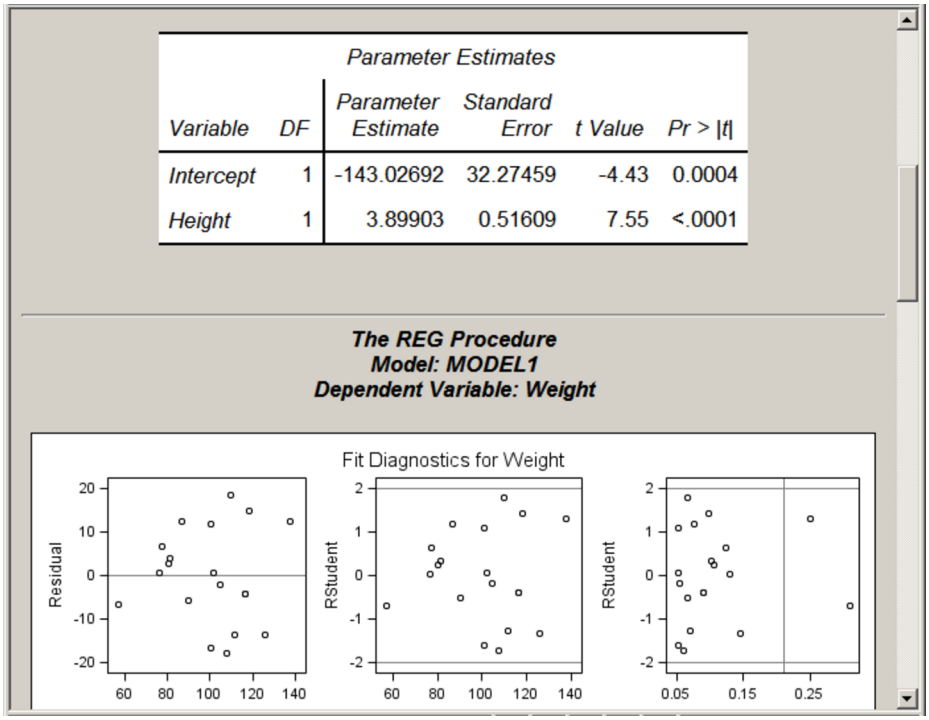


Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-143.02692	32.27459	-4.43	0.0004
Height	1	3.89903	0.51609	7.55	<.0001

The REG Procedure
Model: MODEL1
Dependent Variable: Weight

Figure 21.15 HTML Output from PROC REG with the Journal Style



You specify a style with the STYLE= option in the ODS destination statement. For example, the following statement requests HTML output produced with the JOURNAL style:

```
ods html style=Journal;
```

Similarly, the following statement sets the style for the LISTING destination:

```
ods listing style=Statistical;
```

The style specified with the STYLE= option in the ODS LISTING statement applies only to graphs. The legacy SAS monospace format is used for tables.

The color styles are a compromise in the sense that some graph elements are intentionally over-distinguished to facilitate black-and-white printing. For example, fit lines that correspond to different classification levels are distinguished by both colors and line patterns. You can use the **`ModStyle`** SAS autocall macro (see the section "Creating an All-Color Style by Using the ModStyle Macro" on page 564) to modify a style so that it relies only on color for distinguishability. More generally, you can modify the colors, fonts, and other attributes of graph elements in a style by editing the style definition file. More information is provided in the section "Styles" on page 554, and detailed information is in the *SAS Output Delivery System: User's Guide*.

ODS Destinations

ODS can send your graphs and tables to a number of different destinations including RTF (rich text format), HTML (hypertext markup language), LISTING (the SAS LISTING destination), DOCUMENT (the ODS document), and PDF (portable document format). You use an ODS statement to open a destination, as in the following examples:

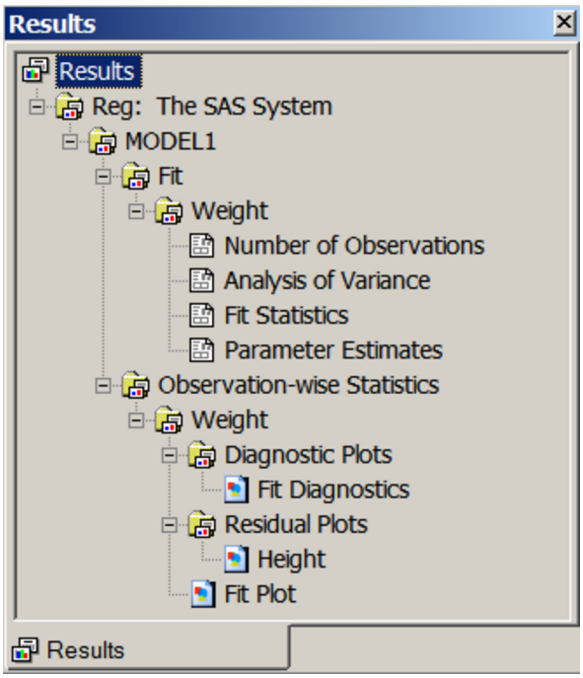
```
ods html body='b.htm';
ods rtf;
ods listing;
ods document name=MyDoc(write);
ods pdf file="contour.pdf";
```

You can close destinations individually or all at once, as in the following examples:

```
ods html close;
ods rtf close;
ods listing close;
ods document close;
ods pdf close;
ods _all_ close;
```

For most ODS destinations (for example, HTML, RTF, and PDF), graphs and tables are integrated in the output, and you view your output with an appropriate viewer, such as a web browser for HTML. However, the default LISTING destination is different. If you are using the LISTING destination in the SAS windowing environment, you view your graphs individually by clicking the graph icons in the Results window, as illustrated in Figure 21.16. This action invokes a host-dependent graph viewer (for example, Microsoft Photo Editor on Windows). The graphs produced with ODS Graphics are *not* displayed with traditional graphs in the Graph window.

Figure 21.16 SAS Results Window



If you are using the SAS windowing environment and you prefer to view integrated output, you should specify a destination such as HTML or RTF. At the same time, you can prevent the Output window from appearing by closing the LISTING destination, as in the following statements:

```
ods listing close;
ods html;
```

A graph is created for every open destination. When you open a new destination, you should close all destinations that you do not need. Closing destinations makes your jobs run faster and with fewer resources, because fewer graphs are produced.

Accessing Individual Graphs

If you are writing a paper or creating a presentation, you need to access your graphs individually. There are various ways to do this, depending on the ODS destination. Three particularly useful methods are as follows:

- If you are viewing RTF output, you can simply copy and paste your graphs from the viewer into a Microsoft Word document or a Microsoft PowerPoint slide.
- If you are viewing HTML output, you can copy and paste your graphs from the viewer, or you can right-click the graph and save it to a file. Copying and pasting from RTF is preferable because the default resolution is higher than with HTML. See the section "Specifying the Size and Resolution of Graphs" on page 525 for details.
- You can save your graphs in image files and then include them into a paper or presentation. For example, you can save your graphs as PNG files and include them into a paper that you are writing with LaTeX or into an HTML document.

You can specify the graphics image format and the file name in the ODS GRAPHICS statement. For example, the following statements, when submitted before a procedure step that produces multiple graphs, save the graphs in PostScript files named *myname.ps*, *myname1.ps*, and so on:

```
ods listing close;
ods latex;
ods graphics on / imagefmt=ps imagename='myname';
```

See the section "Image File Types" on page 541 for details about the file types available with various destinations, how they are named, and how they are saved.

If you are using the LISTING destination and the SAS windowing environment, you can also copy from the default viewer into a Microsoft Word document or a Microsoft PowerPoint slide.

Specifying the Size and Resolution of Graphs

Two factors to consider when you are creating graphs for a paper or presentation are the size of the graph and its resolution. You can specify the size of a graph in the ODS GRAPHICS statement. The following examples show typical ways to change the size of your graphs:

```
ods graphics on / width=6in;
ods graphics on / height=4in;
ods graphics on / width=4.5in height=3.5in;
```

You can change the resolution with the IMAGE_DPI= option in any ODS destination statement, as in the following example:

```
ods html image_dpi=300;
```

The default resolution of graphs created with the HTML and LISTING destinations is 100 DPI (dots per inch), whereas the default with the RTF destination is 200 DPI. An increase in resolution often improves the quality of the graphs, but it also increases the size of the image file. See the section "Graph Size and Resolution" on page 547 for more information about graph size and resolution.

Modifying Your Graphs

Although ODS Graphics is designed to automate the creation of high-quality statistical graphics, on occasion you might need to modify your graphs. There are two ways you can make modifications, depending on whether the changes you want to make are data-dependent and immediate (for a specific graph you are preparing for a paper or presentation), or whether they are persistent (applied to a graph each time you run the procedure). You can make immediate, ad hoc changes by using the ODS Graphics Editor, which provides a point-and-click interface. You can make persistent changes by modifying the ODS graph template for a particular plot. A graph template is a program, written in the Graph Template Language (GTL), that specifies the layout and details of a graph.

NOTE: SAS provides a template for each graph it creates, so you do not need to know anything about templates to create statistical graphics.

You can use the ODS Graphics Editor to customize titles and labels, annotate data points, add text, and change the properties of graph elements. After you have modified your graph, you can save it as a PNG image file or as an SGE file, which preserves the editing context. You can open SGE files with the ODS Graphics Editor and resume editing.

You can invoke the ODS Graphics Editor in the SAS windowing environment, provided that the LISTING destination is open and that you have enabled ODS Graphics to create editable graphs. The steps for doing this are described in the section "ODS Graphics Editor" on page 548. Also see *Getting Started with the SAS/GRAPH Statistical Graphics Editor*.

Figure 21.17 shows the ODS Graphics Editor window for a fit plot created by PROC REG. Figure 21.18 shows modifications made with tools in the ODS Graphics Editor. The title has been changed, and the legend has been repositioned.

Figure 21.17 ODS Graphics Editor Invoked with a Fit Plot

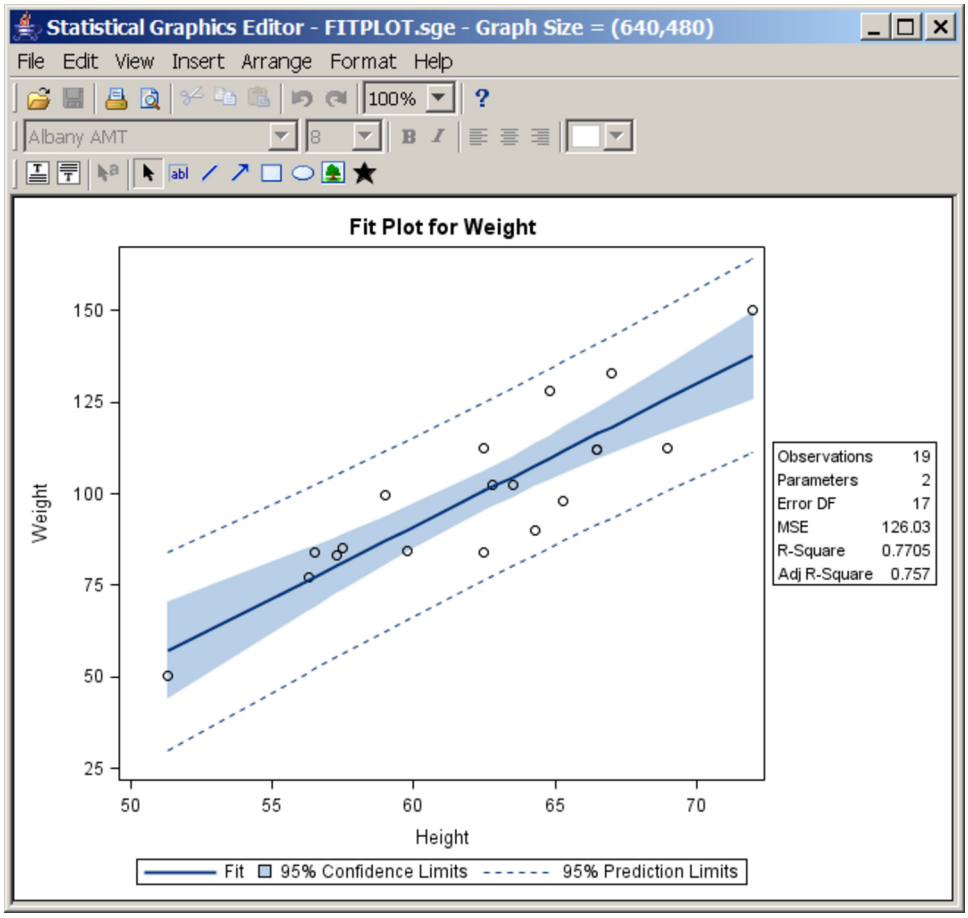
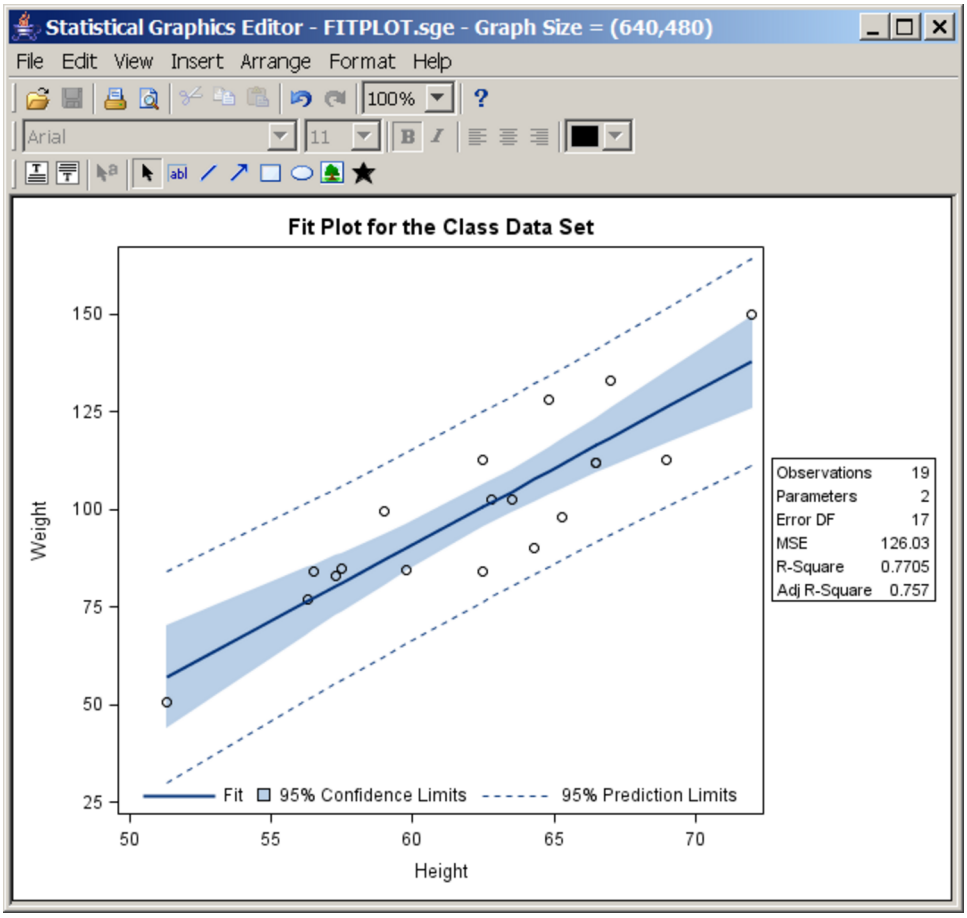


Figure 21.18 Point-and-Click Modifications Made with the ODS Graphics Editor



Procedures That Support ODS Graphics

SAS procedures that support ODS Graphics include the following:

SAS/STAT		**SAS/QC**	**SAS/ETS**
ANOVA	MI	ANOM	ARIMA
BOXPLOT	MIXED	CAPABILITY	AUTOREG
CALIS	MULTTEST	CUSUM	ENTROPY
CLUSTER	NPAR1WAY	MACONTROL	ESM
CORRESP	PHREG	PARETO	EXPAND
FACTOR	PLS	RELIABILITY	MODEL
FREQ	PRINCOMP	SHEWHART	PANEL
GAM	PRINQUAL		SIMILARITY
GENMOD	PROBIT	**Base**	SYSLIN
GLIMMIX	QUANTREG	CORR	TIMESERIES
GLM	REG	FREQ	UCM
GLMSELECT	ROBUSTREG	UNIVARIATE	VARMAX
KDE	RSREG		X12
KRIGE2D	SEQDESIGN		
LIFEREG	SEQTEST		**SAS/HPF**
LIFETEST	SIM2D		HPF
LOESS	TCALIS		HPFENGINE
LOGISTIC	TRANSREG		
MCMC	TTEST		**Risk Dimensions**
MDS	VARIOGRAM		RISK

For details about the specific graphs available with a particular procedure, see the PLOTS= option syntax and the "ODS Graphics" section in the corresponding procedure chapter. The procedure names in the preceding table are links to the "ODS Graphics" section for the SAS/STAT procedures.

Procedures That Support ODS Graphics and Traditional Graphics

A number of procedures that support ODS Graphics produced traditional graphics in previous releases of SAS. These include the UNIVARIATE procedure in Base SAS; the LIFEREG, LIFETEST, and REG procedures in SAS/STAT; and the ANOM, CAPABILITY, CUSUM, MACONTROL, PARETO, RELIABILITY, and SHEWHART procedures in SAS/QC. All of these procedures continue to produce traditional graphics, but in some cases, they do so only when ODS Graphics is not enabled. For more information about the interaction between traditional graphics and ODS graphics in other procedures, see the documentation for that procedure.

Traditional graphs are saved in SAS graphics catalogs and are controlled by the GOPTIONS statement. In contrast, ODS Graphics produces graphs in standard image file formats (not graphics catalogs), and their appearance and layout are controlled by ODS styles and templates.

Syntax

The following sections document some of the most commonly used options in the ODS GRAPHICS statement (section "ODS GRAPHICS Statement" on page 529) and other statements used with ODS Graphics (section "ODS Destination Statements" on page 532). You can find the complete syntax in the *SAS Output Delivery System: User's Guide*. In addition, information about the PLOTS= option is provided in the section "PLOTS= Option" on page 533. Statistical procedures that produce ODS Graphics all have a PLOTS= option that is used to select graphs and control some aspects of the graphs.

ODS GRAPHICS Statement

ODS GRAPHICS < **OFF** | **ON** > < / *options* > **;**

The ODS GRAPHICS statement enables ODS to create graphics. By default, ODS Graphics is not enabled. You can enable ODS Graphics by using either of the following equivalent statements:

```
ods graphics on;
ods graphics;
```

You specify one of these statements prior to your procedure invocation, as illustrated in the examples beginning with "Default Plots for Simple Linear Regression with PROC REG" on page 501. Any procedure that supports ODS Graphics then produces graphics, either by default or when you specify procedure options for requesting particular graphs.

To disable ODS Graphics, specify the following statement:

```
ods graphics off;
```

The following is a subset of the options, syntax, and capabilities available in the ODS GRAPHICS statement. See the *SAS Output Delivery System: User's Guide* for more information.

ANTIALIAS=ON | OFF

controls the use of antialiasing to smooth the components of a graph. Without antialiasing, pixels are simply set or not set. With antialiasing, pixels at the edge of a line or other object are set to an intermediate color, which makes smoother and more professional looking graphics. Text displayed in a graph is always antialiased. Antialiasing is very time consuming for larger graphical displays, and its benefits decrease as the number of points increases, so it is turned off by default for plots with many points. If the number of observations in the ODS output object exceeds the ANTIALIASMAX= threshold (10,000 by default), then antialiasing is not used, even if you specify the option ANTIALIAS=ON. The default is ANTIALIAS=ON.

ANTIALIASMAX=*n*

specifies the maximum number of markers or lines to be antialiased before antialiasing is

disabled. For example, if there are more than 10,000 point markers and ANTIALIASMAX=10,000 (the default), then no markers are antialiased.

BORDER=ON | OFF
specifies whether to draw the graph with a border. BORDER=ON is the default.

HEIGHT=*dimension*
specifies the height of the graph. The default is HEIGHT=480PX (480 pixels). You can also specify height in inches (for example, HEIGHT=5IN) or centimeters (for example, HEIGHT=12CM).

IMAGEFMT=< *image-file-type* **| STATIC >**
specifies the image format for graphs. By default, IMAGEFMT=STATIC and ODS dynamically uses the best quality static image format for the active output destination. The available image formats include: BMP (Microsoft Windows device independent bitmap), DIB (Microsoft Windows device independent bitmap), EMF (Microsoft NT enhanced metafile), EPSI (Adobe encapsulated PostScript interchange), GIF (graphics interchange format), JFIF (JPEG file interchange format), JPEG (Joint Photographic Experts Group), PBM (portable bitmap), PCD (Photo CD), PDF (Portable Document Format), PICT (the QuickDraw Picture Format), PNG (Portable Network Graphic), PS (PostScript image file format), TIFF (tagged image file format), WMF (Microsoft Windows Metafile Format), XBM (X11 Bitmap graphics), and XPM (X11 Pixel Map graphics). If the specified image format is not valid for the active output destination, the device is automatically remapped to the default image format.

IMAGEMAP=ON | OFF
controls tooltip generation in the HTML destination. The default is IMAGEMAP=OFF, which means that no tooltips are generated. Tooltips are text boxes that appear in HTML output when you hover over a part of the plot with your mouse pointer (see Example 21.1).

IMAGENAME=< *base-file-name* **>**
specifies the base image file name. The default is the name of the output object. You can determine the name of the output object by using the ODS TRACE statement (see the section "Determining Graph Names and Labels" on page 538). The base image name should not include an extension. ODS automatically adds the increment value and the appropriate extension (which is specific to the output destination). See the section "Specifying Base File Names" on page 543 for an example.

LABELMAX=*n*
specifies the maximum number of labeled areas before labeling is disabled. For example, if LABELMAX=50, and there are more than 50 points with labels, then no points are labeled. The default is LABELMAX=200.

MAXLEGENDAREA=*n*
specifies the maximum percentage of the overall graph area that a legend can occupy. The default is MAXLEGENDAREA=20. Larger legends are dropped from the display.

RESET< *=option* **>**
resets one or more ODS GRAPHICS options to their default settings. The RESET and RESET=ALL options are equivalent. If you want to reset more than one option, but not all of the

options, then you must specify RESET= separately for each option you reset (for example, `ods graphics on / reset=antialias reset=index;`). The RESET= options include the following:

ALL
resets all of the resettable options to their defaults.

ANTIALIAS
resets the ANTIALIAS= option to its default.

ANTIALIASMAX
resets the ANTIALIASMAX= option to its default.

BORDER
resets the BORDER= option to its default.

INDEX
resets the index counter that is appended to static image files.

HEIGHT
resets the HEIGHT= option to its default.

IMAGEMAP
resets the IMAGEMAP= option to its default.

LABELMAX
resets the LABELMAX= option to its default.

SCALE
resets the SCALE= option to its default.

TIPMAX
resets the TIPMAX= option to its default.

WIDTH
resets the WIDTH= option to its default.

SCALE=ON | OFF
specifies whether the fonts and symbol markers are scaled proportionally with the size of the graph. The default is SCALE=ON. For examples, see Figure 21.29 and Figure 21.30.

TIPMAX=*n*
specifies the maximum number of distinct tooltips permitted before tooltips are disabled. Tooltips are text boxes that appear when you hover over a part of the plot with your mouse pointer. For example, if TIPMAX=400, and there are more than 400 points in a scatter plot, then no tooltips appear. The default is TIPMAX=500.

WIDTH=*dimension*
specifies the width of the graph. The default is WIDTH=640PX (640 pixels). You can also specify widths in inches (for example, WIDTH=5in) or centimeters (for example, WIDTH=12cm).

ODS Destination Statements

ODS has a number of statements that control the destination of ODS output. The ODS destination statements that are most commonly used with ODS Graphics are: ODS DOCUMENT, ODS HTML, ODS LATEX, ODS LISTING, ODS PCL, ODS PDF, ODS PS, and ODS RTF. Specifying a statement opens a destination, unless the CLOSE option is specified. Each of the following statements opens an ODS destination:

```
ods html;
ods rtf;
ods html image_dpi=300;
ods listing style=Statistical;
```

Each of the following statements closes an ODS destination:

```
ods html close;
ods rtf close;
ods listing close;
```

There are two options that are commonly used in ODS destination statements to control aspects of ODS Graphics:

IMAGE_DPI=*dpi*
: specifies the dots per inch (DPI), which is the image resolution for graphical output. The default varies depending on the destination. For example, the default is 100 for HTML and 200 for RTF.

STYLE=*style-name*
: specifies the style definition. Commonly used styles include DEFAULT, LISTING, STATISTICAL, JOURNAL, JOURNAL2, RTF, and ANALYSIS.

Other options provide you with ways to control the files that are created. For example, the following statement opens the HTML destination:

```
ods html body='b.html' contents='c.html' frame='a.html';
```

This statement also writes the body of the output to the file *b.html*, the table of contents to the file *c.html*, and an overall frame containing both the contents and the output to the file *a.html*. Alternatively, you can specify FILE= instead of BODY=.

If you are using a destination for which individual graphs are created (for example, LISTING or HTML), you can use the GPATH= option to specify the directory where your graphics files are saved, as in the following example:

```
ods html gpath="C:\figures";
```

See the sections "Image File Types" on page 541, "Saving Graphics Image Files" on page 544, "LISTING Destination" on page 545, "HTML Destination" on page 545, and "LATEX Destination" on page 546 for more information about individual image files and options specified in the ODS

Destination statements. For complete details about the ODS destination statements, see the *SAS Output Delivery System: User's Guide*.

PLOTS= Option

Each statistical procedure that produces ODS Graphics has a PLOTS= option that is used to select graphs and specify some options. The syntax of the PLOTS= option is as follows:

PLOTS <(*global-plot-options*)> <= *plot-request* <*(options)*>>
PLOTS <(*global-plot-options*)> <= (*plot-request* <*(options)*> <... *plot-request* <*(options)*>>)>

The PLOTS= option has a common overall syntax for all statistical procedures, but the specific global plot options, plot requests, and plot options vary across procedures. This section discusses only a few of the options available in the PLOTS= option. For more information about the PLOTS= option, see the "Syntax" section for each procedure that produces ODS Graphics. There are only a limited number of things that you can control with the PLOTS= option. Most graphical details are controlled either by graph templates (see the section "Graph Templates" on page 577) or by styles (see the section "Styles" on page 554).

The PLOTS= option usually appears in the PROC statement. However, for some procedures, certain analyses and hence certain plots can appear only if an additional statement is specified. These procedures often have a PLOTS= option in that other statement. For example, the PHREG procedure has a PLOTS= option in the BAYES statement, which is used to perform a Bayesian analysis. See the "Syntax" section of each procedure chapter for more information. The following examples illustrate the syntax of the PLOTS= option:

```
plots=all
plots=none
plots=residuals
plots=residuals(smooth)
plots=(trace autocorr)
plots(unpack)
plots(unpack)=diagnostics
plots=diagnostics(unpack)
plots(only)=freqplot
plots=(scree(unpack) loadings(plotref) preloadings(flip))
plots(unpack maxparmlabel=0 stepaxis=number)=coefficients
plots(sigonly)=(rawprob adjusted(unpack))
```

Also see the "Getting Started" sections "Survival Estimate Plot with PROC LIFETEST" on page 504, "Contour and Surface Plots with PROC KDE" on page 505, "Contour Plots with PROC KRIGE2D" on page 507, "LS-Means Diffogram with PROC GLIMMIX" on page 513, and "Principal Component Analysis Plots with PROC PRINCOMP" on page 515 for examples of the PLOTS= option.

The simplest PLOTS= specifications are of the form PLOTS=*plot-request* or PLOTS=(*plot-requests*). When there is more than one plot request, the plot-request list must appear in parentheses. Each plot request either requests a plot (for example, RESIDUALS) or provides you with a

place to specify plot-specific options (for example, DIAGNOSTICS(UNPACK)). Some simple and typical plot requests are explained next:

- PLOTS=ALL requests all plots that are relevant to the analysis. This does not mean that all plots that the procedure can produce are produced. Plots that are produced for one set of options might not appear with PLOTS=ALL and a different set of options. In some cases, certain plots are not produced unless certain options or statements outside the PLOTS= option are specified.
- PLOTS=NONE disables ODS Graphics for just that step. You can use this option instead of specifying ODS GRAPHICS OFF before a procedure step and ODS GRAPHICS ON after the step when you want to suppress graphics for only that step.
- PLOTS=RESIDUALS requests a plot of residuals in a modeling procedure such as PROC REG.
- PLOTS=RESIDUALS(SMOOTH) requests the residuals plot along with a smooth fit function.
- PLOTS=(TRACE AUTOCORR) requests trace and autocorrelation plots in procedures with Bayesian analysis options.

Global plot options appear in parentheses after the option name and before the equal sign. These options affect many or all of the plots. The UNPACK option is a commonly used global plot option. It specifies that plots that are normally produced with multiple plots per panel (or "packed") should be unpacked and appear in multiple panels with one plot in each panel. The specification PLOTS(UNPACK)=(*plot-requests*) unpacks all paneled plots. The UNPACK option is also used as an option in a plot request when you only want to unpack certain panels. For example, the option PLOTS=(DIAGNOSTICS(UNPACK) PARTIAL PREDICTIONS) unpacks just the diagnostics panel. In some cases, unpacked plots contain additional information that is not found in the smaller packed versions. The UNPACK option is not available for all plot requests; it is just available with plots that have multiple panels by default.

Another commonly used global plot option is the ONLY option. Many procedures produce default plots, and additional plots can be requested in the PLOTS= option. Specifying PLOTS=(*plot-requests*) while omitting the default plots does not prevent the default plots from being produced. The ONLY option is used when you only want to see the plots specifically listed in the plot-request list. Procedures that produce no default plots typically do not provide an ONLY option. You can use ODS SELECT and ODS EXCLUDE (see the section "Selecting and Excluding Graphs" on page 540) to select and exclude graphs, but in some situations the ONLY option is more convenient. It is typically more efficient to select plots by using the PLOTS(ONLY)= option, because the procedure does not do extra work to generate a plot that is excluded by the PLOTS(ONLY)= option. In contrast, ODS SELECT and ODS EXCLUDE have their effect after the procedure has done the work to generate the plot.

Selecting and Viewing Graphs

This section describes techniques for selecting and viewing your graphs. Topics include:

- specifying an ODS destination for graphics
- viewing your graphs in the SAS windowing environment
- referring to graphs by name when using ODS
- selecting and excluding graphs from your output

Specifying an ODS Destination for Graphics

If you do not specify an ODS destination, then the LISTING destination is used by default. Here is an example of how you can specify the HTML destination:

```
ods graphics on;
ods html;

proc reg data=sashelp.class;
   model Weight = Height;
run; quit;

ods html close;
```

This ODS HTML statement creates an HTML file with a default name. See the section "Specifying a File for ODS Output" on page 536 to see how to specify a file name. Other destinations are specified in a similar way. For example, you can specify an RTF destination with the following statements:

```
ods graphics on;
ods rtf;

. . .

ods rtf close;
```

The destinations that ODS supports for graphics are as follows:

Destination	Destination Family
DOCUMENT	
HTML	MARKUP
LATEX	MARKUP
LISTING	
PCL	PRINTER
PDF	PRINTER
PS	PRINTER
RTF	

You can close the LISTING destination if you are only interested in displaying your output in a different destination. For example, if you want to see your output only in the RTF destination, you can specify the following statements:

```
ods graphics on;
ods listing close;
ods rtf;

. . .

ods rtf close;
ods listing;
```

Closing unneeded destinations makes your jobs run faster and creates fewer files. More generally, it makes your jobs consume fewer resources, because a graph is otherwise created for every open destination. The last statement opens the LISTING destination after you are finished using the RTF destination.

You can also use the ODS OUTPUT destination to create an output data set from the data object used to make a plot. Here is an example:

```
ods graphics on;

proc reg data=sashelp.class;
   ods output fitplot=myfitplot;
   model Weight = Height;
run; quit;
```

Specifying a File for ODS Output

You can specify a file name for your output with the FILE= option in the ODS destination statement, as in the following example:

```
ods html file="test.htm";
```

The output is written to the file *test.htm*, which is saved in the SAS current folder. At startup, the SAS current folder is the same directory in which you started your SAS session. If you are using the SAS windowing environment, then the current folder is displayed in the status line at the bottom of the main SAS window. If you do not specify a file name for your output, then SAS provides

a default file name, which depends on the ODS destination. This file is saved in the SAS current folder. You can always check the SAS log to verify the name of the file in which your output is saved. For example, suppose you specify the following statement:

```
ods html;
```

Then the following message is displayed in the SAS log:

```
NOTE: Writing HTML Body file: sashtml.htm
```

The default file names for each destination are specified in the SAS Registry. For example, Figure 21.31 shows that the default file name in the SAS Registry for the RTF destination is *sasrtf.rtf*. For more information, see the SAS Companion for your operating system.

Viewing Your Graphs in the SAS Windowing Environment

The mechanism for viewing graphics created with ODS can vary depending on your operating system, which viewers are installed on your computer, and the ODS destination you have selected. If you do not specify an ODS destination, then the LISTING destination is used by default. If you are using the SAS windowing environment, go to the Results window and find the icon for the corresponding graph. You can double-click the graph icon to display the graph in the default viewer that is configured on your computer for the corresponding image file type (see Figure 21.16).

If you are using the SAS windowing environment and you specify an HTML destination, then the results are displayed by default in the SAS Results window as they are being generated. Depending on your configuration, this can also apply to the PDF and RTF destinations. If you are using the LATEX or the PS destinations, you must use a PostScript viewer, such as GSview. For information about the windowing environment in a different operating system, see the SAS Companion for that operating system.

If you do not want to view the results as they are being generated, then select **Tools ► Options ► Preference** from the menu at the top of the main SAS window. Then in the **Results** tab, clear **View results as they are generated** checkbox.

You can change the default to use an external viewer instead of the Results window. Select **Tools ► Options ► Preferences** from the menu at the top of the main SAS window. Then in the **Results** tab select **Preferred web browser**. Your results are displayed in the default viewer that is configured on your computer for the corresponding destination.

You can also choose your browser for HTML output. Select **Tools ► Options ► Preferences** from the menu at the top of the main SAS window. Then in the **Web** tab, select **Other browser**, and type (or browse) the path of your preferred browser.

Determining Graph Names and Labels

Procedures assign a name to each graph they create with ODS Graphics. This enables you to refer to ODS graphs in the same way that you refer to ODS tables (see the section "Paths and Selection" on page 435 in Chapter 20, "Using the Output Delivery System"). You can determine the names of graphs in several ways:

- You can look up graph names in the "ODS Graphics" section of chapters for procedures that use ODS Graphics. For example, see the section "ODS Graphics" on page 5589 in Chapter 73, "The REG Procedure."
- You can use the Results window to view the names of ODS graphs created in your SAS session. See the section "ODS and the SAS Results Window" on page 440 in Chapter 20, "Using the Output Delivery System," for more information.
- You can use the ODS TRACE ON statement to list the names of graphs created by your SAS session. This statement adds identifying information in the SAS log (or optionally in the SAS LISTING) for each graph that is produced. See the section "Paths and Selection" on page 435 in Chapter 20, "Using the Output Delivery System," for more information.

The graph name is not the same as the name of the image file that contains the graph (see the section "Naming Graphics Image Files" on page 542).

This example revisits the analysis described in the section "Contour and Surface Plots with PROC KDE" on page 505. To determine which output objects are created by ODS, you specify the ODS TRACE ON statement prior to the procedure statements as follows:

```
ods graphics on;
ods trace on;

proc kde data=bivnormal;
   bivar x y / plots=contour surface;
run;

ods trace off;
```

The trace record from the SAS log is as follows:

```
Output Added:
-------------
Name:        Inputs
Template:    Stat.KDE.Inputs
Path:        KDE.Bivar1.x_y.Inputs
-------------

Output Added:
-------------
Name:        Controls
Template:    Stat.KDE.Controls
```

```
Path:       KDE.Bivar1.x_y.Controls
-------------

Output Added:
-------------
Name:       ContourPlot
Label:      Contour Plot
Template:   Stat.KDE.Graphics.Contour
Path:       KDE.Bivar1.x_y.ContourPlot
-------------

Output Added:
-------------
Name:       SurfacePlot
Label:      Density Surface
Template:   Stat.KDE.Graphics.Surface
Path:       KDE.Bivar1.x_y.SurfacePlot
-------------
```

By default, PROC KDE creates table objects named **Inputs** and **Controls**, and it creates graph objects named **ContourPlot** and **SurfacePlot**. In addition to the name, the trace record provides the label, template, and path for each output object. Graph templates are distinguished from table templates by a naming convention that uses the procedure name in the second level and **Graphics** in the third level. For example, the fully qualified template name for the surface plot created by PROC KDE is **Stat.KDE.Graphics.SurfacePlot**.

You can specify the LISTING option in the ODS TRACE ON statement to write the trace record to the LISTING destination as follows:

```
ods trace on / listing;
```

Each table and graph has a path, which was previously shown in the trace output. The path consists of the plot name preceded by the names of one or more output groups. Each table and graph also has a label path, which can be seen by adding the LABEL option to the ODS TRACE ON statement, after a slash, as follows:

```
ods trace on / label;

proc kde data=bivnormal;
   bivar x y / plots=contour surface;
run;

ods trace off;
```

A portion of the trace output is shown next:

```
Path:       KDE.Bivar1.x_y.Inputs
Label Path: 'The KDE Procedure'.'Bivariate Analysis'.'x and y'.'KDE.Bivar1.x_y'

Path:       KDE.Bivar1.x_y.Controls
Label Path: 'The KDE Procedure'.'Bivariate Analysis'.'x and y'.'KDE.Bivar1.x_y'

Path:       KDE.Bivar1.x_y.ContourPlot
```

```
Label Path: 'The KDE Procedure'.'Bivariate Analysis'.'x and y'.'Contour Plot'

Path:       KDE.Bivar1.x_y.SurfacePlot
Label Path: 'The KDE Procedure'.'Bivariate Analysis'.'x and y'.'Density Surface'
```

The label path contains the information that you see in the HTML table of contents. Names are fixed, they do not vary, and they are not data- or context-dependent. In contrast, labels often reflect data- or context-dependent information.

Selecting and Excluding Graphs

You can use the ODS SELECT and ODS EXCLUDE statements along with graph and table names to specify which ODS outputs are displayed. See the section "Paths and Selection" on page 435 in Chapter 20, "Using the Output Delivery System," for more information about how to use these statements.

This section shows several examples of selecting and excluding graphs by using the data set and trace output created in the section "Determining Graph Names and Labels" on page 538. The following statements use the ODS SELECT statement to select only two graphs, **ContourPlot** and **SurfacePlot**, for display in the output:

```
proc kde data=bivnormal;
   ods select ContourPlot SurfacePlot;
   bivar x y / plots=contour surface;
run;
```

Equivalently, the following statements use the ODS EXCLUDE statement to exclude the two tables:

```
proc kde data=bivnormal;
   ods exclude Inputs Controls;
   bivar x y / plots=contour surface;
run;
```

You can select or exclude graphs by using either the name or the label. Labels must be specified in quotes. In the context of this example, the following two statements are equivalent:

```
ods select contourplot;
ods select 'Contour Plot';
```

You can also specify multiple levels of the path, as in the following example:

```
ods select x_y.contourplot;
ods select 'x and y'.'Contour Plot';
ods select 'x and y'.contourplot;
ods select x_y.'Contour Plot';
```

Paths and label paths can be mixed, as in the last two statements. All four of the preceding statements select the same plot. Furthermore, selection based directly on the names and labels is case insensitive. The following all select the same plot:

```
ods select x_y.contourplot;
ods select 'x and y'.'Contour Plot';
```

```
ods select X_Y.CONTOURPLOT;
ods select 'X AND Y'.'CONTOUR PLOT';
```

It is sometimes useful to specify a WHERE clause in an ODS SELECT or ODS EXCLUDE statement. This enables you to specify expressions based on either the path or the label path. You can base your selection on two automatic variables _path_ and _label_. The following two statements select every object whose path contains the string 'plot' and every object whose label path contains the string 'plot', respectively, ignoring the case in the name and label:

```
ods select where = (lowcase(_path_)  ? 'plot');
ods select where = (lowcase(_label_) ? 'plot');
```

The question mark operator means that the second expression (the string 'plot') is contained in the first expression (the lowercase version of the name or label). For example, all of the following names match 'plot' in the WHERE clause: plot, SurfacePlot, SURFACEPLOT, FitPlot, pLoTtInG, Splotch, and so on. Since WHERE clause selection is based on SAS string comparisons, selection is case sensitive. The LOWCASE function is used to ensure a match even when the specified string does not match the case of the actual name or label.

WHERE clauses are particularly useful when you want to select all of the objects in a group. A group is a level of the path or label path hierarchy before the last level. In the following step, all of the objects whose name path contains 'DiagnosticPlots' are selected:

```
proc reg data=sashelp.class plots(unpack);
   ods select where = (_path_ ? 'DiagnosticPlots');
   model Weight = Height;
run; quit;
```

These are the plots that come from unpacking the PROC REG diagnostics panel of plots. All are in a group named 'DiagnosticPlots'.

Graphics Image Files

Accessing your graphs as individual image files is useful when you want to include them in various types of documents. The default image file type depends on the ODS destination, but there are other supported image file types that you can specify. You can also specify the names for your graphics image files and the directory in which you want to save them. This section describes the image file types supported by ODS Graphics, and it explains how to name and save graphics image files.

Image File Types

If you are using the LISTING, HTML or LATEX destinations, your graphs are individually produced in a specific image file type, such as PNG (Portable Network Graphics). If you are using a destination in the PRINTER family or the RTF destination, the graphs are contained in the ODS output file and cannot be accessed as individual image files. However, you can open an RTF out-

put file in Microsoft Word and then copy and paste the graphs into another document, such as a Microsoft PowerPoint presentation. This is illustrated in Example 21.2.

Table 21.1 shows the various ODS destinations supported by ODS Graphics, the viewer that is appropriate for displaying graphs in each destination, and the image file types supported for each destination.

Table 21.1 Destinations and Image File Types Supported by ODS Graphics

Destination	Destination Family	Recommended Viewer	Image File Types
DOCUMENT		Not applicable	Not applicable
HTML	MARKUP	Web browser	PNG (default), GIF, JPEG,
LATEX	MARKUP	PostScript or PDF viewer after compiling the LaTeX file	PostScript (default), EPSI, GIF, JPEG, PDF, PNG
LISTING		Default viewer in your system for the specified file type	PNG (default), GIF, BMP, DIB, EMF, EPSI, GIF, JFIF, JPEG, PBM, PS, TIFF, WMF
PCL	PRINTER	not applicable	Contained in PRN file
PDF	PRINTER	PDF viewer, such as Adobe Reader	Contained in PDF file
PS	PRINTER	PostScript viewer, such as GSview	Contained in PostScript file
RTF		Word processor, such as Microsoft Word	Contained in RTF file

For destinations such as PDF and RTF, you can control the types of the images that are contained in the file even though individual files are not made for each image. The default image file type is PNG, and other image types are available. See the *SAS Output Delivery System: User's Guide* for more information.

Naming Graphics Image Files

The following discussion applies to the destinations where ODS graphs are created as individual image files (for example, HTML, LISTING, and LATEX). The names of graphics image files are determined by a base file name, an index counter, and an extension. By default, the base file name is the ODS graph name (see the section "Determining Graph Names and Labels" on page 538). There is an index counter for each base file name. The extension indicates the image file type. The first time a graph object with a given base file name is created, the file name consists only of the base file name and the extension. If a graph with the same base file name is created multiple times, then an index counter is appended to the base file name to avoid overwriting previously created images.

To illustrate, consider the following statements:

```
proc kde data=bivnormal;
   ods select ContourPlot SurfacePlot;
   bivar x y / plots=contour surface;
run;
```

If you run this step at the beginning of a SAS session, the two graphics image files created are *ContourPlot.png* and *SurfacePlot.png*. If you immediately rerun these statements, then ODS creates the same graphs in different image files named *ContourPlot1.png* and *SurfacePlot1.png*. The next time, the image files are named *ContourPlot2.png* and *SurfacePlot2.png*. The index starts at zero, and one is added each time the same name is used. Note, however, that when the index is at zero, it is not added to the file name.

Resetting the Index Counter

You can specify the RESET=INDEX option in the ODS GRAPHICS statement to reset the index counter. This is useful when you need to have predictable names. It is particularly useful when you are running a SAS program multiple times in the same session. The following statement resets the index:

```
ods graphics on / reset=index;
```

The index counter is reinitialized at the beginning of your SAS session or if you specify the RESET=INDEX option in the ODS GRAPHICS statement. Graphics image files with the same name are overwritten.

Specifying Base File Names

You can specify a base file name for all your graphics image files with the IMAGENAME= option in the ODS GRAPHICS statement as follows:

```
ods graphics on / imagename="MyName";
```

You can also specify the RESET=INDEX option as follow:

```
ods graphics on / reset=index imagename="MyName";
```

The IMAGENAME= option overrides the default base file name. With the preceding statement, the graphics image files are named *MyName*, *MyName1*, *MyName2*, and so on.

Specifying Image File Types

You can specify the image file type for the LISTING, HTML, or LATEX destinations with the IMAGEFMT= option in the ODS GRAPHICS statement as follows:

```
ods graphics on / imagefmt=gif;
```

For more information, see the section "ODS GRAPHICS Statement" on page 529.

Naming Graphics Image Files with Multiple Destinations

Since the index counter depends only on the base file name, if you specify multiple ODS destinations for your output, then the index counter is increased independently of the destination. For example, the following statements create image files named *ContourPlot.png* and *SurfacePlot.png* that correspond to the LISTING destination; and *ContourPlot1.png* and *SurfacePlot1.png* that correspond to the HTML destination:

```
ods listing;
ods html;
ods graphics on / reset;

proc kde data=bivnormal;
   ods select ContourPlot SurfacePlot;
   bivar x y / plots=contour surface;
run;

ods _all_ close;
ods listing;
```

When you specify one of the destinations in the PRINTER family or the RTF destination, your ODS graphs are embedded in the document, so the index counter is not affected. For example, the following statements create image files *ContourPlot.png* and *SurfacePlot.png* for the LISTING destinations, but no image files for the RTF destination:

```
ods listing;
ods rtf;
ods graphics on / reset;

proc kde data=bivnormal;
   ods select ContourPlot SurfacePlot;
   bivar x y / plots=contour surface;
run;

ods _all_ close;
```

Saving Graphics Image Files

Knowing where your graphics image files are saved and how they are named is particularly important if you are running in batch mode, if you have disabled the SAS Results window (see the section "Viewing Your Graphs in the SAS Windowing Environment" on page 537), or if you plan to access the files for inclusion in a paper or presentation. The following discussion assumes you are running SAS under the Windows operating system. If you are running on a different operating system, see the SAS Companion for your operating system.

Your graphics image files are saved by default in the SAS current folder. If you are using the SAS windowing environment, the current folder is displayed in the status line at the bottom of the main SAS window. If you are running your SAS programs in batch mode, the graphs are saved by default

in the same directory where you started your SAS session. For example, suppose the SAS current folder is *C:\myfiles*. If you specify the ODS GRAPHICS statement, then your graphics image files are saved in the directory *C:\myfiles*. Unlike traditional graphics, ODS Graphics are not saved in a catalog in your WORK directory.

With the LISTING, HTML, and LATEX destinations, you can specify a directory for saving your graphics image files. With a destination in the PRINTER family and with the RTF destination, you can specify a directory only for your output file. The remainder of this discussion provides details for each destination type.

LISTING Destination

If you are using the LISTING destination, the individual graphs are created as PNG files by default. You can use the GPATH= option in the ODS LISTING statement to specify the directory where your graphics files are saved. For example, if you want to save your graphics image files in *C:\figures*, then you can specify the following:

```
ods listing gpath="C:\figures";
```

It is important to note that the GPATH= option applies only to ODS Graphics. It does not affect the behavior of graphics created with traditional SAS/GRAPH procedures.

HTML Destination

If you are using the HTML destination, the individual graphs are created as PNG files by default. You can use the PATH= and GPATH= options in the ODS HTML statement to specify the directory where your HTML and graphics files are saved, respectively. This also gives you more control over your graphs. For example, if you want to save your HTML file named *test.htm* in the *C:\myfiles* directory, but you want to save your graphics image files in *C:\myfiles\png*, then you can specify the following:

```
ods html path  ="C:\myfiles"
         gpath = "C:\myfiles\png"
         file  = "test.htm";
```

When you specify the URL= suboption with the GPATH= option, SAS creates relative paths for the links and references to the graphics image files in the HTML file. This is useful for building output files that are easily moved from one location to another. For example, the following statements create a relative path to the *png* directory in all the links and references contained in *test.htm*:

```
ods html path  = "C:\myfiles"
         gpath = "C:\myfiles\png" (url="png/")
         file  = "test.htm";
```

If you do not specify the URL= suboption, SAS creates absolute paths that are hard-coded in the HTML file. These can cause broken links if you move the files. For more information, see the ODS HTML statement in the *SAS Output Delivery System: User's Guide*.

LATEX Destination

LaTeX is a document preparation system for high-quality typesetting. The ODS LATEX statement produces output in the form of a LaTeX source file that is ready to compile in LaTeX. When you request ODS Graphics for a LATEX destination, ODS creates the requested graphs as PostScript files by default, and the LaTeX source file includes references to these image graphics files. You can compile the LaTeX file, or you can ignore this file and simply access the individual PostScript files to include your graphs in a different LaTeX document, such as a paper that you are writing. You can specify the PATH= and GPATH= options in the ODS LATEX statement, as explained previously for the ODS HTML statement. See Example 21.3 for an illustration. The ODS LATEX statement is an alias for the ODS MARKUP statement with the TAGSET=LATEX option. For more information, see the *SAS Output Delivery System: User's Guide*.

The default image file type for the LATEX destination is PostScript. When you use LaTeX to compile your document, the graphics format for included images is Postscript. However, if you prefer to use pdfLaTeX, you can specify a different format such as JPEG, PDF, or PNG, any of which can be directly included into your pdfLaTeXdocument. To specify one of these formats, you use the IMAGEFMT= option in the ODS GRAPHICS statement. For more information, see the LaTeX documentation for the `graphicx` package.

Creating Graphs in Multiple Destinations

This section illustrates how to send your output to more than one destination with a single execution of your SAS statements. For example, to create the default LISTING output and also both HTML and RTF output, you can specify the ODS HTML and the ODS RTF statements before your procedure statements.

```
ods html;
ods rtf;

. . .

ods _all_ close;
```

The ODS _ALL_ CLOSE statement closes all open destinations.

You can also specify multiple instances of the same destination. For example, using the data in the section "Contour and Surface Plots with PROC KDE" on page 505, the following statements save the contour plot to the file *contour.pdf* and the surface plot to the file *surface.pdf*:

```
ods pdf file="contour.pdf";
ods pdf select ContourPlot;
ods pdf(id=srf) file="surface.pdf";
ods pdf(id=srf) select SurfacePlot;
ods graphics on;

proc kde data=bivnormal;
   ods select ContourPlot SurfacePlot;
```

```
   bivar x y / plots=contour surface;
run;

ods _all_ close;
```

The ID= option assigns the name **srf** to the second instance of the PDF destination. Without the ID= option, the second ODS PDF statement closes the destination that was opened by the previous ODS PDF statement, and it opens a new instance of the PDF destination. In that case, the file *contour.pdf* is not created. For more information, see the ODS PDF statement in the *SAS Output Delivery System: User's Guide*.

Graph Size and Resolution

ODS provides options for specifying the size and resolution of graphs. You can specify the size of a graph in the ODS GRAPHICS statement and the resolution in an ODS destination statement. There are two other ways to change the size of a graph, but they are rarely needed. The three methods are as follows:

- Usually, you specify WIDTH=, HEIGHT=, or both in the ODS GRAPHICS statement to change the size of a graph.
- To modify the size of a particular graph, specify the dimensions with the DESIGNHEIGHT= and DESIGNWIDTH= options in the BEGINGRAPH statement in the template definition. Some templates contain the specification DESIGNWIDTH=DEFAULTDESIGNHEIGHT, which sets the width of the graph to the default height, or DESIGNHEIGHT=DEFAULTDESIGNWIDTH, which sets the height of the graph to the default width.
- To modify the size of all of your ODS graphs, specify the dimensions with the OUTPUTHEIGHT= and OUTPUTWIDTH= options in the style definition.

The following examples show typical ways to change the size of your graphs:

```
ods graphics on / width=6in;
ods graphics on / height=4in;
ods graphics on / width=4.5in height=3.5in;
```

The dimensions of the graph can be specified in pixels (for example, 200PX), inches (for example, 3IN), or centimeters (for example, 8CM). The default dimensions of ODS Graphics are 640 pixels wide and 480 pixels high, and these values determine the default aspect ratio. The actual size of the graph in inches depends on your printer or display device. For example, if the resolution of your printer is 100 dots per inch, and you want a graph that is 4 inches wide, you should set the width to 400 pixels.

If you specify only one dimension, the other is determined by the default aspect ratio—that is, height=0.75 × width. For best results, you should create your graphs by using the exact size that is

used to display the graphs in your paper or presentation. In other words, avoid generating them at one size and then expanding or shrinking them for inclusion into the your document.

By default, fonts and symbol markers are automatically scaled with the size of the graph. You can suppress this scaling with the SCALE= option, as in the following example:

```
ods graphics on / scale=off;
```

The default resolution of graphs created with HTML and LISTING is 100 DPI (dots per inch), whereas the default with RTF is 200 DPI. The 200 DPI value is recommended if you are copying and pasting graphs into a Microsoft PowerPoint presentation or a Microsoft Word document. Graphs shown in SAS/STAT documentation are typically generated at 300 DPI for display in PDF and 100 DPI for display in HTML.

You can change the resolution with the IMAGE_DPI= option in any ODS destination statement, as in the following example:

```
ods html image_dpi=300;
```

An increase in resolution often improves the quality of the graphs, but it also greatly increases the size of the image file. Going from 100 DPI to 300 DPI increases the size of the image file by roughly a factor of $(300/100)^2 = 9$. Even when you are using a higher DPI for most of your graphs, you should consider using a lower DPI for some, such as contour plots, that create large files even at a lower DPI.

If you increase the resolution, you might need to compensate by reducing the size of the graph, as in the following example:

```
ods graphics on / width=4.5in  height=3.5in;
```

Increasing DPI also increases the amount of memory needed for your program to complete. You can increase the amount of memory available to ODS Graphics with an option when you invoke SAS, as in the following example:

```
-jreoptions '(-Xmx256m)'
```

You can modify the default amount of memory available to ODS Graphics by changing JREOPTIONS in your SAS configuration file to the settings –Xmx*nnn*m –Xms*nnn*m, where *nnn* is the amount of memory in megabytes. An example is –Xmx256m –Xms256m. In either case, the exact syntax varies depending on your operating system, and the amount of memory that you can allocate varies from system to system. For more information, see the SAS Companion for your operating system.

ODS Graphics Editor

The ODS Graphics Editor is a point-and-click interface that you can use to modify a specific graph created by ODS Graphics. For example, if you need to enhance a graph for a paper or presentation, you can use the ODS Graphics Editor to customize the title, modify the axis labels, annotate particular data points, and change graph element properties such as fonts, colors, and line styles.

This section explains how to enable ODS Graphics to create editable graphs and how to invoke the ODS Graphics Editor. You can use the ODS Graphics Editor in the SAS windowing environment, provided that the LISTING destination is open and that you have first enabled ODS Graphics to create editable graphs. **NOTE:** The LISTING destination is typically open by default. There are three steps you must take to edit a graph:

1 You must first enable the creation of editable graphs in one of three ways:
- use an ODS statement to temporarily enable this feature
- use a SAS command to temporarily enable this feature
- use the SAS Registry Editor to permanently enable this feature

Creating editable graphs takes additional resources, so you might not want to permanently enable this feature.

2 You submit your SAS code and create editable graphs.

3 You invoke the ODS Graphics Editor and edit the plot.

Step 2 involves submitting SAS code in the usual way, and no special instructions are needed for creating graphs that can be edited. Steps 1 and 3 are explained in more detail in the following sections.

Enabling the Creation of Editable Graphs

Temporarily Enable Creation of Editable Graphs by Using an ODS Statement

You can enable the creation of editable graphs within a SAS session by submitting the following statement:

```
ods listing sge=on;
```

You can disable the creation of editable graphs by submitting the following statement:

```
ods listing sge=off;
```

Temporarily Enable Creation of Editable Graphs by Using a SAS Command

Alternatively, you can enable the creation of editable graphs for the duration of your SAS session by first selecting the Results window and then entering **`sgedit on`** in the command line. SAS confirms that the creation of editable graphs is enabled by displaying a message in the bottom left corner of the SAS window. The command must be entered from the Results window. If you enter it from any other window, it is ignored.

Permanently Enable Creation of Editable Graphs across SAS Sessions

You can create a default setting that enables or disables the creation of editable graphs across SAS sessions via the 'ODS Graphics Editor' setting in the SAS Registry. You can change this setting in the SAS windowing environment as follows:

1. Open the Registry Editor by entering **regedit** in the command line.
2. Select **SAS_REGISTRY ► ODS ► GUI ► RESULTS**.
3. In the **Value Data** field, click **ODS Graphics Editor** to open the Edit String Value window, and type **On** to enable the creation of editable graphs or type **Off** to disable it.
4. Click **OK**.

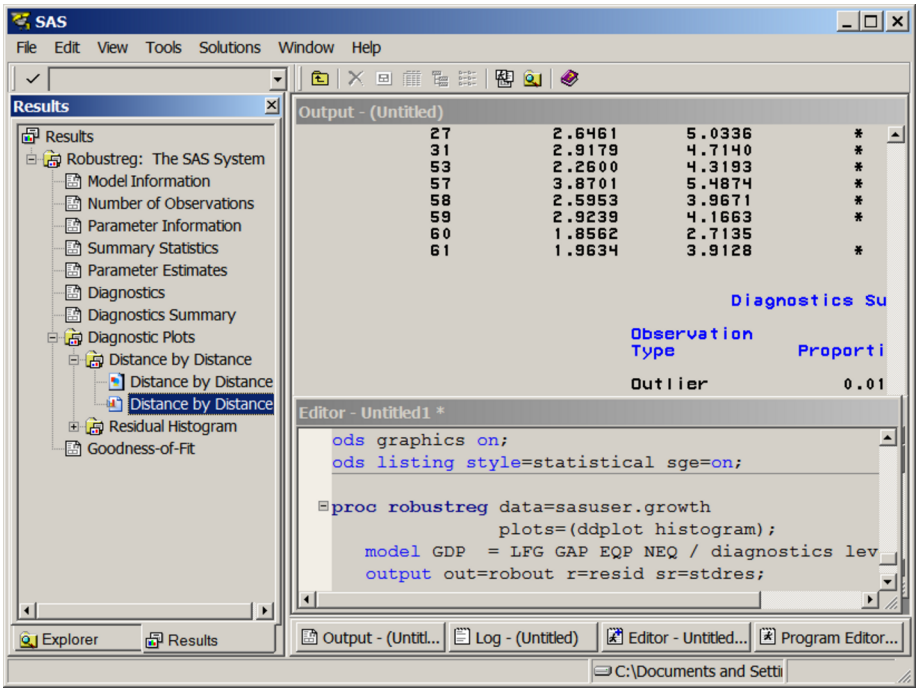
Editing a Graph with the ODS Graphics Editor

The ODS Graphics Editor is illustrated using the following example:

```
data sasuser.growth;
   input country $ GDP LFG EQP NEQ GAP @@;
   datalines;
Argentin  0.0089 0.0118 0.0214 0.2286 0.6079
Austria   0.0332 0.0014 0.0991 0.1349 0.5809

... more lines ...

Zambia   -0.0110 0.0275 0.0702 0.2012 0.8695
Zimbabwe  0.0110 0.0309 0.0843 0.1257 0.8875
;

ods graphics on;
ods listing style=statistical sge=on;

proc robustreg data=sasuser.growth
               plots=(ddplot histogram);
   model GDP  = LFG GAP EQP NEQ / diagnostics leverage;
   output out=robout r=resid sr=stdres;
run;

ods listing sge=off;
```

The DATA and the PROC ROBUSTREG steps are submitted to SAS, in this case from the SAS windowing environment, as shown in Figure 21.19. Two versions of the graph are created: one in an uneditable PNG file (for example, *DDPlot.png*) and one in an editable SGE file (for example, *DDPlot.sge*). Both are saved in the SAS current folder. You can edit the graph in one of three ways:

Figure 21.19 Results Window with Icons for Editable Plots



- In the Results window, double-click the second graph icon for the graph you want to edit (see Figure 21.19). The second icon corresponds to the SGE file, and the first icon corresponds to the PNG file. Clicking the first graph icon invokes a host-dependent graph viewer (for example, Microsoft Photo Editor on Windows), not the ODS Graphics Editor. **NOTE:** The Editor window might be hidden behind other windows in the SAS windowing environment.
- You can edit the graph by selecting it in the SAS Explorer window. You must first navigate to the SAS current folder and to the SGE files.
- You can open the graph from outside of SAS. For example, if you are running SAS under the Windows operating system, you can click on the graph's SGE file to open it with the ODS Graphics Editor.

Figure 21.20 shows the ODS Graphics Editor window for the editable diagnostic plot created by PROC ROBUSTREG. In Figure 21.21, various tools in the ODS Graphics Editor are used to modify the title and annotate a particular point. The edited plot can be saved as a PNG file or as an SGE file by selecting **File ► Save As**. After saving the plot, you can edit it again through the SAS Explorer window or by selecting **File ► Open** from the ODS Graphics Editor window. Alternatively, you can reopen the saved plot for editing without first invoking SAS. For example, if you are running SAS under the Windows operating system, you can click on the plot to open it with the ODS Graphics Editor.

The ODS Graphics Editor does not permit you to make structural changes to a graph (such as moving the positions of data points). The ODS Graphics Editor provides you with a point-and-click way to make one-time changes to a specific graph, whereas the template language (see the section "Graph Templates" on page 577) provides you with a programmatic way to make template changes that persist every time you run the procedure. For complete details about the tools available in the ODS Graphics Editor, see *Getting Started with the SAS/GRAPH Statistical Graphics Editor*.

Figure 21.20 Diagnostic Plot before Editing

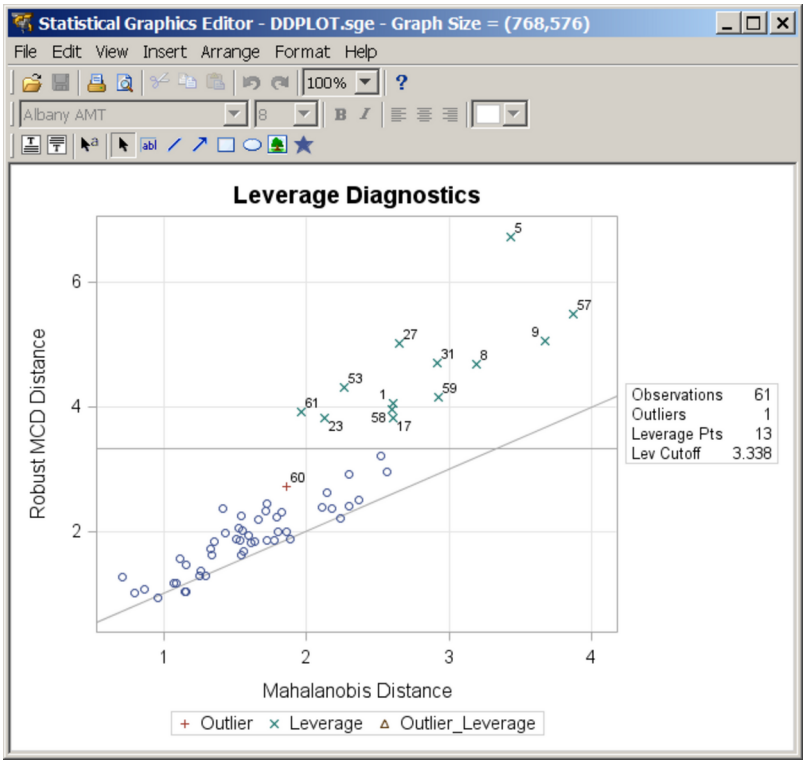
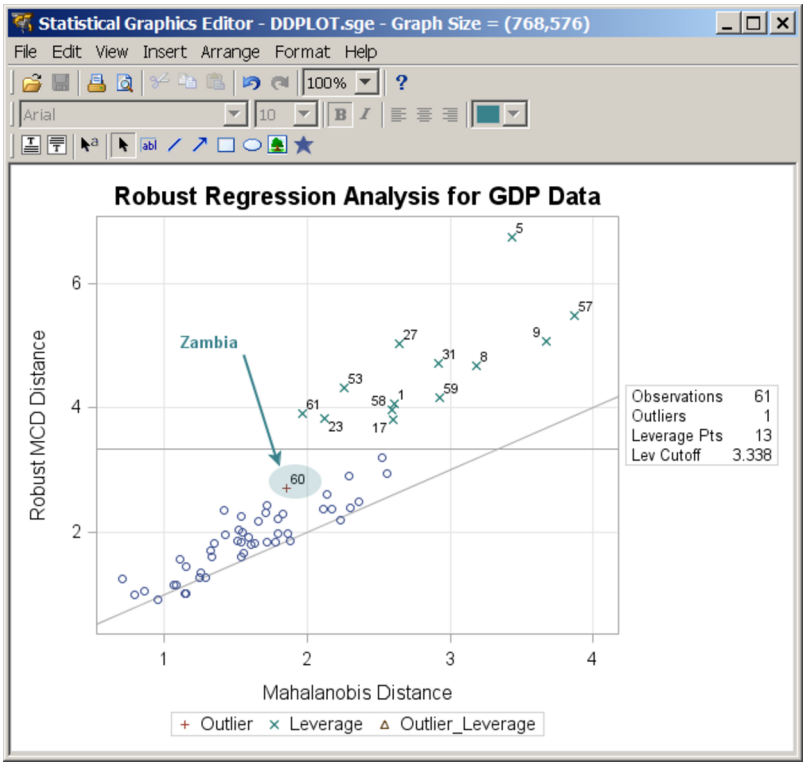


Figure 21.21 Diagnostic Plot after Editing



The Default Template Libraries and the ODS PATH

The default table, graph, and style templates that SAS provides are stored in the library SASHELP.Tmplmst. By default, if you modify a template, it is stored in the library SASUSER.Templat. By default, ODS searches SASUSER.Templat for templates, and then it searches SASHELP.Tmplmst if it does not find the requested template in SASUSER.Templat. You can see the list of template libraries by submitting the following statement:

```
ods path show;
```

Here are the results:

```
Current ODS PATH list is:

1. SASUSER.TEMPLAT(UPDATE)
2. SASHELP.TMPLMST(READ)
```

You can change the ODS PATH to add template libraries that you create or change the order that the libraries are searched. This is discussed in detail in the sections "Saving Customized Templates" on page 587, "Using Customized Templates" on page 587, and "Reverting to the Default Templates" on page 588. However, this section discusses the default template libraries that you use when you have not modified the ODS path. SAS always finds the templates it provides in SASHELP.Tmplmst if you did not store a modified template in SASUSER.Templat. Otherwise, it finds and uses your modified template in SASUSER.Templat. You can see a list of all of the templates that you have modified as follows:

```
proc template;
   list / store=sasuser.templat;
run;
```

You can delete any template that you modified (so that ODS finds the default template that SAS supplied) by specifying it on a DELETE statement, as in the following example:

```
proc template;
   delete Stat.REG.Graphics.ResidualPlot;
run;
```

ODS never deletes the template in SASHELP.Tmplmst, so you can safely run the preceding step, even if the template you specify does not exist in SASUSER.Templat. You can run the following step to delete the entire SASUSER.Templat library of customized templates so that ODS uses only the SAS supplied templates:

```
ods path sashelp.tmplmst(read);
proc datasets library=sasuser;
   delete templat(memtype=itemstor);
run;
ods path sasuser.templat(update) sashelp.tmplmst(read);
```

It is good practice to delete templates that you have customized when you are done with them, so that they are not unexpectedly used later. See the section "Reverting to the Default Templates" on page 588 for more information.

Styles

ODS styles control the overall appearance of your output. Usually, the only thing you need to do with styles is specify them in an ODS destination statement, as in the following example:

```
ods html body='b.html' style=Statistical;
```

However, you can also modify existing styles and even write your own styles. You can also specify style elements in custom templates that you write, or you can modify which style elements are used in templates supplied by SAS. This section provides an overview of styles and style elements, which are the components of a style. It also describes how to customize a style definition and how to specify a default style for your output. Only the most commonly used styles, style elements, and style changes are discussed here. For complete details about styles, see the *SAS Output Delivery System: User's Guide*.

An Overview of Styles

An ODS style definition provides formatting information for specific visual aspects of your SAS output (see the section "Style Elements and Attributes" on page 555). The appearance of tables and graphs is coordinated within a particular style. For tables, this information typically includes a list of font definitions and a list of colors. Each font definition specifies a family, size, weight, and style. Colors are associated with common areas of output, including titles, footnotes, BY groups, table headers, and table cells. For graphs, styles also control the appearance of graph elements including lines, markers, fonts and colors. ODS styles also include elements specific to statistical graphics, such as the style of fitted lines, confidence bands, and prediction limits. For more information about styles, see the *SAS Output Delivery System: User's Guide*.

You can specify a style by using the STYLE= option in an ODS destination statement such as HTML, PDF, RTF, or PRINTER. You can also specify a style in the LISTING destination; however, it affects graphs but not tables. Each style produces output with the same content, but with a different visual appearance. For example, the following statement requests output produced with the JOURNAL style:

```
ods html style=Journal;
```

You can use any SAS style or any style that you define yourself. The following statements list the names of all of the styles and then display four of them:

```
proc template;
   list styles;
   source Styles.Default;
   source Styles.Statistical;
   source Styles.Journal;
   source Styles.RTF;
run;
```

The results of this step (not shown) are a list of over fifty styles in the SAS listing and four style definitions in the SAS log. The style definitions are often hundreds of lines long. See the section "Style Definitions and Colors" on page 557 for more information about style definitions. While you can use any style, only seven styles are typically used with ODS Graphics. They are described in Table 21.2.

Table 21.2 Styles

Style	Default in	Description
DEFAULT	HTML	A color style whose dominant colors are blue, gray, and white, with bold sans-serif fonts. See Figure 21.13.
STATISTICAL	SAS/STAT documentation	A color style whose dominant colors are blue, creamy gray, and white, with sans-serif fonts. See Figure 21.14.
LISTING	LISTING	A color style, similar to DEFAULT, but with a white background. See Figure 21.12.
JOURNAL		A black-and-white style with filled areas, and with sans-serif fonts. See Output 21.3.1.
JOURNAL2		A black-and-white style, similar to JOURNAL, but with empty areas. See Output 21.3.2.
RTF	RTF	A color style whose dominant colors are blue, white, and black, with Times Roman fonts. See Figure 21.11.
ANALYSIS		A color style, similar to STATISTICAL, whose dominant color is tan. See Figure 21.8.

Each ODS destination has its own default style, as shown in Table 21.2. Most output in SAS/STAT documentation uses the STATISTICAL style. However, throughout this chapter, you can see examples of other styles. For more information about styles, see the *SAS Output Delivery System: User's Guide*.

Style Elements and Attributes

An ODS style definition is composed of a set of *style elements*. A style element is a collection of *style attributes* that applies to a particular feature or aspect of the output. A value is specified for each attribute in a style definition. For example, **`GraphFit`** is the style element used for fit lines, and its attributes include: **`LineThickness`**, **`LineStyle`**, **`MarkerSize`**, **`MarkerSymbol`**, **`ContrastColor`**, and **`Color`**.

In general, style definitions control the overall appearance of ODS tables and graphs. For tables, style definitions specify features such as background color, table borders, and color scheme; and they specify the fonts, sizes, and color for the text and values in a table and its headers. For graphs, style definitions specify the following features:

- background color
- graph dimensions (height and width)
- borders
- line styles for axes and grid lines
- fonts, sizes, and colors for titles, footnotes, axis labels, axis values, and data labels. See the section "Modifying Graph Fonts in Styles" on page 570 for an illustration.
- marker symbols, colors, and sizes for data points and outliers
- line styles for needles
- line and curve styles for fitted models and predicted values. See "Modifying Other Graph Elements in Styles" on page 573 for an illustration.
- line and curve styles for confidence and prediction limits
- fill colors for histogram bars, confidence bands, and confidence ellipses
- colors for box plot features
- colors for surfaces
- color ramps for contour plots

SAS supplies a template for each plot that is created by statistical procedures. A graph template is a program that specifies the layout and details of a graph. See the section "Graph Templates" on page 577 for more information about templates. Some template options are specified with a style reference of the form **`style-element`**, or occasionally **`style-element:attribute`**. For example, the symbol, color, and size of markers for basic scatter plots are specified in a template SCATTERPLOT statement as follows:

```
scatterplot x=x y=y / markerattrs=GraphDataDefault;
```

The preceding statement specifies that the appearance for markers is controlled by the **`GraphDataDefault`** element. Consistent use of this element guarantees a common appearance of markers across all scatter plots, based on the style definition that you are using.

In general, ODS Graphics features are determined by style element attributes unless they are overridden by a statement or option in the graph template. For example, suppose that a classification variable is specified with the GROUP= option in a SCATTERPLOT template statement as follows:

```
scatterplot x=X y=Y / group=GroupVar;
```

Then the colors for markers that correspond to the classification levels are assigned by using the style element attributes **`GraphData1:ContrastColor`** through **`GraphData12:ContrastColor`**.

Style definitions are created and modified with PROC TEMPLATE. For more information, see the *SAS Output Delivery System: User's Guide*. You need to understand the relationships between style elements and graph features if you want to create your own style definition or modify a style definition. This is explained in the following sections.

Style Definitions and Colors

The default style definitions that SAS provides are stored in the *Styles* directory of SASHELP.Tmplmst. You can display, edit, and save style definitions by using the same methods available for modifying template definitions, as explained in the section "The Default Template Libraries and the ODS PATH" on page 553 and the series of sections beginning with "Displaying Templates" on page 583. In particular, you can display a style definition by using one of these methods:

- From the Templates window in the SAS windowing environment, expand the SASHELP.Tmplmst node under **Templates**, and then select **Styles** to display the contents of this folder. To open the Templates window, type **odst** in the command line.
- Use the SOURCE statement in PROC TEMPLATE.

For example, the following statements display the DEFAULT style definition in the SAS log:

```
proc template;
   source Styles.Default;
run;
```

The following are some of the results:

```
define style Styles.Default;
   . . .
   class GraphColors
      "Abstract colors used in graph styles" /
      . . .
      'gconramp3cend' = cxFF0000
      'gconramp3cneutral' = cxFF00FF
      'gconramp3cstart' = cx0000FF
      . . .
      'gdata12' = cxDDD17E
      'gdata11' = cxB7AEF1
      'gdata10' = cx87C873
      'gdata9' = cxCF974B
      'gdata8' = cxCD7BA1
      'gdata6' = cxBABC5C
      'gdata7' = cx94BDE1
      'gdata4' = cxA9865B
      'gdata5' = cxB689CD
      'gdata3' = cx66A5A0
      'gdata2' = cxDE7E6F
      'gdata1' = cx7C95CA;
   . . .
```

The first part of this list shows that the shading for certain filled plots, such as some contour plots goes, from blue (`'gconramp3cstart' = cx0000FF`) to magenta (`'gconramp3cneutral' = cxFF00FF`) to red (`'gconramp3cend' = cxFF0000`). All colors are specified in values of the form CX*rrggbb*, where the last six characters specify RGB (red, green, blue) values on the hexadecimal scale of 00 to FF (or 0 to 255 base 10). The second part of the list (`'gdata1' = cx7C95CA`)

shows that the dominant component of the **GraphData1** color is blue because the blue component of the color (CA, which corresponds to 202 base 10) is greater than both the green component (95, which corresponds to 149 base 10) and the red component (7C, which corresponds to 124 base 10).

You can change any part of the style and then submit the style back into SAS, after first submitting a PROC TEMPLATE statement. See the sections "Saving Customized Templates" on page 587, "Using Customized Templates" on page 587, and "Reverting to the Default Templates" on page 588 for more information about modifying, using, and restoring templates. The principles discussed in those sections apply to all templates—table, style, and graph.

Some Common Style Elements

The DEFAULT style is the parent for the styles used for statistical graphics work. You can see all of the elements of the DEFAULT style by running the following step:

```
proc template;
   source styles.default;
run;
```

The source listing of the definition of the DEFAULT style is hundreds of lines long. If you run PROC TEMPLATE with the SOURCE statement for most other styles, you see **parent = styles.default**, and you do not see all of the elements in the style unless you also run the preceding step.

Only a few of the style elements are referenced in the templates that SAS provides for statistical procedures. The most commonly used style elements, along with the defaults for the noncolor attributes of the DEFAULT style, are shown next (**Color** applies to filled areas, and **ContrastColor** applies to markers and lines):

Graph	graph size, outer border appearance, and background color **Padding = 0** **BackgroundColor**
GraphConfidence	graph size, outer border appearance, and background color **LineThickness = 1px** **LineStyle = 1** **MarkerSize = 7px** **MarkerSymbol = "triangle"** **ContrastColor** **Color**
GraphData1	attributes related to first grouped data items **MarkerSymbol = "circle"** **LineStyle = 1** **ContrastColor** **Color**

GraphData2	attributes related to second grouped data items
	MarkerSymbol = "plus"
	LineStyle = 4
	ContrastColor
	Color
GraphData3	attributes related to third grouped data items
	MarkerSymbol = "X"
	LineStyle = 8
	ContrastColor
	Color
GraphData4	attributes related to fourth grouped data items
	MarkerSymbol = "triangle"
	LineStyle = 5
	ContrastColor
	Color
GraphData*n*	attributes related to *n*th grouped data items
	MarkerSymbol
	LineStyle
	ContrastColor
	Color
GraphDataDefault	attributes related to non-grouped data items
	EndColor
	NeutralColor
	StartColor
	MarkerSize = 7px
	MarkerSymbol = "circle"
	LineThickness = 1px
	LineStyle = 1
	ContrastColor
	Color
GraphFit	primary fit line, such as a normal density curve
	LineThickness = 2px
	LineStyle = 1
	MarkerSize = 7px
	MarkerSymbol = "circle"
	ContrastColor
	Color
GraphFit2	secondary fit line, such as a kernel density curve
	LineThickness = 2px
	LineStyle = 4
	MarkerSize = 7px
	MarkerSymbol = "X"
	ContrastColor
	Color

GraphGridLines	horizontal and vertical grid lines drawn at major tick marks **Displayopts = "auto"** **LineThickness = 1px** **LineStyle = 1** **ContrastColor** **Color**
GraphOutlier	outlier data for the graph **LineThickness = 2px** **LineStyle = 42** **MarkerSize = 7px** **MarkerSymbol = "circle"** **ContrastColor** **Color**
GraphPredictionLimits	fills for prediction limits **LineThickness = 1px** **LineStyle = 2** **MarkerSize = 7px** **MarkerSymbol = "chain"** **ContrastColor** **Color**
GraphReference	horizontal and vertical reference lines and drop lines **LineThickness = 1px** **LineStyle = 1** **ContrastColor** **Color**
GraphDataText	text font and color for point and line labels **Font = GraphFonts('GraphDataFont')** (where **'GraphDataFont' = ("<sans-serif>, <MTsans-serif>",7pt))** **Color**
GraphValueText	text font and color for axis tick values and legend values **Font = GraphFonts('GraphValueFont')** (where **'GraphValueFont' = ("<sans-serif>, <MTsans-serif>",9pt)**) **Color**
GraphLabelText	text font and color for axis labels and legend title **Font = GraphFonts('GraphLabelFont')** (where **'GraphLabelFont' = ("<sans-serif>, <MTsans-serif>",10pt,bold)**) **Color**
GraphFootnoteText	text font and color for footnote(s) **Font = GraphFonts('GraphFootnoteFont')** (where **'GraphFootnoteFont' = ("<sans-serif>, <MTsans-serif>",10pt)**) **Color**

GraphTitleText	text font and color for title(s) **Font = GraphFonts('GraphTitleFont')** (where **'GraphTitleFont' = ("<sans-serif>, <MTsans-serif>",11pt,bold)**) **Color**
GraphWalls	vertical wall(s) bounded by axes **LineThickness = 1px** **LineStyle = 1** **FrameBorder = on** **ContrastColor** **BackgroundColor** **Color**

You refer to these elements in graph templates as **style-element** or as **style-element:attribute** (for example **GraphDataDefault:ContrastColor**). The default values are not shown for the color attributes since they are typically defined indirectly. For example, **Graph:BackgroundColor** (the color that fills the box outside the graph) is defined elsewhere in the style as **colors('docbg')**. The style also defines: **'docbg' = color_list('bgA')** and **'bgA' = cxE0E0E0**. This shows that the background is a shade of gray that is much closer to white (CXFFFFFF) than to black (CX000000). You can see the background color in Figure 21.22. This shade of gray might seem darker (closer to CX000000) than you might expect based on just the RGB values. Your perception of a color change is not a linear function of the change in RGB values.

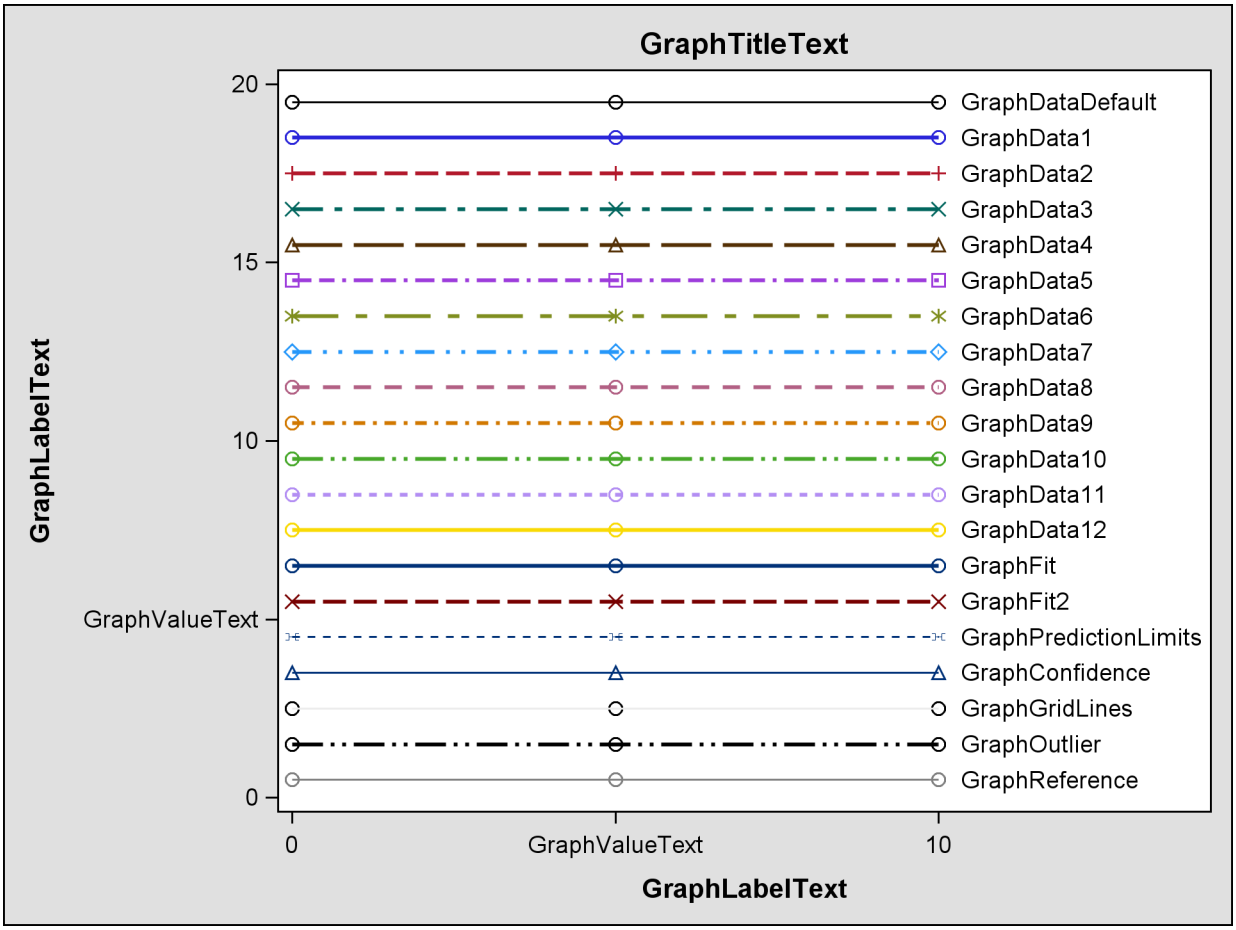
You can use the following program to see the color and other attributes for a number of style elements:

```
proc format; value vf 5 = 'GraphValueText'; run;

data x;
   array y[20] y0 - y19;
   do x = 1 to 20; y[x] = x - 0.5; end;
   do x = 0 to 10 by 5; output; end;
   label y0 = 'GraphLabelText' x = 'GraphLabelText';
   format x y0 vf.;
run;

%macro d;
   %do i = 1 %to 12;
      reg y=y%eval(19-&i) x=x / lineattrs=GraphData&i markerattrs=GraphData&i
                                curvelabel="  GraphData&i" curvelabelpos=max;
   %end;
%mend;

%macro l(i, l);
   reg y=y&i x=x / lineattrs=&l markerattrs=&l curvelabel="  &l"
                   curvelabelpos=max;
%mend;

ods listing style=default;
```

```
proc sgplot noautolegend;
   title 'GraphTitleText';
   %d
   %l(19, GraphDataDefault)
   %l( 6, GraphFit)
   %l( 5, GraphFit2)
   %l( 4, GraphPredictionLimits)
   %l( 3, GraphConfidence)
   %l( 2, GraphGridLines)
   %l( 1, GraphOutlier)
   %l( 0, GraphReference)
   xaxis values=(0 5 10);
run;
```

The results in Figure 21.22 display the attributes for a number of the elements of the DEFAULT style.

Figure 21.22 Attributes of Style Elements in the DEFAULT Style

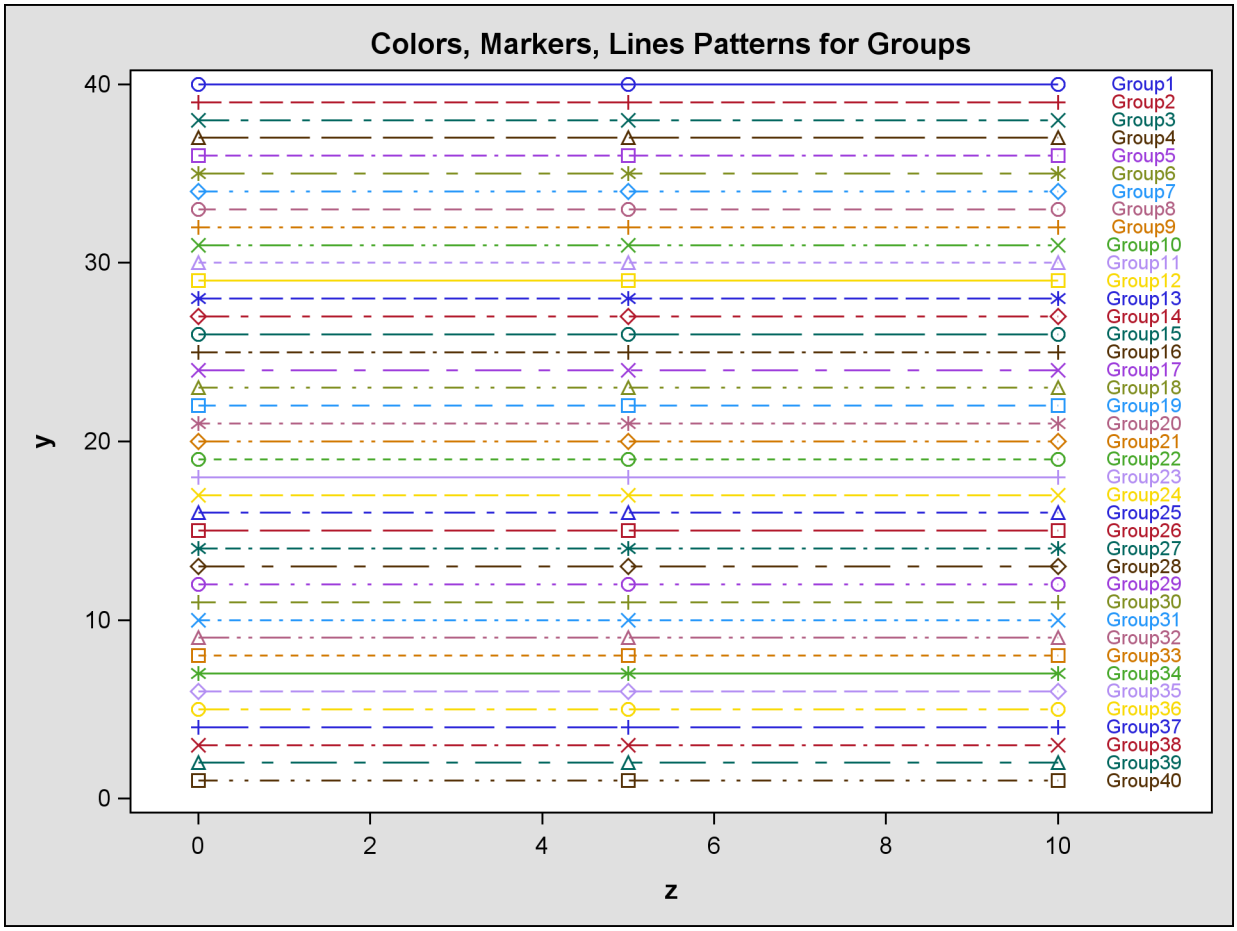


When there is a group or classification variable, the colors, markers, and lines that distinguish the groups are derived from the **GraphData***n* elements that are defined with the style. In the DEFAULT style, these are elements **GraphData1** through **GraphData12**. There can be any number of groups even though only 12 **GraphData***n* style elements are defined in the DEFAULT style. The following

steps create a data set with 40 groups, display one line per group, and produce Figure 21.23:

```
data x;
   do y = 40 to 1 by -1;
      group = 'Group' || put(41 - y, 2. -L);
      do x = 0 to 10 by 5;
         if x = 10 then do; z = 11; l = group; end;
         else           do; z = .;  l = ' ';   end;
         output;
      end;
   end;
run;

proc sgplot data=x;
   title 'Colors, Markers, Lines Patterns for Groups';
   series  y=y x=x / group=group markers;
   scatter y=y x=z / group=group markerchar=l;
run;
```

Figure 21.23 Markers and Lines Cycle with Different Periods in Groups



The colors, markers, and line patterns in Figure 21.23 repeat in cycles. The **GraphData1** – **GraphData8** lines in Figure 21.22 exactly match the **Group1** – **Group8** lines in Figure 21.23. After that, there are differences due to the cyclic construction of the grouped style definition. This is explained next.

The DEFAULT style defines a marker symbol only in **GraphData1** through **GraphData7**. The seven markers are: circle, plus, X, triangle, square, asterisk, and diamond. With the explicit style reference in Figure 21.22, the actual symbol, when no symbol is specified, is the circle. This is what you see for **GraphData8** through **GraphData12**. With the group variable in Figure 21.23, the symbols repeat in cycles. Hence, **Group1**, **Group8**, **Group15**, and so on, are all circles. Similarly, **Group2**, **Group9**, **Group16**, and so on, are all pluses. The DEFAULT style defines 11 different line styles for **GraphData1** through **GraphData11**: 1, 4, 8, 5, 14, 26, 15, 20, 41, 42, and 2. Hence, **Group1**, **Group12**, **Group23**, and so on, all have the same line style, which is a solid line. Similarly, **Group2**, **Group13**, **Group24**, and so on, all have line style 4. There are twelve different colors, so **Group1**, **Group13**, **Group25**, and so on, all have the same colors. Overall, there are $12 \times 11 \times 7 = 924$ color/line/marker combinations that appear before any combination repeats. You can use the **ModStyle** SAS autocall macro (see the section "Creating an All-Color Style by Using the ModStyle Macro" on page 564) to conveniently change these style attributes.

Creating an All-Color Style by Using the ModStyle Macro

Many styles are designed to make color plots where lines, functions, and groups of observations can be distinguished even when the plot is sent to a black-and-white printer. Hence, lines differ not only in color but also in pattern. Similarly, markers differ in color and symbol.

You can change that behavior with the **ModStyle** autocall macro. It creates a new style by modifying a parent style and reordering the colors, line patterns, and marker symbols in the **GraphData***n* style elements (see section "Some Common Style Elements" on page 558). By default, the macro creates a new style that distinguishes lines and groups only by color. The macro is documented in its header.

The following example illustrates the default use of the macro and is taken from the section "Fitting a Curve through a Scatter Plot" on page 7109 of Chapter 90, "The TRANSREG Procedure." The data come from an experiment in which nitrogen oxide emissions from a single cylinder engine are measured for various combinations of fuel and equivalence ratio. The following statements create the SAS data set:

```
data Gas;
   input Fuel :$8. EqRatio NOx @@;
   label EqRatio = 'Equivalence Ratio'
         NOx     = 'Nitrogen Oxide Emissions';
   datalines;
Ethanol  0.907 3.741 Ethanol  0.761 2.295 Ethanol  1.108 1.498

... more lines ...

;
```

The following statements fit separate curves for each group and produce Figure 21.24 and Figure 21.25:

```
ods listing style=statistical;
ods graphics on;

proc transreg data=Gas ss2 plots=transformation lprefix=0;
   model identity(nox) = class(Fuel / zero=none) * pbspline(EqRatio);
run;

%modstyle(parent=statistical, name=StatColor)
ods listing style=StatColor;

proc transreg data=Gas ss2 plots=transformation lprefix=0;
   model identity(nox) = class(Fuel / zero=none) * pbspline(EqRatio);
run;
```

The first PROC TRANSREG step uses the STATISTICAL style to create the fit plot in Figure 21.24, which uses different colors, line patterns, and markers for each group. Then the macro creates a new style, called STATCOLOR, that inherits its characteristics from the STATISTICAL style. Only the attributes of the lines and markers are changed. In Figure 21.25, which is created with the modified style, the groups are differentiated only by color. This is the easiest and most common way for you to use this macro. However, you can use it to perform other style modifications as illustrated in the section "Changing the Default Markers and Lines" on page 567. See the macro header for more information.

You do not have to include autocall macros (for example, with a **%include** statement). You can call them directly once they are properly installed. If your site has installed the autocall libraries supplied by SAS and uses the standard configuration of SAS supplied software, you need to ensure that the SAS system option MAUTOSOURCE is in effect to begin using the autocall macros. For more information about autocall libraries, see the *SAS Macro Language: Reference*. For details about installing autocall macros, consult your host documentation.

Figure 21.24 Fit Plot with the STATISTICAL Style

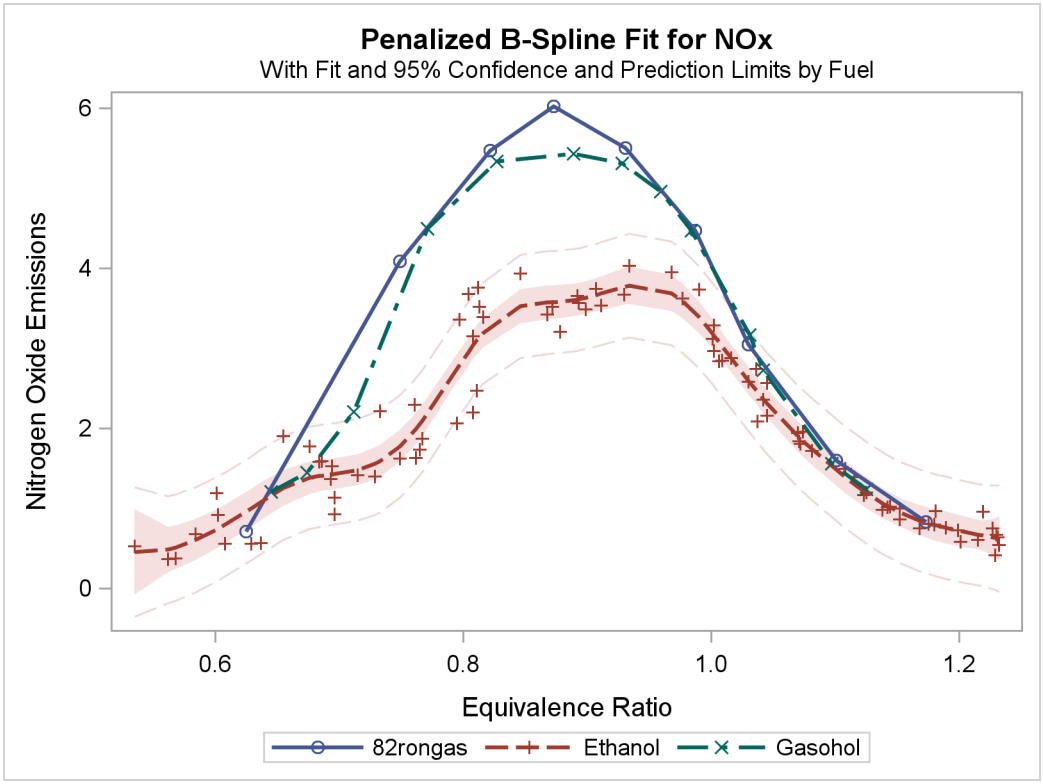
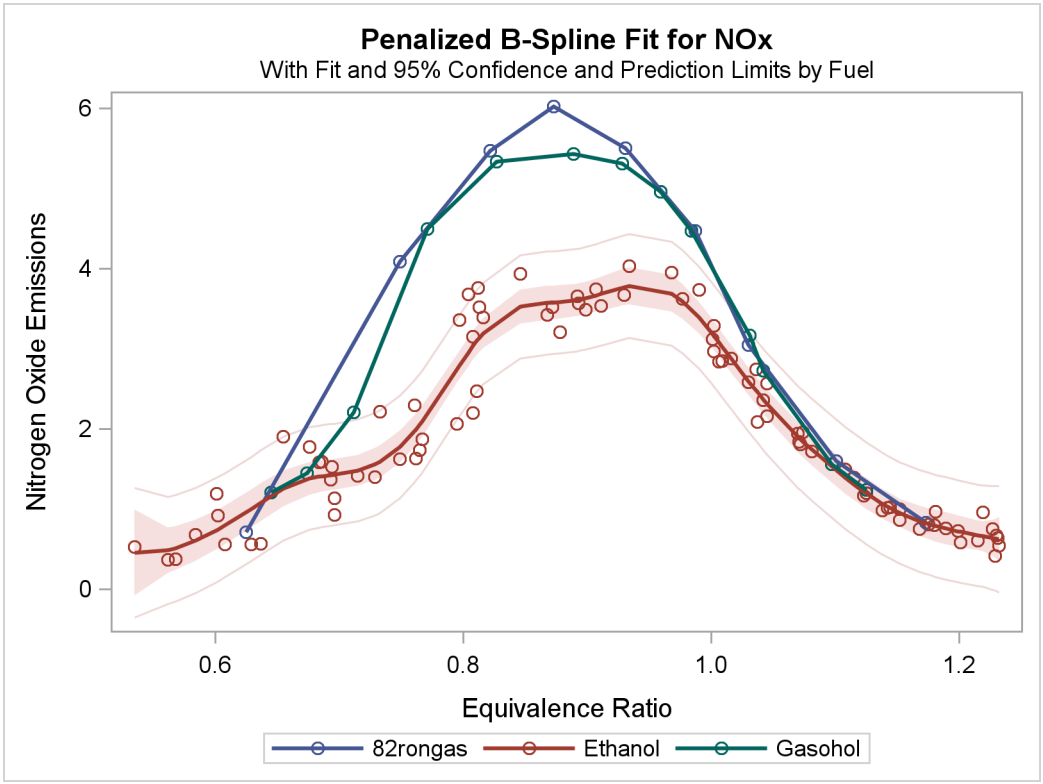


Figure 21.25 Fit Plot with the Modified Style



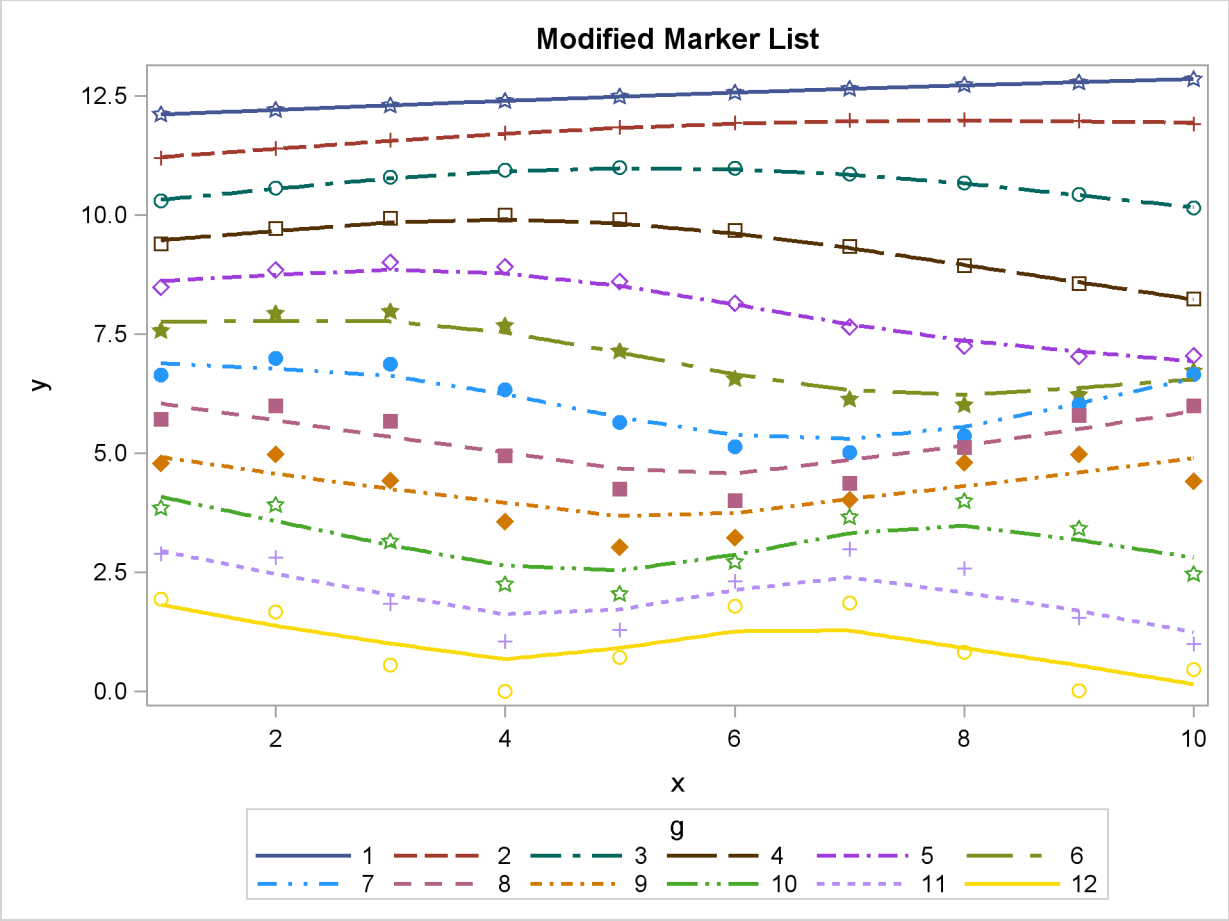
Changing the Default Markers and Lines

The preceding section shows how to use the **ModStyle** autocall macro to create an all-color style. You can also use the **ModStyle** macro to change markers and line styles. This example creates a new style called MARKSTYLE that inherits from the STATISTICAL style but uses a different set of markers. The following statements create artificial data, change the marker list, and display the results:

```
data x;
   do g = 1 to 12;
      do x = 1 to 10;
         y = 13 - g + sin(x * 0.1 * g);
         output;
      end;
   end;
run;

%modstyle(name=markstyle, parent=statistical, type=CLM,
          markers=star plus circle square diamond starfilled
                  circlefilled squarefilled diamondfilled)

ods listing style=markstyle;

proc sgplot;
   title 'Modified Marker List';
   loess y=y x=x / group=g;
run;
```

The NAME= option specifies the new style name, and the PARENT= option specifies the parent style. The TYPE= option controls the method of cycling through colors, lines, and markers. The default, TYPE=LMbyC, fixes (holds constant) the line styles and markers, while cycling through the color list. This is illustrated in the section "Creating an All-Color Style by Using the ModStyle Macro" on page 564. This example uses TYPE=CLM to cycle through colors, line styles, and markers (holding none of them fixed). Other TYPE= values are described in the macro header. The values specified with the TYPE= option are case sensitive ('by' is lower case and the 'L', 'C', and 'M' are upper case). The new marker list is specified with the MARKERS= option. The results are displayed in Figure 21.26. The marker list is reused in the tenth and subsequent groups since only nine markers are defined.

Figure 21.26 A Modified Style with a New List of Markers

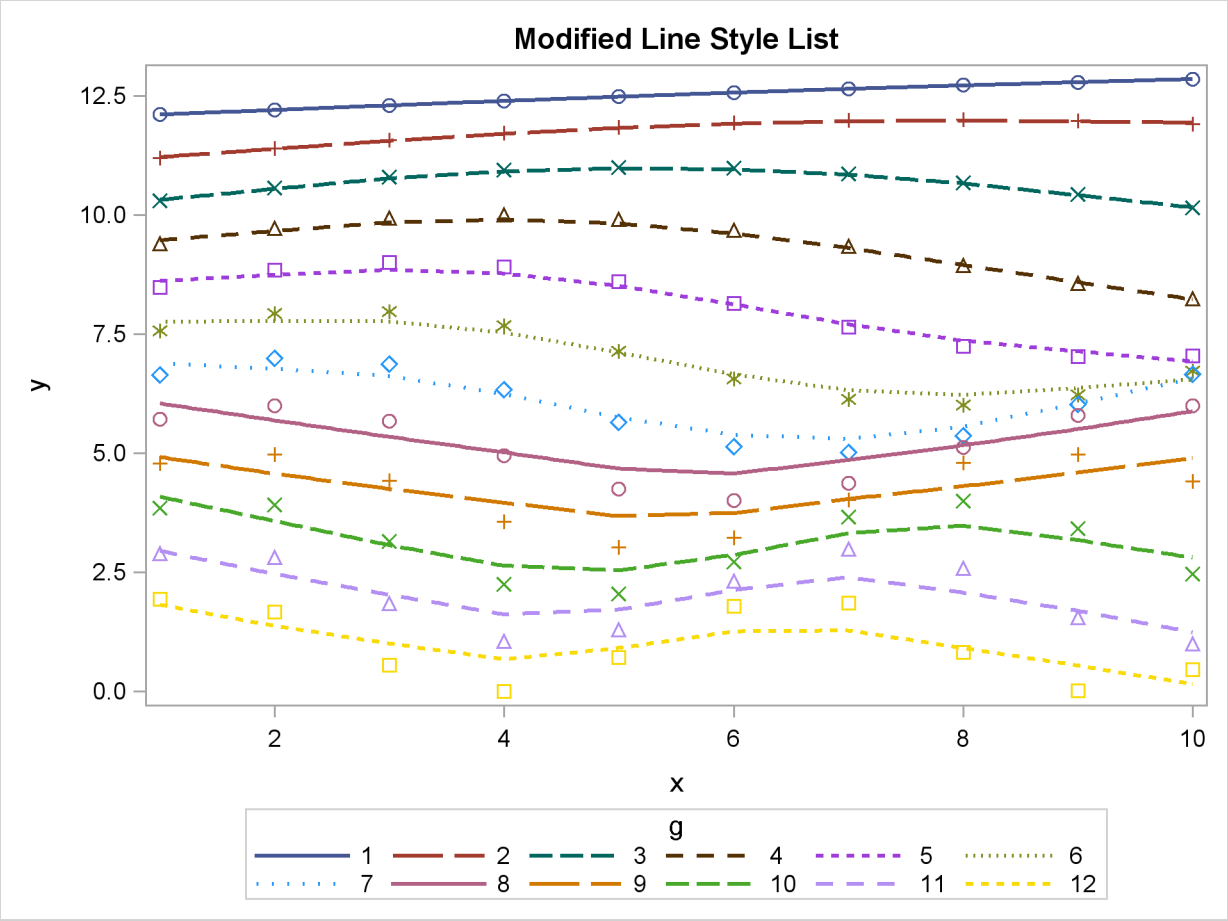


The following statements create a new style called LINESTYLE that inherits from the STATISTICAL style and changes the line list:

```
%modstyle(name=linestyle, parent=statistical, type=CLM,
          linestyles=Solid LongDash MediumDash Dash ShortDash Dot ThinDot)

ods listing style=linestyle;

proc sgplot;
   title 'Modified Line Style List';
   loess y=y x=x / group=g;
run;
```

The new line list is specified with the LINESTYLES= option. The results are displayed in Figure 21.27. In this example, each of the first seven groups uses a dash pattern that is shorter than the previous group. The line list is reused in the eighth and subsequent groups since only seven line patterns are defined.

Figure 21.27 Modified Style with a New List of Line Styles



You can learn more about style modification by examining the new styles, as in the following example:

```
proc template;
   source styles.markstyle;
   source styles.linestyle;
run;
```

The results show the definitions of **GraphData1** through **GraphData32** that the macro created. An abridged listing of the results follows:

```
define style Styles.Markstyle;
   parent = Styles.statistical;
   . . .
   style GraphData1 /
      markersymbol = "star"
      linestyle = 1
      contrastcolor = ColorStyles('c1')
      color = FillStyles('f1');
```

```
   . . .
   style GraphData32 /
      markersymbol = "diamond"
      linestyle = 42
      contrastcolor = ColorStyles('c8')
      color = FillStyles('f8');
end;

define style Styles.Linestyle;
   parent = Styles.statistical;
   . . .
   style GraphData1 /
      markersymbol = "circle"
      linestyle = 1
      contrastcolor = ColorStyles('c1')
      color = FillStyles('f1');
   . . .
   style GraphData32 /
      markersymbol = "triangle"
      linestyle = 20
      contrastcolor = ColorStyles('c8')
      color = FillStyles('f8');
end;
```

You can use the NUMBEROFGROUPS= option in the **ModStyle** macro to control the number of **GraphData***n* style elements created in the new style.

Modifying Graph Fonts in Styles

You can modify an ODS style to customize the general appearance of plots produced with ODS Graphics, just as you can modify a style to customize the general appearance of ODS tables. This section shows you how to customize fonts used in graphs. The following statements show the STATISTICAL style:

```
proc template;
   source Styles.Statistical;
run;
```

The portion of the style that controls fonts is shown next:

```
style GraphFonts /
   'GraphDataFont' = ("<sans-serif>, <MTsans-serif>",7pt)
   'GraphUnicodeFont' = ("<MTsans-serif-unicode>",9pt)
   'GraphValueFont' = ("<sans-serif>, <MTsans-serif>",9pt)
   'GraphLabelFont' = ("<sans-serif>, <MTsans-serif>",10pt)
   'GraphFootnoteFont' = ("<sans-serif>, <MTsans-serif>",10pt,italic)
   'GraphTitleFont' = ("<sans-serif>, <MTsans-serif>",11pt,bold)
   'GraphAnnoFont' = ("<sans-serif>, <MTsans-serif>",10pt);
```

The following fonts are the ones typically used for the text in most graphs:

- **GraphDataFont** is the smallest font. It is used for text that needs to be small (labels for points in scatter plots, labels for contours, and so on)
- **GraphValueFont** is the next largest font. It is used for axis value (tick marks) labels and legend entry labels.
- **GraphLabelFont** is the next largest font. It is used for axis labels and legend titles.
- **GraphFootnoteFont** is the next largest font. It is used for all footnotes.
- **GraphTitleFont** is the largest font. It is used for all titles.

The following statements define a style named NEWSTYLE that replaces the graph fonts in the DEFAULT style with italic Times New Roman fonts, which are available with the Windows operating system:

```
proc template;
   define style Styles.NewStyle;
      parent=Styles.Statistical;
      replace GraphFonts  /
         'GraphDataFont'     = ("<MTserif>, Times New Roman",7pt)
         'GraphUnicodeFont'  = ("<MTserif>, Times New Roman",9pt)
         'GraphValueFont'    = ("<MTserif>, Times New Roman",9pt)
         'GraphLabelFont'    = ("<MTserif>, Times New Roman",10pt)
         'GraphFootnoteFont' = ("<MTserif>, Times New Roman",10pt)
         'GraphTitleFont'    = ("<MTserif>, Times New Roman",11pt)
         'GraphAnnoFont'     = ("<MTserif>, Times New Roman",10pt);
   end;
run;
```

For more information about the DEFINE, PARENT, and REPLACE statements, see the *SAS/GRAPH: Graph Template Language Reference*.

The "Getting Started" section of Chapter 74, "The ROBUSTREG Procedure," creates the following data set to illustrate the use of the PROC ROBUSTREG for robust regression:

```
data stack;
   input  x1 x2 x3 y @@;
   datalines;
80  27  89  42    80  27  88  37    75  25  90  37

... more lines ...

;
```

The following statements create a Q-Q plot that uses the STATISTICAL style (see Figure 21.28) and the NEWSTYLE style (see Figure 21.29):

```
ods listing style=Statistical;
ods graphics on;

proc robustreg data=stack plots=qqplot;
   ods select QQPlot;
   model y = x1 x2 x3;
run;
ods listing close;
ods listing style=NewStyle;

proc robustreg data=stack plots=qqplot;
   ods select QQPlot;
   model y = x1 x2 x3;
run;
```

Figure 21.28 Q-Q Plot That Uses the STATISTICAL Style

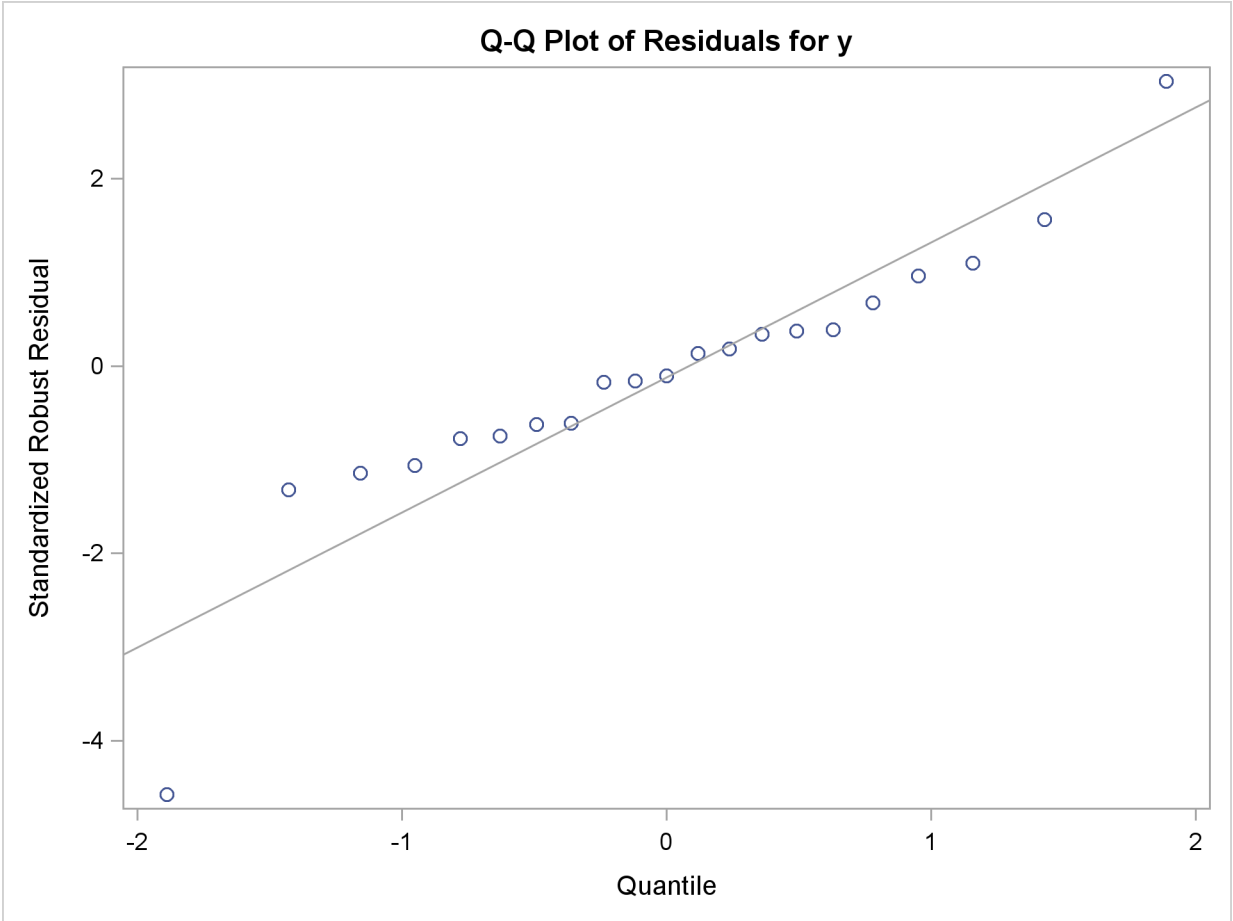
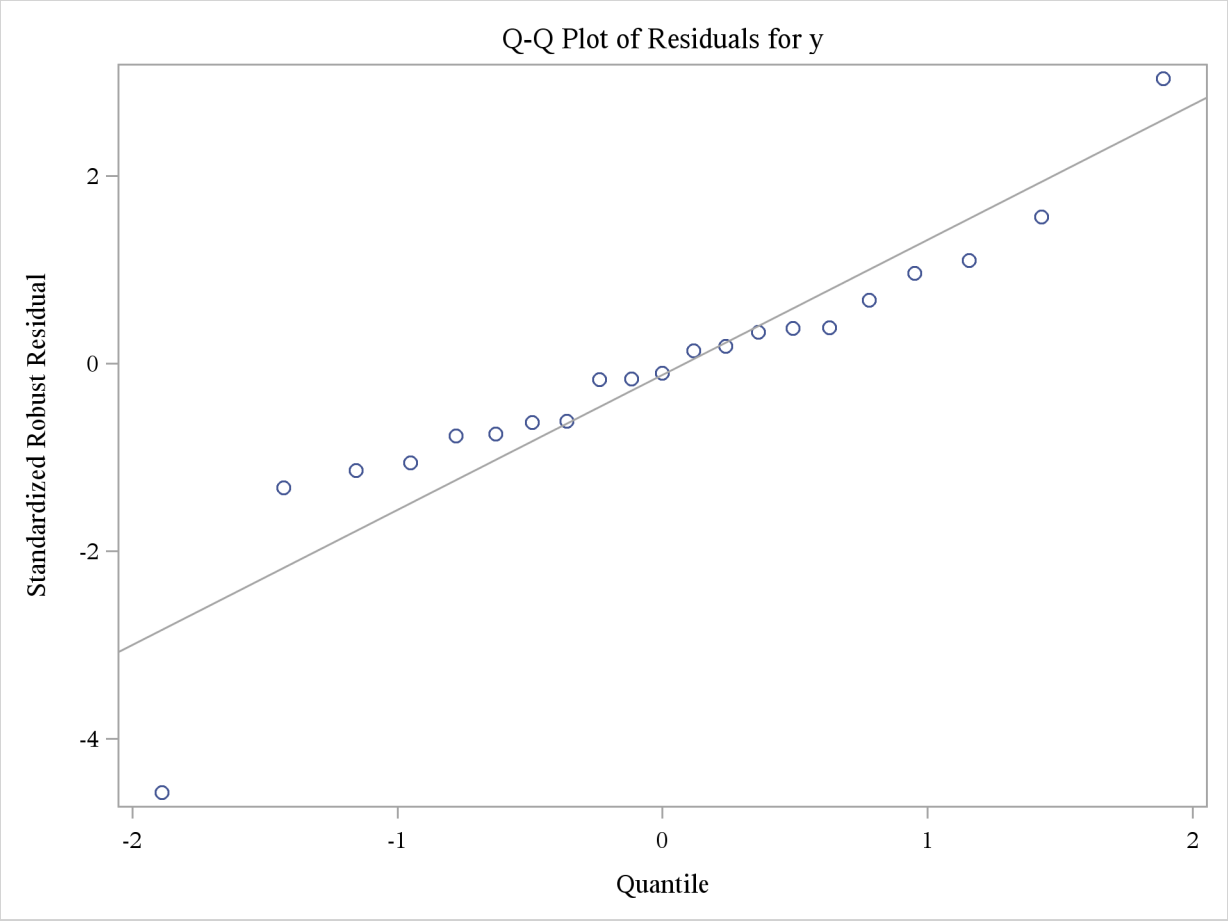


Figure 21.29 Q-Q Plot That Uses the NEWSTYLE Style



Although this example illustrates the use of a style with graphical output from a particular procedure, a style is applied to *all* of your output (graphs and tables) in the destination for which you specify the style. See the section "Changing the Default Style" on page 575 for information about specifying a default style for all your output.

Modifying Other Graph Elements in Styles

This section illustrates how to modify other style elements for graphics, specifically the style element **`GraphReference`**, which controls the attributes of reference lines. You can run the following statements to learn more about the **`GraphReference`** style element:

```
proc template;
   source styles.statistical;
run;
```

The following are the first two lines of the source listing:

```
define style Styles.Statistical;
   parent = styles.default;
```

There is no mention of **GraphReference** in the complete listing of the source because **GraphReference** is inherited from the parent style. Most styles inherit many of their attributes from other styles. To find out more, you must list the parent style, as in the following example:

```
proc template;
   source styles.default;
run;
```

Styles that you typically use with ODS Graphics inherit most of their attributes from only one style, the DEFAULT style. A few of the other styles inherit from several parents. You might have to repeat this process multiple times to find the first parent. However, with the STATISTICAL style, you only need to run this one extra step, and the results contain the following:

```
class GraphReference /
   linethickness = 1px
   linestyle = 1
   contrastcolor = GraphColors('greferencelines');
```

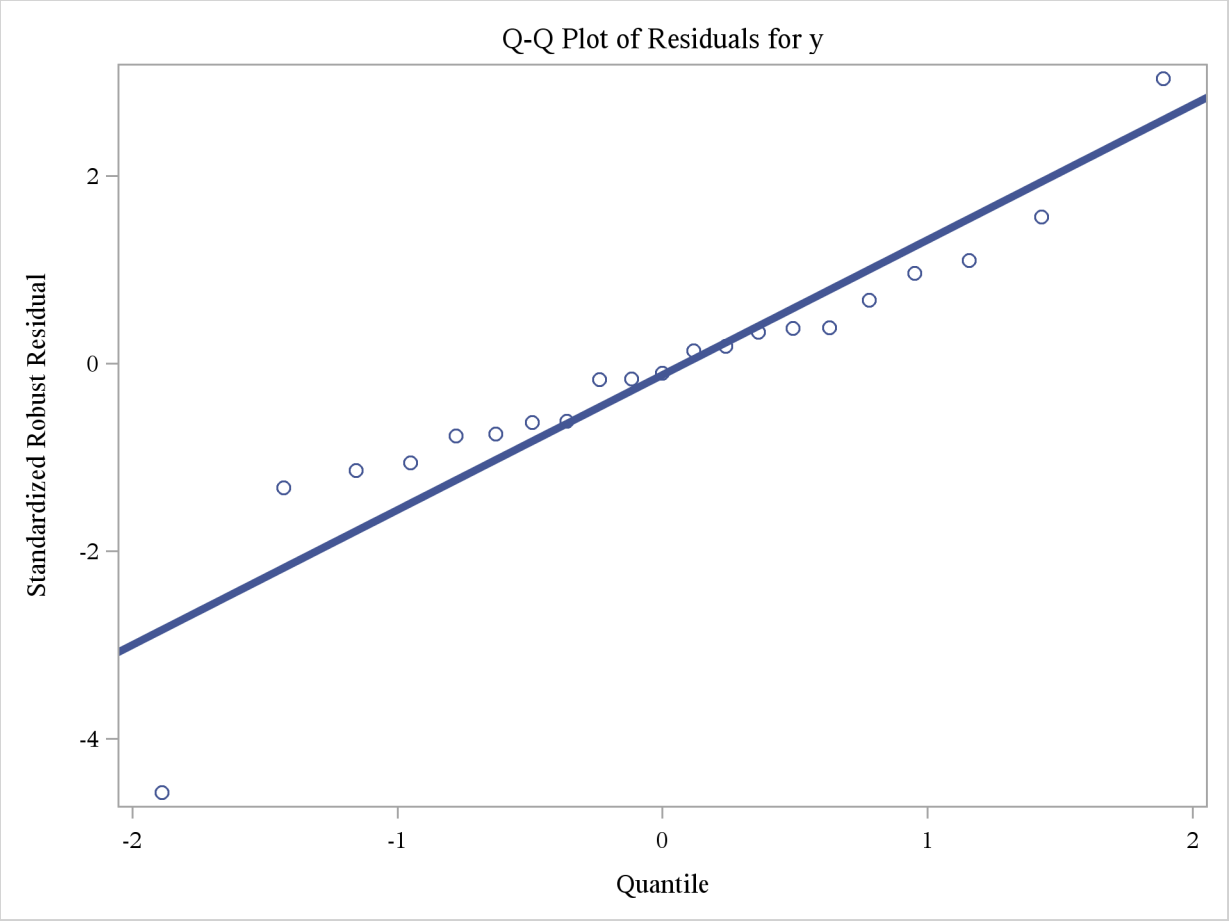
To specify a line thickness of 4 pixels for all reference lines, add the following statement to the definition of the NEWSTYLE style in the section "Modifying Graph Fonts in Styles" on page 570:

```
replace GraphReference / linethickness=4px;
```

The following statements modify the style and produce the Q-Q plot shown in Figure 21.30:

```
proc template;
   define style Styles.NewStyle;
      parent=Styles.Statistical;
      replace GraphFonts  /
         'GraphDataFont'     = ("<MTserif>, Times New Roman",7pt)
         'GraphUnicodeFont'  = ("<MTserif>, Times New Roman",9pt)
         'GraphValueFont'    = ("<MTserif>, Times New Roman",9pt)
         'GraphLabelFont'    = ("<MTserif>, Times New Roman",10pt)
         'GraphFootnoteFont' = ("<MTserif>, Times New Roman",10pt)
         'GraphTitleFont'    = ("<MTserif>, Times New Roman",11pt)
         'GraphAnnoFont'     = ("<MTserif>, Times New Roman",10pt);
      replace GraphReference / linethickness=4px;
   end;
run;

ods listing style=NewStyle;
ods graphics on;

proc robustreg data=stack plots=qqplot;
   ods select QQPlot;
   model y = x1 x2 x3;
run;
```

Figure 21.30 Q-Q Plot That Uses the NEWSTYLE Style with a Thicker Line



You can use this approach to modify other attributes of the line, such as **`LineStyle`** and **`ContrastColor`**. These style modifications apply to all graphs that display reference lines, and not just to Q-Q plots produced by PROC ROBUSTREG. You can control the attributes of specific graphs by modifying the graph template, as discussed in the section "Graph Templates" on page 577. Values specified directly in a graph template override style attributes.

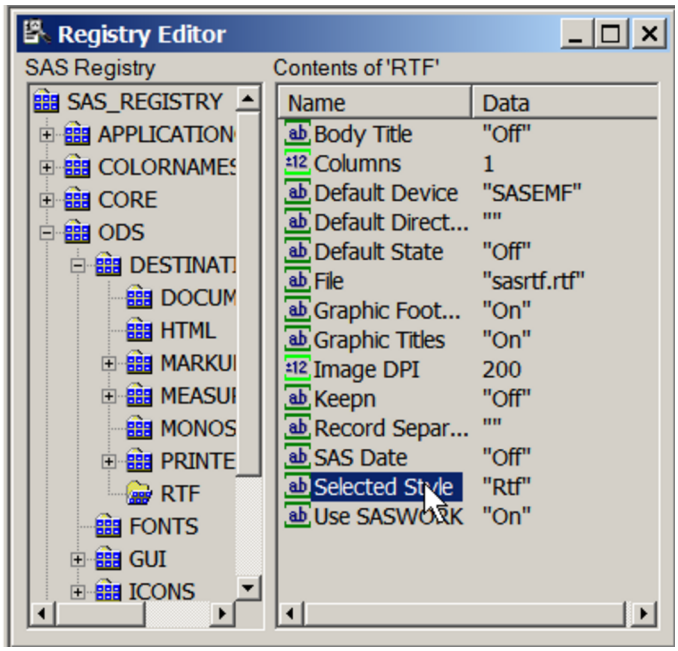
When you are done with the NEWSTYLE style, you do not need to restore the STATISTICAL style template since you did not modify it. Rather, you inherited from the STATISTICAL style.

Changing the Default Style

The default style for each ODS destination is specified in the SAS Registry. For example, the default style for the HTML destination is DEFAULT and the default style for the RTF destination is RTF. You can specify a default style for all of your output in a particular ODS destination. This is useful if you want to use a different SAS style, if you have modified one of the styles supplied by SAS (see the section "Style Definitions and Colors" on page 557), or if you have defined your own style. For example, you can specify the JOURNAL style as the default style for RTF output.

The recommended approach for specifying a default style is as follows. Open the SAS Registry Editor by typing **regedit** in the command line. Expand the node **ODS ► DESTINATIONS** and select a destination (for example, select **RTF**). Double-click the **Selected Style** item, shown in Figure 21.31, and specify a style. This can be any style supplied by SAS or a user-defined style, as long as it can be found with the current ODS path (for example, specify **Journal**). You can specify a default style for the other destinations in a similar way.

Figure 21.31 SAS Registry Editor



ODS searches sequentially through each element of the ODS PATH list for the first style definition that matches the name of the style specified in the SAS Registry. The first style definition found is used. (See the sections "Saving Customized Templates" on page 587, "Using Customized Templates" on page 587, and "Reverting to the Default Templates" on page 588 for more information about the ODS PATH.) If you are specifying a customized style as your default style, the following are useful suggestions:

- If you save your style in SASUSER.Templat, verify that the name of your default style matches the name of the style specified in the SAS Registry. For example suppose the RTF style is specified for the RTF destination in the SAS Registry. You can name your style RTF and save it in SASUSER.Templat. This blocks the RTF style in SASHELP.Tmplmst (provided that you did not alter the default sequence in the ODS PATH).
- If you save your style in a user-defined template library, verify that this template library is the first in the current ODS PATH list. Include the ODS PATH statement in your SAS autoexec file so that it is executed at startup.

For the HTML destination, an alternative approach for specifying a default style is as follows. From the menu at the top of the main SAS window, select **Tools ► Options ► Preferences**. In the **Results** tab, check the **Create HTML** box and select a style from the pull-down menu.

Graph Templates

Graph templates control the layout and details of graphs produced with ODS Graphics. SAS provides a template for every graph produced by statistical procedures. Graph template definitions are written in the Graph Template Language (GTL). This powerful language includes statements for specifying plot layouts (such as lattices or overlays), plot types (such as scatter plots and histograms), and text elements (such as titles, footnotes, and insets). It also provides support for built-in computations (such as histogram binning) and the evaluation of expressions. Options are available for specifying colors, marker symbols, and other attributes of plot features.

Graphs, like all SAS output, are constructed from two underlying components, a data component (or data object) and a template. Procedures supply a table of data values and statistical results to plot. Together, the data object and the template form an output object that ODS displays in one or more output destinations. You can control this display in two ways. You can use the ODS Graphics Editor (discussed in the section "ODS Graphics Editor" on page 548) to modify the output object (but not the underlying data object or template), and you can use the GTL to modify the template. With just a little knowledge of the GTL, you can modify or edit templates, even when you do not understand most of the syntax used in the template definition. See examples starting with Example 21.5.

NOTE: You do not need to know anything about the GTL to create statistical graphics.

This section provides an overview of the Graph Template Language. It also describes how to locate, display, edit, and save templates. A *template definition* is a set of SAS statements that is used together with PROC TEMPLATE to create a compiled template. In addition to graph templates, two other common types of templates are table templates and style templates. A table template describes how to display the output for an output object that is rendered as a table. A style template provides formatting information for visual aspects of your SAS output, including both tables and graphs. In most applications, you do not have to modify the templates that are supplied by SAS. However, when customization is necessary, you can modify the default template with the template language and PROC TEMPLATE.

Compiled templates are saved in a *template library* (also called a *template store*), which is a member of a SAS data library. The default templates supplied by SAS are stored in the SASHELP.Tmplmst template library. If you are using the SAS windowing environment, an easy way to display, edit, and save your templates is by using the Templates window. For detailed information about managing templates, see the *SAS Output Delivery System: User's Guide* and the *SAS/GRAPH: Graph Template Language User's Guide*. For details about the syntax of the graph template language, see the *SAS/GRAPH: Graph Template Language Reference*.

The Graph Template Language

Graph template definitions begin with a DEFINE STATGRAPH statement in PROC TEMPLATE, and they end with an END statement. Embedded in every graph template is a BEGINGRAPH/ENDGRAPH block, and embedded in that block are one or more LAYOUT blocks. You can specify the DYNAMIC statement to define dynamic variables (which the procedure uses to pass

values to the template definition), the MVAR and NMVAR statements to define macro variables (which you can use to pass values to the template definition), and the NOTES statement to provide descriptive information about the graph. The default templates supplied by SAS for statistical procedures are often lengthy and complex, because they provide ODS Graphics with comprehensive and detailed information about graph construction. Here is one of the simpler graph templates for a statistical procedure:

```
define statgraph Stat.MDS.Graphics.Fit;
   notes "MDS Fit Plot";
   dynamic head;
   begingraph / designwidth=defaultdesignheight;
      entrytitle HEAD;
      layout overlayequated / equatetype=square;
         scatterplot y=FITDATA x=FITDIST / markerattrs=(size=5px);
         lineparm slope=1 x=0 y=0 / extend=true lineattrs=GRAPHREFERENCE;
      endlayout;
   endgraph;
end;
```

This template, supplied for the MDS procedure, creates a scatter plot of two variables, FitData and FitDist, along with a diagonal reference line that passes through the origin. The plot is square and the axes are equated so that a centimeter on one axis represents the same data range as a centimeter on the other axis. The plot title is provided by the evaluation of the dynamic variable Head, which is set by the procedure. It is not unusual for this plot to contain hundreds or even thousands of points, so a five-pixel marker is specified, which is smaller than the seven-pixel marker used by default in most styles.

The statements available in the graph template language can be classified as follows:

- Control statements specify the conditional or iterative flow of control. By default, flow of control is sequential. In other words, each statement is used in the order in which it appears.
- Layout statements specify the arrangement of the components of the graph. Layout statements are arranged in blocks that begin with a LAYOUT statement and end with an ENDLAYOUT statement. The blocks can be nested. Within a layout block, there can be plot, text, and other statements that define one or more graph components. Options provide control for attributes of layouts and components.
- Plot statements specify a number of commonly used displays, including scatter plots, histograms, contour plots, surface plots, and box plots. Plot statements are always provided within a layout block. The plot statements include options to specify the data columns from the source objects that are used in the graph. For example, in the SCATTERPLOT statement, there are mandatory X= and Y= arguments that specify which data columns are used for the X (horizontal) and Y (vertical) axes in the plot. (In the preceding example, FitData and FitDist are the names of columns int the data object that PROC MDS creates for this graph.) There is also a GROUP= option that specifies a data column as an optional classification variable.
- Text statements specify the descriptions that accompany graphs. An entry is any textual description, including titles, footnotes, and legends; it can include symbols to identify graph elements.

The following statements display another of the simpler template definitions—the definition of the scatter plot available in PROC KDE (see Figure 45.6.1 in Chapter 45, "The KDE Procedure"):

```
proc template;
   define statgraph Stat.KDE.Graphics.ScatterPlot;
      dynamic _TITLE _DEPLABEL _DEPLABEL2;
      BeginGraph;
         EntryTitle _TITLE;
         layout Overlay;
            scatterplot x=X y=Y / markerattrs=GRAPHDATADEFAULT;
         EndLayout;
      EndGraph;
   end;
run;
```

Here, the PROC TEMPLATE and RUN statements have been added to show how you would compile the template if you wanted to modify it. The DEFINE STATGRAPH statement in PROC TEMPLATE begins the graph template definition, and the END statement ends the definition. The DYNAMIC statement defines three dynamic variables that PROC KDE sets at run time. The variable _Title provides the title of the graph. The variables _DepLabel and _DepLabel2 contain the names of the X- and Y-variables, respectively. If you were to modify this template, you could use these dynamic text variables in any text element of the graph definition.

The overall display is specified with the LAYOUT OVERLAY statement inside the BEGINGRAPH/ENDGRAPH block. The title of the graph is specified with the ENTRYTITLE statement. The main plot is a scatter plot specified with the SCATTERPLOT statement. The options in the SCATTERPLOT statement are given after the slash and specify display options such as marker attributes (symbol, color, and size). These attributes can be specified directly, as in the PROC MDS template, or more typically by using indirect references to style attributes, as in the PROC KDE template. The values of these attributes are specified in the definition of the style you are using and are automatically set to different values if you specify a different style. For more information about style references, see the section "Styles" on page 554. The ENDLAYOUT statement ends the main layout block. For details about the syntax of the graph template language, see the *SAS/GRAPH: Graph Template Language Reference*.

You can write your own templates and use them to display raw data or output from procedures. For example, consider the iris data from Example 31.1 of Chapter 31, "The DISCRIM Procedure." The following statements create the SAS data set:

```
proc format;
   value specname
      1='Setosa    '
      2='Versicolor'
      3='Virginica ';
run;

data iris;
   input SepalLength SepalWidth PetalLength PetalWidth
         Species @@;
   format Species specname.;
   label SepalLength='Sepal Length in mm.'
         SepalWidth ='Sepal Width in mm.'
         PetalLength='Petal Length in mm.'
         PetalWidth ='Petal Width in mm.';
   datalines;
50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3

... more lines ...

;
```

The following statements create a template for a scatter plot of the variables PetalLength and PetalWidth with a legend:

```
proc template;
   define statgraph scatter;
      begingraph;
         entrytitle 'Fisher (1936) Iris Data';
         layout overlayequated / equatetype=fit;
            scatterplot x=petallength y=petalwidth /
                        group=species name='iris';
            layout gridded / autoalign=(topleft);
               discretelegend 'iris' / border=false opaque=false;
            endlayout;
         endlayout;
      endgraph;
   end;
run;
```

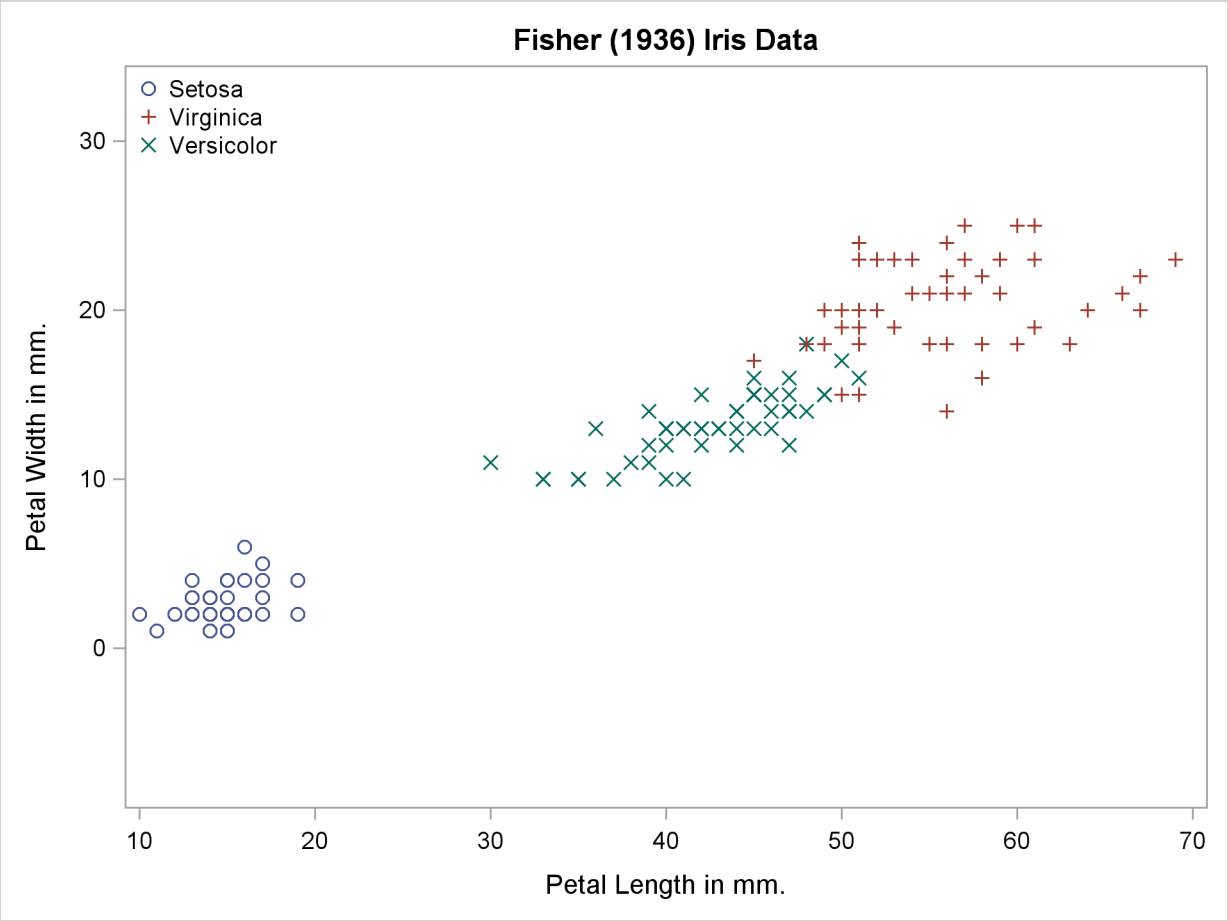
The layout is OVERLAYEQUATED, which equates the plot. However, unlike the PROC MDS template, which used EQUATETYPE=SQUARE to make a square plot, the EQUATETYPE=FIT option specifies that the lengths of the axes in this plot should fill the entire plotting area. A legend is placed internally in the top-left portion of the plot. There are three groups of observations, indicated by the three species, and each group is plotted with a separate color and symbol that depends on the ODS style. The legend identifies each group. The NAME= option provides the link between the SCATTERPLOT statement and the DISCRETELEGEND statement. An explicit link is needed since some graphical displays are based on multiple SCATTERPLOT statements or other plotting statements.

The following step creates the plot by using the SGRENDER procedure, the Iris data set, and the custom template **scatter**:

```
proc sgrender data=iris template=scatter;
run;
```

The syntax of PROC SGRENDER is very simple, because all of the graphical options appear in the template. The scatter plot in Figure 21.32 shows the results.

Figure 21.32 Petal Width and Petal Length in Three Iris Species



The intent of this example is to illustrate how you can write a template to create a scatterplot. PROC TEMPLATE and PROC SGRENDER provide you with the power to create highly customized displays. However, usually you can use the SGPLOT, SGSCATTER or SGPANEL procedures instead, which are much simpler to use. These procedures are discussed in section "Statistical Graphics Procedures" on page 589. See the section "Grouped Scatter Plot with PROC SGPLOT" on page 517 and Figure 21.12 for an example that plots these data with PROC SGPLOT.

Locating Templates

Before you can customize a graph, you must determine which template is used to create the original graph. You can do this by submitting the ODS TRACE ON statement before the procedure statements that create the graph. The fully qualified template name is displayed in the SAS log. Here is an example:

```
ods trace on;
ods graphics on;

proc reg data=sashelp.class;
   model Weight = Height;
run; quit;
```

The preceding statements create the following trace output, which provides information about both the graphs and tables produced by PROC REG:

```
Output Added:
-------------
Name:        NObs
Label:       Number of Observations
Template:    Stat.Reg.NObs
Path:        Reg.MODEL1.Fit.Weight.NObs
-------------

Output Added:
-------------
Name:        ANOVA
Label:       Analysis of Variance
Template:    Stat.REG.ANOVA
Path:        Reg.MODEL1.Fit.Weight.ANOVA
-------------

Output Added:
-------------
Name:        FitStatistics
Label:       Fit Statistics
Template:    Stat.REG.FitStatistics
Path:        Reg.MODEL1.Fit.Weight.FitStatistics
-------------

Output Added:
-------------
Name:        ParameterEstimates
Label:       Parameter Estimates
Template:    Stat.REG.ParameterEstimates
Path:        Reg.MODEL1.Fit.Weight.ParameterEstimates
-------------
```

```
Output Added:
-------------
Name:        DiagnosticsPanel
Label:       Fit Diagnostics
Template:    Stat.REG.Graphics.DiagnosticsPanel
Path:        Reg.MODEL1.ObswiseStats.Weight.DiagnosticPlots.DiagnosticsPanel
-------------

Output Added:
-------------
Name:        ResidualPlot
Label:       Height
Template:    Stat.REG.Graphics.ResidualPlot
Path:        Reg.MODEL1.ObswiseStats.Weight.ResidualPlots.ResidualPlot
-------------

Output Added:
-------------
Name:        FitPlot
Label:       Fit Plot
Template:    Stat.REG.Graphics.Fit
Path:        Reg.MODEL1.ObswiseStats.Weight.FitPlot
-------------
```

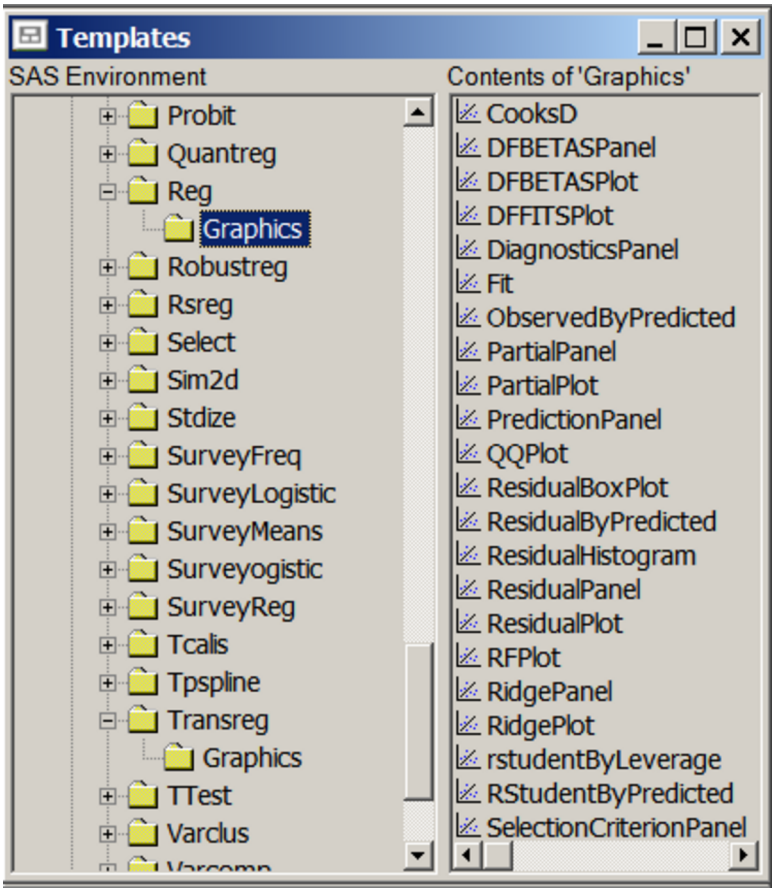
This is also illustrated in Example 21.5 and the section "Paths and Selection" on page 435 in Chapter 20, "Using the Output Delivery System."

Displaying Templates

Once you have found the fully qualified name of a template, you can display its definition (source program) by using one of these methods:

- Open the Templates window by issuing the command **odstemplates** (**odst** for short) in the command line of the SAS windowing environment. The template window is shown in Figure 21.33. If you expand the SASHELP.Tmplmst node, you can view all the available templates and double-click any template icon to display its definition. This is illustrated in Example 21.5.
- Use the SOURCE statement in PROC TEMPLATE to display a template definition in the SAS log or write the definition to a file.

Figure 21.33 Requesting the Templates Window in the Command Line



For example, the following statements display the template for the PROC REG residual plot:

```
proc template;
   source Stat.REG.Graphics.ResidualPlot;
run;
```

The template is displayed as follows:

```
define statgraph Stat.Reg.Graphics.ResidualPlot;
   notes "Residual Plot";
   dynamic _XVAR _SHORTXLABEL _TITLE _LOESSLABEL _DEPNAME
           _MODELLABEL _SMOOTH;
   BeginGraph;
      entrytitle halign=left textattrs=GRAPHVALUETEXT
         _MODELLABEL halign=center
         textattrs=GRAPHTITLETEXT _TITLE " for " _DEPNAME;
      entrytitle textattrs=GRAPHVALUETEXT _LOESSLABEL;
      layout overlay / xaxisopts=(shortlabel=_SHORTXLABEL);
         referenceline y=0;
         scatterplot y=RESIDUAL x=_XVAR / primary=true
            rolename=(_tip1=OBSERVATION _id1=ID1 _id2=ID2
            _id3=ID3 _id4=ID4 _id5=ID5) tip=(y x
```

```
            _tip1 _id1 _id2 _id3 _id4 _id5);
         if (EXISTS(_SMOOTH))
            loessplot y=_SMOOTH x=_XVAR /
               tiplabel=(y="Smoothed Residual");
         endif;
      endlayout;
   EndGraph;
end;
```

PROC TEMPLATE also tells you where the template is located. In this case, it prints the following note:

```
NOTE: Path 'Stat.Reg.Graphics.ResidualPlot' is in: SASHELP.TMPLMST.
```

The word "Path" in ODS refers to any name or label hierarchy. In the note, the levels of the template name form a path. In the trace output, the levels of the plot name form a different path.

Editing Templates

You can modify the format and appearance of a particular graph by doing the following:

- Modify its template definition (source program).
- Submit the revised template to create a new compiled template.
- Ensure that the ODS search path finds and uses your new template.

Template libraries are designated read-only (such as SASHELP.Tmplmst) or updatable (such as SASUSER.Templat).

If you view the templates in an updatable template library from the Templates window, you can select **Open** or **Edit** from the pop-up menu. Either the Template Browser or Template Editor window opens. In the Template Editor window, you can make changes and submit the code directly. For read-only templates or when you select **Open**, the Template Browser window opens and you must copy the definition to an editor window to make changes. Since templates supplied by SAS are in the read-only SASHELP library, an easy way to obtain an editable program file is to use the SOURCE statement with the FILE= option in PROC TEMPLATE to write the template definition to a file as follows:

```
proc template;
   source Stat.REG.Graphics.ResidualPlot / file="residtpl.sas";
run;
```

By default, the file is saved in the SAS current folder. Alternatively, you can omit the slash and the FILE= option and copy and paste the source from the SAS log into an editor. Either way, you must add a PROC TEMPLATE statement before the generated source statements and a RUN statement after the END statement before you submit your modified definition.

Graph definitions are self-contained and do not support inheritance or "parenting" as do table definitions. Consequently, the EDIT statement in PROC TEMPLATE is not supported for graph definitions.

Here are some important points about what you can and cannot change in a template supplied by SAS while preserving its overall functionality:

- Do not change the template name. A statistical procedure can access only a predefined list of templates. If you change the name, the procedure cannot find your template. You must keep the original name and make sure that it is in a template library that is read before SASHELP.Tmplmst. You control this with the ODS PATH statement (see the section "The Default Template Libraries and the ODS PATH" on page 553, the section "Saving Customized Templates" on page 587, and subsequent sections for more information about the ODS PATH).
- Do not change the names of columns. The underlying data object contains predefined column names that you must use. Be very careful if you change how a column is used in a template. Usually, columns are not interchangeable.
- Do not change the names of DYNAMIC variables. Procedures set values only for a predefined list of dynamic variables. Changing dynamic variable names can lead to runtime errors. Do not add dynamic variables, because the procedure cannot set their values. A few procedures document additional dynamic variables that can be defined in the template if you want to add more information to the output, such additional statistics in an inset table. See the section "Modifying the Layout and Adding a New Inset Table" on page 627 for an example.
- Do not change the names of statements (for example, from a SCATTERPLOT to a NEEDLEPLOT or other type of plot).

You can change any of the following:

- You can add macro variables that behave like dynamic variables. They are resolved at the time that the statistical procedure is run, and not at the time that the template is compiled. They are defined with an MVAR or NMVAR statement at the beginning the template. You can set the value of each macro variable with a %LET statement before the statistical procedure is run. See Example 21.10. You can also move a variable from a DYNAMIC statement to an MVAR or NMVAR statement if you want to set it yourself rather than letting the procedure set it.
- You can change the graph size.
- You can change graph titles, footnotes, axis labels, and any other text that appears in the graph.
- You can change which plot features are displayed.
- You can change axis features, such as grid lines, offsets, view ports, tick value formatting, and so on.
- You can change the content and arrangement of insets (small tables of statistics embedded in some graphs).

- You can change the legend location, contents, border, background, title, and so on.

See the *SAS/GRAPH: Graph Template Language Reference* for information about the syntax of the statements in the Graph Template Language.

Saving Customized Templates

After you edit the template definition, you can submit your PROC TEMPLATE statements as you would any other SAS program. If you are using the Template Editor window, select **Submit** from the **Run** menu. See Example 21.5. Alternatively, submit your PROC TEMPLATE statements from the Program Editor. ODS automatically saves the compiled template in the first template library that it can update, according to the currently defined ODS path. If you have not changed the ODS path, then the modified template is saved in the SASUSER.Templat template library. You can display the current ODS path with the following statement:

```
ods path show;
```

Here are the log messages for the default path:

```
Current ODS PATH list is:

1. SASUSER.TEMPLAT(UPDATE)
2. SASHELP.TMPLMST(READ)
```

If you want to store modified templates in another template library, you can use the ODS PATH statement to add that template library to the front of the list. To use these templates, you must make sure the ODS path is set correctly before you attempt to access them in the other SAS sessions. See the section "Using Customized Templates" on page 587.

Using Customized Templates

When you create ODS output (either graphs or tables), ODS searches sequentially through each library in the ODS PATH list for a template that matches the one requested. If you have not changed the default ODS path, then ODS searches the SASUSER.Templat library first, then SASHELP.Tmplmst. ODS uses the first template that it finds with the requested name. **NOTE:** Templates with the same name can exist in more than one template library.

The ODS PATH statement specifies the libraries to search, as well as the order in which to search them. You can change the default path by using the ODS PATH statement. For example, the following statement sets the ODS path so that the template library WORK.Mystore is searched first, followed by SASHELP.Tmplmst:

```
ods path work.mystore(update) sashelp.tmplmst(read);
```

The UPDATE option provides update access as well as read access to WORK.Mystore. The READ option provides read-only access to SASHELP.Tmplmst. With this path, the template library SASUSER.Templat is no longer searched. You can verify this with the following statement:

```
ods path show;
```

Here are the log messages generated by the preceding statement:

```
Current ODS PATH list is:

1. WORK.MYSTORE(UPDATE)
2. SASHELP.TMPLMST(READ)
```

For more information, see the *SAS Output Delivery System: User's Guide* and the *SAS/GRAPH: Graph Template Language User's Guide*. Example 21.5 illustrates all the steps of displaying, editing, saving, and using customized templates.

Reverting to the Default Templates

Customized templates are stored in SASUSER.Templat or in some other template library that you create. The templates supplied by SAS are in the read-only template library SASHELP.Tmplmst. If you have modified any of the supplied templates and you want to use the original default templates, you can change your ODS path as follows:

```
ods path sashelp.tmplmst(read) sasuser.templat(update);
```

This way the default templates are found first. Alternatively, you can save all of your customized templates in a user-defined template library (for example WORK.Mystore). To access these templates, you submit the following statement before running your analysis:

```
ods path mylib.mystore(update) sashelp.tmplmst(read);
```

When you are done, you can reset the default ODS path as follows:

```
ods path reset;
```

This restores the ODS path to its original state (`sasuser.templat(update) sashelp.tmplmst(read)`). You can also save your customized template as part of your SAS program. You can delete it from the SASUSER.Templat template library when you are done, as in the following statements:

```
proc template;
   delete Stat.REG.Graphics.ResidualPlot;
run;
```

The following note is printed in the SAS log:

```
NOTE: 'Stat.REG.Graphics.ResidualPlot' has been deleted from: SASUSER.TEMPLAT
```

You can run the following step to delete the entire SASUSER.Templat library of customized templates:

```
ods path sashelp.tmplmst(read);
proc datasets library=sasuser;
   delete templat(memtype=itemstor);
run;
ods path sasuser.templat(update) sashelp.tmplmst(read);
```

Statistical Graphics Procedures

There are three statistical graphics procedures in SAS/GRAPH that use ODS Graphics and provide convenient syntax for creating a variety of plots from raw data or from procedure output.

SGSCATTER	creates single-cell and multi-cell scatter plots and scatter plot matrices with optional fits and ellipses.
SGPLOT	creates single-cell plots with a variety of plot and chart types.
SGPANEL	creates single-page or multi-page panels of plots and charts conditional on classification variables.

In addition, the SGRENDER procedure in SAS/GRAPH provides a way to create plots from graph templates that you have modified or written yourself.

See the *SAS/GRAPH: Statistical Graphics Procedures Guide* for more information. You do not need to enable ODS Graphics in order to use the SG procedures.

These procedures do much more than make scatter plots. They can produce density plots, dot plots, needle plots, series plots, horizontal and vertical bar charts, histograms, and box plots. They can also compute and display loess fits, polynomial fits, penalized B-spline fits, reference lines, bands, and ellipses. PROC SGRENDER is the most flexible because it uses the Graph Template Language. The syntax for the other SG procedures is much simpler than that of the GTL, and so these procedures are recommended for creating most plots commonly required in statistical work.

The SGPLOT Procedure

PROC SGPLOT provides a simple way to make a variety of scatter plots. This example is taken from Example 50.4 of Chapter 50, "The LOESS Procedure." The following statements create a SAS data set that contains information about differences in ocean pressure over time:

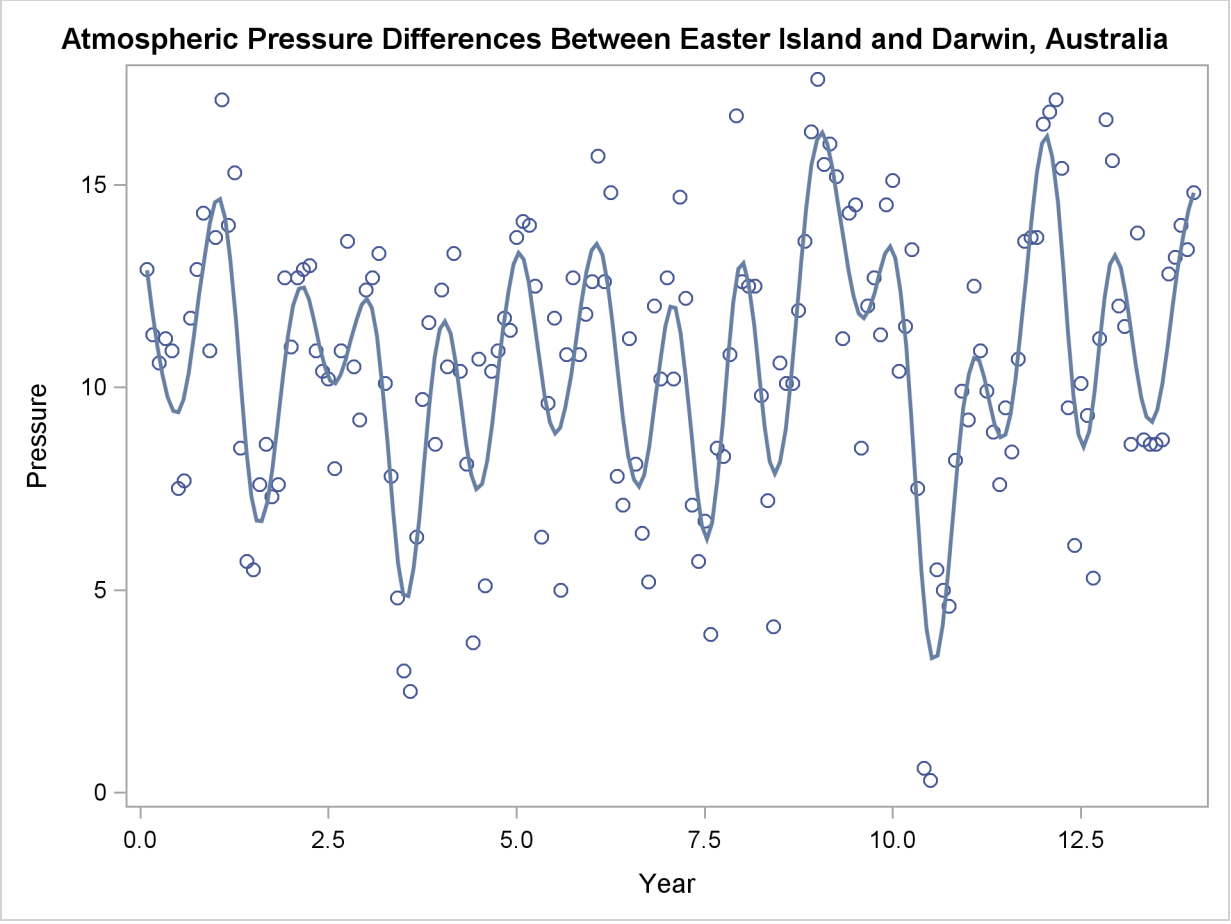
```
data ENSO;
   input Pressure @@;
   Year = _n_ / 12;
   datalines;
12.9  11.3  10.6  11.2  10.9   7.5   7.7  11.7

... more lines ...
```

```
;
```

The following statements create a scatter plot of points along with a penalized B-spline fit to the data and produce Figure 21.34:

```
proc sgplot data=enso noautolegend;
   title 'Atmospheric Pressure Differences Between '
         'Easter Island and Darwin, Australia';
   pbspline y=pressure x=year;
run;
```

Figure 21.34 Penalized B-Spline Fit with PROC SGPLOT



See Chapter 90, "The TRANSREG Procedure," for more information about penalized B-splines. Also see the section "Grouped Scatter Plot with PROC SGPLOT" on page 517 and Figure 21.12 for an example of a scatter plot with groups of observations.

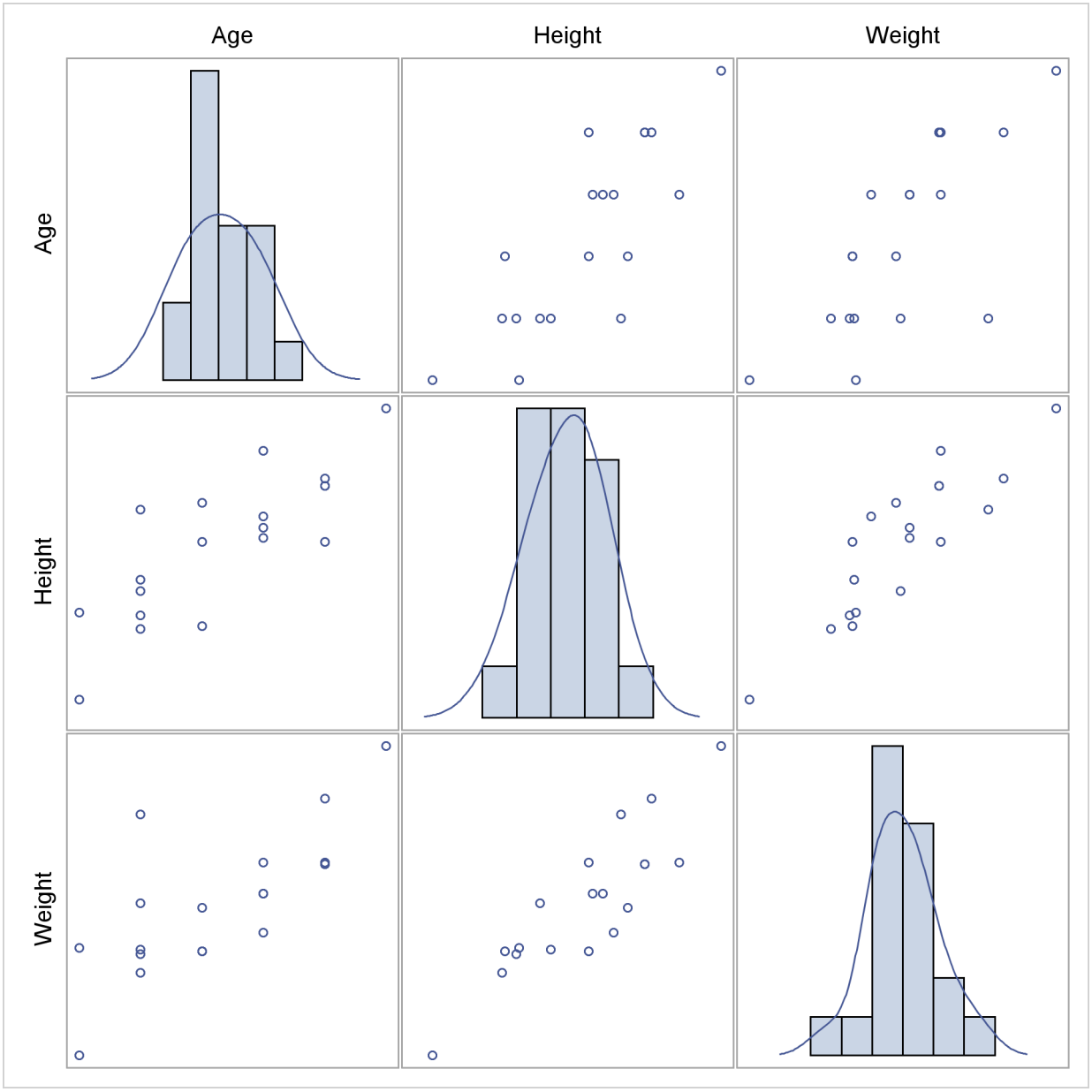
The SGSCATTER Procedure

You can use the SGSCATTER procedure to produce scatter plot matrices. The following step creates a scatter plot matrix from all of the numeric variables in the Class data set available in the SASHELP library and produces Figure 21.35:

```
proc sgscatter data=sashelp.class;
   matrix _numeric_ / diagonal=(kernel histogram);
run;
```

The diagonal cells of Figure 21.35 contain a histogram and a kernel density fit. The off-diagonal cells contain all pairs of scatter plots.

Figure 21.35 Scatter Plot Matrix with PROC SGSCATTER



The MATRIX statement creates a symmetric $n \times n$ scatter plot matrix. Other statements are available as well. The PLOT statement creates a panel containing one or more individual scatter plots. The COMPARE statement creates a rectangular $m \times n$ scatter plot matrix. Linear and nonlinear fits can be added, and many graphical features can be requested with options.

The SGPANEL Procedure

The SGPANEL procedure creates paneled plots and charts with one or more classification variables. Classification variables can be designated as row or column variables, or there can be multiple classifications. Graphs are drawn for each combination of the levels of classification variables, showing a subset of the data in each cell.

This example is taken from Example 38.6 of Chapter 38, "The GLIMMIX Procedure." The following statements create the input SAS data sets:

```
data times;
   input time1-time23;
   datalines;
122  150  166  179  219  247  276  296  324  354  380  445
478  508  536  569  599  627  655  668  723  751  781
;

data cows;
   if _n_ = 1 then merge times;
   array t{23} time1 - time23;
   array w{23} weight1 - weight23;
   input cow iron infection weight1-weight23 @@;
   do i=1 to 23;
      weight = w{i};
      tpoint = (t{i}-t{1})/10;
      output;
   end;
   keep cow iron infection tpoint weight;
   datalines;
 1 0 0  4.7    4.905  5.011  5.075  5.136  5.165  5.298  5.323
        5.416  5.438  5.541  5.652  5.687  5.737  5.814  5.799
        5.784  5.844  5.886  5.914  5.979  5.927  5.94

... more lines ...

;
```

First, PROC GLIMMIX is run to fit the model, and then the results are prepared for plotting:

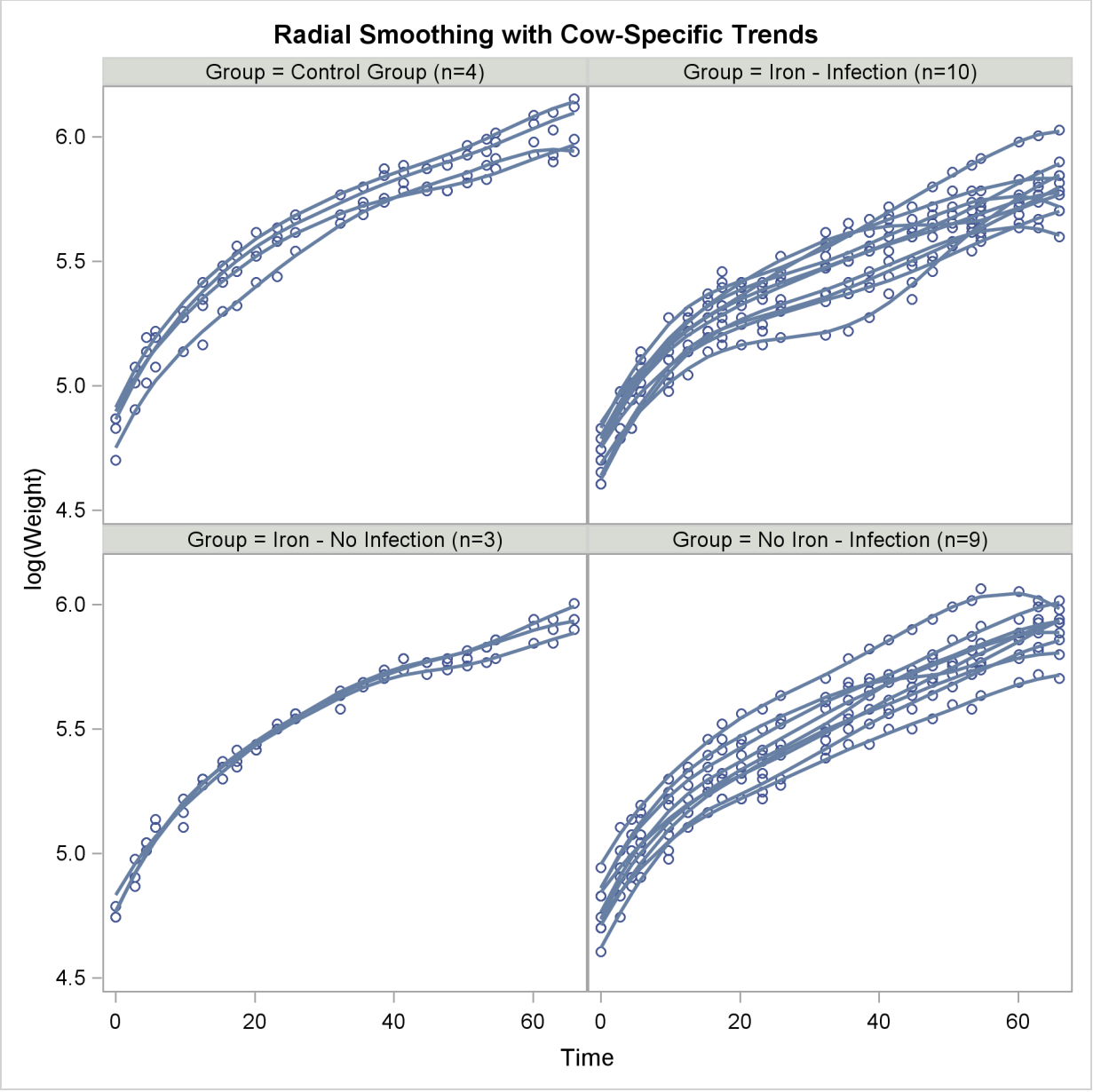
```
proc glimmix data=cows;
  t2 = tpoint / 100;
  class cow iron infection;
  model weight = iron infection iron*infection tpoint;
  random t2 / type=rsmooth subject=cow
                  knotmethod=kdtree(bucket=100 knotinfo);
  output out=gmxout pred(blup)=pred;
  nloptions tech=newrap;
run;

data plot;
  set gmxout;
  length Group $ 26;
       if (iron=0) and (infection=0) then group='Control Group (n=4)';
  else if (iron=1) and (infection=0) then group='Iron - No Infection (n=3)';
  else if (iron=0) and (infection=1) then group='No Iron - Infection (n=9)';
  else group = 'Iron - Infection (n=10)';
run;

proc sort data=plot; by group cow;
run;
```

The following statements produce graphs of the observed data and fitted profiles in the four groups:

```
proc sgpanel data=plot noautolegend;
  title 'Radial Smoothing with Cow-Specific Trends';
  label tpoint='Time' weight='log(Weight)';
  panelby group / columns=2 rows=2;
  scatter x=tpoint y=weight;
  series  x=tpoint y=pred / group=cow lineattrs=GraphFit;
run;
```

The results are shown in Figure 21.36.

Figure 21.36 Fit Using PROC SGPANEL



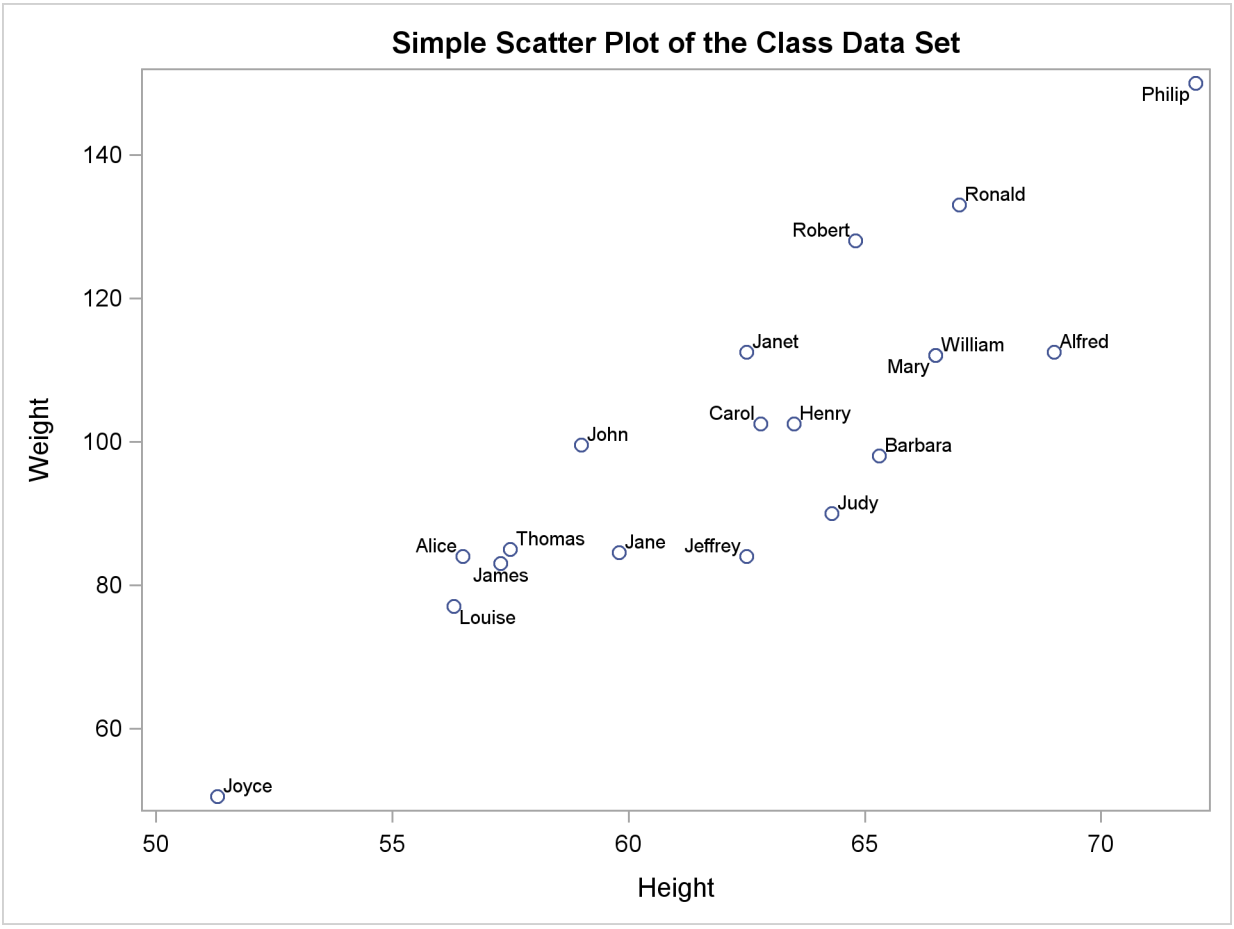
The SGRENDER Procedure

The SGRENDER procedure produces a graph from an input SAS data set and an ODS graph template. With PROC SGRENDER and the Graph Template Language (GTL), you can create highly customized graphs. The following steps create a simple scatter plot of the Class data set available in the SASHELP library and produce Figure 21.37:

```
proc template;
   define statgraph Scatter;
      begingraph;
         entrytitle "Simple Scatter Plot of the Class Data Set";
         layout overlay;
            scatterplot y=weight x=height / datalabel=name;
         endlayout;
      endgraph;
   end;
run;

proc sgrender data=sashelp.class template=scatter;
run;
```

The template definition consists of an outer block that begins with a DEFINE statement and ends with an END statement. Inside of that is a BEGINGRAPH/ENDGRAPH block. Inside that block, the ENTRYTITLE statement provides the plot title, and the LAYOUT OVERLAY block contains the statement or statements that define the graph. In this case, there is just a single SCATTERPLOT statement that names the Y-axis (vertical) variable, the X-axis (horizontal) variable, and an optional variable that contains labels for the points. The PROC SGRENDER statement simply specifies the input data set and the template. The real work in using PROC SGRENDER is writing the template.

Figure 21.37 Scatter Plot of Labeled Points with PROC SGRENDER



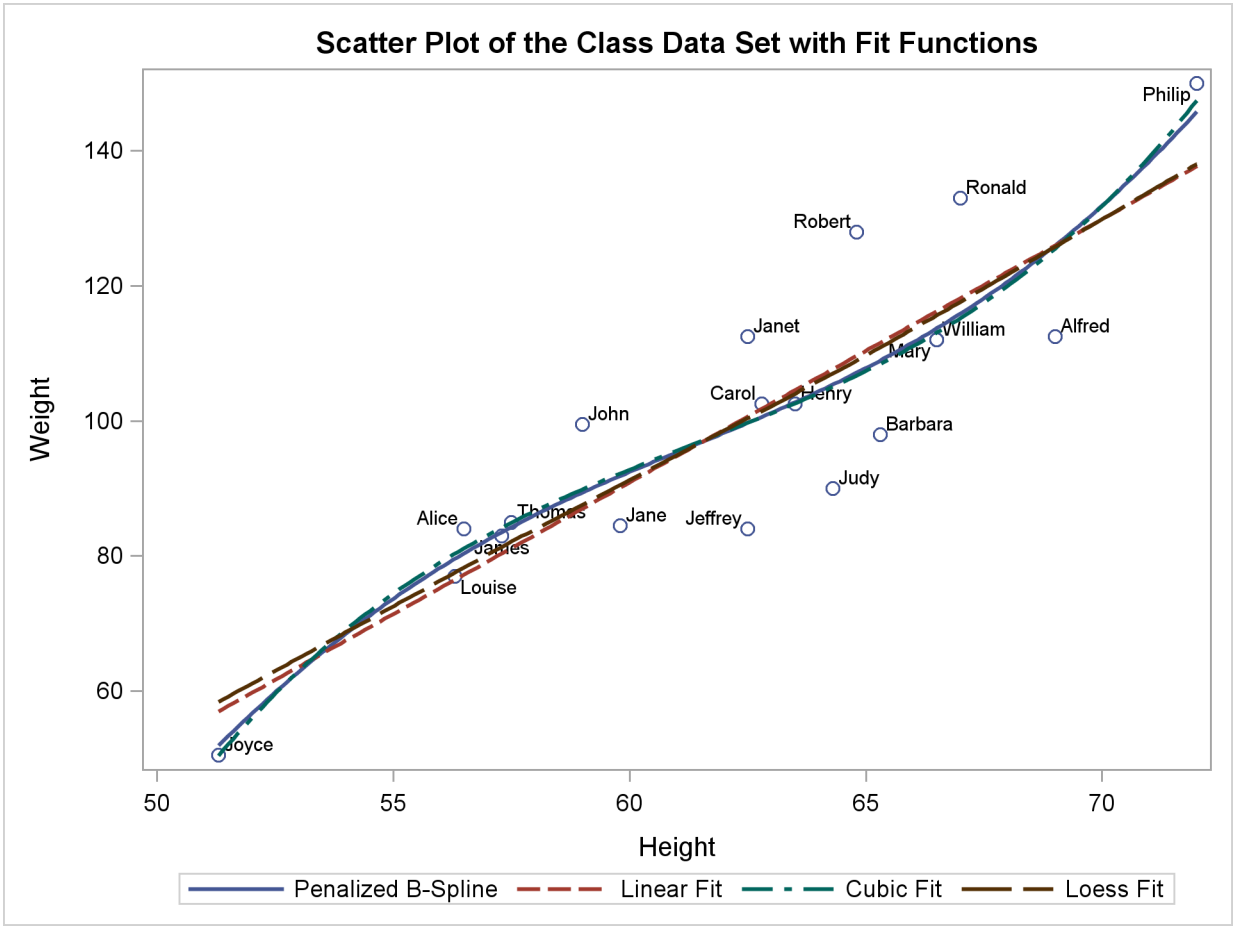
The following steps add a series of fit functions to the scatter plot and create a legend by adding statements to the **Scatter** template:

```
proc template;
   define statgraph Scatter;
      begingraph;
         entrytitle "Scatter Plot of the Class Data Set with Fit Functions";
         layout overlay;
            scatterplot y=weight x=height / datalabel=name;
            pbsplineplot y=weight x=height / name='pbs'
                         legendlabel='Penalized B-Spline'
                         lineattrs=GraphData1;
            regressionplot y=weight x=height / degree=1 name='line'
                         legendlabel='Linear Fit'
                         lineattrs=GraphData2;
            regressionplot y=weight x=height / degree=3 name='cubic'
                         legendlabel='Cubic Fit'
                         lineattrs=GraphData3;
            loessplot y=weight x=height / name='loess'
                         legendlabel='Loess Fit'
                         lineattrs=GraphData4;
```

```
            discretelegend 'pbs' 'line' 'cubic' 'loess';
         endlayout;
      endgraph;
   end;
run;

proc sgrender data=sashelp.class template=scatter;
run;
```

The line attributes for each function are specified with different style elements, **GraphData1** through **GraphData4**, so that the functions are adequately identified in the legend. The preceding statements create Figure 21.38.

Figure 21.38 Scatter Plot and Fit Functions with PROC SGRENDER

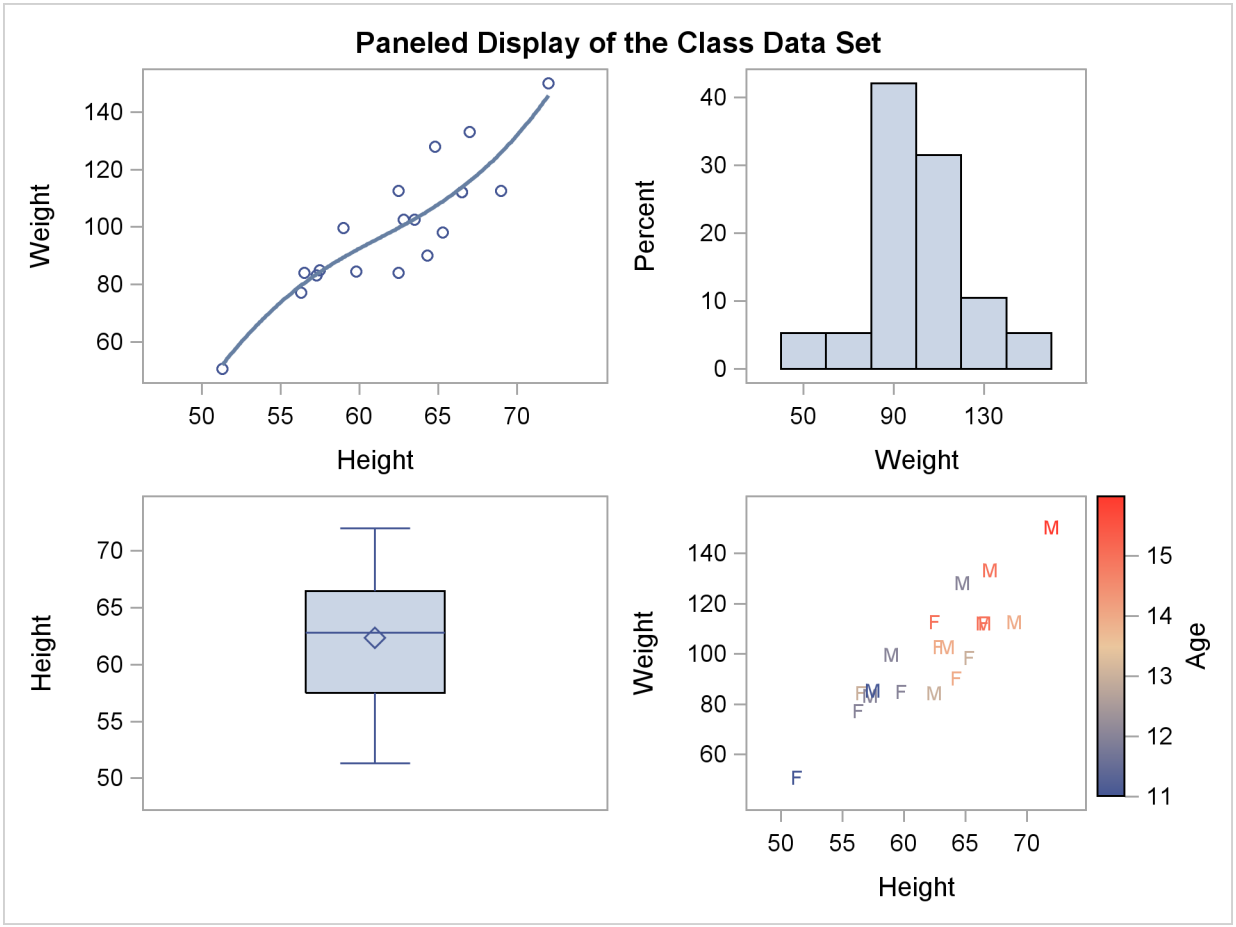


The following statements create a four-panel display of the Class data set and produce Figure 21.39:

```
proc template;
   define statgraph Panel;
      begingraph;
         entrytitle "Paneled Display of the Class Data Set";

         layout lattice / rows=2 columns=2 rowgutter=10 columngutter=10;

            layout overlay;
               scatterplot y=weight x=height;
               pbsplineplot y=weight x=height;
            endlayout;

            layout overlay / xaxisopts=(label='Weight');
               histogram weight;
            endlayout;

            layout overlay / yaxisopts=(label='Height');
               boxplot y=height;
            endlayout;

            layout overlay / xaxisopts=(offsetmin=0.1 offsetmax=0.1)
                             yaxisopts=(offsetmin=0.1 offsetmax=0.1);
               scatterplot  y=weight x=height / markercharacter=sex
                  name='color' markercolorgradient=age;
               continuouslegend 'color'/ title='Age';
            endlayout;

         endlayout;
      endgraph;
   end;
run;

proc sgrender data=sashelp.class template=panel;
run;
```

In this template, the outermost layout is a LAYOUT LATTICE. It creates a 2×2 panel of plots with a ten-pixel separation (or gutter) between each plot. Inside the lattice are four LAYOUT OVERLAY blocks—each defining one of the graphs. The first is a simple scatter plot with a nonlinear penalized B-spline fit. The second is a histogram of the dependent variable Weight. The third is a box plot of the independent variable Height. The fourth simultaneously shows height, weight, age, and sex for the students in the class. Each axis has an offset added at both the maximum and minimum. This provides padding between the axes and the data.

Figure 21.39 Multiple Panels Using PROC SGRENDER



Many other types of graphs are available with the SG procedures. However, even the few examples provided here show the power and flexibility available for making professional-quality statistical graphics. See the *SAS/GRAPH: Graph Template Language User's Guide* and the *SAS/GRAPH: Statistical Graphics Procedures Guide* for more information.

Examples of ODS Statistical Graphics

Example 21.1: Creating Graphs with Tool Tips in HTML

This example demonstrates how to request graphics in HTML that are enhanced with tooltip displays, which appear when you move a mouse over certain features of the graph. When you specify the HTML destination and the IMAGEMAP=ON option in the ODS GRAPHICS statement, an image map of coordinates for tooltips is generated along with the HTML output file. Individual graphs are saved as PNG files.

Example 56.2 and Example 56.8 of Chapter 56, "The MIXED Procedure," analyze a data set with repeated growth measurements for 27 children. The following step creates the data set:

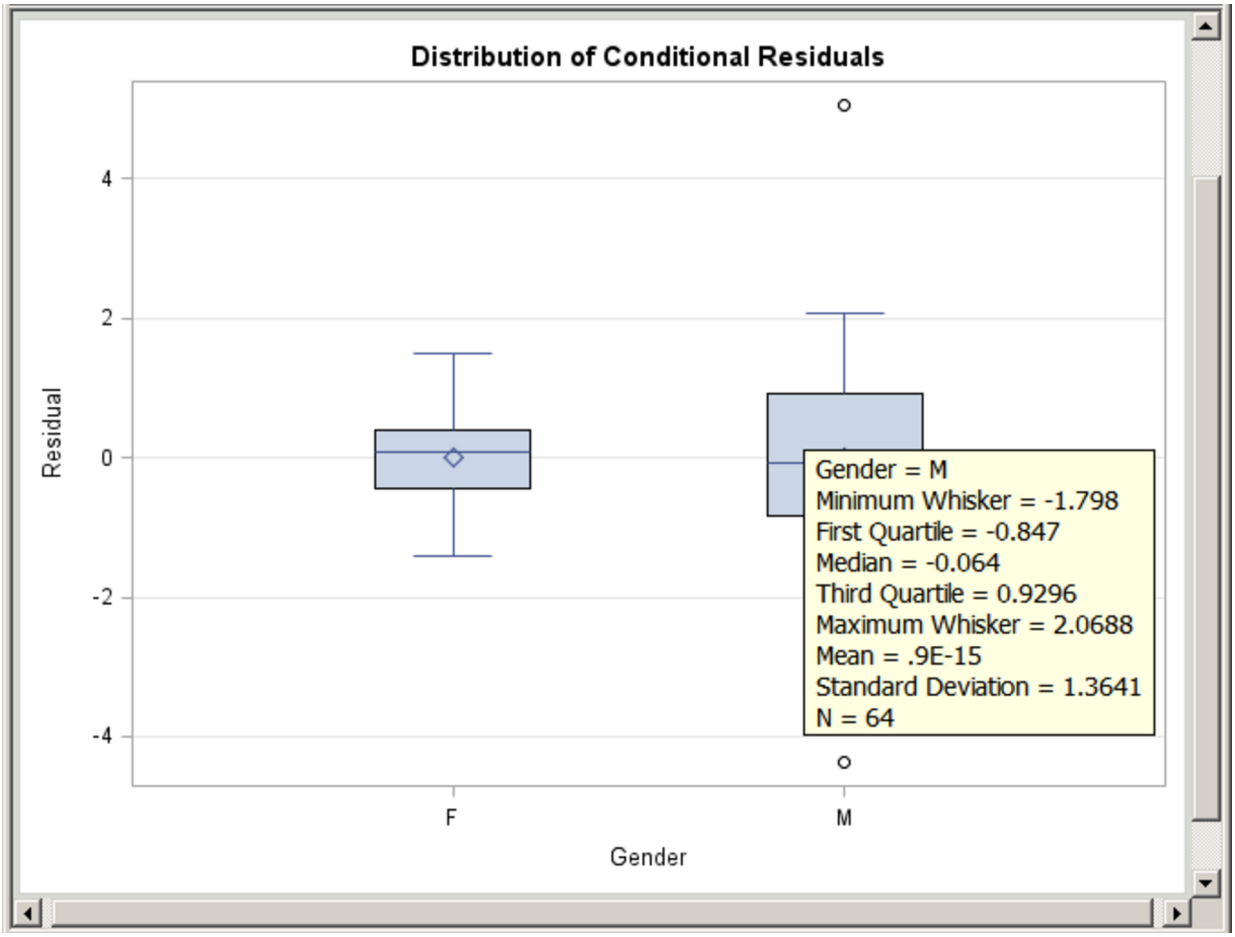
```
data pr;
   input Person Gender $ y1 y2 y3 y4 @@;
   y=y1; Age=8;  output;
   y=y2; Age=10; output;
   y=y3; Age=12; output;
   y=y4; Age=14; output;
   drop y1-y4;
   datalines;
 1  F  21.0  20.0  21.5  23.0      2  F  21.0  21.5  24.0  25.5

... more lines ...

;
```

The following statements fit a mixed model with random intercepts and slopes for each child:

```
ods graphics on / imagemap=on;
ods html body='b.html' style=statistical;

proc mixed data=pr method=ml plots=boxplot;
   ods select 'Conditional Residuals by Gender';
   class Person Gender;
   model y = Gender Age Gender*Age;
   random intercept Age / type=un subject=Person;
run;

ods html close;
```

The PLOTS=BOXPLOT option in the PROC MIXED statement requests box plots of observed values and residuals for each classification main effect in the model (Gender and Person). Only the by-gender box plots are actually created due to the ODS SELECT statement, which uses the plot label to select the plot. Output 21.1.1 displays the results. Moving the mouse over a box plot displays a tooltip with summary statistics for the class level. Graphics with tooltips are supported for only the HTML destination.

Output 21.1.1 Box Plot with Tool Tips



Example 21.2: Creating Graphs for a Presentation

The RTF destination provides the easiest way to create ODS graphs for inclusion into a paper or presentation. You can specify the ODS RTF statement to create a file that is easily imported into a document (such as Microsoft Word or WordPerfect) or a presentation (such as Microsoft PowerPoint).

The following statements simulate 100 observations from the model $y = \log(x) + e$, where $x = 1, \ldots, 100$ and e has a normal distribution with mean 0 and variance 1:

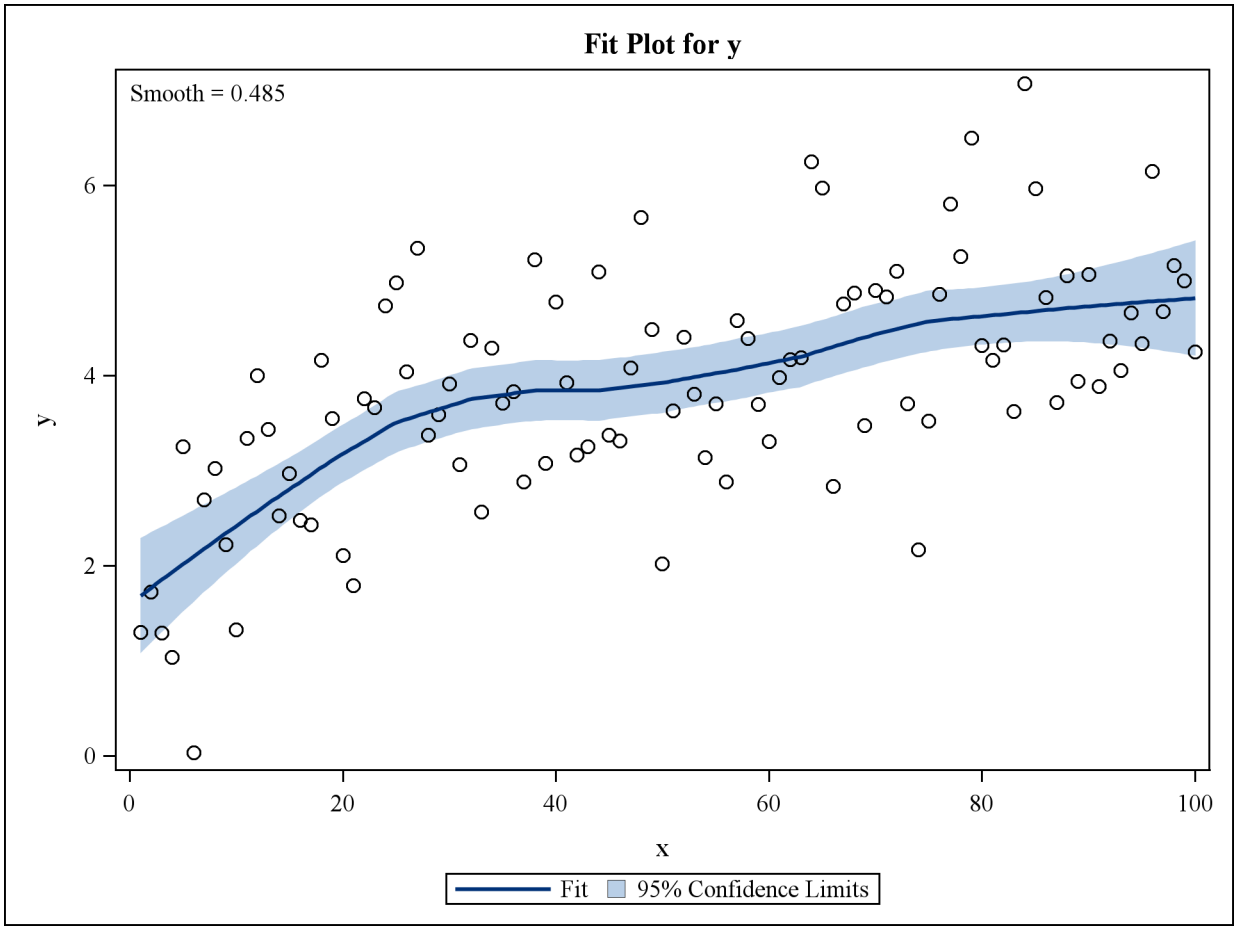
```
data one;
   do x = 1 to 100;
      y = log(x) + normal(104);
      output;
   end;
run;
```

The following statements request a loess fit and save the output in the file *loess.rtf*:

```
ods listing close;
ods rtf file="loess.rtf";
ods graphics on;

proc loess data=one;
   model y = x / clm residual;
run;

ods rtf close;
ods listing;
```

The output file includes various tables and the following plots: a plot of the selection criterion versus smoothing parameter, a fit plot with 95% confidence bands, a plot of residual by regressors, and a diagnostics panel. The fit plot is produced with the RTF style and is shown in Output 21.2.1.

Output 21.2.1 Loess Fit Plot with the RTF Style



If you are running SAS in the Windows operating system, you can open the RTF file in Microsoft Word and simply copy and paste the graphs into Microsoft PowerPoint. In general, RTF output is convenient for exchange of graphical results between Windows applications through the clipboard.

Alternatively, if you request ODS Graphics with the LISTING or HTML destinations, then your individual graphs are created as PNG files by default. You can insert these files into a Microsoft PowerPoint presentation. See the sections "Naming Graphics Image Files" on page 542 and "Saving Graphics Image Files" on page 544 for information about how the image files are named and saved.

Example 21.3: Creating Graphs in PostScript Files

This example illustrates how to create individual graphs in PostScript files. This is particularly useful when you want to include them in a LaTeX document. Consider again the data set Stack created by the following statements:

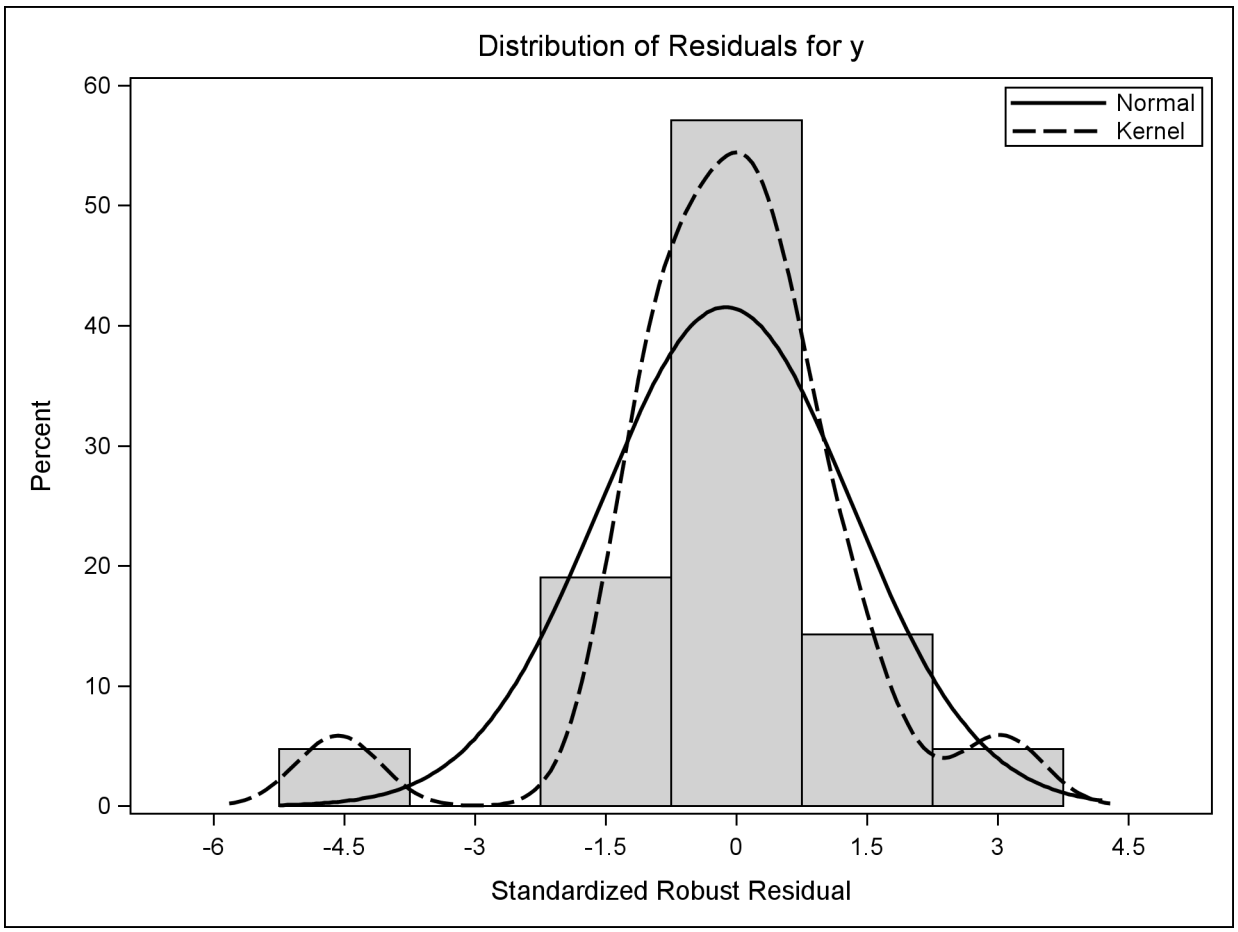
```
data stack;
   input  x1 x2 x3 y @@;
   datalines;
80  27  89  42    80  27  88  37    75  25  90  37

... more lines ...

;
```

The following statements close the LISTING destination, open the LATEX destination with the JOURNAL style, and request a histogram of standardized robust residuals computed with PROC ROBUSTREG:

```
ods graphics on / reset=index;
ods listing close;
ods latex style=Journal;

proc robustreg data=stack plots=histogram;
   model y = x1 x2 x3;
run;

ods latex close;
ods listing;
```

The JOURNAL style displays gray-scale graphs that are suitable for a journal. When you specify the ODS LATEX destination, ODS creates a PostScript file for each individual graph in addition to a LaTeX source file that includes the tabular output and references to the PostScript files. By default, these files are saved in the SAS current folder. The histogram shown in Output 21.3.1 is saved by default in a file named *Histogram.ps*. See the section "Naming Graphics Image Files" on page 542 for details about how graphics image files are named. If both the LISTING destination (which is open unless you explicitly close it) and the LATEX destination are open, then two files are created: *Histogram.ps* and *Histogram1.ps*. If the RESET=INDEX option is not specified in the ODS GRAPHICS statement, and if you run the step more than once, the final name is based on an incremented index.

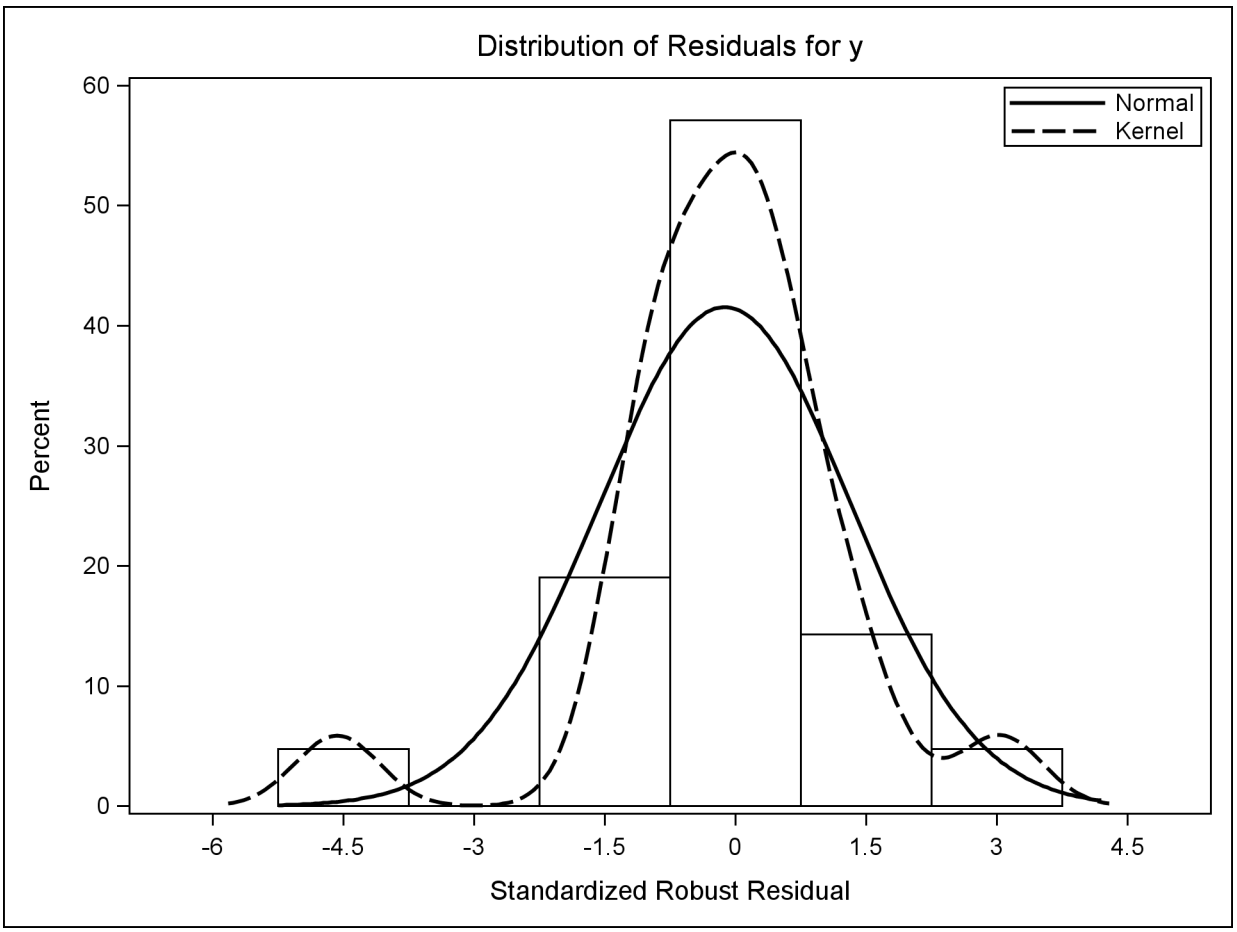
Output 21.3.1 Histogram Using the JOURNAL Style



You can use the JOURNAL2 style for a different appearance—the bars are not filled. The following step produces Output 21.3.2:

```
ods graphics on / reset=index;
ods listing close;
ods latex style=Journal2;

proc robustreg data=stack plots=histogram;
   model y = x1 x2 x3;
run;

ods latex close;
ods listing;
```

Output 21.3.2 Histogram Using the JOURNAL2 Style



If you are writing a paper, you can include the graphs in your own LaTeX source file by referencing the names of the individual PostScript graphics files. In this situation, you might not find it necessary to use the LaTeX source file created by SAS. Alternatively, you can include PNG files into a LaTeX document, after using some other ODS destination (such as HTML) to create the PNG files.

Example 21.4: Displaying Graphs Using the DOCUMENT Procedure

This example illustrates the use of the ODS DOCUMENT destination and the DOCUMENT procedure to display your ODS graphs. You can use this approach whenever you want to generate and save your output (both tables and graphs) and then display or replay it later, potentially in subsets or more than once. This approach is particularly useful when you want to display your output in multiple ODS destinations, or when you want to use different styles without rerunning your SAS program. This approach is also useful when you want to break your output into separate parts for inclusion into different parts of a document such as a LaTeX file.

Consider again the data set Stack created by the following statements:

```
data stack;
   input  x1 x2 x3 y @@;
   datalines;
80  27  89  42    80  27  88  37    75  25  90  37

... more lines ...

;
```

The following statements request a Q-Q plot from PROC ROBUSTREG with the Stack data:

```
ods listing close;
ods document name=QQDoc(write);

proc robustreg data=stack plots=qqplot;
   model y = x1 x2 x3;
run; quit;

ods document close;
ods listing;
```

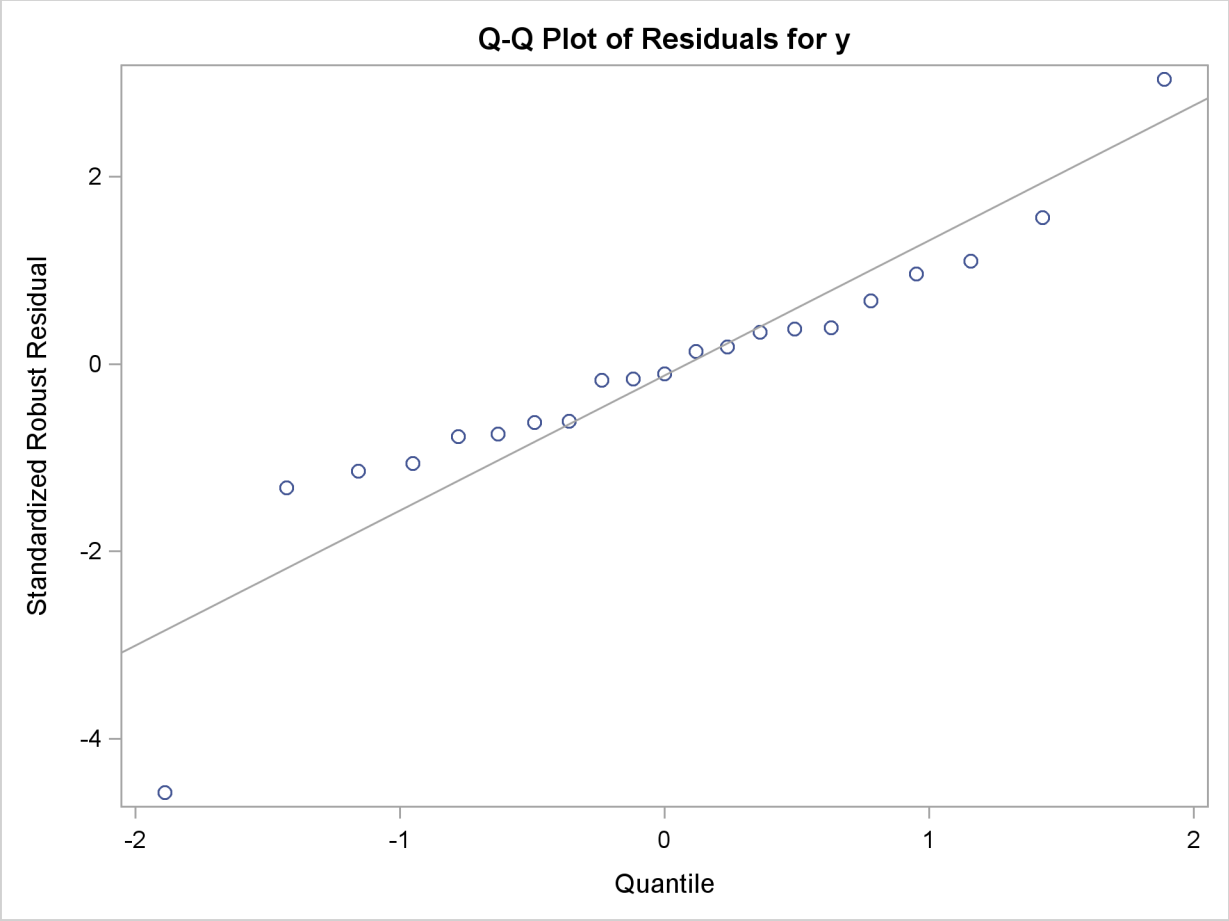
The ODS DOCUMENT statement opens an ODS document named **QQDoc**. All of the results—tables, graphs, titles, notes, footnotes, headers—are stored in the ODS document. None of them are displayed since no other destination is open. In order to display the Q-Q plot with PROC DOCUMENT, you first need to determine its name. You can do this by specifying the ODS TRACE ON statement prior to the procedure statements (see the section "Determining Graph Names and Labels" on page 538 for more information). Alternatively, you can type **odsdocuments** (or **odsd** for short) in the command line to open the Documents window, which you can then use to manage your ODS documents.

The following statements specify an HTML destination and display the residual Q-Q plot by using the REPLAY statement in PROC DOCUMENT:

```
ods html body='b.htm';

proc document name=QQDoc;
   ods select QQPlot;
   replay;
run; quit;

ods html close;
```

Subsequent steps can replay one or more objects from the same ODS document. By default, the REPLAY statement attempts to display every output object stored in the ODS document, but here only the Q-Q plot is displayed because it is specified by the ODS SELECT statement. The plot is displayed in Output 21.4.1.

Output 21.4.1 Q-Q Plot Displayed by PROC DOCUMENT



As an alternative to running PROC DOCUMENT with an ODS SELECT statement, you can run PROC DOCUMENT with a *document path* for the Q-Q plot in the REPLAY statement. This approach is preferable when the ODS document contains a large volume of output, so that PROC DOCUMENT does not attempt to process every piece of output stored in the ODS document.

You can determine the ODS document path for the Q-Q plot by specifying the LIST statement with the LEVELS=ALL option in PROC DOCUMENT as follows:

```
proc document name=QQDoc;
   list / levels=all;
run; quit;
```

The contents of the ODS document **QQDoc** are shown in Output 21.4.2.

Output 21.4.2 Contents of the ODS Document **QQDoc**

```
Listing of: \Work.Qqdoc\
Order by: Insertion
Number of levels: All

  Obs    Path                                                          Type
-------------------------------------------------------------------------------
       1 \Robustreg#1                                                  Dir
       2 \Robustreg#1\ModelInfo#1                                      Table
       3 \Robustreg#1\NObs#1                                           Table
       4 \Robustreg#1\ParmInfo#1                                       Table
       5 \Robustreg#1\SummaryStatistics#1                              Table
       6 \Robustreg#1\ParameterEstimates#1                             Table
       7 \Robustreg#1\DiagSummary#1                                    Table
       8 \Robustreg#1\DiagnosticPlots#1                                Dir
       9 \Robustreg#1\DiagnosticPlots#1\QQPlot#1                       Graph
      10 \Robustreg#1\GoodFit#1                                        Table
```

The ODS document path of the **QQPlot** entry in the **QQDoc** ODS document, as shown in Output 21.4.2, is **\Robustreg#1\DiagnosticPlots#1\QQPlot#1**.

You can use this path to display the residual Q-Q plot with PROC DOCUMENT as follows:

```
proc document name=QQDoc;
   replay \Robustreg#1\DiagnosticPlots#1\QQPlot#1;
run; quit;
```

You can also determine the ODS document path from the Results window or the Documents window. Right-click the object icon and select **Properties**.

The SAS/STAT documentation process uses the ODS document. SAS output is saved into an ODS document that is then replayed into sections of the documentation, which is prepared using LaTeX. In general, when you send your output to the DOCUMENT destination, you can use PROC DOCUMENT to rearrange, duplicate, or remove output from the results of a procedure or a database query. For more information, see the ODS DOCUMENT statement in the section "Dictionary of ODS Language Statements" and the chapter "The DOCUMENT Procedure" in the *SAS Output Delivery System: User's Guide*.

Example 21.5: Customizing Graphs Through Template Changes

This example shows how to use PROC TEMPLATE to customize the appearance and content of an ODS graph. It is divided into several parts; each part illustrates a different aspect of the template that you can easily change. You are never required to change a template, but you can if you want to change aspects of the plot.

Modifying Graph Titles and Axis Labels

This section illustrates the discussion in the section "Graph Templates" on page 577 in the context of changing the default title and Y-axis label for a Q-Q plot created with PROC ROBUSTREG. Consider again the data set Stack created by the following statements:

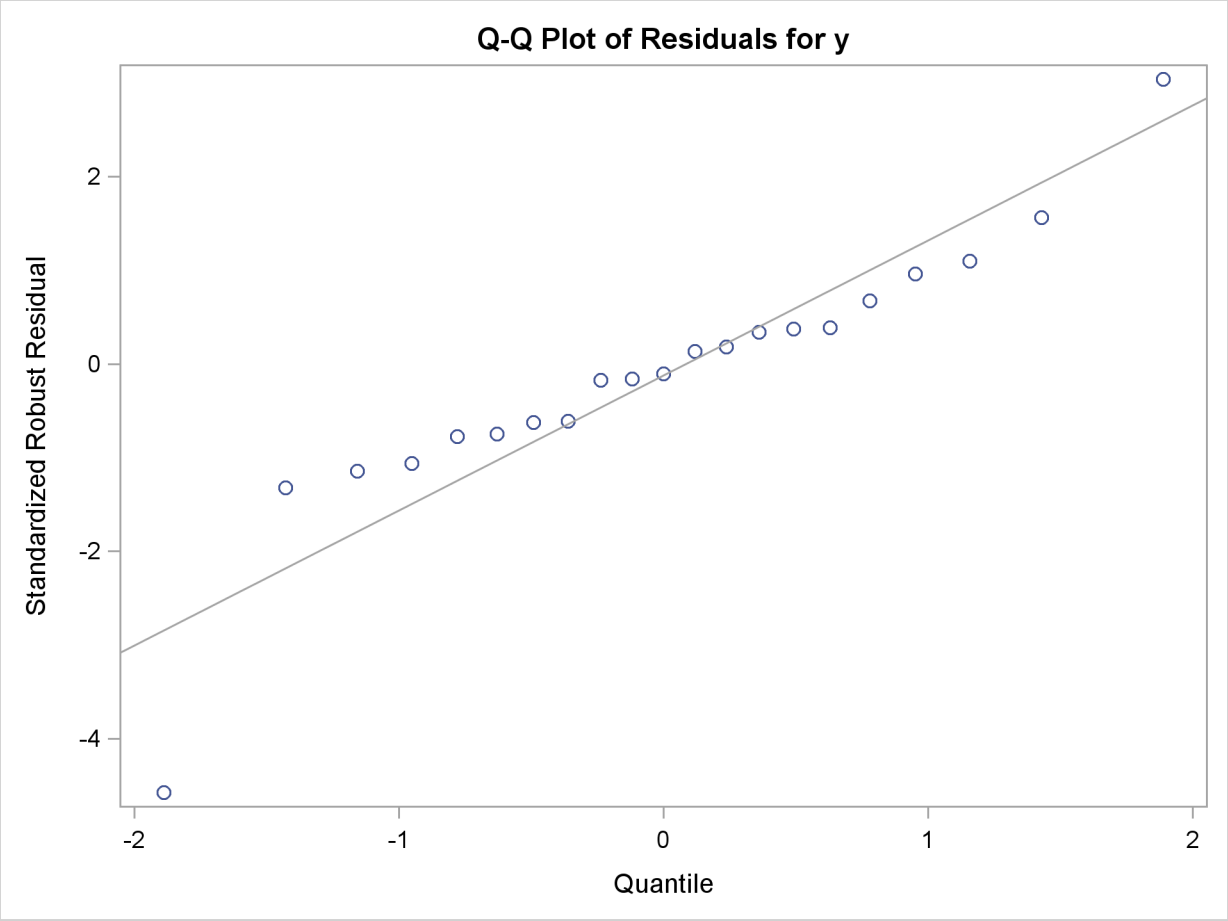
```
data stack;
   input  x1 x2 x3 y @@;
   datalines;
80  27  89  42    80  27  88  37    75  25  90  37

... more lines ...

;
```

The following statements request a Q-Q plot for robust residuals created by PROC ROBUSTREG:

```
ods trace on;
ods graphics on;

proc robustreg data=stack plots=qqplot;
   ods select QQPlot;
   model y = x1 x2 x3;
run;

ods trace off;
```

The Q-Q plot is shown in Output 21.5.1.

Output 21.5.1 Default Q-Q Plot from PROC ROBUSTREG



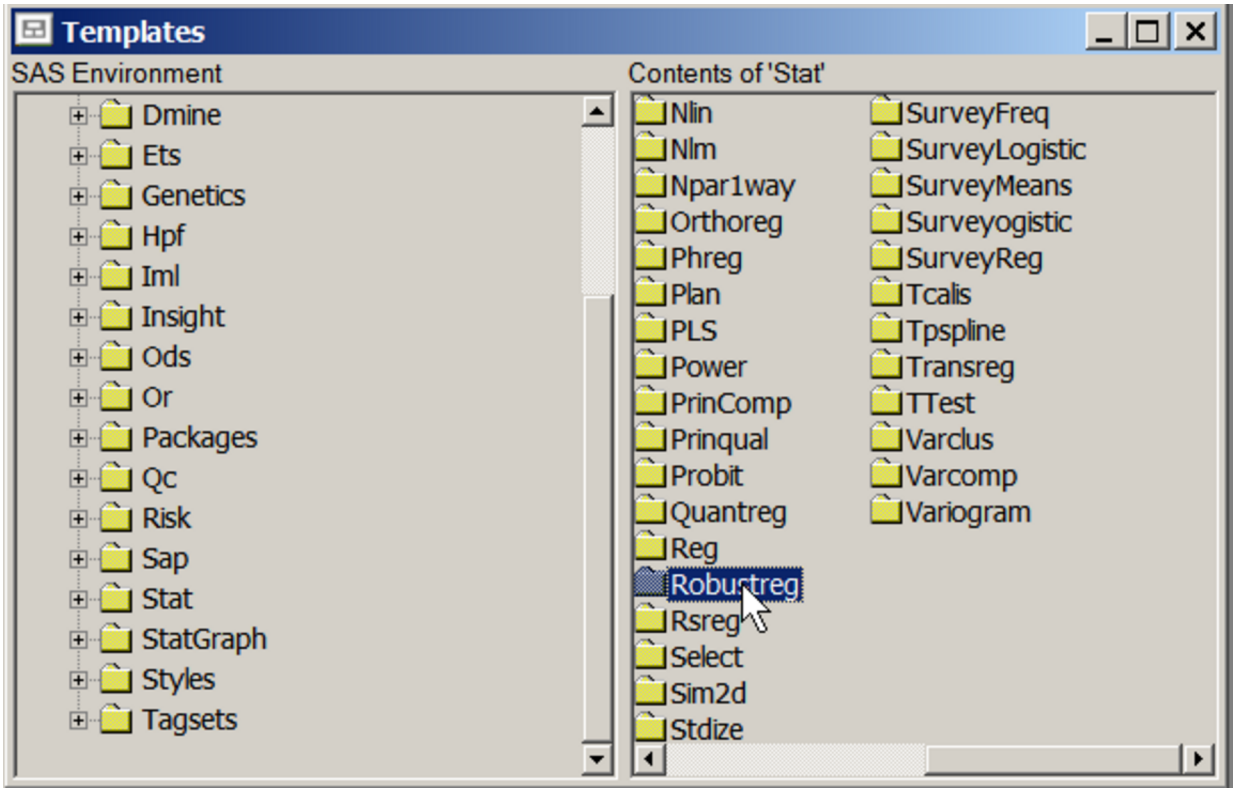
The ODS TRACE ON statement requests a record of all the ODS output objects created by PROC ROBUSTREG. The trace output is as follows:

```
Output Added:
-------------
Name:        QQPlot
Label:       Residual Q-Q Plot
Template:    Stat.Robustreg.Graphics.QQPlot
Path:        Robustreg.DiagnosticPlots.QQPlot
-------------
```

ODS Graphics creates the Q-Q plot from an ODS data object named **QQPlot** and a graph template named **Stat.Robustreg.Graphics.QQPlot**, which is the default template provided by SAS. Default templates supplied by SAS are saved in the SASHELP.Tmplmst template library (see the section "Graph Templates" on page 577).

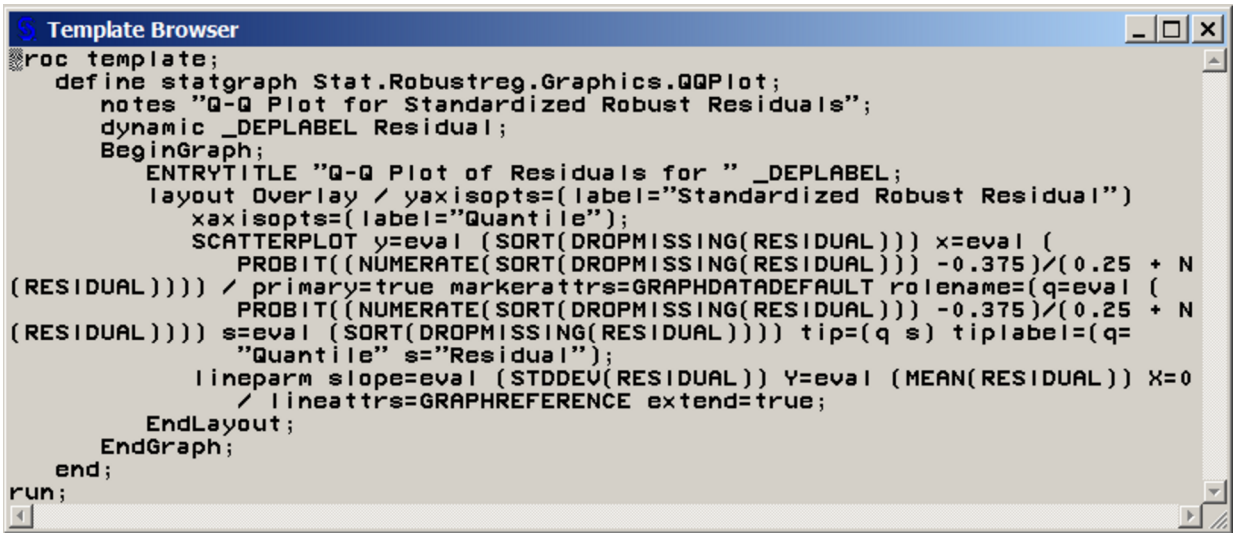
To display the default template definition, open the Templates window by typing **odstemplates** (or **odst** for short) in the command line. Expand SASHELP.Tmplmst and click the **Stat** folder. Output 21.5.2 shows the contents of the **Stat** folder.

Output 21.5.2 The Template Window



Next, open the **Robustreg** folder and then open the **Graphics** folder. Then right-click the `QQPlot` template icon and select **Open**. This opens the Template Browser window shown in Output 21.5.3. You can copy this template to an editor to edit it.

Output 21.5.3 Default Template Definition for Q-Q Plot



Alternatively, you can submit the following statements to display the **QQPlot** template definition in the SAS log:

```
proc template;
   source Stat.Robustreg.Graphics.QQPlot;
run;
```

The SOURCE statement specifies the fully qualified template name. You can copy and paste the template source into the Program Editor and modify it. The template, with a PROC TEMPLATE and RUN statement added, is shown next:

```
proc template;
   define statgraph Stat.Robustreg.Graphics.QQPlot;
      notes "Q-Q Plot for Standardized Robust Residuals";
      dynamic _DEPLABEL Residual;
      BeginGraph;
         ENTRYTITLE "Q-Q Plot of Residuals for " _DEPLABEL;
         layout Overlay / yaxisopts=(label="Standardized Robust Residual")
            xaxisopts=(label="Quantile");
            SCATTERPLOT y=eval (SORT(DROPMISSING(RESIDUAL))) x=eval (
               PROBIT((NUMERATE(SORT(DROPMISSING(RESIDUAL))) -0.375)/(0.25
               + N(RESIDUAL)))) / primary=true markerattrs=GRAPHDATADEFAULT
               rolename=(q=eval (
               PROBIT((NUMERATE(SORT(DROPMISSING(RESIDUAL))) -0.375)/(0.25
               + N(RESIDUAL)))) s=eval (SORT(DROPMISSING(RESIDUAL))))
               tip=(q s) tiplabel=(q="Quantile" s="Residual");
            lineparm slope=eval (STDDEV(RESIDUAL)) Y=eval (MEAN(RESIDUAL))
               X=0 / lineattrs=GRAPHREFERENCE extend=true;
         EndLayout;
      EndGraph;
   end;
run;
```

In the template, the default title of the Q-Q plot is specified by the ENTRYTITLE statement. The variable _DepLabel is a dynamic variable that provides the name of the dependent variable in the regression analysis. In this case, the name is y. In this template, the label for the axes are specified by the LABEL= suboption of the YAXISOPTS= option for the LAYOUT OVERLAY statement. In other templates, the axis labels come from the column labels of the X-axis and Y-axis columns of the data object. You can see these labels by specifying ODS OUTPUT with the plot data object and running PROC CONTENTS with the resulting SAS data set.

Suppose you want to change the default title to "Analysis of Residuals", and you want the Y-axis label to display the name of the dependent variable. First, replace the ENTRYTITLE statement with the following statement:

```
entrytitle "Analysis of Residuals";
```

Next, replace the LABEL= suboption with the following:

```
label=("Standardized Robust Residual for " _DEPLABEL)
```

You can use dynamic text variables such as _DepLabel in any text element.

You can then submit the modified template definition as you would any SAS program, for example, by selecting **Submit** from the **Run** menu. After submitting the PROC TEMPLATE statements, you should see the following message in the SAS log:

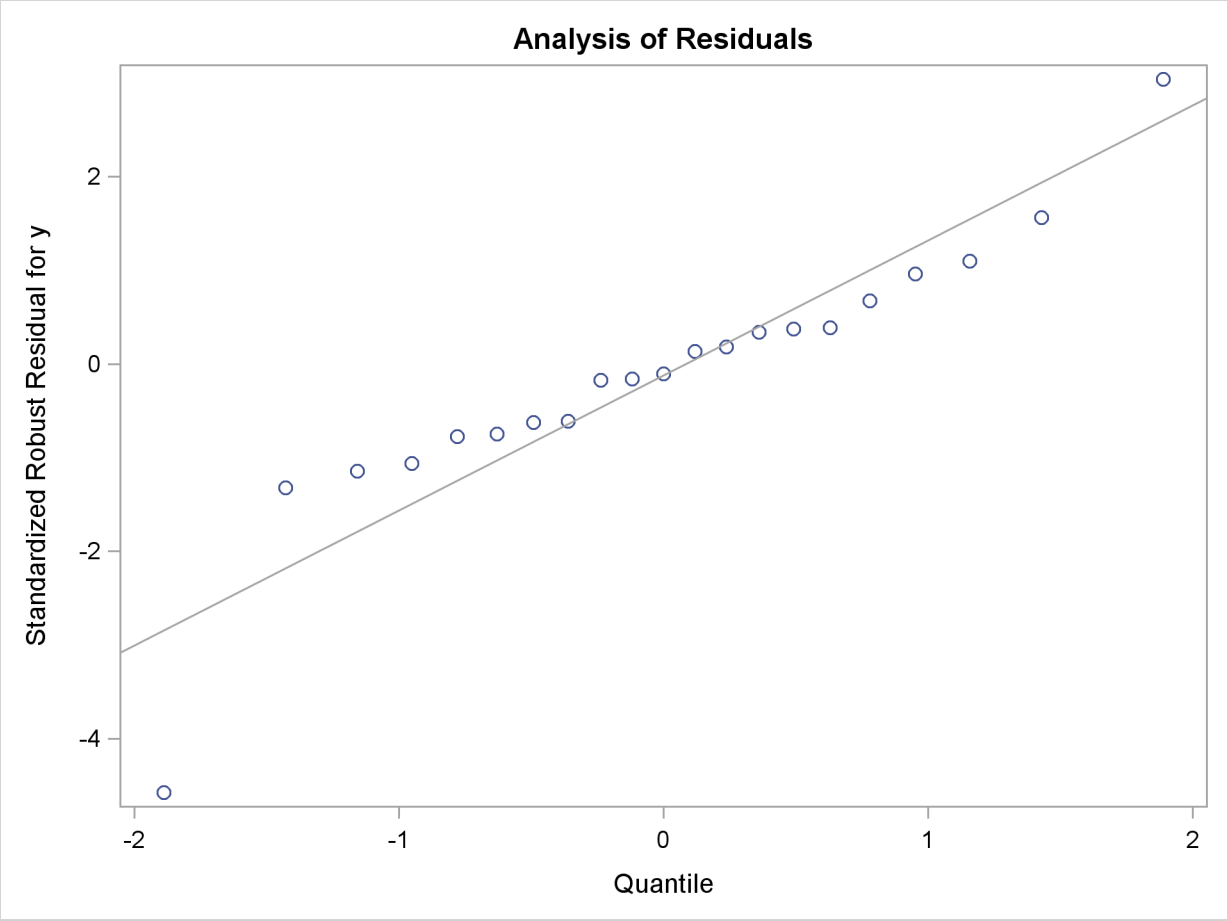
```
NOTE: STATGRAPH 'Stat.Robustreg.Graphics.QQPlot' has been
  saved to: SASUSER.TEMPLAT
```

For more information about graph definitions and the graph template language, see the section "Graph Templates" on page 577.

Finally, resubmit the PROC ROBUSTREG statements to display the Q-Q plot created with your modified template. The following statements create Output 21.5.4:

```
proc template;
   define statgraph Stat.Robustreg.Graphics.QQPlot;
      notes "Q-Q Plot for Standardized Robust Residuals";
      dynamic _DEPLABEL Residual;
      BeginGraph;
         entrytitle "Analysis of Residuals";
         layout Overlay /
            yaxisopts=(label=("Standardized Robust Residual for " _DEPLABEL))
            xaxisopts=(label="Quantile");
            SCATTERPLOT y=eval (SORT(DROPMISSING(RESIDUAL))) x=eval (
               PROBIT((NUMERATE(SORT(DROPMISSING(RESIDUAL))) -0.375)/(0.25
               + N(RESIDUAL)))) / primary=true markerattrs=GRAPHDATADEFAULT
               rolename=(q=eval (
               PROBIT((NUMERATE(SORT(DROPMISSING(RESIDUAL))) -0.375)/(0.25
               + N(RESIDUAL)))) s=eval (SORT(DROPMISSING(RESIDUAL))))
               tip=(q s) tiplabel=(q="Quantile" s="Residual");
            lineparm slope=eval (STDDEV(RESIDUAL)) Y=eval (MEAN(RESIDUAL))
               X=0 / lineattrs=GRAPHREFERENCE extend=true;
         EndLayout;
      EndGraph;
   end;
run;

proc robustreg data=stack plots=qqplot;
   ods select QQPlot;
   model y = x1 x2 x3;
run;
```

Output 21.5.4 Q-Q Plot with Modified Title and Y-Axis Label



If you have not changed the default ODS path, the modified template **QQPlot** is used automatically because SASUSER.Templat occurs before SASHELP.Tmplmst in the ODS search path. See the sections "Saving Customized Templates" on page 587, "Using Customized Templates" on page 587, and "Reverting to the Default Templates" on page 588 for more information about the ODS PATH.

You do not need to rerun the PROC ROBUSTREG analysis after you modify a graph template if you have stored the plot in an ODS document. After you modify your template, you can submit the PROC DOCUMENT statements in Example 21.4 to replay the Q-Q plot with the modified template. You can run the following statements to revert to the default template:

```
proc template;
   delete Stat.Robustreg.Graphics.QQPlot;
run;
```

Modifying Colors, Line Styles, and Markers

This section shows you how to customize colors, line attributes, and marker symbol attributes by modifying a graph template. In the **QQPlot** template definition shown in Output 21.5.3, the SCATTERPLOT statement specifies a scatter plot of normal quantiles versus ordered stan-

dardized residuals. The attributes of the marker symbol in the scatter plot are specified by: **MarkerAttrs=GraphDataDefault**. This is a reference to the style element **GraphDataDefault**. See the section "Style Elements and Attributes" on page 555 for more information.

The actual value of the marker symbol depends on the style that you are using. In this case, since the STATISTICAL style is used, the marker symbol is a circle. You can specify a filled circle as the marker symbol by overriding the symbol portion of the style specification as follows: **MarkerAttrs=GraphDataDefault(symbol=CircleFilled)**.

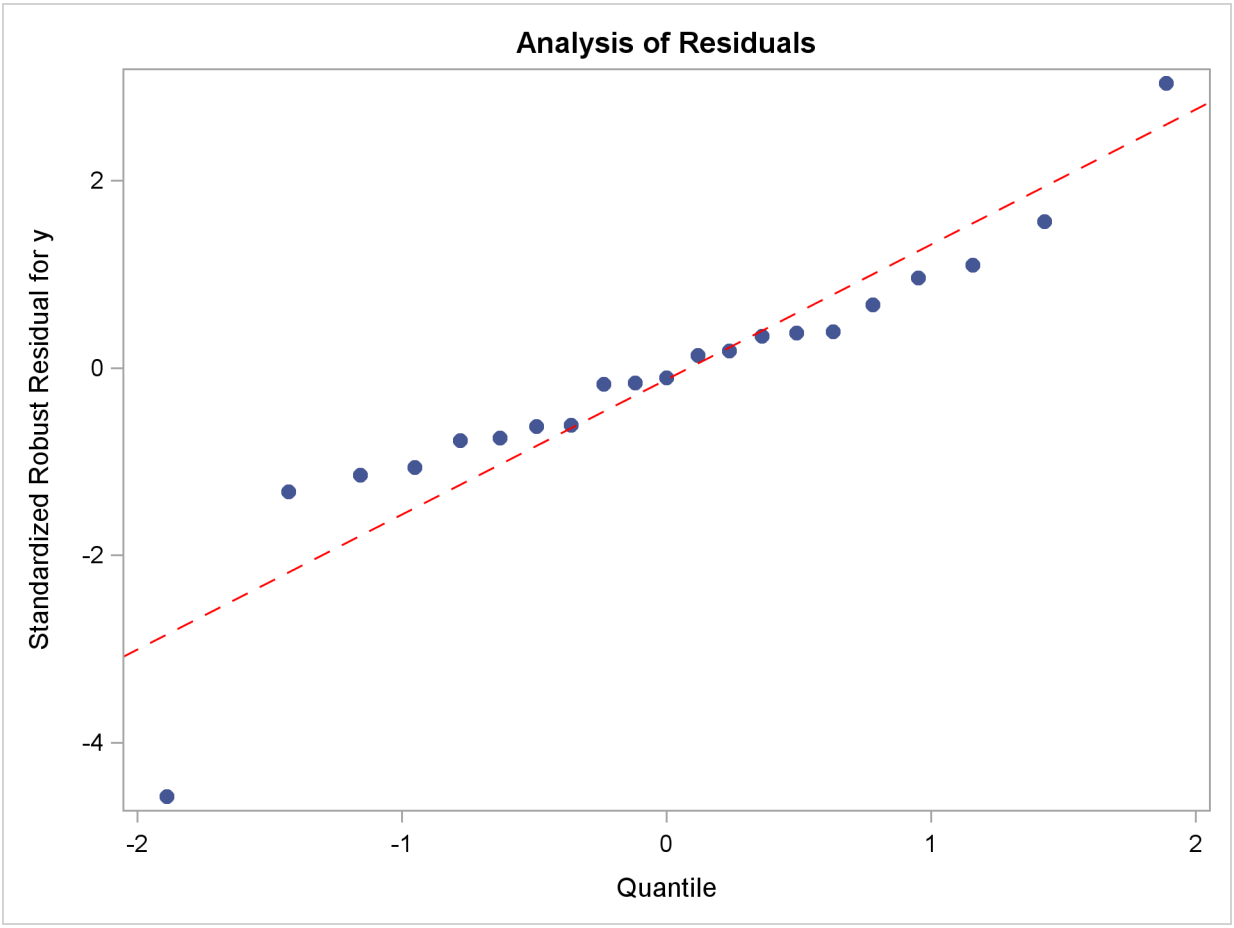
The value of the SYMBOL= option can be any valid marker symbol or a reference to a style attribute of the form **style-element:attribute**. It is recommended that you use style attributes because they are chosen to provide consistency and appropriate emphasis based on display principles for statistical graphics. If you specify values directly in a template, you are overriding the style and you run the risk of creating a graph that is inconsistent with the style definition. For more information about the syntax of the Graph Template Language and style elements for graphics, see the *SAS/GRAPH: Graph Template Language Reference* and the *SAS Output Delivery System: User's Guide*.

Similarly, you can change the line color and pattern with the LINEATTRS= option in the LINEPARM statement. The LINEPARM statement displays a straight line specified by slope and intercept parameters. The following option changes the color of the line to red and the line pattern to dashed, by overriding those aspects of the style specification: **LineAttrs=GraphReference(color=red pattern=dash)**. To see the results, submit the modified template definition and the PROC ROBUSTREG statements as follows to create Output 21.5.5:

```
proc template;
   define statgraph Stat.Robustreg.Graphics.QQPlot;
      notes "Q-Q Plot for Standardized Robust Residuals";
      dynamic _DEPLABEL Residual;
      BeginGraph;
         entrytitle "Analysis of Residuals";
         layout Overlay /
            yaxisopts=(label=("Standardized Robust Residual for " _DEPLABEL))
            xaxisopts=(label="Quantile");
            SCATTERPLOT y=eval (SORT(DROPMISSING(RESIDUAL))) x=eval(
               PROBIT((NUMERATE(SORT(DROPMISSING(RESIDUAL))) -0.375)/(0.25
               + N(RESIDUAL)))) / primary=true
               markerattrs=GraphDataDefault(symbol=CircleFilled)
               rolename=(q=eval (
               PROBIT((NUMERATE(SORT(DROPMISSING(RESIDUAL))) -0.375)/(0.25
               + N(RESIDUAL)))) s=eval (SORT(DROPMISSING(RESIDUAL))))
               tip=(q s) tiplabel=(q="Quantile" s="Residual");
            lineparm slope=eval (STDDEV(RESIDUAL)) Y=eval (MEAN(RESIDUAL))
               X=0 / lineattrs=GraphReference(color=red pattern=dash)
                     extend=true;
         EndLayout;
      EndGraph;
   end;
run;
```

```
ods graphics on;

proc robustreg data=stack plots=qqplot;
   ods select QQPlot;
   model y = x1 x2 x3;
run;
```

Output 21.5.5 Q-Q Plot with Modified Marker Symbols and Line



Alternatively, you can replay the plot with PROC DOCUMENT, as in Example 21.4.

Modifying Tick Marks and Grid Lines

This section illustrates how to modify axis tick marks and control grid lines. For example, you can specify the following statement to request tick marks ranging from –4 to 2 in the Y-axis:

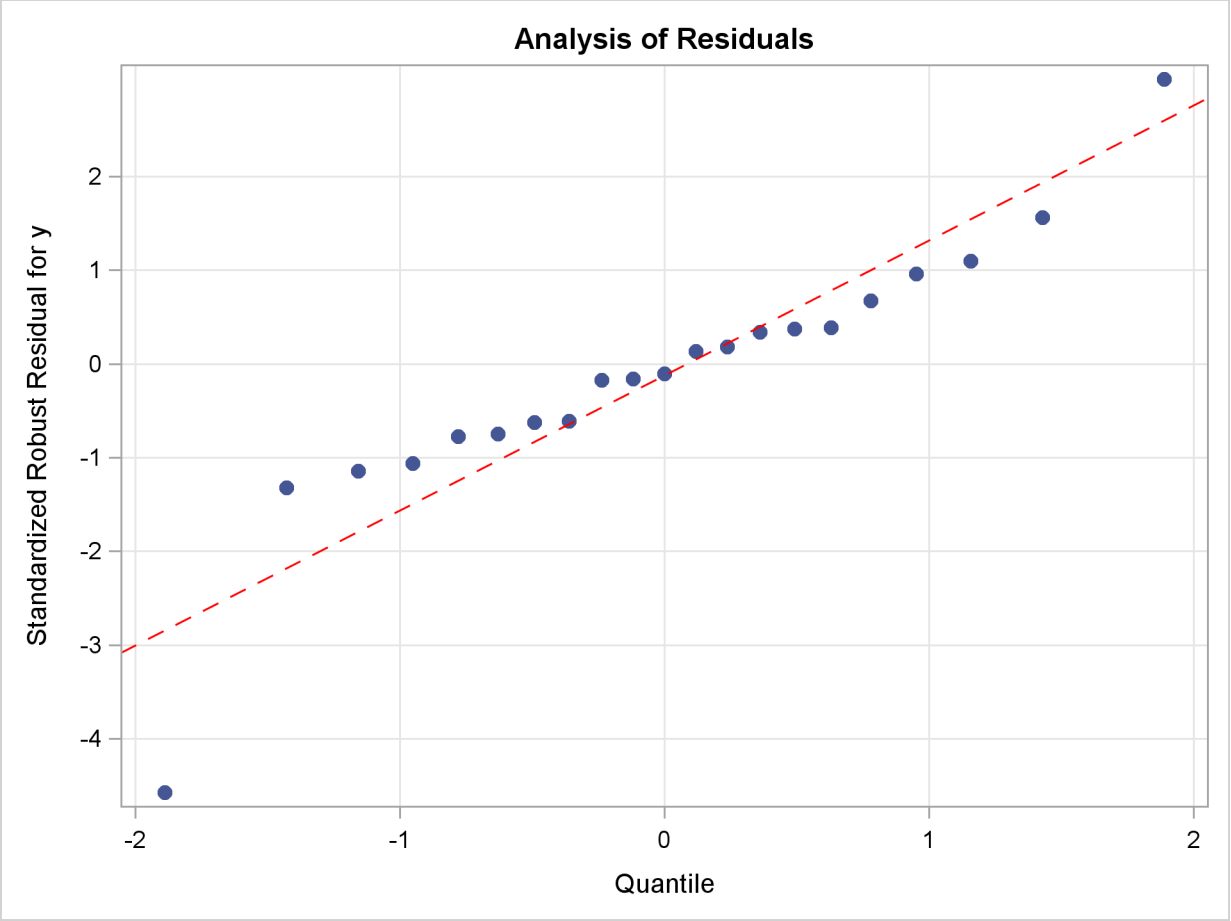
```
layout Overlay / yaxisopts=(linearopts=(tickvaluelist=(-4 -3 -2 -1 0 1 2)));
```

The LINEAROPTS= option is used for standard linearly scaled axes (as opposed to log-scaled axes). You use the TICKVALUELIST= to specify the tick marks.

You can control the grid lines by using the GRIDDISPLAY= suboption in the YAXISOPTS= option. Typically, you specify either GRIDDISPLAY=AUTO_OFF (grid lines are not displayed unless the **GraphGridLines** element in the current style contains **DisplayOpts="ON"**) or GRIDDISPLAY=AUTO_ON (grid lines are displayed unless the **GraphGridLines** element in the current style contains **DisplayOpts="OFF"**). Here, the template is modified by specifying GRIDDISPLAY=AUTO_ON for both axes. The following statements produce Output 21.5.6:

```
proc template;
   define statgraph Stat.Robustreg.Graphics.QQPlot;
      notes "Q-Q Plot for Standardized Robust Residuals";
      dynamic _DEPLABEL Residual;
      BeginGraph;
         entrytitle "Analysis of Residuals";
         layout Overlay / yaxisopts=(gridDisplay=Auto_On
            linearopts=(tickvaluelist=(-4 -3 -2 -1 0 1 2))
            label=("Standardized Robust Residual for " _DEPLABEL))
            xaxisopts=(gridDisplay=Auto_On label="Quantile");
            SCATTERPLOT y=eval (SORT(DROPMISSING(RESIDUAL))) x=eval(
               PROBIT((NUMERATE(SORT(DROPMISSING(RESIDUAL))) -0.375)/(0.25
               + N(RESIDUAL)))) / primary=true
               markerattrs=GraphDataDefault(symbol=CircleFilled)
               rolename=(q=eval (
               PROBIT((NUMERATE(SORT(DROPMISSING(RESIDUAL))) -0.375)/(0.25
               + N(RESIDUAL)))) s=eval (SORT(DROPMISSING(RESIDUAL))))
               tip=(q s) tiplabel=(q="Quantile" s="Residual");
            lineparm slope=eval (STDDEV(RESIDUAL)) Y=eval (MEAN(RESIDUAL))
               X=0 / lineattrs=GraphReference(color=red pattern=dash)
                     extend=true;
         EndLayout;
      EndGraph;
   end;
run;

ods graphics on;

proc robustreg data=stack plots=qqplot;
   ods select QQPlot;
   model y = x1 x2 x3;
run;
```

Output 21.5.6 Q-Q Plot with Modified Y-Axis Tick Marks and Grids



You can restore the default template by running the following step:

```
proc template;
   delete Stat.Robustreg.Graphics.QQPlot;
run;
```

See the section "Modifying the Style to Show Grid Lines" on page 619 for more information about grid lines.

Modifying the Style to Show Grid Lines

The section "Modifying Tick Marks and Grid Lines" on page 618 explains that grid lines in graphs are controlled both by template options and by the style. Some graphs never display grid lines because they would interfere with the display. Some graphs always display grid lines because they are a critical part of the display. In both cases, grid control is so important that the template writer is not willing to give control to the style. If you want to change the grid display setting for these graphs, you must edit their templates. Most templates, however, let the style control the grid lines. They either do not display grid lines unless the style forces them on, or they display grid lines unless the style forces them off. The STATISTICAL, DEFAULT, and most other styles use

the setting **DisplayOpts = "Auto"**. Then templates that specify GRIDDISPLAY=AUTO_OFF (the default) do not display grid lines, and templates that specify GRIDDISPLAY=AUTO_ON do display grid lines. You can easily make a new style with **DisplayOpts = "On"** or **DisplayOpts = "Off"** if you would prefer to see grid lines more or less often. This example shows how to set **DisplayOpts = "On"**.

First, you need to find the style source for setting grid lines. The following step displays the STATISTICAL and DEFAULT styles:

```
proc template;
   source Styles.Statistical;
   source Styles.Default;
run;
```

The advantage of displaying both styles together is you can do one search of the results. If grids are defined in the STATISTICAL style, you will find that first. Otherwise, you will first find the definition in the DEFAULT style. An abridged version of the results follows:

```
. . .
class GraphGridLines /
   displayopts = "auto"
   linethickness = 1px
   linestyle = 1
   contrastcolor = GraphColors('ggrid')
   color = GraphColors('ggrid');
. . .
```

You can use this to create a new style that inherits from the STATISTICAL style, but sets the display options for grids to ON, as in the following example:

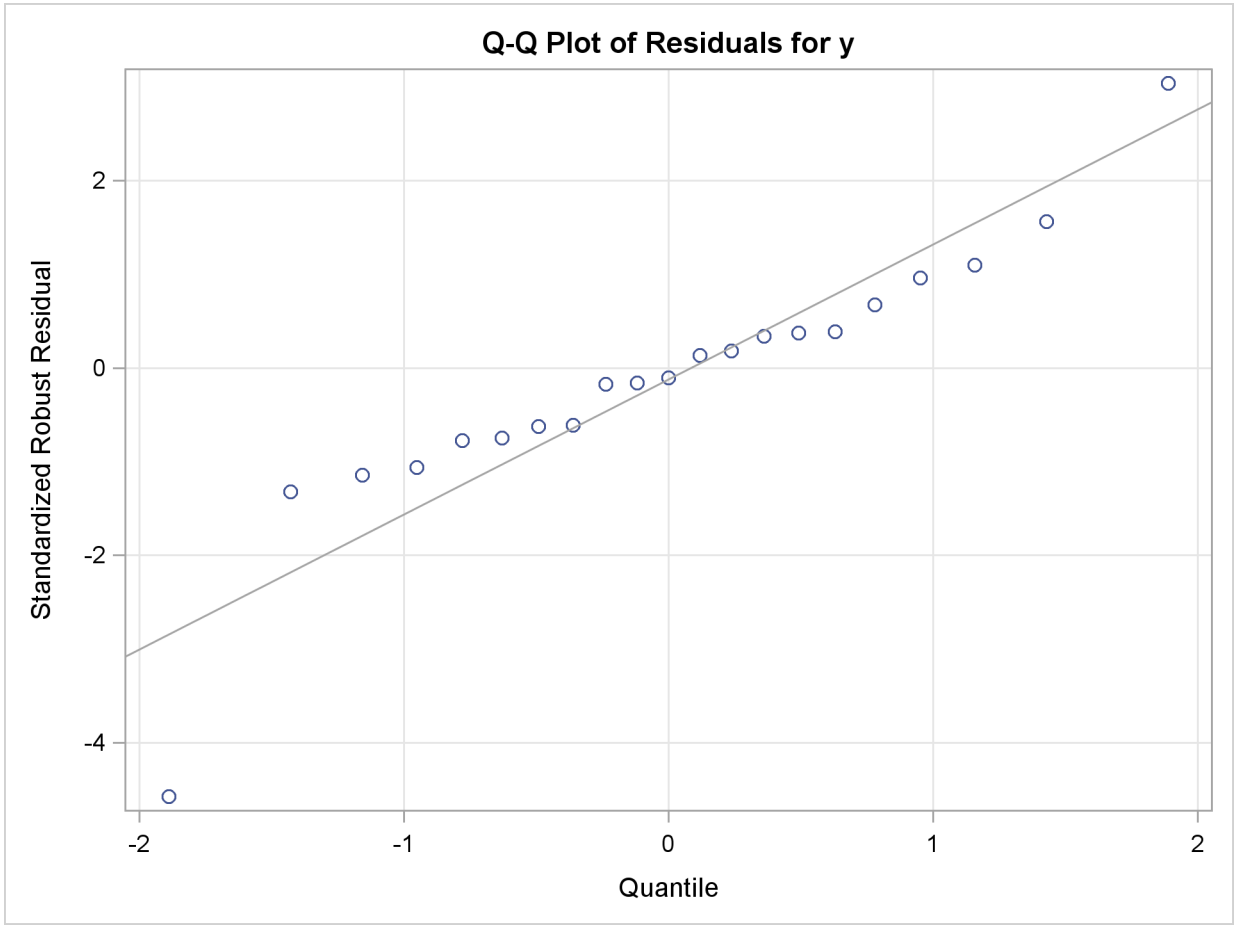
```
proc template;
   define style Styles.MyGrids;
      parent=styles.statistical;
      class GraphGridLines /
         displayopts = "on"
         linethickness = 1px
         linestyle = 1
         contrastcolor = GraphColors('ggrid')
         color = GraphColors('ggrid');
   end;
run;
```

You can use this new style as in the following example:

```
ods graphics on;
ods listing style=mygrids;

proc robustreg data=stack plots=qqplot;
   ods select QQPlot;
   model y = x1 x2 x3;
run;
```

The preceding statements produce Output 21.5.7, which shows the Q-Q plot with grid lines displayed. The default graph template, supplied by SAS, is used because the custom template created in the section "Modifying Tick Marks and Grid Lines" on page 618 is deleted at the end of that section.

Output 21.5.7 A Style that Makes Grid Lines the Typical Default



Example 21.6: Customizing Survival Plots

PROC LIFETEST, like other statistical procedures, provides a PLOTS= option and other options for modifying its output without requiring template changes. See Chapter 49, "The LIFETEST Procedure," for more information. Those options are sufficient for most purposes. This example shows ways that you can change the template when those options are not sufficient.

This example changes the default title of the survival plot in PROC LIFETEST from "Product-Limit Survival Estimate" to "Kaplan-Meier Plot" through a template change. Subsequent parts of this example change other aspects of the plot including the legend, inset table, ticks, and the overall layout—all through template changes.

This example uses the BMT data set from the section "Survival Estimate Plot with PROC LIFETEST" on page 504. The following steps create the SAS data set:

```
proc format;
   value risk 1='ALL' 2='AML-Low Risk' 3='AML-High Risk';
run;

data BMT;
   input Group T Status @@;
   format Group risk.;
   label T='Disease Free Time';
   datalines;
1 2081 0 1 1602 0 1 1496 0 1 1462 0 1 1433 0

... more lines ...

;
```

The following statements run PROC LIFETEST to determine the template name:

```
ods graphics on;
ods trace output;

proc lifetest data=BMT plots=survival(cb=hw test);
   time T * Status(0);
   strata Group / test=logrank;
run;

ods trace off;
```

The trace output results (not shown) show that the template name for the survival plot is `Stat.Lifetest.Graphics.ProductLimitSurvival`. The following statements display the template:

```
proc template;
   source Stat.Lifetest.Graphics.ProductLimitSurvival;
run;
```

Modifying the Plot Title

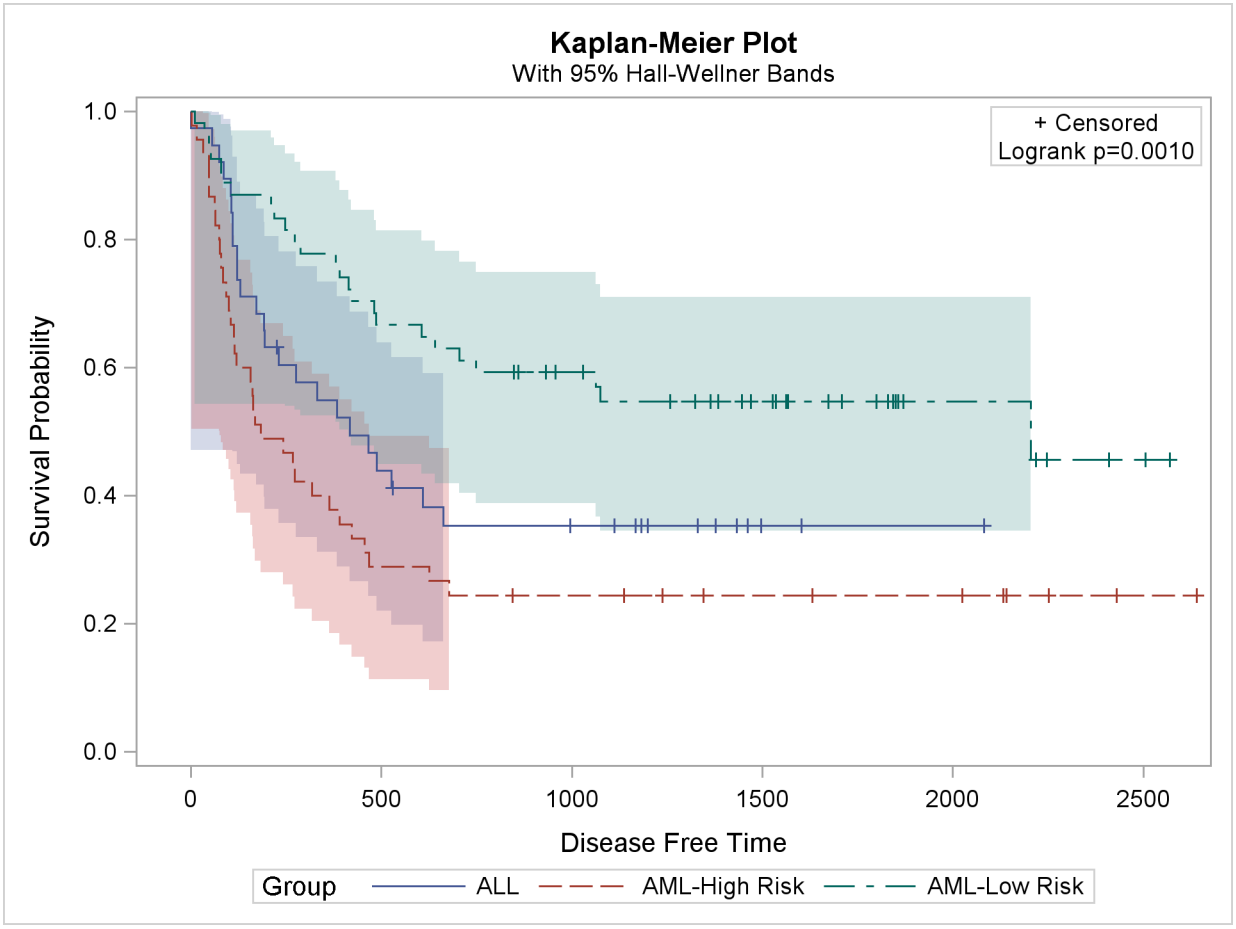
The template source listing is lengthy, but to change the title, you only need to find and modify the ENTRYTITLE statements. Many templates have conditional logic (IF statements, for example) and multiple overlays, so it is important that you find all of the relevant ENTRYTITLE statements.

The following abridged version of the statements show how you redefine the template:

```
proc template;
   define statgraph Stat.Lifetest.Graphics.ProductLimitSurvival;
      dynamic . . .;
      BeginGraph;
         if (NSTRATA=1)
            if (EXISTS(STRATUMID))
            entrytitle "Kaplan-Meier Plot for " STRATUMID;
         else
            entrytitle "Kaplan-Meier Plot";
         endif;
         if (PLOTATRISK)
            entrytitle "with Number of Subjects at Risk" / textattrs=
            GRAPHVALUETEXT;
         endif;
         layout overlay / xaxisopts=(shortlabel=XNAME offsetmin=.05
            linearopts=(viewmax=MAXTIME)) yaxisopts=(label=
            "Survival Probability" shortlabel="Survival" linearopts=(
            viewmin=0 viewmax=1 tickvaluelist=(0 .2 .4 .6 .8 1.0)));
            . . .
         endlayout;
         else
            entrytitle "Kaplan-Meier Plot";
         if (EXISTS(SECONDTITLE))
            entrytitle SECONDTITLE / textattrs=GRAPHVALUETEXT;
         endif;
         layout overlay / xaxisopts=(shortlabel=XNAME offsetmin=.05
            linearopts=(viewmax=MAXTIME)) yaxisopts=(label=
            "Survival Probability" shortlabel="Survival" linearopts=(
            viewmin=0 viewmax=1 tickvaluelist=(0 .2 .4 .6 .8 1.0)));
            . . .
         endlayout;
         endif;
      EndGraph;
   end;
run;

proc lifetest data=BMT plots=survival(cb=hw test);
   time T * Status(0);
   strata Group / test=logrank;
run;
```

The preceding steps create Output 21.6.1.

Output 21.6.1 Kaplan-Meier Plot



You can restore the default template by running the following step:

```
proc template;
   delete Stat.Lifetest.Graphics.ProductLimitSurvival;
run;
```

Modifying the Axes, Legend, and Inset Table

You can also change other aspects of this plot. The template options **linearopts=(viewmin=0 viewmax=1 tickvaluelist=(0 .2 .4 .6 .8 1.0))** control the minimum value displayed, the maximum value displayed, and the ticks. You can change the range of the vertical axis or the ticks by changing this option everywhere that it occurs. The following specification changes the ticks but not the range of values: **linearopts=(viewmin=0 viewmax=1 tickvaluelist=(0 .25 .5 .75 1.0))**.

There are many other features of this template that you can easily modify. For example, you can change the locations of the inset table and the legend, which are controlled by the following two statements, respectively:

```
layout gridded / rows=2 autoalign=(TOPRIGHT BOTTOMLEFT TOP BOTTOM)
                 border=true BackgroundColor=GraphWalls:Color Opaque=true;

DiscreteLegend "Survival" / title=GROUPNAME location=outside;
```

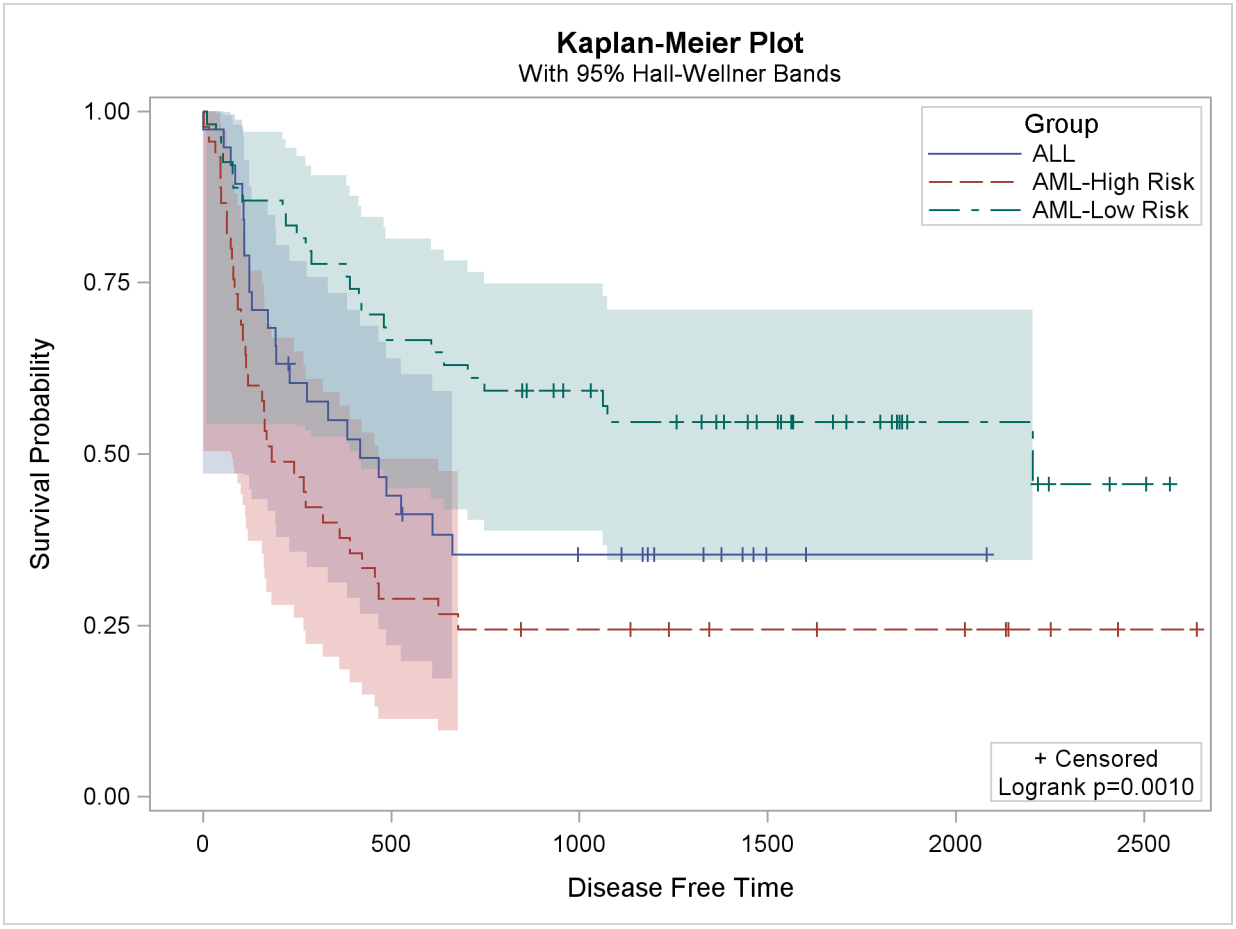
The LAYOUT GRIDDED statement produces the two-row inset table displayed in the top right corner. The AUTOALIGN= option provides the preferred locations inside the plot for this table, ordered from most preferred to least preferred. You can add new locations or rearrange the existing locations. The DISCRETELEGEND statement places the legend outside of the plot. You can move it inside and print only one legend entry across each row instead of three. This has the effect of changing the orientation of the legend from a row to a column. The modified statements are as follows:

```
layout gridded / rows=2
   autoalign=(BottomRight TOPRIGHT BOTTOMLEFT TOP BOTTOM)
    border=true BackgroundColor=GraphWalls:Color Opaque=true;

DiscreteLegend "Survival" / title=GROUPNAME across=1 location=inside
    autoalign=(TopRight BottomLeft Top Bottom);
```

The full template with these modifications is not shown here, but the new template along with the following statements produce Output 21.6.2:

```
proc lifetest data=BMT plots=survival(cb=hw test);
   time T * Status(0);
   strata Group / test=logrank;
run;
```

Output 21.6.2 Kaplan-Meier Plot with Legend Modifications



You can restore the default template by running the following step:

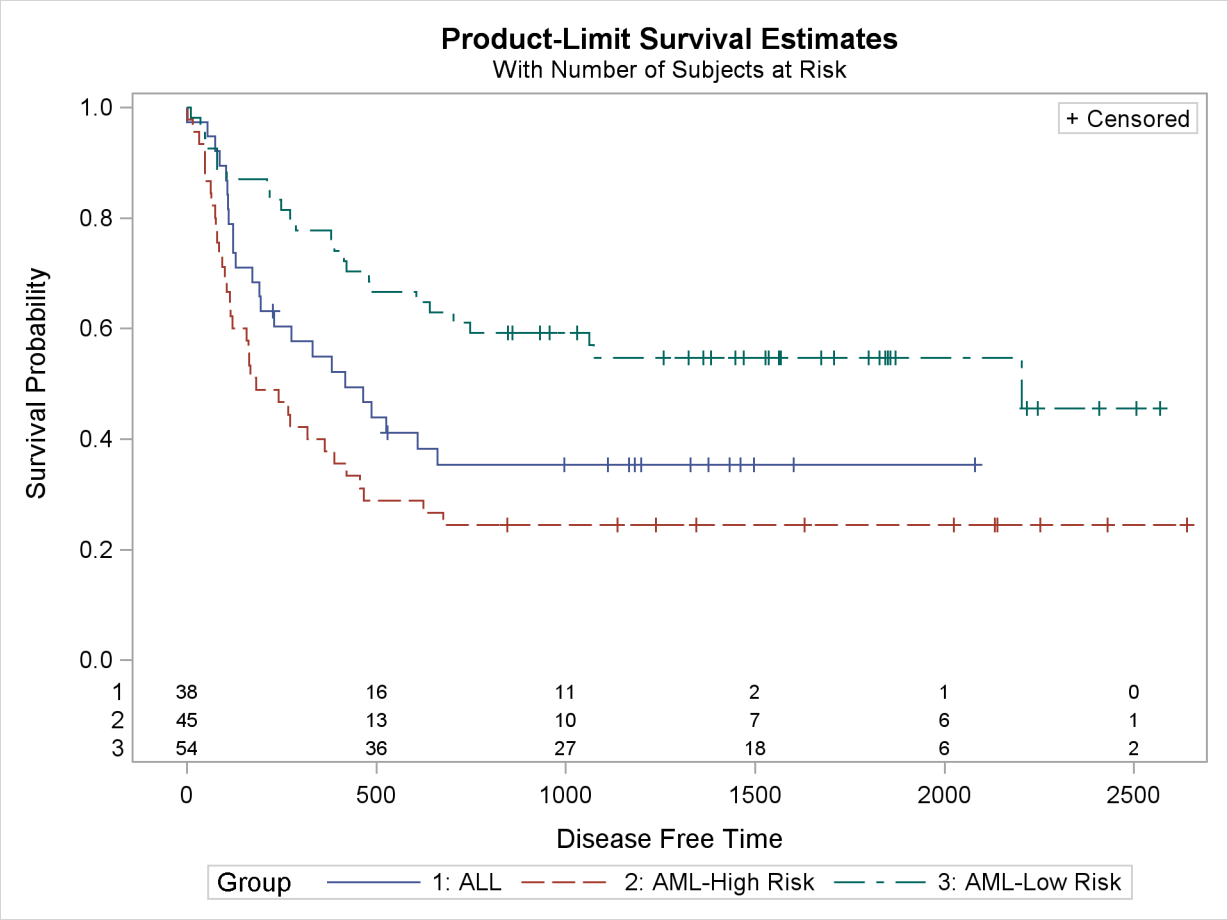
```
proc template;
   delete Stat.Lifetest.Graphics.ProductLimitSurvival;
run;
```

Modifying the Layout and Adding a New Inset Table

Example 49.2 of Chapter 49, "The LIFETEST Procedure," uses the following statements to make the plot shown in Output 21.6.3:

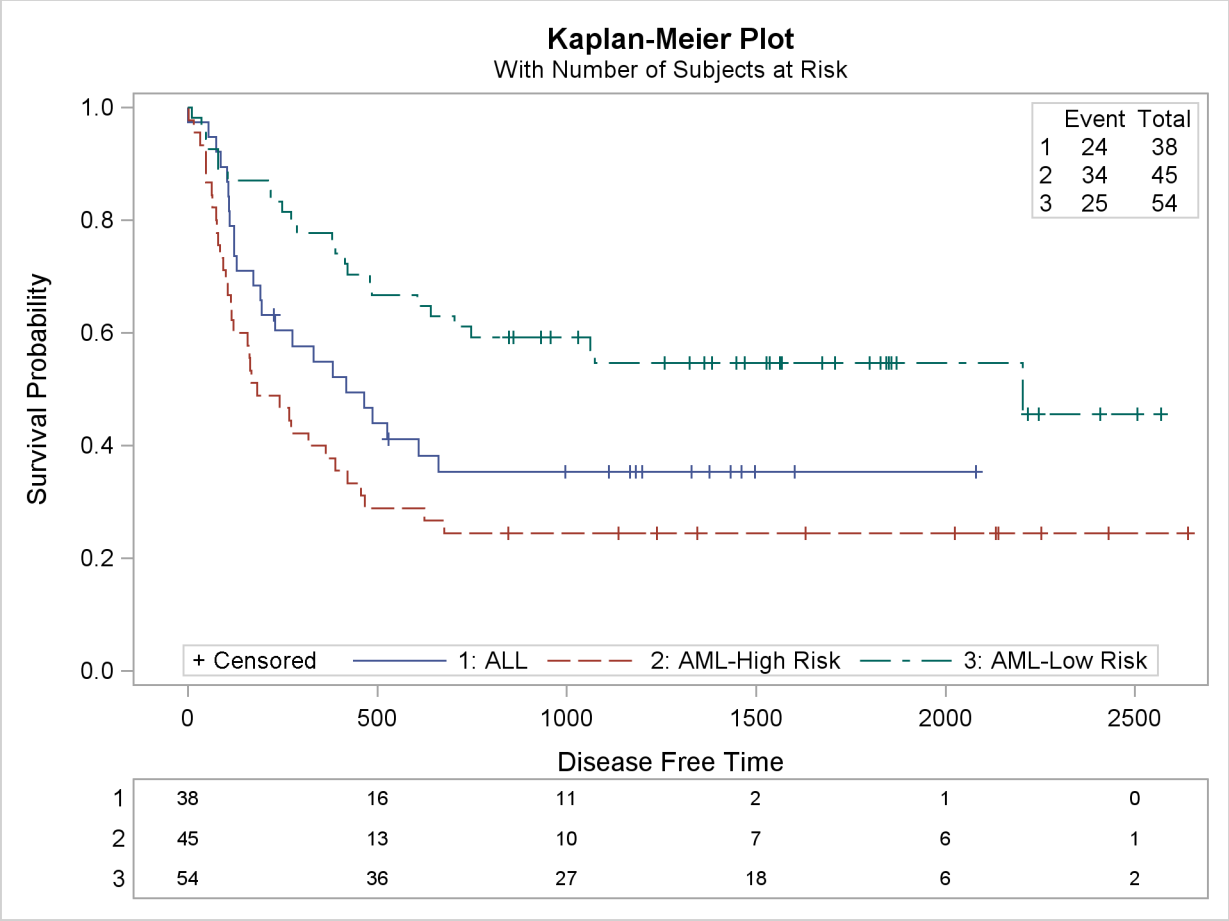
```
proc lifetest data=BMT plots=survival(atrisk=0 to 2500 by 500);
   ods select SurvivalPlot;
   time T * Status(0);
   strata Group / test=logrank adjust=sidak;
   run;
```

Output 21.6.3 Default Survival Plot with Number of Subjects At Risk



Output 21.6.3 displays the estimated disease-free survival functions for the three leukemia groups with the number of subjects at risk at 0, 500, 1000, 1500, 2000, and 2500 days. The rest of this example shows you how to modify the template to produce the plot displayed in Output 21.6.4. This new plot differs from the old plot in several ways. It has a new inset table in the top right corner with the number of observations and the number of events in the each stratum. The legend has been moved inside the plot and combined with the old inset table that showed the marker for censored observations. The information about the subjects at risk has been moved into a table below the plot. Also, the title change from the first part of the example has been retained.

Output 21.6.4 Kaplan-Meier Plot with a Different Layout



These changes are easy, if they are broken down and performed one step at a time. You can use the template with the new title from the beginning of this example as a starting point for these modifications. Before proceeding, you should notice the outermost layouts of this template, which are shown next:

```
proc template;
   define statgraph Stat.Lifetest.Graphics.ProductLimitSurvival;
      . . .
      BeginGraph;
         if (NSTRATA=1)
            . . .
         layout overlay / . . .;
            . . .
         endlayout;
         else
            . . .
         layout overlay / . . .;
            . . .
         endlayout;
         endif;
      EndGraph;
   end;
run;
```

This template consists of two major parts: a layout that is used when there is only one stratum and a layout that is used with more than one stratum. Every section of this example has more than one stratum, so it is the changes to the second layout (or more precisely the ELSE portion of the template) that are affecting the results.

PROC LIFETEST makes available a series of dynamic variables that it does not display by default. See the section "Additional Dynamic Variables for Survival Plots Using ODS Graphics" on page 3157 in Chapter 49, "The LIFETEST Procedure," for information about these dynamic variables. You can use these dynamic variables to add the new inset table to the plot. The following step creates the table in a gridded layout that is added to the second layout:

```
proc template;
   define statgraph Stat.Lifetest.Graphics.ProductLimitSurvival;
      . . .
      BeginGraph;
         if (NSTRATA=1)
            . . .
         layout overlay / . . .;
            . . .
         endlayout;
         else
            . . .
         layout overlay / . . .;
            . . .
            dynamic NObs1 NObs2 NObs3 NEvent1 NEvent2 NEvent3;
            layout gridded / columns=3 border=TRUE autoalign=(TopRight);
               entry "";      entry "Event";   entry "Total";
               entry "1";     entry NEvent1;   entry NObs1;
               entry "2";     entry NEvent2;   entry NObs2;
               entry "3";     entry NEvent3;   entry Nobs3;
            endlayout;
         endlayout;
         endif;
      EndGraph;
   end;
run;
```

The at-risk information in Output 21.6.3 is produced by a BLOCKPLOT statement in the second layout. The modified template, with the first layout removed (since it is not needed in this example) and the BLOCKPLOT statement displayed is as follows:

```
proc template;
   define statgraph Stat.Lifetest.Graphics.ProductLimitSurvival;
      . . .
      BeginGraph;
         layout overlay / . . .;
            . . .
            if (PLOTATRISK)
               innermargin / align=bottom;
               blockplot x=TATRISK block=ATRISK / class=CLASSATRISK
                  repeatedvalues=true display=(label values)
                  valuehalign=start valuefitpolicy=truncate
                  labelposition=left labelattrs=GRAPHVALUETEXT
                  valueattrs=GRAPHDATATEXT(size=7pt)
```

```
                  includemissingclass=false;
               endinnermargin;
               endif;
               . . .
               dynamic NObs1 NObs2 NObs3 NEvent1 NEvent2 NEvent3;
               layout gridded / columns=3 border=TRUE autoalign=(TopRight);
                  entry "";      entry "Event";   entry "Total";
                  entry "1";     entry NEvent1;   entry NObs1;
                  entry "2";     entry NEvent2;   entry NObs2;
                  entry "3";     entry NEvent3;   entry Nobs3;
               endlayout;
            endlayout;
         EndGraph;
      end;
   run;
```

In the next step, the at-risk information is moved out of the plot and into a table below the plot. The template is given a new overall layout—a LAYOUT LATTICE that has two panels stacked vertically, one for the plot and one for the at-risk information. Using ROWWEIGHTS=(.85 .15), the plot on top occupies 85% of the display and the at-risk information in the second panel occupies 15%. The option COLUMNDATARANGE=UNIONALL is used to create a common axis across the two panels. In these next steps, you also move the legend inside (similar to the previous part of this example) and rearrange the three inset boxes. The new template structure is as follows:

```
proc template;
   define statgraph Stat.Lifetest.Graphics.ProductLimitSurvival;
      . . .
      BeginGraph;
         . . .
         layout lattice /  rows=2 columns=1 columndatarange=unionall
                           rowweights=(.85 .15);

            layout overlay / . . .;
               . . .
               DiscreteLegend "Survival" / location=inside
                              autoalign=(BottomRight);
               . . .
               layout gridded / rows=1 autoalign=(BottomLeft)
                     border=true BackgroundColor=GraphWalls:Color
                     Opaque =true;
                  entry "+ Censored";
               endlayout;
               . . .
               dynamic NObs1 NObs2 NObs3 NEvent1 NEvent2 NEvent3;
               layout gridded / columns=3 border=TRUE autoalign=(TopRight);
                  entry "";      entry "Event";   entry "Total";
                  entry "1";     entry NEvent1;   entry NObs1;
                  entry "2";     entry NEvent2;   entry NObs2;
                  entry "3";     entry NEvent3;   entry Nobs3;
               endlayout;
            endlayout;

            layout overlay / xaxisopts=(display=none);
               blockplot x=TATRISK block=ATRISK / class=CLASSATRISK
```

```
                    repeatedvalues=true display=(label values)
                    valuehalign=start valuefitpolicy=truncate
                    labelposition=left labelattrs=GRAPHVALUETEXT
                    valueattrs=GRAPHDATATEXT(size=7pt)
                    includemissingclass=false;
            endlayout;
         endlayout;
      EndGraph;
   end;
run;
```

You can further simplify the plot by removing the title from the legend (which is currently the variable name Group) and instead adding "+ Censored" (the contents of the inset table) to the legend in place of the title, as in the following statement:

```
DiscreteLegend "Survival" / title="+ Censored"
   titleattrs=GraphValueText location=inside autoalign=(Bottom);
```

The option TITLEATTRS=GRAPHVALUETEXT is specified so that the "+ Censored" appears in the same font as the other entries in the legend and appears to be just another part of the legend. All of the statements for making the old inset table can now be removed from the template. The full template also plots bands, which are not used in this example, so they can be removed as well. The resulting template is as follows:

```
proc template;
   define statgraph Stat.Lifetest.Graphics.ProductLimitSurvival;

      dynamic NStrata xName plotAtRisk plotCensored plotCL plotHW plotEP
         labelCL labelHW labelEP maxTime StratumID classAtRisk plotBand
         plotTest GroupName yMin Transparency SecondTitle TestName pValue;

      BeginGraph;

         entrytitle "Kaplan-Meier Plot";
         if (EXISTS(SECONDTITLE))
            entrytitle SECONDTITLE / textattrs=GRAPHVALUETEXT;
         endif;

         layout lattice / rows=2 columns=1 columndatarange=unionall
                          rowweights=(.85 .15);

            layout overlay / xaxisopts=(shortlabel=XNAME offsetmin=.05
               linearopts=(viewmax=MAXTIME)) yaxisopts=(label=
               "Survival Probability" shortlabel="Survival" linearopts=(
               viewmin=0 viewmax=1 tickvaluelist=(0 .2 .4 .6 .8 1.0)));

               stepplot y=SURVIVAL x=TIME / group=STRATUM index=STRATUMNUM
                  name="Survival" rolename=(_tip1=ATRISK _tip2=EVENT) tip=(y
                  x Time _tip1 _tip2);

               if (PLOTCENSORED)
                  scatterplot y=CENSORED x=TIME / group=STRATUM index=
```

```
                  STRATUMNUM markerattrs=(symbol=plus);
               endif;

               DiscreteLegend "Survival" / title="+ Censored"
                  titleattrs=GraphValueText location=inside
                  autoalign=(BOTTOM);

               dynamic NObs1 NObs2 NObs3 NEvent1 NEvent2 NEvent3;
               layout gridded / columns=3 border=TRUE autoalign=(TopRight);
                  entry "";      entry "Event";   entry "Total";
                  entry "1";     entry NEvent1;   entry NObs1;
                  entry "2";     entry NEvent2;   entry NObs2;
                  entry "3";     entry NEvent3;   entry Nobs3;
               endlayout;
            endlayout;

            layout overlay / xaxisopts=(display=none);
               blockplot x=TATRISK block=ATRISK / class=CLASSATRISK
                     repeatedvalues=true display=(label values)
                     valuehalign=start valuefitpolicy=truncate
                     labelposition=left labelattrs=GRAPHVALUETEXT
                     valueattrs=GRAPHDATATEXT(size=7pt)
                     includemissingclass=false;
            endlayout;
         endlayout;
      EndGraph;
   end;
run;
```

The following step uses the new template to create the desired plot:

```
proc lifetest data=BMT plots=survival(atrisk=0 to 2500 by 500);
   ods select SurvivalPlot;
   time T * Status(0);
   strata Group / test=logrank adjust=sidak;
   run;
```

The plot is displayed in Output 21.6.4 at the beginning of this section.

This example removed a great deal of functionality from the default template, so that the final, modified template would be relatively simple and understandable. This is not necessary. The template could have been modified without deleting the first LAYOUT OVERLAY and other statements. The strategy for template modification illustrated in this example can be applied to any complicated template: identify the overall structure, isolate the relevant pieces, and then make changes in stages. Since the modified template will no longer work for all analyses, it is important that you delete it when you are done, as in the following example:

```
proc template;
   delete Stat.Lifetest.Graphics.ProductLimitSurvival;
run;
```

Example 21.7: Customizing Panels

This example illustrates how to modify the regression fit diagnostics panel shown in Figure 21.1 so that it displays a subset of the component plots. The original panel consists of eight plots and a summary statistics box. The ODS trace output from PROC REG shown previously shows that the template for the diagnostics panel is **Stat.REG.Graphics.DiagnosticsPanel**. The following statements display the template:

```
proc template;
   source Stat.REG.Graphics.DiagnosticsPanel;
run;
```

An abridged version of the results is shown next:

```
define statgraph Stat.Reg.Graphics.DiagnosticsPanel;
   notes "Diagnostics Panel";
   dynamic . . .;
   BeginGraph / designheight=defaultDesignWidth;
      entrytitle halign=left textattrs=GRAPHVALUETEXT _MODELLABEL
         halign=center textattrs=GRAPHTITLETEXT "Fit Diagnostics"
         " for " _DEPNAME;
      layout lattice / columns=3 rowgutter=10 columngutter=10
         shrinkfonts=true rows=3;
         layout overlay / xaxisopts=(shortlabel='Predicted');
            . . .
         endlayout;
         layout overlay / xaxisopts=(shortlabel='Predicted');
            . . .
         endlayout;
         layout overlay / xaxisopts=(label='Leverage' offsetmax=0.05)
            . . .
         endlayout;
         layout overlay / yaxisopts=(label="Residual" shortlabel=
            "Resid") xaxisopts=(label="Quantile");
            . . .
         endlayout;
         layout overlayequated / xaxisopts=(shortlabel='Predicted')
            . . .
         endlayout;
         layout overlay / xaxisopts=(linearopts=(integer=true) label=
            "Observation" shortlabel="Obs" offsetmax=0.05) yaxisopts=(
            offsetmin=0.05 offsetmax=0.05);
            . . .
         endlayout;
         layout overlay / xaxisopts=(label="Residual") yaxisopts=(label
            ="Percent");
            . . .
         endlayout;
         layout lattice / columns=2 rows=1 rowdatarange=unionall
            columngutter=0;
            . . .
         endlayout;
         if (_SHOWSTATS =1)
```

```
            layout overlay;
            . . .
         endLayout;
         endif;
         if (_SHOWSTATS = 2)
            layout overlay / yaxisopts=(gridDisplay=auto_off label=
            "Residual");
            . . .
         endlayout;
         endif;
      endlayout;
   EndGraph;
end;
```

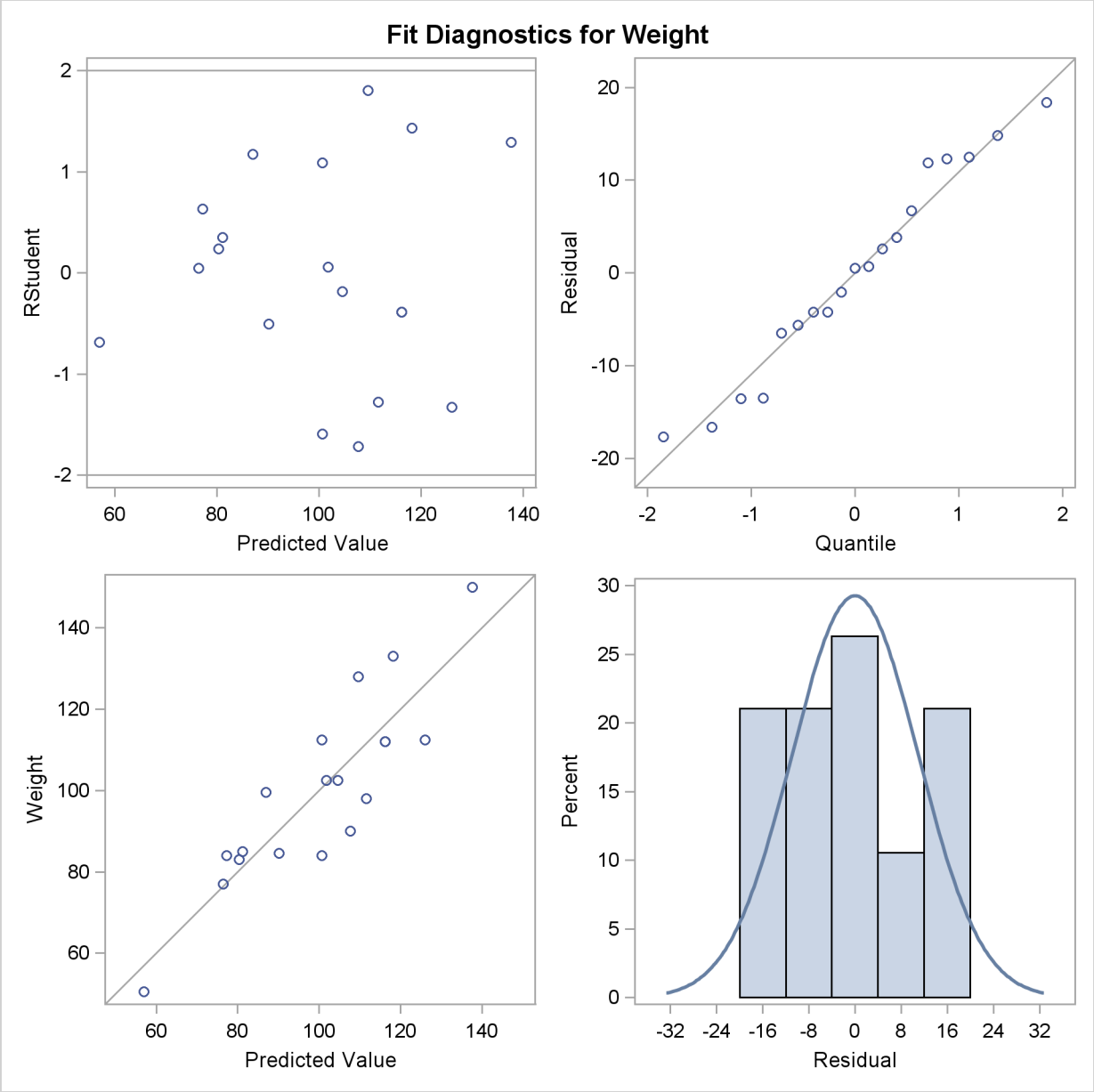
The outermost components of the template are a BEGINGRAPH/ENDGRAPH block with a lattice layout with ROWS=3 and COLUMNS=3 that defines the 3×3 panel of plots. Inside that are nine layouts, one for each cell, the last of which is conditionally defined. The LAYOUT statements define the components of the panel from left to right and top to bottom. You can eliminate some of the panels and produce a 2×2 panel as follows:

```
proc template;
   define statgraph Stat.Reg.Graphics.DiagnosticsPanel;
      notes "Diagnostics Panel";
      dynamic _DEPLABEL _DEPNAME _MODELLABEL _OUTLEVLABEL _TOTFREQ _NPARM
         _NOBS _OUTCOOKSDLABEL _SHOWSTATS _NSTATSCOLS _DATALABEL _SHOWNObs
         _SHOWTOTFREQ _SHOWNParm _SHOWEDF _SHOWMSE _SHOWRSquare
         _SHOWAdjRSq _SHOWSSE _SHOWDepMean _SHOWCV _SHOWAIC _SHOWBIC
         _SHOWCP _SHOWGMSEP _SHOWJP _SHOWPC _SHOWSBC _SHOWSP _EDF _MSE
         _RSquare _AdjRSq _SSE _DepMean _CV _AIC _BIC _CP _GMSEP _JP _PC
         _SBC _SP;
      BeginGraph / designheight=defaultDesignWidth;
         entrytitle halign=left textattrs=GRAPHVALUETEXT _MODELLABEL
            halign=center textattrs=GRAPHTITLETEXT "Fit Diagnostics"
            " for " _DEPNAME;
         layout lattice / columns=2 rowgutter=10 columngutter=10
            shrinkfonts=true rows=2;
            layout overlay / xaxisopts=(shortlabel='Predicted');
               referenceline y=-2;
               referenceline y=2;
               scatterplot y=RSTUDENT x=PREDICTEDVALUE / primary=true
                  datalabel=_OUTLEVLABEL rolename=(_tip1=OBSERVATION _id1=
                  ID1 _id2=ID2 _id3=ID3 _id4=ID4 _id5=ID5) tip=(y x _tip1
                  _id1 _id2 _id3 _id4 _id5);
            endlayout;
            layout overlay / yaxisopts=(label="Residual" shortlabel=
               "Resid") xaxisopts=(label="Quantile");
               lineparm slope=eval (STDDEV(RESIDUAL)) y=eval (
                  MEAN(RESIDUAL)) x=0 / extend=true lineattrs=
                  GRAPHREFERENCE;
               scatterplot y=eval (SORT(DROPMISSING(RESIDUAL))) x=eval (
                  PROBIT((NUMERATE(SORT(DROPMISSING(RESIDUAL))) -0.375)/
                  (0.25 + N(RESIDUAL)))) / markerattrs=GRAPHDATADEFAULT
                  primary=true
                  rolename=(s=eval (SORT(DROPMISSING(RESIDUAL))) nq=eval (
```

```
                PROBIT((NUMERATE(SORT(DROPMISSING(RESIDUAL))) -0.375)
                /(0.25 + N(RESIDUAL)))))) tiplabel=(nq="Quantile"
                s="Residual")
                tip=(nq s);
            endlayout;
            layout overlayequated / xaxisopts=(shortlabel='Predicted')
               yaxisopts=(label=_DEPLABEL shortlabel="Observed")
               equatetype=square;
               lineparm slope=1 x=0 y=0 / extend=true lineattrs=
                  GRAPHREFERENCE;
               scatterplot y=DEPVAR x=PREDICTEDVALUE / primary=true
                  datalabel=_OUTLEVLABEL rolename=(_tip1=OBSERVATION _id1=
                  ID1 _id2=ID2 _id3=ID3 _id4=ID4 _id5=ID5) tip=(y x _tip1
                  _id1 _id2 _id3 _id4 _id5);
            endlayout;
            layout overlay / xaxisopts=(label="Residual") yaxisopts=(label
               ="Percent");
               histogram RESIDUAL / primary=true;
               densityplot RESIDUAL / name="Normal" legendlabel="Normal"
                  lineattrs=GRAPHFIT;
            endlayout;
         endlayout;
      EndGraph;
   end;
run;

proc reg data=sashelp.class;
   model Weight = Height;
run; quit;
```

This template plots the residuals by predicted values, the Q-Q plot, the actual by predicted plot, and the residual histogram. The results are shown in Output 21.7.1.

Output 21.7.1 Diagnostics Panel with Four Plots



This new template is a straightforward modification of the original template. The COLUMNS=2 and ROWS=2 options in the LAYOUT LATTICE statement request a 2×2 lattice. The LAYOUT statement blocks for components 1, 3, 6, 8, and 9 are deleted. **NOTE:** You do not need to understand every aspect of a template to modify it if you can recognize the overall structure and a few key options.

You can restore the original template as follows:

```
proc template;
   delete Stat.REG.Graphics.DiagnosticsPanel;
run;
```

Example 21.8: Customizing Axes and Reference Lines

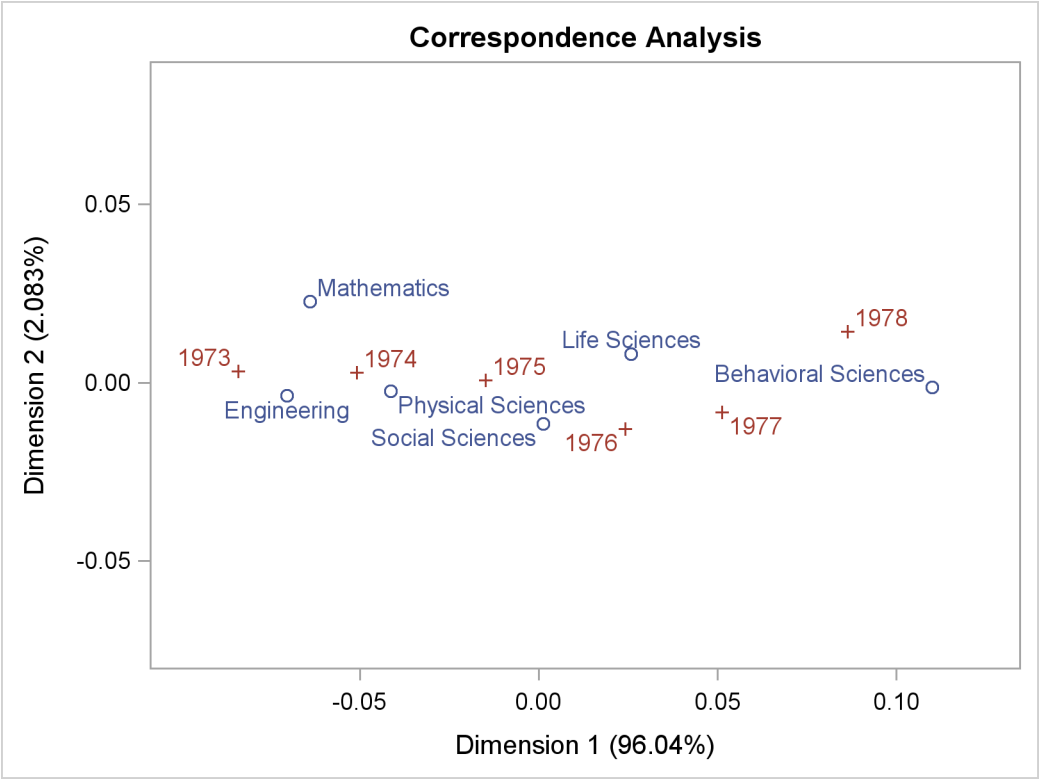
This example illustrates several ways that you can change the plot axes in a scatter plot. The example uses PROC CORRESP to perform a correspondence analysis. It is taken from the section "Getting Started: CORRESP Procedure" on page 1320 of Chapter 30, "The CORRESP Procedure." It uses the following data:

```
title "Number of Ph.D.'s Awarded from 1973 to 1978";

data PhD;
   input Science $ 1-19 y1973-y1978;
   label y1973 = '1973'
         y1974 = '1974'
         y1975 = '1975'
         y1976 = '1976'
         y1977 = '1977'
         y1978 = '1978';
   datalines;
Life Sciences       4489 4303 4402 4350 4266 4361
Physical Sciences   4101 3800 3749 3572 3410 3234
Social Sciences     3354 3286 3344 3278 3137 3008
Behavioral Sciences 2444 2587 2749 2878 2960 3049
Engineering         3338 3144 2959 2791 2641 2432
Mathematics         1222 1196 1149 1003  959  959
;
```

The following steps perform the correspondence analysis and create Output 21.8.1:

```
ods graphics on;
ods trace on;

proc corresp data=PhD short;
   ods select configplot;
   var y1973-y1978;
   id Science;
run;
```

Output 21.8.1 Default Scatter Plot



The trace output for this step (not shown) shows that the template for this plot is **Stat.Corresp.Graphics.Configuration**. The following step displays this template:

```
proc template;
   source Stat.Corresp.Graphics.Configuration;
run;
```

The results are as follows:

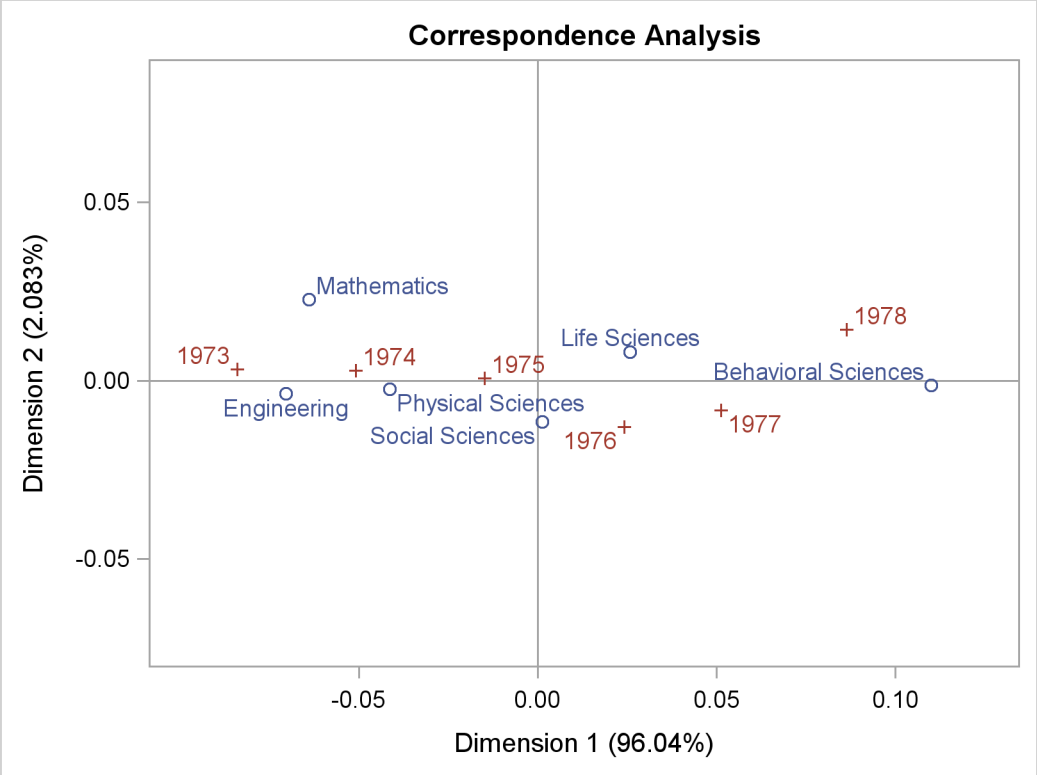
```
define statgraph Stat.Corresp.Graphics.Configuration;
   dynamic xVar yVar head legend;
   begingraph;
      entrytitle HEAD;
      layout overlayequated / equatetype=fit xaxisopts=(offsetmin=0.1
         offsetmax=0.1) yaxisopts=(offsetmin=0.1 offsetmax=0.1);
         scatterplot y=YVAR x=XVAR / group=GROUP index=INDEX
            datalabel=LABEL datalabelattrs=GRAPHVALUETEXT
            name="Type" tip=(y x datalabel group)
            tiplabel=(group="Point");
         if (LEGEND)
            discretelegend "Type";
         endif;
      endlayout;
   endgraph;
end;
```

You can add reference lines to the scatter plot at specified X and Y values by using the REFERENCELINE statement, as in the following example:

```
proc template;
   define statgraph Stat.Corresp.Graphics.Configuration;
      dynamic xVar yVar head legend;
      begingraph;
         entrytitle HEAD;
         layout overlayequated / equatetype=fit xaxisopts=(offsetmin=0.1
            offsetmax=0.1) yaxisopts=(offsetmin=0.1 offsetmax=0.1);

            referenceline x=0;
            referenceline y=0;

            scatterplot y=YVAR x=XVAR / group=GROUP index=INDEX
               datalabel=LABEL datalabelattrs=GRAPHVALUETEXT
               name="Type" tip=(y x datalabel group)
               tiplabel=(group="Point");
            if (LEGEND)
               discretelegend "Type";
            endif;
         endlayout;
      endgraph;
   end;
run;

proc corresp data=PhD short;
   ods select configplot;
   var y1973-y1978;
   id Science;
run;
```

When you modify templates, it is important to note that the order of the statements within the LAYOUT OVERLAYEQUATED (or more typically, the LAYOUT OVERLAY) is significant. Here, the reference lines are added before the scatter plot so that the reference lines are drawn before the scatter plot. Consequently, labels and markers that coincide with the reference lines are drawn over the reference lines. The results, with reference lines, are displayed in Output 21.8.2.

Output 21.8.2 Scatter Plot with Reference Lines Added



You can restore the default graph template as follows:

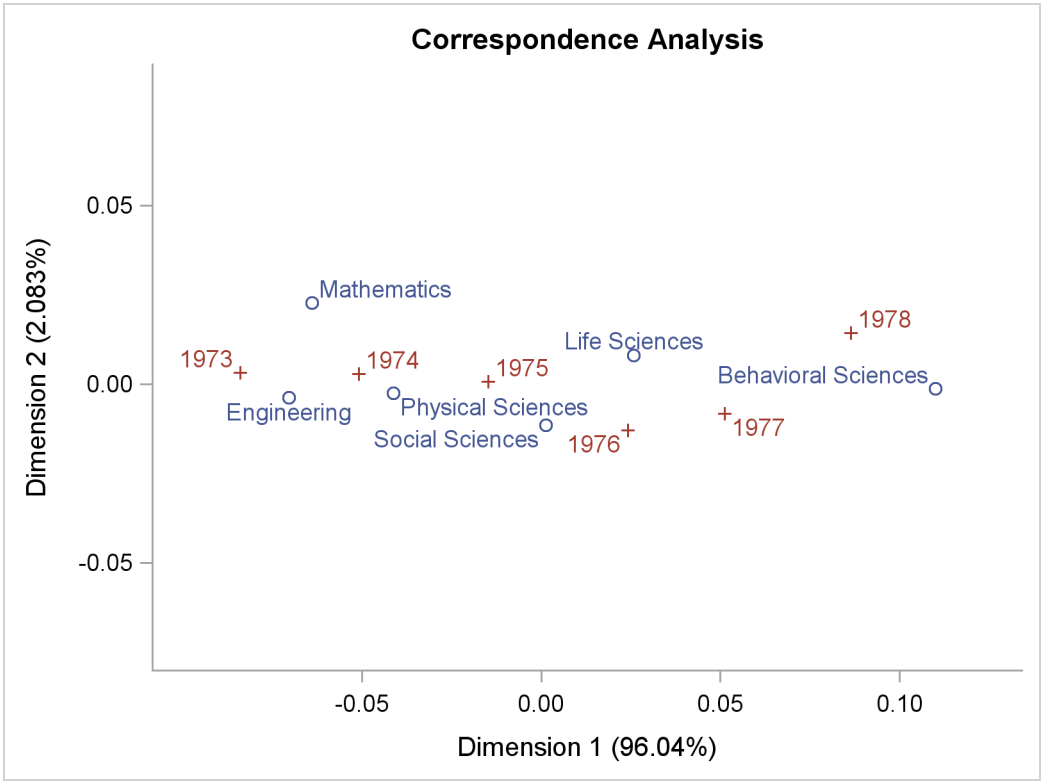
```
proc template;
   delete Stat.Corresp.Graphics.Configuration;
run;
```

The next steps show how you can change the style so that a frame is not shown:

```
proc template;
   define style noframe;
      parent=styles.statistical;
      style graphwalls from graphwalls / frameborder=off;
   end;
run;

ods listing style=noframe;

proc corresp data=PhD short;
   ods select configplot;
   var y1973-y1978;
   id Science;
run;
```

The results, shown in Output 21.8.3, display an X-axis and a Y-axis without a frame. Unlike the previous change, which affects only the **ConfigPlot** display, this change affects all plots created with the NOFRAME style.

Output 21.8.3 Scatter Plot with No Axis Frame



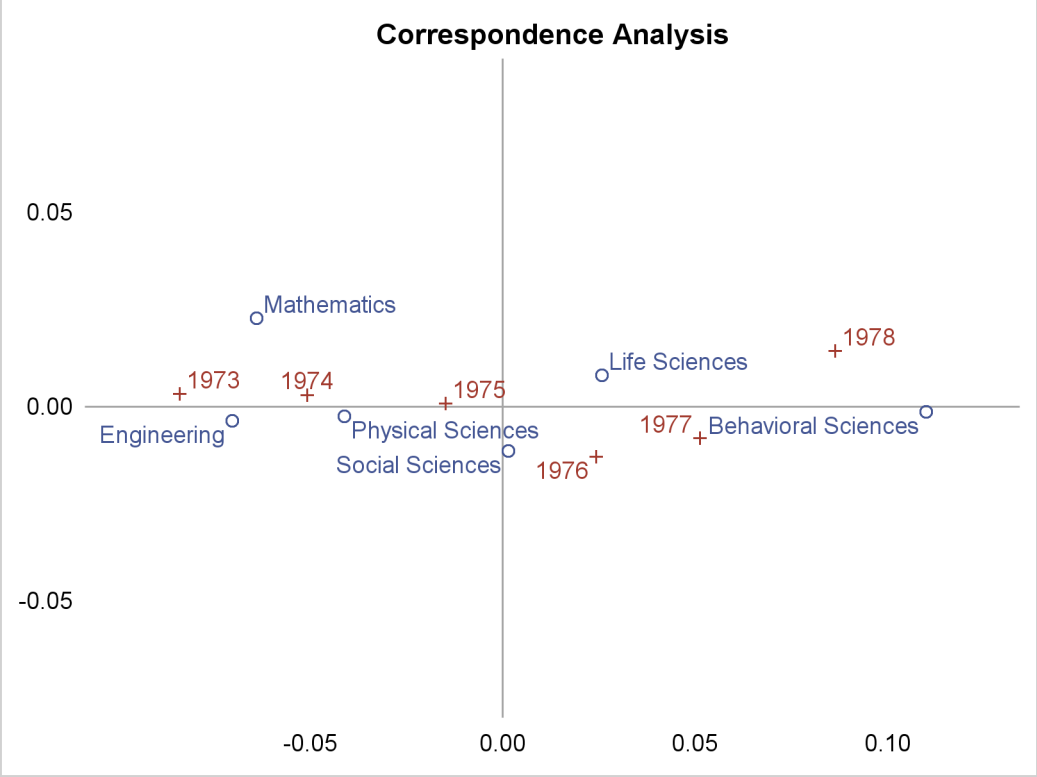
Alternatively, you can also add reference lines and delete the entire axis frame using the WALLDISPLAY=NONE and the DISPLAY= option in the graph template, as in the following example:

```
proc template;
   define statgraph Stat.Corresp.Graphics.Configuration;
      dynamic xVar yVar head legend;
      begingraph;
         entrytitle HEAD;

         layout overlayequated / equatetype=fit walldisplay=none
            xaxisopts=(display=(tickvalues) offsetmin=0.1 offsetmax=0.1)
            yaxisopts=(display=(tickvalues) offsetmin=0.1 offsetmax=0.1);

            referenceline x=0;
            referenceline y=0;

            scatterplot y=YVAR x=XVAR / group=GROUP index=INDEX
               datalabel=LABEL datalabelattrs=GRAPHVALUETEXT
               name="Type" tip=(y x datalabel group)
               tiplabel=(group="Point");
            if (LEGEND)
               discretelegend "Type";
            endif;
         endlayout;
      endgraph;
   end;
run;
```

```
ods listing style=statistical;

proc corresp data=PhD short;
   ods select configplot;
   var y1973-y1978;
   id Science;
run;
```

The results are shown in Output 21.8.4.

Output 21.8.4 Scatter Plot with Internal Axes



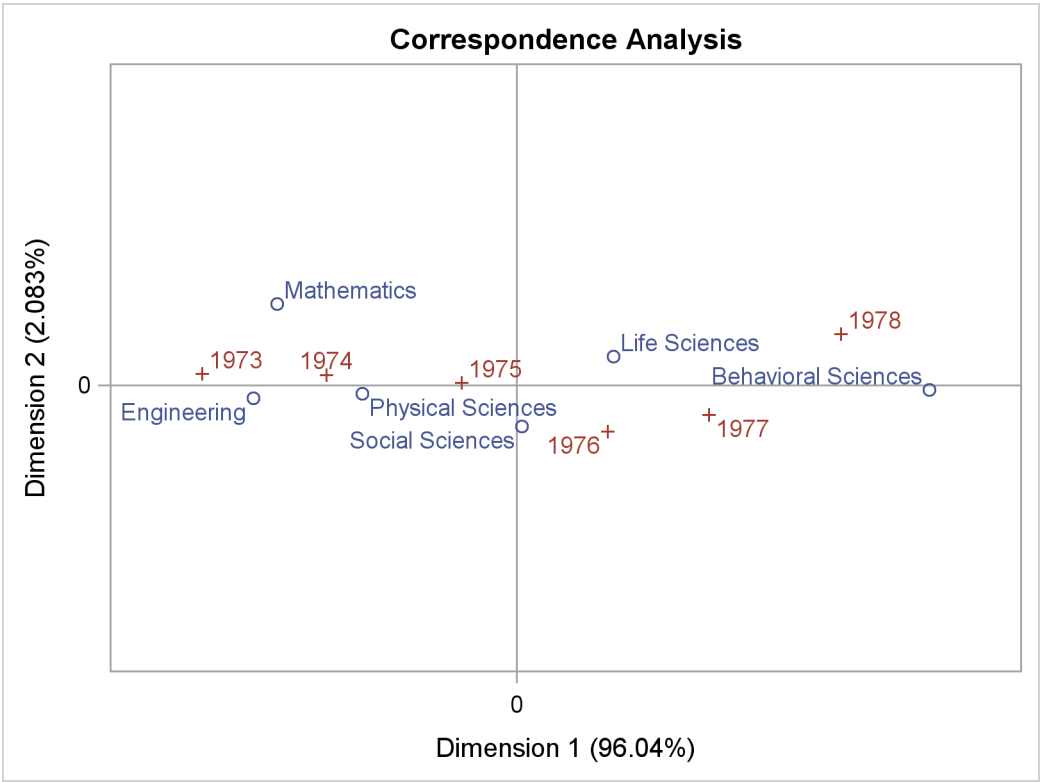
Instead of DISPLAY=(TICKVALUES), you can use DISPLAY=NONE (not shown) to remove the tick values from the display as well. You can change the tick values, as in the following example:

```
proc template;
   define statgraph Stat.Corresp.Graphics.Configuration;
      dynamic xVar yVar head legend;
      begingraph;
         entrytitle HEAD;

         layout overlayequated / equatetype=fit
            commonaxisopts=(tickvaluelist=(0))
            xaxisopts=(offsetmin=0.1 offsetmax=0.1)
            yaxisopts=(offsetmin=0.1 offsetmax=0.1);

            referenceline x=0;
            referenceline y=0;
```

```
            scatterplot y=YVAR x=XVAR / group=GROUP index=INDEX
               datalabel=LABEL datalabelattrs=GRAPHVALUETEXT
               name="Type" tip=(y x datalabel group)
               tiplabel=(group="Point");
            if (LEGEND)
               discretelegend "Type";
            endif;
         endlayout;
      endgraph;
   end;
run;

proc corresp data=PhD short;
   ods select configplot;
   var y1973-y1978;
   id Science;
run;
```

Since the axes in this plot are equated, the ticks are specified using the option **commonaxisopts=(tickvaluelist=(*tick-value-list*))**. This example only shows ticks at zero, but you can specify lists of values instead. The results are shown in Output 21.8.5.

Output 21.8.5 Scatter Plot with Tick Marks Specified



If the axes are not equated, then the tick value list is specified with the LINEAROPTS= option, as in the following statement:

```
layout overlay / xaxisopts=(linearopts=(viewmin=-0.1 viewmax=0.1
                                        tickvaluelist=(-0.1 0 0.1))
                            offsetmin=0.1 offsetmax=0.1)
                 yaxisopts=(linearopts=(viewmin=-0.1 viewmax=0.1
                                        tickvaluelist=(-0.1 0 0.1))
                            offsetmin=0.1 offsetmax=0.1);
```

The preceding statement uses the VIEWMIN= and VIEWMAX= options to specify the beginning and end of the data range that is shown. Specifying a tick value list does not extend or restrict the range of data shown in the plot. When axes share common options, it might be more convenient to use a macro to specify the options. The following two statements are equivalent to the preceding statement:

```
%let opts = linearopts=(viewmin=-0.1 viewmax=0.1
            tickvaluelist=(-0.1 0 0.1)) offsetmin=0.1 offsetmax=0.1;
layout overlay / xaxisopts=(&opts) yaxisopts=(&opts);
```

You can restore the default graph template as follows:

```
proc template;
   delete Stat.Corresp.Graphics.Configuration;
run;
```

Example 21.9: Customizing the Style for Box Plots

This example demonstrates how to modify the style for box plots. This example is taken from Example 21.1. The following step creates the data set:

```
data pr;
   input Person Gender $ y1 y2 y3 y4 @@;
   y=y1; Age=8;  output;
   y=y2; Age=10; output;
   y=y3; Age=12; output;
   y=y4; Age=14; output;
   drop y1-y4;
   datalines;
 1  F  21.0  20.0  21.5  23.0      2  F  21.0  21.5  24.0  25.5

... more lines ...

;
```

The next step displays the STATISTICAL and DEFAULT styles:

```
proc template;
   source Styles.Statistical;
   source Styles.Default;
run;
```

If you search for 'box', you will find the style element that controls some aspects of the box plot:

```
class GraphBox /
   capstyle = "serif"
   connect = "mean"
   displayopts = "fill caps median mean outliers";
```

You can learn more about the **GraphBox** style element and its attributes in the section on the BOXPLOT statement in the *SAS/GRAPH: Graph Template Language Reference* and in the section on "ODS Style Elements" in the *SAS Output Delivery System: User's Guide*.

The following statements create two new styles by modifying attributes of the **GraphBox** style element. The first style is a sparse style; the box is outlined (not filled) and the median is shown but not the mean. In contrast, the second style produces a filled box, with caps on the whiskers that shows the mean, median, and outliers. In addition, the box is notched. The following statements create the two styles:

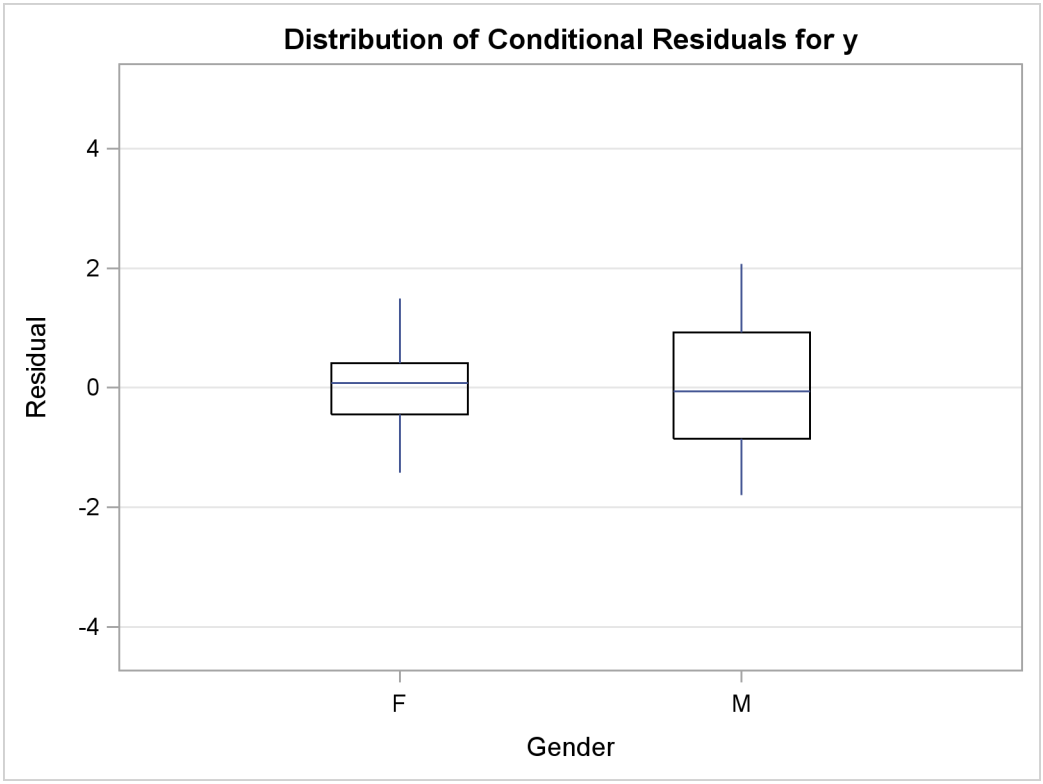
```
proc template;
   define style boxstylesparse;
      parent=styles.statistical;
      style GraphBox / capstyle = "line" displayopts = "median";
   end;
   define style boxstylerich;
      parent=styles.statistical;
      style GraphBox / capstyle = "bracket"
            displayopts = "fill caps median mean outliers notches";
   end;
run;
```

The following steps run PROC MIXED and create box plots that use the two styles:

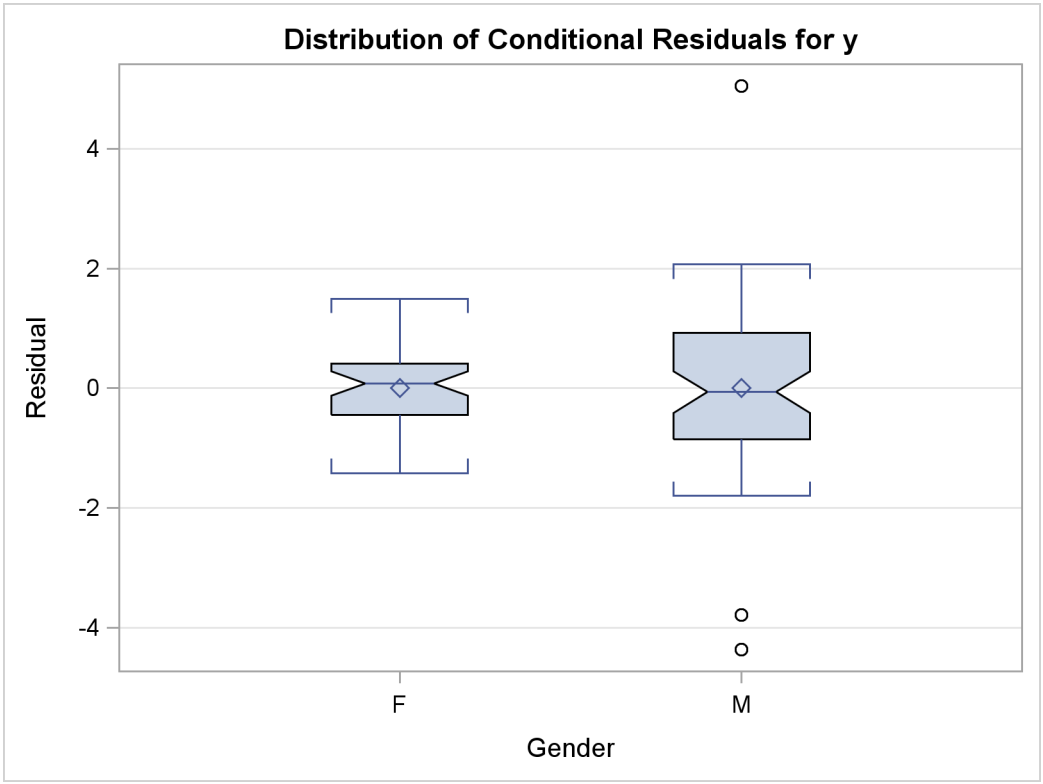
```
ods graphics on;
ods listing style=boxstylesparse;

proc mixed data=pr method=ml plots=boxplot;
   ods select 'Conditional Residuals by Gender';
   class Person Gender;
   model y = Gender Age Gender*Age;
   random intercept Age / type=un subject=Person;
run;

ods listing style=boxstylerich;

proc mixed data=pr method=ml plots=boxplot;
   ods select 'Conditional Residuals by Gender';
   class Person Gender;
   model y = Gender Age Gender*Age;
   random intercept Age / type=un subject=Person;
run;
```

The results with the sparse style are displayed in Output 21.9.1, and the results with the richer style are displayed in Output 21.9.2. See Output 21.1.1 in Example 21.1 to see the results using the STATISTICAL style.

Output 21.9.1 Box Plot with the Sparse Style



Output 21.9.2 Box Plot with the Richer Style



Example 21.10: Adding Text to Every Graph

This example shows how to add text to one or more graphs. For example, you can create a macro variable, with project and date information, as follows:

```
%let date = Project 17.104, &sysdate;
```

In order to add this information to a set of graphs, you need to first know the names of their templates. You can list the names of every graph template for SAS/STAT procedures or for a particular procedure as follows:

```
proc template;
   list stat     / where=(type='Statgraph');
   list stat.reg / where=(type='Statgraph');
run;
```

The results for PROC REG are shown in Output 21.10.1.

Output 21.10.1 PROC REG Templates

```
Listing of: SASHELP.TMPLMST
Path Filter is: Stat.Reg
Sort by: PATH/ASCENDING

Obs    Path                                                    Type
-------------------------------------------------------------------
 1     Stat.Reg.Graphics.CooksD                                Statgraph
 2     Stat.Reg.Graphics.DFBETASPanel                          Statgraph
 3     Stat.Reg.Graphics.DFBETASPlot                           Statgraph
 4     Stat.Reg.Graphics.DFFITSPlot                            Statgraph
 5     Stat.Reg.Graphics.DiagnosticsPanel                      Statgraph
 6     Stat.Reg.Graphics.Fit                                   Statgraph
 7     Stat.Reg.Graphics.ObservedByPredicted                   Statgraph
 8     Stat.Reg.Graphics.PartialPanel                          Statgraph
 9     Stat.Reg.Graphics.PartialPlot                           Statgraph
10     Stat.Reg.Graphics.PredictionPanel                       Statgraph
11     Stat.Reg.Graphics.QQPlot                                Statgraph
12     Stat.Reg.Graphics.RFPlot                                Statgraph
13     Stat.Reg.Graphics.RStudentByPredicted                   Statgraph
14     Stat.Reg.Graphics.ResidualBoxPlot                       Statgraph
15     Stat.Reg.Graphics.ResidualByPredicted                   Statgraph
16     Stat.Reg.Graphics.ResidualHistogram                     Statgraph
17     Stat.Reg.Graphics.ResidualPanel                         Statgraph
18     Stat.Reg.Graphics.ResidualPlot                          Statgraph
19     Stat.Reg.Graphics.RidgePanel                            Statgraph
20     Stat.Reg.Graphics.RidgePlot                             Statgraph
21     Stat.Reg.Graphics.SelectionCriterionPanel               Statgraph
22     Stat.Reg.Graphics.SelectionCriterionPlot                Statgraph
23     Stat.Reg.Graphics.StepSelectionCriterionPanel           Statgraph
24     Stat.Reg.Graphics.StepSelectionCriterionPlot            Statgraph
25     Stat.Reg.Graphics.VIFPlot                               Statgraph
26     Stat.Reg.Graphics.rstudentByLeverage                    Statgraph
```

You can show the source for the graph templates for SAS/STAT procedures or for a particular procedure as follows:

```
options ls=96;
proc template;
   source stat     / where=(type='Statgraph');
   source stat.reg / where=(type='Statgraph');
options ls=80;
```

The results of this step are not shown. However, Example 21.7 shows a portion of the template for the PROC REG diagnostics panel. Here, the OPTIONS statement is used to set a line size of 96, which sometimes works better than the smaller default line sizes when showing the source for large and complicated templates.

An abridged version of the first few lines of the diagnostics panel template is displayed next:

```
define statgraph Stat.Reg.Graphics.DiagnosticsPanel;
   notes "Diagnostics Panel";
   dynamic . . .;
   BeginGraph / designheight=defaultDesignWidth;
      entrytitle halign=left textattrs=GRAPHVALUETEXT _MODELLABEL
         halign=center textattrs=GRAPHTITLETEXT "Fit Diagnostics"
         " for " _DEPNAME;
      . . .
```

Adding a Date and Project Stamp to a Few Graphs

You can add the project and date to the bottom of all graphs produced with PROC REG by putting a PROC TEMPLATE statement in front of the template source code, and adding an MVAR and ENTRYFOOTNOTE statement after every BEGINGRAPH statement, as in the following example:

```
proc template;
   define statgraph Stat.Reg.Graphics.DiagnosticsPanel;
      notes "Diagnostics Panel";
      dynamic . . .;
      BeginGraph / designheight=defaultDesignWidth;

         mvar date;
         entryfootnote halign=left textattrs=GraphValueText date;

         entrytitle halign=left textattrs=GRAPHVALUETEXT _MODELLABEL
            halign=center textattrs=GRAPHTITLETEXT "Fit Diagnostics"
            " for " _DEPNAME;
         . . .
```

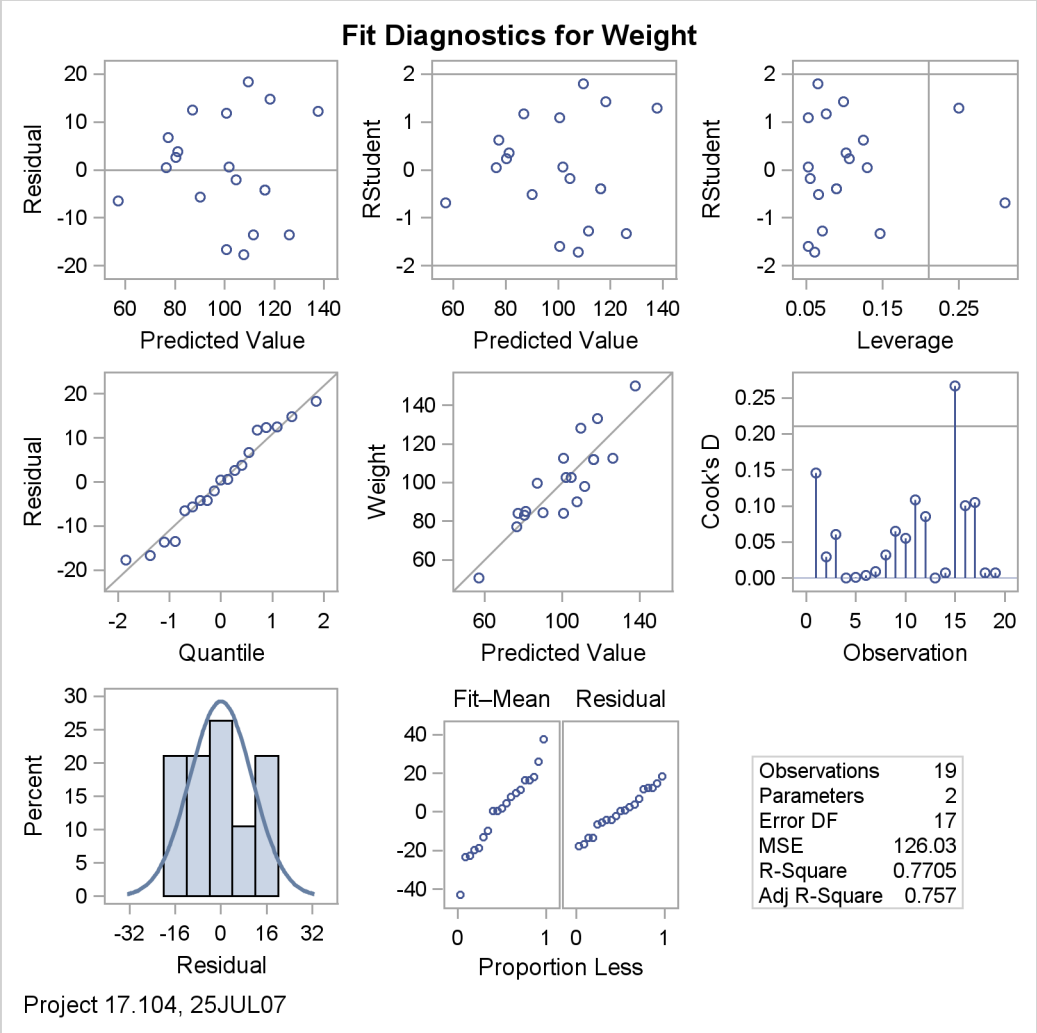
The MVAR statement enables you to dynamically customize the template and graph at procedure run time, just as the DYNAMIC statement enables the procedure to dynamically customize the template and graph. With the MVAR statement, you can modify the template once and reuse that modification as the macro changes over time: Alternatively, you can modify the templates as follows:

```
entryfootnote halign=left textattrs=GRAPHVALUETEXT "&date";
```

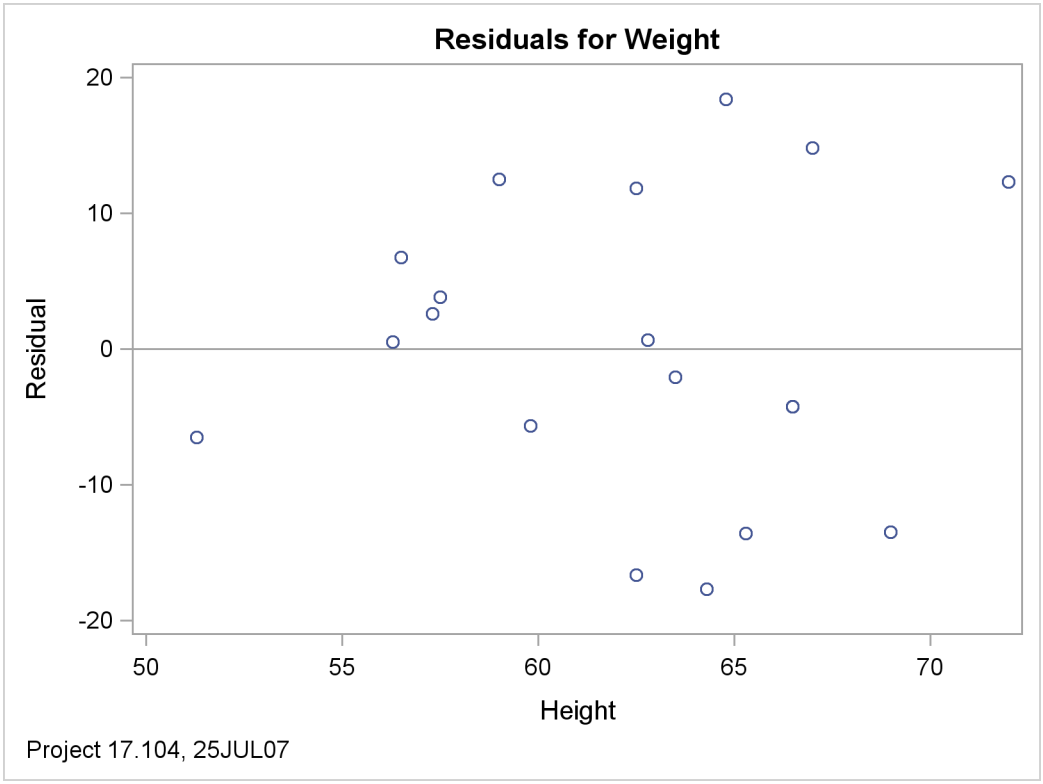
However, you would then have to resubmit your templates every time the macro variable changed. The substitution for the macro variable date occurs at different times in the two preceding cases. In the former case, ODS looks for the value of the macro variable date at the time the template is used, and then the current date variable is used to set the text in the ENTRYFOOTNOTE statement, every time the template is used. In the latter case, SAS substitutes the value of the macro variable once, at the time that the PROC TEMPLATE step is executed.

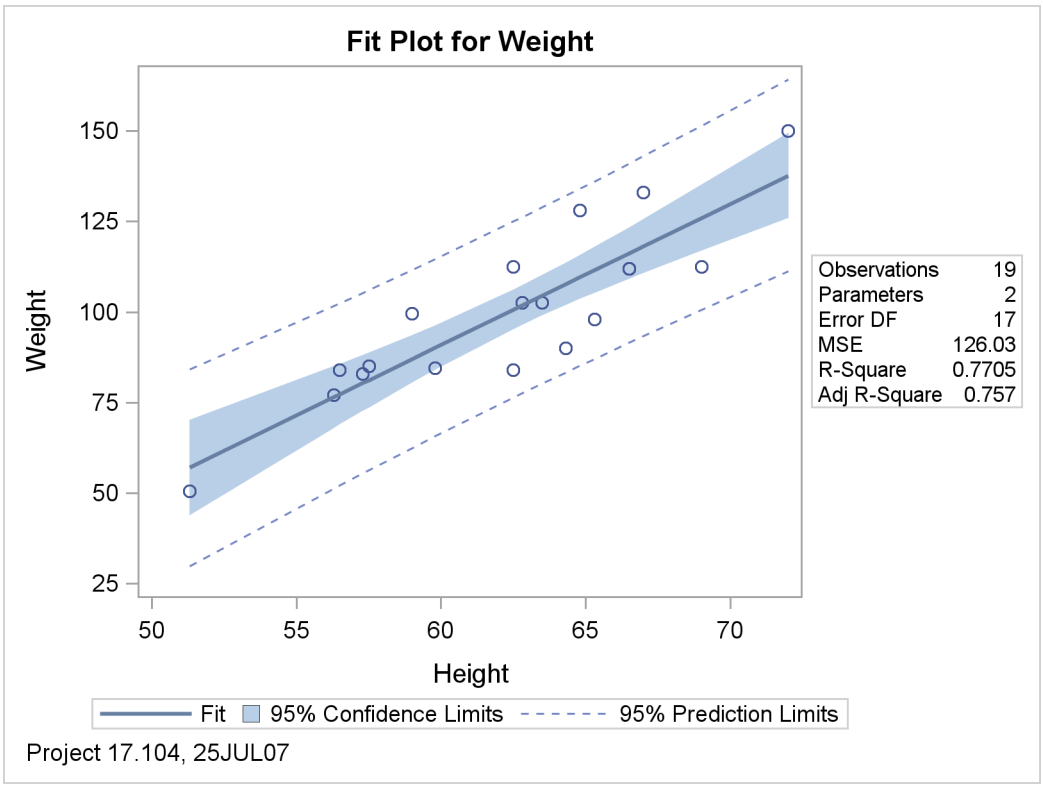
The following steps use the Class data set and produce Output 21.10.2:

Output 21.10.2 PROC REG Plots with Project and Date Stamp



Output 21.10.2 *continued*





You can restore all of the default templates for PROC REG by running the following step:

```
proc template;
   delete stat.reg;
run;
```

Alternatively, you can specify **delete stat** to restore all SAS/STAT templates to their default definitions.

You can add text to the top or the bottom of a graph by using the ENTRYTITLE or the ENTRYFOOTNOTE statement, respectively. With both statements, you can put the text in the HALIGN=RIGHT, HALIGN=LEFT, or HALIGN=CENTER positions. You can add text to titles even if they already have a centered title. For example, the ENTRYTITLE statement in the diagnostic panel has text on the left (which is conditionally displayed) and a centered title:

```
entrytitle halign=left textattrs=GraphValueText _MODELLABEL
   halign=center textattrs=GraphTitleText "Fit Diagnostics"
   " for " _DEPNAME;
```

The current title can be followed by HALIGN=RIGHT and more text.

Adding Data Set Information to a Graph

You might, for example, want to add text to a set of graphs that indicates the most recently created data set. The following example shows you how you can do this with the syslast macro variable:

```
%let data = &syslast;

. . .

mvar data;
entrytitle halign=left textattrs=GraphValueText "Data: " data
   halign=center textattrs=GraphTitleText "Fit Diagnostics"
   " for " _DEPNAME;

. . .
```

Of course, this only makes sense when you are analyzing the last data set created. Alternatively, you can incorporate the name of the data set in the title, as in the following example:

```
%let data = &syslast;

. . .

mvar data;
entrytitle halign=center textattrs=GraphTitleText
           "Fit Diagnostics for Data Set " data;

. . .
```

Adding a Date and Project Stamp to All Graphs

Sometimes, you can automate the process of template modification. For example, you can automatically add an MVAR and ENTRYFOOTNOTE statement to every graph template, as in the following example:

```
options ls=256;

proc template;
   source / where=(type='Statgraph') file="tpls.sas";
run;

options ls=80;

data _null_;
   infile 'tpls.sas' lrecl=256 pad;
   input line $ 1-256;
   file 'newtpls.sas';
   put line;
   line = left(lowcase(line));
   if line =: 'begingraph' then
      put 'mvar __date;' /
          'entryfootnote halign=left textattrs=GraphValueText __date;';

   file log;
   if index(line, '__date') then
      put 'ERROR: Name __date already used.' / line;
   if index(line, 'entryfootnote') then put line;
run;

proc template;
   %include 'newtpls.sas' / nosource;
run;
```

These statements write all ODS graph templates to a file, read that file, and write out a new file with an MVAR and ENTRYFOOTNOTE statement added after every BEGINGRAPH statement. Then these new templates are compiled with PROC TEMPLATE. These steps assume that no BEGINGRAPH statement is longer than 256 characters. Most graphs do not have footnotes. Those that do will now have multiple footnotes. You might want to manually combine them or write a more complicated program to handle them. These steps also assume that the name __date is not used anywhere. However, the program does check this and also lists all ENTRYFOOTNOTE statements. Be careful to check the SAS log to ensure that all templates compile without error. Also, before using templates that are automatically modified, make sure your modifications are reasonable.

You can delete SASUSER.Templat and hence all modified templates (assuming the default ODS PATH) as follows:

```
ods path sashelp.tmplmst(read);
proc datasets library=sasuser;
   delete templat(memtype=itemstor);
run;
ods path sasuser.templat(update) sashelp.tmplmst(read);
```

Example 21.11: PROC TEMPLATE Statement Order and Primary Plots

This example uses artificial data to illustrate two basic principles of template writing: that statement order matters and that one of the plotting statements is the primary statement. The data are a sample from a bivariate normal distribution. A custom graph template and PROC SGRENDER are used to plot the data along with vectors and ellipses. The plot consists of four components: a scatterplot of the data; vectors whose end points come from other variables in the data set; ellipses whose parameters are specified in the template; and reference lines whose locations are specified in the template. Initially, thick lines are used to show what happens at the places where the lines and points intersect.

The following steps create the input SAS data set:

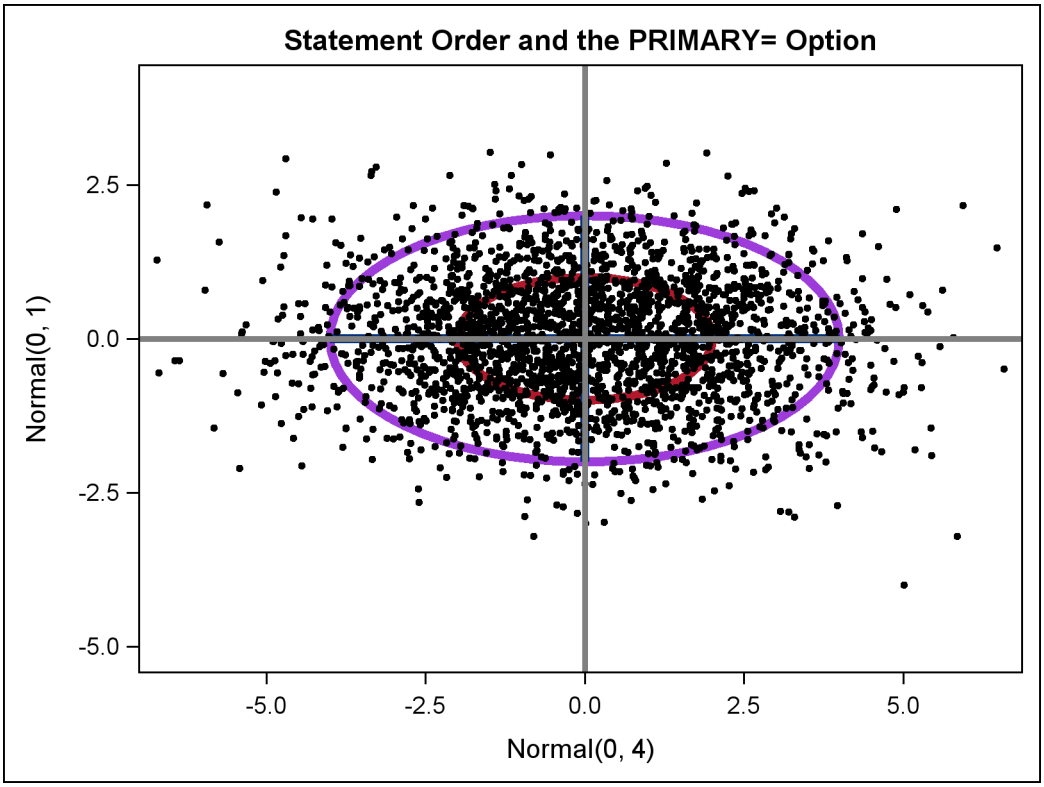
```
data x;
   input x y;
   label x = 'Normal(0, 4)' y = 'Normal(0, 1)';
   datalines;
-4  0
 4  0
 0 -2
 0  2
;

data y(drop=i);
   do i = 1 to 2500;
      r1 = normal( 104 );
      r2 = normal( 104 ) * 2;
      output;
   end;
run;

data all;
   merge x y;
run;
```

The data set All contains four variables. The variables r1 and r2 contain the random data. These variables contain 2500 nonmissing observations. The data set also contains the variables x and y, which contain the end points for the vectors. These variables contain four nonmissing observations and 2496 observations that are all missing. A data set like this is not unusual when creating overlaid plots. Different overlays often require input data with very different sizes.

First, the data are plotted by using a template that is deliberately constructed to demonstrate a number of problems that can occur with statement order. The following steps create Output 21.11.1:

```
proc template;
   define statgraph Plot;
      begingraph;
         entrytitle 'Statement Order and the PRIMARY= Option';
         layout overlayequated / equatetype=fit;
            ellipseparm semimajor=eval(sqrt(4)) semiminor=1
                        slope=0 xorigin=0 yorigin=0 /
                        outlineattrs=GraphData2(pattern=solid thickness=5);
            ellipseparm semimajor=eval(2 * sqrt(4)) semiminor=2
                        slope=0 xorigin=0 yorigin=0 /
                        outlineattrs=GraphData5(pattern=solid thickness=5);
            vectorplot  y=y x=x xorigin=0 yorigin=0 /
                        arrowheads=false lineattrs=GraphFit(thickness=5);
            scatterplot y=r1 x=r2 /
                        markerattrs=(symbol=circlefilled size=3);
            referenceline x=0 / lineattrs=(thickness=3);
            referenceline y=0 / lineattrs=(thickness=3);
         endlayout;
      endgraph;
   end;
run;

ods listing style=listing;

proc sgrender data=all template=plot;
run;
```

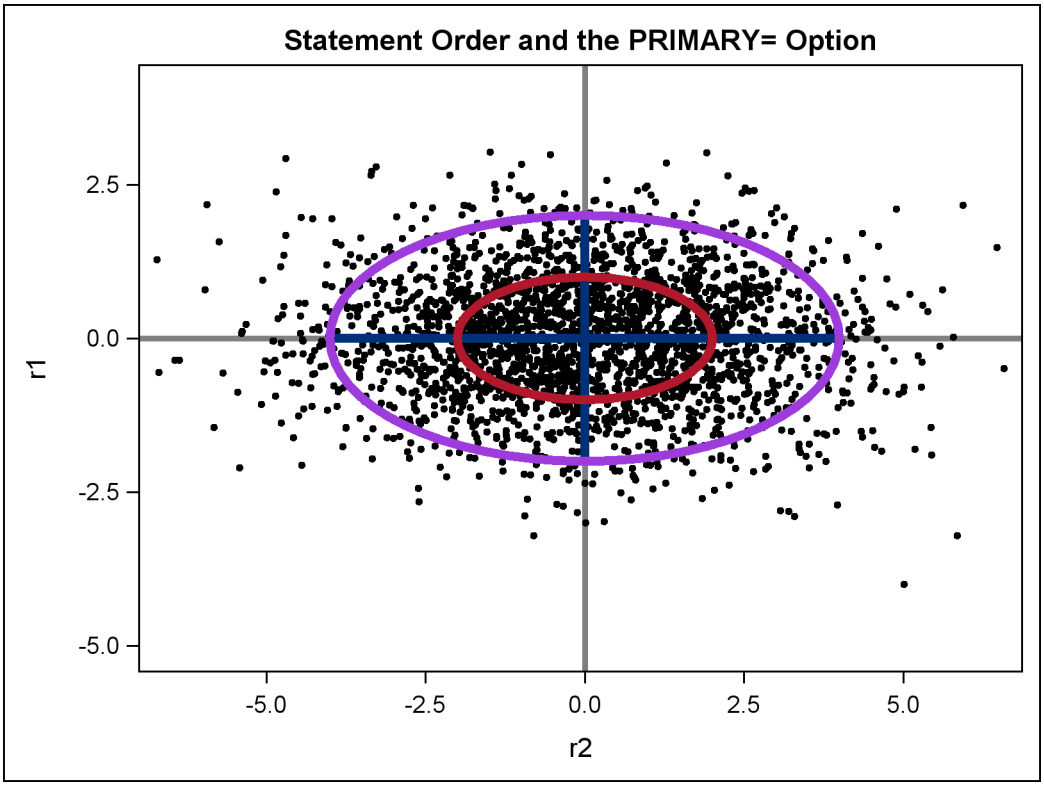
Output 21.11.1 Statements Specified in a Nonoptimal Order



There are a number of problems with the plot in Output 21.11.1. The reference lines obliterate the vectors, and the data are on top of everything but the reference lines. It might be more reasonable to plot the reference lines first, the data next, the vectors next, and the ellipses last. The following steps do this and produce Output 21.11.2:

```
proc template;
   define statgraph Plot;
      begingraph;
         entrytitle 'Statement Order and the PRIMARY= Option';
         layout overlayequated / equatetype=fit;
            referenceline x=0 / lineattrs=(thickness=3);
            referenceline y=0 / lineattrs=(thickness=3);
            scatterplot y=r1 x=r2 /
                        markerattrs=(symbol=circlefilled size=3);
            vectorplot  y=y x=x xorigin=0 yorigin=0 /
                        arrowheads=false lineattrs=GraphFit(thickness=5);
            ellipseparm semimajor=eval(sqrt(4)) semiminor=1
                        slope=0 xorigin=0 yorigin=0 /
                        outlineattrs=GraphData2(pattern=solid thickness=5);
            ellipseparm semimajor=eval(2 * sqrt(4)) semiminor=2
                        slope=0 xorigin=0 yorigin=0 /
                        outlineattrs=GraphData5(pattern=solid thickness=5);
         endlayout;
      endgraph;
   end;
run;

ods listing style=listing;

proc sgrender data=all template=plot;
run;
```

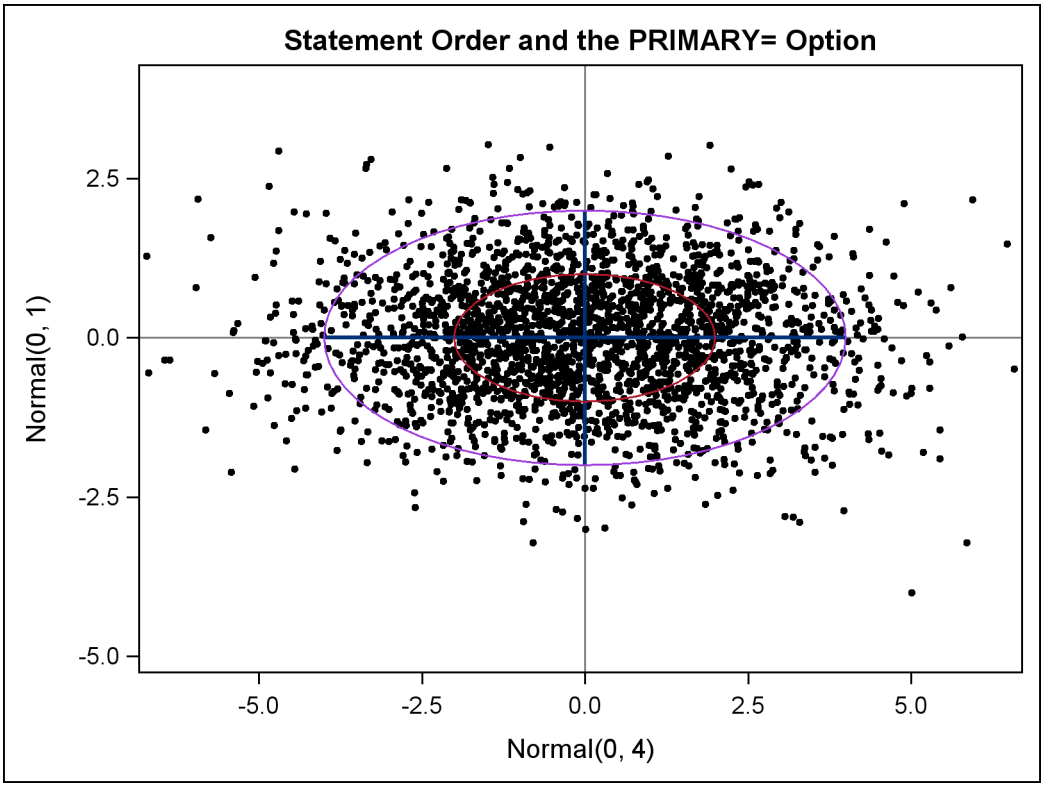
Output 21.11.2 Statement Order Fixed



Output 21.11.2 looks better than Output 21.11.1, but the labels for the axes have changed. Output 21.11.1 has the labels of the variables x and y as axis labels, whereas Output 21.11.2 uses the names of the variables r1 and r2. This is because in the Output 21.11.1, the first plot is the vector plot of x and y (which have labels), and in Output 21.11.2, the first plot is the scatter plot of r1 and r2 (which do not have labels). By default, the first plot is the *primary plot*, and the primary plot is used to determine the axis type and labels. You can designate the vector plot as the primary plot with the PRIMARY=TRUE option. The following statements make the final plot, this time with default line thicknesses, and produce Output 21.11.3:

```
proc template;
   define statgraph Plot;
      begingraph;
         entrytitle 'Statement Order and the PRIMARY= Option';
         layout overlayequated / equatetype=fit;
            referenceline x=0;
            referenceline y=0;
            scatterplot y=r1 x=r2 / markerattrs=(symbol=circlefilled size=3);

            vectorplot  y=y x=x xorigin=0 yorigin=0 / primary=true
                        arrowheads=false lineattrs=GraphFit;

            ellipseparm semimajor=eval(sqrt(4)) semiminor=1
                        slope=0 xorigin=0 yorigin=0 /
                        outlineattrs=GraphData2(pattern=solid);
            ellipseparm semimajor=eval(2 * sqrt(4)) semiminor=2
                        slope=0 xorigin=0 yorigin=0 /
                        outlineattrs=GraphData5(pattern=solid);
         endlayout;
      endgraph;
   end;
run;

ods listing style=listing;

proc sgrender data=all template=plot;
run;
```

Output 21.11.3 Statement Order Fixed and Primary Plot Specified



The axis labels in Output 21.11.3 and the overprinting of plot elements look better than in the previous plots. You can further adjust the line thicknesses if you want to emphasize or deemphasize components of this plot. The following list discusses the syntax of the GTL statements used in this example.

- The template has an ENTRYTITLE statement that specifies the title.
- The template has an equated overlay. This means that a centimeter on one axis represents the same data range as a centimeter on the other axis. This is done instead of the more common LAYOUT OVERLAY since with these data, the shape and geometry of the data have meaning even though the ranges of the two axis variables are different. The option EQUATETYPE=SQUARE is used to make a square plot, but since the X-axis variable has a larger range than the Y-axis variable, and since the default plot size is wider than high, EQUATETYPE=FIT is specified. The axes are equated but use the available space.
- A vertical reference line is drawn at X=0, and a horizontal reference line is drawn at Y=0.
- The scatter plot is based on the Y-axis variable `r2` and the X-axis variable `r1`. The markers are filled circles with a size of three pixels. This is smaller than the default size and works well with a plot that displays many points.
- The vector plot is based on the Y-axis variable `y` and the X-axis variable `x`. The vectors are solid lines with no heads emanating from the origin (X=0 and Y=0). The color and other line attributes such as thickness come from the attributes of the **`GraphFit`** style element. This is the primary plot, so the default axis labels are the variable labels for the X= and Y= variables if they exist or the variable names if the variables do not have labels.
- The plot also displays two ellipses with X=0 and Y=0 at their center. Their widths are expressions, and their heights are constant. The expressions are not needed in this example; they are used to illustrate the syntax. The SEMIMAJOR= option specifies half the length of the major axis for the ellipse, and the SEMIMINOR= option specifies half the length of the minor axis for the ellipse. The SLOPE= option specifies the slope of the major axis for the ellipse. The colors of the ellipses and other line properties are based on the **`GraphData2`** and **`GraphData5`** style elements, but the line pattern attribute from the style is overridden.

Chapter 22

The ACECLUS Procedure

Contents

Overview: ACECLUS Procedure **659**
Background 660
Getting Started: ACECLUS Procedure **665**
Syntax: ACECLUS Procedure **673**
PROC ACECLUS Statement 674
BY Statement 678
FREQ Statement 679
VAR Statement 679
WEIGHT Statement 679
Details: ACECLUS Procedure **679**
Missing Values 679
Output Data Sets 680
Computational Resources 681
Displayed Output 682
ODS Table Names 683
Example: ACECLUS Procedure **683**
Example 22.1: Transformation and Cluster Analysis of Fisher Iris Data . . . 683
References **689**

Overview: ACECLUS Procedure

The ACECLUS (approximate covariance estimation for clustering) procedure obtains approximate estimates of the pooled within-cluster covariance matrix when the clusters are assumed to be multivariate normal with equal covariance matrices. Neither cluster membership nor the number of clusters needs to be known. PROC ACECLUS is useful for preprocessing data to be subsequently clustered by the CLUSTER or FASTCLUS procedure.

Many clustering methods perform well with spherical clusters but poorly with elongated elliptical clusters (Everitt 1980, 77–97). If the elliptical clusters have roughly the same orientation and eccentricity, you can apply a linear transformation to the data to yield a spherical within-cluster covariance matrix—that is, a covariance matrix proportional to the identity. Equivalently, the distance between observations can be measured in the metric of the inverse of the pooled within-cluster

covariance matrix. The remedy is difficult to apply, however, because you need to know what the clusters are in order to compute the sample within-cluster covariance matrix. One approach is to estimate iteratively both cluster membership and within-cluster covariance (Wolfe 1970; Hartigan 1975). Another approach is provided by Art, Gnanadesikan, and Kettenring (1982). They have devised an ingenious method for estimating the within-cluster covariance matrix without knowledge of the clusters. The method can be applied before any of the usual clustering techniques, including hierarchical clustering methods.

First, Art, Gnanadesikan, and Kettenring (1982) obtain a decomposition of the total-sample sum-of-squares-and-crossproducts (SSCP) matrix into within-cluster and between-cluster SSCP matrices computed from pairwise differences between observations, rather than differences between observations and means. Then, they show how the within-cluster SSCP matrix based on pairwise differences can be approximated without knowing the number or the membership of the clusters. The approximate within-cluster SSCP matrix can be used to compute distances for cluster analysis, or it can be used in a canonical analysis similar to canonical discriminant analysis. For more information, see Chapter 27, "The CANDISC Procedure."

Art, Gnanadesikan, and Kettenring demonstrate by Monte Carlo calculations that their method can produce better clusters than the Euclidean metric even when the approximation to the within-cluster SSCP matrix is poor or the within-cluster covariances are moderately heterogeneous. The algorithm used by the ACECLUS procedure differs slightly from the algorithm used by Art, Gnanadesikan, and Kettenring. In the following sections, the PROC ACECLUS algorithm is described first; then, differences between PROC ACECLUS and the method used by Art, Gnanadesikan, and Kettenring are summarized.

Background

It is well known from the literature on nonparametric statistics that variances and, hence, covariances can be computed from pairwise differences instead of deviations from means. (For example, Puri and Sen (1971, pp. 51–52) show that the variance is a U statistic of degree 2.) Let $\mathbf{X} = (x_{ij})$ be the data matrix with n observations (rows) and v variables (columns), and let $\bar{x}_j$ be the mean of the jth variable. The sample covariance matrix $\mathbf{S} = (s_{jk})$ is usually defined as

$$s_{jk} = \frac{1}{n-1} \sum_{i=1}^{n} (x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)$$

The matrix $\mathbf{S}$ can also be computed as

$$s_{jk} = \frac{1}{n(n-1)} \sum_{i=2}^{n} \sum_{h=1}^{i-1} (x_{ij} - x_{hj})(x_{ik} - x_{hk})$$

Let $\mathbf{W} = (w_{jk})$ be the pooled within-cluster covariance matrix, q be the number of clusters, n_c be the number of observations in the cth cluster, and

$$d''_{ic} = \begin{cases} 1 & \text{if observation } i \text{ is in cluster } c \\ 0 & \text{otherwise} \end{cases}$$

The matrix $\mathbf{W}$ is normally defined as

$$w_{jk} = \frac{1}{n-q} \sum_{c=1}^{q} \sum_{i=1}^{n} d''_{ic}(x_{ij} - \bar{x}_{cj})(x_{ik} - \bar{x}_{ck})$$

where $\bar{x}_{cj}$ is the mean of the jth variable in cluster c. Let

$$d'_{ih} = \begin{cases} \frac{1}{n_c} & \text{if observations } i \text{ and } h \text{ are in cluster } c \\ 0 & \text{otherwise} \end{cases}$$

The matrix $\mathbf{W}$ can also be computed as

$$w_{jk} = \frac{1}{n-q} \sum_{i=2}^{n} \sum_{h=1}^{i-1} d'_{ih}(x_{ij} - x_{hj})(x_{ik} - x_{hk})$$

If the clusters are not known, d'_{ih} cannot be determined. However, an approximation to $\mathbf{W}$ can be obtained by using instead

$$d_{ih} = \begin{cases} 1 & \text{if } \sum_{j=1}^{v} \sum_{k=1}^{v} m_{jk}(x_{ij} - x_{hj})(x_{ik} - x_{hk}) \le u^2 \\ 0 & \text{otherwise} \end{cases}$$

where u is an appropriately chosen value and $\mathbf{M} = (m_{jk})$ is an appropriate metric. Let $\mathbf{A} = (a_{jk})$ be defined as

$$a_{jk} = \frac{\sum_{i=2}^{n} \sum_{h=1}^{i-1} d_{ih}(x_{ij} - x_{hj})(x_{ik} - x_{hk})}{2 \sum_{i=2}^{n} \sum_{h=1}^{i-1} d_{ih}}$$

If all of the following conditions hold, $\mathbf{A}$ equals $\mathbf{W}$:

- All within-cluster distances in the metric $\mathbf{M}$ are less than or equal to u.
- All between-cluster distances in the metric $\mathbf{M}$ are greater than u.
- All clusters have the same number of members n_c.

If the clusters are of unequal size, $\mathbf{A}$ gives more weight to large clusters than $\mathbf{W}$ does, but this discrepancy should be of little importance if the population within-cluster covariance matrices are equal. There might be large differences between $\mathbf{A}$ and $\mathbf{W}$ if the cutoff u does not discriminate between pairs in the same cluster and pairs in different clusters. Lack of discrimination might occur for one of the following reasons:

- The clusters are not well separated.
- The metric $\mathbf{M}$ or the cutoff u is not chosen appropriately.

In the former case, little can be done to remedy the problem. The remaining question concerns how to choose $\mathbf{M}$ and u. Consider $\mathbf{M}$ first. The best choice for $\mathbf{M}$ is $\mathbf{W}^{-1}$, but $\mathbf{W}$ is not known. The solution is to use an iterative algorithm:

1. Obtain an initial estimate of **A**, such as the identity or the total-sample covariance matrix. See the INITIAL= option in the PROC ACECLUS statement for more information.
2. Let **M** equal $\mathbf{A}^{-1}$.
3. Recompute **A** by using the preceding formula.
4. Repeat steps 2 and 3 until the estimate stabilizes.

Convergence is assessed by comparing values of **A** on successive iterations. Let $\mathbf{A}_i$ be the value of **A** on the ith iteration and $\mathbf{A}_0$ be the initial estimate of **A**. Let **Z** be a user-specified $v \times v$ matrix. See the METRIC= option in the PROC ACECLUS statement for more information. The convergence measure is

$$e_i = \frac{1}{v} \parallel \mathbf{Z}'(\mathbf{A}_i - \mathbf{A}_{i-1})\mathbf{Z} \parallel$$

where $\parallel \cdots \parallel$ indicates the Euclidean norm—that is, the square root of the sum of the squares of the elements of the matrix. In PROC ACECLUS, **Z** can be the identity or an inverse factor of **S** or diag(**S**). Iteration stops when e_i falls below a user-specified value. See the CONVERGE= option or the MAXITER= option in the PROC ACECLUS statement for more information.

The remaining question of how to choose u has no simple answer. In practice, you must try several different values. PROC ACECLUS provides four different ways of specifying u:

- You can specify a constant value for u. This method is useful if the initial estimate of **A** is quite good. See the ABSOLUTE option and the THRESHOLD= option in the PROC ACECLUS statement for more information.
- You can specify a threshold value $t > 0$ that is multiplied by the root mean square distance between observations in the current metric on each iteration to give u. Thus, the value of u changes from iteration to iteration. This method is appropriate if the initial estimate of **A** is poor. See the THRESHOLD= option in the PROC ACECLUS statement for more information.
- You can specify a value p, $0 < p < 1$, to be transformed into a distance u such that approximately a proportion p of the pairwise Mahalanobis distances between observations in a random sample from a multivariate normal distribution will be less than u in repeated sampling. The transformation can be computed only if the number of observations exceeds the number of variables, preferably by at least 10 percent. This method also requires a good initial estimate of **A**. See the PROPORTION= option and the ABSOLUTE option in the PROC ACECLUS statement for more information.
- You can specify a value p, $0 < p < 1$, to be transformed into a value t that is then multiplied by $1/\sqrt{2v}$ times the root mean square distance between observations in the current metric on each iteration to yield u. The value of u changes from iteration to iteration. This method can be used with a poor initial estimate of **A**. See the PROPORTION= option in the PROC ACECLUS statement for more information.

In most cases, the analysis should begin with the last method, using values of p between 0.5 and 0.01 and using the full covariance matrix as the initial estimate of **A**.

Proportions p are transformed to distances t by using the formula

$$t^2 = 2v \left\{ \left[F^{-1}_{v,n-v}(p) \right]^{\frac{n-v}{n-1}} \right\}$$

where $F^{-1}_{v,n-v}$ is the quantile (inverse cumulative distribution) function of an F random variable with v and $n - v$ degrees of freedom. The squared Mahalanobis distance between a single pair of observations sampled from a multivariate normal distribution is distributed as $2v$ times an F random variable with v and $n - v$ degrees of freedom. The distances between two pairs of observations are correlated if the pairs have an observation in common. The quantile function is raised to the power given in the preceding formula to compensate approximately for the correlations among distances between pairs of observations that share a member. Monte Carlo studies indicate that the approximation is acceptable if the number of observations exceeds the number of variables by at least 10 percent.

If $\mathbf{A}$ becomes singular, step 2 in the iterative algorithm cannot be performed because $\mathbf{A}$ cannot be inverted. In this case, let $\mathbf{Z}$ be the matrix as defined in discussing the convergence measure, and let $\mathbf{Z}'\mathbf{A}\mathbf{Z} = \mathbf{R}'\mathbf{\Lambda}\mathbf{R}$, where $\mathbf{R}'\mathbf{R} = \mathbf{R}\mathbf{R}' = \mathbf{I}$ and $\mathbf{\Lambda} = (\lambda_{jk})$ is diagonal. Let $\mathbf{\Lambda}^* = (\lambda^*_{jk})$ be a diagonal matrix, where $\lambda^*_{jj} = \max(\lambda_{jj}, g \text{ trace}(\mathbf{\Lambda}))$, and $0 < g < 1$ is a user-specified singularity criterion (see the SINGULAR= option in the PROC ACECLUS statement for more information). Then $\mathbf{M}$ is computed as $\mathbf{Z}\mathbf{R}'(\mathbf{\Lambda}^*)^{-1}\mathbf{R}\mathbf{Z}'$.

The ACECLUS procedure differs from the method used by Art, Gnanadesikan, and Kettenring (1982) in several respects:

- The Art, Gnanadesikan, and Kettenring method uses the identity matrix as the initial estimate, whereas the ACECLUS procedure enables you to specify any symmetric matrix as the initial estimate and defaults to the total-sample covariance matrix. The default initial estimate in PROC ACECLUS is chosen to yield invariance under nonsingular linear transformations of the data but might sometimes obscure clusters that become apparent if the identity matrix is used.
- The Art, Gnanadesikan, and Kettenring method carries out all computations with SSCP matrices, whereas the ACECLUS procedure uses estimated covariance matrices because covariances are easier to interpret than crossproducts.
- The Art, Gnanadesikan, and Kettenring method uses the m pairs with the smallest distances to form the new estimate at each iteration, where m is specified by the user, whereas the ACECLUS procedure uses all pairs closer than a given cutoff value. Kettenring (1984) says that the m-closest-pairs method seems to give the user more direct control. PROC ACECLUS uses a distance cutoff because it yields a slight decrease in computer time and because in some cases, such as widely separated spherical clusters, the results are less sensitive to the choice of distance cutoff than to the choice of m. Much research remains to be done on this issue.
- The Art, Gnanadesikan, and Kettenring method uses a different convergence measure. Let $\mathbf{A}_i$ be computed on each iteration by using the m-closest-pairs method, and let $\mathbf{B}_i = \mathbf{A}^{-1}_{i-1}\mathbf{A}_i - \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. The convergence measure is equivalent to trace$(\mathbf{B}^2_i)$.

Analyses of Fisher's (1936) iris data, consisting of measurements of petal and sepal length and width for 50 specimens from each of three iris species, are summarized in Table 22.1. The number of misclassified observations out of 150 is given for four clustering methods:

- k-means as implemented in PROC FASTCLUS with MAXC=3, MAXITER=99, and CONV=0
- Ward's minimum variance method as implemented in PROC CLUSTER
- average linkage on Euclidean distances as implemented in PROC CLUSTER
- the centroid method as implemented in PROC CLUSTER

Each hierarchical analysis is followed by the TREE procedure with NCL=3 to determine cluster assignments at the three-cluster level. Clusters with 20 or fewer observations are discarded by using the DOCK=20 option. The observations in a discarded cluster are considered unclassified.

Each method is applied to the following data:

- the raw data
- the data standardized to unit variance by the STANDARD procedure
- two standardized principal components accounting for 95 percent of the standardized variance and having an identity total-sample covariance matrix, computed by the PRINCOMP procedure with the STD option
- four standardized principal components having an identity total-sample covariance matrix, computed by PROC PRINCOMP with the STD option
- the data transformed by PROC ACECLUS, using seven different settings of the PROPORTION= (P=) option
- four canonical variables having an identity pooled within-species covariance matrix, computed using the CANDISC procedure

Theoretically, the best results should be obtained by using the canonical variables from PROC CANDISC. PROC ACECLUS yields results comparable to those from PROC CANDISC for values of the PROPORTION= option ranging from 0.005 to 0.02. At PROPORTION=0.04, average linkage and the centroid method show some deterioration, but k-means and Ward's method continue to produce excellent classifications. At larger values of the PROPORTION= option, all methods perform poorly, although no worse than with four standardized principal components.

Table 22.1 Number of Misclassified and Unclassified Observations Using Fisher's (1936) Iris Data

Data	Clustering Method			
	k-means	Ward's	Average Linkage	Centroid
raw data	16*	16*	25 + 12**	14*
standardized data	25	26	33+4	33+4
two standardized principal components	29	31	30+9	27+32
four standardized principal components	39	27	32+7	45+11
transformed by ACECLUS P=0.32	39	10+9	7+25	
transformed by ACECLUS P=0.16	39	18+9	7+19	7+26
transformed by ACECLUS P=0.08	19	9	3+13	5+16
transformed by ACECLUS P=0.04	4	5	1+19	3+12
transformed by ACECLUS P=0.02	4	3	3	3
transformed by ACECLUS P=0.01	4	4	3	4
transformed by ACECLUS P=0.005	4	4	4	4
canonical variables	3	5	4	4+1

* A single number represents misclassified observations with no unclassified observations.

** Where two numbers are separated by a plus sign, the first is the number of misclassified observations; the second is the number of unclassified observations.

This example demonstrates the following:

- PROC ACECLUS can produce results as good as those from the optimal transformation.
- PROC ACECLUS can be useful even when the within-cluster covariance matrices are moderately heterogeneous.
- The choice of the distance cutoff as specified by the PROPORTION= or the THRESHOLD= option is important, and several values should be tried.
- Commonly used transformations such as standardization and principal components can produce poor classifications.

Although experience with the Art, Gnanadesikan, and Kettenring and PROC ACECLUS methods is limited, the results so far suggest that these methods help considerably more often than they hinder the subsequent cluster analysis, especially with normal-mixture techniques such as k-means and Ward's minimum variance method.

Getting Started: ACECLUS Procedure

The following example demonstrates how you can use the ACECLUS procedure to obtain approximate estimates of the pooled within-cluster covariance matrix and to compute canonical variables

for subsequent analysis. You use PROC ACECLUS to preprocess data before you cluster it by using the FASTCLUS or CLUSTER procedure.

Suppose you want to determine whether national figures for birth rates, death rates, and infant death rates can be used to determine certain types or categories of countries. You want to perform a cluster analysis to determine whether the observations can be formed into groups suggested by the data. Previous studies indicate that the clusters computed from this type of data can be elongated and elliptical. Thus, you need to perform a linear transformation on the raw data before the cluster analysis.

The following data[1] from Rouncefield (1995) are the birth rates, death rates, and infant death rates for 97 countries. The following statements create the SAS data set Poverty:

```
data poverty;
   input Birth Death InfantDeath Country $15. @@;
   datalines;
24.7  5.7  30.8 Albania         12.5 11.9  14.4 Bulgaria
13.4 11.7  11.3 Czechoslovakia    12 12.4   7.6 Former_E._Germa
11.6 13.4  14.8 Hungary         14.3 10.2    16 Poland
13.6 10.7  26.9 Romania           14    9  20.2 Yugoslavia
17.7   10    23 USSR            15.2  9.5  13.1 Byelorussia
13.4 11.6    13 Ukrainian_SSR   20.7  8.4  25.7 Argentina
46.6   18   111 Bolivia         28.6  7.9    63 Brazil
23.4  5.8  17.1 Chile           27.4  6.1    40 Columbia
32.9  7.4    63 Ecuador         28.3  7.3    56 Guyana
34.8  6.6    42 Paraguay        32.9  8.3 109.9 Peru
18    9.6  21.9 Uruguay         27.5  4.4  23.3 Venezuela
29   23.2    43 Mexico            12 10.6   7.9 Belgium
13.2 10.1   5.8 Finland         12.4 11.9   7.5 Denmark
13.6  9.4   7.4 France          11.4 11.2   7.4 Germany
10.1  9.2    11 Greece          15.1  9.1   7.5 Ireland
9.7   9.1   8.8 Italy           13.2  8.6   7.1 Netherlands
14.3 10.7   7.8 Norway          11.9  9.5  13.1 Portugal
10.7  8.2   8.1 Spain           14.5 11.1   5.6 Sweden
12.5  9.5   7.1 Switzerland     13.6 11.5   8.4 U.K.
14.9  7.4     8 Austria          9.9  6.7   4.5 Japan
14.5  7.3   7.2 Canada          16.7  8.1   9.1 U.S.A.
40.4 18.7 181.6 Afghanistan     28.4  3.8    16 Bahrain
42.5 11.5 108.1 Iran            42.6  7.8    69 Iraq
22.3  6.3   9.7 Israel          38.9  6.4    44 Jordan
26.8  2.2  15.6 Kuwait          31.7  8.7    48 Lebanon
45.6  7.8    40 Oman            42.1  7.6    71 Saudi_Arabia
29.2  8.4    76 Turkey          22.8  3.8    26 United_Arab_Emr
42.2 15.5   119 Bangladesh      41.4 16.6   130 Cambodia
21.2  6.7    32 China           11.7  4.9   6.1 Hong_Kong
30.5 10.2    91 India           28.6  9.4    75 Indonesia
23.5 18.1    25 Korea           31.6  5.6    24 Malaysia
36.1  8.8    68 Mongolia        39.6 14.8   128 Nepal
30.3  8.1 107.7 Pakistan        33.2  7.7    45 Philippines
17.8  5.2   7.5 Singapore       21.3  6.2  19.4 Sri_Lanka
```

[1] These data have been compiled from the United Nations *Demographic Yearbook* 1990 (United Nations publications, Sales No. E/F.91.XII.1, copyright 1991, United Nations, New York) and are reproduced with the permission of the United Nations.

```
22.3  7.7    28 Thailand        31.8  9.5    64 Vietnam
35.5  8.3    74 Algeria         47.2 20.2   137 Angola
48.5 11.6    67 Botswana        46.1 14.6    73 Congo
38.8  9.5  49.4 Egypt           48.6 20.7   137 Ethiopia
39.4 16.8   103 Gabon           47.4 21.4   143 Gambia
44.4 13.1    90 Ghana             47 11.3    72 Kenya
44    9.4    82  Libya          48.3   25   130 Malawi
35.5  9.8    82 Morocco           45 18.5   141 Mozambique
44   12.1   135 Namibia         48.5 15.6   105 Nigeria
48.2 23.4   154 Sierra_Leone    50.1 20.2   132 Somalia
32.1  9.9    72 South_Africa    44.6 15.8   108 Sudan
46.8 12.5   118 Swaziland       31.1  7.3    52 Tunisia
52.2 15.6   103 Uganda          50.5   14   106 Tanzania
45.6 14.2    83 Zaire           51.1 13.7    80 Zambia
41.7 10.3    66 Zimbabwe
;
```

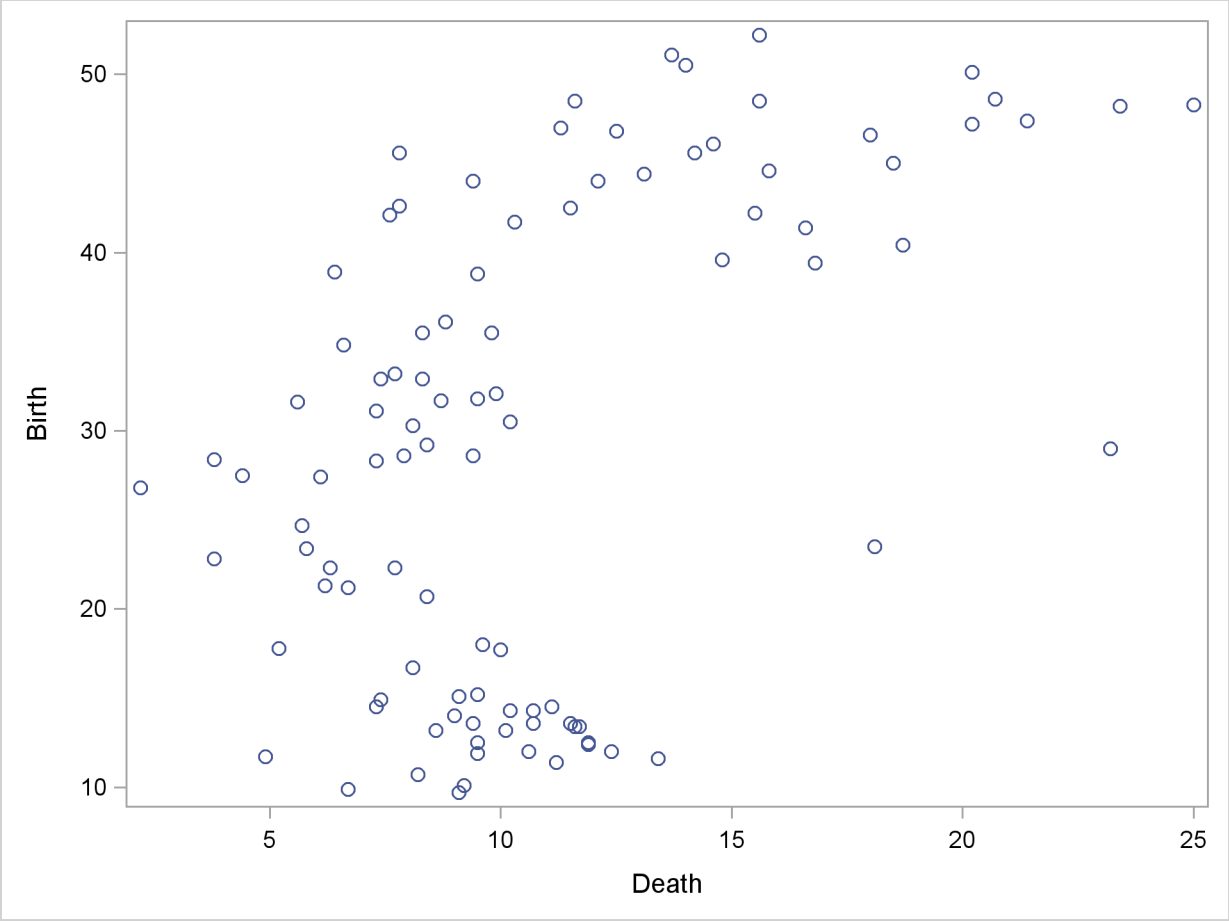
The data set Poverty contains the character variable Country and the numeric variables Birth, Death, and InfantDeath, which represent the birth rate per thousand, death rate per thousand, and infant death rate per thousand, respectively. The \$15. in the INPUT statement specifies that the variable Country is a character variable with a length of 15. The double trailing at sign (@@) in the INPUT statement specifies that observations are input from each line until all values have been read.

It is often useful when beginning a cluster analysis to look at the data graphically. The following statements use the SGPLOT procedure to make a scatter plot of the variables Birth and Death.

```
proc sgplot data=poverty;
   scatter x=Death y=Birth;
run;
```

The plot, displayed in Figure 22.1, indicates the difficulty of dividing the points into clusters. Plots of the other variable pairs (not shown) display similar characteristics. The clusters that comprise these data might be poorly separated and elongated. Data with poorly separated or elongated clusters must be transformed.

Figure 22.1 Scatter Plot of Original Poverty Data: Birth Rate versus Death Rate



If you know the within-cluster covariances, you can transform the data to make the clusters spherical. However, since you do not know what the clusters are, you cannot calculate exactly the within-cluster covariance matrix. The ACECLUS procedure estimates the within-cluster covariance matrix to transform the data, even when you have no knowledge of cluster membership or the number of clusters.

The following statements perform the ACECLUS procedure transformation by using the SAS data set Poverty:

```
proc aceclus data=poverty out=ace proportion=.03;
   var Birth Death InfantDeath;
run;
```

The OUT= option creates an output data set called Ace to contain the canonical variable scores. The PROPORTION= option specifies that approximately 3 percent of the pairs are included in the estimation of the within-cluster covariance matrix. The VAR statement specifies that the variables Birth, Death, and InfantDeath are used in computing the canonical variables.

The results of this analysis are displayed in Figure 22.2 through Figure 22.5.

Figure 22.2 displays the number of observations, the number of variables, and the settings for the PROPORTION and CONVERGE options. The PROPORTION option is set at 0.03, as specified in the previous statements. The CONVERGE parameter is set at its default value of 0.001. Figure 22.2 next displays the means, standard deviations, and sample covariance matrix of the analytical variables.

Figure 22.2 Means, Standard Deviations, and Covariance Matrix from the ACECLUS Procedure

The ACECLUS Procedure

Approximate Covariance Estimation for Cluster Analysis

Observations	97	Proportion	0.0300
Variables	3	Converge	0.00100

Means and Standard Deviations

Variable	Mean	Standard Deviation
Birth	29.2299	13.5467
Death	10.8361	4.6475
InfantDeath	54.9010	45.9926

COV: Total Sample Covariances

	Birth	Death	InfantDeath
Birth	183.512951	30.610056	534.794969
Death	30.610056	21.599205	139.925900
InfantDeath	534.794969	139.925900	2115.317811

The type of matrix used for the initial within-cluster covariance estimate is displayed in Figure 22.3. In this example, that initial estimate is the full covariance matrix. The threshold value that corresponds to the PROPORTION=0.03 setting is given as 0.292815.

Figure 22.3 Table of Iteration History from the ACECLUS Procedure

Initial Within-Cluster Covariance Estimate = Full Covariance Matrix

Threshold = 0.292815

Figure 22.3 *continued*

Iteration History

Iteration	RMS Distance	Distance Cutoff	Pairs Within Cutoff	Convergence Measure
1	2.449	0.717	385.0	0.552025
2	12.534	3.670	446.0	0.008406
3	12.851	3.763	521.0	0.009655
4	12.882	3.772	591.0	0.011193
5	12.716	3.723	628.0	0.008784
6	12.821	3.754	658.0	0.005553
7	12.774	3.740	680.0	0.003010
8	12.631	3.699	683.0	0.000676

Algorithm converged.

Figure 22.3 displays the iteration history. For each iteration, PROC ACECLUS displays the following measures:

- root mean square distance between all pairs of observations
- distance cutoff for including pairs of observations in the estimate of within-cluster covariances (equal to RMS*Threshold)
- number of pairs within the cutoff
- convergence measure

Figure 22.4 displays the approximate within-cluster covariance matrix and the table of eigenvalues from the canonical analysis. The first column of the eigenvalues table contains numbers for the eigenvectors. The next column of the table lists the eigenvalues of Inv(ACE)*(COV-ACE).

Figure 22.4 Approximate Within-Cluster Covariance Estimates

ACE: Approximate Covariance Estimate Within Clusters

	Birth	Death	InfantDeath
Birth	5.94644949	-0.63235725	6.28151537
Death	-0.63235725	2.33464129	1.59005857
InfantDeath	6.28151537	1.59005857	35.10327233

Eigenvalues of Inv(ACE)*(COV-ACE)

	Eigenvalue	Difference	Proportion	Cumulative
1	63.5500	54.7313	0.8277	0.8277
2	8.8187	4.4038	0.1149	0.9425
3	4.4149		0.0575	1.0000

The next three columns of the eigenvalue table (Figure 22.4) display measures of the relative size and importance of the eigenvalues. The first column lists the difference between each eigenvalue and its successor. The last two columns display the individual and cumulative proportions that each eigenvalue contributes to the total sum of eigenvalues.

The raw and standardized canonical coefficients are displayed in Figure 22.5. The coefficients are standardized by multiplying the raw coefficients with the standard deviation of the associated variable. The ACECLUS procedure uses these standardized canonical coefficients to create the transformed canonical variables, which are the linear transformations of the original input variables, Birth, Death, and InfantDeath.

Figure 22.5 Raw and Standardized Canonical Coefficients from the ACECLUS Procedure

Eigenvectors (Raw Canonical Coefficients)

	Can1	Can2	Can3
Birth	0.125610	0.457037	0.003875
Death	0.108402	0.163792	0.663538
InfantDeath	0.134704	-.133620	-.046266

Standardized Canonical Coefficients

	Can1	Can2	Can3
Birth	1.70160	6.19134	0.05249
Death	0.50380	0.76122	3.08379
InfantDeath	6.19540	-6.14553	-2.12790

The following statements invoke the CLUSTER procedure, using the SAS data set Ace created in the previous ACECLUS procedure:

```
proc cluster data=ace outtree=tree noprint method=ward;
   var can1 can2 can3 ;
   copy Birth--Country;
run;
```

The OUTTREE= option creates the output SAS data set Tree that is used in subsequent statements to draw a tree diagram. The NOPRINT option suppresses the display of the output. The METHOD= option specifies Ward's minimum-variance clustering method.

The VAR statement specifies that the canonical variables computed in the ACECLUS procedure are used in the cluster analysis. The COPY statement specifies that all the variables from the SAS data set Poverty (Birth—Country) are added to the output data set Tree.

The following statements use the TREE procedure to create an output SAS data set called New. The NCLUSTERS= option specifies the number of clusters desired in the SAS data set New. The NOPRINT option suppresses the display of the output.

```
proc tree data=tree out=new nclusters=3 noprint;
   copy Birth Death InfantDeath can1 can2 ;
   id Country;
run;
```

The COPY statement copies the canonical variables Can1 and Can2 (computed in the preceding ACECLUS procedure) and the original analytical variables Birth, Death, and InfantDeath into the output SAS data set New.

The following statements invoke the SGPLOT procedure, using the SAS data set created by PROC TREE:

```
proc sgplot data=new;
   scatter y=Birth x= Death / group= cluster;
   keylegend / title= "Cluster Membership";
run;
proc sgplot data=new;
   scatter y=can2 x=can1 / group= cluster;
   keylegend / title= "Cluster Membership";
run;
```

The first PROC SGPLOT statement requests a scatter plot of the two variables Birth and Death, using the variable CLUSTER as the identification variable.

The second PROC SGPLOT statement requests a plot of the two canonical variables, using the value of the variable CLUSTER as the identification variable.

Figure 22.6 and Figure 22.7 display the separation of the clusters when three clusters are calculated.

Figure 22.6 Scatter Plot of Poverty Data, Identified by Cluster

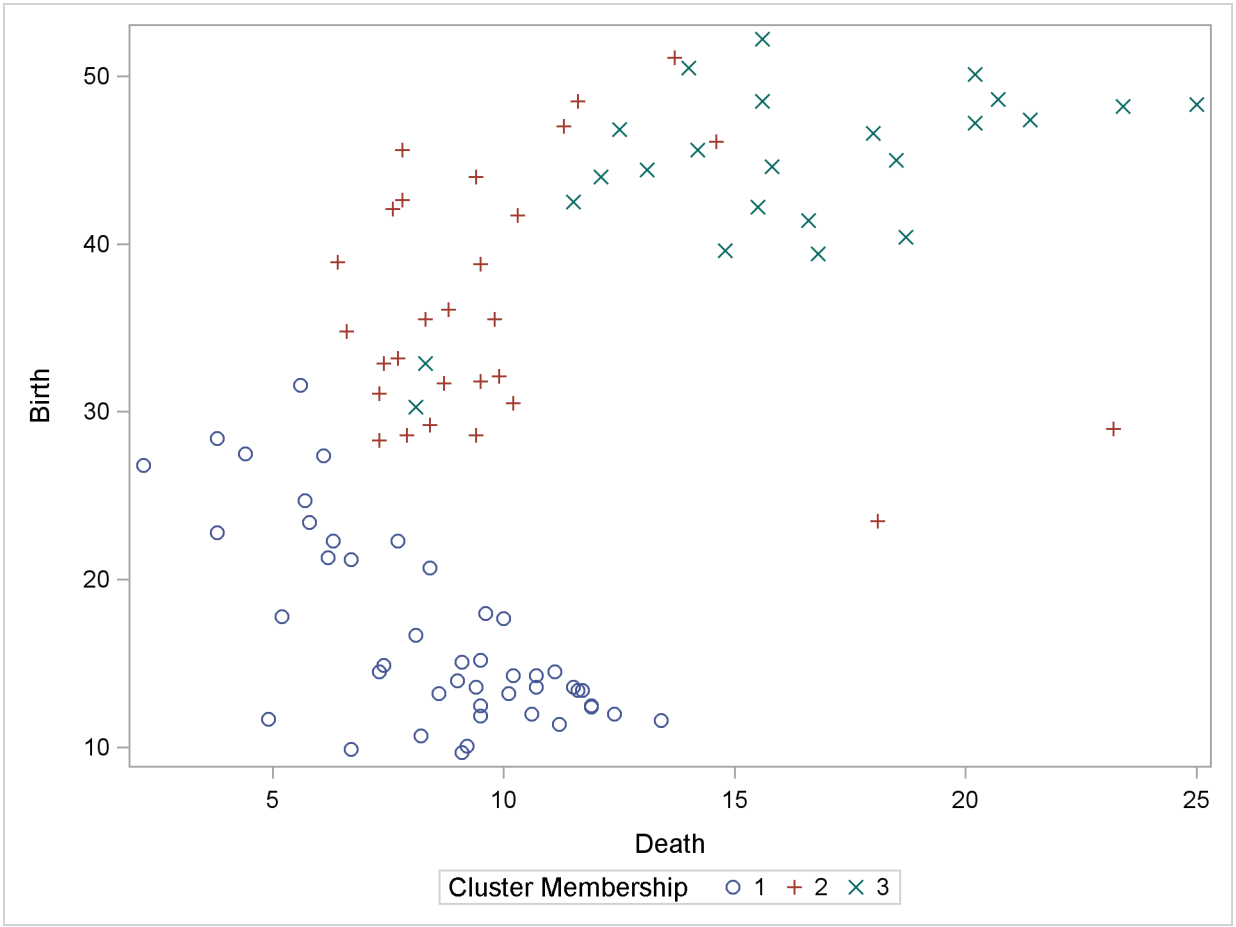
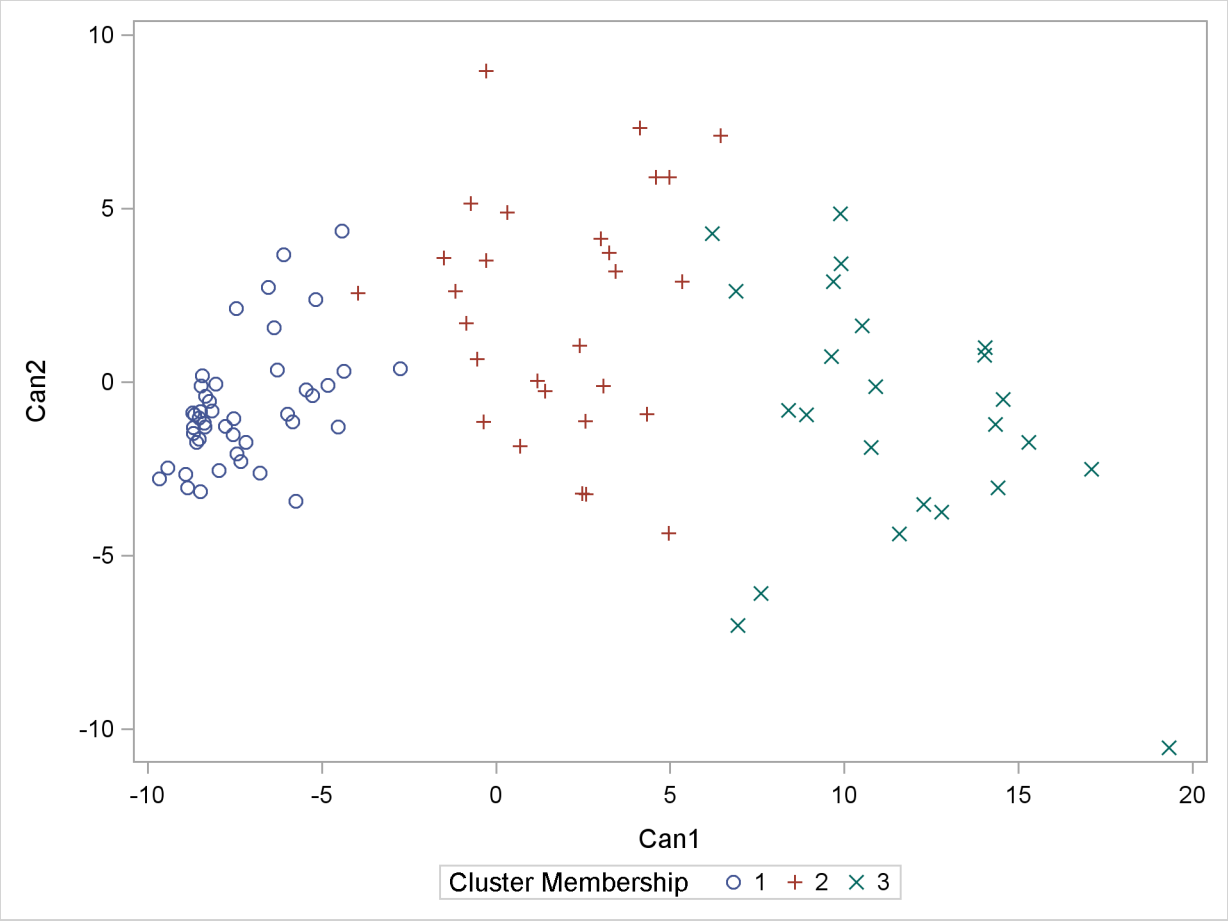


Figure 22.7 Scatter Plot of Canonical Variables



Syntax: ACECLUS Procedure

The following statements are available in the ACECLUS procedure:

PROC ACECLUS PROPORTION=*p* | **THRESHOLD=***t* < *options* > **;**
BY *variables* **;**
FREQ *variable* **;**
VAR *variables* **;**
WEIGHT *variable* **;**

Usually you need only the VAR statement in addition to the required PROC ACECLUS statement. The optional BY, FREQ, VAR, and WEIGHT statements are described in alphabetical order after the PROC ACECLUS statement.

PROC ACECLUS Statement

PROC ACECLUS PROPORTION=*p* | **THRESHOLD=***t* < *options* > **;**

The PROC ACECLUS statement starts the ACECLUS procedure. The options available with the PROC ACECLUS statement are summarized in Table 22.2 and discussed in the following sections. Note that, if you specify the METHOD=COUNT option, you must specify either the PROPORTION= or the MPAIRS= option. Otherwise, you must specify either the PROPORTION= or THRESHOLD= option.

Table 22.2 Summary of PROC ACECLUS Statement Options

Options	Description
Specify clustering options	
METHOD=	specifies the clustering method
MPAIRS=	specifies number of pairs for estimating within-cluster covariance (when you specify the option METHOD=COUNT)
PROPORTION=	specifies proportion of pairs for estimating within-cluster covariance
THRESHOLD=	specifies the threshold for including pairs in the estimation of the within-cluster covariance
Specify input and output data sets	
DATA=	specifies input data set name
OUT=	specifies output data set name
OUTSTAT=	specifies output data set name containing various statistics
Specify iteration options	
ABSOLUTE	uses absolute instead of relative threshold
CONVERGE=	specifies convergence criterion
INITIAL=	specifies initial estimate of within-cluster covariance matrix
MAXITER=	specifies maximum number of iterations
METRIC=	specifies metric in which computations are performed
SINGULAR=	specifies singularity criterion
Specify canonical analysis options	
N=	specifies number of canonical variables
PREFIX=	specifies prefix for naming canonical variables
Control displayed output	
NOPRINT	suppresses the display of the output
PP	produces PP-plot of distances between pairs from last iteration
QQ	produces QQ-plot of power transformation of distances between pairs from last iteration
SHORT	omits all output except for iteration history and eigenvalue table

The following list provides details about the options. The list is in alphabetical order.

ABSOLUTE

causes the THRESHOLD= value or the threshold computed from the PROPORTION= option to be treated absolutely rather than relative to the root mean square distance between observations. Use the ABSOLUTE option only when you are confident that the initial estimate of the within-cluster covariance matrix is close to the final estimate, such as when the INITIAL= option specifies a data set created by a previous execution of PROC ACECLUS by using the OUTSTAT= option.

CONVERGE=*c*

specifies the convergence criterion. By default, CONVERGE= 0.001. Iteration stops when the convergence measure falls below the value specified by the CONVERGE= option or when the iteration limit as specified by the MAXITER= option is exceeded, whichever happens first.

DATA=*SAS-data-set*

specifies the SAS data set to be analyzed. By default, PROC ACECLUS uses the most recently created SAS data set.

INITIAL=*name*

specifies the matrix for the initial estimate of the within-cluster covariance matrix. Valid values for *name* are as follows:

DIAGONAL | D
: uses the diagonal matrix of sample variances as the initial estimate of the within-cluster covariance matrix.

FULL | F
: uses the total-sample covariance matrix as the initial estimate of the within-cluster covariance matrix.

IDENTITY | I
: uses the identity matrix as the initial estimate of the within-cluster covariance matrix.

INPUT=*SAS-data-set*
: specifies a SAS data set from which to obtain the initial estimate of the within-cluster covariance matrix. The data set can be TYPE=CORR, COV, UCORR, UCOV, SSCP, or ACE, or it can be an ordinary SAS data set. See Appendix A, "Special SAS Data Sets," for descriptions of CORR, COV, UCORR, UCOV, and SSCP data sets. See the section "Output Data Sets" on page 680 for a description of ACE data sets.

  If you do not specify the INITIAL= option, the default is the matrix specified by the METRIC= option. If neither the INITIAL= nor the METRIC= option is specified, INITIAL=FULL is used if there are enough observations to obtain a nonsingular total-sample covariance matrix; otherwise, INITIAL=DIAGONAL is used.

MAXITER=*n*

specifies the maximum number of iterations. By default, MAXITER=10.

METHOD=COUNT | C | THRESHOLD | T

specifies the clustering method. The METHOD=THRESHOLD option requests a method (also the default) that uses all pairs closer than a given cutoff value to form the estimate at each iteration. The METHOD=COUNT option requests a method that uses a number of pairs, m, with the smallest distances to form the estimate at each iteration.

METRIC=*name*

specifies the metric in which the computations are performed, implies the default value for the INITIAL= option, and specifies the matrix $\mathbf{Z}$ used in the formula for the convergence measure e_i and for checking singularity of the $\mathbf{A}$ matrix. Valid values for *name* are as follows:

DIAGONAL | D	uses the diagonal matrix of sample variances diag($\mathbf{S}$) and sets $\mathbf{Z} = \text{diag}(\mathbf{S})^{-\frac{1}{2}}$, where the superscript $-\frac{1}{2}$ indicates an inverse factor.
FULL | F	uses the total-sample covariance matrix $\mathbf{S}$ and sets $\mathbf{Z} = \mathbf{S}^{-\frac{1}{2}}$.
IDENTITY | I	uses the identity matrix $\mathbf{I}$ and sets $\mathbf{Z} = \mathbf{I}$.

If you do not specify the METRIC= option, METRIC=FULL is used if there are enough observations to obtain a nonsingular total-sample covariance matrix; otherwise, METRIC=DIAGONAL is used.

The option METRIC= is rather technical. It affects the computations in a variety of ways, but for well-conditioned data the effects are subtle. For most data sets, the METRIC= option is not needed.

MPAIRS=*m*

specifies the number of pairs to be included in the estimation of the within-cluster covariance matrix when METHOD=COUNT is requested. The values of m must be greater than 0 but less than or equal to $(totfq \times (totfq-1))/2$, where *totfq* is the sum of nonmissing frequencies specified in the FREQ statement. If there is no FREQ statement, *totfq* equals the number of total nonmissing observations.

N=*n*

specifies the number of canonical variables to be computed. The default is the number of variables analyzed. N=0 suppresses the canonical analysis.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 20, "Using the Output Delivery System."

OUT=*SAS-data-set*

creates an output SAS data set that contains all the original data as well as the canonical variables having an estimated within-cluster covariance matrix equal to the identity matrix. If you want to create a permanent SAS data set, you must specify a two-level name. See *SAS Language Reference: Concepts* for more information about permanent SAS data sets.

OUTSTAT=*SAS-data-set*

specifies a TYPE=ACE output SAS data set that contains means, standard deviations, number of observations, covariances, estimated within-cluster covariances, eigenvalues, and canonical coefficients. If you want to create a permanent SAS data set, you must specify a two-level name. See *SAS Language Reference: Concepts* for more information about permanent SAS data sets.

PROPORTION=*p*

PERCENT=*p*

P=*p*

specifies the percentage of pairs to be included in the estimation of the within-cluster covariance matrix. The value of p must be greater than 0. If p is greater than or equal to 1, it is interpreted as a percentage and divided by 100; PROPORTION=0.02 and PROPORTION=2 are equivalent. When you specify METHOD=THRESHOLD, a threshold value is computed from the PROPORTION= option under the assumption that the observations are sampled from a multivariate normal distribution.

When you specify METHOD=COUNT, the number of pairs, m, is computed from PROPORTION=p as

$$m = \text{floor}\left(\frac{p}{2} \times totfq \times (totfq - 1)\right)$$

where *totfq* is the number of total nonmissing observations.

PP

produces a PP probability plot of distances between pairs of observations computed in the last iteration.

PREFIX=*name*

specifies a prefix for naming the canonical variables. By default the names are Can1, Can2, ..., CANN. If you specify PREFIX=ABC, the variables are named ABC1, ABC2, ABC3, and so on. The number of characters in the prefix plus the number of digits required to designate the variables should not exceed the name length defined by the VALIDVARNAME= system option. For more information about the VALIDVARNAME= system option, see *SAS Language Reference: Dictionary*.

QQ

produces a QQ probability plot of a power transformation of the distances between pairs of observations computed in the last iteration. **CAUTION:** The QQ plot can require an enormous amount of computer time.

SHORT

omits all items from the standard output except for the iteration history and the eigenvalue table.

SINGULAR=*g*

SING=*g*

specifies a singularity criterion $0 < g < 1$ for the total-sample covariance matrix $\mathbf{S}$ and the approximate within-cluster covariance estimate $\mathbf{A}$. The default is SINGULAR=1E−4.

THRESHOLD=*t*

T=*t*

specifies the threshold for including pairs of observations in the estimation of the within-cluster covariance matrix. A pair of observations is included if the Euclidean distance between them is less than or equal to t times the root mean square distance computed over all pairs of observations.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC ACECLUS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC ACECLUS. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

If you specify the INITIAL=INPUT= option and the INITIAL=INPUT= data set does not contain any of the BY variables, the entire INITIAL=INPUT= data set provides the initial value for the matrix **A** for each BY group in the DATA= data set.

If the INITIAL=INPUT= data set contains some but not all of the BY variables, or if some BY variables do not have the same type or length in the INITIAL=INPUT= data set as in the DATA= data set, then PROC ACECLUS displays an error message and stops.

If all the BY variables appear in the INITIAL=INPUT= data set with the same type and length as in the DATA= data set, then each BY group in the INITIAL=INPUT= data set provides the initial value for **A** for the corresponding BY group in the DATA= data set. All BY groups in the DATA= data set must also appear in the INITIAL= INPUT= data set. The BY groups in the INITIAL=INPUT= data set must be in the same order as in the DATA= data set. If you specify NOTSORTED in the BY statement, identical BY groups must occur in the same order in both data sets. If you do not specify NOTSORTED, some BY groups can appear in the INITIAL= INPUT= data set, but not in the DATA= data set; such BY groups are not used in the analysis.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide.*

FREQ Statement

FREQ *variable* **;**

If a variable in your data set represents the frequency of occurrence for the observation, include the name of that variable in the FREQ statement. The procedure then treats the data set as if each observation appears n times, where n is the value of the FREQ variable for the observation. If a value of the FREQ variable is not integral, it is truncated to the largest integer not exceeding the given value. Observations with FREQ values less than one are not included in the analysis. The total number of observations is considered equal to the sum of the FREQ variable.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the numeric variables to be analyzed. If the VAR statement is omitted, all numeric variables not specified in other statements are analyzed.

WEIGHT Statement

WEIGHT *variable* **;**

If you want to specify relative weights for each observation in the input data set, place the weights in a variable in the data set and specify that variable name in a WEIGHT statement. This is often done when the variance associated with each observation is different and the values of the weight variable are proportional to the reciprocals of the variances. The values of the WEIGHT variable can be nonintegral and are not truncated. An observation is used in the analysis only if the value of the WEIGHT variable is greater than zero.

The WEIGHT and FREQ statements have a similar effect, except in calculating the divisor of the **A** matrix.

Details: ACECLUS Procedure

Missing Values

Observations with missing values are omitted from the analysis and are given missing values for canonical variable scores in the OUT= data set.

Output Data Sets

OUT= Data Set

The OUT= data set contains all the variables in the original data set plus new variables containing the canonical variable scores. The N= option determines the number of new variables. The OUT= data set is not created if N=0. The names of the new variables are formed by concatenating the value given by the PREFIX= option (or the prefix CAN if the PREFIX= option is not specified) and the numbers 1, 2, 3, and so on. The OUT= data set can be used as input to PROC CLUSTER or PROC FASTCLUS. The cluster analysis should be performed on the canonical variables, not on the original variables.

OUTSTAT= Data Set

The OUTSTAT= data set is a TYPE=ACE data set containing the following variables:

- the BY variables, if any
- the two new character variables, _TYPE_ and _NAME_
- the variables analyzed—that is, those in the VAR statement, or, if there is no VAR statement, all numeric variables not listed in any other statement

Each observation in the new data set contains some type of statistic as indicated by the _TYPE_ variable. The values of the _TYPE_ variable are as follows:

MEAN	mean of each variable
STD	standard deviation of each variable
N	number of observations on which the analysis is based. This value is the same for each variable.
SUMWGT	sum of the weights if a WEIGHT statement is used. This value is the same for each variable.
COV	covariances between each variable and the variable named by the _NAME_ variable. The number of observations with _TYPE_=COV is equal to the number of variables being analyzed.
ACE	estimated within-cluster covariances between each variable and the variable named by the _NAME_ variable. The number of observations with _TYPE_=ACE is equal to the number of variables being analyzed.
EIGENVAL	eigenvalues of INV(ACE)*(COV−ACE). If the N= option requests fewer than the maximum number of canonical variables, only the specified number of eigenvalues are produced, with missing values filling out the observation.

RAWSCORE — raw canonical coefficients.

To obtain the canonical variable scores, these coefficients should be multiplied by the raw data centered by means obtained from the observation with _TYPE_='MEAN'.

SCORE — standardized canonical coefficients. The _NAME_ variable contains the name of the corresponding canonical variable as constructed from the PREFIX= option. The number of observations with _TYPE_=SCORE equals the number of canonical variables computed.

To obtain the canonical variable scores, these coefficients should be multiplied by the standardized data, using means obtained from the observation with _TYPE_='MEAN' and standard deviations obtained from the observation with _TYPE_='STD'.

The OUTSTAT= data set can be used in the following conditions:

- to initialize another execution of PROC ACECLUS
- to compute canonical variable scores with the SCORE procedure
- as input to the FACTOR procedure, specifying METHOD=SCORE, to rotate the canonical variables

Computational Resources

Let

n = number of observations

v = number of variables

i = number of iterations

Memory

The memory in bytes required by PROC ACECLUS is approximately

$$8(2n(v+1)+21v+5v^2)$$

bytes. If you request the PP or QQ option, an additional $4n(n-1)$ bytes are needed.

Time

The time required by PROC ACECLUS is roughly proportional to

$$2nv^2+10v^3+i\left(\frac{n^2v}{2}+nv^2+5v^3\right)$$

Displayed Output

Unless the SHORT option is specified, the ACECLUS procedure displays the following items:

- Means and Standard Deviations of the input variables
- the **S** matrix, labeled COV: Total Sample Covariances
- the name or value of the matrix used for the Initial Within-Cluster Covariance Estimate
- the Threshold value if the PROPORTION= option is specified

For each iteration, PROC ACECLUS displays the following items:

- the Iteration number
- RMS Distance, the root mean square distance between all pairs of observations
- the Distance Cutoff (u) for including pairs of observations in the estimate of the within-cluster covariances, which equals the RMS distance times the threshold
- the number of Pairs Within Cutoff
- the Convergence Measure (e_i) as specified by the METRIC= option

If the SHORT option is not specified, PROC ACECLUS also displays the **A** matrix, labeled ACE: Approximate Covariance Estimate Within Clusters.

The ACECLUS procedure displays a table of eigenvalues from the canonical analysis containing the following items:

- Eigenvalues of Inv(ACE)*(COV−ACE)
- the Difference between successive eigenvalues
- the Proportion of variance explained by each eigenvalue
- the Cumulative proportion of variance explained

If the SHORT option is not specified, PROC ACECLUS displays the following items:

- the Eigenvectors or raw canonical coefficients
- the standardized eigenvectors or standard canonical coefficients

ODS Table Names

PROC ACECLUS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 22.3 ODS Tables Produced by PROC ACECLUS

ODS Table Name	Description	Statement	Option
ConvergenceStatus	Convergence status	PROC	default
DataOptionInfo	Data and option information	PROC	default
Eigenvalues	Eigenvalues of Inv(ACE)*(COV–ACE)	PROC	default
Eigenvectors	Eigenvectors (raw canonical coefficients)	PROC	default
InitWithin	Initial within-cluster covariance estimate	PROC	INITIAL=INPUT
IterHistory	Iteration history	PROC	default
SimpleStatistics	Simple statistics	PROC	default
StdCanCoef	Standardized canonical coefficients	PROC	default
Threshold	Threshold value	PROC	PROPORTION=
TotSampleCov	Total sample covariances	PROC	default
Within	Approximate covariance estimate within clusters	PROC	default

Example: ACECLUS Procedure

Example 22.1: Transformation and Cluster Analysis of Fisher Iris Data

The iris data published by Fisher (1936) have been widely used for examples in discriminant analysis and cluster analysis. The sepal length, sepal width, petal length, and petal width are measured in millimeters on 50 iris specimens from each of three species, *Iris setosa, I. versicolor,* and *I. virginica*. Mezzich and Solomon (1980) discuss a variety of cluster analyses of the iris data.

In this example PROC ACECLUS is used to transform the data, and the clustering is performed by PROC FASTCLUS. Compare this with the example in Chapter 34, "The FASTCLUS Procedure." The results from the FREQ procedure display fewer misclassifications when PROC ACECLUS is used. The following statements produce Output 22.1.1 through Output 22.1.5.

```
proc format;
   value specname
      1='Setosa    '
      2='Versicolor'
      3='Virginica ';
run;

data iris;
   title 'Fisher (1936) Iris Data';
   input SepalLength SepalWidth PetalLength PetalWidth Species @@;
   format Species specname.;
   label SepalLength='Sepal Length in mm.'
       SepalWidth ='Sepal Width in mm.'
         PetalLength='Petal Length in mm.'
         PetalWidth ='Petal Width in mm.';
   symbol = put(species, specname10.);
   datalines;
50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3
63 28 51 15 3 46 34 14 03 1 69 31 51 23 3 62 22 45 15 2
59 32 48 18 2 46 36 10 02 1 61 30 46 14 2 60 27 51 16 2
65 30 52 20 3 56 25 39 11 2 65 30 55 18 3 58 27 51 19 3
68 32 59 23 3 51 33 17 05 1 57 28 45 13 2 62 34 54 23 3
77 38 67 22 3 63 33 47 16 2 67 33 57 25 3 76 30 66 21 3
49 25 45 17 3 55 35 13 02 1 67 30 52 23 3 70 32 47 14 2
64 32 45 15 2 61 28 40 13 2 48 31 16 02 1 59 30 51 18 3
55 24 38 11 2 63 25 50 19 3 64 32 53 23 3 52 34 14 02 1
49 36 14 01 1 54 30 45 15 2 79 38 64 20 3 44 32 13 02 1
67 33 57 21 3 50 35 16 06 1 58 26 40 12 2 44 30 13 02 1
77 28 67 20 3 63 27 49 18 3 47 32 16 02 1 55 26 44 12 2
50 23 33 10 2 72 32 60 18 3 48 30 14 03 1 51 38 16 02 1
61 30 49 18 3 48 34 19 02 1 50 30 16 02 1 50 32 12 02 1
61 26 56 14 3 64 28 56 21 3 43 30 11 01 1 58 40 12 02 1
51 38 19 04 1 67 31 44 14 2 62 28 48 18 3 49 30 14 02 1
51 35 14 02 1 56 30 45 15 2 58 27 41 10 2 50 34 16 04 1
46 32 14 02 1 60 29 45 15 2 57 26 35 10 2 57 44 15 04 1
50 36 14 02 1 77 30 61 23 3 63 34 56 24 3 58 27 51 19 3
57 29 42 13 2 72 30 58 16 3 54 34 15 04 1 52 41 15 01 1
71 30 59 21 3 64 31 55 18 3 60 30 48 18 3 63 29 56 18 3
49 24 33 10 2 56 27 42 13 2 57 30 42 12 2 55 42 14 02 1
49 31 15 02 1 77 26 69 23 3 60 22 50 15 3 54 39 17 04 1
66 29 46 13 2 52 27 39 14 2 60 34 45 16 2 50 34 15 02 1
44 29 14 02 1 50 20 35 10 2 55 24 37 10 2 58 27 39 12 2
47 32 13 02 1 46 31 15 02 1 69 32 57 23 3 62 29 43 13 2
74 28 61 19 3 59 30 42 15 2 51 34 15 02 1 50 35 13 03 1
56 28 49 20 3 60 22 40 10 2 73 29 63 18 3 67 25 58 18 3
49 31 15 01 1 67 31 47 15 2 63 23 44 13 2 54 37 15 02 1
56 30 41 13 2 63 25 49 15 2 61 28 47 12 2 64 29 43 13 2
51 25 30 11 2 57 28 41 13 2 65 30 58 22 3 69 31 54 21 3
54 39 13 04 1 51 35 14 03 1 72 36 61 25 3 65 32 51 20 3
61 29 47 14 2 56 29 36 13 2 69 31 49 15 2 64 27 53 19 3
68 30 55 21 3 55 25 40 13 2 48 34 16 02 1 48 30 14 01 1
45 23 13 03 1 57 25 50 20 3 57 38 17 03 1 51 38 15 03 1
55 23 40 13 2 66 30 44 14 2 68 28 48 14 2 54 34 17 02 1
51 37 15 04 1 52 35 15 02 1 58 28 51 24 3 67 30 50 17 2
63 33 60 25 3 53 37 15 02 1
;
```

```
proc aceclus data=iris out=ace p=.02 outstat=score;
   var SepalLength SepalWidth PetalLength PetalWidth ;
run;
proc sgplot data=ace;
   scatter y=can2 x=can1 / group= Species;
   keylegend / title= "Species";
   format Species specname. ;
run;

proc fastclus data=ace maxc=3 maxiter=10 conv=0 out=clus;
   var can:;
run;
proc freq;
   tables cluster*Species;
run;
```

Output 22.1.1 Using PROC ACECLUS to Transform Fisher's Iris Data

Fisher (1936) Iris Data

The ACECLUS Procedure

Approximate Covariance Estimation for Cluster Analysis

Observations	150	Proportion	0.0200
Variables	4	Converge	0.00100

Means and Standard Deviations

Variable	Mean	Standard Deviation	Label
SepalLength	58.4333	8.2807	Sepal Length in mm.
SepalWidth	30.5733	4.3587	Sepal Width in mm.
PetalLength	37.5800	17.6530	Petal Length in mm.
PetalWidth	11.9933	7.6224	Petal Width in mm.

COV: Total Sample Covariances

	SepalLength	SepalWidth	PetalLength	PetalWidth
SepalLength	68.5693512	-4.2434004	127.4315436	51.6270694
SepalWidth	-4.2434004	18.9979418	-32.9656376	-12.1639374
PetalLength	127.4315436	-32.9656376	311.6277852	129.5609396
PetalWidth	51.6270694	-12.1639374	129.5609396	58.1006264

Initial Within-Cluster Covariance Estimate = Full Covariance Matrix

Threshold = 0.334211

Output 22.1.1 *continued*

Iteration History

Iteration	RMS Distance	Distance Cutoff	Pairs Within Cutoff	Convergence Measure
1	2.828	0.945	408.0	0.465775
2	11.905	3.979	559.0	0.013487
3	13.152	4.396	940.0	0.029499
4	13.439	4.491	1506.0	0.046846
5	13.271	4.435	2036.0	0.046859
6	12.591	4.208	2285.0	0.025027
7	12.199	4.077	2366.0	0.009559
8	12.121	4.051	2402.0	0.003895
9	12.064	4.032	2417.0	0.002051
10	12.047	4.026	2429.0	0.000971

Algorithm converged.

Output 22.1.2 Eigenvalues, Raw Canonical Coefficients, and Standardized Canonical Coefficients

ACE: Approximate Covariance Estimate Within Clusters

	SepalLength	SepalWidth	PetalLength	PetalWidth
SepalLength	11.73342939	5.47550432	4.95389049	2.02902429
SepalWidth	5.47550432	6.91992590	2.42177851	1.74125154
PetalLength	4.95389049	2.42177851	6.53746398	2.35302594
PetalWidth	2.02902429	1.74125154	2.35302594	2.05166735

Eigenvalues of Inv(ACE)*(COV-ACE)

	Eigenvalue	Difference	Proportion	Cumulative
1	63.7716	61.1593	0.9367	0.9367
2	2.6123	1.5561	0.0384	0.9751
3	1.0562	0.4167	0.0155	0.9906
4	0.6395		0.00939	1.0000

Eigenvectors (Raw Canonical Coefficients)

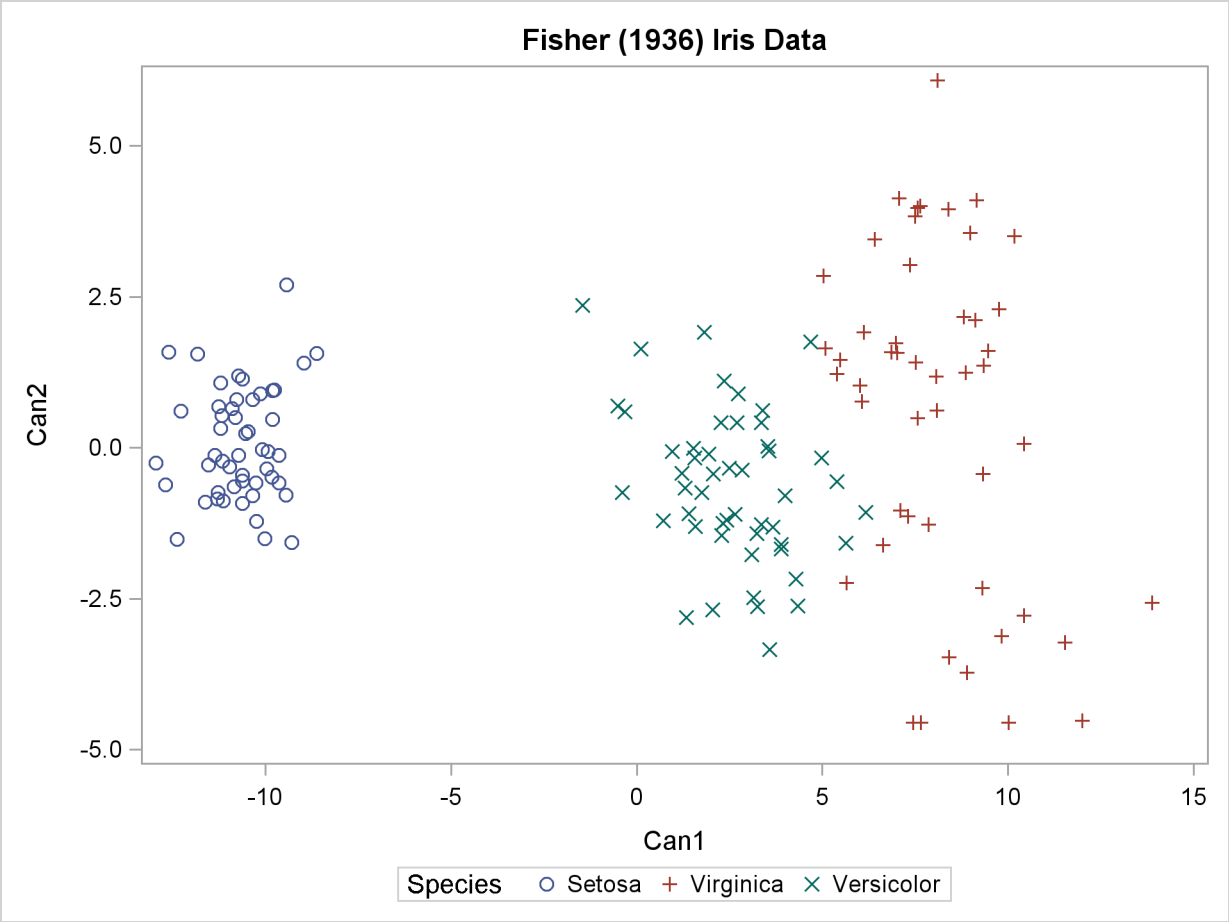
		Can1	Can2	Can3	Can4
SepalLength	Sepal Length in mm.	-.012009	-.098074	-.059852	0.402352
SepalWidth	Sepal Width in mm.	-.211068	-.000072	0.402391	-.225993
PetalLength	Petal Length in mm.	0.324705	-.328583	0.110383	-.321069
PetalWidth	Petal Width in mm.	0.266239	0.870434	-.085215	0.320286

Output 22.1.2 *continued*

Standardized Canonical Coefficients

		Can1	Can2	Can3	Can4
SepalLength	Sepal Length in mm.	-0.09944	-0.81211	-0.49562	3.33174
SepalWidth	Sepal Width in mm.	-0.91998	-0.00031	1.75389	-0.98503
PetalLength	Petal Length in mm.	5.73200	-5.80047	1.94859	-5.66782
PetalWidth	Petal Width in mm.	2.02937	6.63478	-0.64954	2.44134

Output 22.1.3 Plot of Transformed Iris Data: PROC SGPLOT



Output 22.1.4 Clustering of Transformed Iris Data: Partial Output from PROC FASTCLUS

Fisher (1936) Iris Data

The FASTCLUS Procedure
Replace=FULL Radius=0 Maxclusters=3 Maxiter=10 Converge=0

Cluster Summary

Cluster	Frequency	RMS Std Deviation	Maximum Distance from Seed to Observation	Radius Exceeded	Nearest Cluster
1	50	1.1016	5.2768		3
2	50	1.8880	6.8298		3
3	50	1.4138	5.3152		2

Cluster Summary

Cluster	Distance Between Cluster Centroids
1	13.2845
2	5.8580
3	5.8580

Statistics for Variables

Variable	Total STD	Within STD	R-Square	RSQ/(1-RSQ)
Can1	8.04808	1.48537	0.966394	28.756658
Can2	1.90061	1.85646	0.058725	0.062389
Can3	1.43395	1.32518	0.157417	0.186826
Can4	1.28044	1.27550	0.021025	0.021477
OVER-ALL	4.24499	1.50298	0.876324	7.085666

Pseudo F Statistic = 520.80

Approximate Expected Over-All R-Squared = 0.80391

Cubic Clustering Criterion = 5.179

WARNING: The two values above are invalid for correlated variables.

Cluster Means

Cluster	Can1	Can2	Can3	Can4
1	-10.67516964	0.06706906	0.27068819	0.11164209
2	8.12988211	0.52566663	0.51836499	0.14915404
3	2.54528754	-0.59273569	-0.78905317	-0.26079612

Output 22.1.4 *continued*

Cluster Standard Deviations

Cluster	Can1	Can2	Can3	Can4
1	0.953761025	0.931943571	1.398456061	1.058217627
2	1.799159552	2.743869556	1.270344142	1.370523175
3	1.572366584	1.393565864	1.303411851	1.372050319

Output 22.1.5 Crosstabulation of Cluster by Species for Fisher's Iris Data: PROC FREQ

Fisher (1936) Iris Data

The FREQ Procedure

Table of CLUSTER by Species

CLUSTER(Cluster) Species

Frequency / Percent / Row Pct / Col Pct	Setosa	Versicolor	Virginica	Total
1	50	0	0	50
	33.33	0.00	0.00	33.33
	100.00	0.00	0.00	
	100.00	0.00	0.00	
2	0	2	48	50
	0.00	1.33	32.00	33.33
	0.00	4.00	96.00	
	0.00	4.00	96.00	
3	0	48	2	50
	0.00	32.00	1.33	33.33
	0.00	96.00	4.00	
	0.00	96.00	4.00	
Total	50	50	50	150
	33.33	33.33	33.33	100.00

References

Art, D., Gnanadesikan, R., and Kettenring, R. (1982), "Data-Based Metrics for Cluster Analysis," *Utilitas Mathematica*, 21A, 75–99.

Everitt, B. S. (1980), *Cluster Analysis*, Second Edition, London: Heineman Educational Books.

Fisher, R. A. (1936), "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, 7, 179–188.

Hartigan, J. A. (1975), *Clustering Algorithms*, New York: John Wiley & Sons.

Kettenring, R. (1984), personal communication.

Mezzich, J. E and Solomon, H. (1980), *Taxonomy and Behavioral Science*, New York: Academic Press.

Puri, M. L. and Sen, P. K. (1971), *Nonparametric Methods in Multivariate Analysis,* New York: John Wiley & Sons.

Rouncefield, M. (1995), "The Statistics of Poverty and Inequality," *Journal of Statistics Education*, 3(2). [Online]: [http://www.stat.ncsu.edu/info/jse], accessed December 19, 1997.

Wolfe, J. H. (1970), "Pattern Clustering by Multivariate Mixture Analysis," *Multivariate Behavioral Research*, 5, 329–350.

Chapter 23

The ANOVA Procedure

Contents

Overview: ANOVA Procedure **692**
Getting Started: ANOVA Procedure **692**
One-Way Layout with Means Comparisons 693
Randomized Complete Block with One Factor 697
Syntax: ANOVA Procedure **701**
PROC ANOVA Statement 702
ABSORB Statement 703
BY Statement 704
CLASS Statement 705
FREQ Statement 705
MANOVA Statement 706
MEANS Statement 709
MODEL Statement 715
REPEATED Statement 715
TEST Statement 719
Details: ANOVA Procedure **720**
Specification of Effects 720
Using PROC ANOVA Interactively 723
Missing Values 724
Output Data Set 724
Computational Method 725
Displayed Output 726
ODS Table Names 727
ODS Graphics 729
Examples: ANOVA Procedure **730**
Example 23.1: Factorial Treatments in Complete Blocks 730
Example 23.2: Alternative Multiple Comparison Procedures 732
Example 23.3: Split Plot 737
Example 23.4: Latin Square Split Plot 739
Example 23.5: Strip-Split Plot 741
References **746**

Overview: ANOVA Procedure

The ANOVA procedure performs *analysis of variance* (ANOVA) for balanced data from a wide variety of experimental designs. In analysis of variance, a continuous response variable, known as a *dependent variable*, is measured under experimental conditions identified by classification variables, known as *independent variables*. The variation in the response is assumed to be due to effects in the classification, with random error accounting for the remaining variation.

The ANOVA procedure is one of several procedures available in SAS/STAT software for analysis of variance. The ANOVA procedure is designed to handle balanced data (that is, data with equal numbers of observations for every combination of the classification factors), whereas the GLM procedure can analyze both balanced and unbalanced data. Because PROC ANOVA takes into account the special structure of a balanced design, it is faster and uses less storage than PROC GLM for balanced data.

Use PROC ANOVA for the analysis of balanced data only, with the following exceptions: one-way analysis of variance, Latin square designs, certain partially balanced incomplete block designs, completely nested (hierarchical) designs, and designs with cell frequencies that are proportional to each other and are also proportional to the background population. These exceptions have designs in which the factors are all orthogonal to each other.

For further discussion, see Searle (1971, p. 138). PROC ANOVA works for designs with block diagonal $\mathbf{X}'\mathbf{X}$ matrices where the elements of each block all have the same value. The procedure partially tests this requirement by checking for equal cell means. However, this test is imperfect: some designs that cannot be analyzed correctly might pass the test, and designs that can be analyzed correctly might not pass. If your design does not pass the test, PROC ANOVA produces a warning message to tell you that the design is unbalanced and that the ANOVA analyses might not be valid; if your design is not one of the special cases described here, then you should use PROC GLM instead. Complete validation of designs is not performed in PROC ANOVA since this would require the whole $\mathbf{X}'\mathbf{X}$ matrix; if you're unsure about the validity of PROC ANOVA for your design, you should use PROC GLM.

CAUTION: If you use PROC ANOVA for analysis of unbalanced data, you must assume responsibility for the validity of the results.

The ANOVA procedure automatically produces graphics as part of its ODS output. For general information about ODS graphics, see the section "ODS Graphics" on page 729 and Chapter 21, "Statistical Graphics Using ODS."

Getting Started: ANOVA Procedure

The following examples demonstrate how you can use the ANOVA procedure to perform analyses of variance for a one-way layout and a randomized complete block design.

One-Way Layout with Means Comparisons

A one-way analysis of variance considers one treatment factor with two or more treatment levels. The goal of the analysis is to test for differences among the means of the levels and to quantify these differences. If there are two treatment levels, this analysis is equivalent to a t test comparing two group means.

The assumptions of analysis of variance (Steel and Torrie 1980) are that treatment effects are additive and experimental errors are independently random with a normal distribution that has mean zero and constant variance.

The following example studies the effect of bacteria on the nitrogen content of red clover plants. The treatment factor is bacteria strain, and it has six levels. Five of the six levels consist of five different *Rhizobium trifolii* bacteria cultures combined with a composite of five *Rhizobium meliloti* strains. The sixth level is a composite of the five *Rhizobium trifolii* strains with the composite of the *Rhizobium meliloti*. Red clover plants are inoculated with the treatments, and nitrogen content is later measured in milligrams. The data are derived from an experiment by Erdman (1946) and are analyzed in Chapters 7 and 8 of Steel and Torrie (1980). The following DATA step creates the SAS data set Clover:

```
title1 'Nitrogen Content of Red Clover Plants';
data Clover;
   input Strain $ Nitrogen @@;
   datalines;
3DOK1  19.4 3DOK1  32.6 3DOK1  27.0 3DOK1  32.1 3DOK1  33.0
3DOK5  17.7 3DOK5  24.8 3DOK5  27.9 3DOK5  25.2 3DOK5  24.3
3DOK4  17.0 3DOK4  19.4 3DOK4   9.1 3DOK4  11.9 3DOK4  15.8
3DOK7  20.7 3DOK7  21.0 3DOK7  20.5 3DOK7  18.8 3DOK7  18.6
3DOK13 14.3 3DOK13 14.4 3DOK13 11.8 3DOK13 11.6 3DOK13 14.2
COMPOS 17.3 COMPOS 19.4 COMPOS 19.1 COMPOS 16.9 COMPOS 20.8
;
```

The variable Strain contains the treatment levels, and the variable Nitrogen contains the response. The following statements produce the analysis.

```
proc anova data = Clover;
   class strain;
   model Nitrogen = Strain;
run;
```

The classification variable is specified in the CLASS statement. Note that, unlike the GLM procedure, PROC ANOVA does not allow continuous variables on the right-hand side of the model. Figure 23.1 and Figure 23.2 display the output produced by these statements.

Figure 23.1 Class Level Information

```
                    Nitrogen Content of Red Clover Plants

                              The ANOVA Procedure

                            Class Level Information

          Class          Levels    Values

          Strain              6    3DOK1 3DOK13 3DOK4 3DOK5 3DOK7 COMPOS

                     Number of Observations Read          30
                     Number of Observations Used          30
```

The "Class Level Information" table shown in Figure 23.1 lists the variables that appear in the CLASS statement, their levels, and the number of observations in the data set.

Figure 23.2 displays the ANOVA table, followed by some simple statistics and tests of effects.

Figure 23.2 ANOVA Table

```
                    Nitrogen Content of Red Clover Plants

                              The ANOVA Procedure

Dependent Variable: Nitrogen

                                        Sum of
 Source                       DF        Squares    Mean Square   F Value   Pr > F

 Model                         5     847.046667     169.409333     14.37   <.0001

 Error                        24     282.928000      11.788667

 Corrected Total              29    1129.974667

             R-Square     Coeff Var      Root MSE    Nitrogen Mean

             0.749616      17.26515      3.433463         19.88667

 Source                       DF       Anova SS    Mean Square   F Value   Pr > F

 Strain                        5    847.0466667    169.4093333     14.37   <.0001
```

The degrees of freedom (DF) column should be used to check the analysis results. The model degrees of freedom for a one-way analysis of variance are the number of levels minus 1; in this case, $6 - 1 = 5$. The Corrected Total degrees of freedom are always the total number of observations minus one; in this case $30 - 1 = 29$. The sum of Model and Error degrees of freedom equal the Corrected Total.

The overall F test is significant ($F = 14.37$, $p < 0.0001$), indicating that the model as a whole

accounts for a significant portion of the variability in the dependent variable. The F test for Strain is significant, indicating that some contrast between the means for the different strains is different from zero. Notice that the Model and Strain F tests are identical, since Strain is the only term in the model.

The F test for Strain ($F = 14.37, p < 0.0001$) suggests that there are differences among the bacterial strains, but it does not reveal any information about the nature of the differences. Mean comparison methods can be used to gather further information. The interactivity of PROC ANOVA enables you to do this without re-running the entire analysis. After you specify a model with a MODEL statement and execute the ANOVA procedure with a RUN statement, you can execute a variety of statements (such as MEANS, MANOVA, TEST, and REPEATED) without PROC ANOVA recalculating the model sum of squares.

The following command requests means of the Strain levels with Tukey's studentized range procedure.

```
means strain / tukey;
```

Results of Tukey's procedure are shown in Figure 23.3.

Figure 23.3 Tukey's Multiple Comparisons Procedure

Nitrogen Content of Red Clover Plants

The ANOVA Procedure

Tukey's Studentized Range (HSD) Test for Nitrogen

Alpha	0.05
Error Degrees of Freedom	24
Error Mean Square	11.78867
Critical Value of Studentized Range	4.37265
Minimum Significant Difference	6.7142

Means with the same letter are not significantly different.

Tukey Grouping	Mean	N	Strain
A	28.820	5	3DOK1
A			
B A	23.980	5	3DOK5
B			
B C	19.920	5	3DOK7
B C			
B C	18.700	5	COMPOS
C			
C	14.640	5	3DOK4
C			
C	13.260	5	3DOK13

Examples of implications of the multiple comparisons results are as follows:

- Strain 3DOK1 fixes significantly more nitrogen than all but 3DOK5.
- While 3DOK5 is not significantly different from 3DOK1, it is also not significantly better than all the rest, though it is better than the bottom two groups.

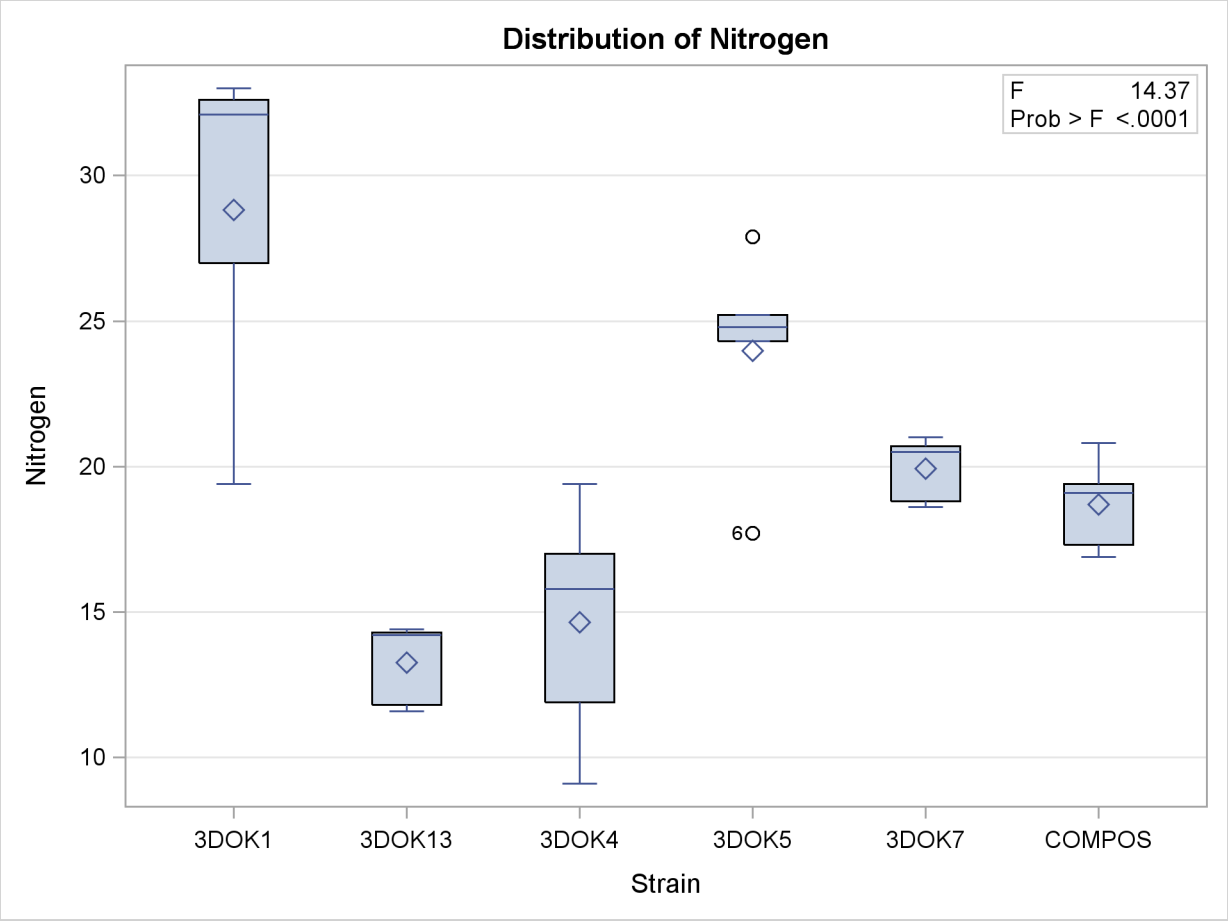
Although the experiment has succeeded in separating the best strains from the worst, more experimentation is required in order to clearly distinguish the very best strain.

If you enable ODS Graphics, ANOVA also displays by default a plot that enables you to visualize the distribution of nitrogen content for each treatment. The following statements, which are the same as the previous analysis but with ODS graphics enabled, additionally produce Figure 23.4.

```
ods graphics on;
proc anova data = Clover;
   class strain;
   model Nitrogen = Strain;
run;
ods graphics off;
```

When you specify the ODS GRAPHICS statement and fit a one-way analysis of variance model, the ANOVA procedure output includes a box plot of the dependent variable values within each classification level of the independent variable. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the ANOVA procedure, see the section "ODS Graphics" on page 729.

Figure 23.4 Box Plot of Nitrogen Content for each Treatment



Randomized Complete Block with One Factor

This example illustrates the use of PROC ANOVA in analyzing a randomized complete block design. Researchers are interested in whether three treatments have different effects on the yield and worth of a particular crop. They believe that the experimental units are not homogeneous. So, a blocking factor is introduced that allows the experimental units to be homogeneous within each block. The three treatments are then randomly assigned within each block.

The data from this study are input into the SAS data set RCB:

```
title1 'Randomized Complete Block';
data RCB;
   input Block Treatment $ Yield Worth @@;
   datalines;
1 A 32.6 112   1 B 36.4 130   1 C 29.5 106
2 A 42.7 139   2 B 47.1 143   2 C 32.9 112
3 A 35.3 124   3 B 40.1 134   3 C 33.6 116
;
```

The variables Yield and Worth are continuous response variables, and the variables Block and Treatment are the classification variables. Because the data for the analysis are balanced, you can use PROC ANOVA to run the analysis.

The statements for the analysis are

```
proc anova data=RCB;
   class Block Treatment;
   model Yield Worth=Block Treatment;
run;
```

The Block and Treatment effects appear in the CLASS statement. The MODEL statement requests an analysis for each of the two dependent variables, Yield and Worth.

Figure 23.5 shows the "Class Level Information" table.

Figure 23.5 Class Level Information

```
                    Randomized Complete Block

                       The ANOVA Procedure

                     Class Level Information

                Class          Levels    Values

                Block               3    1 2 3

                Treatment           3    A B C

              Number of Observations Read           9
              Number of Observations Used           9
```

The "Class Level Information" table lists the number of levels and their values for all effects specified in the CLASS statement. The number of observations in the data set are also displayed. Use this information to make sure that the data have been read correctly.

The overall ANOVA table for Yield in Figure 23.6 appears first in the output because it is the first response variable listed on the left side in the MODEL statement.

Figure 23.6 Overall ANOVA Table for Yield

Randomized Complete Block

The ANOVA Procedure

Dependent Variable: Yield

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	225.2777778	56.3194444	8.94	0.0283
Error	4	25.1911111	6.2977778		
Corrected Total	8	250.4688889			

R-Square	Coeff Var	Root MSE	Yield Mean
0.899424	6.840047	2.509537	36.68889

The overall F statistic is significant ($F = 8.94$, $p = 0.0283$), indicating that the model as a whole accounts for a significant portion of the variation in Yield and that you can proceed to evaluate the tests of effects.

The degrees of freedom (DF) are used to ensure correctness of the data and model. The Corrected Total degrees of freedom are one less than the total number of observations in the data set; in this case, $9-1 = 8$. The Model degrees of freedom for a randomized complete block are $(b-1)+(t-1)$, where b =number of block levels and t =number of treatment levels. In this case, this formula leads to $(3-1)+(3-1) = 4$ model degrees of freedom.

Several simple statistics follow the ANOVA table. The R-Square indicates that the model accounts for nearly 90% of the variation in the variable Yield. The coefficient of variation (C.V.) is listed along with the Root MSE and the mean of the dependent variable. The Root MSE is an estimate of the standard deviation of the dependent variable. The C.V. is a unitless measure of variability.

The tests of the effects shown in Figure 23.7 are displayed after the simple statistics.

Figure 23.7 Tests of Effects for Yield

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Block	2	98.1755556	49.0877778	7.79	0.0417
Treatment	2	127.1022222	63.5511111	10.09	0.0274

For Yield, both the Block and Treatment effects are significant ($F = 7.79$, $p = 0.0417$ and $F = 10.09$, $p = 0.0274$, respectively) at the 95% level. From this you can conclude that blocking is useful for this variable and that some contrast between the treatment means is significantly different from zero.

Figure 23.8 shows the ANOVA table, simple statistics, and tests of effects for the variable Worth.

Figure 23.8 ANOVA Table for Worth

Randomized Complete Block

The ANOVA Procedure

Dependent Variable: Worth

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	1247.333333	311.833333	8.28	0.0323
Error	4	150.666667	37.666667		
Corrected Total	8	1398.000000			

R-Square	Coeff Var	Root MSE	Worth Mean
0.892227	4.949450	6.137318	124.0000

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Block	2	354.6666667	177.3333333	4.71	0.0889
Treatment	2	892.6666667	446.3333333	11.85	0.0209

The overall F test is significant ($F = 8.28$, $p = 0.0323$) at the 95% level for the variable Worth. The Block effect is not significant at the 0.05 level but is significant at the 0.10 confidence level ($F = 4.71$, $p = 0.0889$). Generally, the usefulness of blocking should be determined before the analysis. However, since there are two dependent variables of interest, and Block is significant for one of them (Yield), blocking appears to be generally useful. For Worth, as with Yield, the effect of Treatment is significant ($F = 11.85$, $p = 0.0209$).

Issuing the following command produces the Treatment means.

```
    means Treatment;
run;
```

Figure 23.9 displays the treatment means and their standard deviations for both dependent variables.

Figure 23.9 Means of Yield and Worth

```
                          Randomized Complete Block

                             The ANOVA Procedure

Level of          ------------Yield-----------    ------------Worth-----------
Treatment     N           Mean         Std Dev          Mean         Std Dev

A             3      36.8666667      5.22908532     125.000000      13.5277493
B             3      41.2000000      5.43415127     135.666667       6.6583281
C             3      32.0000000      2.19317122     111.333333       5.0332230
```

Syntax: ANOVA Procedure

The following statements are available in PROC ANOVA.

PROC ANOVA < *options* > ;
  CLASS *variables* < / *option* > ;
  MODEL *dependents=effects* < / *options* > ;
  ABSORB *variables* ;
  BY *variables* ;
  FREQ *variable* ;
  MANOVA < *test-options* >< / *detail-options* > ;
  MEANS *effects* < / *options* > ;
  REPEATED *factor-specification* < / *options* > ;
  TEST < **H=***effects* > **E=***effect* ;

The PROC ANOVA, CLASS, and MODEL statements are required, and they must precede the first RUN statement. The CLASS statement must precede the MODEL statement. If you use the ABSORB, FREQ, or BY statement, it must precede the first RUN statement. The MANOVA, MEANS, REPEATED, and TEST statements must follow the MODEL statement, and they can be specified in any order. These four statements can also appear after the first RUN statement.

Table 23.1 summarizes the function of each statement (other than the PROC statement) in the ANOVA procedure:

Table 23.1 Statements in the ANOVA Procedure

Statement	Description
ABSORB	absorbs classification effects in a model
BY	specifies variables to define subgroups for the analysis
CLASS	declares classification variables
FREQ	specifies a frequency variable
MANOVA	performs a multivariate analysis of variance
MEANS	computes and compares means
MODEL	defines the model to be fit

Table 23.1 *continued*

Statement	Description
REPEATED	performs multivariate and univariate repeated measures analysis of variance
TEST	constructs tests that use the sums of squares for effects and the error term you specify

PROC ANOVA Statement

PROC ANOVA < *options* > **;**

The PROC ANOVA statement starts the ANOVA procedure.

You can specify the following options in the PROC ANOVA statement:

DATA=*SAS-data-set*
names the SAS data set used by the ANOVA procedure. By default, PROC ANOVA uses the most recently created SAS data set.

MANOVA
requests the multivariate mode of eliminating observations with missing values. If any of the dependent variables have missing values, the procedure eliminates that observation from the analysis. The MANOVA option is useful if you use PROC ANOVA in interactive mode and plan to perform a multivariate analysis.

MULTIPASS
requests that PROC ANOVA reread the input data set, when necessary, instead of writing the values of dependent variables to a utility file. This option decreases disk space usage at the expense of increased execution times and is useful only in rare situations where disk space is at an absolute premium.

NAMELEN=*n*
specifies the length of effect names to be *n* characters long, where *n* is a value between 20 and 200 characters. The default length is 20 characters.

NOPRINT
suppresses the normal display of results. The NOPRINT option is useful when you want to create only the output data set with the procedure. Note that this option temporarily disables the Output Delivery System

(ODS); see Chapter 20, "Using the Output Delivery System," for more information.

ORDER=DATA | FORMATTED | FREQ | INTERNAL
specifies the sorting order for the levels of the classification variables (specified in the CLASS statement). This ordering determines which parameters in the model correspond to each level in the data. Note that the ORDER= option applies to the levels for all classification variables. The exception is the default ORDER=FORMATTED for numeric variables for which you

have supplied no explicit format. In this case, the levels are ordered by their internal value. Note that this represents a change from previous releases for how classification levels are ordered. In releases previous to Version 8, numeric classification levels with no explicit format were ordered by their BEST12. formatted values, and in order to revert to the previous ordering you can specify this format explicitly for the affected classification variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and usually required the user to intervene with an explicit format or ORDER=INTERNAL to get the more natural ordering.

The following table shows how PROC ANOVA interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

OUTSTAT=*SAS-data-set*

names an output data set that contains sums of squares, degrees of freedom, F statistics, and probability levels for each effect in the model. If you use the CANONICAL option in the MANOVA statement and do not use an M= specification in the MANOVA statement, the data set also contains results of the canonical analysis. See the section "Output Data Set" on page 724 for more information.

PLOTS=NONE

controls the plots produced through ODS Graphics. When ODS Graphics is enabled, the ANOVA procedure can display a grouped box-plot of the input data with groups defined by an effect in the model. Such a plot is produced by default if you have a one-way model, with only a single classification variable, or if you use a MEANS statement. Specify the PLOTS=NONE option in order to prevent these plots from being produced when ODS Graphics is enabled.

ABSORB Statement

ABSORB *variables* **;**

Absorption is a computational technique that provides a large reduction in time and memory requirements for certain types of models. The *variables* are one or more variables in the input data set.

For a main effect variable that does not participate in interactions, you can absorb the effect by naming it in an ABSORB statement. This means that the effect can be adjusted out before the construction and solution of the rest of the model. This is particularly useful when the effect has a large number of levels.

Several variables can be specified, in which case each one is assumed to be nested in the preceding variable in the ABSORB statement.

NOTE: When you use the ABSORB statement, the data set (or each BY group, if a BY statement appears) must be sorted by the variables in the ABSORB statement. Including an absorbed variable in the CLASS list or in the MODEL statement might produce erroneous sums of squares. If the ABSORB statement is used, it must appear before the first RUN statement or it is ignored.

When you use an ABSORB statement and also use the INT option in the MODEL statement, the procedure ignores the option but produces the uncorrected total sum of squares (SS) instead of the corrected total SS.

See the section "Absorption" on page 2507 in Chapter 39, "The GLM Procedure," for more information.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC ANOVA to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the ANOVA procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in base SAS software).

Since sorting the data changes the order in which PROC ANOVA reads observations, the sorting order for the levels of the classification variables might be affected if you have also specified the ORDER=DATA option in the PROC ANOVA statement.

If the BY statement is used, it must appear before the first RUN statement or it is ignored. When you use a BY statement, the interactive features of PROC ANOVA are disabled.

When both a BY and an ABSORB statement are used, observations must be sorted first by the variables in the BY statement, and then by the variables in the ABSORB statement.

For more information about the BY statement, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *SAS Procedures Guide*.

CLASS Statement

CLASS *variables* < / *option* > **;**

The CLASS statement names the classification variables to be used in the model. Typical classification variables are TREATMENT, SEX, RACE, GROUP, and REPLICATION. The CLASS statement is required, and it must appear before the MODEL statement.

By default, classification levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which classification levels are determined. In releases prior to Version 9, classification levels were determined by using no more than the first 16 characters of the formatted values. If you wish to revert to this previous behavior you can use the TRUNCATE option on the CLASS statement. In any case, you can use formats to group values into levels.

See the discussion of the FORMAT procedure in the *SAS Procedures Guide* and the discussions for the FORMAT statement and SAS formats in *SAS Language Reference: Concepts*.

You can specify the following option in the CLASS statement after a slash(/):

TRUNCATE
specifies that classification levels should be determined by using only up to the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases previous to Version 9.

FREQ Statement

FREQ *variable* **;**

The FREQ statement names a variable that provides frequencies for each observation in the DATA= data set. Specifically, if n is the value of the FREQ variable for a given observation, then that observation is used n times.

The analysis produced by using a FREQ statement reflects the expanded number of observations. For example, means and total degrees of freedom reflect the expanded number of observations. You can produce the same analysis (without the FREQ statement) by first creating a new data set that contains the expanded number of observations. For example, if the value of the FREQ variable is 5 for the first observation, the first 5 observations in the new data set would be identical. Each observation in the old data set would be replicated n_i times in the new data set, where n_i is the value of the FREQ variable for that observation.

If the value of the FREQ variable is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

If the FREQ statement is used, it must appear before the first RUN statement or it is ignored.

MANOVA Statement

MANOVA < *test-options* > < *detail-options* > **;**

If the MODEL statement includes more than one dependent variable, you can perform multivariate analysis of variance with the MANOVA statement. The *test-options* define which effects to test, while the *detail-options* specify how to execute the tests and what results to display.

When a MANOVA statement appears before the first RUN statement, PROC ANOVA enters a multivariate mode with respect to the handling of missing values; in addition to observations with missing independent variables, observations with *any* missing dependent variables are excluded from the analysis. If you want to use this mode of handling missing values but do not need any multivariate analyses, specify the MANOVA option in the PROC ANOVA statement.

Test Options

You can specify the following options in the MANOVA statement as *test-options* in order to define which multivariate tests to perform.

H=*effects* | **INTERCEPT** | **_ALL_**

specifies effects in the preceding model to use as hypothesis matrices. For each SSCP matrix $\mathbf{H}$ associated with an effect, the H= specification computes an analysis based on the characteristic roots of $\mathbf{E}^{-1}\mathbf{H}$, where $\mathbf{E}$ is the matrix associated with the error effect. The characteristic roots and vectors are displayed, along with the Hotelling-Lawley trace, Pillai's trace, Wilks' lambda, and Roy's greatest root. By default, these statistics are tested with approximations based on the F distribution. To test them with exact (but computationally intensive) calculations, use the MSTAT=EXACT option.

Use the keyword INTERCEPT to produce tests for the intercept. To produce tests for all effects listed in the MODEL statement, use the keyword _ALL_ in place of a list of effects.

For background and further details, see the section "Multivariate Analysis of Variance" on page 2531 in Chapter 39, "The GLM Procedure."

E=*effect*

specifies the error effect. If you omit the E= specification, the ANOVA procedure uses the error SSCP (residual) matrix from the analysis.

M=*equation,...,equation* | **(***row-of-matrix,...,row-of-matrix***)**

specifies a transformation matrix for the dependent variables listed in the MODEL statement. The equations in the M= specification are of the form

$$c_1 \times \textit{dependent-variable} \quad \pm \quad c_2 \times \textit{dependent-variable}$$
$$\cdots \quad \pm \quad c_n \times \textit{dependent-variable}$$

where the c_i values are coefficients for the various *dependent-variables*. If the value of a given c_i is 1, it can be omitted; in other words $1 \times Y$ is the same as Y. Equations should involve two or more dependent variables. For sample syntax, see the section "Examples" on page 708.

Alternatively, you can input the transformation matrix directly by entering the elements of the matrix with commas separating the rows, and parentheses surrounding the matrix. When this alternate form of input is used, the number of elements in each row must equal the number of dependent variables. Although these combinations actually represent the columns of the **M** matrix, they are displayed by rows.

When you include an M= specification, the analysis requested in the MANOVA statement is carried out for the variables defined by the equations in the specification, not the original dependent variables. If you omit the M= option, the analysis is performed for the original dependent variables in the MODEL statement.

If an M= specification is included without either the MNAMES= or the PREFIX= option, the variables are labeled MVAR1, MVAR2, and so forth by default.

For further information, see the section "Multivariate Analysis of Variance" on page 2531 in Chapter 39, "The GLM Procedure."

MNAMES=*names*
provides names for the variables defined by the equations in the M= specification. Names in the list correspond to the M= equations or the rows of the **M** matrix (as it is entered).

PREFIX=*name*
is an alternative means of identifying the transformed variables defined by the M= specification. For example, if you specify PREFIX=DIFF, the transformed variables are labeled DIFF1, DIFF2, and so forth.

Detail Options

You can specify the following options in the MANOVA statement after a slash as *detail-options*:

CANONICAL
produces a canonical analysis of the **H** and **E** matrices (transformed by the **M** matrix, if specified) instead of the default display of characteristic roots and vectors.

MSTAT=FAPPROX

MSTAT=EXACT
specifies the method of evaluating the multivariate test statistics. The default is MSTAT=FAPPROX, which specifies that the multivariate tests are evaluated by using the usual approximations based on the F distribution, as discussed in the "Multivariate Tests" section in Chapter 4, "Introduction to Regression Procedures." Alternatively, you can specify MSTAT=EXACT to compute exact p-values for three of the four tests (Wilks' lambda, the Hotelling-Lawley trace, and Roy's greatest root) and an improved F-approximation for the fourth (Pillai's trace). While MSTAT=EXACT provides better control of the significance probability for the tests, especially for Roy's Greatest Root, computations for the exact

p-values can be appreciably more demanding, and are in fact infeasible for large problems (many dependent variables). Thus, although MSTAT=EXACT is more accurate for most data, it is not the default method. For more information about the results of MSTAT=EXACT, see the section "Multivariate Analysis of Variance" on page 2531 in Chapter 39, "The GLM Procedure."

ORTH

requests that the transformation matrix in the M= specification of the MANOVA statement be orthonormalized by rows before the analysis.

PRINTE

displays the error SSCP matrix **E**. If the **E** matrix is the error SSCP (residual) matrix from the analysis, the partial correlations of the dependent variables given the independent variables are also produced.

For example, the statement

```
manova / printe;
```

displays the error SSCP matrix and the partial correlation matrix computed from the error SSCP matrix.

PRINTH

displays the hypothesis SSCP matrix **H** associated with each effect specified by the H= specification.

SUMMARY

produces analysis-of-variance tables for each dependent variable. When no **M** matrix is specified, a table is produced for each original dependent variable from the MODEL statement; with an **M** matrix other than the identity, a table is produced for each transformed variable defined by the **M** matrix.

Examples

The following statements give several examples of using a MANOVA statement.

```
proc anova;
   class A B;
   model Y1-Y5=A B(A);
   manova h=A e=B(A) / printh printe;
   manova h=B(A) / printe;
   manova h=A e=B(A) m=Y1-Y2,Y2-Y3,Y3-Y4,Y4-Y5
          prefix=diff;

   manova h=A e=B(A) m=(1 -1  0  0  0,
                        0  1 -1  0  0,
                        0  0  1 -1  0,
                        0  0  0  1 -1) prefix=diff;
run;
```

The first MANOVA statement specifies A as the hypothesis effect and B(A) as the error effect. As a result of the PRINTH option, the procedure displays the hypothesis SSCP matrix associated with the A effect; and, as a result of the PRINTE option, the procedure displays the error SSCP matrix associated with the B(A) effect.

The second MANOVA statement specifies B(A) as the hypothesis effect. Since no error effect is specified, PROC ANOVA uses the error SSCP matrix from the analysis as the **E** matrix. The PRINTE option displays this **E** matrix. Since the **E** matrix is the error SSCP matrix from the analysis, the partial correlation matrix computed from this matrix is also produced.

The third MANOVA statement requests the same analysis as the first MANOVA statement, but the analysis is carried out for variables transformed to be successive differences between the original dependent variables. The PREFIX=DIFF specification labels the transformed variables as DIFF1, DIFF2, DIFF3, and DIFF4.

Finally, the fourth MANOVA statement has the identical effect as the third, but it uses an alternative form of the M= specification. Instead of specifying a set of equations, the fourth MANOVA statement specifies rows of a matrix of coefficients for the five dependent variables.

As a second example of the use of the M= specification, consider the following:

```
proc anova;
   class group;
   model dose1-dose4=group / nouni;
   manova h = group
          m = -3*dose1 -   dose2 +   dose3 + 3*dose4,
                 dose1 -   dose2 -   dose3 +   dose4,
                -dose1 + 3*dose2 - 3*dose3 +   dose4
          mnames = Linear Quadratic Cubic
          / printe;
run;
```

The M= specification gives a transformation of the dependent variables dose1 through dose4 into orthogonal polynomial components, and the MNAMES= option labels the transformed variables as LINEAR, QUADRATIC, and CUBIC, respectively. Since the PRINTE option is specified and the default residual matrix is used as an error term, the partial correlation matrix of the orthogonal polynomial components is also produced.

For further information, see the section "Multivariate Analysis of Variance" on page 2531 in Chapter 39, "The GLM Procedure."

MEANS Statement

MEANS *effects* < / *options* > ;

PROC ANOVA can compute means of the dependent variables for any effect that appears on the right-hand side in the MODEL statement.

You can use any number of MEANS statements, provided that they appear after the MODEL statement. For example, suppose A and B each have two levels. Then, if you use the following statements

```
proc anova;
   class A B;
   model Y=A B A*B;
   means A B / tukey;
   means A*B;
run;
```

means, standard deviations, and Tukey's multiple comparison tests are produced for each level of the main effects A and B, and just the means and standard deviations for each of the four combinations of levels for A*B. Since multiple comparisons options apply only to main effects, the single MEANS statement

```
means A B A*B / tukey;
```

produces the same results.

Options are provided to perform multiple comparison tests for only main effects in the model. PROC ANOVA does not perform multiple comparison tests for interaction terms in the model; for multiple comparisons of interaction terms, see the LSMEANS statement in Chapter 39, "The GLM Procedure."

Table 23.2 summarizes categories of options available in the MEANS statement.

Table 23.2 Options Available in the MEANS Statement

Task	Available options
Perform multiple comparison tests	BON
	DUNCAN
	DUNNETT
	DUNNETTL
	DUNNETTU
	GABRIEL
	GT2
	LSD
	REGWQ
	SCHEFFE
	SIDAK
	SMM
Perform multiple comparison tests	SNK
	T
	TUKEY
	WALLER

Table 23.2 *continued*

Task	Available options
Specify additional details for multiple comparison tests	ALPHA= CLDIFF CLM E= KRATIO= LINES NOSORT
Test for homogeneity of variances	HOVTEST
Compensate for heterogeneous variances	WELCH

Descriptions of these options follow. For a further discussion of these options, see the section "Multiple Comparisons" on page 2514 in Chapter 39, "The GLM Procedure."

ALPHA=*p*

specifies the level of significance for comparisons among the means. By default, ALPHA=0.05. You can specify any value greater than 0 and less than 1.

BON

performs Bonferroni *t* tests of differences between means for all main effect means in the MEANS statement. See the CLDIFF and LINES options, which follow, for a discussion of how the procedure displays results.

CLDIFF

presents results of the BON, GABRIEL, SCHEFFE, SIDAK,SMM, GT2, T, LSD, and TUKEY options as confidence intervals for all pairwise differences between means, and the results of the DUNNETT, DUNNETTU, and DUNNETTL options as confidence intervals for differences with the control. The CLDIFF option is the default for unequal cell sizes unless the DUNCAN, REGWQ, SNK, or WALLER option is specified.

CLM

presents results of the BON, GABRIEL, SCHEFFE, SIDAK,SMM, T, and LSD options as intervals for the mean of each level of the variables specified in the MEANS statement. For all options except GABRIEL, the intervals are confidence intervals for the true means. For the GABRIEL option, they are *comparison intervals* for comparing means pairwise: in this case, if the intervals corresponding to two means overlap, the difference between them is insignificant according to Gabriel's method.

DUNCAN

performs Duncan's multiple range test on all main effect means given in the MEANS statement. See the LINES option for a discussion of how the procedure displays results.

DUNNETT <(*formatted-control-values***)>**

performs Dunnett's two-tailed *t* test, testing if any treatments are significantly different from a single control for all main effects means in the MEANS statement.

To specify which level of the effect is the control, enclose the formatted value in quotes in parentheses after the keyword. If more than one effect is specified in the MEANS statement, you can use a list of control values within the parentheses. By default, the first level of the effect is used as the control. For example,

```
means a / dunnett('CONTROL');
```

where CONTROL is the formatted control value of A. As another example,

```
means a b c / dunnett('CNTLA' 'CNTLB' 'CNTLC');
```

where CNTLA, CNTLB, and CNTLC are the formatted control values for A, B, and C, respectively.

DUNNETTL <(*formatted-control-value*)>

performs Dunnett's one-tailed *t* test, testing if any treatment is significantly less than the control. Control level information is specified as described previously for the DUNNETT option.

DUNNETTU <(*formatted-control-value*)>

performs Dunnett's one-tailed *t* test, testing if any treatment is significantly greater than the control. Control level information is specified as described previously for the DUNNETT option.

E=*effect*

specifies the error mean square used in the multiple comparisons. By default, PROC ANOVA uses the residual Mean Square (MS). The effect specified with the E= option must be a term in the model; otherwise, the procedure uses the residual MS.

GABRIEL

performs Gabriel's multiple-comparison procedure on all main effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

GT2

see the SMM option.

HOVTEST

HOVTEST=BARTLETT

HOVTEST=BF

HOVTEST=LEVENE <(TYPE=ABS | SQUARE)>

HOVTEST=OBRIEN <(W=*number*)>

requests a homogeneity of variance test for the groups defined by the MEANS effect. You can optionally specify a particular test; if you do not specify a test, Levene's test (Levene 1960) with TYPE=SQUARE is computed. Note that this option is ignored unless your MODEL statement specifies a simple one-way model.

The HOVTEST=BARTLETT option specifies Bartlett's test (Bartlett 1937), a modification of the normal-theory likelihood ratio test.

The HOVTEST=BF option specifies Brown and Forsythe's variation of Levene's test (Brown and Forsythe 1974).

The HOVTEST=LEVENE option specifies Levene's test (Levene 1960), which is widely considered to be the standard homogeneity of variance test. You can use the TYPE= option in parentheses to specify whether to use the absolute residuals (TYPE=ABS) or the squared residuals (TYPE=SQUARE) in Levene's test. The default is TYPE=SQUARE.

The HOVTEST=OBRIEN option specifies O'Brien's test (O'Brien 1979), which is basically a modification of HOVTEST=LEVENE(TYPE=SQUARE). You can use the W= option in parentheses to tune the variable to match the suspected kurtosis of the underlying distribution. By default, W=0.5, as suggested by O'Brien (1979, 1981).

See the section "Homogeneity of Variance in One-Way Models" on page 2526 in Chapter 39, "The GLM Procedure," for more details on these methods. Example 39.10 in the same chapter illustrates the use of the HOVTEST and WELCH options in the MEANS statement in testing for equal group variances.

KRATIO=*value*

specifies the Type 1/Type 2 error seriousness ratio for the Waller-Duncan test. Reasonable values for KRATIO are 50, 100, and 500, which roughly correspond for the two-level case to ALPHA levels of 0.1, 0.05, and 0.01. By default, the procedure uses the default value of 100.

LINES

presents results of the BON, DUNCAN, GABRIEL, REGWQ, SCHEFFE, SIDAK,SMM, GT2, SNK, T, LSD TUKEY, and WALLER options by listing the means in descending order and indicating nonsignificant subsets by line segments beside the corresponding means. The LINES option is appropriate for equal cell sizes, for which it is the default. The LINES option is also the default if the DUNCAN, REGWQ, SNK, or WALLER option is specified, or if there are only two cells of unequal size. If the cell sizes are unequal, the harmonic mean of the cell sizes is used, which might lead to somewhat liberal tests if the cell sizes are highly disparate. The LINES option cannot be used in combination with the DUNNETT, DUNNETTL, or DUNNETTU option. In addition, the procedure has a restriction that no more than 24 overlapping groups of means can exist. If a mean belongs to more than 24 groups, the procedure issues an error message. You can either reduce the number of levels of the variable or use a multiple comparison test that allows the CLDIFF option rather than the LINES option.

LSD

see the T option.

NOSORT

prevents the means from being sorted into descending order when the CLDIFF or CLM option is specified.

REGWQ

performs the Ryan-Einot-Gabriel-Welsch multiple range test on all main effect means in the MEANS statement. See the LINES option for a discussion of how the procedure displays results.

SCHEFFE

performs Scheffé's multiple-comparison procedure on all main effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

SIDAK

performs pairwise t tests on differences between means with levels adjusted according to Sidak's inequality for all main effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

SMM

GT2

performs pairwise comparisons based on the studentized maximum modulus and Sidak's uncorrelated-t inequality, yielding Hochberg's GT2 method when sample sizes are unequal, for all main effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

SNK

performs the Student-Newman-Keuls multiple range test on all main effect means in the MEANS statement. See the LINES option for a discussion of how the procedure displays results.

T

LSD

performs pairwise t tests, equivalent to Fisher's least-significant-difference test in the case of equal cell sizes, for all main effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

TUKEY

performs Tukey's studentized range test (HSD) on all main effect means in the MEANS statement. (When the group sizes are different, this is the Tukey-Kramer test.) See the CLDIFF and LINES options for discussions of how the procedure displays results.

WALLER

performs the Waller-Duncan k-ratio t test on all main effect means in the MEANS statement. See the KRATIO= option for information about controlling details of the test, and see the LINES option for a discussion of how the procedure displays results.

WELCH

requests Welch's (1951) variance-weighted one-way ANOVA. This alternative to the usual analysis of variance for a one-way model is robust to the assumption of equal within-group variances. This option is ignored unless your MODEL statement specifies a simple one-way model.

Note that using the WELCH option merely produces one additional table consisting of Welch's ANOVA. It does not affect all of the other tests displayed by the ANOVA procedure, which still require the assumption of equal variance for exact validity.

See the section "Homogeneity of Variance in One-Way Models" on page 2526 in Chapter 39, "The GLM Procedure," for more details on Welch's ANOVA. Example 39.10 in the same

chapter illustrates the use of the HOVTEST and WELCH options in the MEANS statement in testing for equal group variances.

MODEL Statement

MODEL *dependents=effects* < / *options* > **;**

The MODEL statement names the dependent variables and independent effects. The syntax of effects is described in the section "Specification of Effects" on page 720. For any model effect involving classification variables (interactions as well as main effects), the number of levels cannot exceed 32,767. If no independent effects are specified, only an intercept term is fit. This tests the hypothesis that the mean of the dependent variable is zero. All variables in effects that you specify in the MODEL statement must appear in the CLASS statement because PROC ANOVA does not allow for continuous effects.

You can specify the following options in the MODEL statement; they must be separated from the list of independent effects by a slash.

INTERCEPT

INT

displays the hypothesis tests associated with the intercept as an effect in the model. By default, the procedure includes the intercept in the model but does not display associated tests of hypotheses. Except for producing the uncorrected total SS instead of the corrected total SS, the INT option is ignored when you use an ABSORB statement.

NOUNI

suppresses the display of univariate statistics. You typically use the NOUNI option with a multivariate or repeated measures analysis of variance when you do not need the standard univariate output. The NOUNI option in a MODEL statement does not affect the univariate output produced by the REPEATED statement.

REPEATED Statement

REPEATED *factor-specification* < / *options* > **;**

When values of the dependent variables in the MODEL statement represent repeated measurements on the same experimental unit, the REPEATED statement enables you to test hypotheses about the measurement factors (often called *within-subject factors*), as well as the interactions of within-subject factors with independent variables in the MODEL statement (often called *between-subject factors*). The REPEATED statement provides multivariate and univariate tests as well as hypothesis tests for a variety of single-degree-of-freedom contrasts. There is no limit to the number of within-subject factors that can be specified. For more details, see the section "Repeated Measures Analysis of Variance" on page 2532 in Chapter 39, "The GLM Procedure."

The REPEATED statement is typically used for handling repeated measures designs with one repeated response variable. Usually, the variables on the left-hand side of the equation in the MODEL statement represent one repeated response variable.

This does not mean that only one factor can be listed in the REPEATED statement. For example, one repeated response variable (hemoglobin count) might be measured 12 times (implying variables Y1 to Y12 on the left-hand side of the equal sign in the MODEL statement), with the associated within-subject factors treatment and time (implying two factors listed in the REPEATED statement). See the section "Examples" on page 719 for an example of how PROC ANOVA handles this case.

Designs with two or more repeated response variables can, however, be handled with the IDENTITY transformation; see Example 39.9 in Chapter 39, "The GLM Procedure," for an example of analyzing a doubly-multivariate repeated measures design.

When a REPEATED statement appears, the ANOVA procedure enters a multivariate mode of handling missing values. If any values for variables corresponding to each combination of the within-subject factors are missing, the observation is excluded from the analysis.

The simplest form of the REPEATED statement requires only a *factor-name*. With two repeated factors, you must specify the *factor-name* and number of levels (*levels*) for each factor. Optionally, you can specify the actual values for the levels (*level-values*), a *transformation* that defines single-degree-of freedom contrasts, and *options* for additional analyses and output. When more than one within-subject factor is specified, *factor-names* (and associated level and transformation information) must be separated by a comma in the REPEATED statement. These terms are described in the following section, "Syntax Details."

Syntax Details

You can specify the following terms in the REPEATED statement.

factor-specification

The *factor-specification* for the REPEATED statement can include any number of individual factor specifications, separated by commas, of the following form:

factor-name levels < (level-values) > < transformation >

where

factor-name names a factor to be associated with the dependent variables. The name should not be the same as any variable name that already exists in the data set being analyzed and should conform to the usual conventions of SAS variable names.

When specifying more than one factor, list the dependent variables in the MODEL statement so that the within-subject factors defined in the REPEATED statement are nested; that is, the first factor defined in the REPEATED statement should be the one with values that change least frequently.

levels specifies the number of levels associated with the factor being defined. When there is only one within-subject factor, the number of levels is equal to the number of dependent variables. In this case, *levels* is optional. When more than one within-subject factor is defined, however, *levels* is required, and the product of the number of levels of all the factors must equal the number of dependent variables in the MODEL statement.

(level-values) specifies values that correspond to levels of a repeated-measures factor. These values are used to label output; they are also used as spacings for constructing orthogonal polynomial contrasts if you specify a POLYNOMIAL transformation. The number of level values specified must correspond to the number of levels for that factor in the REPEATED statement. Enclose the *level-values* in parentheses.

The following *transformation* keywords define single-degree-of-freedom contrasts for factors specified in the REPEATED statement. Since the number of contrasts generated is always one less than the number of levels of the factor, you have some control over which contrast is omitted from the analysis by which transformation you select. The only exception is the IDENTITY transformation; this transformation is not composed of contrasts, and it has the same degrees of freedom as the factor has levels. By default, the procedure uses the CONTRAST transformation.

CONTRAST<(*ordinal-reference-level*)>
generates contrasts between levels of the factor and a reference level. By default, the procedure uses the last level; you can optionally specify a reference level in parentheses after the keyword CONTRAST. The reference level corresponds to the ordinal value of the level rather than the level value specified. For example, to generate contrasts between the first level of a factor and the other levels, use

```
contrast(1)
```

HELMERT generates contrasts between each level of the factor and the mean of subsequent levels.

IDENTITY generates an identity transformation corresponding to the associated factor. This transformation is *not* composed of contrasts; it has n degrees of freedom for an n-level factor, instead of $n - 1$. This can be used for doubly-multivariate repeated measures.

MEAN<(*ordinal-reference-level*)>
generates contrasts between levels of the factor and the mean of all other levels of the factor. Specifying a reference level eliminates the contrast between that level and the mean. Without a reference level, the contrast involving the last level is omitted. See the CONTRAST transformation for an example.

POLYNOMIAL generates orthogonal polynomial contrasts. Level values, if provided, are used as spacings in the construction of the polynomials; otherwise, equal spacing is assumed.

PROFILE generates contrasts between adjacent levels of the factor.

For examples of the transformation matrices generated by these contrast transformations, see the section "Repeated Measures Analysis of Variance" on page 2532 in Chapter 39, "The GLM Procedure."

You can specify the following options in the REPEATED statement after a slash:

CANONICAL
performs a canonical analysis of the **H** and **E** matrices corresponding to the transformed variables specified in the REPEATED statement.

MSTAT=FAPPROX

MSTAT=EXACT
specifies the method of evaluating the multivariate test statistics. The default is MSTAT=FAPPROX, which specifies that the multivariate tests are evaluated by using the usual approximations based on the F distribution, as discussed in the "Multivariate Tests" section in Chapter 4, "Introduction to Regression Procedures." Alternatively, you can specify MSTAT=EXACT to compute exact p-values for three of the four tests (Wilks' lambda, the Hotelling-Lawley trace, and Roy's greatest root) and an improved F-approximation for the fourth (Pillai's trace). While MSTAT=EXACT provides better control of the significance probability for the tests, especially for Roy's Greatest Root, computations for the exact p-values can be appreciably more demanding, and are in fact infeasible for large problems (many dependent variables). Thus, although MSTAT=EXACT is more accurate for most data, it is not the default method. For more information about the results of MSTAT=EXACT, see the section "Multivariate Analysis of Variance" on page 2531 in Chapter 39, "The GLM Procedure."

NOM
displays only the results of the univariate analyses.

NOU
displays only the results of the multivariate analyses.

PRINTE
displays the **E** matrix for each combination of within-subject factors, as well as partial correlation matrices for both the original dependent variables and the variables defined by the transformations specified in the REPEATED statement. In addition, the PRINTE option provides sphericity tests for each set of transformed variables. If the requested transformations are not orthogonal, the PRINTE option also provides a sphericity test for a set of orthogonal contrasts.

PRINTH
displays the **H** (SSCP) matrix associated with each multivariate test.

PRINTM
displays the transformation matrices that define the contrasts in the analysis. PROC ANOVA always displays the **M** matrix so that the transformed variables are defined by the rows, not the columns, of the displayed **M** matrix. In other words, PROC ANOVA actually displays $\mathbf{M}'$.

PRINTRV
produces the characteristic roots and vectors for each multivariate test.

SUMMARY
produces analysis-of-variance tables for each contrast defined by the within-subjects factors. Along with tests for the effects of the independent variables specified in the MODEL statement, a term labeled MEAN tests the hypothesis that the overall mean of the contrast is zero.

Examples

When specifying more than one factor, list the dependent variables in the MODEL statement so that the within-subject factors defined in the REPEATED statement are nested; that is, the first factor defined in the REPEATED statement should be the one with values that change least frequently. For example, assume that three treatments are administered at each of four times, for a total of twelve dependent variables on each experimental unit. If the variables are listed in the MODEL statement as Y1 through Y12, then the following REPEATED statement

```
repeated trt 3, time 4;
```

implies the following structure:

	Dependent Variables											
	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12
Value of trt	1	1	1	1	2	2	2	2	3	3	3	3
Value of time	1	2	3	4	1	2	3	4	1	2	3	4

The REPEATED statement always produces a table like the preceding one.

For more information about repeated measures analysis and about using the REPEATED statement, see the section "Repeated Measures Analysis of Variance" on page 2532 in Chapter 39, "The GLM Procedure."

TEST Statement

TEST < **H=** *effects* > **E=** *effect* **;**

Although an F value is computed for all SS in the analysis by using the residual MS as an error term, you can request additional F tests that use other effects as error terms. You need a TEST statement when a nonstandard error structure (as in a split plot) exists.

CAUTION: The ANOVA procedure does not check any of the assumptions underlying the F statistic. When you specify a TEST statement, you assume sole responsibility for the validity of the F statistic produced. To help validate a test, you might want to use the GLM procedure with the RANDOM statement and inspect the expected mean squares. In the GLM procedure, you can also use the TEST option in the RANDOM statement.

You can use as many TEST statements as you want, provided that they appear after the MODEL statement.

You can specify the following terms in the TEST statement.

H=*effects* specifies which effects in the preceding model are to be used as hypothesis (numerator) effects.

E=*effect* specifies one, and only one, effect to use as the error (denominator) term. The **E=** specification is required.

The following example uses two TEST statements and is appropriate for analyzing a split-plot design.

```
proc anova;
   class a b c;
   model y=a|b(a)|c;
   test h=a e=b(a);
   test h=c a*c e=b*c(a);
run;
```

Details: ANOVA Procedure

Specification of Effects

In SAS analysis-of-variance procedures, the variables that identify levels of the classifications are called *classification variables*, and they are declared in the CLASS statement. Classification variables are also called *categorical*, *qualitative*, *discrete*, or *nominal variables*. The values of a classification variable are called *levels*. Classification variables can be either numeric or character. This is in contrast to the *response* (or *dependent*) *variables*, which are continuous. Response variables must be numeric.

The analysis-of-variance model specifies *effects*, which are combinations of classification variables used to explain the variability of the dependent variables in the following manner:

- Main effects are specified by writing the variables by themselves in the CLASS statement: `A B C`. Main effects used as independent variables test the hypothesis that the mean of the dependent variable is the same for each level of the factor in question, ignoring the other independent variables in the model.
- Crossed effects (interactions) are specified by joining the CLASS variables with asterisks in the MODEL statement: `A*B A*C A*B*C`. Interaction terms in a model test the hypothesis that the effect of a factor does not depend on the levels of the other factors in the interaction.

- Nested effects are specified by following a main effect or crossed effect with a CLASS variable or list of CLASS variables enclosed in parentheses in the MODEL statement. The main effect or crossed effect is nested within the effects listed in parentheses: B(A) C*D(A B). Nested effects test hypotheses similar to interactions, but the levels of the nested variables are not the same for every combination within which they are nested.

The general form of an effect can be illustrated by using the CLASS variables A, B, C, D, E, and F:

A ∗ B ∗ C(D E F)

The crossed list should come first, followed by the nested list in parentheses. Note that no asterisks appear within the nested list or immediately before the left parenthesis.

Main Effects Models

For a three-factor main effects model with A, B, and C as the factors and Y as the dependent variable, the necessary statements are

```
proc anova;
   class A B C;
   model Y=A B C;
run;
```

Models with Crossed Factors

To specify interactions in a factorial model, join effects with asterisks as described previously. For example, these statements specify a complete factorial model, which includes all the interactions:

```
proc anova;
   class A B C;
   model Y=A B C A*B A*C B*C A*B*C;
run;
```

Bar Notation

You can shorten the specifications of a full factorial model by using bar notation. For example, the preceding statements can also be written

```
proc anova;
   class A B C;
   model Y=A|B|C;
run;
```

When the bar (|) is used, the expression on the right side of the equal sign is expanded from left to right by using the equivalents of rules 2–4 given in Searle (1971, p. 390). The variables on the right- and left-hand sides of the bar become effects, and the cross of them becomes an effect. Multiple bars are permitted. For instance, A | B | C is evaluated as follows:

$$\begin{aligned} \text{A | B | C} &\rightarrow \{\text{ A | B }\}\text{ | C} \\ &\rightarrow \{\text{ A B A*B }\}\text{ | C} \\ &\rightarrow \text{A B A*B C A*C B*C A*B*C} \end{aligned}$$

You can also specify the maximum number of variables involved in any effect that results from bar evaluation by specifying that maximum number, preceded by an @ sign, at the end of the bar effect. For example, the specification A | B | C@2 results in only those effects that contain two or fewer variables; in this case, A B A*B C A*C and B*C.

The following table gives more examples of using the bar and at operators.

A | C(B)	is equivalent to	A C(B) A*C(B)
A(B) | C(B)	is equivalent to	A(B) C(B) A*C(B)
A(B) | B(D E)	is equivalent to	A(B) B(D E)
A | B(A) | C	is equivalent to	A B(A) C A*C B*C(A)
A | B(A) | C@2	is equivalent to	A B(A) C A*C
A | B | C | D@2	is equivalent to	A B A*B C A*C B*C D A*D B*D C*D

Consult the section "Specification of Effects" on page 2488 in Chapter 39, "The GLM Procedure," for further details on bar notation.

Nested Models

Write the effect that is nested within another effect first, followed by the other effect in parentheses. For example, if A and B are main effects and C is nested within A and B (that is, the levels of C that are observed are not the same for each combination of A and B), the statements for PROC ANOVA are

```
proc anova;
   class A B C;
   model y=A B C(A B);
run;
```

The identity of a level is viewed within the context of the level of the containing effects. For example, if City is nested within State, then the identity of City is viewed within the context of State.

The distinguishing feature of a nested specification is that nested effects never appear as main effects. Another way of viewing nested effects is that they are effects that pool the main effect with the interaction of the nesting variable.

See the "Automatic Pooling" section, which follows.

Models Involving Nested, Crossed, and Main Effects

Asterisks and parentheses can be combined in the MODEL statement for models involving nested and crossed effects:

```
proc anova;
   class A B C;
   model Y=A B(A) C(A) B*C(A);
run;
```

Automatic Pooling

In line with the general philosophy of the GLM procedure, there is no difference between the statements

```
model Y=A B(A);
```

and

```
model Y=A A*B;
```

The effect B becomes a nested effect by virtue of the fact that it does not occur as a main effect. If B is not written as a main effect in addition to participating in A*B, then the sum of squares that is associated with B is pooled into A*B.

This feature allows the automatic pooling of sums of squares. If an effect is omitted from the model, it is automatically pooled with all the higher-level effects containing the CLASS variables in the omitted effect (or within-error). This feature is most useful in split-plot designs.

Using PROC ANOVA Interactively

PROC ANOVA can be used interactively. After you specify a model in a MODEL statement and run PROC ANOVA with a RUN statement, a variety of statements (such as MEANS, MANOVA, TEST, and REPEATED) can be executed without PROC ANOVA recalculating the model sum of squares.

the section "Syntax: ANOVA Procedure" on page 701 describes which statements can be used interactively. You can execute these interactive statements individually or in groups by following the single statement or group of statements with a RUN statement. Note that the MODEL statement cannot be repeated; the ANOVA procedure allows only one MODEL statement.

If you use PROC ANOVA interactively, you can end the procedure with a DATA step, another PROC step, an ENDSAS statement, or a QUIT statement. The syntax of the QUIT statement is

```
quit;
```

When you use PROC ANOVA interactively, additional RUN statements do not end the procedure but tell PROC ANOVA to execute additional statements.

When a WHERE statement is used with PROC ANOVA, it should appear before the first RUN statement. The WHERE statement enables you to select only certain observations for analysis

without using a subsetting DATA step. For example, the statement `where group ne 5` omits observations with GROUP=5 from the analysis. See *SAS Language Reference: Dictionary* for details about this statement.

When a BY statement is used with PROC ANOVA, interactive processing is not possible; that is, once the first RUN statement is encountered, processing proceeds for each BY group in the data set, and no further statements are accepted by the procedure.

Interactivity is also disabled when there are different patterns of missing values among the dependent variables. For details, see the section "Missing Values," which follows.

Missing Values

For an analysis involving one dependent variable, PROC ANOVA uses an observation if values are nonmissing for that dependent variable and for all the variables used in independent effects.

For an analysis involving multiple dependent variables without the MANOVA or REPEATED statement, or without the MANOVA option in the PROC ANOVA statement, a missing value in one dependent variable does not eliminate the observation from the analysis of other nonmissing dependent variables. For an analysis with the MANOVA or REPEATED statement, or with the MANOVA option in the PROC ANOVA statement, the ANOVA procedure requires values for all dependent variables to be nonmissing for an observation before the observation can be used in the analysis.

During processing, PROC ANOVA groups the dependent variables by their pattern of missing values across observations so that sums and cross products can be collected in the most efficient manner.

If your data have different patterns of missing values among the dependent variables, interactivity is disabled. This could occur when some of the variables in your data set have missing values and either of the following conditions obtain:

- You do not use the MANOVA option in the PROC ANOVA statement.
- You do not use a MANOVA or REPEATED statement before the first RUN statement.

Output Data Set

The OUTSTAT= option in the PROC ANOVA statement produces an output data set that contains the following:

- the BY variables, if any
- _TYPE_, a new character variable. This variable has the value 'ANOVA' for observations corresponding to sums of squares; it has the value 'CANCORR', 'STRUCTUR', or 'SCORE' if a canonical analysis is performed through the MANOVA statement and no M= matrix is specified.

- _SOURCE_, a new character variable. For each observation in the data set, _SOURCE_ contains the name of the model effect from which the corresponding statistics are generated.
- _NAME_, a new character variable. The variable _NAME_ contains the name of one of the dependent variables in the model or, in the case of canonical statistics, the name of one of the canonical variables (CAN1, CAN2, and so on).
- four new numeric variables, SS, DF, F, and PROB, containing sums of squares, degrees of freedom, F values, and probabilities, respectively, for each model or contrast sum of squares generated in the analysis. For observations resulting from canonical analyses, these variables have missing values.
- if there is more than one dependent variable, then variables with the same names as the dependent variables represent
  - for _TYPE_='ANOVA', the crossproducts of the hypothesis matrices
  - for _TYPE_='CANCORR', canonical correlations for each variable
  - for _TYPE_='STRUCTUR', coefficients of the total structure matrix
  - for _TYPE_='SCORE', raw canonical score coefficients

The output data set can be used to perform special hypothesis tests (for example, with the IML procedure in SAS/IML software), to reformat output, to produce canonical variates (through the SCORE procedure), or to rotate structure matrices (through the FACTOR procedure).

Computational Method

Let $\mathbf{X}$ represent the $n \times p$ design matrix. The columns of $\mathbf{X}$ contain only 0s and 1s. Let $\mathbf{Y}$ represent the $n \times 1$ vector of dependent variables.

In the GLM procedure, $\mathbf{X}'\mathbf{X}$, $\mathbf{X}'\mathbf{Y}$, and $\mathbf{Y}'\mathbf{Y}$ are formed in main storage. However, in the ANOVA procedure, only the diagonals of $\mathbf{X}'\mathbf{X}$ are computed, along with $\mathbf{X}'\mathbf{Y}$ and $\mathbf{Y}'\mathbf{Y}$. Thus, PROC ANOVA saves a considerable amount of storage as well as time. The memory requirements for PROC ANOVA are asymptotically linear functions of n^2 and nr, where n is the number of dependent variables and r the number of independent parameters.

The elements of $\mathbf{X}'\mathbf{Y}$ are cell totals, and the diagonal elements of $\mathbf{X}'\mathbf{X}$ are cell frequencies. Since PROC ANOVA automatically pools omitted effects into the next higher-level effect containing the names of the omitted effect (or within-error), a slight modification to the rules given by Searle (1971, p. 389) is used.

1. PROC ANOVA computes the sum of squares for each effect as if it is a main effect. In other words, for each effect, PROC ANOVA squares each cell total and divides by its cell frequency. The procedure then adds these quantities together and subtracts the correction factor for the mean (total squared over N).
2. For each effect involving two CLASS variable names, PROC ANOVA subtracts the SS for any main effect with a name that is contained in the two-factor effect.

3. For each effect involving three CLASS variable names, PROC ANOVA subtracts the SS for all main effects and two-factor effects with names that are contained in the three-factor effect. If effects involving four or more CLASS variable names are present, the procedure continues this process.

Displayed Output

PROC ANOVA first displays a table that includes the following:

- the name of each variable in the CLASS statement
- the number of different values or Levels of the CLASS variables
- the Values of the CLASS variables
- the Number of observations in the data set and the number of observations excluded from the analysis because of missing values, if any

PROC ANOVA then displays an analysis-of-variance table for each dependent variable in the MODEL statement. This table breaks down the Total Sum of Squares for the dependent variable into the portion attributed to the Model and the portion attributed to Error. It also breaks down the Mean Square term, which is the Sum of Squares divided by the degrees of freedom (DF). The analysis-of-variance table also lists the following:

- the Mean Square for Error (MSE), which is an estimate of σ^2, the variance of the true errors
- the F Value, which is the ratio produced by dividing the Mean Square for the Model by the Mean Square for Error. It tests how well the model as a whole (adjusted for the mean) accounts for the dependent variable's behavior. This F test is a test of the null hypothesis that all parameters except the intercept are zero.
- the significance probability associated with the F statistic, labeled "Pr > F"
- R-Square, R^2, which measures how much variation in the dependent variable can be accounted for by the model. The R^2 statistic, which can range from 0 to 1, is the ratio of the sum of squares for the model divided by the sum of squares for the corrected total. In general, the larger the R^2 value, the better the model fits the data.
- C.V., the coefficient of variation, which is often used to describe the amount of variation in the population. The C.V. is 100 times the standard deviation of the dependent variable divided by the Mean. The coefficient of variation is often a preferred measure because it is unitless.
- Root MSE, which estimates the standard deviation of the dependent variable. Root MSE is computed as the square root of Mean Square for Error, the mean square of the error term.
- the Mean of the dependent variable

For each effect (or source of variation) in the model, PROC ANOVA then displays the following:

- DF, degrees of freedom
- Anova SS, the sum of squares, and the associated Mean Square
- the F Value for testing the hypothesis that the group means for that effect are equal
- Pr > F, the significance probability value associated with the F Value

When you specify a TEST statement, PROC ANOVA displays the results of the requested tests. When you specify a MANOVA statement and the model includes more than one dependent variable, PROC ANOVA produces these additional statistics:

- the characteristic roots and vectors of $\mathbf{E}^{-1}\mathbf{H}$ for each $\mathbf{H}$ matrix
- the Hotelling-Lawley trace
- Pillai's trace
- Wilks' lambda
- Roy's greatest root

See Example 39.6 in Chapter 39, "The GLM Procedure," for an example of the MANOVA results. These MANOVA tests are discussed in Chapter 4, "Introduction to Regression Procedures."

ODS Table Names

PROC ANOVA assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 23.3. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 23.3 ODS Tables Produced by PROC ANOVA

ODS Table Name	Description	Statement / Option
AltErrTests	Anova tests with error other than MSE	TEST /E=
Bartlett	Bartlett's homogeneity of variance test	MEANS / HOVTEST=BARTLETT
CLDiffs	Multiple comparisons of pairwise differences	MEANS / CLDIFF or DUNNETT or (Unequal cells and not LINES)
CLDiffsInfo	Information for multiple comparisons of pairwise differences	MEANS / CLDIFF or DUNNETT or (Unequal cells and not LINES)

Table 23.3 *continued*

ODS Table Name	Description	Statement / Option
CLMeans	Multiple comparisons of means with confidence/comparison interval	MEANS / CLM with (BON or GABRIEL or SCHEFFE or SIDAK or SMM or T or LSD)
CLMeansInfo	Information for multiple comparisons of means with confidence/comparison interval	MEANS / CLM
CanAnalysis	Canonical analysis	(MANOVA or REPEATED) / CANONICAL
CanCoef	Canonical coefficients	(MANOVA or REPEATED) / CANONICAL
CanStructure	Canonical structure	(MANOVA or REPEATED) / CANONICAL
CharStruct	Characteristic roots and vectors	(MANOVA / not CANONICAL) or (REPEATED / PRINTRV)
ClassLevels	Classification variable levels	CLASS statement
DependentInfo	Simultaneously analyzed dependent variables	default when there are multiple dependent variables with different patterns of missing values
Epsilons	Greenhouse-Geisser and Huynh-Feldt epsilons	REPEATED statement
ErrorSSCP	Error SSCP matrix	(MANOVA or REPEATED) / PRINTE
FitStatistics	R-Square, C.V., Root MSE, and dependent mean	default
HOVFTest	Homogeneity of variance ANOVA	MEANS / HOVTEST
HypothesisSSCP	Hypothesis SSCP matrix	(MANOVA or REPEATED) / PRINTE
MANOVATransform	Multivariate transformation matrix	MANOVA / M=
MCLines	Multiple comparisons LINES output	MEANS / LINES or ((DUNCAN or WALLER or SNK or REGWQ) and not(CLDIFF or CLM)) or (Equal cells and not CLDIFF)
MCLinesInfo	Information for multiple comparison LINES output	MEANS / LINES or ((DUNCAN or WALLER or SNK or REGWQ) and not (CLDIFF or CLM)) or (Equal cells and not CLDIFF)
MCLinesRange	Ranges for multiple range MC tests	MEANS / LINES or ((DUNCAN or WALLER or SNK or REGWQ) and not (CLDIFF or CLM)) or (Equal cells and not CLDIFF)
Means	Group means	MEANS statement
ModelANOVA	ANOVA for model terms	default
MultStat	Multivariate tests	MANOVA statement
NObs	Number of observations	default
OverallANOVA	Over-all ANOVA	default

Table 23.3 *continued*

ODS Table Name	Description	Statement / Option
PartialCorr	Partial correlation matrix	(MANOVA or REPEATED) / PRINTE
RepeatedTransform	Repeated transformation matrix	REPEATED (CONTRAST or HELMERT or MEAN or POLYNOMIAL or PROFILE)
RepeatedLevelInfo	Correspondence between dependents and repeated measures levels	REPEATED statement
Sphericity	Sphericity tests	REPEATED / PRINTE
Tests	Summary ANOVA for specified MANOVA H= effects	MANOVA / H= SUMMARY
Welch	Welch's ANOVA	MEANS / WELCH

ODS Graphics

This section describes the use of ODS for creating statistical graphs with the ANOVA procedure. To request these graphs you must specify the ODS GRAPHICS statement with an appropriate model, as discussed in the following. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

When the ODS Graphics are in effect, if you specify a one-way analysis of variance model, with just one independent classification variable, or if you use a MEANS statement, then the ANOVA procedure will produce a grouped box plot of the response values versus the classification levels. For an example of the box plot, see the section "One-Way Layout with Means Comparisons" on page 693.

ODS Graph Names

PROC ANOVA produces a single graph, the name of which you can use for referencing it in ODS. The name is listed in Table 23.4.

To request this graph you must specify the ODS GRAPHICS statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Table 23.4 ODS Graphic Produced by PROC ANOVA

ODS Graph Name	Plot Description
BoxPlot	Box plot of observed response values by classification levels

Examples: ANOVA Procedure

Example 23.1: Randomized Complete Block With Factorial Treatment Structure

This example uses statements for the analysis of a randomized block with two treatment factors occurring in a factorial structure. The data, from Neter, Wasserman, and Kutner (1990, p. 941), are from an experiment examining the effects of codeine and acupuncture on post-operative dental pain in male subjects. Both treatment factors have two levels. The codeine levels are a codeine capsule or a sugar capsule. The acupuncture levels are two inactive acupuncture points or two active acupuncture points. There are four distinct treatment combinations due to the factorial treatment structure. The 32 subjects are assigned to eight blocks of four subjects each based on an assessment of pain tolerance.

The data for the analysis are balanced, so PROC ANOVA is used. The data are as follows:

```
title1 'Randomized Complete Block With Two Factors';
data PainRelief;
   input PainLevel Codeine Acupuncture Relief @@;
   datalines;
1 1 1 0.0  1 2 1 0.5  1 1 2 0.6  1 2 2 1.2
2 1 1 0.3  2 2 1 0.6  2 1 2 0.7  2 2 2 1.3
3 1 1 0.4  3 2 1 0.8  3 1 2 0.8  3 2 2 1.6
4 1 1 0.4  4 2 1 0.7  4 1 2 0.9  4 2 2 1.5
5 1 1 0.6  5 2 1 1.0  5 1 2 1.5  5 2 2 1.9
6 1 1 0.9  6 2 1 1.4  6 1 2 1.6  6 2 2 2.3
7 1 1 1.0  7 2 1 1.8  7 1 2 1.7  7 2 2 2.1
8 1 1 1.2  8 2 1 1.7  8 1 2 1.6  8 2 2 2.4
;
```

The variable PainLevel is the blocking variable, and Codeine and Acupuncture represent the levels of the two treatment factors. The variable Relief is the pain relief score (the higher the score, the more relief the patient has).

The following statements invokes PROC ANOVA. The blocking variable and treatment factors appear in the CLASS statement. The bar between the treatment factors Codeine and Acupuncture adds their main effects as well as their interaction Codeine*Acupuncture to the model.

```
proc anova data=PainRelief;
   class PainLevel Codeine Acupuncture;
   model Relief = PainLevel Codeine|Acupuncture;
run;
```

The results from the analysis are shown in Output 23.1.1, Output 23.1.2, and Output 23.1.3.

Output 23.1.1 Class Level Information

Randomized Complete Block With Two Factors

The ANOVA Procedure

Class Level Information

Class	Levels	Values
PainLevel	8	1 2 3 4 5 6 7 8
Codeine	2	1 2
Acupuncture	2	1 2

Number of Observations Read	32
Number of Observations Used	32

Output 23.1.2 ANOVA Table

Randomized Complete Block With Two Factors

The ANOVA Procedure

Dependent Variable: Relief

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	10	11.33500000	1.13350000	78.37	<.0001
Error	21	0.30375000	0.01446429		
Corrected Total	31	11.63875000			

R-Square	Coeff Var	Root MSE	Relief Mean
0.973902	10.40152	0.120268	1.156250

The Class Level Information and ANOVA table are shown in Output 23.1.1 and Output 23.1.2. The classification level information summarizes the structure of the design. It is good to check these consistently in search of errors in the DATA step. The overall F test is significant, indicating that the model accounts for a significant amount of variation in the dependent variable.

Output 23.1.3 Tests of Effects

Source	DF	Anova SS	Mean Square	F Value	Pr > F
PainLevel	7	5.59875000	0.79982143	55.30	<.0001
Codeine	1	2.31125000	2.31125000	159.79	<.0001
Acupuncture	1	3.38000000	3.38000000	233.68	<.0001
Codeine*Acupuncture	1	0.04500000	0.04500000	3.11	0.0923

Output 23.1.3 shows tests of the effects. The blocking effect is significant; hence, it is useful. The interaction between codeine and acupuncture is significant at the 90% level but not at the 95% level. The significance level of this test should be determined before the analysis. The main effects of both treatment factors are highly significant.

Example 23.2: Alternative Multiple Comparison Procedures

The following is a continuation of the first example in the section "One-Way Layout with Means Comparisons" on page 693. You are studying the effect of bacteria on the nitrogen content of red clover plants, and the analysis of variance shows a highly significant effect. The following statements create the data set and compute the analysis of variance as well as Tukey's multiple comparisons test for pairwise differences between bacteria strains; the results are shown in Figure 23.1, Figure 23.2, and Figure 23.3

```
title1 'Nitrogen Content of Red Clover Plants';
data Clover;
   input Strain $ Nitrogen @@;
   datalines;
3DOK1  19.4 3DOK1  32.6 3DOK1  27.0 3DOK1  32.1 3DOK1  33.0
3DOK5  17.7 3DOK5  24.8 3DOK5  27.9 3DOK5  25.2 3DOK5  24.3
3DOK4  17.0 3DOK4  19.4 3DOK4   9.1 3DOK4  11.9 3DOK4  15.8
3DOK7  20.7 3DOK7  21.0 3DOK7  20.5 3DOK7  18.8 3DOK7  18.6
3DOK13 14.3 3DOK13 14.4 3DOK13 11.8 3DOK13 11.6 3DOK13 14.2
COMPOS 17.3 COMPOS 19.4 COMPOS 19.1 COMPOS 16.9 COMPOS 20.8
;

proc anova data=Clover;
   class Strain;
   model Nitrogen = Strain;
   means Strain / tukey;
run;
```

The interactivity of PROC ANOVA enables you to submit further MEANS statements without rerunning the entire analysis. For example, the following command requests means of the Strain levels with Duncan's multiple range test and the Waller-Duncan k-ratio t test.

```
   means Strain / duncan waller;
run;
```

Results of the Waller-Duncan k-ratio t test are shown in Output 23.2.1.

Output 23.2.1 Waller-Duncan K-ratio t Test

Nitrogen Content of Red Clover Plants

The ANOVA Procedure

Waller-Duncan K-ratio t Test for Nitrogen

Kratio	100
Error Degrees of Freedom	24
Error Mean Square	11.78867
F Value	14.37
Critical Value of t	1.91873
Minimum Significant Difference	4.1665

Means with the same letter are not significantly different.

Waller Grouping	Mean	N	Strain
A	28.820	5	3DOK1
B	23.980	5	3DOK5
B			
C B	19.920	5	3DOK7
C			
C D	18.700	5	COMPOS
D			
E D	14.640	5	3DOK4
E			
E	13.260	5	3DOK13

The Waller-Duncan k-ratio t test is a multiple range test. Unlike Tukey's test, this test does not operate on the principle of controlling Type I error. Instead, it compares the Type I and Type II error rates based on Bayesian principles (Steel and Torrie 1980).

The Waller Grouping column in Output 23.2.1 shows which means are significantly different. From this test, you can conclude the following:

- The mean nitrogen content for strain 3DOK1 is higher than the means for all other strains.
- The mean nitrogen content for strain 3DOK5 is higher than the means for COMPOS, 3DOK4, and 3DOK13.
- The mean nitrogen content for strain 3DOK7 is higher than the means for 3DOK4 and 3DOK13.
- The mean nitrogen content for strain COMPOS is higher than the mean for 3DOK13.
- Differences between all other means are not significant based on this sample size.

Output 23.2.2 shows the results of Duncan's multiple range test. Duncan's test is a result-guided test that compares the treatment means while controlling the comparison-wise error rate. You should use this test for planned comparisons only (Steel and Torrie 1980). The results and conclusions for this example are the same as for the Waller-Duncan k-ratio t test. This is not always the case.

Output 23.2.2 Duncan's Multiple Range Test

```
                    Alpha                          0.05
                    Error Degrees of Freedom         24
                    Error Mean Square           11.78867

    Number of Means          2          3          4          5          6
    Critical Range       4.482      4.707      4.852      4.954      5.031

         Means with the same letter are not significantly different.

           Duncan Grouping           Mean      N    Strain

                      A             28.820      5    3DOK1

                      B             23.980      5    3DOK5
                      B
                C     B             19.920      5    3DOK7
                C
                C     D             18.700      5    COMPOS
                      D
                E     D             14.640      5    3DOK4
                E
                E                   13.260      5    3DOK13
```

Tukey and Least Significant Difference (LSD) tests are requested with the following MEANS statement. The CLDIFF option requests confidence intervals for both tests.

```
   means Strain/ lsd tukey cldiff ;
run;
```

The LSD tests for this example are shown in Output 23.2.3, and they give the same results as the previous two multiple comparison tests. Again, this is not always the case.

Output 23.2.3 T Tests (LSD)

Nitrogen Content of Red Clover Plants

The ANOVA Procedure

t Tests (LSD) for Nitrogen

Alpha	0.05
Error Degrees of Freedom	24
Error Mean Square	11.78867
Critical Value of t	2.06390
Least Significant Difference	4.4818

Comparisons significant at the 0.05 level are indicated by ***.

Strain Comparison	Difference Between Means	95% Confidence Limits		
3DOK1 - 3DOK5	4.840	0.358	9.322	***
3DOK1 - 3DOK7	8.900	4.418	13.382	***
3DOK1 - COMPOS	10.120	5.638	14.602	***
3DOK1 - 3DOK4	14.180	9.698	18.662	***
3DOK1 - 3DOK13	15.560	11.078	20.042	***
3DOK5 - 3DOK1	-4.840	-9.322	-0.358	***
3DOK5 - 3DOK7	4.060	-0.422	8.542	
3DOK5 - COMPOS	5.280	0.798	9.762	***
3DOK5 - 3DOK4	9.340	4.858	13.822	***
3DOK5 - 3DOK13	10.720	6.238	15.202	***
3DOK7 - 3DOK1	-8.900	-13.382	-4.418	***
3DOK7 - 3DOK5	-4.060	-8.542	0.422	
3DOK7 - COMPOS	1.220	-3.262	5.702	
3DOK7 - 3DOK4	5.280	0.798	9.762	***
3DOK7 - 3DOK13	6.660	2.178	11.142	***
COMPOS - 3DOK1	-10.120	-14.602	-5.638	***
COMPOS - 3DOK5	-5.280	-9.762	-0.798	***
COMPOS - 3DOK7	-1.220	-5.702	3.262	
COMPOS - 3DOK4	4.060	-0.422	8.542	
COMPOS - 3DOK13	5.440	0.958	9.922	***
3DOK4 - 3DOK1	-14.180	-18.662	-9.698	***
3DOK4 - 3DOK5	-9.340	-13.822	-4.858	***
3DOK4 - 3DOK7	-5.280	-9.762	-0.798	***
3DOK4 - COMPOS	-4.060	-8.542	0.422	
3DOK4 - 3DOK13	1.380	-3.102	5.862	
3DOK13 - 3DOK1	-15.560	-20.042	-11.078	***
3DOK13 - 3DOK5	-10.720	-15.202	-6.238	***
3DOK13 - 3DOK7	-6.660	-11.142	-2.178	***
3DOK13 - COMPOS	-5.440	-9.922	-0.958	***
3DOK13 - 3DOK4	-1.380	-5.862	3.102	

If you only perform the LSD tests when the overall model F test is significant, then this is called Fisher's protected LSD test. Note that the LSD tests should be used for planned comparisons.

The TUKEY tests shown in Output 23.2.4 find fewer significant differences than the other three tests. This is not unexpected, as the TUKEY test controls the Type I experimentwise error rate. For a complete discussion of multiple comparison methods, see the section "Multiple Comparisons" on page 2514 in Chapter 39, "The GLM Procedure."

Output 23.2.4 Tukey's Studentized Range Test

Alpha	0.05
Error Degrees of Freedom	24
Error Mean Square	11.78867
Critical Value of Studentized Range	4.37265
Minimum Significant Difference	6.7142

Comparisons significant at the 0.05 level are indicated by ***.

Strain Comparison	Difference Between Means	Simultaneous 95% Confidence Limits		
3DOK1 - 3DOK5	4.840	-1.874	11.554	
3DOK1 - 3DOK7	8.900	2.186	15.614	***
3DOK1 - COMPOS	10.120	3.406	16.834	***
3DOK1 - 3DOK4	14.180	7.466	20.894	***
3DOK1 - 3DOK13	15.560	8.846	22.274	***
3DOK5 - 3DOK1	-4.840	-11.554	1.874	
3DOK5 - 3DOK7	4.060	-2.654	10.774	
3DOK5 - COMPOS	5.280	-1.434	11.994	
3DOK5 - 3DOK4	9.340	2.626	16.054	***
3DOK5 - 3DOK13	10.720	4.006	17.434	***
3DOK7 - 3DOK1	-8.900	-15.614	-2.186	***
3DOK7 - 3DOK5	-4.060	-10.774	2.654	
3DOK7 - COMPOS	1.220	-5.494	7.934	
3DOK7 - 3DOK4	5.280	-1.434	11.994	
3DOK7 - 3DOK13	6.660	-0.054	13.374	
COMPOS - 3DOK1	-10.120	-16.834	-3.406	***
COMPOS - 3DOK5	-5.280	-11.994	1.434	
COMPOS - 3DOK7	-1.220	-7.934	5.494	
COMPOS - 3DOK4	4.060	-2.654	10.774	
COMPOS - 3DOK13	5.440	-1.274	12.154	
3DOK4 - 3DOK1	-14.180	-20.894	-7.466	***
3DOK4 - 3DOK5	-9.340	-16.054	-2.626	***
3DOK4 - 3DOK7	-5.280	-11.994	1.434	
3DOK4 - COMPOS	-4.060	-10.774	2.654	
3DOK4 - 3DOK13	1.380	-5.334	8.094	
3DOK13 - 3DOK1	-15.560	-22.274	-8.846	***
3DOK13 - 3DOK5	-10.720	-17.434	-4.006	***
3DOK13 - 3DOK7	-6.660	-13.374	0.054	
3DOK13 - COMPOS	-5.440	-12.154	1.274	
3DOK13 - 3DOK4	-1.380	-8.094	5.334	

Example 23.3: Split Plot

In some experiments, treatments can be applied only to groups of experimental observations rather than separately to each observation. When there are two nested groupings of the observations on the basis of treatment application, this is known as a *split plot design*. For example, in integrated circuit fabrication it is of interest to see how different manufacturing methods affect the characteristics of individual chips. However, much of the manufacturing process is applied to a relatively large wafer of material, from which many chips are made. Additionally, a chip's position within a wafer might also affect chip performance. These two groupings of chips—by wafer and by position-within-wafer—might form the *whole plots* and the *subplots*, respectively, of a split plot design for integrated circuits.

The following statements produce an analysis for a split-plot design. The CLASS statement includes the variables Block, A, and B, where B defines subplots within BLOCK*A whole plots. The MODEL statement includes the independent effects Block, A, Block*A, B, and A*B. The TEST statement asks for an F test of the A effect that uses the Block*A effect as the error term. The following statements produce Output 23.3.1 and Output 23.3.2:

```
title1 'Split Plot Design';
data Split;
   input Block 1 A 2 B 3 Response;
   datalines;
142 40.0
141 39.5
112 37.9
111 35.4
121 36.7
122 38.2
132 36.4
131 34.8
221 42.7
222 41.6
212 40.3
211 41.6
241 44.5
242 47.6
231 43.6
232 42.8
;

proc anova data=Split;
   class Block A B;
   model Response = Block A Block*A B A*B;
   test h=A e=Block*A;
run;
```

Output 23.3.1 Class Level Information and ANOVA Table

Split Plot Design

The ANOVA Procedure

Class Level Information

Class	Levels	Values
Block	2	1 2
A	4	1 2 3 4
B	2	1 2

Number of Observations Read	16
Number of Observations Used	16

Split Plot Design

The ANOVA Procedure

Dependent Variable: Response

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	11	182.0200000	16.5472727	7.85	0.0306
Error	4	8.4300000	2.1075000		
Corrected Total	15	190.4500000			

R-Square	Coeff Var	Root MSE	Response Mean
0.955736	3.609007	1.451723	40.22500

First, notice that the overall F test for the model is significant.

Output 23.3.2 Tests of Effects

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Block	1	131.1025000	131.1025000	62.21	0.0014
A	3	40.1900000	13.3966667	6.36	0.0530
Block*A	3	6.9275000	2.3091667	1.10	0.4476
B	1	2.2500000	2.2500000	1.07	0.3599
A*B	3	1.5500000	0.5166667	0.25	0.8612

Tests of Hypotheses Using the Anova MS for Block*A as an Error Term

Source	DF	Anova SS	Mean Square	F Value	Pr > F
A	3	40.19000000	13.39666667	5.80	0.0914

The effect of Block is significant. The effect of A is not significant: look at the F test produced by the TEST statement, not at the F test produced by default. Neither the B nor A*B effects are significant. The test for Block*A is irrelevant, as this is simply the main-plot error.

Example 23.4: Latin Square Split Plot

The data for this example is taken from Smith (1951). A Latin square design is used to evaluate six different sugar beet varieties arranged in a six-row (Rep) by six-column (Column) square. The data are collected over two harvests. The variable Harvest then becomes a split plot on the original Latin square design for whole plots. The following statements produce Output 23.4.1, Output 23.4.2, and Output 23.4.3:

```
title1 'Sugar Beet Varieties';
title3 'Latin Square Split-Plot Design';
data Beets;
   do Harvest=1 to 2;
      do Rep=1 to 6;
         do Column=1 to 6;
            input Variety Y @;
            output;
            end;
         end;
      end;
   datalines;
3 19.1 6 18.3 5 19.6 1 18.6 2 18.2 4 18.5
6 18.1 2 19.5 4 17.6 3 18.7 1 18.7 5 19.9
1 18.1 5 20.2 6 18.5 4 20.1 3 18.6 2 19.2
2 19.1 3 18.8 1 18.7 5 20.2 4 18.6 6 18.5
4 17.5 1 18.1 2 18.7 6 18.2 5 20.4 3 18.5
5 17.7 4 17.8 3 17.4 2 17.0 6 17.6 1 17.6
3 16.2 6 17.0 5 18.1 1 16.6 2 17.7 4 16.3
6 16.0 2 15.3 4 16.0 3 17.1 1 16.5 5 17.6
1 16.5 5 18.1 6 16.7 4 16.2 3 16.7 2 17.3
2 17.5 3 16.0 1 16.4 5 18.0 4 16.6 6 16.1
4 15.7 1 16.1 2 16.7 6 16.3 5 17.8 3 16.2
5 18.3 4 16.6 3 16.4 2 17.6 6 17.1 1 16.5
;

proc anova data=Beets;
   class Column Rep Variety Harvest;
   model Y=Rep Column Variety Rep*Column*Variety
           Harvest Harvest*Rep
           Harvest*Variety;
   test h=Rep Column Variety e=Rep*Column*Variety;
   test h=Harvest            e=Harvest*Rep;
run;
```

Output 23.4.1 Class Level Information

Sugar Beet Varieties

Latin Square Split-Plot Design

The ANOVA Procedure

Class Level Information

Class	Levels	Values
Column	6	1 2 3 4 5 6
Rep	6	1 2 3 4 5 6
Variety	6	1 2 3 4 5 6
Harvest	2	1 2

Number of Observations Read 72
Number of Observations Used 72

Output 23.4.2 ANOVA Table

Sugar Beet Varieties

Latin Square Split-Plot Design

The ANOVA Procedure

Dependent Variable: Y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	46	98.9147222	2.1503200	7.22	<.0001
Error	25	7.4484722	0.2979389		
Corrected Total	71	106.3631944			

R-Square	Coeff Var	Root MSE	Y Mean
0.929971	3.085524	0.545838	17.69028

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Rep	5	4.32069444	0.86413889	2.90	0.0337
Column	5	1.57402778	0.31480556	1.06	0.4075
Variety	5	20.61902778	4.12380556	13.84	<.0001
Column*Rep*Variety	20	3.25444444	0.16272222	0.55	0.9144
Harvest	1	60.68347222	60.68347222	203.68	<.0001
Rep*Harvest	5	7.71736111	1.54347222	5.18	0.0021
Variety*Harvest	5	0.74569444	0.14913889	0.50	0.7729

First, note from Output 23.4.2 that the overall model is significant.

Output 23.4.3 Tests of Effects

Tests of Hypotheses Using the Anova MS for Column*Rep*Variety as an Error Term

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Rep	5	4.32069444	0.86413889	5.31	0.0029
Column	5	1.57402778	0.31480556	1.93	0.1333
Variety	5	20.61902778	4.12380556	25.34	<.0001

Tests of Hypotheses Using the Anova MS for Rep*Harvest as an Error Term

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Harvest	1	60.68347222	60.68347222	39.32	0.0015

Output 23.4.3 shows that the effects for Rep and Harvest are significant, while the Column effect is not. The average Ys for the six different Varietys are significantly different. For these four tests, look at the output produced by the two TEST statements, not at the usual ANOVA procedure output. The Variety*Harvest interaction is not significant. All other effects in the default output should either be tested by using the results from the TEST statements or are irrelevant as they are only error terms for portions of the model.

Example 23.5: Strip-Split Plot

In this example, four different fertilizer treatments are laid out in vertical strips, which are then split into subplots with different levels of calcium. Soil type is stripped across the split-plot experiment, and the entire experiment is then replicated three times. The dependent variable is the yield of winter barley. The data come from the notes of G. Cox and A. Rotti.

The input data are the 96 values of Y, arranged so that the calcium value (Calcium) changes most rapidly, then the fertilizer value (Fertilizer), then the Soil value, and, finally, the Rep value. Values are shown for Calcium (0 and 1); Fertilizer (0, 1, 2, 3); Soil (1, 2, 3); and Rep (1, 2, 3, 4). The following example produces Output 23.5.1, Output 23.5.2, Output 23.5.3, and Output 23.5.4.

```
title1 'Strip-split Plot';
data Barley;
   do Rep=1 to 4;
      do Soil=1 to 3; /* 1=d 2=h 3=p */
         do Fertilizer=0 to 3;
            do Calcium=0,1;
               input Yield @;
               output;
            end;
         end;
      end;
   end;
   datalines;
4.91 4.63 4.76 5.04 5.38 6.21 5.60 5.08
4.94 3.98 4.64 5.26 5.28 5.01 5.45 5.62
5.20 4.45 5.05 5.03 5.01 4.63 5.80 5.90
6.00 5.39 4.95 5.39 6.18 5.94 6.58 6.25
5.86 5.41 5.54 5.41 5.28 6.67 6.65 5.94
5.45 5.12 4.73 4.62 5.06 5.75 6.39 5.62
4.96 5.63 5.47 5.31 6.18 6.31 5.95 6.14
5.71 5.37 6.21 5.83 6.28 6.55 6.39 5.57
4.60 4.90 4.88 4.73 5.89 6.20 5.68 5.72
5.79 5.33 5.13 5.18 5.86 5.98 5.55 4.32
5.61 5.15 4.82 5.06 5.67 5.54 5.19 4.46
5.13 4.90 4.88 5.18 5.45 5.80 5.12 4.42
;

proc anova data=Barley;
   class Rep Soil Calcium Fertilizer;
   model Yield =
           Rep
           Fertilizer Fertilizer*Rep
           Calcium Calcium*Fertilizer Calcium*Rep(Fertilizer)
           Soil Soil*Rep
           Soil*Fertilizer Soil*Rep*Fertilizer
           Soil*Calcium Soil*Fertilizer*Calcium
           Soil*Calcium*Rep(Fertilizer);
   test h=Fertilizer                e=Fertilizer*Rep;
   test h=Calcium calcium*fertilizer e=Calcium*Rep(Fertilizer);
   test h=Soil                      e=Soil*Rep;
   test h=Soil*Fertilizer           e=Soil*Rep*Fertilizer;
   test h=Soil*Calcium
          Soil*Fertilizer*Calcium   e=Soil*Calcium*Rep(Fertilizer);
   means Fertilizer Calcium Soil Calcium*Fertilizer;
run;
```

Output 23.5.1 Class Level Information

Strip-split Plot

The ANOVA Procedure

Class Level Information

Class	Levels	Values
Rep	4	1 2 3 4
Soil	3	1 2 3
Calcium	2	0 1
Fertilizer	4	0 1 2 3

Number of Observations Read	96
Number of Observations Used	96

Output 23.5.2 ANOVA Table

Strip-split Plot

The ANOVA Procedure

Dependent Variable: Yield

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	95	31.89149583	0.33569996	.	.
Error	0	0.00000000	.		
Corrected Total	95	31.89149583			

R-Square	Coeff Var	Root MSE	Yield Mean
1.000000	.	.	5.427292

Output 23.5.2 *continued*

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Rep	3	6.27974583	2.09324861	.	.
Fertilizer	3	7.22127083	2.40709028	.	.
Rep*Fertilizer	9	6.08211250	0.67579028	.	.
Calcium	1	0.27735000	0.27735000	.	.
Calcium*Fertilizer	3	1.96395833	0.65465278	.	.
Rep*Calcium(Fertili)	12	1.76705833	0.14725486	.	.
Soil	2	1.92658958	0.96329479	.	.
Rep*Soil	6	1.66761042	0.27793507	.	.
Soil*Fertilizer	6	0.68828542	0.11471424	.	.
Rep*Soil*Fertilizer	18	1.58698125	0.08816563	.	.
Soil*Calcium	2	0.04493125	0.02246562	.	.
Soil*Calcium*Fertili	6	0.18936042	0.03156007	.	.
Rep*Soil*Calc(Ferti)	24	2.19624167	0.09151007	.	.

Notice in Output 23.5.2 that the default tests against the residual error rate are all unavailable. This is because the Soil*Calcium*Rep(Fertilizer) term in the model takes up all the degrees of freedom, leaving none for estimating the residual error rate. This is appropriate in this case since the TEST statements give the specific error terms appropriate for testing each effect. Output 23.5.3 displays the output produced by the various TEST statements. The only significant effect is the Calcium*Fertilizer interaction.

Output 23.5.3 Tests of Effects

Tests of Hypotheses Using the Anova MS for Rep*Fertilizer as an Error Term

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Fertilizer	3	7.22127083	2.40709028	3.56	0.0604

Tests of Hypotheses Using the Anova MS for Rep*Calcium(Fertili) as an Error Term

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Calcium	1	0.27735000	0.27735000	1.88	0.1950
Calcium*Fertilizer	3	1.96395833	0.65465278	4.45	0.0255

Tests of Hypotheses Using the Anova MS for Rep*Soil as an Error Term

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Soil	2	1.92658958	0.96329479	3.47	0.0999

Tests of Hypotheses Using the Anova MS for Rep*Soil*Fertilizer as an Error Term

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Soil*Fertilizer	6	0.68828542	0.11471424	1.30	0.3063

Output 23.5.3 *continued*

Tests of Hypotheses Using the Anova MS for Rep*Soil*Calc(Ferti) as an Error Term

Source	DF	Anova SS	Mean Square	F Value	Pr > F
Soil*Calcium	2	0.04493125	0.02246562	0.25	0.7843
Soil*Calcium*Fertili	6	0.18936042	0.03156007	0.34	0.9059

Output 23.5.4 Results of MEANS statement

Level of Fertilizer	N	Yield Mean	Yield Std Dev
0	24	5.18416667	0.48266395
1	24	5.12916667	0.38337082
2	24	5.75458333	0.53293265
3	24	5.64125000	0.63926801

Level of Calcium	N	Yield Mean	Yield Std Dev
0	48	5.48104167	0.54186141
1	48	5.37354167	0.61565219

Level of Soil	N	Yield Mean	Yield Std Dev
1	32	5.54312500	0.55806369
2	32	5.51093750	0.62176315
3	32	5.22781250	0.51825224

Level of Calcium	Level of Fertilizer	N	Yield Mean	Yield Std Dev
0	0	12	5.34666667	0.45029956
0	1	12	5.08833333	0.44986530
0	2	12	5.62666667	0.44707806
0	3	12	5.86250000	0.52886027
1	0	12	5.02166667	0.47615569
1	1	12	5.17000000	0.31826233
1	2	12	5.88250000	0.59856077
1	3	12	5.42000000	0.68409197

Output 23.5.4 shows the results of the MEANS statement, displaying for various effects and combinations of effects, as requested. You can examine the Calcium*Fertilizer means to understand the interaction better.

In this example, you could reduce memory requirements by omitting the Soil*Calcium*Rep(Fertilizer) effect from the model in the MODEL statement. This effect then becomes the ERROR effect, and you can omit the last TEST statement in the statements shown

earlier. The test for the Soil*Calcium effect is then given in the Analysis of Variance table in the top portion of output. However, for all other tests, you should look at the results from the TEST statement. In large models, this method might lead to significant reductions in memory requirements.

References

Bartlett, M. S. (1937), "Properties of Sufficiency and Statistical Tests," *Proceedings of the Royal Society of London, Series A*, 160, 268–282.

Brown, M. B. and Forsythe, A. B. (1974), "Robust Tests for Equality of Variances," *Journal of the American Statistical Association*, 69, 364–367.

Erdman, L. W. (1946), "Studies to Determine If Antibiosis Occurs among Rhizobia," *Journal of the American Society of Agronomy*, 38, 251–258.

Fisher, R. A. (1942), *The Design of Experiments*, Third Edition, Edinburgh: Oliver & Boyd.

Freund, R. J., Littell, R. C., and Spector, P. C. (1986), *SAS System for Linear Models*, 1986 Edition, Cary, NC: SAS Institute Inc.

Graybill, F. A. (1961), *An Introduction to Linear Statistical Models, Volume I*, New York: McGraw-Hill.

Henderson, C. R. (1953), "Estimation of Variance and Covariance Components," *Biometrics*, 9, 226–252.

Levene, H. (1960), "Robust Tests for the Equality of Variance," in I. Olkin, ed., *Contributions to Probability and Statistics*, 278–292, Palo Alto, CA: Stanford University Press.

Neter, J., Wasserman, W., and Kutner, M. H. (1990), *Applied Linear Statistical Models*, Third Edition, Homewood, IL: Irwin.

O'Brien, R. G. (1979), "A General ANOVA Method for Robust Tests of Additive Models for Variances," *Journal of the American Statistical Association*, 74, 877–880.

O'Brien, R. G. (1981), "A Simple Test for Variance Effects in Experimental Designs," *Psychological Bulletin*, 89, 570–574.

Remington, R. D. and Schork, M. A. (1970), *Statistics with Applications to the Biological and Health Sciences*, Englewood Cliffs, NJ: Prentice-Hall.

Scheffé, H. (1959), *The Analysis of Variance*, New York: John Wiley & Sons.

Schlotzhauer, S. D. and Littell, R. C. (1987), *SAS System for Elementary Statistical Analysis*, Cary, NC: SAS Institute Inc.

Searle, S. R. (1971), *Linear Models*, New York: John Wiley & Sons.

Smith, W. G. (1951), "Dissertation Notes on Canadian Sugar Factories, Ltd." Alberta, Canada: Taber.

Snedecor, G. W. and Cochran, W. G. (1967), *Statistical Methods*, Sixth Edition, Ames: Iowa State University Press.

Steel, R. G. D. and Torrie, J. H. (1980), *Principles and Procedures of Statistics*, Second Edition, New York: McGraw-Hill.

Welch, B. L. (1951), "On the Comparison of Several Mean Values: An Alternative Approach," *Biometrika*, 38, 330–336.

Chapter 24

The BOXPLOT Procedure

Contents

Overview: BOXPLOT Procedure **750**
Getting Started: BOXPLOT Procedure **751**
Creating Box Plots from Raw Data 751
Creating Box Plots from Summary Data 754
Saving Summary Data with Outliers 756
Syntax: BOXPLOT Procedure **759**
PROC BOXPLOT Statement 759
BY Statement 760
ID Statement 760
INSET Statement 761
INSETGROUP Statement 764
PLOT Statement 767
Details: BOXPLOT Procedure **790**
Summary Statistics Represented by Box Plots 790
Output Data Sets 791
Input Data Sets 793
Styles of Box Plots 796
Percentile Definitions 797
Missing Values 798
Continuous Group Variables 798
Positioning Insets 800
Displaying Blocks of Data 805
Clipping Extreme Values 807
ODS Graphics (Experimental) 811
Examples: BOXPLOT Procedure **812**
Example 24.1: Displaying Summary Statistics in a Box Plot 812
Example 24.2: Using Box Plots to Compare Groups 813
Example 24.3: Creating Various Styles of Box-and-Whiskers Plots 817
Example 24.4: Creating Notched Box-and-Whiskers Plots 821
Example 24.5: Creating Box-and-Whiskers Plots with Varying Widths . . . 822
Example 24.6: Creating Box-and-Whiskers Plots Using ODS Graphics (Experimental) 824
References **826**

Overview: BOXPLOT Procedure

The BOXPLOT procedure creates side-by-side box-and-whiskers plots of measurements organized in groups. A box-and-whiskers plot displays the mean, quartiles, and minimum and maximum observations for a group. Throughout this chapter, this type of plot, which can contain one or more box-and-whiskers plots, is referred to as a *box plot*.

The PLOT statement of the BOXPLOT procedure produces a box plot. You can specify more than one PLOT statement to produce multiple box plots. You can use options in the PLOT statement to do the following:

- control the style of the box-and-whiskers plots
- specify one of several methods for calculating quantile statistics (percentiles)
- add block legends and symbol markers to reveal stratification in data
- display vertical and horizontal reference lines
- control axis values and labels
- overlay the box plot with plots of additional variables
- control the layout and appearance of the plot

The INSET and INSETGROUP statements produce boxes or tables (referred to as *insets*) of summary statistics or other data on a box plot. An INSET statement produces an inset of statistics pertaining to the entire box plot. An INSETGROUP statement produces an inset containing statistics calculated separately for each group. An INSET or INSETGROUP statement by itself does not produce a display; it must be used with a PLOT statement.

You can use options in an INSET or INSETGROUP statement to control insets in these ways:

- specify the position of the inset
- specify a header for the inset
- specify graphical enhancements, such as background colors, text colors, text height, text font, and drop shadows

The BOXPLOT procedure can produce two kinds of graphical output:

- traditional graphics
- ODS Statistical Graphics output, supported on an experimental basis for SAS 9.2

PROC BOXPLOT produces traditional graphics box plots by default. These graphs are saved in graphics catalogs. Their appearance is controlled by the SAS/GRAPH GOPTIONS, AXIS, and SYMBOL statements (as described in *SAS/GRAPH: Reference*) and numerous specialized PLOT statement options.

ODS Statistical Graphics (or ODS Graphics for short) is an extension to the Output Delivery System (ODS) that is invoked when you use the ODS GRAPHICS statement prior to your procedure statements. An ODS graph is produced in ODS output (not a graphics catalog), and the details of its appearance and layout are controlled by ODS styles and templates rather than by SAS/GRAPH statements and procedure options. See Chapter 21, "Statistical Graphics Using ODS," for a thorough discussion of ODS Graphics.

Prior to SAS 9.2, the box plots produced by PROC BOXPLOT were extremely basic by default. Producing attractive graphical output required the careful selection of colors, fonts, and other elements, which were specified via SAS/GRAPH statements and PLOT statement options. Beginning with SAS 9.2, the default appearance of box plots is governed by the prevailing ODS style, which automatically produces attractive, consistent output. You can also specify the NOGSTYLE system option to prevent the ODS style from affecting the appearance of traditional graphs.

See the section "Getting Started: BOXPLOT Procedure" on page 751 for examples producing box plots via the traditional graphics system and ODS Graphics.

Getting Started: BOXPLOT Procedure

This section introduces the BOXPLOT procedure with simple examples demonstrating commonly used options. Complete syntax for the BOXPLOT procedure is presented in the section "Syntax: BOXPLOT Procedure" on page 759, and advanced examples are presented in the section "Examples: BOXPLOT Procedure" on page 812.

Creating Box Plots from Raw Data

A petroleum company uses a turbine to heat water into steam that is pumped into the ground to make oil less viscous and easier to extract. This process occurs 20 times daily, and the amount of power (in kilowatts) used to heat the water to the desired temperature is recorded. The following statements create a SAS data set called Turbine that contains the power output measurements for 10 nonconsecutive days:

```
data Turbine;
   informat Day date7.;
   format Day date5.;
   label KWatts='Average Power Output';
   input Day @;
   do i=1 to 10;
      input KWatts @;
      output;
      end;
   drop i;
   datalines;
05JUL94 3196 3507 4050 3215 3583 3617 3789 3180 3505 3454
05JUL94 3417 3199 3613 3384 3475 3316 3556 3607 3364 3721
06JUL94 3390 3562 3413 3193 3635 3179 3348 3199 3413 3562
06JUL94 3428 3320 3745 3426 3849 3256 3841 3575 3752 3347
07JUL94 3478 3465 3445 3383 3684 3304 3398 3578 3348 3369
07JUL94 3670 3614 3307 3595 3448 3304 3385 3499 3781 3711
08JUL94 3448 3045 3446 3620 3466 3533 3590 3070 3499 3457
08JUL94 3411 3350 3417 3629 3400 3381 3309 3608 3438 3567
11JUL94 3568 2968 3514 3465 3175 3358 3460 3851 3845 2983
11JUL94 3410 3274 3590 3527 3509 3284 3457 3729 3916 3633
12JUL94 3153 3408 3741 3203 3047 3580 3571 3579 3602 3335
12JUL94 3494 3662 3586 3628 3881 3443 3456 3593 3827 3573
13JUL94 3594 3711 3369 3341 3611 3496 3554 3400 3295 3002
13JUL94 3495 3368 3726 3738 3250 3632 3415 3591 3787 3478
14JUL94 3482 3546 3196 3379 3559 3235 3549 3445 3413 3859
14JUL94 3330 3465 3994 3362 3309 3781 3211 3550 3637 3626
15JUL94 3152 3269 3431 3438 3575 3476 3115 3146 3731 3171
15JUL94 3206 3140 3562 3592 3722 3421 3471 3621 3361 3370
18JUL94 3421 3381 4040 3467 3475 3285 3619 3325 3317 3472
18JUL94 3296 3501 3366 3492 3367 3619 3550 3263 3355 3510
;
run;
```

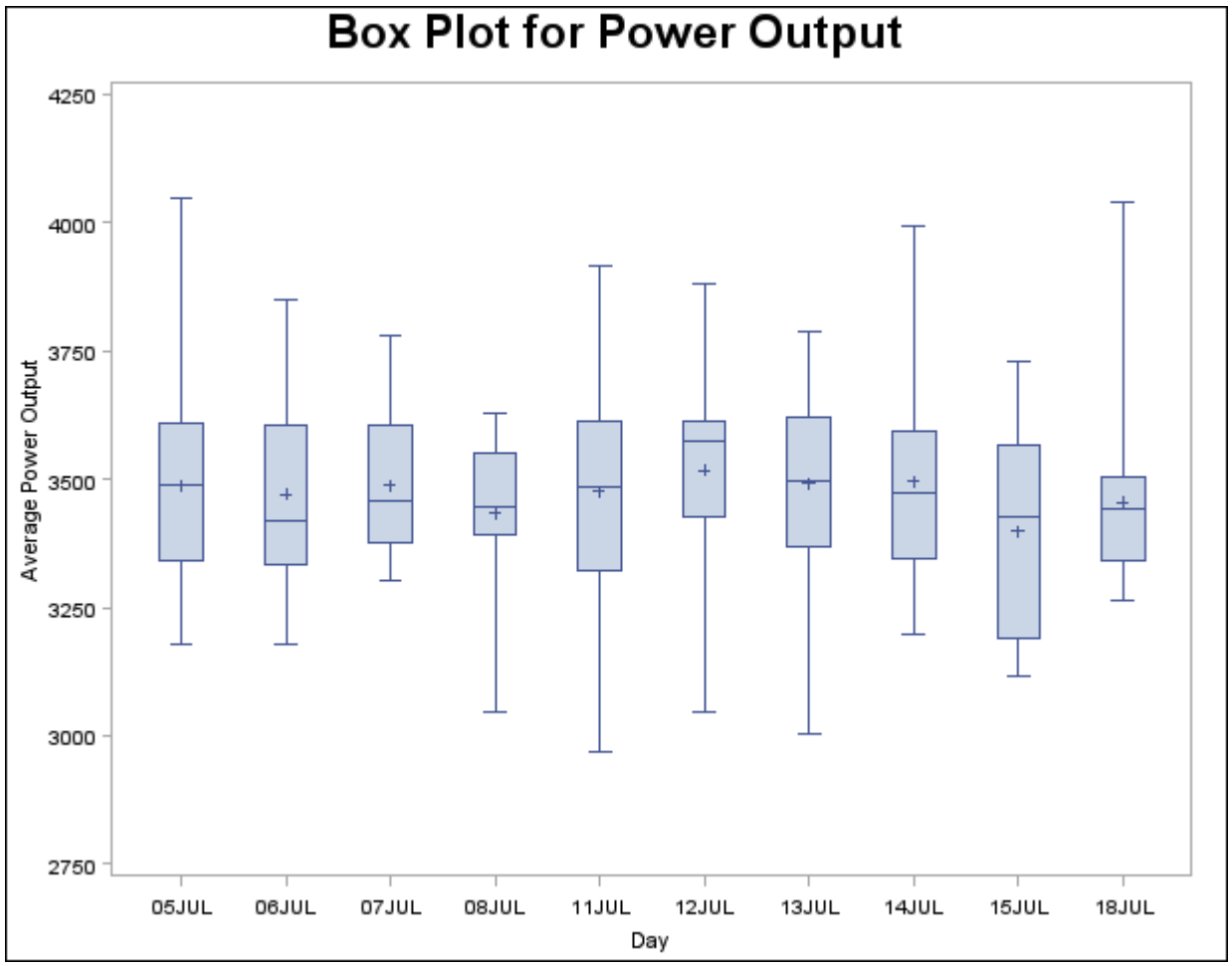
In the data set Turbine, each observation contains the date and the power output for a single heating. The first 20 observations contain the outputs for the first day, the second 20 observations contain the outputs for the second day, and so on. Because the variable Day classifies the observations into groups, it is referred to as the *group variable*. The variable KWatts contains the output measurements and is referred to as the *analysis variable*.

The following statements create a box plot showing the distribution of power output for each day:

```
title 'Box Plot for Power Output';
proc boxplot data=Turbine;
   plot KWatts*Day;
run;
```

The input data set Turbine is specified with the DATA= option in the PROC BOXPLOT statement. The PLOT statement requests a box-and-whiskers plot for each group of data. After the keyword PLOT, you specify the analysis variable (in this case, KWatts), followed by an asterisk and the group variable (Day). The box plot is shown in Figure 24.1.

Figure 24.1 Box Plot for Power Output Data



The box plot displayed in Figure 24.1 represents summary statistics for the analysis variable KWatts. Each of the 10 box-and-whiskers plots describes the variable KWatts for a particular day. The plot elements and the statistics they represent are as follows:

- The length of the box represents the interquartile range (the distance between the 25th and 75th percentiles).
- The symbol in the box interior represents the group mean.
- The horizontal line in the box interior represents the group median.
- The vertical lines (called *whiskers*) issuing from the box extend to the group minimum and maximum values.

Creating Box Plots from Summary Data

The previous example illustrates how you can create box plots from raw data. However, in some applications the data are provided as summary statistics. This example illustrates how you can use the BOXPLOT procedure with data of this type.

The following statements create the data set Oilsum, which provides the data from the preceding example in summarized form:

```
data Oilsum;
   input Day KWattsL KWatts1 KWattsX KWattsM
             KWatts3 KWattsH KWattsS KWattsN;
   informat Day date7. ;
   format Day date5. ;
   label Day    ='Date of Measurement'
         KWattsL='Minimum Power Output'
         KWatts1='25th Percentile'
         KWattsX='Average Power Output'
         KWattsM='Median Power Output'
         KWatts3='75th Percentile'
         KWattsH='Maximum Power Output'
         KWattsS='Standard Deviation of Power Output'
         KWattsN='Group Sample Size';
   datalines;
05JUL94 3180 3340.0 3487.40 3490.0 3610.0 4050 220.3 20
06JUL94 3179 3333.5 3471.65 3419.5 3605.0 3849 210.4 20
07JUL94 3304 3376.0 3488.30 3456.5 3604.5 3781 147.0 20
08JUL94 3045 3390.5 3434.20 3447.0 3550.0 3629 157.6 20
11JUL94 2968 3321.0 3475.80 3487.0 3611.5 3916 258.9 20
12JUL94 3047 3425.5 3518.10 3576.0 3615.0 3881 211.6 20
13JUL94 3002 3368.5 3492.65 3495.5 3621.5 3787 193.8 20
14JUL94 3196 3346.0 3496.40 3473.5 3592.5 3994 212.0 20
15JUL94 3115 3188.5 3398.50 3426.0 3568.5 3731 199.2 20
18JUL94 3263 3340.0 3456.05 3444.0 3505.5 4040 173.5 20
;
run;
```

Oilsum contains exactly one observation for each group. Note that, as in the previous example, the groups are indexed by the variable Day. A listing of Oilsum is shown in Figure 24.2.

Figure 24.2 The Summary Data Set Oilsum

Box Plot for Power Output

Day	KWatts L	KWatts1	KWattsX	KWatts M	KWatts3	KWatts H	KWatts S	KWatts N
05JUL	3180	3340.0	3487.40	3490.0	3610.0	4050	220.3	20
06JUL	3179	3333.5	3471.65	3419.5	3605.0	3849	210.4	20
07JUL	3304	3376.0	3488.30	3456.5	3604.5	3781	147.0	20
08JUL	3045	3390.5	3434.20	3447.0	3550.0	3629	157.6	20
11JUL	2968	3321.0	3475.80	3487.0	3611.5	3916	258.9	20
12JUL	3047	3425.5	3518.10	3576.0	3615.0	3881	211.6	20
13JUL	3002	3368.5	3492.65	3495.5	3621.5	3787	193.8	20
14JUL	3196	3346.0	3496.40	3473.5	3592.5	3994	212.0	20
15JUL	3115	3188.5	3398.50	3426.0	3568.5	3731	199.2	20
18JUL	3263	3340.0	3456.05	3444.0	3505.5	4040	173.5	20

There are eight summary variables in Oilsum:

- KWattsL contains the group minima (low values).
- KWatts1 contains the 25th percentile (first quartile) for each group.
- KWattsX contains the group means.
- KWattsM contains the group medians.
- KWatts3 contains the 75th percentile (third quartile) for each group.
- KWattsH contains the group maxima (high values).
- KWattsS contains the group standard deviations.
- KWattsN contains the group sizes.

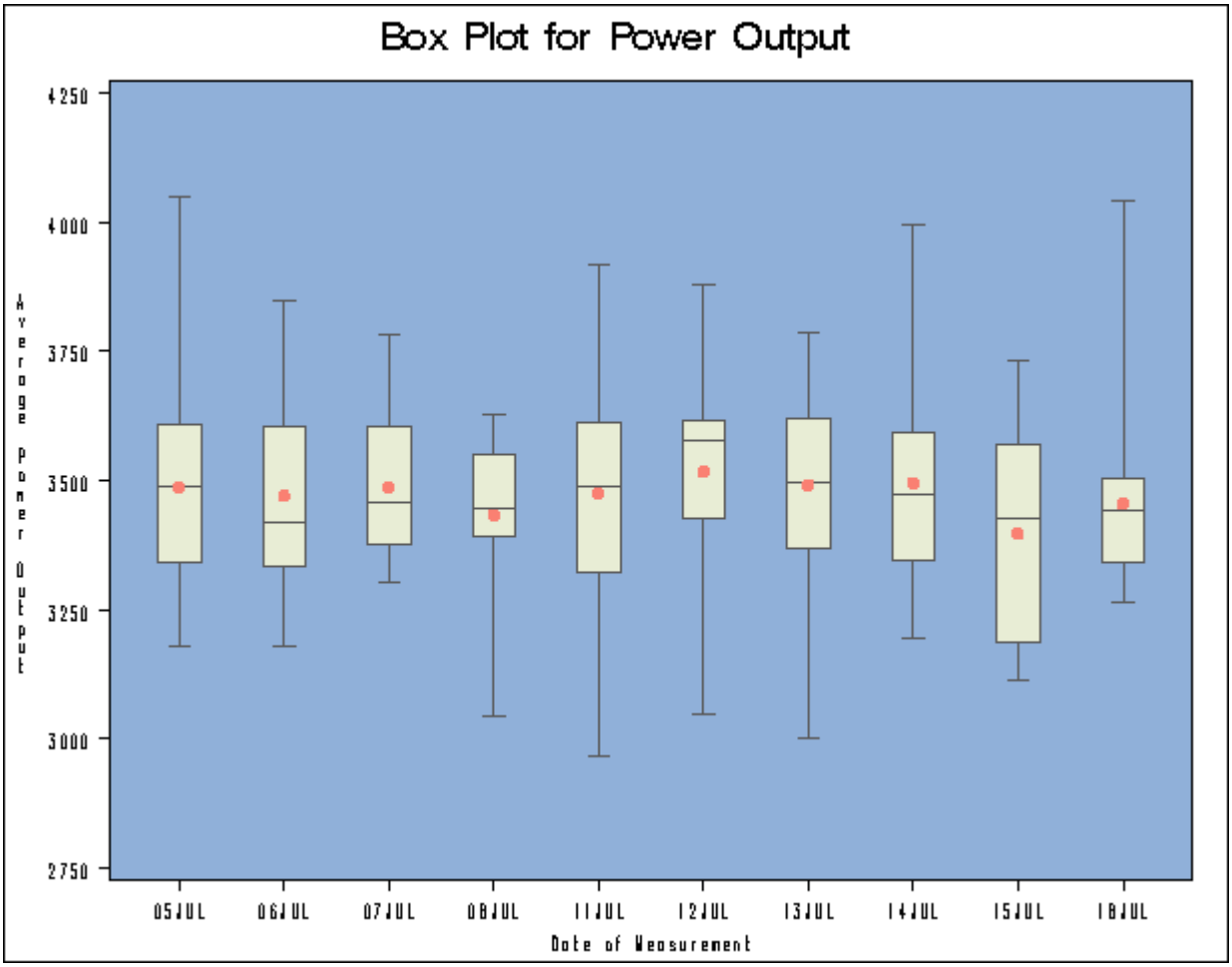
You can use this data set as input to the BOXPLOT procedure by specifying it with the HISTORY= option in the PROC BOXPLOT statement. Detailed requirements for HISTORY= data sets are presented in the section "HISTORY= Data Set" on page 794.

The following statements produce a box plot of the summary data from the Oilsum data set:

```
options nogstyle;
title 'Box Plot for Power Output';
symbol value=dot color=salmon;
proc boxplot history=Oilsum;
   plot KWatts*Day / cframe   = vligb
                     cboxes   = dagr
                     cboxfill = ywh;
run;
options gstyle;
goptions reset=symbol;
```

The NOGSTYLE system option causes PROC BOXPLOT to ignore ODS styles when producing the box plot. Instead, the SYMBOL statement and options specified after the slash (/) in the PLOT statement control its appearance. The GSTYLE system option restores the use of ODS styles for subsequent high-resolution graphics output. For more information about SYMBOL statements, see *SAS/GRAPH: Reference*. The resulting box plot is shown in Figure 24.3.

Figure 24.3 High-Resolution Box Plot with NOGSTYLE



Saving Summary Data with Outliers

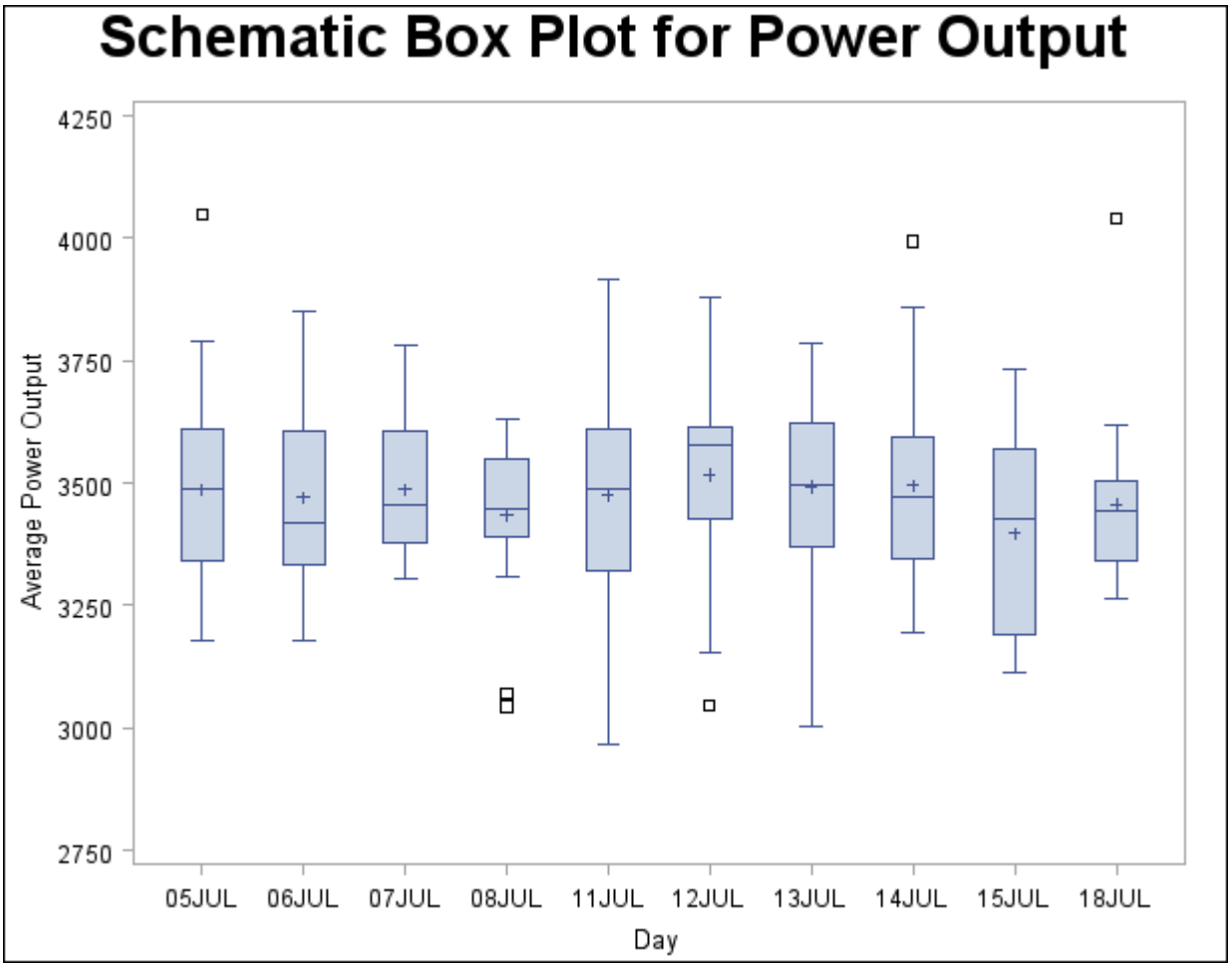
In a *schematic* box plot, outlier values within a group are plotted as separate points beyond the whiskers of the box-and-whiskers plot. See the section "Styles of Box Plots" on page 796 and the description of the BOXSTYLE= option on page 773 for a complete description of schematic box plots.

The following statements use the BOXSTYLE= option to produce a schematic box plot of the data from the Turbine data set. The OUTBOX= option creates a summary data set named OilSchematic.

```
title 'Schematic Box Plot for Power Output';
proc boxplot data=Turbine;
   plot KWatts*Day / boxstyle = schematic
                     outbox   = OilSchematic;
run;
```

The schematic box plot is shown in Figure 24.4. Note the outliers plotted with squares for several of the groups.

Figure 24.4 Schematic Box Plot of Power Output



Whereas the Oilsum data set from the section "Creating Box Plots from Summary Data" on page 754 contains a *variable* for each summary statistic and one observation per group, the OUTBOX= data set OilSchematic contains one *observation* for each summary statistic in each group. The _TYPE_ variable identifies the statistic and the _VALUE_ variable contains its value. In addition, the OilSchematic data set contains an observation recording each outlier value for each group. Figure 24.5 shows a partial listing of the OilSchematic data set.

Figure 24.5 The Summary Data Set OilSchematic

Schematic Box Plot for Power Output

Day	_VAR_	_TYPE_	_VALUE_
05JUL	KWatts	N	20.00
05JUL	KWatts	MIN	3180.00
05JUL	KWatts	Q1	3340.00
05JUL	KWatts	MEAN	3487.40
05JUL	KWatts	MEDIAN	3490.00
05JUL	KWatts	Q3	3610.00
05JUL	KWatts	MAX	4050.00
05JUL	KWatts	STDDEV	220.26
05JUL	KWatts	HIWHISKR	3789.00
05JUL	KWatts	HIGH	4050.00
06JUL	KWatts	N	20.00
06JUL	KWatts	MIN	3179.00
06JUL	KWatts	Q1	3333.50
06JUL	KWatts	MEAN	3471.65
06JUL	KWatts	MEDIAN	3419.50
06JUL	KWatts	Q3	3605.00
06JUL	KWatts	MAX	3849.00
06JUL	KWatts	STDDEV	210.43
07JUL	KWatts	N	20.00
07JUL	KWatts	MIN	3304.00
07JUL	KWatts	Q1	3376.00
07JUL	KWatts	MEAN	3488.30
07JUL	KWatts	MEDIAN	3456.50
07JUL	KWatts	Q3	3604.50
07JUL	KWatts	MAX	3781.00
07JUL	KWatts	STDDEV	147.02
08JUL	KWatts	N	20.00
08JUL	KWatts	MIN	3045.00
08JUL	KWatts	Q1	3390.50
08JUL	KWatts	MEAN	3434.20
08JUL	KWatts	MEDIAN	3447.00
08JUL	KWatts	Q3	3550.00
08JUL	KWatts	MAX	3629.00
08JUL	KWatts	STDDEV	157.64
08JUL	KWatts	LOWHISKR	3309.00
08JUL	KWatts	LOW	3070.00
08JUL	KWatts	LOW	3045.00
11JUL	KWatts	N	20.00
11JUL	KWatts	MIN	2968.00
11JUL	KWatts	Q1	3321.00

Observations with the _TYPE_ variable values “HIGH” and “LOW” contain outlier values. If you want to use a summary data set to re-create a schematic box plot, you *must* create an OUTBOX= data set in order to save the outlier data.

Syntax: BOXPLOT Procedure

The syntax for the BOXPLOT procedure is as follows:

PROC BOXPLOT *options* ;
BY *variables* ;
ID *variables* ;
INSET *keywords* < */options* > ;
INSETGROUP *keywords* < */ options* > ;
PLOT *analysis-variable*group-variable* < *(block-variables)* > < *=symbol-variable* > < */ options* > ;

Both the PROC BOXPLOT and PLOT statements are required. You can specify any number of PLOT statements within a single PROC BOXPLOT invocation.

PROC BOXPLOT Statement

PROC BOXPLOT *options* ;

The PROC BOXPLOT statement starts the BOXPLOT procedure. The following options can appear in the PROC BOXPLOT statement.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an ANNOTATE= type data set, as described in *SAS/GRAPH: Reference*, which enhances high-resolution box plots requested in subsequent PLOT statements.

BOX=*SAS-data-set*

names an input data set containing group summary statistics and outlier values. Typically, this data set is created as an OUTBOX= data set in a previous run of PROC BOXPLOT. Each group summary statistic or outlier value is recorded in a separate observation in a BOX= data set, so there are multiple observations per group. You cannot use a BOX= data set together with a DATA= or HISTORY= data set. If you do not specify one of these input data sets, the procedure uses the most recently created SAS data set as a DATA= data set.

DATA=*SAS-data-set*

names an input data set containing raw data to be analyzed. You cannot use a DATA= data set together with a BOX= or HISTORY= data set. If you do not specify one of these input data sets, the procedure uses the most recently created SAS data set as a DATA= data set.

GOUT=< *libref.* >*output catalog*

specifies the SAS catalog in which to save high-resolution graphics output that is produced by the BOXPLOT procedure. If you omit the libref, PROC BOXPLOT looks for the catalog in the temporary library called WORK and creates the catalog if it does not exist.

HISTORY=*SAS-data-set*
HIST=*SAS-data-set*
names an input data set containing group summary statistics. Typically, this data set is created as an OUTHISTORY= data set in a previous run of PROC BOXPLOT, but it can also be created using a SAS summarization procedure such as the MEANS procedure. The HISTORY= data set can contain only one observation for each value of the group variable. You cannot use a HISTORY= data set with a DATA= or BOX= data set. If you do not specify one of these three input data sets, PROC BOXPLOT uses the most recently created data set as a DATA= data set.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC BOXPLOT to obtain separate box plots for each group defined by the levels of the BY variables. When you specify a BY statement, the input data set must be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the BOXPLOT procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide.*

ID Statement

ID *variables* **;**

The ID statement specifies variables used to identify observations. The ID variables must be variables in the input data set.

If you specify the keyword SCHEMATICID or SCHEMATICIDFAR with the BOXSTYLE= option, the value of an ID variable is used to label each extreme observation. When you specify a BOX= data set, the label values come from the variable _ID_, if it is present in the data set. When you specify a DATA= or HISTORY= input data set, or a BOX= data set that does not contain the variable _ID_, the labels come from the first variable listed in the ID statement. If ID statement is specified, the outliers are not labeled.

INSET Statement

INSET *keywords* < / options > **;**

Each PLOT statement in the BOXPLOT procedure is followed by a series of zero or more INSET and INSETGROUP statements. Each INSET statement in that series produces one inset in the box plot produced by the preceding PLOT statement. If the box plot occupies multiple panels, the inset appears on each panel.

The data requested using the *keywords* are displayed in the order in which they are specified. Summary statistics requested with an INSET statement are calculated using the observations in all groups.

keywords identify summary statistics or other data to be displayed in the inset. By default, inset statistics are identified with appropriate labels, and numeric values are printed using appropriate formats. However, you can provide customized labels and formats. You provide the customized label by specifying the *keyword* for that statistic followed by an equal sign (=) and the label in quotes. Labels can have up to 24 characters. You provide the numeric format in parentheses after the *keyword*. Note that if you specify both a label and a format for a statistic, the label must appear before the format.

The available keywords are listed in Table 24.1.

options control the appearance of the inset. Table 24.2 lists all the options in the INSET statement. Complete descriptions for each option follow.

Table 24.1 INSET Statement Keywords

Keyword	Description
DATA=	(label, value) pairs from *SAS-data-set*
MEAN	mean of all observations
MIN	minimum observed value
MAX	maximum observed value
NMIN	minimum group size
NMAX	maximum group size
NOBS	number of observations in box plot
STDDEV	pooled standard deviation

The DATA= keyword specifies a SAS data set containing (label, value) pairs to be displayed in an inset. The data set must contain the variables _LABEL_ and _VALUE_. _LABEL_ is a character variable of up to 24 characters whose values provide labels for inset entries. _VALUE_ can be character or numeric, and provides values displayed in the inset. The label and value from each observation in the DATA= data set occupy one line in the inset.

The *pooled standard deviation* requested with the STDDEV keyword is defined as

$$s_p = \sqrt{\frac{\sum_{i=1}^{N} s_i^2 (n_i - 1)}{\sum_{i=1}^{N} (n_i - 1)}}$$

where N is the number of groups, n_i is the size of the ith group, and s_i^2 is the variance of the ith group.

Table 24.2 INSET Statement Options

Option	Description
CFILL=*color* | BLANK	specifies color of inset background
CFILLH=*color*	specifies color of inset header background
CFRAME=*color*	specifies color of inset frame
CHEADER=*color*	specifies color of inset header text
CSHADOW=*color*	specifies color of inset drop shadow
CTEXT=*color*	specifies color of inset text
DATA	specifies data units for POSITION=(x, y) coordinates
FONT=*font*	specifies font of inset text
FORMAT=*format*	specifies format of values in inset
HEADER='*string*'	specifies inset header text
HEIGHT=*value*	specifies height of inset and header text
NOFRAME	suppresses frame around inset
POSITION=*position*	specifies position of inset
REFPOINT=BR|BL|TR|TL	specifies reference point of inset positioned with POSITION=(x, y) coordinates

Following are descriptions of the options that you can specify in the INSET statement after a slash (/).

CFILL=*color* **| BLANK**

specifies the color of the inset background (including the header background if you do not specify the CFILLH= option).

If you do not specify the CFILL= option, then by default the background is empty. This means that items that overlap the inset (such as box-and-whiskers plots or reference lines) show through the inset. If you specify any value for the CFILL= option, then overlapping items no longer show through the inset. Specify CFILL=BLANK to leave the background uncolored and also to prevent items from showing through the inset.

CFILLH=*color*

specifies the color of the header background. By default, if you do not specify a CFILLH= color, the CFILL= color is used.

CFRAME=*color*

specifies the color of the frame around the inset. By default, the frame is the same color as the axis of the plot.

CHEADER=*color*

specifies the color of the header text. By default, if you do not specify a CHEADER= color, the INSET statement CTEXT= color is used.

CSHADOW=*color*

CS=*color*

specifies the color of the drop shadow. If you do not specify the CSHADOW= option, a drop shadow is not displayed.

CTEXT=*color*

CT=*color*

specifies the color of the text in the inset. By default, the inset text color is the same as the other text in the box plot.

DATA

specifies that data coordinates be used in positioning the inset with the POSITION= option. The DATA option is available only when you specify POSITION= (x, y), and it must be placed immediately after the coordinates (x, y). See the entry for the POSITION= option.

FONT=*font*

specifies the font of the text.

FORMAT=*format*

specifies a format for all the values displayed in an inset. If you specify a format for a particular statistic, then this format overrides the format you specified with the FORMAT= option.

HEADER='*string***'**

specifies the header text. The *string* can be up to 40 characters. If you do not specify the HEADER= option, no header line appears in the inset.

HEIGHT=*value*

specifies the height of the inset and header text.

NOFRAME

suppresses the frame drawn around the inset.

POSITION=*position*

POS=*position*

determines the position of the inset. The *position* can be a compass point keyword, a margin keyword, or a pair of coordinates (x, y). You can specify coordinates in axis percent units or axis data units. For more information, see the section "Positioning Insets" on page 800. By default, POSITION=NW, which positions the inset in the upper-left (northwest) corner of the plot.

REFPOINT=BR | BL | TR | TL

RP=BR | BL | TR | TL

specifies the reference point for an inset that is positioned by a pair of coordinates with the POSITION= option. Use the REFPOINT= option with POSITION= coordinates. The REFPOINT= option specifies which corner of the inset frame you want positioned at coordinates (x, y). The keywords BL, BR, TL, and TR represent bottom left, bottom right, top left, and top right, respectively. The default is REFPOINT=BL.

If you specify the position of the inset as a compass point or margin keyword, the REFPOINT= option is ignored.

INSETGROUP Statement

INSETGROUP *keywords* < / options > **;**

Each PLOT statement in the BOXPLOT procedure is followed by a series of zero or more INSET and INSETGROUP statements. Each INSETGROUP statement in that series displays statistics associated with individual groups in the box plot produced by the preceding PLOT statement. No more than two INSETGROUP statements can be associated with a given PLOT statement: one that displays group statistics above the box plot and one that displays group statistics below it. The data requested using the *keywords* are displayed in the order in which they are specified.

keywords identify summary statistics to be displayed in the insets. By default, inset statistics are identified with appropriate labels, and numeric values are printed using appropriate formats. However, you can provide customized labels and formats. You provide the customized label by specifying the *keyword* for that statistic followed by an equal sign (=) and the label in quotes. Labels can have up to 24 characters. You provide the numeric format in parentheses after the *keyword*. Note that if you specify both a label and a format for a statistic, the label must appear before the format. The keywords are listed in Table 24.3.

options control the appearance of the insets. Table 24.4 lists all the options in the INSETGROUP statement. Complete descriptions for each option follow.

Table 24.3 INSETGROUP Statement Keywords

Keyword	Description
MEAN	group mean
MIN	group minimum value
MAX	group maximum value
N	number of observations in group
NHIGH	number of outliers above upper fence
NLOW	number of outliers below lower fence
NOUT	total number of outliers in group
Q1	first quartile of group values
Q2	second quartile of group values
Q3	third quartile of group values
RANGE	range of group values
STDDEV	group standard deviation

Table 24.4 lists the options available in the INSETGROUP statement.

Table 24.4 INSETGROUP Statement Options

Option	Description
CFILL=*color* | BLANK	specifies color of inset background
CFILLH=*color*	specifies color of inset header background
CFRAME=*color*	specifies color of inset frame
CHEADER=*color*	specifies color of inset header text
CTEXT=*color*	specifies color of inset text
FONT=*font*	specifies font of inset text
FORMAT=*format*	specifies format of values in inset
HEADER='*string*'	specifies inset header text
HEIGHT=*value*	specifies height of inset and header text
NOFRAME	suppresses frame around inset
POSITION=*position*	specifies position of inset

Following are descriptions of the options that you can specify in the INSETGROUP statement after a slash (/).

CFILL=*color*

specifies the color of the inset background (including the header background if you do not specify the CFILLH= option). If you do not specify the CFILL= option, then by default the background is empty.

CFILLH=*color*

specifies the color of the header background. By default, if you do not specify a CFILLH= color, the CFILL= color is used.

CFRAME=*color*

specifies the color of the frame around the inset. By default, the frame is the same color as the axis of the plot.

CHEADER=*color*

specifies the color of the header text. By default, if you do not specify a CHEADER= color, the CTEXT= color is used.

CTEXT=*color*

CT=*color*

specifies the color of the inset text. By default, the inset text color is the same as the other text in the plot.

FONT=*font*

specifies the font of the inset text. By default, the font is SIMPLEX.

FORMAT=*format*

specifies a format for all the values displayed in an inset. If you specify a format for a particular statistic, then this format overrides the format you specified with the FORMAT= option.

HEADER='*string*'

specifies the header text. The *string* can be up to 40 characters. If you do not specify the HEADER= option, no header line appears in the inset.

HEIGHT=*value*

specifies the height of the inset and header text.

NOFRAME

suppresses the frame drawn around the inset.

POSITION=*position*

POS=*position*

determines the position of the inset. Valid positions are TOP, TOPOFF, AXIS, and BOTTOM. By default, POSITION=TOP.

Position Keyword	Description
TOP	top of plot, immediately above axis frame
TOPOFF	top of plot, offset from axis frame
AXIS	bottom of plot, immediately above horizontal axis
BOTTOM	bottom of plot, below horizontal axis label

PLOT Statement

PLOT *(analysis-variables)*group-variable* <*(block-variables)*> <*=symbol-variable*> </ *options* > ;

You can specify multiple PLOT statements after the PROC BOXPLOT statement. The components of the PLOT statement are as follows:

analysis-variables identify one or more variables to be analyzed. An analysis variable is required. If you specify more than one analysis variable, enclose the list in parentheses. For example, the following statements request distinct box plots for the variables Weight, Length, and Width:

```
proc boxplot data=Summary;
   plot (Weight Length Width)*Day;
run;
```

group-variable specifies the variable that identifies groups in the data. The group variable is required. In the preceding PLOT statement, Day is the group variable.

block-variables specify optional variables that group the data into blocks of consecutive groups. These blocks are labeled in a legend, and each block variable provides one level of labels in the legend.

symbol-variable specifies an optional variable whose levels (unique values) determine the symbol marker used to plot the means. Distinct symbol markers are displayed for points corresponding to the various levels of the symbol variable. You can specify the symbol markers with SYMBOL*n* statements (refer to *SAS/GRAPH: Reference* for complete details).

options enhance the appearance of the box plot, request additional analyses, save results in data sets, and so on. Complete descriptions of each option follow.

Table 24.5 lists all options in the PLOT statement by function.

PLOT Statement Options

Table 24.5 PLOT Statement Options

Option	Description
Options for Controlling Box Appearance	
BOXCONNECT=	connects features of adjacent box-and-whiskers plots with line segments
BOXSTYLE=	specifies style of box-and-whiskers plots
BOXWIDTH=	specifies width of box-and-whiskers plots
BOXWIDTHSCALE=	specifies that widths of box-and-whiskers plots vary proportionately to group size

Table 24.5 *continued*

Option	Description
CBOXES=	specifies color for outlines of box-and-whiskers plots
CBOXFILL=	specifies fill color for interior of box-and-whiskers plots
IDCOLOR=	specifies outlier symbol color in schematic box-and-whiskers plots
IDCTEXT=	specifies outlier label color in schematic box-and-whiskers plots
IDFONT=	specifies outlier label font in schematic box-and-whiskers plots
IDHEIGHT=	specifies outlier label height in schematic box-and-whiskers plots
IDSYMBOL=	specifies outlier symbol in schematic box-and-whiskers plots
LBOXES=	specifies line types for outlines of box-and-whiskers plots
NOSERIFS	eliminates serifs from whiskers of box-and-whiskers plots
NOTCHES	specifies that box-and-whiskers plots be notched
PCTLDEF=	specifies percentile definition used for box-and-whiskers plots
Options for Plotting and Labeling Points	
ALLLABEL=	labels means of box-and-whiskers plots
CLABEL=	specifies color for labels requested with ALLLABEL= option
CCONNECT=	specifies color for line segments requested with BOXCONNECT= option
LABELANGLE=	specifies angle for labels requested with ALLLABEL= option
SYMBOLLEGEND=	specifies LEGEND statement for levels of symbol variable
SYMBOLORDER=	specifies order in which symbols are assigned for levels of symbol variable
Reference Line Options	
CHREF=	specifies color for lines requested by HREF= option
CVREF=	specifies color for lines requested by VREF= option
FRONTREF	draws reference lines in front of boxes
HREF=	requests reference lines perpendicular to horizontal axis
HREFLABELS=	specifies labels for HREF= lines
HREFLABPOS=	specifies position of HREFLABELS= labels
LHREF=	specifies line type for HREF= lines
LVREF=	specifies line type for VREF= lines
NOBYREF	specifies that reference line information in a data set be applied uniformly to plots created for all BY groups
VREF=	requests reference lines perpendicular to vertical axis
VREFLABELS=	specifies labels for VREF= lines
VREFLABPOS=	specifies position of VREFLABELS= labels
Block Variable Legend Options	
BLOCKLABELPOS=	specifies position of label for block variable legend

Table 24.5 *continued*

Option	Description
BLOCKLABTYPE=	specifies text size of block variable legend
BLOCKPOS=	specifies vertical position of block variable legend
BLOCKREP	repeats identical consecutive labels in block variable legend
CBLOCKLAB=	specifies colors for filling frames enclosing block variable labels
CBLOCKVAR=	specifies colors for filling background of block variable legend
Axis and Axis Label Options	
CAXIS=	specifies color for axis lines and tick marks
CFRAME=	specifies fill color for frame for plot area
CONTINUOUS	produces horizontal axis for continuous group variable values (traditional graphics only)
CTEXT=	specifies color for tick mark values and axis labels
HAXIS=	specifies major tick mark values for horizontal axis
HEIGHT=	specifies height of axis label and axis legend text
HMINOR=	specifies number of minor tick marks between major tick marks on horizontal axis
HOFFSET=	specifies length of offset at both ends of horizontal axis
NOHLABEL	suppresses horizontal axis label
NOTICKREP	specifies that only first occurrence of repeated, adjacent character group values be labeled on horizontal axis
NOVANGLE	requests vertical axis labels that are strung out vertically
SKIPHLABELS=	specifies thinning factor for tick mark labels on horizontal axis
TURNHLABELS	requests horizontal tick labels that are strung out vertically
VAXIS=	specifies major tick mark values for vertical axis
VFORMAT=	specifies format for vertical axis tick marks
VMINOR=	specifies number of minor tick marks between major tick marks on vertical axis
VOFFSET=	specifies length of offset at both ends of vertical axis
VZERO	forces origin to be included in vertical axis
WAXIS=	specifies width of axis lines
Input Data Set Options	
MISSBREAK	specifies that a missing value between identical character group values signify the start of a new group
Output Data Set Options	
OUTBOX=	produces an output data set containing group summary statistics and outlier values
OUTHISTORY=	produces an output data set containing group summary statistics

Table 24.5 *continued*

Option	Description
Graphical Enhancement Options	
ANNOTATE=	specifies annotate data set that adds features to box plot
BWSLEGEND	displays a legend identifying the function of group size specified with BOXWIDTHSCALE= option
DESCRIPTION=	specifies string that appears in description field of PROC GREPLAY master menu for high-resolution graphics box plot
FONT=	specifies font for labels and legends on plots
HORIZONTAL	requests a horizontal box plot with ODS Graphics
HTML=	specifies URLs to be associated with box-and-whiskers plots
NAME=	specifies name that appears in name field of PROC GREPLAY master menu for high-resolution graphics box plot
NLEGEND	requests legend displaying group sizes
OUTHIGHHTML=	specifies URLs to be associated with high outliers on box-and-whiskers plots
OUTLOWHTML=	specifies URLs to be associated with low outliers on box-and-whiskers plots
PAGENUM=	specifies form of label used in pagination
PAGENUMPOS=	specifies position of page number requested with PAGENUM= option
Grid Options	
CGRID=	specifies color for grid requested with ENDGRID or GRID option
ENDGRID	adds grid after last box-and-whiskers plot
GRID	adds grid to box plot
LENDGRID=	specifies line type for grid requested with ENDGRID option
LGRID=	specifies line type for grid requested with GRID option
WGRID=	specifies width of grid lines
Plot Layout Options	
INTERVAL=	specifies natural time interval between consecutive group positions when time, date, or datetime format is associated with numeric group variable
INTSTART=	specifies first major tick mark value on horizontal axis when date, time, or datetime format is associated with numeric group variable
MAXPANELS=	specifies maximum number of panels used for box plot
NOCHART	suppresses creation of box plot
NOFRAME	suppresses frame for plot area
NPANELPOS=	specifies number of group positions per panel
REPEAT	repeats last group position on panel as first group position of next panel

Table 24.5 *continued*

Option	Description
TOTPANELS=	specifies number of panels to be used to display box plot
Overlay Options	
CCOVERLAY=	specifies colors for line segments connecting points on overlays
COVERLAY=	specifies colors for points on overlays
LOVERLAY=	specifies line types for line segments connecting points on overlays
NOOVERLAYLEGEND	suppresses overlay legend
OVERLAY=	specifies variables to be plotted on overlays
OVERLAYHTML=	specifies URLs to be associated with overlay plot points
OVERLAYID=	specifies labels for overlay plot points
OVERLAYLEGLAB=	specifies label for overlay legend
OVERLAYSYM=	specifies symbols used for overlays
OVERLAYSYMHT=	specifies heights for overlay symbols
WOVERLAY=	specifies widths for line segments connecting points on overlays
Clipping Options	
CCLIP=	specifies color for plot symbol for clipped points
CLIPFACTOR=	determines extent to which extreme values are clipped
CLIPLEGEND=	specifies text for clipping legend
CLIPLEGPOS=	specifies position of clipping legend
CLIPSUBCHAR	specifies substitution character for CLIPLEGEND= text
CLIPSYMBOL=	specifies plot symbol for clipped points
CLIPSYMBOLHT=	specifies symbol marker height for clipped points
COVERLAYCLIP=	specifies color for clipped points on overlays
OVERLAYCLIPSYM=	specifies symbol for clipped points on overlays
OVERLAYCLIPSYMHT=	specifies symbol height for clipped points on overlays
Options for Box Plots Produced Using Styles	
BLOCKVAR=	groups block legends whose backgrounds are filled with colors from style
BOXES=	groups boxes whose outlines are draw with colors from style
BOXFILL=	groups boxes that are filled with colors from style

Following are explanations of the options you can specify in the PLOT statement after a slash (/).

ALLLABEL=VALUE | (*variable***)**
labels the point plotted for the mean of each box-and-whiskers plot with its VALUE or with the value of a *variable* in the input data set.

ANNOTATE=*SAS-data-set*
specifies an ANNOTATE= type data set, as described in *SAS/GRAPH: Reference*.

BLOCKLABELPOS=ABOVE | LEFT

specifies the position of a block variable label in the block legend. The keyword ABOVE places the label immediately above the legend, and LEFT places the label to the left of the legend. Use the keyword LEFT with labels that are short enough to fit in the margin of the plot; otherwise, they are truncated. The default keyword is ABOVE.

BLOCKLABTYPE=SCALED | TRUNCATED

BLOCKLABTYPE=*height*

specifies how lengthy block variable values are treated when there is insufficient space to display them in the block legend. If you specify BLOCKLABTYPE=SCALED, the values are uniformly reduced in height so that they fit. If you specify BLOCKLABTYPE=TRUNCATED, lengthy values are truncated on the right until they fit. You can also specify a text height in vertical percent screen units for the values. By default, lengthy values are not displayed. For more information, see the section "Displaying Blocks of Data" on page 805.

BLOCKPOS=*n*

specifies the vertical position of the legend for the values of the block variables. Values of *n* and the corresponding positions are as follows. By default, BLOCKPOS=1.

n	Legend Position
1	top of plot, offset from axis frame
2	top of plot, immediately above axis frame
3	bottom of plot, immediately above horizontal axis
4	bottom of plot, below horizontal axis label

BLOCKREP

specifies that block variable values for all groups be displayed. By default, only the first block variable value in any block is displayed, and repeated block variable values are not displayed.

BLOCKVAR=*variable* | **(***variable-list***)**

specifies variables whose values are used to assign colors for filling the background of the legend associated with block variables. A list of BLOCKVAR= variables must be enclosed in parentheses. BLOCKVAR= variables are matched with block variables by their order in the respective variable lists. While the values of a CBLOCKVAR= variable are color names, values of a BLOCKVAR= variable are used to group block legends for assigning fill colors from the ODS style. Block legends with the same BLOCKVAR= variable value are filled with the same color.

BOXCONNECT=MEAN | MEDIAN | MAX | MIN | Q1 | Q3

BOXCONNECT

specifies that the points in adjacent box-and-whiskers plots representing group means, medians, maximum values, minimum values, first quartiles, or third quartiles be connected with line segments. If the BOXCONNECT option is specified without a keyword identifying the points to be connected, group means are connected. By default, no points are connected.

BOXES=(*variable*)

specifies a variable whose values are used to assign colors for the outlines of box-and-whiskers plots. While the values of a CBOXES= variable are color names, values of the BOXES= variable are used to group box-and-whiskers plots for assigning outline colors from the ODS style. The outlines of box-and-whiskers plots of groups with the same BOXES= variable value are drawn using the same color.

BOXFILL=(*variable*)

specifies a variable whose values are used to assign fill colors for box-and-whiskers plots. While the values of a CBOXFILL= variable are color names, values of the BOXFILL= variable are used to group box-and-whiskers plots for assigning fill colors from the ODS style. Box-and-whiskers plots of groups with the same BOXFILL= variable value are filled with the same color.

BOXSTYLE=*keyword*

specifies the style of the box-and-whiskers plots displayed. If you specify BOXSTYLE=SKELETAL, the whiskers are drawn from the edges of the box to the extreme values of the group. This plot is sometimes referred to as a skeletal box-and-whiskers plot. By default, the whiskers are drawn with serifs. You can specify the NOSERIFS option to draw the whiskers without serifs.

In the following descriptions, the terms *fence* and *far fence* refer to the distance from the first and third quartiles (25th and 75th percentiles, respectively), expressed in terms of the interquartile range (IQR). For example, the lower fence is located at $1.5 \times$ IQR below the 25th percentile; the upper fence is located at $1.5 \times$ IQR above the 75th percentile. Similarly, the lower far fence is located at $3 \times$ IQR below the 25th percentile; the upper far fence is located at $3 \times$ IQR above the 75th percentile.

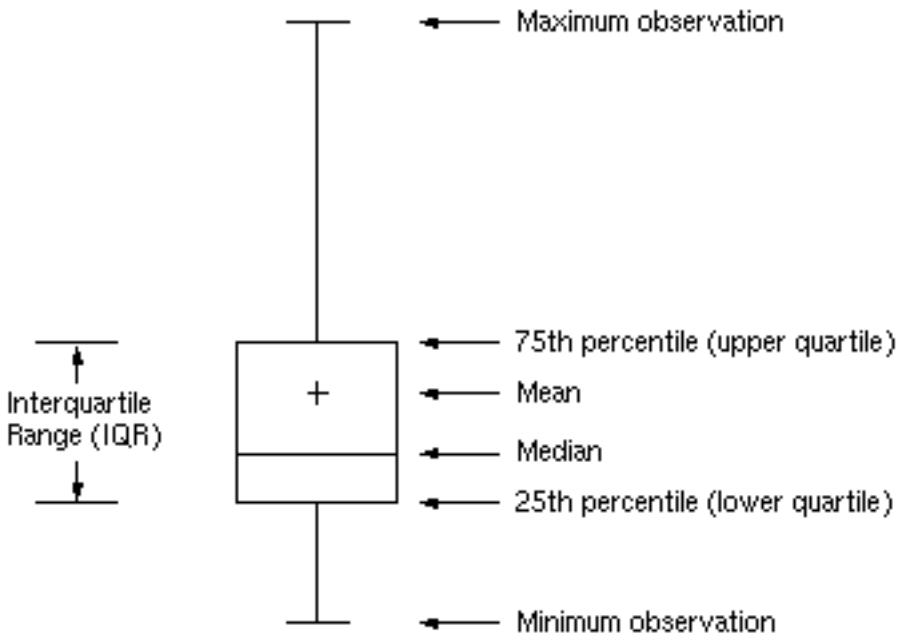
If you specify BOXSTYLE=SCHEMATIC, a whisker is drawn from the upper edge of the box to the largest observed value within the upper fence, and another is drawn from the lower edge of the box to the smallest observed value within the lower fence. Serifs are added to the whiskers by default. Observations outside the fences are identified with a special symbol; you can specify the shape and color for this symbol with the IDSYMBOL= and IDCOLOR= options. The default symbol is a square. This type of plot corresponds to the schematic box-and-whiskers plot described in Chapter 2 of Tukey (1977). See Figure 24.8 and the discussion in the section "Styles of Box Plots" on page 796 for more information.

If you specify BOXSTYLE=SCHEMATICID, a schematic box-and-whiskers plot is displayed in which an ID variable value is used to label the symbol marking each observation outside the upper and lower fences. A BOX= data set can contain a variable named _ID_ that is used as the ID variable. Otherwise, the first variable listed in the ID statement provides the labels.

If you specify BOXSTYLE=SCHEMATICIDFAR, a schematic box-and-whiskers plot is displayed in which the value of the ID variable is used to label the symbol marking each observation outside the lower and upper far fences. Observations between the fences and the far fences are identified with a symbol but are not labeled with the ID variable.

Figure 24.6 illustrates the elements of a skeletal box-and-whiskers plot.

Figure 24.6 Skeletal Box-and-Whiskers Plot



The skeletal style of the box-and-whiskers plot shown in Figure 24.6 is the default.

BOXWIDTH=*value*

specifies the width (in horizontal percent screen units) of the box-and-whiskers plots.

BOXWIDTHSCALE=*value*

specifies that the box-and-whiskers plot width is to vary proportionately to a particular function of the group size n. The function is determined by the *value*.

If you specify a positive value, the widths are proportional to n^{value}. In particular, if you specify BOXWIDTHSCALE=1, the widths are proportional to the group size. If you specify BOXWIDTHSCALE=0.5, the widths are proportional to $\sqrt{n}$, as described by McGill, Tukey, and Larsen (1978). If you specify BOXWIDTHSCALE=0, the widths are proportional to $\log(n)$. See Example 24.4 for an illustration of the BOXWIDTHSCALE= option.

You can specify the BWSLEGEND option to display a legend identifying the function of n used to determine the box-and-whiskers plot widths.

By default, the box widths are constant.

BWSLEGEND

displays a legend identifying the function of group size n specified with the BOXWIDTHSCALE= option. No legend is displayed if all group sizes are equal. The BWSLEGEND option is not applicable unless you also specify the BOXWIDTHSCALE= option.

CAXIS=*color*

CAXES=*color*

CA=*color*

specifies the color for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement.

CBLOCKLAB=*color* | **(***color-list***)**

specifies fill colors for the frames that enclose the block variable labels in a block legend. By default, these areas are not filled. Colors in the CBLOCKLAB= list are matched with block variables in the order in which they appear in the PLOT statement.

CBLOCKVAR=*variable* | **(***variable-list***)**

specifies variables whose values are colors for filling the background of the legend associated with block variables. CBLOCKVAR= variables are matched with block variables by their order in the respective variable lists. Each CBLOCKVAR= variable must be a character variable of no more than eight characters in the input data set, and its values must be valid SAS/GRAPH color names (refer to *SAS/GRAPH: Reference* for complete details). A list of CBLOCKVAR= variables must be enclosed in parentheses.

The procedure matches the CBLOCKVAR= variables with block variables in the order specified. That is, each block legend is filled with the color value of the CBLOCKVAR= variable of the first observation in each block. In general, values of the *i*th CBLOCKVAR= variable are used to fill the block of the legend corresponding to the *i*th block variable.

By default, fill colors are not used for the block variable legend. The CBLOCKVAR= option is available only when block variables are used in the PLOT statement.

CBOXES=*color* | **(***variable***)**

specifies the colors for the outlines of the box-and-whiskers plots created with the PLOT statement. You can use one of the following approaches:

- You can specify CBOXES=*color* to provide a single outline color for all the box-and-whiskers plots.
- You can specify CBOXES=*(variable)* to provide a distinct outline color for each box-and-whiskers plot as the value of the variable. The variable must be a character variable of up to eight characters in the input data set, and its values must be valid SAS/GRAPH color names (refer to *SAS/GRAPH: Reference* for complete details). The outline color of the plot displayed for a particular group is the value of the variable in the observations corresponding to this group. Note that if there are multiple observations per group in the input data set, the values of the variable should be identical for all the observations in a given group.

CBOXFILL=*color* | **(***variable***)**

specifies the interior fill colors for the box-and-whiskers plots. You can use one of the following approaches:

- You can specify CBOXFILL=*color* to provide a single color for all of the box-and-whiskers plots.

- You can specify CBOXFILL=*(variable)* to provide a distinct color for each box-and-whiskers plot as the value of the variable. The variable must be a character variable of up to eight characters in the input data set, and its values must be valid SAS/GRAPH color names (or the value EMPTY, which you can use to suppress color filling). Refer to *SAS/GRAPH: Reference* for complete details. The interior color of the box displayed for a particular group is the value of the variable in the observations corresponding to this group. Note that if there are multiple observations per group in the input data set, the values of the variable should be identical for all the observations in a given group.

By default, the interiors are not filled.

CCLIP=*color*

specifies a color for the plotting symbol that is specified with the CLIPSYMBOL= option to mark clipped values. The default color is the color specified in the COLOR= option in the SYMBOL1 statement.

CCONNECT=*color*

specifies the color for line segments connecting points on the plot. The default color is the color specified in the COLOR= option in the SYMBOL1 statement. This option is not applicable unless you also specify the BOXCONNECT= option.

CCOVERLAY=(*color-list***)**

specifies the colors for line segments connecting points on overlay plots. Colors in the CCOVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list. By default, points are connected by line segments of the same color as the plotted points. You can specify the value NONE to suppress the line segments connecting points of an overlay plot.

CFRAME=*color*

specifies the color for filling the rectangle enclosed by the axes and the frame. By default, this area is not filled. The CFRAME= option cannot be used in conjunction with the NOFRAME option.

CGRID=*color*

specifies the color for the grid requested by the ENDGRID or GRID option. By default, the grid is the same color as the axes.

CHREF=*color*

specifies the color for the lines requested by the HREF= option.

CLABEL=*color*

specifies the color for labels produced by the ALLLABEL= option. The default color is the CTEXT= color.

CLIPFACTOR=*factor*

requests clipping of extreme values on the box plot. The *factor* that you specify determines the extent to which these values are clipped, and it must be greater than 1.

For examples of the CLIPFACTOR= option, see Figure 24.17 and Figure 24.18. Related clipping options are CCLIP=, CLIPLEGEND=, CLIPLEGPOS=, CLIPSUBCHAR=, and CLIPSYMBOL=.

CLIPLEGEND='*label***'**
specifies the *label* for the legend that indicates the number of clipped boxes when the CLIPFACTOR= option is used. The *label* must be no more than 16 characters and must be enclosed in quotes. For an example, see Figure 24.18.

CLIPLEGPOS= TOP | BOTTOM
specifies the position for the legend that indicates the number of clipped boxes when the CLIPFACTOR= option is used. The keyword TOP or BOTTOM positions the legend at the top or bottom of the chart, respectively. Do not specify CLIPLEGPOS=TOP together with the BLOCKPOS=1 or BLOCKPOS=2 option. By default, CLIPLEGPOS=BOTTOM.

CLIPSUBCHAR='*character***'**
specifies a substitution character (such as '#') for the label provided with the CLIPLEGEND= option. The substitution character is replaced with the number of boxes that are clipped. For example, suppose that the following statements produce a chart in which three boxes are clipped:

```
proc boxplot data=Pistons;
   plot Diameter*Hour /
      clipfactor  = 1.5
      cliplegend  = 'Boxes clipped=#'
      clipsubchar = '#' ;
run;
```

Then the clipping legend displayed on the chart will be "Boxes clipped=3".

CLIPSYMBOL=*symbol*
specifies a plot symbol used to identify clipped points on the chart and in the legend when the CLIPFACTOR= option is used. You should use this option in conjunction with the CLIPFACTOR= option. The default *symbol* is CLIPSYMBOL=SQUARE.

CLIPSYMBOLHT=*value*
specifies the height for the symbol marker used to identify clipped points on the chart when the CLIPFACTOR= option is used. The default is the height specified with the H= option in the SYMBOL statement.

For general information about clipping options, refer to the section "Clipping Extreme Values" on page 807.

CONTINUOUS
specifies that numeric group variable values be treated as continuous values. By default, the values of a numeric group variable are considered discrete values unless the HAXIS= option is specified. **NOTE:** The CONTINUOUS option is not supported for ODS Graphics output. For more information, see the discussion in the section "Continuous Group Variables" on page 798.

COVERLAY=(*color-list***)**
specifies the colors used to plot overlay variables. Colors in the COVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list.

COVERLAYCLIP=*color*
specifies the color used to plot clipped values on overlay plots when the CLIPFACTOR= option is used.

CTEXT=*color*
specifies the color for tick mark values and axis labels. The default color is the color specified in the CTEXT= option in the most recent GOPTIONS statement.

CVREF=*color*
specifies the color for the lines requested by the VREF= option.

DESCRIPTION='*string*'
DES='*string*'
specifies a description of a box plot produced with high-resolution graphics. The description appears in the PROC GREPLAY master menu and can be no longer than 256 characters. The default description is the analysis variable name.

ENDGRID
adds a grid to the rightmost portion of the plot, beginning with the first labeled major tick mark position that follows the last box-and-whiskers plot. You can use the HAXIS= option to force space to be added to the horizontal axis.

FONT=*font*
specifies a font for labels and legends. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the GOPTIONS statement. Refer to *SAS/GRAPH: Reference* for more information about the GOPTIONS statement.

FRONTREF
draws reference lines specified with the HREF= and VREF= options in front of box-and-whiskers plots. By default, reference lines are drawn behind the box-and-whiskers plots and can be obscured by filled boxes.

GRID
adds a grid to the box plot. Grid lines are horizontal lines positioned at labeled major tick marks, and they cover the length and height of the plotting area.

HAXIS=*value-list*
HAXIS=AXIS*n*
specifies tick mark values for the horizontal (group) axis. If the group variable is numeric, the values must be numeric and equally spaced. If the group variable is character, values must be quoted strings of up to 16 characters. Optionally, you can specify an axis name defined in a previous AXIS statement. Refer to *SAS/GRAPH: Reference* for more information about the AXIS statement.

If you are producing traditional graphics, specifying the HAXIS= option with a numeric group variable causes the group variable values to be treated as continuous values. For more information, see the description of the CONTINUOUS option and the discussion in the section "Continuous Group Variables" on page 798. Numeric values can be given in an explicit

or implicit list. If a date, time, or datetime format is associated with a numeric group variable, SAS datetime literals can be used. Examples of HAXIS= lists follow:

- haxis=0 2 4 6 8 10
- haxis=0 to 10 by 2
- haxis='LT12A' 'LT12B' 'LT12C' 'LT15A' 'LT15B' 'LT15C'
- haxis='20MAY88'D to '20AUG88'D by 7
- haxis='01JAN88'D to '31DEC88'D by 30

If the group variable is numeric, the HAXIS= list must span the group variable values. If the group variable is character, the HAXIS= list must include all of the group variable values. You can add group positions to the box plot by specifying HAXIS= values that are not group variable values.

If you specify a large number of HAXIS= values, some of these can be thinned to avoid collisions between tick mark labels. To avoid thinning, use one of the following methods.

- Shorten values of the group variable by eliminating redundant characters. For example, if your group variable has values LOT1, LOT2, LOT3, and so on, you can use the SUBSTR function in a DATA step to eliminate LOT from each value, and you can modify the horizontal axis label to indicate that the values refer to lots.
- Use the TURNHLABELS option to turn the labels vertically.
- Use the NPANELPOS= option to force fewer group positions per panel.

HEIGHT=*value*

specifies the height (in vertical screen percent units) of the text for axis labels and legends. This value takes precedence over the HTEXT= value specified in the GOPTIONS statement. This option is recommended for use with fonts specified with the FONT= option or with the FTEXT= option in the GOPTIONS statement. Refer to *SAS/GRAPH: Reference* for complete information about the GOPTIONS statement.

HMINOR=*n*

HM=*n*

specifies the number of minor tick marks between major tick marks on the horizontal axis. Minor tick marks are not labeled. The default is HMINOR=0.

HOFFSET=*value*

specifies the length (in percent screen units) of the offset at both ends of the horizontal axis. You can eliminate the offset by specifying HOFFSET=0.

HORIZONTAL

Experimental

produces a horizontal box plot, with group variable values on the vertical axis and analysis variable values on the horizontal axis. The HORIZONTAL option is supported only with ODS Graphics.

HREF=*value-list*

HREF=*SAS-data-set*

draws reference lines perpendicular to the horizontal (group) axis on the box plot. You can use this option in the following ways:

- You can specify the values for the lines with an HREF= list. If the group variable is numeric, the values must be numeric. If the group variable is character, the values must be quoted strings of up to 16 characters. If the group variable is formatted, the values must be given as internal values. Examples of HREF= values follow:

```
href=5
href=5 10 15 20 25 30
href='Shift 1' 'Shift 2' 'Shift 3'
```

- You can specify reference line values as the values of a variable named _REF_ in an HREF= data set. The type and length of _REF_ must match those of the group variable specified in the PLOT statement. Optionally, you can provide labels for the lines as values of a variable named _REFLAB_, which must be a character variable of up to 16 characters. If you want distinct reference lines to be displayed in plots for different analysis variables specified in the PLOT statement, you must include a character variable named _VAR_, whose values are the analysis variable names. If you do not include the variable _VAR_, all of the lines are displayed in all of the plots. Each observation in an HREF= data set corresponds to a reference line. If BY variables are used in the input data set, the same BY variable structure must be used in the reference line data set unless you specify the NOBYREF option.

Unless the CONTINUOUS or HAXIS= option is specified, numeric group variable values are treated as discrete values, and only HREF= values matching these discrete values are valid. Other values are ignored.

HREFLABELS='*label1*' **...**'*labeln*'

HREFLABEL='*label1*' **...**'*labeln*'

HREFLAB='*label1*' **...**'*labeln*'

specifies labels for the reference lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

HREFLABPOS=*n*

specifies the vertical position of the HREFLABEL= label, as described in the following table. By default, n=2.

HREFLABPOS=	Label Position
1	along top of plot area
2	staggered from top to bottom of plot area
3	along bottom of plot area
4	staggered from bottom to top of plot area

HTML=*variable*

specifies uniform resource locators (URLs) as values of the specified character variable (or formatted values of a numeric variable). These URLs are associated with box-and-whiskers plots when graphics output is directed into HTML. The value of the HTML= variable should be the same for each observation with a given value of the group variable.

IDCOLOR=*color*

specifies the color of the symbol marker used to identify outliers in schematic box-and-whiskers plots (that is, when you specify the keyword SCHEMATIC, SCHEMATICID, or SCHEMATICIDFAR with the BOXSTYLE= option). The default color is the color specified with the CBOXES= option.

IDCTEXT=*color*

specifies the color for the text used to label outliers when you specify the keyword SCHEMATICID or SCHEMATICIDFAR with the BOXSTYLE= option. The default value is the color specified with the CTEXT= option.

IDFONT=*font*

specifies the font for the text used to label outliers when you specify the keyword SCHEMATICID or SCHEMATICIDFAR with the BOXSTYLE= option. The default font is SIMPLEX.

IDHEIGHT=*value*

specifies the height for the text used to label outliers when you specify the keyword SCHEMATICID or SCHEMATICIDFAR with the BOXSTYLE= option. The default value is the height specified with the HTEXT= option in the GOPTIONS statement. Refer to *SAS/GRAPH: Reference* for complete information about the GOPTIONS statement.

IDSYMBOL=*symbol*

specifies the symbol marker used to identify outliers in schematic box plots. The default symbol is SQUARE.

INTERVAL=DAY | DTDAY | HOUR | MINUTE | MONTH | QTR | SECOND

specifies the natural time interval between consecutive group positions when a time, date, or datetime format is associated with a numeric group variable. By default, the INTERVAL= option uses the number of group positions per panel (screen or page) that you specify with the NPANELPOS= option. The default time interval keywords for various time formats are shown in the following table.

Format	Default Keyword	Format	Default Keyword
DATE	DAY	MONYY	MONTH
DATETIME	DTDAY	TIME	SECOND
DDMMYY	DAY	TOD	SECOND
HHMM	HOUR	WEEKDATE	DAY
HOUR	HOUR	WORDDATE	DAY
MMDDYY	DAY	YYMMDD	DAY
MMSS	MINUTE	YYQ	QTR

You can use the INTERVAL= option to modify the effect of the NPANELPOS= option, which specifies the number of group positions per panel. The INTERVAL= option enables you to match the scale of the horizontal axis to the scale of the group variable without having to associate a different format with the group variable.

For example, suppose that your formatted group values span an overall time interval of 100 days and a DATETIME format is associated with the group variable. Since the default interval for the DATETIME format is DTDAY and since NPANELPOS=25 by default, the plot is displayed with four panels.

Now, suppose that your data span an overall time interval of 100 hours and a DATETIME format is associated with the group variable. The plot for these data is created in a single panel, but the data occupy only a small fraction of the plot since the scale of the data (hours) does not match that of the horizontal axis (days). If you specify INTERVAL=HOUR, the horizontal axis is scaled for 25 hours, matching the scale of the data, and the plot is displayed with four panels.

You should use the INTERVAL= option only in conjunction with the CONTINUOUS or HAXIS= option, which produces a horizontal axis of continuous group variable values. For more information, see the descriptions of the CONTINUOUS and HAXIS= options, and the discussion in the section "Continuous Group Variables" on page 798.

INTSTART=*value*

specifies the starting value for a numeric horizontal axis when a date, time, or datetime format is associated with the group variable. If the value specified is greater than the first group variable value, this option has no effect.

LABELANGLE=*angle*

specifies the angle at which labels requested with the ALLLABEL= option are drawn. A positive angle rotates the labels counterclockwise; a negative angle rotates them clockwise. By default, labels are oriented horizontally.

LBOXES=*linetype*

LBOXES=(*variable***)**

specifies the line types for the outlines of the box-and-whiskers plots. You can use one of the following approaches:

- You can specify LBOXES=*linetype* to provide a single linetype for all of the box-and-whiskers plots.
- You can specify LBOXES=*(variable)* to provide a distinct line type for each box-and-whiskers plot. The variable must be a numeric variable in the input data set, and its values must be valid SAS/GRAPH linetype values (numbers ranging from 1 to 46). The line type for the plot displayed for a particular group is the value of the variable in the observations corresponding to this group. Note that if there are multiple observations per group in the input data set, the values of the variable should be identical for all of the observations in a given group.

The default value is 1, which produces solid lines. Refer to the description of the SYMBOL statement in *SAS/GRAPH: Reference* for more information about valid linetypes.

LENDGRID=*linetype*

specifies the line type for the grid requested with the ENDGRID option. The default value is 1, which produces a solid line. If you use the LENDGRID= option, you do not need to specify the ENDGRID option. Refer to the description of the SYMBOL statement in *SAS/GRAPH: Reference* for more information about valid linetypes.

LGRID=*linetype*

specifies the line type for the grid requested with the GRID option. The default value is 1, which produces a solid line. If you use the LGRID= option, you do not need to specify the GRID option. Refer to the description of the SYMBOL statement in *SAS/GRAPH: Reference* for more information about valid linetypes.

LHREF=*linetype*

LH=*linetype*

specifies the line type for reference lines requested with the HREF= option. The default value is 2, which produces a dashed line. Refer to the description of the SYMBOL statement in *SAS/GRAPH: Reference* for more information about valid linetypes.

LOVERLAY=(*linetypes***)**

specifies line types for the line segments connecting points on overlay plots. Line types in the LOVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list.

LVREF=*linetype*

LV=*linetype*

specifies the line type for reference lines requested by the VREF= option. The default value is 2, which produces a dashed line. Refer to the description of the SYMBOL statement in *SAS/GRAPH: Reference* for more information about valid linetypes.

MAXPANELS=*n*

specifies the maximum number of panels used to display a box plot. By default, $n = 20$.

MISSBREAK

determines how groups are formed when observations are read from a DATA= data set and a character group variable is provided. When you specify the MISSBREAK option, observations with missing values of the group variable are not processed. Furthermore, the next observation with a nonmissing value of the group variable is treated as the beginning observation of a new group even if this value is identical to the most recent nonmissing group value. In other words, by specifying the option MISSBREAK and by inserting an observation with a missing group variable value into a group of consecutive observations with the same group variable value, you can split the group into two distinct groups of observations.

By default (that is, when you omit the MISSBREAK option), observations with missing values of the group variable are not processed, and all remaining observations with the same consecutive value of the group variable are treated as a single group.

NAME='*string*'

specifies a name for the box plot, not more than eight characters, that appears in the PROC GREPLAY master menu.

NLEGEND

requests a legend displaying group sizes. If the size is the same for each group, that number is displayed. Otherwise, the minimum and maximum group sizes are displayed.

NOBYREF

specifies that the reference line information in an HREF= or VREF= data set be applied uniformly to box plots created for all the BY groups in the input data set. If you specify the NOBYREF option, you do not need to provide BY variables in the reference line data set. By default, you must provide BY variables.

NOCHART

suppresses the creation of the box plot. You typically specify the NOCHART option when you are using the procedure to compute group summary statistics and save them in an output data set.

NOFRAME

suppresses the default frame drawn around the plot.

NOHLABEL

suppresses the label for the horizontal (group) axis. Use the NOHLABEL option when the meaning of the axis is evident from the tick mark labels, such as when a date format is associated with the group variable.

NOOVERLAYLEGEND

suppresses the legend for overlay plots that is displayed by default when the OVERLAY= option is specified.

NOSERIFS

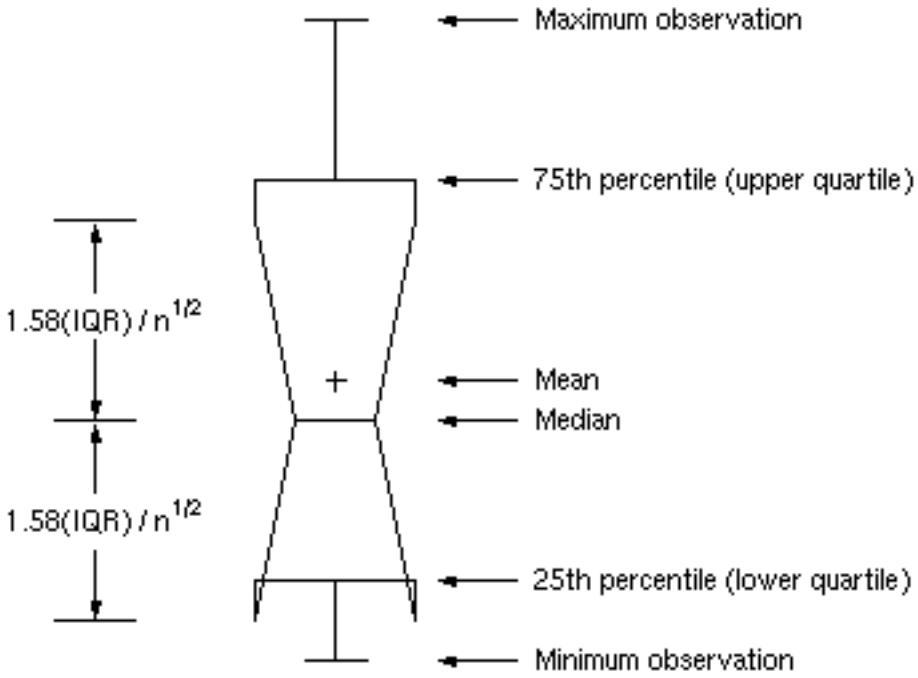
eliminates serifs from the whiskers of box-and-whiskers plots.

NOTCHES

specifies that box-and-whiskers plots be notched. The endpoints of the notches are located at the median plus and minus $1.58(\mathrm{IQR}/\sqrt{n})$, where IQR is the interquartile range and n is the group size. The medians (central lines) of two box-and-whiskers plots are significantly different at approximately the 0.95 confidence level if the corresponding notches do not overlap.

Refer to McGill, Tukey, and Larsen (1978) for more information. Figure 24.7 illustrates the NOTCHES option. Notice the folding effect at the bottom, which happens when the endpoint of a notch is beyond its corresponding quartile. This situation typically occurs when the group size is small.

Figure 24.7 Box Plot: The NOTCHES Option



NOTICKREP

applies to character-valued group variables and specifies that only the first occurrence of repeated, adjacent group values be labeled on the horizontal axis.

NOVANGLE

requests that the vertical axis label be strung out vertically.

NPANELPOS=*n*

NPANEL=*n*

specifies the number of group positions per panel. You typically specify the NPANELPOS= option to display more box-and-whiskers plots on a panel than the default number, which is $n = 25$.

You can specify a positive or negative number for n. The absolute value of n must be at least 5. If n is positive, the number of positions is adjusted so that it is approximately equal to n and so that all panels display approximately the same number of group positions. If n is negative, no balancing is done, and each panel (except possibly the last) displays approximately $|n|$ positions. In this case, the approximation is due only to axis scaling.

You can use the INTERVAL= option to change the effect of the NPANELPOS= option when a date or time format is associated with the group variable. The INTERVAL= option enables you to match the scale of the horizontal axis to the scale of the group variable without having to associate a different format with the group variable.

OUTBOX=*SAS-data-set*

creates an output data set that contains group summary statistics and outlier values for a box

plot. You can use an OUTBOX= data set as a BOX= input data set in a subsequent run of the procedure. See the section "OUTBOX= Data Set" on page 791 for details.

OUTHIGHHTML=*variable*
specifies a variable whose values are URLs to be associated with outlier points above the upper fence on a schematic box plot when graphics output is directed into HTML.

OUTHISTORY=*SAS-data-set*
creates an output data set that contains the group summary statistics. You can use an OUTHISTORY= data set as a HISTORY= input data set in a subsequent run of the procedure. See the section "OUTHISTORY= Data Set" on page 792 for details.

OUTLOWHTML=*variable*
specifies a variable whose values are URLs to be associated with outlier points below the lower fence on a schematic box plot when graphics output is directed into HTML.

OVERLAY=(*variable-list***)**
specifies variables to be plotted as overlays on the box plot. One value for each overlay variable is plotted at each group position. If there are multiple observations with the same group variable value in the input data set, the overlay variable values from the first observation in each group are plotted. By default, the points in an overlay plot are connected with line segments.

OVERLAYCLIPSYM=*symbol*
specifies the symbol used to plot clipped values on overlay plots when the CLIPFACTOR= option is used.

OVERLAYCLIPSYMHT=*value*
specifies the height for the symbol used to plot clipped values on overlay plots when the CLIPFACTOR= option is used.

OVERLAYHTML=(*variable-list***)**
specifies variables whose values are URLs to be associated with points on overlay plots when graphics output is directed into HTML. Variables in the OVERLAYHTML= list are matched with variables in the corresponding positions in the OVERLAY= list.

OVERLAYID=(*variable-list***)**
specifies variables whose formatted values are used to label points on overlays. Variables in the OVERLAYID= list are matched with variables in the corresponding positions in the OVERLAY= list. The value of the OVERLAYID= variable should be the same for each observation with a given value of the group variable.

OVERLAYLEGLAB='*label***'**
specifies the label displayed to the left of the overlay legend produced by the OVERLAY= option. The label can be up to 16 characters and must be enclosed in quotes. The default label is "Overlays:".

OVERLAYSYM=(*symbol-list***)**
specifies symbols used to plot overlay variables. Symbols in the OVERLAYSYM= list are matched with variables in the corresponding positions in the OVERLAY= list.

OVERLAYSYMHT=(*value-list*)

specifies the heights of symbols used to plot overlay variables. Symbol heights in the OVERLAYSYMHT= list are matched with variables in the corresponding positions in the OVERLAY= list.

PAGENUM=‘*string*’

specifies the form of the label used for pagination. The string can be up to 16 characters, and it must include one or two occurrences of the substitution character ‘#’. The first ‘#’ is replaced with the page number, and the optional second ‘#’ is replaced with the total number of pages.

The PAGENUM= option is useful when you are working with a large number of groups, resulting in multiple pages of output. For example, suppose that each of the following PLOT statements produces multiple pages:

```
proc boxplot data=Pistons;
   plot Diameter*Hour / pagenum='Page #';
   plot Diameter*Hour / pagenum='Page # of #';
   plot Diameter*Hour / pagenum='#/#';
run;
```

The third page produced by the first statement would be labeled “Page 3”. The third page produced by the second statement would be labeled “Page 3 of 5”. The third page produced by the third statement would be labeled “3/5”.

By default, no page number is displayed.

PAGENUMPOS=TL | TR | BL | BR | TL100 | TR100 | BL0 | BR0

specifies where to position the page number requested with the PAGENUM= option. The keywords TL, TR, BL, and BR correspond to the positions top left, top right, bottom left, and bottom right, respectively. You can use the TL100 and TR100 keywords to ensure that the page number appears at the very top of a page when a title is displayed. The BL0 and BR0 keywords ensure that the page number appears at the very bottom of a page when footnotes are displayed.

The default value is BR.

PCTLDEF=*index*

specifies one of five definitions used to calculate percentiles in the construction of box-and-whiskers plots. The index can be 1, 2, 3, 4, or 5. The five corresponding percentile definitions are discussed in the section “Percentile Definitions” on page 797. The default index is 5.

REPEAT

REP

specifies that the horizontal axis of a plot that spans multiple panels be arranged so that the last group position on a panel is repeated as the first group position on the next panel. The REPEAT option facilitates cutting and pasting panels together. When a SAS DATETIME format is associated with the group variable, the REPEAT option is the default.

SKIPHLABELS=*n*

SKIPHLABEL=*n*

specifies the number *n* of consecutive tick mark labels, beginning with the second tick mark label, that are thinned (not displayed) on the horizontal (group) axis. For example, specifying SKIPHLABEL=1 causes every other label to be skipped. Specifying SKIPHLABEL=2 causes the second and third labels to be skipped, the fifth and sixth labels to be skipped, and so forth.

The default value of the SKIPHLABELS= option is the smallest value *n* for which tick mark labels do not collide. A specified *n* will be overridden to avoid collision. To reduce thinning, you can use the TURNHLABELS option.

SYMBOLLEGEND=LEGEND*n*

SYMBOLLEGEND=NONE

controls the legend for the levels of a symbol variable (see Example 24.1). You can specify SYMBOLLEGEND=LEGEND*n*, where *n* is the number of a LEGEND statement defined previously. You can specify SYMBOLLEGEND=NONE to suppress the default legend. Refer to *SAS/GRAPH: Reference* for more information about the LEGEND statement.

SYMBOLORDER=DATA | INTERNAL | FORMATTED

SYMORD=DATA | INTERNAL | FORMATTED

specifies the order in which symbols are assigned for levels of the symbol variable. The DATA keyword assigns symbols to values in the order in which values appear in the input data set. The INTERNAL keyword assigns symbols based on sorted order of internal values of the symbol variable, and the FORMATTED keyword assigns them based on sorted formatted values. The default value is FORMATTED.

TOTPANELS=*n*

specifies the total number of panels to be used to display the plot. This option overrides the NPANEL= option.

TURNHLABELS

TURNHLABEL

turns the major tick mark labels for the horizontal (group) axis so that they are arranged vertically. By default, labels are arranged horizontally.

Note that arranging the labels vertically might leave insufficient vertical space on the panel for a plot.

VAXIS=*value-list*

VAXIS=AXIS*n*

specifies major tick mark values for the vertical axis of a box plot. The values must be listed in increasing order, must be evenly spaced, and must span the range of values displayed in the plot. You can specify the values with an explicit list or with an implicit list, as shown in the following example:

```
proc boxplot;
   plot Width*Hour / vaxis=0 2 4 6 8;
   plot Width*Hour / vaxis=0 to 8 by 2;
run;
```

You can also specify a previously defined AXIS statement with the VAXIS= option.

VFORMAT=*format*

specifies a format to be used for displaying tick mark labels on the vertical axis of the box plot.

VMINOR=*n*

VM=*n*

specifies the number of minor tick marks between major tick marks on the vertical axis. Minor tick marks are not labeled. By default, VMINOR=0.

VOFFSET=*value*

specifies the length in percent screen units of the offset at the ends of the vertical axis.

VREF=*value-list*

VREF=*SAS-data-set*

draws reference lines perpendicular to the vertical axis. You can use this option in the following ways:

- Specify the values for the lines with a VREF= list:

  ```
  vref=20
  vref=20 40 80
  ```

- Specify the values for the lines as the values of a numeric variable named _REF_ in a VREF= data set. Optionally, you can provide labels for the lines as values of a variable named _REFLAB_, which must be a character variable of up to 16 characters. If you want distinct reference lines to be displayed in plots for different analysis variables specified in the PLOT statement, you must include a character variable named _VAR_, whose values are the names of the analysis variables. If you do not include the variable _VAR_, all of the lines are displayed in all of the plots. Each observation in the VREF= data set corresponds to a reference line. If BY variables are used in the input data set, the same BY-variable structure must be used in the VREF= data set unless you specify the NOBYREF option.

VREFLABELS='*label1*' **...** '*labeln*'

VREFLABEL='*label1*' **...** '*labeln*'

VREFLAB='*label1*' **...** '*labeln*'

specifies labels for the reference lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

VREFLABPOS=*n*

specifies the horizontal position of the VREFLABEL= label, as described in the following table. By default, *n=1*.

n	Label Position
1	left-justified in plot area
2	right-justified in plot area
3	left-justified in right margin

VZERO
forces the origin to be included in the vertical axis for a box plot.

WAXIS=*n*
specifies the width in pixels for the axis and frame lines. By default, *n=1*.

WGRID=*n*
specifies the width in pixels for grid lines requested with the ENDGRID and GRID options. By default, *n=1*.

WOVERLAY=(*value-list***)**
specifies the widths in pixels for the line segments connecting points on overlay plots. Widths in the WOVERLAY= list are matched with variables in the corresponding positions in the OVERLAY= list. By default, all overlay widths are 1.

Details: BOXPLOT Procedure

Summary Statistics Represented by Box Plots

Table 24.6 lists the summary statistics represented in each box-and-whiskers plot.

Table 24.6 Summary Statistics Represented by Box Plots

Group Summary Statistic	Feature of Box-and-Whiskers Plot
maximum	endpoint of upper whisker
third quartile (75th percentile)	upper edge of box
median (50th percentile)	line inside box
mean	symbol marker
first quartile (25th percentile)	lower edge of box
minimum	endpoint of lower whisker

Note that you can request different box plot styles, as discussed in the section "Styles of Box Plots" on page 796, and as illustrated in Example 24.2.

Output Data Sets

OUTBOX= Data Set

The OUTBOX= data set saves group summary statistics and outlier values. The following variables can be saved:

- the group variable
- the variable _VAR_, containing the analysis variable name
- the variable _TYPE_, identifying features of box-and-whiskers plots
- the variable _VALUE_, containing values of box-and-whiskers plot features
- the variable _ID_, containing labels for outliers
- the variable _HTML_, containing URLs associated with plot features

_ID_ is included in the OUTBOX= data set only if the keyword SCHEMATICID or SCHEMATICIDFAR is specified with the BOXSTYLE= option. _HTML_ is present only if one or more of the HTML=, OUTHIGHHTML=, and OUTLOWHTML= options are specified.

Each observation in an OUTBOX= data set records the value of a single feature of one group's box-and-whiskers plot, such as its mean. The _TYPE_ variable identifies the feature whose value is recorded in _VALUE_. Table 24.7 lists valid _TYPE_ variable values.

Table 24.7 Valid _TYPE_ Values in an OUTBOX= Data Set

TYPE	Description
N	group size
MIN	minimum group value
Q1	group first quartile
MEDIAN	group median
MEAN	group mean
Q3	group third quartile
MAX	group maximum value
STDDEV	group standard deviation
LOW	low outlier value
HIGH	high outlier value
LOWHISKR	low whisker value, if different from MIN
HIWHISKR	high whisker value, if different from MAX
FARLOW	low far outlier value
FARHIGH	high far outlier value

Additionally, the following variables, if specified, are included:

- block variables
- symbol variable
- BY variables
- ID variables

OUTHISTORY= Data Set

The OUTHISTORY= data set saves group summary statistics. The following variables are saved:

- the group variable
- group minimum variables named by *analysis-variable* suffixed with *L*
- group first-quartile variables named by *analysis-variable* suffixed with *1*
- group mean variables named by *analysis-variable* suffixed with *X*
- group median variables named by *analysis-variable* suffixed with *M*
- group third-quartile variables named by *analysis-variable* suffixed with *3*
- group maximum variables named by *analysis-variable* suffixed with *H*
- group standard deviation variables named by *analysis-variable* suffixed with *S*
- group size variables named by *analysis-variable* suffixed with *N*

If an analysis variable name has the maximum length of 32 characters, PROC BOXPLOT forms summary statistic names from its first 16 characters, its last 15 characters, and the appropriate suffix.

Subgroup summary variables are created for each analysis variable specified in the PLOT statement. For example, consider the following statements:

```
proc boxplot data=Steel;
   plot (Width Diameter)*Lot / outhistory=Summary;
run;
```

The data set Summary contains variables named Lot, WidthL, Width1, WidthM, WidthX, Width3, WidthH, WidthS, WidthN, DiameterL, Diameter1, DiameterM, DiameterX, Diameter3, DiameterH, DiameterS, and DiameterN.

Additionally, the following variables, if specified, are included:

- BY variables
- block variables
- symbol variable
- ID variables

Note that an OUTHISTORY= data set does not contain outlier values, and therefore cannot be used, in general, to save a schematic box plot. You can use an OUTBOX= data set to save a schematic box plot summary.

Input Data Sets

DATA= Data Set

You can read analysis variable measurements from a data set specified with the DATA= option in the PROC BOXPLOT statement. Each analysis variable specified in the PLOT statement must be a SAS variable in the data set. This variable provides measurements that are organized into groups indexed by the group variable. The group variable, specified in the PLOT statement, must also be a SAS variable in the DATA= data set. Each observation in a DATA= data set must contain a value for each analysis variable and a value for the group variable. If the ith group contains n_i measurements, there should be n_i consecutive observations for which the value of the group variable is the index of the ith group. For example, if each group contains 20 items and there are 30 groups, the DATA= data set should contain 600 observations. Other variables that can be read from a DATA= data set include the following:

- block variables
- symbol variable
- BY variables
- ID variables

BOX= Data Set

You can read group summary statistics and outlier information from a BOX= data set specified in the PROC BOXPLOT statement. This enables you to reuse OUTBOX= data sets that have been created in previous runs of the BOXPLOT procedure to reproduce schematic box plots.

A BOX= data set must contain the following variables:

- the group variable
- _VAR_, containing the analysis variable name
- _TYPE_, identifying features of box-and-whiskers plots
- _VALUE_, containing values of those features

Each observation in a BOX= data set records the value of a single feature of one group's box-and-whiskers plot, such as its mean. Consequently, a BOX= data set contains multiple observations per group. These must appear consecutively in the BOX= data set.

The _TYPE_ variable identifies the feature whose value is recorded in a given observation. The following table lists valid _TYPE_ variable values.

Table 24.8 Valid _TYPE_ Values in a BOX= Data Set

TYPE	Description
N	group size
MIN	group minimum value
Q1	group first quartile
MEDIAN	group median
MEAN	group mean
Q3	group third quartile
MAX	group maximum value
STDDEV	group standard deviation
LOW	low outlier value
HIGH	high outlier value
LOWHISKR	low whisker value, if different from MIN
HIWHISKR	high whisker value, if different from MAX
FARLOW	low far outlier value
FARHIGH	high far outlier value

The features identified by _TYPE_ values N, MIN, Q1, MEDIAN, MEAN, Q3, and MAX are required for each group.

Other variables that can be read from a BOX= data set include the following:

- the variable _ID_, containing labels for outliers
- the variable _HTML_, containing URLs to be associated with features on box plots
- block variables
- symbol variable
- BY variables
- ID variables

When you specify the keyword SCHEMATICID or SCHEMATICIDFAR with the BOXSTYLE= option, values of _ID_ are used as outlier labels. If _ID_ does not exist in the BOX= data set, the values of the first variable listed in the ID statement are used.

HISTORY= Data Set

You can read group summary statistics from a HISTORY= data set specified in the PROC BOXPLOT statement. This enables you to reuse OUTHISTORY= data sets that have been created in previous runs of the BOXPLOT procedure or to read output data sets created with SAS summarization procedures, such as PROC UNIVARIATE.

Note that a HISTORY= data set does *not* contain outlier information. Therefore, in general you cannot reproduce a schematic box plot from summary statistics saved in an OUTHISTORY= data set. To save and reproduce schematic box plots, use OUTBOX= and BOX= data sets.

A HISTORY= data set must contain the following:

- the group variable
- a group minimum variable for each analysis variable
- a group first-quartile variable for each analysis variable
- a group median variable for each analysis variable
- a group mean variable for each analysis variable
- a group third-quartile variable for each analysis variable
- a group maximum variable for each analysis variable
- a group standard deviation variable for each analysis variable
- a group size variable for each analysis variable

The names of the group summary statistics variables must be the analysis variable name concatenated with the following special suffix characters.

Group Summary Statistic	Suffix Character
group minimum	L
group first quartile	1
group median	M
group mean	X
group third quartile	3
group maximum	H
group standard deviation	S
group size	N

For example, consider the following statements:

```
proc boxplot history=Summary;
   plot (Weight Yieldstrength) * Batch;
run;
```

The data set Summary must include the variables Batch, WeightL, Weight1, WeightM, WeightX, Weight3, WeightH, WeightS, WeightN, YieldstrengthL, Yieldstrength1, YieldstrengthM, YieldstrengthX, Yieldstrength3, YieldstrengthH, YieldstrengthS, and YieldstrengthN.

Note that if you specify an analysis variable whose name contains the maximum of 32 characters, the summary variable names must be formed from the first 16 characters and the last 15 characters of the analysis variable name, suffixed with the appropriate character.

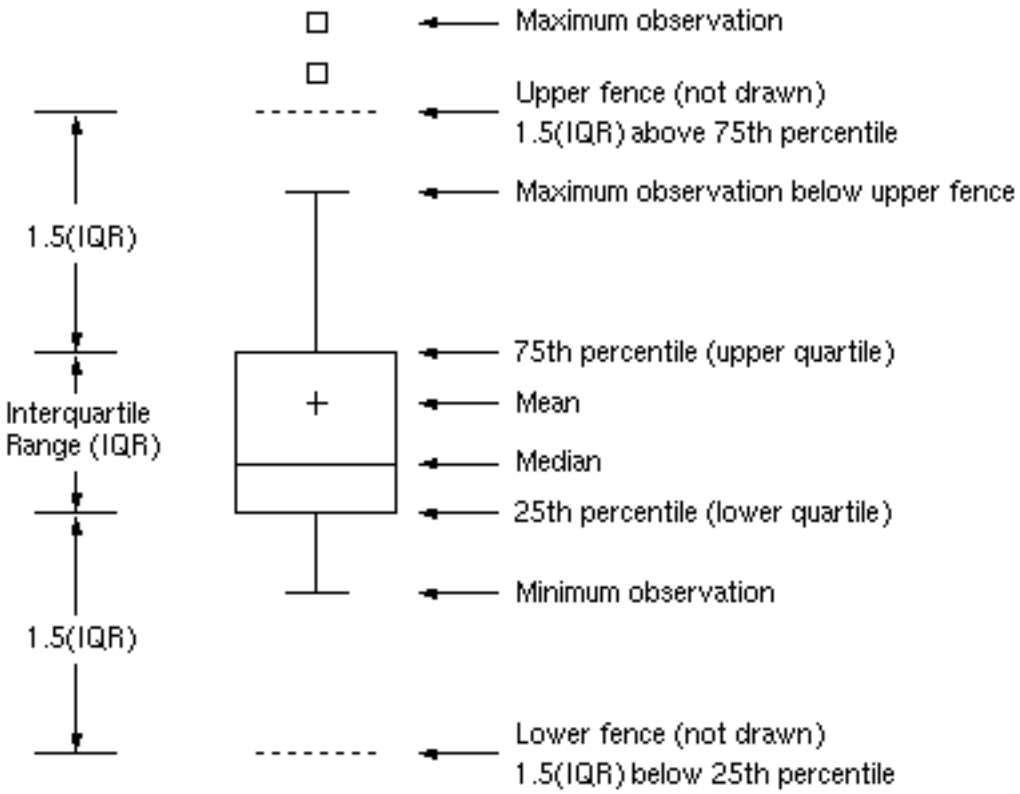
These other variables can be read from a HISTORY= data set:

- block variables
- symbol variable
- BY variables
- ID variables

Styles of Box Plots

A box-and-whiskers plot is displayed for the measurements in each group on the box plot. The *skeletal* style of the box-and-whiskers plot shown in Figure 24.6 is the default. You can produce a *schematic* box plot by specifying the BOXSTYLE=SCHEMATIC option in the PLOT statement. Figure 24.8 illustrates a typical schematic box plot and the locations of the fences (which are not displayed in actual output). See the description of the BOXSTYLE= option on page 773 for complete details.

Figure 24.8 Schematic Box-and-Whiskers Plot



You can draw connecting lines between adjacent box-and-whiskers plots by using the BOXCONNECT=*keyword* option. For example, BOXCONNECT=MEAN connects the points representing the means of adjacent groups. Other available keywords are MIN, Q1, MEDIAN, Q3, and

MAX. Specifying BOXCONNECT without a keyword is equivalent to specifying BOXCONNECT=MEAN. You can specify the color for the connecting lines with the CCONNECT= option.

Percentile Definitions

You can use the PCTLDEF= option to specify one of five definitions for computing quantile statistics (percentiles). Suppose that n is the number of nonmissing values for a variable and that $x_1, x_2, \ldots, x_n$ represent the ordered values of the analysis variable. For the tth percentile, set $p = t/100$.

For the following definitions numbered 1, 2, 3, and 5, express np as

$$np = j + g$$

where j is the integer part of np, and g is the fractional part of np. For definition 4, let

$$(n+1)p = j + g$$

The tth percentile (call it y) can be defined as follows:

PCTLDEF=1 weighted average at x_{np}

$$y = (1-g)x_j + gx_{j+1}$$

where x_0 is taken to be x_1.

PCTLDEF=2 observation numbered closest to np

$$y = x_i$$

where i is the integer part of $np + 1/2$ if $g \neq 1/2$. If $g = 1/2$, then $y = x_j$ if j is even, or $y = x_{j+1}$ if j is odd.

PCTLDEF=3 empirical distribution function

$$y = x_j \text{ if } g = 0$$

$$y = x_{j+1} \text{ if } g > 0$$

PCTLDEF=4 weighted average aimed at $x_{p(n+1)}$

$$y = (1-g)x_j + gx_{j+1}$$

where x_{n+1} is taken to be x_n.

PCTLDEF=5 empirical distribution function with averaging

$$y = (x_j + x_{j+1})/2 \text{ if } g = 0$$

$$y = x_{j+1} \text{ if } g > 0$$

Missing Values

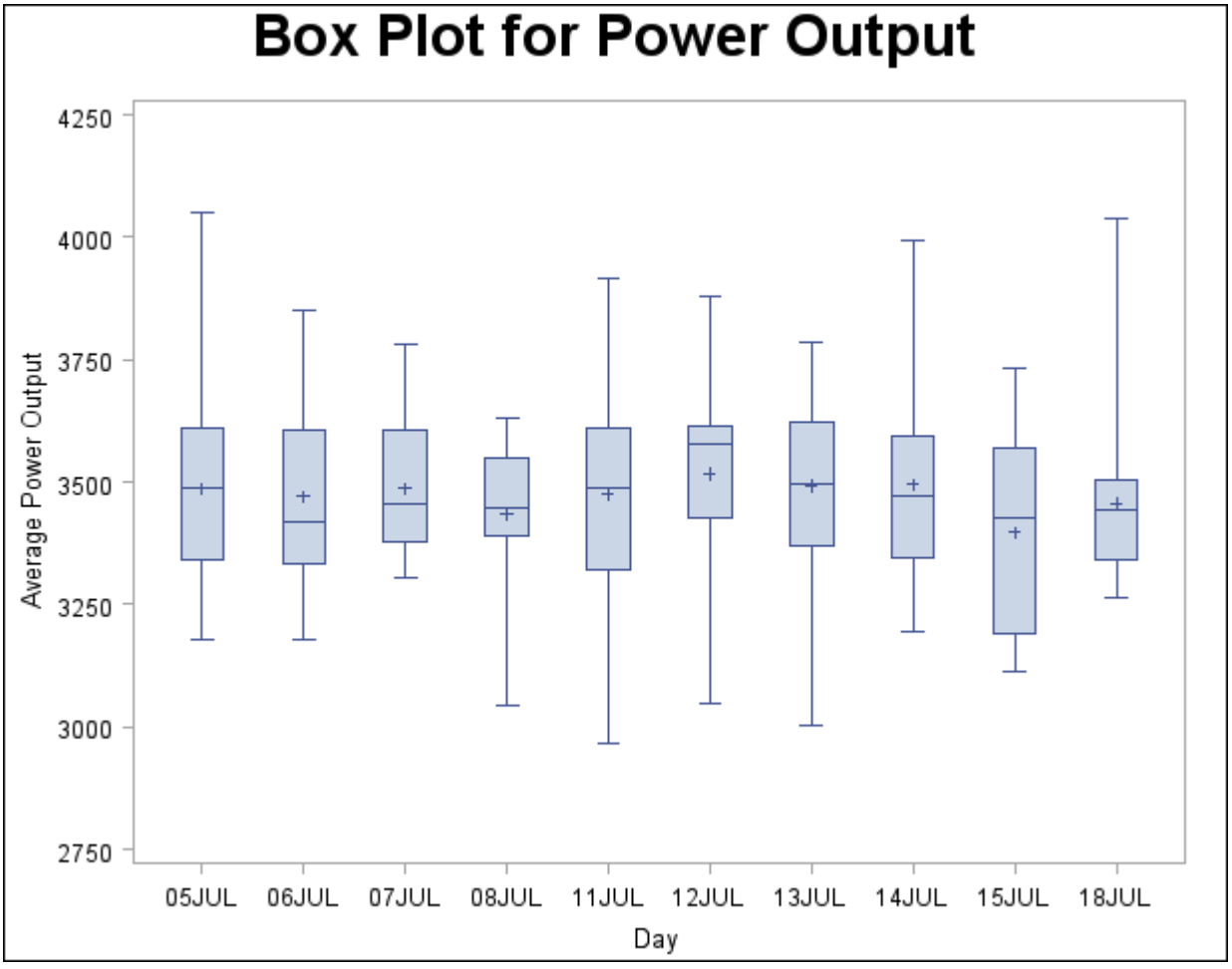
An observation read from an input data set is not analyzed if the value of the group variable is missing. For a particular analysis variable, an observation read from a DATA= data set is not analyzed if the value of the analysis variable is missing.

Continuous Group Variables

By default, the PLOT statement treats numerical group variable values as *discrete* values and spaces the boxes evenly on the plot. The following statements produce the box plot in Figure 24.9:

```
title 'Box Plot for Power Output';
proc boxplot data=Turbine;
   plot KWatts*Day;
run;
```

Figure 24.9 Box Plot with Discrete Group Variable



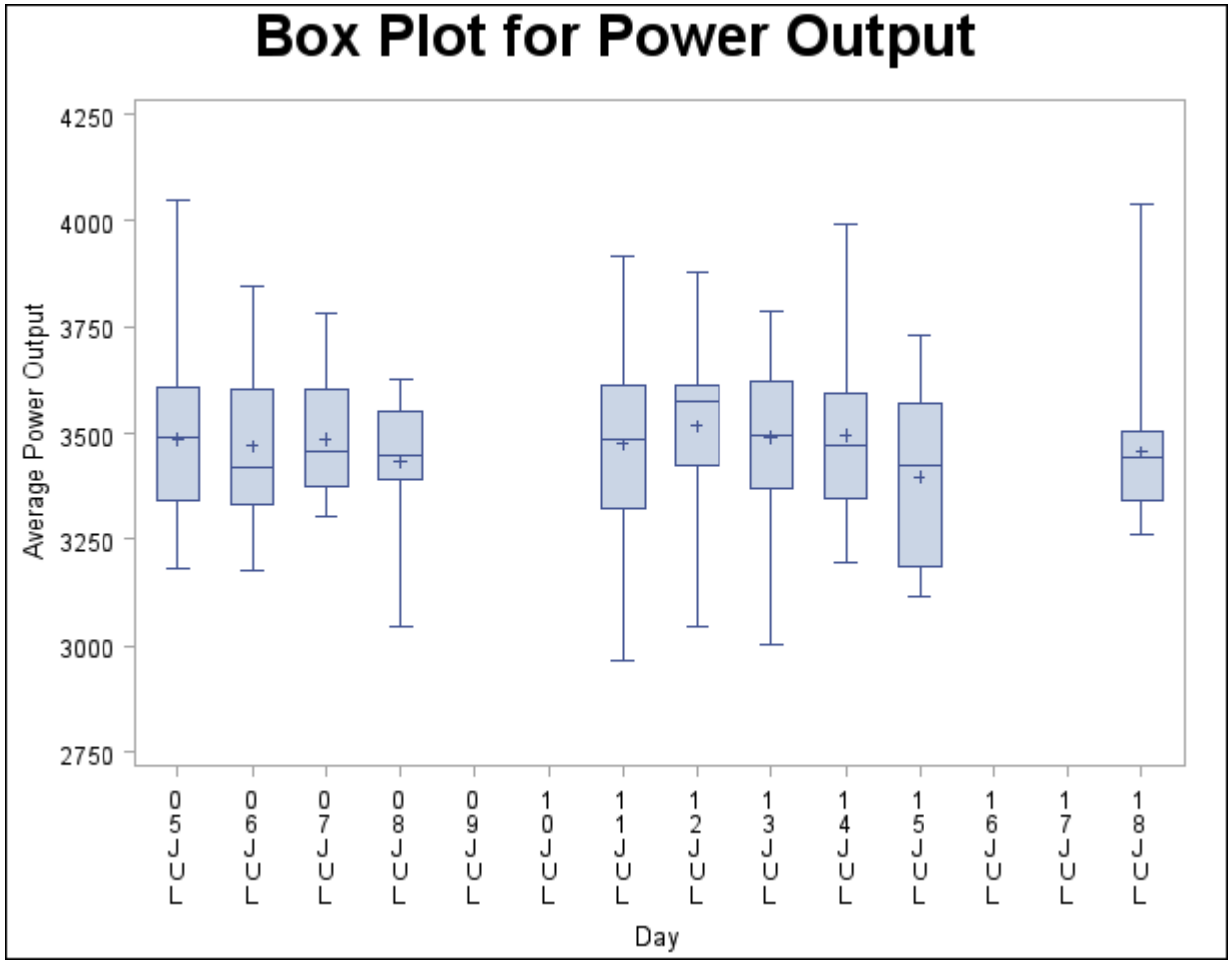
The labels on the horizontal axis in Figure 24.9 do not represent 10 consecutive days, but the box-and-whiskers plots are evenly spaced.

In order to treat the group variable as *continuous*, you can specify the CONTINUOUS or HAXIS= option when producing traditional graphics. Either option produces a box plot with a horizontal axis scaled for continuous group variable values. (ODS Graphics does not support a continuous group axis.)

The following statements produce the plot shown in Figure 24.10. The TURNHLABEL option orients the horizontal axis labels vertically so there is room to display them all.

```
title 'Box Plot for Power Output';
proc boxplot data=Turbine;
   plot KWatts*Day / turnhlabel
                     continuous;
run;
```

Figure 24.10 Box Plot with Continuous Group Variable



Note that the tick values on the horizontal axis represent consecutive days and that no box-and-whiskers plots are displayed for days when no turbine data were collected.

Positioning Insets

This section provides details on three different methods of positioning INSET boxes by using the POSITION= option. With the POSITION= option, you can specify the following:

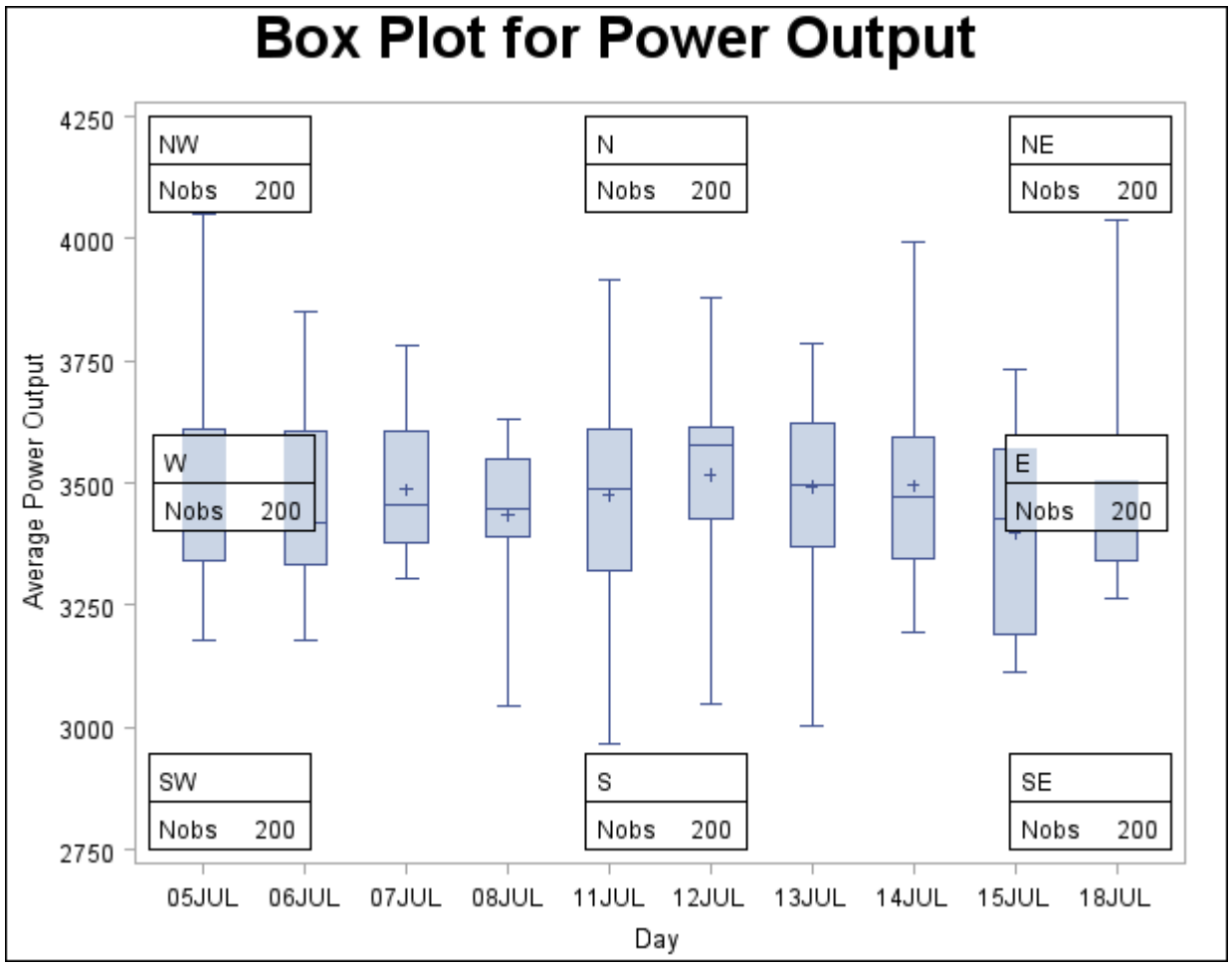
- compass points
- keywords for margin positions
- coordinates in data units or percent axis units

Positioning the Inset Using Compass Points

You can specify the eight compass points (N, NE, E, SE, S, SW, W, and NW) as keywords for the POSITION= option. The default inset position is NW. The following statements create the display in Figure 24.11, which illustrates all eight compass positions:

```
title 'Box Plot for Power Output';
proc boxplot data=Turbine;
   plot KWatts*Day;
   inset nobs / height=2.5 cfill=blank header='NW' pos=nw;
   inset nobs / height=2.5 cfill=blank header='N ' pos=n ;
   inset nobs / height=2.5 cfill=blank header='NE' pos=ne;
   inset nobs / height=2.5 cfill=blank header='E ' pos=e ;
   inset nobs / height=2.5 cfill=blank header='SE' pos=se;
   inset nobs / height=2.5 cfill=blank header='S ' pos=s ;
   inset nobs / height=2.5 cfill=blank header='SW' pos=sw;
   inset nobs / height=2.5 cfill=blank header='W ' pos=w ;
run;
```

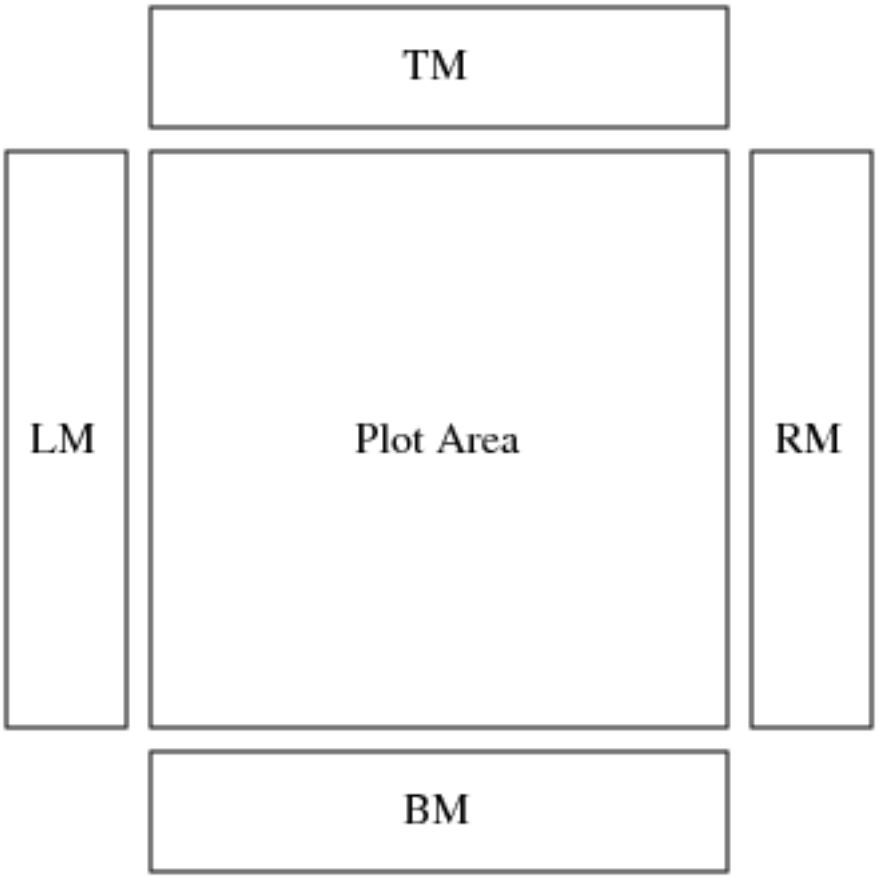
Figure 24.11 Insets Positioned Using Compass Points



Positioning the Inset in the Margins

You can also use the INSET statement to position an inset in one of the four margins surrounding the plot area by using the margin keyword LM, RM, TM, or BM, as illustrated in Figure 24.12.

Figure 24.12 Positioning Insets in the Margins



For an example of an inset placed in the top margin, see Output 24.1.1. Margin positions are recommended for insets containing a large number of statistics. If you attempt to display a lengthy inset in the interior of the plot, it is likely that the inset will collide with the data display.

Positioning the Inset Using Coordinates

You can also specify the position of an inset with coordinates by using the POSITION= (x, y) option. You can specify coordinates in axis percent units (the default) or in axis data units.

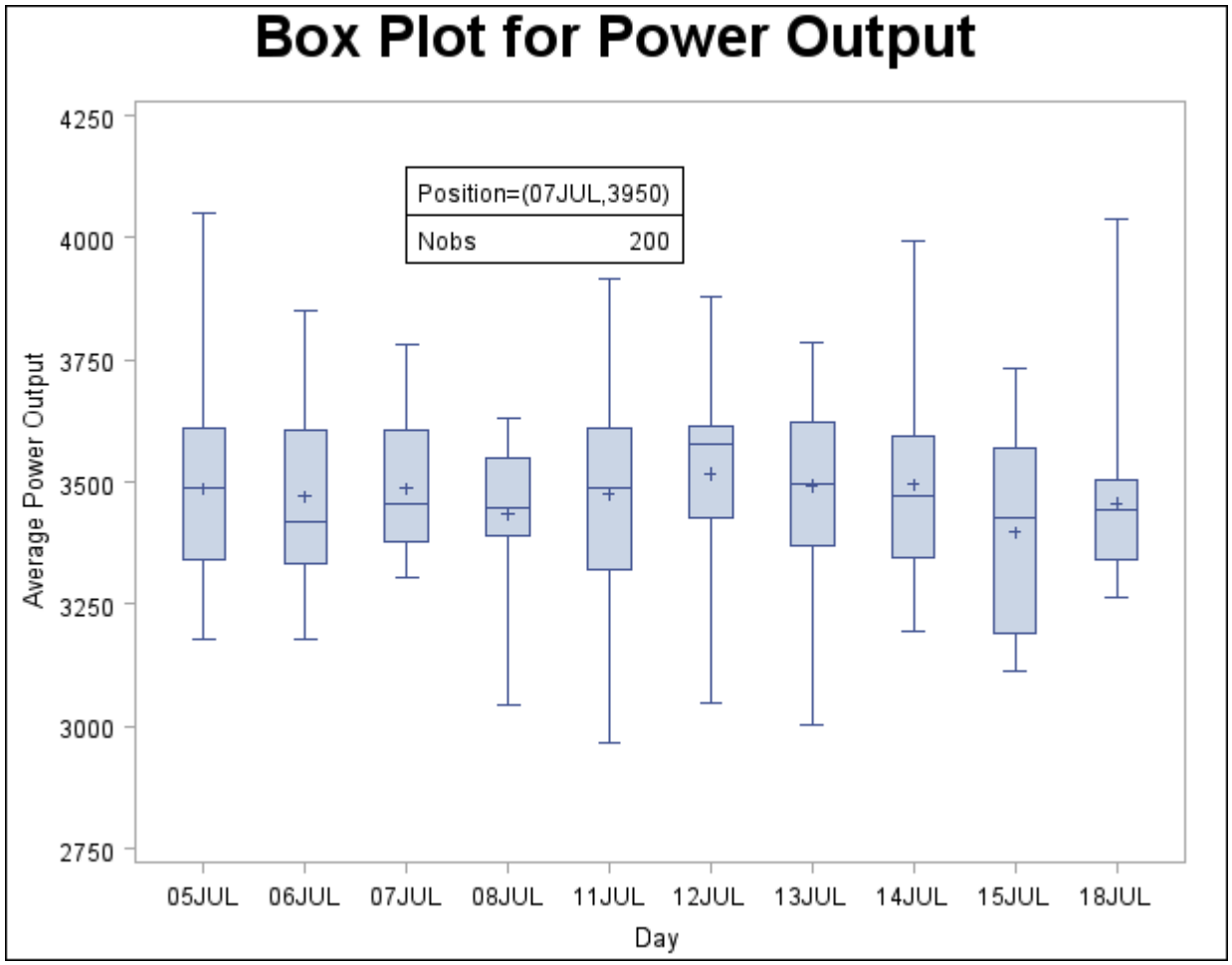
Data Unit Coordinates

If you specify the DATA option immediately following the coordinates, the inset is positioned using axis data units. For example, the following statements place the bottom-left corner of the inset at 07JUL on the horizontal axis and 3950 on the vertical axis:

```
title 'Box Plot for Power Output';
proc boxplot data=Turbine;
   plot KWatts*Day;
   inset nobs /
      header   = 'Position=(07JUL,3950)'
      position = ('07JUL94'd, 3950) data;
run;
```

The box plot is displayed in Figure 24.13. By default, the specified coordinates determine the position of the bottom-left corner of the inset. You can change this reference point with the REFPOINT= option, as in the next example.

Figure 24.13 Inset Positioned Using Data Unit Coordinates



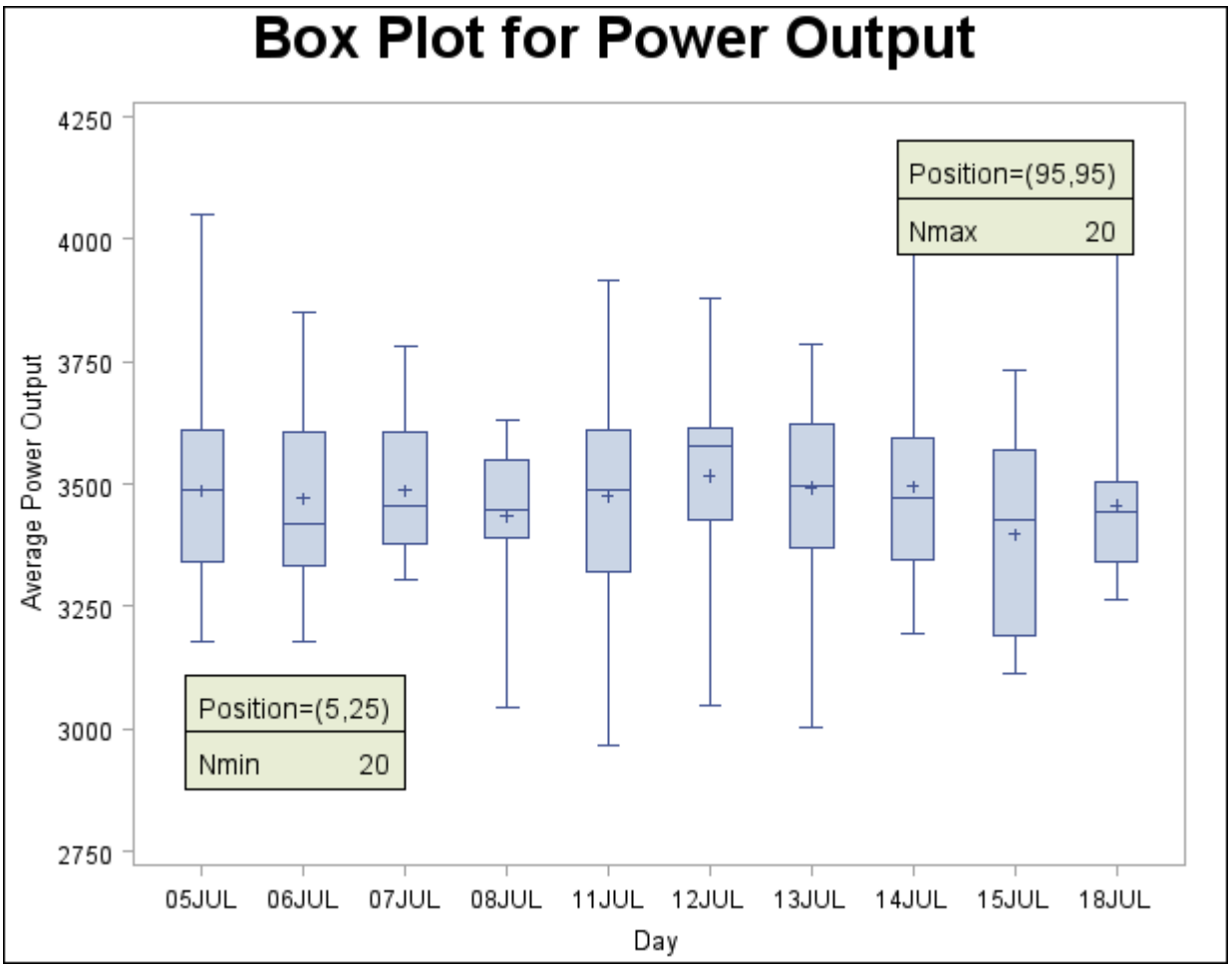
Axis Percent Unit Coordinates

If you do not use the DATA option, the inset is positioned using axis percent units. The coordinates of the bottom-left corner of the display are $(0, 0)$, while the coordinates of the top-right corner are $(100, 100)$. For example, the following statements create a box plot with two insets, both positioned using coordinates in axis percent units:

```
title 'Box Plot for Power Output';
proc boxplot data=Turbine;
   plot KWatts*Day;
   inset nmin / position = (5,25)
                header   = 'Position=(5,25)'
                height   = 3
                cfill    = ywh
                refpoint = tl;
   inset nmax / position = (95,95)
                header   = 'Position=(95,95)'
                height   = 3
                cfill    = ywh
                refpoint = tr;
run;
```

The display is shown in Figure 24.14. Notice that the REFPOINT= option is used to determine which corner of the inset is placed at the coordinates specified with the POSITION= option. The first inset has REFPOINT=TL, so the top-left corner of the inset is positioned 5% of the way across the horizontal axis and 25% of the way up the vertical axis. The second inset has REFPOINT=TR, so the top-right corner of the inset is positioned 95% of the way across the horizontal axis and 95% of the way up the vertical axis. Note also that coordinates in axis percent units must be *between* 0 and 100.

Figure 24.14 Inset Positioned Using Axis Percent Unit Coordinates



Displaying Blocks of Data

To display data organized in blocks of consecutive observations, specify one or more *block variables* in parentheses after the group variable in the PLOT statement. The block variables must be variables in the input data set. The BOXPLOT procedure displays a legend identifying blocks of consecutive observations with identical values of the block variables. The legend displays one track of values for each block variable containing formatted values of the block variable.

The values of a block variable must be the same for all observations with the same value of the group variable. In other words, groups must be nested within blocks determined by block variables.

The following statements create a SAS data set containing diameter measurements for a part produced on three different machines:

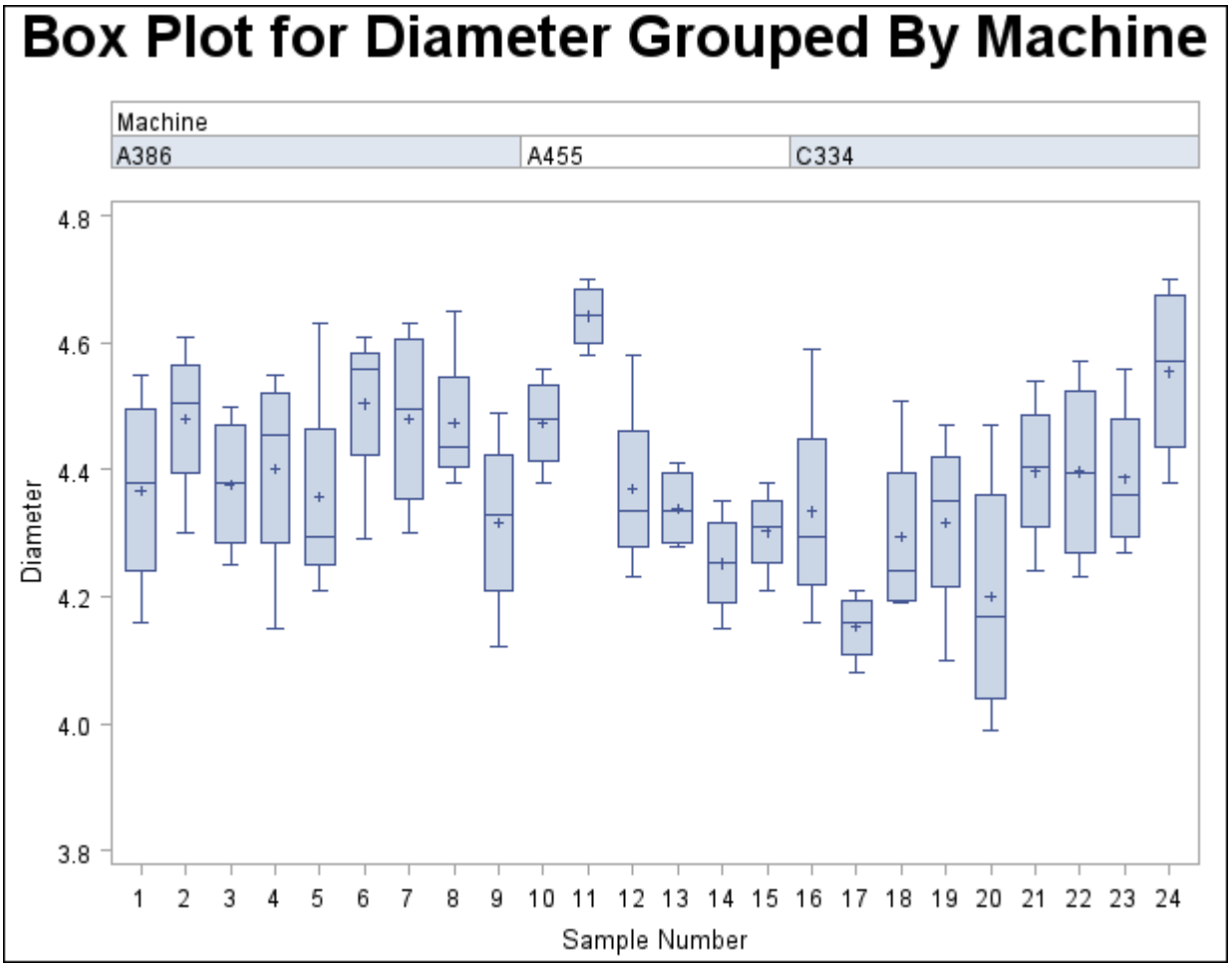
```
data Parts;
   length Machine $ 4;
   input Sample Machine $ @;
   do i= 1 to 4;
      input Diam @;
      output;
   end;
   drop i;
 datalines;
 1  A386  4.32 4.55 4.16 4.44
 2  A386  4.49 4.30 4.52 4.61
 3  A386  4.44 4.32 4.25 4.50
 4  A386  4.55 4.15 4.42 4.49
 5  A386  4.21 4.30 4.29 4.63
 6  A386  4.56 4.61 4.29 4.56
 7  A386  4.63 4.30 4.41 4.58
 8  A386  4.38 4.65 4.43 4.44
 9  A386  4.12 4.49 4.30 4.36
10  A455  4.45 4.56 4.38 4.51
11  A455  4.62 4.67 4.70 4.58
12  A455  4.33 4.23 4.34 4.58
13  A455  4.29 4.38 4.28 4.41
14  A455  4.15 4.35 4.28 4.23
15  A455  4.21 4.30 4.32 4.38
16  C334  4.16 4.28 4.31 4.59
17  C334  4.14 4.18 4.08 4.21
18  C334  4.51 4.20 4.28 4.19
19  C334  4.10 4.33 4.37 4.47
20  C334  3.99 4.09 4.47 4.25
21  C334  4.24 4.54 4.43 4.38
22  C334  4.23 4.48 4.31 4.57
23  C334  4.27 4.40 4.32 4.56
24  C334  4.70 4.65 4.49 4.38
;
```

The following statements create a box plot for the measurements in the Parts data set grouped into blocks by the block variable Machine:

```
title 'Box Plot for Diameter Grouped By Machine';
proc boxplot data=Parts;
   plot Diam*Sample (Machine);
   label Sample  = 'Sample Number'
         Machine = 'Machine'
         Diam    = 'Diameter';
run;
```

Note the LABEL statement used to provide labels for the axes and for the block legend. The plot is shown in Figure 24.15.

Figure 24.15 Box Plot Using a Block Variable



The unique consecutive values of Machine (A386, A455, and C334) are displayed in a legend above the plot. That is the default location of the block legend. You can control the position of the block legend with the BLOCKPOS= option. See the BLOCKPOS= option on page 772 for details.

By default, block variable values that are too long to fit into the available space in a block legend are not displayed. You can specify the BLOCKLABTYPE= option to display lengthy labels. Specify BLOCKLABTYPE=SCALED to scale down the text size of the values so they all fit. Use BLOCKLABTYPE=TRUNCATED to truncate lengthy values. You can also use BLOCKLABTYPE=*height*

to specify a text height in vertical percent screen units for the values.

You can control the position of legend labels with the BLOCKLABELPOS= option. Valid BLOCKLABELPOS= values are ABOVE (the default, as shown in Figure 24.15) and LEFT.

Clipping Extreme Values

By default a box plot's vertical axis is scaled to accommodate all the values in all groups. If the variation between groups is large with respect to the variation within groups, or if some groups contain extreme outlier values, the vertical axis scale can become so large that the box-and-whiskers plots are compressed. In such cases, you can clip the extreme values to produce a more readable plot, as illustrated in the following example.

A company produces copper tubing. The diameter measurements (in millimeters) for 15 batches of five tubes each are provided in the data set Newtubes:

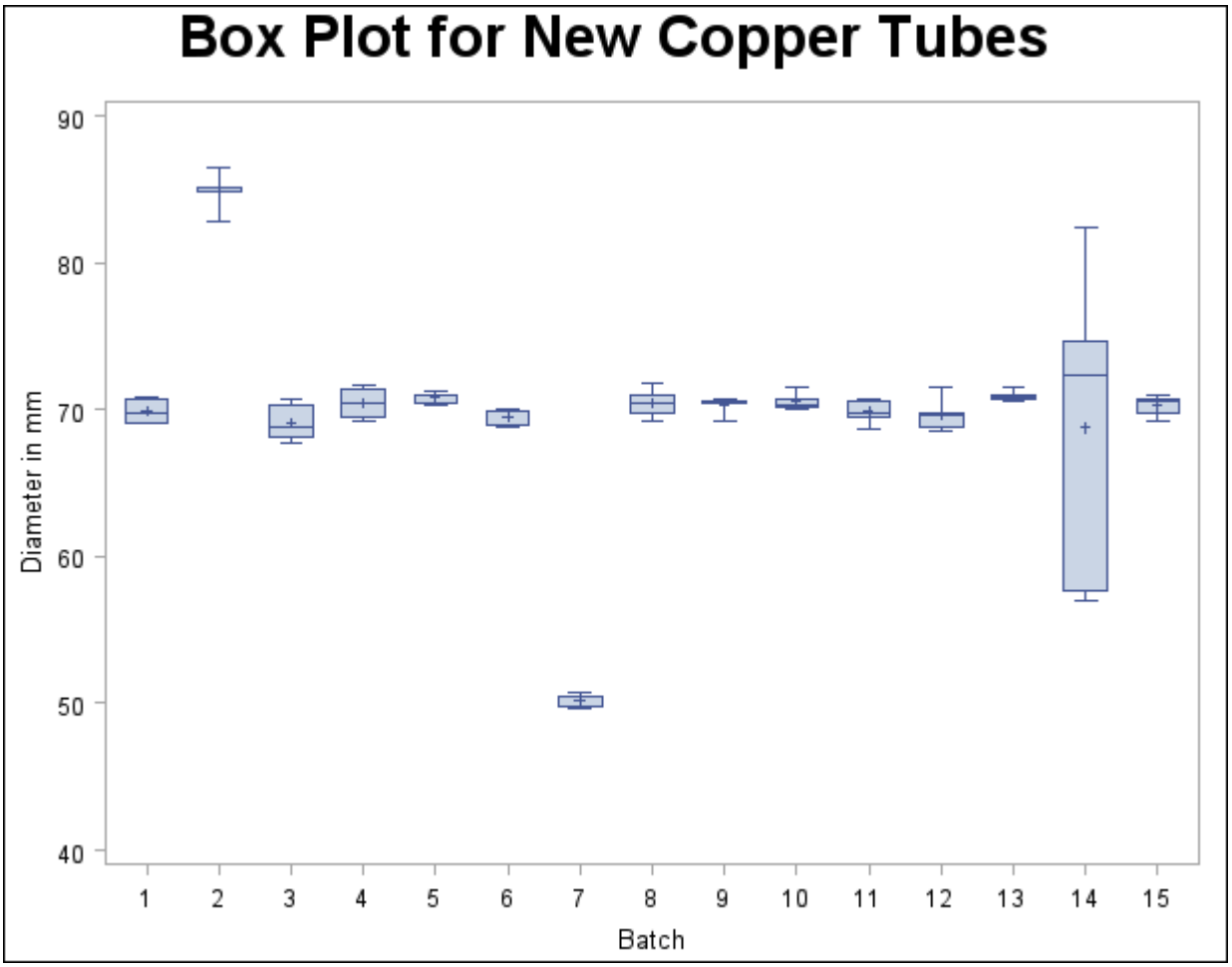
```
data Newtubes;
   label Diameter='Diameter in mm';
   do Batch = 1 to 15;
      do  i = 1 to 5;
         input Diameter @@;
         output;
      end;
   end;
datalines;
69.13  69.83  70.76  69.13  70.81
85.06  82.82  84.79  84.89  86.53
67.67  70.37  68.80  70.65  68.20
71.71  70.46  71.43  69.53  69.28
71.04  71.04  70.29  70.51  71.29
69.01  68.87  69.87  70.05  69.85
50.72  50.49  49.78  50.49  49.69
69.28  71.80  69.80  70.99  70.50
70.76  69.19  70.51  70.59  70.40
70.16  70.07  71.52  70.72  70.31
68.67  70.54  69.50  69.79  70.76
68.78  68.55  69.72  69.62  71.53
70.61  70.75  70.90  71.01  71.53
74.62  56.95  72.29  82.41  57.64
70.54  69.82  70.71  71.05  69.24
;
run;
```

The following statements create a box plot of the tube diameters:

```
title  'Box Plot for New Copper Tubes' ;
proc boxplot data=Newtubes;
   plot Diameter*Batch;
run;
```

The box plot is shown in Figure 24.16.

Figure 24.16 Compressed Box Plots



Note that the diameters in batch 2 are significantly larger, and those in batch 7 significantly smaller, than those in most of the other batches. The default vertical axis scaling causes the box-and-whiskers plots to be compressed.

You can produce a more useful box plot by specifying the CLIPFACTOR=*factor* option, where *factor* is a value greater than one. Clipping is applied as follows:

1. The mean of the first quartile values ($\overline{Q1}$) and the mean of the third quartile values ($\overline{Q3}$) are computed across all groups.
2. The following values define the clipping range:

   $$y_{\max} = \overline{Q1} + (\overline{Q3} - \overline{Q1}) \times factor$$

   and

   $$y_{\min} = \overline{Q3} - (\overline{Q3} - \overline{Q1}) \times factor$$

   Any statistic greater than $y_{\max}$ or less than $y_{\min}$ is ignored during vertical axis scaling.

NOTE:

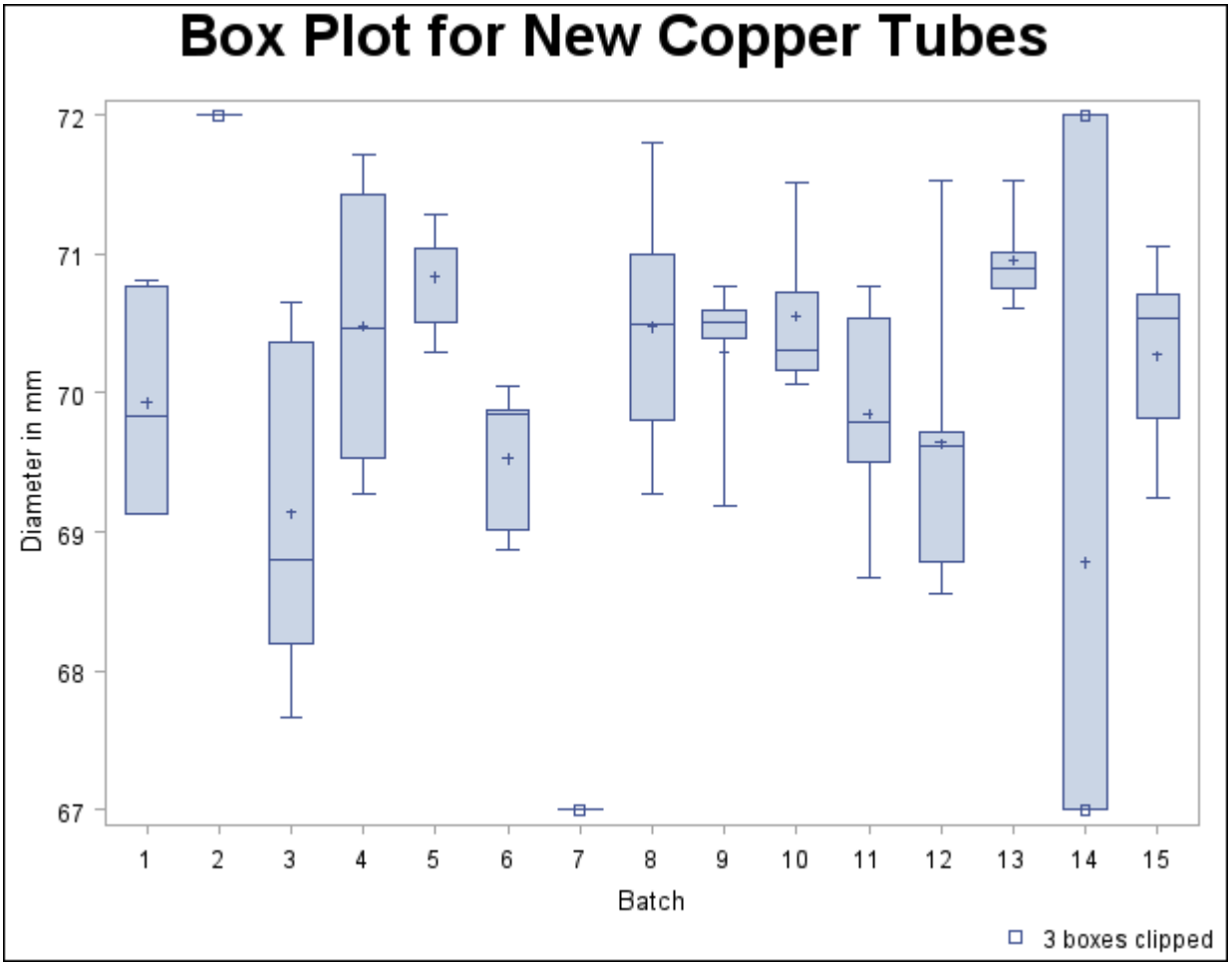
- Clipping is applied only to the plotted statistics and not to the statistics saved in an output data set.
- A special symbol is used for clipped points (the default symbol is a square), and a legend is added to the chart indicating the number of boxes that were clipped.

The following statements use a clipping factor of 1.5 to create a box plot of the same data plotted in Figure 24.16:

```
title  'Box Plot for New Copper Tubes' ;
proc boxplot data=Newtubes;
   plot Diameter*Batch /
      clipfactor = 1.5;
run;
```

The clipped box plot is shown in Figure 24.17.

Figure 24.17 Box Plot with Clip Factor of 1.5



In Figure 24.17 the extreme values are clipped, making the box plot more readable. The box-and-whiskers plots for batches 2 and 7 are clipped completely, while the plot for batch 14 is clipped at both the top and bottom. Clipped points are marked with a square, and a clipping legend is added at the lower right of the display.

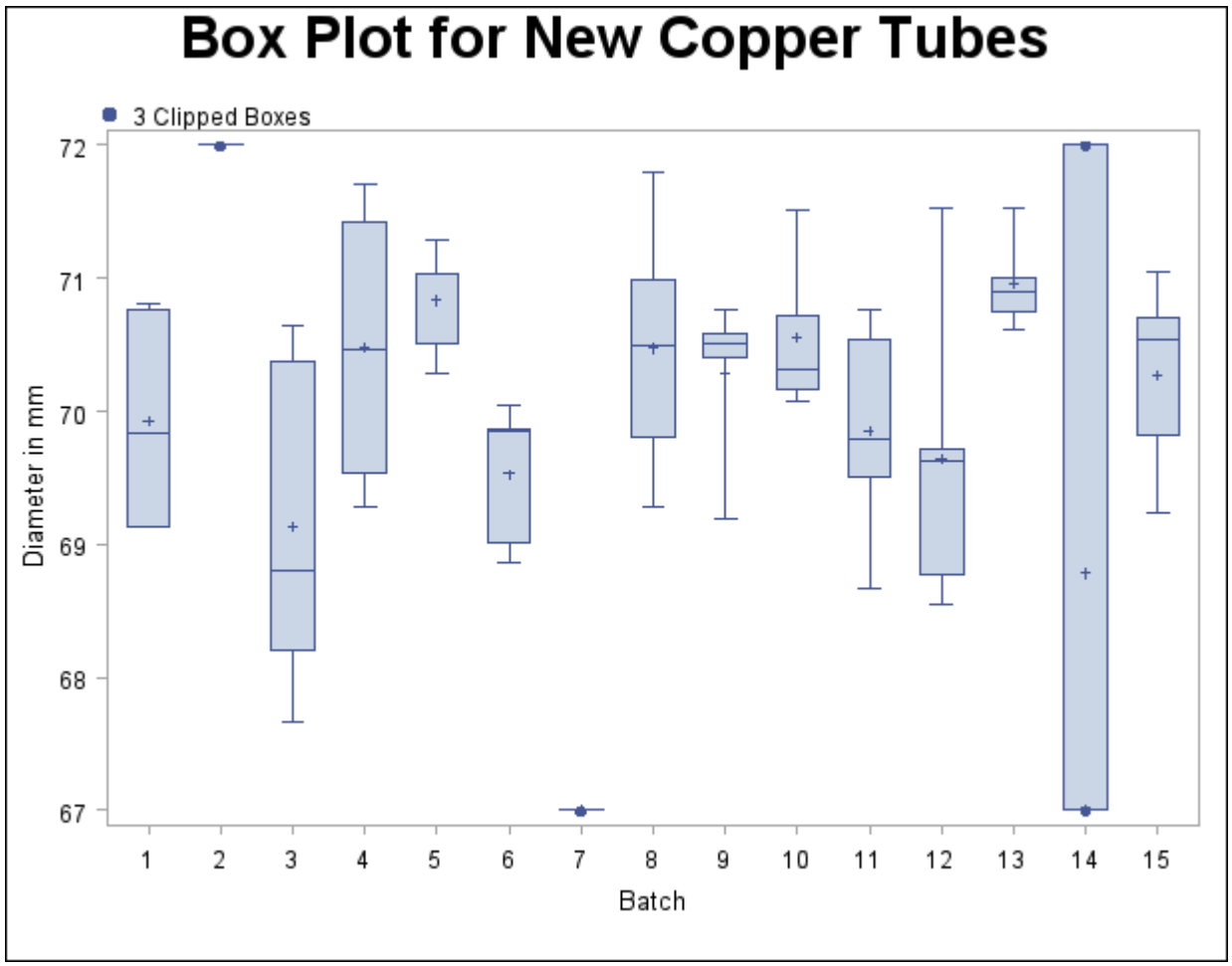
Other clipping options are available, as illustrated by the following statements:

```
title  'Box Plot for New Copper Tubes' ;
proc boxplot data=Newtubes;
   plot Diameter*Batch /
      clipfactor  = 1.5
      clipsymbol  = dot
      cliplegpos  = top
      cliplegend  = '# Clipped Boxes'
      clipsubchar = '#';
run;
```

Specifying CLIPSYMBOL=DOT marks the clipped points with a dot instead of the default square. Specifying CLIPLEGPOS=TOP positions the clipping legend at the top of the chart. The options CLIPLEGEND='# Clipped Boxes' and CLIPSUBCHAR='#' request the clipping legend "3 Clipped Boxes".

Figure 24.18 shows the modified box plot.

Figure 24.18 Box Plot Using Clipping Options



For more information about clipping options, see the appropriate entries in the section "PLOT Statement Options" on page 767.

ODS Graphics (Experimental)

The BOXPLOT procedure supports ODS Graphics on an experimental basis in SAS 9.2. To use ODS Graphics to produce box plots you must specify the ODS GRAPHICS statement prior to the PROC BOXPLOT statement. For more information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

The appearance of a box plot produced using ODS Graphics is determined by the style associated with the ODS destination where the graph is produced. PLOT statement options used to control the appearance of traditional high-resolution graphs are ignored for ODS Graphics output.

When producing ODS graphical displays, the PLOT statement assigns a name to each graph it creates. You can use this name to reference the graph when using ODS. The name is listed in Table 24.9.

Table 24.9 ODS Graphics Produced by PROC BOXPLOT

ODS Graph Name	Plot Description
Boxplot	box-and-whiskers plots for groups

Examples: BOXPLOT Procedure

This section provides advanced examples of the PLOT statement.

Example 24.1: Displaying Summary Statistics in a Box Plot

This example demonstrates how you can use the INSET and INSETGROUP statements to include tables of summary statistics in your box plots. The following statements produce a box plot of the Turbine data set from the section "Getting Started: BOXPLOT Procedure" on page 751, augmented with insets containing summary statistics:

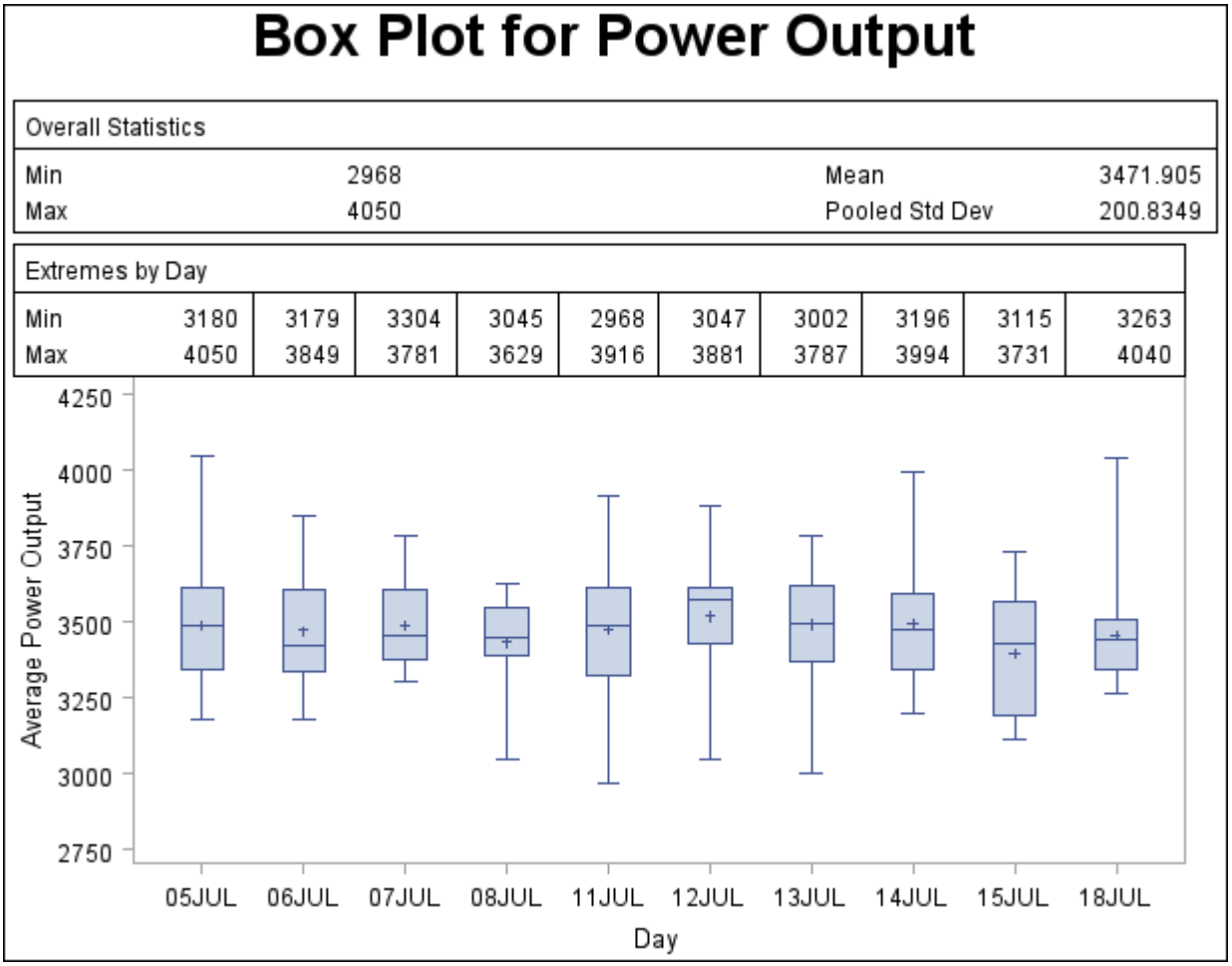
```
title 'Box Plot for Power Output';
proc boxplot data=Turbine;
   plot KWatts*Day;
   inset min mean max stddev /
      header = 'Overall Statistics'
      pos    = tm;
   insetgroup min max /
      header = 'Extremes by Day';
run;
```

The INSET statement produces an inset of overall summary statistics. The keywords listed before the slash (/) request the minimum, mean, maximum, and standard deviation computed over all days. The POS=TM option places the inset in the top margin of the plot.

The INSETGROUP statement produces an inset containing statistics calculated for each group separately. The MIN and MAX keywords request the minimum and maximum observations from each day, respectively.

The resulting plot is shown in Output 24.1.1.

Output 24.1.1 Box Plot with Insets



Example 24.2: Using Box Plots to Compare Groups

In this example a box plot is used to compare the delay times of airline flights during the Christmas holidays with the delay times prior to the holiday period. The following statements create a data set named Times with the delay times in minutes for 25 flights each day. When a flight is canceled, the delay is recorded as a missing value.

```
data Times;
   informat Day date7. ;
   format   Day date7. ;
   input Day @ ;
   do Flight=1 to 25;
      input Delay @ ;
      output;
      end;
datalines;
16DEC88   4  12   2   2  18   5   6  21   0   0
          0  14   3   .   2   3   5   0   6  19
          7   4   9   5  10
17DEC88   1  10   3   3   0   1   5   0   .   .
          1   5   7   1   7   2   2  16   2   1
          3   1  31   5   0
18DEC88   7   8   4   2   3   2   7   6  11   3
          2   7   0   1  10   2   3  12   8   6
          2   7   2   4   5
19DEC88  15   6   9   0  15   7   1   1   0   2
          5   6   5  14   7  20   8   1  14   3
         10   0   1  11   7
20DEC88   2   1   0   4   4   6   2   2   1   4
          1  11   .   1   0   6   5   5   4   2
          2   6   6   4   0
21DEC88   2   6   6   2   7   7   5   2   5   0
          9   2   4   2   5   1   4   7   5   6
          5   0   4  36  28
22DEC88   3   7  22   1  11  11  39  46   7  33
         19  21   1   3  43  23   9   0  17  35
         50   0   2   1   0
23DEC88   6  11   8  35  36  19  21   .   .   4
          6  63  35   3  12  34   9   0  46   0
          0  36   3   0  14
24DEC88  13   2  10   4   5  22  21  44  66  13
          8   3   4  27   2  12  17  22  19  36
          9  72   2   4   4
25DEC88   4  33  35   0  11  11  10  28  34   3
         24   6  17   0   8   5   7  19   9   7
         21  17  17   2   6
26DEC88   3   8   8   2   7   7   8   2   5   9
          2   8   2  10  16   9   5  14  15   1
         12   2   2  14  18
;
run;
```

In the following statements, the MEANS procedure is used to count the number of canceled flights for each day. This information is then added to the data set Times.

```
proc means data=Times noprint;
   var Delay;
   by Day;
   output out=Cancel nmiss=ncancel;

data Times;
   merge Times Cancel;
   by Day;
run;
```

The following statements create a data set named Weather containing information about possible causes for delays, and then merge this data set with the data set Times:

```
data Weather;
   informat Day date7. ;
   format   Day date7. ;
   length Reason $ 16 ;
input Day Flight Reason & ;
datalines;
16DEC88  8   Fog
17DEC88  18  Snow Storm
17DEC88  23  Sleet
21DEC88  24  Rain
21DEC88  25  Rain
22DEC88  7   Mechanical
22DEC88  15  Late Arrival
24DEC88  9   Late Arrival
24DEC88  22  Late Arrival
;
run;

data Times;
   merge Times Weather;
   by Day Flight;
run;
```

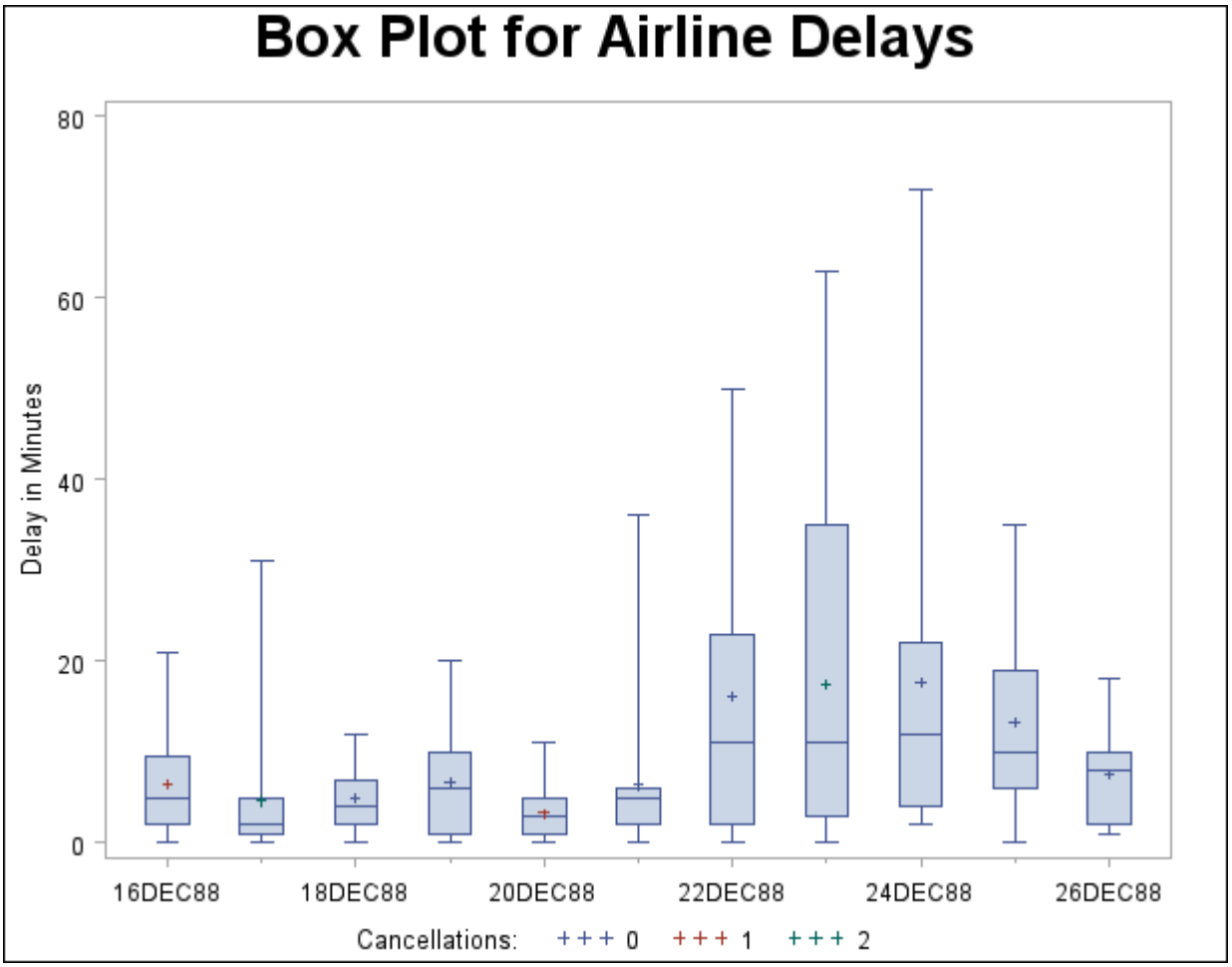
The following statements create a box plot for the complete set of data:

```
symbol1 v=plus;
symbol2 v=square;
symbol3 v=triangle;
title 'Box Plot for Airline Delays';
proc boxplot data=Times;
   plot Delay*Day = ncancel /
       nohlabel
       symbollegend = legend1;
   legend1 label = ('Cancellations:');
   label Delay = 'Delay in Minutes';
run;
goptions reset=symbol;
```

The level of the *symbol variable* ncancel determines the symbol marker for each group mean, and the SYMBOLLEGEND= option controls the appearance of the legend for the symbols. The NOHLA-

BEL option suppresses the horizontal axis label. The resulting box plot is shown in Output 24.2.1.

Output 24.2.1 Box Plot for Airline Data



The delay distributions from December 22 through December 25 are drastically different from the delay distributions during the pre-holiday period. Both the mean delay and the variability of the delays are much greater during the holiday period.

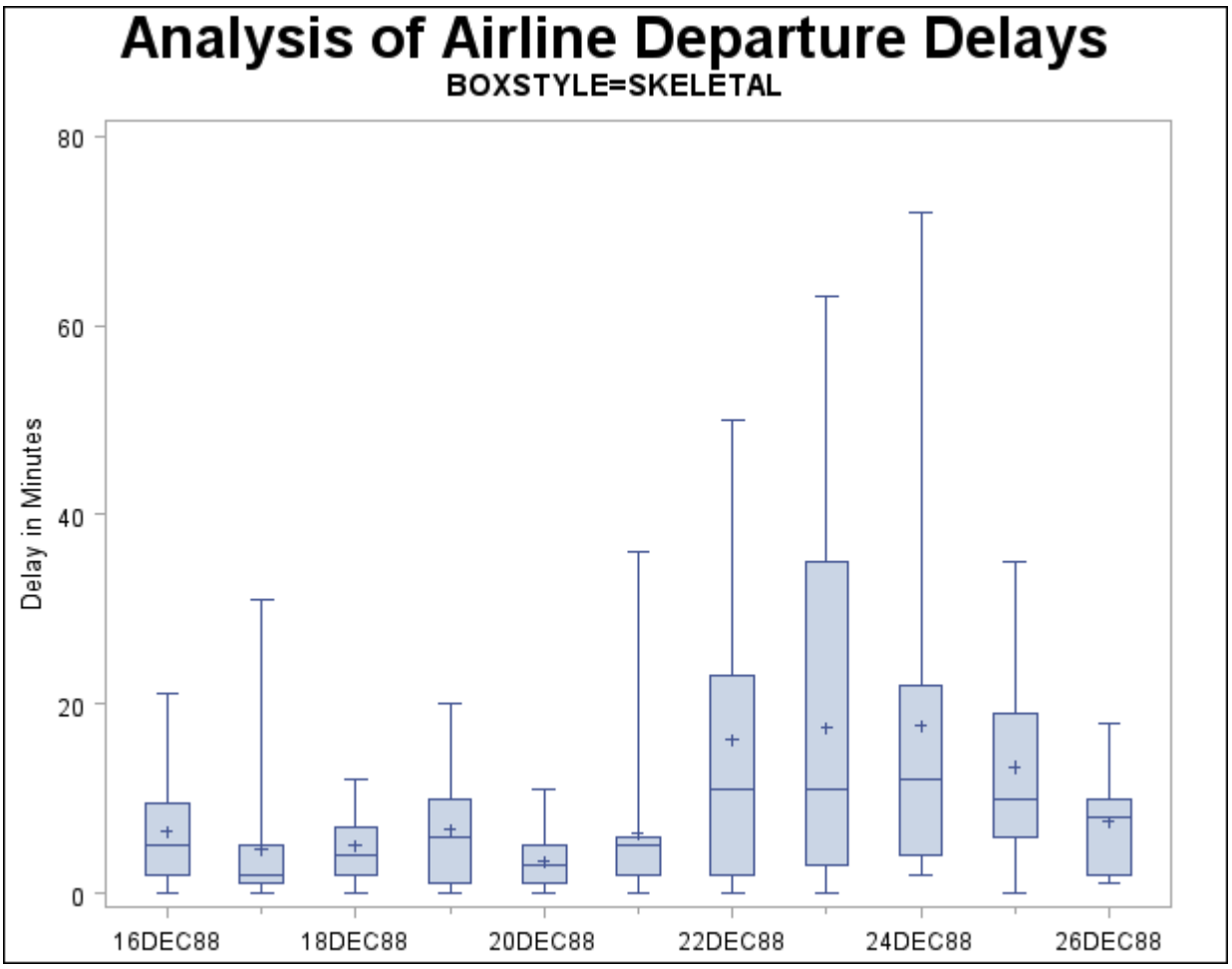
Example 24.3: Creating Various Styles of Box-and-Whiskers Plots

This example uses the flight delay data of the preceding example to illustrate how you can create box plots with various styles of box-and-whiskers plots. The following statements create a plot that displays skeletal box-and-whiskers plots:

```
title 'Analysis of Airline Departure Delays';
title2 'BOXSTYLE=SKELETAL';
proc boxplot data=Times;
   plot Delay*Day /
      boxstyle = skeletal
      nohlabel;
   label Delay = 'Delay in Minutes';
run;
```

In a skeletal box-and-whiskers plot, the whiskers are drawn from the quartiles to the extreme values of the group. The skeletal box plot is the default style, so you can also produce a skeletal box plot by omitting the BOXSTYLE= option. Output 24.3.1 shows the skeletal box plot.

Output 24.3.1 BOXSTYLE=SKELETAL

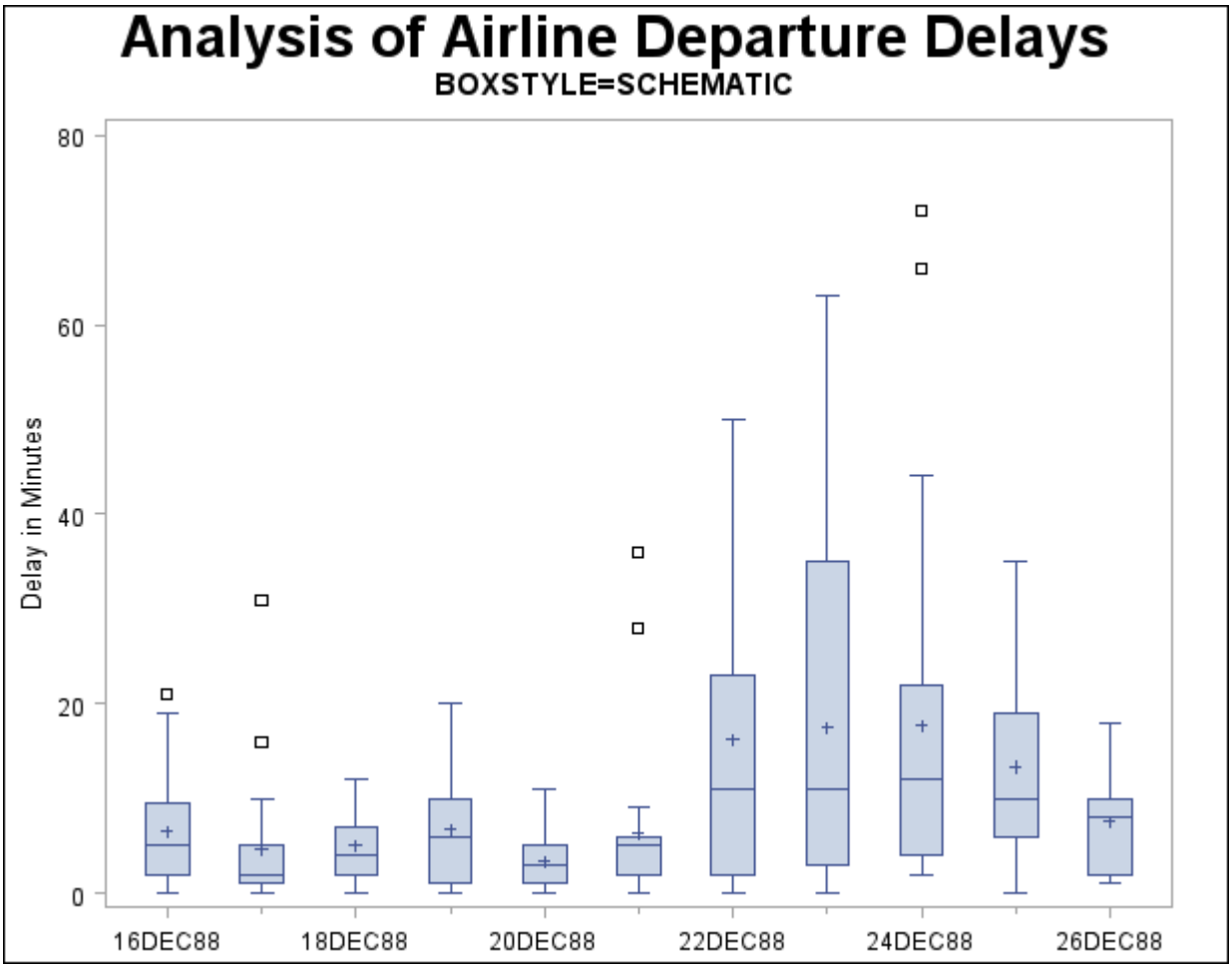


The following statements request a schematic box:

```
title 'Analysis of Airline Departure Delays';
title2 'BOXSTYLE=SCHEMATIC';
proc boxplot data=Times;
   plot Delay*Day /
      boxstyle = schematic
      nohlabel;
   label Delay = 'Delay in Minutes';
run;
```

When you specify BOXSTYLE=SCHEMATIC, the whiskers are drawn to the most extreme points in the group that lie within the *fences*. The *upper fence* is defined as the third quartile (represented by the upper edge of the box) plus 1.5 times the interquartile range (IQR). The *lower fence* is defined as the first quartile (represented by the lower edge of the box) minus 1.5 times the interquartile range. Observations outside the fences are identified with a special symbol. The default symbol is a square, and you can specify the shape and color for this symbol with the IDSYMBOL= and IDCOLOR= options. Serifs are added to the whiskers by default. For further details, see the entry for the BOXSTYLE= option on page 773. The plot is shown in Output 24.3.2.

Output 24.3.2 BOXSTYLE=SCHEMATIC

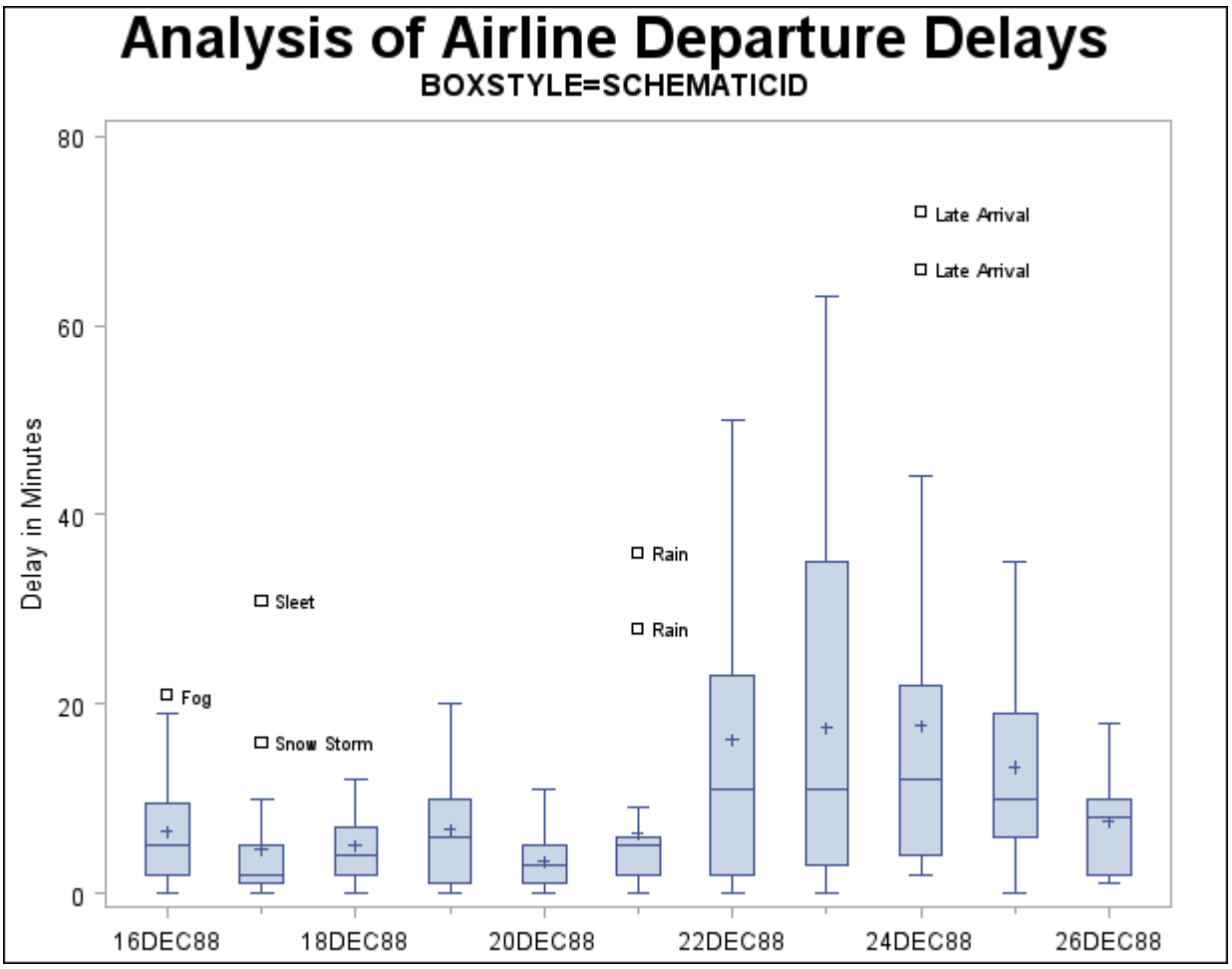


The following statements create a schematic box plot in which the observations outside the fences are labeled:

```
title 'Analysis of Airline Departure Delays';
title2 'BOXSTYLE=SCHEMATICID';
proc boxplot data=Times;
   plot Delay*Day /
      boxstyle = schematicid
      nohlabel;
   id Reason;
   label Delay = 'Delay in Minutes';
run;
```

If you specify BOXSTYLE=SCHEMATICID, schematic box-and-whiskers plots are created and the value of the first ID variable (in this case, Reason) is used to label each observation outside the fences. The box plot is shown in Output 24.3.3.

Output 24.3.3 BOXSTYLE=SCHEMATICID

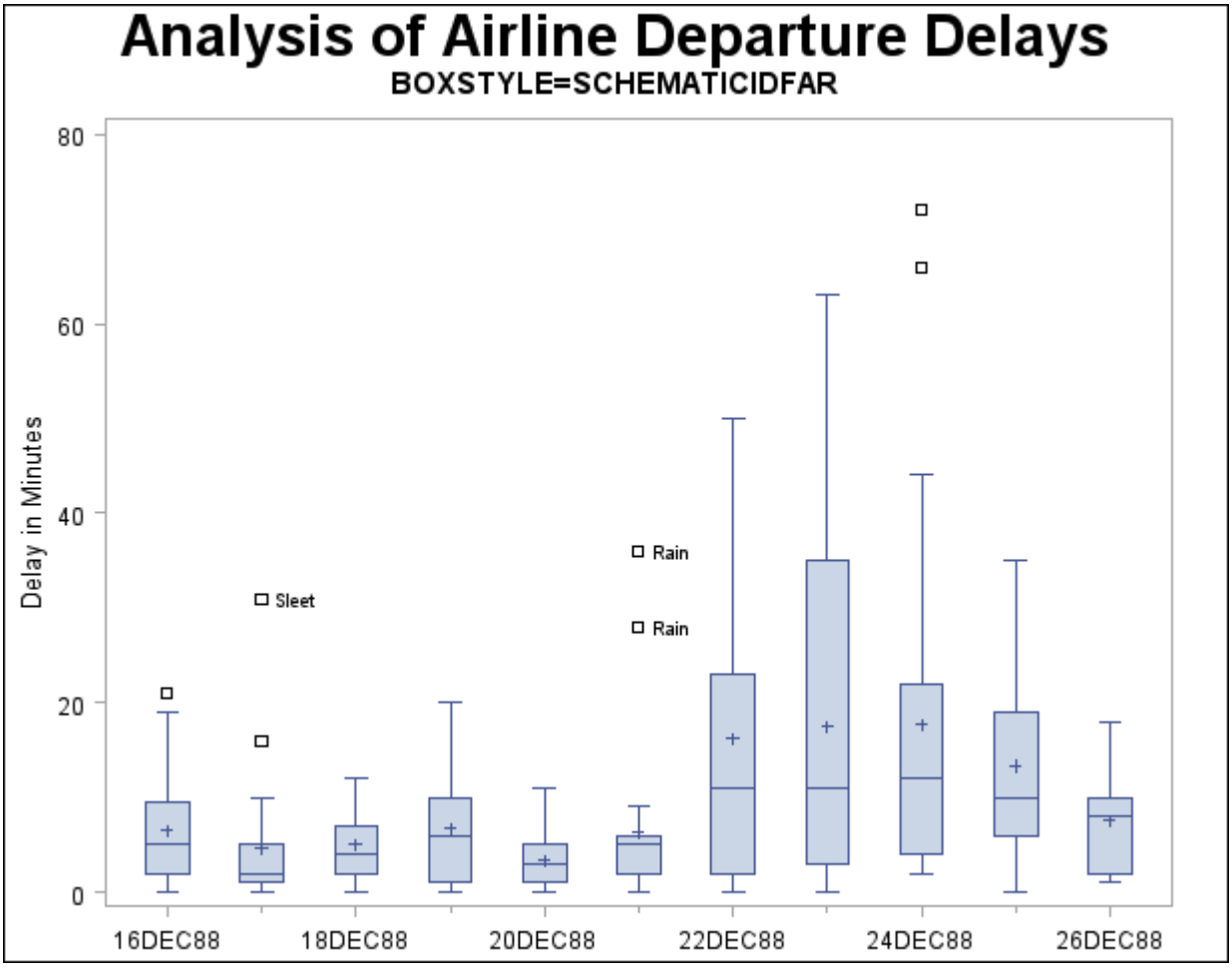


The following statements create a box plot with schematic box-and-whiskers plots in which only the extreme observations outside the fences are labeled:

```
title 'Analysis of Airline Departure Delays';
title2 'BOXSTYLE=SCHEMATICIDFAR';
proc boxplot data=Times;
   plot Delay*Day /
      boxstyle = schematicidfar
      nohlabel;
   id Reason;
   label Delay = 'Delay in Minutes';
run;
```

If you specify BOXSTYLE=SCHEMATICIDFAR, the value of the first ID variable is used to label each observation outside the lower and upper *far fences*. The lower and upper far fences are located 3×IQR below the 25th percentile and 3×IQR above the 75th percentile, respectively. Observations between the fences and the far fences are identified with a symbol but are not labeled. The box plot is shown in Output 24.3.4.

Output 24.3.4 BOXSTYLE=SCHEMATICIDFAR



Other options for controlling the display of high-resolution graphics box plots include the BOXWIDTH=, BOXWIDTHSCALE=, CBOXES=, CBOXFILL=, and LBOXES= options.

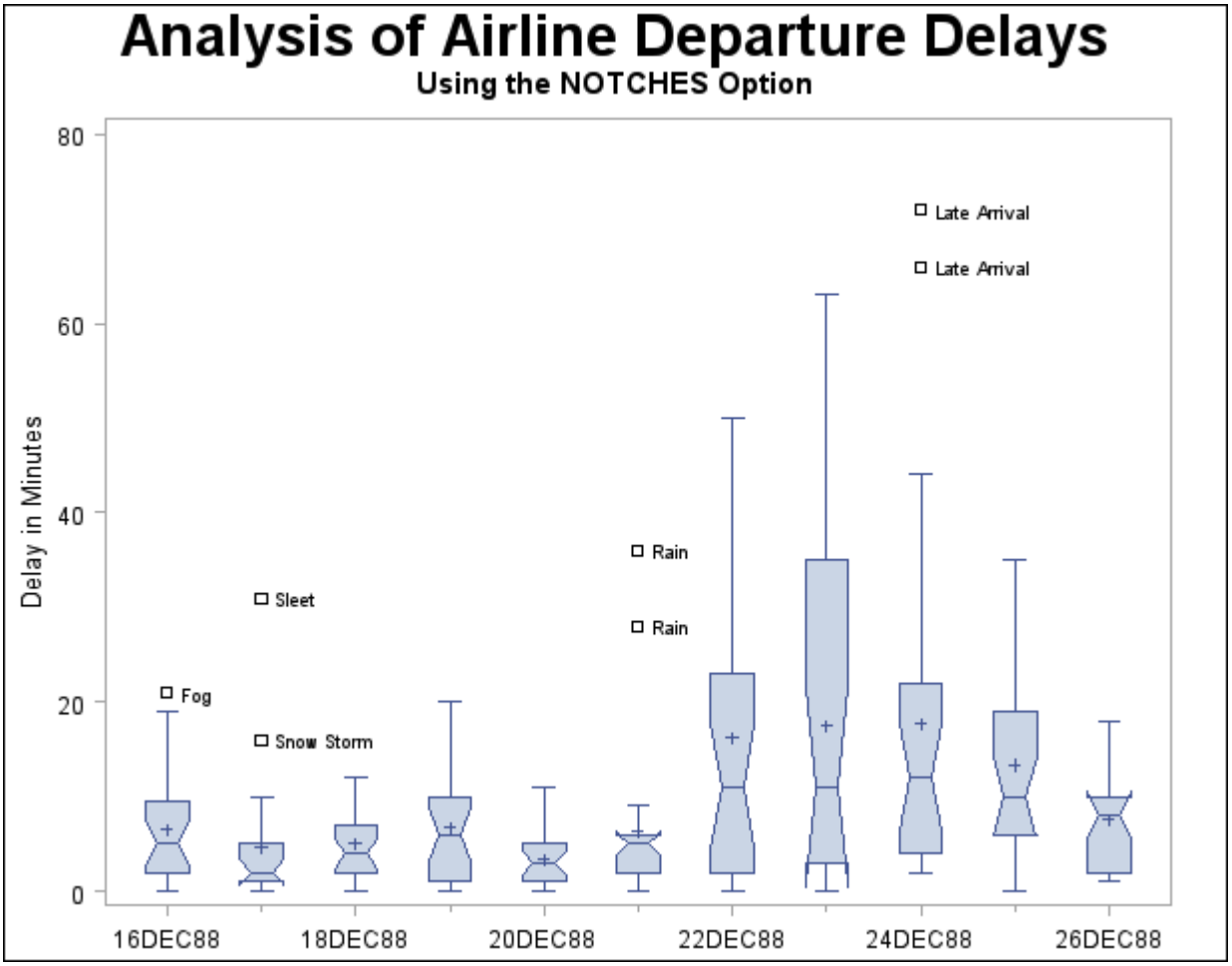
Example 24.4: Creating Notched Box-and-Whiskers Plots

The following statements use the flight delay data of Example 24.1 to create box-and-whiskers plots with notches:

```
title 'Analysis of Airline Departure Delays';
title2 'Using the NOTCHES Option';
proc boxplot data=Times;
   plot Delay*Day /
      boxstyle = schematicid
      nohlabel
      notches;
   id Reason;
   label Delay = 'Delay in Minutes';
run;
```

The notches, requested with the NOTCHES option, measure the significance of the difference between two medians. The medians of two box plots are significantly different at approximately the 0.95 confidence level if the corresponding notches do not overlap. For example, in Output 24.4.1, the median for December 20 is significantly different from the median for December 24.

Output 24.4.1 Notched Side-by-Side Box-and-Whiskers Plots



Example 24.5: Creating Box-and-Whiskers Plots with Varying Widths

This example shows how to create a box plot with box-and-whiskers plots whose widths vary proportionately with the group size. The following statements create a SAS data set named Times2 that contains flight departure delays (in minutes) recorded daily for eight consecutive days:

```
data Times2;
     label Delay = 'Delay in Minutes';
     informat Day date7. ;
     format   Day date7. ;
     input Day @ ;
     do Flight=1 to 25;
        input Delay @ ;
        output;
        end;
datalines;
01MAR90   12  4   2   2  15   8   0  11   0   0
           0 12   3   .   2   3   5   0   6  25
           7  4   9   5  10
02MAR90   1   .   3   .   0   1   5   0   .   .
          1   5   7   .   7   2   2  16   2   1
          3   1  31   .   0
03MAR90   6   8   4   2   3   2   7   6  11   3
          2   7   0   1  10   2   5  12   8   6
          2   7   2   4   5
04MAR90  12   6   9   0  15   7   1   1   0   2
          5   6   5  14   7  21   8   1  14   3
         11   0   1  11   7
05MAR90   2   1   0   4   .   6   2   2   1   4
          1  11   .   1   0   .   5   5   .   2
          3   6   6   4   0
06MAR90   8   6   5   2   9   7   4   2   5   1
          2   2   4   2   5   1   3   9   7   8
          1   0   4  26  27
07MAR90   9   6   6   2   7   8   .   .  10   8
          0   2   4   3   .   .   .   7   .   6
          4   0   .   .   .
08MAR90   1   6   6   2   8   8   5   3   5   0
          8   2   4   2   5   1   6   4   5  10
          2   0   4   1   1
run;
```

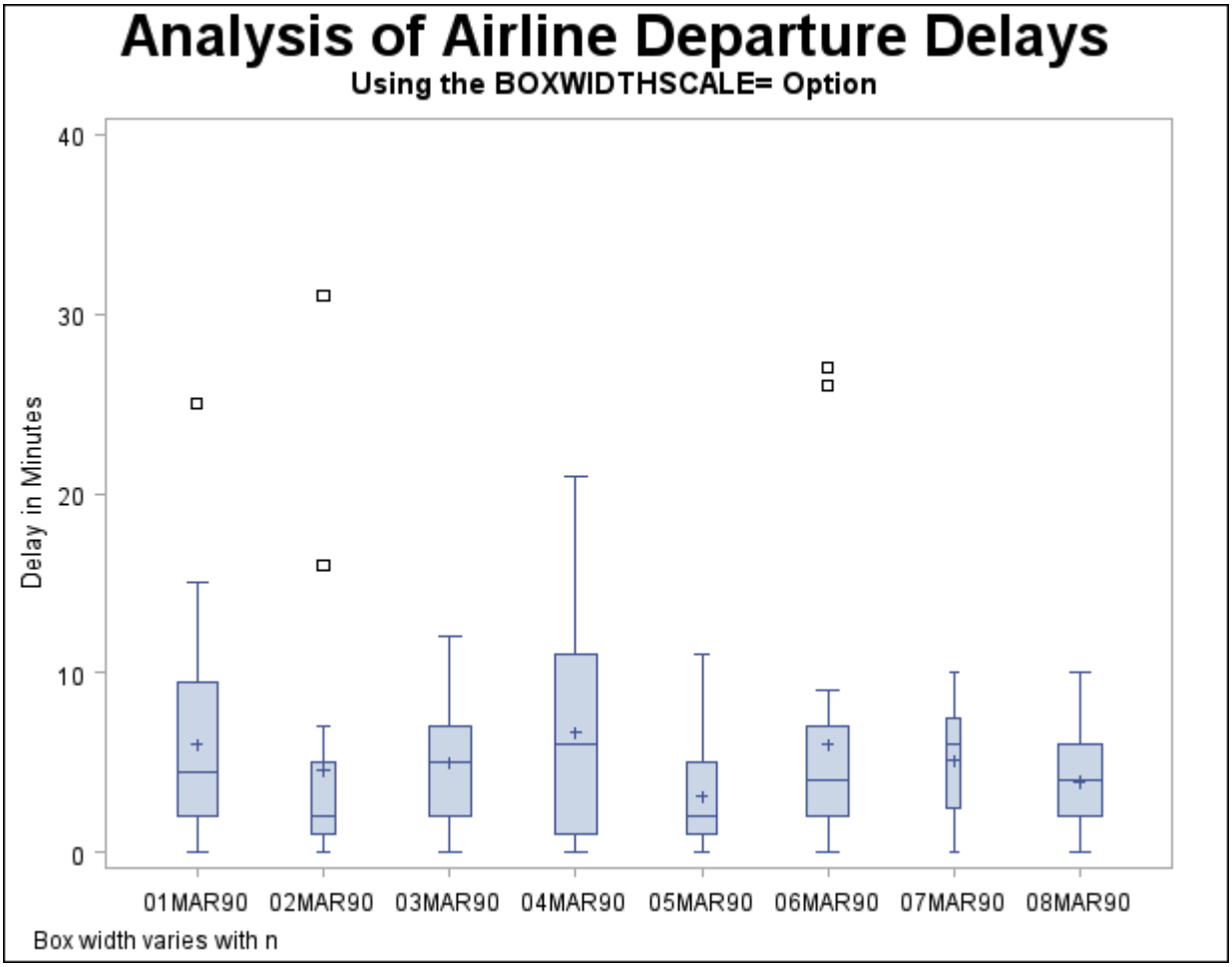
The following statements create a box plot with varying box widths:

```
title 'Analysis of Airline Departure Delays';
title2 'Using the BOXWIDTHSCALE= Option';
proc boxplot data=Times2;
   plot Delay*Day /
      nohlabel
      boxstyle      = schematic
      boxwidthscale = 1
      bwslegend;
run;
```

The BOXWIDTHSCALE=*value* option specifies that the widths of the box-and-whiskers plots vary in proportion to a particular function of the group size n. The function is determined by *value* and is identified on the box plot with a legend if the BWSLEGEND option is specified. The BOXWIDTHSCALE= option is useful in situations where the group sizes vary widely.

Output 24.5.1 shows the resulting box plot.

Output 24.5.1 Box Plot with Box-and-Whiskers Plots of Varying Widths



Example 24.6: Creating Box-and-Whiskers Plots Using ODS Graphics (Experimental)

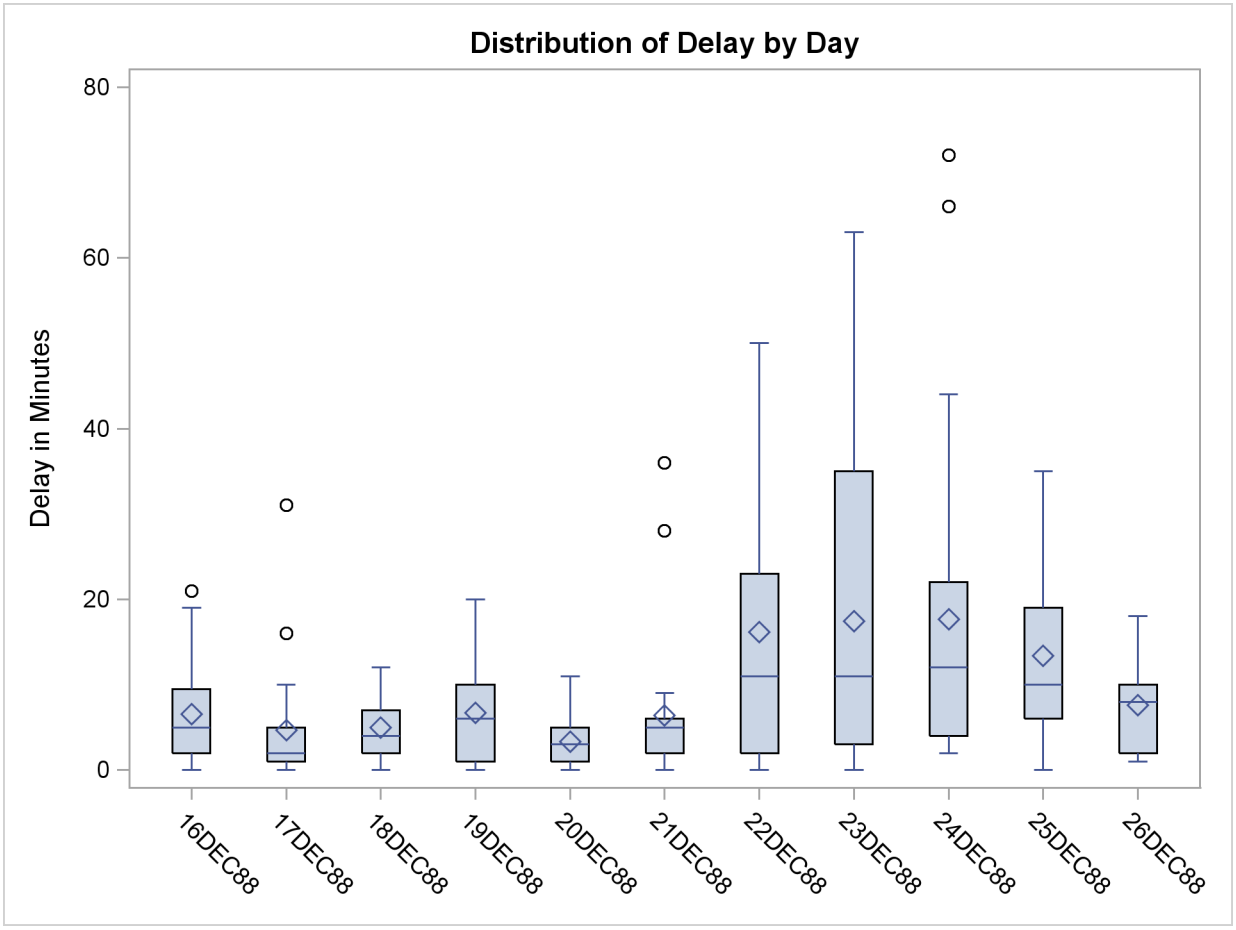
The BOXPLOT procedure supports ODS Graphics on an experimental basis for SAS 9.2.

The following statements use ODS Graphics to produce a box plot of the flight delay data from Example 24.2. You invoke ODS Graphics with the ODS GRAPHICS ON statement.

```
ods graphics on;
title 'Box Plot for Airline Delays';
proc boxplot data=Times;
   plot Delay*Day /
      boxstyle = schematic
      nohlabel;
   label Delay = 'Delay in Minutes';
run;
```

The resulting box plot is shown in Output 24.6.1.

Output 24.6.1 Box Plot Produced Using ODS Graphics



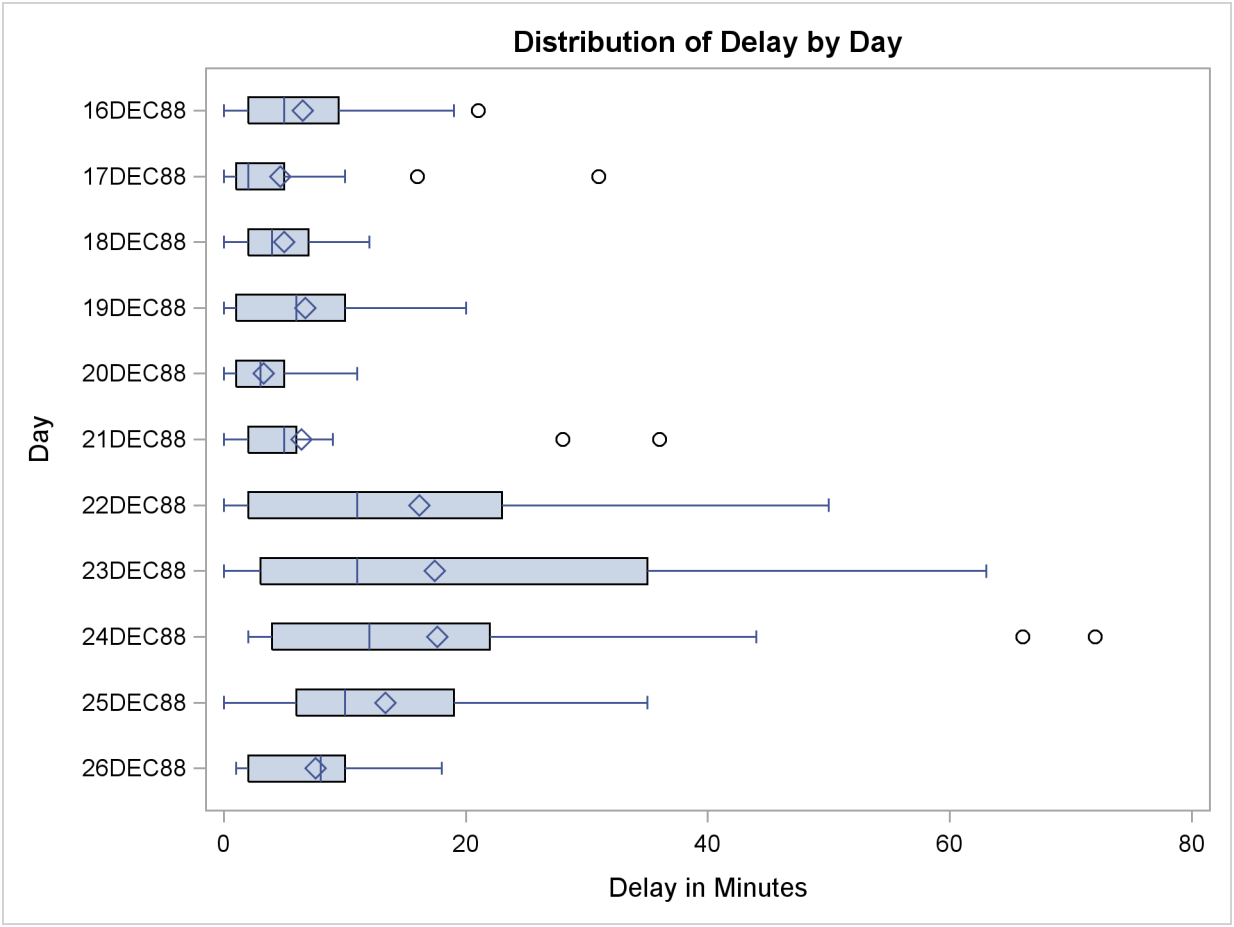
ODS graphical displays, like traditional high-resolution graphs in SAS 9.2, are controlled by the ODS style currently in effect for the output destination where the box plots are produced. However, unlike high-resolution graphs, ODS graphs are unaffected by GOPTIONS and SYMBOL statements, and by PLOT statement options used to specify colors, fonts, and other features affecting box plot appearance. Options such as BOXSTYLE= and NOHLABEL *are* honored by the PLOT statement when producing ODS graphical output.

The following statements use the HORIZONTAL option, which is supported only by ODS Graphics, to produce a horizontal box plot:

```
title 'Box Plot for Airline Delays';
proc boxplot data=Times;
   plot Delay*Day /
      boxstyle = schematic
      horizontal;
   label Delay = 'Delay in Minutes';
run;
ods graphics off;
```

The horizontal box plot is shown in Output 24.6.2.

Output 24.6.2 Horizontal Box Plot



References

McGill, R., Tukey, J. W., and Larsen, W. A. (1978), "Variations of Box Plots," *The American Staistician*, 32, 12–16.

Tukey, J. W. (1977), *Exploratory Data Analysis*, Reading, MA: Addison-Wesley.

Chapter 25

The CALIS Procedure

Contents

Overview: CALIS Procedure **828**
Structural Equation Models 832
Getting Started: CALIS Procedure **839**
Syntax: CALIS Procedure **846**
PROC CALIS Statement 848
BOUNDS Statement 871
BY Statement 872
COSAN Model Statement 873
COV Statement 874
FACTOR Model Statement 877
FREQ Statement 880
LINCON Statement 880
LINEQS Model Statement 881
MATRIX Statement 883
NLINCON Statement 886
NLOPTIONS Statement 886
PARAMETERS Statement 898
PARTIAL Statement 899
RAM Model Statement 899
SAS Programming Statements 903
STD Statement 906
STRUCTEQ Statement 907
VAR Statement 907
VARNAMES Statement 908
WEIGHT Statement 909
Details: CALIS Procedure **909**
Input Data Sets 909
Output Data Sets 913
Missing Values 923
Estimation Criteria 923
Relationships among Estimation Criteria 926
Testing Rank Deficiency in the Approximate Covariance Matrix 927
Approximate Standard Errors 928
Assessment of Fit 928

Latent Variable Scores . . . 937
Measures of Multivariate Kurtosis . . . 938
Initial Estimates . . . 941
Automatic Variable Selection . . . 942
Exogenous Manifest Variables . . . 943
Use of Optimization Techniques . . . 945
Modification Indices . . . 953
Constrained Estimation by Using Program Code . . . 955
Counting the Degrees of Freedom . . . 956
Computational Problems . . . 958
Displayed Output . . . 961
ODS Table Names . . . 967
ODS Graphics . . . 970
Examples: CALIS Procedure . . . **970**
Example 25.1: Path Analysis: Stability of Alienation . . . 970
Example 25.2: Simultaneous Equations with Intercept . . . 993
Example 25.3: Second-Order Confirmatory Factor Analysis . . . 999
Example 25.4: Linear Relations among Factor Loadings . . . 1006
Example 25.5: Ordinal Relations among Factor Loadings . . . 1016
Example 25.6: Longitudinal Factor Analysis . . . 1021
References . . . **1023**

Overview: CALIS Procedure

Structural equation modeling is an important statistical tool in economics and behavioral sciences. Structural equations express relationships among several variables that can be either directly observed variables (manifest variables) or unobserved hypothetical variables (latent variables). For an introduction to latent variable models, refer to Loehlin (1987), Bollen (1989b), Everitt (1984), or Long (1983); and for manifest variables, refer to Fuller (1987).

In structural models, as opposed to functional models, all variables are taken to be random rather than having fixed levels. For maximum likelihood (default) and generalized least squares estimation in PROC CALIS, the random variables are assumed to have an approximately multivariate normal distribution. Nonnormality, especially high kurtosis, can produce poor estimates and grossly incorrect standard errors and hypothesis tests, even in large samples. Consequently, the assumption of normality is much more important than in models with nonstochastic exogenous variables. You should remove outliers and consider transformations of nonnormal variables before using PROC CALIS with maximum likelihood (default) or generalized least squares estimation. If the number of observations is sufficiently large, Browne's asymptotically distribution-free (ADF) estimation method can be used.

You can use the CALIS procedure to estimate parameters and test hypotheses for constrained and unconstrained problems in the following:

- multiple and multivariate linear regression
- linear measurement-error models
- path analysis and causal modeling
- simultaneous equation models with reciprocal causation
- exploratory and confirmatory factor analysis of any order
- canonical correlation
- a wide variety of other (non)linear latent variable models

The parameters are estimated using the following criteria:

- unweighted least squares (ULS)
- generalized least squares (GLS, with optional weight matrix input)
- maximum likelihood (ML, for multivariate normal data)
- weighted least squares (WLS, ADF, with optional weight matrix input)
- diagonally weighted least squares (DWLS, with optional weight matrix input)

The default weight matrix for generalized least squares estimation is the sample covariance or correlation matrix. The default weight matrix for weighted least squares estimation is an estimate of the asymptotic covariance matrix of the sample covariance or correlation matrix. In this case, weighted least squares estimation is equivalent to Browne's (1982, 1984) asymptotic distribution-free estimation. The default weight matrix for diagonally weighted least squares estimation is an estimate of the asymptotic variances of the input sample covariance or correlation matrix. You can also use an input data set to specify the weight matrix in GLS, WLS, and DWLS estimation.

Estimation methods implemented in PROC CALIS do not exhaust all alternatives in the field. For example, partial least squares (PLS) is not implemented. See the section "Estimation Criteria" on page 923 for details about estimation criteria used in PROC CALIS. Note that there is a SAS/STAT procedure called PROC PLS, which employs the partial least squares technique but for a class of models different from those of PROC CALIS. For general path analysis or structural equation model with latent variables you should consider using PROC CALIS.

You can specify the model in several ways:

- If you have a set of structural equations to describe the model, you can use an equation-type LINEQS statement similar to that originally developed by Bentler (1985).
- You can specify simple path models by using an easily formulated list-type RAM statement similar to that originally developed by McArdle (McArdle and McDonald 1984).

- You can do a constrained (confirmatory) first-order factor analysis or component analysis by using the FACTOR statement.
- You can analyze a broad family of matrix models by using COSAN and MATRIX statements that are similar to the COSAN program of McDonald and Fraser (McDonald 1978, 1980). It enables you to specify complex matrix models including nonlinear equation models and higher-order factor models.

You can specify linear and nonlinear equality and inequality constraints on the parameters with several different statements, depending on the type of input. Lagrange multiplier test indices are computed for simple constant and equality parameter constraints and for active boundary constraints. General equality and inequality constraints can be formulated using programming statements. For more information, see the section "SAS Programming Statements" on page 903.

PROC CALIS offers a variety of methods for the automatic generation of initial values for the optimization process:

- two-stage least squares estimation
- instrumental variable factor analysis
- approximate factor analysis
- ordinary least squares estimation
- McDonald's (McDonald and Hartmann 1992) method

In many common applications, these initial values prevent computational problems and save computer time.

Because numerical problems can occur in the (non)linearly constrained optimization process, the CALIS procedure offers several optimization algorithms:

- Levenberg-Marquardt algorithm (Moré, 1978)
- trust-region algorithm (Gay 1983)
- Newton-Raphson algorithm with line search
- ridge-stabilized Newton-Raphson algorithm
- various quasi-Newton and dual quasi-Newton algorithms: Broyden-Fletcher-Goldfarb-Shanno and Davidon-Fletcher-Powell, including a sequential quadratic programming algorithm for processing nonlinear equality and inequality constraints
- various conjugate gradient algorithms: automatic restart algorithm of Powell (1977), Fletcher-Reeves, Polak-Ribiere, and conjugate descent algorithm of Fletcher (1980)

The quasi-Newton and conjugate gradient algorithms can be modified by several line-search methods. All of the optimization techniques can impose simple boundary and general linear constraints on the parameters. Only the dual quasi-Newton algorithm is able to impose general nonlinear equality and inequality constraints.

The procedure creates an OUTRAM= output data set that completely describes the model (except for program statements) and also contains parameter estimates. This data set can be used as input for another execution of PROC CALIS. Small model changes can be made by editing this data set, so you can exploit the old parameter estimates as starting values in a subsequent analysis. An OUTEST= data set contains information about the optimal parameter estimates (parameter estimates, gradient, Hessian, projected Hessian and Hessian of Lagrange function for constrained optimization, the information matrix, and standard errors). The OUTEST= data set can be used as an INEST= data set to provide starting values and boundary and linear constraints for the parameters. An OUTSTAT= data set contains residuals and, for exploratory factor analysis, the rotated and unrotated factor loadings.

Automatic variable selection (using only those variables from the input data set that are used in the model specification) is performed in connection with the RAM and LINEQS input statements or when these models are recognized in an input model file. Also in these cases, the covariances of the exogenous manifest variables are recognized as given constants. With the PREDET option, you can display the predetermined pattern of constant and variable elements in the predicted model matrix before the minimization process starts. For more information, see the section "Automatic Variable Selection" on page 942 and the section "Exogenous Manifest Variables" on page 943.

PROC CALIS offers an analysis of linear dependencies in the information matrix (approximate Hessian matrix) that might be helpful in detecting unidentified models. You also can save the information matrix and the approximate covariance matrix of the parameter estimates (inverse of the information matrix), together with parameter estimates, gradient, and approximate standard errors, in an output data set for further analysis.

PROC CALIS does not provide the analysis of multiple samples with different sample size or a generalized algorithm for missing values in the data. However, the analysis of multiple samples with equal sample size can be performed by the analysis of a moment supermatrix containing the individual moment matrices as block diagonal submatrices.

The new experimental procedure TCALIS is now available. Except for the COSAN model specification, PROC TCALIS supports almost all model specification methods that are available in the CALIS procedure. In addition, there are many new features in PROC TCALIS, including a PATH statement that enables you to specify models by using a path-like syntax, an MSTRUCT statement that enables you to specify patterned covariance structures directly, multiple-group analysis, enhanced mean and covariance structure analysis, a priori parametric function testing, effect analysis with standard error estimates, and more. For more information, see Chapter 88, "The TCALIS Procedure (Experimental)."

The CALIS procedure uses ODS Graphics to create graphs as part of its output. High-quality residual histograms are available in PROC CALIS. See Chapter 21, "Statistical Graphics Using ODS," for general information about ODS Graphics. See the section "ODS Graphics" on page 970 and the PLOTS= option on page 864 for specific information about the statistical graphics available with the CALIS procedure.

Structural Equation Models

The Generalized COSAN Model

PROC CALIS can analyze matrix models of the form

$$\mathbf{C} = \mathbf{F}_1\mathbf{P}_1\mathbf{F}_1' + \cdots + \mathbf{F}_m\mathbf{P}_m\mathbf{F}_m'$$

where $\mathbf{C}$ is a symmetric correlation or covariance matrix, each matrix $\mathbf{F}_k$, $k = 1, \ldots, m$, is the product of $n(k)$ matrices $\mathbf{F}_{k_1}, \ldots, \mathbf{F}_{k_{n(k)}}$, and each matrix $\mathbf{P}_k$ is symmetric; that is,

$$\mathbf{F}_k = \mathbf{F}_{k_1} \cdots \mathbf{F}_{k_{n(k)}} \qquad \text{and} \quad \mathbf{P}_k = \mathbf{P}_k', \quad k = 1, \ldots, m$$

The matrices $\mathbf{F}_{k_j}$ and $\mathbf{P}_k$ in the model are parameterized by the matrices $\mathbf{G}_{k_j}$ and $\mathbf{Q}_k$

$$\mathbf{F}_{k_j} = \begin{cases} \mathbf{G}_{k_j} \\ \mathbf{G}_{k_j}^{-1} \\ (\mathbf{I} - \mathbf{G}_{k_j})^{-1} \end{cases} \qquad j = 1, \ldots, n(k) \qquad \text{and} \qquad \mathbf{P}_k = \begin{cases} \mathbf{Q}_k \\ \mathbf{Q}_k^{-1} \end{cases}$$

where you can specify the type of matrix desired.

The matrices $\mathbf{G}_{k_j}$ and $\mathbf{Q}_k$ can contain the following:

- constant values
- parameters to be estimated
- values computed from parameters via programming statements

The parameters can be summarized in a parameter vector $\mathbf{X} = (x_1, \ldots, x_t)$. For a given covariance or correlation matrix $\mathbf{C}$, PROC CALIS computes the unweighted least squares (ULS), generalized least squares (GLS), maximum likelihood (ML), weighted least squares (WLS), or diagonally weighted least squares (DWLS) estimates of the vector $\mathbf{X}$.

Some Special Cases of the Generalized COSAN Model

Original COSAN (Covariance Structure Analysis) Model (McDonald 1978, 1980)

Covariance structure:

$$\mathbf{C} = \mathbf{F}_1 \cdots \mathbf{F}_n\mathbf{P}\mathbf{F}_n' \cdots \mathbf{F}_1'$$

Reticular Action Model—RAM (McArdle 1980; McArdle and McDonald 1984)

Structural equation model:

$$\mathbf{v} = \mathbf{A}\mathbf{v} + \mathbf{u}$$

where $\mathbf{A}$ is a matrix of coefficients, and $\mathbf{v}$ and $\mathbf{u}$ are vectors of random variables. The variables in $\mathbf{v}$ and $\mathbf{u}$ can be manifest or latent variables. The endogenous variables corresponding to the components in $\mathbf{v}$ are expressed as a linear combination of the remaining variables and a residual component in $\mathbf{u}$ with covariance matrix $\mathbf{P}$.

Covariance structure:

$$\mathbf{C} = \mathbf{J}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{P}((\mathbf{I} - \mathbf{A})^{-1})'\mathbf{J}'$$

with selection matrix $\mathbf{J}$ and

$$\mathbf{C} = \mathcal{E}\{\mathbf{J}\mathbf{v}\mathbf{v}'\mathbf{J}'\} \qquad \text{and} \qquad \mathbf{P} = \mathcal{E}\{\mathbf{u}\mathbf{u}'\}$$

LINEQS (Linear Equations) Model (Bentler and Weeks 1980)

Structural equation model:

$$\eta = \beta\eta + \gamma\xi$$

where β and γ are coefficient matrices, and η and ξ are vectors of random variables. The components of η correspond to the endogenous variables; the components of ξ correspond to the exogenous variables and to error variables. The variables in η and ξ can be manifest or latent variables. The endogenous variables in η are expressed as a linear combination of the remaining endogenous variables, the exogenous variables in ξ, and a residual component in ξ. The coefficient matrix β describes the relationships among the endogenous variables of η, and $I - \beta$ should be nonsingular. The coefficient matrix γ describes the relationships between the endogenous variables of η and the exogenous and error variables of ξ.

Covariance structure:

$$\mathbf{C} = \mathbf{J}(\mathbf{I} - \mathbf{B})^{-1}\mathbf{\Gamma}\mathbf{\Phi}\mathbf{\Gamma}'((\mathbf{I} - \mathbf{B})^{-1})'\mathbf{J}'$$

with selection matrix $\mathbf{J}$, $\mathbf{\Phi} = \mathcal{E}\{\xi\xi'\}$, and

$$\mathbf{B} = \begin{pmatrix} \beta & 0 \\ 0 & 0 \end{pmatrix} \qquad \text{and} \qquad \mathbf{\Gamma} = \begin{pmatrix} \gamma \\ \mathbf{I} \end{pmatrix}$$

Keesling-Wiley-Jöreskog LISREL (Linear Structural Relationship) Model (Keesling 1972; Wiley 1973; Jöreskog 1973)

Structural equation model and measurement models:

$$\eta = \mathbf{B}\eta + \mathbf{\Gamma}\xi + \zeta \quad , \qquad \mathbf{y} = \mathbf{\Lambda}_y\eta + \varepsilon \quad , \qquad \mathbf{x} = \mathbf{\Lambda}_x\xi + \delta$$

where η and ξ are vectors of latent variables (factors), and $\mathbf{x}$ and $\mathbf{y}$ are vectors of manifest variables. The components of η correspond to endogenous latent variables; the components of ξ correspond to exogenous latent variables. The endogenous and exogenous latent variables are connected by a system of linear equations (the structural model) with coefficient matrices $\mathbf{B}$ and $\mathbf{\Gamma}$ and an error vector ζ. It is assumed that matrix $\mathbf{I} - \mathbf{B}$ is nonsingular. The random vectors $\mathbf{y}$ and $\mathbf{x}$ correspond to manifest variables that are related to the latent variables η and ξ by two systems of linear equations (the measurement model) with coefficients $\mathbf{\Lambda}_y$ and $\mathbf{\Lambda}_x$ and with measurement errors ε and δ.

Covariance structure:

$$\begin{aligned} \mathbf{C} &= \mathbf{J}(\mathbf{I}-\mathbf{A})^{-1}\mathbf{P}((\mathbf{I}-\mathbf{A})^{-1})'\mathbf{J}' \\ A &= \begin{pmatrix} 0 & 0 & \mathbf{\Lambda}_y & 0 \\ 0 & 0 & 0 & \mathbf{\Lambda}_x \\ 0 & 0 & \mathbf{B} & \mathbf{\Gamma} \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{and} \quad P = \begin{pmatrix} \mathbf{\Theta}_{\boldsymbol{\varepsilon}} & & & \\ & \mathbf{\Theta}_{\boldsymbol{\delta}} & & \\ & & \mathbf{\Psi} & \\ & & & \mathbf{\Phi} \end{pmatrix} \end{aligned}$$

with selection matrix $\mathbf{J}$, $\mathbf{\Phi} = \mathcal{E}\{\boldsymbol{\xi}\boldsymbol{\xi}'\}$, $\mathbf{\Psi} = \mathcal{E}\{\boldsymbol{\zeta}\boldsymbol{\zeta}'\}$, $\mathbf{\Theta}_\delta = \mathcal{E}\{\boldsymbol{\delta}\boldsymbol{\delta}'\}$, and $\mathbf{\Theta}_\varepsilon = \mathcal{E}\{\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}'\}$.

Higher-Order Factor Analysis Models

First-order model:

$$\mathbf{C} = \mathbf{F}_1\mathbf{P}_1\mathbf{F}_1' + \mathbf{U}_1^2$$

Second-order model:

$$\mathbf{C} = \mathbf{F}_1\mathbf{F}_2\mathbf{P}_2\mathbf{F}_2'\mathbf{F}_1' + \mathbf{F}_1\mathbf{U}_2^2\mathbf{F}_1' + \mathbf{U}_1^2$$

First-Order Autoregressive Longitudinal Factor Model

Example of McDonald (1980): $k = 3$: Occasions of Measurement; $n = 3$: Variables (Tests); $m = 2$: Common Factors

$$\begin{aligned} \mathbf{C} &= \mathbf{F}_1\mathbf{F}_2\mathbf{F}_3\mathbf{L}\mathbf{F}_3^{-1}\mathbf{F}_2^{-1}\mathbf{P}(\mathbf{F}_2^{-1})'(\mathbf{F}_3^{-1})'\mathbf{L}'\mathbf{F}_3'\mathbf{F}_2'\mathbf{F}_1' + \mathbf{U}^2 \\ F_1 &= \begin{pmatrix} B_1 & & \\ & B_2 & \\ & & B_3 \end{pmatrix}, \quad F_2 = \begin{pmatrix} I_2 & & \\ & D_2 & \\ & & D_2 \end{pmatrix}, \quad F_3 = \begin{pmatrix} I_2 & & \\ & I_2 & \\ & & D_3 \end{pmatrix} \\ L &= \begin{pmatrix} I_2 & o & o \\ I_2 & I_2 & o \\ I_2 & I_2 & I_2 \end{pmatrix}, \quad P = \begin{pmatrix} I_2 & & \\ & S_2 & \\ & & S_3 \end{pmatrix}, \quad U = \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ U_{21} & U_{22} & U_{23} \\ U_{31} & U_{32} & U_{33} \end{pmatrix} \\ S_2 &= I_2 - D_2^2 \quad , \qquad S_3 = I_2 - D_3^2 \end{aligned}$$

For more information about this model, see Example 25.6.

A Structural Equation Example

This example from Wheaton et al. (1977) illustrates the relationships among the RAM, LINEQS, and LISREL models. Different structural models for these data are in Jöreskog and Sörbom (1985) and in Bentler (1985, p. 28). The data set contains covariances among six (manifest) variables collected from 932 people in rural regions of Illinois:

Variable 1: V1, y_1 : Anomie 1967

Variable 2: V2, y_2 : Powerlessness 1967

Variable 3: V3, y_3 : Anomie 1971

Variable 4: V4, y_4 : Powerlessness 1971

Variable 5: V5, x_1 : Education (years of schooling)

Variable 6: V6, x_2 : Duncan's Socioeconomic Index (SEI)

It is assumed that anomie and powerlessness are indicators of an alienation factor and that education and SEI are indicators for a socioeconomic status (SES) factor. Hence, the analysis contains three latent variables:

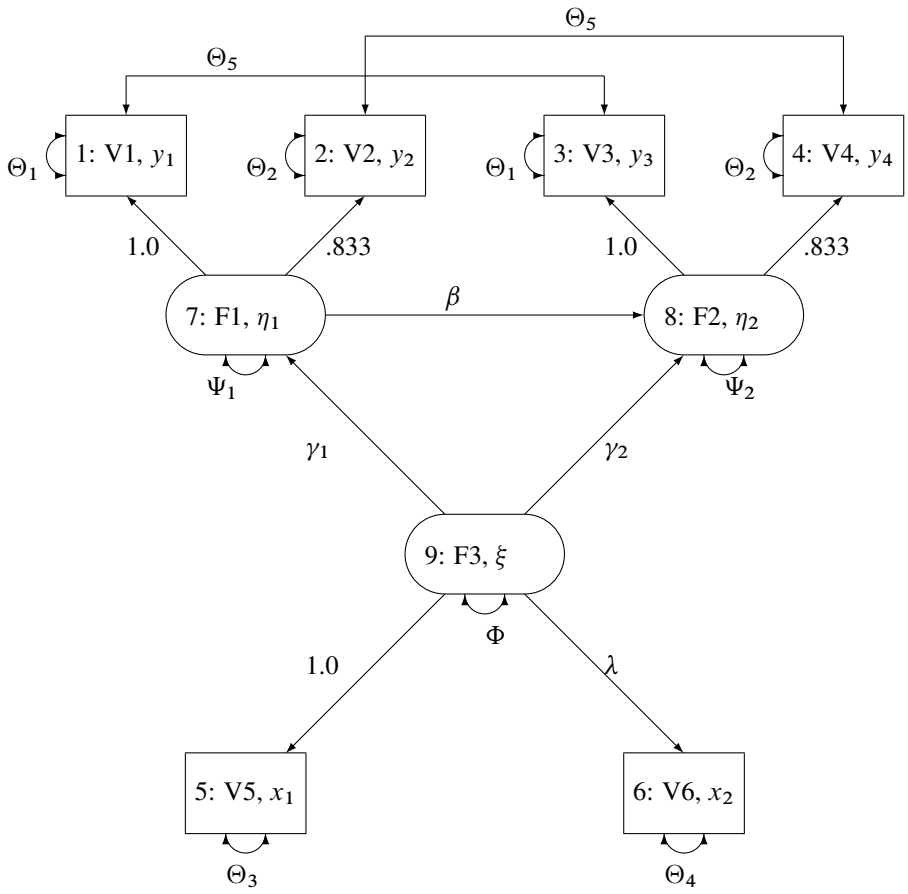
Variable 7: F1, η_1 : Alienation 1967

Variable 8: F2, η_2 : Alienation 1971

Variable 9: F3, ξ_1 : Socioeconomic Status (SES)

The following path diagram shows the structural model used in Bentler (1985, p. 29) and slightly modified in Jöreskog and Sörbom (1985, p. 56). In this notation for the path diagram, regression coefficients between the variables are indicated as one-headed arrows. Variances and covariances among the variables are indicated as two-headed arrows. Indicating error variances and covariances as two-headed arrows with the same source and destination (McArdle 1988; McDonald 1985) is helpful in transforming the path diagram to RAM model list input for the CALIS procedure.

Figure 25.1 Path Diagram of Stability and Alienation Example



Variables in Figure 25.1 are as follows:

Variable 1: V1, y_1 : Anomie 1967

Variable 2: V2, y_2 : Powerlessness 1967

Variable 3: V3, y_3 : Anomie 1971

Variable 4: V4, y_4 : Powerlessness 1971

Variable 5: V5, x_1 : Education (years of schooling)

Variable 6: V6, x_2 : Duncan's Socioeconomic Index (SEI)

Variable 7: F1, η_1 : Alienation 1967

Variable 8: F2, η_2 : Alienation 1971

Variable 9: F3, ξ_1 : Socioeconomic Status (SES)

LINEQS Model

The vector $\boldsymbol{\eta}$ contains the six endogenous manifest variables V1, ..., V6 and the two endogenous latent variables F1 and F2. The vector $\boldsymbol{\xi}$ contains the exogenous error variables E1, ..., E6, D1, and D2 and the exogenous latent variable F3. The path diagram corresponds to the following set of structural equations of the LINEQS model:

$$\begin{aligned}
V1 &= 1.000F1 + E1 \\
V2 &= 0.833F1 + E2 \\
V3 &= 1.000F2 + E3 \\
V4 &= 0.833F2 + E4 \\
V5 &= 1.000F3 + E5 \\
V6 &= \lambda F3 + E6 \\
F1 &= \gamma_1 F3 + D1 \\
F2 &= \beta F1 + \gamma_2 F3 + D2
\end{aligned}$$

This gives the matrices $\boldsymbol{\beta}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\Phi}$ in the LINEQS model:

$$\boldsymbol{\beta} = \begin{pmatrix}
o & o & o & o & o & o & 1.000 & o \\
o & o & o & o & o & o & 0.833 & o \\
o & o & o & o & o & o & o & 1.000 \\
o & o & o & o & o & o & o & 0.833 \\
o & o & o & o & o & o & o & o \\
o & o & o & o & o & o & o & o \\
o & o & o & o & o & o & o & o \\
o & o & o & o & o & o & \beta & o
\end{pmatrix}, \quad
\boldsymbol{\gamma} = \begin{pmatrix}
1 & o & o & o & o & o & o & o & o \\
o & 1 & o & o & o & o & o & o & o \\
o & o & 1 & o & o & o & o & o & o \\
o & o & o & 1 & o & o & o & o & o \\
o & o & o & o & 1 & o & o & o & 1.000 \\
o & o & o & o & o & 1 & o & o & \lambda \\
o & o & o & o & o & o & 1 & o & \gamma_1 \\
o & o & o & o & o & o & o & 1 & \gamma_2
\end{pmatrix}$$

$$\boldsymbol{\Phi} = \begin{pmatrix}
\theta_1 & o & \theta_5 & o & o & o & o & o & o \\
o & \theta_2 & o & \theta_5 & o & o & o & o & o \\
\theta_5 & o & \theta_1 & o & o & o & o & o & o \\
o & \theta_5 & o & \theta_2 & o & o & o & o & o \\
o & o & o & o & \theta_3 & o & o & o & o \\
o & o & o & o & o & \theta_4 & o & o & o \\
o & o & o & o & o & o & \psi_1 & o & o \\
o & o & o & o & o & o & o & \psi_2 & o \\
o & o & o & o & o & o & o & o & \phi
\end{pmatrix}$$

The LINEQS model input specification of this example for the CALIS procedure is given in the section "LINEQS Model Specification" on page 841.

RAM Model

The vector **v** contains the six manifest variables v_1 =V1, ..., v_6 =V6 and the three latent variables v_7 =F1, v_8 =F2, v_9 =F3. The vector **u** contains the corresponding error variables u_1 =E1, ..., u_6 =E6 and u_7 =D1, u_8 =D2, u_9 =D3. The path diagram corresponds to the following set of

structural equations of the RAM model:

$$
\begin{aligned}
v_1 &= 1.000v_7 + u_1 \\
v_2 &= 0.833v_7 + u_2 \\
v_3 &= 1.000v_8 + u_3 \\
v_4 &= 0.833v_8 + u_4 \\
v_5 &= 1.000v_9 + u_5 \\
v_6 &= \lambda v_9 + u_6 \\
v_7 &= \gamma_1 v_9 + u_7 \\
v_8 &= \beta v_7 + \gamma_2 v_9 + u_8 \\
v_9 &= u_9
\end{aligned}
$$

This gives the matrices **A** and **P** in the RAM model:

$$
\mathbf{A} = \begin{pmatrix}
o & o & o & o & o & o & 1.000 & o & o \\
o & o & o & o & o & o & 0.833 & o & o \\
o & o & o & o & o & o & o & 1.000 & o \\
o & o & o & o & o & o & o & 0.833 & o \\
o & o & o & o & o & o & o & o & 1.000 \\
o & o & o & o & o & o & o & o & \lambda \\
o & o & o & o & o & o & o & o & \gamma_1 \\
o & o & o & o & o & o & \beta & o & \gamma_2 \\
o & o & o & o & o & o & o & o & o
\end{pmatrix}
$$

$$
\mathbf{P} = \begin{pmatrix}
\theta_1 & o & \theta_5 & o & o & o & o & o & o \\
o & \theta_2 & o & \theta_5 & o & o & o & o & o \\
\theta_5 & o & \theta_1 & o & o & o & o & o & o \\
o & \theta_5 & o & \theta_2 & o & o & o & o & o \\
o & o & o & o & \theta_3 & o & o & o & o \\
o & o & o & o & o & \theta_4 & o & o & o \\
o & o & o & o & o & o & \psi_1 & o & o \\
o & o & o & o & o & o & o & \psi_2 & o \\
o & o & o & o & o & o & o & o & \phi
\end{pmatrix}
$$

The RAM model input specification of this example for the CALIS procedure is given in the section "RAM Model Specification" on page 842.

LISREL Model

The vector **y** contains the four endogenous manifest variables y_1 =V1, ..., y_4 =V4, and the vector **x** contains the exogenous manifest variables x_1 =V5 and x_2 =V6. The vector $\boldsymbol{\varepsilon}$ contains the error variables ε_1 =E1, ..., ε_4 =E4 corresponding to **y**, and the vector $\boldsymbol{\delta}$ contains the error variables δ_1 =E5 and δ_2 =E6 corresponding to **x**. The vector $\boldsymbol{\eta}$ contains the endogenous latent variables (factors) η_1 =F1 and η_2 =F2, while the vector $\boldsymbol{\xi}$ contains the exogenous latent variable (factor) ξ_1 =F3. The vector $\boldsymbol{\zeta}$ contains the errors ζ_1 =D1 and ζ_2 =D2 in the equations (disturbance terms)

corresponding to $\boldsymbol{\eta}$. The path diagram corresponds to the following set of structural equations of the LISREL model:

$$
\begin{aligned}
y_1 &= 1.000\eta_1 + \epsilon_1 \\
y_2 &= 0.833\eta_1 + \epsilon_2 \\
y_3 &= 1.000\eta_2 + \epsilon_3 \\
y_4 &= 0.833\eta_2 + \epsilon_4 \\
x_1 &= 1.000\xi_1 + \delta_1 \\
x_2 &= \lambda\xi_1 + \delta_2 \\
\eta_1 &= \gamma_1\xi_1 + \zeta_1 \\
\eta_2 &= \beta\eta_1 + \gamma_2\xi_1 + \zeta_2
\end{aligned}
$$

This gives the matrices $\boldsymbol{\Lambda}_y$, $\boldsymbol{\Lambda}_x$, $\mathbf{B}$, $\boldsymbol{\Gamma}$, and $\boldsymbol{\Phi}$ in the LISREL model:

$$
\boldsymbol{\Lambda}_y = \begin{pmatrix} 1.000 & o \\ 0.833 & o \\ o & 1.000 \\ o & 0.833 \end{pmatrix}, \boldsymbol{\Lambda}_x = \begin{pmatrix} 1. \\ \lambda \end{pmatrix}, \mathbf{B} = \begin{pmatrix} o & o \\ \beta & o \end{pmatrix}, \boldsymbol{\Gamma} = \begin{pmatrix} \gamma_1 \\ \gamma_2 \end{pmatrix}
$$

$$
\boldsymbol{\Theta}_\varepsilon^2 = \begin{pmatrix} \theta_1 & o & \theta_5 & o \\ o & \theta_2 & o & \theta_5 \\ \theta_5 & o & \theta_1 & o \\ o & \theta_5 & o & \theta_2 \end{pmatrix}, \boldsymbol{\Theta}_\delta^2 = \begin{pmatrix} \theta_3 & o \\ \theta_4 & o \end{pmatrix}, \boldsymbol{\Psi} = \begin{pmatrix} \psi_1 & o \\ o & \psi_2 \end{pmatrix}, \boldsymbol{\Phi} = (\phi)
$$

The CALIS procedure does not provide a LISREL model input specification. However, any model that can be specified by the LISREL model can also be specified by using the LINEQS, RAM, or COSAN model specifications in PROC CALIS.

Getting Started: CALIS Procedure

Statements in the CALIS procedure are used for various purposes:

- PROC CALIS statement invokes the procedure.
- Main model specification statements: LINEQS, RAM, FACTOR, and COSAN. These statements define the model type and specify the main model.
- Subsidiary model specification statements: STD, COV, VARNAMES, MATRIX, and PARAMETERS. These statements define additional parameters and model features that supplement to the main model specification. The usage of these statements depends on the model type specified in the main model statement. See the section "The Model Type and Related Statements" on page 840 for more details.

- Statements for constraining parameters: BOUNDS, LINCON, and NLINCON. The SAS programming statements can also be used to constrain parameters. See the section "Constrained Estimation" on page 845 for more details.
- General data processing statements: BY, VAR, PARTIAL, FREQ, and WEIGHT. The usages of these statements are common to other SAS procedures.
- Other statements: The STRUCTEQ statement is used for defining "structural equations" in the model. The NLOPTIONS statement is used for fine-tuning the optimization techniques for the analysis. The SAS programming statements are used for computing dependent parametric functions that are useful for model specification or setting up constraints in the model.

The Model Type and Related Statements

There are four sets of statements available in the CALIS procedure to specify a model. Each set represents a model type supported by the CALIS procedure. For the LINEQS model with linear equations input, you can specify the following statements:

LINEQS *analysis model in equations notation*;
  STD *variance pattern*;
  COV *covariance pattern*;
  PARAMETERS *parameter names from programming statements*;

For the RAM model, you can specify the following statements:

RAM *analysis model in list notation*;
  VARNAMES *names of latent and error variables*;
  PARAMETERS *parameter names from programming statements*;

For the (confirmatory) factor model, you can specify the following statements:

FACTOR *options*;
  MATRIX *definition of matrix elements*;
  VARNAMES *names of latent and residual variables*;
  PARAMETERS *parameter names from programming statements*;

For the COSAN model, you can specify the following statements:

COSAN *analysis model in matrix notation*;
  MATRIX *definition of matrix elements* ;
  VARNAMES *names of additional variables* ;
  PARAMETERS *parameter names from programming statements*;

You can also specify a model by using an INRAM= data set, which is usually a version of an OUTRAM= data set produced by a previous PROC CALIS analysis (and possibly modified). If no INRAM= data set is specified, you must use one of the four main model specification statements that defines the analysis model: LINEQS, RAM, FACTOR, or COSAN.

LISREL model input is not directly supported in PROC CALIS. However, with careful constructions you can use either the LINEQS, RAM, or COSAN model input to specify a LISREL model equivalently.

In the following sections, each model type supported by the CALIS procedure is discussed in more details.

LINEQS Model Specification

By using notation similar to that originally developed by Bentler for his EQS program, you can also describe the model by a set of linear equations combined with variance and covariance specifications. The displayed output can be in either equation form or matrix form.

The following statements define the structural model of the alienation example as a LINEQS model:

```
lineqs
     V1 =          F1                    + E1,
     V2 =    .833 F1                     + E2,
     V3 =          F2                    + E3,
     V4 =    .833 F2                     + E4,
     V5 =          F3                    + E5,
     V6 = Lamb (.5) F3                   + E6,
     F1 = Gam1(-.5) F3                   + D1,
     F2 = Beta (.5) F1 + Gam2(-.5) F3 + D2;
std
     E1-E6 = The1-The2 The1-The4 (6 * 3.),
     D1-D2 = Psi1-Psi2 (2 * 4.),
     F3    = Phi (6.) ;
cov
     E1 E3 = The5 (.2),
     E4 E2 = The5 (.2);
```

The LINEQS statement shows the equations in the section "LINEQS Model" on page 837, except that in this case the coefficients to be estimated can be followed (optionally) by the initial value to use in the optimization process. If you do not specify initial values for the parameters in a LINEQS statement, PROC CALIS tries to assign these values automatically. The endogenous variables used on the left side can be manifest variables (with names that must be defined by the input data set) or latent variables (which must have names starting with F). The variables used on the right side can be manifest variables, latent variables (with names that must start with an F), or error variables (which must have names starting with an E or D). Commas separate the equations. The coefficients to be estimated are indicated by names. If no name is used, the coefficient is constant, either equal to a specified number or, if no number is used, equal to 1. The VAR statement in Bentler's notation is replaced here by the STD statement, because the VAR statement in PROC CALIS defines the subset of manifest variables in the data set to be analyzed. The variable names used in the STD or COV statement must be exogenous (that is, they should not occur on the left side of any equation). The STD and COV statements define the diagonal and off-diagonal elements in the $\mathbf{\Phi}$ matrix. The parameter specifications in the STD and COV statements are separated by commas. Using k variable names on the left of an equal sign in a COV statement means that the parameter list on the right side refers to all $k(k-1)/2$ distinct variable pairs in the $\mathbf{\Phi}$ matrix. Identical coefficient names indicate parameters constrained to be equal. You can also use prefix names to specify those parameters for which you do not need a precise name in any parameter constraint.

See the section "LINEQS Model Statement" on page 881 for more information about the precise syntax rules for a LINEQS statement.

RAM Model Specification

The RAM model enables a path diagram to be transcribed into a RAM statement in list form. The displayed output from the RAM statement is in matrix or list form.

The following statement defines the structural model of the alienation example as a RAM model:

```
ram
     1   1  7  1.       ,
     1   2  7  .833     ,
     1   3  8  1.       ,
     1   4  8  .833     ,
     1   5  9  1.       ,
     1   6  9  .5    Lamb ,
     1   7  9  -.5   Gam1 ,
     1   8  7  .5    Beta ,
     1   8  9  -.5   Gam2 ,
     2   1  1  3.    The1 ,
     2   2  2  3.    The2 ,
     2   3  3  3.    The1 ,
     2   4  4  3.    The2 ,
     2   5  5  3.    The3 ,
     2   6  6  3.    The4 ,
     2   1  3  .2    The5 ,
     2   2  4  .2    The5 ,
     2   7  7  4.    Psi1 ,
     2   8  8  4.    Psi2 ,
     2   9  9  6.    Phi ;
```

You must assign numbers to the nodes in the path diagram. In the path diagram of Figure 25.1, the boxes corresponding to the six manifest variables V1, ..., V6 are assigned the number of the variable in the covariance matrix (1, ..., 6); the circles corresponding to the three latent variables F1, F2, and F3 are given the numbers 7, 8, and 9. The path diagram contains 20 paths between the nine nodes; nine of the paths are one-headed arrows and eleven are two-headed arrows.

The RAM statement contains a list of items separated by commas. Each item corresponds to an arrow in the path diagram. The first entry in each item is the number of arrow heads (matrix number), the second entry shows where the arrow points to (row number), the third entry shows where the arrow comes from (column number), the fourth entry gives the (initial) value of the coefficient, and the fifth entry assigns a name if the path represents a parameter rather than a constant. If you specify the fifth entry as a parameter name, then the fourth list entry can be omitted, since PROC CALIS tries to assign an initial value to this parameter.

See the section "RAM Model Statement" on page 899 for more information about the RAM statement.

COSAN Model Specification

You specify the model for a generalized COSAN analysis with a COSAN statement and one or more MATRIX statements. The COSAN statement determines the name, dimension, and type (identity, diagonal, symmetric, upper, lower, general, inverse, and so forth) of each matrix in the model. You can specify the values of the constant elements in each matrix and give names and initial values to

the elements that are to be estimated as parameters or functions of parameters by using MATRIX statements. The resulting displayed output is in matrix form.

The following statements define the structural model of the alienation example as a COSAN model:

```
cosan J(9, Ide) * A(9, Gen, Imi) * P(9, Sym);
matrix A
          [ ,7] = 1. .833  5 * 0. Beta (.5) ,
          [ ,8] = 2 * 0.  1.  .833 ,
          [ ,9] = 4 * 0.  1.  Lamb Gam1-Gam2 (.5 2 * -.5);
matrix P
          [1,1] = The1-The2 The1-The4 (6 * 3.) ,
          [7,7] = Psi1-Psi2 Phi (2 * 4. 6.) ,
          [3,1] = The5 (.2) ,
          [4,2] = The5 (.2) ;
```

The matrix model specified in the COSAN statement is the RAM model

$$\mathbf{C} = \mathbf{J}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{P}((\mathbf{I} - \mathbf{A})^{-1})'\mathbf{J}'$$

with selection matrix **J** and

$$\mathbf{C} = \mathcal{E}\{\mathbf{J}\mathbf{v}\mathbf{v}'\mathbf{J}'\}, \qquad \mathbf{P} = \mathcal{E}\{\mathbf{u}\mathbf{u}'\}$$

The COSAN statement must contain only the matrices up to the central matrix **P** because of the symmetry of each matrix term in a COSAN model. Each matrix name is followed by one to three arguments in parentheses. The first argument is the number of columns. The second and third arguments are optional, and they specify the form of the matrix. The selection matrix **J** in the RAM model is specified by the 6×9 identity (IDE) (sub)matrix **J** because the first six variables in vector **v** correspond to the six manifest variables in the data set. The 9×9 parameter matrix **A** has a general (GEN) form and is used as $(\mathbf{I} - \mathbf{A})^{-1}$ in the analysis, as indicated by the identity-minus-inverse (IMI) argument. The central 9×9 matrix P is specified as a symmetric (SYM) matrix.

The MATRIX statement for matrix **A** specifies the values in columns 7, 8, and 9, which correspond to the three latent variables F1, F2, and F3, in accordance with the RAM model. The other columns of **A** are assumed to be zero. The initial values for the parameter elements in **A** are chosen as in the path diagram to be

$$\lambda = \beta = 0.5, \qquad \gamma_1 = \gamma_2 = -0.5$$

In accordance with matrix **P** of the RAM model and the path model, the nine diagonal elements of matrix **P** are parameters with initial values

$$\theta_1 = \theta_2 = \theta_3 = \theta_4 = 3, \quad \psi_1 = \psi_2 = 4, \quad \phi = 6$$

There are also two off-diagonal elements in each triangle of **P** that are constrained to be equal, and they have an initial value of 0.2.

See the section "COSAN Model Statement" on page 873 for more information about the COSAN statement.

FACTOR Model Specification

You can specify the FACTOR statement to compute factor loadings **F** and unique variances **U** of an exploratory or confirmatory first-order factor (or component) analysis. By default, the factor correlation matrix **P** is an identity matrix.

$$\mathbf{C} = \mathbf{F}\mathbf{F}' + \mathbf{U}, \quad \mathbf{U} = diag$$

For a first-order confirmatory factor analysis, you can use MATRIX statements to define elements in the matrices **F**, **P**, and **U** of the more general model

$$\mathbf{C} = \mathbf{F}\mathbf{P}\mathbf{F}' + \mathbf{U}, \quad \mathbf{P} = \mathbf{P}', \quad \mathbf{U} = diag$$

To perform a component analysis, specify the COMPONENT option to constrain the matrix **U** to a zero matrix; that is, the model is replaced by

$$\mathbf{C} = \mathbf{F}\mathbf{F}'$$

Note that the rank of $\mathbf{F}\mathbf{F}'$ is equal to the number m of components in **F**, and if m is smaller than the number of variables in the moment matrix **C**, the matrix of predicted model values is singular and maximum likelihood estimates for **F** cannot be computed. You should compute ULS estimates in this case.

The HEYWOOD option constrains the diagonal elements of **U** to be nonnegative; that is, the model is replaced by

$$\mathbf{C} = \mathbf{F}\mathbf{F}' + \mathbf{U}^2, \quad \mathbf{U} = diag$$

If the factor loadings are unconstrained, they can be orthogonally rotated by one of the following methods:

- principal axes rotation
- quartimax
- varimax
- equamax
- parsimax

The most common approach to factor analysis consists of two steps:

1. Obtain estimates for factor loadings and unique variances.
2. Apply an orthogonal or oblique rotation method.

PROC CALIS enables you to specify general linear and nonlinear equality and inequality constraints by using the LINCON and NLINCON statements. You can specify the NLINCON statement to estimate orthogonal or oblique rotated factor loadings; refer to Browne and Du Toit (1992). Unlike PROC FACTOR, PROC CALIS currently does not provide standard errors for the rotated factor loadings.

PROC CALIS also computes the factor score regression coefficients for the final factor solution.

For default (exploratory) factor analysis, PROC CALIS computes initial estimates. If you use a MATRIX statement together with a FACTOR model specification, initial values are generally computed by McDonald's (McDonald and Hartmann 1992) method or are set by the START= option. See the section "FACTOR Model Statement" on page 877 and Example 25.3 for more information about the FACTOR statement.

Constrained Estimation

- Simple equality constraints, $x_i = c_i$, $c_i = const$, and $x_i = x_j$, can be defined in each model by specifying constants or using the same name for parameters constrained to be equal.
- BOUNDS statement: You can specify boundary constraints, $l_i \le x_i \le u_i$, l_i, $u_i = const$, with the BOUNDS statement for the COSAN, LINEQS, and RAM models and in connection with an INRAM= data set. There might be serious convergence problems if negative values appear in the diagonal locations (variances) of the central model matrices during the minimization process. You can use the BOUNDS statement to constrain these parameters to have nonnegative values.
- LINCON statement: You can specify general linear equality and inequality constraints of the parameter estimates with the LINCON statement or by using an INEST= data set. The variables listed in the LINCON statements must be (a subset of) the model parameters. All optimization methods can be used with linear constraints.
- NLINCON statement: You can specify general nonlinear equality and inequality constraints of the parameter estimates with the NLINCON statement. The syntax of the NLINCON statement is almost the same as that for the BOUNDS statement with the exception that the BOUNDS statement can contain only names of the model parameters. However, the variables listed in the NLINCON statement can be defined by programming statements. Only the quasi-Newton optimization method can be used when there are nonlinear constraints.
- Reparameterizing the model: Complex linear equality and inequality constraints can be defined by means of programming statements similar to those used in the DATA step. In this case, some of the parameters x_i are not elements of the matrices $\mathbf{G}_{kj}$ and $\mathbf{Q}_k$ but are instead defined in a PARAMETERS statement. Elements of the model matrices can then be computed by programming statements as functions of parameters in the PARAMETERS statement. This approach is similar to the classical COSAN program of McDonald, implemented by Fraser (McDonald 1978, 1980). One advantage of the CALIS procedure is that you need not supply code for the derivatives of the specified functions. The analytic derivatives of the user-written functions are computed automatically by PROC CALIS. The specified functions must be continuous and have continuous first-order partial derivatives. See the section "SAS Programming Statements" on page 903 and the section "Constrained Estimation by

Using Program Code" on page 955 for more information about imposing linear and nonlinear restrictions on parameters by using programming statements.

Although much effort has been made to implement reliable and numerically stable optimization methods, no practical algorithm exists that can always find the global optimum of a nonlinear function, especially when there are nonlinear constraints.

Syntax: CALIS Procedure

The following statements are available in PROC CALIS:

PROC CALIS < *options* > ;
LINEQS *model equations* ;
STD *variance pattern* ;
COV *covariance pattern* ;
RAM *model list* ;
VARNAMES *variables* ;
FACTOR < *options* > ;
COSAN *matrix model* ;
MATRIX *matrix elements* ;
PARAMETERS *parameters* ;
STRUCTEQ *variables* ;
BOUNDS *boundary constraints* ;
LINCON *linear constraints* ;
NLINCON *nonlinear constraints* ;
NLOPTIONS *optimization options* ;
BY *variables* ;
VAR *variables* ;
PARTIAL *variables* ;
FREQ *variable* ;
WEIGHT *variable* ;
SAS programming statements ;

- If no INRAM= data set is specified, one of the four statements that define the input form of the analysis model, LINEQS, RAM, FACTOR, or COSAN, must be used.
- The MATRIX statement can be used multiple times for the same or different matrices along with a COSAN or FACTOR statement. If the MATRIX statement is used multiple times for the same matrix, later definitions override earlier ones.
- The STD and COV statements can be used only with the LINEQS model statement.
- You can formulate a generalized COSAN model by using a COSAN statement. MATRIX statements can be used to define the elements of a matrix used in the COSAN statement. The input notation resembles the COSAN program of McDonald and Fraser (McDonald 1978, 1980).

- The RAM statement uses a simple list input that is especially suitable for describing McArdle's RAM analysis model (McArdle 1980; McArdle and McDonald 1984) for causal and path analysis problems.
- The LINEQS statement formulates the analysis model by means of a system of linear equations similar to Bentler's (1989) EQS program notation. The STD and COV statements can be used to define the variances and covariances corresponding to elements of matrix $\mathbf{\Phi}$ in the LINEQS model.
- A FACTOR statement can be used to compute a first-order exploratory or confirmatory factor (or component) analysis. The analysis of a simple exploratory factor analysis model performed by PROC CALIS is not as efficient as one performed by the FACTOR procedure. The CALIS procedure is designed for more general structural problems, and it needs significantly more computation time for a simple unrestricted factor or component analysis than PROC FACTOR does.
- You can add SAS programming statements to impose linear or nonlinear constraints on the parameters if you specify the model by means of a COSAN, LINEQS, or RAM statement. The PARAMETERS statement defines additional parameters that are needed as independent variables in your program code and that belong to the set of parameters to be estimated. Variable names used in the program code should differ from the preceding statement names. The code should respect the syntax rules of SAS statements usually used in the DATA step. See the section "SAS Programming Statements" on page 903 for more information.
- The BOUNDS statement can be used to specify simple lower and upper boundary constraints for the parameters.
- You can specify general linear equality and inequality constraints with the LINCON statement (or via an INEST= data set). The NLINCON statement can be used to specify general nonlinear equality and inequality constraints by referring to nonlinear functions defined by programming statements.
- The VAR, PARTIAL, WEIGHT, FREQ, and BY statements can be used in the same way as in other procedures, such as the FACTOR or PRINCOMP procedure. You can select a subset of the input variables to analyze with the VAR statement. The PARTIAL statement defines a set of input variables that are chosen as partial variables for the analysis of a matrix of partial correlations or covariances. The BY statement specifies groups in which separate covariance structure analyses are performed.

The following sections describe the PROC CALIS statement and then describe the other statements in alphabetical order.

PROC CALIS Statement

PROC CALIS < *options* > **;**

The options available with the PROC CALIS statement are listed in the following table and then are described in alphabetical order.

Table 25.1 Options Available in the PROC CALIS Statement

Option	Description
Data Set Options	
DATA=	specifies the input data set
INEST=	inputs the initial values and constraints
INRAM=	inputs the model specifications
INWGT=	specifies the input weight matrix
OUTEST=	outputs the covariance matrix of estimates
OUTJAC	outputs the Jacobian into the OUTEST= data set
OUTRAM=	outputs the model specifications
OUTSTAT=	outputs the statistical results
OUTWGT=	outputs the weight matrix
Data Processing	
AUGMENT	analyzes augmented moment matrix
COVARIANCE	analyzes covariance matrix
EDF=	defines the number of observations by the error degrees of freedom
NOBS=	defines the number of observations
NOINT	analyzes uncorrected moments
RDF=	specifies regression degrees of freedom for modifying the number of observations
RIDGE	specifies ridge factor for the covariance matrix
UCORR	analyzes uncorrected correlation matrix
UCOV	analyzes uncorrected covariance matrix
VARDEF=	specifies the method for computing the variance divisor
Estimation Methods	
ASYCOV=	specifies the formula for computing asymptotic covariances
DFREDUCE=	reduces the degrees of freedom for model fit chi-square test
G4=	specifies the algorithm for computing standard errors
METHOD=	specifies the estimation method
NODIAG	excludes the diagonal elements of the covariance matrix from model fitting
WPENALTY=	specifies the penalty weight to fit correlations
WRIDGE=	specifies the ridge factor for weight matrix
Statistical Analysis	
ALPHAECV=	specifies the α level for computing the confidence interval of ECV (Browne and Cudeck 1993)

Table 25.1 *continued*

Option	Description
ALPHARMS=	specifies the α level for computing the confidence interval of RMSEA (Steiger and Lind 1980)
BIASKUR	computes the skewness and kurtosis without bias corrections
KURTOSIS	computes and displays kurtosis
MODIFICATION	computes modification indices
NOMOD	suppresses modification indices
NOSTDERR	suppresses standard error computations
PCOVES	displays the covariance matrix of estimates
PDETERM	computes the determination coefficients
PLATCOV	computes the latent variable covariances and score coefficients
PREDET	displays predetermined moment matrix
RESIDUAL=	specifies the type of residuals being computed
SIMPLE	prints univariate statistics
SLMW=	specifies the probability limit for Wald tests
STDERR	computes the standard errors
TOTEFF	displays total and indirect effects
ODS Graphics	
PLOTS=	specifies ODS Graphics selection
Control Display Output	
NOPRINT	suppresses the displayed output
PALL	displays all output (ALL)
PCORR	displays analyzed and estimated moment matrix
PESTIM	prints parameter estimates
PINITIAL	prints initial pattern and values
PJACPAT	displays structure of variable and constant elements of the Jacobian matrix
PRIMAT	displays output in matrix form
PRINT	adds default displayed output
PRIVEC	displays output in vector form
PSHORT	reduces default output (SHORT)
PSUMMARY	displays fit summary only (SUMMARY)
PWEIGHT	displays the weight matrix
Optimization Techniques	
FCONV=	specifies the objective function convergence criterion
GCONV=	specifies the gradient convergence criterion
INSTEP=	specifies the initial step length (RADIUS=, SALPHA=)
LINESEARCH=	specifies the line-search method
LSPRECISION=	specifies the line-search precision (SPRECISION=)
MAXFUNC=	specifies the maximum number of function calls
MAXITER=	specifies the maximum number of iterations
TECHNIQUE=	specifies the minimization method
UPDATE=	specifies the update technique

Table 25.1 *continued*

Option	Description
Numerical Properties	
ASINGULAR=	specifies the absolute singularity information matrix
COVSING=	specifies the singularity tolerance of information matrix
MSINGULAR=	specifies the relative M singularity of information matrix
SINGULAR=	specifies the singularity criterion
VSINGULAR=	specifies the relative V singularity of information matrix
Miscellaneous	
DEMPHAS=	emphasizes the diagonal entries
FDCODE	uses numeric derivatives for programming code
HESSALG=	specifies the algorithm for computing the Hessian
NOADJDF	requests no degrees of freedom adjustment be made for active constraints
RANDOM=	specifies the seed for randomly generated initial values
START=	specifies the constant for the initial values

Options for General Output Display

Some output display control options enable or disable more than a single display. These are general output display control options, which include the PALL, PRINT, PSHORT, PSUMMARY, and NOPRINT options. If the NOPRINT option is not specified, a default set of output is displayed. The PRINT and PALL options add more output to the default set of output, while the PSHORT and PSUMMARY options reduce from the set of default output.

The relationships of the general output display control options (excluding the NOPRINT option) with other specific options or displays are summarized in the following table.

Output Options	PALL	PRINT	default	PSHORT	PSUMMARY
fit indices	*	*	*	*	*
linear dependencies	*	*	*	*	*
iteration history	*	*	*	*	
model matrices	*	*	*	*	
PESTIM	*	*	*	*	
PREDET	*	(*)	(*)	(*)	
PINITIAL	*	*	*		
SIMPLE	*	*	*		
STDERR	*	*	*		
RESIDUAL	*	*			
KURTOSIS	*	*			
PLATCOV	*	*			
TOTEFF	*	*			
PCORR	*				
MODIFICATION	*				
PWEIGHT	*				
PCOVES					
PDETERM					
PJACPAT					
PRIMAT					
PRIVEC					

Each "*" in the table represents a specific display or a set of displays enabled by the general output display control option. For example, if you specify the PSUMMARY option, the fit indices and the linear dependencies of parameter estimates (if present) will be shown in the printed output. With the PSHORT option, iteration history and model matrices will also be printed. In addition, the PESTIM option and the PREDET option are also enabled by the PSHORT option. Entries with "(*)" represent specific conditions that enable the PREDET option. See the PREDET option for more details.

Alphabetical Listing of Options with the PROC CALIS Statement

ALPHAECV=α

specifies the significance level for a $1-\alpha$ confidence interval, $0 \leq \alpha \leq 1$, for the Browne and Cudeck (1993) expected cross validation index (ECVI). The default value is $\alpha = 0.1$, which corresponds to a 90% confidence interval for the ECVI.

ALPHARMS=α

specifies the significance level for a $1-\alpha$ confidence interval, $0 \leq \alpha \leq 1$, for the Steiger and Lind (1980) root mean squared error of approximation (RMSEA) coefficient (refer to Browne and Du Toit 1992). The default value is $\alpha = 0.1$, which corresponds to a 90% confidence interval for the RMSEA.

ASINGULAR | ASING=*r*

specifies an absolute singularity criterion r, $r > 0$, for the inversion of the information matrix, which is needed to compute the covariance matrix. The following singularity criterion is used:

$$|d_{j,j}| \leq \max(ASING, VSING * |H_{j,j}|, MSING * \max(|H_{1,1}|, \ldots, |H_{n,n}|))$$

In the preceding criterion, $d_{j,j}$ is the diagonal pivot of the matrix, and *VSING* and *MSING* are the specified values of the VSINGULAR= and MSINGULAR= options. The default value for *ASING* is the square root of the smallest positive double-precision value. Note that, in many cases, a normalized matrix $\mathbf{D}^{-1}\mathbf{H}\mathbf{D}^{-1}$ is decomposed, and the singularity criteria are modified correspondingly.

ASYCOV | ASC=*name*

specifies the formula for asymptotic covariances used in the weight matrix $\mathbf{W}$ for WLS and DWLS estimation. The ASYCOV option is effective only if METHOD=WLS or METHOD=DWLS and no INWGT= input data set is specified. The following formulas are implemented:

BIASED — Browne's (1984) formula (3.4)
biased asymptotic covariance estimates; the resulting weight matrix is at least positive semidefinite. This is the default for analyzing a covariance matrix.

UNBIASED — Browne's (1984) formula (3.8)
asymptotic covariance estimates corrected for bias; the resulting weight matrix can be indefinite (that is, can have negative eigenvalues), especially for small N.

CORR — Browne and Shapiro's (1986) formula (3.2), which is identical to DeLeeuw's (1983) formulas (2, 3, 4)
the asymptotic variances of the diagonal elements are set to the reciprocal of the value r specified by the WPENALTY= option (default: $r = 100$). This formula is the default for analyzing a correlation matrix.

CAUTION: Using the WLS and DWLS methods with the ASYCOV=CORR option means that you are fitting a correlation (rather than a covariance) structure. Since the fixed diagonal of a correlation matrix for some models does not contribute to the model's degrees of freedom, you can specify the DFREDUCE=*i* option to reduce the degrees of freedom by the number of manifest variables used in the model. See the section "Counting the Degrees of Freedom" on page 956 for more information.

AUGMENT | AUG

analyzes the augmented correlation or covariance matrix. Using the AUG option is equivalent to specifying UCORR (NOINT but not COV) or UCOV (NOINT and COV) for a data set that

is augmented by an intercept variable INTERCEPT that has constant values equal to 1. The variable INTERCEP can be used instead of the default INTERCEPT only if you specify the SAS option OPTIONS VALIDVARNAME=V6. The dimension of an augmented matrix is one higher than that of the corresponding correlation or covariance matrix. The AUGMENT option is effective only if the data set does not contain a variable called INTERCEPT and if you specify the UCOV, UCORR, or NOINT option.

CAUTION: The INTERCEPT variable is included in the moment matrix as the variable with number $n + 1$. Using the RAM model statement assumes that the first n variable numbers correspond to the n manifest variables in the input data set. Therefore, specifying the AUGMENT option assumes that the numbers of the latent variables used in the RAM or path model have to start with number $n + 2$.

BIASKUR

computes univariate skewness and kurtosis by formulas uncorrected for bias. See the section "Measures of Multivariate Kurtosis" on page 938 for more information.

COVARIANCE | COV

analyzes the covariance matrix instead of the correlation matrix. By default, PROC CALIS (like the FACTOR procedure) analyzes a correlation matrix. If the DATA= input data set is a valid TYPE=CORR data set (containing a correlation matrix and standard deviations), using the COV option means that the covariance matrix is computed and analyzed.

COVSING=*r*

specifies a nonnegative threshold r, which determines whether the eigenvalues of the information matrix are considered to be zero. If the inverse of the information matrix is found to be singular (depending on the VSINGULAR=, MSINGULAR=, ASINGULAR=, or SINGULAR= option), a generalized inverse is computed using the eigenvalue decomposition of the singular matrix. Those eigenvalues smaller than r are considered to be zero. If a generalized inverse is computed and you do not specify the NOPRINT option, the distribution of eigenvalues is displayed.

DATA=*SAS-data-set*

specifies an input data set that can be an ordinary SAS data set or a specially structured TYPE=CORR, TYPE=COV, TYPE=UCORR, TYPE=UCOV, TYPE=SSCP, or TYPE=FACTOR SAS data set, as described in the section "Input Data Sets" on page 909. If the DATA= option is omitted, the most recently created SAS data set is used.

DEMPHAS | DE=*r*

changes the initial values of all parameters that are located on the diagonals of the central model matrices by the relationship

$$diag_{new} = r(|diag_{old}| + 1)$$

The initial values of the diagonal elements of the central matrices should always be nonnegative to generate positive-definite predicted model matrices in the first iteration. By using values of $r > 1$, such as $r = 2, r = 10, \ldots$, you can increase these initial values to produce predicted model matrices with high positive eigenvalues in the first iteration. The DEMPHAS= option is effective independently of the way the initial values are set; that is, it changes the initial values set in the model specification as well as those set by an INRAM= data set and

those automatically generated for RAM, LINEQS, or FACTOR model statements. It also affects the initial values set by the START= option, which uses, by default, DEMPHAS=100 if a covariance matrix is analyzed and DEMPHAS=10 for a correlation matrix.

DFREDUCE | DFRED=*i*

reduces the degrees of freedom of the χ^2 test by i. In general, the number of degrees of freedom is the number of elements of the lower triangle of the predicted model matrix $\mathbf{C}$, $n(n+1)/2$, minus the number of parameters, t. If the NODIAG option is used, the number of degrees of freedom is additionally reduced by n. Because negative values of i are allowed, you can also increase the number of degrees of freedom by using this option. If the DFREDUCE= or NODIAG option is used in a correlation structure analysis, PROC CALIS does not additionally reduce the degrees of freedom by the number of constant elements in the diagonal of the predicted model matrix, which is otherwise done automatically. See the section "Counting the Degrees of Freedom" on page 956 for more information.

EDF | DFE=*n*

makes the effective number of observations $n + i$, where i is 0 if the NOINT, UCORR, or UCOV option is specified without the AUGMENT option or where i is 1 otherwise. You can also use the NOBS= option to specify the number of observations.

FCONV | FTOL=*r*

specifies the relative function convergence criterion. The optimization process is terminated when the relative difference of the function values of two consecutive iterations is smaller than the specified value of r; that is,

$$\frac{|f(x^{(k)}) - f(x^{(k-1)})|}{\max(|f(x^{(k-1)})|, FSIZE)} \leq r$$

where $FSIZE$ can be defined by the FSIZE= option in the NLOPTIONS statement.

The default value is $r = 10^{-FDIGITS}$, where $FDIGITS$ either can be specified in the NLOPTIONS statement or is set by default to $-\log_{10}(\epsilon)$, where ϵ is the machine precision.

FDCODE

replaces the analytic derivatives of the programming statements by numeric derivatives (finite-difference approximations). In general, this option is needed only when you have programming statements that are too difficult for the built-in function compiler to differentiate analytically. For example, if the program code for the nonlinear constraints contains many arrays and many DO loops with array processing, the built-in function compiler can require too much time and memory to compute derivatives of the constraints with respect to the parameters. In this case, the Jacobian matrix of constraints is computed numerically by using finite-difference approximations. The FDCODE option does not modify the kind of derivatives specified with the HESSALG= option.

G4=*i*

specifies the algorithm to compute the approximate covariance matrix of parameter estimates used for computing the approximate standard errors and modification indices when the information matrix is singular. If the number of parameters t used in the model you analyze is smaller than the value of i, the time-expensive Moore-Penrose (G4) inverse of the singular information matrix is computed by eigenvalue decomposition. Otherwise, an inexpensive

pseudo (G1) inverse is computed by sweeping. By default, $i = 60$. For more details, see the section "Estimation Criteria" on page 923.

GCONV | GTOL=*r*

specifies the relative gradient convergence criterion (see the ABSGCONV= option on page 892 for the absolute gradient convergence criterion).

Termination of all techniques (except the CONGRA technique) requires the normalized predicted function reduction to be small,

$$\frac{[g(x^{(k)})]'[\mathbf{G}^{(k)}]^{-1}g(x^{(k)})}{\max(|f(x^{(k)})|, FSIZE)} \leq r$$

where $FSIZE$ can be defined by the FSIZE= option in the NLOPTIONS statement. For the CONGRA technique (where a reliable Hessian estimate $\mathbf{G}$ is not available),

$$\frac{\| g(x^{(k)}) \|_2^2 \quad \| s(x^{(k)}) \|_2}{\| g(x^{(k)}) - g(x^{(k-1)}) \|_2 \max(|f(x^{(k)})|, FSIZE)} \leq r$$

is used. The default value is $r = 10^{-8}$.

Note that prior to SAS 6.11, the GCONV= option specified the absolute gradient convergence criterion.

HESSALG | HA = 1 | 2 | 3 | 4 | 5 | 6 | 11

specifies the algorithm used to compute the (approximate) Hessian matrix when TECHNIQUE=LEVMAR and NEWRAP, to compute approximate standard errors of the parameter estimates, and to compute Lagrange multipliers. There are different groups of algorithms available:

- analytic formulas: HA=*1,2,3,4,11*
- finite-difference approximation: HA=*5,6*
- dense storage: HA=*1,2,3,4,5,6*
- sparse storage: HA=*11*

If the Jacobian is more than 25% dense, the dense analytic algorithm, HA= 1, is used by default. The HA= 1 algorithm is faster than the other dense algorithms, but it needs considerably more memory for large problems than HA= 2,3,4. If the Jacobian is more than 75% sparse, the sparse analytic algorithm, HA= 11, is used by default. The dense analytic algorithm HA= 4 corresponds to the original COSAN algorithm; you are advised not to specify HA= 4 due to its very slow performance. If there is not enough memory available for the dense analytic algorithm HA= 1 and you must specify HA= 2 or HA= 3, it might be more efficient to use one of the quasi-Newton or conjugate-gradient optimization techniques since Levenberg-Marquardt and Newton-Raphson optimization techniques need to compute the Hessian matrix in each iteration. For approximate standard errors and modification indices, the Hessian matrix has to be computed at least once, regardless of the optimization technique.

The algorithms HA= 5 and HA= 6 compute approximate derivatives by using forward-difference formulas. The HA= 5 algorithm corresponds to the analytic HA= 1: it is faster

than HA= 6, but it needs much more memory. The HA= 6 algorithm corresponds to the analytic HA= 2: it is slower than HA= 5, but it needs much less memory.

Test computations of large sparse problems show that the sparse algorithm HA= 11 can be up to 10 times faster than HA= 1 (and needs much less memory).

INEST | INVAR | ESTDATA=*SAS-data-set*

specifies an input data set that contains initial estimates for the parameters used in the optimization process and can also contain boundary and general linear constraints on the parameters. If the model did not change too much, you can specify an OUTEST= data set from a previous PROC CALIS analysis. The initial estimates are taken from the values of the PARMS observation.

INRAM=*SAS-data-set*

specifies an input data set that contains in RAM list form all information needed to specify an analysis model. The INRAM= data set is described in the section "Input Data Sets" on page 909. Typically, this input data set is an OUTRAM= data set (possibly modified) from a previous PROC CALIS analysis. If you use an INRAM= data set to specify the analysis model, you cannot use the model specification statement COSAN, MATRIX, RAM, LINEQS, STD, COV, FACTOR, or VARNAMES, but you can use the BOUNDS and PARAMETERS statements and programming statements. If the INRAM= option is omitted, you must define the analysis model with a COSAN, RAM, LINEQS, or FACTOR statement.

INSTEP=*r*

For highly nonlinear objective functions, such as the EXP function, the default initial radius of the trust-region algorithms TRUREG, DBLDOG, and LEVMAR or the default step length of the line-search algorithms can produce arithmetic overflows. If this occurs, specify decreasing values of $0 < r < 1$ such as INSTEP=1E−1, INSTEP=1E−2, INSTEP=1E−4, ..., until the iteration starts successfully.

- For trust-region algorithms (TRUREG, DBLDOG, and LEVMAR), the INSTEP option specifies a positive factor for the initial radius of the trust-region. The default initial trust-region radius is the length of the scaled gradient, and it corresponds to the default radius factor of $r = 1$.
- For line-search algorithms (NEWRAP, CONGRA, and QUANEW), INSTEP specifies an upper bound for the initial step length for the line search during the first five iterations. The default initial step length is $r = 1$.

For releases prior to SAS 6.11, specify the SALPHA= and RADIUS= options. For more details, see the section "Computational Problems" on page 958.

INWGT=*SAS-data-set*

specifies an input data set that contains the weight matrix **W** used in generalized least squares (GLS), weighted least squares (WLS, ADF), or diagonally weighted least squares (DWLS) estimation. If the weight matrix **W** defined by an INWGT= data set is not positive-definite, it can be ridged by using the WRIDGE= option. See the section "Estimation Criteria" on page 923 for more information. If no INWGT= data set is specified, default settings for the weight matrices are used in the estimation process. The INWGT= data set is described in the section "Input Data Sets" on page 909. Typically, this input data set is an OUTWGT= data set from a previous PROC CALIS analysis.

KURTOSIS | KU

computes and displays univariate kurtosis and skewness, various coefficients of multivariate kurtosis, and the numbers of observations that contribute most to the normalized multivariate kurtosis. See the section "Measures of Multivariate Kurtosis" on page 938 for more information. Using the KURTOSIS option implies the SIMPLE display option. This information is computed only if the DATA= data set is a raw data set, and it is displayed by default if the PRINT option is specified. The multivariate LS kappa and the multivariate mean kappa are displayed only if you specify METHOD=WLS and the weight matrix is computed from an input raw data set. All measures of skewness and kurtosis are corrected for the mean. If an intercept variable is included in the analysis, the measures of multivariate kurtosis do not include the intercept variable in the corrected covariance matrix, as indicated by a displayed message. Using the BIASKUR option displays the biased values of univariate skewness and kurtosis.

LINESEARCH | LIS | SMETHOD | SM=*i*

specifies the line-search method for the CONGRA, QUANEW, and NEWRAP optimization techniques. Refer to Fletcher (1980) for an introduction to line-search techniques. The value of i can be $1, \ldots, 8$; the default is $i = 2$.

LIS=1 specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is similar to one used by the Harwell subroutine library.

LIS=2 specifies a line-search method that needs more function calls than gradient calls for quadratic and cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the LSPRECISION= option.

LIS=3 specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the LSPRECISION= option.

LIS=4 specifies a line-search method that needs the same number of function and gradient calls for stepwise extrapolation and cubic interpolation.

LIS=5 specifies a line-search method that is a modified version of LIS=4.

LIS=6 specifies golden section line search (Polak 1971), which uses only function values for linear approximation.

LIS=7 specifies bisection line search (Polak 1971), which uses only function values for linear approximation.

LIS=8 specifies Armijo line-search technique (Polak 1971), which uses only function values for linear approximation.

LSPRECISION | LSP=*r*

SPRECISION | SP=*r*

specifies the degree of accuracy that should be obtained by the line-search algorithms LIS=2 and LIS=3. Usually an imprecise line search is inexpensive and successful. For more difficult optimization problems, a more precise and more expensive line search might be necessary

(Fletcher 1980, p. 22). The second (default for NEWRAP, QUANEW, and CONGRA) and third line-search methods approach exact line search for small LSPRECISION= values. If you have numerical problems, you should decrease the LSPRECISION= value to obtain a more precise line search. The default LSPRECISION= values are displayed in the following table.

TECH=	UPDATE=	LSP Default
QUANEW	DBFGS, BFGS	$r = 0.4$
QUANEW	DDFP, DFP	$r = 0.06$
CONGRA	all	$r = 0.1$
NEWRAP	no update	$r = 0.9$

For more details, refer to Fletcher (1980, pp. 25–29).

MAXFUNC | MAXFU=*i*

specifies the maximum number i of function calls in the optimization process. The default values are displayed in the following table.

TECH=	MAXFUNC Default
LEVMAR, NEWRAP, NRRIDG, TRUREG	i=125
DBLDOG, QUANEW	i=500
CONGRA	i=1000

The default is used if you specify MAXFUNC=0. The optimization can be terminated only after completing a full iteration. Therefore, the number of function calls that are actually performed can exceed the number that is specified by the MAXFUNC= option.

MAXITER | MAXIT=*i* < *n* >

specifies the maximum number i of iterations in the optimization process. The default values are displayed in the following table.

TECH=	MAXITER Default
LEVMAR, NEWRAP, NRRIDG, TRUREG	i=50
DBLDOG, QUANEW	i=200
CONGRA	i=400

The default is used if you specify MAXITER=0 or if you omit the MAXITER option.

The optional second value n is valid only for TECH=QUANEW with nonlinear constraints. It specifies an upper bound n for the number of iterations of an algorithm and reduces the violation of nonlinear constraints at a starting point. The default is n=20. For example, specifying

```
maxiter= . 0
```

means that you do not want to exceed the default number of iterations during the main optimization process and that you want to suppress the feasible-point algorithm for nonlinear constraints.

METHOD | MET=*name*

specifies the method of parameter estimation. Valid values for *name* are as follows:

ML | M | MAX	performs normal-theory maximum likelihood parameter estimation. The ML method requires a nonsingular covariance or correlation matrix. This is the default method.
GLS | G	performs generalized least squares parameter estimation. If no INWGT= data set is specified, the GLS method uses the inverse sample covariance or correlation matrix as weight matrix **W**. Therefore, METHOD=GLS requires a nonsingular covariance or correlation matrix.
WLS | W | ADF	performs weighted least squares parameter estimation. If no INWGT= data set is specified, the WLS method uses the inverse matrix of estimated asymptotic covariances of the sample covariance or correlation matrix as the weight matrix **W**. In this case, the WLS estimation method is equivalent to Browne's (1982, 1984) asymptotically distribution-free estimation. The WLS method requires a nonsingular weight matrix.
DWLS | D	performs diagonally weighted least squares parameter estimation. If no INWGT= data set is specified, the DWLS method uses the inverse diagonal matrix of asymptotic variances of the input sample covariance or correlation matrix as the weight matrix **W**. The DWLS method requires a nonsingular diagonal weight matrix.
ULS | LS | U	performs unweighted least squares parameter estimation.
LSML | LSM | LSMAX	performs unweighted least squares followed by normal-theory maximum likelihood parameter estimation.
LSGLS | LSG	performs unweighted least squares followed by generalized least squares parameter estimation.
LSWLS | LSW | LSADF	performs unweighted least squares followed by weighted least squares parameter estimation.
LSDWLS | LSD	performs unweighted least squares followed by diagonally weighted least squares parameter estimation.
NONE | NO	uses no estimation method. This option is suitable for checking the validity of the input information and for displaying the model matrices and initial values.

The default estimation method is maximum likelihood (METHOD=ML), assuming a multivariate normal distribution of the observed variables. The two-stage estimation methods METHOD=LSML, METHOD=LSGLS, METHOD=LSWLS, and METHOD=LSDWLS first compute unweighted least squares estimates of the model parameters and their residuals. Afterward, these estimates are used as initial values for the optimization process to compute maximum likelihood, generalized least squares, weighted least squares, or diagonally weighted least squares parameter estimates. You can do the same thing by using an OUTRAM= data set with least squares estimates as an INRAM= data set for a further analysis

to obtain the second set of parameter estimates. This strategy is also discussed in the section "Use of Optimization Techniques" on page 945. For more details, see the section "Estimation Criteria" on page 923.

MODIFICATION | MOD

computes and displays Lagrange multiplier test indices for constant parameter constraints, equality parameter constraints, and active boundary constraints, as well as univariate and multivariate Wald test indices. The modification indices are not computed in the case of unweighted or diagonally weighted least squares estimation.

The Lagrange multiplier test (Bentler 1986; Lee 1985; Buse 1982) provides an estimate of the χ^2 reduction that results from dropping the constraint. For constant parameter constraints and active boundary constraints, the approximate change of the parameter value is displayed also. You can use this value to obtain an initial value if the parameter is allowed to vary in a modified model. For more information, see the section "Modification Indices" on page 953.

MSINGULAR | MSING=*r*

specifies a relative singularity criterion r, $r > 0$, for the inversion of the information matrix, which is needed to compute the covariance matrix. The following singularity criterion is used:

$$|d_{j,j}| \leq \max(ASING, VSING * |H_{j,j}|, MSING * \max(|H_{1,1}|, \ldots, |H_{n,n}|))$$

where $d_{j,j}$ is the diagonal pivot of the matrix, and *ASING* and *VSING* are the specified values of the ASINGULAR= and VSINGULAR= options, respectively. If you do not specify the SINGULAR= option, the default value for *MSING* is 1E−12; otherwise, the default value is 1E−4 * SINGULAR. Note that, in many cases, a normalized matrix $\mathbf{D}^{-1}\mathbf{H}\mathbf{D}^{-1}$ is decomposed, and the singularity criteria are modified correspondingly.

NOADJDF

turns off the automatic adjustment of degrees of freedom when there are active constraints in the analysis. When the adjustment is in effect, most fit statistics and the associated probability levels will be affected. This option should be used when the researcher believes that the active constraints observed in the current sample will have little chance to occur in repeated sampling.

NOBS=*nobs*

specifies the number of observations. If the DATA= input data set is a raw data set, *nobs* is defined by default to be the number of observations in the raw data set. The NOBS= and EDF= options override this default definition. You can use the RDF= option to modify the *nobs* specification. If the DATA= input data set contains a covariance, correlation, or scalar product matrix, you can specify the number of observations either by using the NOBS=, EDF=, and RDF= options in the PROC CALIS statement or by including a _TYPE_='N' observation in the DATA= input data set.

NODIAG | NODI

omits the diagonal elements of the analyzed correlation or covariance matrix from the fit function. This option is useful only for special models with constant error variables. The NODIAG option does not allow fitting of those parameters that contribute to the diagonal of the estimated moment matrix. The degrees of freedom are automatically reduced by n. A

simple example for the usefulness of the NODIAG option is the fit of the first-order factor model, $\mathbf{S} = \mathbf{FF}' + \mathbf{U}^2$. In this case, you do not have to estimate the diagonal matrix of unique variances $\mathbf{U}^2$ that are fully determined by $diag(\mathbf{S} - \mathbf{FF}')$.

NOINT

specifies that no intercept be used in computing covariances and correlations; that is, covariances or correlations are not corrected for the mean. You can specify this option (or UCOV or UCORR) to analyze mean structures in an uncorrected moment matrix—that is, to compute intercepts in systems of structured linear equations (see Example 25.2). The term NOINT is misleading in this case because an uncorrected covariance or correlation matrix is analyzed containing a constant (intercept) variable that is used in the analysis model. The degrees of freedom used in the variance divisor (specified by the VARDEF= option) and some of the assessment of the fit function (see the section "Assessment of Fit" on page 928) depend on whether an intercept variable is included in the model (the intercept is used in computing the corrected covariance or correlation matrix or is used as a variable in the uncorrected covariance or correlation matrix to estimate mean structures) or not included (an uncorrected covariance or correlation matrix is used that does not contain a constant variable).

NOMOD

does not compute modification indices. The NOMOD option is useful in connection with the PALL option because it saves computing time.

NOPRINT | NOP

suppresses the displayed output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 20, "Using the Output Delivery System."

NOSTDERR | NOSE

specifies that standard errors should not be computed. Standard errors are not computed for unweighted least squares (ULS) or diagonally weighted least squares (DWLS) estimation. In general, standard errors are computed even if the STDERR display option is not used (for file output).

OUTEST | OUTVAR=*SAS-data-set*

creates an output data set containing the parameter estimates, their gradient, Hessian matrix, and boundary and linear constraints. For METHOD=ML, METHOD=GLS, and METHOD=WLS, the OUTEST= data set also contains the information matrix, the approximate covariance matrix of the parameter estimates ((generalized) inverse of information matrix), and approximate standard errors. If linear or nonlinear equality or active inequality constraints are present, the Lagrange multiplier estimates of the active constraints, the projected Hessian, and the Hessian of the Lagrange function are written to the data set. The OUTEST= data set also contains the Jacobian if the OUTJAC option is used.

The OUTEST= data set is described in the section "OUTEST= SAS-data-set" on page 913. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to the chapter titled "SAS Data Files" in *SAS Language Reference: Concepts* for more information about permanent data sets.

OUTJAC
writes the Jacobian matrix, if it has been computed, to the OUTEST= data set. This is useful when the information and Jacobian matrices need to be computed for other analyses.

OUTRAM=*SAS-data-set*
creates an output data set containing the model information for the analysis, the parameter estimates, and their standard errors. An OUTRAM= data set can be used as an input INRAM= data set in a subsequent analysis by PROC CALIS. The OUTRAM= data set also contains a set of fit indices; it is described in more detail in the section "OUTRAM= SAS-data-set" on page 917. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to the chapter titled "SAS Data Files" in *SAS Language Reference: Concepts* for more information about permanent data sets.

OUTSTAT=*SAS-data-set*
creates an output data set containing the BY-group variables, the analyzed covariance or correlation matrices, and the predicted and residual covariance or correlation matrices of the analysis. You can specify the correlation or covariance matrix in an OUTSTAT= data set as an input DATA= data set in a subsequent analysis by PROC CALIS. The OUTSTAT= data set is described in the section "OUTSTAT= SAS-data-set" on page 920. If the model contains latent variables, this data set also contains the predicted covariances between latent and manifest variables and the latent variable score regression coefficients (see the PLATCOV option on page 864). If the FACTOR statement is used, the OUTSTAT= data set also contains the rotated and unrotated factor loadings, the unique variances, the matrix of factor correlations, the transformation matrix of the rotation, and the matrix of standardized factor loadings.

You can use the latent variable score regression coefficients with PROC SCORE to compute latent variable or factor scores. For details, see the section "Latent Variable Scores" on page 937.

If you want to create a permanent SAS data set, you must specify a two-level name. Refer to the chapter titled "SAS Data Files" in *SAS Language Reference: Concepts* for more information about permanent data sets.

OUTWGT=*SAS-data-set*
creates an output data set containing the weight matrix $\mathbf{W}$ used in the estimation process. You cannot create an OUTWGT= data set with an unweighted least squares or maximum likelihood estimation. The fit function in GLS, WLS (ADF), and DWLS estimation contains the inverse of the (Cholesky factor of the) weight matrix $\mathbf{W}$ written in the OUTWGT= data set. The OUTWGT= data set contains the weight matrix to which the WRIDGE= and the WPENALTY= options are applied. An OUTWGT= data set can be used as an input INWGT= data set in a subsequent analysis by PROC CALIS. The OUTWGT= data set is described in the section "OUTWGT= SAS-data-set" on page 922. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to the chapter titled "SAS Data Files" in *SAS Language Reference: Concepts* for more information about permanent data sets.

PALL | ALL

displays all optional output except the output generated by the PCOVES, PDETERM, PJACPAT, and PRIVEC options.

CAUTION: The PALL option includes the very expensive computation of the modification indices. If you do not really need modification indices, you can save computing time by specifying the NOMOD option in addition to the PALL option.

PCORR | CORR

displays the (corrected or uncorrected) covariance or correlation matrix that is analyzed and the predicted model covariance or correlation matrix.

PCOVES | PCE

displays the following:

- the information matrix (crossproduct Jacobian)
- the approximate covariance matrix of the parameter estimates (generalized inverse of the information matrix)
- the approximate correlation matrix of the parameter estimates

The covariance matrix of the parameter estimates is not computed for the ULS and DWLS estimation methods. This displayed output is not included in the output generated by the PALL option.

PDETERM | PDE

displays three coefficients of determination: the determination of all equations (DETAE), the determination of the structural equations (DETSE), and the determination of the manifest variable equations (DETMV). These determination coefficients are intended to be global means of the squared multiple correlations for different subsets of model equations and variables. The coefficients are displayed only when you specify a RAM or LINEQS model, but they are displayed for all five estimation methods: ULS, GLS, ML, WLS, and DWLS.

You can use the STRUCTEQ statement to define which equations are structural equations. If you do not use the STRUCTEQ statement, PROC CALIS uses its own default definition to identify structural equations.

The term "structural equation" is not defined in a unique way. The LISREL program defines the structural equations by the user-defined BETA matrix. In PROC CALIS, the default definition of a structural equation is an equation that has a dependent left-side variable that appears at least once on the right side of another equation, or an equation that has at least one right-side variable that is the left-side variable of another equation. Therefore, PROC CALIS sometimes identifies more equations as structural equations than the LISREL program does.

If the model contains structural equations, PROC CALIS also displays the "Stability Coefficient of Reciprocal Causation"—that is, the largest eigenvalue of the $\mathbf{BB}'$ matrix, where $\mathbf{B}$ is the causal coefficient matrix of the structural equations. These coefficients are computed as in the LISREL VI program of Jöreskog and Sörbom (1985). This displayed output is not included in the output generated by the PALL option.

PESTIM | PES
displays the parameter estimates. In some cases, this includes displaying the standard errors and t values.

PINITIAL | PIN
displays the input model matrices and the vector of initial values.

PJACPAT | PJP
displays the structure of variable and constant elements of the Jacobian matrix. This displayed output is not included in the output generated by the PALL option.

PLATCOV | PLC
displays the following:

- the estimates of the covariances among the latent variables
- the estimates of the covariances between latent and manifest variables
- the latent variable score regression coefficients

The estimated covariances between latent and manifest variables and the latent variable score regression coefficients are written to the OUTSTAT= data set. You can use the score coefficients with PROC SCORE to compute factor scores. For details, see the section "Latent Variable Scores" on page 937.

PLOTS <(*global-plot-options*)> = *plot-request* <(*options*)>
PLOTS <(*global-plot-options*)> = (*plot-request* <(*options*)> <... *plot-request* <(*options*)> >)
specifies the ODS graphical plots. Currently, the only available ODS graphical plots in PROC CALIS are for residual distributions. Also, when the residual histograms are requested, the bar charts of residual tallies are suppressed. To display these bar charts with the residual histograms, you must use the RESIDUAL(TALLY) option.

When you specify only one *plot-request*, you can omit the parentheses around the *plot-request*. For example:

```
plots=all
plots=residuals
```

You must enable ODS Graphics before requesting plots; for example:

```
ods graphics on;

proc calis plots=residuals;
run;

ods graphics off;
```

For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

The following table shows the available *plot-requests*:

Plot-request	Plot Description
ALL	all available plots
NONE	no ODS graphical plots
RESIDUALS	distribution of residuals

PREDET | PRE

displays the pattern of variable and constant elements of the predicted moment matrix that is predetermined by the analysis model. It is especially helpful in finding manifest variables that are not used or that are used as exogenous variables in a complex model specified in the COSAN statement. Those entries of the predicted moment matrix for which the model generates variable (rather than constant) elements are displayed as missing values. This output is displayed even without specifying the PREDET option if the model generates constant elements in the predicted model matrix different from those in the analysis moment matrix and if you specify at least the PSHORT amount of displayed output.

If the analyzed matrix is a correlation matrix (containing constant elements of 1s in the diagonal) and the model generates a predicted model matrix with q constant (rather than variable) elements in the diagonal, the degrees of freedom are automatically reduced by q. The output generated by the PREDET option displays those constant diagonal positions. If you specify the DFREDUCE= or NODIAG option, this automatic reduction of the degrees of freedom is suppressed. See the section "Counting the Degrees of Freedom" on page 956 for more information.

PRIMAT | PMAT

displays parameter estimates, approximate standard errors, and t values in matrix form if you specify the analysis model in the RAM or LINEQS statement. When a COSAN statement is used, this occurs by default.

PRINT | PRI

adds the options KURTOSIS, RESIDUAL, PLATCOV, and TOTEFF to the default output.

PRIVEC | PVEC

displays parameter estimates, approximate standard errors, the gradient, and t values in vector form. The values are displayed with more decimal places. This displayed output is not included in the output generated by the PALL option.

PSHORT | SHORT | PSH

excludes the output produced by the PINITIAL, SIMPLE, and STDERR options from the default output.

PSUMMARY | SUMMARY | PSUM

displays the fit assessment table and the ERROR, WARNING, and NOTE messages.

PWEIGHT | PW

displays the weight matrix $\mathbf{W}$ used in the estimation. The weight matrix is displayed after the WRIDGE= and WPENALTY= options are applied to it.

RADIUS=*r*

is an alias for the INSTEP= option for Levenberg-Marquardt minimization.

RANDOM =*i*

specifies a positive integer as a seed value for the pseudo-random number generator to generate initial values for the parameter estimates for which no other initial value assignments in the model definitions are made. Except for the parameters in the diagonal locations of the central matrices in the model, the initial values are set to random numbers in the range

$0 \leq r \leq 1$. The values for parameters in the diagonals of the central matrices are random numbers multiplied by 10 or 100. For more information, see the section "Initial Estimates" on page 941.

RDF | DFR=*n*

makes the effective number of observations the actual number of observations minus the RDF= value. The degree of freedom for the intercept should not be included in the RDF= option. If you use PROC CALIS to compute a regression model, you can specify RDF= *number-of-regressor-variables* to get approximate standard errors equal to those computed by PROC REG.

RESIDUAL | RES <(TALLY | TALLIES)> <= NORM | VARSTAND | ASYSTAND>

displays the raw and normalized residual covariance matrices, the rank order of the largest residuals, and the bar charts of residual tallies. This information is displayed by default when you specify the PRINT option.

Three types of normalized or standardized residual matrices can be chosen with the RESIDUAL= specification:

RESIDUAL= NORM	normalized residuals
RESIDUAL= VARSTAND | VARSTD	variance standardized residuals
RESIDUAL= ASYSTAND | ASYMSTD	asymptotically standardized residuals

When ODS graphical plots of residuals are also requested, the bar charts of residual tallies are suppressed. They are replaced with high-quality graphical histograms showing residual distributions. If you still want to display the bar charts in this situation, use the RESIDUAL(TALLY) or RESIDUAL(TALLY)= option.

For more details, see the section "Assessment of Fit" on page 928.

RIDGE<=*r***>**

defines a ridge factor r for the diagonal of the moment matrix $\mathbf{S}$ that is analyzed. The matrix $\mathbf{S}$ is transformed to

$$\mathbf{S} \longrightarrow \tilde{\mathbf{S}} = \mathbf{S} + r(diag(\mathbf{S}))$$

If you do not specify r in the RIDGE option, PROC CALIS tries to ridge the moment matrix $\mathbf{S}$ so that the smallest eigenvalue is about 10^{-3}.

CAUTION: The moment matrix in the OUTSTAT= output data set does not contain the ridged diagonal.

SALPHA=*r*

is an alias for the INSTEP= option for line-search algorithms.

SIMPLE | S

displays means, standard deviations, skewness, and univariate kurtosis if available. This information is displayed when you specify the PRINT option. If you specify the UCOV, UCORR, or NOINT option, the standard deviations are not corrected for the mean. If the KURTOSIS option is specified, the SIMPLE option is set by default.

SINGULAR | SING=*r*

specifies the singularity criterion r, $0 < r < 1$, used, for example, for matrix inversion. The default value is the square root of the relative machine precision or, equivalently, the square root of the largest double-precision value that, when added to 1, results in 1.

SLMW=*r*

specifies the probability limit used for computing the stepwise multivariate Wald test. The process stops when the univariate probability is smaller than r. The default value is $r = 0.05$.

SPRECISION | SP=*r*

is an alias for the LSPRECISION= option.

START=*r*

In general, this option is needed only in connection with the COSAN model statement, and it specifies a constant r as an initial value for all the parameter estimates for which no other initial value assignments in the pattern definitions are made. Start values in the diagonal locations of the central matrices are set to $100|r|$ if a COV or UCOV matrix is analyzed and $10|r|$ if a CORR or UCORR matrix is analyzed. The default value is $r = 0.5$. Unspecified initial values in a FACTOR, RAM, or LINEQS model are usually computed by PROC CALIS. If none of the initialization methods are able to compute all starting values for a model specified by a FACTOR, RAM, or LINEQS statement, then the start values of parameters that could not be computed are set to r, $10|r|$, or $100|r|$. If the DEMPHAS= option is used, the initial values of the diagonal elements of the central model matrices are multiplied by the value specified in the DEMPHAS= option. For more information, see the section "Initial Estimates" on page 941.

STDERR | SE

displays approximate standard errors if estimation methods other than unweighted least squares (ULS) or diagonally weighted least squares (DWLS) are used (and the NOSTDERR option is not specified). If you specify neither the STDERR nor the NOSTDERR option, the standard errors are computed for the OUTRAM= data set. This information is displayed by default when you specify the PRINT option.

TECHNIQUE | TECH=*name*

OMETHOD | OM=*name*

specifies the optimization technique. Valid values for *name* are as follows:

CONGRA | CG chooses one of four different conjugate-gradient optimization algorithms, which can be more precisely defined with the UPDATE= option and modified with the LINESEARCH= option. The conjugate-gradient techniques need only $O(t)$ memory compared to the $O(t^2)$ memory for the other three techniques, where t is the number of parameters. On the other hand, the conjugate-gradient techniques are significantly slower than other optimization techniques and should be used only when memory is insufficient for more efficient techniques. When you choose this option, UPDATE=PB by default. This is the default optimization technique if there are more than 400 parameters to estimate.

DBLDOG | DD performs a version of double-dogleg optimization, which uses the gradient to update an approximation of the Cholesky factor of the Hessian. This technique is, in many ways, very similar to the dual quasi-Newton method, but it does not use line search. The implementation is based on Dennis and Mei (1979) and Gay (1983).

LEVMAR | LM | MARQUARDT performs a highly stable but, for large problems, memory- and time-consuming Levenberg-Marquardt optimization technique, a slightly improved variant of the Moré (1978) implementation. This is the default optimization technique if there are fewer than 40 parameters to estimate.

NEWRAP | NRA performs a usually stable but, for large problems, memory- and time-consuming Newton-Raphson optimization technique. The algorithm combines a line-search algorithm with ridging, and it can be modified with the LINESEARCH= option. Prior to SAS 6.11, this option invokes the NRRIDG option.

NRRIDG | NRR | NR | NEWTON performs a usually stable but, for large problems, memory- and time-consuming Newton-Raphson optimization technique. This algorithm does not perform a line search. Since TECH=NRRIDG uses an orthogonal decomposition of the approximate Hessian, each iteration of TECH=NRRIDG can be slower than that of TECH=NEWRAP, which works with Cholesky decomposition. However, usually TECH=NRRIDG needs fewer iterations than TECH=NEWRAP.

QUANEW | QN chooses one of four different quasi-Newton optimization algorithms that can be more precisely defined with the UPDATE= option and modified with the LINESEARCH= option. If boundary constraints are used, these techniques sometimes converge slowly. When you choose this option, UPDATE=DBFGS by default. If nonlinear constraints are specified in the NLINCON statement, a modification of Powell's (1982a, 1982b) VMCWD algorithm is used, which is a sequential quadratic programming (SQP) method. This algorithm can be modified by specifying VERSION=1, which replaces the update of the Lagrange multiplier estimate vector μ to the original update of Powell (1978a, 1978b) that is used in the VF02AD algorithm. This can be helpful for applications with linearly dependent active constraints. The QUANEW technique is the default optimization technique if there are nonlinear constraints specified or if there are more than 40 and fewer than 400 parameters to estimate. The QUANEW algorithm uses only first-order derivatives of the objective function and, if available, of the nonlinear constraint functions.

TRUREG | TR performs a usually very stable but, for large problems, memory- and time-consuming trust-region optimization technique. The algorithm is implemented similar to Gay (1983) and Moré and Sorensen (1983).

NONE | NO does not perform any optimization. This option is similar to METHOD=NONE, but TECH=NONE also computes and displays residuals and goodness of fit statistics. If you

specify METHOD=ML, METHOD=LSML, METHOD=GLS, METHOD=LSGLS, METHOD=WLS, or METHOD=LSWLS, this option enables computing and displaying (if the display options are specified) of the standard error estimates and modification indices corresponding to the input parameter estimates.

Since there is no single nonlinear optimization algorithm available that is clearly superior (in terms of stability, speed, and memory) for all applications, different types of optimization techniques are provided in the CALIS procedure. Each technique can be modified in various ways. The default optimization technique for fewer than 40 parameters ($t < 40$) is TECHNIQUE=LEVMAR. For $40 \leq t < 400$, TECHNIQUE=QUANEW is the default method, and for $t \geq 400$, TECHNIQUE=CONGRA is the default method. For more details, see the section "Use of Optimization Techniques" on page 945. You can specify the following set of options in the PROC CALIS statement or in the NLOPTIONS statement.

TOTEFF | TE
computes and displays total effects and indirect effects.

UCORR
analyzes the uncorrected correlation matrix instead of the correlation matrix corrected for the mean. Using the UCORR option is equivalent to specifying the NOINT option but not the COV option.

UCOV
analyzes the uncorrected covariance matrix instead of the covariance matrix corrected for the mean. Using the UCOV option is equivalent to specifying both the COV and NOINT options. You can specify this option to analyze mean structures in an uncorrected covariance matrix—that is, to compute intercepts in systems of linear structural equations (see Example 25.2).

UPDATE | UPD=*name*
specifies the update method for the quasi-Newton or conjugate-gradient optimization technique.

For TECHNIQUE=CONGRA, the following updates can be used:

PB performs the automatic restart update method of Powell (1977) and Beale (1972). This is the default.

FR performs the Fletcher-Reeves update (Fletcher 1980, p. 63).

PR performs the Polak-Ribiere update (Fletcher 1980, p. 66).

CD performs a conjugate-descent update of Fletcher (1987).

For TECHNIQUE=DBLDOG, the following updates (Fletcher 1987) can be used:

DBFGS performs the dual Broyden-Fletcher-Goldfarb-Shanno (BFGS) update of the Cholesky factor of the Hessian matrix. This is the default.

DDFP performs the dual Davidon-Fletcher-Powell (DFP) update of the Cholesky factor of the Hessian matrix.

For TECHNIQUE=QUANEW, the following updates (Fletcher 1987) can be used:

BFGS performs original BFGS update of the inverse Hessian matrix. This is the default for earlier releases.

DFP performs the original DFP update of the inverse Hessian matrix.

DBFGS performs the dual BFGS update of the Cholesky factor of the Hessian matrix. This is the default.

DDFP performs the dual DFP update of the Cholesky factor of the Hessian matrix.

VARDEF= DF | N | WDF | WEIGHT | WGT

specifies the divisor used in the calculation of covariances and standard deviations. The default value is VARDEF=DF. The values and associated divisors are displayed in the following table, where $i = 0$ if the NOINT option is used and $i = 1$ otherwise and where k is the number of partial variables specified in the PARTIAL statement. Using an intercept variable in a mean structure analysis, by specifying the AUGMENT option, includes the intercept variable in the analysis. In this case, $i = 1$. When a WEIGHT statement is used, w_j is the value of the WEIGHT variable in the jth observation, and the summation is performed only over observations with positive weight.

Value	Description	Divisor
DF	degrees of freedom	$N - k - i$
N	number of observations	N
WDF	sum of weights DF	$\sum_j^N w_j - k - i$
WEIGHT | WGT	sum of weights	$\sum_j^N w_j$

VSINGULAR | VSING=*r*

specifies a relative singularity criterion r, $r > 0$, for the inversion of the information matrix, which is needed to compute the covariance matrix. The following singularity criterion is used:

$$|d_{j,j}| \leq \max(ASING, VSING * |H_{j,j}|, MSING * \max(|H_{1,1}|, \ldots, |H_{n,n}|))$$

where $d_{j,j}$ is the diagonal pivot of the matrix, and *ASING* and *MSING* are the specified values of the ASINGULAR= and MSINGULAR= options. If you do not specify the SINGULAR= option, the default value for *VSING* is 1E−8; otherwise, the default value is SINGULAR. Note that in many cases a normalized matrix $\mathbf{D}^{-1}\mathbf{H}\mathbf{D}^{-1}$ is decomposed, and the singularity criteria are modified correspondingly.

WPENALTY | WPEN=*r*

specifies the penalty weight $r \geq 0$ for the WLS and DWLS fit of the diagonal elements of a correlation matrix (constant 1s). The criterion for weighted least squares estimation of a correlation structure is

$$\mathbf{F}_{WLS} = \sum_{i=2}^{n} \sum_{j=1}^{i-1} \sum_{k=2}^{n} \sum_{l=1}^{k-1} w^{ij,kl}(s_{ij} - c_{ij})(s_{kl} - c_{kl}) + r \sum_{i}^{n} (s_{ii} - c_{ii})^2$$

where r is the penalty weight specified by the WPENALTY=r option and the $w^{ij,kl}$ are the elements of the inverse of the reduced $(n(n-1)/2) \times (n(n-1)/2)$ weight matrix that

contains only the nonzero rows and columns of the full weight matrix **W**. The second term is a penalty term to fit the diagonal elements of the correlation matrix. The default value is 100. The reciprocal of this value replaces the asymptotic variance corresponding to the diagonal elements of a correlation matrix in the weight matrix **W**, and it is effective only with the ASYCOV=CORR option. The often used value $r = 1$ seems to be too small in many cases to fit the diagonal elements of a correlation matrix properly. The default WPENALTY= value emphasizes the importance of the fit of the diagonal elements in the correlation matrix. You can decrease or increase the value of r if you want to decrease or increase the importance of the diagonal elements fit. This option is effective only with the WLS or DWLS estimation method and the analysis of a correlation matrix. See the section "Estimation Criteria" on page 923 for more details.

WRIDGE=*r*

defines a ridge factor r for the diagonal of the weight matrix **W** used in GLS, WLS, or DWLS estimation. The weight matrix **W** is transformed to

$$\mathbf{W} \longrightarrow \tilde{\mathbf{W}} = \mathbf{W} + r(diag(\mathbf{W}))$$

The WRIDGE= option is applied to the weight matrix as follows:

- before the WPENALTY= option is applied to it
- before the weight matrix is written to the OUTWGT= data set
- before the weight matrix is displayed

BOUNDS Statement

BOUNDS *constraint* < , *constraint* . . . > **;**

where *constraint* represents

< *number operator* > *parameter-list* < *operator number* >

You can use the BOUNDS statement to define boundary constraints for any parameter that has its name specified in a MATRIX, LINEQS, STD, COV, or RAM statement or that is used in the model of an INRAM= data set. Valid operators are <=, <, >=, >, and = or, equivalently, LE, LT, GE, GT, and EQ. The following is an example of the BOUNDS statement:

```
bounds         0.   <= a1-a9 x    <= 1. ,
              -1.   <= c2-c5              ,
                       b1-b10 y   >= 0. ;
```

You must separate boundary constraints with a comma, and you can specify more than one BOUNDS statement. The feasible region for a parameter is the intersection of all boundary constraints specified for that parameter; if a parameter has a maximum lower boundary constraint larger than its minimum upper bound, the parameter is set equal to the minimum of the upper bounds.

If you need to compute the values of the upper or lower bounds, create a TYPE=EST data set containing _TYPE_='UPPERBD' or _TYPE_='LOWERBD' observations and use it as an INEST= or INVAR= input data set in a later PROC CALIS run.

The BOUNDS statement can contain only parameter names and numerical constants. You cannot use the names of variables created in SAS programming statements.

The active set strategies made available in PROC CALIS cannot realize the strict inequality constraints < or >. For example, you cannot specify **BOUNDS x > 0;** to prevent infinite values for $y = log(x)$. Use **BOUNDS x > 1E-8;** instead.

If the CALIS procedure encounters negative diagonal elements in the central model matrices during the minimization process, serious convergence problems can occur. You can use the BOUNDS statement to constrain these parameters to nonnegative values. Using negative values in these locations can lead to a smaller χ^2 value but uninterpretable estimates.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC CALIS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the CALIS procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

COSAN Model Statement

COSAN *matrix_term* < + *matrix_term*. . . > **;**

where *matrix_term* represents
matrix_definition < ∗ *matrix_definition* . . . >
and *matrix_definition* represents
matrix_name (column_number < *,general_form* < *,transformation* >> *)*

The COSAN statement constructs the symmetric matrix model for the covariance analysis mentioned earlier (see the section "The Generalized COSAN Model" on page 832):

$$\mathbf{C} = \mathbf{F}_1\mathbf{P}_1\mathbf{F}_1' + \cdots + \mathbf{F}_m\mathbf{P}_m\mathbf{F}_m',$$

$$\mathbf{F}_k = \mathbf{F}_{k_1}\cdots\mathbf{F}_{k_{n(k)}}, \qquad \text{and} \quad \mathbf{P}_k = \mathbf{P}_k', \quad k = 1,\ldots,m$$

$$\mathbf{F}_{k_j} = \begin{cases} \mathbf{G}_{k_j} \\ \mathbf{G}_{k_j}^{-1} \\ (\mathbf{I}-\mathbf{G}_{k_j})^{-1} \end{cases} \quad j = 1,\ldots,n(k), \qquad \text{and} \qquad \mathbf{P}_k = \begin{cases} \mathbf{Q}_k \\ \mathbf{Q}_k^{-1} \end{cases}$$

You can specify only one COSAN statement with each PROC CALIS statement. The COSAN statement contains m *matrix_term*s corresponding to the generalized COSAN formula. The *matrix_term*s are separated by plus signs (+) according to the addition of the terms within the model.

Each *matrix_term* of the COSAN statement contains the definitions of the first $n(k)+1$ matrices, $\mathbf{F}_{k_j}$ and $\mathbf{P}_k$, separated by asterisks (*) according to the multiplication of the matrices within the term. The matrices $\mathbf{F}_k'$ of the right-hand-side product are redundant and are not specified within the COSAN statement.

Each *matrix_definition* consists of the name of the matrix (*matrix_name*), followed in parentheses by the number of columns of the matrix (*column_number*) and, optionally, one or two matrix properties, separated by commas, describing the form of the matrix.

The number of rows of the first matrix in each term is defined by the input correlation or covariance matrix. You can reorder and reduce the variables in the input moment matrix by using the VAR statement. The number of rows of the other matrices within the term is defined by the number of columns of the preceding matrix.

The first matrix property is indicated by *general_form*, which describes the general form of the matrix in the model. You can choose one of the following specifications of the first matrix property. The default first matrix property is GEN.

IDE specifies an identity matrix; if the matrix is not square, this specification describes an identity submatrix followed by a rectangular zero submatrix.

ZID specifies an identity matrix; if the matrix is not square, this specification describes a rectangular zero submatrix followed by an identity submatrix.

DIA specifies a diagonal matrix; if the matrix is not square, this specification describes a diagonal submatrix followed by a rectangular zero submatrix.

ZDI specifies a diagonal matrix; if the matrix is not square, this specification describes a rectangular zero submatrix followed by a diagonal submatrix.

LOW specifies a lower triangular matrix; the matrix can be rectangular.

UPP specifies an upper triangular matrix; the matrix can be rectangular.

SYM specifies a symmetric matrix; the matrix cannot be rectangular.

GEN specifies a general rectangular matrix (default).

The second matrix property is indicated by *transformation*, which describes the kind of inverse matrix transformation. If the second matrix property is omitted, no transformation is applied to the matrix.

INV uses the inverse of the matrix.

IMI uses the inverse of the difference between the identity and the matrix.

You cannot specify a nonsquare parameter matrix as an INV or IMI model matrix. Specifying a matrix of type DIA, ZDI, UPP, LOW, or GEN is not necessary if you do not use the *unspecified location* list in the corresponding MATRIX statements. After PROC CALIS processes the corresponding MATRIX statements, the matrix type DIA, ZDI, UPP, LOW, or GEN is recognized from the pattern of possibly nonzero elements. If you do not specify the first matrix property and you use the *unspecified location* list in a corresponding MATRIX statement, the matrix is recognized as a GEN matrix. You can also generate an IDE or ZID matrix by specifying a DIA, ZDI, or IMI matrix and by using MATRIX statements that define the pattern structure. However, PROC CALIS would be unable to take advantage of the fast algorithms that are available for IDE and ZID matrices in this case.

For example, to specify a second-order factor analysis model

$$\mathbf{S} = \mathbf{F}_1\mathbf{F}_2\mathbf{P}_2\mathbf{F}_2'\mathbf{F}_1' + \mathbf{F}_1\mathbf{U}_2^2\mathbf{F}_1' + \mathbf{U}_1^2$$

with $m_1 = 3$ first-order factors and $m_2 = 2$ second-order factors and with $n = 9$ variables, you can use the following COSAN statement:

```
cosan F1(3) * F2(2) * P2(2,SYM)+F1(3) * U2(3,DIA) * I1(3,IDE)
      +U1(9,DIA) * I2(9,IDE)
```

COV Statement

COV *assignment* < , *assignment . . .* > **;**

where *assignment* represents *variables* < ∗ *variables2* >> = *pattern-definition*

The COV statement tells which covariances are parameters to estimate and which are fixed. The COV statement can be used only with the LINEQS statement. The COV statement differs from the STD statement only in the meaning of the left-hand-side *variables* list. You can specify only one COV statement with each LINEQS statement. The COV statement defines the off-diagonal

elements of the central model matrix $\mathbf{\Phi}$. These elements correspond to the covariances of the exogenous variables and to the error covariances of the endogenous variables. Elements that are not defined are assumed to be zero. The *assignment*s in the COV statement must be separated by commas.

The *variables* list on the left-hand side of the equal sign should contain only names of variables that do not appear on the left-hand side of an equation in the LINEQS statement—that is, exogenous, error, and disturbance variables.

The *pattern-definition* on the right-hand side is similar to that used in the MATRIX statement. Each list element on the right-hand side defines the covariance of a pair of variables in the list on the left-hand side. A name on the right-hand side can be followed by a number inside parentheses that gives the initial value. A number on the right-hand side means that the corresponding covariance of the variable on the left-hand side is fixed. If the right-hand-side list is longer than the left-hand-side variable list, the right-hand-side list is shortened to the length of the variable list. If the right-hand-side list is shorter than the variable list, the right-hand-side list is filled with repetitions of the last item in the list.

You can use one of two alternatives to refer to parts of $\mathbf{\Phi}$. The first alternative uses only one variable list and refers to all distinct pairs of variables within the list. The second alternative uses two variable lists separated by an asterisk and refers to all pairs of variables among the two lists.

Within-List Covariances

Using k variable names in the *variables* list on the left-hand side of an equal sign in a COV statement means that the parameter list (*pattern-definition*) on the right-hand side refers to all $k(k-1)/2$ distinct variable pairs in the below-diagonal part of the $\mathbf{\Phi}$ matrix. Order is very important. The order relation between the left-hand-side variable pairs and the right-hand-side parameter list is illustrated by the following example:

```
COV E1-E4 = PHI1-PHI6 ;
```

This is equivalent to the following specification:

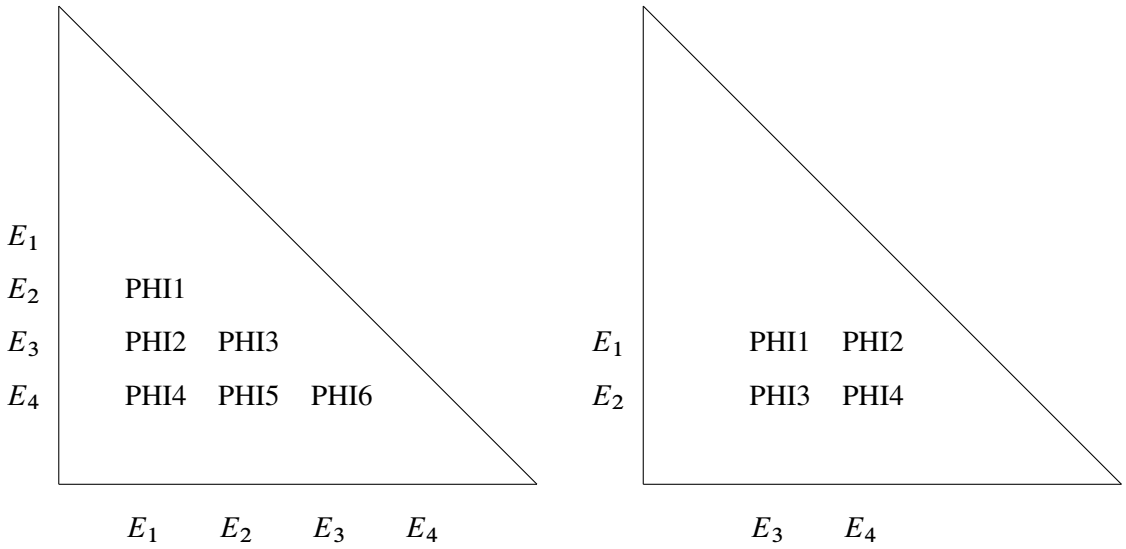
```
COV E2 E1 = PHI1,
    E3 E1 = PHI2, E3 E2 = PHI3,
    E4 E1 = PHI4, E4 E2 = PHI5, E4 E3 = PHI6;
```

The symmetric elements are generated automatically. When you use prefix names on the right-hand sides, you do not have to count the exact number of parameters. For example,

```
COV E1-E4 = PHI: ;
```

generates the same list of parameter names if the prefix PHI is not used in a previous statement. This is illustrated in the left panel of Figure 25.2. Integers starting from 1 are appended to the prefix PHI to form the parameter names for each of the within-list covariances.

Figure 25.2 Within-List and Between-List Covariances



Between-List Covariances

Using k_1 and k_2 variable names in the two lists (separated by an asterisk) on the left-hand side of an equal sign in a COV statement means that the parameter list on the right-hand side refers to all $k_1 \times k_2$ distinct variable pairs in the $\mathbf{\Phi}$ matrix. Order is very important. The order relation between the left-hand-side variable pairs and the right-hand-side parameter list is illustrated by the following example:

```
COV E1 E2 * E3 E4 = PHI1-PHI4 ;
```

This is equivalent to the following specification:

```
COV  E1 E3 = PHI1, E1 E4 = PHI2,
     E2 E3 = PHI3, E2 E4 = PHI4;
```

The symmetric elements are generated automatically. Using prefix names on the right-hand sides lets you achieve the same purpose without counting the number of parameters. That is,

```
COV  E1 E2 * E3 E4 = PHI: ;
```

This is illustrated in the right panel of Figure 25.2. Integers starting from 1 are appended to the prefix PHI to form the parameter names for each of the between-list covariances.

FACTOR Model Statement

FACTOR < *options* > **;**

You can use the FACTOR statement to specify an exploratory or confirmatory first-order factor analysis of the given covariance or correlation matrix $\mathbf{C}$,

$$\mathbf{C} = \mathbf{F}\mathbf{F}' + \mathbf{U}, \quad \mathbf{U} = diag$$

or

$$\mathbf{C} = \mathbf{F}\mathbf{P}\mathbf{F}' + \mathbf{U}, \quad \mathbf{P} = \mathbf{P}'$$

where $\mathbf{U}$ is a diagonal matrix and $\mathbf{P}$ is symmetric. Within this section, n denotes the number of manifest variables corresponding to the rows and columns of matrix $\mathbf{C}$, and m denotes the number of latent variables (factors or components) corresponding to the columns of the loading matrix $\mathbf{F}$.

You can specify only one FACTOR statement with each PROC CALIS statement. You can specify higher-order factor analysis problems by using a COSAN model specification. PROC CALIS requires more computing time and memory than PROC FACTOR because it is designed for more general structural estimation problems and is unable to exploit the special properties of the unconstrained factor analysis model.

For default (exploratory) factor analysis, PROC CALIS computes initial estimates for factor loadings and unique variances by an algebraic method of approximate factor analysis. If you use a MATRIX statement together with a FACTOR model specification, initial values are computed by McDonald's (McDonald and Hartmann 1992) method (if possible). For details, see "Using the FACTOR and MATRIX Statements" on page 879. If neither of the two methods is appropriate, the initial values are set by the START= option.

The unrestricted factor analysis model is not identified because any orthogonal rotated factor loading matrix $\tilde{\mathbf{F}} = \mathbf{F}\boldsymbol{\Theta}$ is equivalent to the result $\mathbf{F}$,

$$\mathbf{C} = \tilde{\mathbf{F}}\tilde{\mathbf{F}}' + \mathbf{U}, \qquad \tilde{\mathbf{F}} = \mathbf{F}\boldsymbol{\Theta}, \quad \text{where} \quad \boldsymbol{\Theta}'\boldsymbol{\Theta} = \boldsymbol{\Theta}\boldsymbol{\Theta}' = \mathbf{I}$$

To obtain an identified factor solution, the FACTOR statement imposes zero constraints on the $m(m-1)/2$ elements in the upper triangle of $\mathbf{F}$ by default.

The following options are available in the FACTOR statement.

COMPONENT | COMP

computes a component analysis instead of a factor analysis (the diagonal matrix $\mathbf{U}$ in the model is set to 0). Note that the rank of $\mathbf{F}\mathbf{F}'$ is equal to the number m of components in $\mathbf{F}$. If m is smaller than the number of variables in the moment matrix $\mathbf{C}$, the matrix of predicted model values is singular and maximum likelihood estimates for $\mathbf{F}$ cannot be computed. You should compute ULS estimates in this case.

HEYWOOD | HEY

constrains the diagonal elements of **U** to be nonnegative; in other words, the model is replaced by

$$\mathbf{C} = \mathbf{F}\mathbf{F}' + \mathbf{U}^2, \quad \mathbf{U} = diag$$

N=*m*

specifies the number of first-order factors or components. The number m of factors should not exceed the number n of variables in the covariance or correlation matrix analyzed. For the saturated model, $m = n$, the COMP option should generally be specified for $\mathbf{U} = 0$; otherwise, $df < 0$. For $m = 0$ no factor loadings are estimated, and the model is $\mathbf{C} = \mathbf{U}$, with $\mathbf{U} = diag$. By default, $m = 1$.

NORM < = KAISER | NONE >

Kaiser-normalizes the rows of the factor pattern for rotation. NORM=KAISER, which is the default, has exactly the same effect as NORM. You can turn off the normalization by NORM=NONE.

RCONVERGE=*p*

RCONV=*p*

specifies the convergence criterion for rotation cycles. The option is applicable to rotation by using either the QUARTIMAX, VARIMAX, EQUAMAX, or PARSIMAX method in the ROTATE= option. Rotation stops when the scaled change of the simplicity function value is less than the RCONVERGE= value. The default convergence criterion is

$$|f_{new} - f_{old}|/K < \epsilon$$

where f_{new} and f_{old} are simplicity function values of the current cycle and the previous cycle, respectively, $K = max(1, |f_{old}|)$ is a scaling factor, and ϵ is 1E–9 by default and is modified by the RCONVERGE= value.

RITER=*n*

specifies the maximum number of cycles n for factor rotation by using either the QUARTIMAX, VARIMAX, EQUAMAX, or PARSIMAX method in the ROTATE= option. The default n is the maximum between 10 times the number of variables and 100.

ROTATE | R=*name*

specifies an orthogonal rotation. By default, ROTATE=NONE. The possible values for *name* are as follows:

PRINCIPAL | PC specifies a principal axis rotation. If ROTATE=PRINCIPAL is used with a factor rather than a component model, the following rotation is performed:

$$\mathbf{F}_{new} = \mathbf{F}_{old}\mathbf{T}, \quad \text{with} \quad \mathbf{F}'_{old}\mathbf{F}_{old} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}'$$

where the columns of matrix $\mathbf{T}$ contain the eigenvectors of $\mathbf{F}'_{old}\mathbf{F}_{old}$.

QUARTIMAX | Q specifies quartimax rotation.

VARIMAX | V specifies varimax rotation.

EQUAMAX | E specifies equamax rotation.

PARSIMAX | P specifies parsimax rotation.

NONE performs no rotation (default).

Using the FACTOR and MATRIX Statements

You can specify the MATRIX statement and the FACTOR statement to compute a confirmatory first-order factor or component analysis. You can define the elements of the matrices **F**, **P**, and **U** of the oblique model,

$$\mathbf{C} = \mathbf{FPF}' + \mathbf{U}^2, \quad \mathbf{P} = \mathbf{P}', \quad \mathbf{U} = diag$$

To specify the structure for matrix **F**, **P**, or **U**, you have to refer to the matrix _F_ , _P_ , or _U_ in the MATRIX statement. Matrix names automatically set by PROC CALIS always start with an underscore. As you name your own matrices or variables, you should avoid leading underscores.

The default matrix forms are as follows:

_F_ lower triangular matrix (0 upper triangle for problem identification, removing rotational invariance)

_P_ identity matrix (constant)

_U_ diagonal matrix

For details about specifying the elements in matrices, see the section "MATRIX Statement" on page 883. If you are using at least one MATRIX statement in connection with a FACTOR model statement, you can also use the BOUNDS or PARAMETERS statement and programming statements to constrain the parameters named in the MATRIX statement. Initial estimates are computed by McDonald's (McDonald and Hartmann 1992) method. McDonald's method of computing initial values works better if you scale the factors by setting the factor variances to 1 rather than by setting the loadings of the reference variables equal to 1.

FREQ Statement

FREQ *variable* **;**

If one variable in your data set represents the frequency of occurrence for the other values in the observation, specify the variable's name in a FREQ statement. PROC CALIS then treats the data set as if each observation appears n_i times, where n_i is the value of the FREQ variable for observation i. Only the integer portion of the value is used. If the value of the FREQ variable is less than 1 or is missing, that observation is not included in the analysis. The total number of observations is considered to be the sum of the FREQ values when the procedure computes significance probabilities. You can use only one FREQ statement with each PROC CALIS statement.

LINCON Statement

LINCON *constraint* < , *constraint* ... > **;**

where *constraint* represents one of the following:

- *number operator linear-term*
- *linear-term operator number*

and *linear-term* is

< + | − >< *coefficient* * > *parameter* << + | − >< *coefficient* * > *parameter* ... >

The LINCON statement specifies a set of linear equality or inequality constraints of the form

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1, \ldots, m$$

The constraints must be separated by commas. Each linear constraint i in the statement consists of a linear combination $\sum_j a_{ij} x_j$ of a subset of the n parameters x_j, $j = 1, \ldots, n$, and a constant value b_i separated by a comparison operator. Valid operators are <=, <, >=, >, and = or, equivalently, LE, LT, GE, GT, and EQ. PROC CALIS cannot enforce the strict inequalities < or >. Note that the coefficients a_{ij} in the linear combination must be constant numbers and must be followed by an asterisk and the name of a parameter (for example, listed in the PARMS, STD, or COV statement). The following is an example of the LINCON statement that sets a linear constraint on parameters x1 and x2:

```
lincon       x1 + 3 * x2 <= 1;
```

Although you can easily express boundary constraints in LINCON statements, for many applications it is much more convenient to specify both the BOUNDS and LINCON statements in the same PROC CALIS call.

The LINCON statement can contain only parameter names, operators, and numerical constants. If you need to compute the values of the coefficients a_{ij} or right-hand sides b_i, you can run a preliminary DATA step and create a TYPE=EST data set containing _TYPE_='LE', _TYPE_='GE', or _TYPE_='EQ' observations, and then specify this data set as an INEST= or INVAR= data set in a subsequent PROC CALIS run.

LINEQS Model Statement

LINEQS *equation* < , *equation* . . . > **;**

where *equation* represents

dependent = *term* < + *term* . . . >

and each *term* represents one of the following:

- *coefficient-name* < *(number)* > *variable-name*
- *prefix-name* < *(number)* > *variable-name*
- < *number* > *variable-name*

The LINEQS statement defines the following LINEQS model:

$$\begin{aligned} \boldsymbol{\eta} &= \boldsymbol{\beta}\boldsymbol{\eta} + \boldsymbol{\gamma}\boldsymbol{\xi} \\ \mathbf{C} &= \mathbf{J}(\mathbf{I}-\mathbf{B})^{-1}\boldsymbol{\Gamma}\boldsymbol{\Phi}\boldsymbol{\Gamma}'((\mathbf{I}-\mathbf{B})^{-1})'\mathbf{J}' \end{aligned}$$

You can specify only one LINEQS statement with each PROC CALIS statement. There are some differences from Bentler's notation in choosing the variable names. The length of each variable name is restricted to eight characters. The names of the manifest variables are defined in the DATA= input data set. The VAR statement can be used to select a subset of manifest variables in the DATA= input data set to analyze. You do not need to use a V prefix for manifest variables in the LINEQS statement, nor do you need to use a numerical suffix in any variable name. The names of the latent variables must start with the prefix letter F (for Factor); the names of the residuals must start with the prefix letters E (for Error) or D (for Disturbance). The trailing part of the variable name can contain letters or digits. The prefix letter E is used for the errors of the manifest variables, and the prefix letter D is used for the disturbances of the latent variables. The names of the manifest variables in the DATA= input data set can start with F, E, or D, but these names should not coincide with the names of latent or error variables used in the model. The left-hand side (that is, endogenous *dependent* variable) of each equation should be either a manifest variable of the data set or a latent variable with prefix letter F. The left-hand-side variable should not appear on the right-hand side of the same equation; this means that matrix $\boldsymbol{\beta}$ should not have a nonzero diagonal element. Each equation should contain, at most, one E or D variable.

The equations must be separated by a comma. The order of the equations is arbitrary. The displayed output generally contains equations and terms in an order different from that of the input.

Coefficients to estimate are indicated in the equations by a name preceding the independent variable's name. The coefficient's name can be followed by a number inside parentheses indicating the initial value for this coefficient. A number preceding the independent variable's name indicates a constant coefficient. If neither a coefficient name nor a number precedes the independent variable's name, a constant coefficient of 1 is assumed.

If the initial value of a parameter is not specified in the equation, the initial value is chosen in one of the following ways:

- If you specify the RANDOM= option in the PROC CALIS statement, the variable obtains a randomly generated initial value r, such that $0 \leq r \leq 1$. The uninitialized parameters in the

diagonals of the central model matrices are given the nonnegative random values r multiplied by 10, 100, or the value specified in the DEMPHAS= option.

- If the RANDOM= option is not used, PROC CALIS tries to estimate the initial values.
- If the initial values cannot be estimated, the value of the START= option is used as an initial value.

In Bentler's notation, estimated coefficients are indicated by asterisks. Referring to a parameter in Bentler's notation requires the specification of two variable names that correspond to the row and column of the position of the parameter in the matrix. Specifying the estimated coefficients by parameter names makes it easier to impose additional constraints with code. You do not need any additional statements to express equality constraints. Simply specify the same name for parameters that should have equal values.

If your model contains many unconstrained parameters and it is too cumbersome to find different parameter names, you can specify all those parameters by the same prefix name. A prefix is a short name followed by a colon. The CALIS procedure then generates a parameter name by appending an integer suffix to this prefix name. The prefix name should have no more than five or six characters so that the generated parameter name is not longer than eight characters. To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names.

For example, consider the following model described in the section "Specifying a Second-Order Factor-Analysis Model" on page 885:

$$\mathbf{S} = \mathbf{F}_1\mathbf{F}_2\mathbf{P}_2\mathbf{F}_2'\mathbf{F}_1' + \mathbf{F}_1\mathbf{U}_2^2\mathbf{F}_1' + \mathbf{U}_1^2$$

You can fit this model by using the LINEQS and STD statements:

```
lineqs
       V1 = X1 F1 + E1,
       V2 = X2 F1 + E2,
       V3 = X3 F1 + E3,
       V4 = X4 F2 + E4,
       V5 = X5 F2 + E5,
       V6 = X6 F2 + E6,
       V7 = X7 F3 + E7,
       V8 = X8 F3 + E8,
       V9 = X9 F3 + E9,
       F1 = Y1 F4 + D1,
       F2 = Y1 F4 + Y2 F5 + D2,
       F3 = Y2 F5 + D3;

std
       E1-E9 = 9 * U:,
       D1-D3 = 3 * V:,
       F4 F5 = 2 * P;
run;
```

MATRIX Statement

MATRIX *matrix-name* < *location* > = *list* < , *location* = *list* . . . > **;**

You can specify one or more MATRIX statements with a COSAN or FACTOR statement. A MATRIX statement specifies which elements of the matrix are constant and which are parameters. You can also assign values to the constant elements and initial values for the parameters. The input notation resembles that used in the COSAN program of McDonald and Fraser (personal communication), except that in PROC CALIS, parameters are distinguished from constants by giving parameters names instead of by using positive and negative integers.

A MATRIX statement cannot be used for an IDE or ZID matrix. For all other types of matrices, each element is assumed to be a constant of 0 unless a MATRIX statement specifies otherwise. Hence, there must be at least one MATRIX statement for each matrix mentioned in the COSAN statement except for IDE and ZID matrices. There can be more than one MATRIX statement for a given matrix. If the same matrix element is given different definitions, later definitions override earlier definitions.

At the start, all elements of each model matrix, except IDE or ZID matrices, are set equal to 0.

Description of *location*:

There are several ways to specify the starting *location* and continuation direction of a *list* with $n+1$, $n \geq 0$, elements within the parameter matrix.

[*i* , *j*] The *list* elements correspond to the diagonally continued matrix elements [*i*,*j*], [*i*+1,*j*+1], . . . , [*i*+*n*,*j*+*n*]. The number of elements is defined by the length of the list and eventually terminated by the matrix boundaries. If the list contains just one element (constant or variable), then it is assigned to the matrix element [*i*,*j*].

[*i* ,] The *list* elements correspond to the horizontally continued matrix elements [*i*,*j*], [*i*,*j*+1], . . . , [*i*,*j*+*n*], where the starting column *j* is the diagonal position for a DIA, ZDI, or UPP matrix and is the first column for all other matrix types. For a SYM matrix, the list elements refer only to the matrix elements in the lower triangle. For a DIA or ZDI matrix, only one list element is accepted.

[, *j*] The *list* elements correspond to the vertically continued matrix elements [*i*,*j*], [*i*+1, *j*], . . . , [*i*+*n*,*j*], where the starting row *i* is equal to the diagonal position for a DIA, ZDI, SYM, or LOW matrix and is the first row for each other matrix type. For a SYM matrix, the list elements refer only to the matrix elements in the lower triangle. For a DIA or ZDI matrix, only one list element is accepted.

[,] unspecified location: The *list* is allocated to all valid matrix positions (except for a ZDI matrix) starting at the element [1,1] and continuing rowwise. The only valid matrix positions for a DIA or ZDI matrix are the diagonal elements; for an UPP or LOW matrix, the valid positions are the elements above or below the diagonal; and for a symmetric matrix, the valid positions are the elements in the lower triangle since the other triangle receives the symmetric allocation automatically. This *location* definition differs from the definitions with specified pattern locations in one important respect: if the number

of elements in the *list* is smaller than the number of valid matrix elements, the list is repeated in the allocation process until all valid matrix elements are filled.

Omitting the left-hand-side term is equivalent to using [,] for an *unspecified location*.

Description of *list*:

The *list* contains numeric values or parameter names, or both, that are assigned to a list of matrix elements starting at a specified position and proceeding in a specified direction. A real number r in the list defines the corresponding matrix element as a constant element with this value. The notation $n * r$ generates n values of r in the list. A name in the list defines the corresponding matrix element as a parameter to be estimated. You can use numbered name lists (X1-X10) or the asterisk notation (5 *X means five occurrences of the parameter X). If a sublist of n_1 names inside a *list* is followed by a list of $n_2 \leq n_1$ real values inside parentheses, the last n_2 parameters in the name sublist are given the initial values mentioned inside the parentheses. For example, the *list*

```
0. 1. A2-A5 (1.4 1.9 2.5) 5.
```

specifies that the first two matrix elements (specified by the *location* to the left of the equal sign) are constants with values 0 and 1. The next element is parameter A2 with no specified initial value. The next three matrix elements are the variable parameters A3, A4, and A5 with initial values 1.4, 1.9, and 2.5, respectively. The next matrix element is specified by the seventh list element to be the constant 5.

If your model contains many unconstrained parameters and it is too cumbersome to find different parameter names, you can specify all those parameters by the same prefix name. A prefix is a short name followed by a colon. The CALIS procedure generates a parameter name by appending an integer suffix to this prefix name. The prefix name should have no more than five or six characters so that the generated parameter name is not longer than eight characters. For example, if the prefix A (the parameter A1) is already used once in a *list*, the previous example would be identical to

```
0. 1. 4 * A: (1.4 1.9 2.5) 5.
```

To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names.

If you do not assign initial values to the parameters (listed in parentheses following a name sublist within the pattern list), PROC CALIS assigns initial values as follows:

- If the PROC CALIS statement contains a START=r option, each uninitialized parameter is given the initial value r. The uninitialized parameters in the diagonals of the central model matrices are given the initial value $10|r|$, $100|r|$, or $|r|$ multiplied by the value specified in the DEMPHAS= option.
- If the PROC CALIS statement contains a RANDOM=i option, each uninitialized parameter is given a random initial value $0 \leq r \leq 1$. The uninitialized parameters in the diagonals of the central model matrices are given the random values multiplied by 10, 100, or the value specified in the DEMPHAS= option.
- Otherwise, the initial value is set corresponding to START=0.5.

Specifying a Second-Order Factor-Analysis Model

For example, to specify a confirmatory second-order factor analysis model

$$\mathbf{S} = \mathbf{F}_1\mathbf{F}_2\mathbf{P}_2\mathbf{F}_2'\mathbf{F}_1' + \mathbf{F}_1\mathbf{U}_2^2\mathbf{F}_1' + \mathbf{U}_1^2$$

with $m_1 = 3$ first-order factors, $m_2 = 2$ second-order factors, and $n = 9$ variables and the following matrix pattern,

$$\mathbf{F}_1 = \begin{pmatrix} X_1 & 0 & 0 \\ X_2 & 0 & 0 \\ X_3 & 0 & 0 \\ 0 & X_4 & 0 \\ 0 & X_5 & 0 \\ 0 & X_6 & 0 \\ 0 & 0 & X_7 \\ 0 & 0 & X_8 \\ 0 & 0 & X_9 \end{pmatrix}, \quad \mathbf{U}_1 = \begin{pmatrix} U_1 & & & & & & & & \\ & U_2 & & & & & & & \\ & & U_3 & & & & & & \\ & & & U_4 & & & & & \\ & & & & U_5 & & & & \\ & & & & & U_6 & & & \\ & & & & & & U_7 & & \\ & & & & & & & U_8 & \\ & & & & & & & & U_9 \end{pmatrix}$$

$$\mathbf{F}_2 = \begin{pmatrix} Y_1 & 0 \\ Y_1 & Y_2 \\ 0 & Y_2 \end{pmatrix}, \quad \mathbf{P}_2 = \begin{pmatrix} P & 0 \\ 0 & P \end{pmatrix}, \quad \mathbf{U}_2 = \begin{pmatrix} V_1 & & \\ & V_2 & \\ & & V_3 \end{pmatrix}$$

you can specify the following COSAN and MATRIX statements:

```
cosan f1(3) * f2(2) * p2(2,dia) + f1(3) * u2(3,dia) * i1(3,ide)
      + u1(9,dia) * i2(9,ide);
matrix f1
         [ ,1]= x1-x3,
         [ ,2]= 3 * 0 x4-x6,
         [ ,3]= 6 * 0 x7-x9;
matrix u1
         [1,1]=u1-u9;
matrix f2
         [ ,1]= 2 * y1,
         [ ,2]= 0. 2 * y2;
matrix u2 = 3 * v:;
matrix p2 = 2 * p;
run;
```

The matrix pattern includes several equality constraints. Two loadings in the first and second factor of $\mathbf{F}_2$ (parameter names Y1 and Y2) and the two factor correlations in the diagonal of matrix $\mathbf{P}_2$ (parameter name P) are constrained to be equal. There are many other ways to specify the same model. See Figure 25.3 for the path diagram of this model.

The MATRIX statement can also be used with the FACTOR model statement. See "Using the FACTOR and MATRIX Statements" on page 879 for the usage.

NLINCON Statement

NLINCON | **NLC** *constraint* < , *constraint . . .* > **;**

where *constraint* represents one of the following:

- *number operator variable-list number operator*
- *variable-list operator number*
- *number operator variable-list*

You can specify nonlinear equality and inequality constraints with the NLINCON or NLC statement. The QUANEW optimization subroutine is used when you specify nonlinear constraints by using the NLINCON statement.

The syntax of the NLINCON statement is similar to that of the BOUNDS statement, except that the NLINCON statement must contain the names of variables that are defined in the programming statements and are defined as continuous functions of parameters in the model. They must not be confused with the variables in the data set.

As with the BOUNDS statement, one- or two-sided constraints are allowed in the NLINCON statement; equality constraints must be one-sided. Valid operators are <=, <, >=, >, and = or, equivalently, LE, LT, GE, GT, and EQ.

PROC CALIS cannot enforce the strict inequalities < or > but instead treats them as <= and >=, respectively. The listed nonlinear constraints must be separated by commas. The following is an example of the NLINCON statement that constrains the nonlinear parametric function $x_1 * x_1 + u_1$, which is defined in a programming statement, to a fixed value of 1:

```
nlincon    xx = 1;
xx = x1 * x1 + u1;
```

Note that x1 and u1 are parameters defined in the model. The following three NLINCON statements, which require xx1, xx2, and xx3 to be between 0 and 10, are equivalent:

```
nlincon  0. <= xx1-xx3,
               xx1-xx3 <= 10;
nlincon 0. <= xx1-xx3 <= 10.;
nlincon 10. >= xx1-xx3 >= 0.;
```

NLOPTIONS Statement

NLOPTIONS *option(s)* **;**

Many options that are available in PROC NLP can now be specified for the optimization subroutines in PROC CALIS by using the NLOPTIONS statement. The NLOPTIONS statement provides more displayed and file output on the results of the optimization process, and it permits the same set of termination criteria as in PROC NLP. These are more technical options that you might not need to

specify in most cases. The available options are summarized in Table 25.2 through Table 25.4, and the options are described in detail in the following three sections.

Table 25.2 Options Documented in the PROC CALIS Statement

Option	Description
Estimation Methods	
G4=*i*	specifies the algorithm for computing standard errors
Optimization Techniques	
FCONV=*r*	specifies the relative change function convergence criterion
GCONV=*r*	specifies the relative gradient convergence criterion
INSTEP=*r*	specifies the initial step length (SALPHA=, RADIUS=)
LINESEARCH=*i*	specifies the line-search method
LSPRECISION=*r*	specifies the line-search precision
MAXFUNC=*i*	specifies the maximum number of function calls
MAXITER=*i* <*n*>	specifies the maximum number of iterations
TECHNIQUE=*name*	specifies the minimization method
UPDATE=*name*	specifies the update technique
Miscellaneous Options	
ASINGULAR=*r*	specifies the absolute singularity criterion for inversion of the information matrix
COVSING=*r*	specifies the singularity tolerance of the information matrix
MSINGULAR=*r*	specifies the relative M singularity criterion for inversion of the information matrix
SINGULAR=*r*	specifies the singularity criterion for inversion of the Hessian
VSINGULAR=*r*	specifies the relative V singularity criterion for inversion of the information matrix

Table 25.3 Termination Criteria Options

Option	Description
Options Used by All Techniques	
ABSCONV=*r*	specifies the absolute function convergence criterion
MAXFUNC=*i*	specifies the maximum number of function calls
MAXITER=*i* <*n*>	specifies the maximum number of iterations
MAXTIME=*r*	specifies the maximum CPU time
MINITER=*i*	specifies the minimum number of iterations
Options for Unconstrained and Linearly Constrained Techniques	
ABSFCONV=*r* <*n*>	specifies the absolute change function convergence criterion
ABSGCONV=*r* <*n*>	specifies the absolute gradient convergence criterion
ABSXCONV=*r* <*n*>	specifies the absolute change parameter convergence criterion
FCONV=*r* <*n*>	specifies the relative change function convergence criterion
FCONV2=*r* <*n*>	specifies the function convergence criterion

Table 25.3 *continued*

Option	Description
FDIGITS=*r*	specifies the precision in computation of the objective function
FSIZE=*r*	specifies the parameter for FCONV= and GCONV=
GCONV=*r* <*n*>	specifies the relative gradient convergence criterion
GCONV2=*r* <*n*>	specifies the relative gradient convergence criterion
XCONV=*r* <*n*>	specifies the relative change parameter convergence criterion
XSIZE=*r*	specifies the parameter for XCONV=
Options for Nonlinearly Constrained Techniques	
ABSGCONV=*r* <*n*>	specifies the maximum absolute gradient of Lagrange function criterion
FCONV2=*r* <*n*>	specifies the predicted objective function reduction criterion
GCONV=*r* <*n*>	specifies the normalized predicted objective function reduction criterion

Table 25.4 Miscellaneous Options

Option	Description
Options for the Approximate Covariance Matrix of Parameter Estimates	
CFACTOR=*r*	specifies the scalar factor for standard errors
NOHLF	uses the Hessian of the objective function for standard errors
Options for Additional Displayed Output	
PALL	displays the initial and final optimization values
PCRPJAC	displays the approximate Hessian matrix
PHESSIAN	displays the Hessian matrix
PHISTORY	displays the optimization history
PINIT	displays the initial values and derivatives
PNLCJAC	displays the Jacobian matrix of nonlinear constraints
PRINT	displays the results of the optimization process
Additional Options for Optimization Techniques	
DAMPSTEP< =*r* >	controls the initial line-search step size
HESCAL=*n*	specifies the scaling version of Hessian or Jacobian
LCDEACT=*r*	specifies the Lagrange multiplier threshold of constraint
LCEPSILON=*r*	specifies the range for boundary and linear constraints
LCSINGULAR=*r*	specifies the QR decomposition linear dependence criterion
NOEIGNUM	suppresses the computation of matrices
RESTART=*i*	specifies the restart algorithm with a steepest descent direction
VERSION=1 | 2	specifies the quasi-Newton optimization technique version

Options Documented in the PROC CALIS Statement

The following options are the same as in the PROC CALIS statement and are documented in the section "PROC CALIS Statement" on page 848.

Estimation Method Option

G4=*i*

specifies the method for computing the generalized (G2 or G4) inverse of a singular matrix needed for the approximate covariance matrix of parameter estimates. This option is valid only for applications where the approximate covariance matrix of parameter estimates is found to be singular.

Optimization Technique Options

FCONV | FTOL=*r*

specifies the relative function convergence criterion. For more details, see the section "Termination Criteria Options" on page 890.

GCONV | GTOL=*r*

specifies the relative gradient convergence criterion. For more details, see the section "Termination Criteria Options" on page 890.

INSTEP | SALPHA | RADIUS=*r*

restricts the step length of an optimization algorithm during the first iterations.

LINESEARCH | LIS=*i*

specifies the line-search method for the CONGRA, QUANEW, and NEWRAP optimization techniques.

LSPRECISION | LSP=*r*

specifies the degree of accuracy that should be obtained by the line-search algorithms LIS=2 and LIS=3.

MAXFUNC | MAXFU=*i*

specifies the maximum number i of function calls in the optimization process. For more details, see the section "Termination Criteria Options" on page 890.

MAXITER | MAXIT=*i* < *n* >

specifies the maximum number i of iterations in the optimization process. For more details, see the section "Termination Criteria Options" on page 890.

TECHNIQUE | TECH=*name*

OMETHOD | OM=*name*

specifies the optimization technique.

UPDATE | UPD=*name*

specifies the update method for the quasi-Newton or conjugate-gradient optimization technique.

Miscellaneous Options

ASINGULAR | ASING=*r*
specifies an absolute singularity criterion r, $r > 0$, for the inversion of the information matrix, which is needed to compute the approximate covariance matrix of parameter estimates.

COVSING=*r*
specifies a nonnegative threshold r, $r > 0$, that decides whether the eigenvalues of the information matrix are considered to be zero. This option is valid only for applications where the approximate covariance matrix of parameter estimates is found to be singular.

MSINGULAR | MSING=*r*
specifies a relative singularity criterion r, $r > 0$, for the inversion of the information matrix, which is needed to compute the approximate covariance matrix of parameter estimates.

SINGULAR | SING =*r*
specifies the singularity criterion r, $0 \leq r \leq 1$, that is used for the inversion of the Hessian matrix. The default value is 1E−8.

VSINGULAR | VSING=*r*
specifies a relative singularity criterion r, $r > 0$, for the inversion of the information matrix, which is needed to compute the approximate covariance matrix of parameter estimates.

Termination Criteria Options

Let x^* be the point at which the objective function $f(\cdot)$ is optimized, and let $x^{(k)}$ be the parameter values attained at the kth iteration. All optimization techniques stop at the kth iteration if at least one of a set of termination criteria is satisfied. The specified termination criteria should enable termination in an area of sufficient size around x^*. You can avoid termination respective to any of the following function, gradient, or parameter criteria by setting the corresponding option to zero. There is a default set of termination criteria for each optimization technique; most of these default settings make the criteria ineffective for termination. PROC CALIS might have problems due to rounding errors (especially in derivative evaluations) that prevent an optimizer from satisfying strong termination criteria.

Note that PROC CALIS also terminates if the point $x^{(k)}$ is fully constrained by linearly independent active linear or boundary constraints, and all Lagrange multiplier estimates of active inequality constraints are greater than a small negative tolerance.

The following options are available only in the NLOPTIONS statement (except for FCONV, GCONV, MAXFUNC, and MAXITER), and they affect the termination criteria.

Options Used by All Optimization Techniques

ABSCONV | ABSTOL=*r*
specifies an absolute function convergence criterion.

- For minimization, termination requires

$$f^{(k)} = f(x^{(k)}) \leq ABSCONV$$

- For maximization, termination requires

$$f^{(k)} = f(x^{(k)}) \geq ABSCONV$$

The default value of ABSCONV is as follows:

- for minimization, the negative square root of the largest double-precision value
- for maximization, the positive square root of the largest double-precision value

MAXFUNC | MAXFU=*i*

requires the number of function calls to be no larger than i. The default values are listed in the following table.

TECH=	MAXFUNC Default
LEVMAR, NEWRAP, NRRIDG, TRUREG	i=125
DBLDOG, QUANEW	i=500
CONGRA	i=1000

The default is used if you specify MAXFUNC=0. The optimization can be terminated only after completing a full iteration. Therefore, the number of function calls that are actually performed can exceed the number that is specified by the MAXFUNC= option.

MAXITER | MAXIT= *i <n>*

requires the number of iterations to be no larger than i. The default values are listed in the following table.

TECH=	MAXITER Default
LEVMAR, NEWRAP, NRRIDG, TRUREG	i=50
DBLDOG, QUANEW	i=200
CONGRA	i=400

The default is used if you specify MAXITER=0 or you omit the MAXITER option.

The optional second value n is valid only for TECH=QUANEW with nonlinear constraints. It specifies an upper bound n for the number of iterations of an algorithm and reduces the violation of nonlinear constraints at a starting point. The default value is n=20. For example, specifying `MAXITER= . 0` means that you do not want to exceed the default number of iterations during the main optimization process and that you want to suppress the feasible-point algorithm for nonlinear constraints.

MAXTIME=*r*

requires the CPU time to be no larger than r. The default value of the MAXTIME= option is the largest double floating-point number on your computer.

MINITER | MINIT=*i*

specifies the minimum number of iterations. The default value is $i = 0$.

The ABSCONV=, MAXITER=, MAXFUNC=, and MAXTIME= options are useful for dividing a time-consuming optimization problem into a series of smaller problems by using the OUTEST= and INEST= data sets.

Options for Unconstrained and Linearly Constrained Optimization Techniques

ABSFCONV | ABSFTOL=*r* < *n* >

specifies the absolute function convergence criterion. Termination requires a small change of the function value in successive iterations,

$$|f(x^{(k-1)}) - f(x^{(k)})| \leq r$$

The default value is $r = 0$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

ABSGCONV | ABSGTOL=*r* < *n* >

specifies the absolute gradient convergence criterion. Termination requires the maximum absolute gradient element to be small,

$$\max_j |g_j^{(k)}| \leq r$$

The default value is *r*=1E−5. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

NOTE: In some applications, the small default value of the ABSGCONV= criterion is too difficult to satisfy for some of the optimization techniques.

ABSXCONV | ABSXTOL=*r* < *n* >

specifies the absolute parameter convergence criterion. Termination requires a small Euclidean distance between successive parameter vectors,

$$\| x^{(k)} - x^{(k-1)} \|_2 \leq r$$

The default value is $r = 0$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

FCONV | FTOL=*r* < *n* >

specifies the relative function convergence criterion. Termination requires a small relative change of the function value in successive iterations,

$$\frac{|f(x^{(k)}) - f(x^{(k-1)})|}{\max(|f(x^{(k-1)})|, FSIZE)} \leq r$$

where $FSIZE$ is defined by the FSIZE= option. The default value is $r = 10^{-FDIGITS}$, where $FDIGITS$ either is specified or is set by default to $-log_{10}(\epsilon)$, where ϵ is the machine precision. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

FCONV2 | FTOL2=*r* < *n* >

specifies another function convergence criterion. For least squares problems, termination requires a small predicted reduction

$$df^{(k)} \approx f(x^{(k)}) - f(x^{(k)} + s^{(k)})$$

of the objective function.

The predicted reduction

$$\begin{aligned} df^{(k)} &= -g^{(k)'}s^{(k)} - \frac{1}{2}s^{(k)'}G^{(k)}s^{(k)} \\ &= -\frac{1}{2}s^{(k)'}g^{(k)} \\ &\leq r \end{aligned}$$

is computed by approximating the objective function f by the first two terms of the Taylor series and substituting the Newton step

$$s^{(k)} = -G^{(k)-1}g^{(k)}$$

The FCONV2 criterion is the unscaled version of the GCONV criterion. The default value is $r = 0$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

FDIGITS=*r*

specifies the number of accurate digits in evaluations of the objective function. Fractional values such as FDIGITS=4.7 are allowed. The default value is $r = -log_{10}\epsilon$, where ϵ is the machine precision. The value of r is used for the specification of the default value of the FCONV= option.

FSIZE=*r*

specifies the $FSIZE$ parameter of the relative function and relative gradient termination criteria. The default value is $r = 0$. See the FCONV= and GCONV= options for details.

GCONV | GTOL=*r* **<** *n* **>**

specifies the relative gradient convergence criterion. For all techniques except the CONGRA technique, termination requires that the normalized predicted function reduction be small,

$$\frac{[g^{(k)}]'[G^{(k)}]^{-1}g^{(k)}}{\max(|f(x^{(k)})|, FSIZE)} \leq r$$

where $FSIZE$ is defined by the FSIZE= option. For the CONGRA technique (where a reliable Hessian estimate $\mathbf{G}$ is not available),

$$\frac{\| g^{(k)} \|_2^2 \quad \| s^{(k)} \|_2}{\| g^{(k)} - g^{(k-1)} \|_2 \max(|f(x^{(k)})|, FSIZE)} \leq r$$

is used. The default value is r=1E−8. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

NOTE: The default setting for the GCONV= option sometimes leads to early termination far from the location of the optimum. This is especially true for the special form of this criterion used in the CONGRA optimization.

GCONV2 | GTOL2=*r* **<** *n* **>**

specifies another relative gradient convergence criterion. For least squares problems and the

TRUREG, LEVMAR, NRRIDG, and NEWRAP techniques, the criterion of Browne (1982) is used,

$$\max_j \frac{|g_j^{(k)}|}{\sqrt{f(x^{(k)})G_{j,j}^{(k)}}} \leq r$$

This criterion is not used by the other techniques. The default value is $r = 0$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

XCONV | XTOL=*r* **<** *n* **>**

specifies the relative parameter convergence criterion. Termination requires a small relative parameter change in subsequent iterations,

$$\frac{\max_j |x_j^{(k)} - x_j^{(k-1)}|}{\max(|x_j^{(k)}|, |x_j^{(k-1)}|, XSIZE)} \leq r$$

The default value is $r = 0$. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

XSIZE=*r*

specifies the $XSIZE$ parameter of the relative function and relative gradient termination criteria. The default value is $r = 0$. See the XCONV= option for details.

Options for Nonlinearly Constrained Techniques

The non-NMSIMP algorithms available for nonlinearly constrained optimization (currently only TECH=QUANEW) do not monotonically reduce either the value of the objective function or some kind of merit function that combines objective and constraint functions. Furthermore, the algorithm uses the watchdog technique with backtracking (Chamberlain et al., 1982). Therefore, no termination criteria are implemented that are based on the values (x or f) of successive iterations. In addition to the criteria used by all optimization techniques, only three more termination criteria are currently available, and they are based on the Lagrange function:

$$L(x, \lambda) = f(x) - \sum_{i=1}^{m} \lambda_i c_i(x)$$

and its gradient

$$\nabla_x L(x, \lambda) = g(x) - \sum_{i=1}^{m} \lambda_i \nabla_x c_i(x)$$

Here, m denotes the total number of constraints, $g = g(x)$ denotes the gradient of the objective function, and λ denotes the m vector of Lagrange multipliers. The Kuhn-Tucker conditions require that the gradient of the Lagrange function is zero at the optimal point (x^*, λ^*):

$$\nabla_x L(x^*, \lambda^*) = 0$$

The termination criteria available for nonlinearly constrained optimization follow.

ABSGCONV | ABSGTOL=*r* < *n* >
specifies that termination requires the maximum absolute gradient element of the Lagrange function to be small,

$$\max_j |\{\nabla_x L(x^{(k)}, \lambda^{(k)})\}_j| \leq r$$

The default value is r=1E−5. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

FCONV2 | FTOL2=*r* < *n* >
specifies that termination requires the predicted objective function reduction to be small:

$$|g(x^{(k)})s(x^{(k)})| + \sum_{i=1}^{m} |\lambda_i c_i| \leq r$$

The default value is r=1E−6. This is the criterion used by the programs VMCWD and VF02AD (Powell 1982b). The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

GCONV | GTOL=*r* < *n* >
specifies that termination requires the normalized predicted objective function reduction to be small:

$$\frac{|g(x^{(k)})s(x^{(k)})| + \sum_{i=1}^{m} |\lambda_i c_i(x^{(k)})|}{\max(|f(x^{(k)})|, FSIZE)} \leq r$$

where $FSIZE$ is defined by the FSIZE= option. The default value is r=1E−8. The optional integer value n determines the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

Miscellaneous Options

Options for the Approximate Covariance Matrix of Parameter Estimates

CFACTOR=*r*
specifies the scalar factor for the covariance matrix of parameter estimates. The scalar $r \geq 0$ replaces the default value c/NM. For more details, see the section "Approximate Standard Errors" on page 928.

NOHLF
specifies that the Hessian matrix of the objective function (rather than the Hessian matrix of the Lagrange function) be used for computing the approximate covariance matrix of parameter estimates and, therefore, the approximate standard errors.

It is theoretically not correct to use the NOHLF option. However, since most implementations use the Hessian matrix of the objective function and not the Hessian matrix of the Lagrange function for computing approximate standard errors, the NOHLF option can be used to compare the results.

Options for Additional Displayed Output

PALL | ALL
displays information about the starting values and final values of the optimization process.

PCRPJAC | PJTJ
displays the approximate Hessian matrix. If general linear or nonlinear constraints are active at the solution, the projected approximate Hessian matrix is also displayed.

PHESSIAN | PHES
displays the Hessian matrix. If general linear or nonlinear constraints are active at the solution, the projected Hessian matrix is also displayed.

PHISTORY | PHIS
displays the optimization history. The PHISTORY option is set automatically if the PALL or PRINT option is set.

PINIT | PIN
displays the initial values and derivatives (if available). The PINIT option is set automatically if the PALL option is set.

PNLCJAC
displays the Jacobian matrix of nonlinear constraints specified by the NLINCON statement. The PNLCJAC option is set automatically if the PALL option is set.

PRINT | PRI
displays the results of the optimization process, such as parameter estimates and constraints.

Options for Fine-Tuning Optimization Techniques

DAMPSTEP | DS *<=r>*
specifies that the initial step-size value $\alpha^{(0)}$ for each line search (used by the QUANEW, CONGRA, or NEWRAP technique) cannot be larger than r times the step-size value used in the former iteration. If the factor r is not specified, the default value is $r = 2$. The DAMPSTEP option can prevent the line-search algorithm from repeatedly stepping into regions where some objective functions are difficult to compute or where they can lead to floating-point overflows during the computation of objective functions and their derivatives. The DAMPSTEP<=*r*> option can prevent time-costly function calls during line searches with very small step sizes α of objective functions. For more information about setting the start values of each line search, see the section "Restricting the Step Length" on page 952.

HESCAL | HS = 0 | 1 | 2 | 3
specifies the scaling version of the Hessian or crossproduct Jacobian matrix used in NRRIDG, TRUREG, LEVMAR, NEWRAP, or DBLDOG optimization. If HS is not equal to zero, the first iteration and each restart iteration set the diagonal scaling matrix $\mathbf{D}^{(0)} = diag(d_i^{(0)})$:

$$d_i^{(0)} = \sqrt{\max(|\mathbf{G}_{i,i}^{(0)}|, \epsilon)}$$

where $\mathbf{G}^{(0)}_{i,i}$ are the diagonal elements of the Hessian or crossproduct Jacobian matrix. In every other iteration, the diagonal scaling matrix $\mathbf{D}^{(0)} = diag(d_i^{(0)})$ is updated depending on the HS option:

HS=0 specifies that no scaling is done.

HS=1 specifies the Moré (1978) scaling update:

$$d_i^{(k+1)} = \max(d_i^{(k)}, \sqrt{\max(|\mathbf{G}^{(k)}_{i,i}|, \epsilon)})$$

HS=2 specifies the Dennis, Gay, and Welsch (1981) scaling update:

$$d_i^{(k+1)} = \max(0.6 * d_i^{(k)}, \sqrt{\max(|\mathbf{G}^{(k)}_{i,i}|, \epsilon)})$$

HS=3 specifies that d_i is reset in each iteration:

$$d_i^{(k+1)} = \sqrt{\max(|\mathbf{G}^{(k)}_{i,i}|, \epsilon)}$$

In the preceding equations, ϵ is the relative machine precision. The default is HS=1 for LEVMAR minimization and HS=0 otherwise. Scaling of the Hessian or crossproduct Jacobian can be time-consuming in the case where general linear constraints are active.

LCDEACT | LCD=*r*

specifies a threshold r for the Lagrange multiplier that decides whether an active inequality constraint remains active or can be deactivated. For maximization, r must be greater than zero; for minimization, r must be smaller than zero. The default is

$$r = \pm \min(0.01, \max(0.1 * ABSGCONV, 0.001 * gmax^{(k)}))$$

where "+" stands for maximization, "−" stands for minimization, $ABSGCONV$ is the value of the absolute gradient criterion, and $gmax^{(k)}$ is the maximum absolute element of the (projected) gradient $g^{(k)}$ or $Z'g^{(k)}$.

LCEPSILON | LCEPS | LCE=*r*

specifies the range r, $r \geq 0$, for active and violated boundary and linear constraints. If the point $x^{(k)}$ satisfies the condition

$$|\sum_{j=1}^{n} a_{ij} x_j^{(k)} - b_i| \leq r * (|b_i| + 1)$$

the constraint i is recognized as an active constraint. Otherwise, the constraint i is either an inactive inequality or a violated inequality or equality constraint. The default value is r=1E−8. During the optimization process, the introduction of rounding errors can force PROC NLP to increase the value of r by factors of 10. If this happens, it is indicated by a message displayed in the log.

LCSINGULAR | LCSING | LCS=*r*

specifies a criterion r, $r \geq 0$, used in the update of the QR decomposition that decides whether an active constraint is linearly dependent on a set of other active constraints. The default is r=1E−8. The larger r becomes, the more the active constraints are recognized as being linearly dependent.

NOEIGNUM

suppresses the computation and displayed output of the determinant and the inertia of the Hessian, crossproduct Jacobian, and covariance matrices. The inertia of a symmetric matrix are the numbers of negative, positive, and zero eigenvalues. For large applications, the NOEIGNUM option can save computer time.

RESTART | REST=*i*

specifies that the QUANEW or CONGRA algorithm is restarted with a steepest descent/ascent search direction after at most i iterations, $i > 0$. Default values are as follows:

- CONGRA: UPDATE=PB: restart is done automatically so specification of i is not used.
- CONGRA: UPDATE≠PB: $i = \min(10n, 80)$, where n is the number of parameters.
- QUANEW: i is the largest integer available.

VERSION | VS = 1 | 2

specifies the version of the quasi-Newton optimization technique with nonlinear constraints.

VS=1 specifies the update of the μ vector as in Powell (1978a, 1978b) (update like VF02AD).

VS=2 specifies the update of the μ vector as in Powell (1982a, 1982b) (update like VMCWD).

The default is VS=2.

PARAMETERS Statement

PARAMETERS | PARMS *parameter(s)* << = > *number(s)* >
<< , > *parameter(s)* << = > *number(s)* > . . . > ;

The PARAMETERS statement defines additional parameters that are not elements of a model matrix to use in your own programming statements. You can specify more than one PARAMETERS statement with each PROC CALIS statement. The *parameters* can be followed by an equal sign and a number list. The values of the *numbers* list are assigned as initial values to the preceding parameters in the *parameters* list. For example, each of the following statements assigns the initial values ALPHA=.5 and BETA=-.5 for the parameters used in programming statements:

```
parameters alfa beta=.5 -.5;
parameters alfa beta (.5 -.5);
parameters alfa beta .5 -.5;
parameters alfa=.5 beta (-.5);
```

The number of parameters and the number of values does not have to match. When there are fewer values than parameter names, either the RANDOM= or START= option is used. When there are more values than parameter names, the extra values are dropped. Parameters listed in the PARAMETERS statement can be assigned initial values by programming statements or by the START= or RANDOM= option in the PROC CALIS statement.

CAUTION: The OUTRAM= and INRAM= data sets do not contain any information about the PARAMETERS statement or additional programming statements.

PARTIAL Statement

PARTIAL *variables* **;**

If you want the analysis to be based on a partial correlation or covariance matrix, use the PARTIAL statement to list the variables used to partial out the variables in the analysis. You can specify only one PARTIAL statement with each PROC CALIS statement.

RAM Model Statement

RAM *list-entry* < , *list-entry* . . . > **;**

where *list-entry* represents
matrix-number row-number column-number < *value* > < *parameter-name* >

The RAM statement defines the elements of the symmetric RAM matrix model

$$\mathbf{v} = \mathbf{A}\mathbf{v} + \mathbf{u}$$

in the form of a list type input (McArdle and McDonald 1984).

The covariance structure is given by

$$\mathbf{C} = \mathbf{J}(\mathbf{I} - \mathbf{A})^{-1}\mathbf{P}((\mathbf{I} - \mathbf{A})^{-1})'\mathbf{J}'$$

with selection matrix $\mathbf{J}$ and

$$\mathbf{C} = \mathcal{E}\{\mathbf{J}\mathbf{v}\mathbf{v}'\mathbf{J}'\}, \qquad \mathbf{P} = \mathcal{E}\{\mathbf{u}\mathbf{u}'\}$$

You can specify only one RAM statement with each PROC CALIS statement. Using the RAM statement requires that the first n variable numbers in the path diagram and in the vector v correspond to the numbers of the n manifest variables of the given covariance or correlation matrix. If you are not sure what the order of the manifest variables in the DATA= data set is, use a VAR statement to specify the order of these observed variables. Using the AUGMENT option includes the INTERCEPT variable as a manifest variable with number $n + 1$ in the RAM model. In this case, latent variables have to start with $n + 2$. The box of each manifest variable in the path diagram is assigned the number of the variable in the covariance or correlation matrix.

The selection matrix **J** is always a rectangular identity (IDE) matrix, and it does not have to be specified in the RAM statement.

A constant matrix element is defined in a RAM statement by a *list-entry* with four numbers. You define a parameter element by three or four numbers followed by a name for the parameter. Separate the list entries with a comma. Each *list-entry* in the RAM statement corresponds to a path in the diagram, as follows:

- The first number in each list entry (*matrix-number*) is the number of arrowheads of the path, which is the same as the number of the matrix in the RAM model (1 := **A** , 2 := **P**).
- The second number in each list entry (*row-number*) is the number of the node in the diagram to which the path points, which is the same as the row number of the matrix element.
- The third number in each list entry (*column-number*) is the number of the node in the diagram from which the path originates, which is the same as the column number of the matrix element.
- The fourth number (*value*) gives the (initial) value of the path coefficient. If you do not specify a fifth *list-entry*, this number specifies a constant coefficient; otherwise, this number specifies the initial value of this parameter. It is not necessary to specify the fourth item. If you specify neither the fourth nor the fifth item, the constant is set to 1 by default. If the fourth item (*value*) is not specified for a parameter, PROC CALIS tries to compute an initial value for this parameter.
- If the path coefficient is a parameter rather than a constant, then a fifth item in the list entry (*parameter-name*) is required to assign a name to the parameter. Using the same name for different paths constrains the corresponding coefficients to be equal.

If the initial value of a parameter is not specified in the list, the initial value is chosen in one of the following ways:

- If the PROC CALIS statement contains a RANDOM=i option, then the parameter obtains a randomly generated initial value r, such that $0 \leq r \leq 1$. The uninitialized parameters in the diagonals of the central model matrices are given the random values r multiplied by 10, 100, or the value specified in the DEMPHAS= option.
- If the RANDOM= option is not used, PROC CALIS tries to estimate the initial values.
- If the initial values cannot be estimated, the value of the START= option is used as an initial value.

If your model contains many unconstrained parameters and it is too cumbersome to find different parameter names, you can specify all those parameters by the same prefix name. A prefix is a short name followed by a colon. The CALIS procedure then generates a parameter name by appending an integer suffix to this prefix name. The prefix name should have no more than five or six characters so that the generated parameter name is not longer than eight characters. To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names.

For example, consider the following model described in the section "Specifying a Second-Order Factor-Analysis Model" on page 885:

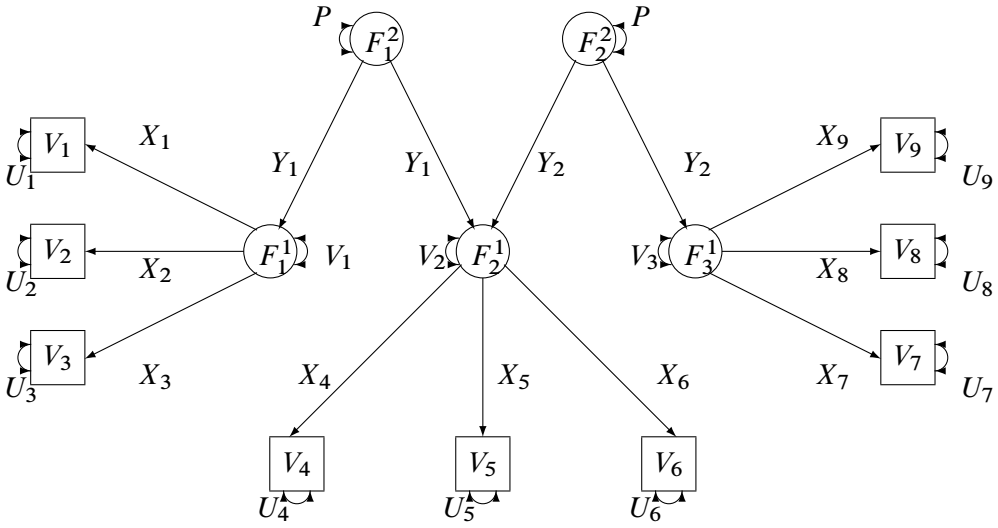
$$\mathbf{S} = \mathbf{F}_1\mathbf{F}_2\mathbf{P}_2\mathbf{F}_2'\mathbf{F}_1' + \mathbf{F}_1\mathbf{U}_2^2\mathbf{F}_1' + \mathbf{U}_1^2$$

You can fit this model by using the following RAM model statement:

```
ram
   1  1 10    x1,
   1  2 10    x2,
   1  3 10    x3,
   1  4 11    x4,
   1  5 11    x5,
   1  6 11    x6,
   1  7 12    x7,
   1  8 12    x8,
   1  9 12    x9,
   1 10 13    y1,
   1 11 13    y1,
   1 11 14    y2,
   1 12 14    y2,
   2  1  1    u:,
   2  2  2    u:,
   2  3  3    u:,
   2  4  4    u:,
   2  5  5    u:,
   2  6  6    u:,
   2  7  7    u:,
   2  8  8    u:,
   2  9  9    u:,
   2 10 10    v:,
   2 11 11    v:,
   2 12 12    v:,
   2 13 13    p ,
   2 14 14    p ;
run;
```

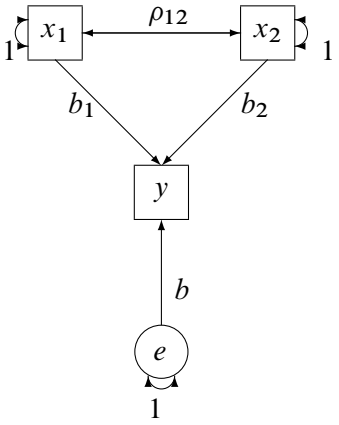
The confirmatory second-order factor analysis model corresponds to the path diagram displayed in Figure 25.3.

Figure 25.3 Path Diagram of Second-Order Factor Analysis Model

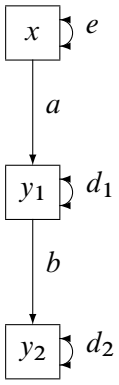


There is a very close relationship between the RAM model algebra and the specification of structural linear models by path diagrams. See Figure 25.4 for an example.

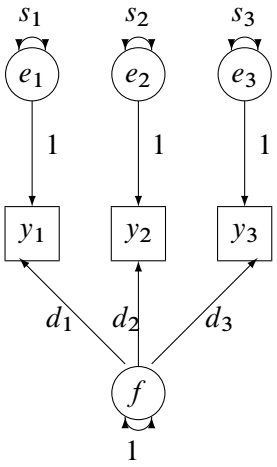
Figure 25.4 Examples of RAM Nomography



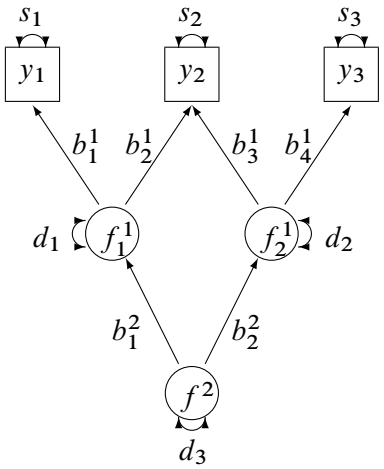
1. Multiple Regression



2. Chain Simplex



3. First-Order Factor Analysis



4. Second-Order Factor Analysis

Refer to McArdle (1980) for the interpretation of the models displayed in Figure 25.4.

SAS Programming Statements

This section lists the programming statements used to express the linear and nonlinear constraints on the parameters and documents the differences between programming statements in PROC CALIS and program statements in the DATA step. The very different use of the ARRAY statement by PROC CALIS is also discussed. Most of the programming statements that can be used in the SAS DATA step also can be used in PROC CALIS. Refer to *SAS Language Reference: Dictionary* for a description of the SAS programming statements. You can specify the following SAS programming

statements to compute parameter constraints with the CALIS procedure:

ABORT;
CALL *name* < **(** *expression* < **,** *expression . . .* > **)** >**;**
DELETE;
DO < *variable* **=** *expression*
< **TO** *expression* > < **BY** *expression* >
< **,** *expression* < **TO** *expression* > < **BY** *expression* > . . . >
>
< **WHILE** *expression* > < **UNTIL** *expression* >**;**
END;
GOTO *statement-label***;**
IF *expression***;**
IF *expression* **THEN** *program-statement***;**
ELSE *program-statement***;**
variable **=** *expression***;**
variable **+** *expression***;**
LINK *statement-label***;**
PUT < *variable* > < **=** > < *...* >**;**
RETURN;
SELECT < **(** *expression* **)** >**;**
STOP;
SUBSTR(*variable, index, length* **)=** *expression***;**
WHEN (*expression* **)** *program-statement***;**
OTHERWISE *program-statement***;**

For the most part, the SAS programming statements work the same as they do in the SAS DATA step as documented in *SAS Language Reference: Concepts*. However, there are several differences that should be noted.

- The ABORT statement does not allow any arguments.
- The DO statement does not allow a character index variable. Thus,

  ```
  do I=1,2,3;
  ```

  is supported; however,

  ```
  do I='A','B','C';
  ```

  is not valid in PROC CALIS, although it is supported in the DATA step.
- The PUT statement, used mostly for program debugging in PROC CALIS, supports only some of the features of the DATA step PUT statement, and it has some new features that the DATA step PUT statement does not have:
  - The CALIS procedure PUT statement does not support line pointers, factored lists, iteration factors, overprinting, _INFILE_, the colon (:) format modifier, or $.
  - The CALIS procedure PUT statement does support expressions enclosed in parentheses. For example, the following statement displays the square root of x:

```
put (sqrt(x));
```

– The CALIS procedure PUT statement supports the print item _PDV_ to display a formatted listing of all variables in the program. For example, the following statement displays a much more readable listing of the variables than the _ALL_ print item:

```
put _pdv_ ;
```

- The WHEN and OTHERWISE statements allow more than one target statement. That is, DO/END groups are not necessary for multiple WHEN statements. For example, the following syntax is valid:

```
select;
   when ( expression1 ) statement1;
                        statement2;
   when ( expression2 ) statement3;
                        statement4;
end;
```

You can specify one or more PARMS statements to define parameters used in the programming statements that are not defined in the model matrices (MATRIX, RAM, LINEQS, STD, or COV statement).

Parameters that are used only on the right-hand side of your programming statements are called independent, and parameters that are used at least once on the left-hand side of an equation in the program code are called dependent parameters. The dependent parameters are used only indirectly in the minimization process. They should be fully defined as functions of the independent parameters. The independent parameters are included in the set **X** of parameters used in the minimization. Be sure that all independent parameters used in your programming statements are somehow connected to elements of the model matrices. Otherwise the minimization function does not depend on those independent parameters, and the parameters vary without control (since the corresponding derivative is the constant 0). You also can specify the PARMS statement to set the initial values of all independent parameters used in the programming statements that are not defined as elements of model matrices.

ARRAY Statement

ARRAY *arrayname* < *dimensions* >< $ >< *variables and constants* > **;**

The ARRAY statement is similar to, but not the same as, the ARRAY statement in the DATA step. The ARRAY statement is used to associate a name with a list of variables and constants. The array name can then be used with subscripts in the program to refer to the items in the list.

The ARRAY statement supported by PROC CALIS does not support all the features of the DATA step ARRAY statement. With PROC CALIS, the ARRAY statement cannot be used to give initial values to array elements. Implicit indexing variables cannot be used; all array references must have explicit subscript expressions. Only exact array dimensions are allowed; lower-bound specifications are not supported. A maximum of six dimensions is allowed.

On the other hand, the ARRAY statement supported by PROC CALIS does allow both variables and constants to be used as array elements. Constant array elements cannot be changed. Both the dimension specification and the list of elements are optional, but at least one must be given. When the list of elements is not given or fewer elements than the size of the array are listed, array variables are created by suffixing element numbers to the array name to complete the element list.

STD Statement

STD *assignment* < *, assignment . . .* > **;**

where *assignment* represents
variables = *pattern-definition*

The STD statement tells which variances are parameters to estimate and which are fixed. The STD statement can be used only with the LINEQS statement. You can specify only one STD statement with each LINEQS model statement. The STD statement defines the diagonal elements of the central model matrix $\mathbf{\Phi}$. These elements correspond to the variances of the exogenous variables and to the error variances of the endogenous variables. Elements that are not defined are assumed to be zero.

Each *assignment* consists of a variable list (*variables*) on the left-hand side and a pattern list (*pattern-definition*) on the right-hand side of an equal sign. The *assignments* in the STD statement must be separated by commas. The *variables* list on the left-hand side of the equal sign should contain only names of variables that do not appear on the left-hand side of an equation in the LINEQS statement—that is, exogenous, error, and disturbance variables.

The *pattern-definition* on the right-hand side is similar to that used in the MATRIX statement. Each list element on the right-hand side defines the variance of the variable on the left-hand side in the same list position. A name on the right-hand side means that the corresponding variance is a parameter to estimate. A name on the right-hand side can be followed by a number inside parentheses that gives the initial value. A number on the right-hand side means that the corresponding variance of the variable on the left-hand side is fixed. If the right-hand-side list is longer than the left-hand-side variable list, the right-hand-side list is shortened to the length of the variable list. If the right-hand-side list is shorter than the variable list, the right-hand-side list is filled with repetitions of the last item in the list.

The right-hand side can also contain prefixes. A prefix is a short name followed by a colon. The CALIS procedure then generates a parameter name by appending an integer suffix to this prefix name. The prefix name should have no more than five or six characters so that the generated parameter name is not longer than eight characters. To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names. For example, if the prefix A is not used in any previous statement, the STD statement

```
std E1-E6=6 * A: (6 * 3.) ;
```

defines the six error variances as free parameters A1, . . . , A6, all with starting values of 3.

STRUCTEQ Statement

STRUCTEQ *variable* < *variable . . .* > **;**

The STRUCTEQ statement is used to list the dependent variables of the structural equations. This statement is ignored if you omit the PDETERM option. This statement is useful because the term *structural equation* is not defined in a unique way, and PROC CALIS has difficulty identifying the structural equations.

If LINEQS statements are used, the names of the left-hand-side (dependent) variables of those equations to be treated as structural equations should be listed in the STRUCTEQ statement.

If the RAM statement is used, variable names in the STRUCTEQ statements depend on the VARNAMES statement:

- If the VARNAMES statement is used, variable names must correspond to those in the VARNAMES statement.
- If the VARNAMES statement is not used, variable names must correspond to the names of manifest variables or latent (F) variables.

The STRUCTEQ statement also defines the names of variables used in the causal coefficient matrix of the structural equations, **B**, for computing the *Stability Coefficient of Reciprocal Causation* (the largest eigenvalue of the $\mathbf{BB}'$ matrix). If the PROC CALIS option PDETERM is used without the STRUCTEQ statement, the structural equations are defined as described in the PDETERM option. See the PROC CALIS option PDETERM on page 863 for more details.

VAR Statement

VAR *variables* **;**

The VAR statement lists the numeric variables to be analyzed. If the VAR statement is omitted, all numeric variables not mentioned in other statements are analyzed. You can use the VAR statement to ensure that the manifest variables appear in correct order for use in the RAM statement. Only one VAR statement can be used with each PROC CALIS statement. If you do not use all manifest variables when you specify the model with a RAM or LINEQS statement, PROC CALIS does automatic variable selection. For more information, see the section "Automatic Variable Selection" on page 942.

VARNAMES Statement

VARNAMES | **VNAMES** *assignment* < , *assignment* . . . > **;**

where *assignment* represents
matrix-id variable-names or *matrix-name* = *matrix-name*

Use the VARNAMES statement in connection with the RAM, COSAN, or FACTOR model statement to allocate names to latent variables including error and disturbance terms. This statement is not needed if you are using the LINEQS statement.

In connection with the RAM model statement, the *matrix-id* must be specified by the integer number as it is used in the RAM list input (1 for matrix **A**, 2 for matrix **P**). Because the first variables of matrix **A** correspond to the manifest variables in the input data set, you can specify names only for the latent variables following the manifest variables in the rows of **A**. For example, in the RAM notation of the alienation example, you can specify the latent variables by names F1, F2, F3 and the error variables by names E1, . . . , E6, D1, D2, D3 with the following statement:

```
vnames 1  F1-F3,
       2  E1-E6 D1-D3;
```

If the RAM model statement is not accompanied by a VNAMES statement, default variable names are assigned using the prefixes "F", "E", and "D" with numerical suffixes: latent variables are F1, F2, . . . , and error variables are E1, E2, ...

The *matrix-id* must be specified by its name when used with the COSAN or FACTOR statement. The *variable-names* following the matrix name correspond to the columns of this matrix. The variable names corresponding to the rows of this matrix are set automatically by the following rules:

- If it is the first matrix of any term, the row variable names are the names of the manifest variables.
- If it is the central symmetric matrix of any term, the row variable names are the same as the column variable names of the same matrix.
- For any other matrices, the row variable names are the same as the column variable names of the preceding matrix.

You also can use the second kind of name assignment in connection with a COSAN statement. Two matrix names separated by an equal sign allocate the column names of one matrix to the column names of the other matrix. This assignment assumes that the column names of at least one of the two matrices are already allocated. For example, in the COSAN notation of the alienation example, you can specify the variable names by using the following statements to allocate names to the columns of **J**, **A**, and **P**:

```
vnames  J  V1-V6 F1-F3 ,
        A =J ,
        P  E1-E6 D1-D3 ;
```

WEIGHT Statement

WEIGHT *variable* **;**

To compute weighted covariances or correlations, specify the name of the weighting variable in a WEIGHT statement. This is often done when the error variance associated with each observation is different and the values of the weight variable are proportional to the reciprocals of the variances. You can use only one WEIGHT statement with each PROC CALIS statement. The WEIGHT and FREQ statements have a similar effect, except the WEIGHT statement does not alter the number of observations unless VARDEF=WGT or VARDEF=WDF. An observation is used in the analysis only if the WEIGHT variable is greater than 0 and is not missing.

Details: CALIS Procedure

Input Data Sets

You can use four different kinds of input data sets in the CALIS procedure, and you can use them simultaneously. The DATA= data set contains the data to be analyzed, and it can be an ordinary SAS data set containing raw data or a special TYPE=COV, TYPE=UCOV, TYPE=CORR, TYPE=UCORR, TYPE=SYMATRIX, TYPE=SSCP, or TYPE=FACTOR data set containing previously computed statistics. The INEST= data set specifies an input data set that contains initial estimates for the parameters used in the optimization process, and it can also contain boundary and general linear constraints on the parameters. If the model does not change too much, you can use an OUTEST= data set from a previous PROC CALIS analysis; the initial estimates are taken from the values of the PARMS observation. The INRAM= data set names a third input data set that contains all information needed to specify the analysis model in RAM list form (except for user-written program statements). Often the INRAM= data set can be the OUTRAM= data set from a previous PROC CALIS analysis. See the section "OUTRAM= SAS-data-set" on page 917 for the structure of both OUTRAM= and INRAM= data sets. Using the INWGT= data set enables you to read in the weight matrix **W** that can be used in generalized least squares, weighted least squares, or diagonally weighted least squares estimation.

DATA= SAS-data-set

A TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set can be created by the CORR procedure or various other procedures. It contains means, standard deviations, the sample size, the covariance or correlation matrix, and possibly other statistics depending on which procedure is used.

If your data set has many observations and you plan to run PROC CALIS several times, you can save computer time by first creating a TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR

data set and using it as input to PROC CALIS. For example, assuming that PROC CALIS is first run with an OUTRAM=MOD option, you can run the following statements:

```
* create TYPE=COV data set;
proc corr cov nocorr data=raw outp=cov(type=cov);
run;
* analysis using correlations;
proc calis data=cov inram=mod;
run;
* analysis using covariances;
proc calis cov data=cov inram=mod;
run;
```

Most procedures automatically set the TYPE= option of an output data set appropriately. However, the CORR procedure sets TYPE=CORR unless an explicit TYPE= option is used. Thus, (TYPE=COV) is needed in the preceding PROC CORR request, since the output data set is a covariance matrix. If you use a DATA step with a SET statement to modify this data set, you must declare the TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR attribute in the new data set.

You can use a VAR statement with PROC CALIS when reading a TYPE=COV, TYPE=UCOV, TYPE=CORR, TYPE=UCORR, or TYPE=SSCP data set to select a subset of the variables or change the order of the variables.

CAUTION: Problems can arise from using the CORR procedure when there are missing data. By default, PROC CORR computes each covariance or correlation from all observations that have values present for the pair of variables involved ("pairwise deletion"). The resulting covariance or correlation matrix can have negative eigenvalues. A correlation or covariance matrix with negative eigenvalues is recognized as a singular matrix in PROC CALIS, and you cannot compute (default) generalized least squares or maximum likelihood estimates. You can specify the RIDGE option to ridge the diagonal of such a matrix to obtain a positive-definite data matrix. If the NOMISS option is used with the CORR procedure, observations with any missing values are completely omitted from the calculations ("listwise deletion"), and there is no possibility of negative eigenvalues (but still a chance for a singular matrix).

PROC CALIS can also create a TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set that includes all the information needed for repeated analyses. If the data set DATA=RAW does not contain missing values, the following statements should give the same PROC CALIS results as the previous example:

```
* using correlations;
proc calis data=raw outstat=cov inram=mod;
run;
* using covariances;
proc calis cov data=cov inram=mod;
run;
```

You can create a TYPE=COV, TYPE=UCOV, TYPE=CORR, TYPE=UCORR, or TYPE=SSCP data set in a DATA step. Be sure to specify the TYPE= option in parentheses after the data set name in the DATA statement, and include the _TYPE_ and _NAME_ variables. If you want to analyze the covariance matrix but your DATA= data set is a TYPE=CORR or TYPE=UCORR data set, you

should include an observation with _TYPE_=STD giving the standard deviation of each variable. If you specify the COV option, PROC CALIS analyzes the recomputed covariance matrix:

```
data correl(type=corr);
   input _type_ $ _name_ $ X1-X3;
   datalines;
std   .   4.  2.  8.
corr  X1  1.0  .   .
corr  X2   .7 1.0  .
corr  X3   .5  .4 1.0
;
proc calis cov inram=model;
run;
```

If you want to analyze the UCOV or UCORR matrix but your DATA= data set is a TYPE=COV or TYPE=CORR data set, you should include observations with _TYPE_=STD and _TYPE_=MEAN giving the standard deviation and mean of each variable.

INEST= SAS-data-set

You can use the INEST= (or INVAR= or ESTDATA=) input data set to specify the initial values of the parameters used in the optimization and to specify boundary constraints and the more general linear constraints that can be imposed on these parameters.

The variables of the INEST= data set must correspond to the following variables:

- a character variable _TYPE_ that indicates the type of the observation
- *n* numeric variables with the parameter names used in the specified PROC CALIS model
- the BY variables that are used in a DATA= input data set
- a numeric variable _RHS_ (right-hand side) (needed only if linear constraints are used)
- additional variables with names corresponding to constants used in the programming statements

The content of the _TYPE_ variable defines the meaning of the observation of the INEST= data set. PROC CALIS recognizes observations with the following _TYPE_ specifications.

PARMS — specifies initial values for parameters that are defined in the model statements of PROC CALIS. The _RHS_ variable is not used. Additional variables can contain the values of constants that are referred to in programming statements. At the beginning of each run of PROC CALIS, the values of the constants are read from the PARMS observation initializing the constants in the program statements.

UPPERBD | UB — specifies upper bounds with nonmissing values. The use of a missing value indicates that no upper bound is specified for the parameter. The _RHS_ variable is not used.

LOWERBD | LB specifies lower bounds with nonmissing values. The use of a missing value indicates that no lower bound is specified for the parameter. The _RHS_ variable is not used.

LE | <= | < specifies the linear constraint $\sum_j a_{ij}x_j \leq b_i$. The n parameter values contain the coefficients a_{ij}, and the _RHS_ variable contains the right-hand-side b_i. The use of a missing value indicates a zero coefficient a_{ij}.

GE | >= | > specifies the linear constraint $\sum_j a_{ij}x_j \geq b_i$. The n parameter values contain the coefficients a_{ij}, and the _RHS_ variable contains the right-hand-side b_i. The use of a missing value indicates a zero coefficient a_{ij}.

EQ | = specifies the linear constraint $\sum_j a_{ij}x_j = b_i$. The n parameter values contain the coefficients a_{ij}, and the _RHS_ variable contains the right-hand-side b_i. The use of a missing value indicates a zero coefficient a_{ij}.

The constraints specified in the INEST=, INVAR=, or ESTDATA= data set are added to the constraints specified in BOUNDS and LINCON statements.

You can use an OUTEST= data set from a PROC CALIS run as an INEST= data set in a new run. However, be aware that the OUTEST= data set also contains the boundary and general linear constraints specified in the previous run of PROC CALIS. When you are using this OUTEST= data set without changes as an INEST= data set, PROC CALIS adds the constraints from the data set to the constraints specified by a BOUNDS and LINCON statement. Although PROC CALIS automatically eliminates multiple identical constraints, you should avoid specifying the same constraint a second time.

INRAM= SAS-data-set

This data set is usually created in a previous run of PROC CALIS. It is useful if you want to reanalyze a problem in a different way, such as by using a different estimation method. You can alter an existing OUTRAM= data set, either in the DATA step or using the FSEDIT procedure, to create the INRAM= data set describing a modified model. For more details on the INRAM= data set, see the section "OUTRAM= SAS-data-set" on page 917.

In the case of a RAM or LINEQS analysis of linear structural equations, the OUTRAM= data set always contains the variable names of the model specified. These variable names and the model specified in the INRAM= data set are the basis of the automatic variable selection algorithm performed after reading the INRAM= data set.

INWGT= SAS-data-set

This data set enables you to specify a weight matrix other than the default matrix for the generalized, weighted, and diagonally weighted least squares estimation methods. The specification of any INWGT= data set for unweighted least squares or maximum likelihood estimation is ignored. For generalized and diagonally weighted least squares estimation, the INWGT= data set must contain a _TYPE_ and a _NAME_ variable as well as the manifest variables used in the analysis. The value of the _NAME_ variable indicates the row index i of the weight w_{ij}. For weighted least squares, the INWGT= data set must contain _TYPE_, _NAME_, _NAM2_, and _NAM3_ variables as well as the

manifest variables used in the analysis. The values of the _NAME_, _NAM2_, and _NAM3_ variables indicate the three indices i, j, k of the weight $w_{ij,kl}$. You can store information other than the weight matrix in the INWGT= data set, but only observations with _TYPE_=WEIGHT are used to specify the weight matrix $\mathbf{W}$. This property enables you to store more than one weight matrix in the INWGT= data set. You can then run PROC CALIS with each of the weight matrices by changing only the _TYPE_ observation in the INWGT= data set with an intermediate DATA step.

For more details on the INWGT= data set, see the section "OUTWGT= SAS-data-set" on page 922.

Output Data Sets

OUTEST= SAS-data-set

The OUTEST= (or OUTVAR=) data set is of TYPE=EST and contains the final parameter estimates, the gradient, the Hessian, and boundary and linear constraints. For METHOD=ML, METHOD=GLS, and METHOD=WLS, the OUTEST= data set also contains the approximate standard errors, the information matrix (crossproduct Jacobian), and the approximate covariance matrix of the parameter estimates ((generalized) inverse of the information matrix). If there are linear or nonlinear equality or active inequality constraints at the solution, the OUTEST= data set also contains Lagrange multipliers, the projected Hessian matrix, and the Hessian matrix of the Lagrange function.

The OUTEST= data set can be used to save the results of an optimization by PROC CALIS for another analysis with either PROC CALIS or another SAS procedure. Saving results to an OUTEST= data set is advised for expensive applications that cannot be repeated without considerable effort.

The OUTEST= data set contains the BY variables, two character variables _TYPE_ and _NAME_, t numeric variables corresponding to the parameters used in the model, a numeric variable _RHS_ (right-hand side) that is used for the right-hand-side value b_i of a linear constraint or for the value $f = f(x)$ of the objective function at the final point x^* of the parameter space, and a numeric variable _ITER_ that is set to zero for initial values, set to the iteration number for the OUTITER output, and set to missing for the result output.

The _TYPE_ observations in Table 25.5 are available in the OUTEST= data set, depending on the request.

Table 25.5 _TYPE_ Observations in the OUTEST= Data Set

TYPE	Description
ACTBC	If there are active boundary constraints at the solution x^*, three observations indicate which of the parameters are actively constrained, as follows.

NAME	Description
GE	indicates the active lower bounds
LE	indicates the active upper bounds
EQ	indicates the active masks

Table 25.5 *continued*

TYPE	Description
COV	contains the approximate covariance matrix of the parameter estimates; used in computing the approximate standard errors.
COVRANK	contains the rank of the covariance matrix of the parameter estimates.
CRPJ_LF	contains the Hessian matrix of the Lagrange function (based on CRPJAC).
CRPJAC	contains the approximate Hessian matrix used in the optimization process. This is the inverse of the information matrix.
EQ	If linear constraints are used, this observation contains the ith linear constraint $\sum_j a_{ij}x_j = b_i$. The parameter variables contain the coefficients a_{ij}, $j = 1, \ldots, n$, the _RHS_ variable contains b_i, and _NAME_=ACTLC or _NAME_=LDACTLC.
GE	If linear constraints are used, this observation contains the ith linear constraint $\sum_j a_{ij}x_j \geq b_i$. The parameter variables contain the coefficients a_{ij}, $j = 1, \ldots, n$, and the _RHS_ variable contains b_i. If the constraint i is active at the solution x^*, then _NAME_=ACTLC or _NAME_=LDACTLC.
GRAD	contains the gradient of the estimates.
GRAD_LF	contains the gradient of the Lagrange function. The _RHS_ variable contains the value of the Lagrange function.
HESSIAN	contains the Hessian matrix.
HESS_LF	contains the Hessian matrix of the Lagrange function (based on HESSIAN).
INFORMAT	contains the information matrix of the parameter estimates (only for METHOD=ML, METHOD=GLS, or METHOD=WLS).
INITIAL	contains the starting values of the parameter estimates.
JACNLC	contains the Jacobian of the nonlinear constraints evaluated at the final estimates.
JACOBIAN	contains the Jacobian matrix (only if the OUTJAC option is used).
LAGM BC	contains Lagrange multipliers for masks and active boundary constraints.

NAME	Description
GE	indicates the active lower bounds
LE	indicates the active upper bounds
EQ	indicates the active masks

Table 25.5 *continued*

TYPE	Description
LAGM LC	contains Lagrange multipliers for linear equality and active inequality constraints in pairs of observations containing the constraint number and the value of the Lagrange multiplier.

NAME	Description
LEC_NUM	number of the linear equality constraint
LEC_VAL	corresponding Lagrange multiplier value
LIC_NUM	number of the linear inequality constraint
LIC_VAL	corresponding Lagrange multiplier value

LAGM NLC	contains Lagrange multipliers for nonlinear equality and active inequality constraints in pairs of observations containing the constraint number and the value of the Lagrange multiplier.

NAME	Description
NLEC_NUM	number of the nonlinear equality constraint
NLEC_VAL	corresponding Lagrange multiplier value
NLIC_NUM	number of the linear inequality constraint
NLIC_VAL	corresponding Lagrange multiplier value

LE	If linear constraints are used, this observation contains the ith linear constraint $\sum_j a_{ij}x_j \leq b_i$. The parameter variables contain the coefficients a_{ij}, $j = 1, \ldots, n$, and the _RHS_ variable contains b_i. If the constraint i is active at the solution x^*, then _NAME_=ACTLC or _NAME_=LDACTLC.
LOWERBD | LB	If boundary constraints are used, this observation contains the lower bounds. Those parameters not subjected to lower bounds contain missing values. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.
NACTBC	All parameter variables contain the number n_{abc} of active boundary constraints at the solution x^*. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.
NACTLC	All parameter variables contain the number n_{alc} of active linear constraints at the solution x^* that are recognized as linearly independent. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.

Table 25.5 *continued*

TYPE	Description
NLC_EQ NLC_GE NLC_LE	contains values and residuals of nonlinear constraints. The _NAME_ variable is described as follows. **_NAME_ — Description** NLC — inactive nonlinear constraint NLCACT — linear independent active nonlinear constr. NLCACTLD — linear dependent active nonlinear constr.
NLDACTBC	contains the number of active boundary constraints at the solution x^* that are recognized as linearly dependent. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.
NLDACTLC	contains the number of active linear constraints at the solution x^* that are recognized as linearly dependent. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.
NOBS	contains the number of observations.
PARMS	contains the final parameter estimates. The _RHS_ variable contains the value of the objective function.
PCRPJ_LF	contains the projected Hessian matrix of the Lagrange function (based on CRPJAC).
PHESS_LF	contains the projected Hessian matrix of the Lagrange function (based on HESSIAN).
PROJCRPJ	contains the projected Hessian matrix (based on CRPJAC).
PROJGRAD	If linear constraints are used in the estimation, this observation contains the $n-n_{act}$ values of the projected gradient $g_Z = Z'g$ in the variables corresponding to the first $n - n_{act}$ parameters. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.
PROJHESS	contains the projected Hessian matrix (based on HESSIAN).
SIGSQ	contains the scalar factor of the covariance matrix of the parameter estimates.
STDERR	contains approximate standard errors (only for METHOD=ML, METHOD=GLS, or METHOD=WLS).
TERMINAT	The _NAME_ variable contains the name of the termination criterion.

Table 25.5 *continued*

TYPE	Description
UPPERBD | UB	If boundary constraints are used, this observation contains the upper bounds. Those parameters not subjected to upper bounds contain missing values. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.

If the technique specified by the TECH= option cannot be performed (for example, no feasible initial values can be computed, or the function value or derivatives cannot be evaluated at the starting point), the OUTEST= data set might contain only some of the observations (usually only the PARMS and GRAD observations).

OUTRAM= SAS-data-set

The OUTRAM= data set is of TYPE=RAM and contains the model specification and the computed parameter estimates. This data set is intended to be reused as an INRAM= data set to specify good initial values in a subsequent analysis by PROC CALIS.

The OUTRAM= data set contains the following variables:

- the BY variables, if any
- the character variable _TYPE_, which takes the values MODEL, ESTIM, VARNAME, METHOD, and STAT
- six additional variables whose meaning depends on the _TYPE_ of the observation

Each observation with _TYPE_=MODEL defines one matrix in the generalized COSAN model. The additional variables are as follows.

Table 25.6 Additional Variables When _TYPE_=MODEL

Variable	Contents
NAME	name of the matrix (character)
MATNR	number for the term and matrix in the model (numeric)
ROW	matrix row number (numeric)
COL	matrix column number (numeric)
ESTIM	first matrix type (numeric)
STDERR	second matrix type (numeric)

If the generalized COSAN model has only one matrix term, the _MATNR_ variable contains only the number of the matrix in the term. If there is more than one term, then it is the term number multiplied by 10,000 plus the matrix number (assuming that there are no more than 9,999 matrices specified in the COSAN model statement).

Each observation with _TYPE_=ESTIM defines one element of a matrix in the generalized COSAN model. The variables are used as follows.

Table 25.7 Additional Variables When _TYPE_=ESTIM

Variable	Contents
NAME	name of the parameter (character)
MATNR	term and matrix location of parameter (numeric)
ROW	row location of parameter (numeric)
COL	column location of parameter (numeric)
ESTIM	parameter estimate or constant value (numeric)
STDERR	standard error of estimate (numeric)

For constants rather than estimates, the _STDERR_ variable is 0. The _STDERR_ variable is missing for ULS and DWLS estimates if NOSTDERR is specified or if the approximate standard errors are not computed.

Each observation with _TYPE_=VARNAME defines a column variable name of a matrix in the generalized COSAN model.

The observations with _TYPE_=METHOD and _TYPE_=STAT are not used to build the model. The _TYPE_=METHOD observation contains the name of the estimation method used to compute the parameter estimates in the _NAME_ variable. If METHOD=NONE is not specified, the _ESTIM_ variable of the _TYPE_=STAT observations contains the information summarized in Table 25.8 (described in the section "Assessment of Fit" on page 928).

Table 25.8 _ESTIM_ Contents for _TYPE_=STAT

NAME	_ESTIM_
N	sample size
NPARM	number of parameters used in the model
DF	degrees of freedom
N_ACT	number of active boundary constraints for ML, GLS, and WLS estimation
FIT	fit function
GFI	goodness of fit index (GFI)
AGFI	adjusted GFI for degrees of freedom
RMR	root mean square residual
SRMR	standardized root mean square residual
PGFI	parsimonious GFI of Mulaik et al. (1989)
CHISQUAR	overall χ^2
P_CHISQ	probability $> \chi^2$
CHISQNUL	null (baseline) model χ^2
RMSEAEST	Steiger and Lind's (1980) RMSEA index estimate
RMSEALOB	lower range of RMSEA confidence interval
RMSEAUPB	upper range of RMSEA confidence interval
P_CLOSFT	Browne and Cudeck's (1993) probability of close fit
ECVI_EST	Browne and Cudeck's (1993) ECV index estimate

Table 25.8 *continued*

NAME	_ESTIM_
ECVI_LOB	lower range of ECVI confidence interval
ECVI_UPB	upper range of ECVI confidence interval
COMPFITI	Bentler's (1989) comparative fit index
ADJCHISQ	adjusted χ^2 for elliptic distribution
P_ACHISQ	probability corresponding adjusted χ^2
RLSCHISQ	reweighted least squares χ^2 (only ML estimation)
AIC	Akaike's information criterion
CAIC	Bozdogan's consistent information criterion
SBC	Schwarz's Bayesian criterion
CENTRALI	McDonald's centrality criterion
PARSIMON	Parsimonious index of James, Mulaik, and Brett
ZTESTWH	z test of Wilson and Hilferty
BB_NONOR	Bentler-Bonett (1980) nonnormed index ρ
BB_NORMD	Bentler-Bonett (1980) normed index Δ
BOL_RHO1	Bollen's (1986) normed index ρ_1
BOL_DEL2	Bollen's (1989a) nonnormed index Δ_2
CNHOELT	Hoelter's critical N index

You can edit the OUTRAM= data set to use its contents for initial estimates in a subsequent analysis by PROC CALIS, perhaps with a slightly changed model. But you should be especially careful for _TYPE_=MODEL when changing matrix types. The codes for the two matrix types are listed in Table 25.9.

Table 25.9 Matrix Type Codes

Code	First Matrix Type	Description
1:	IDE	identity matrix
2:	ZID	zero matrix concatenated by an identity matrix
3:	DIA	diagonal matrix
4:	ZDI	zero matrix concatenated by a diagonal matrix
5:	LOW	lower triangular matrix
6:	UPP	upper triangular matrix
7:		temporarily not used
8:	SYM	symmetric matrix
9:	GEN	general-type matrix
10:	BET	identity minus general-type matrix
11:	PER	selection matrix
12:		first matrix ($\mathbf{J}$) in LINEQS model statement
13:		second matrix ($\boldsymbol{\beta}$) in LINEQS model statement
14:		third matrix ($\boldsymbol{\gamma}$) in LINEQS model statement
Code	**Second Matrix Type**	**Description**
0:		noninverse model matrix
1:	INV	inverse model matrix
2:	IMI	"identity minus inverse" model matrix

OUTSTAT= SAS-data-set

The OUTSTAT= data set is similar to the TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set produced by the CORR procedure. The OUTSTAT= data set contains the following variables:

- the BY variables, if any
- two character variables, _TYPE_ and _NAME_
- the variables analyzed—that is, those in the VAR statement, or if there is no VAR statement, all numeric variables not listed in any other statement but used in the analysis. (**CAUTION:** Using the LINEQS or RAM model statements selects variables automatically.)

The OUTSTAT= data set contains the following information (when available):

- the mean and standard deviation
- the skewness and kurtosis (if the DATA= data set is a raw data set and the KURTOSIS option is specified)
- the number of observations
- if the WEIGHT statement is used, sum of the weights
- the correlation or covariance matrix to be analyzed
- the predicted correlation or covariance matrix
- the standardized or normalized residual correlation or covariance matrix
- if the model contains latent variables, the predicted covariances between latent and manifest variables, and the latent variable (or factor) score regression coefficients (see the PLATCOV display option on page 864)

In addition, if the FACTOR model statement is used, the OUTSTAT= data set contains the following:

- the unrotated factor loadings, the unique variances, and the matrix of factor correlations
- the rotated factor loadings and the transformation matrix of the rotation
- the matrix of standardized factor loadings

Each observation in the OUTSTAT= data set contains some type of statistic as indicated by the _TYPE_ variable. The values of the _TYPE_ variable are given in Table 25.10.

Table 25.10 _TYPE_ Observations in the OUTSTAT= Data Set

TYPE	Contents
MEAN	means
STD	standard deviations
USTD	uncorrected standard deviations
SKEWNESS	univariate skewness
KURTOSIS	univariate kurtosis
N	sample size
SUMWGT	sum of weights (if WEIGHT statement is used)
COV	covariances analyzed
CORR	correlations analyzed
UCOV	uncorrected covariances analyzed
UCORR	uncorrected correlations analyzed
ULSPRED	ULS predicted model values
GLSPRED	GLS predicted model values
MAXPRED	ML predicted model values
WLSPRED	WLS predicted model values
DWLSPRED	DWLS predicted model values
ULSNRES	ULS normalized residuals
GLSNRES	GLS normalized residuals
MAXNRES	ML normalized residuals
WLSNRES	WLS normalized residuals
DWLSNRES	DWLS normalized residuals
ULSSRES	ULS variance standardized residuals
GLSSRES	GLS variance standardized residuals
MAXSRES	ML variance standardized residuals
WLSSRES	WLS variance standardized residuals
DWLSSRES	DWLS variance standardized residuals
ULSASRES	ULS asymptotically standardized residuals
GLSASRES	GLS asymptotically standardized residuals
MAXASRES	ML asymptotically standardized residuals
WLSASRES	WLS asymptotically standardized residuals
DWLSASRS	DWLS asymptotically standardized residuals
UNROTATE	unrotated factor loadings
FCORR	matrix of factor correlations
UNIQUE_V	unique variances
TRANSFOR	transformation matrix of rotation
LOADINGS	rotated factor loadings
STD_LOAD	standardized factor loadings
LSSCORE	latent variable (or factor) score regression coefficients for ULS method
SCORE	latent variable (or factor) score regression coefficients other than ULS method

The _NAME_ variable contains the name of the manifest variable corresponding to each row for the covariance, correlation, predicted, and residual matrices and contains the name of the latent variable in case of factor regression scores. For other observations, _NAME_ is blank.

The unique variances and rotated loadings can be used as starting values in more difficult and constrained analyses.

If the model contains latent variables, the OUTSTAT= data set also contains the latent variable score regression coefficients and the predicted covariances between latent and manifest variables. You can use the latent variable score regression coefficients with PROC SCORE to compute latent variable or factor scores. For details, see the section "Latent Variable Scores" on page 937.

If the analyzed matrix is a (corrected or uncorrected) covariance rather than a correlation matrix, the _TYPE_=STD or _TYPE_=USTD observation is not included in the OUTSTAT= data set. In this case, the standard deviations can be obtained from the diagonal elements of the covariance matrix. Dropping the _TYPE_=STD or _TYPE_=USTD observation prevents PROC SCORE from standardizing the observations before computing the factor scores.

OUTWGT= SAS-data-set

You can create an OUTWGT= data set that is of TYPE=WEIGHT and contains the weight matrix used in generalized, weighted, or diagonally weighted least squares estimation. The *inverse* of the weight matrix is used in the corresponding fit function. The OUTWGT= data set contains the weight matrix to which the WRIDGE= and the WPENALTY= options are applied. For unweighted least squares or maximum likelihood estimation, no OUTWGT= data set can be written. The last weight matrix used in maximum likelihood estimation is the predicted model matrix (observations with _TYPE_=MAXPRED) that is included in the OUTSTAT= data set.

For generalized and diagonally weighted least squares estimation, the weight matrices $\mathbf{W}$ of the OUTWGT= data set contain all elements w_{ij}, where the indices i and j correspond to all manifest variables used in the analysis. Let $varnam_i$ be the name of the ith variable in the analysis. In this case, the OUTWGT= data set contains n observations with variables as displayed in the following table.

Table 25.11 Contents of OUTWGT= Data Set for GLS and DWLS Estimation

Variable	Contents
TYPE	WEIGHT (character)
NAME	name of variable $varnam_i$ (character)
$varnam_1$	weight w_{i1} for variable $varnam_1$ (numeric)
$\vdots$	$\vdots$
$varnam_n$	weight w_{in} for variable $varnam_n$ (numeric)

For weighted least squares estimation, the weight matrix $\mathbf{W}$ of the OUTWGT= data set contains only the nonredundant elements $w_{ij,kl}$. In this case, the OUTWGT= data set contains $n(n+1)(2n+1)/6$ observations with variables as in the following table.

Table 25.12 Contents of OUTWGT= Data Set for WLS Estimation

Variable	Contents
TYPE	WEIGHT (character)
NAME	name of variable $varnam_i$ (character)
NAM2	name of variable $varnam_j$ (character)
NAM3	name of variable $varnam_k$ (character)
$varnam_1$	weight $w_{ij,k1}$ for variable $varnam_1$ (numeric)
⋮	⋮
$varnam_n$	weight $w_{ij,kn}$ for variable $varnam_n$ (numeric)

Symmetric redundant elements are set to missing values.

Missing Values

If the DATA= data set contains raw data (rather than a covariance or correlation matrix), observations with missing values for any variables in the analysis are omitted from the computations. If a covariance or correlation matrix is read, missing values are allowed as long as every pair of variables has at least one nonmissing value.

Estimation Criteria

The following five estimation methods are available in PROC CALIS:

- unweighted least squares (ULS)
- generalized least squares (GLS)
- normal-theory maximum likelihood (ML)
- weighted least squares (WLS, ADF)
- diagonally weighted least squares (DWLS)

An INWGT= data set can be used to specify other than the default weight matrices $\mathbf{W}$ for GLS, WLS, and DWLS estimation.

PROC CALIS do not exhaust all estimation methods in the field. As mentioned in the section "Overview: CALIS Procedure" on page 828, partial least squares (PLS) is not implemented. The PLS method is developed under less restrictive statistical assumptions. It circumvents some computational and theoretical problems encountered by the existing estimation methods in PROC CALIS. However, PLS estimates are less efficient in general. When the statistical assumptions of PROC

CALIS are tenable (for example, large sample size, correct distributional assumptions, etc.), ML, GLS, or WLS methods yield better estimates than the PLS method. Note that there is a SAS/STAT procedure called PROC PLS, which employs the partial least squares technique but for a class of models different from those of PROC CALIS. For example, in a PROC CALIS model each latent variable is typically associated with only a subset of manifest variables (predictor or outcome variables). However, in PROC PLS latent variables are not prescribed with subsets of manifest variables. Rather, they are extracted from linear combinations of all manifest predictor variables. Therefore, for general path analysis with latent variables you should consider using PROC CALIS.

In each estimation method, the parameter vector is estimated iteratively by a nonlinear optimization algorithm that optimizes a goodness of fit function F. When n denotes the number of manifest variables, $\mathbf{S}$ denotes the given sample covariance or correlation matrix for a sample with size N, and $\mathbf{C}$ denotes the predicted moment matrix, then the fit function for unweighted least squares estimation is

$$F_{ULS} = 0.5Tr[(\mathbf{S} - \mathbf{C})^2]$$

For normal-theory generalized least squares estimation, the function is

$$F_{GLS} = 0.5Tr[(\mathbf{S}^{-1}(\mathbf{S} - \mathbf{C}))^2]$$

For normal-theory maximum likelihood estimation, the function is

$$F_{ML} = Tr(\mathbf{S}\mathbf{C}^{-1}) - n + ln(det(\mathbf{C})) - ln(det(\mathbf{S}))$$

The first three functions can be expressed by the generalized weighted least squares criterion (Browne 1982):

$$F_{GWLS} = 0.5Tr[(\mathbf{W}^{-1}(\mathbf{S} - \mathbf{C}))^2]$$

For unweighted least squares, the weight matrix $\mathbf{W}$ is chosen as the identity matrix $\mathbf{I}$; for generalized least squares, the default weight matrix $\mathbf{W}$ is the sample covariance matrix $\mathbf{S}$; and for normal-theory maximum likelihood, $\mathbf{W}$ is the iteratively updated predicted moment matrix $\mathbf{C}$. The values of the normal-theory maximum likelihood function F_{ML} and the generally weighted least squares criterion F_{GWLS} with $\mathbf{W} = \mathbf{C}$ are asymptotically equivalent.

The goodness of fit function that is minimized in weighted least squares estimation is

$$F_{WLS} = Vec(s_{ij} - c_{ij})'\mathbf{W}^{-1}Vec(s_{ij} - c_{ij})$$

where $Vec(s_{ij} - c_{ij})$ denotes the vector of the $n(n+1)/2$ elements of the lower triangle of the symmetric matrix $\mathbf{S} - \mathbf{C}$, and $\mathbf{W} = (w_{ij,kl})$ is a positive-definite symmetric matrix with $n(n+1)/2$ rows and columns.

If the moment matrix **S** is considered as a covariance rather than a correlation matrix, the default setting of $\mathbf{W} = (w_{ij,kl})$ is the consistent but biased estimators of the asymptotic covariances $\sigma_{ij,kl}$ of the sample covariance s_{ij} with the sample covariance s_{kl}, as defined in the following:

$$w_{ij,kl} = s_{ij,kl} - s_{ij}s_{kl}$$

where

$$s_{ij,kl} = \frac{1}{N}\sum_{r=1}^{N}(z_{ri} - \bar{z}_i)(z_{rj} - \bar{z}_j)(z_{rk} - \bar{z}_k)(z_{rl} - \bar{z}_l)$$

The formula of the asymptotic covariances of uncorrected covariances (using the UCOV or NOINT option) is a straightforward generalization of this expression.

The resulting weight matrix **W** is at least positive semidefinite (except for rounding errors). Using the ASYCOV option, you can use Browne's (1984, formula (3.8)) unbiased estimators

$$\begin{aligned} w_{ij,kl} &= \frac{N(N-1)}{(N-2)(N-3)}(s_{ij,kl} - s_{ij}s_{kl}) \\ &\quad -\frac{N}{(N-2)(N-3)}(s_{ik}s_{jl} + s_{il}s_{jk} - \frac{2}{N-1}s_{ij}s_{kl}) \end{aligned}$$

There is no guarantee that this weight matrix is positive semidefinite. However, the second part is of order $O(N^{-1})$ and does not destroy the positive semidefinite first part for sufficiently large N. For a large number of independent observations, default settings of the weight matrix **W** result in asymptotically distribution-free parameter estimates with unbiased standard errors and a correct χ^2 test statistic (Browne 1982, 1984).

If the moment matrix **S** is a correlation (rather than a covariance) matrix, the default setting of $\mathbf{W} = (w_{ij,kl})$ is the estimators of the asymptotic covariances $\sigma_{ij,kl}$ of the correlations $\mathbf{S} = (s_{ij})$ (Browne and Shapiro 1986; DeLeeuw 1983)

$$\begin{aligned} w_{ij,kl} &= r_{ij,kl} - \frac{1}{2}r_{ij}(r_{ii,kl} + r_{jj,kl}) - \frac{1}{2}r_{kl}(r_{kk,ij} + r_{ll,ij}) \\ &\quad +\frac{1}{4}r_{ij}r_{kl}(r_{ii,kk} + r_{ii,ll} + r_{jj,kk} + r_{jj,ll}) \end{aligned}$$

where

$$r_{ij,kl} = \frac{s_{ij,kl}}{\sqrt{s_{ii}s_{jj}s_{kk}s_{ll}}}$$

The asymptotic variances of the diagonal elements of a correlation matrix are 0. Therefore, the weight matrix computed by Browne and Shapiro's formula is always singular. In this case the goodness of fit function for weighted least squares estimation is modified to

$$F_{WLS} = \sum_{i=2}^{n}\sum_{j=1}^{i-1}\sum_{k=2}^{n}\sum_{l=1}^{k-1} w^{ij,kl}(s_{ij} - c_{ij})(s_{kl} - c_{kl}) + r\sum_{i}^{n}(s_{ii} - c_{ii})^2$$

where r is the penalty weight specified by the WPENALTY=r option and the $w^{ij,kl}$ are the elements of the inverse of the reduced $(n(n-1)/2)\times(n(n-1)/2)$ weight matrix that contains only the nonzero rows and columns of the full weight matrix $\mathbf{W}$. The second term is a penalty term to fit the diagonal elements of the moment matrix $\mathbf{S}$. The default value of $r = 100$ can be decreased or increased by the WPENALTY= option. The often used value of $r = 1$ seems to be too small in many cases to fit the diagonal elements of a correlation matrix properly. If your model does not fit the diagonal of the moment matrix $\mathbf{S}$, you can specify the NODIAG option to exclude the diagonal elements from the fit function.

Storing and inverting the huge weight matrix $\mathbf{W}$ in WLS estimation needs considerable computer resources. A compromise is found by implementing the DWLS method that uses only the diagonal of the weight matrix $\mathbf{W}$ from the WLS estimation in the minimization function

$$F_{DWLS} = Vec(s_{ij} - c_{ij})' diag(\mathbf{W})^{-1} Vec(s_{ij} - c_{ij})$$

The statistical properties of DWLS estimates are still not known.

In generalized, weighted, or diagonally weighted least squares estimation, you can change from the default settings of weight matrices $\mathbf{W}$ by using an INWGT= data set. Because the diagonal elements $w_{ii,kk}$ of the weight matrix $\mathbf{W}$ are interpreted as asymptotic variances of the sample covariances or correlations, they cannot be negative. The CALIS procedure requires a positive-definite weight matrix that has positive diagonal elements.

Relationships among Estimation Criteria

The five estimation functions, F_{ULS}, F_{GLS}, F_{ML}, F_{WLS}, and F_{DWLS}, belong to the following two groups:

- The functions F_{ULS}, F_{GLS}, and F_{ML} take into account all n^2 elements of the symmetric residual matrix $\mathbf{S} - \mathbf{C}$. This means that the off-diagonal residuals contribute twice to F, as lower and as upper triangular elements.
- The functions F_{WLS} and F_{DWLS} take into account only the $n(n+1)/2$ lower triangular elements of the symmetric residual matrix $\mathbf{S} - \mathbf{C}$. This means that the off-diagonal residuals contribute to F only once.

The F_{DWLS} function used in PROC CALIS differs from that used by the LISREL 7 program. Formula (1.25) of the LISREL 7 manual (Jöreskog and Sörbom 1988, p. 23) shows that LISREL groups the F_{DWLS} function in the first group by taking into account all n^2 elements of the symmetric residual matrix $\mathbf{S} - \mathbf{C}$.

- Relationship between DWLS and WLS:
  PROC CALIS: The F_{DWLS} and F_{WLS} estimation functions deliver the same results for the special case that the weight matrix $\mathbf{W}$ used by WLS estimation is a diagonal matrix.
  LISREL 7: This is not the case.

- Relationship between DWLS and ULS:
  LISREL 7: The F_{DWLS} and F_{ULS} estimation functions deliver the same results for the special case that the diagonal weight matrix **W** used by DWLS estimation is an identity matrix (contains only 1s).
  PROC CALIS: To obtain the same results with F_{DWLS} and F_{ULS} estimation, set the diagonal weight matrix **W** used in DWLS estimation to

$$w_{ik,ik} = \begin{cases} 1. & \text{if } i = k \\ 0.5 & \text{otherwise} \end{cases}$$

  Because the reciprocal elements of the weight matrix are used in the goodness of fit function, the off-diagonal residuals are weighted by a factor of 2.

Testing Rank Deficiency in the Approximate Covariance Matrix

The inverse of the information matrix (or approximate Hessian matrix) is used for the covariance matrix of the parameter estimates, which is needed for the computation of approximate standard errors and modification indices. The numerical condition of the information matrix (computed as the crossproduct $\mathbf{J}'\mathbf{J}$ of the Jacobian matrix **J**) can be very poor in many practical applications, especially for the analysis of unscaled covariance data. The following four-step strategy is used for the inversion of the information matrix.

1. The inversion (usually of a normalized matrix $\mathbf{D}^{-1}\mathbf{H}\mathbf{D}^{-1}$) is tried using a modified form of the Bunch and Kaufman (1977) algorithm, which allows the specification of a different singularity criterion for each pivot. The following three criteria for the detection of rank loss in the information matrix are used to specify thresholds:

   - *ASING* specifies absolute singularity.
   - *MSING* specifies relative singularity depending on the whole matrix norm.
   - *VSING* specifies relative singularity depending on the column matrix norm.

   If no rank loss is detected, the inverse of the information matrix is used for the covariance matrix of parameter estimates, and the next two steps are skipped.

2. The linear dependencies among the parameter subsets are displayed based on the singularity criteria.

3. If the number of parameters t is smaller than the value specified by the G4= option (the default value is 60), the Moore-Penrose inverse is computed based on the eigenvalue decomposition of the information matrix. If you do not specify the NOPRINT option, the distribution of eigenvalues is displayed, and those eigenvalues that are set to zero in the Moore-Penrose inverse are indicated. You should inspect this eigenvalue distribution carefully.

4. If PROC CALIS did not set the right subset of eigenvalues to zero, you can specify the COVSING= option to set a larger or smaller subset of eigenvalues to zero in a further run of PROC CALIS.

Approximate Standard Errors

Except for unweighted and diagonally weighted least squares estimation, approximate standard errors can be computed as the diagonal elements of the matrix

$$\frac{c}{NM}\mathbf{H}^{-1}, \quad \text{where}$$

$$NM = \begin{cases} (N-1) & \text{if the CORR or COV matrix is analyzed} \\ & \text{or the intercept variable is not used in the model} \\ N & \text{if the UCORR or UCOV matrix is analyzed} \\ & \text{and the intercept variable is not used in the model} \end{cases}$$

The matrix $\mathbf{H}$ is the approximate Hessian matrix of F evaluated at the final estimates, $c = 1$ for the WLS estimation method, $c = 2$ for the GLS and ML method, and N is the sample size. If a given correlation or covariance matrix is singular, PROC CALIS offers two ways to compute a generalized inverse of the information matrix and, therefore, two ways to compute approximate standard errors of implicitly constrained parameter estimates, t values, and modification indices. Depending on the G4= specification, either a Moore-Penrose inverse or a G2 inverse is computed. The expensive Moore-Penrose inverse computes an estimate of the null space by using an eigenvalue decomposition. The cheaper G2 inverse is produced by sweeping the linearly independent rows and columns and zeroing out the dependent ones. The information matrix, the approximate covariance matrix of the parameter estimates, and the approximate standard errors are not computed in the cases of unweighted or diagonally weighted least squares estimation.

Assessment of Fit

This section contains a collection of formulas used in computing indices to assess the goodness of fit by PROC CALIS. The following notation is used:

- N for the sample size
- n for the number of manifest variables
- t for the number of parameters to estimate
- $NM = \begin{cases} (N-1) & \text{if the CORR or COV matrix is analyzed} \\ & \text{or the intercept variable is not used in the model} \\ N & \text{if the UCORR or UCOV matrix is analyzed} \\ & \text{and the intercept variable is not used in the model} \end{cases}$
- df for the degrees of freedom
- $\boldsymbol{\gamma} = \mathbf{X}$ for the t vector of optimal parameter estimates
- $\mathbf{S} = (s_{ij})$ for the $n \times n$ input COV, CORR, UCOV, or UCORR matrix
- $\mathbf{C} = (c_{ij}) = \hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}(\hat{\boldsymbol{\gamma}})$ for the predicted model matrix

- $\mathbf{W}$ for the weight matrix ($\mathbf{W} = \mathbf{I}$ for ULS, $\mathbf{W} = \mathbf{S}$ for default GLS, and $\mathbf{W} = \mathbf{C}$ for ML estimates)
- $\mathbf{U}$ for the $n^2 \times n^2$ asymptotic covariance matrix of sample covariances
- $\Phi(x|\lambda, df)$ for the cumulative distribution function of the noncentral chi-squared distribution with noncentrality parameter λ

The following notation is for indices that support testing nested models by a χ^2 difference test:

- f_0 for the function value of the independence model
- df_0 for the degrees of freedom of the independence model
- $f_{min} = F$ for the function value of the fitted model
- $df_{min} = df$ for the degrees of freedom of the fitted model

The degrees of freedom df_{min} and the number of parameters t are adjusted automatically when there are active constraints in the analysis. The computation of many fit statistics and indices are affected. You can turn off the automatic adjustment by using the NOADJDF option. See the section "Counting the Degrees of Freedom" on page 956 for more information.

Residuals

PROC CALIS computes four types of residuals and writes them to the OUTSTAT= data set.

- **Raw Residuals**

  $$Res = \mathbf{S} - \mathbf{C}, Res_{ij} = s_{ij} - c_{ij}$$

  The raw residuals are displayed whenever the PALL, PRINT, or RESIDUAL option is specified.

- **Variance Standardized Residuals**

  $$VSRes_{ij} = \frac{s_{ij} - c_{ij}}{\sqrt{s_{ii} s_{jj}}}$$

  The variance standardized residuals are displayed when you specify the following:

  – the PALL, PRINT, or RESIDUAL option and METHOD=NONE, METHOD=ULS, or METHOD=DWLS
  – RESIDUAL=VARSTAND

  The variance standardized residuals are equal to those computed by the EQS 3 program (Bentler 1989).

- **Asymptotically Standardized Residuals**

$$ASRes_{ij} = \frac{s_{ij} - c_{ij}}{\sqrt{v_{ij,ij}}}, \quad \text{where}$$

$$v_{ij,ij} = diag(\mathbf{U} - \mathbf{J}Cov(\boldsymbol{\gamma})\mathbf{J}')_{ij}$$

The matrix $\mathbf{J}$ is the $n^2 \times t$ Jacobian matrix $d\,\boldsymbol{\Sigma}/d\,\boldsymbol{\gamma}$, and $Cov(\boldsymbol{\gamma})$ is the $t \times t$ asymptotic covariance matrix of parameter estimates (the inverse of the information matrix). Asymptotically standardized residuals are displayed when one of the following conditions is met:

  - The PALL, PRINT, or RESIDUAL option is specified, and METHOD=ML, METHOD=GLS, or METHOD=WLS, and the expensive information and Jacobian matrices are computed for some other reason.
  - RESIDUAL= ASYSTAND is specified.

The asymptotically standardized residuals are equal to those computed by the LISREL 7 program (Jöreskog and Sörbom 1988) except for the denominator NM in the definition of matrix $\mathbf{U}$.

- **Normalized Residuals**

$$NRes_{ij} = \frac{s_{ij} - c_{ij}}{\sqrt{u_{ij,ij}}}$$

where the diagonal elements $u_{ij,ij}$ of the $n^2 \times n^2$ asymptotic covariance matrix $\mathbf{U}$ of sample covariances are defined for the following methods.

  - **GLS** as $u_{ij,ij} = \frac{1}{NM}(s_{ii}s_{jj} + s_{ij}^2)$
  - **ML** as $u_{ij,ij} = \frac{1}{NM}(c_{ii}c_{jj} + c_{ij}^2)$
  - **WLS** as $u_{ij,ij} = \frac{1}{NM}w_{ij,ij}$

Normalized residuals are displayed when one of the following conditions is met:

  - The PALL, PRINT, or RESIDUAL option is specified, and METHOD=ML, METHOD=GLS, or METHOD=WLS, and the expensive information and Jacobian matrices are **not** computed for some other reason.
  - RESIDUAL=NORM is specified.

The normalized residuals are equal to those computed by the LISREL VI program (Jöreskog and Sörbom 1985) except for the definition of the denominator NM in matrix $\mathbf{U}$.

For estimation methods that are not BGLS estimation methods (Browne 1982, 1984), such as METHOD=NONE, METHOD=ULS, or METHOD=DWLS, the assumption of an asymptotic covariance matrix $\mathbf{U}$ of sample covariances does not seem to be appropriate. In this case, the normalized residuals should be replaced by the more relaxed variance standardized residuals. Computation of asymptotically standardized residuals requires computing the Jacobian and information matrices. This is computationally very expensive and is done only if the Jacobian matrix has to be computed for some other reason—that is, if at least one of the following items is true:

- The default, PRINT, or PALL displayed output is requested, and neither the NOMOD nor NOSTDERR option is specified.
- Either the MODIFICATION (included in PALL), PCOVES, or STDERR (included in default, PRINT, and PALL output) option is requested or RESIDUAL=ASYSTAND is specified.
- The LEVMAR or NEWRAP optimization technique is used.
- An OUTRAM= data set is specified without using the NOSTDERR option.
- An OUTEST= data set is specified without using the NOSTDERR option.

Since normalized residuals use an overestimate of the asymptotic covariance matrix of residuals (the diagonal of **U**), the normalized residuals cannot be larger than the asymptotically standardized residuals (which use the diagonal of $\mathbf{U} - \mathbf{J}Cov(\boldsymbol{\gamma})\mathbf{J}'$).

Together with the residual matrices, the values of the average residual, the average off-diagonal residual, and the rank order of the largest values are displayed. The distribution of the normalized and standardized residuals is displayed also.

Goodness of Fit Indices Based on Residuals

The following items are computed for all five kinds of estimation: ULS, GLS, ML, WLS, and DWLS. All these indices are written to the OUTRAM= data set. The goodness of fit (GFI), adjusted goodness of fit (AGFI), and root mean square residual (RMR) are computed as in the LISREL VI program of Jöreskog and Sörbom (1985).

- **Goodness of Fit Index**
  The goodness of fit index for the ULS, GLS, and ML estimation methods is

  $$GFI = 1 - \frac{Tr((\mathbf{W}^{-1}(\mathbf{S} - \mathbf{C}))^2)}{Tr((\mathbf{W}^{-1}\mathbf{S})^2)}$$

  but for WLS and DWLS estimation, it is

  $$GFI = 1 - \frac{Vec(s_{ij} - c_{ij})'\mathbf{W}^{-1}Vec(s_{ij} - c_{ij})}{Vec(s_{ij})'\mathbf{W}^{-1}Vec(s_{ij})}$$

  where $\mathbf{W} = diag$ for DWLS estimation, and $Vec(s_{ij} - c_{ij})$ denotes the vector of the $n(n+1)/2$ elements of the lower triangle of the symmetric matrix $\mathbf{S} - \mathbf{C}$. For a constant weight matrix **W**, the goodness of fit index is 1 minus the ratio of the minimum function value and the function value before any model has been fitted. The GFI should be between 0 and 1. The data probably do not fit the model if the GFI is negative or much larger than 1.
- **Adjusted Goodness of Fit Index**
  The AGFI is the GFI adjusted for the degrees of freedom of the model

  $$AGFI = 1 - \frac{n(n+1)}{2df}(1 - GFI)$$

The AGFI corresponds to the GFI in replacing the total sum of squares by the mean sum of squares.

CAUTION:

- Large n and small df can result in a negative AGFI. For example, GFI= 0.90, $n = 19$, and $df = 2$ result in an AGFI of -8.5.
- AGFI is not defined for a saturated model, due to division by $df = 0$.
- AGFI is not sensitive to losses in df.

The AGFI should be between 0 and 1. The data probably do not fit the model if the AGFI is negative or much larger than 1. For more information, refer to Mulaik et al. (1989).

- **Root Mean Square Residual**
  The RMR is the root of the mean of the squared residuals:

$$RMR = \sqrt{\frac{2}{n(n+1)} \sum_{i}^{n} \sum_{j}^{i} (s_{ij} - c_{ij})^2}$$

- **Standardized Root Mean Square Residual**
  The SRMR is the root of the mean of the squared standardized residuals:

$$SRMR = \sqrt{\frac{2}{n(n+1)} \sum_{i}^{n} \sum_{j}^{i} \frac{(s_{ij} - c_{ij})^2}{s_{ii} s_{jj}}}$$

- **Parsimonious Goodness of Fit Index**
  The PGFI (Mulaik et al. 1989) is a modification of the GFI that takes the parsimony of the model into account:

$$PGFI = \frac{df_{min}}{df_0} GFI$$

The PGFI uses the same parsimonious factor as the parsimonious normed Bentler-Bonett index (James, Mulaik, and Brett 1982).

Goodness of Fit Indices Based on the χ^2

The following items are transformations of the overall χ^2 value and in general depend on the sample size N. These indices are not computed for ULS or DWLS estimates.

- **Uncorrected χ^2**
  The overall χ^2 measure is the optimum function value F multiplied by $N - 1$ if a CORR or COV matrix is analyzed, or multiplied by N if a UCORR or UCOV matrix is analyzed. This gives the likelihood ratio test statistic for the null hypothesis that the predicted matrix $\mathbf{C}$ has the specified model structure against the alternative that $\mathbf{C}$ is unconstrained. The χ^2 test is valid only if the observations are independent and identically distributed, the analysis is based

on the nonstandardized sample covariance matrix **S**, and the sample size N is sufficiently large (Browne 1982; Bollen 1989b; Jöreskog and Sörbom 1985). For ML and GLS estimates, the variables must also have an approximately multivariate normal distribution. The notation Prob>Chi**2 means "the probability under the null hypothesis of obtaining a greater χ^2 statistic than that observed."

$$\chi^2 = NM * F$$

where F is the function value at the minimum.

- **χ_0^2 Value of the Independence Model**
  The χ_0^2 value of the independence model

  $$\chi_0^2 = NM * f_0$$

  and the corresponding degrees of freedom df_0 can be used (in large samples) to evaluate the gain of explanation by fitting the specific model (Bentler 1989).

- **RMSEA Index (Steiger and Lind 1980)**
  The Steiger and Lind (1980) root mean squared error approximation (RMSEA) coefficient is

  $$\epsilon_\alpha = \sqrt{\max(\frac{F}{df} - \frac{1}{NM}, 0)}$$

  The lower and upper limits of the confidence interval are computed using the cumulative distribution function of the noncentral chi-squared distribution $\Phi(x|\lambda, df) = \alpha$, with $x = NM * F$, λ_L satisfying $\Phi(x|\lambda_L, df) = 1 - \frac{\alpha}{2}$, and λ_U satisfying $\Phi(x|\lambda_U, df) = \frac{\alpha}{2}$:

  $$(\epsilon_{\alpha_L}; \epsilon_{\alpha_U}) = (\sqrt{\frac{\lambda_L}{NM * df}}; \sqrt{\frac{\lambda_U}{NM * df}})$$

  Refer to Browne and Du Toit (1992) for more details. The size of the confidence interval is defined by the option ALPHARMS=α, $0 \leq \alpha \leq 1$. The default is $\alpha = 0.1$, which corresponds to the 90% confidence interval for the RMSEA.

- **Probability for Test of Close Fit (Browne and Cudeck 1993)**
  The traditional exact χ^2 test hypothesis $H_0: \epsilon_\alpha = 0$ is replaced by the null hypothesis of close fit $H_0: \epsilon_\alpha \leq 0.05$ and the exceedance probability P is computed as

  $$P = 1 - \Phi(x|\lambda^*, df)$$

  where $x = NM * F$ and $\lambda^* = 0.05^2 * NM * df$. The null hypothesis of close fit is rejected if P is smaller than a prespecified level (for example, $P < 0.05$).

- **Expected Cross Validation Index (Browne and Cudeck 1993)**
  For GLS and WLS, the estimator c of the ECVI is linearly related to AIC:

  $$c = F(\mathbf{S}, \mathbf{C}) + \frac{2t}{NM}$$

  For ML estimation, c_{ML} is used.

  $$c_{ML} = F_{ML}(\mathbf{S}, \mathbf{C}) + \frac{2t}{NM - n - 1}$$

The confidence interval $(c_L; c_U)$ for c is computed using the cumulative distribution function $\Phi(x|\lambda, df)$ of the noncentral chi-squared distribution,

$$(c_L; c_U) = (\frac{\lambda_L + nnt}{NM}; \frac{\lambda_U + nnt}{NM})$$

with $nnt = n(n+1)/2 + t$, $x = NM * F$, $\Phi(x|\lambda_U, df) = 1 - \frac{\alpha}{2}$, and $\Phi(x|\lambda_L, df) = \frac{\alpha}{2}$. The confidence interval $(c_L^*; c_U^*)$ for c_{ML} is

$$(c_L^*; c_U^*) = (\frac{\lambda_L^* + nnt}{NM - n - 1}); \frac{\lambda_U^* + nnt}{NM - n - 1})$$

where $nnt = n(n+1)/2+t$, $x = (NM-n-1)*F$, $\Phi(x|\lambda_U^*, df) = \frac{\alpha}{2}$ and $\Phi(x|\lambda_L^*, df) = 1 - \frac{\alpha}{2}$. Refer to Browne and Cudeck (1993) for details. The size of the confidence interval is defined by the option ALPHAECV=α, $0 \leq \alpha \leq 1$. The default is $\alpha = 0.1$, which corresponds to the 90% confidence interval for the ECVI.

- **Comparative Fit Index (Bentler 1989)**

$$CFI = 1 - \frac{\max(NM * f_{min} - df_{min}, 0)}{\max(NM * f_0 - df_0, 0)}$$

- **Adjusted χ^2 Value (Browne 1982)**
  If the variables are n-variate elliptic rather than normal and have significant amounts of multivariate kurtosis (leptokurtic or platykurtic), the χ^2 value can be adjusted to

$$\chi^2_{ell} = \frac{\chi^2}{\eta_2}$$

  where η_2 is the multivariate relative kurtosis coefficient.

- **Normal Theory Reweighted LS χ^2 Value**
  This index is displayed only if METHOD=ML. Instead of the function value F_{ML}, the reweighted goodness of fit function F_{GWLS} is used,

$$\chi^2_{GWLS} = NM * F_{GWLS}$$

  where F_{GWLS} is the value of the function at the minimum.

- **Akaike's Information Criterion (AIC) (Akaike 1974; Akaike 1987)**
  This is a criterion for selecting the best model among a number of candidate models. The model that yields the smallest value of AIC is considered the best.

$$AIC = \chi^2 - 2df$$

- **Consistent Akaike's Information Criterion (CAIC) (Bozdogan 1987)**
  This is another criterion, similar to AIC, for selecting the best model among alternatives. The model that yields the smallest value of CAIC is considered the best. CAIC is preferred by some people to AIC or the χ^2 test.

$$CAIC = \chi^2 - (ln(N) + 1)df$$

- **Schwarz's Bayesian Criterion (SBC) (Schwarz 1978; Sclove 1987)**
  This is another criterion, similar to AIC, for selecting the best model. The model that yields the smallest value of SBC is considered the best. SBC is preferred by some people to AIC or the χ^2 test.

  $$SBC = \chi^2 - ln(N)df$$

- **McDonald's Measure of Centrality (McDonald 1989)**

  $$CENT = exp(-\frac{(\chi^2 - df)}{2N})$$

- **Parsimonious Normed Fit Index (James, Mulaik, and Brett 1982)**
  The PNFI is a modification of Bentler-Bonett's normed fit index that takes parsimony of the model into account,

  $$PNFI = \frac{df_{min}}{df_0}\frac{(f_0 - f_{min})}{f_0}$$

  The PNFI uses the same parsimonious factor as the parsimonious GFI of Mulaik et al. (1989).

- **Z-Test (Wilson and Hilferty 1931)**
  The Z-test of Wilson and Hilferty assumes an n-variate normal distribution:

  $$Z = \frac{\sqrt[3]{\frac{\chi^2}{df}} - (1 - \frac{2}{9df})}{\sqrt{\frac{2}{9df}}}$$

  Refer to McArdle (1988) and Bishop, Fienberg, and Holland (1977, p. 527) for an application of the Z-test.

- **Nonnormed Coefficient (Bentler and Bonett 1980)**

  $$\rho = \frac{f_0/df_0 - f_{min}/df_{min}}{f_0/df_0 - 1/NM}$$

  Refer to Tucker and Lewis (1973) for details.

- **Normed Coefficient (Bentler and Bonett 1980)**

  $$\Delta = \frac{f_0 - f_{min}}{f_0}$$

  Mulaik et al. (1989) recommend the parsimonious weighted form PNFI.

- **Normed Index ρ_1 (Bollen 1986)**

  $$\rho_1 = \frac{f_0/df_0 - f_{min}/df_{min}}{f_0/df_0}$$

  ρ_1 is always less than or equal to 1; $\rho_1 < 0$ is unlikely in practice. Refer to the discussion in Bollen (1989a) for details.

- **Nonnormed Index Δ_2 (Bollen 1989a)**

$$\Delta_2 = \frac{f_0 - f_{min}}{f_0 - \frac{df}{NM}}$$

is a modification of Bentler and Bonett's Δ that uses df and "lessens the dependence" on N. Refer to the discussion in Bollen (1989b). Δ_2 is identical to Mulaik et al.'s (1989) IFI2 index.

- **Critical N Index (Hoelter 1983)**

$$CN = \frac{\chi^2_{crit}}{F} + 1$$

where χ^2_{crit} is the critical chi-square value for the given df degrees of freedom and probability $\alpha = 0.05$, and F is the value of the estimation criterion (minimization function). Refer to Bollen (1989b, p. 277) for details. Hoelter (1983) suggests that CN should be at least 200; however, Bollen (1989b) notes that the CN value might lead to an overly pessimistic assessment of fit for small samples.

Squared Multiple Correlation

The following are measures of the squared multiple correlation for manifest and endogenous variables and are computed for all five estimation methods: ULS, GLS, ML, WLS, and DWLS. These coefficients are computed as in the LISREL VI program of Jöreskog and Sörbom (1985). The DETAE, DETSE, and DETMV determination coefficients are intended to be global means of the squared multiple correlations for different subsets of model equations and variables. These coefficients are displayed only when you specify the PDETERM option with a RAM or LINEQS model.

- **R^2 Values Corresponding to Endogenous Variables**

$$R^2_i = 1 - \frac{\widehat{var(\boldsymbol{\xi}_i)}}{\widehat{var(\boldsymbol{\eta}_i)}}$$

- **Total Determination of All Equations**

$$DETAE = 1 - \frac{det(\hat{\boldsymbol{\Theta}}, \hat{\boldsymbol{\Psi}})}{det(\widehat{Cov(y, x, \boldsymbol{\eta})})}$$

- **Total Determination of the Structural Equations**

$$DETSE = 1 - \frac{det(\hat{\boldsymbol{\Psi}})}{det(\widehat{Cov(\boldsymbol{\eta})})}$$

- **Total Determination of the Manifest Variables**

$$DETMV = 1 - \frac{det(\hat{\boldsymbol{\Theta}})}{det(\mathbf{S})}$$

CAUTION: In the LISREL program, the structural equations are defined by specifying the BETA matrix. In PROC CALIS, a structural equation has a dependent left-hand-side variable that appears at least once on the right-hand side of another equation, or the equation has at least one right-hand-side variable that is the left-hand-side variable of another equation. Therefore, PROC CALIS sometimes identifies more equations as structural equations than the LISREL program does.

Latent Variable Scores

Although latent variable or factor scores are unobserved, they can be estimated after you fit your model using PROC CALIS. Latent variable or factor scores are estimated as linear combinations of observed variables, weighted by the latent variable (factor) score regression coefficients.

To display the latent variable (factor) score regression coefficients in the PROC CALIS output, you can use the PLATCOV option. You can also save these coefficients in an output data set by using the OUTSTAT= option. These coefficients can then be used by the SCORE procedure to compute the latent variable (factor) scores.

To summarize, do the following if you need to compute latent variable (factor) scores for each observation by using the SCORE procedure:

- Create an output data set with the OUTSTAT= option in the PROC CALIS statement.
- Use the SCORE procedure with both the raw data and the OUTSTAT= data set.

For example, you can use the following statements to compute the latent variable (factor) scores, which are stored in the OUT= data set named "**scores**":

```
proc calis data=raw outstat=ostat;
   lineqs
      v1 = a1 f1 + e1,
      v2 = a2 f1 + e2,
      v3 = a3 f1 + e3;
   std
      f1 = 1.,
      e1-e3 = evar1-evar3;
run;

proc score data=raw score=ostat out=scores;
run;
```

If you analyze a correlation or covariance matrix in PROC CALIS, the original raw data must be used in the PROC SCORE procedure, as shown in the following statements:

```
proc corr data=raw out=correl;
run;

proc calis data=correl outstat=ostat;
   lineqs
      v1 = a1 f1 + e1,
      v2 = a2 f1 + e2,
      v3 = a3 f1 + e3;
   std
      f1 = 1.,
      e1-e3 = evar1-evar3;
run;

proc score data=raw score=ostat out=scores;
run;
```

For a more detailed example, see Example 76.1 in Chapter 76, "The SCORE Procedure." Although the FACTOR procedure is used in that example, it is also applicable to the CALIS procedure. Essentially, an OUTSTAT= data set containing the latent variable or factor score regression coefficients, either from PROC FACTOR or PROC CALIS, is used with the raw data in the SCORE procedure to create the latent variable or factor scores.

Measures of Multivariate Kurtosis

In many applications, the manifest variables are not even approximately multivariate normal. If this happens to be the case with your data set, the default generalized least squares and maximum likelihood estimation methods are not appropriate, and you should compute the parameter estimates and their standard errors by an asymptotically distribution-free method, such as the WLS estimation method. If your manifest variables are multivariate normal, then they have a zero relative multivariate kurtosis, and all marginal distributions have zero kurtosis (Browne 1982). If your DATA= data set contains raw data, PROC CALIS computes univariate skewness and kurtosis and a set of multivariate kurtosis values. By default, the values of univariate skewness and kurtosis are corrected for bias (as in PROC UNIVARIATE), but using the BIASKUR option enables you to compute the uncorrected values also. The values are displayed when you specify the PROC CALIS statement option KURTOSIS.

- **Corrected Variance for Variable z_j**

$$\sigma_j^2 = \frac{1}{N-1}\sum_i^N (z_{ij} - \bar{z}_j)^2$$

- **Corrected Univariate Skewness for Variable z_j**

$$\gamma_{1(j)} = \frac{N}{(N-1)(N-2)} \frac{\sum_i^N (z_{ij} - \bar{z}_j)^3}{\sigma_j^3}$$

- **Uncorrected Univariate Skewness for Variable z_j**

$$\gamma_{1(j)} = \frac{N \sum_i^N (z_{ij} - \bar{z}_j)^3}{\sqrt{N[\sum_i^N (z_{ij} - \bar{z}_j)^2]^3}}$$

- **Corrected Univariate Kurtosis for Variable z_j**

$$\gamma_{2(j)} = \frac{N(N+1)}{(N-1)(N-2)(N-3)} \frac{\sum_i^N (z_{ij} - \bar{z}_j)^4}{\sigma_j^4} - \frac{3(N-1)^2}{(N-2)(N-3)}$$

- **Uncorrected Univariate Kurtosis for Variable z_j**

$$\gamma_{2(j)} = \frac{N\sum_i^N (z_{ij} - \bar{z}_j)^4}{[\sum_i^N (z_{ij} - \bar{z}_j)^2]^2} - 3$$

- **Mardia's Multivariate Kurtosis**

$$\gamma_2 = \frac{1}{N}\sum_i^N [(z_i - \bar{z})' S^{-1} (z_i - \bar{z})]^2 - n(n+2)$$

- **Relative Multivariate Kurtosis**

$$\eta_2 = \frac{\gamma_2 + n(n+2)}{n(n+2)}$$

- **Normalized Multivariate Kurtosis**

$$\kappa_0 = \frac{\gamma_2}{\sqrt{8n(n+2)/N}}$$

- **Mardia-Based Kappa**

$$\kappa_1 = \frac{\gamma_2}{n(n+2)}$$

- **Mean Scaled Univariate Kurtosis**

$$\kappa_2 = \frac{1}{3n}\sum_j^n \gamma_{2(j)}$$

- **Adjusted Mean Scaled Univariate Kurtosis**

$$\kappa_3 = \frac{1}{3n}\sum_{j}^{n}\gamma^*_{2(j)}$$

with

$$\gamma^*_{2(j)} = \begin{cases} \gamma_{2(j)} & , \quad if \quad \gamma_{2(j)} > \frac{-6}{n+2} \\ \frac{-6}{n+2} & , \quad \text{otherwise} \end{cases}$$

If variable Z_j is normally distributed, the uncorrected univariate kurtosis $\gamma_{2(j)}$ is equal to 0. If Z has an n-variate normal distribution, Mardia's multivariate kurtosis γ_2 is equal to 0. A variable Z_j is called *leptokurtic* if it has a positive value of $\gamma_{2(j)}$ and is called *platykurtic* if it has a negative value of $\gamma_{2(j)}$. The values of κ_1, κ_2, and κ_3 should not be smaller than a lower bound (Bentler 1985):

$$\hat{\kappa} \geq \frac{-2}{n+2}$$

PROC CALIS displays a message if this happens.

If weighted least squares estimates (METHOD=WLS or METHOD=ADF) are specified and the weight matrix is computed from an input raw data set, the CALIS procedure computes two more measures of multivariate kurtosis.

- **Multivariate Mean Kappa**

$$\kappa_4 = \frac{1}{m}\sum_{i}^{n}\sum_{j}^{i}\sum_{k}^{j}\sum_{l}^{k}\hat{\kappa}_{ij,kl} - 1$$

where

$$\hat{\kappa}_{ij,kl} = \frac{s_{ij,kl}}{s_{ij}s_{kl} + s_{ik}s_{jl} + s_{il}s_{jk}}$$

and $m = n(n+1)(n+2)(n+3)/24$ is the number of elements in the vector $s_{ij,kl}$ (Bentler 1985).

- **Multivariate Least Squares Kappa**

$$\kappa_5 = \frac{s_4' s_2}{s_2' s_2} - 1$$

where

$$s_{ij,kl} = \frac{1}{N}\sum_{r=1}^{N}(z_{ri} - \bar{z}_i)(z_{rj} - \bar{z}_j)(z_{rk} - \bar{z}_k)(z_{rl} - \bar{z}_l)$$

s_4 is the vector of the $s_{ij,kl}$, and s_2 is the vector of the elements in the denominator of $\hat{\kappa}$ (Bentler 1985).

The occurrence of significant nonzero values of Mardia's multivariate kurtosis γ_2 and significant amounts of some of the univariate kurtosis values $\gamma_{2(j)}$ indicate that your variables are not multivariate normal distributed. Violating the multivariate normality assumption in (default) generalized least squares and maximum likelihood estimation usually leads to the wrong approximate standard errors and incorrect fit statistics based on the χ^2 value. In general, the parameter estimates are more stable against violation of the normal distribution assumption. For more details, refer to Browne (1974, 1982, 1984).

Initial Estimates

Each optimization technique requires a set of initial values for the parameters. To avoid local optima, the initial values should be as close as possible to the globally optimal solution. You can check for local optima by running the analysis with several different sets of initial values; the RANDOM= option in the PROC CALIS statement is useful in this regard.

- RAM and LINEQS: There are several default estimation methods available in PROC CALIS for initial values of parameters in a linear structural equation model specified by a RAM or LINEQS model statement, depending on the form of the specified model.
  - two-stage least squares estimation
  - instrumental variable method (Hägglund 1982; Jennrich 1987)
  - approximative factor analysis method
  - ordinary least squares estimation
  - estimation method of McDonald (McDonald and Hartmann 1992)
- FACTOR: For default (exploratory) factor analysis, PROC CALIS computes initial estimates for factor loadings and unique variances by an algebraic method of approximate factor analysis. If you use a MATRIX statement together with a FACTOR model specification, initial values are computed by McDonald's (McDonald and Hartmann 1992) method if possible. McDonald's method of computing initial values works better if you scale the factors by setting the factor variances to 1 rather than setting the loadings of the reference variables equal to 1. If neither of the two methods seems to be appropriate, the initial values are set by the START= option.
- COSAN: For the more general COSAN model, there is no default estimation method for the initial values. In this case, the START= or RANDOM= option can be used to set otherwise unassigned initial values.

Poor initial values can cause convergence problems, especially with maximum likelihood estimation. You should not specify a constant initial value for all parameters since this would produce a singular predicted model matrix in the first iteration. Sufficiently large positive diagonal elements in the central matrices of each model matrix term provide a nonnegative definite initial predicted model matrix. If maximum likelihood estimation fails to converge, it might help to use METHOD=LSML, which uses the final estimates from an unweighted least squares analysis as initial estimates for

maximum likelihood. Or you can fit a slightly different but better-behaved model and produce an OUTRAM= data set, which can then be modified in accordance with the original model and used as an INRAM= data set to provide initial values for another analysis.

If you are analyzing a covariance or scalar product matrix, be sure to take into account the scales of the variables. The default initial values might be inappropriate when some variables have extremely large or small variances.

Automatic Variable Selection

You can use the VAR statement to reorder the variables in the model and to delete the variables not used. Using the VAR statement saves memory and computation time. If a linear structural equation model that uses the RAM or LINEQS statement (or an INRAM= data set specifying a RAM or LINEQS model) does not use all the manifest variables given in the input DATA= data set, PROC CALIS automatically deletes those manifest variables not used in the model.

In some special circumstances, the automatic variable selection performed for the RAM and LINEQS statements might be inappropriate—for example, if you are interested in modification indices connected to some of the variables that are not used in the model. You can include such manifest variables as exogenous variables in the analysis by specifying constant zero coefficients.

For example, the first three steps in a stepwise regression analysis of the Werner blood chemistry data (Jöreskog and Sörbom 1988, p. 111) can be performed as follows:

```
proc calis data=dixon method=gls nobs=180 print mod;
   lineqs y=0 x1+0 x2+0 x3+0 x4+0 x5+0 x6+0 x7+e;
   std    e=var;
run;
proc calis data=dixon method=gls nobs=180 print mod;
   lineqs y=g1 x1+0 x2+0 x3+0 x4+0 x5+0 x6+0 x7+e;
   std    e=var;
run;
proc calis data=dixon method=gls nobs=180 print mod;
   lineqs y=g1 x1+0 x2+0 x3+0 x4+0 x5+g6 x6+0 x7+e;
   std    e=var;
run;
```

Using the COSAN statement does not automatically delete those variables from the analysis that are not used in the model. You can use the output of the predetermined values in the predicted model matrix (PREDET option) to detect unused variables. Variables that are not used in the model are indicated by 0 in the rows and columns of the predetermined predicted model matrix.

Exogenous Manifest Variables

If there are exogenous manifest variables in the linear structural equation model, then there is a one-to-one relationship between the given covariances and corresponding estimates in the central model matrix (**P** or **Φ**). In general, using exogenous manifest variables reduces the degrees of freedom since the corresponding sample correlations or covariances are not part of the exogenous information provided for the parameter estimation. See the section "Counting the Degrees of Freedom" on page 956 for more information.

If you specify a RAM or LINEQS model statement, or if such a model is recognized in an INRAM= data set, those elements in the central model matrices that correspond to the exogenous manifest variables are reset to the sample values after computing covariances or correlations within the current BY group.

The COSAN statement does not automatically set the covariances in the central model matrices that correspond to manifest exogenous variables.

You can use the output of the predetermined values in the predicted model matrix (PREDET option) that correspond to manifest exogenous variables to see which of the manifest variables are exogenous variables and to help you set the corresponding locations of the central model matrices with their covariances.

The following two examples show how different the results of PROC CALIS can be if manifest variables are considered as either endogenous or exogenous variables. (See Figure 25.5.) In both examples, a correlation matrix **S** is tested against an identity model matrix **C**; that is, no parameter is estimated. The following three runs of the first example (specified by the COSAN, LINEQS, and RAM statements) consider the two variables y and x as endogenous variables:

```
title2 'Data: FULLER (1987, p.18)';
data corn;
   input y x;
   datalines;
 86  70
115  97
 90  53
 86  64
110  95
 91  64
 99  50
 96  70
 99  94
104  69
 96  51
;
```

```
title3 'Endogenous Y and X';
proc calis data=corn;
   cosan corr(2,ide);
run;
proc calis data=corn;
   lineqs
          y=ey,
          x=ex;
   std    ey ex=2 * 1;
run;
proc calis data=corn;
   ram
       1  1  3  1.,
       1  2  4  1.,
       2  3  3  1.,
       2  4  4  1.;
run;
```

The following two runs of the second example (specified by the LINEQS and RAM statements) consider y and x as exogenous variables:

```
title3 'Exogenous Y and X';
proc calis data=corn;
   std y x=2 * 1;
run;
proc calis data=corn;
   ram
       2  1  1  1.,
       2  2  2  1.;
run;
```

Figure 25.5 Exogenous and Endogenous Variables



Exogenous x, y Endogenous x, y

The LINEQS and the RAM model statements set the covariances (correlations) of exogenous manifest variables in the estimated model matrix and automatically reduce the degrees of freedom.

Use of Optimization Techniques

No algorithm for optimizing general nonlinear functions exists that will always find the global optimum for a general nonlinear minimization problem in a reasonable amount of time. Since no single optimization technique is invariably superior to others, PROC CALIS provides a variety of optimization techniques that work well in various circumstances. However, you can devise problems for which none of the techniques in PROC CALIS will find the correct solution. All optimization techniques in PROC CALIS use $O(n^2)$ memory except the conjugate gradient methods, which use only $O(n)$ of memory and are designed to optimize problems with many parameters.

The PROC CALIS statement NLOPTIONS can be especially helpful for tuning applications with nonlinear equality and inequality constraints on the parameter estimates. Some of the options available in NLOPTIONS might also be invoked as PROC CALIS options. The NLOPTIONS statement can specify almost the same options as the SAS/OR NLP procedure.

Nonlinear optimization requires the repeated computation of the following:

- the function value (optimization criterion)
- the gradient vector (first-order partial derivatives)
- for some techniques, the (approximate) Hessian matrix (second-order partial derivatives)
- values of linear and nonlinear constraints
- the first-order partial derivatives (Jacobian) of nonlinear constraints

For the criteria used by PROC CALIS, computing the gradient takes more computer time than computing the function value, and computing the Hessian takes *much* more computer time and memory than computing the gradient, especially when there are many parameters to estimate. Unfortunately, optimization techniques that do not use the Hessian usually require many more iterations than techniques that do use the (approximate) Hessian, and so they are often slower. Techniques that do not use the Hessian also tend to be less reliable (for example, they might terminate at local rather than global optima).

The available optimization techniques are displayed in Table 25.13 and can be chosen by the TECH=*name* option.

Table 25.13 Optimization Techniques

TECH=	Optimization Technique
LEVMAR	Levenberg-Marquardt method
TRUREG	Trust-region method
NEWRAP	Newton-Raphson method with line search
NRRIDG	Newton-Raphson method with ridging
QUANEW	Quasi-Newton methods (DBFGS, DDFP, BFGS, DFP)
DBLDOG	Double-dogleg method (DBFGS, DDFP)
CONGRA	Conjugate gradient methods (PB, FR, PR, CD)

Table 25.14 shows, for each optimization technique, which derivatives are needed (first-order or second-order) and what kind of constraints (boundary, linear, or nonlinear) can be imposed on the parameters.

Table 25.14 Derivatives Needed and Constraints Allowed

	Derivatives		Constraints		
TECH=	**First Order**	**Second Order**	**Boundary**	**Linear**	**Nonlinear**
LEVMAR	x	x	x	x	-
TRUREG	x	x	x	x	-
NEWRAP	x	x	x	x	-
NRRIDG	x	x	x	x	-
QUANEW	x	-	x	x	x
DBLDOG	x	-	x	x	-
CONGRA	x	-	x	x	-

The Levenberg-Marquardt, trust-region, and Newton-Raphson techniques are usually the most reliable, work well with boundary and general linear constraints, and generally converge after a few iterations to a precise solution. However, these techniques need to compute a Hessian matrix in each iteration. For HESSALG=1, this means that you need about $4(n(n+1)/2)t$ bytes of work memory (n = the number of manifest variables, t = the number of parameters to estimate) to store the Jacobian and its crossproduct. With HESSALG=2 or HESSALG=3, you do not need this work memory, but the use of a utility file increases execution time. Computing the approximate Hessian in each iteration can be very time- and memory-consuming, especially for large problems (more than 60 or 100 parameters, depending on the computer used). For large problems, a quasi-Newton technique, especially with the BFGS update, can be far more efficient.

For a poor choice of initial values, the Levenberg-Marquardt method seems to be more reliable.

If memory problems occur, you can use one of the conjugate gradient techniques, but they are generally slower and less reliable than the methods that use second-order information.

There are several options to control the optimization process. First of all, you can specify various termination criteria. You can specify the GCONV= option to specify a relative gradient termination criterion. If there are active boundary constraints, only those gradient components that correspond to inactive constraints contribute to the criterion. When you want very precise parameter estimates, the GCONV= option is useful. Other criteria that use relative changes in function values or parameter estimates in consecutive iterations can lead to early termination when active constraints cause small steps to occur. The small default value for the FCONV= option helps prevent early termination. Using the MAXITER= and MAXFUNC= options enables you to specify the maximum number of iterations and function calls in the optimization process. These limits are especially useful in combination with the INRAM= and OUTRAM= options; you can run a few iterations at a time, inspect the results, and decide whether to continue iterating.

Nonlinearly Constrained QN Optimization

The algorithm used for nonlinearly constrained quasi-Newton optimization is an efficient modification of Powell's (1978a, 1978b, 1982a, 1982b) *Variable Metric Constrained WatchDog* (VMCWD) algorithm. A similar but older algorithm (VF02AD) is part of the Harwell library. Both VMCWD and VF02AD use Fletcher's VE02AD algorithm (also part of the Harwell library) for positive-definite quadratic programming. The PROC CALIS QUANEW implementation uses a quadratic programming subroutine that updates and downdates the approximation of the Cholesky factor when the active set changes. The nonlinear QUANEW algorithm is not a feasible-point algorithm, and the value of the objective function need not decrease (minimization) or increase (maximization) monotonically. Instead, the algorithm tries to reduce a linear combination of the objective function and constraint violations, called the *merit function*.

The following are similarities and differences between this algorithm and VMCWD:

- A modification of this algorithm can be performed by specifying VERSION=1, which replaces the update of the Lagrange vector μ with the original update of Powell (1978a, 1978b), which is used in VF02AD. This can be helpful for some applications with linearly dependent active constraints.
- If the VERSION= option is not specified or VERSION=2 is specified, the evaluation of the Lagrange vector μ is performed in the same way as Powell (1982a, 1982b) describes.
- Instead of updating an approximate Hessian matrix, this algorithm uses the dual BFGS (or DFP) update that updates the Cholesky factor of an approximate Hessian. If the condition of the updated matrix gets too bad, a restart is done with a positive diagonal matrix. At the end of the first iteration after each restart, the Cholesky factor is scaled.
- The Cholesky factor is loaded into the quadratic programming subroutine, automatically ensuring positive-definiteness of the problem. During the quadratic programming step, the Cholesky factor of the projected Hessian matrix $\mathbf{Z}_k'\mathbf{G}\mathbf{Z}_k$ and the QT decomposition are updated simultaneously when the active set changes. Refer to Gill et al. (1984) for more information.
- The line-search strategy is very similar to that of Powell (1982a, 1982b). However, this algorithm does not call for derivatives during the line search; hence, it generally needs fewer derivative calls than function calls. The VMCWD algorithm always requires the same number of derivative and function calls. It was also found in several applications of VMCWD that Powell's line-search method sometimes uses steps that are too long during the first iterations. In those cases, you can use the INSTEP= option specification to restrict the step length α of the first iterations.
- Also the watchdog strategy is similar to that of Powell (1982a, 1982b). However, this algorithm does not return automatically after a fixed number of iterations to a former better point. A return here is further delayed if the observed function reduction is close to the expected function reduction of the quadratic model.
- Although Powell's termination criterion still is used (as FCONV2), the QUANEW implementation uses two additional termination criteria (GCONV and ABSGCONV).

This algorithm is automatically invoked when you specify the NLINCON statement. The nonlinear QUANEW algorithm needs the Jacobian matrix of the first-order derivatives (constraints normals) of the constraints

$$(\nabla c_i) = (\frac{\partial c_i}{\partial x_j}), \quad i = 1, \ldots, nc, j = 1, \ldots, n$$

where nc is the number of nonlinear constraints for a given point x.

You can specify two update formulas with the UPDATE= option:

- UPDATE=DBFGS performs the dual BFGS update of the Cholesky factor of the Hessian matrix. This is the default.
- UPDATE=DDFP performs the dual DFP update of the Cholesky factor of the Hessian matrix.

This algorithm uses its own line-search technique. All options and parameters (except the INSTEP= option) controlling the line search in the other algorithms do not apply here. In several applications, large steps in the first iterations are troublesome. You can specify the INSTEP= option to impose an upper bound for the step size α during the first five iterations. The values of the LCSINGULAR=, LCEPSILON=, and LCDEACT= options, which control the processing of linear and boundary constraints, are valid only for the quadratic programming subroutine used in each iteration of the nonlinear constraints QUANEW algorithm.

Optimization and Iteration History

The optimization and iteration histories are displayed by default because it is important to check for possible convergence problems.

The optimization history includes the following summary of information about the initial state of the optimization:

- the number of constraints that are active at the starting point, or more precisely, the number of constraints that are currently members of the working set. If this number is followed by a plus sign, there are more active constraints, of which at least one is temporarily released from the working set due to negative Lagrange multipliers.
- the value of the objective function at the starting point
- if the (projected) gradient is available, the value of the largest absolute (projected) gradient element
- for the TRUREG and LEVMAR subroutines, the initial radius of the trust-region around the starting point

The optimization history ends with some information concerning the optimization result:

- the number of constraints that are active at the final point, or more precisely, the number of constraints that are currently members of the working set. If this number is followed by a plus sign, there are more active constraints, of which at least one is temporarily released from the working set due to negative Lagrange multipliers.

- the value of the objective function at the final point
- if the (projected) gradient is available, the value of the largest absolute (projected) gradient element
- other information specific to the optimization technique

The iteration history generally consists of one line of displayed output containing the most important information for each iteration. The _LIST_ variable (see the section "SAS Programming Statements" on page 903) also enables you to display the parameter estimates and the gradient in some or all iterations.

The iteration history always includes the following (the words in parentheses are the column header output):

- the iteration number (Iter)
- the number of iteration restarts (rest)
- the number of function calls (nfun)
- the number of active constraints (act)
- the value of the optimization criterion (optcrit)
- the difference between adjacent function values (difcrit)
- the maximum of the absolute gradient components corresponding to inactive boundary constraints (maxgrad)

An apostrophe trailing the number of active constraints indicates that at least one of the active constraints is released from the active set due to a significant Lagrange multiplier.

For the Levenberg-Marquardt technique (LEVMAR), the iteration history also includes the following information:

- An asterisk trailing the iteration number means that the computed Hessian approximation is singular and consequently ridged with a positive lambda value. If all or the last several iterations show a singular Hessian approximation, the problem is not sufficiently identified. Thus, there are other locally optimal solutions that lead to the same optimum function value for different parameter values. This implies that standard errors for the parameter estimates are not computable without the addition of further constraints.
- the value of the Lagrange multiplier (lambda); this is 0 if the optimum of the quadratic function approximation is inside the trust-region (a trust-region-scaled Newton step can be performed) and is greater than 0 when the optimum of the quadratic function approximation is located at the boundary of the trust-region (the scaled Newton step is too long to fit in the trust-region and a quadratic constraint optimization is performed). Large values indicate optimization difficulties. For a nonsingular Hessian matrix, the value of lambda should go to 0 during the last iterations, indicating that the objective function can be well approximated by a quadratic function in a small neighborhood of the optimum point. An increasing lambda value often indicates problems in the optimization process.

- the value of the ratio ρ (rho) between the actually achieved difference in function values and the predicted difference in the function values on the basis of the quadratic function approximation. Values much less than 1 indicate optimization difficulties. The value of the ratio ρ indicates the goodness of the quadratic function approximation; in other words, $\rho << 1$ means that the radius of the trust-region has to be reduced. A fairly large value of ρ means that the radius of the trust region need not be changed. And a value close to or larger than 1 means that the radius can be increased, indicating a good quadratic function approximation.

For the Newton-Raphson technique (NRRIDG), the iteration history also includes the following information:

- the value of the ridge parameter. This is 0 when a Newton step can be performed, and it is greater than 0 when either the Hessian approximation is singular or a Newton step fails to reduce the optimization criterion. Large values indicate optimization difficulties.
- the value of the ratio ρ (rho) between the actually achieved difference in function values and the predicted difference in the function values on the basis of the quadratic function approximation. Values much less than 1.0 indicate optimization difficulties.

For the Newton-Raphson with line-search technique (NEWRAP), the iteration history also includes the following information:

- the step size α (alpha) computed with one of the line-search algorithms
- the slope of the search direction at the current parameter iterate. For minimization, this value should be significantly negative. Otherwise, the line-search algorithm has difficulty reducing the function value sufficiently.

For the trust-region technique (TRUREG), the iteration history also includes the following information:

- An asterisk after the iteration number means that the computed Hessian approximation is singular and consequently ridged with a positive lambda value.
- the value of the Lagrange multiplier (lambda). This value is zero when the optimum of the quadratic function approximation is inside the trust-region (a trust-region-scaled Newton step can be performed) and is greater than zero when the optimum of the quadratic function approximation is located at the boundary of the trust-region (the scaled Newton step is too long to fit in the trust-region and a quadratically constrained optimization is performed). Large values indicate optimization difficulties. As in Gay (1983), a negative lambda value indicates the special case of an indefinite Hessian matrix (the smallest eigenvalue is negative in minimization).
- the value of the radius Δ of the trust-region. Small trust-region radius values combined with large lambda values in subsequent iterations indicate optimization problems.

For the quasi-Newton (QUANEW) and conjugate gradient (CONGRA) techniques, the iteration history also includes the following information:

- the step size (alpha) computed with one of the line-search algorithms
- the descent of the search direction at the current parameter iterate. This value should be significantly smaller than 0. Otherwise, the line-search algorithm has difficulty reducing the function value sufficiently.

Frequent update restarts (rest) of a quasi-Newton algorithm often indicate numerical problems related to required properties of the approximate Hessian update, and they decrease the speed of convergence. This can happen particularly if the ABSGCONV= termination criterion is too small—that is, when the requested precision cannot be obtained by quasi-Newton optimization. Generally, the number of automatic restarts used by conjugate gradient methods is much higher.

For the nonlinearly constrained quasi-Newton technique, the iteration history also includes the following information:

- the maximum value of all constraint violations,

$$\text{conmax} = \max(|c_i(x)| : c_i(x) < 0)$$

- the value of the predicted function reduction used with the GCONV and FCONV2 termination criteria,

$$\text{pred} = |g(x^{(k)})s(x^{(k)})| + \sum_{i=1}^{m} |\lambda_i c_i(x^{(k)})|$$

- the step size α of the quasi-Newton step. Note that this algorithm works with a special line-search algorithm.
- the maximum element of the gradient of the Lagrange function,

$$\begin{aligned} \text{lfgmax} &= \nabla_x L(x^{(k)}, \lambda^{(k)}) \\ &= \nabla_x f(x^{(k)}) - \sum_{i=1}^{m} \lambda_i^{(k)} \nabla_x c_i(x^{(k)}) \end{aligned}$$

For the double-dogleg technique, the iteration history also includes the following information:

- the parameter λ of the double-dogleg step. A value $\lambda = 0$ corresponds to the full (quasi) Newton step.
- the slope of the search direction at the current parameter iterate. For minimization, this value should be significantly negative.

Line-Search Methods

In each iteration k, the (dual) quasi-Newton, hybrid quasi-Newton, conjugate gradient, and Newton-Raphson minimization techniques use iterative line-search algorithms that try to optimize a linear, quadratic, or cubic approximation of the nonlinear objective function f of n parameters x along a feasible descent search direction $s^{(k)}$

$$f(x^{(k+1)}) = f(x^{(k)} + \alpha^{(k)} s^{(k)})$$

by computing an approximately optimal scalar $\alpha^{(k)} > 0$. Since the outside iteration process is based only on the approximation of the objective function, the inside iteration of the line-search algorithm does not have to be perfect. Usually, it is satisfactory that the choice of α significantly reduces (in a minimization) the objective function. Criteria often used for termination of line-search algorithms are the Goldstein conditions (Fletcher 1987).

Various line-search algorithms can be selected by using the LIS= option. The line-search methods LIS=1, LIS=2, and LIS=3 satisfy the left-hand-side and right-hand-side Goldstein conditions (refer to Fletcher 1987). When derivatives are available, the line-search methods LIS=6, LIS=7, and LIS=8 try to satisfy the right-hand-side Goldstein condition; if derivatives are not available, these line-search algorithms use only function calls.

The line-search method LIS=2 seems to be superior when function evaluation consumes significantly less computation time than gradient evaluation. Therefore, LIS=2 is the default value for Newton-Raphson, (dual) quasi-Newton, and conjugate gradient optimizations.

Restricting the Step Length

Almost all line-search algorithms use iterative extrapolation techniques that can easily lead to feasible points where the objective function f is no longer defined (resulting in indefinite matrices for ML estimation) or is difficult to compute (resulting in floating-point overflows). Therefore, PROC CALIS provides options that restrict the step length or trust-region radius, especially during the first main iterations.

The inner product $g's$ of the gradient g and the search direction s is the slope of $f(\alpha) = f(x + \alpha s)$ along the search direction s with step length α. The default starting value $\alpha^{(0)} = \alpha^{(k,0)}$ in each line-search algorithm ($\min_{\alpha>0} f(x + \alpha s)$) during the main iteration k is computed in three steps.

1. Use either the difference $df = |f^{(k)} - f^{(k-1)}|$ of the function values during the last two consecutive iterations or the final step-size value α^- of the previous iteration $k-1$ to compute a first value $\alpha_1^{(0)}$.

   - Using the DAMPSTEP<=*r*> option:

     $$\alpha_1^{(0)} = \min(1, r\alpha^-)$$

     The initial value for the new step length can be no larger than r times the final step length α^- of the previous iteration. The default is $r = 2$.

- Not using the DAMPSTEP option:

$$\alpha_1^{(0)} = \begin{cases} step & \text{if } 0.1 \leq step \leq 10 \\ 10 & \text{if } step > 10 \\ 0.1 & \text{if } step < 0.1 \end{cases}$$

with

$$step = \begin{cases} df/|g's| & \text{if } |g's| \geq \epsilon \max(100df, 1) \\ 1 & \text{otherwise} \end{cases}$$

This value of $\alpha_1^{(0)}$ can be too large and can lead to a difficult or impossible function evaluation, especially for highly nonlinear functions such as the EXP function.

2. During the first five iterations, the second step enables you to reduce $\alpha_1^{(0)}$ to a smaller starting value $\alpha_2^{(0)}$ using the INSTEP=r option:

$$\alpha_2^{(0)} = \min(\alpha_1^{(0)}, r)$$

After more than five iterations, $\alpha_2^{(0)}$ is set to $\alpha_1^{(0)}$.

3. The third step can further reduce the step length by

$$\alpha_3^{(0)} = \min(\alpha_2^{(0)}, \min(10, u))$$

where u is the maximum length of a step inside the feasible region.

The INSTEP=r option lets you specify a smaller or larger radius of the trust region used in the first iteration by the trust-region, double-dogleg, and Levenberg-Marquardt algorithms. The default initial trust-region radius is the length of the scaled gradient (Moré 1978). This step corresponds to the default radius factor of $r = 1$. This choice is successful in most practical applications of the TRUREG, DBLDOG, and LEVMAR algorithms. However, for bad initial values used in the analysis of a covariance matrix with high variances, or for highly nonlinear constraints (such as using the EXP function) in your programming code, the default start radius can result in arithmetic overflows. If this happens, you can try decreasing values of INSTEP=r, $0 < r < 1$, until the iteration starts successfully. A small factor r also affects the trust-region radius of the next steps because the radius is changed in each iteration by a factor $0 < c \leq 4$ depending on the ρ ratio. Reducing the radius corresponds to increasing the ridge parameter λ that produces smaller steps directed closer toward the gradient direction.

Modification Indices

While fitting structural models, you might want to modify the specified model in order to do the following:

- reduce the χ^2 value significantly
- reduce the number of parameters to estimate without increasing the χ^2 value too much

If you specify the MODIFICATION or MOD option, PROC CALIS computes and displays a default set of modification indices:

- **Univariate Lagrange multiplier test indices** for most elements in the model matrices that are constrained to *equal constants*. These are second-order approximations of the decrease in the χ^2 value that would result from allowing the constant matrix element to vary. Besides the value of the Lagrange multiplier, the corresponding probability ($df = 1$) and the approximate change of the parameter value (should the constant be changed to a parameter) are displayed. If allowing the constant to be a free estimated parameter would result in a singular information matrix, the string 'sing' is displayed instead of the Lagrange multiplier index. Not all elements in the model matrices should be allowed to vary; the diagonal elements of the inverse matrices in the RAM or LINEQS model must be constant ones. The univariate Lagrange multipliers are displayed at the constant locations of the model matrices.

- **Univariate Wald test indices** for those matrix elements that correspond to *parameter estimates* in the model. These are second-order approximations of the increase in the χ^2 value that would result from constraining the parameter to a 0 constant. The univariate Wald test indices are the same as the squared t values that are displayed together with the parameter estimates and standard errors. The univariate Wald test indices are displayed at the parameter locations of the model matrices.

- **Univariate Lagrange multiplier test indices** that are second-order approximations of the decrease in the χ^2 value that would result from the release of *equality constraints*. Multiple equality constraints containing $n > 2$ parameters are tested successively in n steps, each assuming the release of one of the equality-constrained parameters. The expected change of the parameter values of the separated parameter and the remaining parameter cluster are displayed, too.

- **Univariate Lagrange multiplier test indices** for releasing *active boundary constraints* specified by the BOUNDS statement

- **Stepwise multivariate Wald test indices** for constraining estimated parameters to 0 are computed and displayed. In each step, the parameter that would lead to the smallest increase in the multivariate χ^2 value is set to 0. Besides the multivariate χ^2 value and its probability, the univariate increments are also displayed. The process stops when the univariate probability is smaller than the specified value in the SLMW= option.

All of the preceding tests are approximations. You can often get more accurate tests by actually fitting different models and computing likelihood ratio tests. For more details about the Wald and the Lagrange multiplier test, refer to MacCallum (1986), Buse (1982), Bentler (1986), or Lee (1985).

Note that, for large model matrices, the computation time for the default modification indices can considerably exceed the time needed for the minimization process.

The modification indices are not computed for unweighted least squares or diagonally weighted least squares estimation.

CAUTION: Modification indices are not computed if the model matrix is an identity matrix (IDE or ZID), a selection matrix (PER), or the first matrix **J** in the LINEQS model. If you want to display the modification indices for such a matrix, you should specify the matrix as another type; for example,

specify an identity matrix used in the COSAN statement as a diagonal matrix with constant diagonal elements of 1.

Constrained Estimation by Using Program Code

The CALIS procedure offers a very flexible way to constrain parameter estimates. You can use your own programming statements to express special properties of the parameter estimates. This tool is also present in McDonald's COSAN implementation but is considerably easier to use in the CALIS procedure. PROC CALIS is able to compute analytic first- and second-order derivatives that you would have to specify using the COSAN program. There are also three PROC CALIS statements you can use:

- the BOUNDS statement, to specify simple bounds on the parameters used in the optimization process
- the LINCON statement, to specify general linear equality and inequality constraints on the parameters used in the optimization process
- the NLINCON statement, to specify general nonlinear equality and inequality constraints on the parameters used in the optimization process. The variables listed in the NLINCON statement must be specified in the program code.

There are some traditional ways to enforce parameter constraints by using parameter transformations (McDonald 1980).

- **One-sided boundary constraints:** For example, the parameter q_k should be at least as large (or at most as small) as a given constant value a (or b),

  $$q_k \geq a \qquad \text{or} \qquad q_k \leq b$$

  This inequality constraint can be expressed as an equality constraint

  $$q_k = a + x_j^2 \qquad \text{or} \qquad q_k = b - x_j^2$$

  in which the fundamental parameter x_j is unconstrained.

- **Two-sided boundary constraints:** For example, the parameter q_k should be located between two given constant values a and b, $a < b$,

  $$a \leq q_k \leq b$$

  This inequality constraint can be expressed as an equality constraint

  $$q_k = a + (b - a)\frac{exp(x_j)}{1 + exp(x_j)}$$

  in which the fundamental parameter x_j is unconstrained.

- **One-sided order constraints:** For example, the parameters $q_1, \ldots, q_k$ should be ordered in the form

  $$q_1 \leq q_2, \quad q_1 \leq q_3, \quad \ldots, \quad q_1 \leq q_k$$

  These inequality constraints can be expressed as a set of equality constraints

  $$q_1 = x_1, \quad q_2 = x_1 + x_2^2, \quad \ldots, \quad q_k = x_1 + x_k^2$$

  in which the fundamental parameters $x_1, \ldots, x_k$ are unconstrained.

- **Two-sided order constraints:** For example, the parameters $q_1, \ldots, q_k$ should be ordered in the form

  $$q_1 \leq q_2 \leq q_3 \leq \ldots \leq q_k$$

  These inequality constraints can be expressed as a set of equality constraints

  $$q_1 = x_1, \quad q_2 = q_1 + x_2^2, \quad \ldots, \quad q_k = q_{k-1} + x_k^2$$

  in which the fundamental parameters $x_1, \ldots, x_k$ are unconstrained.

- **Linear equation constraints:** For example, the parameters q_1, q_2, q_3 should be linearly constrained in the form

  $$q_1 + q_2 + q_3 = a$$

  which can be expressed in the form of three explicit equations in which the fundamental parameters x_1 and x_2 are unconstrained:

  $$q_1 = x_1, \quad q_2 = x_2, \quad q_3 = a - x_1 - x_2$$

Refer to McDonald (1980) and Browne (1982) for further notes on reparameterizing techniques. If the optimization problem is not too large to apply the Levenberg-Marquardt or Newton-Raphson algorithm, boundary constraints should be requested by the BOUNDS statement rather than by reparameterizing code. If the problem is so large that you must use a quasi-Newton or conjugate gradient algorithm, reparameterizing techniques might be more efficient than the BOUNDS statement.

Counting the Degrees of Freedom

In a regression problem, the number of degrees of freedom for the error estimate is the number of observations in the data set minus the number of parameters. The NOBS=, DFR= (RDF=), and DFE= (EDF=) options refer to degrees of freedom in this sense. However, these values are not related to the degrees of freedom of a test statistic used in a covariance or correlation structure analysis. The NOBS=, DFR=, and DFE= options should be used in PROC CALIS to specify only the effective number of observations in the input DATA= data set.

In general, the number of degrees of freedom in a covariance or correlation structure analysis is defined as the difference between the number of nonredundant values q in the observed $n \times n$

correlation or covariance matrix **S** and the number t of free parameters **X** used in the fit of the specified model, $df = q - t$. Both values, q and t, are counted differently in different situations by PROC CALIS.

The number of nonredundant values q is generally equal to the number of lower triangular elements in the $n \times n$ moment matrix **S** including all diagonal elements, minus a constant c dependent upon special circumstances,

$$q = n(n+1)/2 - c$$

The number c is evaluated by adding the following quantities:

- If you specify a linear structural equation model containing exogenous manifest variables by using the RAM or LINEQS statement, PROC CALIS adds to c the number of variances and covariances among these manifest exogenous variables, which are automatically set in the corresponding locations of the central model matrices (see the section "Exogenous Manifest Variables" on page 943).
- If you specify the DFREDUCE=i option, PROC CALIS adds the specified number i to c. The number i can be a negative integer.
- If you specify the NODIAG option to exclude the fit of the diagonal elements of the data matrix **S**, PROC CALIS adds the number n of diagonal elements to c.
- If all the following conditions hold, then PROC CALIS adds to c the number of the diagonal locations:
  - NODIAG and DFREDUC= options are not specified.
  - A correlation structure is being fitted.
  - The predicted correlation matrix contains constants on the diagonal.

In some complicated models, especially those use programming statements, PROC CALIS might not be able to detect all the constant predicted values. In such cases, you must specify the DFREDUCE= option to get the correct degrees of freedom.

The number t is the number of different parameter names used in constructing the model if you do not use programming statements to impose constraints on the parameters. Using programming statements in general introduces two kinds of parameters:

- independent parameters, which are used only on the right-hand side of the expressions
- dependent parameters, which are used at least once on the left-hand side of the expressions

The independent parameters belong to the parameters involved in the estimation process, whereas the dependent parameters are fully defined by the programming statements and can be computed from the independent parameters. In this case, the number t is the number of different parameter names used in the model specification, but not used in the programming statements, plus the number of independent parameters. The independent parameters and their initial values can be defined in a model specification statement or in a PARMS statement.

The degrees of freedom are automatically increased by the number of active constraints in the solution. Similarly, the number of parameters are decreased by the number of active constraints. This affects the computation of many fit statistics and indices. Refer to Dijkstra (1992) for a discussion of the validity of statistical inferences with active boundary constraints. If the researcher believes that the active constraints will have a small chance of occurrence in repeated sampling, it might be more suitable to turn off the automatic adjustment by using the NOADJDF option.

Computational Problems

First Iteration Overflows

Analyzing a covariance matrix including high variances in the diagonal and using bad initial estimates for the parameters can easily lead to arithmetic overflows in the first iterations of the minimization algorithm. The line-search algorithms that work with cubic extrapolation are especially sensitive to arithmetic overflows. If this occurs with quasi-Newton or conjugate gradient minimization, you can specify the INSTEP= option to reduce the length of the first step. If an arithmetic overflow occurs in the first iteration of the Levenberg-Marquardt algorithm, you can specify the INSTEP= option to reduce the trust-region radius of the first iteration. You also can change the minimization technique or the line-search method. If none of these help, you should consider the following:

- scaling the covariance matrix
- providing better initial values
- changing the model

No Convergence of Minimization Process

If convergence does not occur during the minimization process, perform the following tasks.

- If there are *negative variance estimates* in the diagonal locations of the central model matrices, you can do the following:
  - Specify the BOUNDS statement to obtain nonnegative variance estimates.
  - Specify the HEYWOOD option, if the FACTOR model statement is specified.
- Change the estimation method to obtain a better set of initial estimates. For example, if you use METHOD=ML, you can do the following:
  - Change to METHOD=LSML.
  - Run some iterations with METHOD=DWLS or METHOD=GLS, write the results in an OUTRAM= data set, and use the results as initial values specified by an INRAM= data set in a second run with METHOD=ML.

- Change the optimization technique. For example, if you use the default TECH=LEVMAR, you can do the following:
  - Change to TECH=QUANEW or TECH=NEWRAP.
  - Run some iterations with TECH=CONGRA, write the results in an OUTRAM= data set, and use the results as initial values specified by an INRAM= data set in a second run with a different TECH= technique.
- Change or modify the update technique or the line-search algorithm, or both, when using TECH=QUANEW or TECH=CONGRA. For example, if you use the default update formula and the default line-search algorithm, you can do the following:
  - Change the update formula with the UPDATE= option.
  - Change the line-search algorithm with the LIS= option.
  - Specify a more precise line search with the LSPRECISION= option, if you use LIS=2 or LIS=3.
- Add more iterations and function calls by using the MAXIT= and MAXFU= options.
- Change the initial values. For many categories of model specifications done by the LINEQS, RAM, or FACTOR model, PROC CALIS computes an appropriate set of initial values automatically. However, for some of the model specifications (for example, structural equations with latent variables on the left-hand side and manifest variables on the right-hand side), PROC CALIS can generate very obscure initial values. In these cases, you have to set the initial values yourself.
  - Increase the initial values of the parameters located at the diagonal of central matrices in one of the following ways:
    - manually, by setting the values in the model specification
    - automatically, by using the DEMPHAS= option
  - Use a slightly different, but more stable, model to obtain preliminary estimates.
  - Use additional information to specify initial values, such as by using other SAS software like the FACTOR, REG, SYSLIN, and MODEL (SYSNLIN) procedures for the modified, unrestricted model case.

Unidentified Model

The parameter vector $\mathbf{x}$ in the covariance structure model

$$\mathbf{C} = \mathbf{C}(\mathbf{x})$$

is said to be identified in a parameter space G, if

$$\mathbf{C}(\mathbf{x}) = \mathbf{C}(\tilde{\mathbf{x}}), \quad \tilde{\mathbf{x}} \in G$$

implies $\mathbf{x} = \tilde{\mathbf{x}}$. The parameter estimates that result from an unidentified model can be very far from the parameter estimates of a very similar but identified model. They are usually machine dependent. Do not use parameter estimates of an unidentified model as initial values for another run of PROC CALIS.

Singular Predicted Model Matrix

You can easily specify models with singular predicted model matrices, such as by fixing diagonal elements of central matrices to 0. In such cases, you cannot compute maximum likelihood estimates (the ML function value F is not defined). Since singular predicted model matrices can also occur temporarily in the minimization process, PROC CALIS tries in such cases to change the parameter estimates so that the predicted model matrix becomes positive-definite. In such cases, the following message is displayed:

```
NOTE: Parameter set changed.
```

This process does not always work well, especially if there are fixed instead of variable diagonal elements in the central model matrices. A famous example where you cannot compute ML estimates is a component analysis with fewer components than given manifest variables. See the section "FACTOR Model Statement" on page 877 for more details. If you continue to get a singular predicted model matrix after changing initial values and optimization techniques, then your model is perhaps specified so that ML estimates cannot be computed.

Saving Computing Time

For large models, the most computing time is needed to compute the modification indices. If you do not really need the Lagrange multipliers or multiple Wald test indices (the univariate Wald test indices are the same as the t values), using the NOMOD option can save a considerable amount of computing time.

Central Matrices with Negative Eigenvalues

A covariance matrix cannot have negative eigenvalues, since a negative eigenvalue means that some linear combination of the variables has negative variance. PROC CALIS displays a warning if a central model matrix has negative eigenvalues but does not actually compute the eigenvalues. Sometimes this warning can be triggered by 0 or very small positive eigenvalues that appear negative because of numerical error. If you want to be sure that the central model matrix you are fitting can be considered to be a variance-covariance matrix, you can use the SAS/IML command *VAL=EIGVAL(U)* to compute the vector *VAL* of eigenvalues of matrix **U**.

Negative R^2 Values

The estimated squared multiple correlations R^2 of the endogenous variables are computed using the estimated error variances

$$R_i^2 = 1 - \frac{\widehat{var(\zeta_i)}}{\widehat{var(\eta_i)}}$$

If the model is a poor fit, it is possible that $\widehat{var(\zeta_i)} > \widehat{var(\eta_i)}$, which results in $R_i^2 < 0$.

Displayed Output

The output displayed by PROC CALIS depends on the statement used to specify the model. Since an analysis requested by the LINEQS or RAM statement implies the analysis of a structural equation model, more statistics can be computed and displayed than for a covariance structure analysis following the generalized COSAN model requested by the COSAN statement. The displayed output resulting from use of the FACTOR statement includes all the COSAN displayed output as well as more statistics displayed only when you specify the FACTOR statement. Since the displayed output by using the RAM statement differs only in its form from that generated by the LINEQS statement, in this section distinctions are made between COSAN and LINEQS output only.

The unweighted least squares and diagonally weighted least squares estimation methods do not provide a sufficient statistical basis to provide the following output (neither displayed nor written to an OUTEST= data set):

- most of the fit indices
- approximate standard errors
- normalized or asymptotically standardized residuals
- modification indices
- information matrix
- covariance matrix of parameter estimates

The notation $\mathbf{S} = (s_{ij})$ is used for the analyzed covariance or correlation matrix, $\mathbf{C} = (c_{ij})$ for the predicted model matrix, $\mathbf{W}$ for the weight matrix (for example, $\mathbf{W} = \mathbf{I}$ for ULS, $\mathbf{W} = \mathbf{S}$ for GLS, $\mathbf{W} = \mathbf{C}$ for ML estimates), $\mathbf{X}$ for the vector of optimal parameter estimates, n for the number of manifest variables, t for the number of parameter estimates, and N for the sample size.

The output of PROC CALIS includes the following:

- COSAN and LINEQS: List of the matrices and their properties specified by the generalized COSAN model if you specify at least the PSHORT option.
- LINEQS: List of manifest variables that are not used in the specified model and that are automatically omitted from the analysis. Note that there is no automatic variable reduction with the COSAN or FACTOR statement. If necessary, you should use the VAR statement in these cases.
- LINEQS: List of the endogenous and exogenous variables specified by the LINEQS, STD, and COV statements if you specify at least the PSHORT option.
- COSAN: Initial values of the parameter matrices indicating positions of constants and parameters. The output, or at least the default output, is displayed if you specify the PINITIAL option.

- LINEQS: The set of structural equations containing the initial values and indicating constants and parameters, and output of the initial error variances and covariances. The output, or at least the default output, is displayed if you specify the PINITIAL option.
- COSAN and LINEQS: The weight matrix **W** is displayed if GLS, WLS, or DWLS estimation is used and you specify the PWEIGHT or PALL option.
- COSAN and LINEQS: General information about the estimation problem: number of observations (N), number of manifest variables (n), amount of independent information in the data matrix (information, $n(n+1)/2$), number of terms and matrices in the specified generalized COSAN model, and number of parameters to be estimated (parameters, t). If there are no exogenous manifest variables, the difference between the amount of independent information ($n(n+1)/2$) and the number of requested estimates (t) is equal to the degrees of freedom (df). A necessary condition for a model to be identified is that the degrees of freedom are nonnegative. The output, or at least the default output, is displayed if you specify the SIMPLE option.
- COSAN and LINEQS: Mean and Std Dev (standard deviation) of each variable if you specify the SIMPLE option, as well as skewness and kurtosis if the DATA= data set is a raw data set and you specify the KURTOSIS option.
- COSAN and LINEQS: Various coefficients of multivariate kurtosis and the numbers of observations that contribute most to the normalized multivariate kurtosis if the DATA= data set is a raw data set and the KURTOSIS option, or you specify at least the PRINT option. See the section "Measures of Multivariate Kurtosis" on page 938 for more information.
- COSAN and LINEQS: Covariance or correlation matrix to be analyzed and the value of its determinant if you specify the output option PCORR or PALL. A 0 determinant indicates a singular data matrix. In this case, the generalized least squares estimates with default weight matrix **S** and maximum likelihood estimates cannot be computed.
- LINEQS: If exogenous manifest variables in the linear structural equation model are specified, then there is a one-to-one relationship between the given covariances and corresponding estimates in the central model matrix Φ or P. The output indicates which manifest variables are recognized as exogenous—that is, for which variables the entries in the central model matrix are set to fixed parameters. The output, or at least the default output, is displayed if you specify the PINITIAL option.
- COSAN and LINEQS: Vector of parameter names, initial values, and corresponding matrix locations, also indicating dependent parameter names used in your programming statements that are not allocated to matrix locations and have no influence on the fit function. The output, or at least the default output, is displayed if you specify the PINITIAL option.
- COSAN and LINEQS: The pattern of variable and constant elements of the predicted moment matrix that is predetermined by the analysis model is displayed if there are significant differences between constant elements in the predicted model matrix and the data matrix and you specify at least the PSHORT option. It is also displayed if you specify the PREDET option. The output indicates the differences between constant values in the predicted model matrix and the data matrix that is analyzed.
- COSAN and LINEQS: Special features of the optimization technique chosen if you specify at least the PSHORT option.

- COSAN and LINEQS: Optimization history if at least the PSHORT option is specified. For more details, see the section "Use of Optimization Techniques" on page 945.
- COSAN and LINEQS: Specific output requested by options in the NLOPTIONS statement; for example, parameter estimates, gradient, gradient of Lagrange function, constraints, Lagrange multipliers, projected gradient, Hessian, projected Hessian, Hessian of Lagrange function, Jacobian of nonlinear constraints.
- COSAN and LINEQS: The predicted model matrix and its determinant, if you specify the output option PCORR or PALL.
- COSAN and LINEQS: Residual and normalized residual matrix if you specify the RESIDUAL, or at least the PRINT option. The variance standardized or asymptotically standardized residual matrix can be displayed also. The average residual and the average off-diagonal residual are also displayed. See the section "Assessment of Fit" on page 928 for more details.
- COSAN and LINEQS: Rank order of the largest normalized residuals if you specify the RESIDUAL, or at least the PRINT option.
- COSAN and LINEQS: Bar chart of the normalized residuals if you specify the RESIDUAL, or at least the PRINT option.
- COSAN and LINEQS: Value of the fit function F. See the section "Estimation Criteria" on page 923 for more details. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Goodness of fit index (GFI), adjusted goodness of fit index (AGFI), and root mean square residual (RMR) (Jöreskog and Sörbom 1985). See the section "Assessment of Fit" on page 928 for more details. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Parsimonious goodness of fit index (PGFI) of Mulaik et al. (1989). See the section "Assessment of Fit" on page 928 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Overall χ^2, df, and Prob>Chi**2 if the METHOD= option is not ULS or DWLS. The χ^2 measure is the optimum function value F multiplied by $(N-1)$ if a CORR or COV matrix is analyzed or multiplied by N if a UCORR or UCOV matrix is analyzed; χ^2 measures the likelihood ratio test statistic for the null hypothesis that the predicted matrix $\mathbf{C}$ has the specified model structure against the alternative that $\mathbf{C}$ is unconstrained. The notation Prob>Chi**2 means "the probability under the null hypothesis of obtaining a greater χ^2 statistic than that observed." This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If METHOD= is not ULS or DWLS, the χ_0^2 value of the independence model and the corresponding degrees of freedom can be used (in large samples) to evaluate the gain of explanation by fitting the specific model (Bentler 1989). See the section "Assessment of Fit" on page 928 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If METHOD= is not ULS or DWLS, the value of the Steiger and Lind (1980) root mean squared error of approximation (RMSEA) coefficient and the lower and upper limits of the confidence interval. The size of the confidence interval is defined by the

option ALPHARMS=α, $0 \leq \alpha \leq 1$. The default is $\alpha = 0.1$, which corresponds to a 90% confidence interval. See the section "Assessment of Fit" on page 928 for more detail. This output can be suppressed only by the NOPRINT option.

- COSAN and LINEQS: If the value of the METHOD= option is not ULS or DWLS, the value of the *probability of close fit* (Browne and Cudeck 1993). See the section "Assessment of Fit" on page 928 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If the value of the METHOD= option is not ULS or DWLS, the value of the Browne and Cudeck (1993) expected cross validation (ECVI) index and the lower and upper limits of the confidence interval. The size of the confidence interval is defined by the option ALPHAECV=α, $0 \leq \alpha \leq 1$. The default is $\alpha = 0.1$, which corresponds to a 90% confidence interval. See the section "Assessment of Fit" on page 928 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If the value of the METHOD= option is not ULS or DWLS, Bentler's (1989) comparative fit index. See the section "Assessment of Fit" on page 928 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If you specify METHOD=ML or METHOD=GLS, the χ^2 value and corresponding probability adjusted by the relative kurtosis coefficient η_2, which should be a close approximation of the χ^2 value for elliptically distributed data (Browne 1982). See the section "Assessment of Fit" on page 928 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: The normal theory reweighted LS χ^2 value is displayed if METHOD= ML. Instead of the function value F_{ML}, the reweighted goodness of fit function F_{GWLS} is used. See the section "Assessment of Fit" on page 928 for more detail.
- COSAN and LINEQS: Akaike's information criterion if the value of the METHOD= option is not ULS or DWLS. See the section "Assessment of Fit" on page 928 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Bozdogan's (1987) consistent information criterion, CAIC. See the section "Assessment of Fit" on page 928 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Schwarz's Bayesian criterion (SBC) if the value of the METHOD= option is not ULS or DWLS (Schwarz 1978). See the section "Assessment of Fit" on page 928 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: If the value of the METHOD= option is not ULS or DWLS, the following fit indices based on the overall χ^2 value are displayed:
  - McDonald's (McDonald 1989) measure of centrality
  - Parsimonious index of James, Mulaik, and Brett (1982)
  - Z-test of Wilson and Hilferty (1931)
  - Bentler and Bonett's (1980) nonnormed coefficient
  - Bentler and Bonett's (1980) normed coefficient
  - Bollen's (1986) normed index ρ_1
  - Bollen's (1989a) nonnormed index Δ_2

See the section "Assessment of Fit" on page 928 for more detail. This output can be suppressed only by the NOPRINT option.

- COSAN and LINEQS: Hoelter's (1983) critical N index is displayed (Bollen 1989b, p. 277). See the section "Assessment of Fit" on page 928 for more detail. This output can be suppressed only by the NOPRINT option.
- COSAN and LINEQS: Equations of linear dependencies among the parameters used in the model specification if the information matrix is recognized as singular at the final solution.
- COSAN: Model matrices containing the parameter estimates. Except for ULS or DWLS estimates, the approximate standard errors and t values are also displayed. This output is displayed if you specify the PESTIM option or at least the PSHORT option.
- LINEQS: Linear equations containing the parameter estimates. Except for ULS and DWLS estimates, the approximate standard errors and t values are also displayed. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.
- LINEQS: Variances and covariances of the exogenous variables. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.
- LINEQS: Linear equations containing the standardized parameter estimates. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.
- LINEQS: Table of correlations among the exogenous variables. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.
- LINEQS: Correlations among the exogenous variables. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.
- LINEQS: Squared multiple correlations table, which displays the error variances of the endogenous variables. These are the diagonal elements of the predicted model matrix. Also displayed is the Total Variance and the R^2 values corresponding to all endogenous variables. See the section "Assessment of Fit" on page 928 for more detail. This output is displayed if you specify the PESTIM option, or at least the PSHORT option.
- LINEQS: If you specify the PDETERM or the PALL option, the total determination of all equations (DETAE), the total determination of the structural equations (DETSE), and the total determination of the manifest variables (DETMV) are displayed. See the section "Assessment of Fit" on page 928 for more detail. If one of the determinants in the formulas is 0, the corresponding coefficient is displayed as a missing value. If there are structural equations, PROC CALIS also displays the stability coefficient of reciprocal causation—that is, the largest eigenvalue of the $\mathbf{BB}'$ matrix, where $\mathbf{B}$ is the causal coefficient matrix of the structural equations.
- LINEQS: The matrix of estimated covariances among the latent variables if you specify the PLATCOV option, or at least the PRINT option.
- LINEQS: The matrix of estimated covariances between latent and manifest variables used in the model if you specify the PLATCOV option, or at least the PRINT option.

- LINEQS and FACTOR: The matrix **FSR** of latent variable scores regression coefficients if you specify the PLATCOV option, or at least the PRINT option. The **FSR** matrix is a generalization of Lawley and Maxwell's (1971, p. 109) factor scores regression matrix,

  $$\mathbf{FSR} = \mathbf{C}_{yx}\mathbf{C}_{xx}^{-1}$$

  where $\mathbf{C}_{xx}$ is the $n \times n$ predicted model matrix. (predicted covariances among manifest variables) and $\mathbf{C}_{yx}$ is the $n_{lat} \times n$ matrix of the predicted covariances between latent and manifest variables. You can multiply the manifest observations by this matrix to estimate the scores of the latent variables used in your model.

- LINEQS: The matrix **TEF** of total effects if you specify the TOTEFF option, or at least the PRINT option. For the LINEQS model, the matrix of total effects is

  $$\mathbf{TEF} = (\mathbf{I} - \boldsymbol{\beta})^{-1}\boldsymbol{\gamma} - (\mathbf{O} : \mathbf{I})$$

  (For the LISREL model, refer to Jöreskog and Sörbom 1985.) The matrix of indirect effects is displayed also.

- FACTOR: The matrix of rotated factor loadings and the orthogonal transformation matrix if you specify the ROTATE= and PESTIM options, or at least the PSHORT option.

- FACTOR: Standardized (rotated) factor loadings, variance estimates of endogenous variables, R^2 values, correlations among factors, and factor scores regression matrix, if you specify the PESTIM option, or at least the PSHORT option. The determination of manifest variables is displayed only if you specify the PDETERM option.

- COSAN and LINEQS: Univariate Lagrange multiplier and Wald test indices are displayed in matrix form if you specify the MODIFICATION (or MOD) or PALL option. Those matrix locations that correspond to constants in the model in general contain three values: the value of the Lagrange multiplier, the corresponding probability ($df = 1$), and the estimated change of the parameter value should the constant be changed to a parameter. If allowing the constant to be an estimated parameter would result in a singular information matrix, the string 'sing' is displayed instead of the Lagrange multiplier index. Those matrix locations that correspond to parameter estimates in the model contain the Wald test index and the name of the parameter in the model. See the section "Modification Indices" on page 953 for more detail.

- COSAN and LINEQS: Univariate Lagrange multiplier test indices for releasing equality constraints if you specify the MODIFICATION (or MOD) or PALL option. See the section "Modification Indices" on page 953 for more detail.

- COSAN and LINEQS: Univariate Lagrange multiplier test indices for releasing active boundary constraints specified by the BOUNDS statement if you specify the MODIFICATION (or MOD) or PALL option. See the section "Modification Indices" on page 953 for more detail.

- COSAN and LINEQS: If the MODIFICATION (or MOD) or PALL option is specified, the stepwise multivariate Wald test for constraining estimated parameters to zero constants is performed as long as the univariate probability is larger than the value specified in the PMW= option (default PMW=0.05). See the section "Modification Indices" on page 953 for more detail.

ODS Table Names

PROC CALIS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 25.15 ODS Tables Produced by PROC CALIS

ODS Table Name	Model[1]	Description	Option[2]
AddParms	C, F, L, R	Additional parameters in the PARAMETERS statement	PINITIAL or default
AsymStdRes	C, F, L, R	Asymptotically standardized residual matrix	RESIDUAL= or PRINT
AveAsymStdRes	C, F, L, R	Average absolute asymptotically standardized residuals	RESIDUAL= or PRINT
AveNormRes	C, F, L, R	Average absolute normalized residuals	RESIDUAL= or PRINT
AveRawRes	C, F, L, R	Average absolute raw residuals	RESIDUAL= or PRINT
AveVarStdRes	C, F, L, R	Average absolute variance standardized residuals	RESIDUAL= or PRINT
ContKurtosis	C, F, L, R	Contributions to kurtosis	KURTOSIS or PRINT
ConvergenceStatus	C, F, L, R	Convergence status	PSHORT
CorrExog	L	Correlations among exogenous variables	PESTIM or PSHORT
CorrParm	C, F, L, R	Correlations among parameter estimates	PCOVES and default
CovMat	C, F, L, R	Assorted covariance matrices	PCOVES and default
DependParms	C, F, L, R	Dependent parameters (if specified by programming statements)	PRIVEC and default
Determination	L, F, R	Coefficients of determination	PDETERM and default
DistAsymStdRes	C, F, L, R	Distribution of asymptotically standardized residuals	RESIDUAL= or PRINT
DistNormRes	C, F, L, R	Distribution of normalized residuals	RESIDUAL= or PRINT
DistVarStdRes	C, F, L, R	Distribution of variance standardized residuals	RESIDUAL= or PRINT
Eigenvalues	C, F, L, R	Eigenvalues of the approximate Hessian matrix	PCOVES[3] and default
EndogenousVar	L	Endogenous variables	PESTIM or PSHORT
EstCovExog	L	Estimated covariances among exogenous variables	PESTIM or PSHORT
Estimates	C, F, L, R	Vector of estimates	PRIVEC
EstLatentEq	L	Estimated latent variable equations	PESTIM or PSHORT
EstManifestEq	L	Estimated manifest variable equations	PESTIM or PSHORT
EstParms	C, F	Estimated parameter matrix	PESTIM or PSHORT
EstVarExog	L	Estimated variances of exogenous variables	PESTIM or PSHORT
ExogenousVar	L	List of exogenous variables	PESTIM or PSHORT
FactCorrExog	F	Correlations among factors	PESTIM or PSHORT

Table 25.15 *continued*

ODS Table Name	Model[1]	Description	Option[2]
FactScoreCoef	F	Factor score regression coefficients	PESTIM or PSHORT
Fit	C, F, L, R	Fit statistics	PSUMMARY
GenModInfo	C, F, L, R	General modeling information	PSIMPLE or default
Gradient	C, F, L, R	First partial derivatives (Gradient)	PRIVEC and default
InCorr	C, F, L, R	Input correlation matrix	PCORR or PALL
InCorrDet	C, F, L, R	Determinant of the input correlation matrix	PCORR or PALL
InCov	C, F, L, R	Input covariance matrix	PCORR or PALL
InCovDet	C, F, L, R	Determinant of the input covariance matrix	PCORR or PALL
InCovExog	L	Input covariances among exogenous variables	PESTIM or PSHORT
IndirectEffects	L, R	Indirect effects	TOTEFF or PRINT
Information	C, F, L, R	Information matrix	PCOVES and default
InitEstimates	C, F, L, R	Initial vector of parameter estimates	PINITIAL or default
InitParms	C, F	Initial matrix of parameter estimates	PINITIAL or default
InitParms	L, R	Initial matrix of parameter estimates	PRIMAT and default
InitRAMEstimates	R	Initial RAM estimates	PESTIM or PSHORT
InLatentEq	L	Input latent variable equations	PESTIM or PSHORT
InManifestEq	L	Input manifest variable equations	PESTIM or PSHORT
InSymmetric	C, F, L, R	Input symmetric matrix (SYMATRIX data type)	PCORR or PALL
InVarExog	L	Input variances of exogenous variables	PESTIM or PSHORT
IterHist	C, F, L, R	Iteration history	PSHORT
IterStart	C, F, L, R	Iteration start	PSHORT
IterStop	C, F, L, R	Iteration stop	PSHORT
Jacobian	C, F, L, R	Jacobi column pattern	PJACPAT
Kurtosis	C, F, L, R	Kurtosis, with raw data input	KURTOSIS or PRINT
LagrangeBoundary	C, F, L, R	Lagrange, releasing active boundary constraints	MODIFICATION[4] or PALL
LagrangeEquality	C, F, L, R	Lagrange, releasing equality constraints	MODIFICATION or PALL
LatentScoreCoef	L, R	Latent variable regression score coefficients	PLATCOV or PRINT
ModelStatement	C, F, L, R	Model summary	PSHORT
ModIndices	C, F, L, R	Lagrange multiplier and Wald test statistics	MODIFICATION or PALL
NormRes	C, F, L, R	Normalized residual matrix	RESIDUAL= or PRINT
PredetElements	C, F, L, R	Predetermined elements	PREDET or PALL
PredModel	C, F, L, R	Predicted model matrix	PCORR or PALL
PredModelDet	C, F, L, R	Predicted model determinant	PCORR or PALL
PredMomentLatent	L, R	Predicted latent variable moments	PLATCOV or PRINT
PredMomentManLat	L, R	Predicted manifest and latent variable moments	PLATCOV or PRINT
ProblemDescription	C, F, L, R	Problem description	PSHORT
RAMCorrExog	R	Correlations among exogenous variables	PESTIM or PSHORT

Table 25.15 *continued*

ODS Table Name	Model[1]	Description	Option[2]
RAMEstimates	R	RAM final estimates	PESTIM, or PSHORT
RAMStdEstimates	R	Standardized estimates	PESTIM, or PSHORT
RankAsymStdRes	C, F, L, R	Ranking of the largest asymptotically standardized residuals	RESIDUAL= or PRINT
RankLagrange	C, F, L, R	Ranking of the largest Lagrange indices	RESIDUAL= or PRINT
RankNormRes	C, F, L, R	Ranking of the largest normalized residuals	RESIDUAL= or PRINT
RankRawRes	C, F, L, R	Ranking of the largest raw residuals	RESIDUAL= or PRINT
RankVarStdRes	C, F, L, R	Ranking of the largest variance standardized residuals	RESIDUAL= or PRINT
RawRes	C, F, L, R	Raw residual matrix	RESIDUAL= or PRINT
RotatedLoadings	F	Rotated loadings, with ROTATE= option in FACTOR statement	PESTIM or PSHORT
Rotation	F	Rotation matrix, with ROTATE= option in FACTOR statement	PESTIM or PSHORT
SetCovExog	L, R	Set covariance parameters for manifest exogenous variables	PINITIAL or default
SimpleStatistics	C, F, L, R	Simple statistics, with raw data input	SIMPLE or default
SqMultCorr	F, L, R	Squared multiple correlations	PESTIM or PSHORT
Stability	L, R	Stability of reciprocal causation	PDETERM and default
StdErrs	C, F, L, R	Vector of standard errors	PRIVEC and default
StdLatentEq	L	Standardized latent variable equations	PESTIM or PSHORT
StdLoadings	F	Standardized factor loadings	PESTIM or PSHORT
StdManifestEq	L	Standardized manifest variable equations	PESTIM or PSHORT
StructEq	L, R	Variables in the structural equations	PDETERM and default
SumSqDif	C, F, L, R	Sum of squared differences of pre-determined elements	PREDET or PALL
TotalEffects	L, R	Total effects	TOTEFF or PRINT
tValues	C, F, L, R	Vector of t values	PRIVEC and default
VarSelection	L, R	Manifest variables, if not all are used, selected for modeling	default
VarStdRes	C, F, L, R	Variance standardized residual matrix	RESIDUAL= or PRINT
WaldTest	C, F, L, R	Wald test	MODIFICATION or PALL
Weights	C, F, L, R	Weight matrix	PWEIGHT[5] or PALL
WeightsDet	C, F, L, R	Determinant of the weight matrix	PWEIGHT[5] or PALL

1. Most CALIS output tables are specific to the model statement used. Keys: C: COSAN model, F: FACTOR model, L: LINEQS model, R: RAM model.

2. The printing options PALL, PRINT, "default," PSHORT, and PSUMM form hierarchical levels of output control, with PALL including all the output enabled by the options at the lower levels, and so on. The "default" option means that NOPRINT is not specified. Therefore, in the table, for example, if PSHORT is the printing option for an output, PALL, PRINT, or "default" will also enable the same output printing.

3. The eigenvalues are printed only when there are negative eigenvalues in the approximate Hessian matrix.

4. The printing of LagrangeBoundary is effective only if you have set some boundary constraints for parameters.

5. The printing of Weights or WeightsDet is effective only if your estimation method uses the weight matrix (for example, WLS or LSWLS).

ODS Graphics

To request graphics with PROC CALIS, you must first enable ODS Graphics by specifying the **ods graphics on** statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. The names of the graphs that PROC CALIS generates are listed in Table 25.16, along with the required statements and options.

Table 25.16 ODS Graphics Produced by PROC CALIS

ODS Graph Name	Plot Description	PLOTS= Option
AsymStdResidualHistogram	Asymptotically standardized residuals	PLOTS=RESIDUALS and RESIDUAL=ASYMSTD; METHOD= is not ULS or DWLS
NormResidualHistogram	Normalized residuals	PLOTS=RESIDUALS and RESIDUAL=NORM
RawResidualHistogram	Raw residuals	PLOTS=RESIDUALS
VarStdResidualHistogram	Variance standardized residuals	PLOTS=RESIDUALS and RESIDUAL=VARSTD

Examples: CALIS Procedure

Example 25.1: Path Analysis: Stability of Alienation

The following covariance matrix from Wheaton et al. (1977) has served to illustrate the performance of several implementations for the analysis of structural equation models. Two different models have been analyzed by an early implementation of LISREL and are mentioned in Jöreskog (1978). You can also find a more detailed discussion of these models in the LISREL VI manual (Jöreskog and Sörbom 1985).

A slightly modified model for this covariance matrix is included in the EQS 2.0 manual (Bentler 1985, p. 28). The path diagram of this model is displayed in Figure 25.1. The same model is reanalyzed here by PROC CALIS. However, for the analysis with the EQS implementation, the last variable (V6) is rescaled by a factor of 0.1 to make the matrix less ill-conditioned. Since the Levenberg-Marquardt or Newton-Raphson optimization technique is used with PROC CALIS, rescaling the data matrix is not necessary and, therefore, is not done here. The results reported here reflect the estimates based on the original covariance matrix.

The DATA step and the CALIS model specification are shown as follows:

```
data Wheaton(TYPE=COV);
title "Stability of Alienation";
title2 "Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)";
   _type_ = 'cov'; input _name_ $ v1-v6;
   label v1='Anomie (1967)' v2='Anomie (1971)' v3='Education'
         v4='Powerlessness (1967)' v5='Powerlessness (1971)'
         v6='Occupational Status Index';
   datalines;
v1   11.834     .        .        .       .        .
v2    6.947    9.364     .        .       .        .
v3    6.819    5.091   12.532     .       .        .
v4    4.783    5.028    7.495    9.986    .        .
v5   -3.839   -3.889   -3.841   -3.625   9.610     .
v6  -21.899  -18.831  -21.748  -18.775  35.522  450.288
;


ods graphics on;

proc calis cov data=Wheaton tech=nr edf=931 pall plots=residuals;
   lineqs
      v1 =         f1                  + e1,
      v2 =    .833 f1                  + e2,
      v3 =         f2                  + e3,
      v4 =    .833 f2                  + e4,
      v5 =         f3                  + e5,
      v6 = Lamb (.5) f3                + e6,
      f1 = Gam1(-.5) f3                + d1,
      f2 = Beta (.5) f1 + Gam2(-.5) f3 + d2;
   std
      e1-e6 = The1-The2 The1-The4 (6 * 3.),
      d1-d2 = Psi1-Psi2 (2 * 4.),
      f3    = Phi (6.) ;
   cov
      e1 e3 = The5 (.2),
      e4 e2 = The5 (.2);
run;

ods graphics off;
```

The COV option in the PROC CALIS statement requests the analysis of the covariance matrix. Without the COV option, the correlation matrix would be computed and analyzed. Since no METHOD= option has been used, maximum likelihood estimates are computed by default. The TECH=NR option requests the Newton-Raphson optimization method. The PALL option produces the almost complete set of displayed output, as displayed in Output 25.1.1 through Output 25.1.23. Note that, when you specify the PALL option, you can produce large amounts of output. The PALL option is used in this example to show how you can get a wide spectrum of useful information from PROC CALIS.

PROC CALIS can produce a high-quality residual histogram that is useful for showing the distribution of residuals. To request the residual histogram, you must first enable ODS Graphics by specifying the **ods graphics on** statement, as shown in the preceding code before the PROC CALIS statement. Then, the residual histogram is requested by the **plots=residuals** option in the PROC CALIS statement.

Output 25.1.1 displays the model specification in matrix terms, followed by the lists of endogenous and exogenous variables. Output 25.1.2 displays equations and initial parameter estimates. You can use these output to ensure that the desired model is being analyzed.

Output 25.1.1 Model and Variables

```
                          Stability of Alienation
              Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)

                          The CALIS Procedure
           Covariance Structure Analysis: Pattern and Initial Values

                          LINEQS Model Statement


                       Matrix      Rows    Columns    ------Matrix Type-------

Term 1             1   _SEL_          6         17    SELECTION
                   2   _BETA_        17         17    EQSBETA         IMINUSINV
                   3   _GAMMA_       17          9    EQSGAMMA
                   4   _PHI_          9          9    SYMMETRIC

                             The 8 Endogenous Variables

    Manifest         v1  v2  v3  v4  v5  v6
    Latent           f1  f2

                             The 9 Exogenous Variables

    Manifest
    Latent           f3
    Error            e1  e2  e3  e4  e5  e6  d1  d2
```

Output 25.1.2 Initial Model Specifications

```
                Manifest Variable Equations with Initial Estimates


                     v1      =   1.0000 f1   +  1.0000 e1
                     v2      =   0.8330 f1   +  1.0000 e2
                     v3      =   1.0000 f2   +  1.0000 e3
                     v4      =   0.8330 f2   +  1.0000 e4
                     v5      =   1.0000 f3   +  1.0000 e5
                     v6      =   0.5000*f3   +  1.0000 e6
                                      Lamb
```

Output 25.1.2 *continued*

```
          Latent Variable Equations with Initial Estimates


     f1      =  -0.5000*f3   +  1.0000 d1
                        Gam1
     f2      =   0.5000*f1   + -0.5000*f3   +  1.0000 d2
                        Beta           Gam2

                Variances of Exogenous Variables

                Variable Parameter      Estimate

                f3       Phi             6.00000
                e1       The1            3.00000
                e2       The2            3.00000
                e3       The1            3.00000
                e4       The2            3.00000
                e5       The3            3.00000
                e6       The4            3.00000
                d1       Psi1            4.00000
                d2       Psi2            4.00000

              Covariances Among Exogenous Variables

                Var1 Var2 Parameter      Estimate

                e1   e3   The5            0.20000
                e2   e4   The5            0.20000
```

General modeling information, descriptive statistics, and the input covariance matrix are displayed in Output 25.1.3. Because the input data set contains only the covariance matrix, the means of the manifest variables are assumed to be zero. Note that this has no impact on the estimation, unless a mean structure model is being analyzed.

Output 25.1.3 Modeling Information, Simple Statistics, and Input Covariance Matrix

```
          Observations          932    Model Terms              1
          Variables               6    Model Matrices           4
          Informations           21    Parameters              12

                  Variable                        Mean       Std Dev

     v1    Anomie (1967)                             0       3.44006
     v2    Anomie (1971)                             0       3.06007
     v3    Education                                 0       3.54006
     v4    Powerlessness (1967)                      0       3.16006
     v5    Powerlessness (1971)                      0       3.10000
     v6    Occupational Status Index                 0      21.21999
```

Output 25.1.3 *continued*

Covariances

		v1	v2	v3
v1	Anomie (1967)	11.83400000	6.94700000	6.81900000
v2	Anomie (1971)	6.94700000	9.36400000	5.09100000
v3	Education	6.81900000	5.09100000	12.53200000
v4	Powerlessness (1967)	4.78300000	5.02800000	7.49500000
v5	Powerlessness (1971)	-3.83900000	-3.88900000	-3.84100000
v6	Occupational Status Index	-21.89900000	-18.83100000	-21.74800000

Covariances

		v4	v5	v6
v1	Anomie (1967)	4.78300000	-3.83900000	-21.8990000
v2	Anomie (1971)	5.02800000	-3.88900000	-18.8310000
v3	Education	7.49500000	-3.84100000	-21.7480000
v4	Powerlessness (1967)	9.98600000	-3.62500000	-18.7750000
v5	Powerlessness (1971)	-3.62500000	9.61000000	35.5220000
v6	Occupational Status Index	-18.77500000	35.52200000	450.2880000

Determinant 6080570 Ln 15.620609

The 12 parameter estimates in the model and their respective locations in the parameter matrices are displayed in Output 25.1.4. Each of the parameters, The1, The2, and The5, is specified for two elements in the parameter matrix _PHI_.

Output 25.1.4 Initial Estimates

Vector of Initial Estimates

	Parameter	Estimate	Type		
1	Beta	0.50000	Matrix Entry: _BETA_[8:7]		
2	Lamb	0.50000	Matrix Entry: _GAMMA_[6:1]		
3	Gam1	-0.50000	Matrix Entry: _GAMMA_[7:1]		
4	Gam2	-0.50000	Matrix Entry: _GAMMA_[8:1]		
5	Phi	6.00000	Matrix Entry: _PHI_[1:1]		
6	The1	3.00000	Matrix Entry: _PHI_[2:2]	_PHI_[4:4]	
7	The2	3.00000	Matrix Entry: _PHI_[3:3]	_PHI_[5:5]	
8	The5	0.20000	Matrix Entry: _PHI_[4:2]	_PHI_[5:3]	
9	The3	3.00000	Matrix Entry: _PHI_[6:6]		
10	The4	3.00000	Matrix Entry: _PHI_[7:7]		
11	Psi1	4.00000	Matrix Entry: _PHI_[8:8]		
12	Psi2	4.00000	Matrix Entry: _PHI_[9:9]		

PROC CALIS examines whether each element in the moment matrix is modeled by the parameters defined in the model. If an element is not structured by the model parameters, it is predetermined by its observed value. This occurs, for example, when there are exogenous manifest variables in the

model. If present, the predetermined values of the elements are displayed. In the current example, the '.' displayed for all elements in the predicted moment matrix (Output 25.1.5) indicates that there are no predetermined elements in the model.

Output 25.1.5 Predetermined Elements

Predetermined Elements of the Predicted Moment Matrix

		v1	v2	v3
v1	Anomie (1967)	.	.	.
v2	Anomie (1971)	.	.	.
v3	Education	.	.	.
v4	Powerlessness (1967)	.	.	.
v5	Powerlessness (1971)	.	.	.
v6	Occupational Status Index	.	.	.

Predetermined Elements of the Predicted Moment Matrix

		v4	v5	v6
v1	Anomie (1967)	.	.	.
v2	Anomie (1971)	.	.	.
v3	Education	.	.	.
v4	Powerlessness (1967)	.	.	.
v5	Powerlessness (1971)	.	.	.
v6	Occupational Status Index	.	.	.

Sum of Squared Differences	0

Output 25.1.6 displays the optimization information. You can check this table to determine whether the convergence criterion is satisfied. PROC CALIS displays an error message when problematic solutions are encountered.

Output 25.1.6 Optimization

Newton-Raphson Ridge Optimization

Without Parameter Scaling

Parameter Estimates	12
Functions (Observations)	21

Optimization Start

Active Constraints	0	Objective Function	119.33282242
Max Abs Gradient Element	74.016932345		

Output 25.1.6 *continued*

Iter	Rest arts	Func Calls	Act Con	Objective Function	Obj Fun Change	Max Abs Gradient Element	Ridge	Actual Over Pred Change
1	0	2	0	0.82689	118.5	1.3507	0	0.0154
2	0	3	0	0.09859	0.7283	0.2330	0	0.716
3	0	4	0	0.01581	0.0828	0.00684	0	1.285
4	0	5	0	0.01449	0.00132	0.000286	0	1.042
5	0	6	0	0.01448	9.936E-7	0.000045	0	1.053
6	0	7	0	0.01448	4.227E-9	1.685E-6	0	1.056

Optimization Results

Iterations	6	Function Calls	8
Jacobian Calls	7	Active Constraints	0
Objective Function	0.0144844811	Max Abs Gradient Element	1.6847829E-6
Ridge	0	Actual Over Pred Change	1.0563187228

ABSGCONV convergence criterion satisfied.

Next, the predicted model matrix is displayed in the Output 25.1.7, followed by a list of model test statistics or fit indices (Output 25.1.8). Depending on your modeling philosophy, some indices might be preferred to others. In this example, all indices and test statistics point to a good fit of the model.

Output 25.1.7 Predicted Model Matrix

Predicted Model Matrix

		v1	v2	v3
v1	Anomie (1967)	11.90390632	6.91059048	6.83016211
v2	Anomie (1971)	6.91059048	9.35145064	4.93499582
v3	Education	6.83016211	4.93499582	12.61574998
v4	Powerlessness (1967)	4.93499582	5.01664889	7.50355625
v5	Powerlessness (1971)	-4.16791157	-3.47187034	-4.06565606
v6	Occupational Status Index	-22.37688158	-18.63994236	-21.82788734

Predicted Model Matrix

		v4	v5	v6
v1	Anomie (1967)	4.93499582	-4.16791157	-22.3768816
v2	Anomie (1971)	5.01664889	-3.47187034	-18.6399424
v3	Education	7.50355625	-4.06565606	-21.8278873
v4	Powerlessness (1967)	9.84539112	-3.38669150	-18.1826302
v5	Powerlessness (1971)	-3.38669150	9.61000000	35.5219999
v6	Occupational Status Index	-18.18263015	35.52199986	450.2879993

Determinant 6169285 Ln 15.635094

Output 25.1.8 Fit Statistics

Fit Function	0.0145
Goodness of Fit Index (GFI)	0.9953
GFI Adjusted for Degrees of Freedom (AGFI)	0.9890
Root Mean Square Residual (RMR)	0.2281
Standardized Root Mean Square Residual (SRMR)	0.0150
Parsimonious GFI (Mulaik, 1989)	0.5972
Chi-Square	13.4851
Chi-Square DF	9
Pr > Chi-Square	0.1419
Independence Model Chi-Square	2131.4
Independence Model Chi-Square DF	15
RMSEA Estimate	0.0231
RMSEA 90% Lower Confidence Limit	.
RMSEA 90% Upper Confidence Limit	0.0470
ECVI Estimate	0.0405
ECVI 90% Lower Confidence Limit	.
ECVI 90% Upper Confidence Limit	0.0556
Probability of Close Fit	0.9705
Bentler's Comparative Fit Index	0.9979
Normal Theory Reweighted LS Chi-Square	13.2804
Akaike's Information Criterion	-4.5149
Bozdogan's (1987) CAIC	-57.0509
Schwarz's Bayesian Criterion	-48.0509
McDonald's (1989) Centrality	0.9976
Bentler & Bonett's (1980) Non-normed Index	0.9965
Bentler & Bonett's (1980) NFI	0.9937
James, Mulaik, & Brett (1982) Parsimonious NFI	0.5962
Z-Test of Wilson & Hilferty (1931)	1.0754
Bollen (1986) Normed Index Rho1	0.9895
Bollen (1988) Non-normed Index Delta2	0.9979
Hoelter's (1983) Critical N	1170

PROC CALIS can perform a detailed residual analysis. Large residuals might indicate misspecification of the model. In Output 25.1.9, raw residuals are reported and ranked. Because of the differential scaling of the variables, it is usually more useful to examine the standardized residuals instead. In Output 25.1.10, for example, the table for the 10 largest asymptotically standardized residuals is displayed. The model performs the poorest concerning the variable v5 and its covariance with v2, v1, and v3. This might suggest a misspecification of the model equation for v5. However, because the model fit is quite good, such a possible misspecification is not a serious concern in the analysis.

Output 25.1.9 Raw Residuals and Ranking

Raw Residual Matrix

		v1	v2	v3
v1	Anomie (1967)	-.0699063150	0.0364095216	-.0111621061
v2	Anomie (1971)	0.0364095216	0.0125493646	0.1560041795
v3	Education	-.0111621061	0.1560041795	-.0837499788
v4	Powerlessness (1967)	-.1519958205	0.0113511059	-.0085562504
v5	Powerlessness (1971)	0.3289115712	-.4171296612	0.2246560598
v6	Occupational Status Index	0.4778815840	-.1910576405	0.0798873380

Raw Residual Matrix

		v4	v5	v6
v1	Anomie (1967)	-.1519958205	0.3289115712	0.4778815840
v2	Anomie (1971)	0.0113511059	-.4171296612	-.1910576405
v3	Education	-.0085562504	0.2246560598	0.0798873380
v4	Powerlessness (1967)	0.1406088766	-.2383085022	-.5923698474
v5	Powerlessness (1971)	-.2383085022	0.0000000000	0.0000000000
v6	Occupational Status Index	-.5923698474	0.0000000000	0.0000000000

Average Absolute Residual	0.153928
Average Off-diagonal Absolute Residual	0.195045

Rank Order of the 10 Largest Raw Residuals

Row	Column	Residual
v6	v4	-0.59237
v6	v1	0.47788
v5	v2	-0.41713
v5	v1	0.32891
v5	v4	-0.23831
v5	v3	0.22466
v6	v2	-0.19106
v3	v2	0.15600
v4	v1	-0.15200
v4	v4	0.14061

Output 25.1.10 Asymptotically Standardized Residuals and Ranking

Asymptotically Standardized Residual Matrix

		v1	v2	v3
v1	Anomie (1967)	-0.308548787	0.526654452	-0.056188826
v2	Anomie (1971)	0.526654452	0.054363484	0.876120855
v3	Education	-0.056188826	0.876120855	-0.354347092
v4	Powerlessness (1967)	-0.865070455	0.057354415	-0.121874301
v5	Powerlessness (1971)	2.553366366	-2.763708659	1.697931678
v6	Occupational Status Index	0.464866661	-0.170127806	0.070202664

Asymptotically Standardized Residual Matrix

		v4	v5	v6
v1	Anomie (1967)	-0.865070455	2.553366366	0.464866661
v2	Anomie (1971)	0.057354415	-2.763708659	-0.170127806
v3	Education	-0.121874301	1.697931678	0.070202664
v4	Powerlessness (1967)	0.584930625	-1.557412695	-0.495982427
v5	Powerlessness (1971)	-1.557412695	0.000000000	0.000000000
v6	Occupational Status Index	-0.495982427	0.000000000	0.000000000

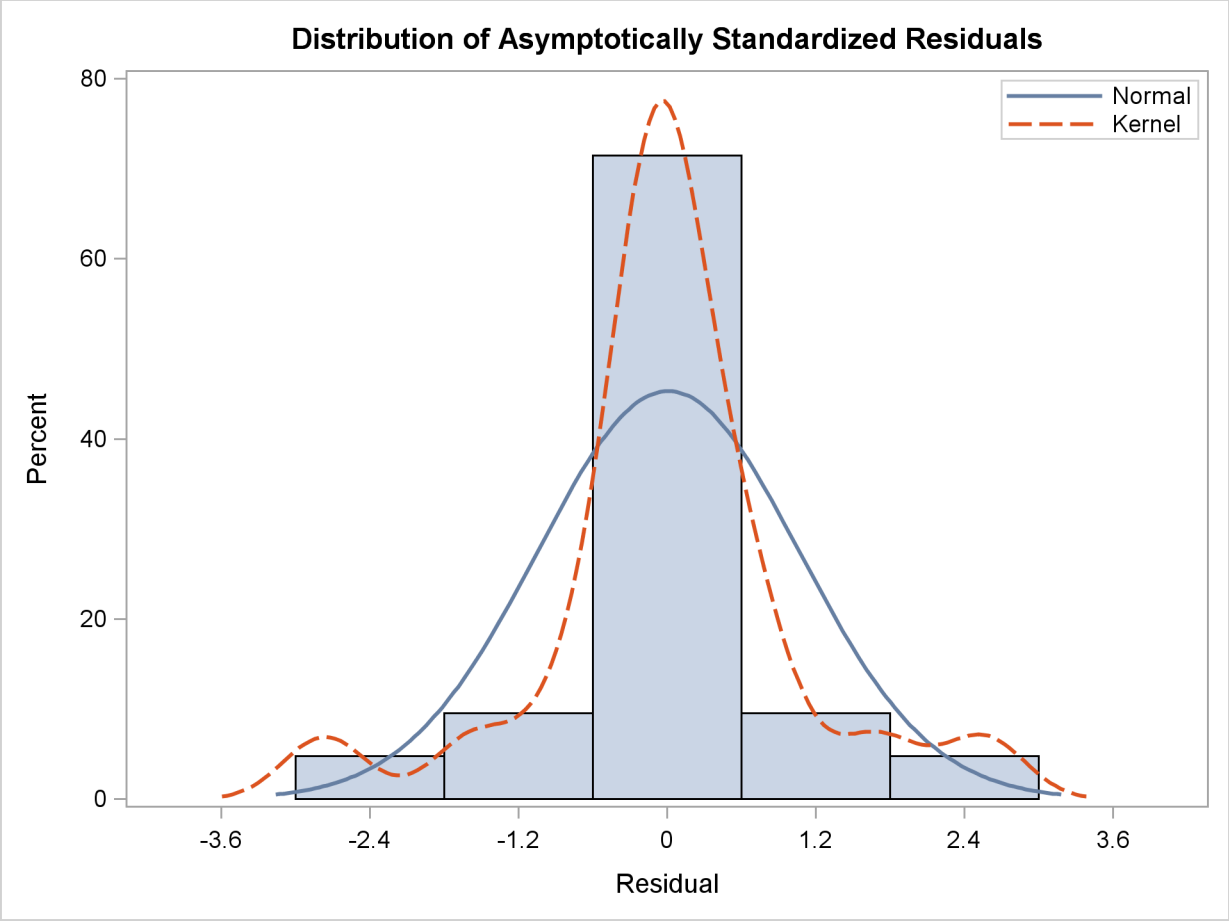
Average Standardized Residual	0.646622
Average Off-diagonal Standardized Residual	0.818457

Rank Order of the 10 Largest Asymptotically Standardized Residuals

Row	Column	Residual
v5	v2	-2.76371
v5	v1	2.55337
v5	v3	1.69793
v5	v4	-1.55741
v3	v2	0.87612
v4	v1	-0.86507
v4	v4	0.58493
v2	v1	0.52665
v6	v4	-0.49598
v6	v1	0.46487

The histogram of the asymptotically standardized residuals is displayed in Output 25.1.11, which also shows the normal and kernel approximations. The residual distribution looks quite symmetrical. It shows a small to medium departure from the normal distribution, as evidenced by the discrepancies between the kernel and the normal distribution curves.

Output 25.1.11 Distribution of Asymptotically Standardized Residuals



Output 25.1.12 displays the equations and parameter estimates. Each parameter estimate is displayed with its standard error and the corresponding *t* ratio. As a general rule, a *t* ratio larger than 2 represents a statistically significant departure from 0. From these results, it is observed that both f1 (Alienation 1967) and f2 (Alienation 1971) are regressed negatively on f3 (Socioeconomic Status), and f1 has a positive effect on f2. The estimates and significance tests for the variance and covariance of the exogenous variables are also displayed.

Output 25.1.12 Equations and Parameter Estimates

```
                 Manifest Variable Equations with Estimates


              v1      =   1.0000 f1       +  1.0000 e1
              v2      =   0.8330 f1       +  1.0000 e2
              v3      =   1.0000 f2       +  1.0000 e3
              v4      =   0.8330 f2       +  1.0000 e4
              v5      =   1.0000 f3       +  1.0000 e5
              v6      =   5.3688*f3       +  1.0000 e6
              Std Err     0.4337 Lamb
              t Value    12.3788
```

Output 25.1.12 *continued*

Latent Variable Equations with Estimates

```
f1      =  -0.6299*f3       +  1.0000 d1
Std Err     0.0563 Gam1
t Value   -11.1809
f2      =   0.5931*f1       + -0.2409*f3       +  1.0000 d2
Std Err     0.0468 Beta        0.0549 Gam2
t Value    12.6788            -4.3885
```

Variances of Exogenous Variables

Variable	Parameter	Estimate	Standard Error	t Value
f3	Phi	6.61632	0.63914	10.35
e1	The1	3.60788	0.20092	17.96
e2	The2	3.59493	0.16448	21.86
e3	The1	3.60788	0.20092	17.96
e4	The2	3.59493	0.16448	21.86
e5	The3	2.99368	0.49861	6.00
e6	The4	259.57580	18.31150	14.18
d1	Psi1	5.67047	0.42301	13.41
d2	Psi2	4.51480	0.33532	13.46

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
e1	e3	The5	0.90580	0.12167	7.44
e2	e4	The5	0.90580	0.12167	7.44

The measurement scale of variables is often arbitrary. Therefore, it can be useful to look at the standardized equations produced by PROC CALIS. Output 25.1.13 displays the standardized equations and predicted moments. From the standardized structural equations for f1 and f2, you can conclude that SES (f3) has a larger impact on earlier Alienation (f1) than on later Alienation (f3). The squared multiple correlation for each equation is also shown in Output 25.1.13. These correlations indicate the proportion of systematic variance in the equations. Finally, correlations among the exogenous variables are shown.

Output 25.1.13 Standardized Solutions

Manifest Variable Equations with Standardized Estimates

```
        v1      =   0.8348 f1   +  0.5505 e1
        v2      =   0.7846 f1   +  0.6200 e2
        v3      =   0.8450 f2   +  0.5348 e3
        v4      =   0.7968 f2   +  0.6043 e4
        v5      =   0.8297 f3   +  0.5581 e5
        v6      =   0.6508*f3   +  0.7593 e6
                        Lamb
```

Latent Variable Equations with Standardized Estimates

```
f1      =  -0.5626*f3   +  0.8268 d1
                Gam1
f2      =   0.5692*f1   + -0.2064*f3   +  0.7080 d2
                Beta           Gam2
```

Squared Multiple Correlations

	Variable	Error Variance	Total Variance	R-Square
1	v1	3.60788	11.90391	0.6969
2	v2	3.59493	9.35145	0.6156
3	v3	3.60788	12.61575	0.7140
4	v4	3.59493	9.84539	0.6349
5	v5	2.99368	9.61000	0.6885
6	v6	259.57580	450.28800	0.4235
7	f1	5.67047	8.29603	0.3165
8	f2	4.51480	9.00787	0.4988

Correlations Among Exogenous Variables

Var1	Var2	Parameter	Estimate
e1	e3	The5	0.25106
e2	e4	The5	0.25197

The predicted covariances among the latent variables and between the observed and the latent variables are displayed in Output 25.1.14.

Output 25.1.14 Predicted Moments

Predicted Moments of Latent Variables

	f1	f2	f3
f1	8.296026985	5.924364730	-4.167911571
f2	5.924364730	9.007870649	-4.065656060
f3	-4.167911571	-4.065656060	6.616317547

Output 25.1.14 *continued*

Predicted Moments between Manifest and Latent Variables

	f1	f2	f3
v1	8.29602698	5.92436473	−4.16791157
v2	6.91059048	4.93499582	−3.47187034
v3	5.92436473	9.00787065	−4.06565606
v4	4.93499582	7.50355625	−3.38669150
v5	−4.16791157	−4.06565606	6.61631755
v6	−22.37688158	−21.82788734	35.52199986

For interpreting the model, these predicted moments are not as useful as the main results shown previously. However, these predicted moments can be useful for further analysis. For example, they can be useful in constructing bootstrap "populations" for resampling. Another use of these moments is to compute the latent variable score regression coefficients. PROC CALIS computes these coefficients automatically, as shown in Output 25.1.15.

Output 25.1.15 Latent Variable Score Regression Coefficients

Latent Variable Score Regression Coefficients

		f1	f2	f3
v1	Anomie (1967)	0.4131113567	0.0482681051	−.0521264408
v2	Anomie (1971)	0.3454029627	0.0400143300	−.0435560637
v3	Education	0.0526632293	0.4306175653	−.0399927539
v4	Powerlessness (1967)	0.0437036855	0.3600452776	−.0334000265
v5	Powerlessness (1971)	−.0749215200	−.0639697183	0.5057060770
v6	Occupational Status Index	−.0046390513	−.0039609288	0.0313127184

In Output 25.1.15, each latent variable is expressed as a linear combination of the observed variables. By computing these linear combinations for each individual, you can estimate the latent variable scores. See Chapter 76, "The SCORE Procedure," for more information about the creation of latent variable scores.

The total effects and indirect effects of the exogenous variables are displayed in Output 25.1.16. These results supplement to those shown in the linear equations in Output 25.1.12, which shows only the direct effects of predictor variables on outcome variables. Total, direct, and indirect effects have the following simple relationship:

$$\text{Total Effect} = \text{Direct Effect} + \text{Indirect Effect}$$

To illustrate, consider the relationships between latent factor f3 and variables v1–v4. In the linear equations shown in Output 25.1.12, latent factor f3 does not have direct effects on variables v1–v4. This does not mean that f3 has no effects on these variables at all. As shown in the first table of Output 25.1.16, latent factor f3 indeed has nonzero total effects on all variables, including variables v1–v4. In the next table that shows indirect effects, latent factor f3, again, has nonzero indirect effects on variables v1–v4, and these effects are identical to the total effects. Because the sum of

direct and indirect effects is the total effect, this means that the effects of f3 on v1–v4 are all *indirect*. Similar decomposition of effects can be made for other relationships. For example, while f1 has a total effect of 1.0 on v1, it has no indirect effect on v1. This means that all the effect of f1 on v1 is *direct*, which is also shown in an equation in Output 25.1.12. Finally, consider the effects of f3 on f2. In Output 25.1.16, latent factor f3 has nonzero total effect (−0.6145) and indirect effect (−0.3736) on f2, and these two effects are not identical. The difference of these two effects is the direct effect −0.2409, as shown in an equation in Output 25.1.12.

Output 25.1.16 Total and Indirect Effects

Total Effects

	f3	f1	f2
v1	−0.629944307	1.000000000	0.000000000
v2	−0.524743608	0.833000000	0.000000000
v3	−0.614489258	0.593112208	1.000000000
v4	−0.511869552	0.494062469	0.833000000
v5	1.000000000	0.000000000	0.000000000
v6	5.368847492	0.000000000	0.000000000
f1	−0.629944307	0.000000000	0.000000000
f2	−0.614489258	0.593112208	0.000000000

Indirect Effects

	f3	f1	f2
v1	−.6299443069	0.0000000000	0
v2	−.5247436076	0.0000000000	0
v3	−.6144892580	0.5931122083	0
v4	−.5118695519	0.4940624695	0
v5	0.0000000000	0.0000000000	0
v6	0.0000000000	0.0000000000	0
f1	0.0000000000	0.0000000000	0
f2	−.3736276589	0.0000000000	0

PROC CALIS can display Lagrange multiplier and Wald statistics for model modifications. Modification indices are displayed for each parameter matrix, as shown in Output 25.1.17 through Output 25.1.22. Only the Lagrange multiplier statistics have significance levels and approximate changes of values displayed. The significance level of the Wald statistic for a given parameter is the same as that shown in the equation output. An insignificant *p*-value for a Wald statistic means that the corresponding parameter can be dropped from the model without significantly worsening the fit of the model.

A significant *p*-value for a Lagrange multiplier test indicates that the model would achieve a better fit if the corresponding parameter were free. To aid in determining significant results, PROC CALIS displays the rank order of the 10 largest Lagrange multiplier statistics. For example, [E5:E2] in the _PHI_ matrix is associated with the largest Lagrange multiplier statistic; the associated *p*-value is 0.0067. This means that adding a parameter for the covariance between E5 and E2 will lead to a significantly better fit of the model. However, adding parameters indiscriminately can result in specification errors. An overfitted model might not perform well with future samples. As always, the decision to add parameters should be accompanied by consideration and knowledge of the ap-

plication area.

Output 25.1.17 Lagrange Multiplier and Wald Tests for _PHI_

Lagrange Multiplier and Wald Test Indices _PHI_ [9:9]
Symmetric Matrix
Univariate Tests for Constant Constraints
Lagrange Multiplier or Wald Index / Probability / Approx Change of Value

	f3	e1	e2	e3	e4
f3	107.1619	3.3903	3.3901	0.5752	0.5753
	.	0.0656	0.0656	0.4482	0.4482
	.	0.5079	−0.4231	0.2090	−0.1741
	[Phi]				
e1	3.3903	322.4501	0.1529	55.4237	1.2037
	0.0656	.	0.6958	.	0.2726
	0.5079	.	0.0900	.	−0.3262
		[The1]		[The5]	
e2	3.3901	0.1529	477.6768	0.5946	55.4237
	0.0656	0.6958	.	0.4406	.
	−0.4231	0.0900	.	0.2328	.
			[The2]		[The5]
e3	0.5752	55.4237	0.5946	322.4501	0.1528
	0.4482	.	0.4406	.	0.6958
	0.2090	.	0.2328	.	−0.0900
		[The5]		[The1]	
e4	0.5753	1.2037	55.4237	0.1528	477.6768
	0.4482	0.2726	.	0.6958	.
	−0.1741	−0.3262	.	−0.0900	.
			[The5]		[The2]
e5	.	5.8025	7.3649	1.5982	1.2044
	.	0.0160	0.0067	0.2062	0.2724
	.	0.5193	−0.5060	0.2709	−0.2037
	Sing				
e6	.	0.7398	1.4168	0.0991	0.0029
	.	0.3897	0.2339	0.7529	0.9568
	.	−1.2587	1.5431	−0.4579	0.0700
	Sing				
d1	.	0.4840	0.4840	1.1825	1.1825
	.	0.4866	0.4866	0.2768	0.2768
	.	0.2276	−0.1896	0.2984	−0.2486
	Sing				
d2	.	0.0000	0.0000	0.5942	0.5942
	.	0.9961	0.9961	0.4408	0.4408
	.	0.0014	−0.0011	−0.2806	0.2338
	Sing				

Output 25.1.17 *continued*

Lagrange Multiplier and Wald Test Indices _PHI_ [9:9]
Symmetric Matrix
Univariate Tests for Constant Constraints
Lagrange Multiplier or Wald Index / Probability / Approx Change of Value

	e5	e6	d1	d2
f3	.	.	.	.
	.	.	.	.
	.	.	.	.
	Sing	Sing	Sing	Sing
e1	5.8025	0.7398	0.4840	0.0000
	0.0160	0.3897	0.4866	0.9961
	0.5193	−1.2587	0.2276	0.0014
e2	7.3649	1.4168	0.4840	0.0000
	0.0067	0.2339	0.4866	0.9961
	−0.5060	1.5431	−0.1896	−0.0011
e3	1.5982	0.0991	1.1825	0.5942
	0.2062	0.7529	0.2768	0.4408
	0.2709	−0.4579	0.2984	−0.2806
e4	1.2044	0.0029	1.1825	0.5942
	0.2724	0.9568	0.2768	0.4408
	−0.2037	0.0700	−0.2486	0.2338
e5	36.0486	.	0.1033	0.1035
	.	.	0.7479	0.7477
	.	.	−0.2776	0.1062
	[The3]	Sing		
e6	.	200.9466	0.1034	0.1035
	.	.	0.7478	0.7477
	.	.	1.4906	−0.5700
	Sing	[The4]		
d1	0.1033	0.1034	179.6950	.
	0.7479	0.7478	.	.
	−0.2776	1.4906	.	.
			[Psi1]	Sing
d2	0.1035	0.1035	.	181.2787
	0.7477	0.7477	.	.
	0.1062	−0.5700	.	.
			Sing	[Psi2]

Output 25.1.18 Ranking of Lagrange Multipliers in _PHI_

Rank Order of the 10 Largest Lagrange Multipliers in _PHI_

Row	Column	Chi-Square	Pr > ChiSq
e5	e2	7.36486	0.0067
e5	e1	5.80246	0.0160
e1	f3	3.39030	0.0656
e2	f3	3.39013	0.0656
e5	e3	1.59820	0.2062
e6	e2	1.41677	0.2339
e5	e4	1.20437	0.2724
e4	e1	1.20367	0.2726
d1	e3	1.18251	0.2768
d1	e4	1.18249	0.2768

Output 25.1.19 Lagrange Multiplier and Wald Tests for _GAMMA_

```
          Lagrange Multiplier and Wald Test Indices _GAMMA_ [8:1]
                              General Matrix
                 Univariate Tests for Constant Constraints
   Lagrange Multiplier or Wald Index / Probability / Approx Change of Value

                                            f3

                            v1          3.3903
                                        0.0656
                                        0.0768


                            v2          3.3901
                                        0.0656
                                       -0.0639


                            v3          0.5752
                                        0.4482
                                        0.0316


                            v4          0.5753
                                        0.4482
                                       -0.0263


                            v5            .
                                          .
                                          .
                                          Sing

                            v6        153.2354
                                          .
                                          .
                                        [Lamb]

                            f1        125.0132
                                          .
                                          .
                                        [Gam1]

                            f2         19.2585
                                          .
                                          .
                                        [Gam2]
```

Output 25.1.20 Ranking of Lagrange Multipliers in _GAMMA_

Rank Order of the 4 Largest Lagrange Multipliers in _GAMMA_

Row	Column	Chi-Square	Pr > ChiSq
v1	f3	3.39030	0.0656
v2	f3	3.39013	0.0656
v4	f3	0.57526	0.4482
v3	f3	0.57523	0.4482

Output 25.1.21 Lagrange Multiplier and Wald Tests for _BETA_

Lagrange Multiplier and Wald Test Indices _BETA_ [8:8]
General Matrix
Identity-Minus-Inverse Model Matrix
Univariate Tests for Constant Constraints
Lagrange Multiplier or Wald Index / Probability / Approx Change of Value

	v1	v2	v3	v4
v1	.	0.1647	0.0511	0.8029
	.	0.6849	0.8212	0.3702
	.	-0.0159	-0.0063	-0.0284
	Sing			
v2	0.5957	.	0.6406	0.0135
	0.4402	.	0.4235	0.9076
	0.0218	.	0.0185	0.0032
		Sing		
v3	0.3839	0.3027	.	0.1446
	0.5355	0.5822	.	0.7038
	0.0178	0.0180	.	-0.0145
			Sing	
v4	0.4487	0.2519	0.0002	.
	0.5030	0.6157	0.9877	.
	-0.0160	-0.0144	-0.0004	.
				Sing
v5	5.4085	8.6455	2.7123	2.1457
	0.0200	0.0033	0.0996	0.1430
	0.1242	-0.1454	0.0785	-0.0674
v6	0.4209	1.4387	0.3044	0.0213
	0.5165	0.2304	0.5811	0.8841
	-0.2189	0.3924	-0.1602	0.0431
f1	1.0998	1.1021	1.6114	1.6128
	0.2943	0.2938	0.2043	0.2041
	0.0977	-0.0817	0.0993	-0.0831
f2	0.0193	0.0194	0.4765	0.4760
	0.8896	0.8892	0.4900	0.4902
	-0.0104	0.0087	-0.0625	0.0522

Output 25.1.21 *continued*

Lagrange Multiplier and Wald Test Indices _BETA_ [8:8]
General Matrix
Identity-Minus-Inverse Model Matrix
Univariate Tests for Constant Constraints
Lagrange Multiplier or Wald Index / Probability / Approx Change of Value

	v5	v6	f1	f2
v1	5.4083	0.1233	0.4047	0.4750
	0.0200	0.7255	0.5247	0.4907
	0.0697	0.0015	-0.0257	-0.0239
v2	5.8858	0.0274	0.4047	0.4750
	0.0153	0.8686	0.5247	0.4907
	-0.0609	-0.0006	0.0214	0.0199
v3	1.1537	0.0296	0.1588	0.0817
	0.2828	0.8634	0.6902	0.7750
	0.0322	0.0007	0.0144	-0.0110
v4	0.9867	0.1442	0.1588	0.0817
	0.3206	0.7041	0.6903	0.7750
	-0.0249	-0.0014	-0.0120	0.0092
v5	.	.	0.1033	0.1035
	.	.	0.7479	0.7476
	.	.	-0.0490	0.0329
	Sing	Sing		
v6	.	.	0.1034	0.1035
	.	.	0.7478	0.7477
	.	.	0.2629	-0.1765
	Sing	Sing		
f1	0.1032	0.1035	.	.
	0.7480	0.7477	.	.
	-0.0927	0.0057	.	.
			Sing	Sing
f2	0.1034	0.1035	160.7520	.
	0.7477	0.7477	.	.
	0.0355	-0.0022	.	.
			[Beta]	Sing

Output 25.1.22 Ranking of Lagrange Multipliers in _BETA_

Rank Order of the 10 Largest Lagrange Multipliers in _BETA_

Row	Column	Chi-Square	Pr > ChiSq
v5	v2	8.64546	0.0033
v2	v5	5.88576	0.0153
v5	v1	5.40848	0.0200
v1	v5	5.40832	0.0200
v5	v3	2.71233	0.0996
v5	v4	2.14572	0.1430
f1	v4	1.61279	0.2041
f1	v3	1.61137	0.2043
v6	v2	1.43867	0.2304
v3	v5	1.15372	0.2828

When you specify equality constraints, PROC CALIS displays Lagrange multiplier tests for releasing the constraints, as shown in Output 25.1.23. In the current example, none of the three constraints achieve a p-value smaller than 0.05. This means that releasing the constraints might not lead to a significantly better fit of the model. Therefore, all constraints are retained in the model.

Output 25.1.23 Tests for Equality Constraints

Univariate Lagrange Multiplier Test for Releasing Equality Constraints

Equality Constraint	-----Changes-----		Chi-Square	Pr > ChiSq
[e1:e1] = [e3:e3]	0.0293	-0.0308	0.02106	0.8846
[e2:e2] = [e4:e4]	-0.1342	0.1388	0.69488	0.4045
[e3:e1] = [e4:e2]	0.2468	-0.1710	1.29124	0.2558

The current model is specified using the LINEQS, STD, and COV statements. As discussed in the section "Getting Started: CALIS Procedure" on page 839, you can also specify the same model by using other specification methods. In the following statements, equivalent COSAN and RAM specifications of the current model are shown. These two specifications would give essentially the same estimation results for the model specified using the LINEQS model statements.

```
proc calis cov data=Wheaton tech=nr edf=931;
   Cosan J(9, Ide) * A(9, Gen, Imi) * P(9, Sym);
   Matrix A
           [ ,7] = 1. .833  5 * 0. Beta (.5) ,
           [ ,8] = 2 * 0.  1.  .833 ,
           [ ,9] = 4 * 0.  1.  Lamb Gam1-Gam2 (.5 2 * -.5);
   Matrix P
           [1,1] = The1-The2 The1-The4 (6 * 3.) ,
           [7,7] = Psi1-Psi2 Phi (2 * 4. 6.) ,
           [3,1] = The5 (.2) ,
           [4,2] = The5 (.2) ;
   Vnames J V1-V6 F1-F3 ,
          A = J ,
          P E1-E6 D1-D3 ;
run;
```

```
proc calis cov data=Wheaton tech=nr edf=931;
   Ram
      1   1  7  1.       ,
      1   2  7  .833     ,
      1   3  8  1.       ,
      1   4  8  .833     ,
      1   5  9  1.       ,
      1   6  9  .5    Lamb ,
      1   7  9  -.5   Gam1 ,
      1   8  7  .5    Beta ,
      1   8  9  -.5   Gam2 ,
      2   1  1  3.    The1 ,
      2   2  2  3.    The2 ,
      2   3  3  3.    The1 ,
      2   4  4  3.    The2 ,
      2   5  5  3.    The3 ,
      2   6  6  3.    The4 ,
      2   1  3  .2    The5 ,
      2   2  4  .2    The5 ,
      2   7  7  4.    Psi1 ,
      2   8  8  4.    Psi2 ,
      2   9  9  6.    Phi ;
   Vnames 1 F1-F3,
          2 E1-E6 D1-D3;
run;
```

Example 25.2: Simultaneous Equations with Intercept

The demand-and-supply food example of Kmenta (1971, pp. 565, 582) is used to illustrate the use of PROC CALIS for the estimation of intercepts and coefficients of simultaneous equations. The model is specified by two simultaneous equations containing two endogenous variables Q and P and three exogenous variables D, F, and Y,

$$Q_t(demand) = \alpha_1 + \beta_1 P_t + \gamma_1 D_t$$

$$Q_t(supply) = \alpha_2 + \beta_2 P_t + \gamma_2 F_t + \gamma_3 Y_t$$

for $t = 1, \ldots, 20$.

The LINEQS statement requires that each endogenous variable appear on the left-hand side of exactly one equation. Instead of analyzing the system

$$\mathbf{B}^*\boldsymbol{\eta} = \boldsymbol{\Gamma}\boldsymbol{\xi} + \boldsymbol{\zeta}$$

PROC CALIS analyzes the equivalent system

$$\boldsymbol{\eta} = \mathbf{B}\boldsymbol{\eta} + \boldsymbol{\Gamma}\boldsymbol{\xi} + \boldsymbol{\zeta}$$

with $\mathbf{B}^* = \mathbf{I} - \mathbf{B}$. This requires that one of the preceding equations be solved for P_t. Solving the second equation for P_t yields

$$P_t = \frac{1}{\beta_2} Q_t - \frac{\alpha_2}{\beta_2} - \frac{\gamma_2}{\beta_2} F_t - \frac{\gamma_3}{\beta_2} Y_t$$

You can estimate the intercepts of a system of simultaneous equations by applying PROC CALIS to the uncorrected covariance (UCOV) matrix of the data set that is augmented by an additional constant variable with the value 1. In the following statements, the uncorrected covariance matrix is augmented by an additional variable INTERCEPT by using the AUGMENT option. The PROC CALIS statement contains the options UCOV and AUG to compute and analyze an augmented UCOV matrix from the input data set FOOD.

```
data food;
title 'Food example of KMENTA(1971, p.565 & 582)';
  input Q P D F Y;
  label  Q='Food Consumption per Head'
         P='Ratio of Food Prices to General Price'
         D='Disposable Income in Constant Prices'
         F='Ratio of Preceding Years Prices'
         Y='Time in Years 1922-1941';
datalines;
  98.485  100.323   87.4   98.0   1
  99.187  104.264   97.6   99.1   2
 102.163  103.435   96.7   99.1   3
 101.504  104.506   98.2   98.1   4
 104.240   98.001   99.8  110.8   5
 103.243   99.456  100.5  108.2   6
 103.993  101.066  103.2  105.6   7
  99.900  104.763  107.8  109.8   8
 100.350   96.446   96.6  108.7   9
 102.820   91.228   88.9  100.6  10
  95.435   93.085   75.1   81.0  11
  92.424   98.801   76.9   68.6  12
  94.535  102.908   84.6   70.9  13
  98.757   98.756   90.6   81.4  14
 105.797   95.119  103.1  102.3  15
 100.225   98.451  105.1  105.0  16
 103.522   86.498   96.4  110.5  17
  99.929  104.016  104.4   92.5  18
 105.223  105.769  110.7   89.3  19
 106.232  113.490  127.1   93.0  20
;

proc calis ucov aug data=food pshort;
   lineqs
      Q = alf1 Intercept + alf2 P + alf3 D + E1,
      P = gam1 Intercept + gam2 Q + gam3 F + gam4 Y + E2;
   std
      E1-E2 = eps1-eps2;
   cov
      E1-E2 = eps3;
   bounds
      eps1-eps2 >= 0. ;
run;
```

The following statements, an essentially equivalent model specification, use program code to reparameterize the model in terms of the original equations; the output is displayed in Output 25.2.1.

```
proc calis data=food ucov aug pshort;
   lineqs
      Q = alphal Intercept + beta1 P + gamma1 D + E1,
      P = alpha2_b Intercept + gamma2_b F + gamma3_b Y + _b Q + E2;
   std
      E1-E2 = eps1-eps2;
   cov
      E1-E2 = eps3;
   parameters alpha2 (50.) beta2 gamma2 gamma3 (3*.25);
      alpha2_b = -alpha2 / beta2;
      gamma2_b = -gamma2 / beta2;
      gamma3_b = -gamma3 / beta2;
      _b       = 1 / beta2;
   bounds
      eps1-eps2 >= 0. ;
run;
```

Output 25.2.1 Food Example of Kmenta (1971)

```
                          LINEQS Model Statement


                       Matrix      Rows    Columns    ------Matrix Type-------

 Term 1            1   _SEL_          6          8    SELECTION
                   2   _BETA_         8          8    EQSBETA        IMINUSINV
                   3   _GAMMA_        8          6    EQSGAMMA
                   4   _PHI_          6          6    SYMMETRIC

                            The 2 Endogenous Variables

     Manifest        Q          P
     Latent

                            The 6 Exogenous Variables

     Manifest        D          F          Y          Intercept
     Latent
     Error           E1         E2

                       Levenberg-Marquardt Optimization

                        Scaling Update of More (1978)

                    Parameter Estimates                  10
                    Functions (Observations)             21
                    Lower Bounds                          2
                    Upper Bounds                          0

                               Optimization Start

Active Constraints                    0  Objective Function         2.3500065042
Max Abs Gradient Element    203.9741437  Radius                     62167.829174
```

Output 25.2.1 *continued*

Iter	Rest arts	Func Calls	Act Con	Objective Function	Obj Fun Change	Max Abs Gradient Element	Lambda	Actual Over Pred Change
1	0	2	0	1.19094	1.1591	3.9410	0	0.688
2	0	5	0	0.32678	0.8642	9.9864	0.00127	2.356
3	0	7	0	0.19108	0.1357	5.5100	0.00006	0.685
4	0	10	0	0.16682	0.0243	2.0513	0.00005	0.867
5	0	12	0	0.16288	0.00393	1.0570	0.00014	0.828
6	0	13	0	0.16132	0.00156	0.3643	0.00004	0.864
7	0	15	0	0.16077	0.000557	0.2176	0.00006	0.984
8	0	16	0	0.16052	0.000250	0.1819	0.00001	0.618
9	0	17	0	0.16032	0.000201	0.0662	0	0.971
10	0	18	0	0.16030	0.000011	0.0195	0	1.108
11	0	19	0	0.16030	6.116E-7	0.00763	0	1.389
12	0	20	0	0.16030	9.454E-8	0.00301	0	1.389
13	0	21	0	0.16030	1.461E-8	0.00118	0	1.388
14	0	22	0	0.16030	2.269E-9	0.000465	0	1.395
15	0	23	0	0.16030	3.59E-10	0.000182	0	1.427

Optimization Results

Iterations	15	Function Calls	24
Jacobian Calls	16	Active Constraints	0
Objective Function	0.1603035477	Max Abs Gradient Element	0.0001820805
Lambda	0	Actual Over Pred Change	1.4266532872
Radius	0.0010322573		

GCONV convergence criterion satisfied.

Output 25.2.1 *continued*

Fit Function	0.1603
Goodness of Fit Index (GFI)	0.9530
GFI Adjusted for Degrees of Freedom (AGFI)	0.0120
Root Mean Square Residual (RMR)	2.0653
Standardized Root Mean Square Residual (SRMR)	0.0009
Parsimonious GFI (Mulaik, 1989)	0.0635
Chi-Square	3.0458
Chi-Square DF	1
Pr > Chi-Square	0.0809
Independence Model Chi-Square	534.27
Independence Model Chi-Square DF	15
RMSEA Estimate	0.3281
RMSEA 90% Lower Confidence Limit	.
RMSEA 90% Upper Confidence Limit	0.7777
ECVI Estimate	1.8270
ECVI 90% Lower Confidence Limit	.
ECVI 90% Upper Confidence Limit	3.3493
Probability of Close Fit	0.0882
Bentler's Comparative Fit Index	0.9961
Normal Theory Reweighted LS Chi-Square	2.8142
Akaike's Information Criterion	1.0458
Bozdogan's (1987) CAIC	-0.9500
Schwarz's Bayesian Criterion	0.0500
McDonald's (1989) Centrality	0.9501
Bentler & Bonett's (1980) Non-normed Index	0.9409
Bentler & Bonett's (1980) NFI	0.9943
James, Mulaik, & Brett (1982) Parsimonious NFI	0.0663
Z-Test of Wilson & Hilferty (1931)	1.4250
Bollen (1986) Normed Index Rho1	0.9145
Bollen (1988) Non-normed Index Delta2	0.9962
Hoelter's (1983) Critical N	25

Manifest Variable Equations with Estimates

```
Q         =  -0.2295*P         +  0.3100*D         + 93.6193*Intercept
                    beta1               gamma1              alphal

                +  1.0000 E1


P         =   4.2140*Q         + -0.9305*F         + -1.5579*Y
                    _b                  gamma2_b            gamma3_b

                +  -218.9*Intercept +  1.0000 E2
                          alpha2_b
```

Output 25.2.1 *continued*

Variances of Exogenous Variables

Variable	Parameter	Estimate
D		10154
F		9989
Y		151.05263
Intercept		1.05263
E1	eps1	3.51274
E2	eps2	105.06746

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate
D	F		9994
D	Y		1101
F	Y		1046
D	Intercept		102.66842
F	Intercept		101.71053
Y	Intercept		11.05263
E1	E2	eps3	-18.87270

Manifest Variable Equations with Standardized Estimates

```
Q         =  -0.2278*P         +  0.3016*D         +  0.9272*Intercept
                  beta1                gamma1               alphal

               +  0.0181 E1


P         =   4.2467*Q         + -0.9048*F         + -0.1863*Y
                  _b                   gamma2_b             gamma3_b

               + -2.1849*Intercept +  0.0997 E2
                         alpha2_b
```

Squared Multiple Correlations

	Variable	Error Variance	Total Variance	R-Square
1	Q	3.51274	10730	0.9997
2	P	105.06746	10565	0.9901

Output 25.2.1 *continued*

Correlations Among Exogenous Variables

Var1	Var2	Parameter	Estimate
D	F		0.99237
D	Y		0.88903
F	Y		0.85184
D	Intercept		0.99308
F	Intercept		0.99188
Y	Intercept		0.87652
E1	E2	eps3	-0.98237

Additional PARMS and Dependent Parameters

The Number of Dependent Parameters is 4

Parameter	Estimate	Standard Error	t Value
alpha2	51.94453	.	.
beta2	0.23731	.	.
gamma2	0.22082	.	.
gamma3	0.36971	.	.
_b	4.21397	.	.
gamma2_b	-0.93053	.	.
gamma3_b	-1.55794	.	.
alpha2_b	-218.89288	.	.

You can obtain almost equivalent results by applying the SAS/ETS procedure SYSLIN to this problem.

Example 25.3: Second-Order Confirmatory Factor Analysis

A second-order confirmatory factor analysis model is applied to a correlation matrix of Thurstone reported by McDonald (1985). Using the LINEQS statement, the three-term second-order factor analysis model is specified in equations notation. The first-order loadings for the three factors, f1, f2, and f3, each refer to three variables, X1-X3, X4-X6, and X7-X9. One second-order factor, f4, reflects the correlations among the three first-order factors. The second-order factor correlation matrix P is defined as a 1×1 identity matrix. Choosing the second-order uniqueness matrix U2 as a diagonal matrix with parameters U21-U23 gives an unidentified model. To compute identified maximum likelihood estimates, the matrix U2 is defined as a 3×3 identity matrix. The following statements generate results that are partially displayed in Output 25.3.1 through Output 25.3.4:

```
data Thurst(TYPE=CORR);
title "Example of THURSTONE resp. McDONALD (1985, p.57, p.105)";
   _TYPE_ = 'CORR'; Input _NAME_ $ Obs1-Obs9;
   label Obs1='Sentences' Obs2='Vocabulary' Obs3='Sentence Completion'
         Obs4='First Letters' Obs5='Four-letter Words' Obs6='Suffices'
         Obs7='Letter series' Obs8='Pedigrees' Obs9='Letter Grouping';
   datalines;
Obs1  1.       .      .      .      .      .      .      .      .
Obs2   .828   1.      .      .      .      .      .      .      .
Obs3   .776   .779   1.      .      .      .      .      .      .
Obs4   .439   .493    .460  1.      .      .      .      .      .
Obs5   .432   .464    .425   .674  1.      .      .      .      .
Obs6   .447   .489    .443   .590   .541  1.      .      .      .
Obs7   .447   .432    .401   .381   .402   .288  1.      .      .
Obs8   .541   .537    .534   .350   .367   .320   .555  1.      .
Obs9   .380   .358    .359   .424   .446   .325   .598   .452  1.
;

proc calis data=Thurst method=max edf=212 pestim se;
lineqs
   Obs1 = X1 F1 + E1,
   Obs2 = X2 F1 + E2,
   Obs3 = X3 F1 + E3,
   Obs4 = X4 F2 + E4,
   Obs5 = X5 F2 + E5,
   Obs6 = X6 F2 + E6,
   Obs7 = X7 F3 + E7,
   Obs8 = X8 F3 + E8,
   Obs9 = X9 F3 + E9,
   F1   = X10 F4 + E10,
   F2   = X11 F4 + E11,
   F3   = X12 F4 + E12;
std
   F4      = 1.,
   E1-E9   = U11-U19,
   E10-E12 = 3 * 1.;
bounds
   0. <= U11-U19;
run;
```

Output 25.3.1 Optimization

```
                  Parameter Estimates                  21
                  Functions (Observations)             45
                  Lower Bounds                          9
                  Upper Bounds                          0

                           Optimization Start

Active Constraints                     0  Objective Function         0.7151823452
Max Abs Gradient Element   0.4067179803  Radius                     2.2578762496
```

Output 25.3.1 *continued*

Iter	Rest arts	Func Calls	Act Con	Objective Function	Obj Fun Change	Max Abs Gradient Element	Lambda	Actual Over Pred Change
1	0	2	0	0.23113	0.4840	0.1299	0	1.363
2	0	3	0	0.18322	0.0479	0.0721	0	1.078
3	0	4	0	0.18051	0.00271	0.0200	0	1.006
4	0	5	0	0.18022	0.000289	0.00834	0	1.093
5	0	6	0	0.18018	0.000041	0.00251	0	1.201
6	0	7	0	0.18017	6.523E-6	0.00114	0	1.289
7	0	8	0	0.18017	1.085E-6	0.000388	0	1.347
8	0	9	0	0.18017	1.853E-7	0.000173	0	1.380
9	0	10	0	0.18017	3.208E-8	0.000063	0	1.399
10	0	11	0	0.18017	5.593E-9	0.000028	0	1.408
11	0	12	0	0.18017	9.79E-10	0.000011	0	1.414

Optimization Results

Iterations	11	Function Calls	13
Jacobian Calls	12	Active Constraints	0
Objective Function	0.1801712147	Max Abs Gradient Element	0.0000105805
Lambda	0	Actual Over Pred Change	1.4135882439
Radius	0.0002026368		

GCONV convergence criterion satisfied.

Output 25.3.2 Fit Statistics

Example of THURSTONE resp. McDONALD (1985, p.57, p.105)
Identified Second Order Confirmatory Factor Analysis
C = F1 * F2 * P * F2' * F1' + F1 * U2 * F1' + U1, With P=U2=Ide

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Fit Function	0.1802
Goodness of Fit Index (GFI)	0.9596
GFI Adjusted for Degrees of Freedom (AGFI)	0.9242
Root Mean Square Residual (RMR)	0.0436
Standardized Root Mean Square Residual (SRMR)	0.0436
Parsimonious GFI (Mulaik, 1989)	0.6397
Chi-Square	38.1963
Chi-Square DF	24
Pr > Chi-Square	0.0331
Independence Model Chi-Square	1101.9
Independence Model Chi-Square DF	36
RMSEA Estimate	0.0528
RMSEA 90% Lower Confidence Limit	0.0153
RMSEA 90% Upper Confidence Limit	0.0831
ECVI Estimate	0.3881
ECVI 90% Lower Confidence Limit	.
ECVI 90% Upper Confidence Limit	0.4888
Probability of Close Fit	0.4088
Bentler's Comparative Fit Index	0.9867
Normal Theory Reweighted LS Chi-Square	40.1947
Akaike's Information Criterion	-9.8037
Bozdogan's (1987) CAIC	-114.4747
Schwarz's Bayesian Criterion	-90.4747
McDonald's (1989) Centrality	0.9672
Bentler & Bonett's (1980) Non-normed Index	0.9800
Bentler & Bonett's (1980) NFI	0.9653
James, Mulaik, & Brett (1982) Parsimonious NFI	0.6436
Z-Test of Wilson & Hilferty (1931)	1.8373
Bollen (1986) Normed Index Rho1	0.9480
Bollen (1988) Non-normed Index Delta2	0.9868
Hoelter's (1983) Critical N	204

Output 25.3.3 Estimation Results

```
          Manifest Variable Equations with Estimates


Obs1     =   0.5151*F1       +  1.0000 E1
Std Err      0.0629 X1
t Value      8.1868
Obs2     =   0.5203*F1       +  1.0000 E2
Std Err      0.0634 X2
t Value      8.2090
Obs3     =   0.4874*F1       +  1.0000 E3
Std Err      0.0608 X3
t Value      8.0151
Obs4     =   0.5211*F2       +  1.0000 E4
Std Err      0.0611 X4
t Value      8.5342
Obs5     =   0.4971*F2       +  1.0000 E5
Std Err      0.0590 X5
t Value      8.4213
Obs6     =   0.4381*F2       +  1.0000 E6
Std Err      0.0560 X6
t Value      7.8283
Obs7     =   0.4524*F3       +  1.0000 E7
Std Err      0.0660 X7
t Value      6.8584
Obs8     =   0.4173*F3       +  1.0000 E8
Std Err      0.0622 X8
t Value      6.7135
Obs9     =   0.4076*F3       +  1.0000 E9
Std Err      0.0613 X9
t Value      6.6484

   Latent Variable Equations with Estimates


F1       =   1.4438*F4       +  1.0000 E10
Std Err      0.2565 X10
t Value      5.6282
F2       =   1.2538*F4       +  1.0000 E11
Std Err      0.2114 X11
t Value      5.9320
F3       =   1.4065*F4       +  1.0000 E12
Std Err      0.2689 X12
t Value      5.2307
```

Output 25.3.3 *continued*

Variances of Exogenous Variables

Variable	Parameter	Estimate	Standard Error	t Value
F4		1.00000		
E1	U11	0.18150	0.02848	6.37
E2	U12	0.16493	0.02777	5.94
E3	U13	0.26713	0.03336	8.01
E4	U14	0.30150	0.05102	5.91
E5	U15	0.36450	0.05264	6.93
E6	U16	0.50642	0.05963	8.49
E7	U17	0.39032	0.05934	6.58
E8	U18	0.48138	0.06225	7.73
E9	U19	0.50509	0.06333	7.98
E10		1.00000		
E11		1.00000		
E12		1.00000		

Output 25.3.4 Standardized and Supplementary Results

Manifest Variable Equations with Standardized Estimates

```
Obs1    =   0.9047*F1   +  0.4260 E1
                X1
Obs2    =   0.9138*F1   +  0.4061 E2
                X2
Obs3    =   0.8561*F1   +  0.5168 E3
                X3
Obs4    =   0.8358*F2   +  0.5491 E4
                X4
Obs5    =   0.7972*F2   +  0.6037 E5
                X5
Obs6    =   0.7026*F2   +  0.7116 E6
                X6
Obs7    =   0.7808*F3   +  0.6248 E7
                X7
Obs8    =   0.7202*F3   +  0.6938 E8
                X8
Obs9    =   0.7035*F3   +  0.7107 E9
                X9
```

Latent Variable Equations with Standardized Estimates

```
F1      =   0.8221*F4   +  0.5694 E10
                X10
F2      =   0.7818*F4   +  0.6235 E11
                X11
F3      =   0.8150*F4   +  0.5794 E12
                X12
```

Output 25.3.4 *continued*

Squared Multiple Correlations

	Variable	Error Variance	Total Variance	R-Square
1	Obs1	0.18150	1.00000	0.8185
2	Obs2	0.16493	1.00000	0.8351
3	Obs3	0.26713	1.00000	0.7329
4	Obs4	0.30150	1.00000	0.6985
5	Obs5	0.36450	1.00000	0.6355
6	Obs6	0.50642	1.00000	0.4936
7	Obs7	0.39032	1.00000	0.6097
8	Obs8	0.48138	1.00000	0.5186
9	Obs9	0.50509	1.00000	0.4949
10	F1	1.00000	3.08452	0.6758
11	F2	1.00000	2.57213	0.6112
12	F3	1.00000	2.97832	0.6642

To compute McDonald's unidentified model, you would have to change the STD and BOUNDS statements to include three more parameters:

```
std
   F4      = 1.,
   E1-E9   = U11-U19,
   E10-E12 = U21-U23;
bounds
   0. <= U11-U19,
   0. <= U21-U23;
```

The unidentified model is indicated in the output by an analysis of the linear dependencies in the approximate Hessian matrix (not shown). Because the information matrix is singular, standard errors are computed based on a Moore-Penrose inverse. The results computed by PROC CALIS differ from those reported by McDonald (1985). In the case of an unidentified model, the parameter estimates are not unique.

To specify the identified model by using the COSAN model statement, you can use the following statements:

```
proc calis data=Thurst method=max edf=212 pestim se;
   cosan F1(3) * F2(1) * P(1,Ide) + F1(3) * U2(3,Ide) + U1(9,Dia);
   matrix F1
          [ ,1] = X1-X3,
          [ ,2] = 3 * 0. X4-X6,
          [ ,3] = 6 * 0. X7-X9;
   matrix F2
          [ ,1] = X10-X12;
   matrix U1
          [1,1] = U11-U19;
   bounds
          0. <= U11-U19;
run;
```

Because PROC CALIS cannot compute initial estimates for a model specified by the general COSAN statement, this analysis might require more iterations than one that uses the LINEQS statement, depending on the precision of the processor.

Example 25.4: Linear Relations among Factor Loadings

In this example, a confirmatory factor-analysis model with linear constraints on loadings is specified by the COSAN statement. SAS programming statements are used to set the constraints. In the context of the current example, the differences between fitting covariance structures and correlation structures will also be discussed.

The correlation matrix of six variables from Kinzer and Kinzer (N=326) is used by Guttman (1957) as an example that yields an approximate simplex. McDonald (1980) uses this data set as an example of factor analysis where he supposes that the loadings of the second factor are linear functions of the loadings on the first factor. Let **B** be the factor loading matrix containing the two factors and six variables so that

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \\ b_{51} & b_{52} \\ b_{61} & b_{62} \end{pmatrix}$$

and

$$b_{j2} = \alpha + \beta b_{j1}, \quad j = 1, \ldots, 6$$

The correlation structures are represented by

$$\mathbf{P} = \mathbf{B}\mathbf{B}' + \Psi$$

where $\Psi = \text{diag}(\psi_{11}, \psi_{22}, \psi_{33}, \psi_{44}, \psi_{55}, \psi_{66})$ represents the diagonal matrix of unique variances for the variables.

With parameters α and β being unconstrained, McDonald (1980) has fitted an underidentified model with seven degrees of freedom. Browne (1982) imposes the following identification condition:

$$\beta = -1$$

In this example, Browne's identification condition is imposed. The following statements specify the confirmatory factor model:

```
Data Kinzer(TYPE=CORR);
Title "Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)";
   _TYPE_ = 'CORR'; INPUT _NAME_ $ var1-var6;
   Datalines;
var1  1.00   .     .     .     .
var2   .51  1.00   .     .     .
var3   .46   .51  1.00   .     .
var4   .46   .47   .54  1.00   .
var5   .40   .39   .49   .57  1.00   .
var6   .33   .39   .47   .45   .56  1.00
  ;

proc calis data=Kinzer method=max nobs=326 nose;
   cosan B(2,Gen) * I(2,Ide) + U(6,Ide) * PSI(6,DIA);
   matrix B
          [ ,1]= b11 b21 b31 b41 b51 b61,
          [ ,2]= b12 b22 b32 b42 b52 b62;
   matrix Psi
          [1,1]= psi1-psi6;
   parameters alpha = .5;
   /* SAS Programming Statements */
   b12  = alpha - b11;
   b22  = alpha - b21;
   b32  = alpha - b31;
   b42  = alpha - b41;
   b52  = alpha - b51;
   b62  = alpha - b61;
   vnames B  Fact1 Fact2,
          I  Fact1 Fact2,
          U  var1-var6,
          PSI var1-var6;
run;
```

In the COSAN statement, you specify the model matrices for the correlation structures. Matrix **B** is the 6×2 factor loading matrix. Matrix **I** is a fixed 2×2 identity matrix, representing the factor variances and covariances. In the next term, matrix **U** is a 6×6 identity matrix, appearing there only because it makes the specification conform to the format of the generalized COSAN model. The **PSI** matrix is the matrix for variances and covariances for errors. The matrix specification in the COSAN statement realizes the following matrix model:

$$\mathbf{P} = \mathbf{BIB}' + \mathbf{U}\Psi\mathbf{U}' = \mathbf{BB}' + \Psi$$

which is the intended correlation structures.

In the MATRIX statements, parameters are defined. The loading parameters are all specified with prefix "b" and are located in the **B** matrix. Error variances are all specified with prefix "psi" and are located in the **PSI** matrix. In this model, all manifest variables have nonzero loadings on the two factors. These loading parameters are specified after the equal signs and are named with the prefix "b".

An additional parameter alpha is specified in the PARAMETERS statement with an initial value of 0.5. Then, six SAS programming statements are used to define the loadings on the second factor as functions of the loadings on the first factor.

Finally, the VNAMES statement is used to name the columns of the model matrices.

In the specification, there are 12 loadings in matrix **B** and six unique variances in matrix **PSI**. Adding the parameter alpha in the list, there are 19 parameters in total. However, the loading parameters are not all independent of each other. As defined in the SAS programming statements, six loadings are dependent. This reduces the number of free parameters to 13. Hence the degrees of freedom for the model is $8 = 21 - 13$.

In Output 25.4.1, a fit summary table is shown. The chi-square test statistic of model fit is 10.337 with df =8 (p = 0.242). This indicates a good fit.

Output 25.4.1 Fit of the Correlation Structures

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
Fitting Correlations As Covariance Structures Directly

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Fit Function	0.0318
Goodness of Fit Index (GFI)	0.9897
GFI Adjusted for Degrees of Freedom (AGFI)	0.9730
Root Mean Square Residual (RMR)	0.0409
Standardized Root Mean Square Residual (SRMR)	0.0409
Parsimonious GFI (Mulaik, 1989)	0.5278
Chi-Square	10.3374
Chi-Square DF	8
Pr > Chi-Square	0.2421
Independence Model Chi-Square	682.87
Independence Model Chi-Square DF	15
RMSEA Estimate	0.0300
RMSEA 90% Lower Confidence Limit	.
RMSEA 90% Upper Confidence Limit	0.0756
ECVI Estimate	0.1136
ECVI 90% Lower Confidence Limit	.
ECVI 90% Upper Confidence Limit	0.1525
Probability of Close Fit	0.7137
Bentler's Comparative Fit Index	0.9965
Normal Theory Reweighted LS Chi-Square	10.1441
Akaike's Information Criterion	-5.6626
Bozdogan's (1987) CAIC	-43.9578
Schwarz's Bayesian Criterion	-35.9578
McDonald's (1989) Centrality	0.9964
Bentler & Bonett's (1980) Non-normed Index	0.9934
Bentler & Bonett's (1980) NFI	0.9849
James, Mulaik, & Brett (1982) Parsimonious NFI	0.5253
Z-Test of Wilson & Hilferty (1931)	0.7019
Bollen (1986) Normed Index Rho1	0.9716
Bollen (1988) Non-normed Index Delta2	0.9965
Hoelter's (1983) Critical N	489

The estimated factor loading matrix and error covariance matrix are presented in Output 25.4.2. The estimates for alpha and other dependent parameters are presented in Output 25.4.3.

Output 25.4.2 Estimates

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
Fitting Correlations As Covariance Structures Directly

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Estimated Parameter Matrix B[6:2]
General Matrix

	Fact1	Fact2
var1	0.3609	0.6174
	[b11]	<b12>
var2	0.3212	0.6571
	[b21]	<b22>
var3	0.4859	0.4923
	[b31]	<b32>
var4	0.5745	0.4038
	[b41]	<b42>
var5	0.7985	0.1797
	[b51]	<b52>
var6	0.6736	0.3046
	[b61]	<b62>

Output 25.4.2 *continued*

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
Fitting Correlations As Covariance Structures Directly

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Estimated Parameter Matrix PSI[6:6]
Diagonal Matrix

	var1	var2	var3	var4	var5	var6
var1	0.5304 [psi1]	0	0	0	0	0
var2	0	0.4499 [psi2]	0	0	0	0
var3	0	0	0.4876 [psi3]	0	0	0
var4	0	0	0	0.4728 [psi4]	0	0
var5	0	0	0	0	0.3112 [psi5]	0
var6	0	0	0	0	0	0.5382 [psi6]

Estimated Parameter Matrix U[6:6]
Identity Matrix
Constant Model Matrix

Output 25.4.3 Additional Parameter

Additional PARMS and Dependent Parameters

The Number of Dependent Parameters is 6

Parameter	Estimate
alpha	0.97825
b12	0.61736
b22	0.65709
b32	0.49234
b42	0.40378
b52	0.17973
b62	0.30462

All these estimates are essentially the same as reported in Browne (1982). Notice that there are no standard error estimates in the output, as requested by the NOSE option in the PROC CALIS statement. Standard error estimates are not of interest in the current example.

Fitting the Correct Correlation Structures

In specifying the previous factor model by using COSAN, covariance structures rather than the correlation structures have been incorrectly assumed. That is, when fitting the correlation structures, the diagonal elements of **P** must always be fixed ones. But these fixed ones have not been handled properly in the previous code, resulting in a specification of covariance structures rather than correlation structures. To fix the problem, additional model constraints on the diagonal elements of the correlation structures represented by $(\mathbf{BB}' + \Psi)$ are required. For example, it can be done by constraining the error variances

$$\Psi_{jj} = 1.0 - b_{j1}^2 - b_{j2}^2, \quad j = 1, \ldots, 6$$

Other constraints could also serve the same purpose, but these are the most convenient and therefore are used subsequently. Without setting up these constraints, the previous PROC CALIS run is analyzing covariance structures. As pointed out in Browne (1982), fitting such covariance structures directly is not entirely appropriate for analyzing correlations. For the current data example, according to Browne (1982), the estimates obtained from fitting the wrong covariance structure model are still consistent (as if they were estimating the population parameters in the correlation structures). However, the chi-square test statistic as reported in the preceding section is not correct.

In addition, due to the fact that discrepancy functions used in PROC CALIS are derived for covariance matrices rather than correlation matrices, PROC CALIS is essentially set up for analyzing covariance structures (with or without mean structures), but not correlation structures. Hence, the statistical theory behind PROC CALIS applies to covariance structure analysis, but might not generalize to correlation structure analysis. Despite that, can PROC CALIS somehow fit the correct correlation structures to the current data without compromising the statistical theory? The answer is yes.

First, recall that the correlation structures are represented by:

$$\mathbf{P} = \mathbf{BB}' + \Psi$$

As before, in the **B** matrix, there are six linear constraints on the factor loadings. In addition, the diagonal elements of $(\mathbf{BB}' + \Psi)$ must be ones. To analyze the correlation structures by using PROC CALIS, a covariance structure model with such correlation structures embedded is now specified. That is, the covariance structure to be fitted by PROC CALIS is:

$$\mathbf{\Sigma} = \mathbf{DPD}' = \mathbf{DBB}'\mathbf{D}' + \mathbf{D}\Psi\mathbf{D}'$$

where **D** is a 6×6 diagonal matrix containing the population standard deviations for the manifest variables. This form of covariance structures implies the specification of the following COSAN model:

```
proc calis data=Kinzer method=max nobs=326 nose;
   cosan D(6,DIA) * B(2,Gen) * I(2,Ide) + D(6,DIA) * PSI(6,DIA);
   matrix B
          [ ,1]= b11 b21 b31 b41 b51 b61,
          [ ,2]= b12 b22 b32 b42 b52 b62;
   matrix Psi
          [1,1]= psi1-psi6;
   matrix D
          [1,1]= d1-d6 (6 * 1.);
   parameters alpha (1.);
   /* SAS Programming Statements */
   /* 12 Constraints on Correlation structures */
   b12  = alpha - b11;
   b22  = alpha - b21;
   b32  = alpha - b31;
   b42  = alpha - b41;
   b52  = alpha - b51;
   b62  = alpha - b61;
   psi1 = 1. - b11 * b11 - b12 * b12;
   psi2 = 1. - b21 * b21 - b22 * b22;
   psi3 = 1. - b31 * b31 - b32 * b32;
   psi4 = 1. - b41 * b41 - b42 * b42;
   psi5 = 1. - b51 * b51 - b52 * b52;
   psi6 = 1. - b61 * b61 - b62 * b62;
   vnames D  var1-var6,
          B  Fact1 Fact2,
          I  Fact1 Fact2,
          Psi var1-var6;
run;
```

In the COSAN statement, the covariance structures are described in the form of the generalized COSAN model. The most significant change is the inclusion of matrix **D** for the population standard deviations. Accompanying this change is the addition of a MATRIX statement for defining the error variance parameters in matrix **D**. Also, in this new specification, Psi1-Psi6 are defined as functions of the loading parameters:

$$\Psi_{jj} = 1. - b_{j1}^2 - b_{j2}^2, \quad j = 1, \ldots, 6$$

This ensures that the correlation structures will have fixed ones for the diagonal elements.

Other constraints remain the same way as defined in the previous run. For example, the loadings on the second factor are functions of the loadings on the first factor:

$$b_{j2} = \alpha - b_{j1}, \quad j = 1, \ldots, 6$$

The fit summary is presented in Output 25.4.4. The chi-square test statistic is 14.63 with df=8 (p = 0.067). This shows that the previous chi-square test based on fitting a wrong covariance structure model is indeed questionable.

Output 25.4.4 Model Fit of the Correlation Structures

Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)
Fitting Correlations By Using Correct Covariance Structures

The CALIS Procedure
Covariance Structure Analysis: Maximum Likelihood Estimation

Fit Function	0.0450
Goodness of Fit Index (GFI)	0.9849
GFI Adjusted for Degrees of Freedom (AGFI)	0.9604
Root Mean Square Residual (RMR)	0.0378
Standardized Root Mean Square Residual (SRMR)	0.0378
Parsimonious GFI (Mulaik, 1989)	0.5253
Chi-Square	14.6269
Chi-Square DF	8
Pr > Chi-Square	0.0668
Independence Model Chi-Square	682.87
Independence Model Chi-Square DF	15
RMSEA Estimate	0.0505
RMSEA 90% Lower Confidence Limit	.
RMSEA 90% Upper Confidence Limit	0.0908
ECVI Estimate	0.1268
ECVI 90% Lower Confidence Limit	.
ECVI 90% Upper Confidence Limit	0.1729
Probability of Close Fit	0.4382
Bentler's Comparative Fit Index	0.9901
Normal Theory Reweighted LS Chi-Square	14.9199
Akaike's Information Criterion	-1.3731
Bozdogan's (1987) CAIC	-39.6683
Schwarz's Bayesian Criterion	-31.6683
McDonald's (1989) Centrality	0.9899
Bentler & Bonett's (1980) Non-normed Index	0.9814
Bentler & Bonett's (1980) NFI	0.9786
James, Mulaik, & Brett (1982) Parsimonious NFI	0.5219
Z-Test of Wilson & Hilferty (1931)	1.5034
Bollen (1986) Normed Index Rho1	0.9598
Bollen (1988) Non-normed Index Delta2	0.9902
Hoelter's (1983) Critical N	346

Estimates of the loadings, population standard deviations, error variances, and the alpha parameter are presented in Output 25.4.5 and Output 25.4.6.

Output 25.4.5 Estimates of Loadings and Standard Deviations

Estimated Parameter Matrix B[6:2]
General Matrix

	Fact1	Fact2
var1	0.3422 [b11]	0.6318 <b12>
var2	0.3210 [b21]	0.6531 <b22>
var3	0.4918 [b31]	0.4822 <b32>
var4	0.5755 [b41]	0.3985 <b42>
var5	0.7769 [b51]	0.1971 <b52>
var6	0.6666 [b61]	0.3074 <b62>

Estimated Parameter Matrix D[6:6]
Diagonal Matrix

	var1	var2	var3	var4	var5	var6
var1	1.0077 [d1]	0	0	0	0	0
var2	0	0.9971 [d2]	0	0	0	0
var3	0	0	0.9908 [d3]	0	0	0
var4	0	0	0	0.9909 [d4]	0	0
var5	0	0	0	0	0.9964 [d5]	0
var6	0	0	0	0	0	1.0169 [d6]

Output 25.4.6 Estimates of Error Variances and Alpha

Estimated Parameter Matrix PSI[6:6]
Diagonal Matrix

	var1	var2	var3	var4	var5	var6
var1	0.4837 <psi1>	0	0	0	0	0
var2	0	0.4705 <psi2>	0	0	0	0
var3	0	0	0.5256 <psi3>	0	0	0
var4	0	0	0	0.5100 <psi4>	0	0
var5	0	0	0	0	0.3576 <psi5>	0
var6	0	0	0	0	0	0.4612 <psi6>

Additional PARMS and Dependent Parameters

The Number of Dependent Parameters is 12

Parameter	Estimate
alpha	0.97400
b12	0.63184
b22	0.65305
b32	0.48222
b42	0.39849
b52	0.19715
b62	0.30741
psi1	0.48370
psi2	0.47051
psi3	0.52561
psi4	0.50999
psi5	0.35763
psi6	0.46115

Except for the population standard deviation parameter d's, all other parameters estimated in the current model can be compared with those from the previous fitting of an incorrect covariance structure model. Although estimates in the current model do not differ very much from those in the previous analysis, it is assuring that they are obtained from fitting a correctly specified covariance structure model with the intended correlation structures embedded.

Example 25.5: Ordinal Relations among Factor Loadings

The same data set as in Example 25.4 is used in McDonald (1980) for analysis with ordinally constrained factor loadings. In Example 25.4, the results of the linearly constrained factor analysis show that the loadings of the two factors are ordered as 2, 1, 3, 4, 6, 5. McDonald (1980) then tests the hypothesis that the factor loadings are all nonnegative and can be ordered in the following manner:

$$b_{11} \le b_{21} \le b_{31} \le b_{41} \le b_{51} \le b_{61}$$

$$b_{12} \ge b_{22} \ge b_{32} \ge b_{42} \ge b_{52} \ge b_{62}$$

In this example, these ordinal relationships are implemented using the LINCON statement in the following COSAN model specification:

```
proc calis data=Kinzer method=max nobs=326 nose;
   cosan D(6,DIA) * B(2,Gen) * I(2,Ide) + D(6,DIA) * PSI(6,DIA);
   matrix B
          [ ,1]= 0.  b21 b31 b41 b51 b61,
          [ ,2]= b12 b22 b32 b42 b52 b62;
   matrix Psi
          [1,1]= psi1-psi6;
   matrix D
          [1,1]= d1-d6 (6 * 1.);
   lincon  b21  <= b31,
           b31  <= b41,
           b41  <= b51,
           b51  <= b61,
           b62  <= b52,
           b52  <= b42,
           b42  <= b32,
           b32  <= b22,
           b22  <= b12;
   bounds b21 >= 0;
   /* SAS Programming Statements */
   /* 6 Constraints on Correlation structures */
   psi1 = 1. - b12 * b12;
   psi2 = 1. - b21 * b21 - b22 * b22;
   psi3 = 1. - b31 * b31 - b32 * b32;
   psi4 = 1. - b41 * b41 - b42 * b42;
   psi5 = 1. - b51 * b51 - b52 * b52;
   psi6 = 1. - b61 * b61 - b62 * b62;
   vnames D  var1-var6,
          B  Fact1 Fact2,
          I  Fact1 Fact2,
          Psi var1-var6;
run;
```

Again, as in Example 25.4, correlation structures are analyzed in the current example so that the error variance parameters psi1-psi6 are defined as functions of the loadings in the SAS programming statements. However, the loading parameters are not constrained the same way as in Example 25.4.

First, b11 is fixed at 0 for identification purposes. Then, the LINCON statement is used to specify the ordinal relations of the factor loadings. Parameter b21 is set to be nonnegative in the BOUNDS statement.

As shown in Output 25.5.1, the solution converges in 10 iterations. In the fit summary table, the chi-square test statistic is 8.48 ($df = 6$, $p = 0.20$). This indicates a good fit. However, in the model there are 11 loading parameters (the b's) and 6 population standard deviation parameters (the d's). The degrees of freedom should have been $4 = 21 - 11 - 6$, but why is this number 6 in the fit summary table?

Output 25.5.1 Final Iteration Status and Fit

```
                          Optimization Results

Iterations                            10  Function Calls                     13
Jacobian Calls                        11  Active Constraints                  2
Objective Function          0.0260990149  Max Abs Gradient Element    6.2794755E-6
Lambda                                 0  Actual Over Pred Change     0.9491590286
Radius                      0.0001243346

ABSGCONV convergence criterion satisfied.

WARNING: There are 2 active constraints at the solution.  The standard errors
         and Chi-Square test statistic assume the solution is located in the
         interior of the parameter space and hence do not apply if it is likely
         that some different set of inequality constraints could be active.


NOTE: The degrees of freedom are increased by the number of active constraints
      (see Dijkstra, 1992). The number of parameters in calculating fit indices
      is decreased by the number of active constraints. To turn off the
      adjustment, use the NOADJDF option.
```

Output 25.5.1 *continued*

Fit Function	0.0261
Goodness of Fit Index (GFI)	0.9914
GFI Adjusted for Degrees of Freedom (AGFI)	0.9699
Root Mean Square Residual (RMR)	0.0169
Standardized Root Mean Square Residual (SRMR)	0.0169
Parsimonious GFI (Mulaik, 1989)	0.3966
Chi-Square	8.4822
Chi-Square DF	6
Pr > Chi-Square	0.2049
Independence Model Chi-Square	682.87
Independence Model Chi-Square DF	15
RMSEA Estimate	0.0357
RMSEA 90% Lower Confidence Limit	.
RMSEA 90% Upper Confidence Limit	0.0860
ECVI Estimate	0.1204
ECVI 90% Lower Confidence Limit	.
ECVI 90% Upper Confidence Limit	0.1576
Probability of Close Fit	0.6154
Bentler's Comparative Fit Index	0.9963
Normal Theory Reweighted LS Chi-Square	8.4575
Akaike's Information Criterion	-3.5178
Bozdogan's (1987) CAIC	-32.2392
Schwarz's Bayesian Criterion	-26.2392
McDonald's (1989) Centrality	0.9962
Bentler & Bonett's (1980) Non-normed Index	0.9907
Bentler & Bonett's (1980) NFI	0.9876
James, Mulaik, & Brett (1982) Parsimonious NFI	0.3950
Z-Test of Wilson & Hilferty (1931)	0.8281
Bollen (1986) Normed Index Rho1	0.9689
Bollen (1988) Non-normed Index Delta2	0.9963
Hoelter's (1983) Critical N	484

The reason is that there are two active constraints in the solution, making it two free parameters fewer in the final solution than originally specified. Active constraints are those inequality constraints that are fulfilled on the boundary equalities. As shown in the "Optimization Results" table, the number of active constraints for the current fitting is two. The default treatment in PROC CALIS is to treat these active constraints as if they were going to happen for all possible repeated sampling. This might as well be seen as fitting the active equality constraints on every possible repeated samples. This results in an increase of the degrees of freedom for model fit, as adjusted in the current fit summary table in Output 25.5.1. To warn you about the degrees-of-freedom adjustment, the following messages are also printed with the output:

```
WARNING: There are 2 active constraints at the solution.  The
         standard errors and Chi-Square test statistic assume
         solution is located in the interior of the parameter
         space and hence do not apply if it is likely that some
         different set of inequality constraints could be
         active.
```

```
NOTE: The degrees of freedom are increased by the number of
      active constraints (see Dijkstra, 1992). The number of
      parameters in calculating fit indices is decreased by the
      number of active constraints. To turn off the adjustment,
      use the NOADJDF option.
```

When active constraints are encountered, you need to be cautious about two implications. First, the estimates fall on the boundary of the parameter space originally specified. As shown in Output 25.5.2, estimates for b51 and b61 are the same, and so are the pair of estimates for b12 and b22. These pairs of parameters were originally constrained by inequalities in the model. For example, b61 was constrained to be at least as big as b51. The fact that this constraint is honored only on the bound means that a better model fit might exist with b61 being smaller than b51. Similarly, a better model fit might result without requiring b12 to be at least as big as b22. Solutions with active constraints thus might imply that the original strict inequality constraints are not appropriate for the data.

Output 25.5.2 Estimates

Estimated Parameter Matrix B[6:2]
General Matrix

	Fact1	Fact2
var1	0	0.7100
		[b12]
var2	0.0393	0.7100
	[b21]	[b22]
var3	0.2463	0.6799
	[b31]	[b32]
var4	0.3295	0.6561
	[b41]	[b42]
var5	0.5432	0.5541
	[b51]	[b52]
var6	0.5432	0.4733
	[b61]	[b62]

Output 25.5.2 *continued*

Estimated Parameter Matrix D[6:6]
Diagonal Matrix

	var1	var2	var3	var4	var5	var6
var1	1.0022 [d1]	0	0	0	0	0
var2	0	0.9985 [d2]	0	0	0	0
var3	0	0	1.0004 [d3]	0	0	0
var4	0	0	0	1.0004 [d4]	0	0
var5	0	0	0	0	0.9990 [d5]	0
var6	0	0	0	0	0	1.0021 [d6]

Estimated Parameter Matrix PSI[6:6]
Diagonal Matrix

	var1	var2	var3	var4	var5	var6
var1	0.4959 <psi1>	0	0	0	0	0
var2	0	0.4944 <psi2>	0	0	0	0
var3	0	0	0.4771 <psi3>	0	0	0
var4	0	0	0	0.4610 <psi4>	0	0
var5	0	0	0	0	0.3979 <psi5>	0
var6	0	0	0	0	0	0.4809 <psi6>

The second implication for the presence of active constraints is that the chi-square test statistic and the standard error estimates are computed as if repeated samples were fitted by the model with the presence of the active equality constraints. The degrees-of-freedom adjustment by PROC CALIS is based on this assumption. However, if the presence of particular active constraints in fitting reflects only a rare sampling event that is not likely to happen in repeated sampling, degrees-of-freedom adjustment or even the computation of chi-square statistic and standard error estimates might not be justified. Unfortunately, whether the active constraints are reflecting the truth of the model or pure sampling fluctuation is usually difficult to determine.

Example 25.6: Longitudinal Factor Analysis

The following example (McDonald 1980) illustrates both the ability of PROC CALIS to formulate complex covariance structure analysis problems by the generalized COSAN matrix model and the use of programming statements to impose nonlinear constraints on the parameters. The example is a longitudinal factor analysis that uses the Swaminathan (1974) model. For $m = 3$ tests, $k = 3$ occasions, and $r = 2$ factors the matrix model is formulated in the section "First-Order Autoregressive Longitudinal Factor Model" on page 834 as follows:

$$\mathbf{C} = \mathbf{F}_1\mathbf{F}_2\mathbf{F}_3\mathbf{L}\mathbf{F}_3^{-1}\mathbf{F}_2^{-1}\mathbf{P}(\mathbf{F}_2^{-1})'(\mathbf{F}_3^{-1})'\mathbf{L}'\mathbf{F}_3'\mathbf{F}_2'\mathbf{F}_1' + \mathbf{U}^2$$

$$F_1 = \begin{pmatrix} B_1 & & \\ & B_2 & \\ & & B_3 \end{pmatrix}, \quad F_2 = \begin{pmatrix} I_2 & & \\ & D_2 & \\ & & D_2 \end{pmatrix}, \quad F_3 = \begin{pmatrix} I_2 & & \\ & I_2 & \\ & & D_3 \end{pmatrix}$$

$$L = \begin{pmatrix} I_2 & o & o \\ I_2 & I_2 & o \\ I_2 & I_2 & I_2 \end{pmatrix}, \quad P = \begin{pmatrix} I_2 & & \\ & S_2 & \\ & & S_3 \end{pmatrix}, \quad U = \begin{pmatrix} U_{11} & U_{12} & U_{13} \\ U_{21} & U_{22} & U_{23} \\ U_{31} & U_{32} & U_{33} \end{pmatrix}$$

$$S_2 = I_2 - D_2^2, \qquad S_3 = I_2 - D_3^2$$

The Swaminathan longitudinal factor model assumes that the factor scores for each (m) common factor change from occasion to occasion (k) according to a first-order autoregressive scheme. The matrix $\mathbf{F}_1$ contains the k factor loading matrices $\mathbf{B}_1, \mathbf{B}_2, \mathbf{B}_3$ (each is $n \times m$). The matrices $\mathbf{D}_2, \mathbf{D}_3, \mathbf{S}_2, \mathbf{S}_3$ and $\mathbf{U}_{ij}, i, j = 1, \ldots, k$, are diagonal, and the matrices $\mathbf{D}_i$ and $\mathbf{S}_i, i = 2, \ldots, k$, are subjected to the constraint

$$\mathbf{S}_i + \mathbf{D}_i^2 = \mathbf{I}$$

Since the constructed correlation matrix given by McDonald (1980) is singular, only unweighted least squares estimates can be computed. The following statements specify the COSAN model for the correlation structures.

```
data Mcdon(TYPE=CORR);
Title "Swaminathan's Longitudinal Factor Model, Data: McDONALD(1980)";
Title2 "Constructed Singular Correlation Matrix, GLS & ML not possible";
   _TYPE_ = 'CORR'; INPUT _NAME_ $ Obs1-Obs9;
   datalines;
Obs1  1.000    .      .      .      .      .      .      .      .
Obs2   .100  1.000    .      .      .      .      .      .      .
Obs3   .250   .400  1.000    .      .      .      .      .      .
Obs4   .720   .108   .270  1.000    .      .      .      .      .
Obs5   .135   .740   .380   .180  1.000    .      .      .      .
Obs6   .270   .318   .800   .360   .530  1.000    .      .      .
Obs7   .650   .054   .135   .730   .090   .180  1.000    .      .
Obs8   .108   .690   .196   .144   .700   .269   .200  1.000    .
Obs9   .189   .202   .710   .252   .336   .760   .350   .580  1.000
     ;
```

```
proc calis data=Mcdon method=ls tech=nr nobs=100;
cosan B(6,Gen) * D1(6,Dia) * D2(6,Dia) * T(6,Low) * D3(6,Dia,Inv)
               * D4(6,Dia,Inv) * P(6,Dia) + U(9,Sym);
   matrix B
            [ ,1]= X1-X3,
            [ ,2]= 0. X4-X5,
            [ ,3]= 3 * 0. X6-X8,
            [ ,4]= 4 * 0. X9-X10,
            [ ,5]= 6 * 0. X11-X13,
            [ ,6]= 7 * 0. X14-X15;
   matrix D1
            [1,1]= 2 * 1. X16 X17 X16 X17;
   matrix D2
            [1,1]= 4 * 1. X18 X19;
   matrix T
            [1,1]= 6 * 1.,
            [3,1]= 4 * 1.,
            [5,1]= 2 * 1.;
   matrix D3
            [1,1]= 4 * 1. X18 X19;
   matrix D4
            [1,1]= 2 * 1. X16 X17 X16 X17;
   matrix P
            [1,1]= 2 * 1. X20-X23;
   matrix U
            [1,1]= X24-X32,
            [4,1]= X33-X38,
            [7,1]= X39-X41;
   bounds 0. <= X24-X32,
         -1. <= X16-X19 <= 1.;
   X20 = 1. - X16 * X16;
   X21 = 1. - X17 * X17;
   X22 = 1. - X18 * X18;
   X23 = 1. - X19 * X19;
run;
```

Because this formulation of Swaminathan's model in general leads to an unidentified problem, the results given here are different from those reported by McDonald (1980). The displayed output of PROC CALIS also indicates that the fitted central model matrices **P** and **U** are not positive-definite. The BOUNDS statement constrains the diagonals of the matrices **P** and **U** to be nonnegative, but this cannot prevent **U** from having three negative eigenvalues. The fact that many of the published results for more complex models in covariance structure analysis are connected to unidentified problems implies that more theoretical work should be done to study the general features of such models.

References

Akaike, H. (1974), "A New Look at the Statistical Identification Model," *IEEE Transactions on Automatic Control*, 19, 716–723.

Akaike, H. (1987), "Factor Analysis and AIC," *Psychometrika*, 52, 317–332.

Beale, E. M. L. (1972), "A Derivation of Conjugate Gradients," in *Numerical Methods for Nonlinear Optimization*, ed. F. A. Lootsma, London: Academic Press.

Bentler, P. M. (1985), *Theory and Implementation of EQS: A Structural Equations Program*, Manual for Program Version 2.0, Los Angeles: BMDP Statistical Software.

Bentler, P. M. (1986), *Lagrange Multiplier and Wald Tests for EQS and EQS/PC*, Los Angeles: BMDP Statistical Software.

Bentler, P. M. (1989), *EQS, Structural Equations, Program Manual*, Program Version 3.0, Los Angeles: BMDP Statistical Software.

Bentler, P. M. and Bonett, D. G. (1980), "Significance Tests and Goodness of Fit in the Analysis of Covariance Structures," *Psychological Bulletin*, 88, 588–606.

Bentler, P. M. and Weeks, D. G. (1980), "Linear Structural Equations with Latent Variables," *Psychometrika*, 45, 289–308.

Bishop, Y. M. M., Fienberg, S. E., and Holland, P. W. (1977), *Discrete Multivariate Analysis: Theory and Practice*, Cambridge and London: MIT Press.

Bollen, K. A. (1986), "Sample Size and Bentler and Bonett's Nonnormed Fit Index," *Psychometrika*, 51, 375–377.

Bollen, K. A. (1989a), "A New Incremental Fit Index for General Structural Equation Models," *Sociological Methods and Research*, 17, 303–316.

Bollen, K. A. (1989b), *Structural Equations with Latent Variables*, New York: John Wiley & Sons.

Bozdogan, H. (1987), "Model Selection and Akaike's Information Criterion (AIC): The General Theory and its Analytical Extensions," *Psychometrika*, 52, 345–370.

Browne, M. W. (1974), "Generalized Least Squares Estimators in the Analysis of Covariance Structures," *South African Statistical Journal*, 8, 1–24.

Browne, M. W. (1982), "Covariance Structures," in *Topics in Multivariate Analyses*, ed. D. M. Hawkins, New York: Cambridge University Press.

Browne, M. W. (1984), "Asymptotically Distribution-Free Methods for the Analysis of Covariance Structures," *British Journal of Mathematical and Statistical Psychology*, 37, 62–83.

Browne, M. W. and Cudeck, R. (1993), "Alternative Ways of Assessing Model Fit," in *Testing Structural Equation Models*, ed. K. A. Bollen and S. Long, Newbury Park, CA: Sage Publications.

Browne, M. W. and Du Toit, S. H. C. (1992), "Automated Fitting of Nonstandard Models," *Multivariate Behavioral Research*, 27, 269–300.

Browne, M. W. and Shapiro, A. (1986), "The Asymptotic Covariance Matrix of Sample Correlation Coefficients under General Conditions," *Linear Algebra and Its Applications*, 82, 169–176.

Bunch, J. R. and Kaufman, K. (1977), "Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Systems," *Mathematics of Computation*, 31, 162–179.

Buse, A. (1982), "The Likelihood Ratio, Wald, and Lagrange Multiplier Tests: An Expository Note," *The American Statistician*, 36, 153–157.

Chamberlain, R. M., Powell, M. J. D., Lemarechal, C., and Pedersen, H. C. (1982), "The Watchdog Technique for Forcing Convergence in Algorithms for Constrained Optimization," *Mathematical Programming*, 16, 1–17.

DeLeeuw, J. (1983), "Models and Methods for the Analysis of Correlation Coefficients," *Journal of Econometrics*, 22, 113–137.

Dennis, J. E., Gay, D. M., and Welsch, R. E. (1981), "An Adaptive Nonlinear Least-Squares Algorithm," *ACM Trans. Math. Software*, 7, 348–368.

Dennis, J. E. and Mei, H. H. W. (1979), "Two New Unconstrained Optimization Algorithms Which Use Function and Gradient Values," *J. Optim. Theory Appl.*, 28, 453–482.

Dijkstra, T. K. (1992), "On Statistical Inference with Parameter Estimates on the Boundary of the Parameter Space," *British Journal of Mathematical and Statistical Psychology*, 45, 289–309.

Everitt, B. S. (1984), *An Introduction to Latent Variable Methods*, London: Chapman & Hall.

Fletcher, R. (1980), *Practical Methods of Optimization*, Vol. 1, Chichester: John Wiley & Sons.

Fletcher, R. (1987), *Practical Methods of Optimization*, Second Edition, Chichester: John Wiley & Sons.

Fuller, W. A. (1987), *Measurement Error Models*, New York: John Wiley & Sons.

Gay, D. M. (1983), "Subroutines for Unconstrained Minimization," *ACM Trans. Math. Software*, 9, 503–524.

Gill, E. P., Murray, W., Saunders, M. A., and Wright, M. H. (1983), "Computing Forward-Difference Intervals for Numerical Optimization," *SIAM J. Sci. Stat. Comput.*, 4, 310–321.

Gill, E. P., Murray, W., and Wright, M. H. (1981), *Practical Optimization*, London: Academic Press.

Gill, E. P., Murray, W., Saunders, M. A., and Wright, M. H. (1984), "Procedures for Optimization Problems with a Mixture of Bounds and General Linear Constraints," *ACM Trans. Math. Software*, 10, 282–298.

Guttman, L. (1957), "Empirical Verification of the Radex Structure of Mental Abilities and Personality Traits," *Educational and Psychological Measurement*, 17, 391–407.

Hägglund, G. (1982), "Factor Analysis by Instrumental Variable Methods," *Psychometrika*, 47, 209–222.

Hoelter, J. W. (1983), "The Analysis of Covariance Structures: Goodness-of-Fit Indices," *Sociological Methods and Research*, 11, 325–344.

James, L. R., Mulaik, S. A., and Brett, J. M. (1982), *Causal Analysis: Assumptions, Models, and Data*, Beverly Hills, CA: Sage Publications.

Jennrich, R. I. (1987), "Tableau Algorithms for Factor Analysis by Instrumental Variable Methods," *Psychometrika*, 52, 469–476.

Jöreskog, K. G. (1973), "A General Method for Estimating a Linear Structural Equation System," in *Structural Equation Models in the Social Sciences*, ed. A. S. Goldberger and O. D. Duncan, New York: Seminar Press.

Jöreskog, K. G. (1978), "Structural Analysis of Covariance and Correlation Matrices," *Psychometrika*, 43, 443–477.

Jöreskog, K. G. and Sörbom, D. (1985), *LISREL VI; Analysis of Linear Structural Relationships by Maximum Likelihood, Instrumental Variables, and Least Squares*, Uppsala: University of Uppsala.

Jöreskog, K. G. and Sörbom, D. (1988), *LISREL 7: A Guide to the Program and Applications*, Chicago: SPSS Inc.

Keesling, J. W. (1972), "Maximum Likelihood Approaches to Causal Analysis," Ph.D. dissertation, University of Chicago, 1972.

Kmenta, J. (1971), *Elements of Econometrics*, New York: Macmillan.

Lawley, D. N. and Maxwell, A. E. (1971), *Factor Analysis as a Statistical Method*, New York: American Elsevier.

Lee, S. Y. (1985), "On Testing Functional Constraints in Structural Equation Models," *Biometrika*, 72, 125–131.

Loehlin, J. C. (1987), *Latent Variable Models, An Introduction to Factor, Path, and Structural Analysis*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Long, J. S. (1983), *Covariance Structure Models, an Introduction to LISREL*, Beverly Hills, CA: Sage Publications.

MacCallum, R. (1986), "Specification Searches in Covariance Structure Modeling," *Psychological Bulletin*, 100, 107–120.

McArdle, J. J. (1980), "Causal Modeling Applied to Psychonomic Systems Simulation," *Behavior Research Methods & Instrumentation*, 12, 193–209.

McArdle, J. J. (1988), "Dynamic but Structural Equation Modeling of Repeated Measures Data," in *The Handbook of Multivariate Experimental Psychology*, ed. J. R. Nesselroade and R. B. Cattell, New York: Plenum Press.

McArdle, J. J. and McDonald, R. P. (1984), "Some Algebraic Properties of the Reticular Action Model," *British Journal of Mathematical and Statistical Psychology*, 37, 234–251.

McDonald, R. P. (1978), "A Simple Comprehensive Model for the Analysis of Covariance Structures," *British Journal of Mathematical and Statistical Psychology*, 31, 59–72.

McDonald, R. P. (1980), "A Simple Comprehensive Model for the Analysis of Covariance Structures: Some Remarks on Applications," *British Journal of Mathematical and Statistical Psychology*, 33, 161–183.

McDonald, R. P. (1985), *Factor Analysis and Related Methods*, Hillsdale, NJ: Lawrence Erlbaum Associates.

McDonald, R. P. (1989), "An Index of Goodness-of-Fit Based on Noncentrality," *Journal of Classification*, 6, 97–103.

McDonald, R. P. and Hartmann, W. (1992), "A Procedure for Obtaining Initial Values of Parameters in the RAM Model," *Multivariate Behavioral Research*, 27, 57–176.

Moré, J. J. (1978), "The Levenberg-Marquardt Algorithm: Implementation and Theory," in *Numerical Analysis—Dundee 1977*, ed. G. A. Watson, Lecture Notes in Mathematics 630, Berlin: Springer-Verlag.

Moré, J. J. and Sorensen, D. C. (1983), "Computing a Trust-Region Step," *SIAM J. Sci. Stat. Comput.*, 4, 553–572.

Mulaik, S. A., James, L. R., Van Alstine, J., Bennett, N., Lind, S. and Stilwell, C. D. (1989), "Evaluation of Goodness-of-Fit Indices for Structural Equation Models," *Psychological Bulletin*, 105, 430–445.

Polak, E. (1971), *Computational Methods in Optimization*, New York: Academic Press.

Powell, J. M. D. (1977), "Restart Procedures for the Conjugate Gradient Method," *Math. Prog.*, 12, 241–254.

Powell, J. M. D. (1978a), "A Fast Algorithm for Nonlinearly Constraint Optimization Calculations," in *Numerical Analysis, Dundee 1977, Lecture Notes in Mathematics 630*, ed. G. A. Watson, Berlin: Springer-Verlag, 144–175.

Powell, J. M. D. (1978b), "Algorithms for Nonlinear Constraints That Use Lagrangian Functions," *Mathematical Programming*, 14, 224–248.

Powell, M. J. D. (1982a), "Extensions to Subroutine VF02AD," in *Systems Modeling and Optimization, Lecture Notes In Control and Information Sciences 38*, ed. R. F. Drenick and F. Kozin, Berlin: Springer-Verlag, 529–538.

Powell, J. M. D. (1982b), "VMCWD: A Fortran Subroutine for Constrained Optimization," *DAMTP 1982/NA4*, Cambridge, England.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Sclove, L. S. (1987), "Application of Model-Selection Criteria to Some Problems in Multivariate Analysis," *Psychometrika*, 52, 333–343.

Steiger, J. H. and Lind, J. C. (1980), "Statistically Based Tests for the Number of Common Factors," paper presented at the annual meeting of the Psychometric Society, Iowa City, IA.

Swaminathan, H. (1974), "A General Factor Model for the Description of Change," Report LR-74-9, Laboratory of Psychometric and Evaluative Research, University of Massachusetts.

Tucker, L. R. and Lewis, C. (1973), "A Reliability Coefficient for Maximum Likelihood Factor Analysis," *Psychometrika*, 38, 1–10.

Wheaton, B., Muthèn, B., Alwin, D. F., and Summers, G. F. (1977), "Assessing Reliability and Stability in Panel Models," in *Sociological Methodology*, ed. D. R. Heise, San Francisco: Jossey Bass.

Wiley, D. E. (1973), "The Identification Problem for Structural Equation Models with Unmeasured Variables," in *Structural Equation Models in the Social Sciences*, ed. A. S. Goldberger and O. D. Duncan, New York: Seminar Press, 69–83.

Wilson, E. B. and Hilferty, M. M. (1931), "The Distribution of Chi-Square," *Proceeding of the National Academy of Science*, 17, 694.

Chapter 26

The CANCORR Procedure

Contents

Overview: CANCORR Procedure **1029**
Background 1030
Getting Started: CANCORR Procedure **1031**
Syntax: CANCORR Procedure **1037**
PROC CANCORR Statement 1037
BY Statement 1043
FREQ Statement 1043
PARTIAL Statement 1043
VAR Statement 1044
WEIGHT Statement 1044
WITH Statement 1044
Details: CANCORR Procedure **1044**
Missing Values 1044
Formulas 1045
Output Data Sets 1046
Computational Resources 1048
Displayed Output 1049
ODS Table Names 1051
Example: CANCORR Procedure **1053**
Example 26.1: Canonical Correlation Analysis of Fitness Club Data 1053
References **1058**

Overview: CANCORR Procedure

The CANCORR procedure performs canonical correlation, partial canonical correlation, and canonical redundancy analysis.

Canonical correlation is a generalization of multiple correlation for analyzing the relationship between two sets of variables. In multiple correlation, you examine the relationship between a linear combination of a set of explanatory variables, **X**, and a *single* response variable, **Y**. In canonical correlation, you examine the relationship between linear combinations of the set of **X** variables and linear combinations of a *set* of **Y** variables. These linear combinations are called *canonical*

variables or *canonical variates*. Either set of variables can be considered explanatory or response variables, since the statistical model is symmetric in the two sets of variables. Simple and multiple correlation are special cases of canonical correlation in which one or both sets contain a single variable.

The CANCORR procedure tests a series of hypotheses that each canonical correlation and all smaller canonical correlations are zero in the population. PROC CANCORR uses an F approximation (Rao 1973; Kshirsagar 1972) that gives better small sample results than the usual χ^2 approximation. At least one of the two sets of variables should have an approximate multivariate normal distribution in order for the probability levels to be valid.

Both standardized and unstandardized canonical coefficients are computed, as well as the four *canonical structure* matrices showing correlations between the two sets of canonical variables and the two sets of original variables. A canonical redundancy analysis (Stewart and Love 1968; Cooley and Lohnes 1971) can also be done. PROC CANCORR provides multiple regression analysis options to aid in interpreting the canonical correlation analysis. You can examine the linear regression of each variable on the opposite set of variables.

PROC CANCORR can produce a data set containing the scores of each observation on each canonical variable, and you can use the PRINT procedure to list these values. A plot of each canonical variable against its counterpart in the other group is often useful, and you can use PROC SGPLOT with the output data set to produce these plots. A second output data set contains the canonical correlations, coefficients, and most other statistics computed by the procedure.

Background

Canonical correlation was developed by Hotelling (1935, 1936). The application of canonical correlation is discussed by Cooley and Lohnes (1971), Tatsuoka (1971), and Mardia, Kent, and Bibby (1979). One of the best theoretical treatments is given by Kshirsagar (1972).

Given a set of p **X** variables and q **Y** variables, the CANCORR procedure finds the linear combinations

$$w_1 = a_{11}x_1 + a_{21}x_2 + \cdots + a_{p1}x_p$$
$$v_1 = b_{11}y_1 + b_{21}y_2 + \cdots + b_{q1}y_q$$

such that the two canonical variables, w_1 and v_1, have the largest possible correlation. This maximized correlation between the two canonical variables is the first canonical correlation. The coefficients of the linear combinations are canonical coefficients or canonical weights. It is customary to normalize the canonical coefficients so that each canonical variable has a variance of 1.

PROC CANCORR continues by finding a second set of canonical variables, uncorrelated with the first pair, that produces the second-highest correlation coefficient. That is, the second pair of canonical variables is

$$w_2 = a_{12}x_1 + a_{22}x_2 + \cdots + a_{p2}x_p$$
$$v_2 = b_{12}y_1 + b_{22}y_2 + \cdots + b_{q2}y_q$$

such that w_2 is uncorrelated with w_1 and v_1, v_2 is uncorrelated with w_1 and v_1, and w_2 and v_2 have the largest possible correlation subject to these constraints. The process of constructing canonical variables continues until the number of pairs of canonical variables is $\min(p, q)$, the number of variables in the smaller group.

Each canonical variable is uncorrelated with all the other canonical variables of either set except for the one corresponding canonical variable in the opposite set. The canonical coefficients are not generally orthogonal, however, so the canonical variables do not represent jointly perpendicular directions through the space of the original variables.

The first canonical correlation is at least as large as the multiple correlation between any variable and the opposite set of variables. It is possible for the first canonical correlation to be very large while all the multiple correlations for predicting one of the original variables from the opposite set of canonical variables are small. Canonical redundancy analysis (Stewart and Love 1968; Cooley and Lohnes 1971; van den Wollenberg 1977) examines how well the original variables can be predicted from the canonical variables.

PROC CANCORR can also perform partial canonical correlation, which is a multivariate generalization of ordinary partial correlation (Cooley and Lohnes 1971; Timm 1975). Most commonly used parametric statistical methods, ranging from *t* tests to multivariate analysis of covariance, are special cases of partial canonical correlation.

Getting Started: CANCORR Procedure

The following example demonstrates how you can use the CANCORR procedure to calculate and test canonical correlations between two sets of variables.

Suppose you want to determine the degree of correspondence between a set of job characteristics and measures of employee satisfaction. Using a survey instrument for employees, you calculate three measures of job satisfaction. With another instrument designed for supervisors, you calculate the corresponding job characteristics profile.

Your three variables associated with job satisfaction are as follows:

- career track satisfaction: employee satisfaction with career direction and the possibility of future advancement, expressed as a percent
- management and supervisor satisfaction: employee satisfaction with supervisor's communication and management style, expressed as a percent
- financial satisfaction: employee satisfaction with salary and other benefits, using a scale measurement from 1 to 10 (1=unsatisfied, 10=satisfied)

The three variables associated with job characteristics are as follows:

- task variety: degree of variety involved in tasks, expressed as a percent
- feedback: degree of feedback required in job tasks, expressed as a percent
- autonomy: degree of autonomy required in job tasks, expressed as a percent

The following statements create the SAS data set Jobs and request a canonical correlation analysis:

```
data Jobs;
   input Career Supervisor Finance Variety Feedback Autonomy;
   label Career    ='Career Satisfaction' Variety ='Task Variety'
         Supervisor='Supervisor Satisfaction' Feedback='Amount of Feedback'
         Finance   ='Financial Satisfaction' Autonomy='Degree of Autonomy';
   datalines;
72  26  9          10  11  70
63  76  7          85  22  93
96  31  7          83  63  73
96  98  6          82  75  97
84  94  6          36  77  97
66  10  5          28  24  75
31  40  9          64  23  75
45  14  2          19  15  50
42  18  6          33  13  70
79  74  4          23  14  90
39  12  2          37  13  70
54  35  3          23  74  53
60  75  5          45  58  83
63  45  5          22  67  53
;

 proc cancorr data=Jobs
                  vprefix=Satisfaction wprefix=Characteristics
                  vname='Satisfaction Areas' wname='Job Characteristics';
    var  Career Supervisor Finance;
    with Variety Feedback Autonomy;
 run;
```

The DATA= option in the PROC CANCORR statement specifies Jobs as the SAS data set to be analyzed. The VPREFIX and WPREFIX options specify the prefixes for naming the canonical variables from the VAR statement and the WITH statement, respectively. The VNAME option specifies 'Satisfaction Areas' to refer to the set of variables from the VAR statement. Similarly, the WNAME option specifies 'Job Characteristics' to refer to the set of variables from the WITH statement.

The VAR statement defines the first of the two sets of variables to be analyzed as Career, Supervisor, and Finance. The WITH statement defines the second set of variables to be Variety, Feedback, and Autonomy. The results of this analysis are displayed in Figure 26.1 to Figure 26.4.

Figure 26.1 displays the canonical correlation, adjusted canonical correlation, approximate standard error, and squared canonical correlation for each pair of canonical variables. The first canonical

correlation (the correlation between the first pair of canonical variables) is 0.9194. This value represents the highest possible correlation between any linear combination of the job satisfaction variables and any linear combination of the job characteristics variables.

Figure 26.1 also lists the likelihood ratio and associated statistics for testing the hypothesis that the canonical correlations in the current row and all that follow are zero.

Figure 26.1 Canonical Correlations, Eigenvalues, and Likelihood Tests

The CANCORR Procedure

Canonical Correlation Analysis

	Canonical Correlation	Adjusted Canonical Correlation	Approximate Standard Error	Squared Canonical Correlation
1	0.919412	0.898444	0.042901	0.845318
2	0.418649	0.276633	0.228740	0.175267
3	0.113366	.	0.273786	0.012852

Eigenvalues of Inv(E)*H
= CanRsq/(1-CanRsq)

	Eigenvalue	Difference	Proportion	Cumulative
1	5.4649	5.2524	0.9604	0.9604
2	0.2125	0.1995	0.0373	0.9977
3	0.0130		0.0023	1.0000

Test of H0: The canonical correlations in the current row and all that follow are zero

	Likelihood Ratio	Approximate F Value	Num DF	Den DF	Pr > F
1	0.12593148	2.93	9	19.621	0.0223
2	0.81413359	0.49	4	18	0.7450
3	0.98714819	0.13	1	10	0.7257

The first approximate F value of 2.93 corresponds to the test that all three canonical correlations are zero. Since the p-value is small (0.0223), you would reject the null hypothesis at the 0.05 level. The second approximate F value of 0.49 corresponds to the test that both the second and the third canonical correlations are zero. Since the p-value is large (0.7450), you would fail to reject the hypothesis and conclude that only the first canonical correlation is significant.

Figure 26.2 lists several multivariate statistics and tests that use approximations based on the F distribution for the null hypothesis that all canonical correlations are zero. Alternatively, you can specify MSTAT=EXACT to compute exact p-values for three of the four tests (Wilks' Lambda, the Hotelling-Lawley Trace, and Roy's greatest root) and an improved F approximation for the fourth (Pillai's Trace). These statistics are described in the section "Multivariate Tests" on page 102 in Chapter 4, "Introduction to Regression Procedures."

Figure 26.2 Multivariate Statistics and F Approximations

Multivariate Statistics and F Approximations

S=3 M=-0.5 N=3

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.12593148	2.93	9	19.621	0.0223
Pillai's Trace	1.03343732	1.75	9	30	0.1204
Hotelling-Lawley Trace	5.69042615	4.76	9	9.8113	0.0119
Roy's Greatest Root	5.46489324	18.22	3	10	0.0002

NOTE: F Statistic for Roy's Greatest Root is an upper bound.

The small p-values for these tests (< 0.05), except for Pillai's trace, suggest rejecting the null hypothesis that all canonical correlations are zero in the population, confirming the results of the preceding likelihood ratio test (Figure 26.1). With only one of the tests resulting in a p-value larger than 0.05, you can assume that the first canonical correlation is significant. The next step is to interpret or identify the two canonical variables corresponding to this significant correlation.

Even though canonical variables are artificial, they can often be "identified" in terms of the original variables. This is done primarily by inspecting the standardized coefficients of the canonical variables and the correlations between the canonical variables and their original variables. Since only the first canonical correlation is significant, only the first pair of canonical variables (Satisfaction1 and Characteristics1) need to be identified.

PROC CANCORR calculates and displays the raw canonical coefficients for the job satisfaction variables and the job characteristic variables. However, since the original variables do not necessarily have equal variance and are not measured in the same units, the raw coefficients must be standardized to allow interpretation. The coefficients are standardized by multiplying the raw coefficients with the standard deviation of the associated variable.

The standardized canonical coefficients in Figure 26.3 show that the first canonical variable for the Satisfaction group is a weighted sum of the variables Supervisor (0.7854) and Career (0.3028), with the emphasis on Supervisor. The coefficient for the variable Finance is near 0. Thus, a person satisfied with his or her supervisor and with a large degree of career satisfaction would score high on the canonical variable Satisfaction1.

Figure 26.3 Standardized Canonical Coefficients from the CANCORR Procedure

Standardized Canonical Coefficients for the Satisfaction Areas

		Satisfaction1	Satisfaction2	Satisfaction3
Career	Career Satisfaction	0.3028	-0.5416	1.0408
Supervisor	Supervisor Satisfaction	0.7854	0.1305	-0.9085
Finance	Financial Satisfaction	0.0538	0.9754	0.3329

Figure 26.3 *continued*

Standardized Canonical Coefficients for the Job Characteristics

		Characteristics1	Characteristics2
Variety	Task Variety	-0.1108	0.8095
Feedback	Amount of Feedback	0.5520	-0.7722
Autonomy	Degree of Autonomy	0.8403	0.1020

Standardized Canonical Coefficients for the Job Characteristics

		Characteristics3
Variety	Task Variety	0.9071
Feedback	Amount of Feedback	0.4194
Autonomy	Degree of Autonomy	-0.8297

The coefficients for the job characteristics variables show that degree of autonomy (Autonomy) and amount of feedback (Feedback) contribute heavily to the Characteristics1 canonical variable (0.8403 and 0.5520, respectively).

Figure 26.4 shows the table of correlations between the canonical variables and the original variables.

Figure 26.4 Canonical Structure Correlations from the CANCORR Procedure

The CANCORR Procedure

Canonical Structure

Correlations Between the Satisfaction Areas and Their Canonical Variables

		Satisfaction1	Satisfaction2	Satisfaction3
Career	Career Satisfaction	0.7499	-0.2503	0.6123
Supervisor	Supervisor Satisfaction	0.9644	0.0362	-0.2618
Finance	Financial Satisfaction	0.2873	0.8814	0.3750

Correlations Between the Job Characteristics and Their Canonical Variables

		Characteristics1	Characteristics2
Variety	Task Variety	0.4863	0.6592
Feedback	Amount of Feedback	0.6216	-0.5452
Autonomy	Degree of Autonomy	0.8459	0.4451

Correlations Between the Job Characteristics and Their Canonical Variables

		Characteristics3
Variety	Task Variety	0.5736
Feedback	Amount of Feedback	0.5625
Autonomy	Degree of Autonomy	-0.2938

Figure 26.4 *continued*

Correlations Between the Satisfaction Areas and the Canonical Variables of the Job Characteristics

		Characteristics1	Characteristics2
Career	Career Satisfaction	0.6895	−0.1048
Supervisor	Supervisor Satisfaction	0.8867	0.0152
Finance	Financial Satisfaction	0.2642	0.3690

Correlations Between the Satisfaction Areas and the Canonical Variables of the Job Characteristics

		Characteristics3
Career	Career Satisfaction	0.0694
Supervisor	Supervisor Satisfaction	−0.0297
Finance	Financial Satisfaction	0.0425

Correlations Between the Job Characteristics and the Canonical Variables of the Satisfaction Areas

		Satisfaction1	Satisfaction2	Satisfaction3
Variety	Task Variety	0.4471	0.2760	0.0650
Feedback	Amount of Feedback	0.5715	−0.2283	0.0638
Autonomy	Degree of Autonomy	0.7777	0.1863	−0.0333

Although these univariate correlations must be interpreted with caution since they do not indicate how the original variables contribute *jointly* to the canonical analysis, they are often useful in the identification of the canonical variables.

Figure 26.4 shows that the supervisor satisfaction variable Supervisor is strongly associated with the Satisfaction1 canonical variable, with a correlation of 0.9644. Slightly less influential is the variable Career, which has a correlation with the canonical variable of 0.7499. Thus, the canonical variable Satisfaction1 seems to represent satisfaction with supervisor and career track.

The correlations for the job characteristics variables show that the canonical variable Characteristics1 seems to represent all three measured variables, with degree of autonomy variable (Autonomy) being the most influential (0.8459).

Hence, you can interpret these results to mean that job characteristics and job satisfaction are related—jobs that possess a high degree of autonomy and level of feedback are associated with workers who are more satisfied with their supervisor and their career. While financial satisfaction is a factor in job satisfaction, it is not as important as the other measured satisfaction-related variables.

Syntax: CANCORR Procedure

The following statements are available in PROC CANCORR:

PROC CANCORR < *options* > ;
  WITH *variables* ;
  BY *variables* ;
  FREQ *variable* ;
  PARTIAL *variables* ;
  VAR *variables* ;
  WEIGHT *variable* ;

The PROC CANCORR statement and the WITH statement are required. The rest of this section provides detailed syntax information for each of the preceding statements, beginning with the PROC CANCORR statement. The remaining statements are covered in alphabetical order.

PROC CANCORR Statement

PROC CANCORR < *options* > ;

The PROC CANCORR statement starts the CANCORR procedure and optionally identifies input and output data sets, specifies the analyses performed, and controls displayed output. Table 26.1 summarizes the options.

Table 26.1 Summary of PROC CANCORR Statement Options

Option	Description
Specify computational details	
EDF=	specifies error degrees of freedom if input observations are regression residuals
MSTAT=	specifies the method of evaluating the multivariate test statistics
NOINT	omits intercept from canonical correlation and regression models
RDF=	specifies regression degrees of freedom if input observations are regression residuals
SINGULAR=	specifies the singularity criterion
Specify input and output data sets	
DATA=	specifies input data set name
OUT=	specifies output data set name
OUTSTAT=	specifies output data set name containing various statistics
Specify labeling options	
PARPREFIX=	specifies a prefix for naming residual variables
VNAME=	specifies a name to refer to VAR statement variables
VPREFIX=	specifies a prefix for naming VAR statement canonical variables

Table 26.1 *continued*

Option	Description
WNAME=	specifies a name to refer to WITH statement variables
WPREFIX=	specifies a prefix for naming WITH statement canonical variables
Control amount of output	
ALL	produces simple statistics, input variable correlations, and canonical redundancy analysis
CORR	produces input variable correlations
NCAN=	specifies number of canonical variables for which full output is desired
NOPRINT	suppresses all displayed output
REDUNDANCY	produces canonical redundancy analysis
SHORT	suppresses default output from canonical analysis
SIMPLE	produces means and standard deviations
Request regression analyses	
VDEP	requests multiple regression analyses with the VAR variables as dependents and the WITH variables as regressors
VREG	requests multiple regression analyses with the VAR variables as regressors and the WITH variables as dependents
WDEP	same as VREG
WREG	same as VDEP
Specify regression statistics	
ALL	produces all regression statistics and includes these statistics in the OUTSTAT= data set
B	produces raw regression coefficients
CLB	produces 95% confidence interval limits for the regression coefficients
CORRB	produces correlations among regression coefficients
INT	requests statistics for the intercept when you specify the B, CLB, SEB, T, or PROBT option
PCORR	displays partial correlations between regressors and dependents
PROBT	displays probability levels for t statistics
SEB	displays standard errors of regression coefficients
SMC	displays squared multiple correlations and F tests
SPCORR	displays semipartial correlations between regressors and dependents
SQPCORR	displays squared partial correlations between regressors and dependents
SQSPCORR	displays squared semipartial correlations between regressors and dependents
TB	displays standardized regression coefficients
T	displays t statistics for regression coefficients

Following are explanations of the options that can be used in the PROC CANCORR statement (in alphabetic order).

ALL

displays simple statistics, correlations among the input variables, the confidence limits for the regression coefficients, and the canonical redundancy analysis. If you specify the VDEP or WDEP option, the ALL option displays all related regression statistics (unless the NOPRINT option is specified) and includes these statistics in the OUTSTAT= data set.

B

produces raw regression coefficients from the regression analyses.

CLB

produces the 95% confidence limits for the regression coefficients from the regression analyses.

CORR

C

produces correlations among the original variables. If you include a PARTIAL statement, the CORR option produces a correlation matrix for all variables in the analysis, the regression statistics (R square, RMSE), the standardized regression coefficients for both the VAR and WITH variables as predicted from the PARTIAL statement variables, and partial correlation matrices.

CORRB

produces correlations among the regression coefficient estimates.

DATA=*SAS-data-set*

names the SAS data set to be analyzed by PROC CANCORR. It can be an ordinary SAS data set or a TYPE=CORR, COV, FACTOR, SSCP, UCORR, or UCOV data set. By default, the procedure uses the most recently created SAS data set.

EDF=*error-df*

specifies the error degrees of freedom if the input observations are residuals from a regression analysis. The effective number of observations is the EDF= value plus one. If you have 100 observations, then specifying EDF=99 has the same effect as omitting the EDF= option.

INT

requests that statistics for the intercept be included when B, CLB, SEB, T, or PROBT is specified for the regression analyses.

MSTAT=FAPPROX | EXACT

specifies the method of evaluating the multivariate test statistics. The default is MSTAT=FAPPROX, which specifies that the multivariate tests are evaluated using the usual approximations based on the F distribution, as discussed in the section "Multivariate Tests" on page 102 in Chapter 4, "Introduction to Regression Procedures." Alternatively, you can specify MSTAT=EXACT to compute exact p-values for three of the four tests (Wilks' lambda, the Hotelling-Lawley trace, and Roy's greatest root) and an improved F approximation for the fourth (Pillai's trace). While MSTAT=EXACT provides better control of the

significance probability for the tests, especially for Roy's greatest root, computations for the exact *p*-values can be appreciably more demanding, and are in fact infeasible for large problems (many dependent variables). Thus, although MSTAT=EXACT is more accurate for most data, it is not the default method.

NCAN=*number*

specifies the number of canonical variables for which full output is desired. The *number* must be less than or equal to the number of canonical variables in the analysis.

The value of the NCAN= option specifies the number of canonical variables for which canonical coefficients and canonical redundancy statistics are displayed, and the number of variables shown in the canonical structure matrices. The NCAN= option does not affect the number of displayed canonical correlations.

If an OUTSTAT= data set is requested, the NCAN= option controls the number of canonical variables for which statistics are output. If an OUT= data set is requested, the NCAN= option controls the number of canonical variables for which scores are output.

NOINT

omits the intercept from the canonical correlation and regression models. Standard deviations, variances, covariances, and correlations are not corrected for the mean. If you use a TYPE=SSCP data set as input to the CANCORR procedure and list the variable Intercept in the VAR or WITH statement, the procedure runs as if you also specified the NOINT option. If you use NOINT and also create an OUTSTAT= data set, the data set is TYPE=UCORR.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 20, "Using the Output Delivery System."

OUT=*SAS-data-set*

creates an output SAS data set to contain all the original data plus scores on the canonical variables. If you want to create a permanent SAS data set, you must specify a two-level name. The OUT= option cannot be used when the DATA= data set is TYPE=CORR, COV, FACTOR, SSCP, UCORR, or UCOV. For details about OUT= data sets, see the section "Output Data Sets" on page 1046. See *SAS Language Reference: Concepts* for more information about permanent SAS data sets.

OUTSTAT=*SAS-data-set*

creates an output SAS data set containing various statistics, including the canonical correlations and coefficients and the multiple regression statistics you request. If you want to create a permanent SAS data set, you must specify a two-level name. For details about OUTSTAT= data sets, see the section "Output Data Sets" on page 1046. See *SAS Language Reference: Concepts* for more information about permanent SAS data sets.

PCORR

produces partial correlations between regressors and dependent variables, removing from each dependent variable and regressor the effects of all other regressors.

PROBT

produces probability levels for the t statistics in the regression analyses.

RDF=*regression-df*

specifies the regression degrees of freedom if the input observations are residuals from a regression analysis. The effective number of observations is the actual number minus the RDF= value. The degrees of freedom for the intercept should not be included in the RDF= option.

REDUNDANCY

RED

produces canonical redundancy statistics.

PARPREFIX=*name*

PPREFIX=*name*

specifies a prefix for naming the residual variables in the OUT= data set and the OUTSTAT= data set. By default, the prefix is R_. The number of characters in the prefix plus the maximum length of the variable names should not exceed the current name length defined by the VALIDVARNAME= system option.

SEB

produces standard errors of the regression coefficients.

SHORT

suppresses all default output from the canonical analysis except the tables of canonical correlations and multivariate statistics.

SIMPLE

S

produces means and standard deviations.

SINGULAR=*p*

SING=*p*

specifies the singularity criterion, where $0 < p < 1$. If a variable in the PARTIAL statement has an R square as large as $1 - p$ (where p is the value of the SINGULAR= option) when predicted from the variables listed before it in the statement, the variable is assigned a standardized regression coefficient of 0, and the SAS log generates a linear dependency warning message. By default, SINGULAR=1E−8.

SMC

produces squared multiple correlations and F tests for the regression analyses.

SPCORR

produces semipartial correlations between regressors and dependent variables, removing from each regressor the effects of all other regressors.

SQPCORR

produces squared partial correlations between regressors and dependent variables, removing from each dependent variable and regressor the effects of all other regressors.

SQSPCORR

produces squared semipartial correlations between regressors and dependent variables, removing from each regressor the effects of all other regressors.

STB

produces standardized regression coefficients.

T

produces *t* statistics for the regression coefficients.

VDEP

WREG

requests multiple regression analyses with the VAR variables as dependent variables and the WITH Variables as regressors.

VNAME=*label*

VN=*label*

specifies a character constant to refer to variables from the VAR statement in the output. Enclose the constant in single or double quotes. If you omit the VNAME= option, these variables are referred to as the VAR variables. The number of characters in the label should not exceed the label length defined by the VALIDVARNAME= system option. For more information about the VALIDVARNAME= system option, see *SAS Language Reference: Dictionary*.

VPREFIX=*name*

VP=*name*

specifies a prefix for naming canonical variables from the VAR statement. By default, these canonical variables are given the names V1, V2, and so on. If you specify VPREFIX=ABC, the names are ABC1, ABC2, and so on. The number of characters in the prefix plus the number of digits required to designate the variables should not exceed the name length defined by the VALIDVARNAME= system option. For more information about the VALIDVARNAME= system option, see *SAS Language Reference: Dictionary*.

WDEP

VREG

requests multiple regression analyses with the WITH variables as dependent variables and the VAR variables as regressors.

WNAME=*label*

WN=*label*

specifies a character constant to refer to variables in the WITH statement in the output. Enclose the constant in single or double quotes. If you omit the WNAME= option, these variables are referred to as the WITH variables. The number of characters in the label should not exceed the label length defined by the VALIDVARNAME= system option. For more information about the VALIDVARNAME= system option, see *SAS Language Reference: Dictionary*.

WPREFIX=*name*

WP=*name*

specifies a prefix for naming canonical variables from the WITH statement. By default, these canonical variables are given the names W1, W2, and so on. If you specify WPREFIX=XYZ, the names are XYZ1, XYZ2, and so on. The number of characters in the prefix plus the number of digits required to designate the variables should not exceed the label length defined by the VALIDVARNAME= system option. For more information about the VALIDVARNAME= system option, see *SAS Language Reference: Dictionary*.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC CANCORR to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC CANCORR. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

FREQ Statement

FREQ *variable* **;**

If one variable in your input data set represents the frequency of occurrence for other values in the observation, specify the variable's name in a FREQ statement. PROC CANCORR then treats the data set as if each observation appeared n times, where n is the value of the FREQ variable for the observation. If the value of the FREQ variable is less than one, the observation is not used in the analysis. Only the integer portion of the value is used. The total number of observations is considered to be equal to the sum of the FREQ variable when PROC CANCORR calculates significance probabilities.

PARTIAL Statement

PARTIAL *variables* **;**

You can use the PARTIAL statement to base the canonical analysis on partial correlations. The variables in the PARTIAL statement are partialed out of the VAR and WITH variables. If you request an OUT= or OUTSTAT= data set, the residual variables are named by prefixing the characters R_ by default or the string specified in the RPREFIX= option to the VAR variables.

VAR Statement

VAR *variables* **;**

The VAR statement lists the variables in the first of the two sets of variables to be analyzed. The variables must be numeric. If you omit the VAR statement, all numeric variables not mentioned in other statements make up the first set of variables. If, however, the DATA= data set is TYPE=SSCP, the default set of variables used as VAR variables does not include the variable Intercept.

WEIGHT Statement

WEIGHT *variable* **;**

If you want to compute weighted product-moment correlation coefficients, specify the name of the weighting variable in a WEIGHT statement. The WEIGHT and FREQ statements have a similar effect, except the WEIGHT statement does not alter the degrees of freedom or number of observations. An observation is used in the analysis only if the WEIGHT variable is greater than zero.

WITH Statement

WITH *variables* **;**

The WITH statement lists the variables in the second set of variables to be analyzed. The variables must be numeric. The WITH statement is required.

Details: CANCORR Procedure

Missing Values

If an observation has a missing value for any of the variables in the analysis, that observation is omitted from the analysis.

Formulas

Assume without loss of generality that the two sets of variables, $\mathbf{X}$ with p variables and $\mathbf{Y}$ with q variables, have means of zero. Let n be the number of observations, and let m be $n-1$.

Note that the scales of eigenvectors and canonical coefficients are arbitrary. PROC CANCORR follows the usual procedure of rescaling the canonical coefficients so that each canonical variable has a variance of one.

There are several different sets of formulas that can be used to compute the canonical correlations, $\rho_i, i = 1, \ldots, \min(p, q)$, and unscaled canonical coefficients:

1. Let $\mathbf{S}_{XX} = \mathbf{X}'\mathbf{X}/m$ be the covariance matrix of $\mathbf{X}$, $\mathbf{S}_{YY} = \mathbf{Y}'\mathbf{Y}/m$ be the covariance matrix of $\mathbf{Y}$, and $\mathbf{S}_{XY} = \mathbf{X}'\mathbf{Y}/m$ be the covariance matrix between $\mathbf{X}$ and $\mathbf{Y}$. Then the eigenvalues of $\mathbf{S}_{YY}^{-1}\mathbf{S}_{XY}'\mathbf{S}_{XX}^{-1}\mathbf{S}_{XY}$ are the squared canonical correlations, and the right eigenvectors are raw canonical coefficients for the $\mathbf{Y}$ variables. The eigenvalues of $\mathbf{S}_{XX}^{-1}\mathbf{S}_{XY}\mathbf{S}_{YY}^{-1}\mathbf{S}_{XY}'$ are the squared canonical correlations, and the right eigenvectors are raw canonical coefficients for the $\mathbf{X}$ variables.

2. Let $\mathbf{T} = \mathbf{Y}'\mathbf{Y}$ and $\mathbf{H} = \mathbf{Y}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$. The eigenvalues ξ_i of $\mathbf{T}^{-1}\mathbf{H}$ are the squared canonical correlations, ρ_i^2, and the right eigenvectors are raw canonical coefficients for the $\mathbf{Y}$ variables. Interchange $\mathbf{X}$ and $\mathbf{Y}$ in the preceding formulas, and the eigenvalues remain the same, but the right eigenvectors are raw canonical coefficients for the $\mathbf{X}$ variables.

3. Let $\mathbf{E} = \mathbf{T} - \mathbf{H}$. The eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$ are $\lambda_i = \rho_i^2/(1-\rho_i^2)$. The right eigenvectors of $\mathbf{E}^{-1}\mathbf{H}$ are the same as the right eigenvectors of $\mathbf{T}^{-1}\mathbf{H}$.

4. Canonical correlation can be viewed as a principal component analysis of the predicted values of one set of variables from a regression on the other set of variables, in the metric of the error covariance matrix. For example, regress the $\mathbf{Y}$ variables on the $\mathbf{X}$ variables. Call the predicted values $\mathbf{P} = \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{Y}$ and the residuals $\mathbf{R} = \mathbf{Y} - \mathbf{P} = (\mathbf{I} - \mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}')\mathbf{Y}$. The error covariance matrix is $\mathbf{R}'\mathbf{R}/m$. Choose a transformation $\mathbf{Q}$ that converts the error covariance matrix to an identity—that is, $(\mathbf{RQ})'(\mathbf{RQ}) = \mathbf{Q}'\mathbf{R}'\mathbf{RQ} = m\mathbf{I}$. Apply the same transformation to the predicted values to yield, say, $\mathbf{Z} = \mathbf{PQ}$. Now do a principal component analysis on the covariance matrix of $\mathbf{Z}$, and you get the eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$. Repeat with $\mathbf{X}$ and $\mathbf{Y}$ variables interchanged, and you get the same eigenvalues.

   To show this relationship between canonical correlation and principal components, note that $\mathbf{P}'\mathbf{P} = \mathbf{H}$, $\mathbf{R}'\mathbf{R} = \mathbf{E}$, and $\mathbf{QQ}' = m\mathbf{E}^{-1}$. Let the covariance matrix of $\mathbf{Z}$ be $\mathbf{G}$. Then $\mathbf{G} = \mathbf{Z}'\mathbf{Z}/m = (\mathbf{PQ})'\mathbf{PQ}/m = \mathbf{Q}'\mathbf{P}'\mathbf{PQ}/m = \mathbf{Q}'\mathbf{HQ}/m$. Let $\mathbf{u}$ be an eigenvector of $\mathbf{G}$ and κ be the corresponding eigenvalue. Then by definition, $\mathbf{Gu} = \kappa\mathbf{u}$; hence $\mathbf{Q}'\mathbf{HQu}/m = \kappa\mathbf{u}$. Premultiplying both sides by $\mathbf{Q}$ yields $\mathbf{QQ}'\mathbf{HQu}/m = \kappa\mathbf{Qu}$ and thus $\mathbf{E}^{-1}\mathbf{HQu} = \kappa\mathbf{Qu}$. Hence $\mathbf{Qu}$ is an eigenvector of $\mathbf{E}^{-1}\mathbf{H}$ and κ is also an eigenvalue of $\mathbf{E}^{-1}\mathbf{H}$.

5. If the covariance matrices are replaced by correlation matrices, the preceding formulas yield standardized canonical coefficients instead of raw canonical coefficients.

The formulas for multivariate test statistics are shown in the section "Multivariate Tests" on page 102 in Chapter 4, "Introduction to Regression Procedures." Formulas for linear regression are provided in other sections of that chapter.

Output Data Sets

OUT= Data Set

The OUT= data set contains all the variables in the original data set plus new variables containing the canonical variable scores. The number of new variables is twice that specified by the NCAN= option. The names of the new variables are formed by concatenating the values given by the VPREFIX= and WPREFIX= options (the defaults are V and W) with the numbers 1, 2, 3, and so on. The new variables have mean 0 and variance equal to 1. An OUT= data set cannot be created if the DATA= data set is TYPE=CORR, COV, FACTOR, SSCP, UCORR, or UCOV.

If you use a PARTIAL statement, the OUT= data set also contains the residuals from predicting the VAR variables from the PARTIAL variables. The names of the residual variables are formed by concatenating the values given by the PARPREFIX= option (the default is R_) with the numbers 1, 2, 3, and so on.

OUTSTAT= Data Set

The OUTSTAT= data set is similar to the TYPE=CORR or TYPE=UCORR data set produced by the CORR procedure, but it contains several results in addition to those produced by PROC CORR.

The new data set contains the following variables:

- the BY variables, if any
- two new character variables, _TYPE_ and _NAME_
- Intercept, if the INT option is used
- the variables analyzed (those in the VAR statement and the WITH statement)

Each observation in the new data set contains some type of statistic as indicated by the _TYPE_ variable. The values of the _TYPE_ variable are as follows.

TYPE	
MEAN	means
STD	standard deviations
USTD	uncorrected standard deviations. When you specify the NOINT option in the PROC CANCORR statement, the OUTSTAT= data set contains standard deviations not corrected for the mean (_TYPE_='USTD').

N	number of observations on which the analysis is based. This value is the same for each variable.
SUMWGT	sum of the weights if a WEIGHT statement is used. This value is the same for each variable.
CORR	correlations. The _NAME_ variable contains the name of the variable corresponding to each row of the correlation matrix.
UCORR	uncorrected correlation matrix. When you specify the NOINT option in the PROC CANCORR statement, the OUTSTAT= data set contains a matrix of correlations not corrected for the means.
CORRB	correlations among the regression coefficient estimates
STB	standardized regression coefficients. The _NAME_ variable contains the name of the dependent variable.
B	raw regression coefficients
SEB	standard errors of the regression coefficients
LCLB	95% lower confidence limits for the regression coefficients
UCLB	95% upper confidence limits for the regression coefficients
T	t statistics for the regression coefficients
PROBT	probability levels for the t statistics
SPCORR	semipartial correlations between regressors and dependent variables
SQSPCORR	squared semipartial correlations between regressors and dependent variables
PCORR	partial correlations between regressors and dependent variables
SQPCORR	squared partial correlations between regressors and dependent variables
RSQUARED	R squares for the multiple regression analyses
ADJRSQ	adjusted R squares
LCLRSQ	approximate 95% lower confidence limits for the R squares
UCLRSQ	approximate 95% upper confidence limits for the R squares
F	F statistics for the multiple regression analyses
PROBF	probability levels for the F statistics
CANCORR	canonical correlations
SCORE	standardized canonical coefficients. The _NAME_ variable contains the name of the canonical variable. To obtain the canonical variable scores, these coefficients should be multiplied by the standardized data, using means obtained from the observation with _TYPE_='MEAN' and standard deviations obtained from the observation with _TYPE_='STD'.
RAWSCORE	raw canonical coefficients. To obtain the canonical variable scores, these coefficients should be multiplied by the raw data centered by means obtained from the observation with _TYPE_='MEAN'.

USCORE	scoring coefficients to be applied without subtracting the mean from the raw variables. These are standardized canonical coefficients computed under a NOINT model. To obtain the canonical variable scores, these coefficients should be multiplied by the data that are standardized by the uncorrected standard deviations obtained from the observation with _TYPE_='USTD'.
STRUCTUR	canonical structure.

Computational Resources

Notation

Let

$$
\begin{aligned}
n &= \text{number of observations} \\
v &= \text{number of variables} \\
w &= \text{number of WITH variables} \\
p &= \max(v, w) \\
q &= \min(v, w) \\
b &= v + w \\
t &= \text{total number of variables (VAR, WITH, and PARTIAL)}
\end{aligned}
$$

Time Requirements

The time required to compute the correlation matrix is roughly proportional to

$$n(p+q)^2$$

The time required for the canonical analysis is roughly proportional to

$$\frac{1}{6}p^3 + p^2q + \frac{3}{2}pq^2 + 5q^3$$

but the coefficient for q^3 varies depending on the number of QR iterations in the singular value decomposition.

Memory Requirements

The minimum memory required is approximately

$$4(v^2 + w^2 + t^2)$$

bytes. Additional memory is required if you request the VDEP or WDEP option.

Displayed Output

If the SIMPLE option is specified, PROC CANCORR produces means and standard deviations for each input variable. If the CORR option is specified, PROC CANCORR produces correlations among the input variables. Unless the NOPRINT option is specified, PROC CANCORR displays a table of canonical correlations containing the following:

- Canonical Correlations. These are always nonnegative.
- Adjusted Canonical Correlations (Lawley 1959), which are asymptotically less biased than the raw correlations and can be negative. The adjusted canonical correlations might not be computable, and they are displayed as missing values if two canonical correlations are nearly equal or if some are close to zero. A missing value is also displayed if an adjusted canonical correlation is larger than a previous adjusted canonical correlation.
- Approx Standard Errors, which are the approximate standard errors of the canonical correlations
- Squared Canonical Correlations
- Eigenvalues of INV(E)*H, which are equal to CanRsq/(1−CanRsq), where CanRsq is the corresponding squared canonical correlation. Also displayed for each eigenvalue is the Difference from the next eigenvalue, the Proportion of the sum of the eigenvalues, and the Cumulative proportion.
- Likelihood Ratio for the hypothesis that the current canonical correlation and all smaller ones are zero in the population. The likelihood ratio for all canonical correlations equals Wilks' lambda.
- Approx F statistic based on Rao's approximation to the distribution of the likelihood ratio (Rao 1973, p. 556; Kshirsagar 1972, p. 326)
- Num DF and Den DF (numerator and denominator degrees of freedom) and Pr $> F$ (probability level) associated with the F statistic

Unless you specify the NOPRINT option, PROC CANCORR produces a table of multivariate statistics for the null hypothesis that all canonical correlations are zero in the population. These statistics, as described in the section "Multivariate Tests" on page 102 in Chapter 4, "Introduction to Regression Procedures," are as follows:

- Wilks' lambda
- Pillai's trace
- Hotelling-Lawley trace
- Roy's greatest root

For each of the preceding statistics, PROC CANCORR displays the following, depending on the specification of the MSTAT= option.

If you specify MSTAT=FAPPROX (also the default value), the following statistics are displayed:

- an F approximation or upper bound
- Num DF, the numerator degrees of freedom
- Den DF, the denominator degrees of freedom
- Pr $> F$, the probability level

If you specify MSTAT=EXACT, the following statistic is displayed:

- a t value

Unless you specify the SHORT or NOPRINT option, PROC CANCORR displays the following:

- both Raw (unstandardized) and Standardized Canonical Coefficients normalized to give canonical variables with unit variance. Standardized coefficients can be used to compute canonical variable scores from the standardized (zero mean and unit variance) input variables. Raw coefficients can be used to compute canonical variable scores from the input variables without standardizing them.
- all four Canonical Structure matrices, giving Correlations Between the canonical variables and the original variables

If you specify the REDUNDANCY option, PROC CANCORR displays the following:

- the Canonical Redundancy Analysis (Stewart and Love 1968; Cooley and Lohnes 1971), including Raw (unstandardized) and Standardized Variance and Cumulative Proportion of the Variance of each set of variables Explained by Their Own Canonical Variables and Explained by The Opposite Canonical Variables
- the Squared Multiple Correlations of each variable with the first m canonical variables of the opposite set, where m varies from 1 to the number of canonical correlations

If you specify the VDEP option, PROC CANCORR performs multiple regression analyses with the VAR variables as dependent variables and the WITH variables as regressors. If you specify the WDEP option, PROC CANCORR performs multiple regression analyses with the WITH variables as dependent variables and the VAR variables as regressors. If you specify the VDEP or WDEP option and also specify the ALL option, PROC CANCORR displays the following items. You can also specify individual options to request a subset of the output generated by the ALL option; or you can suppress the output by specifying the NOPRINT option.

SMC	Squared Multiple Correlations and F Tests. For each regression model, identified by its dependent variable name, PROC CANCORR displays the R square, Adjusted R square (Wherry 1931), F Statistic, and Pr $> F$. Also for each regression model, PROC CANCORR displays an Approximate 95% Confidence Interval for the population R square (Helland 1987). These confidence limits are valid only when the regressors are random and when the regressors and dependent variables are approximately distributed according to a multivariate normal distribution. The average R squares for the models considered, unweighted and weighted by variance, are also given.
CORRB	Correlations Among the Regression Coefficient Estimates
STB	Standardized Regression Coefficients
B	Raw Regression Coefficients
SEB	Standard Errors of the Regression Coefficients
CLB	95% confidence limits for the regression coefficients
T	T Statistics for the Regression Coefficients
PROBT	Probability > |T| for the Regression Coefficients
SPCORR	Semipartial Correlations between regressors and dependent variables, Removing from Each Regressor the Effects of All Other Regressors
SQSPCORR	Squared Semipartial Correlations between regressors and dependent variables, Removing from Each Regressor the Effects of All Other Regressors
PCORR	Partial Correlations between regressors and dependent variables, Removing the Effects of All Other Regressors from Both Regressor and Criterion
SQPCORR	Squared Partial Correlations between regressors and dependent variables, Removing the Effects of All Other Regressors from Both Regressor and Criterion

ODS Table Names

PROC CANCORR assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 26.2.

For more information about ODS, see Chapter 20, "Using the Output Delivery System."

All of the tables in Table 26.2 are created with the specification of the PROC CANCORR statement; a few tables need an additional PARTIAL statement.

Table 26.2 ODS Tables Produced by PROC CANCORR

ODS Table Name	Description	Statement and Option
AvgRSquare	Average R squares (weighted and unweighted)	VDEP, WDEP, SMC, or ALL
CanCorr	Canonical correlations	default
CanStructureVCan	Correlations between the VAR canonical variables and the VAR and WITH variables	default (if SHORT is not specified)
CanStructureWCan	Correlations between the WITH canonical variables and the WITH and VAR variables	default (if SHORT is not specified)
ConfidenceLimits	95% confidence limits for the regression coefficients	VDEP, WDEP, CLB, or ALL
Corr	Correlations among the original variables	CORR or ALL
CorrOnPartial	Partial correlations	PARTIAL statement with CORR or ALL
CorrRegCoefEst	Correlations among the regression coefficient estimates	VDEP, WDEP, CORRB, or ALL
MultStat	Multivariate statistics	default
NObsNVar	Number of observations and variables	SIMPLE or ALL
ParCorr	Partial correlations	VDEP, WDEP, PCORR, or ALL
ProbtRegCoef	Prob > |t| for the regression coefficients	VDEP, WDEP, PROBT, or ALL
RawCanCoefV	Raw canonical coefficients for the VAR variables	default (if SHORT is not specified)
RawCanCoefW	Raw canonical coefficients for the WITH variables	default (if SHORT is not specified)
RawRegCoef	Raw regression coefficients	VDEP, WDEP, B, or ALL
Redundancy	Canonical redundancy analysis	REDUNDANCY or ALL
Regression	Squared multiple correlations and F tests	VDEP, WDEP, SMC, or ALL
RSquareRMSEOnPartial	R squares and RMSEs on PARTIAL variables	PARTIAL statement with CORR or ALL
SemiParCorr	Semipartial correlations	VDEP, WDEP, SPCORR, or ALL
SimpleStatistics	Simple statistics	SIMPLE or ALL
SqMultCorr	Canonical redundancy analysis: squared multiple correlations	REDUNDANCY or ALL
SqParCorr	Squared partial correlations	VDEP, WDEP, SQPCORR, or ALL
SqSemiParCorr	Squared semipartial correlations	VDEP, WDEP, SQSPCORR, or ALL
StdCanCoefV	Standardized canonical coefficients for the VAR variables	default (if SHORT is not specified)

Table 26.2 *continued*

ODS Table Name	Description	Statement and Option
StdCanCoefW	Standardized canonical coefficients for the WITH variables	default (if SHORT is not specified)
StdErrRawRegCoef	Standard errors of the raw regression coefficients	VDEP, WDEP, SEB, or ALL
StdRegCoef	Standardized regression coefficients	VDEP, WDEP, STB, or ALL
StdRegCoefOnPartial	Standardized regression coefficients on PARTIAL variables	PARTIAL statement with CORR or ALL
tValueRegCoef	*t* values for the regression coefficients	VDEP, WDEP, T, or ALL

Example: CANCORR Procedure

Example 26.1: Canonical Correlation Analysis of Fitness Club Data

Three physiological and three exercise variables are measured on 20 middle-aged men in a fitness club. You can use the CANCORR procedure to determine whether the physiological variables are related in any way to the exercise variables. The following statements create the SAS data set Fit and produce Output 26.1.1 through Output 26.1.5:

```
data Fit;
   input Weight Waist Pulse Chins Situps Jumps;
   datalines;
191  36  50   5  162   60
189  37  52   2  110   60
193  38  58  12  101  101
162  35  62  12  105   37
189  35  46  13  155   58
182  36  56   4  101   42
211  38  56   8  101   38
167  34  60   6  125   40
176  31  74  15  200   40
154  33  56  17  251  250
169  34  50  17  120   38
166  33  52  13  210  115
154  34  64  14  215  105
247  46  50   1   50   50
193  36  46   6   70   31
202  37  62  12  210  120
176  37  54   4   60   25
157  32  52  11  230   80
156  33  54  15  225   73
138  33  68   2  110   43
;
```

```
proc cancorr data=Fit all
     vprefix=Physiological vname='Physiological Measurements'
     wprefix=Exercises wname='Exercises';
   var Weight Waist Pulse;
   with Chins Situps Jumps;
   title 'Middle-Aged Men in a Health Fitness Club';
   title2 'Data Courtesy of Dr. A. C. Linnerud, NC State Univ';
run;
```

Output 26.1.1 Correlations among the Original Variables

Middle-Aged Men in a Health Fitness Club
Data Courtesy of Dr. A. C. Linnerud, NC State Univ

The CANCORR Procedure

Correlations Among the Original Variables

Correlations Among the Physiological Measurements

	Weight	Waist	Pulse
Weight	1.0000	0.8702	-0.3658
Waist	0.8702	1.0000	-0.3529
Pulse	-0.3658	-0.3529	1.0000

Correlations Among the Exercises

	Chins	Situps	Jumps
Chins	1.0000	0.6957	0.4958
Situps	0.6957	1.0000	0.6692
Jumps	0.4958	0.6692	1.0000

Correlations Between the Physiological Measurements and the Exercises

	Chins	Situps	Jumps
Weight	-0.3897	-0.4931	-0.2263
Waist	-0.5522	-0.6456	-0.1915
Pulse	0.1506	0.2250	0.0349

Output 26.1.1 displays the correlations among the original variables. The correlations between the physiological and exercise variables are moderate, the largest being -0.6456 between Waist and Situps. There are larger within-set correlations: 0.8702 between Weight and Waist, 0.6957 between Chins and Situps, and 0.6692 between Situps and Jumps.

Output 26.1.2 Canonical Correlations and Multivariate Statistics

Middle-Aged Men in a Health Fitness Club
Data Courtesy of Dr. A. C. Linnerud, NC State Univ

The CANCORR Procedure

Canonical Correlation Analysis

	Canonical Correlation	Adjusted Canonical Correlation	Approximate Standard Error	Squared Canonical Correlation
1	0.795608	0.754056	0.084197	0.632992
2	0.200556	-.076399	0.220188	0.040223
3	0.072570	.	0.228208	0.005266

Eigenvalues of Inv(E)*H
= CanRsq/(1-CanRsq)

	Eigenvalue	Difference	Proportion	Cumulative
1	1.7247	1.6828	0.9734	0.9734
2	0.0419	0.0366	0.0237	0.9970
3	0.0053		0.0030	1.0000

Test of H0: The canonical correlations in the
current row and all that follow are zero

	Likelihood Ratio	Approximate F Value	Num DF	Den DF	Pr > F
1	0.35039053	2.05	9	34.223	0.0635
2	0.95472266	0.18	4	30	0.9491
3	0.99473355	0.08	1	16	0.7748

Multivariate Statistics and F Approximations

S=3 M=-0.5 N=6

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.35039053	2.05	9	34.223	0.0635
Pillai's Trace	0.67848151	1.56	9	48	0.1551
Hotelling-Lawley Trace	1.77194146	2.64	9	19.053	0.0357
Roy's Greatest Root	1.72473874	9.20	3	16	0.0009

NOTE: F Statistic for Roy's Greatest Root is an upper bound.

As Output 26.1.2 shows, the first canonical correlation is 0.7956, which would appear to be substantially larger than any of the between-set correlations. The probability level for the null hypothesis that all the canonical correlations are zero in the population is only 0.0635, so no firm conclusions can be drawn. The remaining canonical correlations are not worthy of consideration, as can be seen from the probability levels and especially from the negative adjusted canonical correlations.

Because the variables are not measured in the same units, the standardized coefficients rather than the raw coefficients should be interpreted. The correlations given in the canonical structure matrices should also be examined.

Output 26.1.3 Raw and Standardized Canonical Coefficients

Raw Canonical Coefficients for the Physiological Measurements

	Physiological1	Physiological2	Physiological3
Weight	-0.031404688	-0.076319506	-0.007735047
Waist	0.4932416756	0.3687229894	0.1580336471
Pulse	-0.008199315	-0.032051994	0.1457322421

Raw Canonical Coefficients for the Exercises

	Exercises1	Exercises2	Exercises3
Chins	-0.066113986	-0.071041211	-0.245275347
Situps	-0.016846231	0.0019737454	0.0197676373
Jumps	0.0139715689	0.0207141063	-0.008167472

Standardized Canonical Coefficients for the Physiological Measurements

	Physiological1	Physiological2	Physiological3
Weight	-0.7754	-1.8844	-0.1910
Waist	1.5793	1.1806	0.5060
Pulse	-0.0591	-0.2311	1.0508

Standardized Canonical Coefficients for the Exercises

	Exercises1	Exercises2	Exercises3
Chins	-0.3495	-0.3755	-1.2966
Situps	-1.0540	0.1235	1.2368
Jumps	0.7164	1.0622	-0.4188

The first canonical variable for the physiological variables, displayed in Output 26.1.3, is a weighted difference of Waist (1.5793) and Weight (−0.7754), with more emphasis on Waist. The coefficient for Pulse is near 0. The correlations between Waist and Weight and the first canonical variable are both positive, 0.9254 for Waist and 0.6206 for Weight. Weight is therefore a suppressor variable, meaning that its coefficient and its correlation have opposite signs.

The first canonical variable for the exercise variables also shows a mixture of signs, subtracting Situps (−1.0540) and Chins (−0.3495) from Jumps (0.7164), with the most weight on Situps. All the correlations are negative, indicating that Jumps is also a suppressor variable.

It might seem contradictory that a variable should have a coefficient of opposite sign from that of its correlation with the canonical variable. In order to understand how this can happen, consider a simplified situation: predicting Situps from Waist and Weight by multiple regression. In informal terms, it seems plausible that obese people should do fewer sit-ups than skinny people. Assume that the men in the sample do not vary much in height, so there is a strong correlation between Waist and Weight (0.8702). Examine the relationships between obesity and the independent variables:

- People with large waists tend to be more obese than people with small waists. Hence, the correlation between Waist and Situps should be negative.
- People with high weights tend to be more obese than people with low weights. Therefore, Weight should correlate negatively with Situps.
- For a fixed value of Weight, people with large waists tend to be shorter and more obese. Thus, the multiple regression coefficient for Waist should be negative.
- For a fixed value of Waist, people with higher weights tend to be taller and skinnier. The multiple regression coefficient for Weight should therefore be positive, of opposite sign from the correlation between Weight and Situps.

Therefore, the general interpretation of the first canonical correlation is that Weight and Jumps act as suppressor variables to enhance the correlation between Waist and Situps. This canonical correlation might be strong enough to be of practical interest, but the sample size is not large enough to draw definite conclusions.

The canonical redundancy analysis (Output 26.1.4) shows that neither of the first pair of canonical variables is a good overall predictor of the opposite set of variables, the proportions of variance explained being 0.2854 and 0.2584. The second and third canonical variables add virtually nothing, with cumulative proportions for all three canonical variables being 0.2969 and 0.2767.

Output 26.1.4 Canonical Redundancy Analysis

Middle-Aged Men in a Health Fitness Club
Data Courtesy of Dr. A. C. Linnerud, NC State Univ

The CANCORR Procedure

Canonical Redundancy Analysis

Standardized Variance of the Physiological Measurements Explained by

Canonical Variable Number	Their Own Canonical Variables: Proportion	Their Own Canonical Variables: Cumulative Proportion	Canonical R-Square	The Opposite Canonical Variables: Proportion	The Opposite Canonical Variables: Cumulative Proportion
1	0.4508	0.4508	0.6330	0.2854	0.2854
2	0.2470	0.6978	0.0402	0.0099	0.2953
3	0.3022	1.0000	0.0053	0.0016	0.2969

Standardized Variance of the Exercises Explained by

Canonical Variable Number	Their Own Canonical Variables: Proportion	Their Own Canonical Variables: Cumulative Proportion	Canonical R-Square	The Opposite Canonical Variables: Proportion	The Opposite Canonical Variables: Cumulative Proportion
1	0.4081	0.4081	0.6330	0.2584	0.2584
2	0.4345	0.8426	0.0402	0.0175	0.2758
3	0.1574	1.0000	0.0053	0.0008	0.2767

The squared multiple correlations (Output 26.1.5) indicate that the first canonical variable of the physiological measurements has some predictive power for Chins (0.3351) and Situps (0.4233) but almost none for Jumps (0.0167). The first canonical variable of the exercises is a fairly good predictor of Waist (0.5421), a poorer predictor of Weight (0.2438), and nearly useless for predicting Pulse (0.0701).

Output 26.1.5 Canonical Redundancy Analysis

Squared Multiple Correlations Between the Physiological Measurements and the First M Canonical Variables of the Exercises

M	1	2	3
Weight	0.2438	0.2678	0.2679
Waist	0.5421	0.5478	0.5478
Pulse	0.0701	0.0702	0.0749

Squared Multiple Correlations Between the Exercises and the First M Canonical Variables of the Physiological Measurements

M	1	2	3
Chins	0.3351	0.3374	0.3396
Situps	0.4233	0.4365	0.4365
Jumps	0.0167	0.0536	0.0539

References

Cooley, W. W. and Lohnes, P. R. (1971), *Multivariate Data Analysis*, New York: John Wiley & Sons.

Fisher, R. A. (1938), *Statistical Methods for Research Workers*, Tenth Edition, Edinburgh: Oliver & Boyd.

Hanson, R. J. and Norris, M. J. (1981), "Analysis of Measurements Based on the Singular Value Decomposition," *SIAM Journal of Scientific and Statistical Computing*, 2, 363–373.

Helland, I. S. (1987), "On the Interpretation and Use of R^2 in Regression Analysis," *Biometrics*, 43, 61–69.

Hotelling, H. (1935), "The Most Predictable Criterion," *Journal of Educational Psychology*, 26, 139–142.

Hotelling, H. (1936), "Relations between Two Sets of Variables," *Biometrika*, 28, 321–377.

Kshirsagar, A. M. (1972), *Multivariate Analysis*, New York: Marcel Dekker.

Lawley, D. N. (1959), "Tests of Significance in Canonical Analysis," *Biometrika*, 46, 59–66.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979), *Multivariate Analysis*, London: Academic Press.

Mulaik, S. A. (1972), *The Foundations of Factor Analysis*, New York: McGraw-Hill.

Rao, C. R. (1964), "The Use and Interpretation of Principal Component Analysis in Applied Research," *Sankhya A*, 26, 329–358.

Rao, C. R. (1973), *Linear Statistical Inference*, New York: John Wiley & Sons.

Stewart, D. K. and Love, W. A. (1968), "A General Canonical Correlation Index," *Psychological Bulletin*, 70, 160–163.

Tatsuoka, M. M. (1971), *Multivariate Analysis*, New York: John Wiley & Sons.

Thompson, B. (1984), "Canonical Correlation Analysis," Sage University Paper series in Quantitative Applications in the Social Sciences, 07-047, Beverly Hills and London: Sage Publications.

Timm, N. H. (1975), *Multivariate Analysis*, Monterey, CA: Brooks-Cole.

van den Wollenberg, A. L. (1977), "Redundancy Analysis—An Alternative to Canonical Correlation Analysis," *Psychometrika*, 42, 207–219.

Wherry, R. J. (1931), "A New Formula for Predicting the Shrinkage of the Coefficient of Multiple Correlation," *Annals of Mathematical Statistics*, 2, 440–457.

Chapter 27

The CANDISC Procedure

Contents

Overview: CANDISC Procedure **1061**
Getting Started: CANDISC Procedure **1063**
Syntax: CANDISC Procedure **1068**
PROC CANDISC Statement 1069
BY Statement 1072
CLASS Statement 1072
FREQ Statement 1073
VAR Statement 1073
WEIGHT Statement 1073
Details: CANDISC Procedure **1074**
Missing Values 1074
Computational Details 1074
Input Data Set 1075
Output Data Sets 1076
Computational Resources 1078
Displayed Output 1079
ODS Table Names 1081
Example: CANDISC Procedure **1082**
Example 27.1: Analysis of Iris Data With PROC CANDISC 1082
References **1089**

Overview: CANDISC Procedure

Canonical discriminant analysis is a dimension-reduction technique related to principal component analysis and canonical correlation. The methodology used in deriving the canonical coefficients parallels that of a one-way MANOVA. MANOVA tests for equality of the mean vector across class levels. Canonical discriminant analysis finds linear combinations of the quantitative variables that provide maximal separation between classes or groups. Given a classification variable and several quantitative variables, the CANDISC procedure derives *canonical variables*, linear combinations of the quantitative variables that summarize between-class variation in much the same way that principal components summarize total variation.

The CANDISC procedure performs a canonical discriminant analysis, computes squared Mahalanobis distances between class means, and performs both univariate and multivariate one-way analyses of variance. Two output data sets can be produced: one containing the canonical coefficients and another containing, among other things, scored canonical variables. The canonical coefficients output data set can be rotated by the FACTOR procedure. It is customary to standardize the canonical coefficients so that the canonical variables have means that are equal to zero and pooled within-class variances that are equal to one. PROC CANDISC displays both standardized and unstandardized canonical coefficients. Correlations between the canonical variables and the original variables as well as the class means for the canonical variables are also displayed; these correlations, sometimes known as loadings, are called canonical structures. The scored canonical variables output data set can be used in conjunction with ODS Graphics to plot pairs of canonical variables in order to aid visual interpretation of group differences.

Given two or more groups of observations with measurements on several quantitative variables, canonical discriminant analysis derives a linear combination of the variables that has the highest possible multiple correlation with the groups. This maximal multiple correlation is called the *first canonical correlation*. The coefficients of the linear combination are the *canonical coefficients* or *canonical weights*. The variable defined by the linear combination is the *first canonical variable* or *canonical component*. The second canonical correlation is obtained by finding the linear combination uncorrelated with the first canonical variable that has the highest possible multiple correlation with the groups. The process of extracting canonical variables can be repeated until the number of canonical variables equals the number of original variables or the number of classes minus one, whichever is smaller.

The first canonical correlation is at least as large as the multiple correlation between the groups and any of the original variables. If the original variables have high within-group correlations, the first canonical correlation can be large even if all the multiple correlations are small. In other words, the first canonical variable can show substantial differences between the classes, even if none of the original variables do. Canonical variables are sometimes called *discriminant functions*, but this usage is ambiguous because the DISCRIM procedure produces very different functions for classification that are also called discriminant functions.

For each canonical correlation, PROC CANDISC tests the hypothesis that it and all smaller canonical correlations are zero in the population. An F approximation (Rao 1973; Kshirsagar 1972) is used that gives better small-sample results than the usual chi-square approximation. The variables should have an approximate multivariate normal distribution within each class, with a common covariance matrix in order for the probability levels to be valid.

Canonical discriminant analysis is equivalent to canonical correlation analysis between the quantitative variables and a set of dummy variables coded from the class variable. Canonical discriminant analysis is also equivalent to performing the following steps:

1. Transform the variables so that the pooled within-class covariance matrix is an identity matrix.
2. Compute class means on the transformed variables.
3. Perform a principal component analysis on the means, weighting each mean by the number of observations in the class. The eigenvalues are equal to the ratio of between-class variation to within-class variation in the direction of each principal component.

4. Back-transform the principal components into the space of the original variables, obtaining the canonical variables.

An interesting property of the canonical variables is that they are uncorrelated whether the correlation is calculated from the total sample or from the pooled within-class correlations. The canonical coefficients are not orthogonal, however, so the canonical variables do not represent perpendicular directions through the space of the original variables.

Getting Started: CANDISC Procedure

The data in this example are measurements of 159 fish caught in Finland's lake Laengelmavesi; this data set is available from the Puranen. For each of the For each of the seven species (bream, roach, whitefish, parkki, perch, pike, and smelt) the weight, length, height, and width of each fish are tallied. Three different length measurements are recorded: from the nose of the fish to the beginning of its tail, from the nose to the notch of its tail, and from the nose to the end of its tail. The height and width are recorded as percentages of the third length variable. The following statements create the SAS data set:

```
title 'Fish Measurement Data';

proc format;
   value specfmt
      1='Bream'
      2='Roach'
      3='Whitefish'
      4='Parkki'
      5='Perch'
      6='Pike'
      7='Smelt';
run;

data fish (drop=HtPct WidthPct);
   input Species Weight Length1 Length2 Length3 HtPct
         WidthPct @@;
   Height=HtPct*Length3/100;
   Width=WidthPct*Length3/100;
   format Species specfmt.;
   symbol = put(Species, specfmt2.);
   datalines;
1  242.0 23.2 25.4 30.0 38.4 13.4 1  290.0 24.0 26.3 31.2 40.0 13.8

... more lines ...

7   19.7 13.2 14.3 15.2 18.9 13.6 7   19.9 13.8 15.0 16.2 18.1 11.6
;
```

The next steps use PROC CANDISC to find the three canonical variables that best separate the species of fish in the fish data and create the output data set outcan. With the NCAN=3 option, only the first three canonical variables are displayed. The ODS EXCLUDE statement is specified to exclude the canonical structure tables and most of the canonical coefficient tables in order to obtain a more compact set of results. The TEMPLATE and SGRENDER procedures within ODS Graphics are invoked to create a plot of the first two canonical variables. The following statements produce Figure 27.1 through Figure 27.6:

```
proc candisc data=fish ncan=3 out=outcan;
   ods exclude tstruc bstruc pstruc tcoef pcoef;
   class Species;
   var Weight Length1 Length2 Length3 Height Width;
run;

proc template;
   define statgraph scatter;
      begingraph;
         entrytitle 'Fish Measurement Data';
         layout overlayequated / equatetype=fit
            xaxisopts=(label='Canonical Variable 1')
            yaxisopts=(label='Canonical Variable 2');
            scatterplot x=Can1 y=Can2 / group=species name='fish';
            layout gridded / autoalign=(topright);
               discretelegend 'fish' / border=false opaque=false;
               endlayout;
         endlayout;
      endgraph;
   end;
run;

proc sgrender data=outcan template=scatter;
run;
```

PROC CANDISC begins by displaying summary information about the variables in the analysis. This information includes the number of observations, the number of quantitative variables in the analysis (specified with the VAR statement), and the number of classes in the classification variable (specified with the CLASS statement). The frequency of each class is also displayed.

Figure 27.1 Summary Information

```
                          Fish Measurement Data

                          The CANDISC Procedure

      Total Sample Size      158          DF Total                 157
      Variables                6          DF Within Classes        151
      Classes                  7          DF Between Classes         6

                Number of Observations Read           159
                Number of Observations Used           158
```

Figure 27.1 *continued*

Class Level Information

Species	Variable Name	Frequency	Weight	Proportion
Bream	Bream	34	34.0000	0.215190
Parkki	Parkki	11	11.0000	0.069620
Perch	Perch	56	56.0000	0.354430
Pike	Pike	17	17.0000	0.107595
Roach	Roach	20	20.0000	0.126582
Smelt	Smelt	14	14.0000	0.088608
Whitefish	Whitefish	6	6.0000	0.037975

PROC CANDISC performs a multivariate one-way analysis of variance (one-way MANOVA) and provides four multivariate tests of the hypothesis that the class mean vectors are equal. These tests, shown in Figure 27.2, indicate that not all of the mean vectors are equal ($p < .0001$).

Figure 27.2 MANOVA and Multivariate Tests

Fish Measurement Data

The CANDISC Procedure

Multivariate Statistics and F Approximations

S=6 M=-0.5 N=72

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.00036325	90.71	36	643.89	<.0001
Pillai's Trace	3.10465132	26.99	36	906	<.0001
Hotelling-Lawley Trace	52.05799676	209.24	36	413.64	<.0001
Roy's Greatest Root	39.13499776	984.90	6	151	<.0001

NOTE: F Statistic for Roy's Greatest Root is an upper bound.

The first canonical correlation is the greatest possible multiple correlation with the classes that can be achieved by using a linear combination of the quantitative variables. The first canonical correlation, displayed in Figure 27.3, is 0.987463. A likelihood ratio test is displayed of the hypothesis that the current canonical correlation and all smaller ones are zero. The first line is equivalent to Wilks' lambda multivariate test.

Figure 27.3 Canonical Correlations

Fish Measurement Data

The CANDISC Procedure

	Canonical Correlation	Adjusted Canonical Correlation	Approximate Standard Error	Squared Canonical Correlation
1	0.987463	0.986671	0.001989	0.975084
2	0.952349	0.950095	0.007425	0.906969
3	0.838637	0.832518	0.023678	0.703313
4	0.633094	0.623649	0.047821	0.400809
5	0.344157	0.334170	0.070356	0.118444
6	0.005701	.	0.079806	0.000033

Eigenvalues of Inv(E)*H
= CanRsq/(1-CanRsq)

	Eigenvalue	Difference	Proportion	Cumulative
1	39.1350	29.3859	0.7518	0.7518
2	9.7491	7.3786	0.1873	0.9390
3	2.3706	1.7016	0.0455	0.9846
4	0.6689	0.5346	0.0128	0.9974
5	0.1344	0.1343	0.0026	1.0000
6	0.0000		0.0000	1.0000

Test of H0: The canonical correlations in the current row and all that follow are zero

	Likelihood Ratio	Approximate F Value	Num DF	Den DF	Pr > F
1	0.00036325	90.71	36	643.89	<.0001
2	0.01457896	46.46	25	547.58	<.0001
3	0.15671134	23.61	16	452.79	<.0001
4	0.52820347	12.09	9	362.78	<.0001
5	0.88152702	4.88	4	300	0.0008
6	0.99996749	0.00	1	151	0.9442

The first canonical variable, Can1, shows that the linear combination of the centered variables $\text{Can1} = -0.0006\times\text{Weight} - 0.33\times\text{Length1} - 2.49\times\text{Length2} + 2.60\times\text{Length3} + 1.12\times\text{Height} - 1.45\times\text{Width}$ separates the species most effectively (see Figure 27.4).

Figure 27.4 Raw Canonical Coefficients

Fish Measurement Data

The CANDISC Procedure

Raw Canonical Coefficients

Variable	Can1	Can2	Can3
Weight	-0.000648508	-0.005231659	-0.005596192
Length1	-0.329435762	-0.626598051	-2.934324102
Length2	-2.486133674	-0.690253987	4.045038893
Length3	2.595648437	1.803175454	-1.139264914
Height	1.121983854	-0.714749340	0.283202557
Width	-1.446386704	-0.907025481	0.741486686

PROC CANDISC computes the means of the canonical variables for each class. The first canonical variable is the linear combination of the variables Weight, Length1, Length2, Length3, Height, and Width that provides the greatest difference (in terms of a univariate F test) between the class means. The second canonical variable provides the greatest difference between class means while being uncorrelated with the first canonical variable.

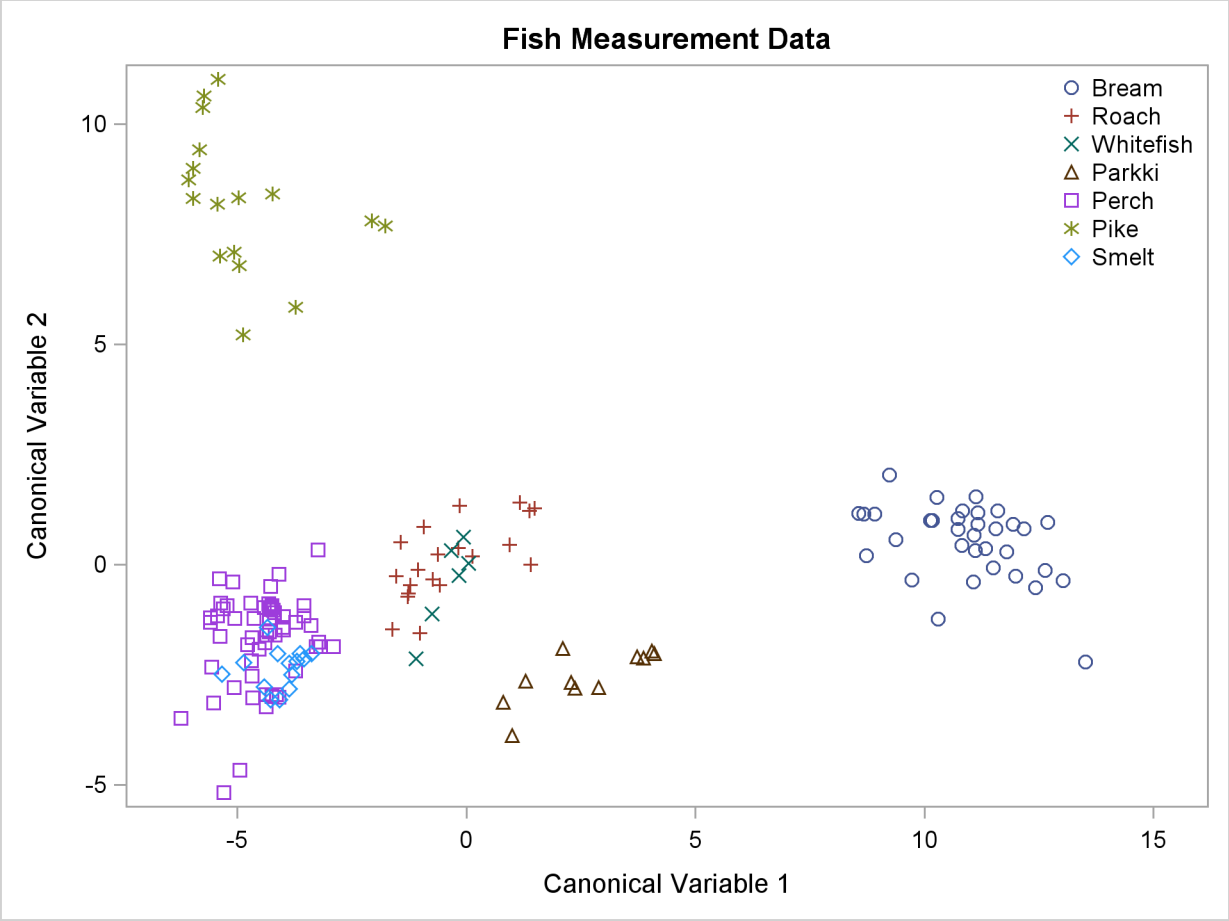
Figure 27.5 Class Means for Canonical Variables

Class Means on Canonical Variables

Species	Can1	Can2	Can3
Bream	10.94142464	0.52078394	0.23496708
Parkki	2.58903743	-2.54722416	-0.49326158
Perch	-4.47181389	-1.70822715	1.29281314
Pike	-4.89689441	8.22140791	-0.16469132
Roach	-0.35837149	0.08733611	-1.10056438
Smelt	-4.09136653	-2.35805841	-4.03836098
Whitefish	-0.39541755	-0.42071778	1.06459242

A plot of the first two canonical variables (Figure 27.6) shows that Can1 discriminates between three groups: 1) bream; 2) whitefish, roach, and parkki; and 3) smelt, pike, and perch. Can2 best discriminates between pike and the other species.

Figure 27.6 Plot of First Two Canonical Variables



Syntax: CANDISC Procedure

The following statements are available in PROC CANDISC:

PROC CANDISC < *options* > ;
  CLASS *variable* ;
  BY *variables* ;
  FREQ *variable* ;
  VAR *variables* ;
  WEIGHT *variable* ;

The BY, CLASS, FREQ, VAR, and WEIGHT statements are described after the PROC CANDISC statement.

PROC CANDISC Statement

PROC CANDISC < *options* > **;**

The PROC CANDISC statement invokes the CANDISC procedure. The options listed in Table 27.1 are available in the PROC CANDISC statement.

Table 27.1 CANDISC Procedure Options

Option	Description
Input Data Set	
DATA=	specifies input SAS data set
Output Data Sets	
OUT=	specifies output data set with canonical scores
OUTSTAT=	specifies output statistics data set
Method Details	
NCAN=	specifies the number of canonical variables
PREFIX=	specifies a prefix for naming the canonical variables
SINGULAR=	specifies the singularity criterion
Control Displayed Output	
ALL	displays all output
ANOVA	displays univariate statistics
BCORR	displays between correlations
BCOV	displays between covariances
BSSCP	displays between SSCPs
DISTANCE	displays squared Mahalanobis distances
NOPRINT	suppresses all displayed output
PCORR	displays pooled correlations
PCOV	displays pooled covariances
PSSCP	displays pooled SSCPs
SHORT	suppresses some displayed output
SIMPLE	displays simple descriptive statistics
STDMEAN	displays standardized class means
TCORR	displays total correlations
TCOV	displays total covariances
TSSCP	displays total SSCPs
WCORR	displays within correlations
WCOV	displays within covariances
WSSCP	displays within SSCPs

ALL
activates all of the display options.

ANOVA
displays univariate statistics for testing the hypothesis that the class means are equal in the population for each variable.

BCORR
displays between-class correlations.

BCOV
displays between-class covariances. The between-class covariance matrix equals the between-class SSCP matrix divided by $n(c-1)/c$, where n is the number of observations and c is the number of classes. The between-class covariances should be interpreted in comparison with the total-sample and within-class covariances, not as formal estimates of population parameters.

BSSCP
displays the between-class SSCP matrix.

DATA=*SAS-data-set*
specifies the data set to be analyzed. The data set can be an ordinary SAS data set or one of several specially structured data sets created by SAS statistical procedures. These specially structured data sets include TYPE=CORR, COV, CSSCP, and SSCP. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

DISTANCE
MAHALANOBIS
displays squared Mahalanobis distances between the group means, F statistics, and the corresponding probabilities of greater squared Mahalanobis distances between the group means.

NCAN=*n*
specifies the number of canonical variables to be computed. The value of n must be less than or equal to the number of variables. If you specify NCAN=0, the procedure displays the canonical correlations, but not the canonical coefficients, structures, or means. A negative value suppresses the canonical analysis entirely. Let v be the number of variables in the VAR statement, and let c be the number of classes. If you omit the NCAN= option, only $\min(v, c-1)$ canonical variables are generated; if you also specify an OUT= output data set, v canonical variables are generated, and the last $v-(c-1)$ canonical variables have missing values.

NOPRINT
suppresses the normal display of results. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

OUT=*SAS-data-set*
creates an output SAS data set containing the original data and the canonical variable scores. To create a permanent SAS data set, specify a two-level name (see *SAS Language Reference: Concepts*), for more information about permanent SAS data sets).

OUTSTAT=*SAS-data-set*

creates a TYPE=CORR output SAS data set that contains various statistics, including class means, standard deviations, correlations, canonical correlations, canonical structures, canonical coefficients, and means of canonical variables for each class. To create a permanent SAS data set, specify a two-level name (see *SAS Language Reference: Concepts*, for more information about permanent SAS data sets).

PCORR

displays pooled within-class correlations (partial correlations based on the pooled within-class covariances).

PCOV

displays pooled within-class covariances.

PREFIX=*name*

specifies a prefix for naming the canonical variables. By default the names are Can1, Can2, Can3, and so forth. If you specify PREFIX=Abc, the components are named Abc1, Abc2, and so on. The number of characters in the prefix plus the number of digits required to designate the canonical variables should not exceed 32. The prefix is truncated if the combined length exceeds 32.

PSSCP

displays the pooled within-class corrected SSCP matrix.

SHORT

suppresses the display of canonical structures, canonical coefficients, and class means on canonical variables; only tables of canonical correlations and multivariate test statistics are displayed.

SIMPLE

displays simple descriptive statistics for the total sample and within each class.

SINGULAR=*p*

specifies the criterion for determining the singularity of the total-sample correlation matrix and the pooled within-class covariance matrix, where $0 < p < 1$. The default is SINGULAR=1E–8.

Let **S** be the total-sample correlation matrix. If the R square for predicting a quantitative variable in the VAR statement from the variables preceding it exceeds $1 - p$, then **S** is considered singular. If **S** is singular, the probability levels for the multivariate test statistics and canonical correlations are adjusted for the number of variables with R square exceeding $1 - p$.

If **S** is considered singular and the inverse of **S** (squared Mahalanobis distances) is required, a quasi inverse is used instead. For details see the section "Quasi-inverse" on page 1408 in Chapter 31, "The DISCRIM Procedure."

STDMEAN

displays total-sample and pooled within-class standardized class means.

TCORR

displays total-sample correlations.

TCOV
displays total-sample covariances.

TSSCP
displays the total-sample corrected SSCP matrix.

WCORR
displays within-class correlations for each class level.

WCOV
displays within-class covariances for each class level.

WSSCP
displays the within-class corrected SSCP matrix for each class level.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC CANDISC to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC CANDISC. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variable* **;**

The values of the CLASS variable define the groups for analysis. Class levels are determined by the formatted values of the CLASS variable. The CLASS variable can be numeric or character. A CLASS statement is required.

FREQ Statement

FREQ *variable* **;**

If a variable in the data set represents the frequency of occurrence for the other values in the observation, include the name of the variable in a FREQ statement. The procedure then treats the data set as if each observation appears n times, where n is the value of the FREQ variable for the observation. The total number of observations is considered to be equal to the sum of the FREQ variable when the procedure determines degrees of freedom for significance probabilities.

If the value of the FREQ variable is missing or is less than one, the observation is not used in the analysis. If the value is not an integer, the value is truncated to an integer.

VAR Statement

VAR *variables* **;**

You specify the quantitative variables to include in the analysis by using a VAR statement. If you do not use a VAR statement, the analysis includes all numeric variables not listed in other statements.

WEIGHT Statement

WEIGHT *variable* **;**

To use relative weights for each observation in the input data set, place the weights in a variable in the data set and specify the name in a WEIGHT statement. This is often done when the variance associated with each observation is different and the values of the WEIGHT variable are proportional to the reciprocals of the variances. If the value of the WEIGHT variable is missing or is less than zero, then a value of zero for the weight is assumed.

The WEIGHT and FREQ statements have a similar effect except that the WEIGHT statement does not alter the degrees of freedom.

Details: CANDISC Procedure

Missing Values

If an observation has a missing value for any of the quantitative variables, it is omitted from the analysis. If an observation has a missing CLASS value but is otherwise complete, it is not used in computing the canonical correlations and coefficients; however, canonical variable scores are computed for that observation for the OUT= data set.

Computational Details

General Formulas

Canonical discriminant analysis is equivalent to canonical correlation analysis between the quantitative variables and a set of dummy variables coded from the class variable. In the following notation the dummy variables are denoted by $\mathbf{y}$ and the quantitative variables by $\mathbf{x}$. The total sample covariance matrix for the $\mathbf{x}$ and $\mathbf{y}$ variables is

$$\mathbf{S} = \begin{bmatrix} \mathbf{S}_{xx} & \mathbf{S}_{xy} \\ \mathbf{S}_{yx} & \mathbf{S}_{yy} \end{bmatrix}$$

When c is the number of groups, n_t is the number of observations in group t, and $\mathbf{S}_t$ is the sample covariance matrix for the $\mathbf{x}$ variables in group t, the within-class pooled covariance matrix for the $\mathbf{x}$ variables is

$$\mathbf{S}_p = \frac{1}{\sum n_t - c} \sum (n_t - 1)\mathbf{S}_t$$

The canonical correlations, ρ_i, are the square roots of the eigenvalues, λ_i, of the following matrix. The corresponding eigenvectors are $\mathbf{v}_i$.

$$\mathbf{S}_p{}^{-1/2}\mathbf{S}_{xy}\mathbf{S}_{yy}{}^{-1}\mathbf{S}_{yx}\mathbf{S}_p{}^{-1/2}$$

Let $\mathbf{V}$ be the matrix with the eigenvectors $\mathbf{v}_i$ that correspond to nonzero eigenvalues as columns. The raw canonical coefficients are calculated as follows:

$$\mathbf{R} = \mathbf{S}_p{}^{-1/2}\mathbf{V}$$

The pooled within-class standardized canonical coefficients are

$$\mathbf{P} = \mathrm{diag}(\mathbf{S}_p)^{1/2}\mathbf{R}$$

The total sample standardized canonical coefficients are

$$\mathbf{T} = \mathrm{diag}(\mathbf{S}_{xx})^{1/2}\mathbf{R}$$

Let $\mathbf{X}_c$ be the matrix with the centered **x** variables as columns. The canonical scores can be calculated by any of the following:

$$\mathbf{X}_c\,\mathbf{R}$$

$$\mathbf{X}_c\,\mathrm{diag}(\mathbf{S}_p)^{-1/2}\mathbf{P}$$

$$\mathbf{X}_c\,\mathrm{diag}(\mathbf{S}_{xx})^{-1/2}\mathbf{T}$$

For the multivariate tests based on $\mathbf{E}^{-1}\mathbf{H}$,

$$\mathbf{E} = (n-1)(\mathbf{S}_{yy} - \mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy})$$

$$\mathbf{H} = (n-1)\mathbf{S}_{yx}\mathbf{S}_{xx}^{-1}\mathbf{S}_{xy}$$

where n is the total number of observations.

Input Data Set

The input DATA= data set can be an ordinary SAS data set or one of several specially structured data sets created by statistical procedures available with SAS/STAT software. For more information about special types of data sets, see Appendix A, "Special SAS Data Sets." The BY variable in these data sets becomes the CLASS variable in PROC CANDISC. These specially structured data sets include the following:

- TYPE=CORR data sets created by PROC CORR by using a BY statement
- TYPE=COV data sets created by PROC PRINCOMP by using both the COV option and a BY statement
- TYPE=CSSCP data sets created by PROC CORR by using the CSSCP option and a BY statement, where the OUT= data set is assigned TYPE=CSSCP with the TYPE= data set option
- TYPE=SSCP data sets created by PROC REG by using both the OUTSSCP= option and a BY statement.

When the input data set is TYPE=CORR, TYPE=COV, or TYPE=CSSCP, then PROC CANDISC reads the number of observations for each class from the observations with _TYPE_='N' and the variable means in each class from the observations with _TYPE_='MEAN'. The CANDISC procedure then reads the within-class correlations from the observations with _TYPE_='CORR', the standard deviations from the observations with _TYPE_='STD' (data set TYPE=CORR), the within-class covariances from the observations with _TYPE_='COV' (data set TYPE=COV), or the within-class corrected sums of squares and crossproducts from the observations with _TYPE_='CSSCP' (data set TYPE=CSSCP).

When the data set does not include any observations with _TYPE_='CORR' (data set TYPE=CORR), _TYPE_='COV' (data set TYPE=COV), or _TYPE_='CSSCP' (data set TYPE=CSSCP) for each class, PROC CANDISC reads the pooled within-class information from the data set. In this case, PROC CANDISC reads the pooled within-class correlations from the observations with _TYPE_='PCORR', the pooled within-class standard deviations from the observations with _TYPE_='PSTD' (data set TYPE=CORR), the pooled within-class covariances from the observations with _TYPE_='PCOV' (data set TYPE=COV), or the pooled within-class corrected SSCP matrix from the observations with_TYPE_='PSSCP' (data set TYPE=CSSCP).

When the input data set is TYPE=SSCP, then PROC CANDISC reads the number of observations for each class from the observations with _TYPE_='N', the sum of weights of observations from the variable INTERCEPT in observations with _TYPE_='SSCP' and _NAME_='INTERCEPT', the variable sums from the variable=*variablenames* in observations with _TYPE_='SSCP' and _NAME_='INTERCEPT', and the uncorrected sums of squares and crossproducts from the variable=*variablenames* in observations with _TYPE_='SSCP' and _NAME_=*variablenames*.

Output Data Sets

OUT= Data Set

The OUT= data set contains all the variables in the original data set plus new variables containing the canonical variable scores. You determine the number of new variables by using the NCAN= option. The names of the new variables are formed as described in the PREFIX= option. The new variables have means equal to zero and pooled within-class variances equal to one. An OUT= data set cannot be created if the DATA= data set is not an ordinary SAS data set.

OUTSTAT= Data Set

The OUTSTAT= data set is similar to the TYPE=CORR data set produced by the CORR procedure but contains many results in addition to those produced by the CORR procedure.

The OUTSTAT= data set is TYPE=CORR, and it contains the following variables:

- the BY variables, if any
- the CLASS variable
- _TYPE_, a character variable of length 8 that identifies the type of statistic
- _NAME_, a character variable of length 32 that identifies the row of the matrix or the name of the canonical variable
- the quantitative variables (those in the VAR statement, or if there is no VAR statement, all numeric variables not listed in any other statement)

The observations, as identified by the variable _TYPE_, have the following _TYPE_ values:

TYPE	**Contents**
N	number of observations both for the total sample (CLASS variable missing) and within each class (CLASS variable present)
SUMWGT	sum of weights both for the total sample (CLASS variable missing) and within each class (CLASS variable present) if a WEIGHT statement is specified
MEAN	means both for the total sample (CLASS variable missing) and within each class (CLASS variable present)
STDMEAN	total-standardized class means
PSTDMEAN	pooled within-class standardized class means
STD	standard deviations both for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PSTD	pooled within-class standard deviations
BSTD	between-class standard deviations
RSQUARED	univariate R squares

The following kinds of observations are identified by the combination of the variables _TYPE_ and _NAME_. When the _TYPE_ variable has one of the following values, the _NAME_ variable identifies the row of the matrix:

TYPE	**Contents**
CSSCP	corrected SSCP matrix for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PSSCP	pooled within-class corrected SSCP matrix
BSSCP	between-class SSCP matrix
COV	covariance matrix for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PCOV	pooled within-class covariance matrix
BCOV	between-class covariance matrix
CORR	correlation matrix for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PCORR	pooled within-class correlation matrix
BCORR	between-class correlation matrix

When the _TYPE_ variable has one of the following values, the _NAME_ variable identifies the canonical variable:

TYPE	**Contents**
CANCORR	canonical correlations
STRUCTUR	canonical structure

BSTRUCT	between canonical structure
PSTRUCT	pooled within-class canonical structure
SCORE	total sample standardized canonical coefficients
PSCORE	pooled within-class standardized canonical coefficients
RAWSCORE	raw canonical coefficients
CANMEAN	means of the canonical variables for each class

You can use this data set with PROC SCORE to get scores on the canonical variables for new data by using one of the following forms:

```
* The CLASS variable C is numeric;
proc score data=NewData score=Coef(where=(c = .  )) out=Scores; run;

* The CLASS variable C is character;
proc score data=NewData score=Coef(where=(c = ' ')) out=Scores;
run;
```

The WHERE clause is used to exclude the within-class means and standard deviations. PROC SCORE standardizes the new data by subtracting the original variable means that are stored in the _TYPE_='MEAN' observations, and dividing by the original variable standard deviations from the _TYPE_='STD' observations. Then PROC SCORE multiplies the standardized variables by the coefficients from the _TYPE_='SCORE' observations to get the canonical scores.

Computational Resources

In the following discussion, let

n = number of observations
c = number of class levels
v = number of variables in the VAR list
l = length of the CLASS variable

Memory Requirements

The amount of memory in bytes for temporary storage needed to process the data is

$$c(4v^2 + 28v + 4l + 68) + 16v^2 + 96v + 4l$$

With the ANOVA option, the temporary storage must be increased by 16v bytes. The DISTANCE option requires an additional temporary storage of $4v^2 + 4v$ bytes.

Time Requirements

The following factors determine the time requirements of the CANDISC procedure:

- The time needed for reading the data and computing covariance matrices is proportional to nv^2. PROC CANDISC must also look up each class level in the list. This is faster if the data are sorted by the CLASS variable. The time for looking up class levels is proportional to a value ranging from n to $n \log(c)$.
- The time for inverting a covariance matrix is proportional to v^3.
- The time required for the canonical discriminant analysis is proportional to v^3.

Each of the preceding factors has a different constant of proportionality.

Displayed Output

The displayed output from PROC CANDISC includes the class level information table. For each level of the classification variable, the following information is provided: the output data set variable name, frequency sum, weight sum, and the proportion of the total sample.

The optional output from PROC CANDISC includes the following:

- Within-class SSCP matrices for each group
- Pooled within-class SSCP matrix
- Between-class SSCP matrix
- Total-sample SSCP matrix
- Within-class covariance matrices for each group
- Pooled within-class covariance matrix
- Between-class covariance matrix, equal to the between-class SSCP matrix divided by $n(c-1)/c$, where n is the number of observations and c is the number of classes
- Total-sample covariance matrix
- Within-class correlation coefficients and $\Pr > |r|$ to test the hypothesis that the within-class population correlation coefficients are zero
- Pooled within-class correlation coefficients and $\Pr > |r|$ to test the hypothesis that the partial population correlation coefficients are zero
- Between-class correlation coefficients and $\Pr > |r|$ to test the hypothesis that the between-class population correlation coefficients are zero

- Total-sample correlation coefficients and Pr $> |r|$ to test the hypothesis that the total population correlation coefficients are zero
- Simple statistics, including N (the number of observations), sum, mean, variance, and standard deviation both for the total sample and within each class
- Total-sample standardized class means, obtained by subtracting the grand mean from each class mean and dividing by the total sample standard deviation
- Pooled within-class standardized class means, obtained by subtracting the grand mean from each class mean and dividing by the pooled within-class standard deviation
- Pairwise squared distances between groups
- Univariate test statistics, including total-sample standard deviations, pooled within-class standard deviations, between-class standard deviations, R square, $R^2/(1-R^2)$, F, and Pr $> F$ (univariate F values and probability levels for one-way analyses of variance)

By default, PROC CANDISC displays these statistics:

- Multivariate statistics and F approximations including Wilks' lambda, Pillai's trace, Hotelling-Lawley trace, and Roy's greatest root with F approximations, numerator and denominator degrees of freedom (Num DF and Den DF), and probability values (Pr $> F$). Each of these four multivariate statistics tests the hypothesis that the class means are equal in the population. See the section "Multivariate Tests" on page 102 in Chapter 4, "Introduction to Regression Procedures," for more information.
- Canonical correlations
- Adjusted canonical correlations (Lawley 1959). These are asymptotically less biased than the raw correlations and can be negative. The adjusted canonical correlations might not be computable and are displayed as missing values if two canonical correlations are nearly equal or if some are close to zero. A missing value is also displayed if an adjusted canonical correlation is larger than a previous adjusted canonical correlation.
- Approximate standard error of the canonical correlations
- Squared canonical correlations
- Eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$. Each eigenvalue is equal to $\rho^2/(1-\rho^2)$, where ρ^2 is the corresponding squared canonical correlation and can be interpreted as the ratio of between-class variation to pooled within-class variation for the corresponding canonical variable. The table includes Eigenvalues, Differences between successive eigenvalues, the Proportion of the sum of the eigenvalues, and the Cumulative proportion.
- Likelihood ratio for the hypothesis that the current canonical correlation and all smaller ones are zero in the population. The likelihood ratio for the hypothesis that all canonical correlations equal zero is Wilks' lambda.
- Approx F statistic based on Rao's approximation to the distribution of the likelihood ratio (Rao 1973, p. 556; Kshirsagar 1972, p. 326)

- Numerator degrees of freedom (Num DF), denominator degrees of freedom (Den DF), and Pr $>$ F, the probability level associated with the F statistic

The following statistics can be suppressed with the SHORT option:

- Total canonical structure, giving total-sample correlations between the canonical variables and the original variables
- Between canonical structure, giving between-class correlations between the canonical variables and the original variables
- Pooled within canonical structure, giving pooled within-class correlations between the canonical variables and the original variables
- Total-sample standardized canonical coefficients, standardized to give canonical variables with zero mean and unit pooled within-class variance when applied to the total-sample standardized variables
- Pooled within-class standardized canonical coefficients, standardized to give canonical variables with zero mean and unit pooled within-class variance when applied to the pooled within-class standardized variables
- Raw canonical coefficients, standardized to give canonical variables with zero mean and unit pooled within-class variance when applied to the centered variables
- Class means on the canonical variables

ODS Table Names

PROC CANDISC assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 27.2. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 27.2 ODS Tables Produced by PROC CANDISC

ODS Table Name	Description	PROC CANDISC Option
ANOVA	Univariate statistics	ANOVA
AveRSquare	Average R square	ANOVA
BCorr	Between-class correlations	BCORR
BCov	Between-class covariances	BCOV
BSSCP	Between-class SSCP matrix	BSSCP
BStruc	Between canonical structure	default
CanCorr	Canonical correlations	default
CanonicalMeans	Class means on canonical variables	default
Counts	Number of observations, variables, classes, df	default

Table 27.2 *continued*

ODS Table Name	Description	PROC CANDISC Option
CovDF	DF for covariance matrices, not printed	any *COV option
Dist	Squared distances	DISTANCE
DistFValues	F statistics based on squared distances	DISTANCE
DistProb	Probabilities for F statistics from squared distances	DISTANCE
Levels	Class level information	default
MultStat	MANOVA	default
NObs	Number of observations	default
PCoef	Pooled standard canonical coefficients	default
PCorr	Pooled within-class correlations	PCORR
PCov	Pooled within-class covariances	PCOV
PSSCP	Pooled within-class SSCP matrix	PSSCP
PStdMeans	Pooled standardized class means	STDMEAN
PStruc	Pooled within canonical structure	default
RCoef	Raw canonical coefficients	default
SimpleStatistics	Simple statistics	SIMPLE
TCoef	Total-sample standard canonical coefficients	default
TCorr	Total-sample correlations	TCORR
TCov	Total-sample covariances	TCOV
TSSCP	Total-sample SSCP matrix	TSSCP
TStdMeans	Total standardized class means	STDMEAN
TStruc	Total canonical structure	default
WCorr	Within-class correlations	WCORR
WCov	Within-class covariances	WCOV
WSSCP	Within-class SSCP matrices	WSSCP

Example: CANDISC Procedure

Example 27.1: Analysis of Iris Data With PROC CANDISC

The iris data published by Fisher (1936) have been widely used for examples in discriminant analysis and cluster analysis. The sepal length, sepal width, petal length, and petal width are measured in millimeters on 50 iris specimens from each of three species: *Iris setosa*, *I. versicolor*, and *I. virginica*.

This example is a canonical discriminant analysis that creates an output data set containing scores on the canonical variables and plots the canonical variables. The following statements produce Output 27.1.1 through Output 27.1.6:

```
title 'Fisher (1936) Iris Data';

proc format;
   value specname
      1='Setosa    '
      2='Versicolor'
      3='Virginica ';
run;

data iris;
   input SepalLength SepalWidth PetalLength PetalWidth
         Species @@;
   format Species specname.;
   label SepalLength='Sepal Length in mm.'
         SepalWidth ='Sepal Width in mm.'
         PetalLength='Petal Length in mm.'
         PetalWidth ='Petal Width in mm.';
   symbol = put(Species, specname10.);
   datalines;
50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3
63 28 51 15 3 46 34 14 03 1 69 31 51 23 3 62 22 45 15 2

... more lines ...

63 33 60 25 3 53 37 15 02 1
;
proc candisc data=iris out=outcan distance anova;
   class Species;
   var SepalLength SepalWidth PetalLength PetalWidth;
run;
```

PROC CANDISC first displays information about the observations and the classes in the data set in Output 27.1.1.

Output 27.1.1 Iris Data: Summary Information

```
                        Fisher (1936) Iris Data

                         The CANDISC Procedure

      Total Sample Size      150          DF Total               149
      Variables                4          DF Within Classes      147
      Classes                  3          DF Between Classes       2

                Number of Observations Read            150
                Number of Observations Used            150
```

Output 27.1.1 *continued*

Class Level Information

Species	Variable Name	Frequency	Weight	Proportion
Setosa	Setosa	50	50.0000	0.333333
Versicolor	Versicolor	50	50.0000	0.333333
Virginica	Virginica	50	50.0000	0.333333

The DISTANCE option in the PROC CANDISC statement displays squared Mahalanobis distances between class means. Results from the DISTANCE option are shown in Output 27.1.2.

Output 27.1.2 Iris Data: Squared Mahalanobis Distances and Distance Statistics

Fisher (1936) Iris Data

The CANDISC Procedure

Squared Distance to Species

From Species	Setosa	Versicolor	Virginica
Setosa	0	89.86419	179.38471
Versicolor	89.86419	0	17.20107
Virginica	179.38471	17.20107	0

F Statistics, NDF=4, DDF=144 for Squared Distance to Species

From Species	Setosa	Versicolor	Virginica
Setosa	0	550.18889	1098
Versicolor	550.18889	0	105.31265
Virginica	1098	105.31265	0

Prob > Mahalanobis Distance for Squared Distance to Species

From Species	Setosa	Versicolor	Virginica
Setosa	1.0000	<.0001	<.0001
Versicolor	<.0001	1.0000	<.0001
Virginica	<.0001	<.0001	1.0000

The ANOVA option uses univariate statistics to test the hypothesis that the class means are equal. The resulting R-square values (see Output 27.1.3) range from 0.4008 for SepalWidth to 0.9414 for PetalLength, and each variable is significant at the 0.0001 level. The multivariate test for differences between the classes (which is displayed by default) is also significant at the 0.0001 level; you would expect this from the highly significant univariate test results.

Output 27.1.3 Iris Data: Univariate and Multivariate Statistics

Fisher (1936) Iris Data

The CANDISC Procedure

Univariate Test Statistics

F Statistics, Num DF=2, Den DF=147

Variable	Label	Total Standard Deviation	Pooled Standard Deviation	Between Standard Deviation	R-Square	R-Square / (1-RSq)	F Value	Pr > F
Sepal Length	Sepal Length in mm.	8.2807	5.1479	7.9506	0.6187	1.6226	119.26	<.0001
Sepal Width	Sepal Width in mm.	4.3587	3.3969	3.3682	0.4008	0.6688	49.16	<.0001
Petal Length	Petal Length in mm.	17.6530	4.3033	20.9070	0.9414	16.0566	1180.16	<.0001
Petal Width	Petal Width in mm.	7.6224	2.0465	8.9673	0.9289	13.0613	960.01	<.0001

Average R-Square

Unweighted	0.7224358
Weighted by Variance	0.8689444

Multivariate Statistics and F Approximations

S=2 M=0.5 N=71

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.02343863	199.15	8	288	<.0001
Pillai's Trace	1.19189883	53.47	8	290	<.0001
Hotelling-Lawley Trace	32.47732024	582.20	8	203.4	<.0001
Roy's Greatest Root	32.19192920	1166.96	4	145	<.0001

NOTE: F Statistic for Roy's Greatest Root is an upper bound.
NOTE: F Statistic for Wilks' Lambda is exact.

The R square between Can1 and the class variable, 0.969872, is much larger than the corresponding R square for Can2, 0.222027. This is displayed in Output 27.1.4.

Output 27.1.4 Iris Data: Canonical Correlations and Eigenvalues

Fisher (1936) Iris Data

The CANDISC Procedure

	Canonical Correlation	Adjusted Canonical Correlation	Approximate Standard Error	Squared Canonical Correlation
1	0.984821	0.984508	0.002468	0.969872
2	0.471197	0.461445	0.063734	0.222027

Eigenvalues of Inv(E)*H
= CanRsq/(1-CanRsq)

	Eigenvalue	Difference	Proportion	Cumulative
1	32.1919	31.9065	0.9912	0.9912
2	0.2854		0.0088	1.0000

Test of H0: The canonical correlations in the current row and all that follow are zero

	Likelihood Ratio	Approximate F Value	Num DF	Den DF	Pr > F
1	0.02343863	199.15	8	288	<.0001
2	0.77797337	13.79	3	145	<.0001

Output 27.1.5 Iris Data: Correlations between Canonical and Original Variables

Fisher (1936) Iris Data

The CANDISC Procedure

Total Canonical Structure

Variable	Label	Can1	Can2
SepalLength	Sepal Length in mm.	0.791888	0.217593
SepalWidth	Sepal Width in mm.	-0.530759	0.757989
PetalLength	Petal Length in mm.	0.984951	0.046037
PetalWidth	Petal Width in mm.	0.972812	0.222902

Between Canonical Structure

Variable	Label	Can1	Can2
SepalLength	Sepal Length in mm.	0.991468	0.130348
SepalWidth	Sepal Width in mm.	-0.825658	0.564171
PetalLength	Petal Length in mm.	0.999750	0.022358
PetalWidth	Petal Width in mm.	0.994044	0.108977

Output 27.1.5 *continued*

Pooled Within Canonical Structure

Variable	Label	Can1	Can2
SepalLength	Sepal Length in mm.	0.222596	0.310812
SepalWidth	Sepal Width in mm.	-0.119012	0.863681
PetalLength	Petal Length in mm.	0.706065	0.167701
PetalWidth	Petal Width in mm.	0.633178	0.737242

The raw canonical coefficients (shown in Output 27.1.6) for the first canonical variable, Can1, show that the classes differ most widely on the linear combination of the centered variables: $-0.0829378\times$SepalLength$-0.153447\times$SepalWidth$+0.220121\times$PetalLength$+0.281046\times$PetalWidth.

Output 27.1.6 Iris Data: Canonical Coefficients

Fisher (1936) Iris Data

The CANDISC Procedure

Total-Sample Standardized Canonical Coefficients

Variable	Label	Can1	Can2
SepalLength	Sepal Length in mm.	-0.686779533	0.019958173
SepalWidth	Sepal Width in mm.	-0.668825075	0.943441829
PetalLength	Petal Length in mm.	3.885795047	-1.645118866
PetalWidth	Petal Width in mm.	2.142238715	2.164135931

Pooled Within-Class Standardized Canonical Coefficients

Variable	Label	Can1	Can2
SepalLength	Sepal Length in mm.	-.4269548486	0.0124075316
SepalWidth	Sepal Width in mm.	-.5212416758	0.7352613085
PetalLength	Petal Length in mm.	0.9472572487	-.4010378190
PetalWidth	Petal Width in mm.	0.5751607719	0.5810398645

Raw Canonical Coefficients

Variable	Label	Can1	Can2
SepalLength	Sepal Length in mm.	-.0829377642	0.0024102149
SepalWidth	Sepal Width in mm.	-.1534473068	0.2164521235
PetalLength	Petal Length in mm.	0.2201211656	-.0931921210
PetalWidth	Petal Width in mm.	0.2810460309	0.2839187853

Output 27.1.7 Iris Data: Canonical Means

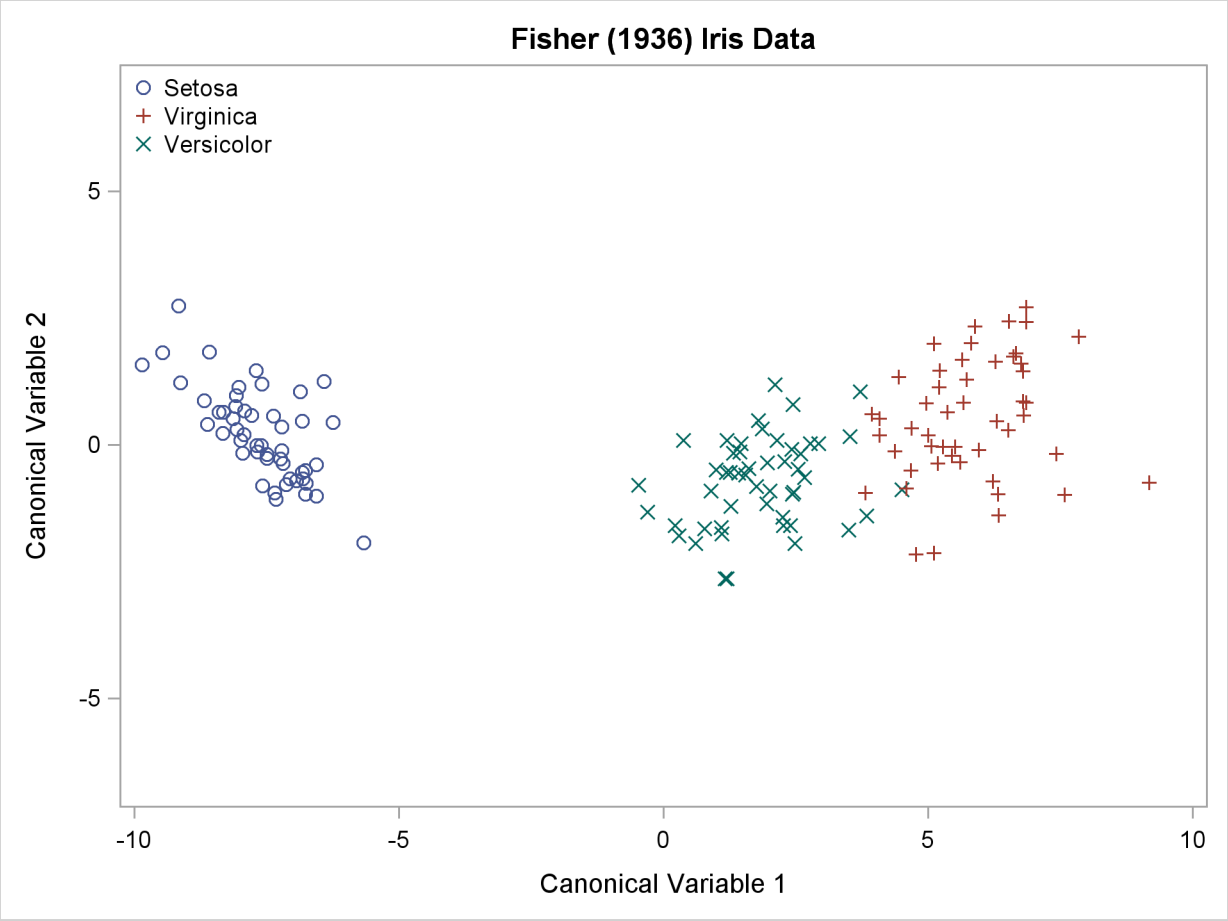
Class Means on Canonical Variables

Species	Can1	Can2
Setosa	-7.607599927	0.215133017
Versicolor	1.825049490	-0.727899622
Virginica	5.782550437	0.512766605

The TEMPLATE and SGRENDER procedures within ODS Graphics are invoked to create a plot of the first two canonical variables. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." The following statements produce Output 27.1.8:

```
proc template;
   define statgraph scatter;
      begingraph;
         entrytitle 'Fisher (1936) Iris Data';
         layout overlayequated / equatetype=fit
            xaxisopts=(label='Canonical Variable 1')
            yaxisopts=(label='Canonical Variable 2');
            scatterplot x=Can1 y=Can2 / group=species name='iris';
            layout gridded / autoalign=(topleft);
               discretelegend 'iris' / border=false opaque=false;
               endlayout;
         endlayout;
      endgraph;
   end;
run;

proc sgrender data=outcan template=scatter;
run;
```

Output 27.1.8 Iris Data: Plot of First Two Canonical Variables



The plot of canonical variables in Output 27.1.8 shows that of the two canonical variables, Can1 has more discriminatory power.

References

Fisher, R. A. (1936), "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, 7, 179–188.

Kshirsagar, A. M. (1972), *Multivariate Analysis*, New York: Marcel Dekker.

Lawley, D. N. (1959), "Tests of Significance in Canonical Analysis," *Biometrika*, 46, 59–66.

Puranen, J. (1917), "Fish Catch data set (1917)," Journal of Statistics Education Data Archive, last accessed May 22, 2009.
URL http://www.amstat.org/publications/jse/jse_data_archive.html

Rao, C. R. (1973), *Linear Statistical Inference*, New York: John Wiley & Sons.

Chapter 28

The CATMOD Procedure

Contents

Overview: CATMOD Procedure **1092**
- Types of Input Data 1093
- Types of Statistical Analyses 1093
- Background: The Underlying Model 1096
- Linear Models Contrasted with Log-Linear Models 1097
- Using PROC CATMOD Interactively 1097

Getting Started: CATMOD Procedure **1098**
- Weighted Least Squares Analysis of Mean Response 1098
- Generalized Logits Model 1103

Syntax: CATMOD Procedure **1107**
- PROC CATMOD Statement 1108
- BY Statement 1109
- CONTRAST Statement 1110
- DIRECT Statement 1113
- FACTORS Statement 1114
- LOGLIN Statement 1117
- MODEL Statement 1118
- POPULATION Statement 1126
- REPEATED Statement 1129
- RESPONSE Statement 1130
- RESTRICT Statement 1137
- WEIGHT Statement 1138

Details: CATMOD Procedure **1138**
- Missing Values 1138
- Input Data Sets 1138
- Ordering of Populations and Responses 1141
- Specification of Effects 1142
- Output Data Sets 1144
- Logistic Analysis 1146
- Log-Linear Model Analysis 1148
- Repeated Measures Analysis 1150
- Generation of the Design Matrix 1153
- Cautions 1163
- Computational Method 1167

Computational Formulas 1168
Memory and Time Requirements 1173
Displayed Output 1173
ODS Table Names 1177
Examples: CATMOD Procedure **1178**
Example 28.1: Linear Response Function, r=2 Responses 1178
Example 28.2: Mean Score Response Function, r=3 Responses 1184
Example 28.3: Logistic Regression, Standard Response Function 1188
Example 28.4: Log-Linear Model, Three Dependent Variables 1193
Example 28.5: Log-Linear Model, Structural and Sampling Zeros 1195
Example 28.6: Repeated Measures, 2 Response Levels, 3 Populations 1202
Example 28.7: Repeated Measures, 4 Response Levels, 1 Population 1206
Example 28.8: Repeated Measures, Logistic Analysis of Growth Curve . . . 1208
Example 28.9: Repeated Measures, Two Repeated Measurement Factors . . 1212
Example 28.10: Direct Input of Response Functions and Covariance Matrix 1219
Example 28.11: Predicted Probabilities 1224
References **1227**

Overview: CATMOD Procedure

The CATMOD procedure performs categorical data modeling of data that can be represented by a contingency table. PROC CATMOD fits linear models to functions of response frequencies, and it can be used for linear modeling, log-linear modeling, logistic regression, and repeated measurement analysis. PROC CATMOD uses the following estimation methods:

- weighted least squares (WLS) estimation of parameters for a wide range of general linear models
- maximum likelihood (ML) estimation of parameters for log-linear models and the analysis of generalized logits

The CATMOD procedure provides a wide variety of categorical data analyses, many of which are generalizations of continuous data analysis methods. For example, analysis of variance, in the traditional sense, refers to the analysis of means and the partitioning of variation among the means into various sources. Here, the term *analysis of variance* is used in a generalized sense to denote the analysis of response functions and the partitioning of variation among those functions into various sources. The response functions might be mean scores if the dependent variables are ordinally scaled. But they can also be marginal probabilities, cumulative logits, or other functions that incorporate the essential information from the dependent variables.

NOTE: PROC CATMOD specializes in WLS modeling and analysis of a wide range of models on contingency tables. For ML modeling on standard models, especially with continuous predictors, it

might be more appropriate to use a procedure such as PROC GENMOD or PROC LOGISTIC; see Chapter 37, "The GENMOD Procedure," and Chapter 51, "The LOGISTIC Procedure," for more information.

Types of Input Data

The data that PROC CATMOD analyzes are usually supplied in one of two ways. First, you can supply raw data, where each observation is a subject. Second, you can supply cell count data, where each observation is a cell in a contingency table. (A third way, which uses direct input of the covariance matrix, is also available; details are given in the section "Inputting Response Functions and Covariances Directly" on page 1140.)

Suppose detergent brand preference is related to three other categorical variables: water softness, water temperature, and previous use of a brand of detergent. In the raw data case, each observation in the input data set identifies a given respondent in the study and contains information about all four variables. The data set contains the same number of observations as the survey had respondents. In the cell count case, each observation identifies a given cell in the four-way table of water softness, water temperature, previous use of brand, and brand preference. A fifth variable contains the number of respondents in the cell. In the analysis, this fifth variable is identified in a WEIGHT statement. The data set contains the same number of observations as the number of cross-classifications formed by the four categorical variables. For more about this particular example, see Example 28.1. For additional details, see the section "Input Data Sets" on page 1138.

Most of the examples in this chapter use cell counts as input and use a WEIGHT statement.

Types of Statistical Analyses

This section illustrates, by example, the wide variety of categorical data analyses that PROC CATMOD provides. For each type of analysis, a brief description of the statistical problem and the SAS statements to provide the analysis are given. For each analysis, assume that the input data set consists of a set of cell counts from a contingency table. The variable specified in the WEIGHT statement contains these counts. In all these analyses, both the dependent and independent variables are categorical.

Linear Model Analysis

Suppose you want to analyze the relationship between the dependent variables (r1, r2) and the independent variables (a, b). Analyze the marginal probabilities of the dependent variables, and use a main-effects model:

```
proc catmod;
   weight wt;
   response marginals;
   model r1*r2=a b;
quit;
```

Log-Linear Model Analysis

Suppose you want to analyze the nominal dependent variables (r1, r2, r3) with a log-linear model. Use maximum likelihood analysis, include the main effects and the r1*r2 interaction in the model, and obtain the predicted cell frequencies:

```
proc catmod;
   weight wt;
   model r1*r2*r3=_response_ / pred=freq;
   loglin r1|r2 r3;
quit;
```

Logistic Regression

Suppose you want to analyze the relationship between the nominal dependent variable (r) and the independent variables (x1, x2) with a logistic regression analysis. Use maximum likelihood estimation:

```
proc catmod;
   weight wt;
   direct x1 x2;
   model r=x1 x2;
quit;
```

If x1 and x2 are continuous so that each observation has a unique value of these two variables, then it might be more appropriate to use the LOGISTIC or GENMOD procedure. (See the section "Logistic Regression" on page 1146.)

Repeated Measures Analysis

Suppose the dependent variables (r1, r2, r3) represent the same type of measurement taken at three different times. Analyze the relationship among the dependent variables, the repeated measurement factor (time), and the independent variable (a):

```
proc catmod;
   weight wt;
   response marginals;
   model r1*r2*r3=_response_|a;
   repeated time 3 / _response_=time;
quit;
```

Analysis of Variance

Suppose you want to investigate the relationship between the dependent variable (r) and the independent variables (a, b). Analyze the mean of the dependent variable, and include all main effects and interactions in the model:

```
proc catmod;
   weight wt;
   response mean;
   model r=a|b;
quit;
```

Linear Regression

PROC CATMOD can analyze the relationship between the dependent variables (r1, r2) and the independent variables (x1, x2). Use a linear regression analysis to analyze the marginal probabilities of the dependent variables:

```
proc catmod;
   weight wt;
   direct x1 x2;
   response marginals;
   model r1*r2=x1 x2;
quit;
```

Logistic Analysis of Ordinal Data

Suppose you want to analyze the relationship between the ordinally scaled dependent variable (r) and the independent variable (a). Use cumulative logits to take into account the ordinal nature of the dependent variable, and use weighted least squares estimation:

```
proc catmod;
   weight wt;
   response clogits;
   model r=_response_ a;
quit;
```

Sample Survey Analysis

Suppose the data set contains estimates of a vector of four functions and their covariance matrix, estimated in such a way as to correspond to the sampling process that is used. Analyze the functions with respect to the independent variables (a, b), and use a main-effects model:

```
proc catmod;
   response read b1-b10;
   model _f_=_response_;
   factors  a 2 , b 5 / _response_=a b;
quit;
```

Background: The Underlying Model

The CATMOD procedure analyzes data that can be represented by a two-dimensional contingency table. The rows of the table correspond to populations (or samples) formed on the basis of one or more independent variables. The columns of the table correspond to observed responses formed on the basis of one or more dependent variables. The frequency in the (i, j)th cell is the number of subjects in the ith population that have the jth response. The frequencies in the table are assumed to follow a product multinomial distribution, corresponding to a sampling design in which a simple random sample is taken for each population. The contingency table can be represented as shown in Table 28.1.

Table 28.1 Contingency Table Representation

	Response				
Sample	**1**	**2**	$\cdots$	r	**Total**
1	n_{11}	n_{12}	$\cdots$	n_{1r}	n_1
2	n_{21}	n_{22}	$\cdots$	n_{2r}	n_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
s	n_{s1}	n_{s2}	$\cdots$	n_{sr}	n_s

For each sample i, the probability of the jth response (π_{ij}) is estimated by the sample proportion, $p_{ij} = n_{ij}/n_i$. The vector ($\mathbf{p}$) of all such proportions is then transformed into a vector of functions, denoted by $\mathbf{F} = \mathbf{F}(\mathbf{p})$. If $\boldsymbol{\pi}$ denotes the vector of true probabilities for the entire table, then the functions of the true probabilities, denoted by $\mathbf{F}(\boldsymbol{\pi})$, are assumed to follow a linear model

$$\mathbf{E_A}(\mathbf{F}) = \mathbf{F}(\boldsymbol{\pi}) = \mathbf{X}\boldsymbol{\beta}$$

where $\mathbf{E_A}$ denotes asymptotic expectation, $\mathbf{X}$ is the design matrix containing fixed constants, and $\boldsymbol{\beta}$ is a vector of parameters to be estimated.

PROC CATMOD provides two estimation methods:

- The weighted least squares method minimizes the weighted residual sum of squares for the model. The weights are contained in the inverse covariance matrix of the functions $\mathbf{F}(\mathbf{p})$. According to central limit theory, if the sample sizes within populations are sufficiently large, the elements of $\mathbf{F}$ and $\mathbf{b}$ (the estimate of $\boldsymbol{\beta}$) are distributed approximately as multivariate normal. This allows the computation of statistics for testing the goodness of fit of the model and the significance of other sources of variation. For details of the theory, see Grizzle, Starmer, and Koch (1969) or Koch et al. (1977, Appendix 1). Weighted least squares estimation is available for all types of response functions.
- The maximum likelihood method estimates the parameters of the linear model so as to maximize the value of the joint multinomial likelihood function of the responses. Maximum likelihood estimation is available only for the standard response functions, logits and generalized logits, which are used for logistic regression analysis and log-linear model analysis. Two methods of maximization are available: Newton-Raphson and iterative proportional fitting. For details of the theory, see Bishop, Fienberg, and Holland (1975).

Following parameter estimation, hypotheses about linear combinations of the parameters can be tested. For that purpose, PROC CATMOD computes generalized Wald (1943) statistics, which are approximately chi-square distributed if the sample sizes are sufficiently large and the null hypotheses are true.

Linear Models Contrasted with Log-Linear Models

Linear model methods typified by the Grizzle, Starmer, and Koch (1969) approach make a very clear distinction between independent and dependent variables. The emphasis of these methods is estimation and hypothesis testing of the model parameters. Therefore, it is easy to test for differences among probabilities, perform repeated measures analysis, and test for marginal homogeneity, but it is difficult to test for independence and generalized independence. These methods are a natural extension of the usual ANOVA approach for continuous data.

In contrast, log-linear model methods typified by the Bishop, Fienberg, and Holland (1975) approach do not make an a priori distinction between independent and dependent variables, although model specifications that allow for the distinction can be made. The emphasis of these methods is on model building, goodness-of-fit tests, and estimation of cell frequencies or probabilities for the underlying contingency table. With these methods, it is easy to test independence and generalized independence, but it is difficult to test for differences among probabilities, do repeated measures analysis, and test for marginal homogeneity.

Using PROC CATMOD Interactively

You can use the CATMOD procedure interactively. After specifying a model with a MODEL statement and running PROC CATMOD with a RUN statement, you can execute any statement without reinvoking PROC CATMOD. You can execute the statements singly or in groups by following the single statement or group of statements with a RUN statement. Note that you can use more than one MODEL statement; this is an important difference from the GLM procedure.

If you use PROC CATMOD interactively, you can end the CATMOD procedure with a DATA step, another PROC step, an ENDSAS statement, or a QUIT statement. The syntax of the QUIT statement is as follows:

```
quit;
```

When you are using PROC CATMOD interactively, additional RUN statements do not end the procedure run but tell the procedure to execute additional statements.

When the CATMOD procedure detects a BY statement, it disables interactive processing; that is, once the BY statement and the next RUN statement are encountered, processing proceeds for each BY group in the data set, and no additional statements are accepted by the procedure. For example, the following statements perform three analyses: one for the entire data set, one for males, and one for females:

```
proc catmod;
   weight wt;
   response marginals;
   model r1*r2=a|b;
run;
   by sex;
run;
```

Note that the BY statement can appear after the first RUN statement; this is an important difference from PROC GLM, which requires that the BY statement appear before the first RUN statement.

Getting Started: CATMOD Procedure

The CATMOD procedure is a general modeling procedure for categorical data analysis, and it can be used for sophisticated analyses that require matrix specification of the response function and the design matrix. It can also be used to perform basic analysis-of-variance-type analyses that require only a few statements. The following is a basic example.

Weighted Least Squares Analysis of Mean Response

Consider the data in Table 28.2 (Stokes, Davis, and Koch 2000).

Table 28.2 Colds in Children

		Periods with Colds			
Sex	**Residence**	**0**	**1**	**2**	**Total**
Female	Rural	45	64	71	180
Female	Urban	80	104	116	300
Male	Rural	84	124	82	290
Male	Urban	106	117	87	310

For male and female children in rural and urban counties, the number of periods (of two) in which subjects report cold symptoms are recorded. So 45 subjects who are female and in rural counties report no cold symptoms, and 71 subjects who are female and from rural counties report colds in both periods.

The question of interest is whether the mean number of periods with colds reported is associated with gender or type of county. There is no reason to believe that the mean number of periods with colds is normally distributed, so a weighted least squares analysis of these data is performed with PROC CATMOD instead of an analysis of variance with PROC ANOVA or PROC GLM.

The input data for categorical data are often recorded in frequency form, with the counts for each particular profile being the input values. For the colds data, the input SAS data set colds is created with the following statements. The variable count contains the frequency of observations that have the particular profile described by the values of the other variables in that input line.

```
data colds;
   input sex $ residence $ periods count @@;
datalines;
female rural 0  45  female rural 1  64  female rural 2  71
female urban 0  80  female urban 1 104  female urban 2 116
male   rural 0  84  male   rural 1 124  male   rural 2  82
male   urban 0 106  male   urban 1 117  male   urban 2  87
;
run;
```

In order to fit a model to the mean number of periods with colds, you have to specify the response function in PROC CATMOD. The default response function is the logit if the response variable has two values, and it is generalized logits if the response variable has more than two values. If you want a different response function, then you specify that function in the RESPONSE statement. To request the mean number of periods with colds, you specify the MEANS option in the RESPONSE statement.

You can request a model consisting of the main effects and interaction of the variables sex and residence just as you would in the GLM procedure. Unlike the GLM procedure, PROC CATMOD does not require you to use a CLASS statement to treat a variable as a classification variable. In the CATMOD procedure, all variables in the MODEL statement are treated as classification variables unless you specify otherwise with a DIRECT statement. To verify that your model is specified correctly, you can specify the DESIGN option in the MODEL statement to display the design matrix.

The PROC CATMOD statements needed to model mean periods of colds with a main-effects and interaction model are as follows:

```
proc catmod data=colds;
   weight count;
   response means;
   model periods = sex residence sex*residence / design;
run;
```

The results of this analysis are shown in Figure 28.1 through Figure 28.3.

In Figure 28.1, the CATMOD procedure first displays a summary of the contingency table you are analyzing. The “Population Profiles” table lists the values of the explanatory variables that define each population, or row of the underlying contingency table, and labels each group with a sample number. The number of observations in each population is also displayed. The “Response Profiles” table lists the variable levels that define the response, or columns of the underlying contingency table.

Figure 28.1 Model Information and Profile Tables

The CATMOD Procedure

Data Summary

Response	periods	Response Levels	3
Weight Variable	count	Populations	4
Data Set	COLDS	Total Frequency	1080
Frequency Missing	0	Observations	12

Population Profiles

Sample	sex	residence	Sample Size
1	female	rural	180
2	female	urban	300
3	male	rural	290
4	male	urban	310

Response Profiles

Response	periods
1	0
2	1
3	2

The "Response Functions and Design Matrix" table in Figure 28.2 contains the observed response functions—in this case, the mean number of periods with colds for each of the populations—and the design matrix. The first column of the design matrix contains the coefficients for the intercept parameter. The second column contains the coefficients for the sex parameter. (Note that the sum-to-zero constraint of the default full-rank parameterization PARAM=EFFECT implies that the coefficient for males is the negative of that for females; the parameter is called the *differential effect* for females.) The third column is similarly set up for residence, and the last column is for the interaction.

Figure 28.2 Observed Response Functions and Design Matrix

Response Functions and Design Matrix

Sample	Response Function	Design Matrix 1	2	3	4
1	1.14444	1	1	1	1
2	1.12000	1	1	-1	-1
3	0.99310	1	-1	1	-1
4	0.93871	1	-1	-1	1

The model-fitting results are displayed in the "Analysis of Variance" table (Figure 28.3), which is similar to an ANOVA table. The effects from the right side of the MODEL statement are listed in the Source column.

Figure 28.3 ANOVA Table for the Saturated Model

Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	1841.13	<.0001
sex	1	11.57	0.0007
residence	1	0.65	0.4202
sex*residence	1	0.09	0.7594
Residual	0	.	.

You can see in Figure 28.3 that the interaction effect is nonsignificant, so the data are reanalyzed using a main-effects model. Since PROC CATMOD is an interactive procedure, you can analyze the main-effects model by simply submitting the new MODEL statement as follows. The resulting tables are displayed in Figure 28.4 and Figure 28.5.

```
    model periods = sex residence / design;
run;
```

From the ANOVA table in Figure 28.4, you can see that the goodness-of-fit chi-square statistic is 0.09 with one degree of freedom and a p-value of 0.7594; hence, the model fits the data. Note that the chi-square tests in Figure 28.4 check whether all the parameters for a given effect are zero. In this model, each effect has only one parameter and therefore only one degree of freedom.

Figure 28.4 Main-Effects Model

The CATMOD Procedure

Data Summary

Response	periods	Response Levels	3
Weight Variable	count	Populations	4
Data Set	COLDS	Total Frequency	1080
Frequency Missing	0	Observations	12

Population Profiles

Sample	sex	residence	Sample Size
1	female	rural	180
2	female	urban	300
3	male	rural	290
4	male	urban	310

Response Profiles

Response	periods
1	0
2	1
3	2

Figure 28.4 *continued*

Response Functions and Design Matrix

Sample	Response Function	Design Matrix 1	2	3
1	1.14444	1	1	1
2	1.12000	1	1	-1
3	0.99310	1	-1	1
4	0.93871	1	-1	-1

Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	1882.77	<.0001
sex	1	12.08	0.0005
residence	1	0.76	0.3839
Residual	1	0.09	0.7594

The "Analysis of Weighted Least Squares Estimates" table in Figure 28.5 lists the parameters and their estimates for the model, as well as the standard errors, Wald statistics, and p-values. These chi-square tests are one-degree-of-freedom tests that the individual parameter is equal to zero. They are equal to the tests shown in Figure 28.4 since each effect is composed of exactly one parameter.

Figure 28.5 Parameter Estimates for the Main-Effects Model

Analysis of Weighted Least Squares Estimates

Parameter		Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept		1.0501	0.0242	1882.77	<.0001
sex	female	0.0842	0.0242	12.08	0.0005
residence	rural	0.0210	0.0241	0.76	0.3839

You can compute the mean number of periods with colds for the first population (Sample 1, females in rural residences) from Table 28.2 as follows:

$$\text{mean colds} = 0 \times \frac{45}{180} + 1 \times \frac{64}{180} + 2 \times \frac{71}{180} = 1.1444$$

This is the same value reported in the Response Function column for Sample 1 in the "Response Functions and Design Matrix" table displayed in Figure 28.4.

PROC CATMOD is fitting a model to the mean number of colds in each population as follows:

$$\begin{bmatrix} \text{Expected number of colds for rural females} \\ \text{urban females} \\ \text{rural males} \\ \text{urban males} \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \end{bmatrix}$$

where the design matrix is the same one displayed in Figure 28.4, β_0 is the mean number of colds averaged over all the populations, β_1 is the differential effect for females, and β_2 is the differential effect for rural residences. The parameter estimates are shown in Figure 28.5; the expected number of periods with colds for rural females from this model is computed as

$$1 \times 1.0501 + 1 \times 0.0842 + 1 \times 0.0210 = 1.1553$$

and the expected number for rural males from this model is

$$1 \times 1.0501 - 1 \times 0.0842 + 1 \times 0.0210 = 0.9869$$

Notice also, in Figure 28.5, that the differential effect for residence is nonsignificant ($p = 0.3839$). If you continue the analysis by fitting a single-effect model (sex), you need to include a POPULATION statement to maintain the same underlying contingency table:

```
    population sex residence;
    model periods = sex;
run;
```

Generalized Logits Model

Over the course of one school year, third-graders from three different schools are exposed to three different styles of mathematics instruction: a self-paced computer-learning style, a team approach, and a traditional class approach. The students are asked which style they prefer, and their responses, classified by the type of program they are in (a regular school day versus a regular school day supplemented with an afternoon school program), are displayed in Table 28.3. The data set is from Stokes, Davis, and Koch (2000), and it is also analyzed in the section "Example 51.4: Nominal Response Data: Generalized Logits Model" on page 3412 of Chapter 51, "The LOGISTIC Procedure."

Table 28.3 School Program Data

		Learning Style Preference		
School	**Program**	**Self**	**Team**	**Class**
1	Regular	10	17	26
1	Afternoon	5	12	50
2	Regular	21	17	26
2	Afternoon	16	12	36
3	Regular	15	15	16
3	Afternoon	12	12	20

The levels of the response variable (self, team, and class) have no essential ordering, so a logistic regression is performed on the generalized logits. The model to be fit is

$$\log\left(\frac{\pi_{hij}}{\pi_{hir}}\right) = \alpha_j + \mathbf{x}'_{hi}\boldsymbol{\beta}_j$$

where π_{hij} is the probability that a student in school h and program i prefers teaching style j, $j \neq r$, and style r is the class style. There are separate sets of intercept parameters α_j and regression parameters β_j for each logit, and the matrix $\mathbf{x}_{hi}$ is the set of explanatory variables for the hith population. Thus, two logits are modeled for each school and program combination (population): the logit comparing self to class and the logit comparing team to class.

The following statements create the data set school and request the analysis. Generalized logits are the default response functions, and maximum likelihood estimation is the default method for analyzing generalized logits, so only the WEIGHT and MODEL statements are required. The option ORDER=DATA means that the response variable levels are ordered as they exist in the data set: self, team, and class; the logits are formed by comparing self to class and by comparing team to class. The results of this analysis are shown in Figure 28.6 and Figure 28.7.

```
data school;
   length Program $ 9;
   input School Program $ Style $ Count @@;
   datalines;
1 regular   self 10  1 regular   team 17  1 regular   class 26
1 afternoon self  5  1 afternoon team 12  1 afternoon class 50
2 regular   self 21  2 regular   team 17  2 regular   class 26
2 afternoon self 16  2 afternoon team 12  2 afternoon class 36
3 regular   self 15  3 regular   team 15  3 regular   class 16
3 afternoon self 12  3 afternoon team 12  3 afternoon class 20
;

proc catmod order=data;
   weight Count;
   model Style=School Program School*Program;
run;
```

A summary of the data set is displayed in Figure 28.6; the variable levels that form the three responses and six populations are listed in the "Response Profiles" and "Population Profiles" tables, respectively.

Figure 28.6 Model Information and Profile Tables

```
                         The CATMOD Procedure

                             Data Summary

            Response           Style      Response Levels    3
            Weight Variable    Count      Populations        6
            Data Set           SCHOOL     Total Frequency  338
            Frequency Missing  0          Observations      18
```

Figure 28.6 *continued*

Population Profiles

Sample	School	Program	Sample Size
1	1	regular	53
2	1	afternoon	67
3	2	regular	64
4	2	afternoon	64
5	3	regular	46
6	3	afternoon	44

Response Profiles

Response	Style
1	self
2	team
3	class

The analysis of variance table is displayed in Figure 28.7. Since this is a saturated model, there are no degrees of freedom remaining for a likelihood ratio test, and missing values are displayed in the table. The interaction effect is clearly nonsignificant, so a main-effects model is fit.

Figure 28.7 Saturated Model: ANOVA Table

Maximum Likelihood Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	2	40.05	<.0001
School	4	14.55	0.0057
Program	2	10.48	0.0053
School*Program	4	1.74	0.7827
Likelihood Ratio	0	.	.

Since PROC CATMOD is an interactive procedure, you can analyze the main-effects model by simply submitting the new MODEL statement as follows:

```
model Style=School Program;
run;
```

You can check the population and response profiles (not shown) to confirm that they are the same as those in Figure 28.6. The analysis of variance table is shown in Figure 28.8. The likelihood ratio chi-square statistic is 1.78 with a p-value of 0.7766, indicating a good fit; the Wald chi-square tests for the school and program effects are also significant. Since School has three levels, two parameters are estimated for each of the two logits they modeled, for a total of four degrees of freedom. Since Program has two levels, one parameter is estimated for each of the two logits, for a total of two degrees of freedom.

Figure 28.8 Main-Effects Model: ANOVA Table

Maximum Likelihood Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	2	39.88	<.0001
School	4	14.84	0.0050
Program	2	10.92	0.0043
Likelihood Ratio	4	1.78	0.7766

The parameter estimates and tests for individual parameters are displayed in Figure 28.9. The order of the parameters corresponds to the order of the population and response variables as shown in the profile tables (see Figure 28.6), with the levels of the response variables varying most rapidly. The first response function is the logit that compares self to class, and the corresponding parameters have Function Number=1. The second logit (Function Number=2) compares team to class. The School=1 parameters are the differential effects versus School=3 for their respective logits, and the School=2 parameters are likewise differential effects versus School=3. The Program parameters are the differential effects of 'regular' versus 'afternoon' for the two response functions.

Figure 28.9 Parameter Estimates

Analysis of Maximum Likelihood Estimates

Parameter		Function Number	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept		1	-0.7979	0.1465	29.65	<.0001
		2	-0.6589	0.1367	23.23	<.0001
School	1	1	-0.7992	0.2198	13.22	0.0003
	1	2	-0.2786	0.1867	2.23	0.1356
	2	1	0.2836	0.1899	2.23	0.1352
	2	2	-0.0985	0.1892	0.27	0.6028
Program	regular	1	0.3737	0.1410	7.03	0.0080
	regular	2	0.3713	0.1353	7.53	0.0061

The Program variable has nearly the same effect on both logits, while School=1 has the largest effect of the schools.

Syntax: CATMOD Procedure

The following statements are available in PROC CATMOD:

PROC CATMOD < *options* > ;
DIRECT < *variables* > ;
MODEL *response-effect=design-effects* < / *options* > ;
CONTRAST *'label' row-description* < , . . ., *row-description* >< / *options* > ;
BY *variables* ;
FACTORS *factor-description* < , . . ., *factor-description* >< / *options* > ;
LOGLIN *effects*< / *option* > ;
POPULATION *variables* ;
REPEATED *factor-description* < , . . ., *factor-description* >< / *options* > ;
RESPONSE < *function* >< / *options* > ;
RESTRICT *parameter=value* < . . . *parameter=value* > ;
WEIGHT *variable* ;

You can use all of the statements in PROC CATMOD interactively. The first RUN statement executes all of the previous statements. Any subsequent RUN statement executes only those statements that appear between the previous RUN statement and the current one. However, if you specify a BY statement, interactive processing is disabled. That is, all statements through the following RUN statement are processed for each BY group in the data set, but no additional statements are accepted by the procedure.

If more than one CONTRAST statement appears between two RUN statements, all the CONTRAST statements are processed. If more than one RESPONSE statement appears between two RUN statements, then analyses associated with each RESPONSE statement are produced. For all other statements, there can be only one occurrence of the statement between any two RUN statements. For example, if there are two LOGLIN statements between two RUN statements, the first LOGLIN statement is ignored.

The PROC CATMOD and MODEL statements are required. If specified, the DIRECT statement must precede the MODEL statement. As a result, if you use the DIRECT statement interactively, you need to specify a MODEL statement in the same RUN group. See the section "DIRECT Statement" on page 1113 for an example.

The CONTRAST statements, if any, must follow the MODEL statement.

You can specify only one of the LOGLIN, REPEATED, and FACTORS statements between any two RUN statements, because they all specify the same information: how to partition the variation among the response functions within a population.

A QUIT statement executes any statements that have not been processed and then ends the CATMOD procedure run.

The purpose of each statement, other than the PROC CATMOD statement, is summarized in the following list:

BY	determines groups in which data are to be processed separately.
CONTRAST	specifies a hypothesis to test.
DIRECT	specifies independent variables that are to be treated quantitatively (like continuous variables) rather than qualitatively (like classification or discrete variables). These variables also help to determine the rows of the contingency table and distinguish response functions in one population from those in other populations.
FACTORS	specifies (1) the factors that distinguish response functions from others in the same population and (2) model effects, based on these factors, which help to determine the design matrix.
LOGLIN	specifies log-linear model effects.
MODEL	specifies (1) dependent variables, which determine the columns of the contingency table, (2) independent variables, which distinguish response functions in one population from those in other populations, and (3) model effects, which determine the design matrix and the way in which total variation among the response functions is partitioned.
POPULATION	specifies variables that determine the rows of the contingency table and distinguish response functions in one population from those in other populations.
REPEATED	specifies (1) the repeated measurement factors that distinguish response functions from others in the same population and (2) model effects, based on these factors, which help to determine the design matrix.
RESPONSE	determines the response functions that are to be modeled.
RESTRICT	restricts values of parameters to the values you specify.
WEIGHT	specifies a variable containing frequency counts.

PROC CATMOD Statement

PROC CATMOD < *options* > **;**

The PROC CATMOD statement invokes the procedure. You can specify the following options.

DATA=*SAS-data-set*
names the SAS data set containing the data to be analyzed. By default, the CATMOD procedure uses the most recently created SAS data set. For details, see the section "Input Data Sets" on page 1138.

NAMELEN=*n*
specifies the length of effect names in tables and output data sets to be *n* characters long, where *n* is a value between 24 and 200. The default length is 24 characters.

NOPRINT
suppresses the normal display of results. The NOPRINT option is useful when you only want to create output data sets with the OUT= or OUTEST= option in the RESPONSE statement. A NOPRINT option is also available in the MODEL statement. Note that this option temporarily

disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of classification variables. This affects the ordering of the populations, responses, and parameters, as well as the definitions of the parameters. The default, ORDER=INTERNAL, orders the variable levels by their unformatted values (for example, numeric order or alphabetical order).

The following table shows how PROC CATMOD interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=INTERNAL. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine dependent. See the section "Ordering of Populations and Responses" on page 1141 for more information and examples. For more information about sorting order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC CATMOD to obtain separate analyses of groups determined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED, or DESCENDING in the BY statement for the CATMOD procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about the BY statement, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

When you specify a BY statement with PROC CATMOD, no further interactive processing is possible. In other words, once the BY statement appears, all statements up to the associated RUN statement are executed for each BY group in the data set. After the RUN statement, no further statements are accepted by the procedure.

CONTRAST Statement

CONTRAST *'label' row-description*<*, . . ., row-description*><*/ options*> **;**

where a *row-description* is defined as follows:

<*@n*> *effect values* <*. . .* <*@n*> *effect values* >

The CONTRAST statement constructs and tests linear functions of the parameters in the MODEL statement or effects listed in the LOGLIN statement. Each set of effects (separated by commas) specifies one row or set of rows of the matrix $\mathbf{C}$ that PROC CATMOD uses to test the hypothesis $\mathbf{C}\boldsymbol{\beta} = \mathbf{0}$.

CONTRAST statements must be preceded by the MODEL statement, and by the LOGLIN statement, if one is used. You can specify the following terms in the CONTRAST statement.

'*label*' specifies up to 256 characters of identifying information displayed with the test. The '*label*' is required.

effect is one of the effects specified in the MODEL or LOGLIN statement, INTERCEPT (for the intercept parameter), or ALL_PARMS (for the complete set of parameters).

The ALL_PARMS option is regarded as an effect with the same number of parameters as the number of columns in the design matrix. This is particularly useful when the design matrix is input directly, as in the following example:

```
model y=(1 0 0 0,
         1 0 1 0,
         1 1 0 0,
         1 1 1 1);
contrast 'Main Effect of B' all_parms 0 1 0 0;
contrast 'Main Effect of C' all_parms 0 0 1 0;
contrast 'B*C Interaction ' all_parms 0 0 0 1;
```

values are numbers that form the coefficients of the parameters associated with the given effect. If there are fewer values than parameters for an effect, the remaining coefficients become zero. For example, if you specify two values and the effect actually has five parameters, the final three are set to zero.

@n points to the parameters in the *n*th set when the model has a separate set of parameters for each of the response functions. The *@n* notation is seldom needed. It enables you to test the variation among response functions in the same population. However,

it is usually easier to model and test such variation by using the _RESPONSE_ effect in the MODEL statement or by using the ALL_PARMS designation. Usually, contrasts are performed with respect to all of the response functions, and this is what the CONTRAST statement does by default (in this case, do not use the @*n* notation).

For example, if there are three response functions per population, then the following contrast results in a three-degree-of-freedom test comparing the first two levels of A simultaneously on the three response functions.

```
contrast 'Level 1 vs. Level 2' A 1 -1 0;
```

If, however, you want to specify a contrast with respect to the parameters in the *n*th set only, then use a single @*n* in a *row-description*. For example, the following statement tests that the first parameter of A and the first parameter of B are zero in the third response function:

```
contrast 'A=0, B=0, Function 3'  @3  A 1  B 1;
```

To specify a contrast with respect to parameters in two or more different sets of effects, use @*n* with each effect. For example:

```
contrast 'Average over Functions' @1 A 1 0 -1
                                  @2 A 1 1 -2;
```

When the model does not have a separate set of parameters for each of the response functions, the @*n* notation is invalid. This type of model is called AVERAGED. For details, see the description of the AVERAGED option and the section "Generation of the Design Matrix" on page 1153.

You can specify the following options in the CONTRAST statement after a slash.

ALPHA=*value*

specifies the significance level of the confidence interval for each contrast when the ESTIMATE= option is specified. The default is ALPHA=0.05, resulting in a 95% confidence interval for each contrast.

ESTIMATE=*keyword*

EST=*keyword*

requests that each individual contrast (that is, each row, $c_i\boldsymbol{\beta}$, of $\mathbf{C}\boldsymbol{\beta}$) or exponentiated contrast ($\exp(c_i\boldsymbol{\beta})$) be estimated and tested. PROC CATMOD displays the point estimate, its standard error, a Wald confidence interval, and a Wald chi-square test for each contrast. The significance level of the confidence interval is controlled by the ALPHA= option.

You can estimate the contrast or the exponentiated contrast, or both, by specifying one of the following keywords:

PARM	specifies that the contrast itself be estimated.
EXP	specifies that the exponentiated contrast be estimated.
BOTH	specifies that both the contrast and the exponentiated contrast be estimated.

Specifying Contrasts

PROC CATMOD is parameterized differently than PROC GLM, so you must be careful not to use the same contrasts that you would with PROC GLM. Since PROC CATMOD uses full-rank parameterizations, all estimable parameters are directly estimable without involving other parameters.

For example, suppose a classification variable A has four levels and uses the default parameterization (PARAM=EFFECT). Then there are four parameters $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$, of which PROC CATMOD uses only the first three. The fourth parameter is related to the others by the equation

$$\alpha_4 = -\alpha_1 - \alpha_2 - \alpha_3$$

To test the first versus the fourth level of A, you would test $\alpha_1 = \alpha_4$, which is

$$\alpha_1 = -\alpha_1 - \alpha_2 - \alpha_3$$

or, equivalently,

$$2\alpha_1 + \alpha_2 + \alpha_3 = 0$$

Therefore, you would use the following CONTRAST statement:

```
contrast '1 vs. 4' A 2 1 1;
```

To contrast the third level with the average of the first two levels, you would test

$$\frac{\alpha_1 + \alpha_2}{2} = \alpha_3$$

or, equivalently,

$$\alpha_1 + \alpha_2 - 2\alpha_3 = 0$$

Therefore, you would use the following CONTRAST statement:

```
contrast '1&2 vs. 3' A 1 1 -2;
```

Other CONTRAST statements are constructed similarly. For example:

```
contrast '1 vs. 2    '  A  1 -1  0;
contrast '1&2 vs. 4  '  A  3  3  2;
contrast '1&2 vs. 3&4'  A  2  2  0;
contrast 'Main Effect'  A  1  0  0,
                        A  0  1  0,
                        A  0  0  1;
```

The actual form of the **C** matrix depends on the effects in the model. The remaining examples in this section assume a single response function for each population.

Recall that the statements to test the first versus the fourth level of A are as follows:

```
proc catmod;
   model y=a;
   contrast '1 vs. 4' A 2 1 1;
run;
```

Since the first parameter corresponds to the intercept, the **C** matrix for the preceding statements is

$$\mathbf{C} = [\,0\;2\;1\;1\,]$$

But suppose you have a variable B with three levels and you use the following statements:

```
proc catmod;
   model y=b a;
   contrast '1 vs. 4' A 2 1 1;
run;
```

Then the CONTRAST statement produces the **C** matrix

$$\mathbf{C} = [\,0\;0\;0\;2\;1\;1\,]$$

since the first parameter corresponds to the intercept and the next two correspond to the B main effect.

You can also use the CONTRAST statement to test the joint effect of two or more effects in the MODEL statement. For example, the joint effect of A and B in the previous model has five degrees of freedom and is obtained by specifying the following:

```
contrast 'Joint Effect of A&B' A 1 0 0,
                               A 0 1 0,
                               A 0 0 1,
                               B 1 0,
                               B 0 1;
```

The ordering of variable levels is determined by the ORDER= option in the PROC CATMOD statement. Whenever you specify a contrast that depends on the order of the variable levels, you should verify the order from the "Population Profiles" table, the "Response Profiles" table, or the "One-Way Frequencies" table.

DIRECT Statement

DIRECT *variables* **;**

The DIRECT statement lists numeric independent variables to be treated in a quantitative, rather than qualitative, way. The DIRECT statement is useful for logistic regression, which is described in the section "Logistic Regression" on page 1146. For limitations of models involving continuous variables, see the section "Continuous Variables" on page 1147.

CAUTION: If a DIRECT variable is formatted, then the unformatted (internal) values are used in the analysis and the formatted values are displayed. If you use a format to group the internal values into one formatted value, then the first internal value is used in the analysis. If specified, the DIRECT statement must precede the MODEL statement. For example:

```
proc catmod;
   direct X;
   model Y=X;
run;
```

Suppose X has five levels. Then the main effect X adds only one column to the design matrix rather than four. The values inserted into the design matrix are the actual values of X.

You can interactively change the variables declared as DIRECT variables by using the statement without listing any variables. The following statements are valid:

```
proc catmod;
   direct X;
   model Y=X;
   weight wt;
run;
   direct;
   model Y=X;
run;
```

The first MODEL statement uses the actual values of X, and the second MODEL statement uses the four variables created when PROC CATMOD generates the design matrix. Note that the preceding statements can be run without a WEIGHT statement if the input data are raw data rather than cell counts.

For more details, see the discussions of main and direct effects in the section "Generation of the Design Matrix" on page 1153.

FACTORS Statement

FACTORS *factor-description* < *, . . ., factor-description* >< */ options* > **;**

where a *factor-description* is defined as follows:

factor-name < *$* >< *levels* >

and *factor-description*s are separated from each other by a comma. The $ is required for character-valued factors. The value of *levels* provides the number of levels of the factor identified by a given *factor-name*. For only one factor, *levels* is optional; for two or more factors, it is required.

The FACTORS statement identifies factors that distinguish response functions from others in the same population. It also specifies how those factors are incorporated into the model. You can use the FACTORS statement whenever there is more than one response function per population and the keyword _RESPONSE_ is specified in the MODEL statement. You can specify the name, type, and number of levels of each factor and the identification of each level.

The FACTORS statement is most useful when the response functions and their covariance matrix are read directly from the input data set. In this case, PROC CATMOD reads the response functions as though they are from one population (this poses no problem in the multiple-population case because the appropriately constructed covariance matrix is also read directly). Thus, you can use

the FACTORS statement to partition the variation among the response functions into appropriate sources, even when the functions actually represent separate populations.

The format of the FACTORS statement is identical to that of the REPEATED statement. In fact, repeated measurement factors are simply special cases of factors in which some of the response functions correspond to multiple dependent variables that are measurements on the same experimental (or sampling) units.

You cannot specify the FACTORS statement for an analysis that also contains the REPEATED or LOGLIN statement since all of them specify the same information: how to partition the variation among the response functions within a population.

You can specify the following terms in the FACTORS statement:

factor-name names a factor that corresponds to two or more response functions. This name must be a valid SAS variable name, and it should not be the same as the name of a variable that already exists in the data set being analyzed.

$ indicates that the factor is character-valued. If the $ is omitted, then the CATMOD procedure assumes that the factor is numeric. The type of the factor is relevant only when you use the PROFILE= option or when the _RESPONSE_= option (described later in this section) specifies nested-by-value effects.

levels specifies the number of levels of the corresponding factor. If there is only one such factor, and the number is omitted, then PROC CATMOD assumes that the number of levels is equal to the number of response functions per population (q). Unless you specify the PROFILE= option, the number q must either be equal to or be a multiple of the product of the number of levels of all the factors.

You can specify the following options in the FACTORS statement after a slash.

PROFILE=(*matrix*)

specifies the values assumed by the factors for each response function. There should be one column for each factor, and the values in a given column (character or numeric) should match the type of the corresponding factor. Character values are restricted to 16 characters or less. If there are q response functions per population, then the matrix must have i rows, where q must either be equal to or be a multiple of i. Adjacent rows of the matrix should be separated by a comma.

The values in the PROFILE matrix are useful for specifying models in those situations where the study design is not a full factorial with respect to the factors. They can also be used to specify nested-by-value effects in the _RESPONSE_= option. If you specify character values in both places (the PROFILE= option and the _RESPONSE_= option), then the values must match with respect to whether or not they are enclosed in quotes (that is, enclosed in quotes in both places or in neither place).

For an example of using the PROFILE= option, see Example 28.10.

RESPONSE=*effects*

specifies design effects. The variables named in the effects must be *factor-names* that appear in the FACTORS statement. If the _RESPONSE_= option is omitted, then PROC CATMOD builds a full factorial _RESPONSE_ effect with respect to the factors.

TITLE=*'title'*

displays the *title* at the top of certain pages of output that correspond to the current FACTORS statement.

For an example of how the FACTORS statement is useful, consider the case where the response functions and their covariance matrix are read directly from the input data set. The TYPE=EST data set might be created in the following manner:

```
data direct(type=est);
   input b1-b4 _type_ $ _name_ $8.;
   datalines;
0.590463   0.384720   0.273269   0.136458   parms      .
0.001690   0.000911   0.000474   0.000432   cov        b1
0.000911   0.001823   0.000031   0.000102   cov        b2
0.000474   0.000031   0.001056   0.000477   cov        b3
0.000432   0.000102   0.000477   0.000396   cov        b4
;
```

Suppose the response functions correspond to four populations that represent the cross-classification of age (two groups) by sex. You can use the FACTORS statement to identify these two factors and to name the effects in the model. The statements required to fit a main-effects model to these data are as follows:

```
proc catmod data=direct;
   response read b1-b4;
   model _f_=_response_;
   factors age 2, sex 2 / _response_=age sex;
run;
```

If you want to specify some nested-by-value effects, you can change the FACTORS statement to the following:

```
factors age $ 2, sex $ 2 /
       _response_=age sex(age='under 30') sex(age='30 & over')
        profile=('under 30'   male,
                 'under 30'   female,
                 '30 & over'  male,
                 '30 & over'  female);
```

If, by design or by chance, the study contains no male subjects under 30 years of age, then there are only three response functions, and you can specify a main-effects model as follows:

```
proc catmod data=direct;
   response read b2-b4;
   model _f_=_response_;
   factors age $ 2, sex $ 2 / _response_=age sex
          profile=('under 30'   female,
                   '30 & over'  male,
                   '30 & over'  female);
run;
```

When you specify two or more factors and omit the PROFILE= option, PROC CATMOD presumes that the response functions are ordered so that the levels of the rightmost factor change

most rapidly. For the preceding example, the order implied by the FACTORS statement is as follows:

Response Function	Dependent Variable	Age	Sex
1	b1	1	1
2	b2	1	2
3	b3	2	1
4	b4	2	2

For additional examples of how to use the FACTORS statement, see the section "Repeated Measures Analysis" on page 1150. All of the examples in that section are applicable, with the REPEATED statement replaced by the FACTORS statement.

LOGLIN Statement

LOGLIN *effects* < / *option* > **;**

The LOGLIN statement is used to define log-linear model effects. It can be used whenever the default response functions (generalized logits) are used.

In the LOGLIN statement, *effects* are design effects that contain dependent variables in the MODEL statement, including interaction, nested, and nested-by-value effects. You can use the bar (|) and at (@) operators as well. The following lists of effects are equivalent:

```
a b c a*b a*c b*c
```

and

```
a|b|c @2
```

When you use the LOGLIN statement, the keyword _RESPONSE_ should be specified in the MODEL statement. For further information about log-linear model analysis, see the section "Log-Linear Model Analysis" on page 1148.

You cannot specify the LOGLIN statement for an analysis that also contains the REPEATED or FACTORS statement since all of them specify the same information: how to partition the variation among the response functions within a population.

You can specify the following option in the LOGLIN statement after a slash.

TITLE=*'title'*
displays the *title* at the top of certain pages of output that correspond to this LOGLIN statement.

The following statements give an example of how to use the LOGLIN statement:

```
proc catmod;
   model a*b*c=_response_;
   loglin a|b|c @ 2;
run;
```

These statements yield a log-linear model analysis that contains all main effects and two-variable interactions. For more examples of log-linear model analysis, see the section "Log-Linear Model Analysis" on page 1148.

MODEL Statement

MODEL *response-effect=< design-effects >< / options >* **;**

PROC CATMOD requires a MODEL statement. You can specify the following in a MODEL statement:

response-effect can be either a single variable, a crossed effect with two or more variables joined by asterisks, or _F_. The _F_ specification indicates that the response functions and their estimated covariance matrix are to be read directly into the procedure (see the section "Inputting Response Functions and Covariances Directly" on page 1140 for details). The *response-effect* indicates the dependent variables that determine the response categories (the columns of the underlying contingency table).

design-effects specify potential sources of variation (such as main effects and interactions) in the model. These effects determine the number of model parameters, as well as the interpretation of such parameters. In addition, if there is no POPULATION statement, PROC CATMOD uses these variables to determine the populations (the rows of the underlying contingency table). When fitting the model, PROC CATMOD adjusts the independent effects in the model for all other independent effects in the model.

Design-effects can be any of those described in the section "Specification of Effects" on page 1142, or they can be defined by specifying the actual design matrix, enclosed in parentheses (see the section "Specifying the Design Matrix Directly" on page 1125). In addition, you can use the keyword _RESPONSE_ alone or as part of an effect. Effects cannot be nested within _RESPONSE_, so effects of the form A(_RESPONSE_) are invalid.

For more information, see the section "Log-Linear Model Analysis" on page 1148 and the section "Repeated Measures Analysis" on page 1150.

Some example MODEL statements are shown in the following table:

Example	Result
`model r=a b;`	main effects only
`model r=a b a*b;`	main effects with interaction
`model r=a b(a);`	nested effect
`model r=a|b;`	complete factorial
`model r=a b(a=1) b(a=2);`	nested-by-value effects
`model r*s=_response_;`	log-linear model
`model r*s=a _response_(a);`	nested repeated measurement factor
`model _f_=_response_;`	direct input of the response functions

The relationship between these specifications and the structure of the design matrix **X** is described in the section "Generation of the Design Matrix" on page 1153.

Table 28.4 summarizes the options available in the MODEL statement.

Table 28.4 MODEL Statement Options

Options	Task
Specify details of computation	
ML=	Generates maximum likelihood estimates
GLS	Generates weighted least squares estimates
WLS	
NOINT	Omits intercept term from the model
PARAM=	Specifies parameterization of classification variables
ADDCELL=	Adds a number to each cell frequency
AVERAGED	Averages main effects across response functions
EPSILON=	Specifies the convergence criterion for maximum likelihood
MAXITER=	Specifies the number of iterations for maximum likelihood
MISSING=	Specifies how missing cells are treated
ZERO=	Specifies how zero cells are treated
Request additional computation and tables	
ALPHA=	Significance level of confidence intervals
CLPARM	Wald confidence intervals of estimates
CORRB	Estimated correlation matrix of estimates
COV	Covariance matrix of response functions
COVB	Estimated covariance matrix of estimates
DESIGN	Design and _RESPONSE_ matrix
FREQ	Two-way frequency tables
ITPRINT	Iterations for maximum likelihood
ONEWAY	One-way frequency tables
PRED=	Predicted values
PREDICT	
PROB	Probability estimates
PROFILE	Population profiles
XPX	Crossproducts matrix
TITLE=	Title

Table 28.4 *continued*

Options	Task
Suppress output	
NODESIGN	Design matrix
NOPARM	Parameter estimates
NOPREDVAR	Variable levels
NOPROFILE	Population and response profiles
NORESPONSE	_RESPONSE_ matrix

The following list describes these options in alphabetical order.

ADDCELL=*number*

adds *number* to the frequency count in each cell, where *number* is any positive number. This option has no effect on maximum likelihood analysis; it is used only for weighted least squares analysis.

ALPHA=*number*

sets the significance level for the Wald confidence intervals for parameter estimates. The value must be between 0 and 1. The default value of 0.05 results in the calculation of a 95% confidence interval. This option has no effect unless the CLPARM option is also specified.

AVERAGED

specifies that dependent variable effects can be modeled and that independent variable main effects are averaged across the response functions in a population. For further information about the effect of using (or not using) the AVERAGED option, see the section "Generation of the Design Matrix" on page 1153. Direct input of the design matrix or specification of the _RESPONSE_ keyword in the MODEL statement automatically uses an AVERAGED model type.

CLPARM

produces Wald confidence limits for the parameter estimates. The confidence coefficient can be specified with the ALPHA= option.

CORRB

displays the estimated correlation matrix of the parameter estimates.

COV

displays $\mathbf{S}_i$, which is the covariance matrix of the response functions for each population.

COVB

displays the estimated covariance matrix of the parameter estimates.

DESIGN

displays the design matrix $\mathbf{X}$ for WLS and ML analyses, and also displays the _RESPONSE_ matrix for log-linear models. For further information, see the section "Generation of the Design Matrix" on page 1153.

EPSILON=*number*

specifies the convergence criterion for the maximum likelihood estimation of the parameters. The iterative estimation process stops when the proportional change in the log likelihood is less than *number*, or after the number of iterations specified by the MAXITER= option, whichever comes first. By default, EPSILON=1E–8.

FREQ

produces the two-way frequency table for the cross-classification of populations by responses.

ITPRINT

displays parameter estimates and other information at each iteration of a maximum likelihood analysis.

MAXITER=*number*

specifies the maximum number of iterations used for the maximum likelihood estimation of the parameters. By default, MAXITER=20.

ML < = NR | IPF< (*ipf-options*) > >

computes maximum likelihood estimates (MLE) by using either a Newton-Raphson algorithm (NR) or an iterative proportional fitting algorithm (IPF).

The option ML=NR (or simply ML) is available when you use generalized logits, and also when you perform binary logistic regression with logits, cumulative logits, or adjacent category logits. For generalized logits (the default response functions), ML=NR is the default estimation method.

The option ML=IPF is available for fitting a hierarchical log-linear model with one population (no independent variables and no population variables). The use of bar notation to express the log-linear effects guarantees that the model is hierarchical (the presence of any interaction term in the model requires the presence of all its lower-order terms). If your table is *incomplete* (that is, your table has a zero or missing entry in at least one cell), then all missing cells and all cells with zero weight are treated as structural zeros by default; this behavior can be modified with the ZERO= and MISSING= options in the MODEL statement.

You can control the convergence of the two algorithms with the EPSILON= and MAXITER= options in the MODEL statement. You can select the convergence criterion for the IPF algorithm with the CONVCRIT= option.

NOTE: The RESTRICT statement is not available with the ML=IPF option.

You can specify the following *ipf-options* within parentheses after the ML=IPF option.

CONVCRIT=*keyword* specifies the method that determines when convergence of the IPF algorithm occurs. You can specify one of the following *keywords*:

CELL termination requires the maximum absolute difference between consecutive cell estimates to be less than 0.001 (or the value of the EPSILON= option, if specified).

LOGL termination requires the relative difference between consecutive estimates of the log likelihood to be less than 1E–8 (or the value of the EPSILON= option, if specified). This is the default.

MARGIN termination requires the maximum absolute difference between consecutive margin estimates to be less than 0.001 (or the value of the EPSILON= option, if specified).

DF=*keyword* specifies the method used to compute the degrees of freedom for the goodness-of-fit G^2 test (labeled "Likelihood Ratio" in the "Estimates" table).

For a *complete* table (a table having nonzero entries in every cell), the degrees of freedom are calculated as the number of cells in the table (n_c) minus the number of independent parameters specified in the model (n_p). For incomplete tables, these degrees of freedom can be adjusted by the number of fitted zeros (n_z, which includes the number of structural zeros) and the number of nonestimable parameters due to the zeros (n_n). If you are analyzing an incomplete table, you should verify that the degrees of freedom are correct.

You can specify one of the following *keywords*:

UNADJ computes the unadjusted degrees of freedom as $n_c - n_p$. These are the same degrees of freedom you would get if all cells in the table were positive.

ADJ computes the degrees of freedom as $(n_c - n_p) - (n_z - n_n)$ (Bishop, Fienberg, and Holland 1975), which adjusts for fitted zeros and nonestimable parameters. This is the default, and for complete tables it gives the same results as the UNADJ option.

ADJEST computes the degrees of freedom as $(n_c - n_p) - n_z$, which adjusts for fitted zeros only. This gives a lower bound on the true degrees of freedom.

PARM computes parameter estimates, generates the "ANOVA," "Parameter Estimates," and "Predicted Values of Response Functions" tables, and includes the predicted standard errors in the "Predicted Values of Frequencies and Probabilities" tables.

When you specify the PARM option, the algorithm used to obtain the maximum likelihood parameter estimates is weighted least squares on the IPF-predicted frequencies. This algorithm can be much faster than the Newton-Raphson algorithm that is used if you specify the ML=NR option. In the resulting ANOVA table, the likelihood ratio is computed from the initial IPF fit while the degrees of freedom are generated from the WLS analysis; the DF= option can override this. Also, the initial response function, which the WLS method usually computes from the raw data, is computed from the IPF-predicted frequencies.

If there are any zero marginals in the configurations that define the model, then there are predicted cell frequencies of zero and WLS cannot be used to compute the estimates. In this case, PROC CATMOD automatically changes the algorithm from ML=IPF to ML=NR and prints a note in the log.

MISSING=*keyword*

MISS=*keyword*

specifies whether a missing cell is treated as a sampling or structural zero.

Structural zero cells are removed from the analysis since their expected values are zero, while sampling zero cells can have nonzero expected value and might be estimable. For a single population, the missing cells are treated as structural zeros by default. For multiple populations, as long as some population has a nonzero count for a given population and response profile, the missing values are treated as sampling zeros by default.

The following table displays the available *keywords* and summarizes how PROC CATMOD treats missing values for one or more populations:

MISSING=	One Population	Multiple Populations
STRUCTURAL (default)	structural zeros	sampling zeros
SAMP | SAMPLING	sampling zeros	sampling zeros
value	sets missing weights and cells to *value*	sets missing weights and cells to *value*

NODESIGN

suppresses the display of the design matrix **X** when the DESIGN option is also specified. This enables you to display only the _RESPONSE_ matrix for log-linear models.

NOINT

suppresses the intercept term in the model.

NOITER

suppresses the display of parameter estimates and other information at each iteration of a maximum likelihood analysis.

NOPARM

suppresses the display of the estimated parameters and the statistics for testing that each parameter is zero.

NOPREDVAR

suppresses the display of the variable levels in tables requested with the PRED= option and in the "Estimates" table. Population profiles are replaced with the sample number, classification variable levels are suppressed, and response profiles are replaced with a function number.

NOPRINT

suppresses the normal display of results. The NOPRINT option is useful when you only want to create output data sets with the OUT= or OUTEST= option in the RESPONSE statement. A NOPRINT option is also available in the PROC CATMOD statement. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

NOPROFILE

suppresses the display of the population profiles and the response profiles.

NORESPONSE

suppresses the display of the _RESPONSE_ matrix for log-linear models when the DESIGN option is also specified. This enables you to display only the design matrix for log-linear models.

ONEWAY

produces a one-way table of frequencies for each variable used in the analysis. This table is useful in determining the order of the observed levels for each variable.

PARAM=EFFECT | REFERENCE

specifies the parameterization method for the classification variable or variables. The default is PARAM=EFFECT. Both the effect and reference parameterizations are full rank. See the section "Generation of the Design Matrix" on page 1153 for further details.

PREDICT

PRED=FREQ | PROB

displays the observed and predicted values of the response functions for each population, together with their standard errors and the residuals (observed minus predicted). In addition, if the response functions are the standard ones (generalized logits), then the PRED=FREQ option specifies the computation and display of predicted cell frequencies, while PRED=PROB (or just PREDICT) specifies the computation and display of predicted cell probabilities.

The OUT= data set always contains the predicted probabilities. If the response functions are the generalized logits, the predicted cell probabilities are output unless the option PRED=FREQ is specified, in which case the predicted cell frequencies are output.

PROB

produces the two-way table of probability estimates for the cross-classification of populations by responses. These estimates sum to one across the response categories for each population.

PROFILE

displays all of the population profiles. If you have more than 60 populations, then by default only the first 40 profiles are displayed; the PROFILE option overrides this default behavior.

TITLE='*title***'**

displays the *title* at the top of certain pages of output that correspond to this MODEL statement.

WLS

GLS

computes weighted least squares estimates. This type of estimation is also called generalized least squares estimation. For response functions other than the default (of generalized logits), WLS is the default estimation method.

XPX

displays $\mathbf{X}'\mathbf{S}^{-1}\mathbf{X}$, the crossproducts matrix for the normal equations.

ZERO=*keyword*

specifies whether a nonmissing cell with zero weight in the data set is treated as a sampling or structural zero.

Structural zero cells are removed from the analysis since their expected values are zero, while sampling zero cells have nonzero expected value and might be estimable. For a single population, the zero cells are treated as structural zeros by default; with multiple populations, as long as some population has a nonzero count for a given population and response profile, the zeros are treated as sampling zeros by default.

The following table displays the available *keywords* and summarizes how PROC CATMOD treats zeros for one or more populations:

ZERO=	One Population	Multiple Populations
STRUCTURAL (default)	structural zeros	sampling zeros
SAMP | SAMPLING	sampling zeros	sampling zeros
value	sets zero weights to *value*	sets zero weights to *value*

Specifying the Design Matrix Directly

If you specify the design matrix directly, adjacent rows of the matrix must be separated by a comma, and the matrix must have $q \times s$ rows, where s is the number of populations and q is the number of response functions per population. The first q rows correspond to the response functions for the first population, the second set of q rows corresponds to the functions for the second population, and so forth. The following is an example of using direct specification of the design matrix.

```
proc catmod;
   model R=(1 0,
            1 1,
            1 2,
            1 3);
run;
```

These statements are appropriate for the case of one population and for R with five levels (generating four response functions), so that $4 \times 1 = 4$. These statements are also appropriate for a situation with two populations and two response functions per population, giving $2 \times 2 = 4$ rows of the design matrix. (To accommodate more than one population, the POPULATION statement is needed.)

When you input the design matrix directly, you also have the option of specifying that any subsets of the parameters be tested for equality to zero. Indicate each subset by specifying the appropriate column numbers of the design matrix, followed by an equal sign and a label (24 characters or less, in quotes) that describes the subset. Adjacent subsets are separated by a comma, and the entire specification is enclosed in parentheses and placed after the design matrix. For example:

```
proc catmod;
   population Group Time;
   model R=(1  1  0  0,
            1  1  0  1,
            1  1  0  2,
            1  0  1  0,
            1  0  1  1,
            1  0  1  2,
            1 -1 -1  0,
            1 -1 -1  1,
            1 -1 -1  2) (1  ='Intercept',
                         2 3='Group main effect',
                         4  ='Linear effect of Time');
run;
```

The preceding statements are appropriate when Group and Time each have three levels and R is dichotomous. The POPULATION statement produces nine populations, and $q = 1$ (since R is dichotomous), so $q \times s = 1 \times 9 = 9$.

If you input the design matrix directly but do not specify any subsets of the parameters to be tested, then PROC CATMOD tests the effect of MODEL | MEAN, which represents the significance of the model beyond what is explained by an overall mean. For the previous example, the MODEL | MEAN effect is the same as that obtained by specifying the following at the end of the MODEL statement:

```
(2 3 4='model|mean');
```

POPULATION Statement

POPULATION *variables* **;**

The POPULATION statement specifies that populations are to be based only on cross-classifications of the specified *variables*. If you do not specify the POPULATION statement, then populations are based only on cross-classifications of the independent variables in the MODEL statement.

The POPULATION statement has two major uses:

- When you enter the design matrix directly, there are no independent variables in the MODEL statement; therefore, the POPULATION statement is the only way to produce more than one population.
- When you fit a reduced model, the POPULATION statement might be necessary if you want to form the same number of populations as there are for the saturated model.

To illustrate the first use, suppose you specify the following statements:

```
data one;
   input A $ B $ wt @@;
   datalines;
yes yes 23   yes no 31   no yes 47   no no 50
;
proc catmod;
   weight wt;
   population B;
   model A=(1 0,
            1 1);
run;
```

Since the dependent variable A has two levels, there is one response function per population. Since the variable B has two levels, there are two populations. The MODEL statement is valid since the number of rows in the design matrix (2) is the same as the total number of response functions. If the POPULATION statement is omitted, there would be only one population and one response function, and the MODEL statement would be invalid.

To illustrate the second use, suppose you specify the following statements:

```
data two;
   input A $ B $ Y wt @@;
   datalines;
yes  yes  1  23       yes  yes  2  63
yes  no   1  31       yes  no   2  70
no   yes  1  47       no   yes  2  80
no   no   1  50       no   no   2  84
;

proc catmod;
   weight wt;
   model Y=A B A*B / wls;
run;
```

These statements form four populations and produce the following design matrix and analysis of variance table:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	48.10	<.0001
A	1	3.47	0.0625
B	1	0.25	0.6186
A*B	1	0.19	0.6638
Residual	0		

Since the B and A*B effects are nonsignificant ($p > 0.10$), fit the reduced model that contains only the A effect:

```
proc catmod;
   weight wt;
   model Y=A / wls;
run;
```

Now only two populations are formed, and the design matrix and the analysis of variance table are as follows:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	47.94	<.0001
A	1	3.33	0.0678
Residual	0		

However, you can form four populations by adding the POPULATION statement to the analysis:

```
proc catmod;
   weight wt;
   population A B;
   model Y=A / wls;
run;
```

The design matrix and the analysis of variance table resulting from these statements are as follows:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 1 & -1 \\ 1 & -1 \end{bmatrix}$$

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	47.76	<.0001
A	1	3.30	0.0694
Residual	2	0.35	0.8374

The advantage of the latter analysis is that it retains four populations for the reduced model, thereby creating a built-in goodness-of-fit test: the residual chi-square. Such a test is important because the cumulative (or joint) effect of deleting two or more effects from the model can be significant, even if the individual effects are not.

The resulting differences between the two analyses are due to the fact that the latter analysis uses pure weighted least squares estimates with respect to the four populations that are actually sampled. The former analysis pools populations and therefore uses parameter estimates that can be regarded as weighted least squares estimates of maximum likelihood predicted cell frequencies. In any case, the estimation methods are asymptotically equivalent; therefore, the results are very similar. If you specify the ML option (instead of the WLS option) in the preceding MODEL statements, then the parameter estimates are identical for the two analyses.

CAUTION: If your model has different covariate profiles within any population, then the first profile is used in the analysis.

REPEATED Statement

REPEATED *factor-description*<*, . . . , factor-description*><*/ options*> **;**

where a *factor-description* is defined as follows:

factor-name < *$*>< *levels* >

and *factor-description*s are separated from each other by a comma. The $ is required for character-valued factors. The value of *levels* provides the number of levels of the repeated measurement factor identified by a given *factor-name*. For only one repeated measurement factor, *levels* is optional; for two or more repeated measurement factors, it is required. The REPEATED statement incorporates repeated measurement factors into the model. You can use this statement whenever there is more than one dependent variable and the keyword _RESPONSE_ is specified in the MODEL statement. If the dependent variables correspond to one or more repeated measurement factors, you can use the REPEATED statement to define _RESPONSE_ in terms of those factors. You can specify the name, type, and number of levels of each factor, as well as the identification of each level.

You cannot specify the REPEATED statement for an analysis that also contains the FACTORS or LOGLIN statement since all of them specify the same information: how to partition the variation among the response functions within a population.

You can specify the following terms in the REPEATED statement:

factor-name names a repeated measurement factor that corresponds to two or more response functions. This name must be a valid SAS variable name, and it should not be the same as the name of a variable that already exists in the data set being analyzed.

$ indicates that the factor is character-valued. If the $ is omitted, then the CATMOD procedure assumes that the factor is numeric. The type of the factor is relevant only when you use the PROFILE= option or when the _RESPONSE_= option specifies nested-by-value effects.

levels specifies the number of levels of the corresponding repeated measurement factor. If there is only one such factor and the number is omitted, then PROC CATMOD assumes that the number of levels is equal to the number of response functions per population (q). Unless you specify the PROFILE= option, the number q must either be equal to or be a multiple of the product of the number of levels of all the factors.

You can specify the following options in the REPEATED statement after a slash.

PROFILE=(*matrix***)**

specifies the values assumed by the factors for each response function. There should be one column for each factor, and the values in a given column should match the type (character or numeric) of the corresponding factor. Character values are restricted to 16 characters or less. If there are q response functions per population, then the matrix must have i rows, where q must either be equal to or be a multiple of i. Adjacent rows of the matrix should be separated by a comma.

The values in the PROFILE matrix are useful for specifying models in those situations where the study design is not a full factorial with respect to the factors. They can also be used

to specify nested-with-value effects in the _RESPONSE_= option. If you specify character values in both the PROFILE= option and the _RESPONSE_= option, then the values must match with respect to whether or not they are enclosed in quotes (that is, they must be enclosed in quotes in both places or in neither place).

RESPONSE=*effects*

specifies design effects. The variables named in the effects must be *factor-names* that appear in the REPEATED statement. If the _RESPONSE_= option is omitted, then PROC CATMOD builds a full factorial _RESPONSE_ effect with respect to the repeated measurement factors. For example, the following two statements are equivalent in that they produce the same parameter estimates:

```
repeated Time 2, Treatment 2;
repeated Time 2, Treatment 2 / _response_=Time|Treatment;
```

However, the second statement produces tests of the Time, Treatment, and Time*Treatment effects in the "Analysis of Variance" table, whereas the first statement produces a single test for the combined effects in _RESPONSE_.

TITLE=*'title'*

displays the *title* at the top of certain pages of output that correspond to this REPEATED statement.

For further information and numerous examples of the REPEATED statement, see the section "Repeated Measures Analysis" on page 1150.

RESPONSE Statement

RESPONSE < *function* >< / *options* > **;**

The RESPONSE statement specifies functions of the response probabilities. The procedure models these response functions as linear combinations of the parameters.

By default, PROC CATMOD uses the standard response functions (generalized logits, which are explained in detail in the section "Understanding the Standard Response Functions" on page 1137). With these standard response functions, the default estimation method is maximum likelihood, but you can use the WLS option in the MODEL statement to request weighted least squares estimation. With other response functions (specified in the RESPONSE statement), the default (and only) estimation method is weighted least squares.

You can specify more than one RESPONSE statement, in which case each RESPONSE statement produces a separate analysis. If the computed response functions for any population are linearly dependent (yielding a singular covariance matrix), then PROC CATMOD displays an error message and stops processing. See the section "Cautions" on page 1163 for methods of dealing with this.

The *function* specification can be any of the items in the following list. For an example of response functions generated and formulas for q (the number of response functions), see the section "More on Response Functions" on page 1132.

ALOGIT

ALOGITS

specifies response functions as adjacent-category logits of the marginal probabilities for each of the dependent variables. For each dependent variable, the response functions are a set of linearly independent adjacent-category logits, obtained by taking the logarithms of the ratios of two probabilities. The denominator of the kth ratio is the marginal probability corresponding to the kth level of the variable, and the numerator is the marginal probability corresponding to the $(k+1)$th level. If a dependent variable has two levels, then the adjacent-category logit is the negative of the generalized logit.

CLOGIT

CLOGITS

specifies that the response functions are cumulative logits of the marginal probabilities for each of the dependent variables. For each dependent variable, the response functions are a set of linearly independent cumulative logits, obtained by taking the logarithms of the ratios of two probabilities. The denominator of the kth ratio is the cumulative probability, c_k, corresponding to the kth level of the variable, and the numerator is $1-c_k$ (Agresti 1984, 113–114). If a dependent variable has two levels, then PROC CATMOD computes its cumulative logit as the negative of its generalized logit. You should use cumulative logits only when the dependent variables are ordinally scaled.

JOINT

specifies that the response functions are the joint response probabilities. A linearly independent set is created by deleting the last response probability. For the case of one dependent variable, the JOINT and MARGINALS specifications are equivalent.

LOGIT

LOGITS

specifies that the response functions are generalized logits of the marginal probabilities for each of the dependent variables. For each dependent variable, the response functions are a set of linearly independent generalized logits, obtained by taking the logarithms of the ratios of two probabilities. The denominator of each ratio is the marginal probability corresponding to the last observed level of the variable, and the numerators are the marginal probabilities corresponding to each of the other levels. If there is one dependent variable, then specifying LOGIT is equivalent to using the standard response functions.

MARGINAL

MARGINALS

specifies that the response functions are marginal probabilities for each of the dependent variables in the MODEL statement. For each dependent variable, the response functions are a set of linearly independent marginals, obtained by deleting the marginal probability corresponding to the last level.

MEAN

MEANS

specifies that the response functions are the means of the dependent variables in the MODEL statement. This specification requires that all of the dependent variables be numeric.

READ *variables*
: specifies that the response functions and their covariance matrix are to be read directly from the input data set with one response function for each variable named. See the section "Inputting Response Functions and Covariances Directly" on page 1140 for more information.

transformation
: specifies response functions that can be expressed by using successive applications of the four operations: **LOG**, **EXP**, $*$ matrix literal, or $+$ matrix literal. The operations are described in detail in the section "Using a Transformation to Specify Response Functions" on page 1135.

You can specify the following options in the RESPONSE statement after a slash.

OUT=*SAS-data-set*
: produces a SAS data set that contains, for each population, the observed and predicted values of the response functions, their standard errors, and the residuals. Moreover, if you use the standard response functions, the data set also includes observed and predicted values of the cell frequencies or the cell probabilities. For further information, see the section "Output Data Sets" on page 1144.

OUTEST=*SAS-data-set*
: produces a SAS data set that contains the estimated parameter vector and its estimated covariance matrix. For further information, see the section "Output Data Sets" on page 1144.

TITLE=*'title'*
: displays the *title* at the top of certain pages of output that correspond to this RESPONSE statement.

More on Response Functions

Suppose the dependent variable A has three levels and is the only *response-effect* in the MODEL statement. The following table shows the proportions upon which the response functions are defined:

Value of A:	1	2	3
Proportions:	p_1	p_2	p_3

Note that $\sum_j p_j = 1$. The following table shows the response functions generated for each population:

Function Specification	Value of q	Response Function
none*	2	$\ln\left(\frac{p_1}{p_3}\right),\ \ln\left(\frac{p_2}{p_3}\right)$
ALOGITS	2	$\ln\left(\frac{p_2}{p_1}\right),\ \ln\left(\frac{p_3}{p_2}\right)$
CLOGITS	2	$\ln\left(\frac{1-p_1}{p_1}\right),\ \ln\left(\frac{1-(p_1+p_2)}{p_1+p_2}\right)$
JOINT	2	$p_1,\ p_2$
LOGITS	2	$\ln\left(\frac{p_1}{p_3}\right),\ \ln\left(\frac{p_2}{p_3}\right)$
MARGINAL	2	$p_1,\ p_2$
MEAN	1	$1p_1 + 2p_2 + 3p_3$

*Without a function specification, the default response functions are generalized logits.

Now, suppose the dependent variables A and B each have three levels (valued 1, 2, and 3 each) and the *response-effect* in the MODEL statement is A*B. The following table shows the proportions upon which the response functions are defined:

Value of A:	1	1	1	2	2	2	3	3	3
Value of B:	1	2	3	1	2	3	1	2	3
Proportions:	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9

The marginal totals for the preceding table are defined as follows:

$$p_{1\cdot} = p_1 + p_2 + p_3 \qquad p_{\cdot 1} = p_1 + p_4 + p_7$$
$$p_{2\cdot} = p_4 + p_5 + p_6 \qquad p_{\cdot 2} = p_2 + p_5 + p_8$$
$$p_{3\cdot} = p_7 + p_8 + p_9 \qquad p_{\cdot 3} = p_3 + p_6 + p_9$$

where $\sum_j p_j = 1$. The following table shows the response functions generated for each population:

Function Specification	Value of q	Response Function
none*	8	$\ln\left(\frac{p_1}{p_9}\right),\ \ln\left(\frac{p_2}{p_9}\right),\ \ln\left(\frac{p_3}{p_9}\right),\ldots,\ \ln\left(\frac{p_8}{p_9}\right)$
ALOGITS	4	$\ln\left(\frac{p_{2\cdot}}{p_{1\cdot}}\right),\ \ln\left(\frac{p_{3\cdot}}{p_{2\cdot}}\right),\ \ln\left(\frac{p_{\cdot 2}}{p_{\cdot 1}}\right),\ \ln\left(\frac{p_{\cdot 3}}{p_{\cdot 2}}\right)$
CLOGITS	4	$\ln\left(\frac{1-p_{1\cdot}}{p_{1\cdot}}\right),\ \ln\left(\frac{1-(p_{1\cdot}+p_{2\cdot})}{p_{1\cdot}+p_{2\cdot}}\right),\ \ln\left(\frac{1-p_{\cdot 1}}{p_{\cdot 1}}\right),\ \ln\left(\frac{1-(p_{\cdot 1}+p_{\cdot 2})}{p_{\cdot 1}+p_{\cdot 2}}\right)$
JOINT	8	$p_1,\ p_2,\ p_3,\ p_4,\ p_5,\ p_6,\ p_7,\ p_8$
LOGITS	4	$\ln\left(\frac{p_{1\cdot}}{p_{3\cdot}}\right),\ \ln\left(\frac{p_{2\cdot}}{p_{3\cdot}}\right),\ \ln\left(\frac{p_{\cdot 1}}{p_{\cdot 3}}\right),\ \ln\left(\frac{p_{\cdot 2}}{p_{\cdot 3}}\right)$
MARGINAL	4	$p_{1\cdot},\ p_{2\cdot},\ p_{\cdot 1},\ p_{\cdot 2}$
MEAN	2	$1p_{1\cdot} + 2p_{2\cdot} + 3p_{3\cdot},\ 1p_{\cdot 1} + 2p_{\cdot 2} + 3p_{\cdot 3}$

* Without a function specification, the default response functions are generalized logits.

The READ and *transformation* function specifications are not shown in the preceding table. For these two situations, there is not a general response function; the response functions that are generated depend on what you specify.

Another important aspect of the function specification is the number of response functions generated per population, q. Let m_i represent the number of levels for the ith dependent variable in the MODEL statement, and let d represent the number of dependent variables in the MODEL statement. Then, if the function specification is ALOGITS, CLOGITS, LOGITS, or MARGINALS, the number of response functions is

$$q = \sum_{i=1}^{d} (m_i - 1)$$

If the function specification is JOINT or the default (generalized logits), the number of response functions per population is

$$q = r - 1$$

where r is the number of response profiles. If every possible cross-classification of the dependent variables is observed in the samples, then

$$r = \prod_{i=1}^{d} m_i$$

Otherwise, r is the number of cross-classifications actually observed.

If the function specification is MEANS, the number of response functions per population is $q = d$.

Response Statement Examples

Some example response statements are shown in the following table:

Example	Result
`response marginals;`	marginals for each dependent variable
`response means;`	the mean of each dependent variable
`response logits;`	generalized logits of the marginal probabilities
`response clogits;`	cumulative logits of the marginal probabilities
`response alogits;`	adjacent-category logits of the marginal probabilities
`response joint;`	the joint probabilities
`response 1 -1 log;`	the logit
`response;`	generalized logits
`response 1 2 3;`	the mean score, with scores of 1, 2, and 3 corresponding to the three response levels
`response read b1-b4;`	four response functions and their covariance matrix, read directly from the input data set

Using a Transformation to Specify Response Functions

If you specify a *transformation*, it is applied to the vector that contains the sample proportions in each population. The *transformation* can be any combination of the following four operations:

Operation	Specification
linear combination	$*$ matrix literal
linear combination	matrix literal
logarithm	**LOG**
exponential	**EXP**
adding constant	$+$ matrix literal

If more than one operation is specified, then PROC CATMOD applies the operations consecutively from right to left.

A matrix literal is a matrix of numbers with each row of the matrix separated from the next by a comma. If you specify a linear combination, in most cases the $*$ is not needed. The following statement defines the response function $p_1 + 1$. The $*$ is needed to separate the two matrix literals '1' and '1 0'.

```
response + 1 * 1 0;
```

The **LOG** of a vector transforms each element of the vector into its natural logarithm; the **EXP** of a vector transforms each element into its exponential function (antilogarithm).

In order to specify a linear response function for data that have $r = 3$ response categories, you can specify either of the following RESPONSE statements:

```
response  * 1 0 0 , 0 1 0;
response    1 0 0 , 0 1 0;
```

The matrix literal in the preceding statements specifies a 2×3 matrix, which is applied to each population as follows:

$$\begin{bmatrix} F_1 \\ F_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} * \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

where p_1, p_2, and p_3 are sample proportions for the three response categories in a population, and F_1 and F_2 are the two response functions computed for that population. Therefore, this response function sets $F1 = p_1$ and $F2 = p_2$ in each population.

As another example of the linear response function, suppose you have two dependent variables corresponding to two observers who evaluate the same subjects. If the observers grade on the same three-point scale and if all nine possible responses are observed, then the following RESPONSE statement would compute the probability that the observers agree on their assessments:

```
response 1 0 0 0 1 0 0 0 1;
```

This response function is then computed as

$$F = p_{11} + p_{22} + p_{33} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} p_{11} \\ p_{12} \\ p_{13} \\ p_{21} \\ p_{22} \\ p_{23} \\ p_{31} \\ p_{32} \\ p_{33} \end{bmatrix}$$

where p_{ij} denotes the probability that a subject gets a grade of i from the first observer and j from the second observer.

If the function is a compound function, requiring more than one operation to specify it, then the operations should be listed so that the first operation to be applied is on the right and the last operation to be applied is on the left. For example, if there are two response levels, you can have the following response function:

```
response 1 -1 log;
```

This is equivalent to the matrix expression

$$F = \begin{bmatrix} 1 & -1 \end{bmatrix} * \begin{bmatrix} \log(p_1) \\ \log(p_2) \end{bmatrix} = \log(p_1) - \log(p_2) = \log\left(\frac{p_1}{p_2}\right)$$

which is the logit response function since $p_2 = 1 - p_1$ when there are only two response levels.

The following statement specifies another example of a compound response function:

```
response exp 1 -1 * 1 0 0 1, 0 1 1 0 log;
```

This is equivalent to the matrix expression

$$F = \mathbf{EXP}(\mathbf{A} * \mathbf{B} * \mathbf{LOG}(\mathbf{P}))$$

where $\mathbf{P}$ is the vector of sample proportions for some population,

$$\mathbf{A} = \begin{bmatrix} 1 & -1 \end{bmatrix} \text{ and } \mathbf{B} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

If the four responses are based on two dependent variables, each with two levels, then the function can also be written as

$$F = \frac{p_{11}p_{22}}{p_{12}p_{21}}$$

which is the odds (crossproduct) ratio for a 2×2 table.

Understanding the Standard Response Functions

If no RESPONSE statement is specified, PROC CATMOD computes the standard response functions, which contrast the log of each response probability with the log of the probability for the last response category. If there are r response categories, then there are $r - 1$ standard response functions. For example, if there are four response categories, using no RESPONSE statement is equivalent to specifying the following:

```
response  1 0 0 -1,
          0 1 0 -1,
          0 0 1 -1  log;
```

This results in three response functions:

$$F = \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} \log(p_1/p_4) \\ \log(p_2/p_4) \\ \log(p_3/p_4) \end{bmatrix}$$

If there are only two response levels, the resulting response function would be a logit, which is why the standard response functions are called generalized logits. They are useful in dealing with the log-linear model:

$$\boldsymbol{\pi} = \mathbf{EXP}(\mathbf{X}\boldsymbol{\beta})$$

If $\mathbf{C}$ denotes the matrix in the preceding RESPONSE statement, then because of the restriction that the probabilities sum to 1, it follows that an equivalent model is

$$\mathbf{C} * \mathbf{LOG}(\boldsymbol{\pi}) = (\mathbf{CX})\boldsymbol{\beta}$$

But $\mathbf{C} * \mathbf{LOG}(\mathbf{P})$ is simply the vector of standard response functions. Thus, fitting a log-linear model on the cell probabilities is equivalent to fitting a linear model on the generalized logits.

RESTRICT Statement

RESTRICT *parameter=value* < . . . *parameter=value* > **;**

where *parameter* is the letter B followed by a number; for example, B3 specifies the third parameter in the model. The *value* is the value to which the parameter is restricted. The RESTRICT statement restricts values of parameters to the values you specify, so that the estimation of the remaining parameters is subject to these restrictions. Consider the following statement:

```
restrict b1=1 b4=0 b6=0;
```

This restricts the values of three parameters. The first parameter is set to 1, and the fourth and sixth parameters are set to zero.

The RESTRICT statement is interactive. A new RESTRICT statement replaces any previous ones. In addition, if you submit two or more MODEL, LOGLIN, FACTORS, or REPEATED statements, then the subsequent occurrences of these statements also delete the previous RESTRICT statement.

WEIGHT Statement

WEIGHT *variable* **;**

You can use a WEIGHT statement to refer to a variable containing the cell frequencies, which do not need to be integers. The WEIGHT statement lets you use summary data sets containing a count variable. See the section "Input Data Sets" on page 1138 for further information about the WEIGHT statement.

Details: CATMOD Procedure

Missing Values

Observations with missing values for any variable listed in the MODEL or POPULATION statement are omitted from the analysis.

If the WEIGHT variable for an observation has a missing value, the observation is by default omitted from the analysis. You can modify this behavior by specifying the MISSING= option in the MODEL statement. The option MISSING=*value* sets all missing weights to *value* and all missing cells to *value*. The option MISSING=SAMPLING causes all missing cells in a contingency table to be treated as sampling zeros.

Any observation with nonpositive weight is also, by default, omitted from the analysis. If it has zero weight, then you can specify the ZERO= option in the MODEL statement.

Input Data Sets

Data to be analyzed by PROC CATMOD must be in a SAS data set containing one of the following:

- raw data values (variable values for every subject)
- frequency counts and the corresponding variable values
- response function values and their covariance matrix

If you specify a WEIGHT statement, then PROC CATMOD uses the values of the WEIGHT variable as the frequency counts. If the READ function is specified in the RESPONSE statement, then the procedure expects the input data set to contain the values of response functions and their covariance matrix. Otherwise, PROC CATMOD assumes that the SAS data set contains raw data values.

Raw Data Values

If you use raw data, PROC CATMOD first counts the number of observations having each combination of values for all variables specified in the MODEL or POPULATION statement. For example, suppose the variables A and B each take on the values 1 and 2, and their frequencies can be represented as follows:

		A	
		1	**2**
B	**1**	2	1
	2	3	1

The SAS data set Raw containing the raw data might be as follows:

Observation	A	B
1	1	1
2	1	1
3	1	2
4	1	2
5	1	2
6	2	1
7	2	2

And the statements for PROC CATMOD are as follows:

```
proc catmod data=Raw;
   model A=B;
run;
```

For discussions of how to handle structural and random zeros with raw data as input data, see the section "Zero Frequencies" on page 1164 and Example 28.5.

Frequency Counts

If your data set contains frequency counts, then use the WEIGHT statement to specify the variable containing the frequencies. For example, you could create and analyze the Summary data set as follows:

```
data Summary;
   input A B Count;
   datalines;
1 1 2
1 2 3
2 1 1
2 2 1
;

proc catmod data=Summary;
   weight Count;
   model A=B;
run;
```

The data set Summary can also be created from the data set Raw by using the FREQ procedure:

```
proc freq data=Raw;
   tables A*B / out=Summary;
run;
```

Inputting Response Functions and Covariances Directly

If you want to read in the response functions and their covariance matrix, rather than have PROC CATMOD compute them, create a TYPE=EST data set. In addition to having one variable name for each function, the data set should have two additional variables: _TYPE_ and _NAME_, both character variables of length 8. The variable _TYPE_ should have the value 'PARMS' when the observation contains the response functions; it should have the value 'COV' when the observation contains elements of the covariance matrix of the response functions. The variable _NAME_ is used only when _TYPE_=COV, in which case it should contain the name of the variable that has its covariance elements stored in that observation. In the following data set, for example, the covariance between the second and fourth response functions is 0.000102:

```
data direct(type=est);
   input b1-b4 _type_ $ _name_ $8.;
   datalines;
0.590463   0.384720   0.273269   0.136458   PARMS      .
0.001690   0.000911   0.000474   0.000432   COV        B1
0.000911   0.001823   0.000031   0.000102   COV        B2
0.000474   0.000031   0.001056   0.000477   COV        B3
0.000432   0.000102   0.000477   0.000396   COV        B4
;
```

In order to tell PROC CATMOD that the input data set contains the values of response functions and their covariance matrix, do the following:

- specify the READ function in the RESPONSE statement
- specify _F_ as the dependent variable in the MODEL statement

For example, suppose the response functions correspond to four populations that represent the cross-classification of two age groups by two race groups. You can use the FACTORS statement to identify these two factors and to name the effects in the model. The following statements are required to fit a main-effects model to these data:

```
proc catmod data=direct;
   response read b1-b4;
   model _f_=_response_;
   factors age 2, race 2 / _response_=age race;
run;
```

Ordering of Populations and Responses

By default, populations and responses are sorted in standard SAS order as follows:

- alphabetical order for character variables
- increasing numeric order for numeric variables

Suppose you specify the following statements:

```
data one;
   length A B $ 6;
   input A $ B $ wt @@;
   datalines;
low       low  23  low    medium  31 low    high  38
medium    low  40  medium medium  42 medium high  50
high      low  52  high   medium  54 high   high  61
;

proc catmod;
   weight wt;
   model A=B / oneway;
run;
```

The ordering of populations and responses corresponds to the alphabetical order of the levels of the character variables. You can specify the ONEWAY option to display the ordering of the variables, while the "Population Profiles" and "Response Profiles" tables display the ordering of the populations and the responses, respectively.

Population Profiles		Response Profiles	
Sample	**B**	**Response**	**A**
1	high	1	high
2	low	2	low
3	medium	3	medium

In this example, if you want to have the levels ordered in the natural order of 'low,' 'medium,' 'high,' you can specify the ORDER=DATA option:

```
proc catmod order=data;
   weight wt;
   model a=b / oneway;
run;
```

The resulting ordering of populations and responses is as follows:

Population Profiles		Response Profiles	
Sample	B	Response	A
1	low	1	low
2	medium	2	medium
3	high	3	high

You can use the ORDER=DATA option to ensure that populations and responses are ordered in a specific way. But since this also affects the definitions and the ordering of the parameters, you must exercise caution when using the _RESPONSE_ effect, the CONTRAST statement, or direct input of the design matrix.

An alternative method of ensuring that populations and responses are ordered in a specific way is to assign a format to your variables and specify the ORDER=FORMATTED option. The levels are then ordered according to their formatted values.

Another method is to replace any character variables with numeric variables and to assign formatted values such as 'yes' and 'no' to the numeric levels. Since ORDER=INTERNAL is the default ordering, PROC CATMOD orders the populations and responses according to the numeric values but displays the formatted values.

Specification of Effects

By default, the CATMOD procedure treats all variables as classification variables. As a result, there is no CLASS statement in PROC CATMOD. The values of a classification variable can be numeric or character. PROC CATMOD builds a set of effects-coded variables to represent the levels of the classification variable and then uses these to fit the model (for details, see the section "Generation of the Design Matrix" on page 1153). You can modify the default by using the DIRECT statement to treat numeric independent continuous variables as continuous variables. The classification variables, combinations of classification variables, and continuous variables are then used in fitting linear models to data.

The parameters of a linear model are generally divided into subsets that correspond to meaningful sources of variation in the response functions. These sources, called *effects*, can be specified in the MODEL, LOGLIN, FACTORS, REPEATED, and CONTRAST statements. Effects can be specified in any of the following ways:

- A main effect is a single classification variable (that is, it produces class levels): `A  B  C`.
- A crossed effect (or interaction) is two or more classification variables joined by asterisks—for example: `A*B  A*B*C`.
- A nested effect is a main effect or an interaction, followed by a parenthetical field containing a main effect or an interaction. Multiple variables within the parentheses are assumed to form a crossed effect even when the asterisk is absent. In the following list, the last two effects are identical: `B(A)  C(A*B)  A*B(C*D)  A*B(C D)`.

- A nested-by-value effect is the same as a nested effect except that any variable in the parentheses can be followed by an equal sign and a value: B(A=1) C(A B=1) C*D(A=1 B=1) A(C='low').
- A direct effect is a variable specified in a DIRECT statement: X Y.
- Direct effects can be crossed with other effects: X*Y X*X*X X*A*B(C D=1).

The variables for crossed and nested effects remain in the order in which they are first encountered. For example, in the following model, the effect A*B is reported as B*A since B appears before A in the statement:

```
model R=B A A*B C(A B);
```

Also, C(A B) is interpreted as C(A*B) and is therefore reported as C(B*A).

Bar Notation

You can shorten the specification of multiple effects by using bar notation. For example, the following statements illustrate two methods of writing a full three-way factorial model:

```
proc catmod;
   model y=a b c a*b a*c b*c a*b*c;
run;

proc catmod;
   model y=a|b|c;
run;
```

When you use the bar (|) notation, the right and left sides become effects, and the interaction between them becomes an effect. Multiple bars are permitted. The expressions are expanded from left to right, using rules 1 through 4 given in Searle (1971, p. 390):

- Multiple bars are evaluated left to right. For example, A|B|C is evaluated as follows:
  A | B | C → {A | B} | C
  → {A B A*B} | C
  → A B A*B C A*C B*C A*B*C
- Crossed and nested groups of variables are combined. For example, A(B) | C(D) generates A*C(B D), among other terms.
- Duplicate variables are removed. For example, A(C) | B(C) generates A*B(C C), among other terms, and the extra C is removed.
- Effects are discarded if a variable occurs on both the crossed and nested sides of an effect. For instance, A(B) | B(D E) generates A*B(B D E), but this effect is deleted.

You can also specify the maximum number of variables involved in any effect that results from bar evaluation by specifying that maximum number, preceded by an @ sign, at the end of the bar effect.

For example, the specification A | B | C @ 2 would result in only those effects that contain two or fewer variables; in this case, the effects A, B, A*B, C, A*C, and B*C are generated.

Other examples of the bar notation follow:

A | C(B)	is equivalent to	A C(B) A*C(B)
A(B) | C(B)	is equivalent to	A(B) C(B) A*C(B)
A(B) | B(D E)	is equivalent to	A(B) B(D E)
A | B(A) | C	is equivalent to	A B(A) C A*C B*C(A)
A | B(A) | C@2	is equivalent to	A B(A) C A*C
A | B | C | D@2	is equivalent to	A B A*B C A*C B*C D A*D B*D C*D

For details about how the effects specified lead to a design matrix, see the section "Generation of the Design Matrix" on page 1153.

Output Data Sets

OUT= Data Set

For each population, the OUT= data set contains the observed and predicted values of the response functions, their standard errors, the residuals, and variables that describe the population and response profiles. In addition, if you use the standard response functions, the data set includes observed and predicted values for the cell frequencies or the cell probabilities, together with their standard errors and residuals.

Number of Observations

For the standard response functions, there are $s \times (2q - 1)$ observations in the data set for each BY group, where s is the number of populations and q is the number of response functions per population. Otherwise, there are $s \times q$ observations in the data set for each BY group.

Variables in the OUT= Data Set

The data set contains the following variables:

BY variables	If you use a BY statement, the BY variables are included in the OUT= data set.
dependent variables	If the response functions are the default ones (generalized logits), then the dependent variables, which describe the response profiles, are included in the OUT= data set. When _TYPE_=FUNCTION, the values of these variables are missing.
independent variables	The independent variables, which describe the population profiles, are included in the OUT= data set.
NUMBER	the sequence number of the response function or the cell probability or the cell frequency

OBS	the observed value
PRED	the predicted value
RESID	the residual (observed minus predicted)
SAMPLE	the population number. This matches the sample number in the "Population Profile" section of the output.
SEOBS	the standard error of the observed value
SEPRED	the standard error of the predicted value
TYPE	specifies a character variable with three possible values. When _TYPE_=FUNCTION, the observed and predicted values are values of the response functions. When _TYPE_=PROB, they are values of the cell probabilities. When _TYPE_=FREQ, they are values of the cell frequencies. Cell probabilities or frequencies are provided only when the default response functions are modeled. In this case, cell probabilities are provided by default, and cell frequencies are provided if you specify the option PRED=FREQ.

OUTEST= Data Set

This TYPE=EST output data set contains the estimated parameter vector and its estimated covariance matrix. If you specify both the ML and WLS options in the MODEL statement, the OUTEST= data set contains both sets of estimates. For each BY group, there are $p + 1$ observations in the data set for each estimation method, where p is the number of estimated parameters. The data set contains the following variables:

B1, B2, and so on	variables for the estimated parameters. The OUTEST= data set contains one variable for each estimated parameter.
BY variables	If you use a BY statement, the BY variables are included in the OUT= data set.
METHOD	the method used to obtain parameter estimates. For weighted least squares estimation, _METHOD_=WLS, and for maximum likelihood estimation, _METHOD_=ML.
NAME	identifies parameter names. When _TYPE_=PARMS, _NAME_ is blank, but when _TYPE_=COV, _NAME_ has one of the values B1, B2, and so on, corresponding to the parameter names.
STATUS	indicates whether the estimates have converged
TYPE	identifies the statistics contained in the variables for parameter estimates (B1, B2, and so on). When _TYPE_=PARMS, the variables contain parameter estimates; when _TYPE_=COV, they contain covariance estimates.

The variables _METHOD_, _NAME_, and _TYPE_ are character variables; the BY variables can be either character or numeric; and the variables for estimated parameters are numeric.

See Appendix A, "Special SAS Data Sets," for more information about special SAS data sets.

Logistic Analysis

In a logistic analysis, the response functions are the logits of the dependent variable.

PROC CATMOD can compute the three following types of logits with the use of keywords in the RESPONSE statement. Note that other types of response functions can be generated by specifying appropriate transformations in the RESPONSE statement.

- Generalized logits are used primarily for nominally scaled dependent variables, but they can also be used for ordinal data modeling. Maximum likelihood estimation is available for the analysis of these logits.
- Cumulative logits are used for ordinally scaled dependent variables. Except for dependent variables with two response levels, only weighted least squares estimation is available for the analysis of these logits.
- Adjacent-category logits are equivalent to generalized logits, but they have some advantages for ordinal data analysis because they automatically incorporate integer scores for the levels of the dependent variable. Except for dependent variables with two response levels, only weighted least squares estimation is available for the analysis of these logits.

If the dependent variable has only two responses, then the cumulative logit and the adjacent-category logit are the negative of the generalized logit, as computed by PROC CATMOD. Consequently, parameter estimates obtained by using these logits are the negative of those obtained from using generalized logits. A simple logistic analysis of variance uses statements like the following:

```
proc catmod;
   model r=a|b;
run;
```

Logistic Regression

If the independent variables are treated quantitatively (like continuous variables), then a logistic analysis is known as a *logistic regression*. If you want PROC CATMOD to treat the independent variables as quantitative variables, specify them in both the DIRECT and MODEL statements, as follows:

```
proc catmod;
   direct x1 x2 x3;
   model r=x1 x2 x3;
run;
```

Since the preceding statements do not include a RESPONSE statement, generalized logits are computed. See Example 28.3 for another example.

The parameter estimates from the CATMOD procedure are the same as those from a logistic regression program such as PROC LOGISTIC (see Chapter 51, "The LOGISTIC Procedure"). The chi-square statistics and the predicted values are also identical. In the binary response case,

PROC CATMOD can be made to model the probability of the maximum value by either (1) organizing the input data so that the maximum value occurs first and specifying ORDER=DATA in the PROC CATMOD statement or (2) specifying cumulative logits (CLOGITS) in the RESPONSE statement.

CAUTION: Computational difficulties might occur if you use a continuous variable with a large number of unique values in a DIRECT statement. See the section "Continuous Variables" on page 1147 for more details.

Cumulative Logits

If your dependent variable is ordinally scaled, you can specify the analysis of cumulative logits that take into account the ordinal nature of the dependent variable:

```
proc catmod;
   response clogits;
   direct x;
   model r=a x;
run;
```

The preceding statements correspond to a simple analysis that addresses the question of existence of an association between the independent variables and the ordinal dependent variable. However, there are some commonly used models for the analysis of ordinal data (Agresti 1984) that address the structure of association (in terms of odds ratios), as well as its existence.

If the independent variables are classification variables, a typical analysis for such a model uses the following statements:

```
proc catmod;
   weight wt;
   response clogits;
   model r=_response_ a b;
run;
```

On the other hand, if the independent variables are ordinally scaled, you might specify numeric scores in variables x1 and x2, and use the following statements:

```
proc catmod;
   weight wt;
   direct x1 x2;
   response clogits;
   model r=_response_ x1 x2;
run;
```

See Agresti (1984) for additional details of estimation, testing, and interpretation.

Continuous Variables

Computational difficulties might occur if you have a continuous variable with a large number of unique values and you use this variable in a DIRECT statement, since an observation often repre-

sents a separate population of size one. At this extreme of sparseness, the weighted least squares method is inappropriate since there are too many zero frequencies. Therefore, you should use the maximum likelihood method. PROC CATMOD is not designed optimally for continuous variables; therefore, it might be less efficient and unable to allocate sufficient memory to handle this problem, as compared with a procedure designed specifically to handle continuous data. In these situations, consider using the LOGISTIC or GENMOD procedure to analyze your data.

Log-Linear Model Analysis

When the response functions are the default generalized logits, then inclusion of the keyword _RESPONSE_ in every effect in the right side of the MODEL statement fits a log-linear model. The keyword _RESPONSE_ tells PROC CATMOD that you want to model the variation among the dependent variables. You then specify the actual model in the LOGLIN statement.

When you perform log-linear model analysis, you can request weighted least squares estimates, maximum likelihood estimates, or both. By default, PROC CATMOD calculates maximum likelihood estimates when the default response functions are used. The following table provides appropriate MODEL statements for the combinations of types of estimates:

Estimation Desired	MODEL Statement
Maximum likelihood (Newton-Raphson)	`model a*b=_response_;`
Maximum likelihood (Iterative Proportional Fitting)	`model a*b=_response_ / ml=ipf;`
Weighted least squares	`model a*b=_response_ / wls;`
Maximum likelihood and weighted least squares	`model a*b=_response_ / wls ml;`

CAUTION: Sampling zeros in the input data set should be specified with the ZERO= option to ensure that these sampling zeros are not treated as structural zeros. Alternatively, you can replace cell counts for sampling zeros with some positive number close to zero (such as 1E–20) in a DATA step. Data containing sampling zeros should be analyzed with maximum likelihood estimation. See the section "Cautions" on page 1163 and Example 28.5 for further information and an illustration that uses both cell count data and raw data.

One Population

The usual log-linear model analysis has one population, which means that all of the variables are dependent variables. For example, the following statements yield a maximum likelihood analysis of a saturated log-linear model for the dependent variables r1 and r2:

```
proc catmod;
   weight wt;
   model r1*r2=_response_;
   loglin r1|r2;
run;
```

If you want to fit a reduced model with respect to the dependent variables (for example, a model of independence or conditional independence), specify the reduced model in the LOGLIN statement. For example, the following statements yield a main-effects log-linear model analysis of the factors r1 and r2:

```
proc catmod;
   weight wt;
   model r1*r2=_response_ / pred;
   loglin r1 r2;
run;
```

The output includes Wald statistics for the individual effects r1 and r2, as well as predicted cell probabilities. Moreover, the goodness-of-fit statistic is the likelihood ratio test for the hypothesis of independence between r1 and r2 or, equivalently, a test of r1*r2.

Multiple Populations

You can do log-linear model analysis with multiple populations by using a POPULATION statement or by including effects on the right side of the MODEL statement that contain independent variables. Each effect must include the _RESPONSE_ keyword.

For example, suppose the dependent variables r1 and r2 are dichotomous, and the independent variable group has three levels. Then the following statements specify a saturated model (three degrees of freedom for _RESPONSE_ and six degrees of freedom for the interaction between _RESPONSE_ and group):

```
proc catmod;
   weight wt;
   model r1*r2=_response_ group*_response_;
   loglin r1|r2;
run;
```

From another point of view, _RESPONSE_*group can be regarded as a main effect for group with respect to the three response functions, while _RESPONSE_ can be regarded as an intercept effect with respect to the functions. In other words, the following statements give essentially the same results as the logistic analysis:

```
proc catmod;
   weight wt;
   model r1*r2=group;
run;
```

The ability to model the interaction between the independent and the dependent variables becomes particularly useful when a reduced model is specified for the dependent variables. For example, the following statements specify a model with two degrees of freedom for _RESPONSE_ (one for r1 and one for r2) and four degrees of freedom for the interaction of _RESPONSE_*group:

```
proc catmod;
   weight wt;
   model r1*r2=_response_ group*_response_;
   loglin r1 r2;
run;
```

The likelihood ratio goodness-of-fit statistic (three degrees of freedom) tests the hypothesis that r1 and r2 are independent in each of the three groups.

Iterative Proportional Fitting

You can use the iterative proportional fitting (IPF) algorithm to fit a hierarchical log-linear model with no independent variables and no population variables.

The advantage of IPF over the Newton-Raphson (NR) algorithm and over the weighted least squares (WLS) method is that, when the contingency table has several dimensions and the parameter vector is large, you can obtain the log likelihood, the goodness-of-fit G^2, and the predicted frequencies or probabilities without performing potentially expensive parameter estimation and covariance matrix calculations. This enables you to do the following:

- compare two models by computing the likelihood ratio statistics to test the significance of the contribution of the variables in one model that are not in the other model
- compute predicted values of the cell probabilities or frequencies for the final model

Each iteration of the IPF algorithm is generally faster than an iteration of the NR algorithm; however, the IPF algorithm converges to the MLEs more slowly than the NR algorithm. Both NR and WLS are more general methods that are able to perform more complex analyses than IPF can.

Repeated Measures Analysis

If there are multiple dependent variables and the variables represent repeated measurements of the same observational unit, then the variation among the dependent variables can be attributed to one or more repeated measurement factors. The factors can be included in the model by specifying _RESPONSE_ on the right side of the MODEL statement and by using a REPEATED statement to identify the factors.

To perform a repeated measures analysis, you also need to specify a RESPONSE statement, since the standard response functions (generalized logits) cannot be used. Typically, the MEANS or MARGINALS response functions are specified in a repeated measures analysis, but other response functions can also be reasonable.

One Population

Consider an experiment in which each subject is measured at three times, and the response functions are marginal probabilities for each of the dependent variables. If the dependent variables each have k levels, then PROC CATMOD computes $k-1$ response functions for each time. Differences among the response functions with respect to these times could be attributed to the repeated measurement factor Time. To incorporate the Time variation into the model, specify the following statements:

```
proc catmod;
   response marginals;
   model t1*t2*t3=_response_;
   repeated Time 3 / _response_=Time;
run;
```

These statements produce a Time effect that has $2(k-1)$ degrees of freedom since there are $k-1$ response functions at each time point. For a dichotomous variable, the Time effect has two degrees of freedom.

Now suppose that at each time point, each subject has X-rays taken, and the X-rays are read by two different radiologists. This creates six dependent variables that represent the 3×2 cross-classification of the repeated measurement factors Time and Reader. A saturated model with respect to these factors can be obtained by specifying the following statements:

```
proc catmod;
   response marginals;
   model r11*r12*r21*r22*r31*r32=_response_;
   repeated Time 3, Reader 2
      / _response_=Time Reader Time*Reader;
run;
```

If you want to fit a main-effects model with respect to Time and Reader, then change the REPEATED statement to the following:

```
   repeated Time 3, Reader 2 / _response_=Time Reader;
```

If you want to fit a main-effects model for Time but for only one of the readers, the REPEATED statement might look like the following:

```
   repeated Time $ 3, Reader $ 2
            /_response_=Time(Reader=Smith)
              profile  =('1'  Smith,
                         '1'  Jones,
                         '2'  Smith,
                         '2'  Jones,
                         '3'  Smith,
                         '3'  Jones);
```

If Jones had been unavailable for a reading at time 3, then there would be only $5(k-1)$ response functions, even though PROC CATMOD would be expecting some multiple of 6 ($= 3 \times 2$). In that case, the PROFILE= option would be necessary to indicate which repeated measurement profiles were actually represented:

```
repeated Time $ 3, Reader $ 2
         /_response_=Time(Reader=Smith)
           profile  =('1'  Smith,
                      '1'  Jones,
                      '2'  Smith,
                      '2'  Jones,
                      '3'  Smith);
```

When two or more repeated measurement factors are specified, PROC CATMOD presumes that the response functions are ordered so that the levels of the rightmost factor change most rapidly. This means that the dependent variables should be specified in the same order. For this example, the order implied by the REPEATED statement is as follows, where the variable r_{ij} corresponds to Time i and Reader j:

Response Function	Dependent Variable	Time	Reader
1	r_{11}	1	1
2	r_{12}	1	2
3	r_{21}	2	1
4	r_{22}	2	2
5	r_{31}	3	1
6	r_{32}	3	2

The order of dependent variables in the MODEL statement must agree with the order implied by the REPEATED statement.

Multiple Populations

When there are variables specified in the POPULATION statement or on the right side of the MODEL statement, these variables produce multiple populations. PROC CATMOD can then model these independent variables, the repeated measurement factors, and the interactions between the two.

For example, suppose that there are five groups of subjects, that each subject in the study is measured at three different times, and that the dichotomous dependent variables are labeled t1, t2, and t3. The following statements computate three response functions for each population:

```
proc catmod;
   weight wt;
   population Group;
   response marginals;
   model t1*t2*t3=_response_;
   repeated Time / _response_=Time;
run;
```

PROC CATMOD then regards _RESPONSE_ as a variable with three levels corresponding to the three response functions in each population and forms an effect with two degrees of freedom. The MODEL and REPEATED statements tell PROC CATMOD to fit the main effect of Time.

In general, the MODEL statement tells PROC CATMOD how to integrate the independent variables and the repeated measurement factors into the model. For example, again suppose that there are five groups of subjects, that each subject is measured at three times, and that the dichotomous independent variables are labeled t1, t2, and t3. If you use the same WEIGHT, POPULATION, RESPONSE, and REPEATED statements as in the preceding program, the following MODEL statements result in the indicated analyses:

`model t1*t2*t3=Group / averaged;`	specifies the Group main effect (with 4 degrees of freedom).
`model t1*t2*t3=_response_;`	specifies the Time main effect (with 2 degrees of freedom).
`model t1*t2*t3=_response_*Group;`	specifies the interaction between Time and Group (with 8 degrees of freedom).
`model t1*t2*t3=_response_|Group;`	specifies both main effects, and the interaction between Time and Group (with a total of 14 degrees of freedom).
`model t1*t2*t3=_response_(Group);`	specifies a Time main effect within each Group (with 10 degrees of freedom).

However, the following MODEL statement is invalid since effects cannot be nested within _RESPONSE_:

```
model t1*t2*t3=Group(_response_);
```

Generation of the Design Matrix

Each row of the design matrix (corresponding to a population) is generated by a unique combination of independent variable values. Each column of the design matrix corresponds to a model parameter. The columns are produced from the effect specifications in the MODEL, LOGLIN, FACTORS, and REPEATED statements. For details about effect specifications, see the section "Specification of Effects" on page 1142.

This section is divided into three parts:

- one response function per population
- two or more response functions per population (excluding log-linear models), beginning on page 1156
- log-linear models, beginning on page 1160

This section assumes that the default effect parameterization is used. Specifying the reference parameterization replaces the "−1"s with zeros in the design matrix for the main effects of classification variables, and makes appropriate changes to interaction terms.

You can display the design matrix by specifying the DESIGN option in the MODEL statement.

One Response Function per Population

Intercept

When there is one response function per population, all design matrices start with a column of 1s for the intercept unless the NOINT option is specified or the design matrix is input directly.

Main Effects

If a classification variable A has k levels, then its main effect has $k-1$ degrees of freedom, and the design matrix has $k-1$ columns that correspond to the first $k-1$ levels of A. The ith column contains a 1 in the ith row, a -1 in the last row, and 0s everywhere else. If α_i denotes the parameter that corresponds to the ith level of variable A, then the $k-1$ columns yield estimates of the independent parameters, $\alpha_1, \alpha_i, \ldots, \alpha_{k-1}$. The last parameter is not needed because PROC CATMOD constrains the k parameters to sum to zero. In other words, PROC CATMOD uses a full-rank center-point parameterization to build design matrices. Here are two examples:

Variable	Data Levels	Effect Parameterization Design Matrix Columns	
A	1	1	0
	2	0	1
	3	−1	−1
B	1	1	
	2	−1	

For an effect with three levels, such as A, PROC CATMOD produces two parameter estimates for each response function. By default, the first (corresponding to the first row in the design columns) estimates the effect of level 1 of A compared to the average effect of the three levels of A. The second (corresponding to the second row in the design columns) estimates the effect of level 2 of A compared to the average effect of the three levels of A. The sum-to-zero constraint requires the effect of level 3 of A to be the negative of the sum of the level 1 and 2 effects (as shown by the third row in the design columns).

Crossed Effects (Interactions)

Crossed effects (such as A*B) are formed by the horizontal direct products of main effects, as illustrated in the following table:

Data Levels		Design Matrix Columns				
A	**B**	**A**		**B**	**A*B**	
1	1	1	0	1	1	0
1	2	1	0	−1	−1	0
2	1	0	1	1	0	1
2	2	0	1	−1	0	−1
3	1	−1	−1	1	−1	−1
3	2	−1	−1	−1	1	1

The number of degrees of freedom for a crossed effect (that is, the number of design matrix columns) is equal to the product of the numbers of degrees of freedom for the separate effects.

Nested Effects

The effect A(B) is read "A within B" and is the same as specifying an A main effect for every value of B. If n_a and n_b are the number of levels in A and B, respectively, then the number of columns for A(B) is $(n_a - 1)n_b$ when every combination of levels exists in the data. The following table gives an example:

Data Levels		Design Matrix Columns			
B	A	A(B)			
1	1	1	0	0	0
1	2	0	1	0	0
1	3	−1	−1	0	0
2	1	0	0	1	0
2	2	0	0	0	1
2	3	0	0	−1	−1

CAUTION: PROC CATMOD actually allocates a column for all possible combinations of values even though some combinations are not present in the data. This can be of particular concern if the data are not balanced with respect to the nested levels.

Nested-by-Value Effects

Instead of nesting an effect within all values of the main effect, you can nest an effect within specified values of the nested variable (A(B=1), for example). The four degrees of freedom for the A(B) effect shown in the preceding section can also be obtained by specifying the two separate nested effects with values, as the following table shows:

Data Levels		Design Matrix Columns			
B	A	A(B=1)		A(B=2)	
1	1	1	0	0	0
1	2	0	1	0	0
1	3	−1	−1	0	0
2	1	0	0	1	0
2	2	0	0	0	1
2	3	0	0	−1	−1

Each effect has $n_a - 1$ degrees of freedom, assuming a complete combination, so each effect in this example has two degrees of freedom.

The procedure compares nested values to data values on the basis of formatted values. If a format is not specified for the variable, the procedure formats internal data values to BEST16, left-justified. The nested values specified in nested-by-value effects are also converted to a BEST16 formatted value, left-justified.

For example, if the numeric variable B has internal data values 1 and 2, then A(B=1), A(B=1.0), and A(B=1E0) are all valid nested-by-value effects. However, if the data value 1 is formatted as 'one', then A(B='one') is a valid effect, but A(B=1) is not since the formatted nested value (1) does not match the formatted data value (one).

To ensure correct nested-by-value effects, look at the tables of population and response profiles. These are displayed by default, and they contain the formatted data values. In addition, the population and response profiles are displayed when you specify the ONEWAY option in the MODEL statement.

Direct Effects

To request that the actual values of a variable be inserted into the design matrix, declare the variable in a DIRECT statement, and specify the effect by the variable name. For example, specifying the effects X1 and X2 in both the MODEL and DIRECT statements results in the following:

Data Levels		Design Columns	
X1	X2	X1	X2
1	1	1	1
2	4	2	4
3	9	3	9

Unless there is a POPULATION statement that excludes the direct variables, the direct variables help to define the sample populations. In general, the variables should not be continuous in the sense that every subject has a different value because this would create a separate population for each subject (note, however, that such a strategy is used purposely for logistic regression).

If there is a POPULATION statement that omits mention of the direct variables, then the values of the direct variables must be identical for all subjects in a given population since there can be only one independent variable profile for each population.

Two or More Response Functions per Population

When there is more than one response function per population, the structure of the design matrix depends on whether or not the model type is AVERAGED (see the AVERAGED option in the MODEL statement). The model type is AVERAGED if independent variable effects are averaged over the multiple responses within a population rather than being nested in them.

The following subsections illustrate the effect of specifying (or not specifying) an AVERAGED model type. This section does not apply to log-linear models; for these models, see the section "Log-Linear Model Design Matrices" on page 1160.

Model Type Not AVERAGED

Suppose the variable A has two levels, and you specify the following statements:

```
proc catmod;
   model Y=A / design;
run;
```

If the variable Y has two levels, then there is only one response function per population, and the design matrix is as follows:

	Design Matrix	
Sample	Intercept	A
1	1	1
2	1	−1

But if the variable Y has three levels, then there are two response functions per population, and the preceding design matrix is assumed to hold for each of the two response functions. The response functions are always ordered so that the multiple response functions within a population are grouped together. For this example, the design matrix would be as follows:

	Response Function	Design Matrix			
Sample	Number	Intercept		A	
1	1	1	0	1	0
1	2	0	1	0	1
2	1	1	0	−1	0
2	2	0	1	0	−1

Since the same submatrix applies to each of the multiple response functions, PROC CATMOD displays only the submatrix (that is, the one it would create if there were only one response function per population) rather than the entire design matrix. PROC CATMOD displays

$$\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Ordering of Parameters

This grouping of multiple response functions within populations also has an effect in the table of parameter estimates displayed by PROC CATMOD. The following table shows some parameter estimates, where the four rows of the table correspond to the four columns in the preceding design matrix:

Effect	Parameter	Estimate
Intercept	1	1.4979
	2	0.8404
A	3	0.1116
	4	−0.3296

Notice that the intercept and the A effect each have two parameter estimates associated with them. The first estimate in each pair is associated with the first response function, and the second in each pair is associated with the second response function. Consequently, 0.1116 is the effect of the first level of A on the first response function. In any table of parameter estimates displayed by PROC CATMOD, as you read down the column of estimates, the response function level changes before levels of the variables making up the effect.

Model Type AVERAGED

When the model type is AVERAGED (for example, when the AVERAGED option is specified in the MODEL statement, when _RESPONSE_ is used in the MODEL statement, or when the design matrix is input directly in the MODEL statement), PROC CATMOD does not assume that the same submatrix applies to each of the q response functions per population. Rather, it averages any independent variable effects across the functions, and it enables you to study variation among the q functions. The first column of the design matrix is always a column of 1s corresponding to the intercept, unless the NOINT option is specified in the MODEL statement or the design matrix is input directly. Also, since the design matrix does not have any special submatrix structure, PROC CATMOD displays the entire matrix.

For example, suppose the dependent variable Y has three levels, the independent variable A has two levels, and you specify the following statements:

```
proc catmod;
   response marginals;
   model y=a / averaged design;
run;
```

Then there are two response functions per population, and the response functions are always ordered so that the multiple response functions within a population are grouped together. For this example, the design matrix would be as follows:

Sample	Response Function Number	Design Matrix	
		Intercept	A
1	1	1	1
1	2	1	1
2	1	1	−1
2	2	1	−1

Note that the model now has only two degrees of freedom. The remaining two degrees of freedom in the residual correspond to variation among the three levels of the dependent variable. Generally, that variation tends to be statistically significant and therefore should not be left out of the model. You can include it in the model by including the two effects, _RESPONSE_ and _RESPONSE_*A, but if the study is not a repeated measures study, those sources of variation tend to be uninteresting. The usual solution for this type of study (one dependent variable) is to exclude the AVERAGED option from the MODEL statement.

An AVERAGED model type is automatically used whenever you use the _RESPONSE_ keyword in the MODEL statement. The _RESPONSE_ effect models variation among the q response

functions per population. If there is no REPEATED, FACTORS, or LOGLIN statement, then PROC CATMOD builds a main effect with $q - 1$ degrees of freedom. For example, three response functions would produce the following design columns:

Response Function Number	Design Columns _RESPONSE_	
1	1	0
2	0	1
3	−1	−1

If there is more than one population, then the _RESPONSE_ effect is averaged over the populations. Also, the _RESPONSE_ effect can be crossed with any other effect, or it can be nested within an effect.

If there is a REPEATED statement that contains only one repeated measurement factor, then PROC CATMOD builds the design columns for _RESPONSE_ in the same way, except that the output labels the main effect with the factor name rather than with the word _RESPONSE_. For example, suppose an independent variable A has two levels, and the input statements are as follows:

```
proc catmod;
   response marginals;
   model Time1*Time2=A _response_ A*_response_ / design;
   repeated Time 2 / _response_=Time;
run;
```

If Time1 and Time2 each have two levels (so that they each have one independent marginal probability), then the RESPONSE statement makes PROC CATMOD compute two response functions per population. The design matrix is as follows:

Sample	Response Function Number	Design Matrix Intercept	A	Time	A*Time
1	1	1	1	1	1
1	2	1	1	−1	−1
2	1	1	−1	1	−1
2	2	1	−1	−1	1

However, if Time1 and Time2 each have three levels (so that they each have two independent marginal probabilities), then the RESPONSE statement causes PROC CATMOD to compute four response functions per population. In that case, since Time has two levels, PROC CATMOD groups the functions into sets of 2 $(= 4/2)$ and constructs the preceding submatrix for each function in the set. This results in the following design matrix, which is obtained from the previous one by multiplying each element by an identity matrix of order two:

Sample	Response Function	Design Matrix							
		Intercept		A		Time		A*Time	
1	P(Time1=1)	1	0	1	0	1	0	1	0
1	P(Time1=2)	0	1	0	1	0	1	0	1
1	P(Time2=1)	1	0	1	0	−1	0	−1	0
1	P(Time2=2)	0	1	0	1	0	−1	0	−1
2	P(Time1=1)	1	0	−1	0	1	0	−1	0
2	P(Time1=2)	0	1	0	−1	0	1	0	−1
2	P(Time2=1)	1	0	−1	0	−1	0	1	0
2	P(Time2=2)	0	1	0	−1	0	−1	0	1

If there is a REPEATED statement that contains two or more repeated measurement factors, then PROC CATMOD builds the design columns for _RESPONSE_ according to the definition of _RESPONSE_ in the REPEATED statement. For example, suppose you specify the following statements:

```
proc catmod;
   response marginals;
   model R11*R12*R21*R22=_response_ / design;
   repeated Time 2, Place 2 / _response_=Time Place;
run;
```

If each of the dependent variables has two levels, then PROC CATMOD builds four response functions. The _RESPONSE_ effect generates a main-effects model with respect to Time and Place, and the design matrix is as follows:

Response Function Number	Variable	Time	Place	Design Matrix	
				Intercept	_RESPONSE_
1	R11	1	1	1	1
2	R12	1	2	1	−1
3	R21	2	1	1	−1
4	R22	2	2	1	−1

Log-Linear Model Design Matrices

When the response functions are the standard ones (generalized logits), then inclusion of the keyword _RESPONSE_ in every design effect fits a log-linear model. The design matrix for a log-linear model looks different from a standard design matrix because the standard one is transformed by the same linear transformation that converts the r response probabilities to $r-1$ generalized logits. For example, suppose the dependent variables X and Y each have two levels, and you specify a saturated log-linear model analysis:

```
proc catmod;
   model X*Y=_response_ / design;
   loglin X Y X*Y;
run;
```

Then the cross-classification of X and Y yields four response probabilities, p_{11}, p_{12}, p_{21}, and p_{22}, which are then reduced to three generalized logit response functions, $F_1 = \log(p_{11}/p_{22})$, $F_2 = \log(p_{12}/p_{22})$, and $F_3 = \log(p_{21}/p_{22})$.

Since the saturated log-linear model implies that

$$\begin{bmatrix} \log(p_{11}) \\ \log(p_{12}) \\ \log(p_{21}) \\ \log(p_{22}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \boldsymbol{\gamma} - \lambda \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix} \boldsymbol{\beta} - \delta \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

where $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are parameter vectors, and λ and δ are normalizing constants required by the restriction that the probabilities sum to 1, it follows that the MODEL statement yields

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} \log(p_{11}) \\ \log(p_{12}) \\ \log(p_{21}) \\ \log(p_{22}) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & -1 & -1 \\ -1 & 1 & -1 \\ -1 & -1 & 1 \end{bmatrix} \boldsymbol{\beta}$$

$$= \begin{bmatrix} 2 & 2 & 0 \\ 2 & 0 & -2 \\ 0 & 2 & -2 \end{bmatrix} \boldsymbol{\beta}$$

The design matrix is as follows:

Sample	Response Function Number	Design Matrix X	Y	X*Y
1	1	2	2	0
1	2	2	0	−2
1	3	0	2	−2

Design matrices for reduced models are constructed similarly. For example, suppose you request a main-effects log-linear model analysis of the factors X and Y:

```
proc catmod;
   model X*Y=_response_ / design;
   loglin X Y;
run;
```

Since the main-effects log-linear model implies that

$$\begin{bmatrix} \log(p_{11}) \\ \log(p_{12}) \\ \log(p_{21}) \\ \log(p_{22}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & -1 \\ 1 & -1 & 1 \\ 1 & -1 & -1 \end{bmatrix} \boldsymbol{\gamma} - \lambda \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{bmatrix} \boldsymbol{\beta} - \delta \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

it follows that the MODEL statement yields

$$\begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} \log(p_{11}) \\ \log(p_{12}) \\ \log(p_{21}) \\ \log(p_{22}) \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & 1 & -1 \end{bmatrix} \times \begin{bmatrix} 1 & 1 \\ 1 & -1 \\ -1 & 1 \\ -1 & -1 \end{bmatrix} \boldsymbol{\beta}$$

$$= \begin{bmatrix} 2 & 2 \\ 2 & 0 \\ 0 & 2 \end{bmatrix} \boldsymbol{\beta}$$

Therefore, the corresponding design matrix is as follows:

Sample	Response Function Number	Design Matrix X	Design Matrix Y
1	1	2	2
1	2	2	0
1	3	0	2

Since it is difficult to tell from the final design matrix whether PROC CATMOD used the parameterization that you intended, the procedure displays the untransformed _RESPONSE_ matrix for log-linear models. For example, specifying the main-effects model in the preceding example displays the following matrix:

Response Function Number	_RESPONSE_ Matrix 1	2
1	1	1
2	1	−1
3	−1	1
4	−1	−1

You can suppress the display of this matrix by specifying the NORESPONSE option in the MODEL statement.

Cautions

Effective Sample Size

Since the method depends on asymptotic approximations, you need to be careful that the sample sizes are sufficiently large to support the asymptotic normal distributions of the response functions. A general guideline is that you would like to have an effective sample size of at least 25 to 30 for each response function that is being analyzed. For example, if you have one dependent variable and $r = 4$ response levels, and you use the standard response functions to compute three generalized logits for each population, then you would like the sample size of each population to be at least 75. Moreover, the subjects should be dispersed throughout the table so that less than 20 percent of the response functions have an effective sample size less than 5. For example, if each population had less than 5 subjects in the first response category, then it would be wiser to pool this category with another category rather than to assume the asymptotic normality of the first response function. Or, if the dependent variable is ordinally scaled, an alternative is to request the mean score response function rather than three generalized logits.

If there is more than one dependent variable, and you specify RESPONSE MEANS, then the effective sample size for each response function is the same as the actual sample size. Thus, a sample size of 30 could be sufficient to support four response functions, provided that the functions are the means of four dependent variables.

A Singular Covariance Matrix

If there is a singular (noninvertible) covariance matrix for the response functions in any population, then PROC CATMOD writes an error message and stops processing. You have several options available to correct this problem:

- You can reduce the number of response functions according to how many can be supported by the populations with the smallest sample sizes.
- If there are three or more levels for any independent variable, you can pool the levels into a fewer number of categories, thereby reducing the number of populations. However, your

interpretation of results must be done more cautiously since such pooling implies a different sampling scheme and masks any differences that existed among the pooled categories.

- If there are two or more independent variables, you can delete at least one of them from the model. However, this is just another form of pooling, and the same cautions that apply to the previous option also apply here.
- If there is one independent variable, then, in some situations, you might simply eliminate the populations that are causing the covariance matrices to be singular.
- You can use the ADDCELL= option in the MODEL statement to add a small amount (for example, 0.5) to every cell frequency, but this can seriously bias the results if the cell frequencies are small.

Zero Frequencies

There are two types of zero cells in a contingency table: structural and sampling. A structural zero cell has an expected value of zero, while a sampling zero cell can have nonzero expected value and can be estimable.

If you use the standard response functions and there are zero frequencies, you should use maximum likelihood estimation (the default is ML=NR) rather than weighted least squares to analyze the data. For weighted least squares analysis, the CATMOD procedure always computes the observed response functions and might need to take the logarithm of a zero proportion. In this case, PROC CATMOD issues a warning and then takes the log of a small value ($0.5/n_i$ for the probability) in order to continue, but this can produce invalid results if the cells contain too few observations. Maximum likelihood analysis, on the other hand, does not require computation of the observed response functions and therefore yields valid results for the parameter estimates and all of the predicted values.

For a log-linear model analysis with WLS or ML=NR, PROC CATMOD creates response profiles only for the observed profiles. For any log-linear model analysis with one population (the usual case), the contingency table does not contain zeros, which means that all zero frequencies are treated as structural zeros. If there is more than one population, then a zero in the body of the contingency table is treated as a sampling zero (as long as some population has a nonzero count for that profile). If you fit the log-linear model by using ML=IPF, the contingency table is incomplete and the zeros are treated like structural zeros. If you want zero frequencies that PROC CATMOD would normally treat as structural zeros to be interpreted as sampling zeros, you can specify the ZERO=SAMPLING and MISSING=SAMPLING options in the MODEL statement. Alternatively, you can specify ZERO=1E–20 and MISSING=1E–20.

See Bishop, Fienberg, and Holland (1975) for a discussion of the issues and Example 28.5 for an illustration of a log-linear model analysis of data that contain both structural and sampling zeros.

If you perform a weighted least squares analysis on a contingency table that contains zero cell frequencies, then avoid using the LOG transformation as the first transformation on the observed proportions. In general, it is better to change the response functions or to pool some of the response categories than to settle for the 0.5 correction or to use the ADDCELL= option.

Testing the Wrong Hypothesis

If you use the keyword _RESPONSE_ in the MODEL statement, and you specify MARGINALS, LOGITS, ALOGITS, or CLOGITS in your RESPONSE statement, you might receive the following warning message:

```
Warning: The _RESPONSE_ effect may be testing the wrong
         hypothesis since the marginal levels of the
         dependent variables do not coincide. Consult the
         response profiles and the CATMOD documentation.
```

The following examples illustrate situations in which the _RESPONSE_ effect tests the wrong hypothesis.

Zeros in the Marginal Frequencies

Suppose you specify the following statements:

```
data A1;
   input Time1 Time2 @@;
   datalines;
1 2    2 3    1 3
;

proc catmod;
   response marginals;
   model Time1*Time2=_response_;
   repeated Time 2 / _response_=Time;
run;
```

One marginal probability is computed for each dependent variable, resulting in two response functions. The model is a saturated one: one degree of freedom for the intercept and one for the main effect of Time. Except for the warning message, PROC CATMOD produces an analysis with no apparent errors, but the "Response Profiles" table displayed by PROC CATMOD is as follows:

Response Profiles

Response	Time1	Time2
1	1	2
2	1	3
3	2	3

Since RESPONSE MARGINALS yields marginal probabilities for every level but the last, the two response functions being analyzed are Prob(Time1=1) and Prob(Time2=2). The Time effect is testing the hypothesis that Prob(Time1=1)=Prob(Time2=2). What it *should* be testing is the hypothesis that

```
Prob(Time1=1) = Prob(Time2=1)
Prob(Time1=2) = Prob(Time2=2)
Prob(Time1=3) = Prob(Time2=3)
```

but there are not enough data to support the test (assuming that none of the probabilities are structural zeros by the design of the study).

The ORDER=DATA Option

Suppose you specify the following statements:

```
data a1;
   input Time1 Time2 @@;
   datalines;
2 1    2 2    1 1    1 2    2 1
;

proc catmod order=data;
   response marginals;
   model Time1*Time2=_response_;
   repeated Time 2 / _response_=Time;
run;
```

As in the preceding example, one marginal probability is computed for each dependent variable, resulting in two response functions. The model is also the same: one degree of freedom for the intercept and one for the main effect of Time. PROC CATMOD issues the warning message and displays the following "Response Profiles" table:

Response Profiles

Response	Time1	Time2
1	2	1
2	2	2
3	1	1
4	1	2

Although the marginal levels are the same for the two dependent variables, they are not in the same order because the ORDER=DATA option specified that they be ordered according to their appearance in the input stream. Since RESPONSE MARGINALS yields marginal probabilities for every level except the last, the two response functions being analyzed are Prob(Time1=2) and Prob(Time2=1). The Time effect is testing the hypothesis that Prob(Time1=2)=Prob(Time2=1). What it *should* be testing is the hypothesis that

```
Prob(Time1=1) = Prob(Time2=1)
Prob(Time1=2) = Prob(Time2=2)
```

Whenever the warning message appears, look at the "Response Profiles" table or the "One-Way Frequencies" table to determine what hypothesis is actually being tested. For the latter example, a correct analysis can be obtained by deleting the ORDER=DATA option or by reordering the data so that the (1,1) observation is first.

Computational Method

The notation used in PROC CATMOD differs slightly from that used in the literature. The following table provides a summary of the basic dimensions and the notation for a contingency table. See the section "Computational Formulas" on page 1168 for a complete description.

Summary of Basic Dimensions

s	=	number of populations or samples (= number of rows in the underlying contingency table)
r	=	number of response categories (= number of columns in the underlying contingency table)
q	=	number of response functions computed for each population
d	=	number of parameters

Notation

$\mathbf{j}$	denotes a column vector of 1s.
$\mathbf{J}$	denotes a square matrix of 1s.
$\sum_k$	is the sum over all the possible values of k.
n_i	denotes the row sum $\sum_j n_{ij}$.
$\mathbf{DIAG}_n(\mathbf{p})$	is the diagonal matrix formed from the first n elements of the vector $\mathbf{p}$.
$\mathbf{DIAG}_n^{-1}(\mathbf{p})$	is the inverse of $\mathbf{DIAG}_n(\mathbf{p})$.
$\mathbf{DIAG}(\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_k)$	denotes a block diagonal matrix with the $\mathbf{A}$ matrices on the main diagonal.

Input data can be represented by a contingency table, as shown in Table 28.5.

Table 28.5 Input Data Represented by a Contingency Table

	Response				
Population	**1**	**2**	$\cdots$	r	**Total**
1	n_{11}	n_{12}	$\cdots$	n_{1r}	n_1
2	n_{21}	n_{22}	$\cdots$	n_{2r}	n_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
s	n_{s1}	n_{s2}	$\cdots$	n_{sr}	n_s

Computational Formulas

The following formulas are shown for each population and for all populations combined.

Source	Formula	Dimension
Probability Estimates		
jth response	$p_{ij} = \frac{n_{ij}}{n_i}$	1×1
ith population	$\mathbf{p}_i = \begin{bmatrix} p_{i1} \\ p_{i2} \\ \vdots \\ p_{ir} \end{bmatrix}$	$r \times 1$
all populations	$\mathbf{p} = \begin{bmatrix} \mathbf{p}_1 \\ \mathbf{p}_2 \\ \vdots \\ \mathbf{p}_s \end{bmatrix}$	$sr \times 1$
Variance of Probability Estimates		
ith population	$\mathbf{V}_i = \frac{1}{n_i}(\mathbf{DIAG}(\mathbf{p}_i) - \mathbf{p}_i\mathbf{p}_i')$	$r \times r$
all populations	$\mathbf{V} = \mathbf{DIAG}(\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_s)$	$sr \times sr$
Response Functions		
ith population	$\mathbf{F}_i = \mathbf{F}(\mathbf{p}_i)$	$q \times 1$
all populations	$\mathbf{F} = \begin{bmatrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \vdots \\ \mathbf{F}_s \end{bmatrix}$	$sq \times 1$
Derivative of Function with Respect to Probability Estimates		
ith population	$\mathbf{H}_i = \frac{\partial \mathbf{F}(\mathbf{p}_i)}{\partial \mathbf{p}_i}$	$q \times r$
all populations	$\mathbf{H} = \mathbf{DIAG}(\mathbf{H}_1, \mathbf{H}_2, \ldots, \mathbf{H}_s)$	$sq \times sr$
Variance of Functions		
ith population	$\mathbf{S}_i = \mathbf{H}_i\mathbf{V}_i\mathbf{H}_i'$	$q \times q$
all populations	$\mathbf{S} = \mathbf{DIAG}(\mathbf{S}_1, \mathbf{S}_2, \ldots, \mathbf{S}_s)$	$sq \times sq$

Source	Formula	Dimension
Inverse Variance of Functions		
ith population	$\mathbf{S}^i = (\mathbf{S}_i)^{-1}$	$q \times q$
all populations	$\mathbf{S}^{-1} = \mathbf{DIAG}(\mathbf{S}^1, \mathbf{S}^2, \ldots, \mathbf{S}^s)$	$sq \times sq$

Derivative Table for Compound Functions: Y=F(G(p))

In the following table, let $\mathbf{G}(\mathbf{p})$ be a vector of functions of $\mathbf{p}$, and let $\mathbf{D}$ denote $\partial\mathbf{G}/\partial\mathbf{p}$, which is the first derivative matrix of $\mathbf{G}$ with respect to $\mathbf{p}$:

Function	$\mathbf{Y} = \mathbf{F}(\mathbf{G})$	Derivative $(\partial\mathbf{Y}/\partial\mathbf{p})$
Multiply matrix	$\mathbf{Y} = \mathbf{A} * \mathbf{G}$	$\mathbf{A} * \mathbf{D}$
Logarithm	$\mathbf{Y} = \mathbf{LOG}(\mathbf{G})$	$\mathbf{DIAG}^{-1}(\mathbf{G}) * \mathbf{D}$
Exponential	$\mathbf{Y} = \mathbf{EXP}(\mathbf{G})$	$\mathbf{DIAG}(\mathbf{Y}) * \mathbf{D}$
Add constant	$\mathbf{Y} = \mathbf{G} + \mathbf{A}$	$\mathbf{D}$

Default Response Functions: Generalized Logits

In the following table, subscripts i for the population are suppressed. Also denote $f_j = \log\left(\frac{p_j}{p_r}\right)$ for $j = 1, \ldots, r-1$ for each population $i = 1, \ldots, s$.

Formula
Inverse of Response Functions for a Population
$p_j = \dfrac{\exp(f_j)}{1 + \sum_k \exp(f_k)}$ for $j = 1, \ldots, r-1$
$p_r = \dfrac{1}{1 + \sum_k \exp(f_k)}$
Form of F and Derivative for a Population
$\mathbf{F} = \mathbf{KLOG}(\mathbf{p}) = (\mathbf{I}_{r-1}, -\mathbf{j})\ \mathbf{LOG}(\mathbf{p})$
$\mathbf{H} = \dfrac{\partial\mathbf{F}}{\partial\mathbf{p}} = \left(\mathbf{DIAG}_{r-1}^{-1}(\mathbf{p}), \dfrac{-1}{p_r}\mathbf{j}\right)$

Formula
Covariance Results for a Population
$\mathbf{S} = \mathbf{HVH}'$
$= \frac{1}{n}\left(\mathbf{DIAG}_{r-1}^{-1}(\mathbf{p}) + \frac{1}{p_r}\mathbf{J}_{r-1}\right)$ where $\mathbf{V}, \mathbf{H},$ and $\mathbf{J}$ are as previously defined.
$\mathbf{S}^{-1} = n(\mathbf{DIAG}_{r-1}(\mathbf{p}) - \mathbf{qq}')$, where $\mathbf{q} = \mathbf{DIAG}_{r-1}(\mathbf{p})\,\mathbf{j}$
$\mathbf{S}^{-1}\mathbf{F} = n\mathbf{DIAG}_{r-1}(\mathbf{p})\mathbf{F} - (n\sum_j p_j f_j)\,\mathbf{q}$
$\mathbf{F}'\mathbf{S}^{-1}\mathbf{F} = n\sum_j p_j f_j^2 - n(\sum_j p_j f_j)^2$

The following calculations are shown for each population and then for all populations combined:

Source	Formula	Dimension
Design Matrix		
ith population	$\mathbf{X}_i$	$q \times d$
all populations	$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \\ \vdots \\ \mathbf{X}_s \end{bmatrix}$	$sq \times d$
Crossproduct of Design Matrix		
ith population	$\mathbf{C}_i = \mathbf{X}_i{}'\mathbf{S}^i\mathbf{X}_i$	$d \times d$
all populations	$\mathbf{C} = \mathbf{X}'\mathbf{S}^{-1}\mathbf{X} = \sum_i \mathbf{C}_i$	$d \times d$

In the following table, z_p is the $100p$th percentile of the standard normal distribution:

Formula	Dimension
Crossproduct of Design Matrix with Function	
$\mathbf{R} = \mathbf{X}'\mathbf{S}^{-1}\mathbf{F} = \sum_i \mathbf{X}_i{}'\mathbf{S}^i\mathbf{F}_i$	$d \times 1$
Weighted Least Squares Estimates	
$\mathbf{b} = \mathbf{C}^{-1}\mathbf{R} = (\mathbf{X}'\mathbf{S}^{-1}\mathbf{X})^{-1}(\mathbf{X}'\mathbf{S}^{-1}\mathbf{F})$	$d \times 1$

Formula	Dimension
Covariance of Weighted Least Squares Estimates	
$\mathbf{COV(b)} = \mathbf{C}^{-1}$	$d \times d$
Wald Confidence Limits for Parameter Estimates	
$b_k \pm z_{1-\alpha/2}\mathbf{C}^{-1}_{kk}$	$k = 1, \ldots, d$
Predicted Response Functions	
$\hat{\mathbf{F}} = \mathbf{Xb}$	$sq \times 1$
Covariance of Predicted Response Functions	
$\mathbf{V}_{\hat{\mathbf{F}}} = \mathbf{XC}^{-1}\mathbf{X}'$	$sq \times sq$
Residual Chi-Square	
$\text{RSS} = \mathbf{F}'\mathbf{S}^{-1}\mathbf{F} - \hat{\mathbf{F}}'\mathbf{S}^{-1}\hat{\mathbf{F}}$	1×1
Chi-Square for H_0: $\mathbf{L}\boldsymbol{\beta} = \mathbf{0}$	
$\text{Q} = (\mathbf{Lb})'(\mathbf{LC}^{-1}\mathbf{L}')^{-1}(\mathbf{Lb})$	1×1

Maximum Likelihood Method

Let $\mathbf{C}$ be the Hessian matrix and $\mathbf{G}$ be the gradient of the log-likelihood function (both functions of $\boldsymbol{\pi}$ and the parameters $\boldsymbol{\beta}$). Let $\mathbf{p}_i^*$ denote the vector containing the first $r-1$ sample proportions from population i, and let $\boldsymbol{\pi}_i^*$ denote the corresponding vector of probability estimates from the current iteration. Starting with the least squares estimates $\mathbf{b}_0$ of $\boldsymbol{\beta}$ (if you use the ML and WLS options; with the ML option alone, the procedure starts with $\mathbf{0}$), the probabilities $\boldsymbol{\pi}(\mathbf{b})$ are computed, and $\mathbf{b}$ is calculated iteratively by the Newton-Raphson method until it converges (see the EPSILON= option). The factor λ is a step-halving factor that equals one at the start of each iteration. For any iteration in which the likelihood decreases, PROC CATMOD uses a series of subiterations in which λ is iteratively divided by two. The subiterations continue until the likelihood is greater than that of the previous iteration. If the likelihood has not reached that point after 10 subiterations, then convergence is assumed, and a warning message is displayed.

Sometimes, infinite parameters are present in the model, either because of the presence of one or more zero frequencies or because of a poorly specified model with collinearity among the estimates. If an estimate is tending toward infinity, then PROC CATMOD flags the parameter as infinite and holds the estimate fixed in subsequent iterations. PROC CATMOD regards a parameter to be infinite when two conditions apply:

- The absolute value of its estimate exceeds five divided by the range of the corresponding variable.
- The standard error of its estimate is at least three times greater than the estimate itself.

The estimator of the asymptotic covariance matrix of the maximum likelihood predicted probabilities is given by Imrey, Koch, and Stokes (1981, eq. 2.18).

The following equations summarize the method:

$$\mathbf{b}_{k+1} = \mathbf{b}_k - \lambda \mathbf{C}^{-1}\mathbf{G}$$

where

$$\begin{aligned}
\mathbf{C} &= \mathbf{X}'\mathbf{S}^{-1}(\boldsymbol{\pi})\mathbf{X} \\
\mathbf{N} &= \begin{bmatrix} n_1(\mathbf{p}_1^* - \boldsymbol{\pi}_1^*) \\ \vdots \\ n_s(\mathbf{p}_s^* - \boldsymbol{\pi}_s^*) \end{bmatrix} \\
\mathbf{G} &= \mathbf{X}'\mathbf{N}
\end{aligned}$$

Iterative Proportional Fitting

The algorithm used by PROC CATMOD for iterative proportional fitting is described in Bishop, Fienberg, and Holland (1975), Haberman (1972), and Agresti (2002). To illustrate the method, consider the observed three-dimensional table $\{n_{ijk}\}$ for the variables X, Y, and Z, and the following hierarchical model:

$$\log(m_{ijk}) = \mu + \lambda_i^X + \lambda_j^Y + \lambda_k^Z + \lambda_{ij}^{XY} + \lambda_{ik}^{XZ} + \lambda_{jk}^{YZ}$$

The following statements request that PROC CATMOD use IPF to fit the preceding model:

```
model X*Y*Z = _response_ / ml=ipf;
loglin X|Y|Z@2;
```

Begin with a table of initial cell estimates $\{\hat{m}_{ijk}^{(0)}\}$. PROC CATMOD produces the initial estimates by setting the n_{sz} structural zero cells to 0 and all other cells to $n/(n_c - n_{sz})$, where n is the total weight of the table and n_c is the total number of cells in the table. Iteratively adjust the estimates at step $s-1$ to the observed marginal tables specified in the model by cycling through the following three-stage process to produce the estimates at step s:

$$\begin{aligned}
\hat{m}_{ijk}^{(s_1)} &= \hat{m}_{ijk}^{(s-1)} \frac{n_{ij\cdot}}{\hat{m}_{ij\cdot}^{(s-1)}} \\
\hat{m}_{ijk}^{(s_2)} &= \hat{m}_{ijk}^{(s_1)} \frac{n_{i\cdot k}}{\hat{m}_{i\cdot k}^{(s_1)}} \\
\hat{m}_{ijk}^{(s)} &= \hat{m}_{ijk}^{(s_2)} \frac{n_{\cdot jk}}{\hat{m}_{\cdot jk}^{(s_2)}}
\end{aligned}$$

The subscript "·" indicates summation over the missing subscript. The log-likelihood l_s is estimated at each step s by

$$l_s = \sum_{i,j,k} n_{ijk} \log\left(\frac{\hat{m}_{ijk}^{(s)}}{n}\right)$$

When the function $|(l_{s-1} - l_s)/l_{s-1}|$ is less than 10^{-8}, the iterations terminate. You can change the comparison value with the EPSILON= option, and you can change the convergence criterion with the CONVCRIT= option. The option CONVCRIT=CELL uses the maximum cell difference

$$\max_{i,j,k} |\hat{m}_{ijk}^{(s-1)} - \hat{m}_{ijk}^{(s)}|$$

as the criterion while the option CONVCRIT=MARGIN computes the maximum difference of the margins

$$\text{Maximum of } \left\{ \max_{i,j} |\hat{m}_{ij\cdot}^{(s-1)} - \hat{m}_{ij\cdot}^{(s)}|, \max_{i,k} |\hat{m}_{i\cdot k}^{(s-1)} - \hat{m}_{i\cdot k}^{(s)}|, \max_{j,k} |\hat{m}_{\cdot jk}^{(s-1)} - \hat{m}_{\cdot jk}^{(s)}| \right\}$$

Memory and Time Requirements

The memory and time required by PROC CATMOD are proportional to the number of parameters in the model.

Displayed Output

PROC CATMOD displays the following information in the "Data Summary" table:

- the response effect
- the weight variable, if one is specified
- the data set name
- the number of response levels
- the number of samples or populations
- the total frequency, which is the total sample size
- the number of observations from the data set (the number of data records)
- the frequency of missing observations, labeled as "Frequency Missing"

Except for the analysis of variance table, all of the following items can be displayed or suppressed, depending on your specification of statements and options.

- The ONEWAY option produces the "One-Way Frequencies" table, which displays the frequencies of each variable value used in the analysis.
- The populations (or samples) are defined in a table labeled "Population Profiles." The sample size and the values of the defining variables are displayed for each sample. This table is suppressed if the NOPROFILE option is specified.
- The observed responses are defined in a table labeled "Response Profiles." The values of the defining variables are displayed for each response. This table is suppressed if the NOPROFILE option is specified.
- If the FREQ option is specified, then the "Response Frequencies" table is displayed, which shows the frequency of each response for each population.
- If the PROB option is specified, then the "Response Probabilities" table is produced. This table displays the probability of each response for each population.
- If the COV option is specified, the "Response Functions, Covariance Matrix" table, which shows the covariance matrix of the response functions for each sample, is displayed.
- If the DESIGN option is specified, the response functions are displayed in the "Response Functions, Design Matrix" table. If the COV option is also specified, the response functions are displayed in the "Response Functions, Covariance Matrix" table.
- If the DESIGN option is specified, the design matrix is displayed in the "Response Functions, Design Matrix" table, and if a log-linear model is being fit, the _RESPONSE_ matrix is displayed in the "_Response_ Matrix" table. If the model type is AVERAGED, then the design matrix is displayed with $q * s$ rows, assuming q response functions for each of s populations. Otherwise, the design matrix is displayed with only s rows since the model is the same for each of the q response functions.
- The "X′*Inv(S)*X" matrix is displayed for weighted least squares analyses if the XPX option is specified.
- The "Analysis of Variance" table for the weighted least squares analysis reports the results of significance tests for each of the *design-effects* on the right side of the MODEL statement. If _RESPONSE_ is a *design-effect* and is defined explicitly in the LOGLIN, FACTORS, or REPEATED statement, then the table contains test statistics for the individual effects constituting the _RESPONSE_ effect. If the design matrix is input directly, then the content of the displayed output depends on whether you specify any subsets of the parameters to be tested. If you specify one or more subsets, then the table contains one test for each subset. Otherwise, the table contains one test for the effect MODEL | MEAN. In every case, the table also contains the residual goodness-of-fit test. Produced for each test of significance are the source of variation, the number of degrees of freedom (DF), the Wald chi-square value, and the significance probability (Pr > ChiSq).
- The "Analysis of Weighted Least Squares Estimates" table lists, for each parameter in the model, the least squares estimate, the estimated standard error of the parameter estimate, the Wald chi-square value (calculated as $((\text{parameter estimate})/(\text{standard error}))^2$) for testing that the parameter is zero, and the significance probability (Pr > ChiSq) of the test. If the CLPARM option is specified, then 95% Wald confidence intervals are displayed.

Each row in the table is labeled with the parameter (the model effect and the class levels) and the response function number; however, if the NOPREDVAR option or a REPEATED or FACTORS statement is specified or if the design matrix is directly input, the rows are labeled by the effect in the model for which parameters are formed and the parameter number.

- The "Covariance Matrix of the Parameter Estimates" table for the weighted least squares analysis displays the estimated covariance matrix of the least squares estimates of the parameters, provided that the COVB option is specified.

- The "Correlation Matrix of the Parameter Estimates" table for the weighted least squares analysis displays the estimated correlation matrix of the least squares estimates of the parameters, provided that the CORRB option is specified.

- The "Maximum Likelihood Analysis" table is produced when the ML and ITPRINT options are specified for the standard response functions. It displays the iteration number, the number of step-halving sub-iterations, -2 log likelihood for that iteration, the convergence criterion, and the parameter estimates for each iteration.

- The "Maximum Likelihood Analysis of Variance" table, displayed when the ML option is specified for the standard response functions, is similar to the table produced for the least squares analysis. The Wald chi-square test for each effect is based on the information matrix from the likelihood calculations. The likelihood ratio statistic compares the specified model with the unrestricted (saturated) model and is an appropriate goodness-of-fit test for the model.

- The "Analysis of Maximum Likelihood Estimates" table, displayed when the ML option is specified for the standard response functions, is similar to the one produced for the least squares analysis. The table includes the maximum likelihood estimates, the estimated standard errors based on the information matrix, and the Wald chi-square statistics based on estimated standard errors.

- The "Covariance Matrix of the Maximum Likelihood Estimates" table displays the estimated covariance matrix of the maximum likelihood estimates of the parameters, provided that the COVB and ML options are specified for the standard response functions.

- The "Correlation Matrix of the Maximum Likelihood Estimates" table displays the estimated correlation matrix of the maximum likelihood estimates of the parameters, provided that the CORRB and ML options are specified for the standard response functions.

- For each source of variation specified in a CONTRAST statement, the "Contrasts" table lists the label for the source (Contrast), the number of degrees of freedom (DF), the Wald chi-square value, and the significance probability (Pr > ChiSq). If the ESTIMATE= option is specified, the "Analysis of Contrasts" table displays, for each row of the contrast, the label (Contrast), the type (PARM or EXP), the row of the contrast, the estimate and its standard error, a Wald confidence interval, the Wald chi-square, and the p-value (Pr > ChiSq) for 1 degree of freedom.

- Specification of the PREDICT option in the MODEL statement has the following effect. Produced for each response function within each population are the observed and predicted function values, their standard errors, and the residual (observed minus predicted). If the response functions are the default ones (generalized logits), additional information displayed

for each response within each population includes the observed and predicted cell probabilities, their standard errors, and the residual. However, specifying PRED=FREQ in the MODEL statement results in the display of the predicted cell frequencies rather than the predicted cell probabilities. The displayed output includes the population profiles and, for the response function table, the function number, while the probability and frequency tables display the response profiles. If the NOPREDVAR option is specified in the MODEL statement, the population profiles are replaced with the sample numbers, and the response profiles are replaced with the labels "P*n*" for the *n*th cell probability, and "F*n*" for the *n*th cell frequency.

- When there are multiple RESPONSE statements, the output for each statement starts on a new page. For each RESPONSE statement, the corresponding title, if specified, is displayed at the top of each page.
- If the ADDCELL= option is specified in the MODEL statement, and if there is a weighted least squares analysis specified, the adjusted sample size for each population (with number added to each cell) is labeled "Adjusted Sample Size" in the "Population Profiles" table. Similarly, the adjusted response frequencies and probabilities are displayed in the "Adjusted Response Frequencies" and "Adjusted Response Probabilities" tables, respectively.
- If _RESPONSE_ is defined explicitly in the LOGLIN, FACTORS, or REPEATED statement, then the definition is displayed as a note whenever _RESPONSE_ appears in the output.

ODS Table Names

PROC CATMOD assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 28.10 ODS Tables Produced by PROC CATMOD

ODS Table Name	Description	Statement	Option
ANOVA	Analysis of variance	MODEL	default
Contrasts	Contrasts	CONTRAST	default
ContrastEstimates	Analysis of contrasts	CONTRAST	ESTIMATE=
ConvergenceStatus	Convergence status	MODEL	ML
CorrB	Correlation matrix of the estimates	MODEL	CORRB
CovB	Covariance matrix of the estimates	MODEL	COVB
DataSummary	Data summary	PROC	default
Estimates	Analysis of estimates	MODEL	default, unless NOPARM
MaxLikelihood	Maximum likelihood analysis	MODEL	ML and ITPRINT
OneWayFreqs	One-way frequencies	MODEL	ONEWAY
PopProfiles	Population profiles	MODEL	default, unless NOPROFILE
PredictedFreqs	Predicted frequencies	MODEL	PRED=FREQ
PredictedProbs	Predicted probabilities	MODEL	PREDICT or PRED=PROB
PredictedValues	Predicted values	MODEL	PREDICT or PRED=
ResponseCov	Response functions, covariance matrix	MODEL	COV
ResponseDesign	Response functions, design matrix	MODEL	DESIGN, unless NODESIGN
ResponseFreqs	Response frequencies	MODEL	FREQ
ResponseMatrix	_RESPONSE_ matrix	MODEL and LOGLIN	DESIGN, unless NORESPONSE
ResponseProbs	Response probabilities	MODEL	PROB
ResponseProfiles	Response profiles	MODEL	default, unless NOPROFILE
XPX	**X**'*Inv(**S**)***X** matrix	MODEL	XPX, for WLS*

* WLS estimation is the default for response functions other than the default (generalized logits).

Examples: CATMOD Procedure

Example 28.1: Linear Response Function, r=2 Responses

In an example from Ries and Smith (1963), the choice of detergent brand (Brand= M or X) is related to three other categorical variables: the softness of the laundry water (Softness= soft, medium, or hard), the temperature of the water (Temperature= high or low), and whether the subject was a previous user of Brand M (Previous= yes or no). The linear response function, which could also be specified as RESPONSE MARGINALS, yields one probability, Pr(brand preference=M), as the response function to be analyzed. Two models are fit in this example: the first model is a saturated one, containing all of the main effects and interactions, while the second is a reduced model containing only the main effects. The following statements produce Output 28.1.1 through Output 28.1.4:

```
data detergent;
   input Softness $ Brand $ Previous $ Temperature $ Count @@;
   datalines;
soft X yes high 19   soft X yes low 57
soft X no  high 29   soft X no  low 63
soft M yes high 29   soft M yes low 49
soft M no  high 27   soft M no  low 53
med  X yes high 23   med  X yes low 47
med  X no  high 33   med  X no  low 66
med  M yes high 47   med  M yes low 55
med  M no  high 23   med  M no  low 50
hard X yes high 24   hard X yes low 37
hard X no  high 42   hard X no  low 68
hard M yes high 43   hard M yes low 52
hard M no  high 30   hard M no  low 42
;

title 'Detergent Preference Study';
proc catmod data=detergent;
   response 1 0;
   weight Count;
   model Brand=Softness|Previous|Temperature / freq prob;
   title2 'Saturated Model';
run;
```

The "Data Summary" table (Output 28.1.1) indicates that you have two response levels and twelve populations.

Output 28.1.1 Detergent Preference Study: Linear Model Analysis

Detergent Preference Study
Saturated Model

The CATMOD Procedure

Data Summary

Response	Brand	Response Levels	2
Weight Variable	Count	Populations	12
Data Set	DETERGENT	Total Frequency	1008
Frequency Missing	0	Observations	24

The "Population Profiles" table in Output 28.1.2 displays the ordering of independent variable levels as used in the table of parameter estimates.

Output 28.1.2 Population Profiles

Population Profiles

Sample	Softness	Previous	Temperature	Sample Size
1	hard	no	high	72
2	hard	no	low	110
3	hard	yes	high	67
4	hard	yes	low	89
5	med	no	high	56
6	med	no	low	116
7	med	yes	high	70
8	med	yes	low	102
9	soft	no	high	56
10	soft	no	low	116
11	soft	yes	high	48
12	soft	yes	low	106

Since Brand M is the first level in the "Response Profiles" table (Output 28.1.3), the RESPONSE statement causes Pr(Brand=M) to be the single response function modeled.

Output 28.1.3 Response Profiles, Frequencies, and Probabilities

Response Profiles

Response	Brand
1	M
2	X

Output 28.1.3 *continued*

Response Frequencies

Sample	Response Number 1	2
1	30	42
2	42	68
3	43	24
4	52	37
5	23	33
6	50	66
7	47	23
8	55	47
9	27	29
10	53	63
11	29	19
12	49	57

Response Probabilities

Sample	Response Number 1	2
1	0.41667	0.58333
2	0.38182	0.61818
3	0.64179	0.35821
4	0.58427	0.41573
5	0.41071	0.58929
6	0.43103	0.56897
7	0.67143	0.32857
8	0.53922	0.46078
9	0.48214	0.51786
10	0.45690	0.54310
11	0.60417	0.39583
12	0.46226	0.53774

The "Analysis of Variance" table in Output 28.1.4 shows that all of the interactions are nonsignificant.

Output 28.1.4 Analysis of Variance

Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	983.13	<.0001
Softness	2	0.09	0.9575
Previous	1	22.68	<.0001
Softness*Previous	2	3.85	0.1457
Temperature	1	3.67	0.0555
Softness*Temperature	2	0.23	0.8914
Previous*Temperature	1	2.26	0.1324
Softnes*Previou*Temperat	2	0.76	0.6850
Residual	0	.	.

Therefore, a main-effects model is fit with the following statements:

```
    model Brand=Softness Previous Temperature
        / clparm noprofile design;
    title2 'Main-Effects Model';
run;
quit;
```

The PROC CATMOD statement is not required due to the interactive capability of the CATMOD procedure. The NOPROFILE option suppresses the redisplay of the "Response Profiles" table. The CLPARM option produces 95% confidence limits for the parameter estimates. Output 28.1.5 through Output 28.1.7 are produced.

The design matrix in Output 28.1.5 displays the results of the differential-effects modeling used in PROC CATMOD.

Output 28.1.5 Main-Effects Design Matrix

Detergent Preference Study
Main-Effects Model

The CATMOD Procedure

Data Summary

Response	Brand	Response Levels	2
Weight Variable	Count	Populations	12
Data Set	DETERGENT	Total Frequency	1008
Frequency Missing	0	Observations	24

Output 28.1.5 *continued*

Response Functions and Design Matrix

Sample	Response Function	Design Matrix 1	2	3	4	5
1	0.41667	1	1	0	1	1
2	0.38182	1	1	0	1	-1
3	0.64179	1	1	0	-1	1
4	0.58427	1	1	0	-1	-1
5	0.41071	1	0	1	1	1
6	0.43103	1	0	1	1	-1
7	0.67143	1	0	1	-1	1
8	0.53922	1	0	1	-1	-1
9	0.48214	1	-1	-1	1	1
10	0.45690	1	-1	-1	1	-1
11	0.60417	1	-1	-1	-1	1
12	0.46226	1	-1	-1	-1	-1

The analysis of variance table in Output 28.1.6 shows that previous use of Brand M, together with the temperature of the laundry water, is a significant factor in whether a subject prefers Brand M laundry detergent. The table also shows that the additive model fits since the goodness-of-fit statistic (the residual chi-square) is nonsignificant.

Output 28.1.6 ANOVA Table for the Main-Effects Model

Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	1004.93	<.0001
Softness	2	0.24	0.8859
Previous	1	20.96	<.0001
Temperature	1	3.95	0.0468
Residual	7	8.26	0.3100

The chi-square test in Output 28.1.7 shows that the Softness parameters are not significantly different from zero; as expected, the Wald confidence limits for these two estimates contain zero. So softness of the water is not a factor in choosing Brand M.

Output 28.1.7 WLS Estimates for the Main-Effects Model

Analysis of Weighted Least Squares Estimates

Parameter		Estimate	Standard Error	Chi-Square	Pr > ChiSq	95% Confidence Limits	
Intercept		0.5080	0.0160	1004.93	<.0001	0.4766	0.5394
Softness	hard	−0.00256	0.0218	0.01	0.9066	−0.0454	0.0402
	med	0.0104	0.0218	0.23	0.6342	−0.0323	0.0530
Previous	no	−0.0711	0.0155	20.96	<.0001	−0.1015	−0.0407
Temperature	high	0.0319	0.0161	3.95	0.0468	0.000446	0.0634

The negative coefficient for Previous (−0.0711) indicates that the first level of Previous (which is shown to be 'no') is associated with a smaller probability of preferring Brand M than the second level of Previous (with coefficient constrained to be 0.0711 since the parameter estimates for a given effect must sum to zero). In other words, previous users of Brand M are much more likely to prefer it than those who have never used it before.

Similarly, the positive coefficient for Temperature indicates that the first level of Temperature (which, from the "Population Profiles" table, is 'high') has a larger probability of preferring Brand M than the second level of Temperature. In other words, those who do their laundry in hot water are more likely to prefer Brand M than those who do their laundry in cold water.

Example 28.2: Mean Score Response Function, r=3 Responses

Four surgical operations for duodenal ulcers are compared in a clinical trial at four hospitals. The operations performed are as follows: Treatment=a, drainage and vagotomy; Treatment=b, 25% resection and vagotomy; Treatment=c, 50% resection and vagotomy; and Treatment=d, 75% resection. The response is severity of an undesirable complication called "dumping syndrome." The data in the following statements are from Grizzle, Starmer, and Koch (1969, pp. 489–504).

```
data operate;
   input Hospital Treatment $ Severity $ wt @@;
   datalines;
1 a none 23    1 a slight  7    1 a moderate 2
1 b none 23    1 b slight 10    1 b moderate 5
1 c none 20    1 c slight 13    1 c moderate 5
1 d none 24    1 d slight 10    1 d moderate 6
2 a none 18    2 a slight  6    2 a moderate 1
2 b none 18    2 b slight  6    2 b moderate 2
2 c none 13    2 c slight 13    2 c moderate 2
2 d none  9    2 d slight 15    2 d moderate 2
3 a none  8    3 a slight  6    3 a moderate 3
3 b none 12    3 b slight  4    3 b moderate 4
3 c none 11    3 c slight  6    3 c moderate 2
3 d none  7    3 d slight  7    3 d moderate 4
4 a none 12    4 a slight  9    4 a moderate 1
4 b none 15    4 b slight  3    4 b moderate 2
4 c none 14    4 c slight  8    4 c moderate 3
4 d none 13    4 d slight  6    4 d moderate 4
;
```

The response variable (Severity) is ordinally scaled with three levels, so assignment of scores is appropriate (0=none, 0.5=slight, 1=moderate). For these scores, the response function yields the mean score. The following statements produce Output 28.2.1 through Output 28.2.3:

```
title 'Dumping Syndrome Data';
proc catmod data=operate order=data ;
   weight wt;
   response 0  0.5  1;
   model Severity=Treatment Hospital / freq oneway design;
   title2 'Main-Effects Model';
quit;
```

The ORDER= option is specified so that the levels of the response variable remain in the correct order. A main-effects model is fit. The ONEWAY option produces a table of the number of subjects within each variable level, and the FREQ option displays the frequency of each response within each sample (Output 28.2.1).

Output 28.2.1 Surgical Data: Analysis of Mean Scores

Dumping Syndrome Data
Main-Effects Model

The CATMOD Procedure

Data Summary

Response	Severity	Response Levels	3
Weight Variable	wt	Populations	16
Data Set	OPERATE	Total Frequency	417
Frequency Missing	0	Observations	48

One-Way Frequencies

Variable	Value	Frequency
Severity	none	240
	slight	129
	moderate	48
Treatment	a	96
	b	104
	c	110
	d	107
Hospital	1	148
	2	105
	3	74
	4	90

Population Profiles

Sample	Treatment	Hospital	Sample Size
1	a	1	32
2	a	2	25
3	a	3	17
4	a	4	22
5	b	1	38
6	b	2	26
7	b	3	20
8	b	4	20
9	c	1	38
10	c	2	28
11	c	3	19
12	c	4	25
13	d	1	40
14	d	2	26
15	d	3	18
16	d	4	23

Output 28.2.1 *continued*

Response Profiles

Response	Severity
1	none
2	slight
3	moderate

Response Frequencies

	Response Number		
Sample	1	2	3
1	23	7	2
2	18	6	1
3	8	6	3
4	12	9	1
5	23	10	5
6	18	6	2
7	12	4	4
8	15	3	2
9	20	13	5
10	13	13	2
11	11	6	2
12	14	8	3
13	24	10	6
14	9	15	2
15	7	7	4
16	13	6	4

You can use the one-way frequencies and the response profiles from Output 28.2.1 to verify that the response levels are in the desired order (none, slight, moderate) so that the response scores (0, 0.5, 1.0) are applied appropriately. If the ORDER=DATA option had not been used, the levels would have been in a different order.

The analysis of variance table (Output 28.2.2) shows that the additive model fits (since the residual chi-square is not significant), that the Treatment effect is significant, and that the Hospital effect is not significant.

Output 28.2.2 Surgical Data: Analysis of Mean Scores

Response Functions and Design Matrix

Sample	Response Function	Design Matrix 1	2	3	4	5	6	7
1	0.17188	1	1	0	0	1	0	0
2	0.16000	1	1	0	0	0	1	0
3	0.35294	1	1	0	0	0	0	1
4	0.25000	1	1	0	0	−1	−1	−1
5	0.26316	1	0	1	0	1	0	0
6	0.19231	1	0	1	0	0	1	0
7	0.30000	1	0	1	0	0	0	1
8	0.17500	1	0	1	0	−1	−1	−1
9	0.30263	1	0	0	1	1	0	0
10	0.30357	1	0	0	1	0	1	0
11	0.26316	1	0	0	1	0	0	1
12	0.28000	1	0	0	1	−1	−1	−1
13	0.27500	1	−1	−1	−1	1	0	0
14	0.36538	1	−1	−1	−1	0	1	0
15	0.41667	1	−1	−1	−1	0	0	1
16	0.30435	1	−1	−1	−1	−1	−1	−1

Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	248.77	<.0001
Treatment	3	8.90	0.0307
Hospital	3	2.33	0.5065
Residual	9	6.33	0.7069

The coefficients of Treatment in Output 28.2.3 show that the first two treatments (with negative coefficients) have lower mean scores than the last two treatments (the fourth coefficient, not shown, must be positive since the four coefficients must sum to zero). In other words, the less severe treatments (the first two) cause significantly less severe dumping syndrome complications.

Output 28.2.3 Surgical Data: Analysis of Mean Scores

Analysis of Weighted Least Squares Estimates

Parameter		Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept		0.2724	0.0173	248.77	<.0001
Treatment	a	−0.0552	0.0270	4.17	0.0411
	b	−0.0365	0.0289	1.59	0.2073
	c	0.0248	0.0280	0.78	0.3757
Hospital	1	−0.0204	0.0264	0.60	0.4388
	2	−0.0178	0.0268	0.44	0.5055
	3	0.0531	0.0352	2.28	0.1312

Example 28.3: Logistic Regression, Standard Response Function

In this data set, from Cox and Snell (1989), ingots are prepared with different heating and soaking times and tested for their readiness to be rolled. The following DATA step creates a response variable Y with value 1 for ingots that are not ready and value 0 otherwise. The explanatory variables are Heat and Soak.

```
data ingots;
   input Heat Soak nready ntotal @@;
   Count=nready;
   Y=1;
   output;
   Count=ntotal-nready;
   Y=0;
   output;
   drop nready ntotal;
   datalines;
7 1.0 0 10   14 1.0 0 31   27 1.0 1 56   51 1.0 3 13
7 1.7 0 17   14 1.7 0 43   27 1.7 4 44   51 1.7 0  1
7 2.2 0  7   14 2.2 2 33   27 2.2 0 21   51 2.2 0  1
7 2.8 0 12   14 2.8 0 31   27 2.8 1 22   51 4.0 0  1
7 4.0 0  9   14 4.0 0 19   27 4.0 1 16
;
```

Logistic regression analysis is often used to investigate the relationship between discrete response variables and continuous explanatory variables. For logistic regression, the continuous *design-effects* are declared in a DIRECT statement. The following statements produce Output 28.3.1 through Output 28.3.6:

```
title 'Maximum Likelihood Logistic Regression';
proc catmod data=ingots;
   weight Count;
   direct Heat Soak;
   model Y=Heat Soak / freq covb corrb itprint design;
quit;
```

You can verify that the populations are defined as you intended by looking at the "Population Profiles" table in Output 28.3.1.

Output 28.3.1 Maximum Likelihood Logistic Regression

```
                Maximum Likelihood Logistic Regression

                        The CATMOD Procedure

                            Data Summary

Response            Y          Response Levels     2
Weight Variable     Count      Populations        19
Data Set            INGOTS     Total Frequency   387
Frequency Missing   0          Observations       25
```

Output 28.3.1 *continued*

Population Profiles

Sample	Heat	Soak	Sample Size
1	7	1	10
2	7	1.7	17
3	7	2.2	7
4	7	2.8	12
5	7	4	9
6	14	1	31
7	14	1.7	43
8	14	2.2	33
9	14	2.8	31
10	14	4	19
11	27	1	56
12	27	1.7	44
13	27	2.2	21
14	27	2.8	22
15	27	4	16
16	51	1	13
17	51	1.7	1
18	51	2.2	1
19	51	4	1

Since the "Response Profiles" table in Output 28.3.2 shows the response level ordering as 0, 1, the default response function, the logit, is defined as $\log\left(\frac{p_{Y=0}}{p_{Y=1}}\right)$.

Output 28.3.2 Response Summaries

Response Profiles

Response	Y
1	0
2	1

Output 28.3.2 *continued*

Response Frequencies

Sample	Response Number 1	2
1	10	0
2	17	0
3	7	0
4	12	0
5	9	0
6	31	0
7	43	0
8	31	2
9	31	0
10	19	0
11	55	1
12	40	4
13	21	0
14	21	1
15	15	1
16	10	3
17	1	0
18	1	0
19	1	0

The values of the continuous variable are inserted into the design matrix (Output 28.3.3).

Output 28.3.3 Design Matrix

Response Functions and Design Matrix

Sample	Response Function	Design Matrix 1	2	3
1	2.99573	1	7	1
2	3.52636	1	7	1.7
3	2.63906	1	7	2.2
4	3.17805	1	7	2.8
5	2.89037	1	7	4
6	4.12713	1	14	1
7	4.45435	1	14	1.7
8	2.74084	1	14	2.2
9	4.12713	1	14	2.8
10	3.63759	1	14	4
11	4.00733	1	27	1
12	2.30259	1	27	1.7
13	3.73767	1	27	2.2
14	3.04452	1	27	2.8
15	2.70805	1	27	4
16	1.20397	1	51	1
17	0.69315	1	51	1.7
18	0.69315	1	51	2.2
19	0.69315	1	51	4

Seven Newton-Raphson iterations are required to find the maximum likelihood estimates (Output 28.3.4).

Output 28.3.4 Iteration History

Maximum Likelihood Analysis

Iteration	Sub Iteration	−2 Log Likelihood	Convergence Criterion	Parameter Estimates 1	2	3
0	0	536.49592	1.0000	0	0	0
1	0	152.58961	0.7156	2.1594	−0.0139	−0.003733
2	0	106.76066	0.3003	3.5334	−0.0363	−0.0120
3	0	96.692171	0.0943	4.7489	−0.0640	−0.0299
4	0	95.383825	0.0135	5.4138	−0.0790	−0.0498
5	0	95.345659	0.000400	5.5539	−0.0819	−0.0564
6	0	95.345613	4.8289E−7	5.5592	−0.0820	−0.0568
7	0	95.345613	7.728E−13	5.5592	−0.0820	−0.0568

Maximum likelihood computations converged.

The analysis of variance table (Output 28.3.5) shows that the model fits since the likelihood ratio goodness-of-fit test is nonsignificant. It also shows that the length of heating time is a significant factor with respect to readiness but that length of soaking time is not.

Output 28.3.5 Analysis of Variance Table

Maximum Likelihood Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	24.65	<.0001
Heat	1	11.95	0.0005
Soak	1	0.03	0.8639
Likelihood Ratio	16	13.75	0.6171

From the table of maximum likelihood estimates in Output 28.3.6, the fitted model is

$$\mathrm{E}(\mathrm{logit}(p)) = 5.559 - 0.082(\mathrm{Heat}) - 0.057(\mathrm{Soak})$$

For example, for Sample 1 with Heat $= 7$ and Soak $= 1$, the estimate is

$$\mathrm{E}(\mathrm{logit}(p)) = 5.559 - 0.082(7) - 0.057(1) = 4.9284$$

Output 28.3.6 Maximum Likelihood Estimates, Covariances, and Correlations

Analysis of Maximum Likelihood Estimates

Parameter	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	5.5592	1.1197	24.65	<.0001
Heat	−0.0820	0.0237	11.95	0.0005
Soak	−0.0568	0.3312	0.03	0.8639

Covariance Matrix of the Maximum Likelihood Estimates

Row	Parameter	Col1	Col2	Col3
1	Intercept	1.2537133	−0.0215664	−0.2817648
2	Heat	−0.0215664	0.0005633	0.0026243
3	Soak	−0.2817648	0.0026243	0.1097020

Correlation Matrix of the Maximum Likelihood Estimates

Row	Parameter	Col1	Col2	Col3
1	Intercept	1.00000	−0.81152	−0.75977
2	Heat	−0.81152	1.00000	0.33383
3	Soak	−0.75977	0.33383	1.00000

Predicted values of the logits, as well as the probabilities of readiness, could be obtained by specifying PRED=PROB in the MODEL statement. For the example of Sample 1 with Heat = 7 and Soak = 1, PRED=PROB would give an estimate of the probability of readiness equal to 0.9928 since

$$4.9284 = \log\left(\frac{\hat{p}}{1-\hat{p}}\right)$$

implies that

$$\hat{p} = \frac{e^{4.9284}}{1+e^{4.9284}} = 0.9928$$

As another consideration, since soaking time is nonsignificant, you could fit another model that deleted the variable Soak.

Example 28.4: Log-Linear Model, Three Dependent Variables

This analysis reproduces the predicted cell frequencies for Bartlett's data by using a log-linear model of no three-variable interaction (Bishop, Fienberg, and Holland 1975, p. 89). Cuttings of two different lengths (Length=short or long) are planted at one of two time points (Time=now or spring), and their survival status (Status=dead or alive) is recorded.

As in the text, the variable levels are simply labeled 1 and 2. The following statements produce Output 28.4.1 through Output 28.4.3:

```
data bartlett;
   input Length Time Status wt @@;
   datalines;
1 1 1 156     1 1 2  84     1 2 1 84     1 2 2 156
2 1 1 107     2 1 2 133     2 2 1 31     2 2 2 209
;

title 'Bartlett''s Data';
proc catmod data=bartlett;
   weight wt;
   model Length*Time*Status=_response_
         / noparm pred=freq;
   loglin Length|Time|Status @ 2;
   title2 'Model with No 3-Variable Interaction';
quit;
```

Output 28.4.1 Analysis of Bartlett's Data: Log-Linear Model

```
                          Bartlett's Data
                 Model with No 3-Variable Interaction

                        The CATMOD Procedure

                            Data Summary

Response            Length*Time*Status     Response Levels     8
Weight Variable     wt                     Populations         1
Data Set            BARTLETT               Total Frequency   960
Frequency Missing   0                      Observations        8

                       Population Profiles

                      Sample    Sample Size
                      ----------------------
                          1             960
```

Output 28.4.1 *continued*

Response Profiles

Response	Length	Time	Status
1	1	1	1
2	1	1	2
3	1	2	1
4	1	2	2
5	2	1	1
6	2	1	2
7	2	2	1
8	2	2	2

Maximum Likelihood Analysis

Maximum likelihood computations converged.

Maximum Likelihood Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Length	1	2.64	0.1041
Time	1	5.25	0.0220
Length*Time	1	5.25	0.0220
Status	1	48.94	<.0001
Length*Status	1	48.94	<.0001
Time*Status	1	95.01	<.0001
Likelihood Ratio	1	2.29	0.1299

The analysis of variance table shows that the model fits since the likelihood ratio test for the three-variable interaction is nonsignificant. All of the two-variable interactions, however, are significant; this shows that there is mutual dependence among all three variables.

The predicted values table (Output 28.4.2) displays observed and predicted values for the generalized logits.

Output 28.4.2 Response Function Predicted Values

Maximum Likelihood Predicted Values for Response Functions

Function Number	Observed Function	Observed Standard Error	Predicted Function	Predicted Standard Error	Residual
1	-0.29248	0.105806	-0.23565	0.098486	-0.05683
2	-0.91152	0.129188	-0.94942	0.129948	0.037901
3	-0.91152	0.129188	-0.94942	0.129948	0.037901
4	-0.29248	0.105806	-0.23565	0.098486	-0.05683
5	-0.66951	0.118872	-0.69362	0.120172	0.024113
6	-0.45199	0.110921	-0.3897	0.102267	-0.06229
7	-1.90835	0.192465	-1.73146	0.142969	-0.17688

The predicted frequencies table (Output 28.4.3) displays observed and predicted cell frequencies, their standard errors, and residuals.

Output 28.4.3 Predicted Frequencies

Maximum Likelihood Predicted Values for Frequencies

			Observed		Predicted		
Length	Time	Status	Frequency	Standard Error	Frequency	Standard Error	Residual
1	1	1	156	11.43022	161.0961	11.07379	-5.09614
1	1	2	84	8.754999	78.90386	7.808613	5.096139
1	2	1	84	8.754999	78.90386	7.808613	5.096139
1	2	2	156	11.43022	161.0961	11.07379	-5.09614
2	1	1	107	9.750588	101.9039	8.924304	5.096139
2	1	2	133	10.70392	138.0961	10.33434	-5.09614
2	2	1	31	5.47713	36.09614	4.826315	-5.09614
2	2	2	209	12.78667	203.9039	12.21285	5.09614

Example 28.5: Log-Linear Model, Structural and Sampling Zeros

This example illustrates a log-linear model of independence, by using data that contain structural zero frequencies as well as sampling (random) zero frequencies.

In a population of six squirrel monkeys, the joint distribution of genital display with respect to active or passive role was observed. The data are from Fienberg (1980, Table 8-2). Since a monkey cannot have both the active and passive roles in the same interaction, the diagonal cells of the table are structural zeros. See Agresti (2002) for more information about the quasi-independence model.

The DATA step replaces the structural zeros with missing values, and the MISSING=STRUCTURAL option is specified in the MODEL statement to remove these zeros from the analysis. The ZERO=SAMPLING option treats the off-diagonal zeros as sampling zeros. Also, the row for Monkey 't' is deleted since it contains all zeros; therefore, the cell frequencies predicted by a model of independence are also zero. In addition, the CONTRAST statement compares the behavior of the two monkeys labeled 'u' and 'v'. See the section "Structural and Sampling Zeros with Raw Data" on page 1200 for information about how to perform this analysis when you have raw data. The following statements produce Output 28.5.1 through Output 28.5.8:

```
data Display;
   input Active $ Passive $ wt @@;
   if Active ne 't';
   if Active eq Passive then wt=.;
   datalines;
r r  0   r s  1   r t  5   r u  8   r v  9   r w  0
s r 29   s s  0   s t 14   s u 46   s v  4   s w  0
t r  0   t s  0   t t  0   t u  0   t v  0   t w  0
u r  2   u s  3   u t  1   u u  0   u v 38   u w  2
v r  0   v s  0   v t  0   v u  0   v v  0   v w  1
w r  9   w s 25   w t  4   w u  6   w v 13   w w  0
;
```

```
title 'Behavior of Squirrel Monkeys';
proc catmod data=Display;
   weight wt;
   model Active*Passive=_response_ /
         missing=structural zero=sampling
         freq pred=freq noparm oneway;
   loglin Active Passive;
   contrast 'Passive, U vs. V' Passive 0 0 0 1 -1;
   contrast 'Active,  U vs. V' Active  0 0 1 -1;
   title2 'Test Quasi-Independence for the Incomplete Table';
quit;
```

Output 28.5.1 Log-Linear Model Analysis with Zero Frequencies

```
                    Behavior of Squirrel Monkeys
              Test Quasi-Independence for the Incomplete Table

                          The CATMOD Procedure

                              Data Summary

          Response            Active*Passive     Response Levels   25
          Weight Variable     wt                 Populations        1
          Data Set            DISPLAY            Total Frequency  220
          Frequency Missing   0                  Observations      25
```

The results of the ONEWAY option are shown in Output 28.5.2. Monkey 't' does not show up as a value for the Active variable since that row was removed.

Output 28.5.2 Output from the ONEWAY option

```
                          One-Way Frequencies

                    Variable    Value    Frequency
                    ----------------------------
                    Active      r               23
                                s               93
                                u               46
                                v                1
                                w               57

                    Passive     r               40
                                s               29
                                t               24
                                u               60
                                v               64
                                w                3
```

Sampling zeros are displayed as 0 in Output 28.5.4. The Response Number column corresponds to the value displayed in the "Response Profiles" table in Output 28.5.3.

Output 28.5.3 Profiles

Population Profiles

Sample	Sample Size
1	220

Response Profiles

Response	Active	Passive
1	r	s
2	r	t
3	r	u
4	r	v
5	r	w
6	s	r
7	s	t
8	s	u
9	s	v
10	s	w
11	u	r
12	u	s
13	u	t
14	u	v
15	u	w
16	v	r
17	v	s
18	v	t
19	v	u
20	v	w
21	w	r
22	w	s
23	w	t
24	w	u
25	w	v

Output 28.5.4 Frequency of Response by Response Number

Response Frequencies

	Response Number							
Sample	1	2	3	4	5	6	7	8
1	1	5	8	9	0	29	14	46

Response Frequencies

	Response Number							
Sample	9	10	11	12	13	14	15	16
1	4	0	2	3	1	38	2	0

Response Frequencies

	Response Number							
Sample	17	18	19	20	21	22	23	24
1	0	0	0	1	9	25	4	6

Response Frequencies

	Response Number
Sample	25
1	13

The analysis of variance table (Output 28.5.5) shows that the model of independence does not fit since the likelihood ratio test for the interaction is significant. In other words, active and passive behaviors of the squirrel monkeys are dependent behavior roles.

Output 28.5.5 Analysis of Variance Table

Maximum Likelihood Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Active	4	56.58	<.0001
Passive	5	47.94	<.0001
Likelihood Ratio	15	135.17	<.0001

If the model fit these data, then the contrasts in Output 28.5.6 show that monkeys ‘u’ and ‘v’ appear to have similar passive behavior patterns but very different active behavior patterns.

Output 28.5.6 Contrasts between Monkeys 'u' and 'v'

Contrasts of Maximum Likelihood Estimates

Contrast	DF	Chi-Square	Pr > ChiSq
Passive, U vs. V	1	1.31	0.2524
Active, U vs. V	1	14.87	0.0001

Output 28.5.7 displays the predicted response functions and Output 28.5.8 displays predicted cell frequencies (from the PRED=FREQ option), but since the model does not fit, these should be ignored. Note that, since the response function is the generalized logit with the 25th response as the baseline, the observed response functions for the sampling zeros are missing.

Output 28.5.7 Response Function Predicted Values

Maximum Likelihood Predicted Values for Response Functions

Function Number	Observed Function	Observed Standard Error	Predicted Function	Predicted Standard Error	Residual
1	-2.56495	1.037749	-0.97355	0.339019	-1.5914
2	-0.95551	0.526235	-1.72504	0.345438	0.769529
3	-0.48551	0.449359	-0.52751	0.309254	0.042007
4	-0.36772	0.433629	-0.73927	0.249006	0.371543
5	.	.	-3.56052	0.634104	.
6	0.802346	0.333775	0.320589	0.26629	0.481758
7	0.074108	0.385164	-0.29934	0.295634	0.37345
8	1.263692	0.314105	0.898184	0.250857	0.365508
9	-1.17865	0.571772	0.686431	0.173396	-1.86509
10	.	.	-2.13482	0.608071	.
11	-1.8718	0.759555	-0.2415	0.287218	-1.63031
12	-1.46634	0.640513	-0.10994	0.303568	-1.3564
13	-2.56495	1.037749	-0.86143	0.314794	-1.70352
14	1.072637	0.321308	0.124346	0.204345	0.94829
15	-1.8718	0.759555	-2.6969	0.617433	0.8251
16	.	.	-4.14787	1.024508	.
17	.	.	-4.01632	1.030062	.
18	.	.	-4.76781	1.032457	.
19	.	.	-3.57028	1.020794	.
20	-2.56495	1.037749	-6.60328	1.161289	4.038332
21	-0.36772	0.433629	-0.36584	0.202959	-0.00188
22	0.653926	0.34194	-0.23429	0.232794	0.888212
23	-1.17865	0.571772	-0.98577	0.239408	-0.19288
24	-0.77319	0.493548	0.211754	0.185007	-0.98494

Output 28.5.8 Predicted Frequencies

Maximum Likelihood Predicted Values for Frequencies

		Observed		Predicted		
Active	Passive	Frequency	Standard Error	Frequency	Standard Error	Residual
r	s	1	0.997725	5.259508	1.36156	-4.25951
r	t	5	2.210512	2.480726	0.691066	2.519274
r	u	8	2.776525	8.215948	1.855146	-0.21595
r	v	9	2.937996	6.648049	1.50932	2.351951
r	w	0	0	0.395769	0.240268	-0.39577
s	r	29	5.017696	19.18599	3.147915	9.814007
s	t	14	3.620648	10.32172	2.169599	3.678284
s	u	46	6.031734	34.18463	4.428706	11.81537
s	v	4	1.981735	27.66096	3.722788	-23.661
s	w	0	0	1.6467	0.952712	-1.6467
u	r	2	1.407771	10.9364	2.12322	-8.9364
u	s	3	1.720201	12.47407	2.554336	-9.47407
u	t	1	0.997725	5.883583	1.380655	-4.88358
u	v	38	5.606814	15.7673	2.684692	22.2327
u	w	2	1.407771	0.938652	0.551645	1.061348
v	r	0	0	0.219966	0.221779	-0.21997
v	s	0	0	0.250893	0.253706	-0.25089
v	t	0	0	0.118338	0.120314	-0.11834
v	u	0	0	0.391924	0.393255	-0.39192
v	w	1	0.997725	0.018879	0.021728	0.981121
w	r	9	2.937996	9.657645	1.808656	-0.65765
w	s	25	4.707344	11.01553	2.275019	13.98447
w	t	4	1.981735	5.195638	1.184452	-1.19564
w	u	6	2.415857	17.2075	2.772098	-11.2075
w	v	13	3.497402	13.92369	2.24158	-0.92369

Structural and Sampling Zeros with Raw Data

The preceding PROC CATMOD step uses cell count data as input. Prior to invoking the CATMOD procedure, structural and sampling zeros are easily identified and manipulated in a single DATA step. For the situation where structural or sampling zeros (or both) exist and the input data set is raw data, use the following steps:

1. Run PROC FREQ on the raw data (see Chapter 35, "The FREQ Procedure"). In the TABLES statement, list all dependent and independent variables, separated by asterisks, and use the SPARSE option and the OUT= option. This creates an output data set that contains all possible zero frequencies. Since the tabled output can be huge, you should also specify the NOPRINT option in the TABLES statement.

2. Use a DATA step to change the zero frequencies associated with either sampling zeros or structural zeros to missing.

3. Use the resulting data set as input to PROC CATMOD, specify the statement WEIGHT COUNT to use adjusted frequencies, and specify the ZERO= and MISSING= options to define your sampling and structural zeros.

For example, suppose the data set RawDisplay contains the raw data for the squirrel monkey data. The following statements show how to obtain the same analysis as shown previously:

```
proc freq data=RawDisplay;
   tables Active*Passive / sparse out=Combos noprint;
run;

data Combos2;
   set Combos;
   if Active ne 't';
   if Active eq Passive then count=.;
run;

proc catmod data=Combos2;
   weight count;
   model Active*Passive=_response_ /
         zero=sampling missing=structural
         freq pred=freq noparm noresponse;
   loglin Active Passive;
quit;
```

The first IF statement in the DATA step is needed only for this particular example; since observations for Monkey 't' were deleted from the Display data set, they also need to be deleted from Combos2.

Example 28.6: Repeated Measures, 2 Response Levels, 3 Populations

In this multiple-population repeated measures example, from Guthrie (1981), subjects from three groups have their responses (0 or 1) recorded in each of four trials. The analysis of the marginal probabilities is directed at assessing the main effects of the repeated measurement factor (Trial) and the independent variable (Group), as well as their interaction. Although the contingency table is incomplete (only 13 of the 16 possible responses are observed), this poses no problem in the computation of the marginal probabilities. The following statements produce Output 28.6.1:

```
data group;
   input a b c d Group wt @@;
   datalines;
1 1 1 1 2 2     0 0 0 0 2 2     0 0 1 0 1 2     0 0 1 0 2 2
0 0 0 1 1 4     0 0 0 1 2 1     0 0 0 1 3 3     1 0 0 1 2 1
0 0 1 1 1 1     0 0 1 1 2 2     0 0 1 1 3 5     0 1 0 0 1 4
0 1 0 0 2 1     0 1 0 1 2 1     0 1 0 1 3 2     0 1 1 0 3 1
1 0 0 0 1 3     1 0 0 0 2 1     0 1 1 1 2 1     0 1 1 1 3 2
1 0 1 0 1 1     1 0 1 1 2 1     1 0 1 1 3 2
;

title 'Multiple-Population Repeated Measures';
proc catmod data=group;
   weight wt;
   response marginals;
   model a*b*c*d=Group _response_ Group*_response_
         / freq;
   repeated Trial 4;
   title2 'Saturated Model';
run;
```

Output 28.6.1 Analysis of Multiple-Population Repeated Measures

```
                    Multiple-Population Repeated Measures
                               Saturated Model

                             The CATMOD Procedure

                                 Data Summary

              Response           a*b*c*d     Response Levels  13
              Weight Variable    wt          Populations       3
              Data Set           GROUP       Total Frequency  45
              Frequency Missing  0           Observations     23

                             Population Profiles

                        Sample    Group    Sample Size
                        ------------------------------
                            1     1                 15
                            2     2                 15
                            3     3                 15
```

Output 28.6.1 *continued*

Response Profiles

Response	a	b	c	d
1	0	0	0	0
2	0	0	0	1
3	0	0	1	0
4	0	0	1	1
5	0	1	0	0
6	0	1	0	1
7	0	1	1	0
8	0	1	1	1
9	1	0	0	0
10	1	0	0	1
11	1	0	1	0
12	1	0	1	1
13	1	1	1	1

Response Frequencies

	Response Number							
Sample	1	2	3	4	5	6	7	8
1	0	4	2	1	4	0	0	0
2	2	1	2	2	1	1	0	1
3	0	3	0	5	0	2	1	2

Response Frequencies

	Response Number				
Sample	9	10	11	12	13
1	3	0	1	0	0
2	1	1	0	1	2
3	0	0	0	2	0

Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	354.88	<.0001
Group	2	24.79	<.0001
Trial	3	21.45	<.0001
Group*Trial	6	18.71	0.0047
Residual	0	.	.

Output 28.6.1 *continued*

Analysis of Weighted Least Squares Estimates

Effect	Parameter	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	1	0.5833	0.0310	354.88	<.0001
Group	2	0.1333	0.0335	15.88	<.0001
	3	-0.0333	0.0551	0.37	0.5450
Trial	4	0.1722	0.0557	9.57	0.0020
	5	0.1056	0.0647	2.66	0.1028
	6	-0.0722	0.0577	1.57	0.2107
Group*Trial	7	-0.1556	0.0852	3.33	0.0679
	8	-0.0556	0.0800	0.48	0.4877
	9	-0.0889	0.0953	0.87	0.3511
	10	0.0111	0.0866	0.02	0.8979
	11	0.0889	0.0822	1.17	0.2793
	12	-0.0111	0.0824	0.02	0.8927

The analysis of variance table in Output 28.6.1 shows that there is a significant interaction between the independent variable Group and the repeated measurement factor Trial. An intermediate model (not shown) is fit in which the effects Trial and Group* Trial are replaced by Trial(Group=1), Trial(Group=2), and Trial(Group=3). Of these three effects, only the last is significant, so it is retained in the final model. The following statements produce Output 28.6.2 and Output 28.6.3:

```
    model a*b*c*d=Group _response_(Group=3)
          / noprofile noparm design;
    title2 'Trial Nested within Group 3';
quit;
```

Output 28.6.2 displays the design matrix resulting from retaining the nested effect.

Output 28.6.2 Final Model: Design Matrix

Multi-Population Repeated Measures
Trial Nested within Group 3

The CATMOD Procedure

Data Summary

Response	a*b*c*d	Response Levels	13
Weight Variable	wt	Populations	3
Data Set	GROUP	Total Frequency	45
Frequency Missing	0	Observations	23

Output 28.6.2 *continued*

Response Functions and Design Matrix

Sample	Function Number	Response Function	Design Matrix 1	2	3	4	5	6
1	1	0.73333	1	1	0	0	0	0
	2	0.73333	1	1	0	0	0	0
	3	0.73333	1	1	0	0	0	0
	4	0.66667	1	1	0	0	0	0
2	1	0.66667	1	0	1	0	0	0
	2	0.66667	1	0	1	0	0	0
	3	0.46667	1	0	1	0	0	0
	4	0.40000	1	0	1	0	0	0
3	1	0.86667	1	−1	−1	1	0	0
	2	0.66667	1	−1	−1	0	1	0
	3	0.33333	1	−1	−1	0	0	1
	4	0.06667	1	−1	−1	−1	−1	−1

The residual goodness-of-fit statistic tests the joint effect of Trial(Group=1) and Trial(Group=2). The analysis of variance table in Output 28.6.3 shows that the final model fits, that there is a significant Group effect, and that there is a significant Trial effect in Group 3.

Output 28.6.3 ANOVA Table

Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	386.94	<.0001
Group	2	25.42	<.0001
Trial(Group=3)	3	75.07	<.0001
Residual	6	5.09	0.5319

Example 28.7: Repeated Measures, 4 Response Levels, 1 Population

This example illustrates a repeated measures analysis in which there are more than two levels of response. In this study, from Grizzle, Starmer, and Koch (1969, p. 493), 7,477 women aged 30–39 are tested for vision in both right and left eyes. Since there are four response levels for each dependent variable, the RESPONSE statement computes three marginal probabilities for each dependent variable, resulting in six response functions for analysis. Since the model contains a repeated measurement factor (Side) with two levels (Right, Left), PROC CATMOD groups the functions into sets of three (=6/2). Therefore, the Side effect has three degrees of freedom (one for each marginal probability), and it is the appropriate test of marginal homogeneity. The following statements produce Output 28.7.1:

```
title 'Vision Symmetry';
data vision;
   input Right Left count @@;
   datalines;
1 1 1520    1 2  266    1 3  124    1 4  66
2 1  234    2 2 1512    2 3  432    2 4  78
3 1  117    3 2  362    3 3 1772    3 4 205
4 1   36    4 2   82    4 3  179    4 4 492
;

proc catmod data=vision;
   weight count;
   response marginals;
   model Right*Left=_response_ / freq design;
   repeated Side 2;
   title2 'Test of Marginal Homogeneity';
quit;
```

Output 28.7.1 Vision Study: Analysis of Marginal Homogeneity

```
                          Vision Symmetry
                     Test of Marginal Homogeneity

                         The CATMOD Procedure

                            Data Summary

Response            Right*Left      Response Levels    16
Weight Variable     count           Populations         1
Data Set            VISION          Total Frequency  7477
Frequency Missing   0               Observations       16

                         Population Profiles

                        Sample    Sample Size
                        ----------------------
                            1            7477
```

Output 28.7.1 *continued*

Response Profiles

Response	Right	Left
1	1	1
2	1	2
3	1	3
4	1	4
5	2	1
6	2	2
7	2	3
8	2	4
9	3	1
10	3	2
11	3	3
12	3	4
13	4	1
14	4	2
15	4	3
16	4	4

Response Frequencies

	Response Number							
Sample	1	2	3	4	5	6	7	8
1	1520	266	124	66	234	1512	432	78

Response Frequencies

	Response Number							
Sample	9	10	11	12	13	14	15	16
1	117	362	1772	205	36	82	179	492

Response Functions and Design Matrix

	Function	Response	Design Matrix					
Sample	Number	Function	1	2	3	4	5	6
1	1	0.26428	1	0	0	1	0	0
	2	0.30173	0	1	0	0	1	0
	3	0.32847	0	0	1	0	0	1
	4	0.25505	1	0	0	−1	0	0
	5	0.29718	0	1	0	0	−1	0
	6	0.33529	0	0	1	0	0	−1

Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	3	78744.17	<.0001
Side	3	11.98	0.0075
Residual	0	.	.

Output 28.7.1 *continued*

Analysis of Weighted Least Squares Estimates

Effect	Parameter	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	1	0.2597	0.00468	3073.03	<.0001
	2	0.2995	0.00464	4160.17	<.0001
	3	0.3319	0.00483	4725.25	<.0001
Side	4	0.00461	0.00194	5.65	0.0174
	5	0.00227	0.00255	0.80	0.3726
	6	-0.00341	0.00252	1.83	0.1757

The analysis of variance table in Output 28.7.1 shows that the Side effect is significant, so there is not marginal homogeneity between left-eye vision and right-eye vision. In other words, the distribution of the quality of right-eye vision differs significantly from the distribution of the quality of left-eye vision in the same subjects. The test of the Side effect is equivalent to Bhapkar's test (Agresti 1990) .

Example 28.8: Repeated Measures, Logistic Analysis of Growth Curve

The following data, from a longitudinal study reported in Koch et al. (1977), are from patients in four populations (2 diagnostic groups × 2 treatments) who are measured at three times to assess their response (n=normal or a=abnormal) to treatment:

```
title 'Growth Curve Analysis';
data growth2;
   input Diagnosis $ Treatment $ week1 $ week2 $ week4 $ count @@;
   datalines;
mild std n n n 16    severe std n n n  2
mild std n n a 13    severe std n n a  2
mild std n a n  9    severe std n a n  8
mild std n a a  3    severe std n a a  9
mild std a n n 14    severe std a n n  9
mild std a n a  4    severe std a n a 15
mild std a a n 15    severe std a a n 27
mild std a a a  6    severe std a a a 28
mild new n n n 31    severe new n n n  7
mild new n n a  0    severe new n n a  2
mild new n a n  6    severe new n a n  5
mild new n a a  0    severe new n a a  2
mild new a n n 22    severe new a n n 31
mild new a n a  2    severe new a n a  5
mild new a a n  9    severe new a a n 32
mild new a a a  0    severe new a a a  6
;
```

The analysis is directed at assessing the effect of the repeated measurement factor, Time, as well as the independent variables, Diagnosis (mild or severe) and Treatment (std or new). The RESPONSE statement is used to compute the logits of the marginal probabilities. The times used in the design matrix (0, 1, 2) correspond to the logarithms (base 2) of the actual times (1, 2, 4). The following statements produce Output 28.8.1 through Output 28.8.4:

```
proc catmod data=growth2 order=data;
   title2 'Reduced Logistic Model';
   weight count;
   population Diagnosis Treatment;
   response logit;
   model week1*week2*week4=(1 0 0 0,  /* mild, std */
                            1 0 1 0,
                            1 0 2 0,

                            1 0 0 0,  /* mild, new */
                            1 0 0 1,
                            1 0 0 2,

                            0 1 0 0,  /* severe, std */
                            0 1 1 0,
                            0 1 2 0,

                            0 1 0 0,  /* severe, new */
                            0 1 0 1,
                            0 1 0 2)
          (1='Mild diagnosis, week 1',
           2='Severe diagnosis, week 1',
           3='Time effect for std trt',
           4='Time effect for new trt')
           / freq design;
   contrast 'Diagnosis effect, week 1' all_parms 1 -1 0 0;
   contrast 'Equal time effects' all_parms 0 0 1 -1;
quit;
```

The samples and the response numbers are defined in Output 28.8.1, and the frequency distribution of the response numbers within the samples is displayed.

Output 28.8.1 Logistic Analysis of Growth Curve

```
                          Growth Curve Analysis
                          Reduced Logistic Model

                           The CATMOD Procedure

                              Data Summary

      Response            week1*week2*week4     Response Levels     8
      Weight Variable     count                 Populations         4
      Data Set            GROWTH2               Total Frequency   340
      Frequency Missing   0                     Observations       29
```

Output 28.8.1 *continued*

Population Profiles

Sample	Diagnosis	Treatment	Sample Size
1	mild	std	80
2	mild	new	70
3	severe	std	100
4	severe	new	90

Response Profiles

Response	week1	week2	week4
1	n	n	n
2	n	n	a
3	n	a	n
4	n	a	a
5	a	n	n
6	a	n	a
7	a	a	n
8	a	a	a

Output 28.8.2 displays the design matrix specified in the MODEL statement, and the observed logits of the marginal probabilities are displayed in the Response Function column.

Output 28.8.2 Response Frequencies

Response Frequencies

	Response Number							
Sample	1	2	3	4	5	6	7	8
1	16	13	9	3	14	4	15	6
2	31	0	6	0	22	2	9	0
3	2	2	8	9	9	15	27	28
4	7	2	5	2	31	5	32	6

Output 28.8.2 *continued*

Response Functions and Design Matrix

Sample	Function Number	Response Function	Design Matrix 1	2	3	4
1	1	0.05001	1	0	0	0
	2	0.35364	1	0	1	0
	3	0.73089	1	0	2	0
2	1	0.11441	1	0	0	0
	2	1.29928	1	0	0	1
	3	3.52636	1	0	0	2
3	1	−1.32493	0	1	0	0
	2	−0.94446	0	1	1	0
	3	−0.16034	0	1	2	0
4	1	−1.53148	0	1	0	0
	2	0.00000	0	1	0	1
	3	1.60944	0	1	0	2

The analysis of variance table in Output 28.8.3 shows that the data can be adequately modeled by two parameters that represent diagnosis effects at week 1 and two log-linear time effects (one for each treatment). Both of the time effects are significant.

Since the estimate of the logit for the severe diagnosis effect (parameter 2) is more negative than it is for the mild diagnosis effect (parameter 1), there is a smaller predicted probability of the first response (normal) for the severe diagnosis group.

Output 28.8.3 ANOVA and Parameter Estimates

Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Mild diagnosis, week 1	1	0.28	0.5955
Severe diagnosis, week 1	1	100.48	<.0001
Time effect for std trt	1	26.35	<.0001
Time effect for new trt	1	125.09	<.0001
Residual	8	4.20	0.8387

Analysis of Weighted Least Squares Estimates

Effect	Parameter	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Model	1	−0.0716	0.1348	0.28	0.5955
	2	−1.3529	0.1350	100.48	<.0001
	3	0.4944	0.0963	26.35	<.0001
	4	1.4552	0.1301	125.09	<.0001

The analysis of contrasts (Output 28.8.4) shows that the diagnosis effect at week 1 is highly significant. In other words, those subjects with a severe diagnosis have a significantly higher probability of abnormal response at week 1 than those subjects with a mild diagnosis.

Output 28.8.4 Contrasts

Analysis of Contrasts

Contrast	DF	Chi-Square	Pr > ChiSq
Diagnosis effect, week 1	1	77.02	<.0001
Equal time effects	1	59.12	<.0001

The analysis of contrasts (Output 28.8.4) also shows that the time effect for the standard treatment is significantly different from the one for the new treatment. The table of parameter estimates (Output 28.8.3) shows that the time effect for the new treatment (parameter 4) is stronger than it is for the standard treatment (parameter 3).

Example 28.9: Repeated Measures, Two Repeated Measurement Factors

This example, from MacMillan et al. (1981), illustrates a repeated measures analysis in which there are two repeated measurement factors. Two diagnostic procedures (standard and test) are performed on each subject, and the results of both are evaluated at each of two times as being positive or negative. In the following DATA step, std1 and std2 are the two measurements of the standard procedure, and test1 and test2 are the two measurements of the test procedure:

```
data a;
   input std1 $ test1 $ std2 $ test2 $ wt @@;
   datalines;
neg neg neg neg 509  neg neg neg pos  4  neg neg pos neg  17
neg neg pos pos   3  neg pos neg neg 13  neg pos neg pos   8
neg pos pos pos   8  pos neg neg neg 14  pos neg neg pos   1
pos neg pos neg  17  pos neg pos pos  9  pos pos neg neg   7
pos pos neg pos   4  pos pos pos neg  9  pos pos pos pos 170
;
```

For the initial model, the response functions are marginal probabilities, and the repeated measurement factors are Time and Treatment. The model is a saturated one, containing effects for Time, Treatment, and Time*Treatment. The following statements produce Output 28.9.1:

```
proc catmod data=a;
   title2 'Marginal Symmetry, Saturated Model';
   weight wt;
   response marginals;
   model std1*test1*std2*test2=_response_ / freq design noparm;
   repeated Time 2, Treatment 2 / _response_=Time Treatment
            Time*Treatment;
run;
```

The analysis of variance table in Output 28.9.1 shows that there is no significant effect of Time, either by itself or in its interaction with Treatment. The second model includes only the Treatment effect. Again, the response functions are marginal probabilities, and the repeated measurement factors are Time and Treatment.

Output 28.9.1 Diagnosis Data: Two Repeated Measurement Factors

Diagnostic Procedure Comparison
Marginal Symmetry, Saturated Model

The CATMOD Procedure

Data Summary

Response	std1*test1*std2*test2	Response Levels	15
Weight Variable	wt	Populations	1
Data Set	A	Total Frequency	793
Frequency Missing	0	Observations	15

Population Profiles

Sample	Sample Size
1	793

Response Profiles

Response	std1	test1	std2	test2
1	neg	neg	neg	neg
2	neg	neg	neg	pos
3	neg	neg	pos	neg
4	neg	neg	pos	pos
5	neg	pos	neg	neg
6	neg	pos	neg	pos
7	neg	pos	pos	pos
8	pos	neg	neg	neg
9	pos	neg	neg	pos
10	pos	neg	pos	neg
11	pos	neg	pos	pos
12	pos	pos	neg	neg
13	pos	pos	neg	pos
14	pos	pos	pos	neg
15	pos	pos	pos	pos

Output 28.9.1 *continued*

Response Frequencies

Sample	Response Number 1	2	3	4	5	6	7	8
1	509	4	17	3	13	8	8	14

Response Frequencies

Sample	Response Number 9	10	11	12	13	14	15
1	1	17	9	7	4	9	170

Response Functions and Design Matrix

Sample	Function Number	Response Function	Design Matrix 1	2	3	4
1	1	0.70870	1	1	1	1
	2	0.72383	1	1	-1	-1
	3	0.70618	1	-1	1	-1
	4	0.73897	1	-1	-1	1

Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	2385.34	<.0001
Time	1	0.85	0.3570
Treatment	1	8.20	0.0042
Time*Treatment	1	2.40	0.1215
Residual	0	.	.

A main effect model with respect to Treatment is fit. The following statements produces Output 28.9.2:

```
    title2 'Marginal Symmetry, Reduced Model';
    model std1*test1*std2*test2=_response_ / corrb design noprofile;
    repeated Time 2, Treatment 2 / _response_=Treatment;
run;
```

The analysis of variance table for the reduced model (Output 28.9.2) shows that the model fits (since the residual chi-square is nonsignificant) and that the treatment effect is significant. The negative parameter estimate for Treatment shows that the first level of treatment (std) has a smaller probability of the first response level (neg) than the second level of treatment (test). In other words, the standard diagnostic procedure gives a significantly higher probability of a positive response than the test diagnostic procedure.

Output 28.9.2 Diagnosis Data: Reduced Model

Diagnostic Procedure Comparison
Marginal Symmetry, Reduced Model

The CATMOD Procedure

Data Summary

Response	std1*test1*std2*test2	Response Levels	15
Weight Variable	wt	Populations	1
Data Set	A	Total Frequency	793
Frequency Missing	0	Observations	15

Response Functions and Design Matrix

Sample	Function Number	Response Function	Design Matrix 1	2
1	1	0.70870	1	1
	2	0.72383	1	-1
	3	0.70618	1	1
	4	0.73897	1	-1

Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	2386.97	<.0001
Treatment	1	9.55	0.0020
Residual	2	3.51	0.1731

Analysis of Weighted Least Squares Estimates

Effect	Parameter	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	1	0.7196	0.0147	2386.97	<.0001
Treatment	2	-0.0128	0.00416	9.55	0.0020

Correlation Matrix of the Parameter Estimates

Row	Col1	Col2
1	1.00000	0.04194
2	0.04194	1.00000

The next example illustrates a RESPONSE statement that, at each time, computes the sensitivity and specificity of the test diagnostic procedure with respect to the standard procedure. Since these are measures of the relative accuracy of the two diagnostic procedures, the repeated measurement factors in this case are labeled Time and Accuracy. Only 15 of the 16 possible responses are observed, so additional care must be taken in formulating the RESPONSE statement for computation of sensitivity and specificity.

The following statements produce Output 28.9.3 and Output 28.9.4:

```
      title2 'Sensitivity and Specificity Analysis, '
             'Main-Effects Model';
      model std1*test1*std2*test2=_response_ / covb design noprofile;
      repeated Time 2, Accuracy 2 / _response_=Time Accuracy;
      response exp 1 -1  0  0  0  0  0  0,
                   0  0  1 -1  0  0  0  0,
                   0  0  0  0  1 -1  0  0,
                   0  0  0  0  0  0  1 -1

                log 0 0 0 0   0 0  0   0 0 0 0   1 1 1 1,
                    0 0 0 0   0 0  0   1 1 1 1   1 1 1 1,
                    1 1 1 1   0 0  0   0 0 0 0   0 0 0 0,
                    1 1 1 1   1 1  1   0 0 0 0   0 0 0 0,
                    0 0 0 1   0 0  1   0 0 0 1   0 0 0 1,
                    0 0 1 1   0 0  1   0 0 1 1   0 0 1 1,
                    1 0 0 0   1 0  0   1 0 0 0   1 0 0 0,
                    1 1 0 0   1 1  0   1 1 0 0   1 1 0 0;
   quit;
```

For the sensitivity and specificity analysis, the four response functions displayed next to the design matrix (Output 28.9.3) represent the following:

1. sensitivity, time 1
2. specificity, time 1
3. sensitivity, time 2
4. specificity, time 2

The sensitivities and specificities are for the test diagnostic procedure relative to the standard procedure.

Output 28.9.3 Diagnosis Data: Sensitivity and Specificity Analysis

```
                    Diagnostic Procedure Comparison
          Sensitivity and Specificity Analysis, Main-Effects Model

                              The CATMOD Procedure

                                 Data Summary

      Response           std1*test1*std2*test2     Response Levels   15
      Weight Variable    wt                        Populations        1
      Data Set           A                         Total Frequency  793
      Frequency Missing  0                         Observations      15
```

Output 28.9.3 *continued*

Response Functions and Design Matrix

Sample	Function Number	Response Function	Design Matrix 1	2	3
1	1	0.82251	1	1	1
	2	0.94840	1	1	-1
	3	0.81545	1	-1	1
	4	0.96964	1	-1	-1

Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	6448.79	<.0001
Time	1	4.10	0.0428
Accuracy	1	38.81	<.0001
Residual	1	1.00	0.3178

The ANOVA table in Output 28.9.3 shows that an additive model fits, that there is a significant effect of time, and that the sensitivity is significantly different from the specificity.

Output 28.9.4 shows that the predicted sensitivities and specificities are lower for time 1 (since parameter 2 is negative). It also shows that the sensitivity is significantly less than the specificity.

Output 28.9.4 Parameter Estimates

Analysis of Weighted Least Squares Estimates

Effect	Parameter	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	1	0.7196	0.0147	2386.97	<.0001
Treatment	2	-0.0128	0.00416	9.55	0.0020

Correlation Matrix of the Parameter Estimates

Row	Col1	Col2
1	1.00000	0.04194
2	0.04194	1.00000

Output 28.9.4 *continued*

Diagnostic Procedure Comparison
Sensitivity and Specificity Analysis, Main-Effects Model

The CATMOD Procedure

Data Summary

Response	std1*test1*std2*test2	Response Levels	15
Weight Variable	wt	Populations	1
Data Set	A	Total Frequency	793
Frequency Missing	0	Observations	15

Response Functions and Design Matrix

Sample	Function Number	Response Function	Design Matrix 1	2	3
1	1	0.82251	1	1	1
	2	0.94840	1	1	-1
	3	0.81545	1	-1	1
	4	0.96964	1	-1	-1

Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	6448.79	<.0001
Time	1	4.10	0.0428
Accuracy	1	38.81	<.0001
Residual	1	1.00	0.3178

Analysis of Weighted Least Squares Estimates

Effect	Parameter	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	1	0.8892	0.0111	6448.79	<.0001
Time	2	-0.00932	0.00460	4.10	0.0428
Accuracy	3	-0.0702	0.0113	38.81	<.0001

Covariance Matrix of the Parameter Estimates

Row	Col1	Col2	Col3
1	0.00012260	0.00000229	0.00010137
2	0.00000229	0.00002116	-.00000587
3	0.00010137	-.00000587	0.00012697

Example 28.10: Direct Input of Response Functions and Covariance Matrix

This example illustrates the ability of PROC CATMOD to operate on an existing vector of functions and the corresponding covariance matrix. The estimates under investigation are composite indices summarizing the responses to 18 psychological questions pertaining to general well-being. These estimates are computed for domains corresponding to an age-by-sex cross-classification, and the covariance matrix is calculated using the method of balanced repeated replications. The analysis is directed at obtaining a description of the variation among these domain estimates. The data are from Koch and Stokes (1979).

In the following statements, the first row of the fbeing data set contains the response functions for the variables b1–b10, while the remaining rows contain the covariance matrix. From the PROC CATMOD statements, the READ option in the RESPONSE statement says that you are inputting the response functions and their covariance matrix, while the PROFILES= option in the FACTORS statement tells you that the variables b1–b5 correspond to the effects for sex='male' at the five different age groupings, and b6–b10 likewise correspond to the effects for sex='female'. See the section "Inputting Response Functions and Covariances Directly" on page 1140 for more information about using the READ option.

```
data fbeing(type=est);
   input   b1-b5   _type_ $  _name_ $  b6-b10 #2;
   datalines;
 7.93726   7.92509   7.82815   7.73696   8.16791  parms     .
 7.24978   7.18991   7.35960   7.31937   7.55184
 0.00739   0.00019   0.00146  -0.00082   0.00076  cov       b1
 0.00189   0.00118   0.00140  -0.00140   0.00039
 0.00019   0.01172   0.00183   0.00029   0.00083  cov       b2
-0.00123  -0.00629  -0.00088  -0.00232   0.00034
 0.00146   0.00183   0.01050  -0.00173   0.00011  cov       b3
 0.00434  -0.00059  -0.00055   0.00023  -0.00013
-0.00082   0.00029  -0.00173   0.01335   0.00140  cov       b4
 0.00158   0.00212   0.00211   0.00066   0.00240
 0.00076   0.00083   0.00011   0.00140   0.01430  cov       b5
-0.00050  -0.00098   0.00239  -0.00010   0.00213
 0.00189  -0.00123   0.00434   0.00158  -0.00050  cov       b6
 0.01110   0.00101   0.00177  -0.00018  -0.00082
 0.00118  -0.00629  -0.00059   0.00212  -0.00098  cov       b7
 0.00101   0.02342   0.00144   0.00369   0.00253
 0.00140  -0.00088  -0.00055   0.00211   0.00239  cov       b8
 0.00177   0.00144   0.01060   0.00157   0.00226
-0.00140  -0.00232   0.00023   0.00066  -0.00010  cov       b9
-0.00018   0.00369   0.00157   0.02298   0.00918
 0.00039   0.00034  -0.00013   0.00240   0.00213  cov      b10
-0.00082   0.00253   0.00226   0.00918   0.01921
;
```

The following statements produce Output 28.10.1:

```
proc catmod data=fbeing;
   title 'Complex Sample Survey Analysis';
   response read b1-b10;
   factors sex $ 2, age $ 5 / _response_=sex age
                              profile=(male     '25-34',
                                       male     '35-44',
                                       male     '45-54',
                                       male     '55-64',
                                       male     '65-74',
                                       female   '25-34',
                                       female   '35-44',
                                       female   '45-54',
                                       female   '55-64',
                                       female   '65-74');
   model _f_=_response_
         / design title='Main Effects for Sex and Age';
run;
```

Output 28.10.1 Health Survey Data: Using Direct Input

Complex Sample Survey Analysis

Main Effects for Sex and Age

The CATMOD Procedure

Response Functions and Design Matrix

Sample	Function Number	Response Function	Design Matrix 1	2	3	4	5	6
1	1	7.93726	1	1	1	0	0	0
	2	7.92509	1	1	0	1	0	0
	3	7.82815	1	1	0	0	1	0
	4	7.73696	1	1	0	0	0	1
	5	8.16791	1	1	-1	-1	-1	-1
	6	7.24978	1	-1	1	0	0	0
	7	7.18991	1	-1	0	1	0	0
	8	7.35960	1	-1	0	0	1	0
	9	7.31937	1	-1	0	0	0	1
	10	7.55184	1	-1	-1	-1	-1	-1

Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	28089.07	<.0001
sex	1	65.84	<.0001
age	4	9.21	0.0561
Residual	4	2.92	0.5713

Output 28.10.1 *continued*

Analysis of Weighted Least Squares Estimates

Effect	Parameter	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept	1	7.6319	0.0455	28089.07	<.0001
sex	2	0.2900	0.0357	65.84	<.0001
age	3	−0.00780	0.0645	0.01	0.9037
	4	−0.0465	0.0636	0.54	0.4642
	5	−0.0343	0.0557	0.38	0.5387
	6	−0.1098	0.0764	2.07	0.1506

The analysis of variance table in Output 28.10.1 shows that the additive model fits and that there is a significant effect of both sex and age. The following statements produce Output 28.10.2:

```
    contrast 'No Age Effect for Age<65' all_parms 0 0 1 0 0 -1,
                                        all_parms 0 0 0 1 0 -1,
                                        all_parms 0 0 0 0 1 -1;
run;
```

The analysis of the contrast shows that there is no significant difference among the four age groups that are under age 65.

Output 28.10.2 Health Survey Data: Age<65 Contrast

Complex Sample Survey Analysis

Main Effects for Sex and Age

The CATMOD Procedure

Analysis of Contrasts

Contrast	DF	Chi-Square	Pr > ChiSq
No Age Effect for Age<65	3	0.72	0.8678

The next model contains a binary age effect (under 65 versus 65 and over). The following statements produce Output 28.10.3:

```
   model _f_=(1  1  1,
              1  1  1,
              1  1  1,
              1  1  1,
              1  1 -1,
              1 -1  1,
              1 -1  1,
              1 -1  1,
              1 -1  1,
              1 -1 -1)
                       (1='Intercept' ,
                        2='Sex'       ,
                        3='Age (25-64 vs. 65-74)')
          / design title='Binary Age Effect (25-64 vs. 65-74)' ;
run;
quit;
```

Output 28.10.3 Health Survey Data: Age<65 Model

Complex Sample Survey Analysis

Binary Age Effect (25-64 vs. 65-74)

The CATMOD Procedure

Response Functions and Design Matrix

Sample	Function Number	Response Function	Design Matrix 1	2	3
1	1	7.93726	1	1	1
	2	7.92509	1	1	1
	3	7.82815	1	1	1
	4	7.73696	1	1	1
	5	8.16791	1	1	-1
	6	7.24978	1	-1	1
	7	7.18991	1	-1	1
	8	7.35960	1	-1	1
	9	7.31937	1	-1	1
	10	7.55184	1	-1	-1

Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	19087.16	<.0001
Sex	1	72.64	<.0001
Age (25-64 vs. 65-74)	1	8.49	0.0036
Residual	7	3.64	0.8198

Output 28.10.3 *continued*

Analysis of Weighted Least Squares Estimates

Effect	Parameter	Estimate	Standard Error	Chi-Square	Pr > ChiSq
Model	1	7.7183	0.0559	19087.16	<.0001
	2	0.2800	0.0329	72.64	<.0001
	3	−0.1304	0.0448	8.49	0.0036

The analysis of variance table in Output 28.10.3 shows that the model fits (note that the goodness-of-fit statistic is the sum of the previous one (Output 28.10.1) plus the chi-square for the contrast matrix in Output 28.10.2). The age and sex effects are significant. Since the second parameter in the table of estimates is positive, males (the first level for the sex variable) have a higher predicted index of well-being than females. Since the third parameter estimate is negative, those younger than age 65 (the first level of age) have a lower predicted index of well-being than those 65 and older.

Example 28.11: Predicted Probabilities

Suppose you have collected marketing research data to examine the relationship between a prospect's likelihood of buying your product and the person's education and income. Specifically, the variables are as follows:

Variable	Levels	Interpretation
Education	high, low	prospect's education level
Income	high, low	prospect's income level
Purchase	yes, no	Did prospect purchase product?

The following statements first create a data set, loan, that contains the marketing research data. Then the CATMOD procedure fits a model, obtains the parameter estimates, and obtains the predicted probabilities of interest. These statements produce Output 28.11.1 and Output 28.11.2.

```
data loan;
   input Education $ Income $ Purchase $ wt;
   datalines;
high  high  yes    54
high  high  no     23
high  low   yes    41
high  low   no     12
low   high  yes    35
low   high  no     42
low   low   yes    19
low   low   no      8
;
ods output PredictedValues=Predicted (keep=Education Income PredFunction);
proc catmod data=loan order=data;
   weight wt;
   response marginals;
   model Purchase=Education Income / pred design;
run;
proc sort data=Predicted;
   by descending PredFunction;
run;
proc print data=Predicted;
run;
```

Notice that the preceding statements use the Output Delivery System (ODS) to output the parameter estimates instead of the OUT= option, though either can be used.

Output 28.11.1 Marketing Research Data: Obtaining Predicted Probabilities

Complex Sample Survey Analysis

The CATMOD Procedure

Data Summary

Response	Purchase	Response Levels	2
Weight Variable	wt	Populations	4
Data Set	LOAN	Total Frequency	234
Frequency Missing	0	Observations	8

Population Profiles

Sample	Education	Income	Sample Size
1	high	high	77
2	high	low	53
3	low	high	77
4	low	low	27

Response Profiles

Response	Purchase
1	yes
2	no

Response Functions and Design Matrix

Sample	Response Function	Design Matrix 1	2	3
1	0.70130	1	1	1
2	0.77358	1	1	-1
3	0.45455	1	-1	1
4	0.70370	1	-1	-1

Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	418.36	<.0001
Education	1	8.85	0.0029
Income	1	4.70	0.0302
Residual	1	1.84	0.1745

Output 28.11.1 *continued*

Analysis of Weighted Least Squares Estimates

Parameter		Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept		0.6481	0.0317	418.36	<.0001
Education	high	0.0924	0.0311	8.85	0.0029
Income	high	−0.0675	0.0312	4.70	0.0302

Predicted Values for Response Functions

			-----Observed-----		-----Predicted----		
Education	Income	Function Number	Function	Standard Error	Function	Standard Error	Residual
high	high	1	0.701299	0.052158	0.67294	0.047794	0.028359
high	low	1	0.773585	0.057487	0.808034	0.051586	−0.03445
low	high	1	0.454545	0.056744	0.48811	0.051077	−0.03356
low	low	1	0.703704	0.087877	0.623204	0.064867	0.080499

Output 28.11.2 Predicted Probabilities Data Set

Complex Sample Survey Analysis

Obs	Education	Income	Pred Function
1	high	low	0.808034
2	high	high	0.67294
3	low	low	0.623204
4	low	high	0.48811

You can use the predicted values (values of PredFunction in Output 28.11.2) as scores representing the likelihood that a randomly chosen subject from one of these populations will purchase the product. Notice that the "Response Profiles" table in Output 28.11.1 shows you that the first sorted level of Purchase is 'yes', indicating that the predicted probabilities are for Pr(Purchase='yes'). For example, someone with high education and low income has an estimated probability of purchase of 0.808. Like any response function estimate given by PROC CATMOD, this estimate can be obtained by cross-multiplying the row from the design matrix corresponding to the sample (sample number 2 in this case) with the vector of parameter estimates: $(1*0.6481)+(1*0.0924)+(-1*(-0.0675))$.

This ranking of scores can help in decision making (for example, with respect to allocation of advertising dollars, choice of advertising media, choice of print media, and so on).

References

Agresti, A. (1984), *Analysis of Ordinal Categorical Data*, New York: John Wiley & Sons.

Agresti, A. (1990), *Categorical Data Analysis*, New York: John Wiley & Sons.

Agresti, A. (1996), *An Introduction to Categorical Data Analysis*, New York: John Wiley & Sons.

Agresti, A. (2002), *Categorical Data Analysis*, Second Edition, New York: John Wiley & Sons.

Bishop, Y. M. M., Fienberg, S. E., and Holland, P. W. (1975), *Discrete Multivariate Analysis: Theory and Practice*, Cambridge, MA: MIT Press.

Christensen, R. (1997), *Log-Linear Models and Logistic Regression*, Second Edition, New York: Springer-Verlag.

Cox, D. R. and Snell, E. J. (1989), *The Analysis of Binary Data*, Second Edition, London: Chapman & Hall.

Fienberg, S. E. (1980), *The Analysis of Cross-Classified Categorical Data*, Second Edition, Cambridge, MA: The MIT Press.

Forthofer, R. N. and Koch, G. G. (1973), "An Analysis of Compounded Functions of Categorical Data," *Biometrics*, 29, 143–157.

Forthofer, R. N. and Lehnen, R. G. (1981), *Public Program Analysis: A New Categorical Data Approach*, Belmont, CA: Wadsworth.

Freeman, D. H., Jr. (1987), *Applied Categorical Data Analysis*, New York: Marcel Dekker.

Grizzle, J. E., Starmer, C. F., and Koch, G. G. (1969), "Analysis of Categorical Data by Linear Models," *Biometrics*, 25, 489–504.

Guthrie, D. (1981), "Analysis of Dichotomous Variables in Repeated Measures Experiments," *Psychological Bulletin*, 90, 189–195.

Haberman, S. J. (1972), "Log-Linear Fit for Contingency Tables," *Applied Statistics*, 21, 218–225.

Haslett, S. (1990), "Degrees of Freedom and Parameter Estimability in Hierarchical Models for Sparse Complete Contingency Tables," *Computational Statistics and Data Analysis*, 9, 179–195.

Imrey, P. B., Koch, G. G., and Stokes, M. E. (1981), "Categorical Data Analysis: Some Reflections on the Log Linear Model and Logistic Regression. Part I: Historical and Methodological Overview," *International Statistical Review*, 49, 265–283.

Koch, G. G., Landis, J. R., Freeman, J. L., Freeman, D. H., and Lehnen, R. G. (1977), "A General Methodology for the Analysis of Experiments with Repeated Measurement of Categorical Data," *Biometrics*, 33, 133–158.

Koch, G. G. and Stokes, M. E. (1979), *Annotated Computer Applications of Weighted Least Squares Methods for Illustrative Analyses of Examples Involving Health Survey Data*, Technical report, prepared for the U.S. National Center for Health Studies.

Landis, J. R., Stanish, W. M., Freeman, J. L., and Koch, G. G. (1976), "A Computer Program for the Generalized Chi-Square Analysis of Categorical Data Using Weighted Least Squares (GENCAT)," *Computer Programs in Biomedicine*, 6, 196–231.

MacMillan, J., Becker, C., Koch, G. G., Stokes, M., and Vandiviere, H. M. (1981), "An Application of Weighted Least Squares Methods to the Analysis of Measurement Process Components of Variability in an Observational Study," *American Statistical Association Proceedings of Survey Research Methods*.

Ries, P. N. and Smith, H. (1963), "The Use of Chi-Square for Preference Testing in Multidimensional Problems," *Chemical Engineering Progress*, 59, 39–43.

Searle, S. R. (1971), "Topics in Variance Component Estimation," *Biometrics*, 26, 1–76.

Stanish, W. M. and Koch, G. G. (1984), "The Use of CATMOD for Repeated Measurement Analysis of Categorical Data," in *Proceedings of the Ninth Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Stokes, M. E., Davis, C. S., and Koch, G. G. (2000), *Categorical Data Analysis Using the SAS System*, Second Edition, Cary, NC: SAS Institute Inc.

Wald, A. (1943), "Tests of Statistical Hypotheses Concerning General Parameters When the Number of Observations Is Large," *Transactions of the American Mathematical Society*, 54, 426–482.

Chapter 29

The CLUSTER Procedure

Contents

Overview: CLUSTER Procedure **1230**
Getting Started: CLUSTER Procedure **1231**
Syntax: CLUSTER Procedure **1239**
PROC CLUSTER Statement 1239
BY Statement 1247
COPY Statement 1247
FREQ Statement 1248
ID Statement 1248
RMSSTD Statement 1249
VAR Statement 1249
Details: CLUSTER Procedure **1250**
Clustering Methods 1250
Miscellaneous Formulas 1258
Ultrametrics 1259
Algorithms 1259
Computational Resources 1260
Missing Values 1261
Ties 1261
Size, Shape, and Correlation 1262
Output Data Set 1263
Displayed Output 1265
ODS Table Names 1268
ODS Graphics 1269
Examples: CLUSTER Procedure **1270**
Example 29.1: Cluster Analysis of Flying Mileages between 10 American Cities 1270
Example 29.2: Crude Birth and Death Rates 1277
Example 29.3: Cluster Analysis of Fisher's Iris Data 1289
Example 29.4: Evaluating the Effects of Ties 1304
References **1316**

Overview: CLUSTER Procedure

The CLUSTER procedure hierarchically clusters the observations in a SAS data set by using one of 11 methods. The data can be coordinates or distances. If the data are coordinates, PROC CLUSTER computes (possibly squared) Euclidean distances. If you want non-Euclidean distances, use the DISTANCE procedure (see Chapter 32) to compute an appropriate distance data set that can then be used as input to PROC CLUSTER.

The clustering methods are: average linkage, the centroid method, complete linkage, density linkage (including Wong's hybrid and kth-nearest-neighbor methods), maximum likelihood for mixtures of spherical multivariate normal distributions with equal variances but possibly unequal mixing proportions, the flexible-beta method, McQuitty's similarity analysis, the median method, single linkage, two-stage density linkage, and Ward's minimum-variance method. Each method is described in the section "Clustering Methods" on page 1250.

All methods are based on the usual agglomerative hierarchical clustering procedure. Each observation begins in a cluster by itself. The two closest clusters are merged to form a new cluster that replaces the two old clusters. Merging of the two closest clusters is repeated until only one cluster is left. The various clustering methods differ in how the distance between two clusters is computed.

The CLUSTER procedure is not practical for very large data sets because the CPU time is roughly proportional to the square or cube of the number of observations. The FASTCLUS procedure (see Chapter 34) requires time proportional to the number of observations and thus can be used with much larger data sets than PROC CLUSTER. If you want to cluster a very large data set hierarchically, use PROC FASTCLUS for a preliminary cluster analysis to produce a large number of clusters. Then use PROC CLUSTER to cluster the preliminary clusters hierarchically. This method is illustrated in Example 29.3.

PROC CLUSTER displays a history of the clustering process, showing statistics useful for estimating the number of clusters in the population from which the data are sampled. PROC CLUSTER also creates an output data set that can be used by the TREE procedure to draw a tree diagram of the cluster hierarchy or to output the cluster membership at any desired level. For example, to obtain the six-cluster solution, you could first use PROC CLUSTER with the OUTTREE= option, and then use this output data set as the input data set to the TREE procedure. With PROC TREE, specify NCLUSTERS=6 and the OUT= options to obtain the six-cluster solution and draw a tree diagram. For an example, see Example 91.1 in Chapter 91, "The TREE Procedure."

For coordinate data, Euclidean distances are computed from differences between coordinate values. The use of differences has several important consequences:

- For differences to be valid, the variables must have an interval or stronger scale of measurement. Ordinal or ranked data are generally not appropriate for cluster analysis.
- For Euclidean distances to be comparable, equal differences should have equal practical importance. You might need to transform the variables linearly or nonlinearly to satisfy this condition. For example, if one variable is measured in dollars and one in euros, you might need to convert to the same currency. Or, if ratios are more meaningful than differences, take logarithms.

- Variables with large variances tend to have more effect on the resulting clusters than variables with small variances. If you consider all variables to be equally important, you can use the STD option in PROC CLUSTER to standardize the variables to mean 0 and standard deviation 1. However, standardization is not always appropriate. See Milligan and Cooper (1987) for a Monte Carlo study on various methods of variable standardization. You should remove outliers before using PROC CLUSTER with the STD option unless you specify the TRIM= option. The STDIZE procedure (see Chapter 81) provides additional methods for standardizing variables and imputing missing values.

The ACECLUS procedure (see Chapter 22) is useful for linear transformations of the variables if any of the following conditions hold:

- You have no idea how the variables should be scaled.
- You want to detect natural clusters regardless of whether some variables have more influence than others.
- You want to use a clustering method designed for finding compact clusters, but you want to be able to detect elongated clusters.

Agglomerative hierarchical clustering is discussed in all standard references on cluster analysis, such as Anderberg (1973), Sneath and Sokal (1973), Hartigan (1975), Everitt (1980), and Spath (1980). An especially good introduction is given by Massart and Kaufman (1983). Anyone considering doing a hierarchical cluster analysis should study the Monte Carlo results of Milligan (1980), Milligan and Cooper (1985), and Cooper and Milligan (1988). Other essential, though more advanced, references on hierarchical clustering include Hartigan (1977, pp. 60–68; 1981), Wong (1982), Wong and Schaack (1982), and Wong and Lane (1983). See Blashfield and Aldenderfer (1978) for a discussion of the confusing terminology in hierarchical cluster analysis.

Getting Started: CLUSTER Procedure

The following example shows how you can use the CLUSTER procedure to compute hierarchical clusters of observations in a SAS data set.

Suppose you want to determine whether national figures for birth rates, death rates, and infant death rates can be used to categorize countries. Previous studies indicate that the clusters computed from this type of data can be elongated and elliptical. Thus, you need to perform a linear transformation on the raw data before the cluster analysis.

The following data[1] from Rouncefield (1995) are birth rates, death rates, and infant death rates for 97 countries. The DATA step creates the SAS data set Poverty:

[1] These data have been compiled from the *United Nations Demographic Yearbook 1990* (United Nations publications, Sales No. E/F.91.XII.1, copyright 1991, United Nations, New York) and are reproduced with the permission of the United Nations.

```
data Poverty;
   input Birth Death InfantDeath Country $20. @@;
   datalines;
24.7  5.7  30.8 Albania             12.5 11.9  14.4 Bulgaria
13.4 11.7  11.3 Czechoslovakia      12   12.4   7.6 Former_E._Germany
11.6 13.4  14.8 Hungary             14.3 10.2    16 Poland
13.6 10.7  26.9 Romania               14    9  20.2 Yugoslavia
17.7   10    23 USSR                15.2  9.5  13.1 Byelorussia_SSR
13.4 11.6    13 Ukrainian_SSR      20.7  8.4  25.7 Argentina
46.6   18   111 Bolivia           28.6  7.9    63 Brazil
23.4  5.8  17.1 Chile              27.4  6.1    40 Columbia
32.9  7.4    63 Ecuador           28.3  7.3    56 Guyana
34.8  6.6    42 Paraguay          32.9  8.3 109.9 Peru
  18  9.6  21.9 Uruguay           27.5  4.4  23.3 Venezuela
  29 23.2    43 Mexico              12 10.6   7.9 Belgium
13.2 10.1   5.8 Finland           12.4 11.9   7.5 Denmark
13.6  9.4   7.4 France            11.4 11.2   7.4 Germany
10.1  9.2    11 Greece            15.1  9.1   7.5 Ireland
 9.7  9.1   8.8 Italy             13.2  8.6   7.1 Netherlands
14.3 10.7   7.8 Norway            11.9  9.5  13.1 Portugal
10.7  8.2   8.1 Spain             14.5 11.1   5.6 Sweden
12.5  9.5   7.1 Switzerland       13.6 11.5   8.4 U.K.
14.9  7.4     8 Austria            9.9  6.7   4.5 Japan
14.5  7.3   7.2 Canada            16.7  8.1   9.1 U.S.A.
40.4 18.7 181.6 Afghanistan       28.4  3.8    16 Bahrain
42.5 11.5 108.1 Iran              42.6  7.8    69 Iraq
22.3  6.3   9.7 Israel            38.9  6.4    44 Jordan
26.8  2.2  15.6 Kuwait            31.7  8.7    48 Lebanon
45.6  7.8    40 Oman              42.1  7.6    71 Saudi_Arabia
29.2  8.4    76 Turkey            22.8  3.8    26 United_Arab_Emirates
42.2 15.5   119 Bangladesh        41.4 16.6   130 Cambodia
21.2  6.7    32 China             11.7  4.9   6.1 Hong_Kong
30.5 10.2    91 India             28.6  9.4    75 Indonesia
23.5 18.1    25 Korea             31.6  5.6    24 Malaysia
36.1  8.8    68 Mongolia          39.6 14.8   128 Nepal
30.3  8.1 107.7 Pakistan          33.2  7.7    45 Philippines
17.8  5.2   7.5 Singapore         21.3  6.2  19.4 Sri_Lanka
22.3  7.7    28 Thailand          31.8  9.5    64 Vietnam
35.5  8.3    74 Algeria           47.2 20.2   137 Angola
48.5 11.6    67 Botswana          46.1 14.6    73 Congo
38.8  9.5  49.4 Egypt             48.6 20.7   137 Ethiopia
39.4 16.8   103 Gabon             47.4 21.4   143 Gambia
44.4 13.1    90 Ghana               47 11.3    72 Kenya
  44  9.4    82 Libya             48.3   25   130 Malawi
35.5  9.8    82 Morocco             45 18.5   141 Mozambique
  44 12.1   135 Namibia           48.5 15.6   105 Nigeria
48.2 23.4   154 Sierra_Leone      50.1 20.2   132 Somalia
32.1  9.9    72 South_Africa      44.6 15.8   108 Sudan
46.8 12.5   118 Swaziland         31.1  7.3    52 Tunisia
52.2 15.6   103 Uganda            50.5   14   106 Tanzania
45.6 14.2    83 Zaire             51.1 13.7    80 Zambia
41.7 10.3    66 Zimbabwe
;
```

The data set Poverty contains the character variable Country and the numeric variables Birth, Death, and InfantDeath, which represent the birth rate per thousand, death rate per thousand, and infant death rate per thousand. The `$20.` in the INPUT statement specifies that the variable Country is a character variable with a length of 20. The double trailing at sign (@@) in the INPUT statement holds the input line for further iterations of the DATA step, specifying that observations are input from each line until all values are read.

Because the variables in the data set do not have equal variance, you must perform some form of scaling or transformation. One method is to standardize the variables to mean zero and variance one. However, when you suspect that the data contain elliptical clusters, you can use the ACECLUS procedure to transform the data such that the resulting within-cluster covariance matrix is spherical. The procedure obtains approximate estimates of the pooled within-cluster covariance matrix and then computes canonical variables to be used in subsequent analyses.

The following statements perform the ACECLUS transformation by using the SAS data set Poverty. The OUT= option creates an output SAS data set called Ace to contain the canonical variable scores:

```
proc aceclus data=Poverty out=Ace p=.03 noprint;
   var Birth Death InfantDeath;
run;
```

The P= option specifies that approximately 3% of the pairs are included in the estimation of the within-cluster covariance matrix. The NOPRINT option suppresses the display of the output. The VAR statement specifies that the variables Birth, Death, and InfantDeath are used in computing the canonical variables.

The following statements invoke the CLUSTER procedure, using the SAS data set ACE created in the previous PROC ACECLUS run:

```
ods graphics on;
proc cluster data=Ace method=ward ccc pseudo print=15 outtree=Tree;
   var can1 can2 can3 ;
   id country;
   format country $12.;
run;
ods graphics off;
```

The **ods graphics on** statement asks procedures to produce ODS graphics where possible. Ward's minimum-variance clustering method is specified by the METHOD= option. The CCC option displays the cubic clustering criterion, and the PSEUDO option displays pseudo F and t^2 statistics. The PRINT=15 option displays only the last 15 generations of the cluster history. The OUTTREE= option creates an output SAS data set called Tree that can be used by the TREE procedure to draw a tree diagram.

The VAR statement specifies that the canonical variables computed in the ACECLUS procedure are used in the cluster analysis. The ID statement specifies that the variable Country should be added to the Tree output data set.

The results of this analysis are displayed in the following figures.

PROC CLUSTER first displays the table of eigenvalues of the covariance matrix (Figure 29.1). These eigenvalues are used in the computation of the cubic clustering criterion. The first two columns list each eigenvalue and the difference between the eigenvalue and its successor. The last two columns display the individual and cumulative proportion of variation associated with each eigenvalue.

Figure 29.1 Table of Eigenvalues of the Covariance Matrix

The CLUSTER Procedure
Ward's Minimum Variance Cluster Analysis

Eigenvalues of the Covariance Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	64.5500051	54.7313223	0.8091	0.8091
2	9.8186828	4.4038309	0.1231	0.9321
3	5.4148519		0.0679	1.0000

Root-Mean-Square Total-Sample Standard Deviation 5.156987

Root-Mean-Square Distance Between Observations 12.63199

Figure 29.2 displays the last 15 generations of the cluster history. First listed are the number of clusters and the names of the clusters joined. The observations are identified either by the ID value or by CL*n*, where *n* is the number of the cluster. Next, PROC CLUSTER displays the number of observations in the new cluster and the semipartial R square. The latter value represents the decrease in the proportion of variance accounted for by joining the two clusters.

Figure 29.2 Cluster History

Cluster History

NCL	------Clusters Joined------		FREQ	SPRSQ	RSQ	ERSQ	CCC	PSF	PST2	Tie
15	Oman	CL37	5	0.0039	.957	.933	6.03	132	12.1	
14	CL31	CL22	13	0.0040	.953	.928	5.81	131	9.7	
13	CL41	CL17	32	0.0041	.949	.922	5.70	131	13.1	
12	CL19	CL21	10	0.0045	.945	.916	5.65	132	6.4	
11	CL39	CL15	9	0.0052	.940	.909	5.60	134	6.3	
10	CL76	CL27	6	0.0075	.932	.900	5.25	133	18.1	
9	CL23	CL11	15	0.0130	.919	.890	4.20	125	12.4	
8	CL10	Afghanistan	7	0.0134	.906	.879	3.55	122	7.3	
7	CL9	CL25	17	0.0217	.884	.864	2.26	114	11.6	
6	CL8	CL20	14	0.0239	.860	.846	1.42	112	10.5	
5	CL14	CL13	45	0.0307	.829	.822	0.65	112	59.2	
4	CL16	CL7	28	0.0323	.797	.788	0.57	122	14.8	
3	CL12	CL6	24	0.0323	.765	.732	1.84	153	11.6	
2	CL3	CL4	52	0.1782	.587	.613	-.82	135	48.9	
1	CL5	CL2	97	0.5866	.000	.000	0.00	.	135	

Next listed is the squared multiple correlation, R square, which is the proportion of variance accounted for by the clusters. Figure 29.2 shows that, when the data are grouped into three clusters, the proportion of variance accounted for by the clusters (R square) is just under 77%. The approximate expected value of R square is given in the ERSQ column. This expectation is approximated under the null hypothesis that the data have a uniform distribution instead of forming distinct clusters.

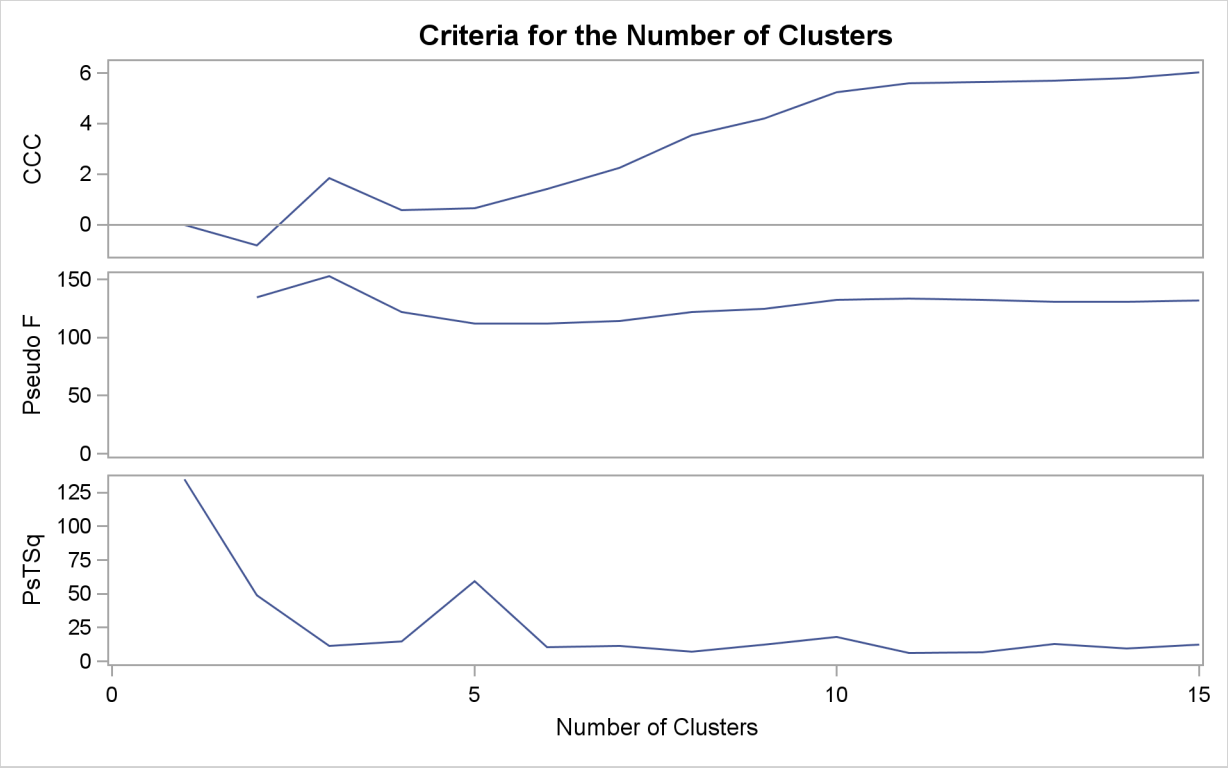
The next three columns display the values of the cubic clustering criterion (CCC), pseudo F (PSF), and t^2 (PST2) statistics. These statistics are useful for estimating the number of clusters in the data.

The final column in Figure 29.2 lists ties for minimum distance; a blank value indicates the absence of a tie. A tie means that the clusters are indeterminate and that changing the order of the observations may change the clusters. See Example 29.4 for ways to investigate the effects of ties.

Figure 29.3 plots the three statistics for estimating the number of clusters. Peaks in the plot of the cubic clustering criterion with values greater than 2 or 3 indicate good clusters; peaks with values between 0 and 2 indicate possible clusters. Large negative values of the CCC can indicate outliers. In Figure 29.3, there is a local peak of the CCC when the number of clusters is 3. The CCC drops at 4 clusters and then steadily increases, leveling off at 11 clusters.

Another method of judging the number of clusters in a data set is to look at the pseudo F statistic (PSF). Relatively large values indicate good numbers of clusters. In Figure 29.3, the pseudo F statistic suggests 3 clusters or 11 clusters.

Figure 29.3 Plot of Statistics for Estimating the Number of Clusters



To interpret the values of the pseudo t^2 statistic, look down the column or look at the plot from right to left until you find the first value markedly larger than the previous value, then move back up the column or to the right in the plot by one step in the cluster history. In Figure 29.3, you can see possibly good clustering levels at 11 clusters, 6 clusters, 3 clusters, and 2 clusters.

Considered together, these statistics suggest that the data can be clustered into 11 clusters or 3 clusters. The following statements examine the results of clustering the data into 3 clusters.

A graphical view of the clustering process can often be helpful in interpreting the clusters. The following statements use the TREE procedure to produce a tree diagram of the clusters:

```
goptions vsize=9in hsize=6.4in htext=.9pct htitle=3pct;
axis1 order=(0 to 1 by 0.2);
proc tree data=Tree out=New nclusters=3
          haxis=axis1 horizontal;
   height _rsq_;
   copy can1 can2 ;
   id country;
run;
```

The AXIS1 statement defines axis parameters that are used in the TREE procedure. The ORDER= option specifies the data values in the order in which they should appear on the axis.

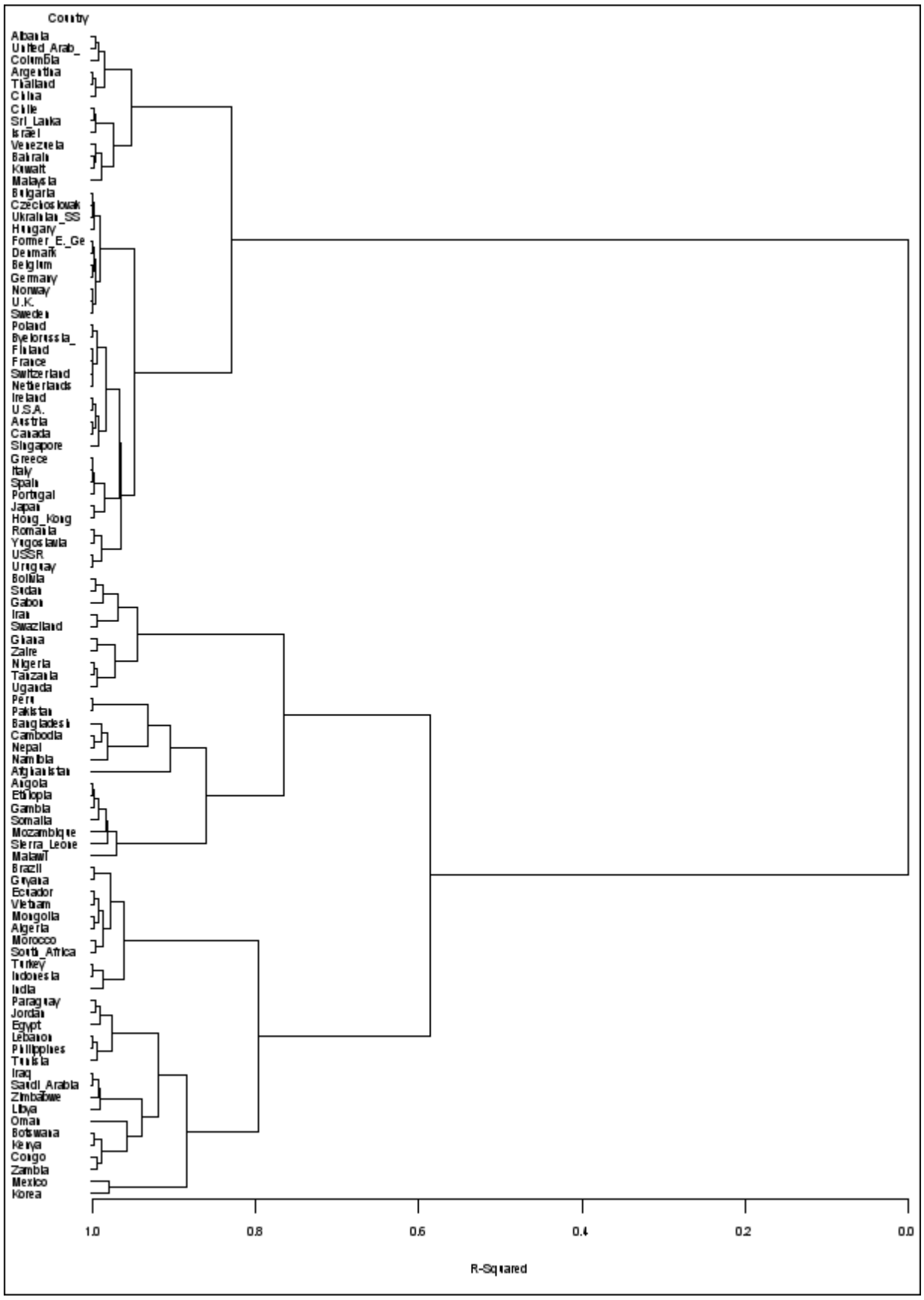
The preceding statements use the SAS data set Tree as input. The OUT= option creates an output SAS data set named New to contain information about cluster membership. The NCLUSTERS= option specifies the number of clusters desired in the data set New.

The TREE procedure produces high-resolution graphics by default. The HAXIS= option specifies AXIS1 to customize the appearance of the horizontal axis. The HORIZONTAL option orients the tree diagram horizontally. The HEIGHT statement specifies the variable _RSQ_ (R square) as the height variable.

The COPY statement copies the canonical variables can1 and can2 (computed in the ACECLUS procedure) into the output SAS data set New. Thus, the SAS output data set New contains information for three clusters and the first two of the original canonical variables.

Figure 29.4 displays the tree diagram. The figure provides a graphical view of the information in Figure 29.2. As the number of branches grows to the left from the root, the R square approaches 1; the first three clusters (branches of the tree) account for over half of the variation (about 77%, from Figure 29.4). In other words, only three clusters are necessary to explain over three-fourths of the variation.

Figure 29.4 Tree Diagram of Clusters versus R-Square Values

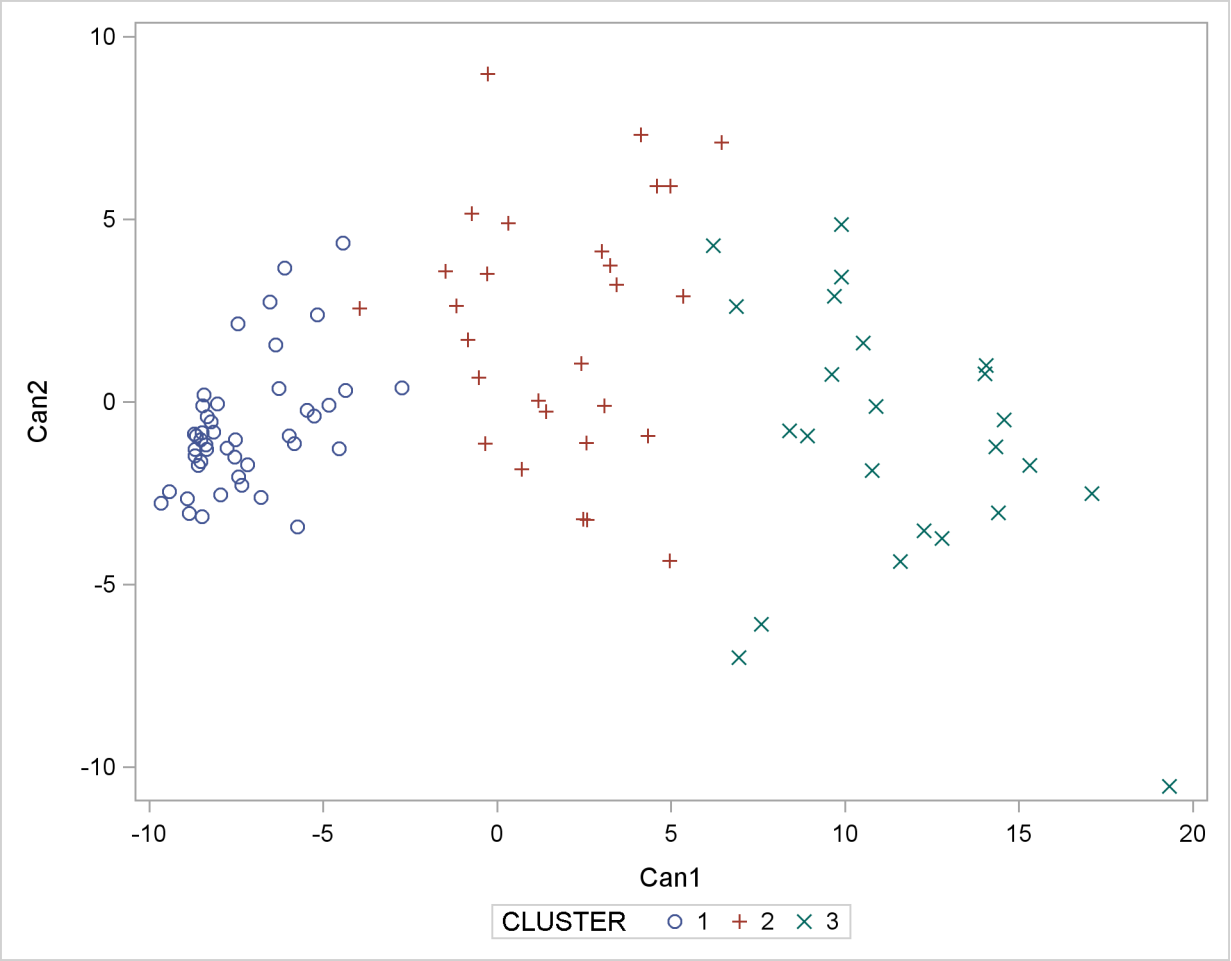


The following statements invoke the SGPLOT procedure on the SAS data set New:

```
proc sgplot data=New ;
   scatter y=can2 x=can1 / group=cluster ;
run;
```

The PLOT statement requests a plot of the two canonical variables, using the value of the variable cluster as the identification variable, as shown in Figure 29.5.

Figure 29.5 Plot of Canonical Variables and Cluster for Three Clusters



The statistics in Figure 29.2 and Figure 29.3, the tree diagram in Figure 29.4, and the plot of the canonical variables in Figure 29.5 assist in the estimation of clusters in the data. There seems to be reasonable separation in the clusters. However, you must use this information, along with experience and knowledge of the field, to help in deciding the correct number of clusters.

Syntax: CLUSTER Procedure

The following statements are available in the CLUSTER procedure:

PROC CLUSTER *METHOD = name* < *options* > **;**
BY *variables* **;**
COPY *variables* **;**
FREQ *variable* **;**
ID *variable* **;**
RMSSTD *variable* **;**
VAR *variables* **;**

Only the PROC CLUSTER statement is required, except that the FREQ statement is required when the RMSSTD statement is used; otherwise the FREQ statement is optional. Usually only the VAR statement and possibly the ID and COPY statements are needed in addition to the PROC CLUSTER statement. The rest of this section provides detailed syntax information for each of the preceding statements, beginning with the PROC CLUSTER statement. The remaining statements are covered in alphabetical order.

PROC CLUSTER Statement

PROC CLUSTER *METHOD=name* < *options* > **;**

The PROC CLUSTER statement starts the CLUSTER procedure, specifies a clustering method, and optionally specifies details for clustering methods, data sets, data processing, and displayed output.

The METHOD= specification determines the clustering method used by the procedure. Any one of the following 11 methods can be specified for *name*:

AVERAGE | AVE — requests average linkage (group average, unweighted pair-group method using arithmetic averages, UPGMA). Distance data are squared unless you specify the NOSQUARE option.

CENTROID | CEN — requests the centroid method (unweighted pair-group method using centroids, UPGMC, centroid sorting, weighted-group method). Distance data are squared unless you specify the NOSQUARE option.

COMPLETE | COM — requests complete linkage (furthest neighbor, maximum method, diameter method, rank order typal analysis). To reduce distortion of clusters by outliers, the TRIM= option is recommended.

DENSITY | DEN — requests density linkage, which is a class of clustering methods using nonparametric probability density estimation. You must also specify either the K=, R=, or HYBRID option to indicate the type of density estimation to be used. See also the MODE= and DIM= options in this section.

EML — requests maximum-likelihood hierarchical clustering for mixtures of spherical multivariate normal distributions with equal variances but possibly unequal mixing proportions. Use METHOD=EML only with coordinate data. See the PENALTY= option for details. The NONORM option does not affect the reported likelihood values but does affect other unrelated criteria. The EML method is much slower than the other methods in the CLUSTER procedure.

FLEXIBLE | FLE — requests the Lance-Williams flexible-beta method. See the BETA= option in this section.

MCQUITTY | MCQ — requests McQuitty's similarity analysis (weighted average linkage, weighted pair-group method using arithmetic averages, WPGMA).

MEDIAN | MED — requests Gower's median method (weighted pair-group method using centroids, WPGMC). Distance data are squared unless you specify the NOSQUARE option.

SINGLE | SIN — requests single linkage (nearest neighbor, minimum method, connectedness method, elementary linkage analysis, or dendritic method). To reduce chaining, you can use the TRIM= option with METHOD=SINGLE.

TWOSTAGE | TWO — requests two-stage density linkage. You must also specify the K=, R=, or HYBRID option to indicate the type of density estimation to be used. See also the MODE= and DIM= options in this section.

WARD | WAR — requests Ward's minimum-variance method (error sum of squares, trace W). Distance data are squared unless you specify the NOSQUARE option. To reduce distortion by outliers, the TRIM= option is recommended. See the NONORM option.

Table 29.1 summarizes the options in the PROC CLUSTER statement.

Table 29.1 PROC CLUSTER Statement Options

Option	Description
Specify input and output data sets	
DATA=	specifies input data set
OUTTREE=	creates output data set
Specify clustering methods	
METHOD=	specifies clustering method
BETA=	specifies beta value for flexible beta method
MODE=	specifies the minimum number of members for modal clusters
PENALTY=	specifies the penalty coefficient for maximum likelihood
HYBRID	specifies Wong's hybrid clustering method
Control data processing prior to clustering	
NOEIGEN	suppresses computation of eigenvalues
NONORM	suppresses normalizing of distances
NOSQUARE	suppresses squaring of distances

Table 29.1 *continued*

Option	Description
STANDARD	standardizes variables
TRIM=	omits points with low probability densities
Control density estimation	
K=	specifies number of neighbors for kth-nearest-neighbor density estimation
R=	specifies radius of sphere of support for uniform-kernel density estimation
Ties	
NOTIE	suppresses checking for ties
Control display of the cluster history	
CCC	displays cubic clustering criterion
NOID	suppresses display of ID values
PRINT=	specifies number of generations to display
PSEUDO	displays pseudo F and t^2 statistics
RMSSTD	displays root mean square standard deviation
RSQUARE	displays R square and semipartial R square
Control other aspects of output	
NOPRINT	suppresses display of all output
SIMPLE	displays simple summary statistics
PLOTS=	specifies ODS graphics details

The following list provides details on these options.

BETA=*n*

specifies the beta parameter for METHOD=FLEXIBLE. The value of n should be less than 1, usually between 0 and -1. By default, BETA=-0.25. Milligan (1987) suggests a somewhat smaller value, perhaps -0.5, for data with many outliers.

CCC

displays the cubic clustering criterion and approximate expected R square under the uniform null hypothesis (Sarle 1983). The statistics associated with the RSQUARE option, R square and semipartial R square, are also displayed. The CCC option applies only to coordinate data. The CCC option is not appropriate with METHOD=SINGLE because of the method's tendency to chop off tails of distributions. Computation of the CCC requires the eigenvalues of the covariance matrix. If the number of variables is large, computing the eigenvalues requires much computer time and memory.

DATA=*SAS-data-set*

names the input data set containing observations to be clustered. By default, the procedure uses the most recently created SAS data set. If the data set is TYPE=DISTANCE, the data are interpreted as a distance matrix; the number of variables must equal the number of observations in the data set or in each BY group. The distances are assumed to be Euclidean, but the procedure accepts other types of distances or dissimilarities. If the data set is not

TYPE=DISTANCE, the data are interpreted as coordinates in a Euclidean space, and Euclidean distances are computed. For more about TYPE=DISTANCE data sets, see Chapter A, "Special SAS Data Sets."

You cannot use a TYPE=CORR data set as input to PROC CLUSTER, since the procedure uses dissimilarity measures. Instead, you can use a DATA step or the IML procedure to extract the correlation matrix from a TYPE=CORR data set and transform the values to dissimilarities such as $1 - r$ or $1 - r^2$, where r is the correlation.

All methods produce the same results when used with coordinate data as when used with Euclidean distances computed from the coordinates. However, the DIM= option must be used with distance data if you specify METHOD=TWOSTAGE or METHOD=DENSITY or if you specify the TRIM= option.

Certain methods that are most naturally defined in terms of coordinates require *squared* Euclidean distances to be used in the combinatorial distance formulas (Lance and Williams 1967). For this reason, distance data are automatically squared when used with METHOD=AVERAGE, METHOD=CENTROID, METHOD=MEDIAN, or METHOD=WARD. If you want the combinatorial formulas to be applied to the (unsquared) distances with these methods, use the NOSQUARE option.

DIM=*n*

specifies the dimensionality used when computing density estimates with the TRIM= option, METHOD=DENSITY, or METHOD=TWOSTAGE. The values of n must be greater than or equal to 1. The default is the number of variables if the data are coordinates; the default is 1 if the data are distances.

HYBRID

requests Wong's (1982) hybrid clustering method in which density estimates are computed from a preliminary cluster analysis using the k-means method. The DATA= data set must contain means, frequencies, and root mean square standard deviations of the preliminary clusters (see the FREQ and RMSSTD statements). To use HYBRID, you must use either a FREQ statement or a DATA= data set that contains a _FREQ_ variable, and you must also use either an RMSSTD statement or a DATA= data set that contains an _RMSSTD_ variable.

The MEAN= data set produced by the FASTCLUS procedure is suitable for input to the CLUSTER procedure for hybrid clustering. Since this data set contains _FREQ_ and _RMSSTD_ variables, you can use it as input and then omit the FREQ and RMSSTD statements.

You must specify either METHOD=DENSITY or METHOD=TWOSTAGE with the HYBRID option. You cannot use this option in combination with the TRIM=, K=, or R= option.

K=*n*

specifies the number of neighbors to use for kth-nearest-neighbor density estimation (Silverman 1986, pp. 19–21 and 96–99). The number of neighbors (n) must be at least two but less than the number of observations. See the MODE= option, which follows.

Density estimation is used with the TRIM=, METHOD=DENSITY, and METHOD=TWOSTAGE options.

MODE=*n*

specifies that, when two clusters are joined, each must have at least n members in order for either cluster to be designated a modal cluster. If you specify MODE=1, each cluster must also have a maximum density greater than the fusion density in order for either cluster to be designated a modal cluster.

Use the MODE= option only with METHOD=DENSITY or METHOD=TWOSTAGE. With METHOD=TWOSTAGE, the MODE= option affects the number of modal clusters formed. With METHOD=DENSITY, the MODE= option does not affect the clustering process but does determine the number of modal clusters reported on the output and identified by the _MODE_ variable in the output data set.

If you specify the K= option, the default value of MODE= is the same as the value of K= because the use of kth-nearest-neighbor density estimation limits the resolution that can be obtained for clusters with fewer than k members. If you do not specify the K= option, the default is MODE=2.

If you specify MODE=0, the default value is used instead of 0.

If you specify a FREQ statement or if a _FREQ_ variable appears in the input data set, the MODE= value is compared with the number of actual observations in the clusters being joined, not with the sum of the frequencies in the clusters.

NOEIGEN

suppresses computation of the eigenvalues of the covariance matrix and substitutes the variances of the variables for the eigenvalues when computing the cubic clustering criterion. The NOEIGEN option saves time if the number of variables is large, but it should be used only if the variables are nearly uncorrelated. If you specify the NOEIGEN option and the variables are highly correlated, the cubic clustering criterion might be very liberal. The NOEIGEN option applies only to coordinate data.

NOID

suppresses the display of ID values for the clusters joined at each generation of the cluster history.

NONORM

prevents the distances from being normalized to unit mean or unit root mean square with most methods. With METHOD=WARD, the NONORM option prevents the between-cluster sum of squares from being normalized by the total sum of squares to yield a squared semipartial correlation. The NONORM option does not affect the reported likelihood values with METHOD=EML, but it does affect other unrelated criteria, such as the _DIST_ variable.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 20, "Using the Output Delivery System."

NOSQUARE

prevents input distances from being squared with METHOD=AVERAGE, METHOD=CENTROID, METHOD=MEDIAN, or METHOD=WARD.

If you specify the NOSQUARE option with distance data, the data are assumed to be squared Euclidean distances for computing R-square and related statistics defined in a Euclidean coordinate system.

If you specify the NOSQUARE option with coordinate data with METHOD=CENTROID, METHOD=MEDIAN, or METHOD=WARD, then the combinatorial formula is applied to unsquared Euclidean distances. The resulting cluster distances do not have their usual Euclidean interpretation and are therefore labeled "False" in the output.

NOTIE

prevents PROC CLUSTER from checking for ties for minimum distance between clusters at each generation of the cluster history. If your data are measured with such precision that ties are unlikely, then you can specify the NOTIE option to reduce slightly the time and space required by the procedure. See the section "Ties" on page 1261 for more information.

OUTTREE=*SAS-data-set*

creates an output data set that can be used by the TREE procedure to draw a tree diagram. You must give the data set a two-level name to save it. See *SAS Language Reference: Concepts* for a discussion of permanent data sets. If you omit the OUTTREE= option, the data set is named by using the DATA*n* convention and is not permanently saved. If you do not want to create an output data set, use OUTTREE=_NULL_.

PENALTY=*p*

specifies the penalty coefficient used with METHOD=EML. See the section "Clustering Methods" on page 1250 for more information. Values for p must be greater than zero. By default, PENALTY=2.

PLOTS <(*global-plot-options***)> <= *plot-request* >**

PLOTS <(*global-plot-options***)> <= (*plot-request* <... *plot-request* >)>**

controls the plots produced through ODS Graphics.

PROC CLUSTER can produce line plots of the cubic clustering criterion, the pseudo F statistic, and the pseudo t^2 statistic from the cluster history table. These statistics are useful for estimating the number of clusters. Each statistic is plotted against the number of clusters.

To obtain ODS Graphics plots from PROC CLUSTER, you must do two things. First, enable ODS Graphics before running PROC CLUSTER. For example:

```
ods graphics on;

proc cluster plots=all;
run;

ods graphics off;
```

Second, request that PROC CLUSTER compute the desired statistics by specifying the CCC or PSEUDO options, or by specifying the statistics in a *plot-request* in the PLOT option.

PROC CLUSTER might be unable to compute the statistics in some cases; for details, see the CCC and PSEUDO options. If a statistic cannot be computed, it cannot be plotted. PROC CLUSTER plots all of these statistics that are computed unless you tell it specifically what to plot using PLOTS=.

The maximum number of clusters shown in all the plots is the minimum of the following quantities:

- the number of observations
- the value of the PRINT= option, if that option is specified
- the maximum number of clusters for which CCC is computed, if CCC is plotted

The *global-plot-options* apply to all plots generated by the CLUSTER procedure. The global plot options are as follows:

UNPACKPANELS breaks a plot that is otherwise paneled into plots separate plots for each statistic. This option can be abbreviated as UNPACK.

ONLY has no effect, but is accepted for consistency with other procedures.

The following *plot-requests* can be specified:

ALL implicitly specifies the CCC and PSEUDO options and, if possible, produces all three plots.

NONE suppresses all plots.

CCC implicitly specifies the CCC option and, if possible, plots the cubic clustering criterion against the number of clusters.

PSEUDO implicitly specifies the PSEUDO option and, if possible, plots the pseudo F statistic and the pseudo t^2 statistic against the number of clusters.

PSF implicitly specifies the PSEUDO option and, if possible, plots the pseudo F statistic against the number of clusters.

PST2 implicitly specifies the PSEUDO option and, if possible, plots the pseudo t^2 statistic against the number of clusters.

When you specify only one *plot-request*, you can omit the parentheses around the *plot-request*. You can specify one or more of the CCC, PSEUDO, PSF, or PST2 plot requests in the same PLOT option. For example, all of the following are valid:

```
PROC CLUSTER PLOTS=(CCC PST2);
PROC CLUSTER PLOTS=(PSF);
PROC CLUSTER PLOTS=PSF;
```

The first statement plots both the cubic clustering criterion and the pseudo t^2 statistic, while the second and third statements plot the pseudo F statistic only.

The names of the graphs that PROC CLUSTER generates are listed in Table 29.5, along with the required statements and options.

PRINT=*n* | **P=***n*

specifies the number of generations of the cluster history to display. The P= option displays the latest n generations; for example, P=5 displays the cluster history from 1 cluster through 5 clusters. The value of P= must be a nonnegative integer. The default is to display all generations. Specify PRINT=0 to suppress the cluster history.

PSEUDO

displays pseudo F and t^2 statistics. This option is effective only when the data are coordinates or when METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD is specified. See the section "Miscellaneous Formulas" on page 1258 for more information. The PSEUDO option is not appropriate with METHOD=SINGLE because of the method's tendency to chop off tails of distributions.

R=*n*

specifies the radius of the sphere of support for uniform-kernel density estimation (Silverman 1986, pp. 11–13 and 75–94). The value of R= must be greater than zero.

Density estimation is used with the TRIM=, METHOD=DENSITY, and METHOD=TWOSTAGE options.

RMSSTD

displays the root mean square standard deviation of each cluster. This option is effective only when the data are coordinates or when METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD is specified.

See the section "Miscellaneous Formulas" on page 1258 for more information.

RSQUARE | RSQ

displays the R square and semipartial R square. This option is effective only when the data are coordinates or when METHOD=AVERAGE or METHOD=CENTROID is specified. The R square and semipartial R square statistics are always displayed with METHOD=WARD. See the section "Miscellaneous Formulas" on page 1258 for more information..

SIMPLE | S

displays means, standard deviations, skewness, kurtosis, and a coefficient of bimodality. The SIMPLE option applies only to coordinate data. See the section "Miscellaneous Formulas" on page 1258 for more information.

STANDARD | STD

standardizes the variables to mean 0 and standard deviation 1. The STANDARD option applies only to coordinate data.

TRIM=*p*

omits points with low estimated probability densities from the analysis. Valid values for the TRIM= option are $0 \leq p < 100$. If $p < 1$, then p is the proportion of observations omitted. If $p \geq 1$, then p is interpreted as a percentage. A specification of TRIM=10, which trims 10% of the points, is a reasonable value for many data sets. Densities are estimated by the kth-nearest-neighbor or uniform-kernel method. Trimmed points are indicated by a negative value of the _FREQ_ variable in the OUTTREE= data set.

You must use either the K= or R= option when you use TRIM=. You cannot use the HYBRID option in combination with TRIM=, so you might want to use the DIM= option instead. If you specify the STANDARD option in combination with TRIM=, the variables are standardized both before and after trimming.

The TRIM= option is useful for removing outliers and reducing chaining. Trimming is highly recommended with METHOD=WARD or METHOD=COMPLETE because clusters from these methods can be severely distorted by outliers. Trimming is also valuable with METHOD=SINGLE since single linkage is the method most susceptible to chaining. Most other methods also benefit from trimming. However, trimming is unnecessary with METHOD=TWOSTAGE or METHOD=DENSITY when kth-nearest-neighbor density estimation is used.

Use of the TRIM= option can spuriously inflate the cubic clustering criterion and the pseudo F and t^2 statistics. Trimming only outliers improves the accuracy of the statistics, but trimming saddle regions between clusters yields excessively large values.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC CLUSTER to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the CLUSTER procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*.

For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

COPY Statement

COPY *variables* **;**

The variables in the COPY statement are copied from the input data set to the OUTTREE= data set. Observations in the OUTTREE= data set that represent clusters of more than one observation from the input data set have missing values for the COPY variables.

FREQ Statement

FREQ *variable* **;**

If one variable in the input data set represents the frequency of occurrence for other values in the observation, specify the variable's name in a FREQ statement. PROC CLUSTER then treats the data set as if each observation appeared n times, where n is the value of the FREQ variable for the observation. Noninteger values of the FREQ variable are truncated to the largest integer less than the FREQ value.

If you omit the FREQ statement but the DATA= data set contains a variable called _FREQ_, then frequencies are obtained from the _FREQ_ variable. If neither a FREQ statement nor an _FREQ_ variable is present, each observation is assumed to have a frequency of one.

If each observation in the DATA= data set represents a cluster (for example, clusters formed by PROC FASTCLUS), the variable specified in the FREQ statement should give the number of original observations in each cluster.

If you specify the RMSSTD statement, a FREQ statement is required. A FREQ statement or _FREQ_ variable is required when you specify the HYBRID option.

With most clustering methods, the same clusters are obtained from a data set with a FREQ variable as from a similar data set without a FREQ variable, if each observation is repeated as many times as the value of the FREQ variable in the first data set. The FLEXIBLE method can yield different results due to the nature of the combinatorial formula. The DENSITY and TWOSTAGE methods are also exceptions because two identical observations can be absorbed one at a time by a cluster with a higher density. If you are using a FREQ statement with either the DENSITY or TWOSTAGE method, see the MODE=option for details.

ID Statement

ID *variable* **;**

The values of the ID variable identify observations in the displayed cluster history and in the OUTTREE= data set. If the ID statement is omitted, each observation is denoted by OBn, where n is the observation number.

RMSSTD Statement

RMSSTD *variable* **;**

If the coordinates in the DATA= data set represent cluster means (for example, formed by the FASTCLUS procedure), you can obtain accurate statistics in the cluster histories for METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD if the data set contains both of the following:

- a variable giving the number of original observations in each cluster (see the discussion of the FREQ statement earlier in this chapter)
- a variable giving the root mean squared standard deviation of each cluster

Specify the name of the variable containing root mean squared standard deviations in the RMSSTD statement. If you specify the RMSSTD statement, you must also specify a FREQ statement.

If you omit the RMSSTD statement but the DATA= data set contains a variable called _RMSSTD_, then the root mean squared standard deviations are obtained from the _RMSSTD_ variable.

An RMSSTD statement or _RMSSTD_ variable is required when you specify the HYBRID option.

A data set created by PROC FASTCLUS, using the MEAN= option, contains _FREQ_ and _RMSSTD_ variables, so you do not have to use FREQ and RMSSTD statements when using such a data set as input to the CLUSTER procedure.

VAR Statement

VAR *variables* **;**

The VAR statement lists numeric variables to be used in the cluster analysis. If you omit the VAR statement, all numeric variables not listed in other statements are used.

Details: CLUSTER Procedure

Clustering Methods

The following notation is used, with lowercase symbols generally pertaining to observations and uppercase symbols pertaining to clusters:

n	number of observations
v	number of variables if data are coordinates
G	number of clusters at any given level of the hierarchy
x_i or $\mathbf{x}_i$	ith observation (row vector if coordinate data)
C_K	Kth cluster, subset of $\{1, 2, \ldots, n\}$
N_K	number of observations in C_K
$\bar{\mathbf{x}}$	sample mean vector
$\bar{\mathbf{x}}_K$	mean vector for cluster C_K
$\|\mathbf{x}\|$	Euclidean length of the vector $\mathbf{x}$—that is, the square root of the sum of the squares of the elements of $\mathbf{x}$
T	$\sum_{i=1}^{n} \|\mathbf{x}_i - \bar{\mathbf{x}}\|^2$
W_K	$\sum_{i \in C_k} \|\mathbf{x}_i - \bar{\mathbf{x}}_K\|^2$
P_G	$\sum W_J$, where summation is over the G clusters at the Gth level of the hierarchy
B_{KL}	$W_M - W_K - W_L$ if $C_M = C_K \cup C_L$
$d(\mathbf{x}, \mathbf{y})$	any distance or dissimilarity measure between observations or vectors $\mathbf{x}$ and $\mathbf{y}$
D_{KL}	any distance or dissimilarity measure between clusters C_K and C_L

The distance between two clusters can be defined either directly or combinatorially (Lance and Williams 1967)—that is, by an equation for updating a distance matrix when two clusters are joined. In all of the following combinatorial formulas, it is assumed that clusters C_K and C_L are merged to form C_M, and the formula gives the distance between the new cluster C_M and any other cluster C_J.

For an introduction to most of the methods used in the CLUSTER procedure, see Massart and Kaufman (1983).

Average Linkage

The following method is obtained by specifying METHOD=AVERAGE. The distance between two clusters is defined by

$$D_{KL} = \frac{1}{N_K N_L} \sum_{i \in C_K} \sum_{j \in C_L} d(x_i, x_j)$$

If $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$, then

$$D_{KL} = \|\bar{\mathbf{x}}_K - \bar{\mathbf{x}}_L\|^2 + \frac{W_K}{N_K} + \frac{W_L}{N_L}$$

The combinatorial formula is

$$D_{JM} = \frac{N_K D_{JK} + N_L D_{JL}}{N_M}$$

In average linkage the distance between two clusters is the average distance between pairs of observations, one in each cluster. Average linkage tends to join clusters with small variances, and it is slightly biased toward producing clusters with the same variance.

Average linkage was originated by Sokal and Michener (1958).

Centroid Method

The following method is obtained by specifying METHOD=CENTROID. The distance between two clusters is defined by

$$D_{KL} = \|\bar{\mathbf{x}}_K - \bar{\mathbf{x}}_L\|^2$$

If $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$, then the combinatorial formula is

$$D_{JM} = \frac{N_K D_{JK} + N_L D_{JL}}{N_M} - \frac{N_K N_L D_{KL}}{N_M^2}$$

In the centroid method, the distance between two clusters is defined as the (squared) Euclidean distance between their centroids or means. The centroid method is more robust to outliers than most other hierarchical methods but in other respects might not perform as well as Ward's method or average linkage (Milligan 1980).

The centroid method was originated by Sokal and Michener (1958).

Complete Linkage

The following method is obtained by specifying METHOD=COMPLETE. The distance between two clusters is defined by

$$D_{KL} = \max_{i \in C_K} \max_{j \in C_L} d(x_i, x_j)$$

The combinatorial formula is

$$D_{JM} = \max(D_{JK}, D_{JL})$$

In complete linkage, the distance between two clusters is the maximum distance between an observation in one cluster and an observation in the other cluster. Complete linkage is strongly biased toward producing clusters with roughly equal diameters, and it can be severely distorted by moderate outliers (Milligan 1980).

Complete linkage was originated by Sorensen (1948).

Density Linkage

The phrase *density linkage* is used here to refer to a class of clustering methods that use nonparametric probability density estimates (for example, Hartigan 1975, pp. 205–212; Wong 1982; Wong and Lane 1983). Density linkage consists of two steps:

1. A new dissimilarity measure, d^*, based on density estimates and adjacencies is computed. If x_i and x_j are adjacent (the definition of *adjacency* depends on the method of density estimation), then $d^*(x_i, x_j)$ is the reciprocal of an estimate of the density midway between x_i and x_j; otherwise, $d^*(x_i, x_j)$ is infinite.
2. A single linkage cluster analysis is performed using d^*.

The CLUSTER procedure supports three types of density linkage: the kth-nearest-neighbor method, the uniform-kernel method, and Wong's hybrid method. These are obtained by using METHOD=DENSITY and the K=, R=, and HYBRID options, respectively.

kth-Nearest-Neighbor Method

The kth-nearest-neighbor method (Wong and Lane 1983) uses kth-nearest-neighbor density estimates. Let $r_k(x)$ be the distance from point x to the kth-nearest observation, where k is the value specified for the K= option. Consider a closed sphere centered at x with radius $r_k(x)$. The estimated density at x, $f(x)$, is the proportion of observations within the sphere divided by the volume of the sphere. The new dissimilarity measure is computed as

$$d^*(x_i, x_j) = \begin{cases} \frac{1}{2}\left(\frac{1}{f(x_i)} + \frac{1}{f(x_j)}\right) & \text{if } d(x_i, x_j) \le \max(r_k(x_i), r_k(x_j)) \\ \infty & \text{otherwise} \end{cases}$$

Wong and Lane (1983) show that kth-nearest-neighbor density linkage is strongly set consistent for high-density (density-contour) clusters if k is chosen such that $k/n \to 0$ and $k/\ln(n) \to \infty$ as $n \to \infty$. Wong and Schaack (1982) discuss methods for estimating the number of population clusters by using kth-nearest-neighbor clustering.

Uniform-Kernel Method

The uniform-kernel method uses uniform-kernel density estimates. Let r be the value specified for the R= option. Consider a closed sphere centered at point x with radius r. The estimated density at x, $f(x)$, is the proportion of observations within the sphere divided by the volume of the sphere. The new dissimilarity measure is computed as

$$d^*(x_i, x_j) = \begin{cases} \frac{1}{2}\left(\frac{1}{f(x_i)} + \frac{1}{f(x_j)}\right) & \text{if } d(x_i, x_j) \le r \\ \infty & \text{otherwise} \end{cases}$$

Wong's Hybrid Method

Wong's (1982) hybrid clustering method uses density estimates based on a preliminary cluster analysis by the k-means method. The preliminary clustering can be done by the FASTCLUS procedure,

by using the MEAN= option to create a data set containing cluster means, frequencies, and root mean squared standard deviations. This data set is used as input to the CLUSTER procedure, and the HYBRID option is specified with METHOD=DENSITY to request the hybrid analysis. The hybrid method is appropriate for very large data sets but should not be used with small data sets—say, than those with fewer than 100 observations in the original data. The term *preliminary cluster* refers to an observation in the DATA= data set.

For preliminary cluster C_K, N_K and W_K are obtained from the input data set, as are the cluster means or the distances between the cluster means. Preliminary clusters C_K and C_L are considered adjacent if the midpoint between $\bar{x}_K$ and $\bar{\mathbf{x}}_L$ is closer to either $\bar{\mathbf{x}}_K$ or $\bar{\mathbf{x}}_L$ than to any other preliminary cluster mean or, equivalently, if $d^2(\bar{\mathbf{x}}_K, \bar{\mathbf{x}}_L) < d^2(\bar{\mathbf{x}}_K, \bar{\mathbf{x}}_M) + d^2(\bar{\mathbf{x}}_L, \bar{\mathbf{x}}_M)$ for all other preliminary clusters C_M, $M \neq K$ or L. The new dissimilarity measure is computed as

$$d^*(\bar{\mathbf{x}}_K, \bar{\mathbf{x}}_L) = \begin{cases} \dfrac{\left(W_K + W_L + \frac{1}{4}(N_K + N_L) d^2(\bar{\mathbf{x}}_K, \bar{\mathbf{x}}_L)\right)^{\frac{v}{2}}}{(N_K + N_L)^{1+\frac{v}{2}}} & \text{if } C_K \text{ and } C_L \text{ are adjacent} \\ \infty & \text{otherwise} \end{cases}$$

Using the K= and R= Options

The values of the K= and R= options are called *smoothing parameters*. Small values of K= or R= produce jagged density estimates and, as a consequence, many modes. Large values of K= or R= produce smoother density estimates and fewer modes. In the hybrid method, the smoothing parameter is the number of clusters in the preliminary cluster analysis. The number of modes in the final analysis tends to increase as the number of clusters in the preliminary analysis increases. Wong (1982) suggests using $n^{0.3}$ preliminary clusters, where n is the number of observations in the original data set. There is no rule of thumb for selecting K= values. For all types of density linkage, you should repeat the analysis with several different values of the smoothing parameter (Wong and Schaack 1982).

There is no simple answer to the question of which smoothing parameter to use (Silverman 1986, pp. 43–61, 84–88, and 98–99). It is usually necessary to try several different smoothing parameters. A reasonable first guess for the R= option in many coordinate data sets is given by

$$\left[\frac{2^{v+2}(v+2)\Gamma(\frac{v}{2}+1)}{nv^2}\right]^{\frac{1}{v+4}} \sqrt{\sum_{l=1}^{v} s_l^2}$$

where s_l^2 is the standard deviation of the lth variable. The estimate for R= can be computed in a DATA step by using the GAMMA function for Γ. This formula is derived under the assumption that the data are sampled from a multivariate normal distribution and tends, therefore, to be too large (oversmooth) if the true distribution is multimodal. Robust estimates of the standard deviations can be preferable if there are outliers. If the data are distances, the factor $\sum s_l^2$ can be replaced by an average (mean, trimmed mean, median, root mean square, and so on) distance divided by $\sqrt{2}$. To prevent outliers from appearing as separate clusters, you can also specify K=2, or more generally K=m, $m \geq 2$, which in most cases forces clusters to have at least m members.

If the variables all have unit variance (for example, if the STANDARD option is used), Table 29.2 can be used to obtain an initial guess for the R= option.

Since infinite d^* values occur in density linkage, the final number of clusters can exceed one when there are wide gaps between the clusters or when the smoothing parameter results in little smoothing.

Density linkage applies no constraints to the shapes of the clusters and, unlike most other hierarchical clustering methods, is capable of recovering clusters with elongated or irregular shapes. Since density linkage uses less prior knowledge about the shape of the clusters than do methods restricted to compact clusters, density linkage is less effective at recovering compact clusters from small samples than are methods that always recover compact clusters, regardless of the data.

Table 29.2 Reasonable First Guess for the R= Option for Standardized Data

Number of Observations	Number of Variables									
	1	2	3	4	5	6	7	8	9	10
20	1.01	1.36	1.77	2.23	2.73	3.25	3.81	4.38	4.98	5.60
35	0.91	1.24	1.64	2.08	2.56	3.08	3.62	4.18	4.77	5.38
50	0.84	1.17	1.56	1.99	2.46	2.97	3.50	4.06	4.64	5.24
75	0.78	1.09	1.47	1.89	2.35	2.85	3.38	3.93	4.50	5.09
100	0.73	1.04	1.41	1.82	2.28	2.77	3.29	3.83	4.40	4.99
150	0.68	0.97	1.33	1.73	2.18	2.66	3.17	3.71	4.27	4.85
200	0.64	0.93	1.28	1.67	2.11	2.58	3.09	3.62	4.17	4.75
350	0.57	0.85	1.18	1.56	1.98	2.44	2.93	3.45	4.00	4.56
500	0.53	0.80	1.12	1.49	1.91	2.36	2.84	3.35	3.89	4.45
750	0.49	0.74	1.06	1.42	1.82	2.26	2.74	3.24	3.77	4.32
1000	0.46	0.71	1.01	1.37	1.77	2.20	2.67	3.16	3.69	4.23
1500	0.43	0.66	0.96	1.30	1.69	2.11	2.57	3.06	3.57	4.11
2000	0.40	0.63	0.92	1.25	1.63	2.05	2.50	2.99	3.49	4.03

EML

The following method is obtained by specifying METHOD=EML. The distance between two clusters is given by

$$D_{KL} = nv \ln\left(1 + \frac{B_{KL}}{P_G}\right) - 2\left(N_M \ln(N_M) - N_K \ln(N_K) - N_L \ln(N_L)\right)$$

The EML method joins clusters to maximize the likelihood at each level of the hierarchy under the following assumptions:

- multivariate normal mixture
- equal spherical covariance matrices
- unequal sampling probabilities

The EML method is similar to Ward's minimum-variance method but removes the bias toward equal-sized clusters. Practical experience has indicated that EML is somewhat biased toward unequal-sized clusters. You can specify the PENALTY= option to adjust the degree of bias. If you specify PENALTY=p, the formula is modified to

$$D_{KL} = nv \ln\left(1 + \frac{B_{KL}}{P_G}\right) - p\left(N_M \ln(N_M) - N_K \ln(N_K) - N_L \ln(N_L)\right)$$

The EML method was derived by W. S. Sarle of SAS Institute from the maximum likelihood formula obtained by Symons (1981, p. 37, Equation 8) for disjoint clustering. There are currently no other published references on the EML method.

Flexible-Beta Method

The following method is obtained by specifying METHOD=FLEXIBLE. The combinatorial formula is

$$D_{JM} = (D_{JK} + D_{JL})\frac{1-b}{2} + D_{KL}b$$

where b is the value of the BETA= option, or -0.25 by default.

The flexible-beta method was developed by Lance and Williams (1967); see also Milligan (1987).

McQuitty's Similarity Analysis

The following method is obtained by specifying METHOD=MCQUITTY. The combinatorial formula is

$$D_{JM} = \frac{D_{JK} + D_{JL}}{2}$$

The method was independently developed by Sokal and Michener (1958) and McQuitty (1966).

Median Method

The following method is obtained by specifying METHOD=MEDIAN. If $d(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|^2$, then the combinatorial formula is

$$D_{JM} = \frac{D_{JK} + D_{JL}}{2} - \frac{D_{KL}}{4}$$

The median method was developed by Gower (1967).

Single Linkage

The following method is obtained by specifying METHOD=SINGLE. The distance between two clusters is defined by

$$D_{KL} = \min_{i \in C_K} \min_{j \in C_L} d(x_i, x_j)$$

The combinatorial formula is

$$D_{JM} = \min(D_{JK}, D_{JL})$$

In single linkage, the distance between two clusters is the minimum distance between an observation in one cluster and an observation in the other cluster. Single linkage has many desirable theoretical properties (Jardine and Sibson 1971; Fisher and Van Ness 1971; Hartigan 1981) but has fared poorly in Monte Carlo studies (for example, Milligan 1980). By imposing no constraints on the shape of clusters, single linkage sacrifices performance in the recovery of compact clusters in return for the ability to detect elongated and irregular clusters. You must also recognize that single linkage tends to chop off the tails of distributions before separating the main clusters (Hartigan 1981). The notorious chaining tendency of single linkage can be alleviated by specifying the TRIM= option (Wishart 1969, pp. 296–298).

Density linkage and two-stage density linkage retain most of the virtues of single linkage while performing better with compact clusters and possessing better asymptotic properties (Wong and Lane 1983).

Single linkage was originated by Florek et al. (1951a, 1951b) and later reinvented by McQuitty (1957) and Sneath (1957).

Two-Stage Density Linkage

If you specify METHOD=DENSITY, the modal clusters often merge before all the points in the tails have clustered. The option METHOD=TWOSTAGE is a modification of density linkage that ensures that all points are assigned to modal clusters before the modal clusters are permitted to join. The CLUSTER procedure supports the same three varieties of two-stage density linkage as of ordinary density linkage: kth-nearest neighbor, uniform kernel, and hybrid.

In the first stage, disjoint modal clusters are formed. The algorithm is the same as the single linkage algorithm ordinarily used with density linkage, with one exception: two clusters are joined only if at least one of the two clusters has fewer members than the number specified by the MODE= option. At the end of the first stage, each point belongs to one modal cluster.

In the second stage, the modal clusters are hierarchically joined by single linkage. The final number of clusters can exceed one when there are wide gaps between the clusters or when the smoothing parameter is small.

Each stage forms a tree that can be plotted by the TREE procedure. By default, the TREE procedure plots the tree from the first stage. To obtain the tree for the second stage, use the option HEIGHT=MODE in the PROC TREE statement. You can also produce a single tree diagram containing both stages, with the number of clusters as the height axis, by using the option HEIGHT=N in the PROC TREE statement. To produce an output data set from PROC TREE containing the modal clusters, use _HEIGHT_ for the HEIGHT variable (the default) and specify LEVEL=0.

Two-stage density linkage was developed by W. S. Sarle of SAS Institute. There are currently no other published references on two-stage density linkage.

Ward's Minimum-Variance Method

The following method is obtained by specifying METHOD=WARD. The distance between two clusters is defined by

$$D_{KL} = B_{KL} = \frac{\|\bar{\mathbf{x}}_K - \bar{\mathbf{x}}_L\|^2}{\frac{1}{N_K} + \frac{1}{N_L}}$$

If $d(\mathbf{x}, \mathbf{y}) = \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|^2$, then the combinatorial formula is

$$D_{JM} = \frac{(N_J + N_K)D_{JK} + (N_J + N_L)D_{JL} - N_J D_{KL}}{N_J + N_M}$$

In Ward's minimum-variance method, the distance between two clusters is the ANOVA sum of squares between the two clusters added up over all the variables. At each generation, the within-cluster sum of squares is minimized over all partitions obtainable by merging two clusters from the previous generation. The sums of squares are easier to interpret when they are divided by the total sum of squares to give proportions of variance (squared semipartial correlations).

Ward's method joins clusters to maximize the likelihood at each level of the hierarchy under the following assumptions:

- multivariate normal mixture
- equal spherical covariance matrices
- equal sampling probabilities

Ward's method tends to join clusters with a small number of observations, and it is strongly biased toward producing clusters with roughly the same number of observations. It is also very sensitive to outliers (Milligan 1980).

Ward (1963) describes a class of hierarchical clustering methods including the minimum variance method.

Miscellaneous Formulas

The root mean squared standard deviation of a cluster C_K is

$$\text{RMSSTD} = \sqrt{\frac{W_K}{v(N_K - 1)}}$$

The R-square statistic for a given level of the hierarchy is

$$R^2 = 1 - \frac{P_G}{T}$$

The squared semipartial correlation for joining clusters C_K and C_L is

$$\text{semipartial } R^2 = \frac{B_{KL}}{T}$$

The bimodality coefficient is

$$b = \frac{m_3^2 + 1}{m_4 + \frac{3(n-1)^2}{(n-2)(n-3)}}$$

where m_3 is skewness and m_4 is kurtosis. Values of b greater than 0.555 (the value for a uniform population) can indicate bimodal or multimodal marginal distributions. The maximum of 1.0 (obtained for the Bernoulli distribution) is obtained for a population with only two distinct values. Very heavy-tailed distributions have small values of b regardless of the number of modes.

Formulas for the cubic-clustering criterion and approximate expected R square are given in Sarle (1983).

The pseudo F statistic for a given level is

$$\text{pseudo } F = \frac{\frac{T - P_G}{G-1}}{\frac{P_G}{n-G}}$$

The pseudo t^2 statistic for joining C_K and C_L is

$$\text{pseudo } t^2 = \frac{B_{KL}}{\frac{W_K + W_L}{N_K + N_L - 2}}$$

The pseudo F and t^2 statistics can be useful indicators of the number of clusters, but they are *not* distributed as F and t^2 random variables. If the data are independently sampled from a multivariate normal distribution with a scalar covariance matrix and if the clustering method allocates observations to clusters randomly (which no clustering method actually does), then the pseudo F statistic is distributed as an F random variable with $v(G-1)$ and $v(n-G)$ degrees of freedom. Under the same assumptions, the pseudo t^2 statistic is distributed as an F random variable with v and $v(N_K + N_L - 2)$ degrees of freedom. The pseudo t^2 statistic differs computationally from Hotelling's T^2 in that the latter uses a general symmetric covariance matrix instead of a scalar

covariance matrix. The pseudo F statistic was suggested by Calinski and Harabasz (1974). The pseudo t^2 statistic is related to the $J_e(2)/J_e(1)$ statistic of Duda and Hart (1973) by

$$\frac{J_e(2)}{J_e(1)} = \frac{W_K + W_L}{W_M} = \frac{1}{1 + \frac{t^2}{N_K + N_L - 2}}$$

See Milligan and Cooper (1985) and Cooper and Milligan (1988) regarding the performance of these statistics in estimating the number of population clusters. Conservative tests for the number of clusters using the pseudo F and t^2 statistics can be obtained by the Bonferroni approach (Hawkins, Muller, and ten Krooden 1982, pp. 337–340).

Ultrametrics

A dissimilarity measure $d(x, y)$ is called an *ultrametric* if it satisfies the following conditions:

- $d(x, x) = 0$ for all x
- $d(x, y) \geq 0$ for all x, y
- $d(x, y) = d(y, x)$ for all x, y
- $d(x, y) \leq \max(d(x, z), d(y, z))$ for all x, y, and z

Any hierarchical clustering method induces a dissimilarity measure on the observations—say, $h(x_i, x_j)$. Let C_M be the cluster with the fewest members that contains both x_i and x_j. Assume C_M was formed by joining C_K and C_L. Then define $h(x_i, x_j) = D_{KL}$.

If the fusion of C_K and C_L reduces the number of clusters from g to $g-1$, then define $D_{(g)} = D_{KL}$. Johnson (1967) shows that if

$$0 \leq D_{(n)} \leq D_{(n-1)} \leq \cdots \leq D_{(2)}$$

then $h(\cdot, \cdot)$ is an ultrametric. A method that always satisfies this condition is said to be a *monotonic* or *ultrametric clustering method*. All methods implemented in PROC CLUSTER except CENTROID, EML, and MEDIAN are ultrametric (Milligan 1979; Batagelj 1981).

Algorithms

Anderberg (1973) describes three algorithms for implementing agglomerative hierarchical clustering: stored data, stored distance, and sorted distance. The algorithms used by PROC CLUSTER for each method are indicated in Table 29.3. For METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD, either the stored data or the stored distance algorithm can be used. For these methods, if the data are distances or if you specify the NOSQUARE option, the stored distance algorithm is used; otherwise, the stored data algorithm is used.

Table 29.3 Three Algorithms for Implementing Agglomerative Hierarchical Clustering

	Algorithm		
Stored Method	**Stored Data**	**Stored Distance**	**Sorted Distance**
AVERAGE	x	x	
CENTROID	x	x	
COMPLETE		x	
DENSITY			x
EML	x		
FLEXIBLE		x	
MCQUITTY		x	
MEDIAN		x	
SINGLE		x	
TWOSTAGE			x
WARD	x	x	

Computational Resources

The CLUSTER procedure stores the data (including the COPY and ID variables) in memory or, if necessary, on disk. If eigenvalues are computed, the covariance matrix is stored in memory. If the stored distance or sorted distance algorithm is used, the distances are stored in memory or, if necessary, on disk.

With coordinate data, the increase in CPU time is roughly proportional to the number of variables. The VAR statement should list the variables in order of decreasing variance for greatest efficiency.

For both coordinate and distance data, the dominant factor determining CPU time is the number of observations. For density methods with coordinate data, the asymptotic time requirements are somewhere between $n \ln(n)$ and n^2, depending on how the smoothing parameter increases. For other methods except EML, time is roughly proportional to n^2. For the EML method, time is roughly proportional to n^3.

PROC CLUSTER runs much faster if the data can be stored in memory and, when the stored distance algorithm is used, if the distance matrix can be stored in memory as well. To estimate the bytes of memory needed for the data, use the following formula and round up to the nearest multiple of d.

$$
\begin{aligned}
n(vd &+ 8d + i \\
&+ i && \text{if density estimation or the sorted distance algorithm is used} \\
&+ 3d && \text{if stored data algorithm is used} \\
&+ 3d && \text{if density estimation is used} \\
&+ \max(8, \text{length of ID variable}) && \text{if ID variable is used} \\
&+ \text{length of ID variable} && \text{if ID variable is used} \\
&+ \text{sum of lengths of COPY variables}) && \text{if COPY variables is used}
\end{aligned}
$$

where

- n is the number of observations
- v is the number of variables
- d is the size of a C variable of type *double*. For most computers, $d = 8$.
- i is the size of a C variable of type *int*. For most computers, $i = 4$.

The number of bytes needed for the distance matrix is $dn(n+1)/2$.

Missing Values

If the data are coordinates, observations with missing values are excluded from the analysis. If the data are distances, missing values are not permitted in the lower triangle of the distance matrix. The upper triangle is ignored. For more about TYPE=DISTANCE data sets, see Chapter A, "Special SAS Data Sets."

Ties

At each level of the clustering algorithm, PROC CLUSTER must identify the pair of clusters with the minimum distance. Sometimes, usually when the data are discrete, there can be two or more pairs with the same minimum distance. In such cases the tie must be broken in some arbitrary way. If there are ties, then the results of the cluster analysis depend on the order of the observations in the data set. The presence of ties is reported in the SAS log and in the column of the cluster history labeled "Tie" unless the NOTIE option is specified.

PROC CLUSTER breaks ties as follows. Each cluster is identified by the smallest observation number among its members. For each pair of clusters, there is a smaller identification number and a larger identification number. If two or more pairs of clusters are tied for minimum distance between clusters, the pair that has the minimum larger identification number is merged. If there is a tie for minimum larger identification number, the pair that has the minimum smaller identification number is merged.

A tie means that the level in the cluster history at which the tie occurred and possibly some of the subsequent levels are not uniquely determined. Ties that occur early in the cluster history usually have little effect on the later stages. Ties that occur in the middle part of the cluster history are cause for further investigation. Ties that occur late in the cluster history indicate important indeterminacies.

The importance of ties can be assessed by repeating the cluster analysis for several different random permutations of the observations. The discrepancies at a given level can be examined by crosstabulating the clusters obtained at that level for all of the permutations. See Example 29.4 for details.

Size, Shape, and Correlation

In some biological applications, the organisms that are being clustered can be at different stages of growth. Unless it is the growth process itself that is being studied, differences in size among such organisms are not of interest. Therefore, distances among organisms should be computed in such a way as to control for differences in size while retaining information about differences in shape.

If coordinate data are measured on an interval scale, you can control for size by subtracting a measure of the overall size of each observation from each data item. For example, if no other direct measure of size is available, you could subtract the mean of each row of the data matrix, producing a row-centered coordinate matrix. An easy way to subtract the mean of each row is to use PROC STANDARD on the transposed coordinate matrix:

```
proc transpose data= coordinate-datatype ;
proc standard m=0;
proc transpose out=row-centered-coordinate-data;
```

Another way to remove size effects from interval-scale coordinate data is to do a principal component analysis and discard the first component (Blackith and Reyment 1971).

If the data are measured on a ratio scale, you can control for size by dividing each observation by a measure of overall size; in this case, the geometric mean is a more natural measure of size than the arithmetic mean. However, it is often more meaningful to analyze the logarithms of ratio-scaled data, in which case you can subtract the arithmetic mean after taking logarithms. You must also consider the dimensions of measurement. For example, if you have measures of both length and weight, you might need to cube the measures of length or take the cube root of the weights. Various other complications can also arise in real applications, such as different growth rates for different parts of the body (Sneath and Sokal 1973).

Issues of size and shape are pertinent to many areas besides biology (for example, Hamer and Cunningham 1981). Suppose you have data consisting of subjective ratings made by several different raters. Some raters tend to give higher overall ratings than other raters. Some raters also tend to spread out their ratings over more of the scale than other raters. If it is impossible for you to adjust directly for rater differences, then distances should be computed in such a way as to control for differences both in size and variability. For example, if the data are considered to be measured on an interval scale, you can subtract the mean of each observation and divide by the standard deviation, producing a row-standardized coordinate matrix. With some clustering methods, analyzing squared Euclidean distances from a row-standardized coordinate matrix is equivalent to analyzing

the matrix of correlations among rows, since squared Euclidean distance is an affine transformation of the correlation (Hartigan 1975, p. 64).

If you do an analysis of row-centered or row-standardized data, you need to consider whether the columns (variables) should be standardized before centering or standardizing the rows, after centering or standardizing the rows, or both before and after. If you standardize the columns after standardizing the rows, then strictly speaking you are not analyzing shape because the profiles are distorted by standardizing the columns; however, this type of double standardization might be necessary in practice to get reasonable results. It is not clear whether iterating the standardization of rows and columns can be of any benefit.

The choice of distance or correlation measure should depend on the meaning of the data and the purpose of the analysis. Simulation studies that compare distance and correlation measures are useless unless the data are generated to mimic data from your field of application. Conclusions drawn from artificial data cannot be generalized, because it is possible to generate data such that distances that include size effects work better or such that correlations work better.

You can standardize the rows of a data set by using a DATA step or by using the TRANSPOSE and STANDARD procedures. You can also use PROC TRANSPOSE and then have PROC CORR create a TYPE=CORR data set containing a correlation matrix. If you want to analyze a TYPE=CORR data set with PROC CLUSTER, you must use a DATA step to perform the following steps:

1. Set the data set TYPE= to DISTANCE.
2. Convert the correlations to dissimilarities by computing $1-r$, $\sqrt{1-r}$, $1-r^2$, or some other decreasing function.
3. Delete observations for which the variable _TYPE_ does not have the value 'CORR'.

Output Data Set

The OUTTREE= data set contains one observation for each observation in the input data set, plus one observation for each cluster of two or more observations (that is, one observation for each node of the cluster tree). The total number of output observations is usually $2n-1$, where n is the number of input observations. The density methods can produce fewer output observations when the number of clusters cannot be reduced to one.

The label of the OUTTREE= data set identifies the type of cluster analysis performed and is automatically displayed when the TREE procedure is invoked.

The variables in the OUTTREE= data set are as follows:

- the BY variables, if you use a BY statement
- the ID variable, if you use an ID statement
- the COPY variables, if you use a COPY statement

- _NAME_, a character variable giving the name of the node. If the node is a cluster, the name is CL*n*, where *n* is the number of the cluster. If the node is an observation, the name is OB*n*, where *n* is the observation number. If the node is an observation and the ID statement is used, the name is the formatted value of the ID variable.
- _PARENT_, a character variable giving the value of _NAME_ of the parent of the node
- _NCL_, the number of clusters
- _FREQ_, the number of observations in the current cluster
- _HEIGHT_, the distance or similarity between the last clusters joined, as defined in the section "Clustering Methods" on page 1250. The variable _HEIGHT_ is used by the TREE procedure as the default height axis. The label of the _HEIGHT_ variable identifies the between-cluster distance measure. For METHOD=TWOSTAGE, the _HEIGHT_ variable contains the densities at which clusters joined in the first stage; for clusters formed in the second stage, _HEIGHT_ is a very small negative number.

If the input data set contains coordinates, the following variables appear in the output data set:

- the variables containing the coordinates used in the cluster analysis. For output observations that correspond to input observations, the values of the coordinates are the same in both data sets except for some slight numeric error possibly introduced by standardizing and unstandardizing if the STANDARD option is used. For output observations that correspond to clusters of more than one input observation, the values of the coordinates are the cluster means.
- _ERSQ_, the approximate expected value of R square under the uniform null hypothesis
- _RATIO_, equal to $\frac{1-\text{\_ERSQ\_}}{1-\text{\_RSQ\_}}$
- _LOGR_, natural logarithm of _RATIO_
- _CCC_, the cubic clustering criterion

The variables _ERSQ_, _RATIO_, _LOGR_, and _CCC_ have missing values when the number of clusters is greater than one-fifth the number of observations.

If the input data set contains coordinates and METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD, then the following variables appear in the output data set:

- _DIST_, the Euclidean distance between the means of the last clusters joined
- _AVLINK_, the average distance between the last clusters joined

If the input data set contains coordinates or METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD, then the following variables appear in the output data set:

- _RMSSTD_, the root mean squared standard deviation of the current cluster

- _SPRSQ_, the semipartial squared multiple correlation or the decrease in the proportion of variance accounted for due to joining two clusters to form the current cluster
- _RSQ_, the squared multiple correlation
- _PSF_, the pseudo F statistic
- _PST2_, the pseudo t^2 statistic

If METHOD=EML, then the following variable appears in the output data set:

- _LNLR_, the log-likelihood ratio

If METHOD=TWOSTAGE or METHOD=DENSITY, the following variable appears in the output data set:

- _MODE_, pertaining to the modal clusters. With METHOD=DENSITY, the _MODE_ variable indicates the number of modal clusters contained by the current cluster. With METHOD=TWOSTAGE, the _MODE_ variable gives the maximum density in each modal cluster and the fusion density, d^*, for clusters containing two or more modal clusters; for clusters containing no modal clusters, _MODE_ is missing.

If nonparametric density estimates are requested (when METHOD=DENSITY or METHOD=TWOSTAGE and the HYBRID option is not used; or when the TRIM= option is used), the output data set contains the following:

- _DENS_, the maximum density in the current cluster

Displayed Output

If you specify the SIMPLE option and the data are coordinates, PROC CLUSTER produces simple descriptive statistics for each variable:

- the Mean
- the standard deviation, Std Dev
- the Skewness
- the Kurtosis
- a coefficient of Bimodality

If the data are coordinates and you do not specify the NOEIGEN option, PROC CLUSTER displays the following:

- the Eigenvalues of the Correlation or Covariance Matrix
- the Difference between successive eigenvalues
- the Proportion of variance explained by each eigenvalue
- the Cumulative proportion of variance explained

If the data are coordinates, PROC CLUSTER displays the Root Mean Squared Total-Sample Standard Deviation of the variables

If the distances are normalized, PROC CLUSTER displays one of the following, depending on whether squared or unsquared distances are used:

- the Root Mean Squared Distance Between Observations
- the Mean Distance Between Observations

For the generations in the clustering process specified by the PRINT= option, PROC CLUSTER displays the following:

- the Number of Clusters or NCL
- the names of the Clusters Joined. The observations are identified by the formatted value of the ID variable, if any; otherwise, the observations are identified by OB*n*, where *n* is the observation number. The CLUSTER procedure displays the entire value of the ID variable in the cluster history instead of truncating at 16 characters. Long ID values might be split onto several lines. Clusters of two or more observations are identified as CL*n*, where *n* is the number of clusters existing after the cluster in question is formed.
- the number of observations in the new cluster, Frequency of New Cluster or FREQ

If you specify the RMSSTD option and the data are coordinates, or if you specify METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD, then PROC CLUSTER displays the root mean squared standard deviation of the new cluster, RMS Std of New Cluster or RMS Std.

PROC CLUSTER displays the following items if you specify METHOD=WARD. It also displays them if you specify the RSQUARE option and either the data are coordinates or you specify METHOD=AVERAGE or METHOD=CENTROID.

- the decrease in the proportion of variance accounted for resulting from joining the two clusters, Semipartial R-Squared or SPRSQ. This equals the between-cluster sum of squares divided by the corrected total sum of squares.
- the squared multiple correlation, R-Squared or RSQ. R square is the proportion of variance accounted for by the clusters.

If you specify the CCC option and the data are coordinates, PROC CLUSTER displays the following:

- Approximate Expected R-Squared or ERSQ, the approximate expected value of R square under the uniform null hypothesis
- the Cubic Clustering Criterion or CCC. The cubic clustering criterion and approximate expected R square are given missing values when the number of clusters is greater than one-fifth the number of observations.

If you specify the PSEUDO option and the data are coordinates, or if you specify METHOD=AVERAGE, METHOD=CENTROID, or METHOD=WARD, then PROC CLUSTER displays the following:

- Pseudo F or PSF, the pseudo F statistic measuring the separation among all the clusters at the current level
- Pseudo t^2 or PST2, the pseudo t^2 statistic measuring the separation between the two clusters most recently joined

If you specify the NOSQUARE option and METHOD=AVERAGE, PROC CLUSTER displays the (Normalized) Average Distance or (Norm) Aver Dist, the average distance between pairs of objects in the two clusters joined with one object from each cluster.

If you do not specify the NOSQUARE option and METHOD=AVERAGE, PROC CLUSTER displays the (Normalized) RMS Distance or (Norm) RMS Dist, the root mean squared distance between pairs of objects in the two clusters joined with one object from each cluster.

If METHOD=CENTROID, PROC CLUSTER displays the (Normalized) Centroid Distance or (Norm) Cent Dist, the distance between the two cluster centroids.

If METHOD=COMPLETE, PROC CLUSTER displays the (Normalized) Maximum Distance or (Norm) Max Dist, the maximum distance between the two clusters.

If METHOD=DENSITY or METHOD=TWOSTAGE, PROC CLUSTER displays the following:

- Normalized Fusion Density or Normalized Fusion Dens, the value of d^* as defined in the section "Clustering Methods" on page 1250
- the Normalized Maximum Density in Each Cluster joined, including the Lesser or Min, and the Greater or Max, of the two maximum density values

If METHOD=EML, PROC CLUSTER displays the following:

- Log Likelihood Ratio or LNLR
- Log Likelihood or LNLIKE

If METHOD=FLEXIBLE, PROC CLUSTER displays the (Normalized) Flexible Distance or (Norm) Flex Dist, the distance between the two clusters based on the Lance-Williams flexible formula.

If METHOD=MEDIAN, PROC CLUSTER displays the (Normalized) Median Distance or (Norm) Med Dist, the distance between the two clusters based on the median method.

If METHOD=MCQUITTY, PROC CLUSTER displays the (Normalized) McQuitty's Similarity or (Norm) MCQ, the distance between the two clusters based on McQuitty's similarity method.

If METHOD=SINGLE, PROC CLUSTER displays the (Normalized) Minimum Distance or (Norm) Min Dist, the minimum distance between the two clusters.

If you specify the NONORM option and METHOD=WARD, PROC CLUSTER displays the Between-Cluster Sum of Squares or BSS, the ANOVA sum of squares between the two clusters joined.

If you specify neither the NOTIE option nor METHOD=TWOSTAGE or METHOD=DENSITY, PROC CLUSTER displays Tie, where a T in the column indicates a tie for minimum distance and a blank indicates the absence of a tie.

After the cluster history, if METHOD=TWOSTAGE or METHOD=DENSITY, PROC CLUSTER displays the number of modal clusters.

ODS Table Names

PROC CLUSTER assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 29.4. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 29.4 ODS Tables Produced by PROC CLUSTER

ODS Table Name	Description	Statement	Option
ClusterHistory	Observation or clusters joined, frequencies and other cluster statistics	PROC	default
SimpleStatistics	Simple statistics, before or after trimming	PROC	SIMPLE
EigenvalueTable	Eigenvalues of the CORR or COV matrix	PROC	default
rmsstd	Root mean square total sample standard deviation	PROC	default
avdist	Root mean square distance between observations	PROC	default

ODS Graphics

To produce graphics from PROC CLUSTER, you must enable ODS Graphics by specifying the **ods graphics on** statement before running PROC CLUSTER. See Chapter 21, "Statistical Graphics Using ODS," for more information.

PROC CLUSTER can produce line plots of the cubic clustering criterion, pseudo F, and pseudo t^2 statistics. To plot a statistic, you must ask for it to be computed via one or more of the CCC, PSEUDO, or PLOT options.

You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC CLUSTER generates are listed in Table 29.5, along with the required statements and options.

Table 29.5 ODS Graphics Produced by PROC CLUSTER

ODS Graph Name	Plot Description	Statement & Option
CubicClusCritPlot	Cubic clustering criterion for the number of clusters	PROC CLUSTER PLOTS=CCC
PseudoFPlot	Pseudo F criterion for the number of clusters	PROC CLUSTER PLOTS=PSF
PseudoTSqPlot	Pseudo t^2 criterion for the number of clusters	PROC CLUSTER PLOTS=PST2
CccAndPsTSqPlot	Cubic clustering criterion and pseudo t^2	PROC CLUSTER PLOTS=(CCC PST2)
CccAndPsfPlot	Cubic clustering criterion and pseudo F	PROC CLUSTER PLOTS=(CCC PSF)
CccPsfAndPsTSqPlot	Cubic clustering criterion, pseudo F, and pseudo t^2	PROC CLUSTER PLOTS=ALL

Examples: CLUSTER Procedure

Example 29.1: Cluster Analysis of Flying Mileages between 10 American Cities

This example clusters 10 American cities based on the flying mileages between them. Six clustering methods are shown with corresponding tree diagrams produced by the TREE procedure. The EML method cannot be used because it requires coordinate data. The other omitted methods produce the same clusters, although not the same distances between clusters, as one of the illustrated methods: complete linkage and the flexible-beta method yield the same clusters as Ward's method, McQuitty's similarity analysis produces the same clusters as average linkage, and the median method corresponds to the centroid method.

All of the methods suggest a division of the cities into two clusters along the east-west dimension. There is disagreement, however, about which cluster Denver should belong to. Some of the methods indicate a possible third cluster containing Denver and Houston.

```
title 'Cluster Analysis of Flying Mileages Between 10 American Cities';
data mileages(type=distance);
   input (Atlanta Chicago Denver Houston LosAngeles
         Miami NewYork SanFran Seattle WashDC) (5.)
         @55 City $15.;
   datalines;
    0                                                    Atlanta
  587    0                                              Chicago
 1212  920    0                                        Denver
  701  940  879    0                                   Houston
 1936 1745  831 1374    0                              Los Angeles
  604 1188 1726  968 2339    0                        Miami
  748  713 1631 1420 2451 1092    0                   New York
 2139 1858  949 1645  347 2594 2571    0              San Francisco
 2182 1737 1021 1891  959 2734 2408  678    0         Seattle
  543  597 1494 1220 2300  923  205 2442 2329    0    Washington D.C.
;

goptions htext=0.15in htitle=0.15in;
```

The following statements produce Output 29.1.1 and Output 29.1.2:

```
/*---------------------- Average linkage --------------------*/
proc cluster data=mileages outtree=tree method=average pseudo;
  id City;
run;

title2 'Using METHOD=AVERAGE' ;
proc tree horizontal; id City; run;
title2;
```

Output 29.1.1 Cluster History Using METHOD=AVERAGE

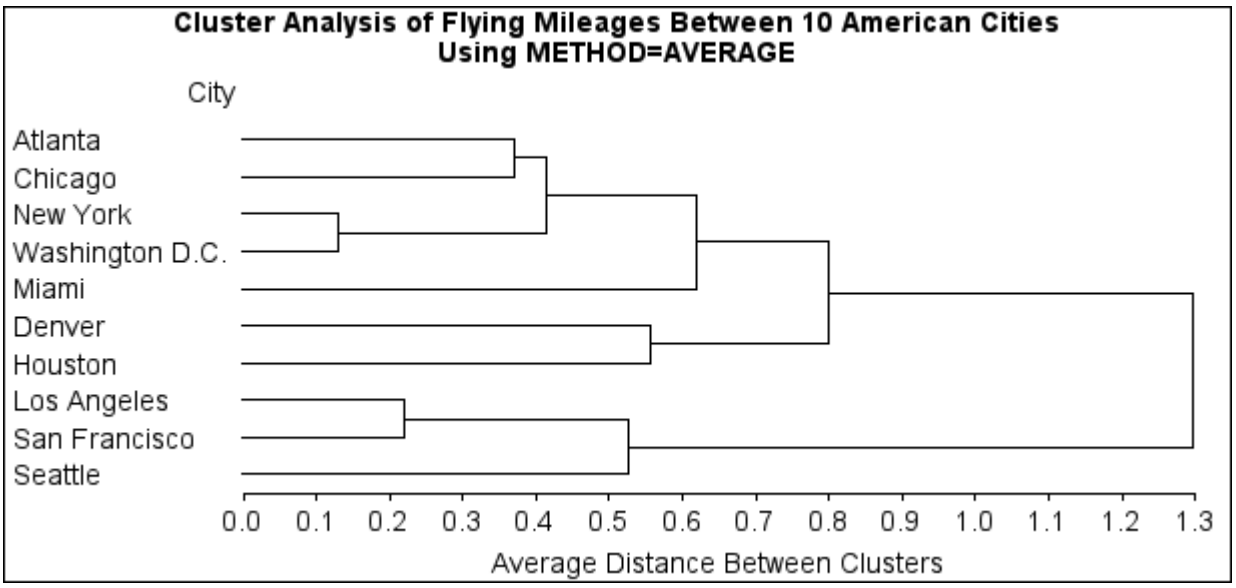
Cluster Analysis of Flying Mileages Between 10 American Cities

The CLUSTER Procedure
Average Linkage Cluster Analysis

Cluster History

NCL	Clusters Joined		FREQ	PSF	PST2	Norm RMS Dist	T i e
9	New York	Washington D.C.	2	66.7	.	0.1297	
8	Los Angeles	San Francisco	2	39.2	.	0.2196	
7	Atlanta	Chicago	2	21.7	.	0.3715	
6	CL7	CL9	4	14.5	3.4	0.4149	
5	CL8	Seattle	3	12.4	7.3	0.5255	
4	Denver	Houston	2	13.9	.	0.5562	
3	CL6	Miami	5	15.5	3.8	0.6185	
2	CL3	CL4	7	16.0	5.3	0.8005	
1	CL2	CL5	10	.	16.0	1.2967	

Output 29.1.2 Tree Diagram Using METHOD=AVERAGE



The following statements produce Output 29.1.3 and Output 29.1.4:

```
/*---------------------- Centroid method --------------------*/
proc cluster data=mileages method=centroid pseudo;
   id City;
run;

title2 'Using METHOD=CENTROID' ;
proc tree horizontal; id City; run;
title2;
```

Output 29.1.3 Cluster History Using METHOD=CENTROID

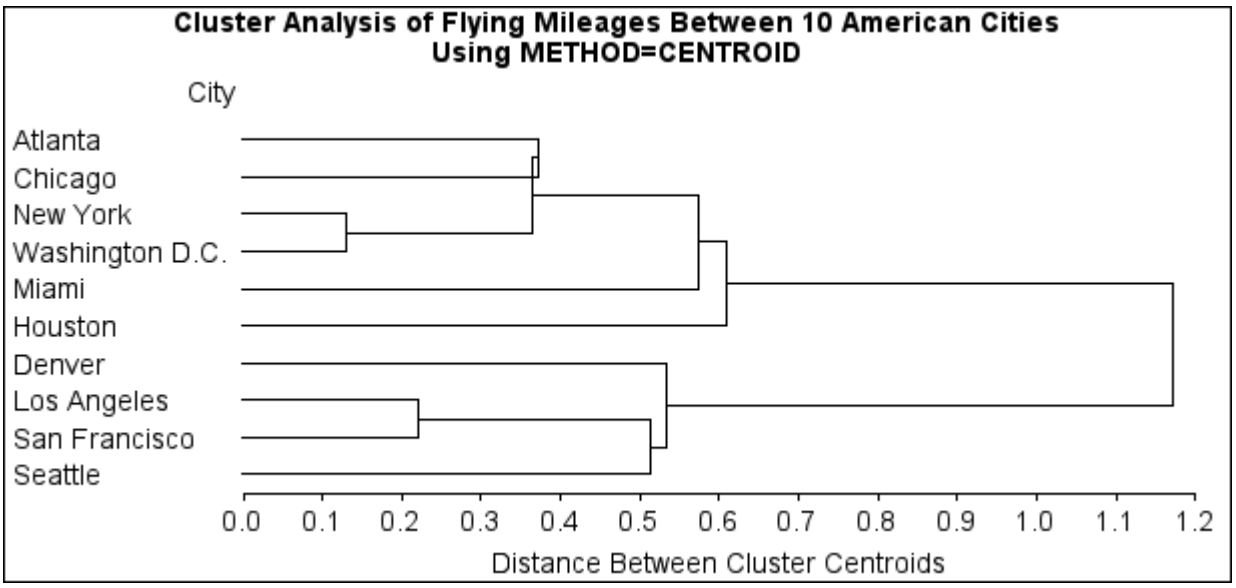
Cluster Analysis of Flying Mileages Between 10 American Cities

The CLUSTER Procedure
Centroid Hierarchical Cluster Analysis

Cluster History

NCL	Clusters Joined		FREQ	PSF	PST2	Norm Cent Dist	T i e
9	New York	Washington D.C.	2	66.7	.	0.1297	
8	Los Angeles	San Francisco	2	39.2	.	0.2196	
7	Atlanta	Chicago	2	21.7	.	0.3715	
6	CL7	CL9	4	14.5	3.4	0.3652	
5	CL8	Seattle	3	12.4	7.3	0.5139	
4	Denver	CL5	4	12.4	2.1	0.5337	
3	CL6	Miami	5	14.2	3.8	0.5743	
2	CL3	Houston	6	22.1	2.6	0.6091	
1	CL2	CL4	10	.	22.1	1.173	

Output 29.1.4 Tree Diagram Using METHOD=CENTROID



The following statements produce Output 29.1.5 and Output 29.1.6:

```
/*-------- Density linkage with 3rd-nearest-neighbor --------*/
proc cluster data=mileages method=density k=3;
   id City;
run;

title2 'Using METHOD=DENSITY K=3' ;
proc tree horizontal; id City; run;
title2;
```

Output 29.1.5 Cluster History Using METHOD=DENSITY K=3

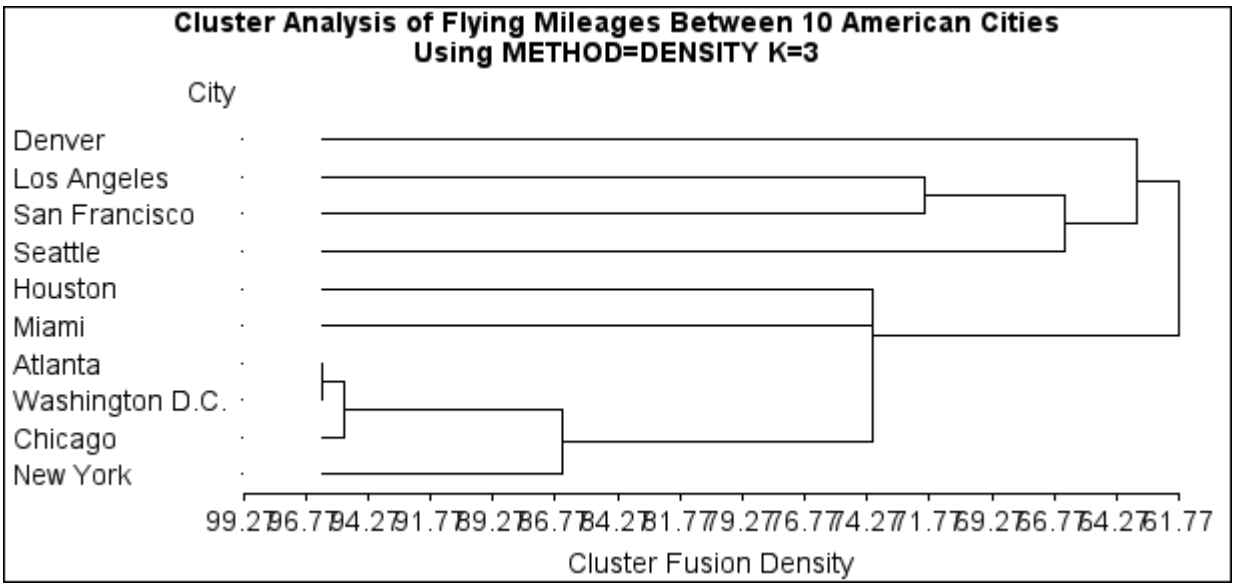
Cluster Analysis of Flying Mileages Between 10 American Cities

The CLUSTER Procedure
Density Linkage Cluster Analysis

Cluster History

NCL	---------Clusters Joined---------		FREQ	Normalized Fusion Density		Maximum Density in Each Cluster Lesser	Maximum Density in Each Cluster Greater	Tie
9	Atlanta	Washington D.C.	2	96.106		92.5043	100.0	
8	CL9	Chicago	3	95.263		90.9548	100.0	
7	CL8	New York	4	86.465		76.1571	100.0	
6	CL7	Miami	5	74.079		58.8299	100.0	T
5	CL6	Houston	6	74.079		61.7747	100.0	
4	Los Angeles	San Francisco	2	71.968		65.3430	80.0885	
3	CL4	Seattle	3	66.341		56.6215	80.0885	
2	CL3	Denver	4	63.509		61.7747	80.0885	
1	CL5	CL2	10	61.775	*	80.0885	100.0	

Output 29.1.6 Tree Diagram Using METHOD=DENSITY K=3



The following statements produce Output 29.1.7 and Output 29.1.8:

```
/*---------------------- Single linkage ----------------------*/
proc cluster data=mileages method=single;
   id City;
run;

title2 'Using METHOD=SINGLE' ;
proc tree horizontal; id City; run;
title2;
```

Output 29.1.7 Cluster History Using METHOD=SINGLE

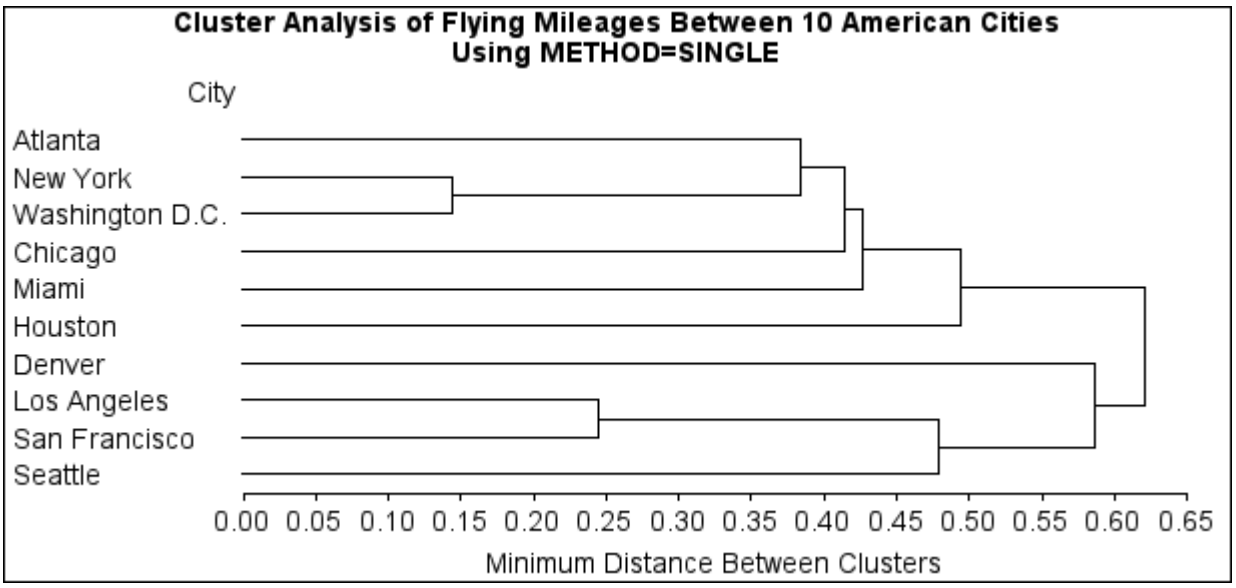
Cluster Analysis of Flying Mileages Between 10 American Cities

The CLUSTER Procedure
Single Linkage Cluster Analysis

Cluster History

NCL	Clusters Joined		FREQ	Norm Min Dist	Time
9	New York	Washington D.C.	2	0.1447	
8	Los Angeles	San Francisco	2	0.2449	
7	Atlanta	CL9	3	0.3832	
6	CL7	Chicago	4	0.4142	
5	CL6	Miami	5	0.4262	
4	CL8	Seattle	3	0.4784	
3	CL5	Houston	6	0.4947	
2	Denver	CL4	4	0.5864	
1	CL3	CL2	10	0.6203	

Output 29.1.8 Tree Diagram Using METHOD=SINGLE



The following statements produce Output 29.1.9 and Output 29.1.10:

```
/*--- Two-stage density linkage with 3rd-nearest-neighbor ---*/
proc cluster data=mileages method=twostage k=3;
   id City;
run;

title2 'Using METHOD=TWOSTAGE K=3' ;
proc tree horizontal; id City; run;
title2;
```

Output 29.1.9 Cluster History Using METHOD=TWOSTAGE K=3

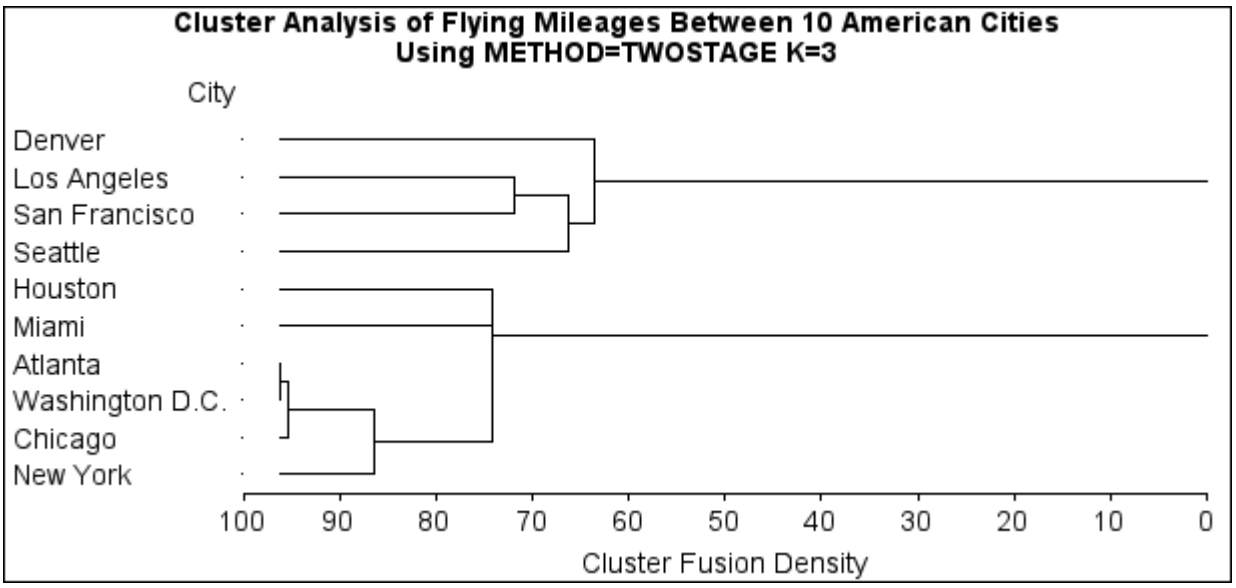
Cluster Analysis of Flying Mileages Between 10 American Cities

The CLUSTER Procedure
Two-Stage Density Linkage Clustering

Cluster History

NCL	---------Clusters Joined---------		FREQ	Normalized Fusion Density	Maximum Density in Each Cluster Lesser	Greater	T i e
9	Atlanta	Washington D.C.	2	96.106	92.5043	100.0	
8	CL9	Chicago	3	95.263	90.9548	100.0	
7	CL8	New York	4	86.465	76.1571	100.0	
6	CL7	Miami	5	74.079	58.8299	100.0	T
5	CL6	Houston	6	74.079	61.7747	100.0	
4	Los Angeles	San Francisco	2	71.968	65.3430	80.0885	
3	CL4	Seattle	3	66.341	56.6215	80.0885	
2	CL3	Denver	4	63.509	61.7747	80.0885	
1	CL5	CL2	10	61.775	80.0885	100.0	

Output 29.1.10 Tree Diagram Using METHOD=TWOSTAGE K=3



The following statements produce Output 29.1.11 and Output 29.1.12:

```
/*------------- Ward's minimum variance method --------------*/
proc cluster data=mileages method=ward pseudo;
   id City;
run;

title2 'Using METHOD=WARD' ;
proc tree horizontal; id City; run;
title2;
```

Output 29.1.11 Cluster History Using METHOD=WARD

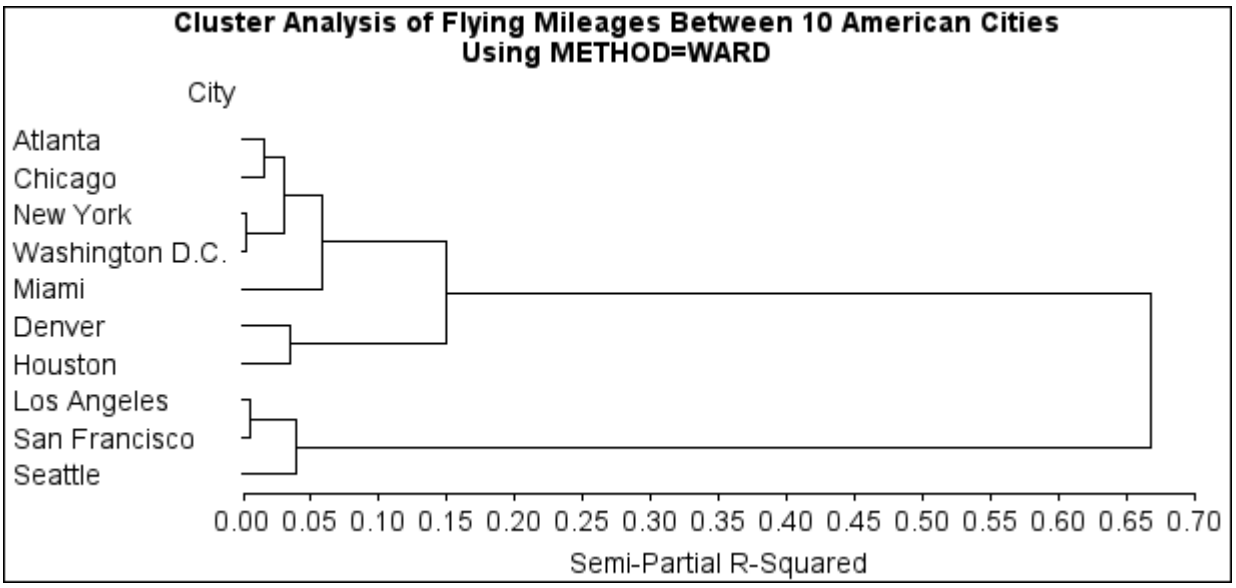
Cluster Analysis of Flying Mileages Between 10 American Cities

The CLUSTER Procedure
Ward's Minimum Variance Cluster Analysis

Cluster History

NCL	Clusters Joined		FREQ	SPRSQ	RSQ	PSF	PST2	Tie
9	New York	Washington D.C.	2	0.0019	.998	66.7	.	
8	Los Angeles	San Francisco	2	0.0054	.993	39.2	.	
7	Atlanta	Chicago	2	0.0153	.977	21.7	.	
6	CL7	CL9	4	0.0296	.948	14.5	3.4	
5	Denver	Houston	2	0.0344	.913	13.2	.	
4	CL8	Seattle	3	0.0391	.874	13.9	7.3	
3	CL6	Miami	5	0.0586	.816	15.5	3.8	
2	CL3	CL5	7	0.1488	.667	16.0	5.3	
1	CL2	CL4	10	0.6669	.000	.	16.0	

Output 29.1.12 Tree Diagram Using METHOD=WARD



Example 29.2: Crude Birth and Death Rates

This example uses the SAS data set Poverty created in the section "Getting Started: CLUSTER Procedure" on page 1231. The data, from Rouncefield (1995), are birth rates, death rates, and infant death rates for 97 countries. Six cluster analyses are performed with eight methods. Scatter plots showing cluster membership at selected levels are produced instead of tree diagrams.

Each cluster analysis is performed by a macro called ANALYZE. The macro takes two arguments. The first, &METHOD, specifies the value of the METHOD= option to be used in the PROC CLUSTER statement. The second, &NCL, must be specified as a list of integers, separated by blanks, indicating the number of clusters desired in each scatter plot. For example, the first invocation of ANALYZE specifies the AVERAGE method and requests plots of 3 and 8 clusters. When two-stage density linkage is used, the K= and R= options are specified as part of the first argument.

The ANALYZE macro first invokes the CLUSTER procedure with METHOD=&METHOD, where &METHOD represents the value of the first argument to ANALYZE. This part of the macro produces the PROC CLUSTER output shown.

The %DO loop processes &NCL, the list of numbers of clusters to plot. The macro variable &K is a counter that indexes the numbers within &NCL. The %SCAN function picks out the &Kth number in &NCL, which is then assigned to the macro variable &N. When &K exceeds the number of numbers in &NCL, %SCAN returns a null string. Thus, the %DO loop executes while &N is not equal to a null string. In the %WHILE condition, a null string is indicated by the absence of any nonblank characters between the comparison operator (NE) and the right parenthesis that terminates the condition.

Within the %DO loop, the TREE procedure creates an output data set containing &N clusters. The SGPLOT procedure then produces a scatter plot in which each observation is identified by the number of the cluster to which it belongs. The TITLE2 statement uses double quotes so that &N and &METHOD can be used within the title. At the end of the loop, &K is incremented by 1, and the next number is extracted from &NCL by %SCAN.

```
title 'Cluster Analysis of Birth and Death Rates';
ods graphics on;

%macro analyze(method,ncl);
   proc cluster data=poverty outtree=tree method=&method print=15 ccc pseudo;
      var birth death;
      title2;
   run;

   %let k=1;
   %let n=%scan(&ncl,&k);
   %do %while(&n NE);

      proc tree data=tree noprint out=out ncl=&n;
         copy birth death;
      run;

      proc sgplot;
         scatter y=death x=birth / group=cluster ;
         title2 "Plot of &n Clusters from METHOD=&METHOD";
      run;

      %let k=%eval(&k+1);
      %let n=%scan(&ncl,&k);
   %end;
%mend;
```

The following statement produces Output 29.2.1, Output 29.2.3, and Output 29.2.4:

```
%analyze(average, 3 8)
```

For average linkage, the CCC has peaks at 3, 8, 10, and 12 clusters, but the 3-cluster peak is lower than the 8-cluster peak. The pseudo F statistic has peaks at 3, 8, and 12 clusters. The pseudo t^2 statistic drops sharply at 3 clusters, continues to fall at 4 clusters, and has a particularly low value at 12 clusters. However, there are not enough data to seriously consider as many as 12 clusters. Scatter plots are given for 3 and 8 clusters. The results are shown in Output 29.2.1 through Output 29.2.4. In Output 29.2.4, the eighth cluster consists of the two outlying observations, Mexico and Korea.

Output 29.2.1 Cluster Analysis for Birth and Death Rates: METHOD=AVERAGE

Cluster Analysis of Birth and Death Rates

The CLUSTER Procedure
Average Linkage Cluster Analysis

Eigenvalues of the Covariance Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	189.106588	173.101020	0.9220	0.9220
2	16.005568		0.0780	1.0000

Root-Mean-Square Total-Sample Standard Deviation 10.127

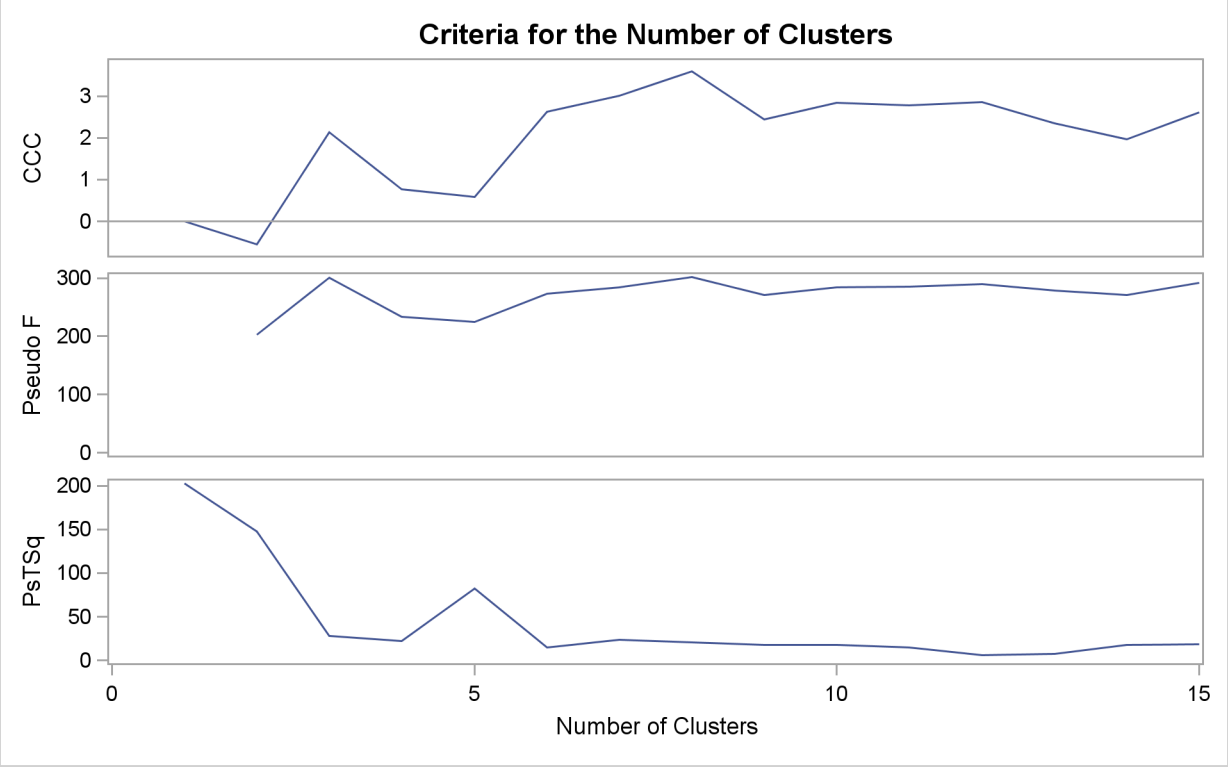
Output 29.2.1 *continued*

Root-Mean-Square Distance Between Observations 20.25399

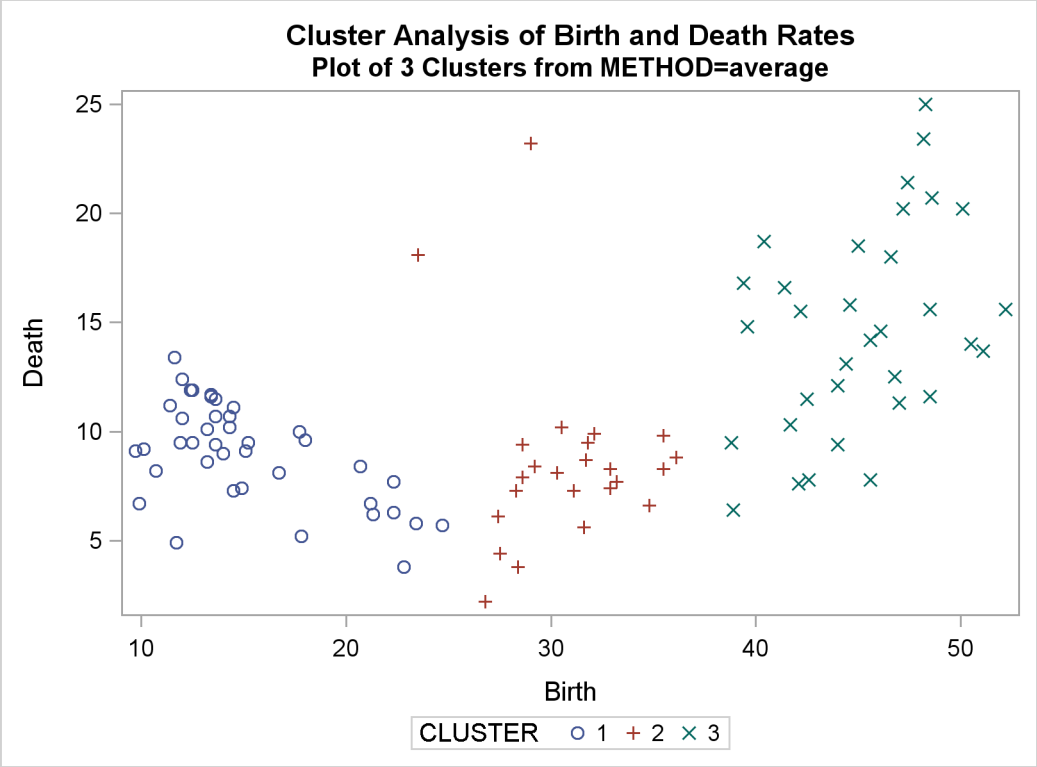
Cluster History

NCL	--Clusters Joined--		FREQ	SPRSQ	RSQ	ERSQ	CCC	PSF	PST2	Norm RMS Dist	T i e
15	CL27	CL20	18	0.0035	.980	.975	2.61	292	18.6	0.2325	
14	CL23	CL17	28	0.0034	.977	.972	1.97	271	17.7	0.2358	
13	CL18	CL54	8	0.0015	.975	.969	2.35	279	7.1	0.2432	
12	CL21	CL26	8	0.0015	.974	.966	2.85	290	6.1	0.2493	
11	CL19	CL24	12	0.0033	.971	.962	2.78	285	14.8	0.2767	
10	CL22	CL16	12	0.0036	.967	.957	2.84	284	17.4	0.2858	
9	CL15	CL28	22	0.0061	.961	.951	2.45	271	17.5	0.3353	
8	OB23	OB61	2	0.0014	.960	.943	3.59	302	.	0.3703	
7	CL25	CL11	17	0.0098	.950	.933	3.01	284	23.3	0.4033	
6	CL7	CL12	25	0.0122	.938	.920	2.63	273	14.8	0.4132	
5	CL10	CL14	40	0.0303	.907	.902	0.59	225	82.7	0.4584	
4	CL13	CL6	33	0.0244	.883	.875	0.77	234	22.2	0.5194	
3	CL9	CL8	24	0.0182	.865	.827	2.13	300	27.7	0.735	
2	CL5	CL3	64	0.1836	.681	.697	-.55	203	148	0.8402	
1	CL2	CL4	97	0.6810	.000	.000	0.00	.	203	1.3348	

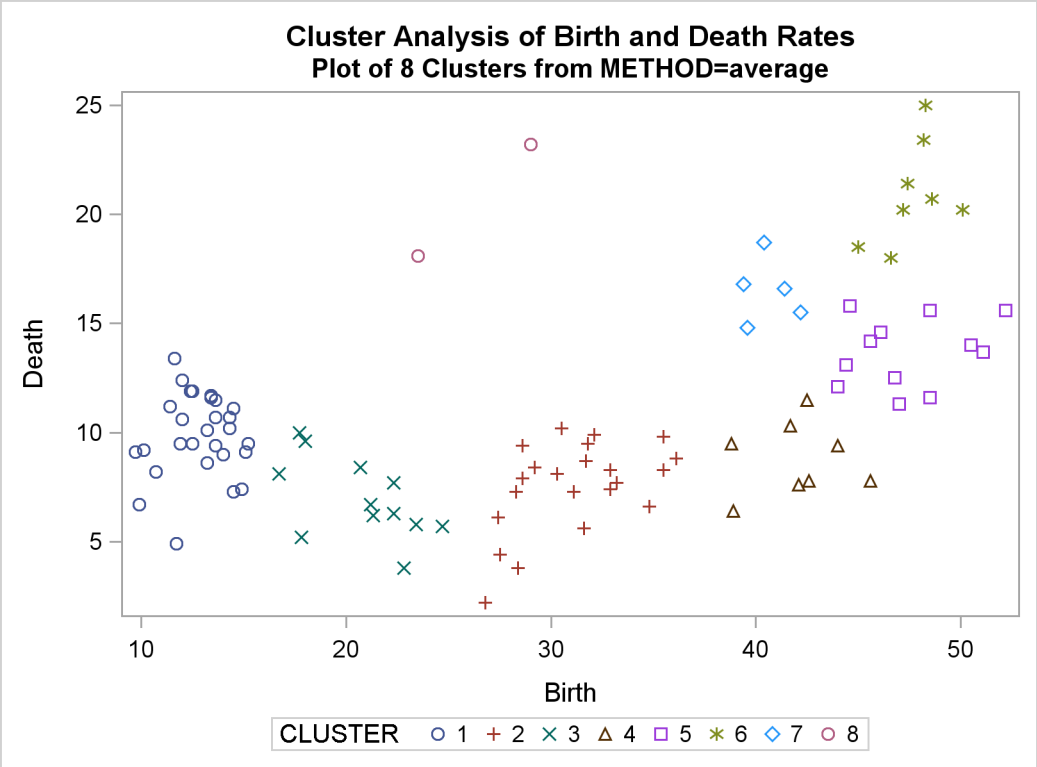
Output 29.2.2 Criteria for the Number of Clusters: METHOD=AVERAGE



Output 29.2.3 Plot of Three Clusters: METHOD=AVERAGE



Output 29.2.4 Plot of Eight Clusters: METHOD=AVERAGE



The following statement produces Output 29.2.5 and Output 29.2.7:

```
%analyze(complete, 3)
```

Complete linkage shows CCC peaks at 3, 8 and 12 clusters. The pseudo F statistic peaks at 3 and 12 clusters. The pseudo t^2 statistic indicates 3 clusters.

The scatter plot for 3 clusters is shown.

Output 29.2.5 Cluster History for Birth and Death Rates: METHOD=COMPLETE

Cluster Analysis of Birth and Death Rates

The CLUSTER Procedure
Complete Linkage Cluster Analysis

Eigenvalues of the Covariance Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	189.106588	173.101020	0.9220	0.9220
2	16.005568		0.0780	1.0000

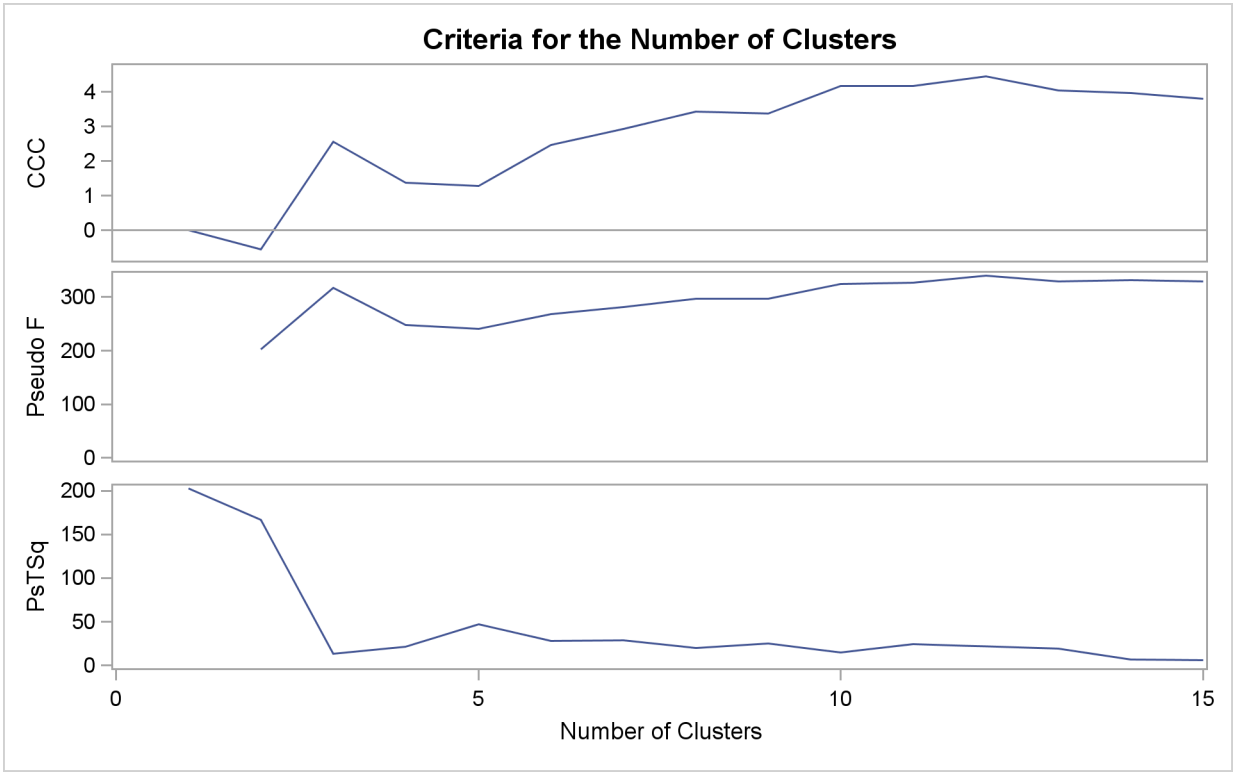
Root-Mean-Square Total-Sample Standard Deviation 10.127

Mean Distance Between Observations 17.13099

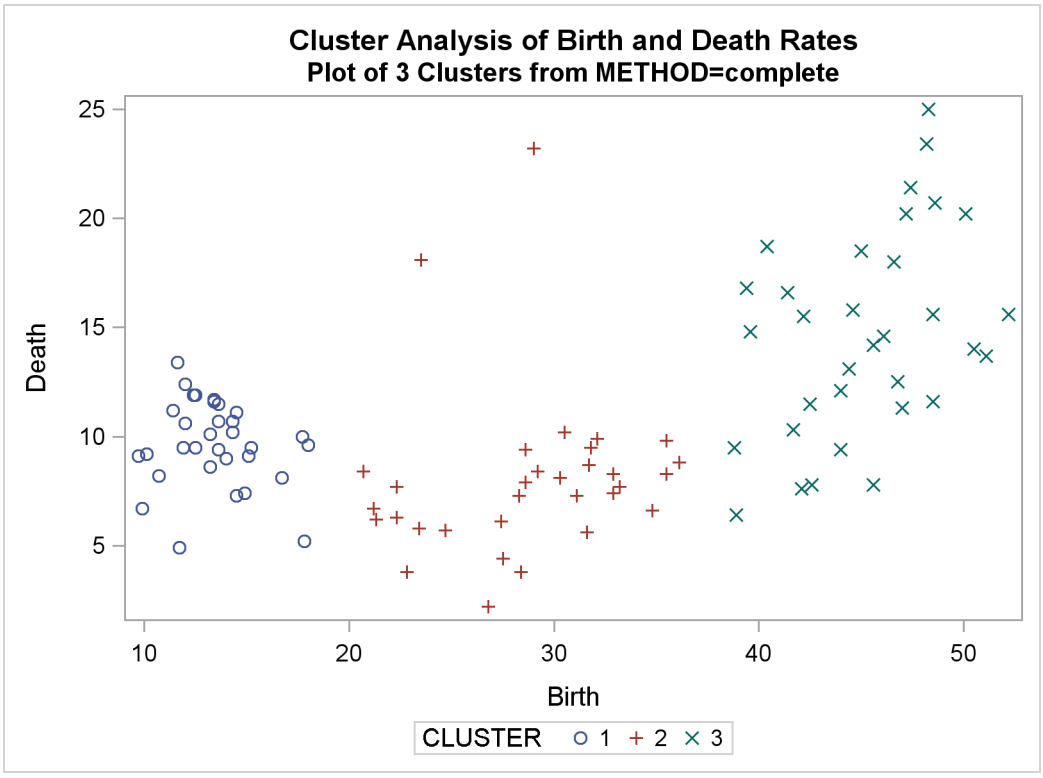
Cluster History

NCL	--Clusters Joined--		FREQ	SPRSQ	RSQ	ERSQ	CCC	PSF	PST2	Norm Max Dist	Tie
15	CL22	CL33	8	0.0015	.983	.975	3.80	329	6.1	0.4092	
14	CL56	CL18	8	0.0014	.981	.972	3.97	331	6.6	0.4255	
13	CL30	CL44	8	0.0019	.979	.969	4.04	330	19.0	0.4332	
12	OB23	OB61	2	0.0014	.978	.966	4.45	340	.	0.4378	
11	CL19	CL24	24	0.0034	.974	.962	4.17	327	24.1	0.4962	
10	CL17	CL28	12	0.0033	.971	.957	4.18	325	14.8	0.5204	
9	CL20	CL13	16	0.0067	.964	.951	3.38	297	25.2	0.5236	
8	CL11	CL21	32	0.0054	.959	.943	3.44	297	19.7	0.6001	
7	CL26	CL15	13	0.0096	.949	.933	2.93	282	28.9	0.7233	
6	CL14	CL10	20	0.0128	.937	.920	2.46	269	27.7	0.8033	
5	CL9	CL16	30	0.0237	.913	.902	1.29	241	47.1	0.8993	
4	CL6	CL7	33	0.0240	.889	.875	1.38	248	21.7	1.2165	
3	CL5	CL12	32	0.0178	.871	.827	2.56	317	13.6	1.2326	
2	CL3	CL8	64	0.1900	.681	.697	-.55	203	167	1.5412	
1	CL2	CL4	97	0.6810	.000	.000	0.00	.	203	2.5233	

Output 29.2.6 Criteria for the Number of Clusters: METHOD=COMPLETE



Output 29.2.7 Plot of Clusters for METHOD=COMPLETE



The following statement produces Output 29.2.8 and Output 29.2.10:

```
%analyze(single, 7 10)
```

The CCC and pseudo F statistics are not appropriate for use with single linkage because of the method's tendency to chop off tails of distributions. The pseudo t^2 statistic can be used by looking for *large* values and taking the number of clusters to be one greater than the level at which the large pseudo t^2 value is displayed. For these data, there are large values at levels 6 and 9, suggesting 7 or 10 clusters.

The scatter plots for 7 and 10 clusters are shown.

Output 29.2.8 Cluster History for Birth and Death Rates: METHOD=SINGLE

Cluster Analysis of Birth and Death Rates

The CLUSTER Procedure
Single Linkage Cluster Analysis

Eigenvalues of the Covariance Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	189.106588	173.101020	0.9220	0.9220
2	16.005568		0.0780	1.0000

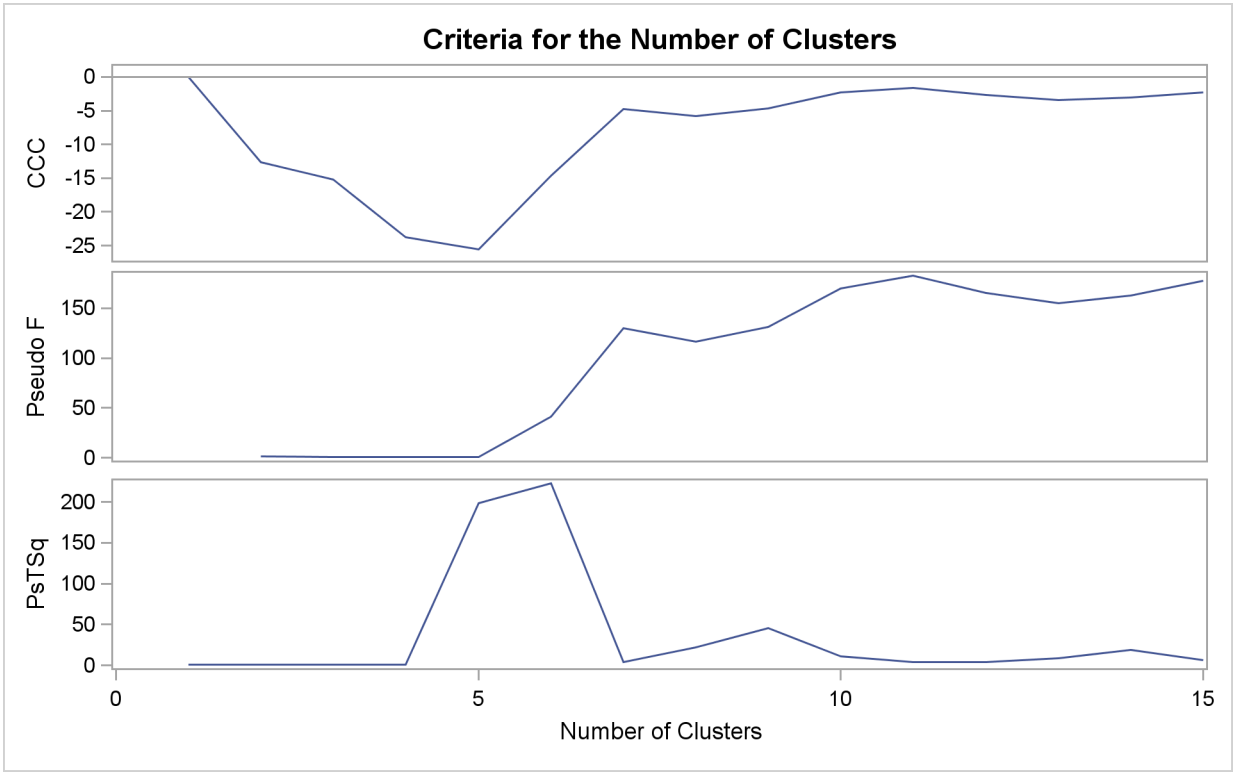
Root-Mean-Square Total-Sample Standard Deviation 10.127

Mean Distance Between Observations 17.13099

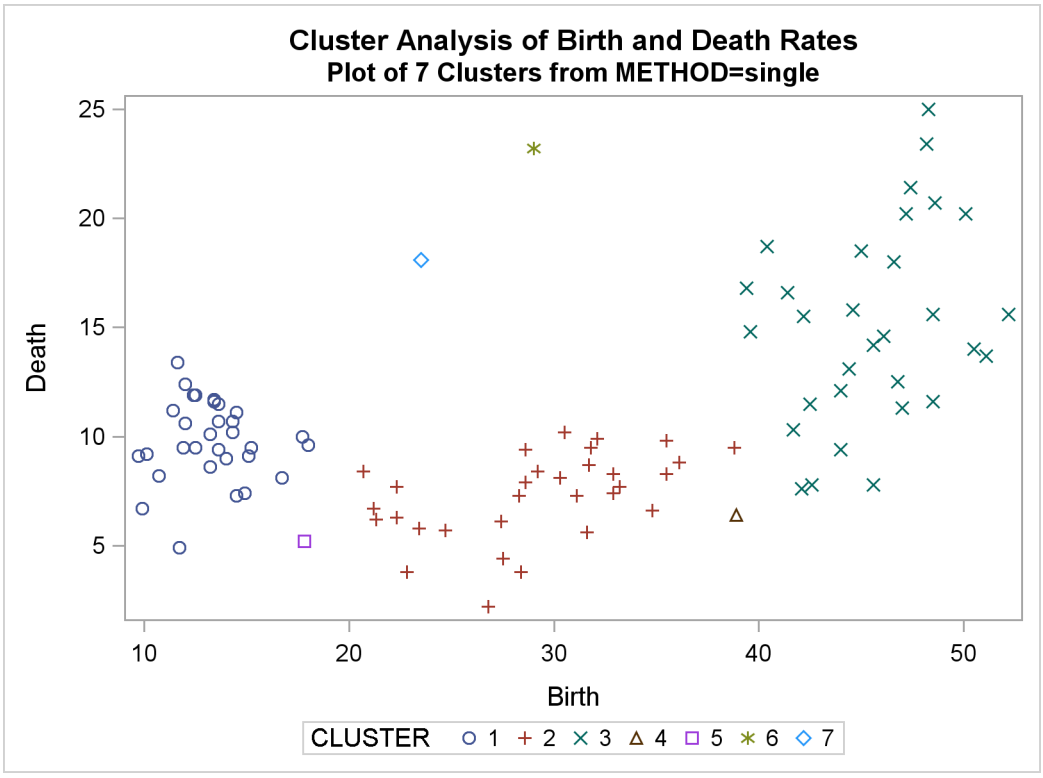
Cluster History

NCL	--Clusters Joined--		FREQ	SPRSQ	RSQ	ERSQ	CCC	PSF	PST2	Norm Min Dist	Tie
15	CL37	CL19	8	0.0014	.968	.975	-2.3	178	6.6	0.1331	
14	CL20	CL23	15	0.0059	.962	.972	-3.1	162	18.7	0.1412	
13	CL14	CL16	19	0.0054	.957	.969	-3.4	155	8.8	0.1442	
12	CL26	OB58	31	0.0014	.955	.966	-2.7	165	4.0	0.1486	
11	OB86	CL18	4	0.0003	.955	.962	-1.6	183	3.8	0.1495	
10	CL13	CL11	23	0.0088	.946	.957	-2.3	170	11.3	0.1518	
9	CL22	CL17	30	0.0235	.923	.951	-4.7	131	45.7	0.1593	T
8	CL15	CL10	31	0.0210	.902	.943	-5.8	117	21.8	0.1593	
7	CL9	OB75	31	0.0052	.897	.933	-4.7	130	4.0	0.1628	
6	CL7	CL12	62	0.2023	.694	.920	-15	41.3	223	0.1725	
5	CL6	CL8	93	0.6681	.026	.902	-26	0.6	199	0.1756	
4	CL5	OB48	94	0.0056	.021	.875	-24	0.7	0.5	0.1811	T
3	CL4	OB67	95	0.0083	.012	.827	-15	0.6	0.8	0.1811	
2	OB23	OB61	2	0.0014	.011	.697	-13	1.0	.	0.4378	
1	CL3	CL2	97	0.0109	.000	.000	0.00	.	1.0	0.5815	

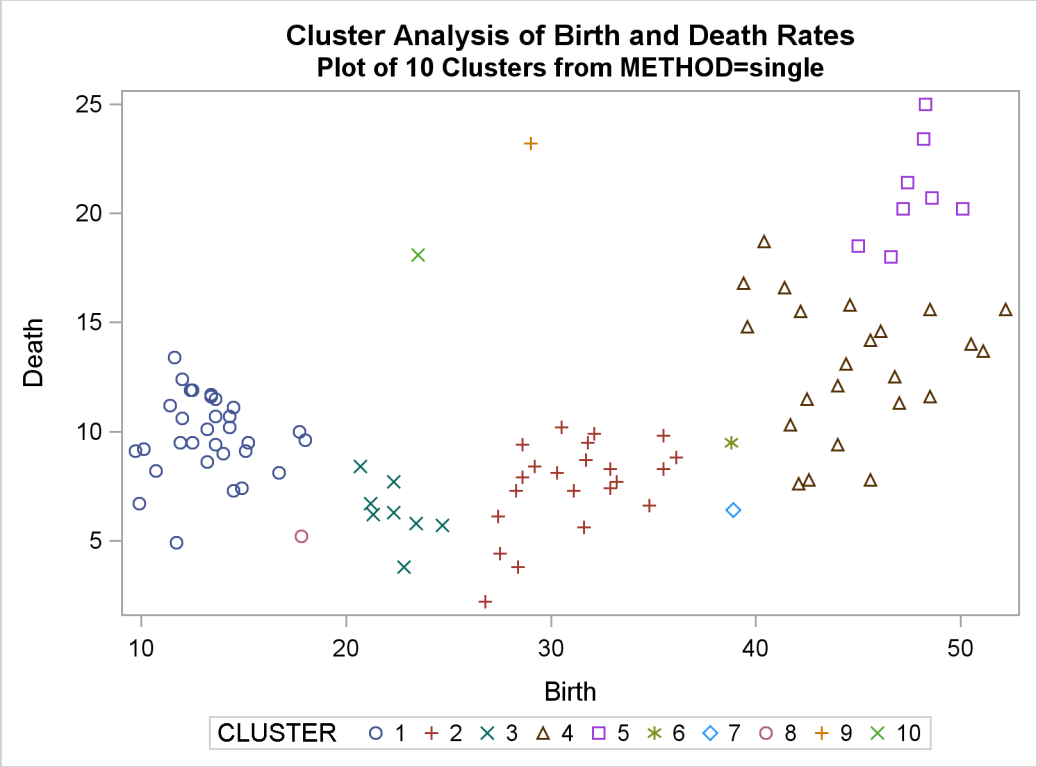
Output 29.2.9 Criteria for the Number of Clusters: METHOD=SINGLE



Output 29.2.10 Plot of Clusters for METHOD=SINGLE



Output 29.2.10 *continued*



The following statements produce Output 29.2.11 through Output 29.2.14, :

```
%analyze(two k=10, 3)

%analyze(two k=18, 2)
```

For kth-nearest-neighbor density linkage, the number of modes as a function of k is as follows (not all of these analyses are shown):

k	modes
3	13
4	6
5-7	4
8-15	3
16-21	2
22+	1

Thus, there is strong evidence of 3 modes and an indication of the possibility of 2 modes. Uniform-kernel density linkage gives similar results. For K=10 (10th-nearest-neighbor density linkage), the scatter plot for 3 clusters is shown; and for K=18, the scatter plot for 2 clusters is shown.

Output 29.2.11 Cluster History for Birth and Death Rates: METHOD=TWOSTAGE K=10

Cluster Analysis of Birth and Death Rates

The CLUSTER Procedure
Two-Stage Density Linkage Clustering

Eigenvalues of the Covariance Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	189.106588	173.101020	0.9220	0.9220
2	16.005568		0.0780	1.0000

K = 10

Root-Mean-Square Total-Sample Standard Deviation 10.127

Cluster History

NCL	--Clusters Joined--		FREQ	SPRSQ	RSQ	ERSQ	CCC	PSF	PST2	Normalized Fusion Density	Maximum Density in Each Cluster Lesser	Greater	Tie
15	CL16	OB94	22	0.0015	.921	.975	-11	68.4	1.4	9.2234	6.7927	15.3069	
14	CL19	OB49	28	0.0021	.919	.972	-11	72.4	1.8	8.7369	5.9334	33.4385	
13	CL15	OB52	23	0.0024	.917	.969	-10	76.9	2.3	8.5847	5.9651	15.3069	
12	CL13	OB96	24	0.0018	.915	.966	-9.3	83.0	1.6	7.9252	5.4724	15.3069	
11	CL12	OB93	25	0.0025	.912	.962	-8.5	89.5	2.2	7.8913	5.4401	15.3069	
10	CL11	OB78	26	0.0031	.909	.957	-7.7	96.9	2.5	7.787	5.4082	15.3069	
9	CL10	OB76	27	0.0026	.907	.951	-6.7	107	2.1	7.7133	5.4401	15.3069	
8	CL9	OB77	28	0.0023	.904	.943	-5.5	120	1.7	7.4256	4.9017	15.3069	
7	CL8	OB43	29	0.0022	.902	.933	-4.1	138	1.6	6.927	4.4764	15.3069	
6	CL7	OB87	30	0.0043	.898	.920	-2.7	160	3.1	4.932	2.9977	15.3069	
5	CL6	OB82	31	0.0055	.892	.902	-1.1	191	3.7	3.7331	2.1560	15.3069	
4	CL22	OB61	37	0.0079	.884	.875	0.93	237	10.6	3.1713	1.6308	100.0	
3	CL14	OB23	29	0.0126	.872	.827	2.60	320	10.4	2.0654	1.0744	33.4385	
2	CL4	CL3	66	0.2129	.659	.697	-1.3	183	172	12.409	33.4385	100.0	
1	CL2	CL5	97	0.6588	.000	.000	0.00	.	183	10.071	15.3069	100.0	

3 modal clusters have been formed.

Output 29.2.12 Cluster History for Birth and Death Rates: METHOD=TWOSTAGE K=18

Cluster Analysis of Birth and Death Rates

The CLUSTER Procedure
Two-Stage Density Linkage Clustering

Eigenvalues of the Covariance Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	189.106588	173.101020	0.9220	0.9220
2	16.005568		0.0780	1.0000

K = 18

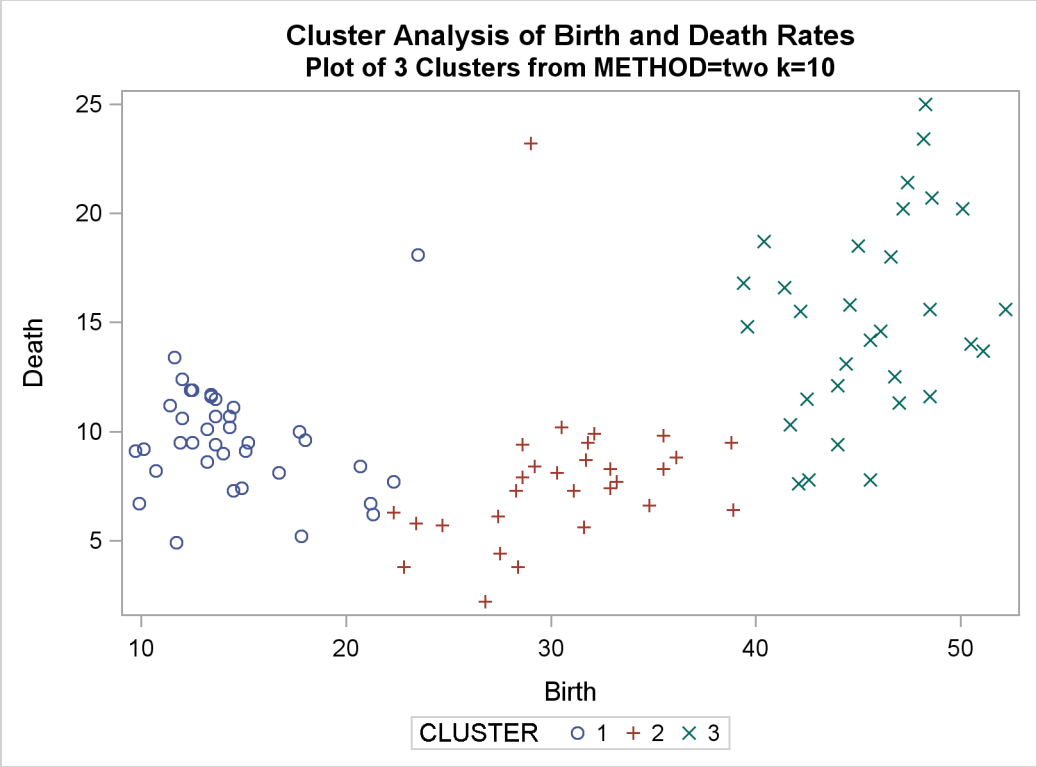
Root-Mean-Square Total-Sample Standard Deviation 10.127

Cluster History

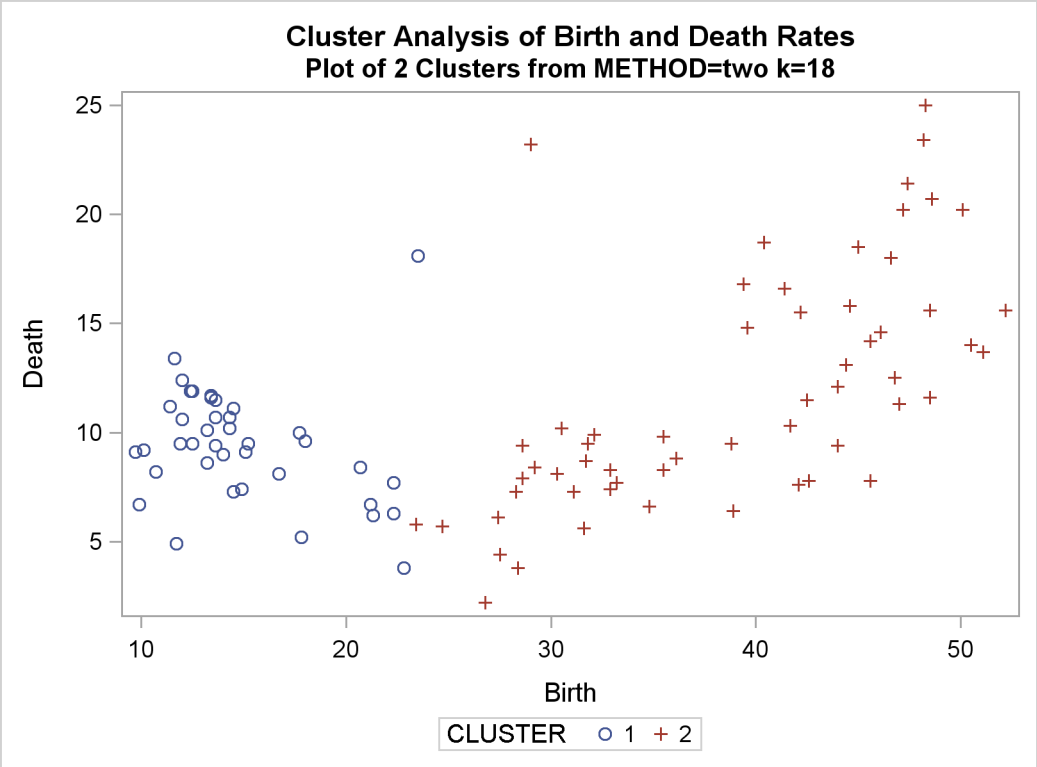
NCL	--Clusters Joined--		FREQ	SPRSQ	RSQ	ERSQ	CCC	PSF	PST2	Normalized Fusion Density	Maximum Density in Each Cluster Lesser	Maximum Density in Each Cluster Greater	Tie
15	CL16	OB72	46	0.0107	.799	.975	−21	23.3	3.0	10.118	7.7445	23.4457	
14	CL15	OB94	47	0.0098	.789	.972	−21	23.9	2.7	9.676	7.1257	23.4457	
13	CL14	OB51	48	0.0037	.786	.969	−20	25.6	1.0	9.409	6.8398	23.4457	T
12	CL13	OB96	49	0.0099	.776	.966	−19	26.7	2.6	9.409	6.8398	23.4457	
11	CL12	OB76	50	0.0114	.764	.962	−19	27.9	2.9	8.8136	6.3138	23.4457	
10	CL11	OB77	51	0.0021	.762	.957	−18	31.0	0.5	8.6593	6.0751	23.4457	
9	CL10	OB78	52	0.0103	.752	.951	−17	33.3	2.5	8.6007	6.0976	23.4457	
8	CL9	OB43	53	0.0034	.748	.943	−16	37.8	0.8	8.4964	5.9160	23.4457	
7	CL8	OB93	54	0.0109	.737	.933	−15	42.1	2.6	8.367	5.7913	23.4457	
6	CL7	OB88	55	0.0110	.726	.920	−13	48.3	2.6	7.916	5.3679	23.4457	
5	CL6	OB87	56	0.0120	.714	.902	−12	57.5	2.7	6.6917	4.3415	23.4457	
4	CL20	OB61	39	0.0077	.707	.875	−9.8	74.7	8.3	6.2578	3.2882	100.0	
3	CL5	OB82	57	0.0138	.693	.827	−5.0	106	3.0	5.3605	3.2834	23.4457	
2	CL3	OB23	58	0.0117	.681	.697	−.54	203	2.5	3.2687	1.7568	23.4457	
1	CL2	CL4	97	0.6812	.000	.000	0.00	.	203	13.764	23.4457	100.0	

2 modal clusters have been formed.

Output 29.2.13 Plot of Clusters for METHOD=TWOSTAGE K=10



Output 29.2.14 Plot of Clusters for METHOD=TWOSTAGE K=18



In summary, most of the clustering methods indicate 3 or 8 clusters. Most methods agree at the 3-cluster level, but at the other levels, there is considerable disagreement about the composition of the clusters. The presence of numerous ties also complicates the analysis; see Example 29.4.

Example 29.3: Cluster Analysis of Fisher's Iris Data

The iris data published by Fisher (1936) have been widely used for examples in discriminant analysis and cluster analysis. The sepal length, sepal width, petal length, and petal width are measured in millimeters on 50 iris specimens from each of three species, *Iris setosa, I. versicolor,* and *I. virginica*. Mezzich and Solomon (1980) discuss a variety of cluster analyses of the iris data.

The following code analyzes the iris data by using Ward's method and two-stage density linkage and then illustrates how the FASTCLUS procedure can be used in combination with PROC CLUSTER to analyze large data sets.

```
   title 'Cluster Analysis of Fisher (1936) Iris Data';
   proc format;
      value specname
         1='Setosa    '
         2='Versicolor'
         3='Virginica ';
   run;

data iris;
   input SepalLength SepalWidth PetalLength PetalWidth Species @@;
   format Species specname.;
   label SepalLength='Sepal Length in mm.'
         SepalWidth ='Sepal Width in mm.'
         PetalLength='Petal Length in mm.'
         PetalWidth ='Petal Width in mm.';
   symbol = put(species, specname10.);
   datalines;
50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3
63 28 51 15 3 46 34 14 03 1 69 31 51 23 3 62 22 45 15 2
59 32 48 18 2 46 36 10 02 1 61 30 46 14 2 60 27 51 16 2
65 30 52 20 3 56 25 39 11 2 65 30 55 18 3 58 27 51 19 3
68 32 59 23 3 51 33 17 05 1 57 28 45 13 2 62 34 54 23 3
77 38 67 22 3 63 33 47 16 2 67 33 57 25 3 76 30 66 21 3
49 25 45 17 3 55 35 13 02 1 67 30 52 23 3 70 32 47 14 2
64 32 45 15 2 61 28 40 13 2 48 31 16 02 1 59 30 51 18 3
55 24 38 11 2 63 25 50 19 3 64 32 53 23 3 52 34 14 02 1
49 36 14 01 1 54 30 45 15 2 79 38 64 20 3 44 32 13 02 1
67 33 57 21 3 50 35 16 06 1 58 26 40 12 2 44 30 13 02 1
77 28 67 20 3 63 27 49 18 3 47 32 16 02 1 55 26 44 12 2
50 23 33 10 2 72 32 60 18 3 48 30 14 03 1 51 38 16 02 1
61 30 49 18 3 48 34 19 02 1 50 30 16 02 1 50 32 12 02 1
61 26 56 14 3 64 28 56 21 3 43 30 11 01 1 58 40 12 02 1
51 38 19 04 1 67 31 44 14 2 62 28 48 18 3 49 30 14 02 1
51 35 14 02 1 56 30 45 15 2 58 27 41 10 2 50 34 16 04 1
46 32 14 02 1 60 29 45 15 2 57 26 35 10 2 57 44 15 04 1
50 36 14 02 1 77 30 61 23 3 63 34 56 24 3 58 27 51 19 3
```

```
57 29 42 13 2 72 30 58 16 3 54 34 15 04 1 52 41 15 01 1
71 30 59 21 3 64 31 55 18 3 60 30 48 18 3 63 29 56 18 3
49 24 33 10 2 56 27 42 13 2 57 30 42 12 2 55 42 14 02 1
49 31 15 02 1 77 26 69 23 3 60 22 50 15 3 54 39 17 04 1
66 29 46 13 2 52 27 39 14 2 60 34 45 16 2 50 34 15 02 1
44 29 14 02 1 50 20 35 10 2 55 24 37 10 2 58 27 39 12 2
47 32 13 02 1 46 31 15 02 1 69 32 57 23 3 62 29 43 13 2
74 28 61 19 3 59 30 42 15 2 51 34 15 02 1 50 35 13 03 1
56 28 49 20 3 60 22 40 10 2 73 29 63 18 3 67 25 58 18 3
49 31 15 01 1 67 31 47 15 2 63 23 44 13 2 54 37 15 02 1
56 30 41 13 2 63 25 49 15 2 61 28 47 12 2 64 29 43 13 2
51 25 30 11 2 57 28 41 13 2 65 30 58 22 3 69 31 54 21 3
54 39 13 04 1 51 35 14 03 1 72 36 61 25 3 65 32 51 20 3
61 29 47 14 2 56 29 36 13 2 69 31 49 15 2 64 27 53 19 3
68 30 55 21 3 55 25 40 13 2 48 34 16 02 1 48 30 14 01 1
45 23 13 03 1 57 25 50 20 3 57 38 17 03 1 51 38 15 03 1
55 23 40 13 2 66 30 44 14 2 68 28 48 14 2 54 34 17 02 1
51 37 15 04 1 52 35 15 02 1 58 28 51 24 3 67 30 50 17 2
63 33 60 25 3 53 37 15 02 1
;
```

The following macro, SHOW, is used in the subsequent analyses to display cluster results. It invokes the FREQ procedure to crosstabulate clusters and species. The CANDISC procedure computes canonical variables for discriminating among the clusters, and the first two canonical variables are plotted to show cluster membership. See Chapter 27, "The CANDISC Procedure," for a canonical discriminant analysis of the iris species.

```
/*--- Define macro show ---*/
%macro show;
   proc freq;
      tables cluster*species / nopercent norow nocol plot=none;
   run;
   proc candisc noprint out=can;
      class cluster;
      var petal: sepal:;
   run;
   proc sgplot data=can ;
      scatter y=can2 x=can1 / group=cluster ;
   run;
%mend;
```

The first analysis clusters the iris data by using Ward's method (see Output 29.3.1) and plots the CCC and pseudo F and t^2 statistics (see Output 29.3.2). The CCC has a local peak at 3 clusters but a higher peak at 5 clusters. The pseudo F statistic indicates 3 clusters, while the pseudo t^2 statistic suggests 3 or 6 clusters.

The TREE procedure creates an output data set containing the 3-cluster partition for use by the SHOW macro. The FREQ procedure reveals 16 misclassifications. The results are shown in Output 29.3.3.

```
title2 'By Ward''s Method';
ods graphics on ;
proc cluster data=iris method=ward print=15 ccc pseudo;
   var petal: sepal:;
   copy species;
run;

proc tree noprint ncl=3 out=out;
   copy petal: sepal: species;
run;

%show;
```

Output 29.3.1 Cluster Analysis of Fisher's Iris Data: PROC CLUSTER with METHOD=WARD

Cluster Analysis of Fisher (1936) Iris Data
By Ward's Method

The CLUSTER Procedure
Ward's Minimum Variance Cluster Analysis

Eigenvalues of the Covariance Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	422.824171	398.557096	0.9246	0.9246
2	24.267075	16.446125	0.0531	0.9777
3	7.820950	5.437441	0.0171	0.9948
4	2.383509		0.0052	1.0000

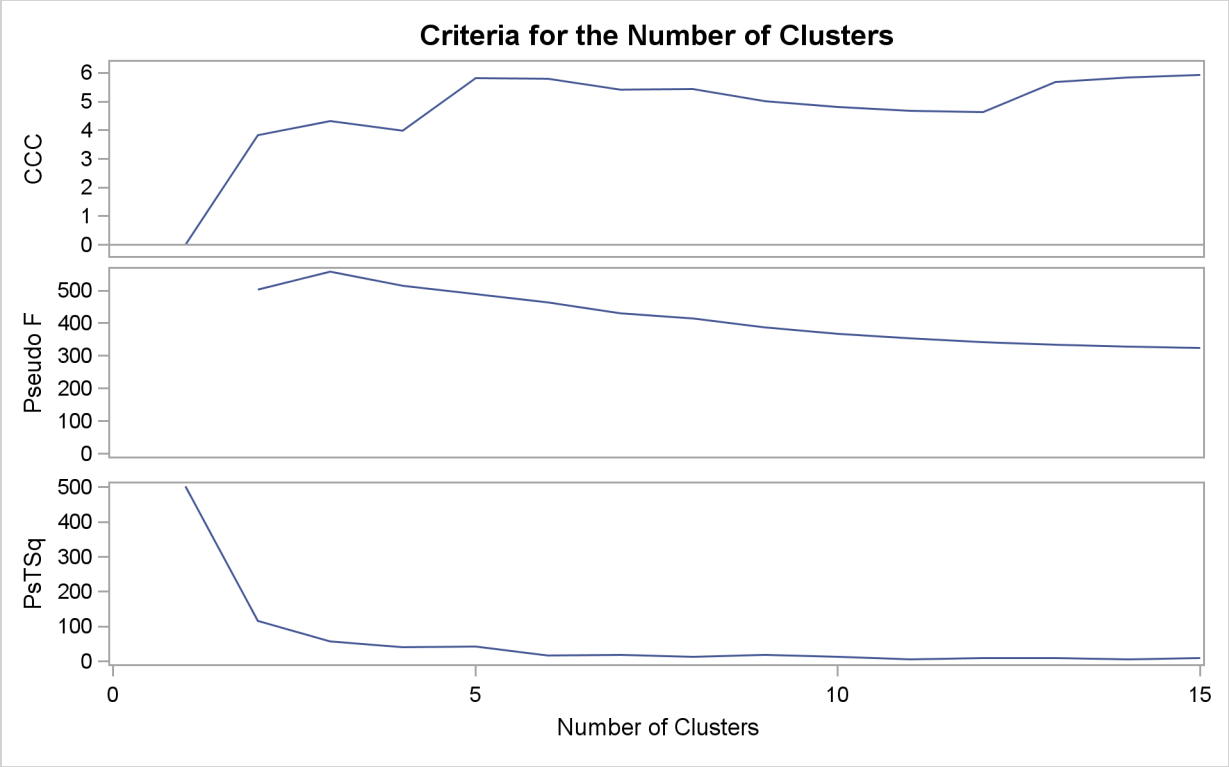
Root-Mean-Square Total-Sample Standard Deviation 10.69224

Root-Mean-Square Distance Between Observations 30.24221

Cluster History

NCL	--Clusters Joined---		FREQ	SPRSQ	RSQ	ERSQ	CCC	PSF	PST2	Tie
15	CL24	CL28	15	0.0016	.971	.958	5.93	324	9.8	
14	CL21	CL53	7	0.0019	.969	.955	5.85	329	5.1	
13	CL18	CL48	15	0.0023	.967	.953	5.69	334	8.9	
12	CL16	CL23	24	0.0023	.965	.950	4.63	342	9.6	
11	CL14	CL43	12	0.0025	.962	.946	4.67	353	5.8	
10	CL26	CL20	22	0.0027	.959	.942	4.81	368	12.9	
9	CL27	CL17	31	0.0031	.956	.936	5.02	387	17.8	
8	CL35	CL15	23	0.0031	.953	.930	5.44	414	13.8	
7	CL10	CL47	26	0.0058	.947	.921	5.43	430	19.1	
6	CL8	CL13	38	0.0060	.941	.911	5.81	463	16.3	
5	CL9	CL19	50	0.0105	.931	.895	5.82	488	43.2	
4	CL12	CL11	36	0.0172	.914	.872	3.99	515	41.0	
3	CL6	CL7	64	0.0301	.884	.827	4.33	558	57.2	
2	CL4	CL3	100	0.1110	.773	.697	3.83	503	116	
1	CL5	CL2	150	0.7726	.000	.000	0.00	.	503	

Output 29.3.2 Criteria for the Number of Clusters with METHOD=WARD



Output 29.3.3 Crosstabulation of Clusters for METHOD=WARD

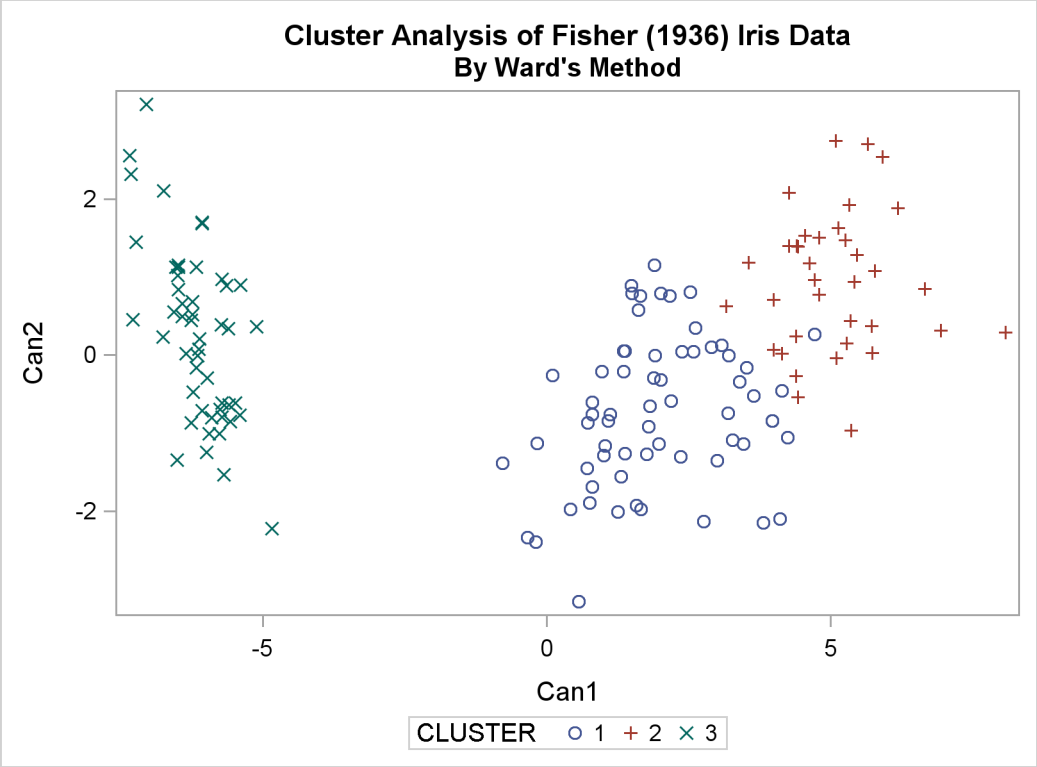
Cluster Analysis of Fisher (1936) Iris Data
By Ward's Method

The FREQ Procedure

Table of CLUSTER by Species

CLUSTER / Species	Setosa	Versicolor	Virginica	Total
1	0	49	15	64
2	0	1	35	36
3	50	0	0	50
Total	50	50	50	150

Output 29.3.4 Scatter Plot of Clusters for METHOD=WARD



The second analysis uses two-stage density linkage. The raw data suggest 2 or 6 modes instead of 3:

k	**modes**
3	12
4-6	6
7	4
8	3
9-50	2
51+	1

The following analysis uses K=8 to produce 3 clusters for comparison with other analyses. There are only 6 misclassifications. The results are shown in Output 29.3.5 and Output 29.3.6.

```
title2 'By Two-Stage Density Linkage';
ods graphics on ;
proc cluster data=iris method=twostage k=8 print=15 ccc pseudo;
   var petal: sepal:;
   copy species;
run;

proc tree noprint ncl=3 out=out;
   copy petal: sepal: species;
run;

%show;
```

Output 29.3.5 Cluster Analysis of Fisher's Iris Data: PROC CLUSTER with METHOD=TWOSTAGE

Cluster Analysis of Fisher (1936) Iris Data
By Two-Stage Density Linkage

The CLUSTER Procedure
Two-Stage Density Linkage Clustering

Eigenvalues of the Covariance Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	422.824171	398.557096	0.9246	0.9246
2	24.267075	16.446125	0.0531	0.9777
3	7.820950	5.437441	0.0171	0.9948
4	2.383509		0.0052	1.0000

K = 8

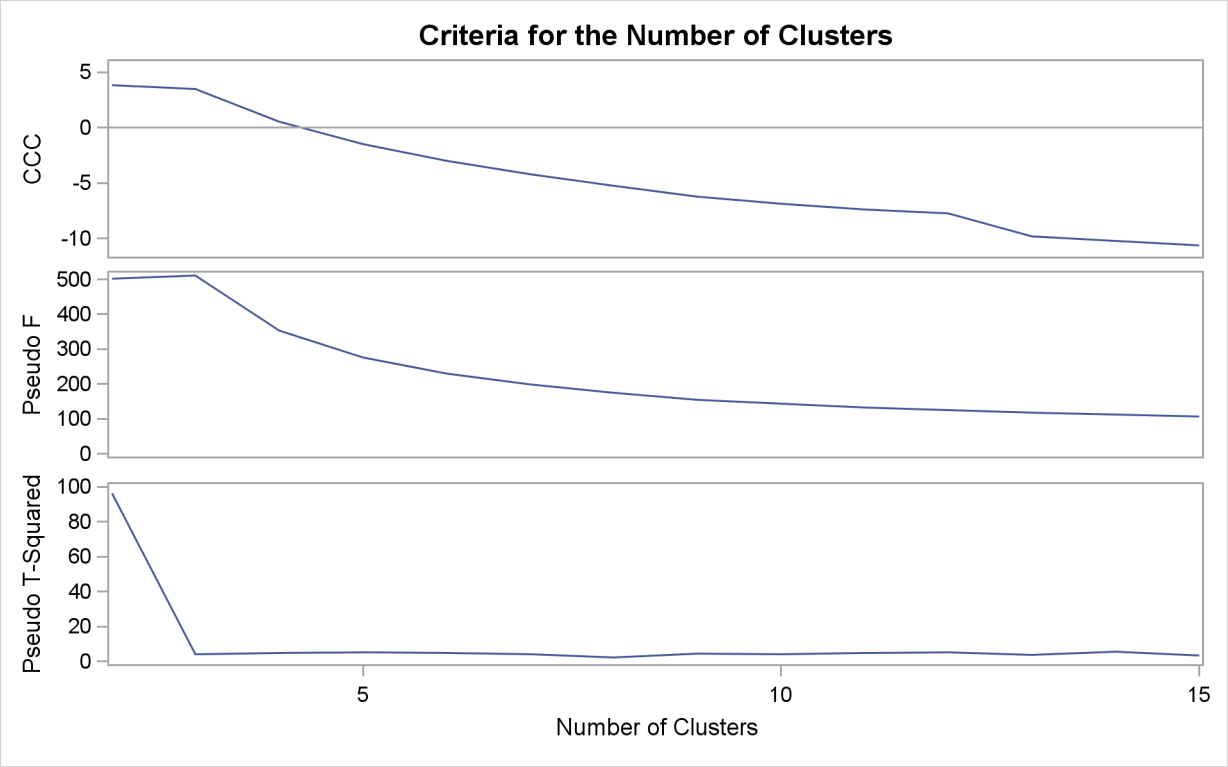
Root-Mean-Square Total-Sample Standard Deviation 10.69224

Cluster History

NCL	--Clusters Joined--		FREQ	SPRSQ	RSQ	ERSQ	CCC	PSF	PST2	Normalized Fusion Density	Maximum Density in Each Cluster Lesser	Maximum Density in Each Cluster Greater	Tie
15	CL17	OB127	43	0.0024	.917	.958	−11	107	3.4	0.3903	0.2066	3.5156	
14	CL16	OB137	50	0.0023	.915	.955	−10	113	5.6	0.3637	0.1837	100.0	
13	CL15	OB74	44	0.0029	.912	.953	−9.8	119	3.8	0.3553	0.2130	3.5156	
12	CL22	OB49	47	0.0036	.909	.950	−7.7	125	5.2	0.3223	0.1736	8.3678	T
11	CL12	OB85	48	0.0036	.905	.946	−7.4	132	4.8	0.3223	0.1736	8.3678	
10	CL11	OB98	49	0.0033	.902	.942	−6.8	143	4.1	0.2879	0.1479	8.3678	
9	CL13	OB24	45	0.0036	.898	.936	−6.2	155	4.5	0.2802	0.2005	3.5156	
8	CL10	OB25	50	0.0019	.896	.930	−5.2	175	2.2	0.2699	0.1372	8.3678	
7	CL8	OB121	51	0.0035	.893	.921	−4.2	198	4.0	0.2586	0.1372	8.3678	
6	CL9	OB45	46	0.0041	.888	.911	−3.0	229	4.7	0.1412	0.0832	3.5156	
5	CL6	OB39	47	0.0048	.884	.895	−1.5	276	5.1	0.107	0.0605	3.5156	
4	CL5	OB21	48	0.0048	.879	.872	0.54	353	4.7	0.0969	0.0541	3.5156	
3	CL4	OB90	49	0.0046	.874	.827	3.49	511	4.2	0.0715	0.0370	3.5156	
2	CL7	CL3	100	0.1017	.773	.697	3.83	503	96.3	2.6277	3.5156	8.3678	

3 modal clusters have been formed.

Output 29.3.6 Criteria for the Number of Clusters with METHOD=TWOSTAGE



Output 29.3.7 Crosstabulation of Clusters for METHOD=TWOSTAGE

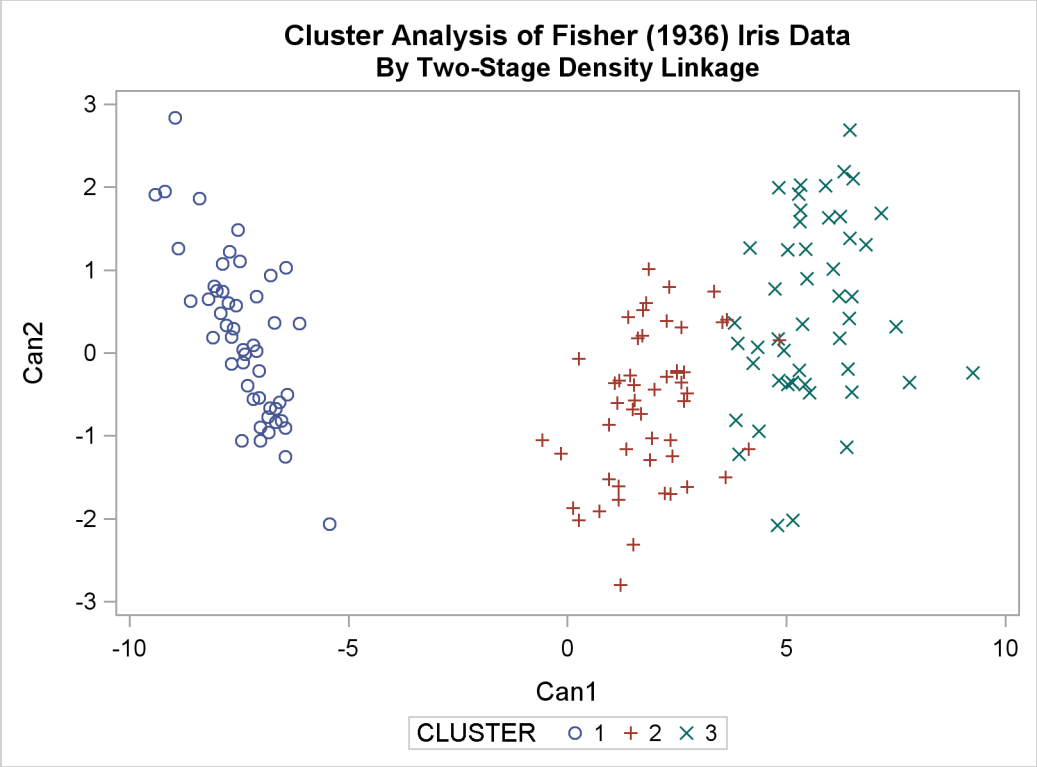
Cluster Analysis of Fisher (1936) Iris Data
By Two-Stage Density Linkage

The FREQ Procedure

Table of CLUSTER by Species

CLUSTER / Species	Setosa	Versicolor	Virginica	Total
1	50	0	0	50
2	0	48	3	51
3	0	2	47	49
Total	50	50	50	150

Output 29.3.8 Scatter Plot of Clusters for METHOD=TWOSTAGE



The CLUSTER procedure is not practical for very large data sets because, with most methods, the CPU time is roughly proportional to the square or cube of the number of observations. The FASTCLUS procedure requires time proportional to the number of observations and can therefore be used with much larger data sets than PROC CLUSTER. If you want to hierarchically cluster a very large data set, you can use PROC FASTCLUS for a preliminary cluster analysis to produce a large number of clusters and then use PROC CLUSTER to hierarchically cluster the preliminary clusters.

FASTCLUS automatically creates the variables _FREQ_ and _RMSSTD_ in the MEAN= output data set. These variables are then automatically used by PROC CLUSTER in the computation of various statistics.

The following SAS code uses the iris data to illustrate the process of clustering clusters. In the preliminary analysis, PROC FASTCLUS produces 10 clusters, which are then crosstabulated with species. The data set containing the preliminary clusters is sorted in preparation for later merges. The results are shown in Output 29.3.9 and Output 29.3.10.

```
title2 'Preliminary Analysis by FASTCLUS';
proc fastclus data=iris summary maxc=10 maxiter=99 converge=0
              mean=mean out=prelim cluster=preclus;
   var petal: sepal:;
run;
proc freq;
   tables preclus*species / nopercent norow nocol plot=none;
run;
proc sort data=prelim;
```

```
   by preclus;
run;
```

Output 29.3.9 Preliminary Analysis of Fisher's Iris Data: Fastclus Procedure

Cluster Analysis of Fisher (1936) Iris Data
Preliminary Analysis by FASTCLUS

The FASTCLUS Procedure
Replace=FULL Radius=0 Maxclusters=10 Maxiter=99 Converge=0

Convergence criterion is satisfied.

Criterion Based on Final Seeds = 2.1389

Cluster Summary

Cluster	Frequency	RMS Std Deviation	Maximum Distance from Seed to Observation	Radius Exceeded	Nearest Cluster
1	9	2.7067	8.2027		5
2	19	2.2001	7.7340		4
3	18	2.1496	6.2173		8
4	4	2.5249	5.3268		2
5	3	2.7234	5.8214		1
6	7	2.2939	5.1508		2
7	17	2.0274	6.9576		10
8	18	2.2628	7.1135		3
9	22	2.2666	7.5029		8
10	33	2.0594	10.0033		7

Cluster Summary

Cluster	Distance Between Cluster Centroids
1	8.7362
2	6.2243
3	7.5049
4	6.2243
5	8.7362
6	9.3318
7	7.9503
8	7.5049
9	9.0090
10	7.9503

Pseudo F Statistic = 370.58

Observed Over-All R-Squared = 0.95971

Approximate Expected Over-All R-Squared = 0.82928

Cubic Clustering Criterion = 27.077

WARNING: The two values above are invalid for correlated variables.

Output 29.3.10 Crosstabulation of Species and Cluster From the Fastclus Procedure

Cluster Analysis of Fisher (1936) Iris Data
Preliminary Analysis by FASTCLUS

The FREQ Procedure

Table of preclus by Species

preclus(Cluster)	Species			
Frequency	Setosa	Versicolor	Virginica	Total
1	0	0	9	9
2	0	19	0	19
3	0	18	0	18
4	0	3	1	4
5	0	0	3	3
6	0	7	0	7
7	17	0	0	17
8	0	3	15	18
9	0	0	22	22
10	33	0	0	33
Total	50	50	50	150

The following macro, CLUS, clusters the preliminary clusters. There is one argument to choose the METHOD= specification to be used by PROC CLUSTER. The TREE procedure creates an output data set containing the 3-cluster partition, which is sorted and merged with the OUT= data set from PROC FASTCLUS to determine which cluster each of the original 150 observations belongs to. The SHOW macro is then used to display the results. In this example, the CLUS macro is invoked using Ward's method, which produces 16 misclassifications, and Wong's hybrid method, which produces 22 misclassifications.

```
/*--- Define macro clus ---*/
%macro clus(method);
   proc cluster data=mean method=&method ccc pseudo;
      var petal: sepal:;
      copy preclus;
   run;
   proc tree noprint ncl=3 out=out;
      copy petal: sepal: preclus;
   run;
   proc sort data=out;
      by preclus;
   run;
   data clus;
      merge out prelim;
      by preclus;
   run;
   %show;
%mend;
```

The following statements produce Output 29.3.11 through Output 29.3.14.

```
title2 'Clustering Clusters by Ward''s Method';
%clus(ward);
```

Output 29.3.11 Clustering Clusters by Ward's Method

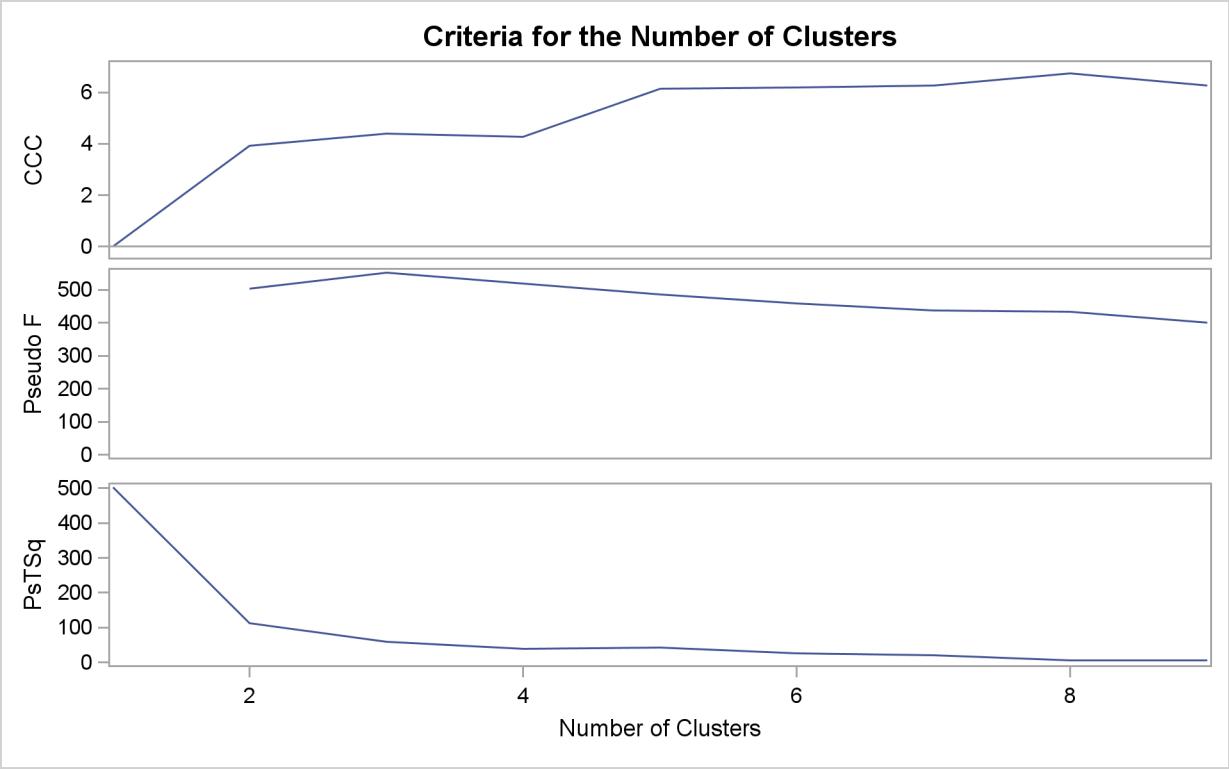
```
          Cluster Analysis of Fisher (1936) Iris Data
              Clustering Clusters by Ward's Method

                    The CLUSTER Procedure
              Ward's Minimum Variance Cluster Analysis

                 Eigenvalues of the Covariance Matrix

          Eigenvalue    Difference    Proportion    Cumulative

   1    416.976349    398.666421        0.9501        0.9501
   2     18.309928     14.952922        0.0417        0.9918
   3      3.357006      3.126943        0.0076        0.9995
   4      0.230063                      0.0005        1.0000

  Root-Mean-Square Total-Sample Standard Deviation    10.69224

   Root-Mean-Square Distance Between Observations    30.24221
```

Output 29.3.11 *continued*

Cluster History

NCL	--Clusters Joined---		FREQ	SPRSQ	RSQ	ERSQ	CCC	PSF	PST2	Tie
9	OB2	OB4	23	0.0019	.958	.932	6.26	400	6.3	
8	OB1	OB5	12	0.0025	.955	.926	6.75	434	5.8	
7	CL9	OB6	30	0.0069	.948	.918	6.28	438	19.5	
6	OB3	OB8	36	0.0074	.941	.907	6.21	459	26.0	
5	OB7	OB10	50	0.0104	.931	.892	6.15	485	42.2	
4	CL8	OB9	34	0.0162	.914	.870	4.28	519	39.3	
3	CL7	CL6	66	0.0318	.883	.824	4.39	552	59.7	
2	CL4	CL3	100	0.1099	.773	.695	3.94	503	113	
1	CL2	CL5	150	0.7726	.000	.000	0.00	.	503	

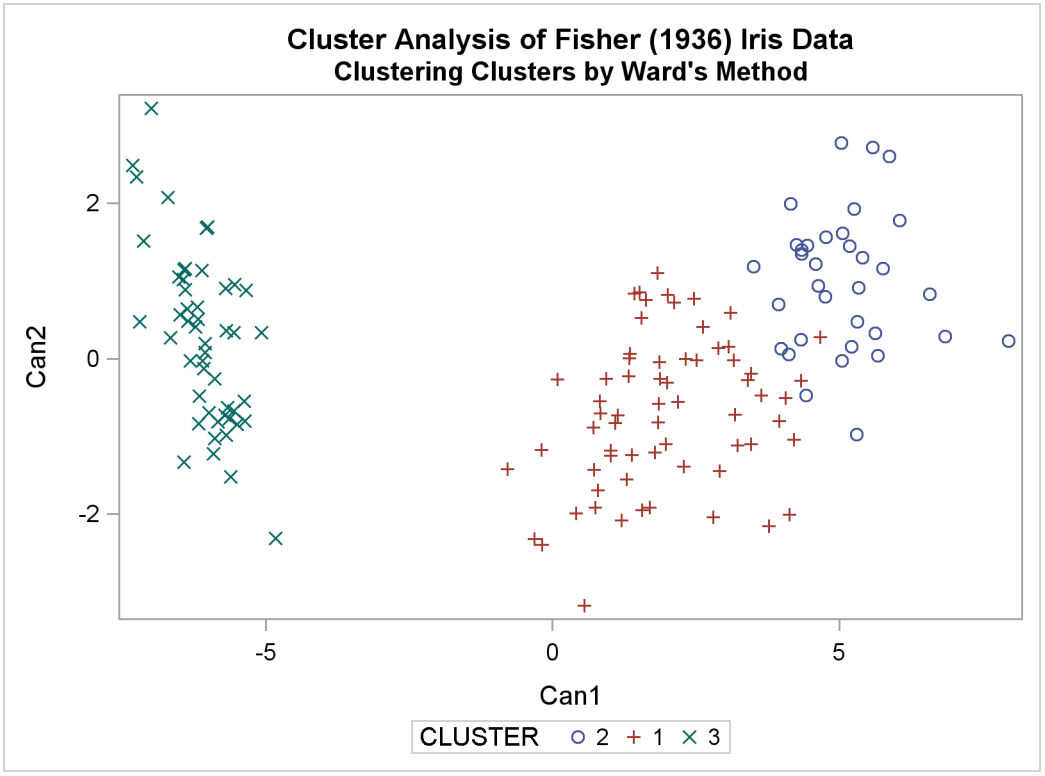
Output 29.3.12 Criteria for the Number of Clusters for Clustering Clusters from Ward's Method



Output 29.3.13 Crosstabulation for Clustering Clusters from Ward's Method

```
Cluster Analysis of Fisher (1936) Iris Data
    Clustering Clusters by Ward's Method

              The FREQ Procedure

         Table of CLUSTER by Species

CLUSTER     Species

Frequency|Setosa   |Versicol|Virginic|  Total
         |         |or      |a       |
---------+---------+--------+--------+
       1 |       0 |     50 |     16 |     66
---------+---------+--------+--------+
       2 |       0 |      0 |     34 |     34
---------+---------+--------+--------+
       3 |      50 |      0 |      0 |     50
---------+---------+--------+--------+
Total           50       50       50      150
```

Output 29.3.14 Scatter Plot for Clustering Clusters using Ward's Method



The following statements produce Output 29.3.15 through Output 29.3.17.

```
title2 "Clustering Clusters by Wong's Hybrid Method";
%clus(twostage hybrid);
```

Output 29.3.15 Clustering Clusters by Wong's Hybrid Method

Cluster Analysis of Fisher (1936) Iris Data
Clustering Clusters by Wong's Hybrid Method

The CLUSTER Procedure
Two-Stage Density Linkage Clustering

Eigenvalues of the Covariance Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	416.976349	398.666421	0.9501	0.9501
2	18.309928	14.952922	0.0417	0.9918
3	3.357006	3.126943	0.0076	0.9995
4	0.230063		0.0005	1.0000

Root-Mean-Square Total-Sample Standard Deviation 10.69224

Cluster History

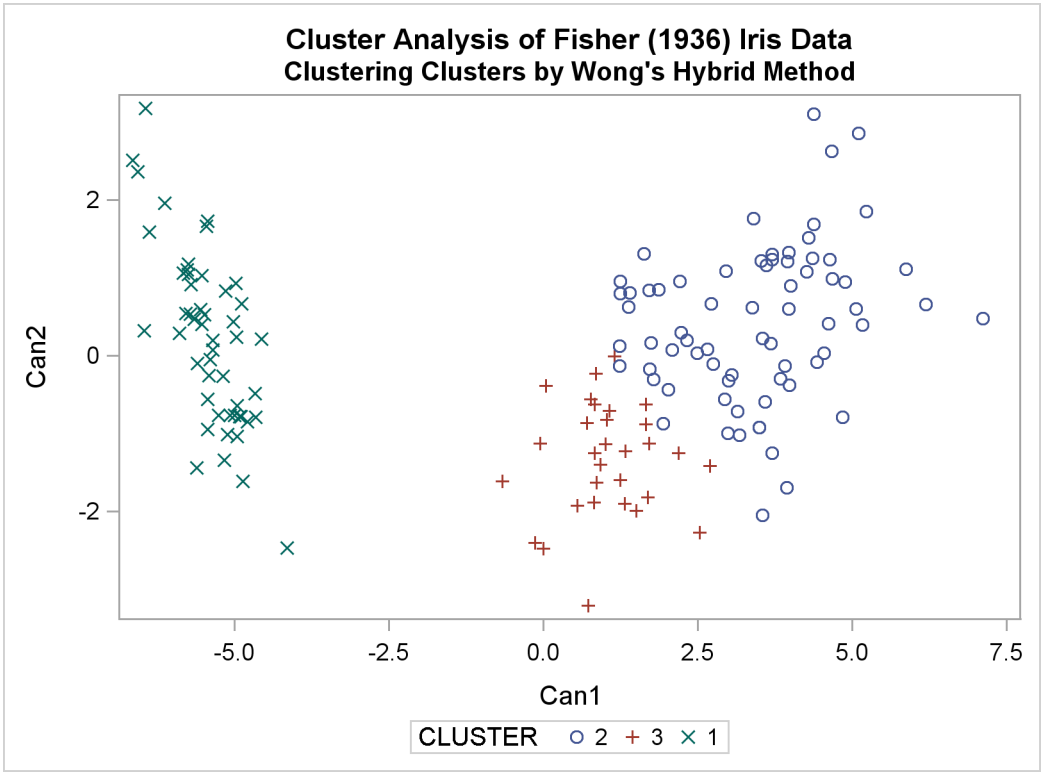
NCL	--Clusters Joined--		FREQ	SPRSQ	RSQ	ERSQ	CCC	PSF	PST2	Normalized Fusion Density	Maximum Density in Each Cluster Lesser	Greater	Tie
9	OB10	OB7	50	0.0104	.949	.932	3.81	330	42.2	40.24	58.2179	100.0	
8	OB3	OB8	36	0.0074	.942	.926	3.22	329	26.0	27.981	39.4511	48.4350	
7	OB2	OB4	23	0.0019	.940	.918	4.24	373	6.3	23.775	8.9675	46.3026	
6	CL8	OB9	58	0.0194	.921	.907	2.13	334	46.3	20.724	46.8846	48.4350	
5	CL7	OB6	30	0.0069	.914	.892	3.09	383	19.5	13.303	17.6360	46.3026	
4	CL6	OB1	67	0.0292	.884	.870	1.21	372	41.0	8.4137	10.8758	48.4350	
3	CL4	OB5	70	0.0138	.871	.824	3.33	494	12.3	5.1855	6.2890	48.4350	
2	CL3	CL5	100	0.0979	.773	.695	3.94	503	89.5	19.513	46.3026	48.4350	
1	CL2	CL9	150	0.7726	.000	.000	0.00	.	503	1.3337	48.4350	100.0	

3 modal clusters have been formed.

Output 29.3.16 Crosstabulation for Clustering Clusters from Wong's Hybrid Method

```
Cluster Analysis of Fisher (1936) Iris Data
Clustering Clusters by Wong's Hybrid Method

The FREQ Procedure

Table of CLUSTER by Species

CLUSTER     Species

Frequency|Setosa   |Versicol|Virginic|  Total
         |         |or      |a       |
---------+---------+--------+--------+
       1 |      50 |       0 |       0 |      50
---------+---------+--------+--------+
       2 |       0 |      21 |      49 |      70
---------+---------+--------+--------+
       3 |       0 |      29 |       1 |      30
---------+---------+--------+--------+
Total           50        50        50       150
```

Output 29.3.17 Scatter Plot for Clustering Clusters using Wong's Hybrid Method



Example 29.4: Evaluating the Effects of Ties

If, at some level of the cluster history, there is a tie for minimum distance between clusters, then one or more levels of the sample cluster tree are not uniquely determined. This example shows how the degree of indeterminacy can be assessed.

Mammals have four kinds of teeth: incisors, canines, premolars, and molars. The following data set gives the number of teeth of each kind on one side of the top and bottom jaws for 32 mammals.

Since all eight variables are measured in the same units, it is not strictly necessary to rescale the data. However, the canines have much less variance than the other kinds of teeth and, therefore, have little effect on the analysis if the variables are not standardized. An average linkage cluster analysis is run with and without standardization to enable comparison of the results.

```
title 'Hierarchical Cluster Analysis of Mammals'' Teeth Data';
title2 'Evaluating the Effects of Ties';
data teeth;
   input mammal $ 1-16
         @21 (v1-v8) (1.);
   label v1='Top incisors'
         v2='Bottom incisors'
         v3='Top canines'
         v4='Bottom canines'
         v5='Top premolars'
         v6='Bottom premolars'
         v7='Top molars'
         v8='Bottom molars';
   datalines;
BROWN BAT           23113333
MOLE                32103333
SILVER HAIR BAT     23112333
PIGMY BAT           23112233
HOUSE BAT           23111233
RED BAT             13112233
PIKA                21002233
RABBIT              21003233
BEAVER              11002133
GROUNDHOG           11002133
GRAY SQUIRREL       11001133
HOUSE MOUSE         11000033
PORCUPINE           11001133
WOLF                33114423
BEAR                33114423
RACCOON             33114432
MARTEN              33114412
WEASEL              33113312
WOLVERINE           33114412
BADGER              33113312
RIVER OTTER         33114312
SEA OTTER           32113312
JAGUAR              33113211
COUGAR              33113211
FUR SEAL            32114411
SEA LION            32114411
GREY SEAL           32113322
ELEPHANT SEAL       21114411
REINDEER            04103333
ELK                 04103333
DEER                04003333
MOOSE               04003333
;
```

The following statements produce Output 29.4.1:

```
title3 'Raw Data';
proc cluster data=teeth method=average nonorm noeigen;
   var v1-v8;
   id mammal;
run;
```

Output 29.4.1 Average Linkage Analysis of Mammals' Teeth Data: Raw Data

Hierarchical Cluster Analysis of Mammals' Teeth Data
Evaluating the Effects of Ties
Raw Data

The CLUSTER Procedure
Average Linkage Cluster Analysis

Root-Mean-Square Total-Sample Standard Deviation 0.898027

Cluster History

NCL	Clusters Joined		FREQ	RMS Dist	Tie
31	BEAVER	GROUNDHOG	2	0	T
30	GRAY SQUIRREL	PORCUPINE	2	0	T
29	WOLF	BEAR	2	0	T
28	MARTEN	WOLVERINE	2	0	T
27	WEASEL	BADGER	2	0	T
26	JAGUAR	COUGAR	2	0	T
25	FUR SEAL	SEA LION	2	0	T
24	REINDEER	ELK	2	0	T
23	DEER	MOOSE	2	0	
22	BROWN BAT	SILVER HAIR BAT	2	1	T
21	PIGMY BAT	HOUSE BAT	2	1	T
20	PIKA	RABBIT	2	1	T
19	CL31	CL30	4	1	T
18	CL28	RIVER OTTER	3	1	T
17	CL27	SEA OTTER	3	1	T
16	CL24	CL23	4	1	
15	CL21	RED BAT	3	1.2247	
14	CL17	GREY SEAL	4	1.291	
13	CL29	RACCOON	3	1.4142	T
12	CL25	ELEPHANT SEAL	3	1.4142	
11	CL18	CL14	7	1.5546	
10	CL22	CL15	5	1.5811	
9	CL20	CL19	6	1.8708	T
8	CL11	CL26	9	1.9272	
7	CL8	CL12	12	2.2278	
6	MOLE	CL13	4	2.2361	
5	CL9	HOUSE MOUSE	7	2.4833	
4	CL6	CL7	16	2.5658	
3	CL10	CL16	9	2.8107	
2	CL3	CL5	16	3.7054	
1	CL2	CL4	32	4.2939	

The following statements produce Output 29.4.2:

```
title3 'Standardized Data';
proc cluster data=teeth std method=average nonorm noeigen;
   var v1-v8;
   id mammal;
run;
```

Output 29.4.2 Average Linkage Analysis of Mammals' Teeth Data: Standardized Data

Hierarchical Cluster Analysis of Mammals' Teeth Data
Evaluating the Effects of Ties
Standardized Data

The CLUSTER Procedure
Average Linkage Cluster Analysis

The data have been standardized to mean 0 and variance 1

Root-Mean-Square Total-Sample Standard Deviation 1

Cluster History

NCL	Clusters Joined		FREQ	RMS Dist	Tie
31	BEAVER	GROUNDHOG	2	0	T
30	GRAY SQUIRREL	PORCUPINE	2	0	T
29	WOLF	BEAR	2	0	T
28	MARTEN	WOLVERINE	2	0	T
27	WEASEL	BADGER	2	0	T
26	JAGUAR	COUGAR	2	0	T
25	FUR SEAL	SEA LION	2	0	T
24	REINDEER	ELK	2	0	T
23	DEER	MOOSE	2	0	
22	PIGMY BAT	RED BAT	2	0.9157	
21	CL28	RIVER OTTER	3	0.9169	
20	CL31	CL30	4	0.9428	T
19	BROWN BAT	SILVER HAIR BAT	2	0.9428	T
18	PIKA	RABBIT	2	0.9428	
17	CL27	SEA OTTER	3	0.9847	
16	CL22	HOUSE BAT	3	1.1437	
15	CL21	CL17	6	1.3314	
14	CL25	ELEPHANT SEAL	3	1.3447	
13	CL19	CL16	5	1.4688	
12	CL15	GREY SEAL	7	1.6314	
11	CL29	RACCOON	3	1.692	
10	CL18	CL20	6	1.7357	
9	CL12	CL26	9	2.0285	
8	CL24	CL23	4	2.1891	
7	CL9	CL14	12	2.2674	
6	CL10	HOUSE MOUSE	7	2.317	
5	CL11	CL7	15	2.6484	
4	CL13	MOLE	6	2.8624	
3	CL4	CL8	10	3.5194	
2	CL3	CL6	17	4.1265	
1	CL2	CL5	32	4.7753	

There are ties at 16 levels for the raw data but at only 10 levels for the standardized data. There are more ties for the raw data because the increments between successive values are the same for all of the raw variables but different for the standardized variables.

One way to assess the importance of the ties in the analysis is to repeat the analysis on several random permutations of the observations and then to see to what extent the results are consistent at the interesting levels of the cluster history. Three macros are presented to facilitate this process, as follows.

```
/* ---------------------------------------------------------- */
/*                                                            */
/* The macro CLUSPERM randomly permutes observations and      */
/* does a cluster analysis for each permutation.              */
/* The arguments are as follows:                              */
/*                                                            */
/*    data    data set name                                   */
/*    var     list of variables to cluster                    */
/*    id      id variable for proc cluster                    */
/*    method  clustering method (and possibly other options)  */
/*    nperm   number of random permutations.                  */
/*                                                            */
/* ---------------------------------------------------------- */
%macro CLUSPERM(data,var,id,method,nperm);

   /* ------CREATE TEMPORARY DATA SET WITH RANDOM NUMBERS------ */
   data _temp_;
      set &data;
      array _random_ _ran_1-_ran_&nperm;
      do over _random_;
        _random_=ranuni(835297461);
      end;
   run;

   /* ------PERMUTE AND CLUSTER THE DATA----------------------- */
   %do n=1 %to &nperm;
       proc sort data=_temp_(keep=_ran_&n &var &id) out=_perm_;
          by _ran_&n;
       run;
       proc cluster method=&method noprint outtree=_tree_&n;
          var &var;
          id &id;
       run;
   %end;
%mend;
```

```
/* ---------------------------------------------------------- */
/*                                                            */
/* The macro PLOTPERM plots various cluster statistics        */
/* against the number of clusters for each permutation.       */
/* The arguments are as follows:                              */
/*                                                            */
/*    nclus   maximum number of clusters to be plotted        */
/*    nperm   number of random permutations.                  */
/*                                                            */
/* ---------------------------------------------------------- */
%macro PLOTPERM(nclus,nperm);

   /* ---CONCATENATE TREE DATA SETS FOR 20 OR FEWER CLUSTERS--- */
   data _plot_;
      set %do n=1 %to &nperm; _tree_&n(in=_in_&n) %end; ;
      if _ncl_<=&nclus;
      %do n=1 %to &nperm;
         if _in_&n then _perm_=&n;
      %end;
      label _perm_='permutation number';
      keep _ncl_ _psf_ _pst2_ _ccc_ _perm_;
   run;

   /* ---PLOT THE REQUESTED STATISTICS BY NUMBER OF CLUSTERS--- */
   proc sgscatter ;
      compare y=(_ccc_ _psf_ _pst2_) x=_ncl_ /group=_perm_ ;
   run;
%mend;

/* ---------------------------------------------------------- */
/*                                                            */
/* The macro TABPERM generates cluster-membership variables */
/* for a specified number of clusters for each permutation.  */
/* PROC TABULATE gives the frequencies and means.             */
/* The arguments are as follows:                              */
/*                                                            */
/*    var     list of variables to cluster                    */
/*            (no "-" or ":" allowed)                         */
/*    id      id variable for proc cluster                    */
/*    meanfmt format for printing means in PROC TABULATE      */
/*    nclus   number of clusters desired                      */
/*    nperm   number of random permutations.                  */
/*                                                            */
/* ---------------------------------------------------------- */
%macro TABPERM(var,id,meanfmt,nclus,nperm);

   /* ------CREATE DATA SETS GIVING CLUSTER MEMBERSHIP--------- */
   %do n=1 %to &nperm;
      proc tree data=_tree_&n noprint n=&nclus
                out=_out_&n(drop=clusname
                            rename=(cluster=_clus_&n));
         copy &var;
         id &id;
```

```
      run;
      proc sort;
         by &id &var;
      run;
   %end;

   /* ------MERGE THE CLUSTER VARIABLES---------------------- */
   data _merge_;
      merge
         %do n=1 %to &nperm;
            _out_&n
         %end; ;
      by &id &var;
      length all_clus $ %eval(3*&nperm);
      %do n=1 %to &nperm;
         substr( all_clus, %eval(1+(&n-1)*3), 3) =
            put( _clus_&n, 3.);
      %end;
   run;

   /* ------ TABULATE CLUSTER COMBINATIONS------------ */
   proc sort;
      by _clus_:;
   run;
   proc tabulate order=data formchar='           ';
      class all_clus;
      var &var;
      table all_clus, n='FREQ'*f=5. mean*f=&meanfmt*(&var) /
         rts=%eval(&nperm*3+1);
   run;
%mend;
```

To use these macros, it is first convenient to define a macro, VLIST, listing the teeth variables, since the forms V1-V8 or V: cannot be used with the TABULATE procedure in the TABPERM macro:

```
/* -TABULATE does not accept hyphens or colons in VAR lists- */
%let vlist=v1 v2 v3 v4 v5 v6 v7 v8;
```

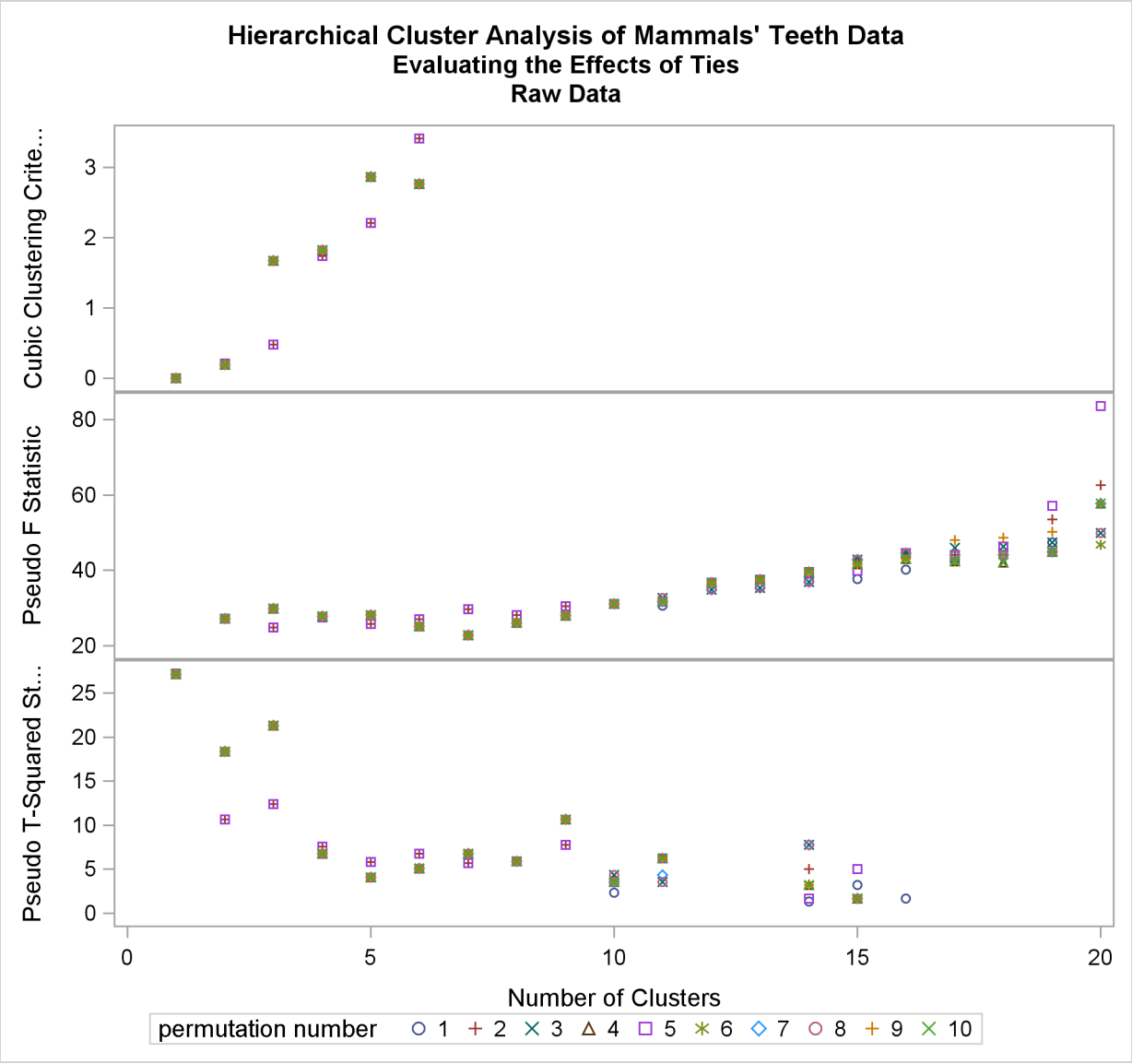
The CLUSPERM macro is then called to analyze 10 random permutations. The PLOTPERM macro plots the pseudo F and t^2 statistics and the cubic clustering criterion. Since the data are discrete, the pseudo F statistic and the cubic clustering criterion can be expected to increase as the number of clusters increases, so local maxima or large jumps in these statistics are more relevant than the global maximum in determining the number of clusters. For the raw data, only the pseudo t^2 statistic indicates the possible presence of clusters, with the 4-cluster level being suggested. Hence, the macros are used as follows to analyze the results at the 4-cluster level:

```
title3 'Raw Data';

/* ------CLUSTER RAW DATA WITH AVERAGE LINKAGE------------- */
%clusperm( teeth, &vlist, mammal, average, 10);
```

The following statements produce Output 29.4.3.

```
/* -----PLOT STATISTICS FOR THE LAST 20 LEVELS------------- */
%plotperm(20, 10);
```

Output 29.4.3 Analysis of 10 Random Permutations of Raw Mammals' Teeth Data



The following statements produce Output 29.4.4.

```
/* ------ANALYZE THE 4-CLUSTER LEVEL----------------------- */
%tabperm( &vlist, mammal, 9.1, 4, 10);
```

Output 29.4.4 Raw Mammals' Teeth Data: Indeterminacy at the 4-Cluster Level

Hierarchical Cluster Analysis of Mammals' Teeth Data
Evaluating the Effects of Ties
Raw Data

		Mean			
		Top	Bottom	Top	Bottom
	FREQ	incisors	incisors	canines	canines
all_clus					
1 3 1 1 1 3 3 3 2 3	4	0.0	4.0	0.5	0.0
2 2 2 2 2 2 1 2 1 1	15	2.9	2.6	1.0	1.0
2 4 2 2 4 2 1 2 1 1	1	3.0	2.0	1.0	0.0
3 1 3 3 3 1 2 1 3 2	5	1.0	1.0	0.0	0.0
3 4 3 3 4 1 2 1 3 2	2	2.0	1.0	0.0	0.0
4 4 4 4 4 4 4 4 4 4	5	1.8	3.0	1.0	1.0

(Continued)

Hierarchical Cluster Analysis of Mammals' Teeth Data
Evaluating the Effects of Ties
Raw Data

	Mean			
	Top	Bottom	Top	Bottom
	premolars	premolars	molars	molars
all_clus				
1 3 1 1 1 3 3 3 2 3	3.0	3.0	3.0	3.0
2 2 2 2 2 2 1 2 1 1	3.6	3.4	1.3	1.8
2 4 2 2 4 2 1 2 1 1	3.0	3.0	3.0	3.0
3 1 3 3 3 1 2 1 3 2	1.2	0.8	3.0	3.0
3 4 3 3 4 1 2 1 3 2	2.5	2.0	3.0	3.0
4 4 4 4 4 4 4 4 4 4	2.0	2.4	3.0	3.0

From the TABULATE output, you can see that two types of clustering are obtained. In one case, the mole is grouped with the carnivores, while the pika and rabbit are grouped with the rodents. In the other case, both the mole and the lagomorphs are grouped with the bats.

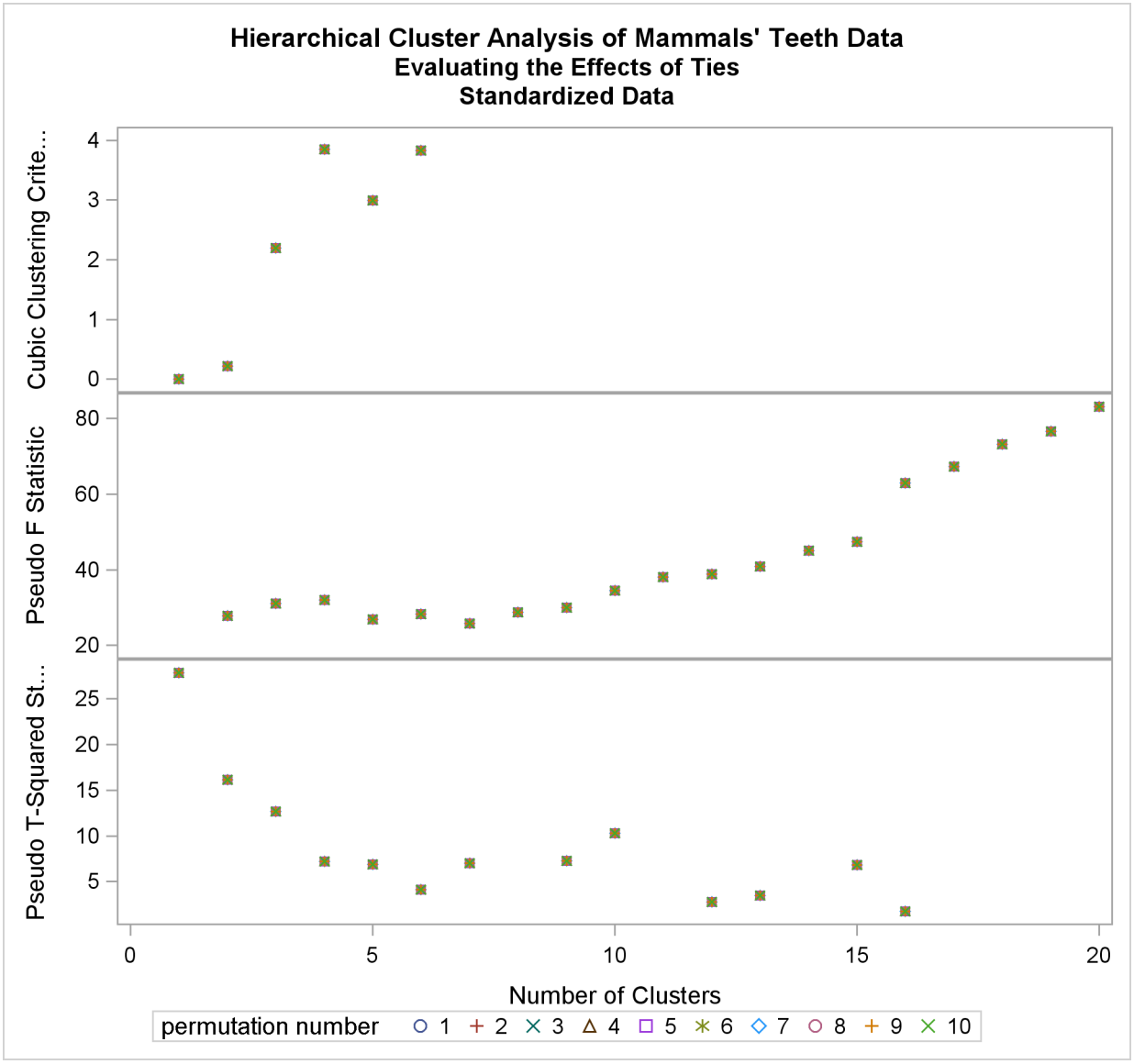
Next, the analysis is repeated with the standardized data as shown in the following statements. The pseudo F and t^2 statistics indicate 3 or 4 clusters, while the cubic clustering criterion shows a sharp rise up to 4 clusters and then levels off up to 6 clusters. So the TABPERM macro is used again at the 4-cluster level. In this case, there is no indeterminacy, because the same four clusters are obtained with every permutation, although in different orders. It must be emphasized, however, that lack of indeterminacy in no way indicates validity.

```
title3 'Standardized Data';

/*------CLUSTER STANDARDIZED DATA WITH AVERAGE LINKAGE------*/
%clusperm( teeth, &vlist, mammal, average std, 10);
```

The following statements produce Output 29.4.5.

```
/* -----PLOT STATISTICS FOR THE LAST 20 LEVELS------------- */
%plotperm(20, 10);
```

Output 29.4.5 Analysis of 10 Random Permutations of Standardized Mammals' Teeth Data



The following statements produce Output 29.4.6.

```
/* ------ANALYZE THE 4-CLUSTER LEVEL------------------------ */
%tabperm( &vlist, mammal, 9.1, 4, 10);
```

Output 29.4.6 Standardized Mammals' Teeth Data: No Indeterminacy at the 4-Cluster Level

Hierarchical Cluster Analysis of Mammals' Teeth Data
Evaluating the Effects of Ties
Standardized Data

all_clus	FREQ	Mean: Top incisors	Mean: Bottom incisors	Mean: Top canines	Mean: Bottom canines
1 3 1 1 1 3 3 3 2 3	4	0.0	4.0	0.5	0.0
2 2 2 2 2 2 1 2 1 1	15	2.9	2.6	1.0	1.0
3 1 3 3 3 1 2 1 3 2	7	1.3	1.0	0.0	0.0
4 4 4 4 4 4 4 4 4 4	6	2.0	2.8	1.0	0.8

(Continued)

Hierarchical Cluster Analysis of Mammals' Teeth Data
Evaluating the Effects of Ties
Standardized Data

all_clus	Mean: Top premolars	Mean: Bottom premolars	Mean: Top molars	Mean: Bottom molars
1 3 1 1 1 3 3 3 2 3	3.0	3.0	3.0	3.0
2 2 2 2 2 2 1 2 1 1	3.6	3.4	1.3	1.8
3 1 3 3 3 1 2 1 3 2	1.6	1.1	3.0	3.0
4 4 4 4 4 4 4 4 4 4	2.2	2.5	3.0	3.0

References

Anderberg, M. R. (1973), *Cluster Analysis for Applications*, New York: Academic Press.

Batagelj, V. (1981), "Note on Ultrametric Hierarchical Clustering Algorithms," *Psychometrika*, 46, 351–352.

Blackith, R. E. and Reyment, R. A. (1971), *Multivariate Morphometrics*, London: Academic Press.

Blashfield, R. K. and Aldenderfer, M. S. (1978), "The Literature on Cluster Analysis," *Multivariate Behavioral Research*, 13, 271–295.

Calinski, T. and Harabasz, J. (1974), "A Dendrite Method for Cluster Analysis," *Communications in Statistics*, 3, 1–27.

Cooper, M. C. and Milligan, G. W. (1988), "The Effect of Error on Determining the Number of Clusters," in *Data, Expert Knowledge, and Decisions*, 319–328, ed. W. Gaul and M. Schrader, London: Springer-Verlag.

Duda, R. O. and Hart, P. E. (1973), *Pattern Classification and Scene Analysis*, New York: John Wiley & Sons.

Everitt, B. S. (1980), *Cluster Analysis*, Second Edition, London: Heineman Educational Books.

Fisher, L. and Van Ness, J. W. (1971), "Admissible Clustering Procedures," *Biometrika*, 58, 91–104.

Fisher, R. A. (1936), "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, 7, 179–188.

Florek, K., Lukaszewicz, J., Perkal, J., and Zubrzycki, S. (1951a), "Sur la Liaison et la Division des Points d'un Ensemble Fini," *Colloquium Mathematicae*, 2, 282–285.

Florek, K., Lukaszewicz, J., Perkal, J., and Zubrzycki, S. (1951b), "Taksonomia Wroclawska," *Przeglad Antropol.*, 17, 193–211.

Gower, J. C. (1967), "A Comparison of Some Methods of Cluster Analysis," *Biometrics*, 23, 623–637.

Hamer, R. M. and Cunningham, J. W. (1981), "Cluster Analyzing Profile Data with Interrater Differences: A Comparison of Profile Association Measures," *Applied Psychological Measurement*, 5, 63–72.

Hartigan, J. A. (1975), *Clustering Algorithms*, New York: John Wiley & Sons.

Hartigan, J. A. (1977), "Distribution Problems in Clustering," in *Classification and Clustering*, ed. J. Van Ryzin, New York: Academic Press.

Hartigan, J. A. (1981), "Consistency of Single Linkage for High-Density Clusters," *Journal of the American Statistical Association*, 76, 388–394.

Hawkins, D. M., Muller, M. W., and ten Krooden, J. A. (1982), "Cluster Analysis," in *Topics in Applied Multivariate Analysis*, ed. D. M. Hawkins, Cambridge: Cambridge University Press.

Jardine, N. and Sibson, R. (1971), *Mathematical Taxonomy*, New York: John Wiley & Sons.

Johnson, S. C. (1967), "Hierarchical Clustering Schemes," *Psychometrika*, 32, 241–254.

Lance, G. N. and Williams, W. T. (1967), "A General Theory of Classificatory Sorting Strategies. I. Hierarchical Systems," *Computer Journal*, 9, 373–380.

Massart, D. L. and Kaufman, L. (1983), *The Interpretation of Analytical Chemical Data by the Use of Cluster Analysis*, New York: John Wiley & Sons.

McQuitty, L. L. (1957), "Elementary Linkage Analysis for Isolating Orthogonal and Oblique Types and Typal Relevancies," *Educational and Psychological Measurement*, 17, 207–229.

McQuitty, L. L. (1966), "Similarity Analysis by Reciprocal Pairs for Discrete and Continuous Data," *Educational and Psychological Measurement*, 26, 825–831.

Mezzich, J. E. and Solomon, H. (1980), *Taxonomy and Behavioral Science*, New York: Academic Press.

Milligan, G. W. (1979), "Ultrametric Hierarchical Clustering Algorithms," *Psychometrika*, 44, 343–346.

Milligan, G. W. (1980), "An Examination of the Effect of Six Types of Error Perturbation on Fifteen Clustering Algorithms," *Psychometrika*, 45, 325–342.

Milligan, G. W. (1987), "A Study of the Beta-Flexible Clustering Method," *College of Administrative Science Working Paper Series*, 87–61 Columbus: Ohio State University.

Milligan, G. W. and Cooper, M. C. (1985), "An Examination of Procedures for Determining the Number of Clusters in a Data Set," *Psychometrika*, 50,159–179.

Milligan, G. W. and Cooper, M. C. (1987), "A Study of Variable Standardization," *College of Administrative Science Working Paper Series*, 87–63, Columbus: Ohio State University.

Rouncefield, M. (1995), "The Statistics of Poverty and Inequality," *Journal of Statistics Education*, 3(2). [Online]: [http://www.stat.ncsu.edu/info/jse], accessed Dec. 19, 1997.

Sarle, W. S. (1983), *Cubic Clustering Criterion*, SAS Technical Report A-108, Cary, NC: SAS Institute Inc.

Silverman, B. W. (1986), *Density Estimation*, New York: Chapman & Hall.

Sneath, P. H. A. (1957), "The Application of Computers to Taxonomy," *Journal of General Microbiology*, 17, 201–226.

Sneath, P. H. A. and Sokal, R. R. (1973), *Numerical Taxonomy*, San Francisco: Freeman.

Sokal, R. R. and Michener, C. D. (1958), "A Statistical Method for Evaluating Systematic Relationships," *University of Kansas Science Bulletin*, 38, 1409–1438.

Sorensen, T. (1948), "A Method of Establishing Groups of Equal Amplitude in Plant Sociology Based on Similarity of Species Content and Its Application to Analyses of the Vegetation on Danish Commons," *Biologiske Skrifter*, 5, 1–34.

Spath, H. (1980), *Cluster Analysis Algorithms*, Chichester, Eng.: Ellis Horwood.

Symons, M. J. (1981), "Clustering Criteria and Multivariate Normal Mixtures," *Biometrics*, 37, 35–43.

Ward, J. H. (1963), "Hierarchical Grouping to Optimize an Objective Function," *Journal of the American Statistical Association*, 58, 236–244.

Wishart, D. (1969), "Mode Analysis: A Generalisation of Nearest Neighbour Which Reduces Chaining Effects," in *Numerical Taxonomy*, ed. A. J. Cole, London: Academic Press.

Wong, M. A. (1982), "A Hybrid Clustering Method for Identifying High-Density Clusters," *Journal of the American Statistical Association*, 77, 841–847.

Wong, M. A. and Lane, T. (1983), "A kth Nearest Neighbor Clustering Procedure," *Journal of the Royal Statistical Society*, Series B, 45, 362–368.

Wong, M. A. and Schaack, C. (1982), "Using the kth Nearest Neighbor Clustering Procedure to Determine the Number of Subpopulations," *American Statistical Association 1982 Proceedings of the Statistical Computing Section*, 40–48.

Chapter 30

The CORRESP Procedure

Contents

Overview: CORRESP Procedure **1320**
Background 1320
Getting Started: CORRESP Procedure **1320**
Syntax: CORRESP Procedure **1323**
PROC CORRESP Statement 1323
BY Statement 1331
ID Statement 1331
SUPPLEMENTARY Statement 1331
TABLES Statement 1332
VAR Statement 1332
WEIGHT Statement 1333
Details: CORRESP Procedure **1334**
Input Data Set 1334
Using the TABLES Statement 1334
Using the VAR Statement 1342
Missing and Invalid Data 1343
Coding, Fuzzy Coding, and Doubling 1343
Creating a Data Set Containing the Crosstabulation 1346
Output Data Sets 1348
Computational Resources 1350
Algorithm and Notation 1351
Displayed Output 1358
ODS Table Names 1361
ODS Graphics 1363
Examples: CORRESP Procedure **1363**
Example 30.1: Simple and Multiple Correspondence Analysis of Automobiles and Their Owners 1363
Example 30.2: Simple Correspondence Analysis of U.S. Population 1375
References **1381**

Overview: CORRESP Procedure

The CORRESP procedure performs simple correspondence analysis and multiple correspondence analysis (MCA). You can use correspondence analysis to find a low-dimensional graphical representation of the rows and columns of a crosstabulation or contingency table. Each row and column is represented by a point in a plot determined from the cell frequencies. PROC CORRESP can also compute coordinates for supplementary rows and columns.

PROC CORRESP can read two kinds of input: raw categorical responses on two or more classification variables or a two-way contingency table. The correspondence analysis plot is displayed with ODS Graphics. For more information about ODS Graphics, see the section "ODS Graphics" on page 1363.

Background

Correspondence analysis is a popular data analysis method in France and Japan. In France, correspondence analysis was developed under the influence of Jean-Paul Benzécri; in Japan, it was developed under Chikio Hayashi. The name *correspondence analysis* is a translation of the French *analyse des correspondances*. The technique apparently has many independent beginnings (for example, Richardson and Kuder 1933; Hirshfield 1935; Horst 1935; Fisher 1940; Guttman 1941; Burt 1950; Hayashi 1950). It has had many other names, including optimal scaling, reciprocal averaging, optimal scoring, and appropriate scoring in the United States; quantification method in Japan; homogeneity analysis in the Netherlands; dual scaling in Canada; and scalogram analysis in Israel.

Correspondence analysis is described in more detail in French in Benzécri (1973) and Lebart, Morineau, and Tabard (1977). In Japanese, the subject is described in Komazawa (1982), Nishisato (1982), and Kobayashi (1981). In English, correspondence analysis is described in Lebart, Morineau, and Warwick (1984), Greenacre (1984), Nishisato (1980), Tenenhaus and Young (1985), Gifi (1990), Greenacre and Hastie (1987), and many other sources. Hoffman and Franke (1986) offer a short, introductory treatment that uses examples from the field of market research.

Getting Started: CORRESP Procedure

Data are available containing the numbers of Ph.D.'s awarded in the United States during the years 1973 through 1978 (U.S. Bureau of the Census 1979). The table has six rows, one for each of six academic disciplines, and six columns for the six years. The following DATA step reads the complete table into a SAS data set, and PROC CORRESP displays correspondence analysis results including the inertia decomposition and coordinates. (See Figure 30.1 and Figure 30.2.) The concept of *inertia* in correspondence analysis is analogous to the concept of variance in principal component analysis, and it is proportional to the chi-square information.

```
title "Number of Ph.D.'s Awarded from 1973 to 1978";

data PhD;
   input Science $ 1-19 y1973-y1978;
   label y1973 = '1973'
         y1974 = '1974'
         y1975 = '1975'
         y1976 = '1976'
         y1977 = '1977'
         y1978 = '1978';
   datalines;
Life Sciences       4489 4303 4402 4350 4266 4361
Physical Sciences   4101 3800 3749 3572 3410 3234
Social Sciences     3354 3286 3344 3278 3137 3008
Behavioral Sciences 2444 2587 2749 2878 2960 3049
Engineering         3338 3144 2959 2791 2641 2432
Mathematics         1222 1196 1149 1003  959  959
;

ods graphics on;

proc corresp data=PhD out=Results short;
   var y1973-y1978;
   id Science;
run;

ods graphics off;
```

Figure 30.1 Inertia and Chi-Square Decomposition

Number of Ph.D.'s Awarded from 1973 to 1978

The CORRESP Procedure

Inertia and Chi-Square Decomposition

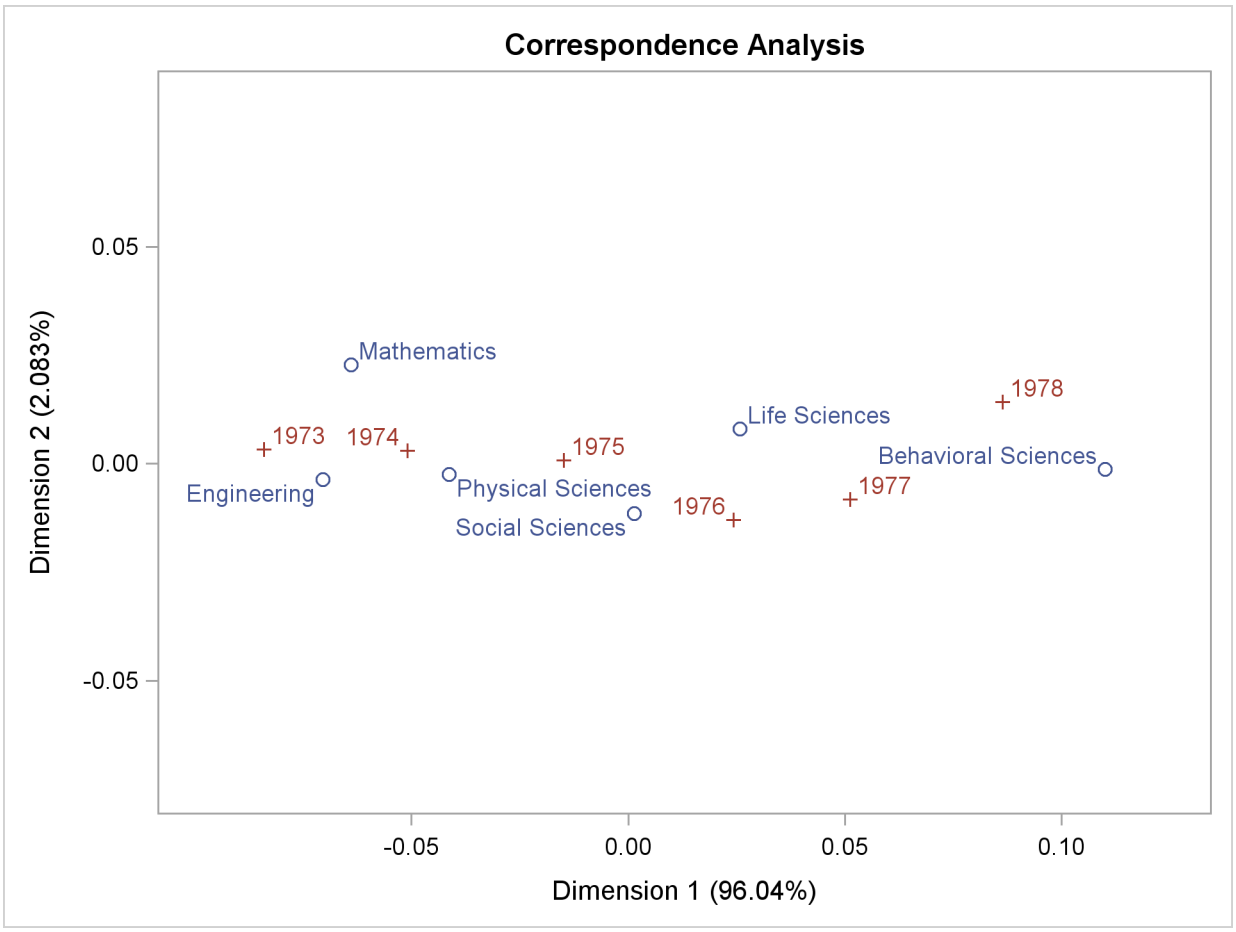
Singular Value	Principal Inertia	Chi-Square	Percent	Cumulative Percent	19 38 57 76 95
0.05845	0.00342	368.653	96.04	96.04	**********
0.00861	0.00007	7.995	2.08	98.12	*
0.00694	0.00005	5.197	1.35	99.48	
0.00414	0.00002	1.852	0.48	99.96	
0.00122	0.00000	0.160	0.04	100.00	
Total	0.00356	383.856	100.00		

Degrees of Freedom = 25

The total chi-square statistic in Figure 30.1, which is a measure of the association between the rows and columns in the full five dimensions of the (centered) table, is 383.856. The maximum number of dimensions (or axes) is the minimum of the number of rows and columns, minus one.

More than 96% of the total chi-square and inertia is explained by the first dimension, indicating that the association between the row and column categories is essentially one-dimensional. The plot in Figure 30.2 shows how the number of doctorates in the different disciplines changes over time. The plot shows that the number of doctorates in the behavioral sciences is associated with later years, and the number of doctorates in mathematics and engineering is associated with earlier years. This is consistent with the data that show that the number of doctorates in the behavioral sciences is increasing, the number of doctorates in every other discipline is decreasing, and the rate of decrease is greatest for mathematics and engineering.

Figure 30.2 Correspondence Analysis of Ph.D. Data



Syntax: CORRESP Procedure

The following statements are available in the CORRESP procedure.

PROC CORRESP < *options* > ;
TABLES < *row-variables,* > *column-variables* ;
VAR *variables* ;
BY *variables* ;
ID *variable* ;
SUPPLEMENTARY *variables* ;
WEIGHT *variable* ;

There are two separate forms of input to PROC CORRESP. One form is specified in the TABLES statement, the other in the VAR statement. You must specify either the TABLES or the VAR statement, but not both, each time you run PROC CORRESP.

Specify the TABLES statement if you are using raw, categorical data, the levels of which define the rows and columns of a table.

Specify the VAR statement if your data are already in tabular form. PROC CORRESP is generally more efficient with VAR statement input than with TABLES statement input.

The other statements are optional. All of the statements are explained, in alphabetical order, following the PROC CORRESP statement. All of the options in PROC CORRESP can be abbreviated to their first three letters, except for the OUTF= option. This is a special feature of PROC CORRESP and is not generally true of SAS/STAT procedures.

PROC CORRESP Statement

PROC CORRESP < *options* > ;

The PROC CORRESP statement invokes the CORRESP procedure. The options listed in Table 30.1 are available in the PROC CORRESP statement. These options are described following the table.

Table 30.1 Summary of PROC CORRESP Statement Options

Option	Description
Data Set Options	
DATA=	specifies input SAS data set
OUTC=	specifies output coordinate SAS data set
OUTF=	specifies output frequency SAS data set
Row and Column Coordinates	
DIMENS=	specifies the number of dimensions or axes
MCA	performs multiple correspondence analysis
PROFILE=	standardizes the row and column coordinates

Table 30.1 *continued*

Option	Description
Table Construction	
BINARY	specifies binary table
CROSS=	specifies cross levels of TABLES variables
FREQOUT	specifies input data in PROC FREQ output
MISSING	includes observations with missing values
Control Displayed Output	
ALL	displays all output
BENZECRI	displays inertias adjusted by Benzécri's method
CELLCHI2	displays cell contributions to chi-square
CP	displays column profile matrix
DEVIATION	displays observed minus expected values
EXPECTED	displays chi-square expected values
GREENACRE	displays inertias adjusted by Greenacre's method
NOCOLUMN=	suppresses the display of column coordinates
NOPRINT	suppresses the display of all output
NOROW=	suppresses the display of row coordinates
OBSERVED	displays contingency table of observed frequencies
PLOTS=	specifies ODS Graphics details
PRINT=	displays percentages or frequencies
RP	displays row profile matrix
SHORT	suppresses all point and coordinate statistics
UNADJUSTED	displays unadjusted inertias
Other Options	
COLUMN=	specifies esoteric column coordinate standardizations
MININERTIA=	specifies minimum inertia
NVARS=	specifies number of classification variables
ROW=	specifies esoteric row coordinate standardizations
SINGULAR=	specifies effective zero
SOURCE	includes level source in the OUTC= data set

The display options control the amount of displayed output. The CELLCHI2, EXPECTED, and DEVIATION options display additional chi-square information. See the "Details: CORRESP Procedure" section for more information. The unit of the matrices displayed by the CELLCHI2, CP, DEVIATION, EXPECTED, OBSERVED, and RP options depends on the value of the PRINT= option. The table construction options control the construction of the contingency table; these options are valid only when you also specify a TABLES statement.

You can specify the following options in the PROC CORRESP statement. They are given in alphabetical order.

ALL

is equivalent to specifying the OBSERVED, RP, CP, CELLCHI2, EXPECTED, and DEVIATION options. Specifying the ALL option does not affect the PRINT= option. Therefore,

only frequencies (not percentages) for these options are displayed unless you specify otherwise with the PRINT= option.

BENZECRI
BEN

displays adjusted inertias when performing multiple correspondence analysis. By default, unadjusted inertias, the usual inertias from multiple correspondence analysis, are displayed. However, adjusted inertias that use a method proposed by Benzécri (1979) and described by Greenacre (1984, p. 145) can be displayed by specifying the BENZECRI option. Specify the UNADJUSTED option to output the usual table of unadjusted inertias as well. See the section "MCA Adjusted Inertias" on page 1355 for more information.

BINARY

enables you to create binary tables easily. When you specify the BINARY option, specify only column variables in the TABLES statement. Each input data set observation forms a single row in the constructed table.

CELLCHI2
CEL

displays the contribution to the total chi-square test statistic for each cell. See also the descriptions of the DEVIATION, EXPECTED, and OBSERVED options.

COLUMN=B | BD | DB | DBD | DBD1/2 | DBID1/2
COL=B | BD | DB | DBD | DBD1/2 | DBID1/2

provides other standardizations of the column coordinates. The COLUMN= option is rarely needed. Typically, you should use the PROFILE= option instead (see the section "The PROFILE=, ROW=, and COLUMN= Options" on page 1352). By default, COLUMN=DBD.

CP

displays the column profile matrix. Column profiles contain the observed conditional probabilities of row membership given column membership. See also the RP option.

CROSS=BOTH | COLUMN | NONE | ROW
CRO=BOT | COL | NON | ROW

specifies the method of crossing (factorially combining) the levels of the TABLES variables. The default is CROSS=NONE.

NONE	causes each level of every row variable to become a row label and each level of every column variable to become a column label.
ROW	causes each combination of levels for all row variables to become a row label, whereas each level of every column variable becomes a column label.
COLUMN	causes each combination of levels for all column variables to become a column label, whereas each level of every row variable becomes a row label.
BOTH	causes each combination of levels for all row variables to become a row label and each combination of levels for all column variables to become a column label.

The section "TABLES Statement" on page 1332 provides a more detailed description of this option.

DATA=*SAS-data-set*

specifies the SAS data set to be used by PROC CORRESP. If you do not specify the DATA= option, PROC CORRESP uses the most recently created SAS data set.

DEVIATION

DEV

displays the matrix of deviations between the observed frequency matrix and the product of its row marginals and column marginals divided by its grand frequency. For ordinary two-way contingency tables, these are the observed minus expected frequencies under the hypothesis of row and column independence and are components of the chi-square test statistic. See also the CELLCHI2, EXPECTED, and OBSERVED options.

DIMENS=*n*

DIM=*n*

specifies the number of dimensions or axes to use. The default is DIMENS=2. The maximum value of the DIMENS= option in an $(n_r \times n_c)$ table is $n_r - 1$ or $n_c - 1$, whichever is smaller. For example, in a table with 4 rows and 5 columns, the maximum specification is DIMENS=3. If your table has 2 rows or 2 columns, specify DIMENS=1.

EXPECTED

EXP

displays the product of the row marginals and the column marginals divided by the grand frequency of the observed frequency table. For ordinary two-way contingency tables, these are the expected frequencies under the hypothesis of row and column independence and are components of the chi-square test statistic. In other situations, this interpretation is not strictly valid. See also the CELLCHI2, DEVIATION, and OBSERVED options.

FREQOUT

FRE

indicates that the PROC CORRESP input data set has the same form as an output data set from the FREQ procedure, even if it was not directly produced by PROC FREQ. The FREQOUT option enables PROC CORRESP to take shortcuts in constructing the contingency table.

When you specify the FREQOUT option, you must also specify a WEIGHT statement. The cell frequencies in a PROC FREQ output data set are contained in a variable called COUNT, so specify COUNT in a WEIGHT statement with PROC CORRESP. The FREQOUT option might produce unexpected results if the DATA= data set is structured incorrectly. Each of the two variable lists specified in the TABLES statement must consist of a single variable, and observations must be grouped by the levels of the row variable and then by the levels of the column variable. It is not required that the observations be sorted by the row variable and column variable, but they must be grouped consistently. There must be as many observations in the input data set (or BY group) as there are cells in the completed contingency table. Zero cells must be specified with zero weights. When you use PROC FREQ to create the PROC CORRESP input data set, you must specify the SPARSE option in the FREQ procedure's TABLES statement so that the zero cells are written to the output data set.

GREENACRE

GRE

displays adjusted inertias when you are performing multiple correspondence analysis. By default, unadjusted inertias, the usual inertias from multiple correspondence analysis, are displayed. However, adjusted inertias that use a method proposed by Greenacre (1984, p. 156) can be displayed by specifying the GREENACRE option. Specify the UNADJUSTED option to output the usual table of unadjusted inertias as well. See the section "MCA Adjusted Inertias" on page 1355 for more information.

MCA

requests a multiple correspondence analysis. This option requires that the input table be a Burt table, which is a symmetric matrix of crosstabulations among several categorical variables. If you specify the MCA option and a VAR statement, you must also specify the NVARS= option, which gives the number of categorical variables that were used to create the table. With raw categorical data, if you want results for the individuals as well as the categories, use the BINARY option instead.

MININERTIA=*n*

MIN=*n*

specifies the minimum inertia ($0 \leq n \leq 1$) used to create the "best" tables—the indicator of which points best explain the inertia of each dimension. By default, MININERTIA=0.8. See the section "Algorithm and Notation" on page 1351 for more information.

MISSING

MIS

specifies that observations with missing values for the TABLES statement variables are included in the analysis. Missing values are treated as a distinct level of each categorical variable. By default, observations with missing values are excluded from the analysis.

NOCOLUMN < = BOTH | DATA | PRINT >

NOC < = BOT | DAT | PRI >

suppresses the display of the column coordinates and statistics and omits them from the output coordinate data set.

BOTH	suppresses all column information from both the SAS listing and the output data set. The NOCOLUMN option is equivalent to the option NOCOLUMN=BOTH.
DATA	suppresses all column information from the output data set.
PRINT	suppresses all column information from the SAS listing.

NOPRINT

NOP

suppresses the display of all output. This option is useful when you need only an output data set. This option disables the Output Delivery System (ODS), including ODS Graphics, for the duration of the PROC. For more information, see Chapter 20, "Using the Output Delivery System."

NOROW < = BOTH | DATA | PRINT >

NOR < = BOT | DAT | PRI >

suppresses the display of the row coordinates and statistics and omits them from the output coordinate data set.

BOTH	suppresses all row information from both the SAS listing and the output data set. The NOROW option is equivalent to the option NOROW=BOTH.
DATA	suppresses all row information from the output data set.
PRINT	suppresses all row information from the SAS listing.

The NOROW option can be useful when the rows of the contingency table are replications.

NVARS=*n*

NVA=*n*

specifies the number of classification variables that were used to create the Burt table. For example, suppose the Burt table was originally created with the following statement:

```
tables a b c;
```

You must specify NVARS=3 to read the table with a VAR statement.

The NVARS= option is required when you specify both the MCA option and a VAR statement. (See the section "VAR Statement" on page 1332 for an example.)

OBSERVED

OBS

displays the contingency table of observed frequencies and its row, column, and grand totals. If you do not specify the OBSERVED or ALL option, the contingency table is not displayed.

OUTC=*SAS-data-set*

OUT=*SAS-data-set*

creates an output coordinate SAS data set to contain the row, column, supplementary observation, and supplementary variable coordinates. This data set also contains the masses, squared cosines, quality of each point's representation in the DIMENS=*n* dimensional display, relative inertias, partial contributions to inertia, and best indicators.

OUTF=*SAS-data-set*

creates an output frequency SAS data set to contain the contingency table, row, and column profiles, the expected values, and the observed minus expected values and contributions to the chi-square statistic.

PLOTS< (*global-plot-options***) > < = ***plot-request* < *(options)* > >

PLOTS< (*global-plot-options***) > < = (***plot-request* < *(options)* > **< ...** *plot-request* < *(options)* > **>) >**

specifies options that control the details of the plots. When you specify only one plot request, you can omit the parentheses around the plot request.

The global plot option is as follows:

FLIP
FLI
: flips or interchanges the X-axis and Y-axis dimensions.

The plot requests include the following:

ALL
: produces all appropriate plots.

NONE
NON
: suppresses all plots.

By default, for simple correspondence analysis, PROC CORRESP prints the configuration of points consisting of the row coordinates and column coordinates. With MCA, only column coordinates are printed. The default plots ($y * x$) are Dim2 * Dim1, Dim3 * Dim1, Dim3 * Dim2, and so on. When you specify PLOTS(FLIP), the plots are Dim1 * Dim2, Dim1 * Dim3, Dim2 * Dim3, and so on.

You must enable ODS Graphics before requesting plots, for example, like this:

```
ods graphics on;

proc corresp;
   tables Marital, Origin;
run;

ods graphics off;
```

PRINT=BOTH | FREQ | PERCENT
PRI=BOT | FRE | PER
: affects the OBSERVED, RP, CP, CELLCHI2, EXPECTED, and DEVIATION options. The default is PRINT=FREQ.

- The PRINT=FREQ option displays output in the appropriate raw or natural units. (That is, PROC CORRESP displays raw frequencies for the OBSERVED option, relative frequencies with row marginals of 1.0 for the RP option, and so on.)
- The PRINT=PERCENT option scales results to percentages for the display of the output. (All elements in the OBSERVED matrix sum to 100.0, the row marginals are 100.0 for the RP option, and so on.)
- The PRINT=BOTH option displays both percentages and frequencies.

PROFILE=BOTH | COLUMN | NONE | ROW
PRO=BOT | COL | NON | ROW
: specifies the standardization for the row and column coordinates. The default is PROFILE=BOTH.

BOTH	specifies a standard correspondence analysis, which jointly displays the principal row and column coordinates. Row coordinates are computed from the row profile matrix, and column coordinates are computed from the column profile matrix.
ROW	specifies a correspondence analysis of the row profile matrix. The row coordinates are weighted centroids of the column coordinates.
COLUMN	specifies a correspondence analysis of the column profile matrix. The column coordinates are weighted centroids of the row coordinates.
NONE	is rarely needed. Row and column coordinates are the generalized singular vectors, without the customary standardizations.

ROW=A | AD | DA | DAD | DAD1/2 | DAID1/2

provides other standardizations of the row coordinates. The ROW= option is rarely needed. Typically, you should use the PROFILE= option instead (see the section "The PROFILE=, ROW=, and COLUMN= Options" on page 1352). By default, ROW=DAD.

RP

displays the row profile matrix. Row profiles contain the observed conditional probabilities of column membership given row membership. See also the CP option.

SHORT

SHO

suppresses the display of all point and coordinate statistics except the coordinates. The following information is suppressed: each point's mass, relative contribution to the total inertia, and quality of representation in the DIMENS=*n* dimensional display; the squared cosines of the angles between each axis and a vector from the origin to the point; the partial contributions of each point to the inertia of each dimension; and the best indicators.

SINGULAR=*n*

SIN=*n*

specifies the largest value that is considered to be within rounding error of zero. The default value is 1E–8. This parameter is used in checking for zero rows and columns, in checking Burt table diagonal sums for equality, in checking denominators before dividing, and so on. Typically, you should not assign a value outside the range 1E–6 to 1E–12.

SOURCE

SOU

adds the variable _VAR_, which contains the name or label of the variable corresponding to the current level, to the OUTC= and OUTF= data sets.

UNADJUSTED

UNA

displays unadjusted inertias when performing multiple correspondence analysis. By default, unadjusted inertias, the usual inertias from multiple correspondence analysis, are displayed. However, if adjusted inertias are requested by either the GREENACRE option or the BENZECRI option, then the unadjusted inertia table is not displayed unless the UNADJUSTED option is specified. See the section "MCA Adjusted Inertias" on page 1355 for more information.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC CORRESP to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC CORRESP. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

ID Statement

ID *variable* **;**

You specify the ID statement only in conjunction with the VAR statement. You cannot specify the ID statement when you use the TABLES statement or the MCA option. When you specify an ID variable, PROC CORRESP labels the rows of the tables with the ID values and places the ID variable in the output data set.

SUPPLEMENTARY Statement

SUPPLEMENTARY *variables* **;**

SUP *variables* **;**

The SUPPLEMENTARY statement specifies variables that are to be represented as points in the joint row and column space but that are not used in determining the locations of the other, active row and column points of the contingency table. Supplementary observations on supplementary variables are ignored in simple correspondence analysis but are needed to compute the squared cosines for multiple correspondence analysis. Variables that are specified in the SUPPLEMENTARY statement must also be specified in the TABLES or VAR statement.

When you specify a VAR statement, each SUPPLEMENTARY variable indicates one supplementary column of the table. Supplementary variables must be numeric with VAR statement input.

When you specify a TABLES statement, each SUPPLEMENTARY variable indicates a set of rows or columns of the table that is supplementary. Supplementary variables can be either character or numeric with TABLES statement input.

TABLES Statement

TABLES < *row-variables,* > *column-variables* **;**

The TABLES statement instructs PROC CORRESP to create a contingency table, Burt table, or binary table from the values of two or more categorical variables. The TABLES statement specifies classification variables that are used to construct the rows and columns of the contingency table. The variables can be either numeric or character. The variable lists in the TABLES statement and the CROSS= option together determine the row and column labels of the contingency table.

You can specify both row variables and column variables separated by a comma, or you can specify only column variables and no comma. If you do not specify row variables (that is, if you list variables but do not use the comma as a delimiter), then you should specify either the MCA or the BINARY option. With the MCA option, PROC CORRESP creates a Burt table, which is a crosstabulation of each variable with itself and every other variable. The Burt table is symmetric. With the BINARY option, PROC CORRESP creates a binary table, which consists of one row for each input data set observation and one column for each category of each TABLES statement variable. If the binary matrix is $\mathbf{Z}$, then the Burt table is $\mathbf{Z}'\mathbf{Z}$. Specifying the BINARY option with the NOROWS option produces the same results as specifying the MCA option (except for the chi-square statistics).

See Figure 30.6 for an example or see the section "The MCA Option" on page 1354 for a detailed description of Burt tables.

You can use the WEIGHT statement with the TABLES statement to read category frequencies. Specify the SUPPLEMENTARY statement to name variables with categories that are supplementary rows or columns. You cannot specify the ID or VAR statement with the TABLES statement. See the section "Using the TABLES Statement" on page 1334 for an example.

VAR Statement

VAR *variables* **;**

You should specify the VAR statement when your data are in tabular form. The VAR variables must be numeric. The VAR statement instructs PROC CORRESP to read an existing contingency table, binary indicator matrix, fuzzy-coded indicator matrix, or Burt table, rather than raw data. See the section "Algorithm and Notation" on page 1351 for a description of a binary indicator matrix and a fuzzy-coded indicator matrix.

You can specify the WEIGHT statement with the VAR statement to read category frequencies and designate supplementary rows. Specify the SUPPLEMENTARY statement to name supplementary variables. You cannot specify the TABLES statement with the VAR statement.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement specifies weights for each observation and indicates supplementary observations for simple correspondence analyses with VAR statement input. You can include only one WEIGHT statement, and the weight variable must be numeric.

If you omit the WEIGHT statement, each observation contributes a value of 1 to the frequency count for its category. That is, each observation represents one subject. When you specify a WEIGHT statement, each observation contributes the value of the weighting variable for that observation. For example, a weight of 3 means that the observation represents three subjects. Weight values are not required to be integers.

You can specify the WEIGHT statement with a TABLES statement to indicate category frequencies, as in the following example:

```
proc freq;
   tables a*b / out=outfreq sparse;
run;

proc corresp freqout;
   tables a, b;
   weight count;
run;
```

If you specify a VAR statement, you can specify the WEIGHT statement to indicate supplementary observations and to weight some rows of the table more heavily than others. When the value of the WEIGHT variable is negative, the observation is treated as supplementary, and the absolute value of the weight is used as the weighting value.

You cannot specify a WEIGHT statement with a VAR statement and the MCA option, because the table must be symmetric. Supplementary variables are indicated with the SUPPLEMENTARY statement, so differential weighting of rows is inappropriate.

Details: CORRESP Procedure

Input Data Set

PROC CORRESP can read two kinds of input:

- raw category responses on two or more classification variables with the TABLES statement
- a two-way contingency table with the VAR statement

You can use output from PROC FREQ as input for PROC CORRESP.

The classification variables referred to by the TABLES statement can be either numeric or character variables. Normally, all observations for a given variable that have the same formatted value are placed in the same level, and observations with different values are placed in different levels.

The variables in the VAR statement must be numeric. The values of the observations specify the cell frequencies. These values are not required to be integers, but only those observations with all nonnegative, nonmissing values are used in the correspondence analysis. Observations with one or more negative values are removed from the analysis.

The WEIGHT variable must be numeric. Observations with negative weights are treated as supplementary observations. The absolute values of the weights are used to weight the observations.

Using the TABLES Statement

This section explains some of the choices for the correspondence analysis input data table and illustrates some table-construction capabilities of PROC CORRESP. The SAS data set Neighbor, which follows, will be used throughout this section to illustrate various ways in which PROC CORRESP can read and process data. This data set consists of one observation for each resident in a fictitious neighborhood along with some personal information.

```
title 'PROC CORRESP Table Construction';

data Neighbor;
   input Name $ 1-10 Age $ 12-18 Sex $ 19-25
         Height $ 26-30 Hair $ 32-37;
   datalines;
Jones      Old    Male   Short White
Smith      Young  Female Tall  Brown
Kasavitz   Old    Male   Short Brown
Ernst      Old    Female Tall  White
Zannoria   Old    Female Short Brown
Spangel    Young  Male   Tall  Blond
```

```
Myers      Young  Male   Tall  Brown
Kasinski   Old    Male   Short Blond
Colman     Young  Female Short Blond
Delafave   Old    Male   Tall  Brown
Singer     Young  Male   Tall  Brown
Igor       Old           Short
;
```

This first step creates a simple contingency table or crosstabulation. In the TABLES statement, each variable list consists of a single variable. The following statements produce the table in Figure 30.3.

```
proc corresp data=Neighbor dimens=1 observed short;
   title2 'Simple Crosstabulation';
   ods select observed;
   tables Sex, Age;
run;
```

These statements create a contingency table with two rows (**Female** and **Male**) and two columns (**Old** and **Young**) and show the neighbors categorized by age and sex. The DIMENS=1 option specifies the number of dimensions in the correspondence analysis. Typically, you do not have to specify this option, because typically your tables will be larger than two by two. The default is DIMENS=2, which is too large for a table with a two-level factor. The OBSERVED option displays the contingency table. The SHORT option limits the displayed output. Because it contains missing values, the observation where Name='Igor' is omitted from the analysis. The table is shown in Figure 30.3.

Figure 30.3 Contingency Table for Sex, Age

PROC CORRESP Table Construction
Simple Crosstabulation

The CORRESP Procedure

Contingency Table

	Old	Young	Sum
Female	2	2	4
Male	4	3	7
Sum	6	5	11

The preceding example showed how to make a two-way contingency table based on the levels of two categorical variables, which, if it were larger, would be a very typical form of data for a correspondence analysis. However, many other types of tables, **N**, can be used as input to a correspondence analysis, and all tables can be defined based on a binary matrix, **Z**. The BINARY option enables you to directly compute and display this matrix. The TABLES statement consists of a single list of all the categorical variables. The following statements produce Figure 30.4.

```
proc corresp data=neighbor observed short binary;
   title2 'Binary Coding';
   ods select binary;
   tables Hair Height Sex Age;
run;
```

Figure 30.4 Binary Table Using the BINARY Option

PROC CORRESP Table Construction
Binary Coding

The CORRESP Procedure

Binary Table

	Blond	Brown	White	Short	Tall	Female	Male	Old	Young
1	0	0	1	1	0	0	1	1	0
2	0	1	0	0	1	1	0	0	1
3	0	1	0	1	0	0	1	1	0
4	0	0	1	0	1	1	0	1	0
5	0	1	0	1	0	1	0	1	0
6	1	0	0	0	1	0	1	0	1
7	0	1	0	0	1	0	1	0	1
8	1	0	0	1	0	0	1	1	0
9	1	0	0	1	0	1	0	0	1
10	0	1	0	0	1	0	1	1	0
11	0	1	0	0	1	0	1	0	1

In this case, $\mathbf{N} = \mathbf{Z}$ is directly analyzed. The binary matrix has one row for each individual or case and one column for each category. A binary table constructed from m categorical variables has m partitions. This binary table has four partitions, one for each of the four categorical variables. Each partition has a 1 in each row, and each row contains exactly four 1s since there are four categorical variables. More generally, the binary design matrix has exactly m 1s in each row. The 1s indicate the categories to which the observation applies. For example, the categorical variable Sex, with two levels (**Female** and **Male**), is coded using two indicator variables. For the variable Sex, a male would be coded Female=0 and Male=1, and a female would be coded Female=1 and Male=0. This is the same kind of coding that procedures like GLM and TRANSREG use for CLASS variables.

Implicitly, the binary table has an automatic row variable that is equal to the observation number. Alternatively, when there is a row ID variable, as there is in this case, you can use it as a row variable in the TABLES statement, and the resulting ordinary observed frequency table is the binary table. This example uses two variable lists: Name for the row variable, and Hair Height Sex Age for the column variables. Since two lists were provided, the BINARY option was not specified. The following statements produce Figure 30.5.

```
proc corresp data=neighbor observed short;
   title2 'Binary Coding';
   ods select observed;
   tables Name, Hair Height Sex Age;
run;
```

Figure 30.5 Binary Table Using a Row Variable

PROC CORRESP Table Construction
Binary Coding

The CORRESP Procedure

Contingency Table

	Blond	Brown	White	Short	Tall	Female	Male	Old	Young	Sum
Colman	1	0	0	1	0	1	0	0	1	4
Delafave	0	1	0	0	1	0	1	1	0	4
Ernst	0	0	1	0	1	1	0	1	0	4
Jones	0	0	1	1	0	0	1	1	0	4
Kasavitz	0	1	0	1	0	0	1	1	0	4
Kasinski	1	0	0	1	0	0	1	1	0	4
Myers	0	1	0	0	1	0	1	0	1	4
Singer	0	1	0	0	1	0	1	0	1	4
Smith	0	1	0	0	1	1	0	0	1	4
Spangel	1	0	0	0	1	0	1	0	1	4
Zannoria	0	1	0	1	0	1	0	1	0	4
Sum	3	6	2	5	6	4	7	6	5	44

With the MCA option, the Burt table ($\mathbf{Z}'\mathbf{Z}$) is analyzed. A Burt table is a partitioned symmetric matrix containing all pairs of crosstabulations among a set of categorical variables. Each diagonal partition is a diagonal matrix containing marginal frequencies (a crosstabulation of a variable with itself). Each off-diagonal partition is an ordinary contingency table. The following statements produce Figure 30.6.

```
proc corresp data=neighbor observed short mca;
   title2 'MCA Burt Table';
   ods select burt;
   tables Hair Height Sex Age;
run;
```

Note that there is a single variable list in the TABLES statement, since the row and column variable lists are the same.

Figure 30.6 MCA Burt Table

PROC CORRESP Table Construction
MCA Burt Table

The CORRESP Procedure

Burt Table

	Blond	Brown	White	Short	Tall	Female	Male	Old	Young
Blond	3	0	0	2	1	1	2	1	2
Brown	0	6	0	2	4	2	4	3	3
White	0	0	2	1	1	1	1	2	0
Short	2	2	1	5	0	2	3	4	1
Tall	1	4	1	0	6	2	4	2	4
Female	1	2	1	2	2	4	0	2	2
Male	2	4	1	3	4	0	7	4	3
Old	1	3	2	4	2	2	4	6	0
Young	2	3	0	1	4	2	3	0	5

This Burt table is composed of all pairs of crosstabulations among the variables Hair, Height, Sex, and Age. It is composed of sixteen individual subtables—the number of variables squared. Both the rows and the columns have the same nine categories (in this case Blond, Brown, White, Short, Tall, Female, Male, Old, and Young). Below the diagonal (from left to right, top to bottom) are the following crosstabulations: Height * Hair, Sex * Hair, Sex * Height, Age * Hair, Age * Height, and Age * Sex. Each crosstabulation below the diagonal has a transposed counterpart above the diagonal. The diagonal contains the crosstabulations: Hair * Hair, Height * Height, Sex * Sex, and Age * Age. The diagonal elements of the diagonal partitions contain marginal frequencies of the off-diagonal partitions. The table Hair * Height, for example, has three rows for Hair and two columns for Height. The values of the Hair * Height table, summed across rows, sum to the diagonal values of the Height * Height table, as displayed in the following results. The following statements produce Figure 30.7.

```
proc corresp data=neighbor observed short dimens=1;
   title2 'Part of the Burt Table';
   ods output observed=o;
   tables Hair Height, Height;
run;

proc print data=o(drop=sum) label noobs;
   where label ne 'Sum';
   label label = '00'x;
run;
```

Figure 30.7 Part of the Burt Table

PROC CORRESP Table Construction
Part of the Burt Table

	Short	Tall
Blond	2	1
Brown	2	4
White	1	1
Short	5	0
Tall	0	6

A simple crosstabulation of Hair × Height is $\mathbf{N} = \mathbf{Z}_{\text{Hair}}'\mathbf{Z}_{\text{Height}}$. Tables such as ($\mathbf{N} = \mathbf{Z}_{\text{Hair}}'\mathbf{Z}_{\text{Height,Sex}}$), made up of several crosstabulations, can also be analyzed in simple correspondence analysis. The following statements produce Figure 30.8.

```
proc corresp data=neighbor observed short dimens=1;
   title2 'Multiple Crosstabulations';
   ods select observed;
   tables Hair, Height Sex;
run;
```

Figure 30.8 Hair × (Height Sex) Crosstabulation

PROC CORRESP Table Construction
Multiple Crosstabulations

The CORRESP Procedure

Contingency Table

	Short	Tall	Female	Male	Sum
Blond	2	1	1	2	6
Brown	2	4	2	4	12
White	1	1	1	1	4
Sum	5	6	4	7	22

The following statements create a table with six rows (**Blond*Short**, **Blond*Tall**, **Brown*Short**, **Brown*Tall**, **White*Short**, and **White*Tall**) and four columns (**Female**, **Male**, **Old**, and **Young**). The levels of the row variables are crossed by the CROSS=ROW option, forming mutually exclusive categories. Hence each individual fits into exactly one row category, but two column categories. The following statements produce Figure 30.9.

```
proc corresp data=Neighbor cross=row observed short;
   title2 'Multiple Crosstabulations with Crossed Rows';
   ods select observed;
   tables Hair Height, Sex Age;
run;
```

Figure 30.9 Contingency Table for Hair * Height, Sex Age

PROC CORRESP Table Construction
Multiple Crosstabulations with Crossed Rows

The CORRESP Procedure

Contingency Table

	Female	Male	Old	Young	Sum
Blond * Short	1	1	1	1	4
Blond * Tall	0	1	0	1	2
Brown * Short	1	1	2	0	4
Brown * Tall	1	3	1	3	8
White * Short	0	1	1	0	2
White * Tall	1	0	1	0	2
Sum	4	7	6	5	22

You can enter supplementary variables with TABLES input by including a SUPPLEMENTARY statement. Variables named in the SUPPLEMENTARY statement indicate TABLES variables with categories that are supplementary. In other words, the categories of the variable Age are represented in the row and column space, but they are not used in determining the scores of the categories of the variables Hair, Height, and Sex. The variable used in the SUPPLEMENTARY statement must be listed in the TABLES statement as well. For example, the following statements create a Burt table with seven active rows and columns (**Blond**, **Brown**, **White**, **Short**, **Tall**, **Female**, **Male**) and two supplementary rows and columns (**Old** and **Young**). The following statements produce Figure 30.10.

```
proc corresp data=Neighbor observed short mca;
   title2 'MCA with Supplementary Variables';
   ods select burt supcols;
   tables Hair Height Sex Age;
   supplementary Age;
run;
```

Figure 30.10 Burt Table from PROC CORRESP with Supplementary Variables

PROC CORRESP Table Construction
MCA with Supplementary Variables

The CORRESP Procedure

Burt Table

	Blond	Brown	White	Short	Tall	Female	Male
Blond	3	0	0	2	1	1	2
Brown	0	6	0	2	4	2	4
White	0	0	2	1	1	1	1
Short	2	2	1	5	0	2	3
Tall	1	4	1	0	6	2	4
Female	1	2	1	2	2	4	0
Male	2	4	1	3	4	0	7

Figure 30.10 *continued*

Supplementary Columns

	Old	Young
Blond	1	2
Brown	3	3
White	2	0
Short	4	1
Tall	2	4
Female	2	2
Male	4	3

The following statements create a binary table with 7 active columns (**Blond**, **Brown**, **White**, **Short**, **Tall**, **Female**, **Male**), 2 supplementary columns (**Old** and **Young**), and 11 rows for the 11 observations with nonmissing values. The following statements produce Figure 30.11.

```
proc corresp data=Neighbor observed short binary;
   title2 'Supplementary Binary Variables';
   ods select binary supcols;
   tables Hair Height Sex Age;
   supplementary Age;
run;
```

Figure 30.11 Binary Table from PROC CORRESP with Supplementary Variables

PROC CORRESP Table Construction
Supplementary Binary Variables

The CORRESP Procedure

Binary Table

	Blond	Brown	White	Short	Tall	Female	Male
1	0	0	1	1	0	0	1
2	0	1	0	0	1	1	0
3	0	1	0	1	0	0	1
4	0	0	1	0	1	1	0
5	0	1	0	1	0	1	0
6	1	0	0	0	1	0	1
7	0	1	0	0	1	0	1
8	1	0	0	1	0	0	1
9	1	0	0	1	0	1	0
10	0	1	0	0	1	0	1
11	0	1	0	0	1	0	1

Figure 30.11 *continued*

Supplementary Columns

	Old	Young
1	1	0
2	0	1
3	1	0
4	1	0
5	1	0
6	0	1
7	0	1
8	1	0
9	0	1
10	1	0
11	0	1

Using the VAR Statement

With VAR statement input, the rows of the contingency table correspond to the observations of the input data set, and the columns correspond to the VAR statement variables. The values of the variables typically contain the table frequencies. The table in Figure 30.3 could be created with VAR statement input by using the following statements:

```
data Ages;
   input Sex $ Old Young;
   datalines;
Female  2 2
Male    4 3
;

proc corresp data=Ages dimens=1 observed short;
   var Old Young;
   id Sex;
run;
```

Only nonnegative values are accepted. Negative values are treated as missing, causing the observation to be excluded from the analysis. The values are not required to be integers. Row labels for the table are specified with an ID variable. Column labels are constructed from the variable name or variable label if one is specified. When you specify multiple correspondence analysis (MCA), the row and column labels are the same and are constructed from the variable names or labels, so you cannot include an ID statement. With MCA, the VAR statement must list the variables in the order in which the rows occur. An example is the table in Figure 30.6, which was created with the following TABLES statement.

```
tables Hair Height Sex Age;
```

This table could have been created with a VAR statement as follows:

```
proc corresp data=table nvars=4 mca;
   var Blond Brown White Short Tall Female Male Old Young;
run;
```

You must specify the NVARS= option in order to specify the number of original categorical variables with the MCA option. The option NVARS=n is needed to find boundaries between the subtables of the Burt table. If f is the sum of all elements in the Burt table $\mathbf{Z}'\mathbf{Z}$, then fn^{-2} is the number of rows in the binary matrix $\mathbf{Z}$. The sum of all elements in each diagonal subtable of the Burt table must be fn^{-2}.

To enter supplementary observations, include a WEIGHT statement with negative weights for those observations. Specify the SUPPLEMENTARY statement to include supplementary variables. You must list supplementary variables in both the VAR and SUPPLEMENTARY statements.

Missing and Invalid Data

With VAR statement input, observations with missing or negative frequencies are excluded from the analysis. Supplementary variables and supplementary observations with missing or negative frequencies are also excluded. Negative weights are valid with VAR statement input.

With TABLES statement input, observations with negative weights are excluded from the analysis. With this form of input, missing cell frequencies cannot occur. Observations with missing values on the categorical variables are excluded unless you specify the MISSING option. If you specify the MISSING option, ordinary missing values and special missing values are treated as additional levels of a categorical variable. In all cases, if any row or column of the constructed table contains only zeros, that row or column is excluded from the analysis.

Observations with missing weights are excluded from the analysis.

Coding, Fuzzy Coding, and Doubling

Sometimes, binary data such as Yes/No data are available—for example, 1 means "Yes, I have bought this brand in the last month" and 0 means "No, I have not bought this brand in the last month". The following statements read a data set with Yes/No purchase data for three hypothetical brands.

```
title 'Doubling Yes/No Data';

proc format;
   value yn 0 = 'No '  1 = 'Yes';
run;
```

```
data BrandChoice;
   input a b c;
   label a = 'Brand A' b = 'Brand B' c = 'Brand B';
   format a b c yn.;
   datalines;
0 0 1
1 1 0
0 1 1
0 1 0
1 0 0
;
```

Data such as these cannot be analyzed directly because the raw data do not consist of partitions, each with one column per level and exactly one 1 in each row. (See the section "Using the TABLES Statement" on page 1334.) The data must be *doubled* so that both Yes and No are represented by a column in the data matrix. The TRANSREG procedure provides one way of doubling. In the following statements, the DESIGN option specifies that PROC TRANSREG is being used only for coding, not analysis. The option SEPARATORS=': ' specifies that labels for the coded columns are constructed from input variable labels, followed by a colon and space, followed by the formatted value. The variables are designated in the MODEL statement as CLASS variables, and the ZERO=NONE option creates binary variables for all levels. The OUTPUT statement specifies the output data set and drops the _NAME_, _TYPE_, and Intercept variables. PROC TRANSREG stores a list of coded variable names in a macro variable &_TRGIND, which in this case has the value "aNo aYes bNo bYes cNo cYes". This macro variable can be used directly in the VAR statement in PROC CORRESP. The following statements produce Figure 30.12. Only the input table is displayed.

```
proc transreg data=BrandChoice design separators=': ';
   model class(a b c / zero=none);
   output out=Doubled(drop=_: Intercept);
run;

proc print label;
run;

proc corresp data=Doubled norow short;
   var &_trgind;
run;
```

Figure 30.12 Doubling Yes/No Data

Doubling Yes/No Data

Obs	Brand A: No	Brand A: Yes	Brand B: No	Brand B: Yes	Brand B: No	Brand B: Yes	Brand A	Brand B	Brand B
1	1	0	1	0	0	1	No	No	Yes
2	0	1	0	1	1	0	Yes	Yes	No
3	1	0	0	1	0	1	No	Yes	Yes
4	1	0	0	1	1	0	No	Yes	No
5	0	1	1	0	1	0	Yes	No	No

A fuzzy-coded indicator also sums to 1.0 across levels of the categorical variable, but it is coded with fractions rather than with 0 and 1. The fractions represent the distribution of the attribute across several levels of the categorical variable.

Ordinal variables, such as survey responses of 1 to 3, can be represented as two fuzzy-coded variables, as shown in Table 30.2.

Table 30.2 Coding an Ordinal Variable

Ordinal Values	Coding	
1	0.25	0.75
2	0.50	0.50
3	0.75	0.25

The values of the coding sum to one across the two coded variables.

These next steps illustrate the use of binary and fuzzy-coded indicator variables. Fuzzy-coded indicators are used to represent missing data. Note that the missing values in the observation Igor are coded with equal proportions. The following statements produce Figure 30.13.

```
title 'Fuzzy Coding of Missing Values';

proc transreg data=Neighbor design cprefix=0;
   model class(Age Sex Height Hair / zero=none);
   output out=Neighbor2(drop=_: Intercept);
   id Name;
run;

data Neighbor3;
   set Neighbor2;
   if Sex = ' ' then do;
      Female = 0.5;
      Male   = 0.5;
      end;
   if Hair = ' ' then do;
      White = 1/3;
      Brown = 1/3;
      Blond = 1/3;
      end;
run;

proc print label noobs data=Neighbor3(drop=age--name);
   format _numeric_ best4.;
run;
```

Figure 30.13 Fuzzy Coding of Missing Values

Fuzzy Coding of Missing Values

Age Old	Age Young	Sex Female	Sex Male	Height Short	Height Tall	Hair Blond	Hair Brown	Hair White
1	0	0	1	1	0	0	0	1
0	1	1	0	0	1	0	1	0
1	0	0	1	1	0	0	1	0
1	0	1	0	0	1	0	0	1
1	0	1	0	1	0	0	1	0
0	1	0	1	0	1	1	0	0
0	1	0	1	0	1	0	1	0
1	0	0	1	1	0	1	0	0
0	1	1	0	1	0	1	0	0
1	0	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1	0
1	0	0.5	0.5	1	0	0.33	0.33	0.33

There is one set of coded variables for each input categorical variable. If observation 12 is excluded, each set is a binary design matrix. Each design matrix has one column for each category and exactly one 1 in each row. Fuzzy coding is shown in the final observation, which corresponds to Igor. The observation for Igor has missing values for the variables Sex and Hair. The design matrix variables are coded with fractions that sum to one within each categorical variable.

An alternative way to represent missing data is to treat missing values as an additional level of the categorical variable. This alternative is available with the MISSING option in the PROC statement. This approach yields coordinates for missing responses, allowing the comparison of "missing" along with the other levels of the categorical variables.

Greenacre and Hastie (1987) discuss additional coding schemes, including one for continuous variables. Continuous variables can be coded with PROC TRANSREG by specifying BSPLINE(*variables* / degree=1) in the MODEL statement.

Creating a Data Set Containing the Crosstabulation

The CORRESP procedure can read or create a contingency or Burt table. PROC CORRESP is generally more efficient with VAR statement input than with TABLES statement input. TABLES statement input requires that the table be created from raw categorical variables, whereas the VAR statement is used to read an existing table. For extremely large problems, if PROC CORRESP runs out of memory, it might be possible to use some other method to create the table and then use VAR statement input with PROC CORRESP.

The following example uses the CORRESP, FREQ, and TRANSPOSE procedures to create rectangular tables from a SAS data set WORK.A that contains the categorical variables V1–V5. The Burt table examples assume that no categorical variable has a value found in any of the other categorical variables (that is, that each row and column label is unique).

You can use PROC CORRESP and the ODS OUTPUT statement as follows to create a rectangular two-way contingency table from two categorical variables:

```
proc corresp data=a observed short;
   ods output Observed=Obs(drop=Sum where=(Label ne 'Sum'));
   tables v1, v2;
run;
```

You can use PROC FREQ and PROC TRANSPOSE to create a rectangular two-way contingency table from two categorical variables, as in the following statements:

```
proc freq data=a;
   tables v1 * v2 / sparse noprint out=freqs;
run;

proc transpose data=freqs out=rfreqs(drop=_:);
   id  v2;
   var count;
   by  v1;
run;
```

You can use PROC CORRESP and the ODS OUTPUT statement as follows to create a Burt table from five categorical variables:

```
proc corresp data=a observed short mca;
   ods output Burt=Obs;
   tables v1-v5;
run;
```

You can use a DATA step, PROC FREQ, and PROC TRANSPOSE to create a Burt table from five categorical variables, as in the following statements:

```
data b;
   set a;
   array v[5] $ v1-v5;
   do i = 1 to 5;
      row = v[i];
      do j = 1 to 5;
         column = v[j];
         output;
         end;
      end;
   keep row column;
run;

proc freq data=b;
   tables row * column / sparse  noprint out=freqs;
run;

proc transpose data=freqs out=rfreqs(drop=_:);
   id  column;
   var count;
   by  row;
run;
```

Output Data Sets

PROC CORRESP has two output data sets. The OUTC= data set contains coordinates and the results of the correspondence analysis. The OUTF= data set contains frequencies and other cross-tabulation results.

The OUTC= Data Set

The OUTC= data set contains two or three character variables and $4n + 4$ numeric variables, where n is the number of axes from DIMENS=n (two by default). The OUTC= data set contains one observation for each row, column, supplementary row, and supplementary column point, and one observation for inertias.

The first variable is named _TYPE_ and identifies the type of observation. The values of _TYPE_ are as follows:

- The 'INERTIA' observation contains the total inertia in the INERTIA variable, and each dimension's inertia in the Contr1–Contr$n$ variables.
- The 'OBS' observations contain the coordinates and statistics for the rows of the table.
- The 'SUPOBS' observations contain the coordinates and statistics for the supplementary rows of the table.
- The 'VAR' observations contain the coordinates and statistics for the columns of the table.
- The 'SUPVAR' observations contain the coordinates and statistics for the supplementary columns of the table.

If you specify the SOURCE option, then the data set also contains a variable _VAR_ containing the name or label of the input variable from which that row originates. The name of the next variable is either _NAME_ or (if you specify an ID statement) the name of the ID variable.

For observations with a value of 'OBS' or 'SUPOBS' for the _TYPE_ variable, the values of the second variable are constructed as follows:

- When you use a VAR statement without an ID statement, the values are 'Row1', 'Row2', and so on.
- When you specify a VAR statement with an ID statement, the values are set equal to the values of the ID variable.
- When you specify a TABLES statement, the _NAME_ variable has values formed from the appropriate row variable values.

For observations with a value of 'VAR' or 'SUPVAR' for the _TYPE_ variable, the values of the second variable are equal to the names or labels of the VAR (or SUPPLEMENTARY) variables.

When you specify a TABLES statement, the values are formed from the appropriate column variable values.

The third and subsequent variables contain the numerical results of the correspondence analysis.

- Quality contains the quality of each point's representation in the DIMENS=*n* dimensional display, which is the sum of squared cosines over the first *n* dimensions.
- Mass contains the masses or marginal sums of the relative frequency matrix.
- Inertia contains each point's relative contribution to the total inertia.
- Dim1–Dim*n* contain the point coordinates.
- Contr1–Contr*n* contain the partial contributions to inertia.
- SqCos1–SqCos*n* contain the squared cosines.
- Best1–Best*n* and Best contain the summaries of the partial contributions to inertia.

The OUTF= Data Set

The OUTF= data set contains frequencies and percentages. It is similar to a PROC FREQ output data set. The OUTF= data set begins with a variable called _TYPE_, which contains the observation type. If the SOURCE option is specified, the data set contains two variables, _ROWVAR_ and _COLVAR_, that contain the names or labels of the row and column input variables from which each cell originates. The next two variables are classification variables that contain the row and column levels. If you use TABLES statement input and each variable list consists of a single variable, the names of the first two variables match the names of the input variables; otherwise, these variables are named Row and Column. The next two variables are Count and Percent, which contain frequencies and percentages.

The _TYPE_ variable can have the following values:

- 'OBSERVED' observations contain the contingency table.
- 'SUPOBS' observations contain the supplementary rows.
- 'SUPVAR' observations contain the supplementary columns.
- 'EXPECTED' observations contain the product of the row marginals and the column marginals divided by the grand frequency of the observed frequency table. For ordinary two-way contingency tables, these are the expected frequency matrix under the hypothesis of row and column independence.
- 'DEVIATION' observations contain the matrix of deviations between the observed frequency matrix and the product of its row marginals and column marginals divided by its grand frequency. For ordinary two-way contingency tables, these are the observed minus expected frequencies under the hypothesis of row and column independence.
- 'CELLCHI2' observations contain contributions to the total chi-square test statistic.

- 'RP' observations contain the row profiles.
- 'SUPRP' observations contain supplementary row profiles.
- 'CP' observations contain the column profiles.
- 'SUPCP' observations contain supplementary column profiles.

Computational Resources

Let

$$
\begin{aligned}
n_r &= \text{number of rows in the table} \\
n_c &= \text{number of columns in the table} \\
n &= \text{number of observations} \\
v &= \text{number of VAR statement variables} \\
t &= \text{number of TABLES statement variables} \\
c &= \max(n_r, n_c) \\
d &= \min(n_r, n_c)
\end{aligned}
$$

For TABLES statement input, more than

$$32(t+1) + 8(\max(2tn, (n_r+3)(n_c+3)))$$

bytes of array space are required.

For VAR statement input, more than

$$16(v+2) + 8(n_r+3)(n_c+3)$$

bytes of array space are required.

Memory

The computational resources formulas are underestimates of the amounts of memory needed to handle most problems. If you use a utility data set, and if memory could be used with perfect efficiency, then roughly the stated amount of memory would be needed. In reality, most problems require at least two or three times the minimum.

PROC CORRESP tries to store the raw data (TABLES input) and the contingency table in memory. If there is not enough memory, a utility data set is used, potentially resulting in a large increase in execution time.

Time

The time required to perform the generalized singular value decomposition is roughly proportional to $2cd^2 + 5d^3$. Overall computation time increases with table size at a rate roughly proportional to $(n_r n_c)^{\frac{3}{2}}$.

Algorithm and Notation

This section is primarily based on the theory of correspondence analysis found in Greenacre (1984). If you are interested in other references, see the section "Background" on page 1320.

Let $\mathbf{N}$ be the contingency table formed from those observations and variables that are not supplementary and from those observations that have no missing values and have a positive weight. This table is an $(n_r \times n_c)$ rank q matrix of nonnegative numbers with nonzero row and column sums. If $\mathbf{Z}_a$ is the binary coding for variable A, and $\mathbf{Z}_b$ is the binary coding for variable B, then $\mathbf{N} = \mathbf{Z}_a'\mathbf{Z}_b$ is a contingency table. Similarly, if $\mathbf{Z}_{b,c}$ contains the binary coding for both variables B and C, then $\mathbf{N} = \mathbf{Z}_a'\mathbf{Z}_{b,c}$ can also be input to a correspondence analysis. With the BINARY option, $\mathbf{N} = \mathbf{Z}$, and the analysis is based on a binary table. In multiple correspondence analysis, the analysis is based on a Burt table, $\mathbf{Z}'\mathbf{Z}$.

Let $\mathbf{1}$ be a vector of 1s of the appropriate order, let $\mathbf{I}$ be an identity matrix, and let diag$(\cdot)$ be a matrix-valued function that creates a diagonal matrix from a vector. Let

$$
\begin{aligned}
f &= \mathbf{1}'\mathbf{N}\mathbf{1} \\
\mathbf{P} &= \frac{1}{f}\mathbf{N} \\
\mathbf{r} &= \mathbf{P}\mathbf{1} \\
\mathbf{c} &= \mathbf{P}'\mathbf{1} \\
\mathbf{D_r} &= \text{diag}(\mathbf{r}) \\
\mathbf{D_c} &= \text{diag}(\mathbf{c}) \\
\mathbf{R} &= \mathbf{D_r^{-1}}\mathbf{P} \\
\mathbf{C}' &= \mathbf{D_c^{-1}}\mathbf{P}'
\end{aligned}
$$

The scalar f is the sum of all elements in $\mathbf{N}$. The matrix $\mathbf{P}$ is a matrix of relative frequencies. The vector $\mathbf{r}$ contains row marginal proportions or row "masses." The vector $\mathbf{c}$ contains column marginal proportions or column masses. The matrices $\mathbf{D_r}$ and $\mathbf{D_c}$ are diagonal matrices of marginals.

The rows of $\mathbf{R}$ contain the "row profiles." The elements of each row of $\mathbf{R}$ sum to one. Each (i, j) element of $\mathbf{R}$ contains the observed probability of being in column j given membership in row i. Similarly, the columns of $\mathbf{C}$ contain the column profiles. The coordinates in correspondence analysis are based on the generalized singular value decomposition of $\mathbf{P}$,

$$\mathbf{P} = \mathbf{A}\mathbf{D_u}\mathbf{B}'$$

where

$$\mathbf{A}'\mathbf{D}_\mathbf{r}^{-1}\mathbf{A} = \mathbf{B}'\mathbf{D}_\mathbf{c}^{-1}\mathbf{B} = \mathbf{I}$$

In multiple correspondence analysis,

$$\mathbf{P} = \mathbf{B}\mathbf{D}_\mathbf{u}^2\mathbf{B}'$$

The matrix $\mathbf{A}$, which is the rectangular matrix of left generalized singular vectors, has n_r rows and q columns; the matrix $\mathbf{D_u}$, which is a diagonal matrix of singular values, has q rows and columns; and the matrix $\mathbf{B}$, which is the rectangular matrix of right generalized singular vectors, has n_c rows and q columns. The columns of $\mathbf{A}$ and $\mathbf{B}$ define the principal axes of the column and row point clouds, respectively.

The generalized singular value decomposition of $\mathbf{P}-\mathbf{rc}'$, discarding the last singular value (which is zero) and the last left and right singular vectors, is exactly the same as a generalized singular value decomposition of $\mathbf{P}$, discarding the first singular value (which is one), the first left singular vector, $\mathbf{r}$, and the first right singular vector, $\mathbf{c}$. The first (trivial) column of $\mathbf{A}$ and $\mathbf{B}$ and the first singular value in $\mathbf{D_u}$ are discarded before any results are displayed. You can obtain the generalized singular value decomposition of $\mathbf{P}-\mathbf{rc}'$ from the ordinary singular value decomposition of $\mathbf{D}_\mathbf{r}^{-1/2}(\mathbf{P}-\mathbf{rc}')\mathbf{D}_\mathbf{c}^{-1/2}$:

$$\mathbf{D}_\mathbf{r}^{-1/2}(\mathbf{P}-\mathbf{rc}')\mathbf{D}_\mathbf{c}^{-1/2} = \mathbf{U}\mathbf{D}_\mathbf{u}\mathbf{V}' = (\mathbf{D}_\mathbf{r}^{-1/2}\mathbf{A})\mathbf{D}_\mathbf{u}(\mathbf{D}_\mathbf{c}^{-1/2}\mathbf{B})'$$

$$\mathbf{P}-\mathbf{rc}' = \mathbf{D}_\mathbf{r}^{1/2}\mathbf{U}\mathbf{D}_\mathbf{u}\mathbf{V}'\mathbf{D}_\mathbf{c}^{1/2} = (\mathbf{D}_\mathbf{r}^{1/2}\mathbf{U})\mathbf{D}_\mathbf{u}(\mathbf{D}_\mathbf{c}^{1/2}\mathbf{V})' = \mathbf{A}\mathbf{D}_\mathbf{u}\mathbf{B}'$$

Hence, $\mathbf{A} = \mathbf{D}_\mathbf{r}^{1/2}\mathbf{U}$ and $\mathbf{B} = \mathbf{D}_\mathbf{c}^{1/2}\mathbf{V}$.

The default row coordinates are $\mathbf{D}_\mathbf{r}^{-1}\mathbf{A}\mathbf{D}_\mathbf{u}$, and the default column coordinates are $\mathbf{D}_\mathbf{c}^{-1}\mathbf{B}\mathbf{D}_\mathbf{u}$. Typically the first two columns of $\mathbf{D}_\mathbf{r}^{-1}\mathbf{A}\mathbf{D}_\mathbf{u}$ and $\mathbf{D}_\mathbf{c}^{-1}\mathbf{B}\mathbf{D}_\mathbf{u}$ are plotted to display graphically associations between the row and column categories. The plot consists of two overlaid plots, one for rows and one for columns. The row points are row profiles, and the column points are column profiles, both rescaled so that distances between profiles can be displayed as ordinary Euclidean distances, then orthogonally rotated to a principal axes orientation. Distances between row points and other row points have meaning, as do distances between column points and other column points. However, distances between column points and row points are not interpretable.

The PROFILE=, ROW=, and COLUMN= Options

The PROFILE=, ROW=, and COLUMN= options standardize the coordinates before they are displayed and placed in the output data set. The options PROFILE=BOTH, PROFILE=ROW, and PROFILE=COLUMN provide the standardizations that are typically used in correspondence analysis. There are six choices each for row and column coordinates (see Table 30.3). However, most of the combinations of the ROW= and COLUMN= options are not useful. The ROW= and COLUMN= options are provided for completeness, but they are not intended for general use.

Table 30.3 Coordinates

ROW=	Matrix Formula
A	$\mathbf{A}$
AD	$\mathbf{AD_u}$
DA	$\mathbf{D_r^{-1}A}$
DAD	$\mathbf{D_r^{-1}AD_u}$
DAD1/2	$\mathbf{D_r^{-1}AD_u^{1/2}}$
DAID1/2	$\mathbf{D_r^{-1}A(I+D_u)^{1/2}}$
COLUMN=	**Matrix Formula**
B	$\mathbf{B}$
BD	$\mathbf{BD_u}$
DB	$\mathbf{D_c^{-1}B}$
DBD	$\mathbf{D_c^{-1}BD_u}$
DBD1/2	$\mathbf{D_c^{-1}BD_u^{1/2}}$
DBID1/2	$\mathbf{D_c^{-1}B(I+D_u)^{1/2}}$

When PROFILE=ROW (ROW=DAD and COLUMN=DB), the row coordinates $\mathbf{D_r^{-1}AD_u}$ and column coordinates $\mathbf{D_c^{-1}B}$ provide a correspondence analysis based on the row profile matrix. The row profile (conditional probability) matrix is defined as $\mathbf{R} = \mathbf{D_r^{-1}P} = \mathbf{D_r^{-1}AD_uB'}$. The elements of each row of $\mathbf{R}$ sum to one. Each (i, j) element of $\mathbf{R}$ contains the observed probability of being in column j given membership in row i. The "principal" row coordinates $\mathbf{D_r^{-1}AD_u}$ and "standard" column coordinates $\mathbf{D_c^{-1}B}$ provide a decomposition of $\mathbf{D_r^{-1}AD_uB'D_c^{-1}} = \mathbf{D_r^{-1}PD_c^{-1}} = \mathbf{RD_c^{-1}}$. Since $\mathbf{D_r^{-1}AD_u} = \mathbf{RD_c^{-1}B}$, the row coordinates are weighted centroids of the column coordinates. Each column point, with coordinates scaled to standard coordinates, defines a vertex in $(n_c - 1)$-dimensional space. All of the principal row coordinates are located in the space defined by the standard column coordinates. Distances among row points have meaning, but distances among column points and distances between row and column points are not interpretable.

The option PROFILE=COLUMN can be described as applying the PROFILE=ROW formulas to the transpose of the contingency table. When PROFILE=COLUMN (ROW=DA and COLUMN=DBD), the principal column coordinates $\mathbf{D_c^{-1}BD_u}$ are weighted centroids of the standard row coordinates $\mathbf{D_r^{-1}A}$. Each row point, with coordinates scaled to standard coordinates, defines a vertex in $(n_r - 1)$-dimensional space. All of the principal column coordinates are located in the space defined by the standard row coordinates. Distances among column points have meaning, but distances among row points and distances between row and column points are not interpretable.

The usual sets of coordinates are given by the default PROFILE=BOTH (ROW=DAD and COLUMN=DBD). All of the summary statistics, such as the squared cosines and contributions to inertia, apply to these two sets of points. One advantage to using these coordinates is that both sets ($\mathbf{D_r^{-1}AD_u}$ and $\mathbf{D_c^{-1}BD_u}$) are postmultiplied by the diagonal matrix $\mathbf{D_u}$, which has diagonal values that are all less than or equal to one. When $\mathbf{D_u}$ is a part of the definition of only one set of coordinates, that set forms a tight cluster near the centroid, whereas the other set of points is more widely dispersed. Including $\mathbf{D_u}$ in both sets makes a better graphical display. However, care must be taken in interpreting such a plot. No correct interpretation of distances between row points and column points can be made.

Another property of this choice of coordinates concerns the geometry of distances between points within each set. The default row coordinates can be decomposed into $\mathbf{D_r^{-1}AD_u} = \mathbf{D_r^{-1}AD_uB'D_c^{-1}B} = (\mathbf{D_r^{-1}P})(\mathbf{D_c^{-1/2}})(\mathbf{D_c^{-1/2}B})$. The row coordinates are row profiles $(\mathbf{D_r^{-1}P})$, rescaled by $\mathbf{D_c^{-1/2}}$ (rescaled so that distances between profiles are transformed from a chi-square metric to a Euclidean metric), then orthogonally rotated (with $\mathbf{D_c^{-1/2}B}$) to a principal axes orientation. Similarly, the column coordinates are column profiles rescaled to a Euclidean metric and orthogonally rotated to a principal axes orientation.

The rationale for computing distances between row profiles by using the non-Euclidean chi-square metric is as follows. Each row of the contingency table can be viewed as a realization of a multinomial distribution conditional on its row marginal frequency. The null hypothesis of row and column independence is equivalent to the hypothesis of homogeneity of the row profiles. A significant chi-square statistic is geometrically interpreted as a significant deviation of the row profiles from their centroid, $\mathbf{c}'$. The chi-square metric is the Mahalanobis metric between row profiles based on their estimated covariance matrix under the homogeneity assumption (Greenacre and Hastie 1987). A parallel argument can be made for the column profiles.

When ROW=DAD1/2 and COLUMN=DBD1/2 (Gifi 1990; van der Heijden and de Leeuw 1985), the row coordinates $\mathbf{D_r^{-1}AD_u^{1/2}}$ and column coordinates $\mathbf{D_c^{-1}BD_u^{1/2}}$ are a decomposition of $\mathbf{D_r^{-1}PD_c^{-1}}$.

In all of the preceding pairs, distances between row and column points are not meaningful. This prompted Carroll, Green, and Schaffer (1986) to propose that row coordinates $\mathbf{D_r^{-1}A(I+D_u)^{1/2}}$ and column coordinates $\mathbf{D_c^{-1}B(I+D_u)^{1/2}}$ be used. These coordinates are (except for a constant scaling) the coordinates from a multiple correspondence analysis of a Burt table created from two categorical variables. This standardization is available with ROW=DAID1/2 and COLUMN=DBID1/2. However, this approach has been criticized on both theoretical and empirical grounds by Greenacre (1989). The Carroll, Green, and Schaffer standardization relies on the assumption that the chi-square metric is an appropriate metric for measuring the distance between the columns of a bivariate indicator matrix. See the section "Using the TABLES Statement" on page 1334 for a description of indicator matrices. Greenacre (1989) showed that this assumption cannot be justified.

The MCA Option

The MCA option performs a multiple correspondence analysis (MCA). This option requires a Burt table. You can specify the MCA option with a table created from a design matrix with fuzzy coding schemes as long as every row of every partition of the design matrix has the same marginal sum. For example, each row of each partition could contain the probabilities that the observation is a member of each level. Then the Burt table constructed from this matrix no longer contains all integers, and the diagonal partitions are no longer diagonal matrices, but MCA is still valid.

A TABLES statement with a single variable list creates a Burt table. Thus, you can always specify the MCA option with this type of input. If you use the MCA option when reading an existing table with a VAR statement, you must ensure that the table is a Burt table.

If you perform MCA on a table that is not a Burt table, the results of the analysis are invalid. If the table is not symmetric, or if the sums of all elements in each diagonal partition are not equal, PROC CORRESP displays an error message and quits.

A subset of the columns of a Burt table is not necessarily a Burt table, so in MCA it is not appropriate to designate arbitrary columns as supplementary. You can, however, designate all columns from one or more categorical variables as supplementary.

The results of a multiple correspondence analysis of a Burt table $\mathbf{Z}'\mathbf{Z}$ are the same as the column results from a simple correspondence analysis of the binary (or fuzzy) matrix $\mathbf{Z}$. Multiple correspondence analysis is not a simple correspondence analysis of the Burt table. It is not appropriate to perform a simple correspondence analysis of a Burt table. The MCA option is based on $\mathbf{P} = \mathbf{B}\mathbf{D}_\mathbf{u}^{-2}\mathbf{B}'$, whereas a simple correspondence analysis of the Burt table would be based on $\mathbf{P} = \mathbf{B}\mathbf{D}_\mathbf{u}\mathbf{B}'$.

Since the rows and columns of the Burt table are the same, no row information is displayed or written to the output data sets. The resulting inertias and the default (COLUMN=DBD) column coordinates are the appropriate inertias and coordinates for an MCA. The supplementary column coordinates, cosines, and quality of representation formulas for MCA differ from the simple correspondence analysis formulas because the design matrix column profiles and left singular vectors are not available.

The following statements create a Burt table and perform a multiple correspondence analysis:

```
proc corresp data=Neighbor observed short mca;
   tables Hair Height Sex Age;
run;
```

Both the rows and the columns have the same nine categories (Blond, Brown, White, Short, Tall, Female, Male, Old, and Young).

MCA Adjusted Inertias

The usual principal inertias of a Burt table constructed from m categorical variables in MCA are the eigenvalues u_k from $\mathbf{D}_\mathbf{u}^2$. The problem with these inertias is that they provide a pessimistic indication of fit. Benzécri (1979) proposed the following inertia adjustment, which is also described by Greenacre (1984, p. 145):

$$\left(\frac{m}{m-1}\right)^2 \times \left(u_k - \frac{1}{m}\right)^2 \qquad \text{for } u_k > \frac{1}{m}$$

The Benzécri adjustment is available with the BENZECRI option.

Greenacre (1994, p. 156) argues that the Benzécri adjustment overestimates the quality of fit. Greenacre proposes instead the following inertia adjustment:

$$\left(\frac{m}{m-1}\right)^2 \times \left(\sqrt{u_k} - \frac{1}{m}\right)^2 \qquad \text{for } \sqrt{u_k} > \frac{1}{m}$$

The Greenacre adjustment is available with the GREENACRE option.

Ordinary unadjusted inertias are printed by default with MCA when neither the BENZECRI nor the GREENACRE option is specified. However, the unadjusted inertias are not printed by default when either the BENZECRI or the GREENACRE option is specified. To display both adjusted and unadjusted inertias, specify the UNADJUSTED option in addition to the relevant adjusted inertia option (BENZECRI, GREENACRE, or both).

Supplementary Rows and Columns

Supplementary rows and columns are represented as points in the joint row and column space, but they are not used in determining the locations of the other active rows and columns of the table. The formulas that are used to compute coordinates for the supplementary rows and columns depend on the PROFILE= option or the ROW= and COLUMN= options. Let $\mathbf{S_o}$ be a matrix with rows that contain the supplementary observations, and let $\mathbf{S_v}$ be a matrix with rows that contain the supplementary variables. Note that $\mathbf{S_v}$ is defined to be the transpose of the supplementary variable partition of the table. Let $\mathbf{R_s} = \text{diag}(\mathbf{S_o 1})^{-1}\mathbf{S_o}$ be the supplementary observation profile matrix, and let $\mathbf{C_s} = \text{diag}(\mathbf{S_v 1})^{-1}\mathbf{S_v}$ be the supplementary variable profile matrix. Note that the notation $\text{diag}(\cdot)^{-1}$ means to convert the vector to a diagonal matrix, then invert the diagonal matrix. The coordinates for the supplementary observations and variables are shown in Table 30.4.

Table 30.4 Coordinates for Supplementary Observations

ROW=	Matrix Formula
A	$\frac{1}{f}\mathbf{S_o D_c^{-1} B D_u^{-1}}$
AD	$\frac{1}{f}\mathbf{S_o D_c^{-1} B}$
DA	$\mathbf{R_s D_c^{-1} B D_u^{-1}}$
DAD	$\mathbf{R_s D_c^{-1} B}$
DAD1/2	$\mathbf{R_s D_c^{-1} B D_u^{-1/2}}$
DAID1/2	$\mathbf{R_s D_c^{-1} B D_u^{-1}(I + D_u)^{1/2}}$
COLUMN=	**Matrix Formula**
B	$\frac{1}{f}\mathbf{S_v D_r^{-1} A D_u^{-1}}$
BD	$\frac{1}{f}\mathbf{S_v D_r^{-1} A}$
DB	$\mathbf{C_s D_r^{-1} A D_u^{-1}}$
DBD	$\mathbf{C_s D_r^{-1} A}$
DBD1/2	$\mathbf{C_s D_r^{-1} A D_u^{-1/2}}$
DBID1/2	$\mathbf{C_s D_r^{-1} A D_u^{-1}(I + D_u)^{1/2}}$
MCA COLUMN=	**Matrix Formula**
B	not allowed
BD	not allowed
DB	$\mathbf{C_s D_r^{-1} B D_u^{-2}}$
DBD	$\mathbf{C_s D_r^{-1} B D_u^{-1}}$
DBD1/2	$\mathbf{C_s D_r^{-1} B D_u^{-3/2}}$
DBID1/2	$\mathbf{C_s D_r^{-1} B D_u^{-2}(I + D_u)^{1/2}}$

Statistics That Aid Interpretation

The partial contributions to inertia, squared cosines, quality of representation, inertia, and mass provide additional information about the coordinates. These statistics are displayed by default.

Include the SHORT or NOPRINT option in the PROC CORRESP statement to avoid having these statistics displayed.

These statistics pertain to the default PROFILE=BOTH coordinates, no matter what values you specify for the ROW=, COLUMN=, or PROFILE= option. Let sq(·) be a matrix-valued function denoting element-wise squaring of the argument matrix. Let t be the total inertia (the sum of the elements in $\mathbf{D}_\mathbf{u}^2$).

In MCA, let $\mathbf{D_s}$ be the Burt table partition containing the intersection of the supplementary columns and the supplementary rows. The matrix $\mathbf{D_s}$ is a diagonal matrix of marginal frequencies of the supplemental columns of the binary matrix $\mathbf{Z}$. Let p be the number of rows in this design matrix. The statistics are defined in Table 30.5.

Table 30.5 Statistics That Aid Interpretation

Statistic	Matrix Formula
Row partial contributions to inertia	$\mathbf{D_r^{-1}}\text{sq}(\mathbf{A})$
Column partial contributions to inertia	$\mathbf{D_c^{-1}}\text{sq}(\mathbf{B})$
Row squared cosines	$\text{diag}(\text{sq}(\mathbf{AD_u})\mathbf{1})^{-1}\text{sq}(\mathbf{AD_u})$
Column squared cosines	$\text{diag}(\text{sq}(\mathbf{BD_u})\mathbf{1})^{-1}\text{sq}(\mathbf{BD_u})$
Row mass	$\mathbf{r}$
Column mass	$\mathbf{c}$
Row inertia	$\frac{1}{t}\mathbf{D_r^{-1}}\ \text{sq}(\mathbf{AD_u})\mathbf{1}$
Column inertia	$\frac{1}{t}\mathbf{D_c^{-1}}\ \text{sq}(\mathbf{BD_u})\mathbf{1}$
Supplementary row squared cosines	$\text{diag}(\text{sq}(\mathbf{R_s} - \mathbf{1c}')\mathbf{D_c^{-1}1})^{-1}\text{sq}(\mathbf{R_s D_c^{-1} B})$
Supplementary column squared cosines	$\text{diag}(\text{sq}(\mathbf{C_s} - \mathbf{1r}')\mathbf{D_r^{-1}1})^{-1}\text{sq}(\mathbf{C_s D_r^{-1} A})$
MCA supplementary column squared cosines	$\mathbf{D_s}(p\mathbf{I} - \mathbf{D_s})^{-1}\ \text{sq}(\mathbf{C_s D_r^{-1} B D_u^{-1}})$

The quality of representation in the DIMENS=n dimensional display of any point is the sum of its squared cosines over only the n dimensions. Inertia and mass are not defined for supplementary points.

A table that summarizes the partial contributions to inertia table is also computed. The points that best explain the inertia of each dimension and the dimension to which each point contributes the most inertia are indicated. The output data set variable names for this table are Best1–Best$n$ (where DIMENS=n) and Best. The Best column contains the dimension number of the largest partial contribution to inertia for each point (the index of the maximum value in each row of $\mathbf{D_r^{-1}}\text{sq}(\mathbf{A})$ or $\mathbf{D_c^{-1}}\text{sq}(\mathbf{B})$).

For each row, the Best1–Best$n$ columns contain either the corresponding value of Best, if the point is one of the biggest contributors to the dimension's inertia, or 0 if it is not. Specifically, Best1 contains

the value of Best for the point with the largest contribution to dimension one's inertia. A cumulative proportion sum is initialized to this point's partial contribution to the inertia of dimension one. If this sum is less than the value for the MININERTIA= option, then Best1 contains the value of Best for the point with the second-largest contribution to dimension one's inertia. Otherwise, this point's Best1 is 0. This point's partial contribution to inertia is added to the sum. This process continues for the point with the third-largest partial contribution, and so on, until adding a point's contribution to the sum increases the sum beyond the value of the MININERTIA= option. This same algorithm is then used for Best2, and so on.

For example, the following table contains contributions to inertia and the corresponding Best variables. The contribution to inertia variables are proportions that sum to 1 within each column. The first point makes its greatest contribution to the inertia of dimension two, so Best for point one is set to 2, and Best1–Best3 for point one must all be 0 or 2. The second point also makes its greatest contribution to the inertia of dimension two, so Best for point two is set to 2, and Best1–Best3 for point two must all be 0 or 2, and so on.

Assume MININERTIA=0.8, the default. Table 30.6 shows some contributions to inertia. In dimension one, the largest contribution is 0.41302 for the fourth point, so Best1 is set to 1, the value of Best for the fourth point. Because this value is less than 0.8, the second-largest value (0.36456 for point five) is found and its Best1 is set to its Best's value of 1. Because $0.41302 + 0.36456 = 0.77758$ is less than 0.8, the third point (0.0882 at point eight) is found and Best1 is set to 3, since the contribution to dimension three for that point is greater than the contribution to dimension one. This increases the sum of the partial contributions to greater than 0.8, so the remaining Best1 values are all 0.

Table 30.6 Best Statistics

Contr1	Contr2	Contr3	Best1	Best2	Best3	Best
0.01593	0.32178	0.07565	0	2	2	2
0.03014	0.24826	0.07715	0	2	2	2
0.00592	0.02892	0.02698	0	0	0	2
0.41302	0.05191	0.05773	1	0	0	1
0.36456	0.00344	0.15565	1	0	1	1
0.03902	0.30966	0.11717	0	2	2	2
0.00019	0.01840	0.00734	0	0	0	2
0.08820	0.00527	0.16555	3	0	3	3
0.01447	0.00024	0.03851	0	0	0	3
0.02855	0.01213	0.27827	0	0	3	3

Displayed Output

The display options control the amount of displayed output. By default, the following information is displayed:

- an inertia and chi-square decomposition table including the total inertia, the principal inertias of each dimension (eigenvalues), the singular values (square roots of the eigenvalues), each

dimension's percentage of inertia, a horizontal bar chart of the percentages, and the total chi-square with its degrees of freedom and decomposition. The chi-square statistics and degrees of freedom are valid only when the constructed table is an ordinary two-way contingency table.

- the coordinates of the rows and columns on the dimensions
- the mass, relative contribution to the total inertia, and quality of representation in the DIMENS=*n* dimensional display of each row and column
- the squared cosines of the angles between each axis and a vector from the origin to the point
- the partial contributions of each point to each dimension's inertia
- the table of indicators of which points best explain the inertia of each dimension

Specific display options and combinations of options display output as follows.

If you specify the OBSERVED or ALL option and you do not specify PRINT=PERCENT, PROC CORRESP displays the following:

- the contingency table, including the row and column marginal frequencies; or with BINARY, the binary table; or the Burt table in MCA
- the supplementary rows
- the supplementary columns

If you specify the OBSERVED or ALL option, with the PRINT=PERCENT or PRINT=BOTH option, PROC CORRESP displays the following:

- the contingency table or Burt table in MCA, scaled to percentages, including the row and column marginal percentages
- the supplementary rows, scaled to percentages
- the supplementary columns, scaled to percentages

If you specify the EXPECTED or ALL option and you do not specify PRINT=PERCENT, PROC CORRESP displays the product of the row marginals and the column marginals divided by the grand frequency of the observed frequency table. For ordinary two-way contingency tables, these are the expected frequencies under the hypothesis of row and column independence.

If you specify the EXPECTED or ALL option with the PRINT=PERCENT or PRINT=BOTH option, PROC CORRESP displays the product of the row marginals and the column marginals divided by the grand frequency of the observed percentages table. For ordinary two-way contingency tables, these are the expected percentages under the hypothesis of row and column independence.

If you specify the DEVIATION or ALL option and you do not specify PRINT=PERCENT, PROC CORRESP displays the observed minus expected frequencies. For ordinary two-way contingency tables, these are the expected frequencies under the hypothesis of row and column independence.

If you specify the DEVIATION or ALL option with the PRINT=PERCENT or PRINT=BOTH option, PROC CORRESP displays the observed minus expected percentages. For ordinary two-way contingency tables, these are the expected percentages under the hypothesis of row and column independence.

If you specify the CELLCHI2 or ALL option and you do not specify PRINT=PERCENT, PROC CORRESP displays contributions to the total chi-square test statistic, including the row and column marginals. The intersection of the marginals contains the total chi-square statistic.

If you specify the CELLCHI2 or ALL option with the PRINT=PERCENT or the PRINT=BOTH option, PROC CORRESP displays contributions to the total chi-square, scaled to percentages, including the row and column marginals.

If you specify the RP or ALL option and you do not specify PRINT=PERCENT, PROC CORRESP displays the row profiles and the supplementary row profiles.

If you specify the RP or ALL option with the PRINT=PERCENT or the PRINT=BOTH option, PROC CORRESP displays the row profiles (scaled to percentages) and the supplementary row profiles (scaled to percentages).

If you specify the CP or ALL option and you do not specify PRINT=PERCENT, PROC CORRESP displays the column profiles and the supplementary column profiles.

If you specify the CP or ALL option with the PRINT=PERCENT or PRINT=BOTH option, PROC CORRESP displays the column profiles (scaled to percentages) and the supplementary column profiles (scaled to percentages).

If you do not specify the NOPRINT option, PROC CORRESP displays the inertia and chi-square decomposition table. This includes the nonzero singular values of the contingency table (or, in MCA, the binary matrix $\mathbf{Z}$ used to create the Burt table), the nonzero principal inertias (or eigenvalues) for each dimension, the total inertia, the total chi-square, the decomposition of chi-square, the chi-square degrees of freedom (appropriate only when the table is an ordinary two-way contingency table), the percentage of the total chi-square and inertia for each dimension, and a bar chart of the percentages.

If you specify the MCA option and you do not specify the NOPRINT option, PROC CORRESP displays the adjusted inertias. This includes the nonzero adjusted inertias, percentages, cumulative percentages, and a bar chart of the percentages.

If you do not specify the NOROW, NOPRINT, or MCA option, PROC CORRESP displays the row coordinates and the supplementary row coordinates (displayed when there are supplementary row points).

If you do not specify the NOROW, NOPRINT, MCA, or SHORT option, PROC CORRESP displays the following:

- the summary statistics for the row points, including the quality of representation of the row points in the n-dimensional display, the mass, and the relative contributions to inertia
- the quality of representation of the supplementary row points in the n-dimensional display (displayed when there are supplementary row points)
- the partial contributions to inertia for the row points

- the table of indicators of which row points best explain the inertia of each dimension
- the squared cosines for the row points
- the squared cosines for the supplementary row points (displayed when there are supplementary row points)

If you do not specify the NOCOLUMN or NOPRINT option, PROC CORRESP displays the column coordinates and the supplementary column coordinates (displayed when there are supplementary column points).

If you do not specify the NOCOLUMN, NOPRINT, or SHORT option, PROC CORRESP displays the following:

- the summary statistics for the column points, including the quality of representation of the column points in the n-dimensional display, the mass, and the relative contributions to inertia for the supplementary column points
- the quality of representation of the supplementary column points in the n-dimensional display (displayed when there are supplementary column points)
- the partial contributions to inertia for the column points
- the table of indicators of which column points best explain the inertia of each dimension
- the squared cosines for the column points
- the squared cosines for the supplementary column points

ODS Table Names

PROC CORRESP assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 30.7 along with the PROC statement options needed to produce the table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 30.7 ODS Tables Produced by PROC CORRESP

ODS Table Name	Description	Option
AdjInGreenacre	Greenacre Inertia Adjustment	GREENACRE
AdjInBenzecri	Benzécri Inertia Adjustment	BENZECRI
Binary	Binary table	OBSERVED, BINARY
BinaryPct	Binary table percentages	OBSERVED, BINARY*
Burt	Burt table	OBSERVED, MCA
BurtPct	Burt table percentages	OBSERVED, MCA*
CellChiSq	Contributions to chi-square	CELLCHI2
CellChiSqPct	Contributions, percentages	CELLCHI2*
ColBest	Col best indicators	default

Table 30.7 *continued*

ODS Table Name	Description	Option
ColContr	Col contributions to inertia	default
ColCoors	Col coordinates	default
ColProfiles	Col profiles	CP
ColProfilesPct	Col profiles, percentages	CP*
ColQualMassIn	Col quality, mass, inertia	default
ColSqCos	Col squared cosines	default
DF	DF, chi-square (not displayed)	default
Deviations	Observed - expected freqs	DEVIATIONS
DeviationsPct	Observed - expected percentages	DEVIATIONS*
Expected	Expected frequencies	EXPECTED
ExpectedPct	Expected percentages	EXPECTED*
Inertias	Inertia decomposition table	default
Observed	Observed frequencies	OBSERVED
ObservedPct	Observed percentages	OBSERVED*
RowBest	Row best indicators	default
RowContr	Row contributions to inertia	default
RowCoors	Row coordinates	default
RowProfiles	Row profiles	RP
RowProfilesPct	Row profiles, percentages	RP*
RowQualMassIn	Row quality, mass, inertia	default
RowSqCos	Row squared cosines	default
SupColCoors	Supp col coordinates	default
SupColProfiles	Supp col profiles	CP
SupColProfilesPct	Supp col profiles, percentages	CP*
SupColQuality	Supp col quality	default
SupCols	Supplementary col freq	OBSERVED
SupColsPct	Supplementary col percentages	OBSERVED*
SupColSqCos	Supp col squared cosines	default
SupRows	Supplementary row freqs	OBSERVED
SupRowCoors	Supp row coordinates	default
SupRowProfiles	Supp row profiles	RP
SupRowProfilesPct	Supp row profiles, percentages	RP*
SupRowQuality	Supp row quality	default
SupRowsPct	Supplementary row percentages	OBSERVED*
SupRowSqCos	Supp row squared cosines	default

*Percentages are displayed when you specify the PRINT=PERCENT or PRINT=BOTH option.

ODS Graphics

To request graphics with PROC CORRESP, you must first enable ODS Graphics by specifying the ODS GRAPHICS ON statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. You can reference every graph produced through ODS Graphics with a name. The names of the graph that PROC CORRESP generates is listed in Table 30.8. It is displayed by default when ODS graphics is enabled.

Table 30.8 ODS Graphics Produced by PROC CORRESP

ODS Graph Name	Plot Description
ConfigPlot	Correspondence analysis plot

Examples: CORRESP Procedure

Example 30.1: Simple and Multiple Correspondence Analysis of Automobiles and Their Owners

In this example, PROC CORRESP creates a contingency table from categorical data and performs a simple correspondence analysis. The data are from a sample of individuals who were asked to provide information about themselves and their automobiles. The questions included origin of the automobile (American, Japanese, European) and family status (single, married, single and living with children, married living with children).

The first steps read the input data and assign formats. PROC CORRESP is used to perform the simple correspondence analysis. The ALL option displays all tables, including the contingency table, chi-square information, profiles, and all results of the correspondence analysis. The OUTC= option creates an output coordinate data set. The TABLES statement specifies the row and column categorical variables. The results are displayed with ODS Graphics.

The following statements produce Output 30.1.1:

```
title1 'Automobile Owners and Auto Attributes';
title2 'Simple Correspondence Analysis';

proc format;
   value Origin  1 = 'American' 2 = 'Japanese' 3 = 'European';
   value Size    1 = 'Small'    2 = 'Medium'   3 = 'Large';
   value Type    1 = 'Family'   2 = 'Sporty'   3 = 'Work';
   value Home    1 = 'Own'      2 = 'Rent';
   value Sex     1 = 'Male'     2 = 'Female';
   value Income  1 = '1 Income' 2 = '2 Incomes';
```

```
   value Marital 1 = 'Single with Kids' 2 = 'Married with Kids'
                 3 = 'Single'           4 = 'Married';
run;

data Cars;
   missing a;
   input (Origin Size Type Home Income Marital Kids Sex) (1.) @@;
   * Check for End of Line;
   if n(of Origin -- Sex) eq 0 then do; input; return; end;
   marital = 2 * (kids le 0) + marital;
   format Origin Origin. Size Size. Type Type. Home Home.
          Sex Sex. Income Income. Marital Marital.;
   output;
   datalines;
13111221212111012111220113121101121122112211212113112212321122221221220 1
12112202312122123221110112212202212111012211210213111221112111011231110 1
21111211321122312112220222112211131112313121110232112222322122022122110 1

... more lines ...

21212201121112213122110112121102221222021212110 1
;

ods graphics on;

* Perform Simple Correspondence Analysis;
proc corresp all data=Cars outc=Coor;
   tables Marital, Origin;
run;
```

Correspondence analysis locates all the categories in a Euclidean space. The first two dimensions of this space are plotted to examine the associations among the categories. Since the smallest dimension of this table is three, there is no loss of information when only two dimensions are plotted. The plot should be thought of as two different overlaid plots, one for each categorical variable. Distances between points within a variable have meaning, but distances between points from different variables do not.

Output 30.1.1 Simple Correspondence Analysis

Automobile Owners and Auto Attributes
Simple Correspondence Analysis

The CORRESP Procedure

Contingency Table

	American	European	Japanese	Sum
Married	37	14	51	102
Married with Kids	52	15	44	111
Single	33	15	63	111
Single with Kids	6	1	8	15
Sum	128	45	166	339

Output 30.1.1 *continued*

Chi-Square Statistic Expected Values

	American	European	Japanese
Married	38.5133	13.5398	49.9469
Married with Kids	41.9115	14.7345	54.3540
Single	41.9115	14.7345	54.3540
Single with Kids	5.6637	1.9912	7.3451

Observed Minus Expected Values

	American	European	Japanese
Married	-1.5133	0.4602	1.0531
Married with Kids	10.0885	0.2655	-10.3540
Single	-8.9115	0.2655	8.6460
Single with Kids	0.3363	-0.9912	0.6549

Contributions to the Total Chi-Square Statistic

	American	European	Japanese	Sum
Married	0.05946	0.01564	0.02220	0.09730
Married with Kids	2.42840	0.00478	1.97235	4.40553
Single	1.89482	0.00478	1.37531	3.27492
Single with Kids	0.01997	0.49337	0.05839	0.57173
Sum	4.40265	0.51858	3.42825	8.34947

Row Profiles

	American	European	Japanese
Married	0.362745	0.137255	0.500000
Married with Kids	0.468468	0.135135	0.396396
Single	0.297297	0.135135	0.567568
Single with Kids	0.400000	0.066667	0.533333

Column Profiles

	American	European	Japanese
Married	0.289063	0.311111	0.307229
Married with Kids	0.406250	0.333333	0.265060
Single	0.257813	0.333333	0.379518
Single with Kids	0.046875	0.022222	0.048193

Output 30.1.1 *continued*

Automobile Owners and Auto Attributes
Simple Correspondence Analysis

The CORRESP Procedure

Inertia and Chi-Square Decomposition

Singular Value	Principal Inertia	Chi-Square	Percent	Cumulative Percent	19 38 57 76 95
0.15122	0.02287	7.75160	92.84	92.84	**********
0.04200	0.00176	0.59787	7.16	100.00	**
Total	0.02463	8.34947	100.00		

Degrees of Freedom = 6

Row Coordinates

	Dim1	Dim2
Married	-0.0278	0.0134
Married with Kids	0.1991	0.0064
Single	-0.1716	0.0076
Single with Kids	-0.0144	-0.1947

Summary Statistics for the Row Points

	Quality	Mass	Inertia
Married	1.0000	0.3009	0.0117
Married with Kids	1.0000	0.3274	0.5276
Single	1.0000	0.3274	0.3922
Single with Kids	1.0000	0.0442	0.0685

Partial Contributions to Inertia for the Row Points

	Dim1	Dim2
Married	0.0102	0.0306
Married with Kids	0.5678	0.0076
Single	0.4217	0.0108
Single with Kids	0.0004	0.9511

Indices of the Coordinates That Contribute Most to Inertia for the Row Points

	Dim1	Dim2	Best
Married	0	0	2
Married with Kids	1	0	1
Single	1	0	1
Single with Kids	0	2	2

Output 30.1.1 *continued*

Squared Cosines for the Row Points

	Dim1	Dim2
Married	0.8121	0.1879
Married with Kids	0.9990	0.0010
Single	0.9980	0.0020
Single with Kids	0.0054	0.9946

Column Coordinates

	Dim1	Dim2
American	0.1847	-0.0166
European	0.0013	0.1073
Japanese	-0.1428	-0.0163

Summary Statistics for the Column Points

	Quality	Mass	Inertia
American	1.0000	0.3776	0.5273
European	1.0000	0.1327	0.0621
Japanese	1.0000	0.4897	0.4106

Partial Contributions to Inertia for the Column Points

	Dim1	Dim2
American	0.5634	0.0590
European	0.0000	0.8672
Japanese	0.4366	0.0737

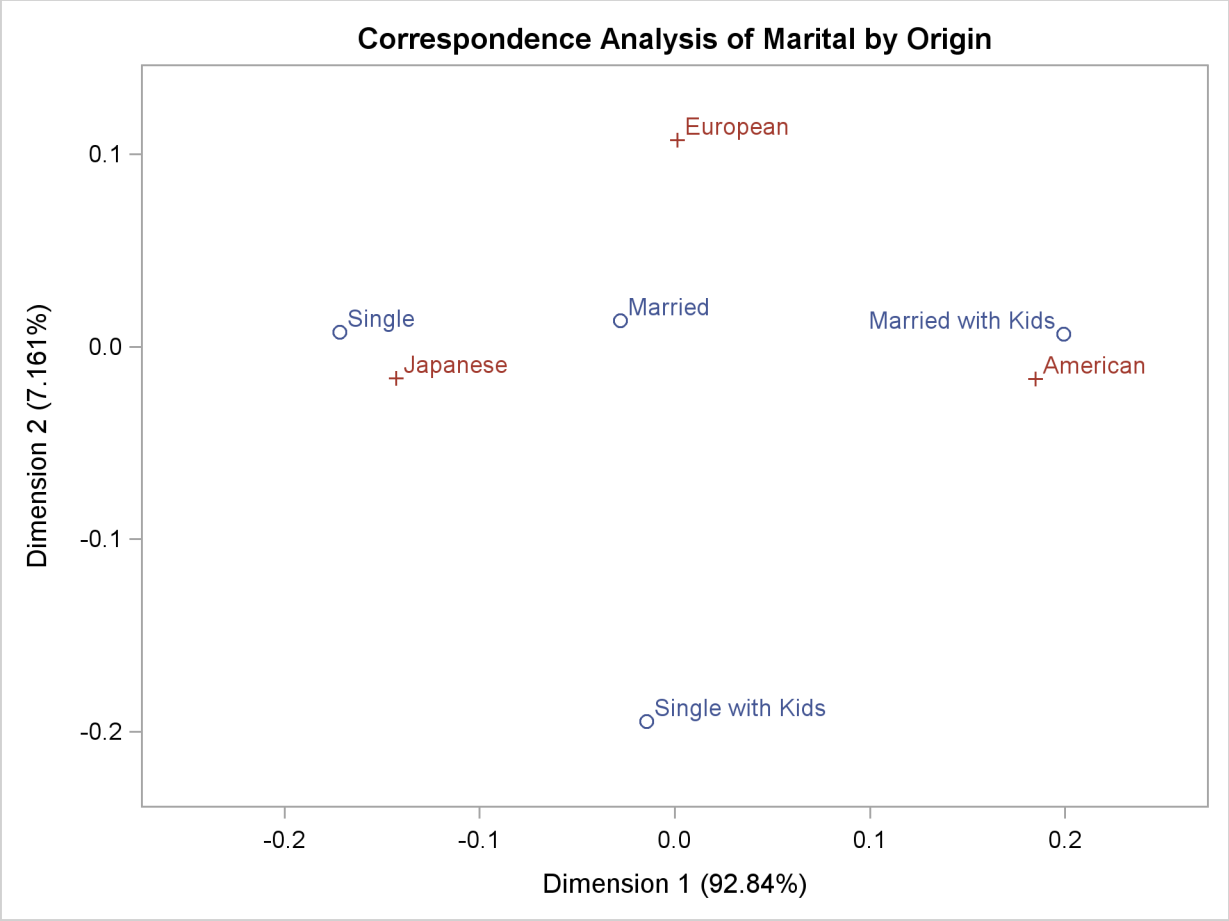
Indices of the Coordinates That Contribute Most to Inertia for the Column Points

	Dim1	Dim2	Best
American	1	0	1
European	0	2	2
Japanese	1	0	1

Squared Cosines for the Column Points

	Dim1	Dim2
American	0.9920	0.0080
European	0.0001	0.9999
Japanese	0.9871	0.0129

Output 30.1.1 *continued*



To interpret the plot, start by interpreting the row points separately from the column points. The European point is near and to the left of the centroid, so it makes a relatively small contribution to the chi-square statistic (because it is near the centroid), it contributes almost nothing to the inertia of dimension one (since its coordinate on dimension one has a small absolute value relative to the other column points), and it makes a relatively large contribution to the inertia of dimension two (since its coordinate on dimension two has a large absolute value relative to the other column points). Its squared cosines for dimension one and two, approximately 0 and 1, respectively, indicate that its position is almost completely determined by its location on dimension two. Its quality of display is 1.0, indicating perfect quality, since the table is two-dimensional after the centering. The American and Japanese points are far from the centroid, and they lie along dimension one. They make relatively large contributions to the chi-square statistic and the inertia of dimension one. The horizontal dimension seems to be largely determined by Japanese versus American automobile ownership.

In the row points, the Married point is near the centroid, and the Single with Kids point has a small coordinate on dimension one that is near zero. The horizontal dimension seems to be largely determined by the Single versus the Married with Kids points. The two interpretations of dimension one show the association with being Married with Kids and owning an American auto, and being single and owning a Japanese auto. The fact that the Married with Kids point is close to the American

point and the fact that the Japanese point is near the Single point should be ignored. Distances between row and column points are not defined. The plot shows that more people who are married with kids than you would expect if the rows and columns were independent drive an American auto, and more people who are single than you would expect if the rows and columns were independent drive a Japanese auto.

In the second part of this example, PROC CORRESP creates a Burt table from categorical data and performs a multiple correspondence analysis. The variables used in this example are Origin, Size, Type, Income, Home, Marital, and Sex. MCA specifies multiple correspondence analysis, OBSERVED displays the Burt table, and the OUTC= option creates an output coordinate data set. The TABLES statement with only a single variable list and no comma creates the Burt table.

The following statements produce Output 30.1.2:

```
title2 'Multiple Correspondence Analysis';

* Perform Multiple Correspondence Analysis;
proc corresp mca observed data=Cars outc=Coor;
   tables Origin Size Type Income Home Marital Sex;
run;

ods graphics off;
```

Output 30.1.2 Multiple Correspondence Analysis

Automobile Owners and Auto Attributes
Multiple Correspondence Analysis

The CORRESP Procedure

Burt Table

	American	European	Japanese	Large	Medium	Small	Family	Sporty	Work
American	125	0	0	36	60	29	81	24	20
European	0	44	0	4	20	20	17	23	4
Japanese	0	0	165	2	61	102	76	59	30
Large	36	4	2	42	0	0	30	1	11
Medium	60	20	61	0	141	0	89	39	13
Small	29	20	102	0	0	151	55	66	30
Family	81	17	76	30	89	55	174	0	0
Sporty	24	23	59	1	39	66	0	106	0
Work	20	4	30	11	13	30	0	0	54
1 Income	58	18	74	20	57	73	69	55	26
2 Incomes	67	26	91	22	84	78	105	51	28
Own	93	38	111	35	106	101	130	71	41
Rent	32	6	54	7	35	50	44	35	13
Married	37	13	51	9	42	50	50	35	16
Married with Kids	50	15	44	21	51	37	79	12	18
Single	32	15	62	11	40	58	35	57	17
Single with Kids	6	1	8	1	8	6	10	2	3
Female	58	21	70	17	70	62	83	44	22
Male	67	23	95	25	71	89	91	62	32

Burt Table

	1 Income	2 Incomes	Own	Rent	Married	Married with Kids	Single	Single with Kids	Female	Male
American	58	67	93	32	37	50	32	6	58	67
European	18	26	38	6	13	15	15	1	21	23
Japanese	74	91	111	54	51	44	62	8	70	95
Large	20	22	35	7	9	21	11	1	17	25
Medium	57	84	106	35	42	51	40	8	70	71
Small	73	78	101	50	50	37	58	6	62	89
Family	69	105	130	44	50	79	35	10	83	91
Sporty	55	51	71	35	35	12	57	2	44	62
Work	26	28	41	13	16	18	17	3	22	32
1 Income	150	0	80	70	10	27	99	14	47	103
2 Incomes	0	184	162	22	91	82	10	1	102	82
Own	80	162	242	0	76	106	52	8	114	128
Rent	70	22	0	92	25	3	57	7	35	57
Married	10	91	76	25	101	0	0	0	53	48
Married with Kids	27	82	106	3	0	109	0	0	48	61
Single	99	10	52	57	0	0	109	0	35	74
Single with Kids	14	1	8	7	0	0	0	15	13	2
Female	47	102	114	35	53	48	35	13	149	0
Male	103	82	128	57	48	61	74	2	0	185

Output 30.1.2 *continued*

Automobile Owners and Auto Attributes
Multiple Correspondence Analysis

The CORRESP Procedure

Inertia and Chi-Square Decomposition

Singular Value	Principal Inertia	Chi-Square	Percent	Cumulative Percent	4 8 12 16 20
0.56934	0.32415	970.77	18.91	18.91	**********
0.48352	0.23380	700.17	13.64	32.55	********
0.42716	0.18247	546.45	10.64	43.19	*******
0.41215	0.16987	508.73	9.91	53.10	*******
0.38773	0.15033	450.22	8.77	61.87	******
0.38520	0.14838	444.35	8.66	70.52	******
0.34066	0.11605	347.55	6.77	77.29	****
0.32983	0.10879	325.79	6.35	83.64	****
0.31517	0.09933	297.47	5.79	89.43	****
0.28069	0.07879	235.95	4.60	94.03	***
0.26115	0.06820	204.24	3.98	98.01	***
0.18477	0.03414	102.24	1.99	100.00	*
Total	1.71429	5133.92	100.00		

Degrees of Freedom = 324

Column Coordinates

	Dim1	Dim2
American	-0.4035	0.8129
European	-0.0568	-0.5552
Japanese	0.3208	-0.4678
Large	-0.6949	1.5666
Medium	-0.2562	0.0965
Small	0.4326	-0.5258
Family	-0.4201	0.3602
Sporty	0.6604	-0.6696
Work	0.0575	0.1539
1 Income	0.8251	0.5472
2 Incomes	-0.6727	-0.4461
Own	-0.3887	-0.0943
Rent	1.0225	0.2480
Married	-0.4169	-0.7954
Married with Kids	-0.8200	0.3237
Single	1.1461	0.2930
Single with Kids	0.4373	0.8736
Female	-0.3365	-0.2057
Male	0.2710	0.1656

Output 30.1.2 *continued*

Summary Statistics for the Column Points

	Quality	Mass	Inertia
American	0.4925	0.0535	0.0521
European	0.0473	0.0188	0.0724
Japanese	0.3141	0.0706	0.0422
Large	0.4224	0.0180	0.0729
Medium	0.0548	0.0603	0.0482
Small	0.3825	0.0646	0.0457
Family	0.3330	0.0744	0.0399
Sporty	0.4112	0.0453	0.0569
Work	0.0052	0.0231	0.0699
1 Income	0.7991	0.0642	0.0459
2 Incomes	0.7991	0.0787	0.0374
Own	0.4208	0.1035	0.0230
Rent	0.4208	0.0393	0.0604
Married	0.3496	0.0432	0.0581
Married with Kids	0.3765	0.0466	0.0561
Single	0.6780	0.0466	0.0561
Single with Kids	0.0449	0.0064	0.0796
Female	0.1253	0.0637	0.0462
Male	0.1253	0.0791	0.0372

Partial Contributions to Inertia for the Column Points

	Dim1	Dim2
American	0.0268	0.1511
European	0.0002	0.0248
Japanese	0.0224	0.0660
Large	0.0268	0.1886
Medium	0.0122	0.0024
Small	0.0373	0.0764
Family	0.0405	0.0413
Sporty	0.0610	0.0870
Work	0.0002	0.0023
1 Income	0.1348	0.0822
2 Incomes	0.1099	0.0670
Own	0.0482	0.0039
Rent	0.1269	0.0103
Married	0.0232	0.1169
Married with Kids	0.0967	0.0209
Single	0.1889	0.0171
Single with Kids	0.0038	0.0209
Female	0.0223	0.0115
Male	0.0179	0.0093

Output 30.1.2 *continued*

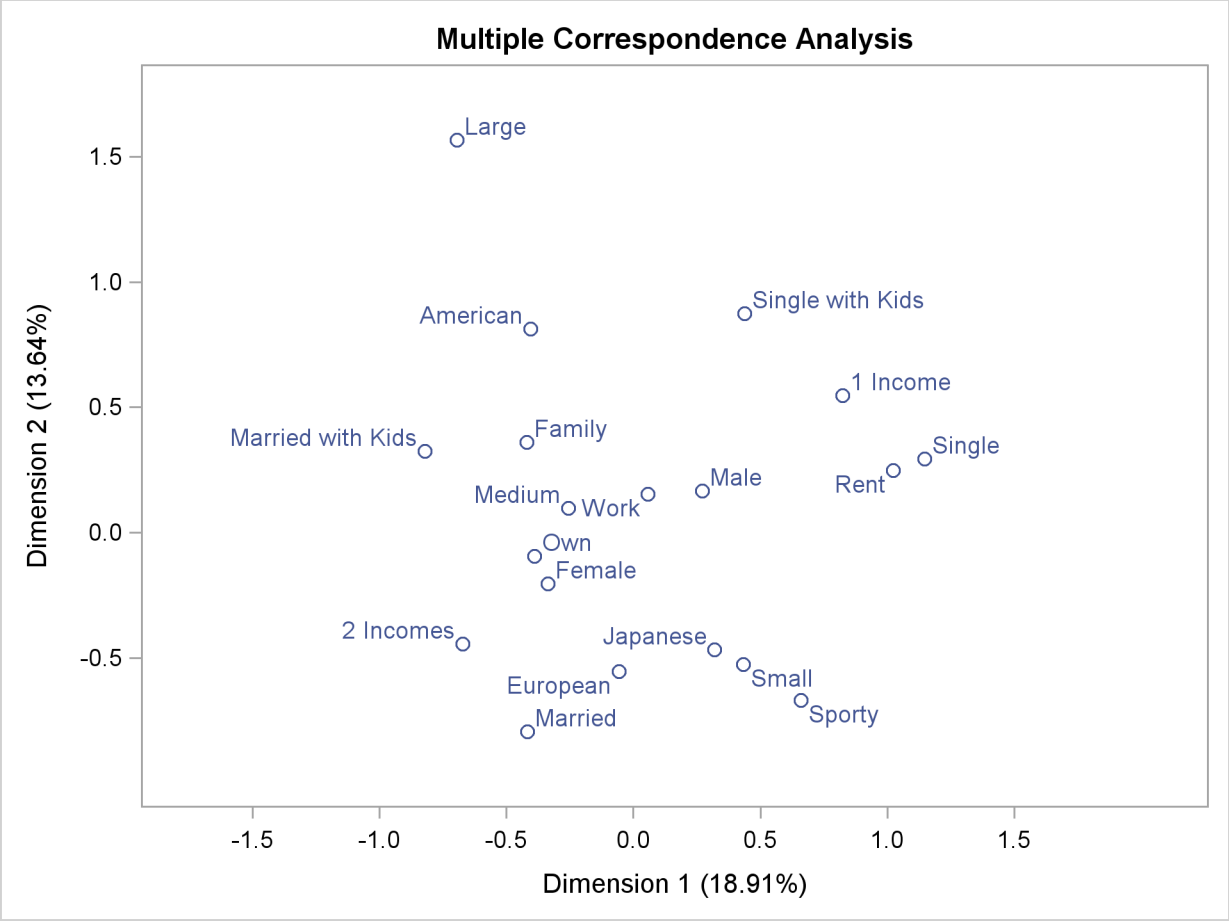
Indices of the Coordinates That Contribute Most to Inertia for the Column Points

	Dim1	Dim2	Best
American	0	2	2
European	0	0	2
Japanese	0	2	2
Large	0	2	2
Medium	0	0	1
Small	0	2	2
Family	2	0	2
Sporty	2	2	2
Work	0	0	2
1 Income	1	1	1
2 Incomes	1	1	1
Own	1	0	1
Rent	1	0	1
Married	0	2	2
Married with Kids	1	0	1
Single	1	0	1
Single with Kids	0	0	2
Female	0	0	1
Male	0	0	1

Squared Cosines for the Column Points

	Dim1	Dim2
American	0.0974	0.3952
European	0.0005	0.0468
Japanese	0.1005	0.2136
Large	0.0695	0.3530
Medium	0.0480	0.0068
Small	0.1544	0.2281
Family	0.1919	0.1411
Sporty	0.2027	0.2085
Work	0.0006	0.0046
1 Income	0.5550	0.2441
2 Incomes	0.5550	0.2441
Own	0.3975	0.0234
Rent	0.3975	0.0234
Married	0.0753	0.2742
Married with Kids	0.3258	0.0508
Single	0.6364	0.0416
Single with Kids	0.0090	0.0359
Female	0.0912	0.0341
Male	0.0912	0.0341

Output 30.1.2 *continued*



Multiple correspondence analysis locates all the categories in a Euclidean space. The first two dimensions of this space are plotted to examine the associations among the categories. The top-right quadrant of the plot shows that the categories Single, Single with Kids, 1 Income, and Rent are associated. Proceeding clockwise, the categories Sporty, Small, and Japanese are associated. The bottom-left quadrant shows the association between being married, owning your own home, and having two incomes. Having children is associated with owning a large American family auto. Such information could be used in market research to identify target audiences for advertisements.

This interpretation is based on points found in approximately the same direction from the origin and in approximately the same region of the space. Distances between points do not have a straightforward interpretation in multiple correspondence analysis. The geometry of multiple correspondence analysis is not a simple generalization of the geometry of simple correspondence analysis (Greenacre and Hastie 1987; Greenacre 1988).

If you want to perform a multiple correspondence analysis and get scores for the individuals, you can specify the BINARY option to analyze the binary table, as in the following statements. In the interest of space, only the first 10 rows of coordinates are printed in Output 30.1.3.

```
title2 'Binary Table';

* Perform Multiple Correspondence Analysis;
```

```
proc corresp data=Cars binary;
   ods select RowCoors;
   tables Origin Size Type Income Home Marital Sex;
run;
```

Output 30.1.3 Correspondence Analysis of a Binary Table

Automobile Owners and Auto Attributes
Binary Table

The CORRESP Procedure

Row Coordinates

	Dim1	Dim2
1	−0.4093	1.0878
2	0.8198	−0.2221
3	−0.2193	−0.5328
4	0.4382	1.1799
5	−0.6750	0.3600
6	−0.1778	0.1441
7	−0.9375	0.6846
8	−0.7405	−0.1539
9	−0.3027	−0.2749
10	−0.7263	−0.0803

Example 30.2: Simple Correspondence Analysis of U.S. Population

In this example, PROC CORRESP reads an existing contingency table with supplementary observations and performs a simple correspondence analysis. The data are populations of the 50 U.S. states, grouped into regions, for each of the census years from 1920 to 1970 (U.S. Bureau of the Census 1979). Alaska and Hawaii are treated as supplementary regions, because they were not states during this entire period and are not physically connected to the other 48 states. Consequently, it is reasonable to expect that population changes in these two states operate differently from population changes in the other states. The correspondence analysis is performed giving the supplementary points negative weight, and then the coordinates for the supplementary points are computed in the solution defined by the other points.

The initial DATA step reads the table, provides labels for the years, flags the supplementary rows with negative weights, and specifies absolute weights of 1000 for all observations since the data were originally reported in units of 1000 people.

In the PROC CORRESP statement, PRINT=PERCENT and the display options display the table of cell percentages (OBSERVED), cell contributions to the total chi-square scaled to sum to 100 (CELLCHI2), row profile rows that sum to 100 (RP), and column profile columns that sum to 100 (CP). The SHORT option specifies that the correspondence analysis summary statistics, contributions to inertia, and squared cosines should not be displayed. The option OUTC=COOR creates the

output coordinate data set. Since the data are already in table form, a VAR statement is used to read the table. Row labels are specified with the ID statement, and column labels come from the variable labels. The WEIGHT statement flags the supplementary observations and restores the table values to populations.

The following statements produce Output 30.2.1:

```
title 'United States Population, 1920-1970';

data USPop;

   * Regions:
   * New England     - ME, NH, VT, MA, RI, CT.
   * Great Lakes     - OH, IN, IL, MI, WI.
   * South Atlantic  - DE, MD, DC, VA, WV, NC, SC, GA, FL.
   * Mountain        - MT, ID, WY, CO, NM, AZ, UT, NV.
   * Pacific         - WA, OR, CA.
   *
   * Note: Multiply data values by 1000 to get populations.;

   input Region $14. y1920 y1930 y1940 y1950 y1960 y1970;

   label y1920 = '1920'    y1930 = '1930'    y1940 = '1940'
         y1950 = '1950'    y1960 = '1960'    y1970 = '1970';

   if region = 'Hawaii' or region = 'Alaska'
      then w = -1000;       /* Flag Supplementary Observations */
      else w =  1000;

   datalines;
New England        7401  8166  8437  9314 10509 11842
NY, NJ, PA        22261 26261 27539 30146 34168 37199
Great Lakes       21476 25297 26626 30399 36225 40252
Midwest           12544 13297 13517 14061 15394 16319
South Atlantic    13990 15794 17823 21182 25972 30671
KY, TN, AL, MS     8893  9887 10778 11447 12050 12803
AR, LA, OK, TX    10242 12177 13065 14538 16951 19321
Mountain           3336  3702  4150  5075  6855  8282
Pacific            5567  8195  9733 14486 20339 25454
Alaska               55    59    73   129   226   300
Hawaii              256   368   423   500   633   769
;
ods graphics on;

* Perform Simple Correspondence Analysis;
proc corresp data=uspop print=percent observed cellchi2 rp cp
     short outc=Coor plot(flip);
   var y1920 -- y1970;
   id Region;
   weight w;
run;

ods graphics off;
```

The contingency table shows that the population of all regions increased over this time period. The row profiles show that population increased at a different rate for the different regions. There was a small increase in population in the Midwest, for example, but the population more than quadrupled in the Pacific region over the same period. The column profiles show that in 1920, the U.S. population was concentrated in the NY, NJ, PA, Great Lakes, Midwest, and South Atlantic regions. With time, the population shifted more to the South Atlantic, Mountain, and Pacific regions. This is also clear from the correspondence analysis. The inertia and chi-square decomposition table shows that there are five nontrivial dimensions in the table, but the association between the rows and columns is almost entirely one-dimensional.

Output 30.2.1 United States Population, 1920–1970

United States Population, 1920-1970

The CORRESP Procedure

Contingency Table

Percents	1920	1930	1940	1950	1960	1970	Sum
New England	0.830	0.916	0.946	1.045	1.179	1.328	6.245
NY, NJ, PA	2.497	2.946	3.089	3.382	3.833	4.173	19.921
Great Lakes	2.409	2.838	2.987	3.410	4.064	4.516	20.224
Midwest	1.407	1.492	1.516	1.577	1.727	1.831	9.550
South Atlantic	1.569	1.772	1.999	2.376	2.914	3.441	14.071
KY, TN, AL, MS	0.998	1.109	1.209	1.284	1.352	1.436	7.388
AR, LA, OK, TX	1.149	1.366	1.466	1.631	1.902	2.167	9.681
Mountain	0.374	0.415	0.466	0.569	0.769	0.929	3.523
Pacific	0.625	0.919	1.092	1.625	2.282	2.855	9.398
Sum	11.859	13.773	14.771	16.900	20.020	22.677	100.000

Supplementary Rows

Percents	1920	1930	1940	1950	1960	1970
Alaska	0.006170	0.006619	0.008189	0.014471	0.025353	0.033655
Hawaii	0.028719	0.041283	0.047453	0.056091	0.071011	0.086268

Contributions to the Total Chi-Square Statistic

Percents	1920	1930	1940	1950	1960	1970	Sum
New England	0.937	0.314	0.054	0.009	0.352	0.469	2.135
NY, NJ, PA	0.665	1.287	0.633	0.006	0.521	2.265	5.378
Great Lakes	0.004	0.085	0.000	0.001	0.005	0.094	0.189
Midwest	5.749	2.039	0.684	0.072	1.546	4.472	14.563
South Atlantic	0.509	1.231	0.259	0.000	0.285	1.688	3.973
KY, TN, AL, MS	1.454	0.711	1.098	0.087	0.946	2.945	7.242
AR, LA, OK, TX	0.000	0.069	0.077	0.001	0.059	0.030	0.238
Mountain	0.391	0.868	0.497	0.098	0.498	1.834	4.187
Pacific	18.591	9.380	5.458	0.074	7.346	21.248	62.096
Sum	28.302	15.986	8.761	0.349	11.558	35.046	100.000

Output 30.2.1 *continued*

Row Profiles

Percents	1920	1930	1940	1950	1960	1970
New England	13.2947	14.6688	15.1557	16.7310	18.8777	21.2722
NY, NJ, PA	12.5362	14.7888	15.5085	16.9766	19.2416	20.9484
Great Lakes	11.9129	14.0325	14.7697	16.8626	20.0943	22.3281
Midwest	14.7348	15.6193	15.8777	16.5167	18.0825	19.1691
South Atlantic	11.1535	12.5917	14.2093	16.8872	20.7060	24.4523
KY, TN, AL, MS	13.5033	15.0126	16.3655	17.3813	18.2969	19.4403
AR, LA, OK, TX	11.8687	14.1111	15.1401	16.8471	19.6433	22.3897
Mountain	10.6242	11.7898	13.2166	16.1624	21.8312	26.3758
Pacific	6.6453	9.7823	11.6182	17.2918	24.2784	30.3841

Supplementary Row Profiles

Percents	1920	1930	1940	1950	1960	1970
Alaska	6.5321	7.0071	8.6698	15.3207	26.8409	35.6295
Hawaii	8.6809	12.4788	14.3438	16.9549	21.4649	26.0766

Column Profiles

Percents	1920	1930	1940	1950	1960	1970
New England	7.0012	6.6511	6.4078	6.1826	5.8886	5.8582
NY, NJ, PA	21.0586	21.3894	20.9155	20.0109	19.1457	18.4023
Great Lakes	20.3160	20.6042	20.2221	20.1788	20.2983	19.9126
Midwest	11.8664	10.8303	10.2660	9.3337	8.6259	8.0730
South Atlantic	13.2343	12.8641	13.5363	14.0606	14.5532	15.1729
KY, TN, AL, MS	8.4126	8.0529	8.1857	7.5985	6.7521	6.3336
AR, LA, OK, TX	9.6888	9.9181	9.9227	9.6503	9.4983	9.5581
Mountain	3.1558	3.0152	3.1519	3.3688	3.8411	4.0971
Pacific	5.2663	6.6748	7.3921	9.6158	11.3968	12.5921

United States Population, 1920–1970

The CORRESP Procedure

Inertia and Chi-Square Decomposition

Singular Value	Principal Inertia	Chi-Square	Percent	Cumulative Percent	20 40 60 80 100
0.10664	0.01137	1.014E7	98.16	98.16	**********
0.01238	0.00015	136586	1.32	99.48	
0.00658	0.00004	38540	0.37	99.85	
0.00333	0.00001	9896.6	0.10	99.95	
0.00244	0.00001	5309.9	0.05	100.00	
Total	0.01159	1.033E7	100.00		

Degrees of Freedom = 40

Output 30.2.1 *continued*

Row Coordinates

	Dim1	Dim2
New England	0.0611	0.0132
NY, NJ, PA	0.0546	−0.0117
Great Lakes	0.0074	−0.0028
Midwest	0.1315	0.0186
South Atlantic	−0.0553	0.0105
KY, TN, AL, MS	0.1044	−0.0144
AR, LA, OK, TX	0.0131	−0.0067
Mountain	−0.1121	0.0338
Pacific	−0.2766	−0.0070

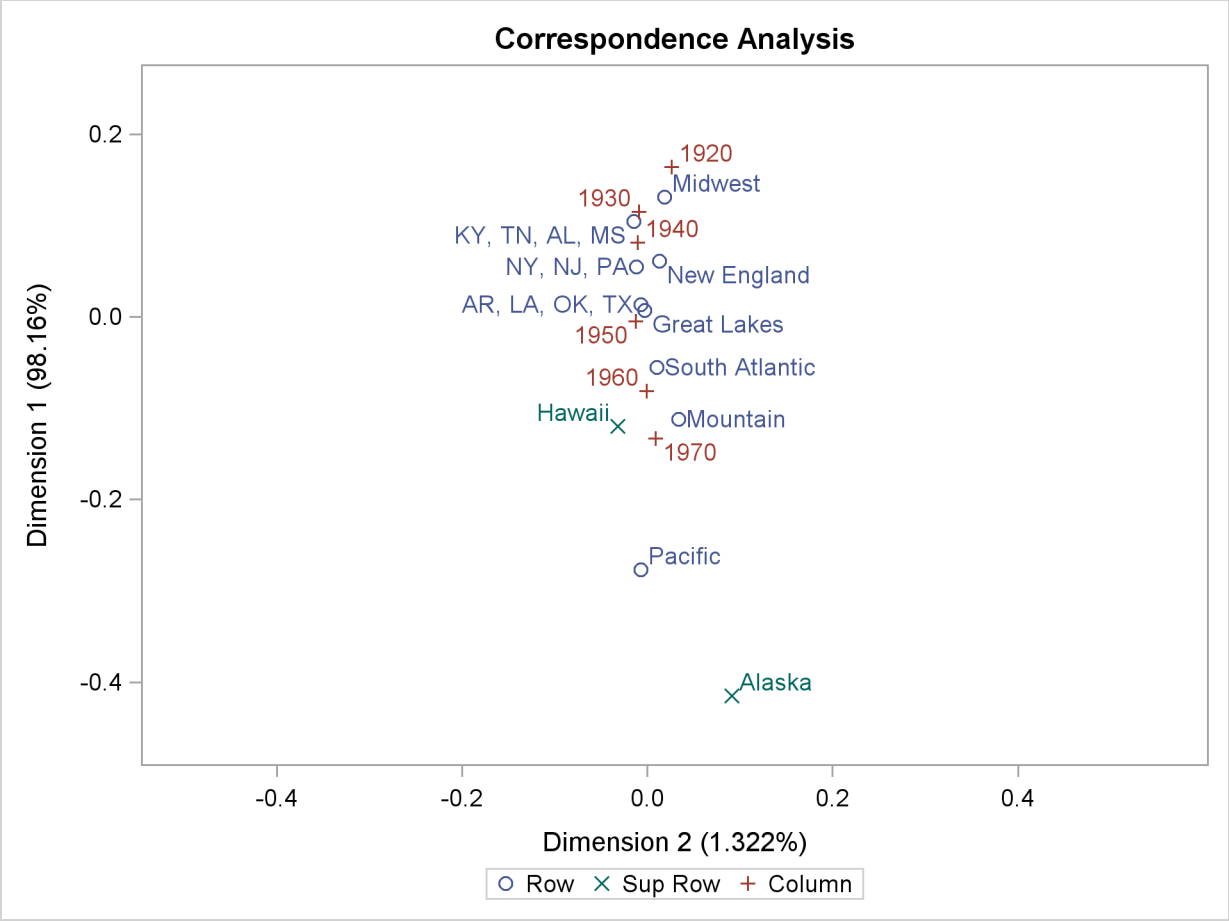
Supplementary Row Coordinates

	Dim1	Dim2
Alaska	−0.4152	0.0912
Hawaii	−0.1198	−0.0321

Column Coordinates

	Dim1	Dim2
1920	0.1642	0.0263
1930	0.1149	−0.0089
1940	0.0816	−0.0108
1950	−0.0046	−0.0125
1960	−0.0815	−0.0007
1970	−0.1335	0.0086

Output 30.2.1 *continued*



ODS Graphics is used to plot the results. The data are essentially one-dimensional. For data such as these, it is better to plot the first dimension vertically, as opposed to the default, which is horizontally. The vertical orientation has fewer opportunities for label collisions. Specifying PLOTS(FLIP) on the PROC statement switches the vertical and horizontal axes to improve the graphical display.

The plot shows that the first dimension correctly orders the years. There is nothing in the correspondence analysis that forces this to happen; the analysis has no information about the inherent ordering of the column categories. The ordering of the regions and the ordering of the years reflect the shift over time of the U.S. population from the Northeast quadrant of the country to the South and to the West. The results show that the West and Southeast grew faster than the rest of the contiguous 48 states during this period.

The plot also shows that the growth pattern for Hawaii was similar to the growth pattern for the mountain states and that Alaska's growth was even more extreme than the Pacific states' growth. The row profiles confirm this interpretation.

The Pacific region is farther from the origin than all other active points. The Midwest is the extreme region in the other direction. The table of contributions to the total chi-square shows that 62% of the total chi-square statistic is contributed by the Pacific region, which is followed by the Midwest at over 14%. Similarly the two extreme years, 1920 and 1970, together contribute over 63% to the total chi-square, whereas the years nearer the origin of the plot contribute less.

References

Benzécri, J. P. (1973), *L'Analyse des Données: T. 2, l'Analyse des Correspondances*, Paris: Dunod.

Benzécri, J. P. (1979), *Sur le Calcul des taux d'inertie dans l'analyse d'un questionaire*, Addendum et erratum á [BIN.MULT.], *Cahiers de l'Analyse des Données* 4, 377–378.

Burt, C. (1950), "The Factorial Analysis of Qualitative Data," *British Journal of Psychology*, 3, 166–185.

Carroll, J., Green, P. E., and Schaffer, C. M. (1986), "Interpoint Distance Comparisons in Correspondence Analysis," *Journal of Marketing Research*, 23, 271–280.

Fisher, R. A. (1940), "The Precision of Discriminant Functions," *Annals of Eugenics*, 10, 422–429.

Gifi, A. (1990), *Nonlinear Multivariate Analysis*, New York: John Wiley & Sons.

Greenacre, M. J. (1984), *Theory and Applications of Correspondence Analysis*, London: Academic Press.

Greenacre, M. J. (1988), "Correspondence Analysis of Multivariate Categorical Data by Weighted Least-Squares," *Biometrika*, 75, 457–467.

Greenacre, M. J. (1989), "The Carroll-Green-Schaffer Scaling in Correspondence Analysis: A Theoretical and Empirical Appraisal," *Journal of Market Research*, 26, 358–365.

Greenacre, M. J. (1994), "Multiple and Joint Correspondence Analysis," in M. J. Greenacre and J. Blasius, eds., *Correspondence Analysis in the Social Sciences*, London: Academic Press.

Greenacre, M. J. and Hastie, T. (1987), "The Geometric Interpretation of Correspondence Analysis," *Journal of the American Statistical Association*, 82, 437–447.

Guttman, L. (1941), "The Quantification of a Class of Attributes: A Theory and Method of Scale Construction," in P. Horst, P. Wallin, and L. Guttman, eds., *The Prediction of Personal Adjustment*, New York: Social Science Research Council.

Hayashi, C. (1950), "On the Quantification of Qualitative Data from the Mathematico-Statistical Point of View," *Annals of the Institute of Statistical Mathematics*, 2, 35–47.

Hirshfield, H. O. (1935), "A Connection between Correlation and Contingency," *Cambridge Philosophical Society Proceedings*, 31, 520–524.

Hoffman, D. L. and Franke, G. R. (1986), "Correspondence Analysis: Graphical Representation of Categorical Data in Marketing Research," *Journal of Marketing Research*, 23, 213–227.

Horst, P. (1935), "Measuring Complex Attitudes," *Journal of Social Psychology*, 6, 369–374.

Kobayashi, R. (1981), *An Introduction to Quantification Theory*, Tokyo: Japan Union of Scientists and Engineers.

Komazawa, T. (1982), *Quantification Theory and Data Processing*, Tokyo: Asakura-shoten.

Lebart, L., Morineau, A., and Tabard, N. (1977), *Techniques de la Description Statistique*, Paris: Dunod.

Lebart, L., Morineau, A., and Warwick, K. M. (1984), *Multivariate Descriptive Statistical Analysis: Correspondence Analysis and Related Techniques for Large Matrices*, New York: John Wiley & Sons.

Nishisato, S. (1980), *Analysis of Categorical Data: Dual Scaling and Its Applications*, Toronto: University of Toronto Press.

Nishisato, S. (1982), *Quantification of Qualitative Data - Dual Scaling and Its Applications*, Tokyo: Asakura-shoten.

Richardson, M. and Kuder, G. F. (1933), "Making a Rating Scale That Measures," *Personnel Journal*, 12, 36–40.

Tenenhaus, M. and Young, F. W. (1985), "An Analysis and Synthesis of Multiple Correspondence Analysis, Optimal Scaling, Dual Scaling, Homogeneity Analysis, and Other Methods of Quantifying Categorical Multivariate Data," *Psychometrika*, 50, 91–119.

U.S. Bureau of the Census (1979), *Statistical Abstract of the United States*, 100th Edition, Washington DC.

van der Heijden, P. G. M. and de Leeuw, J. (1985), "Correspondence Analysis Used Complementary to Loglinear Analysis," *Psychometrika*, 50, 429–447.

Chapter 31

The DISCRIM Procedure

Contents

Overview: DISCRIM Procedure **1384**
Getting Started: DISCRIM Procedure **1385**
Syntax: DISCRIM Procedure **1389**
PROC DISCRIM Statement 1389
BY Statement 1397
CLASS Statement 1398
FREQ Statement 1398
ID Statement 1398
PRIORS Statement 1399
TESTCLASS Statement 1400
TESTFREQ Statement 1400
TESTID Statement 1400
VAR Statement 1400
WEIGHT Statement 1401
Details: DISCRIM Procedure **1401**
Missing Values 1401
Background 1401
Posterior Probability Error-Rate Estimates 1410
Saving and Using Calibration Information 1411
Input Data Sets 1412
Output Data Sets 1414
Computational Resources 1418
Displayed Output 1419
ODS Table Names 1423
Examples: DISCRIM Procedure **1425**
Example 31.1: Univariate Density Estimates and Posterior Probabilities . . 1425
Example 31.2: Bivariate Density Estimates and Posterior Probabilities . . . 1441
Example 31.3: Normal-Theory Discriminant Analysis of Iris Data 1460
Example 31.4: Linear Discriminant Analysis of Remote-Sensing Data on Crops 1469
References **1480**

Overview: DISCRIM Procedure

For a set of observations containing one or more quantitative variables and a classification variable defining groups of observations, the DISCRIM procedure develops a discriminant criterion to classify each observation into one of the groups. The derived discriminant criterion from this data set can be applied to a second data set during the same execution of PROC DISCRIM. The data set that PROC DISCRIM uses to derive the discriminant criterion is called the *training* or *calibration* data set.

When the distribution within each group is assumed to be multivariate normal, a parametric method can be used to develop a discriminant function. The discriminant function, also known as a classification criterion, is determined by a measure of generalized squared distance (Rao 1973). The classification criterion can be based on either the individual within-group covariance matrices (yielding a quadratic function) or the pooled covariance matrix (yielding a linear function); it also takes into account the prior probabilities of the groups. The calibration information can be stored in a special SAS data set and applied to other data sets.

When no assumptions can be made about the distribution within each group, or when the distribution is assumed not to be multivariate normal, nonparametric methods can be used to estimate the group-specific densities. These methods include the kernel and k-nearest-neighbor methods (Rosenblatt 1956; Parzen 1962). The DISCRIM procedure uses uniform, normal, Epanechnikov, biweight, or triweight kernels for density estimation.

Either Mahalanobis or Euclidean distance can be used to determine proximity. Mahalanobis distance can be based on either the full covariance matrix or the diagonal matrix of variances. With a k-nearest-neighbor method, the pooled covariance matrix is used to calculate the Mahalanobis distances. With a kernel method, either the individual within-group covariance matrices or the pooled covariance matrix can be used to calculate the Mahalanobis distances. With the estimated group-specific densities and their associated prior probabilities, the posterior probability estimates of group membership for each class can be evaluated.

Canonical discriminant analysis is a dimension-reduction technique related to principal component analysis and canonical correlation. Given a classification variable and several quantitative variables, PROC DISCRIM derives canonical variables (linear combinations of the quantitative variables) that summarize between-class variation in much the same way that principal components summarize total variation. (See Chapter 27, "The CANDISC Procedure," for more information about canonical discriminant analysis.) A discriminant criterion is always derived in PROC DISCRIM. If you want canonical discriminant analysis without the use of a discriminant criterion, you should use the CANDISC procedure.

The DISCRIM procedure can produce an output data set containing various statistics such as means, standard deviations, and correlations. If a parametric method is used, the discriminant function is also stored in the data set to classify future observations. When canonical discriminant analysis is performed, the output data set includes canonical coefficients that can be rotated by the FACTOR procedure. PROC DISCRIM can also create a second type of output data set containing the classification results for each observation. When canonical discriminant analysis is performed, this output data set also includes canonical variable scores. A third type of output data set containing

the group-specific density estimates at each observation can also be produced.

PROC DISCRIM evaluates the performance of a discriminant criterion by estimating error rates (probabilities of misclassification) in the classification of future observations. These error-rate estimates include error-count estimates and posterior probability error-rate estimates. When the input data set is an ordinary SAS data set, the error rate can also be estimated by cross validation.

Do not confuse discriminant analysis with cluster analysis. All varieties of discriminant analysis require prior knowledge of the classes, usually in the form of a sample from each class. In cluster analysis, the data do not include information about class membership; the purpose is to construct a classification.

See Chapter 10, "Introduction to Discriminant Procedures," for a discussion of discriminant analysis and the SAS/STAT procedures available.

Getting Started: DISCRIM Procedure

The data in this example are measurements of 159 fish caught in Finland's lake Laengelmavesi; this data set is available from the Puranen. For each of the seven species (bream, roach, whitefish, parkki, perch, pike, and smelt) the weight, length, height, and width of each fish are tallied. Three different length measurements are recorded: from the nose of the fish to the beginning of its tail, from the nose to the notch of its tail, and from the nose to the end of its tail. The height and width are recorded as percentages of the third length variable. The goal now is to find a discriminant function based on these six variables that best classifies the fish into species.

First, assume that the data are normally distributed within each group with equal covariances across groups. The following statements use PROC DISCRIM to analyze the Fish data and create Figure 31.1 through Figure 31.5:

```
title 'Fish Measurement Data';

proc format;
   value specfmt
      1='Bream'
      2='Roach'
      3='Whitefish'
      4='Parkki'
      5='Perch'
      6='Pike'
      7='Smelt';
run;

data fish (drop=HtPct WidthPct);
   input Species Weight Length1 Length2 Length3 HtPct
         WidthPct @@;
   Height=HtPct*Length3/100;
   Width=WidthPct*Length3/100;
   format Species specfmt.;
```

```
   datalines;
1  242.0 23.2 25.4 30.0 38.4 13.4 1  290.0 24.0 26.3 31.2 40.0 13.8
1  340.0 23.9 26.5 31.1 39.8 15.1 1  363.0 26.3 29.0 33.5 38.0 13.3

... more lines ...

7   19.7 13.2 14.3 15.2 18.9 13.6 7   19.9 13.8 15.0 16.2 18.1 11.6
;

proc discrim data=fish;
   class Species;
run;
```

The DISCRIM procedure begins by displaying summary information about the variables in the analysis (see Figure 31.1). This information includes the number of observations, the number of quantitative variables in the analysis (specified with the VAR statement), and the number of classes in the classification variable (specified with the CLASS statement). The frequency of each class, its weight, the proportion of the total sample, and the prior probability are also displayed. Equal priors are assigned by default.

Figure 31.1 Summary Information

Fish Measurement Data

The DISCRIM Procedure

Total Sample Size	158	DF Total	157
Variables	6	DF Within Classes	151
Classes	7	DF Between Classes	6

Number of Observations Read	159
Number of Observations Used	158

Class Level Information

Species	Variable Name	Frequency	Weight	Proportion	Prior Probability
Bream	Bream	34	34.0000	0.215190	0.142857
Parkki	Parkki	11	11.0000	0.069620	0.142857
Perch	Perch	56	56.0000	0.354430	0.142857
Pike	Pike	17	17.0000	0.107595	0.142857
Roach	Roach	20	20.0000	0.126582	0.142857
Smelt	Smelt	14	14.0000	0.088608	0.142857
Whitefish	Whitefish	6	6.0000	0.037975	0.142857

The natural log of the determinant of the pooled covariance matrix is displayed in Figure 31.2.

Figure 31.2 Pooled Covariance Matrix Information

Pooled Covariance Matrix Information

Covariance Matrix Rank	Natural Log of the Determinant of the Covariance Matrix
6	4.17613

The squared distances between the classes are shown in Figure 31.3.

Figure 31.3 Squared Distances

Fish Measurement Data

The DISCRIM Procedure

Generalized Squared Distance to Species

From Species	Bream	Parkki	Perch	Pike	Roach	Smelt	Whitefish
Bream	0	83.32523	243.66688	310.52333	133.06721	252.75503	132.05820
Parkki	83.32523	0	57.09760	174.20918	27.00096	60.52076	26.54855
Perch	243.66688	57.09760	0	101.06791	29.21632	29.26806	20.43791
Pike	310.52333	174.20918	101.06791	0	92.40876	127.82177	99.90673
Roach	133.06721	27.00096	29.21632	92.40876	0	33.84280	6.31997
Smelt	252.75503	60.52076	29.26806	127.82177	33.84280	0	46.37326
Whitefish	132.05820	26.54855	20.43791	99.90673	6.31997	46.37326	0

The coefficients of the linear discriminant function are displayed (in Figure 31.4) with the default options METHOD=NORMAL and POOL=YES.

Figure 31.4 Linear Discriminant Function

Linear Discriminant Function for Species

Variable	Bream	Parkki	Perch	Pike	Roach	Smelt	Whitefish
Constant	-185.91682	-64.92517	-48.68009	-148.06402	-62.65963	-19.70401	-67.44603
Weight	-0.10912	-0.09031	-0.09418	-0.13805	-0.09901	-0.05778	-0.09948
Length1	-23.02273	-13.64180	-19.45368	-20.92442	-14.63635	-4.09257	-22.57117
Length2	-26.70692	-5.38195	17.33061	6.19887	-7.47195	-3.63996	3.83450
Length3	50.55780	20.89531	5.25993	22.94989	25.00702	10.60171	21.12638
Height	13.91638	8.44567	-1.42833	-8.99687	-0.26083	-1.84569	0.64957
Width	-23.71895	-13.38592	1.32749	-9.13410	-3.74542	-3.43630	-2.52442

A summary of how the discriminant function classifies the data used to develop the function is displayed last. In Figure 31.5, you see that only three of the observations are misclassified. The error-count estimates give the proportion of misclassified observations in each group. Since you

are classifying the same data that are used to derive the discriminant function, these error-count estimates are biased.

Figure 31.5 Resubstitution Misclassification Summary

Fish Measurement Data

The DISCRIM Procedure
Classification Summary for Calibration Data: WORK.FISH
Resubstitution Summary using Linear Discriminant Function

Number of Observations and Percent Classified into Species

From Species	Bream	Parkki	Perch	Pike	Roach	Smelt	Whitefish	Total
Bream	34	0	0	0	0	0	0	34
	100.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00
Parkki	0	11	0	0	0	0	0	11
	0.00	100.00	0.00	0.00	0.00	0.00	0.00	100.00
Perch	0	0	53	0	0	3	0	56
	0.00	0.00	94.64	0.00	0.00	5.36	0.00	100.00
Pike	0	0	0	17	0	0	0	17
	0.00	0.00	0.00	100.00	0.00	0.00	0.00	100.00
Roach	0	0	0	0	20	0	0	20
	0.00	0.00	0.00	0.00	100.00	0.00	0.00	100.00
Smelt	0	0	0	0	0	14	0	14
	0.00	0.00	0.00	0.00	0.00	100.00	0.00	100.00
Whitefish	0	0	0	0	0	0	6	6
	0.00	0.00	0.00	0.00	0.00	0.00	100.00	100.00
Total	34	11	53	17	20	17	6	158
	21.52	6.96	33.54	10.76	12.66	10.76	3.80	100.00
Priors	0.14286	0.14286	0.14286	0.14286	0.14286	0.14286	0.14286	

Error Count Estimates for Species

	Bream	Parkki	Perch	Pike	Roach	Smelt	Whitefish	Total
Rate	0.0000	0.0000	0.0536	0.0000	0.0000	0.0000	0.0000	0.0077
Priors	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429	0.1429	

One way to reduce the bias of the error-count estimates is to split your data into two sets. One set is used to derive the discriminant function, and the other set is used to run validation tests. Example 31.4 shows how to analyze a test data set. Another method of reducing bias is to classify each observation by using a discriminant function computed from all of the other observations; this method is invoked with the CROSSVALIDATE option.

Syntax: DISCRIM Procedure

The following statements are available in PROC DISCRIM:

PROC DISCRIM < *options* > ;
CLASS *variable* ;
BY *variables* ;
FREQ *variable* ;
ID *variable* ;
PRIORS *probabilities* ;
TESTCLASS *variable* ;
TESTFREQ *variable* ;
TESTID *variable* ;
VAR *variables* ;
WEIGHT *variable* ;

Only the PROC DISCRIM and CLASS statements are required.

The following sections describe the PROC DISCRIM statement and then describe the other statements in alphabetical order.

PROC DISCRIM Statement

PROC DISCRIM < *options* > ;

The PROC DISCRIM statement invokes the DISCRIM procedure. The options listed in Table 31.1 are available in the PROC DISCRIM statement.

Table 31.1 Options Available in the PROC DISCRIM Statement

Option	Description
Input Data Sets	
DATA=	specifies input SAS data set
TESTDATA=	specifies input SAS data set to classify
Output Data Sets	
OUTSTAT=	specifies output statistics data set
OUT=	specifies output data set with classification results
OUTCROSS=	specifies output data set with cross validation results
OUTD=	specifies output data set with densities
TESTOUT=	specifies output data set with TEST= results
TESTOUTD=	specifies output data set with TEST= densities
Method Details	
METHOD=	specifies parametric or nonparametric method
POOL=	specifies whether to pool the covariance matrices

Table 31.1 *continued*

Option	Description
SINGULAR=	specifies the singularity criterion
SLPOOL=	specifies significance level homogeneity test
THRESHOLD=	specifies the minimum threshold for classification
Nonparametric Methods	
K=	specifies k value for k nearest neighbors
KPROP=	specifies proportion, p, for computing k
R=	specifies radius for kernel density estimation
KERNEL=	specifies a kernel density to estimate
METRIC=	specifies metric in for squared distances
Canonical Discriminant Analysis	
CANONICAL	performs canonical discriminant analysis
CANPREFIX=	specifies a prefix for naming the canonical variables
NCAN=	specifies the number of canonical variables
Resubstitution Classification	
LIST	displays the classification results
LISTERR	displays the misclassified observations
NOCLASSIFY	suppresses the classification
TESTLIST	displays the classification results of TEST=
TESTLISTERR	displays the misclassified observations of TEST=
Cross Validation Classification	
CROSSLIST	displays the cross validation results
CROSSLISTERR	displays the misclassified cross validation results
CROSSVALIDATE	specifies cross validation
Control Displayed Output	
ALL	displays all output
ANOVA	displays univariate statistics
BCORR	displays between correlations
BCOV	displays between covariances
BSSCP	displays between SSCPs
DISTANCE	displays squared Mahalanobis distances
MANOVA	displays multivariate ANOVA results
NOPRINT	suppresses all displayed output
PCORR	displays pooled correlations
PCOV	displays pooled covariances
POSTERR	displays posterior probability error-rate estimates
PSSCP	displays pooled SSCPs
SHORT	suppresses some displayed output
SIMPLE	displays simple descriptive statistics
STDMEAN	displays standardized class means
TCORR	displays total correlations
TCOV	displays total covariances
TSSCP	displays total SSCPs

Table 31.1 *continued*

Option	Description
WCORR	displays within correlations
WCOV	displays within covariances
WSSCP	displays within SSCPs

ALL

activates all options that control displayed output. When the derived classification criterion is used to classify observations, the ALL option also activates the POSTERR option.

ANOVA

displays univariate statistics for testing the hypothesis that the class means are equal in the population for each variable.

BCORR

displays between-class correlations.

BCOV

displays between-class covariances. The between-class covariance matrix equals the between-class SSCP matrix divided by $n(c-1)/c$, where n is the number of observations and c is the number of classes. You should interpret the between-class covariances in comparison with the total-sample and within-class covariances, not as formal estimates of population parameters.

BSSCP

displays the between-class SSCP matrix.

CANONICAL

CAN

performs canonical discriminant analysis.

CANPREFIX=*name*

specifies a prefix for naming the canonical variables. By default, the names are Can1, Can2, ..., Can$n$. If you specify CANPREFIX=ABC, the components are named ABC1, ABC2, ABC3, and so on. The number of characters in the prefix, plus the number of digits required to designate the canonical variables, should not exceed 32. The prefix is truncated if the combined length exceeds 32.

The CANONICAL option is activated when you specify either the NCAN= or the CANPREFIX= option. A discriminant criterion is always derived in PROC DISCRIM. If you want canonical discriminant analysis without the use of discriminant criteria, you should use PROC CANDISC.

CROSSLIST

displays the cross validation classification results for each observation.

CROSSLISTERR

displays the cross validation classification results for misclassified observations only.

CROSSVALIDATE

specifies the cross validation classification of the input DATA= data set. When a parametric method is used, PROC DISCRIM classifies each observation in the DATA= data set by using a discriminant function computed from the other observations in the DATA= data set, excluding the observation being classified. When a nonparametric method is used, the covariance matrices used to compute the distances are based on all observations in the data set and do not exclude the observation being classified. However, the observation being classified is excluded from the nonparametric density estimation (if you specify the R= option) or the k nearest neighbors (if you specify the K= or KPROP= option) of that observation. The CROSSVALIDATE option is set when you specify the CROSSLIST, CROSSLISTERR, or OUTCROSS= option.

DATA=*SAS-data-set*

specifies the data set to be analyzed. The data set can be an ordinary SAS data set or one of several specially structured data sets created by SAS/STAT procedures. These specially structured data sets include TYPE=CORR, TYPE=COV, TYPE=CSSCP, TYPE=SSCP, TYPE=LINEAR, TYPE=QUAD, and TYPE=MIXED. The input data set must be an ordinary SAS data set if you specify METHOD=NPAR. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

DISTANCE

MAHALANOBIS

displays the squared Mahalanobis distances between the group means, F statistics, and the corresponding probabilities of greater Mahalanobis squared distances between the group means. The squared distances are based on the specification of the POOL= and METRIC= options.

K=*k*

specifies a k value for the k-nearest-neighbor rule. An observation $\mathbf{x}$ is classified into a group based on the information from the k nearest neighbors of $\mathbf{x}$. Do not specify the K= option with the KPROP= or R= option.

KPROP=*p*

specifies a proportion, p, for computing the k value for the k-nearest-neighbor rule: $k = \max(1, \text{floor}(np))$, where n is the number of valid observations. When there is a FREQ statement, n is the sum of the FREQ variable for the observations used in the analysis (those without missing or invalid values). An observation $\mathbf{x}$ is classified into a group based on the information from the k nearest neighbors of $\mathbf{x}$. Do not specify the KPROP= option with the K= or R= option.

KERNEL=BIWEIGHT | BIW

KERNEL=EPANECHNIKOV | EPA

KERNEL=NORMAL | NOR

KERNEL=TRIWEIGHT | TRI

KERNEL=UNIFORM | UNI

specifies a kernel density to estimate the group-specific densities. You can specify the KERNEL= option only when the R= option is specified. The default is KERNEL=UNIFORM.

LIST

displays the resubstitution classification results for each observation. You can specify this option only when the input data set is an ordinary SAS data set.

LISTERR

displays the resubstitution classification results for misclassified observations only. You can specify this option only when the input data set is an ordinary SAS data set.

MANOVA

displays multivariate statistics for testing the hypothesis that the class means are equal in the population.

METHOD=NORMAL | NPAR

determines the method to use in deriving the classification criterion. When you specify METHOD=NORMAL, a parametric method based on a multivariate normal distribution within each class is used to derive a linear or quadratic discriminant function. The default is METHOD=NORMAL. When you specify METHOD=NPAR, a nonparametric method is used and you must also specify either the K= or R= option.

METRIC=DIAGONAL | FULL | IDENTITY

specifies the metric in which the computations of squared distances are performed. If you specify METRIC=FULL, then PROC DISCRIM uses either the pooled covariance matrix (POOL=YES) or individual within-group covariance matrices (POOL=NO) to compute the squared distances. If you specify METRIC=DIAGONAL, then PROC DISCRIM uses either the diagonal matrix of the pooled covariance matrix (POOL=YES) or diagonal matrices of individual within-group covariance matrices (POOL=NO) to compute the squared distances. If you specify METRIC=IDENTITY, then PROC DISCRIM uses Euclidean distance. The default is METRIC=FULL. When you specify METHOD=NORMAL, the option METRIC=FULL is used.

NCAN=*number*

specifies the number of canonical variables to compute. The value of *number* must be less than or equal to the number of variables. If you specify the option NCAN=0, the procedure displays the canonical correlations but not the canonical coefficients, structures, or means. Let v be the number of variables in the VAR statement, and let c be the number of classes. If you omit the NCAN= option, only $\min(v, c-1)$ canonical variables are generated. If you request an output data set (OUT=, OUTCROSS=, TESTOUT=), v canonical variables are generated. In this case, the last $v-(c-1)$ canonical variables have missing values.

The CANONICAL option is activated when you specify either the NCAN= or the CANPREFIX= option. A discriminant criterion is always derived in PROC DISCRIM. If you want canonical discriminant analysis without the use of discriminant criterion, you should use PROC CANDISC.

NOCLASSIFY

suppresses the resubstitution classification of the input DATA= data set. You can specify this option only when the input data set is an ordinary SAS data set.

NOPRINT

suppresses the normal display of results. Note that this option temporarily disables the Out-

put Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

OUT=*SAS-data-set*
creates an output SAS data set containing all the data from the DATA= data set, plus the posterior probabilities and the class into which each observation is classified by resubstitution. When you specify the CANONICAL option, the data set also contains new variables with canonical variable scores. See the section "OUT= Data Set" on page 1414 for more information.

OUTCROSS=*SAS-data-set*
creates an output SAS data set containing all the data from the DATA= data set, plus the posterior probabilities and the class into which each observation is classified by cross validation. When you specify the CANONICAL option, the data set also contains new variables with canonical variable scores. See the section "OUT= Data Set" on page 1414 for more information.

OUTD=*SAS-data-set*
creates an output SAS data set containing all the data from the DATA= data set, plus the group-specific density estimates for each observation. See the section "OUT= Data Set" on page 1414 for more information.

OUTSTAT=*SAS-data-set*
creates an output SAS data set containing various statistics such as means, standard deviations, and correlations. When the input data set is an ordinary SAS data set or when TYPE=CORR, TYPE=COV, TYPE=CSSCP, or TYPE=SSCP, this option can be used to generate discriminant statistics. When you specify the CANONICAL option, canonical correlations, canonical structures, canonical coefficients, and means of canonical variables for each class are included in the data set. If you specify METHOD=NORMAL, the output data set also includes coefficients of the discriminant functions, and the output data set is TYPE=LINEAR (POOL=YES), TYPE=QUAD (POOL=NO), or TYPE=MIXED (POOL=TEST). If you specify METHOD=NPAR, this output data set is TYPE=CORR. This data set also holds calibration information that can be used to classify new observations. See the sections "Saving and Using Calibration Information" on page 1411 and "OUT= Data Set" on page 1414 for more information.

PCORR
displays pooled within-class correlations.

PCOV
displays pooled within-class covariances.

POOL=NO | TEST | YES
determines whether the pooled or within-group covariance matrix is the basis of the measure of the squared distance. If you specify POOL=YES, then PROC DISCRIM uses the pooled covariance matrix in calculating the (generalized) squared distances. Linear discriminant functions are computed. If you specify POOL=NO, the procedure uses the individual within-group covariance matrices in calculating the distances. Quadratic discriminant functions are computed. The default is POOL=YES. The k-nearest-neighbor method assumes the default

of POOL=YES, and the POOL=TEST option cannot be used with the METHOD=NPAR option.

When you specify METHOD=NORMAL, the option POOL=TEST requests Bartlett's modification of the likelihood ratio test (Morrison 1976; Anderson 1984) of the homogeneity of the within-group covariance matrices. The test is unbiased (Perlman 1980). However, it is not robust to nonnormality. If the test statistic is significant at the level specified by the SLPOOL= option, the within-group covariance matrices are used. Otherwise, the pooled covariance matrix is used. The discriminant function coefficients are displayed only when the pooled covariance matrix is used.

POSTERR

displays the posterior probability error-rate estimates of the classification criterion based on the classification results.

PSSCP

displays the pooled within-class corrected SSCP matrix.

R=*r*

specifies a radius r value for kernel density estimation. With uniform, Epanechnikov, biweight, or triweight kernels, an observation $\mathbf{x}$ is classified into a group based on the information from observations $\mathbf{y}$ in the training set within the radius r of $\mathbf{x}$—that is, the group t observations $\mathbf{y}$ with squared distance $d_t^2(\mathbf{x},\mathbf{y}) \leq r^2$. When a normal kernel is used, the classification of an observation $\mathbf{x}$ is based on the information of the estimated group-specific densities from all observations in the training set. The matrix $r^2\mathbf{V}_t$ is used as the group t covariance matrix in the normal-kernel density, where $\mathbf{V}_t$ is the matrix used in calculating the squared distances. Do not specify the K= or KPROP= option with the R= option. For more information about selecting r, see the section "Nonparametric Methods" on page 1403.

SHORT

suppresses the display of certain items in the default output. If you specify METHOD=NORMAL, then PROC DISCRIM suppresses the display of determinants, generalized squared distances between-class means, and discriminant function coefficients. When you specify the CANONICAL option, PROC DISCRIM suppresses the display of canonical structures, canonical coefficients, and class means on canonical variables; only tables of canonical correlations are displayed.

SIMPLE

displays simple descriptive statistics for the total sample and within each class.

SINGULAR=*p*

specifies the criterion for determining the singularity of a matrix, where $0 < p < 1$. The default is SINGULAR=1E–8.

Let $\mathbf{S}$ be the total-sample correlation matrix. If the R square for predicting a quantitative variable in the VAR statement from the variables preceding it exceeds $1 - p$, then $\mathbf{S}$ is considered singular. If $\mathbf{S}$ is singular, the probability levels for the multivariate test statistics and canonical correlations are adjusted for the number of variables with R square exceeding $1 - p$.

Let $\mathbf{S}_t$ be the group t covariance matrix, and let $\mathbf{S}_p$ be the pooled covariance matrix. In group t, if the R square for predicting a quantitative variable in the VAR statement from the variables

preceding it exceeds $1 - p$, then $\mathbf{S}_t$ is considered singular. Similarly, if the partial R square for predicting a quantitative variable in the VAR statement from the variables preceding it, after controlling for the effect of the CLASS variable, exceeds $1 - p$, then $\mathbf{S}_p$ is considered singular.

If PROC DISCRIM needs to compute either the inverse or the determinant of a matrix that is considered singular, then it uses a quasi inverse or a quasi determinant. For details, see the section "Quasi-inverse" on page 1408.

SLPOOL=*p*

specifies the significance level for the test of homogeneity. You can specify the SLPOOL= option only when POOL=TEST is also specified. If you specify POOL= TEST but omit the SLPOOL= option, PROC DISCRIM uses 0.10 as the significance level for the test.

STDMEAN

displays total-sample and pooled within-class standardized class means.

TCORR

displays total-sample correlations.

TCOV

displays total-sample covariances.

TESTDATA=*SAS-data-set*

names an ordinary SAS data set with observations that are to be classified. The quantitative variable names in this data set must match those in the DATA= data set. When you specify the TESTDATA= option, you can also specify the TESTCLASS, TESTFREQ, and TESTID statements. When you specify the TESTDATA= option, you can use the TESTOUT= and TESTOUTD= options to generate classification results and group-specific density estimates for observations in the test data set. Note that if the CLASS variable is not present in the TESTDATA= data set, the output will not include misclassification statistics.

TESTLIST

lists classification results for all observations in the TESTDATA= data set.

TESTLISTERR

lists only misclassified observations in the TESTDATA= data set but only if a TESTCLASS statement is also used.

TESTOUT=*SAS-data-set*

creates an output SAS data set containing all the data from the TESTDATA= data set, plus the posterior probabilities and the class into which each observation is classified. When you specify the CANONICAL option, the data set also contains new variables with canonical variable scores. See the section "OUT= Data Set" on page 1414 for more information.

TESTOUTD=*SAS-data-set*

creates an output SAS data set containing all the data from the TESTDATA= data set, plus the group-specific density estimates for each observation. See the section "OUT= Data Set" on page 1414 for more information.

THRESHOLD=*p*
specifies the minimum acceptable posterior probability for classification, where $0 \leq p \leq 1$. If the largest posterior probability of group membership is less than the THRESHOLD value, the observation is labeled as 'Other'. The default is THRESHOLD=0.

TSSCP
displays the total-sample corrected SSCP matrix.

WCORR
displays within-class correlations for each class level.

WCOV
displays within-class covariances for each class level.

WSSCP
displays the within-class corrected SSCP matrix for each class level.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC DISCRIM to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC DISCRIM. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

If you specify the TESTDATA= option and the TESTDATA= data set does not contain any of the BY variables, then the entire TESTDATA= data set is classified according to the discriminant functions computed in each BY group in the DATA= data set.

If the TESTDATA= data set contains some but not all of the BY variables, or if some BY variables do not have the same type or length in the TESTDATA= data set as in the DATA= data set, then PROC DISCRIM displays an error message and stops.

If all BY variables appear in the TESTDATA= data set with the same type and length as in the DATA= data set, then each BY group in the TESTDATA= data set is classified by the discriminant function from the corresponding BY group in the DATA= data set. The BY groups in the TESTDATA= data set must be in the same order as in the DATA= data set. If you specify the NOTSORTED option in the BY statement, there must be exactly the same BY groups in the same order in both data sets. If you omit the NOTSORTED option, some BY groups can appear in one data set but not in the other. If some BY groups appear in the TESTDATA= data set but not in the DATA= data set, and you request an output test data set by using the TESTOUT= or TESTOUTD= option, these BY groups are not included in the output data set.

CLASS Statement

CLASS *variable* **;**

The values of the classification variable define the groups for analysis. Class levels are determined by the formatted values of the CLASS variable. The specified variable can be numeric or character. A CLASS statement is required.

FREQ Statement

FREQ *variable* **;**

If a variable in the data set represents the frequency of occurrence for the other values in the observation, include the variable's name in a FREQ statement. The procedure then treats the data set as if each observation appears n times, where n is the value of the FREQ variable for the observation. The total number of observations is considered to be equal to the sum of the FREQ variable when the procedure determines degrees of freedom for significance probabilities.

If the value of the FREQ variable is missing or is less than one, the observation is not used in the analysis. If the value is not an integer, it is truncated to an integer.

ID Statement

ID *variable* **;**

The ID statement is effective only when you specify the LIST or LISTERR option in the PROC DISCRIM statement. When the DISCRIM procedure displays the classification results, the ID variable (rather than the observation number) is displayed for each observation.

PRIORS Statement

PRIORS *EQUAL* **;**

PRIORS *PROPORTIONAL* | *PROP* **;**

PRIORS *probabilities* **;**

The PRIORS statement specifies the prior probabilities of group membership. To set the prior probabilities equal, use the following statement:

```
priors equal;
```

To set the prior probabilities proportional to the sample sizes, use the following statement:

```
priors proportional;
```

For other than equal or proportional priors, specify the prior probability for each level of the classification variable. Each class level can be written as either a SAS name or a quoted string, and it must be followed by an equal sign and a numeric constant between zero and one. A SAS name begins with a letter or an underscore and can contain digits as well. Lowercase character values and data values with leading blanks must be enclosed in quotes. For example, to define prior probabilities for each level of Grade, where Grade's values are A, B, C, and D, the PRIORS statement can be specified as follows:

```
priors A=0.1 B=0.3 C=0.5 D=0.1;
```

If Grade's values are 'a', 'b', 'c', and 'd', each class level must be written as a quoted string as follows:

```
priors 'a'=0.1  'b'=0.3  'c'=0.5  'd'=0.1;
```

If Grade is numeric, with formatted values of '1', '2', and '3', the PRIORS statement can be written as follows:

```
priors '1'=0.3  '2'=0.6  '3'=0.1;
```

The specified class levels must exactly match the formatted values of the CLASS variable. For example, if a CLASS variable C has the format 4.2 and a value 5, the PRIORS statement must specify '5.00', not '5.0' or '5'. If the prior probabilities do not sum to one, these probabilities are scaled proportionally to have the sum equal to one. The default is PRIORS EQUAL.

TESTCLASS Statement

TESTCLASS *variable* **;**

The TESTCLASS statement names the variable in the TESTDATA= data set that is used to determine whether an observation in the TESTDATA= data set is misclassified. The TESTCLASS variable should have the same type (character or numeric) and length as the variable given in the CLASS statement. PROC DISCRIM considers an observation misclassified when the formatted value of the TESTCLASS variable does not match the group into which the TESTDATA= observation is classified. When the TESTCLASS statement is missing and the TESTDATA= data set contains the variable given in the CLASS statement, the CLASS variable is used as the TESTCLASS variable. Note that if the CLASS variable is not present in the TESTDATA= data set, the output will not include misclassification statistics.

TESTFREQ Statement

TESTFREQ *variable* **;**

If a variable in the TESTDATA= data set represents the frequency of occurrence of the other values in the observation, include the variable's name in a TESTFREQ statement. The procedure then treats the data set as if each observation appears n times, where n is the value of the TESTFREQ variable for the observation.

If the value of the TESTFREQ variable is missing or is less than one, the observation is not used in the analysis. If the value is not an integer, it is truncated to an integer.

TESTID Statement

TESTID *variable* **;**

The TESTID statement is effective only when you specify the TESTLIST or TESTLISTERR option in the PROC DISCRIM statement. When the DISCRIM procedure displays the classification results for the TESTDATA= data set, the TESTID variable (rather than the observation number) is displayed for each observation. The variable given in the TESTID statement must be in the TESTDATA= data set.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the quantitative variables to be included in the analysis. The default is all numeric variables not listed in other statements.

WEIGHT Statement

WEIGHT *variable* **;**

To use relative weights for each observation in the input data set, place the weights in a variable in the data set and specify the name in a WEIGHT statement. This is often done when the variance associated with each observation is different and the values of the weight variable are proportional to the reciprocals of the variances. If the value of the WEIGHT variable is missing or is less than zero, then a value of zero for the weight is used.

The WEIGHT and FREQ statements have a similar effect except that the WEIGHT statement does not alter the degrees of freedom.

Details: DISCRIM Procedure

Missing Values

Observations with missing values for variables in the analysis are excluded from the development of the classification criterion. When the values of the classification variable are missing, the observation is excluded from the development of the classification criterion, but if no other variables in the analysis have missing values for that observation, the observation is classified and displayed with the classification results.

Background

The following notation is used to describe the classification methods:

$\mathbf{x}$	a p-dimensional vector containing the quantitative variables of an observation
$\mathbf{S}_p$	the pooled covariance matrix
t	a subscript to distinguish the groups
n_t	the number of training set observations in group t
$\mathbf{m}_t$	the p-dimensional vector containing variable means in group t
$\mathbf{S}_t$	the covariance matrix within group t
$\|\mathbf{S}_t\|$	the determinant of $\mathbf{S}_t$
q_t	the prior probability of membership in group t
$p(t\|\mathbf{x})$	the posterior probability of an observation $\mathbf{x}$ belonging to group t

f_t	the probability density function for group t
$f_t(\mathbf{x})$	the group-specific density estimate at $\mathbf{x}$ from group t
$f(\mathbf{x})$	$\sum_t q_t f_t(\mathbf{x})$, the estimated unconditional density at $\mathbf{x}$
e_t	the classification error rate for group t

Bayes' Theorem

Assuming that the prior probabilities of group membership are known and that the group-specific densities at $\mathbf{x}$ can be estimated, PROC DISCRIM computes $p(t|\mathbf{x})$, the probability of $\mathbf{x}$ belonging to group t, by applying Bayes' theorem:

$$p(t|\mathbf{x}) = \frac{q_t f_t(\mathbf{x})}{f(\mathbf{x})}$$

PROC DISCRIM partitions a p-dimensional vector space into regions R_t, where the region R_t is the subspace containing all p-dimensional vectors $\mathbf{y}$ such that $p(t|\mathbf{y})$ is the largest among all groups. An observation is classified as coming from group t if it lies in region R_t.

Parametric Methods

Assuming that each group has a multivariate normal distribution, PROC DISCRIM develops a discriminant function or classification criterion by using a measure of generalized squared distance. The classification criterion is based on either the individual within-group covariance matrices or the pooled covariance matrix; it also takes into account the prior probabilities of the classes. Each observation is placed in the class from which it has the smallest generalized squared distance. PROC DISCRIM also computes the posterior probability of an observation belonging to each class.

The squared Mahalanobis distance from $\mathbf{x}$ to group t is

$$d_t^2(\mathbf{x}) = (\mathbf{x} - \mathbf{m}_t)' \mathbf{V}_t^{-1} (\mathbf{x} - \mathbf{m}_t)$$

where $\mathbf{V}_t = \mathbf{S}_t$ if the within-group covariance matrices are used, or $\mathbf{V}_t = \mathbf{S}_p$ if the pooled covariance matrix is used.

The group-specific density estimate at $\mathbf{x}$ from group t is then given by

$$f_t(\mathbf{x}) = (2\pi)^{-\frac{p}{2}} |\mathbf{V}_t|^{-\frac{1}{2}} \exp\left(-0.5 d_t^2(\mathbf{x})\right)$$

Using Bayes' theorem, the posterior probability of $\mathbf{x}$ belonging to group t is

$$p(t|\mathbf{x}) = \frac{q_t f_t(\mathbf{x})}{\sum_u q_u f_u(\mathbf{x})}$$

where the summation is over all groups.

The generalized squared distance from $\mathbf{x}$ to group t is defined as

$$D_t^2(\mathbf{x}) = d_t^2(\mathbf{x}) + g_1(t) + g_2(t)$$

where

$$g_1(t) = \begin{cases} \ln|\mathbf{S}_t| & \text{if the within-group covariance matrices are used} \\ 0 & \text{if the pooled covariance matrix is used} \end{cases}$$

and

$$g_2(t) = \begin{cases} -2\ln(q_t) & \text{if the prior probabilities are not all equal} \\ 0 & \text{if the prior probabilities are all equal} \end{cases}$$

The posterior probability of $\mathbf{x}$ belonging to group t is then equal to

$$p(t|\mathbf{x}) = \frac{\exp\left(-0.5D_t^2(\mathbf{x})\right)}{\sum_u \exp\left(-0.5D_u^2(\mathbf{x})\right)}$$

The discriminant scores are $-0.5D_u^2(\mathbf{x})$. An observation is classified into group u if setting $t = u$ produces the largest value of $p(t|\mathbf{x})$ or the smallest value of $D_t^2(\mathbf{x})$. If this largest posterior probability is less than the threshold specified, $\mathbf{x}$ is labeled as 'Other'.

Nonparametric Methods

Nonparametric discriminant methods are based on nonparametric estimates of group-specific probability densities. Either a kernel method or the k-nearest-neighbor method can be used to generate a nonparametric density estimate in each group and to produce a classification criterion. The kernel method uses uniform, normal, Epanechnikov, biweight, or triweight kernels in the density estimation.

Either Mahalanobis distance or Euclidean distance can be used to determine proximity. When the k-nearest-neighbor method is used, the Mahalanobis distances are based on the pooled covariance matrix. When a kernel method is used, the Mahalanobis distances are based on either the individual within-group covariance matrices or the pooled covariance matrix. Either the full covariance matrix or the diagonal matrix of variances can be used to calculate the Mahalanobis distances.

The squared distance between two observation vectors, $\mathbf{x}$ and $\mathbf{y}$, in group t is given by

$$d_t^2(\mathbf{x},\mathbf{y}) = (\mathbf{x}-\mathbf{y})'\mathbf{V}_t^{-1}(\mathbf{x}-\mathbf{y})$$

where $\mathbf{V}_t$ has one of the following forms:

$$\mathbf{V}_t = \begin{cases} \mathbf{S}_p & \text{the pooled covariance matrix} \\ \text{diag}(\mathbf{S}_p) & \text{the diagonal matrix of the pooled covariance matrix} \\ \mathbf{S}_t & \text{the covariance matrix within group } t \\ \text{diag}(\mathbf{S}_t) & \text{the diagonal matrix of the covariance matrix within group } t \\ \mathbf{I} & \text{the identity matrix} \end{cases}$$

The classification of an observation vector $\mathbf{x}$ is based on the estimated group-specific densities from the training set. From these estimated densities, the posterior probabilities of group membership at $\mathbf{x}$ are evaluated. An observation $\mathbf{x}$ is classified into group u if setting $t = u$ produces the largest

value of $p(t|\mathbf{x})$. If there is a tie for the largest probability or if this largest probability is less than the threshold specified, $\mathbf{x}$ is labeled as 'Other'.

The kernel method uses a fixed radius, r, and a specified kernel, K_t, to estimate the group t density at each observation vector $\mathbf{x}$. Let $\mathbf{z}$ be a p-dimensional vector. Then the volume of a p-dimensional unit sphere bounded by $\mathbf{z}'\mathbf{z} = 1$ is

$$v_0 = \frac{\pi^{\frac{p}{2}}}{\Gamma\left(\frac{p}{2}+1\right)}$$

where Γ represents the gamma function (see *SAS Language Reference: Dictionary*).

Thus, in group t, the volume of a p-dimensional ellipsoid bounded by $\{\mathbf{z} \mid \mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z} = r^2\}$ is

$$v_r(t) = r^p |\mathbf{V}_t|^{\frac{1}{2}} v_0$$

The kernel method uses one of the following densities as the kernel density in group t:

Uniform Kernel

$$K_t(\mathbf{z}) = \begin{cases} \dfrac{1}{v_r(t)} & \text{if } \mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z} \le r^2 \\ 0 & \text{elsewhere} \end{cases}$$

Normal Kernel (with mean zero, variance $r^2\mathbf{V}_t$)

$$K_t(\mathbf{z}) = \frac{1}{c_0(t)} \exp\left(-\frac{1}{2r^2}\mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z}\right)$$

where $c_0(t) = (2\pi)^{\frac{p}{2}} r^p |\mathbf{V}_t|^{\frac{1}{2}}$.

Epanechnikov Kernel

$$K_t(\mathbf{z}) = \begin{cases} c_1(t)\left(1 - \dfrac{1}{r^2}\mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z}\right) & \text{if } \mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z} \le r^2 \\ 0 & \text{elsewhere} \end{cases}$$

where $c_1(t) = \dfrac{1}{v_r(t)}\left(1 + \dfrac{p}{2}\right)$.

Biweight Kernel

$$K_t(\mathbf{z}) = \begin{cases} c_2(t)\left(1 - \dfrac{1}{r^2}\mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z}\right)^2 & \text{if } \mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z} \le r^2 \\ 0 & \text{elsewhere} \end{cases}$$

where $c_2(t) = \left(1 + \frac{p}{4}\right) c_1(t)$.

Triweight Kernel

$$K_t(\mathbf{z}) = \begin{cases} c_3(t)\left(1 - \frac{1}{r^2}\mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z}\right)^3 & \text{if } \mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z} \le r^2 \\ 0 & \text{elsewhere} \end{cases}$$

where $c_3(t) = \left(1 + \frac{p}{6}\right) c_2(t)$.

The group t density at $\mathbf{x}$ is estimated by

$$f_t(\mathbf{x}) = \frac{1}{n_t} \sum_{\mathbf{y}} K_t(\mathbf{x} - \mathbf{y})$$

where the summation is over all observations $\mathbf{y}$ in group t, and K_t is the specified kernel function. The posterior probability of membership in group t is then given by

$$p(t|\mathbf{x}) = \frac{q_t f_t(\mathbf{x})}{f(\mathbf{x})}$$

where $f(\mathbf{x}) = \sum_u q_u f_u(\mathbf{x})$ is the estimated unconditional density. If $f(\mathbf{x})$ is zero, the observation $\mathbf{x}$ is labeled as 'Other'.

The uniform-kernel method treats $K_t(\mathbf{z})$ as a multivariate uniform function with density uniformly distributed over $\mathbf{z}'\mathbf{V}_t^{-1}\mathbf{z} \le r^2$. Let k_t be the number of training set observations $\mathbf{y}$ from group t within the closed ellipsoid centered at $\mathbf{x}$ specified by $d_t^2(\mathbf{x}, \mathbf{y}) \le r^2$. Then the group t density at $\mathbf{x}$ is estimated by

$$f_t(\mathbf{x}) = \frac{k_t}{n_t v_r(t)}$$

When the identity matrix or the pooled within-group covariance matrix is used in calculating the squared distance, $v_r(t)$ is a constant, independent of group membership. The posterior probability of $\mathbf{x}$ belonging to group t is then given by

$$p(t|\mathbf{x}) = \frac{\frac{q_t k_t}{n_t}}{\sum_u \frac{q_u k_u}{n_u}}$$

If the closed ellipsoid centered at $\mathbf{x}$ does not include any training set observations, $f(\mathbf{x})$ is zero and $\mathbf{x}$ is labeled as 'Other'. When the prior probabilities are equal, $p(t|\mathbf{x})$ is proportional to k_t/n_t and $\mathbf{x}$ is classified into the group that has the highest proportion of observations in the closed ellipsoid. When the prior probabilities are proportional to the group sizes, $p(t|\mathbf{x}) = k_t / \sum_u k_u$, $\mathbf{x}$ is classified into the group that has the largest number of observations in the closed ellipsoid.

The nearest-neighbor method fixes the number, k, of training set points for each observation $\mathbf{x}$. The method finds the radius $r_k(\mathbf{x})$ that is the distance from $\mathbf{x}$ to the kth-nearest training set point in the metric $\mathbf{V}_t^{-1}$. Consider a closed ellipsoid centered at $\mathbf{x}$ bounded by $\{\mathbf{z} \mid (\mathbf{z}-\mathbf{x})'\mathbf{V}_t^{-1}(\mathbf{z}-\mathbf{x}) = r_k^2(\mathbf{x})\}$;

the nearest-neighbor method is equivalent to the uniform-kernel method with a location-dependent radius $r_k(\mathbf{x})$. Note that, with ties, more than k training set points might be in the ellipsoid.

Using the k-nearest-neighbor rule, the k_n (or more with ties) smallest distances are saved. Of these k distances, let k_t represent the number of distances that are associated with group t. Then, as in the uniform-kernel method, the estimated group t density at $\mathbf{x}$ is

$$f_t(\mathbf{x}) = \frac{k_t}{n_t v_k(\mathbf{x})}$$

where $v_k(\mathbf{x})$ is the volume of the ellipsoid bounded by $\{\mathbf{z} \mid (\mathbf{z}-\mathbf{x})'\mathbf{V}_t^{-1}(\mathbf{z}-\mathbf{x}) = r_k^2(\mathbf{x})\}$. Since the pooled within-group covariance matrix is used to calculate the distances used in the nearest-neighbor method, the volume $v_k(\mathbf{x})$ is a constant independent of group membership. When $k = 1$ is used in the nearest-neighbor rule, $\mathbf{x}$ is classified into the group associated with the $\mathbf{y}$ point that yields the smallest squared distance $d_t^2(\mathbf{x}, \mathbf{y})$. Prior probabilities affect nearest-neighbor results in the same way that they affect uniform-kernel results.

With a specified squared distance formula (METRIC=, POOL=), the values of r and k determine the degree of irregularity in the estimate of the density function, and they are called smoothing parameters. Small values of r or k produce jagged density estimates, and large values of r or k produce smoother density estimates. Various methods for choosing the smoothing parameters have been suggested, and there is as yet no simple solution to this problem.

For a fixed kernel shape, one way to choose the smoothing parameter r is to plot estimated densities with different values of r and to choose the estimate that is most in accordance with the prior information about the density. For many applications, this approach is satisfactory.

Another way of selecting the smoothing parameter r is to choose a value that optimizes a given criterion. Different groups might have different sets of optimal values. Assume that the unknown density has bounded and continuous second derivatives and that the kernel is a symmetric probability density function. One criterion is to minimize an approximate mean integrated square error of the estimated density (Rosenblatt 1956). The resulting optimal value of r depends on the density function and the kernel. A reasonable choice for the smoothing parameter r is to optimize the criterion with the assumption that group t has a normal distribution with covariance matrix $\mathbf{V}_t$. Then, in group t, the resulting optimal value for r is given by

$$\left(\frac{A(K_t)}{n_t}\right)^{1/(p+4)}$$

where the optimal constant $A(K_t)$ depends on the kernel K_t (Epanechnikov 1969). For some useful kernels, the constants $A(K_t)$ are given by the following:

$$A(K_t) = \frac{1}{p}2^{p+1}(p+2)\Gamma\left(\frac{p}{2}\right) \qquad \text{with a uniform kernel}$$

$$A(K_t) = \frac{4}{2p+1} \qquad \text{with a normal kernel}$$

$$A(K_t) = \frac{2^{p+2}p^2(p+2)(p+4)}{2p+1}\Gamma\left(\frac{p}{2}\right) \qquad \text{with an Epanechnikov kernel}$$

These selections of $A(K_t)$ are derived under the assumption that the data in each group are from a multivariate normal distribution with covariance matrix $\mathbf{V}_t$. However, when the Euclidean distances are used in calculating the squared distance ($\mathbf{V}_t = I$), the smoothing constant should be multiplied by s, where s is an estimate of standard deviations for all variables. A reasonable choice for s is

$$s = \left(\frac{1}{p} \sum s_{jj} \right)^{\frac{1}{2}}$$

where s_{jj} are group t marginal variances.

The DISCRIM procedure uses only a single smoothing parameter for all groups. However, the selection of the matrix in the distance formula (from the METRIC= or POOL= option), enables individual groups and variables to have different scalings. When $\mathbf{V}_t$, the matrix used in calculating the squared distances, is an identity matrix, the kernel estimate at each data point is scaled equally for all variables in all groups. When $\mathbf{V}_t$ is the diagonal matrix of a covariance matrix, each variable in group t is scaled separately by its variance in the kernel estimation, where the variance can be the pooled variance ($\mathbf{V}_t = \mathbf{S}_p$) or an individual within-group variance ($\mathbf{V}_t = \mathbf{S}_t$). When $\mathbf{V}_t$ is a full covariance matrix, the variables in group t are scaled simultaneously by $\mathbf{V}_t$ in the kernel estimation.

In nearest-neighbor methods, the choice of k is usually relatively uncritical (Hand 1982). A practical approach is to try several different values of the smoothing parameters within the context of the particular application and to choose the one that gives the best cross validated estimate of the error rate.

Classification Error-Rate Estimates

A classification criterion can be evaluated by its performance in the classification of future observations. PROC DISCRIM uses two types of error-rate estimates to evaluate the derived classification criterion based on parameters estimated by the training sample:

- error-count estimates
- posterior probability error-rate estimates

The error-count estimate is calculated by applying the classification criterion derived from the training sample to a test set and then counting the number of misclassified observations. The group-specific error-count estimate is the proportion of misclassified observations in the group. When the test set is independent of the training sample, the estimate is unbiased. However, the estimate can have a large variance, especially if the test set is small.

When the input data set is an ordinary SAS data set and no independent test sets are available, the same data set can be used both to define and to evaluate the classification criterion. The resulting error-count estimate has an optimistic bias and is called an *apparent error rate*. To reduce the bias, you can split the data into two sets—one set for deriving the discriminant function and the other set for estimating the error rate. Such a split-sample method has the unfortunate effect of reducing the effective sample size.

Another way to reduce bias is cross validation (Lachenbruch and Mickey 1968). Cross validation treats $n - 1$ out of n training observations as a training set. It determines the discriminant functions based on these $n - 1$ observations and then applies them to classify the one observation left out. This is done for each of the n training observations. The misclassification rate for each group is the proportion of sample observations in that group that are misclassified. This method achieves a nearly unbiased estimate but with a relatively large variance.

To reduce the variance in an error-count estimate, smoothed error-rate estimates are suggested (Glick 1978). Instead of summing terms that are either zero or one as in the error-count estimator, the smoothed estimator uses a continuum of values between zero and one in the terms that are summed. The resulting estimator has a smaller variance than the error-count estimate. The posterior probability error-rate estimates provided by the POSTERR option in the PROC DISCRIM statement (see the section "Posterior Probability Error-Rate Estimates" on page 1410) are smoothed error-rate estimates. The posterior probability estimates for each group are based on the posterior probabilities of the observations classified into that same group. The posterior probability estimates provide good estimates of the error rate when the posterior probabilities are accurate. When a parametric classification criterion (linear or quadratic discriminant function) is derived from a nonnormal population, the resulting posterior probability error-rate estimators might not be appropriate.

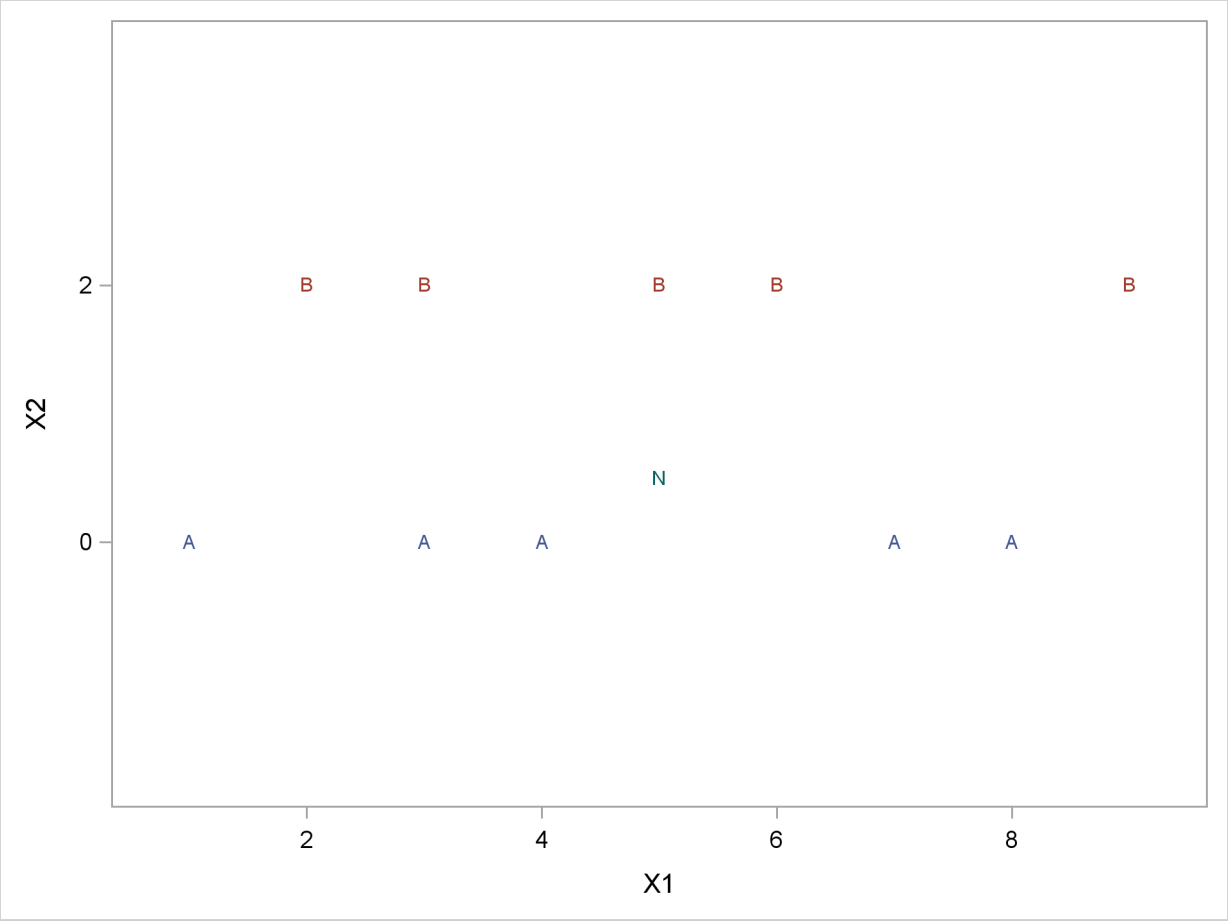
The overall error rate is estimated through a weighted average of the individual group-specific error-rate estimates, where the prior probabilities are used as the weights.

To reduce both the bias and the variance of the estimator, Hora and Wilcox (1982) compute the posterior probability estimates based on cross validation. The resulting estimates are intended to have both low variance from using the posterior probability estimate and low bias from cross validation. They use Monte Carlo studies on two-group multivariate normal distributions to compare the cross validation posterior probability estimates with three other estimators: the apparent error rate, cross validation estimator, and posterior probability estimator. They conclude that the cross validation posterior probability estimator has a lower mean squared error in their simulations.

Quasi-inverse

Consider the plot shown in Figure 31.6 with two variables, X1 and X2, and two classes, A and B. The within-class covariance matrix is diagonal, with a positive value for X1 but zero for X2. Using a Moore-Penrose pseudo-inverse would effectively ignore X2 in doing the classification, and the two classes would have a zero generalized distance and could not be discriminated at all. The quasi inverse used by PROC DISCRIM replaces the zero variance for X2 with a small positive number to remove the singularity. This permits X2 to be used in the discrimination and results correctly in a large generalized distance between the two classes and a zero error rate. It also permits new observations, such as the one indicated by N, to be classified in a reasonable way. PROC CANDISC also uses a quasi inverse when the total-sample covariance matrix is considered to be singular and Mahalanobis distances are requested. This problem with singular within-class covariance matrices is discussed in Ripley (1996, p. 38). The use of the quasi inverse is an innovation introduced by SAS.

Figure 31.6 Plot of Data with Singular Within-Class Covariance Matrix



Let $\mathbf{S}$ be a singular covariance matrix. The matrix $\mathbf{S}$ can be either a within-group covariance matrix, a pooled covariance matrix, or a total-sample covariance matrix. Let v be the number of variables in the VAR statement, and let the nullity n be the number of variables among them with (partial) R square exceeding $1 - p$. If the determinant of $\mathbf{S}$ (Testing of Homogeneity of Within Covariance Matrices) or the inverse of $\mathbf{S}$ (Squared Distances and Generalized Squared Distances) is required, a quasi determinant or quasi inverse is used instead. PROC DISCRIM scales each variable to unit total-sample variance before calculating this quasi inverse. The calculation is based on the spectral decomposition $\mathbf{S} = \mathbf{\Gamma}\mathbf{\Lambda}\mathbf{\Gamma}'$, where $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues λ_j, $j = 1, \ldots, v$, where $\lambda_i \geq \lambda_j$ when $i < j$, and $\mathbf{\Gamma}$ is a matrix with the corresponding orthonormal eigenvectors of $\mathbf{S}$ as columns. When the nullity n is less than v, set $\lambda_j^0 = \lambda_j$ for $j = 1, \ldots, v - n$, and $\lambda_j^0 = p\bar{\lambda}$ for $j = v - n + 1, \ldots, v$, where

$$\bar{\lambda} = \frac{1}{v - n} \sum_{k=1}^{v-n} \lambda_k$$

When the nullity n is equal to v, set $\lambda_j^0 = p$, for $j = 1, \ldots, v$. A quasi determinant is then defined as the product of λ_j^0, $j = 1, \ldots, v$. Similarly, a quasi inverse is then defined as $\mathbf{S}^* = \mathbf{\Gamma}\mathbf{\Lambda}^*\mathbf{\Gamma}'$, where $\mathbf{\Lambda}^*$ is a diagonal matrix of values $1/\lambda_j^0$, $j = 1, \ldots, v$.

Posterior Probability Error-Rate Estimates

The posterior probability error-rate estimates (Fukunaga and Kessel 1973; Glick 1978; Hora and Wilcox 1982) for each group are based on the posterior probabilities of the observations classified into that same group.

A sample of observations with classification results can be used to estimate the posterior error rates. The following notation is used to describe the sample:

$\mathcal{S}$	the set of observations in the (training) sample
n	the number of observations in $\mathcal{S}$
n_t	the number of observations in $\mathcal{S}$ in group t
$\mathcal{R}_t$	the set of observations such that the posterior probability belonging to group t is the largest
$\mathcal{R}_{ut}$	the set of observations from group u such that the posterior probability belonging to group t is the largest

The classification error rate for group t is defined as

$$e_t = 1 - \int_{\mathcal{R}_t} f_t(\mathbf{x}) d\mathbf{x}$$

The posterior probability of $\mathbf{x}$ for group t can be written as

$$p(t|\mathbf{x}) = \frac{q_t f_t(\mathbf{x})}{f(\mathbf{x})}$$

where $f(\mathbf{x}) = \sum_u q_u f_u(\mathbf{x})$ is the unconditional density of $\mathbf{x}$.

Thus, if you replace $f_t(\mathbf{x})$ with $p(t|\mathbf{x}) f(\mathbf{x})/q_t$, the error rate is

$$e_t = 1 - \frac{1}{q_t} \int_{\mathcal{R}_t} p(t|\mathbf{x}) f(\mathbf{x}) d\mathbf{x}$$

An estimator of e_t, unstratified over the groups from which the observations come, is then given by

$$\hat{e}_t \text{ (unstratified)} = 1 - \frac{1}{n q_t} \sum_{\mathcal{R}_t} p(t|\mathbf{x})$$

where $p(t|\mathbf{x})$ is estimated from the classification criterion, and the summation is over all sample observations of $\mathcal{S}$ classified into group t. The true group membership of each observation is not required in the estimation. The term $n q_t$ is the number of observations that are expected to be classified into group t, given the priors. If more observations than expected are classified into group t, then $\hat{e}_t$ can be negative.

Further, if you replace $f(\mathbf{x})$ with $\sum_u q_u f_u(\mathbf{x})$, the error rate can be written as

$$e_t = 1 - \frac{1}{q_t} \sum_u q_u \int_{\mathcal{R}_{ut}} p(t|\mathbf{x}) f_u(\mathbf{x}) d\mathbf{x}$$

and an estimator stratified over the group from which the observations come is given by

$$\hat{e}_t \text{ (stratified)} = 1 - \frac{1}{q_t} \sum_u q_u \frac{1}{n_u} \left(\sum_{\mathcal{R}_{ut}} p(t|\mathbf{x}) \right)$$

The inner summation is over all sample observations of $\mathcal{S}$ coming from group u and classified into group t, and n_u is the number of observations originally from group u. The stratified estimate uses only the observations with known group membership. When the prior probabilities of the group membership are proportional to the group sizes, the stratified estimate is the same as the unstratified estimator.

The estimated group-specific error rates can be less than zero, usually due to a large discrepancy between prior probabilities of group membership and group sizes. To have a reliable estimate for group-specific error rate estimates, you should use group sizes that are at least approximately proportional to the prior probabilities of group membership.

A total error rate is defined as a weighted average of the individual group error rates

$$e = \sum_t q_t e_t$$

and can be estimated from

$$\hat{e} \text{ (unstratified)} = \sum_t q_t \hat{e}_t \text{ (unstratified)}$$

or

$$\hat{e} \text{ (stratified)} = \sum_t q_t \hat{e}_t \text{ (stratified)}$$

The total unstratified error-rate estimate can also be written as

$$\hat{e} \text{ (unstratified)} = 1 - \frac{1}{n} \sum_t \sum_{\mathcal{R}_t} p(t|\mathbf{x})$$

which is one minus the average value of the maximum posterior probabilities for each observation in the sample. The prior probabilities of group membership do not appear explicitly in this overall estimate.

Saving and Using Calibration Information

When you specify METHOD=NORMAL to derive a linear or quadratic discriminant function, you can save the calibration information developed by the DISCRIM procedure in a SAS data set by using the OUTSTAT= option in the procedure. PROC DISCRIM then creates a specially structured SAS data set of TYPE=LINEAR, TYPE=QUAD, or TYPE=MIXED that contains the calibration information. For more information about these data sets, see Appendix A, "Special SAS Data Sets." Calibration information cannot be saved when METHOD=NPAR, but you can classify a TESTDATA= data set in the same step. For an example of this, see Example 31.1.

To use this calibration information to classify observations in another data set, specify both of the following:

- the name of the calibration data set after the DATA= option in the PROC DISCRIM statement
- the name of the data set to be classified after the TESTDATA= option in the PROC DISCRIM statement

Here is an example:

```
data original;
   input position x1 x2;
   datalines;
...[data lines]
;

proc discrim outstat=info;
   class position;
run;

data check;
   input position x1 x2;
   datalines;
...[second set of data lines]
;

proc discrim data=info testdata=check testlist;
   class position;
run;
```

The first DATA step creates the SAS data set Original, which the DISCRIM procedure uses to develop a classification criterion. Specifying OUTSTAT=INFO in the PROC DISCRIM statement causes the DISCRIM procedure to store the calibration information in a new data set called Info. The next DATA step creates the data set Check. The second PROC DISCRIM statement specifies DATA=INFO and TESTDATA=CHECK so that the classification criterion developed earlier is applied to the Check data set. Note that if the CLASS variable is not present in the TESTDATA= data set, the output will not include misclassification statistics.

Input Data Sets

DATA= Data Set

When you specify METHOD=NPAR, an ordinary SAS data set is required as the input DATA= data set. When you specify METHOD=NORMAL, the DATA= data set can be an ordinary SAS data set or one of several specially structured data sets created by SAS/STAT procedures. These specially structured data sets include the following:

- TYPE=CORR data sets created by PROC CORR by using a BY statement
- TYPE=COV data sets created by PROC PRINCOMP by using both the COV option and a BY statement
- TYPE=CSSCP data sets created by PROC CORR by using the CSSCP option and a BY statement, where the OUT= data set is assigned TYPE=CSSCP with the TYPE= data set option
- TYPE=SSCP data sets created by PROC REG by using both the OUTSSCP= option and a BY statement
- TYPE=LINEAR, TYPE=QUAD, and TYPE=MIXED data sets produced by previous runs of PROC DISCRIM that used both METHOD=NORMAL and OUTSTAT= options

When the input data set is TYPE=CORR, TYPE=COV, TYPE=CSSCP, or TYPE=SSCP, the BY variable in these data sets becomes the CLASS variable in the DISCRIM procedure.

When the input data set is TYPE=CORR, TYPE=COV, or TYPE=CSSCP, then PROC DISCRIM reads the number of observations for each class from the observations with _TYPE_='N' and reads the variable means in each class from the observations with _TYPE_='MEAN'. Then PROC DISCRIM reads the within-class correlations from the observations with _TYPE_='CORR' and reads the standard deviations from the observations with _TYPE_='STD' (data set TYPE=CORR), the within-class covariances from the observations with _TYPE_='COV' (data set TYPE=COV), or the within-class corrected sums of squares and crossproducts from the observations with _TYPE_='CSSCP' (data set TYPE=CSSCP).

When you specify POOL=YES and the data set does not include any observations with _TYPE_='CSSCP' (data set TYPE=CSSCP), _TYPE_='COV' (data set TYPE=COV), or _TYPE_='CORR' (data set TYPE=CORR) for each class, PROC DISCRIM reads the pooled within-class information from the data set. In this case, PROC DISCRIM reads the pooled within-class covariances from the observations with _TYPE_='PCOV' (data set TYPE=COV) or reads the pooled within-class correlations from the observations with _TYPE_='PCORR' and the pooled within-class standard deviations from the observations with _TYPE_='PSTD' (data set TYPE=CORR) or the pooled within-class corrected SSCP matrix from the observations with _TYPE_='PSSCP' (data set TYPE=CSSCP).

When the input data set is TYPE=SSCP, the DISCRIM procedure reads the number of observations for each class from the observations with _TYPE_='N', the sum of weights of observations for each class from the variable INTERCEP in observations with _TYPE_='SSCP' and _NAME_='INTERCEPT', the variable sums from the variable=*variablenames* in observations with _TYPE_='SSCP' and _NAME_='INTERCEPT', and the uncorrected sums of squares and crossproducts from the variable=*variablenames* in observations with _TYPE_='SSCP' and _NAME_='variablenames'.

When the input data set is TYPE=LINEAR, TYPE=QUAD, or TYPE=MIXED, then PROC DISCRIM reads the prior probabilities for each class from the observations with variable _TYPE_='PRIOR'.

When the input data set is TYPE=LINEAR, then PROC DISCRIM reads the coefficients of the linear discriminant functions from the observations with variable _TYPE_='LINEAR'.

When the input data set is TYPE=QUAD, then PROC DISCRIM reads the coefficients of the quadratic discriminant functions from the observations with variable _TYPE_='QUAD'.

When the input data set is TYPE=MIXED, then PROC DISCRIM reads the coefficients of the linear discriminant functions from the observations with variable _TYPE_='LINEAR'. If there are no observations with _TYPE_='LINEAR', then PROC DISCRIM reads the coefficients of the quadratic discriminant functions from the observations with variable _TYPE_='QUAD'.

TESTDATA= Data Set

The TESTDATA= data set is an ordinary SAS data set with observations that are to be classified. The quantitative variable names in this data set must match those in the DATA= data set. The TESTCLASS statement can be used to specify the variable containing group membership information of the TESTDATA= data set observations. When the TESTCLASS statement is missing and the TESTDATA= data set contains the variable given in the CLASS statement, this variable is used as the TESTCLASS variable. The TESTCLASS variable should have the same type (character or numeric) and length as the variable given in the CLASS statement. PROC DISCRIM considers an observation misclassified when the value of the TESTCLASS variable does not match the group into which the TESTDATA= observation is classified.

Output Data Sets

When an output data set includes variables containing the posterior probabilities of group membership (OUT=, OUTCROSS=, or TESTOUT= data sets) or group-specific density estimates (OUTD= or TESTOUTD= data sets), the names of these variables are constructed from the formatted values of the class levels converted to valid SAS variable names.

OUT= Data Set

The OUT= data set contains all the variables in the DATA= data set, plus new variables containing the posterior probabilities and the resubstitution classification results. The names of the new variables containing the posterior probabilities are constructed from the formatted values of the class levels converted to SAS names. A new variable, _INTO_, with the same attributes as the CLASS variable, specifies the class to which each observation is assigned. If an observation is labeled as 'Other', the variable _INTO_ has a missing value. When you specify the CANONICAL option, the data set also contains new variables with canonical variable scores. The NCAN= option determines the number of canonical variables. The names of the canonical variables are constructed as described in the CANPREFIX= option. The canonical variables have means equal to zero and pooled within-class variances equal to one.

An OUT= data set cannot be created if the DATA= data set is not an ordinary SAS data set.

OUTD= Data Set

The OUTD= data set contains all the variables in the DATA= data set, plus new variables containing the group-specific density estimates. The names of the new variables containing the density estimates are constructed from the formatted values of the class levels.

An OUTD= data set cannot be created if the DATA= data set is not an ordinary SAS data set.

OUTCROSS= Data Set

The OUTCROSS= data set contains all the variables in the DATA= data set, plus new variables containing the posterior probabilities and the classification results of cross validation. The names of the new variables containing the posterior probabilities are constructed from the formatted values of the class levels. A new variable, _INTO_, with the same attributes as the CLASS variable, specifies the class to which each observation is assigned. When an observation is labeled as 'Other', the variable _INTO_ has a missing value. When you specify the CANONICAL option, the data set also contains new variables with canonical variable scores. The NCAN= option determines the number of new variables. The names of the new variables are constructed as described in the CANPREFIX= option. The new variables have mean zero and pooled within-class variance equal to one.

An OUTCROSS= data set cannot be created if the DATA= data set is not an ordinary SAS data set.

TESTOUT= Data Set

The TESTOUT= data set contains all the variables in the TESTDATA= data set, plus new variables containing the posterior probabilities and the classification results. The names of the new variables containing the posterior probabilities are formed from the formatted values of the class levels. A new variable, _INTO_, with the same attributes as the CLASS variable, gives the class to which each observation is assigned. If an observation is labeled as 'Other', the variable _INTO_ has a missing value. When you specify the CANONICAL option, the data set also contains new variables with canonical variable scores. The NCAN= option determines the number of new variables. The names of the new variables are formed as described in the CANPREFIX= option.

TESTOUTD= Data Set

The TESTOUTD= data set contains all the variables in the TESTDATA= data set, plus new variables containing the group-specific density estimates. The names of the new variables containing the density estimates are formed from the formatted values of the class levels.

OUTSTAT= Data Set

The OUTSTAT= data set is similar to the TYPE=CORR data set produced by the CORR procedure. The data set contains various statistics such as means, standard deviations, and correlations. For an example of an OUTSTAT= data set, see Example 31.3. When you specify the CANONICAL

option, canonical correlations, canonical structures, canonical coefficients, and means of canonical variables for each class are included in the data set.

If you specify METHOD=NORMAL, the output data set also includes coefficients of the discriminant functions, and the data set is TYPE=LINEAR (POOL=YES), TYPE=QUAD (POOL=NO), or TYPE=MIXED (POOL=TEST). If you specify METHOD=NPAR, this output data set is TYPE=CORR.

The OUTSTAT= data set contains the following variables:

- the BY variables, if any
- the CLASS variable
- _TYPE_, a character variable of length 8 that identifies the type of statistic
- _NAME_, a character variable of length 32 that identifies the row of the matrix, the name of the canonical variable, or the type of the discriminant function coefficients
- the quantitative variables—that is, those in the VAR statement, or, if there is no VAR statement, all numeric variables not listed in any other statement

The observations, as identified by the variable _TYPE_, have the following values:

TYPE	**Contents**
N	number of observations both for the total sample (CLASS variable missing) and within each class (CLASS variable present)
SUMWGT	sum of weights both for the total sample (CLASS variable missing) and within each class (CLASS variable present), if a WEIGHT statement is specified
MEAN	means both for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PRIOR	prior probability for each class
STDMEAN	total-standardized class means
PSTDMEAN	pooled within-class standardized class means
STD	standard deviations both for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PSTD	pooled within-class standard deviations
BSTD	between-class standard deviations
RSQUARED	univariate R squares
LNDETERM	the natural log of the determinant or the natural log of the quasi determinant of the within-class covariance matrix either pooled (CLASS variable missing) or not pooled (CLASS variable present)

The following kinds of observations are identified by the combination of the variables _TYPE_ and _NAME_. When the _TYPE_ variable has one of the following values, the _NAME_ variable identifies the row of the matrix:

TYPE	**Contents**
CSSCP	corrected SSCP matrix both for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PSSCP	pooled within-class corrected SSCP matrix
BSSCP	between-class SSCP matrix
COV	covariance matrix both for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PCOV	pooled within-class covariance matrix
BCOV	between-class covariance matrix
CORR	correlation matrix both for the total sample (CLASS variable missing) and within each class (CLASS variable present)
PCORR	pooled within-class correlation matrix
BCORR	between-class correlation matrix

When you request canonical discriminant analysis, the _NAME_ variable identifies a canonical variable, and _TYPE_ variable can have one of the following values:

TYPE	**Contents**
CANCORR	canonical correlations
STRUCTUR	canonical structure
BSTRUCT	between canonical structure
PSTRUCT	pooled within-class canonical structure
SCORE	standardized canonical coefficients
RAWSCORE	raw canonical coefficients
CANMEAN	means of the canonical variables for each class

When you specify METHOD=NORMAL, the _NAME_ variable identifies different types of coefficients in the discriminant function, and the _TYPE_ variable can have one of the following values:

TYPE	**Contents**
LINEAR	coefficients of the linear discriminant functions
QUAD	coefficients of the quadratic discriminant functions

The values of the _NAME_ variable are as follows:

NAME	**Contents**
variable names	quadratic coefficients of the quadratic discriminant functions (a symmetric matrix for each class)
LINEAR	linear coefficients of the discriminant functions
CONST	constant coefficients of the discriminant functions

Computational Resources

In the following discussion, let

n = number of observations in the training data set
v = number of variables
c = number of class levels
k = number of canonical variables
l = length of the CLASS variable

Memory Requirements

The amount of temporary storage required depends on the discriminant method used and the options specified. The least amount of temporary storage in bytes needed to process the data is approximately

$$c(32v + 3l + 128) + 8v^2 + 104v + 4l$$

A parametric method (METHOD=NORMAL) requires an additional temporary memory of $12v^2 + 100v$ bytes. When you specify the CROSSVALIDATE option, this temporary storage must be increased by $4v^2 + 44v$ bytes. When a nonparametric method (METHOD=NPAR) is used, an additional temporary storage of $10v^2 + 94v$ bytes is needed if you specify METRIC=FULL to evaluate the distances.

With the MANOVA option, the temporary storage must be increased by $8v^2 + 96v$ bytes. The CANONICAL option requires a temporary storage of $2v^2 + 94v + 8k(v + c)$ bytes. The POSTERR option requires a temporary storage of $8c^2 + 64c + 96$ bytes. Additional temporary storage is also required for classification summary and for each output data set.

Consider the following statements:

```
proc discrim manova;
   class gp;
   var x1 x2 x3;
run;
```

If the CLASS variable gp has a length of 8 and the input data set contains two class levels, the procedure requires a temporary storage of 1992 bytes. This includes 1104 bytes for processing data, 480 bytes for using a parametric method, and 408 bytes for specifying the MANOVA option.

Time Requirements

The following factors determine the time requirements of discriminant analysis:

- The time needed for reading the data and computing covariance matrices is proportional to nv^2. PROC DISCRIM must also look up each class level in the list. This is faster if the data are sorted by the CLASS variable. The time for looking up class levels is proportional to a value ranging from n to $n \ln(c)$.
- The time for inverting a covariance matrix is proportional to v^3.
- With a parametric method, the time required to classify each observation is proportional to cv for a linear discriminant function and cv^2 for a quadratic discriminant function. When you specify the CROSSVALIDATE option, the discriminant function is updated for each observation in the classification. A substantial amount of time is required.
- With a nonparametric method, the data are stored in a tree structure (Friedman, Bentley, and Finkel 1977). The time required to organize the observations into the tree structure is proportional to $nv \ln(n)$. The time for performing each tree search is proportional to $\ln(n)$. When you specify the normal KERNEL= option, all observations in the training sample contribute to the density estimation and more computer time is needed.
- The time required for the canonical discriminant analysis is proportional to v^3.

Each of the preceding factors has a different machine-dependent constant of proportionality.

Displayed Output

The displayed output from PROC DISCRIM includes the class level information table. For each level of the classification variable, the following information is provided: the output data set variable name, frequency sum, weight sum, proportion of the total sample, and prior probability.

The optional output from PROC DISCRIM includes the following:

- Within-class SSCP matrices for each group
- Pooled within-class SSCP matrix
- Between-class SSCP matrix
- Total-sample SSCP matrix
- Within-class covariance matrices, $\mathbf{S}_t$, for each group
- Pooled within-class covariance matrix, $\mathbf{S}_p$
- Between-class covariance matrix, equal to the between-class SSCP matrix divided by $n(c-1)/c$, where n is the number of observations and c is the number of classes

- Total-sample covariance matrix
- Within-class correlation coefficients and Pr $>$ $|r|$ to test the hypothesis that the within-class population correlation coefficients are zero
- Pooled within-class correlation coefficients and Pr $>$ $|r|$ to test the hypothesis that the partial population correlation coefficients are zero
- Between-class correlation coefficients and Pr $>$ $|r|$ to test the hypothesis that the between-class population correlation coefficients are zero
- Total-sample correlation coefficients and Pr $>$ $|r|$ to test the hypothesis that the total population correlation coefficients are zero
- Simple statistics, including N (the number of observations), sum, mean, variance, and standard deviation both for the total sample and within each class
- Total-sample standardized class means, obtained by subtracting the grand mean from each class mean and dividing by the total-sample standard deviation
- Pooled within-class standardized class means, obtained by subtracting the grand mean from each class mean and dividing by the pooled within-class standard deviation
- Pairwise squared distances between groups
- Univariate test statistics, including total-sample standard deviations, pooled within-class standard deviations, between-class standard deviations, R square, $R^2/(1-R^2)$, F, and Pr $>$ F (univariate F values and probability levels for one-way analyses of variance)
- Multivariate statistics and F approximations, including Wilks' lambda, Pillai's trace, Hotelling-Lawley trace, and Roy's greatest root with F approximations, numerator and denominator degrees of freedom (Num DF and Den DF), and probability values (Pr $>$ F). Each of these four multivariate statistics tests the hypothesis that the class means are equal in the population. See the section "Multivariate Tests" on page 102 in Chapter 4, "Introduction to Regression Procedures," for more information.

If you specify METHOD=NORMAL, the following three statistics are displayed:

- Covariance matrix information, including covariance matrix rank and natural log of determinant of the covariance matrix for each group (POOL=TEST, POOL=NO) and for the pooled within-group (POOL=TEST, POOL=YES)
- Optionally, test of homogeneity of within covariance matrices (the results of a chi-square test of homogeneity of the within-group covariance matrices) (Morrison 1976; Kendall, Stuart, and Ord 1983; Anderson 1984)
- Pairwise generalized squared distances between groups

If the CANONICAL option is specified, the displayed output contains these statistics:

- Canonical correlations
- Adjusted canonical correlations (Lawley 1959). These are asymptotically less biased than the raw correlations and can be negative. The adjusted canonical correlations might not be computable and are displayed as missing values if two canonical correlations are nearly equal or if some are close to zero. A missing value is also displayed if an adjusted canonical correlation is larger than a previous adjusted canonical correlation.
- Approximate standard error of the canonical correlations
- Squared canonical correlations
- Eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$. Each eigenvalue is equal to $\rho^2/(1-\rho^2)$, where ρ^2 is the corresponding squared canonical correlation and can be interpreted as the ratio of between-class variation to within-class variation for the corresponding canonical variable. The table includes eigenvalues, differences between successive eigenvalues, proportion of the sum of the eigenvalues, and cumulative proportion.
- Likelihood ratio for the hypothesis that the current canonical correlation and all smaller ones are zero in the population. The likelihood ratio for all canonical correlations equals Wilks' lambda.
- Approximate F statistic based on Rao's approximation to the distribution of the likelihood ratio (Rao 1973, p. 556; Kshirsagar 1972, p. 326)
- Numerator degrees of freedom (Num DF), denominator degrees of freedom (Den DF), and $\Pr > F$, the probability level associated with the F statistic

The following statistic concerns the classification criterion:

- the linear discriminant function, but only if you specify METHOD=NORMAL and the pooled covariance matrix is used to calculate the (generalized) squared distances

When the input DATA= data set is an ordinary SAS data set, the displayed output includes the following:

- Optionally, the resubstitution results including the observation number (if an ID statement is included, the values of the ID variable are displayed instead of the observation number), the actual group for the observation, the group into which the developed criterion would classify it, and the posterior probability of membership in each group
- Resubstitution summary, a summary of the performance of the classification criterion based on resubstitution classification results
- Error count estimate of the resubstitution classification results
- Optionally, posterior probability error rate estimates of the resubstitution classification results

If you specify the CROSSVALIDATE option, the displayed output contains these statistics:

- Optionally, the cross validation results including the observation number (if an ID statement is included, the values of the ID variable are displayed instead of the observation number), the actual group for the observation, the group into which the developed criterion would classify it, and the posterior probability of membership in each group
- Cross validation summary, a summary of the performance of the classification criterion based on cross validation classification results
- Error count estimate of the cross validation classification results
- Optionally, posterior probability error rate estimates of the cross validation classification results

If you specify the TESTDATA= option, the displayed output contains these statistics:

- Optionally, the classification results including the observation number (if a TESTID statement is included, the values of the ID variable are displayed instead of the observation number), the actual group for the observation (if a TESTCLASS statement is included), the group into which the developed criterion would classify it, and the posterior probability of membership in each group
- Classification summary, a summary of the performance of the classification criterion
- Error count estimate of the test data classification results
- Optionally, posterior probability error rate estimates of the test data classification results

ODS Table Names

PROC DISCRIM assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 31.2. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 31.2 ODS Tables Produced by PROC DISCRIM

ODS Table Name	Description	PROC DISCRIM Option
ANOVA	Univariate statistics	ANOVA
AvePostCrossVal	Average posterior probabilities, cross validation	POSTERR & CROSSVALIDATE
AvePostResub	Average posterior probabilities, resubstitution	POSTERR
AvePostTestClass	Average posterior probabilities, test classification	POSTERR & TEST=
AveRSquare	Average R-square	ANOVA
BCorr	Between-class correlations	BCORR
BCov	Between-class covariances	BCOV
BSSCP	Between-class SSCP matrix	BSSCP
BStruc	Between canonical structure	CANONICAL
CanCorr	Canonical correlations	CANONICAL
CanonicalMeans	Class means on canonical variables	CANONICAL
ChiSq	Chi-square information	POOL=TEST
ClassifiedCrossVal	Number of observations and percent classified, cross validation	CROSSVALIDATE
ClassifiedResub	Number of observations and percent classified, resubstitution	default
ClassifiedTestClass	Number of observations and percent classified, test classification	TEST=
Counts	Number of observations, variables, classes, df	default
CovDF	DF for covariance matrices, not displayed	any *COV option
Dist	Squared distances	DISTANCE
DistFValues	F values based on squared distances	DISTANCE
DistGeneralized	Generalized squared distances	default
DistProb	Probabilities for F values from squared distances	DISTANCE
ErrorCrossVal	Error count estimates, cross validation	CROSSVALIDATE
ErrorResub	Error count estimates, resubstitution	default

Table 31.2 *continued*

ODS Table Name	Description	PROC DISCRIM Option
ErrorTestClass	Error count estimates, test classification	TEST=
Levels	Class level information	default
LinearDiscFunc	Linear discriminant function	POOL=YES
LogDet	Log determinant of the covariance matrix	default
MultStat	MANOVA	MANOVA
PCoef	Pooled standard canonical coefficients	CANONICAL
PCorr	Pooled within-class correlations	PCORR
PCov	Pooled within-class covariances	PCOV
PSSCP	Pooled within-class SSCP matrix	PSSCP
PStdMeans	Pooled standardized class means	STDMEAN
PStruc	Pooled within canonical structure	CANONICAL
PostCrossVal	Posterior probabilities, cross validation	CROSSLIST or CROSSLISTERR
PostErrCrossVal	Posterior error estimates, cross validation	POSTERR & CROSSVALIDATE
PostErrResub	Posterior error estimates, resubstitution	POSTERR
PostErrTestClass	Posterior error estimates, test classification	POSTERR & TEST=
PostResub	Posterior probabilities, resubstitution	LIST or LISTERR
PostTestClass	Posterior probabilities, test classification	TESTLIST or TESTLISTERR
RCoef	Raw canonical coefficients	CANONICAL
SimpleStatistics	Simple statistics	SIMPLE
TCoef	Total-sample standard canonical coefficients	CANONICAL
TCorr	Total-sample correlations	TCORR
TCov	Total-sample covariances	TCOV
TSSCP	Total-sample SSCP matrix	TSSCP
TStdMeans	Total standardized class means	STDMEAN
TStruc	Total canonical structure	CANONICAL
WCorr	Within-class correlations	WCORR
WCov	Within-class covariances	WCOV
WSSCP	Within-class SSCP matrices	WSSCP

Examples: DISCRIM Procedure

The iris data published by Fisher (1936) are widely used for examples in discriminant analysis and cluster analysis. The sepal length, sepal width, petal length, and petal width are measured in millimeters on 50 iris specimens from each of three species: *Iris setosa*, *I. versicolor*, and *I. virginica*. The iris data are used in Example 31.1 through Example 31.3.

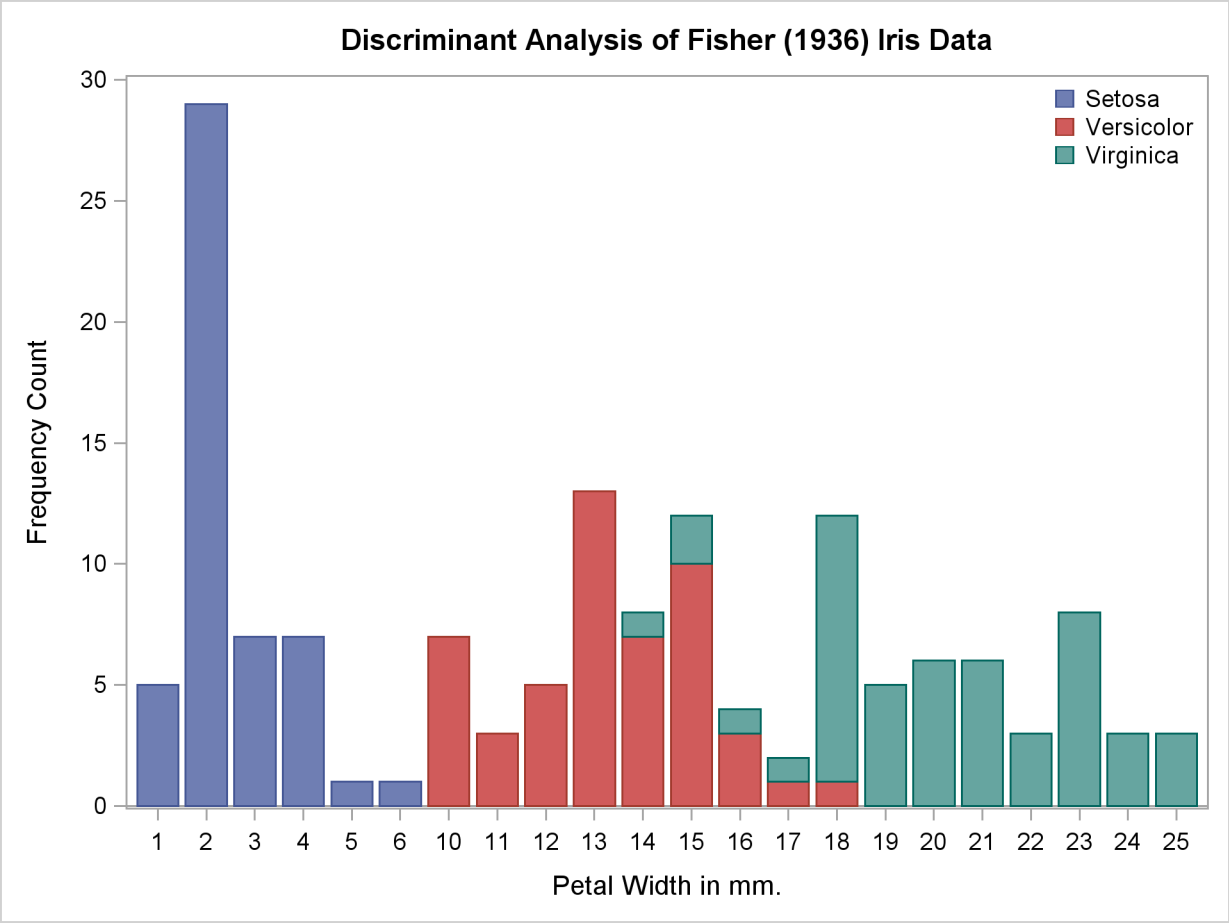
Example 31.1: Univariate Density Estimates and Posterior Probabilities

In this example, several discriminant analyses are run with a single quantitative variable, petal width, so that density estimates and posterior probabilities can be plotted easily. The example produces Output 31.1.1 through Output 31.1.5. ODS Graphics is used to display the sample distribution of petal width in the three species. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." Note the overlap between the species *I. versicolor* and *I. virginica* that the bar chart shows. The following statements produce Output 31.1.1:

```
title 'Discriminant Analysis of Fisher (1936) Iris Data';

proc format;
   value specname
      1='Setosa    '
      2='Versicolor'
      3='Virginica ';
run;

data iris;
   input SepalLength SepalWidth PetalLength PetalWidth
         Species @@;
   format Species specname.;
   label SepalLength='Sepal Length in mm.'
         SepalWidth ='Sepal Width in mm.'
         PetalLength='Petal Length in mm.'
         PetalWidth ='Petal Width in mm.';
   datalines;
50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3
63 28 51 15 3 46 34 14 03 1 69 31 51 23 3 62 22 45 15 2
59 32 48 18 2 46 36 10 02 1 61 30 46 14 2 60 27 51 16 2

... more lines ...

63 33 60 25 3 53 37 15 02 1
;

proc freq data=iris noprint;
   tables petalwidth * species / out=freqout;
run;
```

```
proc sgplot data=freqout;
   vbar petalwidth / response=count group=species;
   keylegend / location=inside position=ne noborder across=1;
run;
```

Output 31.1.1 Sample Distribution of Petal Width in Three Species



In order to plot the density estimates and posterior probabilities, a data set called plotdata is created containing equally spaced values from –5 to 30, covering the range of petal width with a little to spare on each end. The plotdata data set is used with the TESTDATA= option in PROC DISCRIM. The following statements make the data set:

```
data plotdata;
   do PetalWidth=-5 to 30 by 0.5;
      output;
   end;
run;
```

The same plots are produced after each discriminant analysis, so macros are used to reduce the amount of typing required. The macros use two data sets. The data set plotd, containing density estimates, is created by the TESTOUTD= option in PROC DISCRIM. The data set plotp, containing posterior probabilities, is created by the TESTOUT= option. For each data set, the macros remove uninteresting values (near zero) and create an overlay plot showing all three species in a single plot.

The following statements create the macros:

```
%macro plotden;
   title3 'Plot of Estimated Densities';

   data plotd2;
      set plotd;
      if setosa     < .002 then setosa     = .;
      if versicolor < .002 then versicolor = .;
      if virginica  < .002 then virginica  = .;
      g = 'Setosa    '; Density = setosa;     output;
      g = 'Versicolor'; Density = versicolor; output;
      g = 'Virginica '; Density = virginica;  output;
      label PetalWidth='Petal Width in mm.';
   run;

   proc sgplot data=plotd2;
      series y=Density x=PetalWidth / group=g;
      discretelegend;
   run;
   %mend;

%macro plotprob;
   title3 'Plot of Posterior Probabilities';

   data plotp2;
      set plotp;
      if setosa     < .01 then setosa     = .;
      if versicolor < .01 then versicolor = .;
      if virginica  < .01 then virginica  = .;
      g = 'Setosa    '; Probability = setosa;     output;
      g = 'Versicolor'; Probability = versicolor; output;
      g = 'Virginica '; Probability = virginica;  output;
      label PetalWidth='Petal Width in mm.';
   run;

   proc sgplot data=plotp2;
      series y=Probability x=PetalWidth / group=g;
      discretelegend;
   run;
%mend;
```

The first analysis uses normal-theory methods (METHOD=NORMAL) assuming equal variances (POOL=YES) in the three classes. The NOCLASSIFY option suppresses the resubstitution classification results of the input data set observations. The CROSSLISTERR option lists the observations that are misclassified under cross validation and displays cross validation error-rate estimates. The following statements produce Output 31.1.2:

```
title2 'Using Normal Density Estimates with Equal Variance';

proc discrim data=iris method=normal pool=yes
             testdata=plotdata testout=plotp testoutd=plotd
             short noclassify crosslisterr;
   class Species;
   var PetalWidth;
run;

%plotden;
%plotprob;
```

Output 31.1.2 Normal Density Estimates with Equal Variance

```
              Discriminant Analysis of Fisher (1936) Iris Data
             Using Normal Density Estimates with Equal Variance

                           The DISCRIM Procedure

        Total Sample Size      150          DF Total               149
        Variables                1          DF Within Classes      147
        Classes                  3          DF Between Classes       2

                  Number of Observations Read           150
                  Number of Observations Used           150

                           Class Level Information

             Variable                                              Prior
Species      Name          Frequency       Weight    Proportion    Probability

Setosa       Setosa               50      50.0000      0.333333       0.333333
Versicolor   Versicolor           50      50.0000      0.333333       0.333333
Virginica    Virginica            50      50.0000      0.333333       0.333333

              Discriminant Analysis of Fisher (1936) Iris Data
             Using Normal Density Estimates with Equal Variance

                           The DISCRIM Procedure
            Classification Results for Calibration Data: WORK.IRIS
          Cross-validation Results using Linear Discriminant Function

                Posterior Probability of Membership in Species

        From          Classified
  Obs   Species       into Species        Setosa    Versicolor     Virginica

    5   Virginica     Versicolor *        0.0000        0.9610        0.0390
    9   Versicolor    Virginica  *        0.0000        0.0952        0.9048
   57   Virginica     Versicolor *        0.0000        0.9940        0.0060
   78   Virginica     Versicolor *        0.0000        0.8009        0.1991
   91   Virginica     Versicolor *        0.0000        0.9610        0.0390
  148   Versicolor    Virginica  *        0.0000        0.3828        0.6172

                       * Misclassified observation
```

Output 31.1.2 *continued*

Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Equal Variance

The DISCRIM Procedure
Classification Summary for Calibration Data: WORK.IRIS
Cross-validation Summary using Linear Discriminant Function

Number of Observations and Percent Classified into Species

From Species	Setosa	Versicolor	Virginica	Total
Setosa	50	0	0	50
	100.00	0.00	0.00	100.00
Versicolor	0	48	2	50
	0.00	96.00	4.00	100.00
Virginica	0	4	46	50
	0.00	8.00	92.00	100.00
Total	50	52	48	150
	33.33	34.67	32.00	100.00
Priors	0.33333	0.33333	0.33333	

Error Count Estimates for Species

	Setosa	Versicolor	Virginica	Total
Rate	0.0000	0.0400	0.0800	0.0400
Priors	0.3333	0.3333	0.3333	

Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Equal Variance

The DISCRIM Procedure
Classification Summary for Test Data: WORK.PLOTDATA
Classification Summary using Linear Discriminant Function

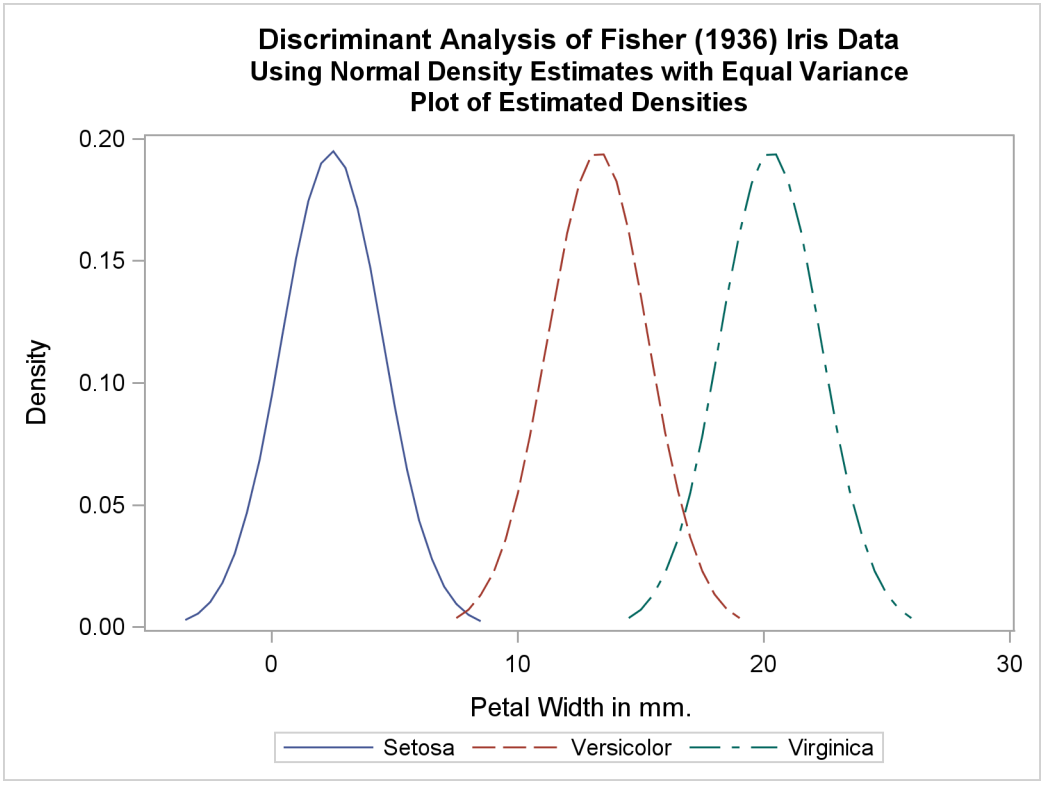
Observation Profile for Test Data

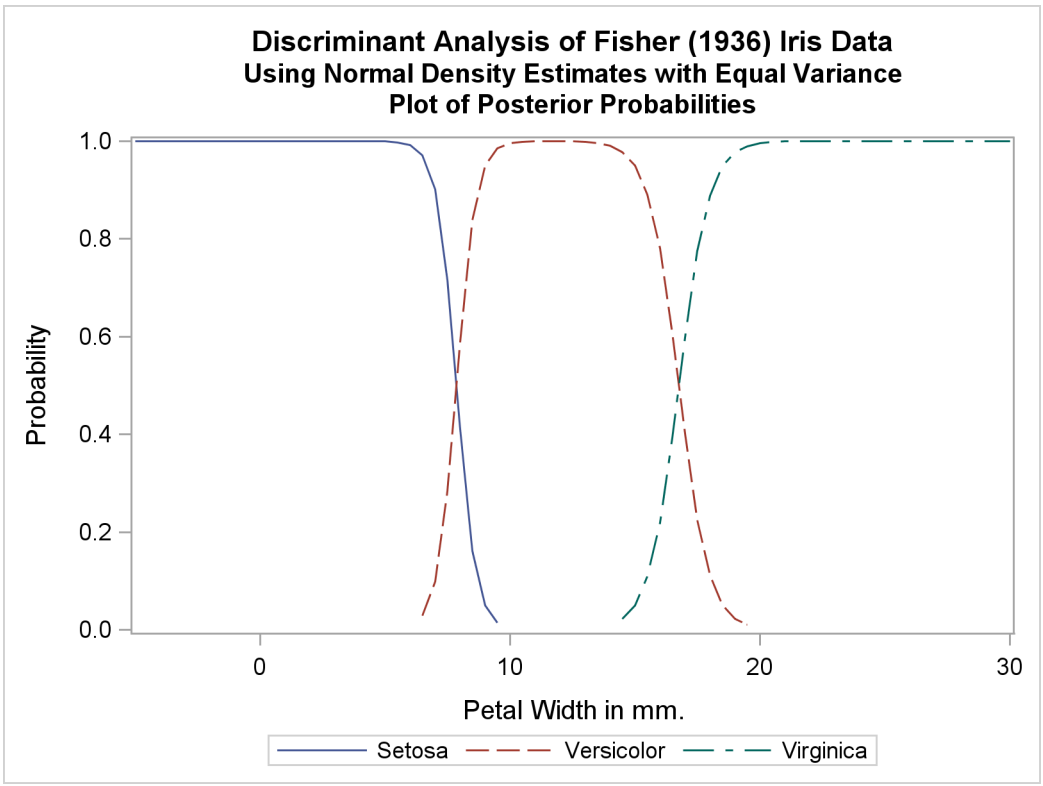
Number of Observations Read	71
Number of Observations Used	71

Number of Observations and Percent Classified into Species

	Setosa	Versicolor	Virginica	Total
Total	26	18	27	71
	36.62	25.35	38.03	100.00
Priors	0.33333	0.33333	0.33333	

Output 31.1.2 *continued*





The next analysis uses normal-theory methods assuming unequal variances (POOL=NO) in the three classes. The following statements produce Output 31.1.3:

```
title2 'Using Normal Density Estimates with Unequal Variance';

proc discrim data=iris method=normal pool=no
             testdata=plotdata testout=plotp testoutd=plotd
             short noclassify crosslisterr;
   class Species;
   var PetalWidth;
run;

%plotden;
%plotprob;
```

Output 31.1.3 Normal Density Estimates with Unequal Variance

Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Unequal Variance

The DISCRIM Procedure

Total Sample Size	150	DF Total	149
Variables	1	DF Within Classes	147
Classes	3	DF Between Classes	2

Number of Observations Read	150
Number of Observations Used	150

Class Level Information

Species	Variable Name	Frequency	Weight	Proportion	Prior Probability
Setosa	Setosa	50	50.0000	0.333333	0.333333
Versicolor	Versicolor	50	50.0000	0.333333	0.333333
Virginica	Virginica	50	50.0000	0.333333	0.333333

Output 31.1.3 *continued*

Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Unequal Variance

The DISCRIM Procedure
Classification Results for Calibration Data: WORK.IRIS
Cross-validation Results using Quadratic Discriminant Function

Posterior Probability of Membership in Species

Obs	From Species	Classified into Species	Setosa	Versicolor	Virginica
5	Virginica	Versicolor *	0.0000	0.8740	0.1260
9	Versicolor	Virginica *	0.0000	0.0686	0.9314
42	Setosa	Versicolor *	0.4923	0.5073	0.0004
57	Virginica	Versicolor *	0.0000	0.9602	0.0398
78	Virginica	Versicolor *	0.0000	0.6558	0.3442
91	Virginica	Versicolor *	0.0000	0.8740	0.1260
148	Versicolor	Virginica *	0.0000	0.2871	0.7129

* Misclassified observation

Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Unequal Variance

The DISCRIM Procedure
Classification Summary for Calibration Data: WORK.IRIS
Cross-validation Summary using Quadratic Discriminant Function

Number of Observations and Percent Classified into Species

From Species	Setosa	Versicolor	Virginica	Total
Setosa	49	1	0	50
	98.00	2.00	0.00	100.00
Versicolor	0	48	2	50
	0.00	96.00	4.00	100.00
Virginica	0	4	46	50
	0.00	8.00	92.00	100.00
Total	49	53	48	150
	32.67	35.33	32.00	100.00
Priors	0.33333	0.33333	0.33333	

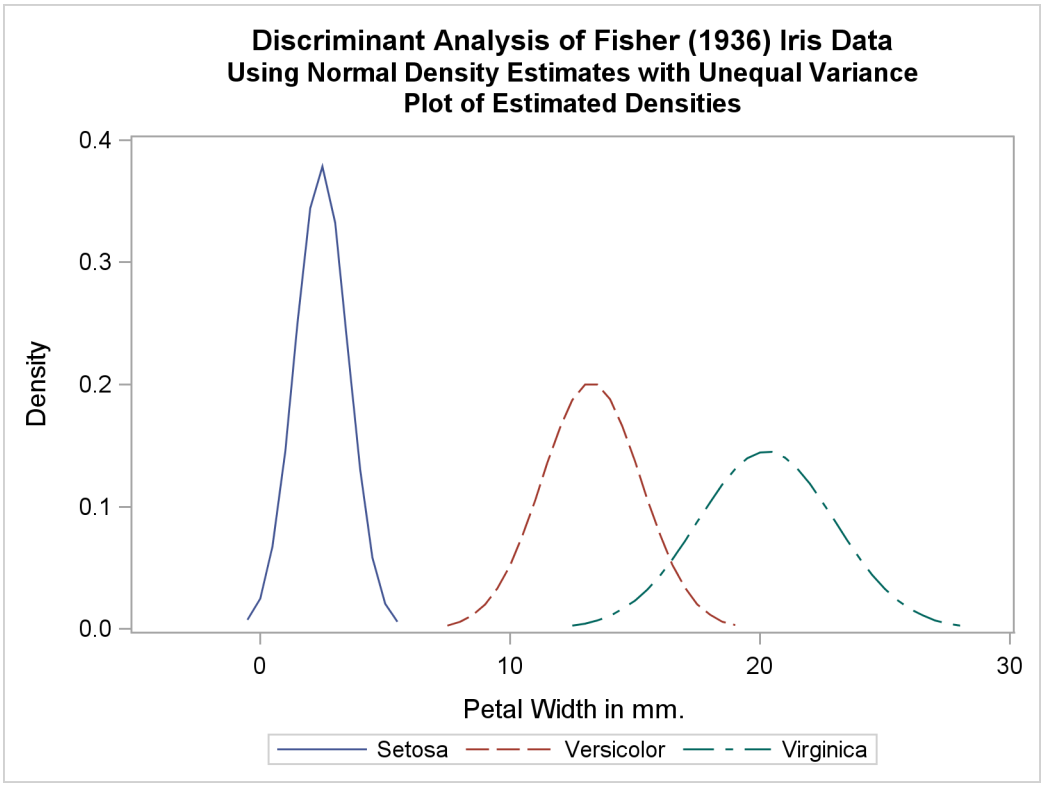
Error Count Estimates for Species

	Setosa	Versicolor	Virginica	Total
Rate	0.0200	0.0400	0.0800	0.0467
Priors	0.3333	0.3333	0.3333	

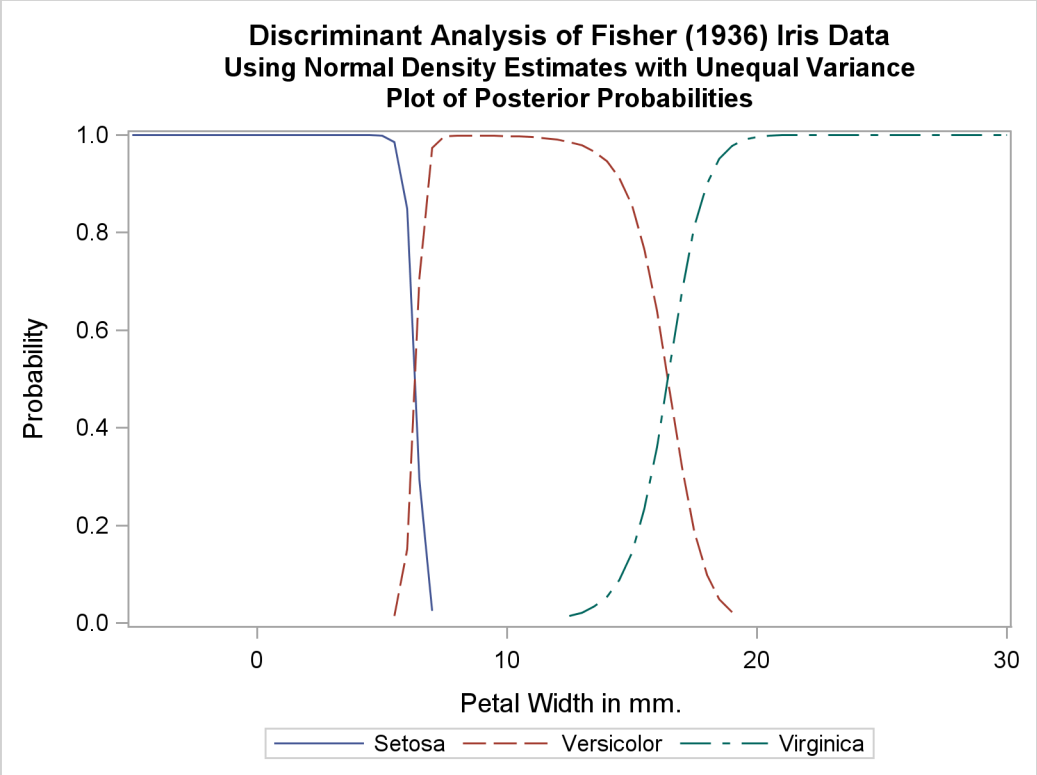
Output 31.1.3 *continued*

```
        Discriminant Analysis of Fisher (1936) Iris Data
      Using Normal Density Estimates with Unequal Variance

                      The DISCRIM Procedure
          Classification Summary for Test Data: WORK.PLOTDATA
       Classification Summary using Quadratic Discriminant Function

                    Observation Profile for Test Data

                  Number of Observations Read            71
                  Number of Observations Used            71

          Number of Observations and Percent Classified into Species

                       Setosa      Versicolor      Virginica        Total

Total                      23              20             28           71
                        32.39           28.17          39.44       100.00

Priors                0.33333         0.33333        0.33333
```

Output 31.1.3 *continued*



Output 31.1.3 *continued*



Two more analyses are run with nonparametric methods (METHOD=NPAR), specifically kernel density estimates with normal kernels (KERNEL=NORMAL). The first of these uses equal bandwidths (smoothing parameters) (POOL=YES) in each class. The use of equal bandwidths does not constrain the density estimates to be of equal variance. The value of the radius parameter that, assuming normality, minimizes an approximate mean integrated square error is 0.48 (see the section "Nonparametric Methods" on page 1403). Choosing $r = 0.4$ gives a more detailed look at the irregularities in the data. The following statements produce Output 31.1.4:

```
title2 'Using Kernel Density Estimates with Equal Bandwidth';

proc discrim data=iris method=npar kernel=normal
                r=.4 pool=yes
             testdata=plotdata testout=plotp
                testoutd=plotd
             short noclassify crosslisterr;
   class Species;
   var PetalWidth;
run;

%plotden;
%plotprob;
```

Output 31.1.4 Kernel Density Estimates with Equal Bandwidth

Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Equal Bandwidth

The DISCRIM Procedure

Total Sample Size	150	DF Total	149
Variables	1	DF Within Classes	147
Classes	3	DF Between Classes	2

Number of Observations Read	150
Number of Observations Used	150

Class Level Information

Species	Variable Name	Frequency	Weight	Proportion	Prior Probability
Setosa	Setosa	50	50.0000	0.333333	0.333333
Versicolor	Versicolor	50	50.0000	0.333333	0.333333
Virginica	Virginica	50	50.0000	0.333333	0.333333

Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Equal Bandwidth

The DISCRIM Procedure
Classification Results for Calibration Data: WORK.IRIS
Cross-validation Results using Normal Kernel Density

Posterior Probability of Membership in Species

Obs	From Species	Classified into Species	Setosa	Versicolor	Virginica
5	Virginica	Versicolor *	0.0000	0.8827	0.1173
9	Versicolor	Virginica *	0.0000	0.0438	0.9562
57	Virginica	Versicolor *	0.0000	0.9472	0.0528
78	Virginica	Versicolor *	0.0000	0.8061	0.1939
91	Virginica	Versicolor *	0.0000	0.8827	0.1173
148	Versicolor	Virginica *	0.0000	0.2586	0.7414

* Misclassified observation

Output 31.1.4 *continued*

Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Equal Bandwidth

The DISCRIM Procedure
Classification Summary for Calibration Data: WORK.IRIS
Cross-validation Summary using Normal Kernel Density

Number of Observations and Percent Classified into Species

From Species	Setosa	Versicolor	Virginica	Total
Setosa	50	0	0	50
	100.00	0.00	0.00	100.00
Versicolor	0	48	2	50
	0.00	96.00	4.00	100.00
Virginica	0	4	46	50
	0.00	8.00	92.00	100.00
Total	50	52	48	150
	33.33	34.67	32.00	100.00
Priors	0.33333	0.33333	0.33333	

Error Count Estimates for Species

	Setosa	Versicolor	Virginica	Total
Rate	0.0000	0.0400	0.0800	0.0400
Priors	0.3333	0.3333	0.3333	

Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Equal Bandwidth

The DISCRIM Procedure
Classification Summary for Test Data: WORK.PLOTDATA
Classification Summary using Normal Kernel Density

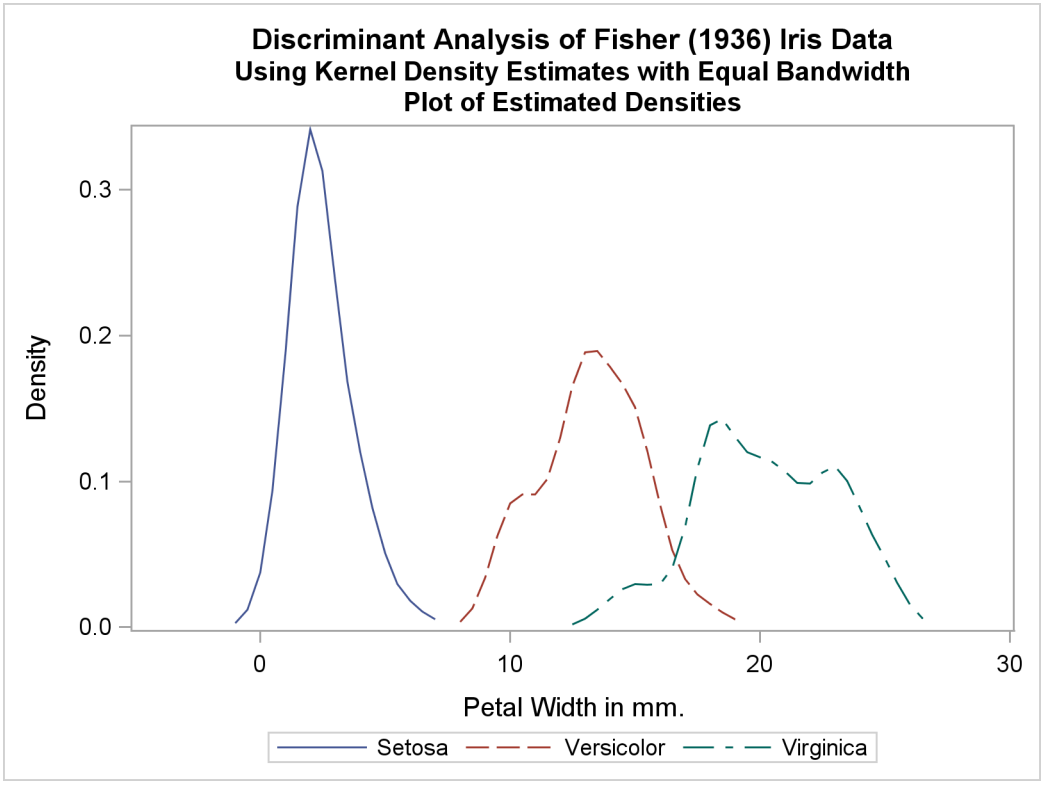
Observation Profile for Test Data

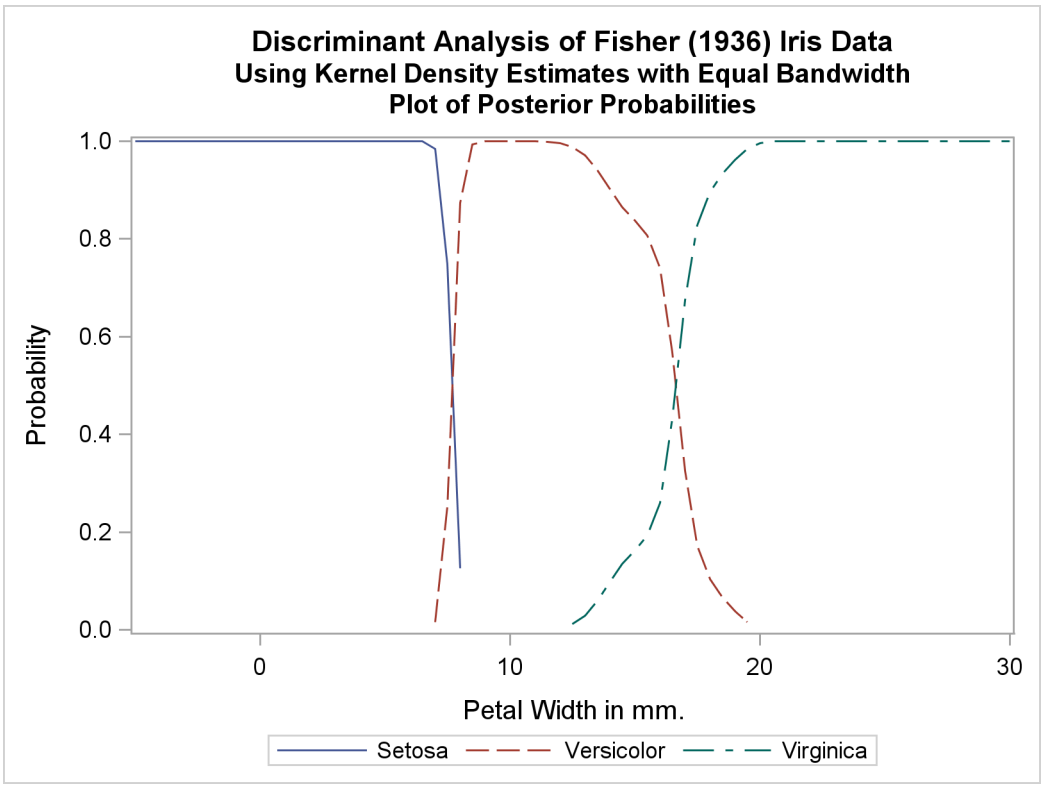
Number of Observations Read	71
Number of Observations Used	71

Number of Observations and Percent Classified into Species

	Setosa	Versicolor	Virginica	Total
Total	26	18	27	71
	36.62	25.35	38.03	100.00
Priors	0.33333	0.33333	0.33333	

Output 31.1.4 *continued*





Another nonparametric analysis is run with unequal bandwidths (POOL=NO). The following statements produce Output 31.1.5:

```
title2 'Using Kernel Density Estimates with Unequal Bandwidth';

proc discrim data=iris method=npar kernel=normal
                r=.4 pool=no
             testdata=plotdata testout=plotp
                testoutd=plotd
             short noclassify crosslisterr;
   class Species;
   var PetalWidth;
run;

%plotden;
%plotprob;
```

Output 31.1.5 Kernel Density Estimates with Unequal Bandwidth

Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Unequal Bandwidth

The DISCRIM Procedure

Total Sample Size	150	DF Total	149
Variables	1	DF Within Classes	147
Classes	3	DF Between Classes	2

Number of Observations Read	150
Number of Observations Used	150

Class Level Information

Species	Variable Name	Frequency	Weight	Proportion	Prior Probability
Setosa	Setosa	50	50.0000	0.333333	0.333333
Versicolor	Versicolor	50	50.0000	0.333333	0.333333
Virginica	Virginica	50	50.0000	0.333333	0.333333

Output 31.1.5 *continued*

Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Unequal Bandwidth

The DISCRIM Procedure
Classification Results for Calibration Data: WORK.IRIS
Cross-validation Results using Normal Kernel Density

Posterior Probability of Membership in Species

Obs	From Species	Classified into Species	Setosa	Versicolor	Virginica
5	Virginica	Versicolor *	0.0000	0.8805	0.1195
9	Versicolor	Virginica *	0.0000	0.0466	0.9534
57	Virginica	Versicolor *	0.0000	0.9394	0.0606
78	Virginica	Versicolor *	0.0000	0.7193	0.2807
91	Virginica	Versicolor *	0.0000	0.8805	0.1195
148	Versicolor	Virginica *	0.0000	0.2275	0.7725

* Misclassified observation

Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Unequal Bandwidth

The DISCRIM Procedure
Classification Summary for Calibration Data: WORK.IRIS
Cross-validation Summary using Normal Kernel Density

Number of Observations and Percent Classified into Species

From Species	Setosa	Versicolor	Virginica	Total
Setosa	50	0	0	50
	100.00	0.00	0.00	100.00
Versicolor	0	48	2	50
	0.00	96.00	4.00	100.00
Virginica	0	4	46	50
	0.00	8.00	92.00	100.00
Total	50	52	48	150
	33.33	34.67	32.00	100.00
Priors	0.33333	0.33333	0.33333	

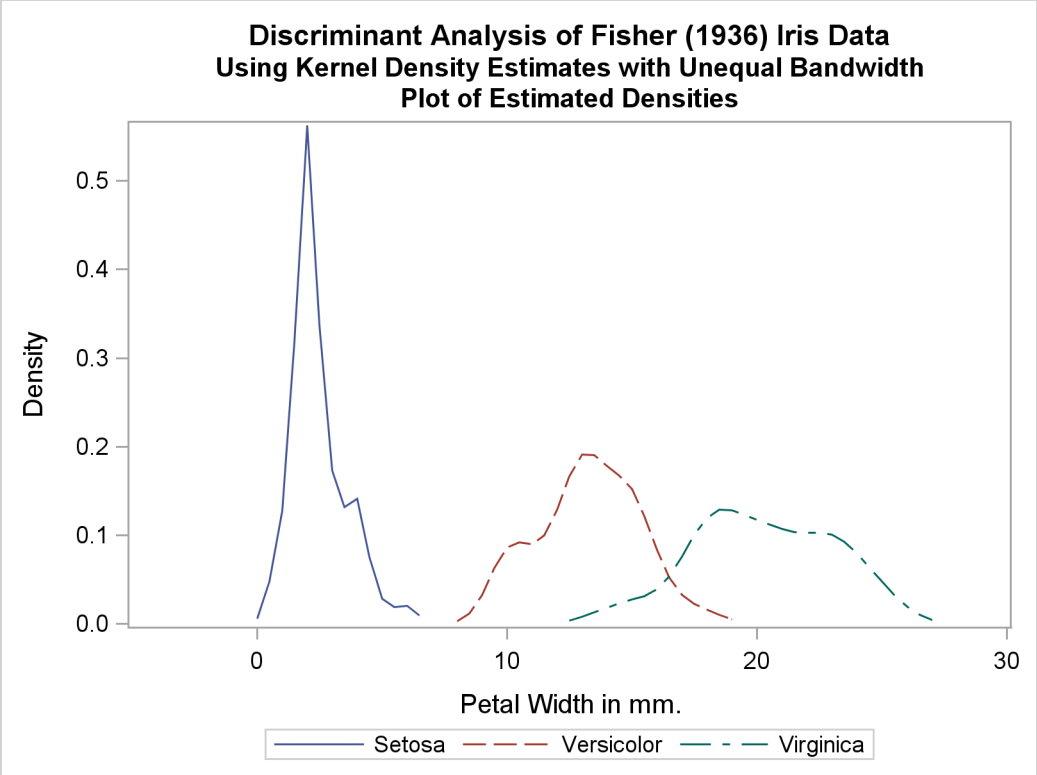
Error Count Estimates for Species

	Setosa	Versicolor	Virginica	Total
Rate	0.0000	0.0400	0.0800	0.0400
Priors	0.3333	0.3333	0.3333	

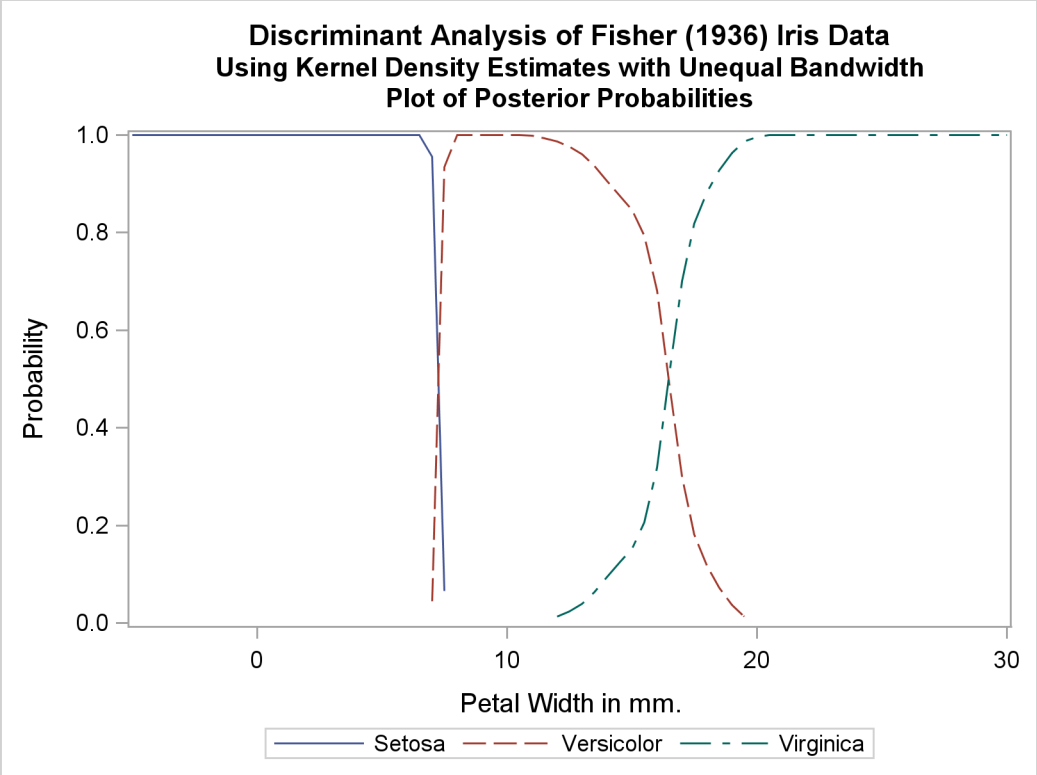
Output 31.1.5 *continued*

```
Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Unequal Bandwidth

The DISCRIM Procedure
Classification Summary for Test Data: WORK.PLOTDATA
Classification Summary using Normal Kernel Density

Observation Profile for Test Data

Number of Observations Read            71
Number of Observations Used            71

Number of Observations and Percent Classified into Species

                  Setosa      Versicolor      Virginica         Total

Total                 25              18             28            71
                   35.21           25.35          39.44        100.00

Priors          0.33333         0.33333        0.33333
```

Output 31.1.5 *continued*



Output 31.1.5 *continued*



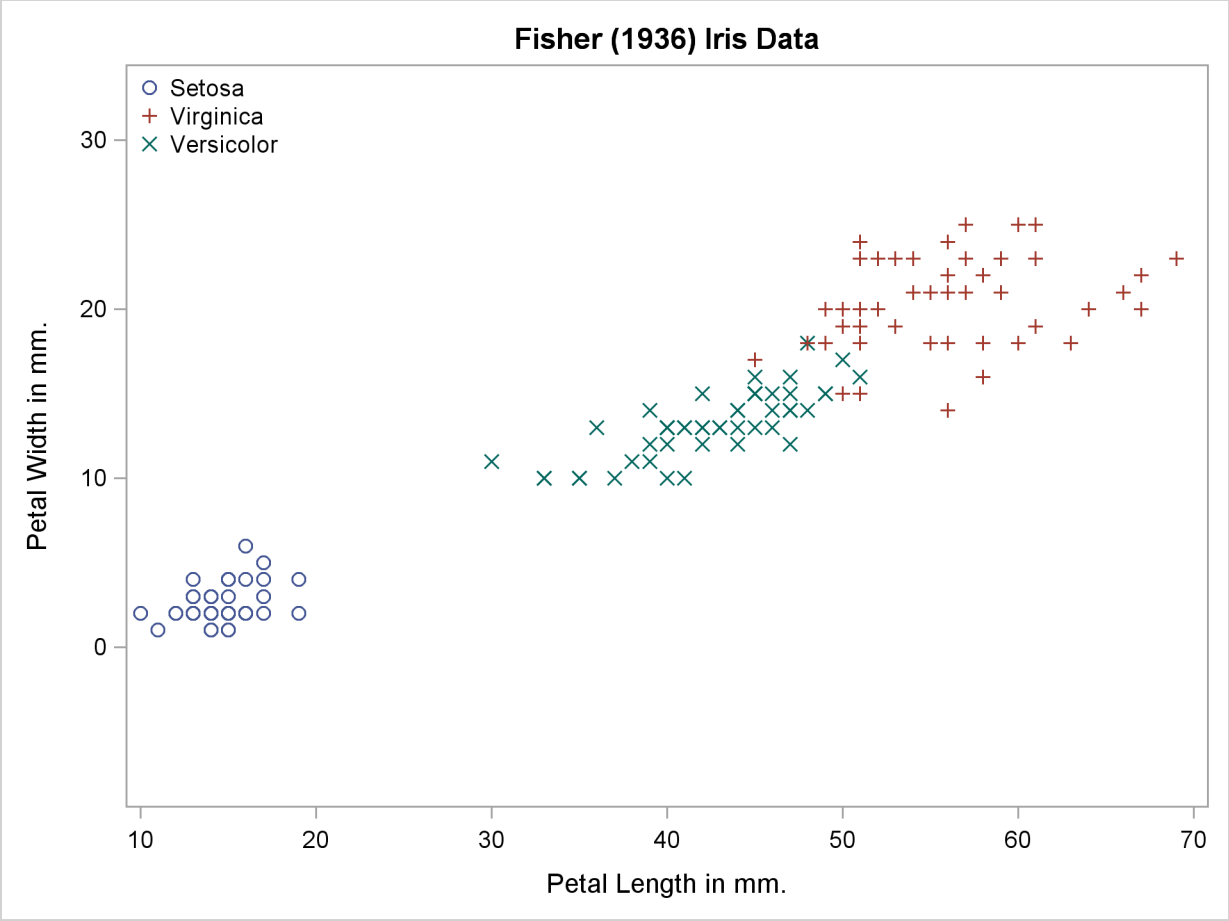
Example 31.2: Bivariate Density Estimates and Posterior Probabilities

In this example, four more discriminant analyses of iris data are run with two quantitative variables: petal width and petal length. The following statements produce Output 31.2.1 through Output 31.2.5:

```
proc template;
   define statgraph scatter;
      begingraph;
         entrytitle 'Fisher (1936) Iris Data';
         layout overlayequated / equatetype=fit;
            scatterplot x=petallength y=petalwidth /
                        group=species name='iris';
            layout gridded / autoalign=(topleft);
               discretelegend 'iris' / border=false opaque=false;
            endlayout;
         endlayout;
      endgraph;
   end;
run;

proc sgrender data=iris template=scatter;
run;
```

The scatter plot in Output 31.2.1 shows the joint sample distribution.

Output 31.2.1 Joint Sample Distribution of Petal Width and Petal Length in Three Species



Another data set is created for plotting, containing a grid of points suitable for contour plots. The following statements create the data set:

```
data plotdata;
   do PetalLength = -2 to 72 by 0.5;
      do PetalWidth= - 5 to 32 by 0.5;
         output;
      end;
   end;
run;
```

Three macros are defined as follows to make contour plots of density estimates, posterior probabilities, and classification results:

```
%let close = thresholdmin=0 thresholdmax=0 offsetmin=0 offsetmax=0;
%let close = xaxisopts=(&close) yaxisopts=(&close);

proc template;
   define statgraph contour;
      begingraph;
         layout overlayequated / equatetype=equate &close;
            contourplotparm x=petallength y=petalwidth z=z /
                            contourtype=fill nhint=30;
            scatterplot x=pl y=pw / group=species name='iris'
                        includemissinggroup=false primary=true;
            layout gridded / autoalign=(topleft);
               discretelegend 'iris' / border=false opaque=false;
            endlayout;
         endlayout;
      endgraph;
   end;
run;

%macro contden;
   data contour(keep=PetalWidth PetalLength species z pl pw);
      merge plotd(in=d) iris(keep=PetalWidth PetalLength species
                             rename=(PetalWidth=pw PetalLength=pl));
      if d then z = max(setosa,versicolor,virginica);
   run;

   title3 'Plot of Estimated Densities';

   proc sgrender data=contour template=contour;
   run;
%mend;

%macro contprob;
   data posterior(keep=PetalWidth PetalLength species z pl pw _into_);
      merge plotp(in=d) iris(keep=PetalWidth PetalLength species
                             rename=(PetalWidth=pw PetalLength=pl));
      if d then z = max(setosa,versicolor,virginica);
   run;

   title3 'Plot of Posterior Probabilities ';

   proc sgrender data=posterior template=contour;
   run;
%mend;

%macro contclass;
   title3 'Plot of Classification Results';

   proc sgrender data=posterior(drop=z rename=(_into_=z)) template=contour;
   run;
%mend;
```

A normal-theory analysis (METHOD=NORMAL) assuming equal covariance matrices (POOL=YES) illustrates the linearity of the classification boundaries. These statements produce Output 31.2.2:

```
title2 'Using Normal Density Estimates with Equal Variance';

proc discrim data=iris method=normal pool=yes
             testdata=plotdata testout=plotp testoutd=plotd
             short noclassify crosslisterr;
   class Species;
   var Petal:;
run;

%contden;
%contprob;
%contclass;
```

Output 31.2.2 Normal Density Estimates with Equal Variance

Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Equal Variance

The DISCRIM Procedure

Total Sample Size	150	DF Total	149
Variables	2	DF Within Classes	147
Classes	3	DF Between Classes	2

Number of Observations Read	150
Number of Observations Used	150

Class Level Information

Species	Variable Name	Frequency	Weight	Proportion	Prior Probability
Setosa	Setosa	50	50.0000	0.333333	0.333333
Versicolor	Versicolor	50	50.0000	0.333333	0.333333
Virginica	Virginica	50	50.0000	0.333333	0.333333

Output 31.2.2 *continued*

Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Equal Variance

The DISCRIM Procedure
Classification Results for Calibration Data: WORK.IRIS
Cross-validation Results using Linear Discriminant Function

Posterior Probability of Membership in Species

Obs	From Species	Classified into Species	Setosa	Versicolor	Virginica
5	Virginica	Versicolor *	0.0000	0.8453	0.1547
9	Versicolor	Virginica *	0.0000	0.2130	0.7870
25	Virginica	Versicolor *	0.0000	0.8322	0.1678
57	Virginica	Versicolor *	0.0000	0.8057	0.1943
91	Virginica	Versicolor *	0.0000	0.8903	0.1097
148	Versicolor	Virginica *	0.0000	0.3118	0.6882

* Misclassified observation

Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Equal Variance

The DISCRIM Procedure
Classification Summary for Calibration Data: WORK.IRIS
Cross-validation Summary using Linear Discriminant Function

Number of Observations and Percent Classified into Species

From Species	Setosa	Versicolor	Virginica	Total
Setosa	50	0	0	50
	100.00	0.00	0.00	100.00
Versicolor	0	48	2	50
	0.00	96.00	4.00	100.00
Virginica	0	4	46	50
	0.00	8.00	92.00	100.00
Total	50	52	48	150
	33.33	34.67	32.00	100.00
Priors	0.33333	0.33333	0.33333	

Error Count Estimates for Species

	Setosa	Versicolor	Virginica	Total
Rate	0.0000	0.0400	0.0800	0.0400
Priors	0.3333	0.3333	0.3333	

Output 31.2.2 *continued*

Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Equal Variance

The DISCRIM Procedure
Classification Summary for Test Data: WORK.PLOTDATA
Classification Summary using Linear Discriminant Function

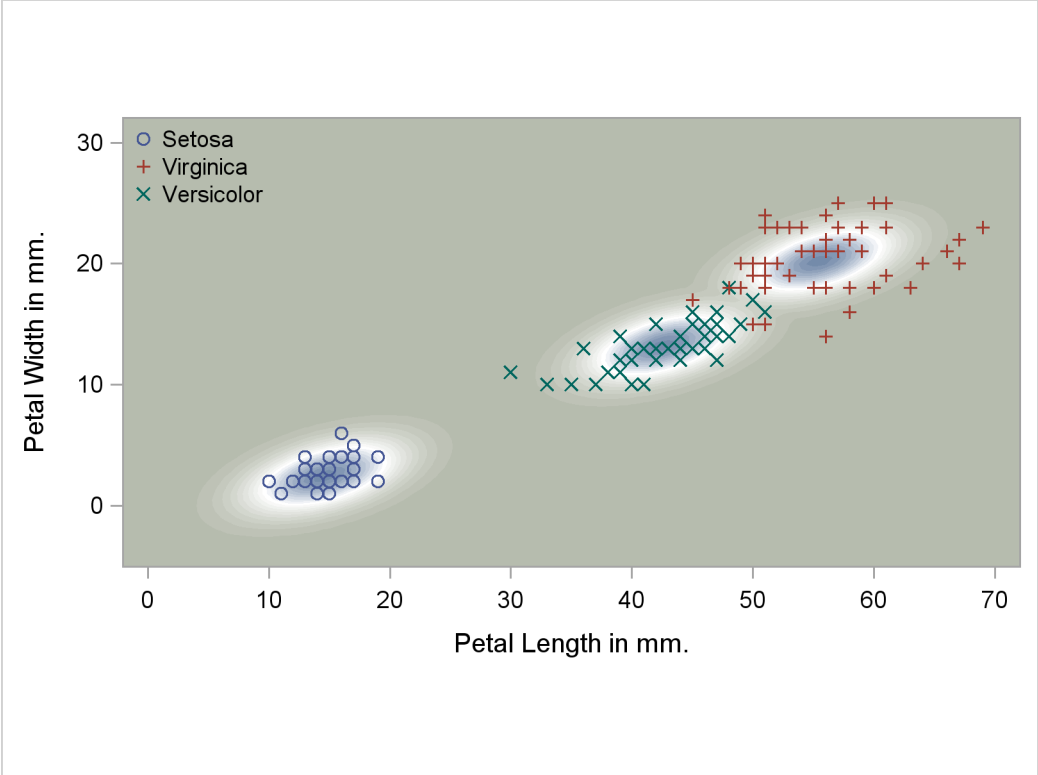
Observation Profile for Test Data

Number of Observations Read	11175
Number of Observations Used	11175

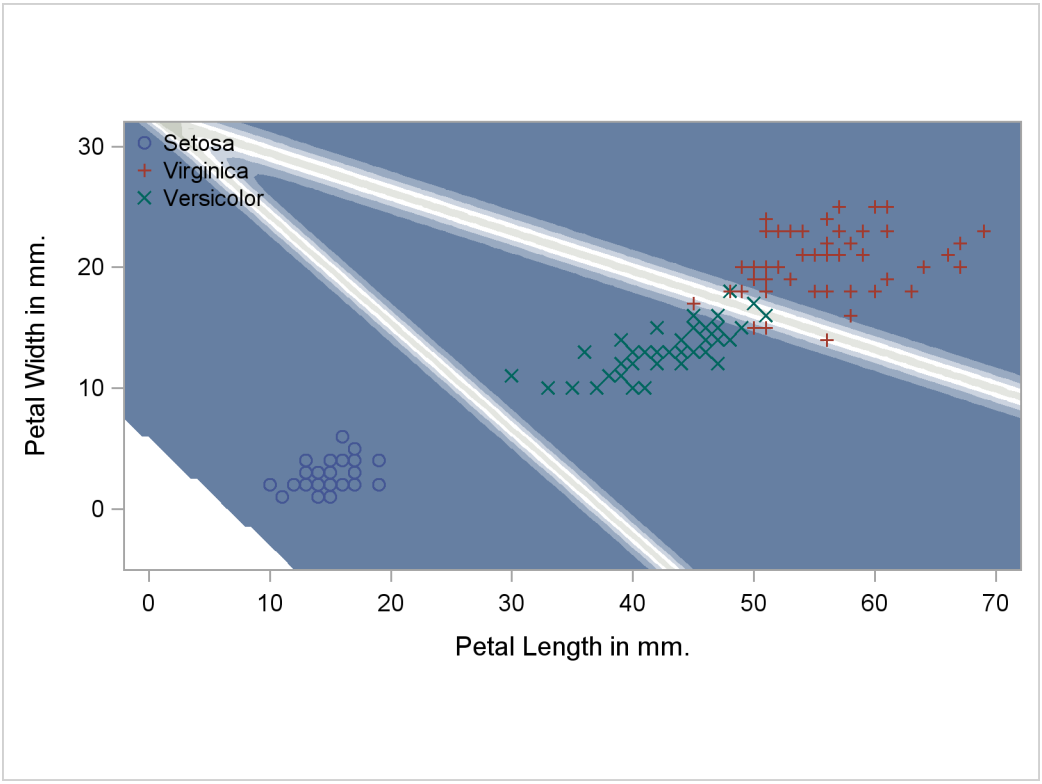
Number of Observations and Percent Classified into Species

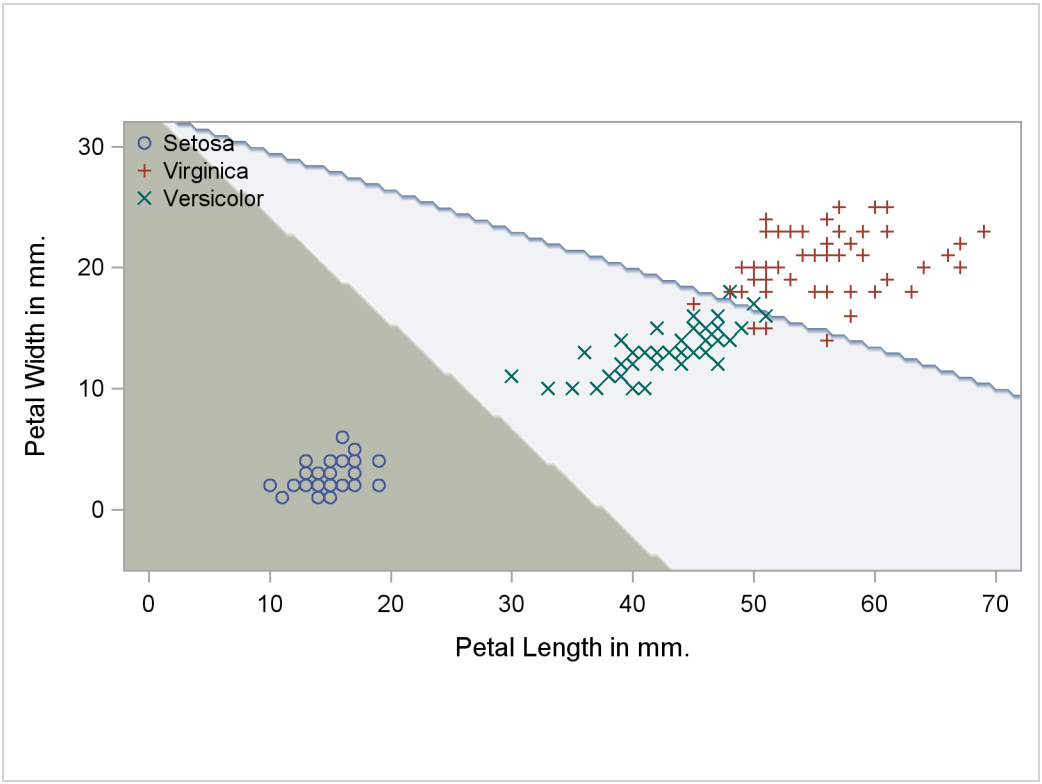
	Setosa	Versicolor	Virginica	Total
Total	3670	4243	3262	11175
	32.84	37.97	29.19	100.00
Priors	0.33333	0.33333	0.33333	

Output 31.2.2 *continued*



Output 31.2.2 *continued*





A normal-theory analysis assuming unequal covariance matrices (POOL=NO) illustrates quadratic classification boundaries. These statements produce Output 31.2.3:

```
title2 'Using Normal Density Estimates with Unequal Variance';

proc discrim data=iris method=normal pool=no
             testdata=plotdata testout=plotp testoutd=plotd
             short noclassify crosslisterr;
   class Species;
   var Petal:;
run;

%contden;
%contprob;
%contclass;
```

Output 31.2.3 Normal Density Estimates with Unequal Variance

Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Unequal Variance

The DISCRIM Procedure

Total Sample Size	150	DF Total	149
Variables	2	DF Within Classes	147
Classes	3	DF Between Classes	2

Number of Observations Read	150
Number of Observations Used	150

Class Level Information

Species	Variable Name	Frequency	Weight	Proportion	Prior Probability
Setosa	Setosa	50	50.0000	0.333333	0.333333
Versicolor	Versicolor	50	50.0000	0.333333	0.333333
Virginica	Virginica	50	50.0000	0.333333	0.333333

Output 31.2.3 *continued*

Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Unequal Variance

The DISCRIM Procedure
Classification Results for Calibration Data: WORK.IRIS
Cross-validation Results using Quadratic Discriminant Function

Posterior Probability of Membership in Species

Obs	From Species	Classified into Species	Setosa	Versicolor	Virginica
5	Virginica	Versicolor *	0.0000	0.7288	0.2712
9	Versicolor	Virginica *	0.0000	0.0903	0.9097
25	Virginica	Versicolor *	0.0000	0.5196	0.4804
91	Virginica	Versicolor *	0.0000	0.8335	0.1665
148	Versicolor	Virginica *	0.0000	0.4675	0.5325

* Misclassified observation

Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Unequal Variance

The DISCRIM Procedure
Classification Summary for Calibration Data: WORK.IRIS
Cross-validation Summary using Quadratic Discriminant Function

Number of Observations and Percent Classified into Species

From Species	Setosa	Versicolor	Virginica	Total
Setosa	50	0	0	50
	100.00	0.00	0.00	100.00
Versicolor	0	48	2	50
	0.00	96.00	4.00	100.00
Virginica	0	3	47	50
	0.00	6.00	94.00	100.00
Total	50	51	49	150
	33.33	34.00	32.67	100.00
Priors	0.33333	0.33333	0.33333	

Error Count Estimates for Species

	Setosa	Versicolor	Virginica	Total
Rate	0.0000	0.0400	0.0600	0.0333
Priors	0.3333	0.3333	0.3333	

Output 31.2.3 *continued*

Discriminant Analysis of Fisher (1936) Iris Data
Using Normal Density Estimates with Unequal Variance

The DISCRIM Procedure
Classification Summary for Test Data: WORK.PLOTDATA
Classification Summary using Quadratic Discriminant Function

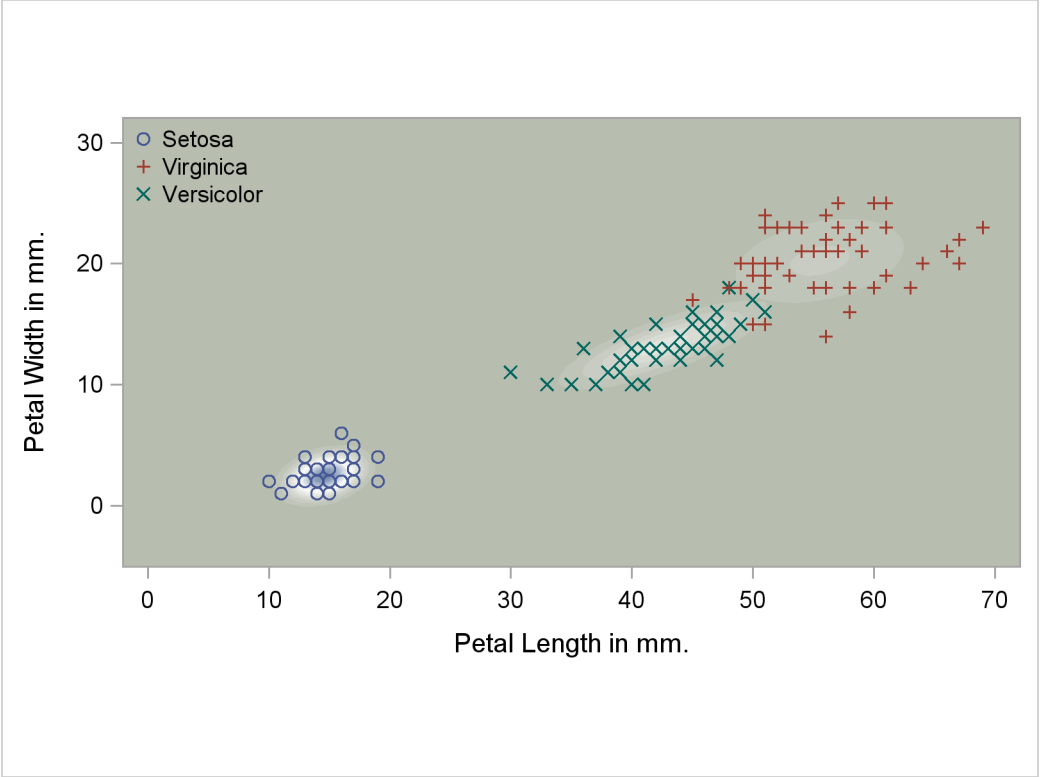
Observation Profile for Test Data

Number of Observations Read	11175
Number of Observations Used	11175

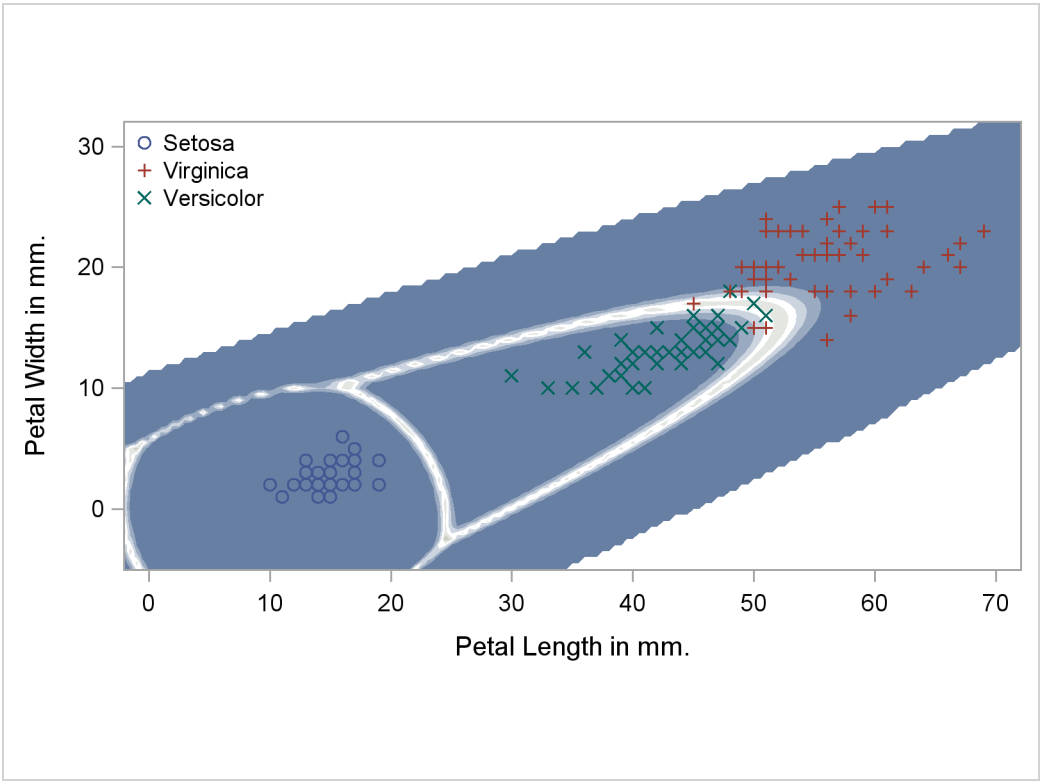
Number of Observations and Percent Classified into Species

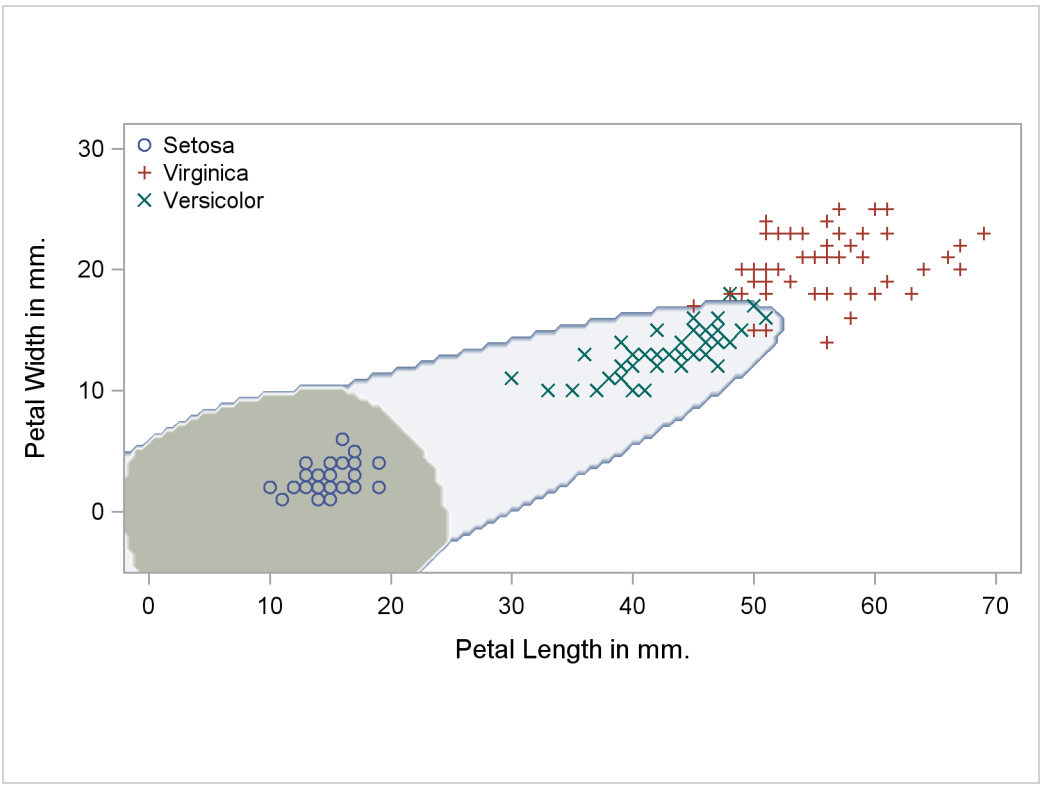
	Setosa	Versicolor	Virginica	Total
Total	1382	1345	8448	11175
	12.37	12.04	75.60	100.00
Priors	0.33333	0.33333	0.33333	

Output 31.2.3 *continued*



Output 31.2.3 *continued*





A nonparametric analysis (METHOD=NPAR) follows, using normal kernels (KERNEL=NORMAL) and equal bandwidths (POOL=YES) in each class. The value of the radius parameter r that, assuming normality, minimizes an approximate mean integrated square error is 0.50 (see the section "Nonparametric Methods" on page 1403). These statements produce Output 31.2.4:

```
title2 'Using Kernel Density Estimates with Equal Bandwidth';

proc discrim data=iris method=npar kernel=normal
             r=.5 pool=yes testoutd=plotd
             testdata=plotdata testout=plotp
             short noclassify crosslisterr;
   class Species;
   var Petal:;
run;

%contden;
%contprob;
%contclass;
```

Output 31.2.4 Kernel Density Estimates with Equal Bandwidth

Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Equal Bandwidth

The DISCRIM Procedure

Total Sample Size	150	DF Total	149
Variables	2	DF Within Classes	147
Classes	3	DF Between Classes	2

Number of Observations Read	150
Number of Observations Used	150

Class Level Information

Species	Variable Name	Frequency	Weight	Proportion	Prior Probability
Setosa	Setosa	50	50.0000	0.333333	0.333333
Versicolor	Versicolor	50	50.0000	0.333333	0.333333
Virginica	Virginica	50	50.0000	0.333333	0.333333

Output 31.2.4 *continued*

Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Equal Bandwidth

The DISCRIM Procedure
Classification Results for Calibration Data: WORK.IRIS
Cross-validation Results using Normal Kernel Density

Posterior Probability of Membership in Species

Obs	From Species	Classified into Species	Setosa	Versicolor	Virginica
5	Virginica	Versicolor *	0.0000	0.7474	0.2526
9	Versicolor	Virginica *	0.0000	0.0800	0.9200
25	Virginica	Versicolor *	0.0000	0.5863	0.4137
91	Virginica	Versicolor *	0.0000	0.8358	0.1642
148	Versicolor	Virginica *	0.0000	0.4123	0.5877

* Misclassified observation

Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Equal Bandwidth

The DISCRIM Procedure
Classification Summary for Calibration Data: WORK.IRIS
Cross-validation Summary using Normal Kernel Density

Number of Observations and Percent Classified into Species

From Species	Setosa	Versicolor	Virginica	Total
Setosa	50	0	0	50
	100.00	0.00	0.00	100.00
Versicolor	0	48	2	50
	0.00	96.00	4.00	100.00
Virginica	0	3	47	50
	0.00	6.00	94.00	100.00
Total	50	51	49	150
	33.33	34.00	32.67	100.00
Priors	0.33333	0.33333	0.33333	

Error Count Estimates for Species

	Setosa	Versicolor	Virginica	Total
Rate	0.0000	0.0400	0.0600	0.0333
Priors	0.3333	0.3333	0.3333	

Output 31.2.4 *continued*

Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Equal Bandwidth

The DISCRIM Procedure
Classification Summary for Test Data: WORK.PLOTDATA
Classification Summary using Normal Kernel Density

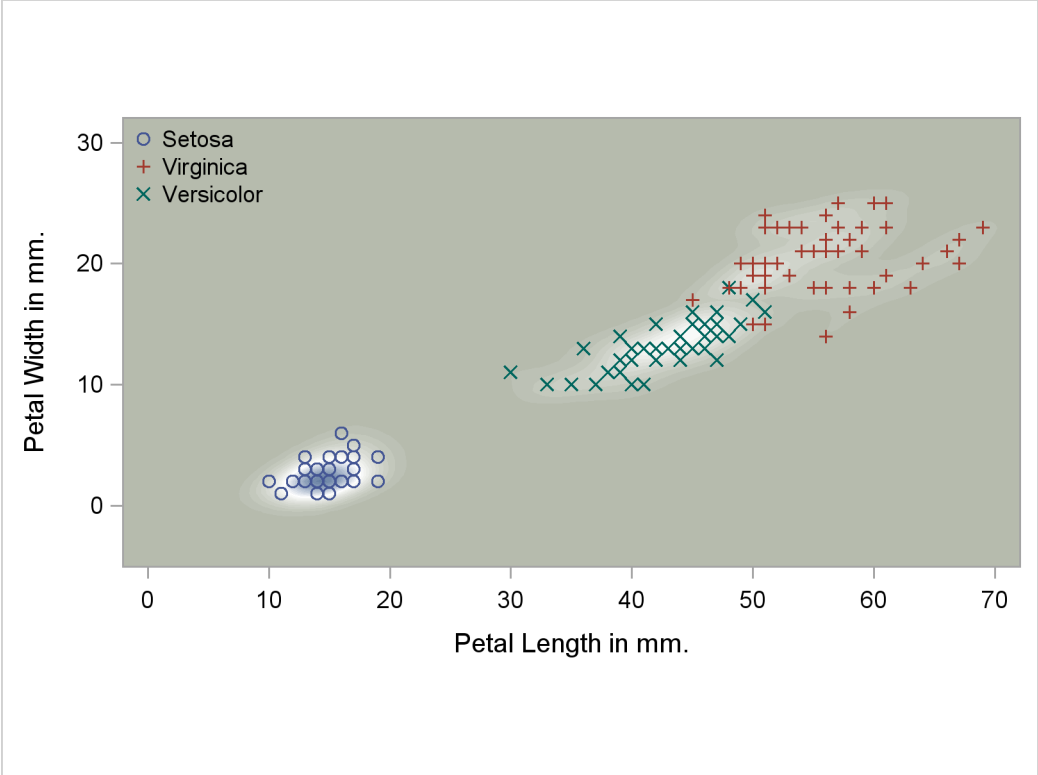
Observation Profile for Test Data

Number of Observations Read	11175
Number of Observations Used	11175

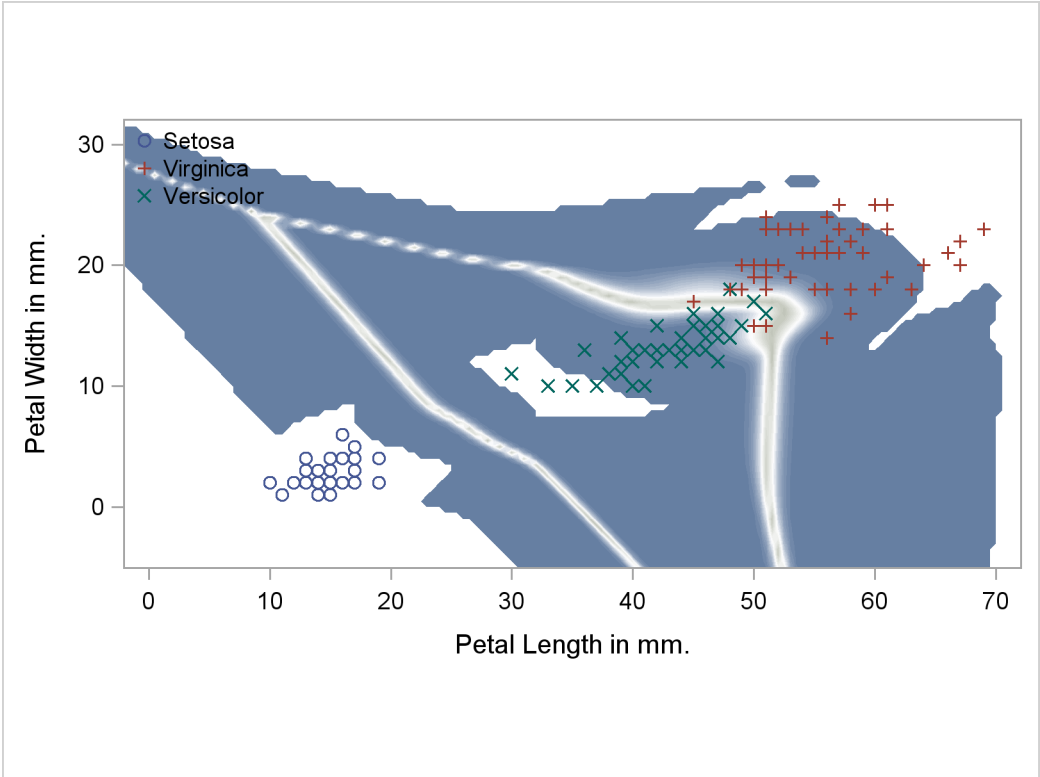
Number of Observations and Percent Classified into Species

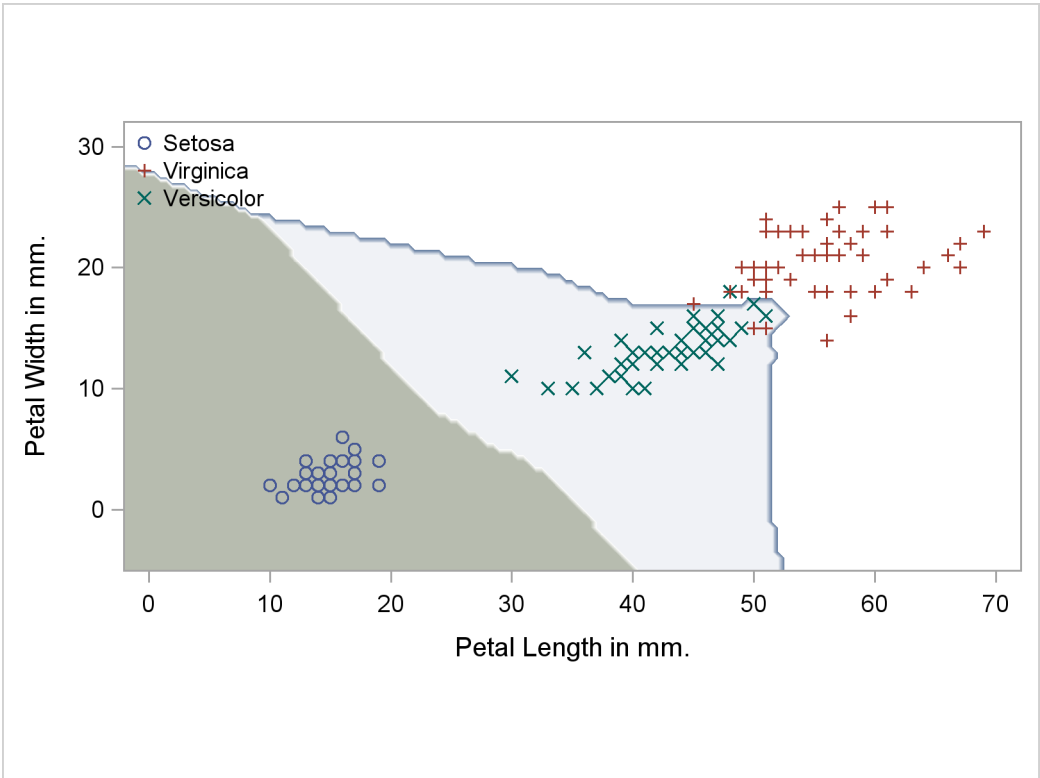
	Setosa	Versicolor	Virginica	Total
Total	3195	2492	5488	11175
	28.59	22.30	49.11	100.00
Priors	0.33333	0.33333	0.33333	

Output 31.2.4 *continued*



Output 31.2.4 *continued*





Another nonparametric analysis is run with unequal bandwidths (POOL=NO). These statements produce Output 31.2.5:

```
title2 'Using Kernel Density Estimates with Unequal Bandwidth';

proc discrim data=iris method=npar kernel=normal
             r=.5 pool=no testoutd=plotd
             testdata=plotdata testout=plotp
             short noclassify crosslisterr;
   class Species;
   var Petal:;
run;

%contden;
%contprob;
%contclass;
```

Output 31.2.5 Kernel Density Estimates with Unequal Bandwidth

```
              Discriminant Analysis of Fisher (1936) Iris Data
            Using Kernel Density Estimates with Unequal Bandwidth

                           The DISCRIM Procedure

        Total Sample Size      150          DF Total               149
        Variables                2          DF Within Classes      147
        Classes                  3          DF Between Classes       2

                  Number of Observations Read           150
                  Number of Observations Used           150

                           Class Level Information

              Variable                                              Prior
Species       Name          Frequency       Weight    Proportion    Probability

Setosa        Setosa               50      50.0000      0.333333       0.333333
Versicolor    Versicolor           50      50.0000      0.333333       0.333333
Virginica     Virginica            50      50.0000      0.333333       0.333333
```

Output 31.2.5 *continued*

Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Unequal Bandwidth

The DISCRIM Procedure
Classification Results for Calibration Data: WORK.IRIS
Cross-validation Results using Normal Kernel Density

Posterior Probability of Membership in Species

Obs	From Species	Classified into Species	Setosa	Versicolor	Virginica
5	Virginica	Versicolor *	0.0000	0.7826	0.2174
9	Versicolor	Virginica *	0.0000	0.0506	0.9494
91	Virginica	Versicolor *	0.0000	0.8802	0.1198
148	Versicolor	Virginica *	0.0000	0.3726	0.6274

* Misclassified observation

Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Unequal Bandwidth

The DISCRIM Procedure
Classification Summary for Calibration Data: WORK.IRIS
Cross-validation Summary using Normal Kernel Density

Number of Observations and Percent Classified into Species

From Species	Setosa	Versicolor	Virginica	Total
Setosa	50	0	0	50
	100.00	0.00	0.00	100.00
Versicolor	0	48	2	50
	0.00	96.00	4.00	100.00
Virginica	0	2	48	50
	0.00	4.00	96.00	100.00
Total	50	50	50	150
	33.33	33.33	33.33	100.00
Priors	0.33333	0.33333	0.33333	

Error Count Estimates for Species

	Setosa	Versicolor	Virginica	Total
Rate	0.0000	0.0400	0.0400	0.0267
Priors	0.3333	0.3333	0.3333	

Output 31.2.5 *continued*

Discriminant Analysis of Fisher (1936) Iris Data
Using Kernel Density Estimates with Unequal Bandwidth

The DISCRIM Procedure
Classification Summary for Test Data: WORK.PLOTDATA
Classification Summary using Normal Kernel Density

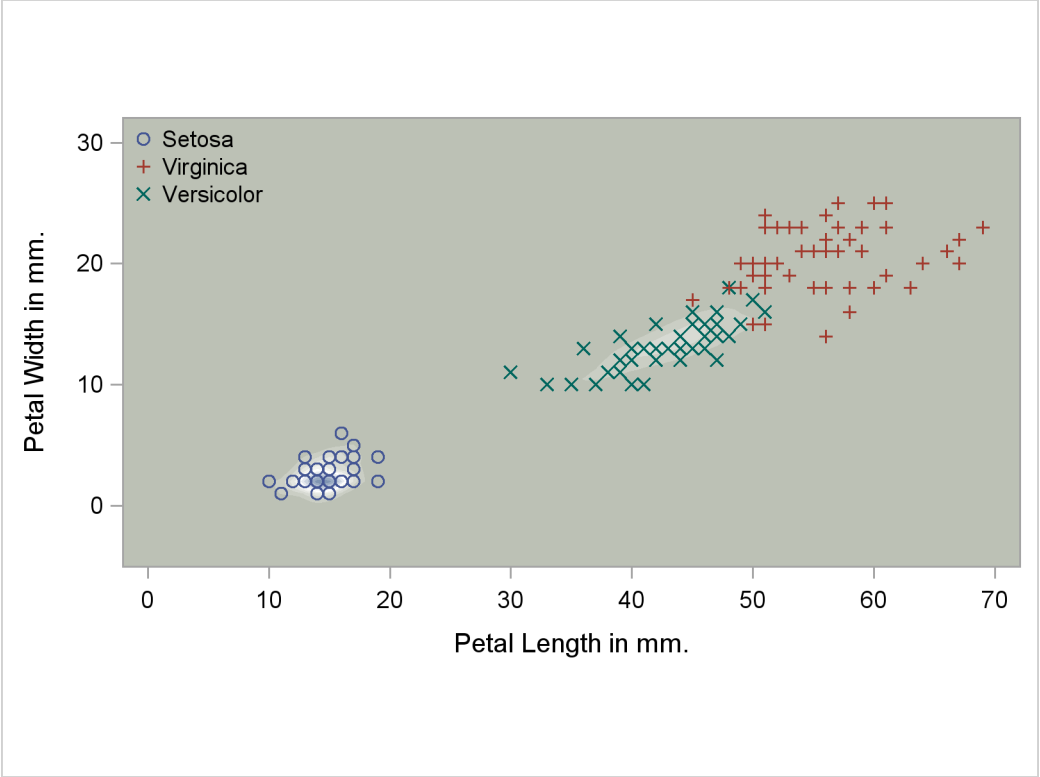
Observation Profile for Test Data

Number of Observations Read	11175
Number of Observations Used	11175

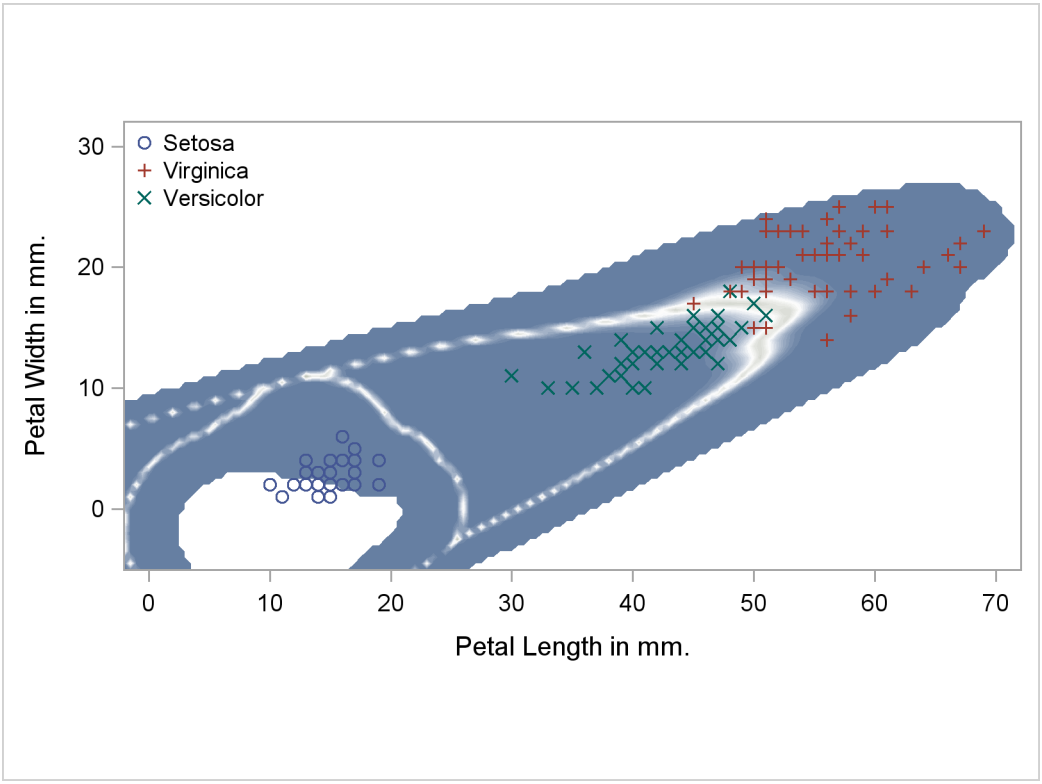
Number of Observations and Percent Classified into Species

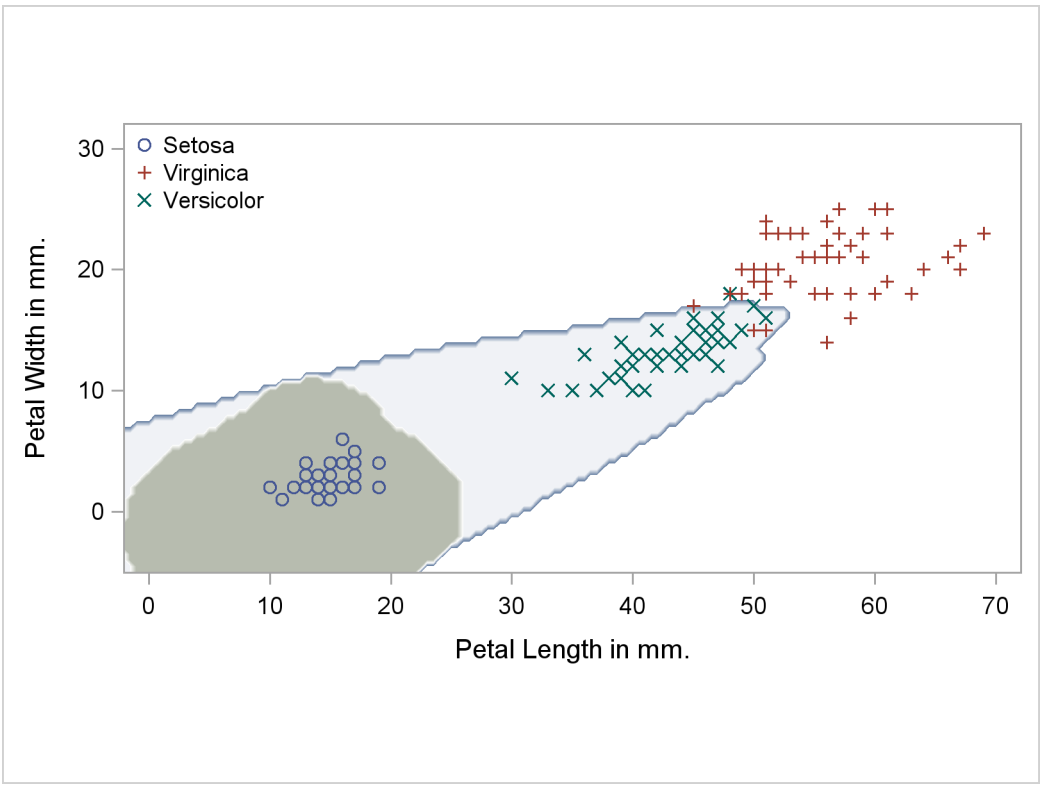
	Setosa	Versicolor	Virginica	Total
Total	1370	1505	8300	11175
	12.26	13.47	74.27	100.00
Priors	0.33333	0.33333	0.33333	

Output 31.2.5 *continued*



Output 31.2.5 *continued*





Example 31.3: Normal-Theory Discriminant Analysis of Iris Data

In this example, PROC DISCRIM uses normal-theory methods to classify the iris data used in Example 31.1. The POOL=TEST option tests the homogeneity of the within-group covariance matrices (Output 31.3.3). Since the resulting test statistic is significant at the 0.10 level, the within-group covariance matrices are used to derive the quadratic discriminant criterion. The WCOV and PCOV options display the within-group covariance matrices and the pooled covariance matrix (Output 31.3.2). The DISTANCE option displays squared distances between classes (Output 31.3.4). The ANOVA and MANOVA options test the hypothesis that the class means are equal, by using univariate statistics and multivariate statistics; all statistics are significant at the 0.0001 level (Output 31.3.5). The LISTERR option lists the misclassified observations under resubstitution (Output 31.3.6). The CROSSLISTERR option lists the observations that are misclassified under cross validation and displays cross validation error-rate estimates (Output 31.3.7). The resubstitution error count estimate, 0.02, is not larger than the cross validation error count estimate, 0.0267, as would be expected because the resubstitution estimate is optimistically biased. The OUTSTAT= option generates a TYPE=MIXED (because POOL=TEST) output data set containing various statistics such as means, covariances, and coefficients of the discriminant function (Output 31.3.8).

The following statements produce Output 31.3.1 through Output 31.3.8:

```
title 'Discriminant Analysis of Fisher (1936) Iris Data';
title2 'Using Quadratic Discriminant Function';

proc discrim data=iris outstat=irisstat
             wcov pcov method=normal pool=test
             distance anova manova listerr crosslisterr;
   class Species;
   var SepalLength SepalWidth PetalLength PetalWidth;
run;

proc print data=irisstat;
   title2 'Output Discriminant Statistics';
run;
```

Output 31.3.1 Quadratic Discriminant Analysis of Iris Data

```
        Discriminant Analysis of Fisher (1936) Iris Data
             Using Quadratic Discriminant Function

                    The DISCRIM Procedure

Total Sample Size      150          DF Total               149
Variables                4          DF Within Classes      147
Classes                  3          DF Between Classes       2

          Number of Observations Read           150
          Number of Observations Used           150
```

Output 31.3.1 *continued*

Class Level Information

Species	Variable Name	Frequency	Weight	Proportion	Prior Probability
Setosa	Setosa	50	50.0000	0.333333	0.333333
Versicolor	Versicolor	50	50.0000	0.333333	0.333333
Virginica	Virginica	50	50.0000	0.333333	0.333333

Output 31.3.2 Covariance Matrices

Discriminant Analysis of Fisher (1936) Iris Data
Using Quadratic Discriminant Function

The DISCRIM Procedure
Within-Class Covariance Matrices

Species = Setosa, DF = 49

Variable	Label	SepalLength	SepalWidth	PetalLength	PetalWidth
Sepal Length	Sepal Length in mm.	12.42489796	9.92163265	1.63551020	1.03306122
SepalWidth	Sepal Width in mm.	9.92163265	14.36897959	1.16979592	0.92979592
Petal Length	Petal Length in mm.	1.63551020	1.16979592	3.01591837	0.60693878
PetalWidth	Petal Width in mm.	1.03306122	0.92979592	0.60693878	1.11061224

Species = Versicolor, DF = 49

Variable	Label	SepalLength	SepalWidth	PetalLength	PetalWidth
Sepal Length	Sepal Length in mm.	26.64326531	8.51836735	18.28979592	5.57795918
SepalWidth	Sepal Width in mm.	8.51836735	9.84693878	8.26530612	4.12040816
Petal Length	Petal Length in mm.	18.28979592	8.26530612	22.08163265	7.31020408
PetalWidth	Petal Width in mm.	5.57795918	4.12040816	7.31020408	3.91061224

Output 31.3.2 *continued*

Species = Virginica, DF = 49

Variable	Label	SepalLength	SepalWidth	PetalLength	PetalWidth
Sepal Length	Sepal Length in mm.	40.43428571	9.37632653	30.32897959	4.90938776
SepalWidth	Sepal Width in mm.	9.37632653	10.40040816	7.13795918	4.76285714
Petal Length	Petal Length in mm.	30.32897959	7.13795918	30.45877551	4.88244898
PetalWidth	Petal Width in mm.	4.90938776	4.76285714	4.88244898	7.54326531

Discriminant Analysis of Fisher (1936) Iris Data
Using Quadratic Discriminant Function

The DISCRIM Procedure

Pooled Within-Class Covariance Matrix, DF = 147

Variable	Label	SepalLength	SepalWidth	PetalLength	PetalWidth
Sepal Length	Sepal Length in mm.	26.50081633	9.27210884	16.75142857	3.84013605
SepalWidth	Sepal Width in mm.	9.27210884	11.53877551	5.52435374	3.27102041
Petal Length	Petal Length in mm.	16.75142857	5.52435374	18.51877551	4.26653061
PetalWidth	Petal Width in mm.	3.84013605	3.27102041	4.26653061	4.18816327

Output 31.3.2 *continued*

Within Covariance Matrix Information

Species	Covariance Matrix Rank	Natural Log of the Determinant of the Covariance Matrix
Setosa	4	5.35332
Versicolor	4	7.54636
Virginica	4	9.49362
Pooled	4	8.46214

Output 31.3.3 Homogeneity Test

```
        Discriminant Analysis of Fisher (1936) Iris Data
             Using Quadratic Discriminant Function

                      The DISCRIM Procedure
        Test of Homogeneity of Within Covariance Matrices

                 Chi-Square        DF    Pr > ChiSq

                 140.943050        20        <.0001

      Since the Chi-Square value is significant at the 0.1 level, the within
      covariance matrices will be used in the discriminant function.
      Reference: Morrison, D.F. (1976) Multivariate Statistical
      Methods p252.
```

Output 31.3.4 Squared Distances

```
        Discriminant Analysis of Fisher (1936) Iris Data
             Using Quadratic Discriminant Function

                      The DISCRIM Procedure

                 Squared Distance to Species

      From
      Species           Setosa    Versicolor     Virginica

      Setosa                 0     103.19382     168.76759
      Versicolor     323.06203             0      13.83875
      Virginica      706.08494      17.86670             0

           Generalized Squared Distance to Species

      From
      Species           Setosa    Versicolor     Virginica

      Setosa           5.35332     110.74017     178.26121
      Versicolor     328.41535       7.54636      23.33238
      Virginica      711.43826      25.41306       9.49362
```

Output 31.3.5 Tests of Equal Class Means

Discriminant Analysis of Fisher (1936) Iris Data
Using Quadratic Discriminant Function

The DISCRIM Procedure

Univariate Test Statistics

F Statistics, Num DF=2, Den DF=147

Variable	Label	Total Standard Deviation	Pooled Standard Deviation	Between Standard Deviation	R-Square	R-Square / (1-RSq)	F Value	Pr > F
Sepal Length	Sepal Length in mm.	8.2807	5.1479	7.9506	0.6187	1.6226	119.26	<.0001
Sepal Width	Sepal Width in mm.	4.3587	3.3969	3.3682	0.4008	0.6688	49.16	<.0001
Petal Length	Petal Length in mm.	17.6530	4.3033	20.9070	0.9414	16.0566	1180.16	<.0001
Petal Width	Petal Width in mm.	7.6224	2.0465	8.9673	0.9289	13.0613	960.01	<.0001

Average R-Square

Unweighted	0.7224358
Weighted by Variance	0.8689444

Multivariate Statistics and F Approximations

S=2 M=0.5 N=71

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.02343863	199.15	8	288	<.0001
Pillai's Trace	1.19189883	53.47	8	290	<.0001
Hotelling-Lawley Trace	32.47732024	582.20	8	203.4	<.0001
Roy's Greatest Root	32.19192920	1166.96	4	145	<.0001

NOTE: F Statistic for Roy's Greatest Root is an upper bound.
NOTE: F Statistic for Wilks' Lambda is exact.

Output 31.3.6 Misclassified Observations: Resubstitution

Discriminant Analysis of Fisher (1936) Iris Data
Using Quadratic Discriminant Function

The DISCRIM Procedure
Classification Results for Calibration Data: WORK.IRIS
Resubstitution Results using Quadratic Discriminant Function

Posterior Probability of Membership in Species

Obs	From Species	Classified into Species	Setosa	Versicolor	Virginica
5	Virginica	Versicolor *	0.0000	0.6050	0.3950
9	Versicolor	Virginica *	0.0000	0.3359	0.6641
12	Versicolor	Virginica *	0.0000	0.1543	0.8457

* Misclassified observation

Discriminant Analysis of Fisher (1936) Iris Data
Using Quadratic Discriminant Function

The DISCRIM Procedure
Classification Summary for Calibration Data: WORK.IRIS
Resubstitution Summary using Quadratic Discriminant Function

Number of Observations and Percent Classified into Species

From Species	Setosa	Versicolor	Virginica	Total
Setosa	50	0	0	50
	100.00	0.00	0.00	100.00
Versicolor	0	48	2	50
	0.00	96.00	4.00	100.00
Virginica	0	1	49	50
	0.00	2.00	98.00	100.00
Total	50	49	51	150
	33.33	32.67	34.00	100.00
Priors	0.33333	0.33333	0.33333	

Error Count Estimates for Species

	Setosa	Versicolor	Virginica	Total
Rate	0.0000	0.0400	0.0200	0.0200
Priors	0.3333	0.3333	0.3333	

Output 31.3.7 Misclassified Observations: Cross Validation

Discriminant Analysis of Fisher (1936) Iris Data
Using Quadratic Discriminant Function

The DISCRIM Procedure
Classification Results for Calibration Data: WORK.IRIS
Cross-validation Results using Quadratic Discriminant Function

Posterior Probability of Membership in Species

Obs	From Species	Classified into Species	Setosa	Versicolor	Virginica
5	Virginica	Versicolor *	0.0000	0.6632	0.3368
8	Versicolor	Virginica *	0.0000	0.3134	0.6866
9	Versicolor	Virginica *	0.0000	0.1616	0.8384
12	Versicolor	Virginica *	0.0000	0.0713	0.9287

* Misclassified observation

Discriminant Analysis of Fisher (1936) Iris Data
Using Quadratic Discriminant Function

The DISCRIM Procedure
Classification Summary for Calibration Data: WORK.IRIS
Cross-validation Summary using Quadratic Discriminant Function

Number of Observations and Percent Classified into Species

From Species	Setosa	Versicolor	Virginica	Total
Setosa	50	0	0	50
	100.00	0.00	0.00	100.00
Versicolor	0	47	3	50
	0.00	94.00	6.00	100.00
Virginica	0	1	49	50
	0.00	2.00	98.00	100.00
Total	50	48	52	150
	33.33	32.00	34.67	100.00
Priors	0.33333	0.33333	0.33333	

Error Count Estimates for Species

	Setosa	Versicolor	Virginica	Total
Rate	0.0000	0.0600	0.0200	0.0267
Priors	0.3333	0.3333	0.3333	

Output 31.3.8 Output Statistics from Iris Data

Discriminant Analysis of Fisher (1936) Iris Data
Output Discriminant Statistics

Obs	Species	_TYPE_	_NAME_	Sepal Length	Sepal Width	Petal Length	Petal Width
1	.	N		150.00	150.00	150.00	150.00
2	Setosa	N		50.00	50.00	50.00	50.00
3	Versicolor	N		50.00	50.00	50.00	50.00
4	Virginica	N		50.00	50.00	50.00	50.00
5	.	MEAN		58.43	30.57	37.58	11.99
6	Setosa	MEAN		50.06	34.28	14.62	2.46
7	Versicolor	MEAN		59.36	27.70	42.60	13.26
8	Virginica	MEAN		65.88	29.74	55.52	20.26
9	Setosa	PRIOR		0.33	0.33	0.33	0.33
10	Versicolor	PRIOR		0.33	0.33	0.33	0.33
11	Virginica	PRIOR		0.33	0.33	0.33	0.33
12	Setosa	CSSCP	SepalLength	608.82	486.16	80.14	50.62
13	Setosa	CSSCP	SepalWidth	486.16	704.08	57.32	45.56
14	Setosa	CSSCP	PetalLength	80.14	57.32	147.78	29.74
15	Setosa	CSSCP	PetalWidth	50.62	45.56	29.74	54.42
16	Versicolor	CSSCP	SepalLength	1305.52	417.40	896.20	273.32
17	Versicolor	CSSCP	SepalWidth	417.40	482.50	405.00	201.90
18	Versicolor	CSSCP	PetalLength	896.20	405.00	1082.00	358.20
19	Versicolor	CSSCP	PetalWidth	273.32	201.90	358.20	191.62
20	Virginica	CSSCP	SepalLength	1981.28	459.44	1486.12	240.56
21	Virginica	CSSCP	SepalWidth	459.44	509.62	349.76	233.38
22	Virginica	CSSCP	PetalLength	1486.12	349.76	1492.48	239.24
23	Virginica	CSSCP	PetalWidth	240.56	233.38	239.24	369.62
24	.	PSSCP	SepalLength	3895.62	1363.00	2462.46	564.50
25	.	PSSCP	SepalWidth	1363.00	1696.20	812.08	480.84
26	.	PSSCP	PetalLength	2462.46	812.08	2722.26	627.18
27	.	PSSCP	PetalWidth	564.50	480.84	627.18	615.66
28	.	BSSCP	SepalLength	6321.21	-1995.27	16524.84	7127.93
29	.	BSSCP	SepalWidth	-1995.27	1134.49	-5723.96	-2293.27
30	.	BSSCP	PetalLength	16524.84	-5723.96	43710.28	18677.40
31	.	BSSCP	PetalWidth	7127.93	-2293.27	18677.40	8041.33
32	.	CSSCP	SepalLength	10216.83	-632.27	18987.30	7692.43
33	.	CSSCP	SepalWidth	-632.27	2830.69	-4911.88	-1812.43
34	.	CSSCP	PetalLength	18987.30	-4911.88	46432.54	19304.58
35	.	CSSCP	PetalWidth	7692.43	-1812.43	19304.58	8656.99
36	.	RSQUARED		0.62	0.40	0.94	0.93
37	Setosa	COV	SepalLength	12.42	9.92	1.64	1.03
38	Setosa	COV	SepalWidth	9.92	14.37	1.17	0.93
39	Setosa	COV	PetalLength	1.64	1.17	3.02	0.61
40	Setosa	COV	PetalWidth	1.03	0.93	0.61	1.11
41	Versicolor	COV	SepalLength	26.64	8.52	18.29	5.58
42	Versicolor	COV	SepalWidth	8.52	9.85	8.27	4.12
43	Versicolor	COV	PetalLength	18.29	8.27	22.08	7.31
44	Versicolor	COV	PetalWidth	5.58	4.12	7.31	3.91
45	Virginica	COV	SepalLength	40.43	9.38	30.33	4.91
46	Virginica	COV	SepalWidth	9.38	10.40	7.14	4.76
47	Virginica	COV	PetalLength	30.33	7.14	30.46	4.88
48	Virginica	COV	PetalWidth	4.91	4.76	4.88	7.54

Output 31.3.8 *continued*

Discriminant Analysis of Fisher (1936) Iris Data
Output Discriminant Statistics

Obs	Species	_TYPE_	_NAME_	Sepal Length	Sepal Width	Petal Length	Petal Width
49	.	PCOV	SepalLength	26.501	9.2721	16.751	3.840
50	.	PCOV	SepalWidth	9.272	11.5388	5.524	3.271
51	.	PCOV	PetalLength	16.751	5.5244	18.519	4.267
52	.	PCOV	PetalWidth	3.840	3.2710	4.267	4.188
53	.	BCOV	SepalLength	63.212	-19.9527	165.248	71.279
54	.	BCOV	SepalWidth	-19.953	11.3449	-57.240	-22.933
55	.	BCOV	PetalLength	165.248	-57.2396	437.103	186.774
56	.	BCOV	PetalWidth	71.279	-22.9327	186.774	80.413
57	.	COV	SepalLength	68.569	-4.2434	127.432	51.627
58	.	COV	SepalWidth	-4.243	18.9979	-32.966	-12.164
59	.	COV	PetalLength	127.432	-32.9656	311.628	129.561
60	.	COV	PetalWidth	51.627	-12.1639	129.561	58.101
61	Setosa	STD		3.525	3.7906	1.737	1.054
62	Versicolor	STD		5.162	3.1380	4.699	1.978
63	Virginica	STD		6.359	3.2250	5.519	2.747
64	.	PSTD		5.148	3.3969	4.303	2.047
65	.	BSTD		7.951	3.3682	20.907	8.967
66	.	STD		8.281	4.3587	17.653	7.622
67	Setosa	CORR	SepalLength	1.000	0.7425	0.267	0.278
68	Setosa	CORR	SepalWidth	0.743	1.0000	0.178	0.233
69	Setosa	CORR	PetalLength	0.267	0.1777	1.000	0.332
70	Setosa	CORR	PetalWidth	0.278	0.2328	0.332	1.000
71	Versicolor	CORR	SepalLength	1.000	0.5259	0.754	0.546
72	Versicolor	CORR	SepalWidth	0.526	1.0000	0.561	0.664
73	Versicolor	CORR	PetalLength	0.754	0.5605	1.000	0.787
74	Versicolor	CORR	PetalWidth	0.546	0.6640	0.787	1.000
75	Virginica	CORR	SepalLength	1.000	0.4572	0.864	0.281
76	Virginica	CORR	SepalWidth	0.457	1.0000	0.401	0.538
77	Virginica	CORR	PetalLength	0.864	0.4010	1.000	0.322
78	Virginica	CORR	PetalWidth	0.281	0.5377	0.322	1.000
79	.	PCORR	SepalLength	1.000	0.5302	0.756	0.365
80	.	PCORR	SepalWidth	0.530	1.0000	0.378	0.471
81	.	PCORR	PetalLength	0.756	0.3779	1.000	0.484
82	.	PCORR	PetalWidth	0.365	0.4705	0.484	1.000
83	.	BCORR	SepalLength	1.000	-0.7451	0.994	1.000
84	.	BCORR	SepalWidth	-0.745	1.0000	-0.813	-0.759
85	.	BCORR	PetalLength	0.994	-0.8128	1.000	0.996
86	.	BCORR	PetalWidth	1.000	-0.7593	0.996	1.000
87	.	CORR	SepalLength	1.000	-0.1176	0.872	0.818
88	.	CORR	SepalWidth	-0.118	1.0000	-0.428	-0.366
89	.	CORR	PetalLength	0.872	-0.4284	1.000	0.963
90	.	CORR	PetalWidth	0.818	-0.3661	0.963	1.000
91	Setosa	STDMEAN		-1.011	0.8504	-1.301	-1.251
92	Versicolor	STDMEAN		0.112	-0.6592	0.284	0.166
93	Virginica	STDMEAN		0.899	-0.1912	1.016	1.085
94	Setosa	PSTDMEAN		-1.627	1.0912	-5.335	-4.658
95	Versicolor	PSTDMEAN		0.180	-0.8459	1.167	0.619
96	Virginica	PSTDMEAN		1.447	-0.2453	4.169	4.039

Output 31.3.8 *continued*

Discriminant Analysis of Fisher (1936) Iris Data
Output Discriminant Statistics

Obs	Species	_TYPE_	_NAME_	Sepal Length	Sepal Width	Petal Length	Petal Width
97	.	LNDETERM		8.462	8.462	8.462	8.462
98	Setosa	LNDETERM		5.353	5.353	5.353	5.353
99	Versicolor	LNDETERM		7.546	7.546	7.546	7.546
100	Virginica	LNDETERM		9.494	9.494	9.494	9.494
101	Setosa	QUAD	SepalLength	-0.095	0.062	0.023	0.024
102	Setosa	QUAD	SepalWidth	0.062	-0.078	-0.006	0.011
103	Setosa	QUAD	PetalLength	0.023	-0.006	-0.194	0.090
104	Setosa	QUAD	PetalWidth	0.024	0.011	0.090	-0.530
105	Setosa	QUAD	_LINEAR_	4.455	-0.762	3.356	-3.126
106	Setosa	QUAD	_CONST_	-121.826	-121.826	-121.826	-121.826
107	Versicolor	QUAD	SepalLength	-0.048	0.018	0.043	-0.032
108	Versicolor	QUAD	SepalWidth	0.018	-0.099	-0.011	0.097
109	Versicolor	QUAD	PetalLength	0.043	-0.011	-0.099	0.135
110	Versicolor	QUAD	PetalWidth	-0.032	0.097	0.135	-0.436
111	Versicolor	QUAD	_LINEAR_	1.801	1.596	0.327	-1.471
112	Versicolor	QUAD	_CONST_	-76.549	-76.549	-76.549	-76.549
113	Virginica	QUAD	SepalLength	-0.053	0.017	0.050	-0.009
114	Virginica	QUAD	SepalWidth	0.017	-0.079	-0.006	0.042
115	Virginica	QUAD	PetalLength	0.050	-0.006	-0.067	0.014
116	Virginica	QUAD	PetalWidth	-0.009	0.042	0.014	-0.097
117	Virginica	QUAD	_LINEAR_	0.737	1.325	0.623	0.966
118	Virginica	QUAD	_CONST_	-75.821	-75.821	-75.821	-75.821

Example 31.4: Linear Discriminant Analysis of Remote-Sensing Data on Crops

In this example, the remote-sensing data are used. In this data set, the observations are grouped into five crops: clover, corn, cotton, soybeans, and sugar beets. Four measures called x1 through x4 make up the descriptive variables.

In the first PROC DISCRIM statement, the DISCRIM procedure uses normal-theory methods (METHOD=NORMAL) assuming equal variances (POOL=YES) in five crops. The PRIORS statement, PRIORS PROP, sets the prior probabilities proportional to the sample sizes. The LIST option lists the resubstitution classification results for each observation (Output 31.4.2). The CROSSVALIDATE option displays cross validation error-rate estimates (Output 31.4.3). The OUTSTAT= option stores the calibration information in a new data set to classify future observations. A second PROC DISCRIM statement uses this calibration information to classify a test data set. Note that the values of the identification variable, xvalues, are obtained by rereading the x1 through x4 fields in the data lines as a single character variable. The following statements produce Output 31.4.1 through Output 31.4.3:

```
title 'Discriminant Analysis of Remote Sensing Data on Five Crops';

data crops;
   input Crop $ 1-10 x1-x4 xvalues $ 11-21;
   datalines;
Corn      16 27 31 33
Corn      15 23 30 30
Corn      16 27 27 26
Corn      18 20 25 23
Corn      15 15 31 32
Corn      15 32 32 15
Corn      12 15 16 73
Soybeans  20 23 23 25
Soybeans  24 24 25 32
Soybeans  21 25 23 24
Soybeans  27 45 24 12
Soybeans  12 13 15 42
Soybeans  22 32 31 43
Cotton    31 32 33 34
Cotton    29 24 26 28
Cotton    34 32 28 45
Cotton    26 25 23 24
Cotton    53 48 75 26
Cotton    34 35 25 78
Sugarbeets22 23 25 42
Sugarbeets25 25 24 26
Sugarbeets34 25 16 52
Sugarbeets54 23 21 54
Sugarbeets25 43 32 15
Sugarbeets26 54  2 54
Clover    12 45 32 54
Clover    24 58 25 34
Clover    87 54 61 21
Clover    51 31 31 16
Clover    96 48 54 62
Clover    31 31 11 11
Clover    56 13 13 71
Clover    32 13 27 32
Clover    36 26 54 32
Clover    53 08 06 54
Clover    32 32 62 16
;

title2 'Using the Linear Discriminant Function';

proc discrim data=crops outstat=cropstat method=normal pool=yes
             list crossvalidate;
   class Crop;
   priors prop;
   id xvalues;
   var x1-x4;
run;
```

Output 31.4.1 Linear Discriminant Function on Crop Data

Discriminant Analysis of Remote Sensing Data on Five Crops
Using the Linear Discriminant Function

The DISCRIM Procedure

Total Sample Size	36	DF Total	35
Variables	4	DF Within Classes	31
Classes	5	DF Between Classes	4

Number of Observations Read	36
Number of Observations Used	36

Class Level Information

Crop	Variable Name	Frequency	Weight	Proportion	Prior Probability
Clover	Clover	11	11.0000	0.305556	0.305556
Corn	Corn	7	7.0000	0.194444	0.194444
Cotton	Cotton	6	6.0000	0.166667	0.166667
Soybeans	Soybeans	6	6.0000	0.166667	0.166667
Sugarbeets	Sugarbeets	6	6.0000	0.166667	0.166667

Pooled Covariance Matrix Information

Covariance Matrix Rank	Natural Log of the Determinant of the Covariance Matrix
4	21.30189

Discriminant Analysis of Remote Sensing Data on Five Crops
Using the Linear Discriminant Function

The DISCRIM Procedure

Generalized Squared Distance to Crop

From Crop	Clover	Corn	Cotton	Soybeans	Sugarbeets
Clover	2.37125	7.52830	4.44969	6.16665	5.07262
Corn	6.62433	3.27522	5.46798	4.31383	6.47395
Cotton	3.23741	5.15968	3.58352	5.01819	4.87908
Soybeans	4.95438	4.00552	5.01819	3.58352	4.65998
Sugarbeets	3.86034	6.16564	4.87908	4.65998	3.58352

Output 31.4.1 *continued*

Linear Discriminant Function for Crop

Variable	Clover	Corn	Cotton	Soybeans	Sugarbeets
Constant	-10.98457	-7.72070	-11.46537	-7.28260	-9.80179
x1	0.08907	-0.04180	0.02462	0.0000369	0.04245
x2	0.17379	0.11970	0.17596	0.15896	0.20988
x3	0.11899	0.16511	0.15880	0.10622	0.06540
x4	0.15637	0.16768	0.18362	0.14133	0.16408

Output 31.4.2 Misclassified Observations: Resubstitution

Discriminant Analysis of Remote Sensing Data on Five Crops
Using the Linear Discriminant Function

The DISCRIM Procedure
Classification Results for Calibration Data: WORK.CROPS
Resubstitution Results using Linear Discriminant Function

Posterior Probability of Membership in Crop

xvalues	From Crop	Classified into Crop		Clover	Corn	Cotton	Soybeans	Sugarbeets
16 27 31 33	Corn	Corn		0.0894	0.4054	0.1763	0.2392	0.0897
15 23 30 30	Corn	Corn		0.0769	0.4558	0.1421	0.2530	0.0722
16 27 27 26	Corn	Corn		0.0982	0.3422	0.1365	0.3073	0.1157
18 20 25 23	Corn	Corn		0.1052	0.3634	0.1078	0.3281	0.0955
15 15 31 32	Corn	Corn		0.0588	0.5754	0.1173	0.2087	0.0398
15 32 32 15	Corn	Soybeans	*	0.0972	0.3278	0.1318	0.3420	0.1011
12 15 16 73	Corn	Corn		0.0454	0.5238	0.1849	0.1376	0.1083
20 23 23 25	Soybeans	Soybeans		0.1330	0.2804	0.1176	0.3305	0.1385
24 24 25 32	Soybeans	Soybeans		0.1768	0.2483	0.1586	0.2660	0.1502
21 25 23 24	Soybeans	Soybeans		0.1481	0.2431	0.1200	0.3318	0.1570
27 45 24 12	Soybeans	Sugarbeets	*	0.2357	0.0547	0.1016	0.2721	0.3359
12 13 15 42	Soybeans	Corn	*	0.0549	0.4749	0.0920	0.2768	0.1013
22 32 31 43	Soybeans	Cotton	*	0.1474	0.2606	0.2624	0.1848	0.1448
31 32 33 34	Cotton	Clover	*	0.2815	0.1518	0.2377	0.1767	0.1523
29 24 26 28	Cotton	Soybeans	*	0.2521	0.1842	0.1529	0.2549	0.1559
34 32 28 45	Cotton	Clover	*	0.3125	0.1023	0.2404	0.1357	0.2091
26 25 23 24	Cotton	Soybeans	*	0.2121	0.1809	0.1245	0.3045	0.1780
53 48 75 26	Cotton	Clover	*	0.4837	0.0391	0.4384	0.0223	0.0166
34 35 25 78	Cotton	Cotton		0.2256	0.0794	0.3810	0.0592	0.2548
22 23 25 42	Sugarbeets	Corn	*	0.1421	0.3066	0.1901	0.2231	0.1381
25 25 24 26	Sugarbeets	Soybeans	*	0.1969	0.2050	0.1354	0.2960	0.1667
34 25 16 52	Sugarbeets	Sugarbeets		0.2928	0.0871	0.1665	0.1479	0.3056
54 23 21 54	Sugarbeets	Clover	*	0.6215	0.0194	0.1250	0.0496	0.1845
25 43 32 15	Sugarbeets	Soybeans	*	0.2258	0.1135	0.1646	0.2770	0.2191
26 54 2 54	Sugarbeets	Sugarbeets		0.0850	0.0081	0.0521	0.0661	0.7887
12 45 32 54	Clover	Cotton	*	0.0693	0.2663	0.3394	0.1460	0.1789
24 58 25 34	Clover	Sugarbeets	*	0.1647	0.0376	0.1680	0.1452	0.4845
87 54 61 21	Clover	Clover		0.9328	0.0003	0.0478	0.0025	0.0165
51 31 31 16	Clover	Clover		0.6642	0.0205	0.0872	0.0959	0.1322
96 48 54 62	Clover	Clover		0.9215	0.0002	0.0604	0.0007	0.0173
31 31 11 11	Clover	Sugarbeets	*	0.2525	0.0402	0.0473	0.3012	0.3588
56 13 13 71	Clover	Clover		0.6132	0.0212	0.1226	0.0408	0.2023
32 13 27 32	Clover	Clover		0.2669	0.2616	0.1512	0.2260	0.0943
36 26 54 32	Clover	Cotton	*	0.2650	0.2645	0.3495	0.0918	0.0292
53 08 06 54	Clover	Clover		0.5914	0.0237	0.0676	0.0781	0.2392
32 32 62 16	Clover	Cotton	*	0.2163	0.3180	0.3327	0.1125	0.0206

* Misclassified observation

Output 31.4.2 *continued*

Discriminant Analysis of Remote Sensing Data on Five Crops
Using the Linear Discriminant Function

The DISCRIM Procedure
Classification Summary for Calibration Data: WORK.CROPS
Resubstitution Summary using Linear Discriminant Function

Number of Observations and Percent Classified into Crop

From Crop	Clover	Corn	Cotton	Soybeans	Sugarbeets	Total
Clover	6	0	3	0	2	11
	54.55	0.00	27.27	0.00	18.18	100.00
Corn	0	6	0	1	0	7
	0.00	85.71	0.00	14.29	0.00	100.00
Cotton	3	0	1	2	0	6
	50.00	0.00	16.67	33.33	0.00	100.00
Soybeans	0	1	1	3	1	6
	0.00	16.67	16.67	50.00	16.67	100.00
Sugarbeets	1	1	0	2	2	6
	16.67	16.67	0.00	33.33	33.33	100.00
Total	10	8	5	8	5	36
	27.78	22.22	13.89	22.22	13.89	100.00
Priors	0.30556	0.19444	0.16667	0.16667	0.16667	

Error Count Estimates for Crop

	Clover	Corn	Cotton	Soybeans	Sugarbeets	Total
Rate	0.4545	0.1429	0.8333	0.5000	0.6667	0.5000
Priors	0.3056	0.1944	0.1667	0.1667	0.1667	

Output 31.4.3 Misclassified Observations: Cross Validation

Discriminant Analysis of Remote Sensing Data on Five Crops
Using the Linear Discriminant Function

The DISCRIM Procedure
Classification Summary for Calibration Data: WORK.CROPS
Cross-validation Summary using Linear Discriminant Function

Number of Observations and Percent Classified into Crop

From Crop	Clover	Corn	Cotton	Soybeans	Sugarbeets	Total
Clover	4	3	1	0	3	11
	36.36	27.27	9.09	0.00	27.27	100.00
Corn	0	4	1	2	0	7
	0.00	57.14	14.29	28.57	0.00	100.00
Cotton	3	0	0	2	1	6
	50.00	0.00	0.00	33.33	16.67	100.00
Soybeans	0	1	1	3	1	6
	0.00	16.67	16.67	50.00	16.67	100.00
Sugarbeets	2	1	0	2	1	6
	33.33	16.67	0.00	33.33	16.67	100.00
Total	9	9	3	9	6	36
	25.00	25.00	8.33	25.00	16.67	100.00
Priors	0.30556	0.19444	0.16667	0.16667	0.16667	

Error Count Estimates for Crop

	Clover	Corn	Cotton	Soybeans	Sugarbeets	Total
Rate	0.6364	0.4286	1.0000	0.5000	0.8333	0.6667
Priors	0.3056	0.1944	0.1667	0.1667	0.1667	

Next, you can use the calibration information stored in the Cropstat data set to classify a test data set. The TESTLIST option lists the classification results for each observation in the test data set. The following statements produce Output 31.4.4 and Output 31.4.5:

```
data test;
   input Crop $ 1-10 x1-x4 xvalues $ 11-21;
   datalines;
Corn       16 27 31 33
Soybeans   21 25 23 24
Cotton     29 24 26 28
Sugarbeets54 23 21 54
Clover     32 32 62 16
;
```

```
title2 'Classification of Test Data';

proc discrim data=cropstat testdata=test testout=tout testlist;
   class Crop;
   testid xvalues;
   var x1-x4;
run;

proc print data=tout;
   title 'Discriminant Analysis of Remote Sensing Data on Five Crops';
   title2 'Output Classification Results of Test Data';
run;
```

Output 31.4.4 Classification of Test Data

```
           Discriminant Analysis of Remote Sensing Data on Five Crops
                          Classification of Test Data

                             The DISCRIM Procedure
                Classification Results for Test Data: WORK.TEST
           Classification Results using Linear Discriminant Function

                  Posterior Probability of Membership in Crop

                       Classified
xvalues     From Crop  into Crop        Clover       Corn     Cotton   Soybeans
                                     Sugarbeets

16 27 31 33 Corn       Corn             0.0894     0.4054     0.1763     0.2392
                                        0.0897
21 25 23 24 Soybeans   Soybeans         0.1481     0.2431     0.1200     0.3318
                                        0.1570
29 24 26 28 Cotton     Soybeans   *     0.2521     0.1842     0.1529     0.2549
                                        0.1559
54 23 21 54 Sugarbeets Clover     *     0.6215     0.0194     0.1250     0.0496
                                        0.1845
32 32 62 16 Clover     Cotton     *     0.2163     0.3180     0.3327     0.1125
                                        0.0206

                        * Misclassified observation

           Discriminant Analysis of Remote Sensing Data on Five Crops
                          Classification of Test Data

                             The DISCRIM Procedure
                Classification Summary for Test Data: WORK.TEST
           Classification Summary using Linear Discriminant Function

                       Observation Profile for Test Data

                   Number of Observations Read              5
                   Number of Observations Used              5
```

Output 31.4.4 *continued*

Number of Observations and Percent Classified into Crop

From Crop	Clover	Corn	Cotton	Soybeans	Sugarbeets	Total
Clover	0	0	1	0	0	1
	0.00	0.00	100.00	0.00	0.00	100.00
Corn	0	1	0	0	0	1
	0.00	100.00	0.00	0.00	0.00	100.00
Cotton	0	0	0	1	0	1
	0.00	0.00	0.00	100.00	0.00	100.00
Soybeans	0	0	0	1	0	1
	0.00	0.00	0.00	100.00	0.00	100.00
Sugarbeets	1	0	0	0	0	1
	100.00	0.00	0.00	0.00	0.00	100.00
Total	1	1	1	2	0	5
	20.00	20.00	20.00	40.00	0.00	100.00
Priors	0.30556	0.19444	0.16667	0.16667	0.16667	

Error Count Estimates for Crop

	Clover	Corn	Cotton	Soybeans	Sugarbeets	Total
Rate	1.0000	0.0000	1.0000	0.0000	1.0000	0.6389
Priors	0.3056	0.1944	0.1667	0.1667	0.1667	

Output 31.4.5 Output Data Set of the Classification Results for Test Data

Discriminant Analysis of Remote Sensing Data on Five Crops
Output Classification Results of Test Data

Obs	Crop	x1	x2	x3	x4	xvalues	Clover	Corn
1	Corn	16	27	31	33	16 27 31 33	0.08935	0.40543
2	Soybeans	21	25	23	24	21 25 23 24	0.14811	0.24308
3	Cotton	29	24	26	28	29 24 26 28	0.25213	0.18420
4	Sugarbeets	54	23	21	54	54 23 21 54	0.62150	0.01937
5	Clover	32	32	62	16	32 32 62 16	0.21633	0.31799

Obs	Cotton	Soybeans	Sugarbeets	_INTO_
1	0.17632	0.23918	0.08972	Corn
2	0.11999	0.33184	0.15698	Soybeans
3	0.15294	0.25486	0.15588	Soybeans
4	0.12498	0.04962	0.18452	Clover
5	0.33266	0.11246	0.02056	Cotton

In this next example, PROC DISCRIM uses normal-theory methods (METHOD=NORMAL) assuming unequal variances (POOL=NO) for the remote-sensing data. The PRIORS statement, PRIORS PROP, sets the prior probabilities proportional to the sample sizes. The CROSSVALIDATE option displays cross validation error-rate estimates. Note that the total error count estimate by cross validation (0.5556) is much larger than the total error count estimate by resubstitution (0.1111). The following statements produce Output 31.4.6:

```
title2 'Using Quadratic Discriminant Function';

proc discrim data=crops method=normal pool=no crossvalidate;
   class Crop;
   priors prop;
   id xvalues;
   var x1-x4;
run;
```

Output 31.4.6 Quadratic Discriminant Function on Crop Data

```
          Discriminant Analysis of Remote Sensing Data on Five Crops
                    Using Quadratic Discriminant Function

                           The DISCRIM Procedure

        Total Sample Size       36          DF Total                 35
        Variables                4          DF Within Classes        31
        Classes                  5          DF Between Classes        4

                  Number of Observations Read             36
                  Number of Observations Used             36

                          Class Level Information

             Variable                                                Prior
Crop         Name          Frequency       Weight    Proportion    Probability

Clover       Clover               11      11.0000      0.305556       0.305556
Corn         Corn                  7       7.0000      0.194444       0.194444
Cotton       Cotton                6       6.0000      0.166667       0.166667
Soybeans     Soybeans              6       6.0000      0.166667       0.166667
Sugarbeets   Sugarbeets            6       6.0000      0.166667       0.166667

                    Within Covariance Matrix Information

                                          Natural Log of the
                             Covariance   Determinant of the
              Crop          Matrix Rank    Covariance Matrix

              Clover                  4             23.64618
              Corn                    4             11.13472
              Cotton                  4             13.23569
              Soybeans                4             12.45263
              Sugarbeets              4             17.76293
```

Output 31.4.6 *continued*

Discriminant Analysis of Remote Sensing Data on Five Crops
Using Quadratic Discriminant Function

The DISCRIM Procedure

Generalized Squared Distance to Crop

From Crop	Clover	Corn	Cotton	Soybeans	Sugarbeets
Clover	26.01743	1320	104.18297	194.10546	31.40816
Corn	27.73809	14.40994	150.50763	38.36252	25.55421
Cotton	26.38544	588.86232	16.81921	52.03266	37.15560
Soybeans	27.07134	46.42131	41.01631	16.03615	23.15920
Sugarbeets	26.80188	332.11563	43.98280	107.95676	21.34645

Discriminant Analysis of Remote Sensing Data on Five Crops
Using Quadratic Discriminant Function

The DISCRIM Procedure
Classification Summary for Calibration Data: WORK.CROPS
Resubstitution Summary using Quadratic Discriminant Function

Number of Observations and Percent Classified into Crop

From Crop	Clover	Corn	Cotton	Soybeans	Sugarbeets	Total
Clover	9	0	0	0	2	11
	81.82	0.00	0.00	0.00	18.18	100.00
Corn	0	7	0	0	0	7
	0.00	100.00	0.00	0.00	0.00	100.00
Cotton	0	0	6	0	0	6
	0.00	0.00	100.00	0.00	0.00	100.00
Soybeans	0	0	0	6	0	6
	0.00	0.00	0.00	100.00	0.00	100.00
Sugarbeets	0	0	1	1	4	6
	0.00	0.00	16.67	16.67	66.67	100.00
Total	9	7	7	7	6	36
	25.00	19.44	19.44	19.44	16.67	100.00
Priors	0.30556	0.19444	0.16667	0.16667	0.16667	

Error Count Estimates for Crop

	Clover	Corn	Cotton	Soybeans	Sugarbeets	Total
Rate	0.1818	0.0000	0.0000	0.0000	0.3333	0.1111
Priors	0.3056	0.1944	0.1667	0.1667	0.1667	

Output 31.4.6 *continued*

Discriminant Analysis of Remote Sensing Data on Five Crops
Using Quadratic Discriminant Function

The DISCRIM Procedure
Classification Summary for Calibration Data: WORK.CROPS
Cross-validation Summary using Quadratic Discriminant Function

Number of Observations and Percent Classified into Crop

From Crop	Clover	Corn	Cotton	Soybeans	Sugarbeets	Total
Clover	9	0	0	0	2	11
	81.82	0.00	0.00	0.00	18.18	100.00
Corn	3	2	0	0	2	7
	42.86	28.57	0.00	0.00	28.57	100.00
Cotton	3	0	2	0	1	6
	50.00	0.00	33.33	0.00	16.67	100.00
Soybeans	3	0	0	2	1	6
	50.00	0.00	0.00	33.33	16.67	100.00
Sugarbeets	3	0	1	1	1	6
	50.00	0.00	16.67	16.67	16.67	100.00
Total	21	2	3	3	7	36
	58.33	5.56	8.33	8.33	19.44	100.00
Priors	0.30556	0.19444	0.16667	0.16667	0.16667	

Error Count Estimates for Crop

	Clover	Corn	Cotton	Soybeans	Sugarbeets	Total
Rate	0.1818	0.7143	0.6667	0.6667	0.8333	0.5556
Priors	0.3056	0.1944	0.1667	0.1667	0.1667	

References

Anderson, T. W. (1984), *An Introduction to Multivariate Statistical Analysis*, Second Edition, New York: John Wiley & Sons.

Epanechnikov, V. A. (1969), "Nonparametric Estimation of a Multivariate Probability Density," *Theory of Probability and Its Applications*, 14, 153–158.

Fisher, R. A. (1936), "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, 7, 179–188.

Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1977), "An Algorithm for Finding Best Matches in Logarithmic Expected Time," *ACM Transactions on Mathematical Software*, 3, 209–226.

Fukunaga, K. and Kessel, D. L. (1973), "Nonparametric Bayes Error Estimation Using Unclassified Samples," *IEEE Transactions on Information Theory*, 19, 434–440.

Glick, N. (1978), "Additive Estimators for Probabilities of Correct Classification," *Pattern Recognition*, 10, 211–222.

Hand, D. J. (1982), *Kernel Discriminant Analysis*, New York: Research Studies Press.

Hora, S. C. and Wilcox, J. B. (1982), "Estimation of Error Rates in Several-Population Discriminant Analysis," *Journal of Marketing Research*.

Kendall, M. G., Stuart, A., and Ord, J. K. (1983), *The Advanced Theory of Statistics*, volume 3, Fourth Edition, New York: Macmillan.

Kshirsagar, A. M. (1972), *Multivariate Analysis*, New York: Marcel Dekker.

Lachenbruch, P. A. and Mickey, M. A. (1968), "Estimation of Error Rates in Discriminant Analysis," *Technometrics*, 10, 1–10.

Lawley, D. N. (1959), "Tests of Significance in Canonical Analysis," *Biometrika*, 46, 59–66.

Morrison, D. F. (1976), *Multivariate Statistical Methods*, Second Edition, New York: McGraw-Hill.

Parzen, E. (1962), "On Estimation of a Probability Density Function and Mode," *Annals of Mathematical Statistics*, 33, 1065–1076.

Perlman, M. D. (1980), "Unbiasedness of the Likelihood Ratio Tests for Equality of Several Covariance Matrices and Equality of Several Multivariate Normal Populations," *Annals of Statistics*, 8, 247–263.

Puranen, J. (1917), "Fish Catch data set (1917)," Journal of Statistics Education Data Archive, last accessed May 22, 2009.
URL http://www.amstat.org/publications/jse/jse_data_archive.html

Rao, C. R. (1973), *Linear Statistical Inference*, New York: John Wiley & Sons.

Ripley, B. D. (1996), *Pattern Recognition and Neural Networks*, Cambridge: Cambridge University Press.

Rosenblatt, M. (1956), "Remarks on Some Nonparametric Estimates of a Density Function," *Annals of Mathematical Statistics*, 27, 832–837.

Chapter 32

The DISTANCE Procedure

Contents

Overview: DISTANCE Procedure **1483**
- Levels of Measurement 1484
- Symmetric versus Asymmetric Nominal Variables 1485
- Standardization 1485

Getting Started: DISTANCE Procedure **1486**
- Creating a Distance Matrix as Input for a Subsequent Cluster Analysis . . . 1486

Syntax: DISTANCE Procedure **1491**
- PROC DISTANCE Statement 1491
- VAR Statement 1500
- ID Statement 1505
- COPY Statement 1505
- BY Statement 1505
- FREQ Statement 1506
- WEIGHT Statement 1506

Details: DISTANCE Procedure **1507**
- Proximity Measures 1507
- Missing Values 1514
- Formatted versus Unformatted Values 1514
- Output Data Sets 1515

Examples: DISTANCE Procedure **1515**
- Example 32.1: Divorce Grounds – the Jaccard Coefficient 1515
- Example 32.2: Financial Data – Stock Dividends 1523

References **1531**

Overview: DISTANCE Procedure

The DISTANCE procedure computes various measures of distance, dissimilarity, or similarity between the observations (rows) of a SAS data set. These proximity measures are stored as a lower triangular matrix or a square matrix in an output data set (depending on the SHAPE= option) that can then be used as input to the CLUSTER, MDS, and MODECLUS procedures. The input data set might contain numeric or character variables, or both, depending on which proximity measure is used.

The number of rows and columns in the output matrix equals the number of observations in the input data set. If there are BY groups, an output matrix is computed for each BY group with the size determined by the maximum number of observations in any BY group.

PROC DISTANCE also provides various nonparametric and parametric methods for standardizing variables. Different variables can be standardized with different methods.

Distance matrices are used frequently in data mining, genomics, marketing, financial analysis, management science, education, chemistry, psychology, biology, and various other fields.

Levels of Measurement

Measurement of some attribute of a set of objects is the process of assigning numbers or other symbols to the objects in such a way that properties of the numbers or symbols reflect properties of the attribute being measured. There are different *levels* of measurement that involve different properties (relations and operations) of the numbers or symbols. Associated with each level of measurement is a set of transformations of the measurements that preserve the relevant properties; these transformations are called *permissible* transformations. A particular way of assigning numbers or symbols to measure something is called a *scale* of measurement.

The most commonly discussed levels of measurement are as follows:

Nominal	Two objects are assigned the same symbol if they have the same value of the attribute. Permissible transformations are any one-to-one or many-to-one transformation, although a many-to-one transformation loses information.
Ordinal	Objects are assigned numbers such that the order of the numbers reflects an order relation defined on the attribute. Two objects x and y with attribute values $a(x)$ and $a(y)$ are assigned numbers $m(x)$ and $m(y)$ such that if $m(x) > m(y)$, then $a(x) > a(y)$. Permissible transformations are any monotone increasing transformation, although a transformation that is not strictly increasing loses information.
Interval	Objects are assigned numbers such that differences between the numbers reflect differences of the attribute. If $m(x)-m(y) > m(u)-m(v)$, then $a(x)-a(y) > a(u)-a(v)$. Permissible transformations are any affine transformation $t(m) = c * m + d$, where c and d are constants; another way of saying this is that the origin and unit of measurement are arbitrary.
Log-interval	Objects are assigned numbers such that ratios between the numbers reflect ratios of the attribute. If $m(x)/m(y) > m(u)/m(v)$, then $a(x)/a(y) > a(u)/a(v)$. Permissible transformations are any power transformation $t(m) = c*m^d$, where c and d are constants.
Ratio	Objects are assigned numbers such that differences and ratios between the numbers reflect differences and ratios of the attribute. Permissible transformations are any linear (similarity) transformation $t(m) = c * m$, where c is a constant; another way of saying this is that the unit of measurement is arbitrary.

Absolute — Objects are assigned numbers such that all properties of the numbers reflect analogous properties of the attribute. The only permissible transformation is the identity transformation.

Proximity measures provided in the DISTANCE procedure accept four levels of measurement: nominal, ordinal, interval, and ratio. Ordinal variables are transformed to interval variables before processing. This is done by replacing the data with their rank scores, and by assuming that the classes of an ordinal variable are spaced equally along the interval scale. See the RANKSCORE= option in the section "PROC DISTANCE Statement" on page 1491 for choices on assigning scores to ordinal variables. There are also different approaches for how to transform an ordinal variable to an interval variable. See Anderberg (1973) for alternatives.

Symmetric versus Asymmetric Nominal Variables

A binary variable contains two possible outcomes: 1 (positive/present) or 0 (negative/absent). If there is no preference for which outcome should be coded as 0 and which as 1, the binary variable is called *symmetric*. For example, the binary variable "is evergreen?" for a plant has the possible states "loses leaves in winter" and "does not lose leaves in winter." Both are equally valuable and carry the same weight when a proximity measure is computed. Commonly used measures that accept symmetric binary variables include the Simple Matching, Hamann, Roger and Tanimoto, Sokal and Sneath 1, and Sokal and Sneath 3 coefficients.

If the outcomes of a binary variable are not equally important, the binary variable is called *asymmetric*. An example of such a variable is the presence or absence of a relatively rare attribute, such as "is color-blind" for a human being. While you say that two people who are color-blind have something in common, you cannot say that people who are not color-blind have something in common. The most important outcome is usually coded as 1 (present) and the other is coded as 0 (absent). The agreement of two 1's (a present-present match or a positive match) is more significant than the agreement of two 0's (an absent-absent match or a negative match). Usually, the negative match is treated as irrelevant. Commonly used measures that accept asymmetric binary variables include Jaccard, Dice, Russell and Rao, Binary Lance and Williams nonmetric, and Kulcynski coefficients.

When nominal variables are employed, the comparison of one data unit with another can only be in terms of whether the data units score the same or different on the variables. If a variable is defined as an asymmetric nominal variable and two data units score the same but fall into the absent category, the absent-absent match is excluded from the computation of the proximity measure.

Standardization

Since variables with large variances tend to have more effect on the proximity measure than those with small variances, it is recommended that you standardize the variables before the computation of the proximity measure. The DISTANCE procedure provides a convenient way to standardize each variable with its own method before the proximity measures are computed. You can also

perform the standardization by using the STDIZE procedure, with the limitation that all variables must be standardized with the same method.

Mandatory Standardization

Variable standardization is not required if any of the following conditions is true:

- if there is only one level of measurement
- if only asymmetric nominal and nominal levels are specified
- if the NOSTD option is specified in the PROC DISTANCE statement

Otherwise, standardization is mandatory.

When standardization is mandatory and no standardization method is specified, a default method of standardization will be used. This default method is determined by the measurement level. In general, the default method is STD for interval variables and is MAXABS for ratio variables except when METHOD=GOWER or METHOD=DGOWER is specified. See the STD= option in the section "VAR Statement" on page 1500 for the default methods for GOWER and DGOWER as well as methods available for standardizing variables.

When standardization is mandatory, PROC DISTANCE ignores the REPONLY option, if it is specified.

Getting Started: DISTANCE Procedure

Creating a Distance Matrix as Input for a Subsequent Cluster Analysis

The following example demonstrates how you can use the DISTANCE procedure to obtain a distance matrix that will be used as input to a subsequent clustering procedure.

The following data, originated by A. Weber and cited in Hand et al. (1994, pp. 297), measure the amount of protein consumed for nine food groups in 25 European countries. The nine food groups are red meat (RedMeat), white meat (WhiteMeat), eggs (Eggs), milk (Milk), fish (Fish), cereal (Cereal), starch (Starch), nuts (Nuts), and fruits and vegetables (FruitVeg). Suppose you want to determine whether national figures in protein consumption can be used to determine certain types or categories of countries; specifically, you want to perform a cluster analysis to determine whether these 25 countries can be formed into groups suggested by the data.

The following DATA step creates the SAS data set Protein:

```
data Protein;
   input Country $14. RedMeat WhiteMeat Eggs Milk
                      Fish Cereal Starch Nuts FruitVeg;
   datalines;
Albania        10.1  1.4  0.5   8.9  0.2  42.3  0.6  5.5  1.7
Austria         8.9 14.0  4.3  19.9  2.1  28.0  3.6  1.3  4.3
Belgium        13.5  9.3  4.1  17.5  4.5  26.6  5.7  2.1  4.0
Bulgaria        7.8  6.0  1.6   8.3  1.2  56.7  1.1  3.7  4.2
Czechoslovakia  9.7 11.4  2.8  12.5  2.0  34.3  5.0  1.1  4.0
Denmark        10.6 10.8  3.7  25.0  9.9  21.9  4.8  0.7  2.4
E Germany       8.4 11.6  3.7  11.1  5.4  24.6  6.5  0.8  3.6
Finland         9.5  4.9  2.7  33.7  5.8  26.3  5.1  1.0  1.4
France         18.0  9.9  3.3  19.5  5.7  28.1  4.8  2.4  6.5
Greece         10.2  3.0  2.8  17.6  5.9  41.7  2.2  7.8  6.5
Hungary         5.3 12.4  2.9   9.7  0.3  40.1  4.0  5.4  4.2
Ireland        13.9 10.0  4.7  25.8  2.2  24.0  6.2  1.6  2.9
Italy           9.0  5.1  2.9  13.7  3.4  36.8  2.1  4.3  6.7
Netherlands     9.5 13.6  3.6  23.4  2.5  22.4  4.2  1.8  3.7
Norway          9.4  4.7  2.7  23.3  9.7  23.0  4.6  1.6  2.7
Poland          6.9 10.2  2.7  19.3  3.0  36.1  5.9  2.0  6.6
Portugal        6.2  3.7  1.1   4.9 14.2  27.0  5.9  4.7  7.9
Romania         6.2  6.3  1.5  11.1  1.0  49.6  3.1  5.3  2.8
Spain           7.1  3.4  3.1   8.6  7.0  29.2  5.7  5.9  7.2
Sweden          9.9  7.8  3.5   4.7  7.5  19.5  3.7  1.4  2.0
Switzerland    13.1 10.1  3.1  23.8  2.3  25.6  2.8  2.4  4.9
UK             17.4  5.7  4.7  20.6  4.3  24.3  4.7  3.4  3.3
USSR            9.3  4.6  2.1  16.6  3.0  43.6  6.4  3.4  2.9
W Germany      11.4 12.5  4.1  18.8  3.4  18.6  5.2  1.5  3.8
Yugoslavia      4.4  5.0  1.2   9.5  0.6  55.9  3.0  5.7  3.2
;
```

The data set Protein contains the character variable Country and the nine numeric variables representing the food groups. The $14. in the INPUT statement specifies that the variable Country has a length of 14.

The following statements create the distance matrix and display part of it:

```
title 'Protein Consumption in Europe';
proc distance data=Protein out=Dist method=Euclid;
   var interval(RedMeat--FruitVeg / std=Std);
   id Country;
run;

proc print data=Dist(Obs=10);
title2 'First 10 observations in the output data set from PROC DISTANCE';
run;
```

An output SAS data set called Dist that contains the distance matrix is created through the OUT= option. METHOD=EUCLID requests that Euclidean (which also is the default) distances should be computed.

The VAR statement lists the variables (RedMeat—FruitVeg) along with their measurement level to be used in the analysis. An interval level of measurement is assigned to those variables. Since variables with large variances tend to have more effect on the proximity measure than those with small variances, each variable is standardized by the STD method to have a mean of 0 and a standard deviation of 1. This is done by adding "/ STD=STD" at the end of the variables list.

The ID statement specifies that the variable Country should be copied to the OUT= data set and used to generate names for the distance variables. The distance variables in the output data set are named by the values in the ID variable, and the maximum length for the names of these variables is 14.

There are 25 observations in the input data set; therefore, the output data set Dist contains a 25-by-25 lower triangular matrix.

The PROC PRINT statement displays the first 10 observations in the output data set Dist as shown in Figure 32.1.

Figure 32.1 First 10 Observations in the Output Data Set from PROC DISTANCE

Protein Consumption in Europe
First 10 observations in the output data set from PROC DISTANCE

Obs	Country	Albania	Austria	Belgium	Bulgaria	Czechoslovakia	Denmark
1	Albania	0.00000	.	.	.	.	.
2	Austria	6.12388	0.00000	.	.	.	.
3	Belgium	5.94109	2.44987	0.00000	.	.	.
4	Bulgaria	2.76446	4.88331	5.22711	0.00000	.	.
5	Czechoslovakia	5.13959	2.11498	2.21330	3.94761	0.00000	.
6	Denmark	6.61002	3.01392	2.52541	6.00803	3.34049	0.00000
7	E Germany	6.39178	2.56341	2.10211	5.40824	1.87962	2.72112
8	Finland	5.81458	4.04271	3.45779	5.74882	3.91378	2.61570
9	France	6.29601	3.58891	2.19329	5.54675	3.36011	3.65772
10	Greece	4.24495	5.16330	4.69515	3.74849	4.86684	5.59084

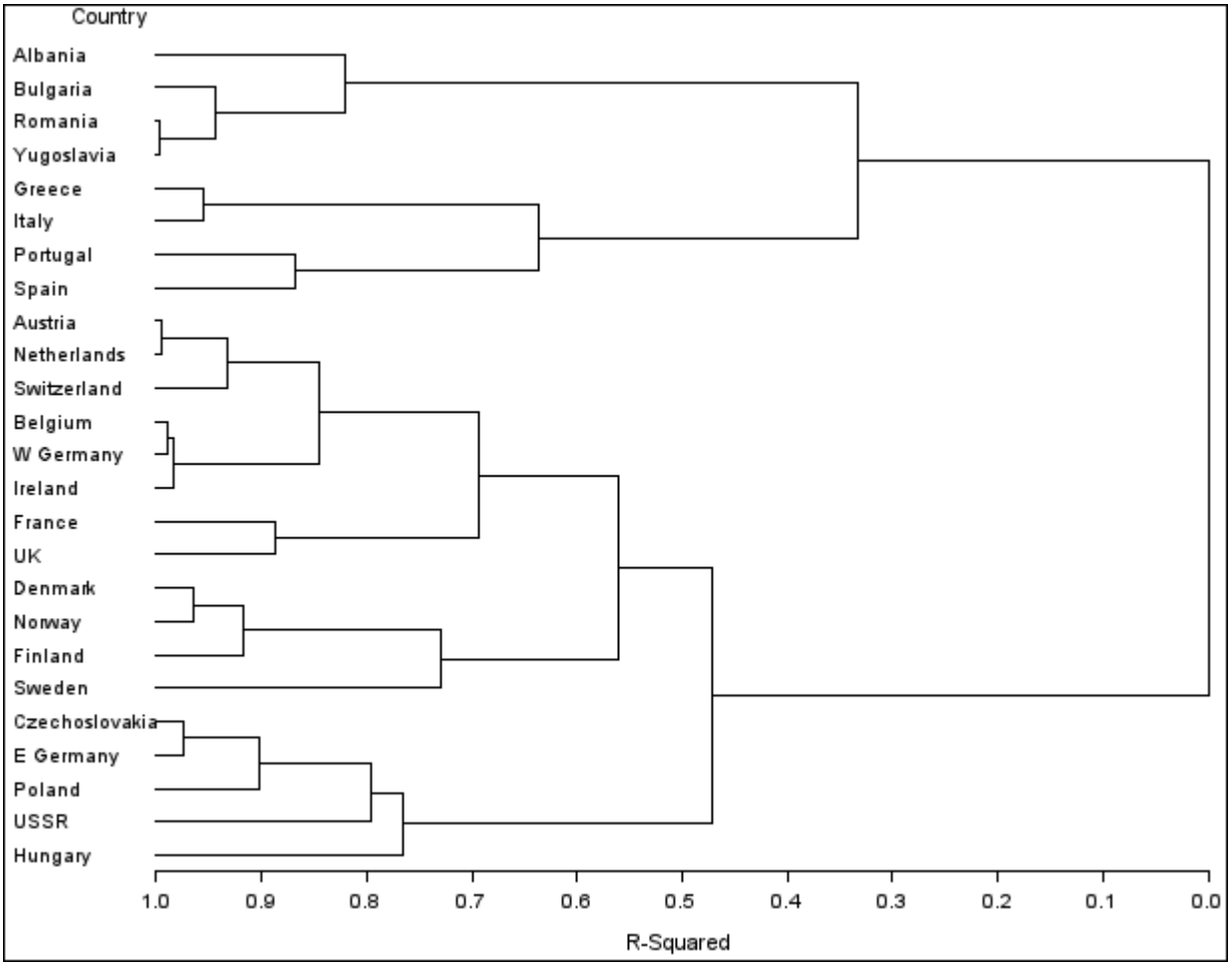
Obs	E_Germany	Finland	France	Greece	Hungary	Ireland	Italy	Netherlands	Norway	Poland	Portugal	Romania	Spain	Sweden	Switzerland	UK	USSR	W_Germany	Yugoslavia
1	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
2	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
3	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
4	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
5	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
6	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
7	0.00000	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
8	3.99426	0.00000	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
9	3.78184	4.56796	0.00000	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.
10	5.61496	5.47453	4.54456	0	.	.	.	.	.	.	.	.	.	.	.	.	.	.	.

The following statements produce the tree diagram in Figure 32.2:

```
proc cluster data=Dist method=Ward outtree=Tree noprint;
   id Country;
run;

axis1 order=(0 to 1 by 0.1);
proc tree data=Tree haxis=axis1 horizontal;
   height _rsq_;
   id Country;
run;
```

The CLUSTER procedure performs a Ward's minimum-variance cluster analysis based on the distance matrix created by the PROC DISTANCE. The printed output has been omitted, but the output data set Tree is created (through OUTTREE=TREE) and used as input to the TREE procedure, which produces the tree diagram shown in Figure 32.2. The HEIGHT statement specifies the variable _RSQ_ (the squared multiple correlation) as the height variable.

Figure 32.2 Tree Diagram of Clusters versus R-Squared Values



After inspecting the tree diagram in Figure 32.2, you will see that when the countries are grouped into six clusters, the proportion of variance accounted for by these clusters is slightly less than 70% (69.3%). The 25 countries are clustered as follows:

- Balkan countries: Albania, Bulgaria, Romania, and Yugoslavia
- Mediterranean countries: Greece and Italy
- Iberian countries: Portugal and Spain
- Western European countries: Austria, Netherlands, Switzerland, Belgium, former West Germany, Ireland, France, and U.K.
- Scandinavian countries: Denmark, Norway, Finland, and Sweden
- Eastern European countries: former Czechoslovakia, former East Germany, Poland, former U.S.S.R., and Hungary

Syntax: DISTANCE Procedure

The following statements are available in the DISTANCE procedure:

PROC DISTANCE < *options* > **;**
BY *variables* **;**
COPY *variables* **;**
FREQ *variable* **;**
ID *variable* **;**
VAR *level(variables* < */ opt-list* >*)* **;**
WEIGHT *variable* **;**

Both the PROC DISTANCE statement and the VAR statement are required.

PROC DISTANCE Statement

PROC DISTANCE < *options* > **;**

The options available with the PROC DISTANCE statement are summarized in Table 32.1 and discussed in the following section.

Table 32.1 Summary of PROC DISTANCE Statement Options

Option	Description
Standardize variables	
ADD=	specifies the constant to add to each value after standardizing and multiplying by the value specified in the MULT= option
FUZZ=	specifies the relative fuzz factor for writing the output
INITIAL=	specifies the method for computing initial estimates for the A-estimates
MULT=	specifies the constant to multiply each value by after standardizing
NORM	normalizes the scale estimator to be consistent for the standard deviation of a normal distribution
NOSTD	suppresses standardization
SNORM	normalizes the scale estimator to have an expectation of approximately 1 for a standard normal distribution
STDONLY	standardizes variables only (suppresses computation of the distance matrix)
VARDEF=	specifies the variances divisor
Generate distance matrix	
ABSENT=	specifies the value to be used as an absence value for all the asymmetric nominal variables
METHOD=	specifies the method for computing proximity measures
PREFIX=	specifies a prefix for naming the distance variables in the OUT= data set
RANKSCORE=	specifies the method of assigning scores to ordinal variables
SHAPE=	specifies the shape of the proximity matrix to be stored in the OUT= data set
UNDEF=	specifies the numeric constant used to replace undefined distances
Replace missing values	
NOMISS	omits observations with missing values from computation of the location and scale measures, if standardization applies; outputs missing values to the distance matrix for observations with missing values
REPLACE	replaces missing data with zero in the standardized data
REPONLY	replaces missing data with the location measure (does not standardize the data)

Table 32.1 *continued*

Option	Description
Specify data set details	
DATA=	specifies the input data set
OUT=	specifies the output data set
OUTSDZ=	specifies the output data set for standardized scores

These options and their abbreviations are described (in alphabetical order) in the remainder of this section.

ABSENT=*number* | *qs*

specifies the value to be used as an absence value in an irrelevant absent-absent match for *all* of the asymmetric nominal variables. If you want to specify a different absence value for a particular variable, use the ABSENT= option in the VAR statement. See the ABSENT= option in the section "VAR Statement" on page 1500 for details.

An absence value for a variable can be either a numeric value or a quoted string consisting of combinations of characters. For instance, ., -999, and "NA" are legal values for the ABSENT= option.

The default absence value for a character variable is "NONE" (notice that a blank value is considered a missing value), and the default absence value for a numeric variable is 0.

ADD=*c*

specifies a constant, c, to add to each value after standardizing and multiplying by the value you specify in the MULT= option. The default value is 0.

DATA=*SAS-data-set*

specifies the input data set containing observations from which the proximity is computed. If you omit the DATA= option, the most recently created SAS data set is used.

FUZZ=*c*

specifies the relative fuzz factor for computing the standardized scores. The default value is 1E–14. For the OUTSDZ= data set, the score is computed as follows:

$$\text{if } |\text{standardized scores}| < m \times c, \text{ then standardized scores} = 0$$

where m is the numeric constant specified in the MULT= option, or 1 if MULT= option is not specified.

INITIAL=*method*

specifies the method of computing initial estimates for the A-estimates (ABW, AWAVE, and AHUBER). The following methods are not allowed for the INITIAL= option: ABW, AHUBER, AWAVE, and IN.

The default value is INITIAL=MAD.

METHOD=*method*

specifies the method of computing proximity measures.

For use in PROC CLUSTER, distance or dissimilarity measures such as METHOD=EUCLID or METHOD=DGOWER should be chosen.

The following six tables outline the proximity measures available for the METHOD= option. These tables are classified by levels of measurement accepted by each method. There are three to four columns in each table: the proximity measures (Method) column, the upper and lower bounds (Range) column(s), and the types of proximity (Type) column.

The Type column has two possible values: "sim" if a method generate similarity or "dis" if a method generates distance or dissimilarity measures.

For formulas and descriptions of these methods, see the section "Details: DISTANCE Procedure" on page 1507.

Table 32.2 lists the range and output matrix type of the GOWER and DGOWER methods. These two methods accept all measurement levels including ratio, interval, ordinal, nominal, and asymmetric nominal. METHOD=GOWER or METHOD=DGOWER always implies standardization. Assuming all the numeric (ordinal, interval, and ratio) variables are standardized by their corresponding default methods, the possible range values for both methods in the second column of this table are on or between 0 and 1. To find out the default methods of standardization for METHOD=GOWER or METHOD=DGOWER, see the STD= option in the section "VAR Statement" on page 1500. Entries in this table are as follows:

GOWER	Gower's similarity
DGOWER	1 minus GOWER

Table 32.2 Methods Accepting All Measurement Levels

Method	Range	Type
GOWER	0 to 1	sim
DGOWER	0 to 1	dis

Table 32.3 lists methods accepting ratio, interval, and ordinal variables. Entries in this table are as follows:

EUCLID	Euclidean distance
SQEUCLID	squared Euclidean distance
SIZE	size distance
SHAPE	shape distance
COV	covariance
CORR	correlation
DCORR	correlation transformed to Euclidean distance
SQCORR	squared correlation

DSQCORR	one minus squared correlation
L(p)	Minkowski (L_p) distance, where p is a positive numeric value
CITYBLOCK	L_1, city-block, or Manhattan distance
CHEBYCHEV	L_∞
POWER(p, r)	generalized Euclidean distance where p is a positive numeric value and r is a nonnegative numeric value. The distance between two observations is the rth root of sum of the absolute differences to the pth power between the values for the observations.

Table 32.3 Methods Accepting Ratio, Interval, and Ordinal Variables

Method	Range	Type
EUCLID	≥ 0	dis
SQEUCLID	≥ 0	dis
SIZE	≥ 0	dis
SHAPE	≥ 0	dis
COV	≥ 0	sim
CORR	–1 to 1	sim
DCORR	0 to 2	dis
SQCORR	0 to 1	sim
DSQCORR	0 to 1	dis
L(p)	≥ 0	dis
CITYBLOCK	≥ 0	dis
CHEBYCHEV	≥ 0	dis
POWER(p, r)	≥ 0	dis

Table 32.4 lists methods accepting ratio variables. Notice that in the second column of this table, all of the possible range values are nonnegative, because ratio variables are assumed to be positive. Entries in this table are as follows:

SIMRATIO	similarity ratio (if variables are binary, this is the Jaccard coefficient)
DISRATIO	one minus similarity ratio
NONMETRIC	Lance and Williams nonmetric coefficient
CANBERRA	Canberra metric distance coefficient
COSINE	cosine coefficient
DOT	dot (inner) product coefficient
OVERLAP	overlap similarity
DOVERLAP	overlap dissimilarity
CHISQ	chi-squared coefficient
CHI	squared root of chi-squared coefficient
PHISQ	phi-squared coefficient
PHI	squared root of phi-squared coefficient

Table 32.4 Methods Accepting Ratio Variables

Method	Range	Type
SIMRATIO	0 to 1	sim
DISRATIO	0 to 1	dis
NONMETRIC	0 to 1	dis
CANBERRA	0 to 1	dis
COSINE	0 to 1	sim
DOT	≥ 0	sim
OVERLAP	≥ 0	sim
DOVERLAP	≥ 0	dis
CHISQ	≥ 0	dis
CHI	≥ 0	dis
PHISQ	≥ 0	dis
PHI	≥ 0	dis

Table 32.5 lists methods accepting nominal variables. Entries in this table are:

HAMMING	Hamming distance
MATCH	simple matching coefficient
DMATCH	simple matching coefficient transformed to Euclidean distance
DSQMATCH	simple matching coefficient transformed to squared Euclidean distance
HAMANN	Hamann coefficient
RT	Roger and Tanimoto
SS1	Sokal and Sneath 1
SS3	Sokal and Sneath 3

Table 32.5 Methods Accepting Nominal Variables

Method	Range	Type
HAMMING	0 to v	dis
MATCH	0 to 1	sim
DMATCH	0 to 1	dis
DSQMATCH	0 to 1	dis
HAMANN	–1 to 1	sim
RT	0 to 1	sim
SS1	0 to 1	sim
SS3	0 to 1	sim

Note that v denotes the number of variables or dimensionality.

Table 32.6 lists methods that accept asymmetric nominal variables. Use the ABSENT= option to create a value to be considered absent. Entries in this table are as follows:

DICE	Dice coefficient or Czekanowski/Sorensen similarity coefficient
RR	Russell and Rao
BLWNM	Binary Lance and Williams nonmetric, or Bray-Curtis coefficient
K1	Kulcynski 1

Table 32.6 Methods Accepting Asymmetric Nominal Variables

Method	Range	Type
DICE	0 to 1	sim
RR	0 to 1	sim
BLWNM	0 to 1	dis
K1	≥ 0	sim

Table 32.7 lists methods accepting asymmetric nominal and ratio variables. Use the ABSENT= option to create a value to be considered absent. There are four instead of three columns in this table. The second column contains possible range values if only one level of measurement (either ratio or asymmetric nominal but not both) is specified; the third column contains possible range values if both levels are specified.

The JACCARD method is equivalent to the SIMRATIO method if there is no asymmetric nominal variable; if both ratio and asymmetric nominal variables are present, the coefficient is computed as the sum of the coefficient from the ratio variables and the coefficient from the asymmetric nominal variables. See "Proximity Measures" in the section "Details: DISTANCE Procedure" on page 1507 for the formula and descriptions of the JACCARD method. Entries in this table are as follows:

JACCARD	Jaccard similarity coefficient
DJACCARD	Jaccard dissimilarity coefficient

Table 32.7 Methods Accepting Asymmetric Nominal and Ratio Variables

Method	Range (one level)	Range (two levels)	Type
JACCARD	0 to 1	0 to 2	sim
DJACCARD	0 to 1	0 to 2	dis

MULT=*c*

specifies a numeric constant, c, by which to multiply each value after standardizing. The default value is 1.

NOMISS

omits observations with missing values from computation of the location and scale measures when standardizing; generates undefined (missing) distances for observations with missing values when computing distances. Use the UNDEF= option to specify the undefined values.

If a distance matrix is created to be used as an input to PROC CLUSTER, the NOMISS option should not be used because the CLUSTER procedure does not accept distance matrices with missing values.

NORM

normalizes the scale estimator to be consistent for the standard deviation of a normal distribution when you specify the option STD=AGK, STD=IQR, STD=MAD, or STD=SPACING in the VAR statement.

NOSTD

suppresses standardization of the variables. The NOSTD option should not be specified with the STDONLY option or with the REPLACE option.

PREFIX=*name*

specifies a prefix for naming the distance variables in the OUT= data set. By default, the names are Dist1, Dist2, ..., Dist*n*. If you specify PREFIX=ABC, the variables are named ABC1, ABC2, ..., ABCn. If the ID statement is also specified, the variables are named by appending the value of the ID variable to the prefix.

OUT=*SAS-data-set*

specifies the name of the SAS data set created by PROC DISTANCE. The output data set contains the BY variables, the ID variable, computed distance variables, the COPY variables, the FREQ variable, and the WEIGHT variables.

If you omit the OUT= option, PROC DISTANCE creates an output data set named according to the DATA*n* convention.

OUTSDZ=*SAS-data-set*

specifies the name of the SAS data set containing the standardized scores. The output data set contains a copy of the DATA= data set, except that the analyzed variables have been standardized. Analyzed variables are those listed in the VAR statement.

RANKSCORE=MIDRANK | INDEX

specifies the method of assigning scores to ordinal variables. The available methods are listed as follows:

MIDRANK	assigns consecutive integers to each category with consideration of the frequency value. This is the default method.
INDEX	assigns consecutive integers to each category regardless of frequencies.

The following example explains how each method assigns the rank scores. Suppose the data contain an ordinal variable ABC with values A, B, C. There are two ways to assign numbers. One is to use midranks, which depend on the frequencies of each category. Another is to assign consecutive integers to each category, regardless of frequencies.

Table 32.8 Example of Assigning Rank Scores

ABC	MIDRANK	INDEX
A	1.5	1
A	1.5	1
B	4	2
B	4	2
B	4	2
C	6	3

REPLACE
replaces missing data with zero in the standardized data (to correspond to the location measure before standardizing). To replace missing data with something else, use the MISSING= option in the VAR statement. The REPLACE option implies standardization.

You cannot specify the following options together:

- both the REPLACE and the REPONLY options
- both the REPLACE and the NOSTD options

REPONLY
replaces missing data with the location measure specified by the MISSING= option or the STD= option (if the MISSING= option is not specified), but does *not* standardize the data. If the MISSING= option is not specified and METHOD=GOWER is specified, missing values are replaced by the location measure from the RANGE method (the minimum value), no matter what the value of the STD= option is.

You cannot specify both the REPLACE and the REPONLY options.

SHAPE=TRIANGLE | TRI | SQUARE | SQU | SQR
specifies the shape of the proximity matrix to be stored in the OUT= data set. SHAPE=TRIANGLE requests the matrix to be stored as a lower triangular matrix; SHAPE=SQUARE requests that the matrix be stored as a squared matrix. Use SHAPE=SQUARE if the output data set is to be used as input to the MODECLUS procedures. The default is TRIANGLE.

SNORM
normalizes the scale estimator to have an expectation of approximately 1 for a standard normal distribution when the STD=SPACING option is specified.

STDONLY
standardizes variables only and computes no distance matrix. You must use the OUTSDZ= option to save the standardized scores. You cannot specify both the STDONLY option and the NOSTD option.

UNDEF=*n*
specifies the numeric constant used to replace undefined distances, such as when an observation has all missing values, or if a divisor is zero.

VARDEF=DF | N | WDF | WEIGHT | WGT

specifies the divisor to be used in the calculation of distance, dissimilarity, or similarity measures, and for standardizing variables whenever a variance or covariance is computed. By default, VARDEF=DF. The values and associated divisors are as follows:

Value	Divisor	Formula
DF	degrees of freedom	$n - 1$
N	number of observations	n
WDF	sum of weights minus 1	$(\sum_i w_i) - 1$
WEIGHT | WGT	sum of weights	$\sum_i w_i$

VAR Statement

VAR *level (variables* < */ opt-list* > *)*
< *level (variables* < */ opt-list* > *) . . . level (variables* < */ opt-list* > *)* > **;**

where the syntax for the *opt-list* is as follows:

ABSENT=*value*
MISSING=*miss-method* | *value*
ORDER=*order-option*
STD=*std-method*
WEIGHTS=*weight-list*

The VAR statement lists variables from which distances are to be computed. The VAR statement is required. The variables can be numeric or character depending on their measurement levels. A variable cannot appear more than once in either the same list or a different list.

level is required. It declares the levels of measurement for those variables specified within the parentheses. Available values for *level* are as follows:

ANOMINAL — variables are asymmetric nominal and can be either numeric or character.

NOMINAL — variables are symmetric nominal and can be either numeric or character.

ORDINAL — variables are ordinal and can be either numeric or character. Values of ordinal variables are replaced by their corresponding rank scores. If standardization is required, the standardized rank scores are output to the data set specified in the OUTSDZ= option. See the RANKSCORE= option in the PROC DISTANCE statement for methods available for assigning rank scores to ordinal variables. After being replaced by scores, ordinal variables are considered interval.

INTERVAL — variables are interval and numeric.

RATIO — variables are ratio and numeric. Ratio variables should always contain positive measurements.

Each variable list can be followed by an option list. Use "/ " after the list of variables to start the option list. An option list contains options that are applied to the variables. The following options are available in the option list:

ABSENT=	specifies the value to be used as an absence value in an irrelevant absent-absent match for asymmetric nominal variables.
MISSING=	specifies the method (or numeric value) with which to replace missing data.
ORDER=	selects the order for assigning scores to ordinal variables.
STD=	selects the standardization method.
WEIGHTS=	assigns weights to the variables in the list.

If an option is missing from the current attribute list, PROC DISTANCE provides default values for all the variables in the current list.

For example, in the VAR statement

```
var ratio(x1-x4/std= mad weights= .5 .5 .1 .5 missing= -99)
    interval(x5/std= range)
    ordinal(x6/order= desc);
```

the first option list defines x1–x4 as ratio variables to be standardized by the MAD method. Also, any missing values in x1–x4 should be replaced by –99. x1 is given a weight of 0.5, x2 is given a weight of 0.5, x3 is given a weight of 0.1, and x4 is given a weight of 0.5.

The second option list defines x5 as an interval variable to be standardized by the RANGE method. If the REPLACE option is specified in the PROC DISTANCE statement, missing values in x5 are replaced by the location estimate from the RANGE method. By default, x5 is given a weight of 1.

The last option list defines x6 as an ordinal variable. The scores are assigned from highest to lowest by its unformatted values. Although the STD= option is not specified, x6 is standardized by the default method (STD) because there is more than one level of measurements (ratio, interval, and ordinal) in the VAR statement. Again, if the REPLACE option is specified, missing values in x6 are replaced by the location estimate from the STD method. Finally, by default, x6 is given a weight of 1.

More details for the options are explained as follows.

STD=*std-method*

specifies the standardization method. Valid values for *std-method* are MEAN, MEDIAN, SUM, EUCLEN, USTD, STD, RANGE, MIDRANGE, MAXABS, IQR, MAD, ABW, AHUBER, AWAVE, AGK, SPACING, and L. Table 32.9 lists available methods of standardization as well as their corresponding location and scale measures.

Table 32.9 Available Standardization Methods

Method	Scale	Location
MEAN	1	mean
MEDIAN	1	median
SUM	sum	0
EUCLEN	Euclidean length	0
USTD	standard deviation about origin	0
STD	standard deviation	mean
RANGE	range	minimum
MIDRANGE	range/2	midrange
MAXABS	maximum absolute value	0
IQR	interval quartile range	median
MAD	median absolute deviation from median	median
ABW(c)	biweight A-estimate	biweight 1-step M-estimate
AHUBER(c)	Huber A-estimate	Huber 1-step M-estimate
AWAVE(c)	Wave 1-step M-estimate	Wave A-estimate
AGK(p)	AGK estimate (ACECLUS)	mean
SPACING(p)	minimum spacing	mid minimum-spacing
L(p)	L_p	L_p

These standardization methods are further documented in the section on the METHOD= option in the PROC STDIZE statement of the STDIZE procedure (see the section "Standardization Methods" on page 6229 in Chapter 81, "The STDIZE Procedure").

Standardization is not required if there is only one level of measurement, or if only asymmetric nominal and nominal levels are specified; otherwise, standardization is mandatory (with the exception when the NOSTD option is specified). When standardization is mandatory, a default method is provided when the STD= option is not specified. The default method is STD for standardizing interval variables and MAXABS for standardizing ratio variables unless METHOD=GOWER or METHOD=DGOWER is specified. If METHOD=GOWER is specified, interval variables are standardized by the RANGE method, and whatever is specified in the STD= option is ignored; if METHOD=DGOWER is specified, the RANGE method is the default standardization method for interval variables. The MAXABS method is the default standardization method for ratio variables for both the GOWER and DGOWER methods.

Notice that a ratio variable should always be positive.

Table 32.10 lists standardization methods and the levels of measurement that can be accepted by each method. For example, the SUM method can be used to standardize ratio variables but not interval or ordinal variables. Also, the AGK and SPACING methods should not be used to standardize ordinal variables. If you apply AGK and SPACING to ranks, the results are degenerate because all the spacings of a given order are equal.

Table 32.10 Legitimate Levels of Measurements for Each Method

Standardization Method	Legitimate Levels of Measurement
MEAN	ratio, interval, ordinal
MEDIAN	ratio, interval, ordinal
SUM	ratio
EUCLEN	ratio
USTD	ratio
STD	ratio, interval, ordinal
RANGE	ratio, interval, ordinal
MIDRANGE	ratio, interval, ordinal
MAXABS	ratio
IQR	ratio, interval, ordinal
MAD	ratio, interval, ordinal
ABW(c)	ratio, interval, ordinal
AHUBER(c)	ratio, interval, ordinal
AWAVE(c)	ratio, interval, ordinal
AGK(p)	ratio, interval
SPACING(p)	ratio, interval
L(p)	ratio, interval, ordinal

ABSENT=*numner* | *qs*

specifies the value to be used as an absence value in an irrelevant absent-absent match for asymmetric nominal variables. The absence value specified here overwrites the absence value specified through the ABSENT= option in the PROC DISTANCE statement for those variables in the current variable list.

An absence value for a variable can be either a numeric value or a quoted string consisting of combinations of characters. For instance, ., –999, "NA" are legal values for the ABSENT= option.

The default for an absence value for a character variable is "NONE" (notice that a blank value is considered a missing value), and the default for an absence value for a numeric variable is 0.

MISSING=*miss-method* | *value*

specifies the method or a numeric value for replacing missing values. If you omit the MISSING= option, the REPLACE option replaces missing values with the location measure given by the STD= option. Specify the MISSING= option when you want to replace missing values with a different value. You can specify any method that is valid in the STD= option. The corresponding location measure is used to replace missing values.

If a numeric value is given, the value replaces missing values after standardizing the data. However, when standardization is not mandatory, you can specify the REPONLY option with the MISSING= option to suppress standardization for cases in which you want only to replace missing values.

If the NOSTD option is specified, there is no standardization, but missing values are replaced by the corresponding location measures or by the numeric value of the MISSING= option. See the section "Missing Values" on page 1514 for details about missing values replacement with and without standardization.

ORDER=ASCENDING | ASC
ORDER=DESCENDING | DESC
ORDER=ASCFORMATTED | ASCFMT
ORDER=DESFORMATTED | DESFMT
ORDER=DSORDER | DATA

specifies the order for assigning score to ordinal variables. The value for the ORDER= option can be one of the following:

ASCENDING	scores are assigned in lowest-to-highest order of unformatted values.
DESCENDING	scores are assigned in highest-to-lowest order of unformatted values.
ASCFORMATTED	scores are assigned in ascending order by their formatted values. This option can be applied to character variables only, since unformatted values are always used for numeric variables.
DESFORMATTED	scores are assigned in descending order by their formatted values. This option can be applied to character variables only, since unformatted values are always used for numeric variables.
DSORDER	scores are assigned according to the order of their appearance in the input data set.

The default value is ASCENDING.

WEIGHTS=*weight-list*

specifies a list of values for weighting individual variables while computing the proximity. Values in this list can be separated by blanks or commas. You can include one or more items of the form *start* TO *stop* BY *increment*. This list should contain at least one weight. The maximum number of weights you can list is equal to the number of variables. If the number of weights is less than the number of variables, the last value in the *weight-list* is used for the rest of the variables; conversely, if the number of weights is greater than the number of variables, the trailing weights are discarded.

The default value is 1.

ID Statement

ID *variable* **;**

The ID statement specifies a single variable to be copied to the OUT= data set and used to generate names for the distance variables. The ID variable must be character.

Typically, each ID value occurs only once in the input data set or, if you use a BY statement, only once within a BY group.

If you specify both the ID and BY statements, the ID variable must have the same values in the same order in each BY group.

COPY Statement

COPY *variables* **;**

The COPY statement specifies a list of additional variables to be copied to the OUT= data set.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC DISTANCE to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC DISTANCE. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide.*

FREQ Statement

FREQ | **FREQUENCY** *variable* **;**

The frequency variable is used for either standardizing variables or assigning rank scores to the ordinal variables. It has no direct effect on computing the distances.

For standardizing variables and assigning rank scores, PROC DISTANCE treats the data set as if each observation appeared n times, where n is the value of the FREQ variable for the observation. Nonintegral values of the FREQ variable are truncated to the largest integer less than the FREQ value. If the FREQ variable has a value that is less than 1 or is missing, the observation is not used in the analysis.

WEIGHT Statement

WGT | **WEIGHT** *variable* **;**

The WEIGHT statement specifies a numeric variable in the input data set with values that are used to weight each observation. This weight variable is used for standardizing variables rather than computing the distances. Only one variable can be specified.

The WEIGHT variable values can be nonintegers. An observation is used in the analysis only if the value of the WEIGHT variable is greater than zero. The WEIGHT variable applies to variables that are standardized by the following options: STD=MEAN, STD=SUM, STD=EUCLEN, STD=USTD, STD=STD, STD=AGK, or STD=L.

PROC DISTANCE uses the value of the WEIGHT variable w_i to compute the sample mean, uncorrected sample variances, and sample variances as follows:

$$\overline{x}_w = \sum_i w_i x_i / \sum_i w_i$$

$$u_w^2 = \sum_i w_i x_i^2 / d$$

$$s_w^2 = \sum_i w_i (x_i - \overline{x}_w)^2 / d$$

w_i is the weight value of the ith observation, x_i is the value of the ith observation, and d is the divisor controlled by the VARDEF= option (see the VARDEF= option in the PROC DISTANCE statement for details).

PROC DISTANCE uses the value of the WEIGHT variable to calculate the following statistics for standardization:

MEAN — the weighted mean, $\overline{x}_w$

SUM — the weighted sum, $\sum_i w_i x_i$

USTD — the weighted uncorrected standard deviation, $\sqrt{u_w^2}$

STD — the weighted standard deviation, $\sqrt{s_w^2}$

EUCLEN — the weighted Euclidean length, computed as the square root of the weighted uncorrected sum of squares:

$$\sqrt{\sum_i w_i x_i^2}$$

AGK — the AGK estimate. This estimate is documented further in the ACECLUS procedure as the METHOD=COUNT option. See the discussion of the WEIGHT statement in Chapter 22, "The ACECLUS Procedure," for information about how the WEIGHT variable is applied to the AGK estimate.

L — the L_p estimate. This estimate is documented further in the FASTCLUS procedure as the LEAST= option. See the discussion of the WEIGHT statement in Chapter 34, "The FASTCLUS Procedure," for information about how the WEIGHT variable is used to compute weighted cluster means. Note that the number of clusters is always 1.

Details: DISTANCE Procedure

Proximity Measures

The following notation is used in this section:

v — the number of variables or the dimensionality

x_j — data for observation x and the jth variable, where $j = 1$ to v

y_j — data for observation y and the jth variable, where $j = 1$ to v

w_j — weight for the jth variable from the WEIGHTS= option in the VAR statement. $w_j = 0$ when either x_j or y_j is missing.

W — the sum of total weights. No matter if the observation is missing or not, its weight is added to this metric.

$\bar{x}$ — mean for observation x

$\bar{x} = \sum_{j=1}^{v} w_j x_j / \sum_{j=1}^{v} w_j$

$\bar{y}$ — mean for observation y

$\bar{y} = \sum_{j=1}^{v} w_j y_j / \sum_{j=1}^{v} w_j$

$d(x, y)$ — the distance or dissimilarity between observations x and y

$s(x, y)$ the similarity between observations x and y

The factor $W/\sum_{j=1}^{v} w_j$ is used to adjust some of the proximity measures for missing values.

Methods Accepting All Measurement Levels

GOWER Gower's similarity

$s_1(x, y) = \sum_{j=1}^{v} w_j \delta_{x,y}^j d_{x,y}^j / \sum_{j=1}^{v} w_j \delta_{x,y}^j$

$\delta_{x,y}^j$ is computed as follows:

For nominal, ordinal, interval, or ratio variable,

$$\delta_{x,y}^j = 1$$

For asymmetric nominal variable,

$$\delta_{x,y}^j = 1, \text{ if either } x_j \text{ or } y_j \text{ is present}$$
$$\delta_{x,y}^j = 0, \text{ if both } x_j \text{ and } y_j \text{ are absent}$$

For nominal or asymmetric nominal variable,

$$d_{x,y}^j = 1, \text{ if } x_j = y_j$$
$$d_{x,y}^j = 0, \text{ if } x_j \neq y_j$$

For ordinal, interval, or ratio variable,

$$d_{x,y}^j = 1 - |x_j - y_j|$$

DGOWER 1 minus Gower

$d_2(x, y) = 1 - s_1(x, y)$

Methods Accepting Ratio, Interval, and Ordinal Variables

EUCLID Euclidean distance

$d_3(x, y) = \sqrt{(\sum_{j=1}^{v} w_j (x_j - y_j)^2) W / (\sum_{j=1}^{v} w_j)}$

SQEUCLID squared Euclidean distance

$d_4(x, y) = (\sum_{j=1}^{v} w_j (x_j - y_j)^2) W / (\sum_{j=1}^{v} w_j)$

SIZE size distance

$d_5(x, y) = |\sum_{j=1}^{v} w_j (x_j - y_j)| \sqrt{W} / (\sum_{j=1}^{v} w_j)$

SHAPE shape distance

$d_6(x, y) = \sqrt{(\sum_{j=1}^{v} w_j [(x_j - \bar{x}) - (y_j - \bar{y})]^2) W / (\sum_{j=1}^{v} w_j)}$

NOTE: squared shape distance plus squared size distance equals squared Euclidean distance.

COV — covariance similarity coefficient

$s_7(x,y) = \sum_{j=1}^{v} w_j(x_j - \bar{x})(y_j - \bar{y})/vardiv$, where

$$\begin{aligned} vardiv &= v \text{ if VARDEF = N} \\ &= v-1 \text{ if VARDEF = DF} \\ &= \sum_{j=1}^{v} w_j \text{ if VARDEF = WEIGHT} \\ &= \sum_{j=1}^{v} w_j - 1 \text{ if VARDEF = WDF} \end{aligned}$$

CORR — correlation similarity coefficient

$$s_8(x,y) = \frac{\sum_{j=1}^{v} w_j(x_j-\bar{x})(y_j-\bar{y})}{\sqrt{\sum_{j=1}^{v} w_j(x_j-\bar{x})^2 \sum_{j=1}^{v} w_j(y_j-\bar{y})^2}}$$

DCORR — correlation transformed to Euclidean distance as sqrt(1–CORR)

$d_9(x,y) = \sqrt{1 - s_8(x,y)}$

SQCORR — squared correlation

$$s_{10}(x,y) = \frac{[\sum_{j=1}^{v} w_j(x_j-\bar{x})(y_j-\bar{y})]^2}{\sum_{j=1}^{v} w_j(x_j-\bar{x})^2 \sum_{j=1}^{v} w_j(y_j-\bar{y})^2}$$

DSQCORR — squared correlation transformed to squared Euclidean distance as (1–SQCORR)

$d_{11}(x,y) = 1 - s_{10}(x,y)$

L(p) — Minkowski (L_p) distance, where p is a positive numeric value

$d_{12}(x,y) = [(\sum_{j=1}^{v} w_j|x_j - y_j|^p)W/(\sum_{j=1}^{v} w_j)]^{1/p}$

CITYBLOCK — L_1

$d_{13}(x,y) = (\sum_{j=1}^{v} w_j|x_j - y_j|)W/(\sum_{j=1}^{v} w_j)$

CHEBYCHEV — L_∞

$d_{14}(x,y) = \max_{j=1}^{v} w_j|x_j - y_j|$

POWER(p,r) — generalized Euclidean distance, where p is a nonnegative numeric value and r is a positive numeric value. The distance between two observations is the rth root of sum of the absolute differences to the pth power between the values for the observations:

$d_{15}(x,y) = [(\sum_{j=1}^{v} w_j|x_j - y_j|^p)W/(\sum_{j=1}^{v} w_j)]^{1/r}$

Methods Accepting Ratio Variables

SIMRATIO — similarity ratio

$$s_{16}(x,y) = \frac{\sum_{j}^{v} w_j(x_j y_j)}{\sum_{j=1}^{v} w_j(x_j y_j) + \sum_{j}^{v} w_j(x_j - y_j)^2}$$

DISRATIO one minus similarity ratio
$d_{17}(x, y) = 1 - s_{16}(x, y)$

NONMETRIC Lance-Williams nonmetric coefficient
$d_{18}(x, y) = \frac{\sum_{j=1}^{v} w_j |x_j - y_j|}{\sum_{j=1}^{v} w_j (x_j + y_j)}$

CANBERRA Canberra metric coefficient. See Sneath and Sokal (1973), pp. 125–126.
$d_{19}(x, y) = \sum_{j=1}^{v} \frac{w_j |x_j - y_j|}{w_j (x_j + y_j)}$

COSINE cosine coefficient
$s_{20}(x, y) = \frac{\sum_{j=1}^{v} w_j (x_j y_j)}{\sqrt{\sum_{j=1}^{v} w_j x_j{}^2 \sum_{j=1}^{v} w_j y_j{}^2}}$

DOT dot (inner) product coefficient
$s_{21}(x, y) = [\sum_{j=1}^{v} w_j (x_j y_j)] / \sum_{j=1}^{v} w_j$

OVERLAP sum of the minimum values
$s_{22}(x, y) = \sum_{j=1}^{v} w_j [\min(x_j, y_j)]$

DOVERLAP maximum of the sum of the x and the sum of y minus overlap
$d_{23}(x, y) = \max(\sum_{j=1}^{v} w_j x_j, \sum_{j=1}^{v} w_j y_j) - s_{22}(x, y)$

CHISQ chi-squared
If the data represent the frequency counts, chi-squared dissimilarity between two sets of frequencies can be computed. A 2-by-v contingency table is illustrated to explain how the chi-squared dissimilarity is computed as follows:

	Variable				Row
Observation	Var 1	Var 2	...	Var v	Sum
X	x_1	x_2	...	x_v	r_x
Y	y_1	y_2	...	y_v	r_y
Column Sum	c_1	c_2	...	c_v	T

where

$$\begin{aligned} r_x &= \sum_{j=1}^{v} w_j x_j \\ r_y &= \sum_{j=1}^{v} w_j y_j \\ c_j &= w_j (x_j + y_j) \\ T &= r_x + r_y = \sum_{j=1}^{v} c_j \end{aligned}$$

The chi-squared measure is computed as follows:

$d_{24}(x, y) = (\sum_{j=1}^{v} \frac{(w_j x_j - E(x_j))^2}{E(x_j)} + \sum_{j=1}^{v} \frac{(w_j y_j - E(y_j))^2}{E(y_j)}) W / (\sum_{j=1}^{v} w_j)$

where for $j = 1, 2, \ldots, v$

$$\begin{aligned} E(x_j) &= r_x c_j / T \\ E(y_j) &= r_y c_j / T \end{aligned}$$

CHI — squared root of chi-squared
$d_{25}(x, y) = \sqrt{d_{23}(x, y)}$

PHISQ — phi-squared
This is the CHISQ dissimilarity normalized by the sum of weights
$d_{26}(x, y) = d_{24}(x, y)/(\sum_{j=1}^{v} w_j)$

PHI — squared root of phi-squared
$d_{27}(x, y) = \sqrt{d_{25}(x, y)}$

Methods Accepting Symmetric Nominal Variables

The following notation is used for computing $d_{28}(x, y)$ to $s_{35}(x, y)$. Notice that only the nonmissing pairs are discussed below; all the pairs with at least one missing value will be excluded from any of the computations in the following section because $w_j = 0$, if either x_j or y_j is missing.

M — nonmissing matches
$M = \sum_{j=1}^{v} w_j \delta_{x,y}^{j}$, where

$$\begin{aligned} \delta_{x,y}^{j} &= 1, \text{ if } x_j = y_j \\ \delta_{x,y}^{j} &= 0, \text{ otherwise} \end{aligned}$$

X — nonmissing mismatches
$X = \sum_{j=1}^{v} w_j \delta_{x,y}^{j}$, where

$$\begin{aligned} \delta_{x,y}^{j} &= 1, \text{ if } x_j \neq y_j \\ \delta_{x,y}^{j} &= 0, \text{ otherwise} \end{aligned}$$

N — total nonmissing pairs
$N = \sum_{j=1}^{v} w_j$

HAMMING — Hamming distance
$d_{28}(x, y) = X$

MATCH — simple matching coefficient
$s_{29}(x, y) = M/N$

DMATCH — simple matching coefficient transformed to Euclidean distance
$d_{30}(x, y) = \sqrt{1 - M/N} = \sqrt{(X/N)}$

DSQMATCH — simple matching coefficient transformed to squared Euclidean distanc
$d_{31}(x, y) = 1 - M/N = X/N$

HAMANN — Hamann coefficient
$s_{32}(x, y) = (M - X)/N$

RT — Roger and Tanimoto
$s_{33}(x, y) = M/(M + 2X)$

SS1 — Sokal and Sneath 1
$s_{34}(x, y) = 2M/(2M + X)$

SS3 — Sokal and Sneath 3. The coefficient between an observation and itself is always indeterminate (missing) since there is no mismatch.
$s_{35}(x, y) = M/X$

The following notation is used for computing $s_{36}(x, y)$ to $d_{41}(x, y)$. Notice that only the nonmissing pairs are discussed in the following section; all the pairs with at least one missing value are excluded from any of the computations in the following section because $w_j = 0$, if either x_j or y_j is missing.

Also, the observed nonmissing data of an asymmetric binary variable can have only two possible outcomes: presence or absence. Therefore, the notation, PX (present mismatches), always has a value of zero for an asymmetric binary variable.

The following methods distinguish between the presence and absence of attributes.

X — mismatches with at least one present
$X = \sum_{j=1}^{v} w_j \delta_{x,y}^j$, where

$$\begin{aligned} \delta_{x,y}^j &= 1, \text{ if } x_j \neq y_j \text{ and not both } x_j \text{ and } y_j \text{ are absent} \\ \delta_{x,y}^j &= 0, \text{ otherwise} \end{aligned}$$

PM — present matches
$PM = \sum_{j=1}^{v} w_j \delta_{x,y}^j$, where

$$\begin{aligned} \delta_{x,y}^j &= 1, \text{ if } x_j = y_j \text{ and both } x_j \text{ and } y_j \text{ are present} \\ \delta_{x,y}^j &= 0, \text{ otherwise} \end{aligned}$$

PX — present mismatches
$PX = \sum_{j=1}^{v} w_j \delta_{x,y}^j$, where

$$\begin{aligned} \delta_{x,y}^j &= 1, \text{ if } x_j \neq y_j \text{ and both } x_j \text{ and } y_j \text{ are present} \\ \delta_{x,y}^j &= 0, \text{ otherwise} \end{aligned}$$

PP — both present = $PM + PX$

P — at least one present = $PM + X$

PAX present-absent mismatches

$PAX = \sum_{j=1}^{v} w_j \delta_{x,y}^j$, where

$$\begin{aligned} \delta_{x,y}^j &= 1, \text{ if } x_j \neq y_j \text{ and either } x_j \text{ is present and } y_j \text{ is absent or} \\ &\quad\ x_j \text{ is absent and } y_j \text{ is present} \\ \delta_{x,y}^j &= 0 \text{ otherwise} \end{aligned}$$

N total nonmissing pairs

$N = \sum_{j=1}^{v} w_j$

Methods Accepting Asymmetric Nominal and Ratio Variables

JACCARD Jaccard similarity coefficient

The JACCARD method is equivalent to the SIMRATIO method if there are only ratio variables; if there are both ratio and asymmetric nominal variables, the coefficient is computed as sum of the coefficient from the ratio variables (SIMRATIO) and the coefficient from the asymmetric nominal variables.

$s_{36}(x, y) = s_{16}(x, y) + PM/P$

DJACCARD Jaccard dissimilarity coefficient

The DJACCARD method is equivalent to the DISRATIO method if there are only ratio variables; if there are both ratio and asymmetric nominal variables, the coefficient is computed as sum of the coefficient from the ratio variables (DISRATIO) and the coefficient from the asymmetric nominal variables.

$d_{37}(x, y) = d_{17}(x, y) + X/P$

Methods Accepting Asymmetric Nominal Variables

DICE Dice coefficient or Czekanowski/Sorensen similarity coefficient

$s_{38}(x, y) = 2PM/(P + PM)$

RR Russell and Rao. This is the binary equivalent of the dot product coefficient.

$s_{39}(x, y) = PM/N$

BLWNM

BRAYCURTIS Binary Lance and Williams, also known as Bray and Curtis coefficient

$d_{40}(x, y) = X/(PAX + 2PP)$

K1 Kulcynski 1. The coefficient between an observation and itself is always indeterminate (missing) since there is no mismatch.
$d_{41}(x, y) = PM/X$

Missing Values

Standardization versus No Standardization

You can replace the missing values with or without standardization. Missing values are replaced after standardization by specifying either the REPLACE option in the PROC DISTANCE statement or the MISSING= option in the VAR statement.

To replace missing values without standardization, use the following two options:

- the NOSTD option in the PROC DISTANCE statement. The NOSTD option suppresses standardization but still replaces the missing values with the location of the method or the numeric value specified in the MISSING= option in the VAR statement.
- the REPONLY option in the PROC DISTANCE statement. PROC DISTANCE replaces missing values with the location of the standardization method or with the numeric value specified in the MISSING= option in the VAR statement. This approach assumes that standardization is not mandatory (see the section "Standardization" on page 1485).

Eliminating Observations with Missing Values

If you specify the NOMISS option, PROC DISTANCE omits observations with any missing values in the analyzed variables from computation of the location and scale measures.

Distance Measures

If you specify the NOMISS option, PROC DISTANCE generates missing distance for observations with missing values. If the NOMISS option is not specified, the sum of total weights, no matter if an observation is missing or not, is incorporated into the computation of some of the proximity measures. See the section "Details: DISTANCE Procedure" on page 1507 for the formulas and descriptions.

Formatted versus Unformatted Values

PROC DISTANCE uses the formatted values from a character variable, if the variable has a format—for example, one assigned by a format statement. PROC DISTANCE uses the unformatted values from a numeric variable, even if it has a format.

Output Data Sets

OUT= Data Set

The DISTANCE procedure always produces an output data set, regardless of whether you specify the OUT= option in the PROC DISTANCE statement. PROC DISTANCE displays no output. Use PROC PRINT, PROC REPORT, or some other SAS reporting tool to print the output data set.

The output data set contains the following variables:

- the ID variable, if any
- the BY variables, if any
- the COPY variables, if any
- the FREQ variable, if any
- the WEIGHT variable, if any
- the new distance variables, named from PREFIX= options along with the ID values, or from the default values

OUTSDZ= Data Set

The output data set is a copy of the DATA= data set except that the analyzed variables have been standardized. Analyzed variables are those listed in the VAR statement.

Examples: DISTANCE Procedure

Example 32.1: Divorce Grounds – the Jaccard Coefficient

A wide variety of distance and similarity measures are used in cluster analysis (Anderberg 1973, Sneath and Sokal 1973). If your data are in coordinate form and you want to use a non-Euclidean distance for clustering, you can compute a distance matrix by using the DISTANCE procedure.

Similarity measures must be converted to dissimilarities before being used in PROC CLUSTER. Such conversion can be done in a variety of ways, such as taking reciprocals or subtracting from a large value. The choice of conversion method depends on the application and the similarity measure. If applicable, PROC DISTANCE provides a corresponding dissimilarity measure for each similarity measure.

In the following example, the observations are states. Binary-valued variables correspond to various grounds for divorce and indicate whether the grounds for divorce apply in each of the U.S. states. A value of "1" indicates that the ground for divorce applies, and a value of "0" indicates the opposite. The 0-0 matches are treated totally irrelevant; therefore, each variable has an asymmetric nominal level of measurement. The absence value is 0.

The DISTANCE procedure is used to compute the Jaccard coefficient (Anderberg 1973, pp. 89, 115, and 117) between each pair of states. The Jaccard coefficient is defined as the number of variables that are coded as 1 for both states divided by the number of variables that are coded as 1 for either or both states. Since dissimilarity measures are required by the CLUSTER procedure, the DJACCARD coefficient is selected. Output 32.1.1 displays the distance matrix between the first 10 states.

The CENTROID method is used to perform the cluster analysis, and the resulting tree diagram from PROC CLUSTER is saved into the tree output data set. Output 32.1.2 displays the cluster history.

The TREE procedure generates nine clusters in the output data set out. After being sorted by the state, the out data set is then merged with the input data set divorce. After being sorted by the state, the merged data set is printed to display the cluster membership as shown in Output 32.1.3.

The following statements produce Output 32.1.1 through Output 32.1.3:

```
data divorce;
   input state $15.
         (incompat cruelty desertn non_supp alcohol
          felony impotenc insanity separate) (1.) @@;
   if mod(_n_,2) then input +4 @@; else input;
   datalines;
Alabama         111111111    Alaska          111011110
Arizona         100000000    Arkansas        011111111
California      100000010    Colorado        100000000
Connecticut     111111011    Delaware        100000001
Florida         100000010    Georgia         111011110
Hawaii          100000001    Idaho           111111011
Illinois        011011100    Indiana         100001110
Iowa            100000000    Kansas          111011110
Kentucky        100000000    Louisiana       000001001
Maine           111110110    Maryland        011001111
Massachusetts   111111101    Michigan        100000000
Minnesota       100000000    Mississippi     111011110
Missouri        100000000    Montana         100000000
Nebraska        100000000    Nevada          100000011
New Hampshire   111111100    New Jersey      011011011
New Mexico      111000000    New York        011001001
North Carolina  000000111    North Dakota    111111110
Ohio            111011101    Oklahoma        111111110
Oregon          100000000    Pennsylvania    011001110
Rhode Island    111111101    South Carolina  011010001
South Dakota    011111000    Tennessee       111111100
Texas           111001011    Utah            011111110
Vermont         011101011    Virginia        010001001
Washington      100000001    West Virginia   111011011
Wisconsin       100000001    Wyoming         100000011
;
```

```
title 'Grounds for Divorce';
proc distance data=divorce method=djaccard absent=0 out=distjacc;
   var anominal(incompat--separate);
   id state;
run;
proc print data=distjacc(obs=10);
   id state; var alabama--georgia;
   title2 'First 10 states';
run;
title2;
proc cluster data=distjacc method=centroid
             pseudo outtree=tree;
   id state;
   var alabama--wyoming;
run;

proc tree data=tree noprint n=9 out=out;
   id state;
run;

proc sort;
   by state;
run;

data clus;
   merge divorce out;
   by state;
run;

proc sort;
   by cluster;
run;
proc print;
   id state;
   var incompat--separate;
   by cluster;
run;
```

Output 32.1.1 Distance Matrix Based on the Jaccard Coefficient

Grounds for Divorce
First 10 states

state	Alabama	Alaska	Arizona	Arkansas	California	Colorado
Alabama	0.00000	.	.	.	.	.
Alaska	0.22222	0.00000	.	.	.	.
Arizona	0.88889	0.85714	0.00000	.	.	.
Arkansas	0.11111	0.33333	1.00000	0.00000	.	.
California	0.77778	0.71429	0.50000	0.88889	0.00000	.
Colorado	0.88889	0.85714	0.00000	1.00000	0.50000	0.00000
Connecticut	0.11111	0.33333	0.87500	0.22222	0.75000	0.87500
Delaware	0.77778	0.87500	0.50000	0.88889	0.66667	0.50000
Florida	0.77778	0.71429	0.50000	0.88889	0.00000	0.50000
Georgia	0.22222	0.00000	0.85714	0.33333	0.71429	0.85714

state	Connecticut	Delaware	Florida	Georgia
Alabama	.	.	.	.
Alaska	.	.	.	.
Arizona	.	.	.	.
Arkansas	.	.	.	.
California	.	.	.	.
Colorado	.	.	.	.
Connecticut	0.00000	.	.	.
Delaware	0.75000	0.00000	.	.
Florida	0.75000	0.66667	0.00000	.
Georgia	0.33333	0.87500	0.71429	0

Output 32.1.2 Clustering History

Grounds for Divorce

The CLUSTER Procedure
Centroid Hierarchical Cluster Analysis

Root-Mean-Square Distance Between Observations 0.694873

Output 32.1.2 *continued*

Cluster History

NCL	Clusters Joined		FREQ	PSF	PST2	Norm Cent Dist	Tie
49	Arizona	Colorado	2	.	.	0	T
48	California	Florida	2	.	.	0	T
47	Alaska	Georgia	2	.	.	0	T
46	Delaware	Hawaii	2	.	.	0	T
45	Connecticut	Idaho	2	.	.	0	T
44	CL49	Iowa	3	.	.	0	T
43	CL47	Kansas	3	.	.	0	T
42	CL44	Kentucky	4	.	.	0	T
41	CL42	Michigan	5	.	.	0	T
40	CL41	Minnesota	6	.	.	0	T
39	CL43	Mississippi	4	.	.	0	T
38	CL40	Missouri	7	.	.	0	T
37	CL38	Montana	8	.	.	0	T
36	CL37	Nebraska	9	.	.	0	T
35	North Dakota	Oklahoma	2	.	.	0	T
34	CL36	Oregon	10	.	.	0	T
33	Massachusetts	Rhode Island	2	.	.	0	T
32	New Hampshire	Tennessee	2	.	.	0	T
31	CL46	Washington	3	.	.	0	T
30	CL31	Wisconsin	4	.	.	0	T
29	Nevada	Wyoming	2	.	.	0	
28	Alabama	Arkansas	2	1561	.	0.1599	T
27	CL33	CL32	4	479	.	0.1799	T
26	CL39	CL35	6	265	.	0.1799	T
25	CL45	West Virginia	3	231	.	0.1799	
24	Maryland	Pennsylvania	2	199	.	0.2399	
23	CL28	Utah	3	167	3.2	0.2468	
22	CL27	Ohio	5	136	5.4	0.2698	
21	CL26	Maine	7	111	8.9	0.2998	
20	CL23	CL21	10	75.2	8.7	0.3004	
19	CL25	New Jersey	4	71.8	6.5	0.3053	T
18	CL19	Texas	5	69.1	2.5	0.3077	
17	CL20	CL22	15	48.7	9.9	0.3219	
16	New York	Virginia	2	50.1	.	0.3598	
15	CL18	Vermont	6	49.4	2.9	0.3797	
14	CL17	Illinois	16	47.0	3.2	0.4425	
13	CL14	CL15	22	29.2	15.3	0.4722	
12	CL48	CL29	4	29.5	.	0.4797	T
11	CL13	CL24	24	27.6	4.5	0.5042	
10	CL11	South Dakota	25	28.4	2.4	0.5449	
9	Louisiana	CL16	3	30.3	3.5	0.5844	
8	CL34	CL30	14	23.3	.	0.7196	
7	CL8	CL12	18	19.3	15.0	0.7175	
6	CL10	South Carolina	26	21.4	4.2	0.7384	
5	CL6	New Mexico	27	24.0	4.7	0.8303	
4	CL5	Indiana	28	28.9	4.1	0.8343	
3	CL4	CL9	31	31.7	10.9	0.8472	
2	CL3	North Carolina	32	55.1	4.1	1.0017	
1	CL2	CL7	50	.	55.1	1.0663	

Output 32.1.3 Cluster Membership

Grounds for Divorce

CLUSTER=1

state	incompat	cruelty	desertn	non_supp	alcohol	felony	impotenc	insanity	separate
Arizona	1	0	0	0	0	0	0	0	0
Colorado	1	0	0	0	0	0	0	0	0
Iowa	1	0	0	0	0	0	0	0	0
Kentucky	1	0	0	0	0	0	0	0	0
Michigan	1	0	0	0	0	0	0	0	0
Minnesota	1	0	0	0	0	0	0	0	0
Missouri	1	0	0	0	0	0	0	0	0
Montana	1	0	0	0	0	0	0	0	0
Nebraska	1	0	0	0	0	0	0	0	0
Oregon	1	0	0	0	0	0	0	0	0

CLUSTER=2

state	incompat	cruelty	desertn	non_supp	alcohol	felony	impotenc	insanity	separate
California	1	0	0	0	0	0	0	1	0
Florida	1	0	0	0	0	0	0	1	0
Nevada	1	0	0	0	0	0	0	1	1
Wyoming	1	0	0	0	0	0	0	1	1

Output 32.1.3 *continued*

Grounds for Divorce

CLUSTER=3

State	incompat	cruelty	desertn	non_support	alcohol	felony	impotency	insanity	separate
Alabama	1	1	1	1	1	1	1	1	1
Alaska	1	1	1	0	1	1	1	1	0
Arkansas	0	1	1	1	1	1	1	1	1
Connecticut	1	1	1	1	1	1	0	1	1
Georgia	1	1	1	0	1	1	1	1	0
Idaho	1	1	1	1	1	1	0	1	1
Illinois	0	1	1	0	1	1	1	0	0
Kansas	1	1	1	0	1	1	1	1	0
Maine	1	1	1	1	1	0	1	1	0
Maryland	0	1	1	0	0	1	1	1	1
Massachusetts	1	1	1	1	1	1	1	0	1
Mississippi	1	1	1	0	1	1	1	1	0
New Hampshire	1	1	1	1	1	1	1	0	0
New Jersey	0	1	1	0	1	1	0	1	1
North Dakota	1	1	1	1	1	1	1	1	0
Ohio	1	1	1	0	1	1	1	0	1
Oklahoma	1	1	1	1	1	1	1	1	0
Pennsylvania	0	1	1	0	0	1	1	1	0
Rhode Island	1	1	1	1	1	1	1	0	1
South Dakota	0	1	1	1	1	1	0	0	0
Tennessee	1	1	1	1	1	1	1	0	0
Texas	1	1	1	0	0	1	0	1	1
Utah	0	1	1	1	1	1	1	1	0
Vermont	0	1	1	1	0	1	0	1	1
West Virginia	1	1	1	0	1	1	0	1	1

Output 32.1.3 *continued*

Grounds for Divorce

CLUSTER=4

State	incompat	cruelty	desertn	nonsupp	alcohol	felony	impotnc	insanity	separate
Delaware	1	0	0	0	0	0	0	0	1
Hawaii	1	0	0	0	0	0	0	0	1
Washington	1	0	0	0	0	0	0	0	1
Wisconsin	1	0	0	0	0	0	0	0	1

CLUSTER=5

State	incompat	cruelty	desertn	nonsupp	alcohol	felony	impotnc	insanity	separate
Louisiana	0	0	0	0	0	1	0	0	1
New York	0	1	1	0	0	1	0	0	1
Virginia	0	1	0	0	0	1	0	0	1

CLUSTER=6

State	incompat	cruelty	desertn	nonsupp	alcohol	felony	impotnc	insanity	separate
South Carolina	0	1	1	0	1	0	0	0	1

Output 32.1.3 *continued*

```
                          Grounds for Divorce

---------------------------------- CLUSTER=7 ----------------------------------

                        i             n             i    i    s
                        n    c    d    o    a        m    n    e
                        c    r    e    n    l    f   p    s    p
              s         o    u    s    _    c    e   o    a    a
              t         m    e    e    s    o    l   t    n    r
              a         p    l    r    u    h    o   e    i    a
              t         a    t    t    p    o    n   n    t    t
              e         t    y    n    p    l    y   c    y    e

          New Mexico    1    1    1    0    0    0   0    0    0


---------------------------------- CLUSTER=8 ----------------------------------

                       i             n             i    i    s
                       n    c    d    o    a        m    n    e
                       c    r    e    n    l    f   p    s    p
              s        o    u    s    _    c    e   o    a    a
              t        m    e    e    s    o    l   t    n    r
              a        p    l    r    u    h    o   e    i    a
              t        a    t    t    p    o    n   n    t    t
              e        t    y    n    p    l    y   c    y    e

            Indiana    1    0    0    0    0    1   1    1    0


---------------------------------- CLUSTER=9 ----------------------------------

                          i             n             i    i    s
                          n    c    d    o    a        m    n    e
                          c    r    e    n    l    f   p    s    p
              s           o    u    s    _    c    e   o    a    a
              t           m    e    e    s    o    l   t    n    r
              a           p    l    r    u    h    o   e    i    a
              t           a    t    t    p    o    n   n    t    t
              e           t    y    n    p    l    y   c    y    e

        North Carolina    0    0    0    0    0    0   1    1    1
```

Example 32.2: Financial Data – Stock Dividends

The following data set contains the average dividend yields for 15 utility stocks in the United States. The observations are names of the companies, and the variables correspond to the annual dividend yields for the period 1986–1990. The objective is to group similar stocks into clusters.

Before the cluster analysis is performed, the correlation similarity is chosen for measuring the closeness between each observation. Since distance type of measures are required by the CLUSTER

procedure, METHOD=DCORR is used in the PROC DISTANCE statement to transform the correlation measures to the distance measures. Notice that in Output 32.2.1, all the values in the distance matrix are between 0 and 2.

The macro function DO_CLUSTER performs cluster analysis and presents the results in graphs. The CLUSTER procedure performs hierarchical clustering by using agglomerative methods based on the distance data created from the previous PROC DISTANCE statement. The resulting tree diagrams can be saved into an output data set and can later be plotted by the TREE procedure. Since the CCC statistic is not suitable for distance type of data, only the Pseudo statistic is requested to identify the number of clusters.

Two clustering methods are invoked in the DO_CLUSTER macro: Ward's and the average linkage methods. Since the results of the Pseudo t^2 statistic from both Ward's and the average linkage methods contain many missing values, only the plot of the Pseudo F statistic versus the number of clusters is requested by specifying PLOTS(ONLY)= PSF in the PROC CLUSTER statement.

Both Output 32.2.2 and Output 32.2.3 suggest a possible clusters of 4. Both methods produce the same clustering result, as shown in Output 32.2.4 and Output 32.2.5. The four clusters are as follows:

- Cincinnati G&E and Detroit Edison
- Texas Utilities and Pennsylvania Power & Light
- Union Electric, Iowa-Ill Gas & Electric, Oklahoma Gas & Electric, and Wisconsin Energy
- Orange & Rockland Utilities, Kentucky Utilities, Kansas Power & Light, Allegheny Power, Green Mountain Power, Dominion Resources, and Minnesota Power & Light

```
   data stock;
      title 'Stock Dividends';
      input compname &$26.  div_1986 div_1987 div_1988
                            div_1989 div_1990;
   datalines;
Cincinnati G&E                8.4    8.2    8.4    8.1    8.0
Texas Utilities               7.9    8.9   10.4    8.9    8.3
Detroit Edison                9.7   10.7   11.4    7.8    6.5
Orange & Rockland Utilities   6.5    7.2    7.3    7.7    7.9
Kentucky Utilities            6.5    6.9    7.0    7.2    7.5
Kansas Power & Light          5.9    6.4    6.9    7.4    8.0
Union Electric                7.1    7.5    8.4    7.8    7.7
Dominion Resources            6.7    6.9    7.0    7.0    7.4
Allegheny Power               6.7    7.3    7.8    7.9    8.3
Minnesota Power & Light       5.6    6.1    7.2    7.0    7.5
Iowa-Ill Gas & Electric       7.1    7.5    8.5    7.8    8.0
Pennsylvania Power & Light    7.2    7.6    7.7    7.4    7.1
Oklahoma Gas & Electric       6.1    6.7    7.4    6.7    6.8
Wisconsin Energy              5.1    5.7    6.0    5.7    5.9
Green Mountain Power          7.1    7.4    7.8    7.8    8.3
;
proc distance data=stock method=dcorr out=distdcorr;
   var interval(div_1986 div_1987 div_1988 div_1989 div_1990);
   id compname;
run;
```

```
proc print data=distdcorr;
   id compname;
   title2 'Distance Matrix for 15 Utility Stocks';
run;
title2;

/* performs cluster analysis and plots the results */
%macro do_cluster(clusmtd);

%let clusmtd = %upcase(&clusmtd);
title2 "Cluster Method= &clusmtd";

/* compute pseudo statistic versus number of cluster and get plot */
proc cluster data=distdcorr method=&clusmtd outtree=Tree
             pseudo plots(only)= psf;
   id compname;
run;

/* plot tree diagram */
proc tree data=Tree horizontal;
   id compname;
run;
%mend;
ods graphics on;

/* METHOD=WARD */
%do_cluster(ward);
/* METHOD=AVERAGE */
%do_cluster(average);

ods graphics off;
```

Output 32.2.1 Distance Matrix Based on the DCORR Coefficient

Stock Dividends
Distance Matrix for 15 Utility Stocks

compname	Cincinnati_ G_E	Texas_ Utilities	Detroit_ Edison	Orange___ Rockland_ Utilitie
Cincinnati G&E	0.00000	.	.	.
Texas Utilities	0.82056	0.00000	.	.
Detroit Edison	0.40511	0.65453	0.00000	.
Orange & Rockland Utilitie	1.35380	0.88583	1.27306	0.00000
Kentucky Utilities	1.35581	0.92539	1.29382	0.12268
Kansas Power & Light	1.34227	0.94371	1.31696	0.19905
Union Electric	0.98516	0.29043	0.89048	0.68798

compname	Kentucky_ Utilities	Kansas_ Power___ Light	Union_ Electric	Dominion_ Resources	Allegheny_ Power
Cincinnati G&E	.	.	.	.	.
Texas Utilities	.	.	.	.	.
Detroit Edison	.	.	.	.	.
Orange & Rockland Utilitie	.	.	.	.	.
Kentucky Utilities	0.00000	.	.	.	.
Kansas Power & Light	0.12874	0.00000	.	.	.
Union Electric	0.71824	0.72082	0.00000	.	.

compname	Minnesota_ Power___ Light	Iowa_Ill_ Gas___ Electric	Pennsylvania_ Power___Light	Oklahoma_ Gas___ Electric
Cincinnati G&E	.	.	.	.
Texas Utilities	.	.	.	.
Detroit Edison	.	.	.	.
Orange & Rockland Utilitie	.	.	.	.
Kentucky Utilities	.	.	.	.
Kansas Power & Light	.	.	.	.
Union Electric	.	.	.	.

compname	Wisconsin_ Energy	Green_ Mountain_ Power
Cincinnati G&E	.	.
Texas Utilities	.	.
Detroit Edison	.	.
Orange & Rockland Utilitie	.	.
Kentucky Utilities	.	.
Kansas Power & Light	.	.
Union Electric	.	.

Output 32.2.1 *continued*

Stock Dividends
Distance Matrix for 15 Utility Stocks

compname	Cincinnati_ G_E	Texas_ Utilities	Detroit_ Edison	Orange___ Rockland_ Utilitie
Dominion Resources	1.32945	0.96853	1.29016	0.33290
Allegheny Power	1.30492	0.81666	1.24565	0.17844
Minnesota Power & Light	1.24069	0.74082	1.20432	0.32581
Iowa-Ill Gas & Electric	1.04924	0.43100	0.97616	0.61166
Pennsylvania Power & Light	0.74931	0.37821	0.44256	1.03566
Oklahoma Gas & Electric	1.00604	0.30141	0.86200	0.68021
Wisconsin Energy	1.17988	0.54830	1.03081	0.45013

compname	Kentucky_ Utilities	Kansas_ Power___ Light	Union_ Electric	Dominion_ Resources	Allegheny_ Power
Dominion Resources	0.21510	0.24189	0.76587	0.00000	.
Allegheny Power	0.15759	0.17029	0.58452	0.27819	0.00000
Minnesota Power & Light	0.30462	0.27231	0.48372	0.35733	0.15615
Iowa-Ill Gas & Electric	0.61760	0.61736	0.16923	0.63545	0.47900
Pennsylvania Power & Light	1.08878	1.12876	0.63285	1.14354	1.02358
Oklahoma Gas & Electric	0.70259	0.73158	0.17122	0.72977	0.58391
Wisconsin Energy	0.47184	0.53381	0.37405	0.51969	0.37522

compname	Minnesota_ Power___ Light	Iowa_Ill_ Gas___ Electric	Pennsylvania_ Power___Light	Oklahoma_ Gas___ Electric
Dominion Resources	.	.	.	.
Allegheny Power	.	.	.	.
Minnesota Power & Light	0.00000	.	.	.
Iowa-Ill Gas & Electric	0.36368	0.00000	.	.
Pennsylvania Power & Light	0.99384	0.75596	0.00000	.
Oklahoma Gas & Electric	0.50744	0.19673	0.60216	0.00000
Wisconsin Energy	0.36319	0.30259	0.76085	0.28070

compname	Wisconsin_ Energy	Green_ Mountain_ Power
Dominion Resources	.	.
Allegheny Power	.	.
Minnesota Power & Light	.	.
Iowa-Ill Gas & Electric	.	.
Pennsylvania Power & Light	.	.
Oklahoma Gas & Electric	.	.
Wisconsin Energy	0.00000	.

Output 32.2.1 *continued*

Stock Dividends
Distance Matrix for 15 Utility Stocks

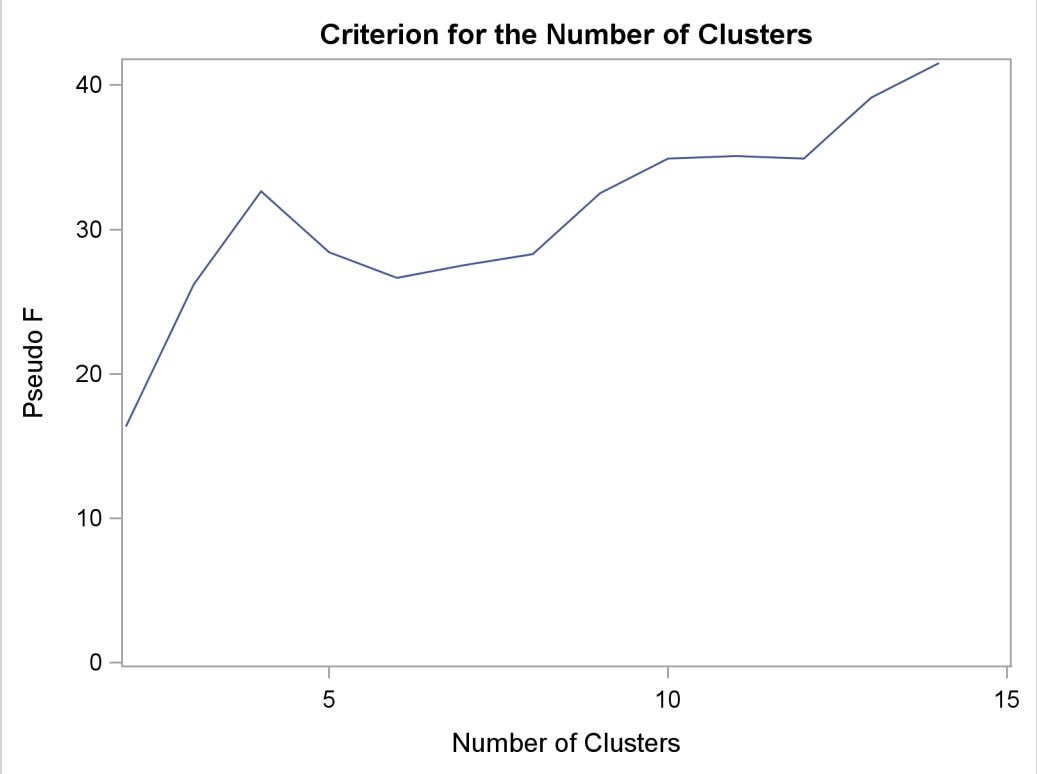
compname	Cincinnati_ G_E	Texas_ Utilities	Detroit_ Edison	Orange___ Rockland_ Utilitie
Green Mountain Power	1.30397	0.88063	1.27176	0.26948

compname	Kentucky_ Utilities	Kansas_ Power___ Light	Union_ Electric	Dominion_ Resources	Allegheny_ Power
Green Mountain Power	0.17909	0.15377	0.64869	0.17360	0.13958

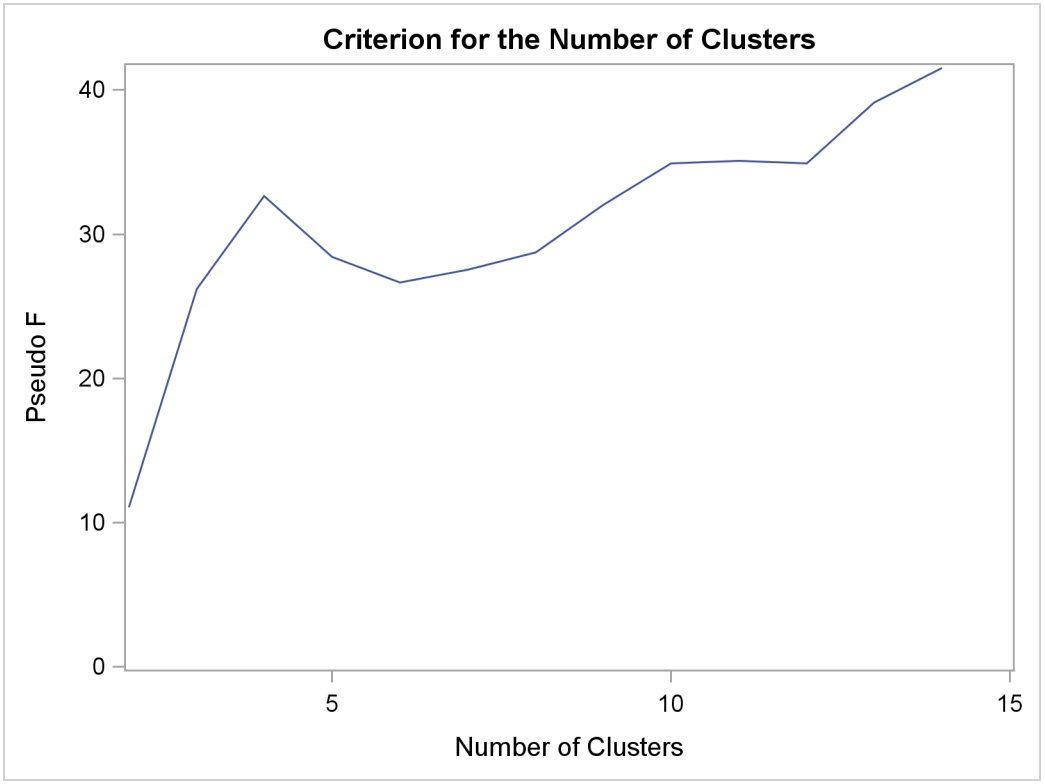
compname	Minnesota_ Power___ Light	Iowa_Ill_ Gas___ Electric	Pennsylvania_ Power___Light	Oklahoma_ Gas___ Electric
Green Mountain Power	0.19370	0.52083	1.09269	0.64175

compname	Wisconsin_ Energy	Green_ Mountain_ Power
Green Mountain Power	0.44814	0

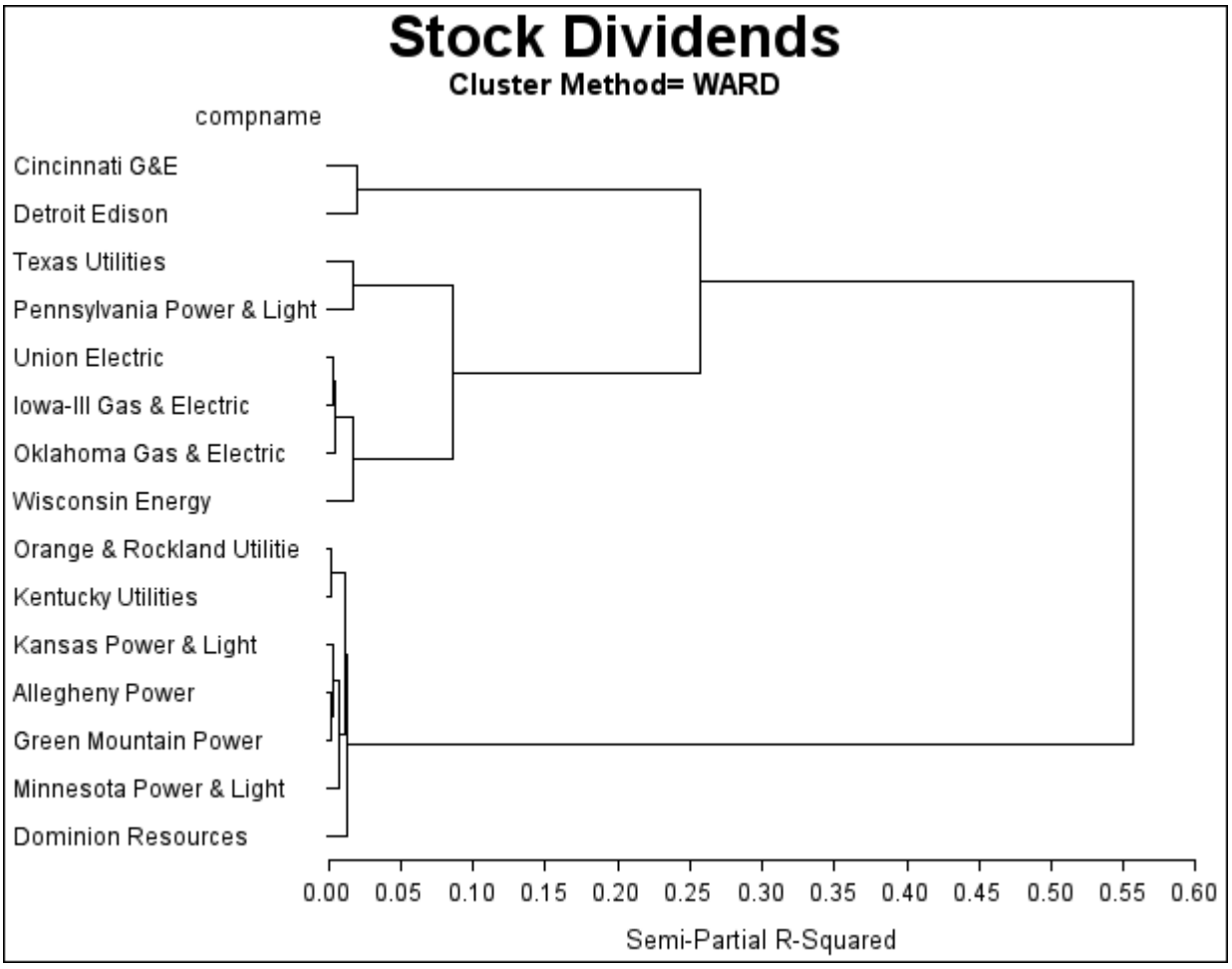
Output 32.2.2 Pseudo *F* versus Number of Clusters When METHOD=WARD



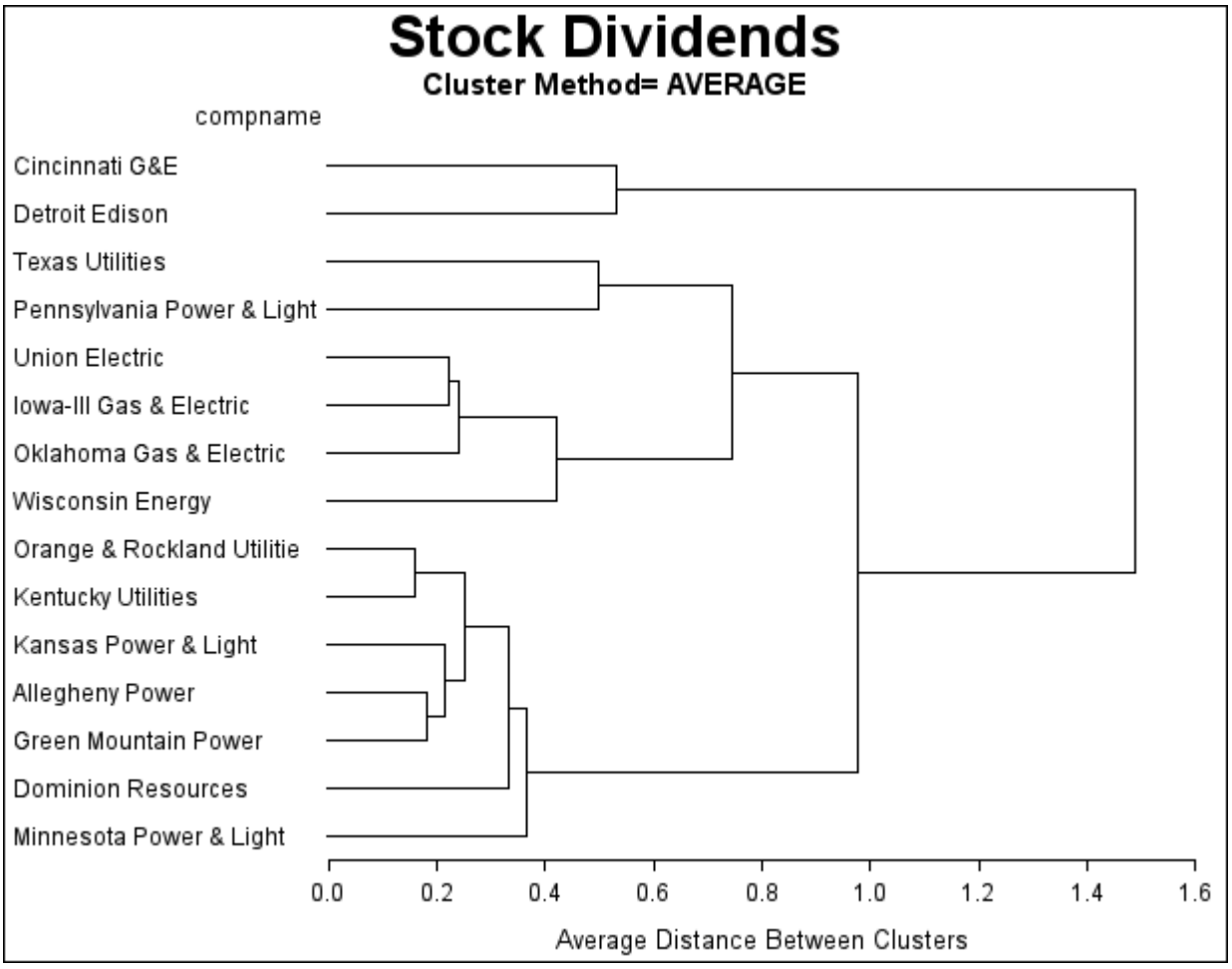
Output 32.2.3 Pseudo *F* versus Number of Clusters When METHOD=AVERAGE



Output 32.2.4 Tree Diagram of Clusters versus Semipartial R-Square Values When METHOD=WARD



Output 32.2.5 Tree Diagram of Clusters versus Average Distance between Clusters When METHOD=AVERAGE



References

Anderberg, M. R. (1973), *Cluster Analysis for Applications*, New York: Academic Press.

Hand, D. J., Daly, F., Lunn, A. D., McConway, K.J., and Ostrowski, E. (1994), *A Handbook of Small Data Sets*, London: Chapman and Hall.

Gower, J. C., and Legendre, P. (1986), "Metric and Euclidean Properties of Dissimilarity Coefficients," *Journal of Classification*, 3, 5–48.

Legendre, L. and Legendre, P. (1983), *Numerical Ecology*, New York: Elsevier Scientific.

Kaufman, L. and Rousseeuw, P. J. (1990), *Finding Groups in Data*, New York: John Wiley & Sons.

Sneath, P. H. A. and Sokal, R. R. (1973), *Numerical Taxonomy*, San Francisco: Freeman.

Chapter 33

The FACTOR Procedure

Contents

Overview: FACTOR Procedure **1534**
Background 1534
Outline of Use 1537
Getting Started: FACTOR Procedure **1542**
Syntax: FACTOR Procedure **1548**
PROC FACTOR Statement 1548
BY Statement 1564
FREQ Statement 1564
PARTIAL Statement 1565
PRIORS Statement 1565
VAR Statement 1565
WEIGHT Statement 1566
Details: FACTOR Procedure **1566**
Input Data Set 1566
Output Data Sets 1568
Confidence Intervals and the Salience of Factor Loadings 1571
Simplicity Functions for Rotations 1572
Missing Values 1573
Cautions 1574
Factor Scores 1574
Variable Weights and Variance Explained 1575
Heywood Cases and Other Anomalies about Communality Estimates 1576
Time Requirements 1578
Displayed Output 1579
ODS Table Names 1583
ODS Graphics 1585
Examples: FACTOR Procedure **1586**
Example 33.1: Principal Component Analysis 1586
Example 33.2: Principal Factor Analysis 1588
Example 33.3: Maximum Likelihood Factor Analysis 1604
Example 33.4: Using Confidence Intervals to Locate Salient Factor Loadings 1612
References **1617**

Overview: FACTOR Procedure

The FACTOR procedure performs a variety of common factor and component analyses and rotations. Input can be multivariate data, a correlation matrix, a covariance matrix, a factor pattern, or a matrix of scoring coefficients. The procedure can factor either the correlation or covariance matrix, and you can save most results in an output data set.

PROC FACTOR can process output from other procedures. For example, it can rotate the canonical coefficients from multivariate analyses in the GLM procedure.

The methods for factor extraction are principal component analysis, principal factor analysis, iterated principal factor analysis, unweighted least squares factor analysis, maximum likelihood (canonical) factor analysis, alpha factor analysis, image component analysis, and Harris component analysis. A variety of methods for prior communality estimation is also available.

Specific methods for orthogonal rotation are varimax, quartimax, biquartimax, equamax, parsimax, and factor parsimax. Oblique versions of these methods are also available. In addition, quartimin, biquartimin, and covarimin methods for (direct) oblique rotation are available. General methods for orthogonal rotation are orthomax with user-specified gamma, Crawford-Ferguson family with user-specified weights on variable parsimony and factor parsimony, and generalized Crawford-Ferguson family with user-specified weights. General methods for oblique rotation are direct oblimin with user-specified tau, Crawford-Ferguson family with user-specified weights on variable parsimony and factor parsimony, generalized Crawford-Ferguson family with user-specified weights, promax with user-specified exponent, Harris-Kaiser case II with user-specified exponent, and Procrustes with a user-specified target pattern.

Output includes means, standard deviations, correlations, Kaiser's measure of sampling adequacy, eigenvalues, a scree plot, eigenvectors, prior and final communality estimates, the unrotated factor pattern, residual and partial correlations, the rotated primary factor pattern, the primary factor structure, interfactor correlations, the reference structure, reference axis correlations, the variance explained by each factor both ignoring and eliminating other factors, plots of both rotated and unrotated factors, squared multiple correlation of each factor with the variables, standard error estimates, confidence limits, coverage displays, and scoring coefficients.

The FACTOR procedure now uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

Any topics that are not given explicit references are discussed in Mulaik (1972) or Harman (1976).

Background

See Chapter 69, "The PRINCOMP Procedure," for a discussion of principal component analysis. See Chapter 25, "The CALIS Procedure," for a discussion of confirmatory factor analysis.

Common factor analysis was invented by Spearman (1904). Kim and Mueller (1978a, 1978b) provide a very elementary discussion of the common factor model. Gorsuch (1974) presents a broad

survey of factor analysis, and Gorsuch (1974) and Cattell (1978) are useful as guides to practical research methodology. Harman (1976) gives a lucid discussion of many of the more technical aspects of factor analysis, especially oblique rotation. Morrison (1976) and Mardia, Kent, and Bibby (1979) provide excellent statistical treatments of common factor analysis. Mulaik (1972) provides the most thorough and authoritative general reference on factor analysis and is highly recommended to anyone familiar with matrix algebra. Stewart (1981) gives a nontechnical presentation of some issues to consider when deciding whether or not a factor analysis might be appropriate.

A frequent source of confusion in the field of factor analysis is the term *factor*. It sometimes refers to a hypothetical, unobservable variable, as in the phrase *common factor*. In this sense, *factor analysis* must be distinguished from component analysis since a component is an observable linear combination. *Factor* is also used in the sense of *matrix factor,* in that one matrix is a factor of a second matrix if the first matrix multiplied by its transpose equals the second matrix. In this sense, *factor analysis* refers to all methods of data analysis that use matrix factors, including component analysis and common factor analysis.

A *common factor* is an unobservable, hypothetical variable that contributes to the variance of at least two of the observed variables. The unqualified term "factor" often refers to a common factor. A *unique factor* is an unobservable, hypothetical variable that contributes to the variance of only one of the observed variables. The model for common factor analysis posits one unique factor for each observed variable.

The equation for the common factor model is

$$y_{ij} = x_{i1}b_{1j} + x_{i2}b_{2j} + \cdots + x_{iq}b_{qj} + e_{ij}$$

where

y_{ij}	is the value of the ith observation on the jth variable
x_{ik}	is the value of the ith observation on the kth common factor
b_{kj}	is the regression coefficient of the kth common factor for predicting the jth variable
e_{ij}	is the value of the ith observation on the jth unique factor
q	is the number of common factors

It is assumed, for convenience, that all variables have a mean of 0. In matrix terms, these equations reduce to

$$\mathbf{Y} = \mathbf{XB} + \mathbf{E}$$

In the preceding equation, $\mathbf{X}$ is the matrix of factor scores, and $\mathbf{B}'$ is the factor pattern.

There are two critical assumptions:

- The unique factors are uncorrelated with each other.
- The unique factors are uncorrelated with the common factors.

In principal component analysis, the residuals are generally correlated with each other. In common factor analysis, the unique factors play the role of residuals and are defined to be uncorrelated both

with each other and with the common factors. Each common factor is assumed to contribute to at least two variables; otherwise, it would be a unique factor.

When the factors are initially extracted, it is also assumed, for convenience, that the common factors are uncorrelated with each other and have unit variance. In this case, the common factor model implies that the covariance s_{jk} between the jth and kth variables, $j \neq k$, is given by

$$s_{jk} = b_{1j}b_{1k} + b_{2j}b_{2k} + \cdots + b_{qj}b_{qk}$$

or

$$\mathbf{S} = \mathbf{B}'\mathbf{B} + \mathbf{U}^2$$

where $\mathbf{S}$ is the covariance matrix of the observed variables, and $\mathbf{U}^2$ is the diagonal covariance matrix of the unique factors.

If the original variables are standardized to unit variance, the preceding formula yields correlations instead of covariances. It is in this sense that common factors explain the correlations among the observed variables. When considering the diagonal elements of standardized $\mathbf{S}$, the variance of the jth variable is expressed as

$$s_{jj} = 1 = b_{1j}^2 + b_{2j}^2 + \cdots + b_{qj}^2 + [\mathbf{U}^2]_{jj}$$

where $b_{1j}^2 + b_{2j}^2 + \cdots + b_{qj}^2$ and $[\mathbf{U}^2]_{jj}$ are the communality and uniqueness, respectively, of the jth variable. The communality represents the extent of the overlap with the common factors. In other words, it is the proportion of variance accounted for by the common factors.

The difference between the correlation predicted by the common factor model and the actual correlation is the *residual correlation*. A good way to assess the goodness of fit of the common factor model is to examine the residual correlations.

The common factor model implies that the partial correlations among the variables, removing the effects of the common factors, must all be zero. When the common factors are removed, only unique factors, which are by definition uncorrelated, remain.

The assumptions of common factor analysis imply that the common factors are, in general, not linear combinations of the observed variables. In fact, even if the data contain measurements on the entire population of observations, you cannot compute the scores of the observations on the common factors. Although the common factor scores cannot be computed directly, they can be estimated in a variety of ways.

The problem of factor score indeterminacy has led several factor analysts to propose methods yielding components that can be considered approximations to common factors. Since these components are defined as linear combinations, they are computable. The methods include Harris component analysis and image component analysis. The advantage of producing determinate component scores is offset by the fact that, even if the data fit the common factor model perfectly, component methods do not generally recover the correct factor solution. You should not use any type of component analysis if you really want a common factor analysis (Dziuban and Harris 1973; Lee and Comrey 1979).

After the factors are estimated, it is necessary to interpret them. Interpretation usually means assigning to each common factor a name that reflects the *salience* of the factor in predicting each of the observed variables—that is, the coefficients in the pattern matrix corresponding to the factor. Factor interpretation is a subjective process. It can sometimes be made less subjective by *rotating* the common factors—that is, by applying a nonsingular linear transformation. A rotated pattern matrix in which all the coefficients are close to 0 or ± 1 is easier to interpret than a pattern with many intermediate elements. Therefore, most rotation methods attempt to optimize a simplicity function of the rotated pattern matrix that measures, in some sense, how close the elements are to 0 or ± 1. Because the loading estimates are subject to sampling variability, it is useful to obtain the standard error estimates for the loadings for assessing the uncertainty due to random sampling. Notice that the *salience* of a factor loading refers to the magnitude of the loading, while statistical *significance* refers to the statistical evidence against a particular hypothetical value. A loading significantly different from 0 does not automatically mean it must be salient. For example, if salience is defined as a magnitude larger than 0.4 while the entire 95% confidence interval for a loading lies between 0.1 and 0.3, the loading is statistically significant larger than 0 but it is not salient. Under the maximum likelihood method, you can obtain standard errors and confidence intervals for judging the salience of factor loadings.

After the initial factor extraction, the common factors are uncorrelated with each other. If the factors are rotated by an *orthogonal transformation,* the rotated factors are also uncorrelated. If the factors are rotated by an *oblique transformation,* the rotated factors become correlated. Oblique rotations often produce more useful patterns than do orthogonal rotations. However, a consequence of correlated factors is that there is no single unambiguous measure of the importance of a factor in explaining a variable. Thus, for oblique rotations, the pattern matrix does not provide all the necessary information for interpreting the factors; you must also examine the *factor structure* and the *reference structure*.

Rotating a set of factors does not change the statistical explanatory power of the factors. You cannot say that any rotation is better than any other rotation from a statistical point of view; all rotations, orthogonal or oblique, are equally good statistically. Therefore, the choice among different rotations must be based on nonstatistical grounds. For most applications, the preferred rotation is that which is most easily interpretable, or most compatible with substantive theories.

If two rotations give rise to different interpretations, those two interpretations must not be regarded as conflicting. Rather, they are two different ways of looking at the same thing, two different points of view in the common-factor space. Any conclusion that depends on one and only one rotation being correct is invalid.

Outline of Use

Principal Component Analysis

One important type of analysis performed by the FACTOR procedure is principal component analysis. The following statements result in a principal component analysis:

```
proc factor;
run;
```

The output includes all the eigenvalues and the pattern matrix for eigenvalues greater than one.

Most applications require additional output. For example, you might want to compute principal component scores for use in subsequent analyses or obtain a graphical aid to help decide how many components to keep. You can save the results of the analysis in a permanent SAS data library by using the OUTSTAT= option. (Refer to the *SAS Language Reference: Dictionary* for more information about permanent SAS data libraries and librefs.) Assuming that your SAS data library has the libref save and that the data are in a SAS data set called raw, you could do a principal component analysis as follows:

```
proc factor data=raw method=principal scree mineigen=0 score
     outstat=save.fact_all;
run;
```

The SCREE option produces a plot of the eigenvalues that is helpful in deciding how many components to use. Alternative, you can use the PLOTS=SCREE option to produce high-quality scree plots. The MINEIGEN=0 option causes all components with variance greater than zero to be retained. The SCORE option requests that scoring coefficients be computed. The OUTSTAT= option saves the results in a specially structured SAS data set. The name of the data set, in this case fact_all, is arbitrary. To compute principal component scores, use the SCORE procedure:

```
proc score data=raw score=save.fact_all out=save.scores;
run;
```

The SCORE procedure uses the data and the scoring coefficients that are saved in save.fact_all to compute principal component scores. The component scores are placed in variables named Factor1, Factor2, . . . , Factor*n* and are saved in the data set save.scores. If you know ahead of time how many principal components you want to use, you can obtain the scores directly from PROC FACTOR by specifying the NFACTORS= and OUT= options. To get scores from three principal components, specify the following:

```
proc factor data=raw method=principal
     nfactors=3 out=save.scores;
run;
```

To plot the scores for the first three components, use the PLOT procedure:

```
proc plot;
   plot factor2*factor1 factor3*factor1 factor3*factor2;
run;
```

Principal Factor Analysis

The simplest and computationally most efficient method of common factor analysis is principal factor analysis, which is obtained in the same way as principal component analysis except for the use of the PRIORS= option. The usual form of the initial analysis is as follows:

```
proc factor data=raw method=principal scree
     mineigen=0 priors=smc outstat=save.fact_all;
run;
```

The squared multiple correlations (SMC) of each variable with all the other variables are used as the prior communality estimates. If your correlation matrix is singular, you should specify PRIORS=MAX instead of PRIORS=SMC. The SCREE and MINEIGEN= options serve the same

purpose as in the preceding principal component analysis. Saving the results with the OUTSTAT= option enables you to examine the eigenvalues and scree plot before deciding how many factors to rotate and to try several different rotations without reextracting the factors. The OUTSTAT= data set is automatically marked TYPE=FACTOR, so the FACTOR procedure realizes that it contains statistics from a previous analysis instead of raw data.

After looking at the eigenvalues to estimate the number of factors, you can try some rotations. Two and three factors can be rotated with the following statements:

```
proc factor data=save.fact_all method=principal n=2
     rotate=promax reorder score outstat=save.fact_2;
proc factor data=save.fact_all method=principal n=3
     rotate=promax reorder score outstat=save.fact_3;
run;
```

The output data set from the previous run is used as input for these analyses. The options N=2 and N=3 specify the number of factors to be rotated. The specification ROTATE=PROMAX requests a promax rotation, which has the advantage of providing both orthogonal and oblique rotations with only one invocation of PROC FACTOR. The REORDER option causes the variables to be reordered in the output so that variables associated with the same factor appear next to each other.

You can now compute and plot factor scores for the two-factor promax-rotated solution as follows:

```
proc score data=raw score=save.fact_2 out=save.scores;
proc plot;
   plot factor2*factor1;
run;
```

Maximum Likelihood Factor Analysis

Although principal factor analysis is perhaps the most commonly used method of common factor analysis, most statisticians prefer maximum likelihood (ML) factor analysis (Lawley and Maxwell 1971). The ML method of estimation has desirable asymptotic properties (Bickel and Doksum 1977) and produces better estimates than principal factor analysis in large samples. You can test hypotheses about the number of common factors by using the ML method. You can also obtain standard error and confidence interval estimates for many classes of rotated or unrotated factor loadings, factor correlations, and structure loadings under the ML theory.

The unrotated ML solution is equivalent to Rao's (1955) canonical factor solution and Howe's solution maximizing the determinant of the partial correlation matrix (Morrison 1976). Thus, as a descriptive method, ML factor analysis does not require a multivariate normal distribution. The validity of Bartlett's χ^2 test for the number of factors does require approximate normality plus additional regularity conditions that are usually satisfied in practice (Geweke and Singleton 1980).

Lawley and Maxwell (1971) derive the standard error formulas for unrotated loadings, while Archer and Jennrich (1973) and Jennrich (1973, 1974) derive the standard error formulas for several classes of rotated solutions. Extended results appear in Browne et al. (2004), Hayashi and Yung (1999), and Yung and Hayashi (2001). A combination of these methods is used in PROC FACTOR to compute standard errors in an efficient manner. Confidence intervals are computed by using the asymptotic normality of the estimates. To ensure that the confidence intervals fall within the admissible parameter range, transformation methods due to Browne (1982) are used. The validity of the standard

error estimates and confidence limits requires the assumptions of multivariate normality and a fixed number of factors.

The ML method is more computationally demanding than principal factor analysis for two reasons. First, the communalities are estimated iteratively, and each iteration takes about as much computer time as principal factor analysis. The number of iterations typically ranges from about five to twenty. Second, if you want to extract different numbers of factors, as is often the case, you must run the FACTOR procedure once for each number of factors. Therefore, an ML analysis can take 100 times as long as a principal factor analysis. This does not include the time for computing standard error estimates, which is even more computationally demanding. For analyses with fewer than 35 variables, the computing time for the ML method, including the computation of standard errors, usually ranges from a few seconds to well under a minute. This seems to be a reasonable performance.

You can use principal factor analysis to get a rough idea of the number of factors before doing an ML analysis. If you think that there are between one and three factors, you can use the following statements for the ML analysis:

```
proc factor data=raw method=ml n=1
     outstat=save.fact1;
run;
proc factor data=raw method=ml n=2 rotate=promax
     outstat=save.fact2;
run;
proc factor data=raw method=ml n=3 rotate=promax
     outstat=save.fact3;
run;
```

The output data sets can be used for trying different rotations, computing scoring coefficients, or restarting the procedure in case it does not converge within the allotted number of iterations.

If you can determine how many factors should be retained before an analysis, as in the following statements, you can get the standard errors and confidence limits to aid interpretations for the ML analysis:

```
proc factor data=raw method=ml n=3 rotate=quartimin se
   cover=.4;
run;
```

In this analysis, you specify the quartimin rotation in the ROTATE= option. The SE option requests the computation of standard error estimates. In the COVER= option, you require absolute values of 0.4 or greater in order for loadings to be salient. In the output of coverage display, loadings that are salient would have their entire confidence intervals spanning beyond the 0.4 mark (or the -0.4 mark in the opposite direction). Only those salient loadings should be used for interpreting the factors. See the section "Confidence Intervals and the Salience of Factor Loadings" on page 1571 for more details.

The ML method cannot be used with a singular correlation matrix, and it is especially prone to Heywood cases. See the section "Heywood Cases and Other Anomalies about Communality Estimates" on page 1576 for a discussion of Heywood cases. If you have problems with ML, the best alternative is to use the METHOD=ULS option for unweighted least squares factor analysis.

Factor Rotation

After the initial factor extraction, the factors are uncorrelated with each other. If the factors are rotated by an *orthogonal transformation,* the rotated factors are also uncorrelated. If the factors are rotated by an *oblique transformation,* the rotated factors become correlated. Oblique rotations often produce more useful patterns than orthogonal rotations do. However, a consequence of correlated factors is that there is no single unambiguous measure of the importance of a factor in explaining a variable. Thus, for oblique rotations, the pattern matrix does not provide all the necessary information for interpreting the factors; you must also examine the *factor structure* and the *reference structure*.

Nowadays, most rotations are done analytically. There are many choices for orthogonal and oblique rotations. An excellent summary of a wide class of analytic rotations is in Crawford and Ferguson (1970). The Crawford-Ferguson family of orthogonal rotations includes the orthomax rotation as a subclass and the popular varimax rotation as a special case. To illustrate these relationships, the following four specifications for orthogonal rotations with different ROTATE= options will give the same results for a data set with nine observed variables:

```
      /* Orthogonal Crawford-Ferguson Family with
         variable parsimony weight = nvar - 1 = 8, and
         factor parsimony weight = 1  */
proc factor data=raw n=3 rotate=orthcf(8,1);
run;

    /* Orthomax without the GAMMA= option */
proc factor data=raw n=3 rotate=orthomax(1);
run;

    /* Orthomax with the GAMMA= option */
proc factor data=raw n=3 rotate=orthomax gamma=1;
run;

    /* Varimax */
proc factor data=raw n=3 rotate=varimax;
run;
```

You can also get the oblique versions of the varimax in two equivalent ways:

```
      /* Oblique Crawford-Ferguson Family with
         variable parsimony weight = nvar - 1 = 8, and
         factor parsimony weight = 1; */
proc factor data=raw n=3 rotate=oblicf(8,1);
run;

    /* Oblique Varimax */
proc factor data=raw n=3 rotate=obvarimax;
run;
```

Jennrich (1973) proposes a generalized Crawford-Ferguson family that includes the Crawford-Ferguson family and the (direct) oblimin family (refer to Harman 1976) as subclasses. The better-known quartimin rotation is a special case of the oblimin class, and hence a special case of the generalized Crawford-Ferguson family. For example, the following four specifications of oblique rotations are equivalent:

```
   /* Oblique generalized Crawford-Ferguson Family
      with weights 0, 1, 0, -1 */
proc factor data=raw n=3 rotate=obligencf(0,1,0,-1);
run;

   /* Oblimin family without the TAU= option */
proc factor data=raw n=3 rotate=oblimin(0);
run;

   /* Oblimin family with the TAU= option */
proc factor data=raw n=3 rotate=oblimin tau=0;
run;

   /* Quartimin */
proc factor data=raw n=3 rotate=quartimin;
run;
```

In addition to the generalized Crawford-Ferguson family, the available oblique rotation methods in PROC FACTOR include Harris-Kaiser, promax, and Procrustes. See the section "Simplicity Functions for Rotations" on page 1572 for details about the definitions of various rotations. Refer to Harman (1976) and Mulaik (1972) for further information.

Getting Started: FACTOR Procedure

The following example demonstrates how you can use the FACTOR procedure to perform common factor analysis and factor rotation.

In this example, 103 police officers were rated by their supervisors on 14 scales (variables). You conduct a common factor analysis on these variables to see what latent factors are operating behind these ratings. The overall rating variable is excluded from the factor analysis.

The following DATA step creates the SAS data set jobratings:

```
options validvarname=any;
data jobratings;
   input ('Communication Skills'n
          'Problem Solving'n
          'Learning Ability'n
          'Judgment Under Pressure'n
          'Observational Skills'n
          'Willingness to Confront Problems'n
          'Interest in People'n
          'Interpersonal Sensitivity'n
          'Desire for Self-Improvement'n
          'Appearance'n
          'Dependability'n
          'Physical Ability'n
          'Integrity'n
          'Overall Rating'n) (1.);
   datalines;
26838853879867
74758876857667
56757863775875
67869777988997
99997798878888
89897899888799
89999889899798

... more lines ...

99899899899899
76656399567486
;
```

The following statements invoke the FACTOR procedure:

```
proc factor data=jobratings(drop='Overall Rating'n) priors=smc
   rotate=varimax;
run;
```

The DATA= option in PROC FACTOR specifies the SAS data set jobratings as the input data set. The DROP= option drops the Overall Rating variable from the analysis. To conduct a common factor analysis, you need to set the prior communality estimate to less than one for each variable. Otherwise, the factor solution would simply be a recast of the principal components solution, in which "factors" are linear combinations of observed variables. However, in the common factor model you always assume that observed variables are functions of underlying factors. In this example, the PRIORS= option specifies that the squared multiple correlations (SMC) of each variable with all the other variables are used as the prior communality estimates. Note that squared multiple correlations are usually less than one. By default, the principal factor extraction is used if the METHOD= option is not specified. To facilitate interpretations, the ROTATE= option specifies the VARIMAX orthogonal factor rotation to be used.

The output from the factor analysis is displayed in Figure 33.1 through Figure 33.5.

As displayed in Figure 33.1, the prior communality estimates are set to the squared multiple correlations. Figure 33.1 also displays the table of eigenvalues (the variances of the principal factors) of the reduced correlation matrix. Each row of the table pertains to a single eigenvalue. Following the column of eigenvalues are three measures of each eigenvalue's relative size and importance. The first of these displays the difference between the eigenvalue and its successor. The last two columns display the individual and cumulative proportions that the corresponding factor contributes to the total variation. The last line displayed in Figure 33.1 states that three factors are retained, as determined by the PROPORTION criterion.

Figure 33.1 Table of Eigenvalues from PROC FACTOR

The FACTOR Procedure
Initial Factor Method: Principal Factors

Prior Communality Estimates: SMC

Communication Skills	Problem Solving	Learning Ability	Judgment Under Pressure	Observational Skills
0.62981394	0.58657431	0.61009871	0.63766021	0.67187583

Willingness to Confront Problems	Interest in People	Interpersonal Sensitivity	Desire for Self-Improvement
0.64779805	0.75641519	0.75584891	0.57460176

Appearance	Dependability	Physical Ability	Integrity
0.45505304	0.63449045	0.42245324	0.68195454

Eigenvalues of the Reduced Correlation Matrix:
Total = 8.06463816 Average = 0.62035678

	Eigenvalue	Difference	Proportion	Cumulative
1	6.17760549	4.71531946	0.7660	0.7660
2	1.46228602	0.90183348	0.1813	0.9473
3	0.56045254	0.28093933	0.0695	1.0168
4	0.27951322	0.04766016	0.0347	1.0515
5	0.23185305	0.16113428	0.0287	1.0802
6	0.07071877	0.07489624	0.0088	1.0890
7	-.00417747	0.03387533	-0.0005	1.0885
8	-.03805279	0.04776534	-0.0047	1.0838
9	-.08581814	0.02438060	-0.0106	1.0731
10	-.11019874	0.01452741	-0.0137	1.0595
11	-.12472615	0.02356465	-0.0155	1.0440
12	-.14829080	0.05823605	-0.0184	1.0256
13	-.20652684		-0.0256	1.0000

3 factors will be retained by the PROPORTION criterion.

Figure 33.2 displays the initial factor pattern matrix. The factor pattern matrix represents standardized regression coefficients for predicting the variables by using the extracted factors. Because the initial factors are uncorrelated, the pattern matrix is also equal to the correlations between variables and the common factors.

Figure 33.2 Factor Pattern Matrix from PROC FACTOR

Factor Pattern

	Factor1	Factor2	Factor3
Communication Skills	0.75441	0.07707	−0.25551
Problem Solving	0.68590	0.08026	−0.34788
Learning Ability	0.65904	0.34808	−0.25249
Judgment Under Pressure	0.73391	−0.21405	−0.23513
Observational Skills	0.69039	0.45292	0.10298
Willingness to Confront Problems	0.66458	0.47460	0.09210
Interest in People	0.70770	−0.53427	0.10979
Interpersonal Sensitivity	0.64668	−0.61284	−0.07582
Desire for Self-Improvement	0.73820	0.12506	0.09062
Appearance	0.57188	0.20052	0.16367
Dependability	0.79475	−0.04516	0.16400
Physical Ability	0.51285	0.10251	0.34860
Integrity	0.74906	−0.35091	0.18656

The pattern matrix suggests that Factor1 represents general ability. All loadings for Factor1 in the Factor Pattern are at least 0.5. Factor2 consists of high positive loadings on certain task-related skills (Willingness to Confront Problems, Observational Skills, and Learning Ability) and high negative loadings on some interpersonal skills (Interpersonal Sensitivity, Interest in People, and Integrity). This factor measures individuals' relative strength in these skills. Theoretically, individuals with high positive scores on this factor would exhibit better task-related skills than interpersonal skills. Individuals with high negative scores would exhibit better interpersonal skills than task-related skills. Individuals with scores near zero would have those skills balanced. Factor3 does not have a cluster of very high or very low factor loadings. Therefore, interpreting this factor is difficult.

Figure 33.3 displays the proportion of variance explained by each factor and the final communality estimates, including the total communality. The final communality estimates are the proportion of variance of the variables accounted for by the common factors. When the factors are orthogonal, the final communalities are calculated by taking the sum of squares of each row of the factor pattern matrix.

Figure 33.3 Variance Explained and Final Communality Estimates

Variance Explained by Each Factor

Factor1	Factor2	Factor3
6.1776055	1.4622860	0.5604525

Figure 33.3 *continued*

Final Communality Estimates: Total = 8.200344

Communication Skills	Problem Solving	Learning Ability	Judgment Under Pressure	Observational Skills
0.64036292	0.59791844	0.61924167	0.63972863	0.69237485

Willingness to Confront Problems	Interest in People	Interpersonal Sensitivity	Desire for Self-Improvement
0.67538695	0.79833968	0.79951357	0.56879171

Appearance	Dependability	Physical Ability	Integrity
0.39403630	0.66056907	0.39504805	0.71903222

Figure 33.4 displays the results of the VARIMAX rotation of the three extracted factors and the corresponding orthogonal transformation matrix. The rotated factor pattern matrix is calculated by postmultiplying the original factor pattern matrix (Figure 33.4) by the transformation matrix.

Figure 33.4 Transformation Matrix and Rotated Factor Pattern

The FACTOR Procedure
Rotation Method: Varimax

Orthogonal Transformation Matrix

	1	2	3
1	0.59125	0.59249	0.54715
2	−0.80080	0.51170	0.31125
3	0.09557	0.62219	−0.77701

Rotated Factor Pattern

	Factor1	Factor2	Factor3
Communication Skills	0.35991	0.32744	0.63530
Problem Solving	0.30802	0.23102	0.67058
Learning Ability	0.08679	0.41149	0.66512
Judgment Under Pressure	0.58287	0.17901	0.51764
Observational Skills	0.05533	0.70488	0.43870
Willingness to Confront Problems	0.02168	0.69391	0.43978
Interest in People	0.85677	0.21422	0.13562
Interpersonal Sensitivity	0.86587	0.02239	0.22200
Desire for Self-Improvement	0.34498	0.55775	0.37242
Appearance	0.19319	0.54327	0.24814
Dependability	0.52174	0.54981	0.29337
Physical Ability	0.25445	0.57321	0.04165
Integrity	0.74172	0.38033	0.15567

The rotated factor pattern matrix is somewhat simpler to interpret. If a magnitude of at least 0.5 is required to indicate a salient variable-factor relationship, Factor1 now represents interpersonal skills (Interpersonal Sensitivity, Interest in People, Integrity, Judgment Under Pressure, and Dependability). Factor2 measures physical skills and job enthusiasm (Observational Skills, Willingness to Confront Problems, Physical Ability, Desire for Self-Improvement, Dependability, and Appearance). Factor3 measures cognitive skills (Communication Skills, Problem Solving, Learning Ability, and Judgment Under Pressure).

However, using 0.5 for determining a salient variable-factor relationship does not take sampling variability into account. If the underlying assumptions for the maximum likelihood estimation are approximately satisfied, you can output standard error estimates and the confidence intervals with METHOD=ML. You can then determine the salience of the variable-factor relationship by using the coverage displays. See the section "Confidence Intervals and the Salience of Factor Loadings" on page 1571 for more details.

Figure 33.5 displays the variance explained by each factor and the final communality estimates after the orthogonal rotation. Even though the variances explained by the rotated factors are different from that of the unrotated factor (compare with Figure 33.3), the cumulative variance explained by the common factors remains the same. Note also that the final communalities for variables, as well as the total communality, remain unchanged after rotation. Although rotating a factor solution will not increase or decrease the statistical quality of the factor model, it can simplify the interpretations of the factors and redistribute the variance explained by the factors.

Figure 33.5 Variance Explained and Final Communality Estimates after Rotation

Variance Explained by Each Factor

Factor1	Factor2	Factor3
3.1024330	2.7684489	2.3294622

Final Communality Estimates: Total = 8.200344

Communication Skills	Problem Solving	Learning Ability	Judgment Under Pressure	Observational Skills
0.64036292	0.59791844	0.61924167	0.63972863	0.69237485

Willingness to Confront Problems	Interest in People	Interpersonal Sensitivity	Desire for Self-Improvement
0.67538695	0.79833968	0.79951357	0.56879171

Appearance	Dependability	Physical Ability	Integrity
0.39403630	0.66056907	0.39504805	0.71903222

Syntax: FACTOR Procedure

You can specify the following statements with the FACTOR procedure:

PROC FACTOR < *options* > ;
  VAR *variables* ;
  PRIORS *communalities* ;
  PARTIAL *variables* ;
  FREQ *variable* ;
  WEIGHT *variable* ;
  BY *variables* ;

Usually only the VAR statement is needed in addition to the PROC FACTOR statement. The descriptions of the BY, FREQ, PARTIAL, PRIORS, VAR, and WEIGHT statements follow the description of the PROC FACTOR statement in alphabetical order.

PROC FACTOR Statement

PROC FACTOR < *options* > ;

The options available with the PROC FACTOR statement are listed in the following table and then described in alphabetical order.

Table 33.1 Options Available in the PROC FACTOR Statement

Option	Description
Data Set Options	
DATA=	specifies input SAS data set
OUT=	specifies output SAS data set
OUTSTAT=	specifies output data set containing statistical results
TARGET=	specifies input data set containing the target pattern for rotation
Factor Extraction and Communalities	
HEYWOOD	sets to 1 any communality greater than 1
METHOD=	specifies the estimation method
PRIORS=	specifies the method for computing prior communality estimates
RANDOM=	specifies the seed for pseudo-random number generation
ULTRAHEYWOOD	allows communalities to exceed 1
Number of Factors	
MINEIGEN=	specifies the smallest eigenvalue for retaining a factor
NFACTORS=	specifies the number of factors to retain

Table 33.1 *continued*

Option	Description
PROPORTION=	specifies the proportion of common variance in extracted factors
Data Analysis Options	
ALPHA=	specifies the confidence level for interval construction
COVARIANCE	requests factoring of the covariance matrix
COVER=	computes the confidence interval and specifies the coverage reference point
NOINT	omits the intercept from computing covariances or correlations
SE	requests the standard error estimates in ML estimation
VARDEF=	specifies the divisor used in calculating covariances or correlations
WEIGHT	factors a weighted correlation or covariance matrix
Rotation Method and Properties	
GAMMA=	specifies the orthomax weight
HKPOWER=	specifies the power in Harris-Kaiser rotation
NORM=	specifies the method for row normalization in rotation
NOPROMAXNORM	turns off row normalization in promax rotation
POWER=	specifies the power to be used in promax rotation
PREROTATE=	specifies the prerotation method in promax rotation
RCONVERGE=	specifies the convergence criterion for rotation cycles
RITER=	specifies the maximum number of cycles for rotation
ROTATE=	specifies the rotation method
TAU=	specifies the oblimin weight
ODS Graphics	
PLOTS=	specifies ODS Graphics selection
Control Display Output	
ALL	displays all optional output except plots
CORR	displays the (partial) correlation matrix
EIGENVECTORS	displays the eigenvectors of the reduced correlation matrix
FLAG=	specifies the minimum absolute value to be flagged in the correlation and loading matrices
FUZZ=	specifies the maximum absolute value to be displayed as missing in the correlation and loading matrices
MSA	computes Kaiser's measure of sampling adequacy and the related partial correlations
NOPRINT	suppresses the display of all output
NPLOT=	specifies the number of factors to be plotted
PLOT	plots the rotated factor pattern
PLOTREF	plots the reference structure
PREPLOT	plots the factor pattern before rotation

Table 33.1 *continued*

Option	Description
PRINT	displays the input factor pattern or scoring coefficients and related statistics
REORDER	reorders the rows (variables) of various factor matrices
RESIDUALS	displays the residual correlation matrix and the associated partial correlation matrix
ROUND	prints correlation and loading matrices with rounded values
SCORE	displays the factor scoring coefficients
SCREE	displays the scree plot of the eigenvalues
SIMPLE	displays means, standard deviations, and number of observations
Numerical Properties	
CONVERGE=	specifies the convergence criterion
MAXITER=	specifies the maximum number of iterations
SINGULAR=	specifies the singularity criterion
Miscellaneous	
NOCORR	excludes the correlation matrix from the OUTSTAT= data set
NOBS=	specifies the number of observations
PARPREFIX=	specifies the prefix for the residual variables in the output data sets
PREFIX=	specifies the prefix for naming factors

ALL

displays all optional output except plots. When the input data set is TYPE=CORR, TYPE=UCORR, TYPE=COV, TYPE=UCOV, or TYPE=FACTOR, simple statistics, correlations, and MSA are not displayed.

ALPHA=*p*

specifies the level of confidence $1-p$ for interval construction. By default, $p = 0.05$, corresponding to $1-p = 95\%$ confidence intervals. If p is greater than one, it is interpreted as a percentage and divided by 100. With multiple confidence intervals to be constructed, the ALPHA= value is applied to each interval construction one at a time. This will not control the coverage probability of the intervals simultaneously. To control familywise coverage probability, you might consider supplying a nonconventional p by using methods such as Bonferroni adjustment.

CONVERGE=*p*

CONV=*p*

specifies the convergence criterion for the METHOD=PRINIT, METHOD=ULS, METHOD=ALPHA, or METHOD=ML option. Iteration stops when the maximum change in the communalities is less than the value of the CONVERGE= option. The default value is 0.001. Negative values are not allowed.

CORR

C

displays the correlation matrix or partial correlation matrix.

COVARIANCE

COV

requests factoring of the covariance matrix instead of the correlation matrix. The COV option is effective only with the METHOD=PRINCIPAL, METHOD=PRINIT, METHOD=ULS, or METHOD=IMAGE option. For other methods, PROC FACTOR produces the same results with or without the COV option.

COVER <=*p*>

CI <=*p*>

computes the confidence intervals and optionally specifies the value of factor loading for coverage detection. By default, $p = 0$. The specified value is represented by an asterisk (*) in the coverage display. This is useful for determining the salience of loadings. For example, if COVER=0.4, a display '0*[ ]' indicates that the entire confidence interval is above 0.4, implying strong evidence for the salience of the loading. See the section "Confidence Intervals and the Salience of Factor Loadings" on page 1571 for more details.

DATA=*SAS-data-set*

specifies the input data set, which can be an ordinary SAS data set or a specially structured SAS data set as described in the section "Input Data Set" on page 1566. If the DATA= option is omitted, the most recently created SAS data set is used.

EIGENVECTORS

EV

displays the eigenvectors of the reduced correlation matrix, of which the diagonal elements are replaced with the communality estimates. When METHOD=ML, the eigenvectors are for the weighted reduced correlation matrix. PROC FACTOR chooses the solution that makes the sum of the elements of each eigenvector nonnegative. If the sum of the elements is equal to zero, then the sign depends on how the number is rounded off.

FLAG=*p*

flags absolute values larger than *p* with an asterisk in the correlation and loading matrices. Negative values are not allowed for *p*. Values printed in the matrices are multiplied by 100 and rounded to the nearest integer (see the ROUND option). The FLAG= option has no effect when standard errors or confidence intervals are also printed.

FUZZ=*p*

prints correlations and factor loadings with absolute values less than *p* printed as missing. For partial correlations, the FUZZ= value is divided by 2. For residual correlations, the FUZZ= value is divided by 4. The exact values in any matrix can be obtained from the OUTSTAT= and ODS output data sets. Negative values are not allowed. The FUZZ= option has no effect when standard errors or confidence intervals are also printed.

GAMMA=*p*

specifies the orthomax weight used with the option ROTATE=ORTHOMAX or PREROTATE=ORTHOMAX. Alternatively, you can use ROTATE=ORTHOMAX(*p*) with *p* representing the orthomax weight. There is no restriction on valid values for the orthomax weight, although the most common values are between zero and the number of variables. The default GAMMA= value is one, resulting in the varimax rotation. See the section "Simplicity Functions for Rotations" on page 1572 for more details.

HEYWOOD

HEY

sets to 1 any communality greater than 1, allowing iterations to proceed. See the section "Heywood Cases and Other Anomalies about Communality Estimates" on page 1576 for a discussion of Heywood cases.

HKPOWER=*p*

HKP=*p*

specifies the power of the square roots of the eigenvalues used to rescale the eigenvectors for Harris-Kaiser (ROTATE=HK) rotation, assuming that the factors are extracted by the principal factor method. If the principal factor method is not used for factor extraction, the eigenvectors are replaced by the normalized columns of the unrotated factor matrix, and the eigenvalues are replaced by the column normalizing constants. HKPOWER= values between 0.0 and 1.0 are reasonable. The default value is 0.0, yielding the independent cluster solution, in which each variable tends to have a large loading on only one factor. An HKPOWER= value of 1.0 is equivalent to an orthogonal rotation, with the varimax rotation as the default. You can also specify the HKPOWER= option with ROTATE=QUARTIMAX, ROTATE=BIQUARTIMAX, ROTATE=EQUAMAX, or ROTATE=ORTHOMAX, and so on. The only restriction is that the Harris-Kaiser rotation must be associated with an orthogonal rotation.

MAXITER=*n*

specifies the maximum number of iterations for factor extraction. You can use the MAXITER= option with the PRINIT, ULS, ALPHA, or ML method. The default is 30.

METHOD=*name*

M=*name*

specifies the method for extracting factors. The default is METHOD=PRINCIPAL unless the DATA= data set is TYPE=FACTOR, in which case the default is METHOD=PATTERN. Valid values for *name* are as follows:

ALPHA | A — produces alpha factor analysis.

HARRIS | H — yields Harris component analysis of $\mathbf{S}^{-1}\mathbf{R}\mathbf{S}^{-1}$ (Harris 1962), a noniterative approximation to canonical component analysis.

IMAGE | I — yields principal component analysis of the image covariance matrix, not Kaiser's (1963, 1970) or Kaiser and Rice's (1974) image analysis. A nonsingular correlation matrix is required.

ML | M — performs maximum likelihood factor analysis with an algorithm due, except for minor details, to Fuller (1987). The option METHOD=ML requires a nonsingular correlation matrix.

PATTERN reads a factor pattern from a TYPE=FACTOR, TYPE=CORR, TYPE=UCORR, TYPE=COV, or TYPE=UCOV data set. If you create a TYPE=FACTOR data set in a DATA step, only observations containing the factor pattern (_TYPE_='PATTERN') and, if the factors are correlated, the interfactor correlations (_TYPE_='FCORR') are required.

PRINCIPAL | PRIN | P yields principal component analysis if no PRIORS option or statement is used or if you specify PRIORS=ONE; if you specify a PRIORS statement or a PRIORS= value other than PRIORS=ONE, a principal factor analysis is performed.

PRINIT yields iterated principal factor analysis.

SCORE reads scoring coefficients (_TYPE_='SCORE') from a TYPE=FACTOR, TYPE=CORR, TYPE=UCORR, TYPE=COV, or TYPE=UCOV data set. The data set must also contain either a correlation or a covariance matrix. Scoring coefficients are also displayed if you specify the OUT= option.

ULS | U produces unweighted least squares factor analysis.

MINEIGEN=*p*

MIN=*p*

specifies the smallest eigenvalue for which a factor is retained. If you specify two or more of the MINEIGEN=, NFACTORS=, and PROPORTION= options, the number of factors retained is the minimum number satisfying any of the criteria. The MINEIGEN= option cannot be used with either the METHOD=PATTERN or the METHOD=SCORE option. Negative values are not allowed. The default is 0 unless you omit both the NFACTORS= and the PROPORTION= options and one of the following conditions holds:

- If you specify the METHOD=ALPHA or METHOD=HARRIS option, then MINEIGEN=1.
- If you specify the METHOD=IMAGE option, then

$$\text{MINEIGEN} = \frac{\text{total image variance}}{\text{number of variables}}$$

- For any other METHOD= specification, if prior communality estimates of 1.0 are used, then

$$\text{MINEIGEN} = \frac{\text{total weighted variance}}{\text{number of variables}}$$

When an unweighted correlation matrix is factored, this value is 1.

MSA

produces the partial correlations between each pair of variables controlling for all other variables (the negative anti-image correlations) and Kaiser's measure of sampling adequacy (Kaiser 1970; Kaiser and Rice 1974; Cerny and Kaiser 1977).

NFACTORS=*n*

NFACT=*n*

N=*n*

specifies the maximum number of factors to be extracted and determines the amount of memory to be allocated for factor matrices. The default is the number of variables. Specifying a number that is small relative to the number of variables can substantially decrease the amount of memory required to run PROC FACTOR, especially with oblique rotations. If you specify two or more of the NFACTORS=, MINEIGEN=, and PROPORTION= options, the number of factors retained is the minimum number satisfying any of the criteria. If you specify the option NFACTORS=0, eigenvalues are computed, but no factors are extracted. If you specify the option NFACTORS=−1, neither eigenvalues nor factors are computed. You can use the NFACTORS= option with the METHOD=PATTERN or METHOD=SCORE option to specify a smaller number of factors than are present in the data set.

NOBS=*n*

specifies the number of observations. If the DATA= input data set is a raw data set, *nobs* is defined by default to be the number of observations in the raw data set. The NOBS= option overrides this default definition. If the DATA= input data set contains a covariance, correlation, or scalar product matrix, the number of observations can be specified either by using the NOBS= option in the PROC FACTOR statement or by including a _TYPE_='N' observation in the DATA= input data set.

NOCORR

prevents the correlation matrix from being transferred to the OUTSTAT= data set when you specify the METHOD=PATTERN option. The NOCORR option greatly reduces memory requirements when there are many variables but few factors. The NOCORR option is not effective if the correlation matrix is required for other requested output; for example, if the scores or the residual correlations are displayed (for example, by using the SCORE, RESIDUALS, or ALL option).

NOINT

omits the intercept from the analysis; covariances or correlations are not corrected for the mean.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 20, "Using the Output Delivery System."

NOPROMAXNORM | NOPMAXNORM

turns off the default row normalization of the prerotated factor pattern, which is used in computing the promax target matrix.

NORM=COV | KAISER | NONE | RAW | WEIGHT

specifies the method for normalizing the rows of the factor pattern for rotation. If you specify the option NORM=KAISER, Kaiser's normalization is used ($\sum_j p_{ij}^2 = 1$). If you specify the option NORM=WEIGHT, the rows are weighted by the Cureton-Mulaik technique (Cureton and Mulaik 1975). If you specify the option NORM=COV, the rows of the pattern matrix are rescaled to represent covariances instead of correlations. If you specify

the option NORM=NONE or NORM=RAW, normalization is not performed. The default is NORM=KAISER.

NPLOTS | NPLOT=*n*

specifies the number of factors to be plotted. The default is to plot all factors. The smallest allowable value is 2. If you specify the option NPLOTS=*n*, all pairs of the first *n* factors are plotted, producing a total of $n(n-1)/2$ plots.

OUT=*SAS-data-set*

creates a data set containing all the data from the DATA= data set plus variables called Factor1, Factor2, and so on, containing estimated factor scores. The DATA= data set must contain multivariate data, not correlations or covariances. You must also specify the NFACTORS= option to determine the number of factor score variables. If you specify partial variables in the PARTIAL statement, the OUT= data set will also contain the residual variables that are used for factor analysis. The output data set is described in detail in the section "Output Data Sets" on page 1568. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to "SAS Files" in *SAS Language Reference: Concepts* for more information about permanent data sets.

OUTSTAT=*SAS-data-set*

specifies an output data set containing most of the results of the analysis. The output data set is described in detail in the section "Output Data Sets" on page 1568. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to "SAS Files" in *SAS Language Reference: Concepts* for more information about permanent data sets.

PARPREFIX=*name*

specifies the prefix for the residual variables in the OUT= and the OUTSTAT= data sets when partial variables are specified in the PARTIAL statement.

PLOT

plots the factor pattern after rotation. This option produces printer plots. High-quality ODS graphical plots for factor patterns can be requested with the PLOTS=LOADINGS or PLOTS=INITLOADINGS option.

PLOTREF

plots the reference structure instead of the default factor pattern after oblique rotation.

PLOTS <(*global-plot-options***)> = *plot-request* <(***options***)>**

PLOTS <(*global-plot-options***)> = (*plot-request* <(***options***)> <... *plot-request* <(***options***)> >)**

specifies one or more ODS graphical plots in PROC FACTOR. When you specify only one *plot-request*, you can omit the parentheses around the *plot-request*. Here are some examples:

```
plots=all
plots(flip)=loadings
plots=(loadings(flip) scree(unpack))
```

You must enable ODS graphics before requesting plots. For example:

```
ods graphics on;
proc factor plots=all;
run;
ods graphics off;
```

For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS." For an example containing graphical displays of factor analysis results, see Example 33.2.

The following table shows the available *plot-requests* and their available suboptions:

Plot-Request	**Plot Description**	**Suboptions**
ALL	all available plots	all
INITLOADINGS	unrotated factor loadings	CIRCLE=, FLIP, NPLOTS=, PLOTREF, and VECTOR
LOADINGS	rotated factor loadings	CIRCLE=, FLIP, NPLOTS=, PLOTREF, and VECTOR
NONE	no ODS graphical plots	
PRELOADINGS	prerotated factor loadings	CIRCLE=, FLIP, NPLOTS=, PLOTREF, and VECTOR
SCREE	scree and variance explained	UNPACK

The following are the available *global-plot-options* or *options* for plots:

CIRCLE | CIRCLES < = *numbers* > specifies circles with prescribed areas to be drawn in scatter plots or vector plots, where the optional *numbers* represent proportions or percentages of areas enclosed by the circles in the plots. These *numbers* must lie between 0 and 100, inclusively. When a number is between 0 and 1 (inclusively), it is interpreted as a proportion; otherwise, it is interpreted as a percentage. A maximum of 5 numbers for the circles will be used. The CIRCLE option applies to the scatter or vector plots requested by the INITLOADINGS, LOADINGS, and PRELOADINGS options. By default, a unit-circle, which represents 100% of the total area, is drawn for the vector plots. However, no circle will be drawn for scatter plots unless the CIRCLE option is specified. Two special cases for this option are: (1) With no *numbers* following the CIRCLE option, a 100% circle will be drawn. (2) With CIRCLE=0, no circle will be drawn. This special case is primarily used to turn off the default unit-circle in vector plots.

FLIP switches the X and Y axes. It applies to the INITLOADINGS, LOADINGS, and PRELOADINGS *plot-requests*.

NPLOT | NPLOTS=*n* specifies the number of factors n ($n \geq 2$) to be plotted. It applies to the INITLOADINGS, LOADINGS, and PRELOADINGS *plot-requests*. Since this option can also be specified in the PROC FACTOR statement, the final value of n is determined by the following steps. The NPLOTS= value of the PROC FACTOR is read first. If the NPLOTS= option is specified as a *global-plot-option*, the value of n will be updated. Then, if the NPLOTS= option is again specified in an individual *plot-request*, the value will be updated again for that individual *plot-request*. For example, in the following statement, four factors are extracted with the N=4 option:

```
proc factor n=4 nplots=3 plots(nplots=4)=
                (loadings preloadings(nplots=2));
```

Initially, plots of the first three factors are specified with the NPLOTS=3 option. When you are producing ODS graphical plots, the *global-plot-option* NPLOTS=4 is used. As a result, the LOADINGS *plot-request* will produce plots for all pairs of the first 4 factors. However, because the NPLOTS=2 is specified locally for the PRELOADINGS *plot-request*, it will produce a prerotated factor loading plot for the first two factors only.

The default NPLOTS= value is 5 or the total number of factors (m), whichever is smaller. If you specify an NPLOTS= value that is greater than m, NPLOTS=m will be used.

PLOTREF plots the reference structures rather than the factor pattern loadings. It applies to the INITLOADINGS, LOADINGS, and PRELOADINGS *plot-requests* when the factor solution is oblique. This option can also be set globally as an option in the PROC FACTOR statement.

UNPACK plots component graphs separately. It applies to the SCREE *plot-request* only.

VECTOR plots loadings in vector form. It applies to the INITLOADINGS, LOADINGS, and PRELOADINGS *plot-requests* when the factor solution is orthogonal. For oblique solutions, the VECTOR option is ignored and the default scatter plots for factor loadings or reference structures are displayed.

Be aware that the PLOT option in the PROC FACTOR statement requests only the printer plots of factor loadings. The current option PLOTS= or PLOT=, however, is for ODS graphical plots.

You can specify options for the requested ODS graphical plots as *global-plot-options* or as local *options*. *Global-plot-options* apply to all appropriate individual *plot-requests* specified. For example, because the SCREE plot is not subject to axes flipping, the following two specifications are equivalent:

```
plots(flip)=(loadings preloadings scree)
plots=(loadings(flip) preloadings(flip) scree)
```

Options specified locally after each *plot-request* apply to that plot-request only. For example, consider the following specification:

```
plots=(scree(unpack) loadings(plotref) preloadings(flip))
```

The FLIP option applies to the PRELOADINGS *plot-request* but not the LOADINGS *plot-request*; the PLOTREF option applies to the LOADINGS *plot-request* but not the PRELOADINGS *plot-request*; and the UNPACK option applies to the SCREE *plot-request* only.

POWER=*n*

specifies the power to be used in computing the target pattern for the option ROTATE=PROMAX. Valid values must be integers ≥ 1. The default value is 3. You can also specify the POWER= value in the ROTATE= option—for example, ROTATE=PROMAX(4).

PREFIX=*name*

specifies a prefix for naming the factors. By default, the names are Factor1, Factor2, ..., Factor*n*. If you specify PREFIX=ABC, the factors are named ABC1, ABC2, ABC3, and so on. The number of characters in the prefix plus the number of digits required to designate the variables should not exceed the current name length defined by the VALIDVARNAME= system option.

PREPLOT

plots the factor pattern before rotation. This option produces printer plots. High-quality ODS graphical plots for factor patterns can be requested with the PLOTS=PRELOADINGS option.

PREROTATE=*name*

PRE=*name*

specifies the prerotation method for the option ROTATE=PROMAX. Any rotation method other than PROMAX or PROCRUSTES can be used. The default is PREROTATE=VARIMAX. If a previously rotated pattern is read using the option METHOD=PATTERN, you should specify the PREROTATE=NONE option.

PRINT

displays the input factor pattern or scoring coefficients and related statistics. In oblique cases, the reference and factor structures are computed and displayed. The PRINT option is effective only with the option METHOD=PATTERN or METHOD=SCORE.

PRIORS=*name*

specifies a method for computing prior communality estimates. You can specify numeric values for the prior communality estimates by using the PRIORS statement. Valid values for *name* are as follows:

ASMC | A — sets the prior communality estimates proportional to the squared multiple correlations but adjusted so that their sum is equal to that of the maximum absolute correlations (Cureton 1968).

INPUT | I — reads the prior communality estimates from the first observation with either _TYPE_='PRIORS' or _TYPE_='COMMUNAL' in the DATA= data set (which cannot be TYPE=DATA).

MAX | M — sets the prior communality estimate for each variable to its maximum absolute correlation with any other variable.

ONE | O — sets all prior communalities to 1.0.

RANDOM | R — sets the prior communality estimates to pseudo-random numbers uniformly distributed between 0 and 1.

SMC | S — sets the prior communality estimate for each variable to its squared multiple correlation with all other variables.

The default prior communality estimates are as follows:

METHOD=	PRIORS=
PRINCIPAL	ONE
PRINIT	ONE
ALPHA	SMC
ULS	SMC
ML	SMC
HARRIS	(not applicable)
IMAGE	(not applicable)
PATTERN	(not applicable)
SCORE	(not applicable)

By default, the options METHOD=PRINIT, METHOD=ULS, METHOD=ALPHA, and METHOD=ML stop iterating and set the number of factors to 0 if an estimated communality exceeds 1. The options HEYWOOD and ULTRAHEYWOOD allow processing to continue.

PROPORTION=*p*

PERCENT=*p*

P=*p*

specifies the proportion of common variance to be accounted for by the retained factors. The proportion of common variance is computed using the total prior communality estimates as the basis. If the value is greater than one, it is interpreted as a percentage and divided by 100. The options PROPORTION=0.75 and PERCENT=75 are equivalent. The default value is 1.0 or 100%. You cannot specify the PROPORTION= option with the METHOD=PATTERN or METHOD=SCORE option. If you specify two or more of the PROPORTION=, NFACTORS=, and MINEIGEN= options, the number of factors retained is the minimum number satisfying any of the criteria.

RANDOM=*n*

specifies a positive integer as a starting value for the pseudo-random number generator for use with the option PRIORS=RANDOM. If you do not specify the RANDOM= option, the time of day is used to initialize the pseudo-random number sequence. Valid values must be integers ≥ 1.

RCONVERGE=*p*

RCONV=*p*

specifies the convergence criterion for rotation cycles. Rotation stops when the scaled change of the simplicity function value is less than the RCONVERGE= value. The default convergence criterion is

$$|f_{new} - f_{old}|/K < \epsilon$$

where f_{new} and f_{old} are simplicity function values of the current cycle and the previous cycle, respectively, $K = max(1, |f_{old}|)$ is a scaling factor, and ϵ is 1E–9 by default and is modified by the RCONVERGE= value.

REORDER

RE

causes the rows (variables) of various factor matrices to be reordered on the output. Variables with their highest absolute loading (reference structure loading for oblique rotations) on the first factor are displayed first, from largest to smallest loading, followed by variables with their highest absolute loading on the second factor, and so on. The order of the variables in the output data set is not affected. The factors are not reordered.

RESIDUALS

RES

displays the residual correlation matrix and the associated partial correlation matrix. The diagonal elements of the residual correlation matrix are the unique variances.

RITER=*n*

specifies the maximum number of cycles *n* for factor rotation. Except for promax and Procrustes, you can use the RITER= option with all rotation methods. The default *n* is the maximum between 10 times the number of variables and 100.

ROTATE=*name*

R=*name*

specifies the rotation method. The default is ROTATE=NONE.

Valid *name*s for orthogonal rotations are as follows:

BIQUARTIMAX | BIQMAX specifies orthogonal biquartimax rotation. This corresponds to the specification ROTATE=ORTHOMAX(.5).

EQUAMAX | E specifies orthogonal equamax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=*number of factors*/2.

FACTORPARSIMAX | FPA specifies orthogonal factor parsimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=*number of variables*.

NONE | N specifies that no rotation be performed, leaving the original orthogonal solution.

ORTHCF(*p1*,*p2*) | ORCF(*p1*,*p2*) specifies the orthogonal Crawford-Ferguson rotation with the weights *p1* and *p2* for variable parsimony and factor parsimony, respectively. See the definitions of weights in the section "Simplicity Functions for Rotations" on page 1572.

ORTHGENCF(*p1*,*p2*,*p3*,*p4*) | ORGENCF(*p1*,*p2*,*p3*,*p4*) specifies the orthogonal generalized Crawford-Ferguson rotation with the four weights *p1*, *p2*, *p3*, and *p4*. See the definitions of weights in the section "Simplicity Functions for Rotations" on page 1572.

ORTHOMAX<(*p*)> | ORMAX<(*p*)> specifies the orthomax rotation with orthomax weight *p*. If ROTATE=ORTHOMAX is used, the default *p* value is 1 unless specified otherwise in the GAMMA= option. Alternatively, ROTATE=ORTHOMAX(*p*) specifies *p* as the orthomax weight or the GAMMA= value. See the definition of the orthomax weight in the section "Simplicity Functions for Rotations" on page 1572.

PARSIMAX | PA specifies orthogonal parsimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with

$$\text{GAMMA} = \frac{nvar \times (nfact - 1)}{nvar + nfact - 2}$$

where *nvar* is the number of variables and *nfact* is the number of factors.

QUARTIMAX | QMAX | Q specifies orthogonal quartimax rotation. This corresponds to the specification ROTATE=ORTHOMAX(0).

VARIMAX | V specifies orthogonal varimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=1.

Valid *name*s for oblique rotations are as follows:

BIQUARTIMIN | BIQMIN specifies biquartimin rotation. It corresponds to the specification ROTATE=OBLIMIN(.5) or ROTATE=OBLIMIN with TAU=0.5.

COVARIMIN | CVMIN specifies covarimin rotation. It corresponds to the specification ROTATE=OBLIMIN(1) or ROTATE=OBLIMIN with TAU=1.

HK<(*p*)> | H<(*p*)> specifies Harris-Kaiser case II orthoblique rotation. When specifying this option, you can use the HKPOWER= option to set the power of the square roots of the eigenvalues by which the eigenvectors are scaled, assuming that the factors are extracted by the principal factor method. For other extraction methods, the unrotated factor pattern is column normalized. The power is then applied to the column normalizing constants, instead of the eigenvalues. You can also use ROTATE=HK(*p*), with *p* representing the HKPOWER= value. The default associated orthogonal rotation with ROTATE=HK is the varimax rotation without Kaiser normalization. You might associate the Harris-Kaiser with other orthogonal rotations by using the ROTATE= option together with the HKPOWER= option.

OBBIQUARTIMAX | OBIQMAX specifies oblique biquartimax rotation.

OBEQUAMAX | OE specifies oblique equamax rotation.

OBFACTORPARSIMAX | OFPA specifies oblique factor parsimax rotation.

OBLICF(*p1*,*p2*) | OBCF(*p1*,*p2*) specifies the oblique Crawford-Ferguson rotation with the weights *p1* and *p2* for variable parsimony and factor parsimony, respectively. See the definitions of weights in the section "Simplicity Functions for Rotations" on page 1572.

OBLIGENCF(*p1*,*p2*,*p3*,*p4*) | OBGENCF(*p1*,*p2*,*p3*,*p4*) specifies the oblique generalized Crawford-Ferguson rotation with the four weights *p1*, *p2*, *p3*, and *p4*. See the definitions of weights in the section "Simplicity Functions for Rotations" on page 1572.

OBLIMIN<(*p*)> | OBMIN<(*p*)> specifies the oblimin rotation with oblimin weight *p*. If ROTATE=OBLIMIN is used, the default *p* value is zero unless specified otherwise in the TAU= option. Alternatively, ROTATE=OBLIMIN(*p*) specifies *p* as the oblimin weight or the TAU= value. See the definition of

the oblimin weight in the section "Simplicity Functions for Rotations" on page 1572.

OBPARSIMAX | OPA specifies oblique parsimax rotation.

OBQUARTIMAX | OQMAX specifies oblique quartimax rotation. This is the same as the QUARTIMIN method.

OBVARIMAX | OV specifies oblique varimax rotation.

PROCRUSTES specifies oblique Procrustes rotation with the target pattern provided by the TARGET= data set. The unrestricted least squares method is used with factors scaled to unit variance after rotation.

PROMAX<(*p*)> | P<(*p*)> specifies oblique promax rotation. You can use the PREROTATE= option to set the desirable prerotation method, orthogonal or oblique. When used with ROTATE=PROMAX, the POWER= option lets you specify the power for forming the target. You can also use ROTATE=PROMAX(*p*), where *p* represents the POWER= value.

QUARTIMIN | QMIN specifies quartimin rotation. It is the same as the oblique quartimax method. It also corresponds to the specification ROTATE=OBLIMIN(0) or ROTATE=OBLIMIN with TAU=0.

ROUND

prints correlation and loading matrices with entries multiplied by 100 and rounded to the nearest integer. The exact values can be obtained from the OUTSTAT= and ODS output data sets. The ROUND option also flags absolute values larger than the FLAG= value with an asterisk in correlation and loading matrices (see the FLAG= option). If the FLAG= option is not specified, the root mean square of all the values in the matrix printed is used as the default FLAG= value. The ROUND option has no effect when standard errors or confidence intervals are also printed.

SCORE

displays the factor scoring coefficients. The squared multiple correlation of each factor with the variables is also displayed except in the case of unrotated principal components. The SCORE option also outputs the factor scoring coefficients in the _TYPE_=SCORE or _TYPE_=USCORE observations in the OUTSTAT= data set. Unless you specify the NOINT option in PROC FACTOR, the scoring coefficients should be applied to standardized variables—variables that are centered by subtracting the original variable means and then divided by the original variable standard deviations. With the NOINT option, the scoring coefficients should be applied to data without centering.

SCREE

displays a scree plot of the eigenvalues (Cattell 1966, 1978; Cattell and Vogelman 1977; Horn and Engstrom 1979). This option produces printer plots. High-quality scree plots can be requested with the PLOTS=SCREE option.

SE

STDERR

computes standard errors for various classes of unrotated and rotated solutions under the maximum likelihood estimation.

SIMPLE
S

displays means, standard deviations, and the number of observations.

SINGULAR=*p*
SING=*p*

specifies the singularity criterion, where $0 < p < 1$. The default value is 1E−8.

TARGET=*SAS-data-set*

specifies an input data set containing the target pattern for Procrustes rotation (see the description of the ROTATE= option). The TARGET= data set must contain variables with the same names as those being factored. Each observation in the TARGET= data set becomes one column of the target factor pattern. Missing values are treated as zeros. The _NAME_ and _TYPE_ variables are not required and are ignored if present.

TAU=*p*

specifies the oblimin weight used with the option ROTATE=OBLIMIN or PREROTATE=OBLIMIN. Alternatively, you can use ROTATE=OBLIMIN(*p*) with *p* representing the oblimin weight. There is no restriction on valid values for the oblimin weight, although for practical purposes a negative or zero value is recommended. The default TAU= value is 0, resulting in the quartimin rotation. See the section "Simplicity Functions for Rotations" on page 1572 for more details.

ULTRAHEYWOOD
ULTRA

allows communalities to exceed 1. The ULTRAHEYWOOD option can cause convergence problems because communalities can become extremely large, and ill-conditioned Hessians might occur. See the section "Heywood Cases and Other Anomalies about Communality Estimates" on page 1576 for a discussion of Heywood cases.

VARDEF=DF | N | WDF | WEIGHT | WGT

specifies the divisor used in the calculation of variances and covariances. The default value is VARDEF=DF. The values and associated divisors are displayed in the following table where $i = 0$ if the NOINT option is used and $i = 1$ otherwise, and where k is the number of partial variables specified in the PARTIAL statement.

Value	Description	Divisor
DF	degrees of freedom	$n - k - i$
N	number of observations	$n - k$
WDF	sum of weights DF	$\sum_i w_i - k - i$
WEIGHT | WGT	sum of weights	$\sum_i w_i - k$

WEIGHT

factors a weighted correlation or covariance matrix. The WEIGHT option can be used only with the METHOD=PRINCIPAL, METHOD=PRINIT, METHOD=ULS, or METHOD=IMAGE option. The input data set must be of type CORR, UCORR, COV, UCOV, or FACTOR, and the variable weights are obtained from an observation with _TYPE_='WEIGHT'.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC FACTOR to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC FACTOR. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

If you specify the TARGET= option and the TARGET= data set does not contain any of the BY variables, then the entire TARGET= data set is used as a Procrustean target for each BY group in the DATA= data set.

If the TARGET= data set contains some but not all of the BY variables, or if some BY variables do not have the same type or length in the TARGET= data set as in the DATA= data set, then PROC FACTOR displays an error message and stops.

If all the BY variables appear in the TARGET= data set with the same type and length as in the DATA= data set, then each BY group in the TARGET= data set is used as a Procrustean target for the corresponding BY group in the DATA= data set. The BY groups in the TARGET= data set must be in the same order as in the DATA= data set. If you specify the NOTSORTED option in the BY statement, there must be identical BY groups in the same order in both data sets. If you do not specify the NOTSORTED option, some BY groups can appear in one data set but not in the other.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

FREQ Statement

FREQ *variable* **;**

If a variable in the data set represents the frequency of occurrence for the other values in the observation, include the variable's name in a FREQ statement. The procedure then treats the data set as if each observation appears n times, where n is the value of the FREQ variable for the observation. The total number of observations is considered to be equal to the sum of the FREQ variable when the procedure determines degrees of freedom for significance probabilities.

If the value of the FREQ variable is missing or is less than one, the observation is not used in the analysis. If the value is not an integer, the value is truncated to an integer.

The WEIGHT and FREQ statements have a similar effect, except in determining the number of observations for significance tests.

PARTIAL Statement

PARTIAL *variables* **;**

If you want the analysis to be based on a partial correlation or covariance matrix, use the PARTIAL statement to list the variables that are used to partial out the variables in the analysis.

PRIORS Statement

PRIORS *communalities* **;**

The PRIORS statement specifies numeric values between 0.0 and 1.0 for the prior communality estimates for each variable. The first numeric value corresponds to the first variable in the VAR statement, the second value to the second variable, and so on. The number of numeric values must equal the number of variables. For example:

```
proc factor;
   var     x  y  z;
   priors .7 .8 .9;
run;
```

You can specify various methods for computing prior communality estimates with the PRIORS= option in the PROC FACTOR statement. Refer to the description of that option for more information about the default prior communality estimates.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the numeric variables to be analyzed. If the VAR statement is omitted, all numeric variables not specified in other statements are analyzed.

WEIGHT Statement

WEIGHT *variable* **;**

If you want to use relative weights for each observation in the input data set, specify a variable containing weights in a WEIGHT statement. This is often done when the variance associated with each observation is different and the values of the weight variable are proportional to the reciprocals of the variances. If a variable value is negative or is missing, it is excluded from the analysis.

Details: FACTOR Procedure

Input Data Set

The FACTOR procedure can read an ordinary SAS data set containing raw data or a special data set specified as a TYPE=CORR, TYPE=UCORR, TYPE=SSCP, TYPE=COV, TYPE=UCOV, or TYPE=FACTOR data set containing previously computed statistics. A TYPE=CORR data set can be created by the CORR procedure or various other procedures such as the PRINCOMP procedure. It contains means, standard deviations, the sample size, the correlation matrix, and possibly other statistics if it is created by some procedure other than PROC CORR. A TYPE=COV data set is similar to a TYPE=CORR data set but contains a covariance matrix. A TYPE=UCORR or TYPE=UCOV data set contains a correlation or covariance matrix that is not corrected for the mean. The default VAR variable list does not include Intercept if the DATA= data set is TYPE=SSCP. If the Intercept variable is explicitly specified in the VAR statement with a TYPE=SSCP data set, the NOINT option is activated. A TYPE=FACTOR data set can be created by the FACTOR procedure and is described in the section "Output Data Sets" on page 1568.

If your data set has many observations and you plan to run FACTOR several times, you can save computer time by first creating a TYPE=CORR data set and using it as input to PROC FACTOR, as in the following statements:

```
proc corr data=raw out=correl;      /* create TYPE=CORR data set */
proc factor data=correl method=ml; /* maximum likelihood        */
proc factor data=correl;           /* principal components      */
```

The data set created by the CORR procedure is automatically given the TYPE=CORR data set option, so you do not have to specify TYPE=CORR. However, if you use a DATA step with a SET statement to modify the correlation data set, you must use the TYPE=CORR attribute in the new data set. You can use a VAR statement with PROC FACTOR when reading a TYPE=CORR data set to select a subset of the variables or change the order of the variables.

Problems can arise from using the CORR procedure when there are missing data. By default, PROC CORR computes each correlation from all observations that have values present for the pair of variables involved (pairwise deletion). The resulting correlation matrix might have negative

eigenvalues. If you specify the NOMISS option with the CORR procedure, observations with any missing values are completely omitted from the calculations (listwise deletion), and there is no danger of negative eigenvalues.

PROC FACTOR can also create a TYPE=FACTOR data set, which includes all the information in a TYPE=CORR data set, and use it for repeated analyses. For a TYPE=FACTOR data set, the default value of the METHOD= option is PATTERN. The following PROC FACTOR statements produce the same results as the previous example:

```
proc factor data=raw method=ml outstat=fact; /* max. likelihood */
proc factor data=fact method=prin;      /* principal components */
```

You can use a TYPE=FACTOR data set to try several different rotation methods on the same data without repeatedly extracting the factors. In the following example, the second and third PROC FACTOR statements use the data set fact created by the first PROC FACTOR statement:

```
proc factor data=raw outstat=fact; /* principal components */
proc factor rotate=varimax;        /* varimax rotation     */
proc factor rotate=quartimax;      /* quartimax rotation   */
```

You can create a TYPE=CORR, TYPE=UCORR, or TYPE=FACTOR data set in a DATA step for PROC FACTOR to read as input. For example, in the following a TYPE=CORR data set is created and is read as input data set by the subsequent PROC FACTOR statement:

```
data correl(type=corr);
   _TYPE_='CORR';
   input _NAME_ $ x y z;
   datalines;
x  1.0  .   .
y   .7 1.0  .
z   .5  .4 1.0
;
proc factor;
run;
```

Be sure to specify the TYPE= option in parentheses after the data set name in the DATA statement and include the _TYPE_ and _NAME_ variables. In a TYPE=CORR data set, only the correlation matrix (_TYPE_='CORR') is necessary. It can contain missing values as long as every pair of variables has at least one nonmissing value.

You can also create a TYPE=FACTOR data set containing only a factor pattern (_TYPE_='PATTERN') and use the FACTOR procedure to rotate it, as these statements show:

```
data pat(type=factor);
   _TYPE_='PATTERN';
   input _NAME_ $ x y z;
   datalines;
factor1  .5  .7  .3
factor2  .8  .2  .8
;
proc factor rotate=promax prerotate=none;
run;
```

If the input factors are oblique, you must also include the interfactor correlation matrix with _TYPE_='FCORR', as shown here:

```
data pat(type=factor);
   input _TYPE_ $ _NAME_ $ x y z;
   datalines;
pattern factor1  .5  .7  .3
pattern factor2  .8  .2  .8
fcorr   factor1 1.0  .2  .
fcorr   factor2  .2 1.0  .
;
proc factor rotate=promax prerotate=none;
run;
```

Some procedures, such as the PRINCOMP and CANDISC procedures, produce TYPE=CORR or TYPE=UCORR data sets containing scoring coefficients (_TYPE_='SCORE' or _TYPE_='USCORE'). These coefficients can be input to PROC FACTOR and rotated by using the METHOD=SCORE option, as in the following statements:

```
proc princomp data=raw n=2 outstat=prin;
run;
proc factor data=prin method=score rotate=varimax;
run;
```

Notice that the input data set prin must contain the correlation matrix as well as the scoring coefficients.

Output Data Sets

The OUT= Data Set

The OUT= data set contains all the data in the DATA= data set plus new variables called Factor1, Factor2, and so on, containing estimated factor scores. Each estimated factor score is computed as a linear combination of the standardized values of the variables that are factored. The coefficients are always displayed if the OUT= option is specified, and they are labeled "Standardized Scoring Coefficients."

If partial variables are specified in the PARTIAL statement, the factor analysis is on the residuals of the variables, which are regressed on the partial variables. In this case, the OUT= data set also contains the (unstandardized) residuals, which are prefixed by R_ by default. For example, the residual of variable X is named R_X in the OUT= data set. You might also assign the prefix by the PARPREFIX= option. Because the residuals are factor-analyzed, the estimated factor scores are computed as linear combinations of the standardized values of the residuals, but not the original variables.

The OUTSTAT= Data Set

The OUTSTAT= data set is similar to the TYPE=CORR or TYPE=UCORR data set produced by the CORR procedure, but it is a TYPE=FACTOR data set and it contains many results in addition to those produced by PROC CORR. The OUTSTAT= data set contains observations with _TYPE_='UCORR' and _TYPE_='USTD' if you specify the NOINT option.

The output data set contains the following variables:

- the BY variables, if any
- two new character variables, _TYPE_ and _NAME_
- the variables analyzed—those in the VAR statement, or, if there is no VAR statement, all numeric variables not listed in any other statement. If partial variables are specified in the PARTIAL statement, the residuals are included instead. By default, the residual variable names are prefixed by R_, unless you specify something different in the PARPREFIX= option.

Each observation in the output data set contains some type of statistic as indicated by the _TYPE_ variable. The _NAME_ variable is blank except where otherwise indicated. The values of the _TYPE_ variable are as follows:

MEAN	means
STD	standard deviations
USTD	uncorrected standard deviations
N	sample size
CORR	correlations. The _NAME_ variable contains the name of the variable corresponding to each row of the correlation matrix.
UCORR	uncorrected correlations. The _NAME_ variable contains the name of the variable corresponding to each row of the uncorrected correlation matrix.
IMAGE	image coefficients. The _NAME_ variable contains the name of the variable corresponding to each row of the image coefficient matrix.
IMAGECOV	image covariance matrix. The _NAME_ variable contains the name of the variable corresponding to each row of the image covariance matrix.
COMMUNAL	final communality estimates
PRIORS	prior communality estimates, or estimates from the last iteration for iterative methods
WEIGHT	variable weights
SUMWGT	sum of the variable weights
EIGENVAL	eigenvalues
UNROTATE	unrotated factor pattern. The _NAME_ variable contains the name of the factor.
SE_UNROT	standard error estimates for the unrotated loadings. The _NAME_ variable contains the name of the factor.

RESIDUAL	residual correlations. The _NAME_ variable contains the name of the variable corresponding to each row of the residual correlation matrix.
PRETRANS	transformation matrix from prerotation. The _NAME_ variable contains the name of the factor.
PREFCORR	prerotated interfactor correlations. The _NAME_ variable contains the name of the factor.
SE_PREFC	standard error estimates for prerotated interfactor correlations. The _NAME_ variable contains the name of the factor.
PREROTAT	prerotated factor pattern. The _NAME_ variable contains the name of the factor.
SE_PREPA	standard error estimates for the prerotated loadings. The _NAME_ variable contains the name of the factor.
PRERCORR	prerotated reference axis correlations. The _NAME_ variable contains the name of the factor.
PREREFER	prerotated reference structure. The _NAME_ variable contains the name of the factor.
PRESTRUC	prerotated factor structure. The _NAME_ variable contains the name of the factor.
SE_PREST	standard error estimates for prerotated structure loadings. The _NAME_ variable contains the name of the factor.
PRESCORE	prerotated scoring coefficients. The _NAME_ variable contains the name of the factor.
TRANSFOR	transformation matrix from rotation. The _NAME_ variable contains the name of the factor.
FCORR	interfactor correlations. The _NAME_ variable contains the name of the factor.
SE_FCORR	standard error estimates for interfactor correlations. The _NAME_ variable contains the name of the factor.
PATTERN	factor pattern. The _NAME_ variable contains the name of the factor.
SE_PAT	standard error estimates for the rotated loadings. The _NAME_ variable contains the name of the factor.
RCORR	reference axis correlations. The _NAME_ variable contains the name of the factor.
REFERENC	reference structure. The _NAME_ variable contains the name of the factor.
STRUCTUR	factor structure. The _NAME_ variable contains the name of the factor.
SE_STRUC	standard error estimates for structure loadings. The _NAME_ variable contains the name of the factor.
SCORE	scoring coefficients to be applied to standardized variables if the SCORE option is specified on the PROC FACTOR statement. The _NAME_ variable contains the name of the factor.
USCORE	scoring coefficients to be applied without subtracting the mean from the raw variables if the SCORE option is specified on the PROC FACTOR statement. The _NAME_ variable contains the name of the factor.

Confidence Intervals and the Salience of Factor Loadings

The traditional approach to determining salient loadings (loadings that are considered large in absolute values) employs rules of thumb such as 0.3 or 0.4. However, this does not use the statistical evidence efficiently. The asymptotic normality of the distribution of factor loadings enables you to construct confidence intervals to gauge the salience of factor loadings. To guarantee the range-respecting properties of confidence intervals, a transformation procedure such as in CEFA (Browne et al. 2004) is used. For example, because the orthogonal rotated factor loading θ must be bounded between -1 and $+1$, the Fisher transformation

$$\varphi = \frac{1}{2}\log(\frac{1+\theta}{1-\theta})$$

is employed so that φ is an unbounded parameter. Assuming the asymptotic normality of $\hat{\varphi}$, a symmetric confidence interval for φ is constructed. Then, a back-transformation on the confidence limits yields an asymmetric confidence interval for θ. Applying the results of Browne (1982), a $(1-\alpha)100\%$ confidence interval for the orthogonal factor loading θ is

$$(\hat{\theta}_l = \frac{a/b-1}{a/b+1},\ \hat{\theta}_u = \frac{a\times b-1}{a\times b+1})$$

where

$$a = \frac{1+\hat{\theta}}{1-\hat{\theta}},\quad b = \exp(z_{\alpha/2}\times\frac{2\hat{\sigma}}{1-\hat{\theta}^2})$$

and $\hat{\theta}$ is the estimated factor loading, $\hat{\sigma}$ is the standard error estimate of the factor loading, and $z_{\alpha/2}$ is the $(1-\alpha/2)100$ percentile point of a standard normal distribution.

Once the confidence limits are constructed, you can use the corresponding coverage displays for determining the salience of the variable-factor relationship. In a coverage display, the COVER= value is represented by an asterisk (*). The following table summarizes various displays and their interpretations.

Table 33.2 Interpretations of the Coverage Displays

Positive Estimate	Negative Estimate	COVER=0 Specified	Interpretation
[0]*	*[0]		The estimate is not significantly different from zero, and the CI covers a region of values that are smaller in magnitude than the COVER= value. This is strong statistical evidence for the nonsalience of the variable-factor relationship.
0[ ]*	*[ ]0		The estimate is significantly different from zero, but the CI covers a region of values that are smaller in magnitude than the COVER= value. This is strong statistical evidence for the nonsalience of the variable-factor relationship.
[0*]	[*0]	[0]	The estimate is not significantly different from zero or the COVER= value. The population value might have been larger or smaller in magnitude than the COVER= value. There is no statistical evidence for the salience of the variable-factor relationship.
0[*]	[*]0		The estimate is significantly different from zero but not from the COVER= value. This is marginal statistical evidence for the salience of the variable-factor relationship.
0*[ ]	[ ]*0	0[ ] or [ ]0	The estimate is significantly different from zero, and the CI covers a region of values that are larger in magnitude than the COVER= value. This is strong statistical evidence for the salience of the variable-factor relationship.

See Example 33.4 for an illustration of the use of confidence intervals for interpreting factors.

Simplicity Functions for Rotations

To rotate a factor pattern is to apply a nonsingular linear transformation to the unrotated factor pattern matrix. An optimal transformation is usually defined as a minimum or maximum point of a simplicity function. Different rotation methods are based on different simplicity functions employed.

For the promax or the Procrustes rotation, the simplicity function used is the sum of squared differences between the rotated factor pattern and the target matrix. The optimal transformation is obtained by minimizing this simplicity function with respect to the choices of all possible transformation.

For the class of the generalized Crawford-Ferguson family (Jennrich 1973), the simplicity function being optimized is

$$f = k_1 Z + k_2 H + k_3 V + k_4 Q$$

where

$$Z = (\sum_j \sum_i b_{ij}^2)^2, \quad H = \sum_i (\sum_j b_{ij}^2)^2$$
$$V = \sum_j (\sum_i b_{ij}^2)^2, \quad Q = \sum_j \sum_i b_{ij}^4$$

k_1, k_2, k_3, and k_4 are constants, and b_{ij} represents an element of the rotated pattern matrix. Except for specialized research purposes, it is uncommon in practice to use this simplicity function directly for rotation. However, this simplicity function reduces to many well-known classes of rotations. One of these is the Crawford-Ferguson family (Crawford and Ferguson 1970), which minimizes

$$f_{cf} = c_1(H - Q) + c_2(V - Q)$$

where c_1 and c_2 are constants, $(H-Q)$ represents variable (row) parsimony, and $(V-Q)$ represents factor (column) parsimony. Therefore, the relative importance of both the variable parsimony and of the factor parsimony is adjusted using the constants c_1 and c_2. The orthomax class (see Harman 1976) maximizes the function

$$f_{or} = pQ - \gamma V$$

where γ is the orthomax weight and is usually between 0 and the number of variables p. The oblimin class minimizes the function

$$f_{ob} = p(H - Q) - \tau(Z - V)$$

where τ is the oblimin weight. For practical purposes, a negative or zero value for τ is recommended.

All of the preceding definitions are for rotations without row normalization. For rotations with Kaiser normalization, the definition of b_{ij} is replaced by b_{ij}/h_i, where h_i is the communality estimate of variable i.

Missing Values

If the DATA= data set contains data (rather than a matrix or factor pattern), then observations with missing values for any variables in the analysis are omitted from the computations. If a correlation or covariance matrix is read, it can contain missing values as long as every pair of variables has at least one nonmissing entry. Missing values in a pattern or scoring coefficient matrix are treated as zeros.

Cautions

- The amount of time that FACTOR takes is roughly proportional to the cube of the number of variables. Factoring 100 variables, therefore, takes about 1,000 times as long as factoring 10 variables. Iterative methods (PRINIT, ALPHA, ULS, ML) can also take 100 times as long as noniterative methods (PRINCIPAL, IMAGE, HARRIS).
- No computer program is capable of reliably determining the optimal number of factors, since the decision is ultimately subjective. You should not blindly accept the number of factors obtained by default; instead, use your own judgment to make a decision.
- Singular correlation matrices cause problems with the options PRIORS=SMC and METHOD=ML. Singularities can result from using a variable that is the sum of other variables, coding too many dummy variables from a classification variable, or having more variables than observations.
- If you use the CORR procedure to compute the correlation matrix and there are missing data and the NOMISS option is not specified, then the correlation matrix might have negative eigenvalues.
- If a TYPE=CORR, TYPE=UCORR, or TYPE=FACTOR data set is copied or modified using a DATA step, the new data set does not automatically have the same TYPE as the old data set. You must specify the TYPE= data set option in the DATA statement. If you try to analyze a data set that has lost its TYPE=CORR attribute, PROC FACTOR displays a warning message saying that the data set contains _NAME_ and _TYPE_ variables but analyzes the data set as an ordinary SAS data set.
- For a TYPE=FACTOR data set, the default is METHOD=PATTERN, not METHOD=PRIN.

Factor Scores

The FACTOR procedure can compute estimated factor scores directly if you specify the NFACTORS= and OUT= options, or indirectly using the SCORE procedure. The latter method is preferable if you use the FACTOR procedure interactively to determine the number of factors, the rotation method, or various other aspects of the analysis. To compute factor scores for each observation by using the SCORE procedure, do the following:

- Use the SCORE option in the PROC FACTOR statement.
- Create a TYPE=FACTOR output data set with the OUTSTAT= option.
- Use the SCORE procedure with both the raw data and the TYPE=FACTOR data set.
- Do not use the TYPE= option in the PROC SCORE statement.

For example, the following statements could be used:

```
proc factor data=raw score outstat=fact;
run;
proc score data=raw score=fact out=scores;
run;
```

or

```
proc corr   data=raw out=correl;
run;
proc factor data=correl score outstat=fact;
run;
proc score  data=raw score=fact out=scores;
run;
```

For a more detailed example, see Example 76.1 in Chapter 76, "The SCORE Procedure."

A component analysis (principal, image, or Harris) produces scores with mean zero and variance one. If you have done a common factor analysis, the true factor scores have mean zero and variance one, but the computed factor scores are only estimates of the true factor scores. These estimates have mean zero but variance equal to the squared multiple correlation of the factor with the variables. The estimated factor scores might have small nonzero correlations even if the true factors are uncorrelated.

Variable Weights and Variance Explained

A principal component analysis of a correlation matrix treats all variables as equally important. A principal component analysis of a covariance matrix gives more weight to variables with larger variances. A principal component analysis of a covariance matrix is equivalent to an analysis of a weighted correlation matrix, where the weight of each variable is equal to its variance. Variables with large weights tend to have larger loadings on the first component and smaller residual correlations than variables with small weights.

You might want to give weights to variables by using values other than their variances. Mulaik (1972) explains how to obtain a maximally reliable component by means of a weighted principal component analysis. With the FACTOR procedure, you can indirectly give arbitrary weights to the variables by using the COV option and rescaling the variables to have variance equal to the desired weight, or you can give arbitrary weights directly by using the WEIGHT option and including the weights in a TYPE=CORR data set.

Arbitrary variable weights can be used with the METHOD=PRINCIPAL, METHOD=PRINIT, METHOD=ULS, or METHOD=IMAGE option. Alpha and ML factor analyses compute variable weights based on the communalities (Harman 1976, pp. 217–218). For alpha factor analysis, the weight of a variable is the reciprocal of its communality. In ML factor analysis, the weight is the reciprocal of the uniqueness. Harris component analysis uses weights equal to the reciprocal of one minus the squared multiple correlation of each variable with the other variables.

For uncorrelated factors, the variance explained by a factor can be computed with or without taking the weights into account. The usual method for computing variance accounted for by a factor is to take the sum of squares of the corresponding column of the factor pattern, yielding an unweighted result. If the square of each loading is multiplied by the weight of the variable before the sum is taken, the result is the weighted variance explained, which is equal to the corresponding eigenvalue except in image analysis. Whether the weighted or unweighted result is more important depends on the purpose of the analysis.

In the case of correlated factors, the variance explained by a factor can be computed with or without taking the other factors into account. If you want to ignore the other factors, the variance explained is given by the weighted or unweighted sum of squares of the appropriate column of the factor structure since the factor structure contains simple correlations. If you want to subtract the variance explained by the other factors from the amount explained by the factor in question (the Type II variance explained), you can take the weighted or unweighted sum of squares of the appropriate column of the reference structure because the reference structure contains semipartial correlations. There are other ways of measuring the variance explained. For example, given a prior ordering of the factors, you can eliminate from each factor the variance explained by previous factors and compute a Type I variance explained. Harman (1976, pp. 268–270) provides another method, which is based on direct and joint contributions.

Heywood Cases and Other Anomalies about Communality Estimates

Since communalities are squared correlations, you would expect them always to lie between 0 and 1. It is a mathematical peculiarity of the common factor model, however, that final communality estimates might exceed 1. If a communality equals 1, the situation is referred to as a Heywood case, and if a communality exceeds 1, it is an ultra-Heywood case. An ultra-Heywood case implies that some unique factor has negative variance, a clear indication that something is wrong. Possible causes include the following:

- bad prior communality estimates
- too many common factors
- too few common factors
- not enough data to provide stable estimates
- the common factor model is not an appropriate model for the data

An ultra-Heywood case renders a factor solution invalid. Factor analysts disagree about whether or not a factor solution with a Heywood case can be considered legitimate.

With METHOD=PRINIT, METHOD=ULS, METHOD=ALPHA, or METHOD=ML, the FACTOR procedure, by default, stops iterating and sets the number of factors to 0 if an estimated communality exceeds 1. To enable processing to continue with a Heywood or ultra-Heywood case, you can use the HEYWOOD or ULTRAHEYWOOD option in the PROC FACTOR statement. The HEYWOOD

option sets the upper bound of any communality to 1, while the ULTRAHEYWOOD option allows communalities to exceed 1.

Theoretically, the communality of a variable should not exceed its reliability. Violation of this condition is called a quasi-Heywood case and should be regarded with the same suspicion as an ultra-Heywood case.

Elements of the factor structure and reference structure matrices can exceed 1 only in the presence of an ultra-Heywood case. On the other hand, an element of the factor pattern might exceed 1 in an oblique rotation.

The maximum likelihood method is especially susceptible to quasi- or ultra-Heywood cases. During the iteration process, a variable with high communality is given a high weight; this tends to increase its communality, which increases its weight, and so on.

It is often stated that the squared multiple correlation of a variable with the other variables is a lower bound to its communality. This is true if the common factor model fits the data perfectly, but it is not generally the case with real data. A final communality estimate that is less than the squared multiple correlation can, therefore, indicate poor fit, possibly due to not enough factors. It is by no means as serious a problem as an ultra-Heywood case. Factor methods that use the Newton-Raphson method can actually produce communalities less than 0, a result even more disastrous than an ultra-Heywood case.

The squared multiple correlation of a factor with the variables might exceed 1, even in the absence of ultra-Heywood cases. This situation is also cause for alarm. Alpha factor analysis seems to be especially prone to this problem, but it does not occur with maximum likelihood. If a squared multiple correlation is negative, there are too many factors retained.

With data that do not fit the common factor model perfectly, you can expect some of the eigenvalues to be negative. If an iterative factor method converges properly, the sum of the eigenvalues corresponding to rejected factors should be 0; hence, some eigenvalues are positive and some negative. If a principal factor analysis fails to yield any negative eigenvalues, the prior communality estimates are probably too large. Negative eigenvalues cause the cumulative proportion of variance explained to exceed 1 for a sufficiently large number of factors. The cumulative proportion of variance explained by the retained factors should be approximately 1 for principal factor analysis and should converge to 1 for iterative methods. Occasionally, a single factor can explain more than 100 percent of the common variance in a principal factor analysis, indicating that the prior communality estimates are too low.

If a squared canonical correlation or a coefficient alpha is negative, there are too many factors retained.

Principal component analysis, unlike common factor analysis, has none of these problems if the covariance or correlation matrix is computed correctly from a data set with no missing values. Various methods for missing value correlation or severe rounding of the correlations can produce negative eigenvalues in principal components.

Time Requirements

n = number of observations
v = number of variables
f = number of factors
i = number of iterations during factor extraction
r = length of iterations during factor rotation

The time required to compute...	is roughly proportional to
an overall factor analysis	iv^3
the correlation matrix	nv^2
PRIORS=SMC or ASMC	v^3
PRIORS=MAX	v^2
eigenvalues	v^3
final eigenvectors	fv^2
generalized Crawford-Ferguson family of rotations, PROMAX, or HK	rvf^2
ROTATE=PROCRUSTES	vf^2

Each iteration in the PRINIT or ALPHA method requires computation of eigenvalues and f eigenvectors.

Each iteration in the ML or ULS method requires computation of eigenvalues and $v - f$ eigenvectors.

The amount of time that PROC FACTOR takes is roughly proportional to the cube of the number of variables. Factoring 100 variables, therefore, takes about 1000 times as long as factoring 10 variables. Iterative methods (PRINIT, ALPHA, ULS, ML) can also take 100 times as long as noniterative methods (PRINCIPAL, IMAGE, HARRIS).

Displayed Output

PROC FACTOR output includes the following.

- Mean and Std Dev (standard deviation) of each variable and the number of observations, if you specify the SIMPLE option
- Correlations, if you specify the CORR option
- Inverse Correlation Matrix, if you specify the ALL option
- Partial Correlations Controlling all other Variables (negative anti-image correlations), if you specify the MSA option. If the data are appropriate for the common factor model, the partial correlations should be small.
- Kaiser's Measure of Sampling Adequacy (Kaiser 1970; Kaiser and Rice 1974; Cerny and Kaiser 1977) both overall and for each variable, if you specify the MSA option. The MSA is a summary of how small the partial correlations are relative to the ordinary correlations. Values greater than 0.8 can be considered good. Values less than 0.5 require remedial action, either by deleting the offending variables or by including other variables related to the offenders.
- Prior Communality Estimates, unless 1.0s are used or unless you specify the METHOD=IMAGE, METHOD=HARRIS, METHOD=PATTERN, or METHOD=SCORE option
- Squared Multiple Correlations of each variable with all the other variables, if you specify the METHOD=IMAGE or METHOD=HARRIS option
- Image Coefficients, if you specify the METHOD=IMAGE option
- Image Covariance Matrix, if you specify the METHOD=IMAGE option
- Preliminary Eigenvalues based on the prior communalities, if you specify the METHOD=PRINIT, METHOD=ALPHA, METHOD=ML, or METHOD=ULS option. The table produced includes the Total and the Average of the eigenvalues, the Difference between successive eigenvalues, the Proportion of variation represented, and the Cumulative proportion of variation.
- the number of factors that are retained, unless you specify the METHOD=PATTERN or METHOD=SCORE option
- the Scree Plot of Eigenvalues, if you specify the SCREE option. The preliminary eigenvalues are used if you specify the METHOD=PRINIT, METHOD=ALPHA, METHOD=ML, or METHOD=ULS option.
- the iteration history, if you specify the METHOD=PRINIT, METHOD=ALPHA, METHOD=ML, or METHOD=ULS option. The table produced contains the iteration number (Iter); the Criterion being optimized (Jöreskog 1977); the Ridge value for the iteration if you specify the METHOD=ML or METHOD=ULS option; the maximum Change in any communality estimate; and the Communalities.

- Significance tests, if you specify the option METHOD=ML, including Bartlett's chi-square, df, and Prob > χ^2 for H_0: No common factors and H_0: factors retained are sufficient to explain the correlations. The variables should have an approximate multivariate normal distribution for the probability levels to be valid. Lawley and Maxwell (1971) suggest that the number of observations should exceed the number of variables by 50 or more, although Geweke and Singleton (1980) claim that as few as 10 observations are adequate with five variables and one common factor. Certain regularity conditions must also be satisfied for Bartlett's χ^2 test to be valid (Geweke and Singleton 1980), but in practice these conditions usually are satisfied. The notation Prob>chi**2 means "the probability under the null hypothesis of obtaining a greater χ^2 statistic than that observed." The chi-square value is displayed with and without Bartlett's correction.

- Akaike's Information Criterion, if you specify the METHOD=ML option. Akaike's information criterion (AIC) (Akaike 1973, 1974, 1987) is a general criterion for estimating the best number of parameters to include in a model when maximum likelihood estimation is used. The number of factors that yields the smallest value of AIC is considered best. Like the chi-square test, AIC tends to include factors that are statistically significant but inconsequential for practical purposes.

- Schwarz's Bayesian Criterion, if you specify the METHOD=ML option. Schwarz's Bayesian Criterion (SBC) (Schwarz 1978) is another criterion, similar to AIC, for determining the best number of parameters. The number of factors that yields the smallest value of SBC is considered best; SBC seems to be less inclined to include trivial factors than either AIC or the chi-square test.

- Tucker and Lewis's Reliability Coefficient, if you specify the METHOD=ML option (Tucker and Lewis 1973)

- Squared Canonical Correlations, if you specify the METHOD=ML option. These are the same as the squared multiple correlations for predicting each factor from the variables.

- Coefficient Alpha for Each Factor, if you specify the METHOD=ALPHA option

- Eigenvectors, if you specify the EIGENVECTORS or ALL option, unless you also specify the METHOD=PATTERN or METHOD=SCORE option

- Eigenvalues of the (Weighted) (Reduced) (Image) Correlation or Covariance Matrix, unless you specify the METHOD=PATTERN or METHOD=SCORE option. Included are the Total and the Average of the eigenvalues, the Difference between successive eigenvalues, the Proportion of variation represented, and the Cumulative proportion of variation.

- the Factor Pattern, which is equal to both the matrix of standardized regression coefficients for predicting variables from common factors and the matrix of correlations between variables and common factors since the extracted factors are uncorrelated. Standard error estimates are included if the SE option is specified with METHOD=ML. Confidence limits and coverage displays are included if COVER= option is specified with METHOD=ML.

- Variance explained by each factor, both Weighted and Unweighted, if variable weights are used

- Final Communality Estimates, including the Total communality; or Final Communality Estimates and Variable Weights, including the Total communality, both Weighted and Unweighted, if variable weights are used. Final communality estimates are the squared multiple correlations for predicting the variables from the estimated factors, and they can be obtained by taking the sum of squares of each row of the factor pattern, or a weighted sum of squares if variable weights are used.
- Residual Correlations with Uniqueness on the Diagonal, if you specify the RESIDUAL or ALL option
- Root Mean Square Off-diagonal Residuals, both Over-all and for each variable, if you specify the RESIDUAL or ALL option
- Partial Correlations Controlling Factors, if you specify the RESIDUAL or ALL option
- Root Mean Square Off-diagonal Partials, both Over-all and for each variable, if you specify the RESIDUAL or ALL option
- Plots of Factor Pattern for unrotated factors, if you specify the PREPLOT option. The number of plots is determined by the NPLOT= option.
- Variable Weights for Rotation, if you specify the NORM=WEIGHT option
- Factor Weights for Rotation, if you specify the HKPOWER= option
- Orthogonal Transformation Matrix, if you request an orthogonal rotation
- Rotated Factor Pattern, if you request an orthogonal rotation. Standard error estimates are included if the SE option is specified with METHOD=ML. Confidence limits and coverage displays are included if COVER= option is specified with METHOD=ML.
- Variance explained by each factor after rotation. If you request an orthogonal rotation and if variable weights are used, both weighted and unweighted values are produced.
- Target Matrix for Procrustean Transformation, if you specify the ROTATE=PROMAX or ROTATE=PROCRUSTES option
- the Procrustean Transformation Matrix, if you specify the ROTATE=PROMAX or ROTATE=PROCRUSTES option
- the Normalized Oblique Transformation Matrix, if you request an oblique rotation, which, for the option ROTATE=PROMAX, is the product of the prerotation and the Procrustes rotation
- Inter-factor Correlations, if you specify an oblique rotation. Standard error estimates are included if the SE option is specified with METHOD=ML. Confidence limits and coverage displays are included if COVER= option is specified with METHOD=ML.
- Rotated Factor Pattern (Std Reg Coefs), if you specify an oblique rotation, giving standardized regression coefficients for predicting the variables from the factors. Standard error estimates are included if the SE option is specified with METHOD=ML. Confidence limits and coverage displays are included if COVER= option is specified with METHOD=ML.

- Reference Axis Correlations if you specify an oblique rotation. These are the partial correlations between the primary factors when all factors other than the two being correlated are partialed out.
- Reference Structure (Semipartial Correlations), if you request an oblique rotation. The reference structure is the matrix of semipartial correlations (Kerlinger and Pedhazur 1973) between variables and common factors, removing from each common factor the effects of other common factors. If the common factors are uncorrelated, the reference structure is equal to the factor pattern.
- Variance explained by each factor eliminating the effects of all other factors, if you specify an oblique rotation. Both Weighted and Unweighted values are produced if variable weights are used. These variances are equal to the (weighted) sum of the squared elements of the reference structure corresponding to each factor.
- Factor Structure (Correlations), if you request an oblique rotation. The (primary) factor structure is the matrix of correlations between variables and common factors. If the common factors are uncorrelated, the factor structure is equal to the factor pattern. Standard error estimates are included if the SE option is specified with METHOD=ML. Confidence limits and coverage displays are included if COVER= option is specified with METHOD=ML.
- Variance explained by each factor ignoring the effects of all other factors, if you request an oblique rotation. Both Weighted and Unweighted values are produced if variable weights are used. These variances are equal to the (weighted) sum of the squared elements of the factor structure corresponding to each factor.
- Final Communality Estimates for the rotated factors if you specify the ROTATE= option. The estimates should equal the unrotated communalities.
- Squared Multiple Correlations of the Variables with Each Factor, if you specify the SCORE or ALL option, except for unrotated principal components
- Standardized Scoring Coefficients, if you specify the SCORE or ALL option
- Plots of the Factor Pattern for rotated factors, if you specify the PLOT option and you request an orthogonal rotation. The number of plots is determined by the NPLOT= option.
- Plots of the Reference Structure for rotated factors, if you specify the PLOT option and you request an oblique rotation. The number of plots is determined by the NPLOT= option. Included are the Reference Axis Correlation and the Angle between the Reference Axes for each pair of factors plotted.

If you specify the ROTATE=PROMAX option, the output includes results for both the prerotation and the Procrustes rotation.

ODS Table Names

PROC FACTOR assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the Table 33.3. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 33.3 ODS Tables Produced by PROC FACTOR

ODS Table Name	Description	Option
AlphaCoef	Coefficient alpha for each factor	METHOD=ALPHA
CanCorr	Squared canonical correlations	METHOD=ML
CondStdDev	Conditional standard deviations	SIMPLE with PARTIAL
ConvergenceStatus	Convergence status	METHOD=PRINIT, ALPHA, ML, or ULS
Corr	Correlations	CORR
Eigenvalues	Eigenvalues	default, SCREE
Eigenvectors	Eigenvectors	EIGENVECTORS
FactorWeightRotate	Factor weights for rotation	HKPOWER=
FactorPattern	Factor pattern	default
FactorStructure	Factor structure	ROTATE= any oblique rotation
FinalCommun	Final communalities	default
FinalCommunWgt	Final communalities with weights	METHOD=ML or ALPHA
FitMeasures	Measures of fit	METHOD=ML
ImageCoef	Image coefficients	METHOD=IMAGE
ImageCov	Image covariance matrix	METHOD=IMAGE
ImageFactors	Image factor matrix	METHOD=IMAGE
InputFactorPattern	Input factor pattern	METHOD=PATTERN with PRINT or ALL
InputScoreCoef	Standardized input scoring coefficients	METHOD=SCORE with PRINT or ALL
InterFactorCorr	Interfactor correlations	ROTATE= any oblique rotation
InvCorr	Inverse correlation matrix	ALL
IterHistory	Iteration history	METHOD=PRINIT, ALPHA, ML, or ULS
MultipleCorr	Squared multiple correlations	METHOD=IMAGE or METHOD=HARRIS
NormObliqueTrans	Normalized oblique transformation matrix	ROTATE= any oblique rotation
ObliqueRotFactPat	Rotated factor pattern	ROTATE= any oblique rotation
ObliqueTrans	Oblique transformation matrix	HKPOWER=
OrthRotFactPat	Rotated factor pattern	ROTATE= any orthogonal rotation
OrthTrans	Orthogonal transformation matrix	ROTATE= any orthogonal rotation
ParCorrControlFactor	Partial correlations controlling factors	RESIDUAL

Table 33.3 *continued*

ODS Table Name	Description	Option
ParCorrControlVar	Partial correlations controlling other variables	MSA
PartialCorr	Partial correlations	MSA, CORR with PARTIAL
PriorCommunalEst	Prior communality estimates	PRIORS=, METHOD=ML or ALPHA
ProcrustesTarget	Target matrix for Procrustean transformation	ROTATE=PROCRUSTES, ROTATE=PROMAX
ProcrustesTrans	Procrustean transformation matrix	ROTATE=PROCRUSTES, ROTATE=PROMAX
RMSOffDiagPartials	Root mean square off-diagonal partials	RESIDUAL
RMSOffDiagResids	Root mean square off-diagonal residuals	RESIDUAL
ReferenceAxisCorr	Reference axis correlations	ROTATE= any oblique rotation
ReferenceStructure	Reference structure	ROTATE= any oblique rotation
ResCorrUniqueDiag	Residual correlations with uniqueness on the diagonal	RESIDUAL
SamplingAdequacy	Kaiser's measure of sampling adequacy	MSA
SignifTests	Significance tests	METHOD=ML
SimpleStatistics	Simple statistics	SIMPLE
StdScoreCoef	Standardized scoring coefficients	SCORE
VarExplain	Variance explained	default
VarExplainWgt	Variance explained with weights	METHOD=ML, or ALPHA
VarFactorCorr	Squared multiple correlations of the variables with each factor	SCORE
VarWeightRotate	Variable weights for rotation	NORM=WEIGHT, ROTATE=

ODS Graphics

To request graphics with PROC FACTOR, you must first enable ODS Graphics by specifying the **ods graphics on** statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. The names of the graphs that PROC FACTOR generates are listed in Table 33.4, along with the required statements and options.

Table 33.4 ODS Graphics Produced by PROC FACTOR

ODS Graph Name	Plot Description	Option
PatternPlot	Rotated factor pattern	PLOTS=LOADINGS
InitPatternPlot	Initial factor pattern	PLOTS=INITLOADINGS
InitRefStructurePlot	Initial reference structures	PLOTS=INITLOADINGS and PLOTREF
PrePatternPlot	Prerotated factor pattern	PLOTS=PRELOADINGS
PreRefStructurePlot	Prerotated reference structures	PLOTS=PRELOADINGS and PLOTREF
RefStructurePlot	Rotated reference structures	PLOTS=LOADINGS and PLOTREF
ScreePlot	Scree and variance explained plots	PLOTS=SCREE
VariancePlot	Plot of explained variance	PLOTS=SCREE(UNPACK)

Examples: FACTOR Procedure

Example 33.1: Principal Component Analysis

This example analyzes socioeconomic data provided by Harman (1976). The five variables represent total population, median school years, total employment, miscellaneous professional services, and median house value. Each observation represents one of twelve census tracts in the Los Angeles Standard Metropolitan Statistical Area.

The first analysis is a principal component analysis. Simple descriptive statistics and correlations are also displayed. The following statements produce Output 33.1.1:

```
data SocioEconomics;
   title 'Five Socioeconomic Variables';
   title2 'See Page 14 of Harman: Modern Factor Analysis, 3rd Ed';
   input Population School Employment Services HouseValue;
   datalines;
5700     12.8      2500      270       25000
1000     10.9      600       10        10000
3400     8.8       1000      10        9000
3800     13.6      1700      140       25000
4000     12.8      1600      140       25000
8200     8.3       2600      60        12000
1200     11.4      400       10        16000
9100     11.5      3300      60        14000
9900     12.5      3400      180       18000
9600     13.7      3600      390       25000
9600     9.6       3300      80        12000
9400     11.4      4000      100       13000
;

title3 'Principal Component Analysis';
proc factor data=SocioEconomics simple corr;
run;
```

There are two large eigenvalues, 2.8733 and 1.7967, which together account for 93.4% of the standardized variance. Thus, the first two principal components provide an adequate summary of the data for most purposes. Three components, explaining 97.7% of the variation, should be sufficient for almost any application. PROC FACTOR retains two components on the basis of the eigenvalues-greater-than-one rule since the third eigenvalue is only 0.2148.

The first component has large positive loadings for all five variables. The correlation with Services (0.9324) is especially high. The second component is a contrast of Population (0.8064) and Employment (0.7261) against School (−0.5448) and HouseValue (−0.5582), with a very small loading on Services (−0.1043).

Output 33.1.1 Principal Component Analysis

Five Socioeconomic Variables
See Page 14 of Harman: Modern Factor Analysis, 3rd Ed
Principal Component Analysis

The FACTOR Procedure

Means and Standard Deviations from 12 Observations

Variable	Mean	Std Dev
Population	6241.667	3439.9943
School	11.442	1.7865
Employment	2333.333	1241.2115
Services	120.833	114.9275
HouseValue	17000.000	6367.5313

Correlations

	Population	School	Employment	Services	HouseValue
Population	1.00000	0.00975	0.97245	0.43887	0.02241
School	0.00975	1.00000	0.15428	0.69141	0.86307
Employment	0.97245	0.15428	1.00000	0.51472	0.12193
Services	0.43887	0.69141	0.51472	1.00000	0.77765
HouseValue	0.02241	0.86307	0.12193	0.77765	1.00000

Five Socioeconomic Variables
See Page 14 of Harman: Modern Factor Analysis, 3rd Ed
Principal Component Analysis

The FACTOR Procedure
Initial Factor Method: Principal Components

Prior Communality Estimates: ONE

Eigenvalues of the Correlation Matrix: Total = 5 Average = 1

	Eigenvalue	Difference	Proportion	Cumulative
1	2.87331359	1.07665350	0.5747	0.5747
2	1.79666009	1.58182321	0.3593	0.9340
3	0.21483689	0.11490283	0.0430	0.9770
4	0.09993405	0.08467868	0.0200	0.9969
5	0.01525537		0.0031	1.0000

Output 33.1.1 *continued*

2 factors will be retained by the MINEIGEN criterion.

Factor Pattern

	Factor1	Factor2
Population	0.58096	0.80642
School	0.76704	−0.54476
Employment	0.67243	0.72605
Services	0.93239	−0.10431
HouseValue	0.79116	−0.55818

Variance Explained by Each Factor

Factor1	Factor2
2.8733136	1.7966601

Final Communality Estimates: Total = 4.669974

Population	School	Employment	Services	HouseValue
0.98782629	0.88510555	0.97930583	0.88023562	0.93750041

The final communality estimates show that all the variables are well accounted for by two components, with final communality estimates ranging from 0.8802 for Services to 0.9878 for Population.

Example 33.2: Principal Factor Analysis

This example uses the data presented in Example 33.1, and performs a principal factor analysis with squared multiple correlations for the prior communality estimates (PRIORS=SMC). Unlike Example 33.1, which analyzes the principal components, the current analysis is based on a common factor model.

To help determine whether the common factor model is appropriate, Kaiser's measure of sampling adequacy (MSA) is requested, and the residual correlations and partial correlations are computed (RESIDUAL).

The ROTATE= and REORDER options are specified to enhance factor interpretability. The ROTATE=PROMAX option produces an orthogonal varimax prerotation (default) followed by an oblique Procrustes rotation, and the REORDER option reorders the variables according to their largest factor loadings. An OUTSTAT= data set is created by PROC FACTOR and displayed in Output 33.2.12.

PROC FACTOR can produce high-quality graphs that are very useful to interpret the factor solutions. To request these graphs, you must first enable ODS Graphics by specifying the **ods**

graphics on statement, as shown in the following code. All ODS graphs in PROC FACTOR are requested with the PLOTS= option. In this example, a scree plot is requested to help you determine the number of factors. Loading plots for the initial unrotated solution, prerotated (varimax) solution, and promax-rotated solution are also requested to help you visualize the patterns of factor loadings in various stages.

```
ods graphics on;

title3 'Principal Factor Analysis with Promax Rotation';
proc factor data=SocioEconomics
   priors=smc msa residual
   rotate=promax reorder
   outstat=fact_all
   plots=(scree initloadings preloadings loadings);
run;

ods graphics off;
```

Output 33.2.1 displays the results of the principal factor extraction.

Output 33.2.1 Principal Factor Analysis

Five Socioeconomic Variables
See Page 14 of Harman: Modern Factor Analysis, 3rd Ed
Principal Factor Analysis with Promax Rotation

The FACTOR Procedure
Initial Factor Method: Principal Factors

Partial Correlations Controlling all other Variables

	Population	School	Employment	Services	HouseValue
Population	1.00000	-0.54465	0.97083	0.09612	0.15871
School	-0.54465	1.00000	0.54373	0.04996	0.64717
Employment	0.97083	0.54373	1.00000	0.06689	-0.25572
Services	0.09612	0.04996	0.06689	1.00000	0.59415
HouseValue	0.15871	0.64717	-0.25572	0.59415	1.00000

Kaiser's Measure of Sampling Adequacy: Overall MSA = 0.57536759

Population	School	Employment	Services	HouseValue
0.47207897	0.55158839	0.48851137	0.80664365	0.61281377

Prior Communality Estimates: SMC

Population	School	Employment	Services	HouseValue
0.96859160	0.82228514	0.96918082	0.78572440	0.84701921

Output 33.2.1 *continued*

Eigenvalues of the Reduced Correlation Matrix:
Total = 4.39280116 Average = 0.87856023

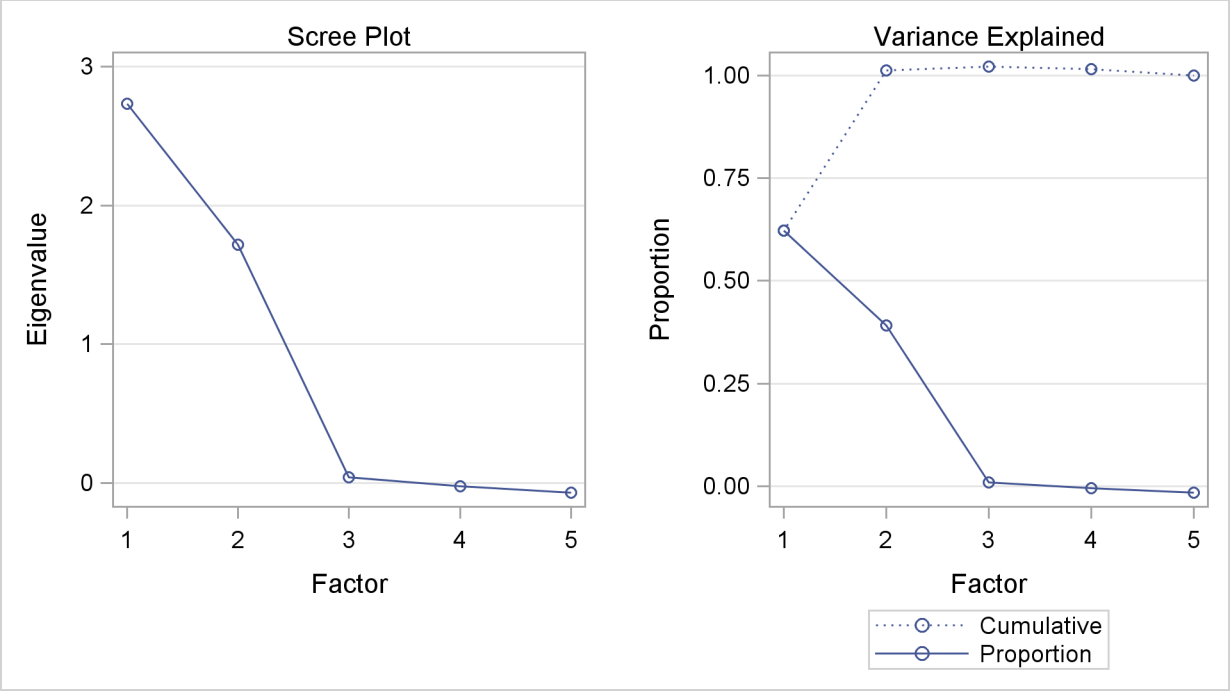
	Eigenvalue	Difference	Proportion	Cumulative
1	2.73430084	1.01823217	0.6225	0.6225
2	1.71606867	1.67650586	0.3907	1.0131
3	0.03956281	0.06408626	0.0090	1.0221
4	-.02452345	0.04808427	-0.0056	1.0165
5	-.07260772		-0.0165	1.0000

If the data are appropriate for the common factor model, the partial correlations controlling the other variables should be small compared to the original correlations. The partial correlation between the variables School and HouseValue, for example, is 0.65, slightly less than the original correlation of 0.86. The partial correlation between Population and School is −0.54, which is much larger in absolute value than the original correlation; this is an indication of trouble. Kaiser's MSA is a summary, for each variable and for all variables together, of how much smaller the partial correlations are than the original correlations. Values of 0.8 or 0.9 are considered good, while MSAs below 0.5 are unacceptable. The variables Population, School, and Employment have very poor MSAs. Only the Services variable has a good MSA. The overall MSA of 0.58 is sufficiently poor that additional variables should be included in the analysis to better define the common factors. A commonly used rule is that there should be at least three variables per factor. In the following analysis, there seems to be two common factors in these data, so more variables are needed for a reliable analysis.

The SMCs are all fairly large; hence, the factor loadings do not differ greatly from those in the principal component analysis.

The eigenvalues in Output 33.2.1 show clearly that two common factors are present. The first two largest positive eigenvalues account for 101.31% of the common variance. This is possible because the reduced correlation matrix, in general, is not necessarily positive definite, and negative eigenvalues for the matrix are possible. These cumulative proportions of common variance explained by factors are plotted in the right panel of Output 33.2.2, which shows that the curve flattens out essentially after the second factor. Showing in the left panel of Output 33.2.2 is the scree plot, which displays a sharp bend at the third eigenvalue, reinforcing the conclusion that two common factors are present.

Output 33.2.2 Scree and Variance Explained Plots



As displayed in Output 33.2.3, the principal factor pattern is similar to the principal component pattern seen in Example 33.1. For example, the variable Services has the largest loading on the first factor, and the Population variable has the smallest. The variables Population and Employment have large positive loadings on the second factor, and the HouseValue and School variables have large negative loadings.

Output 33.2.3 Initial Factor Pattern Matrix and Communalities

Factor Pattern

	Factor1	Factor2
Services	0.87899	-0.15847
HouseValue	0.74215	-0.57806
Employment	0.71447	0.67936
School	0.71370	-0.55515
Population	0.62533	0.76621

Variance Explained by Each Factor

Factor1	Factor2
2.7343008	1.7160687

Final Communality Estimates: Total = 4.450370

Population	School	Employment	Services	HouseValue
0.97811334	0.81756387	0.97199928	0.79774304	0.88494998

The final communality estimates are all fairly close to the priors. Only the communality for the variable HouseValue increased appreciably, from 0.847 to 0.885. Nearly 100% of the common variance is accounted for. The residual correlations (off-diagonal elements) are low, the largest being 0.03 (Output 33.2.4). The partial correlations are not quite as impressive, since the uniqueness values are also rather small. These results indicate that the SMCs are good but not quite optimal communality estimates.

Output 33.2.4 Residual and Partial Correlations

Residual Correlations With Uniqueness on the Diagonal

	Population	School	Employment	Services	HouseValue
Population	0.02189	−0.01118	0.00514	0.01063	0.00124
School	−0.01118	0.18244	0.02151	−0.02390	0.01248
Employment	0.00514	0.02151	0.02800	−0.00565	−0.01561
Services	0.01063	−0.02390	−0.00565	0.20226	0.03370
HouseValue	0.00124	0.01248	−0.01561	0.03370	0.11505

Root Mean Square Off-Diagonal Residuals: Overall = 0.01693282

Population	School	Employment	Services	HouseValue
0.00815307	0.01813027	0.01382764	0.02151737	0.01960158

Partial Correlations Controlling Factors

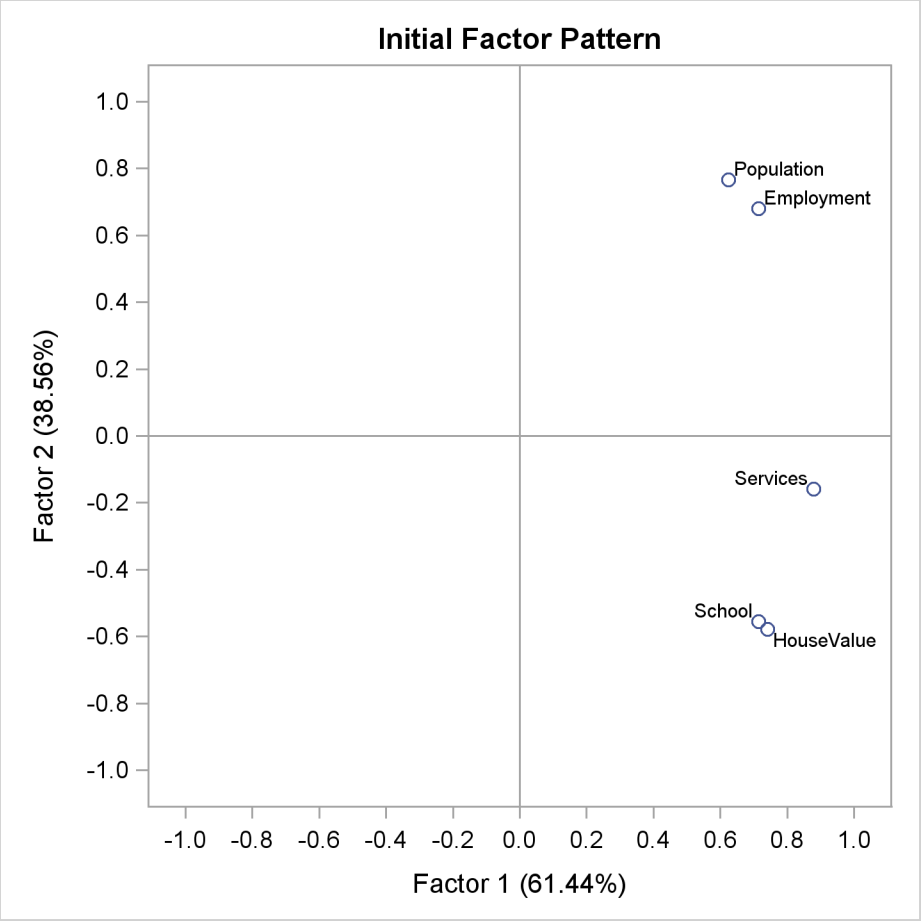
	Population	School	Employment	Services	HouseValue
Population	1.00000	−0.17693	0.20752	0.15975	0.02471
School	−0.17693	1.00000	0.30097	−0.12443	0.08614
Employment	0.20752	0.30097	1.00000	−0.07504	−0.27509
Services	0.15975	−0.12443	−0.07504	1.00000	0.22093
HouseValue	0.02471	0.08614	−0.27509	0.22093	1.00000

Root Mean Square Off-Diagonal Partials: Overall = 0.18550132

Population	School	Employment	Services	HouseValue
0.15850824	0.19025867	0.23181838	0.15447043	0.18201538

As displayed in Output 33.2.5, the unrotated factor pattern reveals two tight clusters of variables, with the variables HouseValue and School at the negative end of Factor2 axis and the variables Employment and Population at the positive end. The Services variable is in between but closer to the HouseValue and School variables. A good rotation would put the reference axes through the two clusters.

Output 33.2.5 Unrotated Factor Pattern Plot



Output 33.2.6 and Output 33.2.7 display the results of the varimax rotation. This rotation puts one axis through the variables HouseValue and School but misses the Population and Employment variables slightly.

Output 33.2.6 Varimax Rotation: Transform Matrix and Rotated Pattern

```
                Five Socioeconomic Variables
          See Page 14 of Harman: Modern Factor Analysis, 3rd Ed
              Principal Factor Analysis with Promax Rotation

                    The FACTOR Procedure
                  Prerotation Method: Varimax

              Orthogonal Transformation Matrix

                                1              2

                1         0.78895        0.61446
                2        -0.61446        0.78895
```

Output 33.2.6 *continued*

Rotated Factor Pattern

	Factor1	Factor2
HouseValue	0.94072	-0.00004
School	0.90419	0.00055
Services	0.79085	0.41509
Population	0.02255	0.98874
Employment	0.14625	0.97499

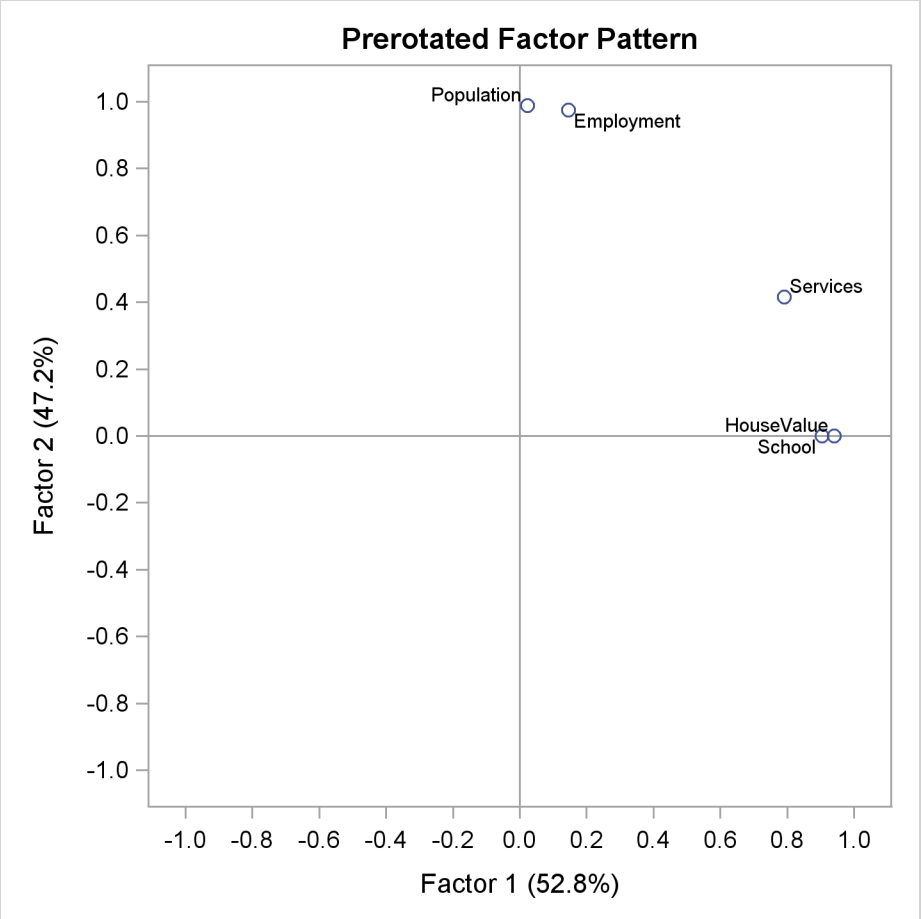
Variance Explained by Each Factor

Factor1	Factor2
2.3498567	2.1005128

Final Communality Estimates: Total = 4.450370

Population	School	Employment	Services	HouseValue
0.97811334	0.81756387	0.97199928	0.79774304	0.88494998

Output 33.2.7 Varimax-Rotated Factor Loadings

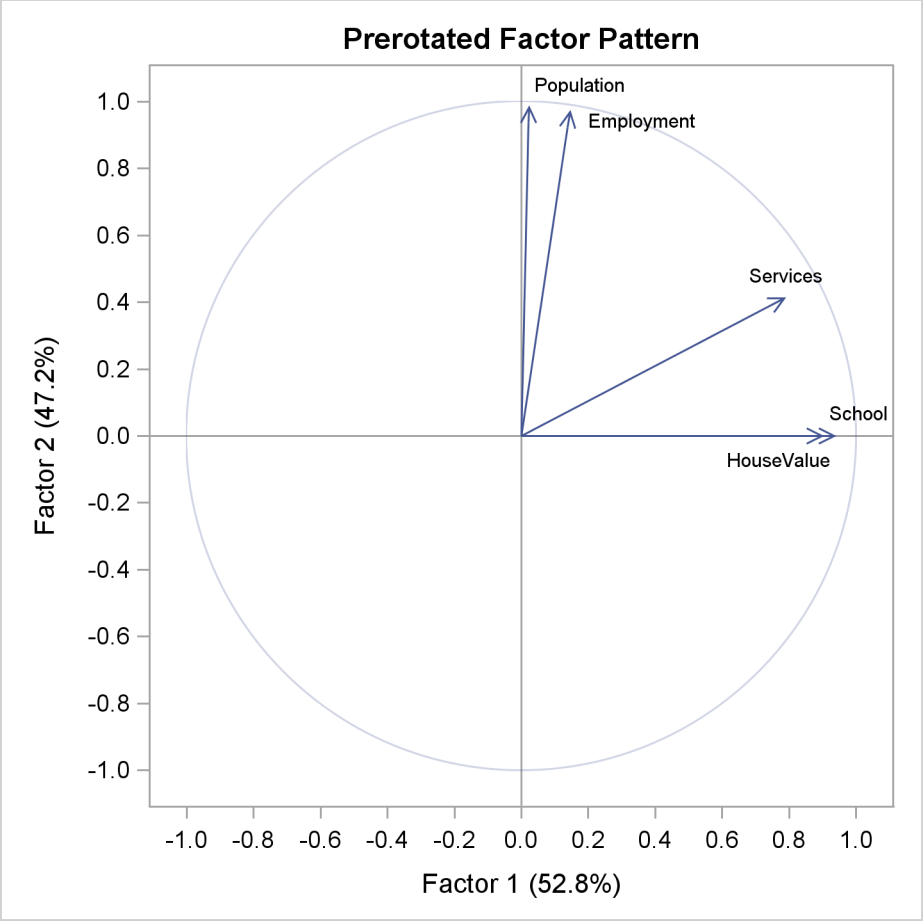


An alternative to the scatter plot of factor loadings as shown in Output 33.2.7 is the so-called vector plot of loadings. The vector plot is requested with the suboption VECTOR in the PLOTS= option. For example:

```
plots=preloadings(vector)
```

This will generate the vector plot of loadings as shown in Output 33.2.8.

Output 33.2.8 Varimax-Rotated Factor Loadings: Vector Plot



The results of oblique promax rotation are shown in Output 33.2.9 and Output 33.2.10. The corresponding plot of factor loadings is shown in Output 33.2.11.

Output 33.2.9 Promax Rotation: Procrustean Target and Transformation

Five Socioeconomic Variables
See Page 14 of Harman: Modern Factor Analysis, 3rd Ed
Principal Factor Analysis with Promax Rotation

The FACTOR Procedure
Rotation Method: Promax (power = 3)

Target Matrix for Procrustean Transformation

	Factor1	Factor2
HouseValue	1.00000	-0.00000
School	1.00000	0.00000
Services	0.69421	0.10045
Population	0.00001	1.00000
Employment	0.00326	0.96793

Output 33.2.9 *continued*

Procrustean Transformation Matrix

	1	2
1	1.04116598	-0.0986534
2	-0.1057226	0.96303019

Normalized Oblique Transformation Matrix

	1	2
1	0.73803	0.54202
2	-0.70555	0.86528

Output 33.2.10 Promax Rotation: Promax-Rotated Factor Solution

Inter-Factor Correlations

	Factor1	Factor2
Factor1	1.00000	0.20188
Factor2	0.20188	1.00000

Rotated Factor Pattern (Standardized Regression Coefficients)

	Factor1	Factor2
HouseValue	0.95558485	-0.0979201
School	0.91842142	-0.0935214
Services	0.76053238	0.33931804
Population	-0.0790832	1.00192402
Employment	0.04799	0.97509085

Reference Axis Correlations

	Factor1	Factor2
Factor1	1.00000	-0.20188
Factor2	-0.20188	1.00000

Reference Structure (Semipartial Correlations)

	Factor1	Factor2
HouseValue	0.93591	-0.09590
School	0.89951	-0.09160
Services	0.74487	0.33233
Population	-0.07745	0.98129
Employment	0.04700	0.95501

Output 33.2.10 *continued*

Variance Explained by Each Factor Eliminating Other Factors

Factor1	Factor2
2.2480892	2.0030200

Factor Structure (Correlations)

	Factor1	Factor2
HouseValue	0.93582	0.09500
School	0.89954	0.09189
Services	0.82903	0.49286
Population	0.12319	0.98596
Employment	0.24484	0.98478

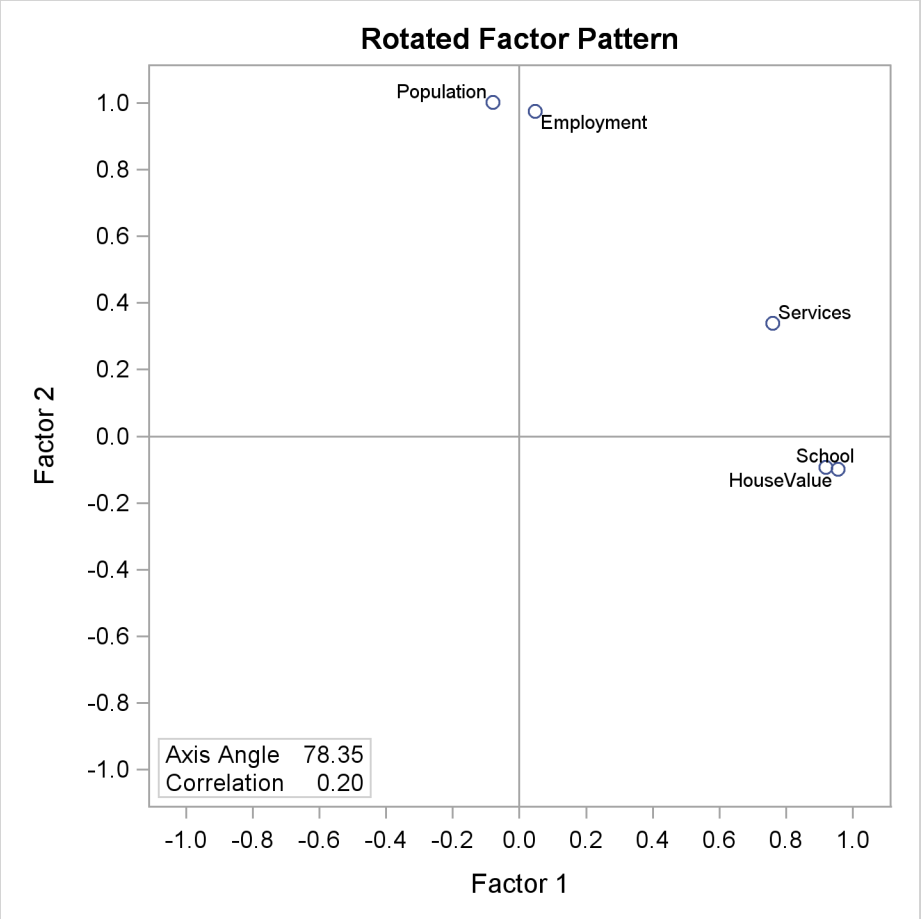
Variance Explained by Each Factor Ignoring Other Factors

Factor1	Factor2
2.4473495	2.2022803

Final Communality Estimates: Total = 4.450370

Population	School	Employment	Services	HouseValue
0.97811334	0.81756387	0.97199928	0.79774304	0.88494998

Output 33.2.11 Promax-Rotated Factor Loadings



As shown in Output 33.2.11, the promax solution places an axis through the variables Population and Employment but misses the HouseValue and School variables. Since an independent-cluster solution would be possible if it were not for the variable Services, a Harris-Kaiser rotation weighted by the Cureton-Mulaik technique could be used. Rather than reanalyze the entire problem with the Harris-Kaiser rotation, you can simply use the preceding results stored in the OUTSTAT= data set.

First, the OUTSTAT= data set is printed using this code:

```
title3 'Factor Output Data Set';
proc print data=fact_all;
run;
```

The output data set is displayed in Output 33.2.12.

Output 33.2.12 Output Data Set

Five Socioeconomic Variables
See Page 14 of Harman: Modern Factor Analysis, 3rd Ed
Factor Output Data Set

Obs	_TYPE_	_NAME_	Population	School	Employment	Services	House Value
1	MEAN		6241.67	11.4417	2333.33	120.833	17000.00
2	STD		3439.99	1.7865	1241.21	114.928	6367.53
3	N		12.00	12.0000	12.00	12.000	12.00
4	CORR	Population	1.00	0.0098	0.97	0.439	0.02
5	CORR	School	0.01	1.0000	0.15	0.691	0.86
6	CORR	Employment	0.97	0.1543	1.00	0.515	0.12
7	CORR	Services	0.44	0.6914	0.51	1.000	0.78
8	CORR	HouseValue	0.02	0.8631	0.12	0.778	1.00
9	COMMUNAL		0.98	0.8176	0.97	0.798	0.88
10	PRIORS		0.97	0.8223	0.97	0.786	0.85
11	EIGENVAL		2.73	1.7161	0.04	-0.025	-0.07
12	UNROTATE	Factor1	0.63	0.7137	0.71	0.879	0.74
13	UNROTATE	Factor2	0.77	-0.5552	0.68	-0.158	-0.58
14	RESIDUAL	Population	0.02	-0.0112	0.01	0.011	0.00
15	RESIDUAL	School	-0.01	0.1824	0.02	-0.024	0.01
16	RESIDUAL	Employment	0.01	0.0215	0.03	-0.006	-0.02
17	RESIDUAL	Services	0.01	-0.0239	-0.01	0.202	0.03
18	RESIDUAL	HouseValue	0.00	0.0125	-0.02	0.034	0.12
19	PRETRANS	Factor1	0.79	-0.6145	.	.	.
20	PRETRANS	Factor2	0.61	0.7889	.	.	.
21	PREROTAT	Factor1	0.02	0.9042	0.15	0.791	0.94
22	PREROTAT	Factor2	0.99	0.0006	0.97	0.415	-0.00
23	TRANSFOR	Factor1	0.74	-0.7055	.	.	.
24	TRANSFOR	Factor2	0.54	0.8653	.	.	.
25	FCORR	Factor1	1.00	0.2019	.	.	.
26	FCORR	Factor2	0.20	1.0000	.	.	.
27	PATTERN	Factor1	-0.08	0.9184	0.05	0.761	0.96
28	PATTERN	Factor2	1.00	-0.0935	0.98	0.339	-0.10
29	RCORR	Factor1	1.00	-0.2019	.	.	.
30	RCORR	Factor2	-0.20	1.0000	.	.	.
31	REFERENC	Factor1	-0.08	0.8995	0.05	0.745	0.94
32	REFERENC	Factor2	0.98	-0.0916	0.96	0.332	-0.10
33	STRUCTUR	Factor1	0.12	0.8995	0.24	0.829	0.94
34	STRUCTUR	Factor2	0.99	0.0919	0.98	0.493	0.09

This output data set can be used for Harris-Kaiser rotation by deleting observations with _TYPE_='PATTERN' and _TYPE_='FCORR', which are for the promax-rotated factors, and changing _TYPE_='UNROTATE' to _TYPE_='PATTERN'. In this way, the initial orthogonal factor pattern matrix is saved in the observations with _TYPE_='PATTERN'. The following factor analysis will then read in the factor pattern in the fact2 data set as an initial factor solution, which will then be rotated by the Harris-Kaiser rotation with Cureton-Mulaik weights.

The following statements produce Output 33.2.13:

```
data fact2(type=factor);
   set fact_all;
   if _TYPE_ in('PATTERN' 'FCORR') then delete;
   if _TYPE_='UNROTATE' then _TYPE_='PATTERN';

ods graphics on;

title3 'Harris-Kaiser Rotation with Cureton-Mulaik Weights';
proc factor rotate=hk norm=weight reorder
     plots=loadings;
run;

ods graphics off;
```

The results of the Harris-Kaiser rotation are displayed in Output 33.2.13.

Output 33.2.13 Harris-Kaiser Rotation

```
                    Five Socioeconomic Variables
            See Page 14 of Harman: Modern Factor Analysis, 3rd Ed
              Harris-Kaiser Rotation with Cureton-Mulaik Weights

                         The FACTOR Procedure
                Rotation Method: Harris-Kaiser (hkpower = 0)

                      Variable Weights for Rotation

 Population          School      Employment        Services      HouseValue

 0.95982747      0.93945424      0.99746396      0.12194766      0.94007263

                     Oblique Transformation Matrix

                                       1               2

                          1         0.73537         0.61899
                          2        -0.68283         0.78987

                       Inter-Factor Correlations

                                 Factor1         Factor2

                   Factor1       1.00000         0.08358
                   Factor2       0.08358         1.00000
```

Output 33.2.13 *continued*

Rotated Factor Pattern (Standardized Regression Coefficients)

	Factor1	Factor2
HouseValue	0.94048	0.00279
School	0.90391	0.00327
Services	0.75459	0.41892
Population	−0.06335	0.99227
Employment	0.06152	0.97885

Reference Axis Correlations

	Factor1	Factor2
Factor1	1.00000	−0.08358
Factor2	−0.08358	1.00000

Reference Structure (Semipartial Correlations)

	Factor1	Factor2
HouseValue	0.93719	0.00278
School	0.90075	0.00326
Services	0.75195	0.41745
Population	−0.06312	0.98880
Employment	0.06130	0.97543

Variance Explained by Each Factor Eliminating Other Factors

Factor1	Factor2
2.2628537	2.1034731

Factor Structure (Correlations)

	Factor1	Factor2
HouseValue	0.94071	0.08139
School	0.90419	0.07882
Services	0.78960	0.48198
Population	0.01958	0.98698
Employment	0.14332	0.98399

Variance Explained by Each Factor Ignoring Other Factors

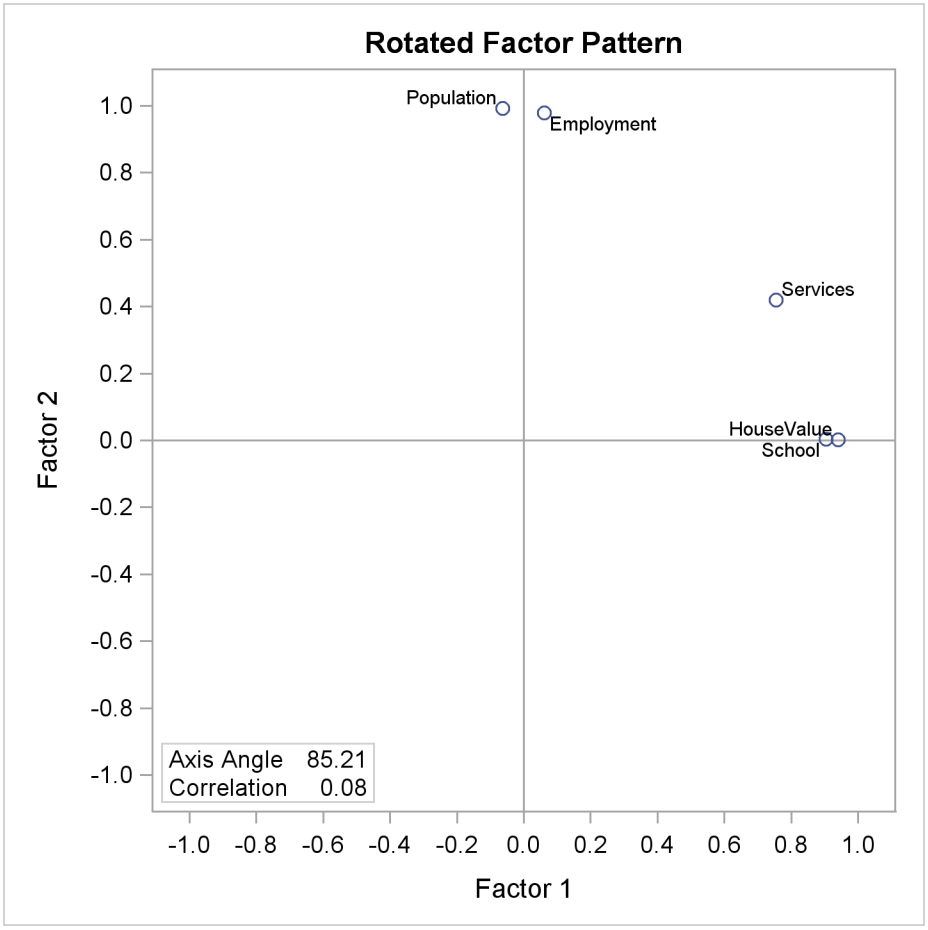
Factor1	Factor2
2.3468965	2.1875158

Final Communality Estimates: Total = 4.450370

Population	School	Employment	Services	HouseValue
0.97811334	0.81756387	0.97199928	0.79774304	0.88494998

A plot of the rotated loadings is shown in Output 33.2.14.

Output 33.2.14 Factor Pattern with Harris-Kaiser Rotation



In the results of the Harris-Kaiser rotation, the variable Services receives a small weight, and the axes are placed as desired.

Example 33.3: Maximum Likelihood Factor Analysis

This example uses maximum likelihood factor analyses for one, two, and three factors. It is already apparent from the principal factor analysis that the best number of common factors is almost certainly two. The one- and three-factor ML solutions reinforce this conclusion and illustrate some of the numerical problems that can occur. The following statements produce Output 33.3.1 through Output 33.3.3:

```
title3 'Maximum Likelihood Factor Analysis with One Factor';
proc factor data=SocioEconomics method=ml heywood n=1;
run;

title3 'Maximum Likelihood Factor Analysis with Two Factors';
proc factor data=SocioEconomics method=ml heywood n=2;
run;

title3 'Maximum Likelihood Factor Analysis with Three Factors';
proc factor data=SocioEconomics method=ml heywood n=3;
run;
```

Output 33.3.1 displays the results of the analysis with one factor.

Output 33.3.1 Maximum Likelihood Factor Analysis

```
                    Five Socioeconomic Variables
          See Page 14 of Harman: Modern Factor Analysis, 3rd Ed
            Maximum Likelihood Factor Analysis with One Factor

                         The FACTOR Procedure
                Initial Factor Method: Maximum Likelihood

                   Prior Communality Estimates: SMC

  Population          School       Employment         Services       HouseValue

  0.96859160       0.82228514       0.96918082       0.78572440       0.84701921

      Preliminary Eigenvalues: Total = 76.1165859  Average = 15.2233172

                Eigenvalue    Difference    Proportion    Cumulative

           1    63.7010086    50.6462895        0.8369        0.8369
           2    13.0547191    12.7270798        0.1715        1.0084
           3     0.3276393     0.6749199        0.0043        1.0127
           4    -0.3472805     0.2722202       -0.0046        1.0081
           5    -0.6195007                     -0.0081        1.0000
```

Output 33.3.1 *continued*

```
              1 factor will be retained by the NFACTOR criterion.



Iteration    Criterion    Ridge   Change              Communalities

    1        6.5429218   0.0000   0.1033   0.93828   0.72227   1.00000   0.71940
                                           0.74371
    2        3.1232699   0.0000   0.7288   0.94566   0.02380   1.00000   0.26493
                                           0.01487

        Convergence criterion satisfied.

                  Significance Tests Based on 12 Observations

                                                                  Pr >
                     Test                     DF    Chi-Square     ChiSq

        H0: No common factors                 10       54.2517    <.0001
        HA: At least one common factor
        H0: 1 Factor is sufficient             5       24.4656    0.0002
        HA: More factors are needed

            Chi-Square without Bartlett's Correction       34.355969
            Akaike's Information Criterion                 24.355969
            Schwarz's Bayesian Criterion                   21.931436
            Tucker and Lewis's Reliability Coefficient      0.120231

                        Squared Canonical Correlations

                                    Factor1

                                  1.0000000

 Eigenvalues of the Weighted Reduced Correlation Matrix: Total = 0  Average = 0

                                Eigenvalue    Difference

                           1         Infty         Infty
                           2    1.92716032    2.15547340
                           3    -.22831308    0.56464322
                           4    -.79295630    0.11293464
                           5    -.90589094

                                 Factor Pattern

                                              Factor1

                           Population         0.97245
                           School             0.15428
                           Employment         1.00000
                           Services           0.51472
                           HouseValue         0.12193
```

Output 33.3.1 *continued*

```
                Variance Explained by Each Factor

                Factor        Weighted    Unweighted

                Factor1     17.8010629    2.24926004

          Final Communality Estimates and Variable Weights
     Total Communality: Weighted = 17.801063   Unweighted = 2.249260

                Variable      Communality        Weight

                Population     0.94565561    18.4011648
                School         0.02380349     1.0243839
                Employment     1.00000000          Infty
                Services       0.26493499     1.3604239
                HouseValue     0.01486595     1.0150903
```

The solution on the second iteration is so close to the optimum that PROC FACTOR cannot find a better solution; hence you receive this message:

```
Convergence criterion satisfied.
```

When this message appears, you should try rerunning PROC FACTOR with different prior communality estimates to make sure that the solution is correct. In this case, other prior estimates lead to the same solution or possibly to worse local optima, as indicated by the information criteria or the chi-square values.

The variable Employment has a communality of 1.0 and, therefore, an infinite weight that is displayed next to the final communality estimate as a missing/infinite value. The first eigenvalue is also infinite. Infinite values are ignored in computing the total of the eigenvalues and the total final communality.

Output 33.3.2 displays the results of the analysis with two factors. The analysis converges without incident. This time, however, the Population variable is a Heywood case.

Output 33.3.2 Maximum Likelihood Factor Analysis: Two Factors

```
                    Five Socioeconomic Variables
            See Page 14 of Harman: Modern Factor Analysis, 3rd Ed
             Maximum Likelihood Factor Analysis with Two Factors

                        The FACTOR Procedure
                Initial Factor Method: Maximum Likelihood

                   Prior Communality Estimates: SMC

  Population          School      Employment        Services      HouseValue

  0.96859160      0.82228514      0.96918082      0.78572440      0.84701921
```

Output 33.3.2 *continued*

Preliminary Eigenvalues: Total = 76.1165859 Average = 15.2233172

	Eigenvalue	Difference	Proportion	Cumulative
1	63.7010086	50.6462895	0.8369	0.8369
2	13.0547191	12.7270798	0.1715	1.0084
3	0.3276393	0.6749199	0.0043	1.0127
4	-0.3472805	0.2722202	-0.0046	1.0081
5	-0.6195007		-0.0081	1.0000

2 factors will be retained by the NFACTOR criterion.

Iteration	Criterion	Ridge	Change	Communalities			
1	0.3431221	0.0000	0.0471	1.00000	0.80672	0.95058	0.79348
				0.89412			
2	0.3072178	0.0000	0.0307	1.00000	0.80821	0.96023	0.81048
				0.92480			
3	0.3067860	0.0000	0.0063	1.00000	0.81149	0.95948	0.81677
				0.92023			
4	0.3067373	0.0000	0.0022	1.00000	0.80985	0.95963	0.81498
				0.92241			
5	0.3067321	0.0000	0.0007	1.00000	0.81019	0.95955	0.81569
				0.92187			

Convergence criterion satisfied.

Significance Tests Based on 12 Observations

Test	DF	Chi-Square	Pr > ChiSq
H0: No common factors	10	54.2517	<.0001
HA: At least one common factor			
H0: 2 Factors are sufficient	1	2.1982	0.1382
HA: More factors are needed			

Chi-Square without Bartlett's Correction	3.3740530
Akaike's Information Criterion	1.3740530
Schwarz's Bayesian Criterion	0.8891463
Tucker and Lewis's Reliability Coefficient	0.7292200

Squared Canonical Correlations

Factor1	Factor2
1.0000000	0.9518891

Output 33.3.2 *continued*

Eigenvalues of the Weighted Reduced Correlation Matrix: Total = 19.7853157 Average = 4.94632893

	Eigenvalue	Difference	Proportion	Cumulative
1	Infty	Infty		
2	19.7853143	19.2421292	1.0000	1.0000
3	0.5431851	0.5829564	0.0275	1.0275
4	-0.0397713	0.4636411	-0.0020	1.0254
5	-0.5034124		-0.0254	1.0000

Factor Pattern

	Factor1	Factor2
Population	1.00000	0.00000
School	0.00975	0.90003
Employment	0.97245	0.11797
Services	0.43887	0.78930
HouseValue	0.02241	0.95989

Variance Explained by Each Factor

Factor	Weighted	Unweighted
Factor1	24.4329707	2.13886057
Factor2	19.7853143	2.36835294

Final Communality Estimates and Variable Weights

Total Communality: Weighted = 44.218285 Unweighted = 4.507214

Variable	Communality	Weight
Population	1.00000000	Infty
School	0.81014489	5.2682940
Employment	0.95957142	24.7246669
Services	0.81560348	5.4256462
HouseValue	0.92189372	12.7996793

The results of the three-factor analysis are shown in Output 33.3.3.

Output 33.3.3 Maximum Likelihood Factor Analysis: Three Factors

```
                    Five Socioeconomic Variables
             See Page 14 of Harman: Modern Factor Analysis, 3rd Ed
             Maximum Likelihood Factor Analysis with Three Factors

                         The FACTOR Procedure
                  Initial Factor Method: Maximum Likelihood

                     Prior Communality Estimates: SMC

   Population           School      Employment        Services      HouseValue

   0.96859160      0.82228514      0.96918082      0.78572440      0.84701921

        Preliminary Eigenvalues: Total = 76.1165859  Average = 15.2233172

                  Eigenvalue    Difference    Proportion    Cumulative

            1    63.7010086    50.6462895        0.8369        0.8369
            2    13.0547191    12.7270798        0.1715        1.0084
            3     0.3276393     0.6749199        0.0043        1.0127
            4    -0.3472805     0.2722202       -0.0046        1.0081
            5    -0.6195007                     -0.0081        1.0000

             3 factors will be retained by the NFACTOR criterion.


               WARNING: Too many factors for a unique solution.


Iteration    Criterion    Ridge   Change              Communalities

    1        0.1798029   0.0313   0.0501   0.96081   0.84184   1.00000   0.80175
                                           0.89716
    2        0.0016405   0.0313   0.0678   0.98081   0.88713   1.00000   0.79559
                                           0.96500
    3        0.0000041   0.0313   0.0094   0.98195   0.88603   1.00000   0.80498
                                           0.96751
    4        0.0000000   0.0313   0.0006   0.98202   0.88585   1.00000   0.80561
                                           0.96735

        ERROR: Converged, but not to a proper optimum.
```

Output 33.3.3 *continued*

```
                    Try a different 'PRIORS' statement.



              Significance Tests Based on 12 Observations

                                                                Pr >
                 Test                      DF    Chi-Square     ChiSq

       H0: No common factors                 10       54.2517    <.0001
       HA: At least one common factor
       H0: 3 Factors are sufficient          -2        0.0000     .
       HA: More factors are needed

           Chi-Square without Bartlett's Correction       0.0000003
           Akaike's Information Criterion                 4.0000003
           Schwarz's Bayesian Criterion                   4.9698136
           Tucker and Lewis's Reliability Coefficient     0.0000000

                      Squared Canonical Correlations

                  Factor1         Factor2         Factor3

                 1.0000000       0.9751895       0.6894465

              Eigenvalues of the Weighted Reduced Correlation
             Matrix: Total = 41.5254193  Average = 10.3813548

                Eigenvalue    Difference    Proportion    Cumulative

          1         Infty         Infty
          2    39.3054826    37.0854258        0.9465        0.9465
          3     2.2200568     2.2199693        0.0535        1.0000
          4     0.0000875     0.0002949        0.0000        1.0000
          5    -0.0002075                     -0.0000        1.0000

                            Factor Pattern

                         Factor1         Factor2         Factor3

        Population       0.97245        -0.11233        -0.15409
        School           0.15428         0.89108         0.26083
        Employment       1.00000         0.00000         0.00000
        Services         0.51472         0.72416        -0.12766
        HouseValue       0.12193         0.97227        -0.08473

                   Variance Explained by Each Factor

                  Factor        Weighted    Unweighted

                  Factor1     54.6115241    2.24926004
                  Factor2     39.3054826    2.27634375
                  Factor3      2.2200568    0.11525433
```

Output 33.3.3 *continued*

Final Communality Estimates and Variable Weights

Total Communality: Weighted = 96.137063 Unweighted = 4.640858

Variable	Communality	Weight
Population	0.98201660	55.6066901
School	0.88585165	8.7607194
Employment	1.00000000	Infty
Services	0.80564301	5.1444261
HouseValue	0.96734687	30.6251078

In the results, a warning message is displayed:

```
WARNING:  Too many factors for a unique solution.
```

The number of parameters in the model exceeds the number of elements in the correlation matrix from which they can be estimated, so an infinite number of different perfect solutions can be obtained. The criterion approaches zero at an improper optimum, as indicated by this message:

```
Converged, but not to a proper optimum.
```

The degrees of freedom for the chi-square test are -2, so a probability level cannot be computed for three factors. Note also that the variable Employment is a Heywood case again.

The probability levels for the chi-square test are 0.0001 for the hypothesis of no common factors, 0.0002 for one common factor, and 0.1382 for two common factors. Therefore, the two-factor model seems to be an adequate representation. Akaike's information criterion and Schwarz's Bayesian criterion attain their minimum values at two common factors, so there is little doubt that two factors are appropriate for these data.

Example 33.4: Using Confidence Intervals to Locate Salient Factor Loadings

This example illustrates how you can use the standard errors and confidence intervals to understand the pattern of factor loadings under the maximum likelihood estimation. There are nine tests and you want a three-factor solution (N=3) for a correlation matrix based on 200 observations. The following statements define the input data set and specify the desirable analysis by the FACTOR procedure:

```
data test(type=corr);
   title 'Quartimin-Rotated Factor Solution with Standard Errors';
   input _name_ $ test1-test9;
   _type_ = 'corr';
   datalines;
Test1      1  .561  .602  .290  .404  .328  .367  .179 -.268
Test2   .561     1  .743  .414  .526  .442  .523  .289 -.399
Test3   .602  .743     1  .286  .343  .361  .679  .456 -.532
Test4   .290  .414  .286     1  .677  .446  .412  .400 -.491
Test5   .404  .526  .343  .677     1  .584  .408  .299 -.466
Test6   .328  .442  .361  .446  .584     1  .333  .178 -.306
Test7   .367  .523  .679  .412  .408  .333     1  .711 -.760
Test8   .179  .289  .456  .400  .299  .178  .711     1 -.725
Test9  -.268 -.399 -.532 -.491 -.466 -.306 -.760 -.725     1
;

title2 'A nine-variable-three-factor example';
proc factor data=test method=ml reorder rotate=quartimin
   nobs=200 n=3 se cover=.45 alpha=.1;
run;
```

In the PROC FACTOR statement, you apply quartimin rotation with (default) Kaiser normalization. You define loadings with magnitudes greater than 0.45 to be salient (COVER=0.45) and use 90% confidence intervals (ALPHA=0.1) to judge the salience. The REORDER option is specified so that variables that have similar loadings with factors are clustered together.

After the quartimin rotation, the correlation matrix for factors is shown in Output 33.4.1.

Output 33.4.1 Quartimin-Rotated Factor Solution with Standard Errors

Quartimin-Rotated Factor Solution with Standard Errors
A nine-variable-three-factor example

The FACTOR Procedure
Rotation Method: Quartimin

Inter-Factor Correlations
With 90% confidence limits
Estimate/StdErr/LowerCL/UpperCL

	Factor1	Factor2	Factor3
Factor1	1.00000	0.41283	0.38304
	0.00000	0.06267	0.06060
	.	0.30475	0.27919
	.	0.51041	0.47804
Factor2	0.41283	1.00000	0.47006
	0.06267	0.00000	0.05116
	0.30475	.	0.38177
	0.51041	.	0.54986
Factor3	0.38304	0.47006	1.00000
	0.06060	0.05116	0.00000
	0.27919	0.38177	.
	0.47804	0.54986	.

The factors are medium to highly correlated. The confidence intervals seem to be very wide, suggesting that the estimation of factor correlations might not be very accurate for this sample size. For example, the 90% confidence interval for the correlation between Factor1 and Factor2 is (0.30, 0.51), a range of 0.21. You might need a larger sample to get a narrower interval, or you might need a better estimation.

Next, coverage displays for factor loadings are shown in Output 33.4.2.

Output 33.4.2 Using the Rotated Factor Pattern to Interpret the Factors

Rotated Factor Pattern (Standardized Regression Coefficients)
With 90% confidence limits; Cover |*| = 0.45?
Estimate/StdErr/LowerCL/UpperCL/Coverage Display

	Factor1	Factor2	Factor3
test8	0.86810	−0.05045	0.00114
	0.03282	0.03185	0.03087
	0.80271	−0.10265	−0.04959
	0.91286	0.00204	0.05187
	0*[]	*[0]	[0]*
test7	0.73204	0.27296	0.01098
	0.04434	0.05292	0.03838
	0.65040	0.18390	−0.05211
	0.79697	0.35758	0.07399
	0*[]	0[]*	[0]*
test9	−0.79654	−0.01230	−0.17307
	0.03948	0.04225	0.04420
	−0.85291	−0.08163	−0.24472
	−0.72180	0.05715	−0.09955
	[]*0	*[0]	*[]0
test3	0.27715	0.91156	−0.19727
	0.05489	0.04877	0.02981
	0.18464	0.78650	−0.24577
	0.36478	0.96481	−0.14778
	0[]*	0*[]	*[]0
test2	0.01063	0.71540	0.20500
	0.05060	0.05148	0.05496
	−0.07248	0.61982	0.11310
	0.09359	0.79007	0.29342
	[0]*	0*[]	0[]*
test1	−0.07356	0.63815	0.13983
	0.04245	0.05380	0.05597
	−0.14292	0.54114	0.04682
	−0.00348	0.71839	0.23044
	*[]0	0*[]	0[]*
test5	0.00863	0.03234	0.91282
	0.04394	0.04387	0.04509
	−0.06356	−0.03986	0.80030
	0.08073	0.10421	0.96323
	[0]*	[0]*	0*[]
test4	0.22357	−0.07576	0.67925
	0.05956	0.03640	0.05434
	0.12366	−0.13528	0.57955
	0.31900	−0.01569	0.75891
	0[]*	*[]0	0*[]
test6	−0.04295	0.21911	0.53183
	0.05114	0.07481	0.06905
	−0.12656	0.09319	0.40893
	0.04127	0.33813	0.63578
	*[0]	0[]*	0[*]

The coverage displays in Output 33.4.2 show that Test8, Test7, and Test9 have salient relationships with Factor1. The coverage displays are either '0*[ ]' or '[ ]*0', indicating that the entire 90% confidence intervals for the corresponding loadings are beyond the salience value at 0.45. On the other hand, the coverage display for Test3 on Factor1 is '0[ ]*'. This indicates that even though the loading estimate is significantly larger than zero, it is not large enough to be salient. Similarly, Test3, Test2, and Test1 have salient relationships with Factor2, while Test5 and Test4 have salient relationships with Factor3. For Test6, its relationship with Factor3 is a little bit ambiguous; the 90% confidence interval approximately covers values between 0.40 and 0.64. This means that the population value might have been smaller or larger than 0.45. It is marginal evidence for a salient relationship.

For oblique factor solutions, some researchers prefer to examine the factor structure loadings, which represent correlations, for determining salient relationships. In Output 33.4.3, the factor structure loadings and the associated standard error estimates and coverage displays are shown.

Output 33.4.3 Using the Factor Structure to Interpret the Factors

Factor Structure (Correlations)
With 90% confidence limits; Cover |*| = 0.45?
Estimate/StdErr/LowerCL/UpperCL/Coverage Display

	Factor1	Factor2	Factor3
test8	0.84771	0.30847	0.30994
	0.02871	0.06593	0.06263
	0.79324	0.19641	0.20363
	0.88872	0.41257	0.40904
	0*[]	0[]*	0[]*
test7	0.84894	0.58033	0.41970
	0.02688	0.05265	0.06060
	0.79834	0.48721	0.31523
	0.88764	0.66041	0.51412
	0*[]	0*[]	0[*]
test9	−0.86791	−0.42248	−0.48396
	0.02522	0.06187	0.05504
	−0.90381	−0.51873	−0.56921
	−0.81987	−0.31567	−0.38841
	[]*0	[*]0	[*]0
test3	0.57790	0.93325	0.33738
	0.05069	0.02953	0.06779
	0.48853	0.86340	0.22157
	0.65528	0.96799	0.44380
	0*[]	0*[]	0[]*
test2	0.38449	0.81615	0.54535
	0.06143	0.03106	0.05456
	0.27914	0.75829	0.44946
	0.48070	0.86126	0.62883
	0[*]	0*[]	0[*]
test1	0.24345	0.67351	0.41162
	0.06864	0.04284	0.05995
	0.12771	0.59680	0.30846
	0.35264	0.73802	0.50522
	0[]*	0*[]	0[*]
test5	0.37163	0.46498	0.93132
	0.06092	0.04979	0.03277
	0.26739	0.37923	0.85159
	0.46727	0.54282	0.96894
	0[*]	0[*]	0*[]
test4	0.45248	0.33583	0.72927
	0.05876	0.06289	0.04061
	0.35072	0.22867	0.65527
	0.54367	0.43494	0.78941
	0[*]	0[]*	0*[]
test6	0.25122	0.45137	0.61837
	0.07140	0.05858	0.05051
	0.13061	0.34997	0.52833
	0.36450	0.54232	0.69465
	0[]*	0[*]	0*[]

The interpretations based on the factor structure matrix do not change much from that based on the factor loadings except for Test3 and Test9. Test9 now has a salient correlation with Factor3. For Test3, it has salient correlations with both Factor1 and Factor2. Fortunately, there are still tests that have salient correlations only with either Factor1 or Factor2 (but not both). This would make interpretations of factors less problematic.

References

Akaike, H. (1973), "Information Theory and the Extension of the Maximum Likelihood Principle," in *Second International Symposium on Information Theory*, ed. V. N. Petrov and F. Csaki, Budapest: Akailseoniai-Kiudo, 267–281.

Akaike, H. (1974), "A New Look at the Statistical Identification Model," *IEEE Transactions on Automatic Control,* 19, 716–723.

Akaike, H. (1987), "Factor Analysis and AIC," *Psychometrika* 52, 317–332.

Archer, C. O. and Jennrich, R. I. (1973), "Standard Errors for Orthogonally Rotated Factor Loadings," *Psychometrika,* 38, 581–592.

Bickel, P. J. and Doksum, K. A. (1977), *Mathematical Statistics,* San Francisco: Holden-Day.

Browne, M. W. (1982), "Covariance Structures," in *Topics in Applied Multivariate Analysis*, ed. D. M. Hawkins, Cambridge: Cambridge University Press, 72–141.

Browne, M. W., Cudeck, R., Tateneni, K., and Mels, G. (2004), *CEFA: Comprehensive Exploratory Factor Analysis, Version 2.00.* http://quantrm2.psy.ohio-state.edu/browne/software.php

Cattell, R. B. (1966), "The Scree Test for the Number of Factors," *Multivariate Behavioral Research,* 1, 245–276.

Cattell, R. B. (1978), *The Scientific Use of Factor Analysis,* New York: Plenum.

Cattell, R. B. and Vogelman, S. (1977), "A Comprehensive Trial of the Scree and KG Criteria for Determining the Number of Factors," *Multivariate Behavioral Research,* 12, 289–325.

Cerny, B. A. and Kaiser, H. F. (1977), "A Study of a Measure of Sampling Adequacy for Factor-Analytic Correlation Matrices," *Multivariate Behavioral Research,* 12, 43–47.

Crawford, C. B. and Ferguson, G. A. (1970), "A General Rotation Criterion and Its Use in Orthogonal Rotation," *Psychometrika,* 35, 321–332.

Cureton, E. E. (1968), *A Factor Analysis of Project TALENT Tests and Four Other Test Batteries* (Interim Report 4 to the U.S. Office of Education, Cooperative Research Project No. 3051), Palo Alto, CA: Project TALENT Office, American Institutes for Research and University of Pittsburgh.

Cureton, E. E. and Mulaik, S. A. (1975), "The Weighted Varimax Rotation and the Promax Rotation," *Psychometrika,* 40, 183–195.

Dziuban, C. D. and Harris, C. W. (1973), "On the Extraction of Components and the Applicability of the Factor Model," *American Educational Research Journal,* 10, 93–99.

Fuller (1987), *Measurement Error Models*, New York: John Wiley & Sons.

Geweke, J. F. and Singleton, K. J. (1980), "Interpreting the Likelihood Ratio Statistic in Factor Models When Sample Size Is Small," *Journal of the American Statistical Association,* 75, 133–137.

Gorsuch, R. L. (1974), *Factor Analysis,* Philadelphia: W. B. Saunders.

Harman, H. H. (1976), *Modern Factor Analysis,* Third Edition, Chicago: University of Chicago Press.

Harris, C. W. (1962), "Some Rao-Guttman Relationships," *Psychometrika,* 27, 247–263.

Hayashi, K. and Yung, Y. F. (1999), "Standard Errors for the Class of Orthomax-Rotated Factor Loadings: Some Matrix Results," *Psychometrika,* 64, 451–460.

Horn, J. L. and Engstrom, R. (1979), "Cattell's Scree Test in Relation to Bartlett's Chi-Square Test and Other Observations on the Number of Factors Problem," *Multivariate Behavioral Research,* 14, 283–300.

Jennrich, R. I. (1973), "Standard Errors for Obliquely Rotated Factor Loadings," *Psychometrika,* 38, 593–604.

Jennrich, R. I. (1974), "Simplified Formulae for Standard Errors in Maximum-Likelihood Factor Analysis," *British Journal of Mathematical and Statistical Psychology,* 27, 122–131.

Jöreskog, K. G. (1962), "On the Statistical Treatment of Residuals in Factor Analysis," *Psychometrika,* 27, 335–354.

Jöreskog, K. G. (1977), "Factor Analysis by Least-Squares and Maximum Likelihood Methods," in *Statistical Methods for Digital Computers*, ed. K. Enslein, A. Ralston, and H. S. Wilf, New York: John Wiley & Sons.

Kaiser, H. F. (1963), "Image Analysis," in *Problems in Measuring Change*, ed. C. W. Harris, Madison: University of Wisconsin Press.

Kaiser, H. F. (1970), "A Second Generation Little Jiffy," *Psychometrika,* 35, 401–415.

Kaiser, H. F. and Rice, J. (1974), "Little Jiffy, Mark IV," *Educational and Psychological Measurement,* 34, 111–117.

Kerlinger, F. N. and Pedhazur, E. J. (1973), *Multiple Regression in Behavioral Research,* New York: Holt, Rinehart & Winston.

Kim, J. O. and Mueller, C. W. (1978a), *Introduction to Factor Analysis: What It Is and How To Do It,* Sage University Paper Series on Quantitative Applications in the Social Sciences, series no. 07-013, Beverly Hills, CA: Sage Publications.

Kim, J. O. and Mueller, C. W. (1978b), *Factor Analysis: Statistical Methods and Practical Issues,*

Sage University Paper Series on Quantitative Applications in the Social Sciences, series no. 07-014, Beverly Hills, CA: Sage Publications.

Lawley, D. N. and Maxwell, A. E. (1971), *Factor Analysis as a Statistical Method,* New York: Macmillan.

Lee, H. B. and Comrey, A. L. (1979), "Distortions in a Commonly Used Factor Analytic Procedure," *Multivariate Behavioral Research,* 14, 301–321.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979), *Multivariate Analysis,* London: Academic Press.

Morrison, D. F. (1976), *Multivariate Statistical Methods,* Second Edition, New York: McGraw-Hill.

Mulaik, S. A. (1972), *The Foundations of Factor Analysis,* New York: McGraw-Hill.

Rao, C. R. (1955), "Estimation and Tests of Significance in Factor Analysis," *Psychometrika,* 20, 93–111.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics,* 6, 461–464.

Spearman, C. (1904), "General Intelligence Objectively Determined and Measured," *American Journal of Psychology,* 15, 201–293.

Stewart, D. W. (1981), "The Application and Misapplication of Factor Analysis in Marketing Research," *Journal of Marketing Research,* 18, 51–62.

Tucker, L. R. and Lewis, C. (1973), "A Reliability Coefficient for Maximum Likelihood Factor Analysis," *Psychometrika,* 38, 1–10.

Yung, Y. F. and Hayashi, K. (2001), "A Computationally Efficient Method for Obtaining Standard Error Estimates for the Promax and Related Solutions," *British Journal of Mathematical and Statistical Psychology,* 54, 125–138.

Chapter 34

The FASTCLUS Procedure

Contents

Overview: FASTCLUS Procedure **1621**
Background 1622
Getting Started: FASTCLUS Procedure **1624**
Syntax: FASTCLUS Procedure **1632**
PROC FASTCLUS Statement 1632
BY Statement 1640
FREQ Statement 1640
ID Statement 1641
VAR Statement 1641
WEIGHT Statement 1641
Details: FASTCLUS Procedure **1642**
Updates in the FASTCLUS Procedure 1642
Missing Values 1642
Output Data Sets 1643
Computational Resources 1647
Using PROC FASTCLUS 1647
Displayed Output 1649
ODS Table Names 1652
Examples: FASTCLUS Procedure **1653**
Example 34.1: Fisher's Iris Data 1653
Example 34.2: Outliers 1662
References **1673**

Overview: FASTCLUS Procedure

The FASTCLUS procedure performs a disjoint cluster analysis on the basis of distances computed from one or more quantitative variables. The observations are divided into clusters such that every observation belongs to one and only one cluster; the clusters do not form a tree structure as they do in the CLUSTER procedure. If you want separate analysis for different numbers of clusters, you can run PROC FASTCLUS once for each analysis. Alternatively, to do hierarchical clustering on a large data set, use PROC FASTCLUS to find initial clusters, and then use those initial clusters as input to PROC CLUSTER.

By default, the FASTCLUS procedure uses Euclidean distances, so the cluster centers are based on least squares estimation. This kind of clustering method is often called a *k-means model*, since the cluster centers are the means of the observations assigned to each cluster when the algorithm is run to complete convergence. Each iteration reduces the least squares criterion until convergence is achieved.

Often there is no need to run the FASTCLUS procedure to convergence. PROC FASTCLUS is designed to find good clusters (but not necessarily the best possible clusters) with only two or three passes through the data set. The initialization method of PROC FASTCLUS guarantees that, if there exist clusters such that all distances between observations in the same cluster are less than all distances between observations in different clusters, and if you tell PROC FASTCLUS the correct number of clusters to find, it can always find such a clustering without iterating. Even with clusters that are not as well separated, PROC FASTCLUS usually finds initial seeds that are sufficiently good that few iterations are required. Hence, by default, PROC FASTCLUS performs only one iteration.

The initialization method used by the FASTCLUS procedure makes it sensitive to outliers. PROC FASTCLUS can be an effective procedure for detecting outliers because outliers often appear as clusters with only one member.

The FASTCLUS procedure can use an L_p (least pth powers) clustering criterion (Spath 1985, pp. 62–63) instead of the least squares (L_2) criterion used in k-means clustering methods. The LEAST=p option specifies the power p to be used. Using the LEAST= option increases execution time since more iterations are usually required, and the default iteration limit is increased when you specify LEAST=p. Values of p less than 2 reduce the effect of outliers on the cluster centers compared with least squares methods; values of p greater than 2 increase the effect of outliers.

The FASTCLUS procedure is intended for use with large data sets, with 100 or more observations. With small data sets, the results can be highly sensitive to the order of the observations in the data set.

PROC FASTCLUS uses algorithms that place a larger influence on variables with larger variance, so it might be necessary to standardize the variables before performing the cluster analysis. See the "Using PROC FASTCLUS" section for standardization details.

PROC FASTCLUS produces brief summaries of the clusters it finds. For more extensive examination of the clusters, you can request an output data set containing a cluster membership variable.

Background

The FASTCLUS procedure combines an effective method for finding initial clusters with a standard iterative algorithm for minimizing the sum of squared distances from the cluster means. The result is an efficient procedure for disjoint clustering of large data sets. PROC FASTCLUS was directly inspired by Hartigan's (1975) *leader algorithm* and MacQueen's (1967) *k-means algorithm*. PROC FASTCLUS uses a method that Anderberg (1973) calls *nearest centroid sorting*. A set of points called *cluster seeds* is selected as a first guess of the means of the clusters. Each observation is assigned to the nearest seed to form temporary clusters. The seeds are then replaced by the means

of the temporary clusters, and the process is repeated until no further changes occur in the clusters. Similar techniques are described in most references on clustering (Anderberg 1973; Hartigan 1975; Everitt 1980; Spath 1980).

The FASTCLUS procedure differs from other nearest centroid sorting methods in the way the initial cluster seeds are selected. The importance of initial seed selection is demonstrated by Milligan (1980).

The clustering is done on the basis of Euclidean distances computed from one or more numeric variables. If there are missing values, PROC FASTCLUS computes an adjusted distance by using the nonmissing values. Observations that are very close to each other are usually assigned to the same cluster, while observations that are far apart are in different clusters.

The FASTCLUS procedure operates in four steps:

1. Observations called *cluster seeds* are selected.
2. If you specify the DRIFT option, temporary clusters are formed by assigning each observation to the cluster with the nearest seed. Each time an observation is assigned, the cluster seed is updated as the current mean of the cluster. This method is sometimes called *incremental*, *on-line*, or *adaptive* training.
3. If the maximum number of iterations is greater than zero, clusters are formed by assigning each observation to the nearest seed. After all observations are assigned, the cluster seeds are replaced by either the cluster means or other location estimates (cluster centers) appropriate to the LEAST=p option. This step can be repeated until the changes in the cluster seeds become small or zero (MAXITER=$n \geq 1$).
4. Final clusters are formed by assigning each observation to the nearest seed.

If PROC FASTCLUS runs to complete convergence, the final cluster seeds will equal the cluster means or cluster centers. If PROC FASTCLUS terminates before complete convergence, which often happens with the default settings, the final cluster seeds might not equal the cluster means or cluster centers. If you want complete convergence, specify CONVERGE=0 and a large value for the MAXITER= option.

The initial cluster seeds must be observations with no missing values. You can specify the maximum number of seeds (and, hence, clusters) by using the MAXCLUSTERS= option. You can also specify a minimum distance by which the seeds must be separated by using the RADIUS= option.

PROC FASTCLUS always selects the first complete (no missing values) observation as the first seed. The next complete observation that is separated from the first seed by at least the distance specified in the RADIUS= option becomes the second seed. Later observations are selected as new seeds if they are separated from all previous seeds by at least the radius, as long as the maximum number of seeds is not exceeded.

If an observation is complete but fails to qualify as a new seed, PROC FASTCLUS considers using it to replace one of the old seeds. Two tests are made to see if the observation can qualify as a new seed.

First, an old seed is replaced if the distance between the observation and the closest seed is greater than the minimum distance between seeds. The seed that is replaced is selected from the two

seeds that are closest to each other. The seed that is replaced is the one of these two with the shortest distance to the closest of the remaining seeds when the other seed is replaced by the current observation.

If the observation fails the first test for seed replacement, a second test is made. The observation replaces the nearest seed if the smallest distance from the observation to all seeds other than the nearest one is greater than the shortest distance from the nearest seed to all other seeds. If the observation fails this test, PROC FASTCLUS goes on to the next observation.

You can specify the REPLACE= option to limit seed replacement. You can omit the second test for seed replacement (REPLACE=PART), causing PROC FASTCLUS to run faster, but the seeds selected might not be as widely separated as those obtained by the default method. You can also suppress seed replacement entirely by specifying REPLACE=NONE. In this case, PROC FASTCLUS runs much faster, but you must choose a good value for the RADIUS= option in order to get good clusters. This method is similar to Hartigan's (1975, pp. 74–78) leader algorithm and the *simple cluster seeking algorithm* described by Tou and Gonzalez (1974, pp. 90–92).

Getting Started: FASTCLUS Procedure

The following example demonstrates how to use the FASTCLUS procedure to compute disjoint clusters of observations in a SAS data set.

The data in this example are measurements taken on 159 freshwater fish caught from the same lake (Laengelmavesi) near Tampere in Finland. This data set is available from the Data Archive of the *Journal of Statistics Education*. The complete data set is displayed in Chapter 82, "The STEPDISC Procedure."

The species (bream, parkki, pike, perch, roach, smelt, and whitefish), weight, three different length measurements (measured from the nose of the fish to the beginning of its tail, the notch of its tail, and the end of its tail), height, and width of each fish are tallied. The height and width are recorded as percentages of the third length variable.

Suppose that you want to group empirically the fish measurements into clusters and that you want to associate the clusters with the species. You can use the FASTCLUS procedure to perform a cluster analysis.

The following DATA step creates the SAS data set Fish:

```
proc format;
   value specfmt
      1='Bream'
      2='Roach'
      3='Whitefish'
      4='Parkki'
      5='Perch'
      6='Pike'
      7='Smelt';
run;
```

```
data fish (drop=HtPct WidthPct);
   title 'Fish Measurement Data';
   input Species Weight Length1 Length2 Length3 HtPct WidthPct @@;

   *** transform variables;
   if Weight <= 0 or Weight =. then delete;
   Weight3=Weight**(1/3);
   Height=HtPct*Length3/(Weight3*100);
   Width=WidthPct*Length3/(Weight3*100);
   Length1=Length1/Weight3;
   Length2=Length2/Weight3;
   Length3=Length3/Weight3;
   logLengthRatio=log(Length3/Length1);

   format Species specfmt.;
   symbol = put(Species, specfmt2.);
   datalines;
1  242.0 23.2 25.4 30.0 38.4 13.4 1  290.0 24.0 26.3 31.2 40.0 13.8
1  340.0 23.9 26.5 31.1 39.8 15.1 1  363.0 26.3 29.0 33.5 38.0 13.3
1  430.0 26.5 29.0 34.0 36.6 15.1 1  450.0 26.8 29.7 34.7 39.2 14.2
1  500.0 26.8 29.7 34.5 41.1 15.3 1  390.0 27.6 30.0 35.0 36.2 13.4
1  450.0 27.6 30.0 35.1 39.9 13.8 1  500.0 28.5 30.7 36.2 39.3 13.7
1  475.0 28.4 31.0 36.2 39.4 14.1 1  500.0 28.7 31.0 36.2 39.7 13.3
1  500.0 29.1 31.5 36.4 37.8 12.0 1     .  29.5 32.0 37.3 37.3 13.6
1  600.0 29.4 32.0 37.2 40.2 13.9 1  600.0 29.4 32.0 37.2 41.5 15.0

... more lines ...

7    9.8 11.4 12.0 13.2 16.7  8.7 7   12.2 11.5 12.2 13.4 15.6 10.4
7   13.4 11.7 12.4 13.5 18.0  9.4 7   12.2 12.1 13.0 13.8 16.5  9.1
7   19.7 13.2 14.3 15.2 18.9 13.6 7   19.9 13.8 15.0 16.2 18.1 11.6
;
```

The double trailing at sign (@@) in the INPUT statement specifies that observations are input from each line until all values are read. The variables are rescaled in order to adjust for dimensionality. Because the new variables Weight3–logLengthRatio depend on the variable Weight, observations with missing values for Weight are not added to the data set. Consequently, there are 157 observations in the SAS data set Fish.

In the Fish data set, the variables are not measured in the same units and cannot be assumed to have equal variance. Therefore, it is necessary to standardize the variables before performing the cluster analysis.

The following statements standardize the variables and perform a cluster analysis on the standardized data:

```
proc standard data=Fish out=Stand mean=0 std=1;
   var Length1 logLengthRatio Height Width Weight3;
proc fastclus data=Stand out=Clust
              maxclusters=7 maxiter=100 ;
   var Length1 logLengthRatio Height Width Weight3;
run;
```

The STANDARD procedure is first used to standardize all the analytical variables to a mean of 0 and standard deviation of 1. The procedure creates the output data set Stand to contain the transformed variables.

The FASTCLUS procedure then uses the data set Stand as input and creates the data set Clust. This output data set contains the original variables and two new variables, Cluster and Distance. The variable Cluster contains the cluster number to which each observation has been assigned. The variable Distance gives the distance from the observation to its cluster seed.

It is usually desirable to try several values of the MAXCLUSTERS= option. A reasonable beginning for this example is to use MAXCLUSTERS=7, since there are seven species of fish represented in the data set Fish.

The VAR statement specifies the variables used in the cluster analysis.

The results from this analysis are displayed in the following figures.

Figure 34.1 Initial Seeds Used in the FASTCLUS Procedure

Fish Measurement Data

The FASTCLUS Procedure

Replace=FULL Radius=0 Maxclusters=7 Maxiter=100 Converge=0.02

Initial Seeds

Cluster	Length1	logLength Ratio	Height	Width	Weight3
1	1.388338414	−0.979577858	−1.594561848	−2.254050655	2.103447062
2	−1.117178039	−0.877218192	−0.336166276	2.528114070	1.170706464
3	2.393997461	−0.662642015	−0.930738701	−2.073879107	−1.839325419
4	−0.495085516	−0.964041012	−0.265106856	−0.028245072	1.536846394
5	−0.728772773	0.540096664	1.130501398	−1.207930053	−1.107018207
6	−0.506924177	0.748211648	1.762482687	0.211507596	1.368987826
7	1.573996573	−0.796593995	−0.824217424	1.561715851	−1.607942726

Criterion Based on Final Seeds = 0.3979

Figure 34.1 displays the table of initial seeds used for each variable and cluster. The first line in the figure displays the option settings for REPLACE, RADIUS, MAXCLUSTERS, and MAXITER. These options, with the exception of MAXCLUSTERS and MAXITER, are set at their respective default values (REPLACE=FULL, RADIUS=0). Both the MAXCLUSTERS= and MAXITER= options are set in the PROC FASTCLUS statement.

Next, PROC FASTCLUS produces a table of summary statistics for the clusters. Figure 34.2 displays the number of observations in the cluster (frequency) and the root mean squared standard deviation. The next two columns display the largest Euclidean distance from the cluster seed to any observation within the cluster and the number of the nearest cluster.

The last column of the table displays the distance between the centroid of the nearest cluster and the centroid of the current cluster. A centroid is the point having coordinates that are the means of all the observations in the cluster.

Figure 34.2 Cluster Summary Table from the FASTCLUS Procedure

Cluster Summary

Cluster	Frequency	RMS Std Deviation	Maximum Distance from Seed to Observation	Radius Exceeded	Nearest Cluster
1	17	0.5064	1.7781		4
2	19	0.3696	1.5007		4
3	13	0.3803	1.7135		1
4	13	0.4161	1.3976		7
5	11	0.2466	0.6966		6
6	34	0.3563	1.5443		5
7	50	0.4447	2.3915		4

Cluster Summary

Cluster	Distance Between Cluster Centroids
1	2.5106
2	1.5510
3	2.6704
4	1.4266
5	1.7301
6	1.7301
7	1.4266

Figure 34.3 displays the table of statistics for the variables. The table lists for each variable the total standard deviation, the pooled within-cluster standard deviation and the R-square value for predicting the variable from the cluster. The ratio of between-cluster variance to within-cluster variance (R^2 to $1 - R^2$) appears in the last column.

Figure 34.3 Statistics for Variables Used in the FASTCLUS Procedure

Statistics for Variables

Variable	Total STD	Within STD	R-Square	RSQ/(1-RSQ)
Length1	1.00000	0.31428	0.905030	9.529606
logLengthRatio	1.00000	0.39276	0.851676	5.741989
Height	1.00000	0.20917	0.957929	22.769295
Width	1.00000	0.55558	0.703200	2.369270
Weight3	1.00000	0.47251	0.785323	3.658162
OVER-ALL	1.00000	0.40712	0.840631	5.274764

Pseudo F Statistic = 131.87

Approximate Expected Over-All R-Squared = 0.57420

The pseudo *F* statistic, approximate expected overall R square, and cubic clustering criterion (CCC) are listed at the bottom of the figure. You can compare values of these statistics by running PROC FASTCLUS with different values for the MAXCLUSTERS= option. The R square and CCC values are not valid for correlated variables.

Values of the cubic clustering criterion greater than 2 or 3 indicate good clusters. Values between 0 and 2 indicate potential clusters, but they should be taken with caution; large negative values can indicate outliers.

PROC FASTCLUS next produces the within-cluster means and standard deviations of the variables, displayed in Figure 34.4.

Figure 34.4 Cluster Means and Standard Deviations from the FASTCLUS Procedure

Cluster Means

Cluster	Length1	logLength Ratio	Height	Width	Weight3
1	1.747808245	-0.868605685	-1.327226832	-1.128760946	0.806373599
2	-0.405231510	-0.979113021	-0.281064162	1.463094486	1.060450065
3	2.006796315	-0.652725165	-1.053213440	-1.224020795	-1.826752838
4	-0.136820952	-1.039312574	-0.446429482	0.162596336	0.278560318
5	-0.850130601	0.550190242	1.245156076	-0.836585750	-0.567022647
6	-0.843912827	1.522291347	1.511408739	-0.380323563	0.763114370
7	-0.165570970	-0.048881276	-0.353723615	0.546442064	-0.668780782

Cluster Standard Deviations

Cluster	Length1	logLength Ratio	Height	Width	Weight3
1	0.3418476428	0.3544065543	0.1666302451	0.6172880027	0.7944227150
2	0.3129902863	0.3592350778	0.1369052680	0.5467406493	0.3720119097
3	0.2962504486	0.1740941675	0.1736086707	0.7528475622	0.0905232968
4	0.3254364840	0.2836681149	0.1884592934	0.4543390702	0.6612055341
5	0.1781837609	0.0745984121	0.2056932592	0.2784540794	0.3832002850
6	0.2273744242	0.3385584051	0.2046010964	0.5143496067	0.4025849044
7	0.3734733622	0.5275768119	0.2551130680	0.5721303628	0.4223181710

It is useful to study further the clusters calculated by the FASTCLUS procedure. One method is to look at a frequency tabulation of the clusters with other classification variables. The following statements invoke the FREQ procedure to crosstabulate the empirical clusters with the variable Species:

```
proc freq data=Clust;
   tables Species*Cluster;
run;
```

Figure 34.5 displays the marked division between clusters.

Figure 34.5 Frequency Table of Cluster versus Species

Fish Measurement Data

The FREQ Procedure

Table of Species by CLUSTER

Species CLUSTER(Cluster)

Frequency / Percent / Row Pct / Col Pct	1	2	3	4	Total
Bream	0	0	0	0	34
	0.00	0.00	0.00	0.00	21.66
	0.00	0.00	0.00	0.00	
	0.00	0.00	0.00	0.00	
Roach	0	0	0	0	19
	0.00	0.00	0.00	0.00	12.10
	0.00	0.00	0.00	0.00	
	0.00	0.00	0.00	0.00	
Whitefish	0	2	0	1	6
	0.00	1.27	0.00	0.64	3.82
	0.00	33.33	0.00	16.67	
	0.00	10.53	0.00	7.69	
Parkki	0	0	0	0	11
	0.00	0.00	0.00	0.00	7.01
	0.00	0.00	0.00	0.00	
	0.00	0.00	0.00	0.00	
Perch	0	17	0	12	56
	0.00	10.83	0.00	7.64	35.67
	0.00	30.36	0.00	21.43	
	0.00	89.47	0.00	92.31	
Pike	17	0	0	0	17
	10.83	0.00	0.00	0.00	10.83
	100.00	0.00	0.00	0.00	
	100.00	0.00	0.00	0.00	
Smelt	0	0	13	0	14
	0.00	0.00	8.28	0.00	8.92
	0.00	0.00	92.86	0.00	
	0.00	0.00	100.00	0.00	
Total	17	19	13	13	157
	10.83	12.10	8.28	8.28	100.00

(Continued)

Figure 34.5 *continued*

Fish Measurement Data

The FREQ Procedure

Table of Species by CLUSTER

Species CLUSTER(Cluster)

Frequency / Percent / Row Pct / Col Pct	5	6	7	Total
Bream	0	34	0	34
	0.00	21.66	0.00	21.66
	0.00	100.00	0.00	
	0.00	100.00	0.00	
Roach	0	0	19	19
	0.00	0.00	12.10	12.10
	0.00	0.00	100.00	
	0.00	0.00	38.00	
Whitefish	0	0	3	6
	0.00	0.00	1.91	3.82
	0.00	0.00	50.00	
	0.00	0.00	6.00	
Parkki	11	0	0	11
	7.01	0.00	0.00	7.01
	100.00	0.00	0.00	
	100.00	0.00	0.00	
Perch	0	0	27	56
	0.00	0.00	17.20	35.67
	0.00	0.00	48.21	
	0.00	0.00	54.00	
Pike	0	0	0	17
	0.00	0.00	0.00	10.83
	0.00	0.00	0.00	
	0.00	0.00	0.00	
Smelt	0	0	1	14
	0.00	0.00	0.64	8.92
	0.00	0.00	7.14	
	0.00	0.00	2.00	
Total	11	34	50	157
	7.01	21.66	31.85	100.00

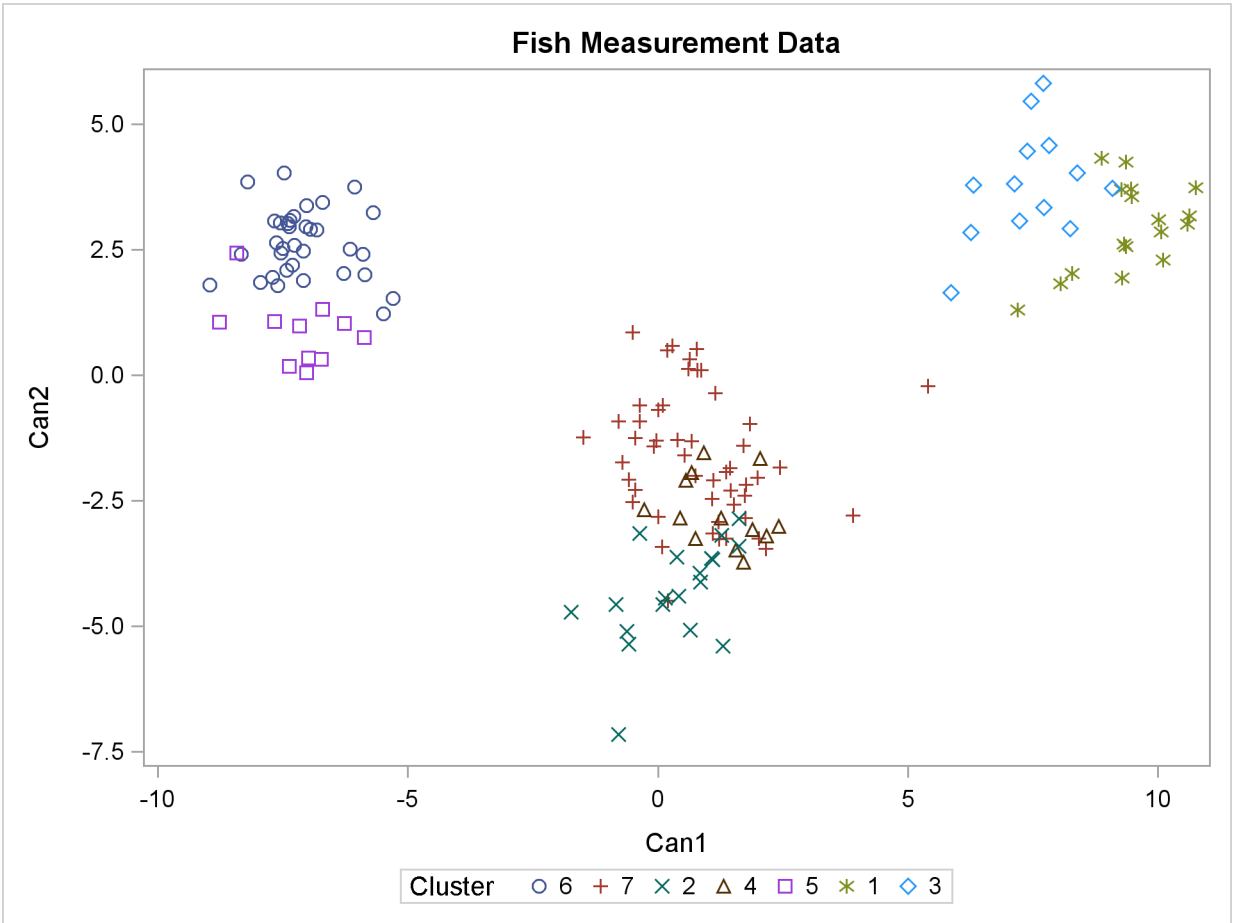
For cases in which you have three or more clusters, you can use the CANDISC and SGPLOT procedures to obtain a graphical check on the distribution of the clusters. In the following statements, the CANDISC and SGPLOT procedures are used to compute canonical variables and plot the clusters:

```
proc candisc data=Clust out=Can noprint;
   class Cluster;
   var Length1 logLengthRatio Height Width Weight3;

proc sgplot data=Can;
   scatter y=Can2 x=Can1 / group=Cluster ;
run;
```

First, the CANDISC procedure is invoked to perform a canonical discriminant analysis by using the data set Clust and creating the output SAS data set Can. The NOPRINT option suppresses display of the output. The CLASS statement specifies the variable Cluster to define groups for the analysis. The VAR statement specifies the variables used in the analysis.

Next, the SGPLOT procedure plots the two canonical variables from PROC CANDISC, Can1 and Can2. The PLOT statement specifies the variable Cluster as the identification variable. The resulting plot (Figure 34.6) illustrates the spatial separation of the clusters calculated in the FASTCLUS procedure.

Figure 34.6 Plot of Canonical Variables and Cluster Value



Syntax: FASTCLUS Procedure

The following statements are available in the FASTCLUS procedure:

PROC FASTCLUS <**DATA=***SAS-data-set*>
<**MAXCLUSTERS=***n*>
<**RADIUS=***t*> ;
VAR *variables* ;
ID *variables* ;
FREQ *variable* ;
WEIGHT *variable* ;
BY *variables* ;

Usually you need only the VAR statement in addition to the PROC FASTCLUS statement. The BY, FREQ, ID, VAR, and WEIGHT statements are described in alphabetical order after the PROC FASTCLUS statement.

PROC FASTCLUS Statement

PROC FASTCLUS MAXCLUSTERS= *n* | *RADIUS=t* <*options*> ;

You must specify the MAXCLUSTERS= option or RADIUS= option or both in the PROC FASTCLUS statement.

MAXCLUSTERS=*n*
MAXC=*n*
specifies the maximum number of clusters permitted. If you omit the MAXCLUSTERS= option, a value of 100 is assumed.

RADIUS=*t*
R=*t*
establishes the minimum distance criterion for selecting new seeds. No observation is considered as a new seed unless its minimum distance to previous seeds exceeds the value given by the RADIUS= option. The default value is 0. If you specify the REPLACE=RANDOM option, the RADIUS= option is ignored.

You can specify the following options in the PROC FASTCLUS statement. Table 34.1 summarizes the options.

Table 34.1 PROC FASTCLUS Statement Options

Option	Description
Specify input and output data sets	
DATA=	specifies input data set
INSTAT=	specifies input SAS data set previously created by the OUTSTAT= option
SEED=	specifies input SAS data set for selecting initial cluster seeds
VARDEF=	specifies divisor for variances
Output Data Processing	
CLUSTER=	specifies name for cluster membership variable in OUTSEED= and OUT= data sets
CLUSTERLABEL=	specifies label for cluster membership variable in OUTSEED= and OUT= data sets
OUT=	specifies output SAS data set containing original data and cluster assignments
OUTITER	specifies writing to OUTSEED= data set on every iteration
OUTSEED= or MEAN=	specifies output SAS data set containing cluster centers
OUTSTAT=	specifies output SAS data set containing statistics
Initial Clusters	
DRIFT	permits cluster to seeds to drift during initialization
MAXCLUSTERS=	specifies maximum number of clusters
RADIUS=	specifies minimum distance for selecting new seeds
RANDOM=	specifies seed to initializes pseudo-random number generator
REPLACE=	specifies seed replacement method
Clustering Methods	
CONVERGE=	specifies convergence criterion
DELETE=	deletes cluster seeds with few observations
LEAST=	optimizes an L_p criterion, where $1 \leq p \leq \infty$
MAXITER=	specifies maximum number of iterations
STRICT	prevents an observation from being assigned to a cluster if its distance to the nearest cluster seed is large

Table 34.1 *continued*

Option	Description
Arcane Algorithmic Options	
BINS=	specifies number of bins used for computing medians for LEAST=1
HC=	specifies criterion for updating the homotopy parameter
HP=	specifies initial value of the homotopy parameter
IRLS	uses an iteratively reweighted least squares method instead of the modified Ekblom-Newton method for $1 < p < 2$
Missing Values	
IMPUTE	imputes missing values after final cluster assignment
NOMISS	excludes observations with missing values
Control Displayed Output	
DISTANCE	displays distances between cluster centers
LIST	displays cluster assignments for all observations
NOPRINT	suppresses displayed output
SHORT	suppresses display of large matrices
SUMMARY	suppresses display of all results except for the cluster summary

The following list provides details on these options. The list is in alphabetical order.

BINS=*n*

specifies the number of bins used in the bin-sort algorithm for computing medians for LEAST=1. By default, PROC FASTCLUS uses from 10 to 100 bins, depending on the amount of memory available. Larger values use more memory and make each iteration somewhat slower, but they can reduce the number of iterations. Smaller values have the opposite effect. The minimum value of *n* is 5.

CLUSTER=*name*

specifies a name for the variable in the OUTSEED= and OUT= data sets that indicates cluster membership. The default name for this variable is CLUSTER.

CLUSTERLABEL=*name*

specifies a label for the variable CLUSTER in the OUTSEED= and OUT= data sets. By default this variable has no label.

CONVERGE=*c*

CONV=*c*

specifies the convergence criterion. Any nonnegative value is permitted. The default value is 0.0001 for all values of *p* if LEAST=*p* is explicitly specified; otherwise, the default value is 0.02. Iterations stop when the maximum relative change in the cluster seeds is less than or

equal to the convergence criterion and additional conditions on the homotopy parameter, if any, are satisfied (see the HP= option). The relative change in a cluster seed is the distance between the old seed and the new seed divided by a scaling factor. If you do not specify the LEAST= option, the scaling factor is the minimum distance between the initial seeds. If you specify the LEAST= option, the scaling factor is an L_1 scale estimate and is recomputed on each iteration. Specify the CONVERGE= option only if you specify a MAXITER= value greater than 1.

DATA=*SAS-data-set*

specifies the input data set containing observations to be clustered. If you omit the DATA= option, the most recently created SAS data set is used. The data must be coordinates, not distances, similarities, or correlations.

DELETE=*n*

deletes cluster seeds to which n or fewer observations are assigned. Deletion occurs after processing for the DRIFT option is completed and after each iteration specified by the MAXITER= option. Cluster seeds are not deleted after the final assignment of observations to clusters, so in rare cases a final cluster might not have more than n members. The DELETE= option is ineffective if you specify MAXITER=0 and do not specify the DRIFT option. By default, no cluster seeds are deleted.

DISTANCE | DIST

computes distances between the cluster means.

DRIFT

executes the second of the four steps described in the section "Background" on page 1622. After initial seed selection, each observation is assigned to the cluster with the nearest seed. After an observation is processed, the seed of the cluster to which it is assigned is recalculated as the mean of the observations currently assigned to the cluster. Thus, the cluster seeds drift about rather than remaining fixed for the duration of the pass.

HC=*c*

HP=*p_1* **<** *p_2* **>**

pertains to the homotopy parameter for LEAST=p, where $1 < p < 2$. You should specify these options only if you encounter convergence problems when you use the default values.

For $1 < p < 2$, PROC FASTCLUS tries to optimize a perturbed variant of the L_p clustering criterion (Gonin and Money 1989, pp. 5–6). When the homotopy parameter is 0, the optimization criterion is equivalent to the clustering criterion. For a large homotopy parameter, the optimization criterion approaches the least squares criterion and is therefore easy to optimize. Beginning with a large homotopy parameter, PROC FASTCLUS gradually decreases it by a factor in the range [0.01,0.5] over the course of the iterations. When both the homotopy parameter and the convergence measure are sufficiently small, the optimization process is declared to have converged.

If the initial homotopy parameter is too large or if it is decreased too slowly, the optimization can require many iterations. If the initial homotopy parameter is too small or if it is decreased too quickly, convergence to a local optimum is likely. The following list gives details on setting the homotopy parameter.

HC=c specifies the criterion for updating the homotopy parameter. The homotopy parameter is updated when the maximum relative change in the cluster seeds is less than or equal to c. The default is the minimum of 0.01 and 100 times the value of the CONVERGE= option.

HP=p_1 specifies p_1 as the initial value of the homotopy parameter. The default is 0.05 if the modified Ekblom-Newton method is used; otherwise, it is 0.25.

HP=p_1 p_2 also specifies p_2 as the minimum value for the homotopy parameter, which must be reached for convergence. The default is the minimum of p_1 and 0.01 times the value of the CONVERGE= option.

IMPUTE

requests imputation of missing values after the final assignment of observations to clusters. If an observation that is assigned (or would have been assigned) to a cluster has a missing value for variables used in the cluster analysis, the missing value is replaced by the corresponding value in the cluster seed to which the observation is assigned (or would have been assigned). If the observation cannot be assigned to a cluster, missing value replacement depends on whether or not the NOMISS option is specified. If NOMISS is not specified, missing values are replaced by the mean of all observations in the DATA= data set having a value for that variable. If NOMISS is specified, missing values are replace by the mean of only observations used in the analysis. (A weighted mean is used if a variable is specified in the WEIGHT statement.) For information about cluster assignment see the section "OUT= Data Set" on page 1643. If you specify the IMPUTE option, the imputed values are not used in computing cluster statistics.

If you also request an OUT= data set, it contains the imputed values.

INSTAT=*SAS-data-set*

reads a SAS data set previously created with the FASTCLUS procedure by using the OUTSTAT= option. If you specify the INSTAT= option, no clustering iterations are performed and no output is displayed. Only cluster assignment and imputation are performed as an OUT= data set is created.

IRLS

causes PROC FASTCLUS to use an iteratively reweighted least squares method instead of the modified Ekblom-Newton method. If you specify the IRLS option, you must also specify LEAST=p, where $1 < p < 2$. Use the IRLS option only if you encounter convergence problems with the default method.

LEAST=*p* | **MAX**

L=*p* | **MAX**

causes PROC FASTCLUS to optimize an L_p criterion, where $1 \leq p \leq \infty$ (Spath 1985, pp. 62–63). Infinity is indicated by LEAST=MAX. The value of this clustering criterion is displayed in the iteration history.

If you do not specify the LEAST= option, PROC FASTCLUS uses the least squares (L_2) criterion. However, the default number of iterations is only 1 if you omit the LEAST= option, so the optimization of the criterion is generally not completed. If you specify the LEAST= option, the maximum number of iterations is increased to permit the optimization process a chance to converge. See the MAXITER= option for details.

Specifying the LEAST= option also changes the default convergence criterion from 0.02 to 0.0001. See the CONVERGE= option for details.

When LEAST=2, PROC FASTCLUS tries to minimize the root mean squared difference between the data and the corresponding cluster means.

When LEAST=1, PROC FASTCLUS tries to minimize the mean absolute difference between the data and the corresponding cluster medians.

When LEAST=MAX, PROC FASTCLUS tries to minimize the maximum absolute difference between the data and the corresponding cluster midranges.

For general values of p, PROC FASTCLUS tries to minimize the pth root of the mean of the pth powers of the absolute differences between the data and the corresponding cluster seeds.

The divisor in the clustering criterion is either the number of nonmissing data used in the analysis or, if there is a WEIGHT statement, the sum of the weights corresponding to all the nonmissing data used in the analysis (that is, an observation with n nonmissing data contributes n times the observation weight to the divisor). The divisor is not adjusted for degrees of freedom.

The method for updating cluster seeds during iteration depends on the LEAST= option, as follows (Gonin and Money 1989).

LEAST=p	Algorithm for Computing Cluster Seeds
$p = 1$	bin sort for median
$1 < p < 2$	modified Merle-Spath if you specify IRLS; otherwise modified Ekblom-Newton
$p = 2$	arithmetic mean
$2 < p < \infty$	Newton
$p = \infty$	midrange

During the final pass, a modified Merle-Spath step is taken to compute the cluster centers for $1 \leq p < 2$ or $2 < p < \infty$.

If you specify the LEAST=p option with a value other than 2, PROC FASTCLUS computes pooled scale estimates analogous to the root mean squared standard deviation but based on pth power deviations instead of squared deviations.

LEAST=p	Scale Estimate
$p = 1$	mean absolute deviation
$1 < p < \infty$	root mean pth-power absolute deviation
$p = \infty$	maximum absolute deviation

The divisors for computing the mean absolute deviation or the root mean pth-power absolute deviation are adjusted for degrees of freedom just like the divisors for computing standard deviations. This adjustment can be suppressed by the VARDEF= option.

LIST

lists all observations, giving the value of the ID variable (if any), the number of the cluster to which the observation is assigned, and the distance between the observation and the final cluster seed.

MAXITER=*n*

specifies the maximum number of iterations for recomputing cluster seeds. When the value of the MAXITER= option is greater than zero, PROC FASTCLUS executes the third of the four steps described in the section "Background" on page 1622. In each iteration, each observation is assigned to the nearest seed, and the seeds are recomputed as the means of the clusters.

The default value of the MAXITER= option depends on the LEAST=p option.

LEAST=p	MAXITER=
not specified	1
$p = 1$	20
$1 < p < 1.5$	50
$1.5 \leq p < 2$	20
$p = 2$	10
$2 < p \leq \infty$	20

MEAN=*SAS-data-set*

creates an output data set to contain the cluster means and other statistics for each cluster. If you want to create a permanent SAS data set, you must specify a two-level name. See "SAS Data Files" in *SAS Language Reference: Concepts* for more information about permanent data sets.

NOMISS

excludes observations with missing values from the analysis. However, if you also specify the IMPUTE option, observations with missing values are included in the final cluster assignments.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 20, "Using the Output Delivery System."

OUT=*SAS-data-set*

creates an output data set to contain all the original data, plus the new variables CLUSTER and DISTANCE. See "SAS Data Files" in *SAS Language Reference: Concepts* for more information about permanent data sets.

OUTITER

outputs information from the iteration history to the OUTSEED= data set, including the cluster seeds at each iteration.

OUTSEED=*SAS-data-set*

OUTS=*SAS-data-set*

is another name for the MEAN= data set, provided because the data set can contain location estimates other than means. The MEAN= option is still accepted.

OUTSTAT=*SAS-data-set*

creates an output data set to contain various statistics, especially those not included in the OUTSEED= data set. Unlike the OUTSEED= data set, the OUTSTAT= data set is not suitable for use as a SEED= data set in a subsequent PROC FASTCLUS step.

RANDOM=*n*

specifies a positive integer as a starting value for the pseudo-random number generator for use with REPLACE=RANDOM. If you do not specify the RANDOM= option, the time of day is used to initialize the pseudo-random number sequence.

REPLACE=FULL | PART | NONE | RANDOM

specifies how seed replacement is performed, as follows:

FULL	requests default seed replacement as described in the section "Background" on page 1622.
PART	requests seed replacement only when the distance between the observation and the closest seed is greater than the minimum distance between seeds.
NONE	suppresses seed replacement.
RANDOM	selects a simple pseudo-random sample of complete observations as initial cluster seeds.

SEED=*SAS-data-set*

specifies an input data set from which initial cluster seeds are to be selected. If you do not specify the SEED= option, initial seeds are selected from the DATA= data set. The SEED= data set must contain the same variables that are used in the data analysis.

SHORT

suppresses the display of the initial cluster seeds, cluster means, and standard deviations.

STRICT

STRICT=*s*

prevents an observation from being assigned to a cluster if its distance to the nearest cluster seed exceeds the value of the STRICT= option. If you specify the STRICT option without a numeric value, you must also specify the RADIUS= option, and its value is used instead. In the OUT= data set, observations that are not assigned due to the STRICT= option are given a negative cluster number, the absolute value of which indicates the cluster with the nearest seed.

SUMMARY

suppresses the display of the initial cluster seeds, statistics for variables, cluster means, and standard deviations.

VARDEF=DF | N | WDF | WEIGHT | WGT

specifies the divisor to be used in the calculation of variances and covariances. The default value is VARDEF=DF. The possible values of the VARDEF= option and associated divisors are as follows.

Value	Description	Divisor
DF	error degrees of freedom	$n - c$
N	number of observations	n
WDF	sum of weights DF	$(\sum_i w_i) - c$
WEIGHT | WGT	sum of weights	$\sum_i w_i$

In the preceding definitions, c represents the number of clusters.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC FASTCLUS to obtain separate analysis on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the FASTCLUS procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

If you specify the SEED= option and the SEED= data set does not contain any of the BY variables, then the entire SEED= data set is used to obtain initial cluster seeds for each BY group in the DATA= data set.

If the SEED= data set contains some but not all of the BY variables, or if some BY variables do not have the same type or length in the SEED= data set as in the DATA= data set, then PROC FASTCLUS displays an error message and stops.

If all the BY variables appear in the SEED= data set with the same type and length as in the DATA= data set, then each BY group in the SEED= data set is used to obtain initial cluster seeds for the corresponding BY group in the DATA= data set. All BY groups in the DATA= data set must also appear in the SEED= data set. The BY groups in the SEED= data set must be in the same order as in the DATA= data set. If you specify the NOTSORTED option in the BY statement, both data sets must contain exactly the same BY groups in the same order. If you do not specify NOTSORTED, some BY groups can appear in the SEED= data set but not in the DATA= data set; such BY groups are not used in the analysis.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

FREQ Statement

FREQ *variable* **;**

If a variable in the data set represents the frequency of occurrence for the other values in the observation, include the variable's name in a FREQ statement. The procedure then treats the data set as if each observation appears n times, where n is the value of the FREQ variable for the observation.

If the value of the FREQ variable is missing or less than or equal to zero, the observation is not used in the analysis. The exact values of the FREQ variable are used in computations: frequency values are not truncated to integers. The total number of observations is considered to be equal to the sum of the FREQ variable when the procedure determines degrees of freedom for significance probabilities.

The WEIGHT and FREQ statements have a similar effect, except in determining the number of observations for significance tests.

ID Statement

ID *variable* **;**

The ID variable, which can be character or numeric, identifies observations on the output when you specify the LIST option.

VAR Statement

VAR *variables* **;**

The VAR statement lists the numeric variables to be used in the cluster analysis. If you omit the VAR statement, all numeric variables not listed in other statements are used.

WEIGHT Statement

WEIGHT *variable* **;**

The values of the WEIGHT variable are used to compute weighted cluster means. The WEIGHT and FREQ statements have a similar effect, except the WEIGHT statement does not alter the degrees of freedom or the number of observations. The WEIGHT variable can take nonintegral values. An observation is used in the analysis only if the value of the WEIGHT variable is greater than zero.

Details: FASTCLUS Procedure

Updates in the FASTCLUS Procedure

Some FASTCLUS procedure options and statements have changed from previous versions. The differences are as follows:

- Values of the FREQ variable are no longer truncated to integers. Noninteger variables specified in the FREQ statement produce results different from those in previous releases.
- The IMPUTE option produces different cluster standard deviations and related statistics. When you specify the IMPUTE option, imputed values are no longer used in computing cluster statistics. This change causes the cluster standard deviations and other statistics computed from the standard deviations to be different from those in previous releases.
- The INSTAT= option reads a SAS data set previously created with the FASTCLUS procedure by using the OUTSTAT= option. If you specify the INSTAT= option, no clustering iterations are performed and no output is produced. Only cluster assignment and imputation are performed as an OUT= data set is created.
- The OUTSTAT= data set contains additional information used for imputation. _TYPE_=SEED corresponds to values that are cluster seeds. Observations previously designated _TYPE_='SCALE' are now _TYPE_='DISPERSION'.

Missing Values

Observations with all missing values are excluded from the analysis. If you specify the NOMISS option, observations with any missing values are excluded. Observations with missing values cannot be cluster seeds.

The distance between an observation with missing values and a cluster seed is obtained by computing the squared distance based on the nonmissing values, multiplying by the ratio of the number of variables, n, to the number of variables having nonmissing values, m, and taking the square root:

$$\sqrt{\left(\frac{n}{m}\right)\sum(x_i - s_i)^2}$$

where

n = number of variables
m = number of variables with nonmissing values
x_i = value of the ith variable for the observation
s_i = value of the ith variable for the seed

If you specify the LEAST=p option with a power p other than 2 (the default), the distance is computed using

$$\left(\left(\frac{n}{m}\right)\sum(x_i - s_i)^p\right)^{\frac{1}{p}}$$

The summation is taken over variables with nonmissing values.

The IMPUTE option fills in missing values in the OUT= output data set.

Output Data Sets

OUT= Data Set

The OUT= data set contains the following:

- the original variables
- a new variable indicating the cluster assignment status of each observation. The value will be less than the permitted number of clusters (see the MAXCLUSTERS= option) if the procedure detects fewer clusters than the maximum. A positive value indicates the cluster to which the observation was assigned. A negative value indicates that the observation was not assigned to a cluster (see the STRICT option), and the absolute value indicates the cluster to which the observation would have been assigned. If the value is missing, the observation cannot be assigned to any cluster. You can specify the variable name with the CLUSTER= option. The default name is CLUSTER.
- a new variable, DISTANCE, giving the distance from the observation to its cluster seed

If you specify the IMPUTE option, the OUT= data set also contains a new variable, _IMPUTE_, giving the number of imputed values in each observation.

OUTSEED= Data Set

The OUTSEED= data set contains one observation for each cluster. The variables are as follows:

- the BY variables, if any
- a new variable giving the cluster number. You can specify the variable name with the CLUSTER= option. The default name is CLUSTER.
- either the FREQ variable or a new variable called _FREQ_ giving the number of observations in the cluster
- the WEIGHT variable, if any

- a new variable, _RMSSTD_, giving the root mean squared standard deviation for the cluster. See Chapter 29, "The CLUSTER Procedure," for details.
- a new variable, _RADIUS_, giving the maximum distance between any observation in the cluster and the cluster seed
- a new variable, _GAP_, containing the distance between the current cluster mean and the nearest other cluster mean. The value is the centroid distance given in the output.
- a new variable, _NEAR_, specifying the cluster number of the nearest cluster
- the VAR variables giving the cluster means

If you specify the LEAST=p option with a value other than 2, the _RMSSTD_ variable is replaced by the _SCALE_ variable, which contains the pooled scale estimate analogous to the root mean squared standard deviation but based on pth-power deviations instead of squared deviations:

LEAST=1	mean absolute deviation
LEAST=p	root mean p-th-power absolute deviation
LEAST=MAX	maximum absolute deviation

If you specify the OUTITER option, there is one set of observations in the OUTSEED= data set for each pass through the data set (that is, one set for initial seeds, one for each iteration, and one for the final clusters). Also, several additional variables appear:

ITER	is the iteration number. For the initial seeds, the value is 0. For the final cluster means or centers, the _ITER_ variable is one greater than the last iteration reported in the iteration history.
CRIT	is the clustering criterion as described under the LEAST= option.
CHANGE	is the maximum over clusters of the relative change in the cluster seed from the previous iteration. The relative change in a cluster seed is the distance between the old seed and the new seed divided by a scaling factor. If you do not specify the LEAST= option, the scaling factor is the minimum distance between the initial seeds. If you specify the LEAST= option, the scaling factor is an L_1 scale estimate and is recomputed on each iteration.
HOMPAR	is the value of the homotopy parameter. This variable appears only for LEAST=p with $1 < p < 2$.
BINSIZ	is the maximum bin size used for estimating medians. This variable appears only for LEAST=1.

If you specify the OUTITER option, the variables _SCALE_ or _RMSSTD_, _RADIUS_, _NEAR_, and _GAP_ have missing values except for the last pass.

You can use the OUTSEED= data set as a SEED= input data set for a subsequent analysis.

OUTSTAT= Data Set

The variables in the OUTSTAT= data set are as follows:

- BY variables, if any
- a new character variable, _TYPE_, specifying the type of statistic given by other variables (see Table 34.2 and Table 34.3)
- a new numeric variable giving the cluster number. You can specify the variable name with the CLUSTER= option. The default name is CLUSTER.
- a new numeric variable, OVER_ALL, containing statistics that apply over all of the VAR variables
- the VAR variables giving statistics for particular variables

The values of _TYPE_ for all LEAST= options are given in Table 34.2.

Table 34.2 _TYPE_

TYPE	Contents of VAR Variables	Contents of OVER_ALL
INITIAL	Initial seeds	Missing
CRITERION	Missing	Optimization criterion (see the LEAST= option); this value is displayed just before the "Cluster Summary" table.
CENTER	Cluster centers (see the LEAST= option)	Missing
SEED	Cluster seeds: additional information used for imputation	
DISPERSION	Dispersion estimates for each cluster (see the LEAST= option); these values are displayed in a separate row with title depending on the LEAST= option	Dispersion estimates pooled over variables (see the LEAST= option); these values are displayed in the "Cluster Summary" table with label depending on the LEAST= option.
FREQ	Frequency of each cluster omitting observations with missing values for the VAR variable; these values are not displayed	Frequency of each cluster based on all observations with any nonmissing value; these values are displayed in the "Cluster Summary" table.

Table 34.2 *continued*

TYPE	Contents of VAR variables	Contents of OVER_ALL
WEIGHT	Sum of weights for each cluster omitting observations with missing values for the VAR variable; these values are not displayed	Sum of weights for each cluster based on all observations with any nonmissing value; these values are displayed in the "Cluster Summary" table.

Observations with _TYPE_='WEIGHT' are included only if you specify the WEIGHT statement.

The _TYPE_ values included only for least squares clustering are given Table 34.3. Least squares clustering is obtained by omitting the LEAST= option or by specifying LEAST=2.

Table 34.3 _TYPE_

TYPE	Contents of VAR Variables	Contents of OVER_ALL
MEAN	Mean for the total sample; this is not displayed	Missing
STD	Standard deviation for the total sample; labeled "Total STD" in the output	Standard deviation pooled over all the VAR variables; labeled "Total STD" in the output
WITHIN_STD	Pooled within-cluster standard deviation	Within cluster standard deviation pooled over clusters and all the VAR variables
RSQ	R square for predicting the variable from the clusters; labeled "R-Squared" in the output	R square pooled over all the VAR variables; labeled "R-Squared" in the output
RSQ_RATIO	$\frac{R^2}{1-R^2}$; labeled "RSQ/(1-RSQ)" in the output	$\frac{R^2}{1-R^2}$; labeled "RSQ/(1-RSQ)" in the output
PSEUDO_F	Missing	Pseudo F statistic
ESRQ	Missing	Approximate expected value of R square under the null hypothesis of a single uniform cluster
CCC	Missing	Cubic clustering criterion

Computational Resources

Let

n = number of observations
v = number of variables
c = number of clusters
p = number of passes over the data set

Memory

The memory required is approximately $4(19v + 12cv + 10c + 2\max(c + 1, v))$ bytes.

If you request the DISTANCE option, an additional $4c(c + 1)$ bytes of space is needed.

Time

The overall time required by PROC FASTCLUS is roughly proportional to $nvcp$ if c is small with respect to n.

Initial seed selection requires one pass over the data set. If the observations are in random order, the time required is roughly proportional to

$$nvc + vc^2$$

unless you specify REPLACE=NONE. In that case, a complete pass might not be necessary, and the time is roughly proportional to mvc, where $c \leq m \leq n$.

The DRIFT option, each iteration, and the final assignment of cluster seeds each require one pass, with time for each pass roughly proportional to nvc.

For greatest efficiency, you should list the variables in the VAR statement in order of decreasing variance.

Using PROC FASTCLUS

Before using PROC FASTCLUS, decide whether your variables should be standardized in some way, since variables with large variances tend to have more effect on the resulting clusters than those with small variances. If all variables are measured in the same units, standardization might not be necessary. Otherwise, some form of standardization is strongly recommended. The STANDARD procedure can standardize all variables to mean zero and variance one. The FACTOR or

PRINCOMP procedure can compute standardized principal component scores. The ACECLUS procedure can transform the variables according to an estimated within-cluster covariance matrix.

Nonlinear transformations of the variables can change the number of population clusters and should therefore be approached with caution. For most applications, the variables should be transformed so that equal differences are of equal practical importance. An interval scale of measurement is required. Ordinal or ranked data are generally not appropriate.

PROC FASTCLUS produces relatively little output. In most cases you should create an output data set and use another procedure such as PRINT, PLOT, CHART, MEANS, DISCRIM, or CANDISC to study the clusters. It is usually desirable to try several values of the MAXCLUSTERS= option. Macros are useful for running PROC FASTCLUS repeatedly with other procedures.

A simple application of PROC FASTCLUS with two variables to examine the 2- and 3-cluster solutions can proceed as follows:

```
proc standard mean=0 std=1 out=stan;
   var v1 v2;
run;

proc fastclus data=stan out=clust maxclusters=2;
   var v1 v2;
run;

proc plot;
   plot v2*v1=cluster;
run;

proc fastclus data=stan out=clust maxclusters=3;
   var v1 v2;
run;

proc plot;
   plot v2*v1=cluster;
run;
```

If you have more than two variables, you can use the CANDISC procedure to compute canonical variables for plotting the clusters. For example:

```
proc standard mean=0 std=1 out=stan;
   var v1-v10;
run;

proc fastclus data=stan out=clust maxclusters=3;
   var v1-v10;
run;

proc candisc out=can;
   var v1-v10;
   class cluster;
run;
```

```
proc plot;
   plot can2*can1=cluster;
run;
```

If the data set is not too large, it might also be helpful to use the following to list the clusters:

```
proc sort;
   by cluster distance;
run;
proc print;
   by cluster;
run;
```

By examining the values of DISTANCE, you can determine if any observations are unusually far from their cluster seeds.

It is often advisable, especially if the data set is large or contains outliers, to make a preliminary PROC FASTCLUS run with a large number of clusters, perhaps 20 to 100. Use MAXITER=0 and OUTSEED=*SAS-data-set*. You can save time on subsequent runs if you select cluster seeds from this output data set by using the SEED= option.

You should check the preliminary clusters for outliers, which often appear as clusters with only one member. Use a DATA step to delete outliers from the data set created by the OUTSEED= option before using it as a SEED= data set in later runs. If there are severe outliers, you should specify the STRICT option in the subsequent PROC FASTCLUS runs to prevent the outliers from distorting the clusters.

You can use the OUTSEED= data set with the PLOT procedure to plot _GAP_ by _FREQ_. An overlay of _RADIUS_ by _FREQ_ provides a baseline against which to compare the values of _GAP_. Outliers appear in the upper-left area of the plot, with large _GAP_ values and small _FREQ_ values. Good clusters appear in the upper-right area, with large values of both _GAP_ and _FREQ_. Good potential cluster seeds appear in the lower right, as well as in the upper-right, since large _FREQ_ values indicate high-density regions. Small _FREQ_ values in the left part of the plot indicate poor cluster seeds because the points are in low-density regions. It often helps to remove all clusters with small frequencies even though the clusters might not be remote enough to be considered outliers. Removing points in low-density regions improves cluster separation and provides visually sharper cluster outlines in scatter plots.

Displayed Output

Unless the SHORT or SUMMARY option is specified, PROC FASTCLUS displays the following:

- Initial Seeds, cluster seeds selected after one pass through the data
- Change in Cluster Seeds for each iteration, if you specify MAXITER=$n > 1$

If you specify the LEAST=*p* option, with $(1 < p < 2)$, and you omit the IRLS option, an additional column is displayed in the Iteration History table. This column contains a character to identify the method used in each iteration. PROC FASTCLUS chooses the most efficient method to cluster the data at each iterative step, given the condition of the data. Thus, the method chosen is data dependent. The possible values are described as follows:

Value	Method
N	Newton's Method
I or L	iteratively weighted least squares (IRLS)
1	IRLS step, halved once
2	IRLS step, halved twice
3	IRLS step, halved three times

PROC FASTCLUS displays a Cluster Summary, giving the following for each cluster:

- Cluster number
- Frequency, the number of observations in the cluster
- Weight, the sum of the weights of the observations in the cluster, if you specify the WEIGHT statement
- RMS Std Deviation, the root mean squared across variables of the cluster standard deviations, which is equal to the root mean square distance between observations in the cluster
- Maximum Distance from Seed to Observation, the maximum distance from the cluster seed to any observation in the cluster
- Nearest Cluster, the number of the cluster with mean closest to the mean of the current cluster
- Centroid Distance, the distance between the centroids (means) of the current cluster and the nearest other cluster

A table of statistics for each variable is displayed unless you specify the SUMMARY option. The table contains the following:

- Total STD, the total standard deviation
- Within STD, the pooled within-cluster standard deviation
- R-Squared, the R square for predicting the variable from the cluster
- RSQ/(1 - RSQ), the ratio of between-cluster variance to within-cluster variance $(R^2/(1-R^2))$
- OVER-ALL, all of the previous quantities pooled across variables

PROC FASTCLUS also displays the following:

- Pseudo F Statistic,

  $$\frac{\frac{R^2}{c-1}}{\frac{1-R^2}{n-c}}$$

  where R square is the observed overall R square, c is the number of clusters, and n is the number of observations. The pseudo F statistic was suggested by Calinski and Harabasz (1974). See Milligan and Cooper (1985) and Cooper and Milligan (1988) regarding the use of the pseudo F statistic in estimating the number of clusters. See Example 29.2 in Chapter 29, "The CLUSTER Procedure," for a comparison of pseudo F statistics.

- Observed Overall R-Squared, if you specify the SUMMARY option

- Approximate Expected Overall R-Squared, the approximate expected value of the overall R square under the uniform null hypothesis assuming that the variables are uncorrelated. The value is missing if the number of clusters is greater than one-fifth the number of observations.

- Cubic Clustering Criterion, computed under the assumption that the variables are uncorrelated. The value is missing if the number of clusters is greater than one-fifth the number of observations.

  If you are interested in the approximate expected R square or the cubic clustering criterion but your variables are correlated, you should cluster principal component scores from the PRINCOMP procedure. Both of these statistics are described by Sarle (1983). The performance of the cubic clustering criterion in estimating the number of clusters is examined by Milligan and Cooper (1985) and Cooper and Milligan (1988).

- Distances Between Cluster Means, if you specify the DISTANCE option

Unless you specify the SHORT or SUMMARY option, PROC FASTCLUS displays the following:

- Cluster Means for each variable

- Cluster Standard Deviations for each variable

ODS Table Names

PROC FASTCLUS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 34.4. For more information on ODS, see Chapter 20, "Using the Output Delivery System."

Table 34.4 ODS Tables Produced by PROC FASTCLUS

ODS Table Name	Description	Statement	Option
ApproxExpOverAllRSq	Approximate expected overall R-squared, single number	PROC	default
CCC	CCC, Cubic Clustering Criterion, single number	PROC	default
ClusterList	Cluster listing, obs, id, and distances	PROC	LIST
ClusterSum	Cluster summary, cluster number, distances	PROC	PRINTALL
ClusterCenters	Cluster centers	PROC	default
ClusterDispersion	Cluster dispersion	PROC	default
ConvergenceStatus	Convergence status	PROC	PRINTALL
Criterion	Criterion based on final seeds, single number	PROC	default
DistBetweenClust	Distance between clusters	PROC	default
InitialSeeds	Initial seeds	PROC	default
IterHistory	Iteration history, various statistics for each iteration	PROC	PRINTALL
MinDist	Minimum distance between initial seeds, single number	PROC	PRINTALL
NumberOfBins	Number of bins	PROC	default
ObsOverAllRSquare	Observed overall R-squared, single number	PROC	SUMMARY
PrelScaleEst	Preliminary L(1) scale estimate, single number	PROC	PRINTALL
PseudoFStat	Pseudo F statistic, single number	PROC	default
SimpleStatistics	Simple statistics for input variables	PROC	default
VariableStat	Statistics for variables within clusters	PROC	default

Examples: FASTCLUS Procedure

Example 34.1: Fisher's Iris Data

The iris data published by Fisher (1936) have been widely used for examples in discriminant analysis and cluster analysis. The sepal length, sepal width, petal length, and petal width are measured in millimeters on 50 iris specimens from each of three species, *Iris setosa, I. versicolor,* and *I. virginica*. Mezzich and Solomon (1980) discuss a variety of cluster analysis of the iris data.

In this example, the FASTCLUS procedure is used to find two and then three clusters. In the following code, an output data set is created, and PROC FREQ is invoked to compare the clusters with the species classification. See Output 34.1.1 and Output 34.1.2 for these results.

For three clusters, you can use the CANDISC procedure to compute canonical variables for plotting the clusters. See Output 34.1.3 and Output 34.1.4 for the results.

```
   proc format;
      value specname
         1='Setosa    '
         2='Versicolor'
         3='Virginica ';
   run;

   data iris;
      title 'Fisher (1936) Iris Data';
      input SepalLength SepalWidth PetalLength PetalWidth Species @@;
      format Species specname.;
      label SepalLength='Sepal Length in mm.'
            SepalWidth ='Sepal Width in mm.'
            PetalLength='Petal Length in mm.'
            PetalWidth ='Petal Width in mm.';
      symbol = put(species, specname10.);
      datalines;
   50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3
   63 28 51 15 3 46 34 14 03 1 69 31 51 23 3 62 22 45 15 2
   59 32 48 18 2 46 36 10 02 1 61 30 46 14 2 60 27 51 16 2
   65 30 52 20 3 56 25 39 11 2 65 30 55 18 3 58 27 51 19 3
   68 32 59 23 3 51 33 17 05 1 57 28 45 13 2 62 34 54 23 3

... more lines ...

   55 23 40 13 2 66 30 44 14 2 68 28 48 14 2 54 34 17 02 1
   51 37 15 04 1 52 35 15 02 1 58 28 51 24 3 67 30 50 17 2
   63 33 60 25 3 53 37 15 02 1
   ;
proc fastclus data=iris maxc=2 maxiter=10 out=clus;
   var SepalLength SepalWidth PetalLength PetalWidth;
run;
```

```
proc freq;
   tables cluster*species;
run;

proc fastclus data=iris maxc=3 maxiter=10 out=clus;
   var SepalLength SepalWidth PetalLength PetalWidth;
run;

proc freq;
   tables cluster*Species;
run;

proc candisc anova out=can;
   class cluster;
   var SepalLength SepalWidth PetalLength PetalWidth;
   title2 'Canonical Discriminant Analysis of Iris Clusters';
run;

proc sgplot data=Can;
   scatter y=Can2 x=Can1 /group=Cluster ;
   title2 'Plot of Canonical Variables Identified by Cluster';
run;
```

Output 34.1.1 Fisher's Iris Data: PROC FASTCLUS with MAXC=2 andPROC FREQ

```
                          Fisher (1936) Iris Data

                           The FASTCLUS Procedure
        Replace=FULL  Radius=0  Maxclusters=2 Maxiter=10  Converge=0.02

                                Initial Seeds

Cluster       SepalLength        SepalWidth       PetalLength        PetalWidth
-------------------------------------------------------------------------------
   1          43.00000000       30.00000000       11.00000000        1.00000000
   2          77.00000000       26.00000000       69.00000000       23.00000000

               Minimum Distance Between Initial Seeds = 70.85196

                                Iteration History

                                              Relative Change
                                             in Cluster Seeds
                 Iteration    Criterion           1           2
                 ---------------------------------------------
                         1      11.0638      0.1904      0.3163
                         2       5.3780      0.0596      0.0264
                         3       5.0718      0.0174     0.00766

        Convergence criterion is satisfied.

                   Criterion Based on Final Seeds =   5.0417
```

Output 34.1.1 *continued*

Cluster Summary

Cluster	Frequency	RMS Std Deviation	Maximum Distance from Seed to Observation	Radius Exceeded	Nearest Cluster
1	53	3.7050	21.1621		2
2	97	5.6779	24.6430		1

Cluster Summary

Cluster	Distance Between Cluster Centroids
1	39.2879
2	39.2879

Statistics for Variables

Variable	Total STD	Within STD	R-Square	RSQ/(1-RSQ)
SepalLength	8.28066	5.49313	0.562896	1.287784
SepalWidth	4.35866	3.70393	0.282710	0.394137
PetalLength	17.65298	6.80331	0.852470	5.778291
PetalWidth	7.62238	3.57200	0.781868	3.584390
OVER-ALL	10.69224	5.07291	0.776410	3.472463

Pseudo F Statistic = 513.92

Approximate Expected Over-All R-Squared = 0.51539

Cubic Clustering Criterion = 14.806

WARNING: The two values above are invalid for correlated variables.

Cluster Means

Cluster	SepalLength	SepalWidth	PetalLength	PetalWidth
1	50.05660377	33.69811321	15.60377358	2.90566038
2	63.01030928	28.86597938	49.58762887	16.95876289

Cluster Standard Deviations

Cluster	SepalLength	SepalWidth	PetalLength	PetalWidth
1	3.427350930	4.396611045	4.404279486	2.105525249
2	6.336887455	3.267991438	7.800577673	4.155612484

Output 34.1.1 *continued*

```
                          Fisher (1936) Iris Data

                             The FREQ Procedure

                      Table of CLUSTER by Species

            CLUSTER(Cluster)     Species

            Frequency|
            Percent  |
            Row Pct  |
            Col Pct  |Setosa  |Versicol|Virginic|  Total
                     |         |or      |a       |
            ---------+---------+--------+--------+
                   1 |      50 |      3 |      0 |     53
                     |   33.33 |   2.00 |   0.00 |  35.33
                     |   94.34 |   5.66 |   0.00 |
                     |  100.00 |   6.00 |   0.00 |
            ---------+---------+--------+--------+
                   2 |       0 |     47 |     50 |     97
                     |    0.00 |  31.33 |  33.33 |  64.67
                     |    0.00 |  48.45 |  51.55 |
                     |    0.00 |  94.00 | 100.00 |
            ---------+---------+--------+--------+
            Total           50       50       50      150
                         33.33    33.33    33.33   100.00
```

Output 34.1.2 Fisher's Iris Data: PROC FASTCLUS with MAXC=3 and PROC FREQ

```
                          Fisher (1936) Iris Data

                            The FASTCLUS Procedure
        Replace=FULL  Radius=0  Maxclusters=3 Maxiter=10  Converge=0.02

                                 Initial Seeds

Cluster       SepalLength        SepalWidth       PetalLength        PetalWidth
-------------------------------------------------------------------------------
   1          58.00000000       40.00000000       12.00000000        2.00000000
   2          77.00000000       38.00000000       67.00000000       22.00000000
   3          49.00000000       25.00000000       45.00000000       17.00000000

              Minimum Distance Between Initial Seeds = 38.23611

                               Iteration History

                                    Relative Change in Cluster Seeds
          Iteration    Criterion           1           2           3
          ------------------------------------------------------------
                  1       6.7591      0.2652      0.3205      0.2985
                  2       3.7097           0      0.0459      0.0317
                  3       3.6427           0      0.0182      0.0124
```

Output 34.1.2 *continued*

Convergence criterion is satisfied.

Criterion Based on Final Seeds = 3.6289

Cluster Summary

Cluster	Frequency	RMS Std Deviation	Maximum Distance from Seed to Observation	Radius Exceeded	Nearest Cluster
1	50	2.7803	12.4803		3
2	38	4.0168	14.9736		3
3	62	4.0398	16.9272		2

Cluster Summary

Cluster	Distance Between Cluster Centroids
1	33.5693
2	17.9718
3	17.9718

Statistics for Variables

Variable	Total STD	Within STD	R-Square	RSQ/(1-RSQ)
SepalLength	8.28066	4.39488	0.722096	2.598359
SepalWidth	4.35866	3.24816	0.452102	0.825156
PetalLength	17.65298	4.21431	0.943773	16.784895
PetalWidth	7.62238	2.45244	0.897872	8.791618
OVER-ALL	10.69224	3.66198	0.884275	7.641194

Pseudo F Statistic = 561.63

Approximate Expected Over-All R-Squared = 0.62728

Cubic Clustering Criterion = 25.021

WARNING: The two values above are invalid for correlated variables.

Cluster Means

Cluster	SepalLength	SepalWidth	PetalLength	PetalWidth
1	50.06000000	34.28000000	14.62000000	2.46000000
2	68.50000000	30.73684211	57.42105263	20.71052632
3	59.01612903	27.48387097	43.93548387	14.33870968

Output 34.1.2 *continued*

Cluster Standard Deviations

Cluster	SepalLength	SepalWidth	PetalLength	PetalWidth
1	3.524896872	3.790643691	1.736639965	1.053855894
2	4.941550255	2.900924461	4.885895746	2.798724562
3	4.664100551	2.962840548	5.088949673	2.974997167

Fisher (1936) Iris Data

The FREQ Procedure

Table of CLUSTER by Species

CLUSTER(Cluster) Species

Frequency Percent Row Pct Col Pct	Setosa	Versicolor	Virginica	Total
1	50 33.33 100.00 100.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	50 33.33
2	0 0.00 0.00 0.00	2 1.33 5.26 4.00	36 24.00 94.74 72.00	38 25.33
3	0 0.00 0.00 0.00	48 32.00 77.42 96.00	14 9.33 22.58 28.00	62 41.33
Total	50 33.33	50 33.33	50 33.33	150 100.00

Output 34.1.3 Fisher's Iris Data using PROC CANDISC

Fisher (1936) Iris Data
Canonical Discriminant Analysis of Iris Clusters

The CANDISC Procedure

Total Sample Size	150	DF Total	149
Variables	4	DF Within Classes	147
Classes	3	DF Between Classes	2

Number of Observations Read	150
Number of Observations Used	150

Output 34.1.3 *continued*

Class Level Information

CLUSTER	Variable Name	Frequency	Weight	Proportion
1	_1	50	50.0000	0.333333
2	_2	38	38.0000	0.253333
3	_3	62	62.0000	0.413333

Fisher (1936) Iris Data
Canonical Discriminant Analysis of Iris Clusters

The CANDISC Procedure

Univariate Test Statistics

F Statistics, Num DF=2, Den DF=147

Variable	Label	Total Standard Deviation	Pooled Standard Deviation	Between Standard Deviation	R-Square	R-Square / (1-RSq)	F Value	Pr > F
Sepal Length	Sepal Length in mm.	8.2807	4.3949	8.5893	0.7221	2.5984	190.98	<.0001
Sepal Width	Sepal Width in mm.	4.3587	3.2482	3.5774	0.4521	0.8252	60.65	<.0001
Petal Length	Petal Length in mm.	17.6530	4.2143	20.9336	0.9438	16.7849	1233.69	<.0001
Petal Width	Petal Width in mm.	7.6224	2.4524	8.8164	0.8979	8.7916	646.18	<.0001

Average R-Square

Unweighted	0.7539604
Weighted by Variance	0.8842753

Multivariate Statistics and F Approximations

S=2 M=0.5 N=71

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.03222337	164.55	8	288	<.0001
Pillai's Trace	1.25669612	61.29	8	290	<.0001
Hotelling-Lawley Trace	21.06722883	377.66	8	203.4	<.0001
Roy's Greatest Root	20.63266809	747.93	4	145	<.0001

NOTE: F Statistic for Roy's Greatest Root is an upper bound.
NOTE: F Statistic for Wilks' Lambda is exact.

Output 34.1.3 *continued*

Fisher (1936) Iris Data
Canonical Discriminant Analysis of Iris Clusters

The CANDISC Procedure

	Canonical Correlation	Adjusted Canonical Correlation	Approximate Standard Error	Squared Canonical Correlation
1	0.976613	0.976123	0.003787	0.953774
2	0.550384	0.543354	0.057107	0.302923

Eigenvalues of Inv(E)*H
= CanRsq/(1-CanRsq)

	Eigenvalue	Difference	Proportion	Cumulative
1	20.6327	20.1981	0.9794	0.9794
2	0.4346		0.0206	1.0000

Test of H0: The canonical correlations in the current row and all that follow are zero

	Likelihood Ratio	Approximate F Value	Num DF	Den DF	Pr > F
1	0.03222337	164.55	8	288	<.0001
2	0.69707749	21.00	3	145	<.0001

Fisher (1936) Iris Data
Canonical Discriminant Analysis of Iris Clusters

The CANDISC Procedure

Total Canonical Structure

Variable	Label	Can1	Can2
SepalLength	Sepal Length in mm.	0.831965	0.452137
SepalWidth	Sepal Width in mm.	-0.515082	0.810630
PetalLength	Petal Length in mm.	0.993520	0.087514
PetalWidth	Petal Width in mm.	0.966325	0.154745

Between Canonical Structure

Variable	Label	Can1	Can2
SepalLength	Sepal Length in mm.	0.956160	0.292846
SepalWidth	Sepal Width in mm.	-0.748136	0.663545
PetalLength	Petal Length in mm.	0.998770	0.049580
PetalWidth	Petal Width in mm.	0.995952	0.089883

Output 34.1.3 *continued*

Pooled Within Canonical Structure

Variable	Label	Can1	Can2
SepalLength	Sepal Length in mm.	0.339314	0.716082
SepalWidth	Sepal Width in mm.	-0.149614	0.914351
PetalLength	Petal Length in mm.	0.900839	0.308136
PetalWidth	Petal Width in mm.	0.650123	0.404282

Fisher (1936) Iris Data
Canonical Discriminant Analysis of Iris Clusters

The CANDISC Procedure

Total-Sample Standardized Canonical Coefficients

Variable	Label	Can1	Can2
SepalLength	Sepal Length in mm.	0.047747341	1.021487262
SepalWidth	Sepal Width in mm.	-0.577569244	0.864455153
PetalLength	Petal Length in mm.	3.341309573	-1.283043758
PetalWidth	Petal Width in mm.	0.996451144	0.900476563

Pooled Within-Class Standardized Canonical Coefficients

Variable	Label	Can1	Can2
SepalLength	Sepal Length in mm.	0.0253414487	0.5421446856
SepalWidth	Sepal Width in mm.	-.4304161258	0.6442092294
PetalLength	Petal Length in mm.	0.7976741592	-.3063023132
PetalWidth	Petal Width in mm.	0.3205998034	0.2897207865

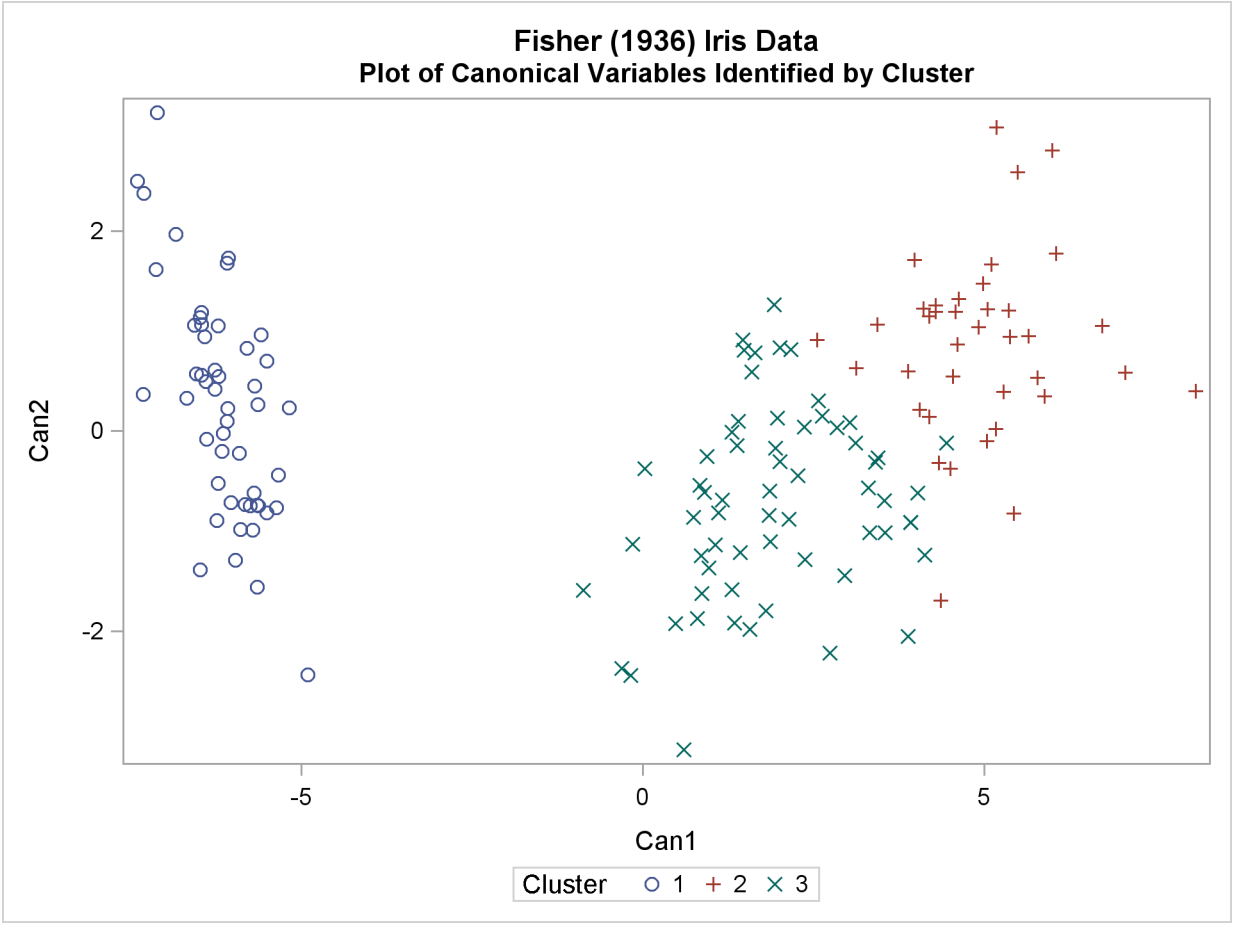
Raw Canonical Coefficients

Variable	Label	Can1	Can2
SepalLength	Sepal Length in mm.	0.0057661265	0.1233581748
SepalWidth	Sepal Width in mm.	-.1325106494	0.1983303556
PetalLength	Petal Length in mm.	0.1892773419	-.0726814163
PetalWidth	Petal Width in mm.	0.1307270927	0.1181359305

Class Means on Canonical Variables

CLUSTER	Can1	Can2
1	-6.131527227	0.244761516
2	4.931414018	0.861972277
3	1.922300462	-0.725693908

Output 34.1.4 Plot of Fisher's Iris Data using PROC CANDISC



Example 34.2: Outliers

This example involves data artificially generated to contain two clusters and several severe outliers. A preliminary analysis specifies 20 clusters and outputs an OUTSEED= data set to be used for a diagnostic plot. The exact number of initial clusters is not important; similar results could be obtained with 10 or 50 initial clusters. Examination of the plot suggests that clusters with more than five (again, the exact number is not important) observations can yield good seeds for the main analysis. A DATA step deletes clusters with five or fewer observations, and the remaining cluster means provide seeds for the next PROC FASTCLUS analysis.

Two clusters are requested; the LEAST= option specifies the mean absolute deviation criterion (LEAST=1). Values of the LEAST= option less than 2 reduce the effect of outliers on cluster centers.

The next analysis also requests two clusters; the STRICT= option is specified to prevent outliers from distorting the results. The STRICT= value is chosen to be close to the _GAP_ and _RADIUS_ values of the larger clusters in the diagnostic plot; the exact value is not critical.

A final PROC FASTCLUS run assigns the outliers to clusters.

The following SAS statements implement these steps, and the results are displayed in Output 34.2.3 through Output 34.2.8. First, an artificial data set is created with two clusters and some outliers. Then PROC FASTCLUS is run with many clusters to produce an OUTSEED= data set. A diagnostic plot using the variables _GAP_ and _RADIUS_ is then produced using the SGSCATTER procedure. The results from these steps are shown in Output 34.2.1 and Output 34.2.2.

```
data x;
title 'Using PROC FASTCLUS to Analyze Data with Outliers';
   drop n;
   do n=1 to 100;
      x=rannor(12345)+2;
      y=rannor(12345);
      output;
   end;
   do n=1 to 100;
      x=rannor(12345)-2;
      y=rannor(12345);
      output;
   end;
   do n=1 to 10;
      x=10*rannor(12345);
      y=10*rannor(12345);
      output;
   end;
run;

title2 'Preliminary PROC FASTCLUS Analysis with 20 Clusters';
proc fastclus data=x outseed=mean1 maxc=20 maxiter=0 summary;
   var x y;
run;

proc sgscatter data=mean1 ;
   compare y=(_gap_ _radius_) x=_freq_ ;
run;
```

Output 34.2.1 Preliminary Analysis of Data with Outliers Using PROC FASTCLUS

```
          Using PROC FASTCLUS to Analyze Data with Outliers
        Preliminary PROC FASTCLUS Analysis with 20 Clusters

                      The FASTCLUS Procedure
        Replace=FULL  Radius=0  Maxclusters=20 Maxiter=0

           Criterion Based on Final Seeds =   0.6873
```

Output 34.2.1 *continued*

Cluster Summary

Cluster	Frequency	RMS Std Deviation	Maximum Distance from Seed to Observation	Radius Exceeded	Nearest Cluster
1	8	0.4753	1.1924		19
2	1	.	0		6
3	44	0.6252	1.6774		5
4	1	.	0		20
5	38	0.5603	1.4528		3
6	2	0.0542	0.1085		2
7	1	.	0		14
8	2	0.6480	1.2961		1
9	1	.	0		7
10	1	.	0		18
11	1	.	0		16
12	20	0.5911	1.6291		16
13	5	0.6682	1.4244		3
14	1	.	0		7
15	5	0.4074	1.2678		3
16	22	0.4168	1.5139		19
17	8	0.4031	1.4794		5
18	1	.	0		10
19	45	0.6475	1.6285		16
20	3	0.5719	1.3642		15

Cluster Summary

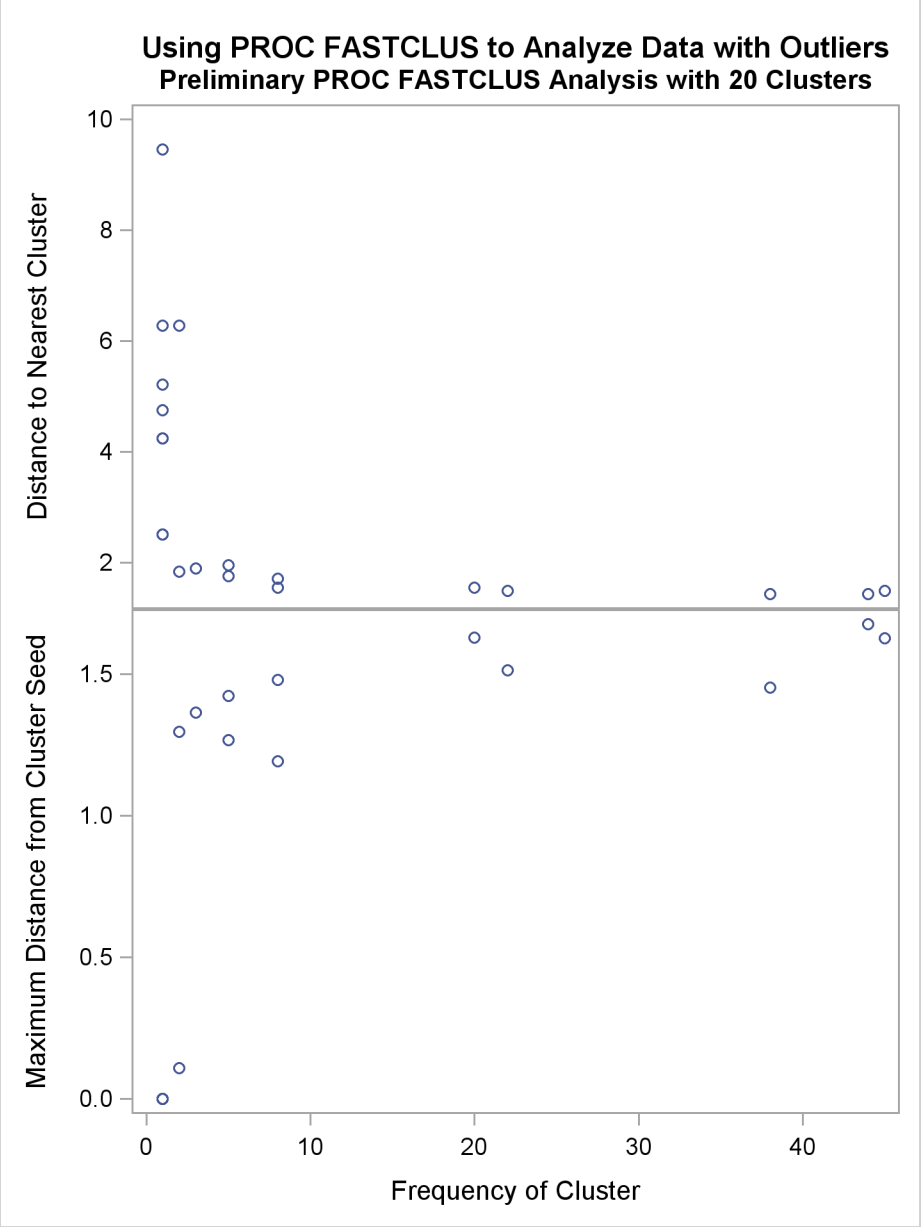
Cluster	Distance Between Cluster Centroids
1	1.7205
2	6.2847
3	1.4386
4	5.2130
5	1.4386
6	6.2847
7	2.5094
8	1.8450
9	9.4534
10	4.2514
11	4.7582
12	1.5601
13	1.9553
14	2.5094
15	1.7609
16	1.4936
17	1.5564
18	4.2514
19	1.4936
20	1.8999

Pseudo F Statistic = 207.58

Output 34.2.1 *continued*

```
                Observed Over-All R-Squared =  0.95404

        Approximate Expected Over-All R-Squared =   0.96103

                Cubic Clustering Criterion =   -2.503

WARNING: The two values above are invalid for correlated variables.
```

Output 34.2.2 Preliminary Analysis of Data with Outliers: Plot Using and PROC SGSCATGTER



In the following SAS statements, a DATA step is used to remove low frequency clusters, then the FASTCLUS procedure is run again, selecting seeds from the high frequency clusters in the previous analysis using LEAST=1 clustering criterion. The results are shown in Output 34.2.3 and Output 34.2.4.

```
data seed;
   set mean1;
   if _freq_>5;
run;

title2 'PROC FASTCLUS Analysis Using LEAST= Clustering Criterion';
title3 'Values < 2 Reduce Effect of Outliers on Cluster Centers';
proc fastclus data=x seed=seed maxc=2 least=1 out=out;
   var x y;
run;
proc sgplot data=out;
   scatter y=y x=x /group=cluster;
run;
```

Output 34.2.3 Analysis of Data with Outliers Using the LEAST= Option

```
              Using PROC FASTCLUS to Analyze Data with Outliers
           PROC FASTCLUS Analysis Using LEAST= Clustering Criterion
           Values < 2 Reduce Effect of Outliers on Cluster Centers

                           The FASTCLUS Procedure
   Replace=FULL  Radius=0  Maxclusters=2 Maxiter=20  Converge=0.0001  Least=1

                               Initial Seeds

                 Cluster                 x                 y
                 -------------------------------------------
                    1          2.794174248      -0.065970836
                    2         -2.027300384      -2.051208579

              Minimum Distance Between Initial Seeds = 6.806712

              Preliminary L(1) Scale Estimate =        2.796579

                          Number of Bins =      100

                               Iteration History

                                                  Relative Change
                                    Maximum      in Cluster Seeds
          Iteration    Criterion    Bin Size           1           2
          ---------------------------------------------------------
                  1       1.3983      0.2263      0.4091      0.6696
                  2       1.0776      0.0226     0.00511      0.0452
                  3       1.0771     0.00226     0.00229     0.00234
                  4       1.0771    0.000396    0.000253    0.000144
                  5       1.0771    0.000396           0           0
```

Output 34.2.3 *continued*

Convergence criterion is satisfied.

Criterion Based on Final Seeds = 1.0771

Cluster Summary

Cluster	Frequency	Mean Absolute Deviation	Maximum Distance from Seed to Observation	Radius Exceeded	Nearest Cluster
1	102	1.1278	24.1622		2
2	108	1.0494	14.8292		1

Cluster Summary

Cluster	Distance Between Cluster Medians
1	4.2585
2	4.2585

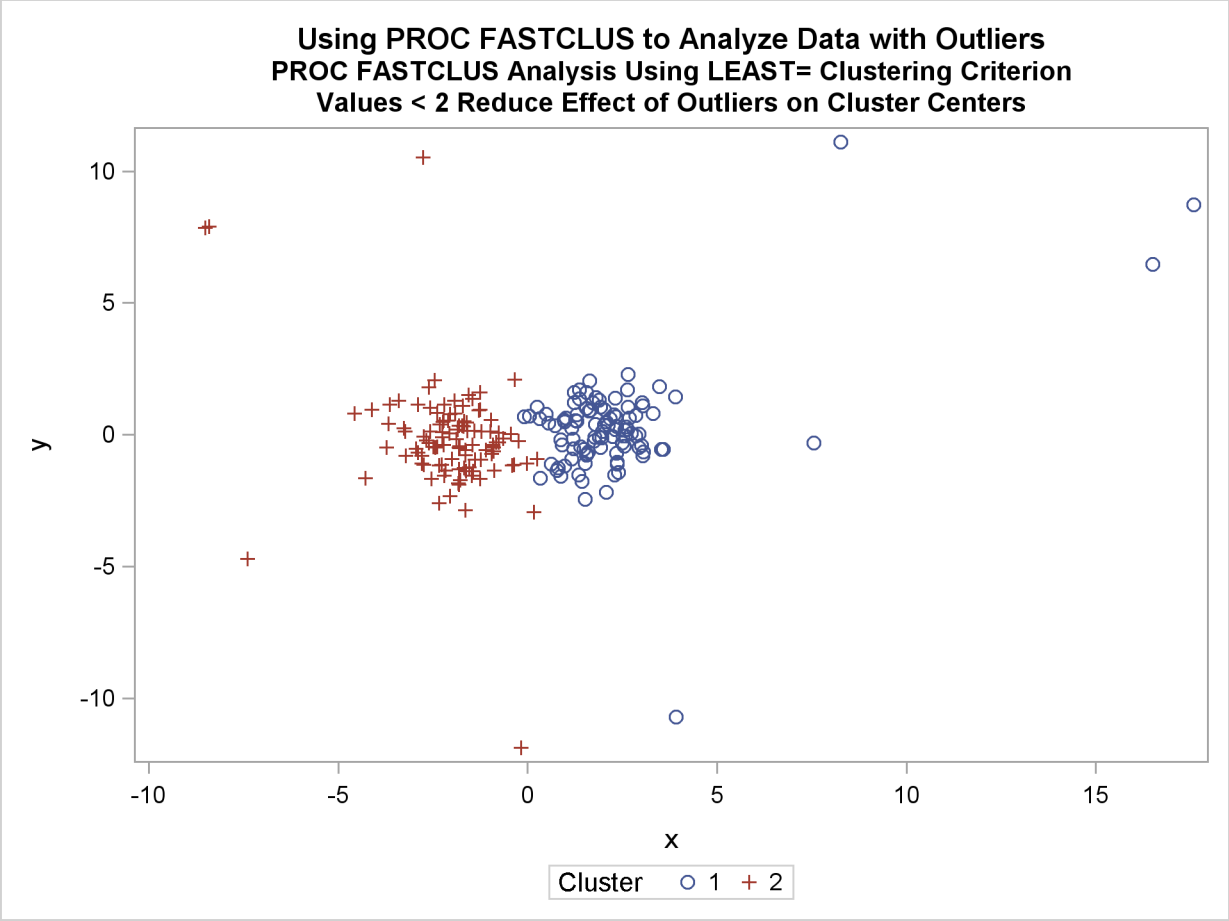
Cluster Medians

Cluster	x	y
1	1.923023887	0.222482918
2	-1.826721743	-0.286253041

Mean Absolute Deviations from Final Seeds

Cluster	x	y
1	1.113465261	1.142120480
2	0.890331835	1.208370913

Output 34.2.4 Analysis Plot of Data with Outliers



The FASTCLUS procedure is run again, selecting seeds from high frequency clusters in the previous analysis. STRICT= prevents outliers from distorting the results. The results are shown in Output 34.2.5 and Output 34.2.6.

```
title2 'PROC FASTCLUS Analysis Using STRICT= to Omit Outliers';
proc fastclus data=x seed=seed
     maxc=2 strict=3.0 out=out outseed=mean2;
   var x y;
run;

proc sgplot data=out;
   scatter y=y  x=x /group=cluster ;
run;
```

Output 34.2.5 Cluster Analysis with Outliers Omitted: PROC FASTCLUS SGPLOT

Using PROC FASTCLUS to Analyze Data with Outliers
PROC FASTCLUS Analysis Using STRICT= to Omit Outliers

The FASTCLUS Procedure
Replace=FULL Radius=0 Strict=3 Maxclusters=2 Maxiter=1

Initial Seeds

Cluster	x	y
1	2.794174248	-0.065970836
2	-2.027300384	-2.051208579

Criterion Based on Final Seeds = 0.9515

Cluster Summary

Cluster	Frequency	RMS Std Deviation	Maximum Distance from Seed to Observation	Radius Exceeded	Nearest Cluster
1	99	0.9501	2.9589		2
2	99	0.9290	2.8011		1

Cluster Summary

Cluster	Distance Between Cluster Centroids
1	3.7666
2	3.7666

12 Observation(s) were not assigned to a cluster
because the minimum distance to a cluster seed exceeded the STRICT= value.

Statistics for Variables

Variable	Total STD	Within STD	R-Square	RSQ/(1-RSQ)
x	2.06854	0.87098	0.823609	4.669219
y	1.02113	1.00352	0.039093	0.040683
OVER-ALL	1.63119	0.93959	0.669891	2.029303

Pseudo F Statistic = 397.74

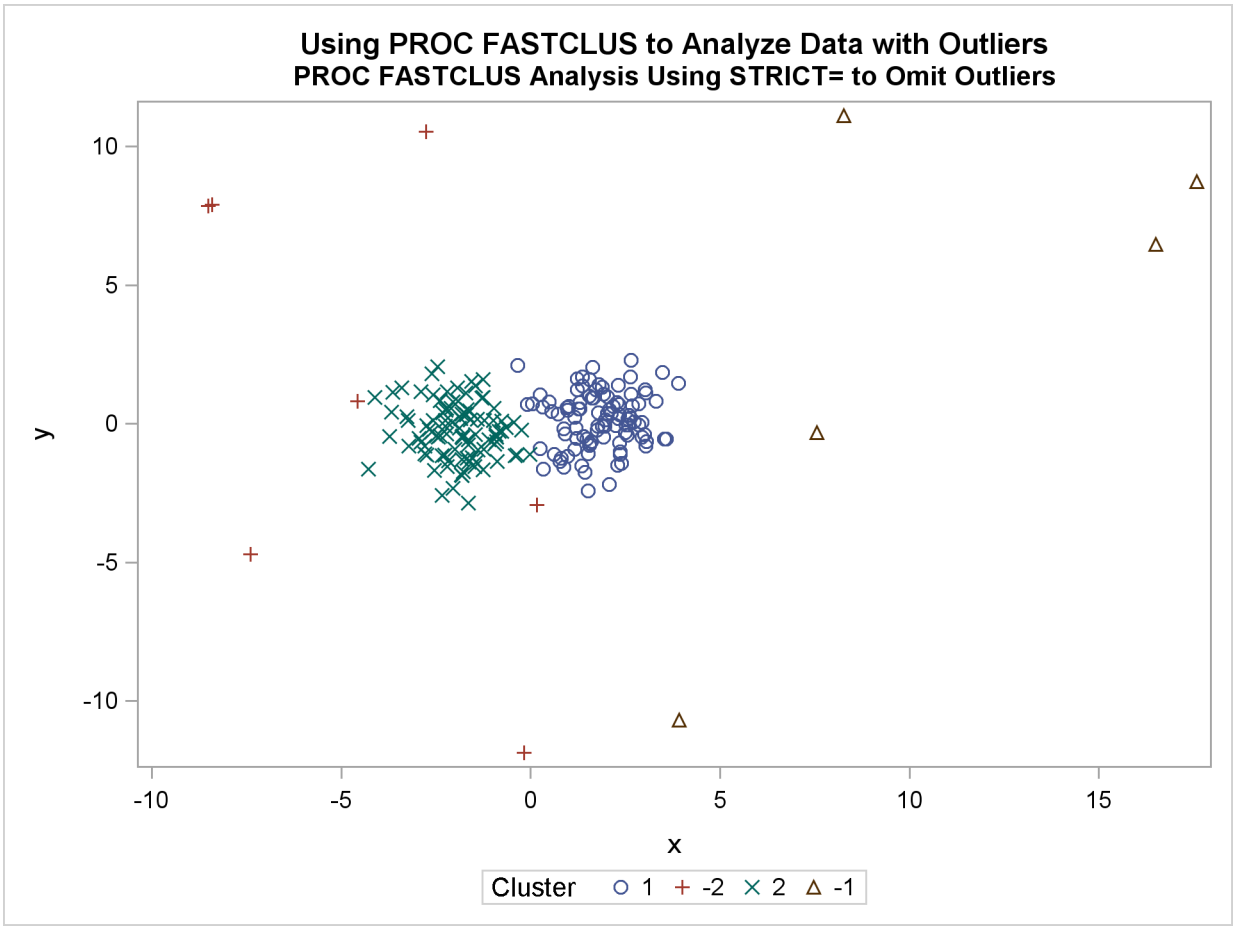
Approximate Expected Over-All R-Squared = 0.60615

Cubic Clustering Criterion = 3.197

Output 34.2.5 *continued*

```
WARNING: The two values above are invalid for correlated variables.


                         Cluster Means

          Cluster                 x                 y
          ------------------------------------------
             1          1.825111432       0.141211701
             2         -1.919910712      -0.261558725

                   Cluster Standard Deviations

          Cluster                 x                 y
          ------------------------------------------
             1          0.889549271       1.006965219
             2          0.852000588       1.000062579
```

Output 34.2.6 Cluster Analysis with Outliers Omitted: Plot Using PROC SGPLOT



Finally, the FASTCLUS procedure is run one more time with zero iterations to assign outliers and tails to clusters. The results are show in Output 34.2.7 and Output 34.2.8.

```
title2 'Final PROC FASTCLUS Analysis Assigning Outliers to '
       'Clusters';
proc fastclus data=x seed=mean2 maxc=2 maxiter=0 out=out;
   var x y;
run;

proc sgplot data=out;
   scatter y=y x=x /group=cluster ;
run;
```

Output 34.2.7 Cluster Analysis with Outliers Omitted: PROC FASTCLUS

```
            Using PROC FASTCLUS to Analyze Data with Outliers
          Final PROC FASTCLUS Analysis Assigning Outliers to Clusters

                          The FASTCLUS Procedure
                Replace=FULL  Radius=0  Maxclusters=2 Maxiter=0

                               Initial Seeds

                  Cluster                x                y
                  ------------------------------------------
                     1          1.825111432       0.141211701
                     2         -1.919910712      -0.261558725

                   Criterion Based on Final Seeds =   2.0594

                               Cluster Summary

                                        Maximum Distance
                             RMS Std           from Seed     Radius     Nearest
 Cluster     Frequency     Deviation      to Observation    Exceeded    Cluster
 ------------------------------------------------------------------------------
    1              103       2.2569             17.9426                     2
    2              107       1.8371             11.7362                     1

                               Cluster Summary

                                    Distance Between
                          Cluster     Cluster Centroids
                          ----------------------------
                             1                  4.3753
                             2                  4.3753

                            Statistics for Variables

       Variable     Total STD    Within STD      R-Square     RSQ/(1-RSQ)
       ------------------------------------------------------------------
       x              2.92721       1.95529      0.555950        1.252000
       y              2.15248       2.14754      0.009347        0.009435
       OVER-ALL       2.56922       2.05367      0.364119        0.572621

                          Pseudo F Statistic =   119.11
```

Output 34.2.7 *continued*

```
Approximate Expected Over-All R-Squared =    0.49090

Cubic Clustering Criterion =    -5.338

WARNING: The two values above are invalid for correlated variables.
```

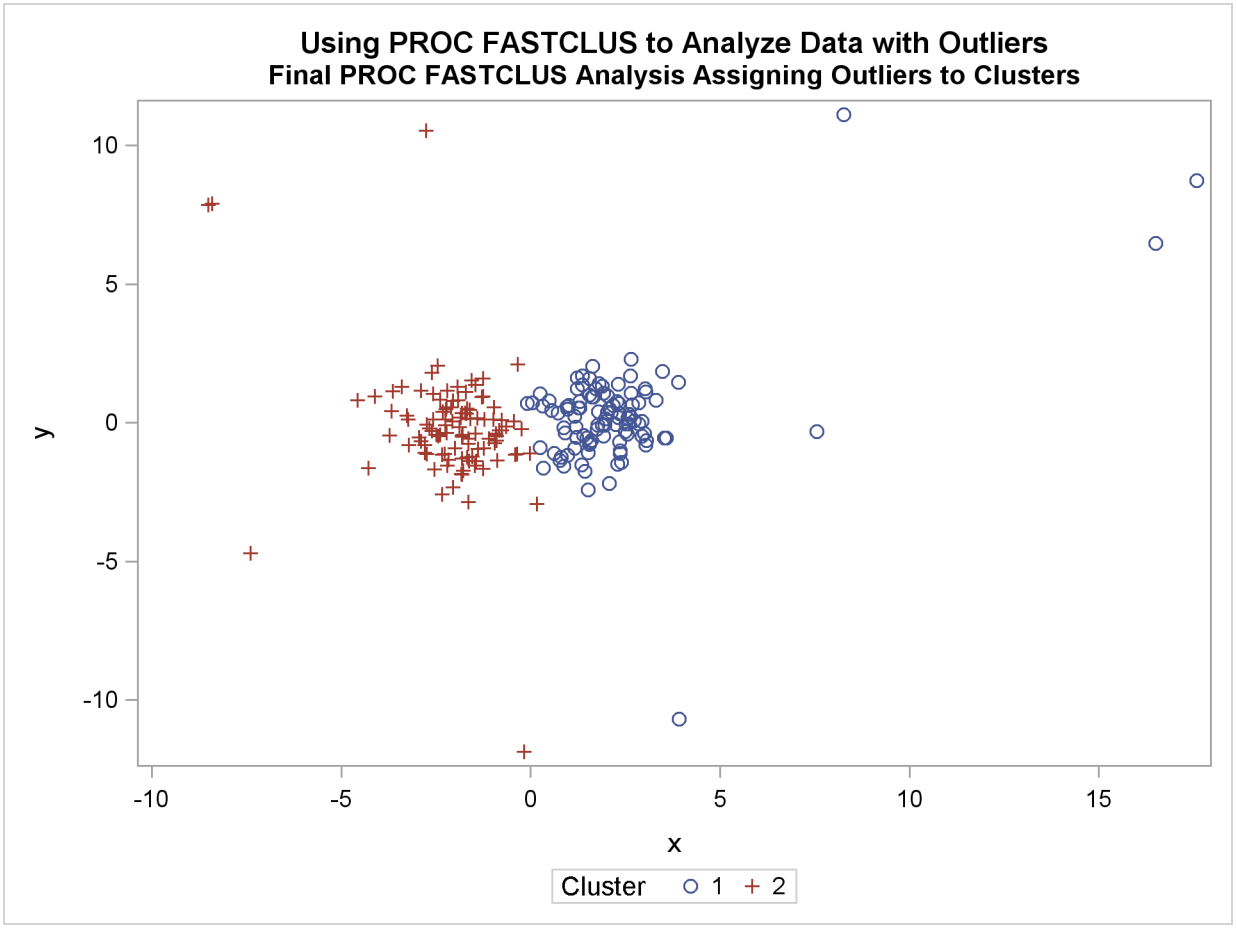
Cluster Means

Cluster	x	y
1	2.280017469	0.263940765
2	-2.075547895	-0.151348765

Cluster Standard Deviations

Cluster	x	y
1	2.412264861	2.089922815
2	1.379355878	2.201567557

Output 34.2.8 Cluster Analysis with Outliers Omitted: Plot Using PROC SGPLOT



References

Anderberg, M. R. (1973), *Cluster Analysis for Applications*, New York: Academic Press.

Bock, H. H. (1985), "On Some Significance Tests in Cluster Analysis," *Journal of Classification*, 2, 77–108.

Calinski, T. and Harabasz, J. (1974), "A Dendrite Method for Cluster Analysis," *Communications in Statistics*, 3, 1–27.

Cooper, M. C. and Milligan, G. W. (1988), "The Effect of Error on Determining the Number of Clusters," *Proceedings of the International Workshop on Data Analysis, Decision Support, and Expert Knowledge Representation in Marketing and Related Areas of Research*, 319–328.

Everitt, B. S. (1980), *Cluster Analysis*, Second Edition, London: Heineman Educational Books.

Fisher, R. A. (1936), "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, 7, 179–188.

Gonin, R. and Money, A. H. (1989), *Nonlinear L_p-Norm Estimation*, New York: Marcel Dekker.

Hartigan, J. A. (1975), *Clustering Algorithms*, New York: John Wiley & Sons.

Hartigan, J. A. (1985), "Statistical Theory in Clustering," *Journal of Classification*, 2, 63–76.

Journal of Statistics Education, "Fish Catch Data Set," http://www.amstat.org/publications/jse/jse_data_archive.html.

MacQueen, J. B. (1967), "Some Methods for Classification and Analysis of Multivariate Observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1, 281–297.

McLachlan, G. J. and Basford, K. E. (1988), *Mixture Models*, New York: Marcel Dekker.

Mezzich, J. E. and Solomon, H. (1980), *Taxonomy and Behavioral Science*, New York: Academic Press.

Milligan, G. W. (1980), "An Examination of the Effect of Six Types of Error Perturbation on Fifteen Clustering Algorithms," *Psychometrika*, 45, 325–342.

Milligan, G. W. and Cooper, M. C. (1985), "An Examination of Procedures for Determining the Number of Clusters in a Data Set," *Psychometrika, 50,* 159–179.

Pollard, D. (1981), "Strong Consistency of *k*-Means Clustering," *Annals of Statistics*, 9, 135–140.

Sarle, W. S. (1983), "The Cubic Clustering Criterion," SAS Technical Report A-108, Cary, NC: SAS Institute Inc.

Spath, H. (1980), *Cluster Analysis Algorithms*, Chichester, Eng.: Ellis Horwood.

Spath, H. (1985), *Cluster Dissection and Analysis*, Chichester, Eng.: Ellis Horwood.

Titterington, D. M., Smith, A. F. M., and Makov, U. E. (1985), *Statistical Analysis of Finite Mixture Distributions*, New York: John Wiley & Sons.

Tou, J. T. and Gonzalez, R. C. (1974), *Pattern Recognition Principles*, Reading, MA: Addison-Wesley.

Chapter 35

The FREQ Procedure

Contents

Overview: FREQ Procedure **1676**
Getting Started: FREQ Procedure **1678**
Frequency Tables and Statistics 1678
Agreement Study 1685
Syntax: FREQ Procedure **1687**
PROC FREQ Statement 1688
BY Statement 1690
EXACT Statement 1691
OUTPUT Statement 1694
TABLES Statement 1697
TEST Statement 1722
WEIGHT Statement 1723
Details: FREQ Procedure **1724**
Inputting Frequency Counts 1724
Grouping with Formats 1725
Missing Values 1726
In-Database Computation 1728
Statistical Computations 1730
Definitions and Notation 1730
Chi-Square Tests and Statistics 1731
Measures of Association 1736
Binomial Proportion 1745
Risks and Risk Differences 1752
Odds Ratio and Relative Risks for 2 x 2 Tables 1760
Cochran-Armitage Test for Trend 1762
Jonckheere-Terpstra Test 1763
Tests and Measures of Agreement 1765
Cochran-Mantel-Haenszel Statistics 1770
Exact Statistics 1779
Computational Resources 1783
Output Data Sets 1784
Displayed Output 1787
ODS Table Names 1795
ODS Graphics 1798

Examples: FREQ Procedure **1799**
Example 35.1: Output Data Set of Frequencies 1799
Example 35.2: Frequency Dot Plots 1802
Example 35.3: Chi-Square Goodness-of-Fit Tests 1805
Example 35.4: Binomial Proportions 1809
Example 35.5: Analysis of a 2x2 Contingency Table 1812
Example 35.6: Output Data Set of Chi-Square Statistics 1815
Example 35.7: Cochran-Mantel-Haenszel Statistics 1817
Example 35.8: Cochran-Armitage Trend Test 1818
Example 35.9: Friedman's Chi-Square Test 1822
Example 35.10: Cochran's Q Test 1824
References **1826**

Overview: FREQ Procedure

The FREQ procedure produces one-way to *n*-way frequency and contingency (crosstabulation) tables. For two-way tables, PROC FREQ computes tests and measures of association. For *n*-way tables, PROC FREQ provides stratified analysis by computing statistics across, as well as within, strata.

For one-way frequency tables, PROC FREQ computes goodness-of-fit tests for equal proportions or specified null proportions. For one-way tables, PROC FREQ also provides confidence limits and tests for binomial proportions, including tests for noninferiority and equivalence.

For contingency tables, PROC FREQ can compute various statistics to examine the relationships between two classification variables. For some pairs of variables, you might want to examine the existence or strength of any association between the variables. To determine if an association exists, chi-square tests are computed. To estimate the strength of an association, PROC FREQ computes measures of association that tend to be close to zero when there is no association and close to the maximum (or minimum) value when there is perfect association. The statistics for contingency tables include the following:

- chi-square tests and measures
- measures of association
- risks (binomial proportions) and risk differences for 2×2 tables
- odds ratios and relative risks for 2×2 tables
- tests for trend
- tests and measures of agreement
- Cochran-Mantel-Haenszel statistics

PROC FREQ computes asymptotic standard errors, confidence intervals, and tests for measures of association and measures of agreement. Exact *p*-values and confidence intervals are available for many test statistics and measures. PROC FREQ also performs analyses that adjust for any stratification variables by computing statistics across, as well as within, strata for *n*-way tables. These statistics include Cochran-Mantel-Haenszel statistics and measures of agreement.

In choosing measures of association to use in analyzing a two-way table, you should consider the study design (which indicates whether the row and column variables are dependent or independent), the measurement scale of the variables (nominal, ordinal, or interval), the type of association that each measure is designed to detect, and any assumptions required for valid interpretation of a measure. You should exercise care in selecting measures that are appropriate for your data.

Similar comments apply to the choice and interpretation of test statistics. For example, the Mantel-Haenszel chi-square statistic requires an ordinal scale for both variables and is designed to detect a linear association. The Pearson chi-square, on the other hand, is appropriate for all variables and can detect any kind of association, but it is less powerful for detecting a linear association because its power is dispersed over a greater number of degrees of freedom (except for 2×2 tables).

For more information about selecting the appropriate statistical analyses, see Agresti (2007) or Stokes, Davis, and Koch (2000).

Several SAS procedures produce frequency counts; only PROC FREQ computes chi-square tests for one-way to *n*-way tables and measures of association and agreement for contingency tables. Other procedures to consider for counting include the TABULATE and UNIVARIATE procedures. When you want to produce contingency tables and tests of association for sample survey data, use PROC SURVEYFREQ. See Chapter 14, "Introduction to Survey Procedures," for more information. When you want to fit models to categorical data, use a procedure such as CATMOD, GENMOD, GLIMMIX, LOGISTIC, PROBIT, or SURVEYLOGISTIC. See Chapter 8, "Introduction to Categorical Data Analysis Procedures," for more information.

PROC FREQ uses the Output Delivery System (ODS), a SAS subsystem that provides capabilities for displaying and controlling the output from SAS procedures. ODS enables you to convert any of the output from PROC FREQ into a SAS data set. See the section "ODS Table Names" on page 1795 for more information.

PROC FREQ now uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the statistical graphics available with the FREQ procedure, see the PLOTS option in the TABLES statement and the section "ODS Graphics" on page 1798.

Getting Started: FREQ Procedure

Frequency Tables and Statistics

The FREQ procedure provides easy access to statistics for testing for association in a crosstabulation table.

In this example, high school students applied for courses in a summer enrichment program; these courses included journalism, art history, statistics, graphic arts, and computer programming. The students accepted were randomly assigned to classes with and without internships in local companies. Table 35.1 contains counts of the students who enrolled in the summer program by gender and whether they were assigned an internship slot.

Table 35.1 Summer Enrichment Data

		Enrollment		
Gender	**Internship**	**Yes**	**No**	**Total**
boys	yes	35	29	64
boys	no	14	27	41
girls	yes	32	10	42
girls	no	53	23	76

The SAS data set SummerSchool is created by inputting the summer enrichment data as cell count data, or providing the frequency count for each combination of variable values. The following DATA step statements create the SAS data set SummerSchool:

```
data SummerSchool;
   input Gender $ Internship $ Enrollment $ Count @@;
   datalines;
boys  yes yes 35   boys  yes no 29
boys   no yes 14   boys   no no 27
girls yes yes 32   girls yes no 10
girls  no yes 53   girls  no no 23
;
```

The variable Gender takes the values 'boys' or 'girls,' the variable Internship takes the values 'yes' and 'no,' and the variable Enrollment takes the values 'yes' and 'no.' The variable Count contains the number of students that correspond to each combination of data values. The double at sign (@@) indicates that more than one observation is included on a single data line. In this DATA step, two observations are included on each line.

Researchers are interested in whether there is an association between internship status and summer program enrollment. The Pearson chi-square statistic is an appropriate statistic to assess the association in the corresponding 2×2 table. The following PROC FREQ statements specify this analysis.

You specify the table for which you want to compute statistics with the TABLES statement. You specify the statistics you want to compute with options after a slash (/) in the TABLES statement.

```
proc freq data=SummerSchool order=data;
   tables Internship*Enrollment / chisq;
   weight Count;
run;
```

The ORDER= option controls the order in which variable values are displayed in the rows and columns of the table. By default, the values are arranged according to the alphanumeric order of their unformatted values. If you specify ORDER=DATA, the data are displayed in the same order as they occur in the input data set. Here, because 'yes' appears before 'no' in the data, 'yes' appears first in any table. Other options for controlling order include ORDER=FORMATTED, which orders according to the formatted values, and ORDER=FREQUENCY, which orders by descending frequency count.

In the TABLES statement, Internship*Enrollment specifies a table where the rows are internship status and the columns are program enrollment. The CHISQ option requests chi-square statistics for assessing association between these two variables. Because the input data are in cell count form, the WEIGHT statement is required. The WEIGHT statement names the variable Count, which provides the frequency of each combination of data values.

Figure 35.1 presents the crosstabulation of Internship and Enrollment. In each cell, the values printed under the cell count are the table percentage, row percentage, and column percentage, respectively. For example, in the first cell, 63.21 percent of the students offered courses with internships accepted them and 36.79 percent did not.

Figure 35.1 Crosstabulation Table

```
                    The FREQ Procedure

             Table of Internship by Enrollment

Internship     Enrollment

Frequency|
Percent  |
Row Pct  |
Col Pct  |yes     |no      |  Total
---------+--------+--------+
yes      |     67 |     39 |    106
         |  30.04 |  17.49 |  47.53
         |  63.21 |  36.79 |
         |  50.00 |  43.82 |
---------+--------+--------+
no       |     67 |     50 |    117
         |  30.04 |  22.42 |  52.47
         |  57.26 |  42.74 |
         |  50.00 |  56.18 |
---------+--------+--------+
Total         134       89      223
            60.09    39.91   100.00
```

Figure 35.2 displays the statistics produced by the CHISQ option. The Pearson chi-square statistic is labeled 'Chi-Square' and has a value of 0.8189 with 1 degree of freedom. The associated p-value is 0.3655, which means that there is no significant evidence of an association between internship status and program enrollment. The other chi-square statistics have similar values and are asymptotically equivalent. The other statistics (phi coefficient, contingency coefficient, and Cramer's V) are measures of association derived from the Pearson chi-square. For Fisher's exact test, the two-sided p-value is 0.4122, which also shows no association between internship status and program enrollment.

Figure 35.2 Statistics Produced with the CHISQ Option

Statistic	DF	Value	Prob
Chi-Square	1	0.8189	0.3655
Likelihood Ratio Chi-Square	1	0.8202	0.3651
Continuity Adj. Chi-Square	1	0.5899	0.4425
Mantel-Haenszel Chi-Square	1	0.8153	0.3666
Phi Coefficient		0.0606	
Contingency Coefficient		0.0605	
Cramer's V		0.0606	

Fisher's Exact Test	
Cell (1,1) Frequency (F)	67
Left-sided Pr <= F	0.8513
Right-sided Pr >= F	0.2213
Table Probability (P)	0.0726
Two-sided Pr <= P	0.4122

The analysis, so far, has ignored gender. However, it might be of interest to ask whether program enrollment is associated with internship status after adjusting for gender. You can address this question by doing an analysis of a set of tables (in this case, by analyzing the set consisting of one for boys and one for girls). The Cochran-Mantel-Haenszel (CMH) statistic is appropriate for this situation: it addresses whether rows and columns are associated after controlling for the stratification variable. In this case, you would be stratifying by gender.

The PROC FREQ statements for this analysis are very similar to those for the first analysis, except that there is a third variable, Gender, in the TABLES statement. When you cross more than two variables, the two rightmost variables construct the rows and columns of the table, respectively, and the leftmost variables determine the stratification.

The following PROC FREQ statements also request frequency plots for the crosstabulation tables. PROC FREQ produces these plots by using ODS Graphics to create graphs as part of the procedure output. Before requesting graphs, you must enable ODS Graphics with the **ODS GRAPHICS ON** statement.

```
ods graphics on;
proc freq data=SummerSchool;
   tables Gender*Internship*Enrollment /
          chisq cmh plots(only)=freqplot;
   weight Count;
run;
ods graphics off;
```

This execution of PROC FREQ first produces two individual crosstabulation tables of Internship by Enrollment: one for boys and one for girls. Frequency plots and chi-square statistics are produced for each individual table. Figure 35.3, Figure 35.4, and Figure 35.5 show the results for boys. Note that the chi-square statistic for boys is significant at the $\alpha = 0.05$ level of significance. Boys offered a course with an internship are more likely to enroll than boys who are not.

Figure 35.4 displays the frequency plot of Internship by Enrollment for boys. By default, the frequency plot is displayed as a bar chart with vertical grouping by the row variable Internship. You can use PLOTS= options to request a dot plot instead of a bar chart or to change the orientation of the bars from vertical to horizontal. You can also use PLOTS= options to specify other two-way layouts such as stacked or horizontal grouping.

Figure 35.6, Figure 35.7, and Figure 35.8 display the crosstabulation table, frequency plot, and chi-square statistics for girls. You can see that there is no evidence of association between internship offers and program enrollment for girls.

Figure 35.3 Crosstabulation Table for Boys

The FREQ Procedure

Table 1 of Internship by Enrollment
Controlling for Gender=boys

Internship	Enrollment		
Frequency | Percent | Row Pct | Col Pct	no	yes	Total
no	27	14	41
	25.71	13.33	39.05
	65.85	34.15	
	48.21	28.57	
yes	29	35	64
	27.62	33.33	60.95
	45.31	54.69	
	51.79	71.43	
Total	56	49	105
	53.33	46.67	100.00

Figure 35.4 Frequency Plot for Boys

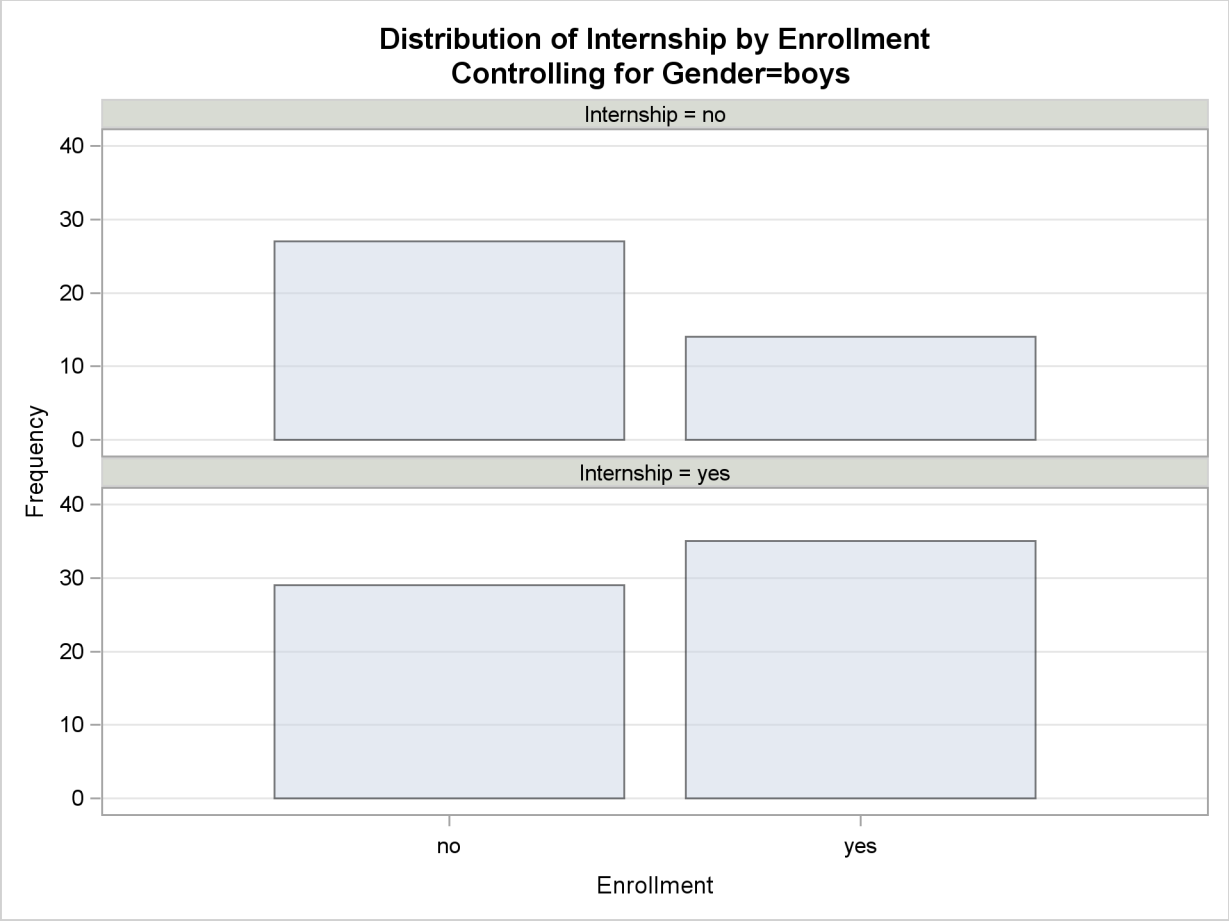


Figure 35.5 Chi-Square Statistics for Boys

Statistic	DF	Value	Prob
Chi-Square	1	4.2366	0.0396
Likelihood Ratio Chi-Square	1	4.2903	0.0383
Continuity Adj. Chi-Square	1	3.4515	0.0632
Mantel-Haenszel Chi-Square	1	4.1963	0.0405
Phi Coefficient		0.2009	
Contingency Coefficient		0.1969	
Cramer's V		0.2009	

Fisher's Exact Test

Cell (1,1) Frequency (F)	27
Left-sided Pr <= F	0.9885
Right-sided Pr >= F	0.0311
Table Probability (P)	0.0196
Two-sided Pr <= P	0.0467

Figure 35.6 Crosstabulation Table for Girls

```
Table 2 of Internship by Enrollment
    Controlling for Gender=girls

Internship     Enrollment

Frequency|
Percent  |
Row Pct  |
Col Pct  |no      |yes     |  Total
---------+--------+--------+
no       |     23 |     53 |     76
         |  19.49 |  44.92 |  64.41
         |  30.26 |  69.74 |
         |  69.70 |  62.35 |
---------+--------+--------+
yes      |     10 |     32 |     42
         |   8.47 |  27.12 |  35.59
         |  23.81 |  76.19 |
         |  30.30 |  37.65 |
---------+--------+--------+
Total          33       85      118
            27.97    72.03   100.00
```

Figure 35.7 Frequency Plot for Girls

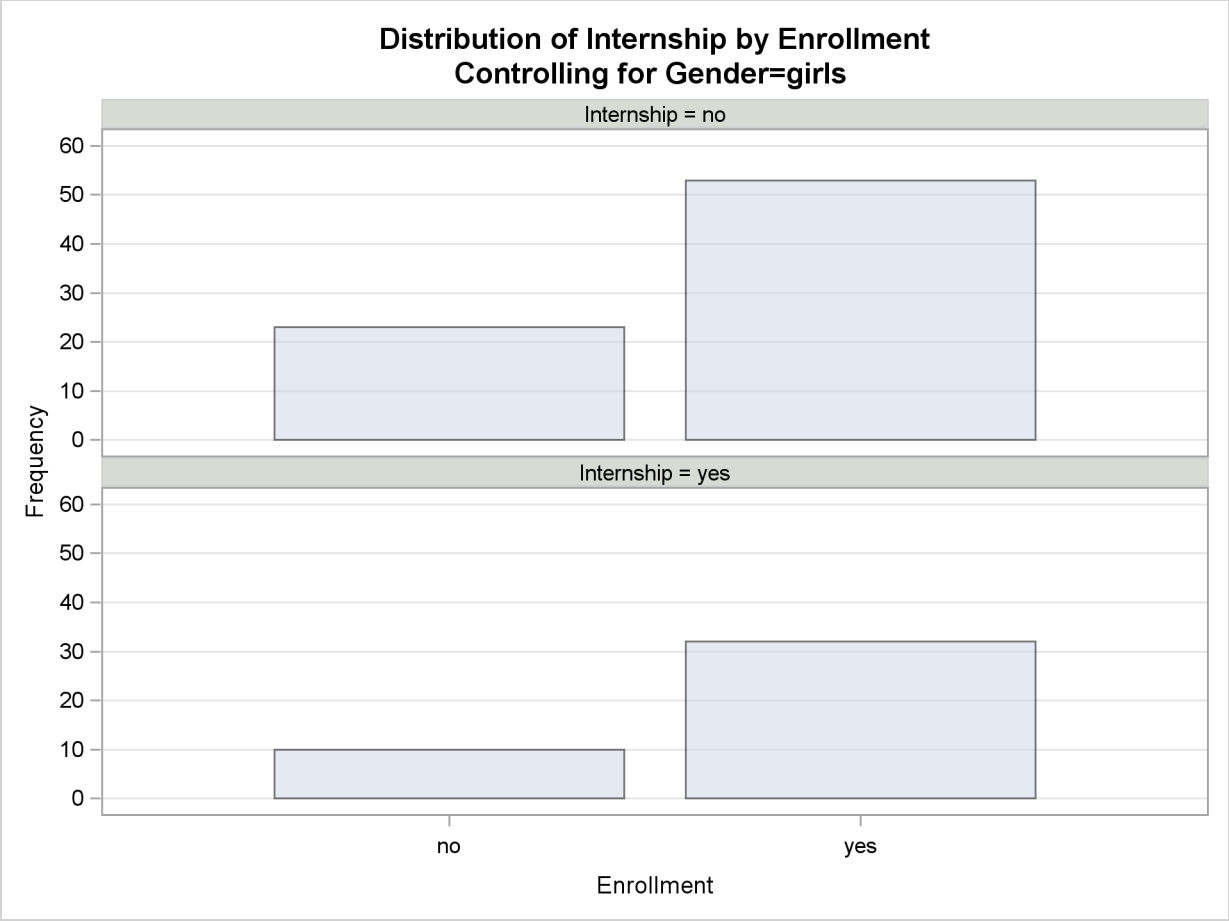


Figure 35.8 Chi-Square Statistics for Girls

Statistic	DF	Value	Prob
Chi-Square	1	0.5593	0.4546
Likelihood Ratio Chi-Square	1	0.5681	0.4510
Continuity Adj. Chi-Square	1	0.2848	0.5936
Mantel-Haenszel Chi-Square	1	0.5545	0.4565
Phi Coefficient		0.0688	
Contingency Coefficient		0.0687	
Cramer's V		0.0688	

Fisher's Exact Test

Cell (1,1) Frequency (F)	23
Left-sided Pr <= F	0.8317
Right-sided Pr >= F	0.2994
Table Probability (P)	0.1311
Two-sided Pr <= P	0.5245

These individual table results demonstrate the occasional problems with combining information into one table and not accounting for information in other variables such as Gender. Figure 35.9 contains the CMH results. There are three summary (CMH) statistics; which one you use depends on whether your rows and/or columns have an order in $r \times c$ tables. However, in the case of 2×2 tables, ordering does not matter and all three statistics take the same value. The CMH statistic follows the chi-square distribution under the hypothesis of no association, and here, it takes the value 4.0186 with 1 degree of freedom. The associated p-value is 0.0450, which indicates a significant association at the $\alpha = 0.05$ level.

Thus, when you adjust for the effect of gender in these data, there is an association between internship and program enrollment. But, if you ignore gender, no association is found. Note that the CMH option also produces other statistics, including estimates and confidence limits for relative risk and odds ratios for 2×2 tables and the Breslow-Day Test. These results are not displayed here.

Figure 35.9 Test for the Hypothesis of No Association

Cochran-Mantel-Haenszel Statistics (Based on Table Scores)

Statistic	Alternative Hypothesis	DF	Value	Prob
1	Nonzero Correlation	1	4.0186	0.0450
2	Row Mean Scores Differ	1	4.0186	0.0450
3	General Association	1	4.0186	0.0450

Agreement Study

Medical researchers are interested in evaluating the efficacy of a new treatment for a skin condition. Dermatologists from participating clinics were trained to conduct the study and to evaluate the condition. After the training, two dermatologists examined patients with the skin condition from a pilot study and rated the same patients. The possible evaluations are terrible, poor, marginal, and clear. Table 35.2 contains the data.

Table 35.2 Skin Condition Data

	Dermatologist 2			
Dermatologist 1	**Terrible**	**Poor**	**Marginal**	**Clear**
Terrible	10	4	1	0
Poor	5	10	12	2
Marginal	2	4	12	5
Clear	0	2	6	13

The following DATA step statements create the SAS dataset SkinCondition. The dermatologists' evaluations of the patients are contained in the variables Derm1 and Derm2; the variable Count is the number of patients given a particular pair of ratings.

```
data SkinCondition;
   input Derm1 $ Derm2 $ Count;
   datalines;
terrible terrible 10
terrible     poor 4
terrible marginal 1
terrible    clear 0
poor     terrible 5
poor         poor 10
poor     marginal 12
poor        clear 2
marginal terrible 2
marginal     poor 4
marginal marginal 12
marginal    clear 5
clear    terrible 0
clear        poor 2
clear    marginal 6
clear       clear 13
;
```

The following PROC FREQ statements request an agreement analysis of the skin condition data. In order to evaluate the agreement of the diagnoses (a possible contribution to measurement error in the study), the *kappa coefficient* is computed. The AGREE option in the TABLES statement requests the kappa coefficient, together with its standard error and confidence limits. The KAPPA option in the TEST statement requests a test for the null hypothesis that kappa equals zero, or that the agreement is purely by chance.

```
proc freq data=SkinCondition order=data;
   tables Derm1*Derm2 / agree noprint;
   test kappa;
   weight Count;
run;
```

Figure 35.10 shows the results. The kappa coefficient has the value 0.3449, which indicates slight agreement between the dermatologists, and the hypothesis test confirms that you can reject the null hypothesis of no agreement. This conclusion is further supported by the confidence interval of (0.2030, 0.4868), which suggests that the true kappa is greater than zero. The AGREE option also produces Bowker's test for symmetry and the weighted kappa coefficient, but that output is not shown here.

Figure 35.10 Agreement Study

The FREQ Procedure

Statistics for Table of Derm1 by Derm2

Simple Kappa Coefficient

Kappa	0.3449
ASE	0.0724
95% Lower Conf Limit	0.2030
95% Upper Conf Limit	0.4868

Test of H0: Kappa = 0

ASE under H0	0.0612
Z	5.6366
One-sided Pr > Z	<.0001
Two-sided Pr > |Z|	<.0001

Syntax: FREQ Procedure

The following statements are available in PROC FREQ:

PROC FREQ < *options* > ;
BY *variables* ;
EXACT *statistic-options* < / *computation-options* > ;
OUTPUT < **OUT=***SAS-data-set* > *options* ;
TABLES *requests* < / *options* > ;
TEST *options* ;
WEIGHT *variable* < / *option* > ;

The PROC FREQ statement is the only required statement for the FREQ procedure. If you specify the following statements, PROC FREQ produces a one-way frequency table for each variable in the most recently created data set.

```
proc freq;
run;
```

The rest of this section gives detailed syntax information for the BY, EXACT, OUTPUT, TABLES, TEST, and WEIGHT statements in alphabetical order after the description of the PROC FREQ statement. Table 35.3 summarizes the basic function of each PROC FREQ statement.

Table 35.3 Summary of PROC FREQ Statements

Statement	Description
BY	provides separate analyses for each BY group
EXACT	requests exact tests
OUTPUT	requests an output data set
TABLES	specifies tables and requests analyses
TEST	requests tests for measures of association and agreement
WEIGHT	identifies a weight variable

PROC FREQ Statement

PROC FREQ < *options* > **;**

The PROC FREQ statement invokes the procedure and optionally identifies the input data set. By default, the procedure uses the most recently created SAS data set.

Table 35.4 lists the options available in the PROC FREQ statement. Descriptions follow in alphabetical order.

Table 35.4 PROC FREQ Statement Options

Option	Description
COMPRESS	begins the next one-way table on the current page
DATA=	names the input data set
FORMCHAR=	specifies the outline and cell divider characters for crosstabulation tables
NLEVELS	displays the number of levels for all TABLES variables
NOPRINT	suppresses all displayed output
ORDER=	specifies the order for reporting variable values
PAGE	displays one table per page

You can specify the following options in the PROC FREQ statement.

COMPRESS

begins display of the next one-way frequency table on the same page as the preceding one-way table if there is enough space to begin the table. By default, the next one-way table begins on the current page only if the entire table fits on that page. The COMPRESS option is not valid with the PAGE option.

DATA=*SAS-data-set*

names the SAS data set to be analyzed by PROC FREQ. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

FORMCHAR(1,2,7)=*'formchar-string'*

defines the characters to be used for constructing the outlines and dividers for the cells of crosstabulation table displays. The *formchar-string* should be three characters long. The characters are used to draw the vertical separators (1), the horizontal separators (2), and the vertical-horizontal intersections (7). If you do not specify the FORMCHAR= option, PROC FREQ uses FORMCHAR(1,2,7)='|-+' by default. Table 35.5 summarizes the formatting characters used by PROC FREQ.

Table 35.5 Formatting Characters Used by PROC FREQ

Position	Default	Used to Draw
1	|	vertical separators
2	-	horizontal separators
7	+	intersections of vertical and horizontal separators

The FORMCHAR= option can specify 20 different SAS formatting characters used to display output; however, PROC FREQ uses only the first, second, and seventh formatting characters. Therefore, the proper specification for PROC FREQ is FORMCHAR(1,2,7)= '*formchar-string*'.

Specifying all blanks for *formchar-string* produces crosstabulation tables with no outlines or dividers—for example, FORMCHAR(1,2,7)=' '. You can use any character in *formchar-string*, including hexadecimal characters. If you use hexadecimal characters, you must put an *x* after the closing quote. For information about which hexadecimal codes to use for which characters, see the documentation for your hardware.

See the CALENDAR, PLOT, and TABULATE procedures in the *Base SAS Procedures Guide* for more information about form characters.

NLEVELS

displays the "Number of Variable Levels" table, which provides the number of levels for each variable named in the TABLES statements. See the section "Number of Variable Levels Table" on page 1787 for details. PROC FREQ determines the variable levels from the formatted variable values, as described in the section "Grouping with Formats" on page 1725.

NOPRINT

suppresses the display of all output. You can use the NOPRINT option when you only want to create an output data set. See the section "Output Data Sets" on page 1784 for information about the output data sets produced by PROC FREQ. Note that the NOPRINT option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 20, "Using the Output Delivery System."

NOTE: A NOPRINT option is also available in the TABLES statement. It suppresses display of the crosstabulation tables but allows display of the requested statistics.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the order in which the values of the frequency and crosstabulation table variables are reported. PROC FREQ interprets the values of the ORDER= option as follows:

DATA	orders values according to their order in the input data set
FORMATTED	orders values by their formatted values (in ascending order). This order is dependent on the operating environment.
FREQ	orders values by their descending frequency counts
INTERNAL	orders values by their unformatted values, which yields the same order that the SORT procedure does. This order is dependent on the operating environment.

By default, ORDER=INTERNAL. The ORDER= option does not apply to missing values, which are always ordered first.

PAGE

displays only one table per page. Otherwise, PROC FREQ displays multiple tables per page as space permits. The PAGE option is not valid with the COMPRESS option.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC FREQ to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the FREQ procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

EXACT Statement

EXACT *statistic-options* < / *computation-options* > **;**

The EXACT statement requests exact tests or confidence limits for the specified statistics. Optionally, PROC FREQ computes Monte Carlo estimates of the exact *p*-values. The *statistic-options* specify the statistics to provide exact tests or confidence limits for. The *computation-options* specify options for the computation of exact statistics. See the section "Exact Statistics" on page 1779 for details.

CAUTION: PROC FREQ computes exact tests with fast and efficient algorithms that are superior to direct enumeration. Exact tests are appropriate when a data set is small, sparse, skewed, or heavily tied. For some large problems, computation of exact tests might require a considerable amount of time and memory. Consider using asymptotic tests for such problems. Alternatively, when asymptotic methods might not be sufficient for such large problems, consider using Monte Carlo estimation of exact *p*-values. See the section "Computational Resources" on page 1781 for more information.

Statistic-Options

The *statistic-options* specify the statistics to provide exact tests or confidence limits for.

For one-way tables, exact *p*-values are available for the binomial proportion tests and the chi-square goodness-of-fit test. Exact confidence limits are available for the binomial proportion.

For two-way tables, exact *p*-values are available for the following tests: Pearson chi-square test, likelihood-ratio chi-square test, Mantel-Haenszel chi-square test, Fisher's exact test, Jonckheere-Terpstra test, and Cochran-Armitage test for trend. Exact *p*-values are also available for tests of the following statistics: Pearson correlation coefficient, Spearman correlation coefficient, simple kappa coefficient, and weighted kappa coefficient.

For 2×2 tables, PROC FREQ provides exact confidence limits for the odds ratio, exact unconditional confidence limits for the proportion difference, and McNemar's exact test. For stratified 2×2 tables, PROC FREQ provides Zelen's exact test for equal odds ratios, exact confidence limits for the common odds ratio, and an exact test for the common odds ratio.

Table 35.6 lists the available *statistic-options* and the exact statistics computed. Most of the option names are identical to the corresponding option names in the TABLES and OUTPUT statements. You can request exact computations for groups of statistics by using options that are identical to the following TABLES statement options: CHISQ, MEASURES, and AGREE. For example, when you specify the CHISQ option in the EXACT statement, PROC FREQ computes exact *p*-values for the Pearson chi-square, likelihood-ratio chi-square, and Mantel-Haenszel chi-square tests. You can request exact computations for an individual statistic by specifying the corresponding *statistic-option* from the list in Table 35.6.

Table 35.6 EXACT Statement *Statistic-Options*

Statistic-Option	Exact Statistics Computed
AGREE	McNemar's test (for 2 × 2 tables), simple kappa coefficient test, weighted kappa coefficient test
BINOMIAL	binomial proportion tests for one-way tables
CHISQ	chi-square goodness-of-fit test for one-way tables; Pearson chi-square, likelihood-ratio chi-square, and Mantel-Haenszel chi-square tests for two-way tables
COMOR	confidence limits for the common odds ratio and common odds ratio test (for $h \times 2 \times 2$ tables)
EQOR | ZELEN	Zelen's test for equal odds ratios (for $h \times 2 \times 2$ tables)
FISHER	Fisher's exact test
JT	Jonckheere-Terpstra test
KAPPA	test for the simple kappa coefficient
LRCHI	likelihood-ratio chi-square test
MCNEM	McNemar's test (for 2 × 2 tables)
MEASURES	tests for the Pearson correlation and Spearman correlation, confidence limits for the odds ratio (for 2 × 2 tables)
MHCHI	Mantel-Haenszel chi-square test
OR	confidence limits for the odds ratio (for 2 × 2 tables)
PCHI	Pearson chi-square test
PCORR	test for the Pearson correlation coefficient
RISKDIFF	confidence limits for the proportion differences (for 2 × 2 tables)
RISKDIFF1	confidence limits for the column 1 proportion difference
RISKDIFF2	confidence limits for the column 2 proportion difference
SCORR	test for the Spearman correlation coefficient
TREND	Cochran-Armitage test for trend
WTKAP	test for the weighted kappa coefficient

Using TABLES Statement Options with the EXACT Statement

If you use only one TABLES statement, you do not need to specify the same options in both the TABLES and EXACT statements; when you specify an option in the EXACT statement, PROC FREQ automatically invokes the corresponding TABLES statement option. However, when you use multiple TABLES statements and want exact computations, you must specify options in the TABLES statements to request the desired statistics. PROC FREQ then performs exact computations for all statistics that you also specify in the EXACT statement.

Note that the TABLES statement group option CHISQ includes tests that correspond to the following EXACT statement individual statistic-options: LRCHI, MHCHI, and PCHI. The MEASURES option in the TABLES statement includes statistics that correspond to the following EXACT statement statistic-options: OR, PCORR, and SCORR. The AGREE option in the TABLES statement produces analyses that correspond to the KAPPA, MCNEM, and WTKAP statistic-options in the EXACT statement. The CMH option in the TABLES statement produces analyses that correspond to the COMOR and EQOR (or ZELEN) statistic-options in the EXACT statement.

Computation-Options

The *computation-options* specify options for computation of exact statistics. You can specify the following *computation-options* in the EXACT statement after a slash (/).

ALPHA=α

specifies the level of the confidence limits for Monte Carlo p-value estimates. The value of α must be between 0 and 1, and the default is 0.01. A confidence level of α produces $100(1-\alpha)\%$ confidence limits. The default of ALPHA=.01 produces 99% confidence limits for the Monte Carlo estimates.

The ALPHA= option invokes the MC option.

MAXTIME=*value*

specifies the maximum clock time (in seconds) that PROC FREQ can use to compute an exact p-value. If the procedure does not complete the computation within the specified time, the computation terminates. The value of MAXTIME= must be a positive number. The MAXTIME= option is valid for Monte Carlo estimation of exact p-values, as well as for direct exact p-value computation. See the section "Computational Resources" on page 1781 for more information.

MC

requests Monte Carlo estimation of exact p-values instead of direct exact p-value computation. Monte Carlo estimation can be useful for large problems that require a considerable amount of time and memory for exact computations but for which asymptotic approximations might not be sufficient. See the section "Monte Carlo Estimation" on page 1782 for more information.

The MC option is available for all EXACT *statistic-options* except the BINOMIAL option and the following options that apply only to 2×2 or $h \times 2 \times 2$ tables: COMOR, EQOR, MCNEM, RISKDIFF, and OR. PROC FREQ computes only exact tests or confidence limits for these statistics.

The ALPHA=, N=, and SEED= options also invoke the MC option.

N=*n*

specifies the number of samples for Monte Carlo estimation. The value of n must be a positive integer, and the default is 10,000. Larger values of n produce more precise estimates of exact p-values. Because larger values of n generate more samples, the computation time increases.

The N= option invokes the MC option.

POINT

requests exact point probabilities for the test statistics.

The POINT option is available for all the EXACT statement *statistic-options* except the OR and RISKDIFF options, which provide exact confidence limits. The POINT option is not available with the MC option.

SEED=*number*

specifies the initial seed for random number generation for Monte Carlo estimation. The value of the SEED= option must be an integer. If you do not specify the SEED= option or if the SEED= value is negative or zero, PROC FREQ uses the time of day from the computer's clock to obtain the initial seed.

The SEED= option invokes the MC option.

OUTPUT Statement

OUTPUT < *OUT= SAS-data-set* > *options* **;**

The OUTPUT statement creates a SAS data set that contains statistics computed by PROC FREQ. You specify which statistics to store in the output data set with the OUTPUT statement *options*. The output data set contains one observation for each two-way table or stratum, and one observation for summary statistics across all strata. For more information about the contents of the output data set, see the section "Contents of the OUTPUT Statement Output Data Set" on page 1786.

Only one OUTPUT statement is allowed for each execution of PROC FREQ. You must specify a TABLES statement with the OUTPUT statement. If you use multiple TABLES statements, the contents of the OUTPUT data set correspond to the last TABLES statement. If you use multiple table requests in a TABLES statement, the contents of the OUTPUT data set correspond to the last table request.

Note that you can use the Output Delivery System (ODS) to create a SAS data set from any piece of PROC FREQ output. For more information, see the section "ODS Table Names" on page 1795.

Also note that the output data set created by the OUTPUT statement is not the same as the output data set created by the OUT= option in the TABLES statement. The OUTPUT statement creates a data set that contains statistics (such as the Pearson chi-square and its p-value), and the OUT= option in the TABLES statement creates a data set that contains frequency table counts and percentages. See the section "Output Data Sets" on page 1784 for more information.

You can specify the following options in an OUTPUT statement:

OUT=*SAS-data-set*

names the output data set. If you omit the OUT= option, the data set is named DATAn, where n is the smallest integer that makes the name unique.

options

specify the statistics you want in the output data set. Table 35.7 lists the available options, together with the TABLES statement options needed to produce the statistics. You can output groups of statistics by using group options identical to those available in the TABLES statement, which include the AGREE, ALL, CHISQ, CMH, and MEASURES options. Or you can request statistics individually.

When you specify an option in the OUTPUT statement, the output data set contains all statistics from that analysis—the estimate or test statistic plus any associated standard error, confidence limits, p-values, and degrees of freedom. See the section "Contents of the OUTPUT Statement Output Data Set" on page 1786 for details.

If you want to store a statistic in the output data set, you must also request computation of that statistic with the appropriate TABLES or EXACT statement option. For example, you cannot specify the option PCHI (Pearson chi-square) in the OUTPUT statement without also specifying a TABLES or EXACT statement option to compute the Pearson chi-square test. The TABLES statement option ALL or CHISQ requests the Pearson chi-square test. If you have only one TABLES statement, the EXACT statement option CHISQ or PCHI also requests the Pearson chi-square test. Table 35.7 lists the TABLES statement options required to produce the OUTPUT data set statistics. Note that the ALL option in the TABLES statement invokes the CHISQ, MEASURES, and CMH options.

Table 35.7 OUTPUT Statement Options

Option	Output Data Set Statistics	Required TABLES Statement Option
AGREE	McNemar's test, Bowker's test, simple and weighted kappas; for multiple strata, overall simple and weighted kappas, tests for equal kappas, and Cochran's Q	AGREE
AJCHI	continuity-adjusted chi-square (2×2 tables)	CHISQ
ALL	CHISQ, MEASURES, and CMH statistics; N	ALL
BDCHI	Breslow-Day test ($h \times 2 \times 2$ tables)	CMH, CMH1, or CMH2
BINOMIAL	binomial statistics for one-way tables	BINOMIAL
CHISQ	for one-way tables, goodness-of-fit test; for two-way tables, Pearson, likelihood-ratio, continuity-adjusted, and Mantel-Haenszel chi-squares, Fisher's exact test (2×2 tables), phi and contingency coefficients, Cramer's V	CHISQ
CMH	Cochran-Mantel-Haenszel (CMH) correlation, row mean scores (ANOVA), and general association statistics; for 2×2 tables, logit and Mantel-Haenszel adjusted odds ratios and relative risks, Breslow-Day test	CMH
CMH1	CMH output, except row mean scores (ANOVA) and general association statistics	CMH or CMH1
CMH2	CMH output, except general association statistic	CMH or CMH2
CMHCOR	CMH correlation statistic	CMH, CMH1, or CMH2
CMHGA	CMH general association statistic	CMH
CMHRMS	CMH row mean scores (ANOVA) statistic	CMH or CMH2
COCHQ	Cochran's Q ($h \times 2 \times 2$ tables)	AGREE
CONTGY	contingency coefficient	CHISQ
CRAMV	Cramer's V	CHISQ
EQKAP	test for equal simple kappas	AGREE
EQOR | ZELEN	Zelen's test for equal odds ratios ($h \times 2 \times 2$ tables)	CMH and EXACT EQOR
EQWKP	test for equal weighted kappas	AGREE
FISHER	Fisher's exact test	CHISQ or FISHER [1]
GAMMA	gamma	MEASURES

[1] CHISQ computes Fisher's exact test for 2×2 tables. Use the FISHER option to compute Fisher's exact test for general *r*x*c* tables.

Table 35.7 *continued*

Option	Output Data Set Statistics	Required TABLES Statement Option
JT	Jonckheere-Terpstra test	JT
KAPPA	simple kappa coefficient	AGREE
KENTB	Kendall's tau-b	MEASURES
LAMCR	lambda asymmetric $(C\|R)$	MEASURES
LAMDAS	lambda symmetric	MEASURES
LAMRC	lambda asymmetric $(R\|C)$	MEASURES
LGOR	adjusted logit odds ratio ($h \times 2 \times 2$ tables)	CMH, CMH1, or CMH2
LGRRC1	adjusted column 1 logit relative risk	CMH, CMH1, or CMH2
LGRRC2	adjusted column 2 logit relative risk	CMH, CMH1, or CMH2
LRCHI	likelihood-ratio chi-square	CHISQ
MCNEM	McNemar's test (2×2 tables)	AGREE
MEASURES	gamma, Kendall's tau-b, Stuart's tau-c, Somers' $D(C\|R)$ and $D(R\|C)$, Pearson and Spearman correlations, lambda asymmetric $(C\|R)$ and $(R\|C)$, lambda symmetric, uncertainty coefficients $(C\|R)$ and $(R\|C)$, symmetric uncertainty coefficient; odds ratio and relative risks (2×2 tables)	MEASURES
MHCHI	Mantel-Haenszel chi-square	CHISQ
MHOR | COMOR	adjusted Mantel-Haenszel odds ratio ($h \times 2 \times 2$ tables)	CMH, CMH1, or CMH2
MHRRC1	adjusted column 1 Mantel-Haenszel relative risk	CMH, CMH1, or CMH2
MHRRC2	adjusted column 2 Mantel-Haenszel relative risk	CMH, CMH1, or CMH2
N	number of nonmissing observations	
NMISS	number of missing observations	
OR	odds ratio (2×2 tables)	MEASURES or RELRISK
PCHI	chi-square goodness-of-fit test for one-way tables, Pearson chi-square for two-way tables	CHISQ
PCORR	Pearson correlation coefficient	MEASURES
PHI	phi coefficient	CHISQ
PLCORR	polychoric correlation coefficient	PLCORR
RDIF1	column 1 risk difference (row 1 - row 2)	RISKDIFF
RDIF2	column 2 risk difference (row 1 - row 2)	RISKDIFF
RELRISK	odds ratio and relative risks (2×2 tables)	MEASURES or RELRISK
RISKDIFF	risks and risk differences (2×2 tables)	RISKDIFF
RISKDIFF1	column 1 risks and risk difference	RISKDIFF
RISKDIFF2	column 2 risks and risk difference	RISKDIFF
RRC1	column 1 relative risk	MEASURES or RELRISK
RRC2	column 2 relative risk	MEASURES or RELRISK
RSK1	column 1 risk, overall	RISKDIFF
RSK11	column 1 risk, for row 1	RISKDIFF
RSK12	column 2 risk, for row 1	RISKDIFF
RSK2	column 2 risk, overall	RISKDIFF
RSK21	column 1 risk, for row 2	RISKDIFF

Table 35.7 *continued*

Option	Output Data Set Statistics	Required TABLES Statement Option
RSK22	column 2 risk, for row 2	RISKDIFF
SCORR	Spearman correlation coefficient	MEASURES
SMDCR	Somers' $D(C\|R)$	MEASURES
SMDRC	Somers' $D(R\|C)$	MEASURES
STUTC	Stuart's tau-c	MEASURES
TREND	Cochran-Armitage test for trend	TREND
TSYMM	Bowker's test of symmetry	AGREE
U	symmetric uncertainty coefficient	MEASURES
UCR	uncertainty coefficient $(C\|R)$	MEASURES
URC	uncertainty coefficient $(R\|C)$	MEASURES
WTKAP	weighted kappa coefficient	AGREE

TABLES Statement

TABLES *requests* < / *options* > **;**

The TABLES statement requests one-way to n-way frequency and crosstabulation tables and statistics for those tables.

If you omit the TABLES statement, PROC FREQ generates one-way frequency tables for all data set variables that are not listed in the other statements.

The following argument is required in the TABLES statement.

requests

specify the frequency and crosstabulation tables to produce. A request is composed of one variable name or several variable names separated by asterisks. To request a one-way frequency table, use a single variable. To request a two-way crosstabulation table, use an asterisk between two variables. To request a multiway table (an n-way table, where n>2), separate the desired variables with asterisks. The unique values of these variables form the rows, columns, and strata of the table. You can include up to 50 variables in a single multiway table request.

For two-way to multiway tables, the values of the last variable form the crosstabulation table columns, while the values of the next-to-last variable form the rows. Each level (or combination of levels) of the other variables forms one stratum. PROC FREQ produces a separate crosstabulation table for each stratum. For example, a specification of A*B*C*D in a TABLES statement produces k tables, where k is the number of different combinations of values for A and B. Each table lists the values for C down the side and the values for D across the top.

You can use multiple TABLES statements in the PROC FREQ step. PROC FREQ builds all the table requests in one pass of the data, so that there is essentially no loss of efficiency. You can also specify any number of table requests in a single TABLES statement. To specify multiple table requests quickly, use a grouping syntax by placing parentheses around several

variables and joining other variables or variable combinations. For example, the statements shown in Table 35.8 illustrate grouping syntax.

Table 35.8 Grouping Syntax

Request	Equivalent to
tables A*(B C);	tables A*B A*C;
tables (A B)*(C D);	tables A*C B*C A*D B*D;
tables (A B C)*D;	tables A*D B*D C*D;
tables A – – C;	tables A B C;
tables (A – – C)*D;	tables A*D B*D C*D;

The TABLES statement variables are one or more variables from the DATA= input data set. These variables can be either character or numeric, but the procedure treats them as categorical variables. PROC FREQ uses the formatted values of the TABLES variable to determine the categorical variable levels. So if you assign a format to a variable with a FORMAT statement, PROC FREQ formats the values before dividing observations into the levels of a frequency or crosstabulation table. See the discussion of the FORMAT procedure in the *Base SAS Procedures Guide* and the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

If you use PROC FORMAT to create a user-written format that combines missing and nonmissing values into one category, PROC FREQ treats the entire category of formatted values as missing. See the discussion in the section "Grouping with Formats" on page 1725 for more information.

The frequency or crosstabulation table lists the values of both character and numeric variables in ascending order based on internal (unformatted) variable values unless you change the order with the ORDER= option. To list the values in ascending order by formatted value, use ORDER=FORMATTED in the PROC FREQ statement.

Without Options

If you request a one-way frequency table for a variable without specifying options, PROC FREQ produces frequencies, cumulative frequencies, percentages of the total frequency, and cumulative percentages for each value of the variable. If you request a two-way or an *n*-way crosstabulation table without specifying options, PROC FREQ produces crosstabulation tables that include cell frequencies, cell percentages of the total frequency, cell percentages of row frequencies, and cell percentages of column frequencies. The procedure excludes observations with missing values from the table but displays the total frequency of missing observations below each table.

Options

Table 35.9 lists the options available in the TABLES statement. Descriptions follow in alphabetical order.

Table 35.9 TABLES Statement Options

Option	Description
Control Statistical Analysis	
AGREE	requests tests and measures of classification agreement
ALL	requests tests and measures of association produced by CHISQ, MEASURES, and CMH
ALPHA=	sets the confidence level for confidence limits
BDT	requests Tarone's adjustment for the Breslow-Day test
BINOMIAL	requests binomial proportion, confidence limits, and tests for one-way tables
BINOMIALC	requests BINOMIAL statistics with a continuity correction
CHISQ	requests chi-square tests and measures based on chi-square
CL	requests confidence limits for the MEASURES statistics
CMH	requests all Cochran-Mantel-Haenszel statistics
CMH1	requests CMH correlation statistic, adjusted odds ratios, and adjusted relative risks
CMH2	requests CMH correlation and row mean scores (ANOVA) statistics, adjusted odds ratios, and adjusted relative risks
CONVERGE=	specifies convergence criterion for polychoric correlation
FISHER	requests Fisher's exact test for tables larger than 2×2
JT	requests Jonckheere-Terpstra test
MAXITER=	specifies maximum number of iterations for polychoric correlation
MEASURES	requests measures of association
MISSING	treats missing values as nonmissing
PLCORR	requests polychoric correlation
RELRISK	requests relative risk measures for 2×2 tables
RISKDIFF	requests risks and risk differences for 2×2 tables
RISKDIFFC	requests RISKDIFF statistics with a continuity correction
SCORES=	specifies the type of row and column scores
TESTF=	specifies expected frequencies for one-way chi-square test
TESTP=	specifies expected proportions for one-way chi-square test
TREND	requests Cochran-Armitage test for trend
Control Additional Table Information	
CELLCHI2	displays cell contributions to the Pearson chi-square statistic
CUMCOL	displays cumulative column percentages
DEVIATION	displays deviations of cell frequencies from expected values
EXPECTED	displays expected cell frequencies
MISSPRINT	displays missing value frequencies
SPARSE	includes all possible combinations of variable levels in LIST and OUT=
TOTPCT	displays percentages of total frequency for n-way tables ($n > 2$)

Table 35.9 *continued*

Option	Description
Control Displayed Output	
CONTENTS=	specifies the contents label for crosstabulation tables
CROSSLIST	displays crosstabulation tables in ODS column format
FORMAT=	formats the frequencies in crosstabulation tables
LIST	displays two-way to *n*-way tables in list format
NOCOL	suppresses display of column percentages
NOCUM	suppresses display of cumulative frequencies and percentages
NOFREQ	suppresses display of frequencies
NOPERCENT	suppresses display of percentages
NOPRINT	suppresses display of crosstabulation tables but displays statistics
NOROW	suppresses display of row percentages
NOSPARSE	suppresses zero frequency levels in CROSSLIST, LIST and OUT=
NOWARN	suppresses log warning message for the chi-square test
PRINTKWT	displays kappa coefficient weights
SCOROUT	displays row and column scores
Produce Statistical Graphics	
PLOTS=	requests plots from ODS Graphics
Create an Output Data Set	
OUT=	names an output data set to contain frequency counts
OUTCUM	includes cumulative frequencies and percentages in the output data set for one-way tables
OUTEXPECT	includes expected frequencies in the output data set
OUTPCT	includes row, column, and two-way table percentages in the output data set

You can specify the following options in a TABLES statement.

AGREE < (WT=FC) >

requests tests and measures of classification agreement for square tables. The AGREE option provides McNemar's test for 2×2 tables and Bowker's test of symmetry for square tables with more than two response categories. The AGREE option also produces the simple kappa coefficient, the weighted kappa coefficient, their asymptotic standard errors, and their confidence limits. When there are multiple strata, the AGREE option provides overall simple and weighted kappas as well as tests for equal kappas among strata. When there are multiple strata and two response categories, PROC FREQ computes Cochran's Q test. See the section "Tests and Measures of Agreement" on page 1765 for details about these statistics.

If you specify the WT=FC option in parentheses following the AGREE option, PROC FREQ uses Fleiss-Cohen weights to compute the weighted kappa coefficient. By default, PROC FREQ uses Cicchetti-Allison weights. See the section "Weighted Kappa Coefficient" on page 1767 for details. You can specify the PRINTKWT option to display the kappa coefficient weights.

AGREE statistics are computed only for square tables, where the number of rows equals the number of columns. If your table is not square due to observations with zero weights, you can specify the ZEROS option in the WEIGHT statement to include these observations. For more details, see the section "Tables with Zero Rows and Columns" on page 1770.

You can use the TEST statement to request asymptotic tests for the simple and weighted kappa coefficients. You can request exact *p*-values for the simple and weighted kappa coefficient tests, as well as for McNemar's test, by specifying the corresponding options in the EXACT statement. See the section "Exact Statistics" on page 1779 for more information.

ALL

requests all of the tests and measures that are computed by the CHISQ, MEASURES, and CMH options. The number of CMH statistics computed can be controlled by the CMH1 and CMH2 options.

ALPHA=α

specifies the level of confidence limits. The value of α must be between 0 and 1, and the default is 0.05. A confidence level of α produces $100(1-\alpha)\%$ confidence limits. The default of ALPHA=0.05 produces 95% confidence limits.

ALPHA= applies to confidence limits requested by TABLES statement options. There is a separate ALPHA= option in the EXACT statement that sets the level of confidence limits for Monte Carlo estimates of exact *p*-values, which are requested in the EXACT statement.

BDT

requests Tarone's adjustment in the Breslow-Day test for homogeneity of odds ratios. (You must specify the CMH option to compute the Breslow-Day test.) See the section "Breslow-Day Test for Homogeneity of the Odds Ratios" on page 1776 for more information.

BINOMIAL < (*binomial-options***) >**

requests the binomial proportion for one-way tables. When you specify the BINOMIAL option, PROC FREQ also provides the asymptotic standard error, asymptotic (Wald) and exact (Clopper-Pearson) confidence limits, and the asymptotic equality test for the binomial proportion by default.

You can specify *binomial-options* inside the parentheses following the BINOMIAL option. The LEVEL= binomial-option identifies the variable level for which to compute the proportion. If you do not specify LEVEL=, PROC FREQ computes the proportion for the first level that appears in the output. The P= binomial-option specifies the null proportion for the binomial tests. If you do not specify P=, PROC FREQ uses P=0.5 by default.

You can also specify *binomial-options* to request confidence limits and tests of noninferiority, superiority, and equivalence for the binomial proportion. Table 35.10 summarizes the *binomial-options*.

Available confidence limits for the binomial proportion include Agresti-Coull, exact (Clopper-Pearson), Jeffreys, Wald, and Wilson (score) confidence limits. You can specify more than one type of binomial confidence limits in the same analysis. If you do not specify any confidence limit requests with *binomial-options*, PROC FREQ computes Wald asymptotic confidence limits and exact (Clopper-Pearson) confidence limits by default. The

ALPHA= option determines the confidence level, and the default of ALPHA=0.05 produces 95% confidence limits for the binomial proportion.

To request exact tests for the binomial proportion, specify the BINOMIAL option in the EXACT statement. PROC FREQ then computes exact *p*-values for all binomial tests that you request with *binomial-options*, which can include tests of noninferiority, superiority, and equivalence, as well as the test of equality.

See the section "Binomial Proportion" on page 1745 for details.

Table 35.10 BINOMIAL Options

Task	*Binomial-Option*
Specify the variable level	LEVEL=
Specify the null proportion	P=
Request a continuity correction	CORRECT
Request confidence limits	AGRESTICOULL | AC ALL EXACT | CLOPPERPEARSON JEFFREYS | J WILSON | W WALD
Request tests	EQUIV | EQUIVALENCE NONINF | NONINFERIORITY SUP | SUPERIORITY
Specify the test margin	MARGIN=
Specify the test variance	VAR=SAMPLE | NULL

You can specify the following *binomial-options* inside parentheses following the BINOMIAL option.

AGRESTICOULL | AC
requests Agresti-Coull confidence limits for the binomial proportion. See the section "Agresti-Coull Confidence Limits" on page 1746 for details.

ALL
requests all available types of confidence limits for the binomial proportion. These include the following: Agresti-Coull, exact (Clopper-Pearson), Jeffreys, Wald, and Wilson (score) confidence limits.

CORRECT
includes a continuity correction in the asymptotic Wald confidence limits and tests. The CORRECT binomial-option has the same effect as the BINOMIALC option.

EQUIV | EQUIVALENCE
requests a test of equivalence for the binomial proportion. See the section "Equivalence Test" on page 1751 for details. You can specify the equivalence test margins, the

null proportion, and the variance type with the MARGIN=, P=, and VAR= binomial-options, respectively.

EXACT

CLOPPERPEARSON

requests exact (Clopper-Pearson) confidence limits for the binomial proportion. See the section "Exact (Clopper-Pearson) Confidence Limits" on page 1747 for details. If you do not request any binomial confidence limits by specifying binomial-options, PROC FREQ produces Wald and exact (Clopper-Pearson) confidence limits by default.

JEFFREYS | J

requests Jeffreys confidence limits for the binomial proportion. See the section "Jeffreys Confidence Limits" on page 1746 for details.

LEVEL=*level-number* | '*level-value*'

specifies the variable level for the binomial proportion. By default, PROC FREQ computes the proportion of observations for the first variable level that appears in the output. To request a different level, use LEVEL=*level-number* or LEVEL='*level-value*', where *level-number* is the variable level's number or order in the output, and *level-value* is the formatted value of the variable level. The value of *level-number* must be a positive integer. You must enclose *level-value* in single quotes.

MARGIN=*value* | **(***lower***,***upper***)**

specifies the margin for the noninferiority, superiority, and equivalence tests, which you request with the NONINF, SUP, and EQUIV binomial-options, respectively. If you do not specify MARGIN=, PROC FREQ uses a margin of 0.2 by default.

For noninferiority and superiority tests, specify a single *value* for the MARGIN= option. The MARGIN= *value* must be a positive number. You can specify *value* as a number between 0 and 1. Or you can specify *value* in percentage form as a number between 1 and 100, and PROC FREQ converts that number to a proportion. The procedure treats the value 1 as 1%.

For noninferiority and superiority tests, the test limits must be between 0 and 1. The limits are determined by the null proportion value (which you can specify with the P= binomial-option) and by the margin value. The noninferiority limit equals the null proportion minus the margin. By default, the null proportion equals 0.5 and the margin equals 0.2, which gives a noninferiority limit of 0.3. The superiority limit equals the null proportion plus the margin, which is 0.7 by default.

For an equivalence test, you can specify a single MARGIN= *value*, or you can specify both *lower* and *upper* values. If you specify a single MARGIN= *value*, it must be a positive number, as described previously. If you specify a single MARGIN= *value* for an equivalence test, PROC FREQ uses –*value* as the lower margin and *value* as the upper margin for the test. If you specify both *lower* and *upper* values for an equivalence test, you can specify them in proportion form as numbers between –1 or 1. Or you can specify them in percentage form as numbers between –100 and 100, and PROC FREQ converts the numbers to proportions. The value of *lower* must be less than the value of *upper*.

The equivalence limits must be between 0 and 1. The equivalence limits are determined by the null proportion value (which you can specify with the P= binomial-option) and by the margin values. The lower equivalence limit equals the null proportion plus the lower margin. By default, the null proportion equals 0.5 and the lower margin equals –0.2, which gives a lower equivalence limit of 0.3. The upper equivalence limit equals the null proportion plus the upper margin, which is 0.7 by default.

See the sections "Noninferiority Test" on page 1749 and "Equivalence Test" on page 1751 for details.

NONINF | NONINFERIORITY

requests a test of noninferiority for the binomial proportion. See the section "Noninferiority Test" on page 1749 for details. You can specify the noninferiority test margin, the null proportion, and the variance type with the MARGIN=, P=, and VAR= binomial-options, respectively.

P=*value*

specifies the null hypothesis proportion for the binomial tests. If you omit the P= option, PROC FREQ uses 0.5 as the null proportion. The null proportion *value* must be a positive number. You can specify *value* as a number between 0 and 1. Or you can specify *value* in percentage form as a number between 1 and 100, and PROC FREQ converts that number to a proportion. The procedure treats the value 1 as 1%.

SUP | SUPERIORITY

requests a test of superiority for the binomial proportion. See the section "Superiority Test" on page 1750 for details. You can specify the superiority test margin, the null proportion, and the variance type with the MARGIN=, P=, and VAR= binomial-options, respectively.

VAR=SAMPLE | NULL

specifies the type of variance estimate to use for the tests of noninferiority, superiority, and equivalence. The default is VAR=SAMPLE, which estimates the variance from the sample proportion. VAR=NULL uses the null proportion to compute the variance. See the sections "Noninferiority Test" on page 1749 and "Equivalence Test" on page 1751 for details.

WALD

requests Wald confidence limits for the binomial proportion. See the section "Wald Confidence Limits" on page 1746 for details. If you do not request any binomial confidence limits by specifying binomial-options, PROC FREQ produces Wald and exact (Clopper-Pearson) confidence limits by default.

WILSON | W

SCORE

requests Wilson confidence limits for the binomial proportion. These are also known as *score* confidence limits. See the section "Wilson (Score) Confidence Limits" on page 1747 for details.

BINOMIALC < (*binomial-options*) >

requests the BINOMIAL option statistics for one-way tables, and includes a continuity correction in the asymptotic Wald confidence limits and tests. The BINOMIAL option statistics include the binomial proportion, its asymptotic standard error, asymptotic (Wald) and exact (Clopper-Pearson) confidence limits, and the asymptotic equality test for the binomial proportion by default. The *binomial-options* available with the BINOMIALC option are the same as those available with BINOMIAL. See the description of the BINOMIAL option for details.

CELLCHI2

displays each crosstabulation table cell's contribution to the total Pearson chi-square statistic. The cell contribution is computed as

$$\frac{(frequency - expected)^2}{expected}$$

where *frequency* is the table cell frequency or count and *expected* is the expected cell frequency, which is computed under the null hypothesis that the row and column variables are independent. See the section "Pearson Chi-Square Test for Two-Way Tables" on page 1732 for details.

The CELLCHI2 option has no effect for one-way tables or for tables that are displayed with the LIST option.

CHISQ

requests chi-square tests of homogeneity or independence and measures of association based on the chi-square statistic. The tests include the Pearson chi-square, likelihood-ratio chi-square, and Mantel-Haenszel chi-square. The measures include the phi coefficient, the contingency coefficient, and Cramer's V. For 2×2 tables, the CHISQ option also provides Fisher's exact test and the continuity-adjusted chi-square. See the section "Chi-Square Tests and Statistics" on page 1731 for details.

For one-way tables, the CHISQ option provides a chi-square goodness-of-fit test for equal proportions. If you specify the null hypothesis proportions with the TESTP= option, PROC FREQ computes a chi-square goodness-of-fit test for the specified proportions. If you specify null hypothesis frequencies with the TESTF= option, PROC FREQ computes a chi-square goodness-of-fit test for the specified frequencies. See the section "Chi-Square Test for One-Way Tables" on page 1732 for more information.

To request Fisher's exact test for tables larger than 2×2, use the FISHER option in the EXACT statement. Exact tests are also available for other CHISQ statistics, including the Pearson, likelihood-ratio, and Mantel-Haenszel chi-square, and the chi-square goodness-of-fit test for one-way tables. You can use the EXACT statement to request these tests. See the section "Exact Statistics" on page 1779 for details.

CL

requests confidence limits for the MEASURES statistics. If you omit the MEASURES option, the CL option invokes MEASURES. You can set the level of the confidence limits by using the ALPHA= option. The default of ALPHA=0.5 produces 95% confidence limits. See the sections "Measures of Association" on page 1736 and "Confidence Limits" on page 1737 for more information.

CMH

requests Cochran-Mantel-Haenszel statistics, which test for association between the row and column variables after adjusting for the remaining variables in a multiway table. The Cochran-Mantel-Haenszel statistics include the nonzero correlation statistic, the row mean scores differ (ANOVA) statistic, and the general association statistic. In addition, for 2×2 tables, the CMH option provides the adjusted Mantel-Haenszel and logit estimates of the odds ratio and relative risks, together with their confidence limits. For stratified 2×2 tables, the CMH option provides the Breslow-Day test for homogeneity of odds ratios. (To request Tarone's adjustment for the Breslow-Day test, specify the BDT option.) See the section "Cochran-Mantel-Haenszel Statistics" on page 1770 for details.

You can use the CMH1 or CMH2 option to control the number of CMH statistics that PROC FREQ computes.

For stratified 2×2 tables, you can request Zelen's exact test for equal odds ratios by specifying the EQOR option in the EXACT statement. See the section "Zelen's Exact Test for Equal Odds Ratios" on page 1776 for details. You can request exact confidence limits for the common odds ratio by specifying the COMOR option in the EXACT statement. This option also provides a common odds ratio test. See the section "Exact Confidence Limits for the Common Odds Ratio" on page 1777 for details.

CMH1

requests the Cochran-Mantel-Haenszel correlation statistic. It does not provide the CMH row mean scores differ (ANOVA) statistic or the general association statistic, which are provided by the CMH option. For tables larger than 2×2, the CMH1 option requires less memory than the CMH option, which can require an enormous amount of memory for large tables.

For 2×2 tables, the CMH1 option also provides the adjusted Mantel-Haenszel and logit estimates of the common odds ratio and relative risks and the Breslow-Day test.

CMH2

requests the Cochran-Mantel-Haenszel correlation statistic and the row mean scores (ANOVA) statistic. It does not provide the CMH general association statistic, which is provided by the CMH option. For tables larger than 2×2, the CMH2 option requires less memory than the CMH option, which can require an enormous amount of memory for large tables.

For 2×2 tables, the CMH2 option also provides the adjusted Mantel-Haenszel and logit estimates of the common odds ratio and relative risks and the Breslow-Day test.

CONTENTS=*'string'*

specifies the label to use for crosstabulation tables in the contents file, the Results window, and the trace record. For information about output presentation, see the *SAS Output Delivery System: User's Guide*.

If you omit the CONTENTS= option, the contents label for crosstabulation tables is "Cross-Tabular Freq Table" by default.

Note that contents labels for all crosstabulation tables that are produced by a single TABLES statement use the same text. To specify different contents labels for different crosstabulation

tables, request the tables in separate TABLES statements and use the CONTENTS= option in each TABLES statement.

To remove the crosstabulation table entry from the contents file, you can specify a null label with CONTENTS=''.

The CONTENTS= option affects only contents labels for crosstabulation tables. It does not affect contents labels for other PROC FREQ tables.

To specify the contents label for any PROC FREQ table, you can use PROC TEMPLATE to create a customized table definition. The CONTENTS_LABEL attribute in the DEFINE TABLE statement of PROC TEMPLATE specifies the contents label for the table. See the chapter "The TEMPLATE Procedure" in the *SAS Output Delivery System: User's Guide* for more information.

CONVERGE=*value*

specifies the convergence criterion for computing the polychoric correlation, which you request with the PLCORR option. The CONVERGE= *value* must be a positive number. By default CONVERGE=0.0001. Iterative computation of the polychoric correlation stops when the convergence measure falls below the value of CONVERGE= or when the number of iterations exceeds the value specified in the MAXITER= option, whichever happens first. See the section "Polychoric Correlation" on page 1742 for details.

CROSSLIST

displays crosstabulation tables in ODS column format instead of the default crosstabulation cell format. In a CROSSLIST table display, the rows correspond to the crosstabulation table cells, and the columns correspond to descriptive statistics such as Frequency and Percent. The CROSSLIST table displays the same information as the default crosstabulation table, but uses an ODS column format instead of the table cell format. See the section "Multiway Tables" on page 1789 for details about the contents of the CROSSLIST table.

You can control the contents of a CROSSLIST table with the same options available for the default crosstabulation table. These include the NOFREQ, NOPERCENT, NOROW, and NOCOL options. You can request additional information in a CROSSLIST table with the CELLCHI2, DEVIATION, EXPECTED, MISSPRINT, and TOTPCT options.

The FORMAT= option and the CUMCOL option have no effect for CROSSLIST tables. You cannot specify both the LIST option and the CROSSLIST option in the same TABLES statement.

You can use the NOSPARSE option to suppress display of variable levels with zero frequency in CROSSLIST tables. By default for CROSSLIST tables, PROC FREQ displays all levels of the column variable within each level of the row variable, including any column variable levels with zero frequency for that row. And for multiway tables displayed with the CROSSLIST option, the procedure displays all levels of the row variable for each stratum of the table by default, including any row variable levels with zero frequency for the stratum.

CUMCOL

displays the cumulative column percentages in the cells of the crosstabulation table. The CUMCOL option does not apply to crosstabulation tables produced with the LIST or CROSSLIST option.

DEVIATION

displays the deviation of the frequency from the expected frequency for each cell of the crosstabulation table. See the section "Pearson Chi-Square Test for Two-Way Tables" on page 1732 for details. The DEVIATION option does not apply to crosstabulation tables produced with the LIST option.

EXPECTED

displays the expected cell frequencies under the hypothesis of independence (or homogeneity) for crosstabulation tables. See the section "Pearson Chi-Square Test for Two-Way Tables" on page 1732 for details. The EXPECTED option does not apply to tables produced with the LIST option.

FISHER | EXACT

requests Fisher's exact test for tables that are larger than 2×2. (For 2×2 tables, the CHISQ option provides Fisher's exact test.) This test is also known as the Freeman-Halton test. See the sections "Fisher's Exact Test" on page 1734 and "Exact Statistics" on page 1779 for more information.

If you omit the CHISQ option in the TABLES statement, the FISHER option invokes CHISQ. You can also request Fisher's exact test by specifying the FISHER option in the EXACT statement.

CAUTION: PROC FREQ computes exact tests with fast and efficient algorithms that are superior to direct enumeration. Exact tests are appropriate when a data set is small, sparse, skewed, or heavily tied. For some large problems, computation of exact tests might require a considerable amount of time and memory. Consider using asymptotic tests for such problems. Alternatively, when asymptotic methods might not be sufficient for such large problems, consider using Monte Carlo estimation of exact *p*-values. See the section "Computational Resources" on page 1781 for more information.

FORMAT=*format-name*

specifies a format for the following crosstabulation table cell values: frequency, expected frequency, and deviation. PROC FREQ also uses the specified format to display the row and column total frequencies and the overall total frequency in crosstabulation tables.

You can specify any standard SAS numeric format or a numeric format defined with the FORMAT procedure. The format length must not exceed 24. If you omit the FORMAT= option, by default PROC FREQ uses the BEST6. format to display frequencies less than 1E6, and the BEST7. format otherwise.

The FORMAT= option applies only to crosstabulation tables displayed in the default format. It does not apply to crosstabulation tables produced with the LIST or CROSSLIST option.

To change display formats in any FREQ table, you can use PROC TEMPLATE. See the chapter "The TEMPLATE Procedure" in the *SAS Output Delivery System: User's Guide* for more information.

JT

requests the Jonckheere-Terpstra test. See the section "Jonckheere-Terpstra Test" on page 1763 for details.

LIST

displays two-way to n-way crosstabulation tables in a list format instead of the default crosstabulation cell format. The LIST option displays the entire multiway table in one table, instead of displaying a separate two-way table for each stratum. See the section "Multiway Tables" on page 1789 for details.

The LIST option is not available when you also specify statistical options. You must use the standard crosstabulation table display or the CROSSLIST display when you request statistical tests or measures.

MAXITER=*number*

specifies the maximum number of iterations for computing the polychoric correlation, which you request with the PLCORR option. The value of the MAXITER= option must be a positive integer. By default MAXITER=20. Iterative computation of the polychoric correlation stops when the number of iterations exceeds the MAXITER= value or when the convergence measures falls below the value of the CONVERGE= option, whichever happens first. See the section "Polychoric Correlation" on page 1742 for details.

MEASURES

requests several measures of association and their asymptotic standard errors. The MEASURES option provides the following statistics: gamma, Kendall's tau-b, Stuart's tau-c, Somers' $D(C|R)$, Somers' $D(R|C)$, the Pearson and Spearman correlation coefficients, lambda (symmetric and asymmetric), and uncertainty coefficients (symmetric and asymmetric). To request confidence limits for these measures of association, you can specify the CL option.

For 2×2 tables, the MEASURES option also provides the odds ratio, column 1 relative risk, column 2 relative risk, and the corresponding confidence limits. Alternatively, you can obtain the odds ratio and relative risks, without the other measures of association, by specifying the RELRISK option.

See the section "Measures of Association" on page 1736 for details.

You can use the TEST statement to request asymptotic tests for the following measures of association: gamma, Kendall's tau-b, Stuart's tau-c, Somers' $D(C|R)$, Somers' $D(R|C)$, and the Pearson and Spearman correlation coefficients. You can use the EXACT statement to request exact tests for the Pearson and Spearman correlation coefficients and exact confidence limits for the odds ratio. See the section "Exact Statistics" on page 1779 for more information.

MISSING

treats missing values as a valid nonmissing level for all TABLES variables. The MISSING option displays the missing levels in frequency and crosstabulation tables and includes them in all calculations of percentages, tests, and measures.

By default, if you do not specify the MISSING or MISSPRINT option, an observation is excluded from a table if it has a missing value for any of the variables in the TABLES request. When PROC FREQ excludes observations with missing values, it displays the total frequency of missing observations below the table. See the section "Missing Values" on page 1726 for more information.

MISSPRINT
displays missing value frequencies in frequency and crosstabulation tables but does not include the missing value frequencies in any computations of percentages, tests, or measures.

By default, if you do not specify the MISSING or MISSPRINT option, an observation is excluded from a table if it has a missing value for any of the variables in the TABLES request. When PROC FREQ excludes observations with missing values, it displays the total frequency of missing observations below the table. See the section "Missing Values" on page 1726 for more information.

NOCOL
suppresses the display of column percentages in crosstabulation table cells.

NOCUM
suppresses the display of cumulative frequencies and percentages in one-way frequency tables. The NOCUM option also suppresses the display of cumulative frequencies and percentages in crosstabulation tables in list format, which you request with the LIST option.

NOFREQ
suppresses the display of cell frequencies in crosstabulation tables. The NOFREQ option also suppresses row total frequencies. This option has no effect for one-way tables or for crosstabulation tables in list format, which you request with the LIST option.

NOPERCENT
suppresses the display of overall percentages in crosstabulation tables. These percentages include the cell percentages of the total (two-way) table frequency, as well as the row and column percentages of the total table frequency. To suppress the display of cell percentages of row or column totals, use the NOROW or NOCOL option, respectively.

For one-way frequency tables and crosstabulation tables in list format, the NOPERCENT option suppresses the display of percentages and cumulative percentages.

NOPRINT
suppresses the display of frequency and crosstabulation tables but displays all requested tests and statistics. To suppress the display of all output, including tests and statistics, use the NOPRINT option in the PROC FREQ statement.

NOROW
suppresses the display of row percentages in crosstabulation table cells.

NOSPARSE
suppresses the display of cells with a zero frequency count in LIST output and omits them from the OUT= data set. The NOSPARSE option applies when you specify the ZEROS option in the WEIGHT statement to include observations with zero weights. By default, the ZEROS option invokes the SPARSE option, which displays table cells with a zero frequency count in the LIST output and includes them in the OUT= data set. See the description of the ZEROS option for more information.

The NOSPARSE option also suppresses the display of variable levels with zero frequency in CROSSLIST tables. By default for CROSSLIST tables, PROC FREQ displays all levels

of the column variable within each level of the row variable, including any column variable levels with zero frequency for that row. For multiway tables displayed with the CROSSLIST option, the procedure displays all levels of the row variable for each stratum of the table by default, including any row variable levels with zero frequency for the stratum.

NOWARN

suppresses the log warning message that the asymptotic chi-square test might not be valid. By default, PROC FREQ displays this log message when you request the CHISQ option and more than 20 percent of the table cells have expected frequencies less than five.

OUT=*SAS-data-set*

names an output data set that contains frequency or crosstabulation table counts and percentages. If more than one table request appears in the TABLES statement, the contents of the OUT= data set correspond to the last table request in the TABLES statement. The OUT= data set variable COUNT contains the frequencies and the variable PERCENT contains the percentages. See the section "Output Data Sets" on page 1784 for details. You can specify the following options to include additional information in the OUT= data set: OUTCUM, OUTEXPECT, and OUTPCT.

OUTCUM

includes cumulative frequencies and cumulative percentages in the OUT= data set for one-way tables. The variable CUM_FREQ contains the cumulative frequencies, and the variable CUM_PCT contains the cumulative percentages. See the section "Output Data Sets" on page 1784 for details. The OUTCUM option has no effect for two-way or multiway tables.

OUTEXPECT

includes expected cell frequencies in the OUT= data set for crosstabulation tables. The variable EXPECTED contains the expected cell frequencies. See the section "Output Data Sets" on page 1784 for details. The EXPECTED option has no effect for one-way tables.

OUTPCT

includes the following additional variables in the OUT= data set for crosstabulation tables:

PCT_COL	percentage of column frequency
PCT_ROW	percentage of row frequency
PCT_TABL	percentage of stratum (two-way table) frequency, for n-way tables where $n > 2$

See the section "Output Data Sets" on page 1784 for details. The OUTPCT option has no effect for one-way tables.

PLCORR

requests the polychoric correlation coefficient. For 2×2 tables, this statistic is more commonly known as the tetrachoric correlation coefficient, and it is labeled as such in the displayed output. See the section "Polychoric Correlation" on page 1742 for details. Also see the descriptions of the CONVERGE= and MAXITER= options, which you can specify to control the iterative computation of the polychoric correlation coefficient.

If you omit the MEASURES option, the PLCORR option invokes MEASURES.

PLOTS < (*global-plot-options*) > < = *plot-request* < (*plot-options*) > >
PLOTS < (*global-plot-options*) > < = (*plot-request* < (*plot-options*) > < ... *plot-request* < (*plot-options*) > >)>

requests plots for PROC FREQ to produce by using ODS Graphics. When you specify only one *plot-request*, you can omit the parentheses around the request. For example:

```
plots=all
plots=freqplot
plots=(freqplot oddsratioplot)
plots(only)=(cumfreqplot deviationplot)
```

For information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." You must enable ODS Graphics before requesting plots, as shown in the following statements:

```
ods graphics on;
proc freq;
   tables treatment*response / chisq plots=freqplot;
   weight wt;
run;
ods graphics off;
```

The PLOTS= option has no effect when you specify the NOPRINT option in the PROC FREQ statement.

If you do not specify the PLOTS= option but have enabled ODS Graphics, then PROC FREQ produces all plots associated with the analyses you request in the current TABLES statement.

Table 35.11 lists the available *plot-requests*, together with their *plot-options* and required TABLES statement options.

Table 35.11 PLOTS= Options

Plot-Request	*Plot-Options*	Required TABLES Statement Option
CUMFREQPLOT	ORIENT= SCALE= TYPE=	one-way table request
DEVIATIONPLOT	NOSTATS ORIENT= TYPE=	CHISQ (one-way table)
FREQPLOT	ORIENT= SCALE= TYPE=	any table request
FREQPLOT	TWOWAY=	two-way or multiway table
KAPPAPLOT	NPANELPOS= ORDER= RANGE= STATS	AGREE ($h \times r \times r$ table)
ODDSRATIOPLOT	LOGBASE= NPANELPOS= ORDER= RANGE= STATS	MEASURES or RELRISK ($h \times 2 \times 2$ table)
WTKAPPAPLOT	NPANELPOS= ORDER= RANGE= STATS	AGREE ($h \times r \times r$ table, $r > 2$)

Global-Plot-Options

A *global-plot-option* applies to all plots for which the option is available, unless it is altered by a specific *plot-option*. You can specify the following *global-plot-options* in parentheses following the PLOTS option:

NPANELPOS=*n*

applies to ODDSRATIOPLOT, KAPPAPLOT, and WTKAPPAPLOT. The NPANELPOS= plot-option breaks the plot into multiple graphics that have at most $|n|$ odds ratios or kappa statistics per graphic. If n is positive, the number of statistics per graphic is balanced; but if n is negative, the number of statistics per graphic is not balanced. By default, $n = 0$ and all statistics are displayed in a single plot. For example, suppose you want to display 21 odds ratios. Then NPANELPOS=20 displays two plots, the first with 11 odds ratios and the second with 10; NPANELPOS=–20 displays 20 odds ratios in the first plot but only 1 in the second.

ONLY

suppresses the default plots and requests only the plots specified as *plot-requests*.

ORDER=ASCENDING | DESCENDING
applies to ODDSRATIOPLOT, KAPPPAPLOT, and WTKAPPAPLOT. The ORDER= plot-option displays the odds ratios or kappa statistics in sorted order. By default, the statistics are displayed in the order that the corresponding strata appear in the multiway table display.

ORIENT=HORIZONTAL | VERTICAL
applies to FREQPLOT, CUMFREQPLOT, and DEVIATIONPLOT. The ORIENT= plot-option controls the orientation of the plot. ORIENT=HORIZONTAL places the variable levels on the *y*-axis and the frequencies or statistic-values on the *x*-axis. ORIENT=VERTICAL places the variable levels on the *x*-axis. The default orientation is ORIENT=VERTICAL for bar charts and ORIENT=HORIZONTAL for dot plots.

RANGE=(< *min* >< ,*max* >) | CLIP
applies to ODDSRATIOPLOT, KAPPAPLOT, and WTKAPPAPLOT. The RANGE= plot-option specifies the range of values to display. If you specify RANGE=CLIP, the confidence intervals are clipped and the display range is determined by the minimum and maximum values of the estimates. By default, the display range includes all confidence limits.

SCALE=FREQ | LOG | PERCENT| SQRT
applies to FREQPLOT and CUMFREQPLOT. The SCALE= plot-option specifies the scale of the frequencies to display. The default is SCALE=FREQ, which displays unscaled frequencies. SCALE=LOG displays log (base 10) frequencies. SCALE=PERCENT displays percentages or relative frequencies. SCALE=SQRT displays square roots of the frequencies, which produces a plot known as a *rootogram*.

STATS
applies to ODDSRATIOPLOT, KAPPAPLOT, and WTKAPPAPLOT. The STATS plot-option displays the values of the statistics and their confidence limits on the right side of the plot. If you do not request the STATS option, the statistic values are not displayed.

TYPE=BARCHART | DOTPLOT
applies to FREQPLOT, CUMFREQPLOT, and DEVIATIONPLOT. The TYPE= plot-option specifies the plot type. TYPE=BARCHART produces a bar chart, and TYPE=DOTPLOT produces a dot plot. The default is TYPE=BARCHART.

Plot-Requests

The following *plot-requests* are available with the PLOTS= option:

ALL
requests all plots associated with the specified analyses. This is the default if you do not specify the PLOTS(ONLY) option.

CUMFREQPLOT < (*plot-options*) >
requests a plot of cumulative frequencies for a one-way frequency table. The following *plot-options* are available for CUMFREQPLOT: ORIENT=, SCALE=, and TYPE=.

DEVIATIONPLOT < (*plot-options*) >
requests a plot of relative deviations from expected frequencies for a one-way table. The DEVIATIONPLOT is associated with the CHISQ option for a one-way table request. The following *plot-options* are available for DEVIATIONPLOT: ORIENT= and TYPE=.

FREQPLOT < (*plot-options*) >
requests a frequency plot. Frequency plots are available for frequency and crosstabulation tables. For multiway tables, PROC FREQ provides a two-way frequency plot for each stratum. The following *plot-options* are available for FREQPLOT for all tables: ORIENT=, SCALE=, and TYPE=. For two-way and multiway tables, you can use the TWOWAY= plot-option to specify the layout of the two-way frequency plot.

KAPPAPLOT < (*plot-options*) >
requests a plot of kappa statistics and confidence limits for a multiway table. The KAPPAPLOT is associated with the AGREE option for multiway square tables. The following *plot-options* are available for KAPPAPLOT: NPANELPOS=, ORDER=, RANGE=, and STATS.

NONE
suppresses all plots.

ODDSRATIOPLOT < (*plot-options*) >
requests a plot of odds ratios and confidence limits for a multiway table. The ODDSRATIOPLOT is associated with the MEASURES or RELRISK option for multiway 2×2 tables. The following *plot-options* are available for ODDSRATIOPLOT: LOGBASE=, NPANELPOS=, ORDER=, RANGE=, and STATS.

WTKAPPAPLOT < (*plot-options*) >
requests a plot of weighted kappa statistics and confidence limits for a multiway table. The WTKAPPAPLOT is associated with the AGREE option for multiway square tables with more than two rows. (For 2×2 tables, the simple kappa and weighted kappa statistics are the same, so weighted kappas are not presented for 2×2 tables.) The following *plot-options* are available for WTKAPPAPLOT: NPANELPOS=, ORDER=, RANGE=, and STATS.

Plot-Options

You can specify the following *plot-options* in parentheses after a *plot-request*.

LOGBASE=2 | E | 10
applies only to ODDSRATIOPLOT. The LOGBASE= plot-option displays the odds ratio axis on the specified log scale.

NOSTATS
applies only to DEVIATIONPLOT. The NOSTATS plot-option suppresses the chi-square *p*-value that is displayed by default in the deviation plot.

NPANELPOS=*n*

applies to ODDSRATIOPLOT, KAPPAPLOT, and WTKAPPAPLOT. The NPANELPOS= plot-option breaks the plot into multiple graphics that have at most $|n|$ odds ratios or kappa statistics per graphic. If n is positive, the number of statistics per graphic is balanced; but if n is negative, the number of statistics per graphic is not balanced. By default, $n = 0$ and all statistics are displayed in a single plot. For example, suppose you want to display 21 odds ratios. Then NPANELPOS=20 displays two plots, the first with 11 odds ratios and the second with 10; NPANELPOS=–20 displays 20 odds ratios in the first plot but only 1 in the second.

ORDER=ASCENDING | DESCENDING

applies to ODDSRATIOPLOT, KAPPPAPLOT, and WTKAPPAPLOT. The ORDER= plot-option displays the odds ratios or kappa statistics in sorted order. By default, the statistics are displayed in the order that the corresponding strata appear in the multiway table display.

ORIENT=HORIZONTAL | VERTICAL

applies to FREQPLOT, CUMFREQPLOT, and DEVIATIONPLOT. The ORIENT= plot-option controls the orientation of the plot. ORIENT=HORIZONTAL places the variable levels on the *y*-axis and the frequencies or statistic-values on the *x*-axis. ORIENT=VERTICAL places the variable levels on the *x*-axis. The default orientation is ORIENT=VERTICAL for bar charts and ORIENT=HORIZONTAL for dot plots.

RANGE=(< *min* >< ,*max* >) | CLIP

applies to ODDSRATIOPLOT, KAPPAPLOT, and WTKAPPAPLOT. The RANGE= plot-option specifies the range of values to display. If you specify RANGE=CLIP, the confidence intervals are clipped and the display range is determined by the minimum and maximum values of the estimates. By default, the display range includes all confidence limits.

SCALE=FREQ | LOG | PERCENT| SQRT

applies to FREQPLOT and CUMFREQPLOT. The SCALE= option specifies the scale of the frequencies to display. The default is SCALE=FREQ, which displays unscaled frequencies. SCALE=LOG displays log (base 10) frequencies. SCALE=PERCENT displays percentages or relative frequencies. SCALE=SQRT displays square roots of the frequencies, which produces a plot known as a *rootogram*.

STATS

applies to ODDSRATIOPLOT, KAPPAPLOT, and WTKAPPAPLOT. The STATS plot-option displays the values of the statistics and their confidence limits on the right side of the plot. If you do not request the STATS option, the statistic values are not displayed.

TWOWAY=GROUPVERTICAL | GROUPHORIZONTAL | STACKED

applies to FREQPLOT for two-way and multiway tables. For multiway tables, PROC FREQ provides a two-way frequency plot for each stratum. The TWOWAY= plot-option specifies the layout for two-way frequency plots. The default is TWOWAY=GROUPVERTICAL, which produces a grouped plot with a vertical common baseline. The plots are grouped by the row variable, which is the first variable you specify in a two-way table request. TWOWAY=GROUPHORIZONTAL produces a grouped plot with a horizontal common baseline.

TWOWAY=STACKED produces stacked frequency plots for two-way tables. In a stacked bar chart, the bars correspond to the column variable values, and the row frequencies are stacked within each column. For dot plots, the dotted lines correspond to the columns, and the row frequencies within columns are plotted as data dots on the same column line.

The TYPE= and ORIENT= plot-options are available for each TWOWAY= layout option.

TYPE=BARCHART | DOTPLOT
applies to FREQPLOT, CUMFREQPLOT, and DEVIATIONPLOT. The TYPE= plot-option specifies the plot type. TYPE=BARCHART produces a bar chart, and TYPE=DOTPLOT produces a dot plot. The default is TYPE=BARCHART.

PRINTKWT
displays the weights that PROC FREQ uses to compute the weighted kappa coefficient. You must also specify the AGREE option to request the weighted kappa coefficient. You can specify (WT=FC) with the AGREE option to request Fleiss-Cohen weights. By default, PROC FREQ uses Cicchetti-Allison weights to compute the weighted kappa coefficient. See the section "Weighted Kappa Coefficient" on page 1767 for details.

RELRISK
requests relative risk measures and their confidence limits for 2×2 tables. These measures include the odds ratio and the column 1 and 2 relative risks. See the section "Odds Ratio and Relative Risks for 2 x 2 Tables" on page 1760 for details.

You can also obtain the RELRISK measures by specifying the MEASURES option, which produces other measures of association in addition to the relative risks.

You can request exact confidence limits for the odds ratio by specifying the OR option in the EXACT statement.

RISKDIFF < (*riskdiff-options*) >
requests risks, or binomial proportions, for 2×2 tables. For column 1 and column 2, PROC FREQ computes the row 1 risk, row 2 risk, total risk, and risk difference (row 1 – row 2), together with their asymptotic standard errors and asymptotic (Wald) confidence limits. PROC FREQ also provides exact (Clopper-Pearson) confidence limits for the row 1, row 2, and total risks. The ALPHA= option determines the confidence level, and the default of ALPHA=0.05 produces 95% confidence limits. See the section "Risks and Risk Differences" on page 1752 for details.

You can specify *riskdiff-options* inside the parentheses following the RISKDIFF option to request tests of noninferiority, superiority, and equivalence for the risk difference. Available test methods include Farrington-Manning, Hauck-Anderson, and Newcombe score (Wilson), in addition to the Wald test. Table 35.12 summarizes the *riskdiff-options*.

You can request exact unconditional confidence limits for the risk difference by specifying the RISKDIFF option in the EXACT statement. See the section "Exact Unconditional Confidence Limits for the Risk Difference" on page 1759 for more information.

Table 35.12 RISKDIFF (Proportion Difference) Options

Task	*Riskdiff-Option*
Specify the column	COLUMN=1 | 2
Request a continuity correction	CORRECT
Request tests	EQUAL EQUIV | EQUIVALENCE NONINF | NONINFERIORITY SUP | SUPERIORITY
Specify the test method	METHOD=
Specify the test margin	MARGIN=
Specify the test variance	VAR=SAMPLE | NULL

You can specify the following *riskdiff-options* inside parentheses following the RISKDIFF option.

COLUMN=1 | 2 | BOTH
specifies the table column for which to compute the risk difference tests of noninferiority, superiority, or equivalence, which you request with the NONINF, SUP, and EQUIV riskdiff-options, respectively. You can specify COLUMN=1, COLUMN=2, or COLUMN=BOTH. If you do not specify the COLUMN= option, PROC FREQ computes the risk difference tests for column 1. The COLUMN= option has no effect on the table of risk estimates and confidence limits or on the equality test; PROC FREQ computes these statistics for both column 1 and column 2.

CORRECT
includes a continuity correction in the asymptotic Wald confidence limits and tests. The CORRECT riskdiff-option also includes a continuity correction in the Newcombe score confidence limits, which you request with the METHOD=SCORE riskdiff-option. METHOD=HA and METHOD=FM do not use continuity corrections. The CORRECT riskdiff-option has the same effect as the RISKDIFFC option.

EQUAL
requests a test of the null hypothesis that the risk difference equals zero. PROC FREQ provides an asymptotic Wald test of equality. See the section "Equality Test" on page 1755 for details. You can specify the test variance type with the VAR= riskdiff-option.

EQUIV | EQUIVALENCE
requests a test of equivalence for the risk difference. See the section "Equivalence Tests" on page 1758 for details. You can specify the equivalence test margins with the MARGIN= riskdiff-option and the test method with the METHOD= riskdiff-option. PROC FREQ uses METHOD=WALD by default.

MARGIN=*value* **|** **(***lower***,***upper***)**
specifies the margin for the noninferiority, superiority, and equivalence tests, which you request with the NONINF, SUP, and EQUIV riskdiff-options, respectively. If you do not specify MARGIN=, PROC FREQ uses a margin of 0.2 by default.

For noninferiority and superiority tests, specify a single *value* for the MARGIN= riskdiff-option. The MARGIN= *value* must be a positive number. You can specify *value* as a number between 0 and 1. Or you can specify *value* in percentage form as a number between 1 and 100, and PROC FREQ converts that number to a proportion. The procedure treats the value 1 as 1%.

For an equivalence test, you can specify a single MARGIN= *value*, or you can specify both *lower* and *upper* values. If you specify a single MARGIN= *value*, it must be a positive number, as described previously. If you specify a single MARGIN= *value* for an equivalence test, PROC FREQ uses –*value* as the lower margin and *value* as the upper margin for the test. If you specify both *lower* and *upper* values for an equivalence test, you can specify them in proportion form as numbers between –1 or 1. Or you can specify them in percentage form as numbers between –100 and 100, and PROC FREQ converts the numbers to proportions. The value of *lower* must be less than the value of *upper*.

METHOD=*method*

specifies the method for the noninferiority, superiority, and equivalence tests, which you request with the NONINF, SUP, and EQUIV riskdiff-options, respectively. The following methods are available:

FM	Farrington-Manning
HA	Hauck-Anderson
SCORE | NEWCOMBE | WILSON	Newcombe score (Wilson)
WALD	Wald

The default is METHOD=WALD. See the section "Noninferiority Test" on page 1755 for descriptions of these methods.

For METHOD=SCORE and METHOD=WALD, you can request a continuity correction with the CORRECT riskdiff-option. For METHOD=WALD, you can specify the variance type with the VAR= riskdiff-option.

NONINF | NONINFERIORITY

requests a test of noninferiority for the risk difference. See the section "Noninferiority Test" on page 1755 for details. You can specify the test margin with the MARGIN= riskdiff-option and the test method with the METHOD= riskdiff-option. PROC FREQ uses METHOD=WALD by default.

SUP | SUPERIORITY

requests a test of superiority for the binomial proportion. See the section "Superiority Test" on page 1758 for details. You can specify the test margin with the MARGIN= riskdiff-option and the test method with the METHOD= riskdiff-option. PROC FREQ uses METHOD=WALD by default.

VAR=SAMPLE | NULL

specifies the type of variance estimate to use for the Wald tests of noninferiority, superiority, equivalence, and equality. The default is VAR=SAMPLE, which estimates

the variance from the sample proportions. VAR=NULL uses the null hypothesis values to compute the variance. See the sections "Equality Test" on page 1755 and "Noninferiority Test" on page 1755 for details.

RISKDIFFC <(*riskdiff-options*)>

requests the RISKDIFF option statistics for 2×2 tables and includes a continuity correction in the asymptotic Wald confidence limits and tests.

The RISKDIFF option statistics include risks, or binomial proportions, for 2×2 tables. For column 1 and column 2, PROC FREQ computes the row 1 risk, row 2 risk, total risk, and risk difference (row 1 – row 2), together with their asymptotic standard errors and asymptotic (Wald) confidence limits. PROC FREQ also provides exact (Clopper-Pearson) confidence limits for the row 1, row 2, and total risks. See the section "Risks and Risk Differences" on page 1752 for details.

You can request additional tests and statistics for the risk difference by specifying *riskdiff-options* in parentheses after RISKDIFFC. The *riskdiff-options* are the same as those available with RISKDIFF. See the description of the RISKDIFF option for details.

You can request exact unconditional confidence limits for the risk difference by specifying the RISKDIFF option in the EXACT statement. See the section "Exact Unconditional Confidence Limits for the Risk Difference" on page 1759 for more information.

SCORES=*type*

specifies the type of row and column scores that PROC FREQ uses to compute the following statistics: Mantel-Haenszel chi-square, Pearson correlation, Cochran-Armitage test for trend, weighted kappa coefficient, and Cochran-Mantel-Haenszel statistics. The value of *type* can be one of the following:

- MODRIDIT
- RANK
- RIDIT
- TABLE

See the section "Scores" on page 1730 for descriptions of these score types.

If you do not specify the SCORES= option, PROC FREQ uses SCORES=TABLE by default. For character variables, the row and column TABLE scores are the row and column numbers. That is, the TABLE score is 1 for row 1, 2 for row 2, and so on. For numeric variables, the row and column TABLE scores equal the variable values. See the section "Scores" on page 1730 for details. Using MODRIDIT, RANK, or RIDIT scores yields nonparametric analyses.

You can use the SCOROUT option to display the row and column scores.

SCOROUT

displays the row and column scores that PROC FREQ uses to compute score-based tests and statistics. You can specify the score type with the SCORES= option. See the section "Scores" on page 1730 for details.

The scores are computed and displayed only when PROC FREQ computes statistics for two-way tables. You can use ODS to store the scores in an output data set. See the section "ODS Table Names" on page 1795 for more information.

SPARSE

reports all possible combinations of the variable values for an n-way table when $n > 1$, even if a combination does not occur in the data. The SPARSE option applies only to crosstabulation tables displayed in LIST format and to the OUT= output data set. If you do not use the LIST or OUT= option, the SPARSE option has no effect.

When you specify the SPARSE and LIST options, PROC FREQ displays all combinations of variable values in the table listing, including those with a frequency count of zero. By default, without the SPARSE option, PROC FREQ does not display zero-frequency levels in LIST output. When you use the SPARSE and OUT= options, PROC FREQ includes empty crosstabulation table cells in the output data set. By default, PROC FREQ does not include zero-frequency table cells in the output data set.

See the section "Missing Values" on page 1726 for more information.

TESTF=(*values*)

specifies the null hypothesis frequencies for a one-way chi-square goodness-of-fit test, which you request with the CHISQ option. See the section "Chi-Square Test for One-Way Tables" on page 1732 for details.

You can separate the TESTF= *values* with blanks or commas. The number of *values* must equal the number of variable levels in the one-way table. The sum of the *values* must equal the total frequency for the one-way table. List the *values* in the order in which the corresponding variable levels appear in the output. If you omit the CHISQ option, the TESTF= option invokes CHISQ.

TESTP=(*values*)

specifies the null hypothesis proportions for a one-way chi-square goodness-of-fit test, which you request with the CHISQ option. See the section "Chi-Square Test for One-Way Tables" on page 1732 for details.

You can separate the TESTP= *values* with blanks or commas. The number of *values* must equal the number of variable levels in the one-way table. List the *values* in the order in which the corresponding variable levels appear in the output. You can specify *values* in probability form as numbers between 0 and 1, where the proportions sum to 1. Or you can specify *values* in percentage form as numbers between 0 and 100, where the percentages sum to 100. If you omit the CHISQ option, the TESTP= option invokes CHISQ.

TOTPCT

displays the percentage of the total multiway table frequency in crosstabulation tables for n-way tables, where $n > 2$. By default, PROC FREQ displays the percentage of the individual two-way table frequency but does not display the percentage of the total frequency for multiway crosstabulation tables. See the section "Multiway Tables" on page 1789 for more information.

The percentage of total multiway table frequency is displayed by default when you specify the LIST option. It is also provided by default in the PERCENT variable in the OUT= output data set.

TREND

requests the Cochran-Armitage test for trend. The table must be $2 \times C$ or $R \times 2$ to compute the trend test. See the section "Cochran-Armitage Test for Trend" on page 1762 for details.

TEST Statement

TEST *options* **;**

The TEST statement requests asymptotic tests for measures of association and measures of agreement. You must use a TABLES statement with the TEST statement.

options

specify the statistics for which to provide asymptotic tests. Table 35.13 lists the available statistics, which include measures of association and agreement. The option names are identical to those in the TABLES and OUTPUT statements. You can request all tests for groups of statistics by using group options MEASURES or AGREE. Or you can request tests individually by using the options shown in Table 35.13.

For each measure of association or agreement that you specify, PROC FREQ provides an asymptotic test that the measure equals zero. PROC FREQ displays the asymptotic standard error under the null hypothesis, the test statistic, and the *p*-values. Additionally, PROC FREQ reports the confidence limits for the measure. The ALPHA= option in the TABLES statement determines the confidence level, which by default equals 0.05 and provides 95% confidence limits. See the sections "Asymptotic Tests" on page 1737 and "Confidence Limits" on page 1737 for details. Also see the section "Statistical Computations" on page 1730 for information about individual measures.

You can request exact tests for selected measures of association and agreement by using the EXACT statement. See the section "Exact Statistics" on page 1779 for more information.

If you use only one TABLES statement, you do not need to specify the same options in both the TABLES and TEST statements; when you specify an option in the TEST statement, PROC FREQ automatically invokes the corresponding TABLES statement option. However, when you use the TEST statement with multiple TABLES statements, you must specify options in the TABLES statements to request the desired statistics. PROC FREQ then provides asymptotic tests for those statistics that you also specify in the TEST statement.

Table 35.13 TEST Statement Options

Option	Asymptotic Tests Computed	Required TABLES Statement Option
AGREE	simple and weighted kappa coefficients	AGREE
GAMMA	gamma	ALL or MEASURES
KAPPA	simple kappa coefficient	AGREE
KENTB	Kendall's tau-b	ALL or MEASURES
MEASURES	gamma, Kendall's tau-b, Stuart's tau-c, Somers' $D(C\|R)$, Somers' $D(R\|C)$, Pearson and Spearman correlations	ALL or MEASURES
PCORR	Pearson correlation coefficient	ALL or MEASURES
SCORR	Spearman correlation coefficient	ALL or MEASURES
SMDCR	Somers' $D(C\|R)$	ALL or MEASURES
SMDRC	Somers' $D(R\|C)$	ALL or MEASURES
STUTC	Stuart's tau-c	ALL or MEASURES
WTKAP	weighted kappa coefficient	AGREE

WEIGHT Statement

WEIGHT *variable* < / *option* > **;**

The WEIGHT statement names a numeric variable that provides a weight for each observation in the input data set. The WEIGHT statement is most commonly used to input cell count data. See the section "Inputting Frequency Counts" on page 1724 for more information. If you use a WEIGHT statement, PROC FREQ assumes that an observation represents n observations, where n is the value of *variable*. The value of the WEIGHT *variable* is not required to be an integer.

If the value of the WEIGHT *variable* is missing, PROC FREQ does not use that observation in the analysis. If the value of the WEIGHT *variable* is zero, PROC FREQ ignores the observation unless you specify the ZEROS option, which includes observations with zero weights. If you do not specify a WEIGHT statement, each observation has a default weight of 1. The sum of the WEIGHT *variable* values represents the total number of observations.

If any value of the WEIGHT *variable* is negative, PROC FREQ displays the frequencies computed from the weighted values but does not compute percentages and statistics. If you create an output data set by using the OUT= option in the TABLES statement, PROC FREQ assigns missing values to the PERCENT variable. PROC FREQ also assigns missing values to the variables that the OUTEXPECT and OUTPCT options provide. If any value of the WEIGHT *variable* is negative, you cannot create an output data set by using the OUTPUT statement because statistics are not computed when there are negative weights.

You can specify the following option in the WEIGHT statement:

ZEROS

includes observations with zero weight values. By default, PROC FREQ ignores observations with zero weights.

If you specify the ZEROS option, frequency and and crosstabulation tables display any levels corresponding to observations with zero weights. Without the ZEROS option, PROC FREQ does not process observations with zero weights, and so does not display levels that contain only observations with zero weights.

With the ZEROS option, PROC FREQ includes levels with zero weights in the chi-square goodness-of-fit test for one-way tables. Also, PROC FREQ includes any levels with zero weights in binomial computations for one-way tables. This makes it possible to compute binomial tests and estimates when the specified level contains no observations with positive weights.

For two-way tables, the ZEROS option enables computation of kappa statistics when there are levels that contain no observations with positive weight. For more information, see the section "Tables with Zero Rows and Columns" on page 1770.

Note that even with the ZEROS option, PROC FREQ does not compute the CHISQ or MEASURES statistics for two-way tables when the table has a zero row or zero column because most of these statistics are undefined in this case.

The ZEROS option invokes the SPARSE option in the TABLES statement, which includes table cells with a zero frequency count in the LIST output and in the OUT= data set. By default, without the SPARSE option, PROC FREQ does not include zero frequency cells in the LIST output or in the OUT= data set. If you specify the ZEROS option in the WEIGHT statement but do not want the SPARSE option, you can specify the NOSPARSE option in the TABLES statement.

Details: FREQ Procedure

Inputting Frequency Counts

PROC FREQ can use either raw data or cell count data to produce frequency and crosstabulation tables. *Raw data*, also known as case-record data, report the data as one record for each subject or sample member. *Cell count data* report the data as a table, listing all possible combinations of data values along with the frequency counts. This way of presenting data often appears in published results.

The following DATA step statements store raw data in a SAS data set:

```
data Raw;
   input Subject $ R C @@;
   datalines;
01 1 1  02 1 1  03 1 1  04 1 1  05 1 1
06 1 2  07 1 2  08 1 2  09 2 1  10 2 1
11 2 1  12 2 1  13 2 2  14 2 2  14 2 2
;
```

You can store the same data as cell counts by using the following DATA step statements:

```
data CellCounts;
   input R C Count @@;
   datalines;
1 1 5   1 2 3
2 1 4   2 2 3
;
```

The variable R contains the values for the rows, and the variable C contains the values for the columns. The variable Count contains the cell count for each row and column combination.

Both the Raw data set and the CellCounts data set produce identical frequency counts, two-way tables, and statistics. When using the CellCounts data set, you must include a WEIGHT statement to specify that the variable Count contains cell counts. For example, the following PROC FREQ statements create a two-way crosstabulation table by using the CellCounts data set:

```
proc freq data=CellCounts;
   tables R*C;
   weight Count;
run;
```

Grouping with Formats

PROC FREQ groups a variable's values according to its formatted values. If you assign a format to a variable with a FORMAT statement, PROC FREQ formats the variable values before dividing observations into the levels of a frequency or crosstabulation table.

For example, suppose that variable X has the values 1.1, 1.4, 1.7, 2.1, and 2.3. Each of these values appears as a level in the frequency table. If you decide to round each value to a single digit, include the following statement in the PROC FREQ step:

```
format X 1.;
```

Now the table lists the frequency count for formatted level 1 as two and for formatted level 2 as three.

PROC FREQ treats formatted character variables in the same way. The formatted values are used to group the observations into the levels of a frequency table or crosstabulation table. PROC FREQ uses the entire value of a character format to classify an observation.

You can also use the FORMAT statement to assign formats that were created with the FORMAT procedure to the variables. User-written formats determine the number of levels for a variable and provide labels for a table. If you use the same data with different formats, then you can produce frequency counts and statistics for different classifications of the variable values.

When you use PROC FORMAT to create a user-written format that combines missing and nonmissing values into one category, PROC FREQ treats the entire category of formatted values as missing. For example, a questionnaire codes 1 as yes, 2 as no, and 8 as a no answer. The following PROC FORMAT statements create a user-written format:

```
proc format;
   value Questfmt 1   ='Yes'
                  2   ='No'
                  8,. ='Missing';
run;
```

When you use a FORMAT statement to assign Questfmt. to a variable, the variable's frequency table no longer includes a frequency count for the response of 8. You must use the MISSING or MISSPRINT option in the TABLES statement to list the frequency for no answer. The frequency count for this level includes observations with either a value of 8 or a missing value (.).

The frequency or crosstabulation table lists the values of both character and numeric variables in ascending order based on internal (unformatted) variable values unless you change the order with the ORDER= option. To list the values in ascending order by formatted values, use ORDER=FORMATTED in the PROC FREQ statement.

For more information about the FORMAT statement, see *SAS Language Reference: Concepts*.

Missing Values

When the value of the WEIGHT variable is missing, PROC FREQ does not include that observation in the analysis.

PROC FREQ treats missing BY variable values like any other BY variable value. The missing values form a separate BY group.

If an observation has a missing value for a variable in a TABLES request, by default PROC FREQ does not include that observation in the frequency or crosstabulation table. Also by default, PROC FREQ does not include observations with missing values in the computation of percentages and statistics. The procedure displays the number of missing observations below each table.

PROC FREQ also reports the number of missing values in output data sets. The TABLES statement OUT= data set includes an observation that contains the missing value frequency. The NMISS option in the OUTPUT statement provides an output data set variable that contains the missing value frequency.

The following options change the way in which PROC FREQ handles missing values of TABLES variables:

MISSPRINT displays missing value frequencies in frequency or crosstabulation tables but does not include them in computations of percentages or statistics.

MISSING treats missing values as a valid nonmissing level for all TABLES variables. Displays missing levels in frequency and crosstabulation tables and includes them in computations of percentages and statistics.

This example shows the three ways that PROC FREQ can handle missing values of TABLES variables. The following DATA step statements create a data set with a missing value for the variable A.

```
data one;
   input A Freq;
   datalines;
1 2
2 2
. 2
;
```

The following PROC FREQ statements request a one-way frequency table for the variable A. The first request does not specify a missing value option. The second request specifies the MISSPRINT option in the TABLES statement. The third request specifies the MISSING option in the TABLES statement.

```
proc freq data=one;
   tables A;
   weight Freq;
   title 'Default';
run;
proc freq data=one;
   tables A / missprint;
   weight Freq;
   title 'MISSPRINT Option';
run;
proc freq data=one;
   tables A / missing;
   weight Freq;
   title 'MISSING Option';
run;
```

Figure 35.11 displays the frequency tables produced by this example. The first table shows PROC FREQ's default behavior for handling missing values. The observation with a missing value of the TABLES variable A is not included in the table, and the frequency of missing values is displayed below the table. The second table, for which the MISSPRINT option is specified, displays the missing observation but does not include its frequency when computing the total frequency and percentages. The third table shows that PROC FREQ treats the missing level as a valid nonmissing level when the MISSING option is specified. The table displays the missing level, and PROC FREQ includes this level when computing frequencies and percentages.

Figure 35.11 Missing Values in Frequency Tables

Default

The FREQ Procedure

A	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	2	50.00	2	50.00
2	2	50.00	4	100.00

Frequency Missing = 2

MISSPRINT Option

The FREQ Procedure

A	Frequency	Percent	Cumulative Frequency	Cumulative Percent
.	2	.	.	.
1	2	50.00	2	50.00
2	2	50.00	4	100.00

Frequency Missing = 2

MISSING Option

The FREQ Procedure

A	Frequency	Percent	Cumulative Frequency	Cumulative Percent
.	2	33.33	2	33.33
1	2	33.33	4	66.67
2	2	33.33	6	100.00

When a combination of variable values for a two-way table is missing, PROC FREQ assigns zero to the frequency count for the table cell. By default, PROC FREQ does not display missing combinations in LIST format. Also, PROC FREQ does not include missing combinations in the OUT= output data set by default. To include missing combinations, you can specify the SPARSE option with the LIST or OUT= option in the TABLES statement.

In-Database Computation

The FREQ procedure can use in-database computation to construct frequency and crosstabulation tables when the DATA= input data set is stored as a table in a Teradata database management system (DBMS). In-database computation can provide the advantages of faster processing and reduced data

transfer between the database and SAS software. For information about in-database computation, see the section "In-Database Procedures in Teradata" in *SAS/ACCESS 9.2 for Relational Databases: Reference*.

PROC FREQ performs in-database computation by using SQL implicit pass-through. The procedure generates SQL queries that are based on the tables you request in the TABLES statement. The database executes these SQL queries to construct initial summary tables, which are then transmitted to PROC FREQ. The procedure uses this summary information to perform the remaining analyses and tasks in the usual way (out of the database). So instead of transferring the entire data set over the network between the database and SAS software, the in-database method transfers only the summary tables. This can substantially reduce processing time when the dimensions of the summary tables (in terms of rows and columns) are much smaller than the dimensions of the entire database table (in terms of individual observations). Additionally, in-database summarization uses efficient parallel processing, which can also provide performance advantages.

In-database computation is controlled by the SQLGENERATION option, which you can specify in either a LIBNAME statement or an OPTIONS statement. See the section "In-Database Computing in Teradata" in *SAS/ACCESS 9.2 for Relational Databases: Reference* for details about the SQLGENERATION option and other options that affect in-database computation. By default, PROC FREQ uses in-database computation when possible. There are no FREQ procedure options that control in-database computation.

PROC FREQ uses formatted values to group observations into the levels of frequency and crosstabulation tables. See the section "Grouping with Formats" on page 1725 for more information. If formats are available in the Teradata database, then in-database summarization uses the formats. If formats are not available in the Teradata database, then in-database summarization is based on the raw data values, and PROC FREQ performs the final, formatted classification (out of the database). For more information, see the section "Deploying and Using SAS Formats in Teradata" in *SAS/ACCESS 9.2 for Relational Databases: Reference*.

The order of observations is not inherently defined for DBMS tables. The following options relate to the order of observations and therefore should not be specified for PROC FREQ in-database computation:

- If you specify the FIRSTOBS= or OBS= data set option, PROC FREQ does not perform in-database computation.
- If you specify the NOTSORTED option in the BY statement, PROC FREQ in-database computation ignores it and uses the default ASCENDING order for BY variables.
- If you specify the ORDER=DATA option for input data in a DBMS table, PROC FREQ computation might produce different results for separate runs of the same analysis. In addition to determining the order of variable levels in crosstabulation table displays, the ORDER= option can also affect the values of many of the test statistics and measures that PROC FREQ computes.

Statistical Computations

Definitions and Notation

A two-way table represents the crosstabulation of row variable X and column variable Y. Let the table row values or levels be denoted by X_i, $i = 1, 2, \ldots, R$, and the column values by Y_j, $j = 1, 2, \ldots, C$. Let n_{ij} denote the frequency of the table cell in the ith row and jth column and define the following notation:

$$n_{i\cdot} = \sum_j n_{ij} \quad \text{(row totals)}$$

$$n_{\cdot j} = \sum_i n_{ij} \quad \text{(column totals)}$$

$$n = \sum_i \sum_j n_{ij} \quad \text{(overall total)}$$

$$p_{ij} = n_{ij}/n \quad \text{(cell percentages)}$$

$$p_{i\cdot} = n_{i\cdot}/n \quad \text{(row percentages of total)}$$

$$p_{\cdot j} = n_{\cdot j}/n \quad \text{(column percentages of total)}$$

$$R_i = \text{score for row } i$$

$$C_j = \text{score for column } j$$

$$\bar{R} = \sum_i n_{i\cdot} R_i / n \quad \text{(average row score)}$$

$$\bar{C} = \sum_j n_{\cdot j} C_j / n \quad \text{(average column score)}$$

$$A_{ij} = \sum_{k>i} \sum_{l>j} n_{kl} + \sum_{k<i} \sum_{l<j} n_{kl}$$

$$D_{ij} = \sum_{k>i} \sum_{l<j} n_{kl} + \sum_{k<i} \sum_{l>j} n_{kl}$$

$$P = \sum_i \sum_j n_{ij} A_{ij} \quad \text{(twice the number of concordances)}$$

$$Q = \sum_i \sum_j n_{ij} D_{ij} \quad \text{(twice the number of discordances)}$$

Scores

PROC FREQ uses scores of the variable values to compute the Mantel-Haenszel chi-square, Pearson correlation, Cochran-Armitage test for trend, weighted kappa coefficient, and Cochran-Mantel-Haenszel statistics. The SCORES= option in the TABLES statement specifies the score type that PROC FREQ uses. The available score types are TABLE, RANK, RIDIT, and MODRIDIT scores. The default score type is TABLE. Using MODRIDIT, RANK, or RIDIT scores yields nonparametric analyses.

For numeric variables, table scores are the values of the row and column levels. If the row or column variable is formatted, then the table score is the internal numeric value corresponding to that level. If two or more numeric values are classified into the same formatted level, then the internal numeric value for that level is the smallest of these values. For character variables, table scores are defined as the row numbers and column numbers (that is, 1 for the first row, 2 for the second row, and so on).

Rank scores, which you request with the SCORES=RANK option, are defined as

$$R1_i = \sum_{k<i} n_{k\cdot} + (n_{i\cdot} + 1)/2 \qquad i = 1, 2, \ldots, R$$
$$C1_j = \sum_{l<j} n_{\cdot l} + (n_{\cdot j} + 1)/2 \qquad j = 1, 2, \ldots, C$$

where $R1_i$ is the rank score of row i, and $C1_j$ is the rank score of column j. Note that rank scores yield midranks for tied values.

Ridit scores, which you request with the SCORES=RIDIT option, are defined as rank scores standardized by the sample size (Bross 1958, Mack and Skillings 1980). Ridit scores are derived from the rank scores as

$$R2_i = R1_i/n \qquad i = 1, 2, \ldots, R$$
$$C2_j = C1_j/n \qquad j = 1, 2, \ldots, C$$

Modified ridit scores (SCORES=MODRIDIT) represent the expected values of the order statistics of the uniform distribution on (0,1) (van Elteren 1960, Lehmann 1975). Modified ridit scores are derived from rank scores as

$$R3_i = R1_i/(n+1) \qquad i = 1, 2, \ldots, R$$
$$C3_j = C1_j/(n+1) \qquad j = 1, 2, \ldots, C$$

Chi-Square Tests and Statistics

The CHISQ option provides chi-square tests of homogeneity or independence and measures of association based on the chi-square statistic. When you specify the CHISQ option in the TABLES statement, PROC FREQ computes the following chi-square tests for each two-way table: the Pearson chi-square, likelihood-ratio chi-square, and Mantel-Haenszel chi-square. PROC FREQ provides the following measures of association based on the Pearson chi-square statistic: the phi coefficient, contingency coefficient, and Cramer's V. For 2×2 tables, the CHISQ option also provides Fisher's exact test and the continuity-adjusted chi-square. You can request Fisher's exact test for general $R \times C$ tables by specifying the FISHER option in the TABLES or EXACT statement.

For one-way frequency tables, the CHISQ option provides a chi-square goodness-of-fit test. The other chi-square tests and statistics described in this section are computed only for two-way tables.

All of the two-way test statistics described in this section test the null hypothesis of no association between the row variable and the column variable. When the sample size n is large, these test statistics have an asymptotic chi-square distribution when the null hypothesis is true. When the sample size is not large, exact tests might be useful. PROC FREQ provides exact tests for the

Pearson chi-square, the likelihood-ratio chi-square, and the Mantel-Haenszel chi-square (in addition to Fisher's exact test). PROC FREQ also provides an exact chi-square goodness-of-fit test for one-way tables. You can request these exact tests by specifying the corresponding options in the EXACT statement. See the section "Exact Statistics" on page 1779 for more information.

Note that the Mantel-Haenszel chi-square statistic is appropriate only when both variables lie on an ordinal scale. The other chi-square tests and statistics in this section are appropriate for either nominal or ordinal variables. The following sections give the formulas that PROC FREQ uses to compute the chi-square tests and statistics. See Agresti (2007), Stokes, Davis, and Koch (2000), and the other references cited for each statistic for more information.

Chi-Square Test for One-Way Tables

For one-way frequency tables, the CHISQ option in the TABLES statement provides a chi-square goodness-of-fit test. Let C denote the number of classes, or levels, in the one-way table. Let f_i denote the frequency of class i (or the number of observations in class i) for $i = 1, 2, \ldots, C$. Then PROC FREQ computes the one-way chi-square statistic as

$$Q_P = \sum_{i=1}^{C} \frac{(f_i - e_i)^2}{e_i}$$

where e_i is the expected frequency for class i under the null hypothesis.

In the test for equal proportions, which is the default for the CHISQ option, the null hypothesis specifies equal proportions of the total sample size for each class. Under this null hypothesis, the expected frequency for each class equals the total sample size divided by the number of classes,

$$e_i = n/C \quad \text{for } i = 1, 2, \ldots, C$$

In the test for specified frequencies, which PROC FREQ computes when you input null hypothesis frequencies by using the TESTF= option, the expected frequencies are the TESTF= values that you specify. In the test for specified proportions, which PROC FREQ computes when you input null hypothesis proportions by using the TESTP= option, the expected frequencies are determined from the specified TESTP= proportions p_i as

$$e_i = p_i \times n \quad \text{for } i = 1, 2, \ldots, C$$

Under the null hypothesis (of equal proportions, specified frequencies, or specified proportions), Q_P has an asymptotic chi-square distribution with $C - 1$ degrees of freedom.

In addition to the asymptotic test, you can request an exact one-way chi-square test by specifying the CHISQ option in the EXACT statement. See the section "Exact Statistics" on page 1779 for more information.

Pearson Chi-Square Test for Two-Way Tables

The Pearson chi-square for two-way tables involves the differences between the observed and expected frequencies, where the expected frequencies are computed under the null hypothesis of in-

dependence. The Pearson chi-square statistic is computed as

$$Q_P = \sum_i \sum_j \frac{(n_{ij} - e_{ij})^2}{e_{ij}}$$

where n_{ij} is the observed frequency in table cell (i, j) and e_{ij} is the expected frequency for table cell (i, j). The expected frequency is computed under the null hypothesis that the row and column variables are independent,

$$e_{ij} = \frac{n_{i\cdot}\, n_{\cdot j}}{n}$$

When the row and column variables are independent, Q_P has an asymptotic chi-square distribution with $(R-1)(C-1)$ degrees of freedom. For large values of Q_P, this test rejects the null hypothesis in favor of the alternative hypothesis of general association.

In addition to the asymptotic test, you can request an exact Pearson chi-square test by specifying the PCHI or CHISQ option in the EXACT statement. See the section "Exact Statistics" on page 1779 for more information.

For 2×2 tables, the Pearson chi-square is also appropriate for testing the equality of two binomial proportions. For $R \times 2$ and $2 \times C$ tables, the Pearson chi-square tests the homogeneity of proportions. See Fienberg (1980) for details.

Likelihood-Ratio Chi-Square Test

The likelihood-ratio chi-square involves the ratios between the observed and expected frequencies. The likelihood-ratio chi-square statistic is computed as

$$G^2 = 2 \sum_i \sum_j n_{ij} \ln\left(\frac{n_{ij}}{e_{ij}}\right)$$

where n_{ij} is the observed frequency in table cell (i, j) and e_{ij} is the expected frequency for table cell (i, j).

When the row and column variables are independent, G^2 has an asymptotic chi-square distribution with $(R-1)(C-1)$ degrees of freedom.

In addition to the asymptotic test, you can request an exact likelihood-ratio chi-square test by specifying the LRCHI or CHISQ option in the EXACT statement. See the section "Exact Statistics" on page 1779 for more information.

Continuity-Adjusted Chi-Square Test

The continuity-adjusted chi-square for 2×2 tables is similar to the Pearson chi-square, but it is adjusted for the continuity of the chi-square distribution. The continuity-adjusted chi-square is most useful for small sample sizes. The use of the continuity adjustment is somewhat controversial; this chi-square test is more conservative (and more like Fisher's exact test) when the sample size is small. As the sample size increases, the continuity-adjusted chi-square becomes more like the Pearson chi-square.

The continuity-adjusted chi-square statistic is computed as

$$Q_C = \sum_i \sum_j \frac{\left(\max(0, |n_{ij} - e_{ij}| - 0.5) \right)^2}{e_{ij}}$$

Under the null hypothesis of independence, Q_C has an asymptotic chi-square distribution with $(R-1)(C-1)$ degrees of freedom.

Mantel-Haenszel Chi-Square Test

The Mantel-Haenszel chi-square statistic tests the alternative hypothesis that there is a linear association between the row variable and the column variable. Both variables must lie on an ordinal scale. The Mantel-Haenszel chi-square statistic is computed as

$$Q_{MH} = (n-1)r^2$$

where r^2 is the Pearson correlation between the row variable and the column variable. For a description of the Pearson correlation, see the "Pearson Correlation Coefficient" on page 1740. The Pearson correlation and thus the Mantel-Haenszel chi-square statistic use the scores that you specify in the SCORES= option in the TABLES statement. See Mantel and Haenszel (1959) and Landis, Heyman, and Koch (1978) for more information.

Under the null hypothesis of no association, Q_{MH} has an asymptotic chi-square distribution with one degree of freedom.

In addition to the asymptotic test, you can request an exact Mantel-Haenszel chi-square test by specifying the MHCHI or CHISQ option in the EXACT statement. See the section "Exact Statistics" on page 1779 for more information.

Fisher's Exact Test

Fisher's exact test is another test of association between the row and column variables. This test assumes that the row and column totals are fixed, and then uses the hypergeometric distribution to compute probabilities of possible tables conditional on the observed row and column totals. Fisher's exact test does not depend on any large-sample distribution assumptions, and so it is appropriate even for small sample sizes and for sparse tables.

2 × 2 Tables For 2 × 2 tables, PROC FREQ gives the following information for Fisher's exact test: table probability, two-sided p-value, left-sided p-value, and right-sided p-value. The table probability equals the hypergeometric probability of the observed table, and is in fact the value of the test statistic for Fisher's exact test.

Where p is the hypergeometric probability of a specific table with the observed row and column totals, Fisher's exact p-values are computed by summing probabilities p over defined sets of tables,

$$PROB = \sum_A p$$

The two-sided p-value is the sum of all possible table probabilties (conditional on the observed row and column totals) that are less than or equal to the observed table probability. For the two-sided

p-value, the set A includes all possible tables with hypergeometric probabilities less than or equal to the probability of the observed table. A small two-sided p-value supports the alternative hypothesis of association between the row and column variables.

For 2×2 tables, one-sided p-values for Fisher's exact test are defined in terms of the frequency of the cell in the first row and first column of the table, the (1,1) cell. Denoting the observed (1,1) cell frequency by n_{11}, the left-sided p-value for Fisher's exact test is the probability that the (1,1) cell frequency is less than or equal to n_{11}. For the left-sided p-value, the set A includes those tables with a (1,1) cell frequency less than or equal to n_{11}. A small left-sided p-value supports the alternative hypothesis that the probability of an observation being in the first cell is actually less than expected under the null hypothesis of independent row and column variables.

Similarly, for a right-sided alternative hypothesis, A is the set of tables where the frequency of the (1,1) cell is greater than or equal to that in the observed table. A small right-sided p-value supports the alternative that the probability of the first cell is actually greater than that expected under the null hypothesis.

Because the (1,1) cell frequency completely determines the 2×2 table when the marginal row and column sums are fixed, these one-sided alternatives can be stated equivalently in terms of other cell probabilities or ratios of cell probabilities. The left-sided alternative is equivalent to an odds ratio less than 1, where the odds ratio equals $(n_{11}n_{22}/n_{12}n_{21})$. Additionally, the left-sided alternative is equivalent to the column 1 risk for row 1 being less than the column 1 risk for row 2, $p_{1|1} < p_{1|2}$. Similarly, the right-sided alternative is equivalent to the column 1 risk for row 1 being greater than the column 1 risk for row 2, $p_{1|1} > p_{1|2}$. See Agresti (2007) for details.

R × C Tables Fisher's exact test was extended to general $R \times C$ tables by Freeman and Halton (1951), and this test is also known as the Freeman-Halton test. For $R \times C$ tables, the two-sided p-value definition is the same as for 2×2 tables. The set A contains all tables with p less than or equal to the probability of the observed table. A small p-value supports the alternative hypothesis of association between the row and column variables. For $R \times C$ tables, Fisher's exact test is inherently two-sided. The alternative hypothesis is defined only in terms of general, and not linear, association. Therefore, Fisher's exact test does not have right-sided or left-sided p-values for general $R \times C$ tables.

For $R \times C$ tables, PROC FREQ computes Fisher's exact test by using the network algorithm of Mehta and Patel (1983), which provides a faster and more efficient solution than direct enumeration. See the section "Exact Statistics" on page 1779 for more details.

Phi Coefficient

The phi coefficient is a measure of association derived from the Pearson chi-square. The range of the phi coefficient is $-1 \leq \phi \leq 1$ for 2×2 tables. For tables larger than 2×2, the range is $0 \leq \phi \leq \min(\sqrt{R-1}, \sqrt{C-1})$ (Liebetrau 1983). The phi coefficient is computed as

$$\phi = (n_{11}n_{22} - n_{12}n_{21}) \,/\, \sqrt{n_{1\cdot}n_{2\cdot}n_{\cdot 1}n_{\cdot 2}} \quad \text{for } 2 \times 2 \text{ tables}$$

$$\phi = \sqrt{Q_P/n} \quad \text{otherwise}$$

See Fleiss, Levin, and Paik (2003, pp. 98–99) for more information.

Contingency Coefficient

The contingency coefficient is a measure of association derived from the Pearson chi-square. The range of the contingency coefficient is $0 \leq P \leq \sqrt{(m-1)/m}$, where $m = \min(R, C)$ (Liebetrau 1983). The contingency coefficient is computed as

$$P = \sqrt{Q_P \,/\, (Q_P + n)}$$

See Kendall and Stuart (1979, pp. 587–588) for more information.

Cramer's V

Cramer's V is a measure of association derived from the Pearson chi-square. It is designed so that the attainable upper bound is always 1. The range of Cramer's V is $-1 \leq V \leq 1$ for 2×2 tables; for tables larger than 2×2, the range is $0 \leq V \leq 1$. Cramer's V is computed as

$$V = \phi \quad \text{for } 2 \times 2 \text{tables}$$

$$V = \sqrt{\frac{Q_P/n}{\min(R-1, C-1)}} \quad \text{otherwise}$$

See Kendall and Stuart (1979, p. 588) for more information.

Measures of Association

When you specify the MEASURES option in the TABLES statement, PROC FREQ computes several statistics that describe the association between the row and column variables of the contingency table. The following are measures of ordinal association that consider whether the column variable Y tends to increase as the row variable X increases: gamma, Kendall's tau-b, Stuart's tau-c, and Somers' D. These measures are appropriate for ordinal variables, and they classify pairs of observations as *concordant* or *discordant*. A pair is concordant if the observation with the larger value of X also has the larger value of Y. A pair is discordant if the observation with the larger value of X has the smaller value of Y. See Agresti (2007) and the other references cited for the individual measures of association.

The Pearson correlation coefficient and the Spearman rank correlation coefficient are also appropriate for ordinal variables. The Pearson correlation describes the strength of the linear association between the row and column variables, and it is computed by using the row and column scores specified by the SCORES= option in the TABLES statement. The Spearman correlation is computed with rank scores. The polychoric correlation (requested by the PLCORR option) also requires ordinal variables and assumes that the variables have an underlying bivariate normal distribution. The following measures of association do not require ordinal variables and are appropriate for nominal variables: lambda asymmetric, lambda symmetric, and the uncertainty coefficients.

PROC FREQ computes estimates of the measures according to the formulas given in the following sections. For each measure, PROC FREQ computes an asymptotic standard error (*ASE*), which is the square root of the asymptotic variance denoted by var in the following sections.

Confidence Limits

If you specify the CL option in the TABLES statement, PROC FREQ computes asymptotic confidence limits for all MEASURES statistics. The confidence coefficient is determined according to the value of the ALPHA= option, which, by default, equals 0.05 and produces 95% confidence limits.

The confidence limits are computed as

$$est \pm (z_{\alpha/2} \times ASE)$$

where est is the estimate of the measure, $z_{\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the standard normal distribution, and ASE is the asymptotic standard error of the estimate.

Asymptotic Tests

For each measure that you specify in the TEST statement, PROC FREQ computes an asymptotic test of the null hypothesis that the measure equals zero. Asymptotic tests are available for the following measures of association: gamma, Kendall's tau-b, Stuart's tau-c, Somers' $D(R|C)$, Somers' $D(C|R)$, the Pearson correlation coefficient, and the Spearman rank correlation coefficient. To compute an asymptotic test, PROC FREQ uses a standardized test statistic z, which has an asymptotic standard normal distribution under the null hypothesis. The test statistic is computed as

$$z = est \; / \; \sqrt{\text{var}_0(est)}$$

where est is the estimate of the measure and $\text{var}_0(est)$ is the variance of the estimate under the null hypothesis. Formulas for $\text{var}_0(est)$ for the individual measures of association are given in the following sections.

Note that the ratio of est to $\sqrt{\text{var}_0(est)}$ is the same for the following measures: gamma, Kendall's tau-b, Stuart's tau-c, Somers' $D(R|C)$, and Somers' $D(C|R)$. Therefore, the tests for these measures are identical. For example, the p-values for the test of H_0: gamma $= 0$ equal the p-values for the test of H_0: tau-$b = 0$.

PROC FREQ computes one-sided and two-sided p-values for each of these tests. When the test statistic z is greater than its null hypothesis expected value of zero, PROC FREQ displays the right-sided p-value, which is the probability of a larger value of the statistic occurring under the null hypothesis. A small right-sided p-value supports the alternative hypothesis that the true value of the measure is greater than zero. When the test statistic is less than or equal to zero, PROC FREQ displays the left-sided p-value, which is the probability of a smaller value of the statistic occurring under the null hypothesis. A small left-sided p-value supports the alternative hypothesis that the true value of the measure is less than zero. The one-sided p-value P_1 can be expressed as

$$P_1 = \begin{cases} \text{Prob}(Z > z) & \text{if } z > 0 \\ \text{Prob}(Z < z) & \text{if } z \leq 0 \end{cases}$$

where Z has a standard normal distribution. The two-sided p-value P_2 is computed as

$$P_2 = \text{Prob}(|Z| > |z|)$$

Exact Tests

Exact tests are available for two measures of association: the Pearson correlation coefficient and the Spearman rank correlation coefficient. If you specify the PCORR option in the EXACT statement, PROC FREQ computes the exact test of the hypothesis that the Pearson correlation equals zero. If you specify the SCORR option in the EXACT statement, PROC FREQ computes the exact test of the hypothesis that the Spearman correlation equals zero. See the section "Exact Statistics" on page 1779 for more information.

Gamma

The gamma (Γ) statistic is based only on the number of concordant and discordant pairs of observations. It ignores tied pairs (that is, pairs of observations that have equal values of X or equal values of Y). Gamma is appropriate only when both variables lie on an ordinal scale. The range of gamma is $-1 \leq \Gamma \leq 1$. If the row and column variables are independent, then gamma tends to be close to zero. Gamma is estimated by

$$G = (P - Q) \,/\, (P + Q)$$

and the asymptotic variance is

$$\mathrm{var}(G) = \frac{16}{(P+Q)^4} \sum_i \sum_j n_{ij} (QA_{ij} - PD_{ij})^2$$

For 2×2 tables, gamma is equivalent to Yule's Q. See Goodman and Kruskal (1979) and Agresti (2002) for more information.

The variance under the null hypothesis that gamma equals zero is computed as

$$\mathrm{var}_0(G) = \frac{4}{(P+Q)^2} \left(\sum_i \sum_j n_{ij} (A_{ij} - D_{ij})^2 - (P-Q)^2/n \right)$$

See Brown and Benedetti (1977) for details.

Kendall's Tau-b

Kendall's tau-b (τ_b) is similar to gamma except that tau-b uses a correction for ties. Tau-b is appropriate only when both variables lie on an ordinal scale. The range of tau-b is $-1 \leq \tau_b \leq 1$. Kendall's tau-b is estimated by

$$t_b = (P - Q) \,/\, \sqrt{w_r w_c}$$

and the asymptotic variance is

$$\mathrm{var}(t_b) = \frac{1}{w^4} \left(\sum_i \sum_j n_{ij} (2wd_{ij} + t_b v_{ij})^2 - n^3 t_b^2 (w_r + w_c)^2 \right)$$

where

$$\begin{aligned} w &= \sqrt{w_r w_c} \\ w_r &= n^2 - \sum_i n_{i.}^2 \\ w_c &= n^2 - \sum_j n_{.j}^2 \\ d_{ij} &= A_{ij} - D_{ij} \\ v_{ij} &= n_{i.} w_c + n_{.j} w_r \end{aligned}$$

See Kendall (1955) for more information.

The variance under the null hypothesis that tau-b equals zero is computed as

$$\text{var}_0(t_b) = \frac{4}{w_r w_c} \left(\sum_i \sum_j n_{ij} (A_{ij} - D_{ij})^2 - (P - Q)^2 / n \right)$$

See Brown and Benedetti (1977) for details.

Stuart's Tau-c

Stuart's tau-c (τ_c) makes an adjustment for table size in addition to a correction for ties. Tau-c is appropriate only when both variables lie on an ordinal scale. The range of tau-c is $-1 \leq \tau_c \leq 1$. Stuart's tau-c is estimated by

$$t_c = m(P - Q) \; / \; n^2 (m - 1)$$

and the asymptotic variance is

$$\text{var}(t_c) = \frac{4m^2}{(m-1)^2 n^4} \left(\sum_i \sum_j n_{ij} d_{ij}^2 - (P - Q)^2 / n \right)$$

where $m = \min(R, C)$ and $d_{ij} = A_{ij} - D_{ij}$. The variance under the null hypothesis that tau-c equals zero is the same as the asymptotic variance var,

$$\text{var}_0(t_c) = \text{var}(t_c)$$

See Brown and Benedetti (1977) for details.

Somers' D

Somers' $D(C|R)$ and Somers' $D(R|C)$ are asymmetric modifications of tau-b. $C|R$ indicates that the row variable X is regarded as the independent variable and the column variable Y is regarded as dependent. Similarly, $R|C$ indicates that the column variable Y is regarded as the independent

variable and the row variable X is regarded as dependent. Somers' D differs from tau-b in that it uses a correction only for pairs that are tied on the independent variable. Somers' D is appropriate only when both variables lie on an ordinal scale. The range of Somers' D is $-1 \leq D \leq 1$. Somers' $D(C|R)$ is computed as

$$D(C|R) = (P - Q) \,/\, w_r$$

and its asymptotic variance is

$$\mathrm{var}(D(C|R)) = \frac{4}{w_r^4} \sum_i \sum_j n_{ij} \left(w_r d_{ij} - (P - Q)(n - n_{i\cdot}) \right)^2$$

where $d_{ij} = A_{ij} - D_{ij}$ and

$$w_r = n^2 - \sum_i n_{i\cdot}^2$$

See Somers (1962), Goodman and Kruskal (1979), and Liebetrau (1983) for more information.

The variance under the null hypothesis that $D(C|R)$ equals zero is computed as

$$\mathrm{var}_0(D(C|R)) = \frac{4}{w_r^2} \left(\sum_i \sum_j n_{ij} (A_{ij} - D_{ij})^2 - (P - Q)^2 / n \right)$$

See Brown and Benedetti (1977) for details.

Formulas for Somers' $D(R|C)$ are obtained by interchanging the indices.

Pearson Correlation Coefficient

The Pearson correlation coefficient (ρ) is computed by using the scores specified in the SCORES= option. This measure is appropriate only when both variables lie on an ordinal scale. The range of the Pearson correlation is $-1 \leq \rho \leq 1$. The Pearson correlation coefficient is estimated by

$$r = v/w = ss_{rc} / \sqrt{ss_r ss_c}$$

and its asymptotic variance is

$$\mathrm{var}(r) = \frac{1}{w^4} \sum_i \sum_j n_{ij} \left(w(R_i - \bar{R})(C_j - \bar{C}) - \frac{b_{ij} v}{2w} \right)^2$$

where R_i and C_j are the row and column scores and

$$ss_r = \sum_i \sum_j n_{ij} (R_i - \bar{R})^2$$

$$ss_c = \sum_i \sum_j n_{ij} (C_j - \bar{C})^2$$

$$ss_{rc} = \sum_i \sum_j n_{ij} (R_i - \bar{R})(C_j - \bar{C})$$

$$
\begin{aligned}
b_{ij} &= (R_i - \bar{R})^2 ss_c + (C_j - \bar{C})^2 ss_r \\
v &= ss_{rc} \\
w &= \sqrt{ss_r ss_c}
\end{aligned}
$$

See Snedecor and Cochran (1989) for more information.

The SCORES= option in the TABLES statement determines the type of row and column scores used to compute the Pearson correlation (and other score-based statistics). The default is SCORES=TABLE. See the section "Scores" on page 1730 for details about the available score types and how they are computed.

The variance under the null hypothesis that the correlation equals zero is computed as

$$\text{var}_0(r) = \left(\sum_i \sum_j n_{ij} (R_i - \bar{R})^2 (C_j - \bar{C})^2 - ss_{rc}^2 / n \right) \Big/ ss_r ss_c$$

Note that this expression for the variance is derived for multinomial sampling in a contingency table framework, and it differs from the form obtained under the assumption that both variables are continuous and normally distributed. See Brown and Benedetti (1977) for details.

PROC FREQ also provides an exact test for the Pearson correlation coefficient. You can request this test by specifying the PCORR option in the EXACT statement. See the section "Exact Statistics" on page 1779 for more information.

Spearman Rank Correlation Coefficient

The Spearman correlation coefficient (ρ_s) is computed by using rank scores, which are defined in the section "Scores" on page 1730. This measure is appropriate only when both variables lie on an ordinal scale. The range of the Spearman correlation is $-1 \leq \rho_s \leq 1$. The Spearman correlation coefficient is estimated by

$$r_s = v \,/\, w$$

and its asymptotic variance is

$$\text{var}(r_s) = \frac{1}{n^2 w^4} \sum_i \sum_j n_{ij} (z_{ij} - \bar{z})^2$$

where $R1_i$ and $C1_j$ are the row and column rank scores and

$$
\begin{aligned}
v &= \sum_i \sum_j n_{ij} R(i) C(j) \\
w &= \frac{1}{12} \sqrt{FG} \\
F &= n^3 - \sum_i n_{i\cdot}^3 \\
G &= n^3 - \sum_j n_{\cdot j}^3
\end{aligned}
$$

$$
\begin{aligned}
R(i) &= R1_i - n/2 \\
C(j) &= C1_j - n/2 \\
\bar{z} &= \frac{1}{n} \sum_i \sum_j n_{ij} z_{ij} \\
z_{ij} &= w v_{ij} - v w_{ij} \\
v_{ij} &= n \left(R(i)C(j) + \frac{1}{2} \sum_l n_{il} C(l) + \frac{1}{2} \sum_k n_{kj} R(k) + \right. \\
&\qquad \left. \sum_l \sum_{k>i} n_{kl} C(l) + \sum_k \sum_{l>j} n_{kl} R(k) \right) \\
w_{ij} &= \frac{-n}{96w} \left(F n_{\cdot j}^2 + G n_{i\cdot}^2 \right)
\end{aligned}
$$

See Snedecor and Cochran (1989) for more information.

The variance under the null hypothesis that the correlation equals zero is computed as

$$\text{var}_0(r_s) = \frac{1}{n^2 w^2} \sum_i \sum_j n_{ij} (v_{ij} - \bar{v})^2$$

where

$$\bar{v} = \sum_i \sum_j n_{ij} v_{ij} / n$$

Note that the asymptotic variance is derived for multinomial sampling in a contingency table framework, and it differs from the form obtained under the assumption that both variables are continuous and normally distributed. See Brown and Benedetti (1977) for details.

PROC FREQ also provides an exact test for the Spearman correlation coefficient. You can request this test by specifying the SCORR option in the EXACT statement. See the section "Exact Statistics" on page 1779 for more information.

Polychoric Correlation

When you specify the PLCORR option in the TABLES statement, PROC FREQ computes the polychoric correlation. This measure of association is based on the assumption that the ordered, categorical variables of the frequency table have an underlying bivariate normal distribution. For 2×2 tables, the polychoric correlation is also known as the tetrachoric correlation. See Drasgow (1986) for an overview of polychoric correlation. The polychoric correlation coefficient is the maximum likelihood estimate of the product-moment correlation between the normal variables, estimating thresholds from the observed table frequencies. The range of the polychoric correlation is from –1 to 1. Olsson (1979) gives the likelihood equations and an asymptotic covariance matrix for the estimates.

To estimate the polychoric correlation, PROC FREQ iteratively solves the likelihood equations by a Newton-Raphson algorithm that uses the Pearson correlation coefficient as the initial approximation. Iteration stops when the convergence measure falls below the convergence criterion or when the maximum number of iterations is reached, whichever occurs first. The CONVERGE= option sets the convergence criterion, and the default value is 0.0001. The MAXITER= option sets the maximum number of iterations, and the default value is 20.

Lambda (Asymmetric)

Asymmetric lambda, $\lambda(C|R)$, is interpreted as the probable improvement in predicting the column variable Y given knowledge of the row variable X. The range of asymmetric lambda is $0 \leq \lambda(C|R) \leq 1$. Asymmetric lambda $(C|R)$ is computed as

$$\lambda(C|R) = \frac{\sum_i r_i - r}{n - r}$$

and its asymptotic variance is

$$\text{var}(\lambda(C|R)) = \frac{n - \sum_i r_i}{(n-r)^3}\left(\sum_i r_i + r - 2\sum_i (r_i \mid l_i = l)\right)$$

where

$$\begin{aligned} r_i &= \max_j(n_{ij}) \\ r &= \max_j(n_{\cdot j}) \\ c_j &= \max_i(n_{ij}) \\ c &= \max_i(n_{i\cdot}) \end{aligned}$$

The values of l_i and l are determined as follows. Denote by l_i the unique value of j such that $r_i = n_{ij}$, and let l be the unique value of j such that $r = n_{\cdot j}$. Because of the uniqueness assumptions, ties in the frequencies or in the marginal totals must be broken in an arbitrary but consistent manner. In case of ties, l is defined as the smallest value of j such that $r = n_{\cdot j}$.

For those columns containing a cell (i, j) for which $n_{ij} = r_i = c_j$, cs_j records the row in which c_j is assumed to occur. Initially cs_j is set equal to –1 for all j. Beginning with $i = 1$, if there is at least one value j such that $n_{ij} = r_i = c_j$, and if $cs_j = -1$, then l_i is defined to be the smallest such value of j, and cs_j is set equal to i. Otherwise, if $n_{il} = r_i$, then l_i is defined to be equal to l. If neither condition is true, then l_i is taken to be the smallest value of j such that $n_{ij} = r_i$.

The formulas for lambda asymmetric $(R|C)$ can be obtained by interchanging the indices.

See Goodman and Kruskal (1979) for more information.

Lambda (Symmetric)

The nondirectional lambda is the average of the two asymmetric lambdas, $\lambda(C|R)$ and $\lambda(R|C)$. Its range is $0 \leq \lambda \leq 1$. Lambda symmetric is computed as

$$\lambda = \frac{\sum_i r_i + \sum_j c_j - r - c}{2n - r - c} = \frac{w - v}{w}$$

and its asymptotic variance is computed as

$$\text{var}(\lambda) = \frac{1}{w^4}\Big(wvy - 2w^2\big(n - \sum_i \sum_j (n_{ij} \mid j = l_i, i = k_j)\big) - 2v^2(n - n_{kl})\Big)$$

where

$$\begin{aligned}
r_i &= \max_j(n_{ij}) \\
r &= \max_j(n_{.j}) \\
c_j &= \max_i(n_{ij}) \\
c &= \max_i(n_{i.}) \\
w &= 2n - r - c \\
v &= 2n - \sum_i r_i - \sum_j c_j \\
x &= \sum_i (r_i \mid l_i = l) + \sum_j (c_j \mid k_j = k) + r_k + c_l \\
y &= 8n - w - v - 2x
\end{aligned}$$

The definitions of l_i and l are given in the previous section. The values k_j and k are defined in a similar way for lambda asymmetric $(R|C)$.

See Goodman and Kruskal (1979) for more information.

Uncertainty Coefficients (Asymmetric)

The uncertainty coefficient $U(C|R)$ measures the proportion of uncertainty (entropy) in the column variable Y that is explained by the row variable X. Its range is $0 \leq U(C|R) \leq 1$. The uncertainty coefficient is computed as

$$U(C|R) = (H(X) + H(Y) - H(XY)) \;/\; H(Y) = v/w$$

and its asymptotic variance is

$$\text{var}(U(C|R)) = \frac{1}{n^2 w^4} \sum_i \sum_j n_{ij}\Big(H(Y)\ln\Big(\frac{n_{ij}}{n_{i.}}\Big) + (H(X) - H(XY))\ln\Big(\frac{n_{.j}}{n}\Big)\Big)^2$$

where

$$\begin{aligned} v &= H(X) + H(Y) - H(XY) \\ w &= H(Y) \\ H(X) &= -\sum_i \left(\frac{n_{i\cdot}}{n}\right) \ln\left(\frac{n_{i\cdot}}{n}\right) \\ H(Y) &= -\sum_j \left(\frac{n_{\cdot j}}{n}\right) \ln\left(\frac{n_{\cdot j}}{n}\right) \\ H(XY) &= -\sum_i \sum_j \left(\frac{n_{ij}}{n}\right) \ln\left(\frac{n_{ij}}{n}\right) \end{aligned}$$

The formulas for the uncertainty coefficient $U(R|C)$ can be obtained by interchanging the indices.

See Theil (1972, pp. 115–120) and Goodman and Kruskal (1979) for more information.

Uncertainty Coefficient (Symmetric)

The uncertainty coefficient U is the symmetric version of the two asymmetric uncertainty coefficients. Its range is $0 \le U \le 1$. The uncertainty coefficient is computed as

$$U = 2\left(H(X) + H(Y) - H(XY)\right) \;/\; \left(H(X) + H(Y)\right)$$

and its asymptotic variance is

$$\text{var}(U) = 4 \sum_i \sum_j \frac{n_{ij}\left(H(XY)\ln\left(\frac{n_{i\cdot}n_{\cdot j}}{n^2}\right) - \left(H(X)+H(Y)\right)\ln\left(\frac{n_{ij}}{n}\right)\right)^2}{n^2\left(H(X)+H(Y)\right)^4}$$

where $H(X)$, $H(Y)$, and $H(XY)$ are defined in the previous section. See Goodman and Kruskal (1979) for more information.

Binomial Proportion

If you specify the BINOMIAL option in the TABLES statement, PROC FREQ computes the binomial proportion for one-way tables. By default, this is the proportion of observations in the first variable level that appears in the output. (You can use the LEVEL= option to specify a different level for the proportion.) The binomial proportion is computed as

$$\hat{p} = n_1 \;/\; n$$

where n_1 is the frequency of the first (or designated) level and n is the total frequency of the one-way table. The standard error of the binomial proportion is computed as

$$se(\hat{p}) = \sqrt{\hat{p}\left(1-\hat{p}\right) \;/\; n}$$

Confidence Limits

By default, PROC FREQ provides asymptotic and exact (Clopper-Pearson) confidence limits for the binomial proportion. If you do not specify any confidence limit requests with *binomial-options*, PROC FREQ computes the standard Wald asymptotic confidence limits. You can also request Agresti-Coull, Jeffreys, and Wilson (score) confidence limits for the binomial proportion. See Brown, Cai, and DasGupta (2001), Agresti and Coull (1998), and Newcombe (1998) for details about these binomial confidence limits, including comparisons of their performance.

Wald Confidence Limits The standard Wald asymptotic confidence limits are based on the normal approximation to the binomial distribution. PROC FREQ computes the Wald confidence limits for the binomial proportion as

$$\hat{p} \pm (z_{\alpha/2} \times \mathrm{se}(\hat{p}))$$

where $z_{\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the standard normal distribution. The confidence level α is determined by the ALPHA= option, which, by default, equals 0.05 and produces 95% confidence limits.

If you specify the CORRECT binomial-option or the BINOMIALC option, PROC FREQ includes a continuity correction of $1/2n$ in the Wald asymptotic confidence limits. The purpose of this correction is to adjust for the difference between the normal approximation and the binomial distribution, which is a discrete distribution. See Fleiss, Levin, and Paik (2003) for more information. With the continuity correction, the asymptotic confidence limits for the binomial proportion are computed as

$$\hat{p} \pm (z_{\alpha/2} \times \mathrm{se}(\hat{p}) + (1/2n))$$

Agresti-Coull Confidence Limits If you specify the AGRESTICOULL binomial-option, PROC FREQ computes Agresti-Coull confidence limits for the binomial proportion as

$$\tilde{p} \pm (z_{\alpha/2} \times \sqrt{\tilde{p}\,(1-\tilde{p}) \,/\, \tilde{n}})$$

where

$$\begin{aligned} \tilde{n}_1 &= n_1 + (z_{\alpha/2})/2 \\ \tilde{n} &= n + z^2_{\alpha/2} \\ \tilde{p} &= \tilde{n}_1 \,/\, \tilde{n} \end{aligned}$$

The Agresti-Coull confidence interval has the same basic form as the standard Wald interval but uses $\tilde{p}$ in place of $\hat{p}$. For $\alpha = 0.05$, the value of $z_{\alpha/2}$ is close to 2, and this interval is the "add 2 successes and 2 failures" adjusted Wald interval in Agresti and Coull (1998).

Jeffreys Confidence Limits If you specify the JEFFREYS binomial-option, PROC FREQ computes the Jeffreys confidence limits for the binomial proportion as

$$\left(\beta(\alpha/2,\ n_1 + 1/2,\ n - n_1 + 1/2),\ \ \beta(1-\alpha/2,\ n_1 + 1/2,\ n - n_1 + 1/2) \right)$$

where $\beta(\alpha, b, c)$ is the αth percentile of the beta distribution with shape parameters b and c. The lower confidence limit is set to 0 when $n_1 = 0$, and the upper confidence limit is set to 1 when $n_1 = n$. This is an equal-tailed interval based on the noninformative Jeffreys prior for a binomial proportion. See Brown, Cai, and DasGupta (2001) for details. See Berger (1985) for information about using beta priors for inference on the binomial proportion.

Wilson (Score) Confidence Limits If you specify the WILSON binomial-option, PROC FREQ computes Wilson confidence limits for the binomial proportion. These are also known as score confidence limits and are attributed to Wilson (1927). The confidence limits are based on inverting the normal test that uses the null proportion in the variance (the score test). Wilson confidence limits are the roots of

$$|p - \hat{p}| = z_{\alpha/2}\sqrt{p(1-p)/n}$$

and are computed as

$$\left(\hat{p} + z^2_{\alpha/2}/2n \pm z_{\alpha/2}\sqrt{\left(\hat{p}(1-\hat{p}) + z^2_{\alpha/2}/4n\right)/n}\right) / \left(1 + z^2_{\alpha/2}/n\right)$$

The Wilson interval has been shown to have better performance than the Wald interval and the exact (Clopper-Pearson) interval. See Agresti and Coull (1998), Brown, Cai, and DasGupta (2001), and Newcombe (1998) for more information.

Exact (Clopper-Pearson) Confidence Limits The exact or Clopper-Pearson confidence limits for the binomial proportion are constructed by inverting the equal-tailed test based on the binomial distribution. This method is attributed to Clopper and Pearson (1934). The exact confidence limits p_L and p_U satisfy the following equations, for $n_1 = 1, 2, \ldots n-1$:

$$\sum_{x=n_1}^{n} \binom{n}{x} p_L{}^x (1-p_L)^{n-x} = \alpha/2$$

$$\sum_{x=0}^{n_1} \binom{n}{x} p_U{}^x (1-p_U)^{n-x} = \alpha/2$$

The lower confidence limit equals 0 when $n_1 = 0$, and the upper confidence limit equals 1 when $n_1 = n$.

PROC FREQ computes the exact (Clopper-Pearson) confidence limits by using the F distribution as

$$p_L = \left(1 + \frac{n - n_1 + 1}{n_1\, F(1-\alpha/2,\ 2n_1,\ 2(n-n_1+1))}\right)^{-1}$$

$$p_U = \left(1 + \frac{n - n_1}{(n_1+1)\, F(\alpha/2,\ 2(n_1+1),\ 2(n-n_1))}\right)^{-1}$$

where $F(\alpha, b, c)$ is the αth percentile of the F distribution with b and c degrees of freedom. See Leemis and Trivedi (1996) for a derivation of this expression. Also see Collett (1991) for more information about exact binomial confidence limits.

Because this is a discrete problem, the confidence coefficient (or coverage probability) of the exact (Clopper-Pearson) interval is not exactly $(1-\alpha)$ but is at least $(1-\alpha)$. Thus, this confidence interval is conservative. Unless the sample size is large, the actual coverage probability can be much larger than the target value. See Agresti and Coull (1998), Brown, Cai, and DasGupta (2001), and Leemis and Trivedi (1996) for more information about the performance of these confidence limits.

Tests

The BINOMIAL option provides an asymptotic equality test for the binomial proportion by default. You can also specify *binomial-options* to request tests of noninferiority, superiority, and equivalence for the binomial proportion. If you specify the BINOMIAL option in the EXACT statement, PROC FREQ also computes exact p-values for the tests that you request with the *binomial-options*.

Equality Test PROC FREQ computes an asymptotic test of the hypothesis that the binomial proportion equals p_0, where you can specify the value of p_0 with the P= binomial-option. If you do not specify a null value with P=, PROC FREQ uses $p_0 = 0.5$ by default. The binomial test statistic is computed as

$$z = (\hat{p} - p_0)/se$$

By default, the standard error is based on the null hypothesis proportion as

$$se = \sqrt{p_0(1-p_0)/n}$$

If you specify the VAR=SAMPLE binomial-option, the standard error is computed from the sample proportion as

$$se = \sqrt{\hat{p}(1-\hat{p})/n}$$

If you specify the CORRECT binomial-option or the BINOMIALC option, PROC FREQ includes a continuity correction in the asymptotic test statistic, towards adjusting for the difference between the normal approximation and the discrete binomial distribution. See Fleiss, Levin, and Paik (2003) for details. The continuity correction of $(1/2n)$ is subtracted from the numerator of the test statistic if $(\hat{p} - p_0)$ is positive; otherwise, the continuity correction is added to the numerator.

PROC FREQ computes one-sided and two-sided p-values for this test. When the test statistic z is greater than zero (its expected value under the null hypothesis), PROC FREQ computes the right-sided p-value, which is the probability of a larger value of the statistic occurring under the null hypothesis. A small right-sided p-value supports the alternative hypothesis that the true value of the proportion is greater than p_0. When the test statistic is less than or equal to zero, PROC FREQ computes the left-sided p-value, which is the probability of a smaller value of the statistic occurring under the null hypothesis. A small left-sided p-value supports the alternative hypothesis that the true value of the proportion is less than p_0. The one-sided p-value P_1 can be expressed as

$$P_1 = \begin{cases} \text{Prob}(Z > z) & \text{if } z > 0 \\ \text{Prob}(Z < z) & \text{if } z \leq 0 \end{cases}$$

where Z has a standard normal distribution. The two-sided p-value is computed as $P_2 = 2 \times P_1$.

If you specify the BINOMIAL option in the EXACT statement, PROC FREQ also computes an exact test of the null hypothesis H_0: $p = p_0$. To compute the exact test, PROC FREQ uses the binomial probability function,

$$\text{Prob}(X = x \mid p_0) = \binom{n}{x} {p_0}^x \, (1 - p_0)^{(n-x)} \quad \text{for} \quad x = 0, 1, 2, \ldots, n$$

where the variable X has a binomial distribution with parameters n and p_0. To compute the left-sided p-value, $\text{Prob}(X \leq n_1)$, PROC FREQ sums the binomial probabilities over x from zero to n_1. To compute the right-sided p-value, $\text{Prob}(X \geq n_1)$, PROC FREQ sums the binomial probabilities over x from n_1 to n. The exact one-sided p-value is the minimum of the left-sided and right-sided p-values,

$$P_1 = \min \left(\text{Prob}(X \leq n_1 \mid p_0), \ \text{Prob}(X \geq n_1 \mid p_0) \right)$$

and the exact two-sided p-value is computed as $P_2 = 2 \times P_1$.

Noninferiority Test If you specify the NONINF binomial-option, PROC FREQ provides a noninferiority test for the binomial proportion. The null hypothesis for the noninferiority test is

$$H_0: p - p_0 \leq -\delta$$

versus the alternative

$$H_a: p - p_0 > -\delta$$

where δ is the noninferiority margin and p_0 is the null proportion. Rejection of the null hypothesis indicates that the binomial proportion is not inferior to the null value. See Chow, Shao, and Wang (2003) for more information.

You can specify the value of δ with the MARGIN= binomial-option, and you can specify p_0 with the P= binomial-option. By default, $\delta = 0.2$ and $p_0 = 0.5$.

PROC FREQ provides an asymptotic Wald test for noninferiority. The test statistic is computed as

$$z = (\hat{p} - p_0^*) \ / \ se$$

where p_0^* is the noninferiority limit,

$$p_0^* = p_0 - \delta$$

By default, the standard error is computed from the sample proportion as

$$se = \sqrt{\hat{p}(1 - \hat{p})/n}$$

If you specify the VAR=NULL binomial-option, the standard error is based on the noninferiority limit (determined by the null proportion and the margin) as

$$se = \sqrt{p_0^*(1 - p_0^*)/n}$$

If you specify the CORRECT binomial-option or the BINOMIALC option, PROC FREQ includes a continuity correction in the asymptotic test statistic z. The continuity correction of $(1/2n)$ is subtracted from the numerator of the test statistic if $(\hat{p} - p_0^*)$ is positive; otherwise, the continuity correction is added to the numerator.

The p-value for the noninferiority test is

$$P_z = \text{Prob}(Z > z)$$

where Z has a standard normal distribution.

As part of the noninferiority analysis, PROC FREQ provides asymptotic Wald confidence limits for the binomial proportion. These confidence limits are computed as described in the section "Wald Confidence Limits" on page 1746 but use the same standard error (VAR=NULL or VAR=SAMPLE) as the noninferiority test statistic z. The confidence coefficient is $100(1-2\alpha)\%$ (Schuirmann 1999). By default, if you do not specify the ALPHA= option, the noninferiority confidence limits are 90% confidence limits. You can compare the confidence limits to the noninferiority limit, $p_0^* = p_0 - \delta$.

If you specify the BINOMIAL option in the EXACT statement, PROC FREQ provides an exact noninferiority test for the binomial proportion. The exact p-value is computed by using the binomial probability function with parameters p_0^* and n,

$$P_x = \sum_{k=n_1}^{k=n} \binom{n}{k} (p_0^*)^{\,k} \, (1 - p_0^*)^{\,(n-k)}$$

See Chow, Shao, Wang (2003, p. 116) for details. If you request exact binomial statistics, PROC FREQ also includes exact (Clopper-Pearson) confidence limits for the binomial proportion in the equivalence analysis display. See the section "Exact (Clopper-Pearson) Confidence Limits" on page 1747 for details.

Superiority Test If you specify the SUP binomial-option, PROC FREQ provides a superiority test for the binomial proportion. The null hypothesis for the superiority test is

$$H_0\colon p - p_0 \leq \delta$$

versus the alternative

$$H_a\colon p - p_0 > \delta$$

where δ is the superiority margin and p_0 is the null proportion. Rejection of the null hypothesis indicates that the binomial proportion is superior to the null value. You can specify the value of δ with the MARGIN= binomial-option, and you can specify the value of p_0 with the P= binomial-option. By default, $\delta = 0.2$ and $p_0 = 0.5$.

The superiority analysis is identical to the noninferiority analysis but uses a positive value of the margin δ in the null hypothesis. The superiority limit equals $p_0 + \delta$. The superiority computations follow those in the section "Noninferiority Test" on page 1749 but replace $-\delta$ with δ. See Chow, Shao, and Wang (2003) for more information.

Equivalence Test If you specify the EQUIV binomial-option, PROC FREQ provides an equivalence test for the binomial proportion. The null hypothesis for the equivalence test is

$$H_0: p - p_0 \leq \delta_L \quad \text{or} \quad p - p_0 \geq \delta_U$$

versus the alternative

$$H_a: \delta_L < p - p_0 < \delta_U$$

where δ_L is the lower margin, δ_U is the upper margin, and p_0 is the null proportion. Rejection of the null hypothesis indicates that the binomial proportion is equivalent to the null value. See Chow, Shao, and Wang (2003) for more information.

You can specify the value of the margins δ_L and δ_U with the MARGIN= binomial-option. If you do not specify MARGIN=, PROC FREQ uses lower and upper margins of –0.2 and 0.2 by default. If you specify a single margin value δ, PROC FREQ uses lower and upper margins of $-\delta$ and δ. You can specify the null proportion p_0 with the P= binomial-option. By default, $p_0 = 0.5$.

PROC FREQ computes two one-sided tests (TOST) for equivalence analysis (Schuirmann 1987). The TOST approach includes a right-sided test for the lower margin and a left-sided test for the upper margin. The overall p-value is taken to be the larger of the two p-values from the lower and upper tests.

For the lower margin, the asymptotic Wald test statistic is computed as

$$z_L = (\hat{p} - p_L^*) \,/\, se$$

where the lower equivalence limit is

$$p_L^* = p_0 + \delta_L$$

By default, the standard error is computed from the sample proportion as

$$se = \sqrt{\hat{p}(1-\hat{p})/n}$$

If you specify the VAR=NULL binomial-option, the standard error is based on the lower equivalence limit (determined by the null proportion and the lower margin) as

$$se = \sqrt{p_L^*(1-p_L^*)/n}$$

If you specify the CORRECT binomial-option or the BINOMIALC option, PROC FREQ includes a continuity correction in the asymptotic test statistic z_L. The continuity correction of $(1/2n)$ is subtracted from the numerator of the test statistic $(\hat{p} - p_L^*)$ if the numerator is positive; otherwise, the continuity correction is added to the numerator.

The p-value for the lower margin test is

$$P_{z,L} = \text{Prob}(Z > z_L)$$

The asymptotic test for the upper margin is computed similarly. The Wald test statistic is

$$z_U = (\hat{p} - p_U^*) \,/\, se$$

where the upper equivalence limit is

$$p_U^* = p_0 + \delta_U$$

By default, the standard error is computed from the sample proportion. If you specify the VAR=NULL binomial-option, the standard error is based on the upper equivalence limit as

$$se = \sqrt{p_U^*(1 - p_U^*)/n}$$

If you specify the CORRECT binomial-option or the BINOMIALC option, PROC FREQ includes a continuity correction of $(1/2n)$ in the asymptotic test statistic z_U.

The p-value for the upper margin test is

$$P_{z,U} = \text{Prob}(Z < z_U)$$

Based on the two one-sided tests (TOST), the overall p-value for the test of equivalence equals the larger p-value from the lower and upper margin tests, which can be expressed as

$$P_z = \max(P_{z,L}\ ,\ P_{z,U})$$

As part of the equivalence analysis, PROC FREQ provides asymptotic Wald confidence limits for the binomial proportion. These confidence limits are computed as described in the section "Wald Confidence Limits" on page 1746, but use the same standard error (VAR=NULL or VAR=SAMPLE) as the equivalence test statistics and have a confidence coefficient of $100(1-2\alpha)\%$ (Schuirmann 1999). By default, if you do not specify the ALPHA= option, the equivalence confidence limits are 90% limits. If you specify VAR=NULL, separate standard errors are computed for the lower and upper margin tests, each based on the null proportion and the corresponding (lower or upper) margin. The confidence limits are computed by using the maximum of these two standard errors. You can compare the confidence limits to the equivalence limits, $(p_0 + \delta_L,\ p_0 + \delta_U)$.

If you specify the BINOMIAL option in the EXACT statement, PROC FREQ also provides an exact equivalence test by using two one-sided exact tests (TOST). The procedure computes lower and upper margin exact tests by using the binomial probability function as described in the section "Noninferiority Test" on page 1749. The overall exact p-value for the equivalence test is taken to be the larger p-value from the lower and upper margin exact tests. If you request exact statistics, PROC FREQ also includes exact (Clopper-Pearson) confidence limits in the equivalence analysis display. The confidence coefficient is $100(1 - 2\alpha)\%$ (Schuirmann 1999). See the section "Exact (Clopper-Pearson) Confidence Limits" on page 1747 for details.

Risks and Risk Differences

The RISKDIFF option in the TABLES statement provides estimates of risks (or binomial proportions) and risk differences for 2×2 tables. This analysis might be appropriate when comparing the proportion of some characteristic for two groups, where row 1 and row 2 correspond to the two groups, and the columns correspond to two possible characteristics or outcomes. For example, the row variable might be a treatment or dose, and the column variable might be the response. See Collett (1991), Fleiss, Levin, and Paik (2003), and Stokes, Davis, and Koch (2000) for more information.

Let the frequencies of the 2×2 table be represented as follows.

	Column 1	Column 2	Total
Row 1	n_{11}	n_{12}	$n_{1\cdot}$
Row 2	n_{21}	n_{22}	$n_{2\cdot}$
Total	$n_{\cdot 1}$	$n_{\cdot 2}$	n

For column 1 and column 2, PROC FREQ provides estimates of the row 1 risk (or proportion), the row 2 risk, the overall risk and the risk difference. The risk difference is defined as the row 1 risk minus the row 2 risk. The risks are binomial proportions of their rows (row 1, row 2, or overall), and the computation of their standard errors and confidence limits follow the binomial proportion computations, which are described in the section "Binomial Proportion" on page 1745.

The column 1 risk for row 1 is the proportion of row 1 observations classified in column 1,

$$p_1 = n_{11} \; / \; n_{1\cdot}$$

This estimates the conditional probability of the column 1 response, given the first level of the row variable.

The column 1 risk for row 2 is the proportion of row 2 observations classified in column 1,

$$p_2 = n_{21} \; / \; n_{2\cdot}$$

and the overall column 1 risk is the proportion of all observations classified in column 1,

$$p = n_{\cdot 1} \; / \; n$$

The column 1 risk difference compares the risks for the two rows, and it is computed as the column 1 risk for row 1 minus the column 1 risk for row 2,

$$d = p_1 - p_2$$

The risks and risk difference are defined similarly for column 2.

The standard error of the column 1 risk for row i is computed as

$$se(p_i) = \sqrt{p_i \; (1 - p_i) \; / \; n_{1\cdot}}$$

The standard error of the overall column 1 risk is computed as

$$se(p) = \sqrt{p \; (1 - p) \; / \; n}$$

If the two rows represent independent binomial samples, the standard error for the column 1 risk difference is computed as

$$se(d) = \sqrt{var(p_1) + var(p_2)}$$

The standard errors are computed in a similar manner for the column 2 risks and risk difference.

Confidence Limits

By default, the RISKDIFF option provides standard Wald asymptotic confidence limits for the risks (row 1, row 2, and overall) and the risk difference. The risks are equivalent to binomial proportions of their corresponding rows, and the computations follow the methods in the section "Wald Confidence Limits" on page 1746.

The standard Wald asymptotic confidence limits are based on the normal approximation to the binomial distribution. PROC FREQ computes the Wald confidence limits for the risks and risk differences as

$$est \pm (z_{\alpha/2} \times \mathrm{se}(est))$$

where est is the estimate, $z_{\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the standard normal distribution, and $\mathrm{se}(est)$ is the standard error of the estimate. The confidence level α is determined from the value of the ALPHA= option, which, by default, equals 0.05 and produces 95% confidence limits.

If you specify the CORRECT *riskdiff-option* or the RISKDIFFC option, PROC FREQ includes continuity corrections in the Wald asymptotic confidence limits for the risks and risk differences. The purpose of a continuity correction is to adjust for the difference between the normal approximation and the binomial distribution, which is discrete. See Fleiss, Levin, and Paik (2003) for more information. With the continuity correction, the asymptotic confidence limits are computed as

$$est \pm (z_{\alpha/2} \times \mathrm{se}(est) + cc)$$

where cc is the continuity correction. For the row 1 risk, $cc = (1/2n_{1\cdot})$; for the row 2 risk, $cc = (1/2n_{2\cdot})$; for the overall risk, $cc = (1/2n)$; and for the risk difference, $cc = ((1/n_{1\cdot} + 1/n_{2\cdot})/2)$. The column 1 and column 2 risks use the same continuity corrections.

PROC FREQ also computes exact (Clopper-Pearson) confidence limits for the column 1, column 2, and overall risks. These confidence limits are constructed by inverting the equal-tailed test based on the binomial distribution. PROC FREQ uses the F distribution to compute the Clopper-Pearson confidence limits. See the section "Exact (Clopper-Pearson) Confidence Limits" on page 1747 for details.

PROC FREQ does not provide exact confidence limits for the risk difference by default. If you specify the RISKDIFF option in the EXACT statement, PROC FREQ provides exact unconditional confidence limits for the risk difference, which are described in the section "Exact Unconditional Confidence Limits for the Risk Difference" on page 1759. Note that the conditional exact approach, which is the basis for other exact tests provided by PROC FREQ such as Fisher's exact test, does not apply to the risk difference due to nuisance parameters. See Agresti (1992) for more information.

Tests

You can specify *riskdiff-options* to request tests of the risk (or proportion) difference. You can request tests of equality, noninferiority, superiority, and equivalence for the risk difference. The test of equality is a standard Wald asymptotic test, available with or without a continuity correction. For noninferiority, superiority, and equivalence tests of the risk difference, the following test methods are provided: Wald (with and without continuity correction), Hauck-Anderson, Farrington-Manning, and Newcombe score (with and without continuity correction). You can specify the test method with the METHOD= riskdiff-option. By default, PROC FREQ uses METHOD=WALD.

Equality Test If you specify the EQUAL riskdiff-option, PROC FREQ computes a test of equality, or a test of the null hypothesis that the risk difference equals zero. For the column 1 (or 2) risk difference, this test can be expressed as $H_0: d = 0$ versus the alternative $H_a: d \neq 0$, where $d = p_1 - p_2$ denotes the column 1 (or 2) risk difference. PROC FREQ computes a standard Wald asymptotic test, and the test statistic is

$$z = \hat{d}/\mathrm{se}(\hat{d})$$

By default, the standard error is computed from the sample proportions as

$$\mathrm{se}(\hat{d}) = \sqrt{\hat{p}_1(1-\hat{p}_1)/n_{1\cdot} + \hat{p}_2(1-\hat{p}_2)/n_{2\cdot}}$$

If you specify the VAR=NULL riskdiff-option, the standard error is based on the null hypothesis that the row 1 and row 2 risks are equal,

$$\mathrm{se}(\hat{d}) = \sqrt{\hat{p}(1-\hat{p}) \times (1/n_{1\cdot} + 1/n_{2\cdot})}$$

where $\hat{p} = n_{\cdot 1}/n$ estimates the overall column 1 risk.

If you specify the CORRECT riskdiff-option or the RISKDIFFC option, PROC FREQ includes a continuity correction in the asymptotic test statistic. If $\hat{d} > 0$, the continuity correction is subtracted from $\hat{d}$ in the numerator of the test statistic; otherwise, the continuity correction is added to the numerator. The value of the continuity correction is $(1/n_{1\cdot} + 1/n_{2\cdot})/2$.

PROC FREQ computes one-sided and two-sided p-values for this test. When the test statistic z is greater than 0, PROC FREQ displays the right-sided p-value, which is the probability of a larger value occurring under the null hypothesis. The one-sided p-value can be expressed as

$$P_1 = \begin{cases} \mathrm{Prob}(Z > z) & \text{if } z > 0 \\ \mathrm{Prob}(Z < z) & \text{if } z \leq 0 \end{cases}$$

where Z has a standard normal distribution. The two-sided p-value is computed as $P_2 = 2 \times P_1$.

Noninferiority Test If you specify the NONINF riskdiff-option, PROC FREQ provides a noninferiority test for the risk difference, or the difference between two proportions. The null hypothesis for the noninferiority test is

$$H_0: p_1 - p_2 \leq -\delta$$

versus the alternative

$$H_a: p_1 - p_2 > -\delta$$

where δ is the noninferiority margin. Rejection of the null hypothesis indicates that the row 1 risk is not inferior to the row 2 risk. See Chow, Shao, and Wang (2003) for more information.

You can specify the value of δ with the MARGIN= riskdiff-option. By default, $\delta = 0.2$. You can specify the test method with the METHOD= riskdiff-option. The following methods are available for the risk difference noninferiority analysis: Wald (with and without continuity correction), Hauck-Anderson, Farrington-Manning, and Newcombe score (with and without continuity correction). The Wald, Hauck-Anderson, and Farrington-Manning methods provide tests and corresponding test-based confidence limits; the Newcombe score method provides only confidence limits. If you do not specify METHOD=, PROC FREQ uses the Wald test by default.

The confidence coefficient for the test-based confidence limits is $100(1-2\alpha)\%$ (Schuirmann 1999). By default, if you do not specify the ALPHA= option, these are 90% confidence limits. You can compare the confidence limits to the noninferiority limit, $-\delta$.

The following sections describe the noninferiority analysis methods for the risk difference.

Wald Test

If you specify the METHOD=WALD riskdiff-option, PROC FREQ provides an asymptotic Wald test of noninferiority for the risk difference. This is also the default method. The Wald test statistic is computed as

$$z = (\hat{d} + \delta) \,/\, \mathrm{se}(\hat{d})$$

where $(\hat{d} = \hat{p}_1 - \hat{p}_2)$ estimates the risk difference and δ is the noninferiority margin.

By default, the standard error for the Wald test is computed from the sample proportions as

$$\mathrm{se}(\hat{d}) = \sqrt{\hat{p}_1(1-\hat{p}_1)/n_{1\cdot} \;+\; \hat{p}_2(1-\hat{p}_2)/n_{2\cdot}}$$

If you specify the VAR=NULL riskdiff-option, the standard error is based on the null hypothesis that the risk difference equals $-\delta$ (Dunnett and Gent 1977). The standard error is computed as

$$\mathrm{se}(\hat{d}) = \sqrt{\tilde{p}(1-\tilde{p})/n_{2\cdot} \;+\; (\tilde{p}-\delta)(1-\tilde{p}+\delta)/n_{1\cdot}}$$

where

$$\tilde{p} = (n_{11} + n_{21} + \delta n_{1\cdot})/n$$

If you specify the CORRECT riskdiff-option or the RISKDIFFC option, a continuity correction is included in the test statistic. The continuity correction is subtracted from the numerator of the test statistic if the numerator is greater than zero; otherwise, the continuity correction is added to the numerator. The value of the continuity correction is $(1/n_{1\cdot} + 1/n_{2\cdot})/2$.

The p-value for the Wald noninferiority test is $P_z = \mathrm{Prob}(Z > z)$, where Z has a standard normal distribution.

Hauck-Anderson Test

If you specify the METHOD=HA riskdiff-option, PROC FREQ provides the Hauck-Anderson test for noninferiority. The Hauck-Anderson test statistic is computed as

$$z = (\hat{d} + \delta \;\pm\; cc) \,/\, \mathrm{s}(\hat{d})$$

where $\hat{d} = \hat{p}_1 - \hat{p}_2$ and the standard error is computed from the sample proportions as

$$\mathrm{se}(\hat{d}) = \sqrt{\hat{p}_1(1-\hat{p}_1)/(n_{1\cdot}-1) \;+\; \hat{p}_2(1-\hat{p}_2)/(n_{2\cdot}-1)}$$

The Hauck-Anderson continuity correction cc is computed as

$$cc = 1 \,/\, \left(2\, \min(n_{1\cdot},\, n_{2\cdot})\right)$$

The p-value for the Hauck-Anderson noninferiority test is $P_z = \mathrm{Prob}(Z > z)$, where Z has a standard normal distribution. See Hauck and Anderson (1986) and Schuirmann (1999) for more information.

Farrington-Manning Test

If you specify the METHOD=FM riskdiff-option, PROC FREQ provides the Farrington-Manning test of noninferiority for the risk difference. The Farrington-Manning test statistic is computed as

$$z = (\hat{d} + \delta) \,/\, \mathrm{se}(\hat{d})$$

where $\hat{d} = \hat{p}_1 - \hat{p}_2$ and

$$\mathrm{se}(\hat{d}) = \sqrt{\tilde{p}_1(1-\tilde{p}_1)/n_{1\cdot} \;+\; \tilde{p}_2(1-\tilde{p}_2)/n_{2\cdot}}$$

where $\tilde{p}_1$ and $\tilde{p}_2$ are the maximum likelihood estimators of p_1 and p_2 under the null hypothesis that the risk difference equals $-\delta$. The p-value for the Farrington-Manning noninferiority test is then $P_z = \mathrm{Prob}(Z > z)$, where Z has a standard normal distribution.

From Farrington and Manning (1990), the solution to the maximum likelihood equation is

$$\tilde{p}_1 = 2u\cos(w) - b/3a \quad \text{and} \quad \tilde{p}_2 = \tilde{p}_1 + \delta$$

where

$$\begin{aligned}
w &= (\pi + \cos^{-1}(v/u^3))/3 \\
v &= b^3/(3a)^3 - bc/6a^2 + d/2a \\
u &= \mathrm{sign}(v)\sqrt{b^2/(3a)^2 - c/3a} \\
a &= 1 + \theta \\
b &= -(1 + \theta + \hat{p}_1 + \theta\hat{p}_2 - \delta(\theta + 2)) \\
c &= \delta^2 - \delta(2\hat{p}_1 + \theta + 1) + \hat{p}_1 + \theta\hat{p}_2 \\
d &= \hat{p}_1\delta(1-\delta) \\
\theta &= n_{2\cdot}/n_{1\cdot}
\end{aligned}$$

Newcombe Score Confidence Limits

If you specify the METHOD=SCORE riskdiff-option, PROC FREQ provides the Newcombe hybrid score (or Wilson) confidence limits for the risk difference. The confidence coefficient for the confidence limits is $100(1-2\alpha)\%$ (Schuirmann 1999). By default, if you do not specify the ALPHA= option, these are 90% confidence limits. You can compare the confidence limits to the noninferiority limit, $-\delta$.

The Newcombe score confidence limits for the risk difference are constructed from the Wilson score confidence limits for each of the two individual proportions. The confidence limits for the individual proportions are used in the standard error terms of the Wald confidence limits for the proportion difference. See Newcombe (1998) and Barker et al. (2001) for more information.

Wilson score confidence limits for p_1 and p_2 are the roots of

$$|p_i - \hat{p}_i| = z_\alpha\sqrt{p_i(1-p_i)/n_{i\cdot}}$$

for $i = 1, 2$. The confidence limits are computed as

$$\left(\hat{p}_i \;+\; z_\alpha^2/2n_{i\cdot} \;\pm\; z_\alpha\sqrt{\left(\hat{p}_i(1-\hat{p}_i) + z_\alpha^2/4n_{i\cdot}\right)/n_{i\cdot}}\;\right) \;/\; \left(1 + z_\alpha^2/n_{i\cdot}\right)$$

See the section "Wilson (Score) Confidence Limits" on page 1747 for details.

Denote the lower and upper Wilson score confidence limits for p_1 as L_1 and U_1, and denote the lower and upper confidence limits for p_2 as L_2 and U_2. The Newcombe score confidence limits for the proportion difference ($d = p_1 - p_2$) are computed as

$$d_L = (\hat{p}_1 - \hat{p}_2) - z_\alpha \sqrt{L_1(1-L_1)/n_{1\cdot} + U_2(1-U_2)/n_{2\cdot}}$$

$$d_U = (\hat{p}_1 - \hat{p}_2) + z_\alpha \sqrt{U_1(1-U_1)/n_{1\cdot} + L_2(1-L_2)/n_{2\cdot}}$$

If you specify the CORRECT riskdiff-option, PROC FREQ provides continuity-corrected Newcombe score confidence limits. By including a continuity correction of $1/2n_{i\cdot}$, the Wilson score confidence limits for the individual proportions are the roots of

$$|p_i - \hat{p}_i| - 1/2n_{i\cdot} = z_\alpha \sqrt{p_i(1-p_i)/n_{i\cdot}}$$

These confidence limits for the individual proportions are then used in the standard error terms of the Wald confidence limits for the proportion difference to compute d_L and d_U.

Superiority Test If you specify the SUP riskdiff-option, PROC FREQ provides a superiority test for the risk difference. The null hypothesis is

$$H_0:: p_1 - p_2 \leq \delta$$

versus the alternative

$$H_a: p_1 - p_2 > \delta$$

where δ is the superiority margin. Rejection of the null hypothesis indicates that the row 1 proportion is superior to the row 2 proportion. You can specify the value of δ with the MARGIN= riskdiff-option. By default, $\delta = 0.2$.

The superiority analysis is identical to the noninferiority analysis but uses a positive value of the margin δ in the null hypothesis. The superiority computations follow those in the section "Noninferiority Test" on page 1755 by replacing $-\delta$ by δ. See Chow, Shao, and Wang (2003) for more information.

Equivalence Tests If you specify the EQUIV riskdiff-option, PROC FREQ provides an equivalence test for the risk difference, or the difference between two proportions. The null hypothesis for the equivalence test is

$$H_0: p_1 - p_2 \leq -\delta_L \quad \text{or} \quad p_1 - p_2 \geq \delta_U$$

versus the alternative

$$H_a: \delta_L < p_1 - p_2 < \delta_U$$

where δ_L is the lower margin and δ_U is the upper margin. Rejection of the null hypothesis indicates that the two binomial proportions are equivalent. See Chow, Shao, and Wang (2003) for more information.

You can specify the value of the margins δ_L and δ_U with the MARGIN= riskdiff-option. If you do not specify MARGIN=, PROC FREQ uses lower and upper margins of –0.2 and 0.2 by default. If you specify a single margin value δ, PROC FREQ uses lower and upper margins of $-\delta$ and δ. You can specify the test method with the METHOD= riskdiff-option. The following methods are available for the risk difference equivalence analysis: Wald (with and without continuity correction), Hauck-Anderson, Farrington-Manning, and Newcombe's score (with and without continuity correction). The Wald, Hauck-Anderson, and Farrington-Manning methods provide tests and corresponding test-based confidence limits; the Newcombe score method provides only confidence limits. If you do not specify METHOD=, PROC FREQ uses the Wald test by default.

PROC FREQ computes two one-sided tests (TOST) for equivalence analysis (Schuirmann 1987). The TOST approach includes a right-sided test for the lower margin δ_L and a left-sided test for the upper margin δ_U. The overall p-value is taken to be the larger of the two p-values from the lower and upper tests.

The section "Noninferiority Test" on page 1755 gives details about the Wald, Hauck-Anderson, Farrington-Manning and Newcombe score methods for the risk difference. The lower margin equivalence test statistic takes the same form as the noninferiority test statistic but uses the lower margin value δ_L in place of $-\delta$. The upper margin equivalence test statistic take the same form as the noninferiority test statistic but uses the upper margin value δ_U in place of $-\delta$.

The test-based confidence limits for the risk difference are computed according to the equivalence test method that you select. If you specify METHOD=WALD with VAR=NULL, or METHOD=FM, separate standard errors are computed for the lower and upper margin tests. In this case, the test-based confidence limits are computed by using the maximum of these two standard errors. The confidence limits have a confidence coefficient of $100(1-2\alpha)\%$ (Schuirmann 1999). By default, if you do not specify the ALPHA= option, these are 90% confidence limits. You can compare the confidence limits to the equivalence limits, (δ_L, δ_U).

Exact Unconditional Confidence Limits for the Risk Difference

If you specify the RISKDIFF option in the EXACT statement, PROC FREQ provides exact unconditional confidence limits for the risk difference. Unconditional computations differ from the exact conditional approach that PROC FREQ uses for other exact statistics such as Fisher's exact test. (See the section "Exact Statistics" on page 1779 for more information.) Exact conditional inference does not apply to the risk difference due to nuisance parameters. See Agresti (1992) for details. The unconditional approach eliminates nuisance parameters by maximizing the p-value over all possible values of the nuisance parameters (Santner and Snell 1980).

Denote the proportion difference by $d = p_1 - p_2$. For a 2×2 table with row totals n_1 and n_2, the joint probability function can be expressed in terms of the table cell frequencies and the parameters d and p_2,

$$f(n_{11}, n_{21}; n_1, n_2, d, p_2) = \binom{n_1}{n_{11}} (d+p_2)^{n}_{11} (1-d-p_2)^{n_1-n_{11}} \times \binom{n_2}{n_{21}} p_2^{n_{21}} (1-p_2)^{n_2-n_{21}}$$

When constructing confidence limits for the proportion difference, the parameter of interest is d and p_2 is a nuisance parameter.

Denote the observed value of the proportion difference by $d_0 = \hat{p}_1 - \hat{p}_2$. The $100(1-\alpha/2)\%$ confidence limits for d are computed as

$$\begin{aligned} d_L &= \sup\left(d_* : P_U(d_*) > \alpha/2\right) \\ d_U &= \inf\left(d_* : P_L(d_*) > \alpha/2\right) \end{aligned}$$

where

$$P_U(d_*) = \sup_{p_2}\Big(\sum_{A, D(a)\geq d_0} f(n_{11}, n_{21}; n_1, n_2, d_*, p_2)\Big)$$

$$P_L(d_*) = \sup_{p_2}\Big(\sum_{A, D(a)\leq d_0} f(n_{11}, n_{21}; n_1, n_2, d_*, p_2)\Big)$$

The set A includes all 2×2 tables with row sums equal to n_1 and n_2, and $D(a)$ denotes the value of the proportion difference $(p_1 - p_2)$ for table a in A. To compute $P_U(d_*)$, the sum includes probabilities of those tables for which $(D(a) \geq d_0)$, where d_0 is the observed value of the proportion difference. For a fixed value of d_*, $P_U(d_*)$ is taken to be the maximum sum over all possible values of p_2. See Santner and Snell (1980) and Agresti and Min (2001) for details.

This method of eliminating the nuisance parameter is considered to be a conservative approach (Agresti and Min 2001). Additionally, the confidence limits are conservative for small samples because this is a discrete problem; the confidence coefficient is not exactly $(1-\alpha)$ but is at least $(1-\alpha)$. See Agresti (1992) for more information.

Odds Ratio and Relative Risks for 2 x 2 Tables

Odds Ratio (Case-Control Studies)

The odds ratio is a useful measure of association for a variety of study designs. For a retrospective design called a *case-control study*, the odds ratio can be used to estimate the relative risk when the probability of positive response is small (Agresti 2002). In a case-control study, two independent samples are identified based on a binary (yes-no) response variable, and the conditional distribution of a binary explanatory variable is examined, within fixed levels of the response variable. See Stokes, Davis, and Koch (2000) and Agresti (2007).

The odds of a positive response (column 1) in row 1 is n_{11}/n_{12}. Similarly, the odds of a positive response in row 2 is n_{21}/n_{22}. The odds ratio is formed as the ratio of the row 1 odds to the row 2 odds. The odds ratio for a 2×2 table is defined as

$$OR = \frac{n_{11}/n_{12}}{n_{21}/n_{22}} = \frac{n_{11}\, n_{22}}{n_{12}\, n_{21}}$$

The odds ratio can be any nonnegative number. When the row and column variables are independent, the true value of the odds ratio equals 1. An odds ratio greater than 1 indicates that the odds of a positive response are higher in row 1 than in row 2. Values less than 1 indicate the odds of positive response are higher in row 2. The strength of association increases with the deviation from 1.

The transformation $G = (OR - 1)/(OR + 1)$ transforms the odds ratio to the range $(-1, 1)$ with $G = 0$ when $OR = 1$; $G = -1$ when $OR = 0$; and G approaches 1 as OR approaches infinity. G is the gamma statistic, which PROC FREQ computes when you specify the MEASURES option.

The asymptotic $100(1 - \alpha)\%$ confidence limits for the odds ratio are

$$\left(OR \times \exp(-z\sqrt{v}),\ \ OR \times \exp(z\sqrt{v}) \right)$$

where

$$v = \text{var}(\ln OR) = \frac{1}{n_{11}} + \frac{1}{n_{12}} + \frac{1}{n_{21}} + \frac{1}{n_{22}}$$

and z is the $100(1 - \alpha/2)$th percentile of the standard normal distribution. If any of the four cell frequencies are zero, the estimates are not computed.

When you specify the OR option in the EXACT statement, PROC FREQ computes exact confidence limits for the odds ratio. Because this is a discrete problem, the confidence coefficient for the exact confidence interval is not exactly $(1 - \alpha)$ but is at least $(1 - \alpha)$. Thus, these confidence limits are conservative. See Agresti (1992) for more information.

PROC FREQ computes exact confidence limits for the odds ratio by using an algorithm based on Thomas (1971). See also Gart (1971). The following two equations are solved iteratively to determine the lower and upper confidence limits, ϕ_1 and ϕ_2:

$$\sum_{i=n_{11}}^{n_{\cdot 1}} \binom{n_{1\cdot}}{i} \binom{n_{2\cdot}}{n_{\cdot 1} - i} \phi_1^i \ / \ \sum_{i=0}^{n_{\cdot 1}} \binom{n_{1\cdot}}{i} \binom{n_{2\cdot}}{n_{\cdot 1} - i} \phi_1^i \quad = \quad \alpha/2$$

$$\sum_{i=0}^{n_{11}} \binom{n_{1\cdot}}{i} \binom{n_{2\cdot}}{n_{\cdot 1} - i} \phi_2^i \ / \ \sum_{i=0}^{n_{\cdot 1}} \binom{n_{1\cdot}}{i} \binom{n_{2\cdot}}{n_{\cdot 1} - i} \phi_2^i \quad = \quad \alpha/2$$

When the odds ratio equals zero, which occurs when either $n_{11} = 0$ or $n_{22} = 0$, PROC FREQ sets the lower exact confidence limit to zero and determines the upper limit with level α. Similarly, when the odds ratio equals infinity, which occurs when either $n_{12} = 0$ or $n_{21} = 0$, PROC FREQ sets the upper exact confidence limit to infinity and determines the lower limit with level α.

Relative Risks (Cohort Studies)

These measures of relative risk are useful in *cohort* (prospective) study designs, where two samples are identified based on the presence or absence of an explanatory factor. The two samples are observed in future time for the binary (yes-no) response variable under study. Relative risk measures are also useful in cross-sectional studies, where two variables are observed simultaneously. See Stokes, Davis, and Koch (2000) and Agresti (2007) for more information.

The column 1 relative risk is the ratio of the column 1 risk for row 1 to row 2. The column 1 risk for row 1 is the proportion of the row 1 observations classified in column 1,

$$p_{1|1} = n_{11} \ / \ n_{1\cdot}$$

Similarly, the column 1 risk for row 2 is

$$p_{1|2} = n_{21} \ / \ n_{2\cdot}$$

The column 1 relative risk is then computed as

$$RR_1 = p_{1|1} \; / \; p_{1|2}$$

A relative risk greater than 1 indicates that the probability of positive response is greater in row 1 than in row 2. Similarly, a relative risk less than 1 indicates that the probability of positive response is less in row 1 than in row 2. The strength of association increases with the deviation from 1.

Asymptotic $100(1-\alpha)\%$ confidence limits for the column 1 relative risk are computed as

$$\left(RR_1 \times \exp(-z\sqrt{v}), \;\; RR_1 \times \exp(z\sqrt{v}) \right)$$

where

$$v = \text{var}(\ln RR_1) = \left((1-p_{1|1})/n_{11}\right) \; + \; \left((1-p_{1|2})/n_{21}\right)$$

and z is the $100(1-\alpha/2)$th percentile of the standard normal distribution. If either n_{11} or n_{21} is zero, the estimates are not computed.

PROC FREQ computes the column 2 relative risks in the same way.

Cochran-Armitage Test for Trend

The TREND option in the TABLES statement provides the Cochran-Armitage test for trend, which tests for trend in binomial proportions across levels of a single factor or covariate. This test is appropriate for a two-way table where one variable has two levels and the other variable is ordinal. The two-level variable represents the response, and the other variable represents an explanatory variable with ordered levels. When the two-way has two columns and R rows, PROC FREQ tests for trend across the R levels of the row variable, and the binomial proportion is computed as the proportion of observations in the first column. When the table has two rows and C columns, PROC FREQ tests for trend across the C levels of the column variable, and the binomial proportion is computed as the proportion of observations in the first row.

The trend test is based on the regression coefficient for the weighted linear regression of the binomial proportions on the scores of the explanatory variable levels. See Margolin (1988) and Agresti (2002) for details. If the table has two columns and R rows, the trend test statistic is computed as

$$T = \sum_{i=1}^{R} n_{i1}(R_i - \bar{R}) \; / \; \sqrt{p_{\cdot 1}\,(1-p_{\cdot 1})\,s^2}$$

where R_i is the score of row i, $\bar{R}$ is the average row score, and

$$s^2 = \sum_{i=1}^{R} n_{i\cdot}(R_i - \bar{R})^2$$

The SCORES= option in the TABLES statement determines the type of row scores used in computing the trend test (and other score-based statistics). The default is SCORES=TABLE. See the section "Scores" on page 1730 for details. For character variables, the table scores for the row variable are the row numbers (for example, 1 for the first row, 2 for the second row, and so on). For

numeric variables, the table score for each row is the numeric value of the row level. When you perform the trend test, the explanatory variable might be numeric (for example, dose of a test substance), and the variable values might be appropriate scores. If the explanatory variable has ordinal levels that are not numeric, you can assign meaningful scores to the variable levels. Sometimes equidistant scores, such as the table scores for a character variable, might be appropriate. For more information on choosing scores for the trend test, see Margolin (1988).

The null hypothesis for the Cochran-Armitage test is no trend, which means that the binomial proportion $p_{i1} = n_{i1}/n_{i\cdot}$ is the same for all levels of the explanatory variable. Under the null hypothesis, the trend statistic has an asymptotic standard normal distribution.

PROC FREQ computes one-sided and two-sided p-values for the trend test. When the test statistic is greater than its null hypothesis expected value of zero, PROC FREQ displays the right-sided p-value, which is the probability of a larger value of the statistic occurring under the null hypothesis. A small right-sided p-value supports the alternative hypothesis of increasing trend in proportions from row 1 to row R. When the test statistic is less than or equal to zero, PROC FREQ displays the left-sided p-value. A small left-sided p-value supports the alternative of decreasing trend.

The one-sided p-value for the trend test is computed as

$$P_1 = \begin{cases} \text{Prob}(Z > T) & \text{if } T > 0 \\ \text{Prob}(Z < T) & \text{if } T \leq 0 \end{cases}$$

where Z has a standard normal distribution. The two-sided p-value is computed as

$$P_2 = \text{Prob}(|Z| > |T|)$$

PROC FREQ also provides exact p-values for the Cochran-Armitage trend test. You can request the exact test by specifying the TREND option in the EXACT statement. See the section "Exact Statistics" on page 1779 for more information.

Jonckheere-Terpstra Test

The JT option in the TABLES statement provides the Jonckheere-Terpstra test, which is a nonparametric test for ordered differences among classes. It tests the null hypothesis that the distribution of the response variable does not differ among classes. It is designed to detect alternatives of ordered class differences, which can be expressed as $\tau_1 \leq \tau_2 \leq \cdots \leq \tau_R$ (or $\tau_1 \geq \tau_2 \geq \cdots \geq \tau_R$), with at least one of the inequalities being strict, where τ_i denotes the effect of class i. For such ordered alternatives, the Jonckheere-Terpstra test can be preferable to tests of more general class difference alternatives, such as the Kruskal–Wallis test (produced by the WILCOXON option in the NPAR1WAY procedure). See Pirie (1983) and Hollander and Wolfe (1999) for more information about the Jonckheere-Terpstra test.

The Jonckheere-Terpstra test is appropriate for a two-way table in which an ordinal column variable represents the response. The row variable, which can be nominal or ordinal, represents the classification variable. The levels of the row variable should be ordered according to the ordering you want the test to detect. The order of variable levels is determined by the ORDER= option in the PROC FREQ statement. The default is ORDER=INTERNAL, which orders by unformatted values. If you

specify ORDER=DATA, PROC FREQ orders values according to their order in the input data set. For more information about how to order variable levels, see the ORDER= option.

The Jonckheere-Terpstra test statistic is computed by first forming $R(R-1)/2$ Mann-Whitney counts $M_{i,i'}$, where $i < i'$, for pairs of rows in the contingency table,

$$\begin{aligned} M_{i,i'} &= \{ \text{ number of times } X_{i,j} < X_{i',j'}, \quad j = 1,\ldots,n_{i\cdot}; \quad j' = 1,\ldots,n_{i'\cdot} \} \\ &+ \tfrac{1}{2} \{ \text{ number of times } X_{i,j} = X_{i',j'}, \quad j = 1,\ldots,n_{i\cdot}; \quad j' = 1,\ldots,n_{i'\cdot} \} \end{aligned}$$

where $X_{i,j}$ is response j in row i. The Jonckheere-Terpstra test statistic is computed as

$$J = \sum_{1 \le i < i' \le R} \sum M_{i,i'}$$

This test rejects the null hypothesis of no difference among classes for large values of J. Asymptotic p-values for the Jonckheere-Terpstra test are obtained by using the normal approximation for the distribution of the standardized test statistic. The standardized test statistic is computed as

$$J^* = (J - \mathrm{E}_0(J)) \; / \; \sqrt{\mathrm{var}_0(J)}$$

where $\mathrm{E}_0(J)$ and $\mathrm{var}_0(J)$ are the expected value and variance of the test statistic under the null hypothesis,

$$\mathrm{E}_0(J) = \left(n^2 - \sum_i n_{i\cdot}^2 \right) / 4$$

$$\mathrm{var}_0(J) = A/72 + B/\left(36n(n-1)(n-2)\right) + C/\left(8n(n-1)\right)$$

where

$$A = n(n-1)(2n+5) - \sum_i n_{i\cdot}(n_{i\cdot}-1)(2n_{i\cdot}+5) - \sum_j n_{\cdot j}(n_{\cdot j}-1)(2n_{\cdot j}+5)$$

$$B = \left(\sum_i n_{i\cdot}(n_{i\cdot}-1)(n_{i\cdot}-2) \right) \left(\sum_j n_{\cdot j}(n_{\cdot j}-1)(n_{\cdot j}-2) \right)$$

$$C = \left(\sum_i n_{i\cdot}(n_{i\cdot}-1) \right) \left(\sum_j n_{\cdot j}(n_{\cdot j}-1) \right)$$

PROC FREQ computes one-sided and two-sided p-values for the Jonckheere-Terpstra test. When the standardized test statistic is greater than its null hypothesis expected value of zero, PROC FREQ displays the right-sided p-value, which is the probability of a larger value of the statistic occurring under the null hypothesis. A small right-sided p-value supports the alternative hypothesis of increasing order from row 1 to row R. When the standardized test statistic is less than or equal to zero, PROC FREQ displays the left-sided p-value. A small left-sided p-value supports the alternative of decreasing order from row 1 to row R.

The one-sided p-value for the Jonckheere-Terpstra test, P_1, is computed as

$$P_1 = \begin{cases} \text{Prob}(Z > J^*) & \text{if } J^* > 0 \\ \text{Prob}(Z < J^*) & \text{if } J^* \leq 0 \end{cases}$$

where Z has a standard normal distribution. The two-sided p-value, P_2, is computed as

$$P_2 = \text{Prob}(|Z| > |J^*|)$$

PROC FREQ also provides exact p-values for the Jonckheere-Terpstra test. You can request the exact test by specifying the JT option in the EXACT statement. See the section "Exact Statistics" on page 1779 for more information.

Tests and Measures of Agreement

When you specify the AGREE option in the TABLES statement, PROC FREQ computes tests and measures of agreement for square tables (that is, for tables where the number of rows equals the number of columns). For two-way tables, these tests and measures include McNemar's test for 2×2 tables, Bowker's test of symmetry, the simple kappa coefficient, and the weighted kappa coefficient. For multiple strata (n-way tables, where $n > 2$), PROC FREQ also computes the overall simple kappa coefficient and the overall weighted kappa coefficient, as well as tests for equal kappas (simple and weighted) among strata. Cochran's Q is computed for multiway tables when each variable has two levels, that is, for $h \times 2 \times 2$ tables.

PROC FREQ computes the kappa coefficients (simple and weighted), their asymptotic standard errors, and their confidence limits when you specify the AGREE option in the TABLES statement. If you also specify the KAPPA option in the TEST statement, then PROC FREQ computes the asymptotic test of the hypothesis that simple kappa equals zero. Similarly, if you specify the WTKAP option in the TEST statement, PROC FREQ computes the asymptotic test for weighted kappa.

In addition to the asymptotic tests described in this section, PROC FREQ provides exact p-values for McNemar's test, the simple kappa coefficient test, and the weighted kappa coefficient test. You can request these exact tests by specifying the corresponding options in the EXACT statement. See the section "Exact Statistics" on page 1779 for more information.

The following sections provide the formulas that PROC FREQ uses to compute the AGREE statistics. For information about the use and interpretation of these statistics, see Agresti (2002), Agresti (2007), Fleiss, Levin, and Paik (2003), and the other references cited for each statistic.

McNemar's Test

PROC FREQ computes McNemar's test for 2×2 tables when you specify the AGREE option. McNemar's test is appropriate when you are analyzing data from matched pairs of subjects with a dichotomous (yes-no) response. It tests the null hypothesis of marginal homogeneity, or $p_{1\cdot} = p_{\cdot 1}$. McNemar's test is computed as

$$Q_M = (n_{12} - n_{21})^2 \,/\, (n_{12} + n_{21})$$

Under the null hypothesis, Q_M has an asymptotic chi-square distribution with one degree of freedom. See McNemar (1947), as well as the general references cited in the preceding section. In addition to the asymptotic test, PROC FREQ also computes the exact p-value for McNemar's test when you specify the MCNEM option in the EXACT statement.

Bowker's Test of Symmetry

For Bowker's test of symmetry, the null hypothesis is that the cell proportions are symmetric, or that $p_{ij} = p_{ji}$ for all pairs of table cells. For 2×2 tables, Bowker's test is identical to McNemar's test, and so PROC FREQ provides Bowker's test for square tables larger than 2×2.

Bowker's test of symmetry is computed as

$$Q_B = \sum_{i<j}\sum (n_{ij} - n_{ji})^2 \;/\; (n_{ij} + n_{ji})$$

For large samples, Q_B has an asymptotic chi-square distribution with $R(R-1)/2$ degrees of freedom under the null hypothesis of symmetry. See Bowker (1948) for details.

Simple Kappa Coefficient

The simple kappa coefficient, introduced by Cohen (1960), is a measure of interrater agreement. PROC FREQ computes the simple kappa coefficient as

$$\hat{\kappa} = (P_o - P_e) \;/\; (1 - P_e)$$

where $P_o = \sum_i p_{ii}$ and $P_e = \sum_i p_{i.}p_{.i}$. If the two response variables are viewed as two independent ratings of the n subjects, the kappa coefficient equals +1 when there is complete agreement of the raters. When the observed agreement exceeds chance agreement, kappa is positive, with its magnitude reflecting the strength of agreement. Although this is unusual in practice, kappa is negative when the observed agreement is less than chance agreement. The minimum value of kappa is between -1 and 0, depending on the marginal proportions.

The asymptotic variance of the simple kappa coefficient is computed as

$$\mathrm{var}(\hat{\kappa}) = (A + B - C) \;/\; (1 - P_e)^2\, n$$

where

$$A = \sum_i p_{ii}\,(1 - (p_{i\cdot} + p_{\cdot i})(1 - \hat{\kappa}))^2$$

$$B = (1 - \hat{\kappa})^2 \sum_{i \neq j}\sum p_{ij}\,(p_{\cdot i} + p_{j\cdot})^2$$

$$C = (\,\hat{\kappa} - P_e(1 - \hat{\kappa})\,)^2$$

See Fleiss, Cohen, and Everitt (1969) for details.

PROC FREQ computes confidence limits for the simple kappa coefficient as

$$\hat{\kappa} \pm \left(z_{\alpha/2} \times \sqrt{\text{var}(\hat{\kappa})} \right)$$

where $z_{\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the standard normal distribution. The value of α is determined by the value of the ALPHA= option, which, by default, equals 0.05 and produces 95% confidence limits.

To compute an asymptotic test for the kappa coefficient, PROC FREQ uses the standardized test statistic $\hat{\kappa}^*$, which has an asymptotic standard normal distribution under the null hypothesis that kappa equals zero. The standardized test statistic is computed as

$$\hat{\kappa}^* = \hat{\kappa} \; / \; \sqrt{\text{var}_0(\hat{\kappa})}$$

where $\text{var}_0(\hat{\kappa})$ is the variance of the kappa coefficient under the null hypothesis,

$$\text{var}_0(\hat{\kappa}) = \left(P_e + P_e^2 - \sum_i p_{i\cdot} p_{\cdot i} (p_{i\cdot} + p_{\cdot i}) \right) \; / \; (1 - P_e)^2 \, n$$

See Fleiss, Levin, and Paik (2003) for details.

PROC FREQ also provides an exact test for the simple kappa coefficient. You can request the exact test by specifying the KAPPA or AGREE option in the EXACT statement. See the section "Exact Statistics" on page 1779 for more information.

Weighted Kappa Coefficient

The weighted kappa coefficient is a generalization of the simple kappa coefficient that uses weights to quantify the relative difference between categories. For 2×2 tables, the weighted kappa coefficient equals the simple kappa coefficient. PROC FREQ displays the weighted kappa coefficient only for tables larger than 2×2. PROC FREQ computes the kappa weights from the column scores, by using either Cicchetti-Allison weights or Fleiss-Cohen weights, both of which are described in the following section. The weights w_{ij} are constructed so that $0 \le w_{ij} < 1$ for all $i \ne j$, $w_{ii} = 1$ for all i, and $w_{ij} = w_{ji}$. The weighted kappa coefficient is computed as

$$\hat{\kappa}_w = \left(P_{o(w)} - P_{e(w)} \right) \; / \; \left(1 - P_{e(w)} \right)$$

where

$$P_{o(w)} = \sum_i \sum_j w_{ij} \, p_{ij}$$

$$P_{e(w)} = \sum_i \sum_j w_{ij} \, p_{i\cdot} \, p_{\cdot j}$$

The asymptotic variance of the weighted kappa coefficient is

$$\mathrm{var}(\hat{\kappa}_w) = \left(\sum_i \sum_j p_{ij} \left(w_{ij} - (\overline{w}_{i\cdot} + \overline{w}_{\cdot j})(1-\hat{\kappa}_w) \right)^2 - \left(\hat{\kappa}_w - P_{e(w)}(1-\hat{\kappa}_w) \right)^2 \right) \Big/ (1-P_{e(w)})^2 \, n$$

where

$$\overline{w}_{i\cdot} = \sum_j p_{\cdot j} \, w_{ij}$$

$$\overline{w}_{\cdot j} = \sum_i p_{i\cdot} w_{ij}$$

See Fleiss, Cohen, and Everitt (1969) for details.

PROC FREQ computes confidence limits for the weighted kappa coefficient as

$$\hat{\kappa}_w \pm \left(z_{\alpha/2} \times \sqrt{\mathrm{var}(\hat{\kappa}_w)} \right)$$

where $z_{\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the standard normal distribution. The value of α is determined by the value of the ALPHA= option, which, by default, equals 0.05 and produces 95% confidence limits.

To compute an asymptotic test for the weighted kappa coefficient, PROC FREQ uses the standardized test statistic $\hat{\kappa}_w^*$, which has an asymptotic standard normal distribution under the null hypothesis that weighted kappa equals zero. The standardized test statistic is computed as

$$\hat{\kappa}_w^* = \hat{\kappa}_w \Big/ \sqrt{\mathrm{var}_0(\hat{\kappa}_w)}$$

where $\mathrm{var}_0(\hat{\kappa}_w)$ is the variance of the weighted kappa coefficient under the null hypothesis,

$$\mathrm{var}_0(\hat{\kappa}_w) = \left(\sum_i \sum_j p_{i\cdot} p_{\cdot j} \left(w_{ij} - (\overline{w}_{i\cdot} + \overline{w}_{\cdot j}) \right)^2 - P_{e(w)}^2 \right) \Big/ (1-P_{e(w)})^2 \, n$$

See Fleiss, Levin, and Paik (2003) for details.

PROC FREQ also provides an exact test for the weighted kappa coefficient. You can request the exact test by specifying the WTKAPPA or AGREE option in the EXACT statement. See the section "Exact Statistics" on page 1779 for more information.

Weights PROC FREQ computes kappa coefficient weights by using the column scores and one of the two available weight types. The column scores are determined by the SCORES= option in the TABLES statement. The two available types of kappa weights are Cicchetti-Allison and Fleiss-Cohen weights. By default, PROC FREQ uses Cicchetti-Allison weights. If you specify (WT=FC) with the AGREE option, then PROC FREQ uses Fleiss-Cohen weights to compute the weighted kappa coefficient.

PROC FREQ computes Cicchetti-Allison kappa coefficient weights as

$$w_{ij} = 1 - \frac{|C_i - C_j|}{C_C - C_1}$$

where C_i is the score for column i and C is the number of categories or columns. See Cicchetti and Allison (1971) for details.

The SCORES= option in the TABLES statement determines the type of column scores used to compute the kappa weights (and other score-based statistics). The default is SCORES=TABLE. See the section "Scores" on page 1730 for details. For numeric variables, table scores are the values of the variable levels. You can assign numeric values to the levels in a way that reflects their level of similarity. For example, suppose you have four levels and order them according to similarity. If you assign them values of 0, 2, 4, and 10, the Cicchetti-Allison kappa weights take the following values: $w_{12} = 0.8$, $w_{13} = 0.6$, $w_{14} = 0$, $w_{23} = 0.8$, $w_{24} = 0.2$, and $w_{34} = 0.4$. Note that when there are only two categories (that is, $C = 2$), the weighted kappa coefficient is identical to the simple kappa coefficient.

If you specify (WT=FC) with the AGREE option in the TABLES statement, PROC FREQ computes Fleiss-Cohen kappa coefficient weights as

$$w_{ij} = 1 - \frac{(C_i - C_j)^2}{(C_C - C_1)^2}$$

See Fleiss and Cohen (1973) for details.

For the preceding example, the Fleiss-Cohen kappa weights are: $w_{12} = 0.96$, $w_{13} = 0.84$, $w_{14} = 0$, $w_{23} = 0.96$, $w_{24} = 0.36$, and $w_{34} = 0.64$.

Overall Kappa Coefficient

When there are multiple strata, PROC FREQ combines the stratum-level estimates of kappa into an overall estimate of the supposed common value of kappa. Assume there are q strata, indexed by $h = 1, 2, \ldots, q$, and let $\text{var}(\hat{\kappa}_h)$ denote the variance of $\hat{\kappa}_h$. The estimate of the overall kappa coefficient is computed as

$$\hat{\kappa}_T = \sum_{h=1}^{q} \frac{\hat{\kappa}_h}{\text{var}(\hat{\kappa}_h)} \Big/ \sum_{h=1}^{q} \frac{1}{\text{var}(\hat{\kappa}_h)}$$

See Fleiss, Levin, and Paik (2003) for details.

PROC FREQ computes an estimate of the overall weighted kappa in the same way.

Tests for Equal Kappa Coefficients

When there are multiple strata, the following chi-square statistic tests whether the stratum-level values of kappa are equal:

$$Q_K = \sum_{h=1}^{q} (\hat{\kappa}_h - \hat{\kappa}_T)^2 \Big/ \text{var}(\hat{\kappa}_h)$$

Under the null hypothesis of equal kappas for the q strata, Q_K has an asymptotic chi-square distribution with $q - 1$ degrees of freedom. See Fleiss, Levin, and Paik (2003) for more information. PROC FREQ computes a test for equal weighted kappa coefficients in the same way.

Cochran's Q Test

Cochran's Q is computed for multiway tables when each variable has two levels, that is, for $2 \times 2 \cdots \times 2$ tables. Cochran's Q statistic is used to test the homogeneity of the one-dimensional margins. Let m denote the number of variables and N denote the total number of subjects. Cochran's Q statistic is computed as

$$Q_C = m(m-1)\left(\sum_{j=1}^{m} T_j^2 - T^2\right) \Big/ \left(mT - \sum_{k=1}^{N} S_k^2\right)$$

where T_j is the number of positive responses for variable j, T is the total number of positive responses over all variables, and S_k is the number of positive responses for subject k. Under the null hypothesis, Cochran's Q has an asymptotic chi-square distribution with $m-1$ degrees of freedom. See Cochran (1950) for details. When there are only two binary response variables ($m = 2$), Cochran's Q simplifies to McNemar's test. When there are more than two response categories, you can test for marginal homogeneity by using the repeated measures capabilities of the CATMOD procedure.

Tables with Zero Rows and Columns

The AGREE statistics are defined only for square tables, where the number of rows equals the number of columns. If the table is not square, PROC FREQ does not compute AGREE statistics. In the kappa statistic framework, where two independent raters assign ratings to each of n subjects, suppose one of the raters does not use all possible r rating levels. If the corresponding table has r rows but only $r-1$ columns, then the table is not square and PROC FREQ does not compute AGREE statistics. To create a square table in this situation, use the ZEROS option in the WEIGHT statement, which requests that PROC FREQ include observations with zero weights in the analysis. Include zero-weight observations in the input data set to represent any rating levels that are not used by a rater, so that the input data set has at least one observation for each possible rater and rating combination. The analysis then includes all rating levels, even when all levels are not actually assigned by both raters. The resulting table (of rater 1 by rater 2) is a square table, and AGREE statistics can be computed.

For more information, see the description of the ZEROS option. By default, PROC FREQ does not process observations that have zero weights, because these observations do not contribute to the total frequency count, and because any resulting zero-weight row or column causes many of the tests and measures of association to be undefined. However, kappa statistics are defined for tables with a zero-weight row or column, and the ZEROS option makes it possible to input zero-weight observations and construct the tables needed to compute kappas.

Cochran-Mantel-Haenszel Statistics

The CMH option in the TABLES statement gives a stratified statistical analysis of the relationship between the row and column variables after controlling for the strata variables in a multiway table. For example, for the table request A*B*C*D, the CMH option provides an analysis of the relationship between C and D, after controlling for A and B. The stratified analysis provides a way to adjust for the possible confounding effects of A and B without being forced to estimate parameters for them.

The CMH analysis produces Cochran-Mantel-Haenszel statistics, which include the correlation statistic, the ANOVA (row mean scores) statistic, and the general association statistic. For 2×2 tables, the CMH option also provides Mantel-Haenszel and logit estimates of the common odds ratio and the common relative risks, as well as the Breslow-Day test for homogeneity of the odds ratios.

Exact statistics are also available for stratified 2×2 tables. If you specify the EQOR option in the EXACT statement, PROC FREQ provides Zelen's exact test for equal odds ratios. If you specify the COMOR option in the EXACT statement, PROC FREQ provides exact confidence limits for the common odds ratio and an exact test that the common odds ratio equals one.

Let the number of strata be denoted by q, indexing the strata by $h = 1, 2, \ldots, q$. Each stratum contains a contingency table with X representing the row variable and Y representing the column variable. For table h, denote the cell frequency in row i and column j by n_{hij}, with corresponding row and column marginal totals denoted by $n_{hi.}$ and $n_{h.j}$, and the overall stratum total by n_h.

Because the formulas for the Cochran-Mantel-Haenszel statistics are more easily defined in terms of matrices, the following notation is used. Vectors are presumed to be column vectors unless they are transposed ($'$).

$$
\begin{aligned}
\mathbf{n}'_{hi} &= (n_{hi1}, n_{hi2}, \ldots, n_{hiC}) && (1 \times C) \\
\mathbf{n}'_{h} &= (\mathbf{n}'_{h1}, \mathbf{n}'_{h2}, \ldots, \mathbf{n}'_{hR}) && (1 \times RC) \\
p_{hi\cdot} &= n_{hi\cdot} \,/\, n_h && (1 \times 1) \\
p_{h\cdot j} &= n_{h\cdot j} \,/\, n_h && (1 \times 1) \\
\mathbf{P}'_{h*\cdot} &= (p_{h1\cdot}, p_{h2\cdot}, \ldots, p_{hR\cdot}) && (1 \times R) \\
\mathbf{P}'_{h\cdot *} &= (p_{h\cdot 1}, p_{h\cdot 2}, \ldots, p_{h\cdot C}) && (1 \times C)
\end{aligned}
$$

Assume that the strata are independent and that the marginal totals of each stratum are fixed. The null hypothesis, H_0, is that there is no association between X and Y in any of the strata. The corresponding model is the multiple hypergeometric; this implies that, under H_0, the expected value and covariance matrix of the frequencies are, respectively,

$$\mathbf{m}_h = \mathbf{E}[\mathbf{n}_h \mid H_0] = n_h(\mathbf{P}_{h\cdot *} \otimes \mathbf{P}_{h*\cdot})$$

$$\mathbf{var}[\mathbf{n}_h \mid H_0] = c \left((\mathbf{D}_{\mathbf{P}h\cdot *} - \mathbf{P}_{h\cdot *}\mathbf{P}'_{h\cdot *}) \otimes (\mathbf{D}_{\mathbf{P}h*\cdot} - \mathbf{P}_{h*\cdot}\mathbf{P}'_{h*\cdot}) \right)$$

where

$$c = n_h^2 \,/\, (n_h - 1)$$

and where $\otimes$ denotes Kronecker product multiplication and $\mathbf{D}_\mathbf{a}$ is a diagonal matrix with the elements of $\mathbf{a}$ on the main diagonal.

The generalized CMH statistic (Landis, Heyman, and Koch 1978) is defined as

$$Q_{CMH} = \mathbf{G}'\mathbf{V}_\mathbf{G}{}^{-1}\mathbf{G}$$

where

$$\mathbf{G} = \sum_h \mathbf{B}_h(\mathbf{n}_h - \mathbf{m}_h)$$

$$\mathbf{V_G} = \sum_h \mathbf{B}_h \left(\mathit{Var}(\mathbf{n}_h \mid H_0)\right) \mathbf{B}_h'$$

and where

$$\mathbf{B}_h = \mathbf{C}_h \otimes \mathbf{R}_h$$

is a matrix of fixed constants based on column scores $\mathbf{C}_h$ and row scores $\mathbf{R}_h$. When the null hypothesis is true, the CMH statistic has an asymptotic chi-square distribution with degrees of freedom equal to the rank of $\mathbf{B}_h$. If $\mathbf{V_G}$ is found to be singular, PROC FREQ prints a message and sets the value of the CMH statistic to missing.

PROC FREQ computes three CMH statistics by using this formula for the generalized CMH statistic, with different row and column score definitions for each statistic. The CMH statistics that PROC FREQ computes are the correlation statistic, the ANOVA (row mean scores) statistic, and the general association statistic. These statistics test the null hypothesis of no association against different alternative hypotheses. The following sections describe the computation of these CMH statistics.

CAUTION: The CMH statistics have low power for detecting an association in which the patterns of association for some of the strata are in the opposite direction of the patterns displayed by other strata. Thus, a nonsignificant CMH statistic suggests either that there is no association or that no pattern of association has enough strength or consistency to dominate any other pattern.

Correlation Statistic

The correlation statistic, popularized by Mantel and Haenszel (1959) and Mantel (1963), has one degree of freedom and is known as the Mantel-Haenszel statistic.

The alternative hypothesis for the correlation statistic is that there is a linear association between X and Y in at least one stratum. If either X or Y does not lie on an ordinal (or interval) scale, then this statistic is not meaningful.

To compute the correlation statistic, PROC FREQ uses the formula for the generalized CMH statistic with the row and column scores determined by the SCORES= option in the TABLES statement. See the section "Scores" on page 1730 for more information about the available score types. The matrix of row scores $\mathbf{R}_h$ has dimension $1 \times R$, and the matrix of column scores $\mathbf{C}_h$ has dimension $1 \times C$.

When there is only one stratum, this CMH statistic reduces to $(n-1)r^2$, where r is the Pearson correlation coefficient between X and Y. When nonparametric (RANK or RIDIT) scores are specified, the statistic reduces to $(n-1)r_s^2$, where r_s is the Spearman rank correlation coefficient between X and Y. When there is more than one stratum, this CMH statistic becomes a stratum-adjusted correlation statistic.

ANOVA (Row Mean Scores) Statistic

The ANOVA statistic can be used only when the column variable Y lies on an ordinal (or interval) scale so that the mean score of Y is meaningful. For the ANOVA statistic, the mean score is computed for each row of the table, and the alternative hypothesis is that, for at least one stratum, the mean scores of the R rows are unequal. In other words, the statistic is sensitive to location differences among the R distributions of Y.

The matrix of column scores $\mathbf{C}_h$ has dimension $1 \times C$, and the column scores are determined by the SCORES= option.

The matrix of row scores $\mathbf{R}_h$ has dimension $(R-1) \times R$ and is created internally by PROC FREQ as

$$\mathbf{R}_h = [\mathbf{I}_{R-1}, -\mathbf{J}_{R-1}]$$

where $\mathbf{I}_{R-1}$ is an identity matrix of rank $R-1$ and $\mathbf{J}_{R-1}$ is an $(R-1) \times 1$ vector of ones. This matrix has the effect of forming $R-1$ independent contrasts of the R mean scores.

When there is only one stratum, this CMH statistic is essentially an analysis of variance (ANOVA) statistic in the sense that it is a function of the variance ratio F statistic that would be obtained from a one-way ANOVA on the dependent variable Y. If nonparametric scores are specified in this case, then the ANOVA statistic is a Kruskal-Wallis test.

If there is more than one stratum, then this CMH statistic corresponds to a stratum-adjusted ANOVA or Kruskal-Wallis test. In the special case where there is one subject per row and one subject per column in the contingency table of each stratum, this CMH statistic is identical to Friedman's chi-square. See Example 35.9 for an illustration.

General Association Statistic

The alternative hypothesis for the general association statistic is that, for at least one stratum, there is some kind of association between X and Y. This statistic is always interpretable because it does not require an ordinal scale for either X or Y.

For the general association statistic, the matrix $\mathbf{R}_h$ is the same as the one used for the ANOVA statistic. The matrix $\mathbf{C}_h$ is defined similarly as

$$\mathbf{C}_h = [\mathbf{I}_{C-1}, -\mathbf{J}_{C-1}]$$

PROC FREQ generates both score matrices internally. When there is only one stratum, then the general association CMH statistic reduces to $Q_P(n-1)/n$, where Q_P is the Pearson chi-square statistic. When there is more than one stratum, then the CMH statistic becomes a stratum-adjusted Pearson chi-square statistic. Note that a similar adjustment can be made by summing the Pearson chi-squares across the strata. However, the latter statistic requires a large sample size in each stratum to support the resulting chi-square distribution with $q(R-1)(C-1)$ degrees of freedom. The CMH statistic requires only a large overall sample size because it has only $(R-1)(C-1)$ degrees of freedom.

See Cochran (1954); Mantel and Haenszel (1959); Mantel (1963); Birch (1965); and Landis, Heyman, and Koch (1978).

Adjusted Odds Ratio and Relative Risk Estimates

The CMH option provides adjusted odds ratio and relative risk estimates for stratified 2×2 tables. For each of these measures, PROC FREQ computes a Mantel-Haenszel estimate and a logit estimate. These estimates apply to n-way table requests in the TABLES statement, when the row and column variables both have two levels.

For example, for the table request A*B*C*D, if the row and column variables C and D both have two levels, PROC FREQ provides odds ratio and relative risk estimates, adjusting for the confounding variables A and B.

The choice of an appropriate measure depends on the study design. For case-control (retrospective) studies, the odds ratio is appropriate. For cohort (prospective) or cross-sectional studies, the relative risk is appropriate. See the section "Odds Ratio and Relative Risks for 2 x 2 Tables" on page 1760 for more information on these measures.

Throughout this section, z denotes the $100(1-\alpha/2)$th percentile of the standard normal distribution.

Odds Ratio, Case-Control Studies PROC FREQ provides Mantel-Haenszel and logit estimates for the common odds ratio for stratified 2×2 tables.

The Mantel-Haenszel estimate of the common odds ratio is computed as

$$OR_{MH} = \left(\sum_h n_{h11}\, n_{h22}/n_h \right) \Big/ \left(\sum_h n_{h12}\, n_{h21}/n_h \right)$$

It is always computed unless the denominator is zero. See Mantel and Haenszel (1959) and Agresti (2002) for details.

To compute confidence limits for the common odds ratio, PROC FREQ uses the Greenland and Robins (1985) variance estimate for $\ln(OR_{MH})$. The $100(1-\alpha/2)$ confidence limits for the common odds ratio are

$$\left(OR_{MH} \times \exp(-z\hat{\sigma}),\quad OR_{MH} \times \exp(z\hat{\sigma}) \right)$$

where

$$\begin{aligned}
\hat{\sigma}^2 &= \widehat{\mathrm{var}}(\ln(OR_{MH})) \\
&= \frac{\sum_h (n_{h11}+n_{h22})(n_{h11}\, n_{h22})/n_h^2}{2\left(\sum_h n_{h11}\, n_{h22}/n_h\right)^2} \\
&\quad + \frac{\sum_h [(n_{h11}+n_{h22})(n_{h12}\, n_{h21}) + (n_{h12}+n_{h21})(n_{h11}\, n_{h22})]/n_h^2}{2\left(\sum_h n_{h11}\, n_{h22}/n_h\right)\left(\sum_h n_{h12}\, n_{h21}/n_h\right)} \\
&\quad + \frac{\sum_h (n_{h12}+n_{h21})(n_{h12}\, n_{h21})/n_h^2}{2\left(\sum_h n_{h12}\, n_{h21}/n_h\right)^2}
\end{aligned}$$

Note that the Mantel-Haenszel odds ratio estimator is less sensitive to small n_h than the logit estimator.

The adjusted logit estimate of the common odds ratio (Woolf 1955) is computed as

$$OR_L = \exp\left(\sum_h w_h \ln(OR_h) \;/\; \sum_h w_h\right)$$

and the corresponding $100(1-\alpha)\%$ confidence limits are

$$\left(OR_L \times \exp\left(-z/\sqrt{\sum_h w_h}\right),\quad OR_L \times \exp\left(z/\sqrt{\sum_h w_h}\right)\right)$$

where OR_h is the odds ratio for stratum h, and

$$w_h = 1/\mathrm{var}(\ln(OR_h))$$

If any table cell frequency in a stratum h is zero, PROC FREQ adds 0.5 to each cell of the stratum before computing OR_h and w_h (Haldane 1955) for the logit estimate. The procedure prints a warning when this occurs.

Relative Risks, Cohort Studies PROC FREQ provides Mantel-Haenszel and logit estimates of the common relative risks for stratified 2×2 tables.

The Mantel-Haenszel estimate of the common relative risk for column 1 is computed as

$$RR_{MH} = \left(\sum_h n_{h11}\, n_{h2\cdot} \,/\, n_h\right) \;/\; \left(\sum_h n_{h21}\, n_{h1\cdot} \,/\, n_h\right)$$

It is always computed unless the denominator is zero. See Mantel and Haenszel (1959) and Agresti (2002) for more information.

To compute confidence limits for the common relative risk, PROC FREQ uses the Greenland and Robins (1985) variance estimate for $\log(RR_{MH})$. The $100(1-\alpha/2)$ confidence limits for the common relative risk are

$$\left(RR_{MH} \times \exp(-z\hat{\sigma}),\quad RR_{MH} \times \exp(z\hat{\sigma})\right)$$

where

$$\hat{\sigma}^2 = \widehat{\mathrm{var}}(\ln(RR_{MH})) = \frac{\sum_h (n_{h1\cdot}\, n_{h2\cdot}\, n_{h\cdot 1} - n_{h11}\, n_{h21}\, n_h)/n_h^2}{\left(\sum_h n_{h11}\, n_{h2\cdot}/n_h\right)\left(\sum_h n_{h21}\, n_{h1\cdot}/n_h\right)}$$

The adjusted logit estimate of the common relative risk for column 1 is computed as

$$RR_L = \exp\left(\sum_h w_h \ln(RR_h) \;/\; \sum w_h\right)$$

and the corresponding $100(1-\alpha)\%$ confidence limits are

$$\left(RR_L \times \exp\left(-z \;/\; \sqrt{\sum_h w_h}\right),\quad RR_L \times \exp\left(z \;/\; \sqrt{\sum_h w_h}\right)\right)$$

where RR_h is the column 1 relative risk estimate for stratum h and

$$w_h = 1 \ / \ \text{var}(\ln(RR_h))$$

If n_{h11} or n_{h21} is zero, then PROC FREQ adds 0.5 to each cell of the stratum before computing RR_h and w_h for the logit estimate. The procedure prints a warning when this occurs. See Kleinbaum, Kupper, and Morgenstern (1982, Sections 17.4 and 17.5) for details.

Breslow-Day Test for Homogeneity of the Odds Ratios

When you specify the CMH option, PROC FREQ computes the Breslow-Day test for stratified 2×2 tables. It tests the null hypothesis that the odds ratios for the q strata are equal. When the null hypothesis is true, the statistic has approximately a chi-square distribution with $q - 1$ degrees of freedom. See Breslow and Day (1980) and Agresti (2007) for more information.

The Breslow-Day statistic is computed as

$$Q_{BD} = \sum_h (n_{h11} - \text{E}(n_{h11} \mid OR_{MH}))^2 \ / \ \text{var}(n_{h11} \mid OR_{MH})$$

where E and var denote expected value and variance, respectively. The summation does not include any table with a zero row or column. If OR_{MH} equals zero or if it is undefined, then PROC FREQ does not compute the statistic and prints a warning message.

For the Breslow-Day test to be valid, the sample size should be relatively large in each stratum, and at least 80% of the expected cell counts should be greater than 5. Note that this is a stricter sample size requirement than the requirement for the Cochran-Mantel-Haenszel test for $q \times 2 \times 2$ tables, in that each stratum sample size (not just the overall sample size) must be relatively large. Even when the Breslow-Day test is valid, it might not be very powerful against certain alternatives, as discussed in Breslow and Day (1980).

If you specify the BDT option, PROC FREQ computes the Breslow-Day test with Tarone's adjustment, which subtracts an adjustment factor from Q_{BD} to make the resulting statistic asymptotically chi-square. The Breslow-Day-Tarone statistic is computed as

$$Q_{BDT} = Q_{BD} - \left(\sum_h (n_{h11} - \text{E}(n_{h11} \mid OR_{MH})) \right)^2 \ / \ \sum_h \text{var}(n_{h11} \mid OR_{MH})$$

See Tarone (1985), Jones et al. (1989), and Breslow (1996) for more information.

Zelen's Exact Test for Equal Odds Ratios

If you specify the EQOR option in the EXACT statement, PROC FREQ computes Zelen's exact test for equal odds ratios for stratified 2×2 tables. Zelen's test is an exact counterpart to the Breslow-Day asymptotic test for equal odds ratios. The reference set for Zelen's test includes all possible $q \times 2 \times 2$ tables with the same row, column, and stratum totals as the observed multiway table and with the same sum of cell $(1, 1)$ frequencies as the observed table. The test statistic is the probability of the observed $q \times 2 \times 2$ table conditional on the fixed margins, which is a product of hypergeometric probabilities.

The p-value for Zelen's test is the sum of all table probabilities that are less than or equal to the observed table probability, where the sum is computed over all tables in the reference set determined by the fixed margins and the observed sum of cell $(1, 1)$ frequencies. This test is similar to Fisher's exact test for two-way tables. See Zelen (1971), Hirji (2006), and Agresti (1992) for more information. PROC FREQ computes Zelen's exact test by using the polynomial multiplication algorithm of Hirji et al. (1996).

Exact Confidence Limits for the Common Odds Ratio

If you specify the COMOR option in the EXACT statement, PROC FREQ computes exact confidence limits for the common odds ratio for stratified 2×2 tables. This computation assumes that the odds ratio is constant over all the 2×2 tables. Exact confidence limits are constructed from the distribution of $S = \sum_h n_{h11}$, conditional on the marginal totals of the 2×2 tables.

Because this is a discrete problem, the confidence coefficient for these exact confidence limits is not exactly $(1 - \alpha)$ but is at least $(1 - \alpha)$. Thus, these confidence limits are conservative. See Agresti (1992) for more information.

PROC FREQ computes exact confidence limits for the common odds ratio by using an algorithm based on Vollset, Hirji, and Elashoff (1991). See also Mehta, Patel, and Gray (1985).

Conditional on the marginal totals of 2×2 table h, let the random variable S_h denote the frequency of table cell $(1, 1)$. Given the row totals $n_{h1\cdot}$ and $n_{h2\cdot}$ and column totals $n_{h\cdot1}$ and $n_{h\cdot2}$, the lower and upper bounds for S_h are l_h and u_h,

$$\begin{aligned} l_h &= \max\,(\,0,\;\; n_{h1\cdot} - n_{h\cdot2}\,) \\ u_h &= \min\,(\,n_{h1\cdot},\;\; n_{h\cdot1}\,) \end{aligned}$$

Let C_{s_h} denote the hypergeometric coefficient,

$$C_{s_h} = \binom{n_{h\cdot1}}{s_h}\binom{n_{h\cdot2}}{n_{h1\cdot} - s_h}$$

and let ϕ denote the common odds ratio. Then the conditional distribution of S_h is

$$P(\,S_h = s_h \mid n_{1\cdot},\; n_{\cdot1},\; n_{\cdot2}\,) = C_{s_h}\,\phi^{\,s_h} \,/ \sum_{x\,=\,l_h}^{x\,=\,u_h} C_x\,\phi^{\,x}$$

Summing over all the 2×2 tables, $S = \sum_h S_h$, and the lower and upper bounds of S are l and u,

$$l = \sum_h l_h \quad \text{and} \quad u = \sum_h u_h$$

The conditional distribution of the sum S is

$$P(\,S = s \mid n_{h1\cdot},\; n_{h\cdot1},\; n_{h\cdot2};\;\; h = 1, \ldots, q\,) = C_s\,\phi^{\,s} \,/ \sum_{x\,=\,l}^{x\,=\,u} C_x\,\phi^{\,x}$$

where

$$C_s = \sum_{s_1+....+s_q\ =\ s} \left(\prod_h C_{s_h} \right)$$

Let s_0 denote the observed sum of cell (1,1) frequencies over the q tables. The following two equations are solved iteratively for lower and upper confidence limits for the common odds ratio, ϕ_1 and ϕ_2:

$$\sum_{x\ =\ s_0}^{x\ =\ u} C_x\ {\phi_1}^x\ /\ \sum_{x\ =\ l}^{x\ =\ u} C_x\ {\phi_1}^x \quad = \quad \alpha/2$$

$$\sum_{x\ =\ l}^{x\ =\ s_0} C_x\ {\phi_2}^x\ /\ \sum_{x\ =\ l}^{x\ =\ u} C_x\ {\phi_2}^x \quad = \quad \alpha/2$$

When the observed sum s_0 equals the lower bound l, PROC FREQ sets the lower confidence limit to zero and determines the upper limit with level α. Similarly, when the observed sum s_0 equals the upper bound u, PROC FREQ sets the upper confidence limit to infinity and determines the lower limit with level α.

When you specify the COMOR option in the EXACT statement, PROC FREQ also computes the exact test that the common odds ratio equals one. Setting $\phi = 1$, the conditional distribution of the sum S under the null hypothesis becomes

$$P_0(\ S = s \mid n_{h1\cdot},\ n_{h\cdot 1},\ n_{h\cdot 2};\ \ h = 1, \ldots, q\ ) = C_s\ /\ \sum_{x\ =\ l}^{x\ =\ u} C_x$$

The point probability for this exact test is the probability of the observed sum s_0 under the null hypothesis, conditional on the marginals of the stratified 2×2 tables, and is denoted by $P_0(s_0)$. The expected value of S under the null hypothesis is

$$\mathrm{E}_0(S) = \sum_{x\ =\ l}^{x\ =\ u} x\ C_x\ /\ \sum_{x\ =\ l}^{x\ =\ u} C_x$$

The one-sided exact p-value is computed from the conditional distribution as $P_0(S >= s_0)$ or $P_0(S \leq s_0)$, depending on whether the observed sum s_0 is greater or less than $\mathrm{E}_0(S)$,

$$P_1 \quad = \quad P_0(\ S >= s_0\ ) = \sum_{x\ =\ s_0}^{x\ =\ u} C_x\ /\ \sum_{x\ =\ l}^{x\ =\ u} C_x \quad \text{if}\ \ s_0 > \mathrm{E}_0(S)$$

$$P_1 \quad = \quad P_0(\ S <= s_0\ ) = \sum_{x\ =\ l}^{x\ =\ s_0} C_x\ /\ \sum_{x\ =\ l}^{x\ =\ u} C_x \quad \text{if}\ \ s_0 \leq \mathrm{E}_0(S)$$

PROC FREQ computes two-sided p-values for this test according to three different definitions. A two-sided p-value is computed as twice the one-sided p-value, setting the result equal to one if it exceeds one,

$$P_2{}^a = 2 \times P_1$$

Additionally, a two-sided p-value is computed as the sum of all probabilities less than or equal to the point probability of the observed sum s_0, summing over all possible values of s, $l \le s \le u$,

$$P_2{}^b = \sum_{l \le s \le u : P_0(s) \le P_0(s_0)} P_0(s)$$

Also, a two-sided p-value is computed as the sum of the one-sided p-value and the corresponding area in the opposite tail of the distribution, equidistant from the expected value,

$$P_2{}^c = P_0 \left(|S - \mathrm{E}_0(S)| \ge |s_0 - \mathrm{E}_0(S)| \right)$$

Exact Statistics

Exact statistics can be useful in situations where the asymptotic assumptions are not met, and so the asymptotic p-values are not close approximations for the true p-values. Standard asymptotic methods involve the assumption that the test statistic follows a particular distribution when the sample size is sufficiently large. When the sample size is not large, asymptotic results might not be valid, with the asymptotic p-values differing perhaps substantially from the exact p-values. Asymptotic results might also be unreliable when the distribution of the data is sparse, skewed, or heavily tied. See Agresti (2007) and Bishop, Fienberg, and Holland (1975) for more information. Exact computations are based on the statistical theory of exact conditional inference for contingency tables, reviewed by Agresti (1992).

In addition to computation of exact p-values, PROC FREQ provides the option of estimating exact p-values by Monte Carlo simulation. This can be useful for problems that are so large that exact computations require a great amount of time and memory, but for which asymptotic approximations might not be sufficient.

Exact statistics are available for many PROC FREQ tests. For one-way tables, PROC FREQ provides exact p-values for the binomial proportion tests and the chi-square goodness-of-fit test. Exact confidence limits are available for the binomial proportion. For two-way tables, PROC FREQ provides exact p-values for the following tests: Pearson chi-square test, likelihood-ratio chi-square test, Mantel-Haenszel chi-square test, Fisher's exact test, Jonckheere-Terpstra test, and Cochran-Armitage test for trend. PROC FREQ also computes exact p-values for tests of the following statistics: Pearson correlation coefficient, Spearman correlation coefficient, simple kappa coefficient, and weighted kappa coefficient. For 2×2 tables, PROC FREQ provides exact confidence limits for the odds ratio, exact unconditional confidence limits for the proportion difference, and McNemar's exact test. For stratified 2×2 tables, PROC FREQ provides Zelen's exact test for equal odds ratios, exact confidence limits for the common odds ratio, and an exact test for the common odds ratio.

The following sections summarize the exact computational algorithms, define the exact p-values that PROC FREQ computes, discuss the computational resource requirements, and describe the Monte Carlo estimation option.

Computational Algorithms

PROC FREQ computes exact p-values for general $R \times C$ tables by using the network algorithm developed by Mehta and Patel (1983). This algorithm provides a substantial advantage over direct enumeration, which can be very time-consuming and feasible only for small problems. See Agresti (1992) for a review of algorithms for computation of exact p-values, and see Mehta, Patel, and Tsiatis (1984) and Mehta, Patel, and Senchaudhuri (1991) for information about the performance of the network algorithm.

The reference set for a given contingency table is the set of all contingency tables with the observed marginal row and column sums. Corresponding to this reference set, the network algorithm forms a directed acyclic network consisting of nodes in a number of stages. A path through the network corresponds to a distinct table in the reference set. The distances between nodes are defined so that the total distance of a path through the network is the corresponding value of the test statistic. At each node, the algorithm computes the shortest and longest path distances for all the paths that pass through that node. For statistics that can be expressed as a linear combination of cell frequencies multiplied by increasing row and column scores, PROC FREQ computes shortest and longest path distances by using the algorithm of Agresti, Mehta, and Patel (1990). For statistics of other forms, PROC FREQ computes an upper bound for the longest path and a lower bound for the shortest path by following the approach of Valz and Thompson (1994).

The longest and shortest path distances or bounds for a node are compared to the value of the test statistic to determine whether all paths through the node contribute to the p-value, none of the paths through the node contribute to the p-value, or neither of these situations occurs. If all paths through the node contribute, the p-value is incremented accordingly, and these paths are eliminated from further analysis. If no paths contribute, these paths are eliminated from the analysis. Otherwise, the algorithm continues, still processing this node and the associated paths. The algorithm finishes when all nodes have been accounted for.

In applying the network algorithm, PROC FREQ uses full numerical precision to represent all statistics, row and column scores, and other quantities involved in the computations. Although it is possible to use rounding to improve the speed and memory requirements of the algorithm, PROC FREQ does not do this because it can result in reduced accuracy of the p-values.

For one-way tables, PROC FREQ computes the exact chi-square goodness-of-fit test by the method of Radlow and Alf (1975). PROC FREQ generates all possible one-way tables with the observed total sample size and number of categories. For each possible table, PROC FREQ compares its chi-square value with the value for the observed table. If the table's chi-square value is greater than or equal to the observed chi-square, PROC FREQ increments the exact p-value by the probability of that table, which is calculated under the null hypothesis by using the multinomial frequency distribution. By default, the null hypothesis states that all categories have equal proportions. If you specify null hypothesis proportions or frequencies by using the TESTP= or TESTF= option in the TABLES statement, then PROC FREQ calculates the exact chi-square test based on that null hypothesis.

Other exact computations are described in sections about the individual statistics. See the section "Binomial Proportion" on page 1745 for details about how PROC FREQ computes exact confidence limits and tests for the binomial proportion. See the section "Odds Ratio and Relative Risks for 2 x 2 Tables" on page 1760 for information about computation of exact confidence limits for the odds ratio for 2×2 tables. Also, see the sections "Exact Unconditional Confidence Limits for the Risk

Difference" on page 1759, "Exact Confidence Limits for the Common Odds Ratio" on page 1777, and "Zelen's Exact Test for Equal Odds Ratios" on page 1776.

Definition of *p*-Values

For several tests in PROC FREQ, the test statistic is nonnegative, and large values of the test statistic indicate a departure from the null hypothesis. Such nondirectional tests include the Pearson chi-square, the likelihood-ratio chi-square, the Mantel-Haenszel chi-square, Fisher's exact test for tables larger than 2×2, McNemar's test, and the one-way chi-square goodness-of-fit test. The exact *p*-value for a nondirectional test is the sum of probabilities for those tables having a test statistic greater than or equal to the value of the observed test statistic.

There are other tests where it might be appropriate to test against either a one-sided or a two-sided alternative hypothesis. For example, when you test the null hypothesis that the true parameter value equals 0 ($T = 0$), the alternative of interest might be one-sided ($T \leq 0$, or $T \geq 0$) or two-sided ($T \neq 0$). Such tests include the Pearson correlation coefficient, Spearman correlation coefficient, Jonckheere-Terpstra test, Cochran-Armitage test for trend, simple kappa coefficient, and weighted kappa coefficient. For these tests, PROC FREQ displays the right-sided *p*-value when the observed value of the test statistic is greater than its expected value. The right-sided *p*-value is the sum of probabilities for those tables for which the test statistic is greater than or equal to the observed test statistic. Otherwise, when the observed test statistic is less than or equal to the expected value, PROC FREQ displays the left-sided *p*-value. The left-sided *p*-value is the sum of probabilities for those tables for which the test statistic is less than or equal to the one observed. The one-sided *p*-value P_1 can be expressed as

$$P_1 = \begin{cases} \text{Prob}(\text{ Test Statistic} \geq t\ ) & \text{if } t > \text{E}_0(T) \\ \text{Prob}(\text{ Test Statistic} \leq t\ ) & \text{if } t \leq \text{E}_0(T) \end{cases}$$

where t is the observed value of the test statistic and $\text{E}_0(T)$ is the expected value of the test statistic under the null hypothesis. PROC FREQ computes the two-sided *p*-value as the sum of the one-sided *p*-value and the corresponding area in the opposite tail of the distribution of the statistic, equidistant from the expected value. The two-sided *p*-value P_2 can be expressed as

$$P_2 = \text{Prob}\left(\ |\text{Test Statistic} - \text{E}_0(T)| \geq |t - \text{E}_0(T)|\right)$$

If you specify the POINT option in the EXACT statement, PROC FREQ also displays exact point probabilities for the test statistics. The exact point probability is the exact probability that the test statistic equals the observed value.

Computational Resources

PROC FREQ uses relatively fast and efficient algorithms for exact computations. These recently developed algorithms, together with improvements in computer power, now make it feasible to perform exact computations for data sets where previously only asymptotic methods could be applied. Nevertheless, there are still large problems that might require a prohibitive amount of time and memory for exact computations, depending on the speed and memory available on your computer. For large problems, consider whether exact methods are really needed or whether asymptotic methods might give results quite close to the exact results, while requiring much less computer time and

memory. When asymptotic methods might not be sufficient for such large problems, consider using Monte Carlo estimation of exact *p*-values, as described in the section "Monte Carlo Estimation" on page 1782.

A formula does not exist that can predict in advance how much time and memory are needed to compute an exact *p*-value for a certain problem. The time and memory required depend on several factors, including which test is being performed, the total sample size, the number of rows and columns, and the specific arrangement of the observations into table cells. Generally, larger problems (in terms of total sample size, number of rows, and number of columns) tend to require more time and memory. Additionally, for a fixed total sample size, time and memory requirements tend to increase as the number of rows and columns increases, because this corresponds to an increase in the number of tables in the reference set. Also for a fixed sample size, time and memory requirements increase as the marginal row and column totals become more homogeneous. See Agresti, Mehta, and Patel (1990) and Gail and Mantel (1977) for more information.

At any time while PROC FREQ is computing exact *p*-values, you can terminate the computations by pressing the system interrupt key sequence (see the *SAS Companion* for your system) and choosing to stop computations. After you terminate exact computations, PROC FREQ completes all other remaining tasks. The procedure produces the requested output and reports missing values for any exact *p*-values that were not computed by the time of termination.

You can also use the MAXTIME= option in the EXACT statement to limit the amount of time PROC FREQ uses for exact computations. You specify a MAXTIME= value that is the maximum amount of clock time (in seconds) that PROC FREQ can use to compute an exact *p*-value. If PROC FREQ does not finish computing an exact *p*-value within that time, it terminates the computation and completes all other remaining tasks.

Monte Carlo Estimation

If you specify the option MC in the EXACT statement, PROC FREQ computes Monte Carlo estimates of the exact *p*-values instead of directly computing the exact *p*-values. Monte Carlo estimation can be useful for large problems that require a great amount of time and memory for exact computations but for which asymptotic approximations might not be sufficient. To describe the precision of each Monte Carlo estimate, PROC FREQ provides the asymptotic standard error and $100(1-\alpha)\%$ confidence limits. The confidence level α is determined by the ALPHA= option in the EXACT statement, which, by default, equals 0.01 and produces 99% confidence limits. The N=*n* option in the EXACT statement specifies the number of samples that PROC FREQ uses for Monte Carlo estimation; the default is 10000 samples. You can specify a larger value for n to improve the precision of the Monte Carlo estimates. Because larger values of n generate more samples, the computation time increases. Alternatively, you can specify a smaller value of n to reduce the computation time.

To compute a Monte Carlo estimate of an exact *p*-value, PROC FREQ generates a random sample of tables with the same total sample size, row totals, and column totals as the observed table. PROC FREQ uses the algorithm of Agresti, Wackerly, and Boyett (1979), which generates tables in proportion to their hypergeometric probabilities conditional on the marginal frequencies. For each sample table, PROC FREQ computes the value of the test statistic and compares it to the value for the observed table. When estimating a right-sided *p*-value, PROC FREQ counts all sample tables for which the test statistic is greater than or equal to the observed test statistic. Then the *p*-value

estimate equals the number of these tables divided by the total number of tables sampled.

$$\begin{aligned} \hat{P}_{MC} &= M \;/\; N \\ M &= \text{number of samples with (Test Statistic} \geq t) \\ N &= \text{total number of samples} \\ t &= \text{observed Test Statistic} \end{aligned}$$

PROC FREQ computes left-sided and two-sided p-value estimates in a similar manner. For left-sided p-values, PROC FREQ evaluates whether the test statistic for each sampled table is less than or equal to the observed test statistic. For two-sided p-values, PROC FREQ examines the sample test statistics according to the expression for P_2 given in the section "Definition of p-Values" on page 1781.

The variable M is a binomially distributed variable with N trials and success probability p. It follows that the asymptotic standard error of the Monte Carlo estimate is

$$\text{se}(\hat{P}_{MC}) = \sqrt{\hat{P}_{MC}\,(1 - \hat{P}_{MC}) \;/\; (N - 1)}$$

PROC FREQ constructs asymptotic confidence limits for the p-values according to

$$\hat{P}_{MC} \pm \left(z_{\alpha/2} \times \text{se}(\hat{P}_{MC}) \right)$$

where $z_{\alpha/2}$ is the $100(1 - \alpha/2)$th percentile of the standard normal distribution and the confidence level α is determined by the ALPHA= option in the EXACT statement.

When the Monte Carlo estimate $\hat{P}_{MC}$ equals 0, PROC FREQ computes the confidence limits for the p-value as

$$(0, \;\; 1 - \alpha^{(1/N)})$$

When the Monte Carlo estimate $\hat{P}_{MC}$ equals 1, PROC FREQ computes the confidence limits as

$$(\alpha^{(1/N)}, \;\; 1)$$

Computational Resources

For each variable in a table request, PROC FREQ stores all of the levels in memory. If all variables are numeric and not formatted, this requires about 84 bytes for each variable level. When there are character variables or formatted numeric variables, the memory that is required depends on the formatted variable lengths, with longer formatted lengths requiring more memory. The number of levels for each variable is limited only by the largest integer that your operating environment can store.

For any single crosstabulation table requested, PROC FREQ builds the entire table in memory, regardless of whether the table has zero cell counts. Thus, if the numeric variables A, B, and C each have 10 levels, PROC FREQ requires 2520 bytes to store the variable levels for the table request A*B*C, as follows:

```
3 variables * 10 levels/variable * 84 bytes/level
```

In addition, PROC FREQ requires 8000 bytes to store the table cell frequencies

```
1000 cells * 8 bytes/cell
```

even though there might be only 10 observations.

When the variables have many levels or when there are many multiway tables, your computer might not have enough memory to construct the tables. If PROC FREQ runs out of memory while constructing tables, it stops collecting levels for the variable with the most levels and returns the memory that is used by that variable. The procedure then builds the tables that do not contain the disabled variables.

If there is not enough memory for your table request and if increasing the available memory is impractical, you can reduce the number of multiway tables or variable levels. If you are not using the CMH or AGREE option in the TABLES statement to compute statistics across strata, reduce the number of multiway tables by using PROC SORT to sort the data set by one or more of the variables or by using the DATA step to create an index for the variables. Then remove the sorted or indexed variables from the TABLES statement and include a BY statement that uses these variables. You can also reduce memory requirements by using a FORMAT statement in the PROC FREQ step to reduce the number of levels. Additionally, reducing the formatted variable lengths reduces the amount of memory that is needed to store the variable levels. For more information about using formats, see the section "Grouping with Formats" on page 1725.

Output Data Sets

PROC FREQ produces two types of output data sets that you can use with other statistical and reporting procedures. You can request these data sets as follows:

- Specify the OUT= option in a TABLES statement. This creates an output data set that contains frequency or crosstabulation table counts and percentages
- Specify an OUTPUT statement. This creates an output data set that contains statistics.

PROC FREQ does not display the output data sets. Use PROC PRINT, PROC REPORT, or any other SAS reporting tool to display an output data set.

In addition to these two output data sets, you can create a SAS data set from any piece of PROC FREQ output by using the Output Delivery System. See the section "ODS Table Names" on page 1795 for more information.

Contents of the TABLES Statement Output Data Set

The OUT= option in the TABLES statement creates an output data set that contains one observation for each combination of variable values (or table cell) in the last table request. By default, each observation contains the frequency and percentage for the table cell. When the input data set contains missing values, the output data set also contains an observation with the frequency of missing values. The output data set includes the following variables:

- BY variables
- table request variables, such as A, B, C, and D in the table request A*B*C*D
- COUNT, which contains the table cell frequency
- PERCENT, which contains the table cell percentage

If you specify the OUTEXPECT option in the TABLES statement for a two-way or multiway table, the output data set also includes expected frequencies. If you specify the OUTPCT option for a two-way or multiway table, the output data set also includes row, column, and table percentages. The additional variables are as follows:

- EXPECTED, which contains the expected frequency
- PCT_TABL, which contains the percentage of two-way table frequency, for n-way tables where $n > 2$
- PCT_ROW, which contains the percentage of row frequency
- PCT_COL, which contains the percentage of column frequency

If you specify the OUTCUM option in the TABLES statement for a one-way table, the output data set also includes cumulative frequencies and cumulative percentages. The additional variables are as follows:

- CUM_FREQ, which contains the cumulative frequency
- CUM_PCT, which contains the cumulative percentage

The OUTCUM option has no effect for two-way or multiway tables.

The following PROC FREQ statements create an output data set of frequencies and percentages:

```
proc freq;
   tables A A*B / out=D;
run;
```

The output data set D contains frequencies and percentages for the table of A by B, which is the last table request listed in the TABLES statement. If A has two levels (1 and 2), B has three levels (1,2, and 3), and no table cell count is zero or missing, then the output data set D includes six

observations, one for each combination of A and B levels. The first observation corresponds to A=1 and B=1; the second observation corresponds to A=1 and B=2; and so on. The data set includes the variables COUNT and PERCENT. The value of COUNT is the number of observations with the given combination of A and B levels. The value of PERCENT is the percentage of the total number of observations with that A and B combination.

When PROC FREQ combines different variable values into the same formatted level, the output data set contains the smallest internal value for the formatted level. For example, suppose a variable X has the values 1.1., 1.4, 1.7, 2.1, and 2.3. When you submit the statement

```
format X 1.;
```

in a PROC FREQ step, the formatted levels listed in the frequency table for X are 1 and 2. If you create an output data set with the frequency counts, the internal values of the levels of X are 1.1 and 1.7. To report the internal values of X when you display the output data set, use a format of 3.1 for X.

Contents of the OUTPUT Statement Output Data Set

The OUTPUT statement creates a SAS data set that contains the statistics that PROC FREQ computes for the last table request. You specify which statistics to store in the output data set. There is an observation with the specified statistics for each stratum or two-way table. If PROC FREQ computes summary statistics for a stratified table, the output data set also contains a summary observation with those statistics.

The OUTPUT data set can include the following variables.

- BY variables
- variables that identify the stratum, such as A and B in the table request A*B*C*D
- variables that contain the specified statistics

The output data set also includes variables with the *p*-values and degrees of freedom, asymptotic standard error (ASE), or confidence limits when PROC FREQ computes these values for a specified statistic.

The variable names for the specified statistics in the output data set are the names of the options enclosed in underscores. PROC FREQ forms variable names for the corresponding *p*-values, degrees of freedom, or confidence limits by combining the name of the option with the appropriate prefix from the following list:

DF_	degrees of freedom
E_	asymptotic standard error (ASE)
L_	lower confidence limit
U_	upper confidence limit
E0_	ASE under the null hypothesis
Z_	standardized value

P_	*p*-value
P2_	two-sided *p*-value
PL_	left-sided *p*-value
PR_	right-sided *p*-value
XP_	exact *p*-value
XP2_	exact two-sided *p*-value
XPL_	exact left-sided *p*-value
XPR_	exact right-sided *p*-value
XPT_	exact point probability
XL_	exact lower confidence limit
XU_	exact upper confidence limit

For example, variable names created for the Pearson chi-square, its degrees of freedom, and its *p*-values are _PCHI_, DF_PCHI, and P_PCHI, respectively.

If the length of the prefix plus the statistic option exceeds eight characters, PROC FREQ truncates the option so that the name of the new variable is eight characters long.

Displayed Output

Number of Variable Levels Table

If you specify the NLEVELS option in the PROC FREQ statement, PROC FREQ displays the "Number of Variable Levels" table. This table provides the number of levels for all variables named in the TABLES statements. PROC FREQ determines the variable levels from the formatted variable values. See "Grouping with Formats" on page 1725 for details. The "Number of Variable Levels" table contains the following information:

- Variable name
- Levels, which is the total number of levels of the variable
- Number of Nonmissing Levels, if there are missing levels for any of the variables
- Number of Missing Levels, if there are missing levels for any of the variables

One-Way Frequency Tables

PROC FREQ displays one-way frequency tables for all one-way table requests in the TABLES statements, unless you specify the NOPRINT option in the PROC statement or the NOPRINT option in the TABLES statement. For a one-way table showing the frequency distribution of a single variable, PROC FREQ displays the name of the variable and its values. For each variable value or level, PROC FREQ displays the following information:

- Frequency count, which is the number of observations in the level
- Test Frequency count, if you specify the CHISQ and TESTF= options to request a chi-square goodness-of-fit test for specified frequencies

- Percent, which is the percentage of the total number of observations. (The NOPERCENT option suppresses this information.)
- Test Percent, if you specify the CHISQ and TESTP= options to request a chi-square goodness-of-fit test for specified percents. (The NOPERCENT option suppresses this information.)
- Cumulative Frequency count, which is the sum of the frequency counts for that level and all other levels listed above it in the table. The last cumulative frequency is the total number of nonmissing observations. (The NOCUM option suppresses this information.)
- Cumulative Percent, which is the percentage of the total number of observations in that level and in all other levels listed above it in the table. (The NOCUM or the NOPERCENT option suppresses this information.)

The one-way table also displays the Frequency Missing, which is the number of observations with missing values.

Statistics for One-Way Frequency Tables

For one-way tables, two statistical options are available in the TABLES statement. The CHISQ option provides a chi-square goodness-of-fit test, and the BINOMIAL option provides binomial proportion statistics and tests. PROC FREQ displays the following information, unless you specify the NOPRINT option in the PROC statement:

- If you specify the CHISQ option for a one-way table, PROC FREQ provides a chi-square goodness-of-fit test, displaying the Chi-Square statistic, the degrees of freedom (DF), and the probability value (Pr > ChiSq). If you specify the CHISQ option in the EXACT statement, PROC FREQ also displays the exact probability value for this test. If you specify the POINT option with the CHISQ option in the EXACT statement, PROC FREQ displays the exact point probability for the test statistic.
- If you specify the BINOMIAL option for a one-way table, PROC FREQ displays the estimate of the binomial Proportion, which is the proportion of observations in the first class listed in the one-way table. PROC FREQ also displays the asymptotic standard error (ASE) and the asymptotic (Wald) and exact (Clopper-Pearson) confidence limits by default. For the binomial proportion test, PROC FREQ displays the asymptotic standard error under the null hypothesis (ASE Under H0), the standardized test statistic (Z), and the one-sided and two-sided probability values.

  If you specify the BINOMIAL option in the EXACT statement, PROC FREQ also displays the exact one-sided and two-sided probability values for this test. If you specify the POINT option with the BINOMIAL option in the EXACT statement, PROC FREQ displays the exact point probability for the test.
- If you request additional binomial confidence limits by specifying *binomial-options*, PROC FREQ provides a table that displays the lower and upper confidence limits for each type that you request. In addition to the Wald and exact (Clopper-Pearson) confidence limits, you can request Agresti-Coull, Jeffreys, and Wilson (score) confidence limits for the binomial proportion.

- If you request a binomial noninferiority or superiority test by specifying the NONINF or SUP binomial-option, PROC FREQ displays the following information: the binomial Proportion, the test ASE (under H0 or Sample), the test statistic Z, the probability value, the noninferiority or superiority limit, and the test confidence limits. If you specify the BINOMIAL option in the EXACT statement, PROC FREQ also provides the exact probability value for the test, and exact test confidence limits.
- If you request a binomial equivalence test by specifying the EQUIV binomial-option, PROC FREQ displays the binomial Proportion and the test ASE (under H0 or Sample). PROC FREQ displays two one-sided tests (TOST) for equivalence, which include test statistics (Z) and probability values for the Lower and Upper tests, together with the Overall probability value. PROC FREQ also displays the equivalence limits and the test-based confidence limits. If you specify the BINOMIAL option in the EXACT statement, PROC FREQ provides exact probability values for the TOST and exact test-based confidence limits.

Multiway Tables

PROC FREQ displays all multiway table requests in the TABLES statements, unless you specify the NOPRINT option in the PROC statement or the NOPRINT option in the TABLES statement.

For two-way to multiway crosstabulation tables, the values of the last variable in the table request form the table columns. The values of the next-to-last variable form the rows. Each level (or combination of levels) of the other variables forms one stratum.

There are three ways to display multiway tables in PROC FREQ. By default, PROC FREQ displays multiway tables as separate two-way crosstabulation tables for each stratum of the multiway table. Also by default, PROC FREQ displays these two-way crosstabulation tables in table cell format. Alternatively, if you specify the CROSSLIST option, PROC FREQ displays the two-way crosstabulation tables in ODS column format. If you specify the LIST option, PROC FREQ displays multiway tables in list format, which presents the entire multiway crosstabulation in a single table.

Crosstabulation Tables

By default, PROC FREQ displays two-way crosstabulation tables in table cell format. The row variable values are listed down the side of the table, the column variable values are listed across the top of the table, and each row and column variable level combination forms a table cell.

Each cell of a crosstabulation table can contain the following information:

- Frequency, which is the number of observations in the table cell. (The NOFREQ option suppresses this information.)
- Expected frequency under the hypothesis of independence, if you specify the EXPECTED option
- Deviation of the cell frequency from the expected value, if you specify the DEVIATION option

- Cell Chi-Square, which is the cell's contribution to the total chi-square statistic, if you specify the CELLCHI2 option
- Tot Pct, which is the cell's percentage of the total multiway table frequency, for n-way tables when $n > 2$, if you specify the TOTPCT option
- Percent, which is the cell's percentage of the total (two-way table) frequency. (The NOPERCENT option suppresses this information.)
- Row Pct, or the row percentage, which is the cell's percentage of the total frequency for its row. (The NOROW option suppresses this information.)
- Col Pct, or column percentage, which is the cell's percentage of the total frequency for its column. (The NOCOL option suppresses this information.)
- Cumulative Col%, or cumulative column percentage, if you specify the CUMCOL option

The table also displays the Frequency Missing, which is the number of observations with missing values.

CROSSLIST Tables

If you specify the CROSSLIST option, PROC FREQ displays two-way crosstabulation tables in ODS column format. The CROSSLIST column format is different from the default crosstabulation table cell format, but the CROSSLIST table provides the same information (frequencies, percentages, and other statistics) as the default crosstabulation table.

In the CROSSLIST table format, the rows of the display correspond to the crosstabulation table cells, and the columns of the display correspond to descriptive statistics such as frequencies and percentages. Each table cell is identified by the values of its TABLES row and column variable levels, with all column variable levels listed within each row variable level. The CROSSLIST table also provides row totals, column totals, and overall table totals.

For a crosstabulation table in CROSSLIST format, PROC FREQ displays the following information:

- the row variable name and values
- the column variable name and values
- Frequency, which is the number of observations in the table cell. (The NOFREQ option suppresses this information.)
- Expected cell frequency under the hypothesis of independence, if you specify the EXPECTED option
- Deviation of the cell frequency from the expected value, if you specify the DEVIATION option
- Cell Chi-Square, which is the cell's contribution to the total chi-square statistic, if you specify the CELLCHI2 option

- Total Percent, which is the cell's percentage of the total multiway table frequency, for n-way tables when $n > 2$, if you specify the TOTPCT option
- Percent, which is the cell's percentage of the total (two-way table) frequency. (The NOPERCENT option suppresses this information.)
- Row Percent, which is the cell's percentage of the total frequency for its row. (The NOROW option suppresses this information.)
- Column Percent, the cell's percentage of the total frequency for its column. (The NOCOL option suppresses this information.)

The table also displays the Frequency Missing, which is the number of observations with missing values.

LIST Tables

If you specify the LIST option in the TABLES statement, PROC FREQ displays multiway tables in a list format rather than as crosstabulation tables. The LIST option displays the entire multiway table in one table, instead of displaying a separate two-way table for each stratum. The LIST option is not available when you also request statistical options. Unlike the default crosstabulation output, the LIST output does not display row percentages, column percentages, and optional information such as expected frequencies and cell chi-squares.

For a multiway table in list format, PROC FREQ displays the following information:

- the variable names and values
- Frequency, which is the number of observations in the level (with the indicated variable values)
- Percent, which is the level's percentage of the total number of observations. (The NOPERCENT option suppresses this information.)
- Cumulative Frequency, which is the accumulated frequency of the level and all other levels listed above it in the table. The last cumulative frequency in the table is the total number of nonmissing observations. (The NOCUM option suppresses this information.)
- Cumulative Percent, which is the accumulated percentage of the level and all other levels listed above it in the table. (The NOCUM or the NOPERCENT option suppresses this information.)

The table also displays the Frequency Missing, which is the number of observations with missing values.

Statistics for Multiway Tables

PROC FREQ computes statistical tests and measures for crosstabulation tables, depending on which statements and options you specify. You can suppress the display of all these results by specifying the NOPRINT option in the PROC statement. With any of the following information, PROC FREQ also displays the Sample Size and the Frequency Missing.

- If you specify the SCOROUT option, PROC FREQ displays the Row Scores and Column Scores that it uses for statistical computations. The Row Scores table displays the row variable values and the Score corresponding to each value. The Column Scores table displays the column variable values and the corresponding Scores. PROC FREQ also identifies the score type used to compute the row and column scores. You can specify the score type with the SCORES= option in the TABLES statement.

- If you specify the CHISQ option, PROC FREQ displays the following statistics for each two-way table: Pearson Chi-Square, Likelihood-Ratio Chi-Square, Continuity-Adjusted Chi-Square (for 2×2 tables), Mantel-Haenszel Chi-Square, the Phi Coefficient, the Contingency Coefficient, and Cramer's V. For each test statistic, PROC FREQ also displays the degrees of freedom (DF) and the probability value (Prob).

- If you specify the CHISQ option for 2×2 tables, PROC FREQ also displays Fisher's exact test. The test output includes the cell (1,1) frequency (F), the exact left-sided and right-sided probability values, the table probability (P), and the exact two-sided probability value.

- If you specify the FISHER option in the TABLES statement (or, equivalently, the FISHER option in the EXACT statement), PROC FREQ displays Fisher's exact test for tables larger than 2×2. The test output includes the table probability (P) and the probability value. In addition, PROC FREQ displays the CHISQ output listed earlier, even if you do not also specify the CHISQ option.

- If you specify the PCHI, LRCHI, or MHCHI option in the EXACT statement, PROC FREQ also displays the corresponding exact test: Pearson Chi-Square, Likelihood-Ratio Chi-Square, or Mantel-Haenszel Chi-Square, respectively. The test output includes the test statistic, the degrees of freedom (DF), and the asymptotic and exact probability values. If you also specify the POINT option in the EXACT statement, PROC FREQ displays the point probability for each exact test requested. If you specify the CHISQ option in the EXACT statement, PROC FREQ displays exact probability values for all three of these chi-square tests.

- If you specify the MEASURES option, PROC FREQ displays the following statistics and their asymptotic standard errors (ASE) for each two-way table: Gamma, Kendall's Tau-*b*, Stuart's Tau-*c*, Somers' $D(C|R)$, Somers' $D(R|C)$, Pearson Correlation, Spearman Correlation, Lambda Asymmetric $(C|R)$, Lambda Asymmetric $(R|C)$, Lambda Symmetric, Uncertainty Coefficient $(C|R)$, Uncertainty Coefficient $(R|C)$, and Uncertainty Coefficient Symmetric. If you specify the CL option, PROC FREQ also displays confidence limits for these measures.

- If you specify the PLCORR option, PROC FREQ displays the tetrachoric correlation for 2×2 tables or the polychoric correlation for larger tables. In addition, PROC FREQ displays the MEASURES output listed earlier, even if you do not also specify the MEASURES option.

- If you specify the option GAMMA, KENTB, STUTC, SMDCR, SMDRC, PCORR, or SCORR in the TEST statement, PROC FREQ displays asymptotic tests for Gamma, Kendall's Tau-*b*, Stuart's Tau-*c*, Somers' $D(C|R)$, Somers' $D(R|C)$, the Pearson Correlation, or the Spearman Correlation, respectively. If you specify the MEASURES option in the TEST statement, PROC FREQ displays all these asymptotic tests. The test output includes the statistic, its asymptotic standard error (ASE), Confidence Limits, the ASE under the null hypothesis H0, the standardized test statistic (Z), and the one-sided and two-sided probability values.

- If you specify the PCORR or SCORR option in the EXACT statement, PROC FREQ displays asymptotic and exact tests for the Pearson Correlation or the Spearman Correlation, respectively. The test output includes the correlation, its asymptotic standard error (ASE), Confidence Limits, the ASE under the null hypothesis H0, the standardized test statistic (Z), and the asymptotic and exact one-sided and two-sided probability values. If you also specify the POINT option in the EXACT statement, PROC FREQ displays the point probability for each exact test requested.

- If you specify the RISKDIFF option for 2×2 tables, PROC FREQ displays the Column 1 and Column 2 Risk Estimates. For each column, PROC FREQ displays the Row 1 Risk, Row 2 Risk, Total Risk, and Risk Difference, together with their asymptotic standard errors (ASE) and Asymptotic Confidence Limits. PROC FREQ also displays Exact Confidence Limits for the Row 1 Risk, Row 2 Risk, and Total Risk. If you specify the RISKDIFF option in the EXACT statement, PROC FREQ provides unconditional Exact Confidence Limits for the Risk Difference.

- If you request a noninferiority or superiority test for the proportion difference by specifying the NONINF or SUP riskdiff-option, and if you specify METHOD=HA (Hauck-Anderson), METHOD=FM (Farrington-Manning), or METHOD=WALD (Wald), PROC FREQ displays the following information: the Proportion Difference, the test ASE (H0, Sample, Sample H-A, or FM, depending on the method you specify), the test statistic Z, the probability value, the Noninferiority or Superiority Limit, and the test-based Confidence Limits. If you specify METHOD=SCORE (Newcombe score), PROC FREQ displays the Proportion Difference, the Noninferiority or Superiority Limit, and the score Confidence Limits.

- If you request an equivalence test for the proportion difference by specifying the EQUIV riskdiff-option, and if you specify METHOD=HA (Hauck-Anderson), METHOD=FM (Farrington-Manning), or METHOD=WALD (Wald), PROC FREQ displays the following information: the Proportion Difference and the test ASE (H0, Sample, Sample H-A, or FM, depending on the method you specify). PROC FREQ displays a two one-sided test (TOST) for equivalence, which includes test statistics (Z) and probability values for the Lower and Upper tests, together with the Overall probability value. PROC FREQ also displays the Equivalence Limits and the test-based Confidence Limits. If you specify METHOD=SCORE (Newcombe score), PROC FREQ displays the Proportion Difference, the Equivalence Limits, and the score Confidence Limits.

- If you request an equality test for the proportion difference by specifying the EQUAL riskdiff-option, PROC FREQ displays the following information: the Proportion Difference and the test ASE (H0 or Sample), the test statistic Z, the One-Sided probability value (Pr > Z or Pr < Z), and the Two-Sided probability value, Pr > |Z|.

- If you specify the MEASURES option or the RELRISK option for 2×2 tables, PROC FREQ displays Estimates of the Relative Risk for Case-Control and Cohort studies, together with their Confidence Limits. These measures are also known as the Odds Ratio and the Column 1 and 2 Relative Risks. If you specify the OR option in the EXACT statement, PROC FREQ also displays Exact Confidence Limits for the Odds Ratio.

- If you specify the TREND option, PROC FREQ displays the Cochran-Armitage Trend Test for tables that are $2 \times C$ or $R \times 2$. For this test, PROC FREQ gives the Statistic (Z) and the one-sided and two-sided probability values. If you specify the TREND option in the EXACT statement, PROC FREQ also displays the exact one-sided and two-sided probability values for this test. If you specify the POINT option with the TREND option in the EXACT statement, PROC FREQ displays the exact point probability for the test statistic.

- If you specify the JT option, PROC FREQ displays the Jonckheere-Terpstra Test, showing the Statistic (JT), the standardized test statistic (Z), and the one-sided and two-sided probability values. If you specify the JT option in the EXACT statement, PROC FREQ also displays the exact one-sided and two-sided probability values for this test. If you specify the POINT option with the JT option in the EXACT statement, PROC FREQ displays the exact point probability for the test statistic.

- If you specify the AGREE option and the PRINTKWT option, PROC FREQ displays the Kappa Coefficient Weights for square tables greater than 2×2.

- If you specify the AGREE option, for two-way tables PROC FREQ displays McNemar's Test and the Simple Kappa Coefficient for 2×2 tables. For square tables larger than 2×2, PROC FREQ displays Bowker's Test of Symmetry, the Simple Kappa Coefficient, and the Weighted Kappa Coefficient. For McNemar's Test and Bowker's Test of Symmetry, PROC FREQ displays the Statistic (S), the degrees of freedom (DF), and the probability value (Pr > S). If you specify the MCNEM option in the EXACT statement, PROC FREQ also displays the exact probability value for McNemar's test. If you specify the POINT option with the MCNEM option in the EXACT statement, PROC FREQ displays the exact point probability for the test statistic. For the simple and weighted kappa coefficients, PROC FREQ displays the kappa values, asymptotic standard errors (ASE), and Confidence Limits.

- If you specify the KAPPA or WTKAP option in the TEST statement, PROC FREQ displays asymptotic tests for the simple kappa coefficient or the weighted kappa coefficient, respectively. If you specify the AGREE option in the TEST statement, PROC FREQ displays both these asymptotic tests. The test output includes the kappa coefficient, its asymptotic standard error (ASE), Confidence Limits, the ASE under the null hypothesis H0, the standardized test statistic (Z), and the one-sided and two-sided probability values.

- If you specify the KAPPA or WTKAP option in the EXACT statement, PROC FREQ displays asymptotic and exact tests for the simple kappa coefficient or the weighted kappa coefficient, respectively. The test output includes the kappa coefficient, its asymptotic standard error (ASE), Confidence Limits, the ASE under the null hypothesis H0, the standardized test statistic (Z), and the asymptotic and exact one-sided and two-sided probability values. If you specify the POINT option in the EXACT statement, PROC FREQ displays the point probability for each exact test requested.

- If you specify the MC option in the EXACT statement, PROC FREQ displays Monte Carlo estimates for all exact *p*-values requested by *statistic-options* in the EXACT statement. The

Monte Carlo output includes the p-value Estimate, its Confidence Limits, the Number of Samples used to compute the Monte Carlo estimate, and the Initial Seed for random number generation.

- If you specify the AGREE option, for multiple strata PROC FREQ displays Overall Simple and Weighted Kappa Coefficients, with their asymptotic standard errors (ASE) and Confidence Limits. PROC FREQ also displays Tests for Equal Kappa Coefficients, giving the Chi-Squares, degrees of freedom (DF), and probability values (Pr > ChiSq) for the Simple Kappa and Weighted Kappa. For multiple strata of 2×2 tables, PROC FREQ displays Cochran's Q, giving the Statistic (Q), the degrees of freedom (DF), and the probability value (Pr > Q).

- If you specify the CMH option, PROC FREQ displays Cochran-Mantel-Haenszel Statistics for the following three alternative hypotheses: Nonzero Correlation, Row Mean Scores Differ (ANOVA Statistic), and General Association. For each of these statistics, PROC FREQ gives the degrees of freedom (DF) and the probability value (Prob). For 2×2 tables, PROC FREQ also displays Estimates of the Common Relative Risk for Case-Control and Cohort studies, together with their confidence limits. These include both Mantel-Haenszel and Logit stratum-adjusted estimates of the common Odds Ratio, Column 1 Relative Risk, and Column 2 Relative Risk. Also for 2×2 tables, PROC FREQ displays the Breslow-Day Test for Homogeneity of the Odds Ratios. For this test, PROC FREQ gives the Chi-Square, the degrees of freedom (DF), and the probability value (Pr > ChiSq).

- If you specify the CMH option in the TABLES statement and also specify the COMOR option in the EXACT statement, PROC FREQ displays exact confidence limits for the Common Odds Ratio for multiple strata of 2×2 tables. PROC FREQ also displays the Exact Test of H0: Common Odds Ratio = 1. The test output includes the Cell (1,1) Sum (S), Mean of S Under H0, One-sided Pr <= S, and Point Pr = S. PROC FREQ also provides exact two-sided probability values for the test, computed according to the following three methods: 2 * One-sided, Sum of probabilities <= Point probability, and Pr >= |S - Mean|.

- If you specify the CMH option in the TABLES statement and also specify the EQOR option in the EXACT statement, PROC FREQ computes Zelen's exact test for equal odds ratios for $h \times 2 \times 2$ tables. PROC FREQ displays Zelen's test along with the asymptotic Breslow-Day test produced by the CMH option. PROC FREQ displays the test statistic, Zelen's Exact Test (P), and the probability value, Exact Pr <= P.

ODS Table Names

PROC FREQ assigns a name to each table it creates. You can use these names to reference the table when you use the Output Delivery System (ODS) to select tables and create output data sets. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 35.14 lists the ODS table names together with their descriptions and the options required to produce the tables. Note that the ALL option in the TABLES statement invokes the CHISQ, MEASURES, and CMH options.

Table 35.14 ODS Tables Produced by PROC FREQ

ODS Table Name	Description	Statement	Option
BinomialCLs	Binomial confidence limits	TABLES	BINOMIAL(AC | J | W)
BinomialEquiv	Binomial equivalence analysis	TABLES	BINOMIAL(EQUIV)
BinomialEquivLimits	Binomial equivalence limits	TABLES	BINOMIAL(EQUIV)
BinomialEquivTest	Binomial equivalence test	TABLES	BINOMIAL(EQUIV)
BinomialNoninf	Binomial noninferiority test	TABLES	BINOMIAL(NONINF)
BinomialProp	Binomial proportion	TABLES	BINOMIAL (one-way tables)
BinomialPropTest	Binomial proportion test	TABLES	BINOMIAL (one-way tables)
BinomialSup	Binomial superiority test	TABLES	BINOMIAL(SUP)
BreslowDayTest	Breslow-Day test	TABLES	CMH ($h \times 2 \times 2$ tables)
CMH	Cochran-Mantel-Haenszel statistics	TABLES	CMH
ChiSq	Chi-square tests	TABLES	CHISQ
CochransQ	Cochran's Q	TABLES	AGREE ($h \times 2 \times 2$ tables)
ColScores	Column scores	TABLES	SCOROUT
CommonOddsRatioCL	Exact confidence limits for the common odds ratio	EXACT	COMOR ($h \times 2 \times 2$ tables)
CommonOddsRatioTest	Common odds ratio exact test	EXACT	COMOR ($h \times 2 \times 2$ tables)
CommonRelRisks	Common relative risks	TABLES	CMH ($h \times 2 \times 2$ tables)
CrossList	Crosstabulation table in column format	TABLES	CROSSLIST (n-way table request, $n > 1$)
CrossTabFreqs	Crosstabulation table	TABLES	(n-way table request, $n > 1$)
EqualKappaTest	Test for equal simple kappas	TABLES	AGREE ($h \times 2 \times 2$ tables)
EqualKappaTests	Tests for equal kappas	TABLES	AGREE ($h \times r \times r$ tables, $r > 2$)
EqualOddsRatios	Tests for equal odds ratios	EXACT	EQOR ($h \times 2 \times 2$ tables)
FishersExact	Fisher's exact test	EXACT or TABLES or TABLES	FISHER FISHER or EXACT CHISQ (2×2 tables)
FishersExactMC	Monte Carlo estimates for Fisher's exact test	EXACT	FISHER / MC
Gamma	Gamma	TEST	GAMMA
GammaTest	Gamma test	TEST	GAMMA
JTTest	Jonckheere-Terpstra test	TABLES	JT
JTTestMC	Monte Carlo estimates for Jonckheere-Terpstra exact test	EXACT	JT / MC
KappaStatistics	Kappa statistics	TABLES	AGREE ($r \times r$ tables, $r > 2$, no TEST or EXACT)
KappaWeights	Kappa weights	TABLES	AGREE and PRINTKWT
List	List format multiway table	TABLES	LIST
LRChiSq	Likelihood-ratio chi-square exact test	EXACT	LRCHI
LRChiSqMC	Monte Carlo exact test for likelihood-ratio chi-square	EXACT	LRCHI / MC

Table 35.14 *continued*

ODS Table Name	Description	Statement	Option
McNemarsTest	McNemar's test	TABLES	AGREE (2×2 tables)
Measures	Measures of association	TABLES	MEASURES
MHChiSq	Mantel-Haenszel chi-square exact test	EXACT	MHCHI
MHChiSqMC	Monte Carlo exact test for Mantel-Haenszel chi-square	EXACT	MHCHI / MC
NLevels	Number of variable levels	PROC	NLEVELS
OddsRatioCL	Exact confidence limits for the odds ratio	EXACT	OR (2×2 tables)
OneWayChiSq	One-way chi-square test	TABLES	CHISQ (one-way tables)
OneWayChiSqMC	Monte Carlo exact test for one-way chi-square	EXACT	CHISQ / MC (one-way tables)
OneWayFreqs	One-way frequencies	PROC	(with no TABLES stmt)
		or TABLES	(one-way table request)
OverallKappa	Overall simple kappa	TABLES	AGREE ($h \times 2 \times 2$ tables)
OverallKappas	Overall kappa coefficients	TABLES	AGREE ($h \times r \times r$ tables, $r > 2$)
PdiffEquiv	Equivalence analysis for the proportion difference	TABLES	RISKDIFF(EQUIV) (2×2 tables)
PdiffEquivLimits	Equivalence limits for the proportion difference	TABLES	RISKDIFF(EQUIV) (2×2 tables)
PdiffEquivTest	Equivalence test for the proportion difference	TABLES	RISKDIFF(EQUIV) (2×2 tables)
PdiffNoninf	Noninferiority test for the proportion difference	TABLES	RISKDIFF(NONINF) (2×2 tables)
PdiffSup	Superiority test for the proportion difference	TABLES	RISKDIFF(SUP) (2×2 tables)
PdiffTest	Proportion difference test	TABLES	RISKDIFF(EQUAL) (2×2 tables)
PearsonChiSq	Pearson chi-square exact test	EXACT	PCHI
PearsonChiSqMC	Monte Carlo exact test for Pearson chi-square	EXACT	PCHI / MC
PearsonCorr	Pearson correlation	TEST	PCORR
		or EXACT	PCORR
PearsonCorrMC	Monte Carlo exact test for Pearson correlation	EXACT	PCORR / MC
PearsonCorrTest	Pearson correlation test	TEST	PCORR
		or EXACT	PCORR
RelativeRisks	Relative risk estimates	TABLES	RELRISK or MEASURES (2×2 tables)
RiskDiffCol1	Column 1 risk estimates	TABLES	RISKDIFF (2×2 tables)
RiskDiffCol2	Column 2 risk estimates	TABLES	RISKDIFF (2×2 tables)
RowScores	Row scores	TABLES	SCOROUT
SimpleKappa	Simple kappa coefficient	TEST	KAPPA
		or EXACT	KAPPA

Table 35.14 *continued*

ODS Table Name	Description	Statement	Option
SimpleKappaMC	Monte Carlo exact test for simple kappa	EXACT	KAPPA / MC
SimpleKappaTest	Simple kappa test	TEST	KAPPA
		or EXACT	KAPPA
SomersDCR	Somers' $D(C\|R)$	TEST	SMDCR
SomersDCRTest	Somers' $D(C\|R)$ test	TEST	SMDCR
SomersDRC	Somers' $D(R\|C)$	TEST	SMDRC
SomersDRCTest	Somers' $D(R\|C)$ test	TEST	SMDRC
SpearmanCorr	Spearman correlation	TEST	SCORR
		or EXACT	SCORR
SpearmanCorrMC	Monte Carlo exact test for Spearman correlation	EXACT	SCORR / MC
SpearmanCorrTest	Spearman correlation test	TEST	SCORR
		or EXACT	SCORR
SymmetryTest	Test of symmetry	TABLES	AGREE
TauB	Kendall's tau-b	TEST	KENTB
TauBTest	Kendall's tau-b test	TEST	KENTB
TauC	Stuart's tau-c	TEST	STUTC
TauCTest	Stuart's tau-c test	TEST	STUTC
TrendTest	Cochran-Armitage trend test	TABLES	TREND
TrendTestMC	Monte Carlo exact test for trend	EXACT	TREND / MC
WeightedKappa	Weighted kappa	TEST	WTKAP
		or EXACT	WTKAP
WeightedKappaMC	Monte Carlo exact test for weighted kappa	EXACT	WTKAP / MC
WeightedKappaTest	Weighted kappa test	TEST	WTKAP
		or EXACT	WTKAP

* The ALL option in the TABLES statement invokes CHISQ, MEASURES, and CMH.

ODS Graphics

PROC FREQ assigns a name to each graph it creates with ODS Graphics. You can use these names to reference the graphs. Table 35.15 lists the names of the graphs that PROC FREQ generates, along with the corresponding analysis options.

To request graphics with PROC FREQ, you must first enable ODS Graphics by specifying the **ODS GRAPHICS ON** statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. When you have enabled ODS Graphics, you can request specific plots with the PLOTS= option in the TABLES statement. If you do not specify the PLOTS= option but have enabled ODS Graphics, then PROC FREQ produces all plots associated with the analyses you request.

Table 35.15 ODS Graphics Produced by PROC FREQ

ODS Graph Name	Plot Description	TABLES Statement Option
CumFreqPlot	Cumulative frequency plot	One-way table request
DeviationPlot	Deviation plot	CHISQ and a one-way table request
FreqPlot	Frequency plot	Any table request
KappaPlot	Kappa plot	AGREE ($h \times r \times r$ table)
ORPlot	Odds ratio plot	MEASURES or RELRISK ($h \times 2 \times 2$ table)
WtKappaPlot	Weighted kappa plot	AGREE ($h \times r \times r$ table, $r > 2$)

Examples: FREQ Procedure

Example 35.1: Output Data Set of Frequencies

The eye and hair color of children from two different regions of Europe are recorded in the data set Color. Instead of recording one observation per child, the data are recorded as cell counts, where the variable Count contains the number of children exhibiting each of the 15 eye and hair color combinations. The data set does not include missing combinations.

The following DATA step statements create the SAS data set Color:

```
data Color;
   input Region Eyes $ Hair $ Count @@;
      label Eyes  ='Eye Color'
            Hair  ='Hair Color'
            Region='Geographic Region';
      datalines;
1 blue  fair   23  1 blue  red     7  1 blue  medium 24
1 blue  dark   11  1 green fair   19  1 green red     7
1 green medium 18  1 green dark   14  1 brown fair   34
1 brown red     5  1 brown medium 41  1 brown dark   40
1 brown black   3  2 blue  fair   46  2 blue  red    21
2 blue  medium 44  2 blue  dark   40  2 blue  black   6
2 green fair   50  2 green red    31  2 green medium 37
2 green dark   23  2 brown fair   56  2 brown red    42
2 brown medium 53  2 brown dark   54  2 brown black  13
;
```

The following PROC FREQ statements read the Color data set and create an output data set that contains the frequencies, percentages, and expected cell frequencies of the two-way table of Eyes by Hair. The TABLES statement requests three tables: a frequency table for Eyes, a frequency table for Hair, and a crosstabulation table for Eyes by Hair. The OUT= option creates the FreqCount data set, which contains the crosstabulation table frequencies. The OUTEXPECT option outputs the expected table cell frequencies to FreqCount, and the SPARSE option includes zero cell frequen-

cies in the output data set. The WEIGHT statement specifies that the variable Count contains the observation weights. These statements create Output 35.1.1 through Output 35.1.3.

```
proc freq data=Color;
   tables Eyes Hair Eyes*Hair / out=FreqCount outexpect sparse;
   weight Count;
   title 'Eye and Hair Color of European Children';
run;

proc print data=FreqCount noobs;
   title2 'Output Data Set from PROC FREQ';
run;
```

Output 35.1.1 displays the two frequency tables produced by PROC FREQ: one showing the distribution of eye color, and one showing the distribution of hair color. By default, PROC FREQ lists the variables values in alphabetical order. The 'Eyes*Hair' specification produces a crosstabulation table, shown in Output 35.1.2, with eye color defining the table rows and hair color defining the table columns. A zero cell frequency for green eyes and black hair indicates that this eye and hair color combination does not occur in the data.

The output data set FreqCount (Output 35.1.3) contains frequency counts and percentages for the last table requested in the TABLES statement, Eyes by Hair. Because the SPARSE option is specified, the data set includes the observation with a zero frequency. The variable Expected contains the expected frequencies, as requested by the OUTEXPECT option.

Output 35.1.1 Frequency Tables

Eye and Hair Color of European Children

The FREQ Procedure

Eye Color

Eyes	Frequency	Percent	Cumulative Frequency	Cumulative Percent
blue	222	29.13	222	29.13
brown	341	44.75	563	73.88
green	199	26.12	762	100.00

Hair Color

Hair	Frequency	Percent	Cumulative Frequency	Cumulative Percent
black	22	2.89	22	2.89
dark	182	23.88	204	26.77
fair	228	29.92	432	56.69
medium	217	28.48	649	85.17
red	113	14.83	762	100.00

Output 35.1.2 Crosstabulation Table

Table of Eyes by Hair

Eyes(Eye Color) Hair(Hair Color)

Frequency Percent Row Pct Col Pct	black	dark	fair	medium	red	Total
blue	6	51	69	68	28	222
	0.79	6.69	9.06	8.92	3.67	29.13
	2.70	22.97	31.08	30.63	12.61	
	27.27	28.02	30.26	31.34	24.78	
brown	16	94	90	94	47	341
	2.10	12.34	11.81	12.34	6.17	44.75
	4.69	27.57	26.39	27.57	13.78	
	72.73	51.65	39.47	43.32	41.59	
green	0	37	69	55	38	199
	0.00	4.86	9.06	7.22	4.99	26.12
	0.00	18.59	34.67	27.64	19.10	
	0.00	20.33	30.26	25.35	33.63	
Total	22	182	228	217	113	762
	2.89	23.88	29.92	28.48	14.83	100.00

Output 35.1.3 Output Data Set of Frequencies

Eye and Hair Color of European Children
Output Data Set from PROC FREQ

Eyes	Hair	COUNT	EXPECTED	PERCENT
blue	black	6	6.409	0.7874
blue	dark	51	53.024	6.6929
blue	fair	69	66.425	9.0551
blue	medium	68	63.220	8.9239
blue	red	28	32.921	3.6745
brown	black	16	9.845	2.0997
brown	dark	94	81.446	12.3360
brown	fair	90	102.031	11.8110
brown	medium	94	97.109	12.3360
brown	red	47	50.568	6.1680
green	black	0	5.745	0.0000
green	dark	37	47.530	4.8556
green	fair	69	59.543	9.0551
green	medium	55	56.671	7.2178
green	red	38	29.510	4.9869

Example 35.2: Frequency Dot Plots

This example produces frequency dot plots for the children's eye and hair color data from Example 35.1.

PROC FREQ produces plots by using ODS Graphics to create graphs as part of the procedure output. Frequency plots are available for any frequency or crosstabulation table request. You can display frequency plots as bar charts or dot plots. You can use *plot-options* to specify the orientation (vertical or horizontal), scale, and layout of the plots.

The following PROC FREQ statements request frequency tables and dot plots. The first TABLES statement requests a one-way frequency table of Hair and a crosstabulation table of Eyes by Hair. The PLOTS= option requests frequency plots for the tables, and the TYPE=DOT plot-option specifies dot plots. By default, frequency plots are produced as bar charts.

The second TABLES statement requests a crosstabulation table of Region by Hair and a frequency dot plot for this table. The SCALE=PERCENT plot-option plots percentages instead of frequency counts. SCALE=LOG and SCALE=SQRT plot-options are also available to plot log frequencies and square roots of frequencies, respectively.

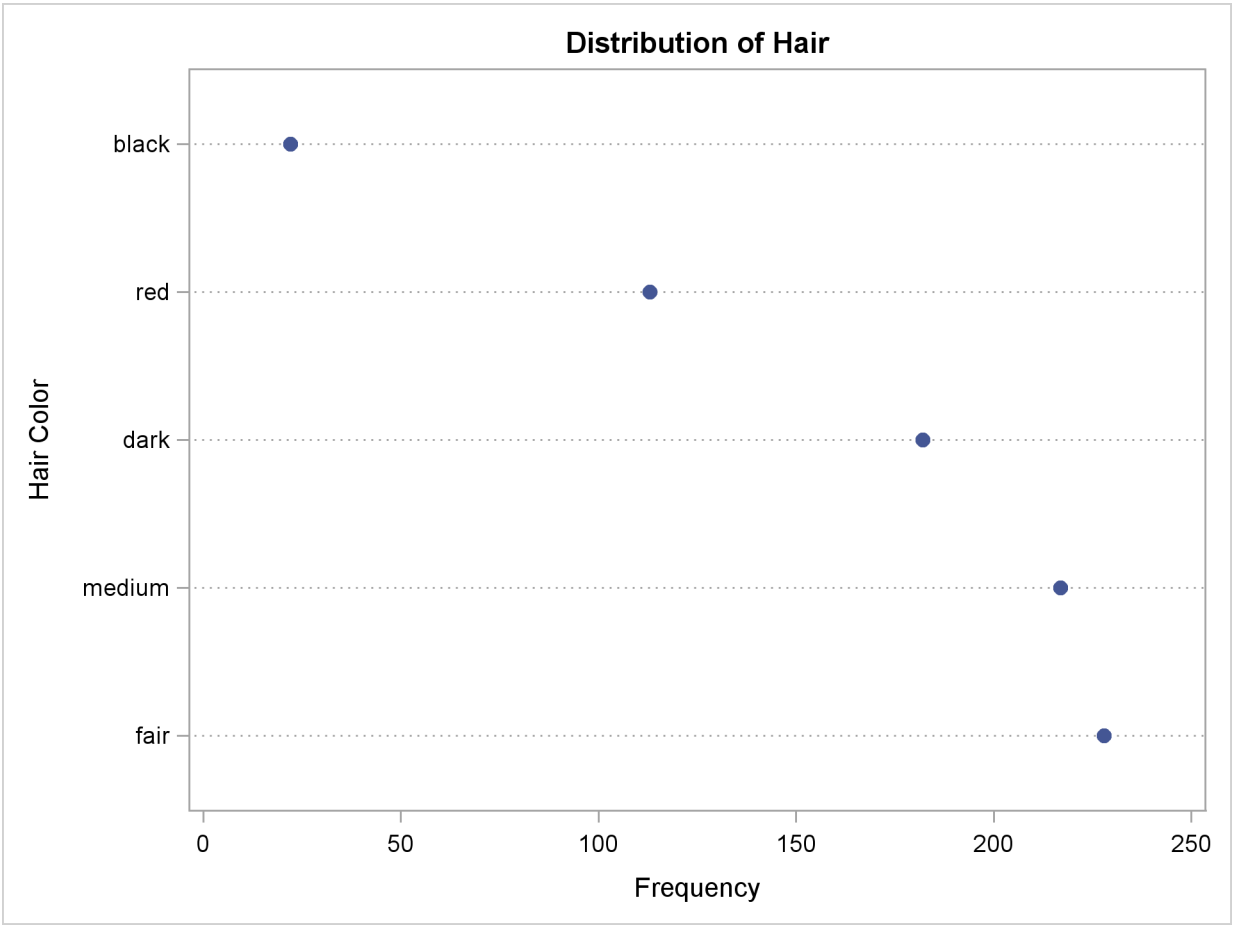
The ORDER=FREQ option in the PROC FREQ statement orders the variable levels by frequency. This order applies to the frequency and crosstabulation table displays and also to the corresponding frequency plots.

Before requesting plots, you must enable ODS Graphics with the **ODS GRAPHICS ON** statement.

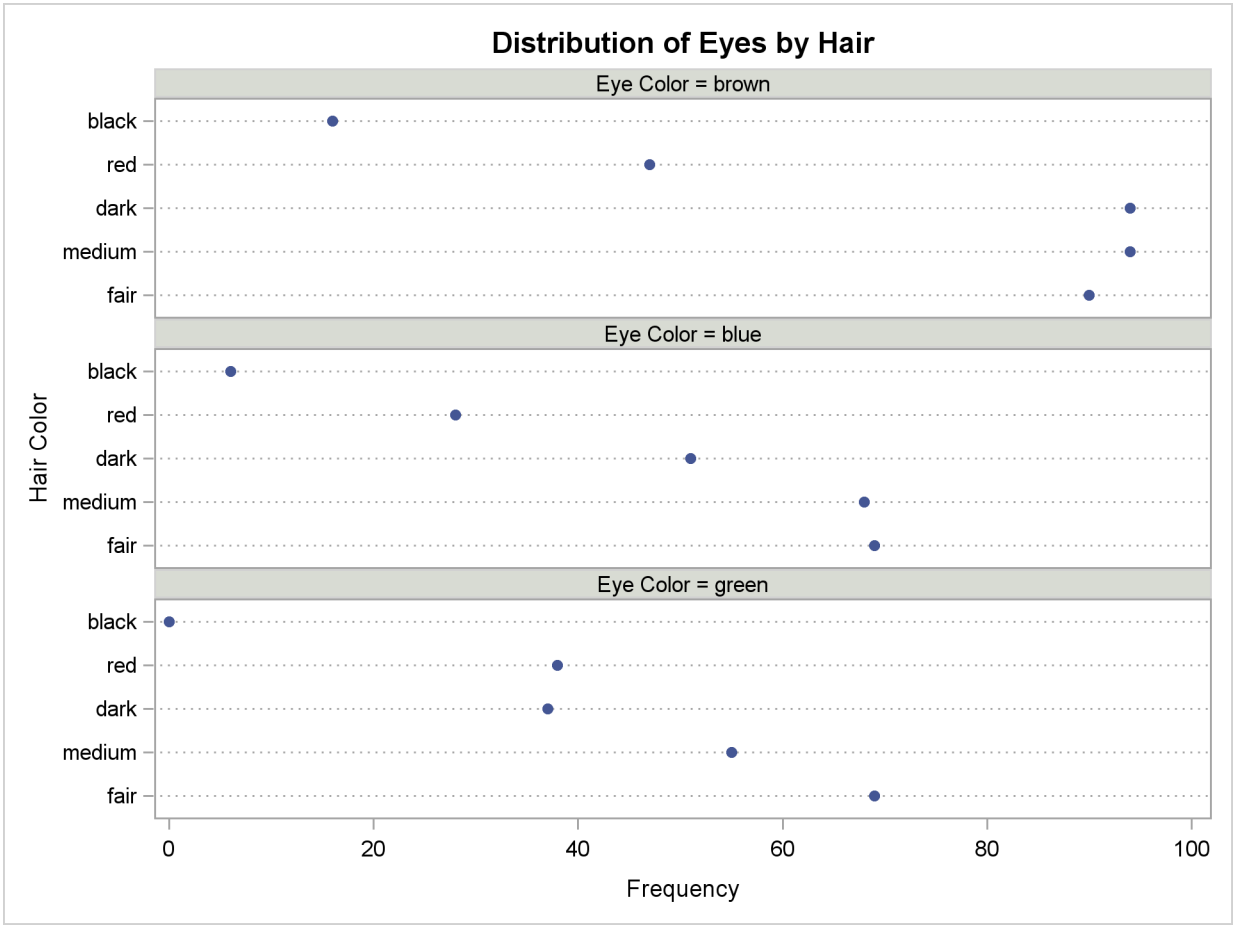
```
ods graphics on;
proc freq data=Color order=freq;
   tables Hair Eyes*Hair / plots=freqplot(type=dot);
   tables Region*Hair / plots=freqplot(type=dot scale=percent);
   weight Count;
   title 'Eye and Hair Color of European Children';
run;
ods graphics off;
```

Output 35.2.1, Output 35.2.2, and Output 35.2.3 display the dot plots produced by PROC FREQ. By default, the orientation of dot plots is horizontal, which places the variable levels on the *y*-axis. You can specify the ORIENT=VERTICAL plot-option to request a vertical orientation. For two-way plots, you can use the TWOWAY= plot-option to specify the plot layout. The default layout (shown in Output 35.2.2 and Output 35.2.3) is GROUPVERTICAL. Two-way layouts STACKED and GROUPHORIZONTAL are also available.

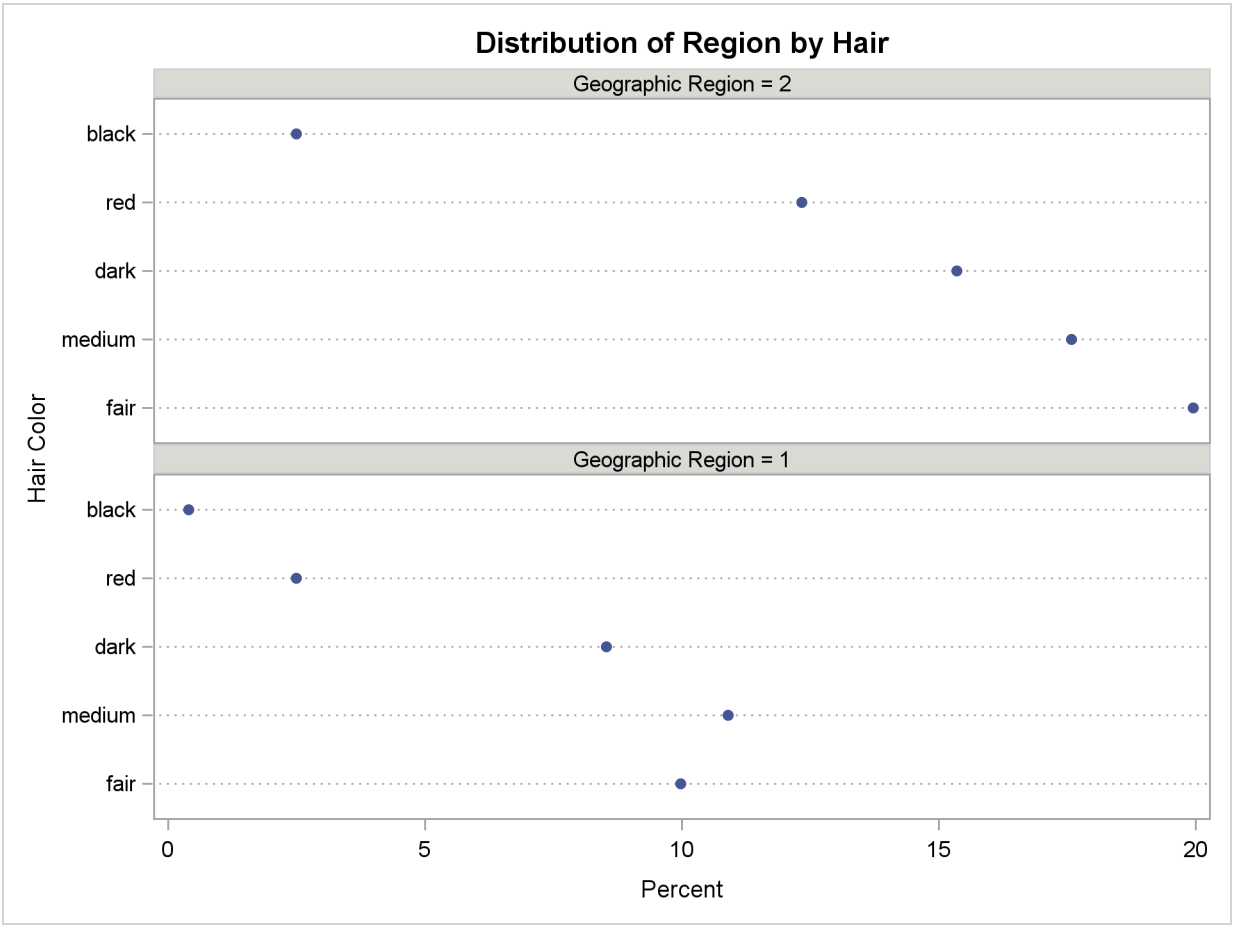
Output 35.2.1 One-Way Frequency Dot Plot



Output 35.2.2 Two-Way Frequency Dot Plot



Output 35.2.3 Two-Way Percent Dot Plot



Example 35.3: Chi-Square Goodness-of-Fit Tests

This example examines whether the children's hair color (from Example 35.1) has a specified multinomial distribution for the two geographical regions. The hypothesized distribution of hair color is 30% fair, 12% red, 30% medium, 25% dark, and 3% black.

In order to test the hypothesis for each region, the data are first sorted by Region. Then the FREQ procedure uses a BY statement to produce a separate table for each BY group (Region). The option ORDER=DATA orders the variable values (hair color) in the frequency table by their order in the input data set. The TABLES statement requests a frequency table for hair color, and the option NOCUM suppresses the display of the cumulative frequencies and percentages.

The CHISQ option requests a chi-square goodness-of-fit test for the frequency table of Hair. The TESTP= option specifies the hypothesized (or test) percentages for the chi-square test; the number of percentages listed equals the number of table levels, and the percentages sum to 100%. The TESTP= percentages are listed in the same order as the corresponding variable levels appear in frequency table.

The PLOTS= option requests a deviation plot, which is associated with the CHISQ option and displays the relative deviations from the test frequencies. The TYPE=DOT plot-option requests a dot plot instead of the default type, which is a bar chart. The ONLY plot-option requests that PROC FREQ produce only the deviation plot. By default, PROC FREQ produces all plots associated with the requested analyses. A frequency plot is associated with a one-way table request but is not produced in this example because ONLY is specified with the DEVIATIONPLOT request. Note that ODS Graphics must be enabled before requesting plots. These statements produce Output 35.3.1 through Output 35.3.4.

```
proc sort data=Color;
   by Region;
run;

ods graphics on;
proc freq data=Color order=data;
   tables Hair / nocum chisq testp=(30 12 30 25 3)
                 plots(only)=deviationplot(type=dot);
   weight Count;
   by Region;
   title 'Hair Color of European Children';
run;
ods graphics off;
```

Output 35.3.1 Frequency Table and Chi-Square Test for Region 1

Hair Color of European Children

Geographic Region=1

The FREQ Procedure

Hair Color

Hair	Frequency	Percent	Test Percent
fair	76	30.89	30.00
red	19	7.72	12.00
medium	83	33.74	30.00
dark	65	26.42	25.00
black	3	1.22	3.00

Geographic Region=1

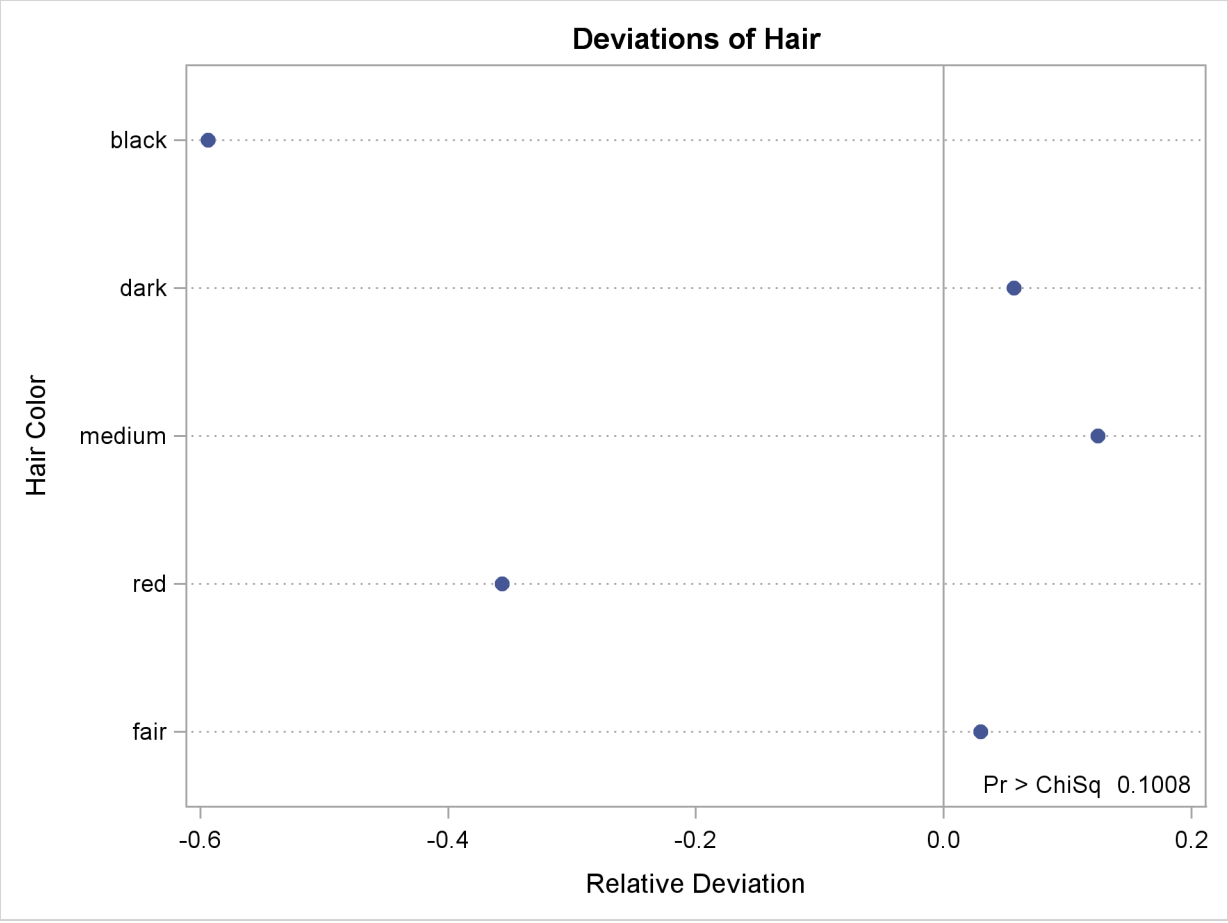
Chi-Square Test for Specified Proportions

Chi-Square	7.7602
DF	4
Pr > ChiSq	0.1008

Output 35.3.1 shows the frequency table and chi-square test for Region 1. The frequency table lists the variable values (hair color) in the order in which they appear in the data set. The "Test Percent" column lists the hypothesized percentages for the chi-square test. Always check that you have ordered the TESTP= percentages to correctly match the order of the variable levels.

Output 35.3.2 shows the deviation plot for Region 1, which displays the relative deviations from the hypothesized values. The relative deviation for a level is the difference between the observed and hypothesized (test) percentage divided by the test percentage. You can suppress the chi-square *p*-value that is displayed by default in the deviation plot by specifying the NOSTATS plot-option.

Output 35.3.2 Deviation Plot for Region 1



Output 35.3.3 and Output 35.3.4 show the results for Region 2. PROC FREQ computes a chi-square statistic for each region. The chi-square statistic is significant at the 0.05 level for Region 2 (p=0.0003) but not for Region 1. This indicates a significant departure from the hypothesized percentages in Region 2.

Output 35.3.3 Frequency Table and Chi-Square Test for Region 2

Hair Color of European Children

Geographic Region=2

The FREQ Procedure

Hair Color

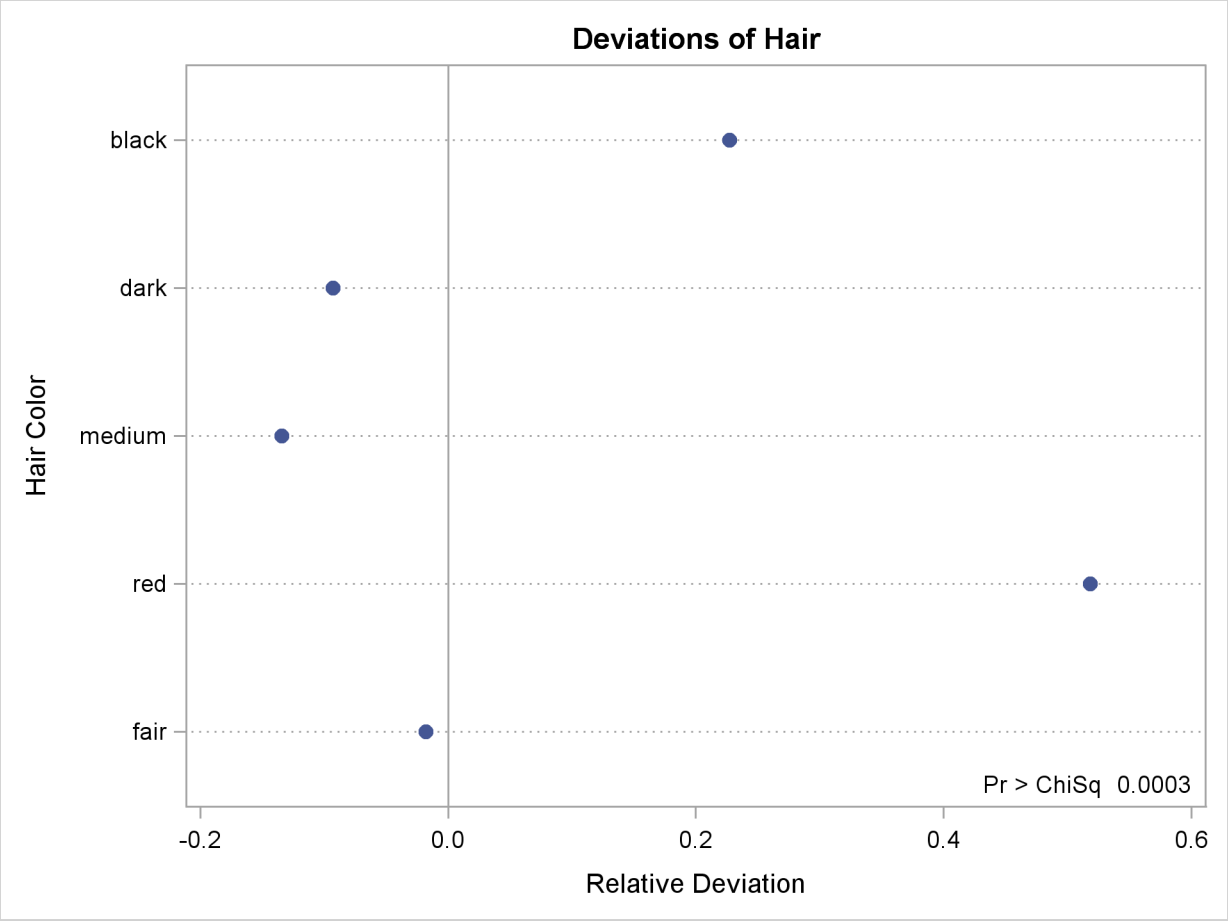
Hair	Frequency	Percent	Test Percent
fair	152	29.46	30.00
red	94	18.22	12.00
medium	134	25.97	30.00
dark	117	22.67	25.00
black	19	3.68	3.00

Geographic Region=2

Chi-Square Test for Specified Proportions

Chi-Square	21.3824
DF	4
Pr > ChiSq	0.0003

Output 35.3.4 Deviation Plot for Region 2



Example 35.4: Binomial Proportions

In this example, PROC FREQ computes binomial proportions, confidence limits, and tests. The example uses the eye and hair color data from Example 35.1. By default, PROC FREQ computes the binomial proportion as the proportion of observations in the first level of the one-way table. You can designate a different level by using the LEVEL= binomial-option.

The following PROC FREQ statements compute the proportion of children with brown eyes (from the data set in Example 35.1) and test the null hypothesis that the population proportion equals 50%. These statements also compute an equivalence for the proportion of children with fair hair.

The first TABLES statement requests a one-way frequency table for the variable Eyes. The BINOMIAL option requests the binomial proportion, confidence limits, and test. PROC FREQ computes the proportion with Eyes = 'brown', which is the first level displayed in the table. The AC, WILSON, and EXACT binomial-options request the following confidence limits types: Agresti-Coull, Wilson (score), and exact (Clopper-Pearson). By default, PROC FREQ provides Wald and exact (Clopper-Pearson) confidence limits for the binomial proportion. The BINOMIAL option also produces an asymptotic Wald test that the proportion equals 0.5. You can specify a different test

proportion with the P= binomial-option. The ALPHA=0.1 option specifies that $\alpha = 10\%$, which produces 90% confidence limits.

The second TABLES statement requests a one-way frequency table for the variable Hair. The BINOMIAL option requests the proportion for the first level, Hair = 'fair'. The EQUIV binomial-option requests an equivalence test for the binomial proportion. The P=.28 option specifies 0.28 as the null hypothesis proportion, and the MARGIN=.1 option specifies 0.1 as the equivalence test margin.

```
proc freq data=Color order=freq;
   tables Eyes / binomial(ac wilson exact) alpha=.1;
   tables Hair / binomial(equiv p=.28 margin=.1);
   weight Count;
   title 'Hair and Eye Color of European Children';
run;
```

Output 35.4.1 displays the results for eye color, and Output 35.4.2 displays the results for hair color.

Output 35.4.1 Binomial Proportion for Eye Color

```
                Hair and Eye Color of European Children

                        The FREQ Procedure

                            Eye Color

                                          Cumulative    Cumulative
Eyes      Frequency     Percent     Frequency      Percent
-----------------------------------------------------------------
brown           341       44.75           341        44.75
blue            222       29.13           563        73.88
green           199       26.12           762       100.00

                     Binomial Proportion
                       for Eyes = brown
                     ----------------------
                     Proportion      0.4475
                     ASE             0.0180

          Type                     90% Confidence Limits

          Wilson                      0.4181    0.4773
          Agresti-Coull               0.4181    0.4773
          Clopper-Pearson (Exact)     0.4174    0.4779

                   Test of H0: Proportion = 0.5

                 ASE under H0              0.0181
                 Z                        -2.8981
                 One-sided Pr <  Z         0.0019
                 Two-sided Pr > |Z|        0.0038
```

The frequency table in Output 35.4.1 displays the values of Eyes in order of descending frequency count. PROC FREQ computes the proportion of children in the first level displayed in the frequency table, Eyes = 'brown'. Output 35.4.1 displays the binomial proportion confidence limits and test. The confidence limits are 90% confidence limits. If you do not specify the ALPHA= option, PROC FREQ computes 95% confidence limits by default. Because the value of Z is less than zero, PROC FREQ displays the a left-sided p-value (0.0019). This small p-value supports the alternative hypothesis that the true value of the proportion of children with brown eyes is less than 50%.

Output 35.4.2 displays the equivalence test results produced by the second TABLES statement. The null hypothesis proportion is 0.28 and the equivalence margins are –0.1 and 0.1, which yield equivalence limits of 0.18 and 0.38. PROC FREQ provides two one-sided tests (TOST) for equivalence. The small p-value indicates rejection of the null hypothesis in favor of the alternative that the proportion is equivalent to the null value.

Output 35.4.2 Binomial Proportion for Hair Color

Hair Color

Hair	Frequency	Percent	Cumulative Frequency	Cumulative Percent
fair	228	29.92	228	29.92
medium	217	28.48	445	58.40
dark	182	23.88	627	82.28
red	113	14.83	740	97.11
black	22	2.89	762	100.00

Equivalence Analysis

H0: P - p0 <= Lower Margin or >= Upper Margin
Ha: Lower Margin < P - p0 < Upper Margin

p0 = 0.28 Lower Margin = -0.1 Upper Margin = 0.1

Proportion	ASE (Sample)
0.2992	0.0166

Two One-Sided Tests (TOST)

Test	Z	P-Value	
Lower Margin	7.1865	Pr > Z	<.0001
Upper Margin	-4.8701	Pr < Z	<.0001
Overall			<.0001

Equivalence Limits		90% Confidence Limits	
0.1800	0.3800	0.2719	0.3265

Example 35.5: Analysis of a 2x2 Contingency Table

This example computes chi-square tests and Fisher's exact test to compare the probability of coronary heart disease for two types of diet. It also estimates the relative risks and computes exact confidence limits for the odds ratio.

The data set FatComp contains hypothetical data for a case-control study of high fat diet and the risk of coronary heart disease. The data are recorded as cell counts, where the variable Count contains the frequencies for each exposure and response combination. The data set is sorted in descending order by the variables Exposure and Response, so that the first cell of the 2×2 table contains the frequency of positive exposure and positive response. The FORMAT procedure creates formats to identify the type of exposure and response with character values.

```
proc format;
   value ExpFmt 1='High Cholesterol Diet'
                0='Low Cholesterol Diet';
   value RspFmt 1='Yes'
                0='No';
run;
data FatComp;
   input Exposure Response Count;
   label Response='Heart Disease';
   datalines;
0 0  6
0 1  2
1 0  4
1 1 11
;
proc sort data=FatComp;
   by descending Exposure descending Response;
run;
```

In the following PROC FREQ statements, ORDER=DATA option orders the contingency table values by their order in the input data set. The TABLES statement requests a two-way table of Exposure by Response. The CHISQ option produces several chi-square tests, while the RELRISK option produces relative risk measures. The EXACT statement requests the exact Pearson chi-square test and exact confidence limits for the odds ratio.

```
proc freq data=FatComp order=data;
   format Exposure ExpFmt. Response RspFmt.;
   tables Exposure*Response / chisq relrisk;
   exact pchi or;
   weight Count;
   title 'Case-Control Study of High Fat/Cholesterol Diet';
run;
```

The contingency table in Output 35.5.1 displays the variable values so that the first table cell contains the frequency for the first cell in the data set (the frequency of positive exposure and positive response).

Output 35.5.1 Contingency Table

Case-Control Study of High Fat/Cholesterol Diet

The FREQ Procedure

Table of Exposure by Response

Exposure	Response(Heart Disease)		
Frequency Percent Row Pct Col Pct	Yes	No	Total
High Cholesterol Diet	11 47.83 73.33 84.62	4 17.39 26.67 40.00	15 65.22
Low Cholesterol Diet	2 8.70 25.00 15.38	6 26.09 75.00 60.00	8 34.78
Total	13 56.52	10 43.48	23 100.00

Output 35.5.2 displays the chi-square statistics. Because the expected counts in some of the table cells are small, PROC FREQ gives a warning that the asymptotic chi-square tests might not be appropriate. In this case, the exact tests are appropriate. The alternative hypothesis for this analysis states that coronary heart disease is more likely to be associated with a high fat diet, so a one-sided test is desired. Fisher's exact right-sided test analyzes whether the probability of heart disease in the high fat group exceeds the probability of heart disease in the low fat group; because this p-value is small, the alternative hypothesis is supported.

The odds ratio, displayed in Output 35.5.3, provides an estimate of the relative risk when an event is rare. This estimate indicates that the odds of heart disease is 8.25 times higher in the high fat diet group; however, the wide confidence limits indicate that this estimate has low precision.

Output 35.5.2 Chi-Square Statistics

Statistic	DF	Value	Prob
Chi-Square	1	4.9597	0.0259
Likelihood Ratio Chi-Square	1	5.0975	0.0240
Continuity Adj. Chi-Square	1	3.1879	0.0742
Mantel-Haenszel Chi-Square	1	4.7441	0.0294
Phi Coefficient		0.4644	
Contingency Coefficient		0.4212	
Cramer's V		0.4644	

WARNING: 50% of the cells have expected counts less than 5.
(Asymptotic) Chi-Square may not be a valid test.

Pearson Chi-Square Test	
Chi-Square	4.9597
DF	1
Asymptotic Pr > ChiSq	0.0259
Exact Pr >= ChiSq	0.0393

Fisher's Exact Test	
Cell (1,1) Frequency (F)	11
Left-sided Pr <= F	0.9967
Right-sided Pr >= F	0.0367
Table Probability (P)	0.0334
Two-sided Pr <= P	0.0393

Output 35.5.3 Relative Risk

Estimates of the Relative Risk (Row1/Row2)

Type of Study	Value	95% Confidence Limits	
Case-Control (Odds Ratio)	8.2500	1.1535	59.0029
Cohort (Col1 Risk)	2.9333	0.8502	10.1204
Cohort (Col2 Risk)	0.3556	0.1403	0.9009

Odds Ratio (Case-Control Study)	
Odds Ratio	8.2500
Asymptotic Conf Limits	
95% Lower Conf Limit	1.1535
95% Upper Conf Limit	59.0029
Exact Conf Limits	
95% Lower Conf Limit	0.8677
95% Upper Conf Limit	105.5488

Example 35.6: Output Data Set of Chi-Square Statistics

This example uses the Color data from Example 35.1 to output the Pearson chi-square and the likelihood-ratio chi-square statistics to a SAS data set. The following PROC FREQ statements create a two-way table of eye color versus hair color.

```
proc freq data=Color order=data;
   tables Eyes*Hair / expected cellchi2 norow nocol chisq;
   output out=ChiSqData n nmiss pchi lrchi;
   weight Count;
   title 'Chi-Square Tests for 3 by 5 Table of Eye and Hair Color';
run;

proc print data=ChiSqData noobs;
   title1 'Chi-Square Statistics for Eye and Hair Color';
   title2 'Output Data Set from the FREQ Procedure';
run;
```

The EXPECTED option displays expected cell frequencies in the crosstabulation table, and the CELLCHI2 option displays the cell contribution to the overall chi-square. The NOROW and NOCOL options suppress the display of row and column percents in the crosstabulation table. The CHISQ option produces chi-square tests.

The OUTPUT statement creates the ChiSqData output data set and specifies the statistics to include. The N option requests the number of nonmissing observations, the NMISS option stores the number of missing observations, and the PCHI and LRCHI options request Pearson and likelihood-ratio chi-square statistics, respectively, together with their degrees of freedom and p-values.

The preceding statements produce Output 35.6.1 and Output 35.6.2. The contingency table in Output 35.6.1 displays eye and hair color in the order in which they appear in the Color data set. The Pearson chi-square statistic in Output 35.6.2 provides evidence of an association between eye and hair color (p=0.0073). The cell chi-square values show that most of the association is due to more green-eyed children with fair or red hair and fewer with dark or black hair. The opposite occurs with the brown-eyed children.

Output 35.6.3 displays the output data set created by the OUTPUT statement. It includes one observation that contains the sample size, the number of missing values, and the chi-square statistics and corresponding degrees of freedom and p-values as in Output 35.6.2.

Output 35.6.1 Contingency Table

Chi-Square Tests for 3 by 5 Table of Eye and Hair Color

The FREQ Procedure

Table of Eyes by Hair

Eyes(Eye Color) / Hair(Hair Color)	fair	red	medium	dark	black	Total
blue	69	28	68	51	6	222
	66.425	32.921	63.22	53.024	6.4094	
	0.0998	0.7357	0.3613	0.0772	0.0262	
	9.06	3.67	8.92	6.69	0.79	29.13
green	69	38	55	37	0	199
	59.543	29.51	56.671	47.53	5.7454	
	1.5019	2.4422	0.0492	2.3329	5.7454	
	9.06	4.99	7.22	4.86	0.00	26.12
brown	90	47	94	94	16	341
	102.03	50.568	97.109	81.446	9.8451	
	1.4187	0.2518	0.0995	1.935	3.8478	
	11.81	6.17	12.34	12.34	2.10	44.75
Total	228	113	217	182	22	762
	29.92	14.83	28.48	23.88	2.89	100.00

Output 35.6.2 Chi-Square Statistics

Statistic	DF	Value	Prob
Chi-Square	8	20.9248	0.0073
Likelihood Ratio Chi-Square	8	25.9733	0.0011
Mantel-Haenszel Chi-Square	1	3.7838	0.0518
Phi Coefficient		0.1657	
Contingency Coefficient		0.1635	
Cramer's V		0.1172	

Output 35.6.3 Output Data Set

Chi-Square Statistics for Eye and Hair Color
Output Data Set from the FREQ Procedure

N	NMISS	_PCHI_	DF_PCHI	P_PCHI	_LRCHI_	DF_LRCHI	P_LRCHI
762	0	20.9248	8	.007349898	25.9733	8	.001061424

Example 35.7: Cochran-Mantel-Haenszel Statistics

The data set Migraine contains hypothetical data for a clinical trial of migraine treatment. Subjects of both genders receive either a new drug therapy or a placebo. Their response to treatment is coded as 'Better' or 'Same'. The data are recorded as cell counts, and the number of subjects for each treatment and response combination is recorded in the variable Count.

```
data Migraine;
   input Gender $ Treatment $ Response $ Count @@;
   datalines;
female Active  Better 16   female Active  Same 11
female Placebo Better  5   female Placebo Same 20
male   Active  Better 12   male   Active  Same 16
male   Placebo Better  7   male   Placebo Same 19
;
```

The following PROC FREQ statements create a multiway table stratified by Gender, where Treatment forms the rows and Response forms the columns. The CMH option produces the Cochran-Mantel-Haenszel statistics. For this stratified 2×2 table, estimates of the common relative risk and the Breslow-Day test for homogeneity of the odds ratios are also displayed. The NOPRINT option suppresses the display of the contingency tables. These statements produce Output 35.7.1 through Output 35.7.3.

```
proc freq data=Migraine;
   tables Gender*Treatment*Response / cmh;
   weight Count;
   title 'Clinical Trial for Treatment of Migraine Headaches';
run;
```

Output 35.7.1 Cochran-Mantel-Haenszel Statistics

```
          Clinical Trial for Treatment of Migraine Headaches

                        The FREQ Procedure

          Summary Statistics for Treatment by Response
                    Controlling for Gender

     Cochran-Mantel-Haenszel Statistics (Based on Table Scores)

Statistic    Alternative Hypothesis    DF       Value      Prob
------------------------------------------------------------------
    1        Nonzero Correlation        1      8.3052    0.0040
    2        Row Mean Scores Differ     1      8.3052    0.0040
    3        General Association        1      8.3052    0.0040
```

For a stratified 2×2 table, the three CMH statistics displayed in Output 35.7.1 test the same hypothesis. The significant p-value (0.004) indicates that the association between treatment and response remains strong after adjusting for gender.

The CMH option also produces a table of relative risks, as shown in Output 35.7.2. Because this is a prospective study, the relative risk estimate assesses the effectiveness of the new drug; the "Cohort (Col1 Risk)" values are the appropriate estimates for the first column (the risk of improvement). The probability of migraine improvement with the new drug is just over two times the probability of improvement with the placebo.

The large p-value for the Breslow-Day test (0.2218) in Output 35.7.3 indicates no significant gender difference in the odds ratios.

Output 35.7.2 CMH Option: Relative Risks

Estimates of the Common Relative Risk (Row1/Row2)

Type of Study	Method	Value	95% Confidence Limits	
Case-Control (Odds Ratio)	Mantel-Haenszel	3.3132	1.4456	7.5934
	Logit	3.2941	1.4182	7.6515
Cohort (Col1 Risk)	Mantel-Haenszel	2.1636	1.2336	3.7948
	Logit	2.1059	1.1951	3.7108
Cohort (Col2 Risk)	Mantel-Haenszel	0.6420	0.4705	0.8761
	Logit	0.6613	0.4852	0.9013

Output 35.7.3 CMH Option: Breslow-Day Test

Breslow-Day Test for Homogeneity of the Odds Ratios

Chi-Square	1.4929
DF	1
Pr > ChiSq	0.2218

Example 35.8: Cochran-Armitage Trend Test

The data set Pain contains hypothetical data for a clinical trial of a drug therapy to control pain. The clinical trial investigates whether adverse responses increase with larger drug doses. Subjects receive either a placebo or one of four drug doses. An adverse response is recorded as Adverse='Yes'; otherwise, it is recorded as Adverse='No'. The number of subjects for each drug dose and response combination is contained in the variable Count.

```
data pain;
   input Dose Adverse $ Count @@;
   datalines;
0 No 26   0 Yes  6
1 No 26   1 Yes  7
2 No 23   2 Yes  9
3 No 18   3 Yes 14
4 No  9   4 Yes 23
;
```

The following PROC FREQ statements provide a trend analysis. The TABLES statement requests a table of Adverse by Dose. The MEASURES option produces measures of association, and the CL option produces confidence limits for these measures. The TREND option tests for a trend across the ordinal values of the variable Dose with the Cochran-Armitage test. The EXACT statement produces exact p-values for this test, and the MAXTIME= option terminates the exact computations if they do not complete within 60 seconds. The TEST statement computes an asymptotic test for Somers' $D(R|C)$.

The PLOTS= option requests a frequency plot for the table of Adverse by Dose. By default, PROC FREQ provides a bar chart for the frequency plot. The TWOWAY=STACKED option requests a stacked layout, where the bars correspond to the column variable (Dose) values, and the row variable (Adverse) frequencies are stacked within each bar.

```
ods graphics on;
proc freq data=Pain;
   tables Adverse*Dose / trend measures cl
          plots=freqplot(twoway=stacked);
   test smdrc;
   exact trend / maxtime=60;
   weight Count;
   title 'Clinical Trial for Treatment of Pain';
run;
ods graphics off;
```

Output 35.8.1 through Output 35.8.4 display the results of the analysis. The "Col Pct" values in Output 35.8.1 show the expected increasing trend in the proportion of adverse effects with the increasing dosage (from 18.75% to 71.88%). The corresponding frequency bar chart (Output 35.8.2) also shows this increasing trend.

Output 35.8.1 Contingency Table

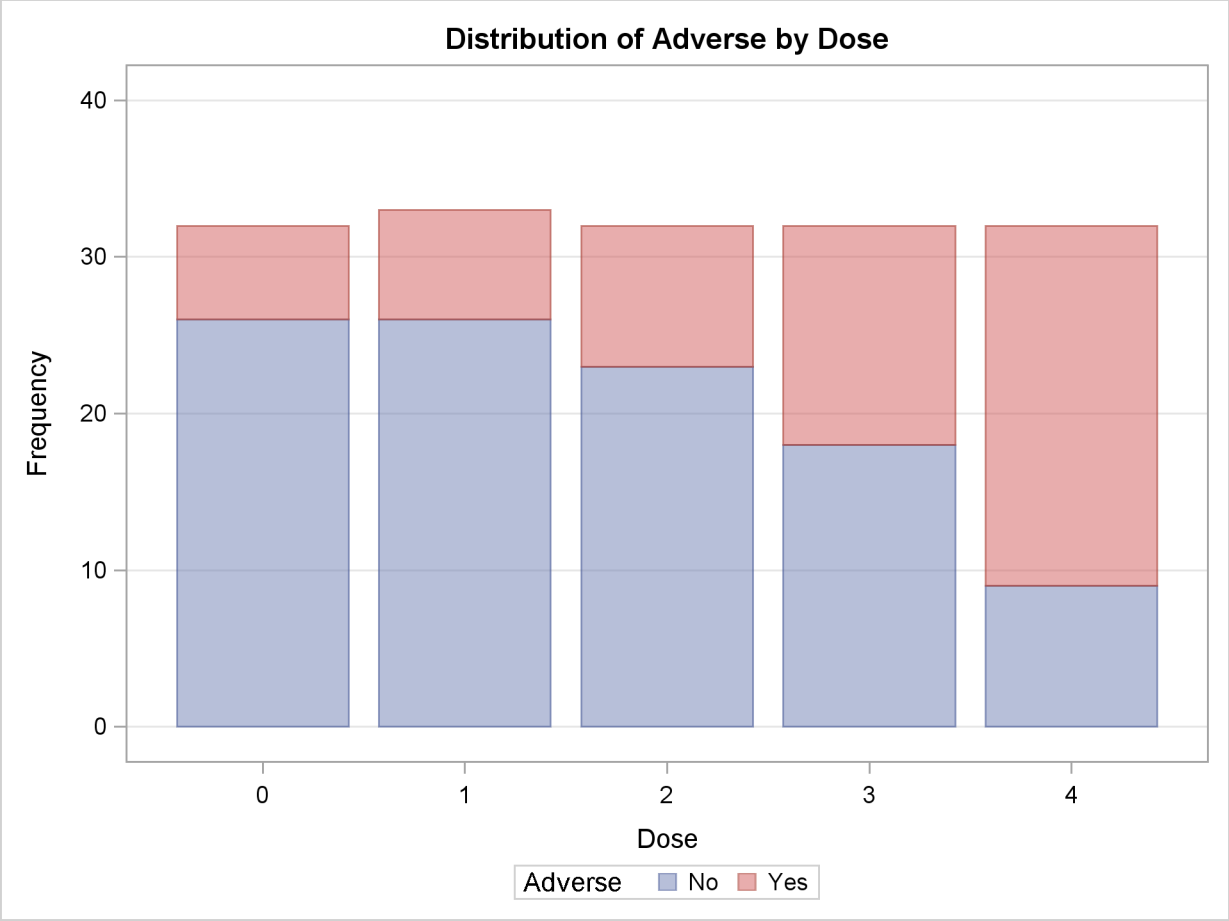
Clinical Trial for Treatment of Pain

The FREQ Procedure

Table of Adverse by Dose

Adverse / Dose	0	1	2	3	4	Total
No	26	26	23	18	9	102
	16.15	16.15	14.29	11.18	5.59	63.35
	25.49	25.49	22.55	17.65	8.82	
	81.25	78.79	71.88	56.25	28.13	
Yes	6	7	9	14	23	59
	3.73	4.35	5.59	8.70	14.29	36.65
	10.17	11.86	15.25	23.73	38.98	
	18.75	21.21	28.13	43.75	71.88	
Total	32	33	32	32	32	161
	19.88	20.50	19.88	19.88	19.88	100.00

Output 35.8.2 Stacked Bar Chart of Frequencies



Output 35.8.3 displays the measures of association produced by the MEASURES option. Somers' $D(R|C)$ measures the association treating the row variable (Adverse) as the response and the column variable (Dose) as a predictor. Because the asymptotic 95% confidence limits do not contain zero, this indicates a strong positive association. Similarly, the Pearson and Spearman correlation coefficients show evidence of a strong positive association, as hypothesized.

The Cochran-Armitage test (Output 35.8.4) supports the trend hypothesis. The small left-sided p-values for the Cochran-Armitage test indicate that the probability of the Row 1 level (Adverse='No') decreases as Dose increases or, equivalently, that the probability of the Row 2 level (Adverse='Yes') increases as Dose increases. The two-sided p-value tests against either an increasing or decreasing alternative. This is an appropriate hypothesis when you want to determine whether the drug has progressive effects on the probability of adverse effects but the direction is unknown.

Output 35.8.3 Measures of Association

Statistic	Value	ASE	95% Confidence Limits	
Gamma	0.5313	0.0935	0.3480	0.7146
Kendall's Tau-b	0.3373	0.0642	0.2114	0.4631
Stuart's Tau-c	0.4111	0.0798	0.2547	0.5675
Somers' D C|R	0.4427	0.0837	0.2786	0.6068
Somers' D R|C	0.2569	0.0499	0.1592	0.3547
Pearson Correlation	0.3776	0.0714	0.2378	0.5175
Spearman Correlation	0.3771	0.0718	0.2363	0.5178
Lambda Asymmetric C|R	0.1250	0.0662	0.0000	0.2547
Lambda Asymmetric R|C	0.2373	0.0837	0.0732	0.4014
Lambda Symmetric	0.1604	0.0621	0.0388	0.2821
Uncertainty Coefficient C|R	0.0515	0.0191	0.0140	0.0890
Uncertainty Coefficient R|C	0.1261	0.0467	0.0346	0.2175
Uncertainty Coefficient Symmetric	0.0731	0.0271	0.0199	0.1262

Somers' D R|C	
Somers' D R|C	0.2569
ASE	0.0499
95% Lower Conf Limit	0.1592
95% Upper Conf Limit	0.3547

Test of H0: Somers' D R|C = 0

ASE under H0	0.0499
Z	5.1511
One-sided Pr > Z	<.0001
Two-sided Pr > |Z|	<.0001

Output 35.8.4 Trend Test

Cochran-Armitage Trend Test	
Statistic (Z)	-4.7918
Asymptotic Test	
One-sided Pr < Z	<.0001
Two-sided Pr > |Z|	<.0001
Exact Test	
One-sided Pr <= Z	7.237E-07
Two-sided Pr >= |Z|	1.324E-06

Example 35.9: Friedman's Chi-Square Test

Friedman's test is a nonparametric test for treatment differences in a randomized complete block design. Each block of the design might be a subject or a homogeneous group of subjects. If blocks are groups of subjects, the number of subjects in each block must equal the number of treatments. Treatments are randomly assigned to subjects within each block. If there is one subject per block, then the subjects are repeatedly measured once under each treatment. The order of treatments is randomized for each subject.

In this setting, Friedman's test is identical to the ANOVA (row means scores) CMH statistic when the analysis uses rank scores (SCORES=RANK). The three-way table uses subject (or subject group) as the stratifying variable, treatment as the row variable, and response as the column variable. PROC FREQ handles ties by assigning midranks to tied response values. If there are multiple subjects per treatment in each block, the ANOVA CMH statistic is a generalization of Friedman's test.

The data set Hypnosis contains data from a study investigating whether hypnosis has the same effect on skin potential (measured in millivolts) for four emotions (Lehmann 1975, p. 264). Eight subjects are asked to display fear, joy, sadness, and calmness under hypnosis. The data are recorded as one observation per subject for each emotion.

```
data Hypnosis;
   length Emotion $ 10;
   input Subject Emotion $ SkinResponse @@;
   datalines;
1 fear 23.1  1 joy 22.7  1 sadness 22.5  1 calmness 22.6
2 fear 57.6  2 joy 53.2  2 sadness 53.7  2 calmness 53.1
3 fear 10.5  3 joy  9.7  3 sadness 10.8  3 calmness  8.3
4 fear 23.6  4 joy 19.6  4 sadness 21.1  4 calmness 21.6
5 fear 11.9  5 joy 13.8  5 sadness 13.7  5 calmness 13.3
6 fear 54.6  6 joy 47.1  6 sadness 39.2  6 calmness 37.0
7 fear 21.0  7 joy 13.6  7 sadness 13.7  7 calmness 14.8
8 fear 20.3  8 joy 23.6  8 sadness 16.3  8 calmness 14.8
;
```

In the following PROC FREQ statements, the TABLES statement creates a three-way table stratified by Subject and a two-way table; the variables Emotion and SkinResponse form the rows and columns of each table. The CMH2 option produces the first two Cochran-Mantel-Haenszel statistics, the option SCORES=RANK specifies that rank scores are used to compute these statistics, and the NOPRINT option suppresses the contingency tables. These statements produce Output 35.9.1 and Output 35.9.2.

```
proc freq data=Hypnosis;
   tables Subject*Emotion*SkinResponse /
          cmh2 scores=rank noprint;
run;
```

```
proc freq data=Hypnosis;
   tables Emotion*SkinResponse /
          cmh2 scores=rank noprint;
run;
```

Because the CMH statistics in Output 35.9.1 are based on rank scores, the Row Mean Scores Differ statistic is identical to Friedman's chi-square ($Q = 6.45$). The p-value of 0.0917 indicates that differences in skin potential response for different emotions are significant at the 10% level but not at the 5% level.

When you do not stratify by subject, the Row Mean Scores Differ CMH statistic is identical to a Kruskal-Wallis test and is not significant (p=0.9038 in Output 35.9.2). Thus, adjusting for subject is critical to reducing the background variation due to subject differences.

Output 35.9.1 CMH Statistics: Stratifying by Subject

Clinical Trial for Treatment of Pain

The FREQ Procedure

Summary Statistics for Emotion by SkinResponse
Controlling for Subject

Cochran-Mantel-Haenszel Statistics (Based on Rank Scores)

Statistic	Alternative Hypothesis	DF	Value	Prob
1	Nonzero Correlation	1	0.2400	0.6242
2	Row Mean Scores Differ	3	6.4500	0.0917

Output 35.9.2 CMH Statistics: No Stratification

Clinical Trial for Treatment of Pain

The FREQ Procedure

Summary Statistics for Emotion by SkinResponse

Cochran-Mantel-Haenszel Statistics (Based on Rank Scores)

Statistic	Alternative Hypothesis	DF	Value	Prob
1	Nonzero Correlation	1	0.0001	0.9933
2	Row Mean Scores Differ	3	0.5678	0.9038

Example 35.10: Cochran's Q Test

When a binary response is measured several times or under different conditions, Cochran's Q tests that the marginal probability of a positive response is unchanged across the times or conditions. When there are more than two response categories, you can use the CATMOD procedure to fit a repeated-measures model.

The data set Drugs contains data for a study of three drugs to treat a chronic disease (Agresti 2002). Forty-six subjects receive drugs A, B, and C. The response to each drug is either favorable ('F') or unfavorable ('U').

```
proc format;
   value $ResponseFmt 'F'='Favorable'
                      'U'='Unfavorable';
run;

data drugs;
   input Drug_A $ Drug_B $ Drug_C $ Count @@;
   datalines;
F F F  6   U F F  2
F F U 16   U F U  4
F U F  2   U U F  6
F U U  4   U U U  6
;
```

The following statements create one-way frequency tables of the responses to each drug. The AGREE option produces Cochran's Q and other measures of agreement for the three-way table. These statements produce Output 35.10.1 through Output 35.10.5.

```
proc freq data=Drugs;
   tables Drug_A Drug_B Drug_C / nocum;
   tables Drug_A*Drug_B*Drug_C / agree noprint;
   format Drug_A Drug_B Drug_C $ResponseFmt.;
   weight Count;
   title 'Study of Three Drug Treatments for a Chronic Disease';
run;
```

The one-way frequency tables in Output 35.10.1 provide the marginal response for each drug. For drugs A and B, 61% of the subjects reported a favorable response while 35% of the subjects reported a favorable response to drug C. Output 35.10.2 and Output 35.10.3 display measures of agreement for the 'Favorable' and 'Unfavorable' levels of drug A, respectively. McNemar's test shows a strong discordance between drugs B and C when the response to drug A is favorable.

Output 35.10.1 One-Way Frequency Tables

Study of Three Drug Treatments for a Chronic Disease

The FREQ Procedure

Drug_A	Frequency	Percent
Favorable	28	60.87
Unfavorable	18	39.13

Drug_B	Frequency	Percent
Favorable	28	60.87
Unfavorable	18	39.13

Drug_C	Frequency	Percent
Favorable	16	34.78
Unfavorable	30	65.22

Output 35.10.2 Measures of Agreement for Drug A Favorable

McNemar's Test	
Statistic (S)	10.8889
DF	1
Pr > S	0.0010

Simple Kappa Coefficient	
Kappa	-0.0328
ASE	0.1167
95% Lower Conf Limit	-0.2615
95% Upper Conf Limit	0.1960

Output 35.10.3 Measures of Agreement for Drug A Unfavorable

McNemar's Test	
Statistic (S)	0.4000
DF	1
Pr > S	0.5271

Simple Kappa Coefficient	
Kappa	-0.1538
ASE	0.2230
95% Lower Conf Limit	-0.5909
95% Upper Conf Limit	0.2832

Output 35.10.4 displays the overall kappa coefficient. The small negative value of kappa indicates no agreement between drug B response and drug C response.

Cochran's Q is statistically significant (p=0.0144 in Output 35.10.5), which leads to rejection of the hypothesis that the probability of favorable response is the same for the three drugs.

Output 35.10.4 Overall Measures of Agreement

Overall Kappa Coefficient	
Kappa	−0.0588
ASE	0.1034
95% Lower Conf Limit	−0.2615
95% Upper Conf Limit	0.1439

Test for Equal Kappa Coefficients	
Chi-Square	0.2314
DF	1
Pr > ChiSq	0.6305

Output 35.10.5 Cochran's *Q* Test

Cochran's Q, for Drug_A by Drug_B by Drug_C	
Statistic (Q)	8.4706
DF	2
Pr > Q	0.0145

References

Agresti, A. (1992), "A Survey of Exact Inference for Contingency Tables," *Statistical Science*, 7(1), 131–177.

Agresti, A. (2002), *Categorical Data Analysis*, Second Edition, New York: John Wiley & Sons.

Agresti, A. (2007), *An Introduction to Categorical Data Analysis*, Second Edition, New York: John Wiley & Sons.

Agresti, A. and Coull, B. A. (1998), "Approximate is Better than "Exact" for Interval Estimation of Binomial Proportions," *The American Statistician*, 52, 119–126.

Agresti, A., Mehta, C. R. and Patel, N. R. (1990), "Exact Inference for Contingency Tables with Ordered Categories," *Journal of the American Statistical Association*, 85, 453–458.

Agresti, A. and Min, Y. (2001), "On Small-Sample Confidence Intervals for Parameters in Discrete Distributions," *Biometrics*, 57, 963–971.

Agresti, A., Wackerly, D., and Boyett, J. M. (1979), "Exact Conditional Tests for Cross-Classifications: Approximation of Attained Significance Levels," *Psychometrika*, 44, 75–83.

Barker, L., Rolka, H., Rolka, D., and Brown, C. (2001), "Equivalence Testing for Binomial Random Variables: Which Test to Use?," *The American Statistician*, 55, 279–287.

Berger, J. O. (1985), *Statistical Decision Theory and Bayesian Analysis*, Second Edition, New York: Springer-Verlag.

Birch, M. W. (1965), "The Detection of Partial Association, II: The General Case," *Journal of the Royal Statistical Society, B*, 27, 111–124.

Bishop, Y., Fienberg, S. E., and Holland, P. W. (1975), *Discrete Multivariate Analysis: Theory and Practice*, Cambridge, MA: MIT Press.

Bowker, A. H. (1948), "Bowker's Test for Symmetry," *Journal of the American Statistical Association*, 43, 572–574.

Breslow, N. E. (1996), "Statistics in Epidemiology: The Case-Control Study," *Journal of the American Statistical Association*, 91, 14–26.

Breslow, N. E. and Day, N. E. (1980), *Statistical Methods in Cancer Research, Volume I: The Analysis of Case-Control Studies*, IARC Scientific Publications, No. 32, Lyon, France: International Agency for Research on Cancer.

Breslow, N. E. and Day, N. E. (1987), *Statistical Methods in Cancer Research, Volume II: The Design and Analysis of Cohort Studies*, IARC Scientific Publications, No. 82, Lyon, France: International Agency for Research on Cancer.

Bross, I. D. J. (1958), "How to Use Ridit Analysis," *Biometrics*, 14, 18–38.

Brown, L. D., Cai, T. T., and DasGupta, A. (2001), "Interval Estimation for a Binomial Proportion," *Statistical Science* 16, 101–133.

Brown, M. B. and Benedetti, J. K. (1977), "Sampling Behavior of Tests for Correlation in Two-Way Contingency Tables," *Journal of the American Statistical Association*, 72, 309–315.

Chow, S., Shao, J., and Wang, H. (2003), *Sample Size Calculations in Clinical Research*, Boca Raton, FL: CRC Press.

Cicchetti, D. V. and Allison, T. (1971), "A New Procedure for Assessing Reliability of Scoring EEG Sleep Recordings," *American Journal of EEG Technology*, 11, 101–109.

Clopper, C. J., and Pearson, E. S. (1934), "The Use of Confidence or Fiducial Limits Illustrated in the Case of the Binomial," *Biometrika* 26, 404–413.

Cochran, W. G. (1950), "The Comparison of Percentages in Matched Samples," *Biometrika*, 37, 256–266.

Cochran, W. G. (1954), "Some Methods for Strengthening the Common χ^2 Tests," *Biometrics*, 10, 417–451.

Collett, D. (1991), *Modelling Binary Data*, London: Chapman & Hall.

Cohen, J. (1960), "A Coefficient of Agreement for Nominal Scales," *Educational and Psychological Measurement*, 20, 37–46.

Drasgow, F. (1986), "Polychoric and Polyserial Correlations" in *Encyclopedia of Statistical Sciences*, vol. 7, ed. S. Kotz and N. L. Johnson, New York: John Wiley & Sons, 68–74.

Dunnett, C. W., and Gent, M. (1977), "Significance Testing to Establish Equivalence Between Treatments, with Special Reference to Data in the Form of 2×2 Tables," *Biometrics*, 33, 593–602.

Farrington, C. P., and Manning, G. (1990), "Test Statistics and Sample Size Formulae for Comparative Binomial Trials with Null Hypothesis of Non-zero Risk Difference or Non-unity Relative Risk," *Statistics in Medicine*, 9, 1447–1454.

Fienberg, S. E. (1980), *The Analysis of Cross-Classified Data*, Second Edition, Cambridge, MA: MIT Press.

Fleiss, J. L., Levin, B., and Paik, M. C. (2003), *Statistical Methods for Rates and Proportions*, Third Edition, New York: John Wiley & Sons.

Fleiss, J. L. and Cohen, J. (1973), "The Equivalence of Weighted Kappa and the Intraclass Correlation Coefficient as Measures of Reliability," *Educational and Psychological Measurement*, 33, 613–619.

Fleiss, J. L., Cohen, J., and Everitt, B. S. (1969), "Large-Sample Standard Errors of Kappa and Weighted Kappa," *Psychological Bulletin*, 72, 323–327.

Freeman, G. H. and Halton, J. H. (1951), "Note on an Exact Treatment of Contingency, Goodness of Fit and Other Problems of Significance," *Biometrika*, 38, 141–149.

Gail, M. and Mantel, N. (1977), "Counting the Number of $r \times c$ Contingency Tables with Fixed Margins," *Journal of the American Statistical Association*, 72, 859–862.

Gart, J. J. (1971), "The Comparison of Proportions: A Review of Significance Tests, Confidence Intervals and Adjustments for Stratification," *Review of the International Statistical Institute*, 39(2), 148–169.

Goodman, L. A. and Kruskal, W. H. (1979), *Measures of Association for Cross Classification*, New York: Springer-Verlag.

Greenland, S. and Robins, J. M. (1985), "Estimators of the Mantel-Haenszel Variance Consistent in Both Sparse Data and Large-Strata Limiting Models," *Biometrics*, 42, 311–323.

Haldane, J. B. S. (1955), "The Estimation and Significance of the Logarithm of a Ratio of Frequencies," *Annals of Human Genetics*, 20, 309–314.

Hauck, W. W. and Anderson, S. (1986), "A Comparison of Large-Sample Confidence Interval Methods for the Difference of Two Binomial Probabilities," *The American Statistician*, 40, 318–322.

Hirji, K. F. (2006), *Exact Analysis of Discrete Data*, Boca Raton, FL: Chapman & Hall/CRC.

Hirji, K. F., Vollset, S. E., Reis, I. M., and Afifi, A. A. (1996), "Exact Tests for Interaction in Several 2×2 Tables," *Journal of Computational and Graphical Statistics*, 5, 209–224.

Hollander, M. and Wolfe, D. A. (1999), *Nonparametric Statistical Methods*, Second Edition, New York: John Wiley & Sons.

Jones, M. P., O'Gorman, T. W., Lemka, J. H., and Woolson, R. F. (1989), "A Monte Carlo Investigation of Homogeneity Tests of the Odds Ratio Under Various Sample Size Configurations," *Biometrics*, 45, 171–181.

Kendall, M. (1955), *Rank Correlation Methods*, Second Edition, London: Charles Griffin and Co.

Kendall, M. and Stuart, A. (1979), *The Advanced Theory of Statistics*, vol. 2, New York: Macmillan.

Kleinbaum, D. G., Kupper, L. L., and Morgenstern, H. (1982), *Epidemiologic Research: Principles and Quantitative Methods*, Research Methods Series, New York: Van Nostrand Reinhold.

Landis, R. J., Heyman, E. R., and Koch, G. G. (1978), "Average Partial Association in Three-way Contingency Tables: A Review and Discussion of Alternative Tests," *International Statistical Review*, 46, 237–254.

Leemis, L. M. and Trivedi, K. S. (1996), "A Comparison of Approximate Interval Estimators for the Bernoulli Parameter," *The American Statistician*, 50, 63–68.

Lehmann, E. L. (1975), *Nonparametrics: Statistical Methods Based on Ranks*, San Francisco: Holden-Day.

Liebetrau, A. M. (1983), *Measures of Association, Quantitative Application in the Social Sciences*, vol. 32, Beverly Hills: Sage Publications.

Mack, G. A. and Skillings, J. H. (1980), "A Friedman-Type Rank Test for Main Effects in a Two-Factor ANOVA," *Journal of the American Statistical Association*, 75, 947–951.

Mantel, N. (1963), "Chi-square Tests with One Degree of Freedom: Extensions of the Mantel-Haenszel Procedure," *Journal of the American Statistical Association*, 58, 690–700.

Mantel, N. and Haenszel, W. (1959), "Statistical Aspects of the Analysis of Data from Retrospective Studies of Disease," *Journal of the National Cancer Institute*, 22, 719–748.

Margolin, B. H. (1988), "Test for Trend in Proportions," in *Encyclopedia of Statistical Sciences*, vol. 9, ed. S. Kotz and N. L. Johnson, New York: John Wiley & Sons, 334–336.

McNemar, Q. (1947), "Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages," *Psychometrika*, 12, 153–157.

Mehta, C. R. and Patel, N. R. (1983), "A Network Algorithm for Performing Fisher's Exact Test in $r \times c$ Contingency Tables," *Journal of the American Statistical Association*, 78, 427–434.

Mehta, C. R., Patel, N. R., and Gray, R. (1985), "On Computing an Exact Confidence Interval for the Common Odds Ratio in Several 2 × 2 Contingency Tables," *Journal of the American Statistical Association*, 80, 969–973.

Mehta, C. R., Patel, N. R., and Senchaudhuri, P. (1991), "Exact Stratified Linear Rank Tests for Binary Data," *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, ed. E.M. Keramidas, 200–207.

Mehta, C. R., Patel, N. R., and Tsiatis, A. A. (1984), "Exact Significance Testing to Establish Treatment Equivalence with Ordered Categorical Data," *Biometrics*, 40, 819–825.

Narayanan, A. and Watts, D. (1996), "Exact Methods in the NPAR1WAY Procedure," in *Proceedings of the Twenty-First Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc., 1290–1294.

Newcombe, R. G. (1998), "Two-sided Confidence Intervals for the Single Proportion: Comparison of Seven Methods," *Statistics in Medicine*, 17, 857–872.

Newcombe, R. G. (1998), "Interval Estimation for the Difference Between Independent Proportions: Comparison of Eleven Methods," *Statistics in Medicine*, 17, 873–890.

Olsson, U. (1979), "Maximum Likelihood Estimation of the Polychoric Correlation Coefficient," *Psychometrika*, 12, 443–460.

Pirie, W. (1983), "Jonckheere Tests for Ordered Alternatives," in *Encyclopedia of Statistical Sciences*, vol. 4, ed. S. Kotz and N. L. Johnson, New York: John Wiley & Sons, 315–318.

Radlow, R. and Alf, E. F. (1975), "An Alternate Multinomial Assessment of the Accuracy of the Chi-Square Test of Goodness of Fit," *Journal of the American Statistical Association*, 70, 811–813.

Robins, J. M., Breslow, N., and Greenland, S. (1986), "Estimators of the Mantel-Haenszel Variance Consistent in Both Sparse Data and Large-Strata Limiting Models," *Biometrics*, 42, 311–323.

Santner, T. J. and Snell, M. K. (1980), "Small-Sample Confidence Intervals for p1-p2 and p1/p2 in 2 × 2 Contingency Tables," *Journal of the American Statistical Association*, 75, 386–394.

Schuirmann, D. J. (1987), "A Comparison of the Two One-Sided Tests Procedure and the Power Approach for Assessing the Equivalence of Average Bioavailability," *Journal of Pharmacokinetics and Biopharmaceutics*, 15, 657–680.

Schuirmann, D. J. (1999), "Confidence Interval Methods for Bioequivalence Testing with Binomial Endpoints," *Proceedings of the Biopharmaceutical Section, ASA*, 227–232.

Snedecor, G. W. and Cochran, W. G. (1989), *Statistical Methods*, Eighth Edition, Ames: Iowa State University Press.

Somers, R. H. (1962), "A New Asymmetric Measure of Association for Ordinal Variables," *American Sociological Review*, 27, 799–811.

Stokes, M. E., Davis, C. S., and Koch, G. G. (2000), *Categorical Data Analysis Using the SAS System*, Second Edition, Cary, NC: SAS Institute Inc.

Tarone, R. E. (1985), "On Heterogeneity Tests Based on Efficient Scores," *Biometrika*, 72, 1, 91–95.

Theil, H. (1972), *Statistical Decomposition Analysis*, Amsterdam: North-Holland Publishing Company.

Thomas, D. G. (1971), "Algorithm AS-36. Exact Confidence Limits for the Odds Ratio in a 2×2 Table," *Applied Statistics*, 20, 105–110.

Valz, P. D. and Thompson, M. E. (1994), "Exact Inference for Kendall's S and Spearman's Rho with Extensions to Fisher's Exact Test in $r \times c$ Contingency Tables," *Journal of Computational and Graphical Statistics*, 3(4), 459–472.

van Elteren, P. H. (1960), "On the Combination of Independent Two-Sample Tests of Wilcoxon," *Bulletin of the International Statistical Institute*, 37, 351–361.

Vollset, S. E., Hirji, K. F., and Elashoff, R. M. (1991), "Fast Computation of Exact Confidence Limits for the Common Odds Ratio in a Series of 2×2 Tables," *Journal of the American Statistical Association*, 86, 404–409.

Wilson, E. B. (1927), "Probable Inference, the Law of Succession, and Statistical Inference," *Journal of the American Statistical Association*, 22, 209–212.

Woolf, B. (1955), "On Estimating the Relationship Between Blood Group and Disease," *Annals of Human Genetics*, 19, 251–253.

Zelen, M. (1971), "The Analysis of Several 2×2 Contingency Tables," *Biometrika*, 58, 129–137.

Chapter 36

The GAM Procedure

Contents

Overview: GAM Procedure **1834**
Getting Started: GAM Procedure **1834**
Syntax: GAM Procedure **1838**
PROC GAM Statement 1838
BY Statement 1840
CLASS Statement 1840
FREQ Statement 1841
MODEL Statement 1842
OUTPUT Statement 1846
SCORE Statement 1847
Details: GAM Procedure **1848**
Missing Values 1848
Nonparametric Regression 1848
Additive Models and Generalized Additive Models 1849
Backfitting and Local Scoring Algorithms 1850
Smoothers 1852
Selection of Smoothing Parameters 1853
Confidence Intervals for Smoothers 1855
Distribution Family and Canonical Link 1856
Dispersion Parameter 1857
Computational Resources 1857
Forms of Additive Models 1860
Estimates from PROC GAM 1860
ODS Table Names 1861
ODS Graphics 1861
Examples: GAM Procedure **1862**
Example 36.1: Generalized Additive Model with Binary Data 1862
Example 36.2: Poisson Regression Analysis of Component Reliability . . . 1871
Example 36.3: Comparing PROC GAM with PROC LOESS 1877
References **1889**

Overview: GAM Procedure

The GAM procedure fits generalized additive models as those models are defined by Hastie and Tibshirani (1990). This procedure provides an array of powerful tools for data analysis, based on nonparametric regression and smoothing techniques.

Nonparametric regression relaxes the usual assumption of linearity and enables you to uncover structure in the relationship between the independent variables and the dependent variable that might otherwise be missed. The SAS System provides many procedures for nonparametric regression, such as the LOESS procedure for local regression and the TPSPLINE procedure for thin-plate smoothing splines. The generalized additive models fit by the GAM procedure combine the following:

- an additivity assumption (Stone 1985) that enables relatively many nonparametric relationships to be explored simultaneously
- the distributional flexibility of generalized linear models (Nelder and Wedderburn 1972)

Thus, you can use the GAM procedure when you have multiple independent variables whose effect you want to model nonparametrically, or when the dependent variable is not normally distributed. See the section "Nonparametric Regression" on page 1848 for more details on the form of generalized additive models.

The GAM procedure does the following:

- provides nonparametric estimates for additive models
- supports the use of multidimensional data
- supports multiple SCORE statements
- fits both generalized semiparametric additive models and generalized additive models
- enables you to choose a particular model by specifying the model degrees of freedom or smoothing parameter
- supports graphical displays produced through ODS Graphics

Getting Started: GAM Procedure

The following example illustrates the use of the GAM procedure to explore in a nonparametric way how two factors affect a response. The data come from a study (Sockett et al. 1987) of the factors affecting patterns of insulin-dependent diabetes mellitus in children. The objective is to investigate the dependence of the level of serum C-peptide on various other factors in order to understand

the patterns of residual insulin secretion. The response measurement is the logarithm of C-peptide concentration (pmol/ml) at diagnosis, and the predictor measurements are age and base deficit (a measure of acidity).

```
title 'Patterns of Diabetes';
data diabetes;
    input Age BaseDeficit CPeptide @@;
    logCP = log(CPeptide);
datalines;
5.2    -8.1  4.8   8.8  -16.1  4.1  10.5   -0.9  5.2
10.6   -7.8  5.5  10.4  -29.0  5.0   1.8  -19.2  3.4
12.7  -18.9  3.4  15.6  -10.6  4.9   5.8   -2.8  5.6
1.9   -25.0  3.7   2.2   -3.1  3.9   4.8   -7.8  4.5
7.9   -13.9  4.8   5.2   -4.5  4.9   0.9  -11.6  3.0
11.8   -2.1  4.6   7.9   -2.0  4.8  11.5   -9.0  5.5
10.6  -11.2  4.5   8.5   -0.2  5.3  11.1   -6.1  4.7
12.8   -1.0  6.6  11.3   -3.6  5.1   1.0   -8.2  3.9
14.5   -0.5  5.7  11.9   -2.0  5.1   8.1   -1.6  5.2
13.8  -11.9  3.7  15.5   -0.7  4.9   9.8   -1.2  4.8
11.0  -14.3  4.4  12.4   -0.8  5.2  11.1  -16.8  5.1
5.1    -5.1  4.6   4.8   -9.5  3.9   4.2  -17.0  5.1
6.9    -3.3  5.1  13.2   -0.7  6.0   9.9   -3.3  4.9
12.5  -13.6  4.1  13.2   -1.9  4.6   8.9  -10.0  4.9
10.8  -13.5  5.1
;
```

The following statements perform the desired analysis. The PROC GAM statement invokes the procedure and specifies the diabetes data set as input. The MODEL statement specifies logCP as the response variable and requests that univariate smoothing splines with the default of 4 degrees of freedom be used to model the effect of Age and BaseDeficit.

```
title 'Patterns of Diabetes';
proc gam data=diabetes;
    model logCP = spline(Age) spline(BaseDeficit);
run;
```

The results are shown in Figure 36.1 and Figure 36.2.

Figure 36.1 Summary Statistics

```
                          Patterns of Diabetes

                            The GAM Procedure
                        Dependent Variable: logCP
        Smoothing Model Component(s): spline(Age) spline(BaseDeficit)

                        Summary of Input Data Set

                 Number of Observations                  43
                 Number of Missing Observations           0
                 Distribution                      Gaussian
                 Link Function                     Identity
```

Figure 36.1 *continued*

Iteration Summary and Fit Statistics	
Final Number of Backfitting Iterations	5
Final Backfitting Criterion	5.542745E-10
The Deviance of the Final Estimate	0.4180791724

Figure 36.1 shows two tables. The first table summarizes the input data set and the distributional family used for the model; the second table summarizes the convergence criterion for backfitting.

Figure 36.2 Analysis of Model

Regression Model Analysis
Parameter Estimates

Parameter	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1.48141	0.05120	28.93	<.0001
Linear(Age)	0.01437	0.00437	3.28	0.0024
Linear(BaseDeficit)	0.00807	0.00247	3.27	0.0025

Smoothing Model Analysis
Fit Summary for Smoothing Components

Component	Smoothing Parameter	DF	GCV	Num Unique Obs
Spline(Age)	0.995582	3.000000	0.011675	37
Spline(BaseDeficit)	0.995299	3.000000	0.012437	39

Smoothing Model Analysis
Analysis of Deviance

Source	DF	Sum of Squares	Chi-Square	Pr > ChiSq
Spline(Age)	3.00000	0.150761	12.2605	0.0065
Spline(BaseDeficit)	3.00000	0.081273	6.6095	0.0854

Figure 36.2 displays summary statistics for the model. It consists of three tables. The first is the "Parameter Estimates" table for the parametric part of the model. It indicates that the linear trends for both Age and BaseDeficit are highly significant. The second table is the summary of smoothing components of the nonparametric part of the model. By default, each smoothing component has approximately 4 degrees of freedom (DF). For univariate spline components, one DF is taken up by the (parametric) linear part of the model, so the remaining approximate DF is 3, and the main point of this table is to present the smoothing parameter values that yield this DF for each component. Finally, the third table is the "Analysis of Deviance" table for the nonparametric component of the model.

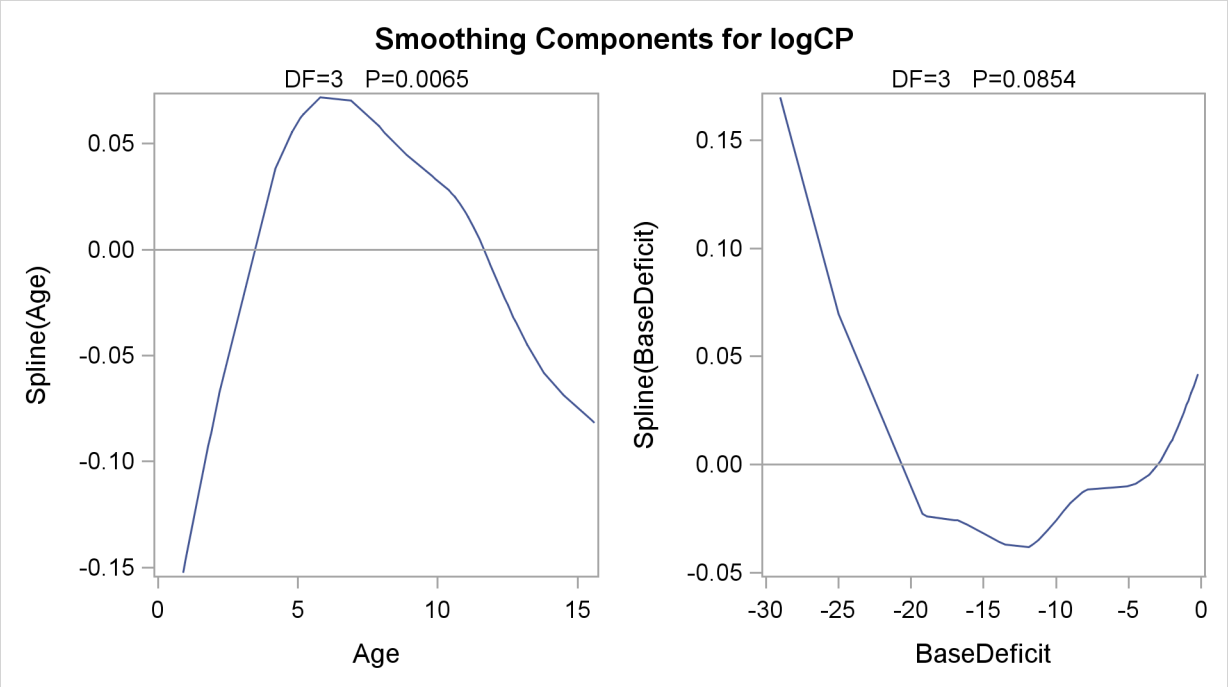
In order to explore the overall shape of the relationship between each factor and the response, use the graphics features of PROC GAM to plot the partial predictions, as in the following statements:

```
ods graphics on;

proc gam data=diabetes;
    model logCP = spline(Age) spline(BaseDeficit);
run;

ods graphics off;
```

These graphical displays are requested by specifying the ODS GRAPHICS statement. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the GAM procedure, see the section "ODS Graphics" on page 1861.

Both plots (Figure 36.3) show a strong quadratic pattern, with a possible indication of higher-order behavior. Further investigation is required to determine whether these patterns are real or not.

Figure 36.3 Partial Predictions for Each Predictor



Syntax: GAM Procedure

The following statements are available in PROC GAM:

PROC GAM < *options* > **;**
CLASS *variable*<(*options*)><*variable*<(*options*)>···></*options*> **;**
MODEL *dependent*</*options*> = < *PARAM*(*effects*) > < *smoothing effects* >< /*options* > **;**
SCORE *data=SAS-data-set out=SAS-data-set* **;**
OUTPUT < *out=SAS-data-set* > *keyword=prefix* <··· *keyword=prefix* > **;**
BY *variables* **;**
FREQ *variable* **;**

The syntax of the GAM procedure is similar to that of other regression procedures in the SAS System. The PROC GAM and MODEL statements are required. The SCORE statement can appear multiple times; all other statements appear only once.

The syntax for PROC GAM is described in the following sections in alphabetical order after the description of the PROC GAM statement.

PROC GAM Statement

PROC GAM < *options* > **;**

The PROC GAM statement invokes the procedure. You can specify the following options.

DATA=*SAS-data-set*
specifies the SAS data set to be read by PROC GAM. The default value is the most recently created data set.

DESCENDING
DESC
reverses the sorting order of the classification variable (specified in the CLASS statement). If both the DESCENDING and ORDER= options are specified, PROC GAM orders the categories according to the ORDER= option and then reverses that order. This option has the same effect as the classification variable option DESCENDING in the CLASS statement and the response variable option DESCENDING in the MODEL statement.

ORDER=DATA | FORMATTED | FREQ | INTERNAL
specifies the sorting order for the levels of all classification variables (specified in the CLASS statement). This ordering determines which parameters in the model correspond to each level in the data. Note that the ORDER= option in the CLASS statement and the ORDER= response variable option in the MODEL statement override the ORDER= option in the PROC GAM statement.

PLOTS <(*global-plot-options*)> <= *plot-request* <(*options*)>>

PLOTS <(*global-plot-options*)> <=(*plot-request* <(*options*)> <··· *plot-request* <(*options*)>>)>

controls the plots produced through ODS Graphics. When you specify only one plot request, you can omit the parentheses around the plot request. Here are some examples:

```
plots=all

plots=components(commonaxes)

plots(unpack)=components(commonaxes clm)
```

You must enable ODS Graphics before requesting plots as shown in the following example. With ODS Graphics enabled, the output graph by default is a panel of multiple plots of partial prediction curves of smoothing components, if PLOTS is not specified or no options are specified for PLOTS. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

```
ods graphics on;

proc gam data=test plots(unpack)=components(commonaxes clm);
   model z=spline(x) spline(y);
run;

ods graphics off;
```

Global Plot Options

The *global-plot-options* apply to all plots generated by the GAM procedure, unless altered by a *specific-plot-option*.

UNPACK

specifies that multiple smoothing component plots that are collected into graphics panels be displayed separately. Use this option if you want to access individual smoothing component plots within the panel.

Specific Plot Options

The following listing describes the specific plots and their options.

ALL

requests that all plots are produced.

NONE

suppresses all plots.

COMPONENTS | COMPONENT <(*components-options*)>

requests the SmoothingComponentPlot that displays a panel of smoothing component plots. The following *components-options* are available:

ADDITIVE requests that the additive component plots are produced for spline effects. The additive component plots combine the linear trend and the nonparametric prediction for each spline effect if its linear part is not specified explicitly in the MODEL statement.

CLM includes confidence limits in the smoothing component plots. By default, 95% confidence limits are produced, but you can change the significance level by specifying the ALPHA= option in the MODEL statement. Note that producing these limits can be computationally intensive for large data sets.

COMMONAXES specifies that smoothing component plots use a common vertical axis. This enables you to visually judge relative effect size.

UNPACK specifies that the smoothing components be displayed individually.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC GAM to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the GAM procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index for the BY variables by using the DATASETS procedure.

For more information about the BY statement, see in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide.*

CLASS Statement

CLASS *variable<(options)><variable<(options)>· · · ></options>* **;**

The CLASS statement names the classification variables to be used in the analysis. The CLASS statement must precede the MODEL statement. You can specify various *options* for each variable be enclosing them in parentheses after the variable name. You can also specify global *options* for the CLASS statement by placing them after a slash (/). Global *options* are applied to all the

variables specified in the CLASS statement. If you specify more than one CLASS statement, the global *options* specified on any one CLASS statement apply to all CLASS statements. However, individual CLASS variable *options* override the global *options*.

DESCENDING

DESC

reverses the sorting order of the classification variable. If both the DESCENDING and ORDER= options are specified, PROC GAM orders the categories according to the ORDER= option and then reverses that order.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the categories of categorical variables. This ordering determines which parameters in the model correspond to each level in the data. When the default ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. The following table shows how PROC GAM interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine-dependent. For more information on sorting order, see the chapter on the SORT procedure in the *SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

TRUNCATE< *=n* >

specifies the length *n* of CLASS variable values to use in determining CLASS variable levels. If you specify TRUNCATE without the length *n*, the first 16 characters of the formatted values are used. When formatted values are longer than 16 characters, you can use this option to revert to the levels as determined in releases previous to SAS 9. The default is to use the full formatted length of the CLASS variable. The TRUNCATE option is available only as a global option.

FREQ Statement

FREQ *variable* **;**

The FREQ statement names a variable that provides frequencies for each observation in the DATA= data set. Specifically, if *n* is the value of the FREQ variable for a given observation, then that observation is used *n* times.

The analysis produced by using a FREQ statement reflects the expanded number of observations. You can produce the same analysis (without the FREQ statement) by first creating a new data set that contains the expanded number of observations. For example, if the value of the FREQ variable is 5 for the first observation, the first five observations in the new data set are identical. Each observation in the old data set is replicated n_i times in the new data set, where n_i is the value of the FREQ variable for that observation.

If the value of the FREQ variable is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

The FREQ statement is not available when a loess smoother is included in the model.

MODEL Statement

MODEL *dependent<(options)>=<PARAM(effects)><smoothing effects> </options>* **;**

MODEL *event/trials=<PARAM(effects)> <smoothing effects> </options>* **;**

The MODEL statement specifies the dependent variable and the independent effects you want to use to model its values. Specify the independent parametric variables inside the parentheses of PARAM( ). The parametric variables can be either classification variables or continuous variables. Classification variables must be declared with a CLASS statement. Interactions between variables can also be included as parametric effects. Multiple PARAM() statements are allowed in the MODEL statement. The syntax for the specification of effects is the same as for the GLM procedure (Chapter 39, "The GLM Procedure").

Only continuous variables can be specified in smoothing effects. Any number of smoothing effects can be specified, as follows:

Smoothing Effect	Meaning
SPLINE(variable <, DF=number>)	fit a smoothing spline with the variable and with DF=number
LOESS(variable <, DF=number>)	fit a local regression with the variable and with DF=number
SPLINE2(variable1, variable2 <,DF=number>)	fit a bivariate thin-plate smoothing spline with variable1 and variable2 and with DF=number

The number specified in the DF= option must be positive. If you specify neither the DF= option nor the METHOD=GCV in the MODEL statement, then the default used is DF=4. Note that for univariate spline components, a degree of freedom is removed by default to account for the linear portion of the model, so the value displayed in the "Fit Summary" and "Analysis of Deviance tables" will be one less than the value you specify.

Both parametric effects and smoothing effects are optional, but at least one of them must be present.

If only parametric variables are present, PROC GAM fits a parametric linear model by using the terms inside the parentheses of PARAM( ). If only smoothing effects are present, PROC GAM fits a

nonparametric additive model. If both types of effect are present, PROC GAM fits a semiparametric model by using the parametric effects as the linear part of the model.

Table 36.1 shows how to specify various models for a dependent variable y and independent variables x, x1, and x2.

Table 36.1 Syntax for Common GAM Models

Type of Model	Syntax for `model`	Mathematical Form
Parametric	`y=param(x1 x2)`	$E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$
Nonparametric	`y=spline(x)`	$E(Y) = \beta_0 + \beta_1 x + s(x)$
Nonparametric	`y=loess(x)`	$E(Y) = \beta_0 + s(x)$
Semiparametric	`y=spline(x1) param(x2)`	$E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + s(x_1)$
Additive	`y=spline(x1) spline(x2)`	$E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + s_1(x_1) + s_2(x_2)$
Thin-plate spline	`y=spline2(x1,x2)`	$E(Y) = \beta_0 + s(x_1, x_2)$

Response Variable Options

Response variable options determine how the GAM procedure models probabilities for binary data.

You can specify the following options by enclosing them in parentheses after the response variable. See the section "CLASS Statement" on page 1840 for more detail.

DESCENDING
DESC

reverses the order of the response categories. If both the DESCENDING and ORDER= options are specified, PROC GAM orders the response categories according to the ORDER= option and then reverses that order.

EVENT='*category***' | *keyword***

specifies the event category for the binary response model. PROC GAM models the probability of the event category. You can specify the value (formatted, if a format is applied) of the event category in quotes, or you can specify one of the following keywords. The default is EVENT=FIRST.

FIRST

designates the first ordered category as the event.

LAST

designates the last ordered category as the event.

One of the most common sets of response levels is $\{0, 1\}$, with 1 representing the event for which the probability is to be modeled. Consider the example where Y takes the value 1 and 0 for event and nonevent, respectively, and X is the explanatory variable. To specify the value 1 as the event category, using the following MODEL statement:

```
model Y (event='1') = X;
```

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sort order for the levels of the response variable. When ORDER=FORMATTED (the default) for numeric variables for which you have supplied no explicit format (that is, for which there is no corresponding FORMAT statement in the current PROC GAM run or in the DATA step that created the data set), the levels are ordered by their internal (numeric) value. If you specify the ORDER= option in the MODEL statement and the ORDER= option in the CLASS statement, the former takes precedence. The following table shows the interpretation of the ORDER= values:

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For the FORMATTED and INTERNAL values, the sort order is machine-dependent.

For more information about sort order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

Model Options

ALPHA=*number*

specifies the significance level α of the confidence limits on the final nonparametric component estimates when you request confidence limits to be included in the output data set. Specify *number* as a value between 0 and 1. The default value is 0.05. See the section "OUTPUT Statement" on page 1846 for more information about the OUTPUT statement.

ANODEV=*type*

specifies the *type* of method to be used to produce the "Analysis of Deviance" table for smoothing effects. The available choices are as follows:

REFIT specifies that PROC GAM perform χ^2 tests by fitting nested GAM models. This is the default choice if you do not specify the ANODEV= option. This choice requires fitting separate GAM models where one smoothing term is omitted from each model.

NOREFIT specifies that PROC GAM perform approximate tests of smoothing effects. To test each smoothing effect, a weighted least squares model is fitted to the remaining parametric part of the model while keeping other nonlinear smoothers fixed. For details, see Hastie (1991). This choice requires only a single GAM fitting to be performed, which reduces the time of the procedure.

NONE requests that the procedure not produce the "Analysis of Deviance" table for smoothing effects.

DIST=*distribution-id*

LINK=*distribution-id*

specifies the distribution family used in the model. The choices for *distribution-id* are displayed in Table 36.2. See "Distribution Family and Canonical Link" on page 1856 for more information.

Table 36.2 Distribution Families for GAM Models

DIST=	Distribution	Response Data Type
GAUSSIAN | GAUS | NORM	Normal (Gaussian)	continuous variables
BINOMIAL | LOGI | BIN	Binomial	binary variables
POISSON | POIS | LOGL	Poisson	nonnegative discrete variables
GAMMA | GAMM	Gamma	positive continuous variables
IGAUSSIAN | IGAU | INVG	Inverse gaussian	positive continuous variables

Canonical link functions are used with those distributions. Although theoretically, alternative links are possible, with nonparametric regression models the final fit is relatively insensitive to the precise choice of link functions. Therefore, only the canonical link for each distribution family is implemented in PROC GAM. The loess smoother is not available for DIST=BINOMIAL when the number of trials is greater than 1.

EPSILON=*number*

specifies the convergence criterion for the backfitting algorithm. The default value is 1E−8.

EPSSCORE=*number*

specifies the convergence criterion for the local scoring algorithm. The default value is 1E−8.

ITPRINT

produces an iteration summary table for the smoothing effects when doing backfitting and local scoring.

MAXITER=*number*

specifies the maximum number of iterations for the backfitting algorithm. The default value is 50.

MAXITSCORE=*number*

specifies the maximum number of iterations for the local scoring algorithm. The default value is 100.

METHOD=GCV

specifies that the value of the smoothing parameter should be selected by generalized cross validation. If you specify both METHOD=GCV and the DF= option for the smoothing effects, the user-specified DF is used, and the METHOD=GCV option is ignored. See the section "Selection of Smoothing Parameters" on page 1853 for more details on the GCV method.

OUTPUT Statement

OUTPUT *OUT=SAS-data-set < keyword < =prefix > ··· keyword < =prefix >>* **;**

The OUTPUT statement creates a new SAS data set containing diagnostic measures calculated after fitting the model.

All the variables in the original data set are included in the new data set, along with the variables created by specifying *keywords* in the OUTPUT statement. These new variables contain the values of a variety of statistics and diagnostic measures that are calculated for each observation in the data set. If no *keywords* are present, the OUT= data set contains only the original data set and predicted values. The predicted values include the linear predictor for the response and each smoothing term in the model. When you specify a distribution family with the DIST= or LINK= option in the MODEL statement, predicted response values after applying the inverse link function are also included. Predicted values are computed for observations with missing response values whose values of the specified explanatory variables are nonmissing, and whose values of the specified smoothing variables are within the smoothing ranges of the fitted model.

Details on the specifications in the OUTPUT statement are as follows.

OUT=*SAS-data-set*
specifies the name of the new data set to contain the diagnostic measures. This specification is required.

keyword < =prefix >
specifies the statistics to include in the output data set. The keywords and the statistics they represent are as follows:

PREDICTED	predicted values for each smoothing component and overall predicted values on the response scale at design points
LINP	linear prediction values on the link scale at design points
UCLM	upper confidence limits for each predicted smoothing component
LCLM	lower confidence limits for each predicted smoothing component
ADIAG	diagonal element of the hat matrix associated with the observation for each smoothing spline component
RESIDUAL	residual standardized by its weights
STD	standard deviation of the prediction for each smoothing component
ALL	all statistics in this list

The names of the new variables that contain the statistics are formed by concatenating the user supplied *prefix* and the corresponding variable names. If you do not specify a *prefix*, the names are formed by using default prefixes listed in the following table.

The default prefixes for the new variables are as follows:

Keyword	Prefix
PRED	P_
LINP	LINP_
UCLM	UCLM_
LCLM	LCLM_
ADIAG	ADIAG_
RESID	R_
STD	STD_ (for spline)
	STDP_ (for loess)

For example, suppose that you have a dependent variable y and an independent smoothing variable x, and you specify the keywords PRED=MyP_ and ADIAG=MyA_. In this case, in addition to the variables in the input data set, the output SAS data set will contain the variables MyP_y, MyP_x, and MyA_x. If the keywords PRED and ADIAG are specified without prefixes, the output SAS data set will contain the variables P_y, P_x, and ADIAG_x.

SCORE Statement

SCORE *DATA=SAS-data-set* *OUT=SAS-data-set* **;**

The SCORE statement calculates predicted values for a new data set. All the variables in the named data set are included in the OUT= data set, along with the predicted values. The predicted values consist of predicted responses after the inverse link function transformation, predicted values of all smoothing terms, and predicted values on the link scale. Predicted values are computed for observations with missing response values whose values of the specified explanatory variables are nonmissing, and whose values of the specified smoothing variables are within the smoothing ranges of the fitted model. The predicted variables use the same naming convention as the OUTPUT statement. If you have multiple data sets to predict, you can specify multiple SCORE statements.

The following options must be specified in the SCORE statement:

DATA=*SAS-data-set*

specifies an input SAS data set containing all the variables included in independent effects in the MODEL statement. The predicted response is computed for each observation in the SCORE DATA= data set.

OUT=*SAS-data-set*

specifies the name of the SAS data set to contain the predictions.

Details: GAM Procedure

Missing Values

PROC GAM excludes any observation with missing values for an explanatory variable or dependent variable when fitting a model. However, all observations in the input data set are included in the output data set with an OUTPUT statement.

Nonparametric Regression

Nonparametric regression relaxes the usual assumption of linearity and enables you to explore the data more flexibly, uncovering structure in the data that might otherwise be missed.

However, many forms of nonparametric regression do not perform well when the number of independent variables in the model is large. The sparseness of data in this setting causes the variances of the estimates to be unacceptably large unless the sample size is extremely large. The problem of rapidly increasing variance for increasing dimensionality is sometimes referred to as the "curse of dimensionality." Interpretability is another problem with nonparametric regression based on kernel and smoothing spline estimates. These estimates contain information about the relationship between the dependent and independent variables, and the information is often difficult to comprehend.

To overcome these difficulties, Stone (1985) proposed additive models. These models estimate an additive approximation to the multivariate regression function. The benefits of an additive approximation are at least twofold. First, since each of the individual additive terms is estimated by using a univariate smoother, the curse of dimensionality is avoided, at the cost of not being able to approximate universally. Second, estimates of the individual terms explain how the dependent variable changes with the corresponding independent variables.

To extend the additive model to a wide range of distribution families, Hastie and Tibshirani (1990) proposed generalized additive models. These models enable the mean of the dependent variable to depend on an additive predictor through a nonlinear link function. The models permit the response probability distribution to be any member of the exponential family of distributions. Many widely used statistical models belong to this general class; they include additive models for Gaussian data, nonparametric logistic models for binary data, and nonparametric log-linear models for Poisson data.

Additive Models and Generalized Additive Models

This section describes the methodology and the fitting procedure behind generalized additive models.

Let Y be a response random variable and $X_1, X_2, \cdots, X_p$ be a set of predictor variables. A regression procedure can be viewed as a method for estimating the expected value of Y given the values of $X_1, X_2, \cdots, X_p$. The standard linear regression model assumes a linear form for the dependency of Y on X:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

where $E(\epsilon) = 0$ and $\text{Var}(\epsilon) = \sigma^2$. Given a sample, estimates of $\beta_0, \beta_1, \cdots, \beta_p$ are usually obtained by the least squares method.

The additive model generalizes the linear model by modeling the dependency as

$$Y = s_0 + s_1(X_1) + s_2(X_2) + \cdots + s_p(X_p) + \epsilon$$

where $s_i(X), i = 1, 2, \ldots, p$, are smooth functions, $E(\epsilon) = 0$ and $\text{Var}(\epsilon) = \sigma^2$.

In order to be estimable, the smooth functions s_i have to satisfy standardized conditions such as $E\left(s_i(X_i)\right) = 0$. These functions are not given a parametric form but instead are estimated in a nonparametric fashion.

While traditional linear models and additive models can be used in most statistical data analysis, there are types of problems for which they are not appropriate. For example, the normal distribution might not be adequate for modeling discrete responses such as counts or bounded responses such as proportions.

Generalized additive models address these difficulties, extending additive models to many other distributions besides just the normal. Thus, generalized additive models can be applied to a much wider range of data analysis problems.

Like generalized linear models, generalized additive models consist of a random component, an additive component, and a link function relating the two components. The response Y, the random component, is assumed to have exponential family density

$$f_Y(y; \theta, \phi) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right\}$$

where θ is called the natural parameter and ϕ is the scale parameter. The mean of the response variable μ is related to the set of covariates $X_1, X_2, \cdots, X_p$ by a link function g. The quantity

$$\eta = s_0 + \sum_{i=1}^{p} s_i(X_i)$$

defines the additive component, where $s_1(\,), \cdots, s_p(\,)$ are smooth functions, and the relationship between μ and η is defined by $g(\mu) = \eta$. The most commonly used link function is the canonical link, for which $\eta = \theta$.

Generalized additive models and generalized linear models can be applied in similar situations, but they serve different analytic purposes. Generalized linear models emphasize estimation and inference for the parameters of the model, while generalized additive models focus on exploring data nonparametrically. Generalized additive models are more suitable for exploring the data and visualizing the relationship between the dependent variable and the independent variables.

Backfitting and Local Scoring Algorithms

Much of the development and notation in this section follows Hastie and Tibshirani (1986). Consider the estimation of the smoothing terms $s_0, s_1(\,), \cdots, s_p(\,)$ in the additive model

$$\eta(X) = s_0 + \sum_{i=1}^{p} s_i(X_i)$$

where $E\left(s_i(X_i)\right) = 0$ for every i. Since the algorithm for additive models is the basis for fitting generalized additive models, the algorithm for additive models is discussed first.

Many ways are available to approach the formulation and estimation of additive models. The backfitting algorithm is a general algorithm that can fit an additive model with any regression-type fitting mechanisms.

Define the jth set of partial residuals as

$$R_j = Y - s_0 - \sum_{k \neq j} s_k(X_k)$$

then $E(R_j|X_j) = s_j(X_j)$. This observation provides a way to estimate each smoothing function $s_j(\,)$ given estimates $\{\hat{s}_i(\,), i \neq j\}$ for all the others. The resulting iterative procedure is known as the backfitting algorithm (Friedman and Stuetzle 1981). The following formulation is taken from Hastie and Tibshirani (1986).

The Backfitting Algorithm

The unweighted form of the backfitting algorithm is as follows:

1. **Initialization:**
   $s_0 = E(Y), s_1^{(1)} = s_2^{(1)} = \cdots = s_p^{(1)} = 0, m = 0$
2. **Iterate:**
   $m = m + 1$;
   for $j = 1$ to p do:
   $R_j = Y - s_0 - \sum_{k=1}^{j-1} s_k^{(m)}(X_k) - \sum_{k=j+1}^{p} s_k^{(m-1)}(X_k)$;
   $s_j^{(m)} = E(R_j|X_j)$;
3. **Until:**
   $RSS = \frac{1}{n}\left\| Y - s_0 - \sum_{j=1}^{p} s_j^{(m)}(X_j) \right\|^2$ fails to decrease, or satisfies the convergence criterion.

In the preceding notation, $s_j^{(m)}(\ )$ denotes the estimate of $s_j(\ )$ at the mth iteration. It can be shown that with many smoothers (including linear regression, univariate and bivariate splines, and combinations of these), RSS never increases at any step. This implies that the algorithm always converges (Hastie and Tibshirani, 1986). Note, however, that for distributions other than Gaussian, numerical instabilities with weights can cause convergence problems. Even when the algorithm converges, the individual functions need not be unique, since dependence among the covariates can lead to more than one representation for the same fitted surface.

A weighted backfitting algorithm has the same form as for the unweighted case, except that the smoothers are weighted. In PROC GAM, weights are used with non-Gaussian data in the local scoring procedure described later in this section.

The GAM procedure uses the following condition as the convergence criterion for the backfitting algorithm:

$$\frac{\sum_{j=1}^{n}\sum_{i=1}^{k}\left(s_i^{(m-1)}(x_j)-s_i^{(m)}(x_j)\right)^2}{1+\sum_{j=1}^{n}\sum_{i=1}^{k}\left(s_i^{(m-1)}(x_j)\right)^2}\leq\epsilon$$

where $\epsilon = 10^{-8}$ by default; you can change this with the EPSILON= option in the MODEL statement.

The algorithm described so far fits only additive models. The algorithm for generalized additive models is a little more complicated. Generalized additive models extend generalized linear models in the same manner that additive models extend linear regression models—that is, by replacing the form $\alpha+\sum_j X_j\beta_j$ with the additive form $\alpha+\sum_j f_j(X_j)$. See "Generalized Linear Models Theory" on page 1964 in Chapter 37, "The GENMOD Procedure," for more information.

PROC GAM fits generalized additive models by using a modified form of adjusted dependent variable regression, as described for generalized linear models in McCullagh and Nelder (1989), with the additive predictor taking the role of the linear predictor. Hastie and Tibshirani (1986) call this the *local scoring algorithm*. Important components of this algorithm depend on the link function for each distribution, as shown in the following table.

Distribution	**Link**	**Adjusted Dependent** (Z)	**Weights** (w)
Normal	μ	y	1
Binomial	$\log\left(\frac{\mu}{n-\mu}\right)$	$\eta+(y-\mu)/n\mu(1-\mu)$	$n\mu(1-\mu)$
Gamma	$-1/\mu$	$\eta+(y-\mu)/\mu^2$	μ^2
Poisson	$\log(\mu)$	$\eta+(y-\mu)/\mu$	μ
Inverse gaussian	$1/\mu^2$	$\eta-2(y-\mu)/\mu^3$	$\mu^3/4$

Once the distribution and hence these quantities are defined, the local scoring algorithm proceeds as follows.

The General Local Scoring Algorithm

1. **Initialization:**
   $s_i = g(E(y)), s_1^0 = s_2^0 = \cdots = s_p^0 = 0, m = 0$

2. **Iterate:**
   $m = m + 1$;
   Form the predictor η_i, mean μ_i, weights w_i, and adjusted dependent variable z_i based on the previous iteration:

$$\begin{aligned}
\eta_i^{(m-1)} &= s_0 + \sum_{j=1}^{p} s_j^{(m-1)}(x_{ij}) \\
\mu_i^{(m-1)} &= g^{-1}\left(\eta_i^{(m-1)}\right) \\
w_i &= \left(V_i^{(m-1)}\right)^{-1} \cdot \left(\frac{\partial \mu_i}{\partial \eta_i}\right)^2_{(m-1)} \\
z_i &= \eta_i^{(m-1)} + \left(y_i - \mu_i^{(m-1)}\right) \cdot \left(\frac{\partial \mu_i}{\partial \eta_i}\right)_{(m-1)}
\end{aligned}$$

   where $V_i^{(m-1)}$ is the variance of Y at $\mu_i^{(m-1)}$. Fit an additive model to Z by using the backfitting algorithm with weights W to obtain estimated functions $s_j^{(m)}(\,)$;

3. **Until:**
   The convergence criterion is satisfied or the deviance fails to decrease. The deviance is an extension to generalized linear models of the RSS; see "Goodness of Fit" on page 1970 in Chapter 37, "The GENMOD Procedure," for a definition.

The GAM procedure uses the following condition as the convergence criterion for local scoring:

$$\frac{\sum_{j=1}^{n}\sum_{i=1}^{k} w(x_i)\left(s_i^{(m-1)}(x_j) - s_i^{(m)}(x_j)\right)^2}{\sum_{j=1}^{n}\sum_{i=1}^{k} w(x_i) + \sum_{j=1}^{n}\sum_{i=1}^{k} w(x_i)\left(s_i^{(m-1)}(x_j)\right)^2} \leq \epsilon^s$$

where $\epsilon^s = 10^{-8}$ by default; you can change this with the EPSSCORE= option in the MODEL statement.

The estimating procedure for generalized additive models consists of two loops. Inside each step of the local scoring algorithm (outer loop), a weighted backfitting algorithm (inner loop) is used until convergence or until the RSS fails to decrease. Then, based on the estimates from this weighted backfitting algorithm, a new set of weights is calculated and the next iteration of the scoring algorithm starts. The scoring algorithm stops when the convergence criterion is satisfied or the deviance of the estimates stops decreasing.

Smoothers

A smoother is a tool for summarizing the trend of a response measurement Y as a function of one or more predictor measurements $X_1, \cdots, X_p$. It produces an estimate of the trend that is less variable than Y itself. An important property of a smoother is its nonparametric nature. It does not assume a rigid form for the dependence of Y on $X_1, \cdots, X_p$. This section gives a brief overview of the smoothers that can be used with the GAM procedure.

Cubic Smoothing Spline

A smoothing spline is the solution to the following optimization problem: among all functions $\eta(x)$ with two continuous derivatives, find one that minimizes the penalized least square

$$\sum_{i=1}^{n}(y_i - \eta(x_i))^2 + \lambda \int_a^b \left(\eta''(t)\right)^2 \mathrm{d}t$$

where λ is a fixed constant and $a \le x_1 \le \cdots \le x_n \le b$. The first term measures closeness to the data while the second term penalizes curvature in the function. It can be shown that there exists an explicit, unique minimizer, and that minimizer is a natural cubic spline with knots at the unique values of x_i.

The value $\lambda/(1+\lambda)$ is the *smoothing parameter*. When λ is large, the smoothing parameter is close to 1, producing a smoother curve; small values of λ, corresponding to smoothing parameters near 0, are apt to produce rougher curves, more nearly interpolating the data.

Local Regression (Experimental)

Local regression was proposed by Cleveland, Devlin, and Grosse (1988). The idea of local regression is that at a predictor x, the regression function $\eta(x)$ can be locally approximated by the value of a function in some specified parametric class. Such a local approximation is obtained by fitting a regression surface to the data points within a chosen neighborhood of the point x. A weighted least squares algorithm is used to fit linear functions of the predictors at the centers of neighborhoods. The radius of each neighborhood is chosen so that the neighborhood contains a specified percentage of the data points. The smoothing parameter for the local regression procedure, which controls the smoothness of the estimated curve, is the fraction of the data in each local neighborhood. Data points in a given local neighborhood are weighted by a smooth decreasing function of their distance from the center of the neighborhood. See Chapter 50, "The LOESS Procedure," for more details.

Thin-Plate Smoothing Spline

The thin-plate smoothing spline is a multivariate version of the cubic smoothing spline. The theoretical foundations for the thin-plate smoothing spline are described in Duchon (1976, 1977) and Meinguet (1979). Further results and applications are given in Wahba and Wendelberger (1980). See Chapter 89, "The TPSPLINE Procedure," for more details.

Selection of Smoothing Parameters

CV and GCV

The smoothers discussed here have a single smoothing parameter. In choosing the smoothing parameter, cross validation can be used. Cross validation works by leaving points (x_i, y_i) out one at a time, estimating the squared residual for smooth function at x_i based on the remaining $n-1$ data

points, and choosing the smoother to minimize the sum of those squared residuals. This mimics the use of training and test samples for prediction. The cross validation function is defined as

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{\eta}_{\lambda}^{(-i)}(x_i)\right)^2$$

where $\hat{\eta}_{\lambda}^{(-i)}(x_i)$ indicates the fit at x_i, computed by leaving out the ith data point. The quantity $nCV(\lambda)$ is sometimes called the prediction sum of squares, or $PRESS$ (Allen 1974).

All of the smoothers fit by the GAM procedure can be formulated as a linear combination of the sample responses

$$\hat{\eta}(x) = A(\lambda)Y$$

for some matrix $A(\lambda)$, which depends on λ. (The matrix $A(\lambda)$ depends on x and the sample data as well, but this dependence is suppressed in the preceding equation.) Let a_{ii} be the ith diagonal element of $A(\lambda)$. Then the CV function can be expressed as

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{(y_i - \hat{\eta}_{\lambda}(x_i))}{1 - a_{ii}}\right)^2$$

In most cases, it is very time-consuming to compute the quantity a_{ii} individually. To solve this computational problem, Wahba (1990) has proposed the generalized cross validation function (GCV) that can be used to solve a wide variety of problems involving selection of a parameter to minimize the prediction risk.

The GCV function is defined as

$$GCV(\lambda) = \frac{n\sum_{i=1}^{n}(y_i - \hat{\eta}_{\lambda}(x_i))^2}{(n - \mathrm{tr}(A(\lambda)))^2}$$

The GCV formula simply replaces the a_{ii} with $\mathrm{tr}(A(\lambda))/n$. Therefore, it can be viewed as a weighted version of CV. In most of the cases of interest, GCV is closely related to CV but much easier to compute. Specify the METHOD=GCV option in the MODEL statement in order to use the GCV function to choose the smoothing parameters.

Degrees of Freedom

The estimated GAM model can be expressed as

$$\hat{\eta}(X) = \hat{s}_0 + \sum_{i=1}^{p} A_i(y,\lambda)Y$$

Because the weights are calculated based on previous iteration during the local scoring iteration, the matrices A_i might depend on Y for non-Gaussian data. However, for the final iteration, the A_i matrix for the spline smoothers has the same role as the projection matrix in linear regression; therefore, nonparametric degrees of freedom (DF) for the ith spline smoother can be defined as

$$DF(\text{spline smoother}) = \mathrm{tr}(A_i(y,\lambda))$$

For loess smoothers A_i is not symmetric and so is not a projection matrix. In this case PROC GAM uses

$$DF(\text{loess smoother}) = \text{tr}(A_i(y,\lambda)' A_i(y,\lambda))$$

The GAM procedure gives you the option of specifying the degrees of freedom for each individual smoothing component. If you choose a particular value for the degrees of freedom, then during every local scoring iteration the procedure will search for a corresponding smoothing parameter lambda that yields the specified value or comes as close as possible. The final estimate for the smoother during this local scoring iteration will be based on this lambda. Note that for univariate spline components, an additional degree of freedom is removed by default to account for the linear portion of the model, so the value displayed in the "Fit Summary" and "Analysis of Deviance" tables will be one less than the value you specify.

Confidence Intervals for Smoothers

In the GAM procedure, curvewise confidence intervals for smoothing splines and pointwise confidence intervals for loess are provided in the output data set.

Curvewise Confidence Interval for Smoothing Spline

Viewing the spline model as a Bayesian model, Wahba (1983) proposes Bayesian confidence intervals for smoothing spline estimates as follows:

$$\hat{s}_\lambda(x_i) \pm z_{\alpha/2}\sqrt{\hat{\sigma}^2 a_{ii}(\lambda)}$$

where $a_{ii}(\lambda)$ is the ith diagonal element of the $A(\lambda)$ matrix and $z_{\alpha/2}$ is the $1-\alpha/2$ quantile of the standard normal distribution. The confidence intervals are interpreted as intervals "across the function" as opposed to pointwise intervals.

Suppose that you fit a spline estimate to experimental data that consists of a true function f and a random error term, ϵ_i. In repeated experiments, it is likely that about $100(1-\alpha)\%$ of the confidence intervals cover the corresponding true values, although some values are covered every time and other values are not covered by the confidence intervals most of the time. This effect is more pronounced when the true response curve or surface has small regions of particularly rapid change.

Pointwise Confidence Interval for Loess Smoothers

As defined in Cleveland, Devlin, and Grosse (1988), a standardized residual for a loess smoother follows a t distribution with ρ degrees of freedom, where ρ is called the "lookup degrees of freedom," defined as

$$\rho = \delta_1^2/\delta_2$$

where $\delta_1 = \mathrm{tr}\left((I - A(\lambda))^T (I - A(\lambda))\right)$ and $\delta_2 = \mathrm{tr}\left((I - A(\lambda))^T (I - A(\lambda))\right)^2$. Therefore an approximate pointwise confidence interval at x_i is

$$\hat{s}_\lambda(x_i) \pm t_{\alpha/2,\rho}\hat{\sigma}(x_i)$$

where $t_{\alpha/2,\rho}$ is the $1-\alpha/2$ quantile of the t distribution with ρ degrees of freedom and $\hat{\sigma}(x_i)$ is the estimate of the standard deviation.

Distribution Family and Canonical Link

In general, there is not just one reasonable link function for a given response variable distribution. For parametric models, the choice of link function can lead to substantively different estimates and tests. However, the inherent flexibility of nonparametric models makes them less likely to be sensitive to the precise choice of link function. Thus, for simplicity and computational efficiency, the GAM procedure uses only the canonical link for each distribution, as discussed in the following sections.

The Gaussian Model

With this model, the link function is the identity function, and the generalized additive model is the additive model.

The Binomial Model

A binomial response model assumes that the proportion of successes Y is such that Y has a $Bi(n, p(x))$ distribution. $Bi(n, p(x))$ refers to the binomial distribution with the parameters n and $p(x)$. Often the data are binary, in which case $n = 1$. The canonical link is

$$g(p) = \log\frac{p}{n-p} = \eta$$

By default, PROC GAM models the probability of response level with *lower ordered value*. Ordered values are assigned to response levels in ascending sorted order and are displayed in the "Response Profiles" table. For binary data, if your event category has a higher Ordered Value, then by default the nonevent is modeled. The effect of modeling the nonevent is to change the signs of the estimated coefficients for linear terms in the model for the event. You can change which probability is modeled by specifying the EVENT=, DESCENDING, or ORDER= response variable options in the MODEL statement.

The Poisson Model

The link function for the Poisson model is the log function. Assuming that the mean of the Poisson distribution is $\mu(x)$, the dependence of $\mu(x)$ and independent variables $x_1, \cdots, x_k$ is

$$g(\mu) = \log(\mu) = \eta$$

The Gamma Model

Let the mean of the gamma distribution be $\mu(x)$. The canonical link function for the gamma distribution is $-1/\mu(x)$. Therefore, the relationship between $\mu(x)$ and the independent variables $x_1, \cdots, x_k$ is

$$g(\mu) = -\frac{1}{\mu} = \eta$$

The Inverse Gaussian Model

Let the mean of the inverse gaussian distribution be $\mu(x)$. The canonical link function for inverse gaussian distribution is $1/\mu^2$. Therefore, the relationship between $\mu(x)$ and the independent variables $x_1, \cdots, x_k$ is

$$g(\mu) = \frac{1}{\mu^2} = \eta$$

Dispersion Parameter

Continuous distributions in the exponential family (Gaussian, gamma, and inverse gaussian) have a dispersion parameter that can be estimated by the scaled deviance. For these continuous response distributions, PROC GAM incorporates this dispersion parameter estimate into standard errors of the parameter estimates, prediction standard errors of spline components, and chi-square statistics. The discrete distributions used in GAM (binomial and Poisson) do not have a dispersion parameter. For more details on the distributions, dispersion parameter, and deviance, see "Generalized Linear Models Theory" on page 1964 in Chapter 37, "The GENMOD Procedure."

Computational Resources

Since PROC GAM implements a doubly iterative method (inner backfitting iterations within each local scoring iteration), data are accessed multiple times in performing a fit. To expedite the data access, PROC GAM keeps the data used in the analysis in memory.

Let

$$
\begin{aligned}
n &= \text{number of observations used in the analysis} \\
p_r &= \text{number of parametric variables} \\
p_s &= \text{number of univariate spline smoothers} \\
p_l &= \text{number of loess smoothers} \\
p_b &= \text{number of bivariate thin-plate spline smoothers} \\
p &= p_r + p_s + p_l + p_b \\
p_n &= p_s + p_l + p_b \\
m &= \text{maximum number of iterations for the backfitting algorithm}
\end{aligned}
$$

In addition to the space to store the data ($8np$ bytes), the minimum working space (in bytes) needed for fitting a model using PROC GAM is

$$(16+8p_r)(n+2p_r)+(160+48p+16p_s+8p_b+8p_l)n+8p+32p_b+32p_s+8m+8n+(4n+4)p_s+4$$

plus $80 + 120n + 8n^2 + 8p_b$ bytes of memory for fitting bivariate thin-plate smoothing spline variables and $48n + 16p_l$ bytes of memory for fitting loess variables. If confidence limits are requested, additional memory is required.

It is difficult to provide accurate estimates of the time required to fit a GAM model. Both the backfitting algorithm and the local scoring algorithm are iterative techniques whose convergence rates depend on the particular data being analyzed. Furthermore, the time required depends on the types of smoothers that you specify, as well as on the inferential information you request.

You can estimate the time required for problems with a larger number of observations by observing the time required for smaller problems and then using the following growth rules (obtained using by simulations) that show that the time required grows proportionally with the following:

- n^3 when at least one bivariate thin-plate spline is used
- $n^{3/2}$ when only loess smoothers are used
- n when only univariate smoothing splines are used

For additive models (models with Gaussian response distribution) with a fixed number of observations, the time required is roughly proportional to $p_n^{3/2}$. For generalized additive models (models with non-Gaussian distributions), the computation time grows more rapidly as p_n increases. This is harder to quantify as it depends on the distribution family and the number of iterations required for the local scoring algorithm to converge.

Figure 36.4 Feasible Problem Sizes for Different Smoothers

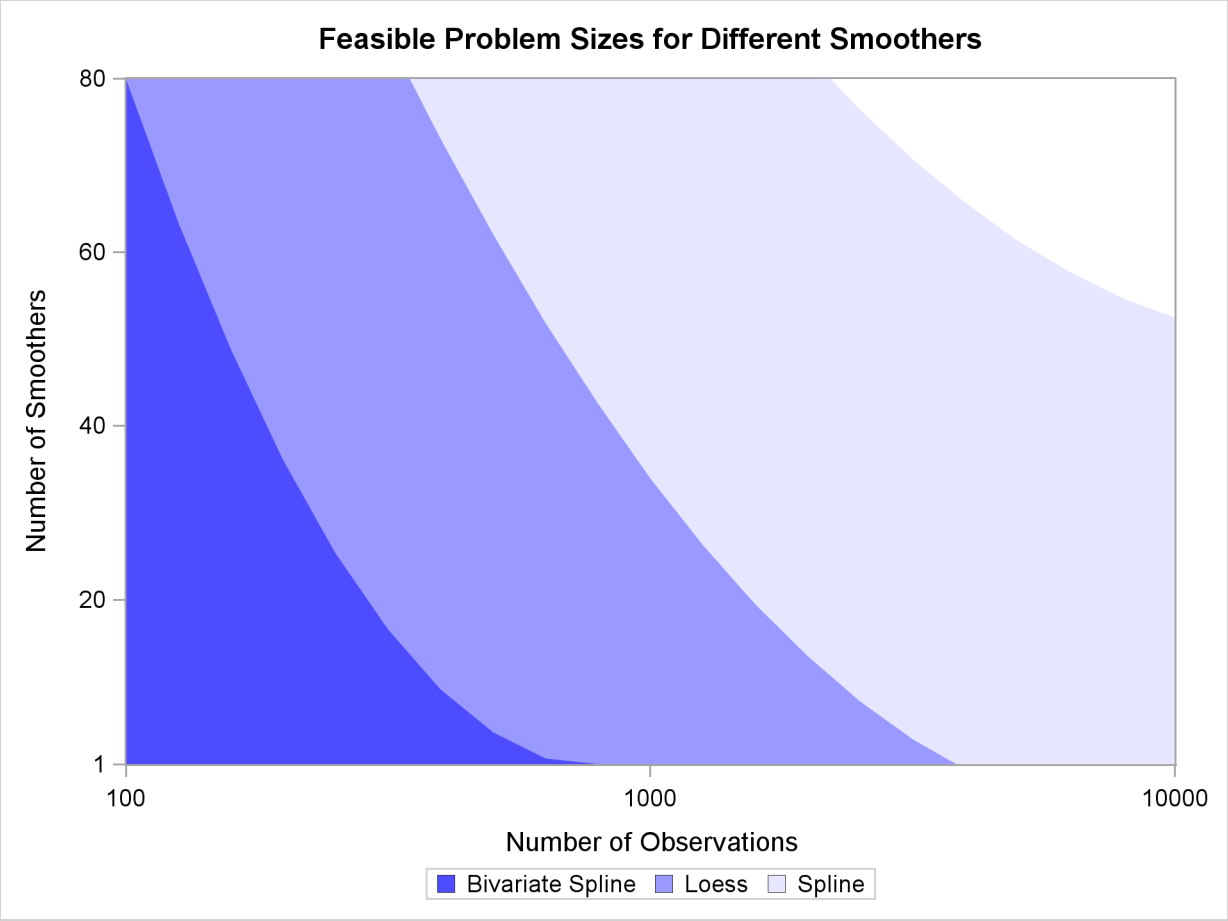


Figure 36.4 shows a rough estimation of feasible sizes for the smoothers that you can use, as a function of the number of observations and number of smoothing components. This figure depicts the regions where you can expect a single fit of an additive model to finish within a few minutes on a typical Pentium 4 system.

Note that the times reflected in Figure 36.4 are based on fitting additive models (no local scoring iterations) when no analysis of deviance or confidence limits are computed. The time required for fitting generalized additive models grows proportionally with the number of the local scoring iterations. Furthermore, analysis of deviance (if you do not request the fast approximations with the ANODEV option) requires fitting multiple GAM models as each smoothing component is omitted sequentially, and so the time estimates need to be multiplied by the number of smoothing components when analysis of deviance is performed. Finally computation of confidence limits for each individual smoother increases the time required, especially when loess smoothers are utilized.

For univariate spline smoothers, subject to the aforementioned caveats, problems that correspond to all shaded regions in Figure 36.4 can be completed within a few minutes. For univariate loess smoothers, the two darkest regions are feasible. For bivariate spline smoothers, problems that correspond to only the darkest shading can be completed in the order of a few minutes. The problems that correspond to the upper right unshaded region might be possible, but they require long computation times.

Forms of Additive Models

Suppose that y is a continuous variable and x1 and x2 are two explanatory variables of interest. To fit an additive model, you can use a MODEL statement similar to that used in many regression procedures in the SAS System:

```
model y = spline(x1) spline(x2);
```

This model statement requires the procedure to fit the following model:

$$\eta(x_1, x_2) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + s_1(x_1) + s_2(x_2)$$

where the $s_i(\,)$ terms denote nonparametric spline functions of the respective explanatory variables.

The GAM procedure can fit semiparametric models. The following MODEL statement assumes a linear relation with x1 and an unknown functional relation with x2:

```
model y = param(x1) spline(x2);
```

If you want to fit a model containing a functional two-way interaction between x1 and x2, you can use the following MODEL statement:

```
model y = spline2(x1,x2);
```

In this case, the GAM procedure fits a model equivalent to that of PROC TPSPLINE.

Estimates from PROC GAM

PROC GAM provides the capability to fit both nonparametric and semiparametric models. So that you can better understand the underlying trend of any given factor, PROC GAM separates the linear trend from any general nonparametric trend during the fitting as well as in the final report. This makes it easy to determine whether the significance of a smoothing variable is associated with a simple linear trend or a more complicated pattern.

For example, suppose you want to fit a semiparametric model as

$$y = \alpha_0 + \alpha_1 z + f_1(x_1) + f_2(x_2)$$

The GAM estimate for this model is

$$y = \widehat{\alpha_0} + \widehat{\alpha_1} z + \widehat{\beta_1} x_1 + \widehat{\beta_2} x_2 + \widehat{s_1}(x_1) + \widehat{s_2}(x_2)$$

where $\widehat{s_1}$ and $\widehat{s_2}$ are linear-adjusted nonparametric estimates of the f_1 and f_2 effects. The p-values for $\widehat{\alpha_0}$, $\widehat{\alpha_1}$, $\widehat{\beta_1}$, and $\widehat{\beta_2}$ are reported in the parameter estimates table. $\widehat{\beta_1}$ and $\widehat{\beta_2}$ are the estimates labeled Linear(x1) and Linear(x2) in the table. The p-values for $\widehat{s_1}$ and $\widehat{s_2}$ are reported in the analysis of deviance table.

Only $\widehat{s_1}$, $\widehat{s_2}$, and $\hat{y}$ are output to the output data set, with the corresponding variable names P_x1, P_x2, and P_y. For Gaussian data, the complete marginal prediction for variable x1 is:

$$\widehat{\beta_1} x_1 + P_{x_1}$$

If the additive component plots are requested by the ADDITIVE option, the additive component for variable x2 is obtained by:

$$\widehat{\beta_2}(x_2 - \bar{x_2}) + P_{x_2}$$

ODS Table Names

PROC GAM assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 36.3 ODS Tables Produced by PROC GAM

ODS Table Name	Description	Statement	Option
ANODEV	Analysis of deviance table for smoothing variables	PROC	default
ClassSummary	Summary of classification variables	PROC	default
ConvergenceStatus	Convergence status of the local scoring algorithm	PROC	default
InputSummary	Input data summary	PROC	default
IterHistory	Iteration history table	MODEL	ITPRINT
IterSummary	Iteration summary	PROC	default
FitSummary	Fit parameters and fit summary	PROC	default
ParameterEstimates	Parameter estimation for regression variables	PROC	default
ResponseProfile	Frequency counts for binary models	MODEL	DIST=BINOMIAL

By referring to the names of such tables, you can use the ODS OUTPUT statement to place one or more of these tables in output data sets.

ODS Graphics

This section describes the use of ODS for creating statistical graphs with the GAM procedure. To request these graphs you must specify the ODS GRAPHICS statement in addition to the options indicated in Table 36.4. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

When ODS Graphics is in effect, the GAM procedure by default produces plots of the partial predictions for each nonparametric predictor in the model. Use the PLOTS option in the PROC GAM statement to control aspects of these plots.

ODS Graph Names

PROC GAM assigns a name to each graph it creates by using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 36.4.

Table 36.4 ODS Graphics Produced by PROC GAM

ODS Graph Name	Plot Description	PLOTS= Option
SmoothingComponentPlot	panel of multiple partial prediction curves	COMPONENTS
SmoothingComponentPlot	unpacked partial prediction curves	COMPONENTS(UNPACK)

By default, partial prediction plots for each component are displayed in panels containing at most six plots. If you specify more than six smoothing components, multiple panels are used. Use the PLOTS(UNPACK) option in the PROC GAM statement to display these plots individually.

Examples: GAM Procedure

Example 36.1: Generalized Additive Model with Binary Data

This example illustrates the capabilities of the GAM procedure and compares it to the GENMOD procedure.

The data used in this example are based on a study by Bell et al. (1994). Bell and his associates studied the result of multiple-level thoracic and lumbar laminectomy, a corrective spinal surgery commonly performed on children. The data in the study consist of retrospective measurements on 83 patients. The specific outcome of interest is the presence (1) or absence (0) of kyphosis, defined as a forward flexion of the spine of at least 40 degrees from vertical. The available predictor variables are age in months at time of the operation (Age), the starting of vertebrae levels involved in the operation (StartVert), and the number of levels involved (NumVert). The goal of this analysis is to identify risk factors for kyphosis. PROC GENMOD can be used to investigate the relationship among kyphosis and the predictors. The following DATA step creates the data kyphosis:

```
title 'Comparing PROC GAM with PROC GENMOD';
data kyphosis;
    input Age StartVert NumVert Kyphosis @@;
datalines;
71  5  3  0     158 14 3 0    128 5   4 1
2   1  5  0     1   15 4 0    1   16  2 0
61  17 2  0     37  16 3 0    113 16  2 0
59  12 6  1     82  14 5 1    148 16  3 0
18  2  5  0     1   12 4 0    243 8   8 0
168 18 3  0     1   16 3 0    78  15  6 0
175 13 5  0     80  16 5 0    27  9   4 0
22  16 2  0     105 5  6 1    96  12  3 1
131 3  2  0     15  2  7 1    9   13  5 0
12  2 14  1     8   6  3 0    100 14  3 0
4   16 3  0     151 16 2 0    31  16  3 0
125 11 2  0     130 13 5 0    112 16  3 0
140 11 5  0     93  16 3 0    1   9   3 0
52  6  5  1     20  9  6 0    91  12  5 1
73  1  5  1     35  13 3 0    143 3   9 0
61  1  4  0     97  16 3 0    139 10  3 1
136 15 4  0     131 13 5 0    121 3   3 1
177 14 2  0     68  10 5 0    9   17  2 0
139 6  10 1     2   17 2 0    140 15  4 0
72  15 5  0     2   13 3 0    120 8   5 1
51  9  7  0     102 13 3 0    130 1   4 1
114 8  7  1     81  1  4 0    118 16  3 0
118 16 4  0     17  10 4 0    195 17  2 0
159 13 4  0     18  11 4 0    15  16  5 0
158 15 4  0     127 12 4 0    87  16  4 0
206 10 4  0     11  15 3 0    178 15  4 0
157 13 3  1     26  13 7 0    120 13  2 0
42  6  7  1     36  13 4 0
;

proc genmod data=kyphosis descending;
    model Kyphosis = Age StartVert NumVert/link=logit dist=binomial;
run;
```

The GENMOD analysis of the independent variable effects is shown in Output 36.1.1. Based on these results, the only significant factor is StartVert with a log odds ratio of -0.1972. The variable NumVert has a p-value of 0.0904 with a log odds ratio of 0.3031.

Output 36.1.1 GENMOD Analysis: Partial Output

Comparing PROC GAM with PROC GENMOD

The GENMOD Procedure

Analysis Of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	-1.2497	1.2424	-3.6848	1.1853	1.01	0.3145
Age	1	0.0061	0.0055	-0.0048	0.0170	1.21	0.2713
StartVert	1	-0.1972	0.0657	-0.3260	-0.0684	9.01	0.0027
NumVert	1	0.3031	0.1790	-0.0477	0.6540	2.87	0.0904
Scale	0	1.0000	0.0000	1.0000	1.0000		

NOTE: The scale parameter was held fixed.

The GENMOD procedure assumes a strict linear relationship between the response and the predictors. The following SAS statements use PROC GAM to investigate a less restrictive model, with moderately flexible spline terms for each of the predictors:

```
title 'Comparing PROC GAM with PROC GENMOD';
proc gam data=kyphosis;
   model Kyphosis (event='1') = spline(Age      ,df=3)
                                spline(StartVert,df=3)
                                spline(NumVert  ,df=3) / dist=binomial;
run;
```

The MODEL statement requests an additive model with a univariate smoothing spline for each term. The response variable option EVENT= chooses Kyphosis= 1 (presence) as the event so that the probability of presence of kyphosis is modeled. The option "DIST=BINOMIAL" with binary responses specifies a logistic model. Each term is fit by using a univariate smoothing spline with three degrees of freedom. Of these three degrees of freedom, one is taken up by the linear portion of the fit and two are left for the nonlinear spline portion. Although this might seem to be an unduly modest amount of flexibility, it is better to be conservative with a data set this small.

Output 36.1.2 and Output 36.1.3 list the output from PROC GAM.

Output 36.1.2 Summary Statistics

```
                    Comparing PROC GAM with PROC GENMOD

                              The GAM Procedure
                          Dependent Variable: Kyphosis
  Smoothing Model Component(s): spline(Age) spline(StartVert) spline(NumVert)

                            Summary of Input Data Set

                  Number of Observations                    83
                  Number of Missing Observations             0
                  Distribution                        Binomial
                  Link Function                          Logit

                               Response Profile

                     Ordered                          Total
                       Value        Kyphosis       Frequency

                           1               0              65
                           2               1              18

NOTE: PROC GAM is modeling the probability that Kyphosis=1. One way to change
      this to model the probability that Kyphosis=0 is to specify the response
      variable option EVENT='0'.


                      Iteration Summary and Fit Statistics

          Number of local scoring iterations                          9
          Local scoring convergence criterion              2.6635648E-9
          Final Number of Backfitting Iterations                      1
          Final Backfitting Criterion                      5.2326574E-9
          The Deviance of the Final Estimate               46.610922438
```

Output 36.1.3 Model Fit Statistics

```
                          Regression Model Analysis
                             Parameter Estimates

                           Parameter       Standard
     Parameter              Estimate          Error    t Value    Pr > |t|

     Intercept             -2.01533        1.45620      -1.38      0.1706
     Linear(Age)            0.01213        0.00794       1.53      0.1308
     Linear(StartVert)     -0.18615        0.07628      -2.44      0.0171
     Linear(NumVert)        0.38347        0.19102       2.01      0.0484
```

Output 36.1.3 *continued*

Smoothing Model Analysis
Fit Summary for Smoothing Components

Component	Smoothing Parameter	DF	GCV	Num Unique Obs
Spline(Age)	0.999996	2.000000	328.512831	66
Spline(StartVert)	0.999551	2.000000	317.646685	16
Spline(NumVert)	0.921758	2.000000	20.144056	10

Smoothing Model Analysis
Analysis of Deviance

Source	DF	Sum of Squares	Chi-Square	Pr > ChiSq
Spline(Age)	2.00000	10.494369	10.4944	0.0053
Spline(StartVert)	2.00000	5.494968	5.4950	0.0641
Spline(NumVert)	2.00000	2.184518	2.1845	0.3355

The critical part of the GAM results is the "Analysis of Deviance" table, shown in Output 36.1.3. For each smoothing effect in the model, this table gives a χ^2 test comparing the deviance between the full model and the model without this variable. In this case the analysis of deviance results indicates that the effect of Age is highly significant, the effect of StartVert is nearly significant, and the effect of NumVert is insignificant at the 5% level.

PROC GAM can also perform approximate analysis of deviance for smoothing effects by using the ANODEV=NOREFIT option, as in the following statements:

```
title 'PROC GAM with Approximate Analysis of Deviance';
proc gam data=kyphosis;
   model Kyphosis (event='1') = spline(Age      ,df=3)
                                spline(StartVert,df=3)
                                spline(NumVert  ,df=3) /
                                    dist=binomial anodev=norefit;
run;
```

Output 36.1.4 Approximate Analysis of Deviance Table

PROC GAM with Approximate Analysis of Deviance

The GAM Procedure
Dependent Variable: Kyphosis
Smoothing Model Component(s): spline(Age) spline(StartVert) spline(NumVert)

Smoothing Model Analysis
Approximate Analysis of Deviance

Source	DF	Chi-Square	Pr > ChiSq
Spline(Age)	2.00000	7.0888	0.0289
Spline(StartVert)	2.00000	5.0431	0.0803
Spline(NumVert)	2.00000	2.2471	0.3251

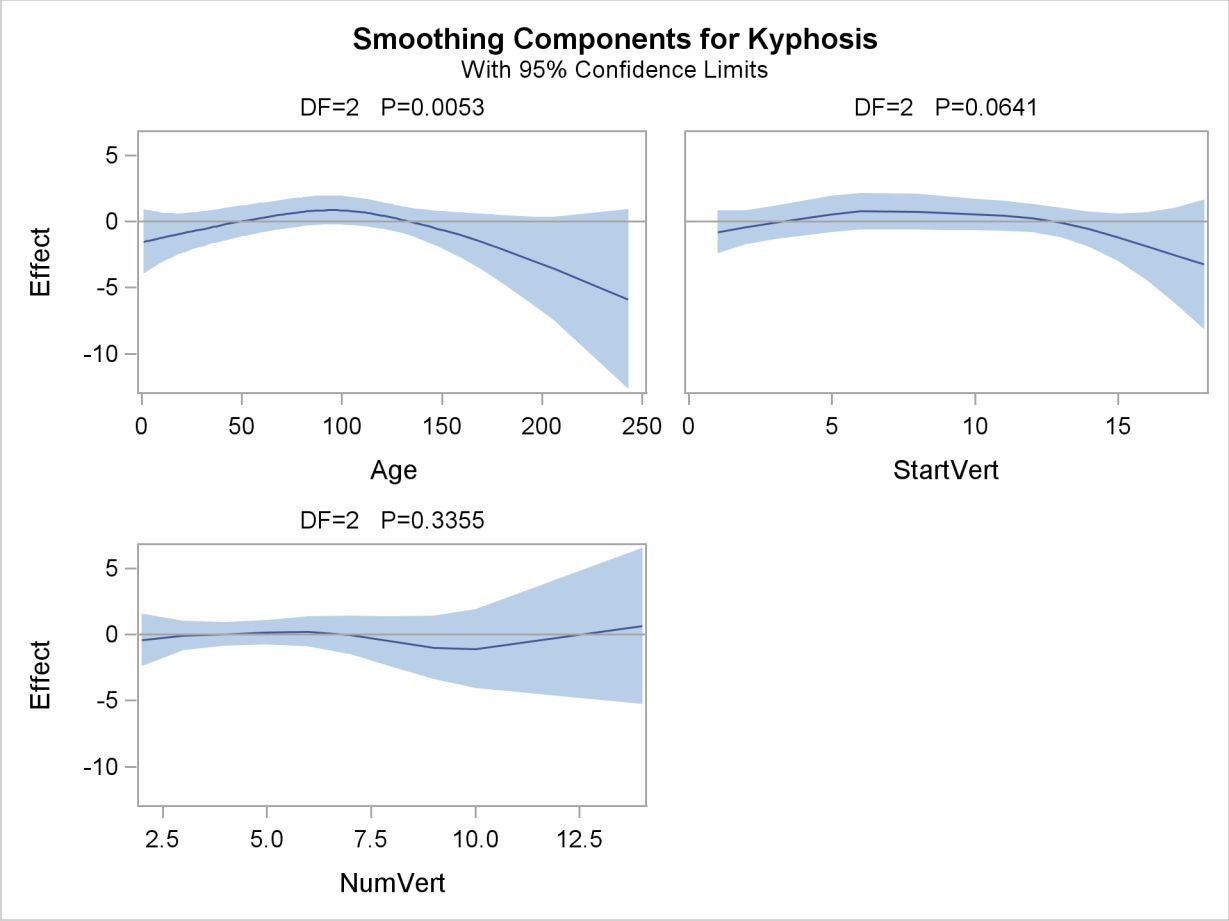
The "Approximate Analysis of Deviance" table shown in Output 36.1.4 yields similar conclusions to those of the "Analysis of Deviance" table (Output 36.1.3). In addition to the GAM fit that uses all the specified smoothing effects, the default ANODEV=REFIT option requires p additional GAM fits to be performed for p smoothing effects. In each of these fits, a submodel is fit by omitting one smoothing term from the model. By contrast, the ANODEV=NOREFIT option keeps the nonparametric terms fixed and requires a weighted least squares fit for only the parametric part of the model. Hence, GAM with the ANODEV=NOREFIT option is computationally inexpensive and is useful for obtaining approximate analysis of deviance results for models with many smoothing effects.

Plots of predictions against predictor can be used to investigate why PROC GAM and PROC GENMOD produce different results. The following statements use ODS Graphics to produce plots of the individual smoothing components. The CLM suboption for the PLOTS option adds a curvewise Bayesian confidence band to each smoothing component, while the COMMONAXES suboption forces all three smoothing component plots to share the same vertical axis limits, allowing a visual judgment of nonparametric effect size.

```
ods graphics on;

proc gam data=kyphosis plots=components(clm commonaxes);
    model Kyphosis (event='1') = spline(Age      ,df=3)
                                 spline(StartVert,df=3)
                                 spline(NumVert  ,df=3) / dist=binomial;
run;
```

For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the GAM procedure, see the section "ODS Graphics" on page 1861. The smoothing component plots are displayed in Output 36.1.5.

Output 36.1.5 Partial Prediction for Each Predictor



The plots show that the partial predictions corresponding to both Age and StartVert have a quadratic pattern, while NumVert has a more complicated but weaker pattern. However, in the plot for NumVert, notice that about half the vertical range of the function is determined by the point at the upper extreme. It would be a good idea, therefore, to rerun the analysis without this point, to explore how much it affects the conclusions. You can do this by simply including a WHERE clause when specifying the data set for the GAM procedure, as in the following statements:

```
proc gam data=kyphosis(where=(NumVert^=14)) plots=components(clm commonaxes);
    model Kyphosis (event='1') = spline(Age      ,df=3)
                                 spline(StartVert,df=3)
                                 spline(NumVert  ,df=3) / dist=binomial;
run;

ods graphics off;
```

The analysis of deviance table from this reanalysis is shown in Output 36.1.6, and Output 36.1.7 shows the recomputed partial predictor plots.

Output 36.1.6 Analysis after Removing NumVert=14

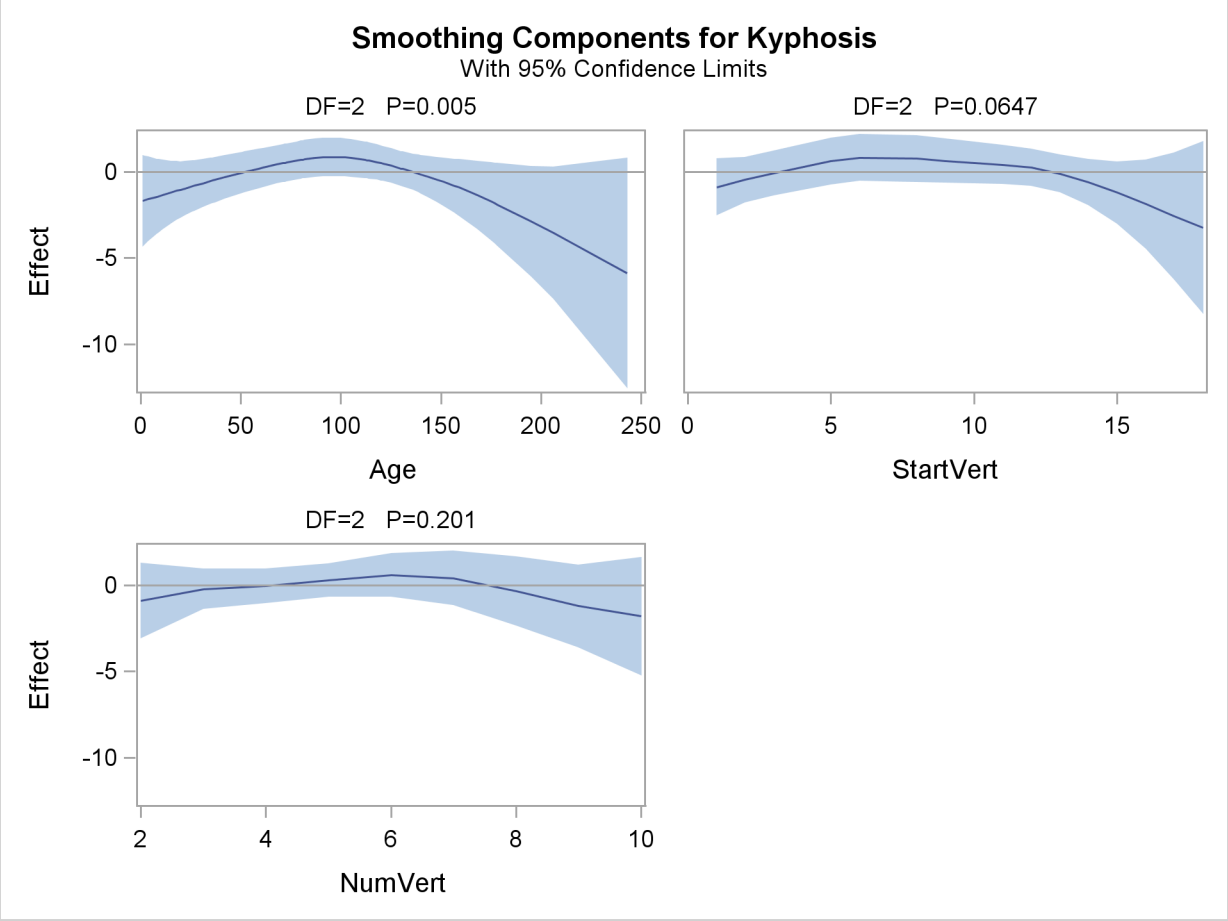
PROC GAM with Approximate Analysis of Deviance

The GAM Procedure
Dependent Variable: Kyphosis
Smoothing Model Component(s): spline(Age) spline(StartVert) spline(NumVert)

Smoothing Model Analysis
Analysis of Deviance

Source	DF	Sum of Squares	Chi-Square	Pr > ChiSq
Spline(Age)	2.00000	10.587556	10.5876	0.0050
Spline(StartVert)	2.00000	5.477094	5.4771	0.0647
Spline(NumVert)	2.00000	3.209089	3.2091	0.2010

Output 36.1.7 Analysis after Removing NumVert=14



Removing data point NumVert=14 has little effect on either the analysis of deviance results or the estimated curves for StartVert and NumVert. But the removal has a noticeable effect on the variable NumVert, whose curve now also seems quadratic, though it is much less pronounced than for the other two variables.

An important difference between the first analysis of these data with GENMOD and the subsequent analysis with GAM is that GAM indicates that age is a significant predictor of kyphosis. The difference is due to the fact that the GENMOD model includes only a linear effect in Age whereas the GAM model allows a more complex relationship, which the plots indicate is nearly quadratic. Having used the GAM procedure to discover an appropriate form of the dependence of Kyphosis on each of the three independent variables, you can use the GENMOD procedure to fit and assess the corresponding parametric model. The following statements fit a GENMOD model with quadratic terms for all three variables, including tests for the joint linear and quadratic effects of each variable. The resulting contrast tests are shown in Output 36.1.8.

```
title 'Comparing PROC GAM with PROC GENMOD';
proc genmod data=kyphosis(where=(NumVert^=14)) descending;
            model kyphosis = Age       Age      *Age
                             StartVert StartVert*StartVert
                             NumVert   NumVert  *NumVert /
                                 link=logit  dist=binomial;
            contrast 'Age'       Age       1, Age*Age             1;
            contrast 'StartVert' StartVert 1, StartVert*StartVert 1;
            contrast 'NumVert'   NumVert   1, NumVert*NumVert     1;
run;
```

Output 36.1.8 Joint Linear and Quadratic Tests

Comparing PROC GAM with PROC GENMOD

The GENMOD Procedure

Contrast Results

Contrast	DF	Chi-Square	Pr > ChiSq	Type
Age	2	13.63	0.0011	LR
StartVert	2	15.41	0.0005	LR
NumVert	2	3.56	0.1684	LR

The results for the quadratic GENMOD model are now quite consistent with the GAM results.

From this example, you can see that PROC GAM is very useful in visualizing the data and detecting the nonlinearity among the variables.

Example 36.2: Poisson Regression Analysis of Component Reliability

In this example, the number of maintenance repairs on a complex system are modeled as realizations of Poisson random variables. The system under investigation has a large number of components, which occasionally break down and are replaced or repaired. During a four-year period, the system was observed to be in a state of steady operation, meaning that the rate of operation remained approximately constant. A monthly maintenance record is available for that period, which tracks the number of components removed for maintenance each month. The data are listed in the following statements, which create a SAS data set:

```
title 'Analysis of Component Reliability';
data equip;
    input year month removals @@;
datalines;
1987   1  2 1987   2  4 1987   3  3
1987   4  3 1987   5  3 1987   6  8
1987   7  2 1987   8  6 1987   9  3
1987  10  9 1987  11  4 1987  12 10
1988   1  4 1988   2  6 1988   3  4
1988   4  4 1988   5  3 1988   6  5
1988   7  3 1988   8  4 1988   9  5
1988  10  3 1988  11  6 1988  12  3
1989   1  2 1989   2  6 1989   3  1
1989   4  5 1989   5  5 1989   6  4
1989   7  2 1989   8  2 1989   9  2
1989  10  5 1989  11  1 1989  12 10
1990   1  3 1990   2  8 1990   3 12
1990   4  7 1990   5  3 1990   6  2
1990   7  4 1990   8  3 1990   9  0
1990  10  6 1990  11  6 1990  12  6
;
```

For planning purposes, it is of interest to understand the long- and short-term trends in the maintenance needs of the system. Over the long term, it is suspected that the quality of new components and repair work improves over time, so the number of component removals would tend to decrease from year to year. It is not known whether the robustness of the system is affected by seasonal variations in the operating environment, but this possibility is also of interest.

Because the maintenance record is in the form of counts, the number of removals are modeled as realizations of Poisson random variables. Denote by λ_{ij} the unobserved component removal rate for year i and month j. Since the data were recorded at regular intervals (from a system operating at a constant rate), each λ_{ij} is assumed to be a function of year and month only.

A preliminary two-way analysis is performed by using PROC GENMOD to make broad inferences on repair trends. A log-link is specified for the model

$$\log \lambda_{ij} \quad = \quad \mu + \alpha_i^{Y} + \alpha_j^{M}$$

where μ is a grand mean, α_i^{Y} is the effect of the ith year, and α_j^{M} is the effect of the jth month. In the following statements, the CLASS statement declares the variables year and month as categorical. Type III sum of squares are requested to test whether there is an overall effect of year and/or month.

```
title2 'Two-way model';
proc genmod data=equip;
    class year month;
    model removals=year month / dist=Poisson link=log type3;
run;
```

Output 36.2.1 displays the listed Type III statistics for the fitted model. With the test for year effects yielding a p-value of 0.4527, there is no evidence of a long-term trend in maintenance rates. Apparently, the quality of new or repaired components did not change between 1987 and 1990. However, the test for monthly trends does yield a small p-value of 0.0321, indicating that seasonal trends are just barely significant at the $\alpha = 0.05$ level.

Output 36.2.1 PROC GENMOD Listing for Type III Analysis

```
                    Analysis of Component Reliability
                              Two-way model

                           The GENMOD Procedure

                    LR Statistics For Type 3 Analysis

                                        Chi-
              Source           DF     Square    Pr > ChiSq

              year              3       2.63        0.4527
              month            11      21.12        0.0321
```

The Type III tests indicate that the year term can be dropped from the model. The focus of the analysis is now on identifying the form of the underlying seasonal trend, which is a task that PROC GAM is especially suited for. PROC GAM will be used to fit both a reduced categorical model, with year eliminated, and a nonparametric spline model. Although PROC GENMOD also has the capability to fit categorical models, as demonstrated earlier, PROC GAM will be used here to fit both models for a better comparison.

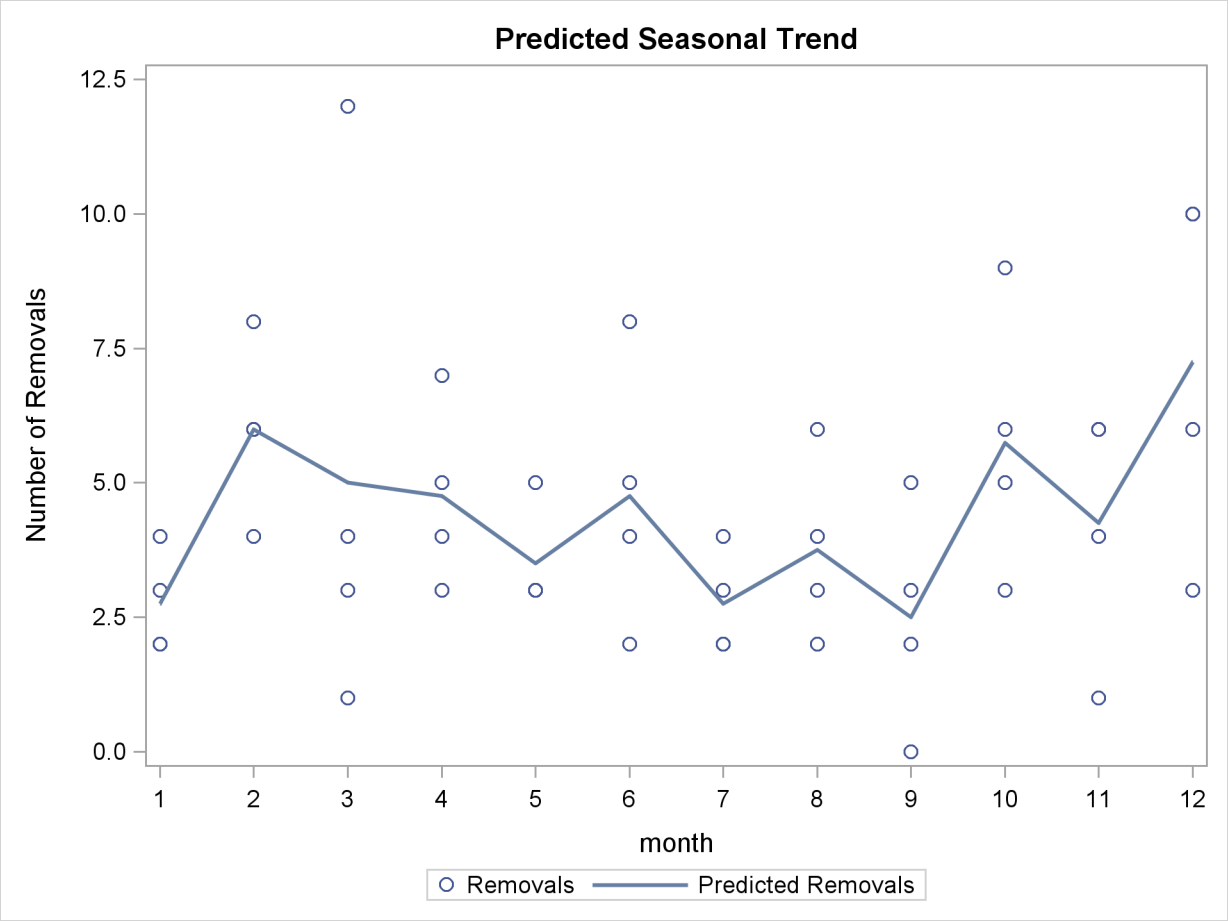
The following PROC GAM statements specify the reduced categorical model. For this part of the analysis, a CLASS statement is again used to specify that month is a categorical variable. In the follow-up, the seasonal effect will be treated as a nonparametric function of month.

```
title2 'One-way model';
proc gam data=equip;
     class month;
     model removals=param(month) / dist=Poisson;
     output out=est p;
run;
```

The following statements use the SGPLOT procedure to generate a plot of the estimated seasonal trend. The plot is displayed in Output 36.2.2.

```
proc sort data=est;by month;run;
proc sgplot data=est;
    title "Predicted Seasonal Trend";
    yaxis label="Number of Removals";
    xaxis integer values=(1 to 12);
    scatter x=Month y=Removals / name="points"
                                 legendLabel="Removals";
    series  x=Month y=p_Removals / name="line"
                                   legendLabel="Predicted Removals"
                                   lineattrs = GRAPHFIT;
    discretelegend "points" "line";
run;
```

Output 36.2.2 Predicted Seasonal Trend from a Parametric Model Fit Using a CLASS Statement



The predicted repair rates shown in Output 36.2.2 form a jagged seasonal pattern. Ignoring the month-to-month fluctuations, which are difficult to explain and can be artifacts of random noise, the general removal rate trend starts by increasing at the beginning of the year; then the trend flattens out in February and then decreases through August; and then it flattens out again in September and begins an increasing trend that continues throughout the rest of the year.

One advantage of nonparametric regression is its ability to highlight general trends in the data, such as those described earlier, and to attribute local fluctuations to unexplained random noise. The nonparametric regression model used by PROC GAM specifies that the underlying removal rates λ_j are of the form

$$\log \lambda_j = \beta_0 + \beta_1 Month_j + s(Month_j)$$

where β_1 is a linear coefficient and $s(\,)$ is a nonparametric regression function. β_1 and $s(\,)$ define the linear and nonparametric parts, respectively, of the seasonal trend.

The following statements request that PROC GAM fit a cubic spline model to the monthly repair data. The output listing is displayed in Output 36.2.3 and Output 36.2.4.

```
title 'Analysis of Component Reliability';
title2 'Spline model';
proc gam data=equip;
    model removals=spline(month) / dist=Poisson method=gcv;
run;
```

The METHOD=GCV option is used to determine an appropriate level of smoothing. The keywords LCLM and UCLM in the OUTPUT statement request that lower and upper 95% confidence bounds on each $s(Month_j)$ be included in the output data set.

Output 36.2.3 PROC GAM Listing for Cubic Spline Regression Using the METHOD=GCV Option

```
                    Analysis of Component Reliability
                                Spline model

                            The GAM Procedure
                       Dependent Variable: removals
                 Smoothing Model Component(s): spline(month)

                         Summary of Input Data Set

                 Number of Observations                 48
                 Number of Missing Observations          0
                 Distribution                      Poisson
                 Link Function                         Log

                   Iteration Summary and Fit Statistics

          Number of local scoring iterations                        5
          Local scoring convergence criterion            7.241527E-12
          Final Number of Backfitting Iterations                    1
          Final Backfitting Criterion                    1.710339E-11
          The Deviance of the Final Estimate             56.901543546
```

Output 36.2.4 Model Fit Statistics

Regression Model Analysis
Parameter Estimates

Parameter	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1.34594	0.14509	9.28	<.0001
Linear(month)	0.02274	0.01893	1.20	0.2362

Smoothing Model Analysis
Fit Summary for Smoothing Components

Component	Smoothing Parameter	DF	GCV	Num Unique Obs
Spline(month)	0.901512	1.879980	0.115848	12

Smoothing Model Analysis
Analysis of Deviance

Source	DF	Sum of Squares	Chi-Square	Pr > ChiSq
Spline(month)	1.87998	8.877764	8.8778	0.0103

Notice in the listing in Output 36.2.4 that the DF value chosen for the nonlinear portion of the spline by minimizing GCV is about 1.88, which is smaller than the default value of 3. This indicates that the spline model of the seasonal trend is relatively simple. As indicated by the "Analysis of Deviance" table, it is a significant feature of the data: the table lists a p-value of 0.0103 for the hypothesis of no seasonal trend. Note also that the "Parameter Estimates" table lists a p-value of 0.2362 for the hypothesis of no linear factor in the seasonal trend.

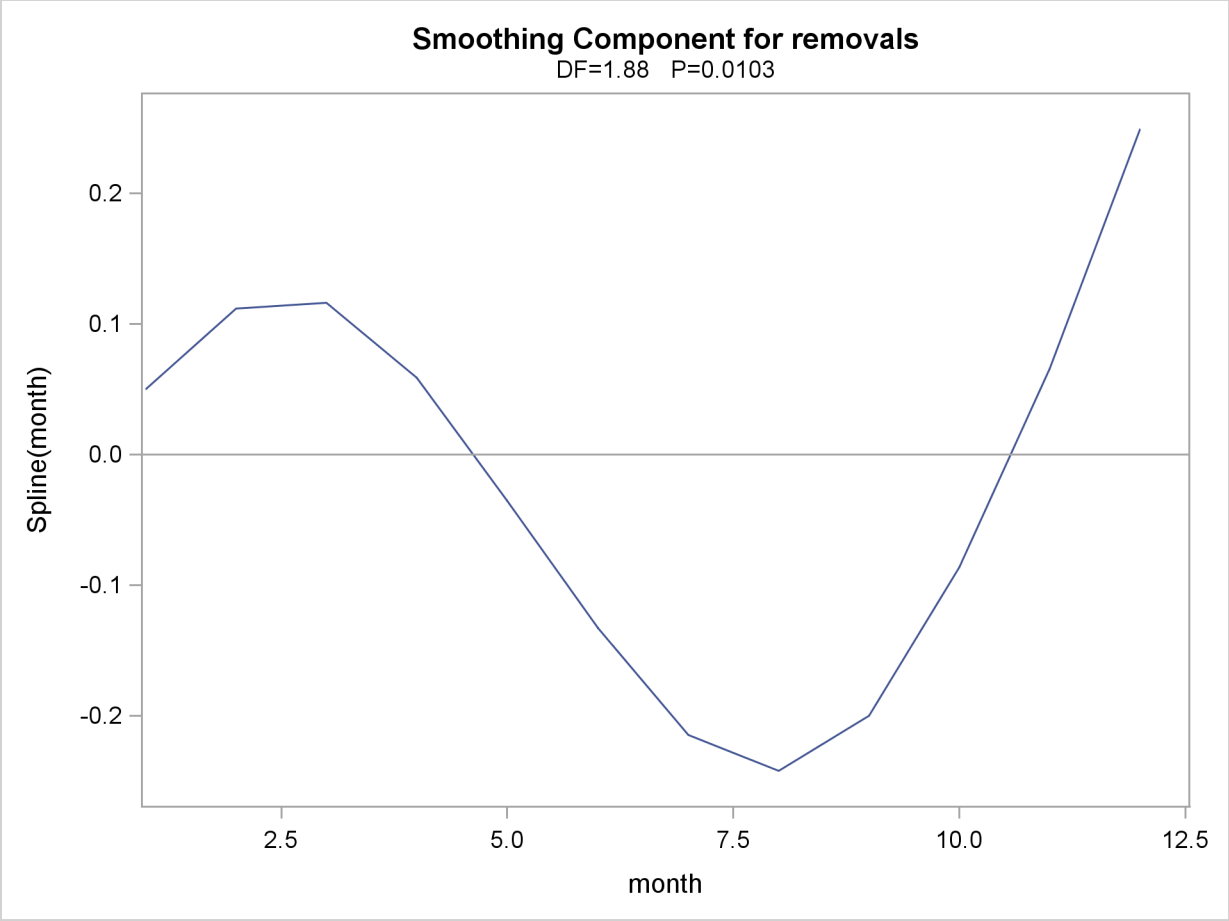
The following statements use ODS Graphics statement to plot the smoothing component for the effect of Month on predicted repair rates:

```
ods graphics on;

proc gam data=equip;
    model removals=spline(month) / dist=Poisson method=gcv;
run;
```

For general information about ODS graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the GAM procedure, see the section "ODS Graphics" on page 1861. The smoothing component plot is displayed in Output 36.2.5.

In Output 36.2.5, it is apparent that the pattern of repair rates follows the general pattern observed in Output 36.2.2. However, the plot in Output 36.2.5 is much cleaner because the month-to-month fluctuations are smoothed out to reveal the broader seasonal trend.

Output 36.2.5 Predicted Seasonal Trend from a Cubic Spline Model

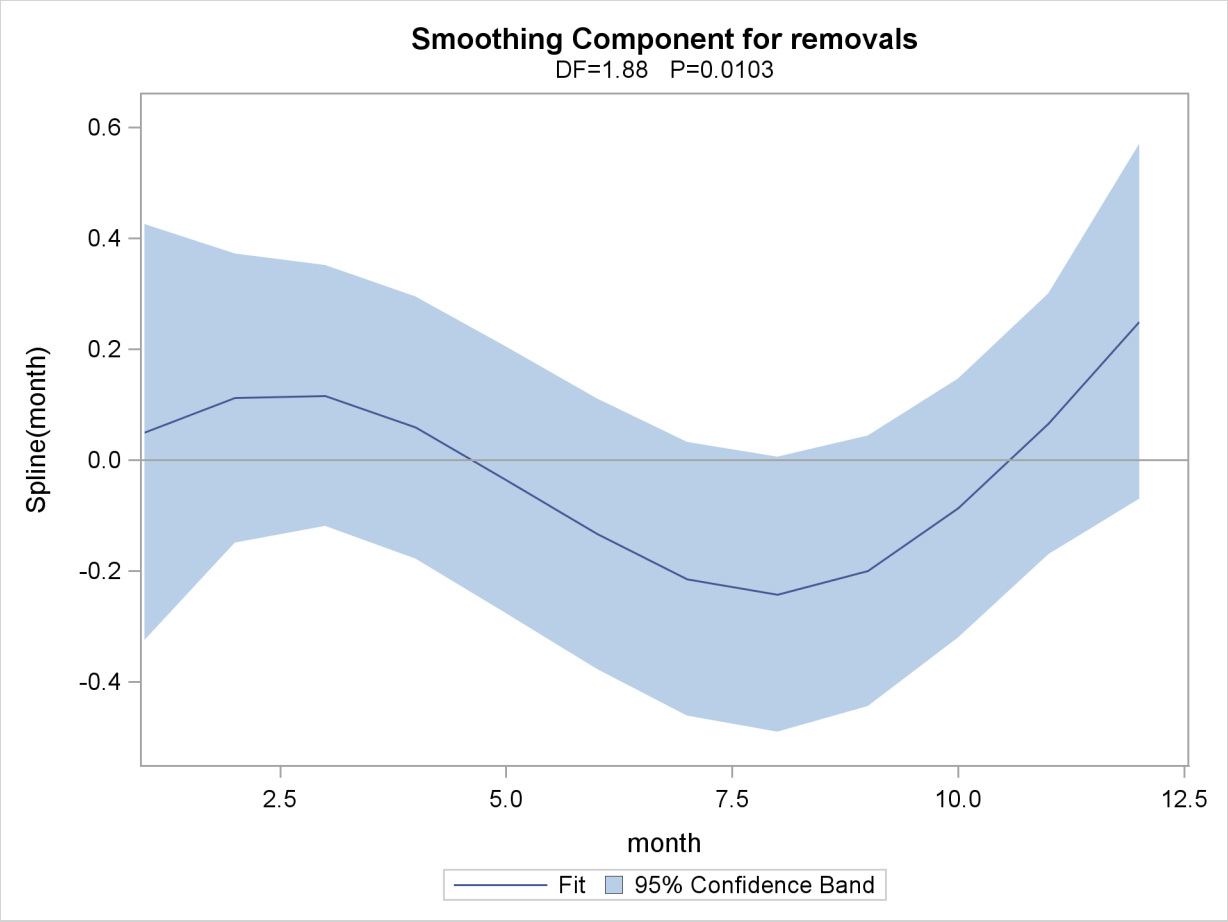


You can use the PLOTS(CLM) option to added a 95% confidence band to the plot of $s(Month_j)$, as in the following statements:

```
proc gam data=equip plots=components(clm);
    model removals=spline(month) / dist=Poisson method=gcv;
run;

ods graphics off;
```

The plot is displayed in Output 36.2.6.

Output 36.2.6 Estimated Nonparametric Factor of Seasonal Trend, Along with 95% Confidence Bounds



The small p-value in Output 36.2.1 of 0.0321 for the hypothesis of no seasonal trend indicates that the data exhibit significant seasonal structure. However, Output 36.2.6 is a graphical illustration of a degree of indistinctness in that structure. For instance, the horizontal reference line at zero is entirely within the 95% confidence band—that is, the estimated nonlinear part of the trend is relatively flat. Thus, despite evidence of seasonality based on the parametric model, it is difficult to narrow down its significant effects to a specific part of the year.

Example 36.3: Comparing PROC GAM with PROC LOESS

In an analysis of simulated data from a hypothetical chemistry experiment, additive nonparametric regression performed by PROC GAM is compared to the unrestricted multidimensional procedure of PROC LOESS.

In each repetition of the experiment, a catalyst is added to a chemical solution, thereby inducing synthesis of a new material. The data are measurements of the temperature of the solution, the

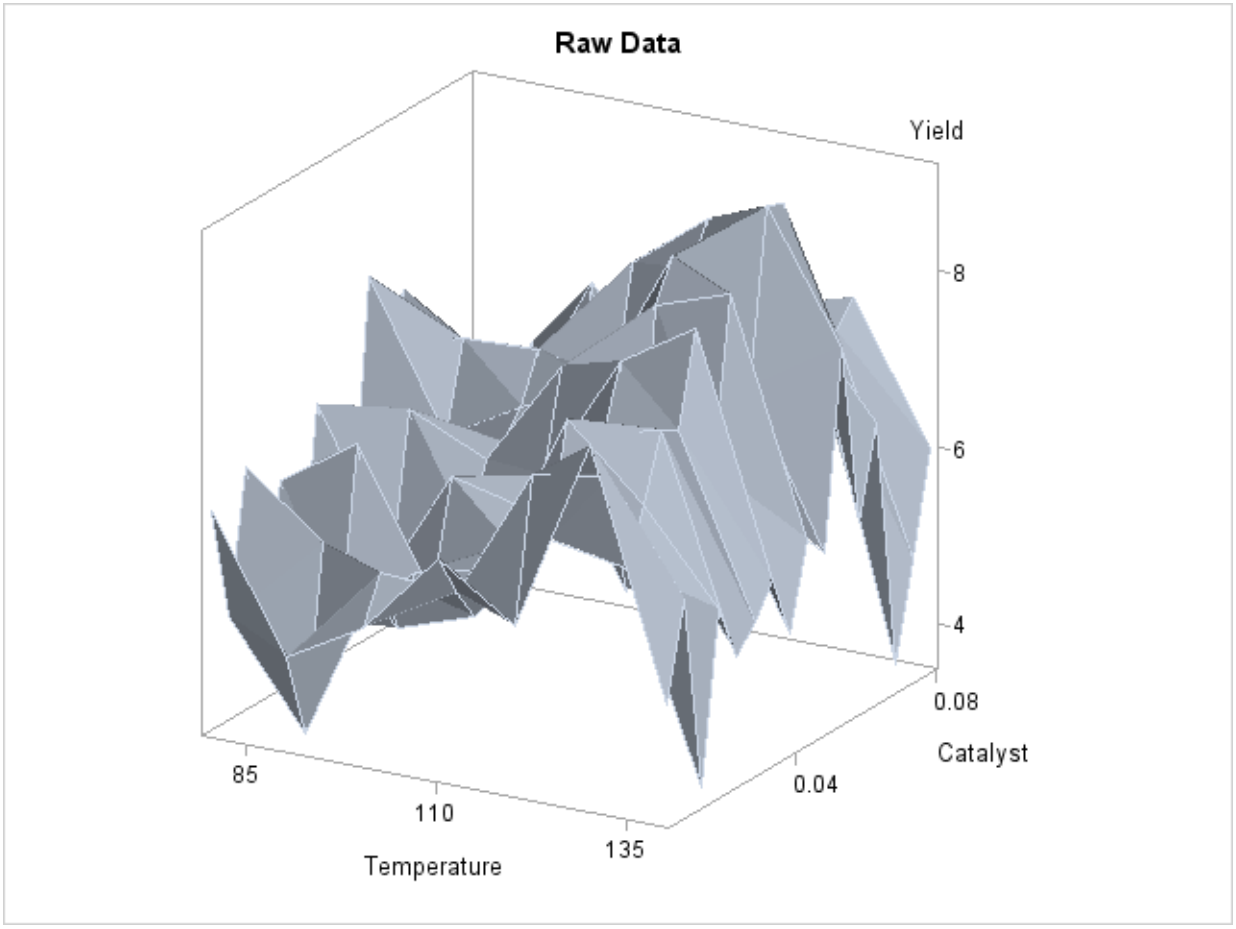
amount of catalyst added, and the yield of the chemical reaction. The following statements read and plots the raw data.

```
data ExperimentA;
    format Temperature f4.0 Catalyst f6.3 Yield f8.3;
    input Temperature Catalyst Yield @@;
datalines;
80  0.005 6.039  80 0.010 4.719  80 0.015 6.301
80  0.020 4.558  80 0.025 5.917  80 0.030 4.365
80  0.035 6.540  80 0.040 5.063  80 0.045 4.668
80  0.050 7.641  80 0.055 6.736  80 0.060 7.255
80  0.065 5.515  80 0.070 5.260  80 0.075 4.813
80  0.080 4.465  90 0.005 4.540  90 0.010 3.553
90  0.015 5.611  90 0.020 4.586  90 0.025 6.503
90  0.030 4.671  90 0.035 4.919  90 0.040 6.536
90  0.045 4.799  90 0.050 6.002  90 0.055 6.988
90  0.060 6.206  90 0.065 5.193  90 0.070 5.783
90  0.075 6.482  90 0.080 5.222 100 0.005 5.042
100 0.010 5.551 100 0.015 4.804 100 0.020 5.313
100 0.025 4.957 100 0.030 6.177 100 0.035 5.433
100 0.040 6.139 100 0.045 6.217 100 0.050 6.498
100 0.055 7.037 100 0.060 5.589 100 0.065 5.593
100 0.070 7.438 100 0.075 4.794 100 0.080 3.692
110 0.005 6.005 110 0.010 5.493 110 0.015 5.107
110 0.020 5.511 110 0.025 5.692 110 0.030 5.969
110 0.035 6.244 110 0.040 7.364 110 0.045 6.412
110 0.050 6.928 110 0.055 6.814 110 0.060 8.071
110 0.065 6.038 110 0.070 6.295 110 0.075 4.308
110 0.080 7.020 120 0.005 5.409 120 0.010 7.009
120 0.015 6.160 120 0.020 7.408 120 0.025 7.123
120 0.030 7.009 120 0.035 7.708 120 0.040 5.278
120 0.045 8.111 120 0.050 8.547 120 0.055 8.279
120 0.060 8.736 120 0.065 6.988 120 0.070 6.283
120 0.075 7.367 120 0.080 6.579 130 0.005 7.629
130 0.010 7.171 130 0.015 5.997 130 0.020 6.587
130 0.025 7.335 130 0.030 7.209 130 0.035 8.259
130 0.040 6.530 130 0.045 8.400 130 0.050 7.218
130 0.055 9.167 130 0.060 9.082 130 0.065 7.680
130 0.070 7.139 130 0.075 7.275 130 0.080 7.544
140 0.005 4.860 140 0.010 5.932 140 0.015 3.685
140 0.020 5.581 140 0.025 4.935 140 0.030 5.197
140 0.035 5.559 140 0.040 4.836 140 0.045 5.795
140 0.050 5.524 140 0.055 7.736 140 0.060 5.628
140 0.065 6.644 140 0.070 3.785 140 0.075 4.853
140 0.080 6.006
;
```

```
proc sort data=ExperimentA;
    by Temperature Catalyst;
run;

proc template;
    define statgraph surface;
    dynamic _X _Y _Z _T;
    begingraph;
        entrytitle _T;
        layout overlay3d/
             xaxisopts=(linearopts=(tickvaluesequence=
                        (start=85 end=135 increment=25)))
             yaxisopts=(linearopts=(tickvaluesequence=
                        (start=0 end=0.08 increment=0.04)))
             rotate=30 cube=false;
        surfaceplotparm x=_X y=_Y z=_Z;
        endlayout;
    endgraph;
    end;
run;

ods graphics on;

proc sgrender data=ExperimentA template=surface;
    dynamic _X='Temperature' _Y='Catalyst' _Z='Yield' _T='Raw Data';
run;
```

The plot is displayed in Output 36.3.1. A surface fitted to the plot of Output 36.3.1 by PROC LOESS will be of a very general (and flexible) type, since the procedure requires only weak assumptions about the structure of the dependencies among the data. PROC GAM, on the other hand, makes stronger structural assumptions by restricting the fitted surface to an additive form. These differences will be demonstrated in this example.

Output 36.3.1 Surface Plot of Yield by Temperature and Amount of Catalyst



The following statements request that both PROC LOESS and PROC GAM fit surfaces to the data:

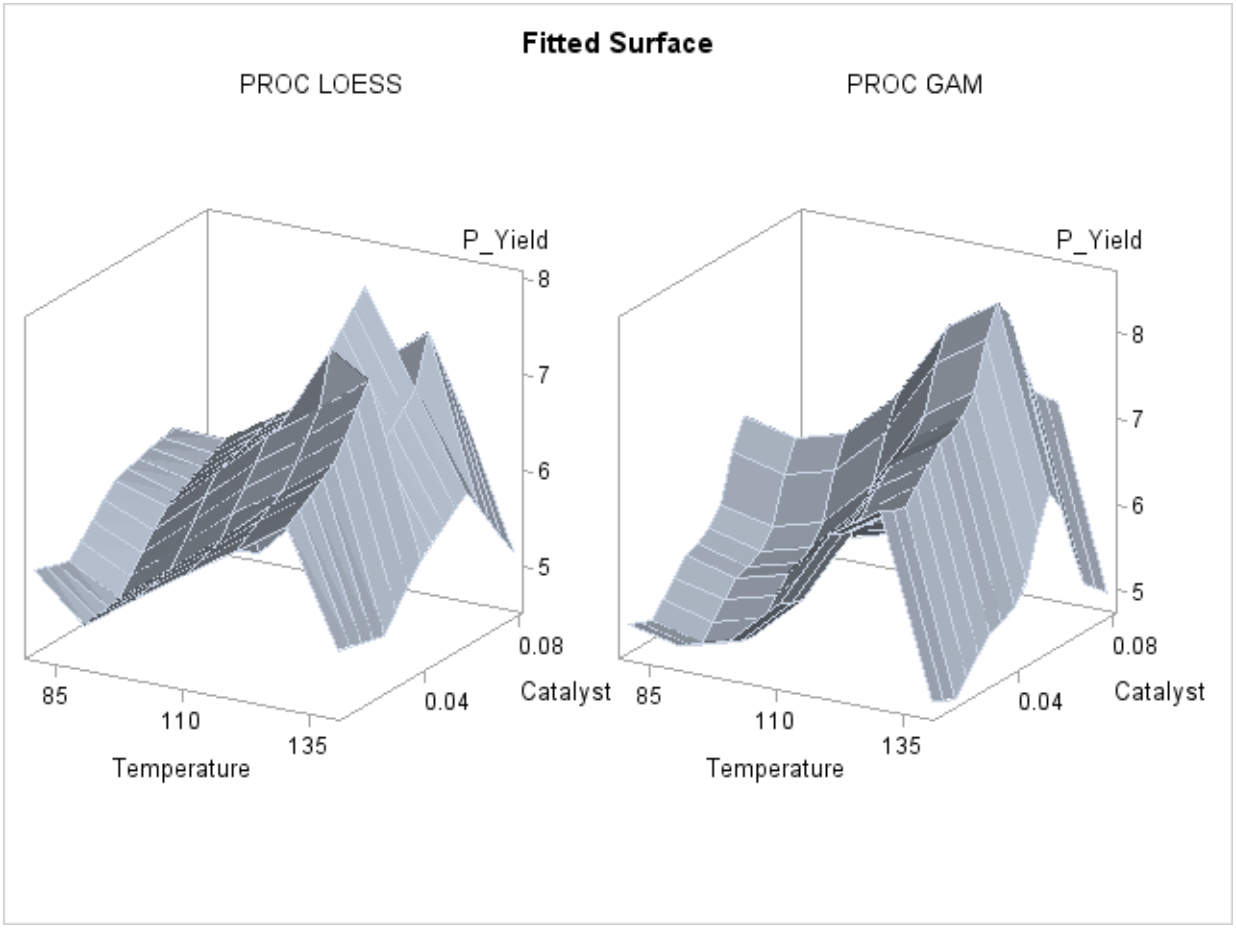
```
ods output ScoreResults=PredLOESS;
proc loess data=ExperimentA;
    model Yield = Temperature Catalyst
                  / scale=sd select=gcv degree=2;
    score;
run;

proc gam data=PredLoess;
    model Yield = loess(Temperature) loess(Catalyst) / method=gcv;
    output out=PredGAM p=Gam_p_;
run;
```

In both cases the smoothing parameter was chosen as the value that minimizes GCV. This is performed automatically by PROC LOESS and PROC GAM.

The following statements generate plots of the predicted yields, which are displayed in Output 36.3.2:

```
proc template;
    define statgraph surface1;
    begingraph;
        entrytitle "Fitted Surface";
        layout lattice/columns=2;
            layout
            overlay3d/xaxisopts=(linearopts=(tickvaluesequence=
                                 (start=85 end=135 increment=25)))
                      yaxisopts=(linearopts=(tickvaluesequence=
                                 (start=0 end=0.08 increment=0.04)))
                      rotate=30 cube=0
                      zaxisopts=(label="P_Yield");
                entry "PROC LOESS"/location=outside valign=top
                                   textattrs=graphlabeltext;
                surfaceplotparm x=Temperature y=Catalyst z=p_Yield;
            endlayout;
            layout
            overlay3d/xaxisopts=(linearopts=(tickvaluesequence=
                                 (start=85 end=135 increment=25)))
                      yaxisopts=(linearopts=(tickvaluesequence=
                                 (start=0 end=0.08 increment=0.04)))
                      rotate=30 cube=0
                      zaxisopts=(label="P_Yield");
                entry "PROC GAM"/location=outside valign=top
                                 textattrs=graphlabeltext;
                surfaceplotparm x=Temperature y=Catalyst z=Gam_p_Yield;
            endlayout;
        endlayout;
    endgraph;
    end;
run;

proc sgrender data=PredGAM template=surface1;
run;
```

Output 36.3.2 Fitted Regression Surfaces



Though both PROC LOESS and PROC GAM use the statistical technique loess, it is apparent from Output 36.3.2 that the manner in which it is applied is very different. By smoothing out the data in local neighborhoods, PROC LOESS essentially fits a surface to the data in pieces, one neighborhood at a time. The local regions are treated independently, so separate areas of the fitted surface are only weakly related. PROC GAM imposes additive structure, requiring that cross sections of the fitted surface always have the same shape and thereby relating regions that have a common value of the same individual regressor variable. Under that restriction, the loess technique need not be applied to the entire multidimensional scatter plot, but only to one-dimensional cross sections of the data.

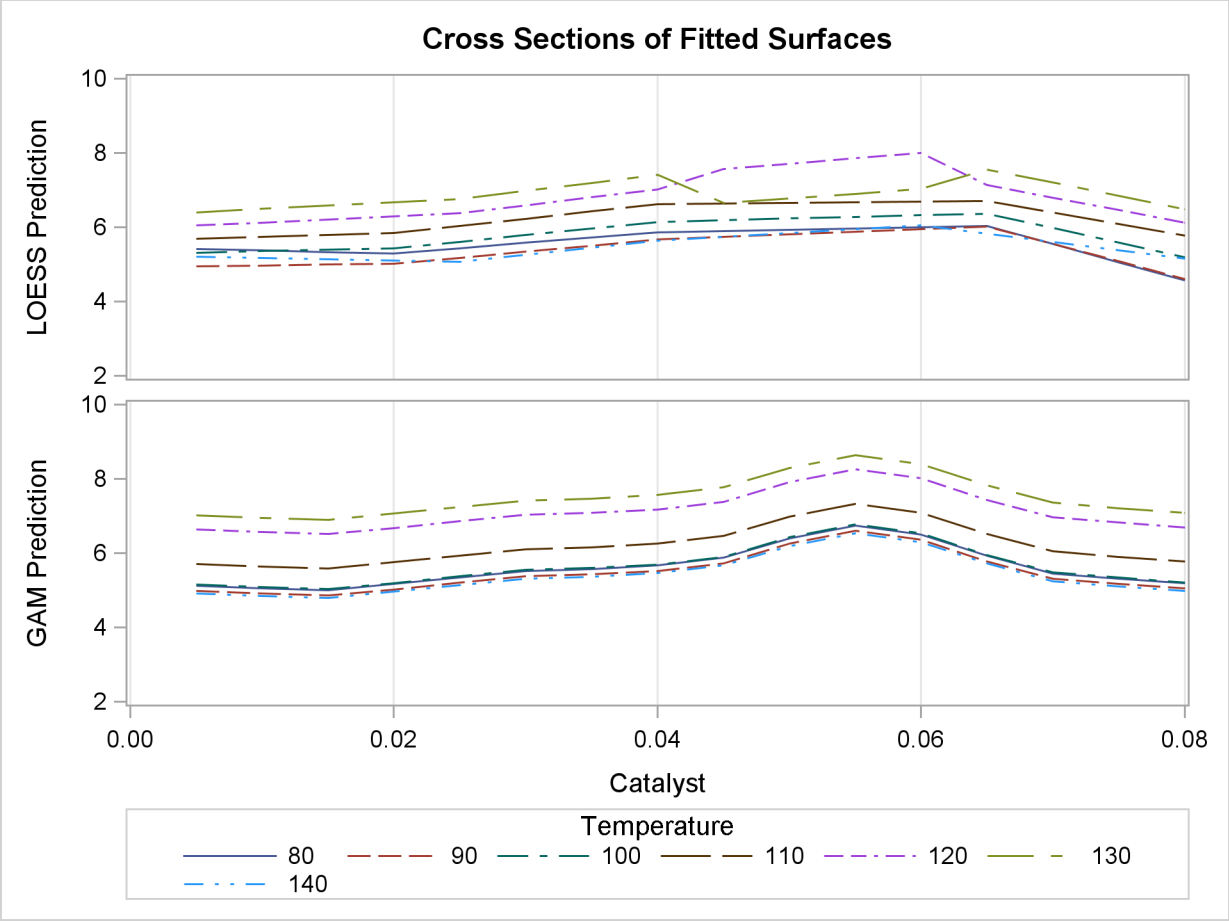
The advantage of using additive model fitting is that its statistical power is directed toward univariate smoothing, and so it is able to discern the finer details of any underlying structure in the data. Regression data can be very sparse when viewed in the context of multidimensional space, even when every individual set of regressor values densely covers its range. This is the familiar curse of dimensionality. Sparse data greatly restrict the effectiveness of nonparametric procedures, but additive model fitting, when appropriate, is one way to overcome this limitation.

To examine these properties, you can use ODS Graphics to generate plots of cross sections of unrestricted (PROC LOESS) and additive (PROC GAM) fitted surfaces for the variable Catalyst, as shown in the following statements:

```
proc template;
    define statgraph projection;
    begingraph;
        entrytitle "Cross Sections of Fitted Surfaces";
        layout lattice/rows=2 columndatarange=unionall
                       columngutter=10;
            columnAxes;
                columnAxis / display=all griddisplay=auto_on;
            endColumnAxes;

            layout overlay/
                xaxisopts=(display=none)
                yaxisopts=(label="LOESS Prediction"
                linearopts=(viewmin=2 viewmax=10));
                seriesplot x=Catalyst y=p_Yield /
                   group=temperature
                   name="Temperature";
            endlayout;

            layout overlay/
                xaxisopts=(display=none)
                yaxisopts=(label="GAM Prediction"
                linearopts=(viewmin=2 viewmax=10));
                seriesplot x=Catalyst y=Gam_p_Yield /
                   group=temperature
                   name="Temperature";
            endlayout;

            columnheaders;
                discreteLegend "Temperature" / title = "Temperature";
            endcolumnheaders;

        endlayout;
    endgraph;
    end;
run;

proc sgrender data=PredGAM template=projection;
run;
```

The plots are displayed in Output 36.3.3.

Output 36.3.3 Cross Sections of Fitted Regression Surfaces



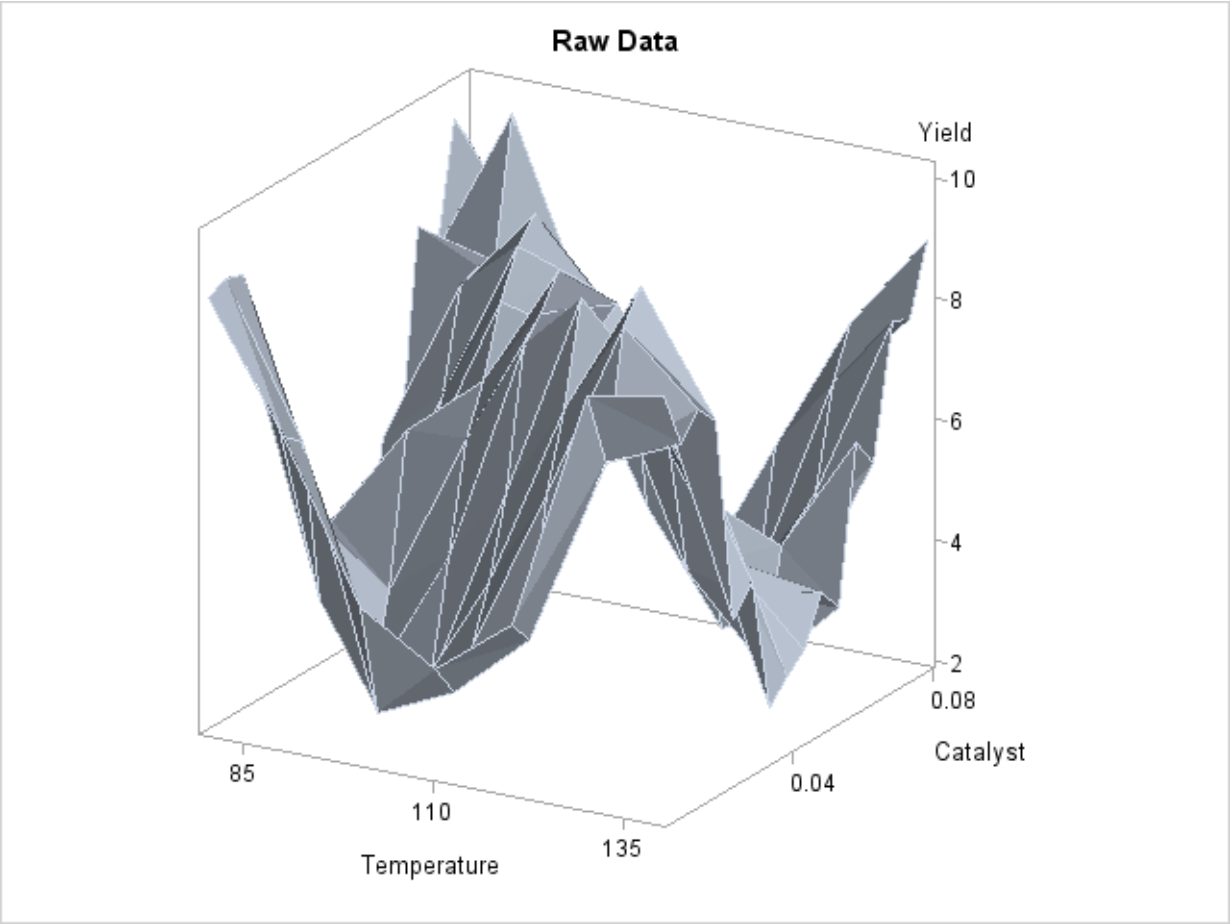
Notice that the cross sections in the top panel (PROC LOESS) of Output 36.3.3 have varying shapes, while every cross section in the bottom panel (PROC GAM) is the same curve shifted vertically. This illustrates precisely the kind of structural differences that distinguish additive models. A second important comparison to make between Output 36.3.2 and Output 36.3.3 is the level of detail in the fitted regression surfaces. Cross sections of the PROC LOESS surface are rather flat, but those of the additive surface have a clear shape. In particular, the ridge near Catalyst=0.055 is only vaguely evident in the PROC LOESS surface, but it is plainly revealed by the additive procedure.

For an example of a situation where unrestricted multidimensional fitting is preferred over additive regression, consider the following simulated data from a similar experiment. The following statements create another SAS data set and plot.

```
data ExperimentB;
    format Temperature f4.0 Catalyst f6.3 Yield f8.3;
    input Temperature Catalyst Yield @@;
datalines;
80  0.005  9.115  80 0.010  9.275  80 0.015  9.160
80  0.020  7.065  80 0.025  6.054  80 0.030  4.899
80  0.035  4.504  80 0.040  4.238  80 0.045  3.232
80  0.050  3.135  80 0.055  5.100  80 0.060  4.802
80  0.065  8.218  80 0.070  7.679  80 0.075  9.669
80  0.080  9.071  90 0.005  7.085  90 0.010  6.814
90  0.015  4.009  90 0.020  4.199  90 0.025  3.377
90  0.030  2.141  90 0.035  3.500  90 0.040  5.967
90  0.045  5.268  90 0.050  6.238  90 0.055  7.847
90  0.060  7.992  90 0.065  7.904  90 0.070 10.184
90  0.075  7.914  90 0.080  6.842 100 0.005  4.497
100 0.010  2.565 100 0.015  2.637 100 0.020  2.436
100 0.025  2.525 100 0.030  4.474 100 0.035  6.238
100 0.040  7.029 100 0.045  8.183 100 0.050  8.939
100 0.055  9.283 100 0.060  8.246 100 0.065  6.927
100 0.070  7.062 100 0.075  5.615 100 0.080  4.687
110 0.005  3.706 110 0.010  3.154 110 0.015  3.726
110 0.020  4.634 110 0.025  5.970 110 0.030  8.219
110 0.035  8.590 110 0.040  9.097 110 0.045  7.887
110 0.050  8.480 110 0.055  6.818 110 0.060  7.666
110 0.065  4.375 110 0.070  3.994 110 0.075  3.630
110 0.080  2.685 120 0.005  4.697 120 0.010  4.268
120 0.015  6.507 120 0.020  7.747 120 0.025  9.412
120 0.030  8.761 120 0.035  8.997 120 0.040  7.538
120 0.045  7.003 120 0.050  6.010 120 0.055  3.886
120 0.060  4.897 120 0.065  2.562 120 0.070  2.714
120 0.075  3.141 120 0.080  5.081 130 0.005  8.729
130 0.010  7.460 130 0.015  9.549 130 0.020 10.049
130 0.025  8.131 130 0.030  7.553 130 0.035  6.191
130 0.040  6.272 130 0.045  4.649 130 0.050  3.884
130 0.055  2.522 130 0.060  4.366 130 0.065  3.272
130 0.070  4.906 130 0.075  6.538 130 0.080  7.380
140 0.005  8.991 140 0.010  8.029 140 0.015  8.417
140 0.020  8.049 140 0.025  4.608 140 0.030  5.025
140 0.035  2.795 140 0.040  3.123 140 0.045  3.407
140 0.050  4.183 140 0.055  3.750 140 0.060  6.316
140 0.065  5.799 140 0.070  7.992 140 0.075  7.835
140 0.080  8.985
;

proc sort data=ExperimentB;
    by Temperature Catalyst;
run;

proc sgrender data=ExperimentB template=surface;
    dynamic _X='Temperature' _Y='Catalyst' _Z='Yield' _T='Raw Data';
run;
```

A plot of the raw data is displayed in Output 36.3.4.

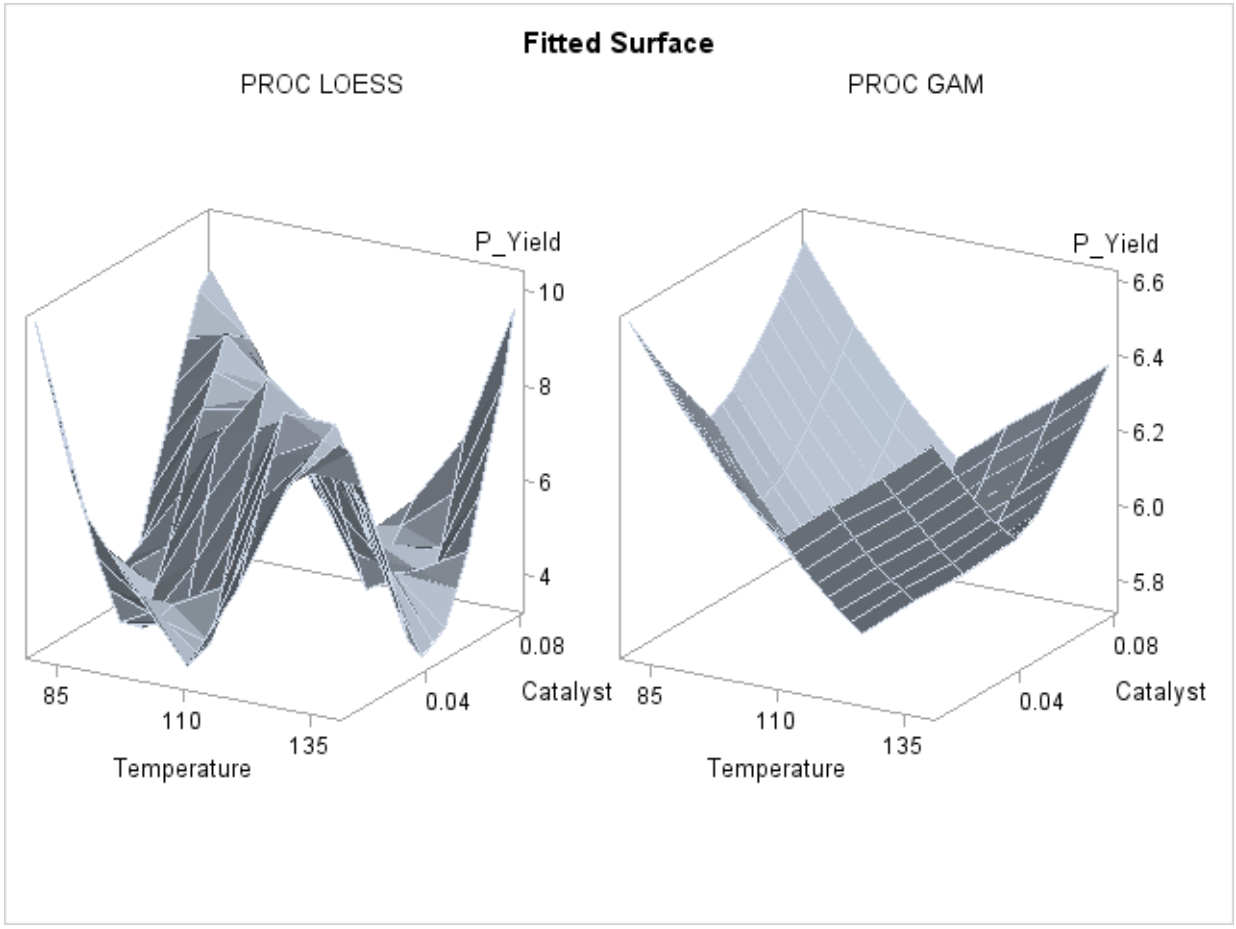
Output 36.3.4 Raw Data from Experiment B



Though the surface displayed in Output 36.3.4 is quite jagged, a distinct feature of the plot is a large ridge that runs diagonally across its surface. One would expect that the ridge would appear in the fitted regression surface of an appropriate nonparametric procedure. Nevertheless, between PROC LOESS and PROC GAM, only PROC LOESS is able to capture this significant feature.

The SAS program for fitting the new data is essentially the same as that for the data set from the first experiment. As in Output 36.3.2, multivariate and additive fitted surfaces for these data are displayed in Output 36.3.5.

Output 36.3.5 Fitted Regression Surfaces



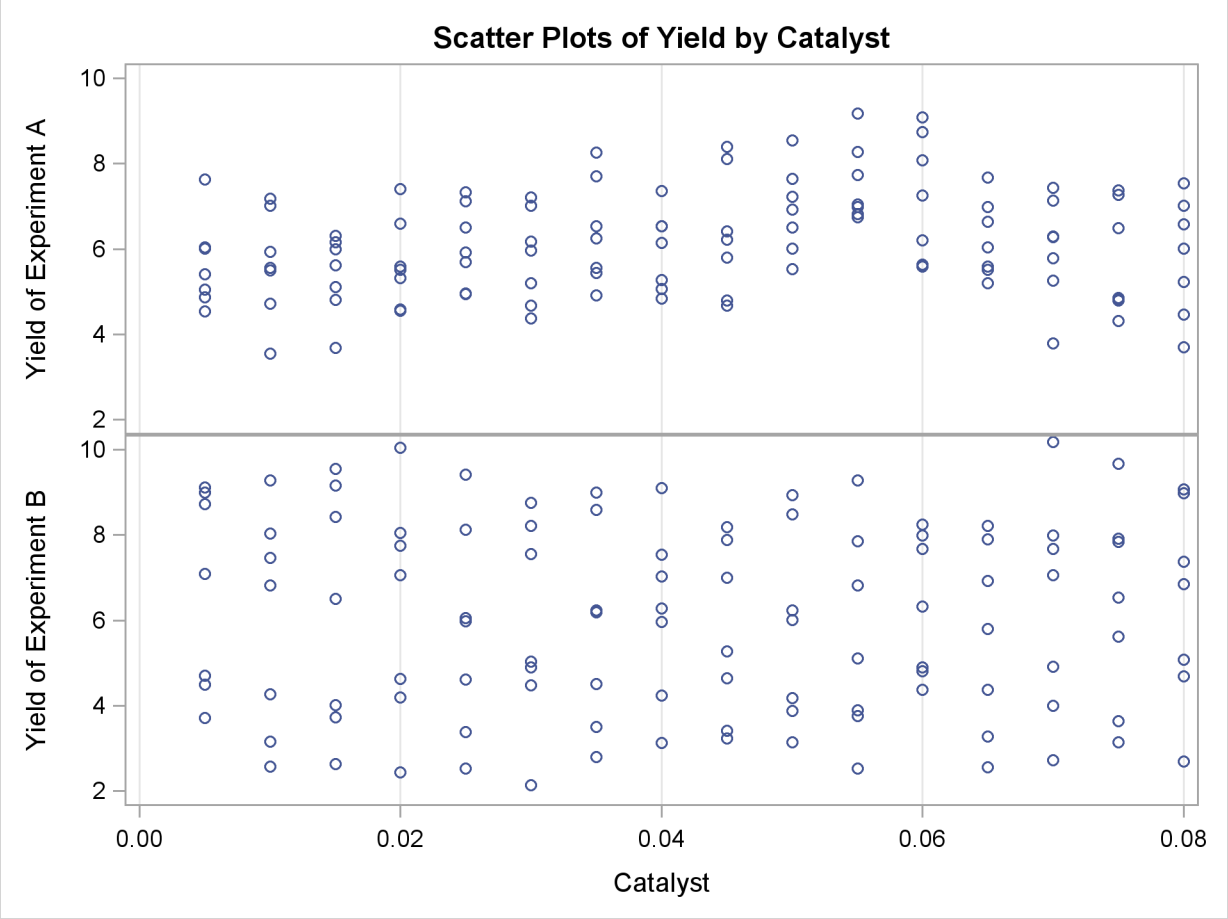
It is clear from Output 36.3.5 that the results of PROC LOESS and PROC GAM are completely different. While the plot in the left panel resembles the raw data plot in Output 36.3.4, the plot in the right panel is essentially featureless.

To understand what is happening, compare the scatter plots of Yield by Catalyst for the two data sets in this example. These are generated by the following statements and displayed in Output 36.3.6.

```
data PredGAM;
    set PredGAM;
        rename Yield=Yield_a;
run;

data PredGAMb;
    set PredGAMb;
    set PredGAM(keep=Yield_a);
run;

proc template;
    define statgraph scatter2;
    dynamic _X _Y1 _Y2;
    begingraph;
        entrytitle "Scatter Plots of Yield by Catalyst";
        layout lattice/rows=2 columndatarange=unionall
                       rowdatarange=unionall
                       columngutter=15;
            columnAxes;
                columnAxis / display=all griddisplay=auto_on;
            endColumnAxes;

            layout overlay/
                xaxisopts=(display=none)
                yaxisopts=(label="Yield of Experiment A"
                linearopts=(viewmin=2 viewmax=10));
                scatterplot x=_X y=_Y1;
            endlayout;

            layout overlay/
                xaxisopts=(display=none)
                yaxisopts=(label="Yield of Experiment B"
                linearopts=(viewmin=2 viewmax=10));
                scatterplot x=_X y=_Y2;
            endlayout;

        endlayout;
    endgraph;
    end;
run;

proc sgrender data=PredGAMb template=scatter2;
    dynamic _X='Catalyst' _Y1='Yield_a' _Y2='Yield';
run;

ods graphics off;
```

The top panel of Output 36.3.6 hints at the same kind of structure exhibited in the fitted cross sections of Output 36.3.3. In PROC GAM, the additive model component corresponding to Catalyst is fit to a similar scatter plot, with the partial residuals computed in the backfitting algorithm, so it is able to capture the trend seen here. In contrast, when the second data set is viewed from the perspective of Output 36.3.6, the diagonal ridge apparent in Output 36.3.4 is washed out, and no clear structure shows up in the scatter plot. As a result, the additive model fit produced by PROC GAM is relatively featureless.

Output 36.3.6 Scatter Plots of Yield by Catalyst



References

Allen, D. M. (1974), "The Relationship between Variable Selection and Data Augmentation and a Method of Prediction," *Technometrics*, 16, 125–127.

Bell, D. F., Walker J. L., O'Connor, G., and Tibshirani, R. J. (1994), "Spinal Deformity after Multiple-Level Cervical Laminectomy in Children." *Spine*, 19, 406–411.

Cleveland, W. S., Devlin, S. J., and Grosse, E. (1988), "Regression by Local Fitting," *Journal of Econometrics,* 37, 87–114.

Duchon, J. (1976), "Fonctions-Spline et Esperances Conditionnelles de Champs Gaussiens," *Ann. Sci. Univ. Clermont Ferrand II Math.*, 14, 19–27.

Duchon, J. (1977), "Splines Minimizing Rotation-Invariant Semi-norms in Sobolev Spaces," in *Constructive Theory of Functions of Several Variables*, ed. W. Schempp and K. Zeller, New York: Springer-Verlag, 85–100.

Friedman, J. H. and Stuetzle, W. (1981), "Projection Pursuit Regression," *Journal of the American Statistical Association*, 76, 817–823.

Hastie, T. J. (1991), "Generalized Additive Models," in *Statistical Models in S*, ed. J. M. Chambers and T. J. Hastie, Pacific Grove: Wadsworth & Brooks/Cole Advanced Books & Software, 249–307.

Hastie, T. J. and Tibshirani, R. J. (1986), "Generalized Additive Models (with discussion)," *Statistical Science*, 1, 297–318.

Hastie, T. J. and Tibshirani, R. J. (1990), *Generalized Additive Models*, New York: Chapman & Hall.

Houghton, A. N., Flannery, J., and Viola, M. V. (1980), "Malignant Melanoma in Connecticut and Denmark," *International Journal of Cancer*, 25, 95–104.

McCullagh, P. and Nelder, J. A. (1989), *Generalized Linear Models*, Second Edition, London: Chapman & Hall.

Meinguet, J. (1979), "Multivariate Interpolation at Arbitrary Points Made Simple," *Journal of Applied Mathematics and Physics (ZAMP)*, 30, 292–304.

Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized Linear Models," *Journal of the Royal Statistical Society, Series A*, 135, 370–384.

SAS Institute Inc. (1999), *SAS Language Reference: Concepts, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *SAS Language Reference: Dictionary, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (1999), *SAS Procedures Guide, Version 8*, Cary, NC: SAS Institute Inc.

SAS Institute Inc. (2004), *SAS/STAT 9.1 User's Guide*, Cary, NC: SAS Institute Inc.

Sockett, E. B., Daneman, D., Clarson, C., and Ehrich, R. M. (1987), "Factors Affecting and Patterns of Residual Insulin Secretion during the First Year of Type I (Insulin Dependent) Diabetes Mellitus in Children," *Diabetologia*, 30, 453–459.

Stone, C. J. (1985), "Additive Regression and Other Nonparametric Models," *Annals of Statistics*, 13, 689–705.

Wahba, G. (1983), "Bayesian 'Confidence Intervals' for the Cross Validated Smoothing Spline," *Journal of the Royal Statistical Society, Series B*, 45, 133–150.

Wahba, G. (1990), *Spline Models for Observational Data,* Philadelphia: Society for Industrial and

Applied Mathematics.

Wahba, G. and Wendelberger, J. (1980), "Some New Mathematical Methods for Variational Objective Analysis Using Splines and Cross Validation," *Monthly Weather Review*, 108, 1122–1145.

Chapter 37

The GENMOD Procedure

Contents

Overview: GENMOD Procedure **1894**
What Is a Generalized Linear Model? 1896
Examples of Generalized Linear Models 1897
The GENMOD Procedure 1898
Getting Started: GENMOD Procedure **1900**
Poisson Regression 1900
Bayesian Analysis of a Linear Regression Model 1906
Generalized Estimating Equations 1918
Syntax: GENMOD Procedure **1921**
PROC GENMOD Statement 1921
ASSESS Statement 1926
BAYES Statement 1928
BY Statement 1937
CLASS Statement 1937
CONTRAST Statement 1940
DEVIANCE Statement 1942
ESTIMATE Statement 1943
FREQ Statement 1944
FWDLINK Statement 1944
INVLINK Statement 1944
LSMEANS Statement 1945
MODEL Statement 1946
OUTPUT Statement 1954
Programming Statements 1957
REPEATED Statement 1958
VARIANCE Statement 1962
WEIGHT Statement 1962
ZEROMODEL Statement 1963
Details: GENMOD Procedure **1964**
Generalized Linear Models Theory 1964
Specification of Effects 1974
Parameterization Used in PROC GENMOD 1975
CLASS Variable Parameterization 1975
Type 1 Analysis 1978

Type 3 Analysis . . . 1979
Confidence Intervals for Parameters . . . 1980
F Statistics . . . 1981
Lagrange Multiplier Statistics . . . 1981
Predicted Values of the Mean . . . 1982
Residuals . . . 1983
Multinomial Models . . . 1984
Zero-Inflated Poisson Models . . . 1985
Generalized Estimating Equations . . . 1986
Assessment of Models Based on Aggregates of Residuals . . . 1995
Case Deletion Diagnostic Statistics . . . 1999
Bayesian Analysis . . . 2003
Missing Values . . . 2007
Displayed Output for Classical Analysis . . . 2007
Displayed Output for Bayesian Analysis . . . 2016
ODS Table Names . . . 2019
ODS Graphics . . . 2022
Examples: GENMOD Procedure . . . 2024
Example 37.1: Logistic Regression . . . 2024
Example 37.2: Normal Regression, Log Link . . . 2027
Example 37.3: Gamma Distribution Applied to Life Data . . . 2029
Example 37.4: Ordinal Model for Multinomial Data . . . 2032
Example 37.5: GEE for Binary Data with Logit Link Function . . . 2036
Example 37.6: Log Odds Ratios and the ALR Algorithm . . . 2039
Example 37.7: Log-Linear Model for Count Data . . . 2042
Example 37.8: Model Assessment of Multiple Regression Using Aggregates of Residuals . . . 2047
Example 37.9: Assessment of a Marginal Model for Dependent Data . . . 2054
Example 37.10: Bayesian Analysis of a Poisson Regression Model . . . 2058
References . . . 2072

Overview: GENMOD Procedure

The GENMOD procedure fits generalized linear models, as defined by Nelder and Wedderburn (1972). The class of generalized linear models is an extension of traditional linear models that allows the mean of a population to depend on a *linear predictor* through a nonlinear *link function* and allows the response probability distribution to be any member of an exponential family of distributions. Many widely used statistical models are generalized linear models. These include classical linear models with normal errors, logistic and probit models for binary data, and log-linear models for multinomial data. Many other useful statistical models can be formulated as generalized linear models by the selection of an appropriate link function and response probability distribution.

See McCullagh and Nelder (1989) for a discussion of statistical modeling using generalized linear models. The books by Aitkin et al. (1989) and Dobson (1990) are also excellent references with many examples of applications of generalized linear models. Firth (1991) provides an overview of generalized linear models.

The analysis of correlated data arising from repeated measurements when the measurements are assumed to be multivariate normal has been studied extensively. However, the normality assumption might not always be reasonable; for example, different methodology must be used in the data analysis when the responses are discrete and correlated. Generalized estimating equations (GEEs) provide a practical method with reasonable statistical efficiency to analyze such data.

Liang and Zeger (1986) introduced GEEs as a method of dealing with correlated data when, except for the correlation among responses, the data can be modeled as a generalized linear model. For example, correlated binary and count data in many cases can be modeled in this way.

The GENMOD procedure can fit models to correlated responses by the GEE method. You can use PROC GENMOD to fit models with most of the correlation structures from Liang and Zeger (1986) by using GEEs. See Hardin and Hilbe (2003), Diggle, Liang, and Zeger (1994), and Lipsitz et al. (1994) for more details on GEEs.

Bayesian analysis of generalized linear models can be requested by using the BAYES statement in the GENMOD procedure. In Bayesian analysis, the model parameters are treated as random variables, and inference about parameters is based on the posterior distribution of the parameters, given the data. The posterior distribution is obtained using Bayes' theorem as the likelihood function of the data weighted with a prior distribution. The prior distribution enables you to incorporate knowledge or experience of the likely range of values of the parameters of interest into the analysis. If you have no prior knowledge of the parameter values, you can use a noninformative prior distribution, and the results of the Bayesian analysis will be very similar to a classical analysis based on maximum likelihood. A closed form of the posterior distribution is often not feasible, and a Markov chain Monte Carlo method by Gibbs sampling is used to simulate samples from the posterior distribution. See Chapter 7, "Introduction to Bayesian Analysis Procedures," for an introduction to the basic concepts of Bayesian statistics. Also see the section "Bayesian Analysis: Advantages and Disadvantages" on page 149 for a discussion of the advantages and disadvantages of Bayesian analysis. See Ibrahim, Chen, and Sinha (2001) for a detailed description of Bayesian analysis.

In a Bayesian analysis, a Gibbs chain of samples from the posterior distribution is generated for the model parameters. Summary statistics (mean, standard deviation, quartiles, HPD and credible intervals, correlation matrix) and convergence diagnostics (autocorrelations; Gelman-Rubin, Geweke, Raftery-Lewis, and Heidelberger and Welch tests; the effective sample size; and Monte Carlo standard errors) are computed for each parameter, as well as the correlation matrix and the covariance matrix of the posterior sample. Trace plots, posterior density plots, and autocorrelation function plots that are created using ODS Graphics are also provided for each parameter.

The GENMOD procedure now uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

What Is a Generalized Linear Model?

A traditional linear model is of the form

$$y_i = \mathbf{x}_i'\boldsymbol{\beta} + \varepsilon_i$$

where y_i is the response variable for the ith observation. The quantity $\mathbf{x}_i$ is a column vector of covariates, or explanatory variables, for observation i that is known from the experimental setting and is considered to be fixed, or nonrandom. The vector of unknown coefficients $\boldsymbol{\beta}$ is estimated by a least squares fit to the data $\mathbf{y}$. The ε_i are assumed to be independent, normal random variables with zero mean and constant variance. The expected value of y_i, denoted by μ_i, is

$$\mu_i = \mathbf{x}_i'\boldsymbol{\beta}$$

While traditional linear models are used extensively in statistical data analysis, there are types of problems such as the following for which they are not appropriate.

- It might not be reasonable to assume that data are normally distributed. For example, the normal distribution (which is continuous) might not be adequate for modeling counts or measured proportions that are considered to be discrete.
- If the mean of the data is naturally restricted to a range of values, the traditional linear model might not be appropriate, since the linear predictor $\mathbf{x}_i'\boldsymbol{\beta}$ can take on any value. For example, the mean of a measured proportion is between 0 and 1, but the linear predictor of the mean in a traditional linear model is not restricted to this range.
- It might not be realistic to assume that the variance of the data is constant for all observations. For example, it is not unusual to observe data where the variance increases with the mean of the data.

A generalized linear model extends the traditional linear model and is therefore applicable to a wider range of data analysis problems. A generalized linear model consists of the following components:

- The linear component is defined just as it is for traditional linear models:

  $$\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$$

- A monotonic differentiable link function g describes how the expected value of y_i is related to the linear predictor η_i:

  $$g(\mu_i) = \mathbf{x}_i'\boldsymbol{\beta}$$

- The response variables y_i are independent for $i = 1, 2, \ldots$ and have a probability distribution from an exponential family. This implies that the variance of the response depends on the mean μ through a *variance function* V:

  $$var(y_i) = \frac{\phi V(\mu_i)}{w_i}$$

  where ϕ is a constant and w_i is a known weight for each observation. The *dispersion parameter* ϕ is either known (for example, for the binomial or Poisson distribution, $\phi = 1$) or must be estimated.

See the section "Response Probability Distributions" on page 1964 for the form of a probability distribution from the exponential family of distributions.

As in the case of traditional linear models, fitted generalized linear models can be summarized through statistics such as parameter estimates, their standard errors, and goodness-of-fit statistics. You can also make statistical inference about the parameters by using confidence intervals and hypothesis tests. However, specific inference procedures are usually based on asymptotic considerations, since exact distribution theory is not available or is not practical for all generalized linear models.

Examples of Generalized Linear Models

You construct a generalized linear model by deciding on response and explanatory variables for your data and choosing an appropriate link function and response probability distribution. Some examples of generalized linear models follow. Explanatory variables can be any combination of continuous variables, classification variables, and interactions.

Traditional Linear Model

- response variable: a continuous variable
- distribution: normal
- link function: identity $g(\mu) = \mu$

Logistic Regression

- response variable: a proportion
- distribution: binomial
- link function: logit $g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$

Poisson Regression in Log-Linear Model

- response variable: a count
- distribution: Poisson
- link function: log $g(\mu) = \log(\mu)$

Gamma Model with Log Link

- response variable: a positive, continuous variable

- distribution: gamma
- link function: $\log g(\mu) = \log(\mu)$

The GENMOD Procedure

The GENMOD procedure fits a generalized linear model to the data by maximum likelihood estimation of the parameter vector $\boldsymbol{\beta}$. There is, in general, no closed form solution for the maximum likelihood estimates of the parameters. The GENMOD procedure estimates the parameters of the model numerically through an iterative fitting process. The dispersion parameter ϕ is also estimated by maximum likelihood or, optionally, by the residual deviance or by Pearson's chi-square divided by the degrees of freedom. Covariances, standard errors, and p-values are computed for the estimated parameters based on the asymptotic normality of maximum likelihood estimators.

A number of popular link functions and probability distributions are available in the GENMOD procedure. The built-in link functions are as follows:

- identity: $g(\mu) = \mu$
- logit: $g(\mu) = \log(\mu/(1-\mu))$
- probit: $g(\mu) = \Phi^{-1}(\mu)$, where Φ is the standard normal cumulative distribution function
- power: $g(\mu) = \begin{cases} \mu^{\lambda} & \text{if } \lambda \neq 0 \\ \log(\mu) & \text{if } \lambda = 0 \end{cases}$
- log: $g(\mu) = \log(\mu)$
- complementary log-log: $g(\mu) = \log(-\log(1-\mu))$

The available distributions and associated variance functions are as follows:

- normal: $V(\mu) = 1$
- binomial (proportion): $V(\mu) = \mu(1-\mu)$
- Poisson: $V(\mu) = \mu$
- gamma: $V(\mu) = \mu^2$
- inverse Gaussian: $V(\mu) = \mu^3$
- negative binomial: $V(\mu) = \mu + k\mu^2$
- geometric: $V(\mu) = \mu + \mu^2$
- multinomial
- zero-inflated Poisson

The negative binomial is a distribution with an additional parameter k in the variance function. PROC GENMOD estimates k by maximum likelihood, or you can optionally set it to a constant value. See McCullagh and Nelder (1989), Hilbe (1994), or Lawless (1987) for discussions of the negative binomial distribution.

The multinomial distribution is sometimes used to model a response that can take values from a number of categories. The binomial is a special case of the multinomial with two categories. See the section "Multinomial Models" on page 1984 and McCullagh and Nelder (1989, Chapter 5) for a description of the multinomial distribution.

The zero-inflated Poisson is included in PROC GENMOD even though it is not a generalized linear model. It is a useful extension of generalized linear models. See the section "Zero-Inflated Poisson Models" on page 1985 for information about the zero-inflated Poisson.

In addition, you can easily define your own link functions or distributions through DATA step programming statements used within the procedure.

An important aspect of generalized linear modeling is the selection of explanatory variables in the model. Changes in goodness-of-fit statistics are often used to evaluate the contribution of subsets of explanatory variables to a particular model. The deviance, defined to be twice the difference between the maximum attainable log likelihood and the log likelihood of the model under consideration, is often used as a measure of goodness of fit. The maximum attainable log likelihood is achieved with a model that has a parameter for every observation. See the section "Goodness of Fit" on page 1970 for formulas for the deviance.

One strategy for variable selection is to fit a sequence of models, beginning with a simple model with only an intercept term, and then to include one additional explanatory variable in each successive model. You can measure the importance of the additional explanatory variable by the difference in deviances or fitted log likelihoods between successive models. Asymptotic tests computed by the GENMOD procedure enable you to assess the statistical significance of the additional term.

The GENMOD procedure enables you to fit a sequence of models, up through a maximum number of terms specified in a MODEL statement. A table summarizes twice the difference in log likelihoods between each successive pair of models. This is called a *Type 1* analysis in the GENMOD procedure, because it is analogous to Type I (sequential) sums of squares in the GLM procedure. As with the PROC GLM Type I sums of squares, the results from this process depend on the order in which the model terms are fit.

The GENMOD procedure also generates a *Type 3* analysis analogous to Type III sums of squares in the GLM procedure. A Type 3 analysis does not depend on the order in which the terms for the model are specified. A GENMOD procedure Type 3 analysis consists of specifying a model and computing likelihood ratio statistics for Type III contrasts for each term in the model. The contrasts are defined in the same way as they are in the GLM procedure. The GENMOD procedure optionally computes Wald statistics for Type III contrasts. This is computationally less expensive than likelihood ratio statistics, but it is thought to be less accurate because the specified significance level of hypothesis tests based on the Wald statistic might not be as close to the actual significance level as it is for likelihood ratio tests.

A Type 3 analysis generalizes the use of Type III estimable functions in linear models. Briefly, a Type III estimable function (contrast) for an effect is a linear function of the model parameters that involves the parameters of the effect and any interactions with that effect. A test of the hypothesis

that the Type III contrast for a main effect is equal to 0 is intended to test the significance of the main effect in the presence of interactions. See Chapter 39, "The GLM Procedure," and Chapter 15, "The Four Types of Estimable Functions," for more information about Type III estimable functions. Also refer to Littell, Freund, and Spector (1991).

Additional features of the GENMOD procedure include the following:

- likelihood ratio statistics for user-defined contrasts—that is, linear functions of the parameters and p-values based on their asymptotic chi-square distributions
- estimated values, standard errors, and confidence limits for user-defined contrasts and least squares means
- ability to create a SAS data set corresponding to most tables displayed by the procedure (see Table 37.4 and Table 37.5)
- confidence intervals for model parameters based on either the profile likelihood function or asymptotic normality
- syntax similar to that of PROC GLM for the specification of the response and model effects, including interaction terms and automatic coding of classification variables
- ability to fit GEE models for clustered response data
- ability to perform Bayesian analysis by Gibbs sampling

Getting Started: GENMOD Procedure

Poisson Regression

You can use the GENMOD procedure to fit a variety of statistical models. A typical use of PROC GENMOD is to perform Poisson regression.

You can use the Poisson distribution to model the distribution of cell counts in a multiway contingency table. Aitkin et al. (1989) have used this method to model insurance claims data. Suppose the following hypothetical insurance claims data are classified by two factors: age group (with two levels) and car type (with three levels).

```
data insure;
   input n c car$ age;
   ln = log(n);
   datalines;
500   42  small  1
1200  37  medium 1
100    1  large  1
400  101  small  2
500   73  medium 2
300   14  large  2
;
run;
```

In the preceding data set, the variable n represents the number of insurance policyholders and the variable c represents the number of insurance claims. The variable car is the type of car involved (classified into three groups) and the variable age is the age group of a policyholder (classified into two groups).

You can use PROC GENMOD to perform a Poisson regression analysis of these data with a log link function. This type of model is sometimes called a *log-linear model*.

Assume that the number of claims c has a Poisson probability distribution and that its mean, μ_i, is related to the factors car and age for observation i by

$$\begin{aligned}\log(\mu_i) &= \log(\mathsf{n}_i) + \mathbf{x}_i'\boldsymbol{\beta} \\ &= \log(\mathsf{n}_i) + \beta_0 + \\ &\quad \mathsf{car}_i(1)\beta_1 + \mathsf{car}_i(2)\beta_2 + \mathsf{car}_i(3)\beta_3 + \\ &\quad \mathsf{age}_i(1)\beta_4 + \mathsf{age}_i(2)\beta_5\end{aligned}$$

The indicator variables $\mathsf{car}_i(j)$ and $\mathsf{age}_i(j)$ are associated with the jth level of the variables car and age for observation i

$$\mathsf{car}_i(j) = \begin{cases} 1 & \text{if car} = j \\ 0 & \text{if car} \neq j \end{cases}$$

The βs are unknown parameters to be estimated by the procedure. The logarithm of the variable n is used as an *offset*—that is, a regression variable with a constant coefficient of 1 for each observation. A log-linear relationship between the mean and the factors car and age is specified by the log link function. The log link function ensures that the mean number of insurance claims for each car and age group predicted from the fitted model is positive.

The following statements invoke the GENMOD procedure to perform this analysis:

```
proc genmod data=insure;
   class car age;
   model c = car age / dist   = poisson
                       link   = log
                       offset = ln;
run;
```

The variables car and age are specified as CLASS variables so that PROC GENMOD automatically generates the indicator variables associated with car and age.

The MODEL statement specifies c as the response variable and car and age as explanatory variables. An intercept term is included by default. Thus, the model matrix $\mathbf{X}$ (the matrix that has as its ith row the transpose of the covariate vector for the ith observation) consists of a column of 1s representing the intercept term and columns of 0s and 1s derived from indicator variables representing the levels of the car and age variables.

That is, the model matrix is

$$\mathbf{X} = \left[\begin{array}{c|ccc|cc} 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 & 1 \end{array}\right]$$

where the first column corresponds to the intercept, the next three columns correspond to the variable car, and the last two columns correspond to the variable age.

The response distribution is specified as Poisson, and the link function is chosen to be log. That is, the Poisson mean parameter μ is related to the linear predictor by

$$\log(\mu) = \mathbf{x}_i'\boldsymbol{\beta}$$

The logarithm of n is specified as an offset variable, as is common in this type of analysis. In this case, the offset variable serves to normalize the fitted cell means to a per-policyholder basis, since the total number of claims, not individual policyholder claims, is observed.

PROC GENMOD produces the following default output from the preceding statements.

Figure 37.1 Model Information

```
                    The GENMOD Procedure

                     Model Information

              Data Set              WORK.INSURE
              Distribution              Poisson
              Link Function                 Log
              Dependent Variable              c
              Offset Variable                ln
```

The "Model Information" table displayed in Figure 37.1 provides information about the specified model and the input data set.

Figure 37.2 Class Level Information

Class Level Information

Class	Levels	Values
car	3	large medium small
age	2	1 2

Figure 37.2 displays the "Class Level Information" table, which identifies the levels of the classification variables that are used in the model. Note that car is a character variable, and the values are sorted in alphabetical order. This is the default sort order, but you can select different sort orders with the ORDER= option in the PROC GENMOD statement.

Figure 37.3 Goodness of Fit

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	2	2.8207	1.4103
Scaled Deviance	2	2.8207	1.4103
Pearson Chi-Square	2	2.8416	1.4208
Scaled Pearson X2	2	2.8416	1.4208
Log Likelihood		837.4533	
Full Log Likelihood		-16.4638	
AIC (smaller is better)		40.9276	
AICC (smaller is better)		80.9276	
BIC (smaller is better)		40.0946	

The "Criteria For Assessing Goodness Of Fit" table displayed in Figure 37.3 contains statistics that summarize the fit of the specified model. These statistics are helpful in judging the adequacy of a model and in comparing it with other models under consideration. If you compare the deviance of 2.8207 with its asymptotic chi-square with 2 degrees of freedom distribution, you find that the p-value is 0.24. This indicates that the specified model fits the data reasonably well.

Figure 37.4 Analysis of Parameter Estimates

Analysis Of Maximum Likelihood Parameter Estimates

Parameter		DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square
Intercept		1	−1.3168	0.0903	−1.4937	−1.1398	212.73
car	large	1	−1.7643	0.2724	−2.2981	−1.2304	41.96
car	medium	1	−0.6928	0.1282	−0.9441	−0.4414	29.18
car	small	0	0.0000	0.0000	0.0000	0.0000	.
age	1	1	−1.3199	0.1359	−1.5863	−1.0536	94.34
age	2	0	0.0000	0.0000	0.0000	0.0000	.
Scale		0	1.0000	0.0000	1.0000	1.0000	

Analysis Of Maximum Likelihood Parameter Estimates

Parameter		Pr > ChiSq
Intercept		<.0001
car	large	<.0001
car	medium	<.0001
car	small	.
age	1	<.0001
age	2	.
Scale		

NOTE: The scale parameter was held fixed.

Figure 37.4 displays the "Analysis Of Parameter Estimates" table, which summarizes the results of the iterative parameter estimation process. For each parameter in the model, PROC GENMOD displays columns with the parameter name, the degrees of freedom associated with the parameter, the estimated parameter value, the standard error of the parameter estimate, the confidence intervals, and the Wald chi-square statistic and associated p-value for testing the significance of the parameter to the model. If a column of the model matrix corresponding to a parameter is found to be linearly dependent, or *aliased*, with columns corresponding to parameters preceding it in the model, PROC GENMOD assigns it zero degrees of freedom and displays a value of zero for both the parameter estimate and its standard error.

This table includes a row for a scale parameter, even though there is no free scale parameter in the Poisson distribution. See the section "Response Probability Distributions" on page 1964 for the form of the Poisson probability distribution. PROC GENMOD allows the specification of a scale parameter to fit overdispersed Poisson and binomial distributions. In such cases, the SCALE row indicates the value of the overdispersion scale parameter used in adjusting output statistics. See the section "Overdispersion" on page 1973 for more about overdispersion and the meaning of the SCALE parameter output by the GENMOD procedure. PROC GENMOD displays a note indicating that the scale parameter is fixed—that is, not estimated by the iterative fitting process.

It is usually of interest to assess the importance of the main effects in the model. Type 1 and Type 3 analyses generate statistical tests for the significance of these effects. You can request these analyses with the TYPE1 and TYPE3 options in the MODEL statement, as follows:

```
proc genmod data=insure;
   class car age;
   model c = car age / dist   = poisson
                       link   = log
                       offset = ln
                       type1
                       type3;
run;
```

The results of these analyses are summarized in the figures that follow.

Figure 37.5 Type 1 Analysis

The GENMOD Procedure

LR Statistics For Type 1 Analysis

Source	Deviance	DF	Chi-Square	Pr > ChiSq
Intercept	175.1536			
car	107.4620	2	67.69	<.0001
age	2.8207	1	104.64	<.0001

In the table for Type 1 analysis displayed in Figure 37.5, each entry in the deviance column represents the deviance for the model containing the effect for that row and all effects preceding it in the table. For example, the deviance corresponding to car in the table is the deviance of the model containing an intercept and car. As more terms are included in the model, the deviance decreases.

Entries in the chi-square column are likelihood ratio statistics for testing the significance of the effect added to the model containing all the preceding effects. The chi-square value of 67.69 for car represents twice the difference in log likelihoods between fitting a model with only an intercept term and a model with an intercept and car. Since the scale parameter is set to 1 in this analysis, this is equal to the difference in deviances. Since two additional parameters are involved, this statistic can be compared with a chi-square distribution with two degrees of freedom. The resulting p-value (labeled Pr>Chi) of less than 0.0001 indicates that this variable is highly significant. Similarly, the chi-square value of 104.64 for age represents the difference in log likelihoods between the model with the intercept and car and the model with the intercept, car, and age. This effect is also highly significant, as indicated by the small p-value.

Figure 37.6 Type 3 Analysis

LR Statistics For Type 3 Analysis

Source	DF	Chi-Square	Pr > ChiSq
car	2	72.82	<.0001
age	1	104.64	<.0001

The Type 3 analysis results in the same conclusions as the Type 1 analysis. The Type 3 chi-square value for the car variable, for example, is twice the difference between the log likelihood for the model with the variables Intercept, car, and age included and the log likelihood for the model with the car variable excluded. The hypothesis tested in this case is the significance of the variable car given that the variable age is in the model. In other words, it tests the additional contribution of car in the model.

The values of the Type 3 likelihood ratio statistics for the car and age variables indicate that both of these factors are highly significant in determining the claims performance of the insurance policyholders.

Bayesian Analysis of a Linear Regression Model

Neter et al. (1996) describe a study of 54 patients undergoing a certain kind of liver operation in a surgical unit. The data set Surg contains survival time and certain covariates for each patient. Observations for the first 20 patients in the data set Surg are shown in Figure 37.7.

Figure 37.7 Surgical Unit Data

Obs	x1	x2	x3	x4	y	logy	Logx1
1	6.7	62	81	2.59	200	2.3010	1.90211
2	5.1	59	66	1.70	101	2.0043	1.62924
3	7.4	57	83	2.16	204	2.3096	2.00148
4	6.5	73	41	2.01	101	2.0043	1.87180
5	7.8	65	115	4.30	509	2.7067	2.05412
6	5.8	38	72	1.42	80	1.9031	1.75786
7	5.7	46	63	1.91	80	1.9031	1.74047
8	3.7	68	81	2.57	127	2.1038	1.30833
9	6.0	67	93	2.50	202	2.3054	1.79176
10	3.7	76	94	2.40	203	2.3075	1.30833
11	6.3	84	83	4.13	329	2.5172	1.84055
12	6.7	51	43	1.86	65	1.8129	1.90211
13	5.8	96	114	3.95	830	2.9191	1.75786
14	5.8	83	88	3.95	330	2.5185	1.75786
15	7.7	62	67	3.40	168	2.2253	2.04122
16	7.4	74	68	2.40	217	2.3365	2.00148
17	6.0	85	28	2.98	87	1.9395	1.79176
18	3.7	51	41	1.55	34	1.5315	1.30833
19	7.3	68	74	3.56	215	2.3324	1.98787
20	5.6	57	87	3.02	172	2.2355	1.72277

Consider the model

$$Y = \beta_0 + \beta_1 LogX1 + \beta_2 X2 + \beta_3 X3 + \beta_4 X4 + \epsilon$$

where Y is the survival time, LogX1 is log(blood-clotting score), X2 is a prognostic index, X3 is an enzyme function test score, X4 is a liver function test score, and ϵ is an $N(0, \sigma^2)$ error term.

A question of scientific interest is whether blood clotting score has a positive effect on survival time. Using PROC GENMOD, you can obtain a maximum likelihood estimate of the coefficient and construct a null point hypothesis to test whether β_1 is equal to 0. However, if you are interested in finding the probability that the coefficient is positive, Bayesian analysis offers a convenient alternative. You can use Bayesian analysis to directly estimate the conditional probability, $\Pr(\beta_1 > 0|\mathbf{Y})$, using the posterior distribution samples, which are produced as part of the output by PROC GENMOD.

The example that follows shows how to use PROC GENMOD to carry out a Bayesian analysis of the linear model with a normal error term. The SEED= option is specified to maintain reproducibility; no other options are specified in the BAYES statement. By default, a uniform prior distribution is assumed on the regression coefficients. The uniform prior is a flat prior on the real line with a distribution that reflects ignorance of the location of the parameter, placing equal likelihood on all possible values the regression coefficient can take. Using the uniform prior in the following example, you would expect the Bayesian estimates to resemble the classical results of maximizing the likelihood. If you can elicit an informative prior distribution for the regression coefficients, you should use the COEFFPRIOR= option to specify it. A default noninformative gamma prior is used for the scale parameter σ.

You should make sure that the posterior distribution samples have achieved convergence before using them for Bayesian inference. PROC GENMOD produces three convergence diagnostics by default. If ODS Graphics is enabled as specified in the following SAS statements, diagnostic plots are also displayed. See the section "Assessing Markov Chain Convergence" on page 156 for more information about convergence diagnostics and their interpretation.

Summary statistics of the posterior distribution samples are produced by default. However, these statistics might not be sufficient for carrying out your Bayesian inference, and further processing of the posterior samples might be necessary. The following SAS statements request the Bayesian analysis, and the OUTPOST= option saves the samples in the SAS data set PostSurg for further processing:

```
ods graphics on;
proc genmod data=Surg;
   model y = Logx1 X2 X3 X4 / dist=normal;
   bayes seed=1 OutPost=PostSurg;
run;
ods graphics off;
```

The results of this analysis are shown in the following figures.

The "Model Information" table in Figure 37.8 summarizes information about the model you fit and the size of the simulation.

Figure 37.8 Model Information

The GENMOD Procedure

Bayesian Analysis

Model Information

Data Set	WORK.SURG
Burn-In Size	2000
MC Sample Size	10000
Thinning	1
Distribution	Normal
Link Function	Identity
Dependent Variable	y Survival Time

The "Analysis of Maximum Likelihood Parameter Estimates" table in Figure 37.9 summarizes maximum likelihood estimates of the model parameters.

Figure 37.9 Maximum Likelihood Parameter Estimates

Analysis Of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits	
Intercept	1	-730.559	85.4333	-898.005	-563.112
Logx1	1	171.8758	38.2250	96.9561	246.7954
x2	1	4.3019	0.5566	3.2109	5.3929
x3	1	4.0309	0.4996	3.0517	5.0100
x4	1	18.1377	12.0721	-5.5232	41.7986
Scale	1	59.8591	5.7599	49.5705	72.2832

NOTE: The scale parameter was estimated by maximum likelihood.

Since no prior distributions for the regression coefficients were specified, the default noninformative uniform distributions shown in the "Uniform Prior for Regression Coefficients" table in Figure 37.10 are used. Noninformative priors are appropriate if you have no prior knowledge of the likely range of values of the parameters, and if you want to make probability statements about the parameters or functions of the parameters. See, for example, Ibrahim, Chen, and Sinha (2001) for more information about choosing prior distributions.

Figure 37.10 Regression Coefficient Priors

The GENMOD Procedure

Bayesian Analysis

Uniform Prior for Regression Coefficients

Parameter	Prior
Intercept	Constant
Logx1	Constant
x2	Constant
x3	Constant
x4	Constant

The default noninformative gamma prior distribution for the normal scale parameter is shown in the "Independent Prior Distributions for Model Parameters" table in Figure 37.11.

Figure 37.11 Scale Parameter Prior

Independent Prior Distributions for Model Parameters

		Hyperparameters	
Parameter	Prior Distribution	Shape	Inverse Scale
Scale	Gamma	0.001	0.001

By default, the maximum likelihood estimates of the regression parameters are used as the starting values for the simulation when noninformative prior distributions are used. These are listed in the "Initial Values and Seeds" table in Figure 37.12.

Figure 37.12 MCMC Initial Values and Seeds

Initial Values of the Chain

Chain	Seed	Intercept	Logx1	x2	x3	x4
1	1	-730.559	171.8758	4.301896	4.030878	18.1377

Initial Values of the Chain

Scale
59.26675

Summary statistics for the posterior sample are displayed in the "Fit Statistics," "Descriptive Statistics for the Posterior Sample," "Interval Statistics for the Posterior Sample," and "Posterior Correlation Matrix" tables in Figure 37.13, Figure 37.14, Figure 37.15, and Figure 37.16, respectively.

Figure 37.13 Fit Statistics

Fit Statistics

DIC (smaller is better)	607.406
pD (effective number of parameters)	5.912

Figure 37.14 Descriptive Statistics

The GENMOD Procedure

Bayesian Analysis

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
Intercept	10000	−732.7	90.0925	−792.7	−732.2	−672.2
Logx1	10000	172.2	40.1142	145.5	171.7	198.6
x2	10000	4.3188	0.5924	3.9209	4.3166	4.7103
x3	10000	4.0477	0.5297	3.6923	4.0390	4.3966
x4	10000	17.8269	12.7611	9.3056	18.0202	26.3786
Scale	10000	63.6906	6.6360	59.0550	63.1311	67.7364

Figure 37.15 Interval Statistics

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
Intercept	0.050	−914.1	−558.4	−912.3	−557.7
Logx1	0.050	95.0527	253.0	93.5157	250.5
x2	0.050	3.1553	5.4898	3.1631	5.4928
x3	0.050	3.0250	5.1098	3.0256	5.1099
x4	0.050	−7.4580	42.6537	−7.3998	42.6693
Scale	0.050	52.3244	78.2094	51.2927	76.7926

Figure 37.16 Posterior Sample Correlation Matrix

Posterior Correlation Matrix

Parameter	Intercept	Logx1	x2	x3	x4	Scale
Intercept	1.000	−0.852	−0.575	−0.707	0.575	0.009
Logx1	−0.852	1.000	0.274	0.482	−0.637	−0.010
x2	−0.575	0.274	1.000	0.295	−0.473	0.004
x3	−0.707	0.482	0.295	1.000	−0.613	−0.000
x4	0.575	−0.637	−0.473	−0.613	1.000	−0.009
Scale	0.009	−0.010	0.004	−0.000	−0.009	1.000

Since noninformative prior distributions were used, the posterior sample means, standard deviations, and interval statistics shown in Figure 37.13 and Figure 37.14 are consistent with the maximum likelihood estimates shown in Figure 37.9.

By default, PROC GENMOD computes three convergence diagnostics: the lag1, lag5, lag10, and lag50 autocorrelations (Figure 37.17); Geweke diagnostic statistics (Figure 37.18); and effective sample sizes (Figure 37.19). There is no indication that the Markov chain has not converged. See the section "Assessing Markov Chain Convergence" on page 156 for more information about convergence diagnostics and their interpretation.

Figure 37.17 Posterior Sample Autocorrelations

The GENMOD Procedure

Bayesian Analysis

Posterior Autocorrelations

Parameter	Lag 1	Lag 5	Lag 10	Lag 50
Intercept	0.4525	0.0478	0.0104	−0.0083
Logx1	0.4226	0.0426	0.0078	−0.0052
x2	0.2328	0.0307	0.0123	−0.0057
x3	0.4020	0.0501	0.0009	−0.0034
x4	0.5906	0.0767	0.0124	0.0026
Scale	0.0941	0.0116	−0.0113	0.0050

Figure 37.18 Geweke Diagnostic Statistics

Geweke Diagnostics

Parameter	z	Pr > |z|
Intercept	−1.0049	0.3149
Logx1	1.2048	0.2283
x2	0.4330	0.6650
x3	0.8662	0.3864
x4	−0.9039	0.3661
Scale	−1.0094	0.3128

Figure 37.19 Effective Sample Sizes

Effective Sample Sizes

Parameter	ESS	Correlation Time	Efficiency
Intercept	3083.6	3.2430	0.3084
Logx1	3185.9	3.1389	0.3186
x2	5051.4	1.9797	0.5051
x3	3281.7	3.0472	0.3282
x4	2534.6	3.9454	0.2535
Scale	8107.8	1.2334	0.8108

Trace, autocorrelation, and density plots for the seven model parameters, shown in Figure 37.20 through Figure 37.25, are useful in diagnosing whether the Markov chain of posterior samples has converged. These plots show no evidence that the chain has not converged. See the section "Visual Analysis via Trace Plots" on page 156 for help with interpreting these diagnostic plots.

Figure 37.20 Diagnostic Plots for Intercept

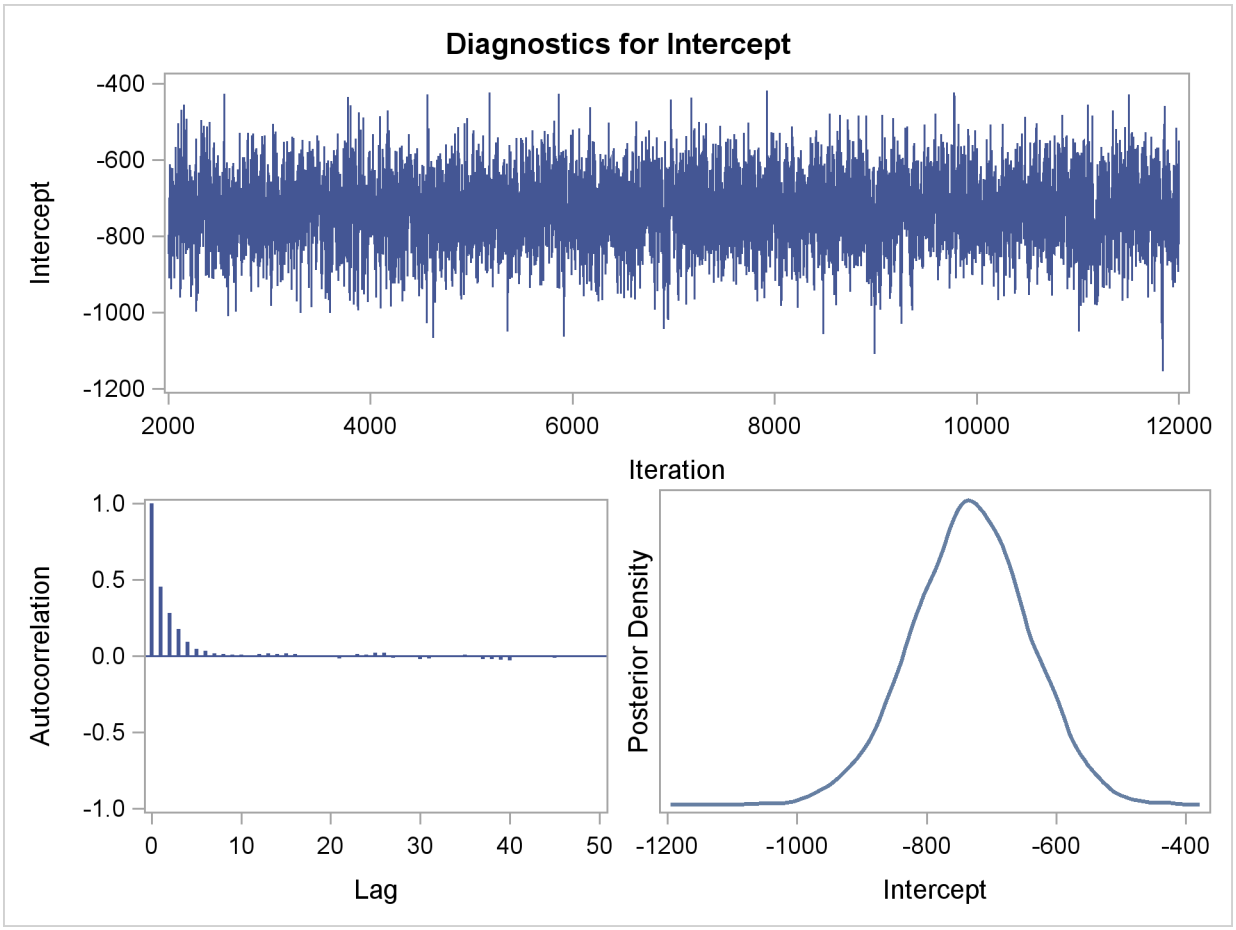


Figure 37.21 Diagnostic Plots for logX1

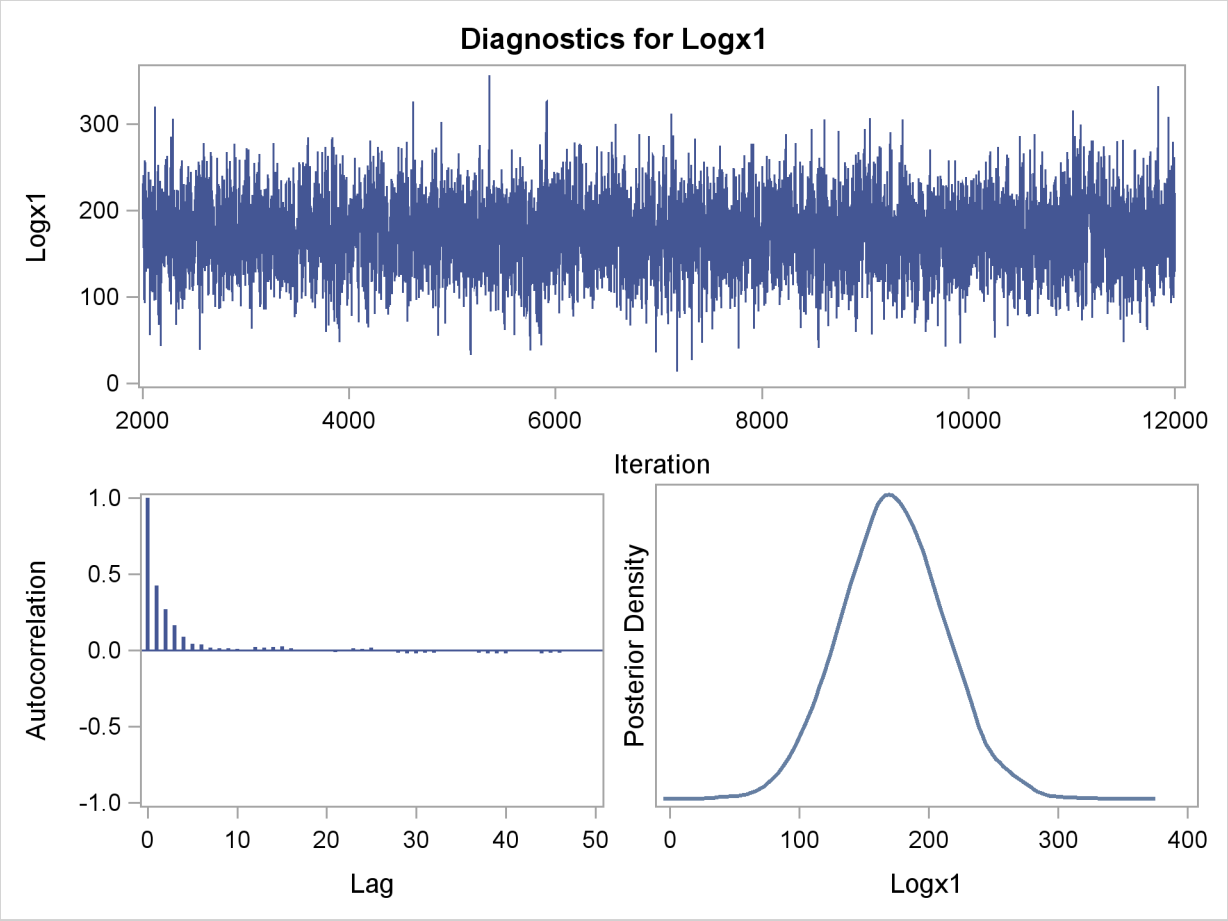


Figure 37.22 Diagnostic Plots for X2

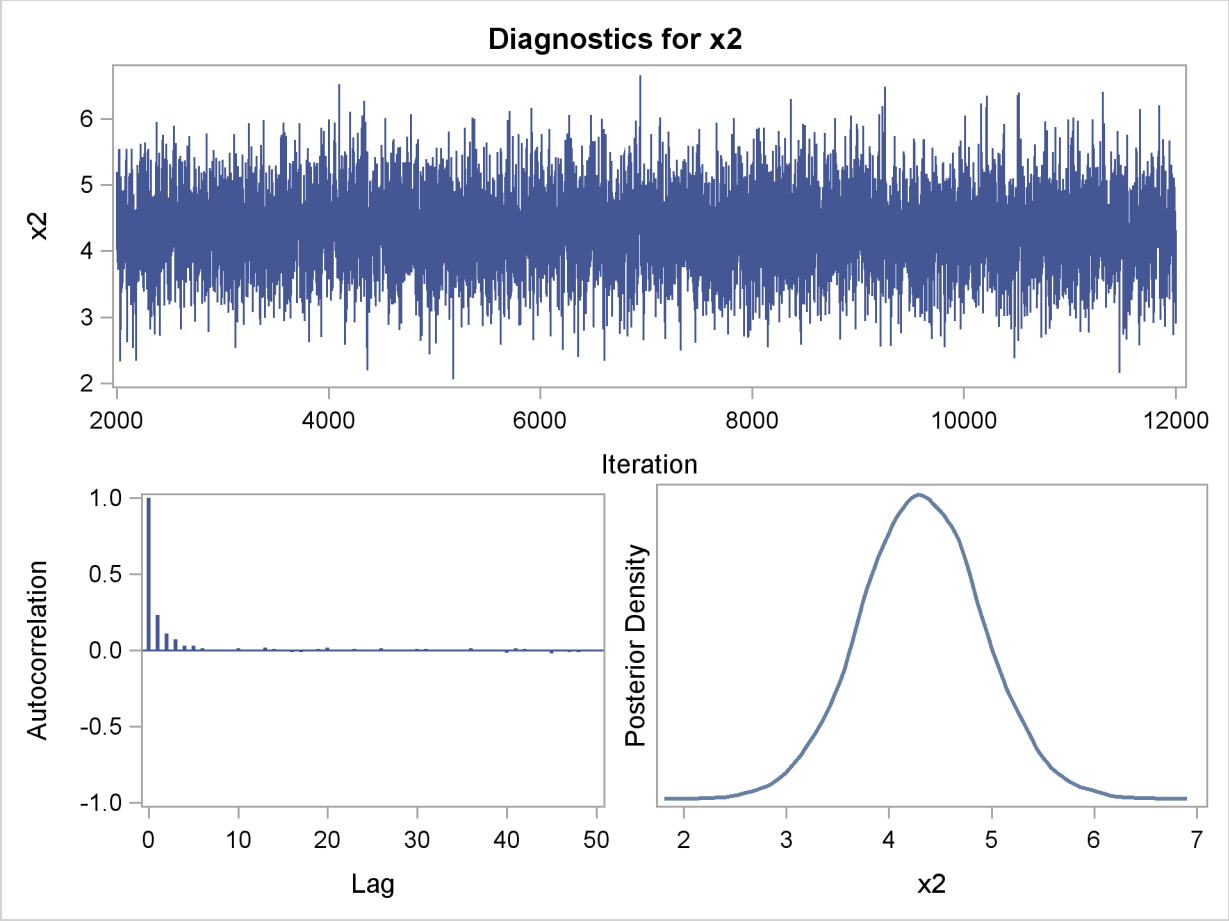


Figure 37.23 Diagnostic Plots for X3

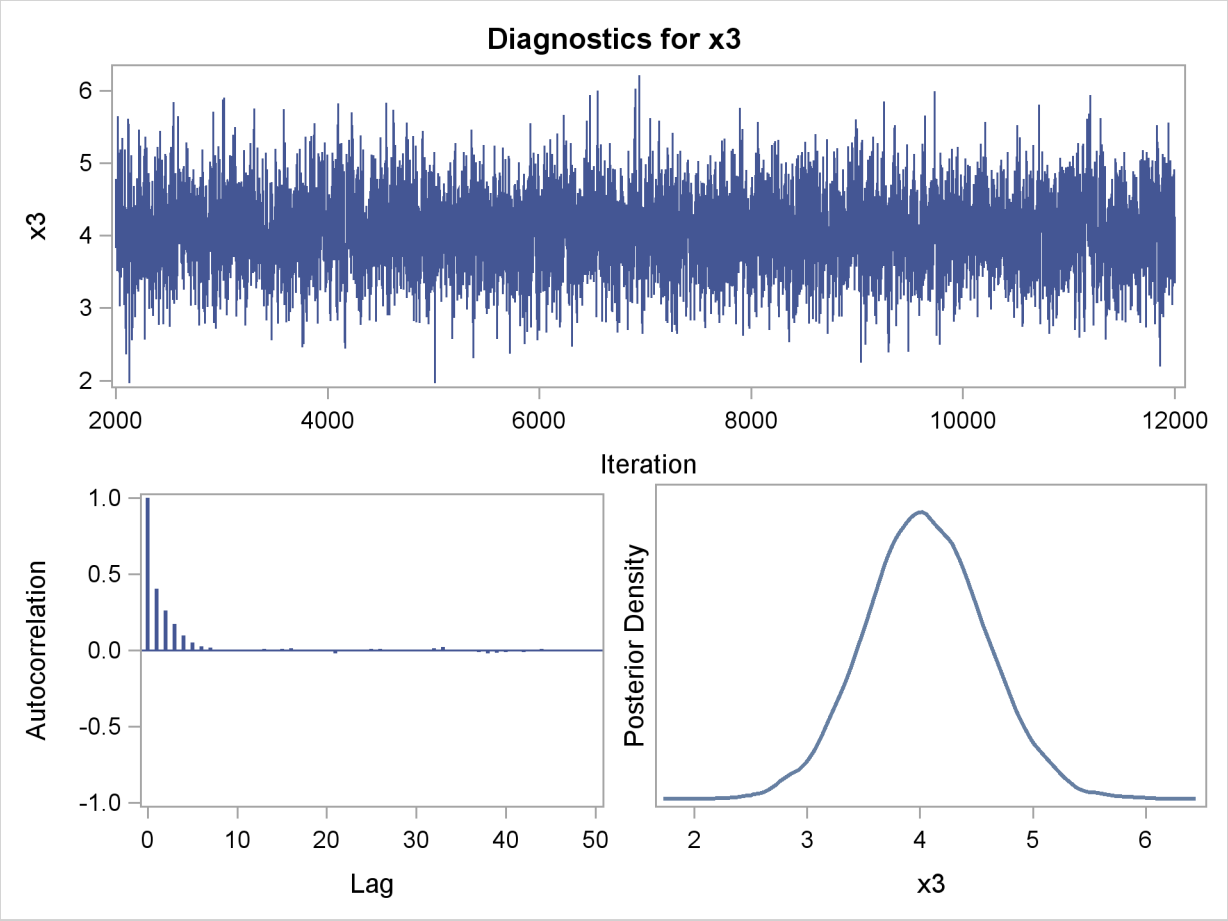


Figure 37.24 Diagnostic Plots for X4

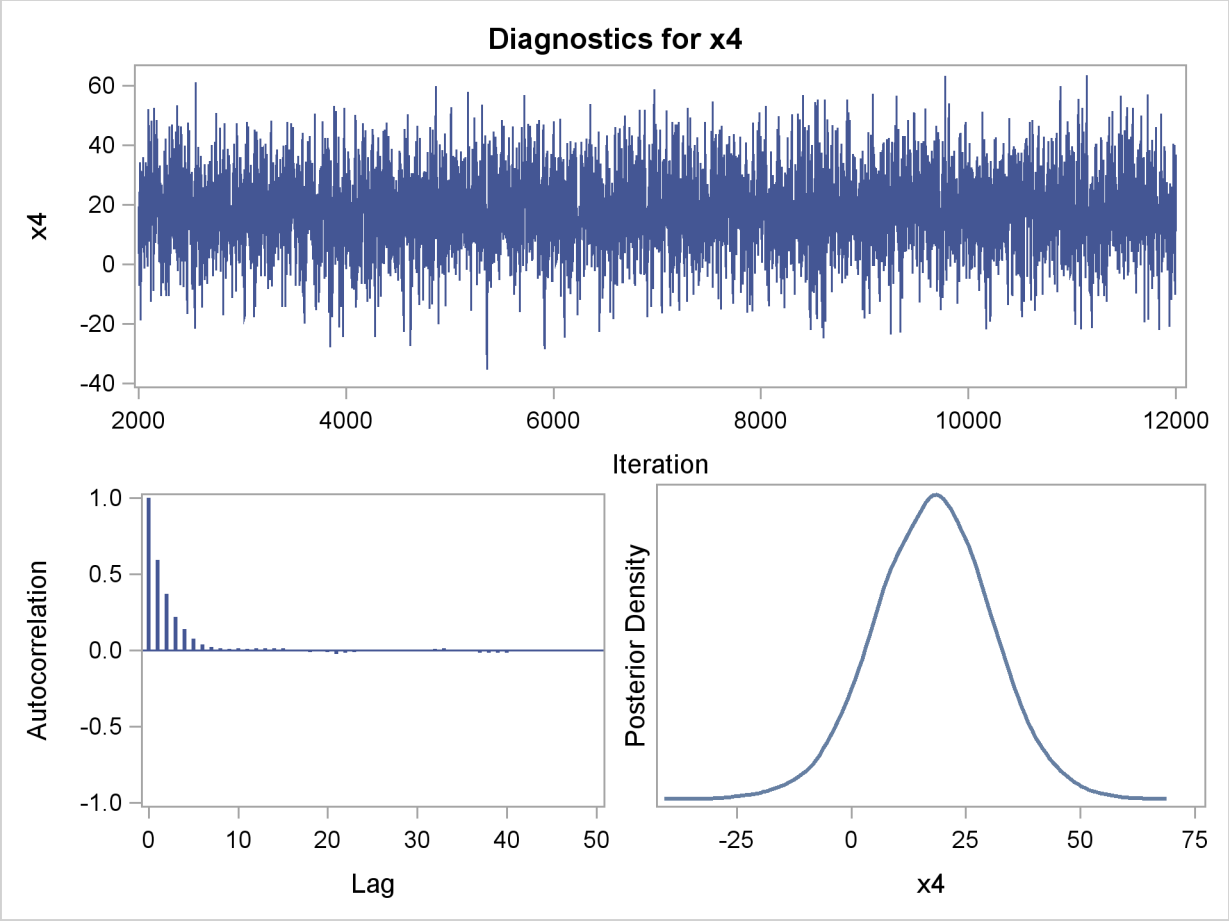
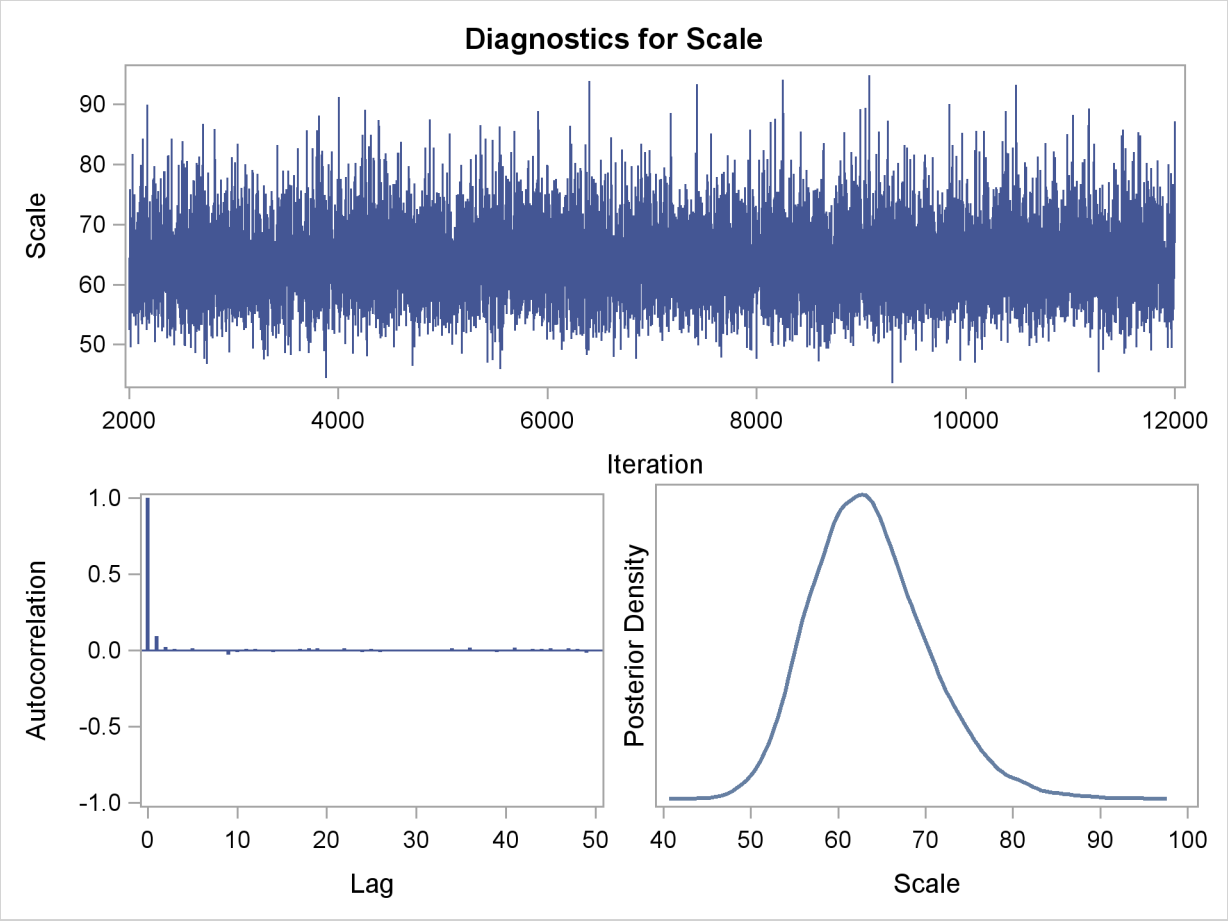


Figure 37.25 Diagnostic Plots for X5



Suppose, for illustration, a question of scientific interest is whether blood clotting score has a positive effect on survival time. Since the model parameters are regarded as random quantities in a Bayesian analysis, you can answer this question by estimating the conditional probability of β_1 being positive, given the data, $\Pr(\beta_1 > 0|\mathbf{Y})$, from the posterior distribution samples. The following SAS statements compute the estimate of the probability of β_1 being positive:

```
data Prob;
   set PostSurg;
   Indicator = (logX1 > 0);
   label Indicator= 'log(Blood Clotting Score) > 0';
run;

proc Means data = Prob(keep=Indicator) n mean;
run;
```

As shown in Figure 37.26, there is a 1.00 probability of a positive relationship between the logarithm of a blood clotting score and survival time, adjusted for the other covariates.

Figure 37.26 Probability That $\beta_1 > 0$

```
                          The MEANS Procedure

        Analysis Variable : Indicator log(Blood Clotting Score) > 0

                             N            Mean
                        ----------------------
                        10000       1.0000000
                        ----------------------
```

Generalized Estimating Equations

This section illustrates the use of the REPEATED statement to fit a GEE model, using repeated measures data from the "Six Cities" study of the health effects of air pollution (Ware et al. 1984). The data analyzed are the 16 selected cases in Lipsitz et al. (1994). The binary response is the wheezing status of 16 children at ages 9, 10, 11, and 12 years. The mean response is modeled as a logistic regression model by using the explanatory variables city of residence, age, and maternal smoking status at the particular age. The binary responses for individual children are assumed to be equally correlated, implying an exchangeable correlation structure.

The data set and SAS statements that fit the model by the GEE method are as follows:

```
data six;
   input case city$ @@;
   do i=1 to 4;
      input age smoke wheeze @@;
      output;
   end;
   datalines;
1 portage   9 0 1  10 0 1  11 0 1  12 0 0
2 kingston  9 1 1  10 2 1  11 2 0  12 2 0
3 kingston  9 0 1  10 0 0  11 1 0  12 1 0
4 portage   9 0 0  10 0 1  11 0 1  12 1 0
5 kingston  9 0 0  10 1 0  11 1 0  12 1 0
6 portage   9 0 0  10 1 0  11 1 0  12 1 0
7 kingston  9 1 0  10 1 0  11 0 0  12 0 0
8 portage   9 1 0  10 1 0  11 1 0  12 2 0
9 portage   9 2 1  10 2 0  11 1 0  12 1 0
10 kingston  9 0 0  10 0 0  11 0 0  12 1 0
11 kingston  9 1 1  10 0 0  11 0 1  12 0 1
12 portage   9 1 0  10 0 0  11 0 0  12 0 0
13 kingston  9 1 0  10 0 1  11 1 1  12 1 1
14 portage   9 1 0  10 2 0  11 1 0  12 2 1
15 kingston  9 1 0  10 1 0  11 1 0  12 2 1
16 portage   9 1 1  10 1 1  11 2 0  12 1 0
;
run;
```

```
proc genmod data=six ;
   class case city ;
   model  wheeze = city age smoke  /  dist=bin;
   repeated  subject=case / type=exch covb corrw;
run;
```

The CLASS statement and the MODEL statement specify the model for the mean of the wheeze variable response as a logistic regression with city, age, and smoke as independent variables, just as for an ordinary logistic regression.

The REPEATED statement invokes the GEE method, specifies the correlation structure, and controls the displayed output from the GEE model. The option SUBJECT=CASE specifies that individual subjects be identified in the input data set by the variable case. The SUBJECT= variable case must be listed in the CLASS statement. Measurements on individual subjects at ages 9, 10, 11, and 12 are in the proper order in the data set, so the WITHINSUBJECT= option is not required. The TYPE=EXCH option specifies an exchangeable working correlation structure, the COVB option specifies that the parameter estimate covariance matrix be displayed, and the CORRW option specifies that the final working correlation be displayed.

Initial parameter estimates for iterative fitting of the GEE model are computed as in an ordinary generalized linear model, as described previously. Results of the initial model fit displayed as part of the generated output are not shown here. Statistics for the initial model fit such as parameter estimates, standard errors, deviances, and Pearson chi-squares do not apply to the GEE model and are valid only for the initial model fit. The following figures display information that applies to the GEE model fit.

Figure 37.27 displays general information about the GEE model fit.

Figure 37.27 GEE Model Information

```
                    The GENMOD Procedure

                   GEE Model Information

        Correlation Structure              Exchangeable
        Subject Effect                 case (16 levels)
        Number of Clusters                            16
        Correlation Matrix Dimension                   4
        Maximum Cluster Size                           4
        Minimum Cluster Size                           4
```

Figure 37.28 displays the parameter estimate covariance matrices specified by the COVB option. Both model-based and empirical covariances are produced.

Figure 37.28 GEE Parameter Estimate Covariance Matrices

Covariance Matrix (Model-Based)

	Prm1	Prm2	Prm4	Prm5
Prm1	5.74947	-0.22257	-0.53472	0.01655
Prm2	-0.22257	0.45478	-0.002410	0.01876
Prm4	-0.53472	-0.002410	0.05300	-0.01658
Prm5	0.01655	0.01876	-0.01658	0.19104

Covariance Matrix (Empirical)

	Prm1	Prm2	Prm4	Prm5
Prm1	9.33994	-0.85104	-0.83253	-0.16534
Prm2	-0.85104	0.47368	0.05736	0.04023
Prm4	-0.83253	0.05736	0.07778	-0.002364
Prm5	-0.16534	0.04023	-0.002364	0.13051

The exchangeable working correlation matrix specified by the CORRW option is displayed in Figure 37.29.

Figure 37.29 GEE Working Correlation Matrix

Working Correlation Matrix

	Col1	Col2	Col3	Col4
Row1	1.0000	0.1648	0.1648	0.1648
Row2	0.1648	1.0000	0.1648	0.1648
Row3	0.1648	0.1648	1.0000	0.1648
Row4	0.1648	0.1648	0.1648	1.0000

The parameter estimates table, displayed in Figure 37.30, contains parameter estimates, standard errors, confidence intervals, Z scores, and p-values for the parameter estimates. Empirical standard error estimates are used in this table. A table that displays model-based standard errors can be created by using the REPEATED statement option MODELSE.

Figure 37.30 GEE Parameter Estimates Table

Analysis Of GEE Parameter Estimates
Empirical Standard Error Estimates

Parameter		Estimate	Standard Error	95% Confidence Limits		Z	Pr > |Z|
Intercept		-1.2751	3.0561	-7.2650	4.7148	-0.42	0.6765
city	kingston	-0.1223	0.6882	-1.4713	1.2266	-0.18	0.8589
city	portage	0.0000	0.0000	0.0000	0.0000	.	.
age		0.2036	0.2789	-0.3431	0.7502	0.73	0.4655
smoke		0.0935	0.3613	-0.6145	0.8016	0.26	0.7957

Syntax: GENMOD Procedure

You can specify the following statements in the GENMOD procedure. Items within the <> are optional.

PROC GENMOD < *options* > ;
ASSESS | ASSESSMENT **keyword** / < *options* > ;
BAYES < *options* > ;
BY *variables* ;
CLASS *variables* ;
CONTRAST *'label' effect values* < . . . *effect values* > / < *options* > ;
DEVIANCE *variable = expression* ;
ESTIMATE *'label' effect values* < . . . *effect values* > / < *options* > ;
FREQ | FREQUENCY *variable* ;
FWDLINK *variable = expression* ;
INVLINK *variable = expression* ;
LSMEANS *effects* / < *options* > ;
MODEL *response = < effects > / < options >* ;
OUTPUT < **OUT=***SAS-data-set* >< **keyword=***name* . . . **keyword=***name* > ;
programming statements ;
REPEATED **SUBJECT=***subject-effect* / < *options* > ;
WEIGHT | SCWGT *variable* ;
VARIANCE *variable = expression* ;
ZEROMODEL < *effects* > / < *options* > ;

The PROC GENMOD statement invokes the procedure. All statements other than the MODEL statement are optional. The CLASS statement, if present, must precede the MODEL statement, and the CONTRAST statement must come after the MODEL statement.

PROC GENMOD Statement

PROC GENMOD < *options* > ;

The PROC GENMOD statement invokes the procedure. You can specify the following options.

DATA=*SAS-data-set*
specifies the SAS data set containing the data to be analyzed. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

DESCENDING
DESCEND
DESC
specifies that the levels of the response variable for the ordinal multinomial model and the binomial model with single variable response syntax be sorted in the reverse of the default order. For example, if RORDER=FORMATTED (the default), the DESCENDING option

causes the levels to be sorted from highest to lowest instead of from lowest to highest. If RORDER=FREQ, the DESCENDING option causes the levels to be sorted from lowest frequency count to highest instead of from highest to lowest.

NAMELEN=*n*

specifies the length of effect names in tables and output data sets to be *n* characters long, where *n* is a value between 20 and 200 characters. The default length is 20 characters.

ORDER=*keyword*

specifies the sorting order for the levels of the classification variables (specified in the CLASS statement). This order determines which parameters in the model correspond to each level in the data, so the ORDER= option can be useful when you use the CONTRAST or ESTIMATE statement. Note that the ORDER= option applies to the levels for all classification variables. The exception is the default ORDER=FORMATTED for numeric variables for which you have supplied no explicit format. In this case, the levels are ordered by their internal value. Note that this represents a change from previous releases for how class levels are ordered. Before SAS 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values, and to revert to the previous order you can specify this format explicitly for the affected classification variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and usually required the user to intervene with an explicit format or ORDER=INTERNAL option to get the more natural order. The following table displays the valid *keywords* and describes how PROC GENMOD interprets them.

ORDER=*keyword*	**Levels Sorted by**
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine dependent.

For more information about sorting order, refer to the chapter on the SORT procedure in the *Base SAS Procedures Guide*.

PLOTS <(*global-plot-option***) >= ***plot-request* *< (options) >***

PLOTS <(*global-plot-options***) > <= (***plot-request* *< (options) >* **< ...** *plot-request* *< (options) >* **>) >**

specifies plots to be created using ODS Graphics. Many of the observational statistics in the output data set can be plotted using this option. You are not required to create an output data set in order to produce a plot. When you specify only one plot request, you can omit the parentheses around the plot request. Here are some examples:

```
PLOTS=ALL
PLOTS=PREDICTED
PLOTS=(PREDICTED RESCHI)
PLOTS(UNPACK)=DFBETA
```

You must enable ODS Graphics before requesting plots, for example, like this:

```
ods graphics on;

proc genmod plots=all;
   model y = x;
run;

ods graphics off;
```

Any specified global plot options apply to all plots that are specified with plot requests. The following global plot options are available.

CLUSTERLABEL
displays formatted levels of the SUBJECT= effect instead of plot symbols. This option applies only to diagnostic statistics for models fit by GEEs that are plotted against cluster number, and provides a way to identify cluster level names with corresponding ordered cluster numbers.

UNPACK
displays multiple plots individually. The default is to display related multiple plots in a panel.

See the section "OUTPUT Statement" on page 1954 for definitions of the statistics specified with the plot requests. The plot requests include the following:

ALL
produces all available plots.

COOKSD
DOBS
plots the Cook's distance statistic as a function of observation number.

DFBETA
plots the β deletion statistic as a function of observation number for each regression parameter in the model.

DFBETAS
plots the standardized β deletion statistic as a function of observation number for each regression parameter in the model.

LEVERAGE
plots the leverage as a function of observation number.

PREDICTED<(option)>
plots predicted values with confidence limits as a function of observation number. The PREDICTED plot request has the following option:

CLM
includes confidence limits in the predicted value plot.

RESCHI<(options)>
plots Pearson residuals. The RESCHI plot request has the following options:

INDEX
plots as a function of observation number.

XBETA
plots as a function of linear predictor.

If you do not specify an option, Pearson residuals are plotted as a function of observation number.

RESDEV< (options) >
plots deviance residuals. The RESDEV plot request has the following options:

INDEX
plots as a function of observation number.

XBETA
plots as a function of linear predictor.

If you do not specify an option, deviance residuals are plotted as a function of observation number.

RESLIK< (options) >
plots likelihood residuals. The RESLIK plot request has the following options:

INDEX
plots as a function of observation number.

XBETA
plots as a function of linear predictor.

If you do not specify an option, likelihood residuals are plotted as a function of observation number.

RESRAW< (options) >
plots raw residuals. The RESRAW plot request has the following options:

INDEX
plots as a function of observation number.

XBETA
plots as a function of linear predictor.

If you do not specify an option, raw residuals are plotted as a function of observation number.

STDRESCHI< (options) >
plots standardized Pearson residuals. The STDRESCHI plot request has the following options:

INDEX
plots as a function of observation number.

XBETA
plots as a function of linear predictor.

If you do not specify an option, standardized Pearson residuals are plotted as a function of observation number.

STDRESDEV< (options) >
plots standardized deviance residuals. The STDRESDEV plot request has the following options:

INDEX
plots as a function of observation number.

XBETA
plots as a function of linear predictor.

If you do not specify an option, standardized deviance residuals are plotted as a function of observation number.

If you fit a model by using generalized estimating equations (GEEs), the following additional plot requests are available:

CLEVERAGE
plots the cluster leverage as a function of ordered cluster.

CLUSTERCOOKSD
DCLS
plots the cluster Cook's distance statistic as a function of ordered cluster.

CLUSTERDFIT
MCLS
plots the studentized cluster Cook's distance statistic as a function of ordered cluster.

DFBETAC
plots the cluster deletion statistic as a function of ordered cluster for each regression parameter in the model.

DFBETACS
plots the standardized cluster deletion statistic as a function of ordered cluster for each regression parameter in the model.

RORDER=*keyword*
specifies the sorting order for the levels of the response variable. This order determines which intercept parameter in the model corresponds to each level in the data. If RORDER=FORMATTED for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. Note that this represents a change from previous releases for how class levels are ordered. Before SAS 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values, and to revert to the previous order you can specify this format explicitly for the response variable. The change was implemented because the former default behavior for RORDER=FORMATTED often resulted

in levels not being ordered numerically and usually required the user to intervene with an explicit format or RORDER=INTERNAL to get the more natural ordering. The following table displays the valid *keywords* and describes how PROC GENMOD interprets them.

RORDER=*keyword*	Levels Sorted by
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, RORDER=FORMATTED. For RORDER=FORMATTED and RORDER=INTERNAL, the sort order is machine dependent. The DESCENDING option in the PROC GENMOD statement causes the response variable to be sorted in the reverse of the order displayed in the previous table. For more information about sorting order, refer to the chapter on the SORT procedure in the *Base SAS Procedures Guide*.

The NOPRINT option, which suppresses displayed output in other SAS procedures, is not available in the PROC GENMOD statement. However, you can use the Output Delivery System (ODS) to suppress all displayed output, store all output on disk for further analysis, or create SAS data sets from selected output. You can suppress all displayed output with the statement **`ODS SELECT NONE;`** and turn displayed output back on with the statement **`ODS SELECT ALL;`**. See Table 37.4 and Table 37.5 for the names of output tables available from PROC GENMOD. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

ASSESS Statement

ASSESS VAR=*(effect)* | **LINK** < / *options* > **;**

ASSESSMENT VAR=*(effect)* | **LINK** < / *options* > **;**

The ASSESS statement computes and plots, using ODS Graphics, model-checking statistics based on aggregates of residuals. See the section "Assessment of Models Based on Aggregates of Residuals" on page 1995 for details about the model assessment methods available in GENMOD.

The types of aggregates available are cumulative residuals, moving sums of residuals, and loess smoothed residuals. If you do not specify which aggregate to use, the assessments are based on cumulative sums. PROC GENMOD uses ODS Graphics for graphical displays. For specific information about the graphics available in PROC GENMOD, see the section "ODS Graphics" on page 2022.

You must specify either LINK or VAR= in order to create an analysis.

LINK requests the assessment of the link function by performing the analysis with respect to the linear predictor.

VAR=(*effect***)** specifies that the functional form of a covariate be checked by performing the analysis with respect to the variable identified by the effect. The effect must be specified in the MODEL statement and must contain only continuous variables (variables not listed in a CLASS statement).

You can specify the following options after the slash (/).

CRPANEL
requests that a plot with four panels showing just a few of the paths from the default aggregate plot to make it easier to compare simulated and observed paths. The plot in each panel contains aggregates of the observed residuals and two simulated curves (fewer if NPATHS= is less than 8).

LOESS<(*number***)>**

LOWESS<(*number***)>**
requests model assessment based on loess smoothed residuals with optional *number* the fraction of data used; *number* must be between zero and one. If *number* is not specified, the default value one-third is used.

NPATHS=*number*

NPATH=*number*

PATHS=*number*

PATH=*number*
specifies the number of simulated paths to plot in the default aggregate residuals plot. The default value of *number* is twenty.

RESAMPLE<=*number***>**

RESAMPLES<=*number***>**
specifies that a p-value be computed based on 1,000 simulated paths, or *number* paths, if *number* is specified.

SEED=*number*
specifies a seed for the normal random number generator used in creating simulated realizations of aggregates of residuals for plots and estimating p-values. Specifying a seed enables you to produce identical graphs and p-values from one run of the procedure to the next run. If a seed is not specified, or if *number* is negative or zero, a random number seed is derived from the time of day.

WINDOW<(*number***)>**
requests assessment based on a moving sum window of width *number*. If *number* is not specified, a value of one-half of the range of the x-coordinate is used.

BAYES Statement

BAYES < *options* > **;**

The BAYES statement requests a Bayesian analysis of the regression model by using Gibbs sampling. The Bayesian posterior samples (also known as the chain) for the regression parameters are not tabulated. The Bayesian posterior samples (also known as the chain) for the regression parameters can be output to a SAS data set. Table 37.1 summarizes the options available in the BAYES statement.

Table 37.1 BAYES Statement Options

Option	Description
Monte Carlo Options	
INITIAL=	specifies initial values of the chain
INITIALMLE	specifies that maximum likelihood estimates be used as initial values of the chain
METROPOLIS=	specifies the use of a Metropolis step
NBI=	specifies the number of burn-in iterations
NMC=	specifies the number of iterations after burn-in
SEED=	specifies the random number generator seed
THINNING=	controls the thinning of the Markov chain
Model and Prior Options	
COEFFPRIOR=	specifies the prior of the regression coefficients
DISPERSIONPRIOR=	specifies the prior of the dispersion parameter
PRECISIONPRIOR=	specifies the prior of the precision parameter
SCALEPRIOR=	specifies the prior of the scale parameter
Summary Statistics and Convergence Diagnostics	
DIAGNOSTICS=	displays convergence diagnostics
PLOTS=	displays diagnostic plots
STATISTICS=	displays summary statistics of the posterior samples
Posterior Samples	
OUTPOST=	names a SAS data set for the posterior samples

The following list describes these options and their suboptions.

COEFFPRIOR=JEFFREYS< *(option)* **> | NORMAL<** *(options)* **> | UNIFORM**

COEFF=JEFFREYS< *(options)* **> | NORMAL<** *(options)* **> | UNIFORM**

CPRIOR=JEFFREYS< *(options)* **> | NORMAL<** *(options)* **> | UNIFORM**

specifies the prior distribution for the regression coefficients. The default is COEFFPRIOR=UNIFORM, which specifies the noninformative and improper prior of a constant.

Jeffreys' prior is specified by COEFFPRIOR=JEFFREYS, which can be followed by the following option in parentheses. Jeffreys' prior is proportional to $|I(\boldsymbol{\beta})|^{\frac{1}{2}}$, where $I(\boldsymbol{\beta})$ is the

Fisher information matrix. See the section "Jeffreys' Prior" on page 2005 and Ibrahim and Laud (1991) for more details.

CONDITIONAL
specifies that the Jeffreys' prior, conditional on the current Markov chain value of the generalized linear model precision parameter τ, is proportional to $|\tau \mathbf{I}(\boldsymbol{\beta})|^{\frac{1}{2}}$.

The normal prior is specified by COEFFPRIOR=NORMAL, which can be followed by one of the following options enclosed in parentheses. However, if you do not specify an option, the normal prior $N(\mathbf{0}, 10^6\mathbf{I})$, where $\mathbf{I}$ is the identity matrix, is used. See the section "Normal Prior" on page 2005 for more details.

CONDITIONAL
specifies that the normal prior, conditional on the current Markov chain value of the generalized linear model precision parameter τ, is $N(\boldsymbol{\mu}, \tau^{-1}\boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance of the normal prior specified by other normal options.

INPUT=*SAS-data-set*
specifies a SAS data set containing the mean and covariance information of the normal prior. The data set must have a _TYPE_ variable to represent the type of each observation and a variable for each regression coefficient. If the data set also contains a _NAME_ variable, the values of this variable are used to identify the covariances for the _TYPE_='COV' observations; otherwise, the _TYPE_='COV' observations are assumed to be in the same order as the explanatory variables in the MODEL statement. PROC GENMOD reads the mean vector from the observation with _TYPE_='MEAN' and reads the covariance matrix from observations with _TYPE_='COV'. For an independent normal prior, the variances can be specified with _TYPE_='VAR'; alternatively, the precisions (inverse of the variances) can be specified with _TYPE_='PRECISION'.

RELVAR< *=c* **>**
specifies the normal prior $N(\mathbf{0}, c\mathbf{J})$, where $\mathbf{J}$ is a diagonal matrix with diagonal elements equal to the variances of the corresponding ML estimator. By default, $c = 10^6$.

VAR< *=c* **>**
specifies the normal prior $N(\mathbf{0}, c\mathbf{I})$, where $\mathbf{I}$ is the identity matrix.

DIAGNOSTICS=ALL | NONE | *(keyword-list)*
DIAG=ALL | NONE | *(keyword-list)*
controls the number of diagnostics produced. You can request all the following diagnostics by specifying DIAGNOSTICS=ALL. If you do not want any of these diagnostics, specify DIAGNOSTICS=NONE. If you want some but not all of the diagnostics, or if you want to change certain settings of these diagnostics, specify a subset of the following keywords. The default is DIAGNOSTICS=(AUTOCORR ESS GEWEKE).

AUTOCORR < (LAGS= *numeric-list***) >**
computes the autocorrelations of lags given by LAGS= list for each parameter. Elements in the list are truncated to integers and repeated values are removed. If the LAGS= option is not specified, autocorrelations of lags 1, 5, 10, and 50 are computed for each variable. See the section "Autocorrelations" on page 169 for details.

ESS

computes Carlin's estimate of the effective sample size, the correlation time, and the efficiency of the chain for each parameter. See the section "Effective Sample Size" on page 169 for details.

GELMAN < *(gelman-options)* >

computes the Gelman and Rubin convergence diagnostics. You can specify one or more of the following *gelman-options*:

NCHAIN | **N=***number*

specifies the number of parallel chains used to compute the diagnostic, and must be 2 or larger. The default is NCHAIN=3. If an INITIAL= data set is used, NCHAIN defaults to the number of rows in the INITIAL= data set. If any number other than this is specified with the NCHAIN= option, the NCHAIN= value is ignored.

ALPHA=*value*

specifies the significance level for the upper bound. The default is ALPHA=0.05, resulting in a 97.5% bound.

See the section "Gelman and Rubin Diagnostics" on page 161 for details.

GEWEKE < *(geweke-options)* >

computes the Geweke spectral density diagnostics, which are essentially a two-sample t test between the first f_1 portion and the last f_2 portion of the chain. The default is $f_1 = 0.1$ and $f_2 = 0.5$, but you can choose other fractions by using the following *geweke-options*:

FRAC1=*value*

specifies the fraction f_1 for the first window.

FRAC2=*value*

specifies the fraction f_2 for the second window.

See the section "Geweke Diagnostics" on page 163 for details.

HEIDELBERGER < *(heidel-options)* >

computes the Heidelberger and Welch diagnostic for each variable, which consists of a stationarity test of the null hypothesis that the sample values form a stationary process. If the stationarity test is not rejected, a halfwidth test is then carried out. Optionally, you can specify one or more of the following *heidel-options*:

SALPHA=*value*

specifies the α level ($0 < \alpha < 1$) for the stationarity test.

HALPHA=*value*

specifies the α level ($0 < \alpha < 1$) for the halfwidth test.

EPS=*value*

specifies a positive number ϵ such that if the halfwidth is less than ϵ times the sample mean of the retained iterates, the halfwidth test is passed.

See the section "Heidelberger and Welch Diagnostics" on page 165 for details.

MCERROR

MCSE

computes an estimate of the Monte Carlo standard error for each parameter. See the section "Standard Error of the Mean Estimate" on page 170 for details.

RAFTERY< *(raftery-options)* **>**

computes the Raftery and Lewis diagnostics that evaluate the accuracy of the estimated quantile ($\hat{\theta}_Q$ for a given $Q \in (0,1)$) of a chain. $\hat{\theta}_Q$ can achieve any degree of accuracy when the chain is allowed to run for a long time. A stopping criterion is when the estimated probability $\hat{P}_Q = \Pr(\theta \leq \hat{\theta}_Q)$ reaches within $\pm R$ of the value Q with probability S; that is, $\Pr(Q - R \leq \hat{P}_Q \leq Q + R) = S$. The following *raftery-options* enable you to specify Q, R, S, and a precision level ϵ for the test:

QUANTILE | Q=*value*

specifies the order (a value between 0 and 1) of the quantile of interest. The default is 0.025.

ACCURACY | R=*value*

specifies a small positive number as the margin of error for measuring the accuracy of estimation of the quantile. The default is 0.005.

PROBABILITY | S=*value*

specifies the probability of attaining the accuracy of the estimation of the quantile. The default is 0.95.

EPSILON | EPS=*value*

specifies the tolerance level (a small positive number) for the stationary test. The default is 0.001.

See the section "Raftery and Lewis Diagnostics" on page 166 for details.

DISPERSIONPRIOR=GAMMA< *(options)* **> | IGAMMA<** *(options)* **> | IMPROPER**

DPRIOR=GAMMA< *(options)* **> | IGAMMA<** *(options)* **> | IMPROPER**

specifies that Gibbs sampling be performed on the generalized linear model dispersion parameter and the prior distribution for the dispersion parameter, if there is a dispersion parameter in the model. For models that do not have a dispersion parameter (the Poisson and binomial), this option is ignored. Note that you can specify Gibbs sampling on either the dispersion parameter ϕ, the scale parameter $\sigma = \phi^{\frac{1}{2}}$, or the precision parameter $\tau = \phi^{-1}$, with the DPRIOR=, SPRIOR=, and PPRIOR= options, respectively. These three parameters are transformations of one another, and you should specify Gibbs sampling for only one of them.

A gamma prior $G(a,b)$ with density $f(t) = \frac{b(bt)^{a-1}\mathrm{e}^{-bt}}{\Gamma(a)}$ is specified by DISPERSIONPRIOR=GAMMA, which can be followed by one of the following *gamma-options* enclosed in parentheses. The hyperparameters a and b are the shape and inverse-scale parameters of the gamma distribution, respectively. See the section "Gamma Prior" on page 2004 for details. The default is $G(10^{-4}, 10^{-4})$.

RELSHAPE< =*c* >
specifies independent $G(c\hat{\phi}, c)$ distribution, where $\hat{\phi}$ is the MLE of the dispersion parameter. With this choice of hyperparameters, the mean of the prior distribution is $\hat{\phi}$ and the variance is $\frac{\hat{\phi}}{c}$. By default, *c*=10^{-4}.

SHAPE=*a*
and

ISCALE=*b*
specify the $G(a, b)$ prior.

SHAPE=*c*
specifies the $G(c, c)$ prior.

ISCALE=*c*
specifies the $G(c, c)$ prior.

An inverse gamma prior $IG(a, b)$ with density $f(t) = \frac{b^a}{\Gamma(a)} t^{-(a+1)} \mathrm{e}^{-b/t}$ is specified by DISPERSIONPRIOR=IGAMMA, which can be followed by one of the following *inverse gamma-options* enclosed in parentheses. The hyperparameters a and b are the shape and scale parameters of the inverse gamma distribution, respectively. See the section "Inverse Gamma Prior" on page 2004 for details. The default is $IG(2.001, 0.001)$.

RELSHAPE< =*c* >
specifies independent $IG(\frac{c+\hat{\phi}}{\hat{\phi}}, c)$ distribution, where $\hat{\phi}$ is the MLE of the dispersion parameter. With this choice of hyperparameters, the mean of the prior distribution is $\hat{\phi}$. By default, *c*=10^{-4}.

SHAPE=*a*
and

SCALE=*b*
specify the $IG(a, b)$ prior.

SHAPE=*c*
specifies the $IG(c, c)$ prior.

SCALE=*c*
specifies the $IG(c, c)$ prior.

An improper prior with density $f(t)$ proportional to t^{-1} is specified with DISPERSIONPRIOR=IMPROPER.

INITIAL=*SAS-data-set*
specifies the SAS data set that contains the initial values of the Markov chains. The INITIAL= data set must contain all the variables of the model. You can specify multiple rows as the initial values of the parallel chains for the Gelman-Rubin statistics, but posterior summaries, diagnostics, and plots are computed only for the first chain. If the data set also contains the variable _SEED_, the value of the _SEED_ variable is used as the seed of the random number generator for the corresponding chain.

INITIALMLE

specifies that maximum likelihood estimates of the model parameters be used as initial values of the Markov chain. If this option is not specified, estimates of the mode of the posterior distribution obtained by optimization are used as initial values.

METROPOLIS=YES

METROPOLIS=NO

specifies the use of a Metropolis step to generate Gibbs samples for posterior distributions that are not log concave. The default value is METROPOLIS=YES.

NBI=*number*

specifies the number of burn-in iterations before the chains are saved. The default is 2000.

NMC=*number*

specifies the number of iterations after the burn-in. The default is 10000.

OUTPOST=*SAS-data-set*

OUT=*SAS-data-set*

names the SAS data set that contains the posterior samples. See the section "OUTPOST= Output Data Set" on page 2007 for more information. Alternatively, you can create the output data set by specifying an ODS OUTPUT statement as follows:

ODS output posteriorsample = *SAS-data-set* **;**

PRECISIONPRIOR=GAMMA< *(options)* **>** | **IMPROPER**

PPRIOR=GAMMA< *(options)* **>** | **IMPROPER**

specifies that Gibbs sampling be performed on the generalized linear model precision parameter and the prior distribution for the precision parameter, if there is a precision parameter in the model. For models that do not have a precision parameter (the Poisson and binomial), this option is ignored. Note that you can specify Gibbs sampling on either the dispersion parameter ϕ, the scale parameter $\sigma = \phi^{\frac{1}{2}}$, or the precision parameter $\tau = \phi^{-1}$, with the DPRIOR=, SPRIOR=, and PPRIOR= options, respectively. These three parameters are transformations of one another, and you should specify Gibbs sampling for only one of them.

A gamma prior $G(a, b)$ with density $f(t) = \frac{b(bt)^{a-1}\mathrm{e}^{-bt}}{\Gamma(a)}$ is specified by PRECISIONPRIOR=GAMMA, which can be followed by one of the following *gamma-options* enclosed in parentheses. The hyperparameters a and b are the shape and inverse-scale parameters of the gamma distribution, respectively. See the section "Gamma Prior" on page 2004 for details. The default is $G(10^{-4}, 10^{-4})$.

RELSHAPE< =*c* **>**

specifies independent $G(c\hat{\tau}, c)$ distribution, where $\hat{\tau}$ is the MLE of the dispersion parameter. With this choice of hyperparameters, the mean of the prior distribution is $\hat{\tau}$ and the variance is $\frac{\hat{\tau}}{c}$. By default, $c = 10^{-4}$.

SHAPE=*a*

and

ISCALE=*b*

specify the $G(a, b)$ prior.

SHAPE=*c*
specifies the $G(c, c)$ prior.

ISCALE=*c*
specifies the $G(c, c)$ prior.

An improper prior with density $f(t)$ proportional to t^{-1} is specified with PRECISION-PRIOR=IMPROPER.

SCALEPRIOR=GAMMA< *(options)* **>** | **IMPROPER**

SPRIOR=GAMMA< *(options)* **>** | **IMPROPER**
specifies that Gibbs sampling be performed on the generalized linear model scale parameter and the prior distribution for the scale parameter, if there is a scale parameter in the model. For models that do not have a scale parameter (the Poisson and binomial), this option is ignored. Note that you can specify Gibbs sampling on either the dispersion parameter ϕ, the scale parameter $\sigma = \phi^{\frac{1}{2}}$, or the precision parameter $\tau = \phi^{-1}$, with the DPRIOR=, SPRIOR=, and PPRIOR= options, respectively. These three parameters are transformations of one another, and you should specify Gibbs sampling for only one of them.

A gamma prior $G(a, b)$ with density $f(t) = \frac{b(bt)^{a-1}\mathrm{e}^{-bt}}{\Gamma(a)}$ is specified by SCALEPRIOR=GAMMA, which can be followed by one of the following *gamma-options* enclosed in parentheses. The hyperparameters a and b are the shape and inverse-scale parameters of the gamma distribution, respectively. See the section "Gamma Prior" on page 2004 for details. The default is $G(10^{-4}, 10^{-4})$.

RELSHAPE< =*c* **>**
specifies independent $G(c\hat{\sigma}, c)$ distribution, where $\hat{\sigma}$ is the MLE of the dispersion parameter. With this choice of hyperparameters, the mean of the prior distribution is $\hat{\sigma}$ and the variance is $\frac{\hat{\sigma}}{c}$. By default, $c = 10^{-4}$.

SHAPE=*a*
and

ISCALE=*b*
specify the $G(a, b)$ prior.

SHAPE=*c*
specifies the $G(c, c)$ prior.

ISCALE=*c*
specifies the $G(c, c)$ prior.

An improper prior with density $f(t)$ proportional to t^{-1} is specified with SCALEPRIOR=IMPROPER.

PLOTS< *(global-plot-options)* **>=** *plot-request*

PLOTS< *(global-plot-options)* **>=** *(plot-request < . . . plot-request>)*
controls the display of diagnostic plots. Three types of plots can be requested: trace plots, autocorrelation function plots, and kernel density plots. By default, the plots are displayed in panels unless the global plot option UNPACK is specified. Also, when you are specifying more than one type of plots, the plots are displayed by parameters unless the global plot

option GROUPBY is specified. When you specify only one plot request, you can omit the parentheses around the plot request. For example:

```
plots=none
plots(unpack)=trace
plots=(trace autocorr)
```

You must enable ODS Graphics before requesting plots. For example, the following SAS statements enable ODS Graphics:

```
ods graphics on;
proc genmod;
   model y=x;
   bayes plots=trace;
   run;
end;
ods graphics off;
```

The global plot options are as follows:

FRINGE
creates a fringe plot on the X axis of the density plot.

GROUPBY=PARAMETER

GROUPBY=TYPE
specifies how the plots are grouped when there is more than one type of plot.

GROUPBY=TYPE
specifies that the plots be grouped by type.

GROUPBY=PARAMETER
specifies that the plots be grouped by parameter.

GROUPBY=PARAMETER is the default.

LAGS=*n*
specifies that autocorrelations be plotted up to lag *n*. If this option is not specified, autocorrelations are plotted up to lag 50.

SMOOTH
displays a fitted penalized B-spline curve for each trace plot.

UNPACKPANEL

UNPACK
specifies that all paneled plots be unpacked, meaning that each plot in a panel is displayed separately.

The plot requests include the following:

ALL
specifies all types of plots. PLOTS=ALL is equivalent to specifying PLOTS=(TRACE AUTOCORR DENSITY).

AUTOCORR
displays the autocorrelation function plots for the parameters.

DENSITY
displays the kernel density plots for the parameters.

NONE
suppresses all diagnostic plots.

TRACE
displays the trace plots for the parameters. See the section "Visual Analysis via Trace Plots" on page 156 for details.

SEED=*number*
specifies an integer seed in the range 1 to $2^{31} - 1$ for the random number generator in the simulation. Specifying a seed enables you to reproduce identical Markov chains for the same specification. If the SEED= option is not specified, or if you specify a nonpositive seed, a random seed is derived from the time of day.

STATISTICS < *(global-options)* > = **ALL** | **NONE** | *keyword* | *(keyword-list)*
STATS < *(global-options)* > = **ALL** | **NONE** | *keyword* | *(keyword-list)*
controls the number of posterior statistics produced. Specifying STATISTICS=ALL is equivalent to specifying STATISTICS= (SUMMARY INTERVAL COV CORR). If you do not want any posterior statistics, you specify STATISTICS=NONE. The default is STATISTICS=(SUMMARY INTERVAL). See the section "Summary Statistics" on page 170 for details. The *global-options* include the following:

ALPHA=*numeric-list*
controls the probabilities of the credible intervals. The ALPHA= values must be between 0 and 1. Each ALPHA= value produces a pair of 100(1–ALPHA)% equal-tail and HPD intervals for each parameters. The default is the value of the ALPHA= option in the MODEL statement, or 0.05 if that option is not specified (yielding the 95% credible intervals for each parameter).

PERCENT=*numeric-list*
requests the percentile points of the posterior samples. The PERCENT= values must be between 0 and 100. The default is PERCENT=25, 50, 75, which yield the 25th, 50th, and 75th percentile points, respectively, for each parameter.

The list of *keywords* includes the following:

CORR
produces the posterior correlation matrix.

COV
produces the posterior covariance matrix.

SUMMARY
produces the means, standard deviations, and percentile points for the posterior samples. The default is to produce the 25th, 50th, and 75th percentile points, but you can use the global PERCENT= option to request specific percentile points.

INTERVAL

produces equal-tail credible intervals and HPD intervals. The defult is to produce the 95% equal-tail credible intervals and 95% HPD intervals, but you can use the global ALPHA= option to request intervals of any probabilities.

THINNING=*number*

THIN=*number*

controls the thinning of the Markov chain. Only one in every k samples is used when THINNING=k, and if NBI=n_0 and NMC=n, the number of samples kept is

$$\left[\frac{n_0+n}{k}\right]-\left[\frac{n_0}{k}\right]$$

where $[a]$ represents the integer part of the number a. The default is THINNING=1.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC GENMOD to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

Since sorting the data changes the order in which PROC GENMOD reads the data, this can affect the sorting order for the levels of classification variables if you have specified ORDER=DATA in the PROC GENMOD statement. This in turn affects specifications in the CONTRAST statement.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the GENMOD procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

The CLASS statement names the classification variables to be used in the analysis. The CLASS statement must precede the MODEL statement. You can specify various *v-options* for each variable

by enclosing them in parentheses after the variable name. You can also specify global *v-options* for the CLASS statement by placing them after a slash (/). Global *v-options* are applied to all the variables specified in the CLASS statement. If you specify more than one CLASS statement, the global *v-options* specified in any one CLASS statement apply to all CLASS statements. However, individual CLASS variable *v-options* override the global *v-options*.

DESCENDING

DESC

reverses the sorting order of the classification variable.

MISSING

allows missing value ('.' for a numeric variable and blank for a character variable) as a valid value for the CLASS variable.

ORDER=DATA

ORDER=FORMATTED

ORDER=FREQ

ORDER=INTERNAL

specifies the sorting order for the levels of classification variables. This order determines which parameters in the model correspond to each level in the data, so the ORDER= option can be useful when you use the CONTRAST or ESTIMATE statement. If ORDER=FORMATTED for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. Note that this represents a change from previous releases for how class levels are ordered. Before SAS 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values, and to revert to the previous ordering you can specify this format explicitly for the affected classification variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and usually required the user to intervene with an explicit format or ORDER=INTERNAL to get the more natural order. The following table shows how PROC GENMOD interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted by
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent.

For more information about sorting order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

PARAM=*keyword*

specifies the parameterization method for the classification variable or variables. Design matrix columns are created from CLASS variables according to the following coding schemes. The default is PARAM=GLM. If PARAM=ORTHPOLY or PARAM=POLY, and the CLASS levels are numeric, then the ORDER= option in the CLASS statement is ignored, and the internal, unformatted values are used. See the section "CLASS Variable Parameterization" on page 1975 for further details. You can use the following *keywords*:

EFFECT specifies effect coding.

GLM specifies less-than-full-rank, reference-cell coding; this option can be used only as a global option.

ORDINAL

THERMOMETER specifies the cumulative parameterization for an ordinal CLASS variable.

POLYNOMIAL

POLY specifies polynomial coding.

REFERENCE

REF specifies reference-cell coding.

ORTHEFFECT orthogonalizes PARAM=EFFECT.

ORTHORDINAL

ORTHOTHERM orthogonalizes PARAM=ORDINAL.

ORTHPOLY orthogonalizes PARAM=POLYNOMIAL.

ORTHREF orthogonalizes PARAM=REFERENCE.

The EFFECT, POLYNOMIAL, REFERENCE, and ORDINAL suboptions and their orthogonal parameterizations are full rank. The REF= option in the CLASS statement determines the reference level for the EFFECT and REFERENCE suboptions and their orthogonal parameterizations.

REF=*'level' | keyword*

specifies the reference level for PARAM=EFFECT, PARAM=REFERENCE, and their orthogonalizations. For an individual (but not a global) variable REF= *option*, you can specify the *level* of the variable to use as the reference level. For a global or individual variable REF= *option*, you can use one of the following *keywords*. The default is REF=LAST.

FIRST designates the first ordered level as reference.

LAST designates the last ordered level as reference.

TRUNCATE< *=n* >

specifies the length *n* of CLASS variable values to use in determining CLASS variable levels. If you specify TRUNCATE without the length *n*, the first 16 characters of the formatted values are used. When formatted values are longer than 16 characters, you can use this option to revert to the levels as determined in releases before SAS 9. The default is to use the full formatted length of the CLASS variable. The TRUNCATE option is available only as a global option.

CONTRAST Statement

CONTRAST *'label' contrast-specification* / < *options* > **;**

The CONTRAST statement provides a means of obtaining a test of a specified hypothesis concerning the model parameters. This is accomplished by specifying a matrix $\mathbf{L}$ for testing the hypothesis $\mathbf{L}'\boldsymbol{\beta} = 0$. You must be familiar with the details of the model parameterization that PROC GENMOD uses. For more information, see the section "Parameterization Used in PROC GENMOD" on page 1975 and the section "CLASS Variable Parameterization" on page 1975. Computed statistics are based on the asymptotic chi-square distribution of the likelihood ratio statistic, or the generalized score statistic for GEE models, with degrees of freedom determined by the number of linearly independent rows in the $\mathbf{L}'$ matrix. You can request Wald chi-square statistics with the Wald option in the CONTRAST statement.

There is no limit to the number of CONTRAST statements that you can specify, but they must appear after the MODEL statement and after the ZEROMODEL statement for zero-inflated Poisson models. Statistics for multiple CONTRAST statements are displayed in a single table.

The elements of the CONTRAST statement are as follows:

label identifies the contrast on the output. A label is required for every contrast specified. Labels can be up to 20 characters and must be enclosed in single quotes.

contrast-specification identifies the effects and their coefficients from which the $\mathbf{L}$ matrix is formed. The *contrast-specification* can be specified in two different ways. The first method applies to all models except the zero-inflated Poisson (ZIP) distribution, and the syntax is:

effect values <,... *effect values*>

The second method of specifying a contrast applies only to ZIP models, and the syntax is:

effect values <,... *effect values*> @*zero effect values* <,... *effect values*>

Specification of sets of *effect values* before the @*zero* separator results in a row of the $\mathbf{L}'$ matrix with coefficients for *effects* in the regression part of the model set to *values* and with the coefficients for the zero-inflation part of the model set to zero. Specification of sets of *effect values* after the @*zero* separator results in a row of the $\mathbf{L}$ matrix with the coefficients for the regression part of the model set to zero and with the coefficients of *effects* in the zero-inflation part of the model set to *values*.

For example, the statements

```
CLASS A;
MODEL y=A;
CONTRAST 'Label1' A 1 -1;
```

specify an $\mathbf{L}'$ matrix with one row with coefficients 1 for the first level of A and –1 for the second level of A.

The statements

```
CLASS A B;
MODEL y=A / Dist=ZIP;
ZEROMODEL B;
CONTRAST 'Label2' A 1 -1 @ZERO B 1 -1;
```

specify an $\mathbf{L}'$ matrix with two rows: the first row has coefficients 1 for the first level of A, –1 for the second level of A, and zeros for all levels of B; the second row has coefficients 0 for all levels of A, 1 for the first level of B, and –1 for the second level of B.

effect identifies an effect that appears in the MODEL statement. The value INTERCEPT or intercept can be used as an effect when an intercept is included in the model. You do not need to include all effects that are included in the MODEL statement.

values are constants that are elements of the $\mathbf{L}$ vector associated with the effect.

The rows of $\mathbf{L}'$ are specified in order and are separated by commas.

If you use the default less-than-full-rank PROC GLM CLASS variable parameterization, each row of the $\mathbf{L}'$ matrix is checked for estimability. If PROC GENMOD finds a contrast to be nonestimable, it displays missing values in corresponding rows in the results. See Searle (1971) for a discussion of estimable functions. If the elements of $\mathbf{L}'$ are not specified for an effect that contains a specified effect, then the elements of the specified effect are distributed over the levels of the higher-order effect just as the GLM procedure does for its CONTRAST and ESTIMATE statements. For example, suppose that the model contains effects A and B and their interaction A*B. If you specify a CONTRAST statement involving A alone, the $\mathbf{L}'$ matrix contains nonzero terms for both A and A*B, since A*B contains A.

When you use any of the full-rank PARAM= CLASS variable options, all parameters are directly estimable, and rows of $\mathbf{L}'$ are not checked for estimability.

If an effect is not specified in the CONTRAST statement, all of its coefficients in the $\mathbf{L}'$ matrix are set to 0. If too many values are specified for an effect, the extra ones are ignored. If too few values are specified, the remaining ones are set to 0.

PROC GENMOD handles missing level combinations of classification variables in the same manner as the GLM and MIXED procedures. Parameters corresponding to missing level combinations are not included in the model. This convention can affect the way in which you specify the $\mathbf{L}$ matrix in your CONTRAST statement.

If you specify the WALD option, the test of hypothesis is based on a Wald chi-square statistic. If you omit the WALD option, the test statistic computed depends on whether an ordinary generalized linear model or a GEE-type model is specified.

For an ordinary generalized linear model, the CONTRAST statement computes the likelihood ratio statistic. This is defined to be twice the difference between the log likelihood of the model unconstrained by the contrast and the log likelihood with the model fitted under the constraint that the linear function of the parameters defined by the contrast is equal to 0. A p-value is computed based on the asymptotic chi-square distribution of the chi-square statistic.

If you specify a GEE model with the REPEATED statement, the test is based on a score statistic. The GEE model is fit under the constraint that the linear function of the parameters defined by the contrast is equal to 0. The score chi-square statistic is computed based on the generalized score function. See the section "Generalized Score Statistics" on page 1995 for more information.

The degrees of freedom is the number of linearly independent constraints implied by the CONTRAST statement—that is, the rank of $\mathbf{L}$.

You can specify the following options after a slash (/).

E

requests that the **L** matrix be displayed.

SINGULAR=*number*

EPSILON=*number*

tunes the estimability checking. If **v** is a vector, define ABS(**v**) to be the absolute value of the element of **v** with the largest absolute value. Let $\mathbf{K}'$ be any row in the contrast matrix **L**. Define C to be equal to ABS($\mathbf{K}'$) if ABS($\mathbf{K}'$) is greater than 0; otherwise, C equals 1. If ABS($\mathbf{K}' - \mathbf{K}'\mathbf{T}$) is greater than C∗*number*, then **K** is declared nonestimable. **T** is the Hermite form matrix $(\mathbf{X}'\mathbf{X})^{-}(\mathbf{X}'\mathbf{X})$, and $(\mathbf{X}'\mathbf{X})^{-}$ represents a generalized inverse of the matrix $\mathbf{X}'\mathbf{X}$. The value for $number$ must be between 0 and 1; the default value is 1E−4. The SINGULAR= option in the MODEL statement affects the computation of the generalized inverse of the matrix $\mathbf{X}'\mathbf{X}$. It might also be necessary to adjust this value for some data.

WALD

requests that a Wald chi-square statistic be computed for the contrast rather than the default likelihood ratio or score statistic. The Wald statistic for testing $\mathbf{L}'\boldsymbol{\beta} = \mathbf{0}$ is defined by

$$S = (\mathbf{L}'\hat{\boldsymbol{\beta}})'(\mathbf{L}'\boldsymbol{\Sigma}\mathbf{L})^{-}(\mathbf{L}'\hat{\boldsymbol{\beta}})$$

where $\hat{\boldsymbol{\beta}}$ is the maximum likelihood estimate and $\boldsymbol{\Sigma}$ is its estimated covariance matrix. The asymptotic distribution of S is χ^2_r, where r is the rank of **L**. Computed p-values are based on this distribution.

If you specify a GEE model with the REPEATED statement, $\boldsymbol{\Sigma}$ is the empirical covariance matrix estimate.

DEVIANCE Statement

DEVIANCE *variable = expression* **;**

You can specify a probability distribution other than those available in PROC GENMOD by using the DEVIANCE and VARIANCE statements. You do not need to specify the DEVIANCE or VARIANCE statement if you use the DIST= MODEL statement option to specify a probability distribution. The *variable* identifies the deviance contribution from a single observation to the procedure, and it must be a valid SAS variable name that does not appear in the input data set. The *expression* can be any arithmetic expression supported by the DATA step language, and it is used to define the functional dependence of the deviance on the mean and the response. You use the automatic variables _MEAN_ and _RESP_ to represent the mean and response in the *expression*.

Alternatively, the deviance function can be defined using programming statements (see the section "Programming Statements" on page 1957) and assigned to a variable, which is then listed as the *expression*. This form is convenient for using complex statements such as IF-THEN/ELSE clauses.

The DEVIANCE statement is ignored unless the VARIANCE statement is also specified.

ESTIMATE Statement

ESTIMATE *'label' effect values <,... effect values> </options>* **;**

The ESTIMATE statement is similar to a CONTRAST statement, except only one-row $\mathbf{L}'$ matrices are permitted.

In the case of zero-inflated Poisson (ZIP) models, the statement syntax is:

ESTIMATE *'label' effect values <,... effect values> @zero effect values <,... effect values> </options>* **;**

where sets of *effects values* before the *@zero* separator correspond to the regression part of the model, and *effects values* after the *@zero* separator correspond to the zero-inflation part of the model. In the case of ZIP models, a one-row $\mathbf{L}'$ matrix is created for the regression part of the model, another one-row $\mathbf{L}'$ matrix is created for the zero-inflation part of the model, and separate estimate estimates for the two $\mathbf{L}$ matrices are computed and displayed.

If you use the default less-than-full-rank GLM CLASS variable parameterization, each row is checked for estimability. If PROC GENMOD finds a contrast to be nonestimable, it displays missing values in corresponding rows in the results. See Searle (1971) for a discussion of estimable functions.

The actual estimate, $\mathbf{L}'\boldsymbol{\gamma}$, its approximate standard error, and its confidence limits are displayed. A Wald chi-square test that $\mathbf{L}'\boldsymbol{\beta} = 0$ is also displayed.

The approximate standard error of the estimate is computed as the square root of $\mathbf{L}'\hat{\boldsymbol{\Sigma}}\mathbf{L}$, where $\hat{\boldsymbol{\Sigma}}$ is the estimated covariance matrix of the parameter estimates. If you specify a GEE model in the REPEATED statement, $\hat{\boldsymbol{\Sigma}}$ is the empirical covariance matrix estimate.

If you specify the EXP option, then $\exp(\mathbf{L}'\boldsymbol{\beta})$, its standard error, and its confidence limits are also displayed.

The construction of the $\mathbf{L}$ vector and the checking for estimability for an ESTIMATE statement follow the same rules as listed under the CONTRAST statement.

You can specify the following options in the ESTIMATE statement after a slash (/).

ALPHA=*number*
: requests that a confidence interval be constructed with confidence level $1 - number$. The value of *number* must be between 0 and 1; the default value is 0.05.

E
: requests that the $\mathbf{L}$ matrix coefficients be displayed.

EXP
: requests that $\exp(\mathbf{L}'\boldsymbol{\beta})$, its standard error, and its confidence limits be computed. If you specify the EXP option, standard errors and confidence intervals are computed using the delta method.

SINGULAR=*number*

EPSILON=*number*

tunes the estimability checking as described for the CONTRAST statement.

FREQ Statement

FREQ *variable* **;**

FREQUENCY *variable* **;**

The *variable* in the FREQ statement identifies a variable in the input data set containing the frequency of occurrence of each observation. PROC GENMOD treats each observation as if it appears *n* times, where *n* is the value of the FREQ variable for the observation. If it is not an integer, the frequency value is truncated to an integer. If it is less than 1 or missing, the observation is not used. In the case of models fit with generalized estimating equations (GEEs), the frequencies apply to the subject/cluster and therefore must be the same for all observations within each subject.

FWDLINK Statement

FWDLINK *variable = expression* **;**

You can define a link function other than a built-in link function by using the FWDLINK statement. If you use the MODEL statement option LINK= to specify a link function, you do not need to use the FWDLINK statement. The *variable* identifies the link function to the procedure. The *expression* can be any arithmetic expression supported by the DATA step language, and it is used to define the functional dependence on the mean.

Alternatively, the link function can be defined by using programming statements (see the section "Programming Statements" on page 1957) and assigned to a variable, which is then listed as the *expression*. The second form is convenient for using complex statements such as IF-THEN/ELSE clauses. The GENMOD procedure automatically computes derivatives of the link function required for iterative fitting. You must specify the inverse of the link function in the INVLINK statement when you specify the FWDLINK statement to define the link function. You use the automatic variable _MEAN_ to represent the mean in the preceding *expression*.

INVLINK Statement

INVLINK *variable = expression* **;**

If you define a link function in the FWDLINK statement, then you must define the inverse link function by using the INVLINK statement. If you use the MODEL statement option LINK= to specify a link function, you do not need to use the INVLINK statement. The *variable* identifies the

inverse link function to the procedure. The *expression* can be any arithmetic expression supported by the DATA step language, and it is used to define the functional dependence on the linear predictor.

Alternatively, the inverse link function can be defined using programming statements (see the section "Programming Statements" on page 1957) and assigned to a variable, which is then listed as the *expression*. The second form is convenient for using complex statements such as IF-THEN/ELSE clauses. The automatic variable _XBETA_ represents the linear predictor in the preceding *expression*.

LSMEANS Statement

LSMEANS *effects* < / *options* > **;**

The LSMEANS statement computes least squares means (LS-means) corresponding to the specified effects for the linear predictor part of the model. The **L** matrix constructed to compute them is precisely the same as the one formed in PROC GLM.

The LSMEANS statement is not available for multinomial distribution models for ordinal response data.

Each LS-mean is computed as $\mathbf{L}'\hat{\boldsymbol{\beta}}$, where $\mathbf{L}$ is the coefficient matrix associated with the least squares mean and $\hat{\boldsymbol{\beta}}$ is the estimate of the parameter vector. The approximate standard errors for the LS-mean is computed as the square root of $\mathbf{L}'\hat{\boldsymbol{\Sigma}}\mathbf{L}$, where $\hat{\boldsymbol{\Sigma}}$ is the estimated covariance matrix of the parameter estimates. If you specify a GEE model in the REPEATED statement, $\hat{\boldsymbol{\Sigma}}$ is the empirical covariance matrix estimate.

LS-means can be computed for any effect in the MODEL statement that involves CLASS variables. You can specify multiple effects in one LSMEANS statement or multiple LSMEANS statements, and all LSMEANS statements must appear after the MODEL statement.

As in the ESTIMATE statement, the **L** matrix is tested for estimability, and if this test fails, PROC GENMOD displays "Non-est" for the LS-means entries.

Assuming the LS-mean is estimable, PROC GENMOD constructs a Wald chi-square test to test the null hypothesis that the associated population quantity equals zero.

You can specify the following options in the LSMEANS statement after a slash (/).

ALPHA=*number*
: requests that a confidence interval be constructed for each of the LS-means with confidence level $(1 - number) \times 100\%$. The value of *number* must be between 0 and 1; the default value is 0.05, corresponding to a 95% confidence interval.

CL
: requests that confidence limits be constructed for each of the LS-means. The confidence level is 0.95 by default; this can be changed with the ALPHA= option.

CORR
: displays the estimated correlation matrix of the LS-means as part of the "Least Squares Means" table.

COV

displays the estimated covariance matrix of the LS-means as part of the "Least Squares Means" table.

DIFF

requests that differences of the LS-means be displayed. All possible differences of LS-means, standard errors, and a Wald chi-square test are computed. Confidence limits are computed if the CL option is also specified.

E

requests that the **L** matrix coefficients for all LSMEANS effects be displayed.

SINGULAR=*number*

EPSILON=*number*

tunes the estimability checking as described for the CONTRAST statement.

MODEL Statement

MODEL *response* = < *effects* > < */options* > **;**

MODEL *events/trials* = < *effects* > < */options* > **;**

The MODEL statement specifies the response, or dependent variable, and the effects, or explanatory variables. If you omit the explanatory variables, the procedure fits an intercept-only model. An intercept term is included in the model by default. The intercept can be removed with the NOINT option.

You can specify the response in the form of a single variable or in the form of a ratio of two variables denoted *events/trials*. The first form is applicable to all responses. The second form is applicable only to summarized binomial response data. When each observation in the input data set contains the number of events (for example, successes) and the number of trials from a set of binomial trials, use the *events/trials* syntax.

In the *events/trials* model syntax, you specify two variables that contain the event and trial counts. These two variables are separated by a slash (/). The values of both *events* and (*trials*−*events*) must be nonnegative, and the value of the *trials* variable must be greater than 0 for an observation to be valid. The variable *events* or *trials* can take noninteger values.

When each observation in the input data set contains a single trial from a binomial or multinomial experiment, use the first form of the preceding MODEL statements. The response variable can be numeric or character. The ordering of response levels is critical in these models. You can use the RORDER= option in the PROC GENMOD statement to specify the response level ordering.

Responses for the Poisson distribution must be all nonnegative, but they can be noninteger values.

The effects in the MODEL statement consist of an explanatory variable or combination of variables. Explanatory variables can be continuous or classification variables. Classification variables can be character or numeric. Explanatory variables representing nominal, or classification, data must be declared in a CLASS statement. Interactions between variables can also be included as effects.

Columns of the design matrix are automatically generated for classification variables and interactions. The syntax for specification of effects is the same as for the GLM procedure. See the section "Specification of Effects" on page 1974 for more information. Also refer to Chapter 39, "The GLM Procedure."

You can specify the following options in the MODEL statement after a slash (/).

AGGREGATE= *(variable-list)*

AGGREGATE= *variable*

AGGREGATE

specifies the subpopulations on which the Pearson chi-square and the deviance are calculated. This option applies only to the multinomial distribution or the binomial distribution with binary (single trial syntax) response. It is ignored if specified for other cases. Observations with common values in the given list of variables are regarded as coming from the same subpopulation. This affects the computation of the deviance and Pearson chi-square statistics. Variables in the list can be any variables in the input data set. Specifying the AGGREGATE option is equivalent to specifying the AGGREGATE= option with a variable list that includes all explanatory variables in the MODEL statement. Pearson chi-square and deviance statistics are not computed for multinomial models unless this option is specified.

ALPHA=*number*

ALPH=*number*

A=*number*

sets the confidence coefficient for parameter confidence intervals to 1−*number*. The value of *number* must be between 0 and 1. The default value of *number* is 0.05.

CICONV=*number*

sets the convergence criterion for profile likelihood confidence intervals. See the section "Confidence Intervals for Parameters" on page 1980 for the definition of convergence. The value of *number* must be between 0 and 1. By default, CICONV=1E−4.

CL

requests that confidence limits for predicted values be displayed (see the OBSTATS option).

CODING=EFFECT

CODING=FULLRANK

specifies that effect coding be used for all classification variables in the model. This is the same as specifying PARAM=EFFECT as a CLASS statement option.

CONVERGE=*number*

sets the convergence criterion. The value of *number* must be between 0 and 1. The iterations are considered to have converged when the maximum change in the parameter estimates between iteration steps is less than the value specified. The change is a relative change if the parameter is greater than 0.01 in absolute value; otherwise, it is an absolute change. By default, CONVERGE=1E−4. This convergence criterion is used in parameter estimation for a single model fit, Type 1 statistics, and likelihood ratio statistics for Type 3 analyses and CONTRAST statements.

CONVH=*number*

sets the relative Hessian convergence criterion. The value of *number* must be between 0 and 1. After convergence is determined with the change in parameter criterion specified with the CONVERGE= option, the quantity $tc = \frac{\mathbf{g}'\mathbf{H}^{-1}\mathbf{g}}{|f|}$ is computed and compared to *number*, where $\mathbf{g}$ is the gradient vector, $\mathbf{H}$ is the Hessian matrix for the model parameters, and f is the log-likelihood function. If tc is greater than *number*, a warning that the relative Hessian convergence criterion has been exceeded is printed. This criterion detects the occasional case where the change in parameter convergence criterion is satisfied, but a maximum in the log-likelihood function has not been attained. By default, CONVH=1E−4.

CORRB

requests that the parameter estimate correlation matrix be displayed.

COVB

requests that the parameter estimate covariance matrix be displayed.

DIAGNOSTICS

INFLUENCE

requests that case deletion diagnostic statistics be displayed (see the OBSTATS option).

DIST=*keyword*

D=*keyword*

ERROR=*keyword*

ERR=*keyword*

specifies the built-in probability distribution to use in the model. If you specify the DIST= option and you omit a user-defined link function, a default link function is chosen as displayed in the following table. If you specify no distribution and no link function, then the GENMOD procedure defaults to the normal distribution with the identity link function.

DIST=	Distribution	Default Link Function
BINOMIAL | BIN | B	binomial	logit
GAMMA | GAM | G	gamma	inverse (power(−1))
GEOMETRIC | GEOM	geometric	log
IGAUSSIAN | IG	inverse Gaussian	inverse squared (power(−2))
MULTINOMIAL | MULT	multinomial	cumulative logit
NEGBIN | NB	negative binomial	log
NORMAL | NOR | N	normal	identity
POISSON | POI | P	Poisson	log
ZIP		

EXPECTED

requests that the expected Fisher information matrix be used to compute parameter estimate covariances and the associated statistics. The default action is to use the observed Fisher information matrix. This option does not affect the model fitting, only the way in which the covariance matrix is computed (see the SCORING= option.)

ID=*variable*

causes the values of *variable* in the input data set to be displayed in the OBSTATS table. If

an explicit format for *variable* has been defined, the formatted values are displayed. If the OBSTATS option is not specified, this option has no effect.

INITIAL=*numbers*

sets initial values for parameter estimates in the model. The default initial parameter values are weighted least squares estimates based on using the response data as the initial mean estimate. This option can be useful in case of convergence difficulty. The intercept parameter is initialized with the INTERCEPT= option and is not included here. The values are assigned to the variables in the MODEL statement in the same order in which they appear in the MODEL statement. The order of levels for CLASS variables is determined by the ORDER= option. Note that some levels of classification variables can be aliased; that is, they correspond to linearly dependent parameters that are not estimated by the procedure. Initial values must be assigned to all levels of classification variables, regardless of whether they are aliased or not. The procedure ignores initial values corresponding to parameters not being estimated. If you specify a BY statement, all classification variables must take on the same number of levels in each BY group. Otherwise, classification variables in some of the BY groups are assigned incorrect initial values. Types of INITIAL= specifications are illustrated in the following table.

Type of List	Specification
list separated by blanks	INITIAL = 3 4 5
list separated by commas	INITIAL = 3, 4, 5
x to y	INITIAL = 3 to 5
x to y by z	INITIAL = 3 to 5 by 1
combination of list types	INITIAL = 1, 3 to 5, 9

INTERCEPT=*number*

INTERCEPT=*number-list*

initializes the intercept term to *number* for parameter estimation. If you specify both the INTERCEPT= and the NOINT options, the intercept term is not estimated, but an intercept term of *number* is included in the model. If you specify a multinomial model for ordinal data, you can specify a *number-list* for the multiple intercepts in the model.

ITPRINT

displays the iteration history for all iterative processes: parameter estimation, fitting constrained models for contrasts and Type 3 analyses, and profile likelihood confidence intervals. The last evaluation of the gradient and the negative of the Hessian (second derivative) matrix are also displayed for parameter estimation. If you perform a Bayesian analysis by specifying the BAYES statement, the iteration history for computing the mode of the posterior distribution is also displayed.

This option might result in a large amount of displayed output, especially if some of the optional iterative processes are selected.

LINK=*keyword*

specifies the link function to use in the model. The keywords and their associated built-in link functions are as follows.

LINK=	Link Function
CUMCLL	
CCLL	cumulative complementary log-log
CUMLOGIT	
CLOGIT	cumulative logit
CUMPROBIT	
CPROBIT	cumulative probit
CLOGLOG	
CLL	complementary log-log
IDENTITY	
ID	identity
LOG	log
LOGIT	logit
PROBIT	probit
POWER(*number*) | POW(*number*)	power with λ= *number*

If no LINK= option is supplied and there is a user-defined link function, the user-defined link function is used. If you specify neither the LINK= option nor a user-defined link function, then the default canonical link function is used if you specify the DIST= option. Otherwise, if you omit the DIST= option, the identity link function is used.

The cumulative link functions are appropriate only for the multinomial distribution.

LRCI

requests that two-sided confidence intervals for all model parameters be computed based on the profile likelihood function. This is sometimes called the partially maximized likelihood function. See the section "Confidence Intervals for Parameters" on page 1980 for more information about the profile likelihood function. This computation is iterative and can consume a relatively large amount of CPU time. The confidence coefficient can be selected with the ALPHA=*number* option. The resulting confidence coefficient is 1−*number*. The default confidence coefficient is 0.95.

MAXITER=*number*

MAXIT=*number*

sets the maximum allowable number of iterations for all iterative computation processes in PROC GENMOD. By default, MAXITER=50.

NOINT

requests that no intercept term be included in the model. An intercept is included unless this option is specified.

NOSCALE

holds the scale parameter fixed. Otherwise, for the normal, inverse Gaussian, and gamma distributions, the scale parameter is estimated by maximum likelihood. If you omit the SCALE= option, the scale parameter is fixed at the value 1.

OBSTATS

specifies that an additional table of statistics be displayed. Formulas for the statistics are given in the section "Predicted Values of the Mean" on page 1982, the section "Residuals"

on page 1983, and the section "Case Deletion Diagnostic Statistics" on page 1999. Residuals and fit diagnostics are not computed for multinomial models.

For each observation, the following items are displayed:

- the value of the response variable (variables if the data are binomial), frequency, and weight variables
- the values of the regression variables
- predicted mean, $\hat{\mu} = g^{-1}(\eta)$, where $\eta = \mathbf{x}_i'\hat{\boldsymbol{\beta}}$ is the linear predictor and g is the link function. If there is an offset, it is included in $\mathbf{x}_i'\hat{\boldsymbol{\beta}}$.
- estimate of the linear predictor $\mathbf{x}_i'\hat{\boldsymbol{\beta}}$. If there is an offset, it is included in $\mathbf{x}_i'\hat{\boldsymbol{\beta}}$.
- standard error of the linear predictor $\mathbf{x}_i'\hat{\boldsymbol{\beta}}$
- the value of the Hessian weight at the final iteration
- lower confidence limit of the predicted value of the mean. The confidence coefficient is specified with the ALPHA= option. See the section "Confidence Intervals on Predicted Values" on page 1982 for the computational method.
- upper confidence limit of the predicted value of the mean
- raw residual, defined as $Y - \mu$
- Pearson, or chi residual, defined as the square root of the contribution for the observation to the Pearson chi-square—that is,

  $$\frac{Y - \mu}{\sqrt{V(\mu)/w}}$$

  where Y is the response, μ is the predicted mean, w is the value of the prior weight variable specified in a WEIGHT statement, and $\mathrm{V}(\mu)$ is the variance function evaluated at μ.
- the standardized Pearson residual
- deviance residual, defined as the square root of the deviance contribution for the observation, with sign equal to the sign of the raw residual
- the standardized deviance residual
- the likelihood residual
- a Cook distance type statistic for assessing the influence of individual observations on overall model fit
- observation leverage
- DFBETA, defined as an approximation to $\hat{\beta} - \hat{\beta_{[i]}}$ for each parameter estimate $\hat{\beta}$, where $\hat{\beta_{[i]}}$ is the parameter estimate with the ith observation deleted
- standardized DFBETA, defined as DFBETA, normalized by its standard deviation
- zero inflation probability for zero inflated models
- the mean of a zero inflated response

The following additional cluster deletion diagnostic statistics are created and displayed for each cluster if a REPEATED statement is specified:

- a Cook distance type statistic for assessing the influence of entire clusters on overall model fit
- a studentized Cook distance for assessing influence of clusters
- cluster leverage
- cluster DFBETA for assessing the influence of entire clusters on individual parameter estimates
- cluster DFBETA normalized by its standard deviation

If you specify the multinomial distribution, only regression variable values, response values, predicted values, confidence limits for the predicted values, and the linear predictor are displayed in the table. Residuals and other diagnostic statistics are not available for the multinomial distribution.

The RESIDUALS, DIAGNOSTICS | INFLUENCE, PREDICTED, XVARS, and CL options cause only subgroups of the observation statistics to be displayed. You can specify more than one of these options to include different subgroups of statistics.

The ID=*variable* option causes the values of *variable* in the input data set to be displayed in the table. If an explicit format for *variable* has been defined, the formatted values are displayed.

If a REPEATED statement is present, a table is displayed for the GEE model specified in the REPEATED statement. Regression variables, response values, predicted values, confidence limits for the predicted values, linear predictor, raw residuals, Pearson residuals for each observation in the input data set are available. Case deletion diagnostic statistics are available for each observation and for each cluster.

OFFSET=*variable*

specifies a variable in the input data set to be used as an offset variable. This variable cannot be a CLASS variable, and it cannot be the response variable or one of the explanatory variables.

PREDICTED

PRED

P

requests that predicted values, the linear predictor, its standard error, and the Hessian weight be displayed (see the OBSTATS option).

RESIDUALS

R

requests that residuals and standardized residuals be displayed. Residuals and other diagnostic statistics are not available for the multinomial distribution (see the OBSTATS option).

SCALE=*number*

SCALE=PEARSON

SCALE=P

PSCALE

SCALE=DEVIANCE

SCALE=D

DSCALE

sets the value used for the scale parameter where the NOSCALE option is used. For the binomial and Poisson distributions, which have no free scale parameter, this can be used to specify an *overdispersed* model. In this case, the parameter covariance matrix and the likelihood function are adjusted by the scale parameter. See the section "Dispersion Parameter" on page 1972 and the section "Overdispersion" on page 1973 for more information. If the NOSCALE option is not specified, then *number* is used as an initial estimate of the scale parameter.

Specifying SCALE=PEARSON or SCALE=P is the same as specifying the PSCALE option. This fixes the scale parameter at the value 1 in the estimation procedure. After the parameter estimates are determined, the exponential family dispersion parameter is assumed to be given by Pearson's chi-square statistic divided by the degrees of freedom, and all statistics such as standard errors and likelihood ratio statistics are adjusted appropriately.

Specifying SCALE=DEVIANCE or SCALE=D is the same as specifying the DSCALE option. This fixes the scale parameter at a value of 1 in the estimation procedure.

After the parameter estimates are determined, the exponential family dispersion parameter is assumed to be given by the deviance divided by the degrees of freedom. All statistics such as standard errors and likelihood ratio statistics are adjusted appropriately.

SCORING=*number*

requests that on iterations up to *number*, the Hessian matrix be computed using the Fisher scoring method. For further iterations, the full Hessian matrix is computed. The default value is 1. A value of 0 causes all iterations to use the full Hessian matrix, and a value greater than or equal to the value of the MAXITER option causes all iterations to use Fisher scoring. The value of the SCORING= option must be 0 or a positive integer.

SINGULAR=*number*

sets the tolerance for testing singularity of the information matrix and the crossproducts matrix. Roughly, the test requires that a pivot be at least this number times the original diagonal value. By default, *number* is 10^7 times the machine epsilon. The default *number* is approximately 10^{-9} on most machines. This value also controls the check on estimability for ESTIMATE and CONTRAST statements.

TYPE1

requests that a Type 1, or sequential, analysis be performed. This consists of sequentially fitting models, beginning with the null (intercept term only) model and continuing up to the model specified in the MODEL statement. The likelihood ratio statistic between each successive pair of models is computed and displayed in a table.

A Type 1 analysis is not available for GEE models, since there is no associated likelihood.

TYPE3

requests that statistics for Type 3 contrasts be computed for each effect specified in the MODEL statement. The default analysis is to compute likelihood ratio statistics for the contrasts or score statistics for GEEs. Wald statistics are computed if the WALD option is also specified.

WALD
requests Wald statistics for Type 3 contrasts. You must also specify the TYPE3 option in order to compute Type 3 Wald statistics.

WALDCI
requests that two-sided Wald confidence intervals for all model parameters be computed based on the asymptotic normality of the parameter estimators. This computation is not as time-consuming as the LRCI method, since it does not involve an iterative procedure. However, it is thought to be less accurate, especially for small sample sizes. The confidence coefficient can be selected with the ALPHA= option in the same way as for the LRCI option.

XVARS
requests that the regression variables be included in the OBSTATS table.

OUTPUT Statement

OUTPUT < **OUT=***SAS-data-set* > < *keyword=name . . . keyword=name* > **;**

The OUTPUT statement creates a new SAS data set that contains all the variables in the input data set and, optionally, the estimated linear predictors (XBETA) and their standard error estimates, the weights for the Hessian matrix, predicted values of the mean, confidence limits for predicted values, residuals, and case deletion diagnostics. Residuals and diagnostic statistics are not computed for multinomial models.

You can also request these statistics with the OBSTATS, PREDICTED, RESIDUALS, DIAGNOSTICS | INFLUENCE, CL, or XVARS option in the MODEL statement. You can then create a SAS data set containing them with ODS OUTPUT commands. You might prefer to specify the OUTPUT statement for requesting these statistics since the following are true:

- The OUTPUT statement produces no tabular output.
- The OUTPUT statement creates a SAS data set more efficiently than ODS. This can be an advantage for large data sets.
- You can specify the individual statistics to be included in the SAS data set.

If you use the multinomial distribution with one of the cumulative link functions for ordinal data, the data set also contains variables named _ORDER_ and _LEVEL_ that indicate the levels of the ordinal response variable and the values of the variable in the input data set corresponding to the sorted levels. These variables indicate that the predicted value for a given observation is the probability that the response variable is as large as the value of the Value variable. Residuals and other diagnostic statistics are not available for the multinomial distribution.

The estimated linear predictor, its standard error estimate, and the predicted values and their confidence intervals are computed for all observations in which the explanatory variables are all nonmissing, even if the response is missing. By adding observations with missing response values to the input data set, you can compute these statistics for new observations or for settings of the explanatory variables not present in the data without affecting the model fit.

The following list explains specifications in the OUTPUT statement.

OUT=*SAS-data-set*
specifies the output data set. If you omit the OUT=option, the output data set is created and given a default name that uses the DATA*n* convention.

keyword=name
specifies the statistics to be included in the output data set and names the new variables that contain the statistics. Specify a keyword for each desired statistic (see the following list of keywords), an equal sign, and the name of the new variable or variables to contain the statistic. You can list only one variable after the equal sign for all the statistics, except for the case deletion diagnostics for individual parameter estimates, DFBETA, DFBETAS, DFBETAC, and DFBETACS. You can list variables enclosed in parentheses to correspond to the variables in the model, or you can specify the keyword *_all_*, without parentheses, to include deletion diagnostics for all of the parameters in the model.

Although you can use the OUTPUT statement without any *keyword=name* specifications, the output data set then contains only the original variables and, possibly, the variables Level and Value (if you use the multinomial model with ordinal data). Note that the residuals and deletion diagnostics are not available for the multinomial model with ordinal data. Some of the case deletion diagnostic statistics apply only to models for correlated data specified with a REPEATED statement. If you request these statistics for ordinary generalized linear models, the values of the corresponding variables are set to missing in the output data set. Formulas for the statistics are given in the section "Predicted Values of the Mean" on page 1982, the section "Residuals" on page 1983, and the section "Case Deletion Diagnostic Statistics" on page 1999. The keywords allowed and the statistics they represent are as follows:

DFBETA | DBETA represents the effect of deleting an observation on parameter estimates. If you specify the keyword *_all_* after the equal sign, variables named DFBETA_*ParameterName* will be included in the output data set to contain the values of the diagnostic statistic to measure the influence of deleting a single observation on the individual parameter estimates. *ParameterName* is the name of the regression model parameter formed from the input variable names concatenated with the appropriate levels, if classification variables are involved.

DFBETAS | DBETAS represents the effect of deleting an observation on standardized parameter estimates. If you specify the keyword *_all_* after the equal sign, variables named DFBETAS_*ParameterName* will be included in the output data set to contain the values of the diagnostic statistic to measure the influence of deleting a single observation on the individual parameter estimates. *ParameterName* is the name of the regression model parameter formed from the input variable names concatenated with the appropriate levels, if classification variables are involved.

DOBS | COOKD | COOKSD represents the Cook distance type statistic to measure the influence of deleting a single observation on the overall model fit.

HESSWGT represents the diagonal element of the weight matrix used in computing the Hessian matrix.

H | LEVERAGE represents the leverage of a single observation.

LOWER | L represents the lower confidence limit for the predicted value of the mean, or the lower confidence limit for the probability that the response is less than or equal to the value of Level or Value. The confidence coefficient is determined by the ALPHA=$number$ option in the MODEL statement as $(1 - number) \times 100\%$. The default confidence coefficient is 95%.

PREDICTED | PRED | PROB | P represents the predicted value of the mean of the response or the predicted probability that the response variable is less than or equal to the value of Level or Value if the multinomial model for ordinal data is used (in other words, Pr(Y $\leq$ Value), where Y is the response variable).

PZERO represents the zero-inflation probability for zero-inflated models.

RESCHI represents the Pearson (chi) residual for identifying observations that are poorly accounted for by the model.

RESDEV represents the deviance residual for identifying poorly fitted observations.

RESLIK represents the likelihood residual for identifying poorly fitted observations.

RESRAW represents the raw residual for identifying poorly fitted observations.

STDRESCHI represents the standardized Pearson (chi) residual for identifying observations that are poorly accounted for by the model.

STDRESDEV represents the standardized deviance residual for identifying poorly fitted observations.

STDXBETA represents the standard error estimate of XBETA (see the XBETA keyword).

UPPER | U represents the upper confidence limit for the predicted value of the mean, or the upper confidence limit for the probability that the response is less than or equal to the value of Level or Value. The confidence coefficient is determined by the ALPHA=$number$ option in the MODEL statement as $(1 - number) \times 100\%$. The default confidence coefficient is 95%.

XBETA represents the estimate of the linear predictor $\mathbf{x}_i'\boldsymbol{\beta}$ for observation i, or $\alpha_j + \mathbf{x}_i'\boldsymbol{\beta}$, where j is the corresponding ordered value of the response variable for the multinomial model with ordinal data. If there is an offset, it is included in $\mathbf{x}_i'\boldsymbol{\beta}$.

The keywords in the following list apply only to models specified with a REPEATED statement, fit by generalized estimating equations (GEEs).

CH | CLUSTERH | CLEVERAGE represents the leverage of a cluster.

CLUSTER represents the numerical cluster index, in order of sorted clusters.

DCLS | CLUSTERCOOKD | CLUSTERCOOKSD represents the Cook distance type statistic to measure the influence of deleting an entire cluster on the overall model fit.

DFBETAC | DBETAC represents the effect of deleting an entire cluster on parameter estimates. If you specify the keyword *_all_* after the equal sign, variables named DFBETAC_*ParameterName* will be included in the output data set to contain the values of the diagnostic statistic to measure the influence of deleting the cluster on the individual parameter estimates. *ParameterName* is the name of the regression model parameter formed from the input variable names concatenated with the appropriate levels, if classification variables are involved.

DFBETACS | DBETACS represents the effect of deleting an entire cluster on normalized parameter estimates. If you specify the keyword *_all_* after the equal sign, variables named DFBETACS_*ParameterName* will be included in the output data set to contain the values of the diagnostic statistic to measure the influence of deleting the cluster on the individual parameter estimates, normalized by their standard errors. *ParameterName* is the name of the regression model parameter formed from the input variable names concatenated with the appropriate levels, if classification variables are involved.

MCLS | CLUSTERDFIT represents the studentized Cook distance type statistic to measure the influence of deleting an entire cluster on the overall model fit.

Programming Statements

Although the most commonly used link and probability distributions are available as built-in functions, the GENMOD procedure enables you to define your own link functions and response probability distributions by using the FWDLINK, INVLINK, VARIANCE, and DEVIANCE statements. The variables assigned in these statements can have values computed in programming statements. These programming statements can occur anywhere between the PROC GENMOD statement and the RUN statement. Variable names used in programming statements must be unique. Variables from the input data set can be referenced in programming statements. The mean, linear predictor, and response are represented by the automatic variables _MEAN_, _XBETA_, and _RESP_, respectively, which can be referenced in your programming statements. Programming statements are used to define the functional dependencies of the link function, the inverse link function, the variance function, and the deviance function on the mean, linear predictor, and response variable.

The following statements illustrate the use of programming statements. Even though you usually request the Poisson distribution by specifying DIST=POISSON as a MODEL statement option, you can define the variance and deviance functions for the Poisson distribution by using the VARIANCE and DEVIANCE statements. For example, the following statements perform the same analysis as the Poisson regression example in the section "Getting Started: GENMOD Procedure" on page 1900.

The statements must be in logical order for computation, just as in a DATA step.

```
proc genmod ;
   class car age;
   a = _MEAN_;
   y = _RESP_;
   d = 2 * ( y * log( y / a ) - ( y - a ) );
   variance var = a;
   deviance dev = d;
   model c = car age / link = log offset = ln;
run;
```

The variables var and dev are dummy variables used internally by the procedure to identify the variance and deviance functions. Any valid SAS variable names can be used.

Similarly, the log link function and its inverse could be defined with the FWDLINK and INVLINK statements, as follows:

```
fwdlink link = log(_MEAN_);
invlink ilink = exp(_XBETA_);
```

These statements are for illustration, and they work well for most Poisson regression problems. If, however, in the iterative fitting process, the mean parameter becomes too close to 0, or a 0 response value occurs, an error condition occurs when the procedure attempts to evaluate the log function. You can circumvent this kind of problem by using IF-THEN/ELSE clauses or other conditional statements to check for possible error conditions and appropriately define the functions for these cases.

Data set variables can be referenced in user definitions of the link function and response distributions by using programming statements and the FWDLINK, INVLINK, DEVIANCE, and VARIANCE statements.

See the DEVIANCE, VARIANCE, FWDLINK, and INVLINK statements for more information.

REPEATED Statement

REPEATED *SUBJECT= subject-effect < / options >* **;**

The REPEATED statement specifies the covariance structure of multivariate responses for GEE model fitting in the GENMOD procedure. In addition, the REPEATED statement controls the iterative fitting algorithm used in GEEs and specifies optional output. Other GENMOD procedure statements, such as the MODEL and CLASS statements, are used in the same way as they are for ordinary generalized linear models to specify the regression model for the mean of the responses.

SUBJECT=*subject-effect*

identifies subjects in the input data set. The *subject-effect* can be a single variable, an interaction effect, a nested effect, or a combination. Each distinct value, or level, of the effect identifies a different subject, or cluster. Responses from different subjects are assumed to be statistically independent, and responses within subjects are assumed to be correlated. A

subject-effect must be specified, and variables used in defining the *subject-effect* must be listed in the CLASS statement. The input data set does not need to be sorted by subject (see the SORTED option).

The *options* control how the model is fit and what output is produced. You can specify the following options after a slash (/).

ALPHAINIT=*numbers*
specifies initial values for log odds ratio regression parameters if the LOGOR= option is specified for binary data. If this option is not specified, an initial value of 0.01 is used for all the parameters.

CONVERGE=*number*
specifies the convergence criterion for GEE parameter estimation. If the maximum absolute difference between regression parameter estimates is less than the value of *number* on two successive iterations, convergence is declared. If the absolute value of a regression parameter estimate is greater than 0.08, then the absolute difference normalized by the regression parameter value is used instead of the absolute difference. The default value of *number* is 0.0001.

CORRW
displays the estimated working correlation matrix. If you specify an exchangeable working correlation structure with the CORR=EXCH option, the CORRW option is not needed to view the estimated correlation, since a table is printed by default that contains the single estimated correlation.

CORRB
displays the estimated regression parameter correlation matrix. Both model-based and empirical correlations are displayed.

COVB
displays the estimated regression parameter covariance matrix. Both model-based and empirical covariances are displayed.

ECORRB
displays the estimated regression parameter empirical correlation matrix.

ECOVB
displays the estimated regression parameter empirical covariance matrix.

INTERCEPT=*number*
specifies either an initial or a fixed value of the intercept regression parameter in the GEE model. If you specify the NOINT option in the MODEL statement, then the intercept is fixed at the value of *number*.

INITIAL=*numbers*
specifies initial values of the regression parameters estimation, other than the intercept parameter, for GEE estimation. If this option is not specified, the estimated regression parameters assuming independence for all responses are used for the initial values.

LOGOR=*log-odds-ratio-structure-keyword*

specifies the regression structure of the log odds ratio used to model the association of the responses from subjects for binary data. The response syntax must be of the single variable type, the distribution must be binomial, and the data must be binary. Table 37.2 displays the log odds ratio structure keywords and the corresponding log odds ratio regression structures. See the section "Alternating Logistic Regressions" on page 1990 for definitions of the log odds ratio types and examples of specifying log odds ratio models. You should specify either the LOGOR= or the TYPE= option, but not both.

Table 37.2 Log Odds Ratio Regression Structures

Keyword	Log Odds Ratio Regression Structure
EXCH	exchangeable
FULLCLUST	fully parameterized clusters
LOGORVAR(*variable*)	indicator variable for specifying block effects
NESTK	*k*-nested
NEST1	1-nested
ZFULL	fully specified *z*-matrix specified in ZDATA= data set
ZREP	single cluster specification for replicated *z*-matrix specified in ZDATA= data set
ZREP(matrix)	single cluster specification for replicated *z*-matrix

MAXITER=*number*

MAXIT=*number*

specifies the maximum number of iterations allowed in the iterative GEE estimation process. The default number is 50.

MCORRB

displays the estimated regression parameter model-based correlation matrix.

MCOVB

displays the estimated regression parameter model-based covariance matrix.

MODELSE

displays an analysis of parameter estimates table that uses model-based standard errors for inference. By default, an "Analysis of Parameter Estimates" table based on empirical standard errors is displayed.

PRINTMLE

displays an analysis of maximum likelihood parameter estimates table. The maximum likelihood estimates are not displayed unless this option is specified.

RUPDATE=*number*

specifies the number of iterations between updates of the working correlation matrix. For example, RUPDATE=5 specifies that the working correlation is updated once for every five regression parameter updates. The default value of *number* is 1; that is, the working correlation is updated every time the regression parameters are updated.

SORTED

specifies that the input data are grouped by subject and sorted within subject. If this option is not specified, then the procedure internally sorts by *subject-effect* and *within subject-effect*, if a *within subject-effect* is specified.

SUBCLUSTER=*variable*

SUBCLUST=*variable*

specifies a variable defining subclusters for the 1-nested or *k*-nested log odds ratio association modeling structures. This variable must be listed in the CLASS statement.

TYPE=*correlation-structure keyword*

CORR=*correlation-structure keyword*

specifies the structure of the working correlation matrix used to model the correlation of the responses from subjects. Table 37.3 displays the correlation structure keywords and the corresponding correlation structures. The default working correlation type is the independent (CORR=IND). See the section "Details: GENMOD Procedure" on page 1964 for definitions of the correlation matrix types. You should specify LOGOR= or TYPE= but not both.

Table 37.3 Correlation Structure Types

Keyword	Correlation Matrix Type
AR	
AR(1)	autoregressive(1)
EXCH	
CS	exchangeable
IND	independent
MDEP(number)	*m*-dependent with *m*=number
UNSTR	
UN	unstructured
USER	
FIXED (matrix)	fixed, user-specified correlation matrix

For example, you can specify a fixed 4×4 correlation matrix with the following option:

```
TYPE=USER( 1.0  0.9  0.8  0.6
           0.9  1.0  0.9  0.8
           0.8  0.9  1.0  0.9
           0.6  0.8  0.9  1.0 )
```

V6CORR

specifies that the SAS 'Version 6' method of computing the normalized Pearson chi-square be used for working correlation estimation and for model-based covariance matrix scale factor.

WITHINSUBJECT | WITHIN=*within subject-effect*

defines an effect specifying the order of measurements within subjects. Each distinct level of the *within subject-effect* defines a different response from the same subject. If the data are in proper order within each subject, you do not need to specify this option.

If some measurements do not appear in the data for some subjects, this option properly orders the existing measurements and treats the omitted measurements as missing values. If the WITHINSUBJECT= option is not used in this situation, measurements might be improperly ordered and missing values assumed for the last measurements in a cluster.

Variables used in defining the *within subject-effect* must be listed in the CLASS statement.

YPAIR=*variable-list*
specifies the variables in the ZDATA= data set corresponding to pairs of responses for log odds ratio association modeling.

ZDATA=*SAS-data-set*
specifies a SAS data set containing either the full z-matrix for log odds ratio association modeling or the z-matrix for a single complete cluster to be replicated for all clusters.

ZROW=*variable-list*
specifies the variables in the ZDATA= data set corresponding to rows of the z-matrix for log odds ratio association modeling.

VARIANCE Statement

VARIANCE *variable = expression* **;**

You can specify a probability distribution other than the built-in distributions by using the VARIANCE and DEVIANCE statements. The variable name *variable* identifies the variance function to the procedure. The *expression* is used to define the functional dependence on the mean, and it can be any arithmetic expression supported by the DATA step language. You use the automatic variable _MEAN_ to represent the mean in the expression.

Alternatively, you can define the variance function with programming statements, as detailed in the section "Programming Statements" on page 1957. This form is convenient for using complex statements such as IF-THEN/ELSE clauses. Derivatives of the variance function for use during optimization are computed automatically. The DEVIANCE statement must also appear when the VARIANCE statement is used to define the variance function.

WEIGHT Statement

WEIGHT | SCWGT *variable* **;**

The WEIGHT statement identifies a variable in the input data set to be used as the exponential family dispersion parameter weight for each observation. The exponential family dispersion parameter is divided by the WEIGHT variable value for each observation. This is true regardless of whether the parameter is estimated by the procedure or specified in the MODEL statement with the SCALE= option. It is also true for distributions such as the Poisson and binomial that are not

usually defined to have a dispersion parameter. For these distributions, a WEIGHT variable weights the overdispersion parameter, which has the default value of 1.

The WEIGHT variable does not have to be an integer; if it is less than or equal to 0 or if it is missing, the corresponding observation is not used.

ZEROMODEL Statement

ZEROMODEL *effects* < */options* > **;**

The effects in the ZEROMODEL statement consist of explanatory variables or combinations of variables for the zero-inflation probability regression model in a zero-inflated model. The same effects can be used in both the ZEROMODEL statement and the MODEL statement, or effects can be used in one statement or the other separately. Explanatory variables can be continuous or classification variables. Classification variables can be character or numeric. Explanatory variables representing nominal, or classification, data must be declared in a CLASS statement. Interactions between variables can also be included as effects. Columns of the design matrix are automatically generated for classification variables and interactions. The syntax for specification of effects is the same as for the GLM procedure. See the section "Specification of Effects" on page 1974 for more information. Also refer to Chapter 39, "The GLM Procedure."

You can specify the following option in the ZEROMODEL statement after a slash (/).

LINK=*keyword*

specifies the link function to use in the model. The keywords and their associated link functions are as follows.

LINK=	Link Function
CLOGLOG	
CLL	complementary log-log
LOGIT	logit
PROBIT	probit

If no LINK= option is supplied, the LOGIT link is used. User-defined link functions are not allowed.

Details: GENMOD Procedure

Generalized Linear Models Theory

This is a brief introduction to the theory of generalized linear models.

Response Probability Distributions

In generalized linear models, the response is assumed to possess a probability distribution of the exponential form. That is, the probability density of the response Y for continuous response variables, or the probability function for discrete responses, can be expressed as

$$f(y) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\}$$

for some functions a, b, and c that determine the specific distribution. For fixed ϕ, this is a one-parameter exponential family of distributions. The functions a and c are such that $a(\phi) = \phi/w$ and $c = c(y, \phi/w)$, where w is a known weight for each observation. A variable representing w in the input data set can be specified in the WEIGHT statement. If no WEIGHT statement is specified, $w_i = 1$ for all observations.

Standard theory for this type of distribution gives expressions for the mean and variance of Y:

$$\begin{aligned} \mathrm{E}(Y) &= b'(\theta) \\ \mathrm{Var}(Y) &= \frac{b''(\theta)\phi}{w} \end{aligned}$$

where the primes denote derivatives with respect to θ. If μ represents the mean of Y, then the variance expressed as a function of the mean is

$$\mathrm{Var}(Y) = \frac{V(\mu)\phi}{w}$$

where V is the *variance function*.

Probability distributions of the response Y in generalized linear models are usually parameterized in terms of the mean μ and dispersion parameter ϕ instead of the *natural parameter* θ. The probability distributions that are available in the GENMOD procedure are shown in the following list. The zero-inflated Poisson distribution is not a generalized linear model. However, the zero-inflated Poisson is included in PROC GENMOD since it is a useful extension of generalized linear models. See Long (1997) for a discussion of the zero-inflated Poisson. The PROC GENMOD scale parameter and the variance of Y are also shown.

- Normal:

$$\begin{aligned} f(y) &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{y-\mu}{\sigma}\right)^2\right] \quad \text{for } -\infty < y < \infty \\ \phi &= \sigma^2 \\ \text{scale} &= \sigma \\ \text{Var}(Y) &= \sigma^2 \end{aligned}$$

- Inverse Gaussian:

$$\begin{aligned} f(y) &= \frac{1}{\sqrt{2\pi y^3}\sigma}\exp\left[-\frac{1}{2y}\left(\frac{y-\mu}{\mu\sigma}\right)^2\right] \quad \text{for } 0 < y < \infty \\ \phi &= \sigma^2 \\ \text{scale} &= \sigma \\ \text{Var}(Y) &= \sigma^2\mu^3 \end{aligned}$$

- Gamma:

$$\begin{aligned} f(y) &= \frac{1}{\Gamma(\nu)y}\left(\frac{y\nu}{\mu}\right)^{\nu}\exp\left(-\frac{y\nu}{\mu}\right) \quad \text{for } 0 < y < \infty \\ \phi &= \nu^{-1} \\ \text{scale} &= \nu \\ \text{Var}(Y) &= \frac{\mu^2}{\nu} \end{aligned}$$

- Geometric: This is a special case of the negative binomial with $k = 1$.

$$\begin{aligned} f(y) &= \frac{(\mu)^y}{(1+\mu)^{y+1}} \quad \text{for } y = 0, 1, 2, \ldots \\ \phi &= 1 \\ \text{Var}(Y) &= \mu(1+\mu) \end{aligned}$$

- Negative binomial:

$$\begin{aligned} f(y) &= \frac{\Gamma(y+1/k)}{\Gamma(y+1)\Gamma(1/k)}\frac{(k\mu)^y}{(1+k\mu)^{y+1/k}} \quad \text{for } y = 0, 1, 2, \ldots \\ \text{dispersion} &= k \\ \text{Var}(Y) &= \mu + k\mu^2 \end{aligned}$$

- Poisson:

$$\begin{aligned} f(y) &= \frac{\mu^y \mathrm{e}^{-\mu}}{y!} \quad \text{for } y = 0, 1, 2, \ldots \\ \phi &= 1 \\ \mathrm{Var}(Y) &= \mu \end{aligned}$$

- Binomial:

$$\begin{aligned} f(y) &= \left(\begin{array}{c} n \\ r \end{array} \right) \mu^r (1-\mu)^{n-r} \quad \text{for } y = \frac{r}{n},\ r = 0, 1, 2, \ldots, n \\ \phi &= 1 \\ \mathrm{Var}(Y) &= \frac{\mu(1-\mu)}{n} \end{aligned}$$

- Multinomial:

$$\begin{aligned} f(y_1, y_2, \cdots, y_k) &= \frac{m!}{y_1! y_2! \cdots y_k!} p_1^{y_1} p_2^{y_2} \cdots p_k^{y_k} \\ \phi &= 1 \end{aligned}$$

- Zero-inflated Poisson:

$$\begin{aligned} f(y) &= \begin{cases} \omega + (1-\omega)\mathrm{e}^{-\lambda} & \text{for } y = 0 \\ (1-\omega)\frac{\lambda^y \mathrm{e}^{-\lambda}}{y!} & \text{for } y = 1, 2, \ldots \end{cases} \\ \phi &= 1 \\ \mu = \mathrm{E}(Y) &= (1-\omega)\lambda \\ \mathrm{Var}(Y) &= \mu + \frac{\omega}{1-\omega}\mu^2 \end{aligned}$$

The negative binomial distribution contains a parameter k, called the negative binomial dispersion parameter. This is not the same as the generalized linear model dispersion ϕ, but it is an additional distribution parameter that must be estimated or set to a fixed value.

For the binomial distribution, the response is the binomial proportion $Y = \textit{events}/\textit{trials}$. The variance function is $V(\mu) = \mu(1-\mu)$, and the binomial trials parameter n is regarded as a weight w.

If a weight variable is present, ϕ is replaced with ϕ/w, where w is the weight variable.

PROC GENMOD works with a scale parameter that is related to the exponential family dispersion parameter ϕ instead of working with ϕ itself. The scale parameters are related to the dispersion parameter as shown previously with the probability distribution definitions. Thus, the scale parameter output in the "Analysis of Parameter Estimates" table is related to the exponential family dispersion parameter. If you specify a constant scale parameter with the SCALE= option in the MODEL statement, it is also related to the exponential family dispersion parameter in the same way.

Link Function

The mean μ_i of the response in the ith observation is related to a linear predictor through a monotonic differentiable link function g.

$$g(\mu_i) = \mathbf{x}_i'\boldsymbol{\beta}$$

Here, $\mathbf{x}_i$ is a fixed known vector of explanatory variables, and $\boldsymbol{\beta}$ is a vector of unknown parameters.

Log-Likelihood Functions

Log-likelihood functions for the distributions that are available in the procedure are parameterized in terms of the means μ_i and the dispersion parameter ϕ. The term y_i represents the response for the ith observation, and w_i represents the known dispersion weight. The log-likelihood functions are of the form

$$L(\mathbf{y}, \boldsymbol{\mu}, \phi) = \sum_i \log\left(f(y_i, \mu_i, \phi)\right)$$

where the sum is over the observations. The forms of the individual contributions

$$l_i = \log\left(f(y_i, \mu_i, \phi)\right)$$

are shown in the following list; the parameterizations are expressed in terms of the mean and dispersion parameters.

For the discrete distributions (binomial, multinomial, negative binomial, and Poisson), the functions computed as the sum of the l_i terms are not proper log-likelihood functions, since terms involving binomial coefficients or factorials of the observed counts are dropped from the computation of the log likelihood, and a dispersion parameter ϕ is included in the computation. Deletion of factorial terms and inclusion of a dispersion parameter do not affect parameter estimates or their estimated covariances for these distributions, and this is the function used in maximum likelihood estimation. The value of ϕ used in computing the reported log-likelihood function is either the final estimated value, or the fixed value, if the dispersion parameter is fixed. Even though it is not a proper log-likelihood function in all cases, the function computed as the sum of the l_i terms is reported in the output as the *log likelihood*. The proper log-likelihood function is also computed as the sum of the ll_i terms in the following list, and it is reported as the *full log likelihood* in the output.

- Normal:

$$ll_i = l_i = -\frac{1}{2}\left[\frac{w_i(y_i - \mu_i)^2}{\phi} + \log\left(\frac{\phi}{w_i}\right) + \log(2\pi)\right]$$

- Inverse Gaussian:

$$ll_i = l_i = -\frac{1}{2}\left[\frac{w_i(y_i - \mu_i)^2}{y_i\mu^2\phi} + \log\left(\frac{\phi y_i^3}{w_i}\right) + \log(2\pi)\right]$$

- Gamma:

$$ll_i = l_i = \frac{w_i}{\phi}\log\left(\frac{w_i y_i}{\phi\mu_i}\right) - \frac{w_i y_i}{\phi\mu_i} - \log(y_i) - \log\left(\Gamma\left(\frac{w_i}{\phi}\right)\right)$$

- Negative binomial:

$$l_i = y_i \log\left(\frac{k\mu}{w_i}\right) - (y_i + w_i/k)\log\left(1 + \frac{k\mu}{w_i}\right) + \log\left(\frac{\Gamma(y_i + w_i/k)}{\Gamma(w_i/k)}\right)$$

$$ll_i = y_i \log\left(\frac{k\mu}{w_i}\right) - (y_i + w_i/k)\log\left(1 + \frac{k\mu}{w_i}\right) + \log\left(\frac{\Gamma(y_i + w_i/k)}{\Gamma(y_i + 1)\Gamma(w_i/k)}\right)$$

- Poisson:

$$l_i = \frac{w_i}{\phi}[y_i \log(\mu_i) - \mu_i]$$

$$ll_i = w_i[y_i \log(\mu_i) - \mu_i - \log(y_i!)]$$

- Binomial:

$$l_i = \frac{w_i}{\phi}[r_i \log(p_i) + (n_i - r_i)\log(1 - p_i)]$$

$$ll_i = w_i[\log\left(\begin{array}{c} n_i \\ r_i \end{array}\right) + r_i \log(p_i) + (n_i - r_i)\log(1 - p_i)]$$

- Multinomial (k categories):

$$l_i = \frac{w_i}{\phi}\sum_{j=1}^{k} y_{ij} \log(\mu_{ij})$$

$$ll_i = w_i[\log(m_i!) + \sum_{j=1}^{k}(y_{ij} \log(\mu_{ij}) - \log(y_{ij}!))]$$

- Zero-inflated Poisson:

$$l_i = ll_i = \begin{cases} w_i \log[\omega_i + (1 - \omega_i)\exp(-\mu_i)] & y_i = 0 \\ w_i[\log(1 - \omega_i) + y_i \log(\mu_i) - \mu_i - \log(y_i!)] & y_i > 0 \end{cases}$$

Maximum Likelihood Fitting

The GENMOD procedure uses a ridge-stabilized Newton-Raphson algorithm to maximize the log-likelihood function $L(\mathbf{y}, \boldsymbol{\mu}, \phi)$ with respect to the regression parameters. By default, the procedure also produces maximum likelihood estimates of the scale parameter as defined in the section "Response Probability Distributions" on page 1964 for the normal, inverse Gaussian, negative binomial, and gamma distributions.

On the rth iteration, the algorithm updates the parameter vector $\boldsymbol{\beta}_r$ with

$$\boldsymbol{\beta}_{r+1} = \boldsymbol{\beta}_r - \mathbf{H}^{-1}\mathbf{s}$$

where $\mathbf{H}$ is the Hessian (second derivative) matrix, and $\mathbf{s}$ is the gradient (first derivative) vector of the log-likelihood function, both evaluated at the current value of the parameter vector. That is,

$$\mathbf{s} = [s_j] = \left[\frac{\partial L}{\partial \beta_j}\right]$$

and

$$\mathbf{H} = [h_{ij}] = \left[\frac{\partial^2 L}{\partial \beta_i \partial \beta_j}\right]$$

In some cases, the scale parameter is estimated by maximum likelihood. In these cases, elements corresponding to the scale parameter are computed and included in $\mathbf{s}$ and $\mathbf{H}$.

If $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$ is the linear predictor for observation i and g is the link function, then $\eta_i = g(\mu_i)$, so that $\mu_i = g^{-1}(\mathbf{x}_i'\boldsymbol{\beta})$ is an estimate of the mean of the ith observation, obtained from an estimate of the parameter vector $\boldsymbol{\beta}$.

The gradient vector and Hessian matrix for the regression parameters are given by

$$\begin{aligned} \mathbf{s} &= \sum_i \frac{w_i(y_i - \mu_i)\mathbf{x}_i}{V(\mu_i)g'(\mu_i)\phi} \\ \mathbf{H} &= -\mathbf{X}'\mathbf{W}_o\mathbf{X} \end{aligned}$$

where $\mathbf{X}$ is the design matrix, $\mathbf{x}_i$ is the transpose of the ith row of $\mathbf{X}$, and V is the variance function. The matrix $\mathbf{W}_o$ is diagonal with its ith diagonal element

$$w_{oi} = w_{ei} + w_i(y_i - \mu_i)\frac{V(\mu_i)g''(\mu_i) + V'(\mu_i)g'(\mu_i)}{(V(\mu_i))^2(g'(\mu_i))^3\phi}$$

where

$$w_{ei} = \frac{w_i}{\phi V(\mu_i)(g'(\mu_i))^2}$$

The primes denote derivatives of g and V with respect to μ. The negative of $\mathbf{H}$ is called the observed information matrix. The expected value of $\mathbf{W}_o$ is a diagonal matrix $\mathbf{W}_e$ with diagonal values w_{ei}. If you replace $\mathbf{W}_o$ with $\mathbf{W}_e$, then the negative of $\mathbf{H}$ is called the expected information matrix. $\mathbf{W}_e$ is the weight matrix for the Fisher scoring method of fitting. Either $\mathbf{W}_o$ or $\mathbf{W}_e$ can be used in the update equation. The GENMOD procedure uses Fisher scoring for iterations up to the number specified by the SCORING option in the MODEL statement, and it uses the observed information matrix on additional iterations.

Covariance and Correlation Matrix

The estimated covariance matrix of the parameter estimator is given by

$$\mathbf{\Sigma} = -\mathbf{H}^{-1}$$

where $\mathbf{H}$ is the Hessian matrix evaluated using the parameter estimates on the last iteration. Note that the dispersion parameter, whether estimated or specified, is incorporated into $\mathbf{H}$. Rows and columns corresponding to aliased parameters are not included in $\mathbf{\Sigma}$.

The correlation matrix is the normalized covariance matrix. That is, if σ_{ij} is an element of $\mathbf{\Sigma}$, then the corresponding element of the correlation matrix is $\sigma_{ij}/\sigma_i\sigma_j$, where $\sigma_i = \sqrt{\sigma_{ii}}$.

Goodness of Fit

Two statistics that are helpful in assessing the goodness of fit of a given generalized linear model are the scaled deviance and Pearson's chi-square statistic. For a fixed value of the dispersion parameter ϕ, the scaled deviance is defined to be twice the difference between the maximum achievable log likelihood and the log likelihood at the maximum likelihood estimates of the regression parameters.

Note that these statistics are not valid for GEE models.

If $l(\mathbf{y}, \boldsymbol{\mu})$ is the log-likelihood function expressed as a function of the predicted mean values $\boldsymbol{\mu}$ and the vector $\mathbf{y}$ of response values, then the scaled deviance is defined by

$$D^*(\mathbf{y}, \boldsymbol{\mu}) = 2(l(\mathbf{y}, \mathbf{y}) - l(\mathbf{y}, \boldsymbol{\mu}))$$

For specific distributions, this can be expressed as

$$D^*(\mathbf{y}, \boldsymbol{\mu}) = \frac{D(\mathbf{y}, \boldsymbol{\mu})}{\phi}$$

where D is the deviance. The following table displays the deviance for each of the probability distributions available in PROC GENMOD. The deviance cannot be directly calculated for the zero-inflated Poisson model. Twice the negative of the log likelihood is reported instead of the proper deviance for the zero-inflated Poisson.

Distribution	Deviance
normal	$\sum_i w_i(y_i - \mu_i)^2$
Poisson	$2\sum_i w_i \left[y_i \log\left(\frac{y_i}{\mu_i}\right) - (y_i - \mu_i) \right]$
binomial	$2\sum_i w_i m_i \left[y_i \log\left(\frac{y_i}{\mu_i}\right) + (1 - y_i) \log\left(\frac{1-y_i}{1-\mu_i}\right) \right]$
gamma	$2\sum_i w_i \left[-\log\left(\frac{y_i}{\mu_i}\right) + \frac{y_i - \mu_i}{\mu_i} \right]$
inverse Gaussian	$\sum_i \frac{w_i(y_i-\mu_i)^2}{\mu_i^2 y_i}$
multinomial	$\sum_i \sum_j w_i y_{ij} \log\left(\frac{y_{ij}}{p_{ij} m_i}\right)$
negative binomial	$2\sum_i \left[y \log(y/\mu) - (y + w_i/k) \log\left(\frac{y + w_i/k}{\mu + w_i/k}\right) \right]$
zero-inflated Poisson	$-2\sum_i \begin{cases} w_i \log[\omega_i + (1-\omega_i)\exp(-\mu_i)] & y_i = 0 \\ w_i[\log(1-\omega_i) + y_i \log(\mu_i) - \mu_i - \log(y_i!)] & y_i > 0 \end{cases}$

In the binomial case, $y_i = r_i/m_i$, where r_i is a binomial count and m_i is the binomial number of trials parameter.

In the multinomial case, y_{ij} refers to the observed number of occurrences of the jth category for the ith subpopulation defined by the AGGREGATE= variable, m_i is the total number in the ith subpopulation, and p_{ij} is the category probability.

Pearson's chi-square statistic is defined as

$$X^2 = \sum_i \frac{w_i(y_i - \mu_i)^2}{V(\mu_i)}$$

and the scaled Pearson's chi-square is X^2/ϕ.

The scaled version of both of these statistics, under certain regularity conditions, has a limiting chi-square distribution, with degrees of freedom equal to the number of observations minus the number of parameters estimated. The scaled version can be used as an approximate guide to the goodness of fit of a given model. Use caution before applying these statistics to ensure that all the conditions for the asymptotic distributions hold. McCullagh and Nelder (1989) advise that differences in deviances for nested models can be better approximated by chi-square distributions than the deviances can themselves.

In cases where the dispersion parameter is not known, an estimate can be used to obtain an approximation to the scaled deviance and Pearson's chi-square statistic. One strategy is to fit a model that contains a sufficient number of parameters so that all systematic variation is removed, estimate ϕ from this model, and then use this estimate in computing the scaled deviance of submodels. The deviance or Pearson's chi-square divided by its degrees of freedom is sometimes used as an estimate of the dispersion parameter ϕ. For example, since the limiting chi-square distribution of the scaled deviance $D^* = D/\phi$ has $n - p$ degrees of freedom, where n is the number of observations and p is the number of parameters, equating D^* to its mean and solving for ϕ yields $\hat{\phi} = D/(n - p)$. Similarly, an estimate of ϕ based on Pearson's chi-square X^2 is $\hat{\phi} = X^2/(n - p)$. Alternatively, a maximum likelihood estimate of ϕ can be computed by the procedure, if desired. See the discussion in the section "Type 1 Analysis" on page 1978 for more about the estimation of the dispersion parameter.

Other Fit Statistics

The Akaike information criterion (AIC) is a measure of goodness of model fit that balances model fit against model simplicity. AIC has the form

$$\text{AIC} = -2LL + 2p$$

where p is the number of parameters estimated in the model, and LL is the log likelihood evaluated at the value of the estimated parameters. An alternative form is the corrected AIC given by

$$\text{AICC} = -2LL + 2p\frac{n}{n - p - 1}$$

where n is the total number of observations used.

The Bayesian information criterion (BIC) is a similar measure. BIC is defined by

$$\text{BIC} = -2LL + p\log(n)$$

See Akaike (1981, 1979) for details of AIC and BIC. See Simonoff (2003) for a discussion of using AIC, AICC, and BIC with generalized linear models. These criteria are useful in selecting among regression models, with smaller values representing better model fit. PROC GENMOD uses the full log likelihoods defined in the section "Log-Likelihood Functions" on page 1967, with all terms included, for computing all of the criteria.

Dispersion Parameter

There are several options available in PROC GENMOD for handling the exponential distribution dispersion parameter. The NOSCALE and SCALE options in the MODEL statement affect the way in which the dispersion parameter is treated. If you specify the SCALE=DEVIANCE option, the dispersion parameter is estimated by the deviance divided by its degrees of freedom. If you specify the SCALE=PEARSON option, the dispersion parameter is estimated by Pearson's chi-square statistic divided by its degrees of freedom.

Otherwise, values of the SCALE and NOSCALE options and the resultant actions are displayed in the following table.

NOSCALE	**SCALE=*value***	**Action**
present	present	scale fixed at *value*
present	not present	scale fixed at 1
not present	not present	scale estimated by ML
not present	present	scale estimated by ML, starting point at *value*
present (negative binomial)	not present	k fixed at 0

The meaning of the scale parameter displayed in the "Analysis Of Parameter Estimates" table is different for the gamma distribution than for the other distributions. The relation of the scale parameter as used by PROC GENMOD to the exponential family dispersion parameter ϕ is displayed in the following table. For the binomial and Poisson distributions, ϕ is the overdispersion parameter, as defined in the "Overdispersion" section, which follows.

Distribution	**Scale**
normal	$\sqrt{\phi}$
inverse Gaussian	$\sqrt{\phi}$
gamma	$1/\phi$
binomial	$\sqrt{\phi}$
Poisson	$\sqrt{\phi}$

In the case of the negative binomial distribution, PROC GENMOD reports the "dispersion" parameter estimated by maximum likelihood. This is the negative binomial parameter k defined in the section "Response Probability Distributions" on page 1964.

Overdispersion

Overdispersion is a phenomenon that sometimes occurs in data that are modeled with the binomial or Poisson distributions. If the estimate of dispersion after fitting, as measured by the deviance or Pearson's chi-square, divided by the degrees of freedom, is not near 1, then the data might be *overdispersed* if the dispersion estimate is greater than 1 or *underdispersed* if the dispersion estimate is less than 1. A simple way to model this situation is to allow the variance functions of these distributions to have a multiplicative overdispersion factor ϕ:

- binomial: $V(\mu) = \phi\mu(1-\mu)$
- Poisson: $V(\mu) = \phi\mu$

An alternative method to allow for overdispersion in the Poisson distribution is to fit a negative binomial distribution, where $V(\mu) = \mu + k\mu^2$, instead of the Poisson. The parameter k can be estimated by maximum likelihood, thus allowing for overdispersion of a specific form. This is different from the multiplicative overdispersion factor ϕ, which can accommodate many forms of overdispersion.

The models are fit in the usual way, and the parameter estimates are not affected by the value of ϕ. The covariance matrix, however, is multiplied by ϕ, and the scaled deviance and log likelihoods used in likelihood ratio tests are divided by ϕ. The profile likelihood function used in computing confidence intervals is also divided by ϕ. If you specify a WEIGHT statement, ϕ is divided by the value of the WEIGHT variable for each observation. This has the effect of multiplying the contributions of the log-likelihood function, the gradient, and the Hessian by the value of the WEIGHT variable for each observation.

The SCALE= option in the MODEL statement enables you to specify a value of $\sigma = \sqrt{\phi}$ for the binomial and Poisson distributions. If you specify the SCALE=DEVIANCE option in the MODEL statement, the procedure uses the deviance divided by degrees of freedom as an estimate of ϕ, and all statistics are adjusted appropriately. You can use Pearson's chi-square instead of the deviance by specifying the SCALE=PEARSON option.

The function obtained by dividing a log-likelihood function for the binomial or Poisson distribution by a dispersion parameter is not a legitimate log-likelihood function. It is an example of a *quasi-likelihood* function. Most of the asymptotic theory for log likelihoods also applies to quasi-likelihoods, which justifies computing standard errors and likelihood ratio statistics by using quasi-likelihoods instead of proper log likelihoods. See McCullagh and Nelder (1989, Chapter 9), McCullagh (1983), and Hardin and Hilbe (2003) for details on quasi-likelihood functions.

Although the estimate of the dispersion parameter is often used to indicate overdispersion or underdispersion, this estimate might also indicate other problems such as an incorrectly specified model or outliers in the data. You should carefully assess whether this type of model is appropriate for your data.

Specification of Effects

Each term in a model is called an effect. Effects are specified in the MODEL statement. You specify effects with a special notation that uses variable names and operators. There are two types of variables, *classification* (or *CLASS*) variables and *continuous* variables. There are two primary types of operators, *crossing* and *nesting*. A third type, the *bar* operator, is used to simplify effect specification. Crossing is the type of operator most commonly used in generalized linear models.

Variables that identify classification levels are called *CLASS* variables in SAS and are identified in a CLASS statement. These might also be called *categorical, qualitative, discrete,* or *nominal* variables. CLASS variables can be either character or numeric. The values of CLASS variables are called *levels*. For example, the CLASS variable Sex could have the levels 'male' and 'female'.

In a model, an explanatory variable that is not declared in a CLASS statement is assumed to be continuous. Continuous variables must be numeric. For example, the heights and weights of subjects in an experiment are continuous variables.

The types of effects most useful in generalized linear models are shown in the following list. Assume that A, B, and C are classification variables and that X1 and X2 are continuous variables.

- Regressor effects are specified by writing continuous variables by themselves: X1, X2.
- Polynomial effects are specified by joining two or more continuous variables with asterisks: X1*X2.
- Main effects are specified by writing classification variables by themselves: A, B, C.
- Crossed effects (interactions) are specified by joining two or more classification variables with asterisks: A*B, B*C, A*B*C.
- Nested effects are specified by following a main effect or crossed effect with a classification variable or list of classification variables enclosed in parentheses: B(A), C(B A), A*B(C). In the preceding example, B(A) is "B nested within A."
- Combinations of continuous and classification variables can be specified in the same way by using the crossing and nesting operators.

The bar operator consists of two effects joined with a vertical bar (|). It is shorthand notation for including the left-hand side, the right-hand side, and the cross between them as effects in the model. For example, A | B is equivalent to A B A*B. The effects in the bar operator can be classification variables, continuous variables, or combinations of effects defined using operators. Multiple bars are permitted. For example, A | B | C means A B C A*B A*C B*C A*B*C.

You can specify the maximum number of variables in any effect that results from bar evaluation by specifying the maximum number, preceded by an @ sign. For example, A | B | C@2 results in effects that involve two or fewer variables: A B C A*B A*C B*C.

Parameterization Used in PROC GENMOD

Design Matrix

The linear predictor part of a generalized linear model is

$$\eta = \mathbf{X}\beta$$

where β is an unknown parameter vector and $\mathbf{X}$ is a known design matrix. By default, all models automatically contain an intercept term; that is, the first column of $\mathbf{X}$ contains all 1s. Additional columns of $\mathbf{X}$ are generated for classification variables, regression variables, and any interaction terms included in the model. It is important to understand the ordering of classification variable parameters when you use the ESTIMATE or CONTRAST statement. The ordering of these parameters is displayed in the "CLASS Level Information" table and in tables displaying the parameter estimates of the fitted model.

When you specify an overparameterized model with the PARAM=GLM option in the CLASS statement, some columns of $\mathbf{X}$ can be linearly dependent on other columns. For example, when you specify a model consisting of an intercept term and a classification variable, the column corresponding to any one of the levels of the classification variable is linearly dependent on the other columns of $\mathbf{X}$. The columns of $\mathbf{X}'\mathbf{X}$ are checked in the order in which the model is specified for dependence on preceding columns. If a dependency is found, the parameter corresponding to the dependent column is set to 0 along with its standard error to indicate that it is not estimated. The order in which the levels of a classification variable are checked for dependencies can be set by the ORDER= option in the PROC GENMOD statement or by the ORDER= option in the CLASS statement. For full-rank parameterizations, the columns of the $\mathbf{X}$ matrix are designed to be linearly independent.

You can exclude the intercept term from the model by specifying the NOINT option in the MODEL statement.

Missing Level Combinations

All levels of interaction terms involving classification variables might not be represented in the data. In that case, PROC GENMOD does not include parameters in the model for the missing levels.

CLASS Variable Parameterization

Consider a model with one CLASS variable A with four levels, 1, 2, 5, and 7. Details of the possible choices for the PARAM= option follow.

EFFECT — Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of -1. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows:

Effect Coding			
	Design Matrix		
A	A1	A2	A5
1	1	0	0
2	0	1	0
5	0	0	1
7	−1	−1	−1

Parameter estimates of CLASS main effects that use the effect coding scheme estimate the difference in the effect of each nonreference level compared to the average effect over all four levels.

GLM

As in PROC GLM, four columns are created to indicate group membership. The design matrix columns for A are as follows:

GLM Coding				
	Design Matrix			
A	A1	A2	A5	A7
1	1	0	0	0
2	0	1	0	0
5	0	0	1	0
7	0	0	0	1

Parameter estimates of CLASS main effects that use the GLM coding scheme estimate the difference in the effects of each level compared to the last level.

ORDINAL

THERMOMETER

Three columns are created to indicate group membership of the higher levels of the effect. For the first level of the effect (which for A is 1), all three dummy variables have a value of 0. The design matrix columns for A are as follows:

Ordinal Coding			
	Design Matrix		
A	A2	A5	A7
1	0	0	0
2	1	0	0
5	1	1	0
7	1	1	1

The first level of the effect is a control or baseline level. Parameter estimates of CLASS main effects that use the ORDINAL coding scheme estimate the effect on the response as the ordinal factor is set to each succeeding level. When the parameters for an ordinal main effect have the same sign, the response effect is monotonic across the levels.

POLYNOMIAL

POLY

Three columns are created. The first represents the linear term (x), the second represents the quadratic term (x^2), and the third represents the cubic term (x^3), where x is the level value. If the CLASS levels are not numeric, they are translated into 1, 2, 3, ... according to their sorting order. The design matrix columns for A are as follows:

Polynomial Coding			
	Design Matrix		
A	APOLY1	APOLY2	APOLY3
1	1	1	1
2	2	4	8
5	5	25	125
7	7	49	343

REFERENCE

REF Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of 0. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows:

Reference Coding			
	Design Matrix		
A	A1	A2	A5
1	1	0	0
2	0	1	0
5	0	0	1
7	0	0	0

Parameter estimates of CLASS main effects that use the reference coding scheme estimate the difference in the effect of each nonreference level compared to the effect of the reference level.

ORTHEFFECT The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=EFFECT. The design matrix columns for A are as follows:

Orthogonal Effect Coding			
	Design Matrix		
A	AOEFF1	AOEFF2	AOEFF3
1	1.41421	−0.81650	−0.57735
2	0.00000	1.63299	−0.57735
5	0.00000	0.00000	1.73205
7	−1.41421	−0.81649	−0.57735

ORTHORDINAL

ORTHOTHERM The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=ORDINAL. The design matrix columns for A are as follows:

Orthogonal Ordinal Coding			
	Design Matrix		
A	AOORD1	AOORD2	AOORD3
1	−1.73205	0.00000	0.00000
2	0.57735	−1.63299	0.00000
5	0.57735	0.81650	−1.41421
7	0.57735	0.81650	1.41421

ORTHPOLY The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=POLY. The design matrix columns for A are as follows:

Orthogonal Polynomial Coding			
	Design Matrix		
A	AOPOLY1	AOPOLY2	AOPOLY5
1	−1.153	0.907	−0.921
2	−0.734	−0.540	1.473
5	0.524	−1.370	−0.921
7	1.363	1.004	0.368

ORTHREF The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=REFERENCE. The design matrix columns for A are as follows:

Orthogonal Reference Coding			
	Design Matrix		
A	AOREF1	AOREF2	AOREF3
1	1.73205	0.00000	0.00000
2	−0.57735	1.63299	0.00000
5	−0.57735	−0.81650	1.41421
7	−0.57735	−0.81650	−1.41421

Type 1 Analysis

A Type 1 analysis consists of fitting a sequence of models, beginning with a simple model with only an intercept term, and continuing through a model of specified complexity, fitting one additional effect on each step. Likelihood ratio statistics—that is, twice the difference of the log likelihoods—are computed between successive models. This type of analysis is sometimes called an analysis of deviance since, if the dispersion parameter is held fixed for all models, it is equivalent to computing differences of scaled deviances. The asymptotic distribution of the likelihood ratio statistics, under the hypothesis that the additional parameters included in the model are equal to 0, is a chi-square with degrees of freedom equal to the difference in the number of parameters estimated in the successive models. Thus, these statistics can be used in a test of hypothesis of the significance of each additional term fit.

This type of analysis is not available for GEE models, since the deviance is not computed for this type of model.

If the dispersion parameter ϕ is known, it can be included in the models; if it is unknown, there are two strategies allowed by PROC GENMOD. The dispersion parameter can be estimated from a maximal model by the deviance or Pearson's chi-square divided by degrees of freedom, as discussed in the section "Goodness of Fit" on page 1970, and this value can be used in all models. An alternative is to consider the dispersion to be an additional unknown parameter for each model and estimate it by maximum likelihood on each step. By default, PROC GENMOD estimates scale by maximum likelihood at each step.

A table of likelihood ratio statistics is produced, along with associated p-values based on the asymptotic chi-square distributions.

If you specify either the SCALE=DEVIANCE or the SCALE=PEARSON option in the MODEL statement, the dispersion parameter is estimated using the deviance or Pearson's chi-square statistic, and F statistics are computed in addition to the chi-square statistics for assessing the significance of each additional term in the Type 1 analysis. See the section "*F* Statistics" on page 1981 for a definition of F statistics.

This Type 1 analysis has the general property that the results depend on the order in which the terms of the model are fitted. The terms are fitted in the order in which they are specified in the MODEL statement.

Type 3 Analysis

A Type 3 analysis is similar to the Type III sums of squares used in PROC GLM, except that likelihood ratios are used instead of sums of squares. First, a Type III estimable function is defined for an effect of interest in exactly the same way as in PROC GLM. Then maximum likelihood estimates are computed under the constraint that the Type III function of the parameters is equal to 0, by using constrained optimization. Let the resulting constrained parameter estimates be $\tilde{\boldsymbol{\beta}}$ and the log likelihood be $l(\tilde{\boldsymbol{\beta}})$. Then the likelihood ratio statistic

$$S = 2(l(\hat{\boldsymbol{\beta}}) - l(\tilde{\boldsymbol{\beta}}))$$

where $\hat{\boldsymbol{\beta}}$ is the unconstrained estimate, has an asymptotic chi-square distribution under the hypothesis that the Type III contrast is equal to 0, with degrees of freedom equal to the number of parameters associated with the effect.

When a Type 3 analysis is requested, PROC GENMOD produces a table that contains the likelihood ratio statistics, degrees of freedom, and p-values based on the limiting chi-square distributions for each effect in the model. If you specify either the DSCALE or PSCALE option in the MODEL statement, F statistics are also computed for each effect.

Options for handling the dispersion parameter are the same as for a Type 1 analysis. The dispersion parameter can be specified to be a known value, estimated from the deviance or Pearson's chi-square divided by degrees of freedom, or estimated by maximum likelihood individually for the unconstrained and constrained models. By default, PROC GENMOD estimates scale by maximum likelihood for each model fit.

The results of this type of analysis do not depend on the order in which the terms are specified in the MODEL statement.

A Type 3 analysis can consume considerable computation time since a constrained model is fitted for each effect. Wald statistics for Type 3 contrasts are computed if you specify the WALD option. Wald statistics for contrasts use less computation time than likelihood ratio statistics but might be less accurate indicators of the significance of the effect of interest. The Wald statistic for testing $\mathbf{L}'\boldsymbol{\beta} = \mathbf{0}$, where $\mathbf{L}$ is the contrast matrix, is defined by

$$S = (\mathbf{L}'\hat{\boldsymbol{\beta}})'(\mathbf{L}'\hat{\boldsymbol{\Sigma}}\mathbf{L})^{-}(\mathbf{L}'\hat{\boldsymbol{\beta}})$$

where $\boldsymbol{\beta}$ is the maximum likelihood estimate and $\boldsymbol{\Sigma}$ is its estimated covariance matrix. The asymptotic distribution of S is chi-square with r degrees of freedom, where r is the rank of $\mathbf{L}$.

See Chapter 39, "The GLM Procedure,"and Chapter 15, "The Four Types of Estimable Functions," for more information about Type III estimable functions. Also refer to Littell, Freund, and Spector (1991).

Generalized score tests for Type III contrasts are computed for GEE models if you specify the TYPE3 option in the MODEL statement when a REPEATED statement is also used. See the section "Generalized Score Statistics" on page 1995 for more information about generalized score statistics. Wald tests are also available with the Wald option in the CONTRAST statement. In this case, the robust covariance matrix estimate is used for $\mathbf{\Sigma}$ in the Wald statistic.

Confidence Intervals for Parameters

Likelihood Ratio-Based Confidence Intervals

PROC GENMOD produces likelihood ratio-based confidence intervals, also known as profile likelihood confidence intervals, for parameter estimates for generalized linear models. These are not computed for GEE models, since there is no likelihood for this type of model. Suppose that the parameter vector is $\boldsymbol{\beta} = [\beta_0, \beta_1, \ldots, \beta_p]'$ and that you want a confidence interval for β_j. The profile likelihood function for β_j is defined as

$$l^*(\beta_j) = \max_{\tilde{\boldsymbol{\beta}}} l(\boldsymbol{\beta})$$

where $\tilde{\boldsymbol{\beta}}$ is the vector $\boldsymbol{\beta}$ with the jth element fixed at β_j and l is the log-likelihood function. If $l = l(\hat{\boldsymbol{\beta}})$ is the log likelihood evaluated at the maximum likelihood estimate $\hat{\boldsymbol{\beta}}$, then $2(l - l^*(\beta_j))$ has a limiting chi-square distribution with one degree of freedom if β_j is the true parameter value. A $(1-\alpha)100\%$ confidence interval for β_j is

$$\{\beta_j : l^*(\beta_j) \geq l_0 = l - 0.5\chi^2_{1-\alpha,1}\}$$

where $\chi^2_{1-\alpha,1}$ is the $100(1-\alpha)$th percentile of the chi-square distribution with one degree of freedom. The endpoints of the confidence interval can be found by solving numerically for values of β_j that satisfy equality in the preceding relation. PROC GENMOD solves this by starting at the maximum likelihood estimate of $\boldsymbol{\beta}$. The log-likelihood function is approximated with a quadratic surface, for which an exact solution is possible. The process is iterated until convergence to an endpoint is attained. The process is repeated for the other endpoint.

Convergence is controlled by the CICONV= option in the MODEL statement. Suppose ϵ is the number specified in the CICONV= option. The default value of ϵ is 10^{-4}. Let the parameter of interest be β_j, and define $\mathbf{r} = \mathbf{u}_j$, the unit vector with a 1 in position j and 0s elsewhere. Convergence is declared on the current iteration if the following two conditions are satisfied:

$$|l^*(\beta_j) - l_0| \leq \epsilon$$
$$(\mathbf{s} + \lambda\mathbf{r})'\mathbf{H}^{-1}(\mathbf{s} + \lambda\mathbf{r}) \leq \epsilon$$

where $l^*(\beta_j)$, $\mathbf{s}$, and $\mathbf{H}$ are the log likelihood, the gradient, and the Hessian evaluated at the current parameter vector and λ is a constant computed by the procedure. The first condition for convergence means that the log-likelihood function must be within ϵ of the correct value, and the second condition means that the gradient vector must be proportional to the restriction vector $\mathbf{r}$.

When you specify the LRCI option in the MODEL statement, PROC GENMOD computes profile likelihood confidence intervals for all parameters in the model, including the scale parameter, if there is one. The interval endpoints are displayed in a table as well as the values of the remaining parameters at the solution.

Wald Confidence Intervals

You can request that PROC GENMOD produce Wald confidence intervals for the parameters. The $(1-\alpha)100\%$ Wald confidence interval for a parameter β is defined as

$$\hat{\beta} \pm z_{1-\alpha/2}\hat{\sigma}$$

where z_p is the $100p$th percentile of the standard normal distribution, $\hat{\beta}$ is the parameter estimate, and $\hat{\sigma}$ is the estimate of its standard error.

F Statistics

Suppose that D_0 is the deviance resulting from fitting a generalized linear model and that D_1 is the deviance from fitting a submodel. Then, under appropriate regularity conditions, the asymptotic distribution of $(D_1-D_0)/\phi$ is chi-square with r degrees of freedom, where r is the difference in the number of parameters between the two models and ϕ is the dispersion parameter. If ϕ is unknown, and $\hat{\phi}$ is an estimate of ϕ based on the deviance or Pearson's chi-square divided by degrees of freedom, then, under regularity conditions, $(n-p)\hat{\phi}/\phi$ has an asymptotic chi-square distribution with $n-p$ degrees of freedom. Here, n is the number of observations and p is the number of parameters in the model that is used to estimate ϕ. Thus, the asymptotic distribution of

$$F = \frac{D_1 - D_0}{r\hat{\phi}}$$

is the F distribution with r and $n-p$ degrees of freedom, assuming that $(D_1-D_0)/\phi$ and $(n-p)\hat{\phi}/\phi$ are approximately independent.

This F statistic is computed for the Type 1 analysis, Type 3 analysis, and hypothesis tests specified in CONTRAST statements when the dispersion parameter is estimated by either the deviance or Pearson's chi-square divided by degrees of freedom, as specified by the DSCALE or PSCALE option in the MODEL statement. In the case of a Type 1 analysis, model 0 is the higher-order model obtained by including one additional effect in model 1. For a Type 3 analysis and hypothesis tests, model 0 is the full specified model and model 1 is the submodel obtained from constraining the Type III contrast or the user-specified contrast to be 0.

Lagrange Multiplier Statistics

When you select the NOINT or NOSCALE option, restrictions are placed on the intercept or scale parameters. Lagrange multiplier, or score, statistics are computed in these cases. These statistics

assess the validity of the restrictions, and they are computed as

$$\chi^2 = \frac{s^2}{V}$$

where s is the component of the score vector evaluated at the restricted maximum corresponding to the restricted parameter and $V = \mathbf{I}_{11} - \mathbf{I}_{12}\mathbf{I}_{22}^{-1}\mathbf{I}_{21}$. The matrix $\mathbf{I}$ is the information matrix, 1 refers to the restricted parameter, and 2 refers to the rest of the parameters.

Under regularity conditions, this statistic has an asymptotic chi-square distribution with one degree of freedom, and p-values are computed based on this limiting distribution.

If you set $k = 0$ in a negative binomial model, s is the score statistic of Cameron and Trivedi (1998) for testing for overdispersion in a Poisson model against alternatives of the form $V(\mu) = \mu + k\mu^2$.

See Rao (1973, p. 417) for more details.

Predicted Values of the Mean

Predicted Values

A predicted value, or fitted value, of the mean μ_i corresponding to the vector of covariates $\mathbf{x}_i$ is given by

$$\hat{\mu_i} = g^{-1}(\mathbf{x}_i'\hat{\boldsymbol{\beta}})$$

where g is the link function, regardless of whether $\mathbf{x}_i$ corresponds to an observation or not. That is, the response variable can be missing and the predicted value is still computed for valid $\mathbf{x}_i$. In the case where $\mathbf{x}_i$ does not correspond to a valid observation, $\mathbf{x}_i$ is not checked for estimability. You should check the estimability of $\mathbf{x}_i$ in this case in order to ensure the uniqueness of the predicted value of the mean. If there is an offset, it is included in the predicted value computation.

Confidence Intervals on Predicted Values

Approximate confidence intervals for predicted values of the mean can be computed as follows. The variance of the linear predictor $\eta_i = \mathbf{x}_i'\hat{\boldsymbol{\beta}}$ is estimated by

$$\sigma_x^2 = \mathbf{x}_i'\boldsymbol{\Sigma}\mathbf{x}_i$$

where $\boldsymbol{\Sigma}$ is the estimated covariance of $\hat{\boldsymbol{\beta}}$. The robust estimate of the covariance is used for $\boldsymbol{\Sigma}$ in the case of models fit with GEEs.

Approximate $100(1-\alpha)\%$ confidence intervals are computed as

$$g^{-1}\left(\mathbf{x}_i'\hat{\beta} \pm z_{1-\alpha/2}\sigma_x\right)$$

where z_p is the $100p$th percentile of the standard normal distribution and g is the link function. If either endpoint in the argument is outside the valid range of arguments for the inverse link function, the corresponding confidence interval endpoint is set to missing.

Residuals

The GENMOD procedure computes three kinds of residuals. Residuals are available for all generalized linear models except multinomial models for ordinal response data, for which residuals are not available. Raw residuals and Pearson residuals are available for models fit with generalized estimating equations (GEEs).

The raw residual is defined as

$$r_i = y_i - \mu_i$$

where y_i is the ith response and μ_i is the corresponding predicted mean. You can request raw residuals in an output data set with the keyword RESRAW in the OUTPUT statement.

The Pearson residual is the square root of the ith contribution to the Pearson's chi-square:

$$r_{Pi} = (y_i - \mu_i)\sqrt{\frac{w_i}{V(\mu_i)}}$$

You can request Pearson residuals in an output data set with the keyword RESCHI in the OUTPUT statement.

Finally, the deviance residual is defined as the square root of the contribution of the ith observation to the deviance, with the sign of the raw residual:

$$r_{Di} = \sqrt{d_i}(\text{sign}(y_i - \mu_i))$$

You can request deviance residuals in an output data set with the keyword RESDEV in the OUTPUT statement.

The adjusted Pearson, deviance, and likelihood residuals are defined by Agresti (2002), Williams (1987), and Davison and Snell (1991). These residuals are useful for outlier detection and for assessing the influence of single observations on the fitted model.

For the generalized linear model, the variance of the ith individual observation is given by

$$v_i = \frac{\phi V(\mu_i)}{w_i}$$

where ϕ is the dispersion parameter, w_i is a user-specified prior weight (if not specified, $w_i = 1$), μ_i is the mean, and $V(\mu_i)$ is the variance function. Let

$$w_{ei} = v_i^{-1}(g'(\mu_i))^{-2}$$

for the ith observation, where $g'(\mu_i)$ is the derivative of the link function, evaluated at μ_i. Let $\mathbf{W}_e$ be the diagonal matrix with w_{ei} denoting the ith diagonal element. The weight matrix $\mathbf{W}_e$ is used in computing the expected information matrix.

Define h_i as the ith diagonal element of the matrix

$$\mathbf{W}_e^{\frac{1}{2}}\mathbf{X}(\mathbf{X}'\mathbf{W}_e\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}_e^{\frac{1}{2}}$$

The Pearson residuals, standardized to have unit asymptotic variance, are given by

$$r_{Pi} = \frac{y_i - \mu_i}{\sqrt{v_i(1-h_i)}}$$

You can request standardized Pearson residuals in an output data set with the keyword STDRESCHI in the OUTPUT statement. The deviance residuals, standardized to have unit asymptotic variance, are given by

$$r_{Di} = \frac{\text{sign}(y_i - \mu_i)\sqrt{d_i}}{\sqrt{\phi(1-h_i)}}$$

where d_i is the contribution to the total deviance from observation i, and $\text{sign}(y_i - \mu_i)$ is 1 if $y_i - \mu_i$ is positive and -1 if $y_i - \mu_i$ is negative. You can request standardized deviance residuals in an output data set with the keyword STDRESDEV in the OUTPUT statement. The likelihood residuals are defined by

$$r_{Gi} = \text{sign}(y_i - \mu_i)\sqrt{(1-h_i)r_{Di}^2 + h_i r_{Pi}^2}$$

You can request likelihood residuals in an output data set with the keyword RESLIK in the OUTPUT statement.

Multinomial Models

This type of model applies to cases where an observation can fall into one of k categories. Binary data occur in the special case where $k = 2$. If there are m_i observations in a subpopulation i, then the probability distribution of the number falling into the k categories $\mathbf{y}_i = (y_{i1}, y_{i2}, \cdots y_{ik})$ can be modeled by the multinomial distribution, defined in the section "Response Probability Distributions" on page 1964, with $\sum_j y_{ij} = m_i$. The multinomial model is an *ordinal* model if the categories have a natural order.

Residuals are not available in the OBSTATS table or the output data set for multinomial models.

By default, and consistently with binomial models, the GENMOD procedure orders the response categories for ordinal multinomial models from lowest to highest and models the probabilities of the lower response levels. You can change the way PROC GENMOD orders the response levels with the RORDER= option in the PROC GENMOD statement. The order that PROC GENMOD uses is shown in the "Response Profiles" output table described in the section "Response Profile" on page 2008.

The GENMOD procedure supports only the ordinal multinomial model. If $(p_{i1}, p_{i2}, \cdots p_{ik})$ are the category probabilities, the cumulative category probabilities are modeled with the same link functions used for binomial data. Let $P_{ir} = \sum_{j=1}^{r} p_{ij}$, $r = 1, 2, \cdots, k-1$, be the cumulative category probabilities (note that $P_{ik} = 1$). The ordinal model is

$$g(P_{ir}) = \mu_r + \mathbf{x}'\boldsymbol{\beta} \quad \text{for} \quad r = 1, 2, \cdots k-1$$

where $\mu_1, \mu_2, \cdots \mu_{k-1}$ are intercept terms that depend only on the categories and $\mathbf{x}_i$ is a vector of covariates that does not include an intercept term. The logit, probit, and complementary log-log link functions g are available. These are obtained by specifying the MODEL statement options

DIST=MULTINOMIAL and LINK=CUMLOGIT (cumulative logit), LINK=CUMPROBIT (cumulative probit), or LINK=CUMCLL (cumulative complementary log-log). Alternatively,

$$P_{ir} = \mathrm{F}(\mu_r + \mathbf{x}'\boldsymbol{\beta}) \quad \text{for} \quad r = 1, 2, \cdots k-1$$

where $\mathrm{F} = g^{-1}$ is a cumulative distribution function for the logistic, normal, or extreme-value distribution.

PROC GENMOD estimates the intercept parameters $\mu_1, \mu_2, \cdots \mu_{k-1}$ and regression parameters $\boldsymbol{\beta}$ by maximum likelihood.

The subpopulations i are defined by constant values of the AGGREGATE= variable. This has no effect on the parameter estimates, but it does affect the deviance and Pearson chi-square statistics; it also affects parameter estimate standard errors if you specify the SCALE=DEVIANCE or SCALE=PEARSON option.

Zero-Inflated Poisson Models

Count data that have an incidence of zero counts greater than expected for the Poisson distribution can be modeled with the zero-inflated Poisson distribution. See Long (1997) and Cameron and Trivedi (1998) for more information about zero-inflated Poisson models. The population is considered to consist of two types of individuals. The first type gives Poisson distributed counts, which might contain zeros. The second type always gives a zero count. Let λ be the Poisson mean and ω be the probability of an individual being of the second type. The parameter ω is called here the *zero-inflation probability*, and is the probability of zero counts in excess of the frequency predicted by the Poisson distribution. You can request that the zero inflation probability be displayed in an output data set with the PZERO keyword. The probability distribution of a zero-inflated random variable Y is given by

$$\Pr(\mathrm{Y} = y) = \begin{cases} \omega + (1-\omega)\mathrm{e}^{-\lambda} & \text{for } y = 0 \\ (1-\omega)\frac{\lambda^y \mathrm{e}^{-\lambda}}{y!} & \text{for } y = 1, 2, \ldots \end{cases}$$

You can model the parameters ω and λ in GENMOD with the regression models:

$$\begin{aligned} h(\omega_i) &= \mathbf{z}_i'\boldsymbol{\gamma} \\ g(\lambda_i) &= \mathbf{x}_i'\boldsymbol{\beta} \end{aligned}$$

where h is one of the binary link functions: logit, probit, or complementary log-log. The link function h is the logit link by default, or the link function option specified in the ZEROMODEL statement. The link function for the Poisson part of the model, g, is the log link function by default, or the link function specified in the MODEL statement. The covariates $\mathbf{z}_i$ for observation i are determined by the model specified in the ZEROMODEL statement, and the covariates $\mathbf{x}_i$ are determined by the model specified in the MODEL statement. The regression parameters $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are estimated by maximum likelihood.

The mean and variance of Y are given by

$$\begin{aligned} \mathrm{E}(Y) &= \mu = (1-\omega)\lambda \\ \mathrm{Var}(Y) &= \mu + \frac{\omega}{1-\omega}\mu^2 \end{aligned}$$

You can request that the mean of Y be displayed for each observation in an output data set with the PRED keyword.

Generalized Estimating Equations

Let y_{ij}, $j = 1, \ldots, n_i$, $i = 1, \ldots, K$, represent the jth measurement on the ith subject. There are n_i measurements on subject i and $\sum_{i=1}^{K} n_i$ total measurements.

Correlated data are modeled using the same link function and linear predictor setup (systematic component) as the independence case. The random component is described by the same variance functions as in the independence case, but the covariance structure of the correlated measurements must also be modeled. Let the vector of measurements on the ith subject be $\mathbf{Y}_i = [y_{i1}, \ldots, y_{in_i}]'$ with corresponding vector of means $\boldsymbol{\mu}_i = [\mu_{i1}, \ldots, \mu_{in_i}]'$, and let $\mathbf{V}_i$ be the covariance matrix of $\mathbf{Y}_i$. Let the vector of independent, or explanatory, variables for the jth measurement on the ith subject be

$$\mathbf{x}_{ij} = [x_{ij1}, \ldots, x_{ijp}]'$$

The generalized estimating equation of Liang and Zeger (1986) for estimating the $p \times 1$ vector of regression parameters $\boldsymbol{\beta}$ is an extension of the independence estimating equation to correlated data and is given by

$$\mathbf{S}(\boldsymbol{\beta}) = \sum_{i=1}^{K} \mathbf{D}_i' \mathbf{V}_i^{-1} (\mathbf{Y}_i - \boldsymbol{\mu}_i(\boldsymbol{\beta})) = \mathbf{0}$$

where

$$\mathbf{D}_i = \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}}$$

Since

$$g(\mu_{ij}) = \mathbf{x}_{ij}{}'\boldsymbol{\beta}$$

where g is the link function, the $p \times n_i$ matrix of partial derivatives of the mean with respect to the regression parameters for the ith subject is given by

$$\mathbf{D}_i' = \frac{\partial \boldsymbol{\mu}_i'}{\partial \boldsymbol{\beta}} = \begin{bmatrix} \frac{x_{i11}}{g'(\mu_{i1})} & \cdots & \frac{x_{in_i1}}{g'(\mu_{in_i})} \\ \vdots & & \vdots \\ \frac{x_{i1p}}{g'(\mu_{i1})} & \cdots & \frac{x_{in_ip}}{g'(\mu_{in_i})} \end{bmatrix}$$

Working Correlation Matrix

Let $\mathbf{R}_i(\boldsymbol{\alpha})$ be an $n_i \times n_i$ "working" correlation matrix that is fully specified by the vector of parameters $\boldsymbol{\alpha}$. The covariance matrix of $\mathbf{Y}_i$ is modeled as

$$\mathbf{V}_i = \phi \mathbf{A}_i^{\frac{1}{2}} \mathbf{W}_i^{-\frac{1}{2}} \mathbf{R}(\boldsymbol{\alpha}) \mathbf{W}_i^{-\frac{1}{2}} \mathbf{A}_i^{\frac{1}{2}}$$

where $\mathbf{A}_i$ is an $n_i \times n_i$ diagonal matrix with $v(\mu_{ij})$ as the jth diagonal element and $\mathbf{W}_i$ is an $n_i \times n_i$ diagonal matrix with w_{ij} as the jth diagonal, where w_{ij} is a weight specified with the WEIGHT statement. If there is no WEIGHT statement, $w_{ij} = 1$ for all i and j. If $\mathbf{R}_i(\boldsymbol{\alpha})$ is the true correlation matrix of $\mathbf{Y}_i$, then $\mathbf{V}_i$ is the true covariance matrix of $\mathbf{Y}_i$.

The working correlation matrix is usually unknown and must be estimated. It is estimated in the iterative fitting process by using the current value of the parameter vector $\boldsymbol{\beta}$ to compute appropriate functions of the Pearson residual

$$e_{ij} = \frac{y_{ij} - \mu_{ij}}{\sqrt{v(\mu_{ij})/w_{ij}}}$$

If you specify the working correlation as $\mathbf{R_0} = \mathbf{I}$, which is the identity matrix, the GEE reduces to the independence estimating equation.

Following are the structures of the working correlation supported by the GENMOD procedure and the estimators used to estimate the working correlations.

Working Correlation Structure	Estimator
Fixed $\text{Corr}(Y_{ij}, Y_{ik}) = r_{jk}$ where r_{jk} is the jkth element of a constant, user-specified correlation matrix $\mathbf{R}_0$.	The working correlation is not estimated in this case.
Independent $\text{Corr}(Y_{ij}, Y_{ik}) = \begin{cases} 1 & j = k \\ 0 & j \neq k \end{cases}$	The working correlation is not estimated in this case.
***m*-dependent** $\text{Corr}(Y_{ij}, Y_{i,j+t}) = \begin{cases} 1 & t = 0 \\ \alpha_t & t = 1, 2, \ldots, m \\ 0 & t > m \end{cases}$	$\hat{\alpha}_t = \frac{1}{(K_t - p)\phi} \sum_{i=1}^{K} \sum_{j \leq n_i - t} e_{ij} e_{i,j+t}$ $K_t = \sum_{i=1}^{K} (n_i - t)$
Exchangeable $\text{Corr}(Y_{ij}, Y_{ik}) = \begin{cases} 1 & j = k \\ \alpha & j \neq k \end{cases}$	$\hat{\alpha} = \frac{1}{(N^* - p)\phi} \sum_{i=1}^{K} \sum_{j<k} e_{ij} e_{ik}$ $N^* = 0.5 \sum_{i=1}^{K} n_i (n_i - 1)$
Unstructured $\text{Corr}(Y_{ij}, Y_{ik}) = \begin{cases} 1 & j = k \\ \alpha_{jk} & j \neq k \end{cases}$	$\hat{\alpha}_{jk} = \frac{1}{(K - p)\phi} \sum_{i=1}^{K} e_{ij} e_{ik}$
Autoregressive **AR(1)** $\text{Corr}(Y_{ij}, Y_{i,j+t}) = \alpha^t$ for $t = 0, 1, 2, \ldots, n_i - j$	$\hat{\alpha} = \frac{1}{(K_1 - p)\phi} \sum_{i=1}^{K} \sum_{j \leq n_i - 1} e_{ij} e_{i,j+1}$ $K_1 = \sum_{i=1}^{K} (n_i - 1)$

Dispersion Parameter

The dispersion parameter ϕ is estimated by

$$\hat{\phi} = \frac{1}{N - p} \sum_{i=1}^{K} \sum_{j=1}^{n_i} e_{ij}^2$$

where $N = \sum_{i=1}^{K} n_i$ is the total number of measurements and p is the number of regression parameters.

The square root of $\hat{\phi}$ is reported by PROC GENMOD as the scale parameter in the "Analysis of GEE Parameter Estimates Model-Based Standard Error Estimates" output table. If a fixed scale parameter is specified with the NOSCALE option in the MODEL statement, then the fixed value is used in estimating the model-based covariance matrix and standard errors.

Fitting Algorithm

The following is an algorithm for fitting the specified model by using GEEs. Note that this is not in general a likelihood-based method of estimation, so that inferences based on likelihoods are not possible for GEE methods.

1. Compute an initial estimate of $\boldsymbol{\beta}$ with an ordinary generalized linear model assuming independence.
2. Compute the working correlations $\mathbf{R}$ based on the standardized residuals, the current $\boldsymbol{\beta}$, and the assumed structure of $\mathbf{R}$.
3. Compute an estimate of the covariance:

$$\mathbf{V}_i = \phi \mathbf{A}_i^{\frac{1}{2}} \mathbf{W}_i^{-\frac{1}{2}} \hat{\mathbf{R}}(\boldsymbol{\alpha}) \mathbf{W}_i^{-\frac{1}{2}} \mathbf{A}_i^{\frac{1}{2}}$$

4. Update $\boldsymbol{\beta}$:

$$\boldsymbol{\beta}_{r+1} = \boldsymbol{\beta}_r + \left[\sum_{i=1}^{K} \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}}' \mathbf{V}_i^{-1} \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}} \right]^{-1} \left[\sum_{i=1}^{K} \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}}' \mathbf{V}_i^{-1} (\mathbf{Y}_i - \boldsymbol{\mu}_i) \right]$$

5. Repeat steps 2-4 until convergence.

Missing Data

See Diggle, Liang, and Zeger (1994, Chapter 11) for a discussion of missing values in longitudinal data. Suppose that you intend to take measurements $Y_{i1}, \ldots, Y_{in}$ for the ith unit. Missing values for which Y_{ij} are missing whenever Y_{ik} is missing for all $j \geq k$ are called *dropouts*. Otherwise, missing values that occur intermixed with nonmissing values are *intermittent* missing values. The GENMOD procedure can estimate the working correlation from data containing both types of missing values by using the *all available pairs* method, in which all nonmissing pairs of data are used in the moment estimators of the working correlation parameters defined previously. The resulting covariances and standard errors are valid under the missing completely at random (MCAR) assumption.

For example, for the unstructured working correlation model,

$$\hat{\alpha}_{jk} = \frac{1}{(K' - p)\phi} \sum e_{ij} e_{ik}$$

where the sum is over the units that have nonmissing measurements at times j and k, and K' is the number of units with nonmissing measurements at j and k. Estimates of the parameters for other working correlation types are computed in a similar manner, using available nonmissing pairs in the appropriate moment estimators.

The contribution of the ith unit to the parameter update equation is computed by omitting the elements of $(\mathbf{Y}_i - \boldsymbol{\mu}_i)$, the columns of $\mathbf{D}_i' = \frac{\partial \boldsymbol{\mu}}{\partial \boldsymbol{\beta}}'$, and the rows and columns of $\mathbf{V}_i$ corresponding to missing measurements.

Parameter Estimate Covariances

The *model-based* estimator of $\mathrm{Cov}(\hat{\boldsymbol{\beta}})$ is given by

$$\boldsymbol{\Sigma}_m(\hat{\boldsymbol{\beta}}) = \mathbf{I}_0^{-1}$$

where

$$\mathbf{I}_0 = \sum_{i=1}^{K} \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}}' \mathbf{V}_i^{-1} \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}}$$

This is the GEE equivalent of the inverse of the Fisher information matrix that is often used in generalized linear models as an estimator of the covariance estimate of the maximum likelihood estimator of $\boldsymbol{\beta}$. It is a consistent estimator of the covariance matrix of $\hat{\boldsymbol{\beta}}$ if the mean model and the working correlation matrix are correctly specified.

The estimator

$$\boldsymbol{\Sigma}_e = \mathbf{I}_0^{-1}\mathbf{I}_1\mathbf{I}_0^{-1}$$

is called the *empirical*, or *robust*, estimator of the covariance matrix of $\hat{\boldsymbol{\beta}}$, where

$$\mathbf{I}_1 = \sum_{i=1}^{K} \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}}' \mathbf{V}_i^{-1}\mathrm{Cov}(\mathbf{Y}_i)\mathbf{V}_i^{-1} \frac{\partial \boldsymbol{\mu}_i}{\partial \boldsymbol{\beta}}$$

It has the property of being a consistent estimator of the covariance matrix of $\hat{\boldsymbol{\beta}}$, even if the working correlation matrix is misspecified—that is, if $\mathrm{Cov}(\mathbf{Y}_i) \neq \mathbf{V}_i$. See Zeger, Liang, and Albert (1988)), Royall (1986), and White (1982) for further information about the robust variance estimate. In computing $\boldsymbol{\Sigma}_e$, $\boldsymbol{\beta}$ and ϕ are replaced by estimates, and $\mathrm{Cov}(\mathbf{Y}_i)$ is replaced by the estimate

$$(\mathbf{Y}_i - \boldsymbol{\mu}_i(\hat{\boldsymbol{\beta}}))(\mathbf{Y}_i - \boldsymbol{\mu}_i(\hat{\boldsymbol{\beta}}))'$$

Multinomial GEEs

Lipsitz, Kim, and Zhao (1994) and Miller, Davis, and Landis (1993) describe how to extend GEEs to multinomial data. Currently, only the independent working correlation is available for multinomial models in PROC GENMOD.

Alternating Logistic Regressions

If the responses are binary (that is, they take only two values), then there is an alternative method to account for the association among the measurements. The alternating logistic regressions (ALR) algorithm of Carey, Zeger, and Diggle (1993) models the association between pairs of responses with log odds ratios, instead of with correlations, as ordinary GEEs do.

For binary data, the correlation between the jth and kth response is, by definition,

$$\mathrm{Corr}(Y_{ij}, Y_{ik}) = \frac{\Pr(Y_{ij} = 1, Y_{ik} = 1) - \mu_{ij}\mu_{ik}}{\sqrt{\mu_{ij}(1-\mu_{ij})\mu_{ik}(1-\mu_{ik})}}$$

The joint probability in the numerator satisfies the following bounds, by elementary properties of probability, since $\mu_{ij} = \Pr(Y_{ij} = 1)$:

$$\max(0, \mu_{ij} + \mu_{ik} - 1) \leq \Pr(Y_{ij} = 1, Y_{ik} = 1) \leq \min(\mu_{ij}, \mu_{ik})$$

The correlation, therefore, is constrained to be within limits that depend in a complicated way on the means of the data.

The odds ratio, defined as

$$\mathrm{OR}(Y_{ij}, Y_{ik}) = \frac{\Pr(Y_{ij} = 1, Y_{ik} = 1)\Pr(Y_{ij} = 0, Y_{ik} = 0)}{\Pr(Y_{ij} = 1, Y_{ik} = 0)\Pr(Y_{ij} = 0, Y_{ik} = 1)}$$

is not constrained by the means and is preferred, in some cases, to correlations for binary data.

The ALR algorithm seeks to model the logarithm of the odds ratio, $\gamma_{ijk} = \log(\mathrm{OR}(Y_{ij}, Y_{ik}))$, as

$$\gamma_{ijk} = \mathbf{z}'_{ijk}\boldsymbol{\alpha}$$

where $\boldsymbol{\alpha}$ is a $q \times 1$ vector of regression parameters and $\mathbf{z}_{ijk}$ is a fixed, specified vector of coefficients.

The parameter γ_{ijk} can take any value in $(-\infty, \infty)$ with $\gamma_{ijk} = 0$ corresponding to no association.

The log odds ratio, when modeled in this way with a regression model, can take different values in subgroups defined by $\mathbf{z}_{ijk}$. For example, $\mathbf{z}_{ijk}$ can define subgroups within clusters, or it can define "block effects" between clusters.

You specify a GEE model for binary data that uses log odds ratios by specifying a model for the mean, as in ordinary GEEs, and a model for the log odds ratios. You can use any of the link functions appropriate for binary data in the model for the mean, such as logistic, probit, or complementary log-log. The ALR algorithm alternates between a GEE step to update the model for the mean and a logistic regression step to update the log odds ratio model. Upon convergence, the ALR algorithm provides estimates of the regression parameters for the mean, $\boldsymbol{\beta}$, the regression parameters for the log odds ratios, $\boldsymbol{\alpha}$, their standard errors, and their covariances.

Specifying Log Odds Ratio Models

Specifying a regression model for the log odds ratio requires you to specify rows of the z-matrix $\mathbf{z}_{ijk}$ for each cluster i and each unique within-cluster pair (j, k). The GENMOD procedure provides several methods of specifying $\mathbf{z}_{ijk}$. These are controlled by the LOGOR=*keyword* and associated options in the REPEATED statement. The supported keywords and the resulting log odds ratio models are described as follows.

EXCH — specifies exchangeable log odds ratios. In this model, the log odds ratio is a constant for all clusters i and pairs (j, k). The parameter α is the common log odds ratio.

$$\mathbf{z}_{ijk} = 1 \quad \text{for all} \quad i, j, k$$

FULLCLUST — specifies fully parameterized clusters. Each cluster is parameterized in the same way, and there is a parameter for each unique pair within clusters. If a complete cluster is of size n, then there are $\frac{n(n-1)}{2}$ parameters

in the vector $\boldsymbol{\alpha}$. For example, if a full cluster is of size 4, then there are $\frac{4\times3}{2} = 6$ parameters, and the z-matrix is of the form

$$Z = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The elements of $\boldsymbol{\alpha}$ correspond to log odds ratios for cluster pairs in the following order:

Pair	Parameter
(1,2)	Alpha1
(1,3)	Alpha2
(1,4)	Alpha3
(2.3)	Alpha4
(2,4)	Alpha5
(3,4)	Alpha6

LOGORVAR(*variable*) specifies log odds ratios by cluster. The argument *variable* is a variable name that defines the "block effects" between clusters. The log odds ratios are constant within clusters, but they take a different value for each different value of the *variable*. For example, if Center is a variable in the input data set taking a different value for k treatment centers, then specifying LOGOR=LOGORVAR(Center) requests a model with different log odds ratios for each of the k centers, constant within center.

NESTK specifies k-nested log odds ratios. You must also specify the SUBCLUST=*variable* option to define subclusters within clusters. Within each cluster, PROC GENMOD computes a log odds ratio parameter for pairs having the same value of *variable* for both members of the pair and one log odds ratio parameter for each unique combination of different values of *variable*.

NEST1 specifies 1-nested log odds ratios. You must also specify the SUBCLUST=*variable* option to define subclusters within clusters. There are two log odds ratio parameters for this model. Pairs having the same value of *variable* correspond to one parameter; pairs having different values of *variable* correspond to the other parameter. For example, if clusters are hospitals and subclusters are wards within hospitals, then patients within the same ward have one log odds ratio parameter, and patients from different wards have the other parameter.

ZFULL specifies the full z-matrix. You must also specify a SAS data set containing the z-matrix with the ZDATA=*data-set-name* option. Each observation in the data set corresponds to one row of the z-matrix. You must specify the ZDATA data set as if all clusters are complete—that is, as if all clusters are the same size and there are no missing observations. The ZDATA data set has $K[n_{max}(n_{max} - 1)/2]$ observations, where K is the number of clusters and n_{max} is the maximum

cluster size. If the members of cluster i are ordered as $1, 2, \cdots, n$, then the rows of the z-matrix must be specified for pairs in the order $(1,2), (1,3), \cdots, (1,n), (2,3), \cdots, (2,n), \cdots, (n-1,n)$. The variables specified in the REPEATED statement for the SUBJECT effect must also be present in the ZDATA= data set to identify clusters. You must specify variables in the data set that define the columns of the z-matrix by the ZROW=*variable-list* option. If there are q columns (q variables in *variable-list*), then there are q log odds ratio parameters. You can optionally specify variables indicating the cluster pairs corresponding to each row of the z-matrix with the YPAIR=(*variable1, variable2*) option. If you specify this option, the data from the ZDATA data set are sorted within each cluster by *variable1* and *variable2*. See Example 37.6 for an example of specifying a full z-matrix.

ZREP

specifies a replicated z-matrix. You specify z-matrix data exactly as you do for the ZFULL case, except that you specify only one complete cluster. The z-matrix for the one cluster is replicated for each cluster. The number of observations in the ZDATA data set is $\frac{n_{max}(n_{max}-1)}{2}$, where n_{max} is the size of a complete cluster (a cluster with no missing observations).

ZREP(matrix)

specifies direct input of the replicated z-matrix. You specify the z-matrix for one cluster with the syntax LOGOR=ZREP ($(y_1 \;\; y_2) z_1 \;\; z_2 \cdots z_q, \cdots$), where y_1 and y_2 are numbers representing a pair of observations and the values $z_1, z_2, \cdots, z_q$ make up the corresponding row of the z-matrix. The number of rows specified is $\frac{n_{max}(n_{max}-1)}{2}$, where n_{max} is the size of a complete cluster (a cluster with no missing observations). For example,

```
LOGOR =  ZREP((1 2) 1 0,
              (1 3) 1 0,
              (1 4) 1 0,
              (2 3) 1 1,
              (2 4) 1 1,
              (3 4) 1 1)
```

specifies the $\frac{4\times3}{2} = 6$ rows of the z-matrix for a cluster of size 4 with $q = 2$ log odds ratio parameters. The log odds ratio for the pairs (1 2), (1 3), (1 4) is α_1, and the log odds ratio for the pairs (2 3), (2 4), (3 4) is $\alpha_1 + \alpha_2$.

Quasi-likelihood Information Criterion

The quasi-likelihood information criterion (QIC) was developed by Pan (2001) as a modification of the Akaike information criterion (AIC) to apply to models fit by GEEs.

Define the quasi-likelihood under the independence working correlation assumption, evaluated with the parameter estimates under the working correlation of interest as

$$Q(\hat{\boldsymbol{\beta}}(R), \phi) = \sum_{i=1}^{K} \sum_{j=1}^{n_i} Q(\hat{\boldsymbol{\beta}}(R), \phi; (Y_{ij}, \mathbf{X}_{ij}))$$

where the quasi-likelihood contribution of the jth observation in the ith cluster is defined in the section "Quasi-likelihood Functions" on page 1994 and $\hat{\boldsymbol{\beta}}(R)$ are the parameter estimates obtained from GEEs with the working correlation of interest R.

QIC is defined as

$$QIC(R) = -2Q(\hat{\boldsymbol{\beta}}(R), \phi) + 2\text{trace}(\hat{\Omega}_I \hat{V}_R)$$

where $\hat{V}_R$ is the robust covariance estimate and $\hat{\Omega}_I$ is the inverse of the model-based covariance estimate under the independent working correlation assumption, evaluated at $\hat{\boldsymbol{\beta}}(R)$, the parameter estimates obtained from GEEs with the working correlation of interest R.

PROC GENMOD also computes an approximation to $QIC(R)$ defined by Pan (2001) as

$$QIC_u(R) = -2Q(\hat{\boldsymbol{\beta}}(R), \phi) + 2p$$

where p is the number of regression parameters.

Pan (2001) notes that QIC is appropriate for selecting regression models and working correlations, whereas QIC_u is appropriate only for selecting regression models.

Quasi-likelihood Functions

See McCullagh and Nelder (1989) and Hardin and Hilbe (2003) for discussions of quasi-likelihood functions. The contribution of observation j in cluster i to the quasi-likelihood function evaluated at the regression parameters $\boldsymbol{\beta}$ is given by $Q(\boldsymbol{\beta}, \phi; (Y_{ij}, \mathbf{X}_{ij})) = \frac{Q_{ij}}{\phi}$, where Q_{ij} is defined in the following list. These are used in the computation of the quasi-likelihood information criteria (QIC) for goodness of fit of models fit with GEEs. The w_{ij} are prior weights, if any, specified with the WEIGHT or FREQ statements. Note that the definition of the quasi-likelihood for the negative binomial differs from that given in McCullagh and Nelder (1989). The definition used here allows the negative binomial quasi-likelihood to approach the Poisson as $k \to 0$.

- Normal:

$$Q_{ij} = -\frac{1}{2} w_{ij} (y_{ij} - \mu_{ij})^2$$

- Inverse Gaussian:

$$Q_{ij} = \frac{w_{ij}(\mu_{ij} - .5y_{ij})}{\mu_{ij}^2}$$

- Gamma:

$$Q_{ij} = -w_{ij} \left[\frac{y_{ij}}{\mu_{ij}} + \log(\mu_{ij}) \right]$$

- Negative binomial:

$$Q_{ij} = w_{ij} [\log \Gamma(y_{ij} + \frac{1}{k}) - \log \Gamma(\frac{1}{k}) + y_{ij} \log \left(\frac{k\mu_{ij}}{1 + k\mu_{ij}} \right) + \frac{1}{k} \log \left(\frac{1}{1 + k\mu_{ij}} \right)]$$

- Poisson:

$$Q_{ij} = w_{ij}(y_{ij}\log(\mu_{ij}) - \mu_{ij})$$

- Binomial:

$$Q_{ij} = w_{ij}[r_{ij}\log(p_{ij}) + (n_{ij} - r_{ij})\log(1 - p_{ij})]$$

- Multinomial (s categories):

$$Q_{ij} = w_{ij}\sum_{k=1}^{s} y_{ijk}\log(\mu_{ijk})$$

Generalized Score Statistics

Boos (1992) and Rotnitzky and Jewell (1990) describe score tests applicable to testing $\mathbf{L}'\boldsymbol{\beta} = \mathbf{0}$ in GEEs, where $\mathbf{L}'$ is a user-specified $r \times p$ contrast matrix or a contrast for a Type 3 test of hypothesis.

Let $\tilde{\boldsymbol{\beta}}$ be the regression parameters resulting from solving the GEE under the restricted model $\mathbf{L}'\boldsymbol{\beta} = \mathbf{0}$, and let $\mathbf{S}(\tilde{\boldsymbol{\beta}})$ be the generalized estimating equation values at $\tilde{\boldsymbol{\beta}}$.

The generalized score statistic is

$$T = \mathbf{S}(\tilde{\boldsymbol{\beta}})'\boldsymbol{\Sigma}_m\mathbf{L}(\mathbf{L}'\boldsymbol{\Sigma}_e\mathbf{L})^{-1}\mathbf{L}'\boldsymbol{\Sigma}_m\mathbf{S}(\tilde{\boldsymbol{\beta}})$$

where $\boldsymbol{\Sigma}_m$ is the model-based covariance estimate and $\boldsymbol{\Sigma}_e$ is the empirical covariance estimate. The p-values for T are computed based on the chi-square distribution with r degrees of freedom.

Assessment of Models Based on Aggregates of Residuals

Lin, Wei, and Ying (2002) present graphical and numerical methods for model assessment based on the cumulative sums of residuals over certain coordinates (such as covariates or linear predictors) or some related aggregates of residuals. The distributions of these stochastic processes under the assumed model can be approximated by the distributions of certain zero-mean Gaussian processes whose realizations can be generated by simulation. Each observed residual pattern can then be compared, both graphically and numerically, with a number of realizations from the null distribution. Such comparisons enable you to assess objectively whether the observed residual pattern reflects anything beyond random fluctuation. These procedures are useful in determining appropriate functional forms of covariates and link function. You use the ASSESS|ASSESSMENT statement to perform this kind of model-checking with cumulative sums of residuals, moving sums of residuals, or LOESS smoothed residuals. See Example 37.8 and Example 37.9 for examples of model assessment.

Let the model for the mean be

$$g(\mu_i) = \mathbf{x}_i'\boldsymbol{\beta}$$

where μ_i is the mean of the response y_i and $\mathbf{x}_i$ is the vector of covariates for the ith observation. Denote the raw residual resulting from fitting the model as

$$e_i = y_i - \hat{\mu}_i$$

and let x_{ij} be the value of the jth covariate in the model for observation i. Then to check the functional form of the jth covariate, consider the cumulative sum of residuals with respect to x_{ij},

$$W_j(x) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} I(x_{ij} \leq x) e_i$$

where $I()$ is the indicator function. For any x, $W_j(x)$ is the sum of the residuals with values of x_j less than or equal to x.

Denote the score, or gradient vector, by

$$U(\boldsymbol{\beta}) = \sum_{i=1}^{n} h(\mathbf{x}'\boldsymbol{\beta}) \mathbf{x}_i (y_i - \nu(\mathbf{x}'\boldsymbol{\beta}))$$

where $\nu(r) = g^{-1}(r)$, and

$$h(r) = \frac{1}{g'(\nu(r)) V(\nu(r))}$$

Let J be the Fisher information matrix

$$J(\boldsymbol{\beta}) = -\frac{\partial U(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}'}$$

Define

$$\hat{W}_j(x) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} [I(x_{ij} \leq x) + \boldsymbol{\eta}'(x; \hat{\boldsymbol{\beta}}) J^{-1}(\hat{\boldsymbol{\beta}}) \mathbf{x}_i h(\mathbf{x}'\hat{\boldsymbol{\beta}})] e_i Z_i$$

where

$$\boldsymbol{\eta}(x; \boldsymbol{\beta}) = -\sum_{i=1}^{n} I(x_{ij} \leq x) \frac{\partial \nu(\mathbf{x}_i'\boldsymbol{\beta})}{\partial \boldsymbol{\beta}}$$

and Z_i are independent $N(0, 1)$ random variables. Then the conditional distribution of $\hat{W}_j(x)$, given $(y_i, \mathbf{x}_i), i = 1, \ldots, n$, under the null hypothesis H_0 that the model for the mean is correct, is the same asymptotically as $n \to \infty$ as the unconditional distribution of $W_j(x)$ (Lin, Wei, and Ying 2002).

You can approximate realizations from the null hypothesis distribution of $W_j(x)$ by repeatedly generating normal samples $Z_i, i = 1, \ldots, n$, while holding $(y_i, \mathbf{x}_i), i = 1, \ldots, n$, at their observed values and computing $\hat{W}_j(x)$ for each sample.

You can assess the functional form of covariate j by plotting a few realizations of $\hat{W}_j(x)$ on the same plot as the observed $W_j(x)$ and visually comparing to see how typical the observed $W_j(x)$ is of the null distribution samples.

You can supplement the graphical inspection method with a Kolmogorov-type supremum test. Let s_j be the observed value of $S_j = \sup_x |W_j(x)|$. The p-value $\Pr[S_j \geq s_j]$ is approximated by $\Pr[\hat{S}_j \geq s_j]$, where $\hat{S}_j = \sup_x |\hat{W}_j(x)|$. $\Pr[\hat{S}_j \geq s_j]$ is estimated by generating realizations of $\hat{W}_j(.)$ (1,000 is the default number of realizations).

You can check the link function instead of the jth covariate by using values of the linear predictor $\mathbf{x}_i'\hat{\boldsymbol{\beta}}$ in place of values of the jth covariate x_{ij}. The graphical and numerical methods described previously are then sensitive to inadequacies in the link function.

An alternative aggregate of residuals is the moving sum statistic

$$W_j(x,b) = \frac{1}{\sqrt{n}} \sum_{i=1}^{n} I(x - b \leq x_{ij} \leq x) e_i$$

If you specify the keyword WINDOW(b), then the moving sum statistic with window size b is used instead of the cumulative sum of residuals, with $I(x - b \leq x_{ij} \leq x)$ replacing $I(x_{ij} \leq x)$ in the earlier equation.

If you specify the keyword LOESS(f), loess smoothed residuals are used in the preceding formulas, where f is the fraction of the data to be used at a given point. If f is not specified, $f = \frac{1}{3}$ is used. For data $(Y_i, X_i), i = 1, \ldots, n$, define r as the nearest integer to nf and h as the rth smallest among $|X_i - x|, i = 1, \ldots, n$. Let

$$K_i(x) = K\left(\frac{X_i - x}{h}\right)$$

where

$$K(t) = \frac{70}{81}(1 - |t|^3)^3 I(-1 \leq t \leq 1)$$

Define

$$w_i(x) = K_i(x)[S_2(x) - (X_i - x)S_1(x)]$$

where

$$S_1(x) = \sum_{i=1}^{n} K_i(x)(X_i - x)$$

$$S_2(x) = \sum_{i=1}^{n} K_i(x)(X_i - x)^2$$

Then the loess estimate of Y at x is defined by

$$\hat{Y}(x) = \sum_{i=1}^{n} \frac{w_i(x)}{\sum_{i=1}^{n} w_i(x)} Y_i$$

Loess smoothed residuals for checking the functional form of the jth covariate are defined by replacing Y_i with e_i and X_i with x_{ij}. To implement the graphical and numerical assessment methods, $I(x_{ij} \le x)$ is replaced with $\frac{w_i(x)}{\sum_{i=1}^{n} w_i(x)}$ in the formulas for $W_j(x)$ and $\hat{W}_j(x)$.

You can perform the model checking described earlier for marginal models for dependent responses fit by generalized estimating equations (GEEs). Let y_{ik} denote the kth measurement on the ith cluster, $i = 1, \ldots, K$, $k = 1, \ldots, n_i$, and let $\mathbf{x}_{ik}$ denote the corresponding vector of covariates. The marginal mean of the response $\mu_{ik} = \mathrm{E}(y_{ik})$ is assumed to depend on the covariate vector by

$$g(\mu_{ik}) = \mathbf{x}_{ik}'\boldsymbol{\beta}$$

where g is the link function.

Define the vector of residuals for the ith cluster as

$$\mathbf{e}_i = (e_{i1}, \ldots, e_{in_i})' = (y_{i1} - \hat{\mu}_{i1}, \ldots, y_{in_i} - \hat{\mu}_{in_i})'$$

You use the following extension of $W_j(x)$ defined earlier to check the functional form of the jth covariate:

$$W_j(x) = \frac{1}{\sqrt{K}} \sum_{i=1}^{K} \sum_{k=1}^{n_i} I(x_{ikj} \le x) e_{ik}$$

where x_{ikj} is the jth component of $\mathbf{x}_{ik}$.

The null distribution of $W_j(x)$ can be approximated by the conditional distribution of

$$\hat{W}_j(x) = \frac{1}{\sqrt{K}} \sum_{i=1}^{K} \left\{ \sum_{k=1}^{n_i} I(x_{ikj} \le x) e_{ik} + \boldsymbol{\eta}'(x, \hat{\boldsymbol{\beta}}) \mathbf{I}_0^{-1} \hat{\mathbf{D}}_i' \hat{\mathbf{V}}_i^{-1} \mathbf{e}_i \right\} Z_i$$

where $\hat{\mathbf{D}}_i$ and $\hat{\mathbf{V}}_i$ are defined as in the section "Generalized Estimating Equations" on page 1986 with the unknown parameters replaced by their estimated values,

$$\boldsymbol{\eta}(x, \boldsymbol{\beta}) = -\sum_{i=1}^{K} \sum_{k=1}^{n_i} I(x_{ikj} \le x) \frac{\partial \mu_{ik}}{\partial \boldsymbol{\beta}}$$

$$\mathbf{I}_0 = \sum_{i=1}^{K} \hat{\mathbf{D}}_i' \hat{\mathbf{V}}_i^{-1} \hat{\mathbf{D}}_i$$

and $Z_i, i = 1, \ldots, K$, are independent $N(0, 1)$ random variables. You replace x_{ikj} with the linear predictor $\mathbf{x}_{ik}'\hat{\boldsymbol{\beta}}$ in the preceding formulas to check the link function.

Case Deletion Diagnostic Statistics

For ordinary generalized linear models, regression diagnostic statistics developed by Williams (1987) can be requested in an output data set or in the OBSTATS table by specifying the DIAGNOSTICS | INFLUENCE option in the MODEL statement. These diagnostics measure the influence of an individual observation on model fit, and generalize the one-step diagnostics developed by Pregibon (1981) for the logistic regression model for binary data.

Preisser and Qaqish (1996) further generalize regression diagnostics to apply to models for correlated data fit by generalized estimating equations (GEEs), where the influence of entire clusters of correlated observations is measured. These diagnostic statistics can be requested in an output data set or in the OBSTATS table if a model for correlated data is specified with a REPEATED statement.

The next two sections use the following notation:

- $\hat{\boldsymbol{\beta}}$ is the maximum likelihood estimate of the regression parameters $\boldsymbol{\beta}$, or, in the case of correlated data, the solution of the GEEs.
- $\hat{\boldsymbol{\beta}}_{[i]}$ is the corresponding estimate evaluated with the ith observation deleted, or, in the case of correlated data, with the ith cluster deleted.
- p is the dimension of the regression parameter vector $\boldsymbol{\beta}$.
- r_{pi} is the standardized Pearson residual $\frac{y_i-\mu_i}{\sqrt{v_i(1-h_i)}}$, where v_i is the variance of the ith response and h_i is the leverage defined in the section "H | LEVERAGE" on page 2000.
- v_i is the variance of response i, $\text{var}(Y_i) = \phi V(\mu_i)$, where $V(\mu)$ is the variance function and ϕ is the dispersion parameter.
- w_i is the prior weight of the ith observation specified with the WEIGHT statement. If there is no WEIGHT statement, $w_i = 1$ for all i.

All unknown quantities are replaced by their estimated values in the following two sections.

Diagnostics for Ordinary Generalized Linear Models

The following statistics are available for generalized linear models.

DFBETA

The DFBETA statistic for measuring the influence of the ith observation is defined as the one-step approximation to the difference in the MLE of the regression parameter vector and the MLE of the regression parameter vector without the ith observation. This one-step approximation assumes a Fisher scoring step, and is given by

$$\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{[i]} \approx DFBETA_i = (X'WX)^{-1}X_i'W_i^{\frac{1}{2}}(1-h_i)^{-\frac{1}{2}}r_{pi}$$

where h_i is the leverage defined in the section "H | LEVERAGE" on page 2000.

DFBETAS

The standardized DFBETA statistic for assessing the influence of the ith observation on the jth regression parameter is defined as the DFBETA statistic for the jth parameter divided by its estimated standard deviation, where the standard deviation is estimated from all the data.

$$DFBETAS_{ij} = DFBETA_{ij}/\hat{\sigma}(\beta_j)$$

DOBS | COOKD | COOKSD

In normal linear regression, the influence of observation i can be measured by Cook's distance (Cook and Weisberg 1982). A measure of influence of observation i for generalized linear models that is equivalent to Cook's distance for normal linear regression is given by

$$DOBS_i = p^{-1}h_i(1-h_i)^{-1}r_{pi}^2$$

where h_i is the leverage defined in the section "H | LEVERAGE" on page 2000. This measure is the one-step approximation to $2p^{-1}[L(\hat{\boldsymbol{\beta}}) - L(\hat{\boldsymbol{\beta}}_{[i]})]$, where $L(\boldsymbol{\beta})$ is the log likelihood evaluated at $\boldsymbol{\beta}$.

H | LEVERAGE

The Fisher scores, or expected, weight for observation i is $w_{ei} = \frac{w_i}{\phi V(\mu_i)(g'(\mu_i))^2}$. Let W be the diagonal matrix with w_{ei} as the ith diagonal. The leverage h_i of the ith observation is defined as the ith diagonal element of the hat matrix

$$H = W^{\frac{1}{2}}(X'WX)^{-1}W^{\frac{1}{2}}$$

Diagnostics for Models Fit by Generalized Estimating Equations (GEEs)

The diagnostic statistics in this section were developed by Preisser and Qaqish (1996). See the section "Generalized Estimating Equations" on page 1986 for further information and notation for generalized estimating equations (GEEs). The following additional notation is used in this section.

Partition the design matrix $\mathbf{X}$ and response vector $\mathbf{Y}$ by cluster; that is, let $\mathbf{X} = (X_1', \ldots, X_K')'$, and $\mathbf{Y} = (Y_1', \ldots, Y_K')'$ corresponding to the K clusters.

Let n_i be the number of responses for cluster i, and denote by $N = \sum_{i=1}^{K} n_i$ the total number of observations. Denote by A_i the $n_i \times n_i$ diagonal matrix with $V(\mu_{ij})$ as the jth diagonal element. If there is a WEIGHT statement, the diagonal element of A_i is $V(\mu_{ij})/w_{ij}$, where w_{ij} is the specified weight of the jth observation in the ith cluster. Let B the $N \times N$ diagonal matrix with $g'(\mu_{ij})$ as diagonal elements, $i = 1, \ldots, K$, $j = 1, \ldots, n_i$. Let B_i the $n_i \times n_i$ diagonal matrix corresponding to cluster i with $g'(\mu_{ij})$ as the jth diagonal element.

Let W be the $N \times N$ block diagonal weight matrix whose ith block, corresponding to the ith cluster, is the $n_i \times n_i$ matrix

$$W_{ei} = B_i^{-1}A_i^{-\frac{1}{2}}R_i^{-1}(\hat{\boldsymbol{\alpha}})A_i^{-\frac{1}{2}}B_i^{-1}$$

where R_i is the working correlation matrix for cluster i.

Let

$$Q_i = X_i(X'WX)^{-1}X_i'$$

where X_i is the $n_i \times p$ design matrix corresponding to cluster i.

Define the adjusted residual vector as

$$E = B(\mathbf{Y} - \hat{\boldsymbol{\mu}})$$

and $E_i = B_i(\mathbf{Y}_i - \hat{\boldsymbol{\mu}}_i)$, the estimated residual for the ith cluster.

Let the subscript $[i]$ denote estimates evaluated without the ith cluster, $[it]$ estimates evaluated using all the data except the tth observation of the ith cluster, and let $i[t]$ denote matrices corresponding to the ith cluster without the tth observation.

The following statistics are available for generalized estimating equation models.

CH | CLUSTERH | CLEVERAGE

The leverage of cluster i is contained in the matrix $H_i = Q_i W_{ei}$, and is summarized by the trace of H_i,

$$ch_i = tr(H_i)$$

The leverage h_i of the tth observation in the ith cluster is the tth diagonal element of H_i.

DFBETAC

The effect of deleting cluster i on the estimated parameter vector is given by the following one-step approximation for $\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{[i]}$:

$$DBETAC_i = (X'WX)^{-1}X_i'(W_{ei}^{-1} - Q_i)^{-1}E_i$$

DFBETACS

The cluster deletion statistic DFBETAC can be standardized using the variances of $\hat{\boldsymbol{\beta}}$ based on the complete data. The standardized one-step approximation for the change in $\hat{\beta}_j$ due to deletion of cluster i is

$$DBETACS_{ij} = \frac{DBETAC_{ij}}{\hat{\phi}[(X'WX)^{-1}]_{jj}^{\frac{1}{2}}}$$

DFBETAO

Partition the matrices W_{ei} and V_i as

$$W_{ei} = \begin{pmatrix} W_{eit} & W_{eit[t]} \\ W_{ei[t]t} & W_{ei[t]} \end{pmatrix}$$

$$V_i = W_{ei}^{-1} = \begin{pmatrix} V_{it} & V_{it[t]} \\ V_{i[t]t} & V_{i[t]} \end{pmatrix}$$

and let $E_{it} = B_{it}(Y_{it} - \hat{\mu}_{it})$ and $E_{i[t]} = B_{i[t]}(Y_{i[t]} - \hat{\mu}_{i[t]})$.

The effect of deleting the tth observation from the ith cluster is given by the following one-step approximation to $\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{[it]}$:

$$DBETAO_{it} = (X'WX)^{-1}\tilde{X}'_{it}\frac{\tilde{E}_{it}}{W_{eit}^{-1} - \tilde{Q}_{it}}$$

where $\tilde{X}_{it} = X_{it} - V_{it[t]}V_{i[t]}^{-1}X_{i[t]}$, $\tilde{Q}_{it} = \tilde{X}_{it}(X'WX)^{-1}\tilde{X}'_{it}$, and $\tilde{E}_{it} = E_{it} - V_{it[t]}V_{i[t]}^{-1}E_{i[t]}$. Note that W_{eit}, $\tilde{Q}_{it}$, and $\tilde{E}_{it}$ are scalars.

DFBETAOS

The observation deletion statistic DFBETAO can be standardized using the variances of $\hat{\boldsymbol{\beta}}$ based on the complete data. The standardized one-step approximation for the change in $\hat{\beta}_j$ due to deletion of observation t in cluster i is

$$DBETAOS_{itj} = \frac{DBETAO_{itj}}{\hat{\phi}[(X'WX)^{-1}]_{jj}^{\frac{1}{2}}}$$

DCLS | CLUSTERCOOKD | CLUSTERCOOKSD

A measure of the standardized influence of the subset m of observations on the overall fit is $(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{[m]})'(X'WX)(\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{[m]})/p\hat{\phi}$. For deletion of cluster i, this is approximated by

$$DCLS_i = E_i'(W_{ei}^{-1} - Q_i)^{-1})Q_i(W_{ei}^{-1} - Q_i)^{-1})E_i/p\hat{\phi}$$

DOBS | COOKD | COOKSD

The measure of overall fit in the section "DCLS | CLUSTERCOOKD | CLUSTERCOOKSD" on page 2002 for the deletion of the tth observation in the ith cluster is approximated by

$$DOBS_{it} = \frac{\tilde{E}_{it}^2\tilde{Q}_{it}}{p\hat{\phi}(W_{eit}^{-1} - \tilde{Q}_{it})^2}$$

where $\tilde{E}_{it}$, $\tilde{Q}_{it}$, and W_{eit} are defined in the section "DFBETAO" on page 2001. In the case of the independence working correlation, this is equal to the measure for ordinary generalized linear models defined in the section "DOBS | COOKD | COOKSD" on page 2000.

MCLS | CLUSTERDFIT

A studentized distance measure of the type defined in the section "DCLS | CLUSTERCOOKD | CLUSTERCOOKSD" on page 2002 of the influence of the ith cluster is given by

$$MCLS_i = E_i'(W_{ei}^{-1} - Q_i)^{-1}H_iE_i/p\hat{\phi}$$

Bayesian Analysis

Gibbs Sampling

This section provides details for Bayesian analysis by Gibbs sampling in generalized linear models. See the section "Gibbs Sampler" on page 154 for a general discussion of Gibbs sampling. See Gilks, Richardson, and Spiegelhalter (1996) for a discussion of applications of Gibbs sampling to a number of different models, including generalized linear models. In generalized linear models, the response has a probability distribution from a family of distributions of the exponential form. That is, the probability density of the response Y for continuous response variables, or the probability function for discrete responses, can be expressed as

$$f(y) = \exp\left\{\frac{y\theta - b(\theta)}{a(\phi)} + c(y,\phi)\right\}$$

for some functions a, b, and c that determine the specific distribution. The canonical parameters θ depend only on the means of the response μ_i, which are related to the regression parameters β through the link function $g(\mu_i) = x'\beta$. The additional parameter ϕ is the dispersion parameter. The GENMOD procedure estimates the regression parameters and the scale parameter $\sigma = \phi^{\frac{1}{2}}$ by maximum likelihood. However, the GENMOD procedure can also provide Bayesian estimates of the regression parameters and either the scale σ, the dispersion ϕ, or the precision $\tau = \phi^{-1}$ by Gibbs sampling. Except where noted, the following discussion applies to either σ, ϕ, or τ, although ϕ is used to illustrate the formulas. Note that the Poisson and binomial distributions do not have a dispersion parameter, and the dispersion is considered to be fixed at $\phi = 1$. The ASSESS, CONTRAST, ESTIMATE, OUTPUT, and REPEATED statements, if specified, are ignored. Also ignored are the PLOTS= option in the PROC GENMOD statement and the following options in the MODEL statement: ALPHA=, CORRB, COVB, TYPE1, TYPE3, SCALE=DEVIANCE (DSCALE), SCALE=PEARSON (PSCALE), OBSTATS, RESIDUALS, XVARS, PREDICTED, DIAGNOSTICS, and SCALE= for Poisson and binomial distributions. The multinomial and zero-inflated Poisson distributions are not available for Bayesian analysis.

Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)'$ be the parameter vector. For generalized linear models, the θ_is are the regression coefficients β_is and the dispersion parameter ϕ. Let $L(D|\boldsymbol{\theta})$ be the likelihood function, where D is the observed data. Let $\pi(\boldsymbol{\theta})$ be the prior distribution. The full conditional distribution of $[\theta_i|\theta_j, i \neq j]$ is proportional to the joint distribution; that is,

$$\pi(\theta_i|\theta_j, i \neq j, D) \propto L(D|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

For instance, the one-dimensional conditional distribution of θ_1 given $\theta_j = \theta_j^*, 2 \leq j \leq k$, is computed as

$$\pi(\theta_1|\theta_j = \theta_j^*, 2 \leq j \leq k, D) = L(D|(\boldsymbol{\theta} = (\theta_1, \theta_2^*, \ldots, \theta_k^*)')p(\boldsymbol{\theta} = (\theta_1, \theta_2^*, \ldots, \theta_k^*)')$$

Suppose you have a set of arbitrary starting values $\{\theta_1^{(0)}, \ldots, \theta_k^{(0)}\}$. Using the ARMS (adaptive rejection Metropolis sampling) algorithm of Gilks and Wild (1992) and Gilks, Best, and Tan (1995), you can do the following:

draw $\theta_1^{(1)}$ from $[\theta_1|\theta_2^{(0)}, \ldots, \theta_k^{(0)}]$

draw $\theta_2^{(1)}$ from $[\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, \ldots, \theta_k^{(0)}]$

...

draw $\theta_k^{(1)}$ from $[\theta_k|\theta_1^{(1)}, \ldots, \theta_{k-1}^{(1)}]$

This completes one iteration of the Gibbs sampler. After one iteration, you have $\{\theta_1^{(1)}, \ldots, \theta_k^{(1)}\}$. After n iterations, you have $\{\theta_1^{(n)}, \ldots, \theta_k^{(n)}\}$. PROC GENMOD implements the ARMS algorithm provided by Gilks (2003) to draw a sample from a full conditional distribution. See the section "Assessing Markov Chain Convergence" on page 156 for information about assessing the convergence of the chain of posterior samples.

You can output these posterior samples into a SAS data set through ODS. The following SAS statement outputs the posterior samples into the SAS data set Post:

```
OUTPOST=Post
```

The data set also includes the variable LogPost, representing the log of the posterior log likelihood.

Priors for Model Parameters

The model parameters are the regression coefficients and the dispersion parameter (or the precision or scale), if the model has one. The priors for the dispersion parameter and the priors for the regression coefficients are assumed to be independent, while you can have a joint multivariate normal prior for the regression coefficients.

Dispersion, Precision, or Scale Parameter

Gamma Prior The gamma distribution $G(a, b)$ has a PDF

$$f(u) = \frac{b(bu)^{a-1}e^{-bu}}{\Gamma(a)}, \qquad u > 0$$

where a is the shape parameter and b is the inverse-scale parameter. The mean is $\frac{a}{b}$ and the variance is $\frac{a}{b^2}$.

Improper Prior The joint prior density is given by

$$p(u) \propto u^{-1}, \qquad u > 0$$

Inverse Gamma Prior The inverse gamma distribution $IG(a, b)$ has a PDF

$$f(u) = \frac{b^a}{\Gamma(a)} u^{-(a+1)} e^{-b/u}, \qquad u > 0$$

where a is the shape parameter and b is the scale parameter. The mean is $\frac{b}{a-1}$ if $a > 1$, and the variance is $\frac{b^2}{(a-1)^2(a-2)}$ if $a > 2$.

Regression Coefficients

Let $\boldsymbol{\beta}$ be the regression coefficients.

Jeffreys' Prior The joint prior density is given by

$$p(\boldsymbol{\beta}) \propto |\mathbf{I}(\boldsymbol{\beta})|^{\frac{1}{2}}$$

where $\mathbf{I}(\boldsymbol{\beta})$ is the Fisher information matrix for the model. If the underlying model has a scale parameter (for example, a normal linear regression model), then the Fisher information matrix is computed with the scale parameter set to a fixed value of one.

If you specify the CONDITIONAL option, then Jeffreys' prior, conditional on the current Markov chain value of the generalized linear model precision parameter τ, is given by

$$|\tau\mathbf{I}(\boldsymbol{\beta})|^{\frac{1}{2}}$$

where τ is the model precision parameter.

See Ibrahim and Laud (1991) for a full discussion, with examples, of Jeffreys' prior for generalized linear models.

Normal Prior Assume $\boldsymbol{\beta}$ has a multivariate normal prior with mean vector $\boldsymbol{\beta}_0$ and covariance matrix $\boldsymbol{\Sigma}_0$. The joint prior density is given by

$$p(\boldsymbol{\beta}) \propto \mathrm{e}^{-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}_0)'\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_0)}$$

If you specify the CONDITIONAL option, then, conditional on the current Markov chain value of the generalized linear model precision parameter τ, the joint prior density is given by

$$p(\boldsymbol{\beta}) \propto \mathrm{e}^{-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}_0)'\tau\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_0)}$$

Uniform Prior The joint prior density is given by

$$p(\boldsymbol{\beta}) \propto 1$$

Deviance Information Criterion

Let θ_i be the model parameters at iteration i of the Gibbs sampler and let $\mathrm{LL}(\theta_i)$ be the corresponding model log likelihood. PROC GENMOD computes the following fit statistics defined by Spiegelhalter et al. (2002):

- Effective number of parameters:

$$p_D = \overline{\mathrm{LL}(\theta)} - \mathrm{LL}(\bar{\theta})$$

- Deviance information criterion (DIC):

$$\mathrm{DIC} = \overline{\mathrm{LL}(\theta)} + p_D$$

where

$$\overline{\mathrm{LL}(\theta)} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{LL}(\theta_i)$$

$$\bar{\theta} = \frac{1}{n}\sum_{i=1}^{n}\theta_i$$

PROC GENMOD uses the full log likelihoods defined in the section "Log-Likelihood Functions" on page 1967, with all terms included, for computing the DIC.

Posterior Distribution

Denote the observed data by D.

The posterior distribution is

$$\pi(\boldsymbol{\beta}|D) \propto L_P(D|\boldsymbol{\beta})p(\boldsymbol{\beta})$$

where $L_P(D|\boldsymbol{\beta})$ is the likelihood function with regression coefficients $\boldsymbol{\beta}$ as parameters.

Starting Values of the Markov Chains

When the BAYES statement is specified, PROC GENMOD generates one Markov chain containing the approximate posterior samples of the model parameters. Additional chains are produced when the Gelman-Rubin diagnostics are requested. Starting values (or initial values) can be specified in the INITIAL= data set in the BAYES statement. If INITIAL= option is not specified, PROC GENMOD picks its own initial values for the chains.

Denote $[x]$ as the integral value of x. Denote $\hat{s}(X)$ as the estimated standard error of the estimator X.

Regression Coefficients

For the first chain that the summary statistics and regression diagnostics are based on, the default initial values are estimates of the mode of the posterior distribution. If the INITIALMLE option is specified, the initial values are the maximum likelihood estimates; that is,

$$\beta_i^{(0)} = \hat{\beta}_i$$

Initial values for the rth chain ($r \geq 2$) are given by

$$\beta_i^{(0)} = \hat{\beta}_i \pm \left(2 + \left[\frac{r}{2}\right]\right)\hat{s}(\hat{\beta}_i)$$

with the plus sign for odd r and minus sign for even r.

Dispersion, Scale, or Precision Parameter λ

Let λ be the generalized linear model parameter you choose to sample, either the dispersion, scale, or precision parameter. Note that the Poisson and binomial distributions do not have this additional parameter.

For the first chain that the summary statistics and regression diagnostics are based on, the default initial values are estimates of the mode of the posterior distribution. If the INITIALMLE option is specified, the initial values are the maximum likelihood estimates; that is,

$$\lambda^{(0)} = \hat{\lambda}$$

The initial values of the rth chain ($r \geq 2$) are given by

$$\lambda^{(0)} = \hat{\lambda} e^{\pm\left([\frac{r}{2}]+2\right)\hat{s}(\hat{\lambda})}$$

with the plus sign for odd r and minus sign for even r.

OUTPOST= Output Data Set

The OUTPOST= data set contains the generated posterior samples. There are 2+n variables, where n is the number of model parameters. The variable Iteration represents the iteration number and the variable LogPost contains the log posterior likelihood values. The other n variables represent the draws of the Markov chain for the model parameters.

Missing Values

For generalized linear models, PROC GENMOD ignores any observation with a missing value for any variable involved in the model. You can score an observation in an output data set by setting only the response value to missing. For models fit with generalized estimating equations (GEEs), observations with missing values within a cluster are not used, and all available pairs are used in estimating the working correlation matrix. Clusters with fewer observations than the full cluster size are treated as having missing observations occurring at the end of the cluster. You can specify the order of missing observations with the WITHINSUBJECT= option. See the section "Missing Data" on page 1989 for more information about missing values in GEEs.

Displayed Output for Classical Analysis

The following output is produced by the GENMOD procedure. Note that some of the tables are optional and appear only in conjunction with the REPEATED statement and its options or with options in the MODEL statement. For details, see the section "ODS Table Names" on page 2019.

Model Information

The "Model Information" table displays the two-level data set name, the response distribution, the link function, the response variable name, the offset variable name, the frequency variable name, the scale weight variable name, the number of observations used, the number of events if events/trials format is used for response, the number of trials if events/trials format is used for response, the sum of frequency weights, the number of missing values in data set, and the number of invalid observations (for example, negative or 0 response values with gamma distribution or number of observations with events greater than trials with binomial distribution).

Class Level Information

If you use classification variables in the model, PROC GENMOD displays the levels of classification variables specified in the CLASS statement and in the MODEL statement. The levels are displayed in the same sorted order used to generate columns in the design matrix.

Response Profile

If you specify an ordinal model for the multinomial distribution, a table titled "Response Profile" is displayed containing the ordered values of the response variable and the number of occurrences of the values used in the model.

Iteration History for Parameter Estimates

If you specify the ITPRINT model option, PROC GENMOD displays a table containing the following for each iteration in the Newton-Raphson procedure for model fitting: the iteration number, the ridge value, the log likelihood, and values of all parameters in the model.

Criteria for Assessing Goodness of Fit

In the "Criteria for Assessing Goodness of Fit" table, PROC GENMOD displays the degrees of freedom for deviance and Pearson's chi-square, equal to the number of observations minus the number of regression parameters estimated, the deviance, the deviance divided by degrees of freedom, the scaled deviance, the scaled deviance divided by degrees of freedom, Pearson's chi-square, Pearson's chi-square divided by degrees of freedom, the scaled Pearson's chi-square, the scaled Pearson's chi-square divided by degrees of freedom, the log likelihood (excludes factorial terms) the full log likelihood, the Akaike information criterion, the corrected Akaike information criterion, and the Bayesian information criterion. The information in this table is valid only for maximum likelihood model fitting, and the table is not printed if the REPEATED statement is specified.

Last Evaluation of the Gradient

If you specify the model option ITPRINT, the GENMOD procedure displays the last evaluation of the gradient vector.

Last Evaluation of the Hessian

If you specify the model option ITPRINT, the GENMOD procedure displays the last evaluation of the Hessian matrix.

Analysis of (Initial) Parameter Estimates

The "Analysis of (Initial) Parameter Estimates" table contains the results from fitting a generalized linear model to the data. If you specify the REPEATED statement, these GLM parameter estimates are used as initial values for the GEE solution, and are displayed only if the PRINTMLE option in the REPEATED statement is specified. For each parameter in the model, PROC GENMOD displays the parameter name, as follows:

- the variable name for continuous regression variables
- the variable name and level for classification variables and interactions involving classification variables
- SCALE for the scale variable related to the dispersion parameter

In addition, PROC GENMOD displays the degrees of freedom for the parameter, the estimate value, the standard error, the Wald chi-square value, the p-value based on the chi-square distribution, and the confidence limits (Wald or profile likelihood) for parameters.

Lagrange Multiplier Statistics

If you specify that either the model intercept or the scale parameter is fixed, for those distributions that have a distribution scale parameter, the GENMOD procedure displays a table of Lagrange multiplier, or score, statistics for testing the validity of the constrained parameter that contains the test statistic, and the p-value.

Estimated Covariance Matrix

If you specify the model option COVB, the GENMOD procedure displays the estimated covariance matrix, defined as the inverse of the information matrix at the final iteration. This is based on the expected information matrix if the EXPECTED option is specified in the MODEL statement. Otherwise, it is based on the Hessian matrix used at the final iteration. This is, by default, the observed Hessian unless altered by the SCORING option in the MODEL statement.

Estimated Correlation Matrix

If you specify the CORRB model option, PROC GENMOD displays the estimated correlation matrix. This is based on the expected information matrix if the EXPECTED option is specified in the MODEL statement. Otherwise, it is based on the Hessian matrix used at the final iteration. This is, by default, the observed Hessian unless altered by the SCORING option in the MODEL statement.

Iteration History for LR Confidence Intervals

If you specify the ITPRINT and LRCI model options, PROC GENMOD displays an iteration history table for profile likelihood-based confidence intervals. For each parameter in the model, PROC GENMOD displays the parameter identification number, the iteration number, the log-likelihood value, parameter values.

Likelihood Ratio-Based Confidence Intervals for Parameters

If you specify the LRCI and the ITPRINT options in the MODEL statement, a table is displayed that summarizes profile likelihood-based confidence intervals for all parameters. For each parameter in the model, the table displays the confidence coefficient, the parameter identification number, lower and upper endpoints of confidence intervals for the parameter, and values of all other parameters at the solution.

LR Statistics for Type 1 Analysis

If you specify the TYPE1 model option, a table is displayed that contains the name of the effect, the deviance for the model including the effect and all previous effects, the degrees of freedom for the effect, the likelihood ratio statistic for testing the significance of the effect, and the p-value computed from the chi-square distribution with the effect's degrees of freedom.

If you specify either the SCALE=DEVIANCE or SCALE=PEARSON option in the MODEL statement, columns are displayed that contain the name of the effect, the deviance for the model including the effect and all previous effects, the numerator degrees of freedom, the denominator degrees of freedom, the chi-square statistic for testing the significance of the effect, the p-value computed from the chi-square distribution with numerator degrees of freedom, the F statistic for testing the significance of the effect, and the p-value based on the F distribution.

Iteration History for Type 3 Contrasts

If you specify the model options ITPRINT and TYPE3, an iteration history table is displayed for fitting the model with Type 3 contrast constraints for each effect that contains the effect name, the iteration number, the ridge value, the log likelihood, and values of all parameters.

LR Statistics for Type 3 Analysis

If you specify the TYPE3 model option, a table is displayed that contains, for each effect in the model, the name of the effect, the likelihood ratio statistic for testing the significance of the effect, the degrees of freedom for the effect, and the p-value computed from the chi-square distribution.

If you specify either the SCALE=DEVIANCE or SCALE=PEARSON option in the MODEL statement, columns are displayed that contain the name of the effect, the likelihood ratio statistic for testing the significance of the effect, the F statistic for testing the significance of the effect, the numerator degrees of freedom, the denominator degrees of freedom, the p-value based on the F distribution, and the p-value computed from the chi-square distribution with the numerator's degrees of freedom.

Wald Statistics for Type 3 Analysis

If you specify the TYPE3 and WALD model options, a table is displayed that contains the name of the effect, the degrees of freedom of the effect, the Wald statistic for testing the significance of the effect, and the p-value computed from the chi-square distribution.

Parameter Information

If you specify the ITPRINT, COVB, CORRB, WALDCI, or LRCI option in the MODEL statement, or if you specify a CONTRAST statement, a table is displayed that identifies parameters with numbers, rather than names, for use in tables and matrices where a compact identifier for parameters is helpful. For each parameter, the table contains an index number that identifies the parameter, and the parameter name, including level information for effects containing classification variables.

Observation Statistics

If you specify the OBSTATS option in the MODEL statement, PROC GENMOD displays a table containing miscellaneous statistics. Residuals and case deletion diagnostic statistics are not available for the multinomial distribution. Case deletion diagnostics are not available for zero-inflated models.

For each observation in the input data set, the following are displayed:

- the value of the response variable
- the predicted value of the mean
- the value of the linear predictor The value of an OFFSET variable is added to the linear predictor.
- the estimated standard error of the linear predictor

- the value of the negative of the weight in the Hessian matrix at the final iteration. This is the expected weight if the EXPECTED option is specified in the MODEL statement. Otherwise, it is the weight used in the final iteration. That is, it is the observed weight unless the SCORING= option has been specified.
- approximate lower and upper endpoints for a confidence interval for the predicted value of the mean
- raw residual
- Pearson residual
- deviance residual
- standardized Pearson residual
- standardized deviance residual
- likelihood residual
- leverage
- Cook's distance statistic
- DFBETA statistic, for each parameter
- standardized DFBETA statistic, for each parameter
- zero-inflation probability for zero-inflated models
- response mean for zero-inflated models

ESTIMATE Statement Results

If you specify a REPEATED statement, the ESTIMATE statement results apply to the specified GEE model. Otherwise, they apply to the specified generalized linear model.

For each ESTIMATE statement, the table contains the contrast label, the estimated value of the contrast, the standard error of the estimate, the significance level α, $(1 - \alpha) \times 100\%$ confidence intervals for contrast, the Wald chi-square statistic for the contrast, and the p-value computed from the chi-square distribution.

If you specify the EXP option, an additional row is displayed with statistics for the exponentiated value of the contrast.

CONTRAST Coefficients

If you specify the CONTRAST or ESTIMATE statement and you specify the E option, a table titled "Coefficients For Contrast *label*" is displayed, where *label* is the label specified in the CONTRAST statement. The table contains the contrast label, and the rows of the contrast matrix.

Iteration History for Contrasts

If you specify the ITPRINT option, an iteration history table is displayed for fitting the model with contrast constraints for each effect. The table contains the contrast label, the iteration number, the ridge value, the log likelihood, and values of all parameters.

CONTRAST Statement Results

If you specify a REPEATED statement, the CONTRAST statement results apply to the specified GEE model. Otherwise, they apply to the specified generalized linear model.

A table is displayed that contains the contrast label, the degrees of freedom for the contrast, and the likelihood ratio, score, or Wald statistic for testing the significance of the contrast. Score statistics are used in GEE models, likelihood ratio statistics are used in generalized linear models, and Wald statistics are used in both. Also displayed are the p-value computed from the chi-square distribution, and the type of statistic computed for this contrast: Wald, LR, or score.

If you specify either the SCALE=DEVIANCE or SCALE=PEARSON option for generalized linear models, columns are displayed that contain the contrast label, the likelihood ratio statistic for testing the significance of the contrast, the F statistic for testing the significance of the contrast, the numerator degrees of freedom, the denominator degrees of freedom, the p-value based on the F distribution, and the p-value computed from the chi-square distribution with numerator degrees of freedom.

LSMEANS Coefficients

If you specify the LSMEANS statement and you specify the E option, the "Coefficients for *effect* Least Squares Means" table is displayed, where *effect* is the effect specified in the LSMEANS statement. The table contains the effect names and the rows of least squares means coefficients.

Least Squares Means

If you specify the LSMEANS statement, the "Least Squares Means" table is displayed. The table contains for each effect the following: the effect name, and for each level of each effect the following:

- the least squares mean estimate
- standard error
- chi-square value
- p-value computed from the chi-square distribution

If you specify the DIFF option, a table titled "Differences of Least Squares Means" is displayed containing corresponding statistics for the differences between the least squares means for the levels of each effect.

GEE Model Information

If you specify the REPEATED statement, the "GEE Model Information" table displays the correlation structure of the working correlation matrix or the log odds ratio structure, the within-subject effect, the subject effect, the number of clusters, the correlation matrix dimension, and the minimum and maximum cluster size.

Log Odds Ratio Parameter Information

If you specify the REPEATED statement and specify a log odds ratio model for binary data with the LOGOR= option, then the "Log Odds Ratio Parameter Information" table is displayed showing the correspondence between data pairs and log odds ratio model parameters.

Iteration History for GEE Parameter Estimates

If you specify the REPEATED statement and the MODEL statement option ITPRINT, the "Iteration History For GEE Parameter Estimates" table is displayed. The table contains the parameter identification number, the iteration number, and values of all parameters.

Last Evaluation of the Generalized Gradient and Hessian

If you specify the REPEATED statement and select ITPRINT as a model option, PROC GENMOD displays the "Last Evaluation Of The Generalized Gradient And Hessian" table.

GEE Parameter Estimate Covariance Matrices

If you specify the REPEATED statement and the COVB option, PROC GENMOD displays the "Covariance Matrix (Model-Based)" and "Covariance Matrix (Empirical)" tables.

GEE Parameter Estimate Correlation Matrices

If you specify the REPEATED statement and the CORRB option, PROC GENMOD displays the "Correlation Matrix (Model-Based)" and "Correlation Matrix (Empirical)" tables.

GEE Working Correlation Matrix

If you specify the REPEATED statement and the CORRW option, PROC GENMOD displays the "Working Correlation Matrix" table.

GEE Fit Criteria

If you specify the REPEATED statement, PROC GENMOD displays the quasi-likelihood information criteria for model fit QIC and QIC_u in the "GEE Fit Criteria" table.

Analysis of GEE Parameter Estimates

If you specify the REPEATED statement, PROC GENMOD uses empirical standard error estimates to compute and display the "Analysis Of GEE Parameter Estimates Empirical Standard Error Estimates" table that contains the parameter names as follows:

- the variable name for continuous regression variables
- the variable name and level for classification variables and interactions involving classification variables
- "Scale" for the scale variable related to the dispersion parameter

In addition, the parameter estimate, the empirical standard error, a 95% confidence interval, and the Z score and p-value are displayed for each parameter.

If you specify the MODELSE option in the REPEATED statement, the "Analysis Of GEE Parameter Estimates Model-Based Standard Error Estimates" table based on model-based standard errors is also produced.

GEE Observation Statistics

If you specify the OBSTATS option in the REPEATED statement, PROC GENMOD displays a table containing miscellaneous statistics. For each observation in the input data set, the following are displayed:

- the value of the response variable and all other variables in the model, denoted by the variable names
- the predicted value of the mean
- the value of the linear predictor
- the standard error of the linear predictor
- confidence limits for the predicted values
- raw residual
- Pearson residual
- cluster number
- leverage

- cluster leverage
- cluster Cook's distance statistic
- studentized cluster Cook's distance statistic
- individual observation Cook's distance statistic
- cluster DFBETA statistic for each parameter
- cluster standardized DFBETA statistic for each parameter
- individual observation DFBETA statistic for each parameter
- individual observation standardized DFBETA statistic for each parameter

Displayed Output for Bayesian Analysis

If a Bayesian analysis is requested with a BAYES statement, the displayed output includes the following.

Model Information

The "Model Information" table displays the two-level data set name, the number of burn-in iterations, the number of iterations after the burn-in, the number of thinning iterations, the response distribution, the link function, the response variable name, the offset variable name, the frequency variable name, the scale weight variable name, the number of observations used, the number of events if events/trials format is used for response, the number of trials if events/trials format is used for response, the sum of frequency weights, the number of missing values in data set, and the number of invalid observations (for example, negative or 0 response values with gamma distribution or number of observations with events greater than trials with binomial distribution).

Class Level Information

The "Class Level Information" table displays the levels of classification variables if you specify a CLASS statement.

Maximum Likelihood Estimates

The "Analysis of Maximum Likelihood Parameter Estimates" table displays the maximum likelihood estimate of each parameter, the estimated standard error of the parameter estimator, and confidence limits for each parameter.

Coefficient Prior

The "Coefficient Prior" table displays the prior distribution of the regression coefficients.

Independent Prior Distributions for Model Parameters

The "Independent Prior Distributions for Model Parameters" table displays the prior distributions of additional model parameters (scale, exponential scale, Weibull scale, Weibull shape, gamma shape).

Initial Values and Seeds

The "Initial Values and Seeds" table displays the initial values and random number generator seeds for the Gibbs chains.

Fit Statistics

The "Fit Statistics" table displays the deviance information criterion (DIC) and the effective number of parameters.

Descriptive Statistics of the Posterior Samples

The "Descriptive Statistics of the Posterior Sample" table contains the size of the sample, the mean, the standard deviation, and the quartiles for each model parameter.

Interval Estimates for Posterior Sample

The "Interval Estimates for Posterior Sample" table contains the HPD intervals and the credible intervals for each model parameter.

Correlation Matrix of the Posterior Samples

The "Correlation Matrix of the Posterior Samples" table is produced if you include the CORR suboption in the SUMMARY= option in the BAYES statement. This table displays the sample correlation of the posterior samples.

Covariance Matrix of the Posterior Samples

The "Covariance Matrix of the Posterior Samples" table is produced if you include the COV suboption in the SUMMARY= option in the BAYES statement. This table displays the sample covariance of the posterior samples.

Autocorrelations of the Posterior Samples

The "Autocorrelations of the Posterior Samples" table displays the lag1, lag5, lag10, and lag50 autocorrelations for each parameter.

Gelman and Rubin Diagnostics

The "Gelman and Rubin Diagnostics" table is produced if you include the GELMAN suboption in the DIAGNOSTIC= option in the BAYES statement. This table displays the estimate of the potential scale reduction factor and its 97.5% upper confidence limit for each parameter.

Geweke Diagnostics

The "Geweke Diagnostics" table displays the Geweke statistic and its p-value for each parameter.

Raftery and Lewis Diagnostics

The "Raftery Diagnostics" tables is produced if you include the RAFTERY suboption in the DIAGNOSTIC= option in the BAYES statement. This table displays the Raftery and Lewis diagnostics for each variable.

Heidelberger and Welch Diagnostics

The "Heidelberger and Welch Diagnostics" table is displayed if you include the HEIDELBERGER suboption in the DIAGNOSTIC= option in the BAYES statement. This table shows the results of a stationary test and a halfwidth test for each parameter.

Effective Sample Size

The "Effective Sample Size" table displays, for each parameter, the effective sample size, the correlation time, and the efficiency.

Monte Carlo Standard Errors

The "Monte Carlo Standard Errors" table displays, for each parameter, the Monte Carlo standard error, the posterior sample standard deviation, and the ratio of the two.

ODS Table Names

PROC GENMOD assigns a name to each table that it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed separately in Table 37.4 for a maximum likelihood analysis and in Table 37.5 for a Bayesian analysis. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 37.4 ODS Tables Produced in PROC GENMOD for a Classical Analysis

ODS Table Name	Description	Statement	Option
AssessmentSummary	Model assessment summary	ASSESS	default
ClassLevels	Classification variable levels	CLASS	default
Contrasts	Tests of contrasts	CONTRAST	default
ContrastCoef	Contrast coefficients	CONTRAST	E
ConvergenceStatus	Convergence status	MODEL	default
CorrB	Parameter estimate correlation matrix	MODEL	CORRB
CovB	Parameter estimate covariance matrix	MODEL	COVB
Estimates	Estimates of contrasts	ESTIMATE	default
EstimateCoef	Contrast coefficients	ESTIMATE	E
GEEEmpPEst	GEE parameter estimates with empirical standard errors	REPEATED	default
GEEFitCriteria	GEE QIC fit criteria	REPEATED	default
GEELogORInfo	GEE log odds ratio model information	REPEATED	LOGOR=
GEEModInfo	GEE model information	REPEATED	default
GEEModPEst	GEE parameter estimates with model-based standard errors	REPEATED	MODELSE
GEENCorr	GEE model-based correlation matrix	REPEATED	MCORRB
GEENCov	GEE model-based covariance matrix	REPEATED	MCOVB
GEERCorr	GEE empirical correlation matrix	REPEATED	ECORRB
GEERCov	GEE empirical covariance matrix	REPEATED	ECOVB
GEEWCorr	GEE working correlation matrix	REPEATED	CORRW
IterContrasts	Iteration history for contrasts	MODEL CONTRAST	ITPRINT

Table 37.4 *continued*

ODS Table Name	Description	Statement	Option
IterLRCI	Iteration history for likelihood ratio confidence intervals	MODEL	LRCI ITPRINT
IterParms	Iteration history for parameter estimates	MODEL	ITPRINT
IterParmsGEE	Iteration history for GEE parameter estimates	MODEL REPEATED	ITPRINT
IterType3	Iteration history for Type 3 statistics	MODEL	TYPE3 ITPRINT
LRCI	Likelihood ratio confidence intervals	MODEL	LRCI ITPRINT
LSMeanCoef	Coefficients for least squares means	LSMEANS	E
LSMeanDiffs	Least squares means differences	LSMEANS	DIFF
LSMeans	Least squares means	LSMEANS	default
LagrangeStatistics	Lagrange statistics	MODEL	NOINT | NOSCALE
LastGEEGrad	Last evaluation of the generalized gradient and Hessian	MODEL REPEATED	ITPRINT
LastGradHess	Last evaluation of the gradient and Hessian	MODEL	ITPRINT
LinDep	Linearly dependent rows of contrasts	CONTRAST	default
ModelInfo	Model information	MODEL	default
Modelfit	Goodness-of-fit statistics	MODEL	default
NObs	Number of observations summary		default
NonEst	Nonestimable rows of contrasts	CONTRAST	default
ObStats	Observation-wise statistics	MODEL	OBSTATS | CL | PREDICTED | RESIDUALS | XVARS
ParameterEstimates	Parameter estimates	MODEL	default
ParmInfo	Parameter indices	MODEL	default
ResponseProfiles	Frequency counts for multinomial and binary models	MODEL	DIST=MULTINOMIAL | DIST=BINOMIAL
Type1	Type 1 tests	MODEL	TYPE1
Type3	Type 3 tests	MODEL	TYPE3
ZeroParameterEstimates	Parameter estimates for zero-inflated model	ZEROMODEL	default

Table 37.5 ODS Tables Produced in PROC GENMOD for a Bayesian Analysis

ODS Table Name	Description	Statement	Option
AutoCorr	Autocorrelations of the posterior samples	BAYES	default
ClassLevels	Classification variable levels	CLASS	default
CoeffPrior	Prior distribution of the regression coefficients	BAYES	default
ConvergenceStatus	Convergence status of maximum likelihood estimation	MODEL	default
Corr	Correlation matrix of the posterior samples	BAYES	SUMMARY=CORR
ESS	Effective sample size	BAYES	default
FitStatistics	Fit statistics	BAYES	default
Gelman	Gelman and Rubin convergence diagnostics	BAYES	DIAG=GELMAN
Geweke	Geweke convergence diagnostics	BAYES	default
Heidelberger	Heidelberger and Welch convergence diagnostics	BAYES	DIAG=HEIDELBERGER
InitialValues	Initial values of the Markov chains	BAYES	default
IterParms	Iteration history for parameter estimates	MODEL	ITPRINT
LastGradHess	Last evaluation of the gradient and Hessian for maximum likelihood estimation	MODEL	ITPRINT
ModelInfo	Model information	PROC	default
NObs	Number of observations		default
ParameterEstimates	Maximum likelihood estimates of model parameters	MODEL	default
ParmInfo	Parameter indices	MODEL	default
ParmPrior	Prior distribution for scale and shape	BAYES	default
PostIntervals	HPD and equal-tail intervals of the posterior samples	BAYES	default
PosteriorSample	Posterior samples (for ODS output data set only)	BAYES	
PostSummaries	Summary statistics of the posterior samples	BAYES	default
Raftery	Raftery and Lewis convergence diagnostics	BAYES	DIAG=RAFTERY

ODS Graphics

To request graphics with PROC GENMOD, you must first enable ODS Graphics by specifying the **ods graphics on** statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. Some graphs are produced by default; other graphs are produced by using statements and options. You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC GENMOD generates are listed in Table 37.6, along with the required statements and options.

ODS Graph Names

PROC GENMOD assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 37.6.

To request these graphs, you must specify the **ods graphics on** statement in addition to the options indicated in Table 37.6.

Table 37.6 ODS Graphics Produced by PROC GENMOD

ODS Graph Name	Description	Statement	Option
ADPanel	Autocorrelation function and density panel	BAYES	PLOTS=(AUTOCORR DENSITY)
AutocorrPanel	Autocorrelation function panel	BAYES	PLOTS= AUTOCORR
AutocorrPlot	Autocorrelation function plot	BAYES	PLOTS(UNPACK)=AUTOCORR
ClusterCooksDPlot	Cluster Cook's *D* by cluster number	PROC	PLOTS=
ClusterDFFITPlot	Cluster DFFIT by cluster number	PROC	PLOTS=
ClusterLeveragePlot	Cluster leverage by cluster number	PROC	PLOTS=
CooksDPlot	Cook's distance	PROC	PLOTS=
CumResidPanel	Panel of aggregates of residuals	ASSESS	CRPANEL
CumulativeResiduals	Model assessment based on aggregates of residuals	ASSESS	default
DevianceResidByXBeta	Deviance residuals by linear predictor	PROC	PLOTS=
DevianceResidualPlot	Deviance values	PROC	PLOTS=
DFBETAByCluster	Cluster DFBeta by cluster number	PROC	PLOTS=
DFBETAPlot	DFBeta	PROC	PLOTS=

Table 37.6 *continued*

ODS Table Name	Description	Statement	Option
DiagnosticPlot	Panel of residuals, influence, and diagnostic statistics	PROC MODEL REPEATED	PLOTS=
LeveragePlot	Leverage	PROC	PLOTS=
LikeResidByXBeta	Likelihood residuals by linear predictor	PROC	PLOTS=
LikeResidualPlot	Likelihood residuals	PROC	PLOTS=
PearsonResidByXBeta	Pearson residuals by linear predictor	PROC	PLOTS=
PearsonResidualPlot	Pearson residuals	PROC	PLOTS=
PredictedByObservation	Predicted values	PROC	PLOTS=
RawResidByXBeta	Raw residuals by linear predictor	PROC	PLOTS=
RawResidualPlot	Raw residuals	PROC	PLOTS=
StdDevianceResidByXBeta	Standardized deviance residuals by linear predictor	PROC	PLOTS=
StdDevianceResidualPlot	Standardized deviance residuals	PROC	PLOTS=
StdDFBETAByCluster	Standardized cluster DFBeta by cluster number	PROC	PLOTS=
StdDFBETAPlot	Standardized DFBeta	PROC	PLOTS=
StdPearsonResidByXBeta	Standardized Pearson residuals by linear predictor	PROC	PLOTS=
StdPearsonResidualPlot	Standardized Pearson residuals	PROC	PLOTS=
TAPanel	Trace and autocorrelation function panel	BAYES	PLOTS=(TRACE AUTOCORR)
TADPanel	Trace, autocorrelation, and density function panel	BAYES	default
TDPanel	Trace and density panel	BAYES	PLOTS=(TRACE DENSITY)
TracePanel	Trace panel	BAYES	PLOTS=TRACE
TracePlot	Trace plot	BAYES	PLOTS(UNPACK)=TRACE
ZeroInflationProbPlot	Zero-inflation probabilities	PROC	PLOTS=

Examples: GENMOD Procedure

The following examples illustrate some of the capabilities of the GENMOD procedure. These are not intended to represent definitive analyses of the data sets presented here. You should refer to the texts cited in the references for guidance on complete analysis of data by using generalized linear models.

Example 37.1: Logistic Regression

In an experiment comparing the effects of five different drugs, each drug is tested on a number of different subjects. The outcome of each experiment is the presence or absence of a positive response in a subject. The following artificial data represent the number of responses r in the n subjects for the five different drugs, labeled A through E. The response is measured for different levels of a continuous covariate x for each drug. The drug type and the continuous covariate x are explanatory variables in this experiment. The number of responses r is modeled as a binomial random variable for each combination of the explanatory variable values, with the binomial number of trials parameter equal to the number of subjects n and the binomial probability equal to the probability of a response.

The following DATA step creates the data set:

```
data drug;
   input drug$ x r n @@;
   datalines;
A  .1   1  10   A  .23  2  12   A  .67  1   9
B  .2   3  13   B  .3   4  15   B  .45  5  16   B  .78  5  13
C  .04  0  10   C  .15  0  11   C  .56  1  12   C  .7   2  12
D  .34  5  10   D  .6   5   9   D  .7   8  10
E  .2  12  20   E  .34 15  20   E  .56 13  15   E  .8  17  20
;
run;
```

A logistic regression for these data is a generalized linear model with response equal to the binomial proportion r/n. The probability distribution is binomial, and the link function is logit. For these data, drug and x are explanatory variables. The probit and the complementary log-log link functions are also appropriate for binomial data.

PROC GENMOD performs a logistic regression on the data in the following SAS statements:

```
proc genmod data=drug;
   class drug;
   model r/n = x drug / dist = bin
                        link = logit
                        lrci;
run;
```

Since these data are binomial, you use the events/trials syntax to specify the response in the MODEL statement. Profile likelihood confidence intervals for the regression parameters are computed using the LRCI option.

General model and data information is produced in Output 37.1.1.

Output 37.1.1 Model Information

The GENMOD Procedure

Model Information

Data Set	WORK.DRUG
Distribution	Binomial
Link Function	Logit
Response Variable (Events)	r
Response Variable (Trials)	n

The five levels of the CLASS variable DRUG are displayed in Output 37.1.2.

Output 37.1.2 CLASS Variable Levels

Class Level Information

Class	Levels	Values
drug	5	A B C D E

In the "Criteria For Assessing Goodness Of Fit" table displayed in Output 37.1.3, the value of the deviance divided by its degrees of freedom is less than 1. A p-value is not computed for the deviance; however, a deviance that is approximately equal to its degrees of freedom is a possible indication of a good model fit. Asymptotic distribution theory applies to binomial data as the number of binomial trials parameter n becomes large for each combination of explanatory variables. McCullagh and Nelder (1989) caution against the use of the deviance alone to assess model fit. The model fit for each observation should be assessed by examination of residuals. The OBSTATS option in the MODEL statement produces a table of residuals and other useful statistics for each observation.

Output 37.1.3 Goodness-of-Fit Criteria

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	12	5.2751	0.4396
Scaled Deviance	12	5.2751	0.4396
Pearson Chi-Square	12	4.5133	0.3761
Scaled Pearson X2	12	4.5133	0.3761
Log Likelihood		-114.7732	
Full Log Likelihood		-23.7343	
AIC (smaller is better)		59.4686	
AICC (smaller is better)		67.1050	
BIC (smaller is better)		64.8109	

In the "Analysis Of Parameter Estimates" table displayed in Output 37.1.4, chi-square values for the explanatory variables indicate that the parameter values other than the intercept term are all significant. The scale parameter is set to 1 for the binomial distribution. When you perform an overdispersion analysis, the value of the overdispersion parameter is indicated here. See the section "Overdispersion" on page 1973 for a discussion of overdispersion.

Output 37.1.4 Parameter Estimates

Analysis Of Maximum Likelihood Parameter Estimates

Parameter		DF	Estimate	Standard Error	Likelihood Ratio 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept		1	0.2792	0.4196	-0.5336	1.1190	0.44	0.5057
x		1	1.9794	0.7660	0.5038	3.5206	6.68	0.0098
drug	A	1	-2.8955	0.6092	-4.2280	-1.7909	22.59	<.0001
drug	B	1	-2.0162	0.4052	-2.8375	-1.2435	24.76	<.0001
drug	C	1	-3.7952	0.6655	-5.3111	-2.6261	32.53	<.0001
drug	D	1	-0.8548	0.4838	-1.8072	0.1028	3.12	0.0773
drug	E	0	0.0000	0.0000	0.0000	0.0000	.	.
Scale		0	1.0000	0.0000	1.0000	1.0000		

NOTE: The scale parameter was held fixed.

The preceding table contains the profile likelihood confidence intervals for the explanatory variable parameters requested with the LRCI option. Wald confidence intervals are displayed by default. Profile likelihood confidence intervals are considered to be more accurate than Wald intervals (see Aitkin et al. (1989)), especially with small sample sizes. You can specify the confidence coefficient with the ALPHA= option in the MODEL statement. The default value of 0.05, corresponding to 95% confidence limits, is used here. See the section "Confidence Intervals for Parameters" on page 1980 for a discussion of profile likelihood confidence intervals.

Example 37.2: Normal Regression, Log Link

Consider the following data, where x is an explanatory variable and y is the response variable. It appears that y varies nonlinearly with x and that the variance is approximately constant. A normal distribution with a log link function is chosen to model these data; that is, $\log(\mu_i) = \mathbf{x}_i'\boldsymbol{\beta}$ so that $\mu_i = \exp(\mathbf{x}_i'\boldsymbol{\beta})$.

```
data nor;
   input x y;
   datalines;
0 5
0 7
0 9
1 7
1 10
1 8
2 11
2 9
3 16
3 13
3 14
4 25
4 24
5 34
5 32
5 30
;
run;
```

The following SAS statements produce the analysis with the normal distribution and log link:

```
proc genmod data=nor;
   model y = x / dist = normal
                 link = log;
   output out       = Residuals
          pred      = Pred
          resraw    = Resraw
          reschi    = Reschi
          resdev    = Resdev
          stdreschi = Stdreschi
          stdresdev = Stdresdev
          reslik    = Reslik;
run;
```

The OUTPUT statement is specified to produce a data set that contains predicted values and residuals for each observation. This data set can be useful for further analysis, such as residual plotting.

The results from these statements are displayed in Output 37.2.1.

Output 37.2.1 Log-Linked Normal Regression

The GENMOD Procedure

Model Information

Data Set	WORK.NOR
Distribution	Normal
Link Function	Log
Dependent Variable	y

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	14	52.3000	3.7357
Scaled Deviance	14	16.0000	1.1429
Pearson Chi-Square	14	52.3000	3.7357
Scaled Pearson X2	14	16.0000	1.1429
Log Likelihood		-32.1783	
Full Log Likelihood		-32.1783	
AIC (smaller is better)		70.3566	
AICC (smaller is better)		72.3566	
BIC (smaller is better)		72.6743	

Analysis Of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	1.7214	0.0894	1.5461	1.8966	370.76	<.0001
x	1	0.3496	0.0206	0.3091	0.3901	286.64	<.0001
Scale	1	1.8080	0.3196	1.2786	2.5566		

NOTE: The scale parameter was estimated by maximum likelihood.

The PROC GENMOD scale parameter, in the case of the normal distribution, is the standard deviation. By default, the scale parameter is estimated by maximum likelihood. You can specify a fixed standard deviation by using the NOSCALE and SCALE= options in the MODEL statement.

```
proc print data=Residuals ;
run;
```

Output 37.2.2 Data Set of Predicted Values and Residuals

Obs	x	y	Pred	Reschi	Resraw	Resdev	Stdreschi	Stdresdev	Reslik
1	0	5	5.5921	-0.59212	-0.59212	-0.59212	-0.34036	-0.34036	-0.34036
2	0	7	5.5921	1.40788	1.40788	1.40788	0.80928	0.80928	0.80928
3	0	9	5.5921	3.40788	3.40788	3.40788	1.95892	1.95892	1.95892
4	1	7	7.9324	-0.93243	-0.93243	-0.93243	-0.54093	-0.54093	-0.54093
5	1	10	7.9324	2.06757	2.06757	2.06757	1.19947	1.19947	1.19947
6	1	8	7.9324	0.06757	0.06757	0.06757	0.03920	0.03920	0.03920
7	2	11	11.2522	-0.25217	-0.25217	-0.25217	-0.14686	-0.14686	-0.14686
8	2	9	11.2522	-2.25217	-2.25217	-2.25217	-1.31166	-1.31166	-1.31166
9	3	16	15.9612	0.03878	0.03878	0.03878	0.02249	0.02249	0.02249
10	3	13	15.9612	-2.96122	-2.96122	-2.96122	-1.71738	-1.71738	-1.71738
11	3	14	15.9612	-1.96122	-1.96122	-1.96122	-1.13743	-1.13743	-1.13743
12	4	25	22.6410	2.35897	2.35897	2.35897	1.37252	1.37252	1.37252
13	4	24	22.6410	1.35897	1.35897	1.35897	0.79069	0.79069	0.79069
14	5	34	32.1163	1.88366	1.88366	1.88366	1.22914	1.22914	1.22914
15	5	32	32.1163	-0.11634	-0.11634	-0.11634	-0.07592	-0.07592	-0.07592
16	5	30	32.1163	-2.11634	-2.11634	-2.11634	-1.38098	-1.38098	-1.38098

The data set of predicted values and residuals (Output 37.2.2) is created by the OUTPUT statement. You can use the PLOTS= option in the PROC GENMOD statement to create plots of predicted values and residuals. Note that raw, Pearson, and deviance residuals are equal in this example. This is a characteristic of the normal distribution and is not true in general for other distributions.

Example 37.3: Gamma Distribution Applied to Life Data

Life data are sometimes modeled with the gamma distribution. Although PROC GENMOD does not analyze censored data or provide other useful lifetime distributions such as the Weibull or log-normal, it can be used for modeling complete (uncensored) data with the gamma distribution, and it can provide a statistical test for the exponential distribution against other gamma distribution alternatives. See Lawless (2003) or Nelson (1982) for applications of the gamma distribution to life data.

The following data represent failure times of machine parts, some of which are manufactured by manufacturer A and some by manufacturer B.

```
data A;
   input lifetime@@ ;
   mfg = 'A';
   datalines;
620  470  260  89   388  242
103  100  39   460  284  1285
218  393  106  158  152  477
403  103  69   158  818  947
399  1274 32   12   134  660
548  381  203  871  193  531
317  85   1410 250  41   1101
```

```
32   421  32   343  376  1512
1792 47   95   76   515  72
1585 253  6    860  89   1055
537  101  385  176  11   565
164  16   1267 352  160  195
1279 356  751  500  803  560
151  24   689  1119 1733 2194
763  555  14   45   776  1
;
run;

data B;
   input lifetime@@ ;
   mfg = 'B';
   datalines;
1747 945  12   1453 14   150
20   41   35   69   195  89
1090 1868 294  96   618  44
142  892  1307 310  230  30
403  860  23   406  1054 1935
561  348  130  13   230  250
317  304  79   1793 536  12
9    256  201  733  510  660
122  27   273  1231 182  289
667  761  1096 43   44   87
405  998  1409 61   278  407
113  25   940  28   848  41
646  575  219  303  304  38
195  1061 174  377  388  10
246  323  198  234  39   308
55   729  813  1216 1618 539
6    1566 459  946  764  794
35   181  147  116  141  19
380  609  546
;
run;

data lifdat;
   set A B;
run;
```

The following SAS statements use PROC GENMOD to compute Type 3 statistics to test for differences between the two manufacturers in machine part life. Type 3 statistics are identical to Type 1 statistics in this case, since there is only one effect in the model. The log link function is selected to ensure that the mean is positive.

```
proc genmod data = lifdat;
   class mfg;
   model lifetime = mfg / dist=gamma
                          link=log
                          type3;
run;
```

The output from these statements is displayed in Output 37.3.1.

Output 37.3.1 Gamma Model of Life Data

The GENMOD Procedure

Model Information

Data Set	WORK.LIFDAT
Distribution	Gamma
Link Function	Log
Dependent Variable	lifetime

Class Level Information

Class	Levels	Values
mfg	2	A B

Criteria For Assessing Goodness Of Fit

Criterion	DF	Value	Value/DF
Deviance	199	287.0591	1.4425
Scaled Deviance	199	237.5335	1.1936
Pearson Chi-Square	199	211.6870	1.0638
Scaled Pearson X2	199	175.1652	0.8802
Log Likelihood		-1432.4177	
Full Log Likelihood		-1432.4177	
AIC (smaller is better)		2870.8353	
AICC (smaller is better)		2870.9572	
BIC (smaller is better)		2880.7453	

Analysis Of Maximum Likelihood Parameter Estimates

Parameter		DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept		1	6.1302	0.1043	5.9257	6.3347	3451.61	<.0001
mfg	A	1	0.0199	0.1559	-0.2857	0.3255	0.02	0.8985
mfg	B	0	0.0000	0.0000	0.0000	0.0000	.	.
Scale		1	0.8275	0.0714	0.6987	0.9800		

NOTE: The scale parameter was estimated by maximum likelihood.

LR Statistics For Type 3 Analysis

Source	DF	Chi-Square	Pr > ChiSq
mfg	1	0.02	0.8985

The p-value of 0.8985 for the chi-square statistic in the Type 3 table indicates that there is no significant difference in the part life between the two manufacturers.

Using the following statements, you can refit the model without using the manufacturer as an effect. The LRCI option in the MODEL statement is specified to compute profile likelihood confidence intervals for the mean life and scale parameters.

```
proc genmod data = lifdat;
   model lifetime = / dist=gamma
                      link=log
                      lrci;
run;
```

Output 37.3.2 displays the results of fitting the model with the mfg effect omitted.

Output 37.3.2 Refitting of the Gamma Model: Omitting the mfg Effect

The GENMOD Procedure

Analysis Of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	Likelihood Ratio 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	6.1391	0.0775	5.9904	6.2956	6268.10	<.0001
Scale	1	0.8274	0.0714	0.6959	0.9762		

NOTE: The scale parameter was estimated by maximum likelihood.

The intercept is the estimated log mean of the fitted gamma distribution, so that the mean life of the parts is

$$\mu = \exp(\text{INTERCEPT}) = \exp(6.1391) = 463.64$$

The SCALE parameter used in PROC GENMOD is the inverse of the gamma dispersion parameter, and it is sometimes called the gamma *index parameter*. See the section "Response Probability Distributions" on page 1964 for the definition of the gamma probability density function. A value of 1 for the index parameter corresponds to the exponential distribution . The estimated value of the scale parameter is 0.8274. The 95% profile likelihood confidence interval for the scale parameter is (0.6959, 0.9762), which does not contain 1. The hypothesis of an exponential distribution for the data is, therefore, rejected at the 0.05 level. A confidence interval for the mean life is

$$(\exp(5.99), \exp(6.30)) = (399.57, 542.18)$$

Example 37.4: Ordinal Model for Multinomial Data

This example illustrates how you can use the GENMOD procedure to fit a model to data measured on an ordinal scale. The following statements create a SAS data set called Icecream. The data set contains the results of a hypothetical taste test of three brands of ice cream. The three brands are rated for taste on a five-point scale from very good (vg) to very bad (vb). An analysis is performed

to assess the differences in the ratings of the three brands. The variable taste contains the ratings, and the variable brand contains the brands tested. The variable count contains the number of testers rating each brand in each category.

The following statements create the Icecream data set:

```
data Icecream;
   input count brand$ taste$;
   datalines;
70  ice1 vg
71  ice1 g
151 ice1 m
30  ice1 b
46  ice1 vb
20  ice2 vg
36  ice2 g
130 ice2 m
74  ice2 b
70  ice2 vb
50  ice3 vg
55  ice3 g
140 ice3 m
52  ice3 b
50  ice3 vb
;
run;
```

The following statements fit a cumulative logit model to the ordinal data with the variable taste as the response and the variable brand as a covariate. The variable count is used as a FREQ variable.

```
proc genmod data=Icecream rorder=data;
   freq count;
   class brand;
   model taste = brand / dist=multinomial
                         link=cumlogit
                         aggregate=brand
                         type1;
   estimate 'LogOR12' brand 1 -1 / exp;
   estimate 'LogOR13' brand 1  0  -1 / exp;
   estimate 'LogOR23' brand 0  1  -1 / exp;
run;
```

The AGGREGATE=BRAND option in the MODEL statement specifies the variable brand as defining multinomial populations for computing deviances and Pearson chi-squares. The RORDER=DATA option specifies that the taste variable levels be ordered by their order of appearance in the input data set—that is, from very good (vg) to very bad (vb). By default, the response is sorted in increasing ASCII order. Always check the "Response Profiles" table to verify that response levels are appropriately ordered. The TYPE1 option requests a Type 1 test for the significance of the covariate brand.

If $\gamma_j(\mathbf{x}) = \Pr(\text{taste} \leq j)$ is the cumulative probability of the jth or lower taste category, then the odds ratio comparing $\mathbf{x}_1$ to $\mathbf{x}_2$ is as follows:

$$\frac{\gamma_j(\mathbf{x}_1)/(1-\gamma_j(\mathbf{x}_1))}{\gamma_j(\mathbf{x}_2)/(1-\gamma_j(\mathbf{x}_2))} = \exp[(\mathbf{x}_1-\mathbf{x}_2)'\boldsymbol{\beta}]$$

See McCullagh and Nelder (1989, Chapter 5) for details on the cumulative logit model. The ESTIMATE statements compute log odds ratios comparing each of brands. The EXP option in the ESTIMATE statements exponentiates the log odds ratios to form odds ratio estimates. Standard errors and confidence intervals are also computed.

Output 37.4.1 displays general information about the model and data, the levels of the CLASS variable brand, and the total number of occurrences of the ordered levels of the response variable taste.

Output 37.4.1 Ordinal Model Information

The GENMOD Procedure

Model Information

Data Set	WORK.ICECREAM
Distribution	Multinomial
Link Function	Cumulative Logit
Dependent Variable	taste
Frequency Weight Variable	count

Class Level Information

Class	Levels	Values
brand	3	ice1 ice2 ice3

Response Profile

Ordered Value	taste	Total Frequency
1	vg	140
2	g	162
3	m	421
4	b	156
5	vb	166

Output 37.4.2 displays estimates of the intercept terms and covariates and associated statistics. The intercept terms correspond to the four cumulative logits defined on the taste categories in the order shown in Output 37.4.1. That is, Intercept1 is the intercept for the first cumulative logit, $\log(\frac{p_1}{1-p_1})$, Intercept2 is the intercept for the second cumulative logit, $\log(\frac{p_1+p_2}{1-(p_1+p_2)})$, and so forth.

Output 37.4.2 Parameter Estimates

Analysis Of Maximum Likelihood Parameter Estimates

Parameter		DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square
Intercept1		1	−1.8578	0.1219	−2.0967	−1.6189	232.35
Intercept2		1	−0.8646	0.1056	−1.0716	−0.6576	67.02
Intercept3		1	0.9231	0.1060	0.7154	1.1308	75.87
Intercept4		1	1.8078	0.1191	1.5743	2.0413	230.32
brand	ice1	1	0.3847	0.1370	0.1162	0.6532	7.89
brand	ice2	1	−0.6457	0.1397	−0.9196	−0.3719	21.36
brand	ice3	0	0.0000	0.0000	0.0000	0.0000	.
Scale		0	1.0000	0.0000	1.0000	1.0000	

Analysis Of Maximum Likelihood Parameter Estimates

Parameter		Pr > ChiSq
Intercept1		<.0001
Intercept2		<.0001
Intercept3		<.0001
Intercept4		<.0001
brand	ice1	0.0050
brand	ice2	<.0001
brand	ice3	.
Scale		

NOTE: The scale parameter was held fixed.

The Type 1 test displayed in Output 37.4.3 indicates that Brand is highly significant; that is, there are significant differences among the brands. The log odds ratios and odds ratios in the "ESTIMATE Statement Results" table indicate the relative differences among the brands. For example, the odds ratio of 2.8 in the "Exp(LogOR12)" row indicates that the odds of brand 1 being in lower taste categories is 2.8 times the odds of brand 2 being in lower taste categories. Since, in this ordering, the lower categories represent the more favorable taste results, this indicates that brand 1 scored significantly better than brand 2. This is also apparent from the data in this example.

Output 37.4.3 Type 1 Tests and Odds Ratios

LR Statistics For Type 1 Analysis

Source	Deviance	DF	Chi-Square	Pr > ChiSq
Intercepts	65.9576			
brand	9.8654	2	56.09	<.0001

Output 37.4.3 *continued*

Contrast Estimate Results

Label	Mean Estimate	Mean Confidence Limits		L'Beta Estimate	Standard Error	Alpha
LogOR12	0.7370	0.6805	0.7867	1.0305	0.1401	0.05
Exp(LogOR12)				2.8024	0.3926	0.05
LogOR13	0.5950	0.5290	0.6577	0.3847	0.1370	0.05
Exp(LogOR13)				1.4692	0.2013	0.05
LogOR23	0.3439	0.2850	0.4081	-0.6457	0.1397	0.05
Exp(LogOR23)				0.5243	0.0733	0.05

Contrast Estimate Results

Label	L'Beta Confidence Limits		Chi-Square	Pr > ChiSq
LogOR12	0.7559	1.3050	54.11	<.0001
Exp(LogOR12)	2.1295	3.6878		
LogOR13	0.1162	0.6532	7.89	0.0050
Exp(LogOR13)	1.1233	1.9217		
LogOR23	-0.9196	-0.3719	21.36	<.0001
Exp(LogOR23)	0.3987	0.6894		

Example 37.5: GEE for Binary Data with Logit Link Function

Output 37.5.1 displays a partial listing of a SAS data set of clinical trial data comparing two treatments for a respiratory disorder. See "Gee Model for Binary Data" in the SAS/STAT Sample Program Library for the complete data set. These data are from Stokes, Davis, and Koch (2000).

Patients in each of two centers are randomly assigned to groups receiving the active treatment or a placebo. During treatment, respiratory status, represented by the variable outcome (coded here as 0=poor, 1=good), is determined for each of four visits. The variables center, treatment, sex, and baseline (baseline respiratory status) are classification variables with two levels. The variable age (age at time of entry into the study) is a continuous variable.

Explanatory variables in the model are Intercept (x_{ij1}), treatment (x_{ij2}), center (x_{ij3}), sex (x_{ij4}), age (x_{ij5}), and baseline (x_{ij6}), so that $x' = [x_{ij1}, x_{ij2}, \ldots, x_{ij6}]$ is the vector of explanatory variables. Indicator variables for the classification explanatory variables can be automatically generated by listing them in the CLASS statement in PROC GENMOD. To be consistent with the analysis in Stokes, Davis, and Koch (2000), the four classification explanatory variables are coded as follows via options in the CLASS statement:

$$x_{ij2} = \begin{cases} 0 \text{ placebo} \\ 1 \text{ active} \end{cases} \qquad x_{ij3} = \begin{cases} 0 \text{ center 1} \\ 1 \text{ center 2} \end{cases}$$

$$x_{ij4} = \begin{cases} 0 \text{ male} \\ 1 \text{ female} \end{cases} \qquad x_{ij6} = \begin{cases} 0\ 0 \\ 1\ 1 \end{cases}$$

Suppose y_{ij} represents the respiratory status of patient i at the jth visit, $j = 1, \ldots, 4$, and $\mu_{ij} = \mathrm{E}(y_{ij})$ represents the mean of the respiratory status. Since the response data are binary, you can use the variance function for the binomial distribution $v(\mu_{ij}) = \mu_{ij}(1-\mu_{ij})$ and the logit link function $g(\mu_{ij}) = \log(\mu_{ij}/(1-\mu_{ij}))$. The model for the mean is $g(\mu_{ij}) = \mathbf{x}_{ij}'\boldsymbol{\beta}$, where $\boldsymbol{\beta}$ is a vector of regression parameters to be estimated.

Output 37.5.1 Respiratory Disorder Data

Obs	center	id	treatment	sex	age	baseline	visit1	visit2	visit3	visit4	visit	outcome
1	1	1	P	M	46	0	0	0	0	0	1	0
2	1	1	P	M	46	0	0	0	0	0	2	0
3	1	1	P	M	46	0	0	0	0	0	3	0
4	1	1	P	M	46	0	0	0	0	0	4	0
5	1	2	P	M	28	0	0	0	0	0	1	0
6	1	2	P	M	28	0	0	0	0	0	2	0
7	1	2	P	M	28	0	0	0	0	0	3	0
8	1	2	P	M	28	0	0	0	0	0	4	0
9	1	3	A	M	23	1	1	1	1	1	1	1
10	1	3	A	M	23	1	1	1	1	1	2	1
11	1	3	A	M	23	1	1	1	1	1	3	1
12	1	3	A	M	23	1	1	1	1	1	4	1
13	1	4	P	M	44	1	1	1	1	0	1	1
14	1	4	P	M	44	1	1	1	1	0	2	1
15	1	4	P	M	44	1	1	1	1	0	3	1
16	1	4	P	M	44	1	1	1	1	0	4	0
17	1	5	P	F	13	1	1	1	1	1	1	1
18	1	5	P	F	13	1	1	1	1	1	2	1
19	1	5	P	F	13	1	1	1	1	1	3	1
20	1	5	P	F	13	1	1	1	1	1	4	1

The GEE solution is requested with the REPEATED statement in the GENMOD procedure. The option SUBJECT=ID(CENTER) specifies that the observations in a single cluster be uniquely identified by center and id within center. The option TYPE=UNSTR specifies the unstructured working correlation structure. The MODEL statement specifies the regression model for the mean with the binomial distribution variance function. The following SAS statements perform the GEE model fit:

```
proc genmod data=resp descend;
   class id treatment(ref="P") center(ref="1") sex(ref="M")
      baseline(ref="0") / param=ref;
   model outcome=treatment center sex age baseline / dist=bin;
   repeated subject=id(center) / corr=unstr corrw;
run;
```

These statements first fit the generalized linear (GLM) model specified in the MODEL statement. The parameter estimates from the generalized linear model fit are not shown in the output, but they are used as initial values for the GEE solution. The DESCEND option in the PROC GENMOD statement specifies that the probability that outcome $= 1$ be modeled. If the DESCEND option had not been specified, the probability that outcome $= 0$ would be modeled by default.

Information about the GEE model is displayed in Output 37.5.2. The results of GEE model fitting are displayed in Output 37.5.3. Model goodness-of-fit criteria are displayed in Output 37.5.4. If you specify no other options, the standard errors, confidence intervals, Z scores, and p-values are based on empirical standard error estimates. You can specify the MODELSE option in the REPEATED statement to create a table based on model-based standard error estimates.

Output 37.5.2 Model Fitting Information

The GENMOD Procedure

GEE Model Information

Correlation Structure	Unstructured
Subject Effect	id(center) (111 levels)
Number of Clusters	111
Correlation Matrix Dimension	4
Maximum Cluster Size	4
Minimum Cluster Size	4

Output 37.5.3 Results of Model Fitting

Working Correlation Matrix

	Col1	Col2	Col3	Col4
Row1	1.0000	0.3351	0.2140	0.2953
Row2	0.3351	1.0000	0.4429	0.3581
Row3	0.2140	0.4429	1.0000	0.3964
Row4	0.2953	0.3581	0.3964	1.0000

Output 37.5.3 *continued*

Analysis Of GEE Parameter Estimates
Empirical Standard Error Estimates

Parameter		Estimate	Standard Error	95% Confidence Limits		Z	Pr > |Z|
Intercept		−0.8882	0.4568	−1.7835	0.0071	−1.94	0.0519
treatment	A	1.2442	0.3455	0.5669	1.9214	3.60	0.0003
center	2	0.6558	0.3512	−0.0326	1.3442	1.87	0.0619
sex	F	0.1128	0.4408	−0.7512	0.9768	0.26	0.7981
age		−0.0175	0.0129	−0.0427	0.0077	−1.36	0.1728
baseline	1	1.8981	0.3441	1.2237	2.5725	5.52	<.0001

Output 37.5.4 Model Fit Criteria

GEE Fit Criteria

QIC	512.3416
QICu	499.6081

The nonsignificance of age and sex make them candidates for omission from the model.

Example 37.6: Log Odds Ratios and the ALR Algorithm

Since the respiratory data in Example 37.5 are binary, you can use the ALR algorithm to model the log odds ratios instead of using working correlations to model associations. In this example, a "fully parameterized cluster" model for the log odds ratio is fit. That is, there is a log odds ratio parameter for each unique pair of responses within clusters, and all clusters are parameterized identically. The following statements fit the same regression model for the mean as in Example 37.5 but use a regression model for the log odds ratios instead of a working correlation. The LOGOR=FULLCLUST option specifies a fully parameterized log odds ratio model.

```
proc genmod data=resp descend;
   class id treatment(ref="P") center(ref="1") sex(ref="M")
      baseline(ref="0") / param=ref;
   model outcome=treatment center sex age baseline / dist=bin;
   repeated  subject=id(center) / logor=fullclust;
run;
```

The results of fitting the model are displayed in Output 37.6.1 along with a table that shows the correspondence between the log odds ratio parameters and the within-cluster pairs. Model goodness-of-fit criteria are shown in Output 37.6.2. The QIC for the ALR model shown in Output 37.6.2 is 511.86, whereas the QIC for the unstructured working correlation model shown in Output 37.5.4 is 512.34, indicating that the ALR model is a slightly better fit.

Output 37.6.1 Results of Model Fitting

The GENMOD Procedure

Log Odds Ratio
Parameter Information

Parameter	Group
Alpha1	(1, 2)
Alpha2	(1, 3)
Alpha3	(1, 4)
Alpha4	(2, 3)
Alpha5	(2, 4)
Alpha6	(3, 4)

Analysis Of GEE Parameter Estimates
Empirical Standard Error Estimates

Parameter		Estimate	Standard Error	95% Confidence Limits		Z	Pr > |Z|
Intercept		-0.9266	0.4513	-1.8111	-0.0421	-2.05	0.0400
treatment	A	1.2611	0.3406	0.5934	1.9287	3.70	0.0002
center	2	0.6287	0.3486	-0.0545	1.3119	1.80	0.0713
sex	F	0.1024	0.4362	-0.7526	0.9575	0.23	0.8144
age		-0.0162	0.0125	-0.0407	0.0084	-1.29	0.1977
baseline	1	1.8980	0.3404	1.2308	2.5652	5.58	<.0001
Alpha1		1.6109	0.4892	0.6522	2.5696	3.29	0.0010
Alpha2		1.0771	0.4834	0.1297	2.0246	2.23	0.0259
Alpha3		1.5875	0.4735	0.6594	2.5155	3.35	0.0008
Alpha4		2.1224	0.5022	1.1381	3.1068	4.23	<.0001
Alpha5		1.8818	0.4686	0.9634	2.8001	4.02	<.0001
Alpha6		2.1046	0.4949	1.1347	3.0745	4.25	<.0001

Output 37.6.2 Model Fit Criteria

GEE Fit Criteria

QIC	511.8589
QICu	499.6516

You can fit the same model by fully specifying the z-matrix. The following statements create a data set containing the full z-matrix:

```
data zin;
   keep id center z1-z6 y1 y2;
   array zin(6) z1-z6;
   set resp ;
   by center id;
   if first.id
      then do;
         t = 0;
         do m = 1 to 4;
            do n = m+1 to 4;
               do j = 1 to 6;
                  zin(j) = 0;
               end;
            y1 = m;
            y2 = n;
            t + 1;
            zin(t) = 1;
            output;
            end;
         end;
      end;
run;

proc print data=zin (obs=12);
```

Output 37.6.3 displays the full z-matrix for the first two clusters. The z-matrix is identical for all clusters in this example.

Output 37.6.3 Full z-Matrix Data Set

Obs	z1	z2	z3	z4	z5	z6	center	id	y1	y2
1	1	0	0	0	0	0	1	1	1	2
2	0	1	0	0	0	0	1	1	1	3
3	0	0	1	0	0	0	1	1	1	4
4	0	0	0	1	0	0	1	1	2	3
5	0	0	0	0	1	0	1	1	2	4
6	0	0	0	0	0	1	1	1	3	4
7	1	0	0	0	0	0	1	2	1	2
8	0	1	0	0	0	0	1	2	1	3
9	0	0	1	0	0	0	1	2	1	4
10	0	0	0	1	0	0	1	2	2	3
11	0	0	0	0	1	0	1	2	2	4
12	0	0	0	0	0	1	1	2	3	4

The following statements fit the model for fully parameterized clusters by fully specifying the z-matrix. The results are identical to those shown previously.

```
proc genmod data=resp descend;
   class id treatment(ref="P") center(ref="1") sex(ref="M")
      baseline(ref="0") / param=ref;
   model outcome=treatment center sex age baseline / dist=bin;
   repeated  subject=id(center) / logor=zfull
                                  zdata=zin
                                  zrow =(z1-z6)
                                  ypair=(y1 y2) ;
run;
```

Example 37.7: Log-Linear Model for Count Data

In this example the data, from Thall and Vail (1990), concern the treatment of people suffering from epileptic seizure episodes. These data are also analyzed in Diggle, Liang, and Zeger (1994). The data consist of the number of epileptic seizures in an eight-week baseline period, before any treatment, and in each of four two-week treatment periods, in which patients received either a placebo or the drug Progabide in addition to other therapy. A portion of the data is displayed in Table 37.7. See "Gee Model for Count Data, Exchangeable Correlation" in the SAS/STAT Sample Program Library for the complete data set.

Table 37.7 Epileptic Seizure Data

Patient ID	Treatment	Baseline	Visit1	Visit2	Visit3	Visit4
104	Placebo	11	5	3	3	3
106	Placebo	11	3	5	3	3
107	Placebo	6	2	4	0	5
.						
.						
.						
101	Progabide	76	11	14	9	8
102	Progabide	38	8	7	9	4
103	Progabide	19	0	4	3	0
.						
.						
.						

Model the data as a log-linear model with $V(\mu) = \mu$ (the Poisson variance function) and

$$\log(E(Y_{ij})) = \beta_0 + x_{i1}\beta_1 + x_{i2}\beta_2 + x_{i1}x_{i2}\beta_3 + \log(t_{ij})$$

where

Y_{ij} = number of epileptic seizures in interval j

t_{ij} = length of interval j

$$x_{i1} = \begin{cases} 1: & \text{weeks 8–16 (treatment)} \\ 0: & \text{weeks 0–8 (baseline)} \end{cases}$$

$$x_{i2} = \begin{cases} 1: & \text{progabide group} \\ 0: & \text{placebo group} \end{cases}$$

The correlations between the counts are modeled as $r_{ij} = \alpha, i \neq j$ (exchangeable correlations). For comparison, the correlations are also modeled as independent (identity correlation matrix). In this model, the regression parameters have the interpretation in terms of the log seizure rate displayed in Table 37.8.

Table 37.8 Interpretation of Regression Parameters

Treatment	Visit	$\log(E(Y_{ij})/t_{ij})$
Placebo	Baseline	β_0
	1–4	$\beta_0 + \beta_1$
Progabide	Baseline	$\beta_0 + \beta_2$
	1–4	$\beta_0 + \beta_1 + \beta_2 + \beta_3$

The difference between the log seizure rates in the pretreatment (baseline) period and the treatment periods is β_1 for the placebo group and $\beta_1 + \beta_3$ for the Progabide group. A value of $\beta_3 < 0$ indicates a reduction in the seizure rate.

Output 37.7.1 lists the first 14 observations of the data, which are arranged as one visit per observation:

Output 37.7.1 Partial Listing of the Seizure Data

Obs	id	y	visit	trt	bline	age
1	104	5	1	0	11	31
2	104	3	2	0	11	31
3	104	3	3	0	11	31
4	104	3	4	0	11	31
5	106	3	1	0	11	30
6	106	5	2	0	11	30
7	106	3	3	0	11	30
8	106	3	4	0	11	30
9	107	2	1	0	6	25
10	107	4	2	0	6	25
11	107	0	3	0	6	25
12	107	5	4	0	6	25
13	114	4	1	0	8	36
14	114	4	2	0	8	36

Some further data manipulations create an observation for the baseline measures, a log time interval variable for use as an offset, and an indicator variable for whether the observation is for a baseline measurement or a visit measurement. Patient 207 is deleted as an outlier, as in the Diggle, Liang, and Zeger (1994) analysis. The following statements prepare the data for analysis with PROC GENMOD:

```
data new;
   set thall;
   output;
   if visit=1 then do;
      y=bline;
      visit=0;
      output;
   end;
run;

data new;
   set new;
   if id ne 207;
   if visit=0 then do;
      x1=0;
      ltime=log(8);
   end;
   else do;
      x1=1;
      ltime=log(2);
   end;
run;
```

For comparison with the GEE results, an ordinary Poisson regression is first fit. The results are shown in Output 37.7.2.

Output 37.7.2 Maximum Likelihood Estimates

The GENMOD Procedure

Analysis Of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	1.3476	0.0341	1.2809	1.4144	1565.44	<.0001
x1	1	0.1108	0.0469	0.0189	0.2027	5.58	0.0181
trt	1	-0.1080	0.0486	-0.2034	-0.0127	4.93	0.0264
x1*trt	1	-0.3016	0.0697	-0.4383	-0.1649	18.70	<.0001
Scale	0	1.0000	0.0000	1.0000	1.0000		

NOTE: The scale parameter was held fixed.

The GEE solution is requested with the REPEATED statement in the GENMOD procedure. The SUBJECT=ID option indicates that the variable id describes the observations for a single cluster, and the CORRW option displays the working correlation matrix. The TYPE= option specifies the correlation structure; the value EXCH indicates the exchangeable structure.

The following statements perform the analysis:

```
proc genmod data=new;
   class id;
   model y=x1 | trt / d=poisson offset=ltime;
   repeated subject=id / corrw covb type=exch;
run;
```

These statements first fit a generalized linear model (GLM) to these data by maximum likelihood. The estimates are not shown in the output, but are used as initial values for the GEE solution.

Information about the GEE model is displayed in Output 37.7.3. The results of fitting the model are displayed in Output 37.7.4. Compare these with the model of independence displayed in Output 37.7.2. The parameter estimates are nearly identical, but the standard errors for the independence case are underestimated. The coefficient of the interaction term, β_3, is highly significant under the independence model and marginally significant with the exchangeable correlations model.

Output 37.7.3 GEE Model Information

The GENMOD Procedure

GEE Model Information

Correlation Structure	Exchangeable
Subject Effect	id (58 levels)
Number of Clusters	58
Correlation Matrix Dimension	5
Maximum Cluster Size	5
Minimum Cluster Size	5

Output 37.7.4 GEE Parameter Estimates

Analysis Of GEE Parameter Estimates
Empirical Standard Error Estimates

Parameter	Estimate	Standard Error	95% Confidence Limits		Z	Pr > |Z|
Intercept	1.3476	0.1574	1.0392	1.6560	8.56	<.0001
x1	0.1108	0.1161	-0.1168	0.3383	0.95	0.3399
trt	-0.1080	0.1937	-0.4876	0.2716	-0.56	0.5770
x1*trt	-0.3016	0.1712	-0.6371	0.0339	-1.76	0.0781

Table 37.9 displays the regression coefficients, standard errors, and normalized coefficients that result from fitting the model with independent and exchangeable working correlation matrices.

Table 37.9 Results of Model Fitting

Variable	Correlation Structure	Coef.	Std. Error	Coef./S.E.
Intercept	Exchangeable	1.35	0.16	8.56
	Independent	1.35	0.03	39.52
Visit (x_1)	Exchangeable	0.11	0.12	0.95
	Independent	0.11	0.05	2.36
Treat (x_2)	Exchangeable	−0.11	0.19	−0.56
	Independent	−0.11	0.05	−2.22
$x_1 * x_2$	Exchangeable	−0.30	0.17	−1.76
	Independent	−0.30	0.07	−4.32

The fitted exchangeable correlation matrix is specified with the CORRW option and is displayed in Output 37.7.5.

Output 37.7.5 Working Correlation Matrix

Working Correlation Matrix

	Col1	Col2	Col3	Col4	Col5
Row1	1.0000	0.5941	0.5941	0.5941	0.5941
Row2	0.5941	1.0000	0.5941	0.5941	0.5941
Row3	0.5941	0.5941	1.0000	0.5941	0.5941
Row4	0.5941	0.5941	0.5941	1.0000	0.5941
Row5	0.5941	0.5941	0.5941	0.5941	1.0000

If you specify the COVB option, you produce both the model-based (naive) and the empirical (robust) covariance matrices. Output 37.7.6 contains these estimates.

Output 37.7.6 Covariance Matrices

Covariance Matrix (Model-Based)

	Prm1	Prm2	Prm3	Prm4
Prm1	0.01223	0.001520	-0.01223	-0.001520
Prm2	0.001520	0.01519	-0.001520	-0.01519
Prm3	-0.01223	-0.001520	0.02495	0.005427
Prm4	-0.001520	-0.01519	0.005427	0.03748

Covariance Matrix (Empirical)

	Prm1	Prm2	Prm3	Prm4
Prm1	0.02476	-0.001152	-0.02476	0.001152
Prm2	-0.001152	0.01348	0.001152	-0.01348
Prm3	-0.02476	0.001152	0.03751	-0.002999
Prm4	0.001152	-0.01348	-0.002999	0.02931

The two covariance estimates are similar, indicating an adequate correlation model.

Example 37.8: Model Assessment of Multiple Regression Using Aggregates of Residuals

This example illustrates the use of cumulative residuals to assess the adequacy of a normal linear regression model. Neter et al. (1996, Section 8.2) describe a study of 54 patients undergoing a certain kind of liver operation in a surgical unit. The data consist of the survival time and certain covariates. After a model selection procedure, they arrived at the following model:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3 + \epsilon$$

where Y is the logarithm (base 10) of the survival time; X_1, X_2, X_3 are *blood-clotting score*, *prognostic index*, and *enzyme function*, respectively; and ϵ is a normal error term. A listing of the SAS data set containing the data is shown in Output 37.8.1. The variables Y, X1, X2, and X3 correspond to Y, X_1, X_2, and X_3, and LogX1 is $\log(X_1)$. The PROC GENMOD fit of the model is shown in Output 37.8.2. The analysis first focuses on the adequacy of the functional form of X_1, *blood-clotting score*.

Output 37.8.1 Surgical Unit Example Data

Obs	Y	X1	X2	X3	LogX1
1	2.3010	6.7	62	81	0.82607
2	2.0043	5.1	59	66	0.70757
3	2.3096	7.4	57	83	0.86923
4	2.0043	6.5	73	41	0.81291
5	2.7067	7.8	65	115	0.89209
6	1.9031	5.8	38	72	0.76343
7	1.9031	5.7	46	63	0.75587
8	2.1038	3.7	68	81	0.56820
9	2.3054	6.0	67	93	0.77815
10	2.3075	3.7	76	94	0.56820
11	2.5172	6.3	84	83	0.79934
12	1.8129	6.7	51	43	0.82607
13	2.9191	5.8	96	114	0.76343
14	2.5185	5.8	83	88	0.76343
15	2.2253	7.7	62	67	0.88649
16	2.3365	7.4	74	68	0.86923
17	1.9395	6.0	85	28	0.77815
18	1.5315	3.7	51	41	0.56820
19	2.3324	7.3	68	74	0.86332
20	2.2355	5.6	57	87	0.74819
21	2.0374	5.2	52	76	0.71600
22	2.1335	3.4	83	53	0.53148
23	1.8451	6.7	26	68	0.82607
24	2.3424	5.8	67	86	0.76343
25	2.4409	6.3	59	100	0.79934
26	2.1584	5.8	61	73	0.76343
27	2.2577	5.2	52	86	0.71600
28	2.7589	11.2	76	90	1.04922
29	1.8573	5.2	54	56	0.71600
30	2.2504	5.8	76	59	0.76343
31	1.8513	3.2	64	65	0.50515
32	1.7634	8.7	45	23	0.93952
33	2.0645	5.0	59	73	0.69897
34	2.4698	5.8	72	93	0.76343
35	2.0607	5.4	58	70	0.73239
36	2.2648	5.3	51	99	0.72428
37	2.0719	2.6	74	86	0.41497
38	2.0792	4.3	8	119	0.63347
39	2.1790	4.8	61	76	0.68124
40	2.1703	5.4	52	88	0.73239
41	1.9777	5.2	49	72	0.71600
42	1.8751	3.6	28	99	0.55630
43	2.6840	8.8	86	88	0.94448
44	2.1847	6.5	56	77	0.81291
45	2.2810	3.4	77	93	0.53148
46	2.0899	6.5	40	84	0.81291
47	2.4928	4.5	73	106	0.65321
48	2.5999	4.8	86	101	0.68124
49	2.1987	5.1	67	77	0.70757
50	2.4914	3.9	82	103	0.59106
51	2.0934	6.6	77	46	0.81954
52	2.0969	6.4	85	40	0.80618
53	2.2967	6.4	59	85	0.80618
54	2.4955	8.8	78	72	0.94448

In order to assess the adequacy of the fitted multiple regression model, the ASSESS statement in the following SAS statements is used to create the plots of cumulative residuals against X1 shown in Output 37.8.3 and Output 37.8.4 and the summary table in Output 37.8.5:

```
ods graphics on;
proc genmod data=Surg;
   model Y = X1 X2 X3 / scale=Pearson;
   assess var=(X1) / resample=10000
                     seed=603708000
                     crpanel ;
run;
ods graphics off;
```

Output 37.8.2 Regression Model for Linear X1

The GENMOD Procedure

Analysis Of Maximum Likelihood Parameter Estimates

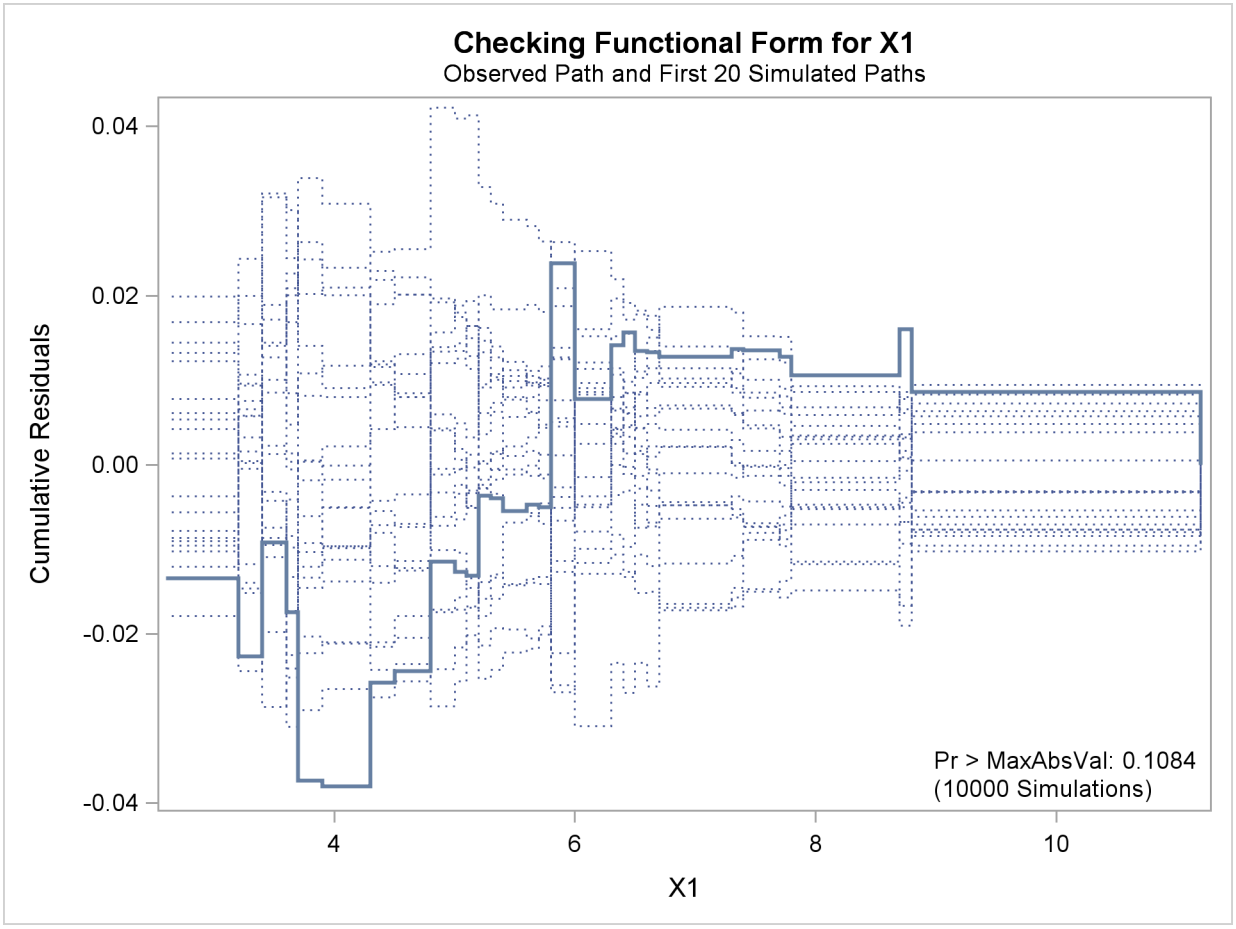
Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	0.4836	0.0426	0.4001	0.5672	128.71	<.0001
X1	1	0.0692	0.0041	0.0612	0.0772	288.17	<.0001
X2	1	0.0093	0.0004	0.0085	0.0100	590.45	<.0001
X3	1	0.0095	0.0003	0.0089	0.0101	966.07	<.0001
Scale	0	0.0469	0.0000	0.0469	0.0469		

NOTE: The scale parameter was estimated by the square root of Pearson's Chi-Square/DOF.

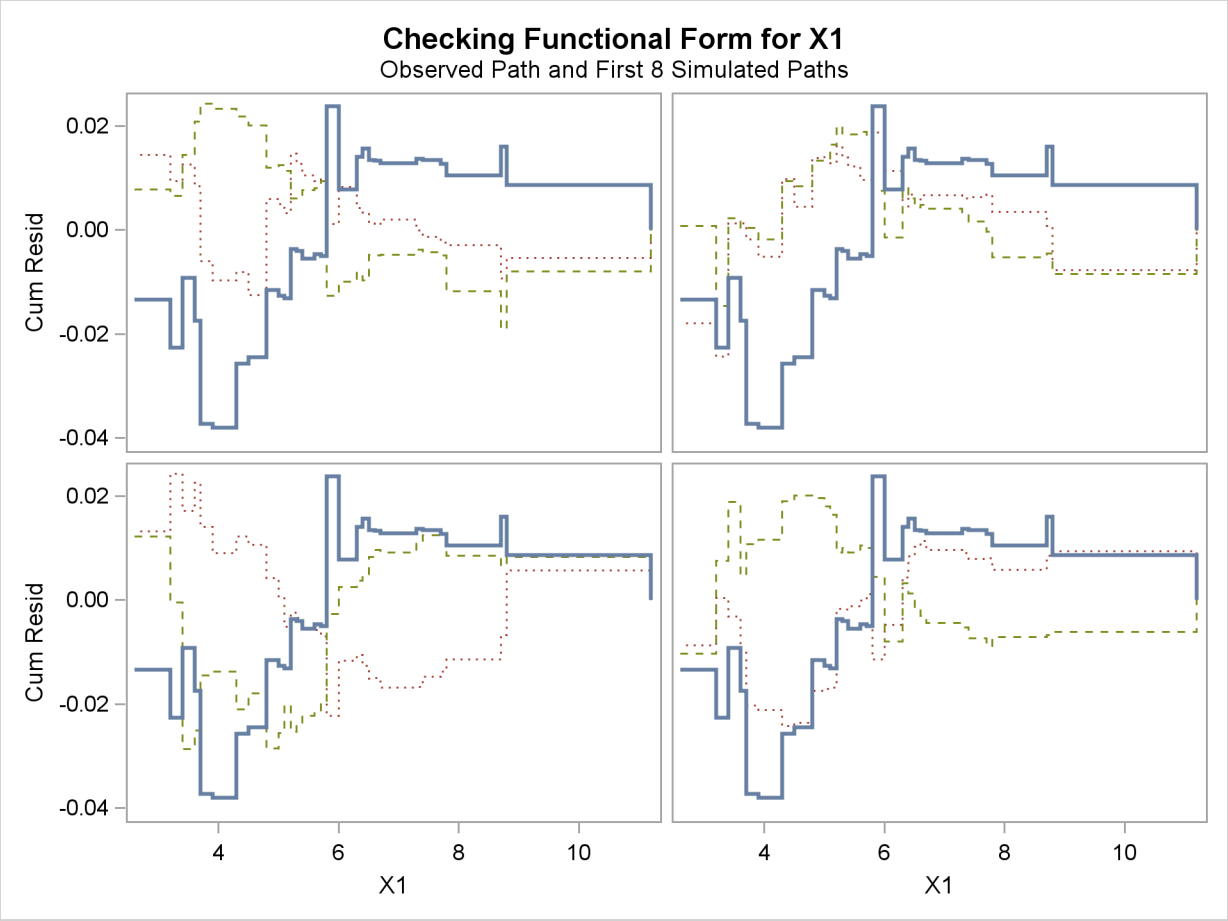
See Lin, Wei, and Ying (2002) for details about model assessment that uses cumulative residual plots. The RESAMPLE= keyword specifies that a p-value be computed based on a sample of 10,000 simulated residual paths. A random number seed is specified by the SEED= keyword for reproducibility. If you do not specify the seed, one is derived from the time of day. The keyword CRPANEL specifies that the panel of four cumulative residual plots shown in Output 37.8.4 be created, each with two simulated paths. The single residual plot with 20 simulated paths in Output 37.8.3 is created by default.

These graphical displays are requested by specifying the ODS GRAPHICS statement and the ASSESS statement. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the GENMOD procedure, see the section "ODS Graphics" on page 2022.

Output 37.8.3 Cumulative Residual Plot for Linear X1 Fit



Output 37.8.4 Cumulative Residual Panel Plot for Linear X1 Fit



Output 37.8.5 Summary of Model Assessment

Assessment Summary

Assessment Variable	Maximum Absolute Value	Replications	Seed	Pr > MaxAbsVal
X1	0.0380	10000	603708000	0.1084

The p-value of 0.1084 reported on Output 37.8.3 and Output 37.8.5 suggests that a more adequate model might be possible. The observed cumulative residuals in Output 37.8.3 and Output 37.8.4, represented by the heavy lines, seem atypical of the simulated curves, represented by the light lines, reinforcing the conclusion that a more appropriate functional form for X1 is possible.

The cumulative residual plots in Output 37.8.6 provide guidance in determining a more appropriate functional form. The four curves were created from simple forms of model misspecification by using simulated data. The mean models of the data and the fitted model are shown in Table 37.10.

Output 37.8.6 Typical Cumulative Residual Patterns

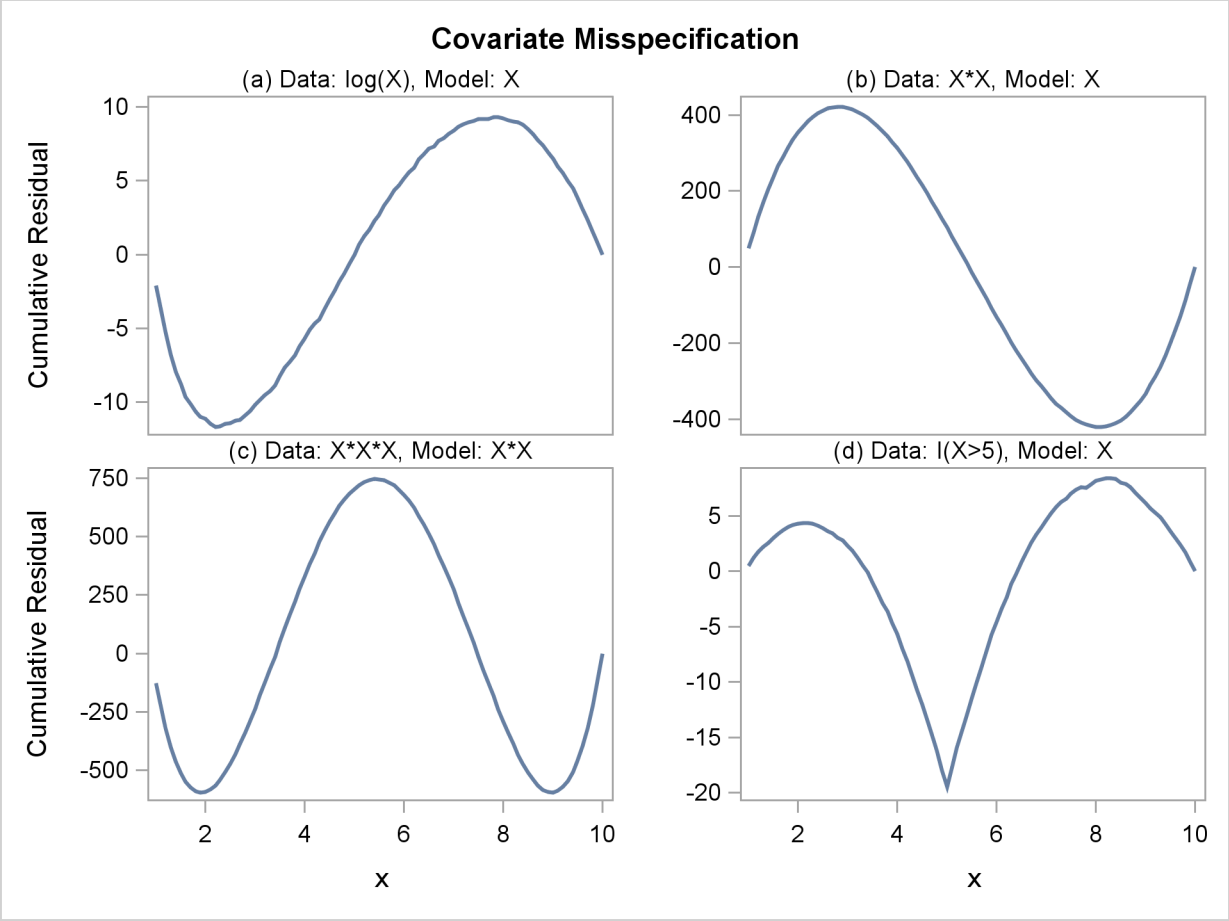


Table 37.10 Model Misspecifications

Plot	Data E(Y)	Fitted Model E(Y)
(a)	$\log(X)$	X
(b)	$X + X^2$	X
(c)	$X + X^2 + X^3$	$X + X^2$
(d)	$I(X > 5)$	X

The observed cumulative residual pattern in Output 37.8.3 and Output 37.8.4 most resembles the behavior of the curve in plot (a) of Output 37.8.6, indicating that $\log(X_1)$ might be a more appropriate term in the model than X_1.

The following SAS statements fit a model with LogX1 in place of X1 and request a model assessment:

```
proc genmod data=Surg;
ods graphics on;
   model Y = LogX1 X2 X3 / scale=Pearson;
   assess var=(LogX1) / resample=10000
                        seed=603708000;
run;
ods graphics off;
```

The revised model fit is shown in Output 37.8.7, the p-value from the simulation is 0.4777, and the cumulative residuals plotted in Output 37.8.8 show no systematic trend. The log transformation for X1 is more appropriate. Under the revised model, the p-values for testing the functional forms of X2 and X3 are 0.20 and 0.63, respectively; and the p-value for testing the linearity of the model is 0.65. Thus, the revised model seems reasonable.

Output 37.8.7 Multiple Regression Model with Log(X1)

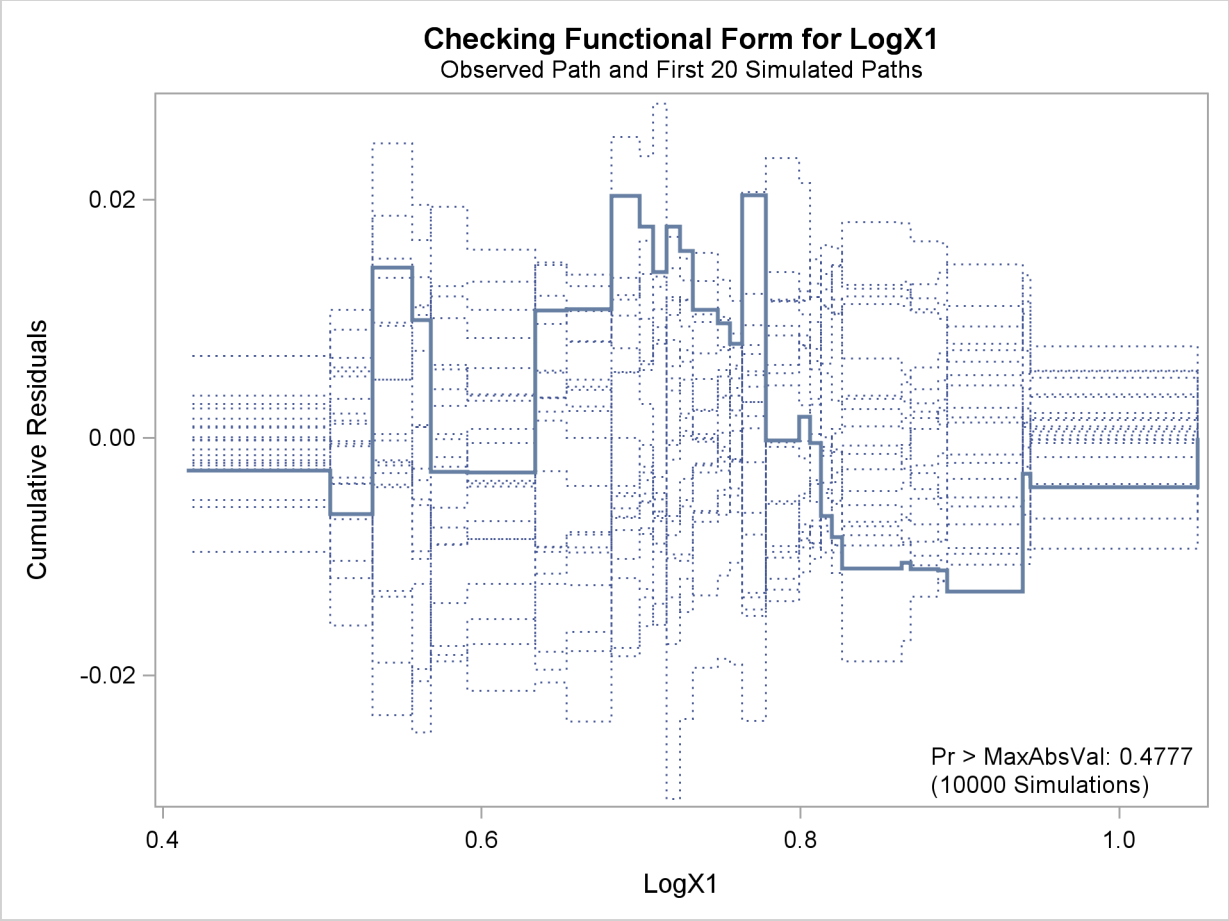
The GENMOD Procedure

Analysis Of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	0.1844	0.0504	0.0857	0.2832	13.41	0.0003
LogX1	1	0.9121	0.0491	0.8158	1.0083	345.05	<.0001
X2	1	0.0095	0.0004	0.0088	0.0102	728.62	<.0001
X3	1	0.0096	0.0003	0.0090	0.0101	1139.73	<.0001
Scale	0	0.0434	0.0000	0.0434	0.0434		

NOTE: The scale parameter was estimated by the square root of Pearson's Chi-Square/DOF.

Output 37.8.8 Cumulative Residual Plot with Log(X1)



Example 37.9: Assessment of a Marginal Model for Dependent Data

This example illustrates the use of cumulative residuals to assess the adequacy of a marginal model for dependent data fit by generalized estimating equations (GEEs). The assessment methods are applied to CD4 count data from an AIDS clinical trial reported by Fischl, Richman, and Hansen (1990) and reanalyzed by Lin, Wei, and Ying (2002). The study randomly assigned 360 HIV patients to the drug AZT and 351 patients to placebo. CD4 counts were measured repeatedly over the course of the study. The data used here are the 4328 measurements taken in the first 40 weeks of the study.

The analysis focuses on the time trend of the response. The first model considered is

$$E(y_{ik}) = \beta_0 + \beta_1 T_{ik} + \beta_2 T_{ik}^2 + \beta_3 R_i T_{ik} + \beta_4 R_i T_{ik}^2$$

where T_{ik} is the time (in weeks) of the kth measurement on the ith patient, y_{ik} is the CD4 count at T_{ik} for the ith patient, and R_i is the indicator of AZT for the ith patient. Normal errors and an independent working correlation are assumed.

The following statements create the SAS data set cd4:

```
data cd4;
   input Id Y Time Time2 TrtTime TrtTime2;
   Time3 = Time2 * Time;
   TrtTime3 = TrtTime2 * Time;
   datalines;
1     264.00024       -0.28571        0.08163       -0.28571        0.08163
1     175.00070        4.14286       17.16327        4.14286       17.16327
1     306.00150        8.14286       66.30612        8.14286       66.30612
1     331.99835       12.14286      147.44898       12.14286      147.44898
1     309.99929       16.14286      260.59184       16.14286      260.59184
1     185.00077       28.71429      824.51020       28.71429      824.51020
1     175.00070       40.14286     1611.44898       40.14286     1611.44898

... more lines ...

711     488.00224       12.14286      147.44898       12.14286      147.44898
711     240.00026       18.14286      329.16327       18.14286      329.16327
;
run;
```

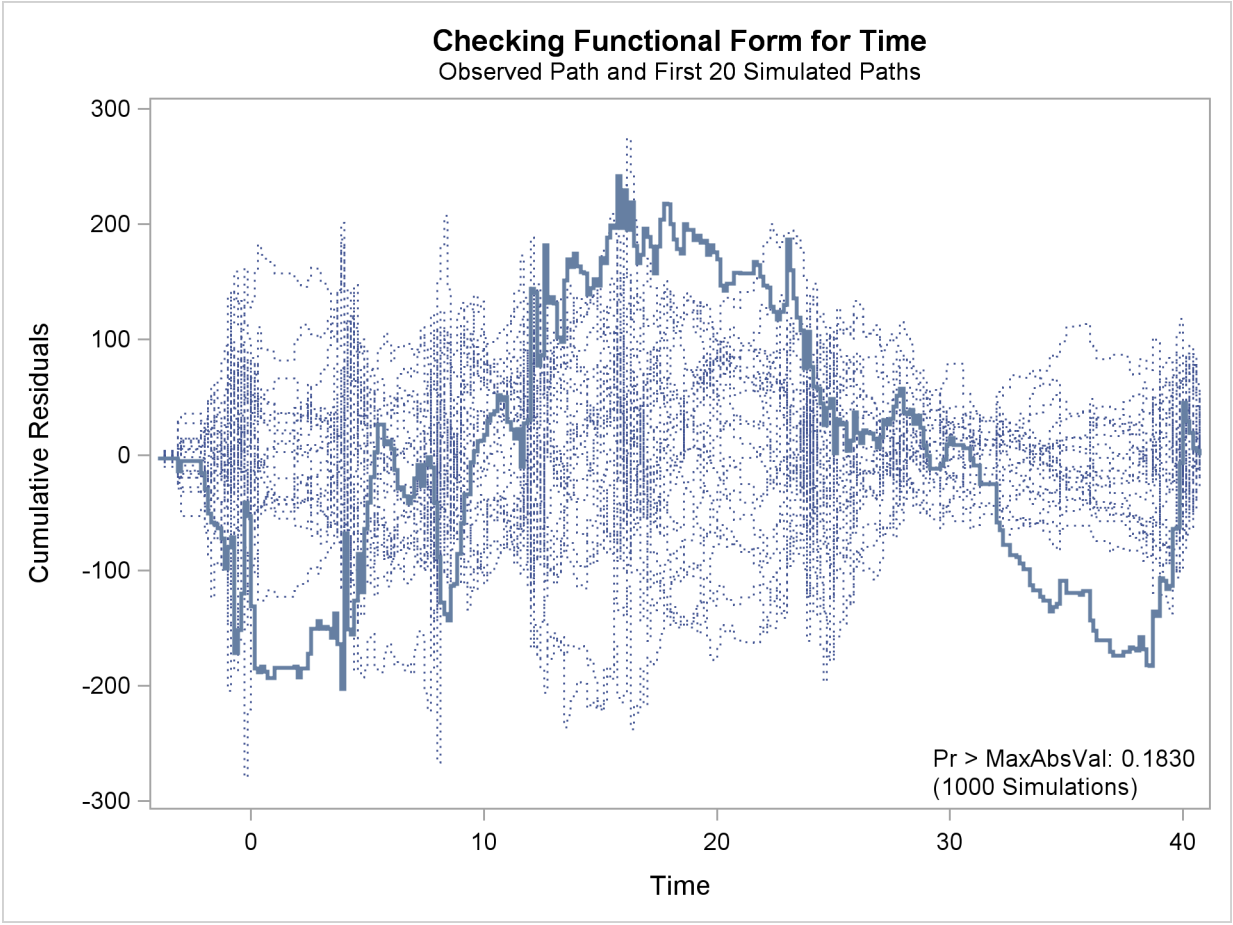
The following SAS statements fit the preceding model, create the cumulative residual plot in Output 37.9.1, and compute a p-value for the model.

These graphical displays are requested by specifying the ODS GRAPHICS statement and the ASSESS statement. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the GENMOD procedure, see the section "ODS Graphics" on page 2022.

Here, the SAS data set variables Time, Time2, TrtTime, and TrtTime2 correspond to T_{ik}, T_{ik}^2, $R_i T_{ik}$, and $R_i T_{ik}^2$, respectively. The variable Id identifies individual patients.

```
ods graphics on;
proc genmod data=cd4;
   class Id;
   model Y = Time Time2 TrtTime TrtTime2;
   repeated sub=Id;
   assess var=(Time) / resample
                       seed=603708000;
run;
ods graphics off;
```

Output 37.9.1 Cumulative Residual Plot for Quadratic Time Fit

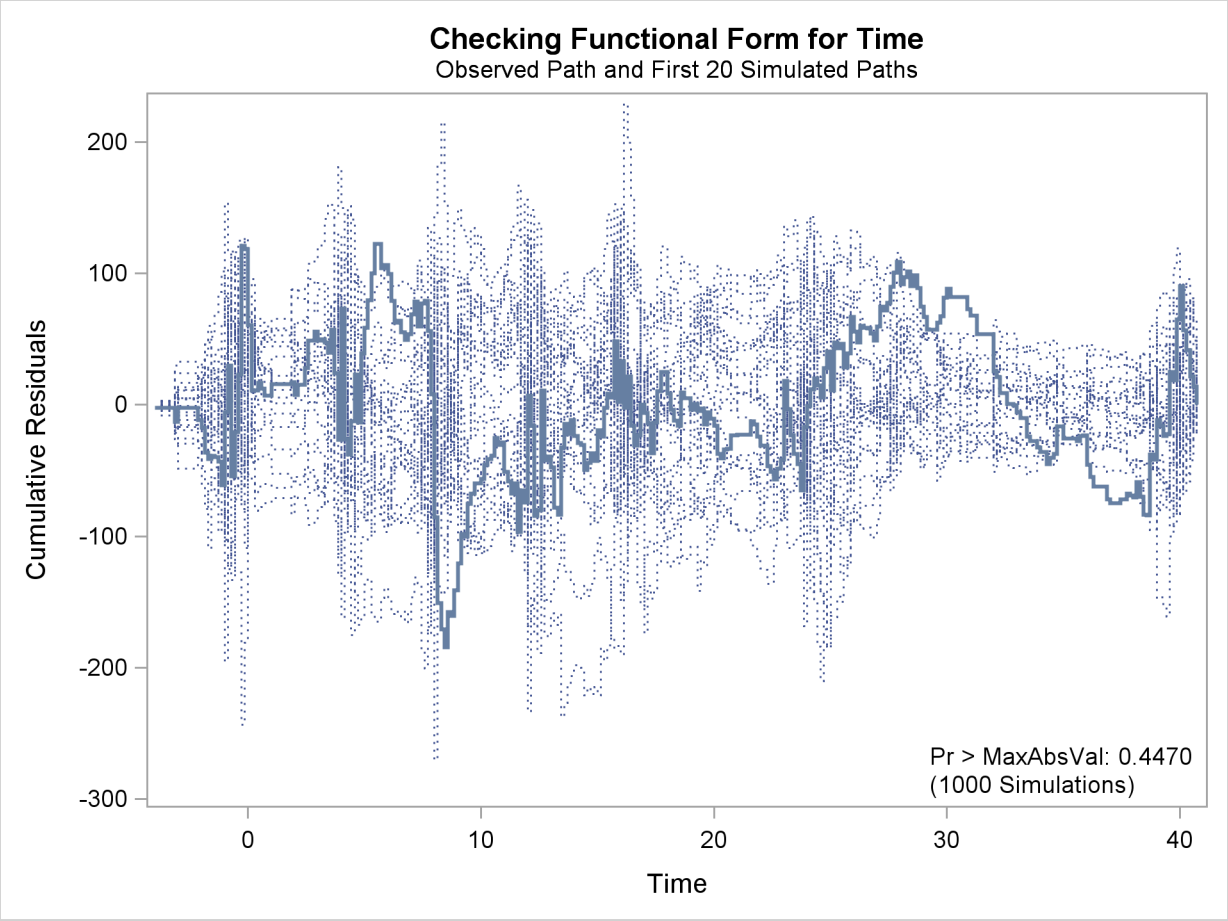


The cumulative residual plot in Output 37.9.1 displays cumulative residuals versus time for the model and 20 simulated realizations. The associated p-value, also shown in Output 37.9.1, is 0.18. These results indicate that a more satisfactory model might be possible. The observed cumulative residual pattern most resembles plot (c) in Output 37.8.6, suggesting cubic time trends.

The following SAS statements fit the model, create the plot in Output 37.9.2, and compute a p-value for a model with the additional terms T_{ik}^3 and $R_i T_{ik}^3$:

```
ods graphics on;
proc genmod data=cd4;
   class Id;
   model Y = Time Time2 Time3 TrtTime TrtTime2 TrtTime3;
   repeated sub=Id;
   assess var=(Time) / resample
                       seed=603708000;
run;
ods graphics off;
```

Output 37.9.2 Cumulative Residual Plot for Cubic Time Fit



The observed cumulative residual pattern appears more typical of the simulated realizations, and the p-value is 0.45, indicating that the model with cubic time trends is more appropriate.

Example 37.10: Bayesian Analysis of a Poisson Regression Model

This example illustrates a Bayesian analysis of a log-linear Poisson regression model. Consider the following data on patients from clinical trials. The data set is a subset of the data described in Ibrahim, Chen, and Lipsitz (1999).

```
data Liver;
input X1-X6 Y;
datalines;
19.1358    50.0110     51.000      0        0       1        3
23.5970    18.4959      3.429      0        0       1        9
20.0474    56.7699      3.429      1        1       0        6
28.0277    59.7836      4.000      0        0       1        6
28.6851    74.1589      5.714      1        0       1        1
18.8092    31.0630      2.286      0        1       1       61
28.7201    52.9178     37.286      1        0       1        6
21.3669    61.6603     54.143      0        1       1        6
23.7332    42.2904      0.571      1        0       1       21

... more lines ...

19.1327    65.3425      2.571      1        0       0        1
17.3010    51.4493      4.429      1        0       0        6
;
run;
```

The primary interest is in prediction of the number of cancerous liver nodes when a patient enters the trials, by using six other baseline characteristics. The number of nodes is modeled by a Poisson regression model with the six baseline characteristics as covariates. The response and regression variables are as follows:

Y	Number of Cancerous Liver Nodes
X1	Body Mass Index
X2	Age, in Years
X3	Time Since Diagnosis of Disease, in Weeks
X4	Two Biochemical Markers (each classified as normal=1 or abnormal=0)
X5	Anti Hepatitis B Antigen
X6	Associated Jaundice (yes=1, no=0)

Two analyses are performed using PROC GENMOD. The first analysis uses noninformative normal prior distributions, and the second analysis uses an informative normal prior for one of the regression parameters.

In the following BAYES statement, COEFFPRIOR=NORMAL specifies a noninformative independent normal prior distribution with zero mean and variance 10^6 for each parameter.

The initial analysis is performed using PROC GENMOD to obtain Bayesian estimates of the regression coefficients by using the following SAS statements:

```
ods graphics ON;
proc genmod data=Liver;
   model Y = X1-X6 / dist=Poisson link=log;
   bayes seed=1 coeffprior=normal;
run;
```

Maximum likelihood estimates of the model parameters are computed by default. These are shown in the "Analysis of Maximum Likelihood Parameter Estimates" table in Output 37.10.1.

Output 37.10.1 Maximum Likelihood Parameter Estimates

The GENMOD Procedure

Bayesian Analysis

Analysis Of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits	
Intercept	1	2.4508	0.2284	2.0032	2.8984
X1	1	-0.0044	0.0080	-0.0201	0.0114
X2	1	-0.0135	0.0024	-0.0181	-0.0088
X3	1	-0.0029	0.0022	-0.0072	0.0014
X4	1	-0.2715	0.0795	-0.4272	-0.1157
X5	1	0.3215	0.0832	0.1585	0.4845
X6	1	0.2077	0.0827	0.0456	0.3698
Scale	0	1.0000	0.0000	1.0000	1.0000

NOTE: The scale parameter was held fixed.

Noninformative independent normal prior distributions with zero means and variances of 10^6 were used in the initial analysis. These are shown in Output 37.10.2.

Output 37.10.2 Regression Coefficient Priors

The GENMOD Procedure

Bayesian Analysis

Independent Normal Prior for Regression Coefficients

Parameter	Mean	Precision
Intercept	0	1E-6
X1	0	1E-6
X2	0	1E-6
X3	0	1E-6
X4	0	1E-6
X5	0	1E-6
X6	0	1E-6

Initial values for the Markov chain are listed in the "Initial Values and Seeds" table in Output 37.10.3. The random number seed is also listed so that you can reproduce the analysis. Since no seed was specified, the seed shown was derived from the time of day.

Output 37.10.3 MCMC Initial Values and Seeds

Initial Values of the Chain

Chain	Seed	Intercept	X1	X2	X3	X4
1	1	2.450813	-0.00435	-0.01347	-0.00291	-0.27149

Initial Values of the Chain

X5	X6
0.321507	0.207713

Summary statistics for the posterior sample are displayed in the "Fit Statistics," "Descriptive Statistics for the Posterior Sample," "Interval Statistics for the Posterior Sample," and "Posterior Correlation Matrix" tables in Output 37.10.4, Output 37.10.5, Output 37.10.6, and Output 37.10.7, respectively. Since noninformative prior distributions for the regression coefficients were used, the mean and standard deviations of the posterior distributions for the model parameters are close to the maximum likelihood estimates and standard errors.

Output 37.10.4 Fit Statistics

Fit Statistics

DIC (smaller is better)	829.729
pD (effective number of parameters)	6.966

Output 37.10.5 Descriptive Statistics

The GENMOD Procedure

Bayesian Analysis

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
Intercept	10000	2.4520	0.2268	2.2997	2.4521	2.6053
X1	10000	-0.00473	0.00801	-0.0100	-0.00465	0.000759
X2	10000	-0.0134	0.00236	-0.0150	-0.0134	-0.0118
X3	10000	-0.00309	0.00220	-0.00455	-0.00305	-0.00158
X4	10000	-0.2705	0.0792	-0.3241	-0.2697	-0.2172
X5	10000	0.3193	0.0834	0.2629	0.3180	0.3762
X6	10000	0.2095	0.0834	0.1538	0.2086	0.2653

Output 37.10.6 Interval Statistics

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
Intercept	0.050	2.0169	2.9056	2.0069	2.8923
X1	0.050	−0.0210	0.0106	−0.0212	0.0103
X2	0.050	−0.0181	−0.00878	−0.0181	−0.00885
X3	0.050	−0.00757	0.00109	−0.00764	0.000989
X4	0.050	−0.4250	−0.1132	−0.4232	−0.1119
X5	0.050	0.1552	0.4821	0.1647	0.4905
X6	0.050	0.0477	0.3749	0.0490	0.3758

Output 37.10.7 Posterior Sample Correlation Matrix

Posterior Correlation Matrix

Parameter	Intercept	X1	X2	X3	X4	X5	X6
Intercept	1.000	−0.705	−0.430	−0.046	−0.225	−0.180	−0.415
X1	−0.705	1.000	−0.211	−0.013	−0.068	0.067	0.128
X2	−0.430	−0.211	1.000	−0.006	0.070	0.057	0.118
X3	−0.046	−0.013	−0.006	1.000	0.016	−0.055	−0.089
X4	−0.225	−0.068	0.070	0.016	1.000	−0.011	0.089
X5	−0.180	0.067	0.057	−0.055	−0.011	1.000	−0.042
X6	−0.415	0.128	0.118	−0.089	0.089	−0.042	1.000

Posterior sample autocorrelations for each model parameter are shown in Output 37.10.8. The autocorrelation after 10 lags is negligible for all parameters, indicating good mixing in the Markov chain.

Output 37.10.8 Posterior Sample Autocorrelations

The GENMOD Procedure

Bayesian Analysis

Posterior Autocorrelations

Parameter	Lag 1	Lag 5	Lag 10	Lag 50
Intercept	0.0551	−0.0134	−0.0101	0.0012
X1	0.0894	−0.0054	−0.0080	0.0019
X2	0.1197	−0.0170	0.0061	0.0006
X3	0.0324	−0.0036	−0.0033	−0.0160
X4	0.0309	0.0056	0.0053	0.0115
X5	0.0402	0.0015	−0.0111	0.0123
X6	0.0696	−0.0047	−0.0024	0.0006

The *p*-values for the Geweke test statistics shown in Output 37.10.9 all indicate convergence of the MCMC. See the section "Assessing Markov Chain Convergence" on page 156 for more information about convergence diagnostics and their interpretation.

Output 37.10.9 Geweke Diagnostic Statistics

Geweke Diagnostics

Parameter	z	Pr > |z|
Intercept	0.9855	0.3244
X1	−1.0835	0.2786
X2	−0.3847	0.7005
X3	0.6715	0.5019
X4	0.1328	0.8943
X5	1.0698	0.2847
X6	−0.1647	0.8692

The effective sample sizes for each parameter are shown in Output 37.10.10.

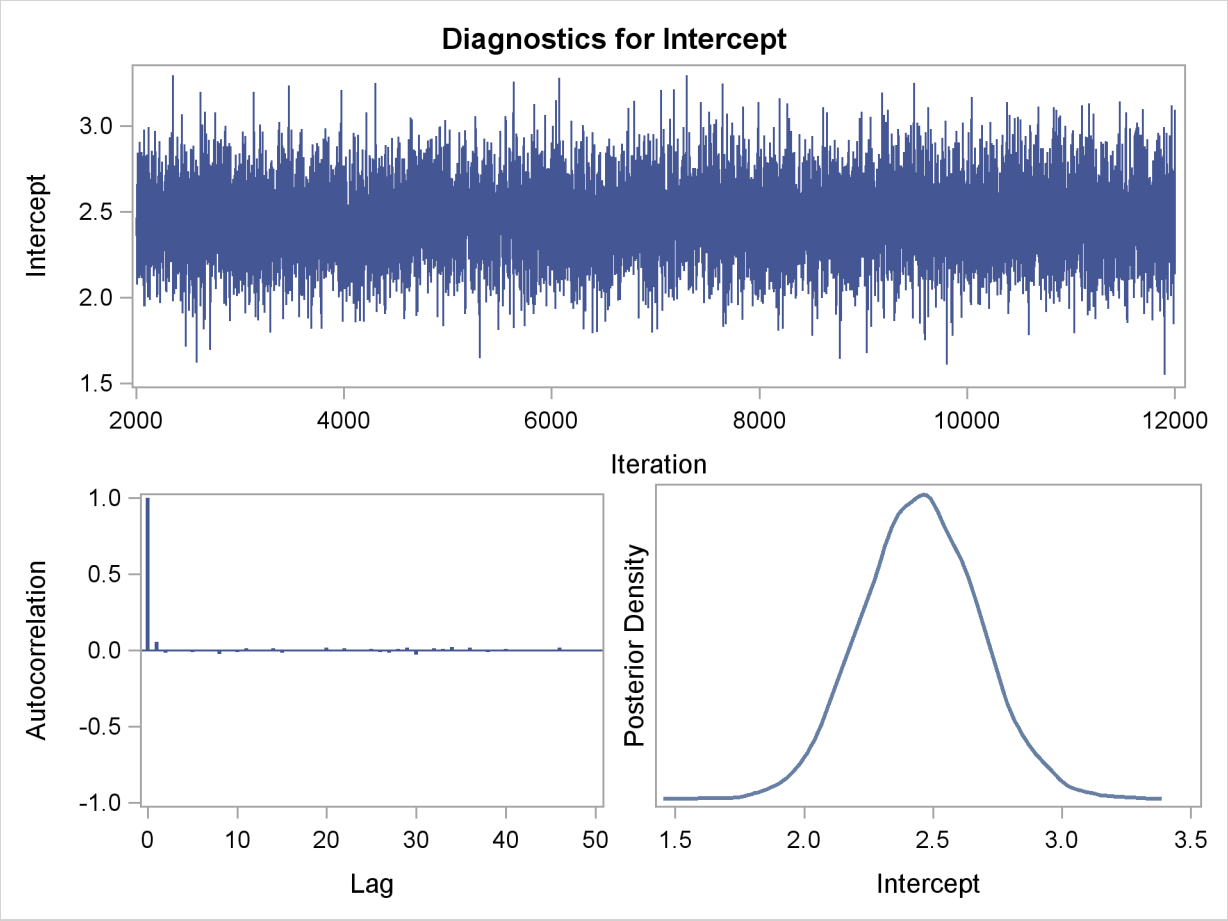
Output 37.10.10 Effective Sample Sizes

Effective Sample Sizes

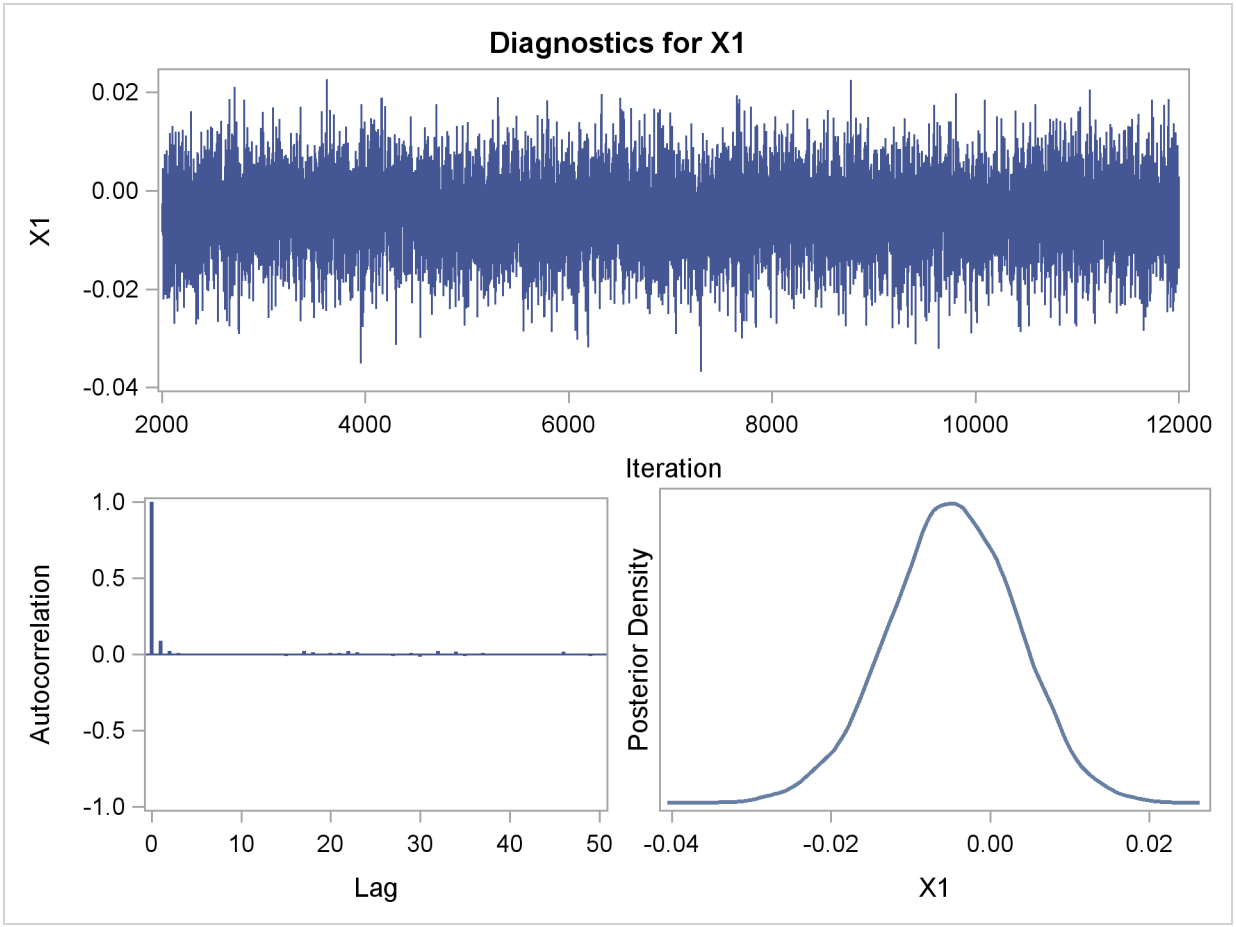
Parameter	ESS	Correlation Time	Efficiency
Intercept	9245.8	1.0816	0.9246
X1	8179.5	1.2226	0.8179
X2	8067.8	1.2395	0.8068
X3	9390.6	1.0649	0.9391
X4	9157.6	1.0920	0.9158
X5	9665.2	1.0346	0.9665
X6	8778.7	1.1391	0.8779

Trace, autocorrelation, and density plots for the seven model parameters are shown in Output 37.10.11 through Output 37.10.17. All indicate satisfactory convergence of the Markov chain.

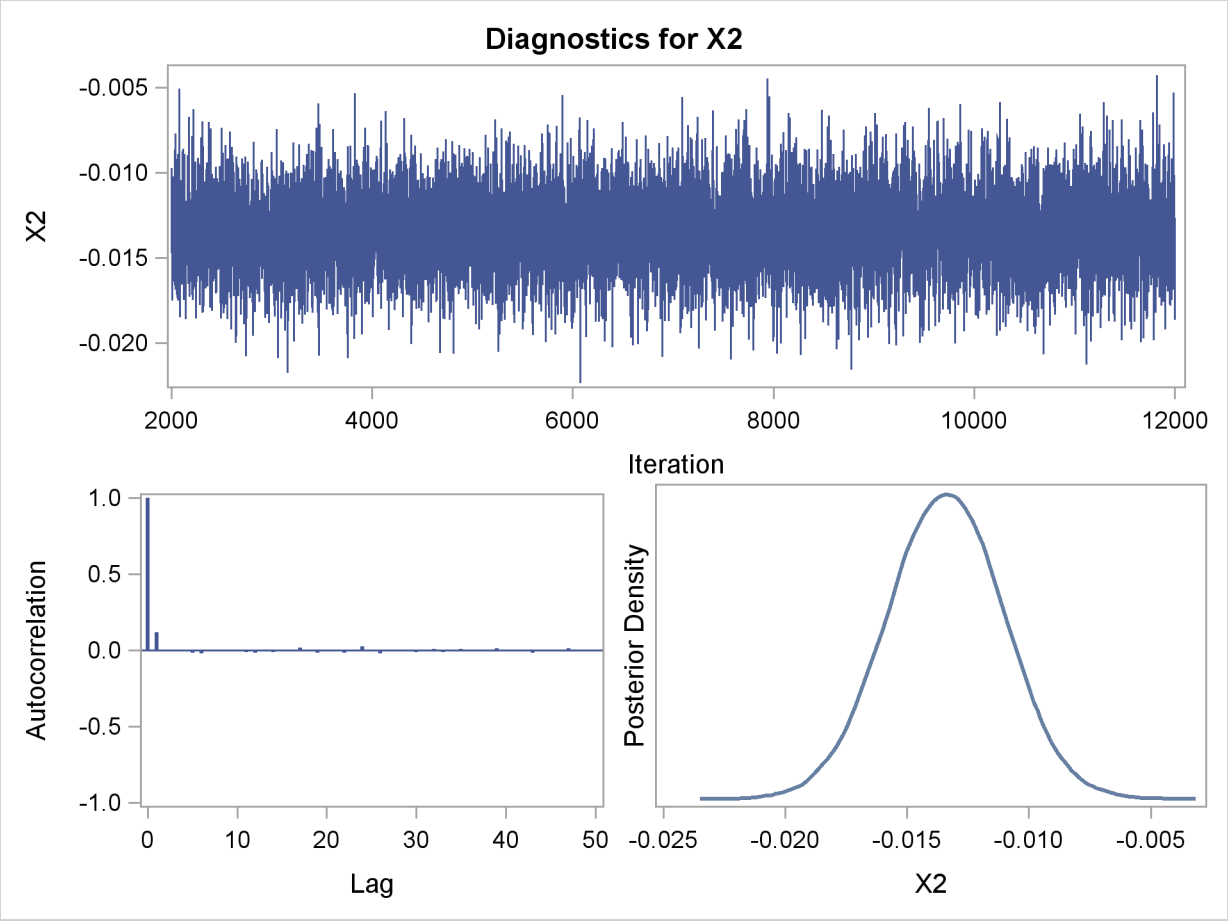
Output 37.10.11 Diagnostic Plots for Intercept



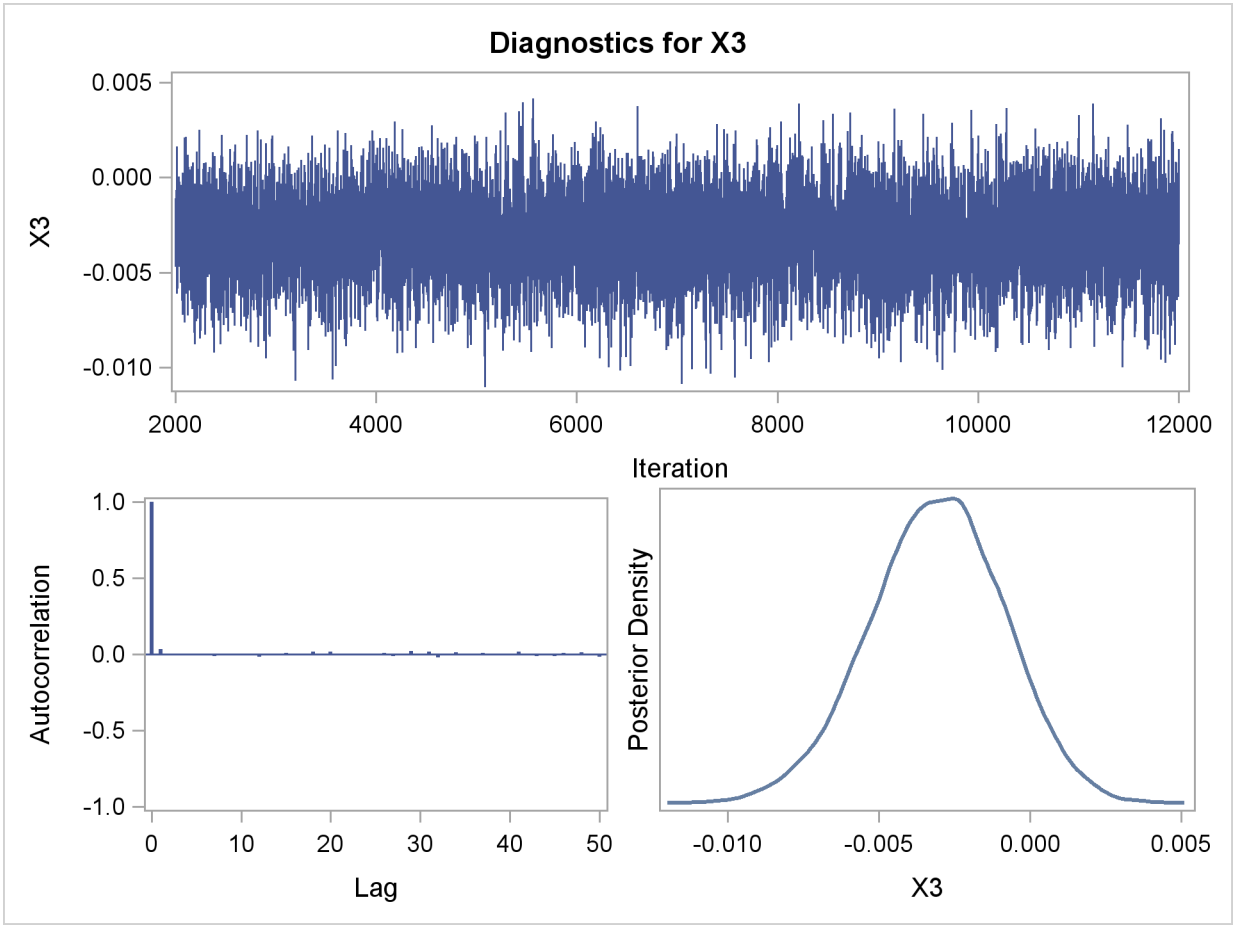
Output 37.10.12 Diagnostic Plots for X1



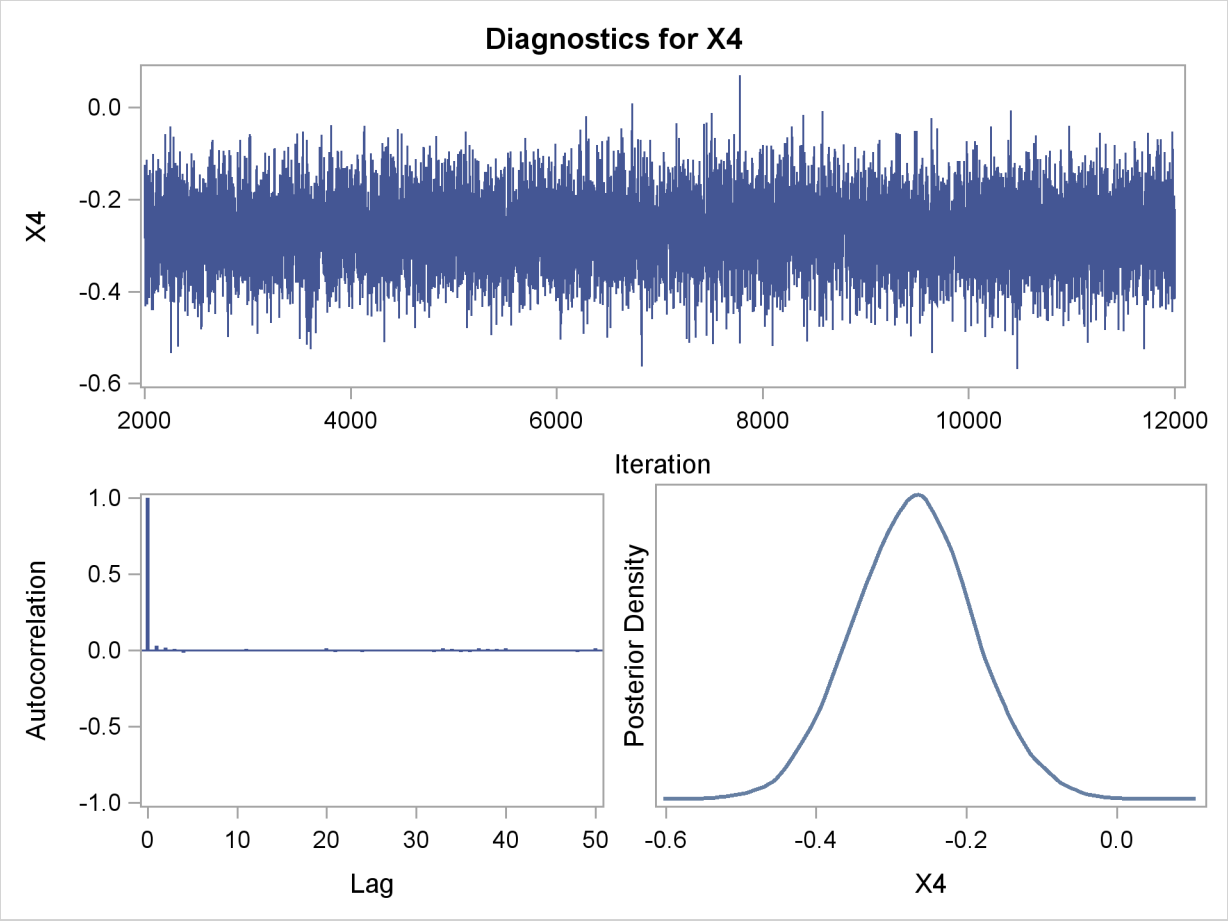
Output 37.10.13 Diagnostic Plots for X2



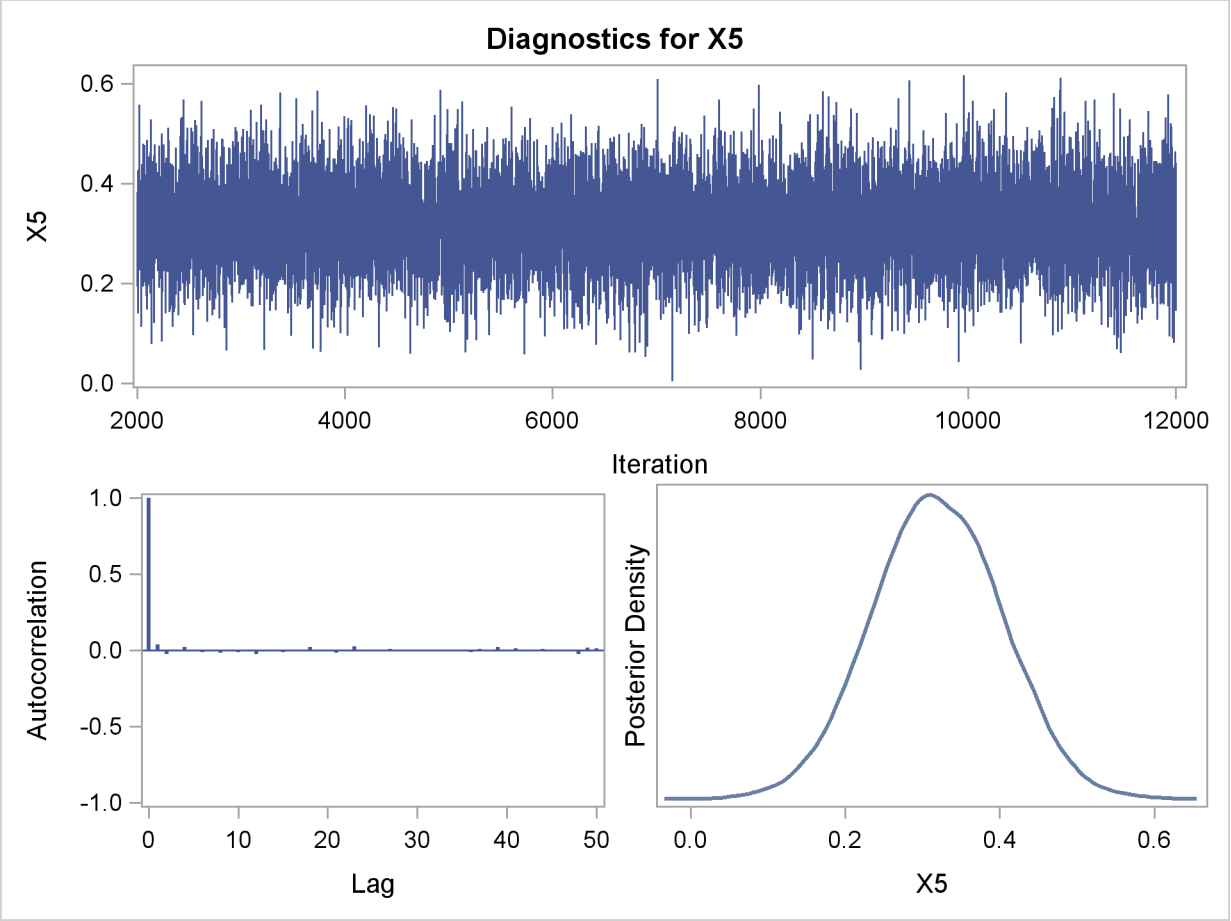
Output 37.10.14 Diagnostic Plots for X3



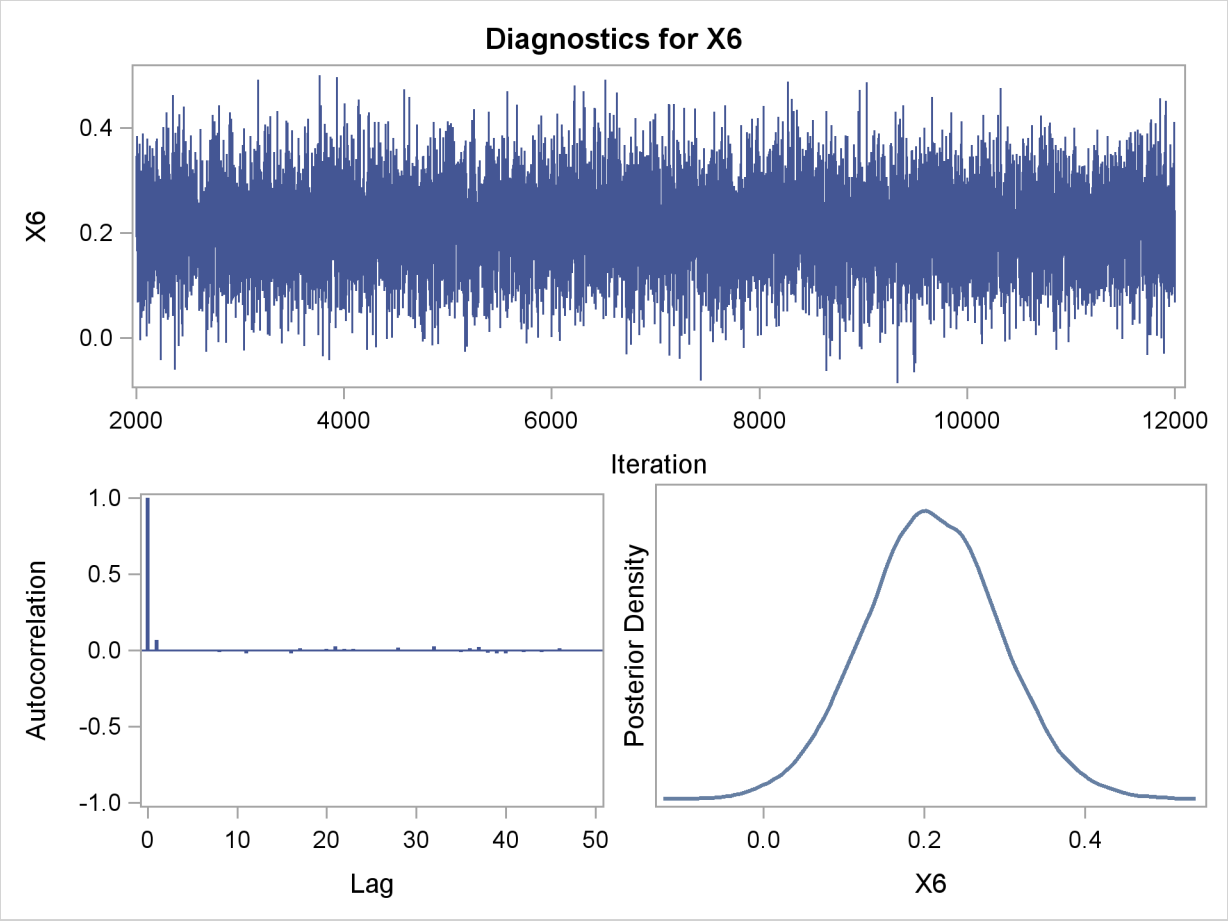
Output 37.10.15 Diagnostic Plots for X4



Output 37.10.16 Diagnostic Plots for X5



Output 37.10.17 Diagnostic Plots for X6



In order to illustrate the use of an informative prior distribution, suppose that researchers expect that a unit increase in body mass index (X1) will be associated with an increase in the mean number of nodes of between 10% and 20%, and they want to incorporate this prior knowledge in the Bayesian analysis. For log-linear models, the mean and linear predictor are related by $\log(\mu_i) = \boldsymbol{x}_i'\boldsymbol{\beta}$. If $\mathrm{X1}_1$ and $\mathrm{X1}_2$ are two values of body mass index, μ_1 and μ_2 are the two mean values, and all other covariates remain equal for the two values of X1, then

$$\frac{\mu_1}{\mu_2} = \exp(\beta(X1_1 - X1_2))$$

so that for a unit change in X1,

$$\frac{\mu_1}{\mu_2} = \exp(\beta)$$

If $1.1 \leq \frac{\mu_1}{\mu_2} \leq 1.2$, then $1.1 \leq \exp(\beta) \leq 1.2$, or $0.095 \leq \beta \leq 0.182$. This gives you guidance in specifying a prior distribution for the β for body mass index. Taking the mean of the prior normal

distribution to be the midrange of the values of β, and taking $\mu \pm 2\sigma$ to be the extremes of the range, an $N(0.1385, 0.0005)$ is the resulting prior distribution. The second analysis uses this informative normal prior distribution for the coefficient of $X1$ and uses independent noninformative normal priors with zero means and variances equal to 10^6 for the remaining model regression parameters.

In the following BAYES statement, COEFFPRIOR=NORMAL(INPUT=NormalPrior) specifies the normal prior distribution for the regression coefficients with means and variances contained in the data set NormalPrior.

An analysis is performed using PROC GENMOD to obtain Bayesian estimates of the regression coefficients by using the following SAS statements:

```
data NormalPrior;
  input _type_ $ Intercept X1-X6;
  datalines;
Var  1e6   0.0005      1e6      1e6      1e6      1e6      1e6
Mean 0.0   0.1385      0.0      0.0      0.0      0.0      0.0
;
run;
proc genmod data=Liver;
   model Y = X1-X6 / dist=Poisson link=log;
   bayes seed=1 plots=none coeffprior=normal(input=NormalPrior) ;
run;
ods graphics off;
```

The prior distributions for the regression parameters are shown in Output 37.10.18.

Output 37.10.18 Regression Coefficient Priors

The GENMOD Procedure

Bayesian Analysis

Independent Normal Prior for Regression Coefficients

Parameter	Mean	Precision
Intercept	0	1E-6
X1	0.1385	2000
X2	0	1E-6
X3	0	1E-6
X4	0	1E-6
X5	0	1E-6
X6	0	1E-6

Initial values for the MCMC are shown in Output 37.10.19. The initial values of the covariates are joint estimates of their posterior modes. The prior distribution for X1 is informative, so the initial value of X1 is further from the MLE than the rest of the covariates. Initial values for the rest of the covariates are close to their MLEs, since noninformative prior distributions were specified for them.

Output 37.10.19 MCMC Initial Values and Seeds

Initial Values of the Chain

Chain	Seed	Intercept	X1	X2	X3	X4
1	1	2.14282	0.010595	−0.01434	−0.00301	−0.28062

Initial Values of the Chain

X5	X6
0.334983	0.231213

Goodness-of-fit, summary, and interval statistics are shown in Output 37.10.20. Except for X1, the statistics shown in Output 37.10.20 are very similar to the previous statistics for noninformative priors shown in Output 37.10.4 through Output 37.10.7. The point estimate for X1 is now positive. This is expected because the prior distribution on β_1 is quite informative. The distribution reflects the belief that the coefficient is positive. The $N(0.1385, 0.0005)$ distribution places the majority of its probability density on positive values. As a result, the posterior density of β_1 places more likelihood on positive values than in the noninformative case.

Output 37.10.20 Fit Statistics

Fit Statistics

DIC (smaller is better)	833.134
pD (effective number of parameters)	6.861

The GENMOD Procedure

Bayesian Analysis

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
Intercept	10000	2.1393	0.2160	1.9929	2.1417	2.2845
X1	10000	0.0104	0.00685	0.00583	0.0106	0.0151
X2	10000	−0.0143	0.00236	−0.0159	−0.0143	−0.0127
X3	10000	−0.00318	0.00218	−0.00463	−0.00313	−0.00170
X4	10000	−0.2801	0.0798	−0.3342	−0.2807	−0.2258
X5	10000	0.3336	0.0834	0.2772	0.3337	0.3902
X6	10000	0.2333	0.0822	0.1791	0.2327	0.2892

Output 37.10.20 *continued*

Posterior Intervals					
Parameter	Alpha	Equal-Tail Interval		HPD Interval	
Intercept	0.050	1.7161	2.5599	1.7075	2.5507
X1	0.050	-0.00323	0.0236	-0.00264	0.0241
X2	0.050	-0.0189	-0.00960	-0.0189	-0.00972
X3	0.050	-0.00754	0.00101	-0.00754	0.000963
X4	0.050	-0.4348	-0.1223	-0.4311	-0.1196
X5	0.050	0.1705	0.4970	0.1661	0.4915
X6	0.050	0.0696	0.3968	0.0655	0.3904

References

Agresti, A. (2002), *Categorical Data Analysis*, Second Edition, New York: John Wiley & Sons.

Aitkin, M., Anderson, D., Francis, B., and Hinde, J. (1989), *Statistical Modelling in GLIM*, Oxford: Oxford Science Publications.

Akaike, H. (1979), "A Bayesian Extension of the Minimum AIC Procedure of Autoregressive Model Fitting," *Biometrika*, 66, 237–242.

Akaike, H. (1981), "Likelihood of a Model and Information Criteria," *Journal of Econometrics*, 16, 3–14.

Boos, D. (1992), "On Generalized Score Tests," *The American Statistician*, 46, 327–333.

Cameron, A. C. and Trivedi, P. K. (1998), *Regression Analysis of Count Data*, Cambridge: Cambridge University Press.

Carey, V., Zeger, S. L., and Diggle, P. (1993), "Modelling Multivariate Binary Data with Alternating Logistic Regressions," *Biometrika*, 80, 517–526.

Cook, R. D. and Weisberg, S. (1982), *Residuals and Influence in Regression*, New York: Chapman & Hall.

Davison, A. C. and Snell, E. J. (1991), "Residuals and Diagnostics," in D. V. Hinkley, N. Reid, and E. J. Snell, eds., *Statistical Theory and Modelling*, London: Chapman & Hall.

Diggle, P. J., Liang, K. Y., and Zeger, S. L. (1994), *Analysis of Longitudinal Data*, Oxford: Clarendon Press.

Dobson, A. (1990), *An Introduction to Generalized Linear Models*, London: Chapman & Hall.

Firth, D. (1991), "Generalized Linear Models," in D. V. Hinkley, N. Reid, and E. J. Snell, eds., *Statistical Theory and Modelling*, London: Chapman & Hall.

Fischl, M. A., Richman, D. D., and Hansen, N. (1990), "The Safety and Efficacy of Zidovudine (AZT) in the Treatment of Subjects with Mildly Symptomatic Human Immunodeficiency Virus Type I (HIV) Infection," *Annals of Internal Medicine*, 112, 727–737.

Gilks, W. (2003), "Adaptive Metropolis Rejection Sampling (ARMS)," software from MRC Biostatistics Unit, Cambridge, UK, http://www.maths.leeds.ac.uk/~wally.gilks/adaptive.rejection/web_page/Welcome.html.

Gilks, W. R., Best, N. G., and Tan, K. K. C. (1995), "Adaptive Rejection Metropolis Sampling with Gibbs Sampling," *Applied Statistics*, 44, 455–472.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996), *Markov Chain Monte Carlo in Practice*, London: Chapman & Hall.

Gilks, W. R. and Wild, P. (1992), "Adaptive Rejection Sampling for Gibbs Sampling," *Applied Statistics*, 41, 337–348.

Hardin, J. W. and Hilbe, J. M. (2003), *Generalized Estimating Equations*, Boca Raton, FL: Chapman & Hall/CRC.

Hilbe, J. (1994), "Log Negative Binomial Regression Using the GENMOD Procedure," in *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Ibrahim, J. G., Chen, M.-H., and Lipsitz, S. R. (1999), "Monte Carlo EM for Missing Covariates in Parametric Regression Models," *Biometrics*, 55, 591–596.

Ibrahim, J. G., Chen, M. H., and Sinha, D. (2001), *Bayesian Survival Analysis*, New York: Springer-Verlag.

Ibrahim, J. G. and Laud, P. W. (1991), "On Bayesian Analysis of Generalized Linear Models Using Jeffreys' Prior," *Journal of the American Statistical Association*, 86, 981–986.

Lawless, J. E. (1987), "Negative Binomial and Mixed Poisson Regression," *The Canadian Journal of Statistics*, 15, 209–225.

Lawless, J. F. (2003), *Statistical Model and Methods for Lifetime Data*, Second Edition, New York: John Wiley & Sons.

Liang, K. Y. and Zeger, S. L. (1986), "Longitudinal Data Analysis Using Generalized Linear Models," *Biometrika*, 73, 13–22.

Lin, D. Y., Wei, L. J., and Ying, Z. (2002), "Model-Checking Techniques Based on Cumulative Residuals," *Biometrics*, 58, 1–12.

Lipsitz, S. H., Fitzmaurice, G. M., Orav, E. J., and Laird, N. M. (1994), "Performance of Generalized Estimating Equations in Practical Situations," *Biometrics*, 50, 270–278.

Lipsitz, S. H., Kim, K., and Zhao, L. (1994), "Analysis of Repeated Categorical Data Using Generalized Estimating Equations," *Statistics in Medicine*, 13, 1149–1163.

Littell, R. C., Freund, R. J., and Spector, P. C. (1991), *SAS System for Linear Models*, Third Edition, Cary, NC: SAS Institute Inc.

Long, J. S. (1997), *Regression Models for Categorical and Limited Dependent Variables*, Thousand Oaks, CA: Sage Publications.

McCullagh, P. (1983), "Quasi-likelihood Functions," *Annals of Statistics*, 11, 59–67.

McCullagh, P. and Nelder, J. A. (1989), *Generalized Linear Models*, Second Edition, London: Chapman & Hall.

Miller, M. E., Davis, C. S., and Landis, J. R. (1993), "The Analysis of Longitudinal Polytomous Data: Generalized Estimating Equations and Connections with Weighted Least Squares," *Biometrics*, 49, 1033–1044.

Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized Linear Models," *Journal of the Royal Statistical Society, Series A*, 135, 370–384.

Nelson, W. (1982), *Applied Life Data Analysis*, New York: John Wiley & Sons.

Neter, J., Kutner, M. H., Nachtsheim, C. J., and Wasserman, W. (1996), *Applied Linear Statistical Models*, Fourth Edition, Chicago: Irwin.

Pan, W. (2001), "Akaike's Information Criterion in Generalized Estimating Equations," *Biometrics*, 57, 120–125.

Pregibon, D. (1981), "Logistic Regression Diagnostics," *Annals of Statistics*, 9, 705–724.

Preisser, J. S. and Qaqish, B. F. (1996), "Deletion Diagnostics for Generalised Estimating Equations," *Biometrika*, 83, 551–562.

Rao, C. R. (1973), *Linear Statistical Inference*, New York: John Wiley & Sons.

Rotnitzky, A. and Jewell, N. P. (1990), "Hypothesis Testing of Regression Parameters in Semiparametric Generalized Linear Models for Cluster Correlated Data," *Biometrika*, 77, 485–497.

Royall, R. M. (1986), "Model Robust Inference Using Maximum Likelihood Estimators," *International Statistical Review*, 54, 221–226.

Searle, S. R. (1971), *Linear Models*, New York: John Wiley & Sons.

Simonoff, J. S. (2003), *Analyzing Categorical Data*, New York: Springer-Verlag.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van der Linde, A. (2002), "Bayesian Measures of Model Complexity and Fit," *Journal of the Royal Statistical Society, Series B*, 64(4), 583–616, with discussion.

Stokes, M. E., Davis, C. S., and Koch, G. G. (2000), *Categorical Data Analysis Using the SAS System*, Second Edition, Cary, NC: SAS Institute Inc.

Thall, P. F. and Vail, S. C. (1990), "Some Covariance Models for Longitudinal Count Data with Overdispersion," *Biometrics*, 46, 657–671.

Ware, J. H., Dockery, S. A. I., Speizer, F. E., and Ferris, B. G., Jr. (1984), "Passive Smoking, Gas Cooking, and Respiratory Health of Children Living in Six Cities," *American Review of Respiratory Diseases*, 129, 366–374.

White, H. (1982), "Maximum Likelihood Estimation of Misspecified Models," *Econometrica*, 50, 1–25.

Williams, D. A. (1987), "Generalized Linear Model Diagnostics Using the Deviance and Single Case Deletions," *Applied Statistics*, 36, 181–191.

Zeger, S. L., Liang, K. Y., and Albert, P. S. (1988), "Models for Longitudinal Data: A Generalized Estimating Equation Approach," *Biometrics*, 44, 1049–1060.

Chapter 38

The GLIMMIX Procedure

Contents

Overview: GLIMMIX Procedure **2080**
Basic Features 2081
Assumptions 2082
Notation for the Generalized Linear Mixed Model 2083
The Basic Model 2083
G-Side and R-Side Random Effects and Covariance Structures . . . 2083
Relationship with Generalized Linear Models 2084
PROC GLIMMIX Contrasted with Other SAS Procedures 2085
Getting Started: GLIMMIX Procedure **2086**
Logistic Regressions with Random Intercepts 2086
Syntax: GLIMMIX Procedure **2093**
PROC GLIMMIX Statement 2094
BY Statement 2121
CLASS Statement 2122
CONTRAST Statement 2123
COVTEST Statement 2127
EFFECT Statement (Experimental) 2134
ESTIMATE Statement 2134
FREQ Statement 2140
ID Statement 2140
LSMEANS Statement 2141
LSMESTIMATE Statement 2155
MODEL Statement 2161
Response Variable Options 2163
Model Options 2164
NLOPTIONS Statement 2175
OUTPUT Statement 2176
PARMS Statement 2181
RANDOM Statement 2186
WEIGHT Statement 2206
Programming Statements 2207
User-Defined Link or Variance Function 2208
Implied Variance Functions 2208
Automatic Variables 2209

Details: GLIMMIX Procedure **2212**
Generalized Linear Models Theory 2212
Maximum Likelihood 2212
Scale and Dispersion Parameters 2215
Quasi-likelihood for Independent Data 2216
Effects of Adding Overdispersion 2217
Generalized Linear Mixed Models Theory 2218
Model or Integral Approximation 2218
Pseudo-likelihood Estimation Based on Linearization 2220
Maximum Likelihood Estimation Based on Laplace Approximation 2224
Maximum Likelihood Estimation Based on Adaptive Quadrature . . 2227
Aspects Common to Adaptive Quadrature and Laplace Approximation 2230
Notes on Bias of Estimators 2232
GLM Mode or GLMM Mode 2233
Statistical Inference for Covariance Parameters 2234
The Likelihood Ratio Test 2234
One- and Two-Sided Testing, Mixture Distributions 2235
Handling the Degenerate Distribution 2237
Wald Versus Likelihood Ratio Tests 2238
Confidence Bounds Based on Likelihoods 2238
Satterthwaite Degrees of Freedom Approximation 2241
Empirical Covariance (“Sandwich”) Estimators 2243
Residual-Based Estimators 2243
Design-Adjusted MBN Estimator 2244
Exploring and Comparing Covariance Matrices 2246
Processing by Subjects 2247
Radial Smoothing Based on Mixed Models 2250
From Penalized Splines to Mixed Models 2250
Knot Selection 2251
Odds and Odds Ratio Estimation 2256
The Odds Ratio Estimates Table 2257
Odds or Odds Ratio 2259
Odds Ratios in Multinomial Models 2260
Parameterization of Generalized Linear Mixed Models 2261
Intercept 2261
Interaction Effects 2261
Nested Effects 2261
Implications of the Non-Full-Rank Parameterization 2262
Missing Level Combinations 2262
Notes on the EFFECT Statement 2262
Positional and Nonpositional Syntax for Contrast Coefficients . . . 2264
Response-Level Ordering and Referencing 2267
Comparing the GLIMMIX and MIXED Procedures 2268
Singly or Doubly Iterative Fitting 2271

Default Estimation Techniques . . . 2273
Default Output . . . 2273
Model Information . . . 2274
Class Level Information . . . 2274
Number of Observations . . . 2274
Response Profile . . . 2274
Dimensions . . . 2275
Optimization Information . . . 2275
Iteration History . . . 2275
Convergence Status . . . 2276
Fit Statistics . . . 2276
Covariance Parameter Estimates . . . 2277
Type III Tests of Fixed Effects . . . 2278
Notes on Output Statistics . . . 2278
ODS Table Names . . . 2280
ODS Graphics . . . 2282
Diagnostic Plots . . . 2282
Graphics for LS-Mean Comparisons . . . 2288
ODS Graph Names . . . 2299
Examples: GLIMMIX Procedure . . . **2300**
Example 38.1: Binomial Counts in Randomized Blocks . . . 2300
Example 38.2: Mating Experiment with Crossed Random Effects . . . 2312
Example 38.3: Smoothing Disease Rates; Standardized Mortality Ratios . . . 2320
Example 38.4: Quasi-likelihood Estimation for Proportions with Unknown Distribution . . . 2330
Example 38.5: Joint Modeling of Binary and Count Data . . . 2338
Example 38.6: Radial Smoothing of Repeated Measures Data . . . 2345
Example 38.7: Isotonic Contrasts for Ordered Alternatives . . . 2357
Example 38.8: Adjusted Covariance Matrices of Fixed Effects . . . 2358
Example 38.9: Testing Equality of Covariance and Correlation Matrices . . . 2364
Example 38.10: Multiple Trends Correspond to Multiple Extrema in Profile Likelihoods . . . 2372
Example 38.11: Maximum Likelihood in Proportional Odds Model with Random Effects . . . 2378
Example 38.12: Fitting a Marginal (GEE-Type) Model . . . 2384
Example 38.13: Response Surface Comparisons with Multiplicity Adjustments 2390
Example 38.14: Generalized Poisson Mixed Model for Overdispersed Count Data . . . 2398
Example 38.15: Comparing Multiple B-Splines . . . 2406
Example 38.16: Diallel Experiment with Multimember Random Effects . . . 2412
Example 38.17: Linear Inference Based on Summary Data . . . 2415
References . . . **2421**

Overview: GLIMMIX Procedure

The GLIMMIX procedure fits statistical models to data with correlations or nonconstant variability and where the response is not necessarily normally distributed. These models are known as generalized linear mixed models (GLMM).

GLMMs, like linear mixed models, assume normal (Gaussian) random effects. Conditional on these random effects, data can have any distribution in the exponential family. The exponential family comprises many of the elementary discrete and continuous distributions. The binary, binomial, Poisson, and negative binomial distributions, for example, are discrete members of this family. The normal, beta, gamma, and chi-square distributions are representatives of the continuous distributions in this family. In the absence of random effects, the GLIMMIX procedure fits generalized linear models (fit by the GENMOD procedure).

GLMMs are useful for the following applications:

- estimating trends in disease rates
- modeling CD4 counts in a clinical trial over time
- modeling the proportion of infected plants on experimental units in a design with randomly selected treatments or randomly selected blocks
- predicting the probability of high ozone levels in counties
- modeling skewed data over time
- analyzing customer preference
- joint modeling of multivariate outcomes

Such data often display correlations among some or all observations as well as nonnormality. The correlations can arise from repeated observation of the same sampling units, shared random effects in an experimental design, spatial (temporal) proximity, multivariate observations, and so on.

The GLIMMIX procedure does not fit hierarchical models with nonnormal random effects. With the GLIMMIX procedure you select the distribution of the response variable conditional on normally distributed random effects.

For more information about the differences between the GLIMMIX procedure and SAS procedures that specialize in certain subsets of the GLMM models, see the section "PROC GLIMMIX Contrasted with Other SAS Procedures" on page 2085.

Basic Features

The GLIMMIX procedure enables you to specify a generalized linear mixed model and to perform confirmatory inference in such models. The syntax is similar to that of the MIXED procedure and includes CLASS, MODEL, and RANDOM statements. For instructions on how to specify PROC MIXED REPEATED effects with PROC GLIMMIX, see the section "Comparing the GLIMMIX and MIXED Procedures" on page 2268. The following are some of the basic features of PROC GLIMMIX.

- SUBJECT= and GROUP= options, which enable blocking of variance matrices and parameter heterogeneity
- choice of linearization approach or integral approximation by quadrature or Laplace method for mixed models with nonlinear random effects or nonnormal distribution
- choice of linearization about expected values or expansion about current solutions of best linear unbiased predictors
- flexible covariance structures for random and residual random effects, including variance components, unstructured, autoregressive, and spatial structures
- CONTRAST, ESTIMATE, LSMEANS, and LSMESTIMATE statements, which produce hypothesis tests and estimable linear combinations of effects
- NLOPTIONS statement, which enables you to exercise control over the numerical optimization. You can choose techniques, update methods, line search algorithms, convergence criteria, and more. Or, you can choose the default optimization strategies selected for the particular class of model you are fitting.
- computed variables with SAS programming statements inside of PROC GLIMMIX (except for variables listed in the CLASS statement). These computed variables can appear in the MODEL, RANDOM, WEIGHT, or FREQ statement.
- grouped data analysis
- user-specified link and variance functions
- choice of model-based variance-covariance estimators for the fixed effects or empirical (sandwich) estimators to make analysis robust against misspecification of the covariance structure and to adjust for small-sample bias
- joint modeling for multivariate data. For example, you can model binary and normal responses from a subject jointly and use random effects to relate (fuse) the two outcomes.
- multinomial models for ordinal and nominal outcomes
- univariate and multivariate low-rank mixed model smoothing

Assumptions

The primary assumptions underlying the analyses performed by PROC GLIMMIX are as follows:

- If the model contains random effects, the distribution of the data conditional on the random effects is known. This distribution is either a member of the exponential family of distributions or one of the supplementary distributions provided by the GLIMMIX procedure. In models without random effects, the unconditional (marginal) distribution is assumed to be known for maximum likelihood estimation, or the first two moments are known in the case of quasi-likelihood estimation.
- The conditional expected value of the data takes the form of a linear mixed model after a monotonic transformation is applied.
- The problem of fitting the GLMM can be cast as a singly or doubly iterative optimization problem. The objective function for the optimization is a function of either the actual log likelihood, an approximation to the log likelihood, or the log likelihood of an approximated model.

For a model containing random effects, the GLIMMIX procedure, by default, estimates the parameters by applying pseudo-likelihood techniques as in Wolfinger and O'Connell (1993) and Breslow and Clayton (1993). In a model without random effects (GLM models), PROC GLIMMIX estimates the parameters by maximum likelihood, restricted maximum likelihood, or quasi-likelihood. See the section "Singly or Doubly Iterative Fitting" on page 2271 about when the GLIMMIX procedure applies noniterative, singly and doubly iterative algorithms, and the section "Default Estimation Techniques" on page 2273 about the default estimation methods. You can also fit generalized linear mixed models by maximum likelihood where the marginal distribution is numerically approximated by the Laplace method (METHOD=LAPLACE) or by adaptive Gaussian quadrature (METHOD=QUAD).

Once the parameters have been estimated, you can perform statistical inferences for the fixed effects and covariance parameters of the model. Tests of hypotheses for the fixed effects are based on Wald-type tests and the estimated variance-covariance matrix. The COVTEST statement enables you to perform inferences about covariance parameters based on likelihood ratio tests.

PROC GLIMMIX uses the Output Delivery System (ODS) for displaying and controlling the output from SAS procedures. ODS enables you to convert any of the output from PROC GLIMMIX into a SAS data set. See the section "ODS Table Names" on page 2280 for more information.

The GLIMMIX procedure now uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the statistical graphics available with the GLIMMIX procedure, see the PLOTS options in the PROC GLIMMIX and LSMEANS statements.

Notation for the Generalized Linear Mixed Model

This section introduces the mathematical notation used throughout the chapter to describe the generalized linear mixed model (GLMM). See the section "Details: GLIMMIX Procedure" on page 2212 for a description of the fitting algorithms and the mathematical-statistical details.

The Basic Model

Suppose $\mathbf{Y}$ represents the $(n \times 1)$ vector of observed data and $\boldsymbol{\gamma}$ is a $(r \times 1)$ vector of random effects. Models fit by the GLIMMIX procedure assume that

$$\mathrm{E}[\mathbf{Y}|\boldsymbol{\gamma}] = g^{-1}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma})$$

where $g(\cdot)$ is a differentiable monotonic link function and $g^{-1}(\cdot)$ is its inverse. The matrix $\mathbf{X}$ is an $(n \times p)$ matrix of rank k, and $\mathbf{Z}$ is an $(n \times r)$ design matrix for the random effects. The random effects are assumed to be normally distributed with mean $\mathbf{0}$ and variance matrix $\mathbf{G}$.

The GLMM contains a linear mixed model inside the inverse link function. This model component is referred to as the linear predictor,

$$\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma}$$

The variance of the observations, conditional on the random effects, is

$$\mathrm{Var}[\mathbf{Y}|\boldsymbol{\gamma}] = \mathbf{A}^{1/2}\mathbf{R}\mathbf{A}^{1/2}$$

The matrix $\mathbf{A}$ is a diagonal matrix and contains the variance functions of the model. The variance function expresses the variance of a response as a function of the mean. The GLIMMIX procedure determines the variance function from the DIST= option in the MODEL statement or from the user-supplied variance function (see the section "Implied Variance Functions" on page 2208). The matrix $\mathbf{R}$ is a variance matrix specified by the user through the RANDOM statement. If the conditional distribution of the data contains an additional scale parameter, it is either part of the variance functions or part of the $\mathbf{R}$ matrix. For example, the gamma distribution with mean μ has the variance function $a(\mu) = \mu^2$ and $\mathrm{Var}[Y|\boldsymbol{\gamma}] = \mu^2\phi$. If your model calls for G-side random effects only (see the next section), the procedure models $\mathbf{R} = \phi\mathbf{I}$, where $\mathbf{I}$ is the identity matrix. Table 38.15 identifies the distributions for which $\phi \equiv 1$.

G-Side and R-Side Random Effects and Covariance Structures

The GLIMMIX procedure distinguishes two types of random effects. Depending on whether the parameters of the covariance structure for random components in your model are contained in $\mathbf{G}$ or in $\mathbf{R}$, the procedure distinguishes between "G-side" and "R-side" random effects. The associated covariance structures of $\mathbf{G}$ and $\mathbf{R}$ are similarly termed the G-side and R-side covariance structure, respectively. R-side effects are also called "residual" effects. Simply put, if a random effect is an element of $\boldsymbol{\gamma}$, it is a G-side effect and you are modeling the G-side covariance structure; otherwise, you are modeling the R-side covariance structure of the model. Models without G-side effects are

also known as marginal (or population-averaged) models. Models fit with the GLIMMIX procedure can have none, one, or more of each type of effect.

Note that an R-side effect in the GLIMMIX procedure is equivalent to a REPEATED effect in the MIXED procedure. The R-side covariance structure in the GLIMMIX procedure is the covariance structure that you would formulate with the REPEATED statement in the MIXED procedure. In the GLIMMIX procedure all random effects and their covariance structures are specified through the RANDOM statement. See the section "Comparing the GLIMMIX and MIXED Procedures" on page 2268 for a comparison of the GLIMMIX and MIXED procedures.

The columns of $\mathbf{X}$ are constructed from effects listed on the right side in the MODEL statement. Columns of $\mathbf{Z}$ and the variance matrices $\mathbf{G}$ and $\mathbf{R}$ are constructed from the RANDOM statement.

The $\mathbf{R}$ matrix is by default the scaled identity matrix, $\mathbf{R} = \phi\mathbf{I}$. The scale parameter ϕ is set to one if the distribution does not have a scale parameter, such as in the case of the binary, binomial, Poisson, and exponential distribution (see Table 38.15). To specify a different $\mathbf{R}$ matrix, use the RANDOM statement with the _RESIDUAL_ keyword or the RESIDUAL option. For example, to specify that the Time effect for each patient is an R-side effect with a first-order autoregressive covariance structure, use the RESIDUAL option:

```
random time / type=ar(1) subject=patient residual;
```

To add a multiplicative overdispersion parameter, use the _RESIDUAL_ keyword:

```
random _residual_;
```

You specify the link function $g(\cdot)$ with the LINK= option in the MODEL statement or with programming statements. You specify the variance function that controls the matrix $\mathbf{A}$ with the DIST= option in the MODEL statement or with programming statements.

Unknown quantities subject to estimation are the fixed-effects parameter vector $\boldsymbol{\beta}$ and the covariance parameter vector $\boldsymbol{\theta}$ that comprises all unknowns in $\mathbf{G}$ and $\mathbf{R}$. The random effects $\boldsymbol{\gamma}$ are not parameters of the model in the sense that they are not estimated. The vector $\boldsymbol{\gamma}$ is a vector of random variables. The solutions for $\boldsymbol{\gamma}$ are predictors of these random variables.

Relationship with Generalized Linear Models

Generalized linear models (Nelder and Wedderburn 1972; McCullagh and Nelder 1989) are a special case of GLMMs. If $\boldsymbol{\gamma} = \mathbf{0}$ and $\mathbf{R} = \phi\mathbf{I}$, the GLMM reduces to either a generalized linear model (GLM) or a GLM with overdispersion. For example, if $\mathbf{Y}$ is a vector of Poisson variables so that $\mathbf{A}$ is a diagonal matrix containing $\mathrm{E}[\mathbf{Y}] = \boldsymbol{\mu}$ on the diagonal, then the model is a Poisson regression model for $\phi = 1$ and overdispersed relative to a Poisson distribution for $\phi > 1$. Because the Poisson distribution does not have an extra scale parameter, you can model overdispersion by adding the following statement to your GLIMMIX program:

```
random _residual_;
```

If the only random effect is an overdispersion effect, PROC GLIMMIX fits the model by (restricted) maximum likelihood and not by one of the methods specific to GLMMs.

PROC GLIMMIX Contrasted with Other SAS Procedures

The GLIMMIX procedure generalizes the MIXED and GENMOD procedures in two important ways. First, the response can have a nonnormal distribution. The MIXED procedure assumes that the response is normally (Gaussian) distributed. Second, the GLIMMIX procedure incorporates random effects in the model and so allows for subject-specific (conditional) and population-averaged (marginal) inference. The GENMOD procedure allows only for marginal inference.

The GLIMMIX and MIXED procedure are closely related; see the syntax and feature comparison in the section "Comparing the GLIMMIX and MIXED Procedures" on page 2268. The remainder of this section compares the GLIMMIX procedure with the GENMOD, NLMIXED, LOGISTIC, and CATMOD procedures.

The GENMOD procedure fits generalized linear models for independent data by maximum likelihood. It can also handle correlated data through the marginal GEE approach of Liang and Zeger (1986) and Zeger and Liang (1986). The GEE implementation in the GENMOD procedure is a marginal method that does not incorporate random effects. The GEE estimation in the GENMOD procedure relies on R-side covariances only, and the unknown parameters in $\mathbf{R}$ are estimated by the method of moments. The GLIMMIX procedure allows G-side random effects and R-side covariances. PROC GLIMMIX can fit marginal (GEE-type) models, but the covariance parameters are not estimated by the method of moments. The parameters are estimated by likelihood-based techniques. When the GLIMMIX and GENMOD procedures fit a generalized linear model where the distribution contains a scale parameter, such as the normal, gamma, inverse gaussian, or negative binomial distribution, the scale parameter is reported in the "Parameter Estimates" table. For some distributions, the parameterization of this parameter differs. See the section "Scale and Dispersion Parameters" on page 2215 for details about how the GLIMMIX procedure parameterizes the log-likelihood functions and information about how the reported quantities differ between the two procedures.

Many of the fit statistics and tests in the GENMOD procedure are based on the likelihood. In a GLMM it is not always possible to derive the log likelihood of the data. Even if the log likelihood is tractable, it might be computationally infeasible. In some cases, the objective function must be constructed based on a substitute model. In other cases, only the first two moments of the marginal distribution can be approximated. Consequently, obtaining likelihood-based tests and statistics is difficult for many generalized linear mixed models. The GLIMMIX procedure relies heavily on linearization and Taylor-series techniques to construct Wald-type test statistics and confidence intervals. Likelihood ratio tests and confidence intervals for covariance parameters are available in the GLIMMIX procedure through the COVTEST statement.

The NLMIXED procedure fits nonlinear mixed models where the conditional mean function is a general nonlinear function. The class of generalized linear mixed models is a special case of the nonlinear mixed models; hence some of the models you can fit with PROC NLMIXED can also be fit with the GLIMMIX procedure. The NLMIXED procedure relies by default on approximating the marginal log likelihood through adaptive Gaussian quadrature. In the GLIMMIX procedure, maximum likelihood estimation by adaptive Gaussian quadrature is available with the METHOD=QUAD option in the PROC GLIMMIX statement. The default estimation methods thus differ between the NLMIXED and GLIMMIX procedures, because adaptive quadrature is possible for only a subset of the models available with the GLIMMIX procedure. If you choose

METHOD=LAPLACE or METHOD=QUAD(QPOINTS=1) in the PROC GLIMMIX statement for a generalized linear mixed model, the GLIMMIX procedure performs maximum likelihood estimation based on a Laplace approximation of the marginal log likelihood. This is equivalent to the QPOINTS=1 option in the NLMIXED procedure.

The LOGISTIC and CATMOD procedures also fit generalized linear models; PROC LOGISTIC accommodates the independence case only. Binary, binomial, multinomial models for ordered data, and generalized logit models that can be fit with PROC LOGISTIC can also be fit with the GLIMMIX procedure. The diagnostic tools and capabilities specific to such data implemented in the LOGISTIC procedure go beyond the capabilities of the GLIMMIX procedure.

Getting Started: GLIMMIX Procedure

Logistic Regressions with Random Intercepts

Researchers investigated the performance of two medical procedures in a multicenter study. They randomly selected 15 centers for inclusion. One of the study goals was to compare the occurrence of side effects for the procedures. In each center n_A patients were randomly selected and assigned to procedure "A," and n_B patients were randomly assigned to procedure "B." The following DATA step creates the data set for the analysis:

```
data multicenter;
  input center group$ n sideeffect;
  datalines;
 1  A  32  14
 1  B  33  18
 2  A  30   4
 2  B  28   8
 3  A  23  14
 3  B  24   9
 4  A  22   7
 4  B  22  10
 5  A  20   6
 5  B  21  12
 6  A  19   1
 6  B  20   3
 7  A  17   2
 7  B  17   6
 8  A  16   7
 8  B  15   9
 9  A  13   1
 9  B  14   5
10  A  13   3
10  B  13   1
11  A  11   1
11  B  12   2
```

```
12  A  10   1
12  B   9   0
13  A   9   2
13  B   9   6
14  A   8   1
14  B   8   1
15  A   7   1
15  B   8   0
;
```

The variable group identifies the two procedures, n is the number of patients who received a given procedure in a particular center, and sideeffect is the number of patients who reported side effects.

If Y_{iA} and Y_{iB} denote the number of patients in center i who report side effects for procedures A and B, respectively, then—for a given center—these are independent binomial random variables. To model the probability of side effects for the two drugs, π_{iA} and π_{iB}, you need to account for the fixed group effect and the random selection of centers. One possibility is to assume a model that relates group and center effects linearly to the logit of the probabilities:

$$\log\left\{\frac{\pi_{iA}}{1-\pi_{iA}}\right\} = \beta_0 + \beta_A + \gamma_i$$
$$\log\left\{\frac{\pi_{iB}}{1-\pi_{iB}}\right\} = \beta_0 + \beta_B + \gamma_i$$

In this model, $\beta_A - \beta_B$ measures the difference in the logits of experiencing side effects, and the γ_i are independent random variables due to the random selection of centers. If you think of β_0 as the overall intercept in the model, then the γ_i are random intercept adjustments. Observations from the same center receive the same adjustment, and these vary randomly from center to center with variance $\text{Var}[\gamma_i] = \sigma_c^2$.

Because π_{iA} is the conditional mean of the sample proportion, $\text{E}[Y_{iA}/n_{iA}|\gamma_i] = \pi_{iA}$, you can model the sample proportions as binomial ratios in a generalized linear mixed model. The following statements request this analysis under the assumption of normally distributed center effects with equal variance and a logit link function:

```
proc glimmix data=multicenter;
   class center group;
   model sideeffect/n = group / solution;
   random intercept / subject=center;
run;
```

The PROC GLIMMIX statement invokes the procedure. The CLASS statement instructs the procedure to treat the variables center and group as classification variables. The MODEL statement specifies the response variable as a sample proportion by using the *events/trials* syntax. In terms of the previous formulas, sideeffect/n corresponds to Y_{iA}/n_{iA} for observations from group A and to Y_{iB}/n_{iB} for observations from group B. The SOLUTION option in the MODEL statement requests a listing of the solutions for the fixed-effects parameter estimates. Note that because of the *events/trials* syntax, the GLIMMIX procedure defaults to the binomial distribution, and that distribution's default link is the logit link. The RANDOM statement specifies that the linear predictor contains an intercept term that randomly varies at the level of the center effect. In other words, a random intercept is drawn separately and independently for each center in the study.

The results of this analysis are shown in Figure 38.1–Figure 38.9.

The "Model Information Table" in Figure 38.1 summarizes important information about the model you fit and about aspects of the estimation technique.

Figure 38.1 Model Information

The GLIMMIX Procedure

Model Information

Data Set	WORK.MULTICENTER
Response Variable (Events)	sideeffect
Response Variable (Trials)	n
Response Distribution	Binomial
Link Function	Logit
Variance Function	Default
Variance Matrix Blocked By	center
Estimation Technique	Residual PL
Degrees of Freedom Method	Containment

PROC GLIMMIX recognizes the variables sideeffect and n as the numerator and denominator in the *events/trials* syntax, respectively. The distribution—conditional on the random center effects—is binomial. The marginal variance matrix is block-diagonal, and observations from the same center form the blocks. The default estimation technique in generalized linear mixed models is residual pseudo-likelihood with a subject-specific expansion (METHOD=RSPL).

The "Class Level Information" table lists the levels of the variables specified in the CLASS statement and the ordering of the levels. The "Number of Observations" table displays the number of observations read and used in the analysis (Figure 38.2).

Figure 38.2 Class Level Information and Number of Observations

Class Level Information

Class	Levels	Values
center	15	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
group	2	A B

Number of Observations Read	30
Number of Observations Used	30
Number of Events	155
Number of Trials	503

There are two variables listed in the CLASS statement. The center variable has fifteen levels, and the group variable has two levels. Because the response is specified through the *events/trial* syntax, the "Number of Observations" table also contains the total number of events and trials used in the analysis.

The "Dimensions" table lists the size of relevant matrices (Figure 38.3).

Figure 38.3 Dimensions

Dimensions	
G-side Cov. Parameters	1
Columns in X	3
Columns in Z per Subject	1
Subjects (Blocks in V)	15
Max Obs per Subject	2

There are three columns in the **X** matrix, corresponding to an intercept and the two levels of the group variable. For each subject (center), the **Z** matrix contains only an intercept column.

The "Optimization Information" table provides information about the methods and size of the optimization problem (Figure 38.4).

Figure 38.4 Optimization Information

Optimization Information	
Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	1
Lower Boundaries	1
Upper Boundaries	0
Fixed Effects	Profiled
Starting From	Data

The default optimization technique for generalized linear mixed models with binomial data is the quasi-Newton method. Because a residual likelihood technique is used to compute the objective function, only the covariance parameters participate in the optimization. A lower boundary constraint is placed on the variance component for the random center effect. The solution for this variance cannot be less than zero.

The "Iteration History" table displays information about the progress of the optimization process. After the initial optimization, the GLIMMIX procedure performed 15 updates before the convergence criterion was met (Figure 38.5). At convergence, the largest absolute value of the gradient was near zero. This indicates that the process stopped at an extremum of the objective function.

Figure 38.5 Iteration History and Convergence Status

Iteration History

Iteration	Restarts	Subiterations	Objective Function	Change	Max Gradient
0	0	5	79.688580269	0.11807224	7.851E-7
1	0	3	81.294622554	0.02558021	8.209E-7
2	0	2	81.438701534	0.00166079	4.061E-8
3	0	1	81.444083567	0.00006263	2.292E-8
4	0	1	81.444265216	0.00000421	0.000025
5	0	1	81.444277364	0.00000383	0.000023
6	0	1	81.444266322	0.00000348	0.000021
7	0	1	81.44427636	0.00000316	0.000019
8	0	1	81.444267235	0.00000287	0.000017
9	0	1	81.44427553	0.00000261	0.000016
10	0	1	81.44426799	0.00000237	0.000014
11	0	1	81.444274844	0.00000216	0.000013
12	0	1	81.444268614	0.00000196	0.000012
13	0	1	81.444274277	0.00000178	0.000011
14	0	1	81.444269129	0.00000162	9.772E-6
15	0	0	81.444273808	0.00000000	9.102E-6

Convergence criterion (PCONV=1.11022E-8) satisfied.

The "Fit Statistics" table lists information about the fitted model (Figure 38.6).

Figure 38.6 Fit Statistics

Fit Statistics

-2 Res Log Pseudo-Likelihood	81.44
Generalized Chi-Square	30.69
Gener. Chi-Square / DF	1.10

Twice the negative of the residual log likelihood in the final pseudo-model equaled 81.44. The ratio of the generalized chi-square statistic and its degrees of freedom is close to 1. This is a measure of the residual variability in the marginal distribution of the data.

The "Covariance Parameter Estimates" table displays estimates and asymptotic estimated standard errors for all covariance parameters (Figure 38.7).

Figure 38.7 Covariance Parameter Estimates

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error
Intercept	center	0.6176	0.3181

The variance of the random center intercepts on the logit scale is estimated as $\widehat{\sigma}_c^2 = 0.6176$.

The "Parameter Estimates" table displays the solutions for the fixed effects in the model (Figure 38.8).

Figure 38.8 Parameter Estimates

Solutions for Fixed Effects

Effect	group	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept		−0.8071	0.2514	14	−3.21	0.0063
group	A	−0.4896	0.2034	14	−2.41	0.0305
group	B	0	.	.	.	.

Because of the fixed-effects parameterization used in the GLIMMIX procedure, the "Intercept" effect is an estimate of $\beta_0 + \beta_B$, and the "A" group effect is an estimate of $\beta_A - \beta_B$, the log odds ratio. The associated estimated probabilities of side effects in the two groups are

$$\widehat{\pi}_A = \frac{1}{1 + \exp\{0.8071 + 0.4896\}} = 0.2147$$
$$\widehat{\pi}_B = \frac{1}{1 + \exp\{0.8071\}} = 0.3085$$

There is a significant difference between the two groups (p=0.0305).

The "Type III Tests of Fixed Effect" table displays significance tests for the fixed effects in the model (Figure 38.9).

Figure 38.9 Type III Tests of Fixed Effects

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
group	1	14	5.79	0.0305

Because the group effect has only two levels, the p-value for the effect is the same as in the "Parameter Estimates" table, and the "F Value" is the square of the "t Value" shown there.

You can produce the estimates of the average logits in the two groups and their predictions on the scale of the data with the LSMEANS statement in PROC GLIMMIX:

```
ods select lsmeans;
proc glimmix data=multicenter;
   class center group;
   model sideeffect/n = group / solution;
   random intercept / subject=center;
   lsmeans group / cl ilink;
run;
```

The LSMEANS statement requests the least squares means of the group effect on the logit scale. The CL option requests their confidence limits. The ILINK option adds estimates, standard errors, and confidence limits on the mean (probability) scale (Figure 38.10).

Figure 38.10 Least Squares Means

The GLIMMIX Procedure

group Least Squares Means

group	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
A	-1.2966	0.2601	14	-4.99	0.0002	0.05	-1.8544	-0.7388
B	-0.8071	0.2514	14	-3.21	0.0063	0.05	-1.3462	-0.2679

group Least Squares Means

group	Mean	Standard Error Mean	Lower Mean	Upper Mean
A	0.2147	0.04385	0.1354	0.3233
B	0.3085	0.05363	0.2065	0.4334

The "Estimate" column displays the least squares mean estimate on the logit scale, and the "Mean" column represents its mapping onto the probability scale. The "Lower" and "Upper" columns are 95% confidence limits for the logits in the two groups. The "Lower Mean" and "Upper Mean" columns are the corresponding confidence limits for the probabilities of side effects. These limits are obtained by inversely linking the confidence bounds on the linear scale, and thus are not symmetric about the estimate of the probabilities.

Syntax: GLIMMIX Procedure

You can specify the following statements in the GLIMMIX procedure:

PROC GLIMMIX < *options* > ;
BY *variables* ;
CLASS *variables* ;
CONTRAST *'label' contrast-specification* < , *contrast-specification* > < , . . . > < / *options* > ;
COVTEST < *'label'* > < *test-specification* > < / *options* > ;
EFFECT *effect-specification* ;
ESTIMATE *'label' contrast-specification* < **(***divisor***=***n***)** >
< , *'label' contrast-specification* < **(***divisor***=***n***)** > > < , . . . > < / *options* > ;
FREQ *variable* ;
ID *variables* ;
LSMEANS *fixed-effects* < / *options* > ;
LSMESTIMATE *fixed-effect* < *'label'* > *values* < *divisor*=*n* >
< , < *'label'* > *values* < *divisor*=*n* > > < , . . . > < / *options* > ;
MODEL *response*< **(***response-options***)** > **=** < *fixed-effects* > < / *model-options* > ;
MODEL *events/trials* **=** < *fixed-effects* > < / *model-options* > ;
NLOPTIONS < *options* > ;
OUTPUT < **OUT=***SAS-data-set* >
< *keyword*< **(***keyword-options***)** > < **=***name* > >. . .
< *keyword*< **(***keyword-options***)** > < **=***name* > > < / *options* > ;
PARMS **(***value-list***)** . . . < / *options* > ;
RANDOM *random-effects* < / *options* > ;
WEIGHT *variable* ;
Programming statements ;

The CLASS, CONTRAST, COVTEST, EFFECT, ESTIMATE, LSMEANS, LSMESTIMATE, and RANDOM statements and the programming statements can appear multiple times. The PROC GLIMMIX and MODEL statements are required, and the MODEL statement must appear after the CLASS statement if a CLASS statement is included. The EFFECT statements must appear before the MODEL statement.

PROC GLIMMIX Statement

PROC GLIMMIX < *options* > **;**

The PROC GLIMMIX statement invokes the procedure. Table 38.1 summarizes some important options in the PROC GLIMMIX statement by function. These and other options in the PROC GLIMMIX statement are then described fully in alphabetical order.

Table 38.1 PROC GLIMMIX Statement Options

Option	Description
Basic Options	
DATA=	specifies the input data set
METHOD=	determines estimation method
NOFIT	does not fit the model
NOPROFILE	includes scale parameter in optimization
NOREML	determines computation of scale parameters in GLM models
ORDER=	determines the sort order of CLASS variables
OUTDESIGN	writes **X** and/or **Z** matrices to a SAS data set
Displayed Output	
ASYCORR	displays the asymptotic correlation matrix of the covariance parameter estimates
ASYCOV	displays the asymptotic covariance matrix of the covariance parameter estimates
GRADIENT	displays the gradient of the objective function with respect to the parameter estimates
HESSIAN	displays the Hessian matrix
ITDETAILS	adds estimates and gradients to the "Iteration History"
NOBSDETAIL	shows data exclusions
NOCLPRINT	suppresses "Class Level Information" completely or in part
ODDSRATIO	requests odds ratios
PLOTS	produces ODS statistical graphics
SUBGRADIENT	writes subject-specific gradients to a SAS data set
Optimization Options	
MAXOPT=	specifies the number of optimizations
Computational Options	
EMPIRICAL	computes empirical ("sandwich") estimators
INFOCRIT	affects the computation of information criteria
INITGLM	uses fixed-effects starting values via generalized linear model
INTITER=	sets the number of initial GLM steps
NOBOUND	unbounds the covariance parameter estimates
SCORING=	applies Fisher scoring where applicable

Table 38.1 *continued*

Option	Description
Singularity Tolerances	
ABSPCONV=	determines the absolute parameter estimate convergence criterion for PL
FDIGITS=	specifies significant digits in computing objective function
PCONV=	specifies the relative parameter estimate convergence criterion for PL
SINGCHOL=	tunes singularity for Cholesky decompositions
SINGRES=	tunes singularity for the residual variance
SINGULAR=	tunes general singularity criterion
Debugging Output	
LIST	lists model program and variables

You can specify the following options in the PROC GLIMMIX statement.

ABSPCONV=*r*

specifies an absolute parameter estimate convergence criterion for doubly iterative estimation methods. For such methods, the GLIMMIX procedure by default examines the *relative* change in parameter estimates between optimizations (see PCONV=). The purpose of the ABSPCONV= criterion is to stop the process when the *absolute* change in parameter estimates is less than the tolerance criterion r. The criterion is based on fixed effects and covariance parameters.

Note that this convergence criterion does not affect the convergence criteria applied within any individual optimization. In order to change the convergence behavior within an optimization, you can change the ABSCONV=, ABSFCONV=, ABSGCONV=, ABSXCONV=, FCONV=, or GCONV= option in the NLOPTIONS statement.

ASYCORR

produces the asymptotic correlation matrix of the covariance parameter estimates. It is computed from the corresponding asymptotic covariance matrix (see the description of the ASYCOV option, which follows).

ASYCOV

requests that the asymptotic covariance matrix of the covariance parameter estimates be displayed. By default, this matrix is the observed inverse Fisher information matrix, which equals $m\mathbf{H}^{-1}$, where $\mathbf{H}$ is the Hessian (second derivative) matrix of the objective function. The factor m equals 1 in a GLM and equals 2 in a GLMM.

When you use the SCORING= option and PROC GLIMMIX converges without stopping the scoring algorithm, the procedure uses the expected Hessian matrix to compute the covariance matrix instead of the observed Hessian. Regardless of whether a scoring algorithm is used or the number of scoring iterations has been exceeded, you can request that the asymptotic covariance matrix be based on the expected Hessian with the EXPHESSIAN option in the PROC GLIMMIX statement. If a residual scale parameter is profiled from the likelihood equation, the asymptotic covariance matrix is adjusted for the presence of this parameter;

details of this adjustment process are found in Wolfinger, Tobias, and Sall (1994) and in the section "Estimated Precision of Estimates" on page 2221.

CHOLESKY
CHOL

requests that the mixed model equations be constructed and solved by using the Cholesky root of the $\mathbf{G}$ matrix. This option applies only to estimation methods that involve mixed model equations. The Cholesky root algorithm has greater numerical stability but also requires more computing resources. When the estimated $\mathbf{G}$ matrix is not positive definite during a particular function evaluation, PROC GLIMMIX switches to the Cholesky algorithm for that evaluation and returns to the regular algorithm if $\widehat{\mathbf{G}}$ becomes positive definite again. When the CHOLESKY option is in effect, the procedure applies the algorithm all the time.

DATA=*SAS-data-set*

names the SAS data set to be used by PROC GLIMMIX. The default is the most recently created data set.

EMPIRICAL<=CLASSICAL | HC0>
EMPIRICAL<=DF | HC1>
EMPIRICAL<=MBN<(*mbn-options***)>>**
EMPIRICAL<=ROOT | HC2>
EMPIRICAL<=FIRORES | HC3>
EMPIRICAL<=FIROEEQ<(*r***)>>**

requests that the covariance matrix of the parameter estimates be computed as one of the asymptotically consistent estimators, known as *sandwich* or *empirical* estimators. The name stems from the layering of the estimator. An empirically based estimate of the inverse variance of the parameter estimates (the "meat") is wrapped by the model-based variance estimate (the "bread").

Empirical estimators are useful for obtaining inferences that are not sensitive to the choice of the covariance model. In nonmixed models, they can help, for example, to allay the effects of variance heterogeneity on the tests of fixed effects. Empirical estimators can coarsely be grouped into likelihood-based and residual-based estimators. The distinction arises from the components used to construct the "meat" and "bread" of the estimator. If you specify the EMPIRICAL option without further qualifiers, the GLIMMIX procedure computes the classical sandwich estimator in the appropriate category.

Likelihood-Based Estimator

Let $\mathbf{H}(\boldsymbol{\alpha})$ denote the second derivative matrix of the log likelihood for some parameter vector $\boldsymbol{\alpha}$, and let $\mathbf{g}_i(\boldsymbol{\alpha})$ denote the gradient of the log likelihood with respect to $\boldsymbol{\alpha}$ for the ith of m independent sampling units. The gradient for the entire data is $\sum_{i=1}^{m}\mathbf{g}_i(\boldsymbol{\alpha})$. A sandwich estimator for the covariance matrix of $\widehat{\boldsymbol{\alpha}}$ can then be constructed as (White 1982)

$$\mathbf{H}(\widehat{\boldsymbol{\alpha}})^{-1}\left(\sum_{i=1}^{m}\mathbf{g}_i(\widehat{\boldsymbol{\alpha}})\mathbf{g}_i(\widehat{\boldsymbol{\alpha}})'\right)\mathbf{H}(\widehat{\boldsymbol{\alpha}})^{-1}$$

If you fit a mixed model by maximum likelihood with Laplace or quadrature approximation (METHOD=LAPLACE, METHOD=QUAD), the GLIMMIX procedure constructs this likelihood-based estimator when you choose EMPIRICAL=CLASSICAL. If you choose EMPIRICAL=MBN, the likelihood-based sandwich estimator is further adjusted (see the section "Design-Adjusted MBN Estimator" on page 2244 for details). Because Laplace and quadrature estimation in GLIMMIX includes the fixed-effects parameters and the covariance parameters in the optimization, this empirical estimator adjusts the covariance matrix of both types of parameters. The following empirical estimators are not available with METHOD=LAPLACE or with METHOD=QUAD: EMPIRICAL=DF, EMPIRICAL=ROOT, EMPIRICAL=FIRORES, and EMPIRICAL=FIROEEQ.

Residual-Based Estimators

For a general model, let $\mathbf{Y}$ denote the response with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\Sigma}$, and let $\mathbf{D}$ be the matrix of first derivatives of $\boldsymbol{\mu}$ with respect to the fixed effects $\boldsymbol{\beta}$. The classical sandwich estimator (Huber 1967; White 1980) is

$$\widehat{\boldsymbol{\Omega}}\left(\sum_{i=1}^{m}\widehat{\mathbf{D}}_i'\widehat{\boldsymbol{\Sigma}}_i^{-1}\mathbf{e}_i\mathbf{e}_i'\widehat{\boldsymbol{\Sigma}}_i^{-1}\widehat{\mathbf{D}}_i\right)\widehat{\boldsymbol{\Omega}}$$

where $\boldsymbol{\Omega} = (\mathbf{D}'\boldsymbol{\Sigma}^{-1}\mathbf{D})^{-}$, $\mathbf{e}_i = \mathbf{y}_i - \widehat{\boldsymbol{\mu}}_i$, and m denotes the number of independent sampling units.

Since the expected value of $\mathbf{e}_i\mathbf{e}_i'$ does not equal $\boldsymbol{\Sigma}_i$, the classical sandwich estimator is biased, particularly if m is small. The estimator tends to underestimate the variance of $\widehat{\boldsymbol{\beta}}$. The EMPIRICAL=DF, ROOT, FIRORES, FIROEEQ, and MBN estimators are bias-corrected sandwich estimators. The DF estimator applies a simple sample size adjustment. The ROOT, FIRORES, and FIROEEQ estimators are based on Taylor series approximations applied to residuals and estimating equations. For uncorrelated data, the EMPIRICAL=FIRORES estimator can be motivated as a jackknife estimator.

In the case of a linear regression model, the various estimators reduce to the *heteroscedasticity-consistent covariance matrix* estimators (HCMM) of White (1980) and MacKinnon and White (1985). The classical estimator, HC0, was found to perform poorly in small samples. Based on simulations in regression models, MacKinnon and White (1985) and Long and Ervin (2000) strongly recommend the HC3 estimator. The sandwich estimators computed by the GLIMMIX procedure can be viewed as an extension of the HC0—HC3 estimators of MacKinnon and White (1985) to accommodate nonnormal data and correlated observations.

The MBN estimator, introduced as a residual-based estimator by Morel (1989) and Morel, Bokossa, and Neerchal (2003), applies an additive adjustment to the residual crossproduct. It is controlled by three suboptions. The valid *mbn-options* are as follows: a sample size adjustment is applied when the DF suboption is in effect. The NODF suboption suppresses this component of the adjustment. The lower bound of the design effect parameter $0 \leq r \leq 1$ can be specified with the R= option. The magnitude of Morel's δ parameter is partly determined with the D= option ($d \geq 1$).

For details about the general expression for the residual-based estimators and their relationship, see the section "Empirical Covariance ("Sandwich") Estimators" on page 2243. The MBN estimator and its parameters are explained for residual- and likelihood-based estimators in the section "Design-Adjusted MBN Estimator" on page 2244.

The EMPIRICAL=DF estimator applies a simple, multiplicative correction factor to the classical estimator (Hinkley 1977). This correction factor is

$$c = \begin{cases} m/(m-k) & m > k \\ 1 & \text{otherwise} \end{cases}$$

where k is the rank of $\mathbf{X}$, and m equals the sum of all frequencies when PROC GLIMMIX is in GLM mode and equals the number of subjects in GLMM mode. For example, the following statements fit an overdispersed GLM:

```
proc glimmix empirical;
   model y = x;
   random _residual_;
run;
```

PROC GLIMMIX is in GLM mode, and the individual observations are the independent sampling units from which the sandwich estimator is constructed. If you use a SUBJECT= effect in the RANDOM statement, however, the procedure fits the model in GLMM mode and the subjects represent the sampling units in the construction of the sandwich estimator. In other words, the following statements fit a GEE-type model with independence working covariance structure and subjects (clusters) defined by the levels of ID:

```
proc glimmix empirical;
   class id;
   model y = x;
   random _residual_ / subject=id type=vc;
run;
```

See the section "GLM Mode or GLMM Mode" on page 2233 for information about how the GLIMMIX procedure determines the estimation mode.

The EMPIRICAL=ROOT estimator is based on the residual approximation in Kauermann and Carroll (2001), and the EMPIRICAL=FIRORES estimator is based on the approximation in Mancl and DeRouen (2001). The Kauermann and Carroll estimator requires the inverse square root of a nonsymmetric matrix. This square root matrix is obtained from the singular value decomposition in PROC GLIMMIX, and thus this sandwich estimator is computationally more demanding than others. In the linear regression case, the Mancl-DeRouen estimator can be motivated as a jackknife estimator, based on the "leave-one-out" estimates of $\widehat{\boldsymbol{\beta}}$; see MacKinnon and White (1985) for details.

The EMPIRICAL=FIROEEQ estimator is based on approximating an unbiased estimating equation (Fay and Graubard 2001). It is computationally less demanding than the estimator of Kauermann and Carroll (2001) and, in certain balanced cases, gives identical results. The optional number $0 \leq r < 1$ is chosen to provide an upper bound on the correction factor. The default value for r is 0.75.

When you specify the EMPIRICAL option with a residual-based estimator, PROC GLIMMIX adjusts all standard errors and test statistics involving the fixed-effects parameters.

Sampling Units

Computation of an empirical variance estimator requires that the data can be processed by independent sampling units. This is always the case in GLMs. In this case, m, the number of independent units, equals the sum of the frequencies used in the analysis (see "Number of Observations" table). In GLMMs, empirical estimators can be computed only if the data comprise more than one subject as per the "Dimensions" table. See the section "Processing by Subjects" on page 2247 for information about how the GLIMMIX procedure determines whether the data can be processed by subjects. If a GLMM comprises only a single subject for a particular BY group, the model-based variance estimator is used instead of the empirical estimator, and a message is written to the log.

EXPHESSIAN

requests that the expected Hessian matrix be used in computing the covariance matrix of the nonprofiled parameters. By default, the GLIMMIX procedure uses the observed Hessian matrix in computing the asymptotic covariance matrix of covariance parameters in mixed models and the covariance matrix of fixed effects in models without random effects. The EXPHESSIAN option is ignored if the (conditional) distribution is not a member of the exponential family or is unknown. It is also ignored in models for nominal data.

FDIGITS=*r*

specifies the number of accurate digits in evaluations of the objective function. Fractional values are allowed. The default value is $r = -\log_{10}\epsilon$, where ϵ is the machine precision. The value of r is used to compute the interval size for the computation of finite-difference approximations of the derivatives of the objective function. It is also used in computing the default value of the FCONV= option in the NLOPTIONS statement.

GRADIENT

displays the gradient of the objective function with respect to the parameter estimates in the "Covariance Parameter Estimates" table and/or the "Parameter Estimates" table.

HESSIAN

HESS

H

displays the Hessian matrix of the optimization.

INFOCRIT=NONE | PQ | Q

IC=NONE | PQ | Q

determines the computation of information criteria in the "Fit Statistics" table. The GLIMMIX procedure computes various information criteria that typically apply a penalty to the (possibly restricted) log likelihood, log pseudo-likelihood, or log quasi-likelihood that depends on the number of parameters and/or the sample size. If IC=NONE, these criteria are suppressed in the "Fit Statistics" table. This is the default for models based on pseudo-likelihoods.

The AIC, AICC, BIC, CAIC, and HQIC fit statistics are various information criteria. AIC and AICC represent Akaike's information criteria (Akaike 1974) and a small sample bias corrected version thereof (for AICC, see Hurvich and Tsai 1989; Burnham and Anderson 1998). BIC represents Schwarz's Bayesian criterion (Schwarz 1978). Table 38.2 gives formulas for the criteria.

Table 38.2 Information Criteria

Criteria	Formula	Reference
AIC	$-2\ell + 2d$	Akaike (1974)
AICC	$-2\ell + 2dn^*/(n^* - d - 1)$	Hurvich and Tsai (1989) Burnham and Anderson (1998)
HQIC	$-2\ell + 2d \log\log n$	Hannan and Quinn (1979)
BIC	$-2\ell + d \log n$	Schwarz (1978)
CAIC	$-2\ell + d(\log n + 1)$	Bozdogan (1987)

Here, ℓ denotes the maximum value of the (possibly restricted) log likelihood, log pseudo-likelihood, or log quasi-likelihood, d is the dimension of the model, and n, n^* reflect the size of the data.

The IC=PQ option requests that the penalties include the number of fixed-effects parameters, when estimation in models with random effects is based on a residual (restricted) likelihood. For METHOD=MSPL, METHOD=MMPL, METHOD=LAPLACE, and METHOD=QUAD, IC=Q and IC=PQ produce the same results. IC=Q is the default for linear mixed models with normal errors, and the resulting information criteria are identical to the IC option in the MIXED procedure.

The quantities d, n, and n^* depend on the model and IC= option in the following way:

GLM: IC=Q and IC=PQ options have no effect on the computation.

- d equals the number of parameters in the optimization whose solutions do not fall on the boundary or are otherwise constrained. The scale parameter is included, if it is part of the optimization. If you use the PARMS statement to place a hold on a scale parameter, that parameter does not count toward d.
- n equals the sum of the frequencies (f) for maximum likelihood and quasi-likelihood estimation and $f - \text{rank}(\mathbf{X})$ for restricted maximum likelihood estimation.
- n^* equals n, unless $n < d + 2$, in which case $n^* = d + 2$.

GLMM, IC=Q:

- d equals the number of effective covariance parameters—that is, covariance parameters whose solution does not fall on the boundary. For estimation of an unrestricted objective function (METHOD=MMPL, METHOD=MSPL, METHOD=LAPLACE, METHOD=QUAD), this value is incremented by rank($\mathbf{X}$).

- n equals the effective number of subjects as displayed in the "Dimensions" table, unless this value equals 1, in which case n equals the number of levels of the first G-side RANDOM effect specified. If the number of effective subjects equals 1 and there are no G-side random effects, n is determined as

$$n = \begin{cases} f - \text{rank}(\mathbf{X}) & \text{METHOD} = \text{RMPL}, \text{METHOD} = \text{RSPL} \\ f & \text{otherwise} \end{cases}$$

  where f is the sum of frequencies used.
- n^* equals f or $f - \text{rank}(\mathbf{X})$ (for METHOD=RMPL and METHOD=RSPL), unless this value is less than $d + 2$, in which case $n^* = d + 2$.

GLMM, IC=PQ: For METHOD=MSPL, METHOD=MMPL, METHOD=LAPLACE, and METHOD=QUAD, the results are the same as for IC=Q. For METHOD=RSPL and METHOD=RMPL, d equals the number of effective covariance parameters plus rank($\mathbf{X}$), and $n = n^*$ equals $f - \text{rank}(\mathbf{X})$. The formulas for the information criteria thus agree with Verbeke and Molenberghs (2000, Table 6.7, p. 74) and Vonesh and Chinchilli (1997, p. 263).

INITGLM

requests that the estimates from a generalized linear model fit (a model without random effects) be used as the starting values for the generalized linear mixed model. This option is the default for METHOD=LAPLACE and METHOD=QUAD.

INITITER=*number*

specifies the maximum number of iterations used when a generalized linear model is fit initially to derive starting values for the fixed effects; see the INITGLM option. By default, the initial fit involves at most four iteratively reweighted least squares updates. You can change the upper limit of initial iterations with *number*. If the model does not contain random effects, this option has no effect.

ITDETAILS

adds parameter estimates and gradients to the "Iteration History" table.

LIST

requests that the model program and variable lists be displayed. This is a debugging feature and is not normally needed. When you use programming statements to define your statistical model, this option enables you to examine the complete set of statements submitted for processing. See the section "Programming Statements" for more details about how to use SAS statements with the GLIMMIX procedure.

MAXLMMUPDATE=*number*

MAXOPT=*number*

specifies the maximum number of optimizations for doubly iterative estimation methods based on linearizations. After each optimization, a new pseudo-model is constructed through a Taylor series expansion. This step is known as the linear mixed model update. The

MAXLMMUPDATE option limits the number of updates and thereby limits the number of optimizations. If this option is not specified, *number* is set equal to the value specified in the MAXITER= option in the NLOPTIONS statement. If no MAXITER= value is given, *number* defaults to 20.

METHOD=RSPL

METHOD=MSPL

METHOD=RMPL

METHOD=MMPL

METHOD=LAPLACE

METHOD=QUAD<(*quad-options***)>**

specifies the estimation method in a generalized linear mixed model (GLMM). The default is METHOD=RSPL.

Pseudo-Likelihood

Estimation methods ending in "PL" are pseudo-likelihood techniques. The first letter of the METHOD= identifier determines whether estimation is based on a residual likelihood ("R") or a maximum likelihood ("M"). The second letter identifies the expansion locus for the underlying approximation. Pseudo-likelihood methods for generalized linear mixed models can be cast in terms of Taylor series expansions (linearizations) of the GLMM. The expansion locus of the expansion is either the vector of random effects solutions ("S") or the mean of the random effects ("M"). The expansions are also referred to as the "S"ubject-specific and "M"arginal expansions. The abbreviation "PL" identifies the method as a pseudo-likelihood technique.

Residual methods account for the fixed effects in the construction of the objective function, which reduces the bias in covariance parameter estimates. Estimation methods involving Taylor series create pseudo-data for each optimization. Those data are transformed to have zero mean in a residual method. While the covariance parameter estimates in a residual method are the maximum likelihood estimates for the transformed problem, the fixed-effects estimates are (estimated) generalized least squares estimates. In a likelihood method that is not residual based, both the covariance parameters and the fixed-effects estimates are maximum likelihood estimates, but the former are known to have greater bias. In some problems, residual likelihood estimates of covariance parameters are unbiased.

For more information about linearization methods for generalized linear mixed models, see the section "Pseudo-likelihood Estimation Based on Linearization" on page 2220.

Maximum Likelihood with Laplace Approximation

If you choose METHOD=LAPLACE with a generalized linear mixed model, PROC GLIMMIX approximates the marginal likelihood by using Laplace's method. Twice the negative of the resulting log-likelihood approximation is the objective function that the procedure minimizes to determine parameter estimates. Laplace estimates typically exhibit better asymptotic

behavior and less small-sample bias than pseudo-likelihood estimators. On the other hand, the class of models for which a Laplace approximation of the marginal log likelihood is available is much smaller compared to the class of models to which PL estimation can be applied.

To determine whether Laplace estimation can be applied in your model, consider the marginal distribution of the data in a mixed model

$$\begin{aligned} p(\mathbf{y}) &= \int p(\mathbf{y}|\boldsymbol{\gamma})\, p(\boldsymbol{\gamma})\, d\boldsymbol{\gamma} \\ &= \int \exp\{\log\{p(\mathbf{y}|\boldsymbol{\gamma})\} + \log\{p(\boldsymbol{\gamma})\}\}\ d\boldsymbol{\gamma} \\ &= \int \exp\{nf(\mathbf{y},\boldsymbol{\gamma})\}\ d\boldsymbol{\gamma} \end{aligned}$$

The function $f(\mathbf{y},\boldsymbol{\gamma})$ plays an important role in the Laplace approximation: it is a function of the joint distribution of the data and the random effects (see the section "Maximum Likelihood Estimation Based on Laplace Approximation" on page 2224). In order to construct a Laplace approximation, PROC GLIMMIX requires a conditional log-likelihood $\log\{p(\mathbf{y}|\boldsymbol{\gamma})\}$ as well as the distribution of the G-side random effects. The random effects are always assumed to be normal with zero mean and covariance structure determined by the RANDOM statement. The conditional distribution is determined by the DIST= option of the MODEL statement or the default associated with a particular response type. Because a valid conditional distribution is required, R-side random effects are not permitted for METHOD=LAPLACE in the GLIMMIX procedure. In other words, the GLIMMIX procedure requires for METHOD=LAPLACE conditional independence without R-side overdispersion or covariance structure.

Because the marginal likelihood of the data is approximated numerically, certain features of the marginal distribution are not available—for example, you cannot display a marginal variance-covariance matrix. Also, the procedure includes both the fixed-effects parameters and the covariance parameters in the optimization for Laplace estimation. Consequently, this setting imposes some restrictions with respect to available options for Laplace estimation. Table 38.3 lists the options that are assumed for METHOD=LAPLACE, and Table 38.4 lists the options that are not compatible with this estimation method.

The section "Maximum Likelihood Estimation Based on Laplace Approximation" contains details about Laplace estimation in PROC GLIMMIX.

Maximum Likelihood with Adaptive Quadrature

If you choose METHOD=QUAD in a generalized linear mixed model, the GLIMMIX procedure approximates the marginal log likelihood with an adaptive Gauss-Hermite quadrature. Compared to METHOD=LAPLACE, the models for which parameters can be estimated by quadrature are further restricted. In addition to the conditional independence assumption and the absence of R-side covariance parameters, it is required that models suitable for METHOD=QUAD can be processed by subjects. (See the section "Processing by Subjects" on page 2247 about how the GLIMMIX procedure determines whether the data can be processed by subjects.) This in turn requires that all RANDOM statements have SUBJECT=

effects and in the case of multiple SUBJECT= effects that these form a containment hierarchy.

In a containment hierarchy each effect is contained by another effect, and the effect contained by all is considered "the" effect for subject processing. For example, the SUBJECT= effects in the following statements form a containment hierarchy:

```
proc glimmix;
   class A B block;
   model y = A B A*B;
   random intercept / subject=block;
   random intercept / subject=A*block;
run;
```

The block effect is contained in the A*block interaction and the data are processed by block. The SUBJECT= effects in the following statements do not form a containment hierarchy:

```
proc glimmix;
   class A B block;
   model y = A B A*B;
   random intercept / subject=block;
   random block     / subject=A;
run;
```

The section "Maximum Likelihood Estimation Based on Adaptive Quadrature" on page 2227 contains important details about the computations involved with quadrature approximations. The section "Aspects Common to Adaptive Quadrature and Laplace Approximation" on page 2230 contains information about issues that apply to Laplace and adaptive quadrature, such as the computation of the prediction variance matrix and the determination of starting values.

You can specify the following *quad-options* for METHOD=QUAD in parentheses:

EBDETAILS
reports details about the empirical Bayes suboptimization process should this suboptimization fail.

EBSSFRAC=*r*
specifies the step-shortening fraction to be used while computing empirical Bayes estimates of the random effects. The default value is $r = 0.8$, and it is required that $r > 0$.

EBSSTOL=*r*
specifies the objective function tolerance for determining the cessation of step shortening while computing empirical Bayes estimates of the random effects, $r \geq 0$. The default value is $r = 1\text{E} - 8$.

EBSTEPS=*n*
specifies the maximum number of Newton steps for computing empirical Bayes estimates of random effects, $n \geq 0$. The default value is $n = 50$.

EBSUBSTEPS=*n*
specifies the maximum number of step shortenings for computing empirical Bayes estimates of random effects. The default value is $n = 20$, and it is required that $n \geq 0$.

EBTOL=*r*

specifies the convergence tolerance for empirical Bayes estimation, $r \geq 0$. The default value is $r = \epsilon \times 1E4$, where ϵ is the machine precision. This default value equals approximately 1E−12 on most machines.

INITPL=*number*

requests that adaptive quadrature commence after performing up to *number* pseudo-likelihood updates. The initial pseudo-likelihood (PL) steps (METHOD=MSPL) can be useful to provide good starting values for the quadrature algorithm. If you choose *number* large enough so that the initial PL estimation converges, the process is equivalent to starting a quadrature from the PL estimates of the fixed-effects and covariance parameters. Because this also makes available the PL random-effects solutions, the adaptive step of the quadrature that determines the number of quadrature points can take this information into account.

Note that you can combine the INITPL option with the NOINITGLM option in the PROC GLIMMIX statement to define a precise path for starting value construction to the GLIMMIX procedure. For example, the following statement generates starting values in these steps:

```
proc glimmix method=quad(initpl=5);
```

1. A GLM without random effects is fit initially to obtain as starting values for the fixed effects. The INITITER= option in the PROC GLIMMIX statement controls the number of iterations in this step.
2. Starting values for the covariance parameters are then obtained by MIVQUE0 estimation (Goodnight 1978b), using the fixed-effects parameter estimates from step 1.
3. With these values up to five pseudo-likelihood updates are computed.
4. The PL estimates for fixed-effects, covariance parameters, and the solutions for the random effects are then used to determine the number of quadrature points and used as the starting values for the quadrature.

The first step (GLM fixed-effects estimates) is omitted, if you modify the previous statement as follows:

```
proc glimmix method=quad(initpl=5) noinitglm;
```

The NOINITGLM option is the default of the pseudo-likelihood methods you select with the METHOD= option.

QCHECK

performs an adaptive recalculation of the objective function (−2 log likelihood) at the solution. The increment of the quadrature points, starting from the number of points used in the optimization, follows the same rules as the determination of the quadrature point sequence at the starting values (see the QFAC= and QMAX= suboptions). For example, the following statement estimates the parameters based on a quadrature with seven nodes in each dimension:

```
proc glimmix method=quad(qpoints=7 qcheck);
```

Because the default search sequence is $1, 3, 5, 7, 9, 11, 21, 31$, the QCHECK option computes the -2 log likelihood at the converged solution for $9, 11, 21$, and 31 quadrature points and reports relative differences to the converged value and among successive values. The ODS table produced by this option is named "QuadCheck."

CAUTION: This option is useful to diagnose the sensitivity of the likelihood approximation at the solution. It does **not** diagnose the stability of the solution under changes in the number of quadrature points. For example, if increasing the number of points from 7 to 9 does not alter the objective function, this does not imply that a quadrature with 9 points would arrive at the same parameter estimates as a quadrature with 7 points.

QFAC=*r*

determines the step size for the quadrature point sequence. If the GLIMMIX procedure determines the quadrature nodes adaptively, the log likelihoods are computed for nodes in a predetermined sequence. If N_{min} and N_{max} denote the values from the QMIN= and QMAX= suboptions, respectively, the sequence for values less than 11 is constructed in increments of 2 starting at N_{min}. Values greater than 11 are incremented in steps of r. The default value is $r = 10$. The default sequence, without specifying the QMIN=, QMAX=, or QFAC= option, is thus $1, 3, 5, 7, 9, 11, 21, 31$. By contrast, the following statement evaluates the sequence $8, 10, 30, 50$:

```
proc glimmix method=quad(qmin=8,qmax=51,qfac=20);
```

QMAX=*n*

specifies an upper bound for the number of quadrature points. The default is $n = 31$.

QMIN=*n*

specifies a lower bound for the number of quadrature points. The default is $n = 1$ and the value must be less than the QMAX= value.

QPOINTS=*n*

determines the number of quadrature points in each dimension of the integral. Note that if there are r random effects for each subject, the GLIMMIX procedure evaluates n^r conditional log likelihoods for each observation to compute one value of the objective function. Increasing the number of quadrature nodes can substantially increase the computational burden. If you choose QPOINTS=1, the quadrature approximation reduces to the Laplace approximation. If you do not specify the number of quadrature points, it is determined adaptively by increasing the number of nodes at the starting values. See the section "Aspects Common to Adaptive Quadrature and Laplace Approximation" on page 2230 for details.

QTOL=*r*

specifies a relative tolerance criterion for the successive evaluation of log likelihoods for different numbers of quadrature points. When the GLIMMIX procedure determines the number of quadrature points adaptively, the number of nodes are increased until the QMAX=*n* limit is reached or until two successive evaluations of the log likelihood have a relative change of less than r. In the latter case, the lesser number of quadrature nodes is used for the optimization.

The EBSSFRAC, EBSSTOL, EBSTEPS, EBSUBSTEPS, and EBTOL suboptions affect the suboptimization that leads to the empirical Bayes estimates of the random effects. Under normal circumstances, there is no reason to change from the default values. When the suboptimizations fail, the optimization process can come to a halt. If the EBDETAILS option is in effect, you might be able to determine why the suboptimization fails and then adjust these values accordingly.

The QMIN, QMAX, QTOL, and QFAC suboptions determine the quadrature point search sequence for the adaptive component of estimation.

As for METHOD=LAPLACE, certain features of the marginal distribution are not available because the marginal likelihood of the data is approximated numerically. For example, you cannot display a marginal variance-covariance matrix. Also, the procedure includes both the fixed-effects and covariance parameters in the optimization for quadrature estimation. Consequently, this setting imposes some restrictions with respect to available options. Table 38.3 lists the options that are assumed for METHOD=QUAD and METHOD=LAPLACE, and Table 38.4 lists the options that are not compatible with these estimation methods.

Table 38.3 Defaults for METHOD=LAPLACE and METHOD=QUAD

Statement	Option
PROC GLIMMIX	NOPROFILE
PROC GLIMMIX	INITGLM
MODEL	NOCENTER

Table 38.4 Options Incompatible with METHOD=LAPLACE and METHOD=QUAD

Statement	Option
PROC GLIMMIX	EXPHESSIAN
PROC GLIMMIX	SCOREMOD
PROC GLIMMIX	SCORING
PROC GLIMMIX	PROFILE
MODEL	DDFM=KENWARDROGER
MODEL	DDFM=SATTERTHWAITE
MODEL	STDCOEF
RANDOM	RESIDUAL
RANDOM _RESIDUAL_	All R-side random effects
RANDOM	V
RANDOM	VC
RANDOM	VCI
RANDOM	VCORR
RANDOM	VI

In addition to the options displayed in Table 38.4, the NOBOUND option in the PROC GLIMMIX and the NOBOUND option in the PARMS statements are not available with METHOD=QUAD. Unbounding the covariance parameter estimates is possible with METHOD=LAPLACE, however.

No Random Effects Present

If the model does not contain G-side random effects or contains only a single overdispersion component, then the model belongs to the family of (overdispersed) generalized linear models if the distribution is known or the quasi-likelihood models for independent data if the distribution is not known. The GLIMMIX procedure then estimates model parameters by the following techniques:

- normally distributed data: residual maximum likelihood
- nonnormal data: maximum likelihood
- data with unknown distribution: quasi-likelihood

The METHOD= specification then has only an effect with respect to the divisor used in estimating the overdispersion component. With a residual method, the divisor is $f - k$, where f denotes the sum of the frequencies and k is the rank of $\mathbf{X}$. Otherwise, the divisor is f.

NAMELEN=*number*
specifies the length to which long effect names are shortened. The default and minimum value is 20.

NOBOUND
requests the removal of boundary constraints on covariance and scale parameters in mixed models. For example, variance components have a default lower boundary constraint of 0, and the NOBOUND option allows their estimates to be negative.

The NOBOUND option cannot be used for adaptive quadrature estimation with METHOD=QUAD. The scaling of the quadrature abscissas requires an inverse Cholesky root that is possibly not well defined when the $\mathbf{G}$ matrix of the mixed model is negative definite or indefinite. The Laplace approximation (METHOD=LAPLACE) is not subject to this limitation.

NOBSDETAIL
adds detailed information to the "Number of Observations" table to reflect how many observations were excluded from the analysis and for which reason.

NOCLPRINT< =*number* **>**
suppresses the display of the "Class Level Information" table, if you do not specify *number*. If you specify *number*, only levels with totals that are less than *number* are listed in the table.

NOFIT
suppresses fitting of the model. When the NOFIT option is in effect, PROC GLIMMIX produces the "Model Information," "Class Level Information," "Number of Observations," and "Dimensions" tables. These can be helpful to gauge the computational effort required to fit the model. For example, the "Dimensions" table informs you as to whether the GLIMMIX procedure processes the data by subjects, which is typically more computationally efficient than processing the data as a single subject. See the section "Processing by Subjects" for more information.

If you request a radial smooth with knot selection by *k-d* tree methods, PROC GLIMMIX also computes the knot locations of the smoother. You can then examine the knots without fitting the model. This enables you to try out different knot construction methods and bucket sizes. See the KNOTMETHOD=KDTREE option (and its suboptions) of the RANDOM statement.

If you combine the NOFIT option with the OUTDESIGN option, you can write the **X** and/or **Z** matrix of your model to a SAS data set without fitting the model.

NOINITGLM

requests that the starting values for the fixed effects not be obtained by first fitting a generalized linear model. This option is the default for the pseudo-likelihood estimation methods and for the linear mixed model. For the pseudo-likelihood methods, starting values can be implicitly defined based on an initial pseudo-data set derived from the data and the link function. For linear mixed models, starting values for the fixed effects are not necessary. The NOINITGLM option is useful in conjunction with the INITPL= suboption of METHOD=QUAD in order to perform initial pseudo-likelihood steps prior to an adaptive quadrature.

NOITPRINT

suppresses the display of the "Iteration History" table.

NOPROFILE

includes the scale parameter ϕ into the optimization for models that have such a parameter (see Table 38.15). By default, the GLIMMIX procedure profiles scale parameters from the optimization in mixed models. In generalized linear models, scale parameters are not profiled.

NOREML

determines the denominator for the computation of the scale parameter in a GLM for normal data and for overdispersion parameters. By default, the GLIMMIX procedure computes the scale parameter for the normal distribution as

$$\widehat{\phi} = \sum_{i=1}^{n} \frac{f_i (y_i - \widehat{y}_i)^2}{f - k}$$

where k is the rank of $\mathbf{X}$, f_i is the frequency associated with the ith observation, and $f = \sum f_i$. Similarly, the overdispersion parameter in an overdispersed GLM is estimated by the ratio of the Pearson statistic and $(f - k)$. If the NOREML option is in effect, the denominators are replaced by f, the sum of the frequencies. In a GLM for normal data, this yields the maximum likelihood estimate of the error variance. For this case, the NOREML option is a convenient way to change from REML to ML estimation.

In GLMM models fit by pseudo-likelihood methods, the NOREML option changes the estimation method to the nonresidual form. See the METHOD= option for the distinction between residual and nonresidual estimation methods.

ODDSRATIO

OR

requests that odds ratios be added to the output when applicable. Odds ratios and their confidence limits are reported only for models with logit, cumulative logit, or generalized logit link. Specifying the ODDSRATIO option in the PROC GLIMMIX statement has the same effect as specifying the ODDSRATIO option in the MODEL statement and in all LSMEANS

statements. Note that the ODDSRATIO option in the MODEL statement has several suboptions that enable you to construct customized odds ratios. These suboptions are available only through the MODEL statement. For details about the interpretation and computation of odds and odds ratios with the GLIMMIX procedure, see the section "Odds and Odds Ratio Estimation" on page 2256.

ORDER=DATA

ORDER=FORMATTED

ORDER=FREQ

ORDER=INTERNAL

specifies the sorting order for the levels of CLASS variables. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option can be useful when you use CONTRAST or ESTIMATE statements.

When the default ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. To order numeric class levels with no explicit format by their BEST12. formatted values, you can specify this format explicitly for the CLASS variables.

The following table shows how PROC GLIMMIX interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

For FORMATTED and INTERNAL values, the sort order is machine dependent.

When the response variable appears in a CLASS statement, the ORDER= option in the PROC GLIMMIX statement applies to its sort order. When you specify a *response-option* in the MODEL statement, this overrides the ORDER= option in the PROC GLIMMIX statement. For example, in the following statements the sort order of the wheeze variable is determined by the formatted value (default):

```
proc glimmix order=data;
   class city;
   model wheeze = city age / dist=binary s;
run;
```

The ORDER= option in the PROC GLIMMIX statement has no effect on the sort order of the wheeze variable because it does not appear in the CLASS statement. However, in the following statements the sort order of the wheeze variable is determined by the order of appearance in the input data set because the response variable appears in the CLASS statement:

```
proc glimmix order=data;
   class city wheeze;
   model wheeze = city age / dist=binary s;
run;
```

For more information about sort order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

OUTDESIGN<(*options*)><=*SAS-data-set*>
creates a data set that contains the contents of the **X** and **Z** matrix. If the data are processed by subjects as shown in the "Dimensions" table, then the **Z** matrix saved to the data set corresponds to a single subject. By default, the GLIMMIX procedure includes in the OUTDESIGN data set the **X** and **Z** matrix (if present) and the variables in the input data set. You can specify the following *options* in parentheses to control the contents of the OUTDESIGN data set:

NAMES
produces tables associating columns in the OUTDESIGN data set with fixed-effects parameter estimates and random-effects solutions.

NOMISS
excludes from the OUTDESIGN data set observations that were not used in the analysis.

NOVAR
excludes from the OUTDESIGN data set variables from the input data set. Variables listed in the BY and ID statements and variables needed for identification of SUBJECT= effects are always included in the OUTDESIGN data set.

X<=*prefix*>
saves the contents of the **X** matrix. The optional *prefix* is used to name the columns. The default naming prefix is "_X".

Z<=*prefix*>
saves the contents of the **Z** matrix. The optional *prefix* is used to name the columns. The default naming prefix is "_Z".

The order of the observations in the OUTDESIGN data set is the same as the order of the input data set. If you do not specify a data set with the OUTDESIGN option, the procedure uses the DATA*n* convention to name the data set.

PCONV=*r*
specifies the parameter estimate convergence criterion for doubly iterative estimation methods. The GLIMMIX procedure applies this criterion to fixed-effects estimates and covariance parameter estimates. Suppose $\widehat{\psi}_i^{(u)}$ denotes the estimate of the ith parameter at the uth optimization. The procedure terminates the doubly iterative process if the largest value

$$2 \times \frac{|\widehat{\psi}_i^{(u)} - \widehat{\psi}_i^{(u-1)}|}{|\widehat{\psi}_i^{(u)}| + |\widehat{\psi}_i^{(u-1)}|}$$

is less than r. To check an absolute convergence criteria as well, you can set the ABSPCONV= option in the PROC GLIMMIX statement. The default value for r is 1E8 times the machine epsilon, a product that equals about 1E−8 on most machines.

Note that this convergence criterion does not affect the convergence criteria applied within any individual optimization. In order to change the convergence behavior within an optimization, you can use the ABSCONV=, ABSFCONV=, ABSGCONV=, ABSXCONV=, FCONV=, or GCONV= option in the NLOPTIONS statement.

PLOTS <(*global-plot-options***)> <= *plot-request* <(***options***)>>**
PLOTS <(*global-plot-options***)> <= (*plot-request* <(***options***)>** **<... *plot-request* <(***options***)>>)>**

requests that the GLIMMIX procedure produce statistical graphics via the Output Delivery System, provided that the ODS GRAPHICS statement has been specified. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For examples of the basic statistical graphics produced by the GLIMMIX procedure and aspects of their computation and interpretation, see the section "ODS Graphics" on page 2282 in this chapter. You can also request statistical graphics for least squares means through the PLOTS option in the LSMEANS statement, which gives you more control over the display compared to the PLOTS option in the PROC GLIMMIX statement.

Global Plot Options

The *global-plot-options* apply to all relevant plots generated by the GLIMMIX procedure. The *global-plot-options* supported by the GLIMMIX procedure are as follows:

OBSNO
uses the data set observation number to identify observations in tooltips, provided that the observation number can be determined. Otherwise, the number displayed in tooltips is the index of the observation as it is used in the analysis within the BY group.

UNPACKPANEL
UNPACK
breaks a graphic that is otherwise paneled into individual component plots.

Specific Plot Options

The following listing describes the specific plots and their options.

ALL
requests that all plots appropriate for the analysis be produced. In models with G-side random effects, residual plots are based on conditional residuals (by using the BLUPs of random effects) on the linear (linked) scale. Plots of least squares means differences are produced for LSMEANS statements without options that would contradict such a display.

ANOMPLOT
ANOM

requests an analysis of means display in which least squares means are compared against an average least squares mean (Ott 1967; Nelson 1982, 1991, 1993). See the DIFF= option in the LSMEANS statement for the computation of this average. Least squares mean ANOM plots are produced only for those fixed effects that are listed in LSMEANS statements that have options that do not contradict the display. For example, if you request ANOM plots with the PLOTS= option in the PROC GLIMMIX statement, the following LSMEANS statements produce analysis of mean plots for effects A and C:

```
lsmeans A / diff=anom;
lsmeans B / diff;
lsmeans C;
```

The DIFF option in the second LSMEANS statement implies all pairwise differences.

When differences against the average LS-mean are adjusted for multiplicity with the ADJUST=NELSON option in the LSMEANS statement, the ANOMPLOT display is adjusted accordingly.

BOXPLOT < (*boxplot-options*) >

requests box plots for the effects in your model that consist of classification effects only. Note that these effects can involve more than one classification variable (interaction and nested effects), but cannot contain any continuous variables. By default, the BOXPLOT request produces box plots of (conditional) residuals for the qualifying effects in the MODEL and RANDOM statements. See the discussion of the *boxplot-options* in a later section for information about how to tune your box plot request.

CONTROLPLOT
CONTROL

requests a display in which least squares means are visually compared against a reference level. LS-mean control plots are produced only for those fixed effects that are listed in LSMEANS statements that have options that do not contradict with the display. For example, the following statements produce control plots for effects A and C if you specify PLOTS=CONTROL in the PROC GLIMMIX statement:

```
lsmeans A / diff=control('1');
lsmeans B / diff;
lsmeans C;
```

The DIFF option in the second LSMEANS statement implies all pairwise differences.

When differences against a control level are adjusted for multiplicity with the ADJUST= option in the LSMEANS statement, the control plot display is adjusted accordingly.

DIFFPLOT< (*diffplot-options*) >
DIFFOGRAM < (*diffplot-options*) >
DIFF< (*diffplot-options*) >

requests a display of all pairwise least squares mean differences and their significance.

When constructed from arithmetic means, the display is also known as a "mean-mean scatter plot" (Hsu 1996; Hsu and Peruggia 1994). For each comparison a line segment, centered at the LS-means in the pair, is drawn. The length of the segment corresponds to the projected width of a confidence interval for the least squares mean difference. Segments that fail to cross the 45-degree reference line correspond to significant least squares mean differences.

If you specify the ADJUST= option in the LSMEANS statement, the lengths of the line segments are adjusted for multiplicity.

LS-mean difference plots are produced only for those fixed effects listed in LSMEANS statements that have options that do not conflict with the display. For example, the following statements request differences against a control level for the A effect, all pairwise differences for the B effect, and the least squares means for the C effect:

```
lsmeans A / diff=control('1');
lsmeans B / diff;
lsmeans C;
```

The DIFF= type in the first statement contradicts a display of all pairwise differences. Difference plots are produced only for the B and C effects if you specify PLOTS=DIFF in the PROC GLIMMIX statement.

You can specify the following *diffplot-options*. The ABS and NOABS options determine the positioning of the line segments in the plot. When the ABS option is in effect (this is the default) all line segments are shown on the same side of the reference line. The NOABS option separates comparisons according to the sign of the difference. The CENTER option marks the center point for each comparison. This point corresponds to the intersection of two least squares means. The NOLINES option suppresses the display of the line segments that represent the confidence bounds for the differences of the least squares means. The NOLINES option implies the CENTER option. The default is to draw line segments in the upper portion of the plot area without marking the center point.

MEANPLOT< (*meanplot-options*) >
requests a display of the least squares means of effects specified in LSMEANS statements. The following *meanplot-options* affect the display. Upper and lower confidence limits are plotted when the CL option is used. When the CLBAND option is in effect, confidence limits are shown as bands and the means are connected. By default, least squares means are not joined by lines. You can achieve that effect with the JOIN or CONNECT option. Least squares means are displayed in the same order in which they appear in the "Least Squares Means" table. You can change that order for plotting purposes with the ASCENDING and DESCENDING options. The ILINK option requests that results be displayed on the inverse linked (the data) scale.

Note that there is also a MEANPLOT suboption of the PLOTS= option in the LSMEANS statement. In addition to the *meanplot-options* just described, you can also specify classification effects that give you more control over the display of interaction means through the PLOTBY= and SLICEBY= options. To display interaction means, you typically want to use the MEANPLOT option in the LSMEANS statement. For

example, the next statement requests a plot in which the levels of A are placed on the horizontal axis and the means that belong to the same level of B are joined by lines:

```
lsmeans A*B / plot=meanplot(sliceby=b join);
```

NONE
requests that no plots be produced.

ODDSRATIO < (*oddsratioplot-options*) >
requests a display of odds ratios and their confidence limits when the link function permits the computation of odds ratios (see the ODDSRATIO option in the MODEL statement). Possible suboptions of the ODDSRATIO plot request are described below under the heading "Odds Ratio Plot Options."

RESIDUALPANEL< (*residualplot-options*) >
requests a paneled display constructed from raw residuals. The panel consists of a plot of the residuals against the linear predictor or predicted mean, a histogram with normal density overlaid, a *Q*-*Q* plot, and a box plot of the residuals. The *residualplot-options* enable you to specify which type of residual is being graphed. These are further discussed below under the heading "Residual Plot Options."

STUDENTPANEL< (*residualplot-options*) >
requests a paneled display constructed from studentized residuals. The same panel organization is applied as for the RESIDUALPANEL plot type.

PEARSONPANEL< (*residualplot-options*) >
requests a paneled display constructed from Pearson residuals. The same panel organization is applied as for the RESIDUALPANEL plot type.

Residual Plot Options

The *residualplot-options* apply to the RESIDUALPANEL, STUDENTPANEL, and PEARSONPANEL displays. The primary function of these options is to control which type of a residual to display. The four types correspond to *keyword-options* as for output statistics in the OUTPUT statement. The *residualplot-options* take on the following values:

BLUP
CONDITIONAL
uses the predictors of the random effects in computing the residual.

ILINK
NONLINEAR
computes the residual on the inverse linked scale (the data scale).

NOBLUP
MARGINAL
does not use the predictors of the random effects in computing the residual.

NOILINK

LINEAR

computes the residual on the linked scale.

UNPACK

produces separate plots from the elements of the panel.

You can list a plot request one or more times with different options. For example, the following statements request a panel of marginal raw residuals, individual plots generated from a panel of the conditional raw residuals, and a panel of marginal studentized residuals:

```
ods graphics on;
proc glimmix plots=(ResidualPanel(marginal)
                    ResidualPanel(unpack conditional)
                    StudentPanel(marginal));
```

The default is to compute conditional residuals on the linear scale if the model contains G-side random effects (BLUP NOILINK). Not all combinations of the BLUP/NOBLUP and ILINK/NOILINK suboptions are possible for all residual types and models. For details, see the description of output statistics for the OUTPUT statement. Pearson residuals are always displayed against the linear predictor; all other residuals are graphed versus the linear predictor if the NOILINK suboption is in effect (default), and against the corresponding prediction on the mean scale if the ILINK option is in effect. See Table 38.11 for a definition of the residual quantities and exclusions.

Box Plot Options

The *boxplot-options* determine whether box plots are produced for residuals or for residuals and observed values, and for which model effects the box plots are constructed. The available *boxplot-options* are as follows:

BLOCK

BLOCKLEGEND

displays levels of up to four classification variables of the box plot effect by using block legends instead of axis tick values.

BLUP

CONDITIONAL

constructs box plots from conditional residuals—that is, residuals that use the estimated BLUPs of random effects.

FIXED

produces box plots for all fixed effects (MODEL statement) consisting entirely of classification variables.

GROUP
produces box plots for all GROUP= effects in RANDOM statements consisting entirely of classification variables.

ILINK
NONLINEAR
computes the residual on the scale of the data (the inverse linked scale).

NOBLUP
MARGINAL
constructs box plots from marginal residuals.

NOILINK
LINEAR
computes the residual on the linked scale.

NPANELPOS=*number*
specifies the number of box positions on the graphic and provides the capability to break a box plot into multiple graphics. If *number* is negative, no balancing of the number of boxes takes place and *number* is the maximum number of boxes per graphic. If *number* is positive, the number of boxes per graphic is balanced. For example, suppose that variable A has 125 levels. The following statements request that the number of boxes per plot results be balanced and result in six plots with 18 boxes each and one plot with 17 boxes:

```
ods graphics on;
proc glimmix plots=boxplot(npanelpos=20);
    class A;
    model y = A;
run;
```

If *number* is zero (this is the default), all levels of the effect are displayed in a single plot.

OBSERVED
adds box plots of the observed data for the selected effects.

PEARSON
constructs box plots from Pearson residuals rather than from the default residuals.

PSEUDO
adds box plots of the pseudo-data for the selected effects. This option is available only for the pseudo-likelihood estimation methods that construct pseudo-data.

RANDOM
produces box plots for all effects in RANDOM statements that consist entirely of classification variables. This does not include effects specified in the GROUP= or SUBJECT= option of the RANDOM statements.

RAW
constructs box plots from raw residuals (observed minus predicted).

STUDENT
constructs box plots from studentized residuals rather than from the default residuals.

SUBJECT
produces box plots for all SUBJECT= effects in RANDOM statements consisting entirely of classification variables.

USEINDEX
uses as the horizontal axis label the index of the effect level, rather than the formatted value(s). For classification variables with many levels or model effects that involve multiple classification variables, the formatted values identifying the effect levels might take up too much space as axis tick values, leading to extensive thinning. The USEINDEX option replaces tick values constructed from formatted values with the internal level number.

By default, box plots of residuals are constructed from the raw conditional residuals (on the linked scale) in linear mixed models and from Pearson residuals in all other models. Note that not all combinations of the BLUP/NOBLUP and ILINK/NOILINK suboptions are possible for all residual types and models. For details, see the description of output statistics for the OUTPUT statement.

Odds Ratio Plot Options

The *oddsratioplot-options* determine the display of odds ratios and their confidence limits. The computation of the odds ratios follows the ODDSRATIO option in the MODEL statement. The available *oddsratioplot-options* are as follows:

LOGBASE= 2 | E | 10
log-scales the odds ratio axis.

NPANELPOS=*n*
provides the capability to break an odds ratio plot into multiple graphics having at most $|n|$ odds ratios per graphic. If n is positive, then the number of odds ratios per graphic is balanced. If n is negative, then no balancing of the number of odds ratios takes place. For example, suppose you want to display 21 odds ratios. Then NPANELPOS=20 displays two plots, the first with 11 and the second with 10 odds ratios, and NPANELPOS=−20 displays 20 odds ratios in the first plot and a single odds ratio in the second. If $n = 0$ (this is the default), then all odds ratios are displayed in a single plot.

ORDER=ASCENDING | DESCENDING
displays the odds ratios in sorted order. By default the odds ratios are displayed in the order in which they appear in the "Odds Ratio Estimates" table.

RANGE=(< *min* **> <** *,max* **>) | CLIP**
specifies the range of odds ratios to display. If you specify RANGE=CLIP, then the confidence intervals are clipped and the range contains the minimum and maximum odds ratios. By default the range of view captures the extent of the odds ratio confidence intervals.

STATS
adds the numeric values of the odds ratio and its confidence limits to the graphic.

PROFILE
requests that scale parameters be profiled from the optimization, if possible. This is the default for generalized linear mixed models. In generalized linear models with normally distributed data, you can use the PROFILE option to request profiling of the residual variance.

SCOREMOD
requests that the Hessian matrix in GLMMs be based on a modified scoring algorithm, provided that PROC GLIMMIX is in scoring mode when the Hessian is evaluated. The procedure is in scoring mode during iteration, if the optimization technique requires second derivatives, the SCORING=*n* option is specified, and the iteration count has not exceeded *n*. The procedure also computes the expected (scoring) Hessian matrix when you use the EXPHESSIAN option in the PROC GLIMMIX statement.

The SCOREMOD option has no effect if the SCORING= or EXPHESSIAN option is not specified. The nature of the SCOREMOD modification to the expected Hessian computation is shown in Table 38.17, in the section "Pseudo-likelihood Estimation Based on Linearization" on page 2220. The modification can improve the convergence behavior of the GLMM compared to standard Fisher scoring and can provide a better approximation of the variability of the covariance parameters. For more details, see the section "Estimated Precision of Estimates" on page 2221.

SCORING=*number*
requests that Fisher scoring be used in association with the estimation method up to iteration *number*. By default, no scoring is applied. When you use the SCORING= option and PROC GLIMMIX converges without stopping the scoring algorithm, the procedure uses the expected Hessian matrix to compute approximate standard errors for the covariance parameters instead of the observed Hessian. If necessary, the standard errors of the covariance parameters as well as the output from the ASYCOV and ASYCORR options are adjusted.

If scoring stopped prior to convergence and you want to use the expected Hessian matrix in the computation of standard errors, use the EXPHESSIAN option in the PROC GLIMMIX statement.

Scoring is not possible in models for nominal data. It is also not possible for GLMs with unknown distribution or for those outside the exponential family. If you perform quasi-likelihood estimation, the GLIMMIX procedure is always in scoring mode and the SCORING= option has no effect. See the section "Quasi-likelihood for Independent Data" for a description of the types of models where GLIMMIX applies quasi-likelihood estimation.

The SCORING= option has no effect for optimization methods that do not involve second derivatives. See the TECHNIQUE= option in the NLOPTIONS statement and the section

"Choosing an Optimization Algorithm" on page 405 in Chapter 18, "Shared Concepts and Topics," for details about first- and second-order algorithms.

SINGCHOL=*number*
tunes the singularity criterion in Cholesky decompositions. The default is 1E4 times the machine epsilon; this product is approximately 1E−12 on most computers.

SINGRES=*number*
sets the tolerance for which the residual variance is considered to be zero. The default is 1E4 times the machine epsilon; this product is approximately 1E−12 on most computers.

SINGULAR=*number*
tunes the general singularity criterion applied by the GLIMMIX procedure in divisions and inversions. The default is 1E4 times the machine epsilon; this product is approximately 1E−12 on most computers.

STARTGLM
is an alias of the INITGLM option.

SUBGRADIENT< =*SAS-data-set* **>**
SUBGRAD< =*SAS-data-set* **>**
creates a data set with information about the gradient of the objective function. The contents and organization of the SUBGRADIENT= data set depend on the type of model. The following paragraphs describe the SUBGRADIENT= data set for the two major estimation modes. See the section "GLM Mode or GLMM Mode" on page 2233 for details about the estimation modes of the GLIMMIX procedure.

GLMM Mode If the GLIMMIX procedure operates in GLMM mode, the SUBGRADIENT= data set contains as many observations as there are usable subjects in the analysis. The maximum number of usable subjects is displayed in the "Dimensions" table. Gradient information is not written to the data set for subjects who do not contribute valid observations to the analysis. Note that the objective function in the "Iteration History" table is in terms of the −2 log (residual, pseudo-) likelihood. The gradients in the SUBGRADIENT= data set are gradients of that objective function.

The gradients are evaluated at the final solution of the estimation problem. If the GLIMMIX procedure fails to converge, then the information in the SUBGRADIENT= data set corresponds to the gradient evaluated at the last iteration or optimization.

The number of gradients saved to the SUBGRADIENT= data set equals the number of parameters in the optimization. For example, with METHOD=LAPLACE or METHOD=QUAD the fixed-effects parameters and the covariance parameters take part in the optimization. The order in which the gradients appear in the data set equals the order in which the gradients are displayed when the ITDETAILS option is in effect: gradients for fixed-effects parameters precede those for covariance parameters, and gradients are not reported for singular columns in the $\mathbf{X}'\mathbf{X}$ matrix. In models where the residual variance is profiled from the optimization,

a subject-specific gradient is not reported for the residual variance. To decompose this gradient by subjects, add the NOPROFILE option in the PROC GLIMMIX statement. When the subject-specific gradients in the SUBGRADIENT= data set are summed, the totals equal the values reported by the GRADIENT option.

GLM Mode When you fit a generalized linear model (GLM) or a GLM with overdispersion, the SUBGRADIENT= data set contains the observation-wise gradients of the negative log-likelihood function with respect to the parameter estimates. Note that this corresponds to the objective function in GLMs as displayed in the "Iteration History" table. However, the gradients displayed in the "Iteration History" for GLMs—when the ITDETAILS option is in effect—are possibly those of the centered and scaled coefficients. The gradients reported in the "Parameter Estimates" table and in the SUBGRADIENT= data set are gradients with respect to the uncentered and unscaled coefficients.

The gradients are evaluated at the final estimates. If the model does not converge, the gradients contain missing values. The gradients appear in the SUBGRADIENT= data set in the same order as in the "Parameter Estimates" table, with singular columns removed.

The variables from the input data set are added to the SUBGRADIENT= data set in GLM mode. The data set is organized in the same way as the input data set; observations that do not contribute to the analysis are transferred to the SUBGRADIENT= data set, but gradients are calculated only for observations that take part in the analysis. If you use an ID statement, then only the variables in the ID statement are transferred to the SUBGRADIENT= data set.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC GLIMMIX to obtain separate analyses for observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement options NOTSORTED or DESCENDING in the BY statement for the GLIMMIX procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.

- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

Because sorting the data changes the order in which PROC GLIMMIX reads observations, the sorting order for the levels of the CLASS variable might be affected if you have specified ORDER=DATA in the PROC GLIMMIX statement. This, in turn, affects specifications in the CONTRAST, ESTIMATE, or LSMESTIMATE statement.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

The CLASS statement names the classification variables to be used in the analysis. If the CLASS statement is used, it must appear before the MODEL statement.

Classification variables can be either character or numeric. By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. Prior to SAS 9, class levels were determined by using no more than the first 16 characters of the formatted values. If you want to revert to this previous behavior you can use the TRUNCATE option in the CLASS statement. In any case, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *Base SAS Procedures Guide* and the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*. You can adjust the order of CLASS variable levels with the ORDER= option in the PROC GLIMMIX statement.

You can specify the following option in the CLASS statement after a slash (/).

TRUNCATE
specifies that class levels should be determined by using no more than the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases previous to SAS 9.

CONTRAST Statement

CONTRAST *'label' contrast-specification*
< , contrast-specification > < , . . . >
< / options > **;**

The CONTRAST statement provides a mechanism for obtaining custom hypothesis tests. It is patterned after the CONTRAST statement in PROC MIXED and enables you to select an appropriate inference space (McLean, Sanders, and Stroup 1991). The GLIMMIX procedure gives you greater flexibility in entering contrast coefficients for random effects, however, because it permits the usual *value*-oriented positional syntax for entering contrast coefficients, as well as a level-oriented syntax that simplifies entering coefficients for interaction terms and is designed to work with constructed effects that are defined through the experimental EFFECT statement. The differences between the traditional and new-style coefficient syntax are explained in detail in the section "Positional and Nonpositional Syntax for Contrast Coefficients" on page 2264.

You can test the hypothesis $\mathbf{L}'\boldsymbol{\phi} = \mathbf{0}$, where $\mathbf{L}' = [\mathbf{K}' \; \mathbf{M}']$ and $\boldsymbol{\phi}' = [\boldsymbol{\beta}' \; \boldsymbol{\gamma}']$, in several inference spaces. The inference space corresponds to the choice of $\mathbf{M}$. When $\mathbf{M} = \mathbf{0}$, your inferences apply to the entire population from which the random effects are sampled; this is known as the *broad* inference space. When all elements of $\mathbf{M}$ are nonzero, your inferences apply only to the observed levels of the random effects. This is known as the *narrow* inference space, and you can also choose it by specifying all of the random effects as fixed. The GLM procedure uses the narrow inference space. Finally, by zeroing portions of $\mathbf{M}$ corresponding to selected main effects and interactions, you can choose *intermediate* inference spaces. The broad inference space is usually the most appropriate; it is used when you do not specify random effects in the CONTRAST statement.

In the CONTRAST statement,

label identifies the contrast in the table. A label is required for every contrast specified. Labels can be up to 200 characters and must be enclosed in quotes.

contrast-specification identifies the fixed effects and random effects and their coefficients from which the $\mathbf{L}$ matrix is formed. The syntax representation of a *contrast-specification* is
< fixed-effect values . . . > < | random-effect values . . . >

fixed-effect identifies an effect that appears in the MODEL statement. The keyword INTERCEPT can be used as an effect when an intercept is fitted in the model. You do not need to include all effects that are in the MODEL statement.

random-effect identifies an effect that appears in the RANDOM statement. The first random effect must follow a vertical bar (|); however, random effects do not have to be specified.

values are constants that are elements of the $\mathbf{L}$ matrix associated with the fixed and random effects. There are two basic methods of specifying the entries of the $\mathbf{L}$ matrix. The traditional representation—also known as the positional syntax—relies on entering coefficients in the position they assume in the $\mathbf{L}$ matrix. For example, in the following statements the elements of $\mathbf{L}$ associated with the b main effect receive a 1 in the first position and a -1 in the second position:

```
class a b;
model y = a b a*b;
contrast 'B at A2' b 1 -1  a*b 0  0  1 -1;
```

The elements associated with the interaction receive a 1 in the third position and a -1 in the fourth position. In order to specify coefficients correctly for the interaction term, you need to know how the levels of a and b vary in the interaction, which is governed by the order of the variables in the CLASS statement. The nonpositional syntax is designed to make it easier to enter coefficients for interactions and is necessary to enter coefficients for effects constructed with the experimental EFFECT statement. In square brackets you enter the coefficient followed by the associated levels of the CLASS variables. If B has two and A has three levels, the previous CONTRAST statement, by using nonpositional syntax for the interaction term, becomes

```
contrast 'B at A2' b 1 -1 a*b [1, 2 1] [-1, 2 2];
```

It assigns value 1 to the interaction where A is at level 2 and B is at level 1, and it assigns -1 to the interaction where both classification variables are at level 2. The comma separating the entry for the **L** matrix from the level indicators is optional. Further details about the nonpositional contrast syntax and its use with constructed effects can be found in the section "Positional and Nonpositional Syntax for Contrast Coefficients" on page 2264. Nonpositional syntax is available only for fixed-effects coefficients.

The rows of $\mathbf{L}'$ are specified in order and are separated by commas. The rows of the $\mathbf{K}'$ component of $\mathbf{L}'$ are specified on the left side of the vertical bars (|). These rows test the fixed effects and are, therefore, checked for estimability. The rows of the $\mathbf{M}'$ component of $\mathbf{L}'$ are specified on the right side of the vertical bars. They test the random effects, and no estimability checking is necessary.

If PROC GLIMMIX finds the fixed-effects portion of the specified contrast to be nonestimable (see the SINGULAR= option), then it displays missing values for the test statistics.

If the elements of **L** are not specified for an effect that contains a specified effect, then the elements of the unspecified effect are automatically "filled in" over the levels of the higher-order effect. This feature is designed to preserve estimability for cases where there are complex higher-order effects. The coefficients for the higher-order effect are determined by equitably distributing the coefficients of the lower-level effect as in the construction of least squares means. In addition, if the intercept is specified, it is distributed over all classification effects that are not contained by any other specified effect. If an effect is not specified and does not contain any specified effects, then all of its coefficients in **L** are set to 0. You can override this behavior by specifying coefficients for the higher-order effect.

If too many values are specified for an effect, the extra ones are ignored; if too few are specified, the remaining ones are set to 0. If no random effects are specified, the vertical bar can be omitted; otherwise, it must be present. If a SUBJECT effect is used in the RANDOM statement, then the coefficients specified for the effects in the RANDOM statement are equitably distributed across the levels of the SUBJECT effect. You can use the E option to see exactly what **L** matrix is used.

PROC GLIMMIX handles missing level combinations of classification variables similarly to PROC

GLM and PROC MIXED. These procedures delete fixed-effects parameters corresponding to missing levels in order to preserve estimability. However, PROC MIXED and PROC GLIMMIX do not delete missing level combinations for random-effects parameters, because linear combinations of the random-effects parameters are always estimable. These conventions can affect the way you specify your CONTRAST coefficients.

The CONTRAST statement computes the statistic

$$F = \frac{\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}' \mathbf{L}(\mathbf{L}'\mathbf{C}\mathbf{L})^{-1}\mathbf{L}' \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}}{r}$$

where $r = \text{rank}(\mathbf{L}'\mathbf{C}\mathbf{L})$, and approximates its distribution with an F distribution unless DDFM=NONE. If you select DDFM=NONE as the degrees-of-freedom method in the MODEL statement, and if you do not assign degrees of freedom to the contrast with the DF= option, then PROC GLIMMIX computes the test statistic $r \times F$ and approximates its distribution with a chi-square distribution. In the expression for F, $\mathbf{C}$ is an estimate of $\text{Var}[\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}]$; see the section "Estimated Precision of Estimates" on page 2221 and the section "Aspects Common to Adaptive Quadrature and Laplace Approximation" on page 2230 for details about the computation of $\mathbf{C}$ in a generalized linear mixed model.

The numerator degrees of freedom in the F approximation and the degrees of freedom in the chi-square approximation are equal to r. The denominator degrees of freedom are taken from the "Tests of Fixed Effects" table and correspond to the final effect you list in the CONTRAST statement. You can change the denominator degrees of freedom by using the DF= option.

You can specify the following options in the CONTRAST statement after a slash (/).

BYCATEGORY

BYCAT

requests that in models for nominal data (generalized logit models) the contrasts not be combined across response categories but reported separately for each category. For example, assume that the response variable Style is multinomial with three (unordered) categories. The following GLIMMIX statements fit a generalized logit model relating the preferred style of instruction to school and educational program effects:

```
proc glimmix data=school;
   class School Program;
   model Style(order=data) = School Program / s ddfm=none
                             dist=multinomial link=glogit;
   freq Count;
   contrast 'School 1 vs. 2' school 1 -1;
   contrast 'School 1 vs. 2' school 1 -1 / bycat;
run;
```

The first contrast compares school effects in all categories. This is a two-degrees-of-freedom contrast because there are two nonredundant categories. The second CONTRAST statement produces two single-degree-of-freedom contrasts, one for each nonreference Style category.

The BYCATEGORY option has no effect unless your model is a generalized (mixed) logit model.

CHISQ

requests that chi-square tests be performed for all contrasts in addition to any F tests. A chi-square statistic equals its corresponding F statistic times the numerator degrees of freedom, and these same degrees of freedom are used to compute the p-value for the chi-square test. This p-value will always be less than that for the F test, because it effectively corresponds to an F test with infinite denominator degrees of freedom.

DF=*number*

specifies the denominator degrees of freedom for the F test. For the degrees of freedom methods DDFM=BETWITHIN, DDFM=CONTAIN, and DDFM=RESIDUAL, the default is the denominator degrees of freedom taken from the "Tests of Fixed Effects" table and corresponds to the final effect you list in the CONTRAST statement. For DDFM=NONE, infinite denominator degrees of freedom are assumed by default, and for DDFM=SATTERTHWAITE and DDFM=KENWARDROGER, the denominator degrees of freedom are computed separately for each contrast.

E

requests that the $\mathbf{L}$ matrix coefficients for the contrast be displayed.

GROUP *coeffs*

sets up random-effect contrasts between different groups when a GROUP= variable appears in the RANDOM statement. By default, CONTRAST statement coefficients on random effects are distributed equally across groups. If you enter a multiple row contrast, you can also enter multiple rows for the GROUP coefficients. If the number of GROUP coefficients is less than the number of contrasts in the CONTRAST statement, the GLIMMIX procedure cycles through the GROUP coefficients. For example, the following two statements are equivalent:

```
contrast 'Trt 1 vs 2 @ x=0.4' trt 1 -1  0 | x 0.4,
                              trt 1  0 -1 | x 0.4,
                              trt 1 -1  0 | x 0.5,
                              trt 1  0 -1 | x 0.5 /
                group 1 -1, 1 0 -1, 1 -1, 1 0 -1;

contrast 'Trt 1 vs 2 @ x=0.4' trt 1 -1  0 | x 0.4,
                              trt 1  0 -1 | x 0.4,
                              trt 1 -1  0 | x 0.5,
                              trt 1  0 -1 | x 0.5 /
                group 1 -1, 1 0 -1;
```

SINGULAR=*number*

tunes the estimability checking. If $\mathbf{v}$ is a vector, define ABS($\mathbf{v}$) to be the largest absolute value of the elements of $\mathbf{v}$. If ABS($\mathbf{K}' - \mathbf{K}'\mathbf{T}$) is greater than c*$number$ for any row of $\mathbf{K}'$ in the contrast, then $\mathbf{K}'\boldsymbol{\beta}$ is declared nonestimable. Here, $\mathbf{T}$ is the Hermite form matrix $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$, and c is ABS($\mathbf{K}'$), except when it equals 0, and then c is 1. The value for *number* must be between 0 and 1; the default is 1E−4.

SUBJECT *coeffs*

sets up random-effect contrasts between different subjects when a SUBJECT= variable appears in the RANDOM statement. By default, CONTRAST statement coefficients on random effects are distributed equally across subjects. Listing subject coefficients for multiple row CONTRAST statements follows the same rules as for GROUP coefficients.

COVTEST Statement

COVTEST < *'label'* > < *test-specification* > < / *options* > **;**

The COVTEST statement provides a mechanism to obtain statistical inferences for the covariance parameters. Significance tests are based on the ratio of (residual) likelihoods or pseudo-likelihoods. Confidence limits and bounds are computed as Wald or likelihood ratio limits. You can specify multiple COVTEST statements.

The likelihood ratio test is obtained by fitting the model subject to the constraints imposed by the *test-specification*. The test statistic is formed as twice the difference of the (possibly restricted) log (pseudo-) likelihoods of the full and the reduced models. Note that fitting the null model does not necessarily require fewer computer resources compared to fitting the full model. The optimization settings for refitting the model are the same as for the full model and can be controlled with the NLOPTIONS statement.

Common questions in mixed modeling are whether variance components are zero, whether random effects are independent, and whether rows (columns) can be added or removed from an unstructured covariance matrix. When the parameters under the null hypothesis fall on the boundary of the parameter space, the distribution of the likelihood ratio statistic can be a complicated mixture of distributions. In certain situations it is known to be a relatively straightforward mixture of central chi-square distributions. When the GLIMMIX procedure recognizes the model and hypothesis as a case for which the mixture is readily available, the *p*-value of the likelihood ratio test is determined accordingly as a linear combination of central chi-square probabilities. The Note column in the "Likelihood Ratio Tests for Covariance Parameters" table along with the table's footnotes informs you about when mixture distributions are used in the calculation of *p*-values. You can find important statistical and computational details about likelihood ratio testing of covariance parameters with the GLIMMIX procedure in the section "Statistical Inference for Covariance Parameters" on page 2234.

In generalized linear mixed models that depend on pseudo-data, the GLIMMIX procedure fits the null model for a test of covariance parameters to the final pseudo-data of the converged optimization.

Test Specification

The *test-specification* in the COVTEST statement draws on keywords that represent a particular null hypothesis, lists or data sets of parameter values, or general contrast specifications. Valid keywords are as follows:

GLM | INDEP	tests the model against a null model of complete independence. All G-side covariance parameters are eliminated and the R-side covariance structure is reduced to a diagonal structure.
DIAGG	tests for a diagonal **G** matrix by constraining off-diagonal elements in **G** to zero. The R-side structure is not modified.
DIAGR | CINDEP	tests for conditional independence by reducing the R-side covariance structure to diagonal form. The G-side structure is not modified.

HOMOGENEITY
: tests homogeneity of covariance parameters across groups by imposing equality constraints. For example, the following statements fit a one-way model with heterogeneous variances and test whether the model could be reduced to a one-way analysis with the same variance across groups:

```
proc glimmix;
   class A;
   model y = a;
   random _residual_ / group=A;
   covtest 'common variance' homogeneity;
run;
```

See Example 38.9 for an application with groups and unstructured covariance matrices.

START | INITIAL
: compares the final estimates to the starting values of the covariance parameter estimates. This option is useful, for example, if you supply starting values in the PARMS statement and want to test whether the optimization produced significantly better values. In GLMMs based on pseudo-data, the likelihoods that use the starting and the final values are based on the final pseudo-data.

ZEROG
: tests whether the **G** matrix can be reduced to a zero matrix. This eliminates all G-side random effects from the model.

Only a single keyword is permitted in the COVTEST statement. To test more complicated hypotheses, you can formulate tests with the following specifications.

TESTDATA=*data-set*

TDATA=*data-set*
: reads in covariance parameter values from a SAS data set. The data set should contain the numerical variable Estimate or numerical variables named Covp*i*. The GLIMMIX procedure associates the values for Covp*i* with the *i*th covariance parameter.

  For data sets containing the numerical variable Estimate, the GLIMMIX procedure fixes the *i*th covariance parameter value at the value of the *i*th observation in the data set. A missing value indicates not to fix the particular parameter. PROC GLIMMIX performs one likelihood ratio test for the TESTDATA= data set.

  For data sets containing numerical variables named Covp*i*, the procedure performs one likelihood ratio test for each observation in the TESTDATA= data set. You do not have to specify a Covp*i* variable for every covariance parameter. If the value for the variable is not missing, PROC GLIMMIX fixes the associated covariance parameter in the null model. Consider the following statements:

```
data TestDataSet;
  input covp1 covp2 covp3;
  datalines;
. 0 .
0 0 .
. 0 0
0 0 0
;
```

```
proc glimmix method=mspl;
  class subject x;
  model y = x age x*age;
  random intercept age / sub=subject type=un;
  covtest testdata=TestDataSet;
run;
```

Because the **G** matrix is a (2×2) unstructured matrix, the first observation of the TestDataSet corresponds to zeroing the covariance between the random intercept and the random slope. When the reduced model is fit, the variances of the intercept and slope are reestimated. The second observation reduces the model to one with only a random slope in age. The third reduces the model to a random intercept model. The last observation eliminates the **G** matrix altogether.

Note that the tests associated with the first and last set of covariance parameters in TestDataSet can also be obtained by using keywords:

```
proc glimmix;
  class subject x;
  model y = x age x*age;
  random intercept age / sub=subject type=un;
  covtest DiagG;
  covtest GLM;
run;
```

value-list

supplies a list of values at which to fix the covariance parameters. A missing value in the list indicates that the covariance parameter is not fixed. If the list is shorter than the number of covariance parameters, missing values are assumed for all parameters not specified. The COVTEST statements that test the random intercept and random slope in the previous example are as follows:

```
proc glimmix;
  class subject x;
  model y = x age x*age;
  random intercept age / sub=subject type=un;
  covtest 0 0;
  covtest . 0 0;
run;
```

GENERAL *coefficients* **<,***coefficients***> <,... >**

CONTRAST *coefficients* **<,***coefficients***> <,... >**

provides a general facility to test linear combinations of covariance parameters. You can specify one or more sets of coefficients. The position of a coefficient in the list corresponds to the position of the parameter in the "Covariance Parameter Estimates" table. The linear combination of covariance parameters that is implied by each set of coefficients is tested against zero. If the list of coefficients is shorter than the number of covariance parameters, a zero coefficient is assumed for the remaining parameters.

For example, in a heterogeneous variance model with four groups, the following statements test the simultaneous hypothesis $H: \sigma_1^2 = \sigma_2^2, \sigma_3^2 = \sigma_4^2$:

```
proc glimmix;
   class A;
   model y = a;
   random _residual_ / group=A;
   covtest 'pair-wise homogeneity'
           general  1 -1  0  0,
                    0  0  1 -1;
run;
```

In a repeated measures study with four observations per subject, the COVTEST statement in the following example tests whether the four correlation parameters are identical:

```
proc glimmix;
   class subject drug time;
   model y = drug time drug*time;
   random _residual_ / sub=subject type=unr;
   covtest 'Homogeneous correlation'
        general  0 0 0 0 1 -1            ,
                 0 0 0 0 1  0 -1         ,
                 0 0 0 0 1  0  0 -1      ,
                 0 0 0 0 1  0  0  0 -1   ,
                 0 0 0 0 1  0  0  0  0 -1;
run;
```

Notice that the variances (the first four covariance parameters) are allowed to vary. The null model for this test is thus a heterogeneous compound symmetry model.

The degrees of freedom associated with these general linear hypotheses are determined as the rank of the matrix $\mathbf{LL}'$, where $\mathbf{L}$ is the $k \times q$ matrix of coefficients and q is the number of covariance parameters. Notice that the coefficients in a row do not have to sum to zero. The following statement tests $H: \theta_1 = 3\theta_2,\ \theta_3 = 0$:

```
covtest general 1 -3, 0 0 1;
```

Covariance Test Options

You can specify the following options in the COVTEST statement after a slash (/).

CL<(*suboptions*)>
requests confidence limits or bounds for the covariance parameter estimates. These limits are displayed as extra columns in the "Covariance Parameter Estimates" table.

The following suboptions determine the computation of confidence bounds and intervals. See the section "Statistical Inference for Covariance Parameters" on page 2234 for details about constructing likelihood ratio confidence limits for covariance parameters with PROC GLIMMIX.

ALPHA=*number*
determines the confidence level for constructing confidence limits for the covariance parameters. The value of *number* must be between 0 and 1, the default is 0.05, and the confidence level is $1 - number$.

LOWERBOUND

LOWER

requests lower confidence bounds.

TYPE=*method*

determines how the GLIMMIX procedure constructs confidence limits for covariance parameters. The valid methods are PLR (or PROFILE), ELR (or ESTIMATED), and WALD.

TYPE=PLR (TYPE=PROFILE) requests confidence bounds by inversion of the profile (restricted) likelihood ratio (PLR). If θ is the parameter of interest, L denotes the likelihood (possibly restricted and possibly a pseudo-likelihood), and $\boldsymbol{\theta}_2$ is the vector of the remaining (nuisance) parameters, then the profile likelihood is defined as

$$L(\boldsymbol{\theta}_2|\widetilde{\theta}) = \sup_{\boldsymbol{\theta}_2} L(\widetilde{\theta}, \boldsymbol{\theta}_2)$$

for a given value $\widetilde{\theta}$ of θ. If $L(\widehat{\boldsymbol{\theta}})$ is the overall likelihood evaluated at the estimates $\widehat{\boldsymbol{\theta}}$, the $(1-\alpha) \times 100\%$ confidence region for θ satisfies the inequality

$$2\left\{L(\widehat{\boldsymbol{\theta}}) - L(\boldsymbol{\theta}_2|\widetilde{\theta})\right\} \leq \chi^2_{1,(1-\alpha)}$$

where $\chi^2_{1,(1-\alpha)}$ is the cutoff from a chi-square distribution with one degree of freedom and α probability to its right. If a residual scale parameter ϕ is profiled from the estimation, and θ is expressed in terms of a ratio with ϕ during estimation, then profile likelihood confidence limits are constructed for the ratio of the parameter with the residual variance. A column showing the ratio estimates is added to the "Covariance Parameter Estimates" table in this case. To obtain profile likelihood ratio limits for the parameters, rather than their ratios, and for the residual variance, use the NOPROFILE option in the PROC GLIMMIX statement. Also note that METHOD=LAPLACE or METHOD=QUAD implies the NOPROFILE option.

The TYPE=ELR (TYPE=ESTIMATED) option constructs bounds from the estimated likelihood (Pawitan 2001), where nuisance parameters are held fixed at the (restricted) maximum (pseudo-) likelihood estimates of the model. Estimated likelihood intervals are computationally less demanding than profile likelihood intervals, but they do not take into account the variability of the nuisance parameters or the dependence among the covariance parameters. See the section "Statistical Inference for Covariance Parameters" on page 2234 for a geometric interpretation and comparison of ELR versus PLR confidence bounds. A $(1-\alpha) \times 100\%$ confidence region based on the estimated likelihood is defined by the inequality

$$2\left\{L(\widehat{\boldsymbol{\theta}}) - L(\widetilde{\theta}, \widehat{\boldsymbol{\theta}}_2)\right\} \leq \chi^2_{1,(1-\alpha)}$$

where $L(\widetilde{\theta}, \widehat{\boldsymbol{\theta}}_2)$ is the likelihood evaluated at $\widetilde{\theta}$ and the component of $\widehat{\boldsymbol{\theta}}$ that corresponds to $\boldsymbol{\theta}_2$. Estimated likelihood ratio intervals tend to perform well when the correlations between the parameter of interest and the nuisance parameters is small. Their coverage probabilities can fall short of the nominal coverage otherwise. You can display the correlation matrix of the covariance parameter estimates with the ASYCORR option in the PROC GLIMMIX statement.

If you choose TYPE=PLR or TYPE=ELR, the GLIMMIX procedure reports the right-tail probability of the associated single-degree-of-freedom likelihood ratio test along with the confidence bounds. This helps you diagnose whether solutions to the inequality could be found. If the reported probability exceeds α, the associated bound does not meet the inequality. This might occur, for example, when the parameter space is bounded and the likelihood at the boundary values has not dropped by a sufficient amount to satisfy the test inequality.

The TYPE=WALD method requests confidence limits based on the Wald-type statistic $Z_\theta = \widehat{\theta}/\text{ease}(\widehat{\theta})$, where ease is the estimated asymptotic standard error of the covariance parameter. For parameters that have a lower boundary constraint of zero, a Satterthwaite approximation is used to construct limits of the form

$$\frac{\nu\widehat{\theta}}{\chi^2_{\nu,1-\alpha/2}} \leq \theta \leq \frac{\nu\widehat{\theta}}{\chi^2_{\nu,\alpha/2}}$$

where $\nu = 2Z^2$, and the denominators are quantiles of the χ^2 distribution with ν degrees of freedom. Refer to Milliken and Johnson (1992) and Burdick and Graybill (1992) for similar techniques. For all other parameters, Wald Z-scores and normal quantiles are used to construct the limits. Such limits are also provided for variance components if you specify the NOBOUND option in the PROC GLIMMIX statement or the PARMS statement.

UPPERBOUND

UPPER

requests upper confidence bounds.

If you do not specify any suboptions, the default is to compute two-sided Wald confidence intervals with confidence level $1 - \alpha = 0.95$.

CLASSICAL

requests that the p-value of the likelihood ratio test be computed by the classical method. If $\widehat{\lambda}$ is the realized value of the test statistic in the likelihood ratio test,

$$p = \Pr\left(\chi^2_\nu \geq \widehat{\lambda}\right)$$

where ν is the degrees of freedom of the hypothesis.

DF=*value-list*

enables you to supply degrees of freedom $\nu_1, \cdots, \nu_k$ for the computation of p-values from chi-square mixtures. The mixture weights $w_1, \cdots, w_k$ are supplied with the WGHT= option. If no weights are specified, an equal weight distribution is assumed. If $\widehat{\lambda}$ is the realized value of the test statistic in the likelihood ratio test, PROC GLIMMIX computes the p-value as (Shapiro 1988)

$$p = \sum_{i=1}^{k} w_i \Pr\left(\chi^2_{\nu_i} \geq \widehat{\lambda}\right)$$

Note that $\chi^2_0 \equiv 0$ and that mixture weights are scaled to sum to one. If you specify more weights than degrees of freedom in *value-list*, the rank of the hypothesis (DF column) is substituted for the missing degrees of freedom.

Specifying a single value ν for *value-list* without giving mixture weights is equivalent to computing the p-value as

$$p = \Pr\left(\chi^2_\nu \geq \widehat{\lambda}\right)$$

For example, the following statements compute the p-value based on a chi-square distribution with one degree of freedom:

```
proc glimmix noprofile;
   class A sub;
   model score = A;
   random _residual_ / type=ar(1) subject=sub;
   covtest 'ELR low' 30.62555 0.7133361 / df=1;
run;
```

The DF column of the COVTEST output will continue to read 2 regardless of the DF= specification, however, because the DF column reflects the rank of the hypothesis and equals the number of constraints imposed on the full model.

ESTIMATES

EST

displays the estimates of the covariance parameters under the null hypothesis. Specifying the ESTIMATES option in one COVTEST statement has the same effect as specifying the option in every COVTEST statement.

MAXITER=*number*

limits the number of iterations when you are refitting the model under the null hypothesis to *number* iterations. If the null model does not converge before the limit is reached, no p-values are produced.

PARMS

displays the values of the covariance parameters under the null hypothesis. This option is useful if you supply multiple sets of parameter values with the TESTDATA= option. Specifying the PARMS option in one COVTEST statement has the same effect as specifying the option in every COVTEST statement.

RESTART

specifies that starting values for the covariance parameters for the null model are obtained by the same mechanism as starting values for the full models. For example, if you do not specify a PARMS statement, the RESTART option computes MIVQUE(0) estimates under the null model (Goodnight 1978b). If you provide starting values with the PARMS statement, the starting values for the null model are obtained by applying restrictions to the starting values for the full model.

By default, PROC GLIMMIX obtains starting values by applying null model restrictions to the converged estimates of the full model. Although this is computationally expedient, the method does not always lead to good starting values for the null model, depending on the nature of the model and hypothesis. In particular, when you receive a warning about parameters not specified under H_0 falling on the boundary, the RESTART option can be useful.

TOLERANCE=*r*

Values within tolerance $r \geq 0$ of the boundary of the parameter space are considered on the boundary when PROC GLIMMIX examines estimates of nuisance parameters under H_0 and determines whether mixture weights and degrees of freedom can be obtained. In certain cases, when parameters not specified under the null hypothesis are on boundaries, the asymptotic distribution of the likelihood ratio statistic is not a mixture of chi-squares (see, for example, case 8 in Self and Liang 1987). The default for r is 1E4 times the machine epsilon; this product is approximately 1E−12 on most computers.

WALD

produces Wald Z tests for the covariance parameters based on the estimates and asymptotic standard errors in the "Covariance Parameter Estimates" table.

WGHT=*value-list*

enables you to supply weights for the computation of p-values from chi-square mixtures. See the DF= option for details. Mixture weights are scaled to sum to one.

EFFECT Statement (Experimental)

EFFECT *effect-specification* **;**

The experimental EFFECT statement enables you to construct special collections of columns for **X** or **Z** matrices in your model. These collections are referred to as *constructed effects* to distinguish them from the usual model effects formed from continuous or classification variables.

For details about the syntax of the EFFECT statement and how columns of constructed effects are computed, see the section "Constructed Effects and the EFFECT Statement (Experimental)" on page 377 of Chapter 18, "Shared Concepts and Topics." For specific details concerning the use of the EFFECT statement with the GLIMMIX procedure, see the section "Notes on the EFFECT Statement" on page 2262.

ESTIMATE Statement

ESTIMATE *'label' contrast-specification* < **(***divisor***=***n***)** >
< , *'label' contrast-specification* < **(***divisor***=***n***)** > > < , . . . >
< / *options* > **;**

The ESTIMATE statement provides a mechanism for obtaining custom hypothesis tests. As in the CONTRAST statement, the basic element of the ESTIMATE statement is the *contrast-specification*, which consists of MODEL and G-side random effects and their coefficients. Specifically, a *contrast-specification* takes the form

< *fixed-effect values* . . . > < | *random-effect values* . . . >

Based on the *contrast-specifications* in your ESTIMATE statement, PROC GLIMMIX constructs the matrix $\mathbf{L}' = [\mathbf{K}' \; \mathbf{M}']$, as in the CONTRAST statement, where $\mathbf{K}$ is associated with the fixed effects and $\mathbf{M}$ is associated with the G-side random effects. The GLIMMIX procedure supports nonpositional syntax for the coefficients of fixed effects in the ESTIMATE statement. For details see the section "Positional and Nonpositional Syntax for Contrast Coefficients" on page 2264.

PROC GLIMMIX then produces for each row $\mathbf{l}$ of $\mathbf{L}'$ an approximate t test of the hypothesis $H: \mathbf{l}\boldsymbol{\phi} = 0$, where $\boldsymbol{\phi} = [\boldsymbol{\beta}' \; \boldsymbol{\gamma}']'$. You can also obtain multiplicity-adjusted p-values and confidence limits for multirow estimates with the ADJUST= option. The output from multiple ESTIMATE statements is organized as follows. Results from unadjusted estimates are reported first in a single table, followed by separate tables for each of the adjusted estimates. Results from all ESTIMATE statements are combined in the "Estimates" ODS table.

Note that multirow estimates are permitted. Unlike the CONTRAST statement, you need to specify a *'label'* for every row of the multirow estimate, because PROC GLIMMIX produces one test per row.

PROC GLIMMIX selects the degrees of freedom to match those displayed in the "Type III Tests of Fixed Effects" table for the final effect you list in the ESTIMATE statement. You can modify the degrees of freedom by using the DF= option. If you select DDFM=NONE and do not modify the degrees of freedom by using the DF= option, PROC GLIMMIX uses infinite degrees of freedom, essentially computing approximate z tests. If PROC GLIMMIX finds the fixed-effects portion of the specified estimate to be nonestimable, then it displays "Non-est" for the estimate entry.

ADJDFE=SOURCE

ADJDFE=ROW

specifies how denominator degrees of freedom are determined when p-values and confidence limits are adjusted for multiple comparisons with the ADJUST= option. When you do not specify the ADJDFE= option, or when you specify ADJDFE=SOURCE, the denominator degrees of freedom for multiplicity-adjusted results are the denominator degrees of freedom for the final effect listed in the ESTIMATE statement from the "Type III Tests of Fixed Effects" table.

The ADJDFE=ROW setting is useful if you want multiplicity adjustments to take into account that denominator degrees of freedom are not constant across estimates. This can be the case, for example, when the DDFM=SATTERTHWAITE or DDFM=KENWARDROGER degrees-of-freedom method is in effect.

ADJUST=BON

ADJUST=SCHEFFE

ADJUST=SIDAK

ADJUST=SIMULATE< (*simoptions*) >

ADJUST=T

requests a multiple comparison adjustment for the p-values and confidence limits for the estimates. The adjusted quantities are produced in addition to the unadjusted quantities. Adjusted confidence limits are produced if the CL or ALPHA= option is in effect. For a description of the adjustments, see Chapter 39, "The GLM Procedure," and Chapter 58, "The MULTTEST Procedure," of the *SAS/STAT User's Guide* and the documentation for the ADJUST= option in the LSMEANS statement. The ADJUST= option is ignored for generalized logit models.

If the STEPDOWN option is in effect, the *p*-values are further adjusted in a step-down fashion.

ALPHA=*number*

requests that a *t*-type confidence interval be constructed with confidence level $1 - number$. The value of *number* must be between 0 and 1; the default is 0.05. If DDFM=NONE and you do not specify degrees of freedom with the DF= option, PROC GLIMMIX uses infinite degrees of freedom, essentially computing a *z* interval.

BYCATEGORY

BYCAT

requests that in models for nominal data (generalized logit models) estimates be reported separately for each category. In contrast to the BYCATEGORY option in the CONTRAST statement, an ESTIMATE statement in a generalized logit model does not distribute coefficients by response category, because ESTIMATE statements always correspond to single rows of the **L** matrix.

For example, assume that the response variable Style is multinomial with three (unordered) categories. The following GLIMMIX statements fit a generalized logit model relating the preferred style of instruction to school and educational program effects:

```
proc glimmix data=school;
   class School Program;
   model Style(order=data) = School Program / s ddfm=none
                             dist=multinomial link=glogit;
   freq Count;
   estimate 'School 1 vs. 2' school 1 -1 / bycat;
   estimate 'School 1 vs. 2' school 1 -1;
run;
```

The first ESTIMATE statement compares school effects separately for each nonredundant category. The second ESTIMATE statement compares the school effects for the first nonreference category.

The BYCATEGORY option has no effect unless your model is a generalized (mixed) logit model.

CL

requests that *t*-type confidence limits be constructed. If DDFM=NONE and you do not specify degrees of freedom with the DF= option, PROC GLIMMIX uses infinite degrees of freedom, essentially computing a *z* interval. The confidence level is 0.95 by default. These intervals are adjusted for multiplicity when you specify the ADJUST= option.

DF=*number*

specifies the degrees of freedom for the *t* test and confidence limits. The default is the denominator degrees of freedom taken from the "Type III Tests of Fixed Effects" table and corresponds to the final effect you list in the ESTIMATE statement.

DIVISOR=*value-list*

specifies a list of values by which to divide the coefficients so that fractional coefficients can be entered as integer numerators. If you do not specify *value-list*, a default value of 1.0 is assumed. Missing values in the *value-list* are converted to 1.0.

If the number of elements in *value-list* exceeds the number of rows of the estimate, the extra values are ignored. If the number of elements in *value-list* is less than the number of rows of the estimate, the last value in *value-list* is copied forward.

If you specify a row-specific divisor as part of the specification of the estimate row, this value multiplies the corresponding divisor implied by the *value-list*. For example, the following statement divides the coefficients in the first row by 8, and the coefficients in the third and fourth row by 3:

```
estimate 'One vs. two'   A 2 -2  (divisor=2),
         'One vs. three' A 1  0 -1         ,
         'One vs. four'  A 3  0  0 -3      ,
         'One vs. five'  A 1  0  0  0  -1  / divisor=4,.,3;
```

Coefficients in the second row are not altered.

E

requests that the **L** matrix coefficients be displayed.

EXP

requests exponentiation of the estimate. When you model data with the logit, cumulative logit, or generalized logit link functions, and the estimate represents a log odds ratio or log cumulative odds ratio, the EXP option produces an odds ratio. See "Odds and Odds Ratio Estimation" on page 2256 for important details about the computation and interpretation of odds and odds ratio results with the GLIMMIX procedure. If you specify the CL or ALPHA= option, the (adjusted) confidence bounds are also exponentiated.

GROUP *coeffs*

sets up random-effect contrasts between different groups when a GROUP= variable appears in the RANDOM statement. By default, ESTIMATE statement coefficients on random effects are distributed equally across groups. If you enter a multirow estimate, you can also enter multiple rows for the GROUP coefficients. If the number of GROUP coefficients is less than the number of contrasts in the ESTIMATE statement, the GLIMMIX procedure cycles through the GROUP coefficients. For example, the following two statements are equivalent:

```
estimate 'Trt 1 vs 2 @ x=0.4' trt 1 -1  0 | x 0.4,
         'Trt 1 vs 3 @ x=0.4' trt 1  0 -1 | x 0.4,
         'Trt 1 vs 2 @ x=0.5' trt 1 -1  0 | x 0.5,
         'Trt 1 vs 3 @ x=0.5' trt 1  0 -1 | x 0.5 /
                group 1 -1, 1 0 -1, 1 -1, 1 0 -1;

estimate 'Trt 1 vs 2 @ x=0.4' trt 1 -1  0 | x 0.4,
         'Trt 1 vs 3 @ x=0.4' trt 1  0 -1 | x 0.4,
         'Trt 1 vs 2 @ x=0.5' trt 1 -1  0 | x 0.5,
         'Trt 1 vs 3 @ x=0.5' trt 1  0 -1 | x 0.5 /
                group 1 -1, 1 0 -1;
```

ILINK

requests that the estimate and its standard error are also reported on the scale of the mean (the inverse linked scale). PROC GLIMMIX computes the value on the mean scale by applying the inverse link to the estimate. The interpretation of this quantity depends on the *fixed-effect values* and *random-effect values* specified in your ESTIMATE statement and on the link function. In a model for binary data with logit link, for example, the following statements

compute

$$\frac{1}{1+\exp\{-(\alpha_1-\alpha_2)\}}$$

where α_1 and α_2 are the fixed-effects solutions associated with the first two levels of the classification effect A:

```
proc glimmix;
   class A;
   model y = A / dist=binary link=logit;
   estimate 'A one vs. two' A 1 -1 / ilink;
run;
```

This quantity is not the difference of the probabilities associated with the two levels,

$$\pi_1-\pi_2 = \frac{1}{1+\exp\{-\beta_0-\alpha_1\}} - \frac{1}{1+\exp\{-\beta_0-\alpha_2\}}$$

The standard error of the inversely linked estimate is based on the delta method. If you also specify the CL option, the GLIMMIX procedure computes confidence limits for the estimate on the mean scale. In multinomial models for nominal data, the limits are obtained by the delta method. In other models they are obtained from the inverse link transformation of the confidence limits for the estimate. The ILINK option is specific to an ESTIMATE statement.

LOWER

LOWERTAILED

requests that the *p*-value for the *t* test be based only on values less than the test statistic. A two-tailed test is the default. A lower-tailed confidence limit is also produced if you specify the CL or ALPHA= option.

Note that for ADJUST=SCHEFFE the one-sided adjusted confidence intervals and one-sided adjusted *p*-values are the same as the corresponding two-sided statistics, because this adjustment is based on only the right tail of the F distribution.

SINGULAR=*number*

tunes the estimability checking as documented for the CONTRAST statement..

STEPDOWN<(*step-down-options***)>**

requests that multiplicity adjustments for the *p*-values of estimates be further adjusted in a step-down fashion. Step-down methods increase the power of multiple testing procedures by taking advantage of the fact that a *p*-value will never be declared significant unless all smaller *p*-values are also declared significant. Note that the STEPDOWN adjustment combined with ADJUST=BON corresponds to the methods of Holm (1979) and Shaffer's "Method 2" (1986); this is the default. Using step-down-adjusted *p*-values combined with ADJUST=SIMULATE corresponds to the method of Westfall (1997).

If the degrees-of-freedom method is DDFM=KENWARDROGER or DDFM=SATTERTHWAITE, then step-down-adjusted *p*-values are produced only if the ADJDFE=ROW option is in effect.

Also, the STEPDOWN option affects only *p*-values, not confidence limits. For ADJUST=SIMULATE, the generalized least squares hybrid approach of Westfall (1997) is employed to increase Monte Carlo accuracy.

You can specify the following *step-down-options* in parentheses after the STEPDOWN option.

MAXTIME=*n*

specifies the time (in seconds) to spend computing the maximal logically consistent sequential subsets of equality hypotheses for TYPE=LOGICAL. The default is MAXTIME=60. If the MAXTIME value is exceeded, the adjusted tests are not computed. When this occurs, you can try increasing the MAXTIME value. However, note that there are common multiple comparisons problems for which this computation requires a huge amount of time—for example, all pairwise comparisons between more than 10 groups. In such cases, try to use TYPE=FREE (the default) or TYPE=LOGICAL(n) for small n.

ORDER=PVALUE

ORDER=ROWS

specifies the order in which the step-down tests are performed. ORDER=PVALUE is the default, with estimates being declared significant only if all estimates with smaller (unadjusted) p-values are significant. If you specify ORDER=ROWS, then significances are evaluated in the order in which they are specified in the syntax.

REPORT

specifies that a report on the step-down adjustment be displayed, including a listing of the sequential subsets (Westfall 1997) and, for ADJUST=SIMULATE, the step-down simulation results.

TYPE=LOGICAL<(*n***)>**

TYPE=FREE

If you specify TYPE=LOGICAL, the step-down adjustments are computed by using maximal logically consistent sequential subsets of equality hypotheses (Shaffer 1986, Westfall 1997). Alternatively, for TYPE=FREE, sequential subsets are computed ignoring logical constraints. The TYPE=FREE results are more conservative than those for TYPE=LOGICAL, but they can be much more efficient to produce for many estimates. For example, it is not feasible to take logical constraints between all pairwise comparisons of more than about 10 groups. For this reason, TYPE=FREE is the default.

However, you can reduce the computational complexity of taking logical constraints into account by limiting the depth of the search tree used to compute them, specifying the optional depth parameter as a number n in parentheses after TYPE=LOGICAL. As with TYPE=FREE, results for TYPE=LOGICAL(n) are conservative relative to the true TYPE=LOGICAL results, but even for TYPE=LOGICAL(0) they can be appreciably less conservative than TYPE=FREE and they are computationally feasible for much larger numbers of estimates. If you do not specify n or if $n = -1$, the full search tree is used.

SUBJECT *coeffs*

sets up random-effect contrasts between different subjects when a SUBJECT= variable appears in the RANDOM statement. By default, ESTIMATE statement coefficients on random effects are distributed equally across subjects. Listing subject coefficients for an ESTIMATE statement with multiple rows follows the same rules as for GROUP coefficients.

UPPER

UPPERTAILED

requests that the p-value for the t test be based only on values greater than the test statistic. A two-tailed test is the default. An upper-tailed confidence limit is also produced if you specify the CL or ALPHA= option.

Note that for ADJUST=SCHEFFE the one-sided adjusted confidence intervals and one-sided adjusted p-values are the same as the corresponding two-sided statistics, because this adjustment is based on only the right tail of the F distribution.

FREQ Statement

FREQ *variable* **;**

The *variable* in the FREQ statement identifies a numeric variable in the data set or one computed through PROC GLIMMIX programming statements that contains the frequency of occurrence for each observation. PROC GLIMMIX treats each observation as if it appears f times, where f is the value of the FREQ variable for the observation. If it is not an integer, the frequency value is truncated to an integer. If the frequency value is less than 1 or missing, the observation is not used in the analysis. When the FREQ statement is not specified, each observation is assigned a frequency of 1.

The analysis produced by using a FREQ statement reflects the expanded number of observations. For an example of a FREQ statement in a model with random effects, see Example 38.11 in this chapter.

ID Statement

ID *variables* **;**

The ID statement specifies which quantities to include in the OUT= data set from the OUTPUT statement in addition to any statistics requested in the OUTPUT statement. If no ID statement is given, the GLIMMIX procedure includes all variables from the input data set in the OUT= data set. Otherwise, only the variables listed in the ID statement are included. Automatic variables such as _LINP_, _MU_, _VARIANCE_, etc. are not transferred to the OUT= data set unless they are listed in the ID statement.

The ID statement can be used to transfer computed quantities that depend on the model to an output data set. In the following example, two sets of Hessian weights are computed in a gamma regression with a noncanonical link. The covariance matrix for the fixed effects can be constructed as the inverse of $\mathbf{X}'\mathbf{W}\mathbf{X}$. $\mathbf{W}$ is a diagonal matrix of the w_{ei} or w_{oi}, depending on whether the expected or observed Hessian matrix is desired, respectively.

```
proc glimmix;
  class group age;
  model cost = group age / s error=gamma link=pow(0.5);
  output out=gmxout pred=pred;
  id _variance_ wei woi;
  vpmu  = 2*_mu_;
  if (_mu_ > 1.0e-8) then do;
     gpmu  =  0.5  * (_mu_**(-0.5));
     gppmu = -0.25 * (_mu_**(-1.5));
     wei   = 1/(_phi_*_variance_*gpmu*gpmu);
     woi   = wei + (cost-_mu_) *
             (_variance_*gppmu + vpmu*gpmu) /
             (_variance_*_variance_*gpmu*gpmu*gpmu*_phi_);
  end;
run;
```

The variables _VARIANCE_ and _MU_ and other symbols are predefined by PROC GLIMMIX and can be used in programming statements. For rules and restrictions, see the section "Programming Statements" on page 2207.

LSMEANS Statement

LSMEANS *fixed-effects* < / *options* > **;**

The LSMEANS statement computes least squares means (LS-means) of fixed effects. As in the GLM and the MIXED procedures, LS-means are *predicted population margins*—that is, they estimate the marginal means over a balanced population. In a sense, LS-means are to unbalanced designs as class and subclass arithmetic means are to balanced designs. The **L** matrix constructed to compute them is the same as the **L** matrix formed in PROC GLM; however, the standard errors are adjusted for the covariance parameters in the model. Least squares means computations are not supported for multinomial models.

Each LS-mean is computed as $\mathbf{L}\widehat{\boldsymbol{\beta}}$, where $\mathbf{L}$ is the coefficient matrix associated with the least squares mean and $\widehat{\boldsymbol{\beta}}$ is the estimate of the fixed-effects parameter vector. The approximate standard error for the LS-mean is computed as the square root of $\mathbf{L}\widehat{\mathrm{Var}}[\widehat{\boldsymbol{\beta}}]\mathbf{L}'$. The approximate variance matrix of the fixed-effects estimates depends on the estimation method.

LS-means are constructed on the linked scale—that is, the scale on which the model effects are additive. For example, in a binomial model with logit link, the least squares means are predicted population margins of the logits.

LS-means can be computed for any effect in the MODEL statement that involves only CLASS variables. You can specify multiple effects in one LSMEANS statement or in multiple LSMEANS statements, and all LSMEANS statements must appear after the MODEL statement. As in the ESTIMATE statement, the **L** matrix is tested for estimability, and if this test fails, PROC GLIMMIX displays "Non-est" for the LS-means entries.

Assuming the LS-mean is estimable, PROC GLIMMIX constructs an approximate *t* test to test the

null hypothesis that the associated population quantity equals zero. By default, the denominator degrees of freedom for this test are the same as those displayed for the effect in the "Type III Tests of Fixed Effects" table. If the DDFM=SATTERTHWAITE or DDFM=KENWARDROGER option is specified in the MODEL statement, PROC GLIMMIX determines degrees of freedom separately for each test, unless the DDF= option overrides it for a particular effect. See the DDFM= option for more information.

Table 38.5 summarizes important options in the LSMEANS statement. All LSMEANS options are subsequently discussed in alphabetical order.

Table 38.5 Summary of Important LSMEANS Statement Options

Option	Description
Construction and Computation of LS-Means	
AT	modifies covariate value in computing LS-means
BYLEVEL	computes separate margins
DIFF	requests differences of LS-means
OM	specifies weighting scheme for LS-mean computation as determined by the input data set
SINGULAR=	tunes estimability checking
SLICE=	partitions F tests (simple effects)
SLICEDIFF=	requests simple effects differences
SLICEDIFFTYPE	determines the type of simple difference
Degrees of Freedom and *P*-values	
ADJDFE=	determines whether to compute row-wise denominator degrees of freedom with DDFM=SATTERTHWAITE or DDFM=KENWARDROGER
ADJUST=	determines the method for multiple comparison adjustment of LS-mean differences
ALPHA=α	determines the confidence level $(1-\alpha)$
DF=	assigns specific value to degrees of freedom for tests and confidence limits
STEPDOWN	adjusts multiple comparison p-values further in a step-down fashion
Statistical Output	
CL	constructs confidence limits for means and or mean differences
CORR	displays correlation matrix of LS-means
COV	displays covariance matrix of LS-means
E	prints the $\mathbf{L}$ matrix
ILINK	computes and displays estimates and standard errors of LS-means (not differences) on the inverse linked scale
LINES	produces "Lines" display for pairwise LS-mean differences
ODDS	reports odds of levels of fixed effects if permissible by the link function
ODDSRATIO	reports (simple) differences of least squares means in terms of odds ratios if permissible by the link function
PLOTS=	requests ODS statistical graphics of means and mean comparisons

You can specify the following options in the LSMEANS statement after a slash (/).

ADJDFE=ROW

ADJDFE=SOURCE

specifies how denominator degrees of freedom are determined when *p*-values and confidence limits are adjusted for multiple comparisons with the ADJUST= option. When you do not specify the ADJDFE= option, or when you specify ADJDFE=SOURCE, the denominator degrees of freedom for multiplicity-adjusted results are the denominator degrees of freedom for the LS-mean effect in the "Type III Tests of Fixed Effects" table. When you specify ADJDFE=ROW, the denominator degrees of freedom for multiplicity-adjusted results correspond to the degrees of freedom displayed in the DF column of the "Differences of Least Squares Means" table.

The ADJDFE=ROW setting is particularly useful if you want multiplicity adjustments to take into account that denominator degrees of freedom are not constant across LS-mean differences. This can be the case, for example, when the DDFM=SATTERTHWAITE or DDFM=KENWARDROGER degrees-of-freedom method is in effect.

In one-way models with heterogeneous variance, combining certain ADJUST= options with the ADJDFE=ROW option corresponds to particular methods of performing multiplicity adjustments in the presence of heteroscedasticity. For example, the following statements fit a heteroscedastic one-way model and perform Dunnett's T3 method (Dunnett 1980), which is based on the studentized maximum modulus (ADJUST=SMM):

```
proc glimmix;
   class A;
   model y = A / ddfm=satterth;
   random _residual_ / group=A;
   lsmeans A / adjust=smm adjdfe=row;
run;
```

If you combine the ADJDFE=ROW option with ADJUST=SIDAK, the multiplicity adjustment corresponds to the T2 method of Tamhane (1979), while ADJUST=TUKEY corresponds to the method of Games-Howell (Games and Howell 1976). Note that ADJUST=TUKEY gives the exact results for the case of fractional degrees of freedom in the one-way model, but it does not take into account that the degrees of freedom are subject to variability. A more conservative method, such as ADJUST=SMM, might protect the overall error rate better.

Unless the ADJUST= option is specified in the LSMEANS statement, the ADJDFE= option has no effect.

ADJUST=BON
ADJUST=DUNNETT
ADJUST=NELSON
ADJUST=SCHEFFE
ADJUST=SIDAK
ADJUST=SIMULATE< (*simoptions***) >**
ADJUST=SMM | GT2
ADJUST=TUKEY

requests a multiple comparison adjustment for the p-values and confidence limits for the differences of LS-means. The adjusted quantities are produced in addition to the unadjusted quantities. By default, PROC GLIMMIX performs all pairwise differences. If you specify ADJUST=DUNNETT, the procedure analyzes all differences with a control level. If you specify ADJUST=NELSON, ANOM differences are taken. The ADJUST= option implies the DIFF option, unless the SLICEDIFF= option is specified.

The BON (Bonferroni) and SIDAK adjustments involve correction factors described in Chapter 39, "The GLM Procedure," and Chapter 58, "The MULTTEST Procedure," of the *SAS/STAT User's Guide*; also see Westfall and Young (1993) and Westfall et al. (1999). When you specify ADJUST=TUKEY and your data are unbalanced, PROC GLIMMIX uses the approximation described in Kramer (1956) and identifies the adjustment as "Tukey-Kramer" in the results. Similarly, when you specify ADJUST=DUNNETT or ADJUST=NELSON and the LS-means are correlated, the GLIMMIX procedure uses the factor-analytic covariance approximation described in Hsu (1992) and identifies the adjustment in the results as "Dunnett-Hsu" or "Nelson-Hsu," respectively. The approximation derives an approximate "effective sample sizes" for which exact critical values are computed. Note that computing the exact adjusted p-values and critical values for unbalanced designs can be computationally intensive, in particular for ADJUST=NELSON. A simulation-based approach, as specified by the ADJUST=SIM option, while nondeterministic, can provide inferences that are sufficiently accurate in much less time. The preceding references also describe the SCHEFFE and SMM adjustments.

Nelson's adjustment applies only to the analysis of means (Ott 1967; Nelson 1982, 1991, 1993), where LS-means are compared against an average LS-mean. It does not apply to all pairwise differences of least squares means, or to slice differences that you specify with the SLICEDIFF= option. See the DIFF=ANOM option for more details regarding the analysis of means with the GLIMMIX procedure.

The SIMULATE adjustment computes adjusted p-values and confidence limits from the simulated distribution of the maximum or maximum absolute value of a multivariate t random vector. All covariance parameters, except the residual scale parameter, are fixed at their estimated values throughout the simulation, potentially resulting in some underdispersion. The simulation estimates q, the true $(1-\alpha)$th quantile, where $1-\alpha$ is the confidence coefficient. The default α is 0.05, and you can change this value with the ALPHA= option in the LSMEANS statement.

The number of samples is set so that the tail area for the simulated q is within γ of $1-\alpha$ with $100(1-\epsilon)\%$ confidence. In equation form,

$$\Pr(|F(\widehat{q}) - (1-\alpha)| \leq \gamma) = 1-\epsilon$$

where $\hat{q}$ is the simulated q and F is the true distribution function of the maximum; see Edwards and Berry (1987) for details. By default, $\gamma = 0.005$ and $\epsilon = 0.01$, placing the tail area of $\hat{q}$ within 0.005 of 0.95 with 99% confidence. The ACC= and EPS= *simoptions* reset γ and ϵ, respectively, the NSAMP= *simoption* sets the sample size directly, and the SEED= *simoption* specifies an integer used to start the pseudo-random number generator for the simulation. If you do not specify a seed, or if you specify a value less than or equal to zero, the seed is generated from reading the time of day from the computer clock. For additional descriptions of these and other simulation options, see the section "LSMEANS Statement" on page 2458 in Chapter 39, "The GLM Procedure."

If the STEPDOWN option is in effect, the p-values are further adjusted in a step-down fashion. For certain options and data, this adjustment is exact under an *iid* $N(0, \sigma^2)$ model for the dependent variable, in particular for the following:

- for ADJUST=DUNNETT when the means are uncorrelated
- for ADJUST=TUKEY with STEPDOWN(TYPE=LOGICAL) when the means are balanced and uncorrelated.

The first case is a consequence of the nature of the successive step-down hypotheses for comparisons with a control; the second employs an extension of the maximum studentized range distribution appropriate for partition hypotheses (Royen 1989). Finally, for STEPDOWN(TYPE=FREE), ADJUST=TUKEY employs the Royen (1989) extension in such a way that the resulting p-values are conservative.

ALPHA=*number*

requests that a t-type confidence interval be constructed for each of the LS-means with confidence level $1 - number$. The value of *number* must be between 0 and 1; the default is 0.05.

AT *variable*=*value*

AT (*variable-list***)=(***value-list***)**

AT MEANS

enables you to modify the values of the covariates used in computing LS-means. By default, all covariate effects are set equal to their mean values for computation of standard LS-means. The AT option enables you to assign arbitrary values to the covariates. Additional columns in the output table indicate the values of the covariates.

If there is an effect containing two or more covariates, the AT option sets the effect equal to the product of the individual means rather than the mean of the product (as with standard LS-means calculations). The AT MEANS option sets covariates equal to their mean values (as with standard LS-means) and incorporates this adjustment to crossproducts of covariates.

As an example, consider the following invocation of PROC GLIMMIX:

```
proc glimmix;
   class A;
   model Y = A x1 x2 x1*x2;
   lsmeans A;
   lsmeans A / at means;
   lsmeans A / at x1=1.2;
   lsmeans A / at (x1 x2)=(1.2 0.3);
run;
```

For the first two LSMEANS statements, the LS-means coefficient for x1 is $\overline{x}_1$ (the mean of x1) and for x2 is $\overline{x}_2$ (the mean of x2). However, for the first LSMEANS statement, the coefficient for x1*x2 is $\overline{x_1 x_2}$, but for the second LSMEANS statement, the coefficient is $\overline{x}_1 \times \overline{x}_2$. The third LSMEANS statement sets the coefficient for x1 equal to 1.2 and leaves it at $\overline{x}_2$ for x2, and the final LSMEANS statement sets these values to 1.2 and 0.3, respectively.

Even if you specify a WEIGHT variable, the unweighted covariate means are used for the covariate coefficients if there is no AT specification. If you specify the AT option, WEIGHT or FREQ variables are taken into account as follows. The weighted covariate means are then used for the covariate coefficients for which no explicit AT values are given, or if you specify AT MEANS. Observations that do not contribute to the analysis because of a missing dependent variable are included in computing the covariate means. You should use the E option in conjunction with the AT option to check that the modified LS-means coefficients are the ones you want.

The AT option is disabled if you specify the BYLEVEL option.

BYLEVEL

requests that separate margins be computed for each level of the LSMEANS effect.

The standard LS-means have equal coefficients across classification effects. The BYLEVEL option changes these coefficients to be proportional to the observed margins. This adjustment is reasonable when you want your inferences to apply to a population that is not necessarily balanced but has the margins observed in the input data set. In this case, the resulting LS-means are actually equal to raw means for fixed-effects models and certain balanced random-effects models, but their estimated standard errors account for the covariance structure that you have specified. If a WEIGHT statement is specified, PROC GLIMMIX uses weighted margins to construct the LS-means coefficients.

If the AT option is specified, the BYLEVEL option disables it.

CL

requests that *t*-type confidence limits be constructed for each of the LS-means. If DDFM=NONE, then PROC GLIMMIX uses infinite degrees of freedom for this test, essentially computing a *z* interval. The confidence level is 0.95 by default; this can be changed with the ALPHA= option. If you specify an ADJUST= option, then the confidence limits are adjusted for multiplicity, but if you also specify STEPDOWN, then only *p*-values are step-down adjusted, not the confidence limits.

CORR

displays the estimated correlation matrix of the least squares means as part of the "Least Squares Means" table.

COV

displays the estimated covariance matrix of the least squares means as part of the "Least Squares Means" table.

DF=*number*

specifies the degrees of freedom for the *t* test and confidence limits. The default is the denominator degrees of freedom taken from the "Type III Tests of Fixed Effects" table corresponding to the LS-means effect.

DIFF<=*difftype*>
PDIFF<=*difftype*>

requests that differences of the LS-means be displayed. The optional *difftype* specifies which differences to produce, with possible values ALL, ANOM, CONTROL, CONTROLL, and CONTROLU. The ALL value requests all pairwise differences, and it is the default. The CONTROL *difftype* requests differences with a control, which, by default, is the first level of each of the specified LSMEANS effects.

The ANOM value requests differences between each LS-mean and the average LS-mean, as in the *analysis of means* (Ott 1967). The average is computed as a weighted mean of the LS-means, the weights being inversely proportional to the diagonal entries of the

$$\mathbf{L}\left(\mathbf{X}'\mathbf{X}\right)^{-}\mathbf{L}'$$

matrix. If LS-means are nonestimable, this design-based weighted mean is replaced with an equally weighted mean. Note that the ANOM procedure in SAS/QC software implements both tables and graphics for the analysis of means with a variety of response types. For one-way designs and normal data with identity link, the DIFF=ANOM computations are equivalent to the results of PROC ANOM. If the LS-means being compared are uncorrelated, exact adjusted *p*-values and critical values for confidence limits can be computed in the analysis of means; see Nelson (1982, 1991, 1993) and Guirguis and Tobias (2004) as well as the documentation for the ADJUST=NELSON option.

To specify which levels of the effects are the controls, list the quoted formatted values in parentheses after the CONTROL keyword. For example, if the effects A, B, and C are classification variables, each having two levels, 1 and 2, the following LSMEANS statement specifies the (1,2) level of A*B and the (2,1) level of B*C as controls:

```
lsmeans A*B B*C / diff=control('1' '2' '2' '1');
```

For multiple effects, the results depend upon the order of the list, and so you should check the output to make sure that the controls are correct.

Two-tailed tests and confidence limits are associated with the CONTROL *difftype*. For one-tailed results, use either the CONTROLL or CONTROLU *difftype*. The CONTROLL *difftype* tests whether the noncontrol levels are significantly smaller than the control; the upper confidence limits for the control minus the noncontrol levels are considered to be infinity and are displayed as missing. Conversely, the CONTROLU *difftype* tests whether the noncontrol levels are significantly larger than the control; the upper confidence limits for the noncontrol levels minus the control are considered to be infinity and are displayed as missing.

If you want to perform multiple comparison adjustments on the differences of LS-means, you must specify the ADJUST= option.

The differences of the LS-means are displayed in a table titled "Differences of Least Squares Means."

E

requests that the $\mathbf{L}$ matrix coefficients for the LSMEANS effects be displayed.

ILINK

requests that estimates and their standard errors in the "Least Squares Means" table also be

reported on the scale of the mean (the inverse linked scale). This enables you to obtain estimates of predicted probabilities and their standard errors in logistic models, for example. The option is specific to an LSMEANS statement. If you also specify the CL option, the GLIMMIX procedure computes confidence intervals for the predicted means by applying the inverse link transform to the confidence limits on the linked (linear) scale. Standard errors on the inverse linked scale are computed by the delta method.

LINES

presents results of comparisons between all pairs of least squares means by listing the means in descending order and indicating nonsignificant subsets by line segments beside the corresponding LS-means. When all differences have the same variance, these comparison lines are guaranteed to accurately reflect the inferences based on the corresponding tests, made by comparing the respective *p*-values to the value of the ALPHA= option (0.05 by default). However, equal variances might not be the case for differences between LS-means. If the variances are not all the same, then the comparison lines might be conservative, in the sense that if you base your inferences on the lines alone, you will detect fewer significant differences than the tests indicate. If there are any such differences, PROC GLIMMIX lists the pairs of means that are inferred to be significantly different by the tests but not by the comparison lines. Note, however, that in many cases, even though the variances are unequal, they are similar enough that the comparison lines accurately reflect the test inferences.

ODDS

requests that in models with logit, cumulative logit, and generalized logit link function the odds of the levels of the fixed effects are reported. If you specify the CL or ALPHA= option, confidence intervals for the odds are also computed. See the section "Odds and Odds Ratio Estimation" on page 2256 for further details about computation and interpretation of odds and odds ratios with the GLIMMIX procedure.

ODDSRATIO

OR

requests that LS-mean differences (DIFF, ADJUST= options) and simple effect comparisons (SLICEDIFF option) are also reported in terms of odds ratios. The ODDSRATIO option is ignored unless you use either the logit, cumulative logit, or generalized logit link function. If you specify the CL or ALPHA= option, confidence intervals for the odds ratios are also computed. These intervals are adjusted for multiplicity when you specify the ADJUST= option. See the section "Odds and Odds Ratio Estimation" on page 2256 for further details about computation and interpretation of odds and odds ratios with the GLIMMIX procedure.

OBSMARGINS

OM

specifies a potentially different weighting scheme for the computation of LS-means coefficients. The standard LS-means have equal coefficients across classification effects; however, the OM option changes these coefficients to be proportional to those found in the input data set. This adjustment is reasonable when you want your inferences to apply to a population that is not necessarily balanced but has the margins observed in your data.

In computing the observed margins, PROC GLIMMIX uses all observations for which there are no missing or invalid independent variables, including those for which there are missing dependent variables. Also, if you use a WEIGHT statement, PROC GLIMMIX computes

weighted margins to construct the LS-means coefficients. If your data are balanced, the LS-means are unchanged by the OM option.

The BYLEVEL option modifies the observed-margins LS-means. Instead of computing the margins across all of the input data set, PROC GLIMMIX computes separate margins for each level of the LSMEANS effect in question. In this case the resulting LS-means are actually equal to raw means for fixed-effects models and certain balanced random-effects models, but their estimated standard errors account for the covariance structure that you have specified.

You can use the E option in conjunction with either the OM or BYLEVEL option to check that the modified LS-means coefficients are the ones you want. It is possible that the modified LS-means are not estimable when the standard ones are estimable, or vice versa.

PDIFF

is the same as the DIFF option. See the description of the DIFF option on page 2146.

PLOT | PLOTS<=*plot-request*<(*options*)>>

PLOT | PLOTS<=(*plot-request*<(*options*)><...*plot-request*<(*options*)> >)>

requests that least squares means related graphics are produced via ODS Graphics, provided that the ODS GRAPHICS statement has been specified and the plot request does not conflict with other options in the LSMEANS statement. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For examples of the basic statistical graphics for least squares means and aspects of their computation and interpretation, see the section "Graphics for LS-Mean Comparisons" on page 2288 in this chapter.

The *options* for a specific plot request (and their suboptions) of the LSMEANS statement include those for the PLOTS= option in the PROC GLIMMIX statement. You can specify classification effects in the MEANPLOT request of the LSMEANS statement to control the display of interaction means with the PLOTBY= and SLICEBY= suboptions; these are not available in the PLOTS= option in the PROC GLIMMIX statement. Options specified in the LSMEANS statement override those in the PLOTS= option in the PROC GLIMMIX statement.

The available options and suboptions are as follows.

ALL

requests that the default plots corresponding to this LSMEANS statement be produced. The default plot depends on the options in the statement.

ANOMPLOT

ANOM

requests an analysis of means display in which least squares means are compared to an average least squares mean. Least squares mean ANOM plots are produced only for those model effects listed in LSMEANS statements that have options that do not contradict with the display. For example, the following statements produce analysis of mean plots for effects A and C:

```
lsmeans A / diff=anom plot=anom;
lsmeans B / diff      plot=anom;
lsmeans C /           plot=anom;
```

The DIFF option in the second LSMEANS statement implies all pairwise differences.

CONTROLPLOT

CONTROL

requests a display in which least squares means are visually compared against a reference level. These plots are produced only for statements with options that are compatible with control differences. For example, the following statements produce control plots for effects A and C:

```
lsmeans A / diff=control('1') plot=control;
lsmeans B / diff              plot=control;
lsmeans C                     plot=control;
```

The DIFF option in the second LSMEANS statement implies all pairwise differences.

DIFFPLOT<(*diffplot-options***)>**

DIFFOGRAM<(*diffplot-options***)>**

DIFF<(*diffplot-options***)>**

requests a display of all pairwise least squares mean differences and their significance. The display is also known as a "mean-mean scatter plot" when it is based on arithmetic means (Hsu 1996; Hsu and Peruggia 1994). For each comparison a line segment, centered at the LS-means in the pair, is drawn. The length of the segment corresponds to the projected width of a confidence interval for the least squares mean difference. Segments that fail to cross the 45-degree reference line correspond to significant least squares mean differences.

LS-mean difference plots are produced only for statements with options that are compatible with the display. For example, the following statements request differences against a control level for the A effect, all pairwise differences for the B effect, and the least squares means for the C effect:

```
lsmeans A / diff=control('1') plot=diff;
lsmeans B / diff              plot=diff;
lsmeans C                     plot=diff;
```

The DIFF= type in the first statement is incompatible with a display of all pairwise differences.

You can specify the following *diffplot-options*. The ABS and NOABS options determine the positioning of the line segments in the plot. When the ABS option is in effect, and this is the default, all line segments are shown on the same side of the reference line. The NOABS option separates comparisons according to the sign of the difference. The CENTER option marks the center point for each comparison. This point corresponds to the intersection of two least squares means. The NOLINES option suppresses the display of the line segments that represent the confidence bounds for the differences of the least squares means. The NOLINES option implies the CENTER option. The default is to draw line segments in the upper portion of the plot area without marking the center point.

MEANPLOT<(*meanplot-options***)>**

requests displays of the least squares means.

The following *meanplot-options* control the display of the least squares means.

ASCENDING
displays the least squares means in ascending order. This option has no effect if means are sliced or displayed in separate plots.

CL
displays upper and lower confidence limits for the least squares means. By default, 95% limits are drawn. You can change the confidence level with the ALPHA= option. Confidence limits are drawn by default if the CL option is specified in the LSMEANS statement.

CLBAND
displays confidence limits as bands. This option implies the JOIN option.

DESCENDING
displays the least squares means in descending order. This option has no effect if means are sliced or displayed in separate plots.

ILINK
requests that means (and confidence limits) are displayed on the inverse linked scale.

JOIN
CONNECT
connects the least squares means with lines. This option is implied by the CLBAND option. If the effect contains nested variables, and a SLICEBY= effect contains classification variables that appear as crossed effects, this option is ignored.

SLICEBY=*fixed-effect*
specifies an effect by which to group the means in a single plot. For example, the following statement requests a plot in which the levels of A are placed on the horizontal axis and the means that belong to the same level of B are joined by lines:

```
lsmeans A*B / plot=meanplot(sliceby=b join);
```

Unless the LS-mean effect contains at least two classification variables, the SLICEBY= option has no effect. The SLICEBY= effect does not have to be an effect in your MODEL statement, but it must consist entirely of classification variables.

PLOTBY=*fixed-effect*
specifies an effect by which to break interaction plots into separate displays. For example, the following statement requests for each level of C one plot of the A*B cell means that are associated with that level of C:

```
lsmeans A*B*C / plot=meanplot(sliceby=b plotby=c clband);
```

In each plot, levels of A are displayed on the horizontal axis, and confidence bands are drawn around the means that share the same level of B.

The PLOTBY= option has no effect unless the LS-mean effect contains at least three classification variables. The PLOTBY= effect does not have to be an effect in the MODEL statement, but it must consist entirely of classification variables.

NONE
requests that no plots be produced.

When LS-mean calculations are adjusted for multiplicity by using the ADJUST= option, the plots are adjusted accordingly.

SINGULAR=*number*
tunes the estimability checking as documented for the CONTRAST statement.

SLICE=*fixed-effect*

SLICE=(*fixed-effects***)**
specifies effects by which to partition interaction LSMEANS effects. This can produce what are known as tests of simple effects (Winer 1971). For example, suppose that A*B is significant, and you want to test the effect of A for each level of B. The appropriate LSMEANS statement is

```
lsmeans A*B / slice=B;
```

This statement tests for the simple main effects of A for B, which are calculated by extracting the appropriate rows from the coefficient matrix for the A*B LS-means and by using them to form an F test.

The SLICE option produces F tests that test the simultaneous equality of cell means at a fixed level of the slice effect (Schabenberger, Gregoire, and Kong 2000). You can request differences of the least squares means while holding one or more factors at a fixed level with the SLICEDIFF= option.

The SLICE option produces a table titled "Tests of Effect Slices."

SLICEDIFF=*fixed-effect*

SLICEDIFF=(*fixed-effects***)**

SIMPLEDIFF=*fixed-effect*

SIMPLEDIFF=(*fixed-effects***)**
requests that differences of simple effects be constructed and tested against zero. Whereas the SLICE option extracts multiple rows of the coefficient matrix and forms an F test, the SLICEDIFF option tests pairwise differences of these rows. This enables you to perform multiple comparisons among the levels of one factor at a fixed level of the other factor. For example, assume that, in a balanced design, factors A and B have $a = 4$ and $b = 3$ levels, respectively. Consider the following statements:

```
proc glimmix;
   class a b;
   model y = a b a*b;
   lsmeans a*b / slice=a;
   lsmeans a*b / slicediff=a;
run;
```

The first LSMEANS statement produces four F tests, one per level of A. The first of these tests is constructed by extracting the three rows corresponding to the first level of A from the coefficient matrix for the A*B interaction. Call this matrix $\mathbf{L}_{a1}$ and its rows $\mathbf{l}_{a1}^{(1)}$, $\mathbf{l}_{a1}^{(2)}$, and $\mathbf{l}_{a1}^{(3)}$. The SLICE tests the two-degrees-of-freedom hypothesis

$$H: \begin{cases} \left(\mathbf{l}_{a1}^{(1)} - \mathbf{l}_{a1}^{(2)}\right)\boldsymbol{\beta} = 0 \\ \left(\mathbf{l}_{a1}^{(1)} - \mathbf{l}_{a1}^{(3)}\right)\boldsymbol{\beta} = 0 \end{cases}$$

In a balanced design, where μ_{ij} denotes the mean response if A is at level i and B is at level j, this hypothesis is equivalent to $H: \mu_{11} = \mu_{12} = \mu_{13}$. The SLICEDIFF option considers the three rows of $\mathbf{L}_{a1}$ in turn and performs tests of the difference between pairs of rows. How these differences are constructed depends on the SLICEDIFFTYPE= option. By default, all pairwise differences within the subset of $\mathbf{L}$ are considered; in the example this corresponds to tests of the form

$$H: \left(\mathbf{l}_{a1}^{(1)} - \mathbf{l}_{a1}^{(2)}\right)\boldsymbol{\beta} = 0$$
$$H: \left(\mathbf{l}_{a1}^{(1)} - \mathbf{l}_{a1}^{(3)}\right)\boldsymbol{\beta} = 0$$
$$H: \left(\mathbf{l}_{a1}^{(2)} - \mathbf{l}_{a1}^{(3)}\right)\boldsymbol{\beta} = 0$$

In the example, with $a = 4$ and $b = 3$, the second LSMEANS statement produces four sets of least squares means differences. Within each set, factor A is held fixed at a particular level and each set consists of three comparisons.

When the ADJUST= option is specified, the GLIMMIX procedure also adjusts the tests for multiplicity. The adjustment is based on the number of comparisons within each level of the SLICEDIFF= effect; see the SLICEDIFFTYPE= option. The Nelson adjustment is not available for slice differences.

SLICEDIFFTYPE< =*difftype*>

SIMPLEDIFFTYPE< =*difftype*>

determines the type of simple effect differences produced with the SLICEDIFF= option.

The possible values for the *difftype* are ALL, CONTROL, CONTROLL, and CONTROLU. The *difftype* ALL requests all simple effects differences, and it is the default. The *difftype* CONTROL requests the differences with a control, which, by default, is the first level of each of the specified LSMEANS effects.

To specify which levels of the effects are the controls, list the quoted formatted values in parentheses after the keyword CONTROL. For example, if the effects A, B, and C are classification variables, each having three levels (1, 2, and 3), the following LSMEANS statement specifies the (1,3) level of A*B as the control:

```
lsmeans A*B / slicediff=(A B)
              slicedifftype=control('1' '3');
```

This LSMEANS statement first produces simple effects differences holding the levels of A fixed, and then it produces simple effects differences holding the levels of B fixed. In the former case, level '3' of B serves as the control level. In the latter case, level '1' of A serves as the control.

For multiple effects, the results depend upon the order of the list, and so you should check the output to make sure that the controls are correct.

Two-tailed tests and confidence limits are associated with the CONTROL *difftype*. For one-tailed results, use either the CONTROLL or CONTROLU *difftype*. The CONTROLL *difftype* tests whether the noncontrol levels are significantly smaller than the control; the upper confidence limits for the control minus the noncontrol levels are considered to be infinity and are displayed as missing. Conversely, the CONTROLU *difftype* tests whether the noncontrol levels are significantly larger than the control; the upper confidence limits for the noncontrol levels minus the control are considered to be infinity and are displayed as missing.

STEPDOWN<(*step-down options*)>

requests that multiple comparison adjustments for the *p*-values of LS-mean differences be further adjusted in a step-down fashion. Step-down methods increase the power of multiple comparisons by taking advantage of the fact that a *p*-value will never be declared significant unless all smaller *p*-values are also declared significant. Note that the STEPDOWN adjustment combined with ADJUST=BON corresponds to the methods of Holm (1979) and Shaffer's "Method 2" (1986); this is the default. Using step-down-adjusted *p*-values combined with ADJUST=SIMULATE corresponds to the method of Westfall (1997).

If the degrees-of-freedom method is DDFM=KENWARDROGER or DDFM=SATTERTHWAITE, then step-down-adjusted *p*-values are produced only if the ADJDFE=ROW option is in effect.

Also, STEPDOWN affects only *p*-values, not confidence limits. For ADJUST=SIMULATE, the generalized least squares hybrid approach of Westfall (1997) is employed to increase Monte Carlo accuracy.

You can specify the following *step-down options* in parentheses:

MAXTIME=*n*

specifies the time (in seconds) to spend computing the maximal logically consistent sequential subsets of equality hypotheses for TYPE=LOGICAL. The default is MAXTIME=60. If the MAXTIME value is exceeded, the adjusted tests are not computed. When this occurs, you can try increasing the MAXTIME value. However, note that there are common multiple comparisons problems for which this computation requires a huge amount of time—for example, all pairwise comparisons between more than 10 groups. In such cases, try to use TYPE=FREE (the default) or TYPE=LOGICAL(*n*) for small *n*.

REPORT

specifies that a report on the step-down adjustment should be displayed, including a listing of the sequential subsets (Westfall 1997) and, for ADJUST=SIMULATE, the step-down simulation results.

TYPE=LOGICAL<(*n*)>

TYPE=FREE

If you specify TYPE=LOGICAL, the step-down adjustments are computed by using maximal logically consistent sequential subsets of equality hypotheses (Shaffer 1986,

Westfall 1997). Alternatively, for TYPE=FREE, sequential subsets are computed ignoring logical constraints. The TYPE=FREE results are more conservative than those for TYPE=LOGICAL, but they can be much more efficient to produce for many comparisons. For example, it is not feasible to take logical constraints between all pairwise comparisons of more than 10 groups. For this reason, TYPE=FREE is the default.

However, you can reduce the computational complexity of taking logical constraints into account by limiting the depth of the search tree used to compute them, specifying the optional depth parameter as a number n in parentheses after TYPE=LOGICAL. As with TYPE=FREE, results for TYPE=LOGICAL(n) are conservative relative to the true TYPE=LOGICAL results, but even for TYPE=LOGICAL(0) they can be appreciably less conservative than TYPE=FREE and they are computationally feasible for much larger numbers of comparisons. If you do not specify n or if $n = -1$, the full search tree is used.

LSMESTIMATE Statement

LSMESTIMATE *fixed-effect* < *'label'* > *values* < *divisor*=*n* >
< , < *'label'* > *values* < *divisor*=*n* > > < , . . . >
< / *options* > ;

The LSMESTIMATE statement provides a mechanism for obtaining custom hypothesis tests among the least squares means. In contrast to the hypotheses tested with the ESTIMATE or CONTRAST statements, the LSMESTIMATE statement enables you to form linear combinations of the least squares means, rather than linear combination of fixed-effects parameter estimates and/or random-effects solutions. Multiple-row sets of coefficients are permitted.

The computation of an LSMESTIMATE involves two coefficient matrices. Suppose that the *fixed-effect* has n_l levels. Then the LS-means are formed as $\mathbf{L}_1\widehat{\boldsymbol{\beta}}$, where $\mathbf{L}_1$ is a $(n_l \times p)$ coefficient matrix. The $(k \times n_l)$ coefficient matrix $\mathbf{K}$ is formed from the *values* that you supply in the k rows of the LSMESTIMATE statement. The least squares means estimates then represent the $(k \times 1)$ vector

$$\mathbf{K}\mathbf{L}_1\boldsymbol{\beta} = \mathbf{L}\boldsymbol{\beta}$$

The GLIMMIX procedure supports nonpositional syntax for the coefficients (*values*) in the LSMESTIMATE statement. For details see the section "Positional and Nonpositional Syntax for Contrast Coefficients" on page 2264.

PROC GLIMMIX produces a t test for each row of coefficients specified in the LSMESTIMATE statement. You can adjust p-values and confidence intervals for multiplicity with the ADJUST= option. You can obtain an F test of single-row or multirow LSMESTIMATEs with the FTEST option.

Note that in contrast to a multirow estimate in the ESTIMATE statement, you specify only a single fixed effect in the LSMESTIMATE statement. The row labels are optional and follow the effects specification. For example, the following statements fit a split-split-plot design and compare the average of the third and fourth LS-mean of the whole-plot factor A to the first LS-mean of the factor:

```
proc glimmix;
   class a b block;
   model y = a b a*b /  s;
   random int a / sub=block;
   lsmestimate A 'a1 vs avg(a3,a4)' 2 0 -1 -1 divisor=2;
run;
```

The order in which coefficients are assigned to the least squares means corresponds to the order in which they are displayed in the "Least Squares Means" table. You can use the ELSM option to see how coefficients are matched to levels of the *fixed-effect*.

The optional *divisor=n* specification enables you to assign a separate divisor to each row of the LSMESTIMATE. You can also assign divisor values through the DIVISOR= option. See the documentation that follows for the interaction between the two ways of specifying divisors.

Many options of the LSMESTIMATE statement affect the computation of least squares means—for example, the AT=, BYLEVEL, and OM options. See the documentation for the LSMEANS statement for details.

You can specify the following options in the LSMESTIMATE statement after a slash (/).

ADJDFE=SOURCE

ADJDFE=ROW

specifies how denominator degrees of freedom are determined when *p*-values and confidence limits are adjusted for multiple comparisons with the ADJUST= option. When you do not specify the ADJDFE= option, or when you specify ADJDFE=SOURCE, the denominator degrees of freedom for multiplicity-adjusted results are the denominator degrees of freedom for the LS-mean effect in the "Type III Tests of Fixed Effects" table.

The ADJDFE=ROW setting is useful if you want multiplicity adjustments to take into account that denominator degrees of freedom are not constant across estimates. This can be the case, for example, when DDFM=SATTERTHWAITE or DDFM=KENWARDROGER is specified in the MODEL statement.

ADJUST=BON

ADJUST=SCHEFFE

ADJUST=SIDAK

ADJUST=SIMULATE< (*simoptions*) >

ADJUST=T

requests a multiple comparison adjustment for the *p*-values and confidence limits for the LS-mean estimates. The adjusted quantities are produced in addition to the unadjusted *p*-values and confidence limits. Adjusted confidence limits are produced if the CL or ALPHA= option is in effect. For a description of the adjustments, see Chapter 39, "The GLM Procedure," and Chapter 58, "The MULTTEST Procedure," of the *SAS/STAT User's Guide* as well as the documentation for the ADJUST= option in the LSMEANS statement.

Note that not all adjustment methods of the LSMEANS statement are available for the LSMESTIMATE statement. Multiplicity adjustments in the LSMEANS statement are designed specifically for differences of least squares means.

If you specify the STEPDOWN option, the *p*-values are further adjusted in a step-down fashion.

ALPHA=*number*

requests that a *t*-type confidence interval be constructed for each of the LS-means with confidence level $1 - number$. The value of *number* must be between 0 and 1; the default is 0.05.

AT *variable*=*value*

AT (*variable-list***)=(***value-list***)**

AT MEANS

enables you to modify the values of the covariates used in computing LS-means. See the AT option in the LSMEANS statement for details.

BYLEVEL

requests that PROC GLIMMIX compute separate margins for each level of the LSMEANS effect.

The standard LS-means have equal coefficients across classification effects. The BYLEVEL option changes these coefficients to be proportional to the observed margins. This adjustment is reasonable when you want your inferences to apply to a population that is not necessarily balanced but has the margins observed in the input data set. In this case, the resulting LS-means are actually equal to raw means for fixed-effects models and certain balanced random-effects models, but their estimated standard errors account for the covariance structure that you have specified. If a WEIGHT statement is specified, PROC GLIMMIX uses weighted margins to construct the LS-means coefficients.

If the AT option is specified, the BYLEVEL option disables it.

CHISQ

requests that chi-square tests be performed in addition to *F* tests, when you request an *F* test with the FTEST option.

CL

requests that *t*-type confidence limits be constructed for each of the LS-means. If DDFM=NONE, then PROC GLIMMIX uses infinite degrees of freedom for this test, essentially computing a *z* interval. The confidence level is 0.95 by default; this can be changed with the ALPHA= option.

CORR

displays the estimated correlation matrix of the linear combination of the least squares means.

COV

displays the estimated covariance matrix of the linear combination of the least squares means.

DF=*number*

specifies the degrees of freedom for the *t* test and confidence limits. The default is the denominator degrees of freedom taken from the "Type III Tests of Fixed Effects" table corresponding to the LS-means effect.

DIVISOR=*value-list*

specifies a list of values by which to divide the coefficients so that fractional coefficients can be entered as integer numerators. If you do not specify *value-list*, a default value of 1.0 is assumed. Missing values in the *value-list* are converted to 1.0.

If the number of elements in *value-list* exceeds the number of rows of the estimate, the extra values are ignored. If the number of elements in *value-list* is less than the number of rows of the estimate, the last value in *value-list* is carried forward.

If you specify a row-specific divisor as part of the specification of the estimate row, this value multiplies the corresponding value in the *value-list*. For example, the following statement divides the coefficients in the first row by 8, and the coefficients in the third and fourth row by 3:

```
lsmestimate A 'One vs. two'   8 -8   divisor=2,
              'One vs. three' 1  0 -1         ,
              'One vs. four'  3  0  0 -3      ,
              'One vs. five'  3  0  0  0  -3  / divisor=4,.,3;
```

Coefficients in the second row are not altered.

E

requests that the **L** coefficients of the estimable function be displayed. These are the coefficients that apply to the fixed-effect parameter estimates. The E option displays the coefficients that you would need to enter in an equivalent ESTIMATE statement.

ELSM

requests that the **K** matrix coefficients be displayed. These are the coefficients that apply to the LS-means. This option is useful to ensure that you assigned the coefficients correctly to the LS-means.

EXP

requests exponentiation of the least squares means estimate. When you model data with the logit link function and the estimate represents a log odds ratio, the EXP option produces an odds ratio. See the section "Odds and Odds Ratio Estimation" on page 2256 for important details concerning the computation and interpretation of odds and odds ratio results with the GLIMMIX procedure. If you specify the CL or ALPHA= option, the (adjusted) confidence limits for the estimate are also exponentiated.

FTEST< (LABEL='*label***') >**

produces an F test that jointly tests the rows of the LSMESTIMATE against zero. You can specify the optional label to identify the results from that test in the "LSMFtest" table.

ILINK

requests that the estimate and its standard error also be reported on the scale of the mean (the inverse linked scale). PROC GLIMMIX computes the value on the mean scale by applying the inverse link to the estimate. The interpretation of this quantity depends on the coefficients specified in your LSMESTIMATE statement and the link function. In a model for binary data with logit link, for example, the following LSMESTIMATE statement computes

$$q = \frac{1}{1 + \exp\{-(\tau_1 - \tau_2)\}}$$

where τ_1 and τ_2 are the least squares means associated with the first two levels of the classification effect A:

```
proc glimmix;
   class A;
   model y = A / dist=binary link=logit;
   lsmestimate A 1 -1 / ilink;
run;
```

The quantity q is not the difference of the probabilities associated with the two levels,

$$\pi_1 - \pi_2 = \frac{1}{1+\exp\{-\tau_1\}} - \frac{1}{1+\exp\{-\tau_2\}}$$

The standard error of the inversely linked estimate is based on the delta method. If you also specify the CL or ALPHA= option, the GLIMMIX procedure computes confidence intervals for the inversely linked estimate. These intervals are obtained by applying the inverse link to the confidence intervals on the linked scale.

LOWER

LOWERTAILED

requests that the p-value for the t test be based only on values less than the test statistic. A two-tailed test is the default. A lower-tailed confidence limit is also produced if you specify the CL or ALPHA= option.

Note that for ADJUST=SCHEFFE the one-sided adjusted confidence intervals and one-sided adjusted p-values are the same as the corresponding two-sided statistics, because this adjustment is based on only the right tail of the F distribution.

OBSMARGINS

OM

specifies a potentially different weighting scheme for the computation of LS-means coefficients. The standard LS-means have equal coefficients across classification effects; however, the OM option changes these coefficients to be proportional to those found in the input data set. See the OBSMARGINS option in the LSMEANS statement for further details.

SINGULAR=*number*

tunes the estimability checking as documented for the CONTRAST statement.

STEPDOWN<(*step-down-options***)>**

requests that multiplicity adjustments for the p-values of LS-mean estimates be further adjusted in a step-down fashion. Step-down methods increase the power of multiple testing procedures by taking advantage of the fact that a p-value will never be declared significant unless all smaller p-values are also declared significant. Note that the STEPDOWN adjustment combined with ADJUST=BON corresponds to the methods of Holm (1979) and Shaffer's "Method 2" (1986); this is the default. Using step-down-adjusted p-values combined with ADJUST=SIMULATE corresponds to the method of Westfall (1997).

If the degrees-of-freedom method is DDFM=KENWARDROGER or DDFM=SATTERTHWAITE, then step-down-adjusted p-values are produced only if the ADJDFE=ROW option is in effect.

Also, the STEPDOWN option affects only p-values, not confidence limits. For ADJUST=SIMULATE, the generalized least squares hybrid approach of Westfall (1997) is employed to increase Monte Carlo accuracy.

You can specify the following *step-down-options* in parentheses:

MAXTIME=*n*

specifies the time (in seconds) to spend computing the maximal logically consistent sequential subsets of equality hypotheses for TYPE=LOGICAL. The default is MAXTIME=60. If the MAXTIME value is exceeded, the adjusted tests are not computed. When this occurs, you can try increasing the MAXTIME value. However, note that there are common multiple comparisons problems for which this computation requires a huge amount of time—for example, all pairwise comparisons between more than 10 groups. In such cases, try to use TYPE=FREE (the default) or TYPE=LOGICAL(n) for small n.

ORDER=PVALUE

ORDER=ROWS

specifies the order in which the step-down tests are performed. ORDER=PVALUE is the default, with LS-mean estimates being declared significant only if all LS-mean estimates with smaller (unadjusted) p-values are significant. If you specify ORDER=ROWS, then significances are evaluated in the order in which they are specified.

REPORT

specifies that a report on the step-down adjustment be displayed, including a listing of the sequential subsets (Westfall 1997) and, for ADJUST=SIMULATE, the step-down simulation results.

TYPE=LOGICAL<(*n***)>**

TYPE=FREE

If you specify TYPE=LOGICAL, the step-down adjustments are computed by using maximal logically consistent sequential subsets of equality hypotheses (Shaffer 1986, Westfall 1997). Alternatively, for TYPE=FREE, sequential subsets are computed ignoring logical constraints. The TYPE=FREE results are more conservative than those for TYPE=LOGICAL, but they can be much more efficient to produce for many estimates. For example, it is not feasible to take logical constraints between all pairwise comparisons of more than about 10 groups. For this reason, TYPE=FREE is the default.

However, you can reduce the computational complexity of taking logical constraints into account by limiting the depth of the search tree used to compute them, specifying the optional depth parameter as a number n in parentheses after TYPE=LOGICAL. As with TYPE=FREE, results for TYPE=LOGICAL(n) are conservative relative to the true TYPE=LOGICAL results, but even for TYPE=LOGICAL(0), they can be appreciably less conservative than TYPE=FREE, and they are computationally feasible for much larger numbers of estimates. If you do not specify n or if $n = -1$, the full search tree is used.

UPPER

UPPERTAILED

requests that the p-value for the t test be based only on values greater than the test statistic. A two-tailed test is the default. An upper-tailed confidence limit is also produced if you specify the CL or ALPHA= option.

Note that for ADJUST=SCHEFFE the one-sided adjusted confidence intervals and one-sided adjusted p-values are the same as the corresponding two-sided statistics, because this adjustment is based on only the right tail of the F distribution.

MODEL Statement

MODEL *response* < **(***response-options***)** > **=** < *fixed-effects* > < / *model-options* > **;**

MODEL *events/trials* **=** < *fixed-effects* > < / *model-options* > **;**

The MODEL statement is required and names the dependent variable and the fixed effects. The *fixed-effects* determine the **X** matrix of the model (see the section "Notation for the Generalized Linear Mixed Model" for details). The specification of effects is the same as in the GLM or MIXED procedure. In contrast to PROC GLM, you do not specify random effects in the MODEL statement. However, in contrast to PROC GLM and PROC MIXED, continuous variables on the left and right side of the MODEL statement can be computed through PROC GLIMMIX programming statements.

An intercept is included in the fixed-effects model by default. It can be removed with the NOINT option.

The dependent variable can be specified by using either the *response* syntax or the *events/trials* syntax. The *events/trials* syntax is specific to models for binomial data. A binomial(n,π) variable is the sum of n independent Bernoulli trials with event probability π. Each Bernoulli trial results in either an event or a nonevent (with probability $1-\pi$). You use the *events/trials* syntax to indicate to the GLIMMIX procedure that the Bernoulli outcomes are grouped. The value of the second variable, *trials*, gives the number n of Bernoulli trials. The value of the first variable, *events*, is the number of events out of n. The values of both *events* and (*trials*−*events*) must be nonnegative and the value of trials must be positive. Observations for which these conditions are not met are excluded from the analysis. If the *events/trials* syntax is used, the GLIMMIX procedure defaults to the binomial distribution. The response is then the *events* variable. The *trials* variable is accounted in model fitting as an additional weight. If you use the *response* syntax, the procedure defaults to the normal distribution.

There are two sets of options in the MODEL statement. The *response-options* determine how the GLIMMIX procedure models probabilities for binary and multinomial data. The *model-options* control other aspects of model formation and inference. Table 38.6 summarizes important *response-options* and *model-options*. These are subsequently discussed in detail in alphabetical order by option category.

Table 38.6 Summary of Important MODEL Statement Options

Option	Description
Response Variable Options	
DESCENDING	reverses the order of response categories
EVENT=	specifies the event category in binary models
ORDER=	specifies the sort order for the response variable
REFERENCE=	specifies the reference category in generalized logit models
Model Building	
DIST=	specifies the response distribution
LINK=	specifies the link function
NOINT	excludes fixed-effect intercept from model
OFFSET=	specifies the offset variable for linear predictor
Statistical Computations	
ALPHA=α	determines the confidence level $(1-\alpha)$
CHISQ	requests chi-square tests
DDF=	specifies the denominator degrees of freedom (list)
DDFM=	specifies the method for computing denominator degrees of freedom
HTYPE=	selects the type of hypothesis test
NOCENTER	suppresses centering and scaling of **X** columns during the estimation phase
ZETA=	tunes sensitivity in computing Type III functions
Statistical Output	
CL	displays confidence limits for fixed-effects parameter estimates
CORRB	displays the correlation matrix of fixed-effects parameter estimates
COVB	displays the covariance matrix of fixed-effects parameter estimates
COVBI	displays the inverse covariance matrix of fixed-effects parameter estimates
E, E1, E2, E3	displays the **L** matrix coefficients
INTERCEPT	adds a row for the intercept to test tables
ODDSRATIO	displays odds ratios and confidence limits
SOLUTION	displays fixed-effects parameter estimates (and scale parameter in GLM models)
STDCOEF	displays standardized coefficients

Response Variable Options

Response variable options determine how the GLIMMIX procedure models probabilities for binary and multinomial data.

You can specify the following options by enclosing them in parentheses after the response variable. See the section "Response-Level Ordering and Referencing" on page 2267 for more detail and examples.

DESCENDING

DESC

reverses the order of the response categories. If both the DESCENDING and ORDER= options are specified, PROC GLIMMIX orders the response categories according to the ORDER= option and then reverses that order.

EVENT='*category***' | *keyword***

specifies the event category for the binary response model. PROC GLIMMIX models the probability of the event category. The EVENT= option has no effect when there are more than two response categories. You can specify the value (formatted, if a format is applied) of the event category in quotes, or you can specify one of the following keywords:

FIRST

designates the first ordered category as the event. This is the default.

LAST

designates the last ordered category as the event.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sort order for the levels of the response variable. When ORDER=FORMATTED (the default) for numeric variables for which you have supplied no explicit format (that is, for which there is no corresponding FORMAT statement in the current PROC GLIMMIX run or in the DATA step that created the data set), the levels are ordered by their internal (numeric) value. If you specify the ORDER= option in the MODEL statement and the ORDER= option in the PROC GLIMMIX statement, the former takes precedence. The following table shows the interpretation of the ORDER= values:

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For the FORMATTED and INTERNAL values, the sort order is machine dependent.

For more information about sort order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

REFERENCE='*category***' | *keyword***
REF='*category***' | *keyword***
specifies the reference category for the generalized logit model and the binary response model. For the generalized logit model, each nonreference category is contrasted with the reference category. For the binary response model, specifying one response category as the reference is the same as specifying the other response category as the event category. You can specify the value (formatted if a format is applied) of the reference category in quotes, or you can specify one of the following keywords:

FIRST
designates the first ordered category as the reference category.

LAST
designates the last ordered category as the reference category. This is the default.

Model Options

ALPHA=*number*
requests that a t-type confidence interval be constructed for each of the fixed-effects parameters with confidence level $1 - number$. The value of *number* must be between 0 and 1; the default is 0.05.

CHISQ
requests that chi-square tests be performed for all specified effects in addition to the F tests. Type III tests are the default; you can produce the Type I and Type II tests by using the HTYPE= option.

CL
requests that t-type confidence limits be constructed for each of the fixed-effects parameter estimates. The confidence level is 0.95 by default; this can be changed with the ALPHA= option.

CORRB
produces the correlation matrix from the approximate covariance matrix of the fixed-effects parameter estimates.

COVB<(DETAILS)>
produces the approximate variance-covariance matrix of the fixed-effects parameter estimates $\widehat{\boldsymbol{\beta}}$. In a generalized linear mixed model this matrix typically takes the form $(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}$ and can be obtained by sweeping the mixed model equations; see the section "Estimated Precision of Estimates" on page 2221. In a model without random effects, it is obtained from the inverse of the observed or expected Hessian matrix. Which Hessian is used in

the computation depends on whether the procedure is in scoring mode (see the SCORING= option in the PROC GLIMMIX statement) and whether the EXPHESSIAN option is in effect. Note that if you use EMPIRICAL= or DDFM=KENWARDROGER, the matrix displayed by the COVB option is the empirical (sandwich) estimator or the adjusted estimator, respectively.

The DETAILS suboption of the COVB option enables you to obtain a table of statistics about the covariance matrix of the fixed effects. If an adjusted estimator is used because of the EMPIRICAL= or DDFM=KENWARDROGER option, the GLIMMIX procedure displays statistics for the adjusted and unadjusted estimators as well as statistics comparing them. This enables you to diagnose, for example, changes in rank (because of an insufficient number of subjects for the empirical estimator) and to assess the extent of the covariance adjustment. In addition, the GLIMMIX procedure then displays the unadjusted (=model-based) covariance matrix of the fixed-effects parameter estimates. For more details, see the section "Exploring and Comparing Covariance Matrices" on page 2246.

COVBI

produces the inverse of the approximate covariance matrix of the fixed-effects parameter estimates.

DDF=*value-list*

DF=*value-list*

enables you to specify your own denominator degrees of freedom for the fixed effects. The *value-list* specification is a list of numbers or missing values (.) separated by commas. The degrees of freedom should be listed in the order in which the effects appear in the "Type III Tests of Fixed Effects" table. If you want to retain the default degrees of freedom for a particular effect, use a missing value for its location in the list. For example, the statement assigns 3 denominator degrees of freedom to A and 4.7 to A*B, while those for B remain the same:

```
model Y = A B A*B / ddf=3,.,4.7;
```

If you select a degrees-of-freedom method with the DDFM= option, then nonmissing, positive values in *value-list* override the degrees of freedom for the particular effect. For example, the statement assigns 3 and 6 denominator degrees of freedom in the test of the A main effect and the A*B interaction, respectively:

```
model Y = A B A*B / ddf=3,.,6  ddfm=Satterth;
```

The denominator degrees of freedom for the test for the B effect are determined from a Satterthwaite approximation.

Note that the DDF= and DDFM= options determine the degrees of freedom in the "Type I Tests of Fixed Effects," "Type II Tests of Fixed Effects," and "Type III Tests of Fixed Effects" tables. These degrees of freedom are also used in determining the degrees of freedom in tests and confidence intervals from the CONTRAST, ESTIMATE, LSMEANS, and LSMESTIMATE statements. Exceptions from this rule are noted in the documentation for the respective statements.

DDFM=BETWITHIN

DDFM=CONTAIN

DDFM=KENWARDROGER<(FIRSTORDER)>

DDFM=NONE

DDFM=RESIDUAL

DDFM=SATTERTHWAITE

specifies the method for computing the denominator degrees of freedom for the tests of fixed effects resulting from the MODEL, CONTRAST, ESTIMATE, LSMEANS, and LSMESTIMATE statements.

Table 38.7 table lists syntax aliases for the degrees-of-freedom methods.

Table 38.7 Aliases for the DDFM= Option

DDFM= Option	Alias
BETWITHIN	BW
CONTAIN	CON
KENWARDROGER	KENROG, KR
RESIDUAL	RES
SATTERTHWAITE	SATTERTH, SAT

The DDFM=BETWITHIN option divides the residual degrees of freedom into between-subject and within-subject portions. PROC GLIMMIX then determines whether a fixed effect changes within any subject. If the GLIMMIX procedure does not process the data by subjects, the DDFM=BETWITHIN option has no effect. See the section "Processing by Subjects" on page 2247 for details. If so, it assigns within-subject degrees of freedom to the effect; otherwise, it assigns the between-subject degrees of freedom to the effect (see Schluchter and Elashoff 1990). If there are multiple within-subject effects containing classification variables, the within-subject degrees of freedom are partitioned into components corresponding to the subject-by-effect interactions.

One exception to the preceding method is the case where you model only R-side covariation with an unstructured covariance matrix (TYPE=UN option). In this case, all fixed effects are assigned the between-subject degrees of freedom to provide for better small-sample approximations to the relevant sampling distributions. The DDFM=BETWITHIN method is the default for models with only R-side random effects and a SUBJECT= option.

The DDFM=CONTAIN option invokes the *containment method* to compute denominator degrees of freedom, and this method is the default when the model contains G-side random effects. The containment method is carried out as follows: Denote the fixed effect in question A and search the G-side random effect list for the effects that *syntactically* contain A. For example, the effect B(A) contains A, but the effect C does not, even if it has the same levels as B(A).

Among the random effects that contain A, compute their rank contributions to the $[\mathbf{X}\ \mathbf{Z}]$ matrix (in order). The denominator degrees of freedom assigned to A is the smallest of these rank contributions. If no effects are found, the denominator df for A is set equal to the residual degrees of freedom, $n - \text{rank}[\mathbf{X}\ \mathbf{Z}]$. This choice of degrees of freedom is the same as for

the tests performed for balanced split-plot designs and should be adequate for moderately unbalanced designs.

CAUTION: If you have a **Z** matrix with a large number of columns, the overall memory requirements and the computing time after convergence can be substantial for the containment method. If it is too large, you might want to use a different degrees-of-freedom method, such as DDFM=RESIDUAL, DDFM=NONE, or DDFM=BETWITHIN.

DDFM=NONE specifies that no denominator degrees of freedom be applied. PROC GLIMMIX then essentially assumes that infinite degrees of freedom are available in the calculation of p-values. The p-values for t tests are then identical to p-values derived from the standard normal distribution. In the case of F tests, the p-values equal those of chi-square tests determined as follows: if F_{obs} is the observed value of the F test with l numerator degrees of freedom, then

$$p = \Pr\{F_{l,\infty} > F_{obs}\} = \Pr\{\chi^2_l > lF_{obs}\}$$

Regardless of the DDFM= method, you can obtain these chi-square p-values with the CHISQ option in the MODEL statement.

The DDFM=RESIDUAL option performs all tests by using the residual degrees of freedom, $n - \text{rank}(\mathbf{X})$, where n is the sum of the frequencies used. It is the default degrees of freedom method for GLMs and overdispersed GLMs.

The DDFM=KENWARDROGER option applies the (prediction) standard error and degrees-of-freedom correction detailed by Kenward and Roger (1997). This approximation involves inflating the estimated variance-covariance matrix of the fixed and random effects in a manner similar to that of Prasad and Rao (1990), Harville and Jeske (1992), and Kackar and Harville (1984). Satterthwaite-type degrees of freedom are then computed based on this adjustment. By default, the observed information matrix of the covariance parameter estimates is used in the calculations. For covariance structures that have nonzero second derivatives with respect to the covariance parameters, the Kenward-Roger covariance matrix adjustment includes a second-order term. This term can result in standard error shrinkage. Also, the resulting adjusted covariance matrix can then be indefinite and is not invariant under reparameterization. The FIRSTORDER suboption of the DDFM=KENWARDROGER option eliminates the second derivatives from the calculation of the covariance matrix adjustment. For the case of scalar estimable functions, the resulting estimator is referred to as the Prasad-Rao estimator $\widetilde{m}^{@}$ in Harville and Jeske (1992). You can use the COVB(DETAILS) option to diagnose the adjustments made to the covariance matrix of fixed-effects parameter estimates by the GLIMMIX procedure. An application with DDFM=KENWARDROGER is presented in Example 38.8. The following are examples of covariance structures that generally lead to nonzero second derivatives: TYPE=ANTE(1), TYPE=AR(1), TYPE=ARH(1), TYPE=ARMA(1,1), TYPE=CHOL, TYPE=CSH, TYPE=FA0(q), TYPE=TOEPH, TYPE=UNR, and all TYPE=SP() structures.

The DDFM=SATTERTHWAITE option performs a general Satterthwaite approximation for the denominator degrees of freedom in a generalized linear mixed model. This method is a generalization of the techniques described in Giesbrecht and Burns (1985), McLean and Sanders (1988), and Fai and Cornelius (1996). The method can also include estimated random effects. The calculations require extra memory to hold q matrices that are the size of

the mixed model equations, where q is the number of covariance parameters. Extra computing time is also required to process these matrices. The Satterthwaite method implemented is intended to produce an accurate F approximation; however, the results can differ from those produced by PROC GLM. Also, the small sample properties of this approximation have not been extensively investigated for the various models available with PROC GLIMMIX. Computational details can be found in the section "Satterthwaite Degrees of Freedom Approximation" on page 2241.

When the asymptotic variance matrix of the covariance parameters is found to be singular, a generalized inverse is used. Covariance parameters with zero variance then do not contribute to the degrees of freedom adjustment for DDFM=SATTERTH and DDFM=KENWARDROGER, and a message is written to the log.

DISTRIBUTION=*keyword*

DIST=*keyword*

D=*keyword*

ERROR=*keyword*

E=*keyword*

specifies the built-in (conditional) probability distribution of the data. If you specify the DIST= option and you do not specify a user-defined link function, a default link function is chosen according to the following table. If you do not specify a distribution, the GLIMMIX procedure defaults to the normal distribution for continuous response variables and to the multinomial distribution for classification or character variables, unless the *events/trial* syntax is used in the MODEL statement. If you choose the *events/trial* syntax, the GLIMMIX procedure defaults to the binomial distribution.

Table 38.8 lists the values of the DIST= option and the corresponding default link functions. For the case of generalized linear models with these distributions, you can find expressions for the log-likelihood functions in the section "Maximum Likelihood" on page 2212.

Table 38.8 Keyword Values of the DIST= Option

DIST=	Distribution	Default Link Function	Numeric Value
BETA	beta	logit	12
BINARY	binary	logit	4
BINOMIAL | BIN | B	binomial	logit	3
EXPONENTIAL | EXPO	exponential	log	9
GAMMA | GAM	gamma	log	5
GAUSSIAN | G | NORMAL | N	normal	identity	1
GEOMETRIC | GEOM	geometric	log	8
INVGAUSS | IGAUSSIAN | IG	inverse Gaussian	inverse squared (power(−2))	6
LOGNORMAL | LOGN	lognormal	identity	11
MULTINOMIAL | MULTI | MULT	multinomial	cumulative logit	NA
NEGBINOMIAL | NEGBIN | NB	negative binomial	log	7
POISSON | POI | P	Poisson	log	2
TCENTRAL | TDIST | T	t	identity	10
BYOBS(*variable*)	multivariate	varied	NA

Note that the PROC GLIMMIX default link for the gamma or exponential distribution is not the canonical link (the reciprocal link).

The numeric value in the last column of Table 38.8 can be used in combination with DIST=BYOBS. The BYOBS(*variable*) syntax designates a variable whose value identifies the distribution to which an observation belongs. If the variable is numeric, its values must match values in the last column of Table 38.8. If the variable is not numeric, an observation's distribution is identified by the first four characters of the distribution's name in the leftmost column of the table. Distributions whose numeric value is "NA" cannot be used with DIST=BYOBS.

If the variable in BYOBS(*variable*) is a data set variable, it can also be used in the CLASS statement of the GLIMMIX procedure. For example, this provides a convenient method to model multivariate data jointly while varying fixed-effects components across outcomes. Assume that, for example, for each patient, a count and a continuous outcome were observed; the count data are modeled as Poisson data and the continuous data are modeled as gamma variates. The following statements fit a Poisson and a gamma regression model simultaneously:

```
proc sort data=yourdata;
   by patient;
run;
data yourdata;
   set yourdata;
   by patient;
   if first.patient then dist='POIS' else dist='GAMM';
run;
proc glimmix data=yourdata;
   class treatment dist;
   model y = dist treatment*dist / dist=byobs(dist);
run;
```

The two models have separate intercepts and treatment effects. To correlate the outcomes, you can share a random effect between the observations from the same patient:

```
proc glimmix data=yourdata;
   class treatment dist patient;
   model y = dist treatment*dist / dist=byobs(dist);
   random intercept / subject=patient;
run;
```

Or, you could use an R-side correlation structure:

```
proc glimmix data=yourdata;
   class treatment dist patient;
   model y = dist treatment*dist / dist=byobs(dist);
   random _residual_ / subject=patient type=un;
run;
```

Although DIST=BYOBS(*variable*) is used to model multivariate data, you only need a single response variable in PROC GLIMMIX. The responses are in "univariate" form. This allows, for example, different missing value patterns across the responses. It does, however, require that all response variables be numeric.

The default links that are assigned when DIST=BYOBS is in effect correspond to the respective default links in Table 38.8.

When you choose DIST=LOGNORMAL, the GLIMMIX procedure models the logarithm of the response variable as a normal random variable. That is, the mean and variance are estimated on the logarithmic scale, assuming a normal distribution, $\log\{Y\} \sim N(\mu, \sigma^2)$. This enables you to draw on options that require a distribution in the exponential family—for example, by using a scoring algorithm in a GLM. To convert means and variances for $\log\{Y\}$ into those of Y, use the relationships

$$\begin{aligned} \mathrm{E}[Y] &= \exp\{\mu\}\sqrt{\omega} \\ \mathrm{Var}[Y] &= \exp\{2\mu\}\omega(\omega - 1) \\ \omega &= \exp\{\sigma^2\} \end{aligned}$$

The DIST=T option models the data as a shifted and scaled central t variable. This enables you to model data with heavy-tailed distributions. If Y denotes the response and X has a t_ν distribution with ν degrees of freedom, then PROC GLIMMIX models

$$Y = \mu + \sqrt{\phi}X$$

In this parameterization, Y has mean μ and variance $\phi\nu/(\nu - 2)$.

By default, $\nu = 3$. You can supply different degrees of freedom for the t variate as in the following statements:

```
proc glimmix;
   class a b;
   model y = b x b*x / dist=tcentral(9.6);
   random a;
run;
```

The GLIMMIX procedure does not accept values for the degrees of freedom parameter less than 3.0. If the t distribution is used with the DIST=BYOBS(*variable*) specification, the degrees of freedom are fixed at $\nu = 3$. For mixed models where parameters are estimated based on linearization, choosing DIST=T instead of DIST=NORMAL affects only the residual variance, which decreases by the factor $\nu/(\nu - 2)$.

Note that in SAS 9.1, the GLIMMIX procedure modeled $Y = \mu + \phi^* \sqrt{\frac{\nu-2}{\nu}} X$. The scale parameter of the parameterizations are related as $\phi = \phi^* \times \phi^* \times (\nu - 2)/\nu$.

The DIST=BETA option implements the parameterization of the beta distribution in Ferrari and Cribari-Neto (2004). If Y has a beta(α, β) density, so that $\mathrm{E}[Y] = \mu = \alpha/(\alpha + \beta)$, this parameterization uses the variance function $a(\mu) = \mu(1 - \mu)$ and $\mathrm{Var}[Y] = a(\mu)/(1 + \phi)$.

See the section "Maximum Likelihood" on page 2212 for the log likelihoods of the distributions fitted by the GLIMMIX procedure.

E

requests that Type I, Type II, and Type III **L** matrix coefficients be displayed for all specified effects.

E1 | EI
requests that Type I **L** matrix coefficients be displayed for all specified effects.

E2 | EII
requests that Type II **L** matrix coefficients be displayed for all specified effects.

E3 | EIII
requests that Type III **L** matrix coefficients be displayed for all specified effects.

HTYPE=*value-list*
indicates the type of hypothesis test to perform on the fixed effects. Valid entries for values in the *value-list* are 1, 2, and 3, corresponding to Type I, Type II, and Type III tests. The default value is 3. You can specify several types by separating the values with a comma or a space. The ODS table names are "Tests1," "Tests2," and "Tests3" for the Type I, Type II, and Type III tests, respectively.

INTERCEPT
adds a row to the tables for Type I, II, and III tests corresponding to the overall intercept.

LINK=*keyword*
specifies the link function in the generalized linear mixed model. The keywords and their associated built-in link functions are shown in Table 38.9.

Table 38.9 Built-in Link Functions of the GLIMMIX Procedure

LINK=	Link Function	$g(\mu) = \eta =$	Numeric Value
CUMCLL | CCLL	cumulative complementary log-log	$\log(-\log(1-\pi))$	NA
CUMLOGIT | CLOGIT	cumulative logit	$\log(\gamma/(1-\pi))$	NA
CUMLOGLOG	cumulative log-log	$-\log(-\log(\pi))$	NA
CUMPROBIT | CPROBIT	cumulative probit	$\Phi^{-1}(\pi)$	NA
CLOGLOG | CLL	complementary log-log	$\log(-\log(1-\mu))$	5
GLOGIT | GENLOGIT	generalized logit		NA
IDENTITY | ID	identity	μ	1
LOG	log	$\log(\mu)$	4
LOGIT	logit	$\log(\mu/(1-\mu))$	2
LOGLOG	log-log	$-\log(-\log(\mu))$	6
PROBIT	probit	$\Phi^{-1}(\mu)$	3
POWER(λ) | POW(λ)	power with exponent λ= *number*	$\begin{cases} \mu^{\lambda} & \text{if } \lambda \neq 0 \\ \log(\mu) & \text{if } \lambda = 0 \end{cases}$	NA
POWERMINUS2	power with exponent -2	$1/\mu^2$	8
RECIPROCAL | INVERSE	reciprocal	$1/\mu$	7
BYOBS(*variable*)	varied	varied	NA

For the probit and cumulative probit links, $\Phi^{-1}(\cdot)$ denotes the quantile function of the standard normal distribution. For the other cumulative links, π denotes a cumulative category probability. The cumulative and generalized logit link functions are appropriate only for the

multinomial distribution. When you choose a cumulative link function, PROC GLIMMIX assumes that the data are ordinal. When you specify LINK=GLOGIT, the GLIMMIX procedure assumes that the data are nominal (not ordered).

The numeric value in the rightmost column of Table 38.9 can be used in conjunction with LINK=BYOBS(*variable*). This syntax designates a *variable* whose values identify the link function associated with an observation. If the variable is numeric, its values must match those in the last column of Table 38.9. If the variable is not numeric, an observation's link function is determined by the first four characters of the link's name in the first column. Those link functions whose numeric value is "NA" cannot be used with LINK=BYOBS(*variable*).

You can define your own link function through programming statements. See the section "User-Defined Link or Variance Function" on page 2208 for more information about how to specify a link function. If a user-defined link function is in effect, the specification in the LINK= option is ignored. If you specify neither the LINK= option nor a user-defined link function, then the default link function is chosen according to Table 38.8.

LWEIGHT=FIRSTORDER | FIRO

LWEIGHT=NONE

LWEIGHT=VAR

determines how weights are used in constructing the coefficients for Type I through Type III **L** matrices. The default is LWEIGHT=VAR, and the values of the WEIGHT variable are used in forming crossproduct matrices. If you specify LWEIGHT=FIRO, the weights incorporate the WEIGHT variable as well as the first-order weights of the linearized model. For LWEIGHT=NONE, the **L** matrix coefficients are based on the raw crossproduct matrix, whether a WEIGHT variable is specified or not.

NOCENTER

requests that the columns of the **X** matrix are not centered and scaled. By default, the columns of **X** are centered and scaled. Unless the NOCENTER option is in effect, **X** is replaced by $\mathbf{X}^*$ during estimation. The columns of $\mathbf{X}^*$ are computed as follows:

- In models with an intercept, the intercept column remains the same and the jth entry in row i of $\mathbf{X}^*$ is

$$x^*_{ij} = \frac{x_{ij} - \overline{x}_j}{\sqrt{\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2}}$$

- In models without intercept, no centering takes place and the jth entry in row i of $\mathbf{X}^*$ is

$$x^*_{ij} = \frac{x_{ij}}{\sqrt{\sum_{i=1}^{n}(x_{ij} - \overline{x}_j)^2}}$$

The effects of centering and scaling are removed when results are reported. For example, if the covariance matrix of the fixed effects is printed with the COVB option of the MODEL statement, the covariances are reported in terms of the original parameters, not the centered and scaled versions. If you specify the STDCOEF option, fixed-effects parameter estimates and their standard errors are reported in terms of the standardized (scaled and/or centered) coefficients in addition to the usual results in noncentered form.

NOINT

requests that no intercept be included in the fixed-effects model. An intercept is included by default.

ODDSRATIO<(*odds-ratio-options***)>**

OR<(*odds-ratio-options***)>**

requests estimates of odds ratios and their confidence limits, provided the link function is the logit, cumulative logit, or generalized logit. Odds ratios are produced for the following:

- classification main effects, if they appear in the MODEL statement
- continuous variables in the MODEL statement, unless they appear in an interaction with a classification effect
- continuous variables in the MODEL statement at fixed levels of a classification effect, if the MODEL statement contains an interaction of the two
- continuous variables in the MODEL statement, if they interact with other continuous variables

You can specify the following *odds-ratio-options* to create customized odds ratio results.

AT *var-list***=***value-list*

specifies the reference values for continuous variables in the model. By default, the average value serves as the reference. Consider, for example, the following statements:

```
proc glimmix;
   class A;
   model y = A x A*x / dist=binary oddsratio;
run;
```

Odds ratios for A are based on differences of least squares means for which x is set to its mean. Odds ratios for x are computed by differencing two sets of least squares mean for the A factor. One set is computed at x = $\overline{x} + 1$, and the second set is computed at x = $\overline{x}$. The following MODEL statement changes the reference value for x to 3:

```
model y = A x A*x / dist=binary
                    oddsratio(at x=3);
```

DIFF<=*difftype***>**

controls the type of differences for classification main effects. By default, odds ratios compare the odds of a response for level j of a factor to the odds of the response for the last level of that factor (DIFF=LAST). The DIFF=FIRST option compares the levels against the first level, DIFF=ALL produces odds ratios based on all pairwise differences, and DIFF=NONE suppresses odds ratios for classification main effects.

LABEL

displays a label in the "Odds Ratio Estimates" table. The table describes the comparison associated with the table row.

UNIT *var-list*=*value-list*

specifies the units in which the effects of continuous variable in the model are assessed. By default, odds ratios are computed for a change of one unit from the average. Consider a model with a classification factor A with 4 levels. The following statements produce an "Odds Ratio Estimates" table with 10 rows:

```
proc glimmix;
   class A;
   model y = A x A*x / dist=binary
                       oddsratio(diff=all unit x=2);
run;
```

The first $4 \times 3/2 = 6$ rows correspond to pairwise differences of levels of A. The underlying log odds ratios are computed as differences of A least squares means. In the least squares mean computation the covariate x is set to $\overline{x}$. The next four rows compare least squares means for A at $x = \overline{x} + 2$ and at $x = \overline{x}$. You can combine the AT and UNIT options to produce custom odds ratios. For example, the following statements produce an "Odds Ratio Estimates" table with 8 rows:

```
proc glimmix;
   class A;
   model y = A x x*z / dist=binary
                       oddsratio(diff=all
                                 at   x   = 3
                                 unit x z = 2 4);
run;
```

The first $4 \times 3/2 = 6$ rows correspond to pairwise differences of levels of A. The underlying log odds ratios are computed as differences of A least squares means. In the least squares mean computation, the covariate x is set to 3, and the covariate x*z is set to $3\overline{z}$. The next odds ratio measures the effect of a change in x. It is based on differencing the linear predictor for $x = 3 + 2$ and $x*z = (3 + 2)\overline{z}$ with the linear predictor for $x = 3$ and $x*z = 3\overline{z}$. The last odds ratio expresses a change in z by contrasting the linear predictors based on $x = 3$ and $x*z = 3(\overline{z} + 4)$ with the predictor based on $x = 3$ and $x*z = 3\overline{z}$.

To compute odds and odds ratios for general estimable functions and least squares means, see the ODDSRATIO option in the LSMEANS statement and the EXP options in the ESTIMATE and LSMESTIMATE statements.

For important details concerning interpretation and computation of odds ratios with the GLIMMIX procedure, see the section "Odds and Odds Ratio Estimation" on page 2256.

OFFSET=*variable*

specifies a variable to be used as an offset for the linear predictor. An offset plays the role of a fixed effect whose coefficient is known to be 1. You can use an offset in a Poisson model, for example, when counts have been obtained in time intervals of different lengths. With a log link function, you can model the counts as Poisson variables with the logarithm of the time interval as the offset variable. The offset variable cannot appear in the CLASS statement or elsewhere in the MODEL or RANDOM statement.

REFLINP=*r*

specifies a value for the linear predictor of the reference level in the generalized logit model for nominal data. By default $r = 0$.

SOLUTION

S

requests that a solution for the fixed-effects parameters be produced. Using notation from the section "Notation for the Generalized Linear Mixed Model" on page 2083, the fixed-effects parameter estimates are $\widehat{\boldsymbol{\beta}}$, and their (approximate) estimated standard errors are the square roots of the diagonal elements of $\widehat{\mathrm{Var}}[\widehat{\boldsymbol{\beta}}]$. This matrix commonly is of the form $(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}$ in GLMMs. You can output this approximate variance matrix with the COVB option. See the section "Details: GLIMMIX Procedure" on page 2212 on the construction of $\widehat{\mathbf{V}}$ in the various models.

Along with the estimates and their approximate standard errors, a t statistic is computed as the estimate divided by its standard error. The degrees of freedom for this t statistic matches the one appearing in the "Type III Tests of Fixed Effects" table under the effect containing the parameter. If DDFM=KENWARDROGER or DDFM=SATTERTHWAITE, the degrees of freedom are computed separately for each fixed-effect estimate, unless you override the value for any specific effect with the DDF=*value-list* option. The "Pr > |t|" column contains the two-tailed p-value corresponding to the t statistic and associated degrees of freedom. You can use the CL option to request confidence intervals for the fixed-effects parameters; they are constructed around the estimate by using a radius of the standard error times a percentage point from the t distribution.

STDCOEF

reports solutions for fixed effects in terms of the standardized (scaled and/or centered) coefficients. This option has no effect when the NOCENTER option is specified or in models for multinomial data.

ZETA=*number*

tunes the sensitivity in forming Type III functions. Any element in the estimable function basis with an absolute value less than *number* is set to 0. The default is 1E−8.

NLOPTIONS Statement

NLOPTIONS < *options* > **;**

Most models fit with the GLIMMIX procedure typically have one or more nonlinear parameters. Estimation requires nonlinear optimization methods. You can control the optimization through options in the NLOPTIONS statement.

Several estimation methods of the GLIMMIX procedure (METHOD=RSPL, MSPL, RMPL, MMPL) are doubly iterative in the following sense. The generalized linear mixed model is approximated by a linear mixed model based on current values of the covariance parameter estimates. The resulting linear mixed model is then fit, which is itself an iterative process (with some exceptions). On convergence, new covariance parameters and fixed-effects estimates are obtained and

the approximated linear mixed model is updated. Its parameters are again estimated iteratively. It is thus reasonable to refer to *outer* and *inner* iterations. The outer iterations involve the repeated updates of the linear mixed models, and the inner iterations are the iterative steps that lead to parameter estimates in any given linear mixed model. The NLOPTIONS statement controls the inner iterations. The outer iteration behavior can be controlled with options in the PROC GLIMMIX statement, such as the MAXLMMUPDATE=, PCONV=, and ABSPCONV= options. If the estimation method involves a singly iterative approach, then there is no need for the outer cycling and the model is fit in a single optimization controlled by the NLOPTIONS statement (see the section "Singly or Doubly Iterative Fitting" on page 2271).

The syntax and options of the NLOPTIONS statement are described in the section "Nonlinear Optimization: The NLOPTIONS Statement" on page 391 of Chapter 18, "Shared Concepts and Topics."

Note that in a GLMM with pseudo-likelihood estimation, specifying TECHNIQUE=NONE has the same effect as specifying the NOITER option in the PARMS statement. If you estimate the parameters by METHOD=LAPLACE or METHOD=QUAD, TECHNIQUE=NONE applies to the optimization after starting values have been determined.

The GLIMMIX procedure applies the default optimization technique shown in Table 38.10, depending on your model.

Table 38.10 Default Techniques

Model Family	Setting	TECHNIQUE=
GLM	DIST=NORMAL LINK=IDENTITY	NONE
GLM	otherwise	NEWRAP
GLMM	PARMS NOITER, PL	NONE
GLMM	binary data, PL	NRRIDG
GLMM	otherwise	QUANEW

OUTPUT Statement

OUTPUT < **OUT=***SAS-data-set* >
< *keyword*< **(***keyword-options***)** > < *=name* >>. . .
< *keyword*< **(***keyword-options***)** > < *=name* >> < */ options* > **;**

The OUTPUT statement creates a data set that contains predicted values and residual diagnostics, computed after fitting the model. By default, all variables in the original data set are included in the output data set.

You can use the ID statement to select a subset of the variables from the input data set as well as computed variables for adding to the output data set. If you reassign a data set variable through programming statements, the value of the variable from the input data set supersedes the recomputed value when observations are written to the output data set. If you list the variable in the ID statement,

however, PROC GLIMMIX saves the current value of the variable after the programming statements have been executed.

For example, suppose that data set Scores contains the variables score, machine, and person. The following statements fit a model with fixed machine and random person effects. The variable score divided by 100 is assumed to follow an inverse Gaussian distribution. The (conditional) mean and residuals are saved to the data set igausout. Because no ID statement is given, the variable score in the output data set contains the values from the input data set.

```
proc glimmix;
   class machine person;
   score = score/100;
   p = 4*_linp_;
   model score = machine  / dist=invgauss;
   random int / sub=person;
   output out=igausout pred=p resid=r;
run;
```

On the contrary, the following statements list explicitly which variables to save to the OUTPUT data set. Because the variable score is listed in the ID statement, and is (re-)assigned through programming statements, the values of score saved to the OUTPUT data set are the input values divided by 100.

```
proc glimmix;
   class machine person;
   score = score / 100;
   model score = machine  / dist=invgauss;
   random int / sub=person;
   output out=igausout pred=p resid=r;
   id machine score _xbeta_ _zgamma_;
run;
```

You can specify the following syntax elements in the OUTPUT statement before the slash (/).

OUT=*SAS-data-set*

DATA=*SAS-data-set*

specifies the name of the output data set. If the OUT= (or DATA=) option is omitted, the procedure uses the DATA*n* convention to name the output data set.

keyword<(*keyword-options***)><=***name***>**

specifies a statistic to include in the output data set and optionally assigns the variable the name *name*. You can use the *keyword-options* to control which type of a particular statistic to compute. The *keyword-options* can take on the following values:

BLUP	uses the predictors of the random effects in computing the statistic.
ILINK	computes the statistic on the scale of the data.
NOBLUP	does not use the predictors of the random effects in computing the statistic.
NOILINK	computes the statistic on the scale of the link function.

The default is to compute statistics by using BLUPs on the scale of the link function (the linearized scale). For example, the following OUTPUT statements are equivalent:

```
output out=out1  pred=predicted lcl=lower;

output out=out1  pred(blup noilink)=predicted
                 lcl (blup noilink)=lower;
```

If a particular combination of keyword and keyword options is not supported, the statistic is not computed and a message is produced in the SAS log.

A *keyword* can appear multiple times in the OUTPUT statement. Table 38.11 lists the keywords and the default names assigned by the GLIMMIX procedure if you do not specify a *name*. In this table, y denotes the observed response, and p denotes the linearized pseudo-data. See the section "Pseudo-likelihood Estimation Based on Linearization" on page 2220 for details on notation and the section "Notes on Output Statistics" on page 2278 for further details regarding the output statistics.

Table 38.11 Keywords for Output Statistics

Keyword	Options	Description	Expression	Name
PREDICTED	Default	Linear predictor	$\widehat{\eta} = \mathbf{x}'\widehat{\boldsymbol{\beta}} + \mathbf{z}'\widehat{\boldsymbol{\gamma}}$	Pred
	NOBLUP	Marginal linear predictor	$\widehat{\eta}_m = \mathbf{x}'\widehat{\boldsymbol{\beta}}$	PredPA
	ILINK	Predicted mean	$g^{-1}(\widehat{\eta})$	PredMu
	NOBLUP ILINK	Marginal mean	$g^{-1}(\widehat{\eta}_m)$	PredMuPA
STDERR	Default	Standard deviation of linear predictor	$\sqrt{\text{Var}[\widehat{\eta} - \mathbf{z}'\boldsymbol{\gamma}]}$	StdErr
	NOBLUP	Standard deviation of marginal linear predictor	$\sqrt{\text{Var}[\widehat{\eta}_m]}$	StdErrPA
	ILINK	Standard deviation of mean	$\sqrt{\text{Var}[g^{-1}(\widehat{\eta} - \mathbf{z}'\boldsymbol{\gamma})]}$	StdErr
	NOBLUP ILINK	Standard deviation of marginal mean	$\sqrt{\text{Var}[g^{-1}(\widehat{\eta}_m)]}$	StdErrMuPA
RESIDUAL	Default	Residual	$r = p - \widehat{\eta}$	Resid
	NOBLUP	Marginal residual	$r_m = p_m - \widehat{\eta}_m$	ResidPA
	ILINK	Residual on mean scale	$r_y = y - g^{-1}(\widehat{\eta})$	ResidMu
	NOBLUP ILINK	Marginal residual on mean scale	$r_{ym} = y - g^{-1}(\widehat{\eta}_m)$	ResidMuPA
PEARSON	Default	Pearson-type residual	$r/\sqrt{\widehat{\text{Var}}[p\|\boldsymbol{\gamma}]}$	Pearson
	NOBLUP	Marginal Pearson-type residual	$r_m/\sqrt{\widehat{\text{Var}}[p_m]}$	PearsonPA
	ILINK	Conditional Pearson-type mean residual	$r_y/\sqrt{\widehat{\text{Var}}[Y\|\boldsymbol{\gamma}]}$	PearsonMu
STUDENT	Default	Studentized residual	$r/\sqrt{\widehat{\text{Var}}[r]}$	Student

Table 38.11 *continued*

Keyword	Options	Description	Expression	Name
	NOBLUP	Studentized marginal residual	$r_m / \sqrt{\widehat{\text{Var}}[r_m]}$	StudentPA
LCL	Default	Lower prediction limit for linear predictor		LCL
	NOBLUP	Lower confidence limit for marginal linear predictor		LCLPA
	ILINK	Lower prediction limit for mean		LCLMu
	NOBLUP ILINK	Lower confidence limit for marginal mean		LCLMuPA
UCL	Default	Upper prediction limit for linear predictor		UCL
	NOBLUP	Upper confidence limit for marginal linear predictor		UCLPA
	ILINK	Upper prediction limit for mean		UCLMu
	NOBLUP ILINK	Upper confidence limit for marginal mean		UCLMuPA
VARIANCE	Default	Conditional variance of pseudo-data	$\widehat{\text{Var}}[p\|\boldsymbol{\gamma}]$	Variance
	NOBLUP	Marginal variance of pseudo-data	$\widehat{\text{Var}}[p_m]$	VariancePA
	ILINK	Conditional variance of response	$\widehat{\text{Var}}[Y\|\boldsymbol{\gamma}]$	Variance_Dep
	NOBLUP ILINK	Marginal variance of response	$\widehat{\text{Var}}[Y]$	Variance_DepPA

Studentized residuals are computed only on the linear scale (scale of the link), unless the link is the identity, in which case the two scales are equal. The keywords RESIDUAL, PEARSON, STUDENT, and VARIANCE are not available with the multinomial distribution. You can use the following shortcuts to request statistics: PRED for PREDICTED, STD for STDERR, RESID for RESIDUAL, and VAR for VARIANCE. Output statistics that depend on the marginal variance $\text{Var}[Y_i]$ are not available with METHOD=LAPLACE or METHOD=QUAD.

You can specify the following options in the OUTPUT statement after a slash (/).

ALLSTATS

requests that all statistics are computed. If you do not use a keyword to assign a name, the GLIMMIX procedure uses the default name.

ALPHA=*number*

determines the coverage probability for two-sided confidence and prediction intervals. The coverage probability is computed as $1 - number$. The value of *number* must be between 0 and 1; the default is 0.05.

DERIVATIVES

DER

adds derivatives of model quantities to the output data set. If, for example, the model fit requires the (conditional) log likelihood of the data, then the DERIVATIVES option writes for each observation the evaluations of the first and second derivatives of the log likelihood with respect to _LINP_ and _PHI_ to the output data set. The particular derivatives produced by the GLIMMIX procedure depend on the type of model and the estimation method.

NOMISS

requests that records be written to the output data only for those observations that were used in the analysis. By default, the GLIMMIX procedure produces output statistics for all observations in the input data set.

NOUNIQUE

requests that names not be made unique in the case of naming conflicts. By default, the GLIMMIX procedure avoids naming conflicts by assigning a unique name to each output variable. If you specify the NOUNIQUE option, variables with conflicting names are not renamed. In that case, the first variable added to the output data set takes precedence.

NOVAR

requests that variables from the input data set not be added to the output data set. This option does not apply to variables listed in the BY statement or to computed variables listed in the ID statement.

OBSCAT

requests that in models for multinomial data statistics be written to the output data set only for the response level that corresponds to the observed level of the observation.

SYMBOLS

SYM

adds to the output data set computed variables that are defined or referenced in the program.

PARMS Statement

PARMS <(*value-list*)> . . . </ *options*> **;**

The PARMS statement specifies initial values for the covariance or scale parameters, or it requests a grid search over several values of these parameters in generalized linear mixed models.

The *value-list* specification can take any of several forms:

m	a single value
$m_1, m_2, \ldots, m_n$	several values
m to n	a sequence where m equals the starting value, n equals the ending value, and the increment equals 1
m to n by i	a sequence where m equals the starting value, n equals the ending value, and the increment equals i
m_1, m_2 to m_3	mixed values and sequences

Using the PARMS Statement with a GLM

If you are fitting a GLM or a GLM with overdispersion, the scale parameters are listed at the end of the "Parameter Estimates" table in the same order as *value-list*. If you specify more than one set of initial values, PROC GLIMMIX uses only the first value listed for each parameter. Grid searches by using scale parameters are not possible for these models, because the fixed effects are part of the optimization.

Using the PARMS Statement with a GLMM

If you are fitting a GLMM, the *value-list* corresponds to the parameters as listed in the "Covariance Parameter Estimates" table. Note that this order can change depending on whether a residual variance is profiled or not; see the NOPROFILE option in the PROC GLIMMIX statement.

If you specify more than one set of initial values, PROC GLIMMIX performs a grid search of the objective function surface and uses the best point on the grid for subsequent analysis. Specifying a large number of grid points can result in long computing times.

Options in the PARMS Statement

You can specify the following options in the PARMS statement after a slash (/).

HOLD=*value-list*
specifies which parameter values PROC GLIMMIX should hold equal to the specified values. For example, the following statement constrains the first and third covariance parameters to equal 5 and 2, respectively:

```
parms (5) (3) (2) (3) / hold=1,3;
```

Covariance or scale parameters that are held fixed with the HOLD= option are treated as constrained parameters in the optimization. This is different from evaluating the objective function, gradient, and Hessian matrix at known values of the covariance parameters. A constrained parameter introduces a singularity in the optimization process. The covariance matrix of the covariance parameters (see the ASYCOV option of the PROC GLIMMIX statement) is then based on the projected Hessian matrix. As a consequence, the variance of parameters subjected to a HOLD= is zero. Such parameters do not contribute to the computation of denominator degrees of freedom with the DDFM=KENWARDROGER and DDFM=SATTERTHWAITE methods, for example. If you want to treat the covariance parameters as known, without imposing constraints on the optimization, you should use the NOITER option.

When you place a hold on all parameters (or when you specify the NOITER) option in a GLMM, you might notice that PROC GLIMMIX continues to produce an iteration history. Unless your model is a linear mixed model, several recomputations of the pseudo-response might be required in linearization-based methods to achieve agreement between the pseudo-data and the covariance matrix. In other words, the GLIMMIX procedure continues to update the fixed-effects estimates (and random-effects solutions) until convergence is achieved.

In certain models, placing a hold on covariance parameters implies that the procedure processes the parameters in the same order as if the NOPROFILE were in effect. This can change the order of the covariance parameters when you place a hold on one or more parameters. Models that are subject to this reordering are those with R-side covariance structures whose scale parameter could be profiled. This includes the TYPE=CS, TYPE=SP, TYPE=AR, TYPE=TOEP, and TYPE=ARMA covariance structures.

LOWERB=*value-list*

enables you to specify lower boundary constraints for the covariance or scale parameters. The *value-list* specification is a list of numbers or missing values (.) separated by commas. You must list the numbers in the same order that PROC GLIMMIX uses for the *value-list* in the PARMS statement, and each number corresponds to the lower boundary constraint. A missing value instructs PROC GLIMMIX to use its default constraint, and if you do not specify numbers for all of the covariance parameters, PROC GLIMMIX assumes that the remaining ones are missing.

This option is useful, for example, when you want to constrain the **G** matrix to be positive definite in order to avoid the more computationally intensive algorithms required when **G** becomes singular. The corresponding statements for a random coefficients model are as follows:

```
proc glimmix;
   class person;
   model y = time;
   random int time / type=chol sub=person;
   parms / lowerb=1e-4,.,1e-4;
run;
```

Here, the TYPE=CHOL structure is used in order to specify a Cholesky root parameterization for the 2×2 unstructured blocks in $\mathbf{G}$. This parameterization ensures that the $\mathbf{G}$ matrix is nonnegative definite, and the PARMS statement then ensures that it is positive definite by constraining the two diagonal terms to be greater than or equal to 1E−4.

NOBOUND

requests the removal of boundary constraints on covariance and scale parameters in mixed models. For example, variance components have a default lower boundary constraint of 0, and the NOBOUND option allows their estimates to be negative. See the NOBOUND option in the PROC GLIMMIX statement for further details.

NOITER

requests that no optimization of the covariance parameters be performed. This option has no effect in generalized linear models.

If you specify the NOITER option, PROC GLIMMIX uses the values for the covariance parameters given in the PARMS statement to perform statistical inferences. Note that the NOITER option is not equivalent to specifying a HOLD= value for all covariance parameters. If you use the NOITER option, covariance parameters are not constrained in the optimization. This prevents singularities that might otherwise occur in the optimization process.

If a residual variance is profiled, the parameter estimates can change from the initial values you provide as the residual variance is recomputed. To prevent an update of the residual variance, combine the NOITER option with the NOPROFILE option in the PROC GLIMMIX statements, as in the following code:

```
proc glimmix noprofile;
   class A B C rep mp sp;
   model y = A | B | C;
   random rep mp sp;
   parms (180) (200) (170) (1000) / noiter;
run;
```

When you specify the NOITER option in a model where parameters are estimated by pseudo-likelihood techniques, you might notice that the GLIMMIX procedure continues to produce an iteration history. Unless your model is a linear mixed model, several recomputations of the pseudo-response might be required in linearization-based methods to achieve agreement between the pseudo-data and the covariance matrix. In other words, the GLIMMIX procedure continues to update the profiled fixed-effects estimates (and random-effects solutions) until convergence is achieved. To prevent these updates, use the MAXLMMUPDATE= option in the PROC GLIMMIX statement. Specifying the NOITER option in the PARMS statement of a GLMM with pseudo-likelihood estimation has the same effect as choosing TECHNIQUE=NONE in the NLOPTIONS statement.

If you want to base initial fixed-effects estimates on the results of fitting a generalized linear model, then you can combine the NOITER option with the TECHNIQUE= option. For example, the following statements determine the starting values for the fixed effects by fitting a logistic model (without random effects) with the Newton-Raphson algorithm:

```
proc glimmix startglm inititer=10;
   class clinic A;
   model y/n = A / link=logit dist=binomial;
   random clinic;
   parms (0.4) / noiter;
   nloptions technique=newrap;
run;
```

The initial GLM fit stops at convergence or after at most 10 iterations, whichever comes first. The pseudo-data for the linearized GLMM is computed from the GLM estimates. The variance of the Clinic random effect is held constant at 0.4 during subsequent iterations that update the fixed effects only.

If you also want to combine the GLM fixed-effects estimates with known and fixed covariance parameter values without updating the fixed effects, you can add the MAXLMMUPDATE=0 option:

```
proc glimmix startglm inititer=10 maxlmmupdate=0;
   class clinic A;
   model y/n = A / link=logit dist=binomial;
   random clinic;
   parms (0.4) / noiter;
   nloptions technique=newrap;
run;
```

In a GLMM with parameter estimation by METHOD=LAPLACE or METHOD=QUAD the NOITER option also leads to an iteration history, since the fixed-effects estimates are part of the optimization and the PARMS statement places restrictions on only the covariance parameters.

Finally, the NOITER option can be useful if you want to obtain minimum variance quadratic unbiased estimates (with 0 priors), also known as MIVQUE0 estimates (Goodnight 1978b). Because MIVQUE0 estimates are starting values for covariance parameters—unless you provide *(value-list)* in the PARMS statement—the following statements produce MIVQUE0 mixed model estimates:

```
proc glimmix noprofile;
   class A B;
   model y = A;
   random int / subject=B;
   parms / noiter;
run;
```

PARMSDATA=*SAS-data-set*

PDATA=*SAS-data-set*

reads in covariance parameter values from a SAS data set. The data set should contain the numerical variable ESTIMATE or the numerical variables Covp1–Covp$q$, where q denotes the number of covariance parameters.

If the PARMSDATA= data set contains multiple sets of covariance parameters, the GLIMMIX procedure evaluates the initial objective function for each set and commences the optimization

step by using the set with the lowest function value as the starting values. For example, the following SAS statements request that the objective function be evaluated for three sets of initial values:

```
data data_covp;
  input covp1-covp4;
  datalines;
  180 200 170 1000
  170 190 160  900
  160 180 150  800
;
proc glimmix;
   class A B C rep mainEU smallEU;
   model yield = A|B|C;
   random rep mainEU smallEU;
   parms / pdata=data_covp;
run;
```

Each set comprises four covariance parameters.

The order of the observations in a data set with the numerical variable Estimate corresponds to the order of the covariance parameters in the "Covariance Parameter Estimates" table. In a GLM, the PARMSDATA= option can be used to set the starting value for the exponential family scale parameter. A grid search is not conducted for GLMs if you specify multiple values.

The PARMSDATA= data set must not contain missing values.

If the GLIMMIX procedure is processing the input data set in BY groups, you can add the BY variables to the PARMSDATA= data set. If this data set is sorted by the BY variables, the GLIMMIX procedure matches the covariance parameter values to the current BY group. If the PARMSDATA= data set does not contain all BY variables, the data set is processed in its entirety for every BY group and a message is written to the log. This enables you to provide a single set of starting values across BY groups, as in the following statements:

```
data data_covp;
  input covp1-covp4;
  datalines;
  180 200 170 1000
;
proc glimmix;
   class A B C rep mainEU smallEU;
   model yield = A|B|C;
   random rep mainEU smallEU;
   parms / pdata=data_covp;
   by year;
run;
```

The same set of starting values is used for each value of the year variable.

UPPERB=*value-list*

enables you to specify upper boundary constraints on the covariance parameters. The *value-list* specification is a list of numbers or missing values (.) separated by commas. You must list the numbers in the same order that PROC GLIMMIX uses for the *value-list* in the PARMS

statement, and each number corresponds to the upper boundary constraint. A missing value instructs PROC GLIMMIX to use its default constraint. If you do not specify numbers for all of the covariance parameters, PROC GLIMMIX assumes that the remaining ones are missing.

RANDOM Statement

RANDOM *random-effects* < / *options* > **;**

Using notation from "Notation for the Generalized Linear Mixed Model" on page 2083, the RANDOM statement defines the $\mathbf{Z}$ matrix of the mixed model, the random effects in the $\boldsymbol{\gamma}$ vector, the structure of $\mathbf{G}$, and the structure of $\mathbf{R}$.

The $\mathbf{Z}$ matrix is constructed exactly like the $\mathbf{X}$ matrix for the fixed effects, and the $\mathbf{G}$ matrix is constructed to correspond to the effects constituting $\mathbf{Z}$. The structures of $\mathbf{G}$ and $\mathbf{R}$ are defined by using the TYPE= option described on page 2193. The random effects can be classification or continuous effects, and multiple RANDOM statements are possible.

Some reserved keywords have special significance in the *random-effects* list. You can specify INTERCEPT (or INT) as a random effect to indicate the intercept. PROC GLIMMIX does not include the intercept in the RANDOM statement by default as it does in the MODEL statement. You can specify the _RESIDUAL_ keyword (or RESID, RESIDUAL, _RESID_) before the option slash (/) to indicate a residual-type (R-side) random component that defines the $\mathbf{R}$ matrix. Basically, the _RESIDUAL_ keyword takes the place of the *random-effect* if you want to specify R-side variances and covariance structures. These keywords take precedence over variables in the data set with the same name. If your data or the covariance structure requires that an effect is specified, you can use the RESIDUAL option to instruct the GLIMMIX procedure to model the R-side variances and covariances.

In order to add an overdispersion component to the variance function, simply specify a single residual random component. For example, the following statements fit a polynomial Poisson regression model with overdispersion. The variance function $a(\mu) = \mu$ is replaced by $\phi a(\mu)$:

```
proc glimmix;
   model count = x x*x / dist=poisson;
   random _residual_;
run;
```

Table 38.12 summarizes important options in the RANDOM statement. All options are subsequently discussed in alphabetical order.

Table 38.12 Summary of Important RANDOM Statement Options

Option	Description
Construction of Covariance Structure	
GCOORD=	determines coordinate association for G-side spatial structures with repeat levels
GROUP=	varies covariance parameters by groups
LDATA=	specifies a data set with coefficient matrices for TYPE= LIN

Table 38.12 *continued*

Option	Description
NOFULLZ	eliminates columns in **Z** corresponding to missing values
RESIDUAL	designates a covariance structure as R-side
SUBJECT=	identifies the subjects in the model
TYPE=	specifies the covariance structure
Mixed Model Smoothing	
KNOTINFO	displays spline knots
KNOTMAX=	specifies the upper limit for knot construction
KNOTMETHOD	specifies the method for constructing knots for radial smoother and penalized B-splines
KNOTMIN=	specifies the lower limit for knot construction
Statistical Output	
ALPHA=α	determines the confidence level $(1-\alpha)$
CL	requests confidence limits for predictors of random effects
G	displays the estimated **G** matrix
GC	displays the Cholesky root (lower) of the estimated **G** matrix
GCI	displays the inverse Cholesky root (lower) of the estimated **G** matrix
GCORR	displays the correlation matrix that corresponds to the estimated **G** matrix
GI	displays the inverse of the estimated **G** matrix
SOLUTION	displays solutions $\widehat{\boldsymbol{\gamma}}$ of the G-side random effects
V	displays blocks of the estimated **V** matrix
VC	displays the lower-triangular Cholesky root of blocks of the estimated **V** matrix
VCI	displays the inverse Cholesky root of blocks of the estimated **V** matrix
VCORR	displays the correlation matrix corresponding to blocks of the estimated **V** matrix
VI	displays the inverse of the blocks of the estimated **V** matrix

You can specify the following options in the RANDOM statement after a slash (/).

ALPHA=*number*

requests that a *t*-type confidence interval with confidence level $1-$ *number* be constructed for the predictors of G-side random effects in this statement. The value of *number* must be between 0 and 1; the default is 0.05. Specifying the ALPHA= option implies the CL option.

CL

requests that *t*-type confidence limits be constructed for each of the predictors of G-side random effects in this statement. The confidence level is 0.95 by default; this can be changed with the ALPHA= option. The CL option implies the SOLUTION option.

G

requests that the estimated **G** matrix be displayed for G-side random effects associated with this RANDOM statement. PROC GLIMMIX displays blanks for values that are 0.

GC

displays the lower-triangular Cholesky root of the estimated **G** matrix for G-side random effects.

GCI

displays the inverse Cholesky root of the estimated **G** matrix for G-side random effects.

GCOORD=LAST

GCOORD=FIRST

GCOORD=MEAN

determines how the GLIMMIX procedure associates coordinates for TYPE=SP() covariance structures with effect levels for G-side random effects. In these covariance structures, you specify one or more variables that identify the coordinates of a data point. The levels of classification variables, on the other hand, can occur multiple times for a particular subject. For example, in the following statements the same level of A can occur multiple times, and the associated values of x might be different:

```
proc glimmix;
   class A B;
   model y = B;
   random A / type=sp(pow)(x);
run;
```

The GCOORD=LAST option determines the coordinates for a level of the random effect from the last observation associated with the level. Similarly, the GCOORD=FIRST and GCOORD=MEAN options determine the coordinate from the first observation and from the average of the observations. Observations not used in the analysis are not considered in determining the first, last, or average coordinate. The default is GCOORD=LAST.

GCORR

displays the correlation matrix that corresponds to the estimated **G** matrix for G-side random effects.

GI

displays the inverse of the estimated **G** matrix for G-side random effects.

GROUP=*effect*

GRP=*effect*

identifies groups by which to vary the covariance parameters. Each new level of the grouping effect produces a new set of covariance parameters. Continuous variables and computed variables are permitted as group effects. PROC GLIMMIX does not sort by the values of the continuous variable; rather, it considers the data to be from a new group whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of groups and also prevents the production of a large "Class Levels Information" table.

Specifying a GROUP effect can greatly increase the number of estimated covariance parameters, which can adversely affect the optimization process.

KNOTINFO
displays the number and coordinates of the knots as determined by the KNOTMETHOD= option.

KNOTMAX=*number-list*
provides upper limits for the values of random effects used in the construction of knots for TYPE=RSMOOTH. The items in *number-list* correspond to the random effects of the radial smooth. If the KNOTMAX= option is not specified, or if the value associated with a particular random effect is set to missing, the maximum is based on the values in the data set for KNOTMETHOD=EQUAL or KNOTMETHOD=KDTREE, and is based on the values in the knot data set for KNOTMETHOD=DATA.

KNOTMETHOD=KDTREE<(*tree-options***)>**
KNOTMETHOD=EQUAL<(*number-list***)>**
KNOTMETHOD=DATA(*SAS-data-set***)**
determines the method of constructing knots for the radial smoother fit with the TYPE=RSMOOTH covariance structure and the TYPE=PSPLINE covariance structure.

Unless you select the TYPE=RSMOOTH or TYPE=PSPLINE covariance structure, the KNOTMETHOD= option has no effect. The default for TYPE=RSMOOTH is KNOTMETHOD=KDTREE. For TYPE=PSPLINE, only equally spaced knots are used and you can use the optional *numberlist* argument of KNOTMETHOD=EQUAL to determine the number of interior knots for TYPE=PSPLINE.

Knot Construction for TYPE=RSMOOTH

PROC GLIMMIX fits a low-rank smoother, meaning that the number of knots is considerably less than the number of observations. By default, PROC GLIMMIX determines the knot locations based on the vertices of a *k-d* tree (Friedman, Bentley, and Finkel 1977; Cleveland and Grosse 1991). The *k-d* tree is a tree data structure that is useful for efficiently determining the m nearest neighbors of a point. The *k-d* tree also can be used to obtain a grid of points that adapts to the configuration of the data. The process starts with a hypercube that encloses the values of the random effects. The space is then partitioned recursively by splitting cells at the median of the data in the cell for the random effect. The procedure is repeated for all cells that contain more than a specified number of points, b. The value b is called the bucket size.

The *k-d* tree is thus a division of the data into cells such that cells representing leaf nodes contain at most b values. You control the building of the *k-d* tree through the BUCKET= *tree-option*. You control the construction of knots from the cell coordinates of the tree with the other options as follows.

BUCKET=*number*
determines the bucket size b. A larger bucket size will result in fewer knots. For *k-d* trees in more than one dimension, the correspondence between bucket size and number of knots is difficult to determine. It depends on the data configuration and on other

suboptions. In the multivariate case, you might need to try out different bucket sizes to obtain the desired number of knots. The default value of *number* is 4 for univariate trees (a single random effect) and $\lfloor 0.1n \rfloor$ in the multidimensional case.

KNOTTYPE=*type*
specifies whether the knots are based on vertices of the tree cells or the centroid. The two possible values of *type* are VERTEX and CENTER. The default is KNOTTYPE=VERTEX. For multidimensional smoothing, such as smoothing across irregularly shaped spatial domains, the KNOTTYPE=CENTER option is useful to move knot locations away from the bounding hypercube toward the convex hull.

NEAREST
specifies that knot coordinates are the coordinates of the nearest neighbor of either the centroid or vertex of the cell, as determined by the KNOTTYPE= suboption.

TREEINFO
displays details about the construction of the *k-d* tree, such as the cell splits and the split values.

See the section "Knot Selection" on page 2251 for a detailed example of how the specification of the bucket size translates into the construction of a *k-d* tree and the spline knots.

The KNOTMETHOD=EQUAL option enables you to define a regular grid of knots. By default, PROC GLIMMIX constructs 10 knots for one-dimensional smooths and 5 knots in each dimension for smoothing in higher dimensions. You can specify a different number of knots with the optional *number-list*. Missing values in the *number-list* are replaced with the default values. A minimum of two knots in each dimension is required. For example, the following statements use a rectangular grid of 35 knots, five knots for x1 combined with seven knots for x2:

```
proc glimmix;
   model y=;
   random x1 x2 / type=rsmooth knotmethod=equal(5 7);
run;
```

When you use the NOFIT option in the PROC GLIMMIX statement, the GLIMMIX procedure computes the knots but does not fit the model. This can be useful if you want to compare knot selections with different suboptions of KNOTMETHOD=KDTREE. Suppose you want to determine the number of knots based on a particular bucket size. The following statements compute and display the knots in a bivariate smooth, constructed from nearest neighbors of the vertices of a *k-d* tree with bucket size 10:

```
proc glimmix nofit;
   model y = Latitude Longitude;
   random Latitude Longitude / type=rsmooth
                  knotmethod=kdtree(knottype=vertex
                  nearest bucket=10) knotinfo;
run;
```

You can specify a data set that contains variables whose values give the knot coordinates with the KNOTMETHOD=DATA option. The data set must contain numeric variables with the same name as the radial smoothing *random-effects*. PROC GLIMMIX uses only the

unique knot coordinates in the knot data set. This option is useful to provide knot coordinates different from those that can be produced from a *k-d* tree. For example, in spatial problems where the domain is irregularly shaped, you might want to determine knots by a space-filling algorithm. The following SAS statements invoke the OPTEX procedure to compute 45 knots that uniformly cover the convex hull of the data locations (see Chapter 30, "Introduction to the OPTEX Procedure," (*SAS/QC User's Guide*) and Chapter 31, "Details of the OPTEX Procedure," (*SAS/QC User's Guide*) for details about the OPTEX procedure).

```
proc optex coding=none;
   model latitude longitude / noint;
   generate n=45 criterion=u method=m_fedorov;
   output out=knotdata;
run;
proc glimmix;
   model y = Latitude Longitude;
   random Latitude Longitude / type=rsmooth
                    knotmethod=data(knotdata);
run;
```

Knot Construction for TYPE=PSPLINE

Only evenly spaced knots are supported when you fit penalized B-splines with the GLIMMIX procedure. For the TYPE=PSPLINE covariance structure, the *numberlist* argument specifies the number m of interior knots, the default is $m = 10$. Suppose that $x_{(1)}$ and $x_{(n)}$ denote the smallest and largest values, respectively. For a B-spline of degree d (de Boor 2001), the interior knots are supplemented with d exterior knots below $x_{(1)}$ and $\max\{1, d\}$ exterior knots above $x_{(n)}$. PROC GLIMMIX computes the location of these $m + d + \max\{1, d\}$ knots as follows. Let $\delta_x = (x_{(n)} - x_{(1)})/(m + 1)$, then interior knots are placed at

$$x_{(1)} + j\delta_x, \quad j = 1, \cdots, m$$

The exterior knots are also evenly spaced with step size δ_x and start at $x_{(1)} \pm 100$ times the machine epsilon. At least one interior knot is required.

KNOTMIN=*number-list*

provides lower limits for the values of random effects used in the construction of knots for TYPE=RSMOOTH. The items in *number-list* correspond to the random effects of the radial smooth. If the KNOTMIN= option is not specified, or if the value associated with a particular random effect is set to missing, the minimum is based on the values in the data set for KNOTMETHOD=EQUAL or KNOTMETHOD=KDTREE, and is based on the values in the knot data set for KNOTMETHOD=DATA.

LDATA=*SAS-data-set*

reads the coefficient matrices $\mathbf{A}_1, \cdots, \mathbf{A}_q$ for the TYPE=LIN(q) option. You can specify the LDATA= data set in a sparse or dense form. In the sparse form the data set must contain the numeric variables Parm, Row, Col, and Value. The Parm variable contains the indices $i = 1, \cdots, q$ of the $\mathbf{A}_i$ matrices. The Row and Col variables identify the position within a matrix and the Value variable contains the matrix element. Values not specified for a particular row and column are set to zero. Missing values are allowed in the Value column of the

LDATA= data set; these values are also replaced by zeros. The sparse form is particularly useful if the **A** matrices have only a few nonzero elements.

In the dense form the LDATA= data set contains the numeric variables Parm and Row (with the same function as above), in addition to the numeric variables Col1–Col$q$. If you omit one or more of the Col1–Col$q$ variables from the data set, zeros are assumed for the respective rows and columns of the **A** matrix. Missing values for Col1–Col$q$ are ignored in the dense form.

The GLIMMIX procedure assumes that the matrices $\mathbf{A}_1, \cdots, \mathbf{A}_q$ are symmetric. In the sparse LDATA= form you do not need to specify off-diagonal elements in position (i, j) and (j, i). One of them is sufficient. Row-column indices are converted in both storage forms into positions in lower triangular storage. If you specify multiple values in row ($\max\{i, j\}$ and column $\min\{i, j\}$) of a particular matrix, only the last value is used. For example, assume you are specifying elements of a 4×4 matrix. The lower triangular storage of matrix $\mathbf{A}_3$ defined by

```
data ldata;
  input parm row col value;
  datalines;
3  2  1  2
3  1  2  5
;
```

is

$$\begin{bmatrix} 0 & & & \\ 5 & 0 & & \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

NOFULLZ

eliminates the columns in **Z** corresponding to missing levels of random effects involving CLASS variables. By default, these columns are included in **Z**. It is sufficient to specify the NOFULLZ option on any G-side RANDOM statement.

RESIDUAL

RSIDE

specifies that the random effects listed in this statement be R-side effects. You use the RESIDUAL option in the RANDOM statement if the nature of the covariance structure requires you to specify an effect. For example, if it is necessary to order the columns of the R-side AR(1) covariance structure by the time variable, you can use the RESIDUAL option as in the following statements:

```
class time id;
random time / subject=id type=ar(1) residual;
```

SOLUTION

S

requests that the solution $\widehat{\boldsymbol{\gamma}}$ for the random-effects parameters be produced, if the statement defines G-side random effects.

The numbers displayed in the Std Err Pred column of the "Solution for Random Effects" table are not the standard errors of the $\widehat{\boldsymbol{\gamma}}$ displayed in the Estimate column; rather, they are the square roots of the prediction errors $\widehat{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i$, where $\widehat{\boldsymbol{\gamma}}_i$ is the predictor of the ith random effect and $\boldsymbol{\gamma}_i$ is the ith random effect. In pseudo-likelihood methods that are based on linearization, these EBLUPs are the estimated best linear unbiased predictors in the linear mixed pseudo-model. In models fit by maximum likelihood by using the Laplace approximation or by using adaptive quadrature, the SOLUTION option displays the empirical Bayes estimates (EBE) of $\boldsymbol{\gamma}_i$.

SUBJECT=*effect*

SUB=*effect*

identifies the subjects in your generalized linear mixed model. Complete independence is assumed across subjects. Specifying a subject effect is equivalent to nesting all other effects in the RANDOM statement within the subject effect.

Continuous variables and computed variables are permitted with the SUBJECT= option. PROC GLIMMIX does not sort by the values of the continuous variable but considers the data to be from a new subject whenever the value of the continuous variable changes from the previous observation. Using a continuous variable can decrease execution time for models with a large number of subjects and also prevents the production of a large "Class Levels Information" table.

TYPE=*covariance-structure*

specifies the covariance structure of $\mathbf{G}$ for G-side effects and the covariance structure of $\mathbf{R}$ for R-side effects.

Although a variety of structures are available, many applications call for either simple diagonal (TYPE=VC) or unstructured covariance matrices. The TYPE=VC (variance components) option is the default structure, and it models a different variance component for each random effect. It is recommended to model unstructured covariance matrices in terms of their Cholesky parameterization (TYPE=CHOL) rather than TYPE=UN.

If you want different covariance structures in different parts of $\mathbf{G}$, you must use multiple RANDOM statements with different TYPE= options.

Valid values for *covariance-structure* are as follows. Examples are shown in Table 38.14.

The variances and covariances in the formulas that follow in the TYPE= descriptions are expressed in terms of generic random variables ξ_i and ξ_j. They represent the G-side random effects or the residual random variables for which the $\mathbf{G}$ or $\mathbf{R}$ matrices are constructed.

ANTE(1)

specifies a first-order ante-dependence structure (Kenward 1987; Patel 1991) parameterized in terms of variances and correlation parameters. If t ordered random variables $\xi_1, \cdots, \xi_t$ have a first-order ante-dependence structure, then each ξ_j, $j > 1$, is independent of all other $\xi_k, k < j$, given ξ_{j-1}. This Markovian structure is characterized by its inverse variance matrix, which is tridiagonal. Parameterizing an ANTE(1) structure for a random vector of size t requires $2t-1$ parameters: variances $\sigma_1^2, \cdots, \sigma_t^2$ and $t-1$ correlation parameters $\rho_1, \cdots, \rho_{t-1}$. The covariances among random variables ξ_i

and ξ_j are then constructed as

$$\text{Cov}\left[\xi_i, \xi_j\right] = \sqrt{\sigma_i^2 \sigma_j^2} \prod_{k=i}^{j-1} \rho_k$$

PROC GLIMMIX constrains the correlation parameters to satisfy $|\rho_k| < 1, \forall k$. For variable-order ante-dependence models see Macchiavelli and Arnold (1994).

AR(1)

specifies a first-order autoregressive structure,

$$\text{Cov}\left[\xi_i, \xi_j\right] = \sigma^2 \rho^{|i^* - j^*|}$$

The values i^* and j^* are derived for the ith and jth observations, respectively, and are not necessarily the observation numbers. For example, in the following statements the values correspond to the class levels for the time effect of the ith and jth observation within a particular subject:

```
proc glimmix;
  class time patient;
  model y = x x*x;
  random time / sub=patient type=ar(1);
run;
```

PROC GLIMMIX imposes the constraint $|\rho| < 1$ for stationarity.

ARH(1)

specifies a heterogeneous first-order autoregressive structure,

$$\text{Cov}\left[\xi_i, \xi_j\right] = \sqrt{\sigma_i^2 \sigma_j^2}\, \rho^{|i^* - j^*|}$$

with $|\rho| < 1$. This covariance structure has the same correlation pattern as the TYPE=AR(1) structure, but the variances are allowed to differ.

ARMA(1,1)

specifies the first-order autoregressive moving-average structure,

$$\text{Cov}\left[\xi_i, \xi_j\right] = \begin{cases} \sigma^2 & i = j \\ \sigma^2 \gamma \rho^{|i^* - j^*| - 1} & i \neq j \end{cases}$$

Here, ρ is the autoregressive parameter, γ models a moving-average component, and σ^2 is a scale parameter. In the notation of Fuller (1976, p. 68), $\rho = \theta_1$ and

$$\gamma = \frac{(1 + b_1\theta_1)(\theta_1 + b_1)}{1 + b_1^2 + 2b_1\theta_1}$$

The example in Table 38.14 and $|b_1| < 1$ imply that

$$b_1 = \frac{\beta - \sqrt{\beta^2 - 4\alpha^2}}{2\alpha}$$

where $\alpha = \gamma - \rho$ and $\beta = 1 + \rho^2 - 2\gamma\rho$. PROC GLIMMIX imposes the constraints $|\rho| < 1$ and $|\gamma| < 1$ for stationarity, although for some values of ρ and γ in this region the resulting covariance matrix is not positive definite. When the estimated value of ρ becomes negative, the computed covariance is multiplied by $\cos(\pi d_{ij})$ to account for the negativity.

CHOL<(*q*)>

specifies an unstructured variance-covariance matrix parameterized through its Cholesky root. This parameterization ensures that the resulting variance-covariance matrix is at least positive semidefinite. If all diagonal values are nonzero, it is positive definite. For example, a 2×2 unstructured covariance matrix can be written as

$$\text{Var}[\boldsymbol{\xi}] = \begin{bmatrix} \theta_1 & \theta_{12} \\ \theta_{12} & \theta_2 \end{bmatrix}$$

Without imposing constraints on the three parameters, there is no guarantee that the estimated variance matrix is positive definite. Even if θ_1 and θ_2 are nonzero, a large value for θ_{12} can lead to a negative eigenvalue of $\text{Var}[\boldsymbol{\xi}]$. The Cholesky root of a positive definite matrix $\mathbf{A}$ is a lower triangular matrix $\mathbf{C}$ such that $\mathbf{CC}' = \mathbf{A}$. The Cholesky root of the above 2×2 matrix can be written as

$$\mathbf{C} = \begin{bmatrix} \alpha_1 & 0 \\ \alpha_{12} & \alpha_2 \end{bmatrix}$$

The elements of the unstructured variance matrix are then simply $\theta_1 = \alpha_1^2$, $\theta_{12} = \alpha_1\alpha_{12}$, and $\theta_2 = \alpha_{12}^2 + \alpha_2^2$. Similar operations yield the generalization to covariance matrices of higher orders.

For example, the following statements model the covariance matrix of each subject as an unstructured matrix:

```
proc glimmix;
   class sub;
   model y = x;
   random _residual_ / subject=sub type=un;
run;
```

The next set of statements accomplishes the same, but the estimated **R** matrix is guaranteed to be nonnegative definite:

```
proc glimmix;
   class sub;
   model y = x;
   random _residual_ / subject=sub type=chol;
run;
```

The GLIMMIX procedure constrains the diagonal elements of the Cholesky root to be positive. This guarantees a unique solution when the matrix is positive definite.

The optional order parameter $q > 0$ determines how many bands below the diagonal are modeled. Elements in the lower triangular portion of $\mathbf{C}$ in bands higher than q are

set to zero. If you consider the resulting covariance matrix $\mathbf{A} = \mathbf{C}\mathbf{C}'$, then the order parameter has the effect of zeroing all off-diagonal elements that are at least q positions away from the diagonal.

Because of its good computational and statistical properties, the Cholesky root parameterization is generally recommended over a completely unstructured covariance matrix (TYPE=UN). However, it is computationally slightly more involved.

CS

specifies the compound-symmetry structure, which has constant variance and constant covariance

$$\text{Cov}\left[\xi_i, \xi_j\right] = \begin{cases} \phi + \sigma & i = j \\ \sigma & i \neq j \end{cases}$$

The compound symmetry structure arises naturally with nested random effects, such as when subsampling error is nested within experimental error. The models constructed with the following two sets of GLIMMIX statements have the same marginal variance matrix, provided σ is positive:

```
proc glimmix;
   class block A;
   model y = block A;
   random block*A / type=vc;
run;


proc glimmix;
   class block A;
   model y = block A;
   random _residual_ / subject=block*A
                       type=cs;
run;
```

In the first case, the block*A random effect models the G-side experimental error. Because the distribution defaults to the normal, the $\mathbf{R}$ matrix is of form $\phi\mathbf{I}$ (see Table 38.15), and ϕ is the subsampling error variance. The marginal variance for the data from a particular experimental unit is thus $\sigma^2_{b*a}\mathbf{J} + \phi\mathbf{I}$. This matrix is of compound symmetric form.

Hierarchical random assignments or selections, such as subsampling or split-plot designs, give rise to compound symmetric covariance structures. This implies exchangeability of the observations on the subunit, leading to constant correlations between the observations. Compound symmetric structures are thus usually not appropriate for processes where correlations decline according to some metric, such as spatial and temporal processes.

Note that R-side compound-symmetry structures do not impose any constraint on σ. You can thus use an R-side TYPE=CS structure to emulate a variance-component model with unbounded estimate of the variance component.

CSH

specifies the heterogeneous compound-symmetry structure, which is an equi-correlation structure but allows for different variances

$$\mathrm{Cov}\left[\xi_i, \xi_j\right] = \begin{cases} \sqrt{\sigma_i^2 \sigma_j^2} & i = j \\ \rho\sqrt{\sigma_i^2 \sigma_j^2} & i \neq j \end{cases}$$

FA(*q*)

specifies the factor-analytic structure with q factors (Jennrich and Schluchter 1986). This structure is of the form $\mathbf{\Lambda}\mathbf{\Lambda}' + \mathbf{D}$, where $\mathbf{\Lambda}$ is a $t \times q$ rectangular matrix and $\mathbf{D}$ is a $t \times t$ diagonal matrix with t different parameters. When $q > 1$, the elements of $\mathbf{\Lambda}$ in its upper-right corner (that is, the elements in the ith row and jth column for $j > i$) are set to zero to fix the rotation of the structure.

FA0(*q*)

specifies a factor-analytic structure with q factors of the form $\mathrm{Var}[\boldsymbol{\xi}] = \mathbf{\Lambda}\mathbf{\Lambda}'$, where $\mathbf{\Lambda}$ is a $t \times q$ rectangular matrix and t is the dimension of $\mathbf{Y}$. When $q > 1$, $\mathbf{\Lambda}$ is a lower triangular matrix. When $q < t$—that is, when the number of factors is less than the dimension of the matrix—this structure is nonnegative definite but not of full rank. In this situation, you can use it to approximate an unstructured covariance matrix.

HF

specifies a covariance structure that satisfies the general Huynh-Feldt condition (Huynh and Feldt 1970). For a random vector with t elements, this structure has $t + 1$ positive parameters and covariances

$$\mathrm{Cov}\left[\xi_i, \xi_j\right] = \begin{cases} \sigma_i^2 & i = j \\ 0.5(\sigma_i^2 + \sigma_j^2) - \lambda & i \neq j \end{cases}$$

A covariance matrix $\mathbf{\Sigma}$ generally satisfies the Huynh-Feldt condition if it can be written as $\mathbf{\Sigma} = \boldsymbol{\tau}\mathbf{1}' + \mathbf{1}\boldsymbol{\tau}' + \lambda\mathbf{I}$. The preceding parameterization chooses $\tau_i = 0.5(\sigma_i^2 - \lambda)$. Several simpler covariance structures give rise to covariance matrices that also satisfy the Huynh-Feldt condition. For example, TYPE=CS, TYPE=VC, and TYPE=UN(1) are nested within TYPE=HF. You can use the COVTEST statement to test the HF structure against one of these simpler structures. Note also that the HF structure is nested within an unstructured covariance matrix.

The TYPE=HF covariance structure can be sensitive to the choice of starting values and the default MIVQUE(0) starting values can be poor for this structure; you can supply your own starting values with the PARMS statement.

LIN(*q*)

specifies a general linear covariance structure with q parameters. This structure consists of a linear combination of known matrices that you input with the LDATA= option. Suppose that you want to model the covariance of a random vector of length t, and further suppose that $\mathbf{A}_1, \cdots, \mathbf{A}_q$ are symmetric $(t \times t)$ matrices constructed from the information in the LDATA= data set. Then,

$$\mathrm{Cov}\left[\xi_i, \xi_j\right] = \sum_{k=1}^{q} \theta_k [\mathbf{A}_k]_{ij}$$

where $[\mathbf{A}_k]_{ij}$ denotes the element in row i, column j of matrix $\mathbf{A}_k$.

Linear structures are very flexible and general. You need to exercise caution to ensure that the variance matrix is positive definite. Note that PROC GLIMMIX does not impose boundary constraints on the parameters $\theta_1, \cdots, \theta_k$ of a general linear covariance structure. For example, if classification variable A has 6 levels, the following statements fit a variance component structure for the random effect without boundary constraints:

```
data ldata;
  retain parm 1 value 1;
  do row=1 to 6; col=row; output; end;
run;

proc glimmix data=MyData;
  class A B;
  model Y = B;
  random A / type=lin(1) ldata=ldata;
run;
```

PSPLINE<(*options*)>

requests that PROC GLIMMIX form a B-spline basis and fits a penalized B-spline (P-spline, Eilers and Marx 1996) with random spline coefficients. This covariance structure is available only for G-side random effects and only a single continuous random effect can be specified with TYPE=PSPLINE. As for TYPE=RSMOOTH, PROC GLIMMIX forms a modified $\mathbf{Z}$ matrix and fits a mixed model in which the random variables associated with the columns of $\mathbf{Z}$ are independent with a common variance. The $\mathbf{Z}$ matrix is constructed as follows.

Denote as $\widetilde{\mathbf{Z}}$ the $(n \times K)$ matrix of B-splines of degree d and denote as $\mathbf{D}_r$ the $(K - r \times K)$ matrix of rth-order differences. For example, for $K = 5$,

$$\mathbf{D}_1 = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

$$\mathbf{D}_2 = \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix}$$

$$\mathbf{D}_3 = \begin{bmatrix} 1 & -3 & 3 & -1 & 0 \\ 0 & 1 & -3 & 3 & -1 \end{bmatrix}$$

Then, the $\mathbf{Z}$ matrix used in fitting the mixed model is the $(n \times K - r)$ matrix

$$\mathbf{Z} = \widetilde{\mathbf{Z}}(\mathbf{D}_r'\mathbf{D}_r)^{-}\mathbf{D}_r'$$

The construction of the B-spline knots is controlled with the KNOTMETHOD=EQUAL(m) option and the DEGREE=d suboption of TYPE=PSPLINE. The total number of knots equals the number m of equally spaced interior knots plus d knots at the

low end and $\max\{1, d\}$ knots at the high end. The number of columns in the B-spline basis equals $K = m + d + 1$. By default, the interior knots exclude the minimum and maximum of the random-effect values and are based on $m - 1$ equally spaced intervals. Suppose $x_{(1)}$ and $x_{(n)}$ are the smallest and largest random-effect values; then interior knots are placed at

$$x_{(1)} + j(x_{(n)} - x_{(1)})/(m + 1), \quad j = 1, \cdots, m$$

In addition, d evenly spaced exterior knots are placed below $x_{(1)}$ and $\max\{d, 1\}$ exterior knots are placed above $x_{(m)}$. The exterior knots are evenly spaced and start at $x_{(1)} \pm 100$ times the machine epsilon. For example, based on the defaults $d = 3$, $r = 3$, the following statements lead to 26 total knots and 21 columns in $\mathbf{Z}$, $m = 20$, $K = m + d + 1 = 24$, $K - r = 21$:

```
proc glimmix;
   model y = x;
   random x / type=pspline knotmethod=equal(20);
run;
```

Details about the computation and properties of B-splines can be found in de Boor (2001).

You can extend or limit the range of the knots with the KNOTMIN= and KNOTMAX= options. Table 38.13 lists some of the parameters that control this covariance type and their relationships.

Table 38.13 P-Spline Parameters

Parameter	Description
d	Degree of B-spline, default $d = 3$
r	Order of differencing in construction of $\mathbf{D}_r$, default $r = 3$
m	Number of interior knots, default $m = 10$
$m + d + \max\{1, d\}$	Total number of knots
$K = m + d + 1$	Number of columns in B-spline basis
$K - r$	Number of columns in $\mathbf{Z}$

You can specify the following *options* for TYPE=PSPLINE:

DEGREE=d specifies the degree of the B-spline. The default is $d = 3$.

DIFFORDER=r specifies the order of the differencing matrix $\mathbf{D}_r$. The default and maximum is $r = 3$.

RSMOOTH<(*m* | NOLOG)>

specifies a radial smoother covariance structure for G-side random effects. This results in an approximate low-rank thin-plate spline where the smoothing parameter is obtained by the estimation method selected with the METHOD= option of the PROC GLIMMIX statement. The smoother is based on the automatic smoother in Ruppert, Wand, and Carroll (2003, Chapter 13.4–13.5), but with a different method of selecting the spline knots. See the section "Radial Smoothing Based on Mixed Models"

on page 2250 for further details about the construction of the smoother and the knot selection.

Radial smoothing is possible in one or more dimensions. A univariate smoother is obtained with a single random effect, while multiple random effects in a RANDOM statement yield a multivariate smoother. Only continuous random effects are permitted with this covariance structure. If n_r denotes the number of continuous random effects in the RANDOM statement, then the covariance structure of the random effects $\boldsymbol{\gamma}$ is determined as follows. Suppose that $\mathbf{z}_i$ denotes the vector of random effects for the ith observation. Let $\boldsymbol{\tau}_k$ denote the $(n_r \times 1)$ vector of knot coordinates, $k = 1, \cdots, K$, and K is the total number of knots. The Euclidean distance between the knots is computed as

$$d_{kp} = ||\boldsymbol{\tau}_k - \boldsymbol{\tau}_p|| = \sqrt{\sum_{j=1}^{n_r} (\tau_{jk} - \tau_{jp})^2}$$

and the distance between knots and effects is computed as

$$h_{ik} = ||\mathbf{z}_i - \boldsymbol{\tau}_k|| = \sqrt{\sum_{j=1}^{n_r} (z_{ij} - \tau_{jk})^2}$$

The $\mathbf{Z}$ matrix for the GLMM is constructed as

$$\mathbf{Z} = \widetilde{\mathbf{Z}}\boldsymbol{\Omega}^{-1/2}$$

where the $(n \times K)$ matrix $\widetilde{\mathbf{Z}}$ has typical element

$$[\widetilde{\mathbf{Z}}]_{ik} = \begin{cases} h_{ik}^p & n_r \text{ odd} \\ h_{ik}^p \log\{h_{ik}\} & n_r \text{ even} \end{cases}$$

and the $(K \times K)$ matrix $\boldsymbol{\Omega}$ has typical element

$$[\boldsymbol{\Omega}]_{kp} = \begin{cases} d_{kp}^p & n_r \text{ odd} \\ d_{kp}^p \log\{d_{kp}\} & n_r \text{ even} \end{cases}$$

The exponent in these expressions equals $p = 2m - n_r$, where the optional value m corresponds to the derivative penalized in the thin-plate spline. A larger value of m will yield a smoother fit. The GLIMMIX procedure requires $p > 0$ and chooses by default $m = 2$ if $n_r < 3$ and $m = (n_r + 1)/2$ otherwise. The NOLOG option removes the $\log\{h_{ik}\}$ and $\log\{d_{kp}\}$ terms from the computation of the $\widetilde{\mathbf{Z}}$ and $\boldsymbol{\Omega}$ matrices when n_r is even; this yields invariance under rescaling of the coordinates.

Finally, the components of $\boldsymbol{\gamma}$ are assumed to have equal variance σ_r^2. The "smoothing parameter" λ of the low-rank spline is related to the variance components in the model, $\lambda^2 = f(\phi, \sigma_r^2)$. See Ruppert, Wand, and Carroll (2003) for details. If the conditional distribution does not provide a scale parameter ϕ, you can add a single R-side residual parameter.

The knot selection is controlled with the KNOTMETHOD= option. The GLIMMIX procedure selects knots automatically based on the vertices of a *k-d* tree or reads knots from a data set that you supply. See the section "Radial Smoothing Based on Mixed Models" on page 2250 for further details on radial smoothing in the GLIMMIX procedure and its connection to a mixed model formulation.

SIMPLE

is an alias for TYPE=VC.

SP(EXP)(*c-list*)

models an exponential spatial or temporal covariance structure, where the covariance between two observations depends on a distance metric d_{ij}. The *c-list* contains the names of the numeric variables used as coordinates to determine distance. For a stochastic process in R^k, there are k elements in *c-list*. If the $(k \times 1)$ vectors of coordinates for observations i and j are $\mathbf{c}_i$ and $\mathbf{c}_j$, then PROC GLIMMIX computes the Euclidean distance

$$d_{ij} = ||\mathbf{c}_i - \mathbf{c}_j|| = \sqrt{\sum_{m=1}^{k} (c_{mi} - c_{mj})^2}$$

The covariance between two observations is then

$$\text{Cov}\left[\xi_i, \xi_j\right] = \sigma^2 \exp\{-d_{ij}/\alpha\}$$

The parameter α is *not* what is commonly referred to as the range parameter in geostatistical applications. The practical range of a (second-order stationary) spatial process is the distance $d^{(p)}$ at which the correlations fall below 0.05. For the SP(EXP) structure, this distance is $d^{(p)} = 3\alpha$. PROC GLIMMIX constrains α to be positive.

SP(GAU)(*c-list*)

models a gaussian covariance structure,

$$\text{Cov}\left[\xi_i, \xi_j\right] = \sigma^2 \exp\{-d_{ij}^2/\alpha^2\}$$

See TYPE=SP(EXP) for the computation of the distance d_{ij}. The parameter α is related to the range of the process as follows. If the practical range $d^{(p)}$ is defined as the distance at which the correlations fall below 0.05, then $d^{(p)} = \sqrt{3}\alpha$. PROC GLIMMIX constrains α to be positive. See TYPE=SP(EXP) for the computation of the distance d_{ij} from the variables specified in *c-list*.

SP(MAT)(*c-list*)

models a covariance structure in the Matérn class of covariance functions (Matérn 1986). The covariance is expressed in the parameterization of Handcock and Stein (1993) and Handcock and Wallis (1994); it can be written as

$$\text{Cov}\left[\xi_i, \xi_j\right] = \sigma^2 \frac{1}{\Gamma(\nu)} \left(\frac{d_{ij}\sqrt{\nu}}{\rho}\right)^{\nu} 2K_\nu \left(\frac{2d_{ij}\sqrt{\nu}}{\rho}\right)$$

The function K_ν is the modified Bessel function of the second kind of (real) order $\nu > 0$. The smoothness (continuity) of a stochastic process with covariance function in the Matérn class increases with ν. This class thus enables data-driven estimation of the smoothness properties of the process. The covariance is identical to the exponential model for $\nu = 0.5$ (TYPE=SP(EXP)(*c-list*)), while for $\nu = 1$ the model advocated by Whittle (1954) results. As $\nu \to \infty$, the model approaches the gaussian covariance structure (TYPE=SP(GAU)(*c-list*)).

Note that the MIXED procedure offers covariance structures in the Matérn class in two parameterizations, TYPE=SP(MATERN) and TYPE=SP(MATHSW). The TYPE=SP(MAT) in the GLIMMIX procedure is equivalent to TYPE=SP(MATHSW) in the MIXED procedure.

Computation of the function K_ν and its derivatives is numerically demanding; fitting models with Matérn covariance structures can be time-consuming. Good starting values are essential.

SP(POW)(*c-list*)

models a power covariance structure,

$$\text{Cov}\left[\xi_i, \xi_j\right] = \sigma^2 \rho^{d_{ij}}$$

where $\rho \geq 0$. This is a reparameterization of the exponential structure, TYPE=SP(EXP). Specifically, $\log\{\rho\} = -1/\alpha$. See TYPE=SP(EXP) for the computation of the distance d_{ij} from the variables specified in *c-list*. When the estimated value of ρ becomes negative, the computed covariance is multiplied by $\cos(\pi d_{ij})$ to account for the negativity.

SP(POWA)(*c-list*)

models an anisotropic power covariance structure in k dimensions, provided that the coordinate list *c-list* has k elements. If c_{im} denotes the coordinate for the ith observation of the mth variable in *c-list*, the covariance between two observations is given by

$$\text{Cov}\left[\xi_i, \xi_j\right] = \sigma^2 \rho_1^{|c_{i1}-c_{j1}|} \rho_2^{|c_{i2}-c_{j2}|} \ldots \rho_k^{|c_{ik}-c_{jk}|}$$

Note that for $k = 1$, TYPE=SP(POWA) is equivalent to TYPE=SP(POW), which is itself a reparameterization of TYPE=SP(EXP). When the estimated value of ρ_m becomes negative, the computed covariance is multiplied by $\cos(\pi|c_{im} - c_{jm}|)$ to account for the negativity.

SP(SPH)(*c-list*)

models a spherical covariance structure,

$$\text{Cov}\left[\xi_i, \xi_j\right] = \begin{cases} \sigma^2 \left\{1 - \frac{3d_{ij}}{2\alpha} + \frac{1}{2}\left(\frac{d_{ij}}{\alpha}\right)^3\right\} & d_{ij} \leq \alpha \\ 0 & d_{ij} > \alpha \end{cases}$$

The spherical covariance structure has a true range parameter. The covariances between observations are exactly zero when their distance exceeds α. See TYPE=SP(EXP) for the computation of the distance d_{ij} from the variables specified in *c-list*.

TOEP

models a Toeplitz covariance structure. This structure can be viewed as an autoregressive structure with order equal to the dimension of the matrix,

$$\text{Cov}\left[\xi_i, \xi_j\right] = \begin{cases} \sigma^2 & i = j \\ \sigma_{|i-j|} & i \neq j \end{cases}$$

TOEP(*q*)

specifies a banded Toeplitz structure,

$$\mathrm{Cov}\left[\xi_i, \xi_j\right] = \begin{cases} \sigma^2 & i = j \\ \sigma_{|i-j|} & |i-j| < q \end{cases}$$

This can be viewed as a moving-average structure with order equal to $q - 1$. The specification TYPE=TOEP(1) is the same as $\sigma^2\mathbf{I}$, and it can be useful for specifying the same variance component for several effects.

TOEPH<(*q*)>

models a Toeplitz covariance structure. The correlations of this structure are banded as the TOEP or TOEP(q) structures, but the variances are allowed to vary:

$$\mathrm{Cov}\left[\xi_i, \xi_j\right] = \begin{cases} \sigma_i^2 & i = j \\ \rho_{|i-j|}\sqrt{\sigma_i^2\sigma_j^2} & i \neq j \end{cases}$$

The correlation parameters satisfy $|\rho_{|i-j|}| < 1$. If you specify the optional value q, the correlation parameters with $|i - j| \geq q$ are set to zero, creating a banded correlation structure. The specification TYPE=TOEPH(1) results in a diagonal covariance matrix with heterogeneous variances.

UN<(*q*)>

specifies a completely general (unstructured) covariance matrix parameterized directly in terms of variances and covariances,

$$\mathrm{Cov}\left[\xi_i, \xi_j\right] = \sigma_{ij}$$

The variances are constrained to be nonnegative, and the covariances are unconstrained. This structure is not constrained to be nonnegative definite in order to avoid nonlinear constraints; however, you can use the TYPE=CHOL structure if you want this constraint to be imposed by a Cholesky factorization. If you specify the order parameter q, then PROC GLIMMIX estimates only the first q bands of the matrix, setting elements in all higher bands equal to 0.

UNR<(*q*)>

specifies a completely general (unstructured) covariance matrix parameterized in terms of variances and correlations,

$$\mathrm{Cov}\left[\xi_i, \xi_j\right] = \sigma_i\sigma_j\rho_{ij}$$

where σ_i denotes the standard deviation and the correlation ρ_{ij} is zero when $i = j$ and when $|i - j| \geq q$, provided the order parameter q is given. This structure fits the same model as the TYPE=UN(q) option, but with a different parameterization. The ith variance parameter is σ_i^2. The parameter ρ_{ij} is the correlation between the ith and jth measurements; it satisfies $|\rho_{ij}| < 1$. If you specify the order parameter q, then PROC GLIMMIX estimates only the first q bands of the matrix, setting all higher bands equal to zero.

VC

specifies standard variance components and is the default structure for both G-side and R-side covariance structures. In a G-side covariance structure, a distinct variance component is assigned to each effect. In an R-side structure TYPE=VC is usually used only to add overdispersion effects or with the GROUP= option to specify a heterogeneous variance model.

Table 38.14 Covariance Structure Examples

Description	Structure	Example
Variance Components	VC (default)	$\begin{bmatrix} \sigma_B^2 & 0 & 0 & 0 \\ 0 & \sigma_B^2 & 0 & 0 \\ 0 & 0 & \sigma_{AB}^2 & 0 \\ 0 & 0 & 0 & \sigma_{AB}^2 \end{bmatrix}$
Compound Symmetry	CS	$\begin{bmatrix} \sigma+\phi & \sigma & \sigma & \sigma \\ \sigma & \sigma+\phi & \sigma & \sigma \\ \sigma & \sigma & \sigma+\phi & \sigma \\ \sigma & \sigma & \sigma & \sigma+\phi \end{bmatrix}$
Heterogeneous CS	CSH	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho & \sigma_1\sigma_3\rho & \sigma_1\sigma_4\rho \\ \sigma_2\sigma_1\rho & \sigma_2^2 & \sigma_2\sigma_3\rho & \sigma_2\sigma_4\rho \\ \sigma_3\sigma_1\rho & \sigma_3\sigma_2\rho & \sigma_3^2 & \sigma_3\sigma_4\rho \\ \sigma_4\sigma_1\rho & \sigma_4\sigma_2\rho & \sigma_4\sigma_3\rho & \sigma_4^2 \end{bmatrix}$
First-Order Autoregressive	AR(1)	$\sigma^2 \begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 \\ \rho & 1 & \rho & \rho^2 \\ \rho^2 & \rho & 1 & \rho \\ \rho^3 & \rho^2 & \rho & 1 \end{bmatrix}$
Heterogeneous AR(1)	ARH(1)	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho & \sigma_1\sigma_3\rho^2 & \sigma_1\sigma_4\rho^3 \\ \sigma_2\sigma_1\rho & \sigma_2^2 & \sigma_2\sigma_3\rho & \sigma_2\sigma_4\rho^2 \\ \sigma_3\sigma_1\rho^2 & \sigma_3\sigma_2\rho & \sigma_3^2 & \sigma_3\sigma_4\rho \\ \sigma_4\sigma_1\rho^3 & \sigma_4\sigma_2\rho & \sigma_4\sigma_3\rho & \sigma_4^2 \end{bmatrix}$
Unstructured	UN	$\begin{bmatrix} \sigma_1^2 & \sigma_{21} & \sigma_{31} & \sigma_{41} \\ \sigma_{21} & \sigma_2^2 & \sigma_{32} & \sigma_{42} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 & \sigma_{43} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_4^2 \end{bmatrix}$
Banded Main Diagonal	UN(1)	$\begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{bmatrix}$
Unstructured Correlations	UNR	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_{21} & \sigma_1\sigma_3\rho_{31} & \sigma_1\sigma_4\rho_{41} \\ \sigma_2\sigma_1\rho_{21} & \sigma_2^2 & \sigma_2\sigma_3\rho_{32} & \sigma_2\sigma_4\rho_{42} \\ \sigma_3\sigma_1\rho_{31} & \sigma_3\sigma_2\rho_{32} & \sigma_3^2 & \sigma_3\sigma_4\rho_{43} \\ \sigma_4\sigma_1\rho_{41} & \sigma_4\sigma_2\rho_{42} & \sigma_4\sigma_3\rho_{43} & \sigma_4^2 \end{bmatrix}$

Table 38.14 *continued*

Description	Structure	Example
Toeplitz	TOEP	$\begin{bmatrix} \sigma^2 & \sigma_1 & \sigma_2 & \sigma_3 \\ \sigma_1 & \sigma^2 & \sigma_1 & \sigma_2 \\ \sigma_2 & \sigma_1 & \sigma^2 & \sigma_1 \\ \sigma_3 & \sigma_2 & \sigma_1 & \sigma^2 \end{bmatrix}$
Toeplitz with Two Bands	TOEP(2)	$\begin{bmatrix} \sigma^2 & \sigma_1 & 0 & 0 \\ \sigma_1 & \sigma^2 & \sigma_1 & 0 \\ 0 & \sigma_1 & \sigma^2 & \sigma_1 \\ 0 & 0 & \sigma_1 & \sigma^2 \end{bmatrix}$
Heterogeneous Toeplitz	TOEPH	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_1 & \sigma_1\sigma_3\rho_2 & \sigma_1\sigma_4\rho_3 \\ \sigma_2\sigma_1\rho_1 & \sigma_2^2 & \sigma_2\sigma_3\rho_1 & \sigma_2\sigma_4\rho_2 \\ \sigma_3\sigma_1\rho_2 & \sigma_3\sigma_2\rho_1 & \sigma_3^2 & \sigma_3\sigma_4\rho_1 \\ \sigma_4\sigma_1\rho_3 & \sigma_4\sigma_2\rho_2 & \sigma_4\sigma_3\rho_1 & \sigma_4^2 \end{bmatrix}$
Spatial Power	SP(POW)(*c-list*)	$\sigma^2 \begin{bmatrix} 1 & \rho^{d_{12}} & \rho^{d_{13}} & \rho^{d_{14}} \\ \rho^{d_{21}} & 1 & \rho^{d_{23}} & \rho^{d_{24}} \\ \rho^{d_{31}} & \rho^{d_{32}} & 1 & \rho^{d_{34}} \\ \rho^{d_{41}} & \rho^{d_{42}} & \rho^{d_{43}} & 1 \end{bmatrix}$
First-Order Autoregressive Moving-Average	ARMA(1,1)	$\sigma^2 \begin{bmatrix} 1 & \gamma & \gamma\rho & \gamma\rho^2 \\ \gamma & 1 & \gamma & \gamma\rho \\ \gamma\rho & \gamma & 1 & \gamma \\ \gamma\rho^2 & \gamma\rho & \gamma & 1 \end{bmatrix}$
First-Order Factor Analytic	FA(1)	$\begin{bmatrix} \lambda_1^2 + d_1 & \lambda_1\lambda_2 & \lambda_1\lambda_3 & \lambda_1\lambda_4 \\ \lambda_2\lambda_1 & \lambda_2^2 + d_2 & \lambda_2\lambda_3 & \lambda_2\lambda_4 \\ \lambda_3\lambda_1 & \lambda_3\lambda_2 & \lambda_3^2 + d_3 & \lambda_3\lambda_4 \\ \lambda_4\lambda_1 & \lambda_4\lambda_2 & \lambda_4\lambda_3 & \lambda_4^2 + d_4 \end{bmatrix}$
Huynh-Feldt	HF	$\begin{bmatrix} \sigma_1^2 & \frac{\sigma_1^2+\sigma_2^2}{2} - \lambda & \frac{\sigma_1^2+\sigma_3^2}{2} - \lambda \\ \frac{\sigma_2^2+\sigma_1^2}{2} - \lambda & \sigma_2^2 & \frac{\sigma_2^2+\sigma_3^2}{2} - \lambda \\ \frac{\sigma_3^2+\sigma_1^2}{2} - \lambda & \frac{\sigma_3^2+\sigma_2^2}{2} - \lambda & \sigma_3^2 \end{bmatrix}$
First-Order Ante-dependence	ANTE(1)	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_1 & \sigma_1\sigma_3\rho_1\rho_2 \\ \sigma_2\sigma_1\rho_1 & \sigma_2^2 & \sigma_2\sigma_3\rho_2 \\ \sigma_3\sigma_1\rho_2\rho_1 & \sigma_3\sigma_2\rho_2 & \sigma_3^2 \end{bmatrix}$

V< =*value-list* **>**

requests that blocks of the estimated marginal variance-covariance matrix $\mathbf{V}(\widehat{\boldsymbol{\theta}})$ be displayed in generalized linear mixed models. This matrix is based on the last linearization as described in the section "The Pseudo-model" on page 2220. You can use the *value-list* to select the subjects for which the matrix is displayed. If *value-list* is not specified, the $\mathbf{V}$ matrix for the first subject is chosen.

Note that the *value-list* refers to subjects as the processing units in the "Dimensions" table. For example, the following statements request that the estimated marginal variance matrix for

the second subject be displayed:

```
proc glimmix;
   class A B;
   model y = B;
   random int / subject=A;
   random int / subject=A*B v=2;
run;
```

The subject effect for processing in this case is the A effect, because it is contained in the A*B interaction. If there is only a single subject as per the "Dimensions" table, then the V option displays an $(n \times n)$ matrix.

See the section "Processing by Subjects" on page 2247 for how the GLIMMIX procedure determines the number of subjects in the "Dimensions" table.

The GLIMMIX procedure displays blanks for values that are 0.

VC<=*value-list***>**
displays the lower-triangular Cholesky root of the blocks of the estimated $\mathbf{V}(\widehat{\boldsymbol{\theta}})$ matrix. See the V option for the specification of *value-list*.

VCI<=*value-list***>**
displays the inverse Cholesky root of the blocks of the estimated $\mathbf{V}(\widehat{\boldsymbol{\theta}})$ matrix. See the V option for the specification of *value-list*.

VCORR<=*value-list***>**
displays the correlation matrix corresponding to the blocks of the estimated $\mathbf{V}(\widehat{\boldsymbol{\theta}})$ matrix. See the V option for the specification of *value-list*.

VI<=*value-list***>**
displays the inverse of the blocks of the estimated $\mathbf{V}(\widehat{\boldsymbol{\theta}})$ matrix. See the V option for the specification of *value-list*.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement replaces $\mathbf{R}$ with $\mathbf{W}^{-1/2}\mathbf{R}\mathbf{W}^{-1/2}$, where $\mathbf{W}$ is a diagonal matrix containing the weights. Observations with nonpositive or missing weights are not included in the resulting PROC GLIMMIX analysis. If a WEIGHT statement is not included, all observations used in the analysis are assigned a weight of 1.

Programming Statements

This section lists the programming statements available in PROC GLIMMIX to compute various aspects of the generalized linear mixed model or output quantities. For example, you can compute model effects, weights, frequency, subject, group, and other variables. You can use programming statements to define the mean and variance functions. This section also documents the differences between programming statements in PROC GLIMMIX and programming statements in the SAS DATA step. The syntax of programming statements used in PROC GLIMMIX is identical to that used in the NLMIXED procedure (see Chapter 61, "The NLMIXED Procedure," of the *SAS/STAT User's Guide*) and the MODEL procedure (see the *SAS/ETS User's Guide*). Most of the programming statements that can be used in the DATA step can also be used in the GLIMMIX procedure. Refer to *SAS Language Reference: Dictionary* for a description of SAS programming statements. The following are valid statements:

ABORT;
CALL *name [(expression [, expression . . .])]*;
DELETE;
DO *[variable = expression*
*[***TO** *expression] [***BY** *expression]*
[, expression [**TO** *expression] [* **BY** *expression] . . .]*
]
[**WHILE** *expression] [* **UNTIL** *expression]*;
END;
GOTO *statement_label*;
IF *expression*;
IF *expression* **THEN** *program_statement*;
ELSE *program_statement*;
variable = expression;
variable + expression;
LINK *statement_label*;
PUT *[variable] [=] [...]*;
RETURN;
SELECT[(*expression* **)];**
STOP;
SUBSTR(*variable, index, length* **)=** *expression*;
WHEN *(expression) program_statement*;
OTHERWISE *program_statement*;

For the most part, the SAS programming statements work the same as they do in the SAS DATA step, as documented in *SAS Language Reference: Concepts*. However, there are several differences:

- The ABORT statement does not allow any arguments.
- The DO statement does not allow a character index variable. Thus

  ```
  do i = 1,2,3;
  ```

  is supported; however, the following statement is not supported:

  ```
  do i = 'A','B','C';
  ```

- The LAG function is not supported with PROC GLIMMIX.
- The PUT statement, used mostly for program debugging in PROC GLIMMIX, supports only some of the features of the DATA step PUT statement, and it has some features not available with the DATA step PUT statement:
  - The PROC GLIMMIX PUT statement does not support line pointers, factored lists, iteration factors, overprinting, _INFILE_, the colon (:) format modifier, or "$".
  - The PROC GLIMMIX PUT statement does support expressions, but the expression must be enclosed in parentheses. For example, the following statement displays the square root of x:

    ```
    put (sqrt(x));
    ```

  - The PROC GLIMMIX PUT statement supports the item _PDV_ to display a formatted listing of all variables in the program. For example:

    ```
    put _pdv_;
    ```

- The WHEN and OTHERWISE statements enable you to specify more than one target statement. That is, DO/END groups are not necessary for multiple statement WHENs. For example, the following syntax is valid:

```
select;
   when (exp1) stmt1;
               stmt2;
   when (exp2) stmt3;
               stmt4;
end;
```

The LINK statement is used in a program to jump immediately to the label *statement_label* and to continue program execution at that point. It is not used to specify a user-defined link function.

When coding your programming statements, you should avoid defining variables that begin with an underscore (_), because they might conflict with internal variables created by PROC GLIMMIX.

User-Defined Link or Variance Function

Implied Variance Functions

While link functions are not unique for each distribution (see Table 38.9 for the default link functions), the distribution does determine the variance function $a(\mu)$. This function expresses the variance of an observation as a function of the mean, apart from weights, frequencies, and additional scale parameters. The implied variance functions $a(\mu)$ of the GLIMMIX procedure are shown in Table 38.15 for the supported distributions. For the binomial distribution, n denotes the number of trials in the *events/trials* syntax. For the negative binomial distribution, k denotes the scale parameter. The multiplicative scale parameter ϕ is not included for the other distributions. The last column of the table indicates whether ϕ has a value equal to 1.0 for the particular distribution.

Table 38.15 Variance Functions in PROC GLIMMIX

DIST=	Distribution	Variance function $a(\mu)$	$\phi \equiv 1$
BETA	beta	$\mu(1-\mu)/(1+\phi)$	No
BINARY	binary	$\mu(1-\mu)$	Yes
BINOMIAL | BIN | B	binomial	$\mu(1-\mu)/n$	Yes
EXPONENTIAL | EXPO	exponential	μ^2	Yes
GAMMA | GAM	gamma	μ^2	No
GAUSSIAN | G | NORMAL | N	normal	1	No
GEOMETRIC | GEOM	geometric	$\mu+\mu^2$	Yes
INVGAUSS | IGAUSSIAN | IG	inverse gaussian	μ^3	No
LOGNORMAL | LOGN	lognormal	1	No
NEGBINOMIAL | NEGBIN | NB	negative binomial	$\mu+k\mu^2$	Yes
POISSON | POI | P	Poisson	μ	Yes
TCENTRAL | TDIST | T	t	$\nu/(\nu-2)$	No

To change the variance function, you can use SAS programming statements and the predefined automatic variables, as outlined in the following section. Your definition of a variance function will override the DIST= option and its implied variance function. This has the following implication for parameter estimation with the GLIMMIX procedure. When a user-defined link is available, the distribution of the data is determined from the DIST= option, or the respective default for the type of response. In a GLM, for example, this enables maximum likelihood estimation. If a user-defined variance function is provided, the DIST= option is not honored and the distribution of the data is assumed unknown. In a GLM framework, only quasi-likelihood estimation is then available to estimate the model parameters.

Automatic Variables

To specify your own link or variance function you can use SAS programming statements and draw on the following automatic variables:

LINP is the current value of the linear predictor. It equals either $\widehat{\eta} = \mathbf{x}'\widehat{\boldsymbol{\beta}} + \mathbf{z}'\widehat{\boldsymbol{\gamma}} + o$ or $\widehat{\eta} = \mathbf{x}'\widehat{\boldsymbol{\beta}} + o$, where o is the value of the offset variable, or 0 if no offset is specified. The estimated random effects solutions $\widehat{\boldsymbol{\gamma}}$ are used in the calculation of the linear predictor during the model fitting phase, if a linearization expands about the current values of $\boldsymbol{\gamma}$. During the computation of output statistics, the EBLUPs are used if statistics depend on them. For example, the following statements add the variable p to the output data set glimmixout:

```
proc glimmix;
   model y = x / dist=binary;
   random int / subject=b;
   p = 1/(1+exp(-_linp_);
   output out=glimmixout;
   id p;
run;
```

Because no output statistics are requested in the OUTPUT statement that depend on the random-effects solutions (BLUPs, EBEs), the value of _LINP_ in this example equals $\mathbf{x}'\widehat{\boldsymbol{\beta}}$. On the contrary, the following statements also request conditional residuals on the logistic scale:

```
proc glimmix;
   model y = x / dist=binary;
   random int / subject=b;
   p = 1/(1+exp(-_linp_);
   output out=glimmixout resid(blup)=r;
   id p;
run;
```

The value of _LINP_ when computing the variable p is $\mathbf{x}'\widehat{\boldsymbol{\beta}} + \mathbf{z}'\widehat{\boldsymbol{\gamma}}$. To ensure that computed statistics are formed from $\mathbf{x}'\widehat{\boldsymbol{\beta}}$ and $\mathbf{z}'\widehat{\boldsymbol{\gamma}}$ terms as needed, it is recommended that you use the automatic variables _XBETA_ and _ZGAMMA_ instead of _LINP_.

_MU_ expresses the mean of an observation as a function of the linear predictor, $\widehat{\mu} = g^{-1}(\widehat{\eta})$.

_N_ is the observation number in the sequence of the data read.

VARIANCE is the estimate of the variance function, $a(\widehat{\mu})$.

XBETA equals $\mathbf{x}'\widehat{\boldsymbol{\beta}}$.

ZGAMMA equals $\mathbf{z}'\widehat{\boldsymbol{\gamma}}$.

The automatic variable _N_ is incremented whenever the procedure reads an observation from the data set. Observations that are not used in the analysis—for example, because of missing values or invalid weights—are counted. The counter is reset to 1 at the start of every new BY group. Only in some circumstances will _N_ equal the actual observation number. The symbol should thus be used sparingly to avoid unexpected results.

You must observe the following syntax rules when you use the automatic variables. The _LINP_ symbol cannot appear on the left side of programming statements; you cannot make an assignment to the _LINP_ variable. The value of the linear predictor is controlled by the CLASS, MODEL, and RANDOM statements as well as the current parameter estimates and solutions. You can, however, use the _LINP_ variable on the right side of other operations. Suppose, for example, that you want to transform the linear predictor prior to applying the inverse log link. The following statements are not valid because the linear predictor appears in an assignment:

```
proc glimmix;
   _linp_ = sqrt(abs(_linp_));
   _mu_   = exp(_linp_);
   model count = logtstd / dist=poisson;
run;
```

The next statements achieve the desired result:

```
proc glimmix;
   _mu_ = exp(sqrt(abs(_linp_)));
   model count = logtstd / dist=poisson;
run;
```

If the value of the linear predictor is altered in any way through programming statements, you need to ensure that an assignment to _MU_ follows. The assignment to variable P in the next set of GLIMMIX statements is without effect:

```
proc glimmix;
   p = _linp_ + rannor(454);
   model count = logtstd / dist=poisson;
run;
```

A user-defined link function is implied by expressing _MU_ as a function of _LINP_. That is, if $\mu = g^{-1}(\eta)$, you are providing an expression for the inverse link function with programming statements. It is neither necessary nor possible to give an expression for the inverse operation, $\eta = g(\mu)$. The variance function is determined by expressing _VARIANCE_ as a function of _MU_. If the _MU_ variable appears in an assignment statement inside PROC GLIMMIX, the LINK= option of the MODEL statement is ignored. If the _VARIANCE_ function appears in an assignment statement, the DIST= option is ignored. Furthermore, the associated variance function per Table 38.15 is not honored. In short, user-defined expressions take precedence over built-in defaults.

If you specify your own link and variance function, the assignment to _MU_ must precede an assignment to the variable _VARIANCE_.

The following two sets of GLIMMIX statements yield the same parameter estimates, but the models differ statistically:

```
proc glimmix;
   class block entry;
   model y/n = block entry / dist=binomial link=logit;
run;

proc glimmix;
   class block entry;
   prob = 1 / (1+exp(- _linp_));
   _mu_ = n * prob ;
   _variance_ = n * prob *(1-prob);
   model y = block entry;
run;
```

The first GLIMMIX invocation models the proportion y/n as a binomial proportion with a logit link. The DIST= and LINK= options are superfluous in this case, because the GLIMMIX procedure defaults to the binomial distribution in light of the *events/trials* syntax. The logit link is that distribution's default link. The second set of GLIMMIX statements models the count variable y and takes the binomial sample size into account through assignments to the mean and variance function. In contrast to the first set of GLIMMIX statements, the distribution of y is unknown. Only its mean and variance are known. The model parameters are estimated by maximum likelihood in the first case and by quasi-likelihood in the second case.

Details: GLIMMIX Procedure

Generalized Linear Models Theory

A generalized linear model consists of the following:

- a linear predictor $\eta = \mathbf{x}'\boldsymbol{\beta}$
- a monotonic mapping between the mean of the data and the linear predictor
- a response distribution in the exponential family of distributions

A density or mass function in this family can be written as

$$f(y) = \exp\left\{\frac{y\theta - b(\theta)}{\phi} + c(y, f(\phi))\right\}$$

for some functions $b(\cdot)$ and $c(\cdot)$. The parameter θ is called the natural (canonical) parameter. The parameter ϕ is a scale parameter, and it is not present in all exponential family distributions. See Table 38.15 for a list of distributions for which $\phi \equiv 1$. In the case where observations are weighted, the scale parameter is replaced with ϕ/w in the preceding density (or mass function), where w is the weight associated with the observation y.

The mean and variance of the data are related to the components of the density, $\mathrm{E}[Y] = \mu = b'(\theta)$, $\mathrm{Var}[Y] = \phi b''(\theta)$, where primes denote first and second derivatives. If you express θ as a function of μ, the relationship is known as the natural link or the canonical link function. In other words, modeling data with a canonical link assumes that $\theta = \mathbf{x}'\boldsymbol{\beta}$; the effect contributions are additive on the canonical scale. The second derivative of $b(\cdot)$, expressed as a function of μ, is the variance function of the generalized linear model, $a(\mu) = b''(\theta(\mu))$. Note that because of this relationship, the distribution determines the variance function and the canonical link function. You cannot, however, proceed in the opposite direction. If you provide a user-specified variance function, the GLIMMIX procedure assumes that only the first two moments of the response distribution are known. The full distribution of the data is then unknown and maximum likelihood estimation is not possible. Instead, the GLIMMIX procedure then estimates parameters by quasi-likelihood.

Maximum Likelihood

The GLIMMIX procedure forms the log likelihoods of generalized linear models as

$$L(\boldsymbol{\mu}, \phi; \mathbf{y}) = \sum_{i=1}^{n} f_i \, l(\mu_i, \phi; y_i, w_i)$$

where $l(\mu_i, \phi; y_i, w_i)$ is the log likelihood contribution of the ith observation with weight w_i and f_i is the value of the frequency variable. For the determination of w_i and f_i, see the WEIGHT and FREQ statements. The individual log likelihood contributions for the various distributions are as follows.

Beta

$$l(\mu_i, \phi; y_i, w_i) = \log\left\{\frac{\Gamma(\phi/w_i)}{\Gamma(\mu\phi/w_i)\Gamma((1-\mu)\phi/w_i)}\right\} + (\mu\phi/w_i - 1)\log\{y_i\} + ((1-\mu)\phi/w_i - 1)\log\{1-y_i\}$$

$\text{Var}[Y] = \mu(1-\mu)/(1+\phi)$, $\phi > 0$. See Ferrari and Cribari-Neto (2004).

Binary

$$l(\mu_i, \phi; y_i, w_i) = w_i(y_i \log\{\mu_i\} + (1-y_i)\log\{1-\mu_i\})$$

$\text{Var}[Y] = \mu(1-\mu)$, $\phi \equiv 1$.

Binomial

$$l(\mu_i, \phi; y_i, w_i) = w_i\,(y_i \log\{\mu_i\} + (n_i - y_i)\log\{1-\mu_i\}) + w_i\,(\log\{\Gamma(n_i+1)\} - \log\{\Gamma(y_i+1)\} - \log\{\Gamma(n_i - y_i + 1)\})$$

where y_i and n_i are the *events* and *trials* in the *events/trials* syntax, and $0 < \mu < 1$. $\text{Var}[Y/n] = \mu(1-\mu)/n$, $\phi \equiv 1$.

Exponential

$$l(\mu_i, \phi; y_i, w_i) = \begin{cases} -\log\{\mu_i\} - y_i/\mu_i & w_i = 1 \\ w_i \log\left\{\frac{w_i y_i}{\mu_i}\right\} - \frac{w_i y_i}{\mu_i} - \log\{y_i \Gamma(w_i)\} & w_i \neq 1 \end{cases}$$

$\text{Var}[Y] = \mu^2$, $\phi \equiv 1$.

Gamma

$$l(\mu_i, \phi; y_i, w_i) = w_i/\phi \log\left\{\frac{w_i y_i}{\phi\mu_i}\right\} - \frac{w_i y_i}{\phi\mu_i} - \log\{y_i\} - \log\{\Gamma(w_i/\phi)\}$$

$\text{Var}[Y] = \phi\mu^2$, $\phi > 0$.

Geometric

$$l(\mu_i, \phi; y_i, w_i) = y_i \log\left\{\frac{\mu_i}{w_i}\right\} - (y_i + w_i)\log\left\{1 + \frac{\mu_i}{w_i}\right\} + \log\left\{\frac{\Gamma(y_i + w_i)}{\Gamma(w_i)\Gamma(y_i+1)}\right\}$$

$\text{Var}[Y] = \mu + \mu^2$, $\phi \equiv 1$.

Inverse Gaussian

$$l(\mu_i, \phi; y_i, w_i) = -\frac{1}{2}\left[\frac{w_i(y_i - \mu_i)^2}{y_i \phi \mu_i^2} + \log\left\{\frac{\phi y_i^3}{w_i}\right\} + \log\{2\pi\}\right]$$

$\text{Var}[Y] = \phi\mu^3$, $\phi > 0$.

"Lognormal"

$$l(\mu_i, \phi; \log\{y_i\}, w_i) = -\frac{1}{2}\left[\frac{w_i(\log\{y_i\} - \mu_i)^2}{\phi} + \log\left\{\frac{\phi}{w_i}\right\} + \log\{2\pi\}\right]$$

$\text{Var}[\log\{Y\}] = \phi$, $\phi > 0$.
If you specify DIST=LOGNORMAL with response variable Y, the GLIMMIX procedure assumes that $\log\{Y\} \sim N(\mu, \sigma^2)$. Note that the preceding density is not the density of Y.

Multinomial

$$l(\boldsymbol{\mu}_i, \phi; \mathbf{y}_i, w_i) = w_i \sum_{j=1}^{J} y_{ij} \log\{\mu_{ij}\}$$

$\phi \equiv 1$.

Negative Binomial

$$l(\mu_i, \phi; y_i, w_i) = y_i \log\left\{\frac{k\mu_i}{w_i}\right\} - (y_i + w_i/k)\log\left\{1 + \frac{k\mu_i}{w_i}\right\} + \log\left\{\frac{\Gamma(y_i + w_i/k)}{\Gamma(w_i/k)\Gamma(y_i + 1)}\right\}$$

$\text{Var}[Y] = \mu + k\mu^2$, $k > 0, \phi \equiv 1$.
For a given k, the negative binomial distribution is a member of the exponential family. The parameter k is related to the scale of the data, because it is part of the variance function. However, it cannot be factored from the variance, as is the case with the ϕ parameter in many other distributions. The parameter k is designated as "Scale" in the "Parameter Estimates" table of the GLIMMIX procedure.

Normal (Gaussian)

$$l(\mu_i, \phi; y_i, w_i) = -\frac{1}{2}\left[\frac{w_i(y_i - \mu_i)^2}{\phi} + \log\left\{\frac{\phi}{w_i}\right\} + \log\{2\pi\}\right]$$

$\text{Var}[Y] = \phi$, $\phi > 0$.

Poisson

$$l(\mu_i, \phi; y_i, w_i) = w_i(y_i \log\{\mu_i\} - \mu_i - \log\{\Gamma(y_i + 1)\})$$

$\text{Var}[Y] = \mu$, $\phi \equiv 1$.

Shifted T

$$\begin{aligned} z_i &= -0.5\log\{\phi/\sqrt{w_i}\} + \log\{\Gamma(0.5(\nu+1)\} \\ &\quad - \log\{\Gamma(0.5\nu)\} - 0.5 \times \log\{\pi\nu\} \\ l(\mu_i, \phi; y_i, w_i) &= -(\nu/2 + 0.5)\log\left\{1 + \frac{w_i}{\nu}\frac{(y_i - \mu_i)^2}{\phi}\right\} + z_i \end{aligned}$$

$\phi > 0, \nu > 0$, $\text{Var}[Y] = \phi\nu/(\nu - 2)$.

Define the parameter vector for the generalized linear model as $\boldsymbol{\theta} = \boldsymbol{\beta}$, if $\phi \equiv 1$, and as $\boldsymbol{\theta} = [\boldsymbol{\beta}', \phi]'$ otherwise. $\boldsymbol{\beta}$ denotes the fixed-effects parameters in the linear predictor. For the negative binomial distribution, the relevant parameter vector is $\boldsymbol{\theta} = [\boldsymbol{\beta}', k]'$. The gradient and Hessian of the negative log likelihood are then

$$\mathbf{g} = -\frac{\partial L(\boldsymbol{\theta}; \mathbf{y})}{\partial \boldsymbol{\theta}} \qquad \mathbf{H} = -\frac{\partial^2 L(\boldsymbol{\theta}; \mathbf{y})}{\partial \boldsymbol{\theta}\, \partial \boldsymbol{\theta}'}$$

The GLIMMIX procedure computes the gradient vector and Hessian matrix analytically, unless your programming statements involve functions whose derivatives are determined by finite differences. If the procedure is in scoring mode, $\mathbf{H}$ is replaced by its expected value. PROC GLIMMIX is in scoring mode when the number n of SCORING=n iterations has not been exceeded and the optimization technique uses second derivatives, or when the Hessian is computed at convergence and the EXPHESSIAN option is in effect. Note that the objective function is the negative log likelihood when the GLIMMIX procedure fits a GLM model. The procedure performs a minimization problem in this case.

In models for independent data with known distribution, parameter estimates are obtained by the method of maximum likelihood. No parameters are profiled from the optimization. The default optimization technique for GLMs is the Newton-Raphson algorithm, except for Gaussian models with identity link, which do not require iterative model fitting. In the case of a Gaussian model, the scale parameter is estimated by restricted maximum likelihood, because this estimate is unbiased. The results from the GLIMMIX procedure agree with those from the GLM and REG procedure for such models. You can obtain the maximum likelihood estimate of the scale parameter with the NOREML option in the PROC GLIMMIX statement. To change the optimization algorithm, use the TECHNIQUE= option in the NLOPTIONS statement.

Standard errors of the parameter estimates are obtained from the inverse of the (observed or expected) second derivative matrix $\mathbf{H}$.

Scale and Dispersion Parameters

The parameter ϕ in the log-likelihood functions is a scale parameter. McCullagh and Nelder (1989, p. 29) refer to it as the dispersion parameter. With the exception of the normal distribution, ϕ does not correspond to the variance of an observation, the variance of an observation in a generalized linear model is a function of ϕ and μ. In a generalized linear model (GLM mode), the GLIMMIX procedure displays the estimate of ϕ is as "Scale" in the "Parameter Estimates" table. Note that for some distributions this scale is different from that reported by the GENMOD procedure in its "Parameter Estimates" table. The scale reported by PROC GENMOD is sometimes a transformation of the dispersion parameter in the log-likelihood function. Table 38.15 displays the relationship between the "Scale" entries reported by the two procedures in terms of the ϕ (or k) parameter in the GLIMMIX log-likelihood functions.

Table 38.15 Scales in Parameter Estimates Table

Distribution	GLIMMIX Reports	GENMOD Reports
Beta	$\widehat{\phi}$	N/A
Gamma	$\widehat{\phi}$	$1/\widehat{\phi}$
Inverse gaussian	$\widehat{\phi}$	$\sqrt{\widehat{\phi}}$
Negative binomial	$\widehat{k}$	$\widehat{k}$
Normal	$\widehat{\phi} = \widehat{\text{Var}}[Y]$	$\sqrt{\widehat{\phi}}$

Note that for normal linear models, PROC GLIMMIX by default estimates the parameters by restricted maximum likelihood, whereas PROC GENMOD estimates the parameters by maximum likelihood. As a consequence, the scale parameter in the "Parameter Estimates" table of the GLIMMIX procedure coincides for these models with the mean-squared error estimate of the GLM or REG procedures. To obtain maximum likelihood estimates in a normal linear model in the GLIMMIX procedure, specify the NOREML option in the PROC GLIMMIX statement.

Quasi-likelihood for Independent Data

Quasi-likelihood estimation uses only the first and second moment of the response. In the case of independent data, this requires only a specification of the mean and variance of your data. The GLIMMIX procedure estimates parameters by quasi-likelihood, if the following conditions are met:

- The response distribution is unknown, because of a user-specified variance function.
- There are no G-side random effects.
- There are no R-side covariance structures or at most an overdispersion parameter.

Under some mild regularity conditions, the function

$$Q(\mu_i, y_i) = \int_{y_i}^{\mu_i} \frac{y_i - t}{\phi a(\mu_i)}\, dt$$

known as the log quasi-likelihood of the ith observation, has some properties of a log-likelihood function (McCullagh and Nelder 1989, p. 325). For example, the expected value of its derivative is zero, and the variance of its derivative equals the negative of the expected value of the second derivative. Consequently,

$$QL(\boldsymbol{\mu}, \phi, \mathbf{y}) = \sum_{i=1}^{n} f_i w_i \frac{Y_i - \mu_i}{\phi a(\mu_i)}$$

can serve as the score function for estimation. Quasi-likelihood estimation takes as the gradient and "Hessian" matrix—with respect to the fixed-effects parameters $\boldsymbol{\beta}$—the quantities

$$\mathbf{g}_{ql} = \left[g_{ql,j}\right] = \left[\frac{\partial QL(\boldsymbol{\mu}, \phi, \mathbf{y})}{\partial \beta_j}\right] = \mathbf{D}'\mathbf{V}^{-1}(\mathbf{Y} - \boldsymbol{\mu})/\phi$$

$$\mathbf{H}_{ql} = \left[h_{ql,jk}\right] = \left[\frac{\partial^2 QL(\boldsymbol{\mu}, \phi, \mathbf{y})}{\partial \beta_j \partial \beta_k}\right] = \mathbf{D}'\mathbf{V}^{-1}\mathbf{D}/\phi$$

In this expression, $\mathbf{D}$ is a matrix of derivatives of $\boldsymbol{\mu}$ with respect to the elements in $\boldsymbol{\beta}$, and $\mathbf{V}$ is a diagonal matrix containing variance functions, $\mathbf{V} = [a(\mu_1), \cdots, a(\mu_n)]$. Notice that $\mathbf{H}_{ql}$ is not the second derivative matrix of $Q(\boldsymbol{\mu}, \mathbf{y})$. Rather, it is the negative of the expected value of $\partial \mathbf{g}_{gl} / \partial \boldsymbol{\beta}$. $\mathbf{H}_{ql}$ thus has the form of a "scoring Hessian."

The GLIMMIX procedure fixes the scale parameter ϕ at 1.0 by default. To estimate the parameter, add the statement

```
random _residual_;
```

The resulting estimator (McCullagh and Nelder 1989, p. 328) is

$$\widehat{\phi} = \frac{1}{m} \sum_{i=1}^{n} f_i w_i \frac{(y_i - \widehat{\mu}_i)^2}{a(\widehat{\mu}_i)}$$

where $m = f - \text{rank}\{\mathbf{X}\}$ if the NOREML option is in effect, $m = f$ otherwise, and f is the sum of the frequencies.

See Example 38.4 for an application of quasi-likelihood estimation with PROC GLIMMIX.

Effects of Adding Overdispersion

You can add a multiplicative overdispersion parameter to a generalized linear model in the GLIMMIX procedure with the statement

```
random _residual_;
```

For models in which $\phi \equiv 1$, this effectively lifts the constraint of the parameter. In models that already contain a ϕ or k scale parameter—such as the normal, gamma, or negative binomial model—the statement adds a multiplicative scalar (the overdispersion parameter, ϕ_o) to the variance function.

The overdispersion parameter is estimated from Pearson's statistic after all other parameters have been determined by (restricted) maximum likelihood or quasi-likelihood. This estimate is

$$\widehat{\phi}_o = \frac{1}{\phi^p m} \sum_{i=1}^{n} f_i w_i \frac{(y_i - \mu_i)^2}{a(\mu_i)}$$

where $m = f - \text{rank}\{\mathbf{X}\}$ if the NOREML option is in effect, and $m = f$ otherwise, and f is the sum of the frequencies. The power p is -1 for the gamma distribution and 1 otherwise.

Adding an overdispersion parameter does not alter any of the other parameter estimates. It only changes the variance-covariance matrix of the estimates by a certain factor. If overdispersion arises from correlations among the observations, then you should investigate more complex random-effects structures.

Generalized Linear Mixed Models Theory

Model or Integral Approximation

In a generalized linear model, the log likelihood is well defined, and an objective function for estimation of the parameters is simple to construct based on the independence of the data. In a GLMM, several problems must be overcome before an objective function can be computed.

- The model might be vacuous in the sense that no valid joint distribution can be constructed either in general or for a particular set of parameter values. For example, if $\mathbf{Y}$ is an equicorrelated $(n \times 1)$ vector of binary responses with the same success probability and a symmetric distribution, then the lower bound on the correlation parameter depends on n and π (Gilliland and Schabenberger 2001). If further restrictions are placed on the joint distribution, as in Bahadur (1961), the correlation is also restricted from above.
- The dependency between mean and variance for nonnormal data places constraints on the possible correlation models that simultaneously yield valid joint distributions and a desired conditional distributions. Thus, for example, aspiring for conditional Poisson variates that are marginally correlated according to a spherical spatial process might not be possible.
- Even if the joint distribution is feasible mathematically, it still can be out of reach computationally. When data are independent, conditional on the random effects, the marginal log likelihood can in principle be constructed by integrating out the random effects from the joint distribution. However, numerical integration is practical only when the number of random effects is small and when the data have a clustered (subject) structure.

Because of these special features of generalized linear mixed models, many estimation methods have been put forth in the literature. The two basic approaches are (1) to approximate the objective function and (2) to approximate the model. Algorithms in the second category can be expressed in terms of Taylor series (linearizations) and are hence also known as linearization methods. They employ expansions to approximate the model by one based on pseudo-data with fewer nonlinear components. The process of computing the linear approximation must be repeated several times until some criterion indicates lack of further progress. Schabenberger and Gregoire (1996) list numerous algorithms based on Taylor series for the case of clustered data alone. The fitting methods based on linearizations are usually doubly iterative. The generalized linear mixed model is approximated by a linear mixed model based on current values of the covariance parameter estimates. The resulting linear mixed model is then fit, which is itself an iterative process. On convergence, the new parameter estimates are used to update the linearization, which results in a new linear mixed model. The process stops when parameter estimates between successive linear mixed model fits change only within a specified tolerance.

Integral approximation methods approximate the log likelihood of the GLMM and submit the approximated function to numerical optimization. Various techniques are used to compute the approximation: Laplace methods, quadrature methods, Monte Carlo integration, and Markov chain Monte Carlo methods. The advantage of integral approximation methods is to provide an actual objective function for optimization. This enables you to perform likelihood ratio tests among nested models and to compute likelihood-based fit statistics. The estimation process is singly iterative.

The disadvantage of integral approximation methods is the difficulty of accommodating crossed random effects and multiple subject effects, and the inability to accommodate R-side covariance structures, even only R-side overdispersion. The number of random effects should be small for integral approximation methods to be practically feasible.

The advantages of linearization-based methods include a relatively simple form of the linearized model that typically can be fit based on only the mean and variance in the linearized form. Models for which the joint distribution is difficult—or impossible—to ascertain can be fit with linearization-based approaches. Models with correlated errors, a large number of random effects, crossed random effects, and multiple types of subjects are thus excellent candidates for linearization methods. The disadvantages of this approach include the absence of a true objective function for the overall optimization process and potentially biased estimates, especially for binary data when the number of observations per subject is small (see the section "Notes on Bias of Estimators" on page 2232 for further comments and considerations about the bias of estimates in generalized linear mixed models). Because the objective function to be optimized after each linearization update depends on the current pseudo-data, objective functions are not comparable across linearizations. The estimation process can fail at both levels of the double iteration scheme.

By default the GLIMMIX procedure fits generalized linear mixed models based on linearizations. The default estimation method in GLIMMIX for models containing random effects is a technique known as restricted pseudo-likelihood (RPL) (Wolfinger and O'Connell 1993) estimation with an expansion around the current estimate of the best linear unbiased predictors of the random effects (METHOD=RSPL).

Two maximum likelihood estimation methods based on integral approximation are available in the GLIMMIX procedure. If you choose METHOD=LAPLACE in a GLMM, the GLIMMIX procedure performs maximum likelihood estimation based on a Laplace approximation of the marginal log likelihood. See the section "Maximum Likelihood Estimation Based on Laplace Approximation" on page 2224 for details about the Laplace approximation with PROC GLIMMIX. If you choose METHOD=QUAD in the PROC GLIMMIX statement in a generalized linear mixed model, the GLIMMIX procedure estimates the model parameters by adaptive Gauss-Hermite quadrature. See the section "Maximum Likelihood Estimation Based on Adaptive Quadrature" on page 2227 for details about the adaptive Gauss-Hermite quadrature approximation with PROC GLIMMIX.

The following subsections discuss the three estimation methods in turn. Keep in mind that your modeling possibilities are increasingly restricted in the order of these subsections. For example, in the class of generalized linear mixed models, the pseudo-likelihood estimation methods place no restrictions on the covariance structure, and Laplace estimation adds restriction with respect to the R-side covariance structure. Adaptive quadrature estimation further requires a clustered data structure—that is, the data must be processed by subjects.

Table 38.16 Model Restrictions Depending on Estimation Method

Method	Restriction
RSPL, RMPL	None
MSPL, MMPL	None
LAPLACE	No R-side effects
QUAD	No R-side effects Requires SUBJECT= effect Requires processing by subjects

Pseudo-likelihood Estimation Based on Linearization

The Pseudo-model

Recall from the section "Notation for the Generalized Linear Mixed Model" on page 2083 that

$$\mathrm{E}[\mathbf{Y}|\boldsymbol{\gamma}] = g^{-1}(\mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma}) = g^{-1}(\boldsymbol{\eta}) = \boldsymbol{\mu}$$

where $\boldsymbol{\gamma} \sim N(\mathbf{0}, \mathbf{G})$ and $\mathrm{Var}[\mathbf{Y}|\boldsymbol{\gamma}] = \mathbf{A}^{1/2}\mathbf{R}\mathbf{A}^{1/2}$. Following Wolfinger and O'Connell (1993), a first-order Taylor series of $\boldsymbol{\mu}$ about $\widetilde{\boldsymbol{\beta}}$ and $\widetilde{\boldsymbol{\gamma}}$ yields

$$g^{-1}(\boldsymbol{\eta}) \doteq g^{-1}(\widetilde{\boldsymbol{\eta}}) + \widetilde{\boldsymbol{\Delta}}\mathbf{X}(\boldsymbol{\beta} - \widetilde{\boldsymbol{\beta}}) + \widetilde{\boldsymbol{\Delta}}\mathbf{Z}(\boldsymbol{\gamma} - \widetilde{\boldsymbol{\gamma}})$$

where

$$\widetilde{\boldsymbol{\Delta}} = \left(\frac{\partial g^{-1}(\boldsymbol{\eta})}{\partial \boldsymbol{\eta}}\right)_{\widetilde{\boldsymbol{\beta}},\widetilde{\boldsymbol{\gamma}}}$$

is a diagonal matrix of derivatives of the conditional mean evaluated at the expansion locus. Rearranging terms yields the expression

$$\widetilde{\boldsymbol{\Delta}}^{-1}(\boldsymbol{\mu} - g^{-1}(\widetilde{\boldsymbol{\eta}})) + \mathbf{X}\widetilde{\boldsymbol{\beta}} + \mathbf{Z}\widetilde{\boldsymbol{\gamma}} \doteq \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma}$$

The left side is the expected value, conditional on $\boldsymbol{\gamma}$, of

$$\widetilde{\boldsymbol{\Delta}}^{-1}(\mathbf{Y} - g^{-1}(\widetilde{\boldsymbol{\eta}})) + \mathbf{X}\widetilde{\boldsymbol{\beta}} + \mathbf{Z}\widetilde{\boldsymbol{\gamma}} \equiv \mathbf{P}$$

and

$$\mathrm{Var}[\mathbf{P}|\boldsymbol{\gamma}] = \widetilde{\boldsymbol{\Delta}}^{-1}\mathbf{A}^{1/2}\mathbf{R}\mathbf{A}^{1/2}\widetilde{\boldsymbol{\Delta}}^{-1}$$

You can thus consider the model

$$\mathbf{P} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

which is a linear mixed model with pseudo-response $\mathbf{P}$, fixed effects $\boldsymbol{\beta}$, random effects $\boldsymbol{\gamma}$, and $\mathrm{Var}[\boldsymbol{\epsilon}] = \mathrm{Var}[\mathbf{P}|\boldsymbol{\gamma}]$.

Objective Functions

Now define

$$\mathbf{V}(\boldsymbol{\theta}) = \mathbf{Z}\mathbf{G}\mathbf{Z}' + \widetilde{\boldsymbol{\Delta}}^{-1}\mathbf{A}^{1/2}\mathbf{R}\mathbf{A}^{1/2}\widetilde{\boldsymbol{\Delta}}^{-1}$$

as the marginal variance in the linear mixed pseudo-model, where $\boldsymbol{\theta}$ is the $(q \times 1)$ parameter vector containing all unknowns in $\mathbf{G}$ and $\mathbf{R}$. Based on this linearized model, an objective function can be defined, assuming that the distribution of $\mathbf{P}$ is known. The GLIMMIX procedure assumes that $\boldsymbol{\epsilon}$ has a normal distribution. The maximum log pseudo-likelihood (M*x*PL) and restricted log pseudo-likelihood (R*x*PL) for $\mathbf{P}$ are then

$$l(\boldsymbol{\theta}, \mathbf{p}) = -\frac{1}{2}\log|\mathbf{V}(\boldsymbol{\theta})| - \frac{1}{2}\mathbf{r}'\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{r} - \frac{f}{2}\log\{2\pi\}$$

$$l_R(\boldsymbol{\theta}, \mathbf{p}) = -\frac{1}{2}\log|\mathbf{V}(\boldsymbol{\theta})| - \frac{1}{2}\mathbf{r}'\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{r} - \frac{1}{2}\log|\mathbf{X}'\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{X}| - \frac{f-k}{2}\log\{2\pi\}$$

with $\mathbf{r} = \mathbf{p} - \mathbf{X}(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-}\mathbf{X}'\mathbf{V}^{-1}\mathbf{p}$. f denotes the sum of the frequencies used in the analysis, and k denotes the rank of $\mathbf{X}$. The fixed-effects parameters $\boldsymbol{\beta}$ are profiled from these expressions. The parameters in $\boldsymbol{\theta}$ are estimated by the optimization techniques specified in the NLOPTIONS statement. The objective function for minimization is $-2l(\boldsymbol{\theta}, \mathbf{p})$ or $-2l_R(\boldsymbol{\theta}, \mathbf{p})$. At convergence, the profiled parameters are estimated and the random effects are predicted as

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}\mathbf{X})^{-}\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}\mathbf{p}$$
$$\widehat{\boldsymbol{\gamma}} = \widehat{\mathbf{G}}\mathbf{Z}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}\widehat{\mathbf{r}}$$

With these statistics, the pseudo-response and error weights of the linearized model are recomputed and the objective function is minimized again. The predictors $\widehat{\boldsymbol{\gamma}}$ are the estimated BLUPs in the approximated linear model. This process continues until the relative change between parameter estimates at two successive (outer) iterations is sufficiently small. See the PCONV= option in the PROC GLIMMIX statement for the computational details about how the GLIMMIX procedure compares parameter estimates across optimizations.

If the conditional distribution contains a scale parameter $\phi \neq 1$ (Table 38.15), the GLIMMIX procedure profiles this parameter in GLMMs from the log pseudo-likelihoods as well. To this end define

$$\mathbf{V}(\boldsymbol{\theta}^*) = \widetilde{\boldsymbol{\Delta}}^{-1}\mathbf{A}^{1/2}\mathbf{R}^*\mathbf{A}^{1/2}\widetilde{\boldsymbol{\Delta}}^{-1} + \mathbf{Z}\mathbf{G}^*\mathbf{Z}'$$

where $\boldsymbol{\theta}^*$ is the covariance parameter vector with $q-1$ elements. The matrices $\mathbf{G}^*$ and $\mathbf{R}^*$ are appropriately reparameterized versions of $\mathbf{G}$ and $\mathbf{R}$. For example, if $\mathbf{G}$ has a variance component structure and $\mathbf{R} = \phi\mathbf{I}$, then $\boldsymbol{\theta}^*$ contains ratios of the variance components and ϕ, and $\mathbf{R}^* = \mathbf{I}$. The solution for $\widehat{\phi}$ is

$$\widehat{\phi} = \widehat{\mathbf{r}}'\mathbf{V}(\widehat{\boldsymbol{\theta}}^*)^{-1}\widehat{\mathbf{r}}/m$$

where $m = f$ for MxPL and $m = f - k$ for RxPL. Substitution into the previous functions yields the profiled log pseudo-likelihoods,

$$l(\boldsymbol{\theta}^*, \mathbf{p}) = -\frac{1}{2}\log|\mathbf{V}(\boldsymbol{\theta}^*)| - \frac{f}{2}\log\left\{\mathbf{r}'\mathbf{V}(\boldsymbol{\theta}^*)^{-1}\mathbf{r}\right\} - \frac{f}{2}(1 + \log\{2\pi/f\})$$
$$l_R(\boldsymbol{\theta}^*, \mathbf{p}) = -\frac{1}{2}\log|\mathbf{V}(\boldsymbol{\theta}^*)| - \frac{f-k}{2}\log\left\{\mathbf{r}'\mathbf{V}(\boldsymbol{\theta}^*)^{-1}\mathbf{r}\right\}$$
$$-\frac{1}{2}\log|\mathbf{X}'\mathbf{V}(\boldsymbol{\theta}^*)^{-1}\mathbf{X}| - \frac{f-k}{2}(1 + \log\{2\pi/(f-k)\})$$

Profiling of ϕ can be suppressed with the NOPROFILE option in the PROC GLIMMIX statement.

Where possible, the objective function, its gradient, and its Hessian employ the sweep-based W-transformation (Hemmerle and Hartley 1973; Goodnight 1979; Goodnight and Hemmerle 1979). Further details about the minimization process in the general linear mixed model can be found in Wolfinger, Tobias, and Sall (1994).

Estimated Precision of Estimates

The GLIMMIX procedure produces estimates of the variability of $\widehat{\boldsymbol{\beta}}$, $\widehat{\boldsymbol{\theta}}$, and estimates of the prediction variability for $\widehat{\boldsymbol{\gamma}}$, $\mathrm{Var}[\widehat{\boldsymbol{\gamma}} - \boldsymbol{\gamma}]$. Denote as $\mathbf{S}$ the matrix

$$\mathbf{S} \equiv \widehat{\mathrm{Var}}[\mathbf{P}|\boldsymbol{\gamma}] = \widetilde{\boldsymbol{\Delta}}^{-1}\mathbf{A}^{1/2}\mathbf{R}\mathbf{A}^{1/2}\widetilde{\boldsymbol{\Delta}}^{-1}$$

where all components on the right side are evaluated at the converged estimates. The mixed model equations (Henderson 1984) in the linear mixed (pseudo-)model are then

$$\begin{bmatrix} \mathbf{X}'\mathbf{S}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{S}^{-1}\mathbf{Z} \\ \mathbf{Z}'\mathbf{S}^{-1}\mathbf{X} & \mathbf{Z}'\mathbf{S}^{-1}\mathbf{Z}+\mathbf{G}(\widehat{\boldsymbol{\theta}})^{-1} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{S}^{-1}\mathbf{p} \\ \mathbf{Z}'\mathbf{S}^{-1}\mathbf{p} \end{bmatrix}$$

and

$$\mathbf{C} = \begin{bmatrix} \mathbf{X}'\mathbf{S}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{S}^{-1}\mathbf{Z} \\ \mathbf{Z}'\mathbf{S}^{-1}\mathbf{X} & \mathbf{Z}'\mathbf{S}^{-1}\mathbf{Z}+\mathbf{G}(\widehat{\boldsymbol{\theta}})^{-1} \end{bmatrix}^{-}$$

$$= \begin{bmatrix} \widehat{\boldsymbol{\Omega}} & -\widehat{\boldsymbol{\Omega}}\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}\mathbf{Z}\mathbf{G}(\widehat{\boldsymbol{\theta}}) \\ -\mathbf{G}(\widehat{\boldsymbol{\theta}})\mathbf{Z}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}\mathbf{X}\widehat{\boldsymbol{\Omega}} & \mathbf{M}+\mathbf{G}(\widehat{\boldsymbol{\theta}})\mathbf{Z}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}\mathbf{X}\widehat{\boldsymbol{\Omega}}\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}\mathbf{Z}\mathbf{G}(\widehat{\boldsymbol{\theta}}) \end{bmatrix}$$

is the approximate estimated variance-covariance matrix of $[\widehat{\boldsymbol{\beta}}', \widehat{\boldsymbol{\gamma}}' - \boldsymbol{\gamma}']'$. Here, $\widehat{\boldsymbol{\Omega}} = (\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}\mathbf{X})^{-}$ and $\mathbf{M} = (\mathbf{Z}'\mathbf{S}^{-1}\mathbf{Z}+\mathbf{G}(\widehat{\boldsymbol{\theta}})^{-1})^{-1}$.

The square roots of the diagonal elements of $\widehat{\boldsymbol{\Omega}}$ are reported in the Standard Error column of the "Parameter Estimates" table. This table is produced with the SOLUTION option in the MODEL statement. The prediction standard errors of the random-effects solutions are reported in the Std Err Pred column of the "Solution for Random Effects" table. This table is produced with the SOLUTION option in the RANDOM statement.

As a cautionary note, $\mathbf{C}$ tends to underestimate the true sampling variability of $[\widehat{\boldsymbol{\beta}}', \widehat{\boldsymbol{\gamma}}']'$, because no account is made for the uncertainty in estimating $\mathbf{G}$ and $\mathbf{R}$. Although inflation factors have been proposed (Kackar and Harville 1984; Kass and Steffey 1989; Prasad and Rao 1990), they tend to be small for data sets that are fairly well balanced. PROC GLIMMIX does not compute any inflation factors by default. The DDFM=KENWARDROGER option in the MODEL statement prompts PROC GLIMMIX to compute a specific inflation factor (Kenward and Roger 1997), along with Satterthwaite-based degrees of freedom.

If $\mathbf{G}(\widehat{\boldsymbol{\theta}})$ is singular, or if you use the CHOL option of the PROC GLIMMIX statement, the mixed model equations are modified as follows. Let $\mathbf{L}$ denote the lower triangular matrix so that $\mathbf{L}\mathbf{L}' = \mathbf{G}(\widehat{\boldsymbol{\theta}})$. PROC GLIMMIX then solves the equations

$$\begin{bmatrix} \mathbf{X}'\mathbf{S}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{S}^{-1}\mathbf{Z}\mathbf{L} \\ \mathbf{L}'\mathbf{Z}'\mathbf{S}^{-1}\mathbf{X} & \mathbf{L}'\mathbf{Z}'\mathbf{S}^{-1}\mathbf{Z}\mathbf{L}+\mathbf{I} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\tau}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{S}^{-1}\mathbf{p} \\ \mathbf{L}'\mathbf{Z}'\mathbf{S}^{-1}\mathbf{p} \end{bmatrix}$$

and transforms $\widehat{\boldsymbol{\tau}}$ and a generalized inverse of the left-side coefficient matrix by using $\mathbf{L}$.

The asymptotic covariance matrix of the covariance parameter estimator $\widehat{\boldsymbol{\theta}}$ is computed based on the observed or expected Hessian matrix of the optimization procedure. Consider first the case where the scale parameter ϕ is not present or not profiled. Because $\boldsymbol{\beta}$ is profiled from the pseudo-likelihood, the objective function for minimization is $f(\boldsymbol{\theta}) = -2l(\boldsymbol{\theta}, \mathbf{p})$ for METHOD=MSPL and METHOD=MMPL and $f(\boldsymbol{\theta}) = -2l_R(\boldsymbol{\theta}, \mathbf{p})$ for METHOD=RSPL and METHOD=RMPL. Denote the observed Hessian (second derivative) matrix as

$$\mathbf{H} = \frac{\partial^2 f(\boldsymbol{\theta})}{\partial\boldsymbol{\theta}\,\partial\boldsymbol{\theta}'}$$

The GLIMMIX procedure computes the variance of $\widehat{\boldsymbol{\theta}}$ by default as $2\mathbf{H}^{-1}$. If the Hessian is not positive definite, a sweep-based generalized inverse is used instead. When the EXPHESSIAN option of

the PROC GLIMMIX statement is used, or when the procedure is in scoring mode at convergence (see the SCORING option in the PROC GLIMMIX statement), the observed Hessian is replaced with an approximated expected Hessian matrix in these calculations.

Following Wolfinger, Tobias, and Sall (1994), define the following components of the gradient and Hessian in the optimization:

$$\begin{aligned}
\mathbf{g}_1 &= \frac{\partial}{\partial \boldsymbol{\theta}} \mathbf{r}'\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{r} \\
\mathbf{H}_1 &= \frac{\partial^2}{\partial \boldsymbol{\theta}\, \partial \boldsymbol{\theta}'} \log\{\mathbf{V}(\boldsymbol{\theta})\} \\
\mathbf{H}_2 &= \frac{\partial^2}{\partial \boldsymbol{\theta}\, \partial \boldsymbol{\theta}'} \mathbf{r}'\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{r} \\
\mathbf{H}_3 &= \frac{\partial^2}{\partial \boldsymbol{\theta}\, \partial \boldsymbol{\theta}'} \log\{|\mathbf{X}'\mathbf{V}(\boldsymbol{\theta})^{-1}\mathbf{X}|\}
\end{aligned}$$

Table 38.17 gives expressions for the Hessian matrix $\mathbf{H}$ depending on estimation method, profiling, and scoring.

Table 38.17 Hessian Computation in GLIMMIX

Profiling	Scoring	M*x*PL	R*x*PL
No	No	$\mathbf{H}_1 + \mathbf{H}_2$	$\mathbf{H}_1 + \mathbf{H}_2 + \mathbf{H}_3$
No	Yes	$-\mathbf{H}_1$	$-\mathbf{H}_1 + \mathbf{H}_3$
No	Mod.	$-\mathbf{H}_1$	$-\mathbf{H}_1 - \mathbf{H}_3$
Yes	No	$\begin{bmatrix} \mathbf{H}_1 + \mathbf{H}_2/\phi & -\mathbf{g}_2/\phi^2 \\ -\mathbf{g}_2'/\phi^2 & f/\phi^2 \end{bmatrix}$	$\begin{bmatrix} \mathbf{H}_1 + \mathbf{H}_2/\phi + \mathbf{H}_3 & -\mathbf{g}_2/\phi^2 \\ -\mathbf{g}_2'/\phi^2 & (f-k)/\phi^2 \end{bmatrix}$
Yes	Yes	$\begin{bmatrix} -\mathbf{H}_1 & -\mathbf{g}_2/\phi^2 \\ -\mathbf{g}_2'/\phi^2 & f/\phi^2 \end{bmatrix}$	$\begin{bmatrix} -\mathbf{H}_1 + \mathbf{H}_3 & -\mathbf{g}_2/\phi^2 \\ -\mathbf{g}_2'/\phi^2 & (f-k)/\phi^2 \end{bmatrix}$
Yes	Mod.	$\begin{bmatrix} -\mathbf{H}_1 & -\mathbf{g}_2/\phi^2 \\ -\mathbf{g}_2'/\phi^2 & f/\phi^2 \end{bmatrix}$	$\begin{bmatrix} -\mathbf{H}_1 - \mathbf{H}_3 & -\mathbf{g}_2/\phi^2 \\ -\mathbf{g}_2'/\phi^2 & (f-k)/\phi^2 \end{bmatrix}$

The "Mod." expressions for the Hessian under scoring in R*x*PL estimation refer to a modified scoring method. In some cases, the modification leads to faster convergence than the standard scoring algorithm. The modification is requested with the SCOREMOD option in the PROC GLIMMIX statement.

Finally, in the case of a profiled scale parameter ϕ, the Hessian for the $(\boldsymbol{\theta}^*, \phi)$ parameterization is converted into that for the $\boldsymbol{\theta}$ parameterization as

$$\mathbf{H}(\boldsymbol{\theta}) = \mathbf{B}\mathbf{H}(\boldsymbol{\theta}^*, \phi)\mathbf{B}'$$

where

$$\mathbf{B} = \begin{bmatrix} 1/\phi & 0 & \cdots & 0 & 0 \\ 0 & 1/\phi & \cdots & 0 & 0 \\ 0 & \cdots & \cdots & 1/\phi & 0 \\ -\theta_1^*/\phi & -\theta_2^*/\phi & \cdots & -\theta_{q-1}^*/\phi & 1 \end{bmatrix}$$

Subject-Specific and Population-Averaged (Marginal) Expansions

There are two basic choices for the expansion locus of the linearization. A subject-specific (SS) expansion uses

$$\widetilde{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}} \quad \widetilde{\boldsymbol{\gamma}} = \widehat{\boldsymbol{\gamma}}$$

which are the current estimates of the fixed effects and estimated BLUPs. The population-averaged (PA) expansion expands about the same fixed effects and the expected value of the random effects

$$\widetilde{\boldsymbol{\beta}} = \widehat{\boldsymbol{\beta}} \quad \widetilde{\boldsymbol{\gamma}} = \mathbf{0}$$

To recompute the pseudo-response and weights in the SS expansion, the BLUPs must be computed every time the objective function in the linear mixed model is maximized. The PA expansion does not require any BLUPs. The four pseudo-likelihood methods implemented in the GLIMMIX procedure are the 2×2 factorial combination between two expansion loci and residual versus maximum pseudo-likelihood estimation. The following table shows the combination and the corresponding values of the METHOD= option (PROC GLIMMIX statement); METHOD=RSPL is the default.

Type of PL	Expansion Locus $\widehat{\boldsymbol{\gamma}}$	Expansion Locus $\mathrm{E}[\boldsymbol{\gamma}]$
residual	RSPL	RMPL
maximum	MSPL	MMPL

Maximum Likelihood Estimation Based on Laplace Approximation

Objective Function

Let $\boldsymbol{\beta}$ denote the vector of fixed-effects parameters and $\boldsymbol{\theta}$ the vector of covariance parameters. For Laplace estimation in the GLIMMIX procedure, $\boldsymbol{\theta}$ includes the G-side parameters and a possible scale parameter ϕ, provided that the conditional distribution of the data contains such a scale parameter. $\boldsymbol{\theta}^*$ is the vector of the G-side parameters.

The marginal distribution of the data in a mixed model can be expressed as

$$\begin{aligned} p(\mathbf{y}) &= \int p(\mathbf{y}|\boldsymbol{\gamma}, \boldsymbol{\beta}, \phi)\, p(\boldsymbol{\gamma}|\boldsymbol{\theta}^*)\, d\boldsymbol{\gamma} \\ &= \int \exp\left\{\log\{p(\mathbf{y}|\boldsymbol{\gamma}, \boldsymbol{\beta}, \phi)\} + \log\{p(\boldsymbol{\gamma}|\boldsymbol{\theta}^*)\}\right\} d\boldsymbol{\gamma} \\ &= \int \exp\left\{c_l\, f(\mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\theta}; \boldsymbol{\gamma})\right\} d\boldsymbol{\gamma} \end{aligned}$$

If the constant c_l is large, the Laplace approximation of this integral is

$$L(\boldsymbol{\beta}, \boldsymbol{\theta}; \widehat{\boldsymbol{\gamma}}, \mathbf{y}) = \left(\frac{2\pi}{c_l}\right)^{n_\gamma/2} |-f''(\mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\theta}; \widehat{\boldsymbol{\gamma}})|^{-1/2} \, e^{c_l f(\mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\theta}; \widehat{\boldsymbol{\gamma}})}$$

where n_γ is the number of elements in $\boldsymbol{\gamma}$, f'' is the second derivative matrix

$$f''(\mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\theta}; \widehat{\boldsymbol{\gamma}}) = \frac{\partial^2 f(\mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\theta}; \boldsymbol{\gamma})}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\gamma}'}|_{\widehat{\boldsymbol{\gamma}}}$$

and $\widehat{\boldsymbol{\gamma}}$ satisfies the first-order condition

$$\frac{\partial f(\mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\theta}; \boldsymbol{\gamma})}{\partial \boldsymbol{\gamma}} = \mathbf{0}$$

The objective function for Laplace parameter estimation in the GLIMMIX procedure is $-2\log\{L(\boldsymbol{\beta}, \boldsymbol{\theta} \;\; \widehat{\boldsymbol{\gamma}}, \mathbf{y})\}$. The optimization process is singly iterative, but because $\widehat{\boldsymbol{\gamma}}$ depends on $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\theta}}$, the GLIMMIX procedure solves a suboptimization problem to determine for given values of $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\theta}}$ the random-effects solution vector that maximizes $f(\mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\theta}; \boldsymbol{\gamma})$.

When you have longitudinal or clustered data with m independent subjects or clusters, the vector of observations can be written as $\mathbf{y} = [\mathbf{y}_1', \cdots, \mathbf{y}_m']'$, where $\mathbf{y}_i$ is an $n_i \times 1$ vector of observations for subject (cluster) i $(i = 1, \cdots, m)$. In this case, assuming conditional independence such that

$$p(\mathbf{y}_i|\boldsymbol{\gamma}_i) = \prod_{j=1}^{n_i} p(y_{ij}|\boldsymbol{\gamma}_i)$$

the marginal distribution of the data can be expressed as

$$\begin{aligned} p(\mathbf{y}) = \prod_{i=1}^{m} p(\mathbf{y}_i) &= \prod_{i=1}^{m} \int p(\mathbf{y}_i|\boldsymbol{\gamma}_i) p(\boldsymbol{\gamma}_i) \, d\boldsymbol{\gamma}_i \\ &= \prod_{i=1}^{m} \int \exp\{n_i f(\mathbf{y}_i, \boldsymbol{\beta}, \boldsymbol{\theta}; \boldsymbol{\gamma}_i)\} \, d\boldsymbol{\gamma}_i \end{aligned}$$

where

$$\begin{aligned} n_i f(\mathbf{y}_i, \boldsymbol{\beta}, \boldsymbol{\theta}; \boldsymbol{\gamma}_i) &= \log\{p(\mathbf{y}_i|\boldsymbol{\gamma}_i) \, p(\boldsymbol{\gamma}_i)\} \\ &= \sum_{j=1}^{n_i} \log\left\{p(y_{ij}|\boldsymbol{\gamma}_i)\right\} + n_i \log\{p(\boldsymbol{\gamma}_i)\} \end{aligned}$$

When the number of observations within a cluster, n_i, is large, the Laplace approximation to the ith individual's marginal probability density function is

$$\begin{aligned} p(\mathbf{y}_i|\boldsymbol{\beta}, \boldsymbol{\theta}) &= \int \exp\{n_i f(\mathbf{y}_i, \boldsymbol{\beta}, \boldsymbol{\theta}; \boldsymbol{\gamma}_i)\} \, d\boldsymbol{\gamma}_i \\ &= \frac{(2\pi)^{n_\gamma}/2}{|-n_i f''(\mathbf{y}_i, \boldsymbol{\beta}, \boldsymbol{\theta}; \widehat{\boldsymbol{\gamma}}_i)|^{-1/2}} \exp\{n_i f(\mathbf{y}_i, \boldsymbol{\beta}, \boldsymbol{\theta}; \widehat{\boldsymbol{\gamma}}_i)\} \end{aligned}$$

where $n_{\gamma i}$ is the common dimension of the random effects, $\boldsymbol{\gamma}_i$. In this case, provided that the constant $c_l = \min\{n_i\}$ is large, the Laplace approximation to the marginal log likelihood is

$$\log\{L(\boldsymbol{\beta},\boldsymbol{\theta};\widehat{\boldsymbol{\gamma}},\mathbf{y})\} = \sum_{i=1}^{m}\left\{n_i f(\mathbf{y},\boldsymbol{\beta},\boldsymbol{\theta};\widehat{\boldsymbol{\gamma}}_i) + \frac{n_{\gamma i}}{2}\log\{2\pi\} - \frac{1}{2}\log|-n_i f''(\boldsymbol{\beta},\boldsymbol{\theta};\widehat{\boldsymbol{\gamma}}_i)|\right\}$$

which serves as the objective function for the METHOD=LAPLACE estimator in PROC GLIMMIX.

The Laplace approximation implemented in the GLIMMIX procedure differs from that in Wolfinger (1993) and Pinheiro and Bates (1995) in important respects. Wolfinger (1993) assumed a flat prior for $\boldsymbol{\beta}$ and expanded the integrand around $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, leaving only the covariance parameters for the overall optimization. The "fixed" effects $\boldsymbol{\beta}$ and the random effects $\boldsymbol{\gamma}$ are determined in a suboptimization that takes the form of a linear mixed model step with pseudo-data. The GLIMMIX procedure involves only the random effects vector $\boldsymbol{\gamma}$ in the suboptimization. Pinheiro and Bates (1995) and Wolfinger (1993) consider a modified Laplace approximation that replaces the second derivative $f''(\mathbf{y},\boldsymbol{\beta},\boldsymbol{\theta};\widehat{\boldsymbol{\gamma}})$ with an (approximate) expected value, akin to scoring. The GLIMMIX procedure does not use an approximation to $f''(\mathbf{y},\boldsymbol{\beta},\boldsymbol{\theta};\widehat{\boldsymbol{\gamma}})$. The METHOD=RSPL estimates in PROC GLIMMIX are equivalent to the estimates obtained with the modified Laplace approximation in Wolfinger (1993). The objective functions of METHOD=RSPL and Wolfinger (1993) differ in a constant that depends on the number of parameters.

Asymptotic Properties and the Importance of Subjects

Suppose that the GLIMMIX procedure processes your data by subjects (see the section "Processing by Subjects" on page 2247) and let n_i denote the number of observations per subject, $i = 1,\ldots,s$. Arguments in Vonesh (1996) show that the maximum likelihood estimator based on the Laplace approximation is a consistent estimator to order $O_p\{\max\{1/\sqrt{s}\}, 1/\min\{n_i\}\}$. In other words, as the number of subjects and the number of observations per subject grows, the small-sample bias of the Laplace estimator disappears. Note that the term involving the number of subjects in this maximum relates to standard asymptotic theory, and the term involving the number of observations per subject relates to the accuracy of the Laplace approximation (Vonesh 1996). In the case where random effects enter the model linearly, the Laplace approximation is exact and the requirement that $\min\{n_i\} \to \infty$ can be dropped.

If your model is not processed by subjects but is equivalent to a subject model, the asymptotics with respect to s still apply, because the Hessian matrix of the suboptimization for $\boldsymbol{\gamma}$ breaks into s separate blocks. For example, the following two models are equivalent with respect to s and n_i, although only for the first model does PROC GLIMMIX process the data explicitly by subjects:

```
proc glimmix method=laplace;
   class sub A;
   model y = A;
   random intercept / subject=sub;
run;
```

```
proc glimmix method=laplace;
   class sub A;
   model y = A;
   random sub;
run;
```

The same holds, for example, for models with independent nested random effects. The following two models are equivalent, and you can derive asymptotic properties related to s and $\min\{n_i\}$ from the model in the first run:

```
proc glimmix method=laplace;
   class A B block;
   model y = A B A*B;
   random intercept A / subject=block;
run;

proc glimmix method=laplace;
   class A B block;
   model y = A B A*B;
   random block a*block;
run;
```

The Laplace approximation requires that the dimension of the integral does not increase with the size of the sample. Otherwise the error of the likelihood approximation does not diminish with n_i. This is the case, for example, with exchangeable arrays (Shun and McCullagh 1995), crossed random effects (Shun 1997), and correlated random effects of arbitrary dimension (Raudenbush, Yang, and Yosef 2000). Results in Shun (1997), for example, show that even in this case the standard Laplace approximation has smaller bias than pseudo-likelihood estimates.

Maximum Likelihood Estimation Based on Adaptive Quadrature

Quadrature methods, like the Laplace approximation, approximate integrals. If you choose METHOD=QUAD for a generalized linear mixed model, the GLIMMIX procedure approximates the marginal log likelihood with an adaptive Gauss-Hermite quadrature rule. Gaussian quadrature is particularly well suited to numerically evaluate integrals against probability measures (Lange 1999, Ch. 16). And Gauss-Hermite quadrature is appropriate when the density has kernel $\exp\{-x^2\}$ and integration extends over the real line, as is the case for the normal distribution. Suppose that $p(x)$ is a probability density function and the function $f(x)$ is to be integrated against it. Then the quadrature rule is

$$\int_{-\infty}^{\infty} f(x)p(x)\,dx \approx \sum_{i=1}^{N} w_i f(x_i)$$

where N denotes the number of quadrature points, the w_i are the quadrature weights, and the x_i are the abscissas. The Gaussian quadrature chooses abscissas in areas of high density, and if $p(x)$ is continuous, the quadrature rule is exact if $f(x)$ is a polynomial of up to degree $2N-1$. In the generalized linear mixed model the roles of $f(x)$ and $p(x)$ are played by the conditional distribution of the data given the random effects, and the random-effects distribution, respectively. Quadrature

abscissas and weights are those of the standard Gauss-Hermite quadrature (Golub and Welsch 1969; see also Table 25.10 of Abramowitz and Stegun 1972; Evans 1993).

A numerical integration rule is called adaptive when it uses a variable step size to control the error of the approximation. For example, an adaptive trapezoidal rule uses serial splitting of intervals at midpoints until a desired tolerance is achieved. The quadrature rule in the GLIMMIX procedure is adaptive in the following sense: if you do not specify the number of quadrature points (nodes) with the QPOINTS= suboption of the METHOD=QUAD option, then the number of quadrature points is determined by evaluating the log likelihood at the starting values at a successively larger number of nodes until a tolerance is met (for more details see the text under the heading "Starting Values" in the next section). Furthermore, the GLIMMIX procedure centers and scales the quadrature points by using the empirical Bayes estimates (EBEs) of the random effects and the Hessian (second derivative) matrix from the EBE suboptimization. This centering and scaling improves the likelihood approximation by placing the abscissas according to the density function of the random effects. It is not, however, adaptiveness in the previously stated sense.

Objective Function

Let $\boldsymbol{\beta}$ denote the vector of fixed-effects parameters and $\boldsymbol{\theta}$ the vector of covariance parameters. For quadrature estimation in the GLIMMIX procedure, $\boldsymbol{\theta}$ includes the G-side parameters and a possible scale parameter ϕ, provided that the conditional distribution of the data contains such a scale parameter. $\boldsymbol{\theta}^*$ is the vector of the G-side parameters. The marginal distribution of the data for subject i in a mixed model can be expressed as

$$p(\mathbf{y}_i) = \int \cdots \int p(\mathbf{y}_i|\boldsymbol{\gamma}_i, \boldsymbol{\beta}, \phi)\; p(\boldsymbol{\gamma}_i|\boldsymbol{\theta}^*)\; d\boldsymbol{\gamma}_i$$

Suppose N_q denotes the number of quadrature points in each dimension (for each random effect) and r denotes the number of random effects. For each subject, obtain the empirical Bayes estimates of $\boldsymbol{\gamma}_i$ as the vector $\widehat{\boldsymbol{\gamma}}_i$ that minimizes

$$-\log\left\{p(\mathbf{y}_i|\boldsymbol{\gamma}_i, \boldsymbol{\beta}, \phi)\, p(\boldsymbol{\gamma}_i|\boldsymbol{\theta}^*)\right\} = f(\mathbf{y}_i, \boldsymbol{\beta}, \boldsymbol{\theta}; \boldsymbol{\gamma}_i)$$

If $\mathbf{z} = [z_1, \cdots, z_{N_q}]$ are the standard abscissas for Gauss-Hermite quadrature, and $\mathbf{z}_j^* = [z_{j_1}, \cdots, z_{j_r}]$ is a point on the r-dimensional quadrature grid, then the centered and scaled abscissas are

$$\mathbf{a}_j^* = \widehat{\boldsymbol{\gamma}}_i + 2^{1/2} f''(\mathbf{y}_i, \boldsymbol{\beta}, \boldsymbol{\theta}; \widehat{\boldsymbol{\gamma}}_i)^{-1/2} \mathbf{z}_j^*$$

As for the Laplace approximation, f'' is the second derivative matrix with respect to the random effects,

$$f''(\mathbf{y}_i, \boldsymbol{\beta}, \boldsymbol{\theta}; \widehat{\boldsymbol{\gamma}}_i) = \frac{\partial^2 f(\mathbf{y}_i, \boldsymbol{\beta}, \boldsymbol{\theta}; \boldsymbol{\gamma}_i)}{\partial \boldsymbol{\gamma}_i \partial \boldsymbol{\gamma}_i'}\Big|_{\widehat{\boldsymbol{\gamma}}_i}$$

These centered and scaled abscissas, along with the Gauss-Hermite quadrature weights $\mathbf{w} = [w_1, \cdots, w_{N_q}]$, are used to construct the r-dimensional integral by a sequence of one-dimensional

rules

$$
\begin{aligned}
p(\mathbf{y}_i) &= \int \cdots \int p(\mathbf{y}_i|\boldsymbol{\gamma}_i, \boldsymbol{\beta}, \phi)\; p(\boldsymbol{\gamma}_i|\boldsymbol{\theta}^*)\; d\boldsymbol{\gamma}_i \\
&\approx 2^{r/2} |f''(\mathbf{y}_i, \boldsymbol{\beta}, \boldsymbol{\theta}; \widehat{\boldsymbol{\gamma}}_i)|^{-1/2} \\
&\sum_{j_1=1}^{N_q} \cdots \sum_{j_r=1}^{N_q} \left[p(\mathbf{y}_i|\mathbf{a}_j^*, \boldsymbol{\beta}, \phi) p(\mathbf{a}_j^*|\boldsymbol{\theta}^*) \prod_{k=1}^{r} w_{jk} \exp z_{j_k}^2 \right]
\end{aligned}
$$

The right-hand side of this expression, properly accumulated across subjects, is the objective function for adaptive quadrature estimation in the GLIMMIX procedure.

Quadrature or Laplace Approximation

If you select the quadrature rule with a single quadrature point, namely

```
proc glimmix method=quad(qpoints=1);
```

the results will be identical to METHOD=LAPLACE. Computationally, the two methods are not identical, however. METHOD=LAPLACE can be applied to a considerably larger class of models. For example, crossed random effects, models without subjects, or models with non-nested subjects can be handled with the Laplace approximation but not with quadrature. Furthermore, METHOD=LAPLACE draws on a number of computational simplifications that can increase its efficiency compared to a quadrature algorithm with a single node. For example, the Laplace approximation is possible with unbounded covariance parameter estimates (NOBOUND option in the PROC GLIMMIX statement) and can permit certain types of negative definite or indefinite **G** matrices. The adaptive quadrature approximation with scaled abscissas typically breaks down when **G** is not at least positive semidefinite.

As the number of random effects grows—for example, if you have nested random effects—quadrature quickly becomes computationally infeasible, due to the high dimensionality of the integral. To this end it is worthwhile to clarify the issues of dimensionality and computational effort as related to the number of quadrature nodes. Suppose that the A effect has 4 levels and consider the following statements:

```
proc glimmix method=quad(qpoints=5);
   class A id;
   model y = / dist=negbin;
   random A / subject=id;
run;
```

For each subject, computing the marginal log likelihood requires the numerical evaluation of a four-dimensional integral. As part of this evaluation $5^4 = 625$ conditional log likelihoods need to be computed for each observation on each pass through the data. As the number of quadrature points or the number of random effects increases, this constitutes a sizable computational effort. Suppose, for example, that an additional random effect with $b = 2$ levels is added as an interaction. The following statements then require evaluation of $5^{(4+8)} = 244140625$ conditional log likelihoods for each observation one each pass through the data:

```
proc glimmix method=quad(qpoints=5);
   class A B id;
   model y = / dist=negbin;
   random A A*B / subject=id;
run;
```

As the number of random effects increases, Laplace approximation presents a computationally more expedient alternative.

If you wonder whether METHOD=LAPLACE would present a viable alternative to a model that you can fit with METHOD=QUAD, the "Optimization Information" table can provide some insights. The table contains as its last entry the number of quadrature points determined by PROC GLIMMIX to yield a sufficiently accurate approximation of the log likelihood (at the starting values). In many cases, a single quadrature node is sufficient, in which case the estimates are identical to those of METHOD=LAPLACE.

Aspects Common to Adaptive Quadrature and Laplace Approximation

Estimated Precision of Estimates

Denote as $\mathbf{H}$ the second derivative matrix

$$\mathbf{H} = -\frac{\partial^2 \log\{L(\boldsymbol{\beta}, \boldsymbol{\theta}\ \widehat{\boldsymbol{\gamma}})\}}{\partial[\boldsymbol{\beta}, \boldsymbol{\theta}]\partial[\boldsymbol{\beta}', \boldsymbol{\theta}']}$$

evaluated at the converged solution of the optimization process. Partition its inverse as

$$\mathbf{H}^{-1} = \begin{bmatrix} \mathbf{C}(\boldsymbol{\beta}, \boldsymbol{\beta}) & \mathbf{C}(\boldsymbol{\beta}, \boldsymbol{\theta}) \\ \mathbf{C}(\boldsymbol{\theta}, \boldsymbol{\beta}) & \mathbf{C}(\boldsymbol{\theta}, \boldsymbol{\theta}) \end{bmatrix}$$

For METHOD=LAPLACE and METHOD=QUAD, the GLIMMIX procedure computes $\mathbf{H}$ by finite forward differences based on the analytic gradient of $\log\{L(\boldsymbol{\beta}, \boldsymbol{\theta}\ \widehat{\boldsymbol{\gamma}})\}$. The partition $\mathbf{C}(\boldsymbol{\theta}, \boldsymbol{\theta})$ serves as the asymptotic covariance matrix of the covariance parameter estimates (ASYCOV option in the PROC GLIMMIX statement). The standard errors reported in the "Covariance Parameter Estimates" table are based on the diagonal entries of this partition.

If you request an empirical standard error matrix with the EMPIRICAL option in the PROC GLIMMIX statement, a likelihood-based sandwich estimator is computed based on the subject-specific gradients of the Laplace or quadrature approximation. The sandwich estimator then replaces $\mathbf{H}^{-1}$ in calculations following convergence.

To compute the standard errors and prediction standard errors of linear combinations of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, PROC GLIMMIX forms an approximate prediction variance matrix for $[\widehat{\boldsymbol{\beta}}, \widehat{\boldsymbol{\gamma}}]'$ from

$$\mathbf{P} = \begin{bmatrix} \mathbf{H}^{-1} & \mathbf{H}^{-1}\left(\frac{\partial \widehat{\boldsymbol{\gamma}}}{\partial[\boldsymbol{\beta}, \boldsymbol{\theta}]}\right) \\ \left(\frac{\partial \widehat{\boldsymbol{\gamma}}}{\partial[\boldsymbol{\beta}', \boldsymbol{\theta}']}\right)\mathbf{H}^{-1} & \boldsymbol{\Gamma}^{-1} + \left(\frac{\partial \widehat{\boldsymbol{\gamma}}}{\partial[\boldsymbol{\beta}', \boldsymbol{\theta}']}\right)\mathbf{H}^{-1}\left(\frac{\partial \widehat{\boldsymbol{\gamma}}}{\partial[\boldsymbol{\beta}, \boldsymbol{\theta}]}\right) \end{bmatrix}$$

where $\boldsymbol{\Gamma}$ is the second derivative matrix from the $\boldsymbol{\gamma}$ suboptimization that maximizes $f(\mathbf{y}, \boldsymbol{\beta}, \boldsymbol{\theta}; \boldsymbol{\gamma})$ for given values of $\boldsymbol{\beta}$ and $\boldsymbol{\theta}$. The prediction variance submatrix for the random effects is based

on approximating the conditional mean squared error of prediction as in Booth and Hobert (1998). Note that even in the normal linear mixed model, the approximate conditional prediction standard errors are not identical to the prediction standard errors you obtain by inversion of the mixed model equations.

Conditional Fit and Output Statistics

When you estimate the parameters of a mixed model by Laplace approximation or quadrature, the GLIMMIX procedure displays fit statistics related to the marginal distribution as well as the conditional distribution $p(\mathbf{y}|\widehat{\boldsymbol{\gamma}}, \widehat{\boldsymbol{\beta}}, \widehat{\phi})$. For ODS purposes, the name of the "Conditional Fit Statistics" table is "CondFitStatistics." Because the marginal likelihood is approximated numerically for these methods, statistics based on the marginal distribution are not available. Instead of the generalized Pearson chi-square statistic in the "Fit Statistics" table, PROC GLIMMIX reports the Pearson statistic of the conditional distribution in the "Conditional Fit Statistics" table.

The unavailability of the marginal distribution also affects the set of output statistics that can be produced with METHOD=LAPLACE and METHOD=QUAD. Output statistics and statistical graphics that depend on the marginal variance of the data are not available with these estimation methods.

User-Defined Variance Function

If you provide your own variance function, PROC GLIMMIX generally assumes that the (conditional) distribution of the data is unknown. Laplace or quadrature estimation would then not be possible. When you specify a variance function with METHOD=LAPLACE or METHOD=QUAD, the procedure assumes that the conditional distribution is normal. For example, consider the following statements to fit a mixed model to count data:

```
proc glimmix method=laplace;
   class sub;
   _variance_ = _phi_*_mu_;
   model count = x / s link=log;
   random int / sub=sub;
run;
```

The variance function and the link suggest an overdispersed Poisson model. The Poisson distribution cannot accommodate the extra scale parameter _PHI_, however. In this situation, the GLIMMIX procedure fits a mixed model with random intercepts, log link function, and variance function $\phi\mu$, assuming that the count variable is normally distributed, given the random effects.

Starting Values

Good starting values for the fixed effects and covariance parameters are important for Laplace and quadrature methods because the process commences with a suboptimization in which the empirical Bayes estimates of the random effects must be obtained before the optimization can get under way. Furthermore, the starting values are important for the adaptive choice of the number of quadrature points.

If you choose METHOD=LAPLACE or METHOD=QUAD and you do not provide starting values for the covariance parameters through the PARMS statement, the GLIMMIX procedure determines starting values in the following steps.

1. A GLM is fit initially to obtain starting values for the fixed-effects parameters. No output is produced from this stage. The number of initial iterations of this GLM fit can be controlled with the INITITER= option in the PROC GLIMMIX statement. You can suppress this step with the NOINITGLM option in the PROC GLIMMIX statement.

2. Given the fixed-effects estimates, starting values for the covariance parameters are computed by a MIVQUE0 step (Goodnight 1978b).

3. For METHOD=QUAD you can follow these steps with several pseudo-likelihood updates to improve on the estimates and to obtain solutions for the random effects. The number of pseudo-likelihood steps is controlled by the INITPL= suboption of METHOD=QUAD.

4. For METHOD=QUAD, if you do not specify the number of quadrature points with the suboptions of the METHOD option, the GLIMMIX procedure attempts to determine a sufficient number of points adaptively as follows. Suppose that N_q denotes the number of nodes in each dimension. If N_{min} and N_{max} denote the values from the QMIN= and QMAX= suboptions, respectively, the sequence for values less than 11 is constructed in increments of 2 starting at N_{min}. Values greater than 11 are incremented in steps of r. The default value is $r = 10$. The default sequence, without specifying the QMIN=, QMAX=, or QFAC= option, is thus $1, 3, 5, 7, 9, 11, 21, 31$. If the relative difference of the log-likelihood approximation for two values in the sequence is less than the QTOL=t value (default $t = 0.0001$), the GLIMMIX procedure uses the lesser value for N_q in the subsequent optimization. If the relative difference does not fall below the tolerance t for any two subsequent values in the sequence, no estimation takes place.

Notes on Bias of Estimators

Generalized linear mixed models are nonlinear models, and the estimation techniques rely on approximations to the log likelihood or approximations of the model. It is thus not surprising that the estimates of the covariance parameters and the fixed effects are usually not unbiased. Whenever estimates are biased, questions arise about the magnitude of the bias, its dependence on other model quantities, and the order of the bias. The order is important because it determines how quickly the bias vanishes while some aspect of the data increases. Typically, studies of asymptotic properties in models for hierarchical data suppose that the number of subjects (clusters) tends to infinity while the size of the clusters is held constant or grows at a particular rate. Note that asymptotic results so established do not extend to designs with fully crossed random effects, for example.

The following paragraphs summarize some important findings from the literature regarding the bias in covariance parameter and fixed-effects estimates with pseudo-likelihood, Laplace, and adaptive quadrature methods. The remarks draw in particular on results in Breslow and Lin (1995), Lin and Breslow (1996), and Pinheiro and Chao (2006). Breslow and Lin (1995) and Lin and Breslow (1996) study the "worst case" scenario of binary responses in a matched-pairs design. Their models have a variance component structure, comprising either a single variance component (a subject-specific random intercept; Breslow and Lin 1995) or a diagonal $\mathbf{G}$ matrix (Lin and Breslow 1996).

They study the bias in the estimates of the fixed-effects $\boldsymbol{\beta}$ and the covariance parameters $\boldsymbol{\theta}$ when the variance components are near the origin and for a canonical link function.

The matched-pairs design gives rise to a generalized linear mixed model with a cluster (subject) size of 2. Recall that the pseudo-likelihood methods rely on a linearization and a probabilistic assumption that the pseudo-data so obtained follow a normal linear mixed model. Obviously, it is difficult to imagine how the subject-specific (conditional) distribution would follow a normal linear mixed models with binary data in a cluster size of 2. The bias in the pseudo-likelihood estimator of $\boldsymbol{\beta}$ is of order $||\boldsymbol{\theta}||$. The bias for the Laplace estimator of $\boldsymbol{\beta}$ is of smaller magnitude; its asymptotic bias has order $||\boldsymbol{\theta}||^2$.

The Laplace methods and the pseudo-likelihood method produce biased estimators of the variance component θ for the model considered in Breslow and Lin (1995). The order of the asymptotic bias for both estimation methods is θ, as θ approaches zero. Breslow and Lin (1995) comment on the fact that even with matched pairs, the bias vanishes very quickly in the binomial setting. If the conditional mean in the two groups is equal to 0.5, then the asymptotic bias factor of the pseudo-likelihood estimator is $1-1/(2n)$, where n is the binomial denominator. This term goes to 1 quickly as n increases. This result underlines the importance of grouping binary observations into binomial responses whenever possible.

The results of Breslow and Lin (1995) and Lin and Breslow (1996) are echoed in the simulation study in Pinheiro and Chao (2006). These authors also consider adaptive quadrature in models with nested, hierarchical, random effects and show that adaptive quadrature with a sufficient number of nodes leads to nearly unbiased—or least biased—estimates. Their results also show that results for binary data cannot so easily be ported to other distributions. Even with a cluster size of 2, the pseudo-likelihood estimates of fixed effects and covariance parameters are virtually unbiased in their simulation of a Poisson GLMM. Breslow and Lin (1995) and Lin and Breslow (1996) "eschew" the residual PL version (METHOD=RSPL) over the maximum likelihood form (METHOD=MSPL). Pinheiro and Chao (2006) consider both forms in their simulation study. As expected, the residual form shows less bias than the MSPL form, for the same reasons REML estimation leads to less biased estimates compared to ML estimation in linear mixed models. The gain is modest, however; see, for example, Table 1 in Pinheiro and Chao (2006). When the variance components are small, there is a sufficient number of observations per cluster, and a reasonable number of clusters, then pseudo-likelihood methods for binary data are very useful—they provide a computational expedient alternative to numerical integration, and they allow the incorporation of R-side covariance structure into the model. Because many group randomized trials involve many observations per group and small random effects variances, Murray et al. (2004) term questioning the use of conditional models for trials with binary outcome an "overreaction."

GLM Mode or GLMM Mode

The GLIMMIX procedure knows two basic modes of parameter estimation, and it can be important for you to understand the differences between the two modes.

In GLM mode, the data are never correlated and there can be no G-side random effects. Typical examples are logistic regression and normal linear models. When you fit a model in GLM mode, the METHOD= option in the PROC GLIMMIX statement has no effect. PROC GLIMMIX esti-

mates the parameters of the model by maximum likelihood, (restricted) maximum likelihood, or quasi-likelihood, depending on the distributional properties of the model (see the section "Default Estimation Techniques" on page 2273). The "Model Information" table tells you which estimation method was applied. In GLM mode, the individual observations are considered the sampling units. This has bearing, for example, on how sandwich estimators are computed (see the EMPIRICAL option and the section "Empirical Covariance ("Sandwich") Estimators" on page 2243).

In GLMM mode, the procedure assumes that the model contains random effects or possibly correlated errors, or that the data have a clustered structure. The parameters are then estimated by the techniques specified with the METHOD= option in the PROC GLIMMIX statement.

In general, adding one overdispersion parameter to a generalized linear model does not trigger the GLMM mode. For example, the model defined by the following statements is fit in GLM mode:

```
proc glimmix;
   model y = x1 x2 / dist=poisson;
   random _residual_;
run;
```

The parameters of the fixed effects are estimated by maximum likelihood, and the covariance matrix of the fixed-effects parameters is adjusted by the overdispersion parameter.

In a model with uncorrelated data you can trigger the GLMM mode by specifying a SUBJECT= or GROUP= effect in the RANDOM statement. For example, the following statements fit the model by using the residual pseudo-likelihood algorithm:

```
proc glimmix;
   class id;
   model y = x1 x2 / dist=poisson;
   random _residual_ / subject=id;
run;
```

If in doubt, you can determine whether a model was fit in GLM mode or GLMM mode. In GLM mode the "Covariance Parameter Estimates" table is not produced. Scale and dispersion parameters in the model appear in the "Parameter Estimates" table.

Statistical Inference for Covariance Parameters

The Likelihood Ratio Test

The likelihood ratio test (LRT) compares the likelihoods of two models where parameter estimates are obtained in two parameter spaces, the space Ω and the restricted subspace Ω_0. In the GLIMMIX procedure, the full model defines Ω and the *test-specification* in the COVTEST statement determines the null parameter space Ω_0. The likelihood ratio procedure consists of the following steps (see, for example, Bickel and Doksum 1977, p. 210):

1. Find the estimate $\widehat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta} \in \Omega$. Compute the likelihood $L(\widehat{\boldsymbol{\theta}})$.

2. Find the estimate $\widehat{\boldsymbol{\theta}}_0$ of $\boldsymbol{\theta} \in \Omega_0$. Compute the likelihood $L(\widehat{\boldsymbol{\theta}}_0)$.
3. Form the likelihood ratio

$$\overline{\lambda} = \frac{L(\widehat{\boldsymbol{\theta}})}{L(\widehat{\boldsymbol{\theta}}_0)}$$

4. Find a function $f\left(\overline{\lambda}\right)$ that has a known distribution. $f(\cdot)$ serves as the test statistic for the likelihood ratio test.

Please note the following regarding the implementation of these steps in the COVTEST statement of the GLIMMIX procedure.

- The function $f(\cdot)$ in step 4 is always taken to be

$$\lambda = 2\log\left\{\overline{\lambda}\right\}$$

  which is twice the difference between the log likelihoods for the full model and the model under the COVTEST restriction.
- For METHOD=RSPL and METHOD=RMPL, the test statistic is based on the restricted likelihood.
- For GLMMs involving pseudo-data, the test statistics are based on the pseudo-likelihood or the restricted pseudo-likelihood and are based on the final pseudo-data.
- The parameter space Ω for the full model is typically not an unrestricted space. The GLIMMIX procedure imposes boundary constraints for variance components and scale parameters, for example. The specification of the subspace Ω_0 must be consistent with these full-model constraints; otherwise the test statistic λ does not have the needed distribution. You can remove the boundary restrictions with the NOBOUND option in the PROC GLIMMIX statement or the NOBOUND option in the PARMS statement.

One- and Two-Sided Testing, Mixture Distributions

Consider testing the hypothesis $H_0: \theta_i = 0$. If Ω is the open interval $(0, \infty)$, then only a one-sided alternative hypothesis is meaningful,

$$H_0: \theta_i = 0 \quad H_a: \theta_i > 0$$

This is the appropriate set of hypotheses, for example, when θ_i is the variance of a G-side random effect. The positivity constraint on Ω is required for valid conditional and marginal distributions of the data. Verbeke and Molenberghs (2003) refer to this situation as the constrained case.

However, if one focuses on the validity of the marginal distribution alone, then negative values for θ_i might be permissible, provided that the marginal variance remains positive definite. In the vernacular or Verbeke and Molenberghs (2003), this is the unconstrained case. The appropriate alternative hypothesis is then two-sided,

$$H_0: \theta_i = 0 \quad H_a: \theta_i \neq 0$$

Several important issues are connected to the choice of hypotheses. The GLIMMIX procedure by default imposes constraints on some covariance parameters. For example, variances and scale parameters have a lower bound of 0. This implies a contrained setting with one-sided alternatives. If you specify the NOBOUND option in the PROC GLIMMIX statement, or the NOBOUND option in the PARMS statement, the boundary restrictions are lifted from the covariance parameters and the GLIMMIX procedure takes an unconstrained stance in the sense of Verbeke and Molenberghs (2003). The alternative hypotheses for variance components are then two-sided.

When $H_0: \theta_i = 0$ and $\Omega = (0, \infty)$, the value of θ_i under the null hypothesis is on the boundary of the parameter space. The distribution of the likelihood ratio test statistic λ is then nonstandard. In general, it is a mixture of distributions, and in certain special cases, it is a mixture of central chi-square distributions. Important contributions to the understanding of the asymptotic behavior of the likelihood ratio and score test statistic in this situation have been made by, for example, Self and Liang (1987), Shapiro (1988), and Silvapulle and Silvapulle (1995). Stram and Lee (1994, 1995) applied the results of Self and Liang (1987) to likelihood ratio testing in the mixed model with uncorrelated errors. Verbeke and Molenberghs (2003) compared the score and likelihood ratio tests in random effects models with unstructured **G** matrix and provide further results on mixture distributions.

The GLIMMIX procedure recognizes the following special cases in the computation of p-values ($\widehat{\lambda}$ denotes the realized value of the test statistic). Notice that the probabilities of general chi-square mixture distributions do not equal linear combination of central chi-square probabilities (Davis 1977; Johnson, Kotz, and Balakrishnan 1994, Section 18.8).

1. ν parameters are tested, and neither parameters specified under H_0 nor nuisance parameters are on the boundary of the parameters space (Case 4 in Self and Liang 1987). The p-value is computed by the classical result:

$$p = \Pr\left(\chi^2_\nu \geq \widehat{\lambda}\right)$$

2. One parameter is specified under H_0 and it falls on the boundary. No other parameters are on the boundary (Case 5 in Self and Liang 1987).

$$p = \begin{cases} 1 & \widehat{\lambda} = 0 \\ 0.5 \Pr\left(\chi^2_1 \geq \widehat{\lambda}\right) & \widehat{\lambda} > 0 \end{cases}$$

   Note that this implies a 50:50 mixture of a χ^2_0 and a χ^2_1 distribution. This is also Case 1 in Verbeke and Molenberghs (2000, p. 69).

3. Two parameters are specified under H_0, and one falls on the boundary. No nuisance parameters are on the boundary (Case 6 in Self and Liang 1987).

$$p = 0.5 \Pr\left(\chi^2_1 \geq \widehat{\lambda}\right) + 0.5 \Pr\left(\chi^2_2 \geq \widehat{\lambda}\right)$$

   A special case of this scenario is the addition of a random effect to a model with a single random effect and unstructured covariance matrix (Case 2 in Verbeke and Molenberghs 2000, p. 70).

4. Removing j random effects from $j + k$ uncorrelated random effects (Verbeke and Molenberghs 2003).

$$p = 2^{-j} \sum_{i=0}^{j} \left(\begin{array}{c} j \\ i \end{array} \right) \Pr\left(\chi_i^2 \geq \widehat{\lambda}\right)$$

   Note that this case includes the case of testing a single random effects variance against zero, which leads to a 50:50 mixture of a χ_0^2 and a χ_1^2 as in 2.

5. Removing a random effect from an unstructured **G** matrix (Case 3 in Verbeke and Molenberghs 2000, p. 71).

$$p = 0.5 \Pr\left(\chi_k^2 \geq \widehat{\lambda}\right) + 0.5 \Pr\left(\chi_{k-1}^2 \geq \widehat{\lambda}\right)$$

   where k is the number of random effects (columns of **G**) in the full model. Case 5 in Self and Liang (1987) describes a special case.

When the GLIMMIX procedure determines that estimates of nuisance parameters (parameters not specified under H_0) fall on the boundary, no mixture results are computed.

You can request that the procedure not use mixtures with the CLASSICAL option in the COVTEST statement. If mixtures are used, the Note column of the "Likelihood Ratio Tests of Covariance Parameters" table contains the "MI" entry. The "DF" entry is used when PROC GLIMMIX determines that the standard computation of p-values is appropriate. The "–" entry is used when the classical computation was used because the testing and model scenario does not match one of the special cases described previously.

Handling the Degenerate Distribution

Likelihood ratio testing in mixed models invariably involves the chi-square distribution with zero degrees of freedom. The χ_0^2 random variable is degenerate at 0, and it occurs in two important circumstances. First, it is a component of mixtures, where typically the value of the test statistic is not zero. In that case, the contribution of the χ_0^2 component of the mixture to the p-value is nil. Second, a degenerate distribution of the test statistic occurs when the null model is identical to the full model—for example, if you test a hypothesis that does not impose any (new) constraints on the parameter space. The following statements test whether the **R** matrix in a variance component model is diagonal:

```
proc glimmix;
   class a b;
   model y = a;
   random b a*b;
   covtest diagR;
run;
```

Because no R-side covariance structure is specified (all random effects are G-side effects), the **R** matrix is diagonal in the full model and the COVTEST statement does not impose any further restriction on the parameter space. The likelihood ratio test statistic is zero. The GLIMMIX procedure

computes the p-value as the probability to observe a value at least as large as the test statistic under the null hypothesis. Hence,

$$p = \Pr(\chi_0^2 \geq 0) = 1$$

Wald Versus Likelihood Ratio Tests

The Wald test and the likelihood ratio tests are asymptotic tests, meaning that the distribution from which p-values are calculated for a finite number of samples draws on the distribution of the test statistic as the sample size grows to infinity. The Wald test is a simple test that is easy to compute based only on parameter estimates and their (asymptotic) standard errors. The likelihood ratio test, on the other hand, requires the likelihoods of the full model and the model reduced under H_0. It is computationally more demanding, but also provides the asymptotically more powerful and reliable test. The likelihood ratio test is almost always preferable to the Wald test, unless computational demands make it impractical to refit the model.

Confidence Bounds Based on Likelihoods

Families of statistical tests can be inverted to produce confidence limits for parameters. The confidence region for parameter θ is the set of values for which the corresponding test fails to reject $H: \theta = \theta_0$. When parameters are estimated by maximum likelihood or a likelihood-based technique, it is natural to consider the likelihood ratio test statistic for H in the test inversion. When there are multiple parameters in the model, however, you need to supply values for these nuisance parameters during the test inversion as well.

In the following, suppose that $\boldsymbol{\theta}$ is the covariance parameter vector and that one of its elements, θ, is the parameter of interest for which you want to construct a confidence interval. The other elements of $\boldsymbol{\theta}$ are collected in the nuisance parameter vector $\boldsymbol{\theta}_2$. Suppose that $\widehat{\boldsymbol{\theta}}$ is the estimate of $\boldsymbol{\theta}$ from the overall optimization and that $L(\widehat{\boldsymbol{\theta}})$ is the likelihood evaluated at that estimate. If estimation is based on pseudo-data, then $L(\widehat{\boldsymbol{\theta}})$ is the pseudo-likelihood based on the final pseudo-data. If estimation uses a residual (restricted) likelihood, then L denotes the restricted maximum likelihood and $\widehat{\boldsymbol{\theta}}$ is the REML estimate.

Profile Likelihood Bounds

The likelihood ratio test statistic for testing $H: \theta = \theta_0$ is

$$2\left\{\log\left\{L(\widehat{\boldsymbol{\theta}})\right\} - \log\left\{L(\theta_0, \widehat{\boldsymbol{\theta}}_2)\right\}\right\}$$

where $\widehat{\boldsymbol{\theta}}_2$ is the likelihood estimate of $\boldsymbol{\theta}_2$ under the restriction that $\theta = \theta_0$. To invert this test, a function is defined that returns the maximum likelihood for a fixed value of θ by seeking the maximum over the remaining parameters. This function is termed the profile likelihood (Pawitan 2001, Ch. 3.4),

$$\lambda_p = L(\boldsymbol{\theta}_2|\widetilde{\theta}) = \sup_{\boldsymbol{\theta}_2} L(\widetilde{\theta}, \boldsymbol{\theta}_2)$$

In computing λ_p, θ is fixed at $\widetilde{\theta}$ and $\boldsymbol{\theta}_2$ is estimated. In mixed models, this step typically requires a separate, iterative optimization to find the estimate of $\boldsymbol{\theta}_2$ while θ is held fixed. The $(1-\alpha)\times 100\%$ profile likelihood confidence interval for θ is then defined as the set of values for $\widetilde{\theta}$ that satisfy

$$2\left\{\log\left\{L(\widehat{\boldsymbol{\theta}})\right\} - \log\left\{L(\boldsymbol{\theta}_2|\widetilde{\theta})\right\}\right\} \leq \chi^2_{1,(1-\alpha)}$$

The GLIMMIX procedure seeks the values $\widetilde{\theta}_l$ and $\widetilde{\theta}_u$ that mark the endpoints of the set around $\widehat{\theta}$ that satisfy the inequality. The values ($\widetilde{\theta}_l$ and $\widetilde{\theta}_u$) are then called the $(1-\alpha)\times 100\%$ confidence bounds for θ. Note that the GLIMMIX procedure assumes that the confidence region is not disjoint and relies on the convexity of $L(\widehat{\boldsymbol{\theta}})$.

It is not always possible to find values $\widetilde{\theta}_l$ and $\widetilde{\theta}_u$ that satisfy the inequalities. For example, when the parameter space is $(0,\infty)$ and

$$2\left\{\log\left\{L(\widehat{\boldsymbol{\theta}})\right\} - \log\left\{L(\boldsymbol{\theta}_2|0)\right\}\right\} > \chi^2_{1,(1-\alpha)}$$

a lower bound cannot be found at the desired confidence level. The GLIMMIX procedure reports the right-tail probabilities that are achieved by the underlying likelihood ratio statistic separately for lower and upper bounds.

Effect of Scale Parameter

When a scale parameter ϕ is eliminated from the optimization by profiling from the likelihood, some parameters might be expressed as ratios with ϕ in the optimization. This is the case, for example, in variance component models. The profile likelihood confidence bounds are reported on the scale of the parameter in the overall optimization. In case parameters are expressed as ratios with ϕ or functions of ϕ, the column RatioEstimate is added to the "Covariance Parameter Estimates" table. If parameters are expressed as ratios with ϕ and you want confidence bounds for the unscaled parameter, you can prevent profiling of ϕ from the optimization with the NOPROFILE option in the PROC GLIMMIX statement, or choose estimated likelihood confidence bounds with the TYPE=ELR suboption of the CL option in the COVTEST statement. Note that the NOPROFILE option is automatically in effect with METHOD=LAPLACE and METHOD=QUAD.

Estimated Likelihood Bounds

Computing profile likelihood ratio confidence bounds can be computationally expensive, because of the need to repeatedly estimate $\boldsymbol{\theta}_2$ in a constrained optimization. A computationally simpler method to construct confidence bounds from likelihood-based quantities is to use the estimated likelihood (Pawitan 2001, Ch. 10.7) instead of the profile likelihood. An estimated likelihood technique replaces the nuisance parameters in the test inversion with some other estimate. If you choose the TYPE=ELR suboption of the CL option in the COVTEST statement, the GLIMMIX procedure holds the nuisance parameters fixed at the likelihood estimates. The estimated likelihood statistic for inversion is then

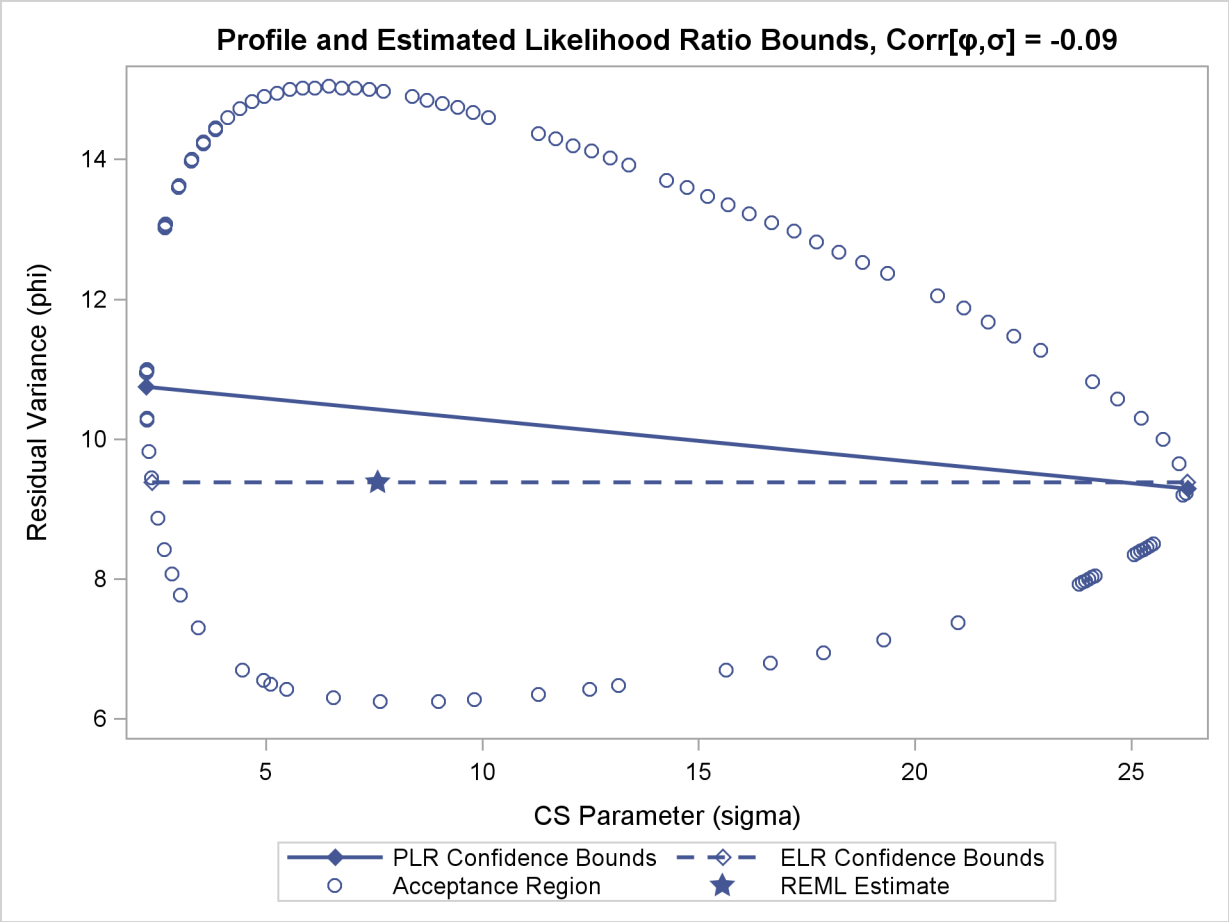
$$\lambda_e = L(\widetilde{\theta}, \widehat{\boldsymbol{\theta}}_2)$$

where $\widehat{\boldsymbol{\theta}}_2$ are the elements of $\widehat{\boldsymbol{\theta}}$ that correspond to the nuisance parameters. As the values of $\widetilde{\theta}$ are varied, no reestimation of $\boldsymbol{\theta}_2$ takes place. Although computationally more economical, estimated

likelihood intervals do not take into account the variability associated with the nuisance parameters. Their coverage can be satisfactory if the parameter of interest is not (or only weakly) correlated with the nuisance parameters. Estimated likelihood ratio intervals can fall short of the nominal coverage otherwise.

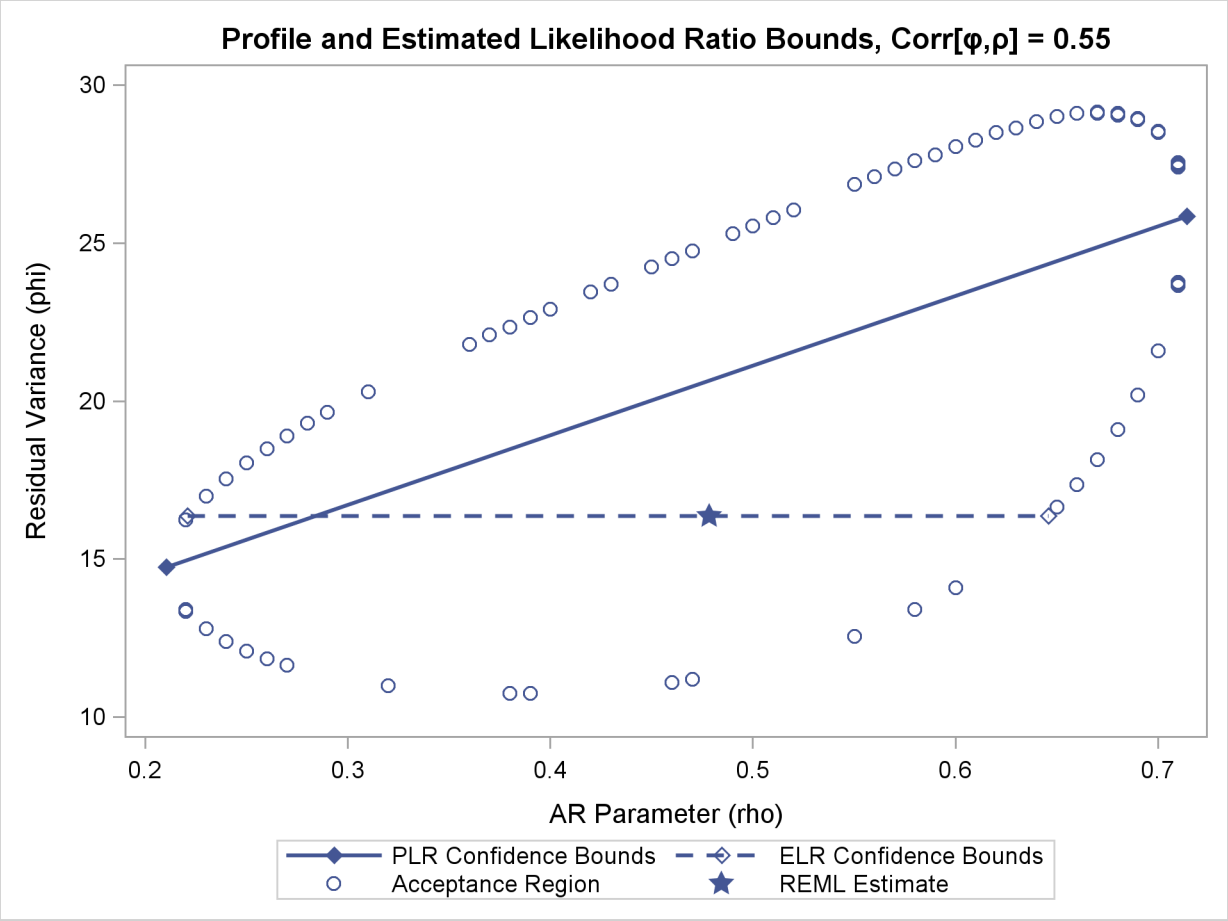
Figure 38.11 depicts profile and estimated likelihood ratio intervals for the parameter σ in a two-parameter compound-symmetric model, $\boldsymbol{\theta} = [\sigma, \phi]'$, in which the correlation between the covariance parameters is small. The elliptical shape traces the set of values for which the likelihood ratio test rejects the hypothesis of equality with the solution. The interior of the ellipse is the "acceptance" region of the test. The solid and dashed lines depict the PLR and ELR confidence limits for σ, respectively. Note that both confidence limits intersect the ellipse and that the ELR interval passes through the REML estimate of ϕ. The PLR bounds are found as those points intersecting the ellipse, where ϕ equals the constrained REML estimate.

Figure 38.11 PLR and ELR Intervals, Small Correlation between Parameters



The major axes of the ellipse in Figure 38.11 are nearly aligned with the major axes of the coordinate system. As a consequence, the line connecting the PLR bounds passes close to the REML estimate in the full model. As a result, ELR bounds will be similar to PLR bounds. Figure 38.12 displays a different scenario, a two-parameter AR(1) covariance structure with a more substantial correlation between the AR(1) parameter (ρ) and the residual variance (ϕ).

Figure 38.12 PLR and ELR Intervals, Large Correlation between Parameters



The correlation between the parameters yields an acceptance region whose major axes are not aligned with the axes of the coordinate system. The ELR bound for ρ passes through the REML estimate of ϕ from the full model and is much shorter than the PLR interval. The PLR interval aligns with the major axis of the acceptance region; it is the preferred confidence interval.

Satterthwaite Degrees of Freedom Approximation

The DDFM=SATTERTHWAITE option in the MODEL statement requests denominator degrees of freedom in t tests and F tests computed according to a general Satterthwaite approximation. The DDFM=KENWARDROGER option also entails the computation of Satterthwaite-type degrees of freedom.

The general Satterthwaite approximation computed in PROC GLIMMIX for the test

$$H: \mathbf{L}\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix} = \mathbf{0}$$

is based on the F statistic

$$F = \frac{\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}' \mathbf{L}'(\mathbf{LCL}')^{-1}\mathbf{L} \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}}{r}$$

where $r = \text{rank}(\mathbf{LCL}')$, and $\mathbf{C}$ is the approximate variance matrix of $[\widehat{\boldsymbol{\beta}}', \widehat{\boldsymbol{\gamma}}' - \boldsymbol{\gamma}']'$; see the section "Estimated Precision of Estimates" on page 2221 and the section "Aspects Common to Adaptive Quadrature and Laplace Approximation" on page 2230.

The approximation proceeds by first performing the spectral decomposition $\mathbf{LCL}' = \mathbf{U}'\mathbf{DU}$, where $\mathbf{U}$ is an orthogonal matrix of eigenvectors and $\mathbf{D}$ is a diagonal matrix of eigenvalues, both of dimension $r \times r$. Define $\mathbf{b}_j$ to be the jth row of $\mathbf{UL}$, and let

$$\nu_j = \frac{2(D_j)^2}{\mathbf{g}_j'\mathbf{A}\mathbf{g}_j}$$

where D_j is the jth diagonal element of $\mathbf{D}$ and $\mathbf{g}_j$ is the gradient of $\mathbf{b}_j\mathbf{C}\mathbf{b}_j'$ with respect to $\boldsymbol{\theta}$, evaluated at $\widehat{\boldsymbol{\theta}}$. The matrix $\mathbf{A}$ is the asymptotic variance-covariance matrix of $\widehat{\boldsymbol{\theta}}$, obtained from the second derivative matrix of the likelihood equations. You can display this matrix with the ASYCOV option in the PROC GLIMMIX statement.

Finally, let

$$E = \sum_{j=1}^{r} \frac{\nu_j}{\nu_j - 2} I(\nu_j > 2)$$

where the indicator function eliminates terms for which $\nu_j \leq 2$. The degrees of freedom for F are then computed as

$$\nu = \frac{2E}{E - \text{rank}(\mathbf{L})}$$

provided $E > r$; otherwise ν is set to zero.

In the one-dimensional case, when PROC GLIMMIX computes a t test, the Satterthwaite degrees of freedom for the t statistic

$$t = \frac{\mathbf{l}' \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}}{\mathbf{l}'\mathbf{Cl}}$$

are computed as

$$\nu = \frac{2(\mathbf{l}'\mathbf{Cl})^2}{\mathbf{g}'\mathbf{Ag}}$$

where $\mathbf{g}$ is the gradient of $\mathbf{l}'\mathbf{Cl}$ with respect to $\boldsymbol{\theta}$, evaluated at $\widehat{\boldsymbol{\theta}}$.

Empirical Covariance ("Sandwich") Estimators

Residual-Based Estimators

The GLIMMIX procedure can compute the classical sandwich estimator of the covariance matrix of the fixed effects, as well as several bias-adjusted estimators. This requires that the model is either an (overdispersed) GLM or a GLMM that can be processed by subjects (see the section "Processing by Subjects" on page 2247).

Consider a statistical model of the form

$$\mathbf{Y} = \boldsymbol{\mu} + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim (\mathbf{0}, \boldsymbol{\Sigma})$$

The general expression of a sandwich covariance estimator is then

$$c \times \widehat{\boldsymbol{\Omega}} \left(\sum_{i=1}^{m} \mathbf{A}_i \widehat{\mathbf{D}}_i' \widehat{\boldsymbol{\Sigma}}_i^{-1} \mathbf{F}_i' \mathbf{e}_i \mathbf{e}_i' \mathbf{F}_i \widehat{\boldsymbol{\Sigma}}_i^{-1} \widehat{\mathbf{D}}_i \mathbf{A}_i \right) \widehat{\boldsymbol{\Omega}}$$

where $\mathbf{e}_i = \mathbf{y}_i - \widehat{\boldsymbol{\mu}}_i$, $\boldsymbol{\Omega} = (\mathbf{D}'\boldsymbol{\Sigma}^{-1}\mathbf{D})^{-}$.

For a GLMM estimated by one of the pseudo-likelihood techniques that involve linearization, you can make the following substitutions: $\mathbf{Y} \rightarrow \mathbf{P}$, $\boldsymbol{\Sigma} \rightarrow \mathbf{V}(\boldsymbol{\theta})$, $\mathbf{D} \rightarrow \mathbf{X}$, $\widehat{\boldsymbol{\mu}} \rightarrow \mathbf{X}\widehat{\boldsymbol{\beta}}$. These matrices are defined in the section "Pseudo-likelihood Estimation Based on Linearization" on page 2220.

The various estimators computed by the GLIMMIX procedure differ in the choice of the constant c and the matrices $\mathbf{F}_i$ and $\mathbf{A}_i$. You obtain the classical estimator, for example, with $c = 1$, and $\mathbf{F}_i = \mathbf{A}_i$ equal to the identity matrix.

The EMPIRICAL=ROOT estimator of Kauermann and Carroll (2001) is based on the approximation

$$\text{Var}\left[\mathbf{e}_i \mathbf{e}_i'\right] \approx (\mathbf{I} - \mathbf{H}_i)\boldsymbol{\Sigma}_i$$

where $\mathbf{H}_i = \mathbf{D}_i \boldsymbol{\Omega} \mathbf{D}_i' \boldsymbol{\Sigma}_i^{-1}$. The EMPIRICAL=FIRORES estimator is based on the approximation

$$\text{Var}\left[\mathbf{e}_i \mathbf{e}_i'\right] \approx (\mathbf{I} - \mathbf{H}_i)\boldsymbol{\Sigma}_i(\mathbf{I} - \mathbf{H}_i')$$

of Mancl and DeRouen (2001). Finally, the EMPIRICAL=FIROEEQ estimator is based on approximating an unbiased estimating equation (Fay and Graubard 2001). For this estimator, $\mathbf{A}_i$ is a diagonal matrix with entries

$$[\mathbf{A}_i]_{jj} = \left(1 - \min\{r, [\mathbf{Q}]_{jj}\}\right)^{-1/2}$$

where $\mathbf{Q} = \mathbf{D}_i' \widehat{\boldsymbol{\Sigma}}_i^{-1} \mathbf{D}_i \widehat{\boldsymbol{\Omega}}$. The optional number $0 \leq r < 1$ is chosen to provide an upper bound on the correction factor. For $r = 0$, the classical sandwich estimator results. PROC GLIMMIX chooses as default value $r = 3/4$. The diagonal entries of $\mathbf{A}_i$ are then no greater than 2.

Table 38.18 summarizes the components of the computation for the GLMM based on linearization, where m denotes the number of subjects and k is the rank of $\mathbf{X}$.

Table 38.18 Empirical Covariance Estimators for a Linearized GLMM

EMPIRICAL=	c	$\mathbf{A}_i$	$\mathbf{F}_i$
CLASSICAL	1	$\mathbf{I}$	$\mathbf{I}$
DF	$\begin{cases} \frac{m}{m-k} & m > k \\ 1 & \text{otherwise} \end{cases}$	$\mathbf{I}$	$\mathbf{I}$
ROOT	1	$\mathbf{I}$	$(\mathbf{I}-\mathbf{H}_i')^{-1/2}$
FIRORES	1	$\mathbf{I}$	$(\mathbf{I}-\mathbf{H}_i')^{-1}$
FIROEEQ(r)	1	$\text{Diag}\{(1-\min\{r,[\mathbf{Q}]_{jj}\})^{-1/2}\}$	$\mathbf{I}$

Computation of an empirical variance estimator requires that the data can be processed by independent sampling units. This is always the case in GLMs. In this case, m equals the sum of all frequencies. In GLMMs, the empirical estimators require that the data consist of multiple subjects. In that case, m equals the number of subjects as per the "Dimensions" table. The following section discusses how the GLIMMIX procedure determines whether the data can be processed by subjects. The section "GLM Mode or GLMM Mode" on page 2233 explains how PROC GLIMMIX determines whether a model is fit in GLM mode or in GLMM mode.

Design-Adjusted MBN Estimator

Morel (1989) and Morel, Bokossa, and Neerchal (2003) suggested a bias correction of the classical sandwich estimator that rests on an additive correction of the residual crossproducts and a sample size correction. This estimator is available with the EMPIRICAL=MBN option in the PROC GLIMMIX statement. In the notation of the previous section, the residual-based MBN estimator can be written as

$$\widehat{\boldsymbol{\Omega}}\left(\sum_{i=1}^{m}\widehat{\mathbf{D}}_i'\widehat{\boldsymbol{\Sigma}}_i^{-1}\left(c\mathbf{e}_i\mathbf{e}_i'+\mathbf{B}_i\right)\widehat{\boldsymbol{\Sigma}}_i^{-1}\widehat{\mathbf{D}}_i\right)\widehat{\boldsymbol{\Omega}}$$

where

- $c=(f-1)/(f-k)\times m/(m-1)$ or $c=1$ when you specify the EMPIRICAL=MBN(NODF) option
- f is the sum of the frequencies
- k equals the rank of $\mathbf{X}$
- $\mathbf{B}_i=\delta_m\phi\widehat{\boldsymbol{\Sigma}}_i$
- $\phi=\max\left\{r,\text{trace}\left(\widehat{\boldsymbol{\Omega}}\mathbf{M}\right)/k^*\right\}$
- $\mathbf{M}=\sum_{i=1}^{m}\widehat{\mathbf{D}}_i'\widehat{\boldsymbol{\Sigma}}_i^{-1}\mathbf{e}_i\mathbf{e}_i'\widehat{\boldsymbol{\Sigma}}_i^{-1}\widehat{\mathbf{D}}_i$
- $k^*=k$ if $m\geq k$, otherwise k^* equals the number of nonzero singular values of $\widehat{\boldsymbol{\Omega}}\mathbf{M}$
- $\delta_m=k/(m-k)$ if $m>(d+1)k$ and $\delta_m=1/d$ otherwise

- $d \geq 1$ and $0 \leq r \leq 1$ are parameters supplied with the *mbn-options* of the EMPIRICAL=MBN(*mbn-options*) option. The default values are $d = 2$ and $r = 1$. When the NODF option is in effect, the factor c is set to 1.

Rearranging terms, the MBN estimator can also be written as an additive adjustment to a sample-size corrected classical sandwich estimator

$$c \times \widehat{\boldsymbol{\Omega}} \left(\sum_{i=1}^{m} \widehat{\mathbf{D}}_i' \widehat{\boldsymbol{\Sigma}}_i^{-1} \mathbf{e}_i \mathbf{e}_i' \widehat{\boldsymbol{\Sigma}}_i^{-1} \widehat{\mathbf{D}}_i \right) \widehat{\boldsymbol{\Omega}} + \delta_m \phi \widehat{\boldsymbol{\Omega}}$$

Because δ_m is of order m^{-1}, the additive adjustment to the classical estimator vanishes as the number of independent sampling units (subjects) increases. The parameter ϕ is a measure of the design effect (Morel, Bokossa, and Neerchal 2003). Besides good statistical properties in terms of Type I error rates in small-m situations, the MBN estimator also has the desirable property of recovering rank when the number of sampling units is small. If $m < k$, the "meat" piece of the classical sandwich estimator is essentially a sum of rank one matrices. A small number of subjects relative to the rank of $\mathbf{X}$ can result in a loss of rank and subsequent loss of numerator degrees of freedom in tests. The additive MBN adjustment counters the rank exhaustion. You can examine the rank of an adjusted covariance matrix with the COVB(DETAILS) option in the MODEL statement.

When the principle of the MBN estimator is applied to the likelihood-based empirical estimator, you obtain

$$\mathbf{H}(\widehat{\boldsymbol{\alpha}})^{-1} \left(\sum_{i=1}^{m} c \mathbf{g}_i(\widehat{\boldsymbol{\alpha}}) \mathbf{g}_i(\widehat{\boldsymbol{\alpha}})' + \mathbf{B}_i \right) \mathbf{H}(\widehat{\boldsymbol{\alpha}})^{-1}$$

where $\mathbf{B}_i = -\delta_m \phi \mathbf{H}_i(\widehat{\boldsymbol{\alpha}})$, and $\mathbf{H}_i(\widehat{\boldsymbol{\alpha}})$ is the second derivative of the log likelihood for the ith sampling unit (subject) evaluated at the vector of parameter estimates, $\widehat{\boldsymbol{\alpha}}$. Also, $\mathbf{g}_i(\widehat{\boldsymbol{\alpha}})$ is the first derivative of the log likelihood for the ith sampling unit. This estimator is computed if you request EMPIRICAL=MBN with METHOD=LAPLACE or METHOD=QUAD.

In terms of adjusting the classical likelihood-based estimator (White 1982), the likelihood MBN estimator can be written as

$$c \times \mathbf{H}(\widehat{\boldsymbol{\alpha}})^{-1} \left(\sum_{i=1}^{m} \mathbf{g}_i(\widehat{\boldsymbol{\alpha}}) \mathbf{g}_i(\widehat{\boldsymbol{\alpha}})' \right) \mathbf{H}(\widehat{\boldsymbol{\alpha}})^{-1} - \delta_m \phi \mathbf{H}(\widehat{\boldsymbol{\alpha}})^{-1}$$

The parameter ϕ is determined as

- $\phi = \max\left\{r, \text{trace}\left(-\mathbf{H}(\widehat{\boldsymbol{\alpha}})^{-1}\mathbf{M}\right)/k^*\right\}$
- $\mathbf{M} = \sum_{i=1}^{m} \mathbf{g}_i(\widehat{\boldsymbol{\alpha}}) \mathbf{g}_i(\widehat{\boldsymbol{\alpha}})'$
- $k^* = k$ if $m \geq k$, otherwise k^* equals the number of nonzero singular values of $-\mathbf{H}(\widehat{\boldsymbol{\alpha}})^{-1}\mathbf{M}$

Exploring and Comparing Covariance Matrices

If you use an empirical (sandwich) estimator with the EMPIRICAL= option in the PROC GLIMMIX statement, the procedure replaces the model-based estimator of the covariance of the fixed effects with the sandwich estimator. This affects aspects of inference, such as prediction standard errors, tests of fixed effects, estimates, contrasts, and so forth. Similarly, if you choose the DDFM=KENWARDROGER degrees-of-freedom method in the MODEL statement, PROC GLIMMIX adjusts the model-based covariance matrix of the fixed effects according to Kenward and Roger (1997) or according to Kackar and Harville (1984) and Harville and Jeske (1992).

In this situation, the COVB(DETAILS) option in the MODEL statement has two effects. The GLIMMIX procedure displays the (adjusted) covariance matrix of the fixed effects and the model-based covariance matrix (for ODS purposes, the name of the table with the model-based covariance matrix is "CovBModelBased"). The procedure also displays a table of statistics for the unadjusted and adjusted covariance matrix and for their comparison. For ODS purposes, the name of this table is "CovBDetails."

If the model-based covariance matrix is not replaced with an adjusted estimator, the COVB(DETAILS) option displays the model-based covariance matrix and provides diagnostic measures for it in the "CovBDetails" table.

The table generated by the COVB(DETAILS) option consists of several sections. See Example 38.8 for an application.

The trace and log determinant of covariance matrices are general scalar summaries that are sometimes used in direct comparisons, or in formulating other statistics, such as the difference of log determinants. The trace simply represents the sum of the variances of all fixed-effects parameters. If a matrix is indefinite, the determinant is reported instead of the log determinant.

The model-based and adjusted covariance matrices should have the same general makeup of eigenvalues. There should not be any negative eigenvalues, and they should have the same numbers of positive and zero eigenvalues. A reduction in rank due to the adjustment is troublesome for aspects of inference. Negative eigenvalues are listed in the table only if they occur, because a covariance matrix should be at least positive semi-definite. However, the GLIMMIX procedure examines the model-based and adjusted covariance matrix for negative eigenvalues. The condition numbers reported by PROC GLIMMIX for positive (semi-)definite matrices are computed as the ratio of the largest and smallest nonzero eigenvalue. A large condition number reflects poor conditioning of the matrix.

Matrix norms are extensions of the concept of vector norms to measure the "length" of a matrix. The Frobenius norm of an $(n \times m)$ matrix $\mathbf{A}$ is the direct equivalent of the Euclidean vector norm, the square root of the sum of the squared elements,

$$||\mathbf{A}||_F = \sqrt{\sum_{i=1}^{n}\sum_{j=1}^{n} a_{ij}^2}$$

The ∞- and 1-norms of matrix $\mathbf{A}$ are the maximum absolute row and column sums, respectively:

$$||\mathbf{A}||_{\infty} = \max\left\{\sum_{j=1}^{m}|a_{ij}| : i = 1,\cdots,n\right\}$$
$$||\mathbf{A}||_{1} = \max\left\{\sum_{i=1}^{n}|a_{ij}| : j = 1,\cdots,m\right\}$$

These two norms are identical for symmetric matrices.

The "Comparison" section of the "CovBDetails" table provides several statistics that set the matrices in relationship. The concordance correlation reported by the GLIMMIX procedure is a standardized measure of the closeness of the model-based and adjusted covariance matrix. It is a slight modification of the covariance concordance correlation in Vonesh, Chinchilli, and Pu (1996) and Vonesh and Chinchilli (1997, Ch. 8.3). Denote as $\boldsymbol{\Omega}$ the $(p \times p)$ model-based covariance matrix and as $\boldsymbol{\Omega}_a$ the adjusted matrix. Suppose that $\mathbf{K}$ is the matrix obtained from the identity matrix of size p by replacing diagonal elements corresponding to singular rows in $\boldsymbol{\Omega}$ with zeros. The lower triangular portion of $\boldsymbol{\Omega}^{-1/2}\boldsymbol{\Omega}_a\boldsymbol{\Omega}^{-1/2}$ is stored in vector $\boldsymbol{\omega}$ and the lower triangular portion of $\mathbf{K}$ is stored in vector $\mathbf{k}$. The matrix $\boldsymbol{\Omega}^{-1/2}$ is constructed from an eigenanalysis of $\boldsymbol{\Omega}$ and is symmetric. The covariance concordance correlation is then

$$r(\boldsymbol{\omega}) = 1 - \frac{||\boldsymbol{\omega} - \mathbf{k}||^2}{||\boldsymbol{\omega}||^2 + ||\mathbf{k}||^2}$$

This measure is 1 if $\boldsymbol{\Omega} = \boldsymbol{\Omega}_a$. If $\boldsymbol{\omega}$ is orthogonal to $\mathbf{k}$, there is total disagreement between the model-based and the adjusted covariance matrix and $r(\boldsymbol{\omega})$ is zero.

The discrepancy function reported by PROC GLIMMIX is computed as

$$d = \log\{|\boldsymbol{\Omega}|\} - \log\{|\boldsymbol{\Omega}_a|\} + \text{trace}\{\boldsymbol{\Omega}_a\boldsymbol{\Omega}^{-}\} - \text{rank}\{\boldsymbol{\Omega}\}$$

In diagnosing departures between an assumed covariance structure and Var[$\mathbf{Y}$]—using an empirical estimator—Vonesh, Chinchilli, and Pu (1996) find that the concordance correlation is useful in detecting gross departures and propose $\lambda = n_s d$ to test the correctness of the assumed model, where n_s denotes the number of subjects.

Processing by Subjects

Some mixed models can be expressed in different but mathematically equivalent ways with PROC GLIMMIX statements. While equivalent statements lead to equivalent statistical models, the data processing and estimation phase can be quite different, depending on how you write the GLIMMIX statements. For example, the particular use of the SUBJECT= option in the RANDOM statement affects data processing and estimation. Certain options are available only when the data are processed by subject, such as the EMPIRICAL option in the PROC GLIMMIX statement.

Consider a GLIMMIX model where variables A and Rep are classification variables with a and r levels, respectively. The following pairs of statements produce the same random-effects structure:

```
class Rep A;
random Rep*A;
```

```
class Rep A;
random intercept / subject=Rep*A;
```

```
class Rep A;
random Rep / subject=A;
```

```
class Rep A;
random A / subject=Rep;
```

In the first case, PROC GLIMMIX does not process the data by subjects because no SUBJECT= option was given. The computation of empirical covariance estimators, for example, will not be possible. The marginal variance-covariance matrix has the same block-diagonal structure as for cases 2–4, where each block consists of the observations belonging to a unique combination of Rep and A. More importantly, the dimension of the $\mathbf{Z}$ matrix of this model will be $n \times ra$, and $\mathbf{Z}$ will be sparse. In the second case, the $\mathbf{Z}_i$ matrix for each of the ra subjects is a vector of ones.

If the data can be processed by subjects, the procedure typically executes faster and requires less memory. The differences can be substantial, especially if the number of subjects is large. Recall that fitting of generalized linear mixed models might be doubly iterative. Small gains in efficiency for any one optimization can produce large overall savings.

If you interpret the intercept as "1," then a RANDOM statement with TYPE=VC (the default) and no SUBJECT= option can be converted into a statement with subject by dividing the random effect by the eventual subject effect. However, the presence of the SUBJECT= option does not imply processing by subject. If a RANDOM statement does not have a SUBJECT= effect, processing by subjects is not possible unless the random effect is a pure R-side overdispersion effect. In the following example, the data will not be processed by subjects, because the first RANDOM statement specifies a G-side component and does not use a SUBJECT= option:

```
proc glimmix;
   class A B;
   model y = B;
   random A;
   random B / subject=A;
run;
```

To allow processing by subjects, you can write the equivalent model with the following statements:

```
proc glimmix;
   class A B;
   model y = B;
   random int / subject=A;
   random B   / subject=A;
run;
```

If you denote a variance component effect X with subject effect S as X–(S), then the "calculus of random effects" applied to the first RANDOM statement reads A = Int*A = Int–(A) = A–(Int). For the second statement there are even more equivalent formulations: A*B = A*B*Int = A*B–(Int) = A–(B) = B–(A) = Int–(A*B).

If there are multiple subject effects, processing by subjects is possible if the effects are equal or contained in each other. Note that in the last example the A*B interaction is a random effect. The following statements give an equivalent specification to the previous model:

```
proc glimmix;
   class A B;
   model y = B;
   random int / subject=A;
   random A   / subject=B;
run;
```

Processing by subjects is not possible in this case, because the two subject effects are not syntactically equal or contained in each other. The following statements depict a case where subject effects are syntactically contained:

```
proc glimmix;
   class A B;
   model y = B;
   random int / subject=A;
   random int / subject=A*B;
run;
```

The A main effect is contained in the A*B interaction. The GLIMMIX procedure chooses as the subject effect for processing the effect that is contained in all other subject effects. In this case, the subjects are defined by the levels of A.

You can examine the "Model Information" and "Dimensions" tables to see whether the GLIMMIX procedure processes the data by subjects and which effect is used to define subjects. The "Model Information" table displays whether the marginal variance matrix is diagonal (GLM models), blocked, or not blocked. The "Dimensions" table tells you how many subjects (=blocks) there are.

Finally, nesting and crossing of interaction effects in subject effects are equivalent. The following two RANDOM statements are equivalent:

```
class Rep A;
random intercept / subject=Rep*A;
```

```
class Rep A;
random intercept / subject=Rep(A);
```

Radial Smoothing Based on Mixed Models

The radial smoother implemented with the TYPE=RSMOOTH option in the RANDOM statement is an approximate low-rank thin-plate spline as described in Ruppert, Wand, and Carroll (2003, Chapter 13.4–13.5). The following sections discuss in more detail the mathematical-statistical connection between mixed models and penalized splines and the determination of the number of spline knots and their location as implemented in the GLIMMIX procedure.

From Penalized Splines to Mixed Models

The connection between splines and mixed models arises from the similarity of the penalized spline fitting criterion to the minimization problem that yields the mixed model equations and solutions for $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. This connection is made explicit in the following paragraphs. An important distinction between classical spline fitting and its mixed model smoothing variant, however, lies in the nature of the spline coefficients. Although they address similar minimization criteria, the solutions for the spline coefficients in the GLIMMIX procedure are the solutions of random effects, not fixed effects. Standard errors of predicted values, for example, account for this source of variation.

Consider the linearized mixed pseudo-model from the section "The Pseudo-model" on page 2220, $\mathbf{P} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$. One derivation of the mixed model equations, whose solutions are $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\gamma}}$, is to maximize the joint density of $f(\boldsymbol{\gamma}, \boldsymbol{\epsilon})$ with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$. This is not a true likelihood problem, because $\boldsymbol{\gamma}$ is not a parameter, but a random vector.

In the special case with $\text{Var}[\boldsymbol{\epsilon}] = \phi\mathbf{I}$ and $\text{Var}[\boldsymbol{\gamma}] = \sigma^2\mathbf{I}$, the maximization of $f(\boldsymbol{\gamma}, \boldsymbol{\epsilon})$ is equivalent to the minimization of

$$Q(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \phi^{-1}(\mathbf{p} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\boldsymbol{\gamma})'(\mathbf{p} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\boldsymbol{\gamma}) + \sigma^{-2}\boldsymbol{\gamma}'\boldsymbol{\gamma}$$

Now consider a linear spline as in Ruppert, Wand, and Carroll (2003, p. 108),

$$p_i = \beta_0 + \beta_1 x_i + \sum_{j=1}^{K} \gamma_j (x_i - t_j)_+$$

where the γ_j denote the spline coefficients at knots $t_1, \cdots, t_K$. The truncated line function is defined as

$$(x - t)_+ = \begin{cases} x - t & x > t \\ 0 & \text{otherwise} \end{cases}$$

If you collect the intercept and regressor x into the matrix $\mathbf{X}$, and if you collect the truncated line functions into the $(n \times K)$ matrix $\mathbf{Z}$, then fitting the linear spline amounts to minimization of the penalized spline criterion

$$Q^*(\boldsymbol{\beta}, \boldsymbol{\gamma}) = (\mathbf{p} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\boldsymbol{\gamma})'(\mathbf{p} - \mathbf{X}\boldsymbol{\beta} - \mathbf{Z}\boldsymbol{\gamma}) + \lambda^2\boldsymbol{\gamma}'\boldsymbol{\gamma}$$

where λ is the smoothing parameter.

Because minimizing $Q^*(\boldsymbol{\beta}, \boldsymbol{\gamma})$ with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ is equivalent to minimizing $Q^*(\boldsymbol{\beta}, \boldsymbol{\gamma})/\phi$, both problems lead to the same solution, and $\lambda = \phi/\sigma$ is the smoothing parameter. The mixed

model formulation of spline smoothing has the advantage that the smoothing parameter is selected "automatically." It is a function of the covariance parameter estimates, which, in turn, are estimated according to the method you specify with the METHOD= option in the PROC GLIMMIX statement.

To accommodate nonnormal responses and general link functions, the GLIMMIX procedure uses $\text{Var}[\boldsymbol{\epsilon}] = \phi\widetilde{\boldsymbol{\Delta}}^{-1}\mathbf{A}\widetilde{\boldsymbol{\Delta}}^{-1}$, where $\mathbf{A}$ is the matrix of variance functions and $\boldsymbol{\Delta}$ is the diagonal matrix of mean derivatives defined earlier. The correspondence between spline smoothing and mixed modeling is then one between a weighted linear mixed model and a weighted spline. In other words, the minimization criterion that yields the estimates $\widehat{\boldsymbol{\beta}}$ and solutions $\widehat{\boldsymbol{\gamma}}$ is then

$$Q(\boldsymbol{\beta},\boldsymbol{\gamma}) = \phi^{-1}(\mathbf{p}-\mathbf{X}\boldsymbol{\beta}-\mathbf{Z}\boldsymbol{\gamma})'\widetilde{\boldsymbol{\Delta}}\mathbf{A}^{-1}\widetilde{\boldsymbol{\Delta}}(\mathbf{p}-\mathbf{X}\boldsymbol{\beta}-\mathbf{Z}\boldsymbol{\gamma})' + \sigma^{-2}\boldsymbol{\gamma}'\boldsymbol{\gamma}$$

If you choose the TYPE=RSMOOTH covariance structure, PROC GLIMMIX chooses radial basis functions as the spline basis and transforms them to approximate a thin-plate spline as in Chapter 13.4 of Ruppert, Wand, and Carroll (2003). For computational expediency, the number of knots is chosen to be less than the number of data points. Ruppert, Wand, and Carroll (2003) recommend one knot per every four unique regressor values for one-dimensional smoothers. In the multivariate case, general recommendations are more difficult, because the optimal number and placement of knots depend on the spatial configuration of samples. Their recommendation for a bivariate smoother is one knot per four samples, but at least 20 and no more than 150 knots (Ruppert, Wand, and Carroll 2003, p. 257).

The magnitude of the variance component σ^2 depends on the metric of the random effects. For example, if you apply radial smoothing in time, the variance changes if you measure time in days or minutes. If the solution for the variance component is near zero, then a rescaling of the random effect data can help the optimization problem by moving the solution for the variance component away from the boundary of the parameter space.

Knot Selection

The GLIMMIX procedure computes knots for low-rank smoothing based on the vertices or centroids of a *k-d* tree. The default is to use the vertices of the tree as the knot locations, if you use the TYPE=RSMOOTH covariance structure. The construction of this tree amounts to a partitioning of the random regressor space until all partitions contain at most b observations. The number b is called the *bucket size* of the *k-d* tree. You can exercise control over the construction of the tree by changing the bucket size with the BUCKET= suboption of the KNOTMETHOD=KDTREE option in the RANDOM statement. A large bucket size leads to fewer knots, but it is not correct to assume that K, the number of knots, is simply $\lfloor n/b \rfloor$. The number of vertices depends on the configuration of the values in the regressor space. Also, coordinates of the bounding hypercube are vertices of the tree. In the one-dimensional case, for example, the extreme values of the random effect are vertices.

To demonstrate how the *k-d* tree partitions the random-effects space based on observed data and the influence of the bucket size, consider the following example from Chapter 50, "The LOESS Procedure." The SAS data set Gas contains the results of an engine exhaust emission study (Brinkman 1981). The covariate in this analysis, E, is a measure of the air-fuel mixture richness. The response, NOx, measures the nitric oxide concentration (in micrograms per joule, and normalized).

```
data Gas;
   input NOx E;
   format NOx E f5.3;
   datalines;
 4.818  0.831
 2.849  1.045
 3.275  1.021
 4.691  0.97
 4.255  0.825
 5.064  0.891
 2.118  0.71
 4.602  0.801
 2.286  1.074
 0.97   1.148
 3.965  1
 5.344  0.928
 3.834  0.767
 1.99   0.701
 5.199  0.807
 5.283  0.902
 3.752  0.997
 0.537  1.224
 1.64   1.089
 5.055  0.973
 4.937  0.98
 1.561  0.665
;
```

There are 22 observations in the data set, and the values of the covariate are unique. If you want to smooth these data with a low-rank radial smoother, you need to choose the number of knots, as well as their placement within the support of the variable E. The *k-d* tree construction depends on the observed values of the variable E; it is independent of the values of nitric oxide in the data. The following statements construct a tree based on a bucket size of $b = 11$ and display information about the tree and the selected knots:

```
ods select KDtree KnotInfo;
proc glimmix data=gas nofit;
  model NOx = e;
  random e  / type=rsmooth
              knotmethod=kdtree(bucket=11 treeinfo knotinfo);
run;
```

The NOFIT option prevents the GLIMMIX procedure from fitting the model. This option is useful if you want to investigate the knot construction for various bucket sizes. The TREEINFO and KNOTINFO suboptions of the KNOTMETHOD=KDTREE option request displays of the *k-d* tree and the knot coordinates derived from it. Construction of the tree commences by splitting the data in half. For $b = 11$, $n = 22$, neither of the two splits contains more than b observations and the process stops. With a single split value, and the two extreme values, the tree has two terminal nodes and leads to three knots (Figure 38.13). Note that for one-dimensional problems, vertices of the *k-d* tree always coincide with data values.

Figure 38.13 *K-d* Tree and Knots for Bucket Size 11

The GLIMMIX Procedure

kd-Tree for RSmooth(E)

Node Number	Left Child	Right Child	Split Direction	Split Value
0	1	2	E	0.9280
1			TERMINAL	
2			TERMINAL	

Radial Smoother Knots for RSmooth(E)

Knot Number	E
1	0.6650
2	0.9280
3	1.2240

If the bucket size is reduced to $b = 8$, the following statements produce the tree and knots in Figure 38.14:

```
ods select KDtree KnotInfo;
proc glimmix data=gas nofit;
  model NOx = e;
  random e / type=rsmooth
             knotmethod=kdtree(bucket=8 treeinfo knotinfo);
run;
```

The initial split value of 0.9280 leads to two sets of 11 observations. In order to achieve a partition into cells that contain at most eight observations, each initial partition is split at its median one more time. Note that one split value is greater and one split value is less than 0.9280.

Figure 38.14 *K-d* Tree and Knots for Bucket Size 8

The GLIMMIX Procedure

kd-Tree for RSmooth(E)

Node Number	Left Child	Right Child	Split Direction	Split Value
0	1	2	E	0.9280
1	3	4	E	0.8070
2	5	6	E	1.0210
3			TERMINAL	
4			TERMINAL	
5			TERMINAL	
6			TERMINAL	

Figure 38.14 *continued*

Radial Smoother Knots for RSmooth(E)

Knot Number	E
1	0.6650
2	0.8070
3	0.9280
4	1.0210
5	1.2240

A further reduction in bucket size to $b = 4$ leads to the tree and knot information shown in Figure 38.15.

Figure 38.15 *K-d* Tree and Knots for Bucket Size 4

The GLIMMIX Procedure

kd-Tree for RSmooth(E)

Node Number	Left Child	Right Child	Split Direction	Split Value
0	1	2	E	0.9280
1	3	4	E	0.8070
2	9	10	E	1.0210
3	5	6	E	0.7100
4	7	8	E	0.8910
5			TERMINAL	
6			TERMINAL	
7			TERMINAL	
8			TERMINAL	
9	11	12	E	0.9800
10	13	14	E	1.0890
11			TERMINAL	
12			TERMINAL	
13			TERMINAL	
14			TERMINAL	

Figure 38.15 *continued*

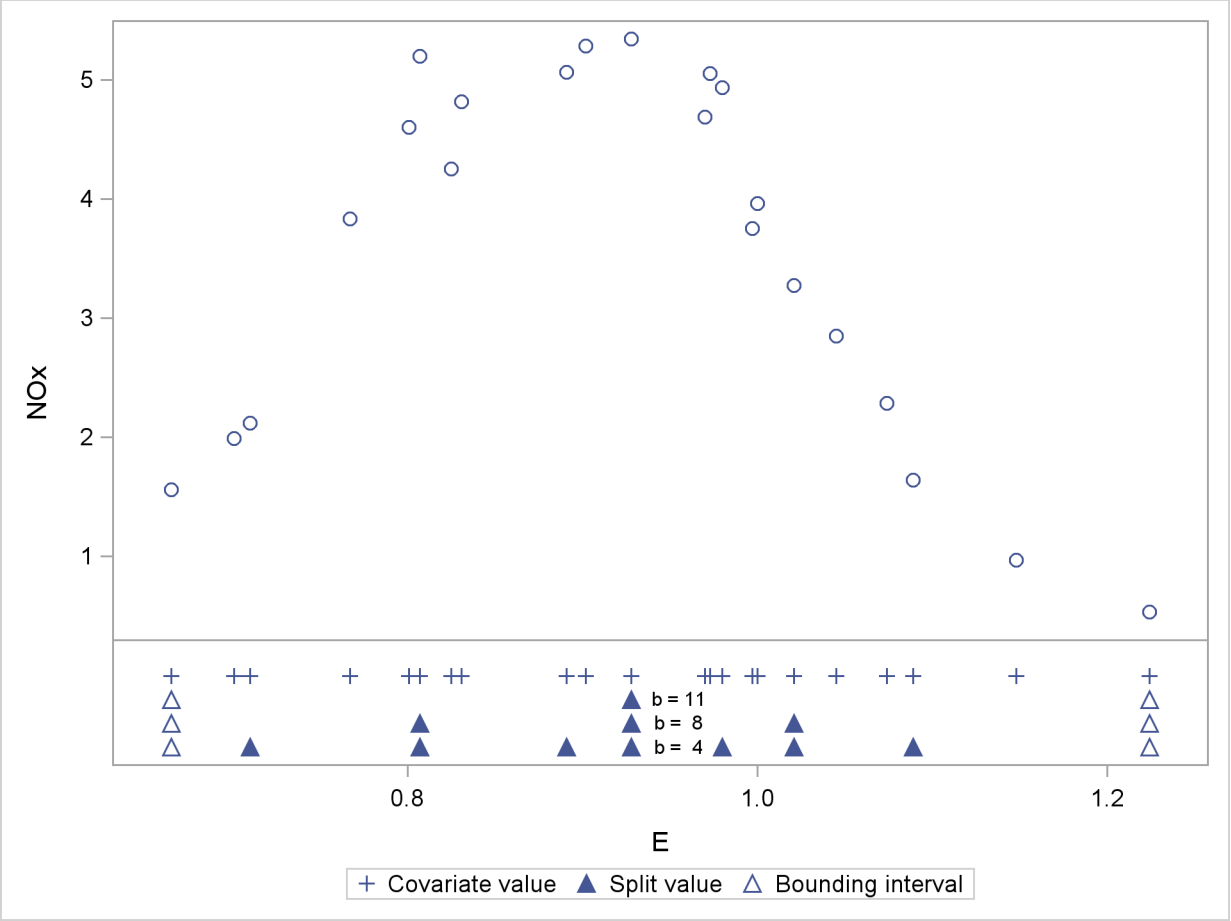
Radial Smoother
Knots for
RSmooth(E)

Knot Number	E
1	0.6650
2	0.7100
3	0.8070
4	0.8910
5	0.9280
6	0.9800
7	1.0210
8	1.0890
9	1.2240

The split value for $b = 11$ is also a split value for $b = 8$, the split values for $b = 8$ are a subset of those for $b = 4$, and so forth. Figure 38.16 displays the data and the location of split values for the three cases. For a one-dimensional problem (a univariate smoother), the vertices comprise the split values and the values on the bounding interval.

You might want to move away from the boundary, in particular for an irregular data configuration or for multivariate smoothing. The KNOTTYPE=CENTER suboption of the KNOTMETHOD= option chooses centroids of the leaf node cells instead of vertices. This tends to move the outer knot locations closer to the convex hull, but not necessarily to data locations. In the emission example, choosing a bucket size of $b = 11$ and centroids as knot locations yields two knots at E=0.7956 and E=1.076. If you choose the NEAREST suboption, then the nearest neighbor of a vertex or centroid will serve as the knot location. In this case, the knot locations are a subset of the data locations, regardless of the dimension of the smooth.

Figure 38.16 Vertices of *k-d* Trees for Various Bucket Sizes



Odds and Odds Ratio Estimation

In models with a logit, generalized logit, or cumulative logit link, you can obtain estimates of odds ratios through the ODDSRATIO options in the PROC GLIMMIX, LSMEANS, and MODEL statements. This section provides details about the computation and interpretation of the computed quantities. Note that for these link functions the EXP option in the ESTIMATE and LSMESTIMATE statements also produces odds or odds ratios.

Consider first a model with a dichotomous outcome variable, linear predictor $\eta = \mathbf{x}'\boldsymbol{\beta} + \mathbf{z}'\boldsymbol{\gamma}$, and logit link function. Suppose that η_0 represents the linear predictor for a condition of interest. For example, in a simple logistic regression model with $\eta = \alpha + \beta x$, η_0 might correspond to the linear predictor at a particular value of the covariate—say, $\eta_0 = \alpha + \beta x_0$.

The modeled probability is $\pi = 1/(1 + \exp\{-\eta\})$, and the odds for $\eta = \eta_0$ are

$$\frac{\pi_0}{1-\pi_0} = \frac{1/(1+\exp\{-\eta_0\})}{\exp\{-\eta_0\}/(1+\exp\{-\eta_0\})} = \exp\{\eta_0\}$$

Because η_0 is a logit, it represents the log odds. The odds ratio $\psi(\eta_1, \eta_0)$ is defined as the ratio of

odds for η_1 and η_0,

$$\psi(\eta_1, \eta_0) = \exp\{\eta_1 - \eta_0\}$$

The odds ratio compares the odds of the outcome under the condition expressed by η_1 to the odds under the condition expressed by η_0. In the preceding simple logistic regression example, this ratio equals $\exp\{\beta(x_1 - x_0)\}$. The exponentiation of the estimate of β is thus an estimate of the odds ratio comparing conditions for which $x_1 - x_0 = 1$. If x and $x + 1$ represent standard and experimental conditions, for example, $\exp\{\beta\}$ compares the odds of the outcome under the experimental condition to the odds under the standard condition. For many other types of models, odds ratios can be expressed as simple functions of parameter estimates. For example, suppose you are fitting a logistic model with a single classification effect with three levels:

```
proc glimmix;
   class A;
   model y = A / dist=binary;
run;
```

The estimated linear predictor for level j of A is $\widehat{\eta}_j = \widehat{\beta} + \widehat{\alpha}_j$, $j = 1, 2, 3$. Because the $\mathbf{X}$ matrix is singular in this model due to the presence of an overall intercept, the solution for the intercept estimates $\beta + \alpha_3$, and the solution for the jth treatment effect estimates $\alpha_j - \alpha_3$. Exponentiating the solutions for α_1 and α_2 thus produces odds ratios comparing the odds for these levels against the third level of A.

Results designated as odds or odds ratios in the GLIMMIX procedure might reduce to simple exponentiations of solutions in the "Parameter Estimates" table, but they are computed by a different mechanism if the model contains classification variables. The computations rely on general estimable functions; for the MODEL, LSMEANS, and LSMESTIMATE statements, these functions are based on least squares means. This enables you to obtain odds ratio estimates in more complicated models that involve main effects and interactions, including interactions between continuous and classification variables.

In all cases, the results represent the exponentiation of a linear function of the fixed-effects parameters, $\eta = \mathbf{l}'\boldsymbol{\beta}$. If L_η and U_η are the confidence limits for η on the logit scale, confidence limits for the odds or the odds ratio are obtained as $\exp\{L_\eta\}$ and $\exp\{U_\eta\}$.

The Odds Ratio Estimates Table

This table is produced by the ODDSRATIO option in the MODEL statement. It consists of estimates of odds ratios and their confidence limits. Odds ratios are produced for the following:

- classification main effects, if they appear in the MODEL statement
- continuous variables in the MODEL statement, unless they appear in an interaction with a classification effect
- continuous variables in the MODEL statement at fixed levels of a classification effect, if the MODEL statement contains an interaction of the two.
- continuous variables in the MODEL statements if they interact with other continuous variables

The Default Table

Consider the following PROC GLIMMIX statements that fit a logistic model with one classification effect, one continuous variable, and their interaction (the ODDSRATIO option in the MODEL statement requests the "Odds Ratio Estimates" table).

```
proc glimmix;
   class A;
   model y = A x A*x / dist=binary oddsratio;
run;
```

By default, odds ratios are computed as follows:

- The covariate is set to its average, $\overline{x}$, and the least squares means for the A effect are obtained. Suppose $\mathbf{L}^{(1)}$ denotes the matrix of coefficients defining the estimable functions that produce the a least squares means $\mathbf{L}\widehat{\boldsymbol{\beta}}$, and $\mathbf{l}_j^{(1)}$ denotes the jth row of $\mathbf{L}^{(1)}$. Differences of the least squares means against the last level of the A factor are computed and exponentiated:

$$\begin{aligned}
\psi(A_1, A_a) &= \exp\left\{\left(\mathbf{l}_1^{(1)} - \mathbf{l}_a^{(1)}\right)\widehat{\boldsymbol{\beta}}\right\} \\
\psi(A_2, A_a) &= \exp\left\{\left(\mathbf{l}_2^{(1)} - \mathbf{l}_a^{(1)}\right)\widehat{\boldsymbol{\beta}}\right\} \\
&\vdots \\
\psi(A_{a-1}, A_a) &= \exp\left\{\left(\mathbf{l}_{a-1}^{(1)} - \mathbf{l}_a^{(1)}\right)\widehat{\boldsymbol{\beta}}\right\}
\end{aligned}$$

  The differences are checked for estimability. Notice that this set of odds ratios can also be obtained with the following LSMESTIMATE statement (assuming A has five levels):

```
lsmestimate A 1  0  0  0 -1,
              0  1  0  0 -1,
              0  0  1  0 -1,
              0  0  0  1 -1 / exp cl;
```

  You can also obtain the odds ratios with this LSMEANS statement (assuming the last level of A is coded as 5):

```
lsmeans A / diff=control('5') oddsratio cl;
```

- The odds ratios for the covariate must take into account that x occurs in an interaction with the A effect. A second set of least squares means are computed, where x is set to $\overline{x} + 1$. Denote the coefficients of the estimable functions for this set of least squares means as $\mathbf{L}^{(2)}$. Differences of the least squares means at a given level of factor A are then computed and exponentiated:

$$\begin{aligned}
\psi(A(\overline{x}+1)_1, A(\overline{x})_1) &= \exp\left\{\left(\mathbf{l}_1^{(2)} - \mathbf{l}_1^{(1)}\right)\widehat{\boldsymbol{\beta}}\right\} \\
\psi(A(\overline{x}+1)_2, A(\overline{x})_2) &= \exp\left\{\left(\mathbf{l}_2^{(2)} - \mathbf{l}_2^{(1)}\right)\widehat{\boldsymbol{\beta}}\right\} \\
&\vdots \\
\psi(A(\overline{x}+1)_a, A(\overline{x})_a) &= \exp\left\{\left(\mathbf{l}_a^{(2)} - \mathbf{l}_a^{(1)}\right)\widehat{\boldsymbol{\beta}}\right\}
\end{aligned}$$

The differences are checked for estimability. If the continuous covariate does not appear in an interaction with the A variable, only a single odds ratio estimate related to x would be produced, relating the odds of a one-unit shift in the regressor from $\overline{x}$.

Suppose you fit a model that contains interactions of continuous variables, as with the following statements:

```
proc glimmix;
   class A;
   model y = A x x*z / dist=binary oddsratio;
run;
```

In the computation of the A least squares means, the continuous effects are set to their means—that is, $\overline{x}$ and $\overline{xz}$. In the computation of odds ratios for x, linear predictors are computed at x = $\overline{x}$, x*z = $\overline{x} \times \overline{z}$ and at x = $\overline{x} + 1$, x*z = $(\overline{x} + 1)\overline{z}$.

Modifying the Default Table, Customized Odds Ratios

Several suboptions of the ODDSRATIO option in the MODEL statement are available to obtain customized odds ratio estimates. For customized odds ratios that cannot be obtained with these suboptions, use the EXP option in the ESTIMATE or LSMESTIMATE statement.

The type of differences constructed when the levels of a classification factor are varied is controlled by the DIFF= suboption. By default, differences against the last level are taken. DIFF=FIRST computes differences from the first level, and DIFF=ALL computes odds ratios based on all pairwise differences.

For continuous variables in the model, you can change both the reference value (with the AT suboption) and the units of change (with the UNIT suboption). By default, a one-unit change from the mean of the covariate is assessed. For example, the following statements produce all pairwise differences for the A factor:

```
proc glimmix;
   class A;
   model y = A x A*x / dist=binary
                       oddsratio(diff=all
                                 at   x=4
                                 unit x=3);
run;
```

The covariate x is set to the reference value $x = 4$ in the computation of the least squares means for the A odds ratio estimates. The odds ratios computed for the covariate are based on differencing this set of least squares means with a set of least squares means computed at $x = 4 + 3$.

Odds or Odds Ratio

The odds ratio is the exponentiation of a difference on the logit scale,

$$\psi(\eta_1, \eta_0) = \exp\{(\mathbf{l}_1 - \mathbf{l}_0)\boldsymbol{\beta}\}$$

and $\exp\{\mathbf{l}_1\boldsymbol{\beta}\}$ and $\exp\{\mathbf{l}_0\boldsymbol{\beta}\}$ are the corresponding odds. If the ODDSRATIO option is specified in a suitable model in the PROC GLIMMIX statement or the individual statements that support the option, odds ratios are computed in the "Odds Ratio Estimates" table (MODEL statement), the "Differences of Least Squares Means" table (LSMEANS / DIFF), and the "Simple Effect Comparisons of Least Squares Means" table (LSMEANS / SLICEDIFF=). Odds are computed in the "Least Squares Means" table.

Odds Ratios in Multinomial Models

The GLIMMIX procedure fits two kinds of models to multinomial data. Models with cumulative link functions apply to ordinal data, and generalized logit models are fit to nominal data. If you model a multinomial response with LINK=CUMLOGIT or LINK=GLOGIT, odds ratio results are available for these models.

In the generalized logit model, you model baseline category logits. By default, the GLIMMIX procedure chooses the last category as the reference category. If your nominal response has J categories, the baseline logit for category j is

$$\log\left\{\pi_j/\pi_J\right\} = \eta_j = \mathbf{x}'\boldsymbol{\beta}_j + \mathbf{z}'\mathbf{u}_j$$

and

$$\pi_j = \frac{\exp\{\eta_j\}}{\sum_{k=1}^{J}\exp\{\eta_k\}}$$
$$\eta_J = 0$$

As before, suppose that the two conditions to be compared are identified with subscripts 1 and 0. The log odds ratio of outcome j versus J for the two conditions is then

$$\log\left\{\psi\left(\eta_{j1},\eta_{j0}\right)\right\} = \log\left\{\frac{\pi_{j1}/\pi_{J1}}{\pi_{j0}/\pi_{J0}}\right\} = \log\left\{\frac{\exp\{\eta_{j1}\}}{\exp\{\eta_{j0}\}}\right\}$$
$$= \eta_{j1} - \eta_{j0}$$

Note that the log odds ratios are again differences on the scale of the linear predictor, but they depend on the response category. The GLIMMIX procedure determines the estimable functions whose differences represent log odds ratios as discussed previously but produces separate estimates for each nonreference response category.

In models for ordinal data, PROC GLIMMIX models the logits of cumulative probabilities. Thus, the estimates on the linear scale represent log cumulative odds. The cumulative logits are formed as

$$\log\left\{\frac{\Pr(Y \le j)}{\Pr(Y > j)}\right\} = \eta_j = \alpha_j + \mathbf{x}'\boldsymbol{\beta} + \mathbf{z}'\boldsymbol{\gamma} = \alpha_j + \tilde{\eta}$$

so that the linear predictor depends on the response category only through the intercepts (cutoffs) $\alpha_1,\cdots,\alpha_{J-1}$. The odds ratio comparing two conditions represented by linear predictors η_{j1} and η_{j0} is then

$$\psi\left(\eta_{j1},\eta_{j0}\right) = \exp\left\{\eta_{j1} - \eta_{j0}\right\}$$
$$= \exp\left\{\tilde{\eta}_1 - \tilde{\eta}_0\right\}$$

and is independent of category.

Parameterization of Generalized Linear Mixed Models

PROC GLIMMIX constructs a generalized linear mixed model according to the specifications in the CLASS, MODEL, and RANDOM statements. Each effect in the MODEL statement generates one or more columns in the matrix **X**, and each G-side effect in the RANDOM statement generates one or more columns in the matrix **Z**. R-side effects in the RANDOM statement do not generate model matrices; they serve only to index observations within subjects. This section shows how the GLIMMIX procedure builds **X** and **Z**. You can output the **X** and **Z** matrices to a SAS data set with the OUTDESIGN= option in the PROC GLIMMIX statement.

The general rules and techniques for parameterization of a linear model are given in "GLM Parameterization of Classification Variables and Effects" on page 369 of Chapter 18, "Shared Concepts and Topics." The following paragraphs discuss how these rules differ in a mixed model, in particular, how parameterization differs between the **X** and the **Z** matrix.

Intercept

By default, all models automatically include a column of 1s in **X** to estimate a fixed-effect intercept parameter. You can use the NOINT option in the MODEL statement to suppress this intercept. The NOINT option is useful when you are specifying a classification effect in the MODEL statement and you want the parameter estimates to be in terms of the (linked) mean response for each level of that effect, rather than in terms of a deviation from an overall mean.

By contrast, the intercept is not included by default in **Z**. To obtain a column of 1s in **Z**, you must specify in the RANDOM statement either the INTERCEPT effect or some effect that has only one level.

Interaction Effects

Often a model includes interaction (crossed) effects. With an interaction, PROC GLIMMIX first reorders the terms to correspond to the order of the variables in the CLASS statement. Thus, `B*A` becomes `A*B` if `A` precedes `B` in the CLASS statement. Then, PROC GLIMMIX generates columns for all combinations of levels that occur in the data. The order of the columns is such that the rightmost variables in the cross index faster than the leftmost variables. Empty columns (which would contain all 0s) are not generated for **X**, but they are for **Z**.

See Table 18.5 in the section "GLM Parameterization of Classification Variables and Effects" on page 369 of Chapter 18, "Shared Concepts and Topics," for an example of an interaction parameterization.

Nested Effects

Nested effects are generated in the same manner as crossed effects. Hence, the design columns generated by the following two statements are the same (but the ordering of the columns is different):

Note that nested effects are often distinguished from interaction effects by the implied randomization structure of the design. That is, they usually indicate random effects within a fixed-effects framework. The fact that random effects can be modeled directly in the RANDOM statement might make the specification of nested effects in the MODEL statement unnecessary.

See Table 18.6 in the section "GLM Parameterization of Classification Variables and Effects" on page 369 of Chapter 18, "Shared Concepts and Topics," for an example of the parameterization of a nested effect.

Implications of the Non-Full-Rank Parameterization

For models with fixed effects involving classification variables, there are more design columns in $\mathbf{X}$ constructed than there are degrees of freedom for the effect. Thus, there are linear dependencies among the columns of $\mathbf{X}$. In this event, all of the parameters are not estimable; there is an infinite number of solutions to the mixed model equations. The GLIMMIX procedure uses a generalized inverse (a g_2-inverse, Pringle and Rayner 1971), to obtain values for the estimates (Searle 1971). The solution values are not displayed unless you specify the SOLUTION option in the MODEL statement. The solution has the characteristic that estimates are 0 whenever the design column for that parameter is a linear combination of previous columns. With this parameterization, hypothesis tests are constructed to test linear functions of the parameters that are estimable.

Some procedures (such as the CATMOD and LOGISTIC procedures) reparameterize models to full rank by using restrictions on the parameters. PROC GLM, PROC MIXED, and PROC GLIMMIX do not reparameterize, making the hypotheses that are commonly tested more understandable. See Goodnight (1978a) for additional reasons for not reparameterizing.

Missing Level Combinations

PROC GLIMMIX handles missing level combinations of classification variables in the same manner as PROC GLM and PROC MIXED. These procedures delete fixed-effects parameters corresponding to missing levels in order to preserve estimability. However, PROC GLIMMIX does not delete missing level combinations for random-effects parameters because linear combinations of the random-effects parameters are always predictable. These conventions can affect the way you specify your CONTRAST and ESTIMATE coefficients.

Notes on the EFFECT Statement

Some restrictions and limitations for models that contain constructed effects are in place with the GLIMMIX procedure. Also, you should be aware of some special defaults and handling that apply only when the model contains constructed fixed and/or random effects.

- Constructed effects can be used in the MODEL and RANDOM statements but not to specify SUBJECT= or GROUP= effects.

- Computed variables are not supported in the specification of a constructed effect. All variables needed to form the collection of columns for a constructed effect must be in the data set.

- You cannot use constructed effects that comprise continuous variables or interactions with other constructed effects as the LSMEANS or LSMESTIMATE effect.

- The calculation of quantities that depend on least squares means, such as odds ratios in the "Odds Ratio Estimates" table, is not possible if the model contains fixed effects that consist of more than one constructed effects, unless all constructed effects are of spline type. For example, least squares means computations are not possible in the following model because the MM_AB*cvars effect contains two constructed effects:

```
proc glimmix;
   class A B C;
   effect MM_AB = MM(A B);
   effect cvars = COLLECTION(x1 x2 x3);
   model y = C MM_AB*cvars;
run;
```

- If the MODEL or RANDOM statement contains constructed effects, the default degrees-of-freedom method for mixed models is DDFM=KENWARDROGER. The containment degrees-of-freedom method (DDFM=CONTAIN) is not available in these models.

- If the model contains fixed spline effects, least squares means are computed at the average spline coefficients across the usable data, possibly further averaged over levels of class variables that interact with the spline effects in the model. You can use the AT option in the LSMEANS and LSMESTIMATE statements to construct the splines for particular values of the covariates involved. Consider, for example, the following statements:

```
proc glimmix;
   class A;
   effect spl = spline(x);
   model y = A spl;
   lsmeans A;
   lsmeans A / at means;
   lsmeans A / at x=0.4;
run;
```

  Suppose that the spl effect contributes seven columns $[\mathbf{s}_1, \cdots, \mathbf{s}_7]$ to the $\mathbf{X}$ matrix. The least squares means coefficients for the spl effect in the first LSMEANS statement are $[\overline{s}_1, \cdots, \overline{s}_7]$ with the averages taken across the observations used in the analysis. The second LSMEANS statement computes the spline coefficient at the average value of x: $[s(\overline{x})_1, \cdots, s(\overline{x})_7]$. The final LSMEANS statement uses $[s(0.4)_1, \cdots, s(0.4)_7]$. Using the AT option for least squares means calculations with spline effects can resolve inestimability issues.

- Using a spline effect with B-spline basis in the RANDOM statement is **not** the same as using a penalized B-spline (P-spline) through the TYPE=PSPLINE option in the RANDOM statement. The following statement constructs a penalized B-spline by using mixed model methodology:

```
random x / type=pspline;
```

The next set of statements defines a set of B-spline columns in the **Z** matrix with uncorrelated random effects and homogeneous variance:

```
effect bspline = spline(x);
random bspline / type=vc;
```

This does not lead to a properly penalized fit. See the documentation on TYPE=PSPLINE about the construction of penalties for B-splines through the covariance matrix of random effects.

Positional and Nonpositional Syntax for Contrast Coefficients

When you define custom linear hypotheses with the CONTRAST or ESTIMATE statement, the GLIMMIX procedure sets up an **L** vector or matrix that conforms to the fixed-effects solutions or the fixed- and random-effects solutions. With the LSMESTIMATE statement, you specify coefficients of the matrix **K** that is then converted into a coefficient matrix that conforms to the fixed-effects solutions.

There are two methods for specifying the entries in a coefficient matrix (hereafter simply referred to as the **L** matrix), termed the positional and nonpositional methods. In the positional form, and this is the traditional method, you provide a list of values that occupy the elements of the **L** matrix associated with the effect in question in the order in which the values are listed. For traditional model effects comprising continuous and classification variables, the positional syntax is simpler in some cases (main effects) and more cumbersome in others (interactions). When you work with effects constructed through the experimental EFFECT statement, the nonpositional syntax is essential.

Consider, for example, the following two-way model with interactions where factors A and B have three and two levels, respectively:

```
proc glimmix;
   class a b block;
   model y = a b a*b / ddfm=kr;
   random block a*block;
run;
```

To test the difference of the B levels at the second level of A with a CONTRAST statement (a slice), you need to assign coefficients 1 and -1 to the levels of B and to the levels of the interaction where A is at the second level. Two examples of equivalent CONTRAST statements by using positional and nonpositional syntax are as follows:

```
contrast 'B at A2' b 1 -1 a*b 0  0  1 -1      ;
contrast 'B at A2' b 1 -1 a*b [1 2 1] [-1 2 2];
```

Because A precedes B in the CLASS statement, the levels of the interaction are formed as $\alpha_1\beta_1, \alpha_1\beta_2, \alpha_2\beta_1, \alpha_2\beta_2, \cdots$. If B precedes A in the CLASS statement, you need to modify the coefficients accordingly:

```
proc glimmix;
   class b a block;
   model y = a b a*b / ddfm=kr;
   random block a*block;
   contrast 'B at A2' b 1 -1 a*b 0  1  0  0 -1     ;
   contrast 'B at A2' b 1 -1 a*b [1  1 2] [-1  2 2];
   contrast 'B at A2' b 1 -1 a*b [1, 1 2] [-1, 2 2];
run;
```

You can optionally separate the **L** value entry from the level indicators with a comma, as in the last CONTRAST statement.

The general syntax for defining coefficients with the nonpositional syntax is as follows:

effect-name [*multiplier* <,> *level-values*] . . . <[*multiplier* <,> *level-values*]>

The first entry in square brackets is the multiplier that is applied to the elements of **L** for the effect after the *level-values* have been resolved and any necessary action forming **L** has been taken.

The *level-values* are organized in a specific form:

- The number of entries should equal the number of terms needed to construct the effect. For effects that do not contain any constructed effects, this number is simply the number of terms in the name of the effect.
- Values of continuous variables needed for the construction of the **L** matrix precede the level indicators of CLASS variables.
- If the effect involves constructed effects, then you need to provide as many continuous and classification variables as are needed for the effect formation. For example, if a grouping effect is defined as

  ```
  class c;
  effect v = vars(x1 x2 c);
  ```

  then a proper nonpositional syntax would be, for example,

  ```
  v [0.5,  0.2 0.3 3]
  ```

- If an effect contains both regular terms (old-style effects) and constructed effects, then the order of the coefficients is as follows: continuous values for old-style effects, class levels for CLASS variables in old-style effects, continuous values for constructed effects, and finally class levels needed for constructed effects.

  Assume that C has four levels so that effect v contributes six elements to the **L** matrix. When PROC GLIMMIX resolves this syntax, the values 0.2 and 0.3 are assigned to the positions for x1 and x2 and a 1 is associated with the third level of C. The resulting vector is then multiplied by 0.5 to produce

  [0.1 0.15 0 0 0.5 0]

Note that you enter the **levels** of the classification variables in the square brackets, not their formatted values. The ordering of the levels of CLASS variables can be gleaned from the "Class Level Information" table.

To specify values for continuous variables, simply give their value as one of the terms in the effect. The nonpositional syntax in the following ESTIMATE statement is read as "1-time the value 0.4 in the column associated with level 2 of A"

```
proc glimmix;
   class a;
   model y = a a*x / s;
   lsmeans a / e at x=0.4;
   estimate 'A2 at x=0.4' intercept 1 a 0 1 a*x [1,0.4 2] / e;
run;
```

Because the value before the comma serves as a multiplier, the same estimable function could also be constructed with the following statements:

```
estimate 'A2 at x=0.4' intercept 1 a 0 1 a*x [ 4,  0.1 2];
estimate 'A2 at x=0.4' intercept 1 a 0 1 a*x [ 2,  0.2 2];
estimate 'A2 at x=0.4' intercept 1 a 0 1 a*x [-1, -0.4 2];
```

Note that continuous variables needed to construct an effect are always listed before any CLASS variables.

When you work with constructed effects, the nonpositional syntax works in the same way. For example, the following model contains a classification effect and a B-spline. The first two ESTIMATE statements produce predicted values for level one of C when the continuous variable x takes on the values 20 and 10, respectively.

```
proc glimmix;
   class c;
   effect spl = spline(x / knotmethod=equal(5));
   model y = c spl;
   estimate 'C = 1 @ x=20' intercept 1 c 1 spl [1,20],
            'C = 1 @ x=10' intercept 1 c 1 spl [1,10];
   estimate 'Difference'   spl [1,20] [-1,10];
run;
```

The GLIMMIX procedure computes the spline coefficients for the first ESTIMATE statement based on $x = 20$, and similarly in the second statement for $x = 10$. The third ESTIMATE statement computes the difference of the predicted values. Because the spline effect does not interact with the classification variable, this difference does not depend on the level of C. If such an interaction is present, you can estimate the difference in predicted values for a given level of C by using the nonpositional syntax. Because the effect C*spl contains both old-style terms (C) and a constructed effect, you specify the values for the old-style terms before assigning values to constructed effects:

```
proc glimmix;
   class c;
   effect spl = spline(x / knotmethod=equal(5));
   model y = spl*c;
   estimate 'C2 = 1, x=20' intercept 1 c*spl [1,1 20];
   estimate 'C2 = 2, x=20' intercept 1 c*spl [1,2 20];
   estimate 'C diff at x=20' c*spl [1,1 20] [-1,2 20];
run;
```

It is recommended to add the E option to the CONTRAST, ESTIMATE, or LSMESTIMATE statement to verify that the **L** matrix is formed according to your expectations.

In any row of an ESTIMATE or CONTRAST statement you can choose positional and nonpositional syntax separately for each effect. You cannot mix the two forms of syntax for coefficients of a single effect, however. For example, the following statement is not proper because both forms of syntax are used for the interaction effect:

```
estimate 'A1B1 - A1B2' b 1 -1  a*b 0 1  [-1, 1 2];
```

Response-Level Ordering and Referencing

In models for binary and multinomial data, the response-level ordering is important because it reflects the following:

- which probability is modeled with binary data
- how categories are ordered for ordinal data
- which category serves as the reference category in nominal generalized logit models (models for nominal data)

You should view the "Response Profile" table to ensure that the categories are properly arranged and that the desired outcome is modeled. In this table, response levels are arranged by *Ordered Value*. The lowest response level is assigned Ordered Value 1, the next lowest is assigned Ordered Value 2, and so forth. In binary models, the probability modeled is the probability of the response level with the lowest Ordered Value.

You can change which probability is modeled and the Ordered Value in the "Response Profile" table with the DESCENDING, EVENT=, ORDER=, and REF= response variable options in the MODEL statement. See the section "Response Level Ordering" on page 3318 in Chapter 51, "The LOGISTIC Procedure," for examples about how to use these options to affect the probability being modeled for binary data.

For multinomial models, the response-level ordering affects two important aspects. In cumulative link models the categories are assumed ordered according to their Ordered Value in the "Response Profile" table. If the response variable is a character variable or has a format, you should check this table carefully as to whether the Ordered Values reflect the correct ordinal scale.

In generalized logit models (for multinomial data with unordered categories), one response category is chosen as the reference category in the formulation of the generalized logits. By default, the linear predictor in the reference category is set to 0, and the reference category corresponds to the entry in the "Response Profile" table with the highest Ordered Value. You can affect the assignment of Ordered Values with the DESCENDING and ORDER= options in the MODEL statement. You can choose a different reference category with the REF= option. The choice of the reference category for generalized logit models affects the results. It is sometimes recommended that you choose the category with the highest frequency as the reference (see, for example, Brown and Prescott 1999, p. 160). You can achieve this with the GLIMMIX procedure by combining the ORDER= and REF= options, as in the following statements:

```
proc glimmix;
   class preference;
   model preference(order=freq ref=first) = feature price /
                   dist=multinomial
                   link=glogit;
   random intercept / subject=store group=preference;
run;
```

The ORDER=FREQ option arranges the categories by descending frequency. The REF=FIRST option then selects the response category with the lowest Ordered Value—the most frequent category—as the reference.

Comparing the GLIMMIX and MIXED Procedures

The MIXED procedure is subsumed by the GLIMMIX procedure in the following sense:

- Linear mixed models are a special case in the family of generalized linear mixed models; a linear mixed model is a generalized linear mixed model where the conditional distribution is normal and the link function is the identity function.
- Most models that can be fit with the MIXED procedure can also be fit with the GLIMMIX procedure.

Despite this overlap in functionality, there are also some important differences between the two procedures. Awareness of these differences enables you to select the most appropriate tool in situations where you have a choice between procedures and to identify situations where a choice does not exist. Furthermore, the %GLIMMIX macro, which fits generalized linear mixed models by linearization methods, essentially calls the MIXED procedure repeatedly. If you are aware of the syntax differences between the procedures, you can easily convert your %GLIMMIX macro statements.

Important functional differences between PROC GLIMMIX and PROC MIXED for linear models and linear mixed models include the following:

- The MIXED procedure models R-side effects through the REPEATED statement and G-side effects through the RANDOM statement. The GLIMMIX procedure models all random components of the model through the RANDOM statement. You use the _RESIDUAL_ keyword

or the RESIDUAL option in the RANDOM statement to model R-side covariance structure in the GLIMMIX procedure. For example, the PROC MIXED statement

```
repeated / subject=id type=ar(1);
```

is equivalent to the following RANDOM statement in the GLIMMIX procedure:

```
random _residual_ / subject=id type=ar(1);
```

If you need to specify an effect for levelization—for example, because the construction of the **R** matrix is order-dependent or because you need to account for missing values—the RESIDUAL option in the RANDOM statement of the GLIMMIX procedure is used to indicate that you are modeling an R-side covariance nature. For example, the PROC MIXED statements

```
class time id;
repeated time / subject=id type=ar(1);
```

are equivalent to the following PROC GLIMMIX statements:

```
class time id;
random time / subject=id type=ar(1) residual;
```

- There is generally considerable overlap in the covariance structures available through the TYPE= option in the RANDOM statement in PROC GLIMMIX and through the TYPE= options in the RANDOM and REPEATED statements in PROC MIXED. However, the Kronecker-type structures, the geometrically anisotropic spatial structures, and the GDATA= option in the RANDOM statement of the MIXED procedure are currently not supported in the GLIMMIX procedure. The MIXED procedure, on the other hand, does not support TYPE=RSMOOTH and TYPE=PSPLINE.

- For normal linear mixed models, the (default) METHOD=RSPL in PROC GLIMMIX is identical to the default METHOD=REML in PROC MIXED. Similarly, METHOD=MSPL in PROC GLIMMIX is identical for these models to METHOD=ML in PROC MIXED. The GLIMMIX procedure does not support Type I through Type III (ANOVA) estimation methods for variance component models. Also, the procedure does not have a METHOD=MIVQUE0 option, but you can produce these estimates through the NOITER option in the PARMS statement.

- The MIXED procedure solves the iterative optimization problem by means of a ridge-stabilized Newton-Raphson algorithm. With the GLIMMIX procedure, you can choose from a variety of optimization methods via the NLOPTIONS statement. The default method for most GLMMs is a quasi-Newton algorithm. A ridge-stabilized Newton-Raphson algorithm, akin to the optimization method in the MIXED procedure, is available in the GLIMMIX procedure through the TECHNIQUE=NRRIDG option in the NLOPTIONS statement. Because of differences in the line-search methods, update methods, and the convergence criteria, you might get slightly different estimates with the two procedures in some instances. The GLIMMIX procedure, for example, monitors several convergence criteria simultaneously.

- You can produce predicted values, residuals, and confidence limits for predicted values with both procedures. The mechanics are slightly different, however. With the MIXED procedure you use the OUTPM= and OUTP= options in the MODEL statement to write statistics to data sets. With the GLIMMIX procedure you use the OUTPUT statement and indicate with keywords which "flavor" of a statistic to compute.

- The following GLIMMIX statements are not available in the MIXED procedure: COVTEST, EFFECT, FREQ, LSMESTIMATE, OUTPUT, and programming statements.
- A sampling-based Bayesian analysis as through the PRIOR statement in the MIXED procedure is not available in the GLIMMIX procedure.
- In the GLIMMIX procedure, several RANDOM statement options apply to the RANDOM statement in which they are specified. For example, the following statements in the GLIMMIX procedure request that the solution vector be printed for the A and A*B*C random effects and that the **G** matrix corresponding to the A*B interaction random effect be displayed:

```
random a     / s;
random a*b   / G;
random a*b*c / alpha=0.04;
```

  Confidence intervals with a 0.96 coverage probability are produced for the solutions of the A*B*C effect. In the MIXED procedure, the S option, for example, when specified in one RANDOM statement, applies to all RANDOM statements.

- If you select nonmissing values in the *value-list* of the DDF= option in the MODEL statement, PROC GLIMMIX uses these values to override degrees of freedom for this effect that might be determined otherwise. For example, the following statements request that the denominator degrees of freedom for tests and confidence intervals involving the A effect be set to 4:

```
proc glimmix;
   class block a b;
   model y = a b a*b / s ddf=4,.,. ddfm=satterthwaite;
   random block a*block / s;
   lsmeans a b a*b  / diff;
run;
```

  In the example, this applies to the "Type III Tests of Fixed Effects," "Least Squares Means," and "Differences of Least Squares Means" tables. In the MIXED procedure, the Satterthwaite approximation overrides the DDF= specification.

- The DDFM=BETWITHIN degrees-of-freedom method in the GLIMMIX procedure requires that the data be processed by subjects; see the section "Processing by Subjects" on page 2247.
- When you add the response variable to the CLASS statement, PROC GLIMMIX defaults to the multinomial distribution. Adding the response variable to the CLASS statement in PROC MIXED has no effect on the fitted model.
- For ODS purposes, the name of the table for the solution of fixed effects is "SolutionF" in the MIXED procedure. In PROC GLIMMIX, the name of the table that contains fixed-effects solutions is "ParameterEstimates." In generalized linear models, this table also contains scale parameters and overdispersion parameters. The MIXED procedure always produces a "Covariance Parameter Estimates" table. The GLIMMIX procedure produces this table only in mixed models or models with nontrivial R-side covariance structure.

- If you compute predicted values in the GLIMMIX procedure in a model with only R-side random components and missing values for the dependent variable, the predicted values will *not* be kriging predictions as is the case with the MIXED procedure.

Singly or Doubly Iterative Fitting

Depending on the structure of your model, the GLIMMIX procedure determines the appropriate approach for estimating the parameters of the model. The elementary algorithms fall into three categories:

1. **Noniterative algorithms**
   A closed form solution exists for all model parameters. Standard linear models with homoscedastic, uncorrelated errors can be fit with noniterative algorithms.

2. **Singly iterative algorithms**
   A single optimization, consisting of one or more iterations, is performed to obtain solutions for the parameter estimates by numerical techniques. Linear mixed models for normal data can be fit with singly iterative algorithms. Laplace and quadrature estimation for generalized linear mixed models uses a singly iterative algorithm with a separate suboptimization to compute the random-effects solutions as modes of the log-posterior distribution.

3. **Doubly iterative algorithms**
   A model of simpler structure is derived from the target model. The parameters of the simpler model are estimated by noniterative or singly iterative methods. Based on these new estimates, the model of simpler structure is rederived and another estimation step follows. The process continues until changes in the parameter estimates are sufficiently small between two recomputations of the simpler model or until some other criterion is met. The rederivation of the model can often be cast as a change of the response to some pseudo-data along with an update of implicit model weights.

Obviously, noniterative algorithms are preferable to singly iterative ones, which in turn are preferable to doubly iterative algorithms. Two drawbacks of doubly iterative algorithms based on linearization are that likelihood-based measures apply to the pseudo-data, not the original data, and that at the outer level the progress of the algorithm is tied to monitoring the parameter estimates. The advantage of doubly iterative algorithms, however, is to offer—at convergence—the statistical inference tools that apply to the simpler models.

The output and log messages contain information about which algorithm is employed. For a noniterative algorithm, PROC GLIMMIX produces a message that no optimization was performed. Noniterative algorithms are employed automatically for normal data with identity link.

You can determine whether a singly or doubly iterative algorithm was used, based on the "Iteration History" table and the "Convergence Status" table (Figure 38.17).

Figure 38.17 Iteration History and Convergence Status in Singly Iterative Fit

The GLIMMIX Procedure

Iteration History

Iteration	Restarts	Evaluations	Objective Function	Change	Max Gradient
0	0	4	83.039723731	.	13.63536
1	0	3	82.189661988	0.85006174	0.281308
2	0	3	82.189255211	0.00040678	0.000174
3	0	3	82.189255211	0.00000000	1.05E-10

Convergence criterion (GCONV=1E-8) satisfied.

The "Iteration History" table contains the Evaluations column that shows how many function evaluations were performed in a particular iteration. The convergence status message informs you which convergence criterion was met when the estimation process concluded. In a singly iterative fit, the criterion is one that applies to the optimization. In other words, it is one of the criteria that can be controlled with the NLOPTIONS statement: see the ABSCONV=, ABSFCONV=, ABSGCONV=, ABSXCONV=, FCONV=, or GCONV= option.

In a doubly iterative fit, the "Iteration History" table does not contain an Evaluations column. Instead it displays the number of iterations within an optimization (Subiterations column in Figure 38.18).

Figure 38.18 Iteration History and Convergence Status in Doubly Iterative Fit

Iteration History

Iteration	Restarts	Subiterations	Objective Function	Change	Max Gradient
0	0	5	79.688580269	0.11807224	7.851E-7
1	0	3	81.294622554	0.02558021	8.209E-7
2	0	2	81.438701534	0.00166079	4.061E-8
3	0	1	81.444083567	0.00006263	2.292E-8
4	0	1	81.444265216	0.00000421	0.000025
5	0	1	81.444277364	0.00000383	0.000023
6	0	1	81.444266322	0.00000348	0.000021
7	0	1	81.44427636	0.00000316	0.000019
8	0	1	81.444267235	0.00000287	0.000017
9	0	1	81.44427553	0.00000261	0.000016
10	0	1	81.44426799	0.00000237	0.000014
11	0	1	81.444274844	0.00000216	0.000013
12	0	1	81.444268614	0.00000196	0.000012
13	0	1	81.444274277	0.00000178	0.000011
14	0	1	81.444269129	0.00000162	9.772E-6
15	0	0	81.444273808	0.00000000	9.102E-6

Convergence criterion (PCONV=1.11022E-8) satisfied.

The Iteration column then counts the number of optimizations. The "Convergence Status" table indicates that the estimation process concludes when a criterion is met that monitors the parameter estimates across optimization, namely the PCONV= or ABSPCONV= criterion.

You can control the optimization process with the GLIMMIX procedure through the NLOPTIONS statement. Its options affect the individual optimizations. In a doubly iterative scheme, these apply to all optimizations.

The default optimization techniques are TECHNIQUE=NONE for noniterative estimation, TECHNIQUE=NEWRAP for singly iterative methods in GLMs, TECHNIQUE=NRRIDG for pseudo-likelihood estimation with binary data, and TECHNIQUE=QUANEW for other mixed models.

Default Estimation Techniques

Based on the structure of the model, the GLIMMIX procedure selects the estimation technique for estimating the model parameters. If you fit a generalized linear mixed model, you can change the estimation technique with the METHOD= option in the PROC GLIMMIX statement. The defaults are determined as follows:

- generalized linear model
  - normal distribution: restricted maximum likelihood
  - all other distributions: maximum likelihood
- generalized linear model with overdispersion
  Parameters ($\boldsymbol{\beta}$; ϕ, if present) are estimated by (restricted) maximum likelihood as for generalized linear models. The overdispersion parameter is estimated from the Pearson statistic after all other parameters have been estimated.
- generalized linear mixed models
  The default technique is METHOD=RSPL, corresponding to maximizing the residual log pseudo-likelihood with an expansion about the current solutions of the best linear unbiased predictors of the random effects. In models for normal data with identity link, METHOD=RSPL and METHOD=RMPL are equivalent to restricted maximum likelihood estimation, and METHOD=MSPL and METHOD=MMPL are equivalent to maximum likelihood estimation. This is reflected in the labeling of statistics in the "Fit Statistics" table.

Default Output

The following sections describe the output that PROC GLIMMIX produces by default. The output is organized into various tables, which are discussed in the order of appearance. Note that the contents of a table can change with the estimation method or the model being fit.

Model Information

The "Model Information" table displays basic information about the fitted model, such as the link and variance functions, the distribution of the response, and the data set. If important model quantities—for example, the response, weights, link, or variance function—are user-defined, the "Model Information" table displays the final assignment to the respective variable, as determined from your programming statements. If the table indicates that the variance matrix is blocked by an effect, then PROC GLIMMIX processes the data by subjects. The "Dimensions" table displays the number of subjects. For more information about processing by subjects, see the section "Processing by Subjects" on page 2247. For ODS purposes, the name of the "Model Information" table is "ModelInfo."

Class Level Information

The "Class Level Information" table lists the levels of every variable specified in the CLASS statement. You should check this information to make sure that the data are correct. You can adjust the order of the CLASS variable levels with the ORDER= option in the PROC GLIMMIX statement. For ODS purposes, the name of the "Class Level Information" table is "ClassLevels."

Number of Observations

The "Number of Observations" table displays the number of observations read from the input data set and the number of observations used in the analysis. If you specify a FREQ statement, the table also displays the sum of frequencies read and used. If the *events/trials* syntax is used for the response, the table also displays the number of events and trials used in the analysis. For ODS purposes, the name of the "Number of Observations" table is "NObs."

Response Profile

For binary and multinomial data, the "Response Profile" table displays the Ordered Value from which the GLIMMIX procedure determines the following:

- the probability being modeled for binary data
- the ordering of categories for ordinal data
- the reference category for generalized logit models

For each response category level, the frequency used in the analysis is reported. The section "Response-Level Ordering and Referencing" on page 2267 explains how you can use the DESCENDING, EVENT=, ORDER=, and REF= options to affect the assignment of Ordered Values to the response categories. For ODS purposes, the name of the "Response Profile" table is "ResponseProfile."

Dimensions

The "Dimensions" table displays information from which you can determine the size of relevant matrices in the model. This table is useful in determining CPU time and memory requirements. For ODS purposes, the name of the "Dimensions" table is "Dimensions."

Optimization Information

The "Optimization Information" table displays important details about the optimization process.

The optimization technique that is displayed in the table is the technique that applies to any single optimization. For singly iterative methods that is *the* optimization method.

The number of parameters that are updated in the optimization equals the number of parameters in this table minus the number of equality constraints. The number of constraints is displayed if you fix covariance parameters with the HOLD= option in the PARMS statement. The GLIMMIX procedure also lists the number of upper and lower boundary constraints. Note that the procedure might impose boundary constraints for certain parameters, such as variance components and correlation parameters. Covariance parameters for which a HOLD= was issued have an upper and lower boundary equal to the parameter value.

If a residual scale parameter is profiled from the optimization, it is also shown in the "Optimization Information" table.

In a GLMM for which the parameters are estimated by one of the linearization methods, you need to initiate the process of computing the pseudo-response. This can be done based on existing estimates of the fixed effects, or by using the data themselves—possibly after some suitable adjustment—as an estimate of the initial mean. The default in PROC GLIMMIX is to use the data themselves to derive initial estimates of the mean function and to construct the pseudo-data. The "Optimization Information" table shows how the pseudo-data are determined initially. Note that this issue is separate from the determination of starting values for the covariance parameters. These are computed as minimum variance quadratic unbiased estimates (with 0 priors, MIVQUE0; Goodnight 1978b) or obtained from the *value-list* in the PARMS statement.

For ODS purposes, the name of the table is "OptInfo."

Iteration History

The "Iteration History" table describes the progress of the estimation process. In singly iterative methods, the table displays the following:

- the iteration count, Iteration
- the number of restarts, Restarts
- the number of function evaluations, Evaluations
- the objective function, Objective

- the change in the objective function, Change
- the absolute value of the largest (projected) gradient, MaxGradient

Note that the change in the objective function is not the convergence criterion monitored by the GLIMMIX procedure. PROC GLIMMIX tracks several convergence criteria simultaneously; see the ABSCONV=, ABSFCONV=, ABSGCONV=, ABSXCONV=, FCONV=, or GCONV= option in the NLOPTIONS statement.

For doubly iterative estimation methods, the "Iteration History" table does not display the progress of the individual optimizations; instead, it reports on the progress of the outer iterations. Every row of the table then corresponds to an update of the linearization, the computation of a new set of pseudo-data, and a new optimization. In the listing, PROC GLIMMIX displays the following:

- the optimization count, Iteration
- the number of restarts, Restarts
- the number of iterations per optimization, Subiterations
- the change in the parameter estimates, Change
- the absolute value of the largest (projected) gradient at the end of the optimization, MaxGradient

By default, the change in the parameter estimates is expressed in terms of the relative PCONV criterion. If you request an absolute criterion with the ABSPCONV option of the PROC GLIMMIX statement, the change reflects the largest absolute difference since the last optimization.

If you specify the ITDETAILS option in the PROC GLIMMIX statement, parameter estimates and their gradients are added to the "Iteration History" table. For ODS purposes, the name of the "Iteration History" table is "IterHistory."

Convergence Status

The "Convergence Status" table contains a status message describing the reason for termination of the optimization. The message is also written to the log. For ODS purposes, the name of the "Convergence Status" table is "ConvergenceStatus," and you can query the nonprinting numeric variable Status to check for a successful optimization. This is useful in batch processing, or when processing BY groups, such as in simulations. Successful optimizations are indicated by the value 0 of the Status variable.

Fit Statistics

The "Fit Statistics" table provides statistics about the estimated model. The first entry of the table corresponds to the negative of twice the (possibly restricted) log likelihood, log pseudo-likelihood, or log quasi-likelihood. If the estimation method permits the true log likelihood or residual log likelihood, the description of the first entry reads accordingly. Otherwise, the fit statistics are preceded by the words *Pseudo-* or *Quasi-*, for Pseudo- and Quasi-Likelihood estimation, respectively.

Note that the (residual) log pseudo-likelihood in a GLMM is the (residual) log likelihood of a linearized model. You should not compare these values across different statistical models, even if the models are nested with respect to fixed and/or G-side random effects. It is possible that between two nested models the larger model has a smaller pseudo-likelihood. For this reason, IC=NONE is the default for GLMMs fit by pseudo-likelihood methods.

See the IC= option of the PROC GLIMMIX statement and Table 38.2 for the definition and computation of the information criteria reported in the "Fit Statistics" table.

For generalized linear models, the GLIMMIX procedure reports Pearson's chi-square statistic

$$X^2 = \sum_i \frac{w_i (y_i - \widehat{\mu}_i)^2}{a(\widehat{\mu}_i)}$$

where $a(\widehat{\mu}_i)$ is the variance function evaluated at the estimated mean.

For GLMMs, the procedure typically reports a generalized chi-square statistic,

$$X_g^2 = \widehat{\mathbf{r}}' \mathbf{V}(\widehat{\boldsymbol{\theta}}^*)^{-1} \widehat{\mathbf{r}}$$

so that the ratio of X^2 or X_g^2 and the degrees of freedom produces the usual residual dispersion estimate.

If the R-side scale parameter ϕ is not extracted from $\mathbf{V}$, the GLIMMIX procedure computes

$$X_g^2 = \widehat{\mathbf{r}}' \mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1} \widehat{\mathbf{r}}$$

as the generalized chi-square statistic. This is the case, for example, if R-side covariance structures are varied by a GROUP= effect or if the scale parameter is not profiled for an R-side TYPE=CS, TYPE=SP, TYPE=AR, TYPE=TOEP, or TYPE=ARMA covariance structure.

For METHOD=LAPLACE, the generalized chi-square statistic is not reported. Instead, the Pearson statistic for the conditional distribution appears in the "Conditional Fit Statistics" table.

If your model contains smooth components (such as TYPE=RSMOOTH), then the "Fit Statistics" table also displays the residual degrees of freedom of the smoother. These degrees of freedom are computed as

$$df_{smooth,res} = f - \text{trace}(\mathbf{S})$$

where $\mathbf{S}$ is the "smoother" matrix—that is, the matrix that produces the predicted values on the linked scale.

For ODS purposes, the name of the "Fit Statistics" table is "FitStatistics."

Covariance Parameter Estimates

In a GLMM, the "Covariance Parameter Estimates" table displays the estimates of the covariance parameters and their asymptotic standard errors. This table is produced only for generalized linear *mixed* models. In generalized linear models with scale parameter, or when an overdispersion parameter is present, the estimates of parameters related to the dispersion are displayed in the "Parameter Estimates" table.

The standard error of the covariance parameters is determined from the diagonal entries of the asymptotic variance matrix of the covariance parameter estimates. You can display this matrix with the ASYCOV option in the PROC GLIMMIX statement.

For ODS purposes, the name of the "Covariance Parameter Estimates" table is "CovParms."

Type III Tests of Fixed Effects

The "Type III Tests of Fixed Effects" table contains hypothesis tests for the significance of each of the fixed effects specified in the MODEL statement. By default, PROC GLIMMIX computes these tests by first constructing a Type III $\mathbf{L}$ matrix for each effect; see Chapter 15, "The Four Types of Estimable Functions." The $\mathbf{L}$ matrix is then used to construct the test statistic

$$F = \frac{\widehat{\boldsymbol{\beta}}'\mathbf{L}'(\mathbf{L}\mathbf{Q}\mathbf{L}')^{-1}\mathbf{L}\widehat{\boldsymbol{\beta}}}{\text{rank}(\mathbf{L}\mathbf{Q}\mathbf{L}')}$$

where the matrix $\mathbf{Q}$ depends on the estimation method and options. For example, in a GLMM, the default is $\mathbf{Q} = (\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}\mathbf{X})^{-}$, where $\mathbf{V}(\boldsymbol{\theta})$ is the marginal variance of the pseudo-response. If you specify the DDFM=KENWARDROGER option, $\mathbf{Q}$ is the estimated variance matrix of the fixed effects, adjusted by the method of Kenward and Roger (1997). If the EMPIRICAL= option is in effect, $\mathbf{Q}$ corresponds to the selected sandwich estimator.

You can use the HTYPE= option in the MODEL statement to obtain tables of Type I (sequential) tests and Type II (adjusted) tests in addition to or instead of the table of Type III (partial) tests.

For ODS purposes, the names of the "Type I Tests of Fixed Effects" through the "Type III Tests of Fixed Effects" tables are "Tests1" through "Tests3," respectively.

Notes on Output Statistics

Table 38.11 lists the statistics computed with the OUTPUT statement of the GLIMMIX procedure and their default names. This section provides further details about these statistics.

The distinction between prediction and confidence limits in Table 38.11 stems from the involvement of the predictors of the random effects. If the random-effect solutions (BLUPs, EBES) are involved, then the associated standard error used in computing the limits are standard errors of prediction rather than standard errors of estimation. The prediction limits are *not* limits for the prediction of a new observation.

The Pearson residuals in Table 38.11 are "Pearson-type" residuals, because the residuals are standardized by the square root of the marginal or conditional variance of an observation. Traditionally, Pearson residuals in generalized linear models are divided by the square root of the variance function. The GLIMMIX procedure divides by the square root of the variance so that marginal and conditional residuals have similar expressions. In other words, scale and overdispersion parameters are included.

When residuals or predicted values involve only the fixed effects part of the linear predictor (that is, $\widehat{\eta}_m = \mathbf{x}'\widehat{\boldsymbol{\beta}}$), then all model quantities are computed based on this predictor. For example, if

the variance by which to standardize a marginal residual involves the variance function, then the variance function is also evaluated at the marginal mean, $g^{-1}(\widehat{\eta}_m)$. Thus the residuals $p - \widehat{\eta}$ and $p_m - \widehat{\eta}_m$ can also be expressed as $(y - \mu)/\partial\mu$ and $(y - \mu_m)/\partial\mu_m$, respectively, where $\partial\mu$ is the derivative with respect to the linear predictor. To construct the residual $p - \widehat{\eta}_m$ in a GLMM, you can add the value of _ZGAMMA_ to the conditional residual $p - \widehat{\eta}$. If the predictor involves the BLUPs, then all relevant expressions and evaluations involve the conditional mean $g^{-1}(\widehat{\eta})$.

The naming convention to add "PA" to quantities not involving the BLUPs is chosen to suggest the concept of a population average. When the link function is nonlinear, these are not truly population-averaged quantities, because $g^{-1}(\mathbf{x}'\boldsymbol{\beta})$ does not equal $\mathrm{E}[Y]$ in the presence of random effects. For example, if

$$\mu_i = g^{-1}(\mathbf{x}_i'\boldsymbol{\beta} + \mathbf{z}_i'\boldsymbol{\gamma}_i)$$

is the conditional mean for subject i, then

$$g^{-1}(\mathbf{x}_i'\widehat{\boldsymbol{\beta}})$$

does not estimate the average response in the population of subjects but the response of the average subject (the subject for which $\boldsymbol{\gamma}_i = \mathbf{0}$). For models with identity link, the average response and the response of the average subject are identical.

The GLIMMIX procedure obtains standard errors on the scale of the mean by the delta method. If the link is a nonlinear function of the linear predictor, these standard errors are only approximate. For example,

$$\mathrm{Var}[g^{-1}(\widehat{\eta}_m)] \doteq \left(\frac{\partial g^{-1}(t)}{\partial t}_{|\widehat{\eta}_m}\right)^2 \mathrm{Var}[\widehat{\eta}_m]$$

Confidence limits on the scale of the data are usually computed by applying the inverse link function to the confidence limits on the linked scale. The resulting limits on the data scale have the same coverage probability as the limits on the linked scale, but they are possibly asymmetric.

In generalized logit models, confidence limits on the mean scale are based on symmetric limits about the predicted mean in a category. Suppose that the multinomial response in such a model has J categories. The probability of a response in category i is computed as

$$\widehat{\mu}_i = \frac{\exp\{\widehat{\eta}_i\}}{\sum_{j=1}^{J}\exp\{\widehat{\eta}_i\}}$$

The variance of $\widehat{\mu}_i$ is then approximated as

$$\mathrm{Var}[\widehat{\mu}_i] \doteq \zeta = \boldsymbol{v}_i'\mathrm{Var}\left[\begin{array}{cccc}\widehat{\eta}_1 & \widehat{\eta}_2 & \cdots & \widehat{\eta}_J\end{array}\right]\boldsymbol{v}_i$$

where $\boldsymbol{v}_i$ is a $J \times 1$ vector with kth element

$$\begin{array}{ll}\widehat{\mu}_i(1-\widehat{\mu}_i) & i = k\\ -\widehat{\mu}_i\widehat{\mu}_k & i \neq k\end{array}$$

The confidence limits in the generalized logit model are then obtained as

$$\widehat{\mu}_i \pm t_{\nu,\alpha/2}\sqrt{\zeta}$$

where $t_{\nu,\alpha/2}$ is the $100 \times (1 - \alpha/2)$ percentile from a t distribution with ν degrees of freedom. Confidence limits are truncated if they fall outside the $[0, 1]$ interval.

ODS Table Names

Each table created by PROC GLIMMIX has a name associated with it, and you must use this name to reference the table when you use ODS statements. These names are listed in Table 38.19.

Table 38.19 ODS Tables Produced by PROC GLIMMIX

Table Name	Description	Required Statement / Option
AsyCorr	asymptotic correlation matrix of covariance parameters	PROC GLIMMIX ASYCORR
AsyCov	asymptotic covariance matrix of covariance parameters	PROC GLIMMIX ASYCOV
CholG	Cholesky root of the estimated **G** matrix	RANDOM / GC
CholV	Cholesky root of blocks of the estimated **V** matrix	RANDOM / VC
ClassLevels	level information from the CLASS statement	default output
Coef	**L** matrix coefficients	E option in MODEL, CONTRAST, ESTIMATE, LSMESTIMATE, or LSMEANS; ELSM option in LSMESTIMATE
ColumnNames	name association for OUTDESIGN data set	PROC GLIMMIX OUTDESIGN(NAMES)
CondFitStatistics	conditional fit statistics	PROC GLIMMIX METHOD=LAPLACE
Contrasts	results from the CONTRAST statements	CONTRAST
ConvergenceStatus	status of optimization at conclusion	default output
CorrB	approximate correlation matrix of fixed-effects parameter estimates	MODEL / CORRB
CovB	approximate covariance matrix of fixed-effects parameter estimates	MODEL / COVB
CovBDetails	details about model-based and/or adjusted covariance matrix of fixed effects	MODEL / COVB(DETAILS)
CovBI	inverse of approximate covariance matrix of fixed-effects parameter estimates	MODEL / COVBI
CovBModelBased	model-based (unadjusted) covariance matrix of fixed effects if DDFM=KR or EMPIRICAL option is used	MODEL / COVB(DETAILS)
CovParms	estimated covariance parameters in GLMMs	default output (in GLMMs)
CovTests	results from COVTEST statements (except for confidence bounds)	COVTEST

Table 38.19 *continued*

Table Name	Description	Required Statement / Option
Diffs	differences of LS-means	LSMEANS / DIFF (or PDIFF)
Dimensions	dimensions of the model	default output
Estimates	results from ESTIMATE statements	ESTIMATE
FitStatistics	fit statistics	default
G	estimated **G** matrix	RANDOM / G
GCorr	correlation matrix from the estimated **G** matrix	RANDOM / GCORR
Hessian	Hessian matrix (observed or expected)	PROC GLIMMIX HESSIAN
InvCholG	inverse Cholesky root of the estimated **G** matrix	RANDOM / GCI
InvCholV	inverse Cholesky root of the blocks of the estimated **V** matrix	RANDOM / VCI
InvG	inverse of the estimated **G** matrix	RANDOM / GI
InvV	inverse of blocks of the estimated **V** matrix	RANDOM / VI
IterHistory	iteration history	default output
kdTree	*k-d* tree information	RANDOM / TYPE=RSMOOTH KNOTMETHOD= KDTREE(TREEINFO)
KnotInfo	knot coordinates of low-rank spline smoother	RANDOM / TYPE=RSMOOTH KNOTINFO
LSMeans	LS-means	LSMEANS
LSMEstimates	estimates among LS-means	LSMESTIMATE
LSMFtest	*F* test for LSMESTIMATEs	LSMESTIMATE / FTEST
LSMLines	lines display for LS-means	LSMEANS / LINES
ModelInfo	model information	default output
NObs	number of observations read and used, number of trials and events	default output
OddsRatios	odds ratios of parameter estimates	MODEL / ODDSRATIO
OptInfo	optimization information	default output
ParameterEstimates	fixed-effects solution; overdispersion and scale parameter in GLMs	MODEL / S
ParmSearch	parameter search values	PARMS
QuadCheck	adaptive recalculation of quadrature approximation at solution	METHOD=QUAD(QCHECK)
ResponseProfile	response categories and category modeled	default output in models with binary or nominal response
Slices	tests of LS-means slices	LSMEANS / SLICE=
SliceDiffs	differences of simple LS-means effects	LSMEANS / SLICEDIFF=
SolutionR	random-effects solution vector	RANDOM / S
StandardizedCoefficients	fixed-effects solutions from centered and/or scaled model	MODEL / STDCOEF
Tests1	Type I tests of fixed effects	MODEL / HTYPE=1

Table 38.19 *continued*

Table Name	Description	Required Statement / Option
Tests2	Type II tests of fixed effects	MODEL / HTYPE=2
Tests3	Type III tests of fixed effects	default output
V	blocks of the estimated **V** matrix	RANDOM / V
VCorr	correlation matrix from the blocks of the estimated **V** matrix	RANDOM / VCORR

ODS Graphics

The following subsections provide information about the basic ODS statistical graphics produced by the GLIMMIX procedure. The graphics fall roughly into two categories: diagnostic plots and graphics for least squares means.

Diagnostic Plots

Residual Panels

There are three types of residual panels in the GLIMMIX procedure. Their makeup of four component plots is the same; the difference lies in the type of residual from which the panel is computed. Raw residuals are displayed with the PLOTS=RESIDUALPANEL option. Studentized residuals are displayed with the PLOTS=STUDENTPANEL option, and Pearson residuals with the PLOTS==PEARSONPANEL option. By default, conditional residuals are used in the construction of the panels if the model contains G-side random effects. For example, consider the following statements:

```
proc glimmix plots=residualpanel;
   class A;
   model y = x1 x2 / dist=Poisson;
   random int / sub=A;
run;
```

The parameters are estimated by a pseudo-likelihood method, and at the final stage pseudo-data are related to a linear mixed model with random intercepts. The residual panel is constructed from

$$r = p - \mathbf{x}'\widehat{\boldsymbol{\beta}} + \mathbf{z}'\widehat{\boldsymbol{\gamma}}$$

where p is the pseudo-data.

The following hypothetical data set contains yields of an industrial process. Material was available from five randomly selected vendors to produce a chemical reaction whose yield depends on two factors (pressure and temperature at 3 and 2 levels, respectively).

```
data Yields;
  input Vendor Pressure Temp Yield @@;
  datalines;
1  1  1  10.20    1  1  2   9.48    1  2  1   9.74
1  2  2   8.92    1  3  1  11.79    1  3  2   8.85
2  1  1  10.43    2  1  2  10.59    2  2  1  10.29
2  2  2  10.15    2  3  1  11.12    2  3  2   9.30
3  1  1   6.46    3  1  2   7.34    3  2  1   9.44
3  2  2   8.11    3  3  1   9.38    3  3  2   8.37
4  1  1   7.36    4  1  2   9.92    4  2  1  10.99
4  2  2  10.34    4  3  1  10.24    4  3  2   9.96
5  1  1  11.72    5  1  2  10.60    5  2  1  11.28
5  2  2   9.03    5  3  1  14.09    5  3  2   8.92
;
```

We consider here a linear mixed model with a two-way factorial fixed-effects structure for pressure and temperature effects and independent, homoscedastic random effects for the vendors. The following statements fit this model and request panels of marginal and conditional residuals:

```
ods graphics on;

proc glimmix data=Yields
      plots=residualpanel(conditional marginal);
   class Vendor Pressure Temp;
   model Yield = Pressure Temp Pressure*Temp;
   random vendor;
run;

ods graphics off;
```

The suboptions of the RESIDUALPANEL request produce two panels. The panel of conditional residuals is constructed from $y - \mathbf{x}'\widehat{\boldsymbol{\beta}} - \mathbf{z}'\widehat{\boldsymbol{\gamma}}$ (Figure 38.19). The panel of marginal residuals is constructed from $y - \mathbf{x}'\widehat{\boldsymbol{\beta}}$ (Figure 38.20). Note that these residuals are deviations from the observed data, because the model is a normal linear mixed model, and hence it does not involve pseudo-data. Whenever the random-effects solutions $\widehat{\boldsymbol{\gamma}}$ are involved in constructing residuals, the title of the residual graphics identifies them as conditional residuals (Figure 38.19).

Figure 38.19 Conditional Residuals

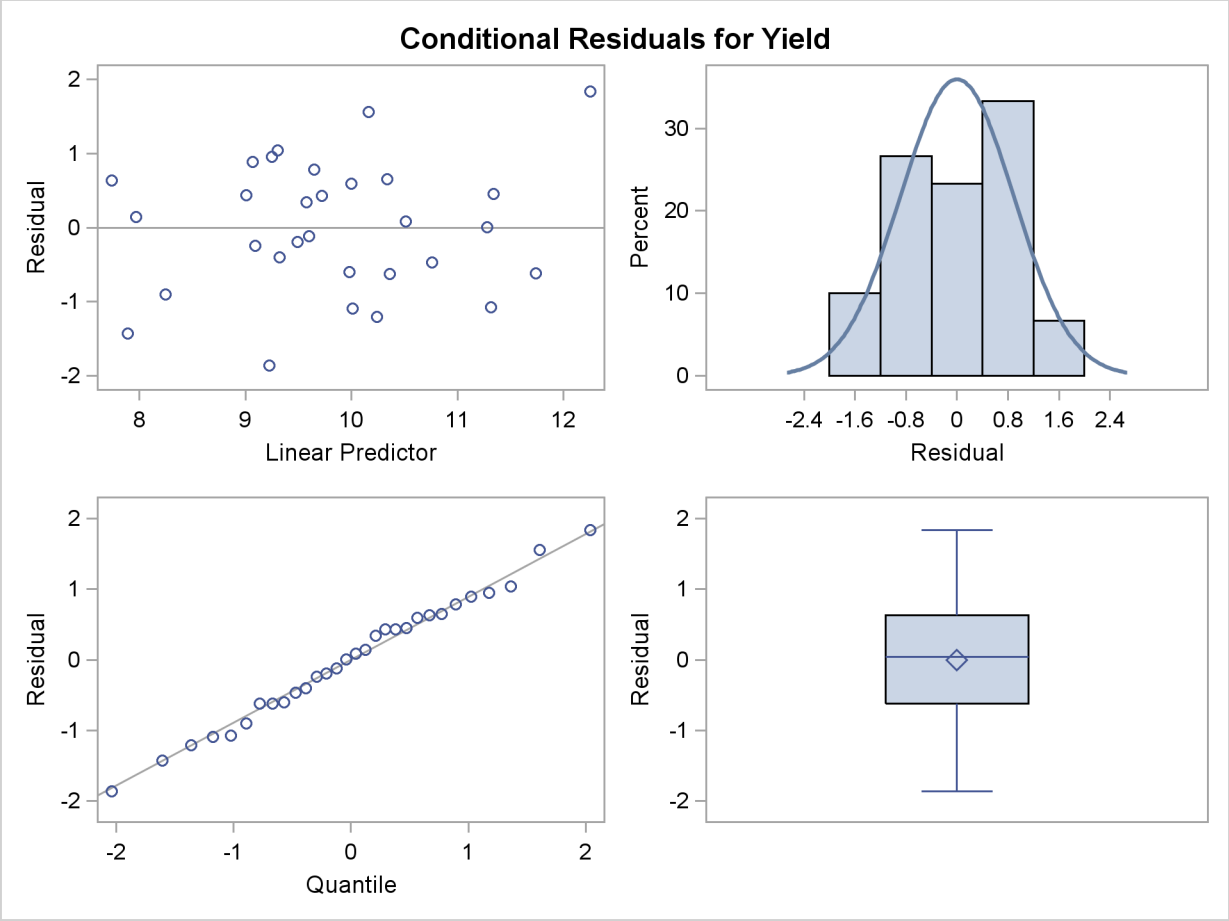
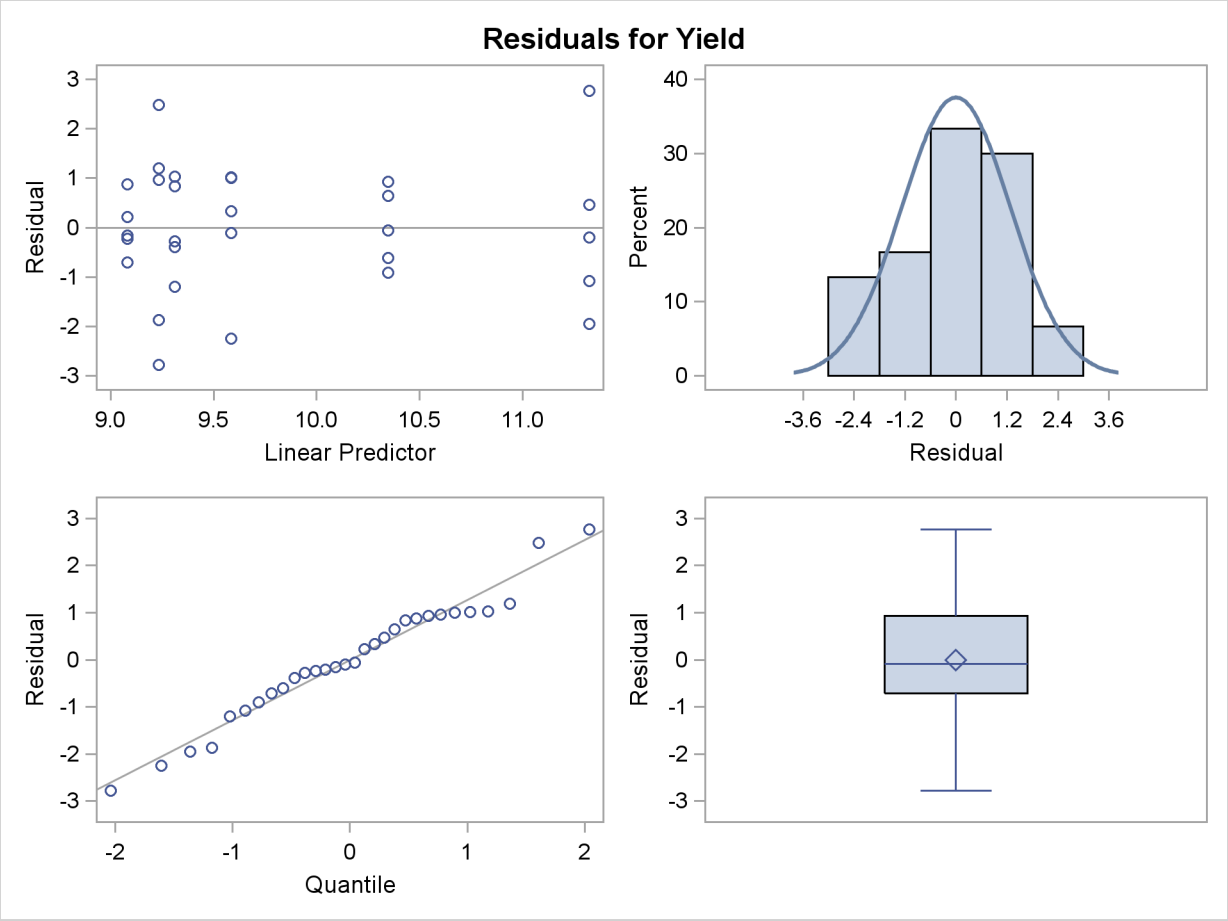


Figure 38.20 Marginal Residuals



The predictor takes on only six values for the marginal residuals, corresponding to the combinations of three temperature and two pressure levels. The assumption of a zero mean for the vendor random effect seems justified; the marginal residuals in the upper-left plot of Figure 38.20 do not exhibit any trend. The conditional residuals in Figure 38.19 are smaller and somewhat closer to normality compared to the marginal residuals.

Box Plots

You can produce box plots of observed data, pseudo-data, and various residuals for effects in your model that consist of classification variables. Because you might not want to produce box plots for all such effects, you can request subsets with the suboptions of the BOXPLOT option in the PLOTS option. The BOXPLOT request in the following PROC GLIMMIX statement produces box plots for the random effects—in this case, the vendor effect. By default, PROC GLIMMIX constructs box plots from conditional residuals. The MARGINAL, CONDITIONAL, and OBSERVED suboptions instruct the procedure to construct three box plots for each random effect: box plots of the observed data (Figure 38.21), the marginal residuals (Figure 38.22), and the conditional residuals (Figure 38.23).

```
ods graphics on;

proc glimmix data=Yields
       plots=boxplot(random marginal conditional observed);
   class Vendor Pressure Temp;
   model Yield = Pressure Temp Pressure*Temp;
   random vendor;
run;

ods graphics off;
```

The observed vendor means in Figure 38.21 are different; in particular, vendors 3 and 5 appear to differ from the other vendors and from each other. There is also heterogeneity of variance in the five groups. The marginal residuals in Figure 38.22 continue to show the differences in means by vendor, because vendor enters the model as a random effect. The marginal means are adjusted for vendor effects only in the sense that the vendor variance component affects the marginal variance that is involved in the generalized least squares solution for the pressure and temperature effects.

Figure 38.21 Box Plots of Observed Values

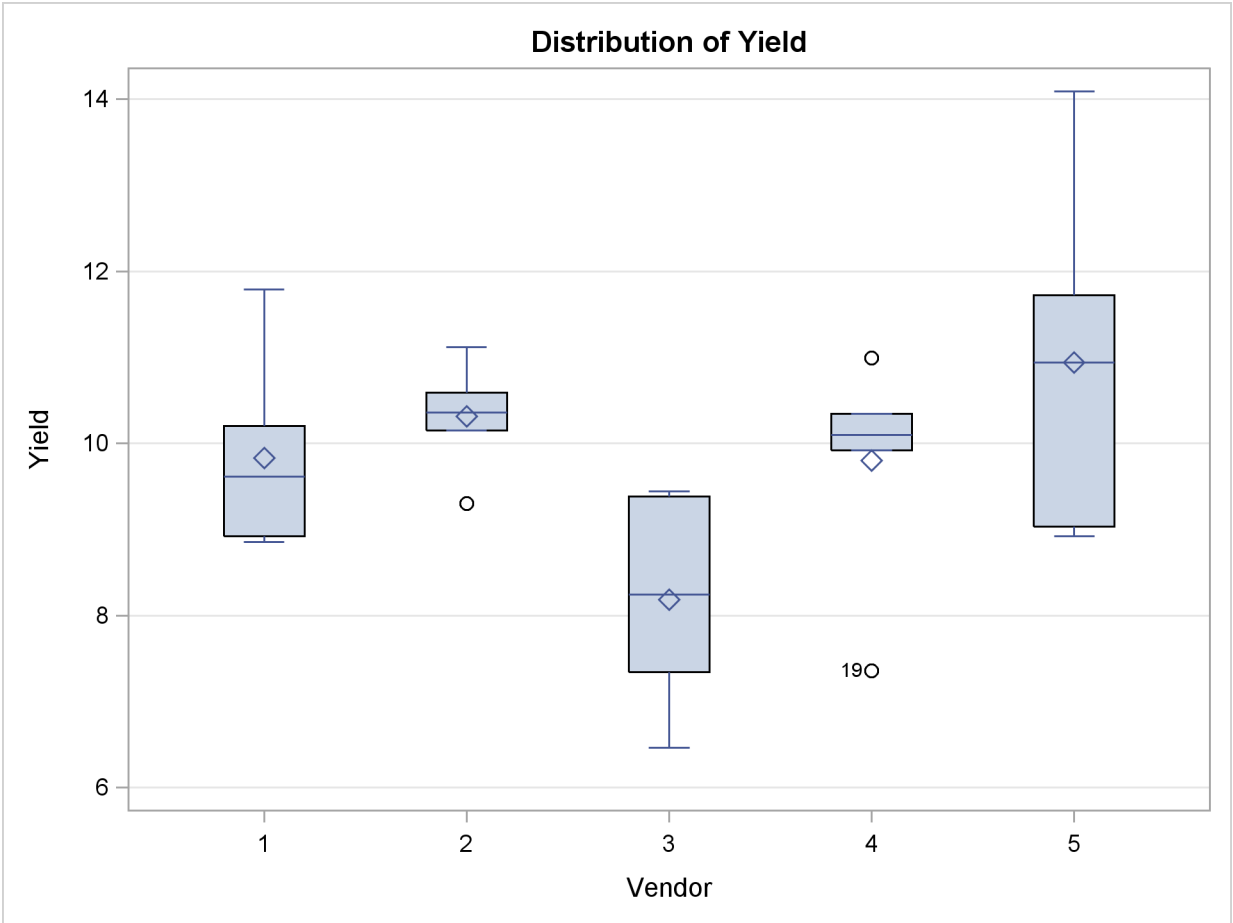
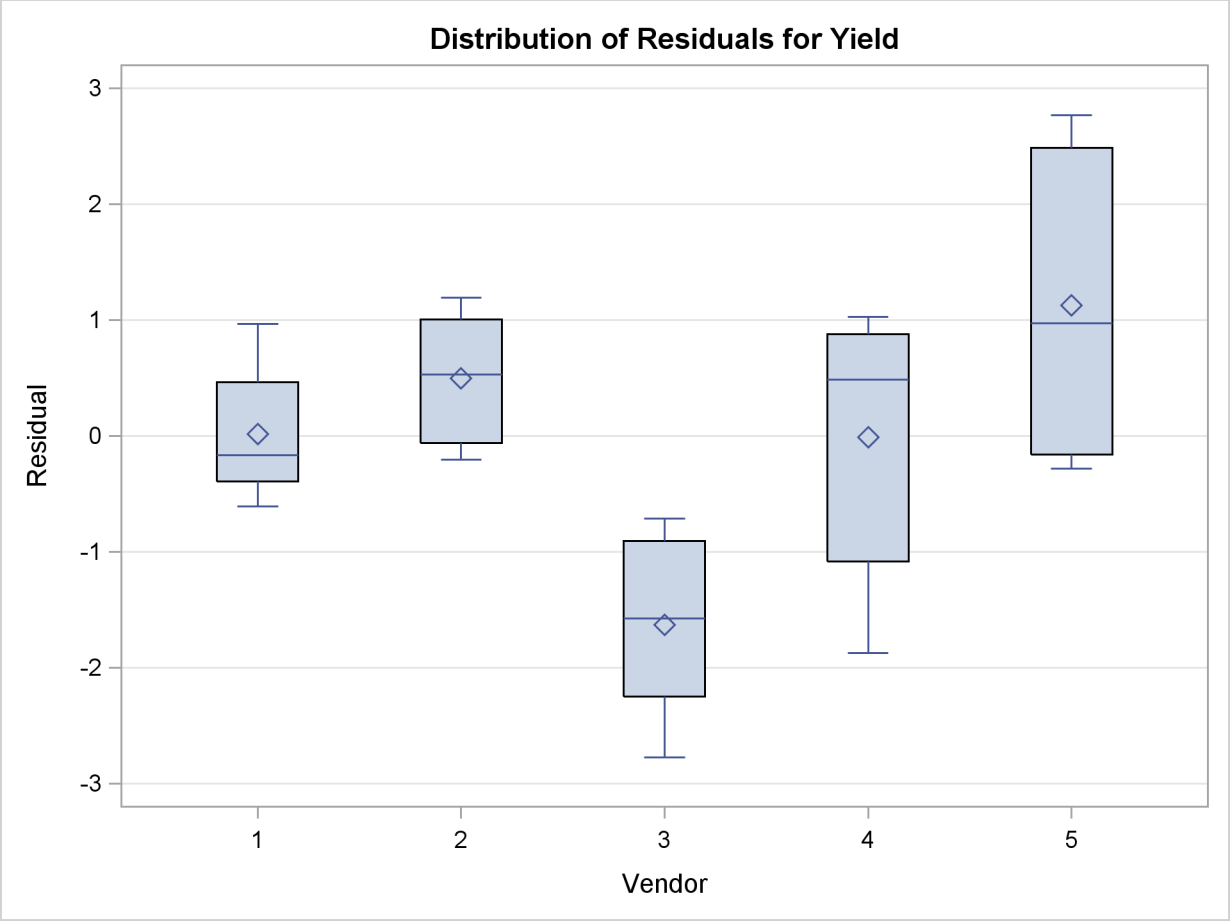
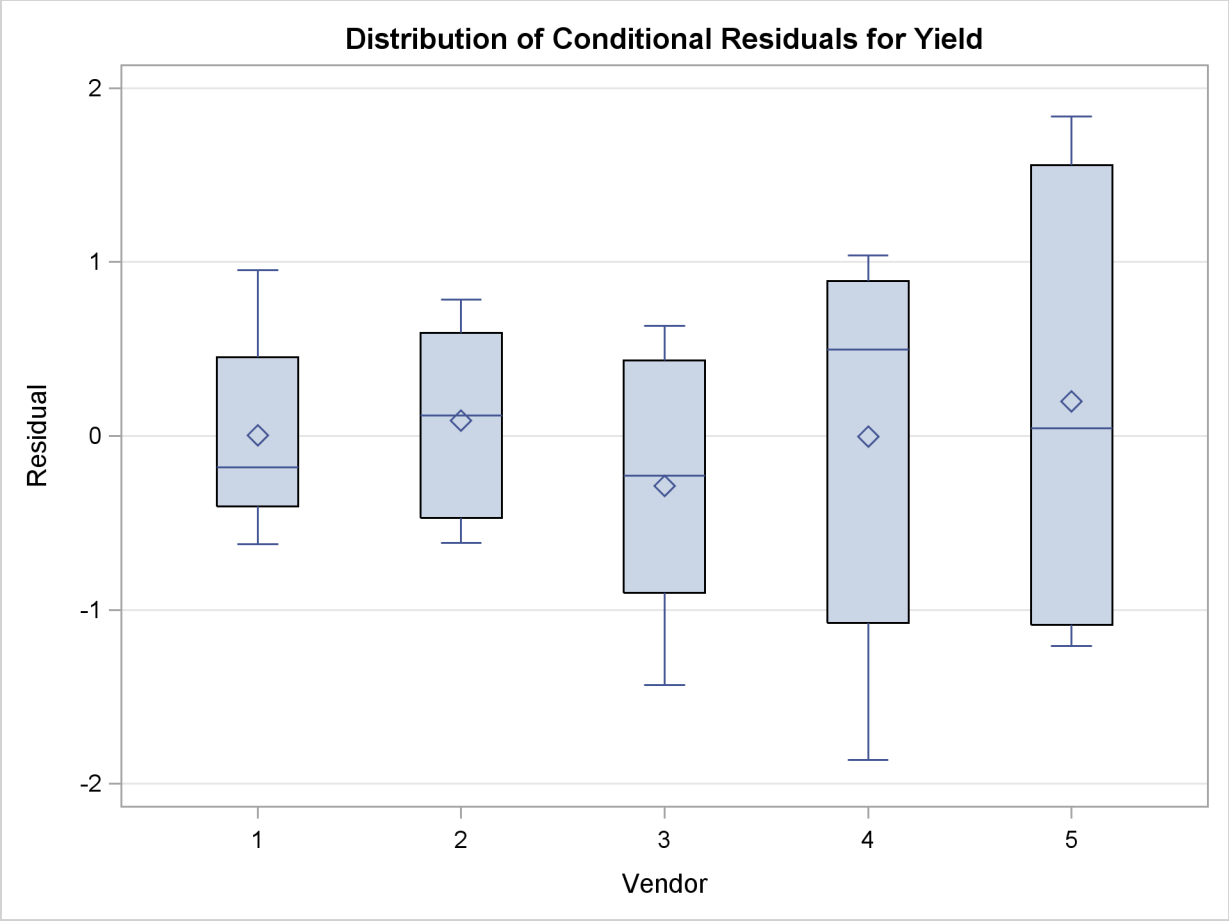


Figure 38.22 Box Plots of Marginal Residuals



The conditional residuals account for the vendor effects through the empirical BLUPs. The means and medians have stabilized near zero, but some heterogeneity in these residuals remains (Figure 38.23).

Figure 38.23 Box Plots of Conditional Residuals



Graphics for LS-Mean Comparisons

The following subsections provide information about the ODS statistical graphics for least squares means produced by the GLIMMIX procedure. Mean plots display marginal or interaction means. The diffogram, control plot, and ANOM plot display least squares mean comparisons.

Mean Plots

The following SAS statements request a plot of the Pressure×Temp means in which the pressure trends are plotted for each temperature.

```
ods graphics on;
ods select CovParms Tests3 MeanPlot;
proc glimmix data=Yields;
   class Vendor Pressure Temp;
   model Yield = Pressure Temp Pressure*Temp;
   random Vendor;
   lsmeans Pressure*Temp / plot=mean(sliceby=Temp join);
run;
ods graphics off;
```

There is a significant effect of temperature and an interaction between pressure and temperature (Figure 38.24). Notice that the pressure main effect might be masked by the interaction. Because of the interaction, temperature comparisons depend on the pressure and vice versa. The mean plot option requests a display of the Pressure × Temp least squares means with separate trends for each temperature (Figure 38.25).

Figure 38.24 Tests for Fixed Effects

The GLIMMIX Procedure

Covariance Parameter Estimates

Cov Parm	Estimate	Standard Error
Vendor	0.8602	0.7406
Residual	1.1039	0.3491

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Pressure	2	20	1.42	0.2646
Temp	1	20	6.48	0.0193
Pressure*Temp	2	20	3.82	0.0393

The interaction between the two effects is evident in the lack of parallelism in Figure 38.25. The masking of the pressure main effect can be explained by slopes of different sign for the two trends. Based on these results, inferences about the pressure effects are conducted for a specific temperature. For example, Figure 38.26 is produced by adding the following statement:

```
lsmeans pressure*temp / slicediff=temp slice=temp;
```

Figure 38.25 Interaction Plot for Pressure x Temperature

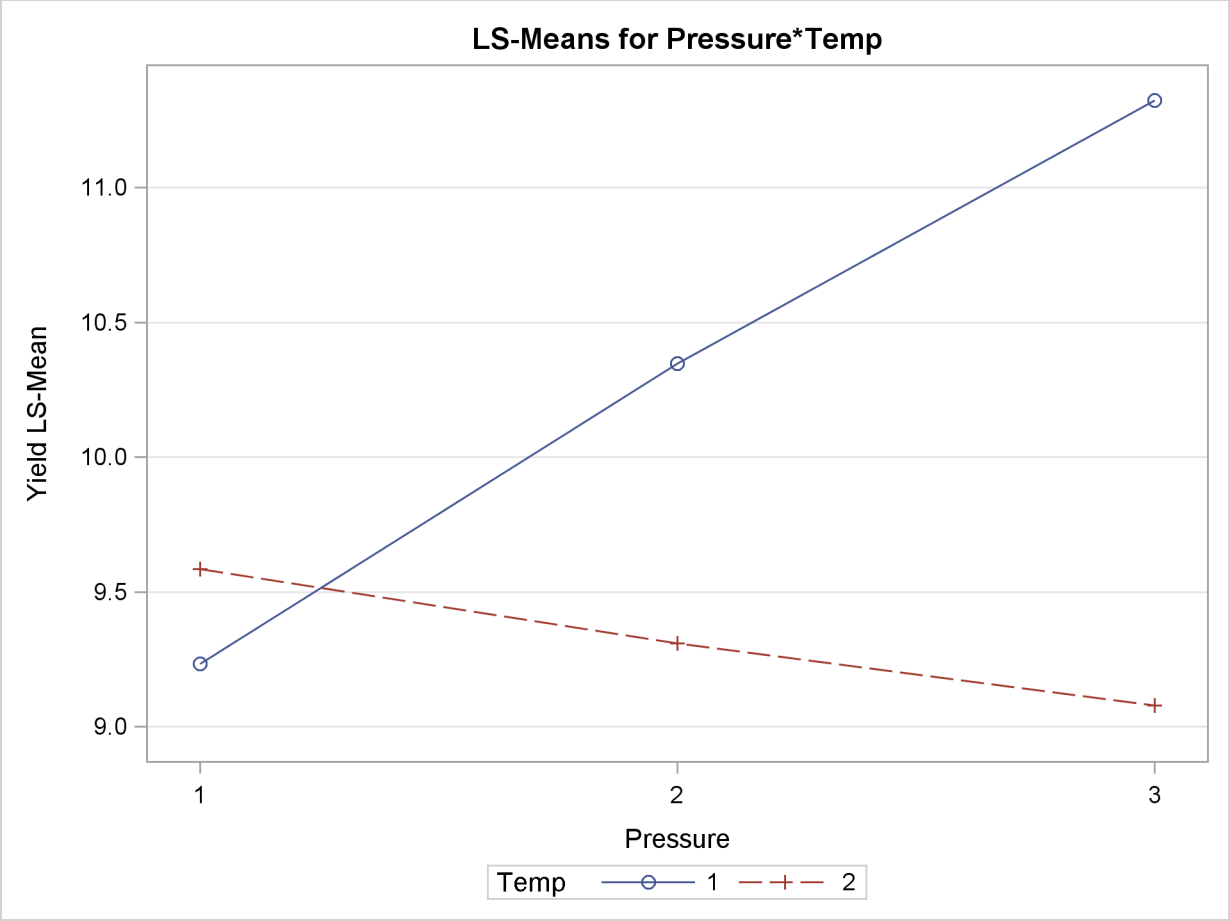


Figure 38.26 Pressure Comparisons at a Given Temperature

The GLIMMIX Procedure

Tests of Effect Slices for Pressure*Temp
Sliced By Temp

Temp	Num DF	Den DF	F Value	Pr > F
1	2	20	4.95	0.0179
2	2	20	0.29	0.7508

Figure 38.26 *continued*

Simple Effect Comparisons of Pressure*Temp Least Squares Means By Temp

Simple Effect Level	Pressure	_Pressure	Estimate	Standard Error	DF	t Value	Pr > |t|
Temp 1	1	2	−1.1140	0.6645	20	−1.68	0.1092
Temp 1	1	3	−2.0900	0.6645	20	−3.15	0.0051
Temp 1	2	3	−0.9760	0.6645	20	−1.47	0.1575
Temp 2	1	2	0.2760	0.6645	20	0.42	0.6823
Temp 2	1	3	0.5060	0.6645	20	0.76	0.4553
Temp 2	2	3	0.2300	0.6645	20	0.35	0.7329

The slope differences are evident by the change in sign for comparisons within temperature 1 and within temperature 2. There is a significant effect of pressure at temperature 1 ($p = 0.0179$), but not at temperature 2 ($p = 0.7508$).

Pairwise Difference Plot (Diffogram)

Graphical displays of LS-means-related analyses consist of plots of all pairwise differences (DiffPlot), plots of differences against a control level (ControlPlot), and plots of differences against an overall average (AnomPlot). The following data set is from an experiment to investigate how snapdragons grow in various soils (Stenstrom 1940). To eliminate the effect of local fertility variations, the experiment is run in blocks, with each soil type sampled in each block. See the "Examples" section of Chapter 39, "The GLM Procedure," for an in-depth analysis of these data.

```
data plants;
  input Type $ @;
  do Block = 1 to 3;
     input StemLength @;
     output;
  end;
  datalines;
  Clarion   32.7 32.3 31.5
  Clinton   32.1 29.7 29.1
  Knox      35.7 35.9 33.1
  ONeill    36.0 34.2 31.2
  Compost   31.8 28.0 29.2
  Wabash    38.2 37.8 31.9
  Webster   32.5 31.1 29.7
;
```

The following statements perform the analysis of the experiment with the GLIMMIX procedure:

```
ods graphics on;
ods select LSMeans DiffPlot;

proc glimmix data=plants order=data plots=Diffogram;
   class Block Type;
   model StemLength = Block Type;
   lsmeans Type;
run;

ods graphics off;
```

The PLOTS= option in the PROC GLIMMIX statement requests that plots of pairwise least squares means differences are produced for effects that are listed in corresponding LSMEANS statements. This is the Type effect.

The Type LS-means are shown in Figure 38.27. Note that the order in which the levels appear corresponds to the order in which they were read from the data set. This was accomplished with the ORDER=DATA option in the PROC GLIMMIX statement.

Figure 38.27 Least Squares Means for Type Effect

The GLIMMIX Procedure

Type Least Squares Means

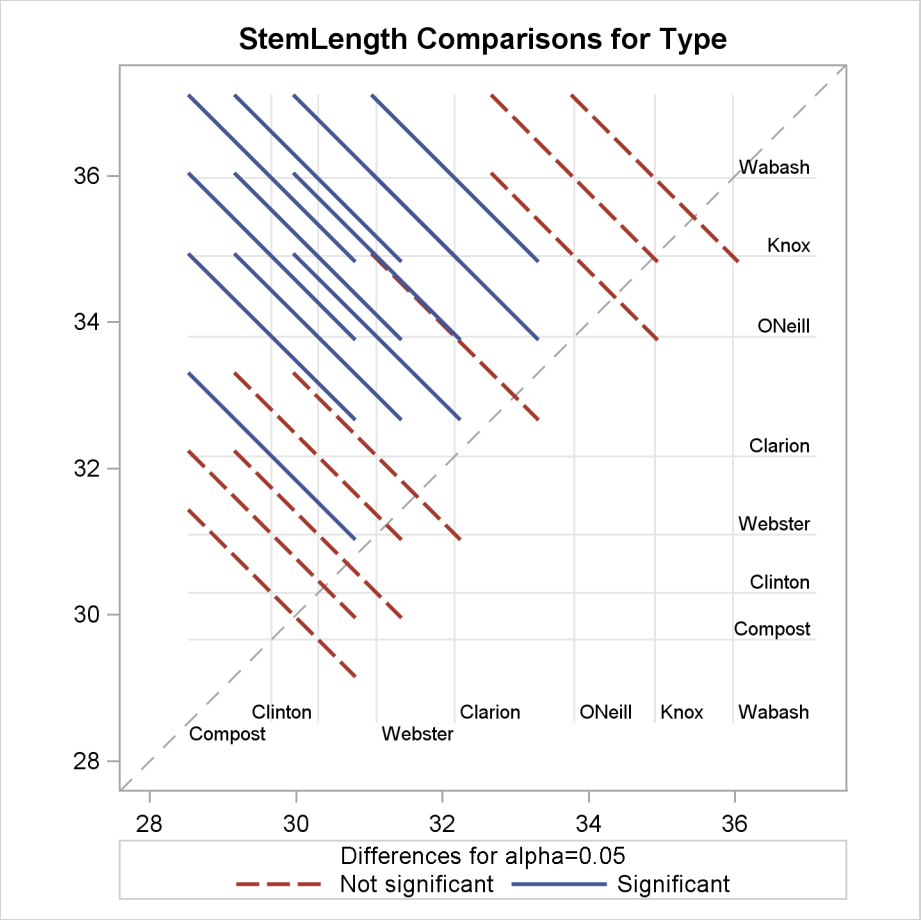
Type	Estimate	Standard Error	DF	t Value	Pr > |t|
Clarion	32.1667	0.7405	12	43.44	<.0001
Clinton	30.3000	0.7405	12	40.92	<.0001
Knox	34.9000	0.7405	12	47.13	<.0001
ONeill	33.8000	0.7405	12	45.64	<.0001
Compost	29.6667	0.7405	12	40.06	<.0001
Wabash	35.9667	0.7405	12	48.57	<.0001
Webster	31.1000	0.7405	12	42.00	<.0001

Because there are seven levels of Type in this analysis, there are $7(6-1)/2 = 21$ pairwise comparisons among the least squares means. The comparisons are performed in the following fashion: the first level of Type is compared against levels 2 through 7; the second level of Type is compared against levels 3 through 7; and so forth.

The default difference plot for these data is shown in Figure 38.28. The display is also known as a "mean-mean scatter plot" (Hsu 1996; Hsu and Peruggia 1994). It contains 21 lines rotated by 45 degrees counterclockwise, and a reference line (dashed 45-degree line). The (x, y) coordinate for the center of each line corresponds to the two least squares means being compared. Suppose that $\widehat{\eta}_{.i}$ and $\widehat{\eta}_{.j}$ denote the ith and jth least squares mean, respectively, for the effect in question, where $i < j$ according to the ordering of the effect levels. If the ABS option is in effect, which is the default, the line segment is centered at $(\min\{\widehat{\eta}_{i.}, \widehat{\eta}_{j.}\}, \max\{\widehat{\eta}_{i.}, \widehat{\eta}_{j.}\})$. Take, for example, the comparison of "Clarion" and "Compost" types. The respective estimates of their LS-means are $\widehat{\eta}_{.1} = 32.1667$ and $\widehat{\eta}_{.5} = 29.6667$. The center of the line segment for $H_0: \eta_{.1} = \eta_{.5}$ is placed at $(29.6667, 32.1667)$.

The length of the line segment for the comparison between means i and j corresponds to the width of the confidence interval for the difference $\eta_{.i} - \eta_{.j}$. This length is adjusted for the rotation in the plot. As a consequence, comparisons whose confidence interval covers zero cross the 45-degree reference line. These are the nonsignificant comparisons. Lines associated with significant comparisons do not touch or cross the reference line. Because these data are balanced, the estimated standard errors of all pairwise comparisons are identical, and the widths of the line segments are the same.

Figure 38.28 LS-Means Plot of Pairwise Differences



The background grid of the difference plot is drawn at the values of the least squares means for the seven type levels. These grid lines are used to find a particular comparison by intersection. Also, the labels of the grid lines indicate the ordering of the least squares means.

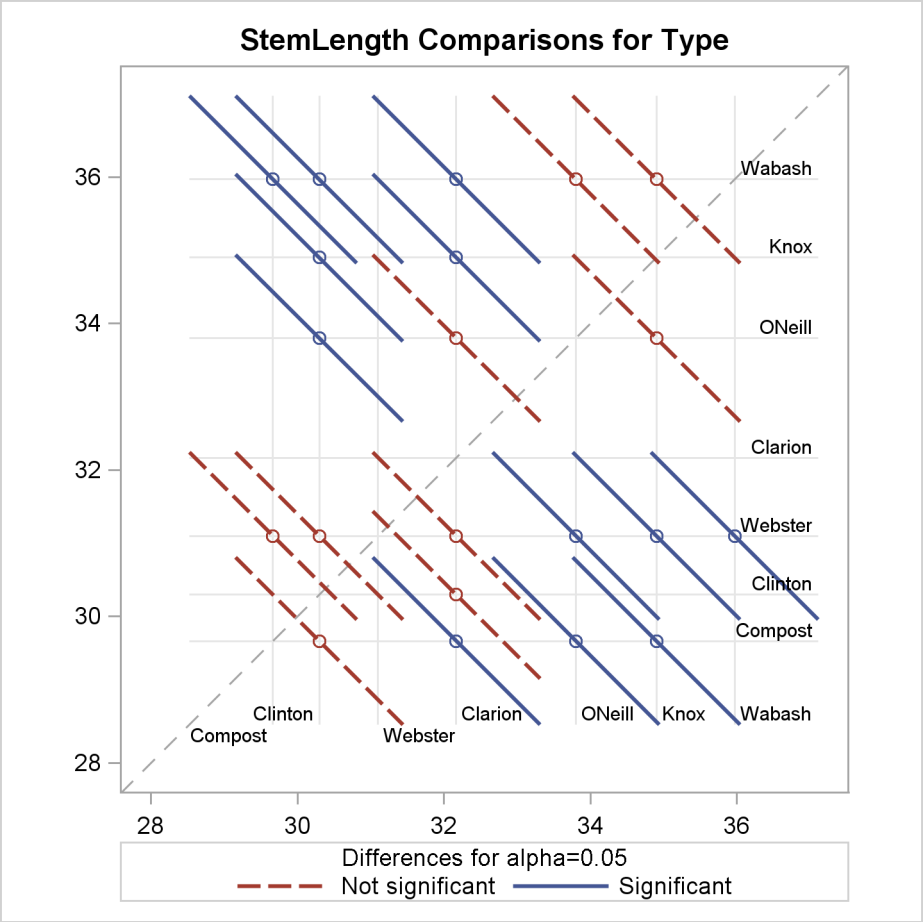
In the next set of statements, the NOABS and CENTER suboptions of the PLOTS=DIFFOGRAM option in the LSMEANS statement modify the appearance of the diffogram:

```
ods graphics on;
proc glimmix data=plants order=data;
   class Block Type;
   model StemLength = Block Type;
   lsmeans Type / plots=diffogram(noabs center);
run;
ods graphics off;
```

The NOABS suboption of the difference plot changes the way in which the GLIMMIX procedure places the line segments (Figure 38.29). If the NOABS suboption is in effect, the line segment is centered at the point $(\widehat{\eta}_{.i}, \widehat{\eta}_{.j})$, $i < j$. For example, the center of the line segment for a comparison of "Clarion" and "Compost" types is centered at $(\widehat{\eta}_{.1}, \widehat{\eta}_{.5}) = (32.1667, 29.6667)$. Whether a line segment appears above or below the reference line depends on the magnitude of the least squares means and the order of their appearance in the "Least Squares Means" table. The CENTER suboption places a marker at the intersection of the least squares means.

Because the ABS option places lines on the same side of the 45-degree reference, it can help to visually discover groups of significant and nonsignificant differences. On the other hand, when the number of levels in the effect is large, the display can get crowded. The NOABS option can then provide a more accessible resolution.

Figure 38.29 Diffogram with NOABS and CENTER Options



Least Squares Mean Control Plot

The following SAS statements create the same data set as before, except that one observation for Type="Knox" has been removed for illustrative purposes:

```
data plants;
  input Type $ @;
  do Block = 1 to 3;
     input StemLength @;
     output;
  end;
  datalines;
  Clarion   32.7 32.3 31.5
  Clinton   32.1 29.7 29.1
  Knox      35.7 35.9   .
  ONeill    36.0 34.2 31.2
  Compost   31.8 28.0 29.2
  Wabash    38.2 37.8 31.9
  Webster   32.5 31.1 29.7
;
```

The following statements request control plots for effects in LSMEANS statements with compatible option:

```
ods graphics on;
ods select Diffs ControlPlot;

proc glimmix data=plants order=data plots=ControlPlot;
   class Block Type;
   model StemLength = Block Type;
   lsmeans Type / diff=control('Clarion') adjust=dunnett;
run;

ods graphics off;
```

The LSMEANS statement for the Type effect is compatible; it requests comparisons of Type levels against "Clarion," adjusted for multiplicity with Dunnett's method. Because "Clarion" is the first level of the effect, the LSMEANS statement is equivalent to

```
lsmeans type / diff=control adjust=dunnett;
```

The "Differences of Type Least Squares Means" table in Figure 38.30 shows the six comparisons between Type levels and the control level.

Figure 38.30 Least Squares Means Differences

The GLIMMIX Procedure

Differences of Type Least Squares Means
Adjustment for Multiple Comparisons: Dunnett

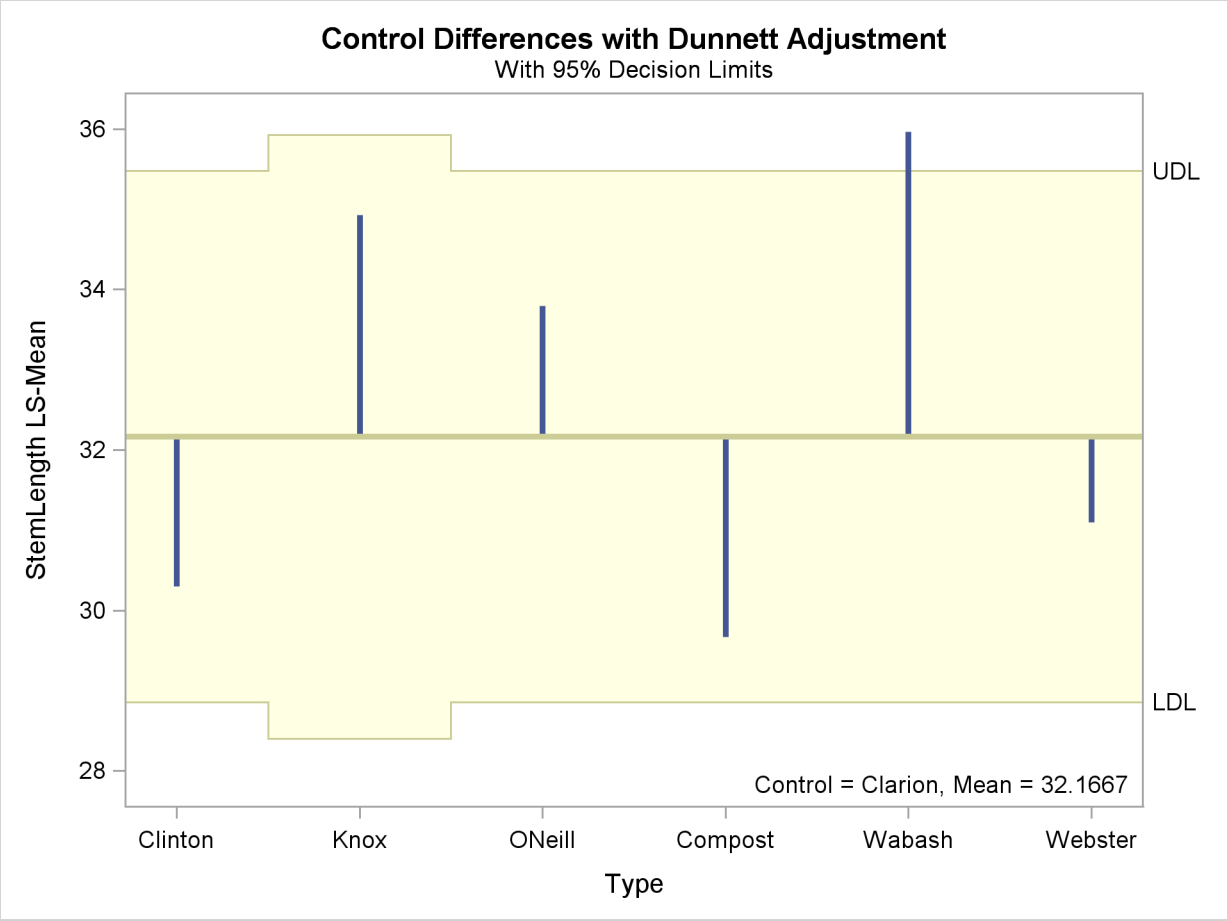
Type	_Type	Estimate	Standard Error	DF	t Value	Pr > |t|	Adj P
Clinton	Clarion	-1.8667	1.0937	11	-1.71	0.1159	0.3936
Knox	Clarion	2.7667	1.2430	11	2.23	0.0479	0.1854
ONeill	Clarion	1.6333	1.0937	11	1.49	0.1635	0.5144
Compost	Clarion	-2.5000	1.0937	11	-2.29	0.0431	0.1688
Wabash	Clarion	3.8000	1.0937	11	3.47	0.0052	0.0236
Webster	Clarion	-1.0667	1.0937	11	-0.98	0.3504	0.8359

The two rightmost columns of the table give the unadjusted and multiplicity-adjusted p-values. At the 5% significance level, both "Knox" and "Wabash" differ significantly from "Clarion" according to the unadjusted tests. After adjusting for multiplicity, only "Wabash" has a least squares mean significantly different from the control mean. Note that the standard error for the comparison involving "Knox" is larger than that for other comparisons because of the reduced sample size for that soil type.

In the plot of control differences a horizontal line is drawn at the value of the "Clarion" least squares mean. Vertical lines emanating from this reference line terminate in the least squares means for the other levels (Figure 38.31).

The dashed upper and lower horizontal reference lines are the upper and lower decision limits for tests against the control level. If a vertical line crosses the upper or lower decision limit, the corresponding least squares mean is significantly different from the LS-mean in the control group. If the data had been balanced, the UDL and LDL would be straight lines, because all estimates $\widehat{\eta}_{.i} - \widehat{\eta}_{.j}$ would have had the same standard error. The limits for the comparison between "Knox" and "Clarion" are wider than for other comparisons, because of the reduced sample size for the "Knox" soil type.

Figure 38.31 LS-Means Plot of Differences against a Control



The significance level of the decision limits is determined from the ALPHA= level in the LSMEANS statement. The default are 95% limits. If you choose one-sided comparisons with DIFF=CONTROLL or DIFF=CONTROLU in the LSMEANS statement, only one of the decision limits is drawn.

Analysis of Means (ANOM) Plot

The analysis of means in PROC GLIMMIX compares least squares means not by contrasting them against each other as with all pairwise differences or control differences. Instead, the least squares means are compared against an average value. Consequently, there are k comparisons for a factor with k levels. The following statements request ANOM differences for the Type least squares means (Figure 38.32) and plots the differences (Figure 38.33):

```
ods graphics on;
ods select Diffs AnomPlot;
proc glimmix data=plants order=data plots=AnomPlot;
   class Block Type;
   model StemLength = Block Type;
   lsmeans Type / diff=anom;
run;
ods graphics off;
```

Figure 38.32 ANOM LS-Mean Differences

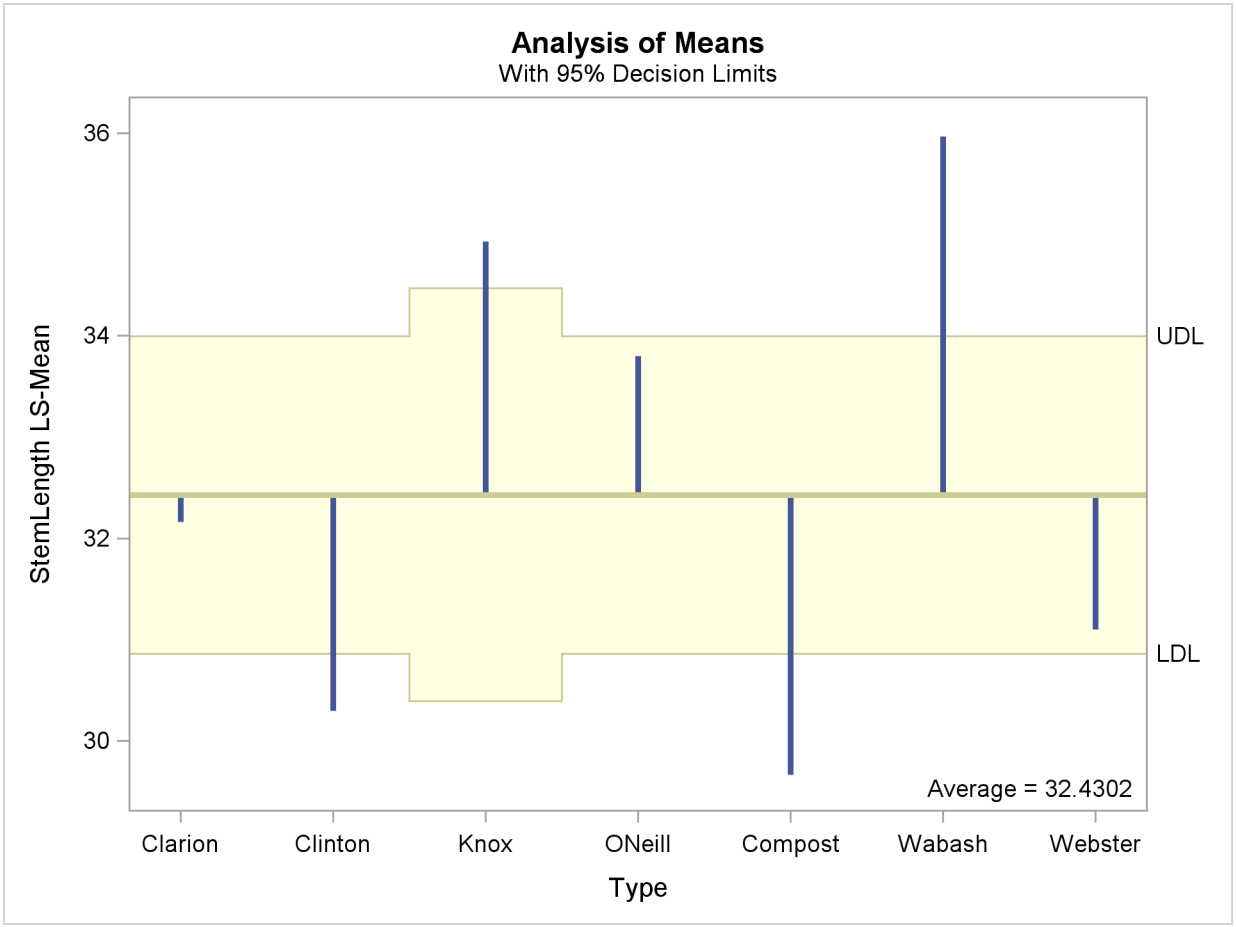
The GLIMMIX Procedure

Differences of Type Least Squares Means

Type	_Type	Estimate	Standard Error	DF	t Value	Pr > |t|
Clarion	Avg	−0.2635	0.7127	11	−0.37	0.7186
Clinton	Avg	−2.1302	0.7127	11	−2.99	0.0123
Knox	Avg	2.5032	0.9256	11	2.70	0.0205
ONeill	Avg	1.3698	0.7127	11	1.92	0.0809
Compost	Avg	−2.7635	0.7127	11	−3.88	0.0026
Wabash	Avg	3.5365	0.7127	11	4.96	0.0004
Webster	Avg	−1.3302	0.7127	11	−1.87	0.0888

At the 5% level, the "Clarion," "O'Neill," and "Webster" soil types are not significantly different from the average. Note that the artificial lack of balance introduced previously reduces the precision of the ANOM comparison for the "Knox" soil type.

Figure 38.33 LS-Means Analysis of Means (ANOM) Plot



The reference line in the ANOM plot is drawn at the average. Vertical lines extend from this reference line upward or downward, depending on the magnitude of the least squares means compared to the reference value. This enables you to quickly see which levels perform above and below the average. The horizontal reference lines are 95% upper and lower decision limits. If a vertical line crosses the limits, you conclude that the least squares mean is significantly different (at the 5% significance level) from the average. You can adjust the comparisons for multiplicity by adding the ADJUST=NELSON option in the LSMEANS statement.

ODS Graph Names

To request graphics with PROC GLIMMIX, you must first enable ODS Graphics by specifying the `ODS GRAPHICS ON` statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. The GLIMMIX procedure does not produce graphs by default. You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC GLIMMIX generates are listed in Table 38.20, along with the required statements and options.

Table 38.20 ODS Graphics Produced by PROC GLIMMIX

ODS Graph Name	Plot Description	Option
AnomPlot	Plot of LS-mean differences against the average LS-mean	PLOTS=ANOMPLOT LSMEANS / PLOTS=ANOMPLOT
Boxplot	Box plots of residuals and/or observed values for model effects	PLOTS=BOXPLOT
ControlPlot	Plot of LS-mean differences against a control level	PLOTS=CONTROLPLOT LSMEANS / PLOTS=CONTROLPLOT
DiffPlot	Plot of LS-mean pairwise differences	PLOTS=DIFFPLOT LSMEANS / PLOTS=DIFFPLOT
MeanPlot	Plot of least squares means	PLOTS=MEANPLOT LSMEANS / PLOTS=MEANPLOT
ORPlot	Plot of odds ratios	PLOTS=ODDSRATIO
PearsonBoxplot	Box plot of Pearson residuals	PLOTS=PEARSONPANEL(UNPACK)
PearsonByPredicted	Pearson residuals vs. mean	PLOTS=PEARSONPANEL(UNPACK)
PearsonHistogram	Histogram of Pearson residuals	PLOTS=PEARSONPANEL(UNPACK)
PearsonPanel	Panel of Pearson residuals	PLOTS=PEARSONPANEL
PearsonQQplot	*Q*-*Q* plot of Pearson residuals	PLOTS=PEARSONPANEL(UNPACK)
ResidualBoxplot	Box plot of (raw) residuals	PLOTS=RESIDUALPANEL(UNPACK)
ResidualByPredicted	Residuals vs. mean or linear predictor	PLOTS=RESIDUALPANEL(UNPACK)
ResidualHistogram	Histogram of (raw) residuals	PLOTS=RESIDUALPANEL(UNPACK)

Table 38.20 *continued*

ODS Graph Name	Plot Description	Option
ResidualPanel	Panel of (raw) residuals	PLOTS=RESIDUALPANEL
ResidualQQplot	*Q*-*Q* plot of (raw) residuals	PLOTS=RESIDUALPANEL(UNPACK)
StudentBoxplot	Box plot of studentized residuals	PLOTS=STUDENTPANEL(UNPACK)
StudentByPredicted	Studentized residuals vs. mean or linear predictor	PLOTS=STUDENTPANEL(UNPACK)
StudentHistogram	Histogram of studentized residuals	PLOTS=STUDENTPANEL(UNPACK)
StudentPanel	Panel of studentized residuals	PLOTS=STUDENTPANEL
StudentQQplot	*Q*-*Q* plot of studentized residuals	PLOTS=STUDENTPANEL(UNPACK)

Examples: GLIMMIX Procedure

Example 38.1: Binomial Counts in Randomized Blocks

In the context of spatial prediction in generalized linear models, Gotway and Stroup (1997) analyze data from an agronomic field trial. Researchers studied 16 varieties (entries) of wheat for their resistance to infestation by the Hessian fly. They arranged the varieties in a randomized complete block design on an 8×8 grid. Each 4×4 quadrant of that arrangement constitutes a block.

The outcome of interest was the number of damaged plants (Y_{ij}) out of the total number of plants growing on the unit (n_{ij}). The two subscripts identify the block ($i = 1, \cdots, 4$) and the entry ($j = 1, \cdots, 16$). The following SAS statements create the data set. The variables lat and lng denote the coordinate of an experimental unit on the 8×8 grid.

```
data HessianFly;
  label Y = 'No. of damaged plants'
        n = 'No. of plants';
  input block entry lat lng n Y @@;
  datalines;
  1 14 1 1  8 2    1 16 1 2  9 1
  1  7 1 3 13 9    1  6 1 4  9 9
  1 13 2 1  9 2    1 15 2 2 14 7
  1  8 2 3  8 6    1  5 2 4 11 8
  1 11 3 1 12 7    1 12 3 2 11 8
  1  2 3 3 10 8    1  3 3 4 12 5
  1 10 4 1  9 7    1  9 4 2 15 8
  1  4 4 3 19 6    1  1 4 4  8 7
```

```
  2 15 5 1 15 6    2  3 5 2 11 9
  2 10 5 3 12 5    2  2 5 4  9 9
  2 11 6 1 20 10   2  7 6 2 10 8
  2 14 6 3 12 4    2  6 6 4 10 7
  2  5 7 1  8 8    2 13 7 2  6 0
  2 12 7 3  9 2    2 16 7 4  9 0
  2  9 8 1 14 9    2  1 8 2 13 12
  2  8 8 3 12 3    2  4 8 4 14 7
  3  7 1 5  7 7    3 13 1 6  7 0
  3  8 1 7 13 3    3 14 1 8  9 0
  3  4 2 5 15 11   3 10 2 6  9 7
  3  3 2 7 15 11   3  9 2 8 13 5
  3  6 3 5 16 9    3  1 3 6  8 8
  3 15 3 7  7 0    3 12 3 8 12 8
  3 11 4 5  8 1    3 16 4 6 15 1
  3  5 4 7 12 7    3  2 4 8 16 12
  4  9 5 5 15 8    4  4 5 6 10 6
  4 12 5 7 13 5    4  1 5 8 15 9
  4 15 6 5 17 6    4  6 6 6  8 2
  4 14 6 7 12 5    4  7 6 8 15 8
  4 13 7 5 13 2    4  8 7 6 13 9
  4  3 7 7  9 9    4 10 7 8  6 6
  4  2 8 5 12 8    4 11 8 6  9 7
  4  5 8 7 11 10   4 16 8 8 15 7
;
```

Analysis as a GLM

If infestations are independent among experimental units, and all plants within a unit have the same propensity for infestation, then the Y_{ij} are binomial random variables. The first model considered is a standard generalized linear model for independent binomial counts:

```
proc glimmix data=HessianFly;
   class block entry;
   model y/n = block entry / solution;
run;
```

The PROC GLIMMIX statement invokes the procedure. The CLASS statement instructs the GLIMMIX procedure to treat both block and entry as classification variables. The MODEL statement specifies the response variable and the fixed effects in the model. PROC GLIMMIX constructs the **X** matrix of the model from the terms on the right side of the MODEL statement. The GLIMMIX procedure supports two kinds of syntax for the response variable. This example uses the *events/trials* syntax. The variable y represents the number of successes (*events*) out of n Bernoulli *trials*. When the *events/trials* syntax is used, the GLIMMIX procedure automatically selects the binomial distribution as the response distribution. Once the distribution is determined, the procedure selects the link function for the model. The default link for binomial data is the logit link. The preceding statements are thus equivalent to the following statements:

```
proc glimmix data=HessianFly;
   class block entry;
   model y/n = block entry / dist=binomial link=logit solution;
run;
```

The SOLUTION option in the MODEL statement requests that solutions for the fixed effects (parameter estimates) be displayed.

The "Model Information" table describes the model and methods used in fitting the statistical model (Output 38.1.1).

The GLIMMIX procedure recognizes that this is a model for uncorrelated data (variance matrix is diagonal) and that parameters can be estimated by maximum likelihood. The default degrees-of-freedom method to denominator degrees of freedom for F tests and t tests is the RESIDUAL method. This corresponds to choosing $f - \text{rank}(\mathbf{X})$ as the degrees of freedom, where f is the sum of the frequencies used in the analysis. You can change the degrees of freedom method with the DDFM= option in the MODEL statement.

Output 38.1.1 Model Information in GLM Analysis

The GLIMMIX Procedure

Model Information

Data Set	WORK.HESSIANFLY
Response Variable (Events)	Y
Response Variable (Trials)	n
Response Distribution	Binomial
Link Function	Logit
Variance Function	Default
Variance Matrix	Diagonal
Estimation Technique	Maximum Likelihood
Degrees of Freedom Method	Residual

The "Class Level Information" table lists the levels of the variables specified in the CLASS statement and the ordering of the levels (Output 38.1.2). The "Number of Observations" table displays the number of observations read and used in the analysis.

Output 38.1.2 Class Level Information and Number of Observations

Class Level Information

Class	Levels	Values
block	4	1 2 3 4
entry	16	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Number of Observations Read	64
Number of Observations Used	64
Number of Events	396
Number of Trials	736

The "Dimensions" table lists the size of relevant matrices (Output 38.1.3).

Output 38.1.3 Model Dimensions Information in GLM Analysis

Dimensions

Columns in X	21
Columns in Z	0
Subjects (Blocks in V)	1
Max Obs per Subject	64

Because of the absence of G-side random effects in this model, there are no columns in the $\mathbf{Z}$ matrix. The 21 columns in the $\mathbf{X}$ matrix comprise the intercept, 4 columns for the block effect and 16 columns for the entry effect. Because no RANDOM statement with a SUBJECT= option was specified, the GLIMMIX procedure does not process the data by subjects (see the section "Processing by Subjects" on page 2247 for details about subject processing).

The "Optimization Information" table provides information about the methods and size of the optimization problem (Output 38.1.4).

Output 38.1.4 Optimization Information in GLM Analysis

Optimization Information

Optimization Technique	Newton-Raphson
Parameters in Optimization	19
Lower Boundaries	0
Upper Boundaries	0
Fixed Effects	Not Profiled

With few exceptions, models fit with the GLIMMIX procedure require numerical methods for parameter estimation. The default optimization method for (overdispersed) GLM models is the Newton-Raphson algorithm. In this example, the optimization involves 19 parameters, corresponding to the number of linearly independent columns of the $\mathbf{X}'\mathbf{X}$ matrix.

The "Iteration History" table shows that the procedure converged after 3 iterations and 13 function evaluations (Output 38.1.5). The Change column measures the change in the objective function between iterations; however, this is not the monitored convergence criterion. The GLIMMIX procedure monitors several features simultaneously to determine whether to stop an optimization.

Output 38.1.5 Iteration History in GLM Analysis

Iteration History

Iteration	Restarts	Evaluations	Objective Function	Change	Max Gradient
0	0	4	134.13393738	.	4.899609
1	0	3	132.85058236	1.28335502	0.206204
2	0	3	132.84724263	0.00333973	0.000698
3	0	3	132.84724254	0.00000009	3.029E-8

Output 38.1.5 *continued*

```
Convergence criterion (GCONV=1E-8) satisfied.
```

The "Fit Statistics" table lists information about the fitted model (Output 38.1.6). The −2 Log Likelihood values are useful for comparing nested models, and the information criteria AIC, AICC, BIC, CAIC, and HQIC are useful for comparing nonnested models. On average, the ratio between the Pearson statistic and its degrees of freedom should equal one in GLMs. Values larger than one indicate overdispersion. With a ratio of 2.37, these data appear to exhibit more dispersion than expected under a binomial model with block and varietal effects.

Output 38.1.6 Fit Statistics in GLM Analysis

Fit Statistics

−2 Log Likelihood	265.69
AIC (smaller is better)	303.69
AICC (smaller is better)	320.97
BIC (smaller is better)	344.71
CAIC (smaller is better)	363.71
HQIC (smaller is better)	319.85
Pearson Chi-Square	106.74
Pearson Chi-Square / DF	2.37

The "Parameter Estimates" table displays the maximum likelihood estimates (Estimate), standard errors, and t tests for the hypothesis that the estimate is zero (Output 38.1.7).

Output 38.1.7 Parameter Estimates in GLM Analysis

Parameter Estimates

Effect	block	entry	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept			−1.2936	0.3908	45	−3.31	0.0018
block	1		−0.05776	0.2332	45	−0.25	0.8055
block	2		−0.1838	0.2303	45	−0.80	0.4289
block	3		−0.4420	0.2328	45	−1.90	0.0640
block	4		0	.	.	.	.
entry		1	2.9509	0.5397	45	5.47	<.0001
entry		2	2.8098	0.5158	45	5.45	<.0001
entry		3	2.4608	0.4956	45	4.97	<.0001
entry		4	1.5404	0.4564	45	3.38	0.0015
entry		5	2.7784	0.5293	45	5.25	<.0001
entry		6	2.0403	0.4889	45	4.17	0.0001
entry		7	2.3253	0.4966	45	4.68	<.0001
entry		8	1.3006	0.4754	45	2.74	0.0089
entry		9	1.5605	0.4569	45	3.42	0.0014
entry		10	2.3058	0.5203	45	4.43	<.0001
entry		11	1.4957	0.4710	45	3.18	0.0027
entry		12	1.5068	0.4767	45	3.16	0.0028
entry		13	−0.6296	0.6488	45	−0.97	0.3370
entry		14	0.4460	0.5126	45	0.87	0.3889
entry		15	0.8342	0.4698	45	1.78	0.0826
entry		16	0	.	.	.	.

The "Type III Tests of Fixed Effect" table displays significance tests for the two fixed effects in the model (Output 38.1.8).

Output 38.1.8 Type III Tests of Block and Entry Effects in GLM Analysis

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
block	3	45	1.42	0.2503
entry	15	45	6.96	<.0001

These tests are Wald-type tests, not likelihood ratio tests. The entry effect is clearly significant in this model with a p-value of <0.0001, indicating that the 16 wheat varieties are not equally susceptible to infestation by the Hessian fly.

Analysis with Random Block Effects

There are several possible reasons for the overdispersion noted in Output 38.1.6 (Pearson ratio = 2.37). The data might not follow a binomial distribution, one or more important effects might not have been accounted for in the model, or the data might be positively correlated. If important fixed

effects have been omitted, then you might need to consider adding them to the model. Because this is a designed experiment, it is reasonable not to expect further effects apart from the block and entry effects that represent the treatment and error control design structure. The reasons for the overdispersion must lie elsewhere.

If overdispersion stems from correlations among the observations, then the model should be appropriately adjusted. The correlation can have multiple sources. First, it might not be the case that the plants within an experimental unit responded independently. If the probability of infestation of a particular plant is altered by the infestation of a neighboring plant within the same unit, the infestation counts are not binomial and a different probability model should be used. A second possible source of correlations is the lack of independence of experimental units. Even if treatments were assigned to units at random, they might not respond independently. Shared spatial soil effects, for example, can be the underlying factor. The following analyses take these spatial effects into account.

First, assume that the environmental effects operate at the scale of the blocks. By making the block effects random, the marginal responses will be correlated due to the fact that observations within a block share the same random effects. Observations from different blocks will remain uncorrelated, in the spirit of separate randomizations among the blocks. The next set of statements fits a generalized linear mixed model (GLMM) with random block effects:

```
proc glimmix data=HessianFly;
   class block entry;
   model y/n = entry / solution;
   random block;
run;
```

Because the conditional distribution—conditional on the block effects—is binomial, the marginal distribution will be overdispersed relative to the binomial distribution. In contrast to adding a multiplicative scale parameter to the variance function, treating the block effects as random changes the estimates compared to a model with fixed block effects.

In the presence of random effects and a conditional binomial distribution, PROC GLIMMIX does not use maximum likelihood for estimation. Instead, the GLIMMIX procedure applies a restricted (residual) pseudo-likelihood algorithm (Output 38.1.9). The "restricted" attribute derives from the same rationale by which restricted (residual) maximum likelihood methods for linear mixed models attain their name; the likelihood equations are adjusted for the presence of fixed effects in the model to reduce bias in covariance parameter estimates.

Output 38.1.9 Model Information in GLMM Analysis

The GLIMMIX Procedure

Model Information

Data Set	WORK.HESSIANFLY
Response Variable (Events)	Y
Response Variable (Trials)	n
Response Distribution	Binomial
Link Function	Logit
Variance Function	Default
Variance Matrix	Not blocked
Estimation Technique	Residual PL
Degrees of Freedom Method	Containment

The "Class Level Information" and "Number of Observations" tables are as before (Output 38.1.10).

Output 38.1.10 Class Level Information and Number of Observations

Class Level Information

Class	Levels	Values
block	4	1 2 3 4
entry	16	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16

Number of Observations Read	64
Number of Observations Used	64
Number of Events	396
Number of Trials	736

The "Dimensions" table indicates that there is a single G-side parameter, the variance of the random block effect (Output 38.1.11). The "Dimensions" table has changed from the previous model (compare Output 38.1.11 to Output 38.1.3). Note that although the block effect has four levels, only a single variance component is estimated. The $\mathbf{Z}$ matrix has four columns, however, corresponding to the four levels of the block effect. Because no SUBJECT= option is used in the RANDOM statement, the GLIMMIX procedure treats these data as having arisen from a single subject with 64 observations.

Output 38.1.11 Model Dimensions Information in GLMM Analysis

Dimensions

G-side Cov. Parameters	1
Columns in X	17
Columns in Z	4
Subjects (Blocks in V)	1
Max Obs per Subject	64

The "Optimization Information" table indicates that a quasi-Newton method is used to solve the optimization problem. This is the default optimization method for GLMM models (Output 38.1.12).

Output 38.1.12 Optimization Information in GLMM Analysis

Optimization Information

Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	1
Lower Boundaries	1
Upper Boundaries	0
Fixed Effects	Profiled
Starting From	Data

In contrast to the Newton-Raphson method, the quasi-Newton method does not require second derivatives. Because the covariance parameters are not unbounded in this example, the procedure enforces a lower boundary constraint (zero) for the variance of the block effect, and the optimization method is changed to a dual quasi-Newton method. The fixed effects are profiled from the likelihood equations in this model. The resulting optimization problem involves only the covariance parameters.

The "Iteration History" table appears to indicate that the procedure converged after four iterations (Output 38.1.13). Notice, however, that this table has changed slightly from the previous analysis (see Output 38.1.5). The Evaluations column has been replaced by the Subiterations column, because the GLIMMIX procedure applied a doubly iterative fitting algorithm. The entire process consisted of five optimizations, each of which was iterative. The initial optimization required four iterations, the next one required three iterations, and so on.

Output 38.1.13 Iteration History in GLMM Analysis

Iteration History

Iteration	Restarts	Subiterations	Objective Function	Change	Max Gradient
0	0	4	173.28473428	0.81019251	0.000197
1	0	3	181.66726674	0.17550228	0.000739
2	0	2	182.20789493	0.00614874	7.018E-6
3	0	1	182.21315596	0.00004386	1.213E-8
4	0	0	182.21317662	0.00000000	3.349E-6

Convergence criterion (PCONV=1.11022E-8) satisfied.

The "Fit Statistics" table shows information about the fit of the GLMM (Output 38.1.14). The log likelihood reported in the table is not the residual log likelihood of the data. It is the residual log likelihood for an approximated model. The generalized chi-square statistic measures the residual sum of squares in the final model, and the ratio with its degrees of freedom is a measure of variability of the observation about the mean model.

Output 38.1.14 Fit Statistics in GLMM Analysis

Fit Statistics	
−2 Res Log Pseudo-Likelihood	182.21
Generalized Chi-Square	107.96
Gener. Chi-Square / DF	2.25

The variance of the random block effects is rather small (Output 38.1.15).

Output 38.1.15 Estimated Covariance Parameters and Approximate Standard Errors

Covariance Parameter Estimates

Cov Parm	Estimate	Standard Error
block	0.01116	0.03116

If the environmental effects operate on a spatial scale smaller than the block size, the random block model does not provide a suitable adjustment. From the coarse layout of the experimental area, it is not surprising that random block effects alone do not account for the overdispersion in the data. Adding a random component to a generalized linear model is different from adding a multiplicative overdispersion component, for example, via the PSCALE option in PROC GENMOD or a

```
random _residual_;
```

statement in PROC GLIMMIX. Such overdispersion components do not affect the parameter estimates, only their standard errors. A genuine random effect, on the other hand, affects both the parameter estimates and their standard errors (compare Output 38.1.16 to Output 38.1.7).

Output 38.1.16 Parameter Estimates for Fixed Effects in GLMM Analysis

Solutions for Fixed Effects

Effect	entry	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept		−1.4637	0.3738	3	−3.92	0.0296
entry	1	2.9609	0.5384	45	5.50	<.0001
entry	2	2.7807	0.5138	45	5.41	<.0001
entry	3	2.4339	0.4934	45	4.93	<.0001
entry	4	1.5347	0.4542	45	3.38	0.0015
entry	5	2.7653	0.5276	45	5.24	<.0001
entry	6	2.0014	0.4865	45	4.11	0.0002
entry	7	2.3518	0.4952	45	4.75	<.0001
entry	8	1.2927	0.4739	45	2.73	0.0091
entry	9	1.5663	0.4554	45	3.44	0.0013
entry	10	2.2896	0.5179	45	4.42	<.0001
entry	11	1.5018	0.4682	45	3.21	0.0025
entry	12	1.5075	0.4752	45	3.17	0.0027
entry	13	−0.5955	0.6475	45	−0.92	0.3626
entry	14	0.4573	0.5111	45	0.89	0.3758
entry	15	0.8683	0.4682	45	1.85	0.0702
entry	16	0	.	.	.	.

Output 38.1.17 Type III Test of Entry in GLMM Analysis

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
entry	15	45	6.90	<.0001

Because the block variance component is small, the Type III test for the variety effect in Output 38.1.17 is affected only very little compared to the GLM (Output 38.1.8).

Analysis with Smooth Spatial Trends

You can also consider these data in an observational sense, where the covariation of the observations is subject to modeling. Rather than deriving model components from the experimental design alone, environmental effects can be modeled by adjusting the mean and/or correlation structure. Gotway and Stroup (1997) and Schabenberger and Pierce (2002) supplant the coarse block effects with smooth-scale spatial components.

The model considered by Gotway and Stroup (1997) is a marginal model in that the correlation structure is modeled through residual-side (R-side) random components. This exponential covariance model is fit with the following statements:

```
proc glimmix data=HessianFly;
   class entry;
   model y/n = entry / solution ddfm=contain;
   random _residual_ / subject=intercept type=sp(exp)(lng lat);
run;
```

Note that the block effects have been removed from the statements. The keyword _RESIDUAL_ in the RANDOM statement instructs the GLIMMIX procedure to model the **R** matrix. Here, **R** is to be modeled as an exponential covariance structure matrix. The SUBJECT=INTERCEPT option means that all observations are considered correlated. Because the random effects are residual-type (R-side) effects, there are no columns in the **Z** matrix for this model (Output 38.1.18).

Output 38.1.18 Model Dimension Information in Marginal Spatial Analysis

The GLIMMIX Procedure

Dimensions

R-side Cov. Parameters	2
Columns in X	17
Columns in Z per Subject	0
Subjects (Blocks in V)	1
Max Obs per Subject	64

In addition to the fixed effects, the GLIMMIX procedure now profiles one of the covariance parameters, the variance of the exponential covariance model (Output 38.1.19). This reduces the size of the optimization problem. Only a single parameter is part of the optimization, the "range" (SP(EXP)) of the spatial process.

Output 38.1.19 Optimization Information in Spatial Analysis

Optimization Information

Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	1
Lower Boundaries	1
Upper Boundaries	0
Fixed Effects	Profiled
Residual Variance	Profiled
Starting From	Data

The practical range of a spatial process is that distance at which the correlation between data points has decreased to at most 0.05. The parameter reported by the GLIMMIX procedure as SP(EXP) in Output 38.1.20 corresponds to one-third of the practical range. The practical range in this process is $3 \times 0.9052 = 2.7156$. Correlations extend beyond a single experimental unit, but they do not appear to exist on the scale of the block size.

Output 38.1.20 Estimates of Covariance Parameters

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error
SP(EXP)	Intercept	0.9052	0.4404
Residual		2.5315	0.6974

The sill of the spatial process, the variance of the underlying residual effect, is estimated as 2.5315.

Output 38.1.21 Type III Test of Entry Effect in Spatial Analysis

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
entry	15	48	3.60	0.0004

The F value for the entry effect has been sharply reduced compared to the previous analyses. The smooth spatial variation accounts for some of the variation among the varieties (Output 38.1.21).

In this example three models were considered for the analysis of a randomized block design with binomial outcomes. If data are correlated, a standard generalized linear model often will indicate overdispersion relative to the binomial distribution. Two courses of action are considered in this example to address this overdispersion. First, the inclusion of G-side random effects models the correlation indirectly; it is induced through the sharing of random effects among responses from the same block. Second, the R-side spatial covariance structure models covariation directly. In generalized linear (mixed) models these two modeling approaches can lead to different inferences, because the models have different interpretation. The random block effects are modeled on the linked (logit) scale, and the spatial effects were modeled on the mean scale. Only in a linear mixed model are the two scales identical.

Example 38.2: Mating Experiment with Crossed Random Effects

McCullagh and Nelder (1989, Ch. 14.5) describe a mating experiment—conducted by S. Arnold and P. Verell at the University of Chicago, Department of Ecology and Evolution—involving two geographically isolated populations of mountain dusky salamanders. One goal of the experiment was to determine whether barriers to interbreeding have evolved in light of the geographical isolation of the populations. In this case, matings within a population should be more successful than matings between the populations. The experiment conducted in the summer of 1986 involved 40 animals, 20 rough butt (R) and 20 whiteside (W) salamanders, with equal numbers of males and females. The animals were grouped into two sets of R males, two sets of R females, two sets of W males, and two sets of W females, so that each set comprised five salamanders. Each set was

mated against one rough butt and one whiteside set, creating eight crossings. Within the pairings of sets, each female was paired to three male animals. The salamander mating data have been used by a number of authors; see, for example, McCullagh and Nelder (1989), Schall (1991), Karim and Zeger (1992), Breslow and Clayton (1993), Wolfinger and O'Connell (1993), and Shun (1997).

The following DATA step creates the data set for the analysis.

```
data salamander;
  input day fpop$ fnum mpop$ mnum mating @@;
  datalines;
4  rb  1 rb  1 1  4  rb  2 rb  5 1
4  rb  3 rb  2 1  4  rb  4 rb  4 1
4  rb  5 rb  3 1  4  rb  6 ws  9 1
4  rb  7 ws  8 0  4  rb  8 ws  6 0
4  rb  9 ws 10 0  4  rb 10 ws  7 0
4  ws  1 rb  9 0  4  ws  2 rb  7 0
4  ws  3 rb  8 0  4  ws  4 rb 10 0
4  ws  5 rb  6 0  4  ws  6 ws  5 0
4  ws  7 ws  4 1  4  ws  8 ws  1 1
4  ws  9 ws  3 1  4  ws 10 ws  2 1
8  rb  1 ws  4 1  8  rb  2 ws  5 1
8  rb  3 ws  1 0  8  rb  4 ws  2 1
8  rb  5 ws  3 1  8  rb  6 rb  9 1
8  rb  7 rb  8 0  8  rb  8 rb  6 1
8  rb  9 rb  7 0  8  rb 10 rb 10 0
8  ws  1 ws  9 1  8  ws  2 ws  6 0
8  ws  3 ws  7 0  8  ws  4 ws 10 1
8  ws  5 ws  8 1  8  ws  6 rb  2 0
8  ws  7 rb  1 1  8  ws  8 rb  4 0
8  ws  9 rb  3 1  8  ws 10 rb  5 0
12 rb  1 rb  5 1  12 rb  2 rb  3 1
12 rb  3 rb  1 1  12 rb  4 rb  2 1
12 rb  5 rb  4 1  12 rb  6 ws 10 1
12 rb  7 ws  9 0  12 rb  8 ws  7 0
12 rb  9 ws  8 1  12 rb 10 ws  6 1
12 ws  1 rb  7 1  12 ws  2 rb  9 0
12 ws  3 rb  6 0  12 ws  4 rb  8 1
12 ws  5 rb 10 0  12 ws  6 ws  3 1
12 ws  7 ws  5 1  12 ws  8 ws  2 1
12 ws  9 ws  1 1  12 ws 10 ws  4 0
16 rb  1 ws  1 0  16 rb  2 ws  3 1
16 rb  3 ws  4 1  16 rb  4 ws  5 0
16 rb  5 ws  2 1  16 rb  6 rb  7 0
16 rb  7 rb  9 1  16 rb  8 rb 10 0
16 rb  9 rb  6 1  16 rb 10 rb  8 0
16 ws  1 ws 10 1  16 ws  2 ws  7 1
16 ws  3 ws  9 0  16 ws  4 ws  8 1
16 ws  5 ws  6 0  16 ws  6 rb  4 0
16 ws  7 rb  2 0  16 ws  8 rb  5 0
16 ws  9 rb  1 1  16 ws 10 rb  3 1
20 rb  1 rb  4 1  20 rb  2 rb  1 1
20 rb  3 rb  3 1  20 rb  4 rb  5 1
20 rb  5 rb  2 1  20 rb  6 ws  6 1
```

```
20 rb  7 ws  7 0  20 rb  8 ws 10 1
20 rb  9 ws  9 1  20 rb 10 ws  8 1
20 ws  1 rb 10 0  20 ws  2 rb  6 0
20 ws  3 rb  7 0  20 ws  4 rb  9 0
20 ws  5 rb  8 0  20 ws  6 ws  2 0
20 ws  7 ws  1 1  20 ws  8 ws  5 1
20 ws  9 ws  4 1  20 ws 10 ws  3 1
24 rb  1 ws  5 1  24 rb  2 ws  2 1
24 rb  3 ws  3 1  24 rb  4 ws  4 1
24 rb  5 ws  1 1  24 rb  6 rb  8 1
24 rb  7 rb  6 0  24 rb  8 rb  9 1
24 rb  9 rb 10 1  24 rb 10 rb  7 0
24 ws  1 ws  8 1  24 ws  2 ws 10 0
24 ws  3 ws  6 1  24 ws  4 ws  9 1
24 ws  5 ws  7 0  24 ws  6 rb  1 0
24 ws  7 rb  5 1  24 ws  8 rb  3 0
24 ws  9 rb  4 0  24 ws 10 rb  2 0
;
```

The first observation, for example, indicates that rough butt female 1 was paired in the laboratory on day 4 of the experiment with rough butt male 1, and the pair mated. On the same day rough butt female 7 was paired with whiteside male 8, but the pairing did not result in mating of the animals.

The model adopted by many authors for these data comprises fixed effects for gender and population, their interaction, and male and female random effects. Specifically, let π_{RR}, π_{RW}, π_{WR}, and π_{WW} denote the mating probabilities between the populations, where the first subscript identifies the female partner of the pair. Then, we model

$$\log\left\{\frac{\pi_{kl}}{1-\pi_{kl}}\right\} = \tau_{kl} + \gamma_f + \gamma_m \quad k,l \in \{R,W\}$$

where γ_f and γ_m are independent random variables representing female and male random effects (20 each), and τ_{kl} denotes the average logit of mating between females of population k and males of population l.

The following statements fit this model by pseudo-likelihood:

```
proc glimmix data=salamander;
   class fpop fnum mpop mnum;
   model mating(event='1') = fpop|mpop / dist=binary;
   random fpop*fnum mpop*mnum;
   lsmeans fpop*mpop / ilink;
run;
```

The response variable is the two-level variable mating. Because it is coded as zeros and ones, and because PROC GLIMMIX models by default the probability of the first level according to the response-level ordering, the EVENT='1' option instructs PROC GLIMMIX to model the probability of a successful mating. The distribution of the mating variable, conditional on the random effects, is binary.

The fpop*fnum effect in the RANDOM statement creates a random intercept for each female animal. Because fpop and fnum are CLASS variables, the effect has 20 levels (10 rb and 10 ws females). Similarly, the mpop*mnum effect creates the random intercepts for the male animals. Because no

TYPE= is specified in the RANDOM statement, the covariance structure defaults to TYPE=VC. The random effects and their levels are independent, and each effect has its own variance component. Because the conditional distribution of the data, conditioned on the random effects, is binary, no extra scale parameter (ϕ) is added.

The LSMEANS statement requests least squares means for the four levels of the fpop*mpop effect, which are estimates of the cell means in the 2×2 classification of female and male populations. The ILINK option in the LSMEANS statement requests that the estimated means and standard errors are also reported on the scale of the data. This yields estimates of the four mating probabilities, π_{RR}, π_{RW}, π_{WR}, and π_{WW}.

The "Model Information" table displays general information about the model being fit (Output 38.2.1).

Output 38.2.1 Analysis of Mating Experiment with Crossed Random Effects

The GLIMMIX Procedure

Model Information

Data Set	WORK.SALAMANDER
Response Variable	mating
Response Distribution	Binary
Link Function	Logit
Variance Function	Default
Variance Matrix	Not blocked
Estimation Technique	Residual PL
Degrees of Freedom Method	Containment

The response variable mating follows a binary distribution (conditional on the random effects). Hence, the mean of the data is an event probability, π, and the logit of this probability is linearly related to the linear predictor of the model. The variance function is the default function that is implied by the distribution, $a(\pi) = \pi(1 - \pi)$. The variance matrix is not blocked, because the GLIMMIX procedure does not process the data by subjects (see the section "Processing by Subjects" on page 2247 for details). The estimation technique is the default method for GLMMs, residual pseudo-likelihood (METHOD=RSPL), and degrees of freedom for tests and confidence intervals are determined by the containment method.

The "Class Level Information" table in Output 38.2.2 lists the levels of the variables listed in the CLASS statement, as well as the order of the levels.

Output 38.2.2 Class Level Information and Number of Observations

Class Level Information

Class	Levels	Values
fpop	2	rb ws
fnum	10	1 2 3 4 5 6 7 8 9 10
mpop	2	rb ws
mnum	10	1 2 3 4 5 6 7 8 9 10

Output 38.2.2 *continued*

Number of Observations Read	120
Number of Observations Used	120

Note that there are two female populations and two male populations; also, the variables fnum and mnum have 10 levels each. As a consequence, the effects fpop*fnum and mpop*mnum identify the 20 females and males, respectively. The effect fpop*mpop identifies the four mating types.

The "Response Profile Table," which is displayed for binary or multinomial data, lists the levels of the response variable and their order (Output 38.2.3). With binary data, the table also provides information about which level of the response variable defines the event. Because of the EVENT='1' response variable option in the MODEL statement, the probability being modeled is that of the higher-ordered value.

Output 38.2.3 Response Profiles

Response Profile

Ordered Value	mating	Total Frequency
1	0	50
2	1	70

The GLIMMIX procedure is modeling the probability that mating='1'.

There are two covariance parameters in this model, the variance of the fpop*fnum effect and the variance of the mpop*mnum effect (Output 38.2.4). Both parameters are modeled as G-side parameters. The nine columns in the **X** matrix comprise the intercept, two columns each for the levels of the fpop and mpop effects, and four columns for their interaction. The **Z** matrix has 40 columns, one for each animal. Because the data are not processed by subjects, PROC GLIMMIX assumes the data consist of a single subject (a single block in **V**).

Output 38.2.4 Model Dimensions Information

Dimensions

G-side Cov. Parameters	2
Columns in X	9
Columns in Z	40
Subjects (Blocks in V)	1
Max Obs per Subject	120

The "Optimization Information" table displays basic information about the optimization (Output 38.2.5). The default technique for GLMMs is the quasi-Newton method. There are two parameters in the optimization, which correspond to the two variance components. The 17 fixed effects parameters are not part of the optimization. The initial optimization computes pseudo-

data based on the response values in the data set rather than from estimates of a generalized linear model fit.

Output 38.2.5 Optimization Information

Optimization Information

Optimization Technique	Newton-Raphson with Ridging
Parameters in Optimization	2
Lower Boundaries	2
Upper Boundaries	0
Fixed Effects	Profiled
Starting From	Data

The GLIMMIX procedure performs eight optimizations after the initial optimization (Output 38.2.6). That is, following the initial pseudo-data creation, the pseudo-data were updated eight more times and a total of nine linear mixed models were estimated.

Output 38.2.6 Iteration History and Convergence Status

Iteration History

Iteration	Restarts	Subiterations	Objective Function	Change	Max Gradient
0	0	4	537.09173501	2.00000000	1.719E-8
1	0	3	544.12516903	0.66319780	1.14E-8
2	0	2	545.89139118	0.13539318	1.609E-6
3	0	2	546.10489538	0.01742065	5.89E-10
4	0	1	546.13075146	0.00212475	9.654E-7
5	0	1	546.13374731	0.00025072	1.346E-8
6	0	1	546.13409761	0.00002931	1.84E-10
7	0	0	546.13413861	0.00000000	4.285E-6

Convergence criterion (PCONV=1.11022E-8) satisfied.

The "Covariance Parameter Estimates" table lists the estimates for the two variance components and their estimated standard errors (Output 38.2.7). The heterogeneity (in the logit of the mating probabilities) among the females is considerably larger than the heterogeneity among the males.

Output 38.2.7 Estimated Covariance Parameters and Approximate Standard Errors

Covariance Parameter Estimates

Cov Parm	Estimate	Standard Error
fpop*fnum	1.4099	0.8871
mpop*mnum	0.08963	0.4102

The "Type III Tests of Fixed Effects" table indicates a significant interaction between the male and female populations (Output 38.2.8). A comparison in the logits of mating success in pairs with R females and W females depends on whether the male partner in the pair is the same species. The "fpop*mpop Least Squares Means" table shows this effect more clearly (Output 38.2.9).

Output 38.2.8 Tests of Main Effects and Interaction

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
fpop	1	18	2.86	0.1081
mpop	1	17	4.71	0.0444
fpop*mpop	1	81	9.61	0.0027

Output 38.2.9 Interaction Least Squares Means

fpop*mpop Least Squares Means

fpop	mpop	Estimate	Standard Error	DF	t Value	Pr > |t|	Mean	Standard Error Mean
rb	rb	1.1629	0.5961	81	1.95	0.0545	0.7619	0.1081
rb	ws	0.7839	0.5729	81	1.37	0.1750	0.6865	0.1233
ws	rb	-1.4119	0.6143	81	-2.30	0.0241	0.1959	0.09678
ws	ws	1.0151	0.5871	81	1.73	0.0876	0.7340	0.1146

In a pairing with a male rough butt salamander, the logit drops sharply from 1.1629 to -1.4119 when the male is paired with a whiteside female instead of a female from its own population. The corresponding estimated probabilities of mating success are $\widehat{\pi}_{RR} = 0.7619$ and $\widehat{\pi}_{WR} = 0.1959$. If the same comparisons are made in pairs with whiteside males, then you also notice a drop in the logit if the female comes from a different population, 1.0151 versus 0.7839. The change is considerably less, though, corresponding to mating probabilities of $\widehat{\pi}_{WW} = 0.7340$ and $\widehat{\pi}_{RW} = 0.6865$. Whiteside females appear to be successful with their own population. Whiteside males appear to succeed equally well with female partners of the two populations.

This insight into the factor-level comparisons can be amplified by graphing the least squares mean comparisons and by subsetting the differences of least squares means. This is accomplished with the following statements:

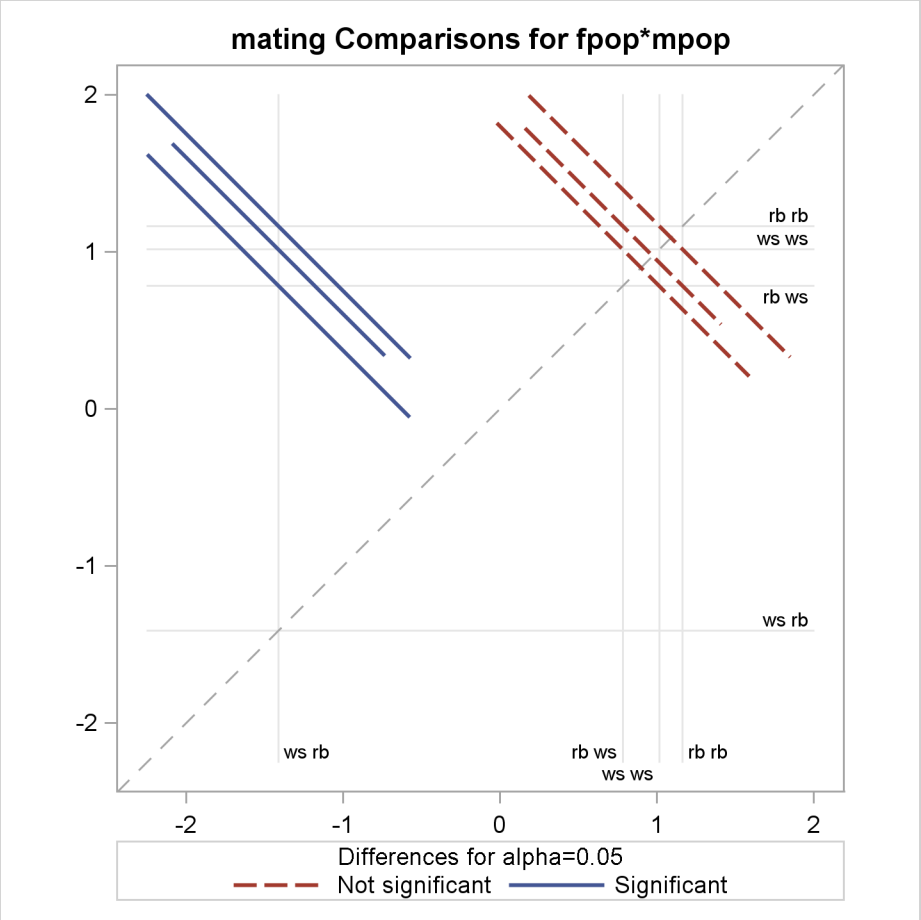
```
ods graphics on;
ods select DiffPlot SliceDiffs;
proc glimmix data=salamander;
   class fpop fnum mpop mnum;
   model mating(event='1') = fpop|mpop / dist=binary;
   random fpop*fnum mpop*mnum;
   lsmeans fpop*mpop / plots=diffplot;
   lsmeans fpop*mpop / slicediff=(mpop fpop);
run;
ods graphics off;
```

The PLOTS=DIFFPLOT option in the first LSMEANS statement requests a comparison plot that displays the result of all pairwise comparisons (Output 38.2.10). The SLICEDIFF=(mpop fpop) option requests differences of simple effects.

The comparison plot in Output 38.2.10 is also known as a mean-mean scatter plot (Hsu 1996). Each solid line in the plot corresponds to one of the possible $4 \times 3/2 = 6$ unique pairwise comparisons. The line is centered at the intersection of two least squares means, and the length of the line segments corresponds to the width of a 95% confidence interval for the difference between the two least squares means. The length of the segment is adjusted for the rotation. If a line segment crosses the dashed 45-degree line, the comparison between the two factor levels is not significant; otherwise, it is significant. The horizontal and vertical axes of the plot are drawn in least squares means units, and the grid lines are placed at the values of the least squares means.

The six pairs of least squares means comparisons separate into two sets of three pairs. Comparisons in the first set are significant; comparisons in the second set are not significant. For the significant set, the female partner in one of the pairs is a whiteside salamander. For the nonsignificant comparisons, the male partner in one of the pairs is a whiteside salamander.

Output 38.2.10 LS-Means Diffogram



The "Simple Effect Comparisons" tables show the results of the SLICEDIFF= option in the second LSMEANS statement (Output 38.2.11).

Output 38.2.11 Simple Effect Comparisons

Simple Effect Comparisons of fpop*mpop Least Squares Means By mpop

Simple Effect Level	fpop	_fpop	Estimate	Standard Error	DF	t Value	Pr > |t|
mpop rb	rb	ws	2.5748	0.8458	81	3.04	0.0031
mpop ws	rb	ws	−0.2312	0.8092	81	−0.29	0.7758

Simple Effect Comparisons of fpop*mpop Least Squares Means By fpop

Simple Effect Level	mpop	_mpop	Estimate	Standard Error	DF	t Value	Pr > |t|
fpop rb	rb	ws	0.3790	0.6268	81	0.60	0.5471
fpop ws	rb	ws	−2.4270	0.6793	81	−3.57	0.0006

The first table of simple effect comparisons holds fixed the level of the mpop factor and compares the levels of the fpop factor. Because there is only one possible comparison for each male population, there are two entries in the table. The first entry compares the logits of mating probabilities when the male partner is a rough butt, and the second entry applies when the male partner is from the whiteside population. The second table of simple effects comparisons applies the same logic, but holds fixed the level of the female partner in the pair. Note that these four comparisons are a subset of all six possible comparisons, eliminating those where both factors are varied at the same time. The simple effect comparisons show that there is no difference in mating probabilities if the male partner is a whiteside salamander, or if the female partner is a rough butt. Rough butt females also appear to mate indiscriminately.

Example 38.3: Smoothing Disease Rates; Standardized Mortality Ratios

Clayton and Kaldor (1987, Table 1) present data on observed and expected cases of lip cancer in the 56 counties of Scotland between 1975 and 1980. The expected number of cases was determined by a separate multiplicative model that accounted for the age distribution in the counties. The goal of the analysis is to estimate the county-specific log-relative risks, also known as standardized mortality ratios (SMR).

If Y_i is the number of incident cases in county i and E_i is the expected number of incident cases, then the ratio of observed to expected counts, Y_i/E_i, is the standardized mortality ratio. Clayton and Kaldor (1987) assume there exists a relative risk λ_i that is specific to each county and is a random variable. Conditional on λ_i, the observed counts are independent Poisson variables with mean $E_i\lambda_i$.

An elementary mixed model for λ_i specifies only a random intercept for each county, in addition to a fixed intercept. Breslow and Clayton (1993), in their analysis of these data, also provide a covariate that measures the percentage of employees in agriculture, fishing, and forestry. The expanded model

for the region-specific relative risk in Breslow and Clayton (1993) is

$$\lambda_i = \exp\{\beta_0 + \beta_1 x_i/10 + \gamma_i\}, \quad i = 1, \cdots, 56$$

where β_0 and β_1 are fixed effects, and the γ_i are county random effects.

The following DATA step creates the data set lipcancer. The expected number of cases is based on the observed standardized mortality ratio for counties with lip cancer cases, and based on the expected counts reported by Clayton and Kaldor (1987, Table 1) for the counties without cases. The sum of the expected counts then equals the sum of the observed counts.

```
data lipcancer;
   input county observed expected employment SMR;
   if (observed > 0) then expCount = 100*observed/SMR;
   else expCount = expected;
   datalines;
 1  9  1.4 16 652.2
 2 39  8.7 16 450.3
 3 11  3.0 10 361.8
 4  9  2.5 24 355.7
 5 15  4.3 10 352.1
 6  8  2.4 24 333.3
 7 26  8.1 10 320.6
 8  7  2.3  7 304.3
 9  6  2.0  7 303.0
10 20  6.6 16 301.7
11 13  4.4  7 295.5
12  5  1.8 16 279.3
13  3  1.1 10 277.8
14  8  3.3 24 241.7
15 17  7.8  7 216.8
16  9  4.6 16 197.8
17  2  1.1 10 186.9
18  7  4.2  7 167.5
19  9  5.5  7 162.7
20  7  4.4 10 157.7
21 16 10.5  7 153.0
22 31 22.7 16 136.7
23 11  8.8 10 125.4
24  7  5.6  7 124.6
25 19 15.5  1 122.8
26 15 12.5  1 120.1
27  7  6.0  7 115.9
28 10  9.0  7 111.6
29 16 14.4 10 111.3
30 11 10.2 10 107.8
31  5  4.8  7 105.3
32  3  2.9 24 104.2
33  7  7.0 10  99.6
34  8  8.5  7  93.8
35 11 12.3  7  89.3
36  9 10.1  0  89.1
37 11 12.7 10  86.8
38  8  9.4  1  85.6
39  6  7.2 16  83.3
```

```
40  4  5.3  0  75.9
41 10 18.8  1  53.3
42  8 15.8 16  50.7
43  2  4.3 16  46.3
44  6 14.6  0  41.0
45 19 50.7  1  37.5
46  3  8.2  7  36.6
47  2  5.6  1  35.8
48  3  9.3  1  32.1
49 28 88.7  0  31.6
50  6 19.6  1  30.6
51  1  3.4  1  29.1
52  1  3.6  0  27.6
53  1  5.7  1  17.4
54  1  7.0  1  14.2
55  0  4.2 16   0.0
56  0  1.8 10   0.0
;
```

Because the mean of the Poisson variates, conditional on the random effects, is $\mu_i = E_i \lambda_i$, applying a log link yields

$$\log\{\mu_i\} = \log\{E_i\} + \beta_0 + \beta_1 x_i/10 + \gamma_i$$

The term $\log\{E_i\}$ is an offset, a regressor variable whose coefficient is known to be one. Note that it is assumed that the E_i are known; they are not treated as random variables.

The following statements fit this model by residual pseudo-likelihood:

```
proc glimmix data=lipcancer;
   class county;
   x    = employment / 10;
   logn = log(expCount);
   model observed = x / dist=poisson offset=logn
                        solution ddfm=none;
   random county;
   SMR_pred = 100*exp(_zgamma_ + _xbeta_);
   id employment SMR SMR_pred;
   output out=glimmixout;
run;
```

The offset is created with the assignment statement

```
logn = log(expCount);
```

and is associated with the linear predictor through the OFFSET= option in the MODEL statement. The statement

```
x = employment / 10;
```

transforms the covariate measuring percentage of employment in agriculture, fisheries, and forestry to agree with the analysis of Breslow and Clayton (1993). The DDFM=NONE option in the MODEL statement requests chi-square tests and z tests instead of the default F tests and t tests by setting the denominator degrees of freedom in tests of fixed effects to ∞.

The statement

```
SMR_pred = 100*exp(_zgamma_ + _xbeta_);
```

calculates the fitted standardized mortality rate. Note that the offset variable does not contribute to the exponentiated term.

The OUTPUT statement saves results of the calculations to the output data set glimmixout. The ID statement specifies that only the listed variables are written to the output data set.

Output 38.3.1 Model Information in Poisson GLMM

```
                        The GLIMMIX Procedure

                          Model Information

          Data Set                      WORK.LIPCANCER
          Response Variable             observed
          Response Distribution         Poisson
          Link Function                 Log
          Variance Function             Default
          Offset Variable               logn = log(expCount);
          Variance Matrix               Not blocked
          Estimation Technique          Residual PL
          Degrees of Freedom Method     None

                       Class Level Information

Class     Levels    Values

county        56    1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
                    19 20 21 22 23 24 25 26 27 28 29 30 31 32 33
                    34 35 36 37 38 39 40 41 42 43 44 45 46 47 48
                    49 50 51 52 53 54 55 56

              Number of Observations Read          56
              Number of Observations Used          56

                           Dimensions

                 G-side Cov. Parameters         1
                 Columns in X                   2
                 Columns in Z                  56
                 Subjects (Blocks in V)         1
                 Max Obs per Subject           56
```

The GLIMMIX procedure displays in the "Model Information" table that the offset variable was computed with programming statements and the final assignment statement from your GLIMMIX statements (Output 38.3.1). There are two columns in the $\mathbf{X}$ matrix, corresponding to the intercept and the regressor $x/10$. There are 56 columns in the $\mathbf{Z}$ matrix, however, one for each observation in the data set (Output 38.3.1).

The optimization involves only a single covariance parameter, the variance of the county effect (Output 38.3.2). Because this parameter is a variance, the GLIMMIX procedure imposes a lower

boundary constraint; the solution for the variance is bounded by zero from below.

Output 38.3.2 Optimization Information in Poisson GLMM

Optimization Information

Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	1
Lower Boundaries	1
Upper Boundaries	0
Fixed Effects	Profiled
Starting From	Data

Following the initial creation of pseudo-data and the fit of a linear mixed model, the procedure goes through five more updates of the pseudo-data, each associated with a separate optimization (Output 38.3.3). Although the objective function in each optimization is the negative of twice the restricted maximum likelihood for that pseudo-data, there is no guarantee that across the outer iterations the objective function decreases in subsequent optimizations. In this example, minus twice the residual maximum likelihood at convergence takes on its smallest value at the initial optimization and increases in subsequent optimizations.

Output 38.3.3 Iteration History in Poisson GLMM

Iteration History

Iteration	Restarts	Subiterations	Objective Function	Change	Max Gradient
0	0	4	123.64113992	0.20997891	3.848E-8
1	0	3	127.05866018	0.03393332	0.000048
2	0	2	127.48839749	0.00223427	5.753E-6
3	0	1	127.50502469	0.00006946	1.938E-7
4	0	1	127.50528068	0.00000118	1.09E-7
5	0	0	127.50528481	0.00000000	1.299E-6

Convergence criterion (PCONV=1.11022E-8) satisfied.

The "Covariance Parameter Estimates" table in Output 38.3.4 shows the estimate of the variance of the region-specific log-relative risks. There is significant county-to-county heterogeneity in risks. If the covariate were removed from the analysis, as in Clayton and Kaldor (1987), the heterogeneity in county-specific risks would increase. (The fitted SMRs in Table 6 of Breslow and Clayton (1993) were obtained without the covariate x in the model.)

Output 38.3.4 Estimated Covariance Parameters in Poisson GLMM

Covariance Parameter Estimates

Cov Parm	Estimate	Standard Error
county	0.3567	0.09869

The "Solutions for Fixed Effects" table displays the estimates of β_0 and β_1 along with their standard errors and test statistics (Output 38.3.5). Because of the DDFM=NONE option in the MODEL statement, PROC GLIMMIX assumes that the degrees of freedom for the t tests of $H_0: \beta_j = 0$ are infinite. The p-values correspond to probabilities under a standard normal distribution. The covariate measuring employment percentages in agriculture, fisheries, and forestry is significant. This covariate might be a surrogate for the exposure to sunlight, an important risk factor for lip cancer.

Output 38.3.5 Fixed-Effects Parameter Estimates in Poisson GLMM

Solutions for Fixed Effects

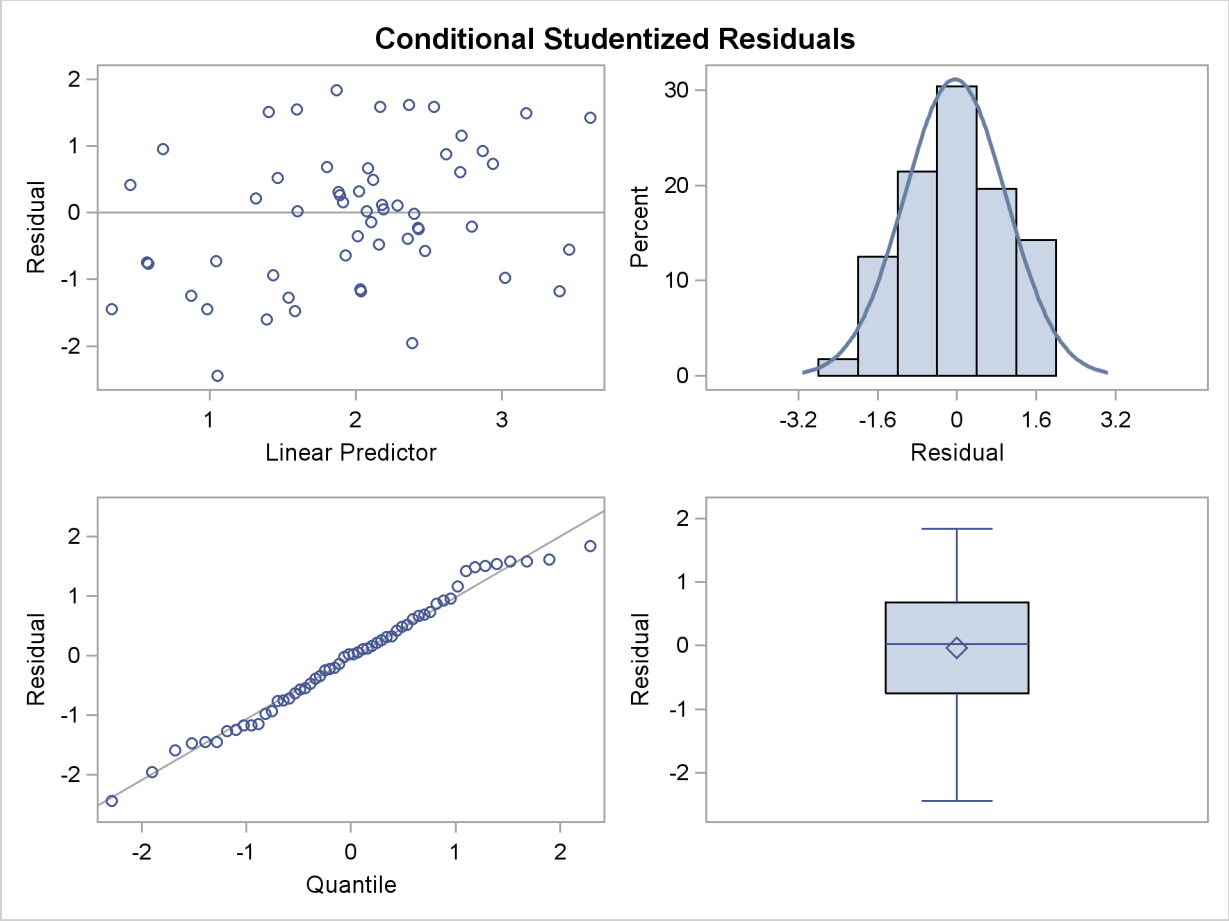
Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	-0.4406	0.1572	Infty	-2.80	0.0051
x	0.6799	0.1409	Infty	4.82	<.0001

You can examine the quality of the fit of this model with various residual plots. A panel of studentized residuals is requested with the following statements:

```
ods graphics on;
ods select StudentPanel;

proc glimmix data=lipcancer plots=studentpanel;
   class county;
   x    = employment / 10;
   logn = log(expCount);
   model observed = x / dist=poisson offset=logn s ddfm=none;
   random county;
run;

ods graphics off;
```

The graph in the upper-left corner of the panel displays studentized residuals plotted against the linear predictor (Output 38.3.6). The default of the GLIMMIX procedure is to use the estimated BLUPs in the construction of the residuals and to present them on the linear scale, which in this case is the logarithmic scale. You can change the type of the computed residual with the TYPE= suboptions of each paneled display. For example, the option PLOTS=STUDENTPANEL(TYPE=NOBLUP) would request a paneled display of the marginal residuals on the linear scale.

Output 38.3.6 Panel of Studentized Residuals



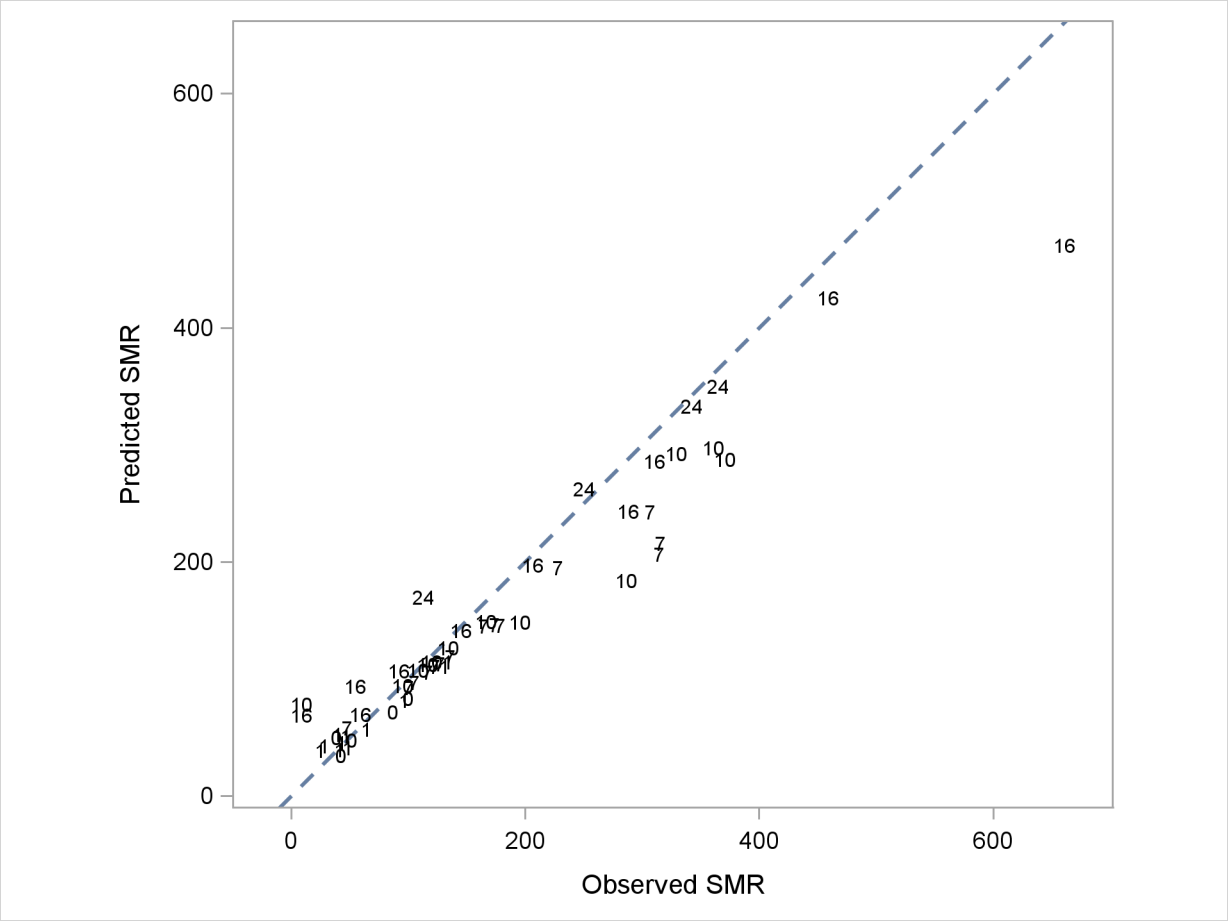
The graph in the upper-right corner of the panel shows a histogram with overlaid normal density. A Q-Q plot and a box plot are shown in the lower cells of the panel.

The following statements produce a graph of the observed and predicted standardized mortality ratios (Output 38.3.7):

```
proc template;
   define statgraph scatter;
   BeginGraph;
     layout overlayequated / yaxisopts=(label='Predicted SMR')
                             xaxisopts=(label='Observed SMR')
                             equatetype=square;
        lineparm y=0 slope=1 x=0 /
               lineattrs = GraphFit(pattern=dash)
               extend    = true;
        scatterplot y=SMR_pred x=SMR /
                markercharacter = employment;
     endlayout;
   EndGraph;
   end;
run;
proc sgrender data=glimmixout template=scatter;
run;
```

In Output 38.3.7, fitted SMRs tend to be larger than the observed SMRs for counties with small observed SMR and smaller than the observed SMRs for counties with high observed SMR.

Output 38.3.7 Observed and Predicted SMRs; Data Labels Indicate Covariate Values



To demonstrate the impact of the random effects adjustment to the log-relative risks, the following statements fit a Poisson regression model (a GLM) by maximum likelihood:

```
proc glimmix data=lipcancer;
   x    = employment / 10;
   logn = log(expCount);
   model observed = x / dist=poisson offset=logn
                        solution ddfm=none;
   SMR_pred = 100*exp(_zgamma_ + _xbeta_);
   id employment SMR SMR_pred;
   output out=glimmixout;
run;
```

The GLIMMIX procedure defaults to maximum likelihood estimation because these statements fit a generalized linear model with nonnormal distribution. As a consequence, the SMRs are county specific only to the extent that the risks vary with the value of the covariate. But risks are no longer adjusted based on county-to-county heterogeneity in the observed incidence count.

Because of the absence of random effects, the GLIMMIX procedure recognizes the model as a generalized linear model and fits it by maximum likelihood (Output 38.3.8). The variance matrix is diagonal because the observations are uncorrelated.

Output 38.3.8 Model Information in Poisson GLM

The GLIMMIX Procedure

Model Information

Data Set	WORK.LIPCANCER
Response Variable	observed
Response Distribution	Poisson
Link Function	Log
Variance Function	Default
Offset Variable	logn = log(expCount);
Variance Matrix	Diagonal
Estimation Technique	Maximum Likelihood
Degrees of Freedom Method	None

The "Dimensions" table shows that there are no G-side random effects in this model and no R-side scale parameter either (Output 38.3.9).

Output 38.3.9 Model Dimensions Information in Poisson GLM

Dimensions

Columns in X	2
Columns in Z	0
Subjects (Blocks in V)	1
Max Obs per Subject	56

Because this is a GLM, the GLIMMIX procedure defaults to the Newton-Raphson algorithm, and the fixed effects (intercept and slope) comprise the parameters in the optimization (Output 38.3.10). (The default optimization technique for a GLM is the Newton-Raphson method.)

Output 38.3.10 Optimization Information in Poisson GLM

Optimization Information

Optimization Technique	Newton-Raphson
Parameters in Optimization	2
Lower Boundaries	0
Upper Boundaries	0
Fixed Effects	Not Profiled

The estimates of β_0 and β_1 have changed from the previous analysis. In the GLMM, the estimates were $\widehat{\beta}_0 = -0.4406$ and $\widehat{\beta}_1 = 0.6799$ (Output 38.3.11).

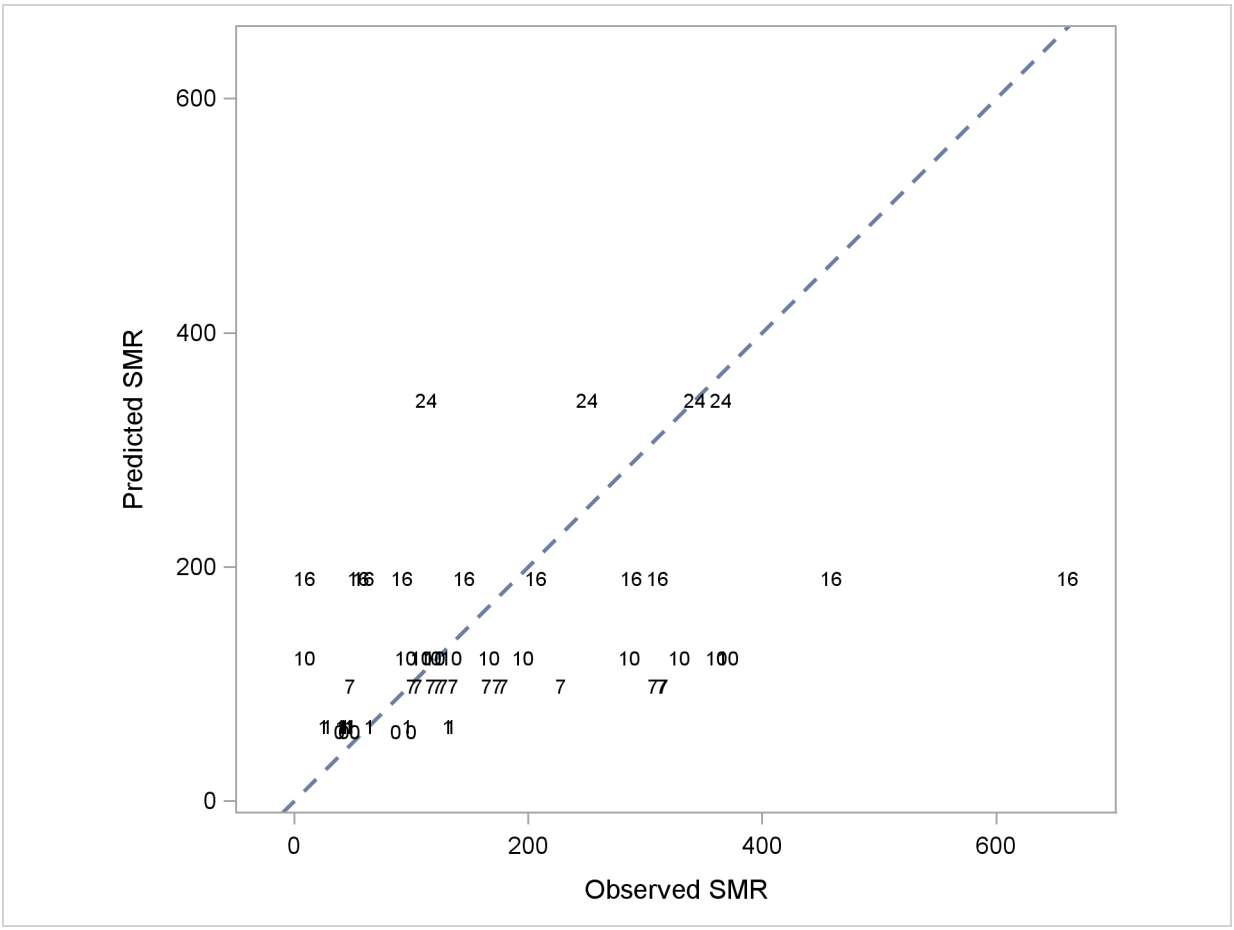
Output 38.3.11 Parameter Estimates in Poisson GLM

Parameter Estimates

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	−0.5419	0.06951	Infty	−7.80	<.0001
x	0.7374	0.05954	Infty	12.38	<.0001

More importantly, without the county-specific adjustments through the best linear unbiased predictors of the random effects, the predicted SMRs are the same for all counties with the same percentage of employees in agriculture, fisheries, and forestry (Output 38.3.12).

Output 38.3.12 Observed and Predicted SMRs in Poisson GLM



Example 38.4: Quasi-likelihood Estimation for Proportions with Unknown Distribution

Wedderburn (1974) analyzes data on the incidence of leaf blotch (*Rhynchosporium secalis*) on barley. The data represent the percentage of leaf area affected in a two-way layout with 10 barley varieties at nine sites. The following DATA step converts these data to proportions, as analyzed in McCullagh and Nelder (1989, Ch. 9.2.4). The purpose of the analysis is to make comparisons among the varieties, adjusted for site effects.

```
data blotch;
  array p{9} pct1-pct9;
  input variety pct1-pct9;
  do site = 1 to 9;
     prop = p{site}/100;
     output;
  end;
  drop pct1-pct9;
  datalines;
1  0.05  0.00  1.25  2.50  5.50  1.00  5.00  5.00 17.50
2  0.00  0.05  1.25  0.50  1.00  5.00  0.10 10.00 25.00
3  0.00  0.05  2.50  0.01  6.00  5.00  5.00  5.00 42.50
4  0.10  0.30 16.60  3.00  1.10  5.00  5.00  5.00 50.00
5  0.25  0.75  2.50  2.50  2.50  5.00 50.00 25.00 37.50
6  0.05  0.30  2.50  0.01  8.00  5.00 10.00 75.00 95.00
7  0.50  3.00  0.00 25.00 16.50 10.00 50.00 50.00 62.50
8  1.30  7.50 20.00 55.00 29.50  5.00 25.00 75.00 95.00
9  1.50  1.00 37.50  5.00 20.00 50.00 50.00 75.00 95.00
10 1.50 12.70 26.25 40.00 43.50 75.00 75.00 75.00 95.00
;
```

Little is known about the distribution of the leaf area proportions. The outcomes are not binomial proportions, because they do not represent the ratio of a count over a total number of Bernoulli trials. However, because the mean proportion μ_{ij} for variety j on site i must lie in the interval $[0, 1]$, you can commence the analysis with a model that treats Prop as a "pseudo-binomial" variable:

$$
\begin{aligned}
\mathrm{E}[\mathsf{Prop}_{ij}] &= \mu_{ij} \\
\mu_{ij} &= 1/(1+\exp\{-\eta_{ij}\}) \\
\eta_{ij} &= \beta_0 + \alpha_i + \tau_j \\
\mathrm{Var}[\mathsf{Prop}_{ij}] &= \phi\mu_{ij}(1-\mu_{ij})
\end{aligned}
$$

Here, η_{ij} is the linear predictor for variety j on site i, α_i denotes the ith site effect, and τ_j denotes the jth barley variety effect. The logit of the expected leaf area proportions is linearly related to these effects. The variance funcion of the model is that of a binomial(n,μ_{ij}) variable, and ϕ is an overdispersion parameter. The moniker "pseudo-binomial" derives not from the pseudo-likelihood methods used to estimate the parameters in the model, but from treating the response variable as if it had first and second moment properties akin to a binomial random variable.

The model is fit in the GLIMMIX procedure with the following statements:

```
proc glimmix data=blotch;
  class site variety;
  model prop = site variety / link=logit dist=binomial;
  random _residual_;
  lsmeans variety / diff=control('1');
run;
```

The MODEL statement specifies the distribution as binomial and the logit link. Because the variance function of the binomial distribution is $a(\mu) = \mu(1-\mu)$, you use the statement

```
random _residual_;
```

to specify the scale parameter ϕ. The LSMEANS statement requests estimates of the least squares means for the barley variety. The DIFF=CONTROL('1') option requests tests of least squares means differences against the first variety.

The "Model Information" table in Output 38.4.1 describes the model and methods used in fitting the statistical model. It is assumed here that the data are binomial proportions.

Output 38.4.1 Model Information in Pseudo-binomial Analysis

The GLIMMIX Procedure

Model Information

Data Set	WORK.BLOTCH
Response Variable	prop
Response Distribution	Binomial
Link Function	Logit
Variance Function	Default
Variance Matrix	Diagonal
Estimation Technique	Maximum Likelihood
Degrees of Freedom Method	Residual

The "Class Level Information" table in Output 38.4.2 lists the number of levels of the Site and Variety effects and their values. All 90 observations read from the data are used in the analysis.

Output 38.4.2 Class Levels and Number of Observations

Class Level Information

Class	Levels	Values
site	9	1 2 3 4 5 6 7 8 9
variety	10	1 2 3 4 5 6 7 8 9 10

Number of Observations Read	90
Number of Observations Used	90

In Output 38.4.3, the "Dimensions" table shows that the model does not contain G-side random effects. There is a single covariance parameter, which corresponds to ϕ. The "Optimization Information" table shows that the optimization comprises 18 parameters (Output 38.4.3). These correspond to the 18 nonsingular columns of the $\mathbf{X}'\mathbf{X}$ matrix.

Output 38.4.3 Model Fit in Pseudo-binomial Analysis

Dimensions

Covariance Parameters	1
Columns in X	20
Columns in Z	0
Subjects (Blocks in V)	1
Max Obs per Subject	90

Optimization Information

Optimization Technique	Newton-Raphson
Parameters in Optimization	18
Lower Boundaries	0
Upper Boundaries	0
Fixed Effects	Not Profiled

Fit Statistics

-2 Log Likelihood	57.15
AIC (smaller is better)	93.15
AICC (smaller is better)	102.79
BIC (smaller is better)	138.15
CAIC (smaller is better)	156.15
HQIC (smaller is better)	111.30
Pearson Chi-Square	6.39
Pearson Chi-Square / DF	0.09

There are significant site and variety effects in this model based on the approximate Type III F tests (Output 38.4.4).

Output 38.4.4 Tests of Site and Variety Effects in Pseudo-binomial Analysis

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
site	8	72	18.25	<.0001
variety	9	72	13.85	<.0001

Output 38.4.5 displays the Variety least squares means for this analysis. These are obtained by averaging

$$\text{logit}(\widehat{\mu}_{ij}) = \widehat{\eta}_{ij}$$

across the sites. In other words, LS-means are computed on the linked scale where the model effects are additive. Note that the least squares means are ordered by variety. The estimate of the expected proportion of infected leaf area for the first variety is

$$\widehat{\mu}_{.,1} = \frac{1}{1 + \exp\{4.38\}} = 0.0124$$

and that for the last variety is

$$\widehat{\mu}_{.,10} = \frac{1}{1 + \exp\{0.127\}} = 0.468$$

Output 38.4.5 Variety Least Squares Means in Pseudo-binomial Analysis

variety Least Squares Means

variety	Estimate	Standard Error	DF	t Value	Pr > |t|
1	-4.3800	0.5643	72	-7.76	<.0001
2	-4.2300	0.5383	72	-7.86	<.0001
3	-3.6906	0.4623	72	-7.98	<.0001
4	-3.3319	0.4239	72	-7.86	<.0001
5	-2.7653	0.3768	72	-7.34	<.0001
6	-2.0089	0.3320	72	-6.05	<.0001
7	-1.8095	0.3228	72	-5.61	<.0001
8	-1.0380	0.2960	72	-3.51	0.0008
9	-0.8800	0.2921	72	-3.01	0.0036
10	-0.1270	0.2808	72	-0.45	0.6523

Because of the ordering of the least squares means, the differences against the first variety are also ordered from smallest to largest (Output 38.4.6).

Output 38.4.6 Variety Differences against the First Variety

Differences of variety Least Squares Means

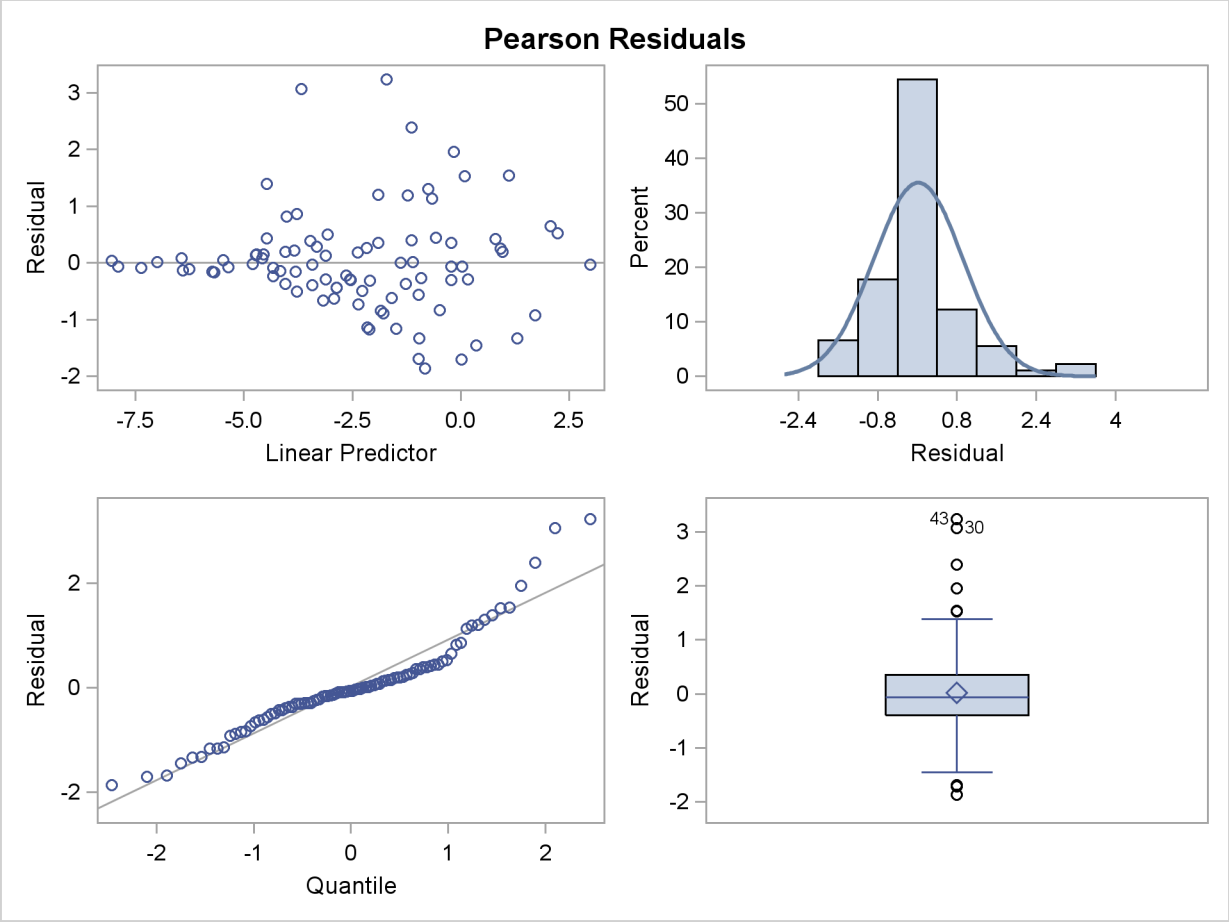
variety	_variety	Estimate	Standard Error	DF	t Value	Pr > |t|
2	1	0.1501	0.7237	72	0.21	0.8363
3	1	0.6895	0.6724	72	1.03	0.3086
4	1	1.0482	0.6494	72	1.61	0.1109
5	1	1.6147	0.6257	72	2.58	0.0119
6	1	2.3712	0.6090	72	3.89	0.0002
7	1	2.5705	0.6065	72	4.24	<.0001
8	1	3.3420	0.6015	72	5.56	<.0001
9	1	3.5000	0.6013	72	5.82	<.0001
10	1	4.2530	0.6042	72	7.04	<.0001

This analysis depends on your choice for the variance function that was implied by the binomial distribution. You can diagnose the distributional assumption by examining various graphical diagnotics measures. The following statements request a panel display of the Pearson-type residuals:

```
ods graphics on;
ods select PearsonPanel;
proc glimmix data=blotch plots=pearsonpanel;
  class site variety;
  model prop = site variety / link=logit dist=binomial;
  random _residual_;
run;
ods graphics off;
```

Output 38.4.7 clearly indicates that the chosen variance function is not appropriate for these data. As μ approaches zero or one, the variability in the residuals is less than that implied by the binomial variance function.

Output 38.4.7 Panel of Pearson-Type Residuals in Pseudo-binomial Analysis



To remedy this situation, McCullagh and Nelder (1989) consider instead the variance function

$$\text{Var}[\text{Prop}_{ij}] = \mu_{ij}^2(1-\mu_{ij})^2$$

Imagine two varieties with $\mu_{.i} = 0.1$ and $\mu_{.k} = 0.5$. Under the binomial variance function, the variance of the proportion for variety k is 2.77 times larger than that for variety i. Under the revised model this ratio increases to $2.77^2 = 7.67$.

The analysis of the revised model is obtained with the next set of GLIMMIX statements. Because you need to model a variance function that does not correspond to any of the built-in distributions, you need to supply a function with an assignment to the automatic variable _VARIANCE_. The GLIMMIX procedure then considers the distribution of the data as unknown. The corresponding estimation technique is quasi-likelihood. Because this model does not include an extra scale parameter, you can drop the RANDOM _RESIDUAL_ statement from the analysis.

```
ods graphics on;
ods select ModelInfo FitStatistics LSMeans Diffs PearsonPanel;
proc glimmix data=blotch plots=pearsonpanel;
  class site variety;
  _variance_ = _mu_**2 * (1-_mu_)**2;
  model prop = site variety / link=logit;
  lsmeans variety / diff=control('1');
run;
ods graphics off;
```

The "Model Information" table in Output 38.4.8 now displays the distribution as "Unknown," because of the assignment made in the GLIMMIX statements to _VARIANCE_. The table also shows the expression evaluated as the variance function.

Output 38.4.8 Model Information in Quasi-likelihood Analysis

The GLIMMIX Procedure

Model Information

Data Set	WORK.BLOTCH
Response Variable	prop
Response Distribution	Unknown
Link Function	Logit
Variance Function	_mu_**2 * (1-_mu_)**2
Variance Matrix	Diagonal
Estimation Technique	Quasi-Likelihood
Degrees of Freedom Method	Residual

The fit statistics of the model are now expressed in terms of the log quasi-likelihood. It is computed as

$$\sum_{i=1}^{9}\sum_{j=1}^{10}\int_{y_{ij}}^{\mu_{ij}} \frac{y_{ij}-t}{t^2(1-t)^2}\,dt$$

Twice the negative of this sum equals −85.74, which is displayed in the "Fit Statistics" table (Output 38.4.9).

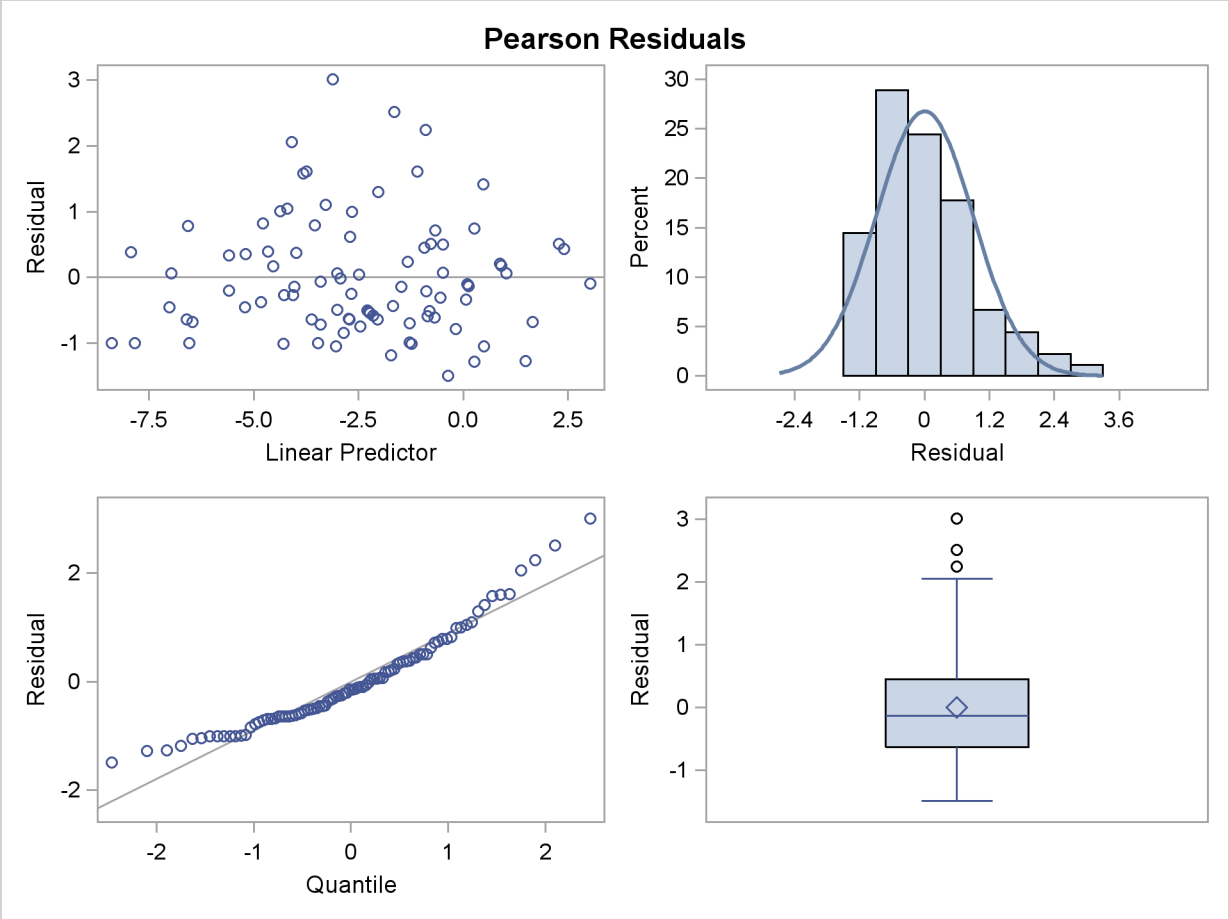
The scaled Pearson statistic is now 0.99. Inclusion of an extra scale parameter ϕ would have little or no effect on the results.

Output 38.4.9 Fit Statistics in Quasi-likelihood Analysis

Fit Statistics	
-2 Log Quasi-Likelihood	-85.74
Quasi-AIC (smaller is better)	-49.74
Quasi-AICC (smaller is better)	-40.11
Quasi-BIC (smaller is better)	-4.75
Quasi-CAIC (smaller is better)	13.25
Quasi-HQIC (smaller is better)	-31.60
Pearson Chi-Square	71.17
Pearson Chi-Square / DF	0.99

The panel of Pearson-type residuals now shows a much more adequate distribution for the residuals and a reduction in the number of outlying residuals (Output 38.4.10).

Output 38.4.10 Panel of Pearson-Type Residuals (Quasi-likelihood)



The least squares means are no longer ordered in size by variety (Output 38.4.11). For example, $\text{logit}(\widehat{\mu}_{.1}) > \text{logit}(\widehat{\mu}_{.2})$. Under the revised model, the second variety has a greater percentage of its leaf area covered by blotch, compared to the first variety. Varieties 5 and 6 and varieties 8 and 9 show similar reversal in ranking.

Output 38.4.11 Variety Least Squares Means in Quasi-likelihood Analysis

variety Least Squares Means

variety	Estimate	Standard Error	DF	t Value	Pr > |t|
1	−4.0453	0.3333	72	−12.14	<.0001
2	−4.5126	0.3333	72	−13.54	<.0001
3	−3.9664	0.3333	72	−11.90	<.0001
4	−3.0912	0.3333	72	−9.27	<.0001
5	−2.6927	0.3333	72	−8.08	<.0001
6	−2.7167	0.3333	72	−8.15	<.0001
7	−1.7052	0.3333	72	−5.12	<.0001
8	−0.7827	0.3333	72	−2.35	0.0216
9	−0.9098	0.3333	72	−2.73	0.0080
10	−0.1580	0.3333	72	−0.47	0.6369

Interestingly, the standard errors are constant among the LS-means (Output 38.4.11) and among the LS-means differences (Output 38.4.12). This is due to the fact that for the logit link

$$\frac{\partial \mu}{\partial \eta} = \mu(1-\mu)$$

which cancels with the square root of the variance function in the estimating equations. The analysis is thus orthogonal.

Output 38.4.12 Variety Differences in Quasi-likelihood Analysis

Differences of variety Least Squares Means

variety	_variety	Estimate	Standard Error	DF	t Value	Pr > |t|
2	1	−0.4673	0.4714	72	−0.99	0.3249
3	1	0.07885	0.4714	72	0.17	0.8676
4	1	0.9541	0.4714	72	2.02	0.0467
5	1	1.3526	0.4714	72	2.87	0.0054
6	1	1.3286	0.4714	72	2.82	0.0062
7	1	2.3401	0.4714	72	4.96	<.0001
8	1	3.2626	0.4714	72	6.92	<.0001
9	1	3.1355	0.4714	72	6.65	<.0001
10	1	3.8873	0.4714	72	8.25	<.0001

Example 38.5: Joint Modeling of Binary and Count Data

Clustered data arise when multiple observations are collected on the same sampling or experimental unit. Often, these multiple observations refer to the same attribute measured at different points in time or space. This leads to repeated measures, longitudinal, and spatial data, which are special forms of multivariate data. A different class of multivariate data arises when the multiple observations refer to different attributes.

The data set hernio, created in the following DATA step, provides an example of a bivariate outcome variable. It reflects the condition and length of hospital stay for 32 herniorrhaphy patients. These data are based on data given by Mosteller and Tukey (1977) and reproduced in Hand et al. (1994, pp. 390, 391). The data set that follows does not contain all the covariates given in these sources. The response variables are leave and los; these denote the condition of the patient upon leaving the operating room and the length of hospital stay after the operation (in days). The variable leave takes on the value one if a patient experiences a routine recovery, and the value zero if postoperative intensive care was required. The binary variable OKstatus distinguishes patients based on their postoperative physical status ("1" implies better status).

```
data hernio;
  input patient age gender$ OKstatus leave los;
  datalines;
 1   78  m   1   0   9
 2   60  m   1   0   4
 3   68  m   1   1   7
 4   62  m   0   1  35
 5   76  m   0   0   9
 6   76  m   1   1   7
 7   64  m   1   1   5
 8   74  f   1   1  16
 9   68  m   0   1   7
10   79  f   1   0  11
11   80  f   0   1   4
12   48  m   1   1   9
13   35  f   1   1   2
14   58  m   1   1   4
15   40  m   1   1   3
16   19  m   1   1   4
17   79  m   0   0   3
18   51  m   1   1   5
19   57  m   1   1   8
20   51  m   0   1   8
21   48  m   1   1   3
22   48  m   1   1   5
23   66  m   1   1   8
24   71  m   1   0   2
25   75  f   0   0   7
26    2  f   1   1   0
27   65  f   1   0  16
28   42  f   1   0   3
29   54  m   1   0   2
```

```
30   43  m   1   1   3
31    4  m   1   1   3
32   52  m   1   1   8
;
```

While the response variable los is a Poisson count variable, the response variable leave is a binary variable. You can perform separate analysis for the two outcomes, for example, by fitting a logistic model for the operating room exit condition and a Poisson regression model for the length of hospital stay. This, however, would ignore the correlation between the two outcomes. Intuitively, you would expect that the length of postoperative hospital stay is longer for those patients who had more tenuous exit conditions.

The following DATA step converts the data set hernio from the multivariate form to the univariate form. In the multivariate form the responses are stored in separate variables. The GLIMMIX procedure requires the univariate data structure.

```
data hernio_uv;
   length dist $7;
   set hernio;
   response = (leave=1);
   dist     = "Binary";
   output;
   response = los;
   dist     = "Poisson";
   output;
   keep patient age OKstatus response dist;
run;
```

This DATA step expands the 32 observations in the data set hernio into 64 observations, stacking two observations per patient. The character variable dist identifies the distribution that is assumed for the respective observations within a patient. The first observation for each patient corresponds to the binary response.

The following GLIMMIX statements fit a logistic regression model with two regressors (age and OKStatus) to the binary observations:

```
proc glimmix data=hernio_uv(where=(dist="Binary"));
   model response(event='1') = age OKStatus / s dist=binary;
run;
```

The EVENT=('1') response option requests that PROC GLIMMIX model the probability Pr(leave $=$ 1)—that is, the probability of routine recovery. The fit statistics and parameter estimates for this univariate analysis are shown in Output 38.5.1. The coefficient for the age effect is negative (-0.07725) and marginally significant at the 5% level ($p = 0.0491$). The negative sign indicates that the probability of routine recovery decreases with age. The coefficient for the OKStatus variable is also negative. Its large standard error and the p-value of 0.7341 indicate, however, that this regressor is not significant.

Output 38.5.1 Univariate Logistic Regression

The GLIMMIX Procedure

Fit Statistics

-2 Log Likelihood	32.77
AIC (smaller is better)	38.77
AICC (smaller is better)	39.63
BIC (smaller is better)	43.17
CAIC (smaller is better)	46.17
HQIC (smaller is better)	40.23
Pearson Chi-Square	30.37
Pearson Chi-Square / DF	1.05

Parameter Estimates

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	5.7694	2.8245	29	2.04	0.0503
age	-0.07725	0.03761	29	-2.05	0.0491
OKstatus	-0.3516	1.0253	29	-0.34	0.7341

Based on the univariate logistic regression analysis, you would probably want to revisit the model, examine other regressor variables, test for gender effects and interactions, and so forth. For this example, we are content with the two-regressor model. It will be illustrative to trace the relative importance of the two regressors through various types of models.

The next statements fit the same regressors to the count data:

```
proc glimmix data=hernio_uv(where=(dist="Poisson"));
   model response = age OKStatus / s dist=Poisson;
run;
```

For this response, both regressors appear to make significant contributions at the 5% significance level (Output 38.5.2). The sign of the coefficient seems appropriate; the length of hospital stay should increase with patient age and be shorter for patients with better preoperative health. The magnitude of the scaled Pearson statistic (4.48) indicates, however, that there is considerable overdispersion in this model. This could be due to omitted variables or an improper distributional assumption. The importance of preoperative health status, for example, can change with a patient's age, which could call for an interaction term.

Output 38.5.2 Univariate Poisson Regression

The GLIMMIX Procedure

Fit Statistics

-2 Log Likelihood	215.52
AIC (smaller is better)	221.52
AICC (smaller is better)	222.38
BIC (smaller is better)	225.92
CAIC (smaller is better)	228.92
HQIC (smaller is better)	222.98
Pearson Chi-Square	129.98
Pearson Chi-Square / DF	4.48

Parameter Estimates

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	1.2640	0.3393	29	3.72	0.0008
age	0.01525	0.004454	29	3.42	0.0019
OKstatus	-0.3301	0.1562	29	-2.11	0.0433

You can also model both responses jointly. The following statements request a multivariate analysis:

```
proc glimmix data=hernio_uv;
   class dist;
   model response(event='1') = dist dist*age dist*OKstatus /
                    noint s dist=byobs(dist);
run;
```

The DIST=BYOBS option in the MODEL statement instructs the GLIMMIX procedure to examine the variable dist in order to identify the distribution of an observation. The variable can be character or numeric. See the DIST= option of the MODEL statement for a list of the numeric codes for the various distributions that are compatible with the DIST=BYOBS formulation. Because no LINK= option is specified, the link functions are chosen as the default links that correspond to the respective distributions. In this case, the logit link is applied to the binary observations and the log link is applied to the Poisson outcomes. The dist variable is also listed in the CLASS statement, which enables you to use interaction terms in the MODEL statement to vary the regression coefficients by response distribution. The NOINT option is used here so that the parameter estimates of the joint model are directly comparable to those in Output 38.5.1 and Output 38.5.2.

The "Fit Statistics" and "Parameter Estimates" tables of this bivariate estimation process are shown in Output 38.5.3.

Output 38.5.3 Bivariate Analysis – Independence

The GLIMMIX Procedure

Fit Statistics

Description	Binary	Poisson	Total
-2 Log Likelihood	32.77	215.52	248.29
AIC (smaller is better)	44.77	227.52	260.29
AICC (smaller is better)	48.13	230.88	261.77
BIC (smaller is better)	53.56	236.32	273.25
CAIC (smaller is better)	59.56	242.32	279.25
HQIC (smaller is better)	47.68	230.44	265.40
Pearson Chi-Square	30.37	129.98	160.35
Pearson Chi-Square / DF	1.05	4.48	2.76

Parameter Estimates

Effect	dist	Estimate	Standard Error	DF	t Value	Pr > |t|
dist	Binary	5.7694	2.8245	58	2.04	0.0456
dist	Poisson	1.2640	0.3393	58	3.72	0.0004
age*dist	Binary	-0.07725	0.03761	58	-2.05	0.0445
age*dist	Poisson	0.01525	0.004454	58	3.42	0.0011
OKstatus*dist	Binary	-0.3516	1.0253	58	-0.34	0.7329
OKstatus*dist	Poisson	-0.3301	0.1562	58	-2.11	0.0389

The "Fit Statistics" table now contains a separate column for each response distribution, as well as an overall contribution. Because the model does not specify any random effects or R-side correlations, the log likelihoods are additive. The parameter estimates and their standard errors in this joint model are identical to those in Output 38.5.1 and Output 38.5.2. The p-values reflect the larger "sample size" in the joint analysis. Note that the coefficients would be different from the separate analyses if the dist variable had not been used to form interactions with the model effects.

There are two ways in which the correlations between the two responses for the same patient can be incorporated. You can induce them through shared random effects or model the dependency directly. The following statements fit a model that induces correlation:

```
proc glimmix data=hernio_uv;
   class patient dist;
   model response(event='1') = dist dist*age dist*OKstatus /
                    noint s dist=byobs(dist);
   random int / subject=patient;
run;
```

Notice that the patient variable has been added to the CLASS statement and as the SUBJECT= effect in the RANDOM statement.

The "Fit Statistics" table in Output 38.5.4 no longer has separate columns for each response distribution, because the data are not independent. The log (pseudo-)likelihood does not factor into

additive component that correspond to distributions. Instead, it factors into components associated with subjects.

Output 38.5.4 Bivariate Analysis – Mixed Model

The GLIMMIX Procedure

Fit Statistics

-2 Res Log Pseudo-Likelihood	226.71
Generalized Chi-Square	52.25
Gener. Chi-Square / DF	0.90

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error
Intercept	patient	0.2990	0.1116

Solutions for Fixed Effects

Effect	dist	Estimate	Standard Error	DF	t Value	Pr > |t|
dist	Binary	5.7783	2.9048	29	1.99	0.0562
dist	Poisson	0.8410	0.5696	29	1.48	0.1506
age*dist	Binary	-0.07572	0.03791	29	-2.00	0.0552
age*dist	Poisson	0.01875	0.007383	29	2.54	0.0167
OKstatus*dist	Binary	-0.4697	1.1251	29	-0.42	0.6794
OKstatus*dist	Poisson	-0.1856	0.3020	29	-0.61	0.5435

The estimate of the variance of the random patient intercept is 0.2990, and the estimated standard error of this variance component estimate is 0.1116. There appears to be significant patient-to-patient variation in the intercepts. The estimates of the fixed effects as well as their estimated standard errors have changed from the bivariate-independence analysis (see Output 38.5.3). When the length of hospital stay and the postoperative condition are modeled jointly, the preoperative health status (variable OKStatus) no longer appears significant. Compare this result to Output 38.5.3; in the separate analyses the initial health status was a significant predictor of the length of hospital stay. A further joint analysis of these data would probably remove this predictor from the model entirely.

A joint model of the second kind, where correlations are modeled directly, is fit with the following GLIMMIX statements:

```
proc glimmix data=hernio_uv;
   class patient dist;
   model response(event='1') = dist dist*age dist*OKstatus /
                   noint s dist=byobs(dist);
   random _residual_ / subject=patient type=chol;
run;
```

Instead of a shared G-side random effect, an R-side covariance structure is used to model the correlations. It is important to note that this is a marginal model that models covariation on the scale of the data. The previous model involves the $\mathbf{Z}\boldsymbol{\gamma}$ random components inside the linear predictor.

The _RESIDUAL_ keyword instructs PROC GLIMMIX to model the R-side correlations. Because of the SUBJECT=PATIENT option, data from different patients are independent, and data from a single patient follow the covariance model specified with the TYPE= option. In this case, a generally unstructured 2×2 covariance matrix is modeled, but in its Cholesky parameterization. This ensures that the resulting covariance matrix is at least positive semidefinite and stabilizes the numerical optimizations.

Output 38.5.5 Bivariate Analysis – Marginal Correlated Error Model

The GLIMMIX Procedure

Fit Statistics

-2 Res Log Pseudo-Likelihood	240.98
Generalized Chi-Square	58.00
Gener. Chi-Square / DF	1.00

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error
CHOL(1,1)	patient	1.0162	0.1334
CHOL(2,1)	patient	0.3942	0.3893
CHOL(2,2)	patient	2.0819	0.2734

Solutions for Fixed Effects

Effect	dist	Estimate	Standard Error	DF	t Value	Pr > |t|
dist	Binary	5.6514	2.8283	26	2.00	0.0563
dist	Poisson	1.2463	0.7189	26	1.73	0.0948
age*dist	Binary	-0.07568	0.03765	26	-2.01	0.0549
age*dist	Poisson	0.01548	0.009432	26	1.64	0.1128
OKstatus*dist	Binary	-0.3421	1.0384	26	-0.33	0.7445
OKstatus*dist	Poisson	-0.3253	0.3310	26	-0.98	0.3349

The "Covariance Parameter Estimates" table in Output 38.5.5 contains three entries for this model, corresponding to a (2×2) covariance matrix for each patient. The Cholesky root of the $\mathbf{R}$ matrix is

$$\mathbf{L} = \begin{bmatrix} 1.0162 & 0 \\ 0.3942 & 2.0819 \end{bmatrix}$$

so that the covariance matrix can be obtained as

$$\mathbf{LL}' = \begin{bmatrix} 1.0162 & 0 \\ 0.3942 & 2.0819 \end{bmatrix} \begin{bmatrix} 1.0162 & 0.3942 \\ 0 & 2.0819 \end{bmatrix} = \begin{bmatrix} 1.0326 & 0.4005 \\ 0.4005 & 4.4897 \end{bmatrix}$$

This is not the covariance matrix of the data, however, because the variance functions need to be accounted for.

The p-values in the "Solutions for Fixed Effects" table indicate the same pattern of significance and nonsignificance as in the conditional model with random patient intercepts.

Example 38.6: Radial Smoothing of Repeated Measures Data

This example of a repeated measures study is taken from Diggle, Liang, and Zeger (1994, p. 100). The data consist of body weights of 27 cows, measured at 23 unequally spaced time points over a period of approximately 22 months. Following Diggle, Liang, and Zeger (1994), one animal is removed from the analysis, one observation is removed according to their Figure 5.7, and the time is shifted to start at 0 and is measured in 10-day increments. The design is a 2×2 factorial, and the factors are the infection of an animal with *M. paratuberculosis* and whether the animal is receiving iron dosing.

The following DATA steps create the data and arrange them in univariate format.

```
data times;
  input time1-time23;
  datalines;
 122  150  166  179  219  247  276  296  324  354  380  445
 478  508  536  569  599  627  655  668  723  751  781
;

data cows;
  if _n_ = 1 then merge times;
  array t{23} time1   - time23;
  array w{23} weight1 - weight23;
  input cow iron infection weight1-weight23 @@;
  do i=1 to 23;
     weight = w{i};
     tpoint = (t{i}-t{1})/10;
     output;
  end;
  keep cow iron infection tpoint weight;
  datalines;
  1 0 0  4.7    4.905  5.011  5.075  5.136  5.165  5.298  5.323
         5.416  5.438  5.541  5.652  5.687  5.737  5.814  5.799
         5.784  5.844  5.886  5.914  5.979  5.927  5.94
  2 0 0  4.868  5.075  5.193  5.22   5.298  5.416  5.481  5.521
         5.617  5.635  5.687  5.768  5.799  5.872  5.886  5.872
         5.914  5.966  5.991  6.016  6.087  6.098  6.153
  3 0 0  4.868  5.011  5.136  5.193  5.273  5.323  5.416  5.46
         5.521  5.58   5.617  5.687  5.72   5.753  5.784  5.784
         5.784  5.814  5.829  5.872  5.927  5.9    5.991
  4 0 0  4.828  5.011  5.136  5.193  5.273  5.347  5.438  5.561
         5.541  5.598  5.67    .     5.737  5.844  5.858  5.872
         5.886  5.927  5.94   5.979  6.052  6.028  6.12
  5 1 0  4.787  4.977  5.043  5.136  5.106  5.298  5.298  5.371
```

```
        5.438  5.501  5.561  5.652  5.67   5.737  5.784  5.768
        5.784  5.784  5.829  5.858  5.914  5.9    5.94
 6 1 0  4.745  4.868  5.043  5.106  5.22   5.298  5.347  5.347
        5.416  5.501  5.561  5.58   5.687  5.72   5.737  5.72
        5.737  5.753  5.768  5.784  5.844  5.844  5.9
 7 1 0  4.745  4.905  5.011  5.106  5.165  5.273  5.371  5.416
        5.416  5.521  5.541  5.635  5.687  5.704  5.784  5.768
        5.768  5.814  5.829  5.858  5.94   5.94   6.004
 8 0 1  4.942  5.106  5.136  5.193  5.298  5.347  5.46   5.521
        5.561  5.58   5.635  5.704  5.784  5.823  5.858  5.9
        5.94   5.991  6.016  6.064  6.052  6.016  5.979
 9 0 1  4.605  4.745  4.868  4.905  4.977  5.22   5.165  5.22
        5.22   5.247  5.298  5.416  5.501  5.521  5.58   5.58
        5.635  5.67   5.72   5.753  5.799  5.829  5.858
10 0 1  4.7    4.868  4.905  4.977  5.011  5.106  5.165  5.22
        5.22   5.22   5.273  5.384  5.438  5.438  5.501  5.501
        5.541  5.598  5.58   5.635  5.687  5.72   5.704
11 0 1  4.828  5.011  5.075  5.165  5.247  5.323  5.394  5.46
        5.46   5.501  5.541  5.609  5.687  5.704  5.72   5.704
        5.704  5.72   5.737  5.768  5.858  5.9    5.94
12 0 1  4.7    4.828  4.905  5.011  5.075  5.165  5.247  5.298
        5.298  5.323  5.416  5.505  5.561  5.58   5.561  5.635
        5.687  5.72   5.72   5.737  5.784  5.814  5.799
13 0 1  4.828  5.011  5.075  5.136  5.22   5.273  5.347  5.416
        5.438  5.416  5.521  5.628  5.67   5.687  5.72   5.72
        5.799  5.858  5.872  5.914  5.94   5.991  6.016
14 0 1  4.828  4.942  5.011  5.075  5.075  5.22   5.273  5.298
        5.323  5.298  5.394  5.489  5.541  5.58   5.617  5.67
        5.704  5.753  5.768  5.814  5.872  5.927  5.927
15 0 1  4.745  4.905  4.977  5.075  5.193  5.22   5.298  5.323
        5.394  5.394  5.438  5.583  5.617  5.652  5.687  5.72
        5.753  5.768  5.814  5.844  5.886  5.886  5.886
16 0 1  4.7    4.868  5.011  5.043  5.106  5.165  5.247  5.298
        5.347  5.371  5.438  5.455  5.617  5.635  5.704  5.737
        5.784  5.768  5.814  5.844  5.886  5.94   5.927
17 1 1  4.605  4.787  4.828  4.942  5.011  5.136  5.22   5.247
        5.273  5.247  5.347  5.366  5.416  5.46   5.541  5.481
        5.501  5.635  5.652  5.598  5.635  5.635  5.598
18 1 1  4.828  4.977  5.011  5.136  5.273  5.298  5.371  5.46
        5.416  5.416  5.438  5.557  5.617  5.67   5.72   5.72
        5.799  5.858  5.886  5.914  5.979  6.004  6.028
19 1 1  4.7    4.905  4.942  5.011  5.043  5.136  5.193  5.193
        5.247  5.22   5.323  5.338  5.371  5.394  5.438  5.416
        5.501  5.561  5.541  5.58   5.652  5.67   5.704
20 1 1  4.745  4.905  4.977  5.043  5.136  5.273  5.347  5.394
        5.416  5.394  5.521  5.617  5.617  5.617  5.67   5.635
        5.652  5.687  5.652  5.617  5.687  5.768  5.814
21 1 1  4.787  4.942  4.977  5.106  5.165  5.247  5.323  5.416
        5.394  5.371  5.438  5.521  5.521  5.561  5.635  5.617
        5.687  5.72   5.737  5.737  5.768  5.768  5.704
22 1 1  4.605  4.828  4.828  4.977  5.043  5.165  5.22   5.273
        5.247  5.22   5.298  5.375  5.371  5.416  5.501  5.501
        5.521  5.561  5.617  5.635  5.72   5.737  5.768
23 1 1  4.7    4.905  5.011  5.075  5.106  5.22   5.22   5.298
```

```
          5.323  5.347  5.416  5.472  5.501  5.541  5.598  5.598
          5.598  5.652  5.67   5.704  5.737  5.768  5.784
 24 1 1   4.745  4.942  5.011  5.075  5.106  5.247  5.273  5.323
          5.347  5.371  5.416  5.481  5.501  5.541  5.598  5.598
          5.635  5.687  5.704  5.72   5.829  5.844  5.9
 25 1 1   4.654  4.828  4.828  4.977  4.977  5.043  5.136  5.165
          5.165  5.165  5.193  5.204  5.22   5.273  5.371  5.347
          5.46   5.58   5.635  5.67   5.753  5.799  5.844
 26 1 1   4.828  4.977  5.011  5.106  5.165  5.22   5.273  5.323
          5.371  5.394  5.46   5.576  5.652  5.617  5.687  5.67
          5.72   5.784  5.784  5.784  5.829  5.814  5.844
;
```

The mean response profiles of the cows are not of particular interest; what matters are inferences about the Iron effect, the Infection effect, and their interaction. Nevertheless, the body weight of the cows changes over the 22-month period, and you need to account for these changes in the analysis. A reasonable approach is to apply the approximate low-rank smoother to capture the trends over time. This approach frees you from having to stipulate a parametric model for the response trajectories over time. In addition, you can test hypotheses about the smoothing parameter; for example, whether it should be varied by treatment.

The following statements fit a model with a 2×2 factorial treatment structure and smooth trends over time, choosing the Newton-Raphson algorithm with ridging for the optimization:

```
proc glimmix data=cows;
  t2 = tpoint / 100;
  class cow iron infection;
  model weight = iron infection iron*infection tpoint;
  random t2 / type=rsmooth subject=cow
                  knotmethod=kdtree(bucket=100 knotinfo);
  output out=gmxout pred(blup)=pred;
  nloptions tech=newrap;
run;
```

The continuous time effect appears in both the MODEL statement (tpoint) and the RANDOM statement (t2). Because the variance of the radial smoothing component depends on the temporal metric, the time scale was rescaled for the RANDOM effect to move the parameter estimate away from the boundary. The knots of the radial smoother are selected as the vertices of a *k-d* tree. Specifying BUCKET=100 sets the bucket size of the tree to $b = 100$. Because measurements at each time point are available for 26 (or 25) cows, this groups approximately four time points in a single bucket. The KNOTINFO keyword of the KNOTMETHOD= option requests a printout of the knot locations for the radial smoother. The OUTPUT statement saves the predictions of the mean of each observations to the data set gmxout. Finally, the TECH=NEWRAP option in the NLOPTIONS statement specifies the Newton-Raphson algorithm for the optimization technique.

The "Class Level Information" table lists the number of levels of the Cow, Iron, and Infection effects (Output 38.6.1).

Output 38.6.1 Model Information and Class Levels in Repeated Measures Analysis

The GLIMMIX Procedure

Model Information

Data Set	WORK.COWS
Response Variable	weight
Response Distribution	Gaussian
Link Function	Identity
Variance Function	Default
Variance Matrix Blocked By	cow
Estimation Technique	Restricted Maximum Likelihood
Degrees of Freedom Method	Containment

Class Level Information

Class	Levels	Values
cow	26	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
iron	2	0 1
infection	2	0 1

The "Radial Smoother Knots for RSmooth(t2)" table displays the knots computed from the vertices of the t2 *k-d* tree (Output 38.6.2). Notice that knots are spaced unequally and that the extreme time points are among the knot locations. The "Number of Observations" table shows that one observation was not used in the analysis. The 12th observation for cow 4 has a missing value.

Output 38.6.2 Knot Information and Number of Observations

Radial Smoother Knots for RSmooth(t2)

Knot Number	t2
1	0
2	0.04400
3	0.1250
4	0.2020
5	0.3230
6	0.4140
7	0.5050
8	0.6010
9	0.6590

Number of Observations Read	598
Number of Observations Used	597

The "Dimensions" table shows that the model contains only two covariance parameters, the G-side variance of the spline coefficients (σ^2) and the R-side scale parameter (ϕ, Output 38.6.3). For

each subject (cow), there are nine columns in the **Z** matrix, one per knot location. The GLIMMIX procedure processes these data by subjects (cows).

Output 38.6.3 Dimensions Information in Repeated Measures Analysis

Dimensions	
G-side Cov. Parameters	1
R-side Cov. Parameters	1
Columns in X	10
Columns in Z per Subject	9
Subjects (Blocks in V)	26
Max Obs per Subject	23

The "Optimization Information" table displays information about the optimization process. Because fixed effects and the residual scale parameter can be profiled from the optimization, the iterative algorithm involves only a single covariance parameter, the variance of the spline coefficients (Output 38.6.4).

Output 38.6.4 Optimization Information in Repeated Measures Analysis

Optimization Information	
Optimization Technique	Newton-Raphson
Parameters in Optimization	1
Lower Boundaries	1
Upper Boundaries	0
Fixed Effects	Profiled
Residual Variance	Profiled
Starting From	Data

After 11 iterations, the optimization process terminates (Output 38.6.5). In this case, the absolute gradient convergence criterion was met.

Output 38.6.5 Iteration History and Convergence Status

Iteration History

Iteration	Restarts	Evaluations	Objective Function	Change	Max Gradient
0	0	4	−1302.549272	.	20.33682
1	0	3	−1451.587367	149.03809501	9.940495
2	0	3	−1585.640946	134.05357887	4.71531
3	0	3	−1694.516203	108.87525722	2.176741
4	0	3	−1775.290458	80.77425512	0.978577
5	0	3	−1829.966584	54.67612585	0.425724
6	0	3	−1862.878184	32.91160012	0.175992
7	0	3	−1879.329133	16.45094875	0.066061
8	0	3	−1885.175082	5.84594887	0.020137
9	0	3	−1886.238032	1.06295071	0.00372
10	0	3	−1886.288519	0.05048659	0.000198
11	0	3	−1886.288673	0.00015425	6.364E−7

Convergence criterion (ABSGCONV=0.00001) satisfied.

The generalized chi-square statistic in the "Fit Statistics" table is small for this model (Output 38.6.6). There is very little residual variation. The radial smoother is associated with 433.55 residual degrees of freedom, computed as 597 minus the trace of the smoother matrix.

Output 38.6.6 Fit Statistics in Repeated Measures Analysis

Fit Statistics

−2 Res Log Likelihood	−1886.29
AIC (smaller is better)	−1882.29
AICC (smaller is better)	−1882.27
BIC (smaller is better)	−1879.77
CAIC (smaller is better)	−1877.77
HQIC (smaller is better)	−1881.56
Generalized Chi-Square	0.47
Gener. Chi-Square / DF	0.00
Radial Smoother df(res)	433.55

The "Covariance Parameter Estimates" table in Output 38.6.7 displays the estimates of the covariance parameters. The variance of the random spline coefficients is estimated as $\widehat{\sigma}^2 = 0.5961$, and the scale parameter (=residual variance) estimate is $\widehat{\phi} = 0.0008$.

Output 38.6.7 Estimated Covariance Parameters

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error
Var[RSmooth(t2)]	cow	0.5961	0.08144
Residual		0.000800	0.000059

The "Type III Tests of Fixed Effects" table displays F tests for the fixed effects in the MODEL statement (Output 38.6.8). There is a strong infection effect as well as the absence of an interaction between infection with *M. paratuberculosis* and iron dosing. It is important to note, however, that the interpretation of these tests rests on the assumption that the random effects in the mixed model have zero mean; in this case, the radial smoother coefficients.

Output 38.6.8 Tests of Fixed Effects

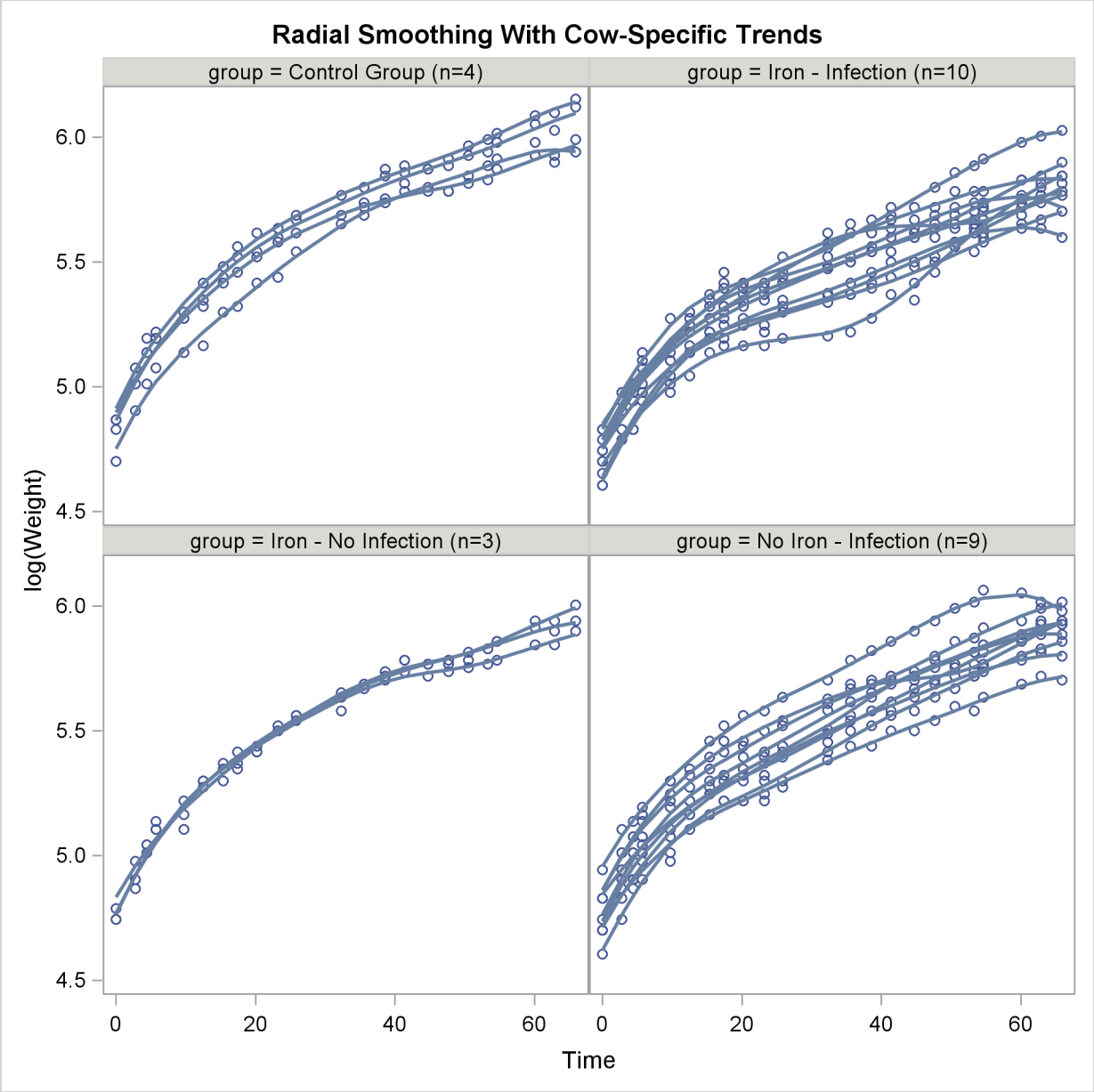
Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
iron	1	358	3.59	0.0588
infection	1	358	21.16	<.0001
iron*infection	1	358	0.09	0.7637
tpoint	1	358	53.88	<.0001

A graph of the observed data and fitted profiles in the four groups is produced with the following statements (Output 38.6.9):

```
data plot; set gmxout;
  length group $26;
       if (iron=0) and (infection=0) then group='Control Group (n=4)';
  else if (iron=1) and (infection=0) then group='Iron - No Infection (n=3)';
  else if (iron=0) and (infection=1) then group='No Iron - Infection (n=9)';
  else group = 'Iron - Infection (n=10)';
run;
proc sort data=plot; by group cow;
run;

proc sgpanel data=plot noautolegend;
  title 'Radial Smoothing With Cow-Specific Trends';
  label tpoint='Time' weight='log(Weight)';
  panelby group / columns=2 rows=2;
  scatter x=tpoint y=weight;
  series  x=tpoint y=pred / group=cow lineattrs=GraphFit;
run;
```

Output 38.6.9 Observed and Predicted Profiles



The trends are quite smooth, and you can see how the radial smoother adapts to the cow-specific profile. This is the reason for the small scale parameter estimate, $\widehat{\phi} = 0.008$. Comparing the panels at the top to the panels at the bottom of Output 38.6.9 reveals the effect of Infection. A comparison of the panels on the left to those on the right indicates the weak Iron effect.

The smoothing parameter in this analysis is related to the covariance parameter estimates. Because there is only one radial smoothing variance component, the amount of smoothing is the same in all four treatment groups. To test whether the smoothing parameter should be varied by group, you can refine the analysis of the previous model. The following statements fit the same general model, but they vary the covariance parameters by the levels of the Iron*Infection interaction. This is accomplished with the GROUP= option in the RANDOM statement.

```
ods select OptInfo FitStatistics CovParms;
proc glimmix data=cows;
  t2 = tpoint / 100;
  class cow iron infection;
  model weight = iron infection iron*infection tpoint;
  random t2 / type=rsmooth
              subject=cow
              group=iron*infection
              knotmethod=kdtree(bucket=100);
  nloptions tech=newrap;
run;
```

All observations that have the same value combination of the Iron and Infection effects share the same covariance parameter. As a consequence, you obtain different smoothing parameters result in the four groups.

In Output 38.6.10, the "Optimization Information" table shows that there are now four covariance parameters in the optimization, one spline coefficient variance for each group.

Output 38.6.10 Analysis with Group-Specific Smoothing Parameter

The GLIMMIX Procedure

Optimization Information

Optimization Technique	Newton-Raphson
Parameters in Optimization	4
Lower Boundaries	4
Upper Boundaries	0
Fixed Effects	Profiled
Residual Variance	Profiled
Starting From	Data

Fit Statistics

-2 Res Log Likelihood	-1887.95
AIC (smaller is better)	-1877.95
AICC (smaller is better)	-1877.85
BIC (smaller is better)	-1871.66
CAIC (smaller is better)	-1866.66
HQIC (smaller is better)	-1876.14
Generalized Chi-Square	0.48
Gener. Chi-Square / DF	0.00
Radial Smoother df(res)	434.72

Covariance Parameter Estimates

Cov Parm	Subject	Group	Estimate	Standard Error
Var[RSmooth(t2)]	cow	iron*infection 0 0	0.4788	0.1922
Var[RSmooth(t2)]	cow	iron*infection 0 1	0.5152	0.1182
Var[RSmooth(t2)]	cow	iron*infection 1 0	0.4904	0.2195
Var[RSmooth(t2)]	cow	iron*infection 1 1	0.7105	0.1409
Residual			0.000807	0.000060

Varying this variance component by groups has changed the −2 Res Log Likelihood from −1886.29 to −1887.95 (Output 38.6.10). The difference, 1.66, can be viewed (asymptotically) as the realization of a chi-square random variable with three degrees of freedom. The difference is not significant ($p = 0.64586$). The "Covariance Parameter Estimates" table confirms that the estimates of the spline coefficient variance are quite similar in the four groups, ranging from 0.4788 to 0.7105.

Finally, you can apply a different technique for varying the temporal trends among the cows. From Output 38.6.9 it appears that an assumption of parallel trends within groups might be reasonable. In other words, you can fit a model in which the "overall" trend over time in each group is modeled nonparametrically, and this trend is shifted up or down to capture the behavior of the individual cow. You can accomplish this with the following statements:

```
ods select FitStatistics CovParms;
proc glimmix data=cows;
   t2 = tpoint / 100;
   class cow iron infection;
   model weight = iron infection iron*infection tpoint;
   random t2 / type=rsmooth
               subject=iron*infection
               knotmethod=kdtree(bucket=100);
   random intercept / subject=cow;
   output out=gmxout pred(blup)=pred;
   nloptions tech=newrap;
run;
```

There are now two subject effects in this analysis. The first RANDOM statement applies the radial smoothing and identifies the experimental conditions as the subject. For each condition, a separate realization of the random spline coefficients is obtained. The second RANDOM statement adds a random intercept to the trend for each cow. This random intercept results in the parallel shift of the trends over time.

Results from this analysis are shown in Output 38.6.11.

Output 38.6.11 Analysis with Parallel Shifts

The GLIMMIX Procedure

Fit Statistics

-2 Res Log Likelihood	-1788.52
AIC (smaller is better)	-1782.52
AICC (smaller is better)	-1782.48
BIC (smaller is better)	-1788.52
CAIC (smaller is better)	-1785.52
HQIC (smaller is better)	-1788.52
Generalized Chi-Square	1.17
Gener. Chi-Square / DF	0.00
Radial Smoother df(res)	547.21

Output 38.6.11 *continued*

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error
Var[RSmooth(t2)]	iron*infection	0.5398	0.1940
Intercept	cow	0.007122	0.002173
Residual		0.001976	0.000121

Because the parallel shift model is not nested within either one of the previous models, the models cannot be compared with a likelihood ratio test. However, you can draw on the other fit statistics.

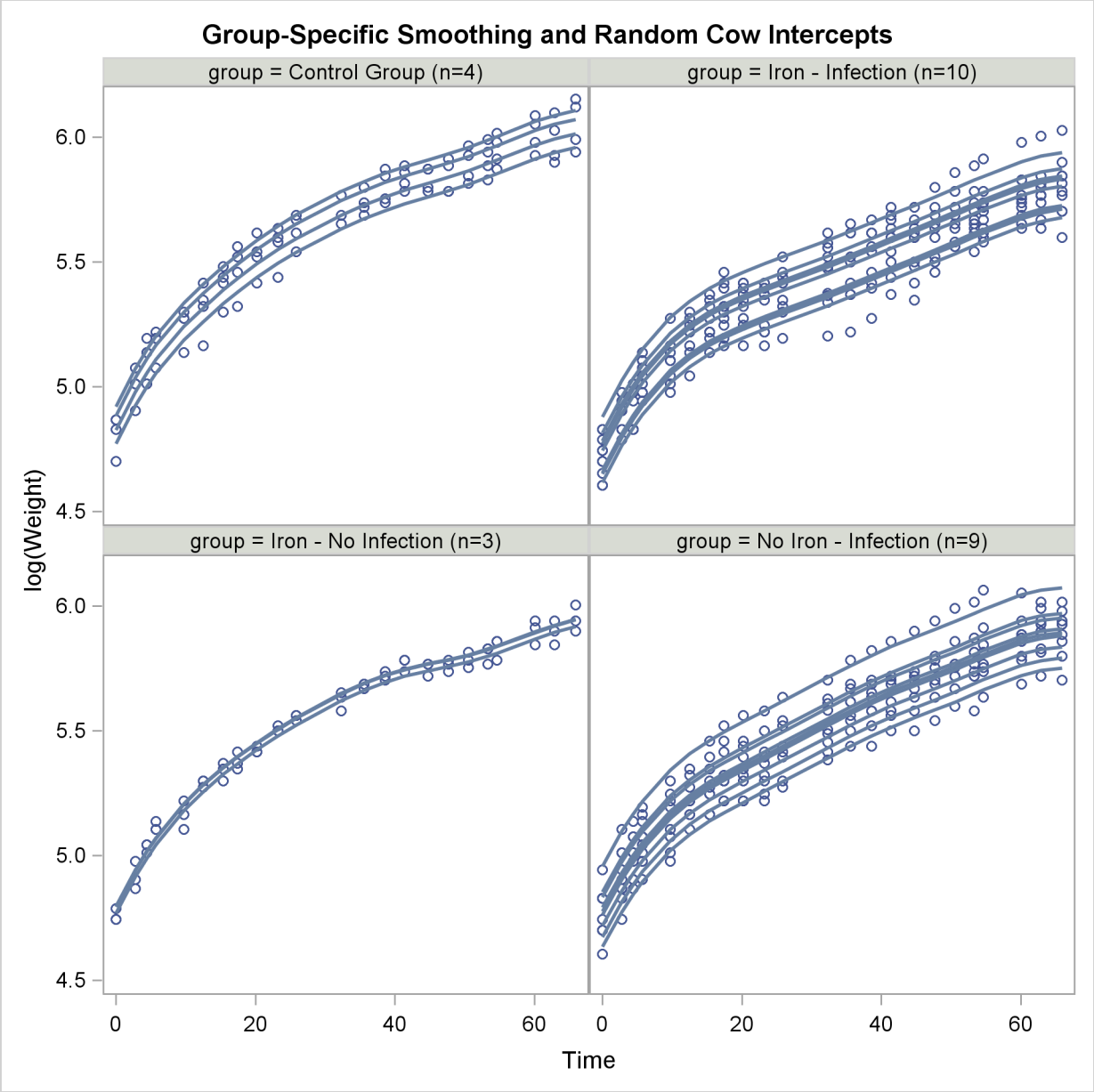
All statistics indicate that this model does not fit the data as well as the initial model that varies the spline coefficients by cow. The Pearson chi-square statistic is more than twice as large as in the previous model, indicating much more residual variation in the fit. On the other hand, this model generates only four sets of spline coefficients, one for each treatment group, and thus retains more residual degrees of freedom.

The "Covariance Parameter Estimates" table in Output 38.6.11 displays the solutions for the covariance parameters. The estimate of the variance of the spline coefficients is not that different from the estimate obtained in the first model (0.5961). The residual variance, however, has more than doubled.

Using similar SAS statements as previously, you can produce a plot of the observed and predicted profiles (Output 38.6.12).

The parallel shifts of the nonparametric smooths are clearly visible in Output 38.6.12. In the groups receiving only iron or only an infection, the parallel lines assumption holds quite well. In the control group and the group receiving iron and the infection, the parallel shift assumption does not hold as well. Two of the profiles in the iron-only group are nearly indistinguishable.

Output 38.6.12 Observed and Predicted Profiles



This example demonstrates that mixed model smoothing techniques can be applied not only to achieve scatter plot smoothing, but also to longitudinal or repeated measures data. You can then use the SUBJECT= option in the RANDOM statement to obtain independent sets of spline coefficients for different subjects, and the GROUP= option in the RANDOM statement to vary the degree of smoothing across groups. Also, radial smoothers can be combined with other random effects. For the data considered here, the appropriate model is one with a single smoothing parameter for all treatment group and cow-specific spline coefficients.

Example 38.7: Isotonic Contrasts for Ordered Alternatives

Dose response studies often focus on testing for monotone increasing or decreasing behavior in the mean values of the dependent variable. Hirotsu and Srivastava (2000) demonstrate one approach by using data that originally appeared in Moriguchi (1976). The data, which follow, consist of ferrite cores subjected to four increasing temperatures. The response variable is the magnetic force of each core.

```
data FerriteCores;
   do Temp = 1 to 4;
      do rep = 1 to 5; drop rep;
         input MagneticForce @@;
         output;
         end;
      end;
datalines;
10.8  9.9 10.7 10.4  9.7
10.7 10.6 11.0 10.8 10.9
11.9 11.2 11.0 11.1 11.3
11.4 10.7 10.9 11.3 11.7
;
```

It is of interest to test whether the magnetic force of the cores rises monotonically with temperature. The approach of Hirotsu and Srivastava (2000) depends on the lower confidence limits of the *isotonic contrasts* of the force means at each temperature, adjusted for multiplicity. The corresponding isotonic contrast compares the average of a particular group and the preceding groups with the average of the succeeding groups. You can compute adjusted confidence intervals for isotonic contrasts by using the LSMESTIMATE statement.

The following statements request an analysis of the FerriteCores data as a one-way design and multiplicity-adjusted lower confidence limits for the isotonic contrasts. For the multiplicity adjustment, the LSMESTIMATE statement employs simulation, which provides adjusted p-values and lower confidence limits that are exact up to Monte Carlo error.

```
proc glimmix data=FerriteCores;
   class Temp;
   model MagneticForce = Temp;
   lsmestimate Temp
        'avg(1:1)<avg(2:4)' -3  1  1  1 divisor=3,
        'avg(1:2)<avg(3:4)' -1 -1  1  1 divisor=2,
        'avg(1:3)<avg(4:4)' -1 -1 -1  3 divisor=3
        / adjust=simulate(seed=1) cl upper;
   ods select LSMestimates;
run;
```

The results are shown in Output 38.7.1.

Output 38.7.1 Analysis of LS-Means with Isotonic Contrasts

The GLIMMIX Procedure

Least Squares Means Estimates
Adjustment for Multiplicity: Simulated

Effect	Label	Estimate	Standard Error	DF	t Value	Pr > t	Adj P
Temp	avg(1:1)<avg(2:4)	0.8000	0.1906	16	4.20	0.0003	0.0010
Temp	avg(1:2)<avg(3:4)	0.7000	0.1651	16	4.24	0.0003	0.0009
Temp	avg(1:3)<avg(4:4)	0.4000	0.1906	16	2.10	0.0260	0.0625

Least Squares Means Estimates
Adjustment for Multiplicity: Simulated

Effect	Label	Alpha	Lower	Upper	Adj Lower	Adj Upper
Temp	avg(1:1)<avg(2:4)	0.05	0.4672	Infty	0.3771	Infty
Temp	avg(1:2)<avg(3:4)	0.05	0.4118	Infty	0.3337	Infty
Temp	avg(1:3)<avg(4:4)	0.05	0.06721	Infty	-0.02291	Infty

With an adjusted p-value of 0.001, the magnetic force at the first temperature is significantly less than the average of the other temperatures. Likewise, the average of the first two temperatures is significantly less than the average of the last two ($p = 0.0009$). However, the magnetic force at the last temperature is not significantly greater than the average magnetic force of the others ($p = 0.0625$). These results indicate a significant monotone increase over the first three temperatures, but not across all four temperatures.

Example 38.8: Adjusted Covariance Matrices of Fixed Effects

The following data are from Pothoff and Roy (1964) and consist of growth measurements for 11 girls and 16 boys at ages 8, 10, 12, and 14. Some of the observations are suspect (for example, the third observation for person 20); however, all of the data are used here for comparison purposes.

```
data pr;
  input child gender$ y1 y2 y3 y4;
  array yy y1-y4;
  do time=1 to 4;
     age = time*2 + 6;
     y   = yy{time};
     output;
  end;
  drop y1-y4;
  datalines;
 1   F   21.0    20.0    21.5    23.0
 2   F   21.0    21.5    24.0    25.5
```

```
 3   F   20.5   24.0   24.5   26.0
 4   F   23.5   24.5   25.0   26.5
 5   F   21.5   23.0   22.5   23.5
 6   F   20.0   21.0   21.0   22.5
 7   F   21.5   22.5   23.0   25.0
 8   F   23.0   23.0   23.5   24.0
 9   F   20.0   21.0   22.0   21.5
10   F   16.5   19.0   19.0   19.5
11   F   24.5   25.0   28.0   28.0
12   M   26.0   25.0   29.0   31.0
13   M   21.5   22.5   23.0   26.5
14   M   23.0   22.5   24.0   27.5
15   M   25.5   27.5   26.5   27.0
16   M   20.0   23.5   22.5   26.0
17   M   24.5   25.5   27.0   28.5
18   M   22.0   22.0   24.5   26.5
19   M   24.0   21.5   24.5   25.5
20   M   23.0   20.5   31.0   26.0
21   M   27.5   28.0   31.0   31.5
22   M   23.0   23.0   23.5   25.0
23   M   21.5   23.5   24.0   28.0
24   M   17.0   24.5   26.0   29.5
25   M   22.5   25.5   25.5   26.0
26   M   23.0   24.5   26.0   30.0
27   M   22.0   21.5   23.5   25.0
;
```

Jennrich and Schluchter (1986) analyze these data with various models for the fixed effects and the covariance structure. The strategy here is to fit a growth curve model for the boys and girls and to account for subject-to-subject variation through G-side random effects. In addition, serial correlation among the observations within each child is accounted for by a time series process. The data are assumed to be Gaussian, and their -2 restricted log likelihood is minimized to estimate the model parameters.

The following statements fit a mixed model in which a separate growth curve is assumed for each gender:

```
proc glimmix data=pr;
   class child gender time;
   model y = gender age gender*age / covb(details) ddfm=kr;
   random intercept age / type=chol sub=child;
   random time / subject=child type=ar(1) residual;
   ods select ModelInfo CovB CovBModelBased CovBDetails;
run;
```

The growth curve for an individual child differs from the gender-specific trend because of a random intercept and a random slope. The two G-side random effects are assumed to be correlated. Their unstructured covariance matrix is parameterized in terms of the Cholesky root to guarantee a positive (semi-)definite estimate. An AR(1) covariance structure is modeled for the observations over time for each child. Notice the RESIDUAL option in the second RANDOM statement. It identifies this as an R-side random effect.

The DDFM=KR option requests that the covariance matrix of the fixed-effect parameter estimates and denominator degrees of freedom for t and F tests are determined according to Kenward and Roger (1997). This is reflected in the "Model Information" table (Output 38.8.1).

Output 38.8.1 Model Information with DDFM=KR

The GLIMMIX Procedure

Model Information

Data Set	WORK.PR
Response Variable	y
Response Distribution	Gaussian
Link Function	Identity
Variance Function	Default
Variance Matrix Blocked By	child
Estimation Technique	Restricted Maximum Likelihood
Degrees of Freedom Method	Kenward-Roger
Fixed Effects SE Adjustment	Kenward-Roger

The COVB option in the MODEL statement requests that the covariance matrix used for inference about fixed effects in this model is displayed; this is the Kenward-Roger-adjusted covariance matrix. The DETAILS suboption requests that the unadjusted covariance matrix is also displayed (Output 38.8.2). In addition, a table of diagnostic measures for the covariance matrices is produced.

Output 38.8.2 Model-Based and Adjusted Covariance Matrix

Model Based Covariance Matrix for Fixed Effects (Unadjusted)

Effect	gender	Row	Col1	Col2	Col3	Col4	Col5	Col6
Intercept		1	0.9969	-0.9969		-0.07620	0.07620	
gender	F	2	-0.9969	2.4470		0.07620	-0.1870	
gender	M	3						
age		4	-0.07620	0.07620		0.007581	-0.00758	
age*gender	F	5	0.07620	-0.1870		-0.00758	0.01861	
age*gender	M	6						

Covariance Matrix for Fixed Effects

Effect	gender	Row	Col1	Col2	Col3	Col4	Col5	Col6
Intercept		1	0.9724	-0.9724		-0.07412	0.07412	
gender	F	2	-0.9724	2.3868		0.07412	-0.1819	
gender	M	3						
age		4	-0.07412	0.07412		0.007256	-0.00726	
age*gender	F	5	0.07412	-0.1819		-0.00726	0.01781	
age*gender	M	6						

Output 38.8.2 *continued*

Diagnostics for Covariance Matrices of Fixed Effects

		Model-Based	Adjusted
Dimensions	Rows	6	6
	Non-zero entries	16	16
Summaries	Trace	3.4701	3.3843
	Log determinant	-11.95	-12.17
Eigenvalues	> 0	4	4
	= 0	2	2
	max abs	2.972	2.8988
	min abs non-zero	0.0009	0.0008
	Condition number	3467.8	3698.2
Norms	Frobenius	3.0124	2.9382
	Infinity	3.7072	3.6153
Comparisons	Concordance correlation		0.9979
	Discrepancy function		0.0084
	Frobenius norm of difference		0.0742
	Trace(Adjusted Inv(MBased))		3.7801

Determinant and inversion results apply to the nonsingular partitions of the covariance matrices.

The "Diagnostics for Covariance Matrices" table in Output 38.8.2 consists of several sections. The trace and log determinant of covariance matrices are general scalar summaries that are sometimes used in direct comparisons, or in formulating further statistics, such as the difference of log determinants. The trace simply represents the sum of the variances of all fixed-effects parameters.

The two matrices have the same number of positive and zero eigenvalues; hence they are of the same rank. There are no negative eigenvalues; hence the matrices are positive semi-definite.

The "Comparisons" section of the table provides several statistics that set the matrices in relationship. The statistics enable you to assess the extent to which the adjustment affected the model-based matrix. If the two matrices are identical, the concordance correlation equals 1, the discrepancy function and the Frobenius norm of the differences equal 0, and the trace of the adjusted and the (generalized) inverse of the model-based matrix equals the rank. See the section "Exploring and Comparing Covariance Matrices" on page 2246 for computational details regarding these statistics. With increasing discrepancy between the matrices, the difference norm and discrepancy function increase, the concordance correlation falls below 1, and the trace deviates from the rank. In this particular example, there is strong agreement between the two matrices; the adjustment to the covariance matrix associated with DDFM=KR is only slight. It is noteworthy, however, that the trace of the adjusted covariance matrix falls short of the trace of the unadjusted one. Indeed, from Output 38.8.2 you can see that the diagonal elements of the adjusted covariance matrices are uniformly smaller than those of the model-based covariance matrix.

Standard error "shrinkage" for the Kenward-Roger covariance adjustment is due to the term $-0.25\mathbf{R}_{ij}$ in equation (3) of Kenward and Roger (1997), which is nonzero for covariance structures with second derivatives, such as the TYPE=ANTE(1), TYPE=AR(1), TYPE=ARH(1), TYPE=ARMA(1,1), TYPE=CHOL, TYPE=CSH, TYPE=FA0(q), TYPE=TOEPH, and TYPE=UNR structures and all TYPE=SP() structures.

For covariance structures that are linear in the parameters, $\mathbf{R}_{ij} = \mathbf{0}$. You can add the FIRSTORDER suboption to the DDFM=KR option to request that second derivative matrices $\mathbf{R}_{ij}$ are excluded from computing the covariance matrix adjustment. The resulting covariance adjustment is that of Kackar and Harville (1984) and Harville and Jeske (1992). This estimator is denoted as $\widetilde{m}^{@}$ in Harville and Jeske (1992) and is referred to there as the Prasad-Rao estimator after related work by Prasad and Rao (1990). This standard error adjustment is guaranteed to be positive (semi-)definite. The following statements fit the model with the Kackar-Harville-Jeske estimator and compare model-based and adjusted covariance matrices:

```
proc glimmix data=pr;
   class child gender time;
   model y = gender age gender*age / covb(details)
                                     ddfm=kr(firstorder);
   random intercept age / type=chol sub=child;
   random time / subject=child type=ar(1) residual;
   ods select ModelInfo CovB CovBDetails;
run;
```

The standard error adjustment is reflected in the "Model Information" table (Output 38.8.3).

Output 38.8.3 Model Information with DDFM=KR(FIRSTORDER)

The GLIMMIX Procedure

Model Information

Data Set	WORK.PR
Response Variable	y
Response Distribution	Gaussian
Link Function	Identity
Variance Function	Default
Variance Matrix Blocked By	child
Estimation Technique	Restricted Maximum Likelihood
Degrees of Freedom Method	Kenward-Roger
Fixed Effects SE Adjustment	Prasad-Rao-Kackar-Harville-Jeske

Output 38.8.4 displays the adjusted covariance matrix. Notice that the elements of this matrix, in particular the diagonal elements, are larger in absolute value than those of the model-based estimator (Output 38.8.2).

Output 38.8.4 Adjusted Covariance Matrix and Comparison to Model-Based Estimator

Covariance Matrix for Fixed Effects

Effect	gender	Row	Col1	Col2	Col3	Col4	Col5	Col6
Intercept		1	1.0122	-1.0122		-0.07758	0.07758	
gender	F	2	-1.0122	2.4845		0.07758	-0.1904	
gender	M	3						
age		4	-0.07758	0.07758		0.007706	-0.00771	
age*gender	F	5	0.07758	-0.1904		-0.00771	0.01891	
age*gender	M	6						

Diagnostics for Covariance Matrices of Fixed Effects

		Model-Based	Adjusted
Dimensions	Rows	6	6
	Non-zero entries	16	16
Summaries	Trace	3.4701	3.5234
	Log determinant	-11.95	-11.91
Eigenvalues	> 0	4	4
	= 0	2	2
	max abs	2.972	3.0176
	min abs non-zero	0.0009	0.0009
	Condition number	3467.8	3513.4
Norms	Frobenius	3.0124	3.0587
	Infinity	3.7072	3.7647
Comparisons	Concordance correlation		0.9999
	Discrepancy function		0.0003
	Frobenius norm of difference		0.0463
	Trace(Adjusted Inv(MBased))		4.0352

Determinant and inversion results apply to the nonsingular partitions of the covariance matrices.

The "Comparisons" statistics show that the model-based and adjusted covariance matrix of the fixed-effects parameter estimates are very similar. The concordance correlation is near 1, the discrepancy is near zero, and the trace is very close to the number of positive eigenvalues. This is due to the balanced nature of these repeated measures data. Shrinkage of standard errors, however, can not occur with the Kackar-Harville-Jeske estimator.

Example 38.9: Testing Equality of Covariance and Correlation Matrices

Fisher's iris data are widely used in multivariate statistics. They comprise measurements in millimeters of four flower attributes, the length and width of sepals and petals for 50 specimens from each of three species, *Iris setosa*, *I. versicolor*, and *I. virginica* (Fisher 1936).

When modeling multiple attributes from the same specimen, correlations among measurements from the same flower must be taken into account. Unstructured covariance matrices are common in this multivariate setting. Species comparisons can focus on comparisons of mean response, but comparisons of the variation and covariation are also of interest. In this example, the equivalence of covariance and correlation matrices among the species are examined.

The following DATA step creates the iris data in multivariate format—that is, each observation contains multiple response variables. The subsequent DATA step creates a data set in univariate form, where each observation corresponds to a single response variable. This is the form needed by the GLIMMIX procedure.

```
proc format;
   value specname
      1='Setosa    '
      2='Versicolor'
      3='Virginica ';
   value variable
      1='Sepal Length'
      2='Sepal Width '
      3='Petal Length'
      4='Petal Width';
run;

data iris;
   input Y1 Y2 Y3 Y4 Species @@;
   format Species specname.;
   label Y1 = 'Sepal Length (mm)'
         Y2 = 'Sepal Width (mm)'
         Y3 = 'Petal Length (mm)'
         Y4 = 'Petal Width (mm)';
   datalines;
50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3
63 28 51 15 3 46 34 14 03 1 69 31 51 23 3 62 22 45 15 2
59 32 48 18 2 46 36 10 02 1 61 30 46 14 2 60 27 51 16 2
65 30 52 20 3 56 25 39 11 2 65 30 55 18 3 58 27 51 19 3
68 32 59 23 3 51 33 17 05 1 57 28 45 13 2 62 34 54 23 3
77 38 67 22 3 63 33 47 16 2 67 33 57 25 3 76 30 66 21 3
49 25 45 17 3 55 35 13 02 1 67 30 52 23 3 70 32 47 14 2
64 32 45 15 2 61 28 40 13 2 48 31 16 02 1 59 30 51 18 3
55 24 38 11 2 63 25 50 19 3 64 32 53 23 3 52 34 14 02 1
49 36 14 01 1 54 30 45 15 2 79 38 64 20 3 44 32 13 02 1
67 33 57 21 3 50 35 16 06 1 58 26 40 12 2 44 30 13 02 1
77 28 67 20 3 63 27 49 18 3 47 32 16 02 1 55 26 44 12 2
50 23 33 10 2 72 32 60 18 3 48 30 14 03 1 51 38 16 02 1
61 30 49 18 3 48 34 19 02 1 50 30 16 02 1 50 32 12 02 1
```

```
61 26 56 14 3 64 28 56 21 3 43 30 11 01 1 58 40 12 02 1
51 38 19 04 1 67 31 44 14 2 62 28 48 18 3 49 30 14 02 1
51 35 14 02 1 56 30 45 15 2 58 27 41 10 2 50 34 16 04 1
46 32 14 02 1 60 29 45 15 2 57 26 35 10 2 57 44 15 04 1
50 36 14 02 1 77 30 61 23 3 63 34 56 24 3 58 27 51 19 3
57 29 42 13 2 72 30 58 16 3 54 34 15 04 1 52 41 15 01 1
71 30 59 21 3 64 31 55 18 3 60 30 48 18 3 63 29 56 18 3
49 24 33 10 2 56 27 42 13 2 57 30 42 12 2 55 42 14 02 1
49 31 15 02 1 77 26 69 23 3 60 22 50 15 3 54 39 17 04 1
66 29 46 13 2 52 27 39 14 2 60 34 45 16 2 50 34 15 02 1
44 29 14 02 1 50 20 35 10 2 55 24 37 10 2 58 27 39 12 2
47 32 13 02 1 46 31 15 02 1 69 32 57 23 3 62 29 43 13 2
74 28 61 19 3 59 30 42 15 2 51 34 15 02 1 50 35 13 03 1
56 28 49 20 3 60 22 40 10 2 73 29 63 18 3 67 25 58 18 3
49 31 15 01 1 67 31 47 15 2 63 23 44 13 2 54 37 15 02 1
56 30 41 13 2 63 25 49 15 2 61 28 47 12 2 64 29 43 13 2
51 25 30 11 2 57 28 41 13 2 65 30 58 22 3 69 31 54 21 3
54 39 13 04 1 51 35 14 03 1 72 36 61 25 3 65 32 51 20 3
61 29 47 14 2 56 29 36 13 2 69 31 49 15 2 64 27 53 19 3
68 30 55 21 3 55 25 40 13 2 48 34 16 02 1 48 30 14 01 1
45 23 13 03 1 57 25 50 20 3 57 38 17 03 1 51 38 15 03 1
55 23 40 13 2 66 30 44 14 2 68 28 48 14 2 54 34 17 02 1
51 37 15 04 1 52 35 15 02 1 58 28 51 24 3 67 30 50 17 2
63 33 60 25 3 53 37 15 02 1
;

data iris_univ;
   set iris;
   retain id 0;
   array y{4};
   format var variable.;
   id+1;
   do var=1 to 4;
      response = y{var};
      output;
   end;
   drop y:;
run;
```

The following GLIMMIX statements fit a model with separate unstructured covariance matrices for each species:

```
ods select FitStatistics CovParms CovTests;
proc glimmix data=iris_univ;
   class species var id;
   model response = species*var;
   random _residual_ / type=un group=species subject=id;
   covtest homogeneity;
run;
```

The mean function is modeled as a cell-means model that allows for different means for each species and outcome variable. The covariances are modeled directly (R-side) rather than through random effects. The ID variable identifies the individual plant, so that responses from different plants are independent. The GROUP=SPECIES option varies the parameters of the unstructured covariance matrix by species. Hence, this model has 30 covariance parameters: 10 unique parameters for a (4×4) covariance matrix for each of three species.

The COVTEST statement requests a test of homogeneity—that is, it tests whether varying the covariance parameters by the group effect provides a significantly better fit compared to a model in which different groups share the same parameter.

Output 38.9.1 Fit Statistics for Analysis of Fisher's Iris Data

The GLIMMIX Procedure

Fit Statistics

-2 Res Log Likelihood	2812.89
AIC (smaller is better)	2872.89
AICC (smaller is better)	2876.23
BIC (smaller is better)	2963.21
CAIC (smaller is better)	2993.21
HQIC (smaller is better)	2909.58
Generalized Chi-Square	588.00
Gener. Chi-Square / DF	1.00

The "Fit Statistics" table shows the -2 restricted (residual) log likelihood in the full model and other fit statistics (Output 38.9.1). The "-2 Res Log Likelihood" sets the benchmark against which a model with homogeneity constraint is compared. Output 38.9.2 displays the 30 covariance parameters in this model.

There appear to be substantial differences among the covariance parameters from different groups. For example, the residual variability of the petal length of the three species is 12.4249, 26.6433, and 40.4343, respectively. The homogeneity hypothesis restricts these variances to be equal and similarly for the other covariance parameters. The results from the COVTEST statement are shown in Output 38.9.3.

Output 38.9.2 Covariance Parameters Varied by Species (TYPE=UN)

Covariance Parameter Estimates

Cov Parm	Subject	Group	Estimate	Standard Error
UN(1,1)	id	Species Setosa	12.4249	2.5102
UN(2,1)	id	Species Setosa	9.9216	2.3775
UN(2,2)	id	Species Setosa	14.3690	2.9030
UN(3,1)	id	Species Setosa	1.6355	0.9052
UN(3,2)	id	Species Setosa	1.1698	0.9552
UN(3,3)	id	Species Setosa	3.0159	0.6093
UN(4,1)	id	Species Setosa	1.0331	0.5508
UN(4,2)	id	Species Setosa	0.9298	0.5859
UN(4,3)	id	Species Setosa	0.6069	0.2755
UN(4,4)	id	Species Setosa	1.1106	0.2244
UN(1,1)	id	Species Versicolor	26.6433	5.3828
UN(2,1)	id	Species Versicolor	8.5184	2.6144
UN(2,2)	id	Species Versicolor	9.8469	1.9894
UN(3,1)	id	Species Versicolor	18.2898	4.3398
UN(3,2)	id	Species Versicolor	8.2653	2.4149
UN(3,3)	id	Species Versicolor	22.0816	4.4612
UN(4,1)	id	Species Versicolor	5.5780	1.6617
UN(4,2)	id	Species Versicolor	4.1204	1.0641
UN(4,3)	id	Species Versicolor	7.3102	1.6891
UN(4,4)	id	Species Versicolor	3.9106	0.7901
UN(1,1)	id	Species Virginica	40.4343	8.1690
UN(2,1)	id	Species Virginica	9.3763	3.2213
UN(2,2)	id	Species Virginica	10.4004	2.1012
UN(3,1)	id	Species Virginica	30.3290	6.6262
UN(3,2)	id	Species Virginica	7.1380	2.7395
UN(3,3)	id	Species Virginica	30.4588	6.1536
UN(4,1)	id	Species Virginica	4.9094	2.5916
UN(4,2)	id	Species Virginica	4.7629	1.4367
UN(4,3)	id	Species Virginica	4.8824	2.2750
UN(4,4)	id	Species Virginica	7.5433	1.5240

Output 38.9.3 Likelihood Ratio Test of Homogeneity

Tests of Covariance Parameters
Based on the Restricted Likelihood

Label	DF	−2 Res Log Like	ChiSq	Pr > ChiSq	Note
Homogeneity	20	2959.55	146.66	<.0001	DF

DF: P-value based on a chi-square with DF degrees of freedom.

Denote as $\mathbf{R}_k$ the covariance matrix for species $k = 1, 2, 3$ with elements σ_{ijk}. In processing the COVTEST hypothesis H_0: $\mathbf{R}_1 = \mathbf{R}_2 = \mathbf{R}_3$, the GLIMMIX procedure fits a model that satistfies the constraints

$$\begin{aligned}
\sigma_{111} &= \sigma_{112} = \sigma_{113} \\
\sigma_{211} &= \sigma_{212} = \sigma_{213} \\
\sigma_{231} &= \sigma_{232} = \sigma_{233} \\
&\vdots \\
\sigma_{441} &= \sigma_{442} = \sigma_{443}
\end{aligned}$$

where σ_{ijk} is the covariance between the ith and jth variable for the kth species. The -2 restricted log likelihood of this restricted model is 2959.55 (Output 38.9.3). The change of 146.66 compared to the full model is highly significant. There is sufficient evidence to reject the notion of equal covariance matrices among the three iris species.

Equality of covariance matrices implies equality of correlation matrices, but the reverse is not true. Fewer constraints are needed to equate correlations because the diagonal entries of the covariance matrices are free to vary. In order to test the equality of the correlation matrices among the three species, you can parameterize the unstructured covariance matrix in terms of the correlations and use a COVTEST statement with general contrasts, as shown in the following statements:

```
ods select FitStatistics CovParms CovTests;
proc glimmix data=iris_univ;
   class species var id;
   model response = species*var;
   random _residual_ / type=unr group=species subject=id;
   covtest 'Equal Covariance Matrices'  homogeneity;
   covtest 'Equal Correlation Matrices' general
        0  0  0  0  1  0  0  0  0  0
        0  0  0  0 -1  0  0  0  0  0,
        0  0  0  0  1  0  0  0  0  0
        0  0  0  0  0  0  0  0  0  0
        0  0  0  0 -1  0  0  0  0  0,
        0  0  0  0  0  1  0  0  0  0
        0  0  0  0  0 -1  0  0  0  0,
        0  0  0  0  0  1  0  0  0  0
        0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0 -1  0  0  0  0,
        0  0  0  0  0  0  1  0  0  0
        0  0  0  0  0  0 -1  0  0  0,
        0  0  0  0  0  0  1  0  0  0
        0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0 -1  0  0  0,
        0  0  0  0  0  0  0  1  0  0
        0  0  0  0  0  0  0 -1  0  0,
        0  0  0  0  0  0  0  1  0  0
        0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0 -1  0  0,
        0  0  0  0  0  0  0  0  1  0
        0  0  0  0  0  0  0  0 -1  0,
        0  0  0  0  0  0  0  0  1  0
```

```
        0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  -1 0,
        0  0  0  0  0  0  0  0  0  1
        0  0  0  0  0  0  0  0  0 -1,
        0  0  0  0  0  0  0  0  0  1
        0  0  0  0  0  0  0  0  0  0
        0  0  0  0  0  0  0  0  0 -1 / estimates;
run;
```

The TYPE=UNR structure is a reparameterization of TYPE=UN. The models provide the same fit, as seen by comparison of the "Fit Statistics" tables in Output 38.9.1 and Output 38.9.4. The covariance parameters are ordered differently, however. In each group, the four variances precede the six correlations (Output 38.9.4). The first COVTEST statement tests the homogeneity hypothesis in terms of the UNR parameterization, and the result is identical to the test in Output 38.9.3. The second COVTEST statement restricts the correlations to be equal across groups. If ρ_{ijk} is the correlation between the ith and jth variable for the kth species, the 12 restrictions are

$$\rho_{211} = \rho_{212} = \rho_{213}$$
$$\rho_{311} = \rho_{312} = \rho_{313}$$
$$\rho_{321} = \rho_{322} = \rho_{323}$$
$$\rho_{411} = \rho_{412} = \rho_{413}$$
$$\rho_{421} = \rho_{422} = \rho_{423}$$
$$\rho_{431} = \rho_{432} = \rho_{433}$$

The ESTIMATES option in the COVTEST statement requests that the GLIMMIX procedure display the covariance parameter estimates in the restricted model (Output 38.9.4).

Output 38.9.4 Fit Statistics, Covariance Parameters (TYPE=UNR), and Likelihood Ratio Tests for Equality of Covariance and Correlation Matrices

```
                    The GLIMMIX Procedure

                        Fit Statistics

                -2 Res Log Likelihood        2812.89
                AIC  (smaller is better)     2872.89
                AICC (smaller is better)     2876.23
                BIC  (smaller is better)     2963.21
                CAIC (smaller is better)     2993.21
                HQIC (smaller is better)     2909.58
                Generalized Chi-Square        588.00
                Gener. Chi-Square / DF          1.00
```

Output 38.9.4 *continued*

Covariance Parameter Estimates

Cov Parm	Subject	Group	Estimate	Standard Error
Var(1)	id	Species Setosa	12.4249	2.5102
Var(2)	id	Species Setosa	14.3690	2.9030
Var(3)	id	Species Setosa	3.0159	0.6093
Var(4)	id	Species Setosa	1.1106	0.2244
Corr(2,1)	id	Species Setosa	0.7425	0.06409
Corr(3,1)	id	Species Setosa	0.2672	0.1327
Corr(3,2)	id	Species Setosa	0.1777	0.1383
Corr(4,1)	id	Species Setosa	0.2781	0.1318
Corr(4,2)	id	Species Setosa	0.2328	0.1351
Corr(4,3)	id	Species Setosa	0.3316	0.1271
Var(1)	id	Species Versicolor	26.6433	5.3828
Var(2)	id	Species Versicolor	9.8469	1.9894
Var(3)	id	Species Versicolor	22.0816	4.4612
Var(4)	id	Species Versicolor	3.9106	0.7901
Corr(2,1)	id	Species Versicolor	0.5259	0.1033
Corr(3,1)	id	Species Versicolor	0.7540	0.06163
Corr(3,2)	id	Species Versicolor	0.5605	0.09797
Corr(4,1)	id	Species Versicolor	0.5465	0.1002
Corr(4,2)	id	Species Versicolor	0.6640	0.07987
Corr(4,3)	id	Species Versicolor	0.7867	0.05445
Var(1)	id	Species Virginica	40.4343	8.1690
Var(2)	id	Species Virginica	10.4004	2.1012
Var(3)	id	Species Virginica	30.4588	6.1536
Var(4)	id	Species Virginica	7.5433	1.5240
Corr(2,1)	id	Species Virginica	0.4572	0.1130
Corr(3,1)	id	Species Virginica	0.8642	0.03616
Corr(3,2)	id	Species Virginica	0.4010	0.1199
Corr(4,1)	id	Species Virginica	0.2811	0.1316
Corr(4,2)	id	Species Virginica	0.5377	0.1015
Corr(4,3)	id	Species Virginica	0.3221	0.1280

Output 38.9.4 *continued*

Tests of Covariance Parameters
Based on the Restricted Likelihood

Label	DF	-2 Res Log Like	ChiSq	Pr > ChiSq	Estimates H0 Est1
Equal Covariance Matrices	20	2959.55	146.66	<.0001	26.5
Equal Correlation Matrices	12	2876.38	63.49	<.0001	16.5

Tests of Covariance Parameters
Based on the Restricted Likelihood

Label	Estimates H0 Est2	Est3	Est4	Est5	Est6	Est7	Est8
Equal Covariance Matrices	11.5	18.52	4.19	0.530	0.756	0.378	0.365
Equal Correlation Matrices	14.9	4.84	1.44	0.561	0.683	0.402	0.384

Tests of Covariance Parameters
Based on the Restricted Likelihood

Label	Estimates H0 Est9	Est10	Est11	Est12	Est13	Est14	Est15
Equal Covariance Matrices	0.471	0.484	26.5	11.54	18.5	4.19	0.530
Equal Correlation Matrices	0.498	0.522	24.4	9.16	17.4	3.00	0.561

Tests of Covariance Parameters
Based on the Restricted Likelihood

Label	Estimates H0 Est16	Est17	Est18	Est19	Est20	Est21	Est22
Equal Covariance Matrices	0.756	0.378	0.365	0.471	0.484	26.5	11.5
Equal Correlation Matrices	0.683	0.402	0.384	0.498	0.522	35.1	10.8

Tests of Covariance Parameters
Based on the Restricted Likelihood

Label	Estimates H0 Est23	Est24	Est25	Est26	Est27	Est28	Est29
Equal Covariance Matrices	18.5	4.19	0.530	0.756	0.378	0.365	0.471
Equal Correlation Matrices	27.4	8.14	0.561	0.683	0.402	0.384	0.498

Tests of Covariance Parameters
Based on the Restricted Likelihood

Label	Estimates H0 Est30	Note
Equal Covariance Matrices	0.484	DF
Equal Correlation Matrices	0.522	DF

DF: P-value based on a chi-square with DF degrees of freedom.

The result of the homogeneity test is identical to that in Output 38.9.3. The hypothesis of equality of the correlation matrices is also rejected with a chi-square value of 63.49 and a p-value of < 0.0001. Notice, however, that the chi-square statistic is smaller than in the test of homogeneity due to the smaller number of restrictions imposed on the full model. The estimate of the common correlation matrix in the restricted model is

$$\begin{bmatrix} 1 & 0.561 & 0.683 & 0.384 \\ 0.561 & 1 & 0.402 & 0.498 \\ 0.683 & 0.402 & 1 & 0.522 \\ 0.384 & 0.498 & 0.522 & 1 \end{bmatrix}$$

Example 38.10: Multiple Trends Correspond to Multiple Extrema in Profile Likelihoods

The following data set contains observations for a period of 168 months for the "Southern Oscillation Index," measurements of monthly averaged atmospheric pressure differences between Easter Island and Darwin, Australia (Kahaner, Moler, and Nash 1989, Ch. 11.9; National Institute of Standards and Technology 1998). These data are also used as an example in Chapter 50, "The LOESS Procedure," in the *SAS/STAT User's Guide*.

```
data ENSO;
  input Pressure @@;
  Year = _n_ / 12;
datalines;
 12.9  11.3  10.6  11.2  10.9   7.5   7.7  11.7
 12.9  14.3  10.9  13.7  17.1  14.0  15.3   8.5
  5.7   5.5   7.6   8.6   7.3   7.6  12.7  11.0
 12.7  12.9  13.0  10.9  10.4  10.2   8.0  10.9
 13.6  10.5   9.2  12.4  12.7  13.3  10.1   7.8
  4.8   3.0   2.5   6.3   9.7  11.6   8.6  12.4
 10.5  13.3  10.4   8.1   3.7  10.7   5.1  10.4
 10.9  11.7  11.4  13.7  14.1  14.0  12.5   6.3
  9.6  11.7   5.0  10.8  12.7  10.8  11.8  12.6
 15.7  12.6  14.8   7.8   7.1  11.2   8.1   6.4
  5.2  12.0  10.2  12.7  10.2  14.7  12.2   7.1
  5.7   6.7   3.9   8.5   8.3  10.8  16.7  12.6
 12.5  12.5   9.8   7.2   4.1  10.6  10.1  10.1
 11.9  13.6  16.3  17.6  15.5  16.0  15.2  11.2
 14.3  14.5   8.5  12.0  12.7  11.3  14.5  15.1
 10.4  11.5  13.4   7.5   0.6   0.3   5.5   5.0
  4.6   8.2   9.9   9.2  12.5  10.9   9.9   8.9
  7.6   9.5   8.4  10.7  13.6  13.7  13.7  16.5
 16.8  17.1  15.4   9.5   6.1  10.1   9.3   5.3
 11.2  16.6  15.6  12.0  11.5   8.6  13.8   8.7
  8.6   8.6   8.7  12.8  13.2  14.0  13.4  14.8
;
```

Differences in atmospheric pressure create wind, and the differences recorded in the data set ENSO drive the trade winds in the southern hemisphere. Such time series often do not consist of a single trend or cycle. In this particular case, there are at least two known cycles that reflect the annual

weather pattern and a longer cycle that represents the periodic warming of the Pacific Ocean (El Niño).

To estimate the trend in these data by using mixed model technology, you can apply a mixed model smoothing technique such as TYPE=RSMOOTH or TYPE=PSPLINE. The following statements fit a radial smoother to the ENSO data and obtain profile likelihoods for a series of values for the variance of the random spline coefficients:

```
data tdata;
  do covp1=0,0.0005,0.05,0.1,0.2,0.5,
           1,2,3,4,5,6,8,10,15,20,50,
           75,100,125,140,150,160,175,
           200,225,250,275,300,350;
     output;
  end;
run;

ods select FitStatistics CovParms CovTests;
proc glimmix data=enso noprofile;
   model pressure = year;
   random year / type=rsmooth knotmethod=equal(50);
   parms (2) (10);
   covtest tdata=tdata / parms;
   ods output covtests=ct;
run;
```

The tdata data set contains value for the variance of the radial smoother variance for which the profile likelihood of the model is to be computed. The profile likelihood is obtained by setting the radial smoother variance at the specified value and estimating all other parameters subject to that constraint.

Because the model contains a residual variance and you need to specify nonzero values for the first covariance parameter, the NOPROFILE is added to the PROC GLIMMIX statements. If the residual variance is profiled from the estimation, you cannot fix covariance parameters at a given value, because they would be reexpressed during model fitting in terms of ratios with the profiled (and changing) variance.

The PARMS statement determines starting values for the covariance parameters for fitting the (full) model. The PARMS option in the COVTEST statement requests that the input parameters be added to the output and the output data set. This is useful for subsequent plotting of the profile likelihood function.

The "Fit Statistics" table displays the −2 restricted log likelihood of the model (897.76, Output 38.10.1). The estimate of the variance of the radial smoother coefficients is 3.5719.

The "Test of Covariance Parameters" table displays the −2 restricted log likelihood for each observation in the tdata set. Because the tdata data set specifies values for only the first covariance parameter, the second covariance parameter is free to vary and the values for −2 Res Log Like are profile likelihoods. Notice that for a number of values of CovP1 the chi-square statistic is missing in this table. For these values the −2 Res Log Like is *smaller* than that of the full model. The model did not converge to a global minimum of the negative restricted log likelihood.

Output 38.10.1 REML and Profile Likelihood Analysis

The GLIMMIX Procedure

Fit Statistics

−2 Res Log Likelihood	897.76
AIC (smaller is better)	901.76
AICC (smaller is better)	901.83
BIC (smaller is better)	897.76
CAIC (smaller is better)	899.76
HQIC (smaller is better)	897.76
Generalized Chi-Square	1554.38
Gener. Chi-Square / DF	9.36
Radial Smoother df(res)	153.52

Covariance Parameter Estimates

Cov Parm	Estimate	Standard Error
Var[RSmooth(Year)]	3.5719	3.7672
Residual	9.3638	1.3014

Output 38.10.1 *continued*

Tests of Covariance Parameters
Based on the Restricted Likelihood

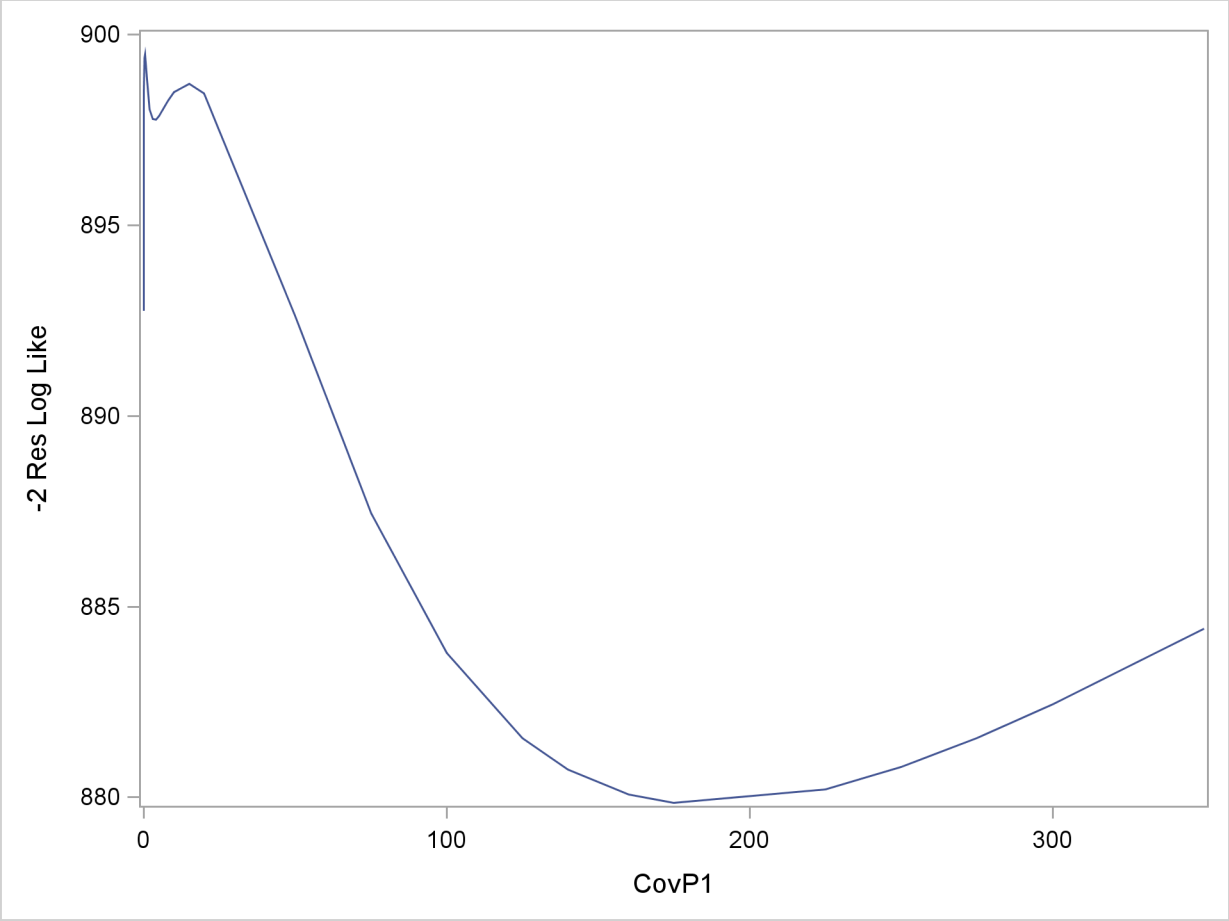
Label	DF	−2 Res Log Like	ChiSq	Pr > ChiSq	------Input Parameters---- CovP1	CovP2	Note
WORK.TDATA	1	893.01	.	1.0000	0.0000	9.36	MI
WORK.TDATA	1	892.76	.	1.0000	0.0005	9.36	DF
WORK.TDATA	1	897.34	.	1.0000	0.0500	9.36	DF
WORK.TDATA	1	898.53	0.77	0.3816	0.1000	9.36	DF
WORK.TDATA	1	899.38	1.62	0.2038	0.2000	9.36	DF
WORK.TDATA	1	899.49	1.73	0.1888	0.5000	9.36	DF
WORK.TDATA	1	898.83	1.07	0.3016	1.0000	9.36	DF
WORK.TDATA	1	898.04	0.28	0.5967	2.0000	9.36	DF
WORK.TDATA	1	897.79	0.03	0.8693	3.0000	9.36	DF
WORK.TDATA	1	897.77	0.01	0.9145	4.0000	9.36	DF
WORK.TDATA	1	897.86	0.10	0.7517	5.0000	9.36	DF
WORK.TDATA	1	897.99	0.23	0.6311	6.0000	9.36	DF
WORK.TDATA	1	898.27	0.51	0.4761	8.0000	9.36	DF
WORK.TDATA	1	898.49	0.73	0.3919	10.0000	9.36	DF
WORK.TDATA	1	898.70	0.94	0.3318	15.0000	9.36	DF
WORK.TDATA	1	898.45	0.69	0.4068	20.0000	9.36	DF
WORK.TDATA	1	892.63	.	1.0000	50.0000	9.36	DF
WORK.TDATA	1	887.44	.	1.0000	75.0000	9.36	DF
WORK.TDATA	1	883.79	.	1.0000	100.0000	9.36	DF
WORK.TDATA	1	881.55	.	1.0000	125.0000	9.36	DF
WORK.TDATA	1	880.72	.	1.0000	140.0000	9.36	DF
WORK.TDATA	1	.	.	.	150.0000	9.36	
WORK.TDATA	1	880.07	.	1.0000	160.0000	9.36	DF
WORK.TDATA	1	879.85	.	1.0000	175.0000	9.36	DF
WORK.TDATA	1	.	.	.	200.0000	9.36	
WORK.TDATA	1	880.21	.	1.0000	225.0000	9.36	DF
WORK.TDATA	1	880.80	.	1.0000	250.0000	9.36	DF
WORK.TDATA	1	881.56	.	1.0000	275.0000	9.36	DF
WORK.TDATA	1	882.44	.	1.0000	300.0000	9.36	DF
WORK.TDATA	1	884.41	.	1.0000	350.0000	9.36	DF

DF: P-value based on a chi-square with DF degrees of freedom.
MI: P-value based on a mixture of chi-squares.

The following statements plot the −2 restricted profile log likelihood (Output 38.10.2):

```
proc sgplot data=ct;
   series y=objective x=covp1;
run;
```

Output 38.10.2 −2 Restricted Profile Log Likelihood for Smoothing Variance



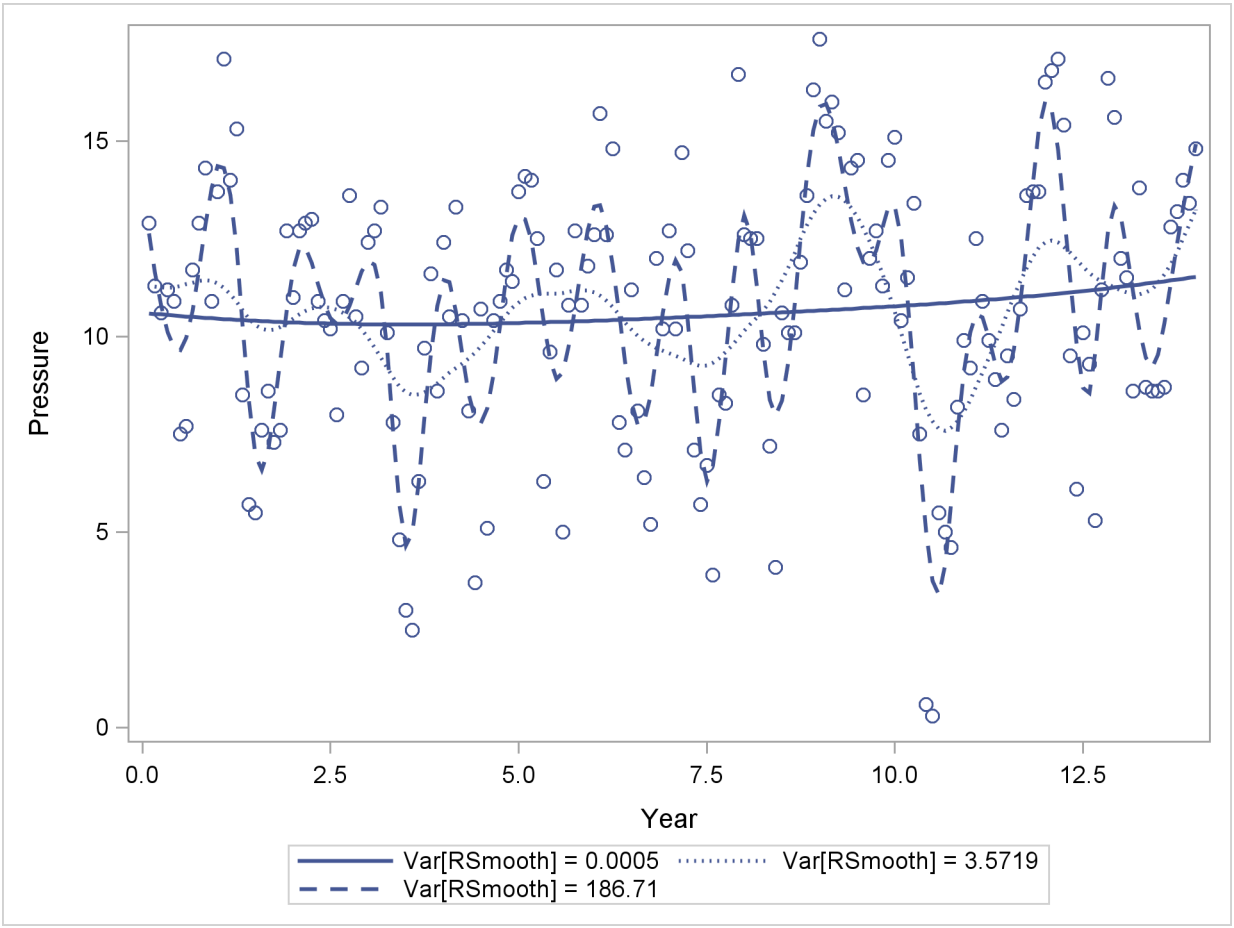
The local minimum at which the optimization stopped is clearly visible, as are a second local minimum near zero and the global minimum near 180.

The observed and predicted pressure differences that correspond to the three minima are shown in Output 38.10.3. These results were produced with the following statements:

```
proc glimmix data=enso;
   model pressure = year;
   random year / type=rsmooth knotmethod=equal(50);
   parms (0) (10);
   output out=gmxout1 pred=pred1;
run;
proc glimmix data=enso;
   model pressure = year;
   random year / type=rsmooth knotmethod=equal(50);
   output out=gmxout2 pred=pred2;
   parms (2) (10);
run;
proc glimmix data=enso;
   model pressure = year;
   random year / type=rsmooth knotmethod=equal(50);
```

```
   output out=gmxout3 pred=pred3;
   parms (200) (10);
run;
data plotthis; merge gmxout1 gmxout2 gmxout3;
run;
proc sgplot data=plotthis;
   scatter x=year y=Pressure;
   series  x=year y=pred1 /
        lineattrs   = (pattern=solid thickness=2)
        legendlabel = "Var[RSmooth] = 0.0005"
        name        = "pred1";
   series  x=year y=pred2 /
        lineattrs   = (pattern=dot thickness=2)
        legendlabel = "Var[RSmooth] = 3.5719"
        name        = "pred2";
   series  x=year y=pred3 /
        lineattrs   = (pattern=dash thickness=2)
        legendlabel = "Var[RSmooth] = 186.71"
        name        = "pred3";
   keylegend "pred1" "pred2" "pred3" / across=2;
run;
```

Output 38.10.3 Observed and Predicted Pressure Differences



The one-year cycle ($\widehat{\sigma}^2_r = 186.71$) and the El Niño cycle ($\widehat{\sigma}^2_r = 3.5719$) are clearly visible. Notice that a larger smoother variance results in larger BLUPs and hence larger adjustments to the fixed-effects model. A large smoother variance thus results in a more wiggly fit. The third local minimum at $\widehat{\sigma}^2_r = 0.0005$ applies only very small adjustments to the linear regression between pressure and time, creating slight curvature.

Example 38.11: Maximum Likelihood in Proportional Odds Model with Random Effects

The data for this example are taken from Gilmour, Anderson, and Rae (1987) and concern the foot shape of 2,513 lambs that represent 34 sires. The foot shape of the animals was scored in three ordered categories. The following DATA step lists the data in multivariate form, where each observation corresponds to a sire and contains the outcomes for the three response categories in the variables k1, k2, and k3. For example, for the first sire the first foot shape category was observed for 52 of its offspring, foot shape category 2 was observed for 25 lambs, and none of its offspring was rated in foot shape category 3. The variables yr, b1, b2, and b3 represent contrasts of fixed effects.

```
data foot_mv;
  input yr b1 b2 b3 k1 k2 k3;
  sire = _n_;
  datalines;
  1  1  0  0  52 25  0
  1  1  0  0  49 17  1
  1  1  0  0  50 13  1
  1  1  0  0  42  9  0
  1  1  0  0  74 15  0
  1  1  0  0  54  8  0
  1  1  0  0  96 12  0
  1 -1  1  0  57 52  9
  1 -1  1  0  55 27  5
  1 -1  1  0  70 36  4
  1 -1  1  0  70 37  3
  1 -1  1  0  82 21  1
  1 -1  1  0  75 19  0
  1 -1 -1  0  17 12 10
  1 -1 -1  0  13 23  3
  1 -1 -1  0  21 17  3
 -1  0  0  1  37 41 23
 -1  0  0  1  47 24 12
 -1  0  0  1  46 25  9
 -1  0  0  1  79 32 11
 -1  0  0  1  50 23  5
 -1  0  0  1  63 18  8
 -1  0  0 -1  30 20  9
 -1  0  0 -1  31 33  3
 -1  0  0 -1  28 18  4
 -1  0  0 -1  42 27  4
 -1  0  0 -1  35 22  2
```

```
 -1   0   0 -1   33 18   3
 -1   0   0 -1   35 17   4
 -1   0   0 -1   26 13   2
 -1   0   0 -1   37 15   2
 -1   0   0 -1   36 14   1
 -1   0   0 -1   63 20   3
 -1   0   0 -1   41  8   1
;
```

In order to analyze these data as multinomial data with PROC GLIMMIX, the data need to be arranged in univariate form. The following DATA step creates three observations from each record in data set foot_mv and stores the category counts in the variable count:

```
data footshape; set foot_mv;
  array k{3};
  do Shape = 1 to 3;
     count = k{Shape};
     output;
  end;
  drop k:;
run;
```

Because the sires were selected at random, we consider here a model for the three-category response with fixed regression effects for yr, b1–b3, and with random sire effects. Because the response categories are ordered, a proportional odds model is chosen (McCullagh 1980). Gilmour, Anderson, and Rae (1987) consider various analyses for these data. The following GLIMMIX statements fit a model with probit link for the cumulative probabilities by maximum likelihood where the marginal log likelihood is approximated by adaptive quadrature:

```
proc glimmix data=footshape method=quad;
   class sire;
   model Shape = yr b1 b2 b3 / s link=cumprobit dist=multinomial;
   random int / sub=sire s cl;
   ods output Solutionr=solr;
   freq count;
run;
```

The number of observations that share a particular response and covariate pattern (variable count) is used in the FREQ statement. The S and CL options request solutions for the sire effects. These are output to the data set solr for plotting.

The "Model Information" table shows that the parameters are estimated by maximum likelihood and that the marginal likelihood is approximated by Gauss-Hermite quadrature (Output 38.11.1).

Output 38.11.1 Model and Data Information

The GLIMMIX Procedure

Model Information

Data Set	WORK.FOOTSHAPE
Response Variable	Shape
Response Distribution	Multinomial (ordered)
Link Function	Cumulative Probit
Variance Function	Default
Frequency Variable	count
Variance Matrix Blocked By	sire
Estimation Technique	Maximum Likelihood
Likelihood Approximation	Gauss-Hermite Quadrature
Degrees of Freedom Method	Containment

Number of Observations Read	102
Number of Observations Used	96
Sum of Frequencies Read	2513
Sum of Frequencies Used	2513

Response Profile

Ordered Value	Shape	Total Frequency
1	1	1636
2	2	731
3	3	146

The GLIMMIX procedure is modeling the probabilities of levels of Shape having lower Ordered Values in the Response Profile table.

The distribution of the data is multinomial with ordered categories. The ordering is implied by the choice of a link function for the cumulative probabilities. Because a frequency variable is specified, the number of observations as well as the number of frequencies is displayed. Observations with zero frequency—that is, foot shape categories that were not observed for a particular sire are not used in the analysis. The "Response Profile Table" shows the ordering of the response variable and gives a breakdown of the frequencies by category.

Output 38.11.2 Information about the Size of the Optimization Problem

Dimensions

G-side Cov. Parameters	1
Columns in X	6
Columns in Z per Subject	1
Subjects (Blocks in V)	34
Max Obs per Subject	3

Output 38.11.2 *continued*

Optimization Information	
Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	7
Lower Boundaries	1
Upper Boundaries	0
Fixed Effects	Not Profiled
Starting From	GLM estimates
Quadrature Points	1

With METHOD=QUAD, the "Dimensions" and "Optimization Information" tables are particularly important, because for this estimation methods both fixed effects and covariance parameters participate in the optimization (Output 38.11.2). For GLM models the optimization involves the fixed effects and possibly a single scale parameter. For mixed models the fixed effects are typically profiled from the optimization. Laplace and quadrature estimations are exceptions to these rules. Consequently, there are seven parameters in this optimization, corresponding to six fixed effects and one variance component. The variance component has a lower bound of 0. Also, because the fixed effects are part of the optimizations, PROC GLIMMIX initially performs a few GLM iterations to obtain starting values for the fixed effects. You can control the number of initial iterations with the INITITER= option in the PROC GLIMMIX statement.

The last entry in the "Optimization Information" table shows that—at the starting values—PROC GLIMMIX determined that a single quadrature point is sufficient to approximate the marginal log likelihood with the required accuracy. This approximation is thus identical to the Laplace method that is available with METHOD=LAPLACE.

For METHOD=LAPLACE and METHOD=QUAD, the GLIMMIX procedure produces fit statistics based on the conditional and marginal distribution (Output 38.11.3). Within the limits of the numeric likelihood approximation, the information criteria shown in the "Fit Statistics" table can be used to compare models, and the -2 log likelihood can be used to compare among nested models (nested with respect to fixed effects and/or the covariance parameters).

Output 38.11.3 Marginal and Conditional Fit Statistics

Fit Statistics	
-2 Log Likelihood	3870.12
AIC (smaller is better)	3884.12
AICC (smaller is better)	3884.17
BIC (smaller is better)	3894.81
CAIC (smaller is better)	3901.81
HQIC (smaller is better)	3887.76

Fit Statistics for Conditional Distribution	
-2 log L(Shape | r. effects)	3807.62

The variance of the sire effect is estimated as 0.04849 with estimated asymptotic standard error of 0.01673 (Output 38.11.4). Based on the magnitude of the estimate relative to the standard error, one might conclude that there is significant sire-to-sire variability. Because parameter estimation is based on maximum likelihood, a formal test of the hypothesis of no sire variability is possible. The category cutoffs for the cumulative probabilities are 0.3781 and 1.6435. Except for b3, all fixed effects contrasts are significant.

Output 38.11.4 Parameter Estimates

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error
Intercept	sire	0.04849	0.01673

Solutions for Fixed Effects

Effect	Shape	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	1	0.3781	0.04907	29	7.71	<.0001
Intercept	2	1.6435	0.05930	29	27.72	<.0001
yr		0.1422	0.04834	2478	2.94	0.0033
b1		0.3781	0.07154	2478	5.28	<.0001
b2		0.3157	0.09709	2478	3.25	0.0012
b3		−0.09887	0.06508	2478	−1.52	0.1289

A likelihood ratio test for the sire variability can be carried out by adding a COVTEST statement to the PROC GLIMMIX statements (Output 38.11.5):

```
ods select FitStatistics CovParms Covtests;
proc glimmix data=footshape method=quad;
   class sire;
   model Shape = yr b1 b2 b3 / link=cumprobit dist=multinomial;
   random int / sub=sire;
   covtest GLM;
   freq count;
run;
```

The statement

```
covtest GLM;
```

compares the fitted model to a generalized linear model for independent data by removing the sire variance component from the model. Equivalently, you can specify

```
covtest 0;
```

which compares the fitted model against one where the sire variance is fixed at zero.

Output 38.11.5 Likelihood Ratio Test for Sire Variance

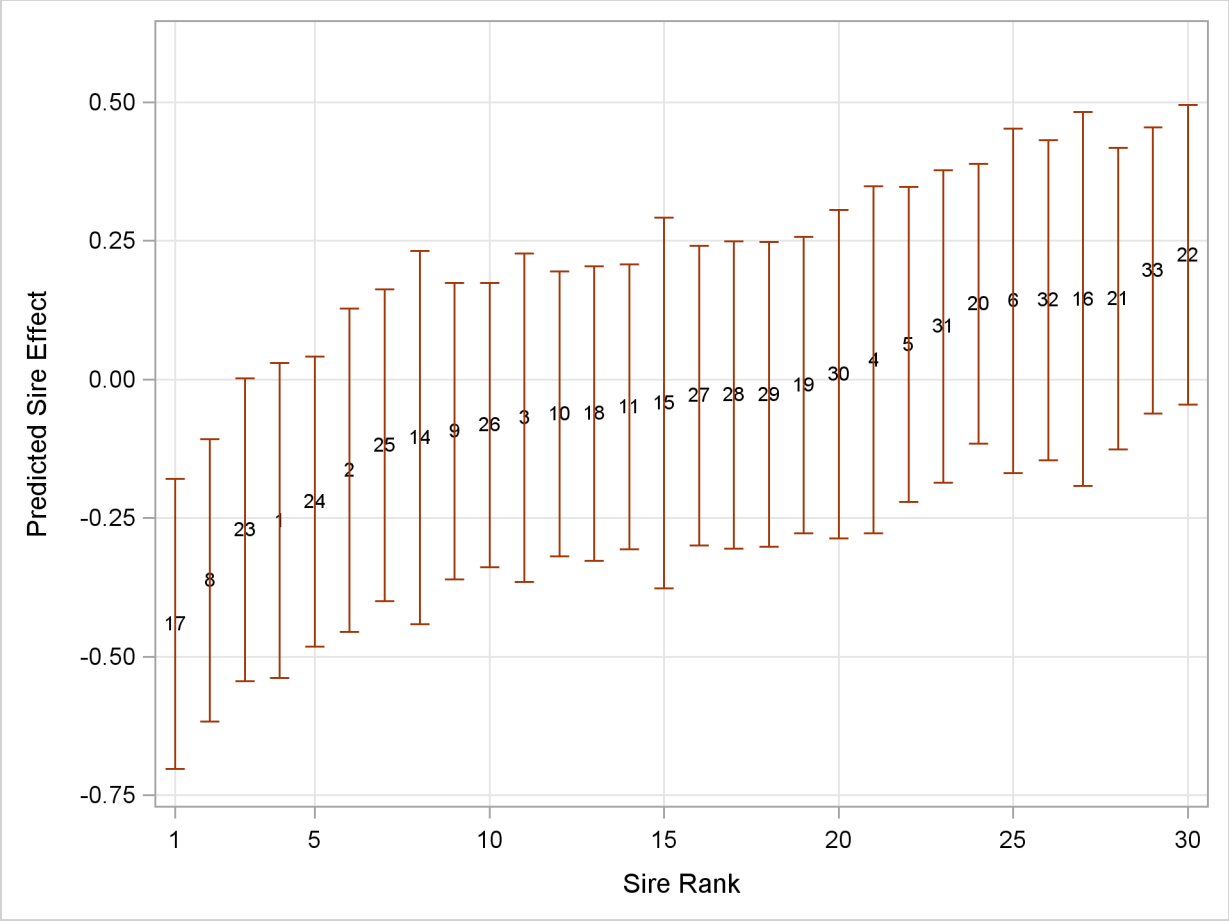
```
                          The GLIMMIX Procedure

                              Fit Statistics

                      -2 Log Likelihood            3870.12
                      AIC  (smaller is better)     3884.12
                      AICC (smaller is better)     3884.17
                      BIC  (smaller is better)     3894.81
                      CAIC (smaller is better)     3901.81
                      HQIC (smaller is better)     3887.76

                         Covariance Parameter Estimates

                                                    Standard
                  Cov Parm     Subject    Estimate       Error

                  Intercept    sire        0.04849     0.01673

                         Tests of Covariance Parameters
                            Based on the Likelihood

      Label              DF    -2 Log Like      ChiSq    Pr > ChiSq    Note

      Independence        1        3915.29      45.17        <.0001    MI

MI: P-value based on a mixture of chi-squares.
```

The −2 Log Likelihood in the reduced model without the sire effect is 3915.29. Compared to the corresponding marginal fit statistic in the full model (3870.12), this results in a chi-square statistic of 45.17. Because the variance component for the sire effect has a natural lower bound of zero, PROC GLIMMIX performs the likelihood ratio test as a one-sided test. As indicated by the note, the *p*-value for this test is computed from a mixture of chi-square distributions, applying the results of Self and Liang (1987). There is significant evidence that the model without sire random effects does not fit the data as well.

In studies of heritability, one is often interested to rank individuals according to some measure of "breeding value." The following statements display the empirical Bayes estimates of the sire effects from ML estimation by quadrature along with prediction standard error bars (Output 38.11.6):

```
proc sort data=solr; by Estimate;
data solr; set solr;
    length sire $2;
    obs  = _n_;
    sire = left(substr(Subject,6,2));
run;
proc sgplot data=solr;
   scatter x=obs y=estimate /
         markerchar  = sire
         yerrorupper = upper
         yerrorlower = lower;
   xaxis grid label='Sire Rank' values=(1 5 10 15 20 25 30);
   yaxis grid label='Predicted Sire Effect';
run;
```

Output 38.11.6 Ranked Predicted Sire Effects and Prediction Standard Errors



Example 38.12: Fitting a Marginal (GEE-Type) Model

A marginal GEE-type model for clustered data is a model for correlated data that is specified through a mean function, a variance function, and a "working" covariance structure. Because the assumed covariance structure can be wrong, the covariance matrix of the parameter estimates is not based on the model alone. Rather, one of the empirical ("sandwich") estimators is used to make inferences robust against the choice of working covariance structure. PROC GLIMMIX can fit marginal models by using R-side random effects and drawing on the distributional specification in the MODEL statement to derive the link and variance functions. The EMPIRICAL= option in the PROC GLIMMIX statement enables you to choose one of a number of empirical covariance estimators.

The data for this example are from Thall and Vail (1990) and reflect the number of seizures of patients suffering fom epileptic episodes. After an eight-week period without treatment, patients were observed four times in two-week intervals during which they received a placebo or the drug Progabide in addition to other therapy. These data are also analyzed in Example 37.7 of Chapter 37, "The GENMOD Procedure." The following DATA step creates the data set seizures. The variable id identifies the subjects in the study, and the variable trt identifies whether a subject received the placebo (trt = 0) or the drug Progabide (trt = 1). The variable x1 takes on value 0 for the baseline

measurement and 1 otherwise.

```
data seizures;
   array c{5};
   input id trt c1-c5;
   do i=1 to 5;
      x1    = (i > 1);
      ltime = (i=1)*log(8) + (i ne 1)*log(2);
      cnt   = c{i};
      output;
   end;
   keep id cnt x1 trt ltime;
   datalines;
101 1  76 11 14  9  8
102 1  38  8  7  9  4
103 1  19  0  4  3  0
104 0  11  5  3  3  3
106 0  11  3  5  3  3
107 0   6  2  4  0  5
108 1  10  3  6  1  3
110 1  19  2  6  7  4
111 1  24  4  3  1  3
112 1  31 22 17 19 16
113 1  14  5  4  7  4
114 0   8  4  4  1  4
116 0  66  7 18  9 21
117 1  11  2  4  0  4
118 0  27  5  2  8  7
121 1  67  3  7  7  7
122 1  41  4 18  2  5
123 0  12  6  4  0  2
124 1   7  2  1  1  0
126 0  52 40 20 23 12
128 1  22  0  2  4  0
129 1  13  5  4  0  3
130 0  23  5  6  6  5
135 0  10 14 13  6  0
137 1  46 11 14 25 15
139 1  36 10  5  3  8
141 0  52 26 12  6 22
143 1  38 19  7  6  7
145 0  33 12  6  8  4
147 1   7  1  1  2  3
201 0  18  4  4  6  2
202 0  42  7  9 12 14
203 1  36  6 10  8  8
204 1  11  2  1  0  0
205 0  87 16 24 10  9
206 0  50 11  0  0  5
208 1  22  4  3  2  4
209 1  41  8  6  5  7
210 0  18  0  0  3  3
211 1  32  1  3  1  5
213 0 111 37 29 28 29
```

```
214 1  56 18 11 28 13
215 0  18  3  5  2  5
217 0  20  3  0  6  7
218 1  24  6  3  4  0
219 0  12  3  4  3  4
220 0   9  3  4  3  4
221 1  16  3  5  4  3
222 0  17  2  3  3  5
225 1  22  1 23 19  8
226 0  28  8 12  2  8
227 0  55 18 24 76 25
228 1  25  2  3  0  1
230 0   9  2  1  2  1
232 1  13  0  0  0  0
234 0  10  3  1  4  2
236 1  12  1  4  3  2
238 0  47 13 15 13 12
;
```

The model fit initially with the following PROC GLIMMIX statements is a Poisson generalized linear model with effects for an intercept, the baseline measurement, the treatment, and their interaction:

```
proc glimmix data=seizures;
   model cnt = x1 trt x1*trt / dist=poisson offset=ltime
                               ddfm=none s;
run;
```

The DDFM=NONE option is chosen in the MODEL statement to produce chi-square and z tests instead of F and t tests.

Because the initial pretreatment time period is four times as long as the subsequent measurement intervals, an offset variable is used to standardize the counts. If Y_{ij} denotes the number of seizures of subject i in time interval j of length t_j, then Y_{ij}/t_j is the number of seizures per time unit. Modeling the average number per time unit with a log link leads to $\log\{\mathrm{E}[Y_{ij}/t_j]\} = \mathbf{x}'\boldsymbol{\beta}$ or $\log\{\mathrm{E}[Y_{ij}]\} = \mathbf{x}'\boldsymbol{\beta} + \log\{t_j\}$. The logarithm of time (variable ltime) thus serves as an offset. Suppose that β_0 denotes the intercept, β_1 the effect of x1, and β_2 the effect of trt. Then $\exp\{\beta_0\}$ is the expected number of seizures per week in the placebo group at baseline. The corresponding numbers in the treatment group are $\exp\{\beta_0 + \beta_2\}$ at baseline and $\exp\{\beta_0 + \beta_1 + \beta_2\}$ for postbaseline visits.

The "Model Information" table shows that the parameters in this Poisson model are estimated by maximum likelihood (Output 38.12.1). In addition to the default link and variance function, the variable ltime is used as an offset.

Output 38.12.1 Model Information in Poisson GLM

The GLIMMIX Procedure

Model Information

Data Set	WORK.SEIZURES
Response Variable	cnt
Response Distribution	Poisson
Link Function	Log
Variance Function	Default
Offset Variable	ltime
Variance Matrix	Diagonal
Estimation Technique	Maximum Likelihood
Degrees of Freedom Method	None

Fit statistics and parameter estimates are shown in Output 38.12.2.

Output 38.12.2 Results from Fitting Poisson GLM

Fit Statistics

-2 Log Likelihood	3442.66
AIC (smaller is better)	3450.66
AICC (smaller is better)	3450.80
BIC (smaller is better)	3465.34
CAIC (smaller is better)	3469.34
HQIC (smaller is better)	3456.54
Pearson Chi-Square	3015.16
Pearson Chi-Square / DF	10.54

Parameter Estimates

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	1.3476	0.03406	Infty	39.57	<.0001
x1	0.1108	0.04689	Infty	2.36	0.0181
trt	-0.1080	0.04865	Infty	-2.22	0.0264
x1*trt	-0.3016	0.06975	Infty	-4.32	<.0001

Because this is a generalized linear model, the large value for the ratio of the Pearson chi-square statistic and its degrees of freedom is indicative of a model shortcoming. The data are considerably more dispersed than is expected under a Poisson model. There could be many reasons for this overdispersion—for example, a misspecified mean model, data that might not be Poisson distributed, an incorrect variance function, and correlations among the observations. Because these data are repeated measurements, the presence of correlations among the observations from the same subject is a likely contributor to the overdispersion.

The following PROC GLIMMIX statements fit a marginal model with correlations. The model is a marginal one, because no G-side random effects are specified on which the distribution could be conditioned. The choice of the id variable as the SUBJECT effect indicates that observations from

different IDs are uncorrelated. Observations from the same ID are assumed to follow a compound symmetry (equicorrelation) model. The EMPIRICAL option in the PROC GLIMMIX statement requests the classical sandwich estimator as the covariance estimator for the fixed effects:

```
proc glimmix data=seizures empirical;
   class id;
   model cnt = x1 trt x1*trt / dist=poisson offset=ltime
                               ddfm=none covb s;
   random _residual_ / subject=id type=cs vcorr;
run;
```

The "Model Information" table shows that the parameters are now estimated by residual pseudo-likelihood (compare Output 38.12.3 and Output 38.12.1). And in this fact lies the main difference between fitting marginal models with PROC GLIMMIX and with GEE methods as per Liang and Zeger (1986), where parameters of the working correlation matrix are estimated by the method of moments.

Output 38.12.3 Model Information in Marginal Model

The GLIMMIX Procedure

Model Information

Data Set	WORK.SEIZURES
Response Variable	cnt
Response Distribution	Poisson
Link Function	Log
Variance Function	Default
Offset Variable	ltime
Variance Matrix Blocked By	id
Estimation Technique	Residual PL
Degrees of Freedom Method	None
Fixed Effects SE Adjustment	Sandwich - Classical

According to the compound symmetry model, there is substantial correlation among the observations from the same subject (Output 38.12.4).

Output 38.12.4 Covariance Parameter Estimates and Correlation Matrix

Estimated V Correlation Matrix for id 101

Row	Col1	Col2	Col3	Col4	Col5
1	1.0000	0.6055	0.6055	0.6055	0.6055
2	0.6055	1.0000	0.6055	0.6055	0.6055
3	0.6055	0.6055	1.0000	0.6055	0.6055
4	0.6055	0.6055	0.6055	1.0000	0.6055
5	0.6055	0.6055	0.6055	0.6055	1.0000

Output 38.12.4 *continued*

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error
CS	id	6.4653	1.3833
Residual		4.2128	0.3928

The parameter estimates in Output 38.12.5 are the same as in the Poisson generalized linear model (Output 38.12.2), because of the balance in these data. The standard errors have increased substantially, however, by taking into account the correlations among the observations.

Output 38.12.5 GEE-Type Inference for Fixed Effects

Solutions for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	1.3476	0.1574	Infty	8.56	<.0001
x1	0.1108	0.1161	Infty	0.95	0.3399
trt	−0.1080	0.1937	Infty	−0.56	0.5770
x1*trt	−0.3016	0.1712	Infty	−1.76	0.0781

Empirical Covariance Matrix for Fixed Effects

Effect	Row	Col1	Col2	Col3	Col4
Intercept	1	0.02476	−0.00115	−0.02476	0.001152
x1	2	−0.00115	0.01348	0.001152	−0.01348
trt	3	−0.02476	0.001152	0.03751	−0.00300
x1*trt	4	0.001152	−0.01348	−0.00300	0.02931

Example 38.13: Response Surface Comparisons with Multiplicity Adjustments

Koch et al. (1990) present data for a multicenter clinical trial testing the efficacy of a respiratory drug in patients with respiratory disease. Within each of two centers, patients were randomly assigned to a placebo (P) or an active (A) treatment. Prior to treatment and at four follow-up visits, patient status was recorded in one of five ordered categories (0=terrible, 1=poor, ..., 4=excellent). The following DATA step creates the SAS data set clinical for this study.

```
data Clinical;
   do Center =   1,  2;
   do Gender = 'F','M';
   do Drug   = 'A','P';
      input nPatient @@;
      do iPatient = 1 to nPatient;
         input ID Age (t0-t4) (1.) @@;
         output;
      end;
   end; end; end;
   datalines;
 2  53 32 12242  18 47 22344
 5   5 13 44444  19 31 21022  25 35 10000  28 36 23322
    36 45 22221
25  54 11 44442  12 14 23332  51 15 02333  20 20 33231
    16 22 12223  50 22 21344   3 23 33443  32 23 23444
    56 25 23323  35 26 12232  26 26 22222  21 26 24142
     8 28 12212  30 28 00121  33 30 33442  11 30 34443
    42 31 12311   9 31 33444  37 31 02321  23 32 34433
     6 34 11211  22 46 43434  24 48 23202  38 50 22222
    48 57 33434
24  43 13 34444  41 14 22123  34 15 22332  29 19 23300
    15 20 44444  13 23 33111  27 23 44244  55 24 34443
    17 25 11222  45 26 24243  40 26 12122  44 27 12212
    49 27 33433  39 23 21111   2 28 20000  14 30 10000
    31 37 10000  10 37 32332   7 43 23244  52 43 11132
     4 44 34342   1 46 22222  46 49 22222  47 63 22222
 4  30 37 13444  52 39 23444  23 60 44334  54 63 44444
12  28 31 34444   5 32 32234  21 36 33213  50 38 12000
     1 39 12112  48 39 32300   7 44 34444  38 47 23323
     8 48 22100  11 48 22222   4 51 34244  17 58 14220
23  12 13 44444  10 14 14444  27 19 33233  47 20 24443
    16 20 21100  29 21 33444  20 24 44444  25 25 34331
    15 25 34433   2 25 22444   9 26 23444  49 28 23221
    55 31 44444  43 34 24424  26 35 44444  14 37 43224
    36 41 34434  51 43 33442  37 52 12122  19 55 44444
    32 55 22331   3 58 44444  53 68 23334
16  39 11 34444  40 14 21232  24 15 32233  41 15 43334
    33 19 42233  34 20 32444  13 20 14444  45 33 33323
    22 36 24334  18 38 43000  35 42 32222  44 43 21000
     6 45 34212  46 48 44000  31 52 23434  42 66 33344
;
```

Westfall and Tobias (2007) define as the measure of efficacy the average of the ratings at the final two visits and model this average as a function of drug, baseline assessment score, and age. Hence, in their model, the expected efficacy for drug $d \in A$, P can be written as

$$\mathrm{E}[Y_d] = \beta_{0d} + \beta_{1d}t + \beta_{2d}a$$

where t is the baseline (pretreatment) assessment score and a is the patient's age at baseline. The age range for these data extends from 11 to 68 years. Suppose that the scientific question of interest is the comparison of the two response surfaces at a set of values $S_t \times S_a = \{0, 1, 2, 3, 4\} \times S_a$. In other words, we would like to know for which values of the covariates the average response differs significantly between the treatment group and the placebo group. If the set of ages of interest is $\{10, 13, 16, \cdots, 70\}$, then this involves $5 \times 21 = 105$ comparisons, a massive multiple testing problem. The large number of comparisons and the fact that the set S_a is chosen somewhat arbitrarily require the application of multiplicity corrections in order to protect the familywise Type I error across the comparisons.

When testing hypotheses that have logical restrictions, the power of multiplicity corrected tests can be increased by taking the restrictions into account. Logical restrictions exist, for example, when not all hypotheses in a set can be simultaneously true. Westfall and Tobias (2007) extend the truncated closed testing procedure (TCTP) of Royen (1989) for pairwise comparisons in ANOVA to general contrasts. Their work is also an extension of the S2 method of Shaffer (1986); see also Westfall (1997). These methods are all *monotonic* in the (unadjusted) p-values of the individual tests, in the sense that if $p_j < p_i$ then the multiple test will never retain H_j while rejecting H_i. In terms of multiplicity-adjusted p-values $\tilde{p}_j$, monotonicity means that if $p_j < p_i$, then $\tilde{p}_j < \tilde{p}_i$.

Analysis as Normal Data with Averaged Endpoints

In order to apply the extended TCTP procedure of Westfall and Tobias (2007) to the problem of comparing response surfaces in the clinical trial, the following convenience macro is helpful to generate the comparisons for the ESTIMATE statement in PROC GLIMMIX:

```
%macro Contrast(from,to,byA,byT);
   %let nCmp = 0;
   %do age = &from %to &to %by &byA;
      %do t0  =  0 %to  4 %by &byT;
         %let nCmp = %eval(&nCmp+1);
      %end;
   %end;
   %let iCmp = 0;
   %do age = &from %to &to %by &byA;
      %do t0  =  0 %to  4 %by &byT;
        %let iCmp = %eval(&iCmp+1);
        "%trim(%left(&age)) %trim(%left(&t0))"
          drug     1    -1
          drug*age &age -&age
          drug*t0  &t0  -&t0
        %if (&icmp < &nCmp) %then %do; , %end;
      %end;
   %end;
%mend;
```

The following GLIMMIX statements fit the model to the data and compute the 105 contrasts that compare the placebo to the active response at 105 points in the two-dimensional regressor space:

```
proc glimmix data=clinical;
   t = (t3+t4)/2;
   class drug;
   model t = drug t0 age drug*age drug*t0;
   estimate %contrast(10,70,3,1)
              / adjust=simulate(seed=1)
                stepdown(type=logical);
   ods output Estimates=EstStepDown;
run;
```

Note that only a single ESTIMATE statement is used. Each of the 105 comparisons is one comparison in the multirow statement. The ADJUST option in the ESTIMATE statement requests multiplicity-adjusted p-values. The extended TCTP method is applied by specifying the STEPDOWN(TYPE=LOGICAL) option to compute step-down-adjusted p-values where logical constraints among the hypotheses are taken into account. The results from the ESTIMATE statement are saved to a data set for subsequent processing. Note also that the response, the average of the ratings at the final two visits, is computed with programming statements in PROC GLIMMIX.

The following statements print the 20 most significant estimated differences (Output 38.13.1):

```
proc sort data=EstStepDown;
   by Probt;
proc print data=EstStepDown(obs=20);
   var Label Estimate StdErr Probt AdjP;
run;
```

Output 38.13.1 The First 20 Observations of the Estimates Data Set

Obs	Label	Estimate	StdErr	Probt	Adjp
1	37 2	0.8310	0.2387	0.0007	0.0071
2	40 2	0.8813	0.2553	0.0008	0.0071
3	34 2	0.7806	0.2312	0.0010	0.0071
4	43 2	0.9316	0.2794	0.0012	0.0071
5	46 2	0.9819	0.3093	0.0020	0.0071
6	31 2	0.7303	0.2338	0.0023	0.0081
7	49 2	1.0322	0.3434	0.0033	0.0107
8	52 2	1.0825	0.3807	0.0054	0.0167
9	40 3	0.7755	0.2756	0.0059	0.0200
10	37 3	0.7252	0.2602	0.0063	0.0201
11	43 3	0.8258	0.2982	0.0066	0.0201
12	28 2	0.6800	0.2461	0.0068	0.0215
13	55 2	1.1329	0.4202	0.0082	0.0239
14	46 3	0.8761	0.3265	0.0085	0.0239
15	34 3	0.6749	0.2532	0.0089	0.0257
16	43 1	1.0374	0.3991	0.0107	0.0329
17	46 1	1.0877	0.4205	0.0111	0.0329
18	49 3	0.9264	0.3591	0.0113	0.0329
19	40 1	0.9871	0.3827	0.0113	0.0329
20	58 2	1.1832	0.4615	0.0118	0.0329

Notice that the adjusted p-values (Adjp) are larger than the unadjusted p-values, as expected. Also notice that several comparisons share the same adjusted p-values. This is a result of the monotonicity of the extended TCTP method.

In order to compare the step-down-adjusted p-values to adjusted p-values that do not use step-down methods, replace the ESTIMATE statement in the previous statements with the following:

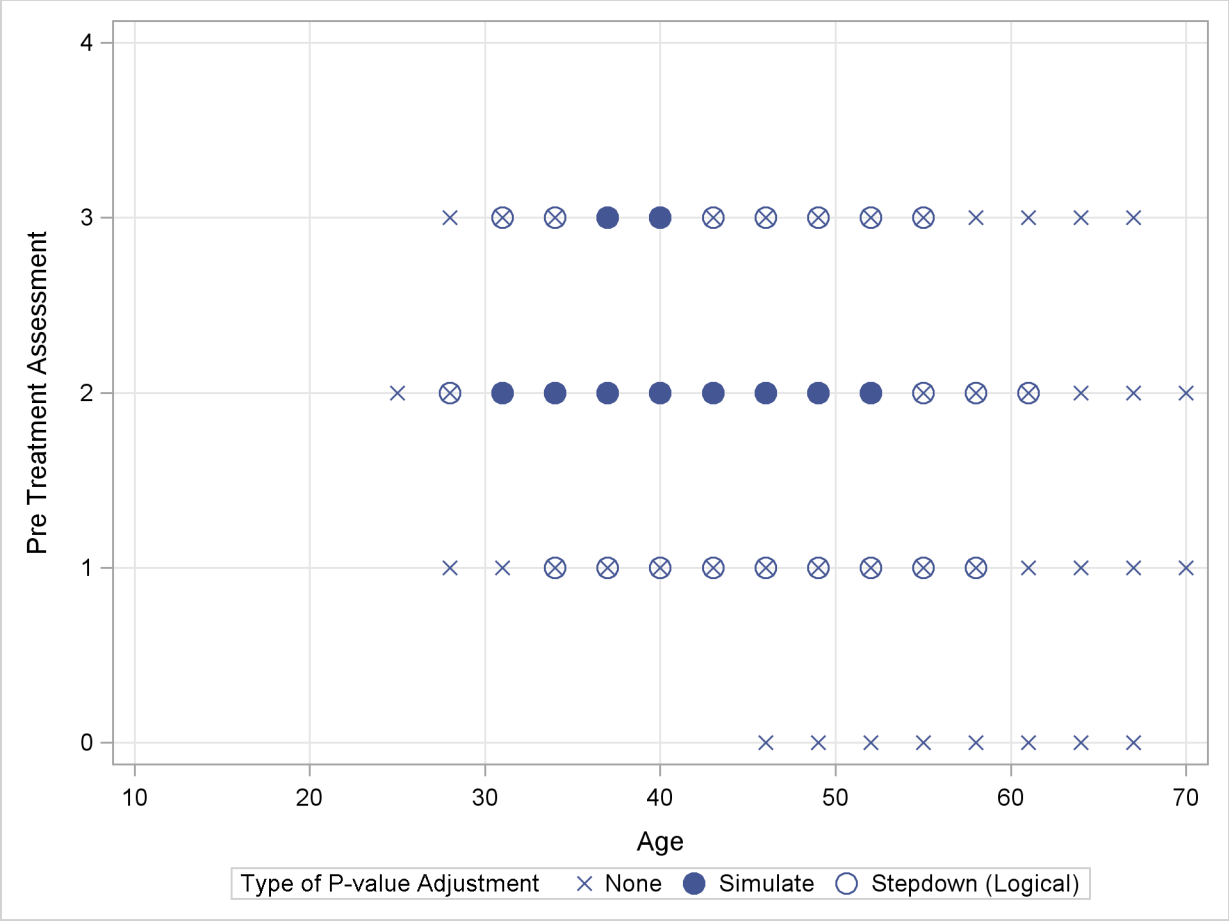
```
   estimate %contrast2(10,70,3,1) / adjust=simulate(seed=1);
   ods output Estimates=EstAdjust;
```

The following GLIMMIX invocations create output data sets named EstAdjust and EstUnAdjust that contain (non-step-down-) adjusted and unadjusted p-values:

```
proc glimmix data=clinical;
   t = (t3+t4)/2;
   class drug;
   model t = drug t0 age drug*age drug*t0;
   estimate %contrast(10,70,3,1)
              / adjust=simulate(seed=1);
   ods output Estimates=EstAdjust;
run;
proc glimmix data=clinical;
   t = (t3+t4)/2;
   class drug;
   model t = drug t0 age drug*age drug*t0;
   estimate %contrast(10,70,3,1);
   ods output Estimates=EstUnAdjust;
run;
```

Output 38.13.2 shows a comparison of the significant comparisons ($p < 0.05$) based on unadjusted, adjusted, and step-down (TCTP) adjusted p-values. Clearly, the unadjusted results indicate the most significant results, but without protecting the Type I error rate for the group of tests. The adjusted p-values (filled circles) lead to a much smaller region in which the response surfaces between treatment and placebo are significantly different. The increased power of the TCTP procedure (open circles) over the standard multiplicity adjustment—without sacrificing Type I error protection—can be seen in the considerably larger region covered by the open circles.

Output 38.13.2 Comparison of Significance Regions



Ordinal Repeated Measure Analysis

The outcome variable in this clinical trial is an ordinal rating of patients in categories 0=terrible, 1=poor, 2=fair, 3=good, and 4=excellent. Furthermore, the observations from repeat visits for the same patients are likely correlated. The previous analysis removes the repeated measures aspect by defining efficacy as the average score at the final two visits. These averages are not normally distributed, however. The response surfaces for the two study arms can also be compared based on a model for ordinal data that takes correlation into account through random effects. Keeping with the theme of the previous analysis, the focus here for illustrative purposes is on the final two visits, and the pretreatment assessment score serves as a covariate in the model.

The following DATA step rearranges the data from the third and fourth on-treatment visits in univariate form with separate observations for the visits by patient:

```
data clinical_uv;
   set clinical;
   array time{2} t3-t4;
   do i=1 to 2; rating = time{i}; output; end;
run;
```

The basic model for the analysis is a proportional odds model with cumulative logit link (McCullagh 1980) and $J = 5$ categories. In this model, separate intercepts (cutoffs) are modeled for the first $J - 1 = 4$ cumulative categories and the intercepts are monotonically increasing. This guarantees ordering of the cumulative probabilities and nonnegative category probabilities. Using the same covariate structure as in the previous analysis, the probability to observe a rating in at most category $k \leq 4$ is

$$\Pr(Y_d \leq k) = \frac{1}{1 + \exp\{-\eta_{kd}\}}$$
$$\eta_{kd} = \alpha_k + \beta_{0d} + \beta_{1d} t + \beta_{2d} a$$

Because only the intercepts are dependent on the category, contrasts comparing regression coefficients can be formulated in standard fashion. To accommodate the random and covariance structure of the repeated measures model, a random intercept γ_i is applied to the observations for each patient:

$$\Pr(Y_{id} \leq k) = \frac{1}{1 + \exp\{-\eta_{ikd}\}}$$
$$\eta_{ikd} = \alpha_k + \beta_{0d} + \beta_{1d} t + \beta_{2d} a + \gamma_i$$
$$\gamma_i \sim iid\, N(0, \sigma^2_\gamma)$$

The shared random effect of the two observations creates a marginal correlation. Note that the random effects do not depend on category.

The following GLIMMIX statements fit this ordinal repeated measures model by maximum likelihood via the Laplace approximation and compute TCTP-adjusted p-values for the 105 estimates:

```
proc glimmix data=clinical_uv method=laplace;
   class center id drug;
   model rating = drug t0 age drug*age drug*t0 /
                  dist=multinomial link=cumlogit;
   random intercept / subject=id(center);
   covtest 0;
   estimate %contrast(10,70,3,1)
             / adjust=simulate(seed=1)
               stepdown(type=logical);
  ods output Estimates=EstStepDownMulti;
run;
```

The combination of DIST=MULTINOMIAL and LINK=CUMLOGIT requests the proportional odds model. The SUBJECT= effect nests patient IDs within centers, because patient IDs in the data set clinical are not unique within centers. (Specifying SUBJECT=ID*CENTER would have the same effect.) The COVTEST statement requests a likelihood ratio test for the significance of the random patient effect.

The estimate of the variance component for the random patient effect is substantial (Output 38.13.3), but so is its standard error.

Output 38.13.3 Model and Covariance Parameter Information

The GLIMMIX Procedure

Model Information

Data Set	WORK.CLINICAL_UV
Response Variable	rating
Response Distribution	Multinomial (ordered)
Link Function	Cumulative Logit
Variance Function	Default
Variance Matrix Blocked By	ID(Center)
Estimation Technique	Maximum Likelihood
Likelihood Approximation	Laplace
Degrees of Freedom Method	Containment

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error
Intercept	ID(Center)	10.3483	3.2599

Tests of Covariance Parameters
Based on the Likelihood

Label	DF	-2 Log Like	ChiSq	Pr > ChiSq	Note
Parameter list	1	604.70	57.64	<.0001	MI

MI: P-value based on a mixture of chi-squares.

The likelihood ratio test provides a better picture of the significance of the variance component. The difference in the -2 log likelihoods is 57.6, highly significant even if one does not apply the Self and Liang (1987) correction that halves the p-value in this instance.

The results for the 20 most significant estimates are requested with the following statements and shown in Output 38.13.4:

```
proc sort data=EstStepDownMulti;
   by Probt;
proc print data=EstStepDownMulti(obs=20);
   var Label Estimate StdErr Probt AdjP;
run;
```

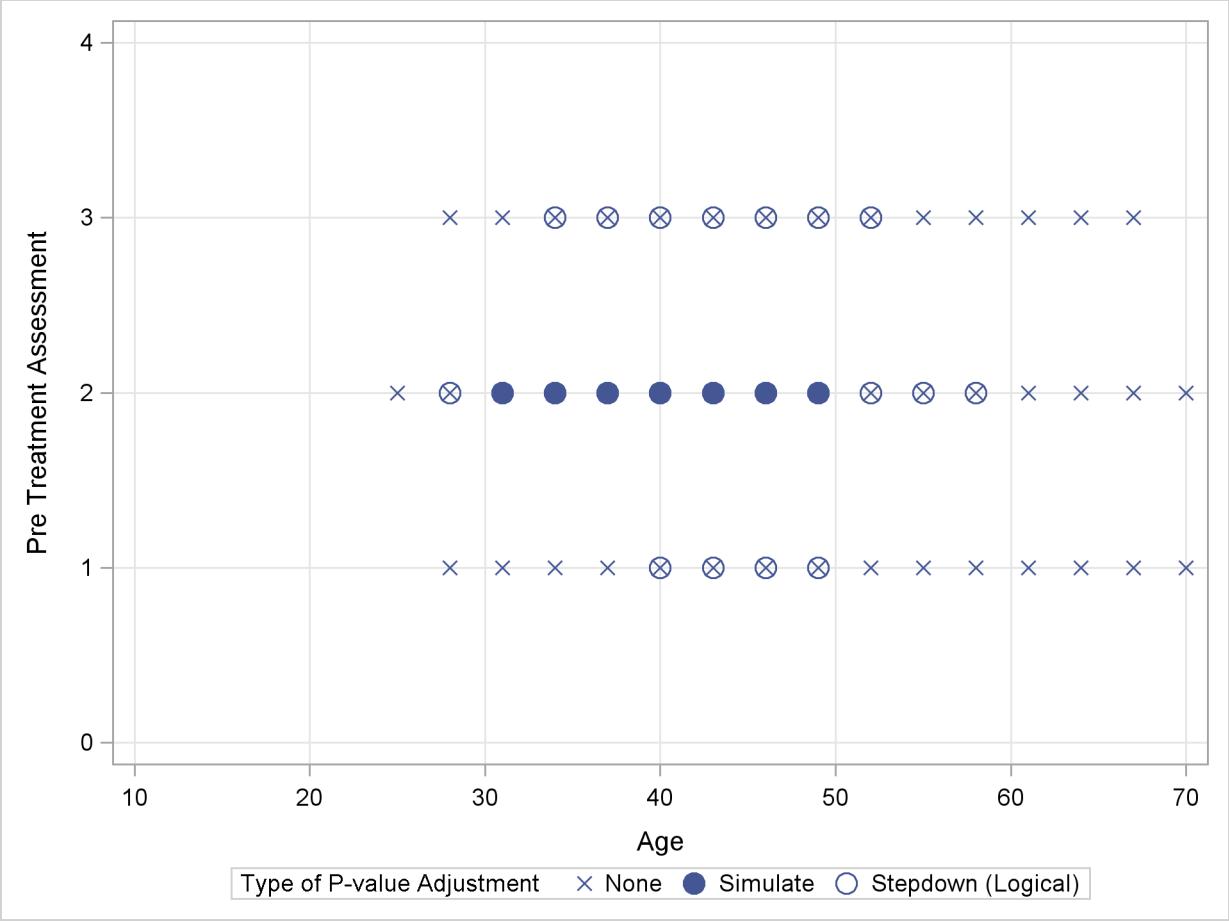
The p-values again show the "repeat" pattern corresponding to the monotonicity of the step-down procedure.

Output 38.13.4 The First 20 Estimates in the Ordinal Analysis

Obs	Label	Estimate	StdErr	Probt	Adjp
1	37 2	-2.7224	0.8263	0.0013	0.0133
2	40 2	-2.8857	0.8842	0.0015	0.0133
3	34 2	-2.5590	0.7976	0.0018	0.0133
4	43 2	-3.0491	0.9659	0.0021	0.0133
5	46 2	-3.2124	1.0660	0.0032	0.0133
6	31 2	-2.3957	0.8010	0.0034	0.0133
7	49 2	-3.3758	1.1798	0.0051	0.0164
8	52 2	-3.5391	1.3037	0.0077	0.0236
9	40 3	-2.6263	0.9718	0.0080	0.0267
10	37 3	-2.4630	0.9213	0.0087	0.0267
11	28 2	-2.2323	0.8362	0.0088	0.0278
12	43 3	-2.7897	1.0451	0.0088	0.0278
13	46 3	-2.9530	1.1368	0.0107	0.0291
14	55 2	-3.7025	1.4351	0.0112	0.0324
15	34 3	-2.2996	0.8974	0.0118	0.0337
16	49 3	-3.1164	1.2428	0.0136	0.0344
17	43 1	-3.3085	1.3438	0.0154	0.0448
18	58 2	-3.8658	1.5722	0.0155	0.0448
19	40 1	-3.1451	1.2851	0.0160	0.0448
20	46 1	-3.4718	1.4187	0.0160	0.0448

As previously, the comparisons were also performed with standard p-value adjustment via simulation. Output 38.13.5 displays the components of the regressor space in which the response surfaces differ significantly ($p < 0.05$) between the two treatment arms. As before, the most significant differences occur with unadjusted p-values at the cost of protecting only the individual Type I error rate. The standard multiplicity adjustment has considerably less power than the TCTP adjustment.

Output 38.13.5 Comparison of Significance Regions, Ordinal Analysis



Example 38.14: Generalized Poisson Mixed Model for Overdispersed Count Data

Overdispersion is the condition by which data appear more dispersed than is expected under a reference model. For count data, the reference models are typically based on the binomial or Poisson distributions. Among the many reasons for overdispersion are an incorrect model, an incorrect distributional specification, incorrect variance functions, positive correlation among the observations, and so forth. In short, correcting an overdispersion problem, if it exists, requires the appropriate remedy. Adding an R-side scale parameter to multiply the variance function is not necessarily the adequate correction. For example, Poisson-distributed data appear overdispersed relative to a Poisson model with regressors when an important regressor is omitted.

If the reference model for count data is Poisson, a number of alternative model formulations are available to increase the dispersion. For example, zero-inflated models add a proportion of zeros (usually from a bernoulli process) to the zeros of a Poisson process. Hurdle models are two-part models where zeros and nonzeros are generated by different stochastic processes. Zero-inflated and hurdle models are described in detail by Cameron and Trivedi (1998) and cannot be fit with the GLIMMIX procedure. See Section 15.5 in Littell et al. (2006) for examples of using the NLMIXED

procedure to fit zero-inflated and hurdle models.

An alternative approach is to derive from the reference distribution a probability distribution that exhibits increased dispersion. By mixing a Poisson process with a gamma distribution for the Poisson parameter, for example, the negative binomial distribution results, which is thus overdispersed relative to the Poisson.

Joe and Zhu (2005) show that the generalized Poisson distribution can also be motivated as a Poisson mixture and hence provides an alternative to the negative binomial (NB) distribution. Like the NB, the generalized Poisson distribution has a scale parameter. It is heavier in the tails than the NB distribution and easily reduces to the standard Poisson. Joe and Zhu (2005) discuss further comparisons between these distributions.

The probability mass function of the generalized Poisson is given by

$$p(y) = \frac{\alpha}{y!}(\alpha + \xi y)^{y-1} \exp\{-\alpha - \xi y\}$$

where $y = 0, 1, 2, \cdots, \alpha > 0$, and $0 \leq \xi < 1$ (Joe and Zhu 2005). Notice that for $\xi = 0$ the mass function of the standard Poisson distribution with mean α results. The mean and variance of Y in terms of the parameters α and ξ are given by

$$\begin{aligned} \mathrm{E}[Y] &= \frac{\alpha}{1-\xi} = \mu \\ \mathrm{Var}[Y] &= \frac{\alpha}{(1-\xi)^3} = \frac{\mu}{(1-\xi)^2} \end{aligned}$$

The log likelihood of the generalized Poisson can thus be written in terms of the mean μ and scale parameter ξ as

$$\begin{aligned} l(\mu, \xi; y) = &\log\{\mu(1-\xi)\} + (y-1)\log\{\mu - \xi(\mu - y)\} \\ &- (\mu - \xi(\mu - y)) - \log\{\Gamma(y+1)\} \end{aligned}$$

The data in the following DATA step are simulated counts. For each of $i = 1, \cdots, 30$ subjects a randomly varying number n_i of observations were drawn from a count regression model with a single covariate and excess zeros (compared to a Poisson distribution).

```
data counts;
   input ni @@;
   sub = _n_;
   do i=1 to ni;
      input x y @@;
      output;
   end;
   datalines;
  1 29 0
  6  2 0 82 5 33 0 15 2 35 0 79 0
 19 81 0 18 0 85 0 99 0 20 0 26 2 29 0 91 2 37 0 39 0  9 1 33 0
     3 0 60 0 87 2 80 0 75 0  3 0 63 1
  9 18 0 64 0 80 0  0 0 58 0  7 0 81 0 22 3 50 0
 15 91 0  2 1 14 0  5 2 27 1  8 1 95 0 76 0 62 0 26 2  9 0 72 1
    98 0 94 0 23 1
```

```
 2 34 0 95 0
18 48 1  5 0 47 0 44 0 27 0 88 0 27 0 68 0 84 0 86 0 44 0 90 0
   63 0 27 0 47 0 25 0 72 0 62 1
13 28 1 31 0 63 0 14 0 74 0 44 0 75 0 65 0 74 1 84 0 57 0 29 0
   41 0
 9 42 0  8 0 91 0 20 0 23 0 22 0 96 0 83 0 56 0
 3 64 0 64 1 15 0
 4  5 0 73 2 50 1 13 0
 2  0 0 41 0
20 21 0 58 0  5 0 61 1 28 0 71 0 75 1 94 16 51 4 51 2 74 0  1 1
   34 0  7 0 11 0 60 3 31 0 75 0 62 0 54 1
 2 66 1 13 0
 5 83 7 98 1 11 1 28 0 18 0
17 29 5 79 0 39 2 47 2 80 1 19 0 37 0 78 1 26 0 72 1  6 0 50 3
   50 4 97 0 37 2 51 0 45 0
17 47 0 57 0 33 0 47 0  2 0 83 0 74 0 93 0 36 0 53 0 26 0 86 0
    6 0 17 0 30 0 70 1 99 0
 7 91 0 25 1 51 4 20 0 61 1 34 0 33 2
14 60 0 87 0 94 0 29 0 41 0 78 0 50 0 37 0 15 0 39 0 22 0 82 0
   93 0  3 0
16 68 0 26 1 19 0 60 1 93 3 65 0 16 0 79 0 14 0  3 1 90 0 28 3
   82 0 34 0 30 0 81 0
19 48 3 48 1 43 2 54 0 45 9 53 0 14 0 92 5 21 1 20 0 73 0 99 0
   66 0 86 2 63 0 10 0 92 14 44 1 74 0
 8 34 1 44 0 62 0 21 0  7 0 17 0  0 2 49 0
13 11 0 27 2 16 1 12 3 52 1 55 0  2 6 89 5 31 5 28 3 51 5 54 13
   64 0
 9  3 0 36 0 57 0 77 0 41 0 39 0 55 0 57 0 88 1
 7  2 0 80 0 41 1 20 0  2 0 27 0 40 0
18 73 1 66 0 10 0 42 0 22 0 59 9 68 0 34 1 96 0 30 0 13 0 35 0
   51 2 47 0 60 1 55 4 83 3 38 0
17 96 0 40 0 34 0 59 0 12 1 47 0 93 0 50 0 39 0 97 0 19 0 54 0
   11 0 29 0 70 2 87 0 47 0
13 59 0 96 0 47 1 64 0 18 0 30 0 37 0 36 1 69 0 78 1 47 1 86 0
   88 0
15 66 0 45 1 96 1 17 0 91 0  4 0 22 0  5 2 47 0 38 0 80 0  7 1
   38 1 33 0 52 0
12 84 6 60 1 33 1 92 0 38 0  6 0 43 3 13 2 18 0 51 0 50 4 68 0
;
```

The following PROC GLIMMIX statements fit a standard Poisson regression model with random intercepts by maximum likelihood. The marginal likelihood of the data is approximated by adaptive quadrature (METHOD=QUAD).

```
proc glimmix data=counts method=quad;
   class sub;
   model y = x / link=log s dist=poisson;
   random int / subject=sub;
run;
```

Output 38.14.1 displays various informational items about the model and the estimation process.

Output 38.14.1 Poisson: Model and Optimization Information

The GLIMMIX Procedure

Model Information

Data Set	WORK.COUNTS
Response Variable	y
Response Distribution	Poisson
Link Function	Log
Variance Function	Default
Variance Matrix Blocked By	sub
Estimation Technique	Maximum Likelihood
Likelihood Approximation	Gauss-Hermite Quadrature
Degrees of Freedom Method	Containment

Optimization Information

Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	3
Lower Boundaries	1
Upper Boundaries	0
Fixed Effects	Not Profiled
Starting From	GLM estimates
Quadrature Points	5

Iteration History

Iteration	Restarts	Evaluations	Objective Function	Change	Max Gradient
0	0	4	862.57645728	.	366.7105
1	0	5	862.43893582	0.13752147	22.36158
2	0	6	854.49131023	7.94762559	28.70814
3	0	2	854.47983504	0.01147519	6.036114
4	0	4	854.47396189	0.00587315	4.238363
5	0	4	854.47006558	0.00389631	0.332454
6	0	3	854.47006484	0.00000074	0.003104

The "Model Information" table shows that the parameters are estimated by ML with quadrature. Using the starting values for fixed effects and covariance parameters that the GLIMMIX procedure generates by default, the procedure determined that five quadrature nodes provide a sufficiently accurate approximation of the marginal log likelihood ("Optimization Information" table). The iterative estimation process converges after nine iterations.

The table of conditional fit statistics displays the sum of the independent contributions to the conditional -2 log likelihood (854.47) and the Pearson statistics for the conditional distribution (Output 38.14.2).

Output 38.14.2 Poisson: Fit Statistics and Estimates

Fit Statistics

−2 Log Likelihood	854.47
AIC (smaller is better)	860.47
AICC (smaller is better)	860.54
BIC (smaller is better)	864.67
CAIC (smaller is better)	867.67
HQIC (smaller is better)	861.81

Fit Statistics for Conditional Distribution

−2 log L(y | r. effects)	777.90
Pearson Chi-Square	649.58
Pearson Chi-Square / DF	1.97

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error
Intercept	sub	1.1959	0.4334

Solutions for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	−1.4947	0.2745	29	−5.45	<.0001
x	0.01207	0.002387	299	5.06	<.0001

The departure of the scaled Pearson statistic from 1.0 is fairly pronounced in this case (1.97). If one deems it to far from 1.0, however, the conclusion has to be that the conditional variation is not properly specified. This could be due to an incorrect variance function, for example. The "Solutions for Fixed Effects" table shows the estimates of the slope and intercept in this model along with their standard errors and tests of significance. Note that the slope in this model is highly significant. The variance of the random subject-specific intercepts is estimated as 1.1959.

To fit the generalized Poisson distribution to these data we cannot draw on the built-in distributions. Instead, the variance function and the log likelihood are computed directly with PROC GLIMMIX programming statements. The CLASS, MODEL, and RANDOM statements in the following PROC GLIMMIX program are as before, except for the omission of the DIST= option in the MODEL statement:

```
proc glimmix data=counts method=quad;
   class sub;
   model y = x / link=log s;
   random int / subject=sub;
   xi = (1 - 1/exp(_phi_));
   _variance_ = _mu_ / (1-xi)/(1-xi);
   if (_mu_=.) or (_linp_ = .) then _logl_ = .;
```

```
   else do;
      mustar = _mu_ - xi*(_mu_ - y);
      if (mustar < 1E-12) or (_mu_*(1-xi) < 1e-12) then
         _logl_ = -1E20;
      else do;
         _logl_ = log(_mu_*(1-xi)) + (y-1)*log(mustar) -
                  mustar - lgamma(y+1);
      end;
   end;
run;
```

The assignments to the variables xi and the reserved symbols _VARIANCE_ and _LOGL_ define the variance function and the log likelihood. Because the scale parameter of the generalized Poisson distribution has the range $0 < \xi < 1$, and the scale parameter _PHI_ in the GLIMMIX procedure is bounded only from below (by 0), a reparameterization is applied so that $\phi = 0 \Leftrightarrow \xi = 0$ and ξ approaches 1 as ϕ increases. The statements preceding the calculation of the actual log likelihood are intended to prevent floating-point exceptions and to trap missing values.

Output 38.14.3 displays information about the model and estimation process. The "Model Information" table shows that the distribution is not a built-in distribution and echoes the expression for the user-specified variance function. As in the case of the Poisson model, the GLIMMIX procedure determines that five quadrature points are sufficient for accurate estimation of the marginal log likelihood at the starting values. The estimation process converges after 11 iterations.

Output 38.14.3 Generalized Poisson: Model, Optimization, and Iteration Information

The GLIMMIX Procedure

Model Information

Data Set	WORK.COUNTS
Response Variable	y
Response Distribution	User specified
Link Function	Log
Variance Function	_mu_ / (1-xi)/(1-xi)
Variance Matrix Blocked By	sub
Estimation Technique	Maximum Likelihood
Likelihood Approximation	Gauss-Hermite Quadrature
Degrees of Freedom Method	Containment

Optimization Information

Optimization Technique	Dual Quasi-Newton
Parameters in Optimization	4
Lower Boundaries	2
Upper Boundaries	0
Fixed Effects	Not Profiled
Starting From	GLM estimates
Quadrature Points	5

Output 38.14.3 *continued*

Iteration History

Iteration	Restarts	Evaluations	Objective Function	Change	Max Gradient
0	0	4	716.12976769	.	161.1184
1	0	5	716.07585953	0.05390816	11.88788
2	0	4	714.27148068	1.80437884	36.09657
3	0	2	711.02643265	3.24504804	108.4615
4	0	2	710.26952196	0.75691069	216.9822
5	0	2	709.96824991	0.30127205	96.2775
6	0	3	709.8419071	0.12634280	19.07487
7	0	3	709.83122731	0.01067980	0.649164
8	0	3	709.83047646	0.00075085	2.127665
9	0	3	709.83046461	0.00001185	0.383319
10	0	3	709.83046436	0.00000025	0.010279

The achieved -2 log likelihood is lower than in the Poisson model (compare "Fit Statistics" tables in Output 38.14.4 and Output 38.14.1). The scaled Pearson statistic is now less than 1.0. The fixed slope estimate remains significant at the 5% level, but the test statistics are not as large as in the Poisson model, partly because the generalized Poisson model permits more variation.

Output 38.14.4 Generalized Poisson: Fit Statistics and Estimates

Fit Statistics

-2 Log Likelihood	709.83
AIC (smaller is better)	717.83
AICC (smaller is better)	717.95
BIC (smaller is better)	723.44
CAIC (smaller is better)	727.44
HQIC (smaller is better)	719.62

Fit Statistics for Conditional Distribution

-2 log L(y | r. effects)	665.56
Pearson Chi-Square	241.42
Pearson Chi-Square / DF	0.73

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error
Intercept	sub	0.5135	0.2400
Scale		0.6401	0.09718

Output 38.14.4 *continued*

Solutions for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	−0.7264	0.2749	29	−2.64	0.0131
x	0.003742	0.003537	299	1.06	0.2910

Based on the large difference in the −2 log likelihoods between the Poisson and generalized Poisson models, we conclude that a mixed model based on the latter provides a better fit to these data. From the "Covariance Parameter Estimates" table in Output 38.14.4 you can see that the estimate of the scale parameter is $\widehat{\phi} = 0.6401$ and is considerably larger than 0, taking into account its standard error. The hypothesis $H: \phi = 0$, which articulates that a Poisson model fits the data as well as the generalized Poisson model, can be formally tested with a likelihood ratio test. Adding the statement

```
covtest 'H: phi = 0' . 0 / est;
```

to the previous PROC GLIMMIX run compares the model to one in which the variance of the random intercepts (the first covariance parameter) is not constrained and the scale parameter is fixed at zero. This COVTEST statement produces Output 38.14.5.

Output 38.14.5 Likelihood Ratio Test for Poisson Assumption

Tests of Covariance Parameters
Based on the Likelihood

Label	DF	−2 Log Like	ChiSq	Pr > ChiSq	--Estimates H0-- Est1	Est2	Note
H:phi = 0	1	854.47	144.64	<.0001	1.2	1.11E−12	MI

MI: P-value based on a mixture of chi-squares.

Note that the −2 Log Like reported in Output 38.14.5 agrees with the value reported in the "Fit Statistics" table for the Poisson model (Output 38.14.2) and that the estimate of the random intercept under the null hypothesis agrees with the "Covariance Parameter Estimates" table in Output 38.14.2. Because the null hypothesis places the parameter ϕ (or ξ) on the boundary of the parameter space, a mixture correction is applied in the p-value calculation. Because of the magnitude of the likelihood ratio statistic (144.64), this correction has no effect on the displayed p-value.

Example 38.15: Comparing Multiple B-Splines

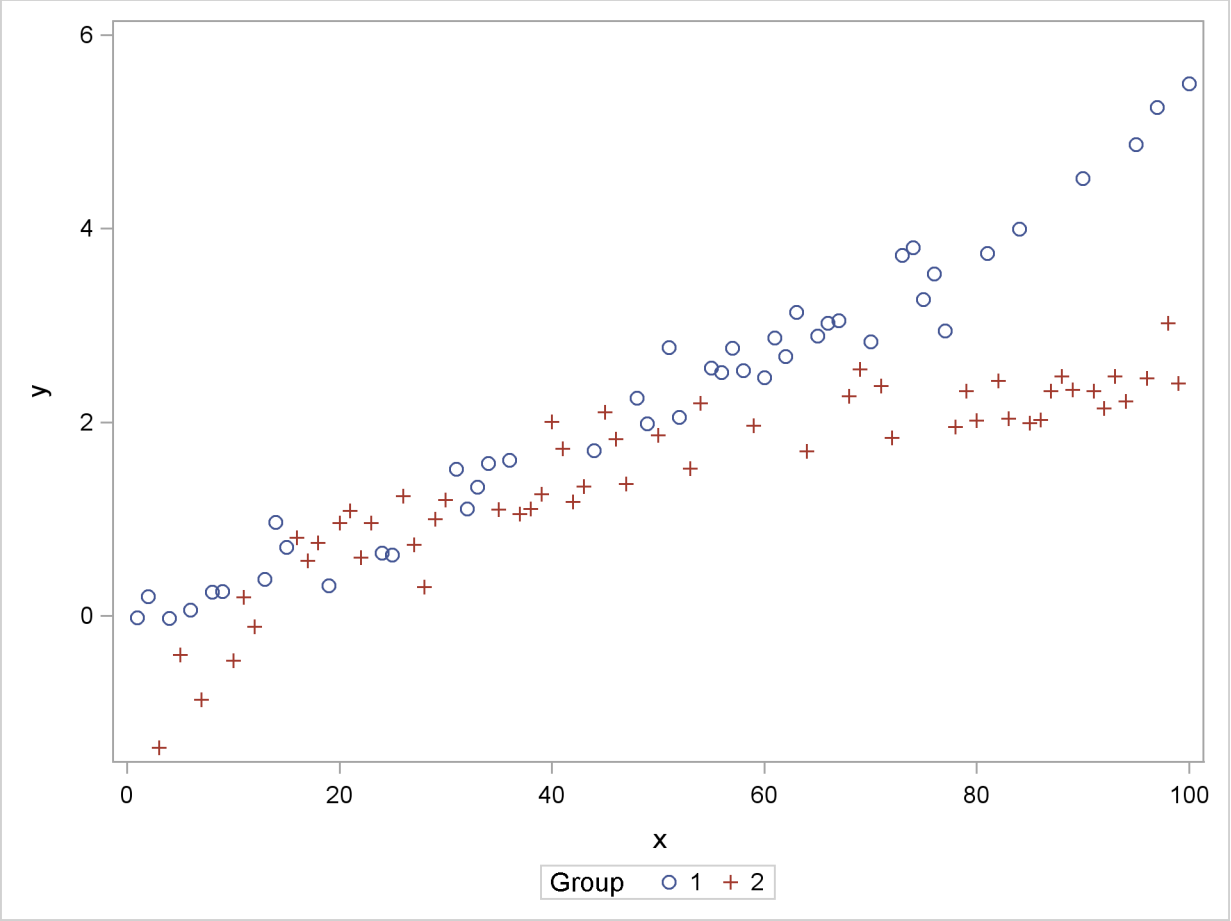
This example uses simulated data to demonstrate the use of the nonpositional syntax (see the section "Positional and Nonpositional Syntax for Contrast Coefficients" on page 2264 for details) in combination with the experimental EFFECT statement to produce interesting predictions and comparisons in models containing fixed spline effects. Consider the data in the following DATA step. Each of the 100 observations for the continuous response variable y is associated with one of two groups.

```
data spline;
  input group y @@;
  x = _n_;
  datalines;
 1    -.020 1    0.199 2    -1.36 1    -.026
 2    -.397 1    0.065 2    -.861 1    0.251
 1    0.253 2    -.460 2    0.195 2    -.108
 1    0.379 1    0.971 1    0.712 2    0.811
 2    0.574 2    0.755 1    0.316 2    0.961
 2    1.088 2    0.607 2    0.959 1    0.653
 1    0.629 2    1.237 2    0.734 2    0.299
 2    1.002 2    1.201 1    1.520 1    1.105
 1    1.329 1    1.580 2    1.098 1    1.613
 2    1.052 2    1.108 2    1.257 2    2.005
 2    1.726 2    1.179 2    1.338 1    1.707
 2    2.105 2    1.828 2    1.368 1    2.252
 1    1.984 2    1.867 1    2.771 1    2.052
 2    1.522 2    2.200 1    2.562 1    2.517
 1    2.769 1    2.534 2    1.969 1    2.460
 1    2.873 1    2.678 1    3.135 2    1.705
 1    2.893 1    3.023 1    3.050 2    2.273
 2    2.549 1    2.836 2    2.375 2    1.841
 1    3.727 1    3.806 1    3.269 1    3.533
 1    2.948 2    1.954 2    2.326 2    2.017
 1    3.744 2    2.431 2    2.040 1    3.995
 2    1.996 2    2.028 2    2.321 2    2.479
 2    2.337 1    4.516 2    2.326 2    2.144
 2    2.474 2    2.221 1    4.867 2    2.453
 1    5.253 2    3.024 2    2.403 1    5.498
;
```

The following statements produce a scatter plot of the response variable by group (Output 38.15.1):

```
proc sgplot data=spline;
   scatter y=y x=x / group=group name="data";
   keylegend "data"  / title="Group";
run;
```

Output 38.15.1 Scatter Plot of Observed Data by Group



The trends in the two groups exhibit curvature, but the type of curvature is not the same in the groups. Also, there appear to be ranges of x values where the groups are similar and areas where the point scatters separate. To model the trends in the two groups separately and with flexibility, you might want to allow for some smooth trends in x that vary by group. Consider the following PROC GLIMMIX statements:

```
proc glimmix data=spline outdesign=x;
   class group;
   effect spl = spline(x);
   model y = group spl*group / s noint;
   output out=gmxout pred=p;
run;
```

The EFFECT statement defines a constructed effect named spl by expanding the x into a spline with seven columns. The group main effect creates separate intercepts for the groups, and the interaction of the group variable with the spline effect creates separate trends. The NOINT option suppresses the intercept. This is not necessary and is done here only for convenience of interpretation. The OUTPUT statement computes predicted values.

The "Parameter Estimates" table contains the estimates of the group-specific "intercepts," the spline coefficients varied by group, and the residual variance ("Scale," Output 38.15.2).

Output 38.15.2 Parameter Estimates in Two-Group Spline Model

The GLIMMIX Procedure

Parameter Estimates

Effect	spl	group	Estimate	Standard Error	DF	t Value	Pr > |t|
group		1	9.7027	3.1342	86	3.10	0.0026
group		2	6.3062	2.6299	86	2.40	0.0187
spl*group	1	1	−11.1786	3.7008	86	−3.02	0.0033
spl*group	1	2	−20.1946	3.9765	86	−5.08	<.0001
spl*group	2	1	−9.5327	3.2576	86	−2.93	0.0044
spl*group	2	2	−5.8565	2.7906	86	−2.10	0.0388
spl*group	3	1	−8.9612	3.0718	86	−2.92	0.0045
spl*group	3	2	−5.5567	2.5717	86	−2.16	0.0335
spl*group	4	1	−7.2615	3.2437	86	−2.24	0.0278
spl*group	4	2	−4.3678	2.7247	86	−1.60	0.1126
spl*group	5	1	−6.4462	2.9617	86	−2.18	0.0323
spl*group	5	2	−4.0380	2.4589	86	−1.64	0.1042
spl*group	6	1	−4.6382	3.7095	86	−1.25	0.2146
spl*group	6	2	−4.3029	3.0479	86	−1.41	0.1616
spl*group	7	1	0	.	.	.	.
spl*group	7	2	0	.	.	.	.
Scale			0.07352	0.01121	.	.	.

Because the B-spline coefficients for an observation sum to 1 and the model contains group-specific constants, the last spline coefficient in each group is zero. In other words, you can achieve exactly the same fit with the MODEL statement

```
model y = spl*group / noint;
```

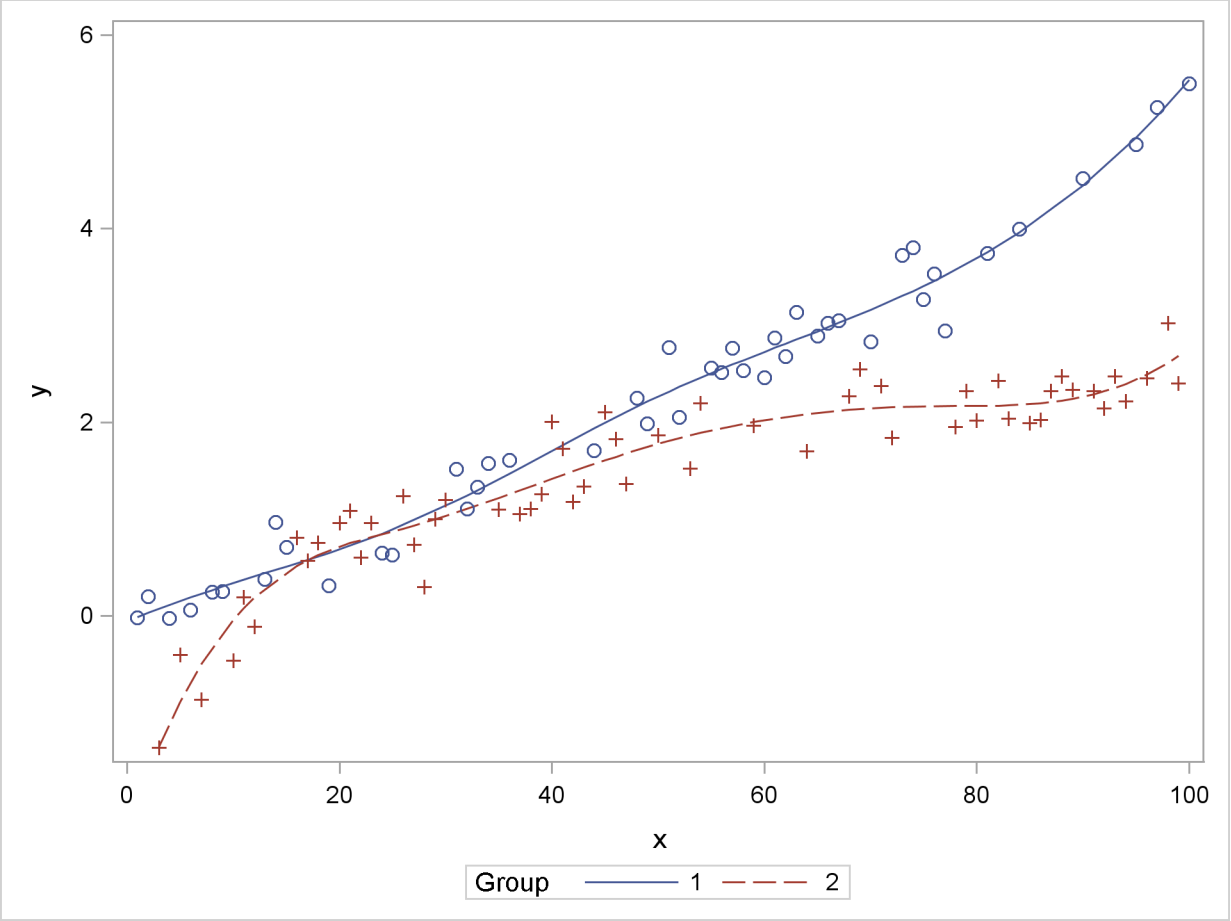
or

```
model y = spl*group;
```

The following statements graph the observed and fitted values in the two groups (Output 38.15.3):

```
proc sgplot data=gmxout;
   series  y=p x=x / group=group name="fit";
   scatter y=y x=x / group=group;
   keylegend "fit" / title="Group";
run;
```

Output 38.15.3 Observed and Predicted Values by Group



Suppose that you are interested in estimating the mean response at particular values of x and in performing comparisons of predicted values. The following program uses ESTIMATE statements with nonpositional syntax to accomplish this:

```
proc glimmix data=spline;
   class group;
   effect spl = spline(x);
   model y = group spl*group / s noint;
   estimate 'Group 1, x=20' group 1    group*spl [1,1 20] / e;
   estimate 'Group 2, x=20' group 0  1 group*spl [1,2 20];
   estimate 'Diff at x=20 ' group 1 -1 group*spl [1,1 20] [-1,2 20];
run;
```

The first ESTIMATE statement predicts the mean response at $x = 20$ in group 1. The E option requests the coefficient vector for this linear combination of the parameter estimates. The coefficient for the group effect is entered with positional (standard) syntax. The coefficients for the group*spl effect are formed based on nonpositional syntax. Because this effect comprises the interaction of a standard effect (group) with a constructed effect, the values and levels for the standard effect must precede those for the constructed effect. A similar statement produces the predicted mean at $x = 20$ in group 2.

The GLIMMIX procedure interprets the syntax

```
group*spl [1,2 20]
```

as follows: construct the spline basis at $x = 20$ as appropriate for group 2; then multiply the resulting coefficients for these columns of the $\mathbf{L}$ matrix with 1.

The final ESTIMATE statement represents the difference between the predicted values; it is a group comparison at $x = 20$.

Output 38.15.4 Coefficients from First ESTIMATE Statement

The GLIMMIX Procedure

Coefficients for Estimate
Group 1, x=20

Effect	spl	group	Row1
group		1	1
group		2	
spl*group	1	1	0.0021
spl*group	1	2	
spl*group	2	1	0.3035
spl*group	2	2	
spl*group	3	1	0.619
spl*group	3	2	
spl*group	4	1	0.0754
spl*group	4	2	
spl*group	5	1	
spl*group	5	2	
spl*group	6	1	
spl*group	6	2	
spl*group	7	1	
spl*group	7	2	

The "Coefficients" table shows how the value 20 supplied in the ESTIMATE statement was expanded into the appropriate spline basis (Output 38.15.4). There is no significant difference between the group means at $x = 20$ ($p = 0.8346$, Output 38.15.5).

Output 38.15.5 Results from ESTIMATE Statements

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|
Group 1, x=20	0.6915	0.09546	86	7.24	<.0001
Group 2, x=20	0.7175	0.07953	86	9.02	<.0001
Diff at x=20	-0.02602	0.1243	86	-0.21	0.8346

The group comparisons you can achieve in this way are comparable to slices of interaction effects with classification effects. There are, however, no preset number of levels at which to perform the comparisons because x is continuous. If you add further x values for the comparisons, a multiplicity correction is in order to control the familywise Type I error. The following statements compare the groups at values $x = 0, 5, 10, \cdots, 80$ and compute simulation-based step-down-adjusted p-values. The results appear in Output 38.15.6. (The numeric results for simulation-based p-value adjustments depend slightly on the value of the random number seed.)

```
ods select Estimates;
proc glimmix data=spline;
   class group;
   effect spl = spline(x);
   model y = group spl*group / s;
   estimate 'Diff at x= 0' group 1 -1 group*spl [1,1  0] [-1,2  0],
            'Diff at x= 5' group 1 -1 group*spl [1,1  5] [-1,2  5],
            'Diff at x=10' group 1 -1 group*spl [1,1 10] [-1,2 10],
            'Diff at x=15' group 1 -1 group*spl [1,1 15] [-1,2 15],
            'Diff at x=20' group 1 -1 group*spl [1,1 20] [-1,2 20],
            'Diff at x=25' group 1 -1 group*spl [1,1 25] [-1,2 25],
            'Diff at x=30' group 1 -1 group*spl [1,1 30] [-1,2 30],
            'Diff at x=35' group 1 -1 group*spl [1,1 35] [-1,2 35],
            'Diff at x=40' group 1 -1 group*spl [1,1 40] [-1,2 40],
            'Diff at x=45' group 1 -1 group*spl [1,1 45] [-1,2 45],
            'Diff at x=50' group 1 -1 group*spl [1,1 50] [-1,2 50],
            'Diff at x=55' group 1 -1 group*spl [1,1 55] [-1,2 55],
            'Diff at x=60' group 1 -1 group*spl [1,1 60] [-1,2 60],
            'Diff at x=65' group 1 -1 group*spl [1,1 65] [-1,2 65],
            'Diff at x=70' group 1 -1 group*spl [1,1 70] [-1,2 70],
            'Diff at x=75' group 1 -1 group*spl [1,1 75] [-1,2 75],
            'Diff at x=80' group 1 -1 group*spl [1,1 80] [-1,2 80] /
            adjust=sim(seed=1) stepdown;
run;
```

Output 38.15.6 Estimates with Multiplicity Adjustments

The GLIMMIX Procedure

Estimates
Adjustment for Multiplicity: Holm-Simulated

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	Adj P
Diff at x= 0	12.4124	4.2130	86	2.95	0.0041	0.0206
Diff at x= 5	1.0376	0.1759	86	5.90	<.0001	<.0001
Diff at x=10	0.3778	0.1540	86	2.45	0.0162	0.0545
Diff at x=15	0.05822	0.1481	86	0.39	0.6952	0.9101
Diff at x=20	-0.02602	0.1243	86	-0.21	0.8346	0.9565
Diff at x=25	0.02014	0.1312	86	0.15	0.8783	0.9565
Diff at x=30	0.1023	0.1378	86	0.74	0.4600	0.7418
Diff at x=35	0.1924	0.1236	86	1.56	0.1231	0.2925
Diff at x=40	0.2883	0.1114	86	2.59	0.0113	0.0450
Diff at x=45	0.3877	0.1195	86	3.24	0.0017	0.0096
Diff at x=50	0.4885	0.1308	86	3.74	0.0003	0.0024
Diff at x=55	0.5903	0.1231	86	4.79	<.0001	<.0001
Diff at x=60	0.7031	0.1125	86	6.25	<.0001	<.0001
Diff at x=65	0.8401	0.1203	86	6.99	<.0001	<.0001
Diff at x=70	1.0147	0.1348	86	7.52	<.0001	<.0001
Diff at x=75	1.2400	0.1326	86	9.35	<.0001	<.0001
Diff at x=80	1.5237	0.1281	86	11.89	<.0001	<.0001

There are significant differences at the low end and high end of the x range. Notice that without the multiplicity adjustment you would have concluded at the 0.05 level that the groups are significantly different at $x = 10$. At the 0.05 level, the groups separate significantly for $x < 10$ and $x > 40$.

Example 38.16: Diallel Experiment with Multimember Random Effects

Cockerham and Weir (1977) apply variance component models in the analysis of reciprocal crosses. In these experiments it is of interest to separate genetically determined variation from variation determined by parentage. We analyze here the data for the diallel experiment in Cockerham and Weir (1977, Appendix C). A diallel is a mating design that consists of all possible crosses of a set of parental lines. It includes reciprocal crossings, but not self-crossings.

The basic model for a cross is $Y_{ijk} = \beta + \alpha_{ij} + \epsilon_{ijk}$, where Y_{ijk} is the observation for offspring k from maternal parent i and paternal parent j. The various models in Cockerham and Weir (1977) are different decompositions of the term α_{ij}, the total effect that is due to the parents. Their "bio model" (model (c)) decomposes α_{ij} into

$$\alpha_{ij} = \eta_i + \eta_j + \mu_i + \phi_j + (\eta\eta)_{ij} + \kappa_{ij}$$

where η_i and η_j are contributions of the female and male parents, respectively. The term $(\eta\eta)_{ij}$ captures the interaction between maternal and paternal effects. In contrast to usual interaction effects, this term must obey a symmetry because of the reciprocals: $(\eta\eta)_{ij} = (\eta\eta)_{ji}$. The terms

μ_i and ϕ_j in the decomposition are extranuclear maternal and paternal effects, and the remaining interactions are captured by the κ_{ij} term.

The following DATA step creates a SAS data set for the diallel example in Appendix C of Cockerham and Weir (1977):

```
data diallel;
   label time = 'Flowering time in days';
   do p = 1 to 8;
      do m = 1 to 8;
         if (m ne p) then do;
            sym = trim(left(min(m,p))) || ',' || trim(left(max(m,p)));
            do block = 1 to 2;
               input time @@;
               output;
            end;
         end;
      end;
   end;
   datalines;
14.4 16.2 27.2 30.8 17.2 27.0 18.3 20.2 16.2 16.8 18.6 14.4 16.4 16.0
15.4 16.5 14.8 14.6 18.6 18.6 15.2 15.3 17.0 15.2 14.4 14.8 10.8 13.2
31.8 30.4 21.0 23.0 24.6 25.4 19.2 20.0 29.8 28.4 12.8 14.2 13.0 14.4
16.2 17.8 11.4 13.0 16.8 16.3 12.4 14.2 16.8 14.8 12.6 12.2  9.6 11.2
14.6 18.8 12.2 13.6 15.2 15.4 15.2 13.8 18.0 16.0 10.4 12.2 13.4 20.0
20.2 23.4 14.2 14.0 18.6 14.8 22.2 17.0 14.3 17.3  9.0 10.2 11.8 12.8
14.0 16.6 12.2  9.2 13.6 16.2 13.8 14.4 15.6 15.6 15.6 11.0 13.0  9.8
15.2 17.2 10.0 11.6 17.0 18.2 20.8 20.8 20.0 17.4 17.0 12.6 13.0  9.8
;
```

The observations represent mean flowering times of *Nicotiana rustica* (Aztec tobacco) from crosses of inbred varieties grown in two blocks. The variables p and m identify the eight paternal and maternal lines, respectively. The variable sym is used to model the interaction between the parents, subject to the symmetry condition $(\eta\eta)_{ij} = (\eta\eta)_{ji}$. For example, the first two observations, 14.4 and 16.2 days, represent the observations from blocks 1 and 2 where paternal line 1 was crossed with maternal line 2.

The following PROC GLIMMIX statements fit the "bio model" in Cockerham and Weir (1977):

```
proc glimmix data=diallel outdesign(z)=zmat;
   class block sym p m;
   effect line = mm(p m);
   model  time = block;
   random line sym p m p*m;
run;
```

The EFFECT statement defines the nuclear parental contributions as a multimember effect based on the CLASS variables p and m. Each observation has two nonzero entries in the design matrix for the effect that identifies the paternal and maternal lines. The terms in the RANDOM statement model the variance components as follows: line $\rightarrow \sigma^2_n$, sym $\rightarrow \sigma^2_{(\eta\eta)}$, p $\rightarrow \sigma^2_\phi$, m $\rightarrow \sigma^2_\mu$, p*m $\rightarrow \sigma^2_\kappa$. The OUTDESIGN= option in the PROC GLIMMIX statement writes the $\mathbf{Z}$ matrix to the SAS data set zmat. The EFFECT statement alleviates the need for complex coding, as in Section 2.3 of Saxton (2004).

Output 38.16.1 displays the "Class Level Information" table of the diallel model. Because the interaction terms are symmetric, there are only $8 \times 7/2 = 28$ levels for the 8 lines. The estimates of the variance components and the residual variance in Output 38.16.1 agree with the results in Table 7 of Cockerham and Weir (1977).

Output 38.16.1 Class Levels and Covariance Parameter Estimates in Diallel Example

The GLIMMIX Procedure

Class Level Information

Class	Levels	Values
block	2	1 2
sym	28	1,2 1,3 1,4 1,5 1,6 1,7 1,8 2,3 2,4 2,5 2,6 2,7 2,8 3,4 3,5 3,6 3,7 3,8 4,5 4,6 4,7 4,8 5,6 5,7 5,8 6,7 6,8 7,8
p	8	1 2 3 4 5 6 7 8
m	8	1 2 3 4 5 6 7 8

Covariance Parameter Estimates

Cov Parm	Estimate	Standard Error
line	5.1047	4.0021
sym	2.3856	1.9025
p	3.3080	3.4053
m	1.9134	2.9891
p*m	4.0196	1.8323
Residual	3.6225	0.6908

The following statements print the **Z** matrix columns that correspond to the multimember line effect for the first 10 observations in block 1 (Output 38.16.2). For each observation there are two nonzero entries, and their column index corresponds to the index of the paternal and maternal line.

```
proc print data=zmat(where=(block=1) obs=10);
  var p m time _z1-_z8;
run;
```

Output 38.16.2 **Z** Matrix for Line Effect of the First 10 Observations in Block 1

Obs	p	m	time	_Z1	_Z2	_Z3	_Z4	_Z5	_Z6	_Z7	_Z8
1	1	2	14.4	1	1	0	0	0	0	0	0
3	1	3	27.2	1	0	1	0	0	0	0	0
5	1	4	17.2	1	0	0	1	0	0	0	0
7	1	5	18.3	1	0	0	0	1	0	0	0
9	1	6	16.2	1	0	0	0	0	1	0	0
11	1	7	18.6	1	0	0	0	0	0	1	0
13	1	8	16.4	1	0	0	0	0	0	0	1
15	2	1	15.4	1	1	0	0	0	0	0	0
17	2	3	14.8	0	1	1	0	0	0	0	0
19	2	4	18.6	0	1	0	1	0	0	0	0

Example 38.17: Linear Inference Based on Summary Data

The GLIMMIX procedure has facilities for multiplicity-adjusted inference through the ADJUST= and STEPDOWN options in the ESTIMATE, LSMEANS, and LSMESTIMATE statements. You can employ these facilities to test linear hypotheses among parameters even in situations where the quantities were obtained outside the GLIMMIX procedure. This example demonstrates the process. The basic idea is to prepare a data set containing the estimates of interest and a data set containing their covariance matrix. These are then passed to the GLIMMIX procedure, preventing updating of the parameters, essentially moving directly into the post-processing stage as if estimates with this covariance matrix had been produced by the GLIMMIX procedure.

The final documentation example in Chapter 60, "The NLIN Procedure," in the *SAS/STAT User's Guide* discusses a nonlinear first-order compartment pharmacokinetic model for theophylline concentration. The data are derived by collapsing and averaging the subject-specific data from Pinheiro and Bates (1995) in a particular—yet unimportant—way that leads to two groups for comparisons. The following DATA step creates these data:

```
data theop;
  input time dose conc @@;
  if (dose = 4) then group=1; else group=2;
  datalines;
 0.00   4  0.1633  0.25   4   2.045
 0.27   4     4.4  0.30   4    7.37
 0.35   4    1.89  0.37   4    2.89
 0.50   4    3.96  0.57   4    6.57
 0.58   4     6.9  0.60   4     4.6
 0.63   4    9.03  0.77   4    5.22
 1.00   4    7.82  1.02   4   7.305
 1.05   4    7.14  1.07   4     8.6
 1.12   4    10.5  2.00   4    9.72
 2.02   4    7.93  2.05   4    7.83
 2.13   4    8.38  3.50   4    7.54
 3.52   4    9.75  3.53   4    5.66
 3.55   4   10.21  3.62   4     7.5
 3.82   4    8.58  5.02   4   6.275
 5.05   4    9.18  5.07   4    8.57
 5.08   4     6.2  5.10   4    8.36
 7.02   4    5.78  7.03   4    7.47
 7.07   4   5.945  7.08   4    8.02
 7.17   4    4.24  8.80   4    4.11
 9.00   4     4.9  9.02   4    5.33
 9.03   4    6.11  9.05   4    6.89
 9.38   4    7.14 11.60   4    3.16
11.98   4    4.19 12.05   4    4.57
12.10   4    5.68 12.12   4    5.94
12.15   4     3.7 23.70   4    2.42
24.15   4    1.17 24.17   4    1.05
24.37   4    3.28 24.43   4    1.12
24.65   4    1.15  0.00   5   0.025
 0.25   5    2.92  0.27   5   1.505
```

```
 0.30   5    2.02  0.50   5   4.795
 0.52   5    5.53  0.58   5    3.08
 0.98   5   7.655  1.00   5   9.855
 1.02   5    5.02  1.15   5    6.44
 1.92   5    8.33  1.98   5    6.81
 2.02   5  7.8233  2.03   5    6.32
 3.48   5    7.09  3.50   5   7.795
 3.53   5    6.59  3.57   5    5.53
 3.60   5    5.87  5.00   5     5.8
 5.02   5  6.2867  5.05   5    5.88
 6.98   5    5.25  7.00   5    4.02
 7.02   5    7.09  7.03   5   4.925
 7.15   5    4.73  9.00   5    4.47
 9.03   5    3.62  9.07   5    4.57
 9.10   5     5.9  9.22   5    3.46
12.00   5    3.69 12.05   5    3.53
12.10   5    2.89 12.12   5    2.69
23.85   5    0.92 24.08   5    0.86
24.12   5    1.25 24.22   5    1.15
24.30   5     0.9 24.35   5    1.57
;
```

In terms of two fixed treatment groups, the nonlinear model for these data can be written as

$$C_{it} = \frac{Dk_{e_i}k_{a_i}}{Cl_i(k_{a_i} - k_{e_i})}[\exp(-k_{e_i}t) - \exp(-k_{a_i}t)] + \epsilon_{it}$$

where C_{it} is the observed concentration in group i at time t, D is the dose of theophylline, k_{e_i} is the elimination rate constant in group i, k_{a_i} is the absorption rate in group i, Cl_i is the clearance in group i, and ϵ_{it} denotes the model error. Because the rates and the clearance must be positive, you can parameterize the model in terms of log rates and the log clearance:

$$\begin{aligned} Cl_i &= \exp\{\beta_{1i}\} \\ k_{a_i} &= \exp\{\beta_{2i}\} \\ k_{e_i} &= \exp\{\beta_{3i}\} \end{aligned}$$

In this parameterization the model contains six parameters, and the rates and clearance vary by group. The following PROC NLIN statements fit the model and obtain the group-specific parameter estimates:

```
proc nlin data=theop outest=cov;
   parms beta1_1=-3.22 beta2_1=0.47 beta3_1=-2.45
         beta1_2=-3.22 beta2_2=0.47 beta3_2=-2.45;
   if (group=1) then do;
      cl   = exp(beta1_1);
      ka   = exp(beta2_1);
      ke   = exp(beta3_1);
   end; else do;
      cl   = exp(beta1_2);
      ka   = exp(beta2_2);
```

```
      ke    = exp(beta3_2);
   end;
   mean = dose*ke*ka*(exp(-ke*time)-exp(-ka*time))/cl/(ka-ke);
   model conc = mean;
   ods output ParameterEstimates=ests;
run;
```

The conditional programming statements determine the clearance, elimination, and absorption rates depending on the value of the group variable. The OUTEST= option in the PROC NLIN statement saves estimates and their covariance matrix to the data set cov. The ODS OUTPUT statement saves the "Parameter Estimates" table to the data set ests.

Output 38.17.1 displays the analysis of variance table and the parameter estimates from this NLIN run. Note that the confidence levels in the "Parameter Estimates" table are based on 92 degrees of freedom, corresponding to the residual degrees of freedom in the analysis of variance table.

Output 38.17.1 Analysis of Variance and Parameter Estimates for Nonlinear Model

The NLIN Procedure

NOTE: An intercept was not specified for this model.

Source	DF	Sum of Squares	Mean Square	F Value	Approx Pr > F
Model	6	3247.9	541.3	358.56	<.0001
Error	92	138.9	1.5097		
Uncorrected Total	98	3386.8			

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits	
beta1_1	-3.5671	0.0864	-3.7387	-3.3956
beta2_1	0.4421	0.1349	0.1742	0.7101
beta3_1	-2.6230	0.1265	-2.8742	-2.3718
beta1_2	-3.0111	0.1061	-3.2219	-2.8003
beta2_2	0.3977	0.1987	0.00305	0.7924
beta3_2	-2.4442	0.1618	-2.7655	-2.1229

The following DATA step extracts the part of the cov data set that contains the covariance matrix of the parameter estimates in Output 38.17.1 and renames the variables as Col1–Col6. Output 38.17.2 shows the result of the DATA step.

```
data covb;
    set cov(where=(_type_='COVB'));
    rename beta1_1=col1 beta2_1=col2 beta3_1=col3
           beta1_2=col4 beta2_2=col5 beta3_2=col6;
    row = _n_;
    Parm = 1;
    keep parm row beta:;
run;

proc print data=covb;
run;
```

Output 38.17.2 Covariance Matrix of NLIN Parameter Estimates

Obs	col1	col2	col3	col4	col5	col6	row	Parm
1	0.007462	-0.005222	0.010234	0.000000	0.000000	0.000000	1	1
2	-0.005222	0.018197	-0.010590	0.000000	0.000000	0.000000	2	1
3	0.010234	-0.010590	0.015999	0.000000	0.000000	0.000000	3	1
4	0.000000	0.000000	0.000000	0.011261	-0.009096	0.015785	4	1
5	0.000000	0.000000	0.000000	-0.009096	0.039487	-0.019996	5	1
6	0.000000	0.000000	0.000000	0.015785	-0.019996	0.026172	6	1

The reason for this transformation of the data is to use the resulting data set to define a covariance structure in PROC GLIMMIX. The following statements reconstitute a model in which the parameter estimates from PROC NLIN are the observations and in which the covariance matrix of the "observations" matches the covariance matrix of the NLIN parameter estimates:

```
proc glimmix data=ests order=data;
   class Parameter;
   model Estimate = Parameter / noint df=92 s;
   random _residual_ / type=lin(1) ldata=covb v;
   parms (1) / noiter;
   lsmeans parameter / cl;
   lsmestimate Parameter
               'beta1 eq. across groups' 1 0 0 -1,
               'beta2 eq. across groups' 0 1 0  0 -1,
               'beta3 eq. across groups' 0 0 1  0  0  -1 /
               adjust=bon stepdown ftest(label='Homogeneity');
run;
```

In other words, you are using PROC GLIMMIX to set up a linear statistical model

$$\mathbf{Y} = \mathbf{I}\boldsymbol{\alpha} + \boldsymbol{\epsilon}$$
$$\boldsymbol{\epsilon} \sim (\mathbf{0}, \mathbf{A})$$

where the covariance matrix $\mathbf{A}$ is given by

$$\mathbf{A} = \begin{bmatrix} 0.007 & -0.005 & 0.010 & 0 & 0 & 0 \\ -0.005 & 0.018 & -0.011 & 0 & 0 & 0 \\ 0.010 & -0.011 & 0.016 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.011 & -0.009 & 0.016 \\ 0 & 0 & 0 & -0.009 & 0.039 & -0.019 \\ 0 & 0 & 0 & 0.016 & -0.019 & 0.026 \end{bmatrix}$$

The generalized least squares estimate for $\boldsymbol{\alpha}$ in this saturated model reproduces the observations:

$$\begin{aligned} \widehat{\boldsymbol{\alpha}} &= \left(\mathbf{I}'\mathbf{A}^{-1}\mathbf{I}\right)^{-1}\mathbf{I}'\mathbf{A}^{-1}\mathbf{y} \\ &= \left(\mathbf{A}^{-1}\right)^{-1}\mathbf{A}^{-1}\mathbf{y} \\ &= \mathbf{y} \end{aligned}$$

The ORDER=DATA option in the PROC GLIMMIX statement requests that the sort order of the Parameter variable be identical to the order in which it appeared in the "Parameter Estimates" table of the NLIN procedure (Output 38.17.1). The MODEL statement uses the Estimate and Parameter variables from that table to form a model in which the $\mathbf{X}$ matrix is the identity; hence the NOINT option. The DF=92 option sets the degrees of freedom equal to the value used in the NLIN procedure. The RANDOM statement specifies a linear covariance structure with a single component and supplies the values for the structure through the LDATA= data set. This structure models the covariance matrix as $\text{Var}[\mathbf{Y}] = \theta\mathbf{A}$, where the $\mathbf{A}$ matrix is given previously. Essentially, the TYPE=LIN(1) structure forces an unstructured covariance matrix onto the data. To make this work, the parameter θ is held fixed at 1 in the PARMS statement.

Output 38.17.3 displays the parameter estimates and least squares means for this model. Note that estimates and least squares means are identical, since the $\mathbf{X}$ matrix is the identity. Also, the confidence limits agree with the values reported by PROC NLIN (see Output 38.17.1).

Output 38.17.3 Parameter Estimates and LS-Means from Summary Data

The GLIMMIX Procedure

Solutions for Fixed Effects

Effect	Parameter	Estimate	Standard Error	DF	t Value	Pr > |t|
Parameter	beta1_1	-3.5671	0.08638	92	-41.29	<.0001
Parameter	beta2_1	0.4421	0.1349	92	3.28	0.0015
Parameter	beta3_1	-2.6230	0.1265	92	-20.74	<.0001
Parameter	beta1_2	-3.0111	0.1061	92	-28.37	<.0001
Parameter	beta2_2	0.3977	0.1987	92	2.00	0.0483
Parameter	beta3_2	-2.4442	0.1618	92	-15.11	<.0001

Parameter Least Squares Means

Parameter	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha
beta1_1	-3.5671	0.08638	92	-41.29	<.0001	0.05
beta2_1	0.4421	0.1349	92	3.28	0.0015	0.05
beta3_1	-2.6230	0.1265	92	-20.74	<.0001	0.05
beta1_2	-3.0111	0.1061	92	-28.37	<.0001	0.05
beta2_2	0.3977	0.1987	92	2.00	0.0483	0.05
beta3_2	-2.4442	0.1618	92	-15.11	<.0001	0.05

Parameter Least Squares Means

Parameter	Lower	Upper
beta1_1	-3.7387	-3.3956
beta2_1	0.1742	0.7101
beta3_1	-2.8742	-2.3718
beta1_2	-3.2219	-2.8003
beta2_2	0.003050	0.7924
beta3_2	-2.7655	-2.1229

The (marginal) covariance matrix of the data is shown in Output 38.17.4 to confirm that it matches the **A** matrix given earlier.

Output 38.17.4 R-Side Covariance Matrix

Estimated V Matrix for Subject 1

Row	Col1	Col2	Col3	Col4	Col5	Col6
1	0.007462	-0.00522	0.01023			
2	-0.00522	0.01820	-0.01059			
3	0.01023	-0.01059	0.01600			
4				0.01126	-0.00910	0.01579
5				-0.00910	0.03949	-0.02000
6				0.01579	-0.02000	0.02617

The LSMESTIMATE statement specifies three linear functions. These set equal the β parameters from the groups. The step-down Bonferroni adjustment requests a multiplicity adjustment for the family of three tests. The FTEST option requests a joint test of the three estimable functions; it is a global test of parameter homogeneity across groups.

Output 38.17.5 displays the result from the LSMESTIMATE statement. The joint test is highly significant ($F = 30.52$, $p < 0.0001$). From the p-values associated with the individual rows of the estimates, you can see that the lack of homogeneity is due to group differences for β_1, the log clearance.

Output 38.17.5 Test of Parameter Homogeneity across Groups

Least Squares Means Estimates
Adjustment for Multiplicity: Holm

Effect	Label	Estimate	Standard Error	DF	t Value	Pr > |t|
Parameter	beta1 eq. across groups	-0.5560	0.1368	92	-4.06	0.0001
Parameter	beta2 eq. across groups	0.04443	0.2402	92	0.18	0.8537
Parameter	beta3 eq. across groups	-0.1788	0.2054	92	-0.87	0.3862

Least Squares Means Estimates
Adjustment for Multiplicity: Holm

Effect	Label	Adj P
Parameter	beta1 eq. across groups	0.0003
Parameter	beta2 eq. across groups	0.8537
Parameter	beta3 eq. across groups	0.7725

Least Squares Means Ftest

Label	Num DF	Den DF	F Value	Pr > F
Homogeneity	3	92	30.52	<.0001

An alternative method to set up this model is given by the following statements, where the data set pdata contains the covariance parameters:

```
random _residual_ / type=un;
parms / pdata=pdata noiter
```

The following DATA step creates an appropriate PDATA= data set from the data set covb constructed earlier:

```
data pdata; set covb;
   array col{6};
   do i=1 to _n_;
      estimate = col{i};
      output;
   end;
   keep estimate;
run;
```

References

Abramowitz, M. and Stegun, I. A. (1972), *Handbook of Mathematical Functions*, New York: Dover Publications.

Akaike, H. (1974), "A New Look at the Statistical Model Identification," *IEEE Transaction on Automatic Control*, AC–19, 716–723.

Bahadur, R. R. (1961), "A Representation of the Joint Distribution of Responses to n Dichotomous Items," in *Studies in Item Analysis and Prediction*, ed. H. Solomon, Stanford, CA: Stanford University Press.

Beale, E. M. L. (1972), "A Derivation of Conjugate Gradients," in *Numerical Methods for Nonlinear Optimization*, ed. F. A. Lootsma, London: Academic Press.

Bell, R. M. and McCaffrey, D. F. (2002), "Bias Reduction in Standard Errors for Linear Regression with Multi-stage Samples," *Survey Methodology*, 28, 169–181.

Bickel, P. J. and Doksum, K. A. (1977), "Mathematical Statistics," Oakland, CA: Holden-Day.

Booth, J. G. and Hobert, J. P. (1998) "Standard Errors of Prediction in Generalized Linear Mixed Models," *Journal of the American Statistical Association,* 93, 262–272.

Bozdogan, H. (1987), "Model Selection and Akaike's Information Criterion (AIC): The General Theory and Its Analytical Extensions," *Psychometrika*, 52, 345–370.

Breslow, N. E. and Clayton, D. G. (1993), "Approximate Inference in Generalized Linear Mixed Models," *Journal of the American Statistical Association,* 88, 9–25.

Breslow, N. E. and Lin, X. (1995), "Bias Correction in Generalised Linear Mixed Models with a Single Component of Dispersion," *Biometrika*, 81, 81–91.

Brinkman, N. D. (1981), "Ethanol Fuel—A Single Engine Study of Efficiency and Exhaust Emission," *SAE Transactions,* 90, No. 810345, 1410–1424.

Brown, H. and Prescott. R. (1999), *Applied Mixed Models in Medicine,* New York: John Wiley & Sons.

Burdick, R. K. and Graybill, F. A. (1992), *Confidence Intervals on Variance Components,* New York: Marcel Dekker.

Burnham, K. P. and Anderson, D. R. (1998), *Model Selection and Inference: A Practical Information-Theoretic Approach,* New York: Springer-Verlag.

Cameron, A. C. and Trivedi, P. K. (1998), *Regression Analysis of Count Data*, Cambridge: Cambridge University Press.

Clayton, D. and Kaldor, J. (1987) "Empirical Bayes Estimates of Age-Standardized Relative Risks for Use in Disease Mapping," *Biometrics*, 43, 671–681.

Cleveland, W. S. and Grosse, E. (1991) "Computational Methods for Local Regression," *Statistics and Computing*, 1, 47–62.

Cockerham, C. C., and Weir, B. S. (1977), "Quadratic Analyses of Reciprocal Crosses," *Biometrics*, 33, 187–203.

Davis, A. W. (1977), "A Differential Equation Approach to Linear Combinations of Independent Chi-Squares," *Journal of the American Statistical Association*, 72, 212–214.

de Boor, C. (2001), *A Practical Guide to Splines*, Revised Edition, New York: Springer-Verlag.

Dennis, J. E., Gay, D. M., and Welsch, R. E. (1981), "An Adaptive Nonlinear Least-Squares Algorithm," *ACM Transactions on Mathematical Software*, 7, 348–368.

Dennis, J. E. and Mei, H. H. W. (1979), "Two New Unconstrained Optimization Algorithms Which Use Function and Gradient Values," *J. Optim. Theory Appl.*, 28, 453–482.

Dennis, J. E. and Schnabel, R. B. (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations,* Englewood Cliffs, NJ: Prentice-Hall.

Diggle, P. J., Liang, K.-Y., and Zeger, S. L. (1994), *Analysis of Longitudinal Data,* Oxford, UK: Oxford University Press.

Dunnett, C. W. (1980), "Pairwise Multiple Comparisons in the Unequal Variance Case," *Journal of the American Statistical Association*, 75, 796–800.

Edwards, D. and Berry, J. J. (1987), "The Efficiency of Simulation-Based Multiple Comparisons," *Biometrics*, 43, 913–928.

Eilers, P. H. C. and Marx, B. D. (1996), "Flexible Smoothing with *B*-Splines and Penalties," *Statistical Science*, 11, 89–121.

Eskow, E. and Schnabel, R. B. (1991), "Algorithm 695: Software for a New Modified Cholesky Factorization," *Transactions on Mathematical Software*, 17(3), 306–312.

Evans, G. (1993), *Practical Numerical Integration*, New York: John Wiley & Sons.

Fai, A. H. T. and Cornelius, P. L. (1996), "Approximate *F*-Tests of Multiple Degree of Freedom Hypotheses in Generalized Least Squares Analyses of Unbalanced Split-Plot Experiments," *Journal of Statistical Computation and Simulation,* 54, 363–378.

Fay, M. P. and Graubard, B. I. (2001), "Small-Sample Adjustments for Wald-Type Tests Using Sandwich Estimators," *Biometrics*, 57, 1198–1206.

Ferrari. S. L. P. and Cribari-Neto, F. (2004), "Beta Regression for Modelling Rates and Proportions," *Journal of Applied Statistics*, 31, 799–815.

Fisher, R. A. (1936), "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, 7, 179–188.

Fletcher, R. (1987), *Practical Methods of Optimization,* Second Edition, Chichester: John Wiley & Sons.

Friedman, J. H., Bentley, J. L., and Finkel, R. A. (1977) "An Algorithm for Finding Best Matches in Logarithmic Expected Time," *ACM Transactions on Mathematical Software*, 3, 209–226.

Fuller, W. A. (1976), *Introduction to Statistical Time Series,* New York: John Wiley & Sons.

Games, P. A., and Howell, J. F. (1976), "Pairwise Multiple Comparison Procedures with Unequal *n*'s and/or Variances: A Monte Carlo Study," *Journal of Educational Statistics*, 1, 113–125.

Gay, D. M. (1983), "Subroutines for Unconstrained Minimization," *ACM Transactions on Mathematical Software*, 9, 503–524.

Giesbrecht, F. G. and Burns, J. C. (1985), "Two-Stage Analysis Based on a Mixed Model: Large-Sample Asymptotic Theory and Small-Sample Simulation Results," *Biometrics*, 41, 477–486.

Gilliland, D. and Schabenberger, O. (2001), "Limits on Pairwise Association for Equi-Correlated Binary Variables," *Journal of Applied Statistical Sciences*, 10, 279–285.

Gilmour, A. R., Anderson, R. D., and Rae, A. L. (1987) "Variance Components on an Underlying Scale for Ordered Multiple Threshold Categorical Data Using a Generalized Linear Mixed Model," *Journal of Animal Breeding and Genetics,* 104, 149–155.

Golub, G. H., and Welsch, J. H. (1969), "Calculation of Gaussian Quadrature Rules," *Mathematical Computing,* 23, 221–230.

Goodnight, J. H. (1978a), SAS Technical Report R-101, *Tests of Hypotheses in Fixed-Effects Linear Models*, Cary, NC: SAS Institute Inc.

Goodnight, J. H. (1978b), SAS Technical Report R-105, *Computing MIVQUE0 Estimates of Variance Components*, Cary, NC: SAS Institute Inc.

Goodnight, J. H. (1979), "A Tutorial on the Sweep Operator," *American Statistician*, 33, 149–158.

Goodnight, J. H. and Hemmerle, W. J. (1979), "A Simplified Algorithm for the W-Transformation in Variance Component Estimation," *Technometrics*, 21, 265–268.

Gotway, C. A. and Stroup, W. W. (1997), "A Generalized Linear Model Approach to Spatial Data and Prediction," *Journal of Agricultural, Biological, and Environmental Statistics,* 2, 157–187.

Guirguis, G. H. and Tobias, R. D. (2004), "On the Computation of the Distribution for the Analysis of Means," *Communications in Statistics: Simulation and Computation,* 33, 861–888.

Hand, D. J., Daly, F., Lunn, A. D., McConway, K. J., and Ostrowski, E. (1994), *A Handbook of Small Data Sets*, London: Chapman & Hall.

Handcock, M. S. and Stein, M. L. (1993), "A Bayesian Analysis of Kriging," *Technometrics*, 35(4), 403–410.

Handcock, M. S. and Wallis, J. R. (1994), "An Approach to Statistical Spatial-Temporal Modeling of Meteorological Fields (with Discussion)," *Journal of the American Statistical Association*, 89, 368–390.

Hannan, E. J. and Quinn, B. G. (1979), "The Determination of the Order of an Autoregression," *Journal of the Royal Statistical Society*, Series B, 41, 190–195.

Harville, D. A. and Jeske, D. R. (1992), "Mean Squared Error of Estimation or Prediction under a General Linear Model," *Journal of the American Statistical Association*, 87, 724–731.

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning,* New York: Springer-Verlag.

Hemmerle, W. J. and Hartley, H. O. (1973), "Computing Maximum Likelihood Estimates for the Mixed AOV Model Using the W-Transformation," *Technometrics*, 15, 819–831.

Henderson, C. R. (1984), *Applications of Linear Models in Animal Breeding*, University of Guelph.

Hinkley, D. V. (1977), "Jackknifing in Unbalanced Situations," *Technometrics*, 19, 285–292.

Hirotsu, C. and Srivastava, M. (2000), "Simultaneous Confidence Intervals Based on One-Sided Max t Test," *Statistics and Probability Letters,* 49, 25–37.

Holm, S. (1979), "A Simple Sequentially Rejective Multiple Test Procedure," *Scandinavian Journal of Statistics*, 6, 65 –70.

Hsu, J. C. (1992), "The Factor Analytic Approach to Simultaneous Inference in the General Linear Model," *Journal of Computational and Graphical Statistics*, 1, 151–168.

Hsu, J. C. (1996), *Multiple Comparisons. Theory and Methods*, London: Chapman & Hall.

Hsu, J. C. and Peruggia, M. (1994), "Graphical Representation of Tukey's Multiple Comparison Method," *Journal of Computational and Graphical Statistics*, 3: 143–161.

Huber, P. J. (1967), "The Behavior of Maximum Likelihood Estimates under Nonstandard Conditions," *Proc. Fifth Berkeley Symp. Math. Statist. Prob.*, 1, 221–233.

Hurvich, C. M. and Tsai, C.-L. (1989), "Regression and Time Series Model Selection in Small Samples," *Biometrika*, 76, 297–307.

Huynh, H. and Feldt, L. S. (1970), "Conditions under Which Mean Square Ratios in Repeated Measurements Designs Have Exact F-Distributions," *Journal of the American Statistical Association*, 65, 1582–1589.

Jennrich, R. I. and Schluchter, M. D. (1986), "Unbalanced Repeated-Measures Models with Structured Covariance Matrices," *Biometrics,* 42, 805–820.

Joe, H. and Zhu, R. (2005), "Generalized Poisson Distribution: The Property of Mixture of Poisson and Comparison with Negative Binomial Distribution," *Biometrical Journal*, 47, 219–229.

Johnson, N. L., Kotz, S., and Balakrishnan, N. (1994), *Continuous Univariate Distributions, Volume 1*, Second Edition, New York: John Wiley & Sons.

Kackar, R. N. and Harville, D. A. (1984), "Approximations for Standard Errors of Estimators of Fixed and Random Effects in Mixed Linear Models," *Journal of the American Statistical Association*, 79, 853–862.

Kahaner, D., Moler, C., and Nash, S. (1989), *Numerical Methods and Software*, Englewood Cliffs, NJ: Prentice-Hall.

Karim, M. Z. and Zeger, S. L. (1992), "Generalized Linear Models with Random Effects; Salamander Mating Revisited," *Biometrics,* 48, 631–644.

Kass, R. E. and Steffey, D. (1989), "Approximate Bayesian Inference in Conditionally Independent Hierarchical Models (Parametric Empirical Bayes Models)," *Journal of the American Statistical Association*, 84, 717–726.

Kauermann, G. and Carroll, R. J. (2001), "A Note on the Efficiency of Sandwich Covariance Estimation," *Journal of the American Statistical Association*, 96, 1387–1396.

Kenward, M. G. (1987), "A Method for Comparing Profiles of Repeated Measurements," *Applied Statistics*, 36, 296–308.

Kenward, M. G. and Roger, J. H. (1997), "Small Sample Inference for Fixed Effects from Restricted Maximum Likelihood," *Biometrics*, 53, 983–997.

Koch, G. G., Carr, G. J., Amara, I. A., Stokes, M. E., and Uryniak, T. J. (1990), "Categorical Data Analysis," Ch. 13 in *Statistical Methodology in the Pharmaceutical Sciences,* Donald A. Berry, ed., New York: Marcel Dekker.

Kramer, C. Y. (1956), "Extension of Multiple Range Tests to Group Means with Unequal Numbers of Replications," *Biometrics*, 12, 309–310.

Lange, K. (1999), *Numerical Analysis for Statisticians*, New York: Springer-Verlag.

Liang, K. Y. and Zeger, S. L. (1986), "Longitudinal Data Analysis Using Generalized Linear Models," *Biometrika*, 73, 13–22.

Lin, X. and Breslow, N. W. (1996), "Bias Correction in Generalized Linear Mixed Models with Multiple Components of Dispersion," *Journal of the American Statistical Association*, 91, 1007–1116.

Littell, R. C., Milliken, G. A., Stroup, W. W., Wolfinger, R. D., and Schabenberger, O. (2006), *SAS for Mixed Models*, Second Edition, Cary, NC: SAS Institute Inc.

Long, J. S., and Ervin, L. H. (2000), "Using Heteroscedasticity Consistent Standard Errors in the Linear Regression Model," *The American Statistician*, 217–224.

Macchiavelli, R. E. and Arnold, S. F. (1994), "Variable Order Ante-dependence Models," *Communications in Statistics–Theory and Methods*, 23(9), 2683–2699.

MacKinnon, J. G. and White, H. (1985), "Some Heteroskedasticity-Consistent Covariance Matrix Estimators with Improved Finite Sample Properties," *Journal of Econometrics*, 29, 305–325.

Mancl, L. A. and DeRouen, T. A. (2001), "A Covariance Estimator for GEE with Improved Small-Sample Properties," *Biometrics*, 57, 126–134.

Matérn, B. (1986), *Spatial Variation*, Second Edition, Lecture Notes in Statistics, New York: Springer-Verlag.

McCullagh, P. (1980), "Regression Models for Ordinal Data," *Journal of the Royal Statistical Society, Series B,* 42, 109–142.

McCullagh, P. and Nelder. J. A. (1989), *Generalized Linear Models*, Second Edition, London: Chapman & Hall.

McLean, R. A. and Sanders, W. L. (1988), "Approximating Degrees of Freedom for Standard Errors in Mixed Linear Models," *Proceedings of the Statistical Computing Section*, American Statistical Association, New Orleans, 50–59.

McLean, R. A., Sanders, W. L., and Stroup, W. W. (1991), "A Unified Approach to Mixed Linear Models," *The American Statistician*, 45, 54–64.

Milliken, G. A. and Johnson, D. E. (1992), *Analysis of Messy Data, Volume 1: Designed Experiments*, New York: Chapman & Hall.

Moré, J. J. (1978), "The Levenberg-Marquardt Algorithm: Implementation and Theory," in *Lecture Notes in Mathematics 630*, ed. G. A. Watson, Berlin-Heidelberg-New York: Springer-Verlag.

Moré, J. J. and Sorensen, D. C. (1983), "Computing a Trust-Region Step," *SIAM Journal on Scientific and Statistical Computing*, 4, 553–572.

Morel, J. G. (1989), "Logistic Regression under Complex Survey Designs," *Survey Methodology*, 15, 203–223.

Morel, J. G., Bokossa, M. C. and Neerchal, N. K. (2003), "Small Sample Correction for the Variance of GEE Estimators," *Biometrical Journal*, 4, 395–409.

Moriguchi, S., ed. (1976), *Statistical Method for Quality Control,* (in Japanese), Tokyo: Japan

Standards Association.

Mosteller, F. and Tukey, J. W. (1977), *Data Analysis and Regression*, Reading, MA: Addison-Wesley.

Murray, D. M., Varnell, S. P., and Blitstein, J. L. (2004), "Design and Analysis of Group-Randomized Trials: A Review of Recent Methodological Developments," *American Journal of Public Health*, 94, 423–432.

National Institute of Standards and Technology (1998), *Statistical Reference Data Sets*, http://www.nist.gov/itl/div898/strd: last accessed July 28, 2005.

Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized Linear Models," *Journal of the Royal Statistical Society A*, 135, 370–384.

Nelson, P. R. (1982), "Exact Critical Points for the Analysis of Means," *Communications in Statistics*, 11, 699–709.

Nelson, P. R. (1991), "Numerical Evaluation of Multivariate Normal Integrals with Correlations $\rho_{lj} = -\alpha_l \alpha_j$," *The Frontiers of Statistical Scientific Theory & Industrial Applications*, 97–114.

Nelson, P. R. (1993), "Additional Uses for the Analysis of Means and Extended Tables of Critical Values," *Technometrics*, 35, 61–71.

Ott, E. R. (1967), "Analysis of Means—A Graphical Procedure," *Industrial Quality Control*, 101–109. Reprinted in *Journal of Quality Technology*, 15 (1983), 10–18.

Patel, H. I. (1991), "Analysis of Incomplete Data from a Clinical Trial with Repeated Measurements," *Biometrika*, 78, 609–619.

Pawitan, Y. (2001), *In All Likelihood: Statistical Modelling and Inference Using Likelihood*, Oxford, UK: Oxford University Press.

Pinheiro, J. C. and Bates, D. M. (1995), "Approximations to the Log-Likelihood Function in the Nonlinear Mixed-Effects Model," *Journal of Computational and Graphical Statistics,* 4, 12–35.

Pinheiro, J. C. and Chao, E. C. (2006), "Efficient Laplacian and Adaptive Gaussian Quadrature Algorithms for Multilevel Generalized Linear Mixed Models," *Journal of Computational and Graphical Statistics*, 15, 58–81.

Polak, E. (1971), *Computational Methods in Optimization*, New York: Academic Press.

Pothoff, R. F. and Roy, S. N. (1964), "A Generalized Multivariate Analysis of Variance Model Useful Especially for Growth Curve Problems," *Biometrika*, 51, 313–326.

Powell, J. M. D. (1977), "Restart Procedures for the Conjugate Gradient Method," *Math. Prog.*, 12, 241–254.

Prasad, N. G. N. and Rao, J. N. K. (1990), "The Estimation of Mean Squared Error of Small-Area Estimators," *Journal of the American Statistical Association*, 85, 163–171.

Pringle, R. M. and Rayner, A. A. (1971), *Generalized Inverse Matrices with Applications to Statis-*

tics, New York: Hafner Publishing.

Raudenbush, S. M., Yang, M.-L., and Yosef, M. (2000), "Maximum Likelihood for Generalized Linear Models with Nested Random Effects via Higher-Order, Multivariate Laplace Approximation," *Journal of Computational and Graphical Statistics,* 9, 141–157.

Royen, T. (1989), "Generalized Maximum Range Tests for Pairwise Comparisons of Several Populations," *Biometrical Journal,* 31, 905–929.

Ruppert, D., Wand, M. P., and Carroll, R. J. (2003), *Semiparametric Regression*, Cambridge: Cambridge University Press.

Saxton, A., ed. (2004), *Genetic Analysis of Complex Traits Using SAS,* Cary, NC: SAS Institute Inc.

Schabenberger, O. and Gregoire, T. G. (1996), "Population-Averaged and Subject-Specific Approaches for Clustered Categorical Data," *Journal of Statistical Computation and Simulation,* 54, 231–253.

Schabenberger, O. Gregoire, T. G., and Kong, F. (2000), "Collections of Simple Effects and Their Relationship to Main Effects and Interactions in Factorials," *The American Statistician*, 54, 210–214.

Schabenberger, O. and Pierce, F. J. (2002), *Contemporary Statistical Models for the Plant and Soil Sciences,* Boca Raton, FL: CRC Press.

Schall, R. (1991), "Estimation in Generalized Linear Models with Random Effects," *Biometrika,* 78, 719–727.

Schluchter, M. D. and Elashoff, J. D. (1990), "Small-Sample Adjustments to Tests with Unbalanced Repeated Measures Assuming Several Covariance Structures," *Journal of Statistical Computation and Simulation*, 37, 69–87.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Searle, S. R. (1971), *Linear Models*, New York: John Wiley & Sons.

Self, S. G., and Liang, K.-Y. (1987), "Asymptotic Properties of Maximum Likelihood Estimators and Likelihood Ratio Tests under Nonstandard Conditions," *Journal of the American Statistical Association*, 398, 605–610.

Shaffer, J. P. (1986), "Modified Sequentially Rejective Multiple Test Procedures," *Journal of the American Statistical Association*, 81, 329 –335.

Shapiro, A. (1988), "Towards a Unified Theory of Inequality Constrained Testing in Multivariate Analysis," *International Statistical Review*, 56, 49–62.

Shun, Z. (1997), "Another Look at the Salamander Mating Data: A Modified Laplace Approximation Approach," *Journal of the American Statistical Association,* 92, 341–349.

Shun, Z. and McCullagh, P. (1995) "Laplace Approximation of High Dimensional Integrals," *Jour-*

nal of the Royal Statistical Society, Series B, 57, 749–760.

Silvapulle, M. J. and Silvapulle, P. (1995), "A Score Test against One-Sided Alternatives," *Journal of the American Statistical Association*, 429, 342–349.

Stenstrom, F. H. (1940), "The Growth of Snapdragons, Stocks, Cinerarias and Carnations on Six Iowa Soils," master's thesis, Iowa State College.

Stram, D. O. and Lee, J. W. (1994), "Variance Components Testing in the Longitudinal Mixed Effects Model," *Biometrics*, 50, 1171–1177.

Stram, D. O. and Lee, J. W. (1995), "Correction to 'Variance Components Testing in the Longitudinal Mixed Effects Model'," *Biometrics*, 51, 1196.

Tamhane, A. C. (1979), "A Comparison of Procedures for Multiple Comparisons of Means with Unequal Variances," *Journal of the American Statistical Association*, 74, 471–480.

Thall, P. F. and Vail, S. C. (1990), "Some Covariance Models for Longitudinal Count Data with Overdispersion," *Biometrics*, 46, 657–671.

Verbeke, G. and Molenberghs, G. (2000), *Linear Mixed Models for Longitudinal Data,* Berlin-Heidelberg-New York: Springer-Verlag.

Verbeke, G. and Molenberghs, G. (2003), "The Use of Score Tests for Inference on Variance Components," *Biometrics*, 59, 254–262.

Vonesh, E. F. (1996) "A Note on the Use of Laplace's Approximation for Nonlinear Mixed-Effects Models," *Biometrika,* 83, 447–452.

Vonesh, E. F. and Chinchilli, V. M. (1997), *Linear and Nonlinear Models for the Analysis of Repeated Measurements*, New York: Marcel Dekker.

Vonesh, E. F., Chinchilli, V. M., and Pu, K. (1996), "Goodness-of-Fit in Generalized Nonlinear Mixed-Effects Models," *Biometrics*, 52, 572–587.

Wedderburn, R. W. M. (1974), "Quasilikelihood Functions, Generalized Linear Models and the Gauss-Newton Method," *Biometrika*, 61, 439–447.

Westfall, P. H. (1997), "Multiple Testing of General Contrasts Using Logical Constraints and Correlations," *Journal of the American Statistical Association*, 92, 299–306.

Westfall, P. H. and Tobias, R. D. (2007), "Multiple Testing of General Contrasts: Truncated Closure and the Extended Shaffer-Royen Method," *Journal of the American Statistical Association*, 478, 487–494.

Westfall, P. H., Tobias, R. D., Rom, D., Wolfinger, R. D., and Hochberg, Y. (1999), *Multiple Comparisons and Multiple Tests Using the SAS System*, Cary, NC: SAS Institute Inc.

Westfall, P. J. and Young, S. S. (1993), *Resampling-Based Multiple Testing*, New York: John Wiley & Sons.

White, H. (1980), "A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test

for Heteroskedasticity," *Econometrica*, 48, 817–838.

White, H. (1982), "Maximum Likelihood Estimation of Misspecified Models," *Econometrica,* 50, 1–25.

Whittle, P. (1954), "On Stationary Processes in the Plane," *Biometrika*, 41, 434–449.

Winer, B. J. (1971), *Statistical Principles in Experimental Design*, Second Edition, New York: McGraw-Hill.

Wolfinger, R. (1993), "Laplace's Approximation for Nonlinear Mixed Models," *Biometrika*, 80, 791–795.

Wolfinger, R. and O'Connell, M. (1993), "Generalized Linear Mixed Models: A Pseudo-likelihood Approach," *Journal of Statistical Computation and Simulation*, 4, 233–243.

Wolfinger, R., Tobias, R., and Sall, J. (1994) "Computing Gaussian Likelihoods and Their Derivatives for General Linear Mixed Models," *SIAM Journal on Scientific Computing*, 15(6), 1294–1310.

Zeger, S. L. and Liang, K.-Y. (1986), "Longitudinal Data Analysis for Discrete and Continuous Outcomes," *Biometrics*, 42, 121–130.

Chapter 39

The GLM Procedure

Contents

Overview: GLM Procedure **2432**
PROC GLM Features 2433
PROC GLM Contrasted with Other SAS Procedures 2434
Getting Started: GLM Procedure **2435**
PROC GLM for Unbalanced ANOVA 2435
PROC GLM for Quadratic Least Squares Regression 2438
Syntax: GLM Procedure **2444**
PROC GLM Statement 2446
ABSORB Statement 2452
BY Statement 2452
CLASS Statement 2453
CONTRAST Statement 2454
ESTIMATE Statement 2456
FREQ Statement 2457
ID Statement 2458
LSMEANS Statement 2458
MANOVA Statement 2464
MEANS Statement 2468
MODEL Statement 2474
OUTPUT Statement 2477
RANDOM Statement 2480
REPEATED Statement 2481
TEST Statement 2485
WEIGHT Statement 2486
Details: GLM Procedure **2487**
Statistical Assumptions for Using PROC GLM 2487
Specification of Effects 2488
Using PROC GLM Interactively 2490
Parameterization of PROC GLM Models 2491
Hypothesis Testing in PROC GLM 2496
Effect Size Measures for *F* Tests in GLM (Experimental) 2502
Absorption 2507
Specification of ESTIMATE Expressions 2509
Comparing Groups 2511

Means versus LS-Means 2511
Multiple Comparisons 2514
Simple Effects 2525
Homogeneity of Variance in One-Way Models 2526
Weighted Means 2527
Construction of Least Squares Means 2528
Multivariate Analysis of Variance 2531
Repeated Measures Analysis of Variance 2532
Random-Effects Analysis 2540
Missing Values 2543
Computational Resources 2544
Computational Method 2547
Output Data Sets 2548
Displayed Output 2550
ODS Table Names 2551
ODS Graphics 2554
Examples: GLM Procedure 2556
Example 39.1: Randomized Complete Blocks with Means Comparisons and Contrasts 2556
Example 39.2: Regression with Mileage Data 2562
Example 39.3: Unbalanced ANOVA for Two-Way Design with Interaction . 2565
Example 39.4: Analysis of Covariance 2571
Example 39.5: Three-Way Analysis of Variance with Contrasts 2578
Example 39.6: Multivariate Analysis of Variance 2582
Example 39.7: Repeated Measures Analysis of Variance 2590
Example 39.8: Mixed Model Analysis of Variance with the RANDOM Statement 2595
Example 39.9: Analyzing a Doubly Multivariate Repeated Measures Design 2598
Example 39.10: Testing for Equal Group Variances 2604
Example 39.11: Analysis of a Screening Design 2608
References 2613

Overview: GLM Procedure

The GLM procedure uses the method of least squares to fit general linear models. Among the statistical methods available in PROC GLM are regression, analysis of variance, analysis of covariance, multivariate analysis of variance, and partial correlation.

PROC GLM analyzes data within the framework of general linear models. PROC GLM handles models relating one or several continuous dependent variables to one or several independent variables. The independent variables can be either *classification* variables, which divide the observations into discrete groups, or *continuous* variables. Thus, the GLM procedure can be used for many different analyses, including the following:

- simple regression
- multiple regression
- analysis of variance (ANOVA), especially for unbalanced data
- analysis of covariance
- response surface models
- weighted regression
- polynomial regression
- partial correlation
- multivariate analysis of variance (MANOVA)
- repeated measures analysis of variance

PROC GLM Features

The following list summarizes the features in PROC GLM:

- PROC GLM enables you to specify any degree of interaction (crossed effects) and nested effects. It also provides for polynomial, continuous-by-class, and continuous-nesting-class effects.
- Through the concept of estimability, the GLM procedure can provide tests of hypotheses for the effects of a linear model regardless of the number of missing cells or the extent of confounding. PROC GLM displays the sum of squares (SS) associated with each hypothesis tested and, upon request, the form of the estimable functions employed in the test. PROC GLM can produce the general form of all estimable functions.
- The REPEATED statement enables you to specify effects in the model that represent repeated measurements on the same experimental unit for the same response, providing both univariate and multivariate tests of hypotheses.
- The RANDOM statement enables you to specify random effects in the model; expected mean squares are produced for each Type I, Type II, Type III, Type IV, and contrast mean square used in the analysis. Upon request, F tests that use appropriate mean squares or linear combinations of mean squares as error terms are performed.
- The ESTIMATE statement enables you to specify an $\mathbf{L}$ vector for estimating a linear function of the parameters $\mathbf{L}\boldsymbol{\beta}$.
- The CONTRAST statement enables you to specify a contrast vector or matrix for testing the hypothesis that $\mathbf{L}\boldsymbol{\beta} = 0$. When specified, the contrasts are also incorporated into analyses that use the MANOVA and REPEATED statements.

- The MANOVA statement enables you to specify both the hypothesis effects and the error effect to use for a multivariate analysis of variance.
- PROC GLM can create an output data set containing the input data set in addition to predicted values, residuals, and other diagnostic measures.
- PROC GLM can be used interactively. After you specify and fit a model, you can execute a variety of statements without recomputing the model parameters or sums of squares.
- For analysis involving multiple dependent variables but not the MANOVA or REPEATED statements, a missing value in one dependent variable does not eliminate the observation from the analysis for other dependent variables. PROC GLM automatically groups together those variables that have the same pattern of missing values within the data set or within a BY group. This ensures that the analysis for each dependent variable brings into use all possible observations.
- The GLM procedure automatically produces graphics as part of its ODS output. For general information about ODS Graphics, see the section "ODS Graphics" on page 2554 and Chapter 21, "Statistical Graphics Using ODS."

PROC GLM Contrasted with Other SAS Procedures

As described previously, PROC GLM can be used for many different analyses and has many special features not available in other SAS procedures. However, for some types of analyses, other procedures are available. As discussed in the sections "PROC GLM for Unbalanced ANOVA" on page 2435 and "PROC GLM for Quadratic Least Squares Regression" on page 2438, sometimes these other procedures are more efficient than PROC GLM. The following procedures perform some of the same analyses as PROC GLM:

ANOVA	performs analysis of variance for balanced designs. The ANOVA procedure is generally more efficient than PROC GLM for these designs.
MIXED	fits mixed linear models by incorporating covariance structures in the model fitting process. Its RANDOM and REPEATED statements are similar to those in PROC GLM but offer different functionalities.
NESTED	performs analysis of variance and estimates variance components for nested random models. The NESTED procedure is generally more efficient than PROC GLM for these models.
NPAR1WAY	performs nonparametric one-way analysis of rank scores. This can also be done using the RANK procedure and PROC GLM.
REG	performs simple linear regression. The REG procedure allows several MODEL statements and gives additional regression diagnostics, especially for detection of collinearity.
RSREG	performs quadratic response surface regression, and canonical and ridge analysis. The RSREG procedure is generally recommended for data from a response surface experiment.

TTEST	compares the means of two groups of observations. Also, tests for equality of variances for the two groups are available. The TTEST procedure is usually more efficient than PROC GLM for this type of data.
VARCOMP	estimates variance components for a general linear model.

Getting Started: GLM Procedure

PROC GLM for Unbalanced ANOVA

Analysis of variance, or ANOVA, typically refers to partitioning the variation in a variable's values into variation between and within several groups or classes of observations. The GLM procedure can perform simple or complicated ANOVA for balanced or unbalanced data.

This example discusses the analysis of variance for the unbalanced 2×2 data shown in Table 39.1. The experimental design is a full factorial, in which each level of one treatment factor occurs at each level of the other treatment factor. Note that there is only one value for the cell with A='A2' and B='B2'. Since one cell contains a different number of values from the other cells in the table, this is an unbalanced design.

Table 39.1 Unbalanced Two-Way Data

	A1	**A2**
B1	12, 14	20, 18
B2	11, 9	17

The following statements read the data into a SAS data set and then invoke PROC GLM to produce the analysis.

```
title 'Analysis of Unbalanced 2-by-2 Factorial';
data exp;
   input A $ B $ Y @@;
   datalines;
A1 B1 12 A1 B1 14     A1 B2 11 A1 B2 9
A2 B1 20 A2 B1 18     A2 B2 17
;

proc glm data=exp;
   class A B;
   model Y=A B A*B;
run;
```

Both treatments are listed in the CLASS statement because they are classification variables. A*B denotes the interaction of the A effect and the B effect. The results are shown in Figure 39.1 and Figure 39.2.

Figure 39.1 Class Level Information

Analysis of Unbalanced 2-by-2 Factorial

The GLM Procedure

Class Level Information

Class	Levels	Values
A	2	A1 A2
B	2	B1 B2

Number of Observations Read	7
Number of Observations Used	7

Figure 39.1 displays information about the classes as well as the number of observations in the data set. Figure 39.2 shows the ANOVA table, simple statistics, and tests of effects.

Figure 39.2 ANOVA Table and Tests of Effects

Analysis of Unbalanced 2-by-2 Factorial

The GLM Procedure

Dependent Variable: Y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	91.71428571	30.57142857	15.29	0.0253
Error	3	6.00000000	2.00000000		
Corrected Total	6	97.71428571			

R-Square	Coeff Var	Root MSE	Y Mean
0.938596	9.801480	1.414214	14.42857

Source	DF	Type I SS	Mean Square	F Value	Pr > F
A	1	80.04761905	80.04761905	40.02	0.0080
B	1	11.26666667	11.26666667	5.63	0.0982
A*B	1	0.40000000	0.40000000	0.20	0.6850

Source	DF	Type III SS	Mean Square	F Value	Pr > F
A	1	67.60000000	67.60000000	33.80	0.0101
B	1	10.00000000	10.00000000	5.00	0.1114
A*B	1	0.40000000	0.40000000	0.20	0.6850

The degrees of freedom can be used to check your data. The Model degrees of freedom for a 2×2 factorial design with interaction are $(ab - 1)$, where a is the number of levels of A and b is the number of levels of B; in this case, $(2 \times 2 - 1) = 3$. The Corrected Total degrees of freedom are always one less than the number of observations used in the analysis; in this case, $7 - 1 = 6$.

The overall F test is significant ($F = 15.29$, $p = 0.0253$), indicating strong evidence that the means for the four different A×B cells are different. You can further analyze this difference by examining the individual tests for each effect.

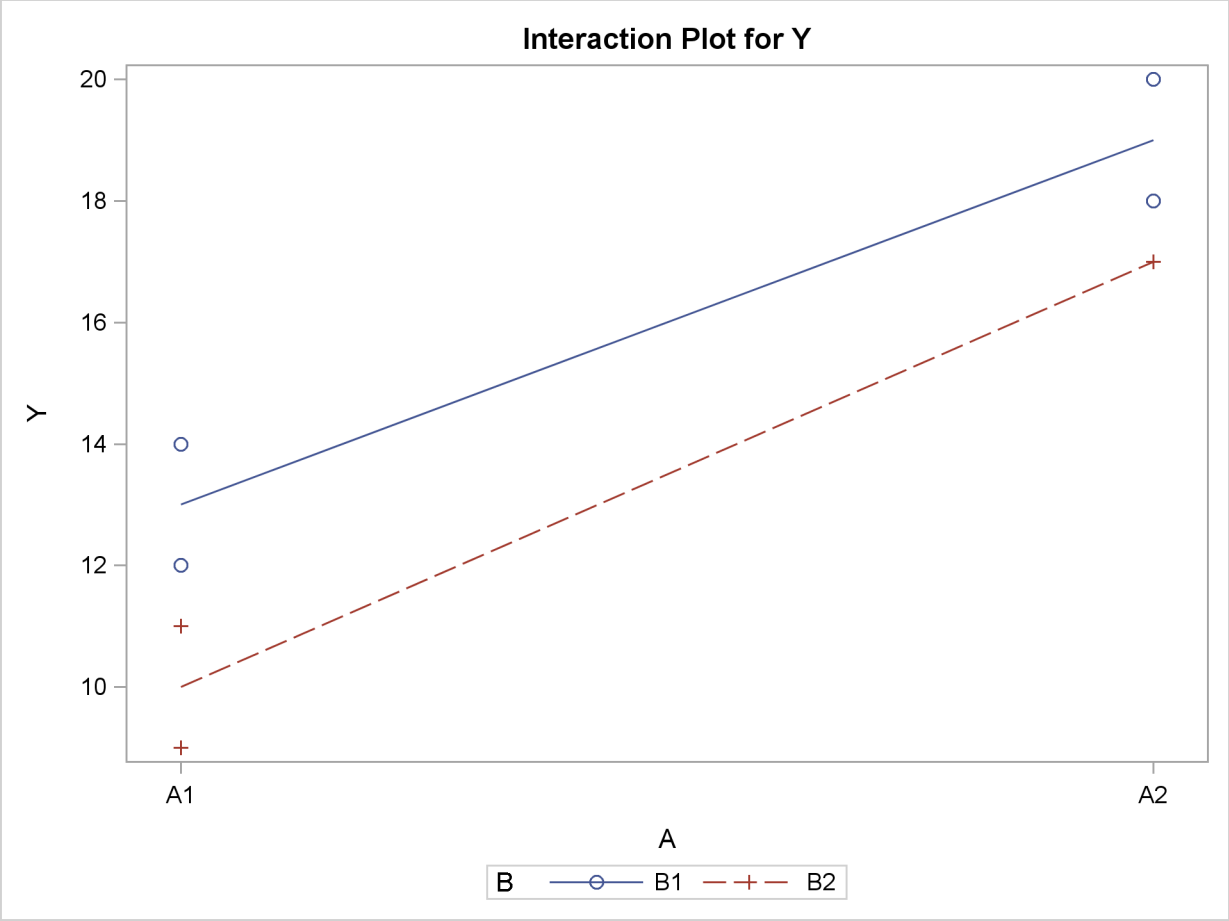
Four types of estimable functions of parameters are available for testing hypotheses in PROC GLM. For data with no missing cells, the Type III and Type IV estimable functions are the same and test the same hypotheses that would be tested if the data were balanced. Type I and Type III sums of squares are typically not equal when the data are unbalanced; Type III sums of squares are preferred in testing effects in unbalanced cases because they test a function of the underlying parameters that is independent of the number of observations per treatment combination.

According to a significance level of 5% ($\alpha = 0.05$), the A*B interaction is not significant ($F = 0.20$, $p = 0.6850$). This indicates that the effect of A does not depend on the level of B and vice versa. Therefore, the tests for the individual effects are valid, showing a significant A effect ($F = 33.80$, $p = 0.0101$) but no significant B effect ($F = 5.00$, $p = 0.1114$).

If you enable ODS Graphics, GLM also displays by default an interaction plot for this analysis. The following statements, which are the same as in the previous analysis but with ODS Graphics enabled, additionally produce Figure 39.3.

```
ods graphics on;
proc glm data=exp;
   class A B;
   model Y=A B A*B;
run;
ods graphics off;
```

Figure 39.3 Plot of Y by A and B



The insignificance of the A*B interaction is reflected in the fact that two lines in Figure 39.3 are nearly parallel. For more information about the graphics that GLM can produce, see the section "ODS Graphics" on page 2554.

PROC GLM for Quadratic Least Squares Regression

In polynomial regression, the values of a dependent variable (also called a response variable) are described or predicted in terms of polynomial terms involving one or more independent or explanatory variables. An example of quadratic regression in PROC GLM follows. These data are taken from Draper and Smith (1966, p. 57). Thirteen specimens of 90/10 Cu-Ni alloys are tested in a corrosion-wheel setup in order to examine corrosion. Each specimen has a certain iron content. The wheel is rotated in salt sea water at 30 ft/sec for 60 days. Weight loss is used to quantify the corrosion. The fe variable represents the iron content, and the loss variable denotes the weight loss in milligrams/square decimeter/day in the following DATA step.

```
title 'Regression in PROC GLM';
data iron;
```

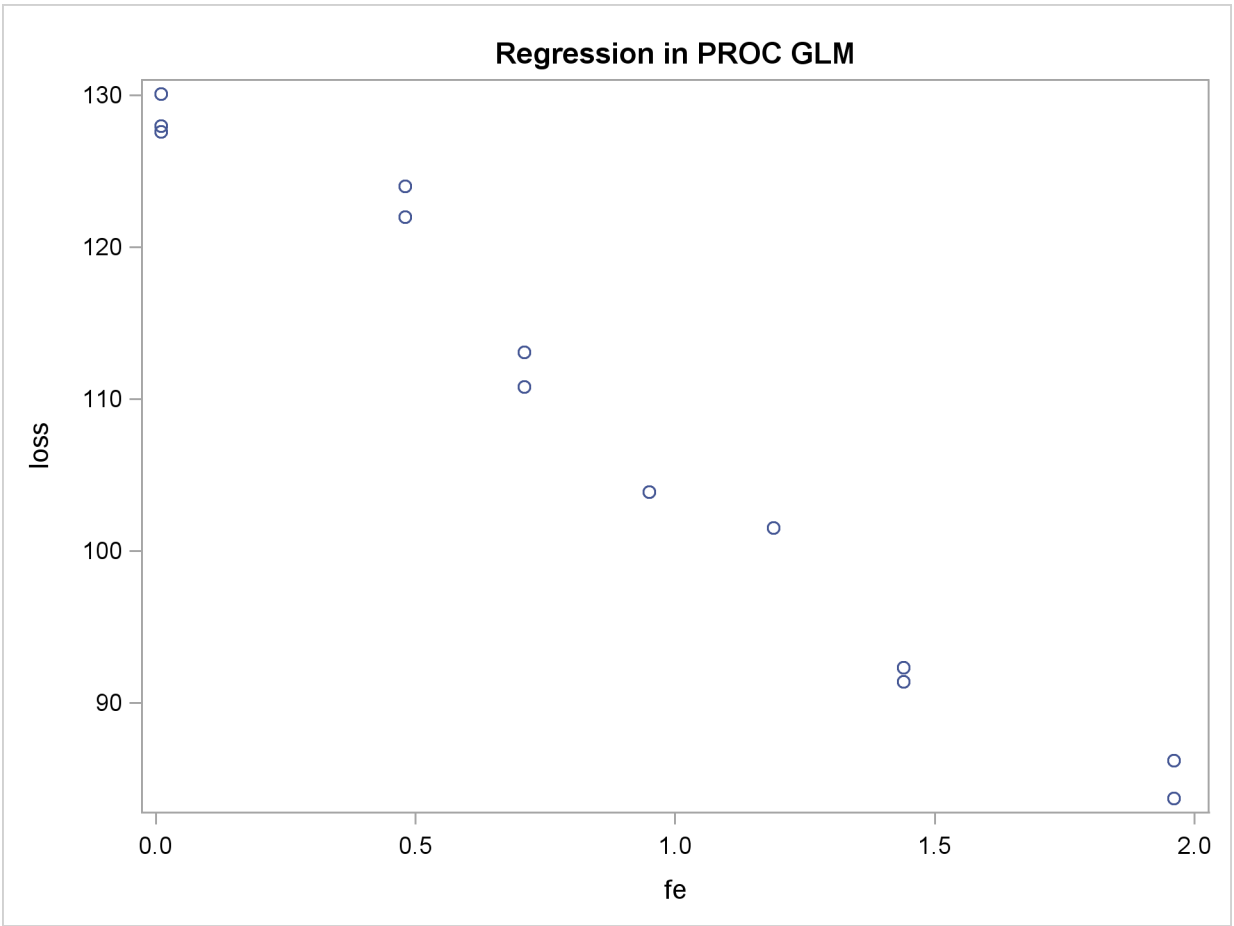
```
   input fe loss @@;
   datalines;
0.01 127.6   0.48 124.0   0.71 110.8   0.95 103.9
1.19 101.5   0.01 130.1   0.48 122.0   1.44  92.3
0.71 113.1   1.96  83.7   0.01 128.0   1.44  91.4
1.96  86.2
;
```

The SGSCATTER procedure is used in the following statements to request a scatter plot of the response variable versus the independent variable.

```
ods graphics on;
proc sgscatter data=iron;
   plot loss*fe;
run;
ods graphics off;
```

The plot in Figure 39.4 displays a strong negative relationship between iron content and corrosion resistance, but it is not clear whether there is curvature in this relationship.

Figure 39.4 Plot of Observed Corrosion Resistance by Iron Content



The following statements fit a quadratic regression model to the data. This enables you to estimate the linear relationship between iron content and corrosion resistance and to test for the presence of a quadratic component. The intercept is automatically fit unless the NOINT option is specified.

```
proc glm data=iron;
   model loss=fe fe*fe;
run;
```

The CLASS statement is omitted because a regression line is being fitted. Unlike PROC REG, PROC GLM allows polynomial terms in the MODEL statement.

PROC GLM first displays preliminary information, shown in Figure 39.5, telling you that the GLM procedure has been invoked and stating the number of observations in the data set. If the model involves classification variables, they are also listed here, along with their levels.

Figure 39.5 Data Information

```
                         Regression in PROC GLM

                           The GLM Procedure

                 Number of Observations Read          13
                 Number of Observations Used          13
```

Figure 39.6 shows the overall ANOVA table and some simple statistics. The degrees of freedom can be used to check that the model is correct and that the data have been read correctly. The Model degrees of freedom for a regression is the number of parameters in the model minus 1. You are fitting a model with three parameters in this case,

$$\text{loss} \quad = \quad \beta_0 + \beta_1 \times (\text{fe}) + \beta_2 \times (\text{fe})^2 + error$$

so the degrees of freedom are $3 - 1 = 2$. The Corrected Total degrees of freedom are always one less than the number of observations used in the analysis.

Figure 39.6 ANOVA Table

```
                         Regression in PROC GLM

                           The GLM Procedure

Dependent Variable: loss

                                      Sum of
 Source                    DF        Squares    Mean Square   F Value   Pr > F

 Model                      2    3296.530589    1648.265295    164.68   <.0001

 Error                     10     100.086334      10.008633

 Corrected Total           12    3396.616923
```

Figure 39.6 *continued*

R-Square	Coeff Var	Root MSE	loss Mean
0.970534	2.907348	3.163642	108.8154

The R^2 indicates that the model accounts for 97% of the variation in LOSS. The coefficient of variation (Coeff Var), Root MSE (Mean Square for Error), and mean of the dependent variable are also listed.

The overall F test is significant ($F = 164.68$, $p < 0.0001$), indicating that the model as a whole accounts for a significant amount of the variation in LOSS. Thus, it is appropriate to proceed to testing the effects.

Figure 39.7 contains tests of effects and parameter estimates. The latter are displayed by default when the model contains only continuous variables.

Figure 39.7 Tests of Effects and Parameter Estimates

Source	DF	Type I SS	Mean Square	F Value	Pr > F
fe	1	3293.766690	3293.766690	329.09	<.0001
fe*fe	1	2.763899	2.763899	0.28	0.6107

Source	DF	Type III SS	Mean Square	F Value	Pr > F
fe	1	356.7572421	356.7572421	35.64	0.0001
fe*fe	1	2.7638994	2.7638994	0.28	0.6107

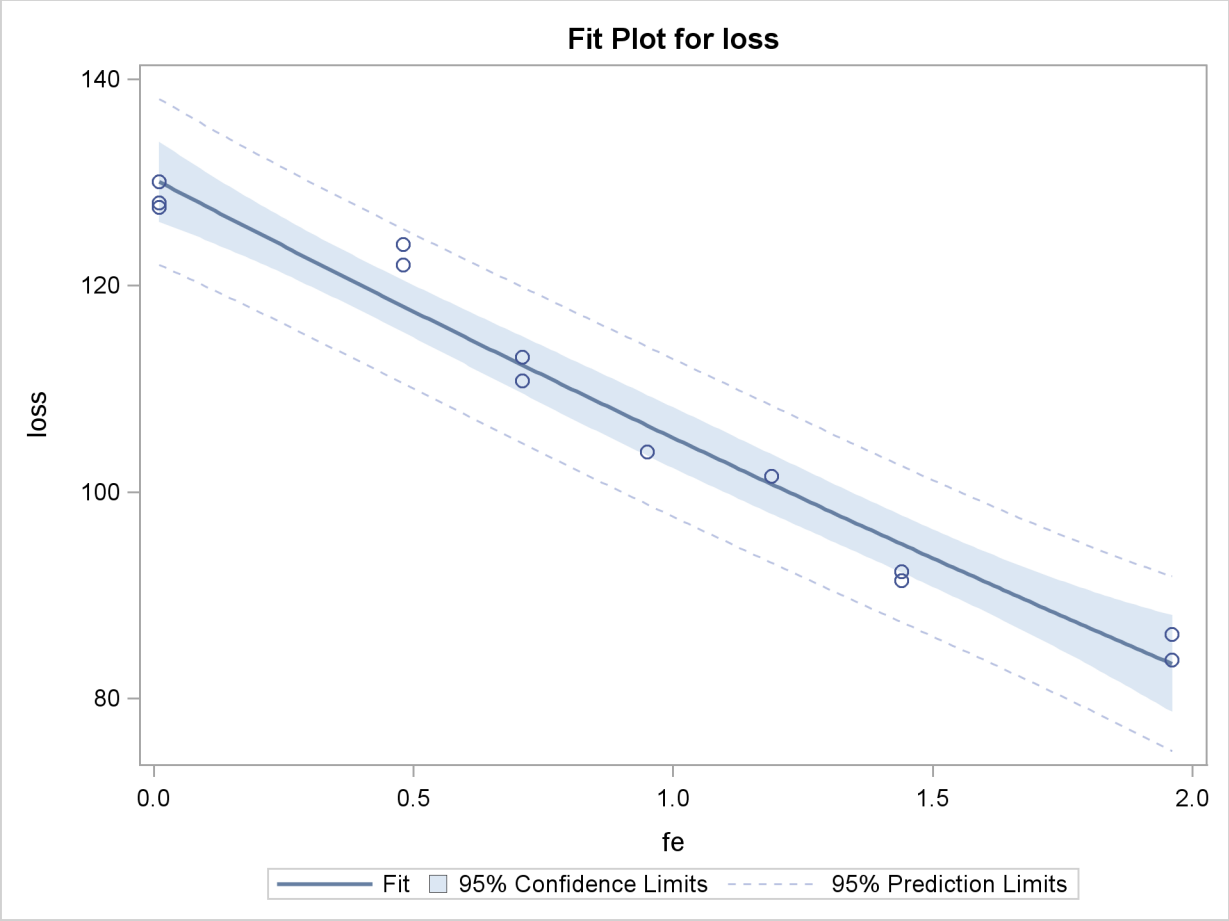
Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	130.3199337	1.77096213	73.59	<.0001
fe	-26.2203900	4.39177557	-5.97	0.0001
fe*fe	1.1552018	2.19828568	0.53	0.6107

The t tests provided are equivalent to the Type III F tests. The quadratic term is not significant ($p = 0.6107$) and thus can be removed from the model; the linear term is significant ($p < 0.0001$). This suggests that there is indeed a straight-line relationship between loss and fe.

Finally, if you enable ODS Graphics, PROC GLM also displays by default a scatter plot of the original data, as in Figure 39.4, with the quadratic fit overlaid. The following statements, which are the same as the previous analysis but with ODS Graphics enabled, additionally produce Figure 39.8.

```
ods graphics on;
proc glm data=iron;
   model loss=fe fe*fe;
run;
ods graphics off;
```

Figure 39.8 Plot of Observed and Fit Corrosion Resistance by Iron Content, Quadratic Model



The insignificance of the quadratic term in the model is reflected in the fact that the fit is nearly linear.

Fitting the model without the quadratic term provides more accurate estimates for β_0 and β_1. PROC GLM allows only one MODEL statement per invocation of the procedure, so the PROC GLM statement must be issued again. The following statements are used to fit the linear model.

```
proc glm data=iron;
   model loss=fe;
run;
```

Figure 39.9 displays the output produced by these statements. The linear term is still significant ($F = 352.27$, $p < 0.0001$). The estimated model is now

$$\text{loss} = 129.79 - 24.02 \times \text{fe}$$

Figure 39.9 Linear Model Output

```
                          Regression in PROC GLM

                            The GLM Procedure

Dependent Variable: loss

                                    Sum of
Source                      DF       Squares    Mean Square   F Value   Pr > F

Model                        1    3293.766690    3293.766690    352.27   <.0001

Error                       11     102.850233       9.350021

Corrected Total             12    3396.616923

              R-Square     Coeff Var      Root MSE     loss Mean

              0.969720      2.810063      3.057780      108.8154

Source                      DF      Type I SS    Mean Square   F Value   Pr > F

fe                           1    3293.766690    3293.766690    352.27   <.0001

Source                      DF    Type III SS    Mean Square   F Value   Pr > F

fe                           1    3293.766690    3293.766690    352.27   <.0001

                                          Standard
      Parameter         Estimate             Error    t Value    Pr > |t|

      Intercept      129.7865993      1.40273671      92.52      <.0001
      fe             -24.0198934      1.27976715     -18.77      <.0001
```

Syntax: GLM Procedure

The following statements are available in PROC GLM:

PROC GLM < *options* > ;
CLASS *variables* < / *option* > ;
MODEL *dependents=independents* < / *options* > ;
ABSORB *variables* ;
BY *variables* ;
FREQ *variable* ;
ID *variables* ;
WEIGHT *variable* ;
CONTRAST *'label' effect values* < . . . *effect values* > < / *options* > ;
ESTIMATE *'label' effect values* < . . . *effect values* > < / *options* > ;
LSMEANS *effects* < / *options* > ;
MANOVA < *test-options* >< / *detail-options* > ;
MEANS *effects* < / *options* > ;
OUTPUT < **OUT=***SAS-data-set* > *keyword=names* < . . . *keyword=names* > < / *option* > ;
RANDOM *effects* < / *options* > ;
REPEATED *factor-specification* < / *options* > ;
TEST < **H=***effects* > **E=***effect* < / *options* > ;

Although there are numerous statements and options available in PROC GLM, many applications use only a few of them. Often you can find the features you need by looking at an example or by quickly scanning through this section.

To use PROC GLM, the PROC GLM and MODEL statements are required. You can specify only one MODEL statement (in contrast to the REG procedure, for example, which allows several MODEL statements in the same PROC REG run). If your model contains classification effects, the classification variables must be listed in a CLASS statement, and the CLASS statement must appear before the MODEL statement. In addition, if you use a CONTRAST statement in combination with a MANOVA, RANDOM, REPEATED, or TEST statement, the CONTRAST statement must be entered first in order for the contrast to be included in the MANOVA, RANDOM, REPEATED, or TEST analysis.

Table 39.2 summarizes the positional requirements for the statements in the GLM procedure.

Table 39.2 Positional Requirements for PROC GLM Statements

Statement	Must Precede…	Must Follow…
ABSORB	first RUN statement	
BY	first RUN statement	
CLASS	MODEL statement	
CONTRAST	MANOVA, REPEATED, or RANDOM statement	MODEL statement
ESTIMATE		MODEL statement
FREQ	first RUN statement	
ID	first RUN statement	
LSMEANS		MODEL statement
MANOVA		CONTRAST or MODEL statement
MEANS		MODEL statement
MODEL	CONTRAST, ESTIMATE, LSMEANS, or MEANS statement	CLASS statement
OUTPUT		MODEL statement
RANDOM		CONTRAST or MODEL statement
REPEATED		CONTRAST, MODEL, or TEST statement
TEST	MANOVA or REPEATED statement	MODEL statement
WEIGHT	first RUN statement	

Table 39.3 summarizes the function of each statement (other than the PROC statement) in the GLM procedure.

Table 39.3 Statements in the GLM Procedure

Statement	Description
ABSORB	absorbs classification effects in a model
BY	specifies variables to define subgroups for the analysis
CLASS	declares classification variables
CONTRAST	constructs and tests linear functions of the parameters
ESTIMATE	estimates linear functions of the parameters
FREQ	specifies a frequency variable
ID	identifies observations on output
LSMEANS	computes least squares (marginal) means
MANOVA	performs a multivariate analysis of variance
MEANS	computes and optionally compares arithmetic means
MODEL	defines the model to be fit

Table 39.3 *continued*

Statement	Description
OUTPUT	requests an output data set containing diagnostics for each observation
RANDOM	declares certain effects to be random and computes expected mean squares
REPEATED	performs multivariate and univariate repeated measures analysis of variance
TEST	constructs tests that use the sums of squares for effects and the error term you specify
WEIGHT	specifies a variable for weighting observations

The rest of this section gives detailed syntax information for each of these statements, beginning with the PROC GLM statement. The remaining statements are covered in alphabetical order.

PROC GLM Statement

PROC GLM < *options* > **;**

The PROC GLM statement starts the GLM procedure. You can specify the following options in the PROC GLM statement.

ALPHA=*p*

specifies the level of significance p for $100(1-p)\%$ confidence intervals. The value must be between 0 and 1; the default value of $p = 0.05$ results in 95% intervals. This value is used as the default confidence level for limits computed by the following options.

Statement	Options
LSMEANS	CL
MEANS	CLM CLDIFF
MODEL	CLI CLM CLPARM
OUTPUT	UCL= LCL= UCLM= LCLM=

You can override the default in each of these cases by specifying the ALPHA= option for each statement individually.

DATA=*SAS-data-set*

names the SAS data set used by the GLM procedure. By default, PROC GLM uses the most recently created SAS data set.

MANOVA

requests the multivariate mode of eliminating observations with missing values. If any of the dependent variables have missing values, the procedure eliminates that observation from the analysis. The MANOVA option is useful if you use PROC GLM in interactive mode and plan to perform a multivariate analysis.

MULTIPASS

requests that PROC GLM reread the input data set when necessary, instead of writing the necessary values of dependent variables to a utility file. This option decreases disk space usage at the expense of increased execution times, and is useful only in rare situations where disk space is at an absolute premium.

NAMELEN=*n*

specifies the length of effect names in tables and output data sets to be *n* characters long, where *n* is a value between 20 and 200 characters. The default length is 20 characters.

NOPRINT

suppresses the normal display of results. The NOPRINT option is useful when you want only to create one or more output data sets with the procedure. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of all classification variables (specified in the CLASS statement). This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option may be useful when you use CONTRAST or ESTIMATE statements. Note that the ORDER= option applies to the levels for all classification variables. The exception is the default ORDER=FORMATTED for numeric variables for which you have supplied no explicit format. In this case, the levels are ordered by their internal value. Note that this represents a change from previous releases for how class levels are ordered. Before SAS 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values, and in order to revert to the previous ordering you can specify this format explicitly for the affected classification variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and usually required the user to intervene with an explicit format or ORDER=INTERNAL to get the more natural ordering. The following table shows how PROC GLM interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent.

For more information about sorting order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

OUTSTAT=*SAS-data-set*

names an output data set that contains sums of squares, degrees of freedom, F statistics, and probability levels for each effect in the model, as well as for each CONTRAST that uses the overall residual or error mean square (MSE) as the denominator in constructing the F statistic. If you use the CANONICAL option in the MANOVA statement and do not use an M= specification in the MANOVA statement, the data set also contains results of the canonical analysis.

See the section "Output Data Sets" on page 2548 for more information.

PLOTS <(*global-plot-options***)> <=** *plot-request* **<(***options***)>>**

PLOTS <(*global-plot-options***)> <= (***plot-request* **<(***options***)> <...** *plot-request* **<(***options***)> >) >**

controls the plots produced through ODS Graphics. When you specify only one plot request, you can omit the parentheses from around the plot request. For example:

```
PLOTS=NONE
PLOTS=(DIAGNOSTICS RESIDUALS)
PLOTS(UNPACK)=RESIDUALS
PLOT=MEANPLOT(CLBAND)
```

You must enable ODS Graphics before requesting plots, as in the following statements.

```
ods graphics on;
proc glm data=iron;
   model loss=fe fe*fe;
run;
ods graphics off;
```

For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." If you have enabled ODS Graphics but do not specify the PLOTS= option, then PROC GLM produces a default set of plots, which might be different for different models, as discussed in the following.

- If you specify a one-way analysis of variance model, with just one CLASS variable, the GLM procedure will produce a grouped box plot of the response values versus the CLASS levels. For an example of the box plot, see the section "One-Way Layout with Means Comparisons" on page 693.
- If you specify a two-way analysis of variance model, with just two CLASS variables, the GLM procedure will produce an interaction plot of the response values, with horizontal position representing one CLASS variable and marker style representing the other; and with predicted response values connected by lines representing the two-way analysis. For an example of the interaction plot, see the section "PROC GLM for Unbalanced ANOVA" on page 2435.
- If you specify a model with a single continuous predictor, the GLM procedure will produce a fit plot of the response values versus the covariate values, with a curve representing the fitted relationship and a band representing the confidence limits for individual mean values. For an example of the fit plot, see the section "PROC GLM for Quadratic Least Squares Regression" on page 2438.
- If you specify a model with two continuous predictors and no CLASS variables, the GLM procedure will produce a contour fit plot, overlaying a scatter plot of the data and a contour plot of the predicted surface.

- If you specify an analysis of covariance model, with one or two CLASS variables and one continuous variable, the GLM procedure will produce an analysis of covariance plot of the response values versus the covariate values, with lines representing the fitted relationship within each classification level. For an example of the analysis of covariance plot, see Example 39.4.
- If you specify an LSMEANS statement with the PDIFF option, the GLM procedure will produce a plot appropriate for the type of LS-means comparison. For PDIFF=ALL (which is the default if you specify only PDIFF), the procedure produces a diffogram, which displays all pairwise LS-means differences and their significance. The display is also known as a "mean-mean scatter plot" (Hsu 1996). For PDIFF=CONTROL, the procedure produces a display of each noncontrol LS-mean compared to the control LS-mean, with two-sided confidence intervals for the comparison. For PDIFF=CONTROLL and PDIFF=CONTROLU a similar display is produced, but with one-sided confidence intervals. Finally, for the PDIFF=ANOM option, the procedure produces an "analysis of means" plot, comparing each LS-mean to the average LS-mean.
- If you specify a MEANS statement, the GLM procedure will produce a grouped box plot of the response values versus the effect for which means are being calculated.

The global plot options include the following:

ONLY
suppresses the default plots. Only plots specifically requested are displayed.

UNPACKPANEL
UNPACK
suppresses paneling. By default, multiple plots can appear in some output panels. Specify UNPACKPANEL to get each plot in a separate panel. You can specify PLOTS(UNPACKPANEL) to just unpack the default plots. You can also specify UNPACKPANEL as a suboption with DIAGNOSTICS and RESIDUALS.

The following individual plots and plot options are available. If you specify only one *plot*, then you can omit the parentheses.

ALL
produces all appropriate plots. You can specify other options with ALL; for example, to request all plots and unpack just the residuals, specify: PLOTS=(ALL RESIDUALS(UNPACK)).

ANCOVAPLOT< (CLM CLI LIMITS) >
modifies the analysis of covariance plot produced by default when you have an analysis of covariance model, with one or two CLASS variables and one continuous variable. By default the plot does not show confidence limits around the predicted values. The PLOTS=ANCOVAPLOT(CLM) option adds limits for the expected predicted values, and PLOTS=ANCOVAPLOT(CLI) adds limits for new predictions. Use PLOTS=ANCOVAPLOT(LIMITS) to add both kinds of limits.

ANOMPLOT

requests an analysis of means display, in which least squares means are compared against an average least squares mean (Ott 1967; Nelson 1982, 1991, 1993). LS-mean ANOM plots are produced only if you also specify PDIFF=ANOM or ADJUST=NELSON in the LSMEANS statement, and in this case they are produced by default.

BOXPLOT< (NPANELPOS=*n*) >

modifies the plot produced by default for the model effect in a one-way analysis of variance model, or for an effect specified in the MEANS statement. Suppose the effect has m levels. By default, or if you specify PLOTS=BOXPLOT(NPANELPOS=0), all m levels of the effect are displayed in a single plot. Specifying a nonzero value of n will result in P panels, where P is the integer part of $m/n + 1$. If $n > 0$, then the levels will be approximately balanced across the P panels; whereas if $n < 0$, precisely $|n|$ levels will be displayed on each panel except possibly the last.

CONTOURFIT< (OBS=*obs-options*) >

modifies the contour fit plot produced by default when you have a model involving only two continuous predictors. The plot displays a contour plot of the predicted surface overlaid with a scatter plot of the observed data. You can use the following *obs-options* to control how the observations are displayed:

OBS=GRADIENT

specifies that observations are displayed as circles colored by the observed response. The same color gradient is used to display the fitted surface and the observations. Observations where the predicted response is close to the observed response have similar colors: the greater the contrast between the color of an observation and the surface, the larger the residual is at that point.

OBS=NONE

suppresses the observations.

OBS=OUTLINE

specifies that observations are displayed as circles with a border but with a completely transparent fill.

OBS=OUTLINEGRADIENT

is the same as OBS=GRADIENT except that a border is shown around each observation. This option is useful to identify the location of observations where the residuals are small, since at these points the color of the observations and the color of the surface are indistinguishable. OBS=OUTLINEGRADIENT is the default if you do not specify any *obs-options*.

CONTROLPLOT

requests a display in which least squares means are compared against a reference level. LS-mean control plots are produced only when you specify PDIFF=CONTROL or ADJUST=DUNNETT in the LSMEANS statement, and in this case they are produced by default.

DIAGNOSTICS< (LABEL UNPACK) >

requests that a panel of summary diagnostics for the fit be displayed. The panel displays scatter plots of residuals, absolute residuals, studentized residuals, and observed responses by predicted values; studentized residuals by leverage; Cook's D by observation; a Q-Q plot of residuals; a residual histogram; and a residual-fit spread plot. The LABEL option displays labels on observations satisfying RSTUDENT > 2, LEVERAGE $> 2p/n$, and on the Cook's D plot, COOKSD $> 4/n$, where n is the number of observations used in fitting the model, and p is the number of parameters in the model. The label is the first ID variable if the ID statement is specified; otherwise, it is the observation number. The UNPACK option unpanels the diagnostic display and produces the series of individual plots that form the paneled display.

DIFFPLOT< (ABS NOABS CENTER NOLINES) >

modifies the plot produced by an LSMEANS statement with the PDIFF=ALL option (or just PDIFF, since ALL is the default argument). The ABS and NOABS options determine the positioning of the line segments in the plot. When the ABS option is in effect, and this is the default, all line segments are shown on the same side of the reference line. The NOABS option separates comparisons according to the sign of the difference. The CENTER option marks the center point for each comparison. This point corresponds to the intersection of two least squares means. The NOLINES option suppresses the display of the line segments that represent the confidence bounds for the differences of the least squares means. The NOLINES option implies the CENTER option. The default is to draw line segments in the upper portion of the plot area without marking the center point.

FITPLOT< (NOCLM NOCLI NOLIMITS) >

modifies the fit plot produced by default when you have a model with a single continuous predictor. By default the plot includes confidence limits for both the expected predicted values and individual new predictions. The PLOTS=FITPLOT(NOCLM) option removes the limits on the expected values and the PLOTS=FITPLOT(NOCLI) option removes the limits on new predictions. The PLOTS=FITPLOT(NOLIMITS) option removes both kinds of confidence limits.

INTPLOT< (CLM CLI LIMITS) >

modifies the interaction plot produced by default when you have a two-way analysis of variance model, with just two CLASS variables. By default the plot does not show confidence limits around the predicted values. The PLOTS=INTPLOT(CLM) option adds limits for the expected predicted values and PLOTS=INTPLOT(CLI) adds limits for new predictions. Use PLOTS=ANCOVAPLOT(LIMITS) to add both kinds of limits.

MEANPLOT< (CL CLBAND CONNECT ASCENDING DESCENDING) >

modifies the grouped box plot produced by an MEANS statement. Upper and lower confidence limits are plotted when the CL option is used. When the CLBAND option is in effect, confidence limits are shown as bands and the means are connected. By default, means are not joined by lines. You can achieve that effect with the CONNECT option. Means are displayed in the same order as they appear in the "Means" table. You can change that order for plotting with the ASCENDING and DESCENDING options.

NONE
specifies that no graphics be displayed.

RESIDUALS< (SMOOTH UNPACK) >
requests that scatter plots of the residuals against each continuous covariate be displayed. The SMOOTH option overlays a Loess smooth on each residual plot. Note that if a WEIGHT variable is specified, then it is not used to weight the smoother. See Chapter 50, "The LOESS Procedure," for more information. The UNPACK option unpanels the residual display and produces a series of individual plots that form the paneled display.

ABSORB Statement

ABSORB *variables* **;**

Absorption is a computational technique that provides a large reduction in time and memory requirements for certain types of models. The *variables* are one or more variables in the input data set.

For a main-effect variable that does not participate in interactions, you can absorb the effect by naming it in an ABSORB statement. This means that the effect can be adjusted out before the construction and solution of the rest of the model. This is particularly useful when the effect has a large number of levels.

Several variables can be specified, in which case each one is assumed to be nested in the preceding variable in the ABSORB statement.

NOTE: When you use the ABSORB statement, the data set (or each BY group, if a BY statement appears) must be sorted by the variables in the ABSORB statement. The GLM procedure cannot produce predicted values or least squares means (LS-means) or create an output data set of diagnostic values if an ABSORB statement is used. If the ABSORB statement is used, it must appear before the first RUN statement or it is ignored.

When you use an ABSORB statement and also use the INT option in the MODEL statement, the procedure ignores the option but computes the uncorrected total sum of squares (SS) instead of the corrected total sums of squares.

See the section "Absorption" on page 2507 for more information.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC GLM to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the GLM procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

Since sorting the data changes the order in which PROC GLM reads observations, the sorting order for the levels of the classification variables might be affected if you have also specified ORDER=DATA in the PROC GLM statement. This, in turn, affects specifications in CONTRAST and ESTIMATE statements.

If you specify the BY statement, it must appear before the first RUN statement or it is ignored. When you use a BY statement, the interactive features of PROC GLM are disabled.

When both BY and ABSORB statements are used, observations must be sorted first by the variables in the BY statement, and then by the variables in the ABSORB statement.

For more information about the BY statement, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* < / *option* > **;**

The CLASS statement names the classification variables to be used in the model. Typical classification variables are TREATMENT, SEX, RACE, GROUP, and REPLICATION. If you specify the CLASS statement, it must appear before the MODEL statement.

By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. Before SAS 9, class levels were determined using no more than the first 16 characters of the formatted values. If you want to revert to this previous behavior, you can use the TRUNCATE option in the CLASS statement. In any case, you can use formats to group values into levels.

See the discussion of the FORMAT procedure in the *Base SAS Procedures Guide*, and the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

The GLM procedure displays a table summarizing the CLASS variables and their levels, and you can use this to check the ordering of levels and, hence, of the corresponding parameters for main effects. If you need to check the ordering of parameters for interaction effects, use the E option in the

MODEL, CONTRAST, ESTIMATE, and LSMEANS statements. See the section "Parameterization of PROC GLM Models" on page 2491 for more information.

You can specify the following option in the CLASS statement after a slash (/):

TRUNCATE

specifies that class levels should be determined using only up to the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases previous to SAS 9.

CONTRAST Statement

CONTRAST *'label' effect values < . . . effect values > < / options >* **;**

The CONTRAST statement enables you to perform custom hypothesis tests by specifying an **L** vector or matrix for testing the univariate hypothesis $\mathbf{L}\boldsymbol{\beta} = 0$ or the multivariate hypothesis $\mathbf{LBM} = 0$. Thus, to use this feature you must be familiar with the details of the model parameterization that PROC GLM uses. For more information, see the section "Parameterization of PROC GLM Models" on page 2491. All of the elements of the **L** vector might be given, or if only certain portions of the **L** vector are given, the remaining elements are constructed by PROC GLM from the context (in a manner similar to rule 4 discussed in the section "Construction of Least Squares Means" on page 2528).

There is no limit to the number of CONTRAST statements you can specify, but they must appear after the MODEL statement. In addition, if you use a CONTRAST statement and a MANOVA, REPEATED, or TEST statement, appropriate tests for contrasts are carried out as part of the MANOVA, REPEATED, or TEST analysis. If you use a CONTRAST statement and a RANDOM statement, the expected mean square of the contrast is displayed. As a result of these additional analyses, the CONTRAST statement must appear before the MANOVA, REPEATED, RANDOM, or TEST statement.

In the CONTRAST statement,

label identifies the contrast on the output. A label is required for every contrast specified. Labels must be enclosed in quotes.

effect identifies an effect that appears in the MODEL statement, or the INTERCEPT effect. The INTERCEPT effect can be used when an intercept is fitted in the model. You do not need to include all effects that are in the MODEL statement.

values are constants that are elements of the **L** vector associated with the effect.

You can specify the following options in the CONTRAST statement after a slash (/).

E

displays the entire **L** vector. This option is useful in confirming the ordering of parameters for specifying **L**.

E=*effect*

specifies an error term, which must be one of the effects in the model. The procedure uses this effect as the denominator in F tests in univariate analysis. In addition, if you use a MANOVA or REPEATED statement, the procedure uses the effect specified by the E= option as the basis of the **E** matrix. By default, the procedure uses the overall residual or error mean square (MSE) as an error term.

ETYPE=*n*

specifies the type (1, 2, 3, or 4, corresponding to a Type I, II, III, or IV test, respectively) of the E= effect. If the E= option is specified and the ETYPE= option is not, the procedure uses the highest type computed in the analysis.

SINGULAR=*number*

tunes the estimability checking. If ABS($\mathbf{L} - \mathbf{LH}$) $> C \times$*number* for any row in the contrast, then **L** is declared nonestimable. **H** is the $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$ matrix, and C is ABS(**L**) except for rows where **L** is zero, and then it is 1. The default value for the SINGULAR= option is 10^{-4}. Values for the SINGULAR= option must be between 0 and 1.

As stated previously, the CONTRAST statement enables you to perform custom hypothesis tests. If the hypothesis is testable in the univariate case, SS(H_0: $\mathbf{L}\boldsymbol{\beta} = 0$) is computed as

$$(\mathbf{Lb})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{Lb})$$

where $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{y}$. This is the sum of squares displayed on the analysis-of-variance table.

For multivariate testable hypotheses, the usual multivariate tests are performed using

$$\mathbf{H} = \mathbf{M}'(\mathbf{LB})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{LB})\mathbf{M}$$

where $\mathbf{B} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{Y}$ and **Y** is the matrix of multivariate responses or dependent variables. The degrees of freedom associated with the hypothesis are equal to the row rank of **L**. The sum of squares computed in this situation is equivalent to the sum of squares computed using an **L** matrix with any row deleted that is a linear combination of previous rows.

Multiple-degrees-of-freedom hypotheses can be specified by separating the rows of the **L** matrix with commas.

For example, for the model

```
proc glm;
   class A B;
   model Y=A B;
run;
```

with A at 5 levels and B at 2 levels, the parameter vector is

$$(\mu \;\; \alpha_1 \;\; \alpha_2 \;\; \alpha_3 \;\; \alpha_4 \;\; \alpha_5 \;\; \beta_1 \;\; \beta_2)$$

To test the hypothesis that the pooled A linear and A quadratic effect is zero, you can use the following **L** matrix:

$$\mathbf{L} = \begin{bmatrix} 0 & -2 & -1 & 0 & 1 & 2 & 0 & 0 \\ 0 & 2 & -1 & -2 & -1 & 2 & 0 & 0 \end{bmatrix}$$

The corresponding CONTRAST statement is

```
contrast 'A LINEAR & QUADRATIC'
         a -2 -1  0  1  2,
         a  2 -1 -2 -1  2;
```

If the first level of A is a control level and you want a test of control versus others, you can use this statement:

```
contrast 'CONTROL VS OTHERS'  a -1 0.25 0.25 0.25 0.25;
```

See the following discussion of the ESTIMATE statement and the section "Specification of ESTIMATE Expressions" on page 2509 for rules on specification, construction, distribution, and estimability in the CONTRAST statement.

ESTIMATE Statement

ESTIMATE *'label' effect values < . . . effect values > < / options >* **;**

The ESTIMATE statement enables you to estimate linear functions of the parameters by multiplying the vector $\mathbf{L}$ by the parameter estimate vector $\mathbf{b}$, resulting in $\mathbf{Lb}$. All of the elements of the $\mathbf{L}$ vector might be given, or, if only certain portions of the $\mathbf{L}$ vector are given, the remaining elements are constructed by PROC GLM from the context (in a manner similar to rule 4 discussed in the section "Construction of Least Squares Means" on page 2528).

The linear function is checked for estimability. The estimate $\mathbf{Lb}$, where $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{y}$, is displayed along with its associated standard error, $\sqrt{\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}'s^2}$, and t test. If you specify the CLPARM option in the MODEL statement (see page 2475), confidence limits for the true value are also displayed.

There is no limit to the number of ESTIMATE statements that you can specify, but they must appear after the MODEL statement. In the ESTIMATE statement,

label identifies the estimate on the output. A label is required for every contrast specified. Labels must be enclosed in quotes.

effect identifies an effect that appears in the MODEL statement, or the INTERCEPT effect. The INTERCEPT effect can be used as an effect when an intercept is fitted in the model. You do not need to include all effects that are in the MODEL statement.

values are constants that are the elements of the $\mathbf{L}$ vector associated with the preceding effect. For example,

```
estimate 'A1 VS A2' A  1 -1;
```

forms an estimate that is the difference between the parameters estimated for the first and second levels of the CLASS variable A.

You can specify the following options in the ESTIMATE statement after a slash (/):

DIVISOR=*number*

specifies a value by which to divide all coefficients so that fractional coefficients can be entered as integer numerators. For example, you can use

```
estimate '1/3(A1+A2) - 2/3A3' a 1 1 -2 / divisor=3;
```

instead of

```
estimate '1/3(A1+A2) - 2/3A3' a 0.33333 0.33333 -0.66667;
```

E

displays the entire **L** vector. This option is useful in confirming the ordering of parameters for specifying **L**.

SINGULAR=*number*

tunes the estimability checking. If ABS($\mathbf{L}-\mathbf{LH}$) $> C \times number$, then the **L** vector is declared nonestimable. **H** is the $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$ matrix, and C is ABS(**L**) except for rows where **L** is zero, and then it is 1. The default value for the SINGULAR= option is 10^{-4}. Values for the SINGULAR= option must be between 0 and 1.

See also the section "Specification of ESTIMATE Expressions" on page 2509.

FREQ Statement

FREQ *variable* **;**

The FREQ statement names a variable that provides frequencies for each observation in the DATA= data set. Specifically, if n is the value of the FREQ variable for a given observation, then that observation is used n times.

The analysis produced using a FREQ statement reflects the expanded number of observations. For example, means and total degrees of freedom reflect the expanded number of observations. You can produce the same analysis (without the FREQ statement) by first creating a new data set that contains the expanded number of observations. For example, if the value of the FREQ variable is 5 for the first observation, the first 5 observations in the new data set are identical. Each observation in the old data set is replicated n_i times in the new data set, where n_i is the value of the FREQ variable for that observation.

If the value of the FREQ variable is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

If you specify the FREQ statement, it must appear before the first RUN statement or it is ignored.

ID Statement

ID *variables* **;**

When predicted values are requested as a MODEL statement option, values of the variables given in the ID statement are displayed beside each observed, predicted, and residual value for identification. Although there are no restrictions on the length of ID variables, PROC GLM might truncate the number of values listed in order to display them on one line. The GLM procedure displays a maximum of five ID variables.

If you specify the ID statement, it must appear before the first RUN statement or it is ignored.

LSMEANS Statement

LSMEANS *effects* **<** */ options* **> ;**

Least squares means (LS-means) are computed for each *effect* listed in the LSMEANS statement. You can specify only classification effects in the LSMEANS statement—that is, effects that contain only classification variables. You can also specify options to perform multiple comparisons. In contrast to the MEANS statement, the LSMEANS statement performs multiple comparisons on interactions as well as main effects.

LS-means are *predicted population margins*; that is, they estimate the marginal means over a balanced population. In a sense, LS-means are to unbalanced designs as class and subclass arithmetic means are to balanced designs. Each LS-mean is computed as $\mathbf{L}'\mathbf{b}$ for a certain column vector $\mathbf{L}$, where $\mathbf{b}$ is the vector of parameter estimates—that is, the solution of the normal equations. For further information, see the section "Construction of Least Squares Means" on page 2528.

Multiple effects can be specified in one LSMEANS statement, or multiple LSMEANS statements can be used, but they must all appear after the MODEL statement. For example:

```
proc glm;
   class A B;
   model Y=A B A*B;
   lsmeans A B A*B;
run;
```

LS-means are displayed for each level of the A, B, and A*B effects.

You can specify the following options in the LSMEANS statement after a slash (/):

ADJUST=BON

ADJUST=DUNNETT

ADJUST=NELSON

ADJUST=SCHEFFE

ADJUST=SIDAK

ADJUST=SIMULATE <(*simoptions***)>**

ADJUST=SMM | GT2

ADJUST=TUKEY

ADJUST=T

requests a multiple comparison adjustment for the p-values and confidence limits for the differences of LS-means. The ADJUST= option modifies the results of the TDIFF and PDIFF options; thus, if you omit the TDIFF or PDIFF option then the ADJUST= option has no effect. By default, PROC GLM analyzes all pairwise differences. If you specify ADJUST=DUNNETT, PROC GLM analyzes all differences with a control level. If you specify the ADJUST=NELSON option, PROC GLM analyzes all differences with the average LS-mean. The default is ADJUST=T, which really signifies no adjustment for multiple comparisons.

The BON (Bonferroni) and SIDAK adjustments involve correction factors described in the section "Multiple Comparisons" on page 2514 and in Chapter 58, "The MULTTEST Procedure." When you specify ADJUST=TUKEY and your data are unbalanced, PROC GLM uses the approximation described in Kramer (1956) and identifies the adjustment as "Tukey-Kramer" in the results. Similarly, when you specify either ADJUST=DUNNETT or the ADJUST=NELSON option and the LS-means are correlated, PROC GLM uses the factor-analytic covariance approximation described in Hsu (1992) and identifies the adjustment in the results as "Dunnett-Hsu" or "Nelson-Hsu," respectively. The preceding references also describe the SCHEFFE and SMM adjustments.

The SIMULATE adjustment computes the adjusted p-values from the simulated distribution of the maximum or maximum absolute value of a multivariate t random vector. The simulation estimates q, the true $(1-\alpha)$th quantile, where $1-\alpha$ is the confidence coefficient. The default α is the value of the ALPHA= option in the PROC GLM statement or 0.05 if that option is not specified. You can change this value with the ALPHA= option in the LSMEANS statement.

The number of samples for the SIMULATE adjustment is set so that the tail area for the simulated q is within a certain *accuracy radius* γ of $1-\alpha$ with an *accuracy confidence* of $100(1-\epsilon)\%$. In equation form,

$$P(|F(\hat{q}) - (1-\alpha)| \leq \gamma) \quad = \quad 1-\epsilon$$

where $\hat{q}$ is the simulated q and F is the true distribution function of the maximum; see Edwards and Berry (1987) for details. By default, $\gamma = 0.005$ and $\epsilon = 0.01$, so that the tail area of $\hat{q}$ is within 0.005 of 0.95 with 99% confidence.

You can specify the following *simoptions* in parentheses after the ADJUST=SIMULATE option.

ACC=*value* specifies the target accuracy radius γ of a $100(1-\epsilon)\%$ confidence interval for the true probability content of the estimated $(1-\alpha)$th quantile. The default value is ACC=0.005. Note that, if you also specify the CVADJUST *simoption*, then the actual accuracy radius will probably be substantially less than this target.

CVADJUST specifies that the quantile should be estimated by the control variate adjustment method of Hsu and Nelson (1998) instead of simply as the quan-

tile of the simulated sample. Specifying the CVADJUST option typically has the effect of significantly reducing the accuracy radius γ of a $100 \times (1 - \epsilon)\%$ confidence interval for the true probability content of the estimated $(1 - \alpha)$th quantile. The control-variate-adjusted quantile estimate takes roughly twice as long to compute, but it is typically much more accurate than the sample quantile.

EPS=*value* specifies the value ϵ for a $100 \times (1 - \epsilon)\%$ confidence interval for the true probability content of the estimated $(1 - \alpha)$th quantile. The default value for the accuracy confidence is 99%, corresponding to EPS=0.01.

NSAMP=*n* specifies the sample size for the simulation. By default, n is set based on the values of the target accuracy radius γ and accuracy confidence $100 \times (1 - \epsilon)\%$ for an interval for the true probability content of the estimated $(1 - \alpha)$th quantile. With the default values for γ, ϵ, and α (0.005, 0.01, and 0.05, respectively), NSAMP=12604 by default.

REPORT specifies that a report on the simulation should be displayed, including a listing of the parameters, such as γ, ϵ, and α, as well as an analysis of various methods for estimating or approximating the quantile.

SEED=*number* specifies an integer used to start the pseudo-random number generator for the simulation. If you do not specify a seed, or specify a value less than or equal to zero, the seed is by default generated from reading the time of day from the computer's clock.

ALPHA=*p*

specifies the level of significance p for $100(1 - p)\%$ confidence intervals. This option is useful only if you also specify the CL option, and, optionally, the PDIFF option. By default, p is equal to the value of the ALPHA= option in the PROC GLM statement or 0.05 if that option is not specified, This value is used to set the endpoints for confidence intervals for the individual means as well as for differences between means.

AT *variable* **=** *value*

AT (*variable-list***) = (***value-list***)**

AT MEANS

enables you to modify the values of the covariates used in computing LS-means. By default, all covariate effects are set equal to their mean values for computation of standard LS-means. The AT option enables you to set the covariates to whatever values you consider interesting. For more information, see the section "Setting Covariate Values" on page 2529.

BYLEVEL

requests that PROC GLM process the OM data set by each level of the LS-mean effect in question. For more details, see the entry for the OM option in this section.

CL

requests confidence limits for the individual LS-means. If you specify the PDIFF option, confidence limits for differences between means are produced as well. You can control the confidence level with the ALPHA= option. Note that, if you specify an ADJUST= option, the confidence limits for the differences are adjusted for multiple inference but the confidence intervals for individual means are **not** adjusted.

COV

includes variances and covariances of the LS-means in the output data set specified in the OUT= option in the LSMEANS statement. Note that this is the covariance matrix for the LS-means themselves, not the covariance matrix for the differences between the LS-means, which is used in the PDIFF computations. If you omit the OUT= option, the COV option has no effect. When you specify the COV option, you can specify only one effect in the LSMEANS statement.

E

displays the coefficients of the linear functions used to compute the LS-means.

E=*effect*

specifies an effect in the model to use as an error term. The procedure uses the mean square for the *effect* as the error mean square when calculating estimated standard errors (requested with the STDERR option) and probabilities (requested with the STDERR, PDIFF, or TDIFF option). Unless you specify STDERR, PDIFF or TDIFF, the E= option is ignored. By default, if you specify the STDERR, PDIFF, or TDIFF option and do not specify the E= option, the procedure uses the error mean square for calculating standard errors and probabilities.

ETYPE=*n*

specifies the type (1, 2, 3, or 4, corresponding to a Type I, II, III, or IV test, respectively) of the E= effect. If you specify the E= option but not the ETYPE= option, the highest type computed in the analysis is used. If you omit the E= option, the ETYPE= option has no effect.

LINES

presents results of comparisons between all pairs of means (specified by the PDIFF=ALL option) by listing the means in descending order and indicating nonsignificant subsets by line segments beside the corresponding means. When all differences have the same variance, these comparison lines are guaranteed to accurately reflect the inferences based on the corresponding tests, made by comparing the respective p-values to the value of the ALPHA= option (0.05 by default). However, equal variances are rarely the case for differences between LS-means. If the variances are not all the same, then the comparison lines might be conservative, in the sense that if you base your inferences on the lines alone, you will detect fewer significant differences than the tests indicate. If there are any such differences, a note is appended to the table that lists the pairs of means that are inferred to be significantly different by the tests but not by the comparison lines. Note, however, that in many cases, even though the variances are unbalanced, they are near enough that the comparison lines in fact accurately reflect the test inferences.

NOPRINT

suppresses the normal display of results from the LSMEANS statement. This option is useful when an output data set is created with the OUT= option in the LSMEANS statement.

OBSMARGINS

OM

specifies a potentially different weighting scheme for computing LS-means coefficients. The standard LS-means have equal coefficients across classification effects; however, the OM option changes these coefficients to be proportional to those found in the input data set. For more information, see the section "Changing the Weighting Scheme" on page 2530.

The BYLEVEL option modifies the observed-margins LS-means. Instead of computing the margins across the entire data set, the procedure computes separate margins for each level of the LS-mean effect in question. The resulting LS-means are actually equal to raw means in this case. If you specify the BYLEVEL option, it disables the AT option.

OUT=*SAS-data-set*

creates an output data set that contains the values, standard errors, and, optionally, the covariances (see the COV option) of the LS-means.

For more information, see the section "Output Data Sets" on page 2548.

PDIFF< *=difftype* >

requests that *p*-values for differences of the LS-means be produced. The optional *difftype* specifies which differences to display. Possible values for *difftype* are ALL, CONTROL, CONTROLL, CONTROLU, and ANOM. The ALL value requests all pairwise differences, and it is the default. The CONTROL value requests the differences with a control that, by default, is the first level of each of the specified LS-mean effects. The ANOM value requests differences between each LS-mean and the average LS-mean, as in the *analysis of means* (Ott 1967). The average is computed as a weighted mean of the LS-means, the weights being inversely proportional to the variances. Note that the ANOM procedure in SAS/QC software implements both tables and graphics for the analysis of means with a variety of response types. For one-way designs, the PDIFF=ANOM computations are equivalent to the results of PROC ANOM. See the section "Analysis of Means: Comparing Each Treatments to the Average" on page 2520 for more details.

To specify which levels of the effects are the controls, list the quoted formatted values in parentheses after the keyword CONTROL. For example, if the effects A, B, and C are CLASS variables, each having two levels, '1' and '2', the following LSMEANS statement specifies the '1' '2' level of A*B and the '2' '1' level of B*C as controls:

```
lsmeans A*B B*C / pdiff=control('1' '2', '2' '1');
```

For multiple-effect situations such as this one, the ordering of the list is significant, and you should check the output to make sure that the controls are correct.

Two-tailed tests and confidence limits are associated with the CONTROL difftype. For one-tailed results, use either the CONTROLL or CONTROLU difftype.

- PDIFF=CONTROLL tests whether the noncontrol levels are less than the control; you declare a noncontrol level to be significantly less than the control if the associated upper confidence limit for the noncontrol level minus the control is less than zero, and you ignore the associated lower confidence limits (which are set to minus infinity).
- PDIFF=CONTROLU tests whether the noncontrol levels are greater than the control; you declare a noncontrol level to be significantly greater than the control if the associated lower confidence limit for the noncontrol level minus the control is greater than zero, and you ignore the associated upper confidence limits (which are set to infinity).

The default multiple comparisons adjustment for each *difftype* is shown in the following table.

difftype	Default ADJUST=
Not specified	T
ALL	TUKEY
CONTROL CONTROLL CONTROLU	DUNNETT
ANOM	NELSON

If no *difftype* is specified, the default for the ADJUST= option is T (that is, no adjustment); for PDIFF=ALL, ADJUST=TUKEY is the default; for PDIFF=CONTROL, PDIFF=CONTROLL, or PDIFF=CONTROLU, the default value for the ADJUST= option is DUNNETT. For PDIFF=ANOM, ADJUST=NELSON is the default. If there is a conflict between the PDIFF= and ADJUST= options, the ADJUST= option takes precedence.

For example, in order to compute one-sided confidence limits for differences with a control, adjusted according to Dunnett's procedure, the following statements are equivalent:

```
lsmeans Treatment / pdiff=controll cl;
lsmeans Treatment / pdiff=controll cl adjust=dunnett;
```

SLICE=*fixed-effect*

SLICE=(*fixed-effects***)**

specifies effects within which to test for differences between interaction LS-mean effects. This can produce what are known as tests of simple effects (Winer 1971). For example, suppose that A*B is significant and you want to test for the effect of A within each level of B. The appropriate LSMEANS statement is

```
lsmeans A*B / slice=B;
```

This statement tests for the simple main effects of A for B, which are calculated by extracting the appropriate rows from the coefficient matrix for the A*B LS-means and using them to form an F test as performed by the CONTRAST statement.

SINGULAR=*number*

tunes the estimability checking. If ABS($\mathbf{L} - \mathbf{LH}$) $> C \times$*number* for any row, then $\mathbf{L}$ is declared nonestimable. $\mathbf{H}$ is the $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$ matrix, and C is ABS($\mathbf{L}$) except for rows where $\mathbf{L}$ is zero, and then it is 1. The default value for the SINGULAR= option is 10^{-4}. Values for the SINGULAR= option must be between 0 and 1.

STDERR

produces the standard error of the LS-means and the probability level for the hypothesis H_0: LS-mean $= 0$.

TDIFF

produces the t values for all hypotheses H_0: LS-mean(i) $=$ LS-mean(j) and the corresponding probabilities.

MANOVA Statement

MANOVA < *test-options* > < *detail-options* > **;**

If the MODEL statement includes more than one dependent variable, you can perform multivariate analysis of variance with the MANOVA statement. The *test-options* define which effects to test, while the *detail-options* specify how to execute the tests and what results to display.

When a MANOVA statement appears before the first RUN statement, PROC GLM enters a multivariate mode with respect to the handling of missing values; in addition to observations with missing independent variables, observations with *any* missing dependent variables are excluded from the analysis. If you want to use this mode of handling missing values and do not need any multivariate analyses, specify the MANOVA option in the PROC GLM statement.

If you use both the CONTRAST and MANOVA statements, the MANOVA statement must appear after the CONTRAST statement.

Test Options

The following options can be specified in the MANOVA statement as *test-options* in order to define which multivariate tests to perform.

H=*effects* **| INTERCEPT | _ALL_**
specifies effects in the preceding model to use as hypothesis matrices. For each $\mathbf{H}$ matrix (the SSCP matrix associated with an effect), the H= specification displays the characteristic roots and vectors of $\mathbf{E}^{-1}\mathbf{H}$ (where $\mathbf{E}$ is the matrix associated with the error effect), along with the Hotelling-Lawley trace, Pillai's trace, Wilks' lambda, and Roy's greatest root. By default, these statistics are tested with approximations based on the F distribution. To test them with exact (but computationally intensive) calculations, use the MSTAT=EXACT option.

Use the keyword INTERCEPT to produce tests for the intercept. To produce tests for all effects listed in the MODEL statement, use the keyword _ALL_ in place of a list of effects.

For background and further details, see the section "Multivariate Analysis of Variance" on page 2531.

E=*effect*
specifies the error effect. If you omit the E= specification, the GLM procedure uses the error SSCP (residual) matrix from the analysis.

M=*equation,...,equation* **|** (*row-of-matrix,...,row-of-matrix*)
specifies a transformation matrix for the dependent variables listed in the MODEL statement. The equations in the M= specification are of the form

$$\begin{aligned} c_1 \times \textit{dependent-variable} \quad &\pm \quad c_2 \times \textit{dependent-variable} \\ \cdots \quad &\pm \quad c_n \times \textit{dependent-variable} \end{aligned}$$

where the c_i values are coefficients for the various *dependent-variables*. If the value of a given c_i is 1, it can be omitted; in other words $1 \times Y$ is the same as Y. Equations should involve two or more dependent variables. For sample syntax, see the section "Examples" on page 2466.

Alternatively, you can input the transformation matrix directly by entering the elements of the matrix with commas separating the rows and parentheses surrounding the matrix. When this alternate form of input is used, the number of elements in each row must equal the number of dependent variables. Although these combinations actually represent the columns of the **M** matrix, they are displayed by rows.

When you include an M= specification, the analysis requested in the MANOVA statement is carried out for the variables defined by the equations in the specification, not the original dependent variables. If you omit the M= option, the analysis is performed for the original dependent variables in the MODEL statement.

If an M= specification is included without either the MNAMES= or PREFIX= option, the variables are labeled MVAR1, MVAR2, and so forth, by default. For further information, see the section "Multivariate Analysis of Variance" on page 2531.

MNAMES=*names*
provides names for the variables defined by the equations in the M= specification. Names in the list correspond to the M= equations or to the rows of the **M** matrix (as it is entered).

PREFIX=*name*
is an alternative means of identifying the transformed variables defined by the M= specification. For example, if you specify PREFIX=DIFF, the transformed variables are labeled DIFF1, DIFF2, and so forth.

Detail Options

You can specify the following options in the MANOVA statement after a slash (/) as *detail-options*.

CANONICAL
displays a canonical analysis of the **H** and **E** matrices (transformed by the **M** matrix, if specified) instead of the default display of characteristic roots and vectors.

ETYPE=*n*
specifies the type (1, 2, 3, or 4, corresponding to a Type I, II, III, or IV test, respectively) of the **E** matrix, the SSCP matrix associated with the E= effect. You need this option if you use the E= specification to specify an error effect other than residual error and you want to specify the type of sums of squares used for the effect. If you specify ETYPE=*n*, the corresponding test must have been performed in the MODEL statement, either by options SS*n*, E*n*, or the default Type I and Type III tests. By default, the procedure uses an ETYPE= value corresponding to the highest type (largest *n*) used in the analysis.

HTYPE=*n*
specifies the type (1, 2, 3, or 4, corresponding to a Type I, II, III, or IV test, respectively) of the **H** matrix. See the ETYPE= option for more details.

MSTAT=FAPPROX

MSTAT=EXACT

specifies the method of evaluating the multivariate test statistics. The default is MSTAT=FAPPROX, which specifies that the multivariate tests are evaluated using the usual approximations based on the F distribution, as discussed in the section "Multivariate Tests" in Chapter 4, "Introduction to Regression Procedures." Alternatively, you can specify MSTAT=EXACT to compute exact p-values for three of the four tests (Wilks' lambda, the Hotelling-Lawley trace, and Roy's greatest root) and an improved F approximation for the fourth (Pillai's trace). While MSTAT=EXACT provides better control of the significance probability for the tests, especially for Roy's greatest root, computations for the exact p-values can be appreciably more demanding, and are in fact infeasible for large problems (many dependent variables). Thus, although MSTAT=EXACT is more accurate for most data, it is not the default method. For more information about the results of MSTAT=EXACT, see the section "Multivariate Analysis of Variance" on page 2531.

ORTH

requests that the transformation matrix in the M= specification of the MANOVA statement be orthonormalized by rows before the analysis.

PRINTE

displays the error SSCP matrix **E**. If the **E** matrix is the error SSCP (residual) matrix from the analysis, the partial correlations of the dependent variables given the independent variables are also produced.

For example, the statement

```
manova / printe;
```

displays the error SSCP matrix and the partial correlation matrix computed from the error SSCP matrix.

PRINTH

displays the hypothesis SSCP matrix **H** associated with each effect specified by the H= specification.

SUMMARY

produces analysis-of-variance tables for each dependent variable. When no **M** matrix is specified, a table is displayed for each original dependent variable from the MODEL statement; with an **M** matrix other than the identity, a table is displayed for each transformed variable defined by the **M** matrix.

Examples

The following statements provide several examples of using a MANOVA statement.

```
proc glm;
   class A B;
   model Y1-Y5=A B(A) / nouni;
   manova h=A e=B(A) / printh printe htype=1 etype=1;
   manova h=B(A) / printe;
```

```
   manova h=A e=B(A) m=Y1-Y2,Y2-Y3,Y3-Y4,Y4-Y5
          prefix=diff;
   manova h=A e=B(A) m=(1 -1  0  0  0,
                        0  1 -1  0  0,
                        0  0  1 -1  0,
                        0  0  0  1 -1) prefix=diff;
run;
```

Since this MODEL statement requests no options for type of sums of squares, the procedure uses Type I and Type III sums of squares. The first MANOVA statement specifies A as the hypothesis effect and B(A) as the error effect. As a result of the PRINTH option, the procedure displays the hypothesis SSCP matrix associated with the A effect; and, as a result of the PRINTE option, the procedure displays the error SSCP matrix associated with the B(A) effect. The option HTYPE=1 specifies a Type I **H** matrix, and the option ETYPE=1 specifies a Type I **E** matrix.

The second MANOVA statement specifies B(A) as the hypothesis effect. Since no error effect is specified, PROC GLM uses the error SSCP matrix from the analysis as the **E** matrix. The PRINTE option displays this **E** matrix. Since the **E** matrix is the error SSCP matrix from the analysis, the partial correlation matrix computed from this matrix is also produced.

The third MANOVA statement requests the same analysis as the first MANOVA statement, but the analysis is carried out for variables transformed to be successive differences between the original dependent variables. The option PREFIX=DIFF labels the transformed variables as DIFF1, DIFF2, DIFF3, and DIFF4.

Finally, the fourth MANOVA statement has the identical effect as the third, but it uses an alternative form of the M= specification. Instead of specifying a set of equations, the fourth MANOVA statement specifies rows of a matrix of coefficients for the five dependent variables.

As a second example of the use of the M= specification, consider the following:

```
proc glm;
   class group;
   model dose1-dose4=group / nouni;
   manova h = group
          m = -3*dose1 -   dose2 +   dose3 + 3*dose4,
                 dose1 -   dose2 -   dose3 +   dose4,
                -dose1 + 3*dose2 - 3*dose3 +   dose4
          mnames = Linear Quadratic Cubic
          / printe;
run;
```

The M= specification gives a transformation of the dependent variables dose1 through dose4 into orthogonal polynomial components, and the MNAMES= option labels the transformed variables LINEAR, QUADRATIC, and CUBIC, respectively. Since the PRINTE option is specified and the default residual matrix is used as an error term, the partial correlation matrix of the orthogonal polynomial components is also produced.

MEANS Statement

MEANS *effects* < / *options* > **;**

Within each group corresponding to each effect specified in the MEANS statement, PROC GLM computes the arithmetic means and standard deviations of all continuous variables in the model (both dependent and independent). You can specify only classification effects in the MEANS statement—that is, effects that contain only classification variables.

Note that the arithmetic means are not adjusted for other effects in the model; for adjusted means, see the section "LSMEANS Statement" on page 2458.

If you use a WEIGHT statement, PROC GLM computes weighted means; see the section "Weighted Means" on page 2527.

You can also specify options to perform multiple comparisons. However, the MEANS statement performs multiple comparisons only for main-effect means; for multiple comparisons of interaction means, see the section "LSMEANS Statement" on page 2458.

You can use any number of MEANS statements, provided that they appear after the MODEL statement. For example, suppose A and B each have two levels. Then, if you use the statements

```
proc glm;
   class A B;
   model Y=A B A*B;
   means A B / tukey;
   means A*B;
run;
```

the means, standard deviations, and Tukey's multiple comparisons tests are displayed for each level of the main effects A and B, and just the means and standard deviations are displayed for each of the four combinations of levels for A*B. Since multiple comparisons tests apply only to main effects, the single MEANS statement

```
means A B A*B / tukey;
```

produces the same results.

PROC GLM does not compute means for interaction effects containing continuous variables. Thus, if you have the model

```
class A;
model Y=A X A*X;
```

then the effects X and A*X cannot be used in the MEANS statement. However, if you specify the effect A in the means statement

```
means A;
```

then PROC GLM, by default, displays within-A arithmetic means of both Y and X. You can use the DEPONLY option to display means of only the dependent variables.

```
means A / deponly;
```

If you use a WEIGHT statement, PROC GLM computes weighted means and estimates their variance as inversely proportional to the corresponding sum of weights (see the section "Weighted Means" on page 2527). However, note that the statistical interpretation of multiple comparison tests for weighted means is not well understood. See the section "Multiple Comparisons" on page 2514 for formulas. Table 39.4 summarizes categories of options available in the MEANS statement.

Table 39.4 MEANS Statement Options

Task	Available Options
Modify output	DEPONLY
Perform multiple comparison tests	BON DUNCAN DUNNETT DUNNETTL DUNNETTU GABRIEL GT2 LSD REGWQ SCHEFFE SIDAK SMM SNK T TUKEY WALLER
Specify additional details for multiple comparison tests	ALPHA= CLDIFF CLM E= ETYPE= HTYPE= KRATIO= LINES NOSORT
Test for homogeneity of variances	HOVTEST
Compensate for heterogeneous variances	WELCH

The options available in the MEANS statement are described in the following list.

ALPHA=

ALPHA=*p* specifies the level of significance for comparisons among the means. By default, *p* is equal to the value of the ALPHA= option in the PROC GLM statement or 0.05 if that option is not specified. You can specify any value greater than 0 and less than 1.

BON

performs Bonferroni *t* tests of differences between means for all main-effect means in the MEANS statement. See the CLDIFF and LINES options for a discussion of how the procedure displays results.

CLDIFF

presents results of the BON, GABRIEL, SCHEFFE, SIDAK, SMM, GT2, T, LSD, and TUKEY options as confidence intervals for all pairwise differences between means, and the results of the DUNNETT, DUNNETTU, and DUNNETTL options as confidence intervals for differences with the control. The CLDIFF option is the default for unequal cell sizes unless the DUNCAN, REGWQ, SNK, or WALLER option is specified.

CLM

presents results of the BON, GABRIEL, SCHEFFE, SIDAK, SMM, T, and LSD options as intervals for the mean of each level of the variables specified in the MEANS statement. For all options except GABRIEL, the intervals are confidence intervals for the true means. For the GABRIEL option, they are *comparison intervals* for comparing means pairwise: in this case, if the intervals corresponding to two means overlap, then the difference between them is insignificant according to Gabriel's method.

DEPONLY

displays only means for the dependent variables. By default, PROC GLM produces means for all continuous variables, including continuous independent variables.

DUNCAN

performs Duncan's multiple range test on all main-effect means given in the MEANS statement. See the LINES option for a discussion of how the procedure displays results.

DUNNETT <(*formatted-control-values*)>

performs Dunnett's two-tailed *t* test, testing if any treatments are significantly different from a single control for all main-effect means in the MEANS statement.

To specify which level of the effect is the control, enclose the formatted value in quotes and parentheses after the keyword. If more than one effect is specified in the MEANS statement, you can use a list of control values within the parentheses. By default, the first level of the effect is used as the control. For example:

```
means A  / dunnett('CONTROL');
```

where CONTROL is the formatted control value of A. As another example:

```
means A B C / dunnett('CNTLA' 'CNTLB' 'CNTLC');
```

where CNTLA, CNTLB, and CNTLC are the formatted control values for A, B, and C, respectively.

DUNNETTL <(*formatted-control-value*)>

performs Dunnett's one-tailed *t* test, testing if any treatment is significantly less than the control. Control level information is specified as described for the DUNNETT option.

DUNNETTU <(*formatted-control-value*)>

performs Dunnett's one-tailed *t* test, testing if any treatment is significantly greater than the control. Control level information is specified as described for the DUNNETT option.

E=*effect*

specifies the error mean square used in the multiple comparisons. By default, PROC GLM uses the overall residual or error mean square (MS). The effect specified with the E= option must be a term in the model; otherwise, the procedure uses the residual MS.

ETYPE=*n*

specifies the type of mean square for the error effect. When you specify E=*effect*, you might need to indicate which type (1, 2, 3, or 4) of MS is to be used. The *n* value must be one of the types specified in or implied by the MODEL statement. The default MS type is the highest type used in the analysis.

GABRIEL

performs Gabriel's multiple-comparison procedure on all main-effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

GT2

See the SMM option.

HOVTEST

HOVTEST=BARTLETT

HOVTEST=BF

HOVTEST=LEVENE < (TYPE= ABS | SQUARE) >

HOVTEST=OBRIEN < (W=*number* **) >**

requests a homogeneity of variance test for the groups defined by the MEANS effect. You can optionally specify a particular test; if you do not specify a test, Levene's test (Levene 1960) with TYPE=SQUARE is computed. Note that this option is ignored unless your MODEL statement specifies a simple one-way model.

The HOVTEST=BARTLETT option specifies Bartlett's test (Bartlett 1937), a modification of the normal-theory likelihood ratio test.

The HOVTEST=BF option specifies Brown and Forsythe's variation of Levene's test (Brown and Forsythe 1974).

The HOVTEST=LEVENE option specifies Levene's test (Levene 1960), which is widely considered to be the standard homogeneity of variance test. You can use the TYPE= option in parentheses to specify whether to use the absolute residuals (TYPE=ABS) or the squared residuals (TYPE=SQUARE) in Levene's test. TYPE=SQUARE is the default.

The HOVTEST=OBRIEN option specifies O'Brien's test (O'Brien 1979), which is basically a modification of HOVTEST=LEVENE(TYPE=SQUARE). You can use the W= option in parentheses to tune the variable to match the suspected kurtosis of the underlying distribution. By default, W=0.5, as suggested by O'Brien (1979, 1981).

See the section "Homogeneity of Variance in One-Way Models" on page 2526 for more details on these methods. Example 39.10 illustrates the use of the HOVTEST and WELCH options in the MEANS statement in testing for equal group variances and adjusting for unequal group variances in a one-way ANOVA.

HTYPE=*n*

specifies the MS type for the hypothesis MS. The HTYPE= option is needed only when the WALLER option is specified. The default HTYPE= value is the highest type used in the model.

KRATIO=*value*

specifies the Type 1/Type 2 error seriousness ratio for the Waller-Duncan test. Reasonable values for the KRATIO= option are 50, 100, 500, which roughly correspond for the two-level case to ALPHA levels of 0.1, 0.05, and 0.01, respectively. By default, the procedure uses the value of 100.

LINES

presents results of the BON, DUNCAN, GABRIEL, REGWQ, SCHEFFE, SIDAK, SMM, GT2, SNK, T, LSD, TUKEY, and WALLER options by listing the means in descending order and indicating nonsignificant subsets by line segments beside the corresponding means. The LINES option is appropriate for equal cell sizes, for which it is the default. The LINES option is also the default if the DUNCAN, REGWQ, SNK, or WALLER option is specified, or if there are only two cells of unequal size. The LINES option cannot be used in combination with the DUNNETT, DUNNETTL, or DUNNETTU option. In addition, the procedure has a restriction that no more than 24 overlapping groups of means can exist. If a mean belongs to more than 24 groups, the procedure issues an error message. You can either reduce the number of levels of the variable or use a multiple comparison test that allows the CLDIFF option rather than the LINES option.

NOTE: If the cell sizes are unequal, the harmonic mean of the cell sizes is used to compute the critical ranges. This approach is reasonable if the cell sizes are not too different, but it can lead to liberal tests if the cell sizes are highly disparate. In this case, you should not use the LINES option for displaying multiple comparisons results; use the TUKEY and CLDIFF options instead.

LSD

See the T option.

NOSORT

prevents the means from being sorted into descending order when the CLDIFF or CLM option is specified.

REGWQ

performs the Ryan-Einot-Gabriel-Welsch multiple range test on all main-effect means in the MEANS statement. See the LINES option for a discussion of how the procedure displays results.

SCHEFFE

performs Scheffé's multiple-comparison procedure on all main-effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

SIDAK

performs pairwise t tests on differences between means with levels adjusted according to Sidak's inequality for all main-effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

SMM

GT2

performs pairwise comparisons based on the studentized maximum modulus and Sidak's uncorrelated-t inequality, yielding Hochberg's GT2 method when sample sizes are unequal, for all main-effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

SNK

performs the Student-Newman-Keuls multiple range test on all main-effect means in the MEANS statement. See the LINES option for discussions of how the procedure displays results.

T

LSD

performs pairwise t tests, equivalent to Fisher's least significant difference test in the case of equal cell sizes, for all main-effect means in the MEANS statement. See the CLDIFF and LINES options for discussions of how the procedure displays results.

TUKEY

performs Tukey's studentized range test (HSD) on all main-effect means in the MEANS statement. (When the group sizes are different, this is the Tukey-Kramer test.) See the CLDIFF and LINES options for discussions of how the procedure displays results.

WALLER

performs the Waller-Duncan k-ratio t test on all main-effect means in the MEANS statement. See the KRATIO= and HTYPE= options for information about controlling details of the test, and the LINES option for a discussion of how the procedure displays results.

WELCH

requests the variance-weighted one-way ANOVA of Welch (1951). This alternative to the usual analysis of variance for a one-way model is robust to the assumption of equal within-group variances. This option is ignored unless your MODEL statement specifies a simple one-way model.

Note that using the WELCH option merely produces one additional table consisting of Welch's ANOVA. It does not affect all of the other tests displayed by the GLM procedure, which still require the assumption of equal variance for exact validity.

See the section "Homogeneity of Variance in One-Way Models" on page 2526 for more details on Welch's ANOVA. Example 39.10 illustrates the use of the HOVTEST and WELCH options in the MEANS statement in testing for equal group variances and adjusting for unequal group variances in a one-way ANOVA.

MODEL Statement

MODEL *dependents=independents* < / *options* > **;**

The MODEL statement names the dependent variables and independent effects. The syntax of effects is described in the section "Specification of Effects" on page 2488. For any model effect involving classification variables (interactions as well as main effects), the number of levels cannot exceed 32,767. If no independent effects are specified, only an intercept term is fit. You can specify only one MODEL statement (in contrast to the REG procedure, for example, which allows several MODEL statements in the same PROC REG run).

Table 39.5 summarizes options available in the MODEL statement.

Table 39.5 MODEL Statement Options

Task	Options
Produce effect size information	EFFECTSIZE (experimental)
Produce tests for the intercept	INTERCEPT
Omit the intercept parameter from model	NOINT
Produce parameter estimates	SOLUTION
Produce tolerance analysis	TOLERANCE
Suppress univariate tests and output	NOUNI
Display estimable functions	E E1 E2 E3 E4 ALIASING
Control hypothesis tests performed	SS1 SS2 SS3 SS4
Produce confidence intervals	ALPHA= CLI CLM CLPARM
Display predicted and residual values	P
Display intermediate calculations	INVERSE XPX
Tune sensitivity	SINGULAR= ZETA=

The options available in the MODEL statement are described in the following list.

ALIASING

specifies that the estimable functions should be displayed as an *aliasing structure*, for which each row says which linear combination of the parameters is estimated by each estimable function; also, this option adds a column of the same information to the table of parameter estimates, giving for each parameter the expected value of the estimate associated with that parameter. This option is most useful in fractional factorial experiments that can be analyzed without a CLASS statement.

ALPHA=*p*

specifies the level of significance p for $100(1-p)\%$ confidence intervals. By default, p is equal to the value of the ALPHA= option in the PROC GLM statement, or 0.05 if that option is not specified. You can use values between 0 and 1.

CLI

produces confidence limits for individual predicted values for each observation. The CLI option is ignored if the CLM option is also specified.

CLM

produces confidence limits for a mean predicted value for each observation.

CLPARM

produces confidence limits for the parameter estimates (if the SOLUTION option is also specified) and for the results of all ESTIMATE statements.

E

displays the general form of all estimable functions. This is useful for determining the order of parameters when you are writing CONTRAST and ESTIMATE statements.

E1

displays the Type I estimable functions for each effect in the model and computes the corresponding sums of squares.

E2

displays the Type II estimable functions for each effect in the model and computes the corresponding sums of squares.

E3

displays the Type III estimable functions for each effect in the model and computes the corresponding sums of squares.

E4

displays the Type IV estimable functions for each effect in the model and computes the corresponding sums of squares.

EFFECTSIZE

Experimental

adds measures of effect size to each analysis of variance table displayed by the procedure, except for those displayed by the TEST option in the RANDOM statement and by CONTRAST statements with the E= option. The effect size measures include the intraclass correlation and both estimates and confidence intervals for the noncentrality for the F test, the semipartial R^2, and the partial R^2. For more information about the computation and interpretation of

these measures, see the section "Effect Size Measures for *F* Tests in GLM (Experimental)" on page 2502.

INTERCEPT
INT

produces the hypothesis tests associated with the intercept as an effect in the model. By default, the procedure includes the intercept in the model but does not display associated tests of hypotheses. Except for producing the uncorrected total sum of squares instead of the corrected total sum of squares, the INT option is ignored when you use an ABSORB statement.

INVERSE
I

displays the augmented inverse (or generalized inverse) $\mathbf{X}'\mathbf{X}$ matrix:

$$\begin{bmatrix} (\mathbf{X}'\mathbf{X})^{-} & (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'Y \\ Y'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-} & Y'Y - Y'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'Y \end{bmatrix}$$

The upper-left corner is the generalized inverse of $\mathbf{X}'\mathbf{X}$, the upper-right corner is the parameter estimates, and the lower-right corner is the error sum of squares.

NOINT

omits the intercept parameter from the model. The NOINT option is ignored when you use an ABSORB statement.

NOUNI

suppresses the display of univariate statistics. You typically use the NOUNI option with a multivariate or repeated measures analysis of variance when you do not need the standard univariate results. The NOUNI option in a MODEL statement does not affect the univariate output produced by the REPEATED statement.

P

displays observed, predicted, and residual values for each observation that does not contain missing values for independent variables. The Durbin-Watson statistic is also displayed when the P option is specified. The PRESS statistic is also produced if either the CLM or CLI option is specified.

SINGULAR=*number*

tunes the sensitivity of the regression routine to linear dependencies in the design. If a diagonal pivot element is less than $C \times number$ as PROC GLM sweeps the $\mathbf{X}'\mathbf{X}$ matrix, the associated design column is declared to be linearly dependent with previous columns, and the associated parameter is zeroed.

The C value adjusts the check to the relative scale of the variable. The C value is equal to the corrected sum of squares for the variable, unless the corrected sum of squares is 0, in which case C is 1. If you specify the NOINT option but not the ABSORB statement, PROC GLM uses the uncorrected sum of squares instead.

The default value of the SINGULAR= option, 10^{-7}, might be too small, but this value is necessary in order to handle the high-degree polynomials used in the literature to compare regression routines.

SOLUTION

produces a solution to the normal equations (parameter estimates). PROC GLM displays a solution by default when your model involves no classification variables, so you need this option only if you want to see the solution for models with classification effects.

SS1

displays the sum of squares associated with Type I estimable functions for each effect. These are also displayed by default.

SS2

displays the sum of squares associated with Type II estimable functions for each effect.

SS3

displays the sum of squares associated with Type III estimable functions for each effect. These are also displayed by default.

SS4

displays the sum of squares associated with Type IV estimable functions for each effect.

TOLERANCE

displays the tolerances used in the SWEEP routine. The tolerances are of the form C/USS or C/CSS, as described in the discussion of the SINGULAR= option. The tolerance value for the intercept is not divided by its uncorrected sum of squares.

XPX

displays the augmented $\mathbf{X}'\mathbf{X}$ crossproducts matrix:

$$\begin{bmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'Y \\ Y'\mathbf{X} & Y'Y \end{bmatrix}$$

ZETA=*value*

tunes the sensitivity of the check for estimability for Type III and Type IV functions. Any element in the estimable function basis with an absolute value less than the ZETA= option is set to zero. The default value for the ZETA= option is 10^{-8}.

Although it is possible to generate data for which this absolute check can be defeated, the check suffices in most practical examples. Additional research is needed in order to make this check relative rather than absolute.

OUTPUT Statement

OUTPUT < **OUT=***SAS-data-set* > *keyword=names* < *. . . keyword=names* > < *option* > **;**

The OUTPUT statement creates a new SAS data set that saves diagnostic measures calculated after fitting the model. At least one specification of the form *keyword=names* is required.

All the variables in the original data set are included in the new data set, along with variables created in the OUTPUT statement. These new variables contain the values of a variety of diagnostic

measures that are calculated for each observation in the data set. If you want to create a permanent SAS data set, you must specify a two-level name (see *SAS Language Reference: Concepts* for more information about permanent SAS data sets).

Details on the specifications in the OUTPUT statement follow.

keyword=names

specifies the statistics to include in the output data set and provides names to the new variables that contain the statistics. Specify a keyword for each desired statistic (see the following list of keywords), an equal sign, and the variable or variables to contain the statistic.

In the output data set, the first variable listed after a keyword in the OUTPUT statement contains that statistic for the first dependent variable listed in the MODEL statement; the second variable contains the statistic for the second dependent variable in the MODEL statement, and so on. The list of variables following the equal sign can be shorter than the list of dependent variables in the MODEL statement. In this case, the procedure creates the new names in order of the dependent variables in the MODEL statement. See the section "Examples" on page 2479.

The keywords allowed and the statistics they represent are as follows:

COOKD Cook's D influence statistic

COVRATIO standard influence of observation on covariance of parameter estimates

DFFITS standard influence of observation on predicted value

H leverage, $h_i = x_i(\mathbf{X}'\mathbf{X})^{-1}x_i'$

LCL lower bound of a $100(1-p)\%$ confidence interval for an individual prediction. The p-level is equal to the value of the ALPHA= option in the OUTPUT statement or, if this option is not specified, to the ALPHA= option in the PROC GLM statement. If neither of these options is set, then $p = 0.05$ by default, resulting in the lower bound for a 95% confidence interval. The interval also depends on the variance of the error, as well as the variance of the parameter estimates. For the corresponding upper bound, see the UCL keyword.

LCLM lower bound of a $100(1-p)\%$ confidence interval for the expected value (mean) of the predicted value. The p-level is equal to the value of the ALPHA= option in the OUTPUT statement or, if this option is not specified, to the ALPHA= option in the PROC GLM statement. If neither of these options is set, then $p = 0.05$ by default, resulting in the lower bound for a 95% confidence interval. For the corresponding upper bound, see the UCLM keyword.

PREDICTED | P predicted values

PRESS residual for the ith observation that results from dropping it and predicting it on the basis of all other observations. This is the residual divided by $(1-h_i)$, where h_i is the leverage, defined previously.

RESIDUAL | R residuals, calculated as ACTUAL − PREDICTED

RSTUDENT a studentized residual with the current observation deleted

STDI standard error of the individual predicted value

STDP standard error of the mean predicted value

STDR standard error of the residual

STUDENT studentized residuals, the residual divided by its standard error

UCL upper bound of a $100(1-p)\%$ confidence interval for an individual prediction. The p-level is equal to the value of the ALPHA= option in the OUTPUT statement or, if this option is not specified, to the ALPHA= option in the PROC GLM statement. If neither of these options is set, then $p = 0.05$ by default, resulting in the upper bound for a 95% confidence interval. The interval also depends on the variance of the error, as well as the variance of the parameter estimates. For the corresponding lower bound, see the LCL keyword.

UCLM upper bound of a $100(1-p)\%$ confidence interval for the expected value (mean) of the predicted value. The p-level is equal to the value of the ALPHA= option in the OUTPUT statement or, if this option is not specified, to the ALPHA= option in the PROC GLM statement. If neither of these options is set, then $p = 0.05$ by default, resulting in the upper bound for a 95% confidence interval. For the corresponding lower bound, see the LCLM keyword.

OUT=*SAS-data-set*

gives the name of the new data set. By default, the procedure uses the DATA*n* convention to name the new data set.

The following option is available in the OUTPUT statement and is specified after a slash (/):

ALPHA=*p*

specifies the level of significance p for $100(1-p)\%$ confidence intervals. By default, p is equal to the value of the ALPHA= option in the PROC GLM statement or 0.05 if that option is not specified. You can use values between 0 and 1.

See Chapter 4, "Introduction to Regression Procedures," and the section "Influence Statistics" on page 5559 in Chapter 73, "The REG Procedure," for details on the calculation of these statistics.

Examples

The following statements show the syntax for creating an output data set with a single dependent variable.

```
proc glm;
   class a b;
   model y=a b a*b;
   output out=new p=yhat r=resid stdr=eresid;
run;
```

These statements create an output data set named new. In addition to all the variables from the original data set, new contains the variable yhat, with values that are predicted values of the dependent variable y; the variable resid, with values that are the residual values of y; and the variable eresid, with values that are the standard errors of the residuals.

The following statements show a situation with five dependent variables.

```
proc glm;
   by group;
   class a;
   model y1-y5=a x(a);
   output out=pout predicted=py1-py5;
run;
```

The data set pout contains five new variables, py1 through py5. The values of py1 are the predicted values of y1; the values of py2 are the predicted values of y2; and so on.

For more information about the data set produced by the OUTPUT statement, see the section "Output Data Sets" on page 2548.

RANDOM Statement

RANDOM *effects* < / *options* > **;**

When some model effects are random (that is, assumed to be sampled from a normal population of effects), you can specify these effects in the RANDOM statement in order to compute the expected values of mean squares for various model effects and contrasts and, optionally, to perform random effects analysis of variance tests. You can use as many RANDOM statements as you want, provided that they appear after the MODEL statement. If you use a CONTRAST statement with a RANDOM statement and you want to obtain the expected mean squares for the contrast hypothesis, you must enter the CONTRAST statement before the RANDOM statement.

NOTE: PROC GLM uses only the information pertaining to expected mean squares when you specify the TEST option in the RANDOM statement and, even then, only in the extra F tests produced by the RANDOM statement. Other features in the GLM procedure—including the results of the LSMEANS and ESTIMATE statements—assume that all effects are fixed, so that all tests and estimability checks for these statements are based on a fixed-effects model, even when you use a RANDOM statement. Therefore, you should use the MIXED procedure to compute tests involving these features that take the random effects into account; see the section "PROC GLM versus PROC MIXED for Random-Effects Analysis" on page 2540 and Chapter 56, "The MIXED Procedure," for more information.

When you use the RANDOM statement, by default the GLM procedure produces the Type III expected mean squares for model effects and for contrasts specified before the RANDOM statement in the program statements. In order to obtain expected values for other types of mean squares, you need to specify which types of mean squares are of interest in the MODEL statement. See the section "Computing Type I, II, and IV Expected Mean Squares" on page 2542 for more information.

The list of effects in the RANDOM statement should contain one or more of the pure classification

effects specified in the MODEL statement (that is, main effects, crossed effects, or nested effects involving only classification variables). The coefficients corresponding to each effect specified are assumed to be normally and independently distributed with common variance. Levels in different effects are assumed to be independent.

You can specify the following options in the RANDOM statement after a slash (/):

Q

displays all quadratic forms in the fixed effects that appear in the expected mean squares. For some designs, such as large mixed-level factorials, the Q option might generate a substantial amount of output.

TEST

performs hypothesis tests for each effect specified in the model, using appropriate error terms as determined by the expected mean squares.

CAUTION: PROC GLM does not automatically declare interactions to be random when the effects in the interaction are declared random. For example,

```
random a b / test;
```

does not produce the same expected mean squares or tests as

```
random a b a*b / test;
```

To ensure correct tests, you need to list all random interactions and random main effects in the RANDOM statement.

See the section "Random-Effects Analysis" on page 2540 for more information about the calculation of expected mean squares and tests and on the similarities and differences between the GLM and MIXED procedures. See Chapter 5, "Introduction to Analysis of Variance Procedures," and Chapter 56, "The MIXED Procedure," for more information about random effects.

REPEATED Statement

REPEATED *factor-specification* < / *options* > **;**

When values of the dependent variables in the MODEL statement represent repeated measurements on the same experimental unit, the REPEATED statement enables you to test hypotheses about the measurement factors (often called *within-subject factors*) as well as the interactions of within-subject factors with independent variables in the MODEL statement (often called *between-subject factors*). The REPEATED statement provides multivariate and univariate tests as well as hypothesis tests for a variety of single-degree-of-freedom contrasts. There is no limit to the number of within-subject factors that can be specified.

The REPEATED statement is typically used for handling repeated measures designs with one repeated response variable. Usually, the variables on the left-hand side of the equation in the MODEL statement represent one repeated response variable. This does not mean that only one factor can

be listed in the REPEATED statement. For example, one repeated response variable (hemoglobin count) might be measured 12 times (implying variables Y1 to Y12 on the left-hand side of the equal sign in the MODEL statement), with the associated within-subject factors treatment and time (implying two factors listed in the REPEATED statement). See the section "Examples" on page 2485 for an example of how PROC GLM handles this case.

Designs with two or more repeated response variables can, however, be handled with the IDENTITY transformation; see the description of this transformation in the following section, and see Example 39.9 for an example of analyzing a doubly multivariate repeated measures design.

When a REPEATED statement appears, the GLM procedure enters a multivariate mode of handling missing values. If any values for variables corresponding to each combination of the within-subject factors are missing, the observation is excluded from the analysis.

If you use a CONTRAST or TEST statement with a REPEATED statement, you must enter the CONTRAST or TEST statement before the REPEATED statement.

The simplest form of the REPEATED statement requires only a *factor-name*. With two repeated factors, you must specify the *factor-name* and number of levels (*levels*) for each factor. Optionally, you can specify the actual values for the levels (*level-values*), a *transformation* that defines single-degree-of-freedom contrasts, and *options* for additional analyses and output. When you specify more than one within-subject factor, the *factor-names* (and associated level and transformation information) must be separated by a comma in the REPEATED statement.

These terms are described in the following section, "Syntax Details."

Syntax Details

You can specify the following terms in the REPEATED statement.

factor-specification

The *factor-specification* for the REPEATED statement can include any number of individual factor specifications, separated by commas, of the following form:

factor-name levels < (level-values) > < transformation >

where

factor-name names a factor to be associated with the dependent variables. The name should not be the same as any variable name that already exists in the data set being analyzed and should conform to the usual conventions of SAS variable names.

When specifying more than one factor, list the dependent variables in the MODEL statement so that the within-subject factors defined in the REPEATED statement are nested; that is, the first factor defined in the REPEATED statement should be the one with values that change least frequently.

levels gives the number of levels associated with the factor being defined. When there is only one within-subject factor, the number of levels is equal to the number of dependent variables. In this case, *levels* is optional. When more than one within-subject factor is defined, however, *levels* is required, and the product of the number of levels of all the factors must equal the number of dependent variables in the MODEL statement.

(*level-values*) gives values that correspond to levels of a repeated-measures factor. These values are used to label output and as spacings for constructing orthogonal polynomial contrasts if you specify a POLYNOMIAL transformation. The number of values specified must correspond to the number of levels for that factor in the REPEATED statement. Enclose the *level-values* in parentheses.

The following *transformation* keywords define single-degree-of-freedom contrasts for factors specified in the REPEATED statement. Since the number of contrasts generated is always one less than the number of levels of the factor, you have some control over which contrast is omitted from the analysis by which transformation you select. The only exception is the IDENTITY transformation; this transformation is not composed of contrasts and has the same degrees of freedom as the factor has levels. By default, the procedure uses the CONTRAST transformation.

CONTRAST<(*ordinal-reference-level*)> generates contrasts between levels of the factor and a reference level. By default, the procedure uses the last level as the reference level; you can optionally specify a reference level in parentheses after the keyword CONTRAST. The reference level corresponds to the ordinal value of the level rather than the level value specified. For example, to generate contrasts between the first level of a factor and the other levels, use

```
contrast(1)
```

HELMERT generates contrasts between each level of the factor and the mean of subsequent levels.

IDENTITY generates an identity transformation corresponding to the associated factor. This transformation is *not* composed of contrasts; it has n degrees of freedom for an n-level factor, instead of $n-1$. This can be used for doubly multivariate repeated measures.

MEAN<(*ordinal-reference-level*)> generates contrasts between levels of the factor and the mean of all other levels of the factor. Specifying a reference level eliminates the contrast between that level and the mean. Without a reference level, the contrast involving the last level is omitted. See the CONTRAST transformation for an example.

POLYNOMIAL generates orthogonal polynomial contrasts. Level values, if provided, are used as spacings in the construction of the polynomials; otherwise, equal spacing is assumed.

PROFILE generates contrasts between adjacent levels of the factor.

You can specify the following options in the REPEATED statement after a slash (/).

CANONICAL
performs a canonical analysis of the **H** and **E** matrices corresponding to the transformed variables specified in the REPEATED statement.

HTYPE=*n*
specifies the type of the **H** matrix used in the multivariate tests and the type of sums of squares used in the univariate tests. See the HTYPE= option in the specifications for the MANOVA statement for further details.

MEAN
generates the overall arithmetic means of the within-subject variables.

MSTAT=FAPPROX

MSTAT=EXACT
specifies the method of evaluating the test statistics for the multivariate analysis. The default is MSTAT=FAPPROX, which specifies that the multivariate tests are evaluated using the usual approximations based on the F distribution, as discussed in the section "Multivariate Tests" in Chapter 4, "Introduction to Regression Procedures." Alternatively, you can specify MSTAT=EXACT to compute exact p-values for three of the four tests (Wilks' lambda, the Hotelling-Lawley trace, and Roy's greatest root) and an improved F approximation for the fourth (Pillai's trace). While MSTAT=EXACT provides better control of the significance probability for the tests, especially for Roy's greatest root, computations for the exact p-values can be appreciably more demanding, and are in fact infeasible for large problems (many dependent variables). Thus, although MSTAT=EXACT is more accurate for most data, it is not the default method. For more information about the results of MSTAT=EXACT, see the section "Multivariate Analysis of Variance" on page 2531.

NOM
displays only the results of the univariate analyses.

NOU
displays only the results of the multivariate analyses.

PRINTE
displays the **E** matrix for each combination of within-subject factors, as well as partial correlation matrices for both the original dependent variables and the variables defined by the transformations specified in the REPEATED statement. In addition, the PRINTE option provides sphericity tests for each set of transformed variables. If the requested transformations are not orthogonal, the PRINTE option also provides a sphericity test for a set of orthogonal contrasts.

PRINTH
displays the **H** (SSCP) matrix associated with each multivariate test.

PRINTM
displays the transformation matrices that define the contrasts in the analysis. PROC GLM always displays the **M** matrix so that the transformed variables are defined by the rows, not the columns, of the displayed **M** matrix. In other words, PROC GLM actually displays $\mathbf{M}'$.

PRINTRV
displays the characteristic roots and vectors for each multivariate test.

SUMMARY
produces analysis-of-variance tables for each contrast defined by the within-subject factors. Along with tests for the effects of the independent variables specified in the MODEL statement, a term labeled MEAN tests the hypothesis that the overall mean of the contrast is zero.

Examples

When specifying more than one factor, list the dependent variables in the MODEL statement so that the within-subject factors defined in the REPEATED statement are nested; that is, the first factor defined in the REPEATED statement should be the one with values that change least frequently. For example, assume that three treatments are administered at each of four times, for a total of twelve dependent variables on each experimental unit. If the variables are listed in the MODEL statement as Y1 through Y12, then the REPEATED statement in

```
proc glm;
    classes group;
    model Y1-Y12=group / nouni;
    repeated trt 3, time 4;
run;
```

implies the following structure:

	Dependent Variables											
	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8	Y9	Y10	Y11	Y12
Value of trt	1	1	1	1	2	2	2	2	3	3	3	3
Value of time	1	2	3	4	1	2	3	4	1	2	3	4

The REPEATED statement always produces a table like the preceding one. For more information, see the section "Repeated Measures Analysis of Variance" on page 2532.

TEST Statement

TEST < **H=***effects* > **E=***effect* < / *options* > **;**

By default, for each sum of squares in the analysis an F value is computed that uses the residual MS as an error term. Use a TEST statement to request additional F tests that use other effects as error terms. You need a TEST statement when a nonstandard error structure (as in a split-plot design) exists. Note, however, that this might not be appropriate if the design is unbalanced, since in most unbalanced designs with nonstandard error structures, mean squares are not necessarily independent with equal expectations under the null hypothesis.

CAUTION: The GLM procedure does not check any of the assumptions underlying the F statistic. When you specify a TEST statement, you assume sole responsibility for the validity of the F statistic

produced. To help validate a test, you can use the RANDOM statement and inspect the expected mean squares, or you can use the TEST option of the RANDOM statement.

You can use as many TEST statements as you want, provided that they appear after the MODEL statement.

You can specify the following terms in the TEST statement.

H=*effects* specifies which effects in the preceding model are to be used as hypothesis (numerator) effects.

E=*effect* specifies one, and only one, effect to use as the error (denominator) term. The E= specification is required.

By default, the sum of squares type for all hypothesis sum of squares and error sum of squares is the highest type computed in the model. If the hypothesis type or error type is to be another type that was computed in the model, you should specify one or both of the following options after a slash (/).

ETYPE=*n*
specifies the type of sum of squares to use for the error term. The type must be a type computed in the model (*n*=1, 2, 3, or 4).

HTYPE=*n*
specifies the type of sum of squares to use for the hypothesis. The type must be a type computed in the model (*n*=1, 2, 3, or 4).

This example illustrates the TEST statement with a split-plot model:

```
proc glm;
   class a b c;
   model y=a  b(a) c a*c b*c(a);
   test h=a e=b(a)/ htype=1 etype=1;
   test h=c a*c e=b*c(a) / htype=1 etype=1;
run;
```

WEIGHT Statement

WEIGHT *variable* **;**

When a WEIGHT statement is used, a weighted residual sum of squares

$$\sum_i w_i (y_i - \hat{y}_i)^2$$

is minimized, where w_i is the value of the variable specified in the WEIGHT statement, y_i is the observed value of the response variable, and $\hat{y}_i$ is the predicted value of the response variable.

If you specify the WEIGHT statement, it must appear before the first RUN statement or it is ignored.

An observation is used in the analysis only if the value of the WEIGHT statement variable is nonmissing and greater than zero.

The WEIGHT statement has no effect on degrees of freedom or number of observations, but it is used by the MEANS statement when calculating means and performing multiple comparison tests (as described in the section "MEANS Statement" on page 2468).

The normal equations used when a WEIGHT statement is present are

$$\mathbf{X}'\mathbf{W}\mathbf{X}\boldsymbol{\beta} = \mathbf{X}'\mathbf{W}\mathbf{Y}$$

where $\mathbf{W}$ is a diagonal matrix consisting of the values of the variable specified in the WEIGHT statement.

If the weights for the observations are proportional to the reciprocals of the error variances, then the weighted least squares estimates are best linear unbiased estimators (BLUE).

Details: GLM Procedure

Statistical Assumptions for Using PROC GLM

The basic statistical assumption underlying the least squares approach to general linear modeling is that the observed values of each dependent variable can be written as the sum of two parts: a fixed component $x'\beta$, which is a linear function of the independent coefficients, and a random noise, or error, component ϵ:

$$y = x'\boldsymbol{\beta} + \epsilon$$

The independent coefficients x are constructed from the model effects as described in the section "Parameterization of PROC GLM Models" on page 2491. Further, the errors for different observations are assumed to be uncorrelated with identical variances. Thus, this model can be written

$$E(Y) = \mathbf{X}\beta, \qquad \mathrm{Var}(Y) = \sigma^2 I$$

where Y is the vector of dependent variable values, $\mathbf{X}$ is the matrix of independent coefficients, I is the identity matrix, and σ^2 is the common variance for the errors. For multiple dependent variables, the model is similar except that the errors for different dependent variables within the same observation are not assumed to be uncorrelated. This yields a multivariate linear model of the form

$$E(Y) = \mathbf{X}B, \qquad \mathrm{Var}(\mathrm{vec}(Y)) = \Sigma \otimes I$$

where Y and B are now matrices, with one column for each dependent variable, $\mathrm{vec}(Y)$ strings Y out by rows, and $\otimes$ indicates the Kronecker matrix product.

Under the assumptions thus far discussed, the least squares approach provides estimates of the linear parameters that are unbiased and have minimum variance among linear estimators. Under the further assumption that the errors have a normal (or Gaussian) distribution, the least squares estimates are the maximum likelihood estimates and their distribution is known. All of the significance levels ("*p* values") and confidence limits calculated by the GLM procedure require this assumption of normality in order to be exactly valid, although they are good approximations in many other cases.

Specification of Effects

Each term in a model, called an *effect*, is a variable or combination of variables. Effects are specified with a special notation that uses variable names and operators. There are two kinds of variables: *classification* (or *CLASS*) *variables* and *continuous variables*. There are two primary operators: *crossing* and *nesting*. A third operator, the *bar operator*, is used to simplify effect specification.

In an analysis-of-variance model, independent variables must be variables that identify classification levels. In the SAS System, these are called *classification* (or *class*) *variables* and are declared in the CLASS statement. (They can also be called *categorical*, *qualitative*, *discrete*, or *nominal variables*.) Classification variables can be either *numeric* or *character*. The values of a classification variable are called *levels*. For example, the classification variable Sex has the levels "male" and "female."

In a model, an independent variable that is not declared in the CLASS statement is assumed to be continuous. Continuous variables, which must be numeric, are used for response variables and covariates. For example, the heights and weights of subjects are continuous variables.

Types of Effects

There are seven different types of effects used in the GLM procedure. In the following list, assume that A, B, C, D, and E are CLASS variables and that X1, X2, and Y are continuous variables:

- Regressor effects are specified by writing continuous variables by themselves: X1 X2.
- Polynomial effects are specified by joining two or more continuous variables with asterisks: X1*X1 X1*X2.
- Main effects are specified by writing CLASS variables by themselves: A B C.
- Crossed effects (interactions) are specified by joining classification variables with asterisks: A*B B*C A*B*C.
- Nested effects are specified by following a main effect or crossed effect with a classification variable or list of classification variables enclosed in parentheses. The main effect or crossed effect is nested within the effects listed in parentheses: B(A) C(B*A) D*E(C*B*A). In this example, B(A) is read "B nested within A."
- Continuous-by-class effects are written by joining continuous variables and classification variables with asterisks: X1*A.

- Continuous-nesting-class effects consist of continuous variables followed by a classification variable interaction enclosed in parentheses: `X1(A)  X1*X2(A*B)`.

One example of the general form of an effect involving several variables is

```
X1*X2*A*B*C(D*E)
```

This example contains crossed continuous terms by crossed classification terms nested within more than one classification variable. The continuous list comes first, followed by the crossed list, followed by the nesting list in parentheses. Note that asterisks can appear within the nested list but not immediately before the left parenthesis. For details on how the design matrix and parameters are defined with respect to the effects specified in this section, see the section "Parameterization of PROC GLM Models" on page 2491.

The MODEL statement and several other statements use these effects. Some examples of MODEL statements that use various kinds of effects are shown in the following table; `a`, `b`, and `c` represent classification variables, and `y`, `y1`, `y2`, `x`, and `z` represent continuous variables.

Specification	Type of Model
`model y=x;`	simple regression
`model y=x z;`	multiple regression
`model y=x x*x;`	polynomial regression
`model y1 y2=x z;`	multivariate regression
`model y=a;`	one-way ANOVA
`model y=a b c;`	main-effects ANOVA
`model y=a b a*b;`	factorial ANOVA with interaction
`model y=a b(a) c(b a);`	nested ANOVA
`model y1 y2=a b;`	multivariate analysis of variance (MANOVA)
`model y=a x;`	analysis of covariance
`model y=a x(a);`	separate-slopes regression
`model y=a x x*a;`	homogeneity-of-slopes regression

The Bar Operator

You can shorten the specification of a large factorial model by using the bar operator. For example, two ways of writing the model for a full three-way factorial model follow:

```
model Y = A B C   A*B A*C B*C   A*B*C;

model Y = A|B|C;
```

When the bar (|) is used, the right and left sides become effects, and the cross of them becomes an effect. Multiple bars are permitted. The expressions are expanded from left to right, using rules 2–4 given in Searle (1971, p. 390).

- Multiple bars are evaluated from left to right. For instance, A|B|C is evaluated as follows:

$$\begin{aligned} \text{A | B | C} &\rightarrow \{\text{ A | B }\}\text{ | C} \\ &\rightarrow \{\text{ A B A*B }\}\text{ | C} \\ &\rightarrow \text{A B A*B C A*C B*C A*B*C} \end{aligned}$$

- Crossed and nested groups of variables are combined. For example, A(B) | C(D) generates A*C(B D), among other terms.
- Duplicate variables are removed. For example, A(C) | B(C) generates A*B(C C), among other terms, and the extra C is removed.
- Effects are discarded if a variable occurs on both the crossed and nested parts of an effect. For instance, A(B) | B(D E) generates A*B(B D E), but this effect is eliminated immediately.

You can also specify the maximum number of variables involved in any effect that results from bar evaluation by specifying that maximum number, preceded by an @ sign, at the end of the bar effect. For example, the specification A | B | C@2 would result in only those effects that contain 2 or fewer variables: in this case, A B A*B C A*C and B*C.

More examples of using the bar and at operators follow:

A | C(B)	is equivalent to	A C(B) A*C(B)
A(B) | C(B)	is equivalent to	A(B) C(B) A*C(B)
A(B) | B(D E)	is equivalent to	A(B) B(D E)
A | B(A) | C	is equivalent to	A B(A) C A*C B*C(A)
A | B(A) | C@2	is equivalent to	A B(A) C A*C
A | B | C | D@2	is equivalent to	A B A*B C A*C B*C D A*D B*D C*D
A*B(C*D)	is equivalent to	A*B(C D)

Using PROC GLM Interactively

You can use the GLM procedure interactively. After you specify a model with a MODEL statement and run PROC GLM with a RUN statement, you can execute a variety of statements without reinvoking PROC GLM.

The section "Syntax: GLM Procedure" on page 2444 describes which statements can be used interactively. These interactive statements can be executed singly or in groups by following the single statement or group of statements with a RUN statement. Note that the MODEL statement cannot be repeated; PROC GLM allows only one MODEL statement.

If you use PROC GLM interactively, you can end the GLM procedure with a DATA step, another PROC step, an ENDSAS statement, or a QUIT statement.

When you are using PROC GLM interactively, additional RUN statements do not end the procedure but tell PROC GLM to execute additional statements.

When you specify a WHERE statement with PROC GLM, it should appear before the first RUN statement. The WHERE statement enables you to select only certain observations for analysis without using a subsetting DATA step. For example, `where group ne 5` omits observations with GROUP=5 from the analysis. See *SAS Language Reference: Dictionary* for details on this statement.

When you specify a BY statement with PROC GLM, interactive processing is not possible; that is, once the first RUN statement is encountered, processing proceeds for each BY group in the data set, and no further statements are accepted by the procedure.

Interactivity is also disabled when there are different patterns of missing values among the dependent variables. For details, see the section "Missing Values" on page 2543.

Parameterization of PROC GLM Models

The GLM procedure constructs a linear model according to the specifications in the MODEL statement. Each effect generates one or more columns in a design matrix **X**. This section shows precisely how **X** is built.

Intercept

All models include a column of 1s by default to estimate an intercept parameter μ. You can use the NOINT option to suppress the intercept.

Regression Effects

Regression effects (covariates) have the values of the variables copied into the design matrix directly. Polynomial terms are multiplied out and then installed in **X**.

Main Effects

If a classification variable has m levels, PROC GLM generates m columns in the design matrix for its main effect. Each column is an indicator variable for one of the levels of the classification variable. The default order of the columns is the sort order of the values of their levels; this order can be controlled with the ORDER= option in the PROC GLM statement, as shown in the following table.

Data		Design Matrix					
			A		B		
A	B	μ	A1	A2	B1	B2	B3
1	1	1	1	0	1	0	0
1	2	1	1	0	0	1	0
1	3	1	1	0	0	0	1
2	1	1	0	1	1	0	0
2	2	1	0	1	0	1	0
2	3	1	0	1	0	0	1

There are more columns for these effects than there are degrees of freedom for them; in other words, PROC GLM is using an over-parameterized model.

Crossed Effects

First, PROC GLM reorders the terms to correspond to the order of the variables in the CLASS statement; thus, B*A becomes A*B if A precedes B in the CLASS statement. Then, PROC GLM generates columns for all combinations of levels that occur in the data. The order of the columns is such that the rightmost variables in the cross index faster than the leftmost variables. No columns are generated corresponding to combinations of levels that do not occur in the data.

Data		Design Matrix											
			A		B			A*B					
A	B	μ	A1	A2	B1	B2	B3	A1B1	A1B2	A1B3	A2B1	A2B2	A2B3
1	1	1	1	0	1	0	0	1	0	0	0	0	0
1	2	1	1	0	0	1	0	0	1	0	0	0	0
1	3	1	1	0	0	0	1	0	0	1	0	0	0
2	1	1	0	1	1	0	0	0	0	0	1	0	0
2	2	1	0	1	0	1	0	0	0	0	0	1	0
2	3	1	0	1	0	0	1	0	0	0	0	0	1

In this matrix, main-effects columns are not linearly independent of crossed-effect columns; in fact, the column space for the crossed effects contains the space of the main effect.

Nested Effects

Nested effects are generated in the same manner as crossed effects. Hence, the design columns generated by the following statements are the same (but the ordering of the columns is different):

```
model y=a b(a);
```
(B nested within A)

```
model y=a a*b;
```
(omitted main effect for B)

The nesting operator in PROC GLM is more a notational convenience than an operation distinct from crossing. Nested effects are characterized by the property that the nested variables never

appear as main effects. The order of the variables within nesting parentheses is made to correspond to the order of these variables in the CLASS statement. The order of the columns is such that variables outside the parentheses index faster than those inside the parentheses, and the rightmost nested variables index faster than the leftmost variables.

Data		**Design Matrix**								
			A		B(A)					
A	B	μ	A1	A2	B1A1	B2A1	B3A1	B1A2	B2A2	B3A2
1	1	1	1	0	1	0	0	0	0	0
1	2	1	1	0	0	1	0	0	0	0
1	3	1	1	0	0	0	1	0	0	0
2	1	1	0	1	0	0	0	1	0	0
2	2	1	0	1	0	0	0	0	1	0
2	3	1	0	1	0	0	0	0	0	1

Continuous-Nesting-Class Effects

When a continuous variable nests with a classification variable, the design columns are constructed by multiplying the continuous values into the design columns for the class effect.

Data		**Design Matrix**				
			A		X(A)	
X	A	μ	A1	A2	X(A1)	X(A2)
21	1	1	1	0	21	0
24	1	1	1	0	24	0
22	1	1	1	0	22	0
28	2	1	0	1	0	28
19	2	1	0	1	0	19
23	2	1	0	1	0	23

This model estimates a separate slope for X within each level of A.

Continuous-by-Class Effects

Continuous-by-class effects generate the same design columns as continuous-nesting-class effects. The two models differ by the presence of the continuous variable as a regressor by itself, in addition to being a contributor to X*A.

Data		**Design Matrix**					
				A		X*A	
X	A	μ	X	A1	A2	X*A1	X*A2
21	1	1	21	1	0	21	0
24	1	1	24	1	0	24	0
22	1	1	22	1	0	22	0
28	2	1	28	0	1	0	28
19	2	1	19	0	1	0	19
23	2	1	23	0	1	0	23

Continuous-by-class effects are used to test the homogeneity of slopes. If the continuous-by-class effect is nonsignificant, the effect can be removed so that the response with respect to X is the same for all levels of the classification variables.

General Effects

An example that combines all the effects is

X1*X2*A*B*C(D E)

The continuous list comes first, followed by the crossed list, followed by the nested list in parentheses.

The sequencing of parameters is important to learn if you use the CONTRAST or ESTIMATE statement to compute or test some linear function of the parameter estimates.

Effects might be retitled by PROC GLM to correspond to ordering rules. For example, B*A(E D) might be retitled A*B(D E) to satisfy the following:

- Classification variables that occur outside parentheses (crossed effects) are sorted in the order in which they appear in the CLASS statement.
- Variables within parentheses (nested effects) are sorted in the order in which they appear in a CLASS statement.

The sequencing of the parameters generated by an effect can be described by which variables have their levels indexed faster:

- Variables in the crossed part index faster than variables in the nested list.
- Within a crossed or nested list, variables to the right index faster than variables to the left.

For example, suppose a model includes four effects—A, B, C, and D—each having two levels, 1 and 2. If the CLASS statement is

```
class A B C D;
```

then the order of the parameters for the effect B*A(C D), which is retitled A*B(C D), is as follows.

$$
\begin{array}{l}
A_1B_1C_1D_1 \\
A_1B_2C_1D_1 \\
A_2B_1C_1D_1 \\
A_2B_2C_1D_1 \\
A_1B_1C_1D_2 \\
A_1B_2C_1D_2 \\
A_2B_1C_1D_2
\end{array}
$$

$$A_2B_2C_1D_2$$
$$A_1B_1C_2D_1$$
$$A_1B_2C_2D_1$$
$$A_2B_1C_2D_1$$
$$A_2B_2C_2D_1$$
$$A_1B_1C_2D_2$$
$$A_1B_2C_2D_2$$
$$A_2B_1C_2D_2$$
$$A_2B_2C_2D_2$$

Note that first the crossed effects B and A are sorted in the order in which they appear in the CLASS statement so that A precedes B in the parameter list. Then, for each combination of the nested effects in turn, combinations of A and B appear. The B effect changes fastest because it is rightmost in the (renamed) cross list. Then A changes next fastest. The D effect changes next fastest, and C is the slowest since it is leftmost in the nested list.

When numeric classification variables are used, their levels are sorted by their character format, which might not correspond to their numeric sort sequence. Therefore, it is advisable to include a format for numeric classification variables or to use the ORDER=INTERNAL option in the PROC GLM statement to ensure that levels are sorted by their internal values.

Degrees of Freedom

For models with classification (categorical) effects, there are more design columns constructed than there are degrees of freedom for the effect. Thus, there are linear dependencies among the columns. In this event, the parameters are not jointly estimable; there is an infinite number of least squares solutions. The GLM procedure uses a generalized g_2-inverse to obtain values for the estimates; see the section "Computational Method" on page 2547 for more details. The solution values are not produced unless the SOLUTION option is specified in the MODEL statement. The solution has the characteristic that estimates are zero whenever the design column for that parameter is a linear combination of previous columns. (Strictly termed, the solution values should not be called estimates, since the parameters might not be formally estimable.) With this full parameterization, hypothesis tests are constructed to test linear functions of the parameters that are estimable.

Other procedures (such as the CATMOD procedure) reparameterize models to full rank by using certain restrictions on the parameters. PROC GLM does not reparameterize, making the hypotheses that are commonly tested more understandable. See Goodnight (1978a) for additional reasons for not reparameterizing.

PROC GLM does not actually construct the entire design matrix $\mathbf{X}$; rather, a row x_i of $\mathbf{X}$ is constructed for each observation in the data set and used to accumulate the crossproduct matrix $\mathbf{X}'\mathbf{X} = \sum_i x_i' x_i$.

Hypothesis Testing in PROC GLM

See Chapter 15, "The Four Types of Estimable Functions," for a complete discussion of the four standard types of hypothesis tests.

Example

To illustrate the four types of tests and the principles upon which they are based, consider a two-way design with interaction based on the following data:

		B	
		1	2
A	1	23.5 23.7	28.7
	2	8.9	5.6 8.9
	3	10.3 12.5	13.6 14.6

Invoke PROC GLM and specify all the estimable functions options to examine what the GLM procedure can test. The following statements produce the summary ANOVA table displayed in Figure 39.10.

```
data example;
   input a b y @@;
   datalines;
1 1 23.5  1 1 23.7  1 2 28.7  2 1  8.9  2 2  5.6
2 2  8.9  3 1 10.3  3 1 12.5  3 2 13.6  3 2 14.6
;

proc glm;
   class a b;
   model y=a b a*b / e e1 e2 e3 e4;
run;
```

Figure 39.10 Summary ANOVA Table from PROC GLM

```
                              The GLM Procedure

Dependent Variable: y

                                        Sum of
 Source                     DF         Squares    Mean Square   F Value   Pr > F

 Model                       5     520.4760000    104.0952000     49.66   0.0011

 Error                       4       8.3850000      2.0962500

 Corrected Total             9     528.8610000

               R-Square     Coeff Var      Root MSE        y Mean

               0.984145      9.633022      1.447843      15.03000
```

The following sections show the general form of estimable functions and discuss the four standard tests, their properties, and abbreviated output for the two-way crossed example.

Estimability

Figure 39.11 is the general form of estimable functions for the example. In order to be testable, a hypothesis must be able to fit within the framework displayed here.

Figure 39.11 General Form of Estimable Functions

```
                              The GLM Procedure

                     General Form of Estimable Functions

                     Effect            Coefficients

                     Intercept         L1

                     a         1       L2
                     a         2       L3
                     a         3       L1-L2-L3

                     b         1       L5
                     b         2       L1-L5

                     a*b       1 1     L7
                     a*b       1 2     L2-L7
                     a*b       2 1     L9
                     a*b       2 2     L3-L9
                     a*b       3 1     L5-L7-L9
                     a*b       3 2     L1-L2-L3-L5+L7+L9
```

If a hypothesis is estimable, the Ls in the preceding scheme can be set to values that match the hypothesis. All the standard tests in PROC GLM can be shown in the preceding format, with some of the Ls zeroed and some set to functions of other Ls.

The following sections show how many of the hypotheses can be tested by comparing the model sum-of-squares regression from one model to a submodel. The notation used is

$$\mathrm{SS}(B \textit{ effects} | A \textit{ effects}) = \mathrm{SS}(B \textit{ effects}, A \textit{ effects}) - \mathrm{SS}(A \textit{ effects})$$

where SS(*A effects*) denotes the regression model sum of squares for the model consisting of *A effects*. This notation is equivalent to the reduction notation defined by Searle (1971) and summarized in Chapter 15, "The Four Types of Estimable Functions."

Type I Tests

Type I sums of squares (SS), also called *sequential sums of squares*, are the incremental improvement in error sums of squares as each effect is added to the model. They can be computed by fitting the model in steps and recording the difference in error sum of squares at each step.

Source	Type I SS
A	$\mathrm{SS}(A \mid \mu)$
B	$\mathrm{SS}(B \mid \mu, A)$
$A * B$	$\mathrm{SS}(A * B \mid \mu, A, B)$

Type I sums of squares are displayed by default because they are easy to obtain and can be used in various hand calculations to produce sum of squares values for a series of different models. Nelder (1994) and others have argued that Type I and II sums are essentially the only appropriate ones for testing ANOVA effects; however, see also the discussion of Nelder's article, especially Rodriguez, Tobias, and Wolfinger (1995) and Searle (1995).

The Type I hypotheses have these properties:

- Type I sum of squares for all effects add up to the model sum of squares. None of the other sum of squares types have this property, except in special cases.
- Type I hypotheses can be derived from rows of the Forward-Dolittle transformation of $\mathbf{X}'\mathbf{X}$ (a transformation that reduces $\mathbf{X}'\mathbf{X}$ to an upper triangular matrix by row operations).
- Type I sum of squares are statistically independent of each other under the usual assumption that the true residual errors are independent and identically normally distributed (see page 2487).
- Type I hypotheses depend on the order in which effects are specified in the MODEL statement.

- Type I hypotheses are uncontaminated by parameters corresponding to effects that precede the effect being tested; however, the hypotheses usually involve parameters for effects following the tested effect in the model. For example, in the model

```
Y=A B;
```

  the Type I hypothesis for B does not involve A parameters, but the Type I hypothesis for A does involve B parameters.

- Type I hypotheses are functions of the cell counts for unbalanced data; the hypotheses are not usually the same hypotheses that are tested if the data are balanced.

- Type I sums of squares are useful for polynomial models where you want to know the contribution of a term as though it had been made orthogonal to preceding effects. Thus, in polynomial models, Type I sums of squares correspond to tests of the orthogonal polynomial effects.

The Type I estimable functions and associated tests for the example are shown in Figure 39.12.

Figure 39.12 Type I Estimable Functions and Tests

Type I Estimable Functions

Effect		Coefficients: a	Coefficients: b	Coefficients: a*b
Intercept		0	0	0
a	1	L2	0	0
a	2	L3	0	0
a	3	-L2-L3	0	0
b	1	0.1667*L2-0.1667*L3	L5	0
b	2	-0.1667*L2+0.1667*L3	-L5	0
a*b	1 1	0.6667*L2	0.2857*L5	L7
a*b	1 2	0.3333*L2	-0.2857*L5	-L7
a*b	2 1	0.3333*L3	0.2857*L5	L9
a*b	2 2	0.6667*L3	-0.2857*L5	-L9
a*b	3 1	-0.5*L2-0.5*L3	0.4286*L5	-L7-L9
a*b	3 2	-0.5*L2-0.5*L3	-0.4286*L5	L7+L9

Source	DF	Type I SS	Mean Square	F Value	Pr > F
a	2	494.0310000	247.0155000	117.84	0.0003
b	1	10.7142857	10.7142857	5.11	0.0866
a*b	2	15.7307143	7.8653571	3.75	0.1209

Type II Tests

The Type II tests can also be calculated by comparing the error sums of squares (SS) for subset models. The Type II SS are the reduction in error SS due to adding the term after all other terms

have been added to the model except terms that contain the effect being tested. An effect is contained in another effect if it can be derived by deleting variables from the latter effect. For example, A and B are both contained in A*B. For this model, the Type II SS are given by the reduced sums of squares as shown in the following table.

Source	Type II SS
A	$\mathrm{SS}(A \mid \mu, B)$
B	$\mathrm{SS}(B \mid \mu, A)$
$A * B$	$\mathrm{SS}(A * B \mid \mu, A, B)$

Type II SS have these properties:

- Type II SS do not necessarily sum to the model SS.
- The hypothesis for an effect does not involve parameters of other effects except for containing effects (which it must involve to be estimable).
- Type II SS are invariant to the ordering of effects in the model.
- For unbalanced designs, Type II hypotheses for effects that are contained in other effects are not usually the same hypotheses that are tested if the data are balanced. The hypotheses are generally functions of the cell counts.

The Type II estimable functions and associated tests for the example are shown in Figure 39.13.

Figure 39.13 Type II Estimable Functions and Tests

Type II Estimable Functions

		Coefficients		
Effect		a	b	a*b
Intercept		0	0	0
a	1	L2	0	0
a	2	L3	0	0
a	3	-L2-L3	0	0
b	1	0	L5	0
b	2	0	-L5	0
a*b	1 1	0.619*L2+0.0476*L3	0.2857*L5	L7
a*b	1 2	0.381*L2-0.0476*L3	-0.2857*L5	-L7
a*b	2 1	-0.0476*L2+0.381*L3	0.2857*L5	L9
a*b	2 2	0.0476*L2+0.619*L3	-0.2857*L5	-L9
a*b	3 1	-0.5714*L2-0.4286*L3	0.4286*L5	-L7-L9
a*b	3 2	-0.4286*L2-0.5714*L3	-0.4286*L5	L7+L9

Source	DF	Type II SS	Mean Square	F Value	Pr > F
a	2	499.1202857	249.5601429	119.05	0.0003
b	1	10.7142857	10.7142857	5.11	0.0866
a*b	2	15.7307143	7.8653571	3.75	0.1209

Type III and Type IV Tests

Type III and Type IV sums of squares (SS), sometimes referred to as *partial sums of squares*, are considered by many to be the most desirable; see Searle (1987, Section 4.6). Using PROC GLM's singular parameterization, these SS cannot, in general, be computed by comparing model SS from different models. However, they can sometimes be computed by reduction for methods that reparameterize to full rank, when such a reparameterization effectively imposes Type III linear constraints on the parameters. In PROC GLM, they are computed by constructing a hypothesis matrix $\mathbf{L}$ and then computing the SS associated with the hypothesis $\mathbf{L}\boldsymbol{\beta} = 0$. As long as there are no missing cells in the design, Type III and Type IV SS are the same.

These are properties of Type III and Type IV SS:

- The hypothesis for an effect does not involve parameters of other effects except for containing effects (which it must involve to be estimable).
- The hypotheses to be tested are invariant to the ordering of effects in the model.
- The hypotheses are the same hypotheses that are tested if there are no missing cells. They are not functions of cell counts.
- The SS do not generally add up to the model SS and, in some cases, can exceed the model SS.

The SS are constructed from the general form of estimable functions. Type III and Type IV tests are different only if the design has missing cells. In this case, the Type III tests have an orthogonality property, while the Type IV tests have a balancing property. These properties are discussed in Chapter 15, "The Four Types of Estimable Functions." For this example, since the data contain observations for all pairs of levels of A and B, Type IV tests are identical to the Type III tests that are shown in Figure 39.14. (This combines tables from several pages of output.)

Figure 39.14 Type III Estimable Functions and Tests

Type III Estimable Functions

Effect		Coefficients: a	Coefficients: b	Coefficients: a*b
Intercept		0	0	0
a	1	L2	0	0
a	2	L3	0	0
a	3	-L2-L3	0	0
b	1	0	L5	0
b	2	0	-L5	0
a*b	1 1	0.5*L2	0.3333*L5	L7
a*b	1 2	0.5*L2	-0.3333*L5	-L7
a*b	2 1	0.5*L3	0.3333*L5	L9
a*b	2 2	0.5*L3	-0.3333*L5	-L9
a*b	3 1	-0.5*L2-0.5*L3	0.3333*L5	-L7-L9
a*b	3 2	-0.5*L2-0.5*L3	-0.3333*L5	L7+L9

Source	DF	Type III SS	Mean Square	F Value	Pr > F
a	2	479.1078571	239.5539286	114.28	0.0003
b	1	9.4556250	9.4556250	4.51	0.1009
a*b	2	15.7307143	7.8653571	3.75	0.1209

Effect Size Measures for *F* Tests in GLM (Experimental)

A significant F test in a linear model indicates that the effect of the term or contrast being tested might be real. The next thing you want to know is, How big is the effect? Various measures have been devised to give answers to this question that are comparable over different experimental designs. If you specify the experimental EFFECTSIZE option in the MODEL statement, then GLM adds to each ANOVA table estimates and confidence intervals for three different measures of effect size:

- the noncentrality parameter for the F test
- the proportion of total variation accounted for (also known as the semipartial correlation ratio or the squared semipartial correlation)
- the proportion of partial variation accounted for (also known as the full partial correlation ratio or the squared full partial correlation)

The adjectives "semipartial" and "full partial" might seem strange. They refer to how other effects are "partialed out" of the dependent variable and the effect being tested. For "semipartial" statistics, all other effects are partialed out of the effect in question, but not the dependent variable. This measures the (adjusted) effect as a proportion of the total variation in the dependent variable. On

the other hand, for "full partial" statistics, all other effects are partialed out of *both* the dependent variable and the effect in question. This measures the (adjusted) effect as a proportion of only the dependent variation remaining after partialing, or in other words the partial variation. Details about the computation and interpretation of these estimates and confidence intervals are discussed in the remainder of this section.

The noncentrality parameter is directly related to the true distribution of the F statistic when the effect being tested has a non-null effect. The uniformly minimum variance unbiased estimate for the noncentrality is

$$\mathrm{NC}_{\mathrm{UMVUE}} = \frac{\mathrm{DF}(\mathrm{DFE}-2)\mathrm{FValue}}{\mathrm{DFE}} - \mathrm{DF}$$

where FValue is the observed value of the F statistic for the test and DF and DFE are the numerator and denominator degrees of freedom for the test, respectively. An alternative estimate that can be slightly biased but has a somewhat lower expected mean square error is

$$\mathrm{NC}_{\mathrm{minMSE}} = \frac{\mathrm{DF}(\mathrm{DFE}-4)\mathrm{FValue}}{\mathrm{DFE}} - \frac{\mathrm{DF}(\mathrm{DFE}-4)}{\mathrm{DFE}-2}$$

(See Perlman and Rasmussen (1975), cited in Johnson, Kotz, and Balakrishnan (1994).) A $p \times 100\%$ lower confidence bound for the noncentrality is given by the value of NC for which probf(FValue,DF,DFE,NC) = p, where probf() is the cumulative probability function for the noncentral F distribution. This result can be used to form a $(1-\alpha) \times 100\%$ confidence interval for the noncentrality.

The partial proportion of variation accounted for by the effect being tested is easiest to define by its natural sample estimate,

$$\hat{\eta}^2_{partial} = \frac{\mathrm{SS}}{\mathrm{SS}+\mathrm{SSE}}$$

where SSE is the sample error sum of squares. Note that $\hat{\eta}^2_{partial}$ is actually sometimes denoted $R^2_{partial}$ or just R^2, but in this context the R^2 notation is reserved for the $\hat{\eta}^2$ corresponding to the overall model, which is just the familiar R^2 for the model. $\hat{\eta}^2_{partial}$ is actually a biased estimate of the true $\eta^2_{partial}$; an alternative that is approximately unbiased is given by

$$\omega^2_{partial} = \frac{\mathrm{SS} - DF \times MSE}{\mathrm{SS} + (N - \mathrm{DF})MSE}$$

where MSE = SSE/DFE is the sample mean square for error and N is the number of observations. The true $\eta^2_{partial}$ is related to the true noncentrality parameter NC by the formula

$$\eta^2_{partial} = \frac{\mathrm{NC}}{\mathrm{NC}+N}$$

This fact can be employed to transform a confidence interval for NC into one for $\eta^2_{partial}$. Note that some authors (Steiger and Fouladi 1997; Fidler and Thompson 2001; Smithson 2003) have

published slightly different confidence intervals for $\eta^2_{partial}$, based on a slightly different formula for the relationship between $\eta^2_{partial}$ and NC, apparently due to Cohen (1988). Cohen's formula appears to be approximately correct for random predictor values (Maxwell 2000), but the one given previously is correct if the predictor values are assumed fixed, as is standard for the GLM procedure.

Finally, the proportion of total variation accounted for by the effect being tested is again easiest to define by its natural sample estimate, which is known as the *(semipartial)* $\hat{\eta}^2$ statistic,

$$\hat{\eta}^2 = \frac{\mathrm{SS}}{\mathrm{SS}_{total}}$$

where SS_{total} is the total sample (corrected) sum of squares, and SS is the observed sum of squares due to the effect being tested. As with $\hat{\eta}^2_{partial}$, $\hat{\eta}^2$ is actually a biased estimate of the true η^2; an alternative that is approximately unbiased is the *(semipartial)* ω^2 statistic

$$\omega^2 = \frac{\mathrm{SS} - \mathrm{DF} \times \mathrm{MSE}}{\mathrm{SS}_{total} + \mathrm{MSE}}$$

where MSE = SSE/DFE is the sample mean square for error. Whereas $\eta^2_{partial}$ depends only on the noncentrality for its associated F test, the presence of the total sum of squares in the previous formulas indicates that η^2 depends on the noncentralities for all effects in the model. An exact confidence interval is not available, but if you write the formula for $\hat{\eta}^2$ as

$$\hat{\eta}^2 = \frac{\mathrm{SS}}{\mathrm{SS} + (\mathrm{SS}_{total} - \mathrm{SS})}$$

then a conservative confidence interval can be constructed as for $\eta^2_{partial}$, treating $\mathrm{SS}_{total} - \mathrm{SS}$ as the SSE and $N - \mathrm{DF} - 1$ as the DFE (Smithson 2004). This confidence interval is conservative in the sense that it implies values of the true η^2 that are smaller than they should be.

Estimates and confidence intervals for effect sizes require some care in interpretation. For example, while the true proportions of total and partial variation accounted for are nonnegative quantities, their estimates might be less than zero. Also, confidence intervals for effect sizes are not directly related to the corresponding estimates. In particular, it is possible for the estimate to lie outside the confidence interval.

As for interpreting the actual values of effect size measures, the approximately unbiased ω^2 estimates are usually preferred for point estimates. Some authors have proposed certain ranges as indicating "small," "medium," and "large" effects (Cohen 1988), but general benchmarks like this depend on the nature of the data and the typical signal-to-noise ratio; they should not be expected to apply across various disciplines. For example, while an ω^2 value of 10% might be viewed as "large" for psychometric data, it can be a relatively small effect for industrial experimentation. Whatever the standard, confidence intervals for true effect sizes typically span more than one category, indicating that in small experiments, it can be difficult to make firm statements about the size of effects.

Example

The data for this example are similar to data analyzed in Steiger and Fouladi (1997), Fidler and Thompson (2001), and Smithson (2003). Consider the following hypothetical design, testing 28 men and 28 women on seven different tasks.

```
data Test;
   do Task   = 1 to 7;
   do Gender = 'M','F';
      do i = 1 to 4;
         input Response @@;
         output;
         end;
      end;
      end;
datalines;
7.1 2.8 3.9 3.7     6.5 6.5 6.5 6.6
7.1 5.5 4.8 2.6     3.6 5.4 5.6 4.5
7.2 4.6 4.9 4.6     3.3 5.4 2.8 1.5
5.6 6.2 5.4 6.5     5.6 2.7 3.8 2.3
2.2 5.4 5.6 8.4     1.2 2.0 4.3 4.6
9.1 4.5 7.6 4.9     4.3 7.7 6.5 7.7
4.5 3.8 5.9 6.1     1.7 2.5 4.3 2.7
;
```

This is a balanced two-way design with four replicates per cell. The following statements analyze this data. Since this is a balanced design, you can use the SS1 option in the MODEL statement to display only the Type I sums of squares.

```
proc glm data=Test;
   class Gender Task;
   model Response = Gender|Task / ss1;
run;
```

The analysis of variance results are shown in Figure 39.15.

Figure 39.15 Two-Way Analysis of Variance

The GLM Procedure

Dependent Variable: Response

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Gender	1	14.40285714	14.40285714	6.00	0.0185
Task	6	38.15964286	6.35994048	2.65	0.0285
Gender*Task	6	35.99964286	5.99994048	2.50	0.0369

You can see that the two main effects as well as their interaction are all significant. Suppose you want to compare the main effect of Gender with the interaction between Gender and Task. The sums of squares for the interaction are more than twice as large, but it's not clear how experimental variability might affect this. The following statements perform the same analysis as before, but add

the EFFECTSIZE option to the MODEL statement; also, with ALPHA=0.1 option displays 90% confidence intervals, ensuring that inferences based on the p-values at the 0.05 levels will agree with the lower confidence limit.

```
proc glm data=Test;
   class Gender Task;
   model Response = Gender|Task / ss1 effectsize alpha=0.1;
run;
```

The Type I analysis of variance results with added effect size information are shown in Figure 39.16.

Figure 39.16 Two-Way Analysis of Variance with Effect Sizes

The GLM Procedure

Dependent Variable: Response

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Gender	1	14.40285714	14.40285714	6.00	0.0185
Task	6	38.15964286	6.35994048	2.65	0.0285
Gender*Task	6	35.99964286	5.99994048	2.50	0.0369

Noncentrality Parameter

Source	Min Var Unbiased Estimate	Low MSE Estimate	90% Confidence Limits	
Gender	4.72	4.48	0.521	17.1
Task	9.14	8.69	0.870	27.3
Gender*Task	8.29	7.87	0.463	25.9

Total Variation Accounted For

Source	Semipartial Eta-Square	Semipartial Omega-Square	Conservative 90% Confidence Limits	
Gender	0.0761	0.0626	0.0019	0.2030
Task	0.2015	0.1239	0.0000	0.2772
Gender*Task	0.1901	0.1126	0.0000	0.2639

Partial Variation Accounted For

Source	Partial Eta-Square	Partial Omega-Square	90% Confidence Limits	
Gender	0.1250	0.0820	0.0092	0.2342
Task	0.2746	0.1502	0.0153	0.3277
Gender*Task	0.2632	0.1385	0.0082	0.3160

The estimated effect sizes for Gender and the interaction all tell pretty much the same story: the effect of the interaction is appreciably greater than the effect of Gender. However, the confidence

intervals suggest that this inference should be treated with some caution, since the lower confidence bound for the Gender effect is greater than the lower confidence bound for the interaction in all three cases. Follow-up testing is probably in order, using the estimated effect sizes in this preliminary study to design a large enough sample to distinguish the sizes of the effects.

Absorption

Absorption is a computational technique used to reduce computing resource needs in certain cases. The classic use of absorption occurs when a blocking factor with a large number of levels is a term in the model.

For example, the statements

```
proc glm;
   absorb herd;
   class a b;
   model y=a b a*b;
run;
```

are equivalent to

```
proc glm;
   class herd a b;
   model y=herd a b a*b;
run;
```

The exception to the previous statements is that the Type II, Type III, or Type IV SS for HERD are not computed when HERD is absorbed.

The algorithm for absorbing variables is similar to the one used by the NESTED procedure for computing a nested analysis of variance. As each new row of $[X|Y]$ (corresponding to the nonabsorbed independent effects and the dependent variables) is constructed, it is adjusted for the absorbed effects in a Type I fashion. The efficiency of the absorption technique is due to the fact that this adjustment can be done in one pass of the data and without solving any linear equations, assuming that the data have been sorted by the absorbed variables.

Several effects can be absorbed at one time. For example, these statements

```
proc glm;
   absorb herd cow;
   class a b;
   model y=a b a*b;
run;
```

are equivalent to

```
proc glm;
   class herd cow a b;
   model y=herd cow(herd) a b a*b;
run;
```

When you use absorption, the size of the $\mathbf{X}'\mathbf{X}$ matrix is a function only of the effects in the MODEL statement. The effects being absorbed do not contribute to the size of the $\mathbf{X}'\mathbf{X}$ matrix.

For the preceding example, a and b can be absorbed:

```
proc glm;
   absorb a b;
   class herd cow;
   model y=herd cow(herd);
run;
```

Although the sources of variation in the results are listed as

```
a b(a) herd cow(herd)
```

all types of estimable functions for herd and cow(herd) are free of a, b, and a*b parameters.

To illustrate the savings in computing by using the ABSORB statement, PROC GLM is run on generated data with 1147 degrees of freedom in the model with the following statements.

```
data a;
   do herd=1 to 40;
      do cow=1 to 30;
         do treatment=1 to 3;
            do rep=1 to 2;
               y = herd/5 + cow/10 + treatment + rannor(1);
               output;
            end;
         end;
      end;
   end;
run;

proc glm data=a;
   class herd cow treatment;
   model y=herd cow(herd) treatment;
run;
```

This analysis would have required over 6 megabytes of memory for the $\mathbf{X}'\mathbf{X}$ matrix had PROC GLM solved it directly. However, in the following statements, the GLM procedure needs only a 4×4 matrix for the intercept and treatment because the other effects are absorbed.

```
proc glm data=a;
   absorb herd cow;
   class treatment;
   model y = treatment;
run;
```

These statements produce the results shown in Figure 39.17.

Figure 39.17 Absorption of Effects

```
                              The GLM Procedure

                           Class Level Information

                       Class          Levels    Values

                       treatment           3    1 2 3

                   Number of Observations Read        7200
                   Number of Observations Used        7200

                              The GLM Procedure

Dependent Variable: y

                                      Sum of
 Source                     DF        Squares    Mean Square   F Value   Pr > F

 Model                    1201    49465.40242       41.18685     41.57   <.0001

 Error                    5998     5942.23647        0.99070

 Corrected Total          7199    55407.63889

               R-Square     Coeff Var      Root MSE        y Mean

               0.892754      13.04236      0.995341      7.631598

 Source                     DF      Type I SS    Mean Square   F Value   Pr > F

 herd                       39    38549.18655      988.44068    997.72   <.0001
 cow(herd)                1160     6320.18141        5.44843      5.50   <.0001
 treatment                   2     4596.03446     2298.01723   2319.58   <.0001

 Source                     DF    Type III SS    Mean Square   F Value   Pr > F

 treatment                   2    4596.034455    2298.017228   2319.58   <.0001
```

Specification of ESTIMATE Expressions

Consider the model

$$E(Y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3$$

The corresponding MODEL statement for PROC GLM is

```
model y=x1 x2 x3;
```

To estimate the difference between the parameters for x_1 and x_2,

$$\beta_1 - \beta_2 \quad = \quad (\ 0 \quad 1 \quad -1 \quad 0\ )\boldsymbol{\beta}, \text{ where } \boldsymbol{\beta} = (\ \beta_0 \quad \beta_1 \quad \beta_2 \quad \beta_3\ )'$$

you can use the following ESTIMATE statement:

```
estimate 'B1-B2'  x1 1  x2 -1;
```

To predict y at $x_1 = 1$, $x_2 = 0$, and $x_3 = -2$, you can estimate

$$\beta_0 + \beta_1 - 2\beta_3 \quad = \quad (\ 1 \quad 1 \quad 0 \quad -2\ )\boldsymbol{\beta}$$

with the following ESTIMATE statement:

```
estimate 'B0+B1-2B3' intercept 1 x1 1 x3 -2;
```

Now consider models involving classification variables such as

```
model y=A B A*B;
```

with the associated parameters:

$$(\ \mu \quad \alpha_1 \quad \alpha_2 \quad \alpha_3 \quad \beta_1 \quad \beta_2 \quad \gamma_{11} \quad \gamma_{12} \quad \gamma_{21} \quad \gamma_{22} \quad \gamma_{31} \quad \gamma_{32}\ )$$

The LS-mean for the first level of A is $\mathbf{L}\boldsymbol{\beta}$, where

$$\mathbf{L} = (\ 1 \quad | \quad 1 \quad 0 \quad 0 \quad | \quad 0.5 \quad 0.5 \quad | \quad 0.5 \quad 0.5 \quad 0 \quad 0 \quad 0 \quad 0\ )$$

You can estimate this with the following ESTIMATE statement:

```
estimate 'LS-mean(A1)' intercept 1 A 1 B 0.5 0.5 A*B 0.5 0.5;
```

Note in this statement that only one element of $\mathbf{L}$ is specified following the A effect, even though A has three levels. Whenever the list of constants following an effect name is shorter than the effect's number of levels, zeros are used as the remaining constants. (If the list of constants is longer than the number of levels for the effect, the extra constants are ignored, and a warning message is displayed.)

To estimate the A linear effect in the preceding model, assuming equally spaced levels for A, you can use the following $\mathbf{L}$:

$$\mathbf{L} = (\ 0 \quad | \quad -1 \quad 0 \quad 1 \quad | \quad 0 \quad 0 \quad | \quad -0.5 \quad -0.5 \quad 0 \quad 0 \quad 0.5 \quad 0.5\ )$$

The ESTIMATE statement for this $\mathbf{L}$ is written as

```
estimate 'A Linear' A -1 0 1;
```

If you do not specify the elements of $\mathbf{L}$ for an effect that contains a specified effect, then the elements of the specified effect are equally distributed over the corresponding levels of the higher-order effect. In addition, if you specify the intercept in an ESTIMATE or CONTRAST statement, it is distributed over all classification effects that are not contained by any other specified effect.

The distribution of lower-order coefficients to higher-order effect coefficients follows the same general rules as in the LSMEANS statement, and it is similar to that used to construct Type IV tests. In the previous example, the -1 associated with α_1 is divided by the number n_{1j} of γ_{1j} parameters; then each γ_{1j} coefficient is set to $-1/n_{1j}$. The 1 associated with α_3 is distributed among the γ_{3j} parameters in a similar fashion. In the event that an unspecified effect contains several specified effects, only that specified effect with the most factors in common with the unspecified effect is used for distribution of coefficients to the higher-order effect.

Numerous syntactical expressions for the ESTIMATE statement were considered, including many that involved specifying the effect and level information associated with each coefficient. For models involving higher-level effects, the requirement of specifying level information can lead to very bulky specifications. Consequently, the simpler form of the ESTIMATE statement described earlier was implemented.

The syntax of this ESTIMATE statement puts a burden on you to know a priori the order of the parameter list associated with each effect. You can use the ORDER= option in the PROC GLM statement to ensure that the levels of the classification effects are sorted appropriately.

NOTE: If you use the ESTIMATE statement with unspecified effects, use the E option to make sure that the actual $\mathbf{L}$ constructed by the preceding rules is the one you intended.

A Check for Estimability

Each $\mathbf{L}$ is checked for estimability using the relationship $\mathbf{L} = \mathbf{LH}$, where $\mathbf{H} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$. The $\mathbf{L}$ vector is declared nonestimable, if for any i

$$\text{ABS}(\mathbf{L}_i - (\mathbf{LH})_i) > \begin{cases} \epsilon & \text{if } \mathbf{L}_i = 0 \text{ or} \\ \epsilon \times \text{ABS}(\mathbf{L}_i) & \text{otherwise} \end{cases}$$

where $\epsilon = 10^{-4}$ by default; you can change this with the SINGULAR= option. Continued fractions (like 1/3) should be specified to at least six decimal places, or the DIVISOR parameter should be used.

Comparing Groups

An important task in analyzing data with classification effects is to estimate the typical response for each level of a given effect; often, you also want to compare these estimates to determine which levels are equivalent in terms of the response. You can perform this task in two ways with the GLM procedure: with direct, arithmetic group means; and with so-called *least squares means* (LS-means).

Means versus LS-Means

Computing and comparing arithmetic means—either simple or weighted within-group averages of the input data—is a familiar and well-studied statistical process. This is the right approach to summarizing and comparing groups for one-way and balanced designs. However, in unbalanced

designs with more than one effect, the arithmetic mean for a group might not accurately reflect the "typical" response for that group, since it does not take other effects into account.

For example, the following analysis of an unbalanced two-way design produces the ANOVA, means, and LS-means shown in Figure 39.18, Figure 39.19, and Figure 39.20.

```
data twoway;
   input Treatment Block y @@;
   datalines;
1 1 17   1 1 28   1 1 19   1 1 21   1 1 19
1 2 43   1 2 30   1 2 39   1 2 44   1 2 44
1 3 16
2 1 21   2 1 21   2 1 24   2 1 25
2 2 39   2 2 45   2 2 42   2 2 47
2 3 19   2 3 22   2 3 16
3 1 22   3 1 30   3 1 33   3 1 31
3 2 46
3 3 26   3 3 31   3 3 26   3 3 33   3 3 29   3 3 25
;

title "Unbalanced Two-way Design";
ods select ModelANOVA Means LSMeans;

proc glm data=twoway;
   class Treatment Block;
   model y = Treatment|Block;
   means Treatment;
   lsmeans Treatment;
run;

ods select all;
```

Figure 39.18 ANOVA Results for Unbalanced Two-Way Design

Unbalanced Two-way Design

The GLM Procedure

Dependent Variable: y

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Treatment	2	8.060606	4.030303	0.24	0.7888
Block	2	2621.864124	1310.932062	77.95	<.0001
Treatment*Block	4	32.684361	8.171090	0.49	0.7460

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Treatment	2	266.130682	133.065341	7.91	0.0023
Block	2	1883.729465	941.864732	56.00	<.0001
Treatment*Block	4	32.684361	8.171090	0.49	0.7460

Figure 39.19 Treatment Means for Unbalanced Two-Way Design

Unbalanced Two-way Design

The GLM Procedure

Level of Treatment	N	y Mean	y Std Dev
1	11	29.0909091	11.5104695
2	11	29.1818182	11.5569735
3	11	30.1818182	6.3058414

Figure 39.20 Treatment LS-means for Unbalanced Two-Way Design

Unbalanced Two-way Design

The GLM Procedure
Least Squares Means

Treatment	y LSMEAN
1	25.6000000
2	28.3333333
3	34.4444444

No matter how you look at them, these data exhibit a strong effect due to the blocks (F test $p < 0.0001$) and no significant interaction between treatments and blocks (F test $p > 0.7$). But the lack of balance affects how the treatment effect is interpreted: in a main-effects-only model, there are no significant differences between the treatment means themselves (Type I F test $p > 0.7$), but there are highly significant differences between the treatment means corrected for the block effects (Type III F test $p < 0.01$).

LS-means are, in effect, within-group means appropriately adjusted for the other effects in the model. More precisely, they estimate the marginal means for a balanced population (as opposed to the unbalanced design). For this reason, they are also called *estimated population marginal means* by Searle, Speed, and Milliken (1980). In the same way that the Type I F test assesses differences between the arithmetic treatment means (when the treatment effect comes first in the model), the Type III F test assesses differences between the LS-means. Accordingly, for the unbalanced two-way design, the discrepancy between the Type I and Type III tests is reflected in the arithmetic treatment means and treatment LS-means, as shown in Figure 39.19 and Figure 39.20. See the section "Construction of Least Squares Means" on page 2528 for more on LS-means.

Note that, while the arithmetic means are always uncorrelated (under the usual assumptions for analysis of variance; see page 2487), the LS-means might not be. This fact complicates the problem of multiple comparisons for LS-means; see the following section.

Multiple Comparisons

When comparing more than two means, an ANOVA F test tells you whether the means are significantly different from each other, but it does not tell you which means differ from which other means. Multiple-comparison procedures (MCPs), also called *mean separation tests*, give you more detailed information about the differences among the means. The goal in multiple comparisons is to compare the average effects of three or more "treatments" (for example, drugs, groups of subjects) to decide which treatments are better, which ones are worse, and by how much, while controlling the probability of making an incorrect decision. A variety of multiple-comparison methods are available with the MEANS and LSMEANS statement in the GLM procedure.

The following classification is due to Hsu (1996). Multiple-comparison procedures can be categorized in two ways: by the comparisons they make and by the strength of inference they provide. With respect to which comparisons are made, the GLM procedure offers two types:

- comparisons between all pairs of means
- comparisons between a control and all other means

The strength of inference says what can be inferred about the structure of the means when a test is significant; it is related to what type of error rate the MCP controls. MCPs available in the GLM procedure provide one of the following types of inference, in order from weakest to strongest:

- Individual: differences between means, unadjusted for multiplicity
- Inhomogeneity: means are different
- Inequalities: which means are different
- Intervals: simultaneous confidence intervals for mean differences

Methods that control only individual error rates are not true MCPs at all. Methods that yield the strongest level of inference, simultaneous confidence intervals, are usually preferred, since they enable you not only to say which means are different but also to put confidence bounds on *how much* they differ, making it easier to assess the practical significance of a difference. They are also less likely to lead nonstatisticians to the invalid conclusion that nonsignificantly different sample means imply equal population means. Interval MCPs are available for both arithmetic means and LS-means via the MEANS and LSMEANS statements, respectively.[1]

Table 39.7 and Table 39.8 display MCPs available in PROC GLM for all pairwise comparisons and comparisons with a control, respectively, along with associated strength of inference and the syntax (when applicable) for both the MEANS and the LSMEANS statements.

[1] The Duncan-Waller method does not fit into the preceding scheme, since it is based on the Bayes risk rather than any particular error rate.

Table 39.7 Multiple-Comparison Procedures for All Pairwise Comparisons

Method	Strength of Inference	Syntax	
		MEANS	LSMEANS
Student's *t*	Individual	T	PDIFF ADJUST=T
Duncan	Individual	DUNCAN	
Student-Newman-Keuls	Inhomogeneity	SNK	
REGWQ	Inequalities	REGWQ	
Tukey-Kramer	Intervals	TUKEY	PDIFF ADJUST=TUKEY
Bonferroni	Intervals	BON	PDIFF ADJUST=BON
Sidak	Intervals	SIDAK	PDIFF ADJUST=SIDAK
Scheffé	Intervals	SCHEFFE	PDIFF ADJUST=SCHEFFE
SMM	Intervals	SMM	PDIFF ADJUST=SMM
Gabriel	Intervals	GABRIEL	
Simulation	Intervals		PDIFF ADJUST=SIMULATE

Table 39.8 Multiple-Comparison Procedures for Comparisons with a Control

Method	Strength of Inference	Syntax	
		MEANS	LSMEANS
Student's *t*	Individual		PDIFF=CONTROL ADJUST=T
Dunnett	Intervals	DUNNETT	PDIFF=CONTROL ADJUST=DUNNETT
Bonferroni	Intervals		PDIFF=CONTROL ADJUST=BON
Sidak	Intervals		PDIFF=CONTROL ADJUST=SIDAK
Scheffé	Intervals		PDIFF=CONTROL ADJUST=SCHEFFE
SMM	Intervals		PDIFF=CONTROL ADJUST=SMM
Simulation	Intervals		PDIFF=CONTROL ADJUST=SIMULATE

NOTE: One-sided Dunnett's tests are also available from the MEANS statement with the DUNNETTL and DUNNETTU options and from the LSMEANS statement with PDIFF=CONTROLL and PDIFF=CONTROLU.

A note concerning the ODS tables for the results of the PDIFF or TDIFF options in the LSMEANS statement: The p/t-values for differences are displayed in columns of the LSMeans table for PDIFF/TDIFF=CONTROL or PDIFF/TDIFF=ANOM, and for PDIFF/TDIFF=ALL when there are only two LS-means. Otherwise (for PDIFF/TDIFF=ALL when there are more than two LS-means), the p/t-values for differences are displayed in a separate table called Diff.

Details of these multiple comparison methods are given in the following sections.

Pairwise Comparisons

All the methods discussed in this section depend on the standardized pairwise differences $t_{ij} = (\bar{y}_i - \bar{y}_j)/\hat{\sigma}_{ij}$, where the parts of this expression are defined as follows:

- i and j are the indices of two groups
- $\bar{y}_i$ and $\bar{y}_j$ are the means or LS-means for groups i and j

- $\hat{\sigma}_{ij}$ is the square root of the estimated variance of $\bar{y}_i - \bar{y}_j$. For simple arithmetic means, $\hat{\sigma}^2_{ij} = s^2(1/n_i + 1/n_j)$, where n_i and n_j are the sizes of groups i and j, respectively, and s^2 is the mean square for error, with ν degrees of freedom. For weighted arithmetic means, $\hat{\sigma}^2_{ij} = s^2(1/w_i + 1/w_j)$, where w_i and w_j are the sums of the weights in groups i and j, respectively. Finally, for LS-means defined by the linear combinations $l_i'b$ and $l_j'b$ of the parameter estimates, $\hat{\sigma}^2_{ij} = s^2 l_i'(\mathbf{X}'\mathbf{X})^- l_j$.

Furthermore, all of the methods are discussed in terms of significance tests of the form

$$|t_{ij}| \quad \geq \quad c(\alpha)$$

where $c(\alpha)$ is some constant depending on the significance level. Such tests can be inverted to form confidence intervals of the form

$$(\bar{y}_i - \bar{y}_j) - \hat{\sigma}_{ij}c(\alpha) \quad \leq \quad \mu_i - \mu_j \quad \leq \quad (\bar{y}_i - \bar{y}_j) + \hat{\sigma}_{ij}c(\alpha)$$

The simplest approach to multiple comparisons is to do a t test on every pair of means (the T option in the MEANS statement, ADJUST=T in the LSMEANS statement). For the ith and jth means, you can reject the null hypothesis that the population means are equal if

$$|t_{ij}| \quad \geq \quad t(\alpha; \nu)$$

where α is the significance level, ν is the number of error degrees of freedom, and $t(\alpha; \nu)$ is the two-tailed critical value from a Student's t distribution. If the cell sizes are all equal to, say, n, the preceding formula can be rearranged to give

$$|\bar{y}_i - \bar{y}_j| \quad \geq \quad t(\alpha; \nu)s\sqrt{\frac{2}{n}}$$

the value of the right-hand side being Fisher's least significant difference (LSD).

There is a problem with repeated t tests, however. Suppose there are 10 means and each t test is performed at the 0.05 level. There are $10(10-1)/2 = 45$ pairs of means to compare, each with a 0.05 probability of a type 1 error (a false rejection of the null hypothesis). The chance of making at least one type 1 error is much higher than 0.05. It is difficult to calculate the exact probability, but you can derive a pessimistic approximation by assuming that the comparisons are independent, giving an upper bound to the probability of making at least one type 1 error (the experimentwise error rate) of

$$1 - (1 - 0.05)^{45} \quad = \quad 0.90$$

The actual probability is somewhat less than 0.90, but as the number of means increases, the chance of making at least one type 1 error approaches 1.

If you decide to control the individual type 1 error rates for each comparison, you are controlling the individual or comparisonwise error rate. On the other hand, if you want to control the overall

type 1 error rate for all the comparisons, you are controlling the experimentwise error rate. It is up to you to decide whether to control the comparisonwise error rate or the experimentwise error rate, but there are many situations in which the experimentwise error rate should be held to a small value. Statistical methods for comparing three or more means while controlling the probability of making at least one type 1 error are called *multiple-comparison procedures*.

It has been suggested that the experimentwise error rate can be held to the α level by performing the overall ANOVA F test at the α level and making further comparisons only if the F test is significant, as in Fisher's protected LSD. This assertion is false if there are more than three means (Einot and Gabriel 1975). Consider again the situation with 10 means. Suppose that one population mean differs from the others by such a sufficiently large amount that the power (probability of correctly rejecting the null hypothesis) of the F test is near 1 but that all the other population means are equal to each other. There will be $9(9-1)/2 = 36$ t tests of true null hypotheses, with an upper limit of 0.84 on the probability of at least one type 1 error. Thus, you must distinguish between the experimentwise error rate under the complete null hypothesis, in which all population means are equal, and the experimentwise error rate under a partial null hypothesis, in which some means are equal but others differ. The following abbreviations are used in the discussion:

CER	comparisonwise error rate
EERC	experimentwise error rate under the complete null hypothesis
MEER	maximum experimentwise error rate under any complete or partial null hypothesis

These error rates are associated with the different strengths of inference discussed on page 2514: individual tests control the CER; tests for inhomogeneity of means control the EERC; tests that yield confidence inequalities or confidence intervals control the MEER. A preliminary F test controls the EERC but not the MEER.

You can control the MEER at the α level by setting the CER to a sufficiently small value. The Bonferroni inequality (Miller, R. G., Jr. 1981) has been widely used for this purpose. If

$$\text{CER} = \frac{\alpha}{c}$$

where c is the total number of comparisons, then the MEER is less than α. Bonferroni t tests (the BON option in the MEANS statement, ADJUST=BON in the LSMEANS statement) with MEER $< \alpha$ declare two means to be significantly different if

$$|t_{ij}| \geq t(\epsilon; \nu)$$

where

$$\epsilon = \frac{2\alpha}{k(k-1)}$$

for comparison of k means.

Šidák (1967) has provided a tighter bound, showing that

$$\text{CER} = 1 - (1-\alpha)^{1/c}$$

also ensures that MEER $\leq \alpha$ for any set of c comparisons. A Sidak t test (Games 1977), provided by the SIDAK option, is thus given by

$$|t_{ij}| \geq t(\epsilon; \nu)$$

where

$$\epsilon = 1 - (1-\alpha)^{\frac{2}{k(k-1)}}$$

for comparison of k means.

You can use the Bonferroni additive inequality and the Sidak multiplicative inequality to control the MEER for any set of contrasts or other hypothesis tests, not just pairwise comparisons. The Bonferroni inequality can provide simultaneous inferences in any statistical application requiring tests of more than one hypothesis. Other methods discussed in this section for pairwise comparisons can also be adapted for general contrasts (Miller, R. G., Jr. 1981).

Scheffé (1953, 1959) proposes another method to control the MEER for any set of contrasts or other linear hypotheses in the analysis of linear models, including pairwise comparisons, obtained with the SCHEFFE option. Two means are declared significantly different if

$$|t_{ij}| \geq \sqrt{(k-1)F(\alpha; k-1, \nu)}$$

where $F(\alpha; k-1, \nu)$ is the α-level critical value of an F distribution with $k-1$ numerator degrees of freedom and ν denominator degrees of freedom.

Scheffé's test is compatible with the overall ANOVA F test in that Scheffé's method never declares a contrast significant if the overall F test is nonsignificant. Most other multiple-comparison methods can find significant contrasts when the overall F test is nonsignificant and, therefore, suffer a loss of power when used with a preliminary F test.

Scheffé's method might be more powerful than the Bonferroni or Sidak method if the number of comparisons is large relative to the number of means. For pairwise comparisons, Sidak t tests are generally more powerful.

Tukey (1952, 1953) proposes a test designed specifically for pairwise comparisons based on the studentized range, sometimes called the "honestly significant difference test," that controls the MEER when the sample sizes are equal. Tukey (1953) and Kramer (1956) independently propose a modification for unequal cell sizes. The Tukey or Tukey-Kramer method is provided by the TUKEY option in the MEANS statement and the ADJUST=TUKEY option in the LSMEANS statement. This method has fared extremely well in Monte Carlo studies (Dunnett 1980). In addition, Hayter (1984) gives a proof that the Tukey-Kramer procedure controls the MEER for means comparisons, and Hayter (1989) describes the extent to which the Tukey-Kramer procedure has been proven to control the MEER for LS-means comparisons. The Tukey-Kramer method is more powerful than the Bonferroni, Sidak, or Scheffé method for pairwise comparisons. Two means are considered significantly different by the Tukey-Kramer criterion if

$$|t_{ij}| \geq q(\alpha; k, \nu)$$

where $q(\alpha; k, \nu)$ is the α-level critical value of a studentized range distribution of k independent normal random variables with ν degrees of freedom.

Hochberg (1974) devised a method (the GT2 or SMM option) similar to Tukey's, but it uses the studentized maximum modulus instead of the studentized range and employs the uncorrelated t inequality of Šidák (1967). It is proven to hold the MEER at a level not exceeding α with unequal sample sizes. It is generally less powerful than the Tukey-Kramer method and always less powerful than Tukey's test for equal cell sizes. Two means are declared significantly different if

$$|t_{ij}| \quad \geq \quad m(\alpha; c, \nu)$$

where $m(\alpha; c, \nu)$ is the α-level critical value of the studentized maximum modulus distribution of c independent normal random variables with ν degrees of freedom and $c = k(k-1)/2$.

Gabriel (1978) proposes another method (the GABRIEL option) based on the studentized maximum modulus. This method is applicable only to arithmetic means. It rejects if

$$\frac{|\bar{y}_i - \bar{y}_j|}{s\left(\frac{1}{\sqrt{2n_i}} + \frac{1}{\sqrt{2n_j}}\right)} \quad \geq \quad m(\alpha; k, \nu)$$

For equal cell sizes, Gabriel's test is equivalent to Hochberg's GT2 method. For unequal cell sizes, Gabriel's method is more powerful than GT2 but might become liberal with highly disparate cell sizes (see also Dunnett (1980)). Gabriel's test is the only method for unequal sample sizes that lends itself to a graphical representation as intervals around the means. Assuming $\bar{y}_i > \bar{y}_j$, you can rewrite the preceding inequality as

$$\bar{y}_i - m(\alpha; k, \nu)\frac{s}{\sqrt{2n_i}} \quad \geq \quad \bar{y}_j + m(\alpha; k, \nu)\frac{s}{\sqrt{2n_j}}$$

The expression on the left does not depend on j, nor does the expression on the right depend on i. Hence, you can form what Gabriel calls an (l, u)-interval around each sample mean and declare two means to be significantly different if their (l, u)-intervals do not overlap. See Hsu (1996, section 5.2.1.1) for a discussion of other methods of graphically representing all pairwise comparisons.

Comparing All Treatments to a Control

One special case of means comparison is that in which the only comparisons that need to be tested are between a set of new treatments and a single control. In this case, you can achieve better power by using a method that is restricted to test only comparisons to the single control mean. Dunnett (1955) proposes a test for this situation that declares a mean significantly different from the control if

$$|t_{i0}| \quad \geq \quad d(\alpha; k, \nu, \rho_1, \ldots, \rho_{k-1})$$

where $\bar{y}_0$ is the control mean and $d(\alpha; k, \nu, \rho_1, \ldots, \rho_{k-1})$ is the critical value of the "many-to-one t statistic" (Miller, R. G., Jr. 1981; Krishnaiah and Armitage 1966) for k means to be compared to

a control, with ν error degrees of freedom and correlations $\rho_1, \ldots, \rho_{k-1}$, $\rho_i = n_i/(n_0 + n_i)$. The correlation terms arise because each of the treatment means is being compared to the same control. Dunnett's test holds the MEER to a level not exceeding the stated α.

Analysis of Means: Comparing Each Treatments to the Average

Analysis of means (ANOM) refers to a technique for comparing group means and displaying the comparisons graphically so that you can easily see which ones are different. Means are judged as different if they are significantly different from the overall average, with significance adjusted for multiplicity. The overall average is computed as a weighted mean of the LS-means, the weights being inversely proportional to the variances. If you use the PDIFF=ANOM option in the LSMEANS statement, the procedure will display the p-values (adjusted for multiplicity, by default) for tests of the differences between each LS-mean and the average LS-mean. The ANOM procedure in SAS/QC software displays both tables and graphics for the analysis of means with a variety of response types. For one-way designs, confidence intervals for PDIFF=ANOM comparisons are equivalent to the results of PROC ANOM. The difference is that PROC GLM directly displays the confidence intervals for the differences, while the graphical output of PROC ANOM displays them as decision limits around the overall mean.

If the LS-means being compared are uncorrelated, exact adjusted p-values and critical values for confidence limits can be computed; see Nelson (1982, 1991, 1993) and Guirguis and Tobias (2004). For correlated LS-means, an approach similar to that of Hsu (1992) is employed, using a factor-analytic approximation of the correlation between the LS-means to derive approximate "effective sample sizes" for which exact critical values are computed. Note that computing the exact adjusted p-values and critical values for unbalanced designs can be computationally intensive. A simulation-based approach, as specified by the ADJUST=SIM option, while nondeterministic, might provide inferences that are accurate enough in much less time. See the section "Approximate and Simulation-Based Methods" on page 2520 for more details.

Approximate and Simulation-Based Methods

Tukey's, Dunnett's, and Nelson's tests are all based on the same general quantile calculation:

$$q^t(\alpha, \nu, R) \quad = \quad \{q \ni P(\max(|t_1|, \ldots, |t_n|) > q) = \alpha\}$$

where the t_i have a joint multivariate t distribution with ν degrees of freedom and correlation matrix R. In general, evaluating $q^t(\alpha, \nu, R)$ requires repeated numerical calculation of an $(n + 1)$-fold integral. This is usually intractable, but the problem reduces to a feasible 2-fold integral when R has a certain symmetry in the case of Tukey's test, and a *factor analytic structure* (Hsu 1992) in the case of Dunnett's and Nelson's tests. The R matrix has the required symmetry for exact computation of Tukey's test in the following two cases:

- The t_is are studentized differences between $k(k-1)/2$ pairs of k uncorrelated means with equal variances—that is, equal sample sizes.
- The t_is are studentized differences between $k(k-1)/2$ pairs of k LS-means from a *variance-balanced* design (for example, a balanced incomplete block design).

See Hsu (1992, 1996) for more information. The R matrix has the factor analytic structure for exact computation of Dunnett's and Nelson's tests in the following two cases:

- if the t_is are studentized differences between $k-1$ means and a control mean, all uncorrelated. (Dunnett's one-sided methods depend on a similar probability calculation, without the absolute values.) Note that it is not required that the variances of the means (that is, the sample sizes) be equal.
- if the t_is are studentized differences between $k-1$ LS-means and a control LS-mean from either a *variance-balanced* design, or a design in which the other factors are *orthogonal* to the treatment factor (for example, a randomized block design with proportional cell frequencies)

However, other important situations that do **not** result in a correlation matrix R that has the structure for exact computation are the following:

- all pairwise differences with unequal sample sizes
- differences between LS-means in many unbalanced designs

In these situations, exact calculation of $q^t(\alpha, \nu, R)$ is intractable in general. Most of the preceding methods can be viewed as using various approximations for $q^t(\alpha, \nu, R)$. When the sample sizes are unequal, the Tukey-Kramer test is equivalent to another approximation. For comparisons with a control when the correlation R does not have a factor analytic structure, Hsu (1992) suggests approximating R with a matrix R^* that does have such a structure and correspondingly approximating $q^t(\alpha, \nu, R)$ with $q^t(\alpha, \nu, R^*)$. When you request Dunnett's or Nelson's test for LS-means (the PDIFF=CONTROL and ADJUST=DUNNETT options or the PDIFF=ANOM and ADJUST=NELSON options, respectively), the GLM procedure automatically uses Hsu's approximation when appropriate.

Finally, Edwards and Berry (1987) suggest calculating $q^t(\alpha, \nu, R)$ by simulation. Multivariate t vectors are sampled from a distribution with the appropriate ν and R parameters, and Edwards and Berry (1987) suggest estimating $q^t(\alpha, \nu, R)$ by $\hat{q}$, the α percentile of the observed values of $\max(|t_1|, \ldots, |t_n|)$. Sufficient samples are generated for the true $P(\max(|t_1|, \ldots, |t_n|) > \hat{q})$ to be within a certain accuracy radius γ of α with accuracy confidence $100(1-\epsilon)$. You can approximate $q^t(\alpha, \nu, R)$ by simulation for comparisons between LS-means by specifying ADJUST=SIM (with any PDIFF= type). By default, $\gamma = 0.005$ and $\epsilon = 0.01$, so that the tail area of $\hat{q}$ is within 0.005 of α with 99% confidence. You can use the ACC= and EPS= options with ADJUST=SIM to reset γ and ϵ, or you can use the NSAMP= option to set the sample size directly. You can also control the random number sequence with the SEED= option.

Hsu and Nelson (1998) suggest a more accurate simulation method for estimating $q^t(\alpha, \nu, R)$, using a control variate adjustment technique. The same independent, standardized normal variates that are used to generate multivariate t vectors from a distribution with the appropriate ν and R parameters are also used to generate multivariate t vectors from a distribution for which the exact value of $q^t(\alpha, \nu, R)$ is known. $\max(|t_1|, \ldots, |t_n|)$ for the second sample is used as a control variate for adjusting the quantile estimate based on the first sample; see Hsu and Nelson (1998) for more details. The control variate adjustment has the drawback that it takes somewhat longer than the crude technique of Edwards and Berry (1987), but it typically yields an estimate that is many

times more accurate. In most cases, if you are using ADJUST=SIM, then you should specify ADJUST=SIM(CVADJUST). You can also specify ADJUST=SIM(CVADJUST REPORT) to display a summary of the simulation that includes, among other things, the actual accuracy radius γ, which should be substantially smaller than the target accuracy radius (0.005 by default).

Multiple-Stage Tests

You can use all of the methods discussed so far to obtain simultaneous confidence intervals (Miller, R. G., Jr. 1981). By sacrificing the facility for simultaneous estimation, you can obtain simultaneous tests with greater power by using multiple-stage tests (MSTs). MSTs come in both step-up and step-down varieties (Welsch 1977). The step-down methods, which have been more widely used, are available in SAS/STAT software.

Step-down MSTs first test the homogeneity of all the means at a level γ_k. If the test results in a rejection, then each subset of $k-1$ means is tested at level γ_{k-1}; otherwise, the procedure stops. In general, if the hypothesis of homogeneity of a set of p means is rejected at the γ_p level, then each subset of $p-1$ means is tested at the γ_{p-1} level; otherwise, the set of p means is considered not to differ significantly and none of its subsets are tested. The many varieties of MSTs that have been proposed differ in the levels γ_p and the statistics on which the subset tests are based. Clearly, the EERC of a step-down MST is not greater than γ_k, and the CER is not greater than γ_2, but the MEER is a complicated function of γ_p, $p = 2, \ldots, k$.

With unequal cell sizes, PROC GLM uses the harmonic mean of the cell sizes as the common sample size. However, since the resulting operating characteristics can be undesirable, MSTs are recommended only for the balanced case. When the sample sizes are equal, using the range statistic enables you to arrange the means in ascending or descending order and test only contiguous subsets. But if you specify the F statistic, this shortcut cannot be taken. For this reason, only range-based MSTs are implemented. It is common practice to report the results of an MST by writing the means in such an order and drawing lines parallel to the list of means spanning the homogeneous subsets. This form of presentation is also convenient for pairwise comparisons with equal cell sizes.

The best-known MSTs are the Duncan (the DUNCAN option) and Student-Newman-Keuls (the SNK option) methods (Miller, R. G., Jr. 1981). Both use the studentized range statistic and, hence, are called *multiple range tests*. Duncan's method is often called the "new" multiple range test despite the fact that it is one of the oldest MSTs in current use.

The Duncan and SNK methods differ in the γ_p values used. For Duncan's method, they are

$$\gamma_p = 1 - (1-\alpha)^{p-1}$$

whereas the SNK method uses

$$\gamma_p = \alpha$$

Duncan's method controls the CER at the α level. Its operating characteristics appear similar to those of Fisher's unprotected LSD or repeated t tests at level α (Petrinovich and Hardyck 1969). Since repeated t tests are easier to compute, easier to explain, and applicable to unequal sample sizes, Duncan's method is not recommended. Several published studies (for example, Carmer and

Swanson (1973)) have claimed that Duncan's method is superior to Tukey's because of greater power without considering that the greater power of Duncan's method is due to its higher type 1 error rate (Einot and Gabriel 1975).

The SNK method holds the EERC to the α level but does not control the MEER (Einot and Gabriel 1975). Consider ten population means that occur in five pairs such that means within a pair are equal, but there are large differences between pairs. If you make the usual sampling assumptions and also assume that the sample sizes are very large, all subset homogeneity hypotheses for three or more means are rejected. The SNK method then comes down to five independent tests, one for each pair, each at the α level. Letting α be 0.05, the probability of at least one false rejection is

$$1 - (1 - 0.05)^5 \quad = \quad 0.23$$

As the number of means increases, the MEER approaches 1. Therefore, the SNK method cannot be recommended.

A variety of MSTs that control the MEER have been proposed, but these methods are not as well known as those of Duncan and SNK. An approach developed by Ryan (1959, 1960), Einot and Gabriel (1975), and Welsch (1977) sets

$$\gamma_p \quad = \quad \begin{cases} 1 - (1 - \alpha)^{p/k} & \text{for } p < k - 1 \\ \alpha & \text{for } p \geq k - 1 \end{cases}$$

You can use range statistics, leading to what is called the REGWQ method, after the authors' initials. If you assume that the sample means have been arranged in descending order from $\bar{y}_1$ through $\bar{y}_k$, the homogeneity of means $\bar{y}_i, \ldots, \bar{y}_j, i < j$, is rejected by REGWQ if

$$\bar{y}_i - \bar{y}_j \quad \geq \quad q(\gamma_p; p, \nu)\frac{s}{\sqrt{n}}$$

where $p = j - i + 1$ and the summations are over $u = i, \ldots, j$ (Einot and Gabriel 1975). To ensure that the MEER is controlled, the current implementation checks whether $q(\gamma_p; p, \nu)$ is monotonically increasing in p. If not, then a set of critical values that are increasing in p is substituted instead.

REGWQ appears to be the most powerful step-down MST in the current literature (for example, Ramsey 1978). Use of a preliminary F test decreases the power of all the other multiple-comparison methods discussed previously except for Scheffé's test.

Bayesian Approach

Waller and Duncan (1969) and Duncan (1975) take an approach to multiple comparisons that differs from all the methods previously discussed in minimizing the Bayes risk under additive loss rather than controlling type 1 error rates. For each pair of population means μ_i and μ_j, null (H_0^{ij}) and alternative (H_a^{ij}) hypotheses are defined:

$$H_0^{ij}: \quad \mu_i - \mu_j \leq 0$$
$$H_a^{ij}: \quad \mu_i - \mu_j > 0$$

For any i, j pair, let d_0 indicate a decision in favor of H_0^{ij} and d_a indicate a decision in favor of H_a^{ij}, and let $\delta = \mu_i - \mu_j$. The loss function for the decision on the i, j pair is

$$L(d_0 \mid \delta) = \begin{cases} 0 & \text{if } \delta \leq 0 \\ \delta & \text{if } \delta > 0 \end{cases}$$

$$L(d_a \mid \delta) = \begin{cases} -k\delta & \text{if } \delta \leq 0 \\ 0 & \text{if } \delta > 0 \end{cases}$$

where k represents a constant that you specify rather than the number of means. The loss for the joint decision involving all pairs of means is the sum of the losses for each individual decision. The population means are assumed to have a normal prior distribution with unknown variance, the logarithm of the variance of the means having a uniform prior distribution. For the i, j pair, the null hypothesis is rejected if

$$\bar{y}_i - \bar{y}_j \geq t_B s \sqrt{\frac{2}{n}}$$

where t_B is the Bayesian t value (Waller and Kemp 1976) depending on k, the F statistic for the one-way ANOVA, and the degrees of freedom for F. The value of t_B is a decreasing function of F, so the Waller-Duncan test (specified by the WALLER option) becomes more liberal as F increases.

Recommendations

In summary, if you are interested in several individual comparisons and are not concerned about the effects of multiple inferences, you can use repeated t tests or Fisher's unprotected LSD. If you are interested in all pairwise comparisons or all comparisons with a control, you should use Tukey's or Dunnett's test, respectively, in order to make the strongest possible inferences. If you have weaker inferential requirements and, in particular, if you do not want confidence intervals for the mean differences, you should use the REGWQ method. Finally, if you agree with the Bayesian approach and Waller and Duncan's assumptions, you should use the Waller-Duncan test.

Interpretation of Multiple Comparisons

When you interpret multiple comparisons, remember that failure to reject the hypothesis that two or more means are equal should not lead you to conclude that the population means are, in fact, equal. Failure to reject the null hypothesis implies only that the difference between population means, if any, is not large enough to be detected with the given sample size. A related point is that nonsignificance is nontransitive: that is, given three sample means, the largest and smallest might be significantly different from each other, while neither is significantly different from the middle one. Nontransitive results of this type occur frequently in multiple comparisons.

Multiple comparisons can also lead to counterintuitive results when the cell sizes are unequal. Consider four cells labeled A, B, C, and D, with sample means in the order A>B>C>D. If A and D each have two observations, and B and C each have 10,000 observations, then the difference between B and C might be significant, while the difference between A and D is not.

Simple Effects

Suppose you use the following statements to fit a full factorial model to a two-way design:

```
data twoway;
   input A B Y @@;
   datalines;
1 1 10.6   1 1 11.0   1 1 10.6   1 1 11.3
1 2 -0.2   1 2  1.3   1 2 -0.2   1 2  0.2
1 3  0.1   1 3  0.4   1 3 -0.4   1 3  1.0
2 1 19.7   2 1 19.3   2 1 18.5   2 1 20.4
2 2 -0.2   2 2  0.5   2 2  0.8   2 2 -0.4
2 3 -0.9   2 3 -0.1   2 3 -0.2   2 3 -1.7
3 1 29.7   3 1 29.6   3 1 29.0   3 1 30.2
3 2  1.5   3 2  0.2   3 2 -1.5   3 2  1.3
3 3  0.2   3 3  0.4   3 3 -0.4   3 3 -2.2
;
proc glm data=twoway;
   class A B;
   model Y = A B A*B;
run;
```

Partial results for the analysis of variance are shown in Figure 39.21. The Type I and Type III results are the same because this is a balanced design.

Figure 39.21 Two-Way Design with Significant Interaction

The GLM Procedure

Dependent Variable: Y

Source	DF	Type I SS	Mean Square	F Value	Pr > F
A	2	219.905000	109.952500	165.11	<.0001
B	2	3206.101667	1603.050833	2407.25	<.0001
A*B	4	487.103333	121.775833	182.87	<.0001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
A	2	219.905000	109.952500	165.11	<.0001
B	2	3206.101667	1603.050833	2407.25	<.0001
A*B	4	487.103333	121.775833	182.87	<.0001

The interaction A*B is significant, indicating that the effect of A depends on the level of B. In some cases, you might be interested in looking at the differences between predicted values across A for different levels of B. Winer (1971) calls this the *simple effects* of A. You can compute simple effects with the LSMEANS statement by specifying the SLICE= option. In this case, since the GLM procedure is interactive, you can compute the simple effects of A by submitting the following statements after the preceding statements.

```
   lsmeans A*B / slice=B;
run;
```

The results are shown Figure 39.22. Note that A has a significant effect for B=1 but not for B=2 and B=3.

Figure 39.22 Interaction LS-Means and Simple Effects

The GLM Procedure
Least Squares Means

A	B	Y LSMEAN
1	1	10.8750000
1	2	0.2750000
1	3	0.2750000
2	1	19.4750000
2	2	0.1750000
2	3	-0.7250000
3	1	29.6250000
3	2	0.3750000
3	3	-0.5000000

The GLM Procedure
Least Squares Means

A*B Effect Sliced by B for Y

B	DF	Sum of Squares	Mean Square	F Value	Pr > F
1	2	704.726667	352.363333	529.13	<.0001
2	2	0.080000	0.040000	0.06	0.9418
3	2	2.201667	1.100833	1.65	0.2103

Homogeneity of Variance in One-Way Models

One of the usual assumptions in using the GLM procedure is that the underlying errors are all uncorrelated with homogeneous variances (see page 2487). You can test this assumption in PROC GLM by using the HOVTEST option in the MEANS statement, requesting a *homogeneity of variance* test. This section discusses the computational details behind these tests. Note that the GLM procedure allows homogeneity of variance testing for simple one-way models only. Homogeneity of variance testing for more complex models is a subject of current research.

Bartlett (1937) proposes a test for equal variances that is a modification of the normal-theory likelihood ratio test (the HOVTEST=BARTLETT option). While Bartlett's test has accurate Type I error rates and optimal power when the underlying distribution of the data is normal, it can be very inaccurate if that distribution is even slightly nonnormal (Box 1953). Therefore, Bartlett's test is not recommended for routine use.

An approach that leads to tests that are much more robust to the underlying distribution is to transform the original values of the dependent variable to derive a *dispersion variable* and then to perform analysis of variance on this variable. The significance level for the test of homogeneity of variance is the p-value for the ANOVA F test on the dispersion variable. All of the homogeneity of variance tests available in PROC GLM except Bartlett's use this approach.

Levene's test (Levene 1960) is widely considered to be the standard homogeneity of variance test (the HOVTEST=LEVENE option). Levene's test is of the dispersion-variable-ANOVA form discussed previously, where the dispersion variable is either of the following:

$$\begin{aligned} z_{ij}^2 &= (y_{ij} - \bar{y}_i)^2 \quad \text{(TYPE=SQUARE, the default)} \\ z_{ij} &= |y_{ij} - \bar{y}_i| \quad \text{(TYPE=ABS)} \end{aligned}$$

O'Brien (1979) proposes a test (HOVTEST=OBRIEN) that is basically a modification of Levene's z_{ij}^2, using the dispersion variable

$$z_{ij}^W = \frac{(W + n_i - 2)n_i(y_{ij} - \bar{y}_i)^2 - W(n_i - 1)\sigma_i^2}{(n_i - 1)(n_i - 2)}$$

where n_i is the size of the ith group and σ_i^2 is its sample variance. You can use the W= option in parentheses to tune O'Brien's z_{ij}^W dispersion variable to match the suspected kurtosis of the underlying distribution. The choice of the value of the W= option is rarely critical. By default, W=0.5, as suggested by O'Brien (1979, 1981).

Finally, Brown and Forsythe (1974) suggest using the absolute deviations from the group *medians*:

$$z_{ij}^{\mathrm{BF}} = |y_{ij} - m_i|$$

where m_i is the median of the ith group. You can use the HOVTEST=BF option to specify this test.

Simulation results (Conover, Johnson, and Johnson 1981; Olejnik and Algina 1987) show that, while all of these ANOVA-based tests are reasonably robust to the underlying distribution, the Brown-Forsythe test seems best at providing power to detect variance differences while protecting the Type I error probability. However, since the within-group medians are required for the Brown-Forsythe test, it can be resource intensive if there are very many groups or if some groups are very large.

If one of these tests rejects the assumption of homogeneity of variance, you should use Welch's ANOVA instead of the usual ANOVA to test for differences between group means. However, this conclusion holds only if you use one of the robust homogeneity of variance tests (that is, not for HOVTEST=BARTLETT); even then, any homogeneity of variance test has too little power to be relied upon to always detect when Welch's ANOVA is appropriate. Unless the group variances are extremely different or the number of groups is large, the usual ANOVA test is relatively robust when the groups are all about the same size. As Box (1953) notes, "To make the preliminary test on variances is rather like putting to sea in a rowing boat to find out whether conditions are sufficiently calm for an ocean liner to leave port!"

Example 39.10 illustrates the use of the HOVTEST and WELCH options in the MEANS statement in testing for equal group variances and adjusting for unequal group variances in a one-way ANOVA.

Weighted Means

In previous releases, if you specified a WEIGHT statement and one or more of the multiple comparisons options, PROC GLM estimated the variance of the difference between weighted group means

for group i and j as

$$MSE \times \left(\frac{1}{n_i} + \frac{1}{n_j} \right)$$

where MSE is the (weighted) mean square for error and n_i is the size of group i. This variance is involved in all of the multiple-comparison methods. Beginning with SAS 6.12, the variance of the difference between weighted group means for group i and j is computed as

$$MSE \times \left(\frac{1}{w_i} + \frac{1}{w_j} \right)$$

where w_i is the sum of the weights for the observations in group i.

Construction of Least Squares Means

To construct a least squares mean (LS-mean) for a given level of a given effect, construct a row vector L according to the following rules and use it in an ESTIMATE statement to compute the value of the LS-mean:

1. Set all L_i corresponding to covariates (continuous variables) to their mean value.
2. Consider effects contained by the given effect. Set the L_i corresponding to levels associated with the given level equal to 1. Set all other L_i in these effects equal to 0. (See Chapter 15, "The Four Types of Estimable Functions," for a definition of *containing*.)
3. Consider the given effect. Set the L_i corresponding to the given level equal to 1. Set the L_i corresponding to other levels equal to 0.
4. Consider the effects that contain the given effect. If these effects are not nested within the given effect, then set the L_i corresponding to the given level to $1/k$, where k is the number of such columns. If these effects are nested within the given effect, then set the L_i corresponding to the given level to $1/(k_1 k_2)$, where k_1 is the number of nested levels within this combination of nested effects, and k_2 is the number of such combinations. For L_i corresponding to other levels, use 0.
5. Consider the other effects not yet considered. If there are no nested factors, then set all L_i corresponding to this effect to $1/j$, where j is the number of levels in the effect. If there are nested factors, then set all L_i corresponding to this effect to $1/(j_1 j_2)$, where j_1 is the number of nested levels within a given combination of nested effects and j_2 is the number of such combinations.

The consequence of these rules is that the sum of the Xs within any classification effect is 1. This set of Xs forms a linear combination of the parameters that is checked for estimability before it is evaluated.

For example, consider the following model:

```
proc glm;
   class A B C;
   model Y=A B A*B C Z;
   lsmeans A B A*B C;
run;
```

Assume A has 3 levels, B has 2 levels, and C has 2 levels, and assume that every combination of levels of A and B exists in the data. Assume also that Z is a continuous variable with an average of 12.5. Then the least squares means are computed by the following linear combinations of the parameter estimates:

		A			B		A*B						C		
	μ	1	2	3	1	2	11	12	21	22	31	32	1	2	Z
LSM( )	1	1/3	1/3	1/3	1/2	1/2	1/6	1/6	1/6	1/6	1/6	1/6	1/2	1/2	12.5
LSM(A1)	1	1	0	0	1/2	1/2	1/2	1/2	0	0	0	0	1/2	1/2	12.5
LSM(A2)	1	0	1	0	1/2	1/2	0	0	1/2	1/2	0	0	1/2	1/2	12.5
LSM(A3)	1	0	0	1	1/2	1/2	0	0	0	0	1/2	1/2	1/2	1/2	12.5
LSM(B1)	1	1/3	1/3	1/3	1	0	1/3	0	1/3	0	1/3	0	1/2	1/2	12.5
LSM(B2)	1	1/3	1/3	1/3	0	1	0	1/3	0	1/3	0	1/3	1/2	1/2	12.5
LSM(AB11)	1	1	0	0	1	0	1	0	0	0	0	0	1/2	1/2	12.5
LSM(AB12)	1	1	0	0	0	1	0	1	0	0	0	0	1/2	1/2	12.5
LSM(AB21)	1	0	1	0	1	0	0	0	1	0	0	0	1/2	1/2	12.5
LSM(AB22)	1	0	1	0	0	1	0	0	0	1	0	0	1/2	1/2	12.5
LSM(AB31)	1	0	0	1	1	0	0	0	0	0	1	0	1/2	1/2	12.5
LSM(AB32)	1	0	0	1	0	1	0	0	0	0	0	1	1/2	1/2	12.5
LSM(C1)	1	1/3	1/3	1/3	1/2	1/2	1/6	1/6	1/6	1/6	1/6	1/6	1	0	12.5
LSM(C2)	1	1/3	1/3	1/3	1/2	1/2	1/6	1/6	1/6	1/6	1/6	1/6	0	1	12.5

Setting Covariate Values

By default, all covariate effects are set equal to their mean values for computation of standard LS-means. The AT option in the LSMEANS statement enables you to set the covariates to whatever values you consider interesting.

If there is an effect containing two or more covariates, the AT option sets the effect equal to the product of the individual means rather than the mean of the product (as with standard LS-means calculations). The AT MEANS option leaves covariates equal to their mean values (as with standard LS-means) and incorporates this adjustment to crossproducts of covariates.

As an example, the following is a model with a classification variable A and two continuous variables, x1 and x2:

```
class A;
model y = A x1 x2 x1*x2;
```

The coefficients for the continuous effects with various AT specifications are shown in the following table.

Syntax	x1	x2	x1*x2
`lsmeans A;`	$\overline{x_1}$	$\overline{x_2}$	$\overline{x_1 x_2}$
`lsmeans A / at means;`	$\overline{x_1}$	$\overline{x_2}$	$\overline{x_1} \cdot \overline{x_2}$
`lsmeans A / at x1=1.2;`	1.2	$\overline{x_2}$	$1.2 \cdot \overline{x_2}$
`lsmeans A / at (x1 x2)=(1.2 0.3);`	1.2	0.3	$1.2 \cdot 0.3$

For the first two LSMEANS statements, the A LS-mean coefficient for x1 is $\overline{x_1}$ (the mean of x1) and for x2 is $\overline{x_2}$ (the mean of x2). However, for the first LSMEANS statement, the coefficient for x1*x2 is $\overline{x_1 x_2}$, but for the second LSMEANS statement the coefficient is $\overline{x_1} \cdot \overline{x_2}$. The third LSMEANS statement sets the coefficient for x1 equal to 1.2 and leaves that for x2 at $\overline{x_2}$, and the final LSMEANS statement sets these values to 1.2 and 0.3, respectively.

Even if you specify a WEIGHT variable, the unweighted covariate means are used for the covariate coefficients if there is no AT specification. However, if you also use an AT specification, then weighted covariate means are used for the covariate coefficients for which no explicit AT values are given, or if you specify AT MEANS. Also, observations with missing dependent variables are included in computing the covariate means, unless these observations form a missing cell. You can use the E option in conjunction with the AT option to check that the modified LS-means coefficients are the ones you want.

The AT option is disabled if you specify the BYLEVEL option, in which case the coefficients for the covariates are set equal to their means within each level of the LS-mean effect in question.

Changing the Weighting Scheme

The standard LS-means have equal coefficients across classification effects; however, the OM option in the LSMEANS statement changes these coefficients to be proportional to those found in the input data set. This adjustment is reasonable when you want your inferences to apply to a population that is not necessarily balanced but has the margins observed in the original data set.

In computing the observed margins, PROC GLM uses all observations for which there are no missing independent variables, including those for which there are missing dependent variables. Also, if there is a WEIGHT variable, PROC GLM uses weighted margins to construct the LS-means coefficients. If the analysis data set is balanced or if you specify a simple one-way model, the LS-means will be unchanged by the OM option.

The BYLEVEL option modifies the observed-margins LS-means. Instead of computing the margins across the entire data set, PROC GLM computes separate margins for each level of the LS-mean effect in question. The resulting LS-means are actually equal to raw means in this case. The BYLEVEL option disables the AT option if it is specified.

Note that the MIXED procedure implements a more versatile form of the OM option, enabling you to specifying an alternative data set over which to compute observed margins. If you use the BYLEVEL option, too, then this data set is effectively the "population" over which the population marginal means are computed. See Chapter 56, "The MIXED Procedure," for more information.

You might want to use the E option in conjunction with either the OM or BYLEVEL option to check that the modified LS-means coefficients are the ones you want. It is possible that the modified LS-means are not estimable when the standard ones are, or vice versa.

Multivariate Analysis of Variance

If you fit several dependent variables to the same effects, you might want to make joint tests involving parameters of several dependent variables. Suppose you have p dependent variables, k parameters for each dependent variable, and n observations. The models can be collected into one equation:

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\mathbf{Y}$ is $n \times p$, $\mathbf{X}$ is $n \times k$, $\boldsymbol{\beta}$ is $k \times p$, and $\boldsymbol{\epsilon}$ is $n \times p$. Each of the p models can be estimated and tested separately. However, you might also want to consider the joint distribution and test the p models simultaneously.

For multivariate tests, you need to make some assumptions about the errors. With p dependent variables, there are $n \times p$ errors that are independent across observations but not across dependent variables. Assume

$$\text{vec}(\boldsymbol{\epsilon}) \sim N(\mathbf{0}, \mathbf{I}_n \otimes \boldsymbol{\Sigma})$$

where $\text{vec}(\boldsymbol{\epsilon})$ strings $\boldsymbol{\epsilon}$ out by rows, $\otimes$ denotes Kronecker product multiplication, and $\boldsymbol{\Sigma}$ is $p \times p$. $\boldsymbol{\Sigma}$ can be estimated by

$$\mathbf{S} = \frac{\mathbf{e}'\mathbf{e}}{n-r} = \frac{(\mathbf{Y}-\mathbf{X}\mathbf{b})'(\mathbf{Y}-\mathbf{X}\mathbf{b})}{n-r}$$

where $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{Y}$, r is the rank of the $\mathbf{X}$ matrix, and $\mathbf{e}$ is the matrix of residuals.

If $\mathbf{S}$ is scaled to unit diagonals, the values in $\mathbf{S}$ are called *partial correlations of the Ys adjusting for the Xs*. This matrix can be displayed by PROC GLM if PRINTE is specified as a MANOVA option.

The multivariate general linear hypothesis is written

$$\mathbf{L}\boldsymbol{\beta}\mathbf{M} = 0$$

You can form hypotheses for linear combinations across columns, as well as across rows of $\boldsymbol{\beta}$.

The MANOVA statement of the GLM procedure tests special cases where $\mathbf{L}$ corresponds to Type I, Type II, Type III, or Type IV tests, and $\mathbf{M}$ is the $p \times p$ identity matrix. These tests are joint tests that the given type of hypothesis holds for all dependent variables in the model, and they are often sufficient to test all hypotheses of interest.

Finally, when these special cases are not appropriate, you can specify your own $\mathbf{L}$ and $\mathbf{M}$ matrices by using the CONTRAST statement before the MANOVA statement and the M= specification in the MANOVA statement, respectively. Another alternative is to use a REPEATED statement, which automatically generates a variety of $\mathbf{M}$ matrices useful in repeated measures analysis of variance. See

the section "REPEATED Statement" on page 2481 and the section "Repeated Measures Analysis of Variance" on page 2532 for more information.

One useful way to think of a MANOVA analysis with an **M** matrix other than the identity is as an analysis of a set of transformed variables defined by the columns of the **M** matrix. You should note, however, that PROC GLM always displays the **M** matrix in such a way that the transformed variables are defined by the rows, not the columns, of the displayed **M** matrix.

All multivariate tests carried out by the GLM procedure first construct the matrices **H** and **E** corresponding to the numerator and denominator, respectively, of a univariate F test:

$$\begin{aligned} \mathbf{H} &= \mathbf{M}'(\mathbf{Lb})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{Lb})\mathbf{M} \\ \mathbf{E} &= \mathbf{M}'(\mathbf{Y}'\mathbf{Y} - \mathbf{b}'(\mathbf{X}'\mathbf{X})\mathbf{b})\mathbf{M} \end{aligned}$$

The diagonal elements of **H** and **E** correspond to the hypothesis and error SS for univariate tests. When the **M** matrix is the identity matrix (the default), these tests are for the original dependent variables on the left side of the MODEL statement. When an **M** matrix other than the identity is specified, the tests are for transformed variables defined by the columns of the **M** matrix. These tests can be studied by requesting the SUMMARY option, which produces univariate analyses for each original or transformed variable.

Four multivariate test statistics, all functions of the eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$ (or $(\mathbf{E}+\mathbf{H})^{-1}\mathbf{H}$), are constructed:

- Wilks' lambda = $\det(\mathbf{E})/\det(\mathbf{H}+\mathbf{E})$
- Pillai's trace = $\mathrm{trace}(\mathbf{H}(\mathbf{H}+\mathbf{E})^{-1})$
- Hotelling-Lawley trace = $\mathrm{trace}(\mathbf{E}^{-1}\mathbf{H})$
- Roy's greatest root = λ, largest eigenvalue of $\mathbf{E}^{-1}\mathbf{H}$

By default, all four are reported with p-values based on F approximations, as discussed in the "Multivariate Tests" section in Chapter 4, "Introduction to Regression Procedures." Alternatively, if you specify MSTAT=EXACT in the associated MANOVA or REPEATED statement, p-values for three of the four tests are computed exactly (Wilks' lambda, the Hotelling-Lawley trace, and Roy's greatest root), and the p-values for the fourth (Pillai's trace) are based on an F approximation that is more accurate than the default. See the "Multivariate Tests" section in Chapter 4, "Introduction to Regression Procedures," for more details on the exact calculations.

Repeated Measures Analysis of Variance

When several measurements are taken on the same experimental unit (person, plant, machine, and so on), the measurements tend to be correlated with each other. When the measurements represent qualitatively different things, such as weight, length, and width, this correlation is best taken

into account by use of multivariate methods, such as multivariate analysis of variance. When the measurements can be thought of as responses to levels of an experimental factor of interest, such as time, treatment, or dose, the correlation can be taken into account by performing a repeated measures analysis of variance.

PROC GLM provides both univariate and multivariate tests for repeated measures for one response. For an overall reference on univariate repeated measures, see Winer (1971). The multivariate approach is covered in Cole and Grizzle (1966). For a discussion of the relative merits of the two approaches, see LaTour and Miniard (1983).

Another approach to analysis of repeated measures is via general mixed models. This approach can handle balanced as well as unbalanced or missing within-subject data, and it offers more options for modeling the within-subject covariance. The main drawback of the mixed models approach is that it generally requires iteration and, thus, might be less computationally efficient. For further details on this approach, see Chapter 56, "The MIXED Procedure," and Wolfinger and Chang (1995).

Organization of Data for Repeated Measure Analysis

In order to deal efficiently with the correlation of repeated measures, the GLM procedure uses the multivariate method of specifying the model, even if only a univariate analysis is desired. In some cases, data might already be entered in the univariate mode, with each repeated measure listed as a separate observation along with a variable that represents the experimental unit (subject) on which measurement is taken. Consider the following data set Old:

```
data Old;
   input Subject Group Time y;
datalines;
 1 1 1 15
 1 1 2 19
 1 1 3 25
 2 1 1 21
 2 1 2 18
 2 1 3 17
 1 2 1 14
 1 2 2 12
 1 2 3 16
 2 2 1 11
 2 2 2 20

... more lines ...

10 3 1 14
10 3 2 18
10 3 3 16
;
```

There are three observations for each subject, corresponding to measurements taken at times 1, 2, and 3. These data could be analyzed using the following statements:

```
proc glm data=Old;
   class Group Subject Time;
   model y=Group Subject(Group) Time Group*Time;
   test h=Group e=Subject(Group);
run;
```

However, this analysis assumes subjects' measurements are uncorrelated across time. A repeated measures analysis does not make this assumption. It uses the following data set New:

```
data New;
   input Group y1 y2 y3;
datalines;
1  15 19 25
1  21 18 17
2  14 12 16
2  11 20 21

... more lines ...

3  14 18 16
;
```

In the data set New, the three measurements for a subject are all in one observation. For example, the measurements for subject 1 for times 1, 2, and 3 are 15, 19, and 25, respectively. For these data, the statements for a repeated measures analysis (assuming default options) are

```
proc glm data=New;
   class Group;
   model y1-y3 = Group / nouni;
   repeated Time;
run;
```

To convert the univariate form of repeated measures data to the multivariate form, you can use a program like the following:

```
proc sort data=Old;
   by Group Subject;
run;

data New(keep=y1-y3 Group);
   array yy(3)  y1-y3;
   do Time = 1 to 3;
      set Old;
      by Group Subject;
      yy(Time) = y;
      if last.Subject then return;
   end;
run;
```

Alternatively, you could use PROC TRANSPOSE to achieve the same results with a program like this one:

```
proc sort data=Old;
   by Group Subject;
run;

proc transpose out=New(rename=(_1=y1 _2=y2 _3=y3));
   by Group Subject;
   id Time;
run;
```

See the discussions in *SAS Language Reference: Concepts* for more information about rearrangement of data sets.

Hypothesis Testing in Repeated Measures Analysis

In repeated measures analysis of variance, the effects of interest are as follows:

- between-subject effects (such as GROUP in the previous example)
- within-subject effects (such as TIME in the previous example)
- interactions between the two types of effects (such as GROUP*TIME in the previous example)

Repeated measures analyses are distinguished from MANOVA because of interest in testing hypotheses about the within-subject effects and the within-subject-by-between-subject interactions.

For tests that involve only between-subjects effects, both the multivariate and univariate approaches give rise to the same tests. These tests are provided for all effects in the MODEL statement, as well as for any CONTRASTs specified. The ANOVA table for these tests is labeled "Tests of Hypotheses for Between Subjects Effects" in the PROC GLM results. These tests are constructed by first adding together the dependent variables in the model. Then an analysis of variance is performed on the sum divided by the square root of the number of dependent variables. For example, the statements

```
model y1-y3=group;
repeated time;
```

give a one-way analysis of variance that uses $(Y1 + Y2 + Y3)/\sqrt{3}$ as the dependent variable for performing tests of hypothesis on the between-subject effect GROUP. Tests for between-subject effects are equivalent to tests of the hypothesis $\mathbf{L}\boldsymbol{\beta}\mathbf{M} = 0$, where $\mathbf{M}$ is simply a vector of 1s.

For within-subject effects and for within-subject-by-between-subject interaction effects, the univariate and multivariate approaches yield different tests. These tests are provided for the within-subject effects and for the interactions between these effects and the other effects in the MODEL statement, as well as for any CONTRASTs specified. The univariate tests are displayed in a table labeled "Univariate Tests of Hypotheses for Within Subject Effects." Results for multivariate tests are displayed in a table labeled "Repeated Measures Analysis of Variance."

The multivariate tests provided for within-subjects effects and interactions involving these effects are Wilks' lambda, Pillai's trace, Hotelling-Lawley trace, and Roy's greatest root. For further details on these four statistics, see the "Multivariate Tests" section in Chapter 4, "Introduction to Regression Procedures." As an example, the statements

```
model y1-y3=group;
repeated time;
```

produce multivariate tests for the within-subject effect TIME and the interaction TIME*GROUP.

The multivariate tests for within-subject effects are produced by testing the hypothesis $\mathbf{L}\boldsymbol{\beta}\mathbf{M} = 0$, where the $\mathbf{L}$ matrix is the usual matrix corresponding to the Type I, Type II, Type III, or Type IV hypotheses test, and the $\mathbf{M}$ matrix is one of several matrices depending on the transformation that you specify in the REPEATED statement. The only assumption required for valid tests is that the dependent variables in the model have a multivariate normal distribution with a common covariance matrix across the between-subject effects.

The univariate tests for within-subject effects and interactions involving these effects require some assumptions for the probabilities provided by the ordinary F tests to be correct. Specifically, these tests require certain patterns of covariance matrices, known as Type H covariances (Huynh and Feldt 1970). Data with these patterns in the covariance matrices are said to satisfy the Huynh-Feldt condition. You can test this assumption (and the Huynh-Feldt condition) by applying a sphericity test (Anderson 1958) to any set of variables defined by an orthogonal contrast transformation. Such a set of variables is known as a set of orthogonal components. When you use the PRINTE option in the REPEATED statement, this sphericity test is applied both to the transformed variables defined by the REPEATED statement and to a set of orthogonal components if the specified transformation is not orthogonal. It is the test applied to the orthogonal components that is important in determining whether your data have a Type H covariance structure. When there are only two levels of the within-subject effect, there is only one transformed variable, and a sphericity test is not needed. The sphericity test is labeled "Test for Sphericity" in the output.

If your data satisfy the preceding assumptions, use the usual F tests to test univariate hypotheses for the within-subject effects and associated interactions.

If your data do not satisfy the assumption of Type H covariance, an adjustment to numerator and denominator degrees of freedom can be used. Two such adjustments, based on a degrees-of-freedom adjustment factor known as ϵ (epsilon) (Box 1954), are provided in PROC GLM. Both adjustments estimate ϵ and then multiply the numerator and denominator degrees of freedom by this estimate before determining significance levels for the F tests. Significance levels associated with the adjusted tests are labeled "Adj Pr > F" in the output. The first adjustment, initially proposed for use in data analysis by Greenhouse and Geisser (1959), is labeled "Greenhouse-Geisser Epsilon" and represents the maximum likelihood estimate of Box's ϵ factor. Significance levels associated with adjusted F tests are labeled "G-G" in the output. Huynh and Feldt (1976) have shown that the G-G estimate tends to be biased downward (that is, too conservative), especially for small samples, and they have proposed an alternative estimator that is constructed using unbiased estimators of the numerator and denominator of Box's ϵ. Huynh and Feldt's estimator is labeled "Huynh-Feldt Epsilon" in the PROC GLM output, and the significance levels associated with adjusted F tests are labeled "H-F." Although ϵ must be in the range of 0 to 1, the H-F estimator can be outside this range. When the H-F estimator is greater than 1, a value of 1 is used in all calculations for probabilities, and the H-F probabilities are not adjusted. In summary, if your data do not meet the assumptions,

use adjusted F tests. However, when you strongly suspect that your data might not have Type H covariance, all these univariate tests should be interpreted cautiously. In such cases, you should consider using the multivariate tests instead.

The univariate sums of squares for hypotheses involving within-subject effects can be easily calculated from the **H** and **E** matrices corresponding to the multivariate tests described in the section "Multivariate Analysis of Variance" on page 2531. If the **M** matrix is orthogonal, the univariate sums of squares is calculated as the trace (sum of diagonal elements) of the appropriate **H** matrix; if it is not orthogonal, PROC GLM calculates the trace of the **H** matrix that results from an orthogonal **M** matrix transformation. The appropriate error term for the univariate F tests is constructed in a similar way from the error SSCP matrix and is labeled Error(*factorname*), where *factorname* indicates the **M** matrix that is used in the transformation.

When the design specifies more than one repeated measures factor, PROC GLM computes the **M** matrix for a given effect as the direct (Kronecker) product of the **M** matrices defined by the REPEATED statement if the factor is involved in the effect or as a vector of 1s if the factor is not involved. The test for the main effect of a repeated measures factor is constructed using an **L** matrix that corresponds to a test that the mean of the observation is zero. Thus, the main effect test for repeated measures is a test that the means of the variables defined by the **M** matrix are all equal to zero, while interactions involving repeated measures effects are tests that the between-subjects factors involved in the interaction have no effect on the means of the transformed variables defined by the **M** matrix. In addition, you can specify other **L** matrices to test hypotheses of interest by using the CONTRAST statement, since hypotheses defined by CONTRAST statements are also tested in the REPEATED analysis. To see which combinations of the original variables the transformed variables represent, you can specify the PRINTM option in the REPEATED statement. This option displays the transpose of **M**, which is labeled as M in the PROC GLM results. The tests produced are the same for any choice of transformation (**M**) matrix specified in the REPEATED statement; however, depending on the nature of the repeated measurements being studied, a particular choice of transformation matrix, coupled with the CANONICAL or SUMMARY option, can provide additional insight into the data being studied.

Transformations Used in Repeated Measures Analysis of Variance

As mentioned in the specifications of the REPEATED statement, several different **M** matrices can be generated automatically, based on the transformation that you specify in the REPEATED statement. Remember that both the univariate and multivariate tests that PROC GLM performs are unaffected by the choice of transformation; the choice of transformation is important only when you are trying to study the nature of a repeated measures effect, particularly with the CANONICAL and SUMMARY options. If one of these matrices does not meet your needs for a particular analysis, you might want to use the M= option in the MANOVA statement to perform the tests of interest.

The following sections describe the transformations available in the REPEATED statement, provide an example of the **M** matrix that is produced, and give guidelines for the use of the transformation. As in the PROC GLM output, the displayed matrix is labeled M. This is the $\mathbf{M}'$ matrix.

CONTRAST Transformation

This is the default transformation used by the REPEATED statement. It is useful when one level of the repeated measures effect can be thought of as a control level against which the others are compared. For example, if five drugs are administered to each of several animals and the first drug is a control or placebo, the statements

```
proc glm;
   model d1-d5= / nouni;
   repeated drug 5 contrast(1) / summary printm;
run;
```

produce the following $\mathbf{M}$ matrix:

$$\mathbf{M} = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 \\ -1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

When you examine the analysis of variance tables produced by the SUMMARY option, you can tell which of the drugs differed significantly from the placebo.

POLYNOMIAL Transformation

This transformation is useful when the levels of the repeated measure represent quantitative values of a treatment, such as dose or time. If the levels are unequally spaced, *level values* can be specified in parentheses after the number of levels in the REPEATED statement. For example, if five levels of a drug corresponding to 1, 2, 5, 10, and 20 milligrams are administered to different treatment groups, represented by the variable group, the statements

```
proc glm;
   class group;
   model r1-r5=group / nouni;
   repeated dose 5 (1 2 5 10 20) polynomial / summary printm;
run;
```

produce the following $\mathbf{M}$ matrix:

$$\mathbf{M} = \begin{bmatrix} -0.4250 & -0.3606 & -0.1674 & 0.1545 & 0.7984 \\ 0.4349 & 0.2073 & -0.3252 & -0.7116 & 0.3946 \\ -0.4331 & 0.1366 & 0.7253 & -0.5108 & 0.0821 \\ 0.4926 & -0.7800 & 0.3743 & -0.0936 & 0.0066 \end{bmatrix}$$

The SUMMARY option in this example provides univariate ANOVAs for the variables defined by the rows of this $\mathbf{M}$ matrix. In this case, they represent the linear, quadratic, cubic, and quartic trends for dose and are labeled dose_1, dose_2, dose_3, and dose_4, respectively.

HELMERT Transformation

Since the Helmert transformation compares a level of a repeated measure to the mean of subsequent levels, it is useful when interest lies in the point at which responses cease to change. For example, if four levels of a repeated measures factor represent responses to treatments administered over time to males and females, the statements

```
proc glm;
   class sex;
   model resp1-resp4=sex / nouni;
   repeated trtmnt 4 helmert / canon printm;
run;
```

produce the following $\mathbf{M}$ matrix:

$$\mathbf{M} = \begin{bmatrix} 1 & -0.33333 & -0.33333 & -0.33333 \\ 0 & 1 & -0.50000 & -0.50000 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

MEAN Transformation

This transformation can be useful in the same types of situations in which the CONTRAST transformation is useful. If you substitute the following statement for the REPEATED statement shown in the CONTRAST Transformation section,

```
repeated drug 5 mean / printm;
```

the following $\mathbf{M}$ matrix is produced:

$$\mathbf{M} = \begin{bmatrix} 1 & -0.25 & -0.25 & -0.25 & -0.25 \\ -0.25 & 1 & -0.25 & -0.25 & -0.25 \\ -0.25 & -0.25 & 1 & -0.25 & -0.25 \\ -0.25 & -0.25 & -0.25 & 1 & -0.25 \end{bmatrix}$$

As with the CONTRAST transformation, if you want to omit a level other than the last, you can specify it in parentheses after the keyword MEAN in the REPEATED statement.

PROFILE Transformation

When a repeated measure represents a series of factors administered over time, but a polynomial response is unreasonable, a profile transformation might prove useful. As an example, consider a training program in which four different methods are employed to teach students at several different schools. The repeated measure is the score on tests administered after each of the methods is completed. The statements

```
proc glm;
   class school;
   model t1-t4=school / nouni;
   repeated method 4 profile / summary nom printm;
run;
```

produce the following **M** matrix:

$$\mathbf{M} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

To determine the point at which an improvement in test scores takes place, you can examine the analyses of variance for the transformed variables representing the differences between adjacent tests. These analyses are requested by the SUMMARY option in the REPEATED statement, and the variables are labeled METHOD.1, METHOD.2, and METHOD.3.

Random-Effects Analysis

When some model effects are random (that is, assumed to be sampled from a normal population of effects), you can specify these effects in the RANDOM statement in order to compute the expected values of mean squares for various model effects and contrasts and, optionally, to perform random-effects analysis of variance tests.

PROC GLM versus PROC MIXED for Random-Effects Analysis

Other SAS procedures that can be used to analyze models with random effects include the MIXED and VARCOMP procedures. Note that, for these procedures, the random-effects specification is an integral part of the model, affecting how both random and fixed effects are fit; for PROC GLM, the random effects are treated in a *post hoc* fashion after the complete fixed-effect model is fit. This distinction affects other features in the GLM procedure, such as the results of the LSMEANS and ESTIMATE statements. These features assume that all effects are fixed, so that all tests and estimability checks for these statements are based on a fixed-effects model, even when you use a RANDOM statement. Standard errors for estimates and LS-means based on the fixed-effects model might be significantly smaller than those based on a true random-effects model; in fact, some functions that are estimable under a true random-effects model might not even be estimable under the fixed-effects model. Therefore, you should use the MIXED procedure to compute tests involving these features that take the random effects into account; see Chapter 56, "The MIXED Procedure," for more information.

Note that, for balanced data, the test statistics computed when you specify the TEST option in the RANDOM statement have an exact F distribution only when the design is balanced; for unbalanced designs, the p values for the F tests are approximate. For balanced data, the values obtained by PROC GLM and PROC MIXED agree; for unbalanced data, they usually do not.

Computation of Expected Mean Squares for Random Effects

The RANDOM statement in PROC GLM declares one or more effects in the model to be random rather than fixed. By default, PROC GLM displays the coefficients of the expected mean squares for all terms in the model. In addition, when you specify the TEST option in the RANDOM statement,

the procedure determines what tests are appropriate and provides F ratios and probabilities for these tests.

The expected mean squares are computed as follows. Consider the model

$$Y = X_0\boldsymbol{\beta}_0 + X_1\boldsymbol{\beta}_1 + \cdots + X_k\boldsymbol{\beta}_k + \boldsymbol{\epsilon}$$

where $\boldsymbol{\beta}_0$ represents the fixed effects and $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \cdots, \boldsymbol{\epsilon}$ represent the random effects. Random effects are assumed to be normally and independently distributed. For any $\mathbf{L}$ in the row space of $\mathbf{X} = (X_0 \mid X_1 \mid X_2 \mid \cdots \mid X_k)$, the expected value of the sum of squares for $\mathbf{L}\beta$ is

$$E(\mathrm{SS}_L) = \boldsymbol{\beta}_0'\mathbf{C}_0'\mathbf{C}_0\boldsymbol{\beta}_0 + \mathrm{SSQ}(\mathbf{C}_1)\sigma_1^2 + \mathrm{SSQ}(\mathbf{C}_2)\sigma_2^2 + \cdots + \mathrm{SSQ}(\mathbf{C}_k)\sigma_k^2 + \mathrm{rank}(\mathbf{L})\sigma_\epsilon^2$$

where $\mathbf{C}$ is of the same dimensions as $\mathbf{L}$ and is partitioned as the $\mathbf{X}$ matrix. In other words,

$$\mathbf{C} = (\mathbf{C}_0 \mid \mathbf{C}_1 \mid \cdots \mid \mathbf{C}_k)$$

Furthermore, $\mathbf{C} = \mathbf{ML}$, where $\mathbf{M}$ is the inverse of the lower triangular Cholesky decomposition matrix of $\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}'$. SSQ($\mathbf{A}$) is defined as tr($\mathbf{A}'\mathbf{A}$).

For the model in the following MODEL statement

```
model Y=A B(A) C A*C;
random B(A);
```

with B(A) declared as random, the expected mean square of each effect is displayed as

$$\mathrm{Var(Error)} + \mathit{constant} \times \mathrm{Var(B(A))} + Q(\mathsf{A}, \mathsf{C}, \mathsf{A} * \mathsf{C})$$

If any fixed effects appear in the expected mean square of an effect, the letter Q followed by the list of fixed effects in the expected value is displayed. The actual numeric values of the quadratic form ($\mathbf{Q}$ matrix) can be displayed using the Q option.

To determine appropriate means squares for testing the effects in the model, the TEST option in the RANDOM statement performs the following steps:

1. First, it forms a matrix of coefficients of the expected mean squares of those effects that were declared to be random.
2. Next, for each effect in the model, it determines the combination of these expected mean squares that produce an expectation that includes all the terms in the expected mean square of the effect of interest except the one corresponding to the effect of interest. For example, if the expected mean square of an effect A*B is

   $$\mathrm{Var(Error)} + 3 \times \mathrm{Var(A)} + \mathrm{Var(A} * \mathrm{B)}$$

   PROC GLM determines the combination of other expected mean squares in the model that has expectation

   $$\mathrm{Var(Error)} + 3 \times \mathrm{Var(A)}$$

3. If the preceding criterion is met by the expected mean square of a single effect in the model (as is often the case in balanced designs), the F test is formed directly. In this case, the mean square of the effect of interest is used as the numerator, the mean square of the single effect with an expected mean square that satisfies the criterion is used as the denominator, and the degrees of freedom for the test are simply the usual model degrees of freedom.

4. When more than one mean square must be combined to achieve the appropriate expectation, an approximation is employed to determine the appropriate degrees of freedom (Satterthwaite 1946). When effects other than the effect of interest are listed after the Q in the output, tests of hypotheses involving the effect of interest are not valid unless all other fixed effects involved in it are assumed to be zero. When tests such as these are performed by using the TEST option in the RANDOM statement, a note is displayed reminding you that further assumptions are necessary for the validity of these tests. Remember that although the tests are not valid unless these assumptions are made, this does not provide a basis for these assumptions to be true. The particulars of a given experiment must be examined to determine whether the assumption is reasonable.

See Goodnight and Speed (1978), Milliken and Johnson (1984, Chapters 22 and 23), and Hocking (1985) for further theoretical discussion.

Sum-to-Zero Assumptions

The formulation and parameterization of the expected mean squares for random effects in mixed models are ongoing items of controversy in the statistical literature. Confusion arises over whether or not to assume that terms involving fixed effects sum to zero. Cornfield and Tukey (1956), Winer (1971), and others assume that they do sum to zero; Searle (1971), Hocking (1973), and others (including PROC GLM) do not.

Different assumptions about these sum-to-zero constraints can lead to different expected mean squares for certain terms, and hence to different F and p values.

For arguments in favor of not assuming that terms involving fixed effects sum to zero, see Section 9.7 of Searle (1971) and Sections 1 and 4 of McLean, Sanders, and Stroup (1991). Other references are Hartley and Searle (1969) and Searle, Casella, and McCulloch (1992).

Computing Type I, II, and IV Expected Mean Squares

When you use the RANDOM statement, by default the GLM procedure produces the Type III expected mean squares for model effects and for contrasts specified before the RANDOM statement. In order to obtain expected values for other types of mean squares, you need to specify which types of mean squares are of interest in the MODEL statement. For example, in order to obtain the Type IV expected mean squares for effects in the RANDOM and CONTRAST statements, specify the SS4 option in the MODEL statement. If you want both Type III and Type IV expected mean squares, specify both the SS3 and SS4 options in the MODEL statement. Since the estimable function basis is not automatically calculated for Type I and Type II SS, the E1 (for Type I) or E2 (for Type II) option must be specified in the MODEL statement in order for the RANDOM statement to produce

the expected mean squares for the Type I or Type II sums of squares. Note that it is important to list the fixed effects first in the MODEL statement when requesting the Type I expected mean squares.

For example, suppose you have a two-way design with factors A and B in which the main effect for B and the interaction are random. In order to compute the Type III expected mean squares (in addition to the fixed-effect analysis), you can use the following statements:

```
proc glm;
   class A B;
   model Y = A B A*B;
   random B A*B;
run;
```

Suppose you use the SS4 option in the MODEL statement, as follows:

```
proc glm;
   class A B;
   model Y = A B A*B / ss4;
   random B A*B;
run;
```

Then only the Type IV expected mean squares are computed (as well as the Type IV fixed-effect tests). For the Type I expected mean squares, you can use the following statements:

```
proc glm;
   class A B;
   model Y = A B A*B / e1;
   random B A*B;
run;
```

For each of these cases, in order to perform random-effect analysis of variance tests for each effect specified in the model, you need to specify the TEST option in the RANDOM statement, as follows:

```
proc glm;
   class A B;
   model Y = A B A*B;
   random B A*B / test;
run;
```

The GLM procedure automatically determines the appropriate error term for each test, based on the expected mean squares.

Missing Values

For an analysis involving one dependent variable, PROC GLM uses an observation if values are nonmissing for that dependent variable and all the classification variables.

For an analysis involving multiple dependent variables without the MANOVA or REPEATED statement, or without the MANOVA option in the PROC GLM statement, a missing value in one depen-

dent variable does not eliminate the observation from the analysis of other nonmissing dependent variables. On the other hand, for an analysis with the MANOVA or REPEATED statement, or with the MANOVA option in the PROC GLM statement, PROC GLM uses an observation if values are nonmissing for all dependent variables and all the variables used in independent effects.

During processing, the GLM procedure groups the dependent variables by their pattern of missing values across observations so that sums and crossproducts can be collected in the most efficient manner.

If your data have different patterns of missing values among the dependent variables, interactivity is disabled. This can occur when some of the variables in your data set have missing values and either of the following conditions obtain:

- You do not use the MANOVA option in the PROC GLM statement.
- You do not use a MANOVA or REPEATED statement before the first RUN statement.

Note that the REG procedure handles missing values differently in this case; see Chapter 73, "The REG Procedure," for more information.

Computational Resources

Memory

For large problems, most of the memory resources are required for holding the $\mathbf{X}'\mathbf{X}$ matrix of the sums and crossproducts. The section "Parameterization of PROC GLM Models" on page 2491 describes how columns of the $\mathbf{X}$ matrix are allocated for various types of effects. For each level that occurs in the data for a combination of classification variables in a given effect, a row and a column for $\mathbf{X}'\mathbf{X}$ are needed.

The following example illustrates the calculation. Suppose A has 20 levels, B has 4 levels, and C has 3 levels. Then consider the model

```
proc glm;
  class A B C;
  model Y1 Y2 Y3=A B A*B C A*C B*C A*B*C X1 X2;
run;
```

The $\mathbf{X}'\mathbf{X}$ matrix (bordered by $\mathbf{X}'\mathbf{Y}$ and $\mathbf{Y}'\mathbf{Y}$) can have as many as 425 rows and columns:

1	for the intercept term
20	for A
4	for B
80	for A*B
3	for C

60	for A*C
12	for B*C
240	for A*B*C
2	for X1 and X2 (continuous variables)
3	for Y1, Y2, and Y3 (dependent variables)

The matrix has 425 rows and columns only if all combinations of levels occur for each effect in the model. For m rows and columns, $8m^2$ bytes are needed for crossproducts. In this case, $8 \cdot 425^2 = 1,445,000$ bytes, or about $1,445,000/1024 = 1411K$.

The required memory grows as the square of the number of columns of $\mathbf{X}$; most of the memory is for the A*B*C interaction. Without A*B*C, you have 185 columns and need 268K for $\mathbf{X}'\mathbf{X}$. Without either A*B*C or A*B, you need 86K. If A is recoded to have 10 levels, then the full model has only 220 columns and requires 378K.

The second time that a large amount of memory is needed is when Type III, Type IV, or contrast sums of squares are being calculated. This memory requirement is a function of the number of degrees of freedom of the model being analyzed and the maximum degrees of freedom for any single source. Let Rank equal the sum of the model degrees of freedom, MaxDF be the maximum number of degrees of freedom for any single source, and N_y be the number of dependent variables in the model. Then the memory requirement in bytes is 8 times

$$\begin{aligned} N_y \times \text{Rank} \quad &+ \quad (\text{Rank} \times (\text{Rank} + 1))\,/2 \\ &+ \quad \text{MaxDF} \times \text{Rank} \\ &+ \quad (\text{MaxDF} \times (\text{MaxDF} + 1))\,/2 \\ &+ \quad \text{MaxDF} \times N_y \end{aligned}$$

The first two components of this formula are for the estimable model coefficients and their variance; the rest correspond to $\mathbf{L}$, $\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}'$, and $\mathbf{Lb}$ in the computation of $\text{SS}(\mathbf{L}\boldsymbol{\beta} = 0) = (\mathbf{Lb})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{Lb})$. If the operating system enables SAS to run parallel computational threads on multiple CPUs, then GLM will attempt to allocate another $8 \times \text{Rank} \times \text{Rank}$ bytes in order to perform these calculations in parallel. If this much memory is not available, then the estimability calculations are performed in a single thread.

Unfortunately, these quantities are not available when the $\mathbf{X}'\mathbf{X}$ matrix is being constructed, so PROC GLM might occasionally request additional memory even after you have increased the memory allocation available to the program.

If you have a large model that exceeds the memory capacity of your computer, these are your options:

- Eliminate terms, especially high-level interactions.
- Reduce the number of levels for variables with many levels.
- Use the ABSORB statement for parts of the model that are large.

- Use the REPEATED statement for repeated measures variables.
- Use PROC ANOVA or PROC REG rather than PROC GLM, if your design allows.

A related limitation is that for any model effect involving classification variables (interactions as well as main effects), the number of levels cannot exceed 32,767. This is because GLM internally indexes effect levels with signed short (16-bit) integers, for which the maximum value is $2^{15} - 1 = 32,767$.

CPU Time

Typically, if the GLM procedure requires a lot of CPU time, it will be for one of several reasons. Suppose that the input data has n rows (observations) and the model has E effects that together produce a design matrix $\mathbf{X}$ with m columns. Then if m or n is relatively large, the procedure might spend a lot of time in any of the following areas:

- collecting the sums of squares and crossproducts
- solving the normal equations
- computing the Type III tests

The time required for collecting sums and crossproducts is difficult to calculate because it is a complicated function of the model. The worst case occurs if all columns are continuous variables, involving $nm^2/2$ multiplications and additions. If the columns are levels of a classification, then only m sums might be needed, but a significant amount of time might be spent in look-up operations. Solving the normal equations requires time for approximately $m^3/2$ multiplications and additions, and the number of operations required to compute the Type III tests is also proportional to both E and m^3.

Suppose that you know that Type IV sums of squares are appropriate for the model you are analyzing (for example, if your design has no missing cells). You can specify the SS4 option in your MODEL statement, which saves CPU time by requesting the Type IV sums of squares instead of the more computationally burdensome Type III sums of squares. This proves especially useful if you have a factor in your model that has many levels and is involved in several interactions.

If the operating system enables SAS to run parallel computational threads on multiple CPUs, then both the solution of the normal equations and the computation of Type III tests can take advantage of this to reduce the computational time for large models. In solving the normal equations, the fundamental row sweep operations (Goodnight 1979) are performed in parallel. In computing the Type III tests, both the orthogonalization for the estimable functions and the sums of squares calculation have been parallelized (if there is sufficient memory).

The reduction in computational time due to parallel processing depends on the size of the model, the number of processors, and the parallel architecture of the operating system. If the model is large enough that the overwhelming proportion of CPU time for the procedure is accounted for in solving the normal equations and/or computing the Type III tests, then you can expect a reduction in computational time approximately inversely proportional to the number of CPUs. However, as you

increase the number of processors, the efficiency of this scaling can be reduced by several effects. One mitigating factor is a purely mathematical one known as "Amdahl's law," which is related to the fact that only part of the processing time for the procedure can be parallelized. Even taking Amdahl's law into account, the parallelization efficiency can be reduced by cache effects related to how fast the multiple processors can access memory. See Cohen (2002) for a discussion of these issues. For additional information about parallel processing in SAS, see the chapter on "Support for Parallel Processing" in *SAS Language Reference: Concepts.*

Computational Method

Let $\mathbf{X}$ represent the $n \times p$ design matrix and $\mathbf{Y}$ the $n \times 1$ vector of dependent variables. (See the section "Parameterization of PROC GLM Models" on page 2491 for information about how $\mathbf{X}$ is formed from your model specification.)

The normal equations $\mathbf{X}'\mathbf{X}\boldsymbol{\beta} = \mathbf{X}'\mathbf{Y}$ are solved using a modified sweep routine that produces a generalized inverse $(\mathbf{X}'\mathbf{X})^-$ and a solution $\mathbf{b} = (\mathbf{X}'\mathbf{X})^-\mathbf{X}'\mathbf{y}$. The modification is that rows and columns corresponding to diagonal elements that are found during sweeping to be zero (or within the expected level of numerical error of zero) are zeroed out. The $(\mathbf{X}'\mathbf{X})^-$ produced by this procedure satisfies the following two equations:

$$\begin{aligned}(\mathbf{X}'\mathbf{X})\ (\mathbf{X}'\mathbf{X})^-(\mathbf{X}'\mathbf{X}) &= (\mathbf{X}'\mathbf{X})\\ (\mathbf{X}'\mathbf{X})^-(\mathbf{X}'\mathbf{X})\ (\mathbf{X}'\mathbf{X})^- &= (\mathbf{X}'\mathbf{X})^-\end{aligned}$$

Pringle and Rayner (1971) call a generalized inverse with these characteristics a g_2-inverse, and this is the term usually used in SAS documentation and output. Urquardt (1968) uses the term *reflexive g-inverse* to emphasize that $(\mathbf{X}'\mathbf{X})^-$ is a generalized inverse of $\mathbf{X}'\mathbf{X}$ in the same way that $\mathbf{X}'\mathbf{X}$ is a generalized inverse of $(\mathbf{X}'\mathbf{X})^-$. Note that a g_2-inverse is not necessarily unique: if $\mathbf{X}'\mathbf{X}$ is singular, then sweeping the matrix in a different order will result in a different g_2-inverse that also satisfies the two preceding equations.

For each effect in the model, a matrix $\mathbf{L}$ is computed such that the rows of $\mathbf{L}$ are estimable. Tests of the hypothesis $\mathbf{L}\boldsymbol{\beta} = 0$ are then made by first computing

$$\mathrm{SS}(\mathbf{L}\boldsymbol{\beta} = 0) = (\mathbf{L}\mathbf{b})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^-\mathbf{L}')^{-1}(\mathbf{L}\mathbf{b})$$

and then computing the associated F value by using the mean squared error.

Output Data Sets

OUT= Data Set Created by the OUTPUT Statement

The OUTPUT statement produces an output data set that contains the following:

- all original data from the SAS data set input to PROC GLM
- the new variables corresponding to the diagnostic measures specified with statistics keywords in the OUTPUT statement (PREDICTED=, RESIDUAL=, and so on)

With multiple dependent variables, a name can be specified for any of the diagnostic measures for each of the dependent variables in the order in which they occur in the MODEL statement.

For example, suppose that the input data set A contains the variables y1, y2, y3, x1, and x2. Then you can use the following statements:

```
proc glm data=A;
   model y1 y2 y3=x1;
   output out=out p=y1hat y2hat y3hat
                  r=y1resid lclm=y1lcl uclm=y1ucl;
run;
```

The output data set out contains y1, y2, y3, x1, x2, y1hat, y2hat, y3hat, y1resid, y1lcl, and y1ucl. The variable x2 is output even though it is not used by PROC GLM. Although predicted values are generated for all three dependent variables, residuals are output for only the first dependent variable.

When any independent variable in the analysis (including all class variables) is missing for an observation, then all new variables that correspond to diagnostic measures are missing for the observation in the output data set.

When a dependent variable in the analysis is missing for an observation, then some new variables that correspond to diagnostic measures are missing for the observation in the output data set, and some are still available. Specifically, in this case, the new variables that correspond to COOKD, COVRATIO, DFFITS, PRESS, R, RSTUDENT, STDR, and STUDENT are missing in the output data set. The variables corresponding to H, LCL, LCLM, P, STDI, STDP, UCL, and UCLM are not missing.

OUT= Data Set Created by the LSMEANS Statement

The OUT= option in the LSMEANS statement produces an output data set that contains the following:

- the unformatted values of each classification variable specified in any effect in the LSMEANS statement
- a new variable, LSMEAN, which contains the LS-mean for the specified levels of the classification variables

- a new variable, STDERR, which contains the standard error of the LS-mean

The variances and covariances among the LS-means are also output when the COV option is specified along with the OUT= option. In this case, only one effect can be specified in the LSMEANS statement, and the following variables are included in the output data set:

- new variables, COV1, COV2, ..., COV*n*, where *n* is the number of levels of the effect specified in the LSMEANS statement. These variables contain the covariances of each LS-mean with every other LS-mean.
- a new variable, NUMBER, which provides an index for each observation to identify the covariances that correspond to that observation. The covariances for the observation with NUMBER equal to *n* can be found in the variable COV*n*.

OUTSTAT= Data Set

The OUTSTAT= option in the PROC GLM statement produces an output data set that contains the following:

- the BY variables, if any
- _TYPE_, a new character variable. _TYPE_ can take the values 'SS1', 'SS2', 'SS3', 'SS4', or 'CONTRAST', corresponding to the various types of sums of squares generated, or the values 'CANCORR', 'STRUCTUR', or 'SCORE', if a canonical analysis is performed through the MANOVA statement and no M= matrix is specified.
- _SOURCE_, a new character variable. For each observation in the data set, _SOURCE_ contains the name of the model effect or contrast label from which the corresponding statistics are generated.
- _NAME_, a new character variable. For each observation in the data set, _NAME_ contains the name of one of the dependent variables in the model or, in the case of canonical statistics, the name of one of the canonical variables (CAN1, CAN2, and so forth).
- four new numeric variables: SS, DF, F, and PROB, containing sums of squares, degrees of freedom, *F* values, and probabilities, respectively, for each model or contrast sum of squares generated in the analysis. For observations resulting from canonical analyses, these variables have missing values.
- if there is more than one dependent variable, then variables with the same names as the dependent variables represent the following:
  - for _TYPE_=SS1, SS2, SS3, SS4, or CONTRAST, the crossproducts of the hypothesis matrices
  - for _TYPE_=CANCORR, canonical correlations for each variable
  - for _TYPE_=STRUCTUR, coefficients of the total structure matrix
  - for _TYPE_=SCORE, raw canonical score coefficients

The output data set can be used to perform special hypothesis tests (for example, with the IML procedure in SAS/IML software), to reformat output, to produce canonical variates (through the SCORE procedure), or to rotate structure matrices (through the FACTOR procedure).

Displayed Output

The GLM procedure produces the following output by default:

- The overall analysis-of-variance table breaks down the Total Sum of Squares for the dependent variable into the portion attributed to the Model and the portion attributed to Error.
- The Mean Square term is the Sum of Squares divided by the degrees of freedom (DF).
- The Mean Square for Error is an estimate of σ^2, the variance of the true errors.
- The F Value is the ratio produced by dividing the Mean Square for the Model by the Mean Square for Error. It tests how well the model as a whole (adjusted for the mean) accounts for the dependent variable's behavior. An F test is a joint test to determine that all parameters except the intercept are zero.
- A small significance probability, Pr > F, indicates that some linear function of the parameters is significantly different from zero.
- R-Square, R^2, measures how much variation in the dependent variable can be accounted for by the model. R^2, which can range from 0 to 1, is the ratio of the sum of squares for the model to the corrected total sum of squares. In general, the larger the value of R^2, the better the model's fit.
- Coeff Var, the coefficient of variation, which describes the amount of variation in the population, is 100 times the standard deviation estimate of the dependent variable, Root MSE (Mean Square for Error), divided by the Mean. The coefficient of variation is often a preferred measure because it is unitless.
- Root MSE estimates the standard deviation of the dependent variable (or equivalently, the error term) and equals the square root of the Mean Square for Error.
- Mean is the sample mean of the dependent variable.

These tests are used primarily in analysis-of-variance applications:

- The Type I SS (sum of squares) measures incremental sums of squares for the model as each variable is added.
- The Type III SS is the sum of squares for a balanced test of each effect, adjusted for every other effect.

These items are used primarily in regression applications:

- The Estimates for the model Parameters (the intercept and the coefficients)
- t Value is the Student's t value for testing the null hypothesis that the parameter (if it is estimable) equals zero.
- The significance level, Pr > |t|, is the probability of getting a larger value of t if the parameter is truly equal to zero. A very small value for this probability leads to the conclusion that the independent variable contributes significantly to the model.
- The Standard Error is the square root of the estimated variance of the estimate of the true value of the parameter.

Other portions of output are discussed in the following examples.

ODS Table Names

PROC GLM assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 39.9. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 39.9 ODS Tables Produced by PROC GLM

ODS Table Name	Description	Statement / Option
Aliasing	Type 1,2,3,4 aliasing structure	MODEL / (E1 E2 E3 or E4) and ALIASING
AltErrContrasts	ANOVA table for contrasts with alternative error	CONTRAST / E=
AltErrTests	ANOVA table for tests with alternative error	TEST / E=
Bartlett	Bartlett's homogeneity of variance test	MEANS / HOVTEST=BARTLETT
CLDiffs	Multiple comparisons of pairwise differences	MEANS / CLDIFF or DUNNETT or (Unequal cells and not LINES)
CLDiffsInfo	Information for multiple comparisons of pairwise differences	MEANS / CLDIFF or DUNNETT or (Unequal cells and not LINES)
CLMeans	Multiple comparisons of means with confidence/comparison interval	MEANS / CLM
CLMeansInfo	Information for multiple comparison of means with confidence/comparison interval	MEANS / CLM
CanAnalysis	Canonical analysis	(MANOVA or REPEATED) / CANONICAL

Table 39.9 *continued*

ODS Table Name	Description	Statement / Option
CanCoef	Canonical coefficients	(MANOVA or REPEATED) / CANONICAL
CanStructure	Canonical structure	(MANOVA or REPEATED) / CANONICAL
CharStruct	Characteristic roots and vectors	(MANOVA / not CANONICAL) or (REPEATED / PRINTRV)
ClassLevels	Classification variable levels	CLASS statement
ContrastCoef	**L** matrix for contrast	CONTRAST / EST
Contrasts	ANOVA table for contrasts	CONTRAST statement
DependentInfo	Simultaneously analyzed dependent variables	default when there are multiple dependent variables with different patterns of missing values
Diff	PDiff matrix of least squares means	LSMEANS / PDIFF=ALL and more than two LS-means
Epsilons	Greenhouse-Geisser and Huynh-Feldt epsilons	REPEATED statement
ErrorSSCP	Error SSCP matrix	(MANOVA or REPEATED) / PRINTE
EstFunc	Type 1,2,3,4 estimable functions	MODEL / (E1 E2 E3 or E4)
Estimates	Estimate statement results	ESTIMATE statement
ExpectedMeanSquares	Expected mean squares	RANDOM statement
FitStatistics	R-Square, Coeff Var, Root MSE, and dependent mean	default
GAliasing	General form of aliasing structure	MODEL / E and ALIASING
GEstFunc	General form of estimable functions	MODEL / E
HOVFTest	Homogeneity of variance ANOVA	MEANS / HOVTEST
HypothesisSSCP	Hypothesis SSCP matrix	(MANOVA or REPEATED) / PRINTH
InvXPX	inv(**X**'**X**) matrix	MODEL / INVERSE
LSMeanCL	Confidence interval for LS-means	LSMEANS / CL
LSMeanCoef	Coefficients of least squares means	LSMEANS / E
LSMeanDiffCL	Confidence interval for LS-mean differences	LSMEANS / PDIFF and CL
LSMeans	Least squares means	LSMEANS statement
LSMLines	Least squares means comparison lines	LSMEANS / PDIFF=ALL LINES
MANOVATransform	Multivariate transformation matrix	MANOVA / M=

Table 39.9 *continued*

ODS Table Name	Description	Statement / Option
MCLines	Multiple comparisons LINES output	MEANS / LINES or ((DUNCAN or WALLER or SNK or REGWQ) and not (CLDIFF or CLM)) or (Equal cells and not CLDIFF)
MCLinesInfo	Information for multiple comparison LINES output	MEANS / LINES or ((DUNCAN or WALLER or SNK or REGWQ) and not (CLDIFF or CLM)) or (Equal cells and not CLDIFF)
MCLinesRange	Ranges for multiple range MC tests	MEANS / LINES or ((DUNCAN or WALLER or SNK or REGWQ) and not (CLDIFF or CLM)) or (Equal cells and not CLDIFF)
MatrixRepresentation	**X** matrix element representation	as needed for other options
Means	Group means	MEANS statement
ModelANOVA	ANOVA for model terms	default
MultStat	Multivariate tests	MANOVA statement
NObs	Number of observations	default
OverallANOVA	Overall ANOVA	default
ParameterEstimates	Estimated linear model coefficients	MODEL / SOLUTION
PartialCorr	Partial correlation matrix	(MANOVA or REPEATED) / PRINTE
PredictedInfo	Predicted values info	MODEL / PREDICTED or CLM or CLI
PredictedValues	Predicted values	MODEL / PREDICTED or CLM or CLI
QForm	Quadratic form for expected mean squares	RANDOM / Q
RandomModelANOVA	Random-effect tests	RANDOM / TEST
RepeatedLevelInfo	Correspondence between dependents and repeated measures levels	REPEATED statement
RepeatedTransform	Repeated measures transformation matrix	REPEATED / PRINTM
SimDetails	Details of difference quantile simulation	LSMEANS / ADJUST=SIMULATE(REPORT)
SimResults	Evaluation of difference quantile simulation	LSMEANS / ADJUST=SIMULATE(REPORT)
SlicedANOVA	Sliced-effect ANOVA table	LSMEANS / SLICE
Sphericity	Sphericity tests	REPEATED / PRINTE
Tests	Summary ANOVA for specified MANOVA H= effects	MANOVA / H= SUMMARY

Table 39.9 *continued*

ODS Table Name	Description	Statement / Option
Tolerances	**X'X** tolerances	MODEL / TOLERANCE
Welch	Welch's ANOVA	MEANS / WELCH
XPX	**X'X** matrix	MODEL / XPX

With the PDIFF or TDIFF option in the LSMEANS statement, the p/t-values for differences are displayed in columns of the LSMeans table for PDIFF/TDIFF=CONTROL or PDIFF/TDIFF=ANOM, and for PDIFF/TDIFF=ALL when there are only two LS-means. Otherwise (for PDIFF/TDIFF=ALL when there are more than two LS-means), the p/t-values for differences are displayed in a separate table called Diff.

ODS Graphics

This section describes the use of ODS for creating statistical graphs with the GLM procedure. To request these graphs you must specify the ODS GRAPHICS statement with an appropriate model, as discussed in the following list. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

When the ODS Graphics are in effect, then for particular models the GLM procedure will produce default graphics.

- If you specify a one-way analysis of variance model, with just one CLASS variable, the GLM procedure will produce a grouped box plot of the response values versus the CLASS levels. For an example of the box plot, see the section "One-Way Layout with Means Comparisons" on page 693.
- If you specify a two-way analysis of variance model, with just two CLASS variables, the GLM procedure will produce an interaction plot of the response values, with horizontal position representing one CLASS variable and marker style representing the other; and with predicted response values connected by lines representing the two-way analysis. For an example of the interaction plot, see the section "PROC GLM for Unbalanced ANOVA" on page 2435.
- If you specify a model with a single continuous predictor, the GLM procedure will produce a fit plot of the response values versus the covariate values, with a curve representing the fitted relationship. For an example of the fit plot, see the section "PROC GLM for Quadratic Least Squares Regression" on page 2438.
- If you specify a model with a two continuous predictors and no CLASS variables, the GLM procedure will produce a panel of fit plots as in the single predictor case, with a plot of the response values versus one of the covariates at each of several values of the other covariate.
- If you specify an analysis of covariance model, with one or two CLASS variables and one continuous variable, the GLM procedure will produce an analysis of covariance plot of the response values versus the covariate values, with lines representing the fitted relationship within each classification level. For an example of the analysis of covariance plot, see Example 39.4.

- If you specify an LSMEANS statement with the PDIFF option, the GLM procedure will produce a plot appropriate for the type of LS-means comparison. For PDIFF=ALL (which is the default if you specify only PDIFF), the procedure produces a diffogram, which displays all pairwise LS-means differences and their significance. The display is also known as a "mean-mean scatter plot" (Hsu 1996). For PDIFF=CONTROL, the procedure produces a display of each noncontrol LS-mean compared to the control LS-mean, with two-sided confidence intervals for the comparison. For PDIFF=CONTROLL and PDIFF=CONTROLU a similar display is produced, but with one-sided confidence intervals. Finally, for the PDIFF=ANOM option, the procedure produces an "analysis of means" plot, comparing each LS-mean to the average LS-mean.
- If you specify a MEANS statement, the GLM procedure will produce a grouped box plot of the response values versus the effect for which means are being calculated.

In addition to the default graphics mentioned previously, you can request plots that help you diagnose the quality of the fitted model.

- The PLOTS=DIAGNOSTICS option in the PROC GLM statement requests that a panel of summary diagnostics for the fit be displayed. The panel displays scatter plots of residuals, absolute residuals, studentized residuals, and observed responses by predicted values; studentized residuals by leverage; Cook's D by observation; a Q-Q plot of residuals; a residual histogram; and a residual-fit spread plot.
- The PLOTS=RESIDUALS option in the PROC GLM statement requests scatter plots of the residuals against each continuous covariate.

ODS Graph Names

PROC GLM assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 39.10.

To request these graphs you must specify the ODS Graphics statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Table 39.10 ODS Graphics Produced by PROC GLM

ODS Graph Name	Plot Description	Option
ANCOVAPlot	Analysis of covariance plot	Analysis of covariance model
ANOMPlot	Plot of LS-mean differences against average LS-mean	LSMEANS / PDIFF=ANOM
BoxPlot	Box plot of group means	One-way ANOVA model or MEANS statement
ContourFit	Plot of predicted response surface	Two-predictor response surface model
ControlPlot	Plot of LS-mean differences against a control level	LSMEANS / PDIFF=CONTROL

Table 39.10 *continued*

ODS Graph Name	Plot Description	Option
DiagnosticsPanel	Panel of summary diagnostics for the fit	PLOTS=DIAGNOSTICS
CooksDPlot	Cook's *D* plot	PLOTS=DIAGNOSTICS(UNPACK)
ObservedByPredicted	Observed by predicted	PLOTS=DIAGNOSTICS(UNPACK)
QQPlot	Residual Q-Q plot	PLOTS=DIAGNOSTICS(UNPACK)
ResidualByPredicted	Residual by predicted values	PLOTS=DIAGNOSTICS(UNPACK)
ResidualHistogram	Residual histogram	PLOTS=DIAGNOSTICS(UNPACK)
RFPlot	RF plot	PLOTS=DIAGNOSTICS(UNPACK)
RStudentByPredicted	Studentized residuals by predicted	PLOTS=DIAGNOSTICS(UNPACK)
RStudentByLeverage	RStudent by hat diagonals	PLOTS=DIAGNOSTICS(UNPACK)
DiffPlot	Plot of LS-mean pairwise differences	LSMEANS / PDIFF
IntPlot	Interaction plot	Two-way ANOVA model
FitPlot	Plot of predicted response by predictor	Model with one continuous predictor
ResidualPlots	Plots of the residuals against each continuous covariate	PLOTS=RESIDUALS

Examples: GLM Procedure

Example 39.1: Randomized Complete Blocks with Means Comparisons and Contrasts

This example, reported by Stenstrom (1940), analyzes an experiment to investigate how snapdragons grow in various soils. To eliminate the effect of local fertility variations, the experiment is run in blocks, with each soil type sampled in each block. Since these data are balanced, the Type I and Type III SS are the same and are equal to the traditional ANOVA SS.

First, the standard analysis is shown, followed by an analysis that uses the SOLUTION option and includes MEANS and CONTRAST statements. The ORDER=DATA option in the second PROC GLM statement is used so that the ordering of coefficients in the CONTRAST statement can correspond to the ordering in the input data. The SOLUTION option requests a display of the parameter estimates, which are produced by default only if there are no CLASS variables. A MEANS statement is used to request a table of the means with two multiple-comparison procedures requested. In experiments with focused treatment questions, CONTRAST statements are preferable to general means comparison methods. The following statements produce Output 39.1.1 through Output 39.1.4.

```
title 'Balanced Data from Randomized Complete Block';
data plants;
   input Type $ @;
   do Block = 1 to 3;
      input StemLength @;
      output;
      end;
   datalines;
Clarion  32.7 32.3 31.5
Clinton  32.1 29.7 29.1
Knox     35.7 35.9 33.1
O'Neill  36.0 34.2 31.2
Compost  31.8 28.0 29.2
Wabash   38.2 37.8 31.9
Webster  32.5 31.1 29.7
;

proc glm;
   class Block Type;
   model StemLength = Block Type;
run;

proc glm order=data;
   class Block Type;
   model StemLength = Block Type / solution;

/*---------------------------------clrn-cltn-knox-onel-cpst-wbsh-wstr */
contrast 'Compost vs. others'  Type   -1   -1   -1   -1    6   -1   -1;
contrast 'River soils vs. non' Type   -1   -1   -1   -1    0    5   -1,
                               Type   -1    4   -1   -1    0    0   -1;
contrast 'Glacial vs. drift'   Type   -1    0    1    1    0    0   -1;
contrast 'Clarion vs. Webster' Type   -1    0    0    0    0    0    1;
contrast "Knox vs. O'Neill"    Type    0    0    1   -1    0    0    0;
run;

    means Type / waller regwq;
run;
```

Output 39.1.1 Analysis of Variance for Randomized Complete Blocks

```
                Balanced Data from Randomized Complete Block

                            The GLM Procedure

                          Class Level Information

  Class         Levels    Values

  Block              3    1 2 3

  Type               7    Clarion Clinton Compost Knox O'Neill Wabash Webster

                   Number of Observations Read          21
                   Number of Observations Used          21
```

Output 39.1.1 *continued*

Balanced Data from Randomized Complete Block

The GLM Procedure

Dependent Variable: StemLength

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	8	142.1885714	17.7735714	10.80	0.0002
Error	12	19.7428571	1.6452381		
Corrected Total	20	161.9314286			

R-Square	Coeff Var	Root MSE	StemLength Mean
0.878079	3.939745	1.282668	32.55714

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Block	2	39.0371429	19.5185714	11.86	0.0014
Type	6	103.1514286	17.1919048	10.45	0.0004

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Block	2	39.0371429	19.5185714	11.86	0.0014
Type	6	103.1514286	17.1919048	10.45	0.0004

This analysis shows that the stem length is significantly different for the different soil types. In addition, there are significant differences in stem length among the three blocks in the experiment.

The GLM procedure is invoked again, this time with the ORDER=DATA option. This enables you to write accurate contrast statements more easily because you know the order SAS is using for the levels of the variable Type. The standard analysis is displayed again, this time including the tests for contrasts that you specified as well as the estimated parameters. These additional results are shown in Output 39.1.2.

Output 39.1.2 Contrasts and Solutions

Balanced Data from Randomized Complete Block

The GLM Procedure

Dependent Variable: StemLength

Contrast	DF	Contrast SS	Mean Square	F Value	Pr > F
Compost vs. others	1	29.24198413	29.24198413	17.77	0.0012
River soils vs. non	2	48.24694444	24.12347222	14.66	0.0006
Glacial vs. drift	1	22.14083333	22.14083333	13.46	0.0032
Clarion vs. Webster	1	1.70666667	1.70666667	1.04	0.3285
Knox vs. O'Neill	1	1.81500000	1.81500000	1.10	0.3143

Parameter		Estimate	Standard Error	t Value	Pr > |t|
Intercept		29.35714286 B	0.83970354	34.96	<.0001
Block	1	3.32857143 B	0.68561507	4.85	0.0004
Block	2	1.90000000 B	0.68561507	2.77	0.0169
Block	3	0.00000000 B	.	.	.
Type	Clarion	1.06666667 B	1.04729432	1.02	0.3285
Type	Clinton	-0.80000000 B	1.04729432	-0.76	0.4597
Type	Knox	3.80000000 B	1.04729432	3.63	0.0035
Type	O'Neill	2.70000000 B	1.04729432	2.58	0.0242
Type	Compost	-1.43333333 B	1.04729432	-1.37	0.1962
Type	Wabash	4.86666667 B	1.04729432	4.65	0.0006
Type	Webster	0.00000000 B	.	.	.

NOTE: The X'X matrix has been found to be singular, and a generalized inverse was used to solve the normal equations. Terms whose estimates are followed by the letter 'B' are not uniquely estimable.

The contrast label, degrees of freedom, sum of squares, Mean Square, F Value, and Pr > F are shown for each contrast requested. In this example, the contrast results indicate the following inferences, at the 5% significance level:

- The stem length of plants grown in compost soil is significantly different from the average stem length of plants grown in other soils.
- The stem length of plants grown in river soils is significantly different from the average stem length of those grown in nonriver soils.
- The average stem length of plants grown in glacial soils (Clarion and Webster types) is significantly different from the average stem length of those grown in drift soils (Knox and O'Neill types).
- Stem lengths for Clarion and Webster types are not significantly different.
- Stem lengths for Knox and O'Neill types are not significantly different.

In addition to the estimates for the parameters of the model, the results of t tests about the parameters are also displayed. The 'B' following the parameter estimates indicates that the estimates are biased and do not represent a unique solution to the normal equations.

Output 39.1.3 Waller-Duncan tests

```
                Balanced Data from Randomized Complete Block

                          The GLM Procedure

                Waller-Duncan K-ratio t Test for StemLength

NOTE: This test minimizes the Bayes risk under additive loss and certain other
                                assumptions.


                  Kratio                              100
                  Error Degrees of Freedom             12
                  Error Mean Square              1.645238
                  F Value                           10.45
                  Critical Value of t              2.12034
                  Minimum Significant Difference    2.2206


          Means with the same letter are not significantly different.


           Waller Grouping          Mean      N    Type

                        A        35.967      3    Wabash
                        A
                        A        34.900      3    Knox
                        A
                   B    A        33.800      3    O'Neill
                   B
                   B    C        32.167      3    Clarion
                        C
                   D    C        31.100      3    Webster
                   D    C
                   D    C        30.300      3    Clinton
                   D
                   D             29.667      3    Compost
```

Output 39.1.4 Ryan-Einot-Gabriel-Welsch Multiple Range Test

```
                 Balanced Data from Randomized Complete Block

                              The GLM Procedure

         Ryan-Einot-Gabriel-Welsch Multiple Range Test for StemLength

        NOTE: This test controls the Type I experimentwise error rate.


                     Alpha                        0.05
                     Error Degrees of Freedom       12
                     Error Mean Square        1.645238


 Number of Means         2         3         4         5         6         7
 Critical Range  2.9876673 3.2838304 3.4396326  3.540245  3.540245 3.6654165


         Means with the same letter are not significantly different.


              REGWQ Grouping          Mean      N    Type

                     A             35.967      3    Wabash
                     A
                 B   A             34.900      3    Knox
                 B   A
                 B   A    C        33.800      3    O'Neill
                 B        C
                 B   D    C        32.167      3    Clarion
                     D    C
                     D    C        31.100      3    Webster
                     D
                     D             30.300      3    Clinton
                     D
                     D             29.667      3    Compost
```

The final two pages of output (Output 39.1.3 and Output 39.1.4) present results of the Waller-Duncan and REGWQ multiple-comparison procedures. For each test, notes and information pertinent to the test are given in the output. The Type means are arranged from highest to lowest. Means with the same letter are not significantly different. For this example, while some pairs of means are significantly different, there are no clear equivalence classes among the different soils.

For an alternative method of analyzing and displaying mean differences, including high-resolution graphics, see Example 39.3.

Example 39.2: Regression with Mileage Data

A car is tested for gas mileage at various speeds to determine at what speed the car achieves the highest gas mileage. A quadratic model is fit to the experimental data. The following statements produce Output 39.2.1 through Output 39.2.4.

```
title 'Gasoline Mileage Experiment';
data mileage;
   input mph mpg @@;
   datalines;
20 15.4
30 20.2
40 25.7
50 26.2  50 26.6  50 27.4
55   .
60 24.8
;

ods graphics on;
proc glm;
   model mpg=mph mph*mph / p clm;
run;
ods graphics off;
```

Output 39.2.1 Standard Regression Analysis

```
                          Gasoline Mileage Experiment

                               The GLM Procedure

                    Number of Observations Read           8
                    Number of Observations Used           7

                          Gasoline Mileage Experiment

                               The GLM Procedure

Dependent Variable: mpg

                                      Sum of
 Source                       DF        Squares    Mean Square   F Value   Pr > F

 Model                         2    111.8086183     55.9043091     77.96   0.0006

 Error                         4      2.8685246      0.7171311

 Corrected Total               6    114.6771429

                R-Square     Coeff Var      Root MSE      mpg Mean

                0.974986      3.564553      0.846836      23.75714
```

Output 39.2.1 *continued*

Source	DF	Type I SS	Mean Square	F Value	Pr > F
mph	1	85.64464286	85.64464286	119.43	0.0004
mph*mph	1	26.16397541	26.16397541	36.48	0.0038

Source	DF	Type III SS	Mean Square	F Value	Pr > F
mph	1	41.01171219	41.01171219	57.19	0.0016
mph*mph	1	26.16397541	26.16397541	36.48	0.0038

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	-5.985245902	3.18522249	-1.88	0.1334
mph	1.305245902	0.17259876	7.56	0.0016
mph*mph	-0.013098361	0.00216852	-6.04	0.0038

The overall F statistic is significant. The tests of mph and mph*mph in the Type I sums of squares show that both the linear and quadratic terms in the regression model are significant. The model fits well, with an R^2 of 0.97. The table of parameter estimates indicates that the estimated regression equation is

$$\text{mpg} = -5.9852 + 1.3052 \times \text{mph} - 0.0131 \times \text{mph}^2$$

Output 39.2.2 Results of Requesting the P and CLM Options

Observation	Observed	Predicted	Residual
1	15.40000000	14.88032787	0.51967213
2	20.20000000	21.38360656	-1.18360656
3	25.70000000	25.26721311	0.43278689
4	26.20000000	26.53114754	-0.33114754
5	26.60000000	26.53114754	0.06885246
6	27.40000000	26.53114754	0.86885246
7 *	.	26.18073770	.
8	24.80000000	25.17540984	-0.37540984

Observation	95% Confidence Limits for Mean Predicted Value	
1	12.69701317	17.06364257
2	20.01727192	22.74994119
3	23.87460041	26.65982582
4	25.44573423	27.61656085
5	25.44573423	27.61656085
6	25.44573423	27.61656085
7 *	24.88679308	27.47468233
8	23.05954977	27.29126990

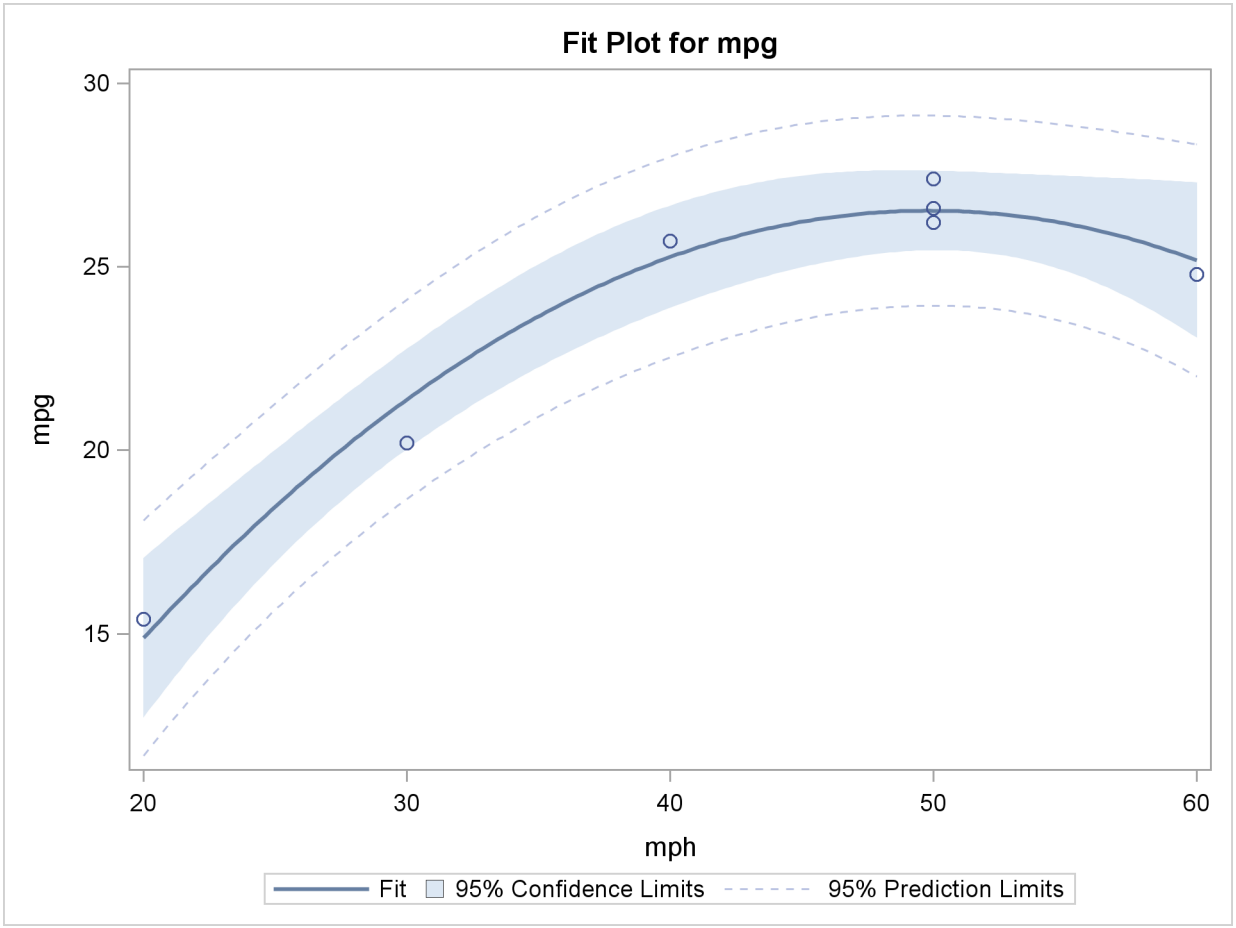
The P and CLM options in the MODEL statement produce the table shown in Output 39.2.2. For each observation, the observed, predicted, and residual values are shown. In addition, the 95% confidence limits for a mean predicted value are shown for each observation. Note that the observation with a missing value for mph is not used in the analysis, but predicted and confidence limit values are shown.

Output 39.2.3 Additional Results of Requesting the P and CLM Options

Sum of Residuals	−0.00000000
Sum of Squared Residuals	2.86852459
Sum of Squared Residuals − Error SS	−0.00000000
PRESS Statistic	23.18107335
First Order Autocorrelation	−0.54376613
Durbin-Watson D	2.94425592

The last portion of the output listing, shown in Output 39.2.3, gives some additional information about the residuals. The Press statistic gives the sum of squares of predicted residual errors, as described in Chapter 4, "Introduction to Regression Procedures." The First Order Autocorrelation and the Durbin-Watson D statistic, which measures first-order autocorrelation, are also given.

Output 39.2.4 Plot of Mileage Data



Finally, the ODS GRAPHICS ON command in the previous statements enables ODS Graphics, which in this case produces the plot shown in Output 39.2.4 of the actual and predicted values for the data, as well as a band representing the confidence limits for individual predictions. The quadratic relationship between mpg and mph is evident.

Example 39.3: Unbalanced ANOVA for Two-Way Design with Interaction

This example uses data from Kutner (1974, p. 98) to illustrate a two-way analysis of variance. The original data source is Afifi and Azen (1972, p. 166). These statements produce Output 39.3.1 and Output 39.3.2.

```
title 'Unbalanced Two-Way Analysis of Variance';
data a;
   input drug disease @;
   do i=1 to 6;
      input y @;
      output;
   end;
   datalines;
1 1 42 44 36 13 19 22
1 2 33  . 26  . 33 21
1 3 31 -3  . 25 25 24
2 1 28  . 23 34 42 13
2 2  . 34 33 31  . 36
2 3  3 26 28 32  4 16
3 1  .  .  1 29  . 19
3 2  . 11  9  7  1 -6
3 3 21  1  .  9  3  .
4 1 24  .  9 22 -2 15
4 2 27 12 12 -5 16 15
4 3 22  7 25  5 12  .
;

proc glm;
   class drug disease;
   model y=drug disease drug*disease / ss1 ss2 ss3 ss4;
run;
```

Output 39.3.1 Classes and Levels for Unbalanced Two-Way Design

Unbalanced Two-Way Analysis of Variance

The GLM Procedure

Class Level Information

Class	Levels	Values
drug	4	1 2 3 4
disease	3	1 2 3

Number of Observations Read	72
Number of Observations Used	58

Output 39.3.2 Analysis of Variance for Unbalanced Two-Way Design

Unbalanced Two-Way Analysis of Variance

The GLM Procedure

Dependent Variable: y

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	11	4259.338506	387.212591	3.51	0.0013
Error	46	5080.816667	110.452536		
Corrected Total	57	9340.155172			

R-Square	Coeff Var	Root MSE	y Mean
0.456024	55.66750	10.50964	18.87931

Source	DF	Type I SS	Mean Square	F Value	Pr > F
drug	3	3133.238506	1044.412835	9.46	<.0001
disease	2	418.833741	209.416870	1.90	0.1617
drug*disease	6	707.266259	117.877710	1.07	0.3958

Source	DF	Type II SS	Mean Square	F Value	Pr > F
drug	3	3063.432863	1021.144288	9.25	<.0001
disease	2	418.833741	209.416870	1.90	0.1617
drug*disease	6	707.266259	117.877710	1.07	0.3958

Source	DF	Type III SS	Mean Square	F Value	Pr > F
drug	3	2997.471860	999.157287	9.05	<.0001
disease	2	415.873046	207.936523	1.88	0.1637
drug*disease	6	707.266259	117.877710	1.07	0.3958

Output 39.3.2 *continued*

Source	DF	Type IV SS	Mean Square	F Value	Pr > F
drug	3	2997.471860	999.157287	9.05	<.0001
disease	2	415.873046	207.936523	1.88	0.1637
drug*disease	6	707.266259	117.877710	1.07	0.3958

Note the differences among the four types of sums of squares. The Type I sum of squares for drug essentially tests for differences between the expected values of the arithmetic mean response for different drugs, unadjusted for the effect of disease. By contrast, the Type II sum of squares for drug measures the differences between arithmetic means for each drug after adjusting for disease. The Type III sum of squares measures the differences between predicted drug means over a balanced drug×disease population—that is, between the LS-means for drug. Finally, the Type IV sum of squares is the same as the Type III sum of squares in this case, since there are data for every drug-by-disease combination.

No matter which sum of squares you prefer to use, this analysis shows a significant difference among the four drugs, while the disease effect and the drug-by-disease interaction are not significant. As the previous discussion indicates, Type III sums of squares correspond to differences between LS-means, so you can follow up the Type III tests with a multiple-comparison analysis of the drug LS-means. Since the GLM procedure is interactive, you can accomplish this by submitting the following statements after the previous ones that performed the ANOVA.

```
    lsmeans drug / pdiff=all adjust=tukey;
run;
```

Both the LS-means themselves and a matrix of adjusted p-values for pairwise differences between them are displayed; see Output 39.3.3 and Output 39.3.4.

Output 39.3.3 LS-Means for Unbalanced ANOVA

Unbalanced Two-Way Analysis of Variance

The GLM Procedure
Least Squares Means
Adjustment for Multiple Comparisons: Tukey-Kramer

drug	y LSMEAN	LSMEAN Number
1	25.9944444	1
2	26.5555556	2
3	9.7444444	3
4	13.5444444	4

Output 39.3.4 Adjusted p-Values for Pairwise LS-Mean Differences

Least Squares Means for effect drug
Pr > |t| for H0: LSMean(i)=LSMean(j)

Dependent Variable: y

i/j	1	2	3	4
1		0.9989	0.0016	0.0107
2	0.9989		0.0011	0.0071
3	0.0016	0.0011		0.7870
4	0.0107	0.0071	0.7870	

The multiple-comparison analysis shows that drugs 1 and 2 have very similar effects, and that drugs 3 and 4 are also insignificantly different from each other. Evidently, the main contribution to the significant drug effect is the difference between the 1/2 pair and the 3/4 pair.

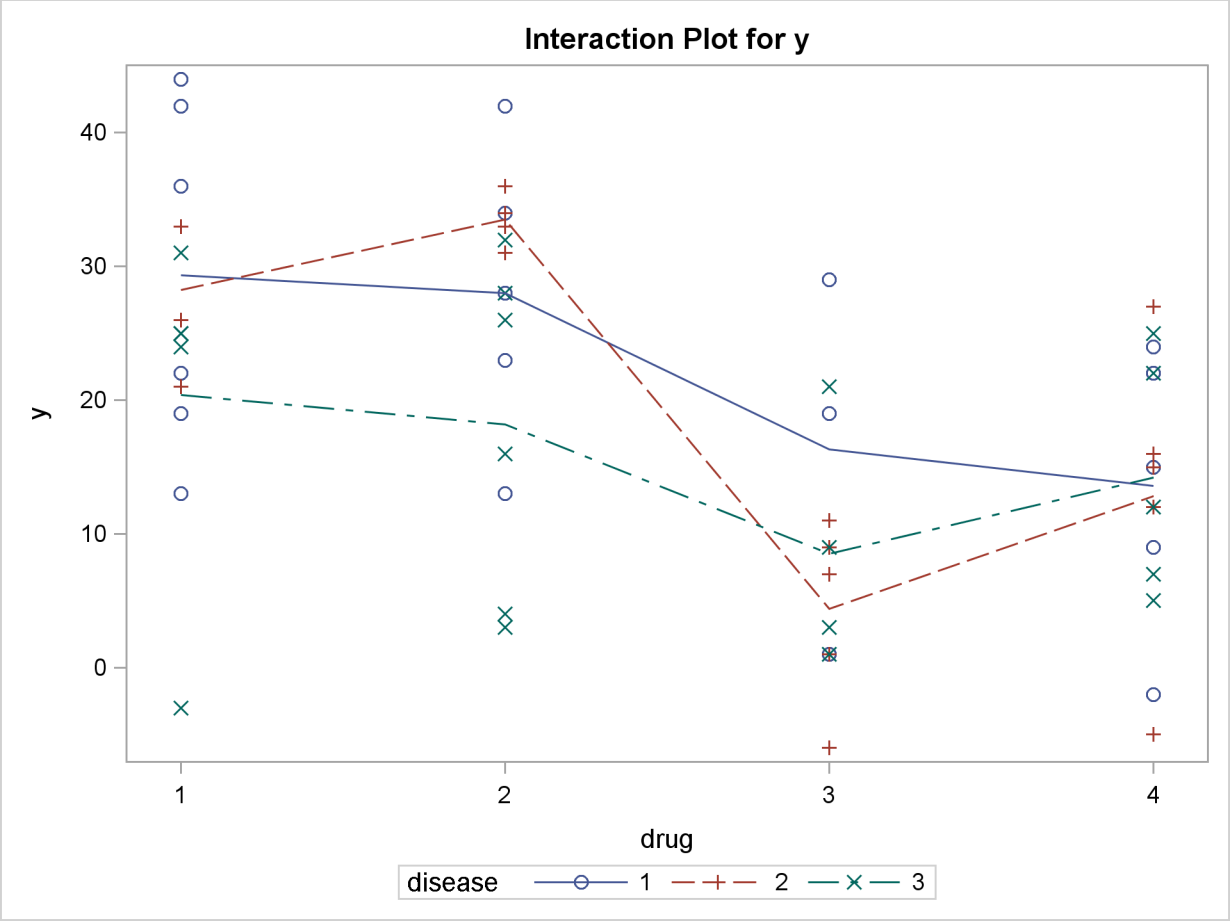
If you enable ODS Graphics for the previous analysis, GLM also displays three additional plots by default:

- an interaction plot for the effects of disease and drug
- a mean plot of the drug LS-means
- a plot of the adjusted pairwise differences and their significance levels

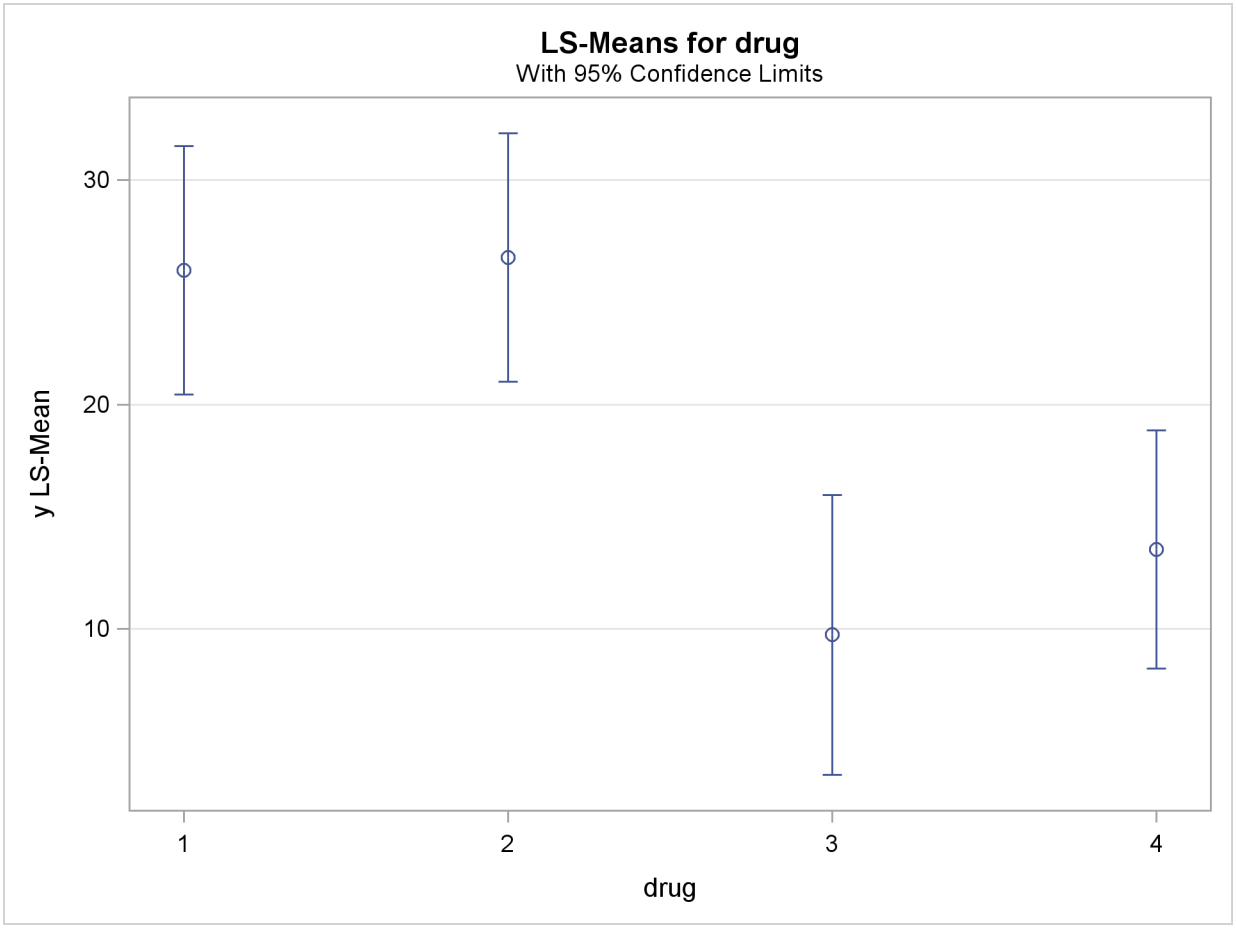
The following statements reproduce the previous analysis with ODS Graphics enabled. Additionally, the PLOTS=MEANPLOT(CL) option specifies that confidence limits for the LS-means should also be displayed in the mean plot. The graphical results are shown in Output 39.3.5 through Output 39.3.7.

```
ods graphics on;
proc glm plot=meanplot(cl);
   class drug disease;
   model y=drug disease drug*disease;
   lsmeans drug / pdiff=all adjust=tukey;
run;
ods graphics off;
```

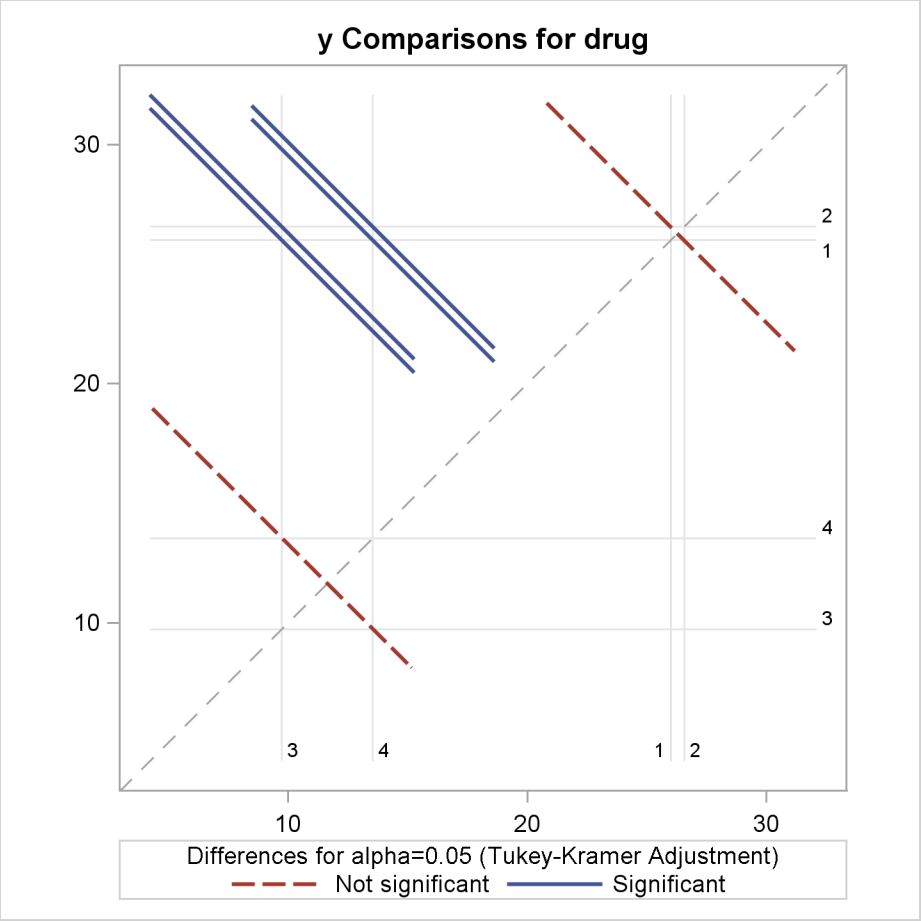
Output 39.3.5 Plot of Response by Drug and Disease



Output 39.3.6 Plot of Response LS-Means for Drug



Output 39.3.7 Plot of Response LS-Mean Differences for Drug



The significance of the drug differences is difficult to discern in the original data, as displayed in Output 39.3.5, but the plot of just the LS-means and their individual confidence limits in Output 39.3.6 makes it clearer. Finally, Output 39.3.7 indicates conclusively that the significance of the effect of drug is due to the difference between the two drug pairs (1, 2) and (3, 4).

Example 39.4: Analysis of Covariance

Analysis of covariance combines some of the features of both regression and analysis of variance. Typically, a continuous variable (the covariate) is introduced into the model of an analysis-of-variance experiment.

Data in the following example are selected from a larger experiment on the use of drugs in the treatment of leprosy (Snedecor and Cochran 1967, p. 422).

Variables in the study are as follows:

Drug	two antibiotics (A and D) and a control (F)
PreTreatment	a pretreatment score of leprosy bacilli
PostTreatment	a posttreatment score of leprosy bacilli

Ten patients are selected for each treatment (Drug), and six sites on each patient are measured for leprosy bacilli.

The covariate (a pretreatment score) is included in the model for increased precision in determining the effect of drug treatments on the posttreatment count of bacilli.

The following statements create the data set, perform a parallel-slopes analysis of covariance with PROC GLM, and compute Drug LS-means. These statements produce Output 39.4.1 and Output 39.4.2.

```
data DrugTest;
   input Drug $ PreTreatment PostTreatment @@;
   datalines;
A 11  6   A  8  0   A  5  2   A 14  8   A 19 11
A  6  4   A 10 13   A  6  1   A 11  8   A  3  0
D  6  0   D  6  2   D  7  3   D  8  1   D 18 18
D  8  4   D 19 14   D  8  9   D  5  1   D 15  9
F 16 13   F 13 10   F 11 18   F  9  5   F 21 23
F 16 12   F 12  5   F 12 16   F  7  1   F 12 20
;

proc glm data=DrugTest;
   class Drug;
   model PostTreatment = Drug PreTreatment / solution;
   lsmeans Drug / stderr pdiff cov out=adjmeans;
run;

proc print data=adjmeans;
run;
```

Output 39.4.1 Classes and Levels

```
                    The GLM Procedure

                 Class Level Information

              Class         Levels    Values

              Drug               3    A D F

          Number of Observations Read          30
          Number of Observations Used          30
```

Output 39.4.2 Overall Analysis of Variance

The GLM Procedure

Dependent Variable: PostTreatment

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	871.497403	290.499134	18.10	<.0001
Error	26	417.202597	16.046254		
Corrected Total	29	1288.700000			

R-Square	Coeff Var	Root MSE	PostTreatment Mean
0.676261	50.70604	4.005778	7.900000

This model assumes that the slopes relating posttreatment scores to pretreatment scores are parallel for all drugs. You can check this assumption by including the class-by-covariate interaction, Drug*PreTreatment, in the model and examining the ANOVA test for the significance of this effect. This extra test is omitted in this example, but it is insignificant, justifying the equal-slopes assumption.

In Output 39.4.3, the Type I SS for Drug (293.6) gives the between-drug sums of squares that are obtained for the analysis-of-variance model PostTreatment=Drug. This measures the difference between arithmetic means of posttreatment scores for different drugs, disregarding the covariate. The Type III SS for Drug (68.5537) gives the Drug sum of squares adjusted for the covariate. This measures the differences between Drug LS-means, controlling for the covariate. The Type I test is highly significant ($p = 0.001$), but the Type III test is not. This indicates that, while there is a statistically significant difference between the arithmetic drug means, this difference is reduced to below the level of background noise when you take the pretreatment scores into account. From the table of parameter estimates, you can derive the least squares predictive formula model for estimating posttreatment score based on pretreatment score and drug:

$$\text{post} = \begin{cases} (-0.435 + -3.446) + 0.987 \cdot \text{pre}, & \text{if Drug=A} \\ (-0.435 + -3.337) + 0.987 \cdot \text{pre}, & \text{if Drug=D} \\ -0.435 + 0.987 \cdot \text{pre}, & \text{if Drug=F} \end{cases}$$

Output 39.4.3 Tests and Parameter Estimates

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Drug	2	293.6000000	146.8000000	9.15	0.0010
PreTreatment	1	577.8974030	577.8974030	36.01	<.0001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Drug	2	68.5537106	34.2768553	2.14	0.1384
PreTreatment	1	577.8974030	577.8974030	36.01	<.0001

Parameter		Estimate	Standard Error	t Value	Pr > |t|
Intercept		−0.434671164 B	2.47135356	−0.18	0.8617
Drug	A	−3.446138280 B	1.88678065	−1.83	0.0793
Drug	D	−3.337166948 B	1.85386642	−1.80	0.0835
Drug	F	0.000000000 B	.	.	.
PreTreatment		0.987183811	0.16449757	6.00	<.0001

Output 39.4.4 displays the LS-means, which are, in a sense, the means adjusted for the covariate. The STDERR option in the LSMEANS statement causes the standard error of the LS-means and the probability of getting a larger t value under the hypothesis H_0: LS-mean $= 0$ to be included in this table as well. Specifying the PDIFF option causes all probability values for the hypothesis H_0: LS-mean(i) $=$ LS-mean(j) to be displayed, where the indexes i and j are numbered treatment levels.

Output 39.4.4 LS-Means

The GLM Procedure
Least Squares Means

Drug	Post Treatment LSMEAN	Standard Error	Pr > |t|	LSMEAN Number
A	6.7149635	1.2884943	<.0001	1
D	6.8239348	1.2724690	<.0001	2
F	10.1611017	1.3159234	<.0001	3

Least Squares Means for effect Drug
Pr > |t| for H0: LSMean(i)=LSMean(j)

Dependent Variable: PostTreatment

i/j	1	2	3
1		0.9521	0.0793
2	0.9521		0.0835
3	0.0793	0.0835	

The OUT= and COV options in the LSMEANS statement create a data set of the estimates,

their standard errors, and the variances and covariances of the LS-means, which is displayed in Output 39.4.5.

Output 39.4.5 LS-Means Output Data Set

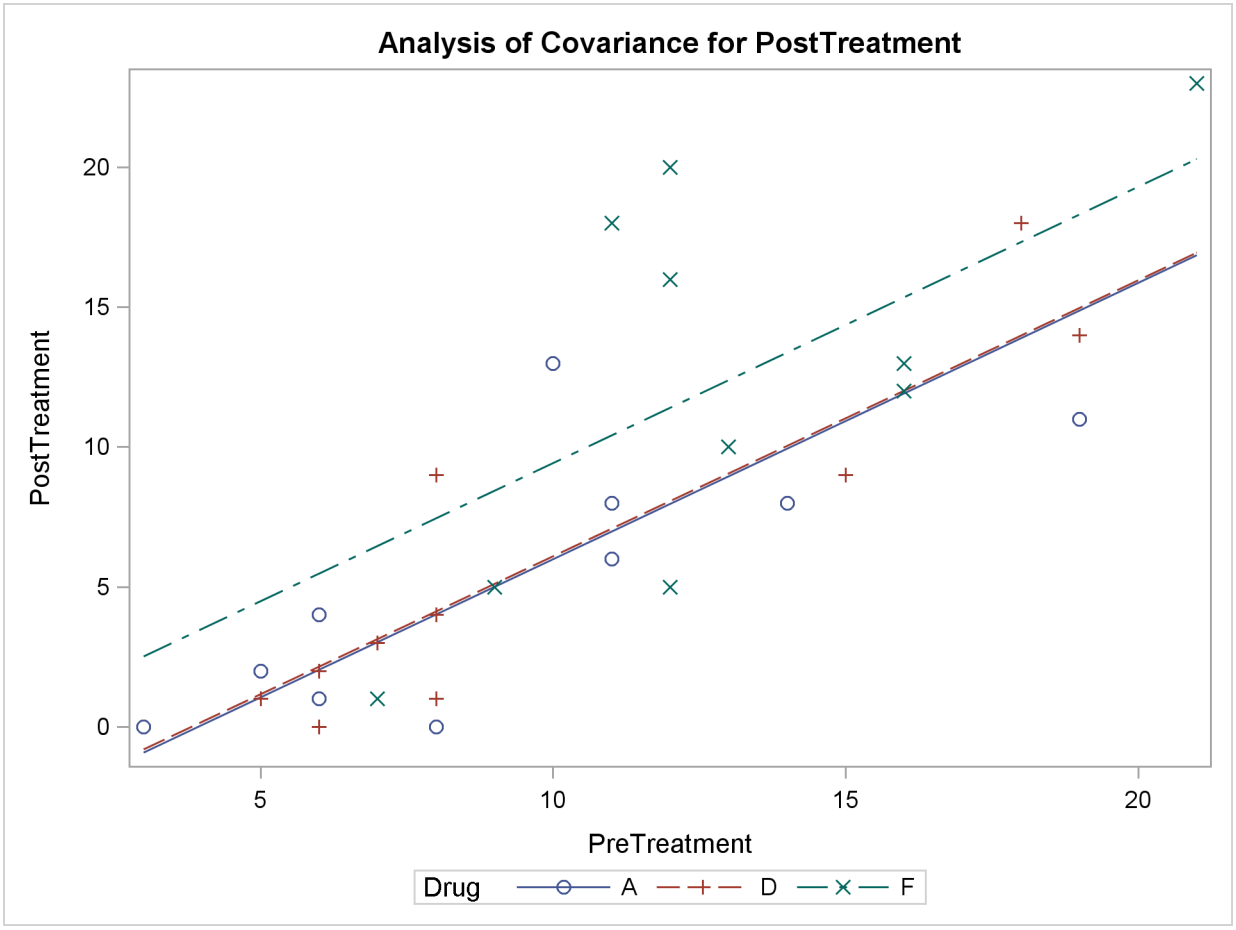
Obs	_NAME_	Drug	LSMEAN	STDERR	NUMBER	COV1	COV2	COV3
1	PostTreatment	A	6.7150	1.28849	1	1.66022	0.02844	−0.08403
2	PostTreatment	D	6.8239	1.27247	2	0.02844	1.61918	−0.04299
3	PostTreatment	F	10.1611	1.31592	3	−0.08403	−0.04299	1.73165

The new graphical features of PROC GLM enable you to visualize the fitted analysis of covariance model. The following statements enable the graphics by specifying the ODS GRAPHICS statement and then fit an analysis-of-covariance model with LS-means for Drug.

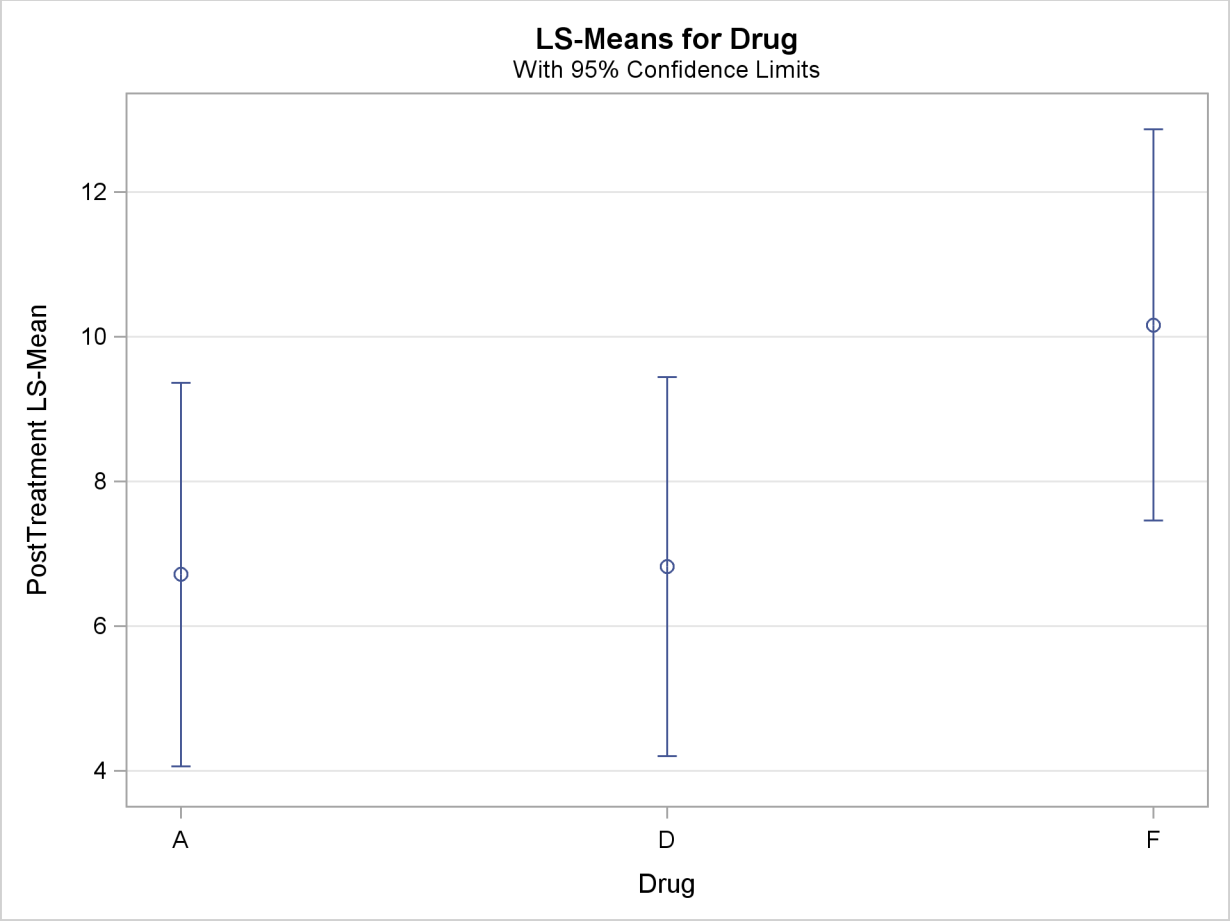
```
ods graphics on;

proc glm data=DrugTest plot=meanplot(cl);
   class Drug;
   model PostTreatment = Drug PreTreatment;
   lsmeans Drug / pdiff;
run;

ods graphics off;
```

With graphics enabled, the GLM procedure output includes an analysis-of-covariance plot, as in Output 39.4.6. The LSMEANS statement produces a plot of the LS-means; the SAS statements previously shown use the PLOTS=MEANPLOT(CL) option to add confidence limits for the individual LS-means, shown in Output 39.4.7. If you also specify the PDIFF option in the LSMEANS statement, the output also includes a plot appropriate for the type of LS-mean differences computed. In this case, the default is to compare all LS-means with each other pairwise, so the plot is a "diffogram" or "mean-mean scatter plot" (Hsu 1996), as in Output 39.4.8. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the GLM procedure, see the section "ODS Graphics" on page 2554.

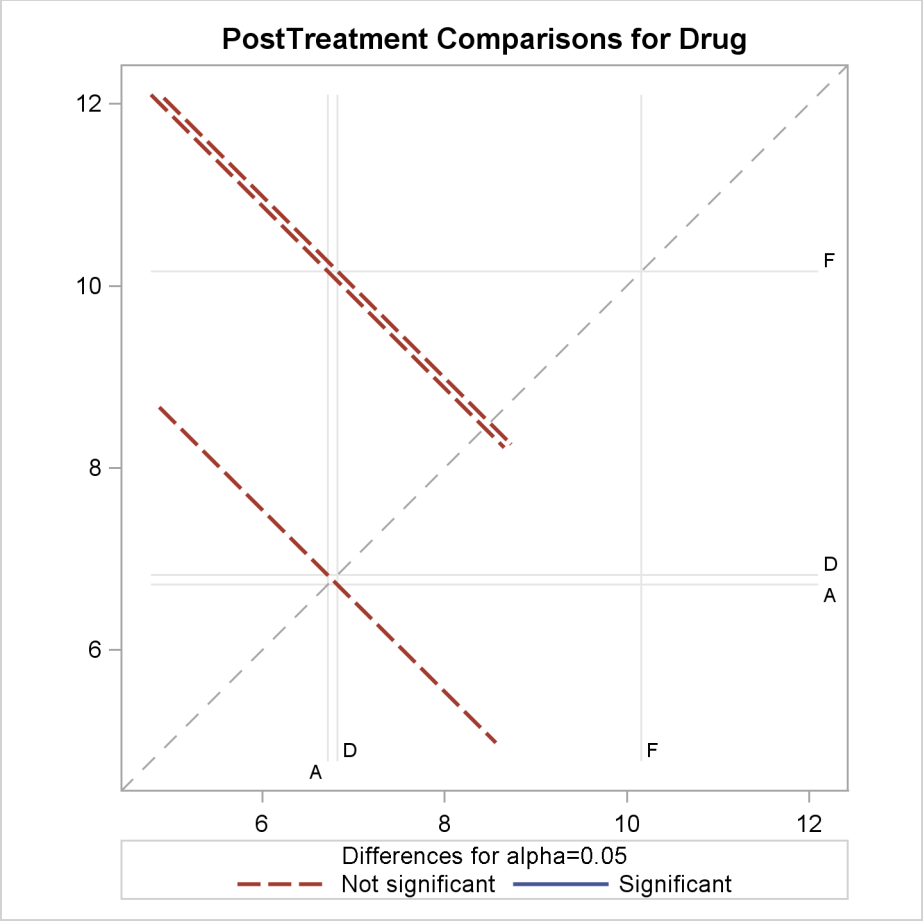
Output 39.4.6 Analysis of Covariance Plot of PostTreatment Score by Drug and PreTreatment Score



Output 39.4.7 LS-Means for PostTreatment Score by Drug



Output 39.4.8 Plot of Differences between Drug LS-Means for PostTreatment Scores



The analysis of covariance plot Output 39.4.6 makes it clear that the control (drug F) has higher posttreatment scores across the range of pretreatment scores, while the fitted models for the two antibiotics (drugs A and D) nearly coincide. Similarly, while the diffogram Output 39.4.7 indicates that none of the LS-mean differences are significant at the 5% level, the difference between the LS-means for the two antibiotics is much closer to zero than the differences between either one and the control.

Example 39.5: Three-Way Analysis of Variance with Contrasts

This example uses data from Cochran and Cox (1957, p. 176) to illustrate the analysis of a three-way factorial design with replication, including the use of the CONTRAST statement with interactions, the OUTSTAT= data set, and the SLICE= option in the LSMEANS statement.

The object of the study is to determine the effects of electric current on denervated muscle. The variables are as follows:

Rep	the replicate number, 1 or 2
Time	the length of time the current is applied to the muscle, ranging from 1 to 4
Current	the level of electric current applied, ranging from 1 to 4
Number	the number of treatments per day, ranging from 1 to 3
MuscleWeight	the weight of the denervated muscle

The following statements produce Output 39.5.1 through Output 39.5.4.

```
data muscles;
   do Rep=1 to 2;
      do Time=1 to 4;
         do Current=1 to 4;
            do Number=1 to 3;
               input MuscleWeight @@;
               output;
            end;
         end;
      end;
   end;
   datalines;
72 74 69 61 61 65 62 65 70 85 76 61
67 52 62 60 55 59 64 65 64 67 72 60
57 66 72 72 43 43 63 66 72 56 75 92
57 56 78 60 63 58 61 79 68 73 86 71
46 74 58 60 64 52 71 64 71 53 65 66
44 58 54 57 55 51 62 61 79 60 78 82
53 50 61 56 57 56 56 56 71 56 58 69
46 55 64 56 55 57 64 66 62 59 58 88
;
proc glm outstat=summary;
   class Rep Current Time Number;
   model MuscleWeight = Rep Current|Time|Number;
   contrast 'Time in Current 3'
      Time 1 0 0 -1 Current*Time 0 0 0 0 0 0 0 0 1 0 0 -1,
      Time 0 1 0 -1 Current*Time 0 0 0 0 0 0 0 0 0 1 0 -1,
      Time 0 0 1 -1 Current*Time 0 0 0 0 0 0 0 0 0 0 1 -1;
   contrast 'Current 1 versus 2' Current 1 -1;
   lsmeans Current*Time / slice=Current;
run;
proc print data=summary;
run;
```

The first CONTRAST statement examines the effects of Time within level 3 of Current. This is also called the *simple effect* of Time within Current*Time. Note that, since there are three degrees of freedom, it is necessary to specify three rows in the CONTRAST statement, separated by commas. Since the parameterization that PROC GLM uses is determined in part by the ordering of the variables in the CLASS statement, Current is specified before Time so that the Time parameters are nested within the Current*Time parameters; thus, the Current*Time contrast coefficients in each row are simply the Time coefficients of that row within the appropriate level of Current.

The second CONTRAST statement isolates a single-degree-of-freedom effect corresponding to the difference between the first two levels of Current. You can use such a contrast in a large experiment where certain preplanned comparisons are important, but you want to take advantage of the additional error degrees of freedom available when all levels of the factors are considered.

The LSMEANS statement with the SLICE= option is an alternative way to test for the simple effect of Time within Current*Time. In addition to listing the LS-means for each current strength and length of time, it gives a table of F tests for differences between the LS-means across Time within each Current level. In some cases, this can be a way to disentangle a complex interaction.

Output 39.5.1 Overall Analysis

The GLM Procedure

Class Level Information

Class	Levels	Values
Rep	2	1 2
Current	4	1 2 3 4
Time	4	1 2 3 4
Number	3	1 2 3

Number of Observations Read	96
Number of Observations Used	96

The GLM Procedure

Dependent Variable: MuscleWeight

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	48	5782.916667	120.477431	1.77	0.0261
Error	47	3199.489583	68.074246		
Corrected Total	95	8982.406250			

R-Square	Coeff Var	Root MSE	MuscleWeight Mean
0.643805	13.05105	8.250712	63.21875

The output, shown in Output 39.5.2 and Output 39.5.3, indicates that the main effects for Rep, Current, and Number are significant (with p-values of 0.0045, <0.0001, and 0.0461, respectively), but the main effect for Time is not significant, indicating that, in general, it does not matter how long the current is applied. None of the interaction terms are significant, nor are the contrasts significant. Notice that the row in the sliced ANOVA table corresponding to level 3 of current matches the "Time in Current 3" contrast.

Output 39.5.2 Individual Effects and Contrasts

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Rep	1	605.010417	605.010417	8.89	0.0045
Current	3	2145.447917	715.149306	10.51	<.0001
Time	3	223.114583	74.371528	1.09	0.3616
Current*Time	9	298.677083	33.186343	0.49	0.8756
Number	2	447.437500	223.718750	3.29	0.0461
Current*Number	6	644.395833	107.399306	1.58	0.1747
Time*Number	6	367.979167	61.329861	0.90	0.5023
Current*Time*Number	18	1050.854167	58.380787	0.86	0.6276

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Rep	1	605.010417	605.010417	8.89	0.0045
Current	3	2145.447917	715.149306	10.51	<.0001
Time	3	223.114583	74.371528	1.09	0.3616
Current*Time	9	298.677083	33.186343	0.49	0.8756
Number	2	447.437500	223.718750	3.29	0.0461
Current*Number	6	644.395833	107.399306	1.58	0.1747
Time*Number	6	367.979167	61.329861	0.90	0.5023
Current*Time*Number	18	1050.854167	58.380787	0.86	0.6276

Contrast	DF	Contrast SS	Mean Square	F Value	Pr > F
Time in Current 3	3	34.83333333	11.61111111	0.17	0.9157
Current 1 versus 2	1	99.18750000	99.18750000	1.46	0.2334

Output 39.5.3 Simple Effects of Time

The GLM Procedure
Least Squares Means

Current*Time Effect Sliced by Current for MuscleWeight

Current	DF	Sum of Squares	Mean Square	F Value	Pr > F
1	3	271.458333	90.486111	1.33	0.2761
2	3	120.666667	40.222222	0.59	0.6241
3	3	34.833333	11.611111	0.17	0.9157
4	3	94.833333	31.611111	0.46	0.7085

The SS, F statistics, and p-values can be stored in an OUTSTAT= data set, as shown in Output 39.5.4.

Output 39.5.4 Contents of the OUTSTAT= Data Set

Obs	_NAME_	_SOURCE_	_TYPE_	DF	SS	F	PROB
1	MuscleWeight	ERROR	ERROR	47	3199.49	.	.
2	MuscleWeight	Rep	SS1	1	605.01	8.8875	0.00454
3	MuscleWeight	Current	SS1	3	2145.45	10.5054	0.00002
4	MuscleWeight	Time	SS1	3	223.11	1.0925	0.36159
5	MuscleWeight	Current*Time	SS1	9	298.68	0.4875	0.87562
6	MuscleWeight	Number	SS1	2	447.44	3.2864	0.04614
7	MuscleWeight	Current*Number	SS1	6	644.40	1.5777	0.17468
8	MuscleWeight	Time*Number	SS1	6	367.98	0.9009	0.50231
9	MuscleWeight	Current*Time*Number	SS1	18	1050.85	0.8576	0.62757
10	MuscleWeight	Rep	SS3	1	605.01	8.8875	0.00454
11	MuscleWeight	Current	SS3	3	2145.45	10.5054	0.00002
12	MuscleWeight	Time	SS3	3	223.11	1.0925	0.36159
13	MuscleWeight	Current*Time	SS3	9	298.68	0.4875	0.87562
14	MuscleWeight	Number	SS3	2	447.44	3.2864	0.04614
15	MuscleWeight	Current*Number	SS3	6	644.40	1.5777	0.17468
16	MuscleWeight	Time*Number	SS3	6	367.98	0.9009	0.50231
17	MuscleWeight	Current*Time*Number	SS3	18	1050.85	0.8576	0.62757
18	MuscleWeight	Time in Current 3	CONTRAST	3	34.83	0.1706	0.91574
19	MuscleWeight	Current 1 versus 2	CONTRAST	1	99.19	1.4570	0.23344

Example 39.6: Multivariate Analysis of Variance

This example employs multivariate analysis of variance (MANOVA) to measure differences in the chemical characteristics of ancient pottery found at four kiln sites in Great Britain. The data are from Tubb, Parker, and Nickless (1980), as reported in Hand et al. (1994).

For each of 26 samples of pottery, the percentages of oxides of five metals are measured. The following statements create the data set and invoke the GLM procedure to perform a one-way MANOVA. Additionally, it is of interest to know whether the pottery from one site in Wales (Llanederyn) differs from the samples from other sites; a CONTRAST statement is used to test this hypothesis.

```
title "Romano-British Pottery";
data pottery;
   input Site $12. Al Fe Mg Ca Na;
   datalines;
Llanederyn   14.4 7.00 4.30 0.15 0.51
Llanederyn   13.8 7.08 3.43 0.12 0.17
Llanederyn   14.6 7.09 3.88 0.13 0.20
Llanederyn   11.5 6.37 5.64 0.16 0.14
Llanederyn   13.8 7.06 5.34 0.20 0.20
Llanederyn   10.9 6.26 3.47 0.17 0.22
Llanederyn   10.1 4.26 4.26 0.20 0.18
Llanederyn   11.6 5.78 5.91 0.18 0.16
Llanederyn   11.1 5.49 4.52 0.29 0.30
Llanederyn   13.4 6.92 7.23 0.28 0.20
Llanederyn   12.4 6.13 5.69 0.22 0.54
```

```
Llanederyn   13.1 6.64 5.51 0.31 0.24
Llanederyn   12.7 6.69 4.45 0.20 0.22
Llanederyn   12.5 6.44 3.94 0.22 0.23
Caldicot     11.8 5.44 3.94 0.30 0.04
Caldicot     11.6 5.39 3.77 0.29 0.06
IslandThorns 18.3 1.28 0.67 0.03 0.03
IslandThorns 15.8 2.39 0.63 0.01 0.04
IslandThorns 18.0 1.50 0.67 0.01 0.06
IslandThorns 18.0 1.88 0.68 0.01 0.04
IslandThorns 20.8 1.51 0.72 0.07 0.10
AshleyRails  17.7 1.12 0.56 0.06 0.06
AshleyRails  18.3 1.14 0.67 0.06 0.05
AshleyRails  16.7 0.92 0.53 0.01 0.05
AshleyRails  14.8 2.74 0.67 0.03 0.05
AshleyRails  19.1 1.64 0.60 0.10 0.03
;

proc glm data=pottery;
   class Site;
   model Al Fe Mg Ca Na = Site;
   contrast 'Llanederyn vs. the rest' Site 1 1 1 -3;
   manova h=_all_ / printe printh;
run;
```

After the summary information, displayed in Output 39.6.1, PROC GLM produces the univariate analyses for each of the dependent variables, as shown in Output 39.6.2 through Output 39.6.6. These analyses show that sites are significantly different for all oxides individually. You can suppress these univariate analyses by specifying the NOUNI option in the MODEL statement.

Output 39.6.1 Summary Information about Groups

```
                          Romano-British Pottery

                            The GLM Procedure

                         Class Level Information

     Class         Levels    Values

     Site               4    AshleyRails Caldicot IslandThorns Llanederyn

                  Number of Observations Read          26
                  Number of Observations Used          26
```

Output 39.6.2 Univariate Analysis of Variance for Aluminum Oxide

Romano-British Pottery

The GLM Procedure

Dependent Variable: Al

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	175.6103187	58.5367729	26.67	<.0001
Error	22	48.2881429	2.1949156		
Corrected Total	25	223.8984615			

R-Square	Coeff Var	Root MSE	Al Mean
0.784330	10.22284	1.481525	14.49231

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Site	3	175.6103187	58.5367729	26.67	<.0001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Site	3	175.6103187	58.5367729	26.67	<.0001

Contrast	DF	Contrast SS	Mean Square	F Value	Pr > F
Llanederyn vs. the rest	1	58.58336640	58.58336640	26.69	<.0001

Output 39.6.3 Univariate Analysis of Variance for Iron Oxide

Romano-British Pottery

The GLM Procedure

Dependent Variable: Fe

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	134.2216158	44.7405386	89.88	<.0001
Error	22	10.9508457	0.4977657		
Corrected Total	25	145.1724615			

R-Square	Coeff Var	Root MSE	Fe Mean
0.924567	15.79171	0.705525	4.467692

Output 39.6.3 *continued*

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Site	3	134.2216158	44.7405386	89.88	<.0001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Site	3	134.2216158	44.7405386	89.88	<.0001

Contrast	DF	Contrast SS	Mean Square	F Value	Pr > F
Llanederyn vs. the rest	1	71.15144132	71.15144132	142.94	<.0001

Output 39.6.4 Univariate Analysis of Variance for Calcium Oxide

Romano-British Pottery

The GLM Procedure

Dependent Variable: Ca

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	0.20470275	0.06823425	29.16	<.0001
Error	22	0.05148571	0.00234026		
Corrected Total	25	0.25618846			

R-Square	Coeff Var	Root MSE	Ca Mean
0.799032	33.01265	0.048376	0.146538

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Site	3	0.20470275	0.06823425	29.16	<.0001

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Site	3	0.20470275	0.06823425	29.16	<.0001

Contrast	DF	Contrast SS	Mean Square	F Value	Pr > F
Llanederyn vs. the rest	1	0.03531688	0.03531688	15.09	0.0008

Output 39.6.5 Univariate Analysis of Variance for Magnesium Oxide

```
                          Romano-British Pottery

                            The GLM Procedure

Dependent Variable: Mg

                                     Sum of
 Source                    DF        Squares    Mean Square   F Value   Pr > F

 Model                      3    103.3505270     34.4501757     49.12   <.0001

 Error                     22     15.4296114      0.7013460

 Corrected Total           25    118.7801385

              R-Square     Coeff Var      Root MSE       Mg Mean

              0.870099      26.65777      0.837464      3.141538

 Source                    DF      Type I SS    Mean Square   F Value   Pr > F

 Site                       3    103.3505270     34.4501757     49.12   <.0001

 Source                    DF    Type III SS    Mean Square   F Value   Pr > F

 Site                       3    103.3505270     34.4501757     49.12   <.0001

  Contrast                    DF   Contrast SS   Mean Square  F Value  Pr > F

  Llanederyn vs. the rest      1   56.59349339   56.59349339    80.69  <.0001
```

Output 39.6.6 Univariate Analysis of Variance for Sodium Oxide

```
                          Romano-British Pottery

                            The GLM Procedure

Dependent Variable: Na

                                     Sum of
 Source                    DF        Squares    Mean Square   F Value   Pr > F

 Model                      3     0.25824560     0.08608187      9.50   0.0003

 Error                     22     0.19929286     0.00905877

 Corrected Total           25     0.45753846

              R-Square     Coeff Var      Root MSE       Na Mean

              0.564424      60.06350      0.095178      0.158462
```

Output 39.6.6 *continued*

Source	DF	Type I SS	Mean Square	F Value	Pr > F
Site	3	0.25824560	0.08608187	9.50	0.0003

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Site	3	0.25824560	0.08608187	9.50	0.0003

Contrast	DF	Contrast SS	Mean Square	F Value	Pr > F
Llanederyn vs. the rest	1	0.23344446	0.23344446	25.77	<.0001

The PRINTE option in the MANOVA statement displays the elements of the error matrix, also called the Error Sums of Squares and Crossproducts matrix. (See Output 39.6.7.) The diagonal elements of this matrix are the error sums of squares from the corresponding univariate analyses.

The PRINTE option also displays the partial correlation matrix associated with the E matrix. In this example, none of the oxides are very strongly correlated; the strongest correlation ($r = 0.488$) is between magnesium oxide and calcium oxide.

Output 39.6.7 Error SSCP Matrix and Partial Correlations

Romano-British Pottery

The GLM Procedure
Multivariate Analysis of Variance

E = Error SSCP Matrix

	Al	Fe	Mg	Ca	Na
Al	48.288142857	7.0800714286	0.6080142857	0.1064714286	0.5889571429
Fe	7.0800714286	10.950845714	0.5270571429	-0.155194286	0.0667585714
Mg	0.6080142857	0.5270571429	15.429611429	0.4353771429	0.0276157143
Ca	0.1064714286	-0.155194286	0.4353771429	0.0514857143	0.0100785714
Na	0.5889571429	0.0667585714	0.0276157143	0.0100785714	0.1992928571

Output 39.6.7 *continued*

Partial Correlation Coefficients from the Error SSCP Matrix / Prob > |r|

DF = 22	Al	Fe	Mg	Ca	Na
Al	1.000000	0.307889 0.1529	0.022275 0.9196	0.067526 0.7595	0.189853 0.3856
Fe	0.307889 0.1529	1.000000	0.040547 0.8543	−0.206685 0.3440	0.045189 0.8378
Mg	0.022275 0.9196	0.040547 0.8543	1.000000	0.488478 0.0180	0.015748 0.9431
Ca	0.067526 0.7595	−0.206685 0.3440	0.488478 0.0180	1.000000	0.099497 0.6515
Na	0.189853 0.3856	0.045189 0.8378	0.015748 0.9431	0.099497 0.6515	1.000000

The PRINTH option produces the SSCP matrix for the hypotheses being tested (Site and the contrast); see Output 39.6.8 and Output 39.6.9. Since the Type III SS are the highest-level SS produced by PROC GLM by default, and since the HTYPE= option is not specified, the SSCP matrix for Site gives the Type III $\mathbf{H}$ matrix. The diagonal elements of this matrix are the model sums of squares from the corresponding univariate analyses.

Four multivariate tests are computed, all based on the characteristic roots and vectors of $\mathbf{E}^{-1}\mathbf{H}$. These roots and vectors are displayed along with the tests. All four tests can be transformed to variates that have F distributions under the null hypothesis. Note that the four tests all give the same results for the contrast, since it has only one degree of freedom. In this case, the multivariate analysis matches the univariate results: there is an overall difference between the chemical composition of samples from different sites, and the samples from Llanederyn are different from the average of the other sites.

Output 39.6.8 Hypothesis SSCP Matrix and Multivariate Tests for Overall Site Effect

Romano-British Pottery

The GLM Procedure
Multivariate Analysis of Variance

H = Type III SSCP Matrix for Site

	Al	Fe	Mg	Ca	Na
Al	175.61031868	−149.295533	−130.8097066	−5.889163736	−5.372264835
Fe	−149.295533	134.22161582	117.74503516	4.8217865934	5.3259491209
Mg	−130.8097066	117.74503516	103.35052703	4.2091613187	4.7105458242
Ca	−5.889163736	4.8217865934	4.2091613187	0.2047027473	0.154782967
Na	−5.372264835	5.3259491209	4.7105458242	0.154782967	0.2582456044

Output 39.6.8 *continued*

Characteristic Roots and Vectors of: E Inverse * H, where
H = Type III SSCP Matrix for Site
E = Error SSCP Matrix

Characteristic Root	Percent	Characteristic Vector V'EV=1 Al	Fe	Mg	Ca	Na
34.1611140	96.39	0.09562211	−0.26330469	−0.05305978	−1.87982100	−0.47071123
1.2500994	3.53	0.02651891	−0.01239715	0.17564390	−4.25929785	1.23727668
0.0275396	0.08	0.09082220	0.13159869	0.03508901	−0.15701602	−1.39364544
0.0000000	0.00	0.03673984	−0.15129712	0.20455529	0.54624873	−0.17402107
0.0000000	0.00	0.06862324	0.03056912	−0.10662399	2.51151978	1.23668841

MANOVA Test Criteria and F Approximations for the Hypothesis of No Overall Site Effect
H = Type III SSCP Matrix for Site
E = Error SSCP Matrix

S=3 M=0.5 N=8

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.01230091	13.09	15	50.091	<.0001
Pillai's Trace	1.55393619	4.30	15	60	<.0001
Hotelling-Lawley Trace	35.43875302	40.59	15	29.13	<.0001
Roy's Greatest Root	34.16111399	136.64	5	20	<.0001

NOTE: F Statistic for Roy's Greatest Root is an upper bound.

Output 39.6.9 Hypothesis SSCP Matrix and Multivariate Tests for Differences between Llanederyn and the Other Sites

H = Contrast SSCP Matrix for Llanederyn vs. the rest

	Al	Fe	Mg	Ca	Na
Al	58.583366402	−64.56230291	−57.57983466	−1.438395503	−3.698102513
Fe	−64.56230291	71.151441323	63.456352116	1.5851961376	4.0755256878
Mg	−57.57983466	63.456352116	56.593493386	1.4137558201	3.6347541005
Ca	−1.438395503	1.5851961376	1.4137558201	0.0353168783	0.0907993915
Na	−3.698102513	4.0755256878	3.6347541005	0.0907993915	0.2334444577

Characteristic Roots and Vectors of: E Inverse * H, where
H = Contrast SSCP Matrix for Llanederyn vs. the rest
E = Error SSCP Matrix

Characteristic Root	Percent	Characteristic Vector V'EV=1 Al	Fe	Mg	Ca	Na
16.1251646	100.00	−0.08883488	0.25458141	0.08723574	0.98158668	0.71925759
0.0000000	0.00	−0.00503538	0.03825743	−0.17632854	5.16256699	−0.01022754
0.0000000	0.00	0.00162771	−0.08885364	−0.01774069	−0.83096817	2.17644566
0.0000000	0.00	0.04450136	−0.15722494	0.22156791	0.00000000	0.00000000
0.0000000	0.00	0.11939206	0.10833549	0.00000000	0.00000000	0.00000000

Output 39.6.9 *continued*

MANOVA Test Criteria and Exact F Statistics for the Hypothesis of No Overall Llanederyn vs. the rest Effect
H = Contrast SSCP Matrix for Llanederyn vs. the rest
E = Error SSCP Matrix

S=1 M=1.5 N=8

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.05839360	58.05	5	18	<.0001
Pillai's Trace	0.94160640	58.05	5	18	<.0001
Hotelling-Lawley Trace	16.12516462	58.05	5	18	<.0001
Roy's Greatest Root	16.12516462	58.05	5	18	<.0001

Example 39.7: Repeated Measures Analysis of Variance

This example uses data from Cole and Grizzle (1966) to illustrate a commonly occurring repeated measures ANOVA design. Sixteen dogs are randomly assigned to four groups. (One animal is removed from the analysis due to a missing value for one dependent variable.) Dogs in each group receive either morphine or trimethaphan (variable Drug) and have either depleted or intact histamine levels (variable Depleted) before receiving the drugs. The dependent variable is the blood concentration of histamine at 0, 1, 3, and 5 minutes after injection of the drug. Logarithms are applied to these concentrations to minimize correlation between the mean and the variance of the data.

The following SAS statements perform both univariate and multivariate repeated measures analyses and produce Output 39.7.1 through Output 39.7.7.

```
data dogs;
   input Drug $12. Depleted $ Histamine0 Histamine1
         Histamine3 Histamine5;
   LogHistamine0=log(Histamine0);
   LogHistamine1=log(Histamine1);
   LogHistamine3=log(Histamine3);
   LogHistamine5=log(Histamine5);
   datalines;
Morphine      N  .04  .20  .10  .08
Morphine      N  .02  .06  .02  .02
Morphine      N  .07 1.40  .48  .24
Morphine      N  .17  .57  .35  .24
Morphine      Y  .10  .09  .13  .14
Morphine      Y  .12  .11  .10   .
Morphine      Y  .07  .07  .06  .07
Morphine      Y  .05  .07  .06  .07
Trimethaphan  N  .03  .62  .31  .22
Trimethaphan  N  .03 1.05  .73  .60
Trimethaphan  N  .07  .83 1.07  .80
Trimethaphan  N  .09 3.13 2.06 1.23
Trimethaphan  Y  .10  .09  .09  .08
```

```
Trimethaphan  Y  .08  .09  .09  .10
Trimethaphan  Y  .13  .10  .12  .12
Trimethaphan  Y  .06  .05  .05  .05
;

proc glm;
   class Drug Depleted;
   model LogHistamine0--LogHistamine5 =
         Drug Depleted Drug*Depleted / nouni;
   repeated Time 4 (0 1 3 5) polynomial / summary printe;
run;
```

The NOUNI option in the MODEL statement suppresses the individual ANOVA tables for the original dependent variables. These analyses are usually of no interest in a repeated measures analysis. The POLYNOMIAL option in the REPEATED statement indicates that the transformation used to implement the repeated measures analysis is an orthogonal polynomial transformation, and the SUMMARY option requests that the univariate analyses for the orthogonal polynomial contrast variables be displayed. The parenthetical numbers (0 1 3 5) determine the spacing of the orthogonal polynomials used in the analysis.

Output 39.7.1 Summary Information about Groups

```
                          The GLM Procedure

                       Class Level Information

              Class         Levels    Values

              Drug               2    Morphine Trimethaphan

              Depleted           2    N Y

                 Number of Observations Read          16
                 Number of Observations Used          15
```

The "Repeated Measures Level Information" table gives information about the repeated measures effect; it is displayed in Output 39.7.2. In this example, the within-subject (within-dog) effect is Time, which has the levels 0, 1, 3, and 5.

Output 39.7.2 Repeated Measures Levels

```
                          The GLM Procedure
                 Repeated Measures Analysis of Variance

                  Repeated Measures Level Information

                                Log        Log        Log        Log
      Dependent Variable    Histamine0 Histamine1 Histamine3 Histamine5

           Level of Time             0          1          3          5
```

The multivariate analyses for within-subject effects and related interactions are displayed in Output 39.7.3. For the example, the first table displayed shows that the TIME effect is significant. In addition, the Time*Drug*Depleted interaction is significant, as shown in the fourth table. This means that the effect of Time on the blood concentration of histamine is different for the four Drug*Depleted combinations studied.

Output 39.7.3 Multivariate Tests of Within-Subject Effects

MANOVA Test Criteria and Exact F Statistics for the Hypothesis of no Time Effect
H = Type III SSCP Matrix for Time
E = Error SSCP Matrix

S=1 M=0.5 N=3.5

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.11097706	24.03	3	9	0.0001
Pillai's Trace	0.88902294	24.03	3	9	0.0001
Hotelling-Lawley Trace	8.01087137	24.03	3	9	0.0001
Roy's Greatest Root	8.01087137	24.03	3	9	0.0001

MANOVA Test Criteria and Exact F Statistics
for the Hypothesis of no Time*Drug Effect
H = Type III SSCP Matrix for Time*Drug
E = Error SSCP Matrix

S=1 M=0.5 N=3.5

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.34155984	5.78	3	9	0.0175
Pillai's Trace	0.65844016	5.78	3	9	0.0175
Hotelling-Lawley Trace	1.92774470	5.78	3	9	0.0175
Roy's Greatest Root	1.92774470	5.78	3	9	0.0175

MANOVA Test Criteria and Exact F Statistics for
the Hypothesis of no Time*Depleted Effect
H = Type III SSCP Matrix for Time*Depleted
E = Error SSCP Matrix

S=1 M=0.5 N=3.5

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.12339988	21.31	3	9	0.0002
Pillai's Trace	0.87660012	21.31	3	9	0.0002
Hotelling-Lawley Trace	7.10373567	21.31	3	9	0.0002
Roy's Greatest Root	7.10373567	21.31	3	9	0.0002

Output 39.7.3 *continued*

MANOVA Test Criteria and Exact F Statistics for the Hypothesis of no Time*Drug*Depleted Effect
H = Type III SSCP Matrix for Time*Drug*Depleted
E = Error SSCP Matrix

S=1 M=0.5 N=3.5

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.19383010	12.48	3	9	0.0015
Pillai's Trace	0.80616990	12.48	3	9	0.0015
Hotelling-Lawley Trace	4.15915732	12.48	3	9	0.0015
Roy's Greatest Root	4.15915732	12.48	3	9	0.0015

Output 39.7.4 displays tests of hypotheses for between-subject (between-dog) effects. This section tests the hypotheses that the different Drugs, Depleteds, and their interactions have no effects on the dependent variables, while ignoring the within-dog effects. From this analysis, there is a significant between-dog effect for Depleted (p-value=0.0229). The interaction and the main effect for Drug are not significant (p-values=0.1734 and 0.1281, respectively).

Output 39.7.4 Tests of Between-Subject Effects

The GLM Procedure
Repeated Measures Analysis of Variance
Tests of Hypotheses for Between Subjects Effects

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Drug	1	5.99336243	5.99336243	2.71	0.1281
Depleted	1	15.44840703	15.44840703	6.98	0.0229
Drug*Depleted	1	4.69087508	4.69087508	2.12	0.1734
Error	11	24.34683348	2.21334850		

Univariate analyses for within-subject (within-dog) effects and related interactions are displayed in Output 39.7.6. The results for this example are the same as for the multivariate analyses; this is not always the case. In addition, before the univariate analyses are used to make conclusions about the data, the result of the sphericity test (requested with the PRINTE option in the REPEATED statement and displayed in Output 39.7.5) should be examined. If the sphericity test is rejected, use the adjusted G-G or H-F probabilities. See the section "Repeated Measures Analysis of Variance" on page 2532 for more information.

Output 39.7.5 Sphericity Test

Sphericity Tests

Variables	DF	Mauchly's Criterion	Chi-Square	Pr > ChiSq
Transformed Variates	5	0.1752641	16.930873	0.0046
Orthogonal Components	5	0.1752641	16.930873	0.0046

Output 39.7.6 Univariate Tests of Within-Subject Effects

The GLM Procedure
Repeated Measures Analysis of Variance
Univariate Tests of Hypotheses for Within Subject Effects

Source	DF	Type III SS	Mean Square	F Value	Pr > F	Adj Pr > F G - G	Adj Pr > F H - F
Time	3	12.05898677	4.01966226	53.44	<.0001	<.0001	<.0001
Time*Drug	3	1.84429514	0.61476505	8.17	0.0003	0.0039	0.0008
Time*Depleted	3	12.08978557	4.02992852	53.57	<.0001	<.0001	<.0001
Time*Drug*Depleted	3	2.93077939	0.97692646	12.99	<.0001	0.0005	<.0001
Error(Time)	33	2.48238887	0.07522391				

Greenhouse-Geisser Epsilon	0.5694
Huynh-Feldt Epsilon	0.8475

Output 39.7.7 is produced by the SUMMARY option in the REPEATED statement. If the POLYNOMIAL option is not used, a similar table is displayed using the default CONTRAST transformation. The linear, quadratic, and cubic trends for Time, labeled as 'Time_1', 'Time_2', and 'Time_3', are displayed, and in each case, the Source labeled 'Mean' gives a test for the respective trend.

Output 39.7.7 Tests of Between-Subject Effects for Transformed Variables

The GLM Procedure
Repeated Measures Analysis of Variance
Analysis of Variance of Contrast Variables

Time_N represents the nth degree polynomial contrast for Time

Contrast Variable: Time_1

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	2.00963483	2.00963483	34.99	0.0001
Drug	1	1.18069076	1.18069076	20.56	0.0009
Depleted	1	1.36172504	1.36172504	23.71	0.0005
Drug*Depleted	1	2.04346848	2.04346848	35.58	<.0001
Error	11	0.63171161	0.05742833		

Output 39.7.7 *continued*

Contrast Variable: Time_2

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	5.40988418	5.40988418	57.15	<.0001
Drug	1	0.59173192	0.59173192	6.25	0.0295
Depleted	1	5.94945506	5.94945506	62.86	<.0001
Drug*Depleted	1	0.67031587	0.67031587	7.08	0.0221
Error	11	1.04118707	0.09465337		

Contrast Variable: Time_3

Source	DF	Type III SS	Mean Square	F Value	Pr > F
Mean	1	4.63946776	4.63946776	63.04	<.0001
Drug	1	0.07187246	0.07187246	0.98	0.3443
Depleted	1	4.77860547	4.77860547	64.94	<.0001
Drug*Depleted	1	0.21699504	0.21699504	2.95	0.1139
Error	11	0.80949018	0.07359002		

Example 39.8: Mixed Model Analysis of Variance with the RANDOM Statement

Milliken and Johnson (1984) present an example of an unbalanced mixed model. Three machines, which are considered as a fixed effect, and six employees, which are considered a random effect, are studied. Each employee operates each machine for either one, two, or three different times. The dependent variable is an overall rating, which takes into account the number and quality of components produced.

The following statements form the data set and perform a mixed model analysis of variance by requesting the TEST option in the RANDOM statement. Note that the machine*person interaction is declared as a random effect; in general, when an interaction involves a random effect, it too should be declared as random. The results of the analysis are shown in Output 39.8.1 through Output 39.8.4.

```
data machine;
   input machine person rating @@;
   datalines;
1 1 52.0  1 2 51.8  1 2 52.8  1 3 60.0  1 4 51.1  1 4 52.3  1 5 50.9
1 5 51.8  1 5 51.4  1 6 46.4  1 6 44.8  1 6 49.2  2 1 64.0  2 2 59.7
2 2 60.0  2 2 59.0  2 3 68.6  2 3 65.8  2 4 63.2  2 4 62.8  2 4 62.2
2 5 64.8  2 5 65.0  2 6 43.7  2 6 44.2  2 6 43.0  3 1 67.5  3 1 67.2
3 1 66.9  3 2 61.5  3 2 61.7  3 2 62.3  3 3 70.8  3 3 70.6  3 3 71.0
3 4 64.1  3 4 66.2  3 4 64.0  3 5 72.1  3 5 72.0  3 5 71.1  3 6 62.0
3 6 61.4  3 6 60.5
;
```

```
proc glm data=machine;
   class machine person;
   model rating=machine person machine*person;
   random person machine*person / test;
run;
```

The TEST option in the RANDOM statement requests that PROC GLM determine the appropriate F tests based on person and machine*person being treated as random effects. As you can see in Output 39.8.4, this requires that a linear combination of mean squares be constructed to test both the machine and person hypotheses; thus, F tests that use Satterthwaite approximations are needed.

Output 39.8.1 Summary Information about Groups

The GLM Procedure

Class Level Information

Class	Levels	Values
machine	3	1 2 3
person	6	1 2 3 4 5 6

Number of Observations Read	44
Number of Observations Used	44

Output 39.8.2 Fixed-Effect Model Analysis of Variance

The GLM Procedure

Dependent Variable: rating

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	17	3061.743333	180.102549	206.41	<.0001
Error	26	22.686667	0.872564		
Corrected Total	43	3084.430000			

R-Square	Coeff Var	Root MSE	rating Mean
0.992645	1.560754	0.934111	59.85000

Source	DF	Type I SS	Mean Square	F Value	Pr > F
machine	2	1648.664722	824.332361	944.72	<.0001
person	5	1008.763583	201.752717	231.22	<.0001
machine*person	10	404.315028	40.431503	46.34	<.0001

Output 39.8.2 *continued*

Source	DF	Type III SS	Mean Square	F Value	Pr > F
machine	2	1238.197626	619.098813	709.52	<.0001
person	5	1011.053834	202.210767	231.74	<.0001
machine*person	10	404.315028	40.431503	46.34	<.0001

Output 39.8.3 Expected Values of Type III Mean Squares

Source	Type III Expected Mean Square
machine	Var(Error) + 2.137 Var(machine*person) + Q(machine)
person	Var(Error) + 2.2408 Var(machine*person) + 6.7224 Var(person)
machine*person	Var(Error) + 2.3162 Var(machine*person)

Output 39.8.4 Mixed Model Analysis of Variance

The GLM Procedure
Tests of Hypotheses for Mixed Model Analysis of Variance

Dependent Variable: rating

Source	DF	Type III SS	Mean Square	F Value	Pr > F
machine	2	1238.197626	619.098813	16.57	0.0007
Error	10.036	375.057436	37.370384		
Error: 0.9226*MS(machine*person) + 0.0774*MS(Error)					

Source	DF	Type III SS	Mean Square	F Value	Pr > F
person	5	1011.053834	202.210767	5.17	0.0133
Error	10.015	392.005726	39.143708		
Error: 0.9674*MS(machine*person) + 0.0326*MS(Error)					

Source	DF	Type III SS	Mean Square	F Value	Pr > F
machine*person	10	404.315028	40.431503	46.34	<.0001
Error: MS(Error)	26	22.686667	0.872564		

Note that you can also use the MIXED procedure to analyze mixed models. The following statements use PROC MIXED to reproduce the mixed model analysis of variance; the relevant part of the PROC MIXED results is shown in Output 39.8.5.

```
proc mixed data=machine method=type3;
   class machine person;
   model rating = machine;
   random person machine*person;
run;
```

Output 39.8.5 PROC MIXED Mixed Model Analysis of Variance (Partial Output)

```
                              The Mixed Procedure

                          Type 3 Analysis of Variance

                                                Sum of
              Source                DF         Squares     Mean Square

              machine                2     1238.197626      619.098813
              person                 5     1011.053834      202.210767
              machine*person        10      404.315028       40.431503
              Residual              26       22.686667        0.872564

                          Type 3 Analysis of Variance

Source           Expected Mean Square

machine          Var(Residual) + 2.137 Var(machine*person) + Q(machine)
person           Var(Residual) + 2.2408 Var(machine*person) + 6.7224 Var(person)
machine*person   Var(Residual) + 2.3162 Var(machine*person)
Residual         Var(Residual)

                          Type 3 Analysis of Variance

                                                                Error
Source           Error Term                                        DF  F Value  Pr > F

machine          0.9226 MS(machine*person)                     10.036    16.57  0.0007
                 + 0.0774 MS(Residual)
person           0.9674 MS(machine*person)                     10.015     5.17  0.0133
                 + 0.0326 MS(Residual)
machine*person   MS(Residual)                                      26    46.34  <.0001
Residual         .                                                  .      .     .
```

The advantage of PROC MIXED is that it offers more versatility for mixed models; the disadvantage is that it can be less computationally efficient for large data sets. See Chapter 56, "The MIXED Procedure," for more details.

Example 39.9: Analyzing a Doubly Multivariate Repeated Measures Design

This example shows how to analyze a doubly multivariate repeated measures design by using PROC GLM with an IDENTITY factor in the REPEATED statement. Note that this differs from previous

releases of PROC GLM, in which you had to use a MANOVA statement to get a doubly repeated measures analysis.

Two responses, Y1 and Y2, are each measured three times for each subject (pretreatment, posttreatment, and in a later follow-up). Each subject receives one of three treatments; A, B, or the control. In PROC GLM, you use a REPEATED factor of type IDENTITY to identify the different responses and another repeated factor to identify the different measurement times. The repeated measures analysis includes multivariate tests for time and treatment main effects, as well as their interactions, across responses. The following statements produce Output 39.9.1 through Output 39.9.3.

```
options ls=96;
data Trial;
   input Treatment $ Repetition PreY1 PostY1 FollowY1
                                PreY2 PostY2 FollowY2;
   datalines;
A        1  3  13  9  0  0  9
A        2  0  14 10  6  6  3
A        3  4   6 17  8  2  6
A        4  7   7 13  7  6  4
A        5  3  12 11  6 12  6
A        6 10  14  8 13  3  8
B        1  9  11 17  8 11 27
B        2  4  16 13  9  3 26
B        3  8  10  9 12  0 18
B        4  5   9 13  3  0 14
B        5  0  15 11  3  0 25
B        6  4  11 14  4  2  9
Control  1 10  12 15  4  3  7
Control  2  2   8 12  8  7 20
Control  3  4   9 10  2  0 10
Control  4 10   8  8  5  8 14
Control  5 11  11 11  1  0 11
Control  6  1  5  15  8  9 10
;

proc glm data=Trial;
   class Treatment;
   model PreY1 PostY1 FollowY1
         PreY2 PostY2 FollowY2 = Treatment / nouni;
   repeated Response 2 identity, Time 3;
run;
```

Output 39.9.1 A Doubly Multivariate Repeated Measures Design

```
                          The GLM Procedure

                       Class Level Information

                 Class          Levels    Values

                 Treatment           3    A B Control
```

Output 39.9.1 *continued*

Number of Observations Read	18
Number of Observations Used	18

The levels of the repeated factors are displayed in Output 39.9.2. Note that RESPONSE is 1 for all the Y1 measurements and 2 for all the Y2 measurements, while the three levels of Time identify the pretreatment, posttreatment, and follow-up measurements within each response. The multivariate tests for within-subject effects are displayed in Output 39.9.3.

Output 39.9.2 Repeated Factor Levels

The GLM Procedure
Repeated Measures Analysis of Variance

Repeated Measures Level Information

Dependent Variable	PreY1	PostY1	FollowY1	PreY2	PostY2	FollowY2
Level of Response	1	1	1	2	2	2
Level of Time	1	2	3	1	2	3

Output 39.9.3 Within-Subject Tests

MANOVA Test Criteria and Exact F Statistics for the Hypothesis of no Response Effect
H = Type III SSCP Matrix for Response
E = Error SSCP Matrix

S=1 M=0 N=6

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.02165587	316.24	2	14	<.0001
Pillai's Trace	0.97834413	316.24	2	14	<.0001
Hotelling-Lawley Trace	45.17686368	316.24	2	14	<.0001
Roy's Greatest Root	45.17686368	316.24	2	14	<.0001

MANOVA Test Criteria and F Approximations for the Hypothesis of no Response*Treatment Effect
H = Type III SSCP Matrix for Response*Treatment
E = Error SSCP Matrix

S=2 M=-0.5 N=6

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.72215797	1.24	4	28	0.3178
Pillai's Trace	0.27937444	1.22	4	30	0.3240
Hotelling-Lawley Trace	0.38261660	1.31	4	15.818	0.3074
Roy's Greatest Root	0.37698780	2.83	2	15	0.0908

NOTE: F Statistic for Roy's Greatest Root is an upper bound.
NOTE: F Statistic for Wilks' Lambda is exact.

Output 39.9.3 *continued*

MANOVA Test Criteria and Exact F Statistics for the Hypothesis of no Response*Time Effect
H = Type III SSCP Matrix for Response*Time
E = Error SSCP Matrix

S=1 M=1 N=5

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.14071380	18.32	4	12	<.0001
Pillai's Trace	0.85928620	18.32	4	12	<.0001
Hotelling-Lawley Trace	6.10662362	18.32	4	12	<.0001
Roy's Greatest Root	6.10662362	18.32	4	12	<.0001

MANOVA Test Criteria and F Approximations for the
Hypothesis of no Response*Time*Treatment Effect
H = Type III SSCP Matrix for Response*Time*Treatment
E = Error SSCP Matrix

S=2 M=0.5 N=5

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.22861451	3.27	8	24	0.0115
Pillai's Trace	0.96538785	3.03	8	26	0.0151
Hotelling-Lawley Trace	2.52557514	3.64	8	15	0.0149
Roy's Greatest Root	2.12651905	6.91	4	13	0.0033

NOTE: F Statistic for Roy's Greatest Root is an upper bound.
NOTE: F Statistic for Wilks' Lambda is exact.

The table for Response*Treatment tests for an overall treatment effect across the two responses; likewise, the tables for Response*Time and Response*Treatment*Time test for time and the treatment-by-time interaction, respectively. In this case, there is a strong main effect for time and possibly for the interaction, but not for treatment.

In previous releases (before the IDENTITY transformation was introduced), in order to perform a doubly repeated measures analysis, you had to use a MANOVA statement with a customized transformation matrix M. You might still want to use this approach to see details of the analysis, such as the univariate ANOVA for each transformed variate. The following statements demonstrate this approach by using the MANOVA statement to test for the overall main effect of time and specifying the SUMMARY option.

```
proc glm data=Trial;
   class Treatment;
   model PreY1 PostY1 FollowY1
         PreY2 PostY2 FollowY2 = Treatment / nouni;
   manova  h=intercept  m=prey1 - posty1,
                          prey1 - followy1,
                          prey2 - posty2,
                          prey2 - followy2 / summary;
run;
```

The M matrix used to perform the test for time effects is displayed in Output 39.9.4, while the results of the multivariate test are given in Output 39.9.5. Note that the test results are the same as for the Response*Time effect in Output 39.9.3.

Output 39.9.4 M Matrix to Test for Time Effect (Repeated Measure)

The GLM Procedure
Multivariate Analysis of Variance

M Matrix Describing Transformed Variables

	PreY1	PostY1	FollowY1	PreY2	PostY2	FollowY2
MVAR1	1	−1	0	0	0	0
MVAR2	1	0	−1	0	0	0
MVAR3	0	0	0	1	−1	0
MVAR4	0	0	0	1	0	−1

Output 39.9.5 Tests for Time Effect (Repeated Measure)

The GLM Procedure
Multivariate Analysis of Variance

Characteristic Roots and Vectors of: E Inverse * H, where
H = Type III SSCP Matrix for Intercept
E = Error SSCP Matrix

Variables have been transformed by the M Matrix

Characteristic Root	Percent	Characteristic Vector V'EV=1 MVAR1	MVAR2	MVAR3	MVAR4
6.10662362	100.00	−0.00157729	0.04081620	−0.04210209	0.03519437
0.00000000	0.00	0.00796367	0.00493217	0.05185236	0.00377940
0.00000000	0.00	−0.03534089	−0.01502146	−0.00283074	0.04259372
0.00000000	0.00	−0.05672137	0.04500208	0.00000000	0.00000000

MANOVA Test Criteria and Exact F Statistics for the Hypothesis of No Overall Intercept Effect
on the Variables Defined by the M Matrix Transformation
H = Type III SSCP Matrix for Intercept
E = Error SSCP Matrix

S=1 M=1 N=5

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.14071380	18.32	4	12	<.0001
Pillai's Trace	0.85928620	18.32	4	12	<.0001
Hotelling-Lawley Trace	6.10662362	18.32	4	12	<.0001
Roy's Greatest Root	6.10662362	18.32	4	12	<.0001

The SUMMARY option in the MANOVA statement creates an ANOVA table for each transformed variable as defined by the M matrix. MVAR1 and MVAR2 contrast the pretreatment measurement for Y1 with the posttreatment and follow-up measurements for Y1, respectively; MVAR3 and MVAR4 are the same contrasts for Y2. Output 39.9.6 displays these univariate ANOVA tables and shows that the contrasts are all strongly significant except for the pre-versus-post difference for Y2.

Output 39.9.6 Summary Output for the Test for Time Effect

```
                              The GLM Procedure
                       Multivariate Analysis of Variance

Dependent Variable: MVAR1

      Source                      DF     Type III SS     Mean Square    F Value    Pr > F

      Intercept                    1     512.0000000     512.0000000      22.65    0.0003
      Error                       15     339.0000000      22.6000000

                              The GLM Procedure
                       Multivariate Analysis of Variance

Dependent Variable: MVAR2

      Source                      DF     Type III SS     Mean Square    F Value    Pr > F

      Intercept                    1     813.3888889     813.3888889      32.87    <.0001
      Error                       15     371.1666667      24.7444444

                              The GLM Procedure
                       Multivariate Analysis of Variance

Dependent Variable: MVAR3

      Source                      DF     Type III SS     Mean Square    F Value    Pr > F

      Intercept                    1      68.0555556      68.0555556       3.49    0.0814
      Error                       15     292.5000000      19.5000000

                              The GLM Procedure
                       Multivariate Analysis of Variance

Dependent Variable: MVAR4

      Source                      DF     Type III SS     Mean Square    F Value    Pr > F

      Intercept                    1     800.0000000     800.0000000      26.43    0.0001
      Error                       15     454.0000000      30.2666667
```

Example 39.10: Testing for Equal Group Variances

This example demonstrates how you can test for equal group variances in a one-way design. The data come from the University of Pennsylvania Smell Identification Test (UPSIT), reported in O'Brien and Heft (1995). The study is undertaken to explore how age and gender are related to sense of smell. A total of 180 subjects 20 to 89 years old are exposed to 40 different odors: for each odor, subjects are asked to choose which of four words best describes the odor. The Freeman-Tukey modified arcsine transformation (Bishop, Fienberg, and Holland 1975) is applied to the proportion of correctly identified odors to arrive at an olfactory index. For the following analysis, subjects are divided into five age groups:

$$\text{agegroup} = \begin{cases} 1 & \text{if} \quad \text{age} \le 25 \\ 2 & \text{if} \quad 25 < \text{age} \le 40 \\ 3 & \text{if} \quad 40 < \text{age} \le 55 \\ 4 & \text{if} \quad 55 < \text{age} \le 70 \\ 5 & \text{if} \quad 70 < \text{age} \end{cases}$$

The following statements create a data set named upsit, containing the age group and olfactory index for each subject.

```
data upsit;
   input agegroup smell @@;
   datalines;
1 1.381  1 1.322  1 1.162  1 1.275  1 1.381  1 1.275  1 1.322
1 1.492  1 1.322  1 1.381  1 1.162  1 1.013  1 1.322  1 1.322
1 1.275  1 1.492  1 1.322  1 1.322  1 1.492  1 1.322  1 1.381
1 1.234  1 1.162  1 1.381  1 1.381  1 1.381  1 1.322  1 1.381
1 1.322  1 1.381  1 1.275  1 1.492  1 1.275  1 1.322  1 1.275
1 1.381  1 1.234  1 1.105
2 1.234  2 1.234  2 1.381  2 1.322  2 1.492  2 1.234  2 1.381
2 1.381  2 1.492  2 1.492  2 1.275  2 1.492  2 1.381  2 1.492
2 1.322  2 1.275  2 1.275  2 1.275  2 1.322  2 1.492  2 1.381
2 1.322  2 1.492  2 1.196  2 1.322  2 1.275  2 1.234  2 1.322
2 1.098  2 1.322  2 1.381  2 1.275  2 1.492  2 1.492  2 1.381
2 1.196
3 1.381  3 1.381  3 1.492  3 1.492  3 1.492  3 1.098  3 1.492
3 1.381  3 1.234  3 1.234  3 1.129  3 1.069  3 1.234  3 1.322
3 1.275  3 1.230  3 1.234  3 1.234  3 1.322  3 1.322  3 1.381
4 1.322  4 1.381  4 1.381  4 1.322  4 1.234  4 1.234  4 1.234
4 1.381  4 1.322  4 1.275  4 1.275  4 1.492  4 1.234  4 1.098
4 1.322  4 1.129  4 0.687  4 1.322  4 1.322  4 1.234  4 1.129
4 1.492  4 0.810  4 1.234  4 1.381  4 1.040  4 1.381  4 1.381
4 1.129  4 1.492  4 1.129  4 1.098  4 1.275  4 1.322  4 1.234
4 1.196  4 1.234  4 0.585  4 0.785  4 1.275  4 1.322  4 0.712
4 0.810
5 1.322  5 1.234  5 1.381  5 1.275  5 1.275  5 1.322  5 1.162
5 0.909  5 0.502  5 1.234  5 1.322  5 1.196  5 0.859  5 1.196
5 1.381  5 1.322  5 1.234  5 1.275  5 1.162  5 1.162  5 0.585
5 1.013  5 0.960  5 0.662  5 1.129  5 0.531  5 1.162  5 0.737
5 1.098  5 1.162  5 1.040  5 0.558  5 0.960  5 1.098  5 0.884
5 1.162  5 1.098  5 0.859  5 1.275  5 1.162  5 0.785  5 0.859
;
```

Older people are more at risk for problems with their sense of smell, and this should be reflected in significant differences in the mean of the olfactory index across the different age groups. However, many older people also have an excellent sense of smell, which implies that the older age groups should have greater variability. In order to test this hypothesis and to compute a one-way ANOVA for the olfactory index that is robust to the possibility of unequal group variances, you can use the HOVTEST and WELCH options in the MEANS statement for the GLM procedure, as shown in the following statements.

```
proc glm data=upsit;
   class agegroup;
   model smell = agegroup;
   means agegroup / hovtest welch;
run;
```

Output 39.10.1, Output 39.10.2, and Output 39.10.3 display the usual ANOVA test for equal age group means, Levene's test for equal age group variances, and Welch's test for equal age group means, respectively. The hypotheses of age effects for mean and variance of the olfactory index are both confirmed.

Output 39.10.1 Usual ANOVA Test for Age Group Differences in Mean Olfactory Index

```
                          The GLM Procedure

Dependent Variable: smell

 Source                      DF     Type III SS     Mean Square   F Value   Pr > F

 agegroup                     4      2.13878141      0.53469535     16.65   <.0001
```

Output 39.10.2 Levene's Test for Age Group Differences in Olfactory Variability

```
                          The GLM Procedure

            Levene's Test for Homogeneity of smell Variance
             ANOVA of Squared Deviations from Group Means

                              Sum of          Mean
      Source           DF     Squares        Square    F Value    Pr > F

      agegroup          4      0.0799        0.0200       6.35    <.0001
      Error           175      0.5503       0.00314
```

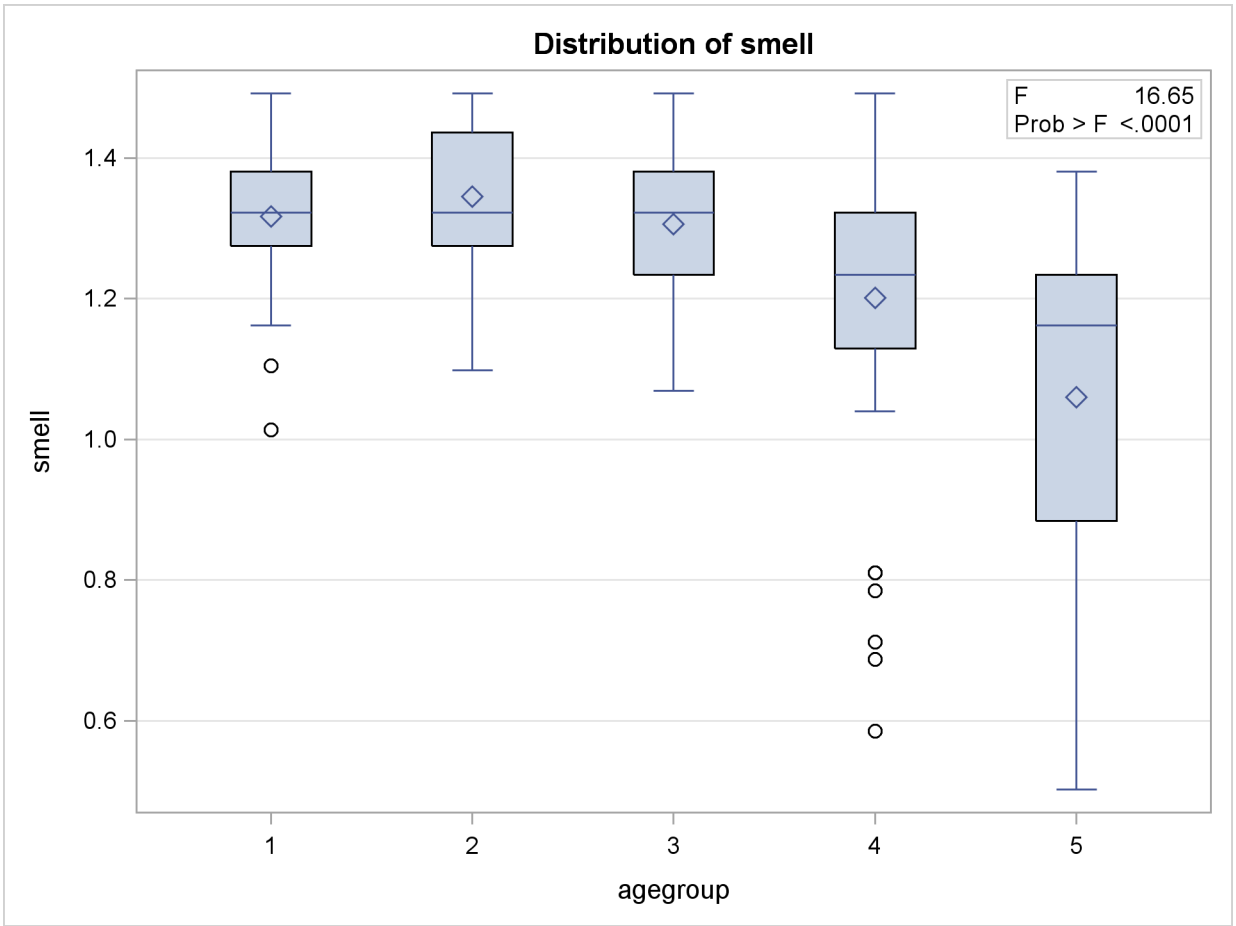
Output 39.10.3 Welch's Test for Age Group Differences in Mean Olfactory Index

```
                      Welch's ANOVA for smell

                Source            DF    F Value    Pr > F

                agegroup      4.0000      13.72    <.0001
                Error        78.7489
```

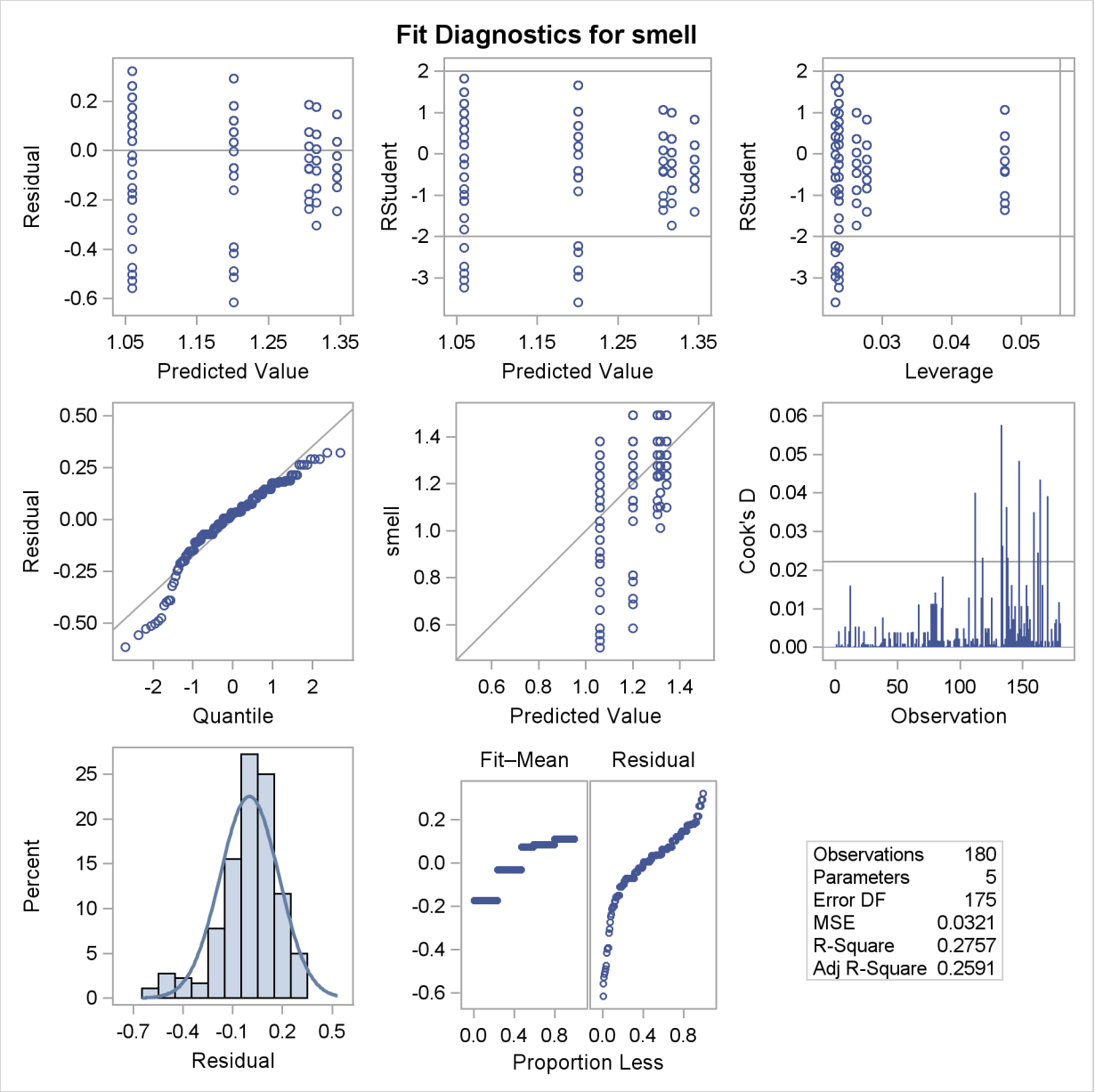
As discussed in "Homogeneity of Variance in One-Way Models" on page 2526, Levene's test or any other test for homogeneity of variance should not be used as a diagnostic for the assumption of equal group variances that underlies the usual analysis of variance. However, graphical diagnostics can be a useful informal tool for monitoring whether your data meet the assumptions of a GLM analysis. The following statements perform a one-way ANOVA as before, but with ODS Graphics enabled. In addition to the box plot that is produced by default, the PLOTS=DIAGNOSTICS option requests a panel of summary diagnostics for the fit. These additional plots are shown in Output 39.10.4 and Output 39.10.5.

```
ods graphics on;
proc glm data=upsit plot=diagnostics;
   class agegroup;
   model smell = agegroup;
run;
ods graphics off;
```

Output 39.10.4 Box Plot of Olfactory Index by Age Group



Output 39.10.5 Diagnostics for One-Way ANOVA of Olfactory Index by Age Group



Output 39.10.4 clearly shows different degrees of variability for olfactory index within different age groups, with the variability generally rising with age. Likewise, several of the plots in the diagnostics panel shown in Output 39.10.5 indicate a relationship between olfactory variability and mean olfactory index. Also, note that the plot of Cook's D statistic indicates that observations in the higher, more variable age groups are overly influential on the analysis of group means. The overall inference from these plots is that an assumption of equal group variances is probably untenable and that the analysis of the group means should thus take this into account.

Example 39.11: Analysis of a Screening Design

Yin and Jillie (1987) describe an experiment performed on a nitride etch process for a single wafer plasma etcher. The experiment is run using four factors: cathode power (power), gas flow (flow), reactor chamber pressure (pressure), and electrode gap (gap). Of interest are the main effects and interaction effects of the factors on the nitride etch rate (rate). The following statements create a SAS data set named HalfFraction, containing the factor settings and the observed etch rate for each of eight experimental runs.

```
data HalfFraction;
   input power flow pressure gap rate;
   datalines;
0.8   4.5 125 275     550
0.8   4.5 200 325     650
0.8 550.0 125 325     642
0.8 550.0 200 275     601
1.2   4.5 125 325     749
1.2   4.5 200 275    1052
1.2 550.0 125 275    1075
1.2 550.0 200 325     729
;
```

Notice that each of the factors has just two values. This is a common experimental design when the intent is to screen from the many factors that *might* affect the response the few that actually *do*. Since there are $2^4 = 16$ different possible settings of four two-level factors, this design with only eight runs is called a "half fraction." The eight runs are chosen specifically to provide unambiguous information on main effects at the cost of confounding interaction effects with each other.

One way to analyze these data is simply to use PROC GLM to compute an analysis of variance, including both main effects and interactions in the model. The following statements demonstrate this approach.

```
proc glm data=HalfFraction;
   class power flow pressure gap;
   model rate=power|flow|pressure|gap@2;
run;
```

The "`@2`" notation in the MODEL statement includes all main effects and two-factor interactions between the factors. The output is shown in Output 39.11.1.

Output 39.11.1 Analysis of Variance for Nitride Etch Process Half Fraction

The GLM Procedure

Class Level Information

Class	Levels	Values
power	2	0.8 1.2
flow	2	4.5 550
pressure	2	125 200
gap	2	275 325

Number of Observations Read 8
Number of Observations Used 8

The GLM Procedure

Dependent Variable: rate

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	280848.0000	40121.1429	.	.
Error	0	0.0000	.		
Corrected Total	7	280848.0000			

R-Square	Coeff Var	Root MSE	rate Mean
1.000000	.	.	756.0000

Source	DF	Type I SS	Mean Square	F Value	Pr > F
power	1	168780.5000	168780.5000	.	.
flow	1	264.5000	264.5000	.	.
power*flow	1	200.0000	200.0000	.	.
pressure	1	32.0000	32.0000	.	.
power*pressure	1	1300.5000	1300.5000	.	.
flow*pressure	1	78012.5000	78012.5000	.	.
gap	1	32258.0000	32258.0000	.	.
power*gap	0	0.0000	.	.	.
flow*gap	0	0.0000	.	.	.
pressure*gap	0	0.0000	.	.	.

Output 39.11.1 *continued*

Source	DF	Type III SS	Mean Square	F Value	Pr > F
power	1	168780.5000	168780.5000	.	.
flow	1	264.5000	264.5000	.	.
power*flow	0	0.0000	.	.	.
pressure	1	32.0000	32.0000	.	.
power*pressure	0	0.0000	.	.	.
flow*pressure	0	0.0000	.	.	.
gap	1	32258.0000	32258.0000	.	.
power*gap	0	0.0000	.	.	.
flow*gap	0	0.0000	.	.	.
pressure*gap	0	0.0000	.	.	.

Notice that there are no error degrees of freedom. This is because there are 10 effects in the model (4 main effects plus 6 interactions) but only 8 observations in the data set. This is another cost of using a fractional design: not only is it impossible to estimate all the main effects and interactions, but there is also no information left to estimate the underlying error rate in order to measure the significance of the effects that are estimable.

Another thing to notice in Output 39.11.1 is the difference between the Type I and Type III ANOVA tables. The rows corresponding to main effects in each are the same, but no Type III interaction tests are estimable, while some Type I interaction tests are estimable. This indicates that there is *aliasing* in the design: some interactions are completely confounded with each other.

In order to analyze this confounding, you should examine the aliasing structure of the design by using the ALIASING option in the MODEL statement. Before doing so, however, it is advisable to *code* the design, replacing low and high levels of each factor with the values -1 and $+1$, respectively. This puts each factor on an equal footing in the model and makes the aliasing structure much more interpretable. The following statements code the data, creating a new data set named Coded.

```
data Coded; set HalfFraction;
   power    = -1*(power   =0.80) + 1*(power   =1.20);
   flow     = -1*(flow    =4.50) + 1*(flow    =550 );
   pressure = -1*(pressure=125 ) + 1*(pressure=200 );
   gap      = -1*(gap     =275 ) + 1*(gap     =325 );
run;
```

The following statements use the GLM procedure to reanalyze the coded design, displaying the parameter estimates as well as the functions of the parameters that they each estimate.

```
proc glm data=Coded;
   model rate=power|flow|pressure|gap@2 / solution aliasing;
run;
```

The parameter estimates table is shown in Output 39.11.2.

Output 39.11.2 Parameter Estimates and Aliases for Nitride Etch Process Half Fraction

```
                                    The GLM Procedure

Dependent Variable: rate

                                    Standard
Parameter             Estimate         Error   t Value   Pr > |t|   Expected Value

Intercept          756.0000000             .       .        .       Intercept
power              145.2500000             .       .        .       power
flow                 5.7500000             .       .        .       flow
power*flow          -5.0000000 B           .       .        .       power*flow + pressure*gap
pressure             2.0000000             .       .        .       pressure
power*pressure     -12.7500000 B           .       .        .       power*pressure + flow*gap
flow*pressure      -98.7500000 B           .       .        .       flow*pressure + power*gap
gap                -63.5000000             .       .        .       gap
power*gap            0.0000000 B           .       .        .
flow*gap             0.0000000 B           .       .        .
pressure*gap         0.0000000 B           .       .        .
```

In the "Expected Value" column, notice that, while each of the main effects is unambiguously estimated by its associated term in the model, the expected values of the interaction estimates are more complicated. For example, the relatively large effect (−98.75) corresponding to flow*pressure actually estimates the combined effect of flow*pressure and power*gap. Without further information, it is impossible to disentangle these aliased interactions; however, since the main effects of both power and gap are large and those for flow and pressure are small, it is reasonable to suspect that power*gap is the more "active" of the two interactions.

Fortunately, eight more runs are available for this experiment (the other half fraction). The following statements create a data set containing these extra runs and add it to the previous eight, resulting in a full $2^4 = 16$ run replicate. Then PROC GLM displays the analysis of variance again.

```
data OtherHalf;
   input power flow pressure gap rate;
   datalines;
0.8   4.5 125 325     669
0.8   4.5 200 275     604
0.8 550.0 125 275     633
0.8 550.0 200 325     635
1.2   4.5 125 275    1037
1.2   4.5 200 325     868
1.2 550.0 125 325     860
1.2 550.0 200 275    1063
;
data FullRep;
   set HalfFraction OtherHalf;
run;

proc glm data=FullRep;
   class power flow pressure gap;
   model rate=power|flow|pressure|gap@2;
run;
```

The results are displayed in Output 39.11.3.

Output 39.11.3 Analysis of Variance for Nitride Etch Process Full Replicate

The GLM Procedure

Class Level Information

Class	Levels	Values
power	2	0.8 1.2
flow	2	4.5 550
pressure	2	125 200
gap	2	275 325

Number of Observations Read	16
Number of Observations Used	16

The GLM Procedure

Dependent Variable: rate

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	10	521234.1250	52123.4125	25.58	0.0011
Error	5	10186.8125	2037.3625		
Corrected Total	15	531420.9375			

R-Square	Coeff Var	Root MSE	rate Mean
0.980831	5.816175	45.13715	776.0625

Source	DF	Type I SS	Mean Square	F Value	Pr > F
power	1	374850.0625	374850.0625	183.99	<.0001
flow	1	217.5625	217.5625	0.11	0.7571
power*flow	1	18.0625	18.0625	0.01	0.9286
pressure	1	10.5625	10.5625	0.01	0.9454
power*pressure	1	1.5625	1.5625	0.00	0.9790
flow*pressure	1	7700.0625	7700.0625	3.78	0.1095
gap	1	41310.5625	41310.5625	20.28	0.0064
power*gap	1	94402.5625	94402.5625	46.34	0.0010
flow*gap	1	2475.0625	2475.0625	1.21	0.3206
pressure*gap	1	248.0625	248.0625	0.12	0.7414

Output 39.11.3 *continued*

Source	DF	Type III SS	Mean Square	F Value	Pr > F
power	1	374850.0625	374850.0625	183.99	<.0001
flow	1	217.5625	217.5625	0.11	0.7571
power*flow	1	18.0625	18.0625	0.01	0.9286
pressure	1	10.5625	10.5625	0.01	0.9454
power*pressure	1	1.5625	1.5625	0.00	0.9790
flow*pressure	1	7700.0625	7700.0625	3.78	0.1095
gap	1	41310.5625	41310.5625	20.28	0.0064
power*gap	1	94402.5625	94402.5625	46.34	0.0010
flow*gap	1	2475.0625	2475.0625	1.21	0.3206
pressure*gap	1	248.0625	248.0625	0.12	0.7414

With 16 runs, the analysis of variance tells the whole story: all effects are estimable and there are five degrees of freedom left over to estimate the underlying error. The main effects of power and gap and their interaction are all significant, and no other effects are. Notice that the Type I and Type III ANOVA tables are the same; this is because the design is orthogonal and all effects are estimable.

This example illustrates the use of the GLM procedure for the model analysis of a screening experiment. Typically, there is much more involved in performing an experiment of this type, from selecting the design points to be studied to graphically assessing significant effects, optimizing the final model, and performing subsequent experimentation. Specialized tools for this are available in SAS/QC software, in particular the ADX Interface and the FACTEX and OPTEX procedures. See *SAS/QC User's Guide* for more information.

References

Afifi, A. A. and Azen, S. P. (1972), *Statistical Analysis: A Computer-Oriented Approach*, New York: Academic Press.

Anderson, T. W. (1958), *An Introduction to Multivariate Statistical Analysis*, New York: John Wiley & Sons.

Bartlett, M. S. (1937), "Properties of Sufficiency and Statistical Tests," *Proceedings of the Royal Society of London, Series A*, 160, 268–282.

Begun, J. M. and Gabriel, K. R. (1981), "Closure of the Newman-Keuls Multiple Comparisons Procedure," *Journal of the American Statistical Association*, 76, 374.

Belsley, D. A., Kuh, E., and Welsch, R. E. (1980), *Regression Diagnostics*, New York: John Wiley & Sons.

Bishop, Y. M. M., Fienberg, S. E., and Holland, P. W. (1975), *Discrete Multivariate Analysis: Theory and Practice*, Cambridge, MA: MIT Press.

Box, G. E. P. (1953), "Non-normality and Tests on Variance," *Biometrika*, 40, 318–335.

Box, G. E. P. (1954), "Some Theorems on Quadratic Forms Applied in the Study of Analysis of Variance Problems, II. Effects of Inequality of Variance and of Correlation between Errors in the Two-Way Classification," *Annals of Mathematical Statistics*, 25, 484–498.

Brown, M. B. and Forsythe, A. B. (1974), "Robust Tests for Equality of Variances," *Journal of the American Statistical Association*, 69, 364–367.

Carmer, S. G. and Swanson, M. R. (1973), "Evaluation of Ten Pairwise Multiple Comparison Procedures by Monte-Carlo Methods," *Journal of the American Statistical Association*, 68, 66–74.

Cochran, W. G. and Cox, G. M. (1957), *Experimental Designs*, Second Edition, New York: John Wiley & Sons.

Cohen, J. (1988), *Statistical Power Analysis for the Behavioral Sciences*, Hillsdale, NJ: Erlbaum.

Cohen, R. (2002), "SAS Meets Big Iron: High Performance Computing in SAS Analytical Procedures," in *Proceedings of the Twenty-seventh Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Cole, J. W. L. and Grizzle, J. E. (1966), "Applications of Multivariate Analysis of Variance to Repeated Measures Experiments," *Biometrics*, 22, 810–828.

Conover, W. J., Johnson, M. E., and Johnson, M. M. (1981), "A Comparative Study of Tests for Homogeneity of Variances, with Applications to the Outer Continental Shelf Bidding Data," *Technometrics*, 23, 351–361.

Cornfield, J. and Tukey, J. W. (1956), "Average Values of Mean Squares in Factorials," *Annals of Mathematical Statistics*, 27, 907–949.

Draper, N. R. and Smith, H. (1966), *Applied Regression Analysis*, New York: John Wiley & Sons.

Duncan, D. B. (1975), "*t* Tests and Intervals for Comparisons Suggested by the Data," *Biometrics*, 31, 339–359.

Dunnett, C. W. (1955), "A Multiple Comparisons Procedure for Comparing Several Treatments with a Control," *Journal of the American Statistical Association*, 50, 1096–1121.

Dunnett, C. W. (1980), "Pairwise Multiple Comparisons in the Homogeneous Variance, Unequal Sample Size Case," *Journal of the American Statistical Association*, 75, 789–795.

Edwards, D. and Berry, J. J. (1987), "The Efficiency of Simulation-Based Multiple Comparisons," *Biometrics*, 43, 913–928.

Einot, I. and Gabriel, K. R. (1975), "A Study of the Powers of Several Methods of Multiple Comparisons," *Journal of the American Statistical Association*, 70, 351.

Fidler, F. and Thompson, B. (2001), "Computing Correct Confidence Intervals for ANOVA Fixed- and Random-Effects Effect Sizes," *Educational and Psychological Measurement*, 61, 575–604.

Freund, R. J., Littell, R. C., and Spector, P. C. (1986), *SAS System for Linear Models*, 1986 Edition, Cary, NC: SAS Institute Inc.

Gabriel, K. R. (1978), "A Simple Method of Multiple Comparisons of Means," *Journal of the American Statistical Association*, 73, 364.

Games, P. A. (1977), "An Improved *t* Table for Simultaneous Control on *g* Contrasts," *Journal of the American Statistical Association*, 72, 531–534.

Goodnight, J. H. (1976), "The New General Linear Models Procedure," in *Proceedings of the First Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Goodnight, J. H. (1978a), *The SWEEP Operator: Its Importance in Statistical Computing*, Technical Report R-106, SAS Institute Inc, Cary, NC.

Goodnight, J. H. (1978b), *Tests of the Hypotheses in Fixed-Effects Linear Models*, Technical Report R-101, SAS Institute Inc, Cary, NC.

Goodnight, J. H. (1979), "A Tutorial on the Sweep Operator," *The American Statistician*, 33, 149–158.

Goodnight, J. H. and Harvey, W. R. (1978), *Least-Squares Means in the Fixed-Effects General Linear Models*, Technical Report R-103, SAS Institute Inc, Cary, NC.

Goodnight, J. H. and Speed, F. M. (1978), *Computing Expected Mean Squares*, Technical Report R-102, SAS Institute Inc, Cary, NC.

Graybill, F. A. (1961), *An Introduction to Linear Statistical Models, Volume I*, New York: McGraw-Hill.

Greenhouse, S. W. and Geisser, S. (1959), "On Methods in the Analysis of Profile Data," *Psychometrika*, 32, 95–112.

Guirguis, G. and Tobias, R. (2004), "On the Computation of the Distribution for the Analysis of Means," *Communications in Statistics: Simulation and Computation*, 33.

Hand, D. J., Daly, F., Lunn, A. D., McConway, K. J., and Ostrowski, E. (1994), *A Handbook of Small Data Sets*, London: Chapman & Hall.

Hartley, H. O. and Searle, S. R. (1969), "On Interaction Variance Components in Mixed Models," *Biometrics*, 25, 573–576.

Harvey, W. R. (1975), *Least-Squares Analysis of Data with Unequal Subclass Numbers*, Report ARS H-4: USDA.

Hayter, A. J. (1984), "A Proof of the Conjecture That the Tukey-Kramer Method Is Conservative," *The Annals of Statistics*, 12, 61–75.

Hayter, A. J. (1989), "Pairwise Comparisons of Generally Correlated Means," *Journal of the American Statistical Association*, 84, 208–213.

Heck, D. L. (1960), "Charts of Some Upper Percentage Points of the Distribution of the Largest Characteristic Root," *Annals of Mathematical Statistics*, 31, 625–642.

Hochberg, Y. (1974), "Some Conservative Generalizations of the T-Method in Simultaneous Inference," *Journal of Multivariate Analysis*, 4, 224–234.

Hocking, R. R. (1973), "A Discussion of the Two-Way Mixed Model," *The American Statistician*, 27, 148–152.

Hocking, R. R. (1976), "The Analysis and Selection of Variables in a Linear Regression," *Biometrics*, 32, 1–50.

Hocking, R. R. (1985), *The Analysis of Linear Models*, Belmont, CA: Brooks/Cole.

Hsu, J. C. (1992), "The Factor Analytic Approach to Simultaneous Inference in the General Linear Model," *Journal of Computational and Graphical Statistics*, 1, 151–168.

Hsu, J. C. (1996), *Multiple Comparisons: Theory and Methods*, London: Chapman & Hall.

Hsu, J. C. and Nelson, B. (1998), "Multiple Comparisons in the General Linear Model," *Journal of Computational and Graphical Statistics*, 7, 23–41.

Huynh, H. and Feldt, L. S. (1970), "Conditions Under Which Mean Square Ratios in Repeated Measurements Designs Have Exact F-Distributions," *Journal of the American Statistical Association*, 65, 1582–1589.

Huynh, H. and Feldt, L. S. (1976), "Estimation of the Box Correction for Degrees of Freedom from Sample Data in the Randomized Block and Split Plot Designs," *Journal of Educational Statistics*, 1, 69–82.

Johnson, N. L., Kotz, S., and Balakrishnan, N. (1994), *Continuous Univariate Distributions-1*, Second Edition, New York: John Wiley & Sons.

Kennedy, W. J., Jr. and Gentle, J. E. (1980), *Statistical Computing*, New York: Marcel Dekker.

Kramer, C. Y. (1956), "Extension of Multiple Range Tests to Group Means with Unequal Numbers of Replications," *Biometrics*, 12, 307–310.

Krishnaiah, P. R. and Armitage, J. V. (1966), "Tables for Multivariate *t* Distribution," *Sankhya, Series B*, 31–56.

Kutner, M. H. (1974), "Hypothesis Testing in Linear Models (Eisenhart Model)," *American Statistician*, 28, 98–100.

LaTour, S. A. and Miniard, P. W. (1983), "The Misuse of Repeated Measures Analysis in Marketing Research," *Journal of Marketing Research*, 45–57.

Levene, H. (1960), "Robust Tests for the Equality of Variance," in I. Olkin, ed., *Contributions to Probability and Statistics*, 278–292, Palo Alto, CA: Stanford University Press.

Marcus, R., Peritz, E., and Gabriel, K. R. (1976), "On Closed Testing Procedures with Special Reference to Ordered Analysis of Variance," *Biometrika*, 63, 655–660.

Maxwell, S. E. (2000), "Sample Size and Multiple Regression Analysis," *Psychological Methods*, 5, 434–458.

McLean, R. A., Sanders, W. L., and Stroup, W. W. (1991), "A Unified Approach to Mixed Linear Models," *The American Statistician*, 45, 54–64.

Miller, R. G., Jr. (1981), *Simultaneous Statistical Inference*, New York: Springer-Verlag.

Milliken, G. A. and Johnson, D. E. (1984), *Analysis of Messy Data, Volume I: Designed Experiments*, Belmont, CA: Lifetime Learning Publications.

Morrison, D. F. (1976), *Multivariate Statistical Methods*, Second Edition, New York: McGraw-Hill.

Nelder, J. A. (1994), "The Statistics of Linear Models: Back to Basics," *Statistics and Computing*, 4.

Nelson, P. R. (1982), "Exact Critical Points for the Analysis of Means," *Communications in Statistics, Part A: Theory and Methods*, 699–709.

Nelson, P. R. (1991), "Numerical Evaluation of Multivariate Normal Integrals with Correlations $\rho_{lj} = -\alpha_l \alpha_j$," *The Frontiers of Statistical Scientific Theory & Industrial Applications*, 97–114.

Nelson, P. R. (1993), "Additional Uses for the Analysis of Means and Extended Tables of Critical Values," *Technometrics*, 35, 61–71.

O'Brien, R. G. (1979), "A General ANOVA Method for Robust Tests of Additive Models for Variances," *Journal of the American Statistical Association*, 74, 877–880.

O'Brien, R. G. (1981), "A Simple Test for Variance Effects in Experimental Designs," *Psychological Bulletin*, 89, 570–574.

O'Brien, R. G. and Heft, M. W. (1995), "New Discrimination Indexes and Models for Studying Sensory Functioning in Aging," *Journal of Applied Statistics*, 22, 9–27.

Olejnik, S. F. and Algina, J. (1987), "Type I Error Rates and Power Estimates of Selected Parametric and Non-parametric Tests of Scale," *Journal of Educational Statistics*, 12, 45–61.

Ott, E. R. (1967), "Analysis of Means—A Graphical Procedure," *Industrial Quality Control*, 24, 101–109. Reprinted in *Journal of Quality Technology*, 15 (1983), 10–18.

Perlman, M. D. and Rasmussen, U. A. (1975), "Some Remarks on Estimating a Noncentrality Parameter," *Communications in Statistics*, 4, 455–468.

Petrinovich, L. F. and Hardyck, C. D. (1969), "Error Rates for Multiple Comparison Methods: Some Evidence Concerning the Frequency of Erroneous Conclusions," *Psychological Bulletin*, 71, 43–54.

Pillai, K. C. S. (1960), *Statistical Table for Tests of Multivariate Hypotheses*, Manila: The Statistical Center, University of Philippines.

Pringle, R. M. and Rayner, A. A. (1971), *Generalized Inverse Matrices with Applications to Statistics*, New York: Hafner Publishing.

Ramsey, P. H. (1978), "Power Differences between Pairwise Multiple Comparisons," *Journal of the American Statistical Association*, 73, 363.

Rao, C. R. (1965), *Linear Statistical Inference and Its Applications*, New York: John Wiley & Sons.

Rodriguez, R., Tobias, R., and Wolfinger, R. (1995), "Comments on J. A. Nelder 'The Statistics of Linear Models: Back to Basics'," *Statistics and Computing*, 5, 97–101.

Ryan, T. A. (1959), "Multiple Comparisons in Psychological Research," *Psychological Bulletin*, 56, 26–47.

Ryan, T. A. (1960), "Significance Tests for Multiple Comparison of Proportions, Variances, and Other Statistics," *Psychological Bulletin*, 57, 318–328.

Satterthwaite, F. E. (1946), "An Approximate Distribution of Estimates of Variance Components," *Biometrics Bulletin*, 2, 110–114.

Schatzoff, M. (1966), "Exact Distributions of Wilks' Likelihood Ratio Criterion," *Biometrika*, 53, 347–358.

Scheffé, H. (1953), "A Method for Judging All Contrasts in the Analysis of Variance," *Biometrika*, 40, 87–104.

Scheffé, H. (1959), *The Analysis of Variance*, New York: John Wiley & Sons.

Searle, S. R. (1971), *Linear Models*, New York: John Wiley & Sons.

Searle, S. R. (1987), *Linear Models for Unbalanced Data*, New York: John Wiley & Sons.

Searle, S. R. (1995), "Comments on J. A. Nelder, 'The Statistics of Linear Models: Back to Basics'," *Statistics and Computing*, 5, 103–107.

Searle, S. R., Casella, G., and McCulloch, C. E. (1992), *Variance Components*, New York: John Wiley & Sons.

Searle, S. R., Speed, F. M., and Milliken, G. A. (1980), "Population Marginal Means in the Linear Model: An Alternative to Least Squares Means," *The American Statistician*, 34, 216–221.

Šidák (1967), "Rectangular Confidence Regions for the Means of Multivariate Normal Distributions," *Journal of the American Statistical Association*, 62, 626–633.

Smithson, M. (2003), *Confidence Intervals*, Thousand Oaks, CA: Sage Publications.

Smithson, M. (2004), personal communication.

Snedecor, G. W. and Cochran, W. G. (1967), *Statistical Methods*, Sixth Edition, Ames: Iowa State University Press.

Steel, R. G. D. and Torrie, J. H. (1960), *Principles and Procedures of Statistics*, New York: McGraw-Hill.

Steiger, J. H. and Fouladi, R. T. (1997), "Noncentrality Interval Estimation and the Evaluation of Statistical Models," in L. Harlow, S. Mulaik, and J. H. Steiger, eds., *What If There Were No Significance Tests?*, 222–257, Hillsdale, NJ: Erlbaum.

Stenstrom, F. H. (1940), *The Growth of Snapdragons, Stocks, Cinerarias and Carnations on Six Iowa Soils*, Master's thesis, Iowa State College.

Tubb, A., Parker, A. J., and Nickless, G. (1980), "The Analysis of Romano-British Pottery by Atomic Absorption Spectrophotometry," *Archaeometry*, 22, 153–171.

Tukey, J. W. (1952), "Allowances for Various Types of Error Rates," Unpublished invited address presented at Blacksburg meeting of Institute of Mathematical Studies.

Tukey, J. W. (1953), "The Problem of Multiple Comparisons," in H. I. Braun, ed., *The Collected Works of John W. Tukey*, volume 8, 1994, New York: Chapman & Hall.

Urquardt, N. S. (1968), "Computation of Generalized Inverse Matrices Which Satisfy Specific Conditions," *SIAM Review*, 10(2), 216–218.

Waller, R. A. and Duncan, D. B. (1969), "A Bayes Rule for the Symmetric Multiple Comparison Problem," *Journal of the American Statistical Association*, 64, 1484–1499.

Waller, R. A. and Duncan, D. B. (1972), "Corrigenda to 'A Bayes Rule for the Symmetric Multiple Comparison Problem'," *Journal of the American Statistical Association*, 67, 253–255.

Waller, R. A. and Kemp, K. E. (1976), "Computations of Bayesian t-Values for Multiple Comparisons," *Journal of Statistical Computation and Simulation*, 75, 169–172.

Welch, B. L. (1951), "On the Comparison of Several Mean Values: An Alternative Approach," *Biometrika*, 38, 330–336.

Welsch, R. E. (1977), "Stepwise Multiple Comparison Procedures," *Journal of the American Statistical Association*, 72, 359.

Westfall, P. J. and Young, S. S. (1993), *Resampling-Based Multiple Testing*, New York: John Wiley & Sons.

Winer, B. J. (1971), *Statistical Principles in Experimental Design*, Second Edition, New York: McGraw-Hill.

Wolfinger, R. D. and Chang, M. (1995), "Comparing the SAS GLM and MIXED Procedures for Repeated Measures," in *Proceedings of the Twentieth Annual SAS Users Group Conference*, Cary, NC: SAS Institute Inc.

Yin, G. Z. and Jillie, D. W. (1987), "Orthogonal Design for Process Optimization and Its Application in Plasma Etching," *Solid State Technology*, May, 127–132.

Chapter 40

The GLMMOD Procedure

Contents

Overview: GLMMOD Procedure **2621**
Getting Started: GLMMOD Procedure **2622**
A One-Way Design 2622
Syntax: GLMMOD Procedure **2626**
PROC GLMMOD Statement 2626
BY Statement 2628
CLASS Statement 2628
FREQ and WEIGHT Statements 2629
MODEL Statement 2629
Details: GLMMOD Procedure **2629**
Displayed Output 2629
Missing Values 2630
OUTPARM= Data Set 2630
OUTDESIGN= Data Set 2631
ODS Table Names 2631
Examples: GLMMOD Procedure **2632**
Example 40.1: A Two-Way Design 2632
Example 40.2: Factorial Screening 2637
References **2639**

Overview: GLMMOD Procedure

The GLMMOD procedure constructs the design matrix for a general linear model; it essentially constitutes the model-building front end for the GLM procedure. You can use the GLMMOD procedure in conjunction with other SAS/STAT software regression procedures or with SAS/IML software to obtain specialized analyses for general linear models that you cannot obtain with the GLM procedure.

While some of the regression procedures in SAS/STAT software provide for general linear effects modeling with classification variables and interaction or polynomial effects, many others do not. For such procedures, you must specify the model directly in terms of distinct variables. For example, if you want to use the REG procedure to fit a polynomial model, you must first create the crossproduct

and power terms as new variables, usually in a DATA step. Alternatively, you can use the GLMMOD procedure to create a data set that contains the design matrix for a model as specified using the effects modeling facilities of the GLM procedure.

Note that the TRANSREG procedure provides alternative methods to construct design matrices for full-rank and less-than-full-rank models, polynomials, and splines. See Chapter 90, "The TRANSREG Procedure," for more information.

Getting Started: GLMMOD Procedure

A One-Way Design

A one-way analysis of variance considers one treatment factor with two or more treatment levels. This example employs PROC GLMMOD together with PROC REG to perform a one-way analysis of variance to study the effect of bacteria on the nitrogen content of red clover plants. The treatment factor is bacteria strain, and it has six levels. Red clover plants are inoculated with the treatments, and nitrogen content is later measured in milligrams. The data are derived from an experiment by Erdman (1946) and are analyzed in Chapters 7 and 8 of Steel and Torrie (1980). PROC GLMMOD is used to create the design matrix. The following DATA step creates the SAS data set Clover.

```
title 'Nitrogen Content of Red Clover Plants';
data Clover;
   input Strain $ Nitrogen @@;
   datalines;
3DOK1  19.4 3DOK1  32.6 3DOK1  27.0 3DOK1  32.1 3DOK1  33.0
3DOK5  17.7 3DOK5  24.8 3DOK5  27.9 3DOK5  25.2 3DOK5  24.3
3DOK4  17.0 3DOK4  19.4 3DOK4   9.1 3DOK4  11.9 3DOK4  15.8
3DOK7  20.7 3DOK7  21.0 3DOK7  20.5 3DOK7  18.8 3DOK7  18.6
3DOK13 14.3 3DOK13 14.4 3DOK13 11.8 3DOK13 11.6 3DOK13 14.2
COMPOS 17.3 COMPOS 19.4 COMPOS 19.1 COMPOS 16.9 COMPOS 20.8
;
```

The variable Strain contains the treatment levels, and the variable Nitrogen contains the response. The following statements produce the design matrix:

```
proc glmmod data=Clover;
   class Strain;
   model Nitrogen = Strain;
run;
```

The classification variable, or treatment factor, is specified in the CLASS statement. The MODEL statement defines the response and independent variables. The design matrix produced corresponds to the model

$$Y_{i,j} = \mu + \alpha_i + \epsilon_{i,j}$$

where $i = 1, \ldots, 6$ and $j = 1, \ldots, 5$.

Figure 40.1 and Figure 40.2 display the output produced by these statements. Figure 40.1 displays information about the data set, which is useful for checking your data.

Figure 40.1 Class Level Information and Parameter Definitions

Nitrogen Content of Red Clover Plants

The GLMMOD Procedure

Class Level Information

Class	Levels	Values
Strain	6	3DOK1 3DOK13 3DOK4 3DOK5 3DOK7 COMPOS

Number of Observations Read	30
Number of Observations Used	30

Parameter Definitions

Column Number	Name of Associated Effect	CLASS Variable Values Strain
1	Intercept	
2	Strain	3DOK1
3	Strain	3DOK13
4	Strain	3DOK4
5	Strain	3DOK5
6	Strain	3DOK7
7	Strain	COMPOS

The design matrix, shown in Figure 40.2, consists of seven columns: one for the mean and six for the treatment levels. The vector of responses, Nitrogen, is also displayed.

Figure 40.2 Design Matrix

Design Points

Observation Number	Nitrogen	Column Number 1	2	3	4	5	6	7
1	19.4	1	1	0	0	0	0	0
2	32.6	1	1	0	0	0	0	0
3	27.0	1	1	0	0	0	0	0
4	32.1	1	1	0	0	0	0	0
5	33.0	1	1	0	0	0	0	0
6	17.7	1	0	0	0	1	0	0
7	24.8	1	0	0	0	1	0	0
8	27.9	1	0	0	0	1	0	0
9	25.2	1	0	0	0	1	0	0
10	24.3	1	0	0	0	1	0	0
11	17.0	1	0	0	1	0	0	0
12	19.4	1	0	0	1	0	0	0
13	9.1	1	0	0	1	0	0	0
14	11.9	1	0	0	1	0	0	0
15	15.8	1	0	0	1	0	0	0
16	20.7	1	0	0	0	0	1	0
17	21.0	1	0	0	0	0	1	0
18	20.5	1	0	0	0	0	1	0
19	18.8	1	0	0	0	0	1	0
20	18.6	1	0	0	0	0	1	0
21	14.3	1	0	1	0	0	0	0
22	14.4	1	0	1	0	0	0	0
23	11.8	1	0	1	0	0	0	0
24	11.6	1	0	1	0	0	0	0
25	14.2	1	0	1	0	0	0	0
26	17.3	1	0	0	0	0	0	1
27	19.4	1	0	0	0	0	0	1
28	19.1	1	0	0	0	0	0	1
29	16.9	1	0	0	0	0	0	1
30	20.8	1	0	0	0	0	0	1

Usually, you will find PROC GLMMOD most useful for the data sets it can create rather than for its displayed output. For example, the following statements use PROC GLMMOD to save the design matrix for the clover study to the data set CloverDesign instead of displaying it.

```
proc glmmod data=Clover outdesign=CloverDesign noprint;
   class Strain;
   model Nitrogen = Strain;
run;
```

Now you can use the REG procedure to analyze the data, as the following statements demonstrate:

```
proc reg data=CloverDesign;
   model Nitrogen = Col2-Col7;
run;
```

The results are shown in Figure 40.3.

Figure 40.3 Regression Analysis Using the REG Procedure

Nitrogen Content of Red Clover Plants

The REG Procedure
Model: MODEL1
Dependent Variable: Nitrogen

Number of Observations Read	30
Number of Observations Used	30

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	847.04667	169.40933	14.37	<.0001
Error	24	282.92800	11.78867		
Corrected Total	29	1129.97467			

Root MSE	3.43346	R-Square	0.7496
Dependent Mean	19.88667	Adj R-Sq	0.6975
Coeff Var	17.26515		

NOTE: Model is not full rank. Least-squares solutions for the parameters are not unique. Some statistics will be misleading. A reported DF of 0 or B means that the estimate is biased.
NOTE: The following parameters have been set to 0, since the variables are a linear combination of other variables as shown.

Col7 = Intercept - Col2 - Col3 - Col4 - Col5 - Col6

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	B	18.70000	1.53549	12.18	<.0001
Col2	Strain 3DOK1	B	10.12000	2.17151	4.66	<.0001
Col3	Strain 3DOK13	B	-5.44000	2.17151	-2.51	0.0194
Col4	Strain 3DOK4	B	-4.06000	2.17151	-1.87	0.0738
Col5	Strain 3DOK5	B	5.28000	2.17151	2.43	0.0229
Col6	Strain 3DOK7	B	1.22000	2.17151	0.56	0.5794
Col7	Strain COMPOS	0	0	.	.	.

Syntax: GLMMOD Procedure

The following statements are available in PROC GLMMOD.

PROC GLMMOD < *options* > ;
  BY *variables* ;
  CLASS *variables* ;
  FREQ *variable* ;
  MODEL *dependents=independents* / < *options* > ;
  WEIGHT *variable* ;

The PROC GLMMOD and MODEL statements are required. If classification effects are used, the classification variables must be declared in a CLASS statement, and the CLASS statement must appear before the MODEL statement.

PROC GLMMOD Statement

PROC GLMMOD < *options* > ;

The PROC GLMMOD statement invokes the GLMMOD procedure. It has the following options:

DATA=*SAS-data-set*
specifies the SAS data set to be used by the GLMMOD procedure. If you do not specify the DATA= option, the most recently created SAS data set is used.

NAMELEN=*n*
specifies the maximum length for an effect name. Effect names are listed in the table of parameter definitions and stored in the EFFNAME variable in the OUTPARM= data set. By default, $n = 20$. You can specify $20 < n \leq 200$ if 20 characters are not enough to distinguish between effects, which might be the case if the model includes a high-order interaction between variables with relatively long, similar names.

NOPRINT
suppresses the normal display of results. This option is generally useful only when one or more output data sets are being produced by the GLMMOD procedure. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

ORDER=DATA | FORMATTED | FREQ | INTERNAL
specifies the order in which you want the levels of the classification variables (specified in the CLASS statement) to be sorted. This ordering determines which parameters in the model correspond to each level in the data. Note that the ORDER= option applies to the levels for all classification variables. The exception is the default ORDER=FORMATTED for numeric variables for which you have supplied no explicit format. In this case, the levels are ordered by their internal (numeric) value. Note that this represents a change from previous releases

for how class levels are ordered. Before SAS 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values, and to revert to the previous ordering you can specify this format explicitly for the affected classification variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and usually required the user to intervene with an explicit format or ORDER=INTERNAL to get the more natural ordering.

The ORDER= option can take the following values.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

If you omit the ORDER= option, PROC GLMMOD orders by the external formatted value.

OUTPARM=*SAS-data-set*

names an output data set to contain the information regarding the association between model effects and design matrix columns.

OUTDESIGN=*SAS-data-set*

names an output data set to contain the columns of the design matrix.

PREFIX=*name*

specifies a prefix to use in naming the columns of the design matrix in the OUTDESIGN= data set. The default prefix is Col and the column name is formed by appending the column number to the prefix, so that by default the columns are named Col1, Col2, and so on. If you specify the ZEROBASED option, the column numbering starts at zero, so that with the default value of PREFIX= the columns of the design matrix in the OUTDESIGN= data set are named Col0, Col1, and so on.

ZEROBASED

specifies that the numbering for the columns of the design matrix in the OUTDESIGN= data set begin at 0. By default it begins at 1, so that with the default value of PREFIX= the columns of the design matrix in the OUTDESIGN= data set are named Col1, Col2, and so on. If you use the ZEROBASED option, the column names are instead Col0, Col1, and so on.

BY Statement

BY *variables* **;**

You can specify a BY statement with the GLMMOD procedure to obtain separate designs for observations in groups defined by the BY variables. When you specify a BY statement, the procedure expects the input DATA= data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the GLMMOD procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* < *option* > **;**

The CLASS statement names the classification variables to be used in the analysis. Typical classification variables are Treatment, Sex, Race, Group, and Replication. If you specify the CLASS statement, it must appear before the MODEL statement.

By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. Before SAS 9, class levels were determined using no more than the first 16 characters of the formatted values. If you want to revert to this previous behavior you can use the TRUNCATE option in the CLASS statement. In any case, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *Base SAS Procedures Guide* and the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

You can specify the following option in the CLASS statement after a slash(/):

TRUNCATE
specifies that class levels should be determined using only up to the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases before SAS 9.

FREQ and WEIGHT Statements

FREQ *variable* **;**

WEIGHT *variable* **;**

FREQ and WEIGHT variables are transferred to the output data sets without change.

MODEL Statement

MODEL *dependents=independents* / < *options* > **;**

The MODEL statement names the dependent variables and independent effects. For the syntax of effects, see the section "Specification of Effects" on page 2488 in Chapter 39, "The GLM Procedure."

You can specify the following option in the MODEL statement after a slash (/):

NOINT
requests that the intercept parameter not be included in the model.

Details: GLMMOD Procedure

Displayed Output

For each pass of the data (that is, for each BY group and for each pass required by the pattern of missing values for the dependent variables), the GLMMOD procedure displays the definitions of the columns of the design matrix along with the following:

- the number of the column
- the name of the associated effect
- the values that the classification variables take for this level of the effect

The design matrix itself is also displayed, along with the following:

- the observation number
- the dependent variable values

- the FREQ and WEIGHT values, if any
- the columns of the design matrix

Missing Values

If some variables have missing values for some observations, then PROC GLMMOD handles missing values in the same way as PROC GLM; see the section "Missing Values" on page 2543 in Chapter 39, "The GLM Procedure," for further details.

OUTPARM= Data Set

An output data set containing information regarding the association between model effects and design matrix columns is created whenever you specify the OUTPARM= option in the PROC GLMMOD statement. The OUTPARM= data set contains an observation for each column of the design matrix with the following variables:

- a numeric variable, _COLNUM_, identifying the number of the column of the design matrix corresponding to this observation
- a character variable, EFFNAME, containing the name of the effect that generates the column of the design matrix corresponding to this observation
- the CLASS variables, with the values they have for the column corresponding to this observation, or blanks if they are not involved with the effect associated with this column

If there are BY-group variables or if the pattern of missing values for the dependent variables requires it, the single data set defines several design matrices. In this case, for each of these design matrices, the OUTPARM= data set also contains the following:

- the current values of the BY variables, if you specify a BY statement
- a numeric variable, _YPASS_, containing the current pass of the data, if the pattern of missing values for the dependent variables requires multiple passes

OUTDESIGN= Data Set

An output data set containing the design matrix is created whenever you specify the OUTDESIGN= option in the PROC GLMMOD statement. The OUTDESIGN= data set contains an observation for each observation in the DATA= data set, with the following variables:

- the dependent variables
- the FREQ variable, if any
- the WEIGHT variable, if any
- a variable for each column of the design matrix, with names COL1, COL2, and so forth

If there are BY-group variables or if the pattern of missing values for the dependent variables requires it, the single data set contains several design matrices. In this case, for each of these, the OUTDESIGN= data set also contains the following:

- the current values of the BY variables, if you specify a BY statement
- a numeric variable, _YPASS_, containing the current pass of the data, if the pattern of missing values for the dependent variables requires multiple passes

ODS Table Names

PROC GLMMOD assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 40.1 ODS Tables Produced by PROC GLMMOD

ODS Table Name	Description	Statement
ClassLevels	Table of class levels	CLASS statement
DependentInfo	Simultaneously analyzed dependent variables	default when there are multiple dependent variables
DesignPoints	Design matrix	default
NObs	Number of observations	default
Parameters	Parameters and associated column numbers	default

Examples: GLMMOD Procedure

Example 40.1: A Two-Way Design

The following program uses the GLMMOD procedure to produce the design matrix for a two-way design. The two classification factors have seven and three levels, respectively, so the design matrix contains $1 + 7 + 3 + 21 = 32$ columns in all. Output 40.1.1, Output 40.1.2, and Output 40.1.3 display the output produced by the following statements.

```
   data Plants;
      input Type $ @;
      do Block=1 to 3;
         input StemLength @;
         output;
         end;
      datalines;
   Clarion   32.7 32.3 31.5
   Clinton   32.1 29.7 29.1
   Knox      35.7 35.9 33.1
   O'Neill   36.0 34.2 31.2
   Compost   31.8 28.0 29.2
   Wabash    38.2 37.8 31.9
   Webster   32.5 31.1 29.7
   ;

proc glmmod outparm=Parm outdesign=Design;
   class Type Block;
   model StemLength = Type|Block;
run;

proc print data=Parm;
run;

proc print data=Design;
run;
```

Output 40.1.1 A Two-Way Design

The GLMMOD Procedure

Class Level Information

Class	Levels	Values
Type	7	Clarion Clinton Compost Knox O'Neill Wabash Webster
Block	3	1 2 3

Number of Observations Read	21
Number of Observations Used	21

Parameter Definitions

Column Number	Name of Associated Effect	CLASS Variable Values Type	Block
1	Intercept		
2	Type	Clarion	
3	Type	Clinton	
4	Type	Compost	
5	Type	Knox	
6	Type	O'Neill	
7	Type	Wabash	
8	Type	Webster	
9	Block		1
10	Block		2
11	Block		3
12	Type*Block	Clarion	1
13	Type*Block	Clarion	2
14	Type*Block	Clarion	3
15	Type*Block	Clinton	1
16	Type*Block	Clinton	2
17	Type*Block	Clinton	3
18	Type*Block	Compost	1
19	Type*Block	Compost	2
20	Type*Block	Compost	3
21	Type*Block	Knox	1
22	Type*Block	Knox	2
23	Type*Block	Knox	3
24	Type*Block	O'Neill	1
25	Type*Block	O'Neill	2
26	Type*Block	O'Neill	3
27	Type*Block	Wabash	1
28	Type*Block	Wabash	2
29	Type*Block	Wabash	3
30	Type*Block	Webster	1
31	Type*Block	Webster	2
32	Type*Block	Webster	3

Output 40.1.1 *continued*

Design Points

Observation Number	Stem Length	Column Number 1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	32.7	1	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
2	32.3	1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0
3	31.5	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0
4	32.1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0
5	29.7	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0
6	29.1	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1
7	35.7	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0
8	35.9	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
9	33.1	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0
10	36.0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0
11	34.2	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
12	31.2	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
13	31.8	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0
14	28.0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0
15	29.2	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0
16	38.2	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
17	37.8	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0
18	31.9	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0
19	32.5	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0
20	31.1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
21	29.7	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0

Design Points

Observation Number	Column Number 18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
13	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Output 40.1.2 The OUTPARM= Data Set

Obs	_COLNUM_	EFFNAME	Type	Block
1	1	Intercept		
2	2	Type	Clarion	
3	3	Type	Clinton	
4	4	Type	Compost	
5	5	Type	Knox	
6	6	Type	O'Neill	
7	7	Type	Wabash	
8	8	Type	Webster	
9	9	Block		1
10	10	Block		2
11	11	Block		3
12	12	Type*Block	Clarion	1
13	13	Type*Block	Clarion	2
14	14	Type*Block	Clarion	3
15	15	Type*Block	Clinton	1
16	16	Type*Block	Clinton	2
17	17	Type*Block	Clinton	3
18	18	Type*Block	Compost	1
19	19	Type*Block	Compost	2
20	20	Type*Block	Compost	3
21	21	Type*Block	Knox	1
22	22	Type*Block	Knox	2
23	23	Type*Block	Knox	3
24	24	Type*Block	O'Neill	1
25	25	Type*Block	O'Neill	2
26	26	Type*Block	O'Neill	3
27	27	Type*Block	Wabash	1
28	28	Type*Block	Wabash	2
29	29	Type*Block	Wabash	3
30	30	Type*Block	Webster	1
31	31	Type*Block	Webster	2
32	32	Type*Block	Webster	3

Output 40.1.3 The OUTDESIGN= Data Set

Obs	Stem Length	Col1	Col2	Col3	Col4	Col5	Col6	Col7	Col8	Col9	Col10	Col11	Col12	Col13	Col14	Col15	Col16	Col17	Col18	Col19	Col20	Col21	Col22	Col23	Col24	Col25	Col26	Col27	Col28	Col29	Col30	Col31	Col32
1	32.7	1	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	32.3	1	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	31.5	1	1	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	32.1	1	0	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	29.7	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	29.1	1	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	35.7	1	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
8	35.9	1	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
9	33.1	1	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
10	36.0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
11	34.2	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
12	31.2	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
13	31.8	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	28.0	1	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
15	29.2	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
16	38.2	1	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
17	37.8	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
18	31.9	1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
19	32.5	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
20	31.1	1	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
21	29.7	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Example 40.2: Factorial Screening

Screening experiments are undertaken to select from among the many possible factors that might affect a response the few that actually do, either simply (main effects) or in conjunction with other factors (interactions). One method of selecting significant factors is forward model selection, in which the model is built by successively adding the most statistically significant effects. Forward selection is an option in the REG procedure, but the REG procedure does not allow you to specify interactions directly (as the GLM procedure does, for example). You can use the GLMMOD procedure to create the screening model for a design and then use the REG procedure on the results to perform the screening.

The following statements create the SAS data set Screening, which contains the results of a screening experiment:

```
title 'PROC GLMMOD and PROC REG for Forward Selection Screening';
data Screening;
   input a b c d e y;
   datalines;
-1 -1 -1 -1  1  -6.688
-1 -1 -1  1 -1 -10.664
-1 -1  1 -1 -1  -1.459
-1 -1  1  1  1   2.042
-1  1 -1 -1 -1  -8.561
-1  1 -1  1  1  -7.095
-1  1  1 -1  1   0.553
-1  1  1  1 -1  -2.352
 1 -1 -1 -1 -1  -4.802
 1 -1 -1  1  1   5.705
 1 -1  1 -1  1  14.639
 1 -1  1  1 -1   2.151
 1  1 -1 -1  1   5.884
 1  1 -1  1 -1  -3.317
 1  1  1 -1 -1   4.048
 1  1  1  1  1  15.248
;
run;
```

The data set contains a single dependent variable (y) and five independent factors (a, b, c, d, and e). The design is a half-fraction of the full 2^5 factorial, the precise half-fraction having been chosen to provide uncorrelated estimates of all main effects and two-factor interactions.

The following statements use the GLMMOD procedure to create a design matrix data set containing all the main effects and two-factor interactions for the preceding screening design.

```
ods output DesignPoints = DesignMatrix;
proc glmmod data=Screening;
   model y = a|b|c|d|e@2;
run;
```

Notice that the preceding statements use ODS to create the design matrix data set, instead of the OUTDESIGN= option in the PROC GLMMOD statement. The results are equivalent, but the

columns of the data set produced by ODS have names that are directly related to the names of their corresponding effects.

Finally, the following statements use the REG procedure to perform forward model selection for the screening design. Two MODEL statements are used, one without the selection options (which produces the regression analysis for the full model) and one with the selection options. Output 40.2.1 and Output 40.2.2 show the results of the PROC REG analysis.

```
proc reg data=DesignMatrix;
   model y = a--d_e;
   model y = a--d_e / selection = forward
                      details   = summary
                      slentry   = 0.05;
run;
```

Output 40.2.1 PROC REG Full Model Fit

PROC GLMMOD and PROC REG for Forward Selection Screening

The REG Procedure
Model: MODEL1
Dependent Variable: y

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	15	861.48436	57.43229	.	.
Error	0	0	.		
Corrected Total	15	861.48436			

Root MSE	.	R-Square	1.0000
Dependent Mean	0.33325	Adj R-Sq	.
Coeff Var	.		

Output 40.2.1 *continued*

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	0.33325	.	.	.
a		1	4.61125	.	.	.
b		1	0.21775	.	.	.
a_b	a*b	1	0.30350	.	.	.
c		1	4.02550	.	.	.
a_c	a*c	1	0.05150	.	.	.
b_c	b*c	1	-0.20225	.	.	.
d		1	-0.11850	.	.	.
a_d	a*d	1	0.12075	.	.	.
b_d	b*d	1	0.18850	.	.	.
c_d	c*d	1	0.03200	.	.	.
e		1	3.45275	.	.	.
a_e	a*e	1	1.97175	.	.	.
b_e	b*e	1	-0.35625	.	.	.
c_e	c*e	1	0.30900	.	.	.
d_e	d*e	1	0.30750	.	.	.

Output 40.2.2 PROC REG Screening Results

Summary of Forward Selection

Step	Variable Entered	Label	Number Vars In	Partial R-Square	Model R-Square	C(p)	F Value	Pr > F
1	a		1	0.3949	0.3949	.	9.14	0.0091
2	c		2	0.3010	0.6959	.	12.87	0.0033
3	e		3	0.2214	0.9173	.	32.13	0.0001
4	a_e	a*e	4	0.0722	0.9895	.	75.66	<.0001

The full model has 16 parameters (the intercept + 5 main effects + 10 interactions). These are all estimable, but since there are only 16 observations in the design, there are no degrees of freedom left to estimate error; consequently, there is no way to use the full model to test for the statistical significance of effects. However, the forward selection method chooses only four effects for the model: the main effects of factors a, c, and e, and the interaction between a and e. Using this reduced model enables you to estimate the underlying level of noise, although note that the selection method biases this estimate somewhat.

References

Erdman, L. W. (1946), "Studies to Determine If Antibiosis Occurs among Rhizobia," *Journal of the American Society of Agronomy*, 38, 251–258.

Steel, R. G. D. and Torrie, J. H. (1980), *Principles and Procedures of Statistics*, Second Edition, New York: McGraw-Hill.

Chapter 41

The GLMPOWER Procedure

Contents

Overview: GLMPOWER Procedure **2641**
Getting Started: GLMPOWER Procedure **2643**
Simple Two-Way ANOVA 2643
Incorporating Contrasts, Unbalanced Designs, and Multiple Means Scenarios 2647
Syntax: GLMPOWER Procedure **2649**
PROC GLMPOWER Statement 2650
BY Statement 2651
CLASS Statement 2652
CONTRAST Statement 2652
MODEL Statement 2653
PLOT Statement 2654
POWER Statement 2657
WEIGHT Statement 2660
Details: GLMPOWER Procedure **2660**
Specifying Value Lists in the POWER Statement 2660
Number-Lists 2660
Sample Size Adjustment Options 2661
Error and Information Output 2661
Displayed Output 2662
ODS Table Names 2663
Computational Methods and Formulas 2664
Contrasts in Fixed-Effect Univariate Models 2664
Adjustments for Covariates 2666
Examples: GLMPOWER Procedure **2666**
Example 41.1: One-Way ANOVA 2666
Example 41.2: Two-Way ANOVA with Covariate 2672
References **2680**

Overview: GLMPOWER Procedure

Power and sample size analysis optimizes the resource usage and design of a study, improving chances of conclusive results with maximum efficiency. The GLMPOWER procedure performs prospective power and sample size analysis for linear models, with a variety of goals:

- determining the sample size required to get a significant result with adequate probability (power)
- characterizing the power of a study to detect a meaningful effect
- conducting what-if analyses to assess sensitivity of the power or required sample size to other factors

Here *prospective* indicates that the analysis pertains to planning for a future study. This is in contrast to *retrospective* analysis for a past study, which is not supported by this procedure.

The statistical analyses that are covered include Type III tests and contrasts of fixed effects in univariate linear models, optionally with covariates. The covariates can be continuous or categorical. Tests and contrasts involving random effects are not supported. For power and sample size analyses in a variety of other statistical situations, see Chapter 67, "The POWER Procedure."

Input for PROC GLMPOWER includes the components considered in study planning:

- design (including subject profiles and their allocation weights)
- statistical model
- contrasts of class effects
- significance level (alpha)
- surmised response means for subject profiles (often called "cell means")
- surmised variability
- power
- sample size

In order to identify power or sample size as the result parameter, you designate it by a missing value in the input. The procedure calculates this result value over one or more scenarios of input values for all other components.

You specify the design and the cell means by using an *exemplary data set*, a data set of artificial values constructed to represent the intended sampling design and the surmised response means in the underlying population. You specify the model and contrasts by using MODEL and CONTRAST statements similar to those in the GLM, ANOVA, and MIXED procedures. You specify the remaining parameters with the POWER statement, which is similar to analysis statements in the POWER procedure.

In addition to tabular results, PROC GLMPOWER produces graphs. You can produce the most common types of plots easily with default settings and use a variety of options for more customized graphics. For example, you can control the choice of axis variables, axis ranges, number of plotted points, mapping of graphical features (such as color, line style, symbol, and panel) to analysis parameters, and legend appearance.

The GLMPOWER procedure is one of several tools available in SAS/STAT software for power and sample size analysis. PROC POWER covers a variety of other analyses such as t tests, equivalence

tests, confidence intervals, binomial proportions, multiple regression, one-way ANOVA, survival analysis, logistic regression, and the Wilcoxon rank-sum test. The Power and Sample Size application provides a user interface and implements many of the analyses supported in the procedures. See Chapter 67, "The POWER Procedure," and Chapter 68, "The Power and Sample Size Application," for details.

The following sections of this chapter describe how to use PROC GLMPOWER and discuss the underlying statistical methodology. The section "Getting Started: GLMPOWER Procedure" on page 2643 introduces PROC GLMPOWER with examples of power computation for a two-way analysis of variance. The section "Syntax: GLMPOWER Procedure" on page 2649 describes the syntax of the procedure. The section "Details: GLMPOWER Procedure" on page 2660 summarizes the methods employed by PROC GLMPOWER and provides details on several special topics. The section "Examples: GLMPOWER Procedure" on page 2666 illustrates the use of the GLMPOWER procedure with several applications.

For an overview of methodology and SAS tools for power and sample size analysis, see Chapter 19, "Introduction to Power and Sample Size Analysis." For more discussion and examples for linear models, see Castelloe and O'Brien (2001), O'Brien and Shieh (1992), Muller et al. (1992), and O'Brien and Muller (1993). For additional discussion of general power and sample size concepts, see O'Brien and Castelloe (2007), Castelloe (2000), Muller and Benignus (1992), and Lenth (2001).

Getting Started: GLMPOWER Procedure

Simple Two-Way ANOVA

This example demonstrates how to use PROC GLMPOWER to compute and plot power for each effect test in a two-way analysis of variance (ANOVA).

Suppose you are planning an experiment to study the effect of light exposure at three levels on the growth of two varieties of flowers. The planned data analysis is a two-way ANOVA with flower height (measured at two weeks) as the response and a model consisting of the effects of light exposure, flower variety, and their interaction. You want to calculate the power of each effect test for a balanced design with a total of 60 specimens (10 for each combination of exposure and variety) with $\alpha = 0.05$ for each test.

As a first step, create an *exemplary data set* describing your conjectures about the underlying population means. You believe that the mean flower height for each combination of variety and exposure level (that is, for each design profile, or for each *cell* in the design) roughly follows Table 41.1.

Table 41.1 Mean Flower Height (in cm) by Variety and Exposure

	Exposure		
Variety	1	2	3
1	14	16	21
2	10	15	16

The following statements create a data set named Exemplary containing these cell means.

```
data Exemplary;
   do Variety = 1 to 2;
      do Exposure = 1 to 3;
         input Height @@;
         output;
      end;
   end;
   datalines;
      14 16 21
      10 15 16
   ;
run;
```

You also conjecture that the error standard deviation is about 5 cm.

Use the DATA= option in the PROC GLMPOWER statement to specify Exemplary as the exemplary data set. Identify the classification variables (Variety and Exposure) by using the CLASS statement. Specify the model by using the MODEL statement. Use the POWER statement to specify power as the result parameter and provide values for the other analysis parameters, error standard deviation and total sample size. The following SAS statements perform the power analysis:

```
proc glmpower data=Exemplary;
   class Variety Exposure;
   model Height = Variety | Exposure;
   power
      stddev = 5
      ntotal = 60
      power  = .;
run;
```

The MODEL statement defines the full model including both main effects and the interaction. The POWER= option in the POWER statement identifies power as the result parameter with a missing value (POWER=.). The STDDEV= option specifies an error standard deviation of 5, and the NTOTAL= option specifies a total sample size of 60. The default value for the ALPHA= option sets the significance level to $\alpha = 0.05$.

Figure 41.1 shows the output.

Figure 41.1 Sample Size Analysis for Two-Way ANOVA

The GLMPOWER Procedure

Fixed Scenario Elements

Dependent Variable	Height
Error Standard Deviation	5
Total Sample Size	60
Alpha	0.05
Error Degrees of Freedom	54

Computed Power

Index	Source	Test DF	Power
1	Variety	1	0.718
2	Exposure	2	0.957
3	Variety*Exposure	2	0.191

The power is about 0.72 for the test of the Variety effect. In other words, there is a probability of 0.72 that the test of the Variety effect will produce a significant result (given the assumptions for the means and error standard deviation). The power is 0.96 for the test of the Exposure effect and 0.19 for the interaction test.

Now, suppose you want to account for some of your uncertainty in conjecturing the true error standard deviation by evaluating the power at reasonable low and high values, 4 and 6.5. You also want to plot power for sample sizes between 30 and 90. The following statements perform the analysis:

```
proc glmpower data=Exemplary;
   class Variety Exposure;
   model Height = Variety | Exposure;
   power
      stddev = 4 6.5
      ntotal = 60
      power  = .;
   plot x=n min=30 max=90;
run;
```

The PLOT statement with the X=N option requests a plot with sample size on the X axis. (The result parameter—in this case, power—is always plotted on the other axis.) The MIN= and MAX= options in the PLOT statement specify the sample size range.

Figure 41.2 shows the output, and Figure 41.3 shows the plot.

Figure 41.2 Sample Size Analysis for Two-Way ANOVA with Input Ranges

The GLMPOWER Procedure

Fixed Scenario Elements

Dependent Variable	Height
Total Sample Size	60
Alpha	0.05
Error Degrees of Freedom	54

Computed Power

Index	Source	Std Dev	Test DF	Power
1	Variety	4.0	1	0.887
2	Variety	6.5	1	0.496
3	Exposure	4.0	2	0.996
4	Exposure	6.5	2	0.793
5	Variety*Exposure	4.0	2	0.280
6	Variety*Exposure	6.5	2	0.130

Figure 41.3 Plot of Power versus Sample Size for Two-Way ANOVA with Input Ranges

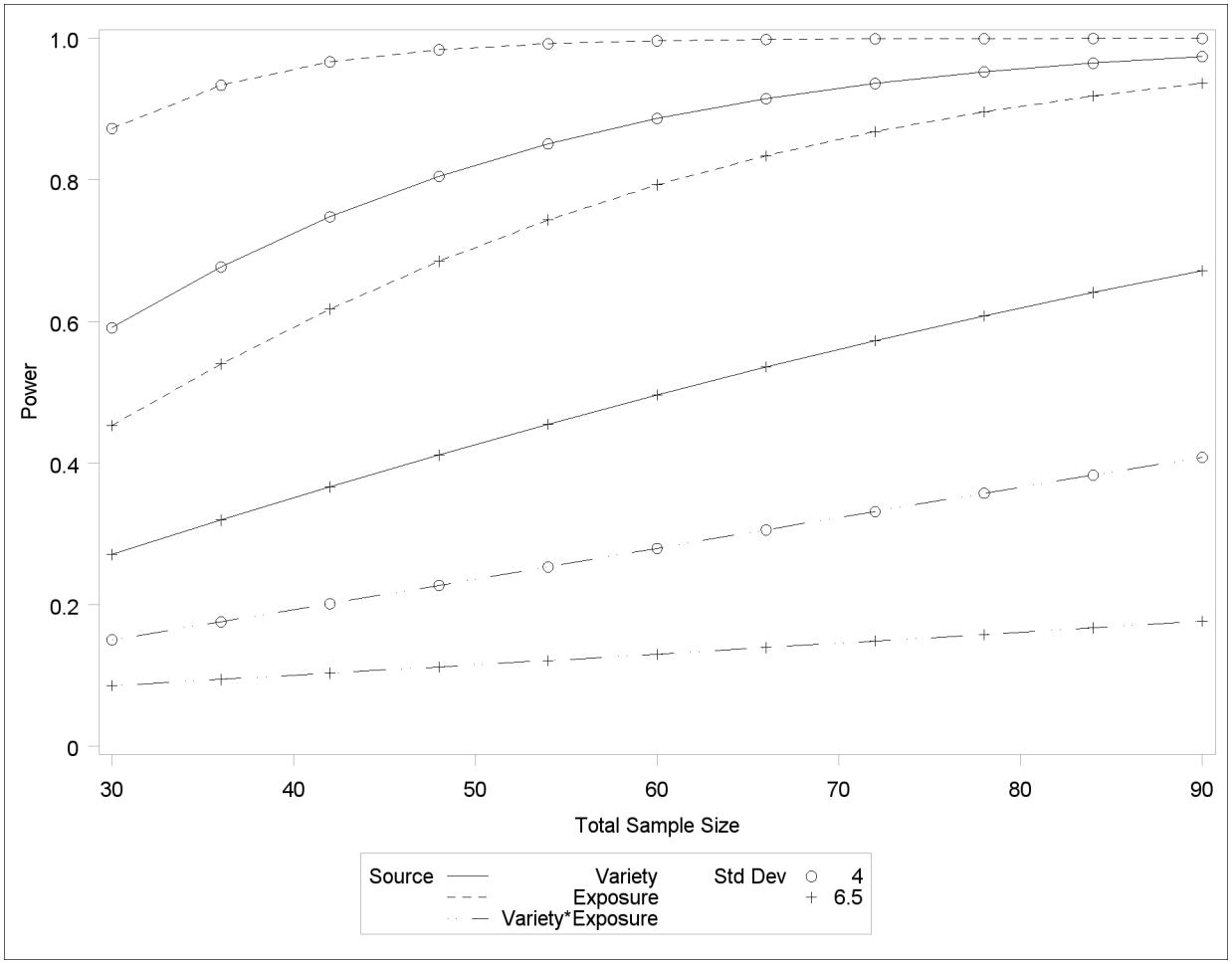


Figure 41.2 reveals that the power ranges from about 0.130 to 0.996 for the different effect tests and scenarios for standard deviation, with a sample size of 60. In Figure 41.3, the line style identifies the effect test, and the plotting symbol identifies the standard deviation. The locations of the plotting symbols identify actual computed powers; the curves are linear interpolations of these points. Note that the computed points in the plot occur at sample size multiples of 6, because there are 6 cells in the design (and by default, sample sizes are rounded to produce integer cell sizes).

Incorporating Contrasts, Unbalanced Designs, and Multiple Means Scenarios

Suppose you want to compute power for the two-way ANOVA described in the section "Simple Two-Way ANOVA" on page 2643, but you want to additionally perform the following tasks:

- try an unbalanced sample size allocation with respect to Exposure, using twice as many samples for levels 2 and 3 as for level 1

- consider an additional, less optimistic scenario for the cell means, shown in Table 41.2
- test a contrast of Exposure comparing levels 1 and 3

Table 41.2 Additional Cell Means Scenario

	Exposure		
Variety	1	2	3
1	15	16	20
2	11	14	15

To specify the unbalanced design and the additional cell means scenario, you can add two new variables to the exemplary data set (Weight for the sample size weights, and HeightNew for the new cell means scenario). Change the name of the original cell means scenario to HeightOrig. The following statements define the exemplary data set:

```
data Exemplary;
   input Variety $ Exposure $ HeightOrig HeightNew Weight;
   datalines;
         1   1   14  15  1
         1   2   16  16  2
         1   3   21  20  2
         2   1   10  11  1
         2   2   15  14  2
         2   3   16  15  2
   ;
run;
```

In PROC GLMPOWER, specify the name of the weight variable by using the WEIGHT statement, and specify the name of the cell means variables as dependent variables in the MODEL statement. Use the CONTRAST statement to specify the contrast as you would in PROC GLM. The following statements perform the sample size analysis.

```
proc glmpower data=Exemplary;
   class Variety Exposure;
   model HeightOrig HeightNew = Variety | Exposure;
   weight Weight;
   contrast 'Exposure=1 vs Exposure=3' Exposure 1 0 -1;
   power
      stddev = 5
      ntotal = 60
      power  = .;
run;
```

Figure 41.4 shows the output.

Figure 41.4 Sample Size Analysis for More Complex Two-Way ANOVA

The GLMPOWER Procedure

Fixed Scenario Elements

Weight Variable	Weight
Error Standard Deviation	5
Total Sample Size	60
Alpha	0.05
Error Degrees of Freedom	54

Computed Power

Index	Dependent	Type	Source	Test DF	Power
1	HeightOrig	Effect	Variety	1	0.672
2	HeightOrig	Effect	Exposure	2	0.911
3	HeightOrig	Effect	Variety*Exposure	2	0.217
4	HeightOrig	Contrast	Exposure=1 vs Exposure=3	1	0.951
5	HeightNew	Effect	Variety	1	0.754
6	HeightNew	Effect	Exposure	2	0.633
7	HeightNew	Effect	Variety*Exposure	2	0.137
8	HeightNew	Contrast	Exposure=1 vs Exposure=3	1	0.705

The power of the contrast of Exposure levels 1 and 3 is about 0.95 for the original cell means scenario (HeightOrig) and only 0.71 for the new one (HeightNew). The power is higher for the test of Variety, but lower for the tests of Exposure and of Variety*Exposure for the new cell means scenario compared to the original one. Note also for the HeightOrig scenario that the power for the unbalanced design (Figure 41.4) compared to the balanced design (Figure 41.1) is slightly lower for the tests of Variety and Exposure, but slightly higher for the test of Variety*Exposure.

Syntax: GLMPOWER Procedure

The following statements are available in PROC GLMPOWER:

PROC GLMPOWER < *options* > ;
BY *variables* ;
CLASS *variables* ;
CONTRAST *'label' effect values* < *. . . effect values* > < */ options* > ;
MODEL *dependents = independents* ;
PLOT < *plot-options* > < */ graph-options* > ;
POWER < *options* > ;
WEIGHT *variable* ;

The PROC GLMPOWER statement, the MODEL statement, and the POWER statement are required. If your model contains classification effects, the classification variables must be listed in

a CLASS statement, and the CLASS statement must appear before the MODEL statement. In addition, CONTRAST and POWER statements must appear after the MODEL statement. PLOT statements must appear after the POWER statement that defines the analysis for the plot.

You can use multiple CONTRAST, POWER, and PLOT statements. Each CONTRAST statement defines a separate contrast. Each POWER statement produces a separate analysis and uses the information contained in the CLASS, MODEL, WEIGHT, and all CONTRAST statements. Each PLOT statement refers to the previous POWER statement and generates a separate graph (or set of graphs).

Table 41.3 summarizes the basic functions of each statement in PROC GLMPOWER. The syntax of each statement in Table 41.3 is described in the following pages.

Table 41.3 Statements in the GLMPOWER Procedure

Statement	Description
PROC GLMPOWER	invokes procedure and specifies exemplary data set
BY	specifies variables to define subgroups for the analysis
CLASS	declares classification variables
CONTRAST	defines linear tests of model parameters
MODEL	defines model and specifies dependent variable(s) used for cell means scenarios
PLOT	displays graphs for preceding POWER statement
POWER	identifies parameter to solve for and provides one or more scenarios for values of other analysis parameters
WEIGHT	specifies variable for allocating sample sizes to different subject profiles

PROC GLMPOWER Statement

PROC GLMPOWER < *options* > **;**

The PROC GLMPOWER statement invokes the GLMPOWER procedure. You can specify the following options.

DATA=*SAS-data-set*
names a SAS data set to be used as the exemplary data set, which is an artificial data set constructed to represent the intended sampling design and the conjectured response means for the underlying population.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of all classification variables (specified in the CLASS statement). This order determines which parameters in the model correspond to each level in the data; so the ORDER= option might be useful when you use CONTRAST statements. Note that the ORDER= option applies to the levels for all classification variables. The exception is the default ORDER=FORMATTED for numeric variables for which you have supplied no explicit format. In this case, the levels are ordered by their internal values.

The following table shows how PROC GLMPOWER interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent. For more information about sorting order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

PLOTONLY

specifies that only graphical results from the PLOT statement be produced.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC GLMPOWER to obtain separate analyses for observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the exemplary data set to be sorted in order of the BY variables.

If your exemplary data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the option NOTSORTED or DESCENDING in the BY statement for the GLMPOWER procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.

- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

Because sorting the data changes the order in which PROC GLMPOWER reads observations, the sorting order for the levels of the classification variables might be affected if you have also specified ORDER=DATA in the PROC GLMPOWER statement. This, in turn, affects specifications in CONTRAST statements.

For more information about the BY statement, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

The CLASS statement names the classification variables to be used in the analysis. Classification variables can be either character or numeric.

CONTRAST Statement

CONTRAST *'label' effect values < . . . effect values > < / options >* **;**

The CONTRAST statement enables you to define custom hypothesis tests by specifying an $\mathbf{L}$ vector or matrix for testing the hypothesis $\mathbf{L}\boldsymbol{\beta} = 0$. Thus, to use this feature you must be familiar with the details of the model parameterization used in PROC GLM. For more information, see the section "Parameterization of PROC GLM Models" on page 2491 of Chapter 39, "The GLM Procedure." All of the elements of the $\mathbf{L}$ vector can be given, or if only certain portions of the $\mathbf{L}$ vector are given, the remaining elements are constructed by PROC GLMPOWER from the context (in a manner similar to rule 4 discussed in the section "Construction of Least Squares Means" on page 2528 of Chapter 39, "The GLM Procedure").

There is no limit to the number of CONTRAST statements you can specify. Each sample size analysis includes tests for all CONTRAST statements.

In the CONTRAST statement,

label identifies the contrast on the output. A label is required for every contrast specified. Labels must be enclosed in quotes.

effect identifies an effect that appears in the MODEL statement, or the INTERCEPT effect. You do not need to include all effects that are in the MODEL statement.

values are constants that are elements of the $\mathbf{L}$ vector associated with the effect.

You can specify the following option in the CONTRAST statement after a slash (/):

SINGULAR=*number*

tunes the estimability checking. If ABS($\mathbf{L} - \mathbf{LH}$) $> C \times$ *number* for any row in the contrast, then $\mathbf{L}$ is declared nonestimable. $\mathbf{H}$ is the $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$ matrix, and C is ABS($\mathbf{L}$) except for rows where $\mathbf{L}$ is zero, and then it is 1. The default value for the SINGULAR= option is 10^{-4}. Values for the SINGULAR= option must be between 0 and 1.

The CONTRAST statement enables you to perform custom hypothesis tests. If the hypothesis is estimable, then the sum of squares due to it, SS(H_0: $\mathbf{L}\boldsymbol{\beta} = 0$), is computed as

$$(\mathbf{Lb})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{Lb})$$

where $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{y}$ is the estimated solution vector.

The degrees of freedom associated with the hypothesis are equal to the row rank of $\mathbf{L}$. The sum of squares computed in this situation is equivalent to the sum of squares computed using an $\mathbf{L}$ matrix with any row deleted that is a linear combination of previous rows.

Multiple-degrees-of-freedom hypotheses can be specified by separating the rows of the $\mathbf{L}$ matrix with commas.

MODEL Statement

MODEL *dependents = independents* **;**

The MODEL statement serves two basic purposes:

- The *dependents* specify scenarios for the cell means.
- The *independents* specify the independent effects.

The *independents* can involve classification variables, continuous variables, or both. You can include main effects and interactions by using the effects notation of PROC GLM; see the section "Specification of Effects" on page 2488 in Chapter 39, "The GLM Procedure" for further details. For any model effect involving classification variables (interactions as well as main effects), the number of levels cannot exceed 32,767. If no independent effects are specified, only an intercept term is fit. The MODEL statement must appear before the POWER statement if the EFFECTS option is used in the POWER statement.

You can account for covariates in the model by using the NCOVARIATES= option and either the CORRXY= or PROPVARREDUCTION= option in the POWER statement.

Each dependent variable refers to a set of surmised cell means in the exemplary data set (named by the DATA= option in the PROC GLMPOWER statement). These cell means are response means for all of the subject profiles. Multiple dependent variables correspond to multiple scenarios for these cell means. All models are univariate; the GLMPOWER procedure currently does not support multivariate analyses.

The MODEL statement is required. You can specify only one MODEL statement.

PLOT Statement

PLOT < *plot-options* > < / *graph-options* > **;**

The PLOT statement produces a graph or set of graphs for the sample size analysis defined by the previous POWER statement. The *plot-options* define the plot characteristics, and the *graph-options* are like those in SAS/GRAPH software.

Options

You can specify the following *plot-options* in the PLOT statement.

INTERPOL=JOIN | NONE
specifies the type of curve to draw through the computed points. The INTERPOL=JOIN option connects computed points with straight lines. The INTERPOL=NONE option leaves computed points unconnected.

KEY=BYCURVE < (*bycurve-options*) >
KEY=BYFEATURE < (*byfeature-options*) >
KEY=ONCURVES
specifies the style of key (or "legend") for the plot. The default is KEY=BYFEATURE, which specifies a key with a column of entries for each plot feature (line style, color, and/or symbol). Each entry shows the mapping between a value of the feature and the value(s) of the analysis parameter(s) linked to that feature. The KEY=BYCURVE option specifies a key with each row identifying a distinct curve in the plot. The KEY=ONCURVES option places a curve-specific label adjacent to each curve.

You can specify the following *byfeature-options* in parentheses after the KEY=BYCURVE option.

NUMBERS=OFF | ON
specifies how the key should identify curves. If NUMBERS=OFF, then the key includes symbol, color, and line style samples to identify the curves. If NUMBERS=ON, then the key includes numbers matching numeric labels placed adjacent to the curves. The default is NUMBERS=ON.

POS=BOTTOM | INSET
specifies the position of the key. The POS=BOTTOM option places the key below the X axis. The POS=INSET option places the key inside the plotting region and attempts to choose the least crowded corner. The default is POS=BOTTOM.

You can specify the following *byfeature-options* in parentheses after KEY=BYFEATURE option.

POS=BOTTOM | INSET
specifies the position of the key. The POS=BOTTOM option places the key below the X axis. The POS=INSET option places the key inside the plotting region and attempts to choose the least crowded corner. The default is POS=BOTTOM.

MARKERS=ANALYSIS | COMPUTED | NICE | NONE
specifies the locations for plotting symbols.

The MARKERS=ANALYSIS option places plotting symbols at locations corresponding to the values of the relevant input parameter from the POWER statement preceding the PLOT statement.

The MARKERS=COMPUTED option (the default) places plotting symbols at the locations of actual computed points from the sample size analysis.

The MARKERS=NICE option places plotting symbols at tick mark locations (corresponding to the argument axis).

The MARKERS=NONE option disables plotting symbols.

MAX=*number* | **DATAMAX**
specifies the maximum of the range of values for the parameter associated with the "argument" axis (the axis that is *not* representing the parameter being solved for). The default is DATAMAX, which specifies the maximum value that occurs for this parameter in the POWER statement that precedes the PLOT statement.

MIN=*number* | **DATAMIN**
specifies the minimum of the range of values for the parameter associated with the "argument" axis (the axis that is *not* representing the parameter being solved for). The default is DATAMIN, which specifies the minimum value that occurs for this parameter in the POWER statement that precedes the PLOT statement.

NPOINTS=*number*
NPTS=*number*
specifies the number of values for the parameter associated with the "argument" axis (the axis that is *not* representing the parameter being solved for). You cannot use the NPOINTS= and STEP= options simultaneously. The default value for typical situations is 20.

STEP=*number*
specifies the increment between values of the parameter associated with the "argument" axis (the axis that is *not* representing the parameter being solved for). You cannot use the STEP= and NPOINTS= options simultaneously. By default, the NPOINTS= option is used instead of the STEP= option.

VARY (*feature* **<** **BY** *parameter-list* **>** **<** **,** **...** **,** *feature* **<** **BY** *parameter-list* **>** **>** **)**
specifies how plot features should be linked to varying analysis parameters. Available *features* are COLOR, LINESTYLE, PANEL, and SYMBOL. A "panel" refers to a separate plot with a heading identifying the subset of values represented in the plot.

The *parameter-list* is a list of one or more names separated by spaces. Each name must match the name of an analysis option used in the POWER statement preceding the PLOT statement, *or* one of the following keywords: SOURCE (for the tests) and DEPENDENT (for the cell means scenarios). Also, the name must be the *primary* name for the analysis option—that is, the one listed first in the syntax description.

If you omit the < BY *parameter-list* > portion for a feature, then one or more multivalued parameters from the analysis will be automatically selected for you.

X=N | POWER
specifies a plot with the requested type of parameter on the X axis and the parameter being solved for on the Y axis. When X=N, sample size is assigned to the X axis. When X=POWER, power is assigned to the X axis. You cannot use the X= and Y= options simultaneously. The default is X=POWER, unless the result parameter is power, in which case the default is X=N.

XOPTS= (*x-options*)
specifies plot characteristics pertaining to the X axis.

You can specify the following *x-options* in parentheses.

CROSSREF=NO | YES
specifies whether the reference lines defined by the REF= *x-option* should be crossed with a reference line on the Y axis that indicates the solution point on the curve.

REF=*number-list*
specifies locations for reference lines extending from the X axis across the entire plotting region. See the section "Specifying Value Lists in the POWER Statement" on page 2660 for information about specifying the *number-list*.

Y=N | POWER
specifies a plot with the requested type of parameter on the Y axis and the parameter being solved for on the X axis. When Y=N, sample size is assigned to the Y axis. When Y=POWER, power is assigned to the Y axis. You cannot use the Y= and X= options simultaneously. By default, the X= option is used instead of the Y= option.

YOPTS= (*y-options*)
specifies plot characteristics pertaining to the Y axis.

You can specify the following *y-options* in parentheses.

CROSSREF=NO | YES
specifies whether the reference lines defined by the REF= *y-option* should be crossed with a reference line on the X axis that indicates the solution point on the curve.

REF=*number-list*
specifies locations for reference lines extending from the Y axis across the entire plotting region. See the section "Specifying Value Lists in the POWER Statement" on page 2660 for information about specifying the *number-list*.

You can specify the following *graph-options* in the PLOT statement after a slash (/).

DESCRIPTION='*string* '
specifies a descriptive string of up to 40 characters that appears in the "Description" field of the graphics catalog. The description does not appear on the plots. By default, PROC GLMPOWER assigns a description either of the form "*Y* versus *X*" (for a single-panel plot) or of the form "*Y* versus *X* (*S*)," where *Y* is the parameter on the Y axis, *X* is the parameter on the X axis, and *S* is a description of the subset represented on the current panel of a multipanel plot.

NAME='*string* '
specifies a name of up to eight characters for the catalog entry for the plot. The default name is PLOT*n*, where *n* is the number of the plot statement within the current invocation of PROC GLMPOWER. If the name duplicates the name of an existing entry, SAS/GRAPH software adds a number to the duplicate name to create a unique entry—for example, PLOT11 and PLOT12 for the second and third panels of a multipanel plot generated in the first PLOT statement in an invocation of PROC GLMPOWER.

POWER Statement

POWER < *options* > **;**

The POWER statement performs power and sample size analyses for the Type III test of each effect in the model defined by the MODEL statements and for the contrasts defined by all CONTRAST statements. The MODEL statement must appear before the POWER statement if the EFFECTS option is used in the POWER statement.

Summary of Options

Table 41.4 summarizes categories of options available in the POWER statement.

Table 41.4 Summary of Options in the POWER Statement

Task	Options
Specify effects	EFFECTS=
Specify significance level	ALPHA=
Specify covariates	CORRXY= NCOVARIATES= PROPVARREDUCTION=
Specify error standard deviation	STDDEV=
Specify sample size	NTOTAL=
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Control ordering in output	OUTPUTORDER= DEPENDENT

Table 41.5 summarizes the valid result parameters.

Table 41.5 Summary of Result Parameters in the POWER Statement

Solve for	Syntax
Power	POWER = .
Sample size	NTOTAL = .

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of each test. The default is 0.05, corresponding to the usual $0.05 \times 100\% = 5\%$ level of significance. Note that this is a test-wise significance level with the same value for all tests, not incorporating any corrections for multiple testing. See the section "Specifying Value Lists in the POWER Statement" on page 2660 for information about specifying the *number-list*.

CORRXY=*number-list*

specifies the multiple correlation (ρ) between all covariates and the response. The error standard deviation given by the STDDEV= option is consequently reduced by multiplying it by a factor of $(1 - \rho^2)^{\frac{1}{2}}$, provided that the number of covariates (as determined by the NCOVARIATES= option) is greater than zero. You cannot use the CORRXY= and the PROPVARREDUCTION= options simultaneously. See the section "Specifying Value Lists in the POWER Statement" on page 2660 for information about specifying the *number-list*.

DEPENDENT

specifies the location of the Dependent column in the output when the OUTPUTORDER=REVERSE option or OUTPUTORDER=SYNTAX option is used, according to its relative position in the POWER statement.

EFFECTS < = < (effect . . . effect) > >

specifies the model effects to include in the power analysis. By default, or if the EFFECTS keyword is specified without the equal sign (=), all model effects are included. Specify EFFECTS=() to exclude all model effect tests from the power analysis. You can include main effects and interactions by using the effects notation of PROC GLM; see the section "Specification of Effects" on page 2488 in Chapter 39, "The GLM Procedure" for further details. The MODEL statement must appear before the POWER statement if the EFFECTS option is used.

NCOVARIATES=*number-list*

NCOVARIATE=*number-list*

NCOVS=*number-list*

NCOV=*number-list*

specifies the number of additional degrees of freedom to accommodate covariate effects—both class and continuous—not listed in the MODEL statement. The error degrees of freedom are consequently reduced by the value of the NCOVARIATES= option, and the error standard deviation (whose unadjusted value is provided with the STDDEV= option) is reduced according to the value of the CORRXY= or PROPVARREDUCTION= option. See the section "Specifying Value Lists in the POWER Statement" on page 2660 for information about specifying the *number-list*.

NFRACTIONAL

NFRAC

enables fractional input and output for sample sizes. See the section "Sample Size Adjustment Options" on page 2661 for information about the ramifications of the presence (and absence) of the NFRACTIONAL option.

NTOTAL=*number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). Values for the sample size must be no smaller than the model degrees of freedom (counting the covariates). See the section "Specifying Value Lists in the POWER Statement" on page 2660 for information about specifying the *number-list*.

OUTPUTORDER=INTERNAL | REVERSE | SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) arranges the parameters in the output according to the following order of their corresponding options:

- DEPENDENT
- EFFECTS=
- weight variable (from the WEIGHT statement)
- ALPHA=
- NCOVARIATES=
- CORRXY=
- PROPVARREDUCTION=
- STDDEV=
- NTOTAL=
- POWER=

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order in which their corresponding options are specified in the POWER statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order in which their corresponding options are specified in the POWER statement.

POWER=*number-list*

specifies the desired power of each test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability (for example, 0.9) rather than a percentage. Note that this is a test-wise power with the same value for all tests, without any correction for multiple testing. See the section "Specifying Value Lists in the POWER Statement" on page 2660 for information about specifying the *number-list*.

PROPVARREDUCTION=*number-list*

PVRED=*number-list*

specifies the proportional reduction (r) in total R^2 incurred by the covariates—in other words, the amount of additional variation explained by the covariates. The error standard deviation given by the STDDEV= option is consequently reduced by multiplying it by a factor of $(1-r)^{\frac{1}{2}}$, provided that the number of covariates (as determined by the NCOVARIATES= option) is greater than zero. You cannot use the PROPVARREDUCTION= and the CORRXY= options simultaneously. See the section "Specifying Value Lists in the POWER Statement" on page 2660 for information about specifying the *number-list*.

STDDEV=*number-list*

specifies the error standard deviation, or root MSE. If covariates are specified using the

NCOVARIATES= option, then the STDDEV= option denotes the error standard deviation before accounting for these covariates. See the section "Specifying Value Lists in the POWER Statement" on page 2660 for information about specifying the *number-list*.

Restrictions on Option Combinations

For the relationship between covariates and response, specify either the multiple correlation (by using the CORRXY= option) or the proportional reduction in total R^2 (by using the PROPVARREDUCTION= option).

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement names a variable that provides a profile weight ("cell weight") for each observation in the exemplary data set specified by the DATA= option in the PROC GLMPOWER statement.

If the WEIGHT statement is not used, then a balanced design is assumed with default cell weights of 1.

Details: GLMPOWER Procedure

Specifying Value Lists in the POWER Statement

To specify one or more scenarios for an analysis parameter (or set of parameters) in the POWER statement, you provide a list of values for the option that corresponds to the parameter(s). To identify the parameter you want to solve for, you place a missing value in the appropriate list.

Scenarios for scalar-valued parameters, such as power, are represented by a *number-list*.

Number-Lists

A *number-list* can be one of two things: a series of one or more numbers expressed in the form of one or more DOLISTs, or a missing value indicator (.).

The DOLIST format is the same as in the DATA step. For example, you can specify four scenarios (30, 50, 70, and 100) for a total sample size in either of the following ways:

```
NTOTAL = 30 50 70 100
NTOTAL = 30 to 70 by 20 100
```

A missing value identifies a parameter as the result parameter; it is valid only with options representing parameters you can solve for in a given analysis. For example, you can request a solution for NTOTAL:

```
NTOTAL = .
```

Sample Size Adjustment Options

By default, PROC GLMPOWER rounds sample sizes conservatively (down in the input, up in the output) so that all total sizes *and* sample sizes for individual design profiles are integers. This is generally considered conservative because it selects the closest realistic design providing *at most* the power of the (possibly fractional) input or mathematically optimized design. In addition, all design profile sizes are adjusted to be multiples of their corresponding weights. If a design profile is present more than once in the exemplary data set, then the weights for that design profile are summed. For example, if a particular design profile is present twice in the exemplary data set with weight values 2 and 6, then all sample sizes for this design profile become multiples of $2 + 6 = 8$.

With the NFRACTIONAL option, sample size input is not rounded, and sample size output is reported in two versions, a raw "fractional" version and a "ceiling" version rounded up to the nearest integer.

Whenever an input sample size is adjusted, both the original ("nominal") and adjusted ("actual") sample sizes are reported. Whenever computed output sample sizes are adjusted, both the original input ("nominal") power and the achieved ("actual") power at the adjusted sample size are reported.

Error and Information Output

The Error column in the main output table explains reasons for missing results and flags numerical results that are bounds rather than exact answers.

The Info column provides further information about Error entries, warnings about any boundary conditions detected, and notes about any adjustments to input. Note that the Info column is hidden by default in the main output. You can view it by using the ODS OUTPUT statement to save the output as a data set and the PRINT procedure. For example, the following SAS statements print both the Error and Info columns for a power computation in a one-way ANOVA:

```
data MyExemp;
   input A $ Y1 Y2;
   datalines;
         1   10 11
         2   12 11
         3   15 11
   ;
run;
```

```
proc glmpower data=MyExemp;
   class A;
   model Y1 Y2 = A;
   power
      stddev = 2
      ntotal = 3 10
      power  = .;
   ods output output=Power;
run;

proc print noobs data=Power;
   var NominalNTotal NTotal Dependent Power Error Info;
run;
```

The output is shown in Figure 41.5.

Figure 41.5 Error and Information Columns

Nominal NTotal	NTotal	Dependent	Power	Error	Info
3	3	Y1	.	Invalid input	Error DF=0
10	9	Y1	0.557		Input N adjusted
3	3	Y2	.	Invalid input	Error DF=0 / No effect
10	9	Y2	0.050		Input N adjusted / No effect

The sample size of 3 specified with the NTOTAL= option causes an "Invalid input" message in the Error column and an "Error DF=0" message in the Info column, because a sample size of 3 is so small that there are no degrees of freedom left for the error term. The sample size of 10 causes an "Input N adjusted" message in the Info column, because it is rounded down to 9 to produce integer group sizes of 3 per cell. The cell means scenario represented by the dependent variable Y2 causes a "No effect" message to appear in the Info column, because the means in this scenario are all equal.

Displayed Output

If you use the PLOTONLY option in the PROC GLMPOWER statement, the procedure displays only graphical output. Otherwise, the displayed output of the GLMPOWER procedure includes the following:

- the "Fixed Scenario Elements" table, which shows all applicable single-valued analysis parameters, in the following order: the dependent variable representing the cell means, the source of the test, the weight variable, parameters input explicitly, parameters supplied with defaults, and ancillary results

- an output table showing the following when applicable (in order): the index of the scenario, the dependent variable representing the cell means, the type of the test, the source of the test, all multivalued input, ancillary results, the primary computed result, and error descriptions
- plots (if requested)

The exception to these ordering conventions is that the DEPENDENT and EFFECTS options may be used along with the OUTPUTORDER=SYNTAX or OUTPUTORDER=REVERSE option in the POWER statement to specify the relative location of the output for dependent variable and type and source of test.

Ancillary results include the following:

- Actual Power, the achieved power, if it differs from the input (Nominal) power value
- fractional sample size, if the NFRACTIONAL option is used in the POWER statement

If sample size is the result parameter and the NFRACTIONAL option is used in the POWER statement, then both "Fractional" and "Ceiling" sample size results are displayed. Fractional sample sizes correspond to the "Nominal" values of power. Ceiling sample sizes are simply the fractional sample sizes rounded up to the nearest integer; they correspond to "Actual" values of power.

The noncentrality parameter is computed and stored in a hidden column called Noncentrality in the "Output" table.

ODS Table Names

PROC GLMPOWER assigns a name to each table that it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 41.6. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 41.6 ODS Tables Produced by PROC GLMPOWER

ODS Table Name	Description	Statement
FixedElements	factoid with single-valued analysis parameters	default
Output	all input and computed analysis parameters, error messages, and information messages for each scenario	default
PlotContent	data contained in plots, including analysis parameters and indices identifying plot features. (**NOTE:** This table is saved as a data set and not displayed in PROC GLMPOWER output.)	PLOT

The ODS pathnames are created as follows:

- Glmpower.Power< n >.FixedElements
- Glmpower.Power< n >.Output
- Glmpower.Power< n >.PlotContent
- Glmpower.Power< n >.Plot< m >

where

- the Plot< m > objects are the graphs.
- the < n > indexing the POWER statement is used only if there is more than one instance.
- the < n > indexing the plots increases with every panel in every plot statement, resetting to 1 only at new analysis statements.

Computational Methods and Formulas

This section describes the approaches used in PROC GLMPOWER to compute power and sample size.

Contrasts in Fixed-Effect Univariate Models

The univariate linear model has the form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\mathbf{y}$ is the $N \times 1$ vector of responses, $\mathbf{X}$ is the $N \times p$ design matrix, $\boldsymbol{\beta}$ is the $p \times 1$ vector of model parameters corresponding to the columns of $\mathbf{X}$, and $\boldsymbol{\epsilon}$ is an $N \times 1$ vector of errors with

$$\epsilon_1, \ldots, \epsilon_N \sim \mathrm{N}(0, \sigma^2) \quad \text{(i.i.d.)}$$

In PROC GLMPOWER, the model parameters $\boldsymbol{\beta}$ are not specified directly, but rather indirectly as $\mathbf{y}^\star$, which represents either conjectured response means or typical response values for each design profile. The $\mathbf{y}^\star$ values are manifested as the dependent variable in the MODEL statement. The vector $\boldsymbol{\beta}$ is obtained from $\mathbf{y}^\star$ according to the least squares equation,

$$\boldsymbol{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y}^\star$$

Note that, in general, there is not a 1-to-1 mapping between $\mathbf{y}^\star$ and $\boldsymbol{\beta}$. Many different scenarios for $\mathbf{y}^\star$ might lead to the same $\boldsymbol{\beta}$. If you specify $\mathbf{y}^\star$ with the intention of representing cell means, keep in mind that PROC GLMPOWER allows scenarios that are *not* valid cell means according to the model specified in the MODEL statement. For example, if $\mathbf{y}^\star$ exhibits an interaction effect but the corresponding interaction term is left out of the model, then the cell means ($\mathbf{X}\boldsymbol{\beta}$) derived from

$\boldsymbol{\beta}$ differ from $\mathbf{y}^\star$. In particular, the cell means thus derived are the projection of $\mathbf{y}^\star$ onto the model space.

It is convenient in power analysis to parameterize the design matrix $\mathbf{X}$ in three parts, $\{\ddot{\mathbf{X}}, \mathbf{w}, N\}$, defined as follows:

1. The $q \times p$ essence design matrix $\ddot{\mathbf{X}}$ is the collection of unique rows of $\mathbf{X}$. Its rows are sometimes referred to as "design profiles." Here, $q \leq N$ is defined simply as the number of unique rows of $\mathbf{X}$.
2. The $q \times 1$ weight vector $\mathbf{w}$ reveals the relative proportions of design profiles. Row i of $\ddot{\mathbf{X}}$ is to be included in the design w_i times for every w_j times row j is included. The weights are assumed to be standardized (that is, sum up to 1).
3. The total sample size is N. This is the number of rows in $\mathbf{X}$. If you gather $Nw_i = n_i$ copies of the ith row of $\ddot{\mathbf{X}}$, for $i = 1, \ldots, q$, then you end up with $\mathbf{X}$.

It is useful to express the crossproduct matrix $\mathbf{X}'\mathbf{X}$ in terms of these three parts,

$$\mathbf{X}'\mathbf{X} = N\ddot{\mathbf{X}}'\text{diag}(\mathbf{w})\ddot{\mathbf{X}}$$

since this factors out the portion (N) depending on sample size and the portion $(\ddot{\mathbf{X}}'\text{diag}(\mathbf{w})\ddot{\mathbf{X}})$ depending only on the design structure.

A general linear hypothesis for the univariate model has the form

$$\begin{aligned} H_0&: \mathbf{L}\boldsymbol{\beta} = \boldsymbol{\theta}_0 \\ H_A&: \mathbf{L}\boldsymbol{\beta} \neq \boldsymbol{\theta}_0 \end{aligned}$$

where $\mathbf{L}$ is an $r_L \times p$ contrast matrix (assumed to be full rank) and $\boldsymbol{\theta}_0$ is the null value (usually just a vector of zeros). Note that effect tests are just contrasts that use special forms of $\mathbf{L}$. Thus, this scheme covers both effect tests and custom contrasts.

The test statistic is

$$F = \frac{\left(\frac{\text{SS}_\text{H}}{r_L}\right)}{\hat{\sigma}^2}$$

where

$$\begin{aligned} \text{SS}_\text{H} &= \frac{1}{N}\left(\mathbf{L}\hat{\boldsymbol{\beta}} - \boldsymbol{\theta}_0\right)' \left(\mathbf{L}\left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{L}'\right)^{-1} \left(\mathbf{L}\hat{\boldsymbol{\beta}} - \boldsymbol{\theta}_0\right) \\ \hat{\boldsymbol{\beta}} &= (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \\ \hat{\sigma}^2 &= \frac{1}{\text{DF}_\text{E}}\left(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right)' \left(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\right) \end{aligned}$$

where $\text{DF}_\text{E} = N - \text{rank}(\mathbf{X})$. Note that $\text{DF}_\text{E} = N - p$ if $\mathbf{X}$ has full rank.

Under H_0, $F \sim F(r_L, \text{DF}_\text{E})$. Under H_A, F is distributed as $F(r_L, \text{DF}_\text{E}, \lambda)$ with noncentrality

$$\lambda = N\left(\mathbf{L}\boldsymbol{\beta} - \boldsymbol{\theta}_0\right)' \left(\mathbf{L}\left(\ddot{\mathbf{X}}'\text{diag}(\mathbf{w})\ddot{\mathbf{X}}\right)^{-1}\mathbf{L}'\right)^{-1} \left(\mathbf{L}\boldsymbol{\beta} - \boldsymbol{\theta}_0\right)\sigma^{-2}$$

Muller and Peterson (1984) give the exact power of the test as

$$\text{power} = P\left(F(r_L, \text{DF}_\text{E}, \lambda) \geq F_{1-\alpha}(r_L, \text{DF}_\text{E})\right)$$

Sample size is computed by inverting the power equation.

See Muller et al. (1992) and O'Brien and Shieh (1992) for additional discussion.

Adjustments for Covariates

If you specify covariates in the model (whether continuous or categorical), then two adjustments are made in order to compute approximate power in the presence of the covariates. Let n_v denote the number of covariates (counting dummy variables for categorical covariates individually). In other words, n_v is the total degrees of freedom used by the covariates. The adjustments are as follows:

1. The error degrees of freedom decrease by n_v.
2. The error standard deviation σ shrinks by a factor of $(1-\rho^2)^{\frac{1}{2}}$ (if the CORRXY= option to specify the correlation ρ between covariates and response) or $(1-r)^{\frac{1}{2}}$ (if the PROPVARREDUCTION= option is used to specify the proportional reduction in total R^2 incurred by the covariates). Let $\sigma^\star$ represent the updated value of σ.

As a result of these changes, the power is computed as

$$\text{power} = P\left(F(r_L, \text{DF}_\text{E} - n_v, \lambda^\star) \geq F_{1-\alpha}(r_L, N - r_x - n_v)\right)$$

where $\lambda^\star$ is calculated using $\sigma^\star$ rather than σ:

$$\lambda^\star = N\left(\mathbf{L}\boldsymbol{\beta} - \boldsymbol{\theta}_0\right)'\left(\mathbf{L}\left(\ddot{\mathbf{X}}'\text{diag}(\mathbf{w})\ddot{\mathbf{X}}\right)^{-1}\mathbf{L}'\right)^{-1}\left(\mathbf{L}\boldsymbol{\beta} - \boldsymbol{\theta}_0\right)(\sigma^\star)^{-2}$$

Examples: GLMPOWER Procedure

Example 41.1: One-Way ANOVA

This example deals with the same situation as in Example 67.1 in Chapter 67, "The POWER Procedure."

Hocking (1985, p. 109) describes a study of the effectiveness of electrolytes in reducing lactic acid buildup for long-distance runners. You are planning a similar study in which you will allocate five different fluids to runners on a 10-mile course and measure lactic acid buildup immediately after the race. The fluids consist of water and two commercial electrolyte drinks, EZDure and LactoZap, each prepared at two concentrations, low (EZD1 and LZ1) and high (EZD2 and LZ2).

You conjecture that the standard deviation of lactic acid measurements given any particular fluid is about 3.75, and that the expected lactic acid values will correspond roughly to Table 41.7. You are least familiar with the LZ1 drink and hence decide to consider a range of reasonable values for that mean.

Table 41.7 Mean Lactic Acid Buildup by Fluid

Water	EZD1	EZD2	LZ1	LZ2
35.6	33.7	30.2	29 or 28	25.9

You are interested in four different comparisons, shown in Table 41.8 with appropriate contrast coefficients.

Table 41.8 Planned Comparisons

	Contrast Coefficients				
Comparison	**Water**	**EZD1**	**EZD2**	**LZ1**	**LZ2**
Water versus electrolytes	4	-1	-1	-1	-1
EZD versus LZ	0	1	1	-1	-1
EZD1 versus EZD2	0	1	-1	0	0
LZ1 versus LZ2	0	0	0	1	-1

For each of these contrasts you want to determine the sample size required to achieve a power of 0.9 for detecting an effect with magnitude in accord with Table 41.7. You are not yet attempting to choose a single sample size for the study, but rather checking the range of sample sizes needed for individual contrasts. You plan to test each contrast at $\alpha = 0.025$. In the interests of reducing costs, you will provide twice as many runners with water as with any of the electrolytes; that is, you will use a sample size weighting scheme of 2:1:1:1:1.

Before calling PROC GLMPOWER, you need to create the *exemplary data set* to specify means and weights for the design profiles:

```
data Fluids;
   input Fluid $ LacticAcid1 LacticAcid2 CellWgt;
   datalines;
         Water       35.6         35.6         2
         EZD1        33.7         33.7         1
         EZD2        30.2         30.2         1
         LZ1         29           28           1
         LZ2         25.9         25.9         1
   ;
run;
```

The variable LacticAcid1 represents the cell means scenario with the larger LZ1 mean (29), and LacticAcid2 represents the scenario with the smaller LZ1 mean (28). The variable CellWgt contains the sample size allocation weights.

Use the DATA= option in the PROC GLMPOWER statement to specify Fluids as the exemplary data set. The following statements perform the sample size analysis:

```
proc glmpower data=Fluids;
   class Fluid;
   model LacticAcid1 LacticAcid2 = Fluid;
   weight CellWgt;
   contrast "Water vs. others" Fluid  -1 -1 -1 -1 4;
   contrast "EZD vs. LZ"       Fluid   1  1 -1 -1 0;
   contrast "EZD1 vs. EZD2"    Fluid   1 -1  0  0 0;
   contrast "LZ1 vs. LZ2"      Fluid   0  0  1 -1 0;
   power
      stddev = 3.75
      alpha  = 0.025
      ntotal = .
      power  = 0.9;
run;
```

The CLASS statement identifies Fluid as a classification variable. The MODEL statement specifies the model and the two cell means scenarios LacticAcid1 and LacticAcid2. The WEIGHT statement identifies CellWgt as the weight variable. The CONTRAST statement specifies the contrasts. Since PROC GLMPOWER by default processes class levels in order of formatted values, the contrast coefficients correspond to the following order: EZD1, EZD2, LZ1, LZ2, Water. (**NOTE:** You could use the ORDER=DATA option in the PROC GLMPOWER statement to achieve the same ordering as in Table 41.8 instead.) The POWER statement specifies total sample size as the result parameter and provides values for the other analysis parameters (error standard deviation, alpha, and power).

Output 41.1.1 displays the results.

Output 41.1.1 Sample Sizes for One-Way ANOVA Contrasts

The GLMPOWER Procedure

Fixed Scenario Elements

Weight Variable	CellWgt
Alpha	0.025
Error Standard Deviation	3.75
Nominal Power	0.9

Computed N Total

Index	Dependent	Type	Source	Test DF	Error DF	Actual Power	N Total
1	LacticAcid1	Effect	Fluid	4	25	0.958	30
2	LacticAcid1	Contrast	Water vs. others	1	25	0.947	30
3	LacticAcid1	Contrast	EZD vs. LZ	1	55	0.929	60
4	LacticAcid1	Contrast	EZD1 vs. EZD2	1	169	0.901	174
5	LacticAcid1	Contrast	LZ1 vs. LZ2	1	217	0.902	222
6	LacticAcid2	Effect	Fluid	4	25	0.972	30
7	LacticAcid2	Contrast	Water vs. others	1	19	0.901	24
8	LacticAcid2	Contrast	EZD vs. LZ	1	43	0.922	48
9	LacticAcid2	Contrast	EZD1 vs. EZD2	1	169	0.901	174
10	LacticAcid2	Contrast	LZ1 vs. LZ2	1	475	0.902	480

The sample sizes range from 24 for the comparison of water versus electrolytes to 480 for the comparison of LZ1 versus LZ2, both assuming the smaller LZ1 mean. The sample size for the latter comparison is relatively large because the small mean difference of $28 - 25.9 = 2.1$ is hard to detect. PROC GLMPOWER also includes the effect test for Fluid. Note that, in this case, it is equivalent to TEST=OVERALL_F in the ONEWAYANOVA statement of PROC POWER, since there is only one effect in the model.

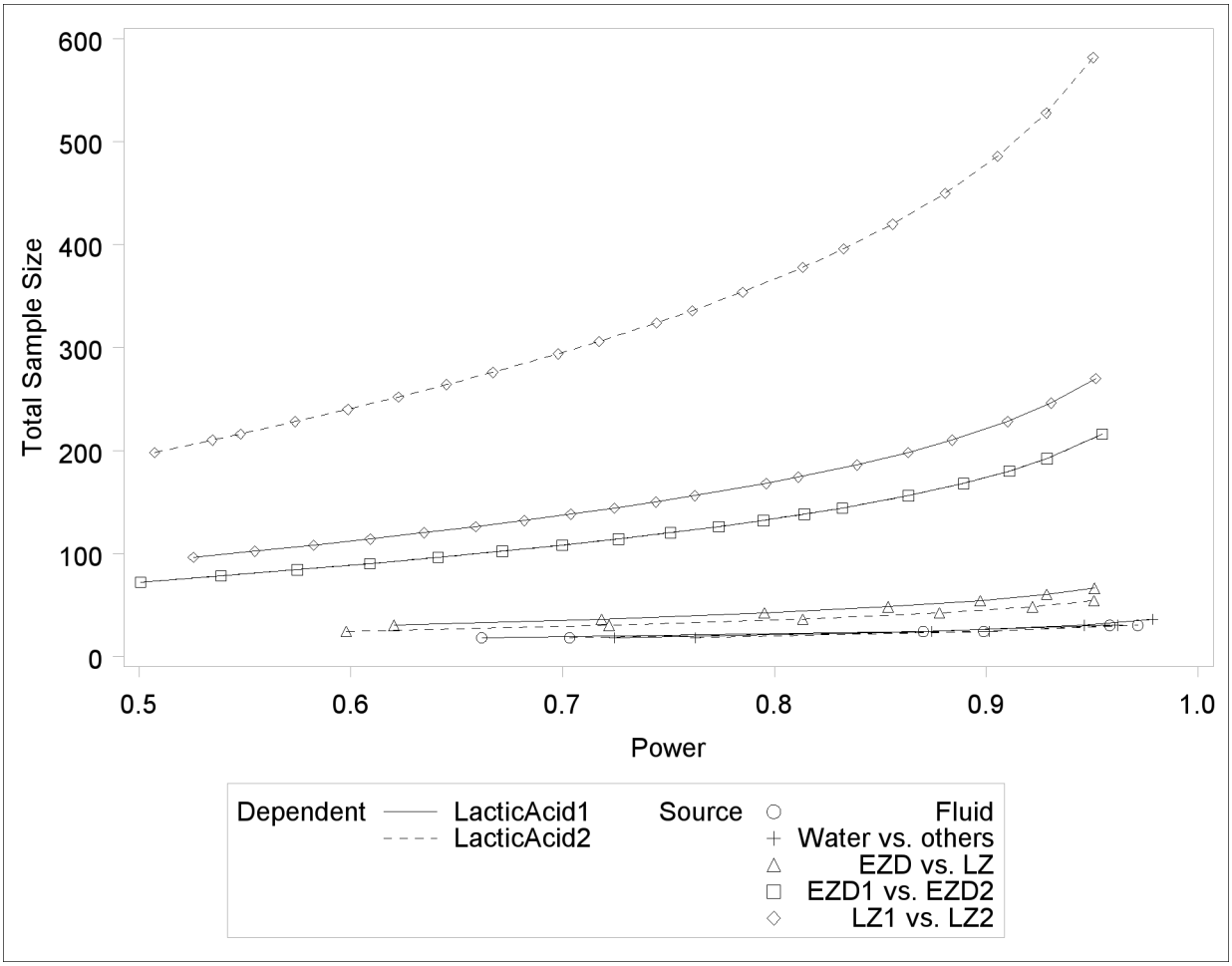
The Nominal Power of 0.9 in the "Fixed Scenario Elements" table in Output 41.1.1 represents the input target power, and the Actual Power column in the "Computed N Total" table is the power at the sample size (N Total) adjusted to achieve the specified sample weighting. Note that all of the sample sizes are rounded up to multiples of 6 to preserve integer group sizes (since the group weights add up to 6). You can use the NFRACTIONAL option in the POWER statement to compute raw fractional sample sizes.

Suppose you want to plot the required sample size for the range of power values from 0.5 to 0.95. First, define the analysis by specifying the same statements as before, but add the PLOTONLY option to the PROC GLMPOWER statement to disable the nongraphical results. Next, specify the PLOT statement with X=POWER to request a plot with power on the X axis. (The result parameter—here sample size—is always plotted on the other axis.) Use the MIN= and MAX= options in the PLOT statement to specify the power range. The following statements produce the plot:

```
proc glmpower data=Fluids plotonly;
   class Fluid;
   model LacticAcid1 LacticAcid2 = Fluid;
   weight CellWgt;
   contrast "Water vs. others" Fluid  -1 -1 -1 -1 4;
   contrast "EZD vs. LZ"       Fluid   1  1 -1 -1 0;
   contrast "EZD1 vs. EZD2"    Fluid   1 -1  0  0 0;
   contrast "LZ1 vs. LZ2"      Fluid   0  0  1 -1 0;
   power
      stddev = 3.75
      alpha  = 0.025
      ntotal = .
      power  = 0.9;
   plot x=power min=.5 max=.95;
run;
```

See Output 41.1.2 for the resulting plot.

Output 41.1.2 Plot of Sample Size versus Power for One-Way ANOVA Contrasts



In Output 41.1.2, the line style identifies the cell means scenario, and the plotting symbol identifies the test. The plotting symbol locations identify actual computed powers; the curves are linear interpolations of these points. The plot shows that the required sample size is highest for the test of LZ1 versus LZ2, which was previously found to require the most resources.

Note that some of the plotted points in Output 41.1.2 are unevenly spaced. This is because the plotted points are the *rounded* sample size results at their corresponding *actual* power levels. The range specified with the MIN= and MAX= values in the PLOT statement corresponds to *nominal* power levels. In some cases, actual power is substantially higher than nominal power. To obtain plots with evenly spaced points (but with *fractional* sample sizes at the computed points), you can use the NFRACTIONAL option in the POWER statement preceding the PLOT statement.

Finally, suppose you want to plot the power for the range of sample sizes you will likely consider for the study (the range of 24 to 480 that achieves 0.9 power for different comparisons). In the POWER statement, identify power as the result (POWER=.), and specify any total sample size value (say, NTOTAL=100). Specify the PLOT statement with X=N to request a plot with sample size on the X axis.

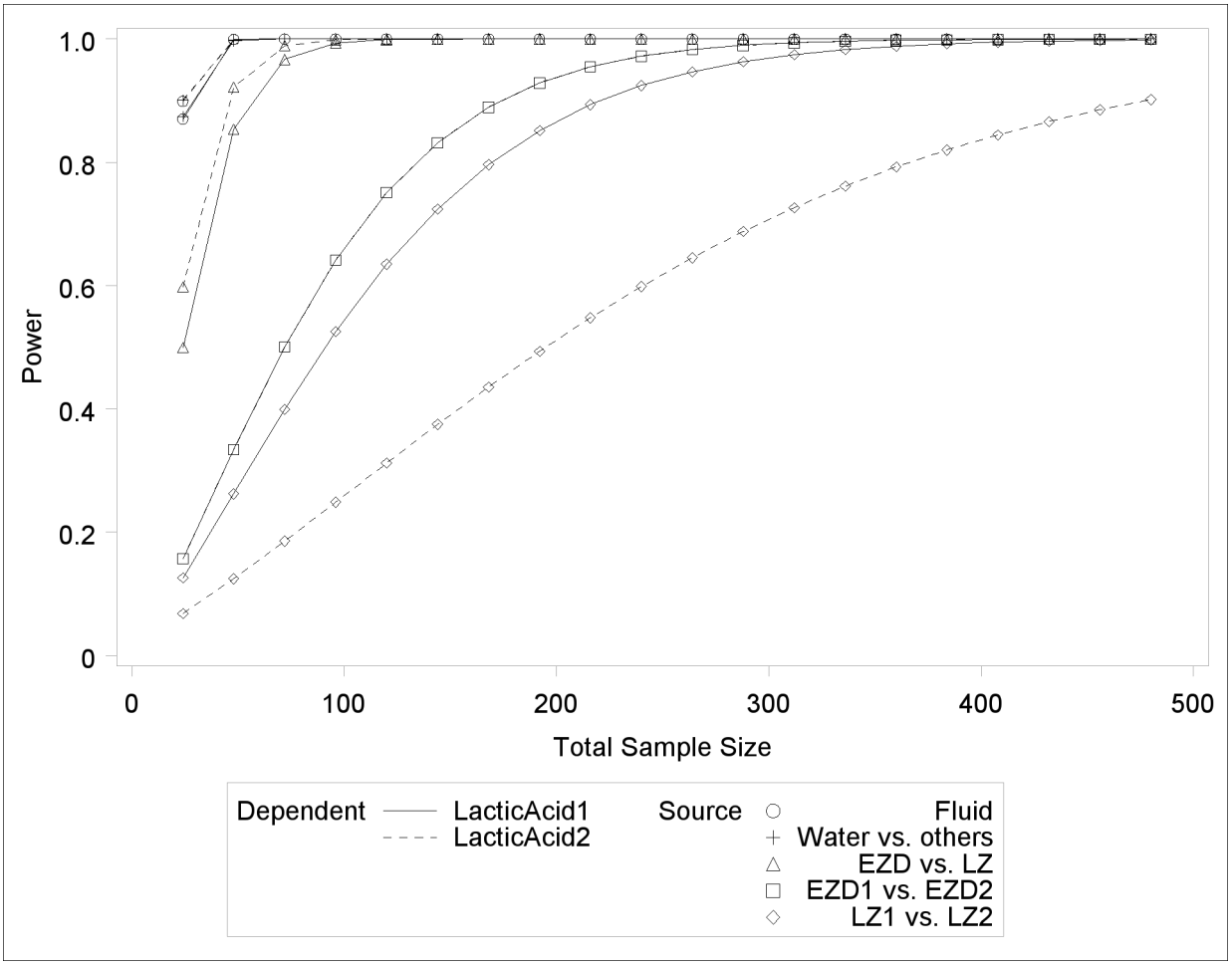
The following statements produce the plot:

```
proc glmpower data=Fluids plotonly;
   class Fluid;
   model LacticAcid1 LacticAcid2 = Fluid;
   weight CellWgt;
   contrast "Water vs. others" Fluid  -1 -1 -1 -1 4;
   contrast "EZD vs. LZ"       Fluid   1  1 -1 -1 0;
   contrast "EZD1 vs. EZD2"    Fluid   1 -1  0  0 0;
   contrast "LZ1 vs. LZ2"      Fluid   0  0  1 -1 0;
   power
      stddev = 3.75
      alpha  = 0.025
      ntotal = 24
      power  = .;
   plot x=n min=24 max=480;
run;
```

Note that the value 100 specified with the NTOTAL=100 option is not used. It is overridden in the plot by the MIN= and MAX= options in the PLOT statement, and the PLOTONLY option in the PROC GLMPOWER statement disables nongraphical results. But the NTOTAL= option (along with a value) is still needed in the POWER statement as a placeholder, to identify the desired parameterization for sample size.

See Output 41.1.3 for the plot.

Output 41.1.3 Plot of Power versus Sample Size for One-Way ANOVA Contrasts



Although Output 41.1.2 and Output 41.1.3 surface essentially the same computations for practical power ranges, they each provide a different quick visual assessment. Output 41.1.2 reveals the range of required sample sizes for powers of interest, and Output 41.1.3 reveals the range of achieved powers for sample sizes of interest.

Example 41.2: Two-Way ANOVA with Covariate

Suppose you can enhance the planned study discussed in Example 41.1 in two ways:

- incorporate results from races at two different altitudes (“high” and “low”)
- measure the body mass index of each runner before the race

This is equivalent to adding a second fixed effect and a continuous covariate to your model.

Since lactic acid buildup is more pronounced at higher altitudes, you will include altitude as a factor in the model along with fluid, extending the one-way ANOVA to a two-way ANOVA. In

doing so, you expect to lower the residual standard deviation from about 3.75 to 3.5 (in addition to generalizing the study results). You assume there is negligible interaction between fluid and altitude and plan to use a main-effects-only model. You conjecture that the mean lactic acid buildup follows Table 41.9.

Table 41.9 Mean Lactic Acid Buildup by Fluid and Altitude

	Fluid				
Altitude	**Water**	**EZD1**	**EZD2**	**LZ1**	**LZ2**
High	36.9	35.0	31.5	30	27.1
Low	34.3	32.4	28.9	27	24.7

By including a measurement of body mass index as a covariate in the study, you hope to further reduce the error variability. The extent of this reduction in variability is commonly expressed in two alternative ways: (1) the correlation between the covariates and the response or (2) the proportional reduction in total R^2 incurred by the covariates. You prefer the former and guess that the correlation between body mass index and lactic acid buildup is between 0.2 and 0.3. You specify these estimates with the NCOVARIATES= and CORRXY= options in the POWER statement. The covariate is not included in the MODEL statement.

You are interested in the same four fluid comparisons as in Example 41.1, shown in Table 41.8, except this time you want to marginalize over the effect of altitude.

For each of these contrasts, you want to determine the sample size required to achieve a power of 0.9 to detect an effect with magnitude according to Table 41.9. You are not yet attempting to choose a single sample size for the study, but rather checking the range of sample sizes needed by individual contrasts. You plan to test each contrast at $\alpha = 0.025$. You will provide twice as many runners with water as with any of the electrolytes, and you predict that you can study approximately two-thirds as many runners at high altitude than at low altitude. The resulting planned sample size weighting scheme is shown in Table 41.10. Since the scheme is only approximate, you use the NFRACTIONAL option in the POWER statement to disable the rounding of sample sizes up to integers satisfying the weights exactly.

Table 41.10 Approximate Sample Size Allocation Weights

	Fluid				
Altitude	**Water**	**EZD1**	**EZD2**	**LZ1**	**LZ2**
High	4	2	2	2	2
Low	6	3	3	3	3

First, you create the exemplary data set to specify means and weights for the design profiles:

```
data Fluids2;
   input Altitude $ Fluid $ LacticAcid CellWgt;
   datalines;
         High       Water      36.9       4
         High       EZD1       35.0       2
         High       EZD2       31.5       2
         High       LZ1        30         2
         High       LZ2        27.1       2
         Low        Water      34.3       6
         Low        EZD1       32.4       3
         Low        EZD2       28.9       3
         Low        LZ1        27         3
         Low        LZ2        24.7       3
   ;
run;
```

The variables Altitude, Fluid, and LacticAcid specify the factors and cell means in Table 41.9. The variable CellWgt contains the sample size allocation weights in Table 41.10.

Use the DATA= option in the PROC GLMPOWER statement to specify Fluids2 as the exemplary data set. The following statements perform the sample size analysis:

```
proc glmpower data=Fluids2;
   class Altitude Fluid;
   model LacticAcid = Altitude Fluid;
   weight CellWgt;
   contrast "Water vs. others" Fluid  -1 -1 -1 -1 4;
   contrast "EZD vs. LZ"       Fluid   1  1 -1 -1 0;
   contrast "EZD1 vs. EZD2"    Fluid   1 -1  0  0 0;
   contrast "LZ1 vs. LZ2"      Fluid   0  0  1 -1 0;
   power
      nfractional
      stddev      = 3.5
      ncovariates = 1
      corrxy      = 0.2 0.3 0
      alpha       = 0.025
      ntotal      = .
      power       = 0.9;
run;
```

The CLASS statement identifies Altitude and Fluid as classification variables. The MODEL statement specifies the model, and the WEIGHT statement identifies CellWgt as the weight variable. The CONTRAST statement specifies the contrasts in Table 41.8. As in Example 41.1, the order of the contrast coefficients corresponds to the formatted class levels (EZD1, EZD2, LZ1, LZ2, Water). The POWER statement specifies total sample size as the result parameter and provides values for the other analysis parameters. The NCOVARIATES= option specifies the single covariate (body mass index), and the CORRXY= option specifies the two scenarios for its correlation with lactic acid buildup (0.2 and 0.3). Output 41.2.1 displays the results.

Output 41.2.1 Sample Sizes for Two-Way ANOVA Contrasts

The GLMPOWER Procedure

Fixed Scenario Elements

Dependent Variable	LacticAcid
Weight Variable	CellWgt
Alpha	0.025
Number of Covariates	1
Std Dev Without Covariate Adjustment	3.5
Nominal Power	0.9

Output 41.2.1 *continued*

Computed Ceiling N Total

Index	Type	Source	Corr XY	Adj Std Dev	Test DF	Error DF	Fractional N Total
1	Effect	Altitude	0.2	3.43	1	84	90.418451
2	Effect	Altitude	0.3	3.34	1	79	85.862649
3	Effect	Altitude	0.0	3.50	1	88	94.063984
4	Effect	Fluid	0.2	3.43	4	16	22.446173
5	Effect	Fluid	0.3	3.34	4	15	21.687544
6	Effect	Fluid	0.0	3.50	4	17	23.055716
7	Contrast	Water vs. others	0.2	3.43	1	15	21.720195
8	Contrast	Water vs. others	0.3	3.34	1	14	20.848805
9	Contrast	Water vs. others	0.0	3.50	1	16	22.422381
10	Contrast	EZD vs. LZ	0.2	3.43	1	35	41.657424
11	Contrast	EZD vs. LZ	0.3	3.34	1	33	39.674037
12	Contrast	EZD vs. LZ	0.0	3.50	1	37	43.246415
13	Contrast	EZD1 vs. EZD2	0.2	3.43	1	139	145.613657
14	Contrast	EZD1 vs. EZD2	0.3	3.34	1	132	138.173983
15	Contrast	EZD1 vs. EZD2	0.0	3.50	1	145	151.565917
16	Contrast	LZ1 vs. LZ2	0.2	3.43	1	268	274.055008
17	Contrast	LZ1 vs. LZ2	0.3	3.34	1	253	259.919126
18	Contrast	LZ1 vs. LZ2	0.0	3.50	1	279	285.363976

Computed Ceiling N Total

Index	Actual Power	Ceiling N Total
1	0.902	91
2	0.901	86
3	0.903	95
4	0.912	23
5	0.908	22
6	0.919	24
7	0.905	22
8	0.903	21
9	0.910	23
10	0.903	42
11	0.903	40
12	0.906	44
13	0.901	146
14	0.902	139
15	0.901	152
16	0.901	275
17	0.900	260
18	0.901	286

The sample sizes in Output 41.2.1 range from 21 for the comparison of water versus electrolytes (assuming a correlation of 0.3 between body mass and lactic acid buildup) to 275 for the comparison of LZ1 versus LZ2 (assuming a correlation of 0.2). PROC GLMPOWER also includes the effect tests for Altitude and Fluid. Note that the required sample sizes for this study are lower than those for the study in Example 41.1.

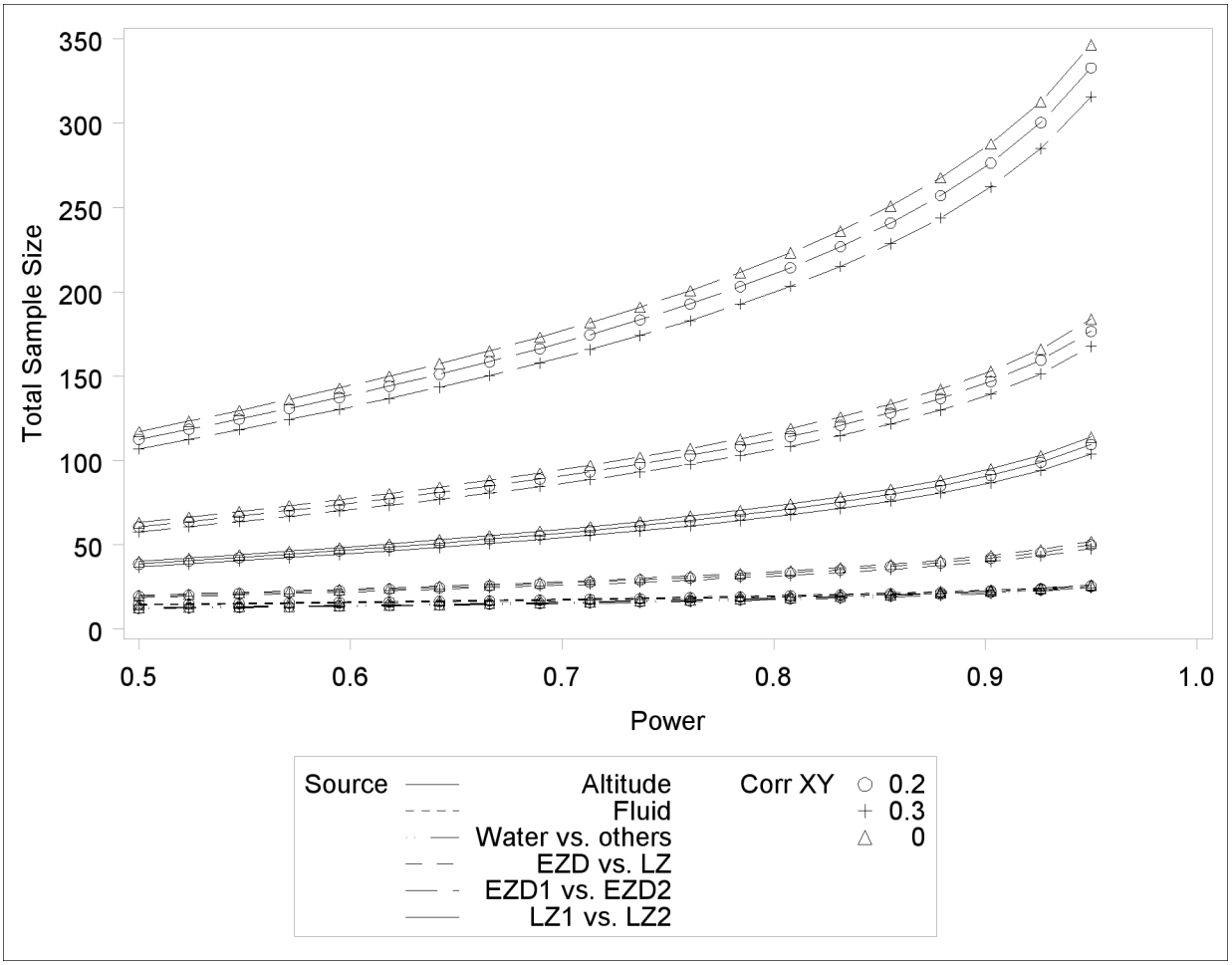
Note that the error standard deviation has been reduced from 3.5 to 3.43 (when correlation is 0.2) or 3.34 (when correlation is 0.3) in the approximation of the effect of the body mass index covariate. The error degrees of freedom has also been automatically adjusted, lowered by 1 (the number of covariates).

Suppose you want to plot the required sample size for the range of power values from 0.5 to 0.95. First, define the analysis by specifying the same statements as before, but add the PLOTONLY option to the PROC GLMPOWER statement to disable the nongraphical results. Next, specify the PLOT statement with X=POWER to request a plot with power on the X axis. Sample size is automatically placed on the Y axis. Use the MIN= and MAX= options in the PLOT statement to specify the power range. The following statements produce the plot:

```
proc glmpower data=Fluids2 plotonly;
   class Altitude Fluid;
   model LacticAcid = Altitude Fluid;
   weight CellWgt;
   contrast "Water vs. others" Fluid  -1 -1 -1 -1 4;
   contrast "EZD vs. LZ"       Fluid   1  1 -1 -1 0;
   contrast "EZD1 vs. EZD2"    Fluid   1 -1  0  0 0;
   contrast "LZ1 vs. LZ2"      Fluid   0  0  1 -1 0;
   power
      nfractional
      stddev      = 3.5
      ncovariates = 1
      corrxy      = 0.2 0.3 0
      alpha       = 0.025
      ntotal      = .
      power       = 0.9;
   plot x=power min=.5 max=.95;
run;
```

See Output 41.2.2 for the resulting plot.

Output 41.2.2 Plot of Sample Size versus Power for Two-Way ANOVA Contrasts



In Output 41.1.2, the line style identifies the test, and the plotting symbol identifies the scenario for the correlation between covariate and response. The plotting symbol locations identify actual computed powers; the curves are linear interpolations of these points. As in Example 41.1, the required sample size is highest for the test of LZ1 versus LZ2.

Finally, suppose you want to plot the power for the range of sample sizes you will likely consider for the study (the range of 21 to 275 that achieves 0.9 power for different comparisons). In the POWER statement, identify power as the result (POWER=.), and specify NTOTAL=21. Specify the PLOT statement with X=N to request a plot with sample size on the X axis.

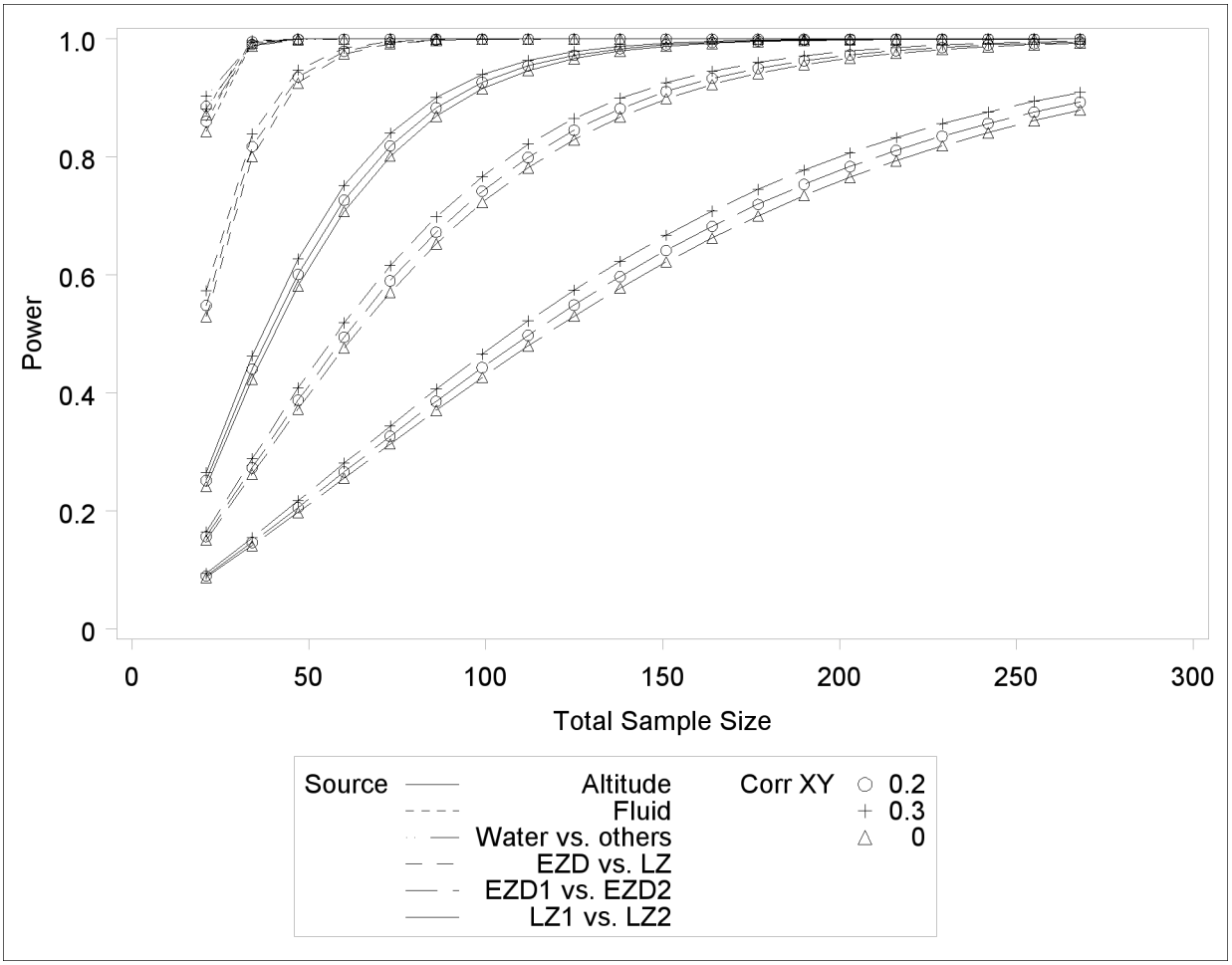
The following statements produce the plot:

```
proc glmpower data=Fluids2 plotonly;
   class Altitude Fluid;
   model LacticAcid = Altitude Fluid;
   weight CellWgt;
   contrast "Water vs. others" Fluid  -1 -1 -1 -1 4;
   contrast "EZD vs. LZ"       Fluid   1  1 -1 -1 0;
   contrast "EZD1 vs. EZD2"    Fluid   1 -1  0  0 0;
   contrast "LZ1 vs. LZ2"      Fluid   0  0  1 -1 0;
   power
      nfractional
      stddev      = 3.5
      ncovariates = 1
      corrxy      = 0.2 0.3 0
      alpha       = 0.025
      ntotal      = 21
      power       = .;
   plot x=n min=21 max=275;
run;
```

The MAX=275 option in the PLOT statement sets the maximum sample size value. The MIN= option automatically defaults to the value of 21 from the NTOTAL= option in the POWER statement.

See Output 41.2.3 for the plot.

Output 41.2.3 Plot of Power versus Sample Size for Two-Way ANOVA Contrasts



Although Output 41.2.2 and Output 41.2.3 surface essentially the same computations for practical power ranges, they each provide a different quick visual assessment. Output 41.2.2 reveals the range of required sample sizes for powers of interest, and Output 41.2.3 reveals the range of powers achieved for sample sizes of interest.

References

Castelloe, J. M. (2000), "Sample Size Computations and Power Analysis with the SAS System," *Proceedings of the Twenty-fifth Annual SAS Users Group International Conference,* Paper 265-25, Cary, NC: SAS Institute Inc.

Castelloe, J. M. and O'Brien, R. G. (2001), "Power and Sample Size Determination for Linear Models," *Proceedings of the Twenty-sixth Annual SAS Users Group International Conference,* Paper 240-26, Cary, NC: SAS Institute Inc.

Hocking, R. R. (1985), *The Analysis of Linear Models*, Monterey, CA: Brooks/Cole.

Lenth, R. V. (2001), "Some Practical Guidelines for Effective Sample Size Determination," *The American Statistician*, 55, 187–193.

Muller, K. E. and Benignus, V.A. (1992), "Increasing Scientific Power with Statistical Power," *Neurotoxicology and Teratology*, 14, 211–219.

Muller, K. E., LaVange, L. M., Ramey, S. L., and Ramey, C. T. (1992), "Power Calculations for General Linear Multivariate Models Including Repeated Measures Applications," *Journal of the American Statistical Association*, 87 (420), 1209–1226.

Muller, K. E. and Peterson, B. L. (1984), "Practical Methods for Computing Power in Testing the Multivariate General Linear Hypothesis," *Computational Statistics & Data Analysis*, 2, 143–158.

O'Brien, R. G. and Castelloe, J. (2007), "Sample-Size Analysis for Traditional Hypothesis Testing: Concepts and Issues," in *Pharmaceutical Statistics Using SAS: A Practical Guide*, ed. A. Dmitrienko, C. Chuang-Stein and R. D'Agostino, Cary, NC: SAS Institute Inc., Chapter 10, 237–271.

O'Brien, R. G. and Muller, K. E. (1993), "Unified Power Analysis for *t*-Tests through Multivariate Hypotheses," in *Applied Analysis of Variance in Behavioral Science*, ed. L. K. Edwards, Chapter 8, 297–344, New York: Marcel Dekker.

O'Brien, R. G. and Shieh, G. (1992), "Pragmatic, Unifying Algorithm Gives Power Probabilities for Common F Tests of the Multivariate General Linear Hypothesis," poster presented at the American Statistical Association Meetings, Boston, Statistical Computing Section. Also, paper in review, downloadable in PDF form from www.bio.ri.ccf.org/UnifyPow.

Chapter 42

The GLMSELECT Procedure

Contents

Overview: GLMSELECT Procedure **2684**
Features 2684
Getting Started: GLMSELECT Procedure **2686**
Syntax: GLMSELECT Procedure **2694**
PROC GLMSELECT Statement 2694
BY Statement 2702
CLASS Statement 2702
EFFECT Statement 2706
FREQ Statement 2706
MODEL Statement 2706
OUTPUT Statement 2714
PARTITION Statement 2715
PERFORMANCE Statement 2716
SCORE Statement 2717
WEIGHT Statement 2718
Details: GLMSELECT Procedure **2719**
Model-Selection Methods 2719
Full Model Fitted (NONE) 2719
Forward Selection (FORWARD) 2719
Backward Elimination (BACKWARD) 2722
Stepwise Selection(STEPWISE) 2723
Least Angle Regression (LAR) 2725
Lasso Selection (LASSO) 2725
Model Selection Issues 2726
Criteria Used in Model Selection Methods 2727
CLASS Variable Parameterization 2730
Macro Variables Containing Selected Models 2734
Building the SSCP Matrix 2737
Parallel BY-Group Computation 2738
Using Validation and Test Data 2739
Cross Validation 2741
Displayed Output 2743
ODS Table Names 2748
ODS Graphics 2749

Examples: GLMSELECT Procedure **2756**
Example 42.1: Modeling Baseball Salaries Using Performance Statistics . . 2756
Example 42.2: Using Validation and Cross Validation 2769
Example 42.3: Scatter Plot Smoothing by Selecting Spline Functions 2786
Example 42.4: Multimember Effects and the Design Matrix 2795
References **2801**

Overview: GLMSELECT Procedure

The GLMSELECT procedure performs effect selection in the framework of general linear models. A variety of model selection methods are available, including the LASSO method of Tibshirani (1996) and the related LAR method of Efron et al. (2004). The procedure offers extensive capabilities for customizing the selection with a wide variety of selection and stopping criteria, from traditional and computationally efficient significance-level-based criteria to more computationally intensive validation-based criteria. The procedure also provides graphical summaries of the selection search.

The GLMSELECT procedure compares most closely to REG and GLM. The REG procedure supports a variety of model-selection methods but does not support a CLASS statement. The GLM procedure supports a CLASS statement but does not include effect selection methods. The GLMSELECT procedure fills this gap. GLMSELECT focuses on the standard independently and identically distributed general linear model for univariate responses and offers great flexibility for and insight into the model selection algorithm. GLMSELECT provides results (displayed tables, output data sets, and macro variables) that make it easy to take the selected model and explore it in more detail in a subsequent procedure such as REG or GLM.

Features

The main features of the GLMSELECT procedure are as follows:

- **Model Specification**
  - supports different parameterizations for classification effects
  - supports any degree of interaction (crossed effects) and nested effects
  - supports hierarchy among effects
  - supports partitioning of data into training, validation, and testing roles
  - supports constructed effects including spline and multimember effects

- **Selection Control**
  - provides multiple effect selection methods
  - enables selection from a very large number of effects (tens of thousands)
  - offers selection of individual levels of classification effects
  - provides effect selection based on a variety of selection criteria
  - provides stopping rules based on a variety of model evaluation criteria
  - provides leave-one-out and k-fold cross validation
- **Display and Output**
  - produces graphical representation of selection process
  - produces output data sets containing predicted values and residuals
  - produces an output data set containing the design matrix
  - produces macro variables containing selected models
  - supports parallel processing of BY groups
  - supports multiple SCORE statements

The GLMSELECT procedure supports the following effect selection methods. These methods are explained in detail in the section "Model-Selection Methods" on page 2719.

FORWARD	Forward selection. This method starts with no effects in the model and adds effects.
BACKWARD	Backward elimination. This method starts with all effects in the model and deletes effects.
STEPWISE	Stepwise regression. This is similar to the FORWARD method except that effects already in the model do not necessarily stay there.
LAR	Least angle regression. This method, like forward selection, starts with no effects in the model and adds effects. The parameter estimates at any step are "shrunk" when compared to the corresponding least squares estimates.
LASSO	This method adds and deletes parameters based on a version of ordinary least squares where the sum of the absolute regression coefficients is constrained.

Hybrid versions of LAR and LASSO are also supported. They use LAR or LASSO to select the model, but then estimate the regression coefficients by ordinary weighted least squares.

The GLMSELECT procedure is intended primarily as a model selection procedure and does not include regression diagnostics or other post-selection facilities such as hypothesis testing, testing of contrasts, and LS-means analyses. The intention is that you use PROC GLMSELECT to select a model or a set of candidate models. Further investigation of these models can be done by using these models in existing regression procedures.

Getting Started: GLMSELECT Procedure

The following data set contains salary and performance information for Major League Baseball players who played at least one game in both the 1986 and 1987 seasons, excluding pitchers. The salaries (*Sports Illustrated,* April 20, 1987) are for the 1987 season and the performance measures are from 1986 (Collier Books, *The 1987 Baseball Encyclopedia Update*).

```
data baseball;
  length name $ 18;
  length team $ 12;
  input name $ 1-18 nAtBat nHits nHome nRuns nRBI nBB
        yrMajor crAtBat crHits crHome crRuns crRbi crBB
        league $ division $ team $ position $ nOuts nAssts
        nError salary;
  label name="Player's Name"
     nAtBat="Times at Bat in 1986"
     nHits="Hits in 1986"
     nHome="Home Runs in 1986"
     nRuns="Runs in 1986"
     nRBI="RBIs in 1986"
     nBB="Walks in 1986"
     yrMajor="Years in the Major Leagues"
     crAtBat="Career times at bat"
     crHits="Career Hits"
     crHome="Career Home Runs"
     crRuns="Career Runs"
     crRbi="Career RBIs"
     crBB="Career Walks"
     league="League at the end of 1986"
     division="Division at the end of 1986"
     team="Team at the end of 1986"
     position="Position(s) in 1986"
     nOuts="Put Outs in 1986"
     nAssts="Assists in 1986"
     nError="Errors in 1986"
     salary="1987 Salary in $ Thousands";
     logSalary = log(Salary);
  datalines;
Allanson, Andy       293    66     1    30    29    14
                       1   293    66     1    30    29    14
                    American East Cleveland C 446 33 20 .
Ashby, Alan          315    81     7    24    38    39
                      14  3449   835    69   321   414   375
                    National West Houston C 632 43 10 475

... more lines ...

Wilson, Willie       631   170     9    77    44    31
                      11  4908  1457    30   775   357   249
                    American West KansasCity CF 408 4 3 1000
;
```

Suppose you want to investigate whether you can model the players' salaries for the 1987 season based on performance measures for the previous season. The aim is to obtain a parsimonious model that does not overfit this particular data, making it useful for prediction. This example shows how you can use PROC GLMSELECT as a starting point for such an analysis. Since the variation of salaries is much greater for the higher salaries, it is appropriate to apply a log transformation to the salaries before doing the model selection.

The following code selects a model with the default settings:

```
ods graphics on;
proc glmselect data=baseball plots=all;
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                    yrMajor crAtBat crHits crHome crRuns crRbi
                    crBB league division nOuts nAssts nError
                  / details=all stats=all;
run;
ods graphics off;
```

PROC GLMSELECT performs effect selection where effects can contain classification variables that you specify in a CLASS statement. The "Class Level Information" table shown in Figure 42.1 lists the levels of the classification variables "division" and "league."

Figure 42.1 Class Level Information

The GLMSELECT Procedure

Class Level Information

Class	Levels	Values
league	2	American National
division	2	East West

When you specify effects that contain classification variables, the number of parameters is usually larger than the number of effects. The "Dimensions" table in Figure 42.2 shows the number of effects and the number of parameters considered.

Figure 42.2 Dimensions

Dimensions

Number of Effects	19
Number of Parameters	21

Figure 42.3 Model Information

The GLMSELECT Procedure

Data Set	WORK.BASEBALL
Dependent Variable	logSalary
Selection Method	Stepwise
Select Criterion	SBC
Stop Criterion	SBC
Effect Hierarchy Enforced	None

You find details of the default search settings in the "Model Information" table shown in Figure 42.3. The default selection method is a variant of the traditional stepwise selection where the decisions about what effects to add or drop at any step and when to terminate the selection are both based on the Schwarz Bayesian information criterion (SBC). The effect in the current model whose removal yields the maximal decrease in the SBC statistic is dropped provided this lowers the SBC value. Once no decrease in the SBC value can be obtained by dropping an effect in the model, the effect whose addition to the model yields the lowest SBC statistic is added and the whole process is repeated. The method terminates when dropping or adding any effect increases the SBC statistic.

Figure 42.4 Candidates for Entry at Step Two

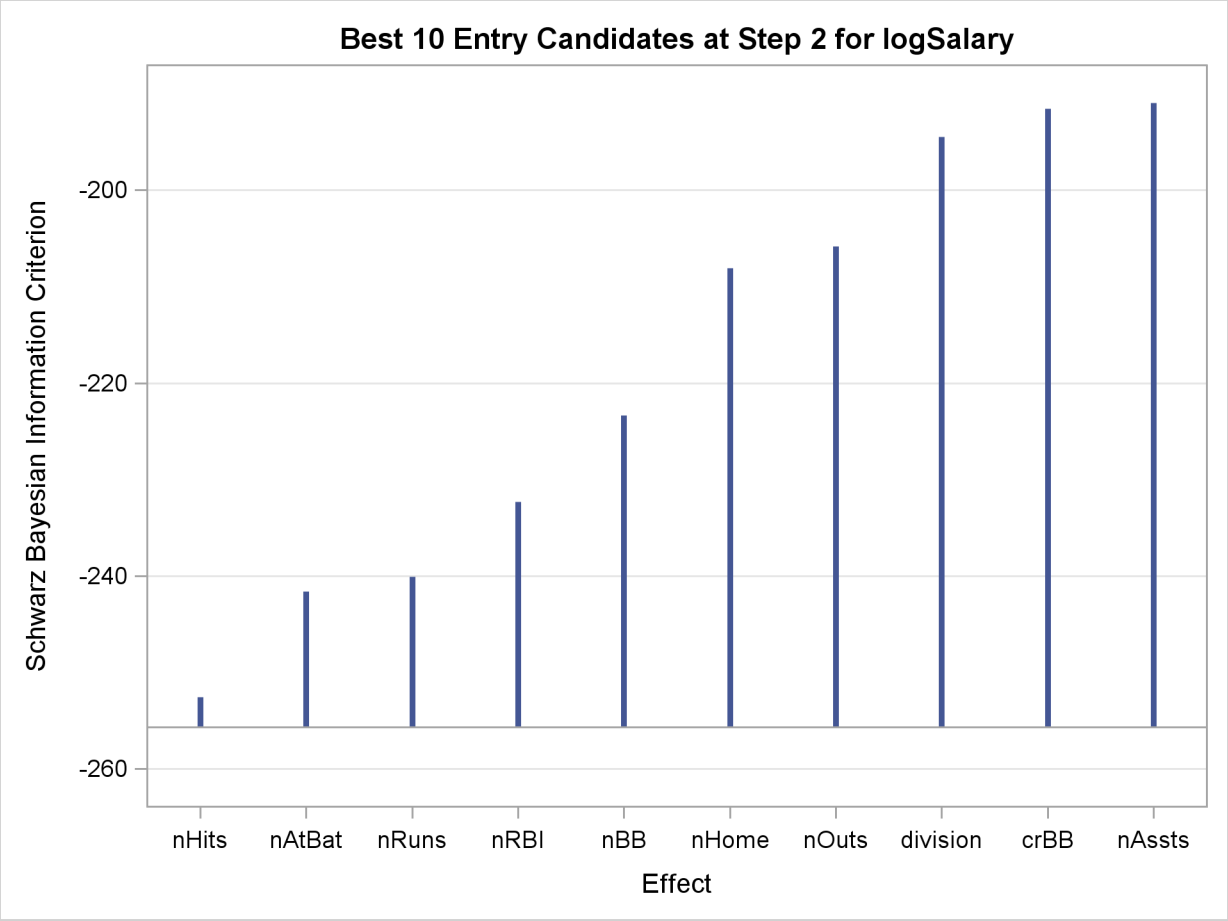
Best 10 Entry Candidates

Rank	Effect	SBC
1	nHits	-252.5794
2	nAtBat	-241.5789
3	nRuns	-240.1010
4	nRBI	-232.2880
5	nBB	-223.3741
6	nHome	-208.0565
7	nOuts	-205.8107
8	division	-194.4688
9	crBB	-191.5141
10	nAssts	-190.9425

The DETAILS=ALL option requests details of each step of the selection process. The "Best 10 Entry Candidates" table at each step shows the candidates for inclusion or removal at that step ranked from best to worst in terms of the selection criterion, which in this example is the SBC statistic. By default only the 10 best candidates are shown. Figure 42.4 shows the candidate table at step two.

To help in the interpretation of the selection process, you can use graphics supported by PROC GLMSELECT. You enable these graphical displays by specifying the ODS GRAPHICS statement. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." With ODS Graphics enabled, the PLOTS=ALL option together with the DETAILS=STEPS option in the MODEL statement produces a needle plot view of the "Candidates" tables. The plot corresponding to the "Candidates" table at step two is shown in Figure 42.5. You can see that adding the effect "nHits" yields the smallest SBC value, and so this effect is added at step two.

Figure 42.5 Needle Plot of Entry Candidates at Step Two



The "Stepwise Selection Summary" table in Figure 42.6 shows the effect that was added or dropped at each step of the selection process together with fit statistics for the model at each step. The STATS=ALL option in the MODEL statement requests that all the available fit statistics are displayed. See the section "Criteria Used in Model Selection Methods" on page 2727 for descriptions and formulas. The criterion panel in Figure 42.7 provides a graphical view of the progression of these fit criteria as the selection process evolves. Note that none of these criteria has a local optimum before step five.

Figure 42.6 Selection Summary Table

The GLMSELECT Procedure

Stepwise Selection Summary

Step	Effect Entered	Effect Removed	Number Effects In	Number Parms In	Model R-Square	Adjusted R-Square
0	Intercept		1	1	0.0000	0.0000
1	crRuns		2	2	0.4187	0.4165
2	nHits		3	3	0.5440	0.5405
3	yrMajor		4	4	0.5705	0.5655
4		crRuns	3	3	0.5614	0.5581
5	nBB		4	4	0.5818	0.5770*

* Optimal Value Of Criterion

Stepwise Selection Summary

Step	Effect Entered	Effect Removed	AIC	AICC	BIC
0	Intercept		204.2238	204.2699	-60.6397
1	crRuns		63.5391	63.6318	-200.7872
2	nHits		1.7041	1.8592	-261.8807
3	yrMajor		-12.0208	-11.7873	-275.3333
4		crRuns	-8.5517	-8.3967	-271.9095
5	nBB		-19.0690*	-18.8356*	-282.1700*

* Optimal Value Of Criterion

Stepwise Selection Summary

Step	Effect Entered	Effect Removed	CP	SBC	PRESS
0	Intercept		375.9275	-57.2041	208.7381
1	crRuns		111.2315	-194.3166	123.9195
2	nHits		33.4438	-252.5794	97.6368
3	yrMajor		18.5870	-262.7322	92.2998
4		crRuns	22.3357	-262.8353	93.1482
5	nBB		11.3524*	-269.7804*	89.5434*

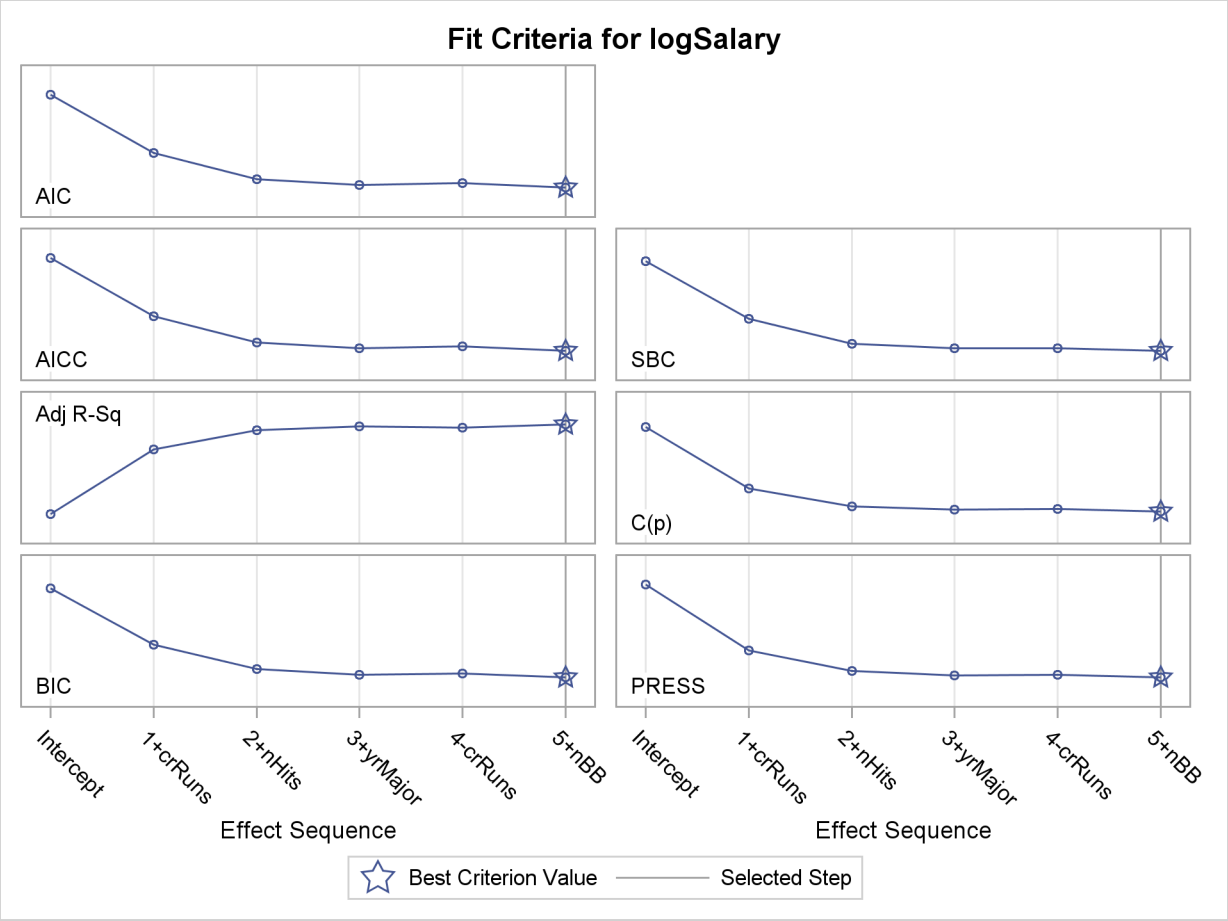
* Optimal Value Of Criterion

Stepwise Selection Summary

Step	Effect Entered	Effect Removed	ASE	F Value	Pr > F
0	Intercept		0.7877	0.00	1.0000
1	crRuns		0.4578	188.01	<.0001
2	nHits		0.3592	71.42	<.0001
3	yrMajor		0.3383	15.96	<.0001
4		crRuns	0.3454	5.44	0.0204
5	nBB		0.3294	12.62	0.0005

* Optimal Value Of Criterion

Figure 42.7 Criterion Panel



The stop reason and stop details tables in Figure 42.8 gives details of why the selection process terminated. This table shows that at step five the best add candidate, "division," and the best drop candidate, "nBB," yield models with SBC values of −268.6094 and −262.8353, respectively. Both of these values are larger than the current SBC value of −269.7804, and so the selection process stops at the model at step five.

Figure 42.8 Stopping Details

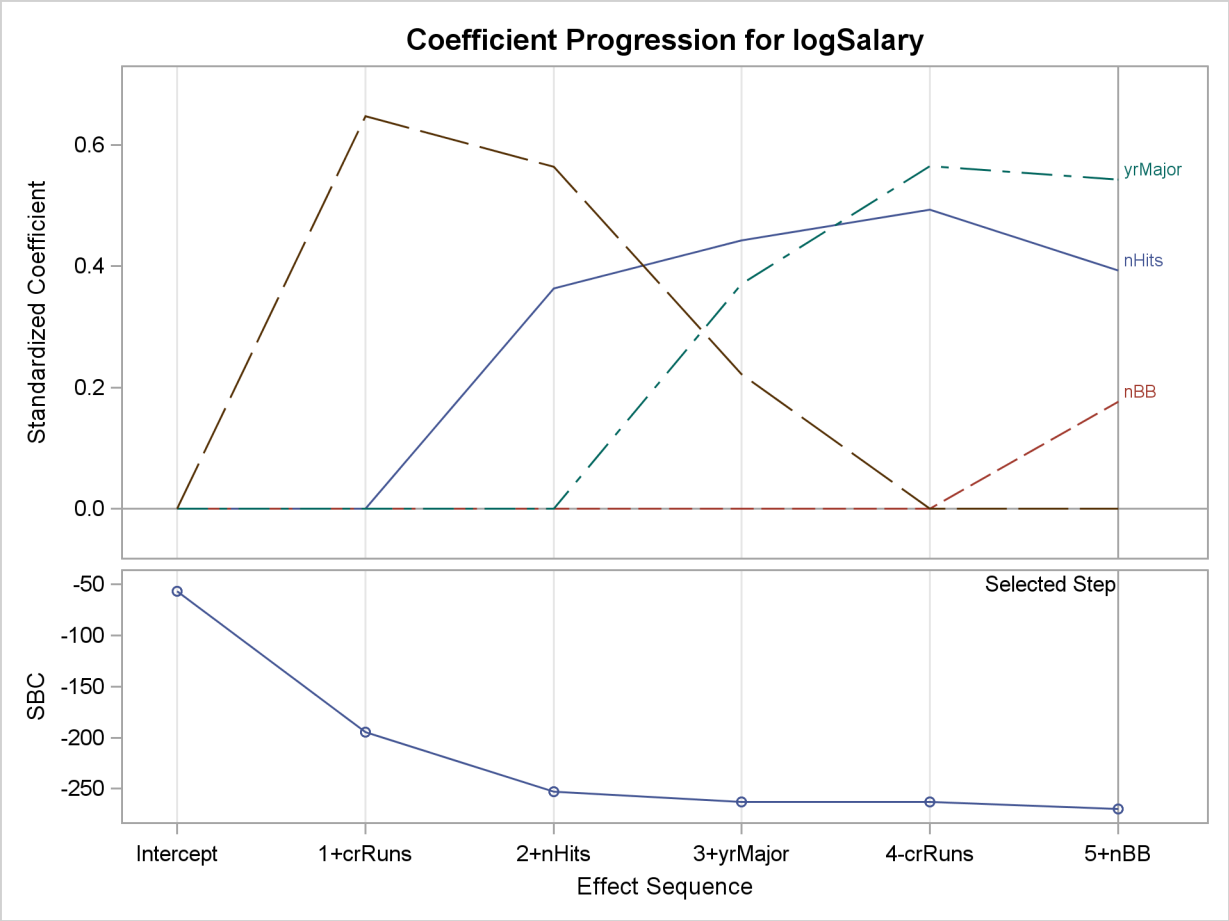
Selection stopped at a local minimum of the SBC criterion.

Stop Details

Candidate For	Effect	Candidate SBC		Compare SBC
Entry	division	−268.6094	>	−269.7804
Removal	nBB	−262.8353	>	−269.7804

The coefficient panel in Figure 42.9 enables you to visualize the selection process. In this plot, standardized coefficients of all the effects selected at some step of the stepwise method are plotted as a function of the step number. This enables you to assess the relative importance of the effects selected at any step of the selection process as well as providing information as to when effects entered the model. The lower plot in the panel shows how the criterion used to choose the selected model changes as effects enter or leave the model.

Figure 42.9 Coefficient Progression



The selected effects, analysis of variance, fit statistics, and parameter estimates tables shown in Figure 42.10 give details of the selected model.

Figure 42.10 Details of the Selected Model

```
                    The GLMSELECT Procedure
                         Selected Model

          The selected model is the model at the last step (Step 5).


                 Effects: Intercept nHits nBB yrMajor
```

Figure 42.10 *continued*

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value
Model	3	120.52553	40.17518	120.12
Error	259	86.62820	0.33447	
Corrected Total	262	207.15373		

Root MSE	0.57834
Dependent Mean	5.92722
R-Square	0.5818
Adj R-Sq	0.5770
AIC	-19.06903
AICC	-18.83557
BIC	-282.17004
C(p)	11.35235
PRESS	89.54336
SBC	-269.78041
ASE	0.32938

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value
Intercept	1	4.013911	0.111290	36.07
nHits	1	0.007929	0.000994	7.98
nBB	1	0.007280	0.002049	3.55
yrMajor	1	0.100663	0.007551	13.33

PROC GLMSELECT provides you with the flexibility to use several selection methods and many fit criteria for selecting effects that enter or leave the model. You can also specify criteria to determine when to stop the selection process and to choose among the models at each step of the selection process. You can find continued exploration of the baseball data that uses a variety of these methods in Example 42.1.

Syntax: GLMSELECT Procedure

The following statements are available in PROC GLMSELECT:

PROC GLMSELECT < *options* > ;
BY *variables* ;
CLASS *variable* < *(v-options)* > < *variable* < *(v-options . . .)* > > < */ v-options* > < *options* > ;
EFFECT *name = effect-type (variables* < */ options* > *)* ;
FREQ *variable* ;
MODEL *variable* = < *effects* > < */ options* > ;
OUTPUT < *OUT=SAS-data-set* > < *keyword* < *=name* > > < *. . . keyword=name* > ;
PARTITION < *options* > ;
PERFORMANCE < *options* > ;
SCORE < *DATA=SAS-data-set* > < *OUT=SAS-data-set* > ;
WEIGHT *variable* ;

The PROC GLMSELECT statement invokes the procedure. All statements other than the MODEL statement are optional and multiple SCORE statements can be used. CLASS and EFFECT statements, if present, must precede the MODEL statement.

PROC GLMSELECT Statement

PROC GLMSELECT < *options* > ;

Table 42.1 lists the options available in the PROC GLMSELECT statement.

Table 42.1 PROC GLMSELECT Statement Options

Option	Description
Data Set Options	
DATA=	names a data set to use for the regression
MAXMACRO=	sets the maximum number of macro variables produced
TESTDATA=	names a data set containing test data
VALDATA=	names a data set containing validation data
ODS Graphics Options	
PLOTS=	produces ODS graphical displays
Other Options	
OUTDESIGN=	requests a data set containing the design matrix
NAMELEN=	sets the length of effect names in tables and output data sets
NOPRINT	suppresses displayed output including plots
SEED=	sets the seed used for pseudo-random number generation

Following are explanations of the options that you can specify in the PROC GLMSELECT statement (in alphabetical order).

DATA=*SAS-data-set*

names the SAS data set to be used by PROC GLMSELECT. If the DATA= option is not specified, PROC GLMSELECT uses the most recently created SAS data set. If the named data set contains a variable named _ROLE_, then this variable is used to assign observations for training, validation, and testing roles. See the section "Using Validation and Test Data" on page 2739 for details on using the _ROLE_ variable.

MAXMACRO=*n*

specifies the maximum number of macro variables with selected effects to create. By default, MAXMACRO=100. PROC GLMSELECT saves the list of selected effects in a macro variable, &_GLSIND. Say your input effect list consists of x1-x10. Then &_GLSIND would be set to x1 x3 x4 x10 if, for example, the first, third, fourth, and tenth effects were selected for the model. This list can be used, for example, in the model statement of a subsequent procedure. If you specify the OUTDESIGN= option in the PROC GLMSELECT statement, then PROC GLMSELECT saves the list of columns in the design matrix in a macro variable named &_GLSMOD.

With BY processing, one macro variable is created for each BY group, and the macro variables are indexed by the BY group number. The MAXMACRO= option can be used to either limit or increase the number of these macro variables when you are processing data sets with many BY groups.

With no BY processing, PROC GLMSELECT creates the following:

_GLSIND	selected effects
_GLSIND1	selected effects
_GLSMOD	design matrix columns
_GLSMOD1	design matrix columns
_GLSNUMBYS	number of BY groups
_GLSNUMMACROBYS	number of _GLSIND*i* macro variables actually made

With BY processing, PROC GLMSELECT creates the following:

_GLSIND	selected effects for BY group 1
_GLSIND1	selected effects for BY group 1
_GLSIND2	selected effects for BY group 2
.	
.	
.	
_GLSIND*m*	selected effects for BY group *m*, where a number is substituted for *m*
_GLSMOD	design matrix columns for BY group 1
_GLSMOD1	design matrix columns for BY group 1
_GLSMOD2	design matrix columns for BY group 2
.	
.	
.	
_GLSMOD*m*	design matrix columns for BY group *m*, where a number is substituted for *m*
_GLSNUMBYS	*n*, the number of BY groups
_GLSNUMMACROBYS	the number *m* of _GLSIND*i* macro variables actually made. This value can be less than _GLSNUMBYS = *n*, and it is less than or equal to the MAXMACRO= value.

See the section "Macro Variables Containing Selected Models" on page 2734 for further details.

NOPRINT

suppresses all displayed output including plots.

NAMELEN=*n*

specifies the length of effect names in tables and output data sets to be *n* characters long, where *n* is a value between 20 and 200 characters. The default length is 20 characters.

OUTDESIGN <(*options*)><=SAS-data-set>

creates a data set that contains the design matrix. By default, the GLMSELECT procedure includes in the OUTDESIGN data set the **X** matrix corresponding to the parameters in the selected model. Two schemes for naming the columns of the design matrix are available. In the first scheme, names of the parameters are constructed from the parameter labels that appear in the "ParameterEstimates" table. This naming scheme is the default when you do not request BY processing and is not available when you do use BY processing. In the second scheme, the design matrix column names consist of a prefix followed by an index. The default naming prefix is "_X".

You can specify the following *options* in parentheses to control the contents of the OUTDESIGN data set:

ADDINPUTVARS

requests that all variables in the input data set be included in the OUTDESIGN= data set.

FULLMODEL
specifies that parameters corresponding to all the effects specified in the MODEL statement be included in the OUTDESIGN= data set. By default, only parameters corresponding to the selected model are included.

NAMES
produces a table associating columns in the OUTDESIGN data set with the labels of the parameters they represent.

PREFIX< =prefix >
requests that the design matrix column names consist of a prefix followed by an index. The default naming prefix is "_X". You can optionally specify a different prefix.

PARMLABELSTYLE=*options*
specifies how parameter names and labels are constructed for nested and crossed effects.

The following *options* are available:

INTERLACED < (SEPARATOR=*quoted string***) >**
forms parameter names and labels by positioning levels of classification variables and constructed effects adjacent to the associated variable or constructed effect name and using " * " as the delimiter for both crossed and nested effects. This style of naming parameters and labels is used in the TRANSREG procedure. You can request truncation of the classification variable names used in forming the parameter names and labels by using the CPREFIX= and LPREFIX= options in the CLASS statement. You can use the SEPARATOR= suboption to change the delimiter between the crossed variables in the effect. PARMLABELSTYLE=INTERLACED is not supported if you specify the SPLIT option in an EFFECT statement or a CLASS statement. The following are examples of the parameter labels in this style (Age is a continuous variable, Gender and City are classification variables):

```
Age
Gender male * City Beijing
City London * Age
```

SEPARATE
specifies that in forming parameter names and labels, the effect name appears before the levels associated with the classification variables and constructed effects in the effect. You can control the length of the effect name by using the NAMELEN= option in the PROC GLMSELECT statement. In forming parameter labels, the first level that is displayed is positioned so that it starts at the same offset in every parameter label—this enables you to easily distinguish the effect name from the levels when the parameter labels are displayed in a column in the "Parameter Estimates" table. This style of labeling is used in the GLM procedure and is the default if you do not specify the PARMLABELSTYLE option. The following are examples of the parameter labels in this style (Age is a continuous variable, Gender and City are classification variables):

```
Age
Gender*City male Beijing
Age*City    London
```

SEPARATECOMPACT

requests the same parameter naming and labeling scheme as PARMLABELSTYLE=SEPARATE except that the first level in the parameter label is separated from the effect name by a single blank. This style of labeling is used in the PLS procedure. The following are examples of the parameter labels in this style (Age is a continuous variable, Gender and City are classification variables):

```
Age
Gender*City male Beijing
Age*City London
```

PLOTS < (*global-plot-options*) > < = *plot-request* < *(options)* > >

PLOTS < (*global-plot-options*) > < = (*plot-request* < *(options)* > < ... *plot-request* < *(options)* > >) >

controls the plots produced through ODS Graphics. When you specify only one plot request, you can omit the parentheses around the plot request. Here are some examples:

```
plots=all
plots=coefficients(unpack)
plots(unpack)=(criteria candidates)
```

You must enable ODS Graphics before requesting plots as shown in the following example. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

```
ods graphics on;

proc glmselect plots=all;
   model y = x1-x100;
run;

ods graphics off;
```

Global Plot Options

The *global-options* apply to all plots generated by the GLMSELECT procedure, unless it is altered by a *specific-plot-option*.

ENDSTEP=*n*

specifies that the step ranges shown on the horizontal axes of plots terminates at specified step. By default, the step range shown terminates at the final step of the selection process. If you specify the ENDSTEP= option as both a global plot option and a specific plot option, then the ENDSTEP= value on the specific plot is used.

LOGP | LOGPVALUE

requests that the natural logarithm of the entry and removal significance levels be displayed. This option is ignored if the select criterion is not significance level.

MAXSTEPLABEL=*n*

specifies the maximum number of characters beyond which labels of effects on plots are truncated.

MAXPARMLABEL= *n*
specifies the maximum number of characters beyond which parameter labels on plots are truncated.

STARTSTEP=*n*
specifies that the step ranges shown on the horizontal axes of plots start at the specified step. By default, the step range shown starts at the initial step of the selection process. If you specify the STARTSTEP= option both as a global plot option and a specific plot option, then the STARTSTEP= value on the specific plot is used.

STEPAXIS=EFFECT | NORMB | NUMBER
specifies the horizontal axis to be used on the plots, where this axis represents the sequence of entering or departing effects.

STEPAXIS=EFFECT
requests that each step be labeled by a prefix followed by the name of the effect that enters or leaves at that step. The prefix consists of the step number followed by a "+" sign or a "-" sign depending on whether the effect enters or leaves at that step.

STEPAXIS=NORMB
is valid only with LAR and LASSO selection methods and requests that the horizontal axis value at step i be the L1 norm of the parameters at step i, normalized by the L1 norm of the parameters at the final step.

STEPAXIS=NUMBER
requests that each step be labeled by the step number.

UNPACK
suppresses paneling. By default, multiple plots can appear in some output panels. Specify UNPACK to get each plot individually. You can also specify UNPACK as a suboption with CRITERIA and COEFFICIENTS.

Specific Plot Options

The following listing describes the specific plots and their options.

ALL
requests that all default plots be produced. Note that candidate plots are produced only if you specify DETAILS=STEPS or DETAILS=ALL in the MODEL statement.

ASE | ASEPLOT <(*aseplot-option***)>**
plots the progression of the average square error on the training data, and the test and validation data whenever these data are provided with the TESTDATA= and VALDATA= options or are produced by using a PARTITION statement. The following *aseplot-option* option is available:

STEPAXIS=EFFECT | NORMB | NUMBER
specifies the horizontal axis to be used.

CANDIDATES | CANDIDATESPLOT <(*candidatesplot-options*)>
produces a needle plot of the select criterion values for the candidates for entry or removal at each step of the selection process, ordered from best to worst. Candidates plots are not available if you specify SELECTION=NONE, SELECTION=LAR, or SELECTION=LASSO in the MODEL statement, or if you have not specified DETAILS=ALL or DETAILS=STEPS in the MODEL statement. The following *candidatesplot-options* are available:

LOGP | LOGPVALUE
requests that the natural logarithm of the entry and removal significance levels be displayed. This option is ignored if the select criterion is not significance level.

SHOW=*number*
specifies the maximum number of candidates displayed at each step. The default is SHOW=10.

COEFFICIENTS | COEFFICIENTPANEL <(*coefficientPanel-options*)>
plots a panel of two plots. The upper plot shows the progression of the parameter values as the selection process proceeds. The lower plot shows the progression of the CHOOSE= criterion. If no choose criterion is in effect, then the AICC criterion is displayed. The following *coefficientPanel-options* are available:

LABELGAP=*percentage*
specifies the percentage of the vertical axis range that forms the minimum gap between successive parameter labels at the final step of the coefficient progression plot. If the values of more than one parameter at the final step are closer than this gap, then the labels on all but one of these parameters is suppressed. The default value is LABELGAP=5. Planned enhancements to the automatic label collision avoidance algorithm will obviate the need for this option in future releases of the GLMSELECT procedure.

LOGP | LOGPVALUE
requests that the natural logarithm of the entry and removal significance levels be displayed if the choose criterion is significance level.

STEPAXIS=EFFECT | NORMB | NUMBER
specifies the horizontal axis to be used.

UNPACK | UNPACKPANEL
displays the coefficient progression and the choose criterion progression in separate plots.

CRITERIA | CRITERIONPANEL <(*criterionPanel-options*)>
plots a panel of model fit criteria. The criteria that are displayed are ADJRSQ, AIC, AICC, and SBC, as well as any other criteria that are named in the CHOOSE=, SELECT=, STOP=, or STATS= option in the MODEL statement. The following *criterionPanel-options* are available:

STEPAXIS=EFFECT | NORMB | NUMBER
specifies the horizontal axis to be used.

UNPACK | UNPACKPANEL
displays each criterion progression on a separate plot.

NONE
suppresses all plots.

SEED=*number*
specifies an integer used to start the pseudo-random number generator for random cross validation and random partitioning of data for training, testing, and validation. If you do not specify a seed, or if you specify a value less than or equal to zero, the seed is generated from reading the time of day from the computer's clock.

TESTDATA=*SAS-data-set*
names a SAS data set containing test data. This data set must contain all the variables specified in the MODEL statement. Furthermore, when a BY statement is used and the TESTDATA=data set contains any of the BY variables, then the TESTDATA= data set must also contain all the BY variables sorted in the order of the BY variables. In this case, only the test data for a specific BY group is used with the corresponding BY group in the analysis data. If the TESTDATA= data set contains none of the BY variables, then the entire TESTDATA = data set is used with each BY group of the analysis data.

If you specify a TESTDATA=data set, then you cannot also reserve observations for testing by using a PARTITION statement.

VALDATA=*SAS-data-set*
names a SAS data set containing validation data. This data set must contain all the variables specified in the MODEL statement. Furthermore, when a BY statement is used and the VALDATA=data set contains any of the BY variables, then the VALDATA= data set must also contain all the BY variables sorted in the order of the BY variables. In this case, only the validation data for a specific BY group are used with the corresponding BY group in the analysis data. If the VALDATA= data set contains none of the BY variables, then the entire VALDATA = data set is used with each BY group of the analysis data.

If you specify a VALDATA=data set, then you cannot also reserve observations for validation by using a PARTITION statement.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC GLMSELECT to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the GLMSELECT procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in base SAS software).

For more information about the BY statement, see *SAS Language Reference: Contents*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

The CLASS statement names the classification variables to be used in the analysis. The CLASS statement must precede the MODEL statement.

The following options can be specified after a slash (/):

DELIMITER=*quoted character*
specifies the delimiter that is used between levels of classification variables in building parameter names and lists of class level values. The default if you do not specify DELIMITER= is a space. This option is useful if the levels of a classification variable contain embedded blanks.

SHOW | SHOWCODING
requests a table for each classification variable that shows the coding used for that variable.

You can specify various *v-options* for each variable by enclosing them in parentheses after the variable name. You can also specify global *v-options* for the CLASS statement by placing them after a slash (/). Global *v-options* are applied to all the variables specified in the CLASS statement. If you specify more than one CLASS statement, the global *v-options* specified in any one CLASS statement apply to all CLASS statements. However, individual CLASS variable *v-options* override the global *v-options*.

The following *v-options* are available:

CPREFIX=*n*

specifies that, at most, the first n characters of a CLASS variable name be used in creating names for the corresponding design variables. The default is $32 - \min(32, \max(2, f))$, where f is the formatted length of the CLASS variable. The CPREFIX= applies only when you specify the PARMLABELSTYLE=INTERLACED option in the PROC GLMSELECT statement.

DESCENDING

DESC

reverses the sorting order of the classification variable.

LPREFIX=*n*

specifies that, at most, the first n characters of a CLASS variable label be used in creating labels for the corresponding design variables. The default is $256 - \min(256, \max(2, f))$, where f is the formatted length of the CLASS variable. The LPREFIX= applies only when you specify the PARMLABELSTYLE=INTERLACED option in the PROC GLMSELECT statement.

MISSING

allows missing value ('.' for a numeric variable and blanks for a character variables) as a valid value for the CLASS variable.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of classification variables. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option might be useful when you use the CONTRAST or ESTIMATE statement. If ORDER=FORMATTED for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. Note that this represents a change from previous releases for how class levels are ordered. Before SAS 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values, and in order to revert to the previous ordering you can specify this format explicitly for the affected classification variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and usually required the user to intervene with an explicit format or ORDER=INTERNAL to get the more natural ordering. The following table shows how PROC GLMSELECT interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent.

For more information about sorting order, refer to the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

PARAM=*keyword*

specifies the parameterization method for the classification variable or variables. Design matrix columns are created from CLASS variables according to the following coding schemes. The default is PARAM=GLM. If PARAM=ORTHPOLY or PARAM=POLY, and the CLASS levels are numeric, then the ORDER= option in the CLASS statement is ignored, and the internal, unformatted values are used. See the section "CLASS Variable Parameterization" on page 2730 for further details.

EFFECT	specifies effect coding.
GLM	specifies less-than-full-rank, reference-cell coding; this option can be used only as a global option.
ORDINAL THERMOMETER	specifies the cumulative parameterization for an ordinal CLASS variable.
POLYNOMIAL POLY	specifies polynomial coding.
REFERENCE REF	specifies reference-cell coding.
ORTHEFFECT	orthogonalizes PARAM=EFFECT.
ORTHORDINAL	
ORTHOTHERM	orthogonalizes PARAM=ORDINAL.
ORTHPOLY	orthogonalizes PARAM=POLYNOMIAL.
ORTHREF	orthogonalizes PARAM=REFERENCE.

The EFFECT, POLYNOMIAL, REFERENCE, and ORDINAL schemes and their orthogonal parameterizations are full rank. The REF= option in the CLASS statement determines the reference level for the EFFECT and REFERENCE schemes and their orthogonal parameterizations.

REF=*'level'* | *keyword*

specifies the reference level for PARAM=EFFECT, PARAM=REFERENCE, and their orthogonalizations. For an individual (but not a global) variable REF= *option*, you can specify the *level* of the variable to use as the reference level. For a global or individual variable REF= *option*, you can use one of the following *keywords*. The default is REF=LAST.

FIRST	designates the first-ordered level as reference.
LAST	designates the last-ordered level as reference.

SPLIT

requests that the columns of the design matrix corresponding to any effect containing a split classification variable can be selected to enter or leave a model independently of the other design columns of that effect. For example, suppose a variable named temp has three levels with values "hot," "warm," and "cold," and a variable named sex has two levels with values "M" and "F" are used in a PROC GLMSELECT job as follows:

```
proc glmselect;
   class temp sex/split;
   model depVar = sex sex*temp;
run;
```

As both the classification variables are split, the two effects named in the MODEL statement are split into eight independent effects. The effect "sex" is split into two effects labeled "sex_M" and "sex_F". The effect "sex*temp" is split into six effects labeled "sex_M*temp_hot", "sex_F*temp_hot", "sex_M*temp_warm", "sex_F*temp_warm", "sex_M*temp_cold", and "sex_F*temp_cold", and the previous PROC GLMSELECT step is equivalent to the following:

```
proc glmselect;
   model depVar =  sex_M sex_F sex_M*temp_hot  sex_F*temp_hot
                               sex_M*temp_warm sex_F*temp_warm
                               sex_M*temp_cold sex_F*temp_cold;
run;
```

The split option can be used on individual classification variables. For example, consider the following PROC GLMSELECT step:

```
proc glmselect;
   class temp(split) sex;
   model depVar = sex sex*temp;
run;
```

In this case the effect "sex" is not split and the effect "sex*temp" is split into three effects labeled "sex*temp_hot", "sex*temp_warm", and "sex*temp_cold". Furthermore each of these three split effects now has two parameters corresponding to the two levels of "sex," and the PROC GLMSELECT step is equivalent to the following:

```
proc glmselect;
   class sex;
   model depVar = sex sex*temp_hot sex*temp_warm sex*temp_cold;
run;
```

EFFECT Statement

EFFECT *effect-specification* **;**

The experimental EFFECT statement enables you to construct special collections of columns for the **X** matrix in your model. These collections are referred to as *constructed effects* to distinguish them from the usual model effects formed from continuous or classification variables.

For details about the syntax of the EFFECT statement and how columns of constructed effects are computed, see the section "Constructed Effects and the EFFECT Statement (Experimental)" on page 377 of Chapter 18, "Shared Concepts and Topics."

FREQ Statement

FREQ *variable* **;**

The variable specified in the FREQ statement identifies a variable in the input data set containing the frequency of occurrence of each observation. PROC GLMSELECT treats each observation as if it appears n times, where n is the value of the FREQ variable for the observation. If it is not an integer, the frequency value is truncated to an integer. If it is less than 1 or if it is missing, the observation is not used.

MODEL Statement

MODEL *dependent*=< *effects* > / < *options* > **;**

The MODEL statement names the dependent variable and the explanatory effects, including covariates, main effects, constructed effects, interactions, and nested effects; see the section "Specification of Effects" on page 2488 in Chapter 39, "The GLM Procedure," for more information. If you omit the explanatory effects, the procedure fits an intercept-only model.

After the keyword MODEL, the dependent (response) variable is specified, followed by an equal sign. The explanatory effects follow the equal sign.

Table 42.2 lists the options available in the MODEL statement.

Table 42.2 MODEL Statement Options

Option	Description
CVDETAILS=	requests details when cross validation is used
CVMETHOD=	specifies how subsets for cross validation are formed
DETAILS=	specifies details to be displayed
HIERARCHY=	specifies hierarchy of effects to impose
NOINT	specifies models without an explicit intercept
ORDERSELECT	requests that parameter estimates be displayed in the order in which the parameters first entered the model

Table 42.2 *continued*

Option	Description
SELECTION=	specifies model selection method
STATS=	specifies additional statistics to be displayed
SHOWPVALUES	requests p-values in "ANOVA" and "Parameter Estimates" tables
STB	adds standardized coefficients to "Parameter Estimates" tables

You can specify the following options in the MODEL statement after a slash (/):

CVDETAILS=ALL

CVDETAILS=COEFFS

CVDETAILS=CVPRESS

specifies the details produced when cross validation is requested as the CHOOSE=, SELECT=, or STOP= criterion in the MODEL statement. If n-fold cross validation is being used, then the training data are subdivided into n parts, and at each step of the selection process, models are obtained on each of the n subsets of the data obtained by omitting one of these parts. CVDETAILS=COEFFS requests that the parameter estimates obtained for each of these n subsets be included in the parameter estimates table. CVDETAILS=CVPRESS requests a table containing the predicted residual sum of squares of each of these models scored on the omitted subset. CVDETAILS=ALL requests both CVDETAILS=COEFFS and CVDETAILS=CVPRESS. If DETAILS=STEPS or DETAILS=ALL has been specified in the MODEL statement, then the requested CVDETAILS are produced for every step of the selection process.

CVMETHOD=BLOCK < (*n*) >

CVMETHOD=RANDOM < (*n*) >

CVMETHOD=SPLIT < (*n*) >

CVMETHOD=INDEX (*variable*)

specifies how the training data are subdivided into n parts when you request n-fold cross validation by using any of the CHOOSE=CV, SELECT=CV, and STOP=CV suboptions of the SELECTION= option in the MODEL statement.

- CVMETHOD=BLOCK requests that parts be formed of n blocks of consecutive training observations.
- CVMETHOD=SPLIT requests that the ith part consist of training observations $i, i + n, i + 2n, \ldots$.
- CVMETHOD=RANDOM assigns each training observation randomly to one of the n parts.
- CVMETHOD=INDEX(*variable*) assigns observations to parts based on the formatted value of the named variable. This input data set variable is treated as a classification variable and the number of parts n is the number of distinct levels of this variable. By optionally naming this variable in a CLASS statement you can use the CLASS statement options ORDER= and MISSING to control how the levelization of this variable is done.

n defaults to 5 with CVMETHOD=BLOCK, CVMETHOD=SPLIT, or CVMETHOD=RANDOM. If you do not specify the CVMETHOD= option, then the CVMETHOD defaults to CVMETHOD=RANDOM(5).

DETAILS=*level*

DETAILS=STEPS <(*step options***)>**

specifies the level of detail produced, where *level* can be ALL, STEPS, or SUMMARY. The default if the DETAILS= option is omitted is DETAILS=SUMMARY. The DETAILS=ALL option produces the following:

- entry and removal statistics for each variable selected in the model building process
- ANOVA, fit statistics, and parameter estimates
- entry and removal statistics for the top 10 candidates for inclusion or exclusion at each step
- a selection summary table

The DETAILS=SUMMARY option produces only the selection summary table.

The option DETAILS=STEPS <(*step options*)> provides the step information and the selection summary table. The following options can be specified within parentheses after the DETAILS=STEPS option:

ALL

requests ANOVA, fit statistics, parameter estimates, and entry or removal statistics for the top 10 candidates for inclusion or exclusion at each selection step.

ANOVA

requests ANOVA at each selection step.

FITSTATISTICS | FITSTATS | FIT

requests fit statistics at each selection step. The default set of statistics includes all of the statistics named in the CHOOSE=, SELECT=, and STOP= suboptions specified in the MODEL statement SELECTION= option, but additional statistics can be requested with the STATS= option in the MODEL statement.

PARAMETERESTIMATES | PARMEST

requests parameter estimates at each selection step.

CANDIDATES <(SHOW= ALL | *n*)>

requests entry or removal statistics for the best *n* candidate effects for inclusion or exclusion at each step. If you specify SHOW=ALL, then all candidates are shown. If SHOW= is not specified. then the best 10 candidates are shown. The entry or removal statistic is the statistic named in the SELECT= option that is specified in the MODEL statement SELECTION= option.

HIERARCHY=*keyword*

HIER=*keyword*

specifies whether and how the model hierarchy requirement is applied. This option also controls whether a single effect or multiple effects are allowed to enter or leave the model in

one step. You can specify that only classification effects, or both classification and continuous effects, be subject to the hierarchy requirement. The HIERARCHY= option is ignored unless you also specify one of the following options: SELECTION=FORWARD, SELECTION=BACKWARD, or SELECTION=STEPWISE.

Model hierarchy refers to the requirement that for any term to be in the model, all model effects contained in the term must be present in the model. For example, in order for the interaction A*B to enter the model, the main effects A and B must be in the model. Likewise, neither effect A nor effect B can leave the model while the interaction A*B is in the model.

The keywords you can specify in the HIERARCHY= option are as follows:

NONE
specifies that model hierarchy not be maintained. Any single effect can enter or leave the model at any given step of the selection process.

SINGLE
specifies that only one effect enter or leave the model at one time, subject to the model hierarchy requirement. For example, suppose that the model contains the main effects A and B and the interaction A*B. In the first step of the selection process, either A or B can enter the model. In the second step, the other main effect can enter the model. The interaction effect can enter the model only when both main effects have already entered. Also, before A or B can be removed from the model, the A*B interaction must first be removed. All effects (CLASS and interval) are subject to the hierarchy requirement.

SINGLECLASS
is the same as HIERARCHY=SINGLE except that only CLASS effects are subject to the hierarchy requirement.

The default value is HIERARCHY=NONE.

NOINT
suppresses the intercept term that is otherwise included in the model.

ORDERSELECT
specifies that for the selected model, effects be displayed in the order in which they first entered the model. If you do not specify the ORDERSELECT option, then effects in the selected model are displayed in the order in which they appeared in the MODEL statement.

SELECTION=*method* < (method options) >
specifies the method used to select the model, optionally followed by parentheses enclosing options applicable to the specified method. The default if the SELECTION= option is omitted is SELECTION=STEPWISE.

The following methods are available and are explained in detail in the section "Model-Selection Methods" on page 2719.

NONE	no model selection
FORWARD	forward selection. This method starts with no effects in the model and adds effects.

BACKWARD — backward elimination. This method starts with all effects in the model and deletes effects.

STEPWISE — stepwise regression. This is similar to the FORWARD method except that effects already in the model do not necessarily stay there.

LAR — least angle regression. This method, like forward selection, starts with no effects in the model and adds effects. The parameter estimates at any step are "shrunk" when compared to the corresponding least squares estimates. If the model contains classification variables, then these classification variables are split. See the SPLIT option in the CLASS statement for details.

LASSO — This method adds and deletes parameters based on a version of ordinary least squares where the sum of the absolute regression coefficients is constrained. If the model contains classification variables, then these classification variables are split. See the SPLIT option in the CLASS statement for details.

Table 42.3 lists the applicable suboptions for each of these methods.

Table 42.3 Applicable SELECTION= Options by Method

Option	FORWARD	BACKWARD	STEPWISE	LAR	LASSO
STOP =	x	x	x	x	
CHOOSE =	x	x	x	x	
STEPS =	x	x	x	x	
MAXSTEPS =	x	x	x	x	
SELECT =	x	x	x		
INCLUDE =	x	x	x		
SLENTRY =	x		x		
SLSTAY =		x	x		
DROP =			x		
LSCOEFFS					x

The syntax of the suboptions that you can specify in parentheses after the SELECTION= option method follows. Note that, as described in Table 42.3, not all selection suboptions are applicable to every SELECTION= method.

CHOOSE=*criterion*

chooses from the list of models at the steps of the selection process the model that yields the best value of the specified criterion. If the optimal value of the specified criterion occurs for models at more than one step, then the model with the smallest number of parameters is chosen. If you do not specify the CHOOSE= option, then the model selected is the model at the final step in the selection process.

The criteria that you can specify in the CHOOSE= option are shown in Table 42.4. See the section "Criteria Used in Model Selection Methods" on page 2727 for more detailed descriptions of these criteria.

Table 42.4 Criteria for the CHOOSE= Option

Option	Criteria
ADJRSQ	Adjusted R-square statistic
AIC	Akaike information criterion
AICC	Corrected Akaike information criterion
BIC	Sawa Bayesian information criterion
CP	Mallows C(p) statistic
CV	Predicted residual sum of square with k-fold cross validation
PRESS	Predicted residual sum of squares
SBC	Schwarz Bayesian information criterion
VALIDATE	Average square error for the validation data

For ADJRSQ the chosen value is the largest one; for all other criteria, the smallest value is chosen. You can use the VALIDATE option only if you have specified a VALDATA= data set in the PROC GLMSELECT statement or if you have reserved part of the input data for validation by using either a PARTITION statement or a _ROLE_ variable in the input data.

DROP=*policy*

specifies when effects are eligible to be dropped in the STEPWISE method. Valid values for policy are BEFOREADD and COMPETITIVE.

If you specify DROP=BEFOREADD, then effects currently in the model are examined to see if any meet the requirements to be removed from the model. If so, the effect that gives the best value of the removal criterion is dropped from the model and the stepwise method proceeds to the next step. Only when no effect currently in the model meets the requirement to be removed from the model are any effects added to the model.

DROP=COMPETITIVE can be specified only if the SELECT= criterion is not SL. If you specify DROP=COMPETITIVE, then the SELECT= criterion is evaluated for all models where an effect currently in the model is dropped or an effect not yet in the model is added. The effect whose removal or addition to the model yields the maximum improvement to the SELECT= criterion is dropped or added.

The default if you do not specify DROP= suboption with the STEPWISE method is DROP=BEFOREADD. If SELECT=SL, then this yields the traditional stepwise method as implemented in PROC REG.

INCLUDE=*n*

forces the first n effects listed in the MODEL statement to be included in all models. The selection methods are performed on the other effects in the MODEL statement. The INCLUDE= option is available only with SELECTION=FORWARD, SELECTION=STEPWISE, and SELECTION=BACKWARD.

LSCOEFFS

requests a hybrid version of the LAR and LASSO methods, where the sequence of models is determined by the LAR or LASSO algorithm but the coefficients of the parameters for the model at any step are determined by using ordinary least squares.

MAXSTEP=*n*

specifies the maximum number of selection steps that are done. The default value of *n* is the number of effects in the model statement for the FORWARD, BACKWARD, and LAR methods and is three times the number of effects for the STEPWISE and LASSO methods.

SELECT=*criterion*

specifies the criterion that PROC GLMSELECT uses to determine the order in which effects enter and/or leave at each step of the specified selection method. The SELECT option is not valid with the LAR and LASSO methods. The criteria that you can specify with the SELECT= option are ADJRSQ, AIC, AICC, BIC, CP, CV, PRESS, RSQUARE, SBC, SL, and VALIDATE. See the section "Criteria Used in Model Selection Methods" on page 2727 for a description of these criteria. The default value of the SELECT= criterion is SELECT=SBC. You can use SELECT=SL to request the traditional approach where effects enter and leave the model based on the significance level. With other SELECT= criteria, the effect that is selected to enter or leave at a step of the selection process is the effect whose addition to or removal from the current model gives the maximum improvement in the specified criterion.

SLENTRY=*value*

SLE=*value*

specifies the significance level for entry, used when the STOP=SL or SELECT=SL option is in effect. The default is 0.15.

SLSTAY=*value*

SLS=*value*

specifies the significance level for staying in the model, used when the STOP=SL or SELECT=SL option is in effect. The default is 0.15.

STEPS=*n*

specifies the number of selection steps to be done. If the STEPS= option is specified, the STOP= and MAXSTEP= options are ignored.

STOP=*n*

STOP=*criterion*

specifies when PROC GLMSELECT stops the selection process. If the STEPS= option is specified, then the STOP= option is ignored. If the STOP=option does not cause the selection process to stop before the maximum number of steps for the selection method, then the selection process terminates at the maximum number of steps.

If you do not specify the STOP= option but do specify the SELECT= option, then the criterion named in the SELECT=option is also used as the STOP= criterion. If you do not specify either the STOP= or SELECT= option, then the default is STOP=SBC.

If STOP=*n* is specified, then PROC GLMSELECT stops selection at the first step for which the selected model has *n* effects.

The nonnumeric arguments that you can specify in the STOP= option are shown in Table 42.5. See the section "Criteria Used in Model Selection Methods" on page 2727 for more detailed descriptions of these criteria.

Table 42.5 Nonnumeric Criteria for the STOP= Option

Option	Criteria
NONE	
ADJRSQ	Adjusted R-square statistic
AIC	Akaike information criterion
AICC	Corrected Akaike information criterion
BIC	Sawa Bayesian information criterion
CP	Mallows C(p) statistic
CV	Predicted residual sum of square with k-fold cross validation
PRESS	Predicted residual sum of squares
SBC	Schwarz Bayesian information criterion
SL	Significance level
VALIDATE	Average square error for the validation data

With the SL criterion, selection stops at the step where the significance level for entry of all the effects not yet in the model is greater than the SLE= value for addition steps in the FORWARDS and STEPWISE methods and where the significance level for removal of any effect in the current model is greater than the SLS= value in the BACKWARD and STEPWISE methods. With the ADJRSQ criterion, selection stops at the step where the next step would yield a model with a smaller value of the Adjusted R-square statistic; for all other criteria, selection stops at the step where the next step would yield a model with a larger value of the criteria. You can use the VALIDATE option only if you have specified a VALDATA= data set in the PROC GLMSELECT statement or if you have reserved part of the input data for validation by using either a PARTITION statement or a _ROLE_ variable in the input data.

STAT|STATS=*name*

STATS=(*names*)

specifies which model fit statistics are displayed in the fit summary table and fit statistics tables. If you omit the STATS= option, the default set of statistics that are displayed in these tables includes all the criteria specified in any of the CHOOSE=, SELECT=, and STOP= options specified in the MODEL statement SELECTION= option.

The statistics that you can specify follow:

ADJRSQ — the adjusted R-square statistic

AIC — the Akaike information criterion

AICC — the corrected Akaike information criterion

ASE — the average square errors for the training, test, and validation data. The ASE statistics for the test and validation data are reported only if you have specified TESTDATA= and/or VALDATA= in the PROC GLMSELECT statement or if you have reserved part of the input data for testing and/or validation by using either a PARTITION statement or a _ROLE_ variable in the input data.

BIC — the Sawa Bayesian information criterion

CP	the Mallows C(p) statistic
FVALUE	the F statistic for entering or departing effects
PRESS	the predicted residual sum of squares statistic
RSQUARE	the R-square statistic
SBC	the Schwarz Bayesian information criterion
SL	the significance level of the F statistic for entering or departing effects

The statistics ADJRSQ, AIC, AICC, FVALUE, RSQUARE, SBC, and SL can be computed with little computation cost. However, computing BIC, CP, CVPRESS, PRESS, and ASE for test and validation data when these are not used in any of the CHOOSE=, SELECT=, and STOP= options specified in the MODEL statement SELECTION= option can hurt performance.

SHOWPVALUES
SHOWPVALS

displays p-values in the "ANOVA" and "Parameter Estimates" tables. These p-values are generally liberal because they are not adjusted for the fact that the terms in the model have been selected.

STB

produces standardized regression coefficients. A standardized regression coefficient is computed by dividing a parameter estimate by the ratio of the sample standard deviation of the dependent variable to the sample standard deviation of the regressor.

OUTPUT Statement

OUTPUT < **OUT=***SAS-data-set* > < *keyword* < *=name* > > . . . < *keyword* < *=name* > > **;**

The OUTPUT statement creates a new SAS data set that saves diagnostic measures calculated for the selected model. If you do not specify a *keyword*, then the only diagnostic included is the predicted response.

All the variables in the original data set are included in the new data set, along with variables created in the OUTPUT statement. These new variables contain the values of a variety of statistics and diagnostic measures that are calculated for each observation in the data set. If you specify a BY statement, then a variable _BY_ that indexes the BY groups is included. For each observation, the value of _BY_ is the index of the BY group to which this observation belongs. This variable is useful for matching BY groups with macro variables that PROC GLMSELECT creates. See the section "Macro Variables Containing Selected Models" on page 2734 for details.

If you have requested n-fold cross validation by requesting CHOOSE=CV, SELECT=CV, or STOP=CV in the MODEL statement, then a variable _CVINDEX_ is included in the output data set. For each observation used for model training the value of _CVINDEX_ is i if that observation is omitted in forming the ith subset of the training data. See the CVMETHOD= for additional details. The value of _CVINDEX_ is 0 for all observations in the input data set that are not used for model training.

If you have partitioned the input data with a PARTITION statement, then a character variable _ROLE_ is included in the output data set. For each observation the value of _ROLE_ is as follows:

ROLE	Observation Role
TEST	testing
TRAIN	training
VALIDATE	validation

If you want to create a permanent SAS data set, you must specify a two-level name (for example, *libref.data-set-name*).

For more information on permanent SAS data sets, refer to the section "SAS Files" in *SAS Language Reference: Concepts*.

Details on the specifications in the OUTPUT statement follow.

keyword < *=name* >
: specifies the statistics to include in the output data set and optionally names the new variables that contain the statistics. Specify a keyword for each desired statistic (see the following list of keywords), followed optionally by an equal sign, and a variable to contain the statistic.

  If you specify *keyword=name*, the new variable that contains the requested statistic has the specified name. If you omit the optional *=name* after a *keyword*, then the new variable name is formed by using a prefix of one or more characters that identify the statistic, followed by an underscore (_), followed by the dependent variable name.

  The keywords allowed and the statistics they represent are as follows:

  PREDICTED | PRED | P predicted values. The prefix for the default name is *p*.

  RESIDUAL | RESID | R residual, calculated as ACTUAL − PREDICTED. The prefix for the default name is *r*.

OUT=*SAS data set*
: gives the name of the new data set. By default, the procedure uses the DATA*n* convention to name the new data set.

PARTITION Statement

The PARTITION statement specifies how observations in the input data set are logically partitioned into disjoint subsets for model training, validation, and testing. Either you can designate a variable in the input data set and a set of formatted values of that variable to determine the role of each observation, or you can specify proportions to use for random assignment of observations for each role.

An alternative to using a PARTITION statement is to provide a variable named _ROLE_ in the input data set to define roles of observations in the input data. If you specify a PARTITION statement then the _ROLE_ variable if present in the input data set is ignored. If you do not use a PARTITION

statement and the input data do not contain a variable named _ROLE_, then all observations in the input data set are assigned to model training.

The following mutually exclusive options are available:

ROLEVAR | ROLE=variable(< TEST='value' > < TRAIN='value' > < VALIDATE='value' >)
names the variable in the input data set whose values are used to assign roles to each observation. The formatted values of this variable that are used to assign observations roles are specified in the TEST=, TRAIN=, and VALIDATE= suboptions. If you do not specify the TRAIN= suboption, then all observations whose role is not determined by the TEST= or VALIDATE= suboptions are assigned to training. If you specify a TESTDATA= data set in the PROC GLMSELECT statement, then you cannot also specify the TEST= suboption in the PARTITION statement. If you specify a VALDATA= data set in the PROC GLMSELECT statement, then you cannot also specify the VALIDATE= suboption in the PARTITION statement.

FRACTION(< TEST=fraction > < VALIDATE=fraction >)
requests that specified proportions of the observations in the input data set be randomly assigned training and validation roles. You specify the proportions for testing and validation by using the TEST= and VALIDATE= suboptions. If you specify both the TEST= and the VALIDATE= suboptions, then the sum of the specified fractions must be less than one and the remaining fraction of the observations are assigned to the training role. If you specify a TESTDATA= data set in the PROC GLMSELECT statement, then you cannot also specify the TEST= suboption in the PARTITION statement. If you specify a VALDATA= data set in the PROC GLMSELECT statement, then you cannot also specify the VALIDATE= suboption in the PARTITION statement.

PERFORMANCE Statement

PERFORMANCE < *options* > **;**

The PERFORMANCE statement is used to change default options that affect the performance of PROC GLMSELECT and to request tables that show the performance options in effect and timing details.

The following options are available:

CPUCOUNT= *1-1024*

CPUCOUNT=*ACTUAL*
specifies the number of processors that PROC GLMSELECT assumes are available for multithreaded BY-group processing. CPUCOUNT=ACTUAL sets CPUCOUNT to be the number of physical processors available. Note that this can be less than the physical number of CPUs if the SAS process has been restricted by system administration tools. Setting CPUCOUNT= to a number greater than the actual number of available CPUs might result in reduced performance. This option overrides the SAS system option CPUCOUNT=. If CPUCOUNT=1, then NOTHREADS is in effect, or if no BY processing is being used, then PROC GLMSELECT uses singly threaded code.

DETAILS

requests the "PerfSettings" table that shows the performance settings in effect and the "Timing" table that provides a broad timing breakdown of the PROC GLMSELECT step.

BUILDSSCP=FULL

BUILDSSCP=INCREMENTAL

specifies whether the SSCP matrix is built incrementally as the selection process progresses or whether the SCCP matrix for the full model is built at the outset. Building the SSCP matrix incrementally can significantly reduce the memory required and the time taken to perform model selection in cases where the number of parameters in the selected model is much smaller than the number of parameters in the full model, but it can hurt performance in other cases since it requires at least one pass through the model training data at each step. If you use backward selection or no selection, or if the BIC or CP statistics are required in the selection process, then the BUILDSSCP=INCREMENTAL option is ignored. In other cases, BUILDSSCP=INCREMENTAL is used by default if the number of effects is greater than 100. See the section "Building the SSCP Matrix" on page 2737 for further details.

THREADS

enables multithreaded BY-group computation. This option overrides the SAS system option THREADS | NOTHREADS. If no BY processing is being used, then PROC GLMSELECT ignores this option and uses singly threaded code.

NOTHREADS

disables multithreaded BY-group computation. This option overrides the SAS system option THREADS | NOTHREADS.

SCORE Statement

SCORE < *DATA=SAS-data-set* > < *OUT=SAS-data-set* > < *keyword* <=*name*> > . . . < *keyword* <=*name*> > ;

The SCORE statement creates a new SAS data set containing predicted values and optionally residuals for data in a new data set that you name. If you do not specify a DATA= data set, then the input data are scored. If you have multiple data sets to predict, you can specify multiple SCORE statements. If you want to create a permanent SAS data set, you must specify a two-level name (for example, *libref.data-set-name*) in the OUT= option. For more information on permanent SAS data sets, refer to the section "SAS Files" in *SAS Language Reference: Concepts*.

When a BY statement is used, the score data set must either contain all the BY variables sorted in the order of the BY variables or contain none of the BY variables. If the score data set contains all of the BY variables, then the model selected for a given BY group is used to score just the matching observations in the score data set. If the score data set contains none of the BY variables, then the entire score data set is scored for each BY group.

All observations in the score data set are retained in the output data set. However, only those observations that contain nonmissing values for all the continuous regressors in the selected model and whose levels of the class variables appearing in effects of the selected model are represented in

the corresponding class variable in the procedure's input data set are scored. All the variables in the input data set are included in the output data set, along with variables containing predicted values and optionally residuals.

Details on the specifications in the SCORE statement follow:

DATA=*SAS data set*
names the data set to be scored. If you omit this option, then the input data set named in the DATA= option in the PROC GLMSELECT statement is scored.

keyword < =*name* >
specifies the statistics to include in the output data set and optionally names the new variables that contain the statistics. Specify a keyword for each desired statistic (see the following list of keywords), followed optionally by an equal sign, and a variable to contain the statistic.

If you specify *keyword*=*name*, the new variable that contains the requested statistic has the specified name. If you omit the optional =*name* after a *keyword*, then the new variable name is formed by using a prefix of one or more characters that identify the statistic, followed by an underscore (_), followed by the dependent variable name.

The keywords allowed and the statistics they represent are as follows:

PREDICTED | PRED | P predicted values. The prefix for the default name is *p*.

RESIDUAL | RESID | R residual, calculated as ACTUAL − PREDICTED. The prefix for the default name is *r*.

OUT=*SAS data set*
gives the name of the new output data set. By default, the procedure uses the DATA*n* convention to name the new data set.

WEIGHT Statement

WEIGHT *variable* **;**

A WEIGHT statement names a variable in the input data set with values that are relative weights for a weighted least squares fit. If the weight value is proportional to the reciprocal of the variance for each observation, then the weighted estimates are the best linear unbiased estimates (BLUE).

Values of the weight variable must be nonnegative. If an observation's weight is zero, the observation is deleted from the analysis. If a weight is negative or missing, it is set to zero, and the observation is excluded from the analysis. A more complete description of the WEIGHT statement can be found in Chapter 39, "The GLM Procedure."

Details: GLMSELECT Procedure

Model-Selection Methods

The model selection methods implemented in PROC GLMSELECT are specified with the SELECTION= option in the MODEL statement.

Full Model Fitted (NONE)

The complete model specified in the MODEL statement is used to fit the model and no effect selection is done. You request this by specifying SELECTION=NONE in the MODEL statement.

Forward Selection (FORWARD)

The forward selection technique begins with just the intercept and then sequentially adds the effect that most improves the fit. The process terminates when no significant improvement can be obtained by adding any effect.

In the traditional implementation of forward selection, the statistic used to gauge improvement in fit is an F statistic that reflects an effect's contribution to the model if it is included. At each step, the effect that yields the most significant F statistic is added. Note that because effects can contribute different degrees of freedom to the model, it is necessary to compare the p-values corresponding to these F statistics.

More precisely, if the current model has p parameters excluding the intercept, and if you denote its residual sum of squares by RSS_p and you add an effect with k degrees of freedom and denote the residual sum of squares of the resulting model by RSS_{p+k}, then the F statistic for entry with k numerator degrees of freedom and $n-(p+k)-1$ denominator degrees of freedom is given by

$$F = \frac{(RSS_p - RSS_{p+k})/k}{RSS_{p+k}/(n-(p+k)-1)}$$

where n is number of observations used in the analysis.

The process stops when the significance level for adding any effect is greater than some specified entry significance level. A well-known problem with this methodology is that these F statistics do not follow an F distribution (Draper, Guttman, and Kanemasu 1971). Hence these p-values cannot reliably be interpreted as probabilities. Various ways to approximate this distribution are described by Miller (2002). Another issue when you use significance levels of entering effects as a stopping criterion arises because the entry significance level is an a priori specification that does not depend on the data. Thus, the same entry significance level can result in overfitting for some data and underfitting for other data.

One approach to address the critical problem of when to stop the selection process is to assess the quality of the models produced by the forward selection method and choose the model from this sequence that "best" balances goodness of fit against model complexity. PROC GLMSELECT supports several criteria that you can use for this purpose. These criteria fall into two groups—information criteria and criteria based on out-of-sample prediction performance.

You use the CHOOSE= option of forward selection to specify the criterion for selecting one model from the sequence of models produced. If you do not specify a CHOOSE= criterion, then the model at the final step is the selected model.

For example, if you specify

```
selection=forward(select=SL choose=AIC SLE=0.2)
```

then forward selection terminates at the step where no effect can be added at the 0.2 significance level. However, the selected model is the first one with the minimal value of the Akaike information criterion. Note that in some cases this minimal value might occur at a step much earlier that the final step, while in other cases the AIC criterion might start increasing only if more steps are done (that is, a larger value of SLE is used). If what you are interested in is minimizing AIC, then too many steps are done in the former case and too few in the latter case. To address this issue, PROC GLMSELECT enables you to specify a stopping criterion with the STOP= option. With a stopping criterion specified, forward selection continues until a local extremum of the stopping criterion in the sequence of models generated is reached. You can also specify STOP= number, which causes forward selection to continue until there are the specified number of effects in the model.

For example, if you specify

```
selection=forward(select=SL stop=AIC)
```

then forward selection terminates at the step where the effect to be added at the next step would produce a model with an AIC statistic larger than the AIC statistic of the current model. Note that in most cases, provided that the entry significance level is large enough that the local extremum of the named criterion occurs before the final step, specifying

```
selection=forward(select=SL choose=CRITERION)
```

or

```
selection=forward(select=SL stop=CRITERION)
```

selects the same model, but more steps are done in the former case. In some cases there might be a better local extremum that cannot be reached if you specify the STOP= option but can be found if you use the CHOOSE= option. Also, you can use the CHOOSE= option in preference to the STOP= option if you want examine how the named criterion behaves as you move beyond the step where the first local minimum of this criterion occurs.

Note that you can specify both the CHOOSE= and STOP= options. You might want to consider models generated by forward selection that have at most some fixed number of effects but select from within this set based on a criterion you specify. For example, specifying

```
selection=forward(stop=20 choose=ADJRSQ)
```

requests that forward selection continue until there are 20 effects in the final model and chooses among the sequence of models the one that has the largest value of the adjusted R-square statistic.

You can also combine these options to select a model where one of two conditions is met. For example,

```
selection=forward(stop=AICC choose=PRESS)
```

chooses whatever occurs first between a local minimum of the predicted residual sum of squares (PRESS) and a local minimum of the corrected Akaike information criterion (AICC).

It is important to keep in mind that forward selection bases the decision about what effect to add at any step by considering models that differ by one effect from the current model. This search paradigm cannot guarantee reaching a "best" subset model. Furthermore, the add decision is greedy in the sense that the effect deemed most significant is the effect that is added. However, if your goal is to find a model that is best in terms of some selection criterion other than the significance level of the entering effect, then even this one step choice might not be optimal. For example, the effect you would add to get a model with the smallest value of the PRESS statistic at the next step is not necessarily the same effect that has the most significant entry F statistic. PROC GLMSELECT enables you to specify the criterion to optimize at each step by using the SELECT= option. For example,

```
selection=forward(select=CP)
```

requests that at each step the effect that is added be the one that gives a model with the smallest value of the Mallows' $C(p)$ statistic. Note that in the case where all effects are variables (that is, effects with one degree of freedom and no hierarchy), using ADJRSQ, AIC, AICC, BIC, CP, RSQUARE, or SBC as the selection criterion for forward selection produces the same sequence of additions. However, if the degrees of freedom contributed by different effects are not constant, or if an out-of-sample prediction-based criterion is used, then different sequences of additions might be obtained.

You can use SELECT= together with CHOOSE= and STOP=. If you specify only the SELECT= criterion, then this criterion is also used as the stopping criterion. In the previous example where only the selection criterion is specified, not only do effects enter based on the Mallows' $C(p)$ statistic, but the selection terminates when the $C(p)$ statistic first increases.

You can find discussion and references to studies about criteria for variable selection in Burnham and Anderson (2002), along with some cautions and recommendations.

Examples of Forward Selection Specifications

```
selection=forward
```

adds effects that at each step give the lowest value of the SBC statistic and stops at the step where adding any effect would increase the SBC statistic.

```
selection=forward(select=SL)
```

adds effects based on significance level and stops when all candidate effects for entry at a step have a significance level greater than the default entry significance level of 0.15.

```
selection=forward(select=SL stop=validation)
```

adds effects based on significance level and stops at a step where adding any effect increases the error sum of squares computed on the validation data.

```
selection=forward(select=AIC)
```

adds effects that at each step give the lowest value of the AIC statistic and stops at the step where adding any effect would increase the AIC statistic.

```
selection=forward(select=ADJRSQ stop=SL SLE=0.2)
```

adds effects that at each step give the largest value of the adjusted R-square statistic and stops at the step where the significance level corresponding to the addition of this effect is greater than 0.2.

Backward Elimination (BACKWARD)

The backward elimination technique starts from the full model including all independent effects. Then effects are deleted one by one until a stopping condition is satisfied. At each step, the effect showing the smallest contribution to the model is deleted. In traditional implementations of backward elimination, the contribution of an effect to the model is assessed by using an F statistic. At any step, the predictor producing the least significant F statistic is dropped and the process continues until all effects remaining in the model have F statistics significant at a stay significance level (SLS).

More precisely, if the current model has p parameters excluding the intercept, and if you denote its residual sum of squares by RSS_p and you drop an effect with k degrees of freedom and denote the residual sum of squares of the resulting model by RSS_{p-k}, then the F statistic for removal with k numerator degrees of freedom and $n - p - k$ denominator degrees of freedom is given by

$$F = \frac{(RSS_{p-k} - RSS_p)/k}{RSS_p/(n-p-k)}$$

where n is number of observations used in the analysis.

Just as with forward selection, you can change the criterion used to assess effect contributions with the SELECT= option. You can also specify a stopping criterion with the STOP= option and use a CHOOSE= option to provide a criterion used to select among the sequence of models produced. See the discussion in the section "Forward Selection (FORWARD)" on page 2719 for additional details.

Examples of Backward Selection Specifications

```
selection=backward
```

removes effects that at each step produce the largest value of the Schwarz Bayesian information criterion (SBC) statistic and stops at the step where removing any effect increases the SBC statistic.

```
selection=backward(stop=press)
```

removes effects based on the SBC statistic and stops at the step where removing any effect increases the predicted residual sum of squares (PRESS).

```
selection=backward(select=SL)
```

removes effects based on significance level and stops when all candidate effects for removal at a step have a significance level less than the default stay significance level of 0.15.

```
selection=backward(select=SL choose=validate SLS=0.1)
```

removes effects based on significance level and stops when all effects in the model are significant at the 0.1 level. Finally, from the sequence of models generated, choose the one that gives the smallest average square error when scored on the validation data.

Stepwise Selection(STEPWISE)

The stepwise method is a modification of the forward selection technique that differs in that effects already in the model do not necessarily stay there.

In the traditional implementation of stepwise selection method, the same entry and removal F statistics for the forward selection and backward elimination methods are used to assess contributions of effects as they are added to or removed from a model. If at a step of the stepwise method, any effect in the model is not significant at the SLSTAY= level, then the least significant of these effects is removed from the model and the algorithm proceeds to the next step. This ensures that no effect can be added to a model while some effect currently in the model is not deemed significant. Only after all necessary deletions have been accomplished can another effect be added to the model. In this case the effect whose addition yields the most significant F value is added to the model and the algorithm proceeds to the next step. The stepwise process ends when none of the effects outside the model has an F statistic significant at the SLENTRY= level and every effect in the model is significant at the SLSTAY= level. In some cases, neither of these two conditions for stopping is met and the sequence of models cycles. In this case, the stepwise method terminates at the end of the second cycle.

Just as with forward selection and backward elimination, you can change the criterion used to assess effect contributions, with the SELECT= option. You can also specify a stopping criterion with the STOP= option and use a CHOOSE= option to provide a criterion used to select among the sequence of models produced. See the discussion in the section "Forward Selection (FORWARD)" on page 2719 for additional details.

For selection criteria other than significance level, PROC GLMSELECT optionally supports a further modification in the stepwise method. In the standard stepwise method, no effect can enter the model if removing any effect currently in the model would yield an improved value of the selection criterion. In the modification, you can use the DROP=COMPETITIVE option to specify that addition and deletion of effects should be treated competitively. The selection criterion is evaluated for all models obtained by deleting an effect from the current model or by adding an effect to this model. The action that most improves the selection criterion is the action taken.

Examples of Stepwise Selection Specifications

```
selection=stepwise
```

requests stepwise selection based on the SBC criterion. First, if removing any effect yields a model with a lower SBC statistic than the current model, then the effect producing the smallest SBC statistic is removed. When removing any effect increases the SBC statistic, then provided that adding some effect lowers the SBC statistic, the effect producing the model with the lowest SBC is added.

```
selection=stepwise(select=SL)
```

requests the traditional stepwise method. First, if the removal of any effect yields an F statistic that is not significant at the default stay level of 0.15, then the effect whose removal produces the least significant F statistic is removed and the algorithm proceeds to the next step. Otherwise the effect whose addition yields the most significant F statistic is added, provided that it is significant at the default entry level of 0.15.

```
selection=stepwise(select=SL stop=SBC)
```

is the traditional stepwise method, where effects enter and leave based on significance levels, but with the following extra check: If any effect to be added or removed yields a model whose SBC statistic is greater than the SBC statistic of the current model, then the stepwise method terminates at the current model. Note that in this case, the entry and stay significance levels still play a role as they determine whether an effect is deleted from or added to the model. This might result in the selection terminating before a local minimum of the SBC criterion is found.

```
selection=stepwise(select=SL SLE=0.1 SLS=0.08 choose=AIC)
```

selects effects to enter or drop as in the previous example except that the significance level for entry is now 0.1 and the significance level to stay is 0.08. From the sequence of models produced, the selected model is chosen to yield the minimum AIC statistic.

```
selection=stepwise(select=AICC drop=COMPETITIVE)
```

requests stepwise selection based on the AICC criterion with steps treated competitively. At any step, evaluate the AICC statistics corresponding to the removal of any effect in the current model or the addition of any effect to the current model. Choose the addition or removal that produced this minimum value, provided that this minimum is lower than the AICC statistic of the current model.

```
selection=stepwise(select=SBC drop=COMPETITIVE stop=VALIDATE)
```

requests stepwise selection based on the SBC criterion with steps treated competitively and where stopping is based on the average square error over the validation data. At any step, SBC statistics corresponding to the removal of any effect from the current model or the addition of any effect to the current model are evaluated. The addition or removal that produces the minimum SBC value is made. The average square error on the validation data for the model with this addition or removal is evaluated. If this average square error is greater than the average square error on the validation data prior to this addition or deletion, then the algorithm terminates at this prior model.

Least Angle Regression (LAR)

Least angle regression was introduced by Efron et al. (2004). Not only does this algorithm provide a selection method in its own right, but with one additional modification it can be used to efficiently produce LASSO solutions. Just like the forward selection method, the LAR algorithm produces a sequence of regression models where one parameter is added at each step, terminating at the full least squares solution when all parameters have entered the model.

The algorithm starts by centering the covariates and response, and scaling the covariates so that they all have the same corrected sum of squares. Initially all coefficients are zero, as is the predicted response. The predictor that is most correlated with the current residual is determined and a step is taken in the direction of this predictor. The length of this step determines the coefficient of this predictor and is chosen so that some other predictor and the current predicted response have the same correlation with the current residual. At this point, the predicted response moves in the direction that is equiangular between these two predictors. Moving in this direction ensures that these two predictors continue to have a common correlation with the current residual. The predicted response moves in this direction until a third predictor has the same correlation with the current residual as the two predictors already in the model. A new direction is determined that is equiangular between these three predictors and the predicted response moves in this direction until a fourth predictor joins the set having the same correlation with the current residual. This process continues until all predictors are in the model.

As with other selection methods, the issue of when to stop the selection process is crucial. You can specify a criterion to use to choose among the models at each step with the CHOOSE= option. You can also specify a stopping criterion with the STOP= option. See the section "Criteria Used in Model Selection Methods" on page 2727 for details and Table 42.6 for the formulas for evaluating these criteria. These formulas use the approximation that at step k of the LAR algorithm, the model has k degrees of freedom. See Efron et al. (2004) for a detailed discussion of this so-called simple approximation.

A modification of LAR selection suggested in Efron et al. (2004) uses the LAR algorithm to select the set of covariates in the model at any step, but uses ordinary least squares regression with just these covariates to obtain the regression coefficients. You can request this hybrid method by specifying the LSCOEFFS suboption of SELECTION=LAR.

Lasso Selection (LASSO)

LASSO (least absolute shrinkage and selection operator) selection arises from a constrained form of ordinary least squares regression where the sum of the absolute values of the regression coefficients is constrained to be smaller than a specified parameter. More precisely let $X = (x_1, x_2, \ldots, x_m)$ denote the matrix of covariates and let y denote the response, where the x_is have been centered and scaled to have unit standard deviation and mean zero, and y has mean zero. Then for a given parameter t, the LASSO regression coefficients $\beta = (\beta_1, \beta_2, \ldots, \beta_m)$ are the solution to the constrained optimization problem

$$\text{minimize} ||y - X\beta||^2 \qquad \text{subject to} \quad \sum_{j=1}^{m} |\beta_j| \leq t$$

Provided that the LASSO parameter t is small enough, some of the regression coefficients will be exactly zero. Hence, you can view the LASSO as selecting a subset of the regression coefficients for each LASSO parameter. By increasing the LASSO parameter in discrete steps, you obtain a sequence of regression coefficients where the nonzero coefficients at each step correspond to selected parameters.

Early implementations (Tibshirani 1996) of LASSO selection used quadratic programming techniques to solve the constrained least squares problem for each LASSO parameter of interest. Later Osborne, Presnell, and Turlach (2000) developed a "homotopy method" that generates the LASSO solutions for all values of t. Efron et al. (2004) derived a variant of their algorithm for least angle regression that can be used to obtain a sequence of LASSO solutions from which all other LASSO solutions can be obtained by linear interpolation. This algorithm for SELECTION=LASSO is used in PROC GLMSELECT. It can be viewed as a stepwise procedure with a single addition to or deletion from the set of nonzero regression coefficients at any step.

As with the other selection methods supported by PROC GLMSELECT, you can specify a criterion to choose among the models at each step of the LASSO algorithm with the CHOOSE= option. You can also specify a stopping criterion with the STOP= option. See the discussion in the section "Forward Selection (FORWARD)" on page 2719 for additional details. The model degrees of freedom PROC GLMSELECT uses at any step of the LASSO are simply the number of nonzero regression coefficients in the model at that step. Efron et al. (2004) cite empirical evidence for doing this but do not give any mathematical justification for this choice.

A modification of LASSO selection suggested in Efron et al. (2004) uses the LASSO algorithm to select the set of covariates in the model at any step, but uses ordinary least squares regression with just these covariates to obtain the regression coefficients. You can request this hybrid method by specifying the LSCOEFFS suboption of SELECTION=LASSO.

Model Selection Issues

Many authors caution against the use of "automatic variable selection" methods and describe pitfalls that plague many such methods. For example, Harrell (2001) states that "stepwise variable selection has been a very popular technique for many years, but if this procedure had just been proposed as a statistical method, it would most likely be rejected because it violates every principle of statistical estimation and hypothesis testing." He lists and discusses several of these issues and cites a variety of studies that highlight these problems. He also notes that many of these issues are not restricted to stepwise selection, but affect forward selection and backward elimination, as well as methods based on all-subset selection.

In their introductory chapter, Burnham and Anderson (2002) discuss many issues involved in model selection. They also strongly warn against "data dredging," which they describe as "the process of analyzing data with few or no a priori questions, by subjectively and iteratively searching the data for patterns and 'significance'." However, Burnham and Anderson also discuss the desirability of finding parsimonious models. They note that using "full models" that contain many insignificant predictors might avoid some of the inferential problems arising in models with automatically selected variables but will lead to overfitting the particular sample data and produce a model that performs poorly in predicting data not used in training the model.

One problem in the traditional implementations of forward, backward, and stepwise selection methods is that they are based on sequential testing with specified entry (SLE) and stay (SLS) significance levels. However, it is known that the "F-to-enter" and "F-to-delete" statistics do not follow an F distribution (Draper, Guttman, and Kanemasu 1971). Hence the SLE and SLS values cannot reliably be viewed as probabilities. One way to address this difficulty is to replace hypothesis testing as a means of selecting a model with information criteria or out-of-sample prediction criteria. While Harrell (2001) points out that information criteria were developed for comparing only prespecified models, Burnham and Anderson (2002) note that AIC criteria have routinely been used for several decades for performing model selection in time series analysis.

Problems also arise when the selected model is interpreted as if it were prespecified. There is a "selection bias" in the parameter estimates that is discussed in detail in Miller (2002). This bias occurs because a parameter is more likely to be selected if it is above its expected value than if it is below its expected value. Furthermore, because multiple comparisons are made in obtaining the selected model, the p-values obtained for the selected model are not valid. When a single best model is selected, inference is conditional on that model. Model averaging approaches provide a way to make more stable inferences based on a set of models. Methods for doing this are presented in Burnham and Anderson (2002), and Bayesian approaches are discussed in Raftery, Madigan, and Hoeting (1997).

Despite these difficulties, careful and informed use of variable selection methods still has its place in modern data analysis. For example, Foster and Stine (2004) use a modified version of stepwise selection to build a predictive model for bankruptcy from over 67,000 possible predictors and show that this yields a model whose predictions compare favorably with other recently developed data mining tools. In particular, when the goal is prediction rather than estimation or hypothesis testing, variable selection with careful use of validation to limit both under and over fitting is often a useful starting point of model development.

Criteria Used in Model Selection Methods

PROC GLMSELECT supports a variety of fit statistics that you can specify as criteria for the CHOOSE=, SELECT=, and STOP= options in the MODEL statement. The following statistics are available:

ADJRSQ	adjusted R-square statistic (Darlington 1968; Judge et al. 1985)
AIC	Akaike information criterion (Darlington 1968; Judge et al. 1985)
AICC	corrected Akaike information criterion (Hurvich and Tsai 1989)
BIC	Sawa Bayesian information criterion (Sawa 1978; Judge et al. 1985)
CP	Mallows C_p statistic (Mallows 1973; Hocking 1976)
PRESS	predicted residual sum of squares statistic
SBC	Schwarz Bayesian information criterion (Schwarz 1978; Judge et al. 1985)
SL	significance level of the F statistic used to assess an effect's contribution to the fit when it is added to or removed from a model
VALIDATE	average square error over the validation data

Table 42.6 provides formulas and definitions for the fit statistics.

Table 42.6 Formulas and Definitions for Model Fit Summary Statistics

Statistic	Definition or Formula
n	number of observations
p	number of parameters including the intercept
$\hat{\sigma}^2$	estimate of pure error variance from fitting the full model
SST	total sum of squares corrected for the mean for the dependent variable
SSE	error sum of squares
ASE	$\frac{\text{SSE}}{n}$
MSE	$\frac{\text{SSE}}{n-p}$
R^2	$1-\frac{\text{SSE}}{\text{SST}}$
ADJRSQ	$1-\frac{(n-1)(1-R^2)}{n-p}$
AIC	$n\ln\left(\frac{\text{SSE}}{n}\right)+2p$
AICC	$1+\ln\left(\frac{\text{SSE}}{n}\right)+\frac{2(p+1)}{n-p-2}$
BIC	$n\ln\left(\frac{\text{SSE}}{n}\right)+2(p+2)q-2q^2$ where $q=\frac{n\hat{\sigma}^2}{\text{SSE}}$
CP (C_p)	$\frac{\text{SSE}}{\hat{\sigma}^2}+2p-n$
PRESS	$\sum_{i=1}^{n}\frac{r_i^2}{(1-h_i)^2}$ where r_i = residual at observation i and h_i = leverage of observation i = $\mathbf{x}_i(\mathbf{X}'\mathbf{X})^{-}\mathbf{x}_i'$
RMSE	$\sqrt{\text{MSE}}$
SBC	$n\ln\left(\frac{\text{SSE}}{n}\right)+p\ln(n)$

Changes in Formulas for AIC and AICC

The formulas used for the AIC and AICC statistics have been changed in SAS 9.2. However, the models selected at each step of the selection process and the final selected model are unchanged from the experimental download release of PROC GLMSELECT, even in the case where you specify AIC or AICC in the SELECT=, CHOOSE=, and STOP= options in the MODEL statement. The reason for making this change is to make the connection between the AIC statistic and the AICC statistic more transparent.

In the context of linear regression, several different versions of the formulas for AIC and AICC appear in the statistics literature. However, for a fixed number of observations, these different

versions differ by additive and positive multiplicative constants. Because the model selected to yield a minimum of a criterion is not affected if the criterion is changed by additive and positive multiplicative constants, these changes in the formula for AIC and AICC do not affect the selection process.

The following section provides details about these changes. Formulas used in the experimental download release are denoted with a superscript of (d) and n, p and SSE are defined in Table 42.6.

In the experimental download release of PROC GLMSELECT the following formulas are used for AIC (Darlington 1968; Judge et al. 1985) and AICC (Hurvich, Simonoff, and Tsai 1998):

$$\text{AIC}^{(d)} = n\log\left(\frac{\text{SSE}}{n}\right) + 2p$$

and

$$\text{AICC}^{(d)} = \log\left(\frac{\text{SSE}}{n}\right) + 1 + \frac{2(p+1)}{n-p-2}$$

The definitions of AIC and AICC used in this release are found in Hurvich and Tsai (1989). These formulas are

$$\text{AIC} = n\log\left(\frac{\text{SSE}}{n}\right) + 2p + n + 2$$

and

$$\text{AICC} = \text{AIC} + \frac{2(p+1)(p+2)}{n-p-2}$$

Hurvich and Tsai (1989) show that the formula for AICC can also be written as

$$\text{AICC} = n\log\left(\frac{\text{SSE}}{n}\right) + \frac{n(n+p)}{n-p-2}$$

The relationships between the alternative forms of the formulas are

$$\text{AIC} = \text{AIC}^{(d)} + n + 2$$

$$\text{AICC} = n\ \text{AICC}^{(d)}$$

CLASS Variable Parameterization

Consider a model with one classification variable A with four levels, 1, 2, 5, and 7. Details of the possible choices for the PARAM= option follow.

EFFECT — Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of −1. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

Effect Coding			
	Design Matrix		
A	A1	A2	A5
1	1	0	0
2	0	1	0
5	0	0	1
7	−1	−1	−1

Parameter estimates of classification main effects that use the effect coding scheme estimate the difference in the effect of each nonreference level compared to the average effect over all four levels.

GLM — As in PROC GLM, four columns are created to indicate group membership. The design matrix columns for A are as follows.

GLM Coding				
	Design Matrix			
A	A1	A2	A5	A7
1	1	0	0	0
2	0	1	0	0
5	0	0	1	0
7	0	0	0	1

Parameter estimates of classification main effects that use the GLM coding scheme estimate the difference in the effects of each level compared to the last level.

ORDINAL

THERMOMETER Three columns are created to indicate group membership of the higher levels of the effect. For the first level of the effect (which for A is 1), all three dummy variables have a value of 0. The design matrix columns for A are as follows.

Ordinal Coding			
	Design Matrix		
A	A2	A5	A7
1	0	0	0
2	1	0	0
5	1	1	0
7	1	1	1

The first level of the effect is a control or baseline level. Parameter estimates of classification main effects that use the ORDINAL coding scheme estimate the effect on the response as the ordinal factor is set to each succeeding level. When the parameters for an ordinal main effect have the same sign, the response effect is monotonic across the levels.

POLYNOMIAL

POLY Three columns are created. The first represents the linear term (x), the second represents the quadratic term (x^2), and the third represents the cubic term (x^3), where x is the level value. If the classification levels are not numeric, they are translated into 1, 2, 3, ... according to their sorting order. The design matrix columns for A are as follows.

Polynomial Coding			
	Design Matrix		
A	APOLY1	APOLY2	APOLY3
1	1	1	1
2	2	4	8
5	5	25	125
7	7	49	343

REFERENCE

REF Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of 0. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

Reference Coding			
	Design Matrix		
A	A1	A2	A5
1	1	0	0
2	0	1	0
5	0	0	1
7	0	0	0

Parameter estimates of CLASS main effects that use the reference coding scheme estimate the difference in the effect of each nonreference level compared to the effect of the reference level.

ORTHEFFECT The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=EFFECT. The design matrix columns for A are as follows.

Orthogonal Effect Coding			
	Design Matrix		
A	AOEFF1	AOEFF2	AOEFF3
1	1.41421	−0.81650	−0.57735
2	0.00000	1.63299	−0.57735
5	0.00000	0.00000	1.73205
7	−1.41421	−0.81649	−0.57735

ORTHORDINAL

ORTHOTHERM The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=ORDINAL. The design matrix columns for A are as follows.

Orthogonal Ordinal Coding			
	Design Matrix		
A	AOORD1	AOORD2	AOORD3
1	−1.73205	0.00000	0.00000
2	0.57735	−1.63299	0.00000
5	0.57735	0.81650	−1.41421
7	0.57735	0.81650	1.41421

ORTHPOLY The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=POLY. The design matrix columns for A are as follows.

Orthogonal Polynomial Coding			
	Design Matrix		
A	AOPOLY1	AOPOLY2	AOPOLY5
1	−1.153	0.907	−0.921
2	−0.734	−0.540	1.473
5	0.524	−1.370	−0.921
7	1.363	1.004	0.368

ORTHREF The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=REFERENCE. The design matrix columns for A are as follows.

Orthogonal Reference Coding			
	Design Matrix		
A	AOREF1	AOREF2	AOREF3
1	1.73205	0.00000	0.00000
2	−0.57735	1.63299	0.00000
5	−0.57735	−0.81650	1.41421
7	−0.57735	−0.81650	−1.41421

The following example illustrates several features of the CLASS statement.

```
data codingExample;
   drop i;
   do i=1 to 1000;
     c1 = 1 + mod(i,6);
     if      i < 50  then c2 = 'very low ';
     else if i < 250 then c2 = 'low';
     else if i < 500 then c2 = 'medium';
     else if i < 800 then c2 = 'high';
     else                 c2 = 'very high';
     x1 = ranuni(1);
     x2 = ranuni(1);
     y = x1 + 10*(c1=3) +5*(c1=5) +rannor(1);
     output;
   end;
run;
proc glmselect data=codingExample;
   class c1(param=ref split) c2(param=ordinal order=data) /
          delimiter = ',' showcoding;
   model y = c1 c2 x1 x2/orderselect;
run;
```

Figure 42.11 Class Level Information

The GLMSELECT Procedure

Class Level Information

Class	Levels	Values
c1	6 *	1,2,3,4,5,6
c2	5	very low,low,medium,high,very high

* Associated Parameters Split

The "Class Level Information" table shown in Figure 42.11 is produced by default whenever you specify a CLASS statement. Note that because the levels of the variable "c2" contain embedded blanks, the DELIMITER="," option has been specified. The SHOWCODING option requests the display of the "Class Level Coding" table shown in Figure 42.12. An ordinal parameterization is used for "c2" because its levels have a natural order. Furthermore, because these levels appear in their natural order in the data, you can preserve this order by specifying the ORDER=*data* option.

Figure 42.12 Class Level Coding

Class Level Coding

c1	Design Variables				
Level	1	2	3	4	5
1	1	0	0	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	0	0	1	0
5	0	0	0	0	1
6	0	0	0	0	0

The SPLIT option has been specified for the classification variable "c1." This permits the parameters associated with the effect "c1" to enter or leave the model individually. The "Parameter Estimates" table in Figure 42.13 shows that for this example the parameters corresponding to only levels 3 and 5 of "c1" are in the selected model. Finally, note that the ORDERSELECT option in the MODEL statement specifies that the parameters are displayed in the order in which they first entered the model.

Figure 42.13 Parameter Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value
Intercept	1	−0.216680	0.068650	−3.16
c1_3	1	10.160900	0.087898	115.60
c1_5	1	5.018015	0.087885	57.10
x1	1	1.315468	0.109772	11.98

Macro Variables Containing Selected Models

Often you might want to perform post-selection analysis by using other SAS procedures. To facilitate this, PROC GLMSELECT saves the list of selected effects in a macro variable. This list does not explicitly include the intercept so that you can use it in the MODEL statement of other SAS/STAT regression procedures.

The following table describes the macro variables that PROC GLMSELECT creates. Note that when BY processing is used, one macro variable, indexed by the BY group number, is created for each BY group.

Macro Variable	Description
No BY processing	
_GLSIND1	Selected model
BY processing	
_GLSNUMBYS	Number of BY groups
_GLSIND1	Selected model for BY group 1
_GLSIND2	Selected model for BY group 2
...	

You can use the macro variable _GLSIND as a synonym for _GLSIND1. If you do not use BY processing, _GLSNUMBYS is still defined and has the value 1.

To aid in associating indexed macro variables with the appropriate observations when BY processing is used, PROC GLMSELECT creates a variable _BY_ in the output data set specified in an OUTPUT statement (see the section "OUTPUT Statement" on page 2714) that tags observations with an index that matches the index of the appropriate macro variable.

The following statements create a data set with two BY groups and run PROC GLMSELECT to select a model for each BY group.

```
data one(drop=i j);
  array x{5} x1-x5;
  do i=1 to 1000;
    classVar = mod(i,4)+1;
    do j=1 to 5;
      x{j} = ranuni(1);
    end;
    if i<400 then do;
         byVar = 'group 1';
         y     = 3*classVar+7*x2+5*x2*x5+rannor(1);
       end;
    else do;
         byVar = 'group 2';
         y     = 2*classVar+x5+rannor(1);
       end;
    output;
  end;
run;

proc glmselect data=one;
  by     byVar;
  class  classVar;
  model  y = classVar x1|x2|x3|x4|x5 @2 /
                selection=stepwise(stop=aicc);
  output out=glmselectOutput;
run;
```

The preceding PROC GLMSELECT step produces three macro variables:

Macro Variable	Value	Description
_GLSNUMBYS	2	Number of BY groups
_GLSIND1	classVar x2 x2*x5	Selected model for the first BY group
_GLSIND2	classVar x5	Selected model for the second BY group

You can now leverage these macro variables and the output data set created by PROC GLMSELECT to perform post-selection analyses that match the selected models with the appropriate BY-group observations. For example, the following statements create and run a macro that uses PROC GLM to perform LSMeans analyses.

```
%macro LSMeansAnalysis;
   %do i=1 %to &_GLSNUMBYS;
      title1  "Analysis Using the Selected Model for BY group number &i";
      title2 "Selected Effects: &&_GLSIND&i";

      ods select LSMeans;
      proc glm data=glmselectOutput(where = (_BY_ = &i));
         class classVar;
         model y = &&_GLSIND&i;
         lsmeans classVar;
      run;quit;
   %end;
%mend;
%LSMeansAnalysis;
```

The LSMeans analysis output from PROC GLM is shown in Output 42.14.

Figure 42.14 LS-Means Analyses for Selected Models

```
Analysis Using the Selected Model for BY group number 1
Selected Effects: classVar x2 x2*x5

The GLM Procedure
Least Squares Means

class
Var          y LSMEAN

1           7.8832052
2          10.9528618
3          13.9412216
4          16.7929355
```

Figure 42.14 *continued*

```
Analysis Using the Selected Model for BY group number 2
Selected Effects: classVar x5

The GLM Procedure
Least Squares Means

class
Var          y LSMEAN

1          2.46805014
2          4.52102826
3          6.53369479
4          8.49354763
```

Building the SSCP Matrix

Traditional implementations of FORWARD and STEPWISE selection methods start by computing the augmented crossproduct matrix for all the specified effects. This initial crossproduct matrix is updated as effects enter or leave the current model by sweeping the columns corresponding to the parameters of the entering or departing effects. Building the starting crossproduct matrix can be done with a single pass through the data and requires $O(m^2)$ storage and $O(nm^2)$ work, where n is the number of observations and m is the number of parameters. If k selection steps are done, then the total work sweeping effects in and out of the model is $O(km^2)$. When $n >> m$, the work required is dominated by the time spent forming the crossproduct matrix. However, when m is large (tens of thousands), just storing the crossproduct matrix becomes intractable even though the number of selected parameters might be small. Note also that when interactions of classification effects are considered, the number of parameters considered can be large, even though the number of effects considered is much smaller.

When the number of selected parameters is smaller than the total number of parameters, it turns out that many of the crossproducts are not needed in the selection process. Let y denote the dependent variable, and suppose at some step of the selection process that X denotes the $n \times p$ design matrix columns corresponding to the currently selected model. Let $Z = Z_1, Z_2, \ldots, Z_{m-p}$ denote the design matrix columns corresponding to the $m - p$ effects not yet in the model. Then in order to compute the reduction in the residual sum of squares when Z_j is added to the model, the only additional crossproducts needed are $Z_j'y$, $Z_j'X$, and $Z_j'Z_j$. Note that it is not necessary to compute any of $Z_j'Z_i$ with $i \neq j$ and if $p << m$, and this yields a substantial saving in both memory required and computational work. Note, however, that this strategy does require a pass through the data at any step where adding an effect to the model is considered.

PROC GLMSELECT supports both of these strategies for building the crossproduct matrix. You can choose which of these strategies to use by specifying the BUILDSSCP=FULL or BUILDSSCP=INCREMENTAL option in the PERFORMANCE statement. If you request BACKWARD selection, then the full SSCP matrix is required. Similarly, if you request the BIC or CP criterion as the SELECT=, CHOOSE=, or STOP= criterion, or if you request the display of one or both of these criteria with the STATS=BIC, STATS=CP, or STATS=ALL option, then the full

model needs to be computed. If you do not specify the BUILDSSCP= option, then PROC GLMSELECT switches to the incremental strategy if the number of effects is greater than one hundred. This default strategy is designed to give good performance when the number of selected parameters is less than about 20% of the total number of parameters. Hence if you choose options that you know will cause the selected model to contain a significantly higher percentage of the total number of candidate parameters, then you should consider specifying BUILDSSCP=FULL. Conversely, if you specify fewer than 100 effects in the MODEL statement but many of these effects have a large number of associated parameters, then specifying BUILSSCP=INCREMENTAL might result in improved performance.

Parallel BY-Group Computation

The BY-group processing in PROC GLMSELECT is multithreaded, enabling parallel processing of BY groups when more than one processor is available. Before effect selection begins, PROC GLMSELECT preprocesses the entire input data set and writes the preprocessed data to one or more utility files. When you specify BY-group processing and there are multiple CPUs available, one utility file is created for each processor and each BY group is assigned to one of these utility files. Once this preprocessing phase is complete, each utility file is read and all the BY groups it contains are processed in a dedicated thread. Finally, once model selection has been completed for each thread, then the results are displayed sequentially from a single thread.

You can control the use of threading in PROC GLMSELECT and other multithreaded procedures by using the SAS system options THREADS | NOTHREADS and CPUCOUNT=. You can override these SAS system options by specifying THREADS | NOTHREADS and CPUCOUNT= in the PERFORMANCE statement. Note that if NOTHREADS is in effect, or CPUCOUNT=1, or no BY processing is used, then PROC GLMSELECT will use singly threaded code. If BY groups are to be processed in multiple threads, then the memory required is proportional to the number of threads used. PROC GLMSELECT tries to predict how much memory is required for model selection and reduces the number of threads it uses if there is insufficient memory for simultaneously processing more threads. You can find out the number of threads actually used by PROC GLMSELECT in the "Performance Settings" table that you request by using the DETAILS option in the PERFORMANCE statement.

The speedup you will obtain processing BY groups in parallel depends on several factors. First, if the time required to perform model selection on each BY group is small, then the multithreading overhead might negate any gains obtained by parallel processing. Another limiting factor is the I/O speed. When PROC GLMSELECT builds crossproduct matrices incrementally (see the section "Building the SSCP Matrix" on page 2737), data in each BY group are read multiple times. If there is insufficient I/O bandwidth, then the parallel processing can stall as CPUs wait on I/O. Optimal results will be obtained if each of the utility files used by PROC GLMSELECT is assigned to its own I/O controller. The locations of utility files can be controlled by using the SAS system option UTILLOC, which must be specified when SAS is invoked. For information about the UTILLOC option and additional information about parallel processing in SAS, see the chapter "Support for Parallel Processing" in *SAS Language Reference: Concepts.*

Using Validation and Test Data

When you have sufficient data, you can subdivide your data into three parts called the training, validation, and test data. During the selection process, models are fit on the training data, and the prediction error for the models so obtained is found by using the validation data. This prediction error on the validation data can be used to decide when to terminate the selection process or to decide what effects to include as the selection process proceeds. Finally, once a selected model has been obtained, the test set can be used to assess how the selected model generalizes on data that played no role in selecting the model.

In some cases you might want to use only training and test data. For example, you might decide to use an information criterion to decide what effects to include and when to terminate the selection process. In this case no validation data are required, but test data can still be useful in assessing the predictive performance of the selected model. In other cases you might decide to use validation data during the selection process but forgo assessing the selected model on test data. Hastie, Tibshirani, and Friedman (2001) note that it is difficult to give a general rule on how many observations you should assign to each role. They note that a typical split might be 50% for training and 25% each for validation and testing.

PROC GLMSELECT provides several methods for partitioning data into training, validation, and test data. You can provide data for each role in separate data sets that you specify with the DATA=, TESTDATA=, and VALDATA= options in the PROC GLMSELECT procedure. An alternative method is to use a PARTITION statement to logically subdivide the DATA= data set into separate roles. You can name the fractions of the data that you want to reserve as test data and validation data. For example, specifying

```
proc glmselect data=inData;
  partition fraction(test=0.25 validate=0.25);
  ...
run;
```

randomly subdivides the “inData” data set, reserving 50% for training and 25% each for validation and testing.

In some cases you might need to exercise more control over the partitioning of the input data set. You can do this by naming a variable in the input data set as well as a formatted value of that variable that correspond to each role. For example, specifying

```
proc glmselect data=inData;
  partition roleVar=group(test='group 1' train='group 2')
  ...
run;
```

assigns all roles observations in the “inData” data set based on the value of the variable named group in that data set. Observations where the value of group is ’group 1’ are assigned for testing, and those with value ’group 2’ are assigned to training. All other observations are ignored.

You can also combine the use of the PARTITION statement with named data sets for specifying data roles. For example,

```
proc glmselect data=inData testData=inTest;
  partition fraction(validate=0.4);
  ...
run;
```

reserves 40% of the "inData" data set for validation and uses the remaining 60% for training. Data for testing is supplied in the "inTest" data set. Note that in this case, because you have supplied a TESTDATA= data set, you cannot reserve additional observations for testing with the PARTITION statement.

When you use a PARTITION statement, the output data set created with an OUTPUT statement contains a character variable _ROLE_ whose values "TRAIN," "TEST," and "VALIDATE" indicate the role of each observation. _ROLE_ is blank for observations that were not assigned to any of these three roles. When the input data set specified in the DATA= option in the PROC GLMSELECT statement contains an _ROLE_ variable and no PARTITION statement is used, and TESTDATA= and VALDATA= are not specified, then the _ROLE_ variable is used to define the roles of each observation. This is useful when you want to rerun PROC GLMSELECT but use the same data partitioning as in a previous PROC GLMSELECT step. For example, the following statements use the same data for testing and training in both PROC GLMSELECT steps:

```
proc glmselect data=inData;
  partition fraction(test=0.5);
  model y=x1-x10/selection=forward;
  output out=outDataForward;
run;

proc glmselect data=outDataForward;
  model y=x1-x10/selection=backward;
run;
```

When you have reserved observations for training, validation, and testing, a model fit on the training data is scored on the validation and test data, and the average squared error, denoted by ASE, is computed separately for each of these subsets. The ASE for each data role is the error sum of squares for observations in that role divided by the number of observations in that role.

Using the Validation ASE as the STOP= Criterion

If you have provided observations for validation, then you can specify STOP=VALIDATE as a suboption of the SELECTION= option in the MODEL statement. At step k of the selection process, the best candidate effect to enter or leave the current model is determined. Note that here "best candidate" means the effect that gives the best value of the SELECT= criterion that need not be based on the validation data. The validation ASE for the model with this candidate effect added is computed. If this validation ASE is greater than the validation ASE for the model at step k, then the selection process terminates at step k.

Using the Validation ASE as the CHOOSE= Criterion

When you specify the CHOOSE=VALIDATE suboption of the SELECTION= option in the MODEL statement, the validation ASE is computed for the models at each step of the selection process. The model at the first step yielding the smallest validation ASE is selected.

Using the Validation ASE as the SELECT= Criterion

You request the validation ASE as the selection criterion by specifying the SELECT=VALIDATE suboption of the SELECTION= option in the MODEL statement. At step k of the selection process, the validation ASE is computed for each model where a candidate for entry is added or candidate for removal is dropped. The selected candidate for entry or removal is the one that yields a model with the minimal validation ASE.

Cross Validation

Deciding when to stop a selection method is a crucial issue in performing effect selection. Predictive performance of candidate models on data not used in fitting the model is one approach supported by PROC GLMSELECT for addressing this problem (see the section "Using Validation and Test Data" on page 2739). However, in some cases, you might not have sufficient data to create a sizable training set and a validation set that represent the predictive population well. In these cases, cross validation is an attractive alternative for estimating prediction error.

In k-fold cross validation, the data are split into k roughly equal-sized parts. One of these parts is held out for validation, and the model is fit on the remaining $k-1$ parts. This fitted model is used to compute the predicted residual sum of squares on the omitted part, and this process is repeated for each of k parts. The sum of the k predicted residual sum of squares so obtained is the estimate of the prediction error that is denoted by CVPRESS. Note that computing the CVPRESS statistic for k-fold cross validation requires fitting k different models, and so the work and memory requirements increase linearly with the number of cross validation folds.

You can use the CVMETHOD= option in the MODEL statement to specify the method for splitting the data into k parts. CVMETHOD=BLOCK(k) requests that the k parts be made of blocks of floor(n/k) or floor(n/k) + 1 successive observations, where n is the number of observations. CVMETHOD=SPLIT(k) requests that parts consist of observations $\{1, k+1, 2k+1, 3k+1, \ldots\}$, $\{2, k+2, 2k+2, 3k+2, \ldots\}$, $\ldots$, $\{k, 2k, 3k, \ldots\}$. CVMETHOD=RANDOM(k) partitions the data into random subsets each with roughly floor(n/k) observations. Finally, you can use the formatted value of an input data set variable to define the parts by specifying CVMETHOD=*variable*. This last partitioning method is useful in cases where you need to exercise extra control over how the data are partitioned by taking into account factors such as important but rare observations that you want to "spread out" across the various parts.

You can request details of the CVPRESS computations by specifying the CVDETAILS= option in the MODEL statement. When you use cross validation, the output data set created with an OUTPUT statement contains an integer-valued variable, _CVINDEX_, whose values indicate the subset to which an observation is assigned.

The widely used special case of n-fold cross validation when you have n observations is known as *leave-one-out* cross validation. In this case, each omitted part consists of one observation, and CVPRESS statistic can be efficiently obtained without refitting the model n times. In this case, the CVPRESS statistic is denoted simply by PRESS and is given by

$$\text{PRESS} = \sum_{i=1}^{n} \left(\frac{r_i}{1 - h_i} \right)^2$$

where r_i is the residual and h_i is the leverage of the ith observation. You can request *leave-one-out* cross validation by specifying PRESS instead of CV with the options SELECT=, CHOOSE=, and STOP= in the MODEL statement. For example, if the number of observations in the data set is 100, then the following two PROC GLMSELECT steps are mathematically equivalent, but the second step is computed much more efficiently:

```
proc glmselect;
    model y=x1-x10/selection=forward(stop=CV) cvMethod=split(100);
run;

proc glmselect;
    model y=x1-x10/selection=forward(stop=PRESS);
run;
```

Hastie, Tibshirani, and Friedman (2001) include a discussion about choosing the cross validation fold. They note that as an estimator of true prediction error, cross validation tends to have decreasing bias but increasing variance as the number of folds increases. They recommend five- or tenfold cross validation as a good compromise. By default, PROC GLMSELECT uses CVMETHOD=RANDOM(5) for cross validation.

Using Cross Validation as the STOP= Criterion

You request cross validation as the stopping criterion by specifying the STOP=CV suboption of the SELECTION= option in the MODEL statement. At step k of the selection process, the best candidate effect to enter or leave the current model is determined. Note that here "best candidate" means the effect that gives the best value of the SELECT= criterion that need not be the CV criterion. The CVPRESS score for the model with this candidate effect added or removed is determined. If this CVPRESS score is greater than the CVPRESS score for the model at step k, then the selection process terminates at step k.

Using Cross Validation as the CHOOSE= Criterion

When you specify the CHOOSE=CV suboption of the SELECTION= option in the MODEL statement, the CVPRESS score is computed for the models at each step of the selection process. The model at the first step yielding the smallest CVPRESS score is selected.

Using Cross Validation as the SELECT= Criterion

You request cross validation as the selection criterion by specifying the SELECT=CV suboption of the SELECTION= option in the MODEL statement. At step k of the selection process, the CVPRESS score is computed for each model where a candidate for entry is added or a candidate for removal is dropped. The selected candidate for entry or removal is the one that yields a model with the minimal CVPRESS score. Note that at each step of the selection process, this requires forming the CVPRESS statistic for all possible candidate models at the next step. Since forming the CVPRESS statistic for k-fold requires fitting k models, using cross validation as the selection criterion is computationally very demanding compared to using other selection criteria.

Displayed Output

The following sections describe the displayed output produced by PROC GLMSELECT. The output is organized into various tables, which are discussed in the order of appearance. Note that the contents of a table might change depending on the options you specify.

Model Information

The "Model Information" table displays basic information about the data sets and the settings used to control effect selection. These settings include the following:

- the selection method
- the criteria used to select effects, stop the selection, and choose the selected model
- the effect hierarchy enforced

For ODS purposes, the name of the "Model Information" table is "ModelInfo."

Performance Settings

The "Performance Settings" table displays settings that affect performance. These settings include whether threading is enabled and the number of CPUs available as well as the method used to build the crossproduct matrices. This table is displayed only if you specify the DETAILS option in the PERFORMANCE statement. For ODS purposes, the name of the "Performance Settings" table is "PerfSettings."

Number of Observations

The "Number of Observations" table displays the number of observations read from the input data set and the number of observations used in the analysis. If you specify a FREQ statement, the table

also displays the sum of frequencies read and used. If you use a PARTITION statement, the table also displays the number of observations used for each data role. If you specify TESTDATA= or VALDATA= data sets in the PROC GLMSELECT statement, then "Number of Observations" tables are also produced for these data sets. For ODS purposes, the name of the "Number of Observations" table is "NObs."

Class Level Information

The "Class Level Information" table lists the levels of every variable specified in the CLASS statement. For ODS purposes, the name of the "Class Level Information" table is "ClassLevelInfo."

Class Level Coding

The "Class Level Coding" table shows the coding used for variables specified in the CLASS statement. For ODS purposes, the name of the "Class Level Coding" table is "ClassLevelCoding."

Dimensions

The "Dimensions" table displays information about the number of effects and the number of parameters from which the selected model is chosen. If you use split classification variables, then this table also includes the number of effects after splitting is taken into account. For ODS purposes, the name of the "Dimensions" table is "Dimensions."

Candidates

The "Candidates" table displays the effect names and values of the criterion used to select entering or departing effects at each step of the selection process. The effects are displayed in sorted order from best to worst of the selection criterion. You request this table with the DETAILS= option in the MODEL statement. For ODS purposes, the name of the "Candidates" table is "Candidates."

Selection Summary

The "Selection Summary" table displays details about the sequence of steps of the selection process. For each step, the effect that was entered or dropped is displayed along with the statistics used to select the effect, stop the selection, and choose the selected model. You can request that additional statistics be displayed with the STATS= option in the MODEL statement. For all criteria that you can use for model selection, the steps at which the optimal values of these criteria occur are also indicated. For ODS purposes, the name of the "Selection Summary" table is "SelectionSummary."

Stop Reason

The "Stop Reason" table displays the reason why the selection stopped. To facilitate programmatic use of this table, an integer code is assigned to each reason and is included if you output this table by using an ODS OUTPUT statement. The reasons and their associated codes follow:

Code	Stop Reason
1	maximum number of steps done
2	specified number of steps done
3	specified number of effects in model
4	stopping criterion at local optimum
5	model is an exact fit
6	all entering effects are linearly dependent on those in the model
7	all effects are in the model
8	all effects have been dropped
9	requested full least squares fit completed
10	stepwise selection is cycling
11	dropping any effect does not improve the selection criterion
12	no effects are significant at the specified SLE or SLS levels
13	adding or dropping any effect does not improve the selection criterion
14	all remaining effects are required

For ODS purposes, the name of the "Stop Reason" table is "StopReason."

Stop Details

The "Stop Details" table compares the optimal value of the stopping criterion at the final model with how it would change if the best candidate effect were to enter or leave the model. For ODS purposes, the name of the "Stop Details" table is "StopDetails."

Selected Effects

The "Selected Effects" table displays a string containing the list of effects in the selected model. For ODS purposes, the name of the "Selected Effects" table is "SelectedEffects."

ANOVA

The "ANOVA" table displays an analysis of variance for the selected model. This table includes the following:

- the Source of the variation, Model for the fitted regression, Error for the residual error, and C Total for the total variation after correcting for the mean. The Uncorrected Total Variation is produced when the NOINT option is used.

- the degrees of freedom (DF) associated with the source
- the Sum of Squares for the term
- the Mean Square, the sum of squares divided by the degrees of freedom
- the F Value for testing the hypothesis that all parameters are zero except for the intercept. This is formed by dividing the mean square for Model by the mean square for Error.
- the Prob>F, the probability of getting a greater F statistic than that observed if the hypothesis is true. Note that these p-values are displayed only if you specify the "SHOWPVALUES" option in the MODEL statement. These p-values are generally liberal because they are not adjusted for the fact that the terms in the model have been selected.

You can request "ANOVA" tables for the models at each step of the selection process with the DETAILS= option in the MODEL statement. For ODS purposes, the name of the "ANOVA" table is "ANOVA."

Fit Statistics

The "Fit Statistics" table displays fit statistics for the selected model. The statistics displayed include the following:

- Root MSE, an estimate of the standard deviation of the error term. It is calculated as the square root of the mean square error.
- Dep Mean, the sample mean of the dependent variable
- R-square, a measure between 0 and 1 that indicates the portion of the (corrected) total variation attributed to the fit rather than left to residual error. It is calculated as SS(Model) divided by SS(Total). It is also called the *coefficient of determination*. It is the square of the multiple correlation—in other words, the square of the correlation between the dependent variable and the predicted values.
- Adj R-Sq, the adjusted R^2, a version of R^2 that has been adjusted for degrees of freedom. It is calculated as

$$\bar{R}^2 = 1 - \frac{(n-i)(1-R^2)}{n-p}$$

  where i is equal to 1 if there is an intercept and 0 otherwise, n is the number of observations used to fit the model, and p is the number of parameters in the model.
- fit criteria AIC, AICC, BIC, CP, and PRESS if they are used in the selection process or are requested with the STATS= option. See the section "Criteria Used in Model Selection Methods" on page 2727 for details and Table 42.6 for the formulas for evaluating these criteria.
- the CVPRESS statistic when cross validation is used in the selection process. See the section "Cross Validation" on page 2741 for details.
- the average square errors (ASE) on the training, validation, and test data. See the section "Using Validation and Test Data" on page 2739 for details.

You can request "Fit Statistics" tables for the models at each step of the selection process with the DETAILS= option in the MODEL statement. For ODS purposes, the name of the "Fit Statistics" table is "FitStatistics."

Cross Validation Details

The "Cross Validation Details" table displays the following:

- the fold number
- the number of observations used for fitting
- the number of observations omitted
- the predicted residual sum of squares on the omitted observations

You can request this table with the CVDETAILS= option in the MODEL statement whenever cross validation is used in the selection process. This table is displayed for the selected model, but you can request this table at each step of the selection process by using the DETAILS= option in the MODEL statement. For ODS purposes, the name of the "Cross Validation Details" table is "CVDetails."

Parameter Estimates

The "Parameter Estimates" table displays the parameters in the selected model and their estimates. The information displayed for each parameter in the selected model includes the following:

- the parameter label that includes the effect name and level information for effects containing classification variables
- the degrees of freedom (DF) for the parameter. There is one degree of freedom unless the model is not full rank.
- the parameter estimate
- the standard error, which is the estimate of the standard deviation of the parameter estimate
- T for H0: Parameter=0, the t test that the parameter is zero. This is computed as the parameter estimate divided by the standard error.
- the Prob > |T|, the probability that a t statistic would obtain a greater absolute value than that observed given that the true parameter is zero. This is the two-tailed significance probability. Note that these p-values are displayed only if you specify the "SHOWPVALUES" option in the MODEL statement. These p-values are generally liberal because they are not adjusted for the fact that the terms in the model have been selected.

If cross validation is used in the selection process, then you can request that estimates of the parameters for each cross validation fold be included in the "Parameter Estimates" table by using the CVDETAILS= option in the MODEL statement. You can request "Parameter Estimates" tables for the models at each step of the selection process with the DETAILS= option in the MODEL statement. For ODS purposes, the name of the "Parameter Estimates" table is "ParameterEstimates."

Score Information

For each SCORE statement, the "Score Information" table displays the names of the score input and output data sets, and the number of observations that were read and successfully scored. For ODS purposes, the name of the "Score Information" table is "ScoreInfo."

Timing Breakdown

The "Timing Breakdown" table displays a broad breakdown of where time was spent in the PROC GLMSELECT step. This table is displayed only if you specify the DETAILS option in the PERFORMANCE statement. If multithreaded BY-group processing is employed, then the number of threads used for the various phases of the computation is displayed. For ODS purposes, the name of the "Timing Breakdown" table is "Timing."

ODS Table Names

PROC GLMSELECT assigns a name to each table it creates. You can use these names to reference the table when you use the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 42.7.

For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 42.7 ODS Tables Produced by PROC GLMSELECT

ODS Table Name	Description	Statement	Option
ANOVA	selected model ANOVA table	MODEL	default
BSplineDetails	B-spline basis details	EFFECT	DETAILS
Candidates	entry/removal effect ranking	MODEL	DETAILS=
ClassLevelCoding	classification variable coding	CLASS	SHOWCODING
ClassLevelInfo	classification variable levels	CLASS	default
CollectionLevelInfo	levels of collection effects	EFFECT	DETAILS
CVDetails	cross validation PRESS by fold	MODEL	CVDETAILS=
Dimensions	number of effects and parameters	MODEL	default
FitStatistics	selected model fit statistics	MODEL	default
MMLevelInfo	levels of MM effects	EFFECT	DETAILS
ModelInfo	model information	MODEL	default
NObs	number of observations	MODEL	default
ParameterNames	labels for column names in the design matrix	PROC	OUTDESIGN(names)
ParameterEstimates	selected model parameter estimates	MODEL	default
PerfSettings	performance settings	PERFORMANCE	DETAILS
PolynomialDetails	polynomial details	EFFECT	DETAILS
PolynomialScaling	polynomial scaling	EFFECT	DETAILS

Table 42.7 *continued*

ODS Table Name	Description	Statement	Option
ScoreInfo	score request information	SCORE	default
SelectedEffects	list of selected effects	MODEL	default
SelectionSummary	selection summary	MODEL	default
StopDetails	stopping criterion details	MODEL	default
StopReason	reason why selection stopped	MODEL	default
Timing	timing details	PERFORMANCE	DETAILS
TPFSplineDeatils	TPF spline basis details	EFFECT	DETAILS

ODS Graphics

This section describes the use of ODS for creating statistical graphs with the GLMSELECT procedure. To request these graphs you must specify the ODS GRAPHICS statement and request plots with the PLOTS= option in the PROC GLMSELECT statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS." The following sections describe the ODS graphical displays produced by PROC GLMSELECT. The examples use the Baseball data set that is described in the section "Getting Started: GLMSELECT Procedure" on page 2686.

Candidates Plot

You request the "Candidates Plot" by specifying the PLOTS=CANDIDATES option in the PROC GLMSELECT statement and the DETAILS=STEPS option in the MODEL statement. This plot shows the values of selection criterion for the candidate effects for entry or removal, sorted from best to worst from left to right across the plot. The leftmost candidate displayed is the effect selected for entry or removal at that step. You can use this plot to see at what steps the decision about which effect to add or drop is clear-cut. See Figure 42.5 for an example.

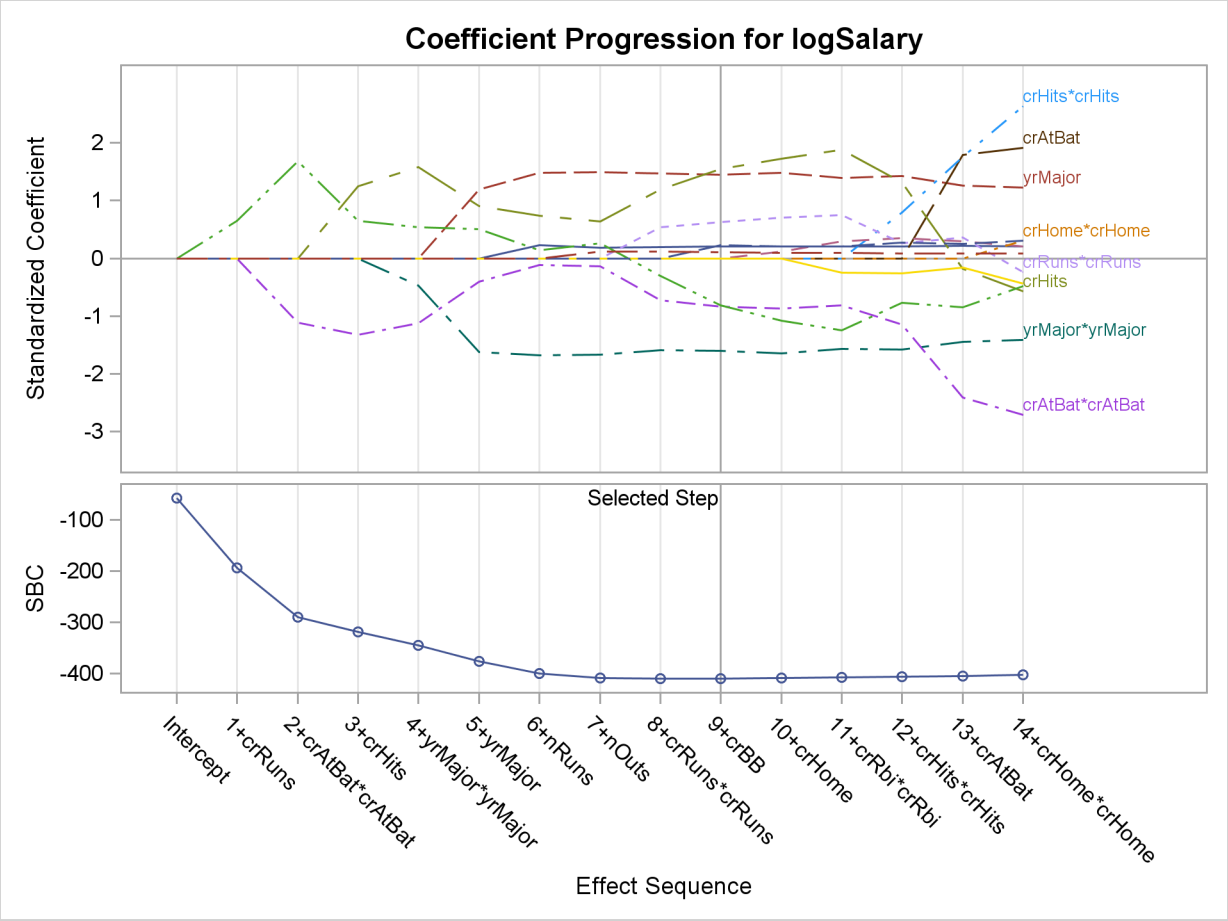
Coefficient Panel

When you specify the PLOTS=COEFFICIENTS option in the PROC GLMSELECT statement, PROC GLMSELECT produces a panel of two plots showing how the standardized coefficients and the criterion used to choose the final model evolve as the selection progresses. The following statements provide an example:

```
ods graphics on;

proc glmselect data=baseball plots=coefficients;
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                    yrMajor|yrMajor crAtBat|crAtBat crHits|crHits
                    crHome|crHome crRuns|crRuns crRbi|crRbi
                    crBB|crBB league division nOuts nAssts nError /
                    selection=forward(stop=AICC CHOOSE=SBC);
run;
```

Figure 42.15 shows the requested graphic. The upper plot in the panel displays the standardized coefficients as a function of the step number. You can request standardized coefficients in the parameter estimates tables by specifying the STB option in the MODEL statement, but this option is not required to produce this plot. To help in tracing the changes in a parameter, the standardized coefficients for each parameter are connected by lines. Coefficients corresponding to effects that are not in the selected model at a step are zero and hence not observable. For example, consider the parameter crAtBat*crAtBat in Output 42.15. Because crAtBat*crAtBat enters the model at step 2, the line that represents this parameter starts rising from zero at step 1 when crRuns enters the model. Parameters that are nonzero at the final step of the selection are labeled if their magnitudes are greater than 1% of the range of the magnitudes of all the nonzero parameters at this step. To avoid collision, labels corresponding to parameters with similar values at the final step might get suppressed. You can control when this label collision avoidance occurs by using the LABELGAP= suboption of the PLOTS=COEFFICIENTS option. Planned enhancements to the automatic label collision avoidance algorithm will obviate the need for this option in future releases of the GLMSELECT procedure.

Figure 42.15 Coefficient Panel



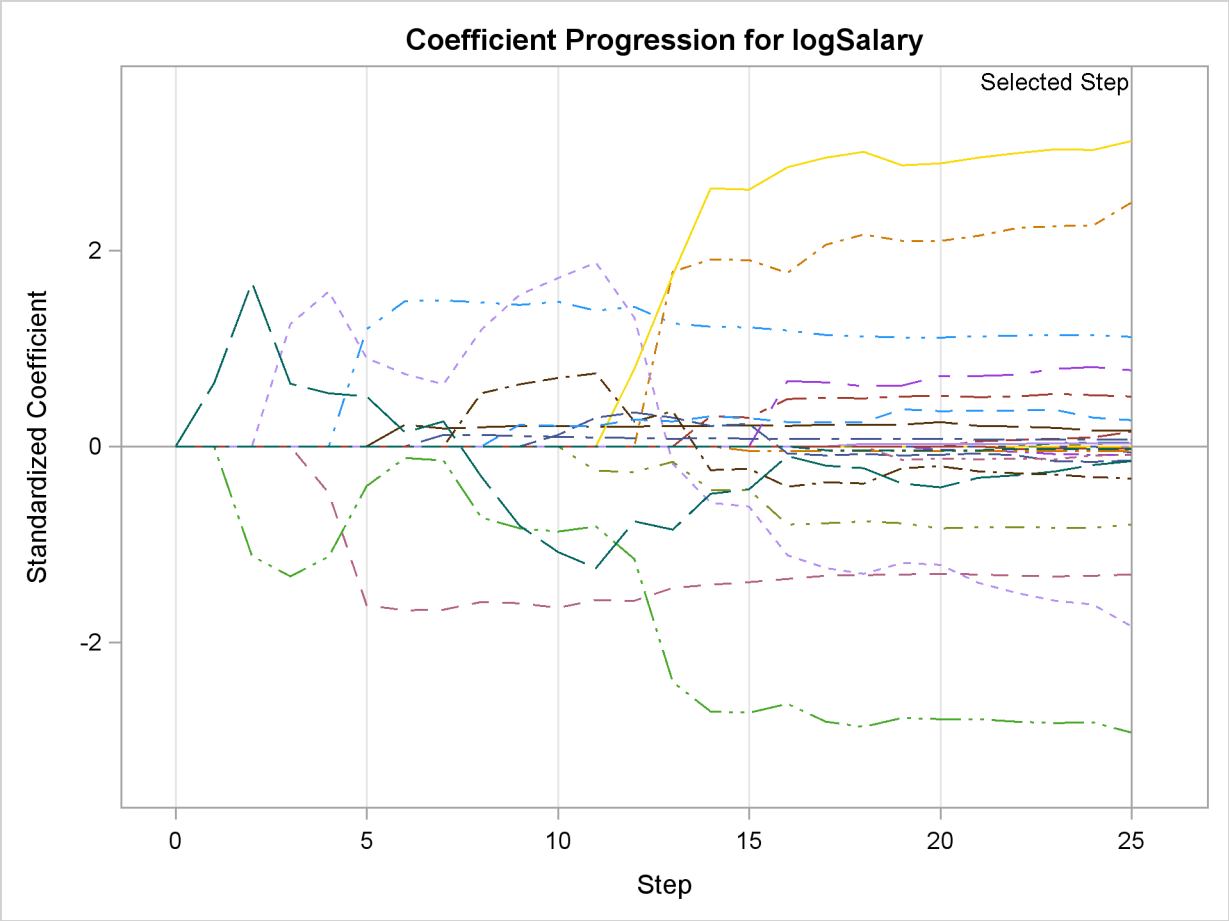
The lower plot in the panel shows how the criterion used to choose among the examined models progresses. The selected step occurs at the optimal value of this criterion. In this example, this criterion is the SBC criterion and it achieves its minimal value at step 9 of the forward selection.

In some cases, particularly when the final step contains a large number of parameters, you might be interested in using this plot only to discern if and when the parameters in the model are essential unchanged beyond a certain step. In such cases, you might want to suppress the labeling of the parameters and use a numeric axis on the horizontal axis of the plot. You can do this using the STEPAXIS= and MAXPARMLABEL= suboptions of the PLOTS=CRITERIA option. The following statements provide an example:

```
proc glmselect data=baseball
     plots(unpack maxparmlabel=0 stepaxis=number)=coefficients;

  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                    yrMajor|yrMajor crAtBat|crAtBat crHits|crHits
                    crHome|crHome crRuns|crRuns crRbi|crRbi
                    crBB|crBB league division nOuts nAssts nError /
                    selection=forward(stop=none);
run;
```

The UNPACK = option requests that the plots of the coefficients and CHOOSE= criterion be shown in separate plots. The STEPAXIS=NUMBER option requests a numeric horizontal axis showing step number, and the MAXPAMLABEL=0 option suppresses the labels for the parameters. The "Coefficient Plot" is shown in Figure 42.16. You can see that the standardized coefficients do not vary greatly after step 16.

Figure 42.16 Coefficient Plot



Criterion Panel

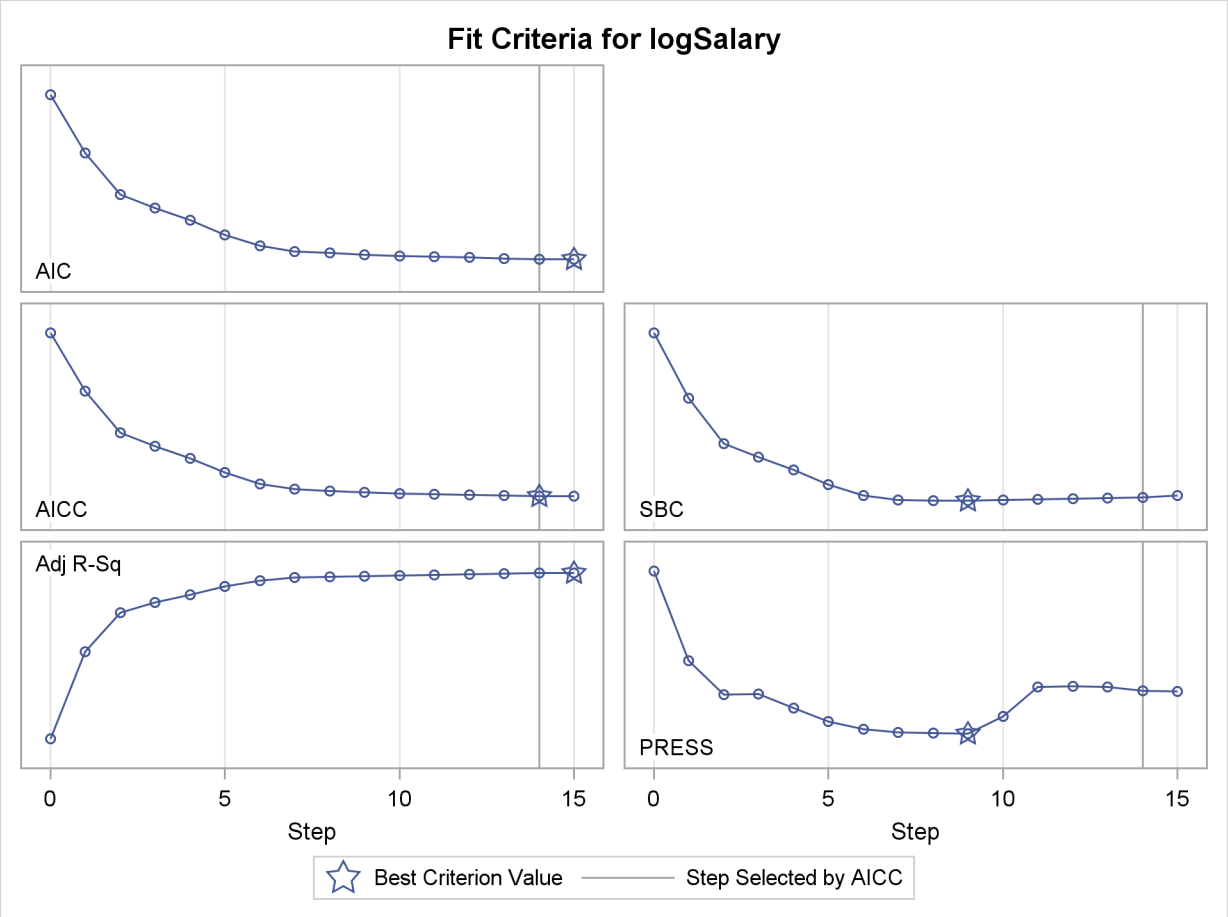
You request the criterion panel by specifying the PLOTS=CRITERIA option in the PROC GLMSELECT statement. This panel displays the progression of the ADJRSQ, AIC, AICC, and SBC criteria, as well as any other criteria that are named in the CHOOSE=, SELECT=, STOP=, or STATS= option in the MODEL statement.

The following statements provide an example:

```
proc glmselect data=baseball plots=criteria;
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                    yrMajor|yrMajor crAtBat|crAtBat crHits|crHits
                    crHome|crHome crRuns|crRuns crRbi|crRbi
                    crBB|crBB league division nOuts nAssts nError /
                    selection=forward(steps=15 choose=AICC)
                    stats=PRESS;
run;
```

Figure 42.17 shows the requested criterion panel. Note that the PRESS criterion is included in the panel because it is named in the STATS= option in the MODEL statement. The selected step is displayed as a vertical reference line on the plot of each criterion, and the legend indicates which of these criteria is used to make the selection. If the selection terminates for a reason other than optimizing a criterion displayed on this plot, then the legend will not report a reason for the selected step. The optimal value of each criterion is indicated with the "Star" marker. Note that it is possible that a better value of a criterion might have been reached had more steps of the selection process been done.

Figure 42.17 Criterion Panel



Average Square Error Plot

You request the average square error plot by specifying the PLOTS=ASE option in the PROC GLMSELECT statement. This plot shows the progression of the average square error (ASE) evaluated separately on the training data, and the test and validation data whenever these data are provided with the TESTDATA= and VALDATA= options or are produced by using a PARTITION statement. You use the plot to detect when overfitting the training data occurs. The ASE decreases monotonically on the training data as parameters are added to a model. However, the average square error on test and validation data typically starts increasing when overfitting occurs. See Output 42.1.9 and Output 42.2.6 for examples.

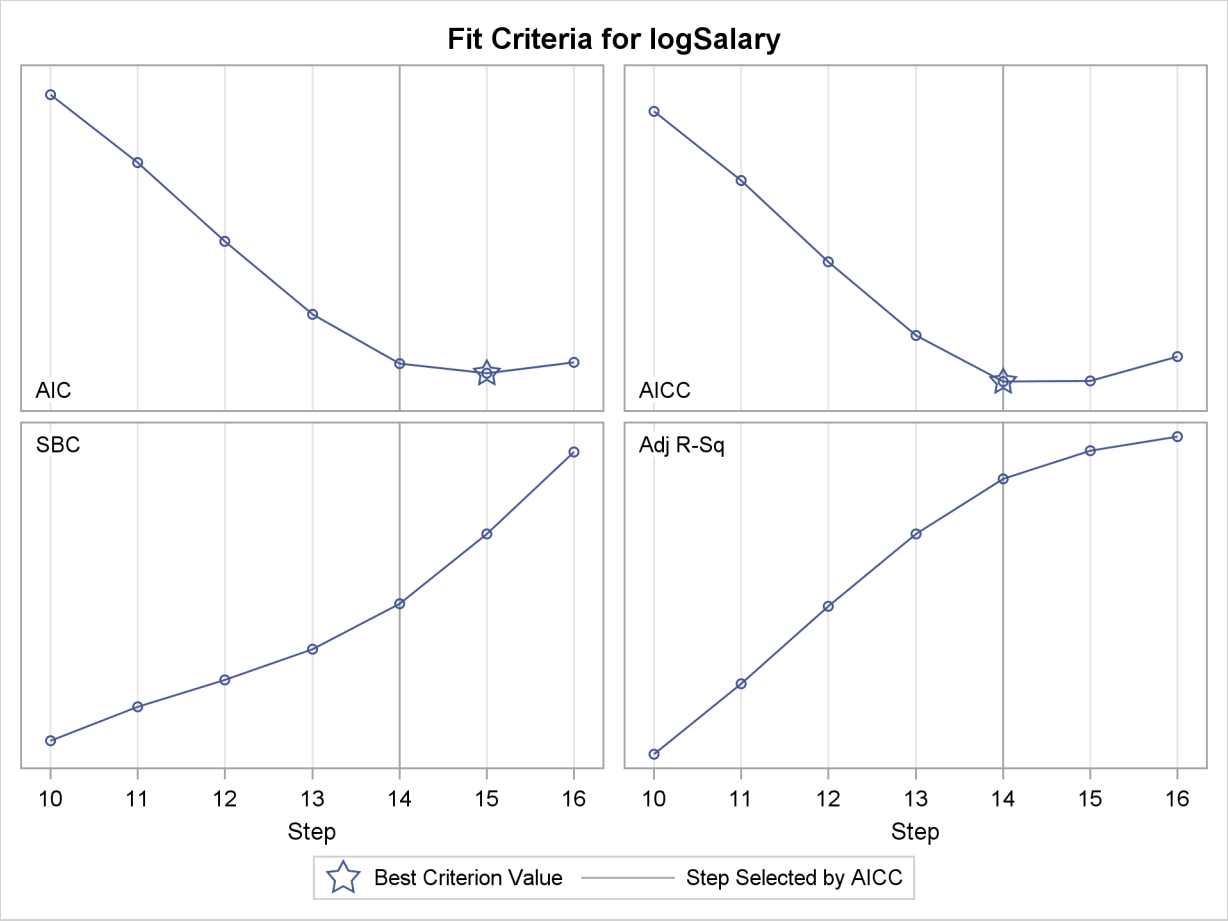
Examining Specific Step Ranges

The coefficient panel, criterion panel, and average square error plot display information for all the steps examined in the selection process. In some cases, you might want to focus attention on just a particular step range. For example, it is hard to discern the variation in the criteria displayed in Figure 42.17 near the selected step because the variation in these criteria in the steps close to the selected step is small relative to the variation across all steps. You can request a range of steps to display using the STARTSTEP= and ENDSTEP= suboptions of the PLOTS= option. You can specify these options as both global and specific plot options, with the specific options taking precedence if both are specified. The following statements provide an example:

```
proc glmselect data=baseball plots=criteria(startstep=10 endstep=16);
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                    yrMajor|yrMajor crAtBat|crAtBat crHits|crHits
                    crHome|crHome crRuns|crRuns crRbi|crRbi
                    crBB|crBB league division nOuts nAssts nError /
                    selection=forward(stop=none choose=AICC);
run;

ods graphics off;
```

Figure 42.18 shows the progression of the fit criteria between steps 10 and 16. Note that if the optimal value of a criterion does not occur in this specified step range, then no optimal marker appears for that criterion. The plot of the SBC criterion in Figure 42.18 is one such case.

Figure 42.18 Criterion Panel for Specified Step Range



ODS Graph Names

PROC GLMSELECT assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 42.8.

To request these graphs you must specify the ODS GRAPHICS statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Table 42.8 ODS Graphics Produced by PROC GLMSELECT

ODS Graph Name	Plot Description	PLOTS Option
AdjRSqPlot	Adjusted R-square by step	CRITERIA(UNPACK)
AICCPlot	Corrected Akaike information criterion by step	CRITERIA(UNPACK)
AICPlot	Akaike information criterion by step	CRITERIA(UNPACK)
ASEPlot	Average square errors by step	ASE
BICPlot	Sawa's Bayesian information criterion by step	CRITERIA(UNPACK)
CandidatesPlot	SELECT criterion by effect	CANDIDATES
ChooseCriterionPlot	CHOOSE criterion by step	COEFFICIENTS(UNPACK)

Table 42.8 *continued*

ODS Graph Name	Plot Description	PLOTS= Option
CoefficientPanel	Coefficients and CHOOSE criterion by step	COEFFICIENTS
CoefficientPlot	Coefficients by step	COEFFICIENTS(UNPACK)
CPPlot	Mallows C_p by step	CRITERIA(UNPACK)
CriterionPanel	Fit criteria by step	CRITERIA
CVPRESSPlot	Cross validation predicted RSS by step	CRITERIA(UNPACK)
PRESSPlot	Predicted RSS by step	CRITERIA(UNPACK)
SBCPlot	Schwarz Bayesian information criterion by step	CRITERIA(UNPACK)
ValidateASEPlot	Average square error on validation data by step	CRITERIA(UNPACK)

Examples: GLMSELECT Procedure

Example 42.1: Modeling Baseball Salaries Using Performance Statistics

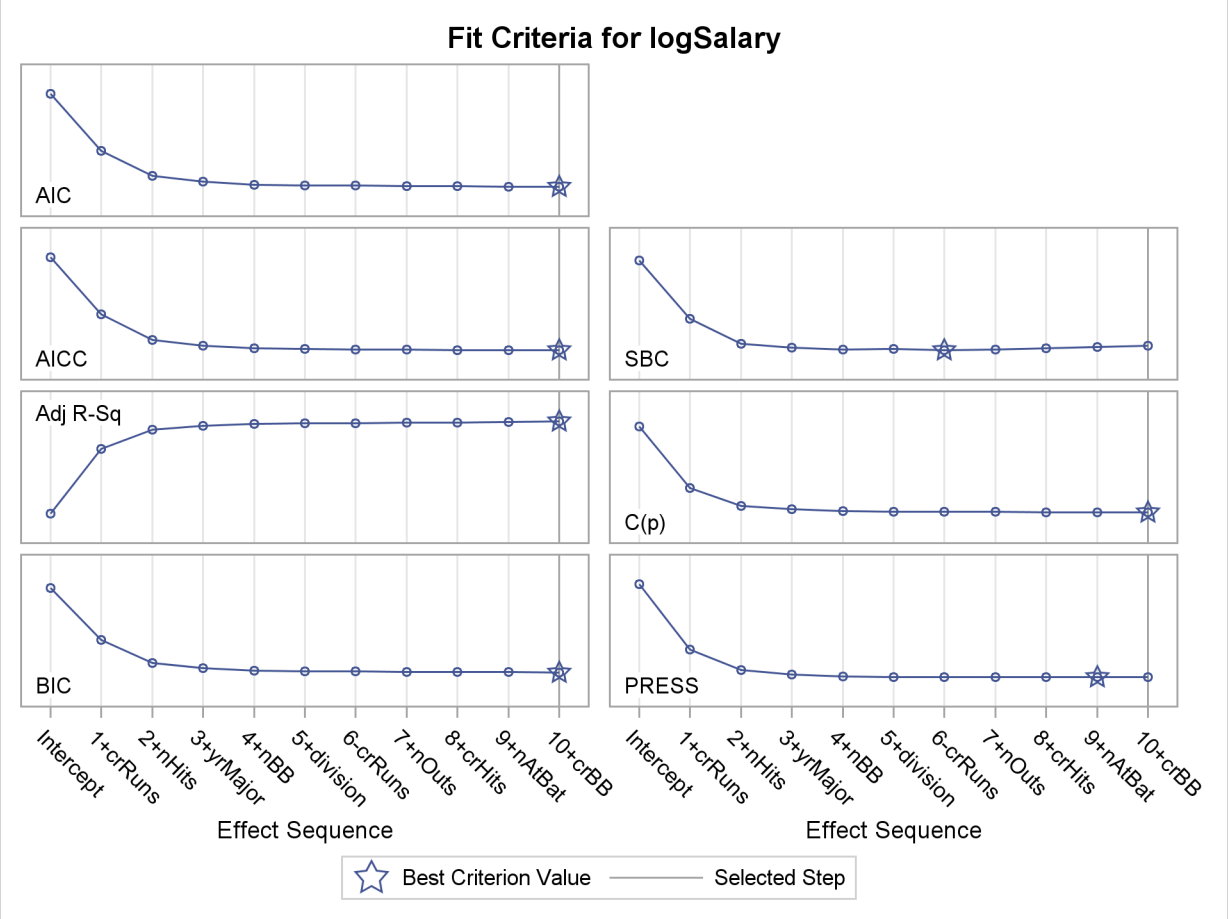
This example continues the investigation of the baseball data set introduced in the section "Getting Started: GLMSELECT Procedure" on page 2686. In that example, the default stepwise selection method based on the SBC criterion was used to select a model. In this example, model selection that uses other information criteria and out-of-sample prediction criteria is explored.

PROC GLMSELECT provides several selection algorithms that you can customize by specifying criteria for selecting effects, stopping the selection process, and choosing a model from the sequence of models at each step. For more details on the criteria available, see the section "Criteria Used in Model Selection Methods" on page 2727. The SELECT=SL suboption of the SELECTION= option in the MODEL statement in the following code requests the traditional hypothesis test-based stepwise selection approach, where effects in the model that are not significant at the stay significance level (SLS) are candidates for removal and effects not yet in the model whose addition is significant at the entry significance level (SLE) are candidates for addition to the model.

```
ods graphics on;

proc glmselect data=baseball plot=CriterionPanel;
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                    yrMajor crAtBat crHits crHome crRuns crRbi
                    crBB league division nOuts nAssts nError
                  / selection=stepwise(select=SL) stats=all;
run;
```

The default SLE and SLS values of 0.15 might not be appropriate for these data. One way to investigate alternative ways to stop the selection process is to assess the sequence of models in terms of model fit statistics. The STATS=ALL option in the MODEL statement requests that all model fit statistics for assessing the sequence of models of the selection process be displayed. To help in the interpretation of the selection process, you can use graphics supported by PROC GLMSELECT. You enable these graphical displays by specifying the ODS GRAPHICS statement. For general information about ODS graphics, see Chapter 21, "Statistical Graphics Using ODS." With ODS graphics enabled, the PLOTS=CRITRERIONPANEL option in the PROC GLMSELECT statement produces the criterion panel shown in Output 42.1.1.

Output 42.1.1 Criterion Panel



You can see in Output 42.1.1 that this stepwise selection process would stop at an earlier step if you use the Schwarz Bayesian information criterion (SBC) or predicted residual sum of squares (PRESS) to assess the selected models as stepwise selection progresses. You can use the CHOOSE= suboption of the SELECTION= option in the MODEL statement to specify the criterion you want to use to select among the evaluated models. The following statements use the PRESS statistic to choose among the models evaluated during the stepwise selection.

```
proc glmselect data=baseball;
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                    yrMajor crAtBat crHits crHome crRuns crRbi
                    crBB league division nOuts nAssts nError
                  / selection=stepwise(select=SL choose=PRESS);
run;
```

Note that the selected model is the model at step 9. By default, PROC GLMSELECT displays the selected model, ANOVA and fit statistics, and parameter estimates for the selected model. These are shown in Output 42.1.2.

Output 42.1.2 Details of Selected Model

The GLMSELECT Procedure
Selected Model

The selected model, based on PRESS, is the model at Step 9.

Effects: Intercept nAtBat nHits nBB yrMajor crHits division nOuts

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value
Model	7	124.67715	17.81102	55.07
Error	255	82.47658	0.32344	
Corrected Total	262	207.15373		

Root MSE	0.56872
Dependent Mean	5.92722
R-Square	0.6019
Adj R-Sq	0.5909
AIC	-23.98522
AICC	-23.27376
PRESS	88.55275
SBC	-260.40799

Parameter Estimates

Parameter		DF	Estimate	Standard Error	t Value
Intercept		1	4.176133	0.150539	27.74
nAtBat		1	-0.001468	0.000946	-1.55
nHits		1	0.011078	0.002983	3.71
nBB		1	0.007226	0.002115	3.42
yrMajor		1	0.070056	0.018911	3.70
crHits		1	0.000247	0.000143	1.72
division	East	1	0.143082	0.070972	2.02
division	West	0	0	.	.
nOuts		1	0.000241	0.000134	1.81

Even though the model that is chosen to give the smallest value of the PRESS statistic is the model at step 9, the stepwise selection process continues to the step where the stopping condition based on entry and stay significance levels is met. If you use the PRESS statistic as the stopping criterion, the stepwise selection process stops at step 9. This ability to stop at the first extremum of the criterion you specify can significantly reduce the amount of computation done, especially in the cases where you are selecting from a large number of effects. The following statements request stopping based on the PRESS statistic. The stop reason and stop details tables are shown in Output 42.1.3.

```
proc glmselect data=baseball plot=Coefficients;
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                    yrMajor crAtBat crHits crHome crRuns crRbi
                    crBB league division nOuts nAssts nError
                  / selection=stepwise(select=SL stop=PRESS);
run;
```

Output 42.1.3 Stopping Based on PRESS Statistic

The GLMSELECT Procedure

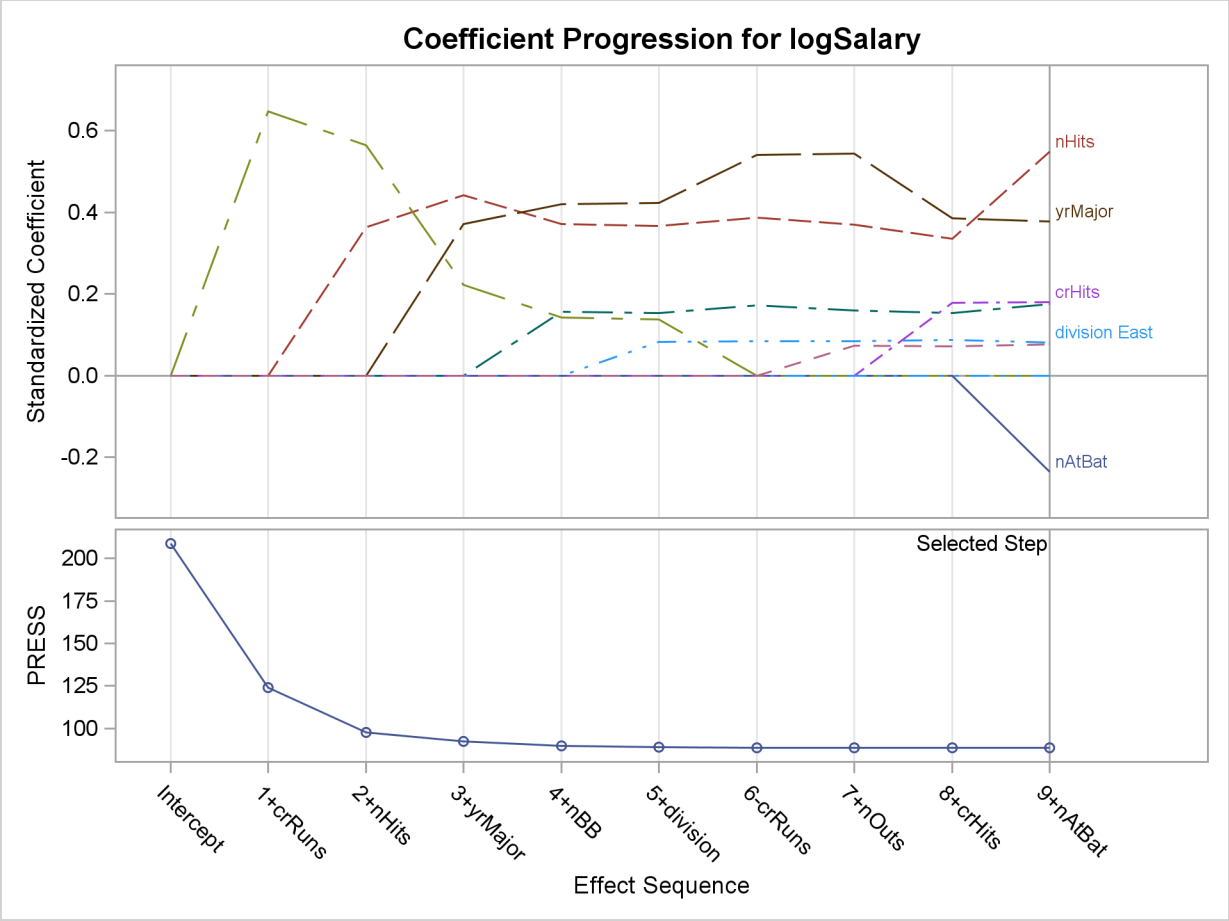
Selection stopped at a local minimum of the PRESS criterion.

Stop Details

Candidate For	Effect	Candidate PRESS		Compare PRESS
Entry	crBB	88.6321	>	88.5528
Removal	nAtBat	88.6866	>	88.5528

The PLOTS=COEFFICIENTS specification in the PROC GLMSELECT statement requests a plot that enables you to visualize the selection process.

Output 42.1.4 Coefficient Progression Plot



Output 42.1.4 shows the standardized coefficients of all the effects selected at some step of the stepwise method plotted as a function of the step number. This enables you to assess the relative importance of the effects selected at any step of the selection process as well as providing information as to when effects entered the model. The lower plot in the panel shows how the criterion used to choose the selected model changes as effects enter or leave the model.

Model selection is often done in order to obtain a parsimonious model that can be used for prediction on new data. An ever-present danger is that of selecting a model that overfits the "training" data used in the fitting process, yielding a model with poor predictive performance. Using cross validation is one way to assess the predictive performance of the model. Using k-fold cross validation, the training data are subdivided into k parts, and at each step of the selection process, models are obtained on each of the k subsets of the data obtained by omitting one of these parts. The cross validation predicted residual sum of squares, denoted CV PRESS, is obtained by summing the squares of the residuals when each of these submodels is scored on the data omitted in fitting the submodel. Note that the PRESS statistic corresponds to the special case of "leave-one-out" cross validation.

In the preceding example, the PRESS statistic was used to choose among models that were chosen based on entry and stay significance levels. In the following statements, the SELECT=CVPRESS suboption of the SELECTION= option in the MODEL statement requests that the CV PRESS statis-

tic itself be used as the selection criterion. The DROP=COMPETITIVE suboption requests that additions and deletions be considered simultaneously when deciding whether to add or remove an effect. At any step, the CV PRESS statistic for all models obtained by deleting one effect from the model or adding one effect to the model is computed. Among these models, the one yielding the smallest value of the CV PRESS statistic is selected and the process is repeated from this model. The stepwise selection terminates if all additions or deletions increase the CV PRESS statistic. The CVMETHOD=SPLIT(5) option requests five-fold cross validation with the five subsets consisting of observations $\{1, 6, 11, \ldots\}$, $\{2, 7, 12, \ldots\}$, and so on.

```
proc glmselect data=baseball plot=Candidates;
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                    yrMajor crAtBat crHits crHome crRuns crRbi
                    crBB league division nOuts nAssts nError
                  / selection=stepwise(select=CV drop=competitive)
                    cvMethod=split(5);
run;
```

The selection summary table is shown in Output 42.1.5. By comparing Output 42.1.5 and Output 42.6 you can see that the sequence of models produced is different from the sequence when the stepwise selection is based on the SBC statistic.

Output 42.1.5 Stepwise Selection Based on Cross Validation

The GLMSELECT Procedure

Stepwise Selection Summary

Step	Effect Entered	Effect Removed	Number Effects In	Number Parms In	CV PRESS
0	Intercept		1	1	208.9638
1	crRuns		2	2	122.5755
2	nHits		3	3	96.3949
3	yrMajor		4	4	92.2117
4	nBB		5	5	89.5242
5		crRuns	4	4	88.6917
6	league		5	5	88.0417
7	nError		6	6	87.3170
8	division		7	7	87.2147
9	nHome		8	8	87.0960*

* Optimal Value Of Criterion

If you have sufficient data, another way you can assess the predictive performance of your model is to reserve part of your data for testing your model. You score the model obtained using the training data on the test data and assess the predictive performance on these data that had no role in the selection process. You can also reserve part of your data to validate the model you obtain in the training process. Note that the validation data are not used in obtaining the coefficients of the model, but they are used to decide when to stop the selection process to limit overfitting.

PROC GLMSELECT enables you to partition your data into disjoint subsets for training validation and testing roles. This partitioning can be done by using random proportions of the data, or you can designate a variable in your data set that defines which observations to use for each role. See the section "PARTITION Statement" on page 2715 for more details.

The following statements randomly partition the baseball data set, using 50% for training, 30% for validation, and 20% for testing. The model selected at each step is scored on the validation data, and the average residual sums of squares (ASE) is evaluated. The model yielding the lowest ASE on the validation data is selected. The ASE on the test data is also evaluated, but these data play no role in the selection process. Note that a seed for the pseudo-random number generator is specified in the PROC GLMSELECT statement.

```
proc glmselect data=baseball plots=(CriterionPanel ASE) seed=1;
  partition fraction(validate=0.3 test=0.2);
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                yrMajor crAtBat crHits crHome crRuns crRbi
                crBB league division nOuts nAssts nError
               / selection=forward(choose=validate stop=10);
run;
```

Output 42.1.6 Number of Observations Table

The GLMSELECT Procedure

Number of Observations Read	322
Number of Observations Used	263
Number of Observations Used for Training	132
Number of Observations Used for Validation	80
Number of Observations Used for Testing	51

Output 42.1.6 shows the number of observation table. You can see that of the 263 observations that were used in the analysis, 132 (50.2%) observations were used for model training, 80 (30.4%) for model validation, and 51 (19.4%) for model testing.

Output 42.1.7 Selection Summary and Stop Reason

The GLMSELECT Procedure

Forward Selection Summary

Step	Effect Entered	Number Effects In	Number Parms In	SBC	ASE
0	Intercept	1	1	-30.8531	0.7628
1	crRuns	2	2	-93.9367	0.4558
2	nHits	3	3	-126.2647	0.3439
3	yrMajor	4	4	-128.7570	0.3252
4	nBB	5	5	-132.2409*	0.3052
5	division	6	6	-130.7794	0.2974
6	nOuts	7	7	-128.5897	0.2914
7	nRBI	8	8	-125.7825	0.2868
8	nHome	9	9	-124.7709	0.2786
9	nAtBat	10	10	-121.3767	0.2754

* Optimal Value Of Criterion

Forward Selection Summary

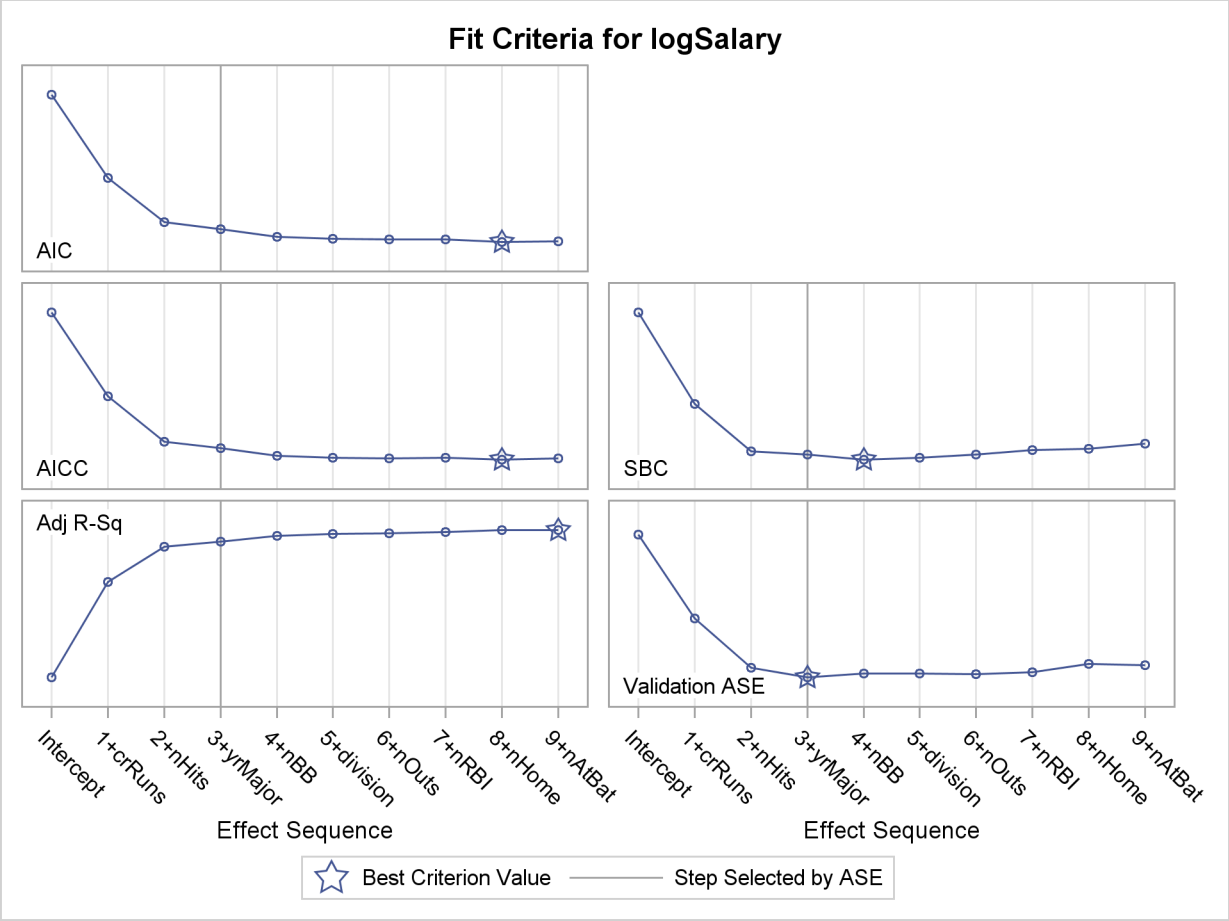
Step	Effect Entered	Validation ASE	Test ASE
0	Intercept	0.7843	0.8818
1	crRuns	0.4947	0.4210
2	nHits	0.3248	0.4697
3	yrMajor	0.2920*	0.4614
4	nBB	0.3065	0.4297
5	division	0.3050	0.4218
6	nOuts	0.3028	0.4186
7	nRBI	0.3097	0.4489
8	nHome	0.3383	0.4533
9	nAtBat	0.3337	0.4580

* Optimal Value Of Criterion

Selection stopped at the first model containing the specified number of effects (10).

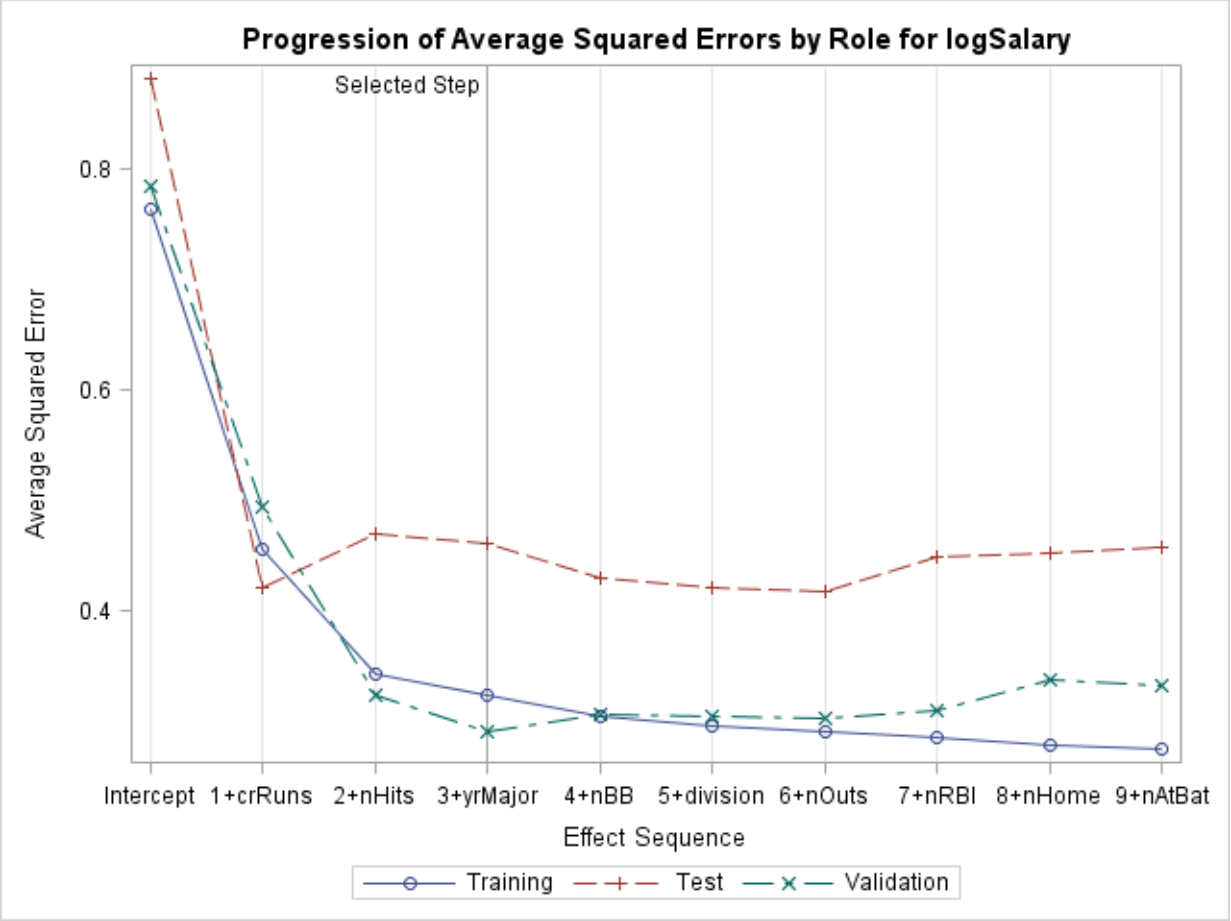
Output 42.1.7 shows the selection summary table and the stop reason. The forward selection stops at step 9 since the model at this step contains 10 effects, and so it satisfies the stopping criterion requested with the "STOP=10" suboption. However, the selected model is the model at step 3, where the validation ASE, the CHOOSE= criterion, achieves its minimum.

Output 42.1.8 Criterion Panel



The criterion panel in Output 42.1.8 shows how the various criteria evolved as the stepwise selection method proceeded. Note that other than the ASE evaluated on the validation data, these criteria are evaluated on the training data.

Output 42.1.9 Average Square Errors by Role



Finally, the ASE plot in Output 42.1.9 shows how the average square error evolves on the training, validation, and test data. Note that while the ASE on the training data continued decreasing as the selection steps proceeded, the ASE on the test and validation data behave more erratically.

LASSO selection, pioneered by Tibshirani (1996), is a constrained least squares method that can be viewed as a stepwise-like method where effects enter and leave the model sequentially. You can find additional details about the LASSO method in the section "Lasso Selection (LASSO)" on page 2725. Note that when classification effects are used with LASSO, the design matrix columns for all effects containing classification variables can enter or leave the model individually. The following statements perform LASSO selection for the baseball data. The LASSO selection summary table is shown in Output 42.1.10.

```
 proc glmselect data=baseball plot=CriterionPanel ;
  class league division;
  model logSalary = nAtBat nHits nHome nRuns nRBI nBB
                    yrMajor crAtBat crHits crHome crRuns crRbi
                    crBB league division nOuts nAssts nError
                  / selection=lasso(choose=CP steps=20);
run;

ods graphics off;
```

Output 42.1.10 Selection Summary for LASSO Selection

The GLMSELECT Procedure

LASSO Selection Summary

Step	Effect Entered	Effect Removed	Number Effects In	CP
0	Intercept		1	375.9275
1	crRuns		2	328.6492
2	crHits		3	239.5392
3	nHits		4	134.0374
4	nBB		5	111.6638
5	crRbi		6	81.7296
6	yrMajor		7	75.0428
7	nRBI		8	30.4494
8	division_East		9	29.9913
9	nOuts		10	25.1656
10		crRuns	9	18.7295
11		crRbi	8	15.1683
12	nError		9	16.6233
13	nHome		10	16.3741
14	league_American		11	14.8794
15		nRBI	10	8.8477*
16	crBB		11	9.2242
17	crRuns		12	10.7608
18	nAtBat		13	11.6266
19	nAssts		14	11.8572
20	crAtBat		15	13.4020

* Optimal Value Of Criterion

LASSO Selection Summary

Step	Effect Entered	Effect Removed	SBC
0	Intercept		-57.2041
1	crRuns		-72.8102
2	crHits		-111.5450
3	nHits		-170.1414
4	nBB		-181.5853
5	crRbi		-200.1160
6	yrMajor		-201.5014
7	nRBI		-236.7565
8	division_East		-233.5248
9	nOuts		-234.5631

* Optimal Value Of Criterion

Output 42.1.10 *continued*

The GLMSELECT Procedure

LASSO Selection Summary

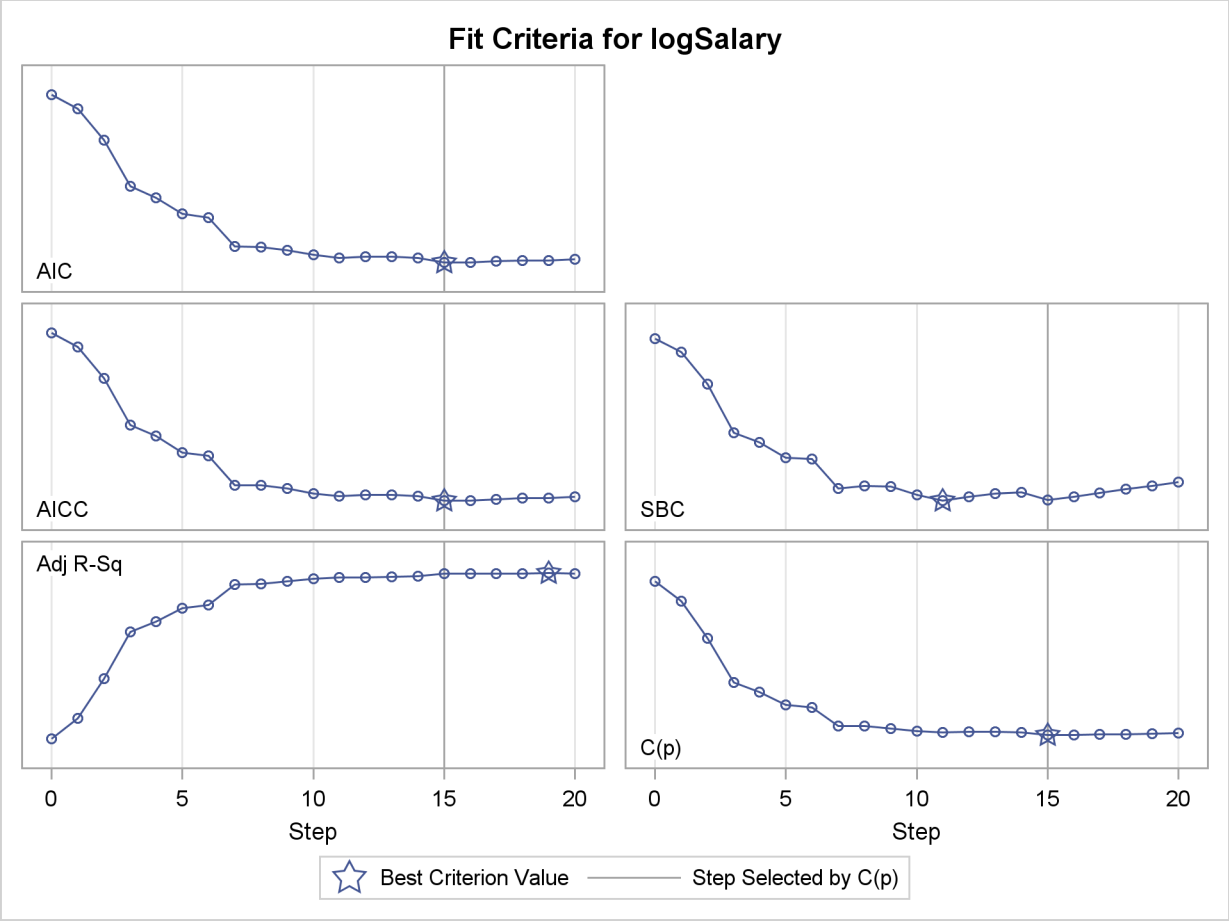
Step	Effect Entered	Effect Removed	SBC
10		crRuns	−244.5223
11		crRbi	−251.6560*
12	nError		−246.6311
13	nHome		−243.3298
14	league_American		−241.3253
15		nRBI	−251.0743
16	crBB		−247.2030
17	crRuns		−242.1184
18	nAtBat		−237.7433
19	nAssts		−234.0495
20	crAtBat		−228.9628

* Optimal Value Of Criterion

Selection stopped at the specified number of steps (20).

Note that effects enter and leave sequentially. In this example, the STEPS= suboption of the SELECTION= option specifies that 20 steps of LASSO selection be done. You can see how the various model fit statistics evolved in Output 42.1.11.

Output 42.1.11 Criterion Panel



The CHOOSE=CP suboption specifies that the selected model be the model at step 15 that yields the optimal value of Mallows' C(p) statistic. Details of this selected model are shown in Output 42.1.12.

Output 42.1.12 Selected Model

```
                         The GLMSELECT Procedure
                              Selected Model

          The selected model, based on C(p), is the model at Step 15.


Effects: Intercept nHits nHome nBB yrMajor crHits league_American division_East
         nOuts nError

                           Analysis of Variance

                                      Sum of          Mean
       Source                 DF      Squares        Square    F Value

       Model                   9    125.24302      13.91589      42.98
       Error                 253     81.91071       0.32376
       Corrected Total       262    207.15373
```

Output 42.1.12 *continued*

Root MSE	0.56900
Dependent Mean	5.92722
R-Square	0.6046
Adj R-Sq	0.5905
AIC	-21.79589
AICC	-20.74409
BIC	-283.91417
C(p)	8.84767
SBC	-251.07435

Parameter Estimates

Parameter	DF	Estimate
Intercept	1	4.204236
nHits	1	0.006942
nHome	1	0.002785
nBB	1	0.005727
yrMajor	1	0.067054
crHits	1	0.000249
league_American	1	-0.079607
division_East	1	0.134723
nOuts	1	0.000183
nError	1	-0.007213

Example 42.2: Using Validation and Cross Validation

This example shows how you can use both test set and cross validation to monitor and control variable selection. It also demonstrates the use of split classification variables.

The following statements produce analysis and test data sets. Note that the same statements are used to generate the observations that are randomly assigned for analysis and test roles in the ratio of approximately two to one.

```
data analysisData testData;
   drop i j c3Num;
   length c3$ 7;

   array x{20} x1-x20;

   do i=1 to 1500;
      do j=1 to 20;
         x{j} = ranuni(1);
      end;

      c1 = 1 + mod(i,8);
      c2 = ranbin(1,3,.6);

      if      i < 50   then do; c3 = 'tiny';     c3Num=1;end;
      else if i < 250  then do; c3 = 'small';    c3Num=1;end;
      else if i < 600  then do; c3 = 'average';  c3Num=2;end;
      else if i < 1200 then do; c3 = 'big';      c3Num=3;end;
      else                  do; c3 = 'huge';     c3Num=5;end;

      y = 10 + x1 + 2*x5 + 3*x10 + 4*x20  + 3*x1*x7 + 8*x6*x7
             + 5*(c1=3)*c3Num + 8*(c1=7)  + 5*rannor(1);

      if ranuni(1) < 2/3 then output analysisData;
                         else output testData;
  end;
run;
```

Suppose you suspect that the dependent variable depends on both main effects and two-way interactions. You can use the following statements to select a model:

```
ods graphics on;

proc glmselect data=analysisData testdata=testData
               seed=1 plots(stepAxis=number)=(criterionPanel ASEPlot);
   partition fraction(validate=0.5);
   class c1 c2 c3(order=data);
   model y =  c1|c2|c3|x1|x2|x3|x4|x5|x5|x6|x7|x8|x9|x10
             |x11|x12|x13|x14|x15|x16|x17|x18|x19|x20 @2
           / selection=stepwise(choose = validate
                                select = sl)
             hierarchy=single stb;
run;
```

Note that a TESTDATA= data set is named in the PROC GLMSELECT statement and that a PARTITION statement is used to randomly assign half the observations in the analysis data set for model validation and the rest for model training. You find details about the number of observations used for each role in the number of observations tables shown in Output 42.2.1.

Output 42.2.1 Number of Observations Tables

```
                    The GLMSELECT Procedure

              Observation Profile for Analysis Data

        Number of Observations Read                       1010
        Number of Observations Used                       1010
        Number of Observations Used for Training           510
        Number of Observations Used for Validation         500
```

The "Class Level Information" and "Dimensions" tables are shown in Output 42.2.2. The "Dimensions" table shows that at each step of the selection process, 278 effects are considered as candidates for entry or removal. Since several of these effects have multilevel classification variables as members, there are 661 parameters.

Output 42.2.2 Class Level Information and Problem Dimensions

```
                    Class Level Information

        Class      Levels    Values

        c1              8    1 2 3 4 5 6 7 8
        c2              4    0 1 2 3
        c3              5    tiny small average big huge

                        Dimensions

                  Number of Effects       278
                  Number of Parameters    661
```

The model statement options request stepwise selection with the default entry and stay significance levels used for both selecting entering and departing effects and stopping the selection method. The CHOOSE=VALIDATE suboption specifies that the selected model is chosen to minimize the predicted residual sum of squares when the models at each step are scored on the observations reserved for validation. The HIERARCHY=SINGLE option specifies that interactions can enter the model only if the corresponding main effects are already in the model, and that main effects cannot be dropped from the model if an interaction with such an effect is in the model. These settings are listed in the model information table shown in Output 42.2.3.

Output 42.2.3 Model Information

The GLMSELECT Procedure

Data Set	WORK.ANALYSISDATA
Test Data Set	WORK.TESTDATA
Dependent Variable	y
Selection Method	Stepwise
Select Criterion	Significance Level
Stop Criterion	Significance Level
Choose Criterion	Validation ASE
Entry Significance Level (SLE)	0.15
Stay Significance Level (SLS)	0.15
Effect Hierarchy Enforced	Single
Random Number Seed	1

The stop reason and stop details tables are shown in Output 42.2.4. Note that because the STOP= suboption of the SELECTION= option was not explicitly specified, the stopping criterion used is the selection criterion, namely significance level.

Output 42.2.4 Stop Details

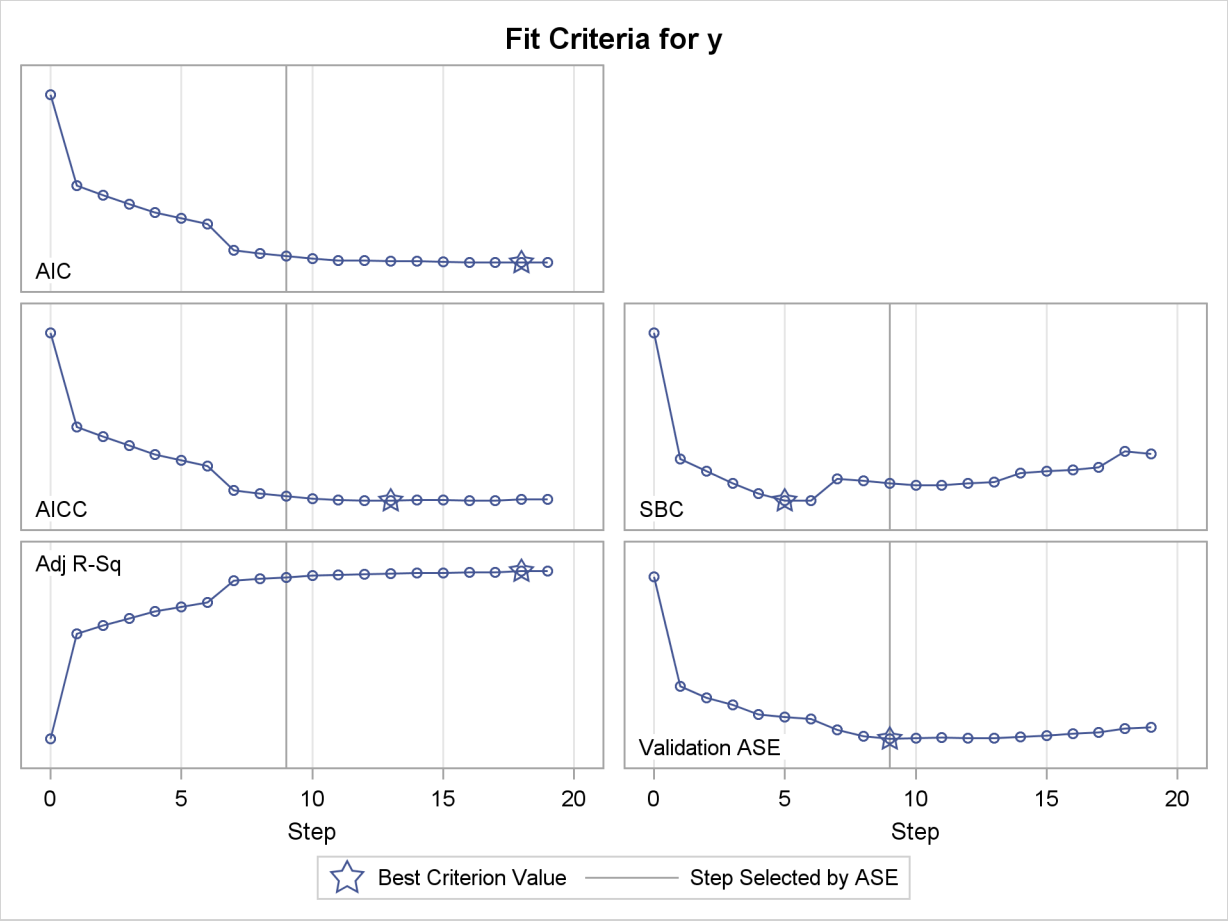
Selection stopped because the candidate for entry has SLE > 0.15 and the candidate for removal has SLS < 0.15.

Stop Details

Candidate For	Effect	Candidate Significance		Compare Significance	
Entry	x2*x5	0.1742	>	0.1500	(SLE)
Removal	x5*x10	0.0534	<	0.1500	(SLS)

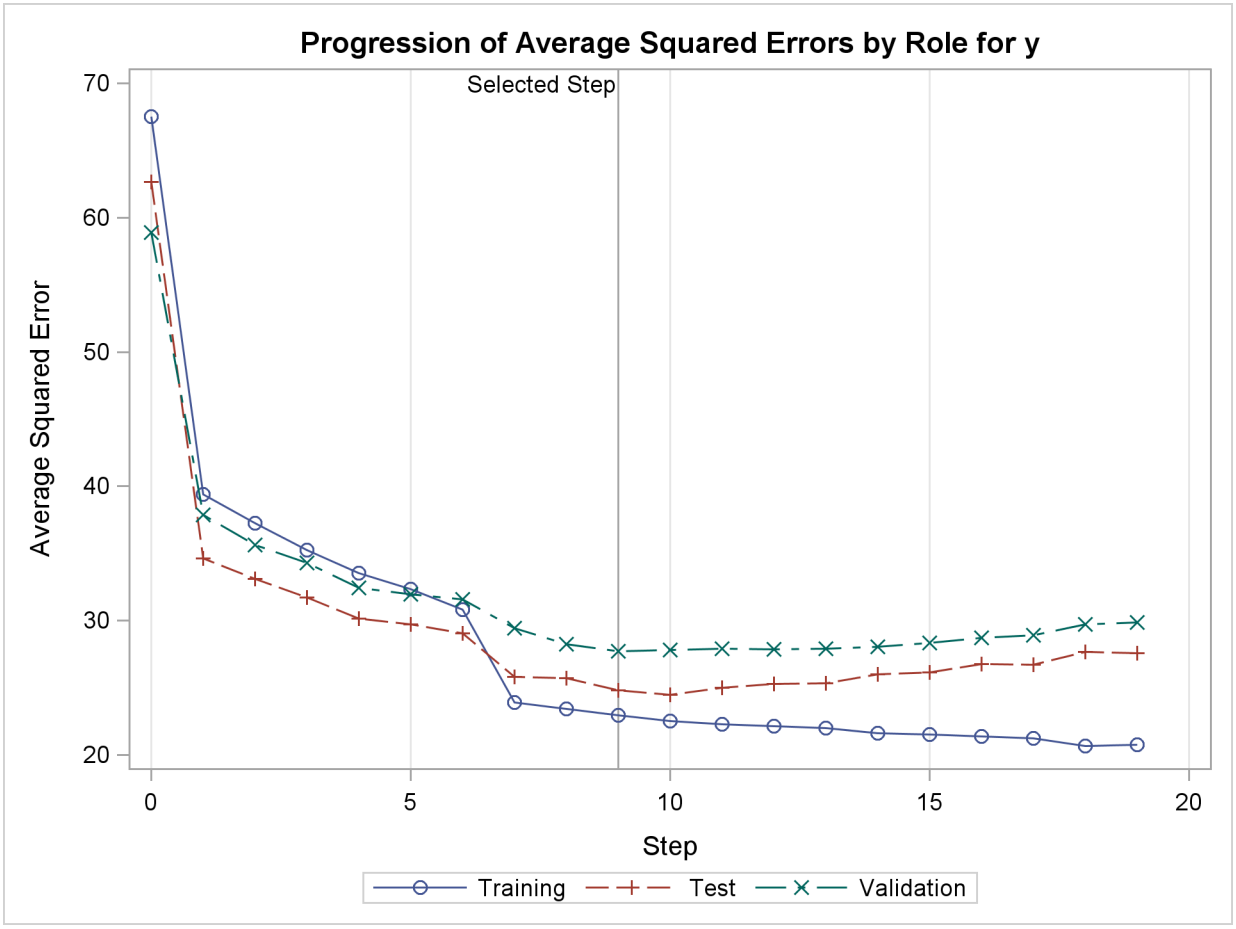
The criterion panel in Output 42.2.5 shows how the various fit criteria evolved as the stepwise selection method proceeded. Note that other than the ASE evaluated on the validation data, these criteria are evaluated on the training data. You see that the minimum of the validation ASE occurs at step 9, and hence the model at this step is selected.

Output 42.2.5 Criterion Panel



Output 42.2.6 shows how the average squared error (ASE) evolved on the training, validation, and test data. Note that while the ASE on the training data decreases monotonically, the errors on both the validation and test data start increasing beyond step 9. This indicates that models after step 9 are beginning to overfit the training data.

Output 42.2.6 Average Squared Errors



Output 42.2.7 shows the selected effects, analysis of variance, and fit statistics tables for the selected model. Output 42.2.8 shows the parameter estimates table.

Output 42.2.7 Selected Model Details

The GLMSELECT Procedure
Selected Model

The selected model, based on Validation ASE, is the model at Step 9.

Effects: Intercept c1 c3 c1*c3 x1 x5 x6 x7 x10 x20

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value
Model	44	22723	516.43621	20.49
Error	465	11722	25.20856	
Corrected Total	509	34445		

Root MSE	5.02081
Dependent Mean	21.09705
R-Square	0.6597
Adj R-Sq	0.6275
AIC	2200.75319
AICC	2210.09228
SBC	1879.30167
ASE (Train)	22.98427
ASE (Validate)	27.71105
ASE (Test)	24.82947

Output 42.2.8 Parameter Estimates

Parameter Estimates

Parameter		DF	Estimate	Standardized Estimate	Standard Error	t Value
Intercept		1	6.867831	0	1.524446	4.51
c1	1	1	0.226602	0.008272	2.022069	0.11
c1	2	1	-1.189623	-0.048587	1.687644	-0.70
c1	3	1	25.968930	1.080808	1.693593	15.33
c1	4	1	1.431767	0.054892	1.903011	0.75
c1	5	1	1.972622	0.073854	1.664189	1.19
c1	6	1	-0.094796	-0.004063	1.898700	-0.05
c1	7	1	5.971432	0.250037	1.846102	3.23
c1	8	0	0	0	.	.
c3	tiny	1	-2.919282	-0.072169	2.756295	-1.06
c3	small	1	-4.635843	-0.184338	2.218541	-2.09
c3	average	1	0.736805	0.038247	1.793059	0.41
c3	big	1	-1.078463	-0.063580	1.518927	-0.71
c3	huge	0	0	0	.	.
c1*c3	1 tiny	1	-2.449964	-0.018632	4.829146	-0.51
c1*c3	1 small	1	5.265031	0.069078	3.470382	1.52
c1*c3	1 average	1	-3.489735	-0.064365	2.850381	-1.22
c1*c3	1 big	1	0.725263	0.017929	2.516502	0.29
c1*c3	1 huge	0	0	0	.	.
c1*c3	2 tiny	1	5.455122	0.050760	4.209507	1.30
c1*c3	2 small	1	7.439196	0.131499	2.982411	2.49
c1*c3	2 average	1	-0.739606	-0.014705	2.568876	-0.29
c1*c3	2 big	1	3.179351	0.078598	2.247611	1.41
c1*c3	2 huge	0	0	0	.	.
c1*c3	3 tiny	1	-19.266847	-0.230989	3.784029	-5.09
c1*c3	3 small	1	-15.578909	-0.204399	3.266216	-4.77
c1*c3	3 average	1	-18.119398	-0.395770	2.529578	-7.16
c1*c3	3 big	1	-10.650012	-0.279796	2.205331	-4.83
c1*c3	3 huge	0	0	0	.	.
c1*c3	4 tiny	0	0	0	.	.
c1*c3	4 small	1	4.432753	0.047581	3.677008	1.21
c1*c3	4 average	1	-3.976295	-0.091632	2.625564	-1.51
c1*c3	4 big	1	-1.306998	-0.033003	2.401064	-0.54
c1*c3	4 huge	0	0	0	.	.
c1*c3	5 tiny	1	6.714186	0.062475	4.199457	1.60
c1*c3	5 small	1	1.565637	0.022165	3.182856	0.49
c1*c3	5 average	1	-4.286085	-0.068668	2.749142	-1.56
c1*c3	5 big	1	-2.046468	-0.045949	2.282735	-0.90
c1*c3	5 huge	0	0	0	.	.
c1*c3	6 tiny	1	5.135111	0.039052	4.754845	1.08
c1*c3	6 small	1	4.442898	0.081945	3.079524	1.44
c1*c3	6 average	1	-2.287870	-0.056559	2.601384	-0.88
c1*c3	6 big	1	1.598086	0.043542	2.354326	0.68
c1*c3	6 huge	0	0	0	.	.
c1*c3	7 tiny	1	1.108451	0.010314	4.267509	0.26

Output 42.2.8 *continued*

Parameter Estimates

Parameter		DF	Estimate	Standardized Estimate	Standard Error	t Value
c1*c3	7 small	1	7.441059	0.119214	3.135404	2.37
c1*c3	7 average	1	1.796483	0.038106	2.630570	0.68
c1*c3	7 big	1	3.324160	0.095173	2.303369	1.44
c1*c3	7 huge	0	0	0	.	.
c1*c3	8 tiny	0	0	0	.	.
c1*c3	8 small	0	0	0	.	.
c1*c3	8 average	0	0	0	.	.
c1*c3	8 big	0	0	0	.	.
c1*c3	8 huge	0	0	0	.	.
x1		1	2.713527	0.091530	0.836942	3.24
x5		1	2.810341	0.098303	0.816290	3.44
x6		1	4.837022	0.167394	0.810402	5.97
x7		1	5.844394	0.207035	0.793775	7.36
x10		1	2.463916	0.087712	0.794599	3.10
x20		1	4.385924	0.156155	0.787766	5.57

The magnitudes of the standardized estimates and the t statistics of the parameters of the effect "c1" reveal that only levels "3" and "7" of this effect contribute appreciably to the model. This suggests that a more parsimonious model with similar or better predictive power might be obtained if parameters corresponding to the levels of "c1" are allowed to enter or leave the model independently. You request this with the SPLIT option in the CLASS statement as shown in the following statements:

```
proc glmselect data=analysisData testdata=testData
               seed=1 plots(stepAxis=number)=all;
   partition fraction(validate=0.5);
   class c1(split) c2 c3(order=data);
   model y =  c1|c2|c3|x1|x2|x3|x4|x5|x5|x6|x7|x8|x9|x10
             |x11|x12|x13|x14|x15|x16|x17|x18|x19|x20 @2
           / selection=stepwise(stop   = validate
                                select = sl)
             hierarchy=single;
   output out=outData;
run;
```

The "Class Level Information" and "Dimensions" tables are shown in Output 42.2.9. The "Dimensions" table shows that while the model statement specifies 278 effects, after splitting the parameters corresponding to the levels of "c1," there are 439 split effects that are considered for entry or removal at each step of the selection process. Note that the total number of parameters considered is not affected by the split option.

Output 42.2.9 Class Level Information and Problem Dimensions

```
                    The GLMSELECT Procedure

                    Class Level Information

        Class      Levels      Values

        c1              8 *    1 2 3 4 5 6 7 8
        c2              4      0 1 2 3
        c3              5      tiny small average big huge

                   * Associated Parameters Split

                          Dimensions

              Number of Effects                 278
              Number of Effects after Splits    439
              Number of Parameters              661
```

The stop reason and stop details tables are shown in Output 42.2.10. Since the validation ASE is specified as the stopping criterion, the selection stops at step 11, where the validation ASE achieves a local minimum and the model at this step is the selected model.

Output 42.2.10 Stop Details

```
Selection stopped at a local minimum of the residual sum of squares of the
validation data.

                                Stop Details

        Candidate                   Candidate         Compare
        For          Effect    Validation ASE         Validation ASE

        Entry        x18              25.9851    >    25.7462
        Removal      x6*x7            25.7611    >    25.7462
```

You find details of the selected model in Output 42.2.11. The list of selected effects confirms that parameters corresponding to levels “3” and “7” only of “c1” are in the selected model. Notice that the selected model with classification variable “c1” split contains 18 parameters, whereas the selected model without splitting “c1” has 45 parameters. Furthermore, by comparing the fit statistics in Output 42.2.7 and Output 42.2.11, you see that this more parsimonious model has smaller prediction errors on both the validation and test data.

Output 42.2.11 Details of the Selected Model

The GLMSELECT Procedure
Selected Model

The selected model is the model at the last step (Step 11).

Effects: Intercept c1_3 c1_7 c3 c1_3*c3 x1 x5 x6 x7 x6*x7 x10 x20

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value
Model	17	22111	1300.63200	51.88
Error	492	12334	25.06998	
Corrected Total	509	34445		

Root MSE	5.00699
Dependent Mean	21.09705
R-Square	0.6419
Adj R-Sq	0.6295
AIC	2172.72685
AICC	2174.27787
SBC	1736.94624
ASE (Train)	24.18515
ASE (Validate)	25.74617
ASE (Test)	22.57297

When you use a PARTITION statement to subdivide the analysis data set, an output data set created with the OUTPUT statement contains a variable named "_ROLE_" that shows the role each observation was assigned to. See the section "OUTPUT Statement" on page 2714 and the section "Using Validation and Test Data" on page 2739 for additional details.

The following statements use PROC PRINT to produce Output 42.2.12, which shows the first five observations of the outData data set.

```
proc print data=outData(obs=5);
run;
```

Output 42.2.12 Output Data Set with _ROLE_ Variable

Obs	c3	x1	x2	x3	x4	x5	x6	x7	x8
1	tiny	0.18496	0.97009	0.39982	0.25940	0.92160	0.96928	0.54298	0.53169
2	tiny	0.47579	0.84499	0.63452	0.59036	0.58258	0.37701	0.72836	0.50660
3	tiny	0.51132	0.43320	0.17611	0.66504	0.40482	0.12455	0.45349	0.19955
4	tiny	0.42071	0.07174	0.35849	0.71143	0.18985	0.14797	0.56184	0.27011
5	tiny	0.42137	0.03798	0.27081	0.42773	0.82010	0.84345	0.87691	0.26722

Obs	x9	x10	x11	x12	x13	x14	x15	x16	x17
1	0.04979	0.06657	0.81932	0.52387	0.85339	0.06718	0.95702	0.29719	0.27261
2	0.93121	0.92912	0.58966	0.29722	0.39104	0.47243	0.67953	0.16809	0.16653
3	0.57484	0.73847	0.43981	0.04937	0.52238	0.34337	0.02271	0.71289	0.93706
4	0.32520	0.56918	0.04259	0.43921	0.91744	0.52584	0.73182	0.90522	0.57600
5	0.30602	0.39705	0.34905	0.76593	0.54340	0.61257	0.55291	0.73591	0.37186

Obs	x18	x19	x20	c1	c2	y	_ROLE_	p_y
1	0.68993	0.97676	0.22651	2	1	11.4391	VALIDATE	18.5069
2	0.87110	0.29879	0.93464	3	1	31.4596	TRAIN	26.2188
3	0.44599	0.94694	0.71290	4	3	16.4294	VALIDATE	17.0979
4	0.18794	0.33133	0.69887	5	3	15.4815	VALIDATE	16.1567
5	0.64565	0.55718	0.87504	6	2	26.0023	TRAIN	24.6358

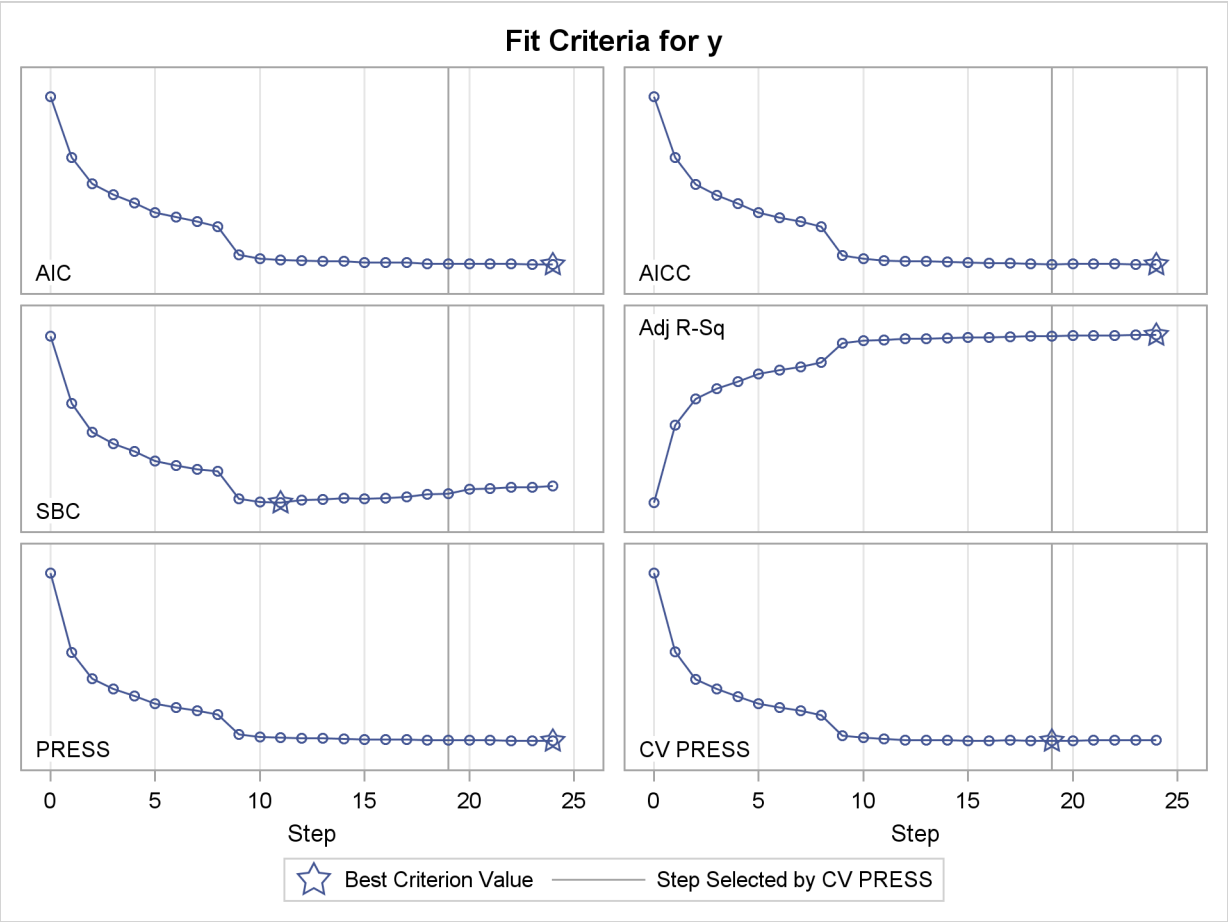
Cross validation is often used to assess the predictive performance of a model, especially for when you do not have enough observations for test set validation. See the section "Cross Validation" on page 2741 for further details. The following statements provide an example where cross validation is used as the CHOOSE= criterion.

```
proc glmselect data=analysisData testdata=testData
               plots(stepAxis=number)=(criterionPanel ASEPlot);
   class c1(split) c2 c3(order=data);
   model y =  c1|c2|c3|x1|x2|x3|x4|x5|x5|x6|x7|x8|x9|x10
             |x11|x12|x13|x14|x15|x16|x17|x18|x19|x20 @2
           / selection = stepwise(choose = cv
                                  select = sl)
             stats     = press
             cvMethod  = split(5)
             cvDetails = all
             hierarchy = single;
    output out=outData;
run;
```

The CVMETHOD=SPLIT(5) option in the MODEL statement requests five-fold cross validation with the five subsets consisting of observations $\{1, 6, 11, \ldots\}$, $\{2, 7, 12, \ldots\}$, and so on. The STATS=PRESS option requests that the leave-one-out cross validation predicted residual sum of squares (PRESS) also be computed and displayed at each step, even though this statistic is not used in the selection process.

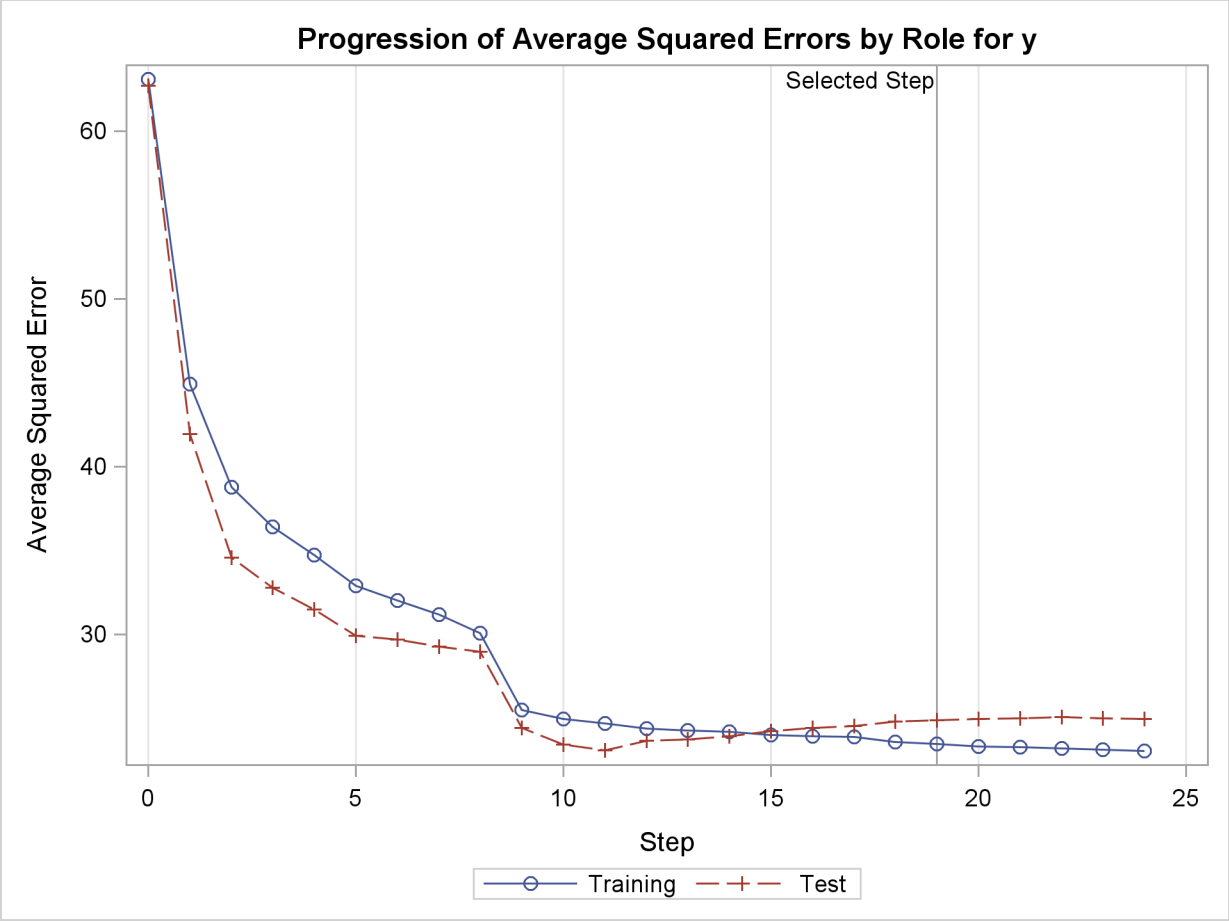
Output 42.2.13 shows how several fit statistics evolved as the selection process progressed. The five-fold CV PRESS statistic achieves its minimum at step 19. Note that this gives a larger model than was selected when the stopping criterion was determined using validation data. Furthermore, you see that the PRESS statistic has not achieved its minimum within 25 steps, so an even larger model would have been selected based on leave-one-out cross validation.

Output 42.2.13 Criterion Panel



Output 42.2.14 shows how the average squared error compares on the test and training data. Note that the ASE error on the test data achieves a local minimum at step 11 and is already slowly increasing at step 19, which corresponds to the selected model.

Output 42.2.14 Average Squared Error Plot



The CVDETAILS=ALL option in the MODEL statement requests the "Cross Validation Details" table in Output 42.2.15 and the cross validation parameter estimates that are included in the "Parameter Estimates" table in Output 42.2.16. For each cross validation index, the predicted residual sum of squares on the observations omitted is shown in the "Cross Validation Details" table and the parameter estimates of the corresponding model are included in the "Parameter Estimates" table. By default, these details are shown for the selected model, but you can request this information at every step with the DETAILS= option in the MODEL statement. You use the "_CVINDEX_" variable in the output data set shown in Output 42.2.17 to find out which observations in the analysis data are omitted for each cross validation fold.

Output 42.2.15 Breakdown of CV Press Statistic by Fold

Cross Validation Details

Index	---Observations--- Fitted	Left Out	CV PRESS
1	808	202	5059.7375
2	808	202	4278.9115
3	808	202	5598.0354
4	808	202	4950.1750
5	808	202	5528.1846
Total			25293.5024

Output 42.2.16 Cross Validation Parameter Estimates

Parameter Estimates

Parameter		Cross Validation Estimates 1	2	3	4	5
Intercept		10.7617	10.1200	9.0254	13.4164	12.3352
c1_3		28.2715	27.2977	27.0696	28.6835	27.8070
c1_7		7.6530	7.6445	7.9257	7.4217	7.6862
c3	tiny	-3.1103	-4.4041	-5.1793	-8.4131	-7.2096
c3	small	2.2039	1.5447	1.0121	-0.3998	1.4927
c3	average	0.3021	-1.3939	-1.2201	-3.3407	-2.1467
c3	big	-0.9621	-1.2439	-1.6092	-3.7666	-3.4389
c3	huge	0	0	0	0	0
c1_3*c3	tiny	-21.9104	-21.7840	-22.0173	-22.6066	-21.9791
c1_3*c3	small	-20.8196	-20.2725	-19.5850	-20.4515	-20.7586
c1_3*c3	average	-16.8500	-15.1509	-15.0134	-15.3851	-13.4339
c1_3*c3	big	-12.7212	-12.1554	-12.0354	-12.3282	-13.0174
c1_3*c3	huge	0	0	0	0	0
x1		0.9238	1.7286	2.5976	-0.2488	1.2093
x1*c3	tiny	-1.5819	-1.1748	-3.2523	-1.7016	-2.7624
x1*c3	small	-3.7669	-3.2984	-2.9755	-1.8738	-4.0167
x1*c3	average	2.2253	2.4489	1.5675	4.0948	2.0159
x1*c3	big	0.9222	0.5330	0.7960	2.6061	1.2694
x1*c3	huge	0	0	0	0	0
x5		-1.3562	0.5639	0.3022	-0.4700	-2.5063
x6		-0.9165	-3.2944	-1.2163	-2.2063	-0.5696
x7		5.2295	5.3015	6.2526	4.1770	5.8364
x6*x7		6.4211	7.5644	6.1182	7.0020	5.8730
x10		1.9591	1.4932	0.7196	0.6504	-0.3989
x5*x10		3.6058	1.7274	4.3447	2.4388	3.8967
x15		-0.0079	0.6896	1.6811	0.0136	0.1799
x15*c1_3		-3.5022	-2.7963	-2.6003	-4.2355	-4.7546
x7*x15		-5.1438	-5.8878	-5.9465	-3.6155	-5.3337
x18		-2.1347	-1.5656	-2.4226	-4.0592	-1.4985
x18*c3	tiny	2.2988	1.1931	2.6491	6.1615	5.6204
x18*c3	small	4.6033	3.2359	4.4183	5.5923	1.7270
x18*c3	average	-2.3712	-2.5392	-0.6361	-1.1729	-1.6481
x18*c3	big	2.3160	1.4654	2.7683	3.0487	2.5768
x18*c3	huge	0	0	0	0	0
x6*x18		3.0716	4.2036	4.1354	4.9196	2.7165
x20		4.1229	4.5773	4.5774	4.6555	4.2655

The following statements display the first eight observations in the outData data set.

```
proc print data=outData(obs=8);
run;
```

Output 42.2.17 First Eight Observations in the Output Data Set

Obs	c3	x1	x2	x3	x4	x5	x6	x7	x8
1	tiny	0.18496	0.97009	0.39982	0.25940	0.92160	0.96928	0.54298	0.53169
2	tiny	0.47579	0.84499	0.63452	0.59036	0.58258	0.37701	0.72836	0.50660
3	tiny	0.51132	0.43320	0.17611	0.66504	0.40482	0.12455	0.45349	0.19955
4	tiny	0.42071	0.07174	0.35849	0.71143	0.18985	0.14797	0.56184	0.27011
5	tiny	0.42137	0.03798	0.27081	0.42773	0.82010	0.84345	0.87691	0.26722
6	tiny	0.81722	0.65822	0.02947	0.85339	0.36285	0.37732	0.51054	0.71194
7	tiny	0.19480	0.81673	0.08548	0.18376	0.33264	0.70558	0.92761	0.29642
8	tiny	0.04403	0.51697	0.68884	0.45333	0.83565	0.29745	0.40325	0.95684

Obs	x9	x10	x11	x12	x13	x14	x15	x16	x17
1	0.04979	0.06657	0.81932	0.52387	0.85339	0.06718	0.95702	0.29719	0.27261
2	0.93121	0.92912	0.58966	0.29722	0.39104	0.47243	0.67953	0.16809	0.16653
3	0.57484	0.73847	0.43981	0.04937	0.52238	0.34337	0.02271	0.71289	0.93706
4	0.32520	0.56918	0.04259	0.43921	0.91744	0.52584	0.73182	0.90522	0.57600
5	0.30602	0.39705	0.34905	0.76593	0.54340	0.61257	0.55291	0.73591	0.37186
6	0.37533	0.22954	0.68621	0.55243	0.58182	0.17472	0.04610	0.64380	0.64545
7	0.22404	0.14719	0.59064	0.46326	0.41860	0.25631	0.23045	0.08034	0.43559
8	0.42194	0.78079	0.33106	0.17210	0.91056	0.26897	0.95602	0.13720	0.27190

Obs	x18	x19	x20	c1	c2	y	_CVINDEX_	p_y
1	0.68993	0.97676	0.22651	2	1	11.4391	1	18.1474
2	0.87110	0.29879	0.93464	3	1	31.4596	2	24.7930
3	0.44599	0.94694	0.71290	4	3	16.4294	3	16.5752
4	0.18794	0.33133	0.69887	5	3	15.4815	4	14.7605
5	0.64565	0.55718	0.87504	6	2	26.0023	5	24.7479
6	0.09317	0.62008	0.07845	7	1	16.6503	1	21.4444
7	0.67020	0.42272	0.49827	1	1	14.0342	2	20.9661
8	0.55692	0.65825	0.68465	2	3	14.9830	3	17.5644

This example demonstrates the usefulness of effect selection when you suspect that interactions of effects are needed to explain the variation in your dependent variable. Ideally, a priori knowledge should be used to decide what interactions to allow, but in some cases this information might not be available. Simply fitting a least squares model allowing all interactions produces a model that overfits your data and generalizes very poorly.

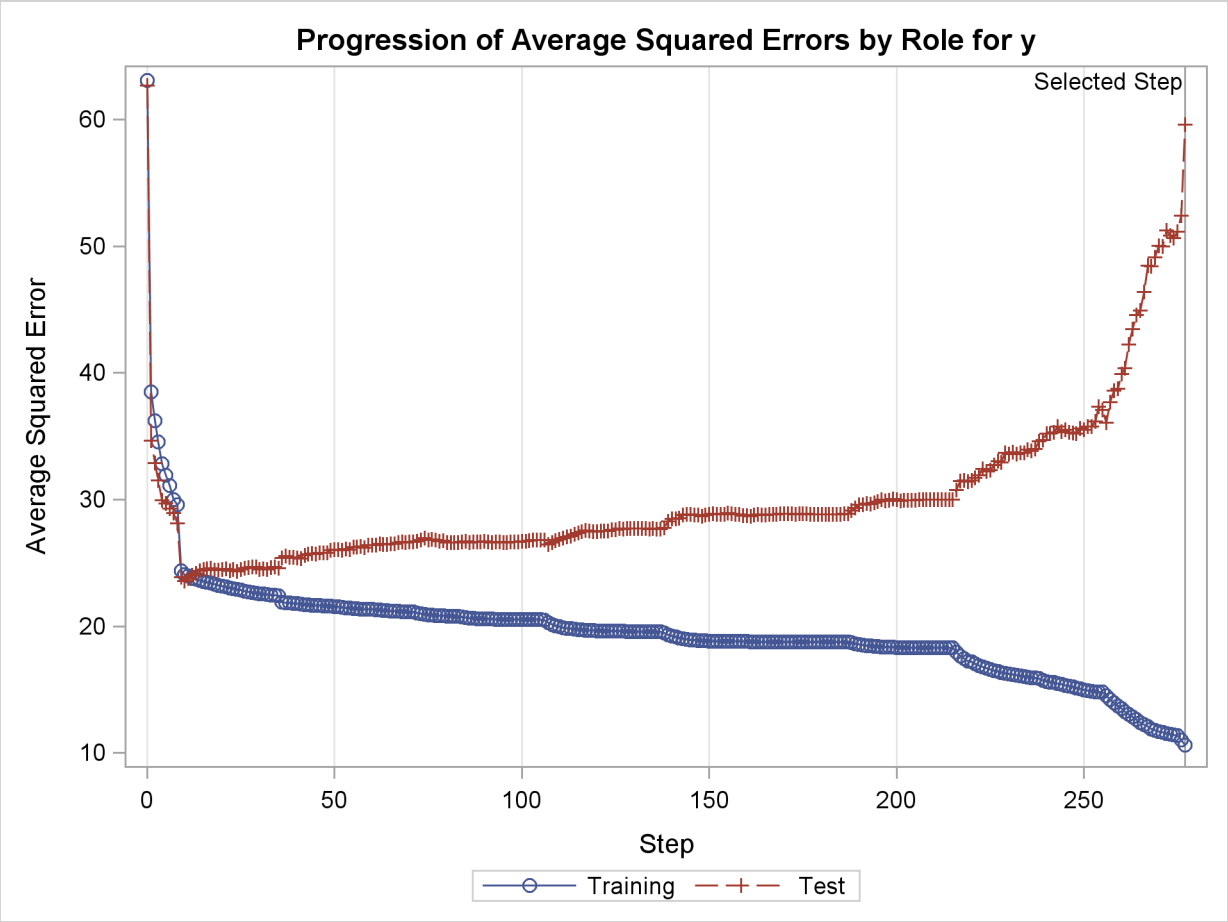
The following statements use forward selection with selection based on the SBC criterion, which is the default selection criterion. At each step, the effect whose addition to the model yields the smallest SBC value is added. The STOP=NONE suboption specifies that this process continue even when the SBC statistic grows whenever an effect is added, and so it terminates at a full least squares model. The BUILDSSCP=FULL option is specified in a PERFORMANCE statement, since building the SSCP matrix incrementally is counterproductive in this case. See the section "BUILDSSCP=FULL" on page 2717 for details. Note that if all you are interested in is a full least squares model, then it is much more efficient to simply specify SELECTION=NONE in the

MODEL statement. However, in this example the aim is to add effects in roughly increasing order of explanatory power.

```
proc glmselect data=analysisData testdata=testData plots=ASEPlot;
   class c1 c2 c3(order=data);
   model y =  c1|c2|c3|x1|x2|x3|x4|x5|x5|x6|x7|x8|x9|x10
             |x11|x12|x13|x14|x15|x16|x17|x18|x19|x20 @2
           / selection=forward(stop=none)
             hierarchy=single;
   performance buildSSCP = full;
run;

ods graphics off;
```

The ASE plot shown in Output 42.2.18 clearly demonstrates the danger in overfitting the training data. As more insignificant effects are added to the model, the growth in test set ASE shows how the predictions produced by the resulting models worsen. This decline is particularly rapid in the latter stages of the forward selection, because the use of the SBC criterion results in insignificant effects with lots of parameters being added after insignificant effects with fewer parameters.

Output 42.2.18 Average Squared Error Plot



Example 42.3: Scatter Plot Smoothing by Selecting Spline Functions

This example shows how you can use model selection to perform scatter plot smoothing. It illustrates how you can use the experimental EFFECT statement to generate a large collection of B-spline basis functions from which a subset is selected to fit scatter plot data.

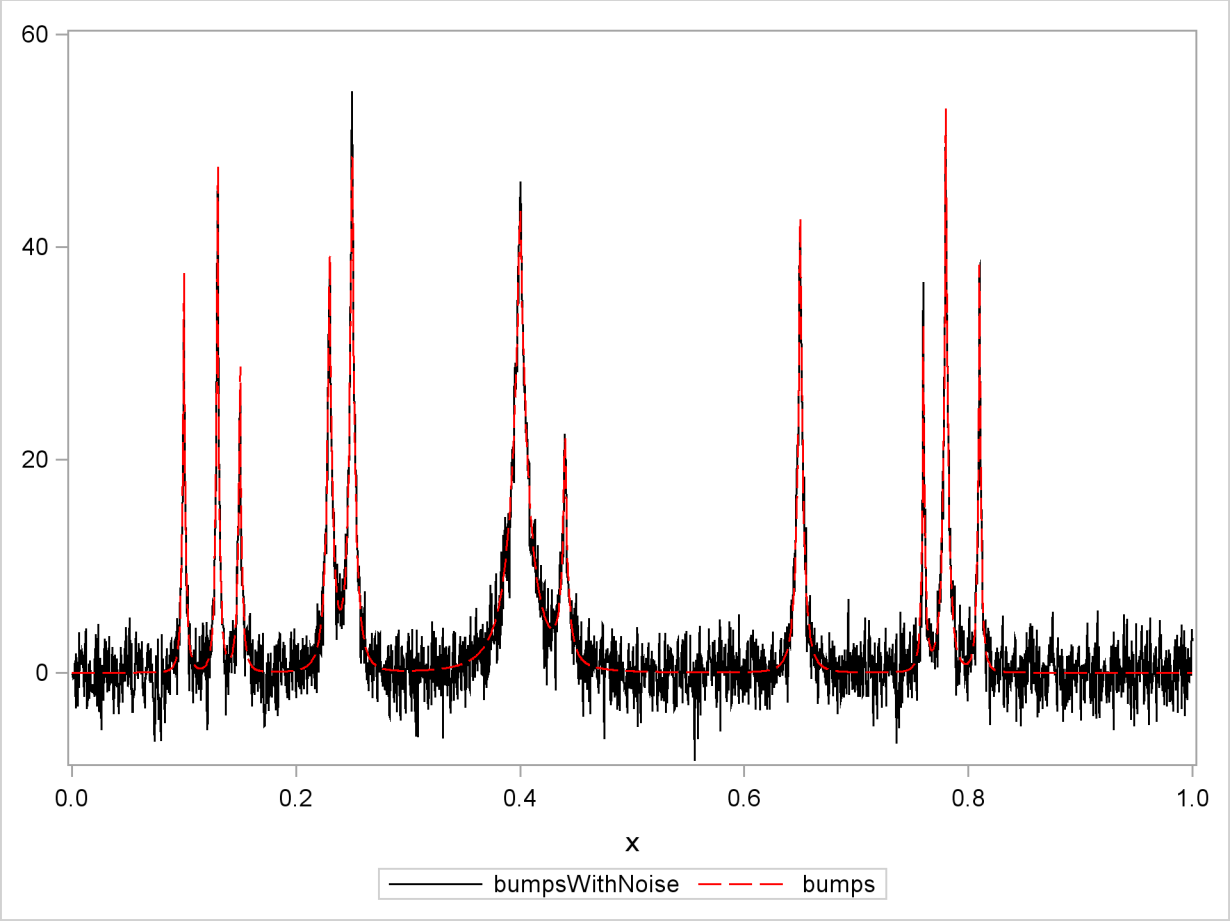
The data for this example come from a set of benchmarks developed by Donoho and Johnstone (1994) that have become popular in the statistics literature. The particular benchmark used is the "Bumps" functions to which random noise has been added to create the test data. The following DATA step, extracted from Sarle (2001), creates the data. The constants are chosen so that the noise-free data have a standard deviation of 7. The standard deviation of the noise is $\sqrt{5}$, yielding bumpsNoise with a signal-to-noise ratio of 3.13 ($7/\sqrt{5}$).

```
%let random=12345;

data DoJoBumps;
   keep x bumps bumpsWithNoise;

   pi = arcos(-1);

   do n=1 to 2048;
      x=(2*n-1)/4096;
      link compute;
      bumpsWithNoise=bumps+rannor(&random)*sqrt(5);
      output;
   end;
stop;

compute:
   array t(11) _temporary_ (.1 .13 .15 .23 .25 .4 .44 .65 .76 .78 .81);
   array b(11) _temporary_ (   4    5    3   4   5 4.2 2.1 4.3  3.1  5.1  4.2);
   array w(11) _temporary_ (.005 .005 .006 .01 .01 .03 .01 .01 .005 .008 .005);

   bumps=0;
   do i=1 to 11;
      bumps=bumps+b[i]*(1+abs((x-t[i])/w[i]))**-4;
   end;
   bumps=bumps*10.528514619;
return;
run;
```

The following statements use the SGPLOT procedure to produce the plot in Output 42.3.1. The plot shows the bumps function superimposed on the function with added noise.

```
proc sgplot data=DoJoBumps;
  yaxis display=(nolabel);
  series  x=x y=bumpsWithNoise/lineattrs=(color=black);
  series  x=x y=bumps/lineattrs=(color=red);
run;
```

Output 42.3.1 Donoho-Johnstone Bumps Function

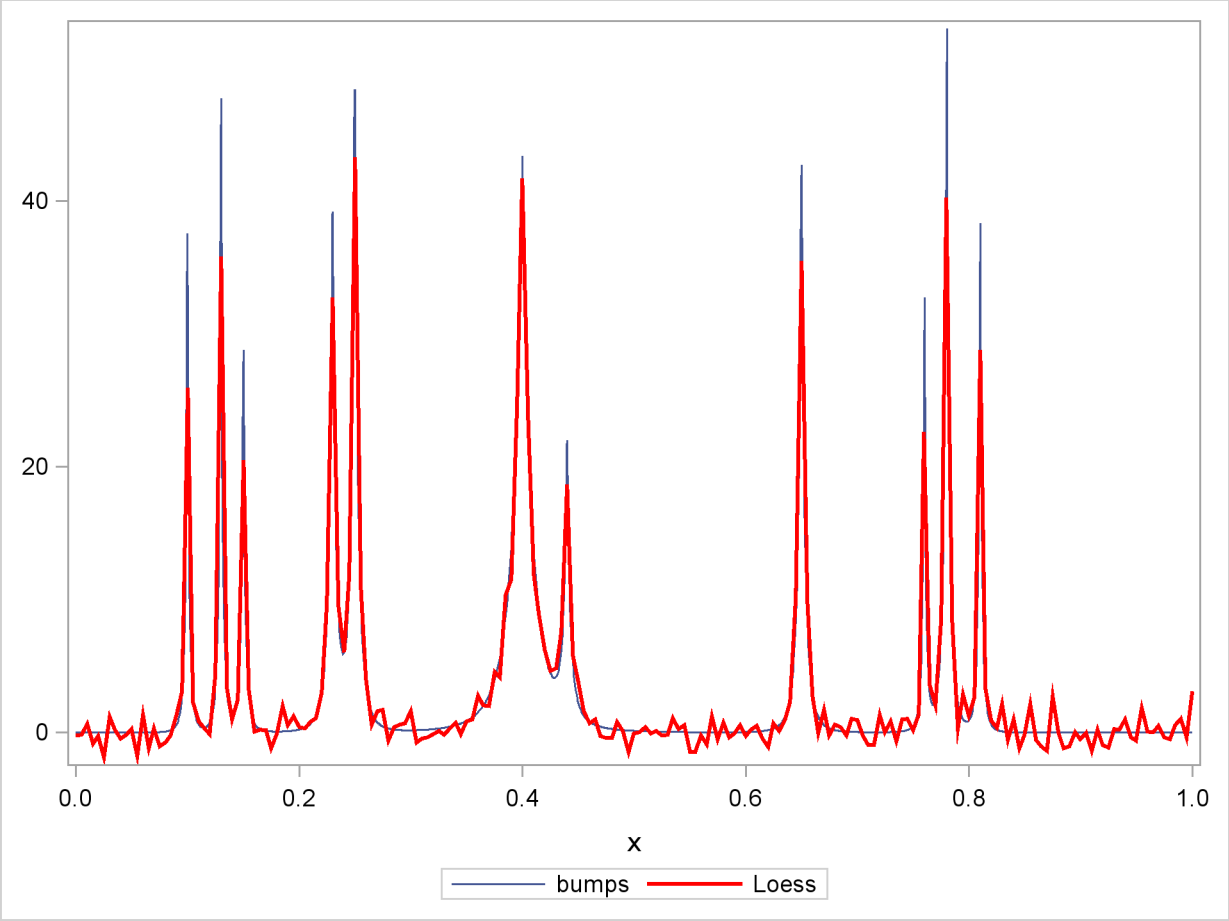


Suppose you want to smooth the noisy data to recover the underlying function. This problem is studied by Sarle (2001), who shows how neural nets can be used to perform the smoothing. The following statements use the LOESS statement in the SGPLOT procedure to show a loess fit superimposed on the noisy data (Output 42.3.2). (See Chapter 50, "The LOESS Procedure," for information about the loess method.)

```
proc sgplot data=DoJoBumps;
   yaxis display=(nolabel);
   series x=x y=bumps;
   loess  x=x y=bumpsWithNoise / lineattrs=(color=red) nomarkers;
run;
```

The algorithm selects a smoothing parameter that is small enough to enable bumps to be resolved. Because there is a single smoothing parameter that controls the number of points for all local fits, the loess method undersmooths the function in the intervals between the bumps.

Output 42.3.2 Loess Fit



Another approach to doing nonparametric fitting is to approximate the unknown underlying function as a linear combination of a set of basis functions. Once you specify the basis functions, then you can use least squares regression to obtain the coefficients of the linear combination. A problem with this approach is that for most data, you do not know a priori what set of basis functions to use. You need to supply a sufficiently rich set to enable the features in the data to be approximated. However, if you use too rich a set of functions, then this approach yields a fit that undersmooths the data and captures spurious features in the noise.

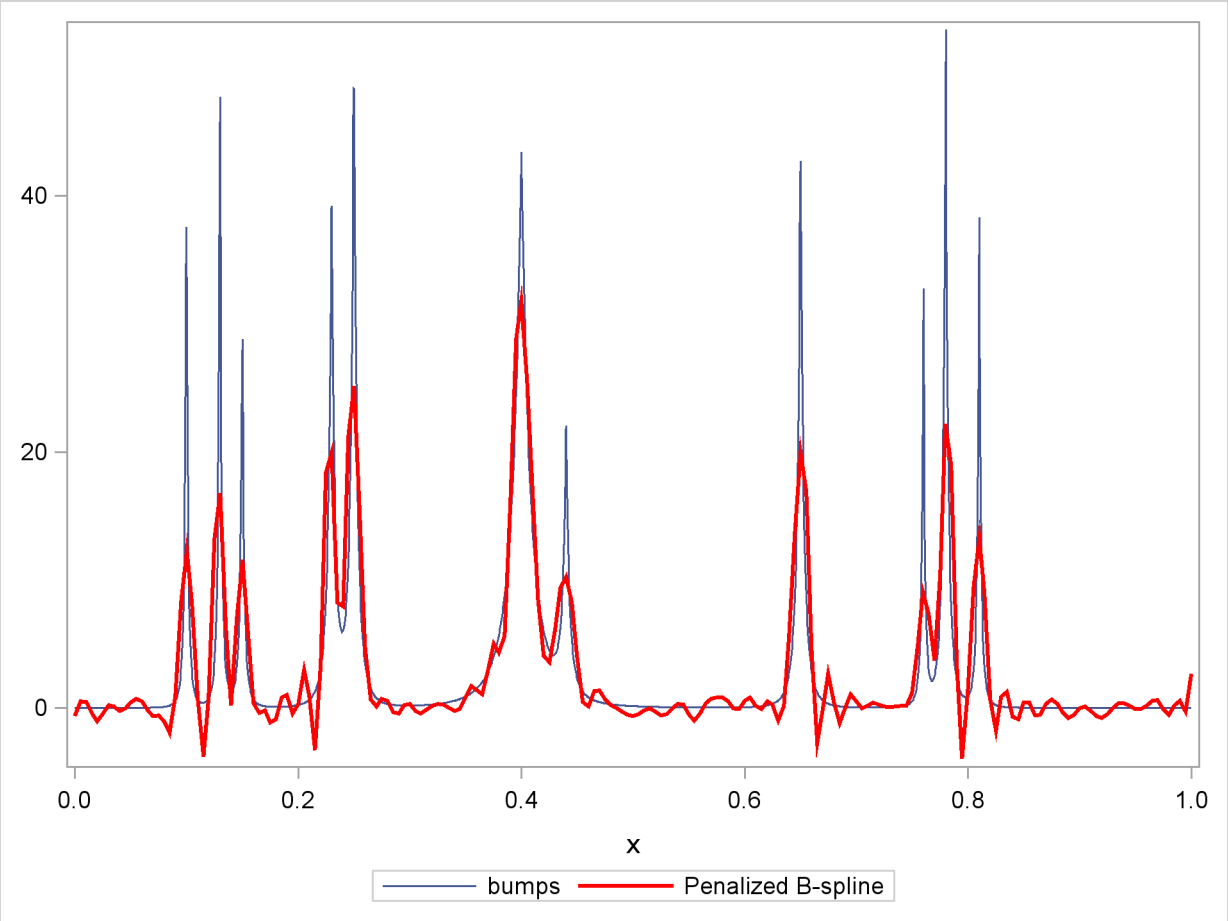
The penalized B-spline method (Eilers and Marx 1996) uses a basis of B-splines (see the section "Constructed Effects and the EFFECT Statement (Experimental)" on page 377 of Chapter 18, "Shared Concepts and Topics") corresponding to a large number of equally spaced knots as the set of approximating functions. To control the potential overfitting, their algorithm modifies the least squares objective function to include a penalty term that grows with the complexity of the fit.

The following statements use the PBSPLINE statement in the SGPLOT procedure to show a penalized B-spline fit superimposed on the noisy data (Output 42.3.3). See Chapter 90, "The TRANSREG Procedure," for details about the implementation of the penalized B-spline method.

```
proc sgplot data=DoJoBumps;
   yaxis display=(nolabel);
   series    x=x y=bumps;
   pbspline  x=x y=bumpsWithNoise /
               lineattrs=(color=red) nomarkers;
run;
```

As in the case of loess fitting, you see undersmoothing in the intervals between the bumps because there is only a single smoothing parameter that controls the overall smoothness of the fit.

Output 42.3.3 Penalized B-spline Fit



An alternative to using a smoothness penalty to control the overfitting is to use variable selection to obtain an appropriate subset of the basis functions. In order to be able to represent features in the data that occur at multiple scales, it is useful to select from B-spline functions defined on just a few knots to capture large scale features of the data as well as B-spline functions defined on many knots to capture fine details of the data. The following statements show how you can use PROC GLMSELECT to implement this strategy:

```
proc glmselect data=dojoBumps;
   effect spl = spline(x / knotmethod=multiscale(endscale=8)
                           split details);
   model bumpsWithNoise=spl;
   output out=out1 p=pBumps;
run;

proc sgplot data=out1;
   yaxis display=(nolabel);
   series    x=x y=bumps;
   series    x=x y=pBumps / lineattrs=(color=red);
run;
```

The KNOTMETHOD=MULTISCALE suboption of the **EFFECT spl = SPLINE** statement provides a convenient way to generate B-spline basis functions at multiple scales. The ENDSCALE=8 option requests that the finest scale use B-splines defined on 2^8 equally spaced knots in the interval $[0, 1]$. Because the cubic B-splines are nonzero over five adjacent knots, at the finest scale, the support of each B-spline basis function is an interval of length about 0.02 (5/256), enabling the bumps in the underlying data to be resolved. The default value is ENDSCALE=7. At this scale you will still be able to capture the bumps, but with less sharp resolution. For these data, using a value of ENDSCALE= greater than eight provides unneeded resolution, making it more likely that basis functions that fit spurious features in the noise are selected.

Output 42.3.4 shows the model information table. Since no options are specified in the MODEL statement, PROC GLMSELECT uses the stepwise method with selection and stopping based on the SBC criterion.

Output 42.3.4 Model Settings

The GLMSELECT Procedure

Data Set	WORK.DOJOBUMPS
Dependent Variable	bumpsWithNoise
Selection Method	Stepwise
Select Criterion	SBC
Stop Criterion	SBC
Effect Hierarchy Enforced	None

The DETAILS suboption in the EFFECT statement requests the display of spline knots and spline basis tables. These tables contain information about knots and basis functions at all scales. The results for scale four are shown in Output 42.3.5 and Output 42.3.6.

Output 42.3.5 Spline Knots

Knots for Spline Effect spl

Knot Number	Scale	Scale Knot Number	Boundary	x
40	4	1	*	−0.11735
41	4	2	*	−0.05855
42	4	3	*	0.00024414
43	4	4		0.05904
44	4	5		0.11783
45	4	6		0.17663
46	4	7		0.23542
47	4	8		0.29422
48	4	9		0.35301
49	4	10		0.41181
50	4	11		0.47060
51	4	12		0.52940
52	4	13		0.58819
53	4	14		0.64699
54	4	15		0.70578
55	4	16		0.76458
56	4	17		0.82337
57	4	18		0.88217
58	4	19		0.94096
59	4	20	*	0.99976
60	4	21	*	1.05855
61	4	22	*	1.11735

Output 42.3.6 Spline Details

Basis Details for Spline Effect spl

Column	Scale	Scale Column	Support		Support Knots
32	4	1	-0.11735	0.05904	1-4
33	4	2	-0.11735	0.11783	1-5
34	4	3	-0.05855	0.17663	2-6
35	4	4	0.00024414	0.23542	3-7
36	4	5	0.05904	0.29422	4-8
37	4	6	0.11783	0.35301	5-9
38	4	7	0.17663	0.41181	6-10
39	4	8	0.23542	0.47060	7-11
40	4	9	0.29422	0.52940	8-12
41	4	10	0.35301	0.58819	9-13
42	4	11	0.41181	0.64699	10-14
43	4	12	0.47060	0.70578	11-15
44	4	13	0.52940	0.76458	12-16
45	4	14	0.58819	0.82337	13-17
46	4	15	0.64699	0.88217	14-18
47	4	16	0.70578	0.94096	15-19
48	4	17	0.76458	0.99976	16-20
49	4	18	0.82337	1.05855	17-21
50	4	19	0.88217	1.11735	18-22
51	4	20	0.94096	1.11735	19-22

The "Dimensions" table in Output 42.3.7 shows that at each step of the selection process, 548 effects are considered as candidates for entry or removal. Note that although the MODEL statement specifies a single constructed effect spl, the SPLIT suboption causes each of the parameters in this constructed effect to be treated as an individual effect.

Output 42.3.7 Dimensions

Dimensions

Number of Effects	548
Number of Parameters	548

Output 42.3.8 shows the parameter estimates for the selected model. You can see that the selected model contains 31 B-spline basis functions and that all the selected B-spline basis functions are from scales four though eight. For example, the first basis function listed in the parameter estimates table is spl_S4:9—the 9th B-spline function at scale 4. You see from Output 42.3.6 that this function is nonzero on the interval $(0.29, 0.52)$.

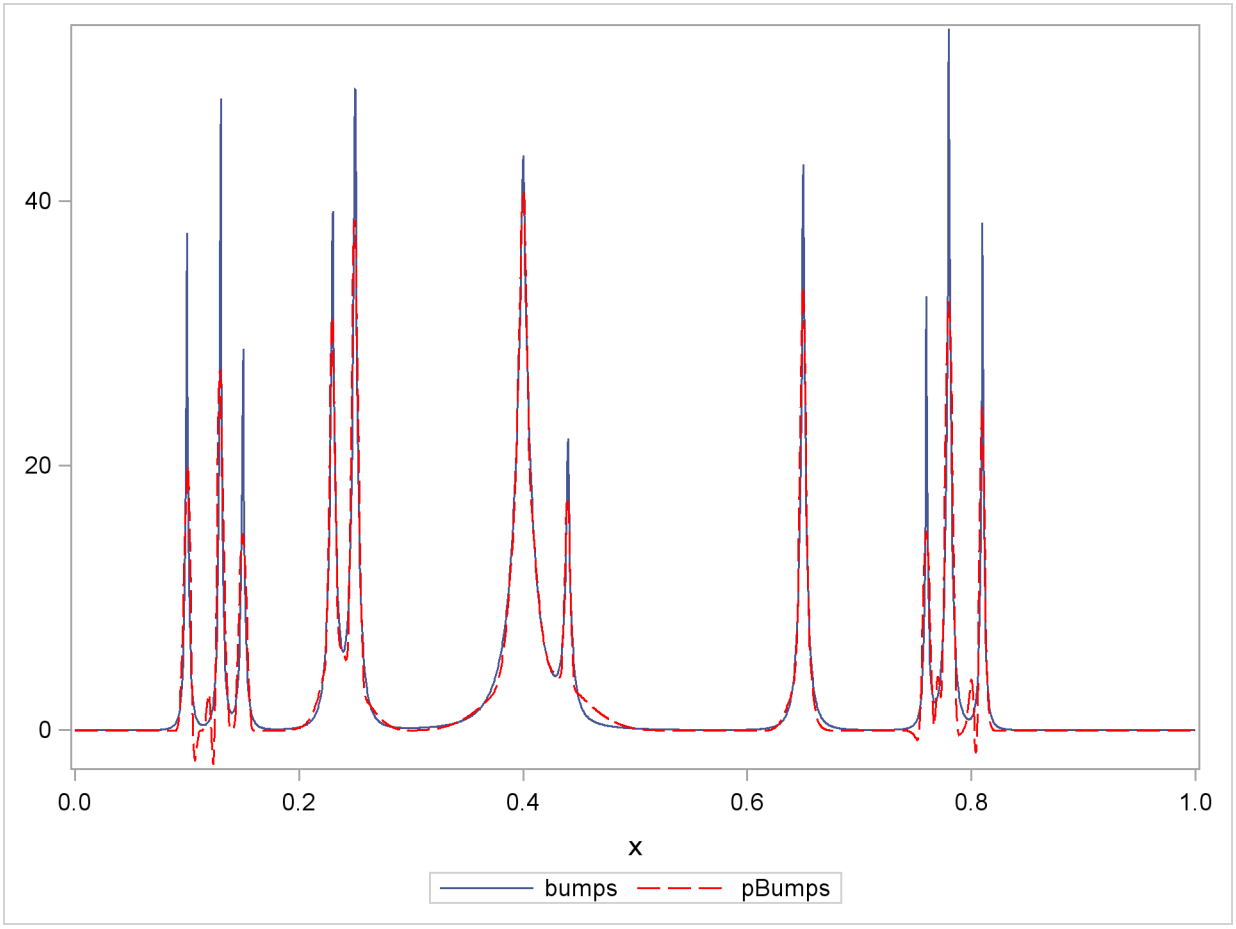
Output 42.3.8 Parameter Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	t Value
Intercept	1	-0.009039	0.077412	-0.12
spl_S4:9	1	7.070207	0.586990	12.04
spl_S5:10	1	5.323121	1.199824	4.44
spl_S6:17	1	5.222808	1.728910	3.02
spl_S6:28	1	24.562103	1.490639	16.48
spl_S6:44	1	4.930829	1.243552	3.97
spl_S6:52	1	-7.046308	2.487700	-2.83
spl_S7:86	1	9.592742	2.626471	3.65
spl_S7:106	1	16.268550	3.334015	4.88
spl_S8:27	1	10.626586	1.752152	6.06
spl_S8:28	1	27.882444	2.004520	13.91
spl_S8:29	1	-6.129939	1.752151	-3.50
spl_S8:33	1	5.855648	1.766912	3.31
spl_S8:34	1	-11.782303	2.092484	-5.63
spl_S8:35	1	38.705178	2.092486	18.50
spl_S8:36	1	13.823256	1.766916	7.82
spl_S8:40	1	15.975124	1.691679	9.44
spl_S8:41	1	14.898716	1.691679	8.81
spl_S8:61	1	37.441965	2.084375	17.96
spl_S8:66	1	47.484506	1.883409	25.21
spl_S8:67	1	16.811502	1.910358	8.80
spl_S8:104	1	11.098484	1.958676	5.67
spl_S8:105	1	26.704556	2.042735	13.07
spl_S8:115	1	21.102920	1.576185	13.39
spl_S8:169	1	36.572294	2.914521	12.55
spl_S8:197	1	20.869716	1.882529	11.09
spl_S8:198	1	16.210987	2.693183	6.02
spl_S8:200	1	13.113942	3.458187	3.79
spl_S8:202	1	38.463549	2.462314	15.62
spl_S8:203	1	34.164644	1.757908	19.43
spl_S8:209	1	-22.645471	3.598587	-6.29
spl_S8:210	1	29.024741	2.557567	11.35

The OUTPUT statement captures the predicted values in a data set named out1, and Output 42.3.9 shows a fit plot produced by PROC SGPLOT.

Output 42.3.9 Fit by Selecting B-splines



Example 42.4: Multimember Effects and the Design Matrix

This example shows how you can use multimember effects to build predictive models. It also demonstrates several features of the OUTDESIGN= option in the PROC GLMSELECT statement.

The simulated data for this example describe a two-week summer tennis camp. The tennis ability of each camper was assessed and ratings were assigned at the beginning and end of the camp. The camp consisted of supervised group instruction in the mornings with a number of different options in the afternoons. Campers could elect to participate in unsupervised practice and play. Some campers paid for one or more individual lessons from 30 to 90 minutes in length, focusing on forehand and backhand strokes and volleying. The aim of this example is to build a predictive model for the rating improvement of each camper based on the times the camper spent doing each activity and several other variables, including the age, gender, and initial rating of the camper.

The following statements produce the TennisCamp data set:

```
data TennisCamp;
   length forehandCoach $6 backhandCoach $6 volleyCoach $6 gender $1;
   input forehandCoach backhandCoach volleyCoach tLessons tPractice tPlay
         gender inRating nPastCamps age tForehand tBackhand tVolley
         improvement;
   label forehandCoach = "Forehand lesson coach"
         backhandCoach = "Backhand lesson coach"
         volleyCoach   = "Volley   lesson coach"
         tForehand     = "time (1/2 hours) of forehand lesson"
         tBackhand     = "time (1/2 hours) of backhand lesson"
         tVolley       = "time (1/2 hours) of volley lesson"
         tLessons      = "time (1/2 hours) of all lessons"
         tPractice     = "total practice time (hours)"
         tPlay         = "total play time (hours)"
         nPastCamps    = "Number of previous camps attended"
         age           = "age (years)"
         inRating      = "Rating at camp start"
         improvement   = "Rating improvement at end of camp";
datalines;
.        .        Tom      1   30   19   f   44   0   13   0   0   1    6
Greg     .        .        2   12   33   f   48   2   15   2   0   0   14
.        .        Mike     2   12   24   m   53   0   15   0   0   2   13
.        Mike     .        1   12   28   f   48   0   13   0   1   0   11

... more lines ...

.        .        .        0   12   38   m   47   1   15   0   0   0    8
Greg     Tom      Tom      6    3   41   m   48   2   15   2   1   3   19
.        Greg     Mike     5   30   16   m   52   0   13   0   2   3   18
;
```

A multimember effect (see the section "Constructed Effects and the EFFECT Statement (Experimental)" on page 377 of Chapter 18, "Shared Concepts and Topics") is appropriate for modeling the effect of coaches on the campers' improvement, because campers might have worked with multiple coaches. Furthermore, since the time a coach spent with each camper varies, it is

appropriate to use these times to weight each coach's contribution in the multimember effect. It is also important not to exclude campers from the analysis if they did not receive any individual instruction. You can accomplish all these goals by using a multimember effect defined as follows:

```
class forehandCoach backhandCoach volleyCoach;
effect coach = MM(forehandCoach backhandCoach volleyCoach/ noeffect
                  weight=(tForehand tBackhand tVolley));
```

Based on similar previous studies, it is known that the time spent practicing should not be included linearly, because there are diminishing returns and perhaps even counterproductive effects beyond about 25 hours. A spline effect with a single knot at 25 provides flexibility in modeling effect of practice time.

The following statements use PROC GLMSELECT to select effects for the model.

```
proc glmselect data=TennisCamp outdesign=designCamp;
   class forehandCoach backhandCoach volleyCoach gender;

   effect coach    = mm(forehandCoach backhandCoach volleyCoach / noeffect details
                        weight=(tForehand tBackhand tVolley));
   effect practice = spline(tPractice/knotmethod=list(25) details);

   model improvement = coach practice tLessons tPlay age gender
                       inRating nPastCamps;
run;
```

Output 42.4.1 shows the class level and MM level information. The levels of the constructed MM effect are the union of the levels of its constituent classification variables. The MM level information is not displayed by default—you request this table by specifying the DETAILS suboption in the relevant EFFECT statement.

Output 42.4.1 Levels of MM EFFECT Coach

```
                         The GLMSELECT Procedure

                         Class Level Information

           Class              Levels    Values

           forehandCoach           5    Bruna Elaine Greg Mike Tom
           backhandCoach           5    Bruna Elaine Greg Mike Tom
           volleyCoach             5    Andy Bruna Greg Mike Tom
           gender                  2    f m

                         The GLMSELECT Procedure

                     Level Details for MM Effect coach

                  Levels    Values

                       6    Andy Bruna Elaine Greg Mike Tom
```

Output 42.4.2 shows the parameter estimates for the selected model. You can see that the constructed multimember effect coach and the spline effect practice are both included in the selected model. All coaches provided benefit (all the parameters of the multimember effect coach are positive), with Greg and Mike being the most effective.

Output 42.4.2 Parameter Estimates

Parameter Estimates

Parameter		DF	Estimate	Standard Error	t Value
Intercept		1	0.379873	0.513431	0.74
coach	Andy	1	1.444370	0.318078	4.54
coach	Bruna	1	1.446063	0.110179	13.12
coach	Elaine	1	1.312290	0.281877	4.66
coach	Greg	1	3.042828	0.112256	27.11
coach	Mike	1	2.840728	0.121166	23.45
coach	Tom	1	1.248946	0.115266	10.84
practice	1	1	2.538938	1.015772	2.50
practice	2	1	3.837684	1.104557	3.47
practice	3	1	2.574775	0.930816	2.77
practice	4	1	-0.034747	0.717967	-0.05
practice	5	0	0	.	.
tPlay		1	0.139409	0.023043	6.05

Suppose you want to examine regression diagnostics for the selected model. PROC GLMSELECT does not support such diagnostics, so you might want to use the REG procedure to produce these diagnostics. You can overcome the difficulty that PROC REG does not support CLASS and EFFECT statements by using the OUTDESIGN= option in the PROC GLMSELECT statement to obtain the design matrix that you can use as an input data set for further analysis with other SAS procedures.

The following statements use PROC PRINT to produce Output 42.4.3, which shows the first five observations of the design matrix designCamp.

```
proc print data=designCamp(obs=5);
run;
```

Output 42.4.3 First Five Observations of the designCamp Data Set

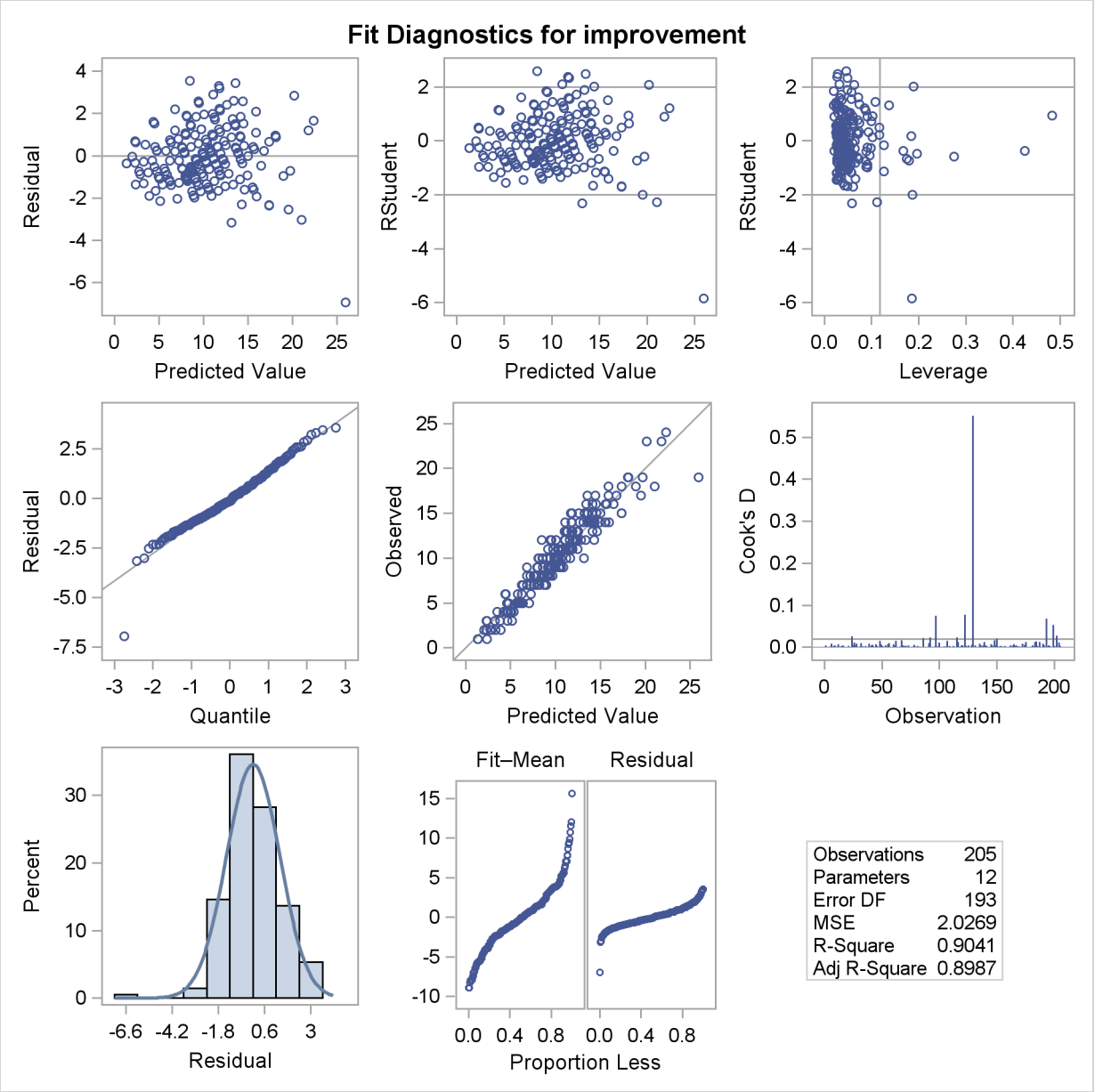
Obs	Intercept	coach_Andy	coach_Bruna	coach_Elaine	coach_Greg	coach_Mike	coach_Tom	tPlay	practice_1	practice_2	practice_3	practice_4	practice_5	improvement
1	1	0	0	0	0	0	1	19	0.00000	0.00136	0.05077	0.58344	0.36443	6
2	1	0	0	0	2	0	0	33	0.20633	0.50413	0.25014	0.03940	0.00000	14
3	1	0	0	0	0	2	0	24	0.20633	0.50413	0.25014	0.03940	0.00000	13
4	1	0	0	0	0	1	0	28	0.20633	0.50413	0.25014	0.03940	0.00000	11
5	1	0	2	0	0	0	0	34	0.16228	0.49279	0.29088	0.05405	0.00000	12

To facilitate specifying the columns of the design matrix corresponding to the selected model, you can use the macro variable named _GLSMOD that PROC GLMSELECT creates whenever you specify the OUTDESIGN= option. The following statements use PROC REG to produce a panel of regression diagnostics corresponding to the model selected by PROC GLMSELECT.

```
ods graphics on;

proc reg data=designCamp;
  model improvement = &_GLSMOD;
quit;

ods graphics off;
```

The regression diagnostics shown in Output 42.4.4 indicate a reasonable model. However, they also reveal the presence of one large outlier and several influential observations that you might want to investigate.

Output 42.4.4 Fit Diagnostics



Sometimes you might want to use subsets of the columns of the design matrix. In such cases, it might be convenient to produce a design matrix with generic names for the columns. You might also want a design matrix containing the columns corresponding to the full model that you specify in the MODEL statement. By default, the design matrix includes only the columns that correspond to effects in the selected model. The following statements show how to do this.

```
   proc glmselect data=TennisCamp
        outdesign(fullmodel prefix=parm names)=designCampGeneric;
   class forehandCoach backhandCoach volleyCoach gender;

   effect coach    = mm(forehandCoach backhandCoach volleyCoach / noeffect details
                        weight=(tForehand tBackhand tVolley));
   effect practice = spline(tPractice/knotmethod=list(25) details);

   model improvement = coach practice tLessons tPlay age gender
                       inRating nPastCamps;
run;
```

The PREFIX=parm suboption of the OUTDESIGN= option specifies that columns in the design matrix be given the prefix parm with a trailing index. The NAMES suboption requests the table in Output 42.4.5 that associates descriptive labels with the names of columns in the design matrix. Finally, the FULLMODEL suboption specifies that the design matrix include columns corresponding to all effects specified in the MODEL statement.

Output 42.4.5 Descriptive Names of Design Matrix Columns

The GLMSELECT Procedure
Selected Model

Parameter Names

Name	Parameter
parm1	Intercept
parm2	coach Andy
parm3	coach Bruna
parm4	coach Elaine
parm5	coach Greg
parm6	coach Mike
parm7	coach Tom
parm8	practice 1
parm9	practice 2
parm10	practice 3
parm11	practice 4
parm12	practice 5
parm13	tLessons
parm14	tPlay
parm15	age
parm16	gender f
parm17	gender m
parm18	inRating
parm19	nPastCamps

The following statements produce Output 42.4.6, displaying the first five observations of the designCampGeneric data set:

```
proc print data=designCampGeneric(obs=5);
run;
```

Output 42.4.6 First Five Observations of designCampGeneric Data Set

Obs	parm1	parm2	parm3	parm4	parm5	parm6	parm7	parm8	parm9	parm10	parm11	parm12	parm13	parm14	parm15	parm16	parm17	parm18	parm19	improve
1	1	0	0	0	0	0	1	0.00000	0.00136	0.05077	0.58344	0.36443	1	19	13	1	0	44	0	6
2	1	0	0	0	2	0	0	0.20633	0.50413	0.25014	0.03940	0.00000	2	33	15	1	0	48	2	14
3	1	0	0	0	0	2	0	0.20633	0.50413	0.25014	0.03940	0.00000	2	24	15	0	1	53	0	13
4	1	0	0	0	0	1	0	0.20633	0.50413	0.25014	0.03940	0.00000	1	28	13	1	0	48	0	11
5	1	0	2	0	0	0	0	0.16228	0.49279	0.29088	0.05405	0.00000	2	34	16	1	0	57	0	12

References

Burnham, K. P., and Anderson, D. R. (2002), *Model Selection and Multimodel Inference,* Second Edition, New York: Springer-Verlag.

Collier Books (1987), *The 1987 Baseball Encyclopedia Update,* New York: Macmillan.

Darlington, R. B. (1968), "Multiple Regression in Psychological Research and Practice," *Psychological Bulletin*, 69, 161–182.

Donoho, D. L. and Johnstone, I. M. (1994), "Ideal Spatial Adaptation via Wavelet Shrinkage," *Biometrika*, 81, 425–455.

Draper, N. R., Guttman, I., and Kanemasu, H. (1971), "The Distribution of Certain Regression Statistics," *Biometrika*, 58, 295–298.

Efron, B., Hastie, T., Johnstone, I., and Tibshirani, R. (2004), "Least Angle Regression (with discussion)," *Annals of Statistics*, 32, 407–499.

Eilers, P. H. C. and Marx, B. D. (1996) "Flexible Smoothing with B-splines and Penalties," *Statistical Science*, 11, 89–121, with discussion.

Foster, D. P. and Stine, R. A. (2004), "Variable Selection in Data Mining: Building a Predictive Model for Bankruptcy," *Journal of the American Statistical Association*, 99, 303–313.

Harrell, F. E. (2001), *Regression Modeling Strategies,* New York: Springer-Verlag.

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning,* New York: Springer-Verlag.

Hocking, R. R. (1976), "The Analysis and Selection of Variables in Linear Regression," *Biometrics*, 32, 1–50.

Hurvich, C. M., Simonoff, J. S., and Tsai, C.-L. (1998), "Smoothing Parameter Selection in Nonparametric Regression Using an Improved Akaike Information Criterion," *Journal of the Royal Statistical Society B*, 60, Part 2, 271–293.

Hurvich, C. M. and Tsai, C.-L. (1989), "Regression and Time Series Model Selection in Small Samples," *Biometrika*, 76, 297–307.

Judge, G. G., Griffiths, W. E., Hill, R. C., Lutkepohl, H., and Lee, T. C. (1985), *The Theory and Practice of Econometrics,* Second Edition, New York: John Wiley & Sons.

Mallows, C. L. (1967), "Choosing a Subset Regression," unpublished report, Bell Telephone Laboratories.

Mallows, C. L. (1973), "Some Comments on C_p," *Technometrics*, 15, 661–675.

Miller, A. (2002), *Subset Selection in Regression,* Second Edition, Chapman & Hall/CRC.

Osborne, M., Presnell, B., and Turlach, B. (2000), "A New Approach to Variable Selection in Least Squares Problems," *IMA Journal of Numerical Analysis*, 20, 389–404.

Raftery, A. E., Madigan, D., and Hoeting, J. A. (1997), "Bayesian Model Averaging for Linear Regression Models," *Journal of the American Statistical Association*, 92, 179–191.

Sarle, W. S. (2001) "Donoho-Johnstone Benchmarks: Neural Net Results," ftp://ftp.sas.com/pub/neural/dojo/dojo.html: last accessed March 27, 2007.

Sawa, T. (1978), "Information Criteria for Discriminating among Alternative Regression Models," *Econometrica*, 46, 1273–1282.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Sports Illustrated, April 20, 1987.

Tibshirani, R. (1996), "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society Series B*, 58, 267–288.

Chapter 43

The HPMIXED Procedure (Experimental)

Contents

Overview: HPMIXED Procedure **2804**
Basic Features 2804
Assumptions and Notation 2805
Computational Approach 2806
The HPMIXED Procedure Contrasted with the MIXED Procedure 2807
Getting Started: HPMIXED Procedure **2808**
Mixed Model with Large Number of Fixed and Random Effects 2808
Syntax: HPMIXED Procedure **2811**
PROC HPMIXED Statement 2811
BY Statement 2814
CLASS Statement 2815
CONTRAST Statement 2816
ESTIMATE Statement 2818
ID Statement 2821
LSMEANS Statement 2821
MODEL Statement 2822
NLOPTIONS Statement 2824
OUTPUT Statement 2824
PARMS Statement 2826
RANDOM Statement 2829
TEST Statement 2831
WEIGHT Statement 2832
Details: HPMIXED Procedure **2832**
Model Assumptions 2832
Computing and Maximizing the Likelihood 2833
Computing Starting Values by EM-REML 2834
Sparse Matrix Techniques 2835
Hypothesis Tests for Fixed Effects 2837
Default Output 2837
ODS Table Names 2839
Examples: HPMIXED Procedure **2840**
Example 43.1: Ranking Many Random-Effect Coefficients 2840
Example 43.2: Comparing Results from PROC HPMIXED and PROC MIXED 2843
Example 43.3: Using PROC GLIMMIX for Further Analysis of PROC HPMIXED Fit 2848

Example 43.4: Mixed Model Analysis of Microarray Data 2850
References **2854**

Overview: HPMIXED Procedure

The experimental HPMIXED procedure uses a number of specialized high-performance techniques to fit linear mixed models with variance component structure. The HPMIXED procedure is specifically designed to cope with estimation problems involving a large number of fixed effects, a large number of random effects, or a large number of observations.

The HPMIXED procedure complements the MIXED procedure and other SAS/STAT procedures for mixed modeling. On the one hand, the models supported by the HPMIXED procedure are a subset of the models that you can fit with the MIXED procedure, and the confirmatory inferences available in the HPMIXED procedure are also a subset of the general analyses available with the MIXED procedure. On the other hand, the HPMIXED procedure can have considerably better performance than other SAS/STAT mixed modeling tools, in terms of memory requirements and computational speed.

A mixed model can be large in a number of ways, not all of which are suited for the specialized algorithms and storage techniques implemented in the HPMIXED procedure. The following are examples of linear mixed modeling problems for which the HPMIXED procedure has been specifically designed:

- linear mixed models with thousands of levels for the fixed and/or random effects
- linear mixed models with hierarchically nested fixed and/or random effects, possibly with hundreds or thousands of levels at each level of the hierarchy

Basic Features

The HPMIXED procedure enables you to specify a linear mixed model with variance component structure, to estimate the covariance parameters by restricted maximum likelihood, and to perform confirmatory inference in such models. The HPMIXED procedure fits the specified linear mixed model and produces appropriate statistics.

The following are some of the basic features of the HPMIXED procedure:

- capacity to handle large linear mixed model problems for balanced or unbalanced data
- MIXED-type MODEL and RANDOM statements for model specification and CONTRAST, ESTIMATE, LSMEANS, and TEST statements for inferences
- estimate covariance parameters by restricted maximum likelihood (REML)

- output statistics by using the OUTPUT statement
- computation of appropriate standard errors for all specified estimable linear combinations of fixed and random effects, and corresponding t and F tests
- subject and group effects that enable blocking and heterogeneity, respectively
- NLOPTIONS statement, which enables you to exercise control over the numerical optimization

The HPMIXED procedure uses the Output Delivery System (ODS), a SAS subsystem that provides capabilities for displaying and controlling the output from SAS procedures. ODS enables you to convert any of the output from the HPMIXED procedure into a SAS data set. See the section "ODS Table Names" on page 2839 and Chapter 20, "Using the Output Delivery System," for further information about using ODS with the HPMIXED procedure.

Assumptions and Notation

The linear mixed models fit by the HPMIXED procedure can be represented as linear statistical models in the following form:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

$$\boldsymbol{\gamma} \sim N(\mathbf{0}, \mathbf{G})$$

$$\boldsymbol{\epsilon} \sim N(\mathbf{0}, \sigma^2\mathbf{I})$$

$$\mathrm{Cov}[\boldsymbol{\gamma}, \boldsymbol{\epsilon}] = \mathbf{0}$$

The symbols in these expressions denote the following:

$\mathbf{y}$	the $(n \times 1)$ vector of responses
$\mathbf{X}$	the $(n \times k)$ design matrix for the fixed effects
$\boldsymbol{\beta}$	the $(k \times 1)$ vector of fixed-effects parameters
$\mathbf{Z}$	the $(n \times q)$ design matrix for the random effects
$\boldsymbol{\gamma}$	the $(q \times 1)$ vector of random effects
$\boldsymbol{\epsilon}$	the $(n \times 1)$ vector of unobservable residual errors

As is customary for statistical models in the linear mixed model family, the random effects are assumed normally distributed. The same holds for the residual errors and these are furthermore distributed independently of the random effects. As a consequence, these assumptions imply that the response vector $\mathbf{y}$ has a multivariate normal distribution.

Further assumptions, implicit in the preceding expression, are as follows:

- The conditional mean of the data—given the random effects—is linear in the fixed effects and the random effects.
- The marginal mean of the data is linear in the fixed-effects parameters.

Computational Approach

The computational methods to efficiently solve large mixed model problems with the HPMIXED procedure rely on a combination of several techniques, including sparse matrix storage, specialized solving of sparse linear systems, and dedicated nonlinear optimization.

Sparse Storage and Computation

One of the fundamental computational tasks in analyzing a linear mixed model is solving the mixed model equations

$$\begin{bmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{Z} \\ \mathbf{Z}'\mathbf{X} & \mathbf{Z}'\mathbf{Z}+\sigma^2\mathbf{G}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\gamma} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{y} \\ \mathbf{Z}'\mathbf{y} \end{bmatrix}$$

where $\mathbf{G}$ denotes the variance matrix of the random effects. The mixed model crossproduct matrix

$$\begin{bmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{Z} \\ \mathbf{Z}'\mathbf{X} & \mathbf{Z}'\mathbf{Z}+\sigma^2\mathbf{G}^{-1} \end{bmatrix}$$

is a key component of these equations, and it often has many zero values (George and Liu 1981). Sparse storage techniques can result in significant savings in both memory and CPU resources. The HPMIXED procedure draws on sparse matrix representation and storage where appropriate or necessary.

Conjugate Gradient Algorithm and Iteration-on-Data Technology

Solving the mixed model equations is a critical component of linear mixed model analysis. The two main components of the preconditioned conjugate gradient (PCCG) algorithm are preconditioning and matrix-vector product computing (Shewchuk 1994). The algorithm is guaranteed to converge to the solution within n_e iterations, where n_e is equal to the number of distinct eigenvalues of the mixed model equations. This simple yet powerful algorithm can be easily implemented with an iteration-on-data (IOD) technique (Tsuruta, Misztal, and Stranden 2001) that can yield significant savings of memory resources.

The combination of the PCCG algorithm and iteration on data makes it possible to efficiently compute best linear unbiased predictors (BLUPs) for the random effects in mixed models with large mixed model equations.

Average Information Algorithm

The HPMIXED procedure estimates covariance parameters by restricted maximum likelihood. The default optimization method is a quasi-Newton algorithm. When the Hessian or information matrix is required, the HPMIXED procedure takes advantage of the computational simplifications that are available by *averaging information* (AI). The AI algorithm (Johnson and Thompson 1995; Gilmour, Thompson, and Cullis 1995) replaces the second derivative matrix with the average of the observed

and expected information matrices. The computationally intensive trace terms in these information matrices cancel upon averaging. Coarsely, the AI algorithm can be viewed as a hybrid of a Newton-Raphson approach and Fisher scoring.

The HPMIXED Procedure Contrasted with the MIXED Procedure

The HPMIXED procedure is designed to solve large mixed model problems by using sparse matrix techniques. A mixed model can be large in many ways: a large number of observations, a large number of columns in the $\mathbf{X}$ matrix, a large number of columns in the $\mathbf{Z}$ matrix, and a large number of covariance parameters. The aim of the HPMIXED procedure is parameter estimation, inference, and prediction in linear mixed models with large $\mathbf{X}$ and/or $\mathbf{Z}$ matrices and many observations, but with relatively few covariance parameters.

The models that you can fit with the HPMIXED procedure and the available postprocessing analyses are a subset of the models and analyses available with the MIXED procedure. With the HPMIXED procedure you can model only G-side random effects with variance component structure or an unstructured covariance matrix in a Cholesky parameterization. R-side random effects and direct modeling of their covariance structures are not supported.

The MIXED and HPMIXED procedures offer different balances for computing performance and statistical generality. To some extent the generality of the MIXED procedure means that it cannot serve as a high-performance computing tool for all of the model-data scenarios that it can potentially handle. For example, although efficient sparse algorithms are available to estimate variance components in large linear mixed models, the computational configuration changes profoundly when, for example, Kenward-Roger degree-of-freedom adjustments are requested.

On the other hand, the HPMIXED procedure can handle only a small subset of the models that PROC MIXED can fit. Invariably, some features of high-performance sparse computing methods might be surprising at first. For example, the best computational path depends on the model and the data, so that in models with a singular $\mathbf{X}'\mathbf{X}$ matrix, the order in which singularities are detected and accounted for can change from one data set to the next.

The following is a list of features available in the MIXED procedure, but *not* available in the HPMIXED procedure:

- a REPEATED statement to model R-side covariance structures
- a variety of covariance structures by using the TYPE= option in the RANDOM statement
- automatic Type III tests of fixed effects. You request tests of fixed effects in the HPMIXED procedure with the TEST statement.
- ODS statistical graphics
- advanced degree-of-freedom adjustments available by using the DDFM= option
- maximum likelihood or method-of-moments estimation for the covariance parameters
- a PRIOR statement for a sampling-based Bayesian analysis

Getting Started: HPMIXED Procedure

Mixed Model with Large Number of Fixed and Random Effects

In animal breeding, it is common to model genetic and environmental effects with a random effect for the animal. When there are many animals being studied, this can lead to very large mixed model equations to be solved. In this example we present an analysis of simulated data with this structure.

Suppose you have 3000 animals from five different genetic species raised on 100 different farms. The following DATA step simulates 40000 observations of milk yield (Yield) from a linear mixed model with variables Species and Farm in the fixed-effect model and Animal as a random effect. The random effect due to Animal is simulated with a variance of 4.0, while the residual error variance is 8.0. These variance component values reflect the fact that variation in milk yield is typically genetically controlled to be no more than 33% (4/(4+8)).

```
data Sim;
   keep Species Farm Animal Yield;
   array AnimalEffect{3000};
   array AnimalFarm{3000};
   array AnimalSpecies{3000};
   do i = 1 to dim(AnimalEffect);
      AnimalEffect{i} = sqrt(4.0)*rannor(12345);
      AnimalFarm{i}    = 1 + int(100*ranuni(12345));
      AnimalSpecies{i} = 1 + int(5*ranuni(12345));
      end;
   do i = 1 to 40000;
      Animal  = 1 + int(3000*ranuni(12345));
      Species = AnimalSpecies{Animal};
      Farm    = AnimalFarm{Animal};
      Yield   = 1 + Species + Farm/10 + AnimalEffect{Animal}
                  + sqrt(8.0)*rannor(12345);
      output;
      end;
run;
```

A simple linear mixed model analysis is performed by using the following SAS statements:

```
proc hpmixed data=Sim;
   class Species Farm Animal;
   model Yield = Species Species*Farm;
   random Animal;
   test Species*Farm;
   contrast 'Species1 = Species2 = Species3'
      Species 1 0 -1,
      Species 0 1 -1;
run;
```

Selected results from the preceding SAS statements are shown in Figure 43.1 through Figure 43.4.

The "Class Level Information" table in Figure 43.1 shows that the three model effects have 5, 100, and 3000 levels, respectively. Only a portion of the levels are displayed by default. The "Dimensions" table shows that the model contains a single G-side covariance parameter and a single R-side covariance parameter. R-side covariance parameters are those associated with the covariance matrix $\mathbf{R}$ in the conditional distribution, given the random effects. In the case of the HPMIXED procedure this matrix is simply $\mathbf{R} = \sigma^2\mathbf{I}$ and the single R-side covariance parameter corresponds to the residual variance. The G-side parameter is the variance of the random Animal effect; the $\mathbf{G}$ matrix is a diagonal (3000×3000) matrix with the common variance on the diagonal.

Figure 43.1 Class Levels and Dimensions

```
                        The HPMIXED Procedure

                      Class Level Information

Class       Levels    Values

Species          5    1 2 3 4 5
Farm           100    1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
                      19 20 ...
Animal        3000    1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
                      19 20 ...

                             Dimensions

                  G-side Cov. Parameters             1
                  R-side Cov. Parameters             1
                  Columns in X                     506
                  Columns in Z                    3000
                  Subjects (Blocks in V)             1
```

Taking into account the intercept as well as the number of levels of the Species and Species*Farm effects, the $\mathbf{X}$ matrix for this problem has 506 columns, so that the mixed model equations

$$\begin{bmatrix} \mathbf{X}'\mathbf{X} & \mathbf{X}'\mathbf{Z} \\ \mathbf{Z}'\mathbf{X} & \mathbf{Z}'\mathbf{Z} + \sigma^2\mathbf{G}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\gamma} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{y} \\ \mathbf{Z}'\mathbf{y} \end{bmatrix}$$

have 3506 rows and columns. This is a substantial computational problem: simply storing a single copy of this matrix in dense format requires nearly 50 megabytes of memory. The sparse matrix techniques of PROC HPMIXED use a small fraction of this amount of memory and a similarly small fraction of the CPU time required to solve the equations with dense techniques. For more information about sparse versus dense techniques, see the section "Sparse Matrix Techniques" on page 2835.

Figure 43.2 displays the covariance parameter estimates at convergence of the REML algorithm. The variance component estimate for animal effect is $\widehat{\sigma}^2_a = 3.9889$ and for residual $\widehat{\sigma}^2 = 7.9623$. These estimates are close to the simulated values (4.0 and 8.0).

Figure 43.2 Estimates of Variance Components

Covariance Parameter Estimates

Cov Parm	Estimate
Animal	3.9889
Residual	7.9623

The TEST statement requests a Type III test of the fixed effect in the model. By default, the HPMIXED procedure does not compute Type III tests, because they can be computationally demanding. The tests of the Species*Farm effect is highly significant. That indicates animals of a genetic species perform differently in different environments.

Figure 43.3 Type III Tests of Fixed Effect

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Species*Farm	495	39500	11.72	<.0001

You can use the CONTRAST or ESTIMATE statement to test custom linear hypotheses involving the fixed and/or random effects. The CONTRAST statement in the preceding program tests the null hypothesis that there are no differences among the first three genetic species. Results from this analysis are shown in Figure 43.4. The small p-value indicates that there are significant differences among the first three genetics species.

Figure 43.4 Result of CONTRAST Statement

Contrasts

Label	Num DF	Den DF	F Value	Pr > F
Species1 = Species2 = Species3	2	39500	92.93	<.0001

Syntax: HPMIXED Procedure

The following statements are available in PROC HPMIXED:

PROC HPMIXED < *options* > ;
BY *variables* ;
CLASS *variables* ;
ID *variables* ;
MODEL *dependent* = < *fixed-effects* > < / *options* > ;
RANDOM *random-effects* < / *options* > ;
PARMS < *(value-list) . . .* > < / *options* > ;
TEST *fixed-effects* < / *options* > ;
CONTRAST *'label' contrast-specification* < , *contrast-specification* > < , . . . > < / *options* > ;
ESTIMATE *'label' contrast-specification* < **(***divisor=n***)** >
< , *'label' contrast-specification* < **(***divisor=n***)** > > < , . . . > < /*options* > ;
LSMEANS *fixed-effects* < / *options* > ;
NLOPTIONS < *options* > ;
OUTPUT < **OUT=***SAS-data-set* >
< *keyword*< **(***keyword-options***)** >< *=name* > >. . .
< *keyword*< **(***keyword-options***)** >< *=name* > > < / *options* > ;
WEIGHT *variable* ;

Items within angle brackets (< >) are optional. The CONTRAST, ESTIMATE, LSMEANS, RANDOM, and TEST statements can appear multiple times; all other statements can appear only once.

The PROC HPMIXED and MODEL statements are required, and the MODEL statement must appear after the CLASS statement if these statements are included. The syntax of each statement is described in the following sections in alphabetical order after the description of the PROC HPMIXED statement.

PROC HPMIXED Statement

PROC HPMIXED < *options* > ;

The PROC HPMIXED statement invokes the procedure. You can specify the following options.

DATA=*SAS-data-set*
names the SAS data set to be used by PROC HPMIXED. The default is the most recently created data set.

INFOCRIT=NONE | PQ | Q
IC=NONE | PQ | Q
determines the computation of information criteria in the "Fit Statistics" table. The criteria are all in smaller-is-better form, and are described in Table 43.1.

Table 43.1 Information Criteria

Criteria	Formula	Reference
AIC	$-2\ell + 2d$	Akaike (1974)
AICC	$-2\ell + 2dn^*/(n^* - d - 1)$ for $n^* \geq d + 2$ $-2\ell + 2d(d + 2)$ for $n^* < d + 2$	Hurvich and Tsai (1989) and Burnham and Anderson (1998)
HQIC	$-2\ell + 2d \log(\log(n))$	Hannan and Quinn (1979)
BIC	$-2\ell + d \log(n)$	Schwarz (1978)
CAIC	$-2\ell + d(\log(n) + 1)$	Bozdogan (1987)

Here ℓ denotes the maximum value of the restricted log likelihood, d is the dimension of the model, and n, n^* reflect the size of the data.

The quantities d, n, and n^* depend on the model and IC= option.

- models without random effects:
  The IC=Q and IC=PQ options have no effect on the computation.
  - d equals the number of parameters in the optimization whose solutions do not fall on the boundary or are otherwise constrained.
  - n equals the number of used observations minus rank($\mathbf{X}$).
  - n^* equals n, unless $n < d + 2$, in which case $n^* = d + 2$.
- models with random effects:
  - d equals the number of parameters in the optimization whose solutions do not fall on the boundary or are otherwise constrained. If IC=PQ, this value is incremented by rank($\mathbf{X}$).
  - n equals the effective number of subjects as displayed in the "Dimensions" table, unless this value equals 1, in which case n equals the number of levels of the first random effect specified. The IC=Q and IC=PQ options have no effect.
  - n^* equals n, unless $n < d + 2$, in which case $n^* = d + 2$. The IC=Q and IC=PQ options have no effect.

The IC=NONE option suppresses the "Fit Statistics" table. IC=Q is the default.

ITDETAILS
displays the parameter values at each iteration and enables the writing of notes to the SAS log pertaining to "infinite likelihood" and "singularities" during optimization iterations.

MAXCLPRINT=*number*
specifies the maximum levels of CLASS variables to print in the ODS table "ClassLevels." The default value is 20. MAXCLPRINT=0 enables you to print all levels of each CLASS variable. However, the option NOCLPRINT takes precedence over MAXCLPRINT.

METHOD=REML
specifies the estimation method for the covariance parameters. The REML specification performs residual (restricted) maximum likelihood, and it is the default method. There is no other choice at this experimental stage.

MMEQ

displays coefficients of the mixed model equations. These are

$$\begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{Z} \\ \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}+\widehat{\mathbf{G}}^{-1} \end{bmatrix} \begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \\ \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \end{bmatrix}$$

assuming $\widehat{\mathbf{G}}$ is nonsingular. If $\widehat{\mathbf{G}}$ is singular, PROC HPMIXED produces the following coefficients

$$\begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}\widehat{\mathbf{G}} \\ \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}\widehat{\mathbf{G}}+\widehat{\mathbf{G}} \end{bmatrix} \begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \\ \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \end{bmatrix}$$

See the section "Model and Assumptions" on page 2832 for further information about these equations.

NAMELEN=*number*

specifies the length to which long effect names are shortened. The default and minimum value is 20.

NOCLPRINT< =*number*>

suppresses the display of the "Class Level Information" table if you do not specify *number*. If you do specify *number*, only levels with totals that are less than *number* are listed in the table.

NOFIT

suppresses fitting of the model. When the NOFIT option is in effect, PROC HPMIXED produces the "Model Information," "Class Level Information," "Number of Observations," "Dimensions," and "Descriptive Statistics" tables. These can be helpful in gauging the computational effort required to fit the model.

NOINFO

suppresses the display of the "Model Information," "Number of Observations," and "Dimensions" tables.

NOITPRINT

suppresses the display of the "Iteration History" table.

NOPRINT

suppresses the normal display of results. The NOPRINT option is useful when you want only to create one or more output data sets with the procedure by using the OUTPUT statement. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

NOPROFILE

includes the residual variance as one of the covariance parameters in the optimization iterations. This option applies only to models that have a residual variance parameter. By default, this parameter is profiled out of the optimization iterations, except when you have specified the HOLD= option in the PARMS statement.

ORDER=DATA | FORMATTED | FREQ | INTERNAL
specifies the sorting order for the levels of all CLASS variables. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option can be useful when you use CONTRAST or ESTIMATE statements.

The default is ORDER=FORMATTED.

The following table shows how PROC HPMIXED interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

For FORMATTED and INTERNAL, the sort order is machine dependent. For more information about sorting order, see the chapter about the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

SINGCHOL=*number*
tunes the singularity criterion in Cholesky decompositions. The default is 1E6 times the machine epsilon; this product is approximately 1E−10 on most computers.

SINGRES=*number*
sets the tolerance for which the residual variance is considered to be zero. The default is 1E4 times the machine epsilon; this product is approximately 1E−12 on most computers.

SINGULAR=*number*
tunes the general singularity criterion applied by the HPMIXED procedure in divisions and inversions. The default is 1E4 times the machine epsilon; this product is approximately 1E−12 on most computers.

BY Statement

BY *variables* **;**

The BY statement separates analyses about observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement options NOTSORTED or DESCENDING in the BY statement for the HPMIXED procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

Since sorting the data changes the order in which PROC HPMIXED reads observations, the sorting order for the levels of the CLASS variable might be affected if you have specified ORDER=DATA in the PROC HPMIXED statement. This, in turn, affects specifications in the CONTRAST and ESTIMATE statements.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

The CLASS statement names the classification variables to be used in the analysis. If the CLASS statement is used, it must appear before the MODEL statement.

Classification variables can be either character or numeric. By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous SAS releases in the way in which class levels are determined. Prior to SAS 9, class levels were determined by using no more than the first 16 characters of the formatted values. If you want to revert to this previous behavior you can use the TRUNCATE option in the CLASS statement. In any case, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *Base SAS Procedures Guide* and to the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*. You can adjust the order of CLASS variable levels with the ORDER= option in the PROC HPMIXED statement.

You can specify the following option in the CLASS statement after a slash(/):

TRUNCATE
specifies that class levels should be determined by using only no more than the first 16 characters of the formatted values of CLASS variables.

CONTRAST Statement

CONTRAST *'label' contrast-specification* < *, contrast-specification* > < *, . . .* > < */ options* > **;**

The CONTRAST statement provides a mechanism for obtaining custom hypothesis tests. It is patterned after the CONTRAST statement in PROC MIXED and enables you to select an appropriate inference space (McLean, Sanders, and Stroup 1991).

You can test the hypothesis $\mathbf{L}'\boldsymbol{\phi} = \mathbf{0}$, where $\mathbf{L}' = [\mathbf{K}'\ \mathbf{M}']$ and $\boldsymbol{\phi}' = [\boldsymbol{\beta}'\ \boldsymbol{\gamma}']$, in several inference spaces. The inference space corresponds to the choice of $\mathbf{M}$. When $\mathbf{M} = \mathbf{0}$, your inferences apply to the entire population from which the random effects are sampled; this is known as the *broad* inference space. When all elements of $\mathbf{M}$ are nonzero, your inferences apply only to the observed levels of the random effects. This is known as the *narrow* inference space, and you can also choose it by specifying all of the random effects as fixed. The GLM procedure uses the narrow inference space. Finally, by zeroing portions of $\mathbf{M}$ corresponding to selected main effects and interactions, you can choose *intermediate* inference spaces. The broad inference space is usually the most appropriate, and it is used when you do not specify any random effects in the CONTRAST statement.

In the CONTRAST statement,

label identifies the contrast in the table. A label is required for every contrast specified. Labels can be up to 20 characters and must be enclosed in single quotes.

contrast-specification identifies the fixed effects and random effects and their coefficients from which the $\mathbf{L}$ matrix is formed. The syntax representation of a *contrast-specification* is
< fixed-effect values . . . > < | random-effect values . . . >

fixed-effect identifies an effect that appears in the MODEL statement. The keyword INTERCEPT can be used as an effect when an intercept is fitted in the model. You do not need to include all effects that are in the MODEL statement.

random-effect identifies an effect that appears in the RANDOM statement. The first random effect must follow a vertical bar (|); however, random effects do not have to be specified.

values are constants that are elements of the $\mathbf{L}$ matrix associated with the fixed and random effects.

The rows of $\mathbf{L}'$ are specified in order and are separated by commas. The rows of the $\mathbf{K}'$ component of $\mathbf{L}'$ are specified on the left side of the vertical bars (|). These rows test the fixed effects and are, therefore, checked for estimability. The rows of the $\mathbf{M}'$ component of $\mathbf{L}'$ are specified on the right side of the vertical bars. They test the random effects, and no estimability checking is necessary.

If PROC HPMIXED finds the fixed-effects portion of the specified contrast to be nonestimable (see the SINGULAR= option on page 2818), then it displays missing values for the test statistics and a note in the log.

If the elements of $\mathbf{L}$ are not specified for an effect that contains a specified effect, then the elements of the specified effect are automatically "filled in" over the levels of the higher-order effect. This

feature is designed to preserve estimability for cases where there are complex higher-order effects. The coefficients for the higher-order effect are determined by equitably distributing the coefficients of the lower-level effect as in the construction of least squares means. In addition, if the intercept is specified, it is distributed over all classification effects that are not contained by any other specified effect. If an effect is not specified and does not contain any specified effects, then all of its coefficients in $\mathbf{L}$ are set to 0. You can override this behavior by specifying coefficients for the higher-order effect.

If too many values are specified for an effect, the extra ones are ignored; if too few are specified, the remaining ones are set to 0. If no random effects are specified, the vertical bar can be omitted; otherwise, it must be present. If a SUBJECT effect is used in the RANDOM statement, then the coefficients specified for the effects in the RANDOM statement are equitably distributed across the levels of the SUBJECT effect. You can use the E option to see exactly what $\mathbf{L}$ matrix is used.

The SUBJECT and GROUP options in the CONTRAST statement are useful for the case where a SUBJECT= or GROUP= variable appears in the RANDOM statement, and you want to contrast different subjects or groups. By default, CONTRAST statement coefficients about random effects are distributed equally across subjects and groups.

PROC HPMIXED handles missing level combinations of CLASS variables similarly to the way PROC GLM does. Both procedures delete fixed-effects parameters corresponding to missing levels in order to preserve estimability. However, PROC HPMIXED does not delete missing level combinations for random-effects parameters because linear combinations of the random-effects parameters are always estimable. These conventions can affect the way you specify your CONTRAST coefficients.

The CONTRAST statement computes the statistic

$$F = \frac{\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}' \mathbf{L}(\mathbf{L}'\widehat{\mathbf{C}}\mathbf{L})^{-1}\mathbf{L}' \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}}{r}$$

where $r = \text{rank}(\mathbf{L}'\widehat{\mathbf{C}}\mathbf{L})$ and approximates its distribution with an F distribution. In this expression, $\widehat{\mathbf{C}}$ is an estimate of the generalized inverse of the coefficient matrix in the mixed model equations.

The numerator degree of freedom in the F approximation is $r = \text{rank}(\mathbf{L}'\widehat{\mathbf{C}}\mathbf{L})$, and the denominator degree of freedom is taken from the "Type III Tests of Fixed Effects" table and corresponds to the final effect you list in the CONTRAST statement. You can change the denominator degrees of freedom by using the DF= option.

You can specify the following options in the CONTRAST statement after a slash (/).

CHISQ

requests that χ^2 tests be performed in addition to any F tests. A χ^2 statistic equals its corresponding F statistic times the associate numerator degree of freedom, and this same degree of freedom is used to compute the p-value for the χ^2 test. This p-value will always be less than that for the F test, as it effectively corresponds to an F test with infinite denominator degrees of freedom.

DF=*number*

specifies the denominator degrees of freedom for the F test. The default is the denominator

degrees of freedom taken from the "Type III Tests of Fixed Effects" table and corresponds to the final effect you list in the CONTRAST statement.

E

requests that the **L** matrix coefficients for the contrast be displayed. For ODS purposes, the name of this "L Matrix Coefficients" table is "Coef."

GROUP *coeffs*

sets up random-effect contrasts between different groups when a GROUP variable appears in the RANDOM statement. By default, CONTRAST statement coefficients about random effects are distributed equally across groups. If you enter a multi-row contrast, you can also enter multiple rows for the GROUP coefficients. If the number of GROUP coefficients is less than the number of contrasts in the CONTRAST statement, the HPMIXED procedure cycles through the GROUP coefficients. For example, the following two statements are equivalent:

```
contrast 'Trt @ x=0.4 and 0.5' trt 1 -1  0 | x 0.4,
                               trt 1  0 -1 | x 0.4,
                               trt 1 -1  0 | x 0.5,
                               trt 1  0 -1 | x 0.5 /
                 group 1 -1, 1 0 -1, 1 -1, 1 0 -1;

contrast 'Trt @ x=0.4 and 0.5' trt 1 -1  0 | x 0.4,
                               trt 1  0 -1 | x 0.4,
                               trt 1 -1  0 | x 0.5,
                               trt 1  0 -1 | x 0.5 /
                 group 1 -1, 1 0 -1;
```

SINGULAR=*number*

tunes the estimability checking. If **v** is a vector, define ABS(**v**) to be the largest absolute value of the element of **v** with the largest absolute value. If ABS($\mathbf{K}' - \mathbf{K}'\mathbf{T}$) is greater than *c*number* for any row of $\mathbf{K}'$ in the contrast, then **K** is declared nonestimable. Here **T** is the Hermite form matrix $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$, and *c* is ABS($\mathbf{K}'$) except when it equals 0, and then *c* is 1. The value for *number* must be between 0 and 1; the default is 1E−4.

SUBJECT *coeffs*

sets up random-effect contrasts between different subjects when a SUBJECT= variable appears in the RANDOM statement. By default, CONTRAST statement coefficients about random effects are distributed equally across subjects. Listing subject coefficients for multiple row CONTRASTS follows the same rules as for GROUP coefficients.

ESTIMATE Statement

ESTIMATE *'label' contrast-specification* < **(***divisor=n***)** > < *, 'label' contrast-specification* < **(***divisor=n***)** > > < *, . . .* > < */options* > **;**

The ESTIMATE statement provides a mechanism for obtaining custom hypothesis tests. As in the CONTRAST statement, the basic element of the ESTIMATE statement is the *contrast-specification*,

which consists of MODEL and RANDOM effects and their coefficients. Specifically, a *contrast-specification* takes the form

< *fixed-effect values* ... > < | *random-effect values* ... >

Based on the *contrast-specifications* in your ESTIMATE statement, PROC HPMIXED constructs the matrix $\mathbf{L}' = [\mathbf{K}' \ \mathbf{M}']$, as in the CONTRAST statement, where $\mathbf{K}$ is associated with the fixed effects and $\mathbf{M}$ is associated with the G-side random effects.

PROC HPMIXED then produces for each row $\mathbf{l}$ of $\mathbf{L}'$ an approximate t test of the hypothesis $H: \mathbf{l}\boldsymbol{\phi} = 0$, where $\boldsymbol{\phi} = [\boldsymbol{\beta}' \ \boldsymbol{\gamma}']'$. Results from all ESTIMATE statement are combined in the "Estimates" ODS table.

Note that multi-row estimates are permitted. Unlike the CONTRAST statement, you need to specify a *'label'* for every row of the multi-row estimate, since PROC HPMIXED produces one test per row.

PROC HPMIXED selects the degrees of freedom to match those displayed in the "Type III Tests of Fixed Effects" table for the final effect you list in the ESTIMATE statement. You can modify the degrees of freedom by using the DF= option. If you select DDFM=NONE and do not modify the degrees of freedom by using the DF= option, PROC HPMIXED uses infinite degrees of freedom, essentially computing approximate z tests.

If PROC HPMIXED finds the fixed-effects portion of the specified estimate to be nonestimable, then it displays "Non-est" for the estimate entry.

The construction of the $\mathbf{L}$ matrix for an ESTIMATE statement follows the same rules as listed under the CONTRAST statement.

You can specify the following options in the ESTIMATE statement after a slash (/).

ALPHA=*number*

requests that a t-type confidence interval be constructed with confidence level $1-$*number*. The value of *number* must be between 0 and 1 exclusively; the default is 0.05. If DDFM=NONE and you do not specify degrees of freedom with the DF= option, PROC HPMIXED uses infinite degrees of freedom, essentially computing a z interval.

CL

requests that t-type confidence limits be constructed. If DDFM=NONE and you do not specify degrees of freedom with the DF= option, PROC HPMIXED uses infinite degrees of freedom, essentially computing a z interval. The confidence level is 0.95 by default.

DF=*number*

specifies the degrees of freedom for the t-test. The default is the denominator degrees of freedom taken from the "Type III Tests of Fixed Effects" table and corresponds to the final effect you list in the ESTIMATE statement.

DIVISOR=*value-list*

specifies a list of values by which to divide the coefficients so that fractional coefficients can be entered as integer numerators. If you do not specify *value-list*, a default value of 1.0 is assumed. Missing values in the *value-list* are converted to 1.0.

If the number of elements in *value-list* exceeds the number of rows of the estimate, the extra values are ignored. If the number of elements in *value-list* is less than the number of rows of the estimate, the last value in *value-list* is copied forward.

If you specify a row-specific divisor as part of the specification of the estimate row, this value multiplies the corresponding divisor implied by the *value-list*. For example, the following statement divides the coefficients in the first row by 8, and the coefficients in the third and fourth row by 3:

```
estimate 'One vs. two'   A 2 -2  (divisor=2),
         'One vs. three' A 1  0 -1         ,
         'One vs. four'  A 3  0  0 -3      ,
         'One vs. five'  A 1  0  0  0  -1  / divisor=4,.,3;
```

E

requests that the matrix coefficients be displayed. For ODS purposes, the name of this "L Matrix Coefficients" table is "Coef."

GROUP *coeffs*

sets up random-effect contrasts between different groups when a GROUP= variable appears in the RANDOM statement. By default, ESTIMATE statement coefficients about random effects are distributed equally across groups. If you enter a multi-row estimate, you can also enter multiple rows for the GROUP coefficients. If the number of GROUP coefficients is less than the number of contrasts in the ESTIMATE statement, the HPMIXED procedure cycles through the GROUP coefficients. For example, the following two statements are equivalent:

```
estimate 'Trt 1 vs 2 @ x=0.4' trt 1 -1  0 | x 0.4,
         'Trt 1 vs 3 @ x=0.4' trt 1  0 -1 | x 0.4,
         'Trt 1 vs 2 @ x=0.5' trt 1 -1  0 | x 0.5,
         'Trt 1 vs 3 @ x=0.5' trt 1  0 -1 | x 0.5 /
                group 1 -1, 1 0 -1, 1 -1, 1 0 -1;

estimate 'Trt 1 vs 2 @ x=0.4' trt 1 -1  0 | x 0.4,
         'Trt 1 vs 3 @ x=0.4' trt 1  0 -1 | x 0.4,
         'Trt 1 vs 2 @ x=0.5' trt 1 -1  0 | x 0.5,
         'Trt 1 vs 3 @ x=0.5' trt 1  0 -1 | x 0.5 /
                group 1 -1, 1 0 -1;
```

SINGULAR=*number*

tunes the estimability checking as documented for the SINGULAR= in the CONTRAST statement.

SUBJECT *coeffs*

sets up random-effect estimates between different subjects when a SUBJECT= variable appears in the RANDOM statement. By default, ESTIMATE statement coefficients about random effects are distributed equally across subjects. Listing subject coefficients for an ESTIMATE statement with multiple rows follows the same rules as for GROUP coefficients.

ID Statement

ID *variables* **;**

The ID statement specifies which variables from the input data set are to be included in the OUT= data sets from the OUTPUT statement. If you do not specify an ID statement, then all variables are included in these data sets. Otherwise, only the variables you list in the ID statement are included. Specifying an ID statement with no variables prevents any variables from being included in these data sets.

LSMEANS Statement

LSMEANS *fixed-effects* < / *options* > **;**

The LSMEANS statement computes least squares means (LS-means) of fixed effects. As in the GLM procedure, LS-means are *predicted population margins*—that is, they estimate the marginal means over a balanced population. In a sense, LS-means are to unbalanced designs as classification and subclassification arithmetic means are to balanced designs. The **L** matrix constructed to compute them is the same as the **L** matrix formed in PROC GLM; however, the standard errors are adjusted for the covariance parameters in the model.

Each LS-mean is computed as $\mathbf{L}'\widehat{\boldsymbol{\beta}}$, where $\mathbf{L}$ is the coefficient matrix associated with the least squares mean and $\widehat{\boldsymbol{\beta}}$ is the estimate of the fixed-effects parameter vector. The approximate standard errors for the LS-mean is computed as the square root of $\mathbf{L}'(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}\mathbf{L}$.

LS-means can be computed for any effect in the MODEL statement that involves CLASS variables. You can specify multiple effects in one LSMEANS statement or in multiple LSMEANS statements, and all LSMEANS statements must appear after the MODEL statement. As in the ESTIMATE statement, the **L** matrix is tested for estimability, and if this test fails, PROC HPMIXED displays "Non-est" for the LS-means entries.

Assuming the LS-mean is estimable, PROC HPMIXED constructs an approximate *t* test to test the null hypothesis that the associated population quantity equals zero. By default, the denominator degrees of freedom for this test are the same as those displayed for the effect in the "Type III Tests of Fixed Effects" table (see the section "TEST Statement" on page 2831).

You can specify the following options in the LSMEANS statement after a slash (/).

ALPHA=*number*
: requests that a *t*-type confidence interval be constructed for each of the LS-means with confidence level $1 - number$. The value of *number* must be between 0 and 1; the default is 0.05.

CL
: requests that *t*-type confidence limits be constructed for each of the LS-means. If DDFM=NONE, then PROC HPMIXED uses infinite degrees of freedom for this test, es-

sentially computing a z interval. The confidence level is 0.95 by default; this can be changed with the ALPHA= option.

CORR

displays the estimated correlation matrix of the least squares means as part of the "Least Squares Means" table.

COV

displays the estimated covariance matrix of the least squares means as part of the "Least Squares Means" table.

DF=*number*

specifies the degrees of freedom for the t test and confidence limits. The default is the denominator degrees of freedom taken from the "Type III Tests of Fixed Effects" table corresponding to the LS-means effect. For these DDFM= methods, degrees of freedom are determined separately for each test; see the DDFM= option on page 2823 for more information.

E

requests that the matrix coefficients for all LSMEANS effects be displayed. For ODS purposes, the name of this "Matrix Coefficients" table is "Coef."

SINGULAR=*number*

tunes the estimability checking as documented for the SINGULAR= in the CONTRAST statement.

MODEL Statement

MODEL *dependent* = < *fixed-effects* >< / *options* > **;**

The MODEL statement names a single dependent variable and the fixed effects, which determine the **X** matrix of the mixed model. The specification of effects is the same as in the GLM procedure; however, unlike PROC GLM, you do not specify random effects in the MODEL statement. The MODEL statement is required.

An intercept is included in the fixed-effects model by default. If no fixed effects are specified, only this intercept term is fit. The intercept can be removed by using the NOINT option.

You can specify the following options in the MODEL statement after a slash (/).

ALPHA=*number*

requests that a t-type confidence interval be constructed for each of the fixed-effects parameters with confidence level $1 - number$. The value of *number* must be between 0 and 1; the default is 0.05.

CL

requests that t-type confidence limits be constructed for each of the fixed-effects parameter estimates. The confidence level is 0.95 by default; this can be changed with the ALPHA= option.

DDF=*value-list*

enables you to specify your own denominator degrees of freedom for the fixed effects. The *value-list* specification is a list of numbers or missing values (.) separated by commas. The degrees of freedom should be listed in the order in which the effects appear in the "Type III Tests of Fixed Effects" table. If you want to retain the default degrees of freedom for a particular effect, use a missing value for its location in the list. For example, the following statement assigns 3 denominator degrees of freedom to A and 4.7 to A*B, while those for B remain the same:

```
model Y = A B A*B / ddf=3,.,4.7;
```

DDFM=RESIDUAL | NONE

specifies the method for computing the denominator degrees of freedom for the tests of fixed effects resulting from the MODEL, CONTRAST, ESTIMATE, LSMEANS, and TEST statements.

The DDFM=RESIDUAL option performs all tests by using the residual degrees of freedom, $n - \text{rank}(\mathbf{X})$, where n is the number of observations used. It is the default degrees of freedom method.

DDFM=NONE specifies that no denominator degrees of freedom be applied. PROC HPMIXED then essentially assumes that infinite degrees of freedom are available in the calculation of p-values. The p-values for t tests are then identical to p-values derived from the standard normal distribution. In the case of F tests, the p-values equal those of chi-square tests determined as follows: if F_{obs} is the observed value of the F test with l numerator degrees of freedom, then

$$p = \Pr\{F_{l,\infty} > F_{obs}\} = \Pr\{\chi^2_l > lF_{obs}\}$$

NOINT

requests that no intercept be included in the model. An intercept is included by default.

SOLUTION | S

requests that a solution for the fixed-effects parameters be produced. Using notation from the section "Model Assumptions" on page 2832, the fixed-effects parameter estimates are $\widehat{\boldsymbol{\beta}}$ and their approximate standard errors are the square roots of the diagonal elements of $(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}$.

Along with the estimates and their approximate standard errors, a t statistic is computed as the estimate divided by its standard error. The degree of freedom for this t statistic matches the one appearing in the "Type III Tests of Fixed Effects" table under the effect containing the parameter. The "Pr > |t|" column contains the two-tailed p-value corresponding to the t statistic and associated degrees of freedom.

ZETA=*number*

tunes the sensitivity in forming Type III functions. Any element in the estimable function basis with an absolute value less than *number* is set to 0. The default is 1E−8.

NLOPTIONS Statement

NLOPTIONS < *options* > **;**

For more information about the NLOPTIONS, see the section "Nonlinear Optimization: The NLOPTIONS Statement" on page 391 in Chapter 18, "Shared Concepts and Topics."

If you choose TECH=NEWRAP, then the default value of LSPRECISION is 0.4 in the HPMIXED procedure.

OUTPUT Statement

OUTPUT < **OUT**=*SAS-data-set* >
< *keyword*< **(***keyword-options***)** > < *=name* > >. . .
< *keyword*< **(***keyword-options***)** > < *=name* > > < */ options* > **;**

The OUTPUT statement creates a data set that contains predicted values and residual diagnostics, computed after fitting the model. By default, all variables in the original data set are included in the output data set.

You can use the ID statement to select a subset of the variables from the input data set to be added to the output data set.

For example, suppose that the data set Scores contains the variables score, machine, and person. The following statements fit a model with fixed machine and random person effects and save the predicted and residual values to the data set igausout:

```
proc hpmixed data = Scores;
   class machine person score;
   model score = machine;
   random person;
   output out=igausout pred=p resid=r;
run;
```

You can specify the following options in the OUTPUT statement before the slash (/).

OUT=*SAS data set*
DATA=*SAS data set*
specifies the name of the output data set. If the OUT= (or DATA=) option is omitted, the procedure uses the DATA*n* convention to name the output data set.

keyword < **(***keyword-options***)** >< *=name* >
specifies a statistic to include in the output data set and optionally assigns the variable the name name. You can use the *keyword-options* to control which type of a particular statistic to compute. The *keyword-options* can take on the following values:

BLUP	uses the predictors of the random effects in computing the statistic.
NOBLUP	does not use the predictors of the random effects in computing the statistic.

The default is to compute statistics by using BLUPs. For example, the following two OUTPUT statements are equivalent:

```
output out=out1  pred=predicted lcl=lower;
output out=out1  pred(blup)=predicted lcl(blup)=lower;
```

If a particular combination of keyword and keyword options is not supported, the statistic is not computed and a message is produced in the SAS log.

A *keyword* can appear multiple times in the OUTPUT statement. Table 43.2 lists the keywords and the default names assigned by the HPMIXED procedure if you do not specify a *name*. In this table, y denotes the response variable.

Table 43.2 Keywords for Output Statistics

Keyword	Options	Description	Expression	Name
PREDICTED	BLUP	Linear predictor	$\widehat{\eta} = \mathbf{x}'\widehat{\boldsymbol{\beta}} + \mathbf{z}'\widehat{\boldsymbol{\gamma}}$	Pred
	NOBLUP	Marginal linear predictor	$\widehat{\eta}_m = \mathbf{x}'\widehat{\boldsymbol{\beta}}$	PredPA
STDERR	BLUP	Standard deviation of linear predictor	$\sqrt{\mathrm{Var}[\widehat{\eta} - \mathbf{z}'\boldsymbol{\gamma}]}$	StdErr
	NOBLUP	Standard deviation of marginal linear predictor	$\sqrt{\mathrm{Var}[\widehat{\eta}_m]}$	StdErrPA
RESIDUAL	BLUP	Residual	$r = y - \widehat{\eta}$	Resid
	NOBLUP	Marginal residual	$r_m = y - \widehat{\eta}_m$	ResidPA
PEARSON	BLUP	Pearson-type residual	$r / \sqrt{\widehat{\mathrm{Var}}[y\|\boldsymbol{\gamma}]}$	Pearson
	NOBLUP	Marginal Pearson-type residual	$r_m / \sqrt{\widehat{\mathrm{Var}}[y]}$	PearsonPA
STUDENT	BLUP	Studentized residual	$r / \sqrt{\widehat{\mathrm{Var}}[r]}$	Student
	NOBLUP	Studentized marginal residual	$r_m / \sqrt{\widehat{\mathrm{Var}}[r_m]}$	StudentPA
LCL	BLUP	Lower prediction limit for linear predictor		LCL
	NOBLUP	Lower confidence limit for marginal linear predictor		LCLPA
UCL	BLUP	Upper prediction limit for linear predictor		UCL
	NOBLUP	Upper confidence limit for marginal linear predictor		UCLPA
VARIANCE	BLUP	Conditional variance of response variable	$\widehat{\mathrm{Var}}[y\|\boldsymbol{\gamma}]$	Variance
	NOBLUP	Marginal variance of response variable	$\widehat{\mathrm{Var}}[y]$	VariancePA

You can use the following shortcuts to request statistics: PRED for PREDICTED, STD for STDERR, RESID for RESIDUAL, VAR for VARIANCE.

You can specify the following options of the OUTPUT statement after the slash (/).

ALLSTATS
requests that all statistics are computed. If you do not use a keyword to assign a name, the HPMIXED procedure uses the default name.

ALPHA=*number*
determines the coverage probability for two-sided confidence and prediction intervals. The coverage probability is computed as $1 - number$. The value of *number* must be between 0 and 1 inclusively; the default is 0.05.

NOMISS
requests that records from the input data set be written to the output data only for those observations that were used in the analysis. By default, the HPMIXED procedure produces output statistics for all observations in the input data set.

NOUNIQUE
requests that names not be made unique in the case of naming conflicts. By default, the HPMIXED procedure avoids naming conflicts by assigning a unique name to each output variable. If you specify the NOUNIQUE option, variables with conflicting names are not renamed. In that case, the first variable added to the output data set takes precedence.

NOVAR
requests that variables from the input data set not be added to the output data set. This option ignores ID statement but does not apply to variables listed in a BY statement.

PARMS Statement

PARMS < *(value-list) . . .* > < */ options* > **;**

The PARMS statement specifies initial values for the covariance parameters, or it requests a grid search over several values of these parameters. You must specify the values in the order in which they appear in the "Covariance Parameter Estimates" table.

The *value-list* specification can take any of several forms:

m	a single value
$m_1, m_2, \ldots, m_n$	several values
m to n	a sequence where m equals the starting value, n equals the ending value, and the increment equals 1
m to n by i	a sequence where m equals the starting value, n equals the ending value, and the increment equals i
m_1, m_2 to m_3	mixed values and sequences

You can use the PARMS statement to input known parameters. Suppose the three variance components are known to be 2, 1, and 3. The SAS statements to fix the variance components at these values are as follows:

```
proc hpmixed;
   class Family Gender;
   model Height = Gender;
   random Family Family*Gender;
   parms (2) (1) (3) / noiter;
run;
```

The NOPROFILE option in the PROC HPMIXED statement suppresses profiling the residual variance parameter during its calculations, thereby enabling its value to be held at 6 as specified in the PARMS statement.

If you specify more than one set of initial values, PROC HPMIXED performs a grid search of the likelihood surface and uses the best point on the grid for subsequent analysis. Specifying a large number of grid points can result in long computing times. The grid search feature is also useful for exploring the likelihood surface.

The results from the PARMS statement are the values of the parameters on the specified grid (denoted by CovP1–CovP*n*), the residual variance (possibly estimated) for models with a residual variance parameter, and various functions of the likelihood.

For ODS purposes, the name of the "Parameter Search" table is "ParmSearch."

You can specify the following options in the PARMS statement after a slash (/).

HOLD=*value-list*

HOLD

specifies which parameter values PROC HPMIXED should hold to equal the specified values. To hold all parameters, you can use the second form without giving the *value-list*. For example, the following statement constrains the first and third covariance parameters to equal 5 and 2, respectively.

Specifying the HOLD= option implies the NOPROFILE option in the PROC HPMIXED statement:

```
parms (5) (3) (2) (3) / hold=1,3;
```

LOWERB=*value-list*

enables you to specify lower boundary constraints on the covariance parameters. The *value-list* specification is a list of numbers or missing values (.) separated by commas. You must list the numbers in the order that PROC HPMIXED uses for the covariance parameters, and each number corresponds to the lower boundary constraint. A missing value instructs PROC HPMIXED to use its default constraint, and if you do not specify numbers for all of the covariance parameters, PROC MIXED assumes the remaining ones are missing.

NOITER

requests that no optimization iterations be performed and that PROC HPMIXED use the best value from the grid search to perform inferences. By default, iterations begin at the best value from the PARMS grid search. This option is ignored when you specify the HOLD= option.

If a residual variance is profiled, the parameter estimates can change from the initial values you provide as the residual variance is recomputed. To prevent an update of the residual variance, combine the NOITER option with the NOPROFILE option in the PROC HPMIXED statements, as in the following program:

```
proc hpmixed noprofile;
   class A B C rep mp sp;
   model y = A | B | C;
   random rep mp sp;
   parms (180) (200) (170) (1000) / noiter;
run;
```

Specifying the NOITER option in the PARMS statement has the same effect as specifying TECHNIQUE=NONE in the NLOPTIONS statement.

Notice that the NOITER option can be useful if you want to obtain the starting values HPMIXED computes. The following statements produce the starting values:

```
proc hpmixed noprofile;
   class A B;
   model y = A;
   random int / subject=B;
   parms / noiter;
run;
```

PARMSDATA=*SAS-data-set*

PDATA=*SAS data set*

reads in covariance parameter values from a SAS data set. The data set should contain the numerical variable ESTIMATE or the numerical variables Covp1–Covp$q$, where q denotes the number of covariance parameters.

If the PARMSDATA= data set contains multiple sets of covariance parameters, the HPMIXED procedure evaluates the initial objective function for each set and commences the optimization step by using the set with the lowest function value as the starting values. For example, the following SAS statements request that the objective function be evaluated for three sets of initial values:

```
data data_covp;
  input covp1-covp4;
  datalines;
  180 200 170 1000
  170 190 160  900
  160 180 150  800
;
proc hpmixed;
   class A B C rep;
   model yield = A;
   random rep B C;
   parms / pdata=data_covp;
run;
```

Each set comprises four covariance parameters.

The order of the observations in a data set with the numerical variable Estimate corresponds to the order of the covariance parameters in the "Covariance Parameter Estimates" table.

The PARMSDATA= data set must contain at least one set of covariance parameters with no missing values.

If the HPMIXED procedure is processing the input data set in BY groups, you can add the BY variables to the PARMSDATA= data set. If this data set is sorted by the BY variables, the HPMIXED procedure matches the covariance parameter values to the current BY group. If the PARMSDATA= data set does not contain all BY variables, the data set is processed in its entirety for every BY group and a message is written to the log. This enables you to provide a single set of starting values across BY groups, as in the following statements:

```
data data_covp;
  input covp1-covp4;
  datalines;
  180 200 170 1000
;
proc hpmixed;
   class A B C rep;
   model yield = A;
   random rep B C;
   parms / pdata=data_covp;
   by year;
run;
```

The same set of starting values is used for each value of the year variable.

UPPERB=*value-list*

enables you to specify upper boundary constraints on the covariance parameters. The *value-list* specification is a list of numbers or missing values (.) separated by commas. You must list the numbers in the order that PROC HPMIXED uses for the covariance parameters, and each number corresponds to the upper boundary constraint. A missing value instructs PROC HPMIXED to use its default constraint, and if you do not specify numbers for all of the covariance parameters, PROC HPMIXED assumes that the remaining ones are missing.

RANDOM Statement

RANDOM *random-effects* < / *options* > **;**

The RANDOM statement defines the random effects in the mixed model. It can be used to specify traditional variance component models (as in the VARCOMP procedure) and to specify random coefficients. The random effects can be classification or continuous. Multiple RANDOM statements are possible. Random effects specified in a RANDOM statement could be correlated with each other for certain types of covariance structures (see the TYPE= option on page 2831). It is, however, assumed that random effects specified using different RANDOM statements are not correlated.

Using notation from the section "Model Assumptions" on page 2832, the purpose of the RANDOM statement is to define the $\mathbf{Z}$ matrix of the mixed model, the random effects in the $\boldsymbol{\gamma}$ vector, and the

structure of $\mathbf{G}$. The $\mathbf{Z}$ matrix is constructed exactly like the $\mathbf{X}$ matrix for the fixed effects, and the $\mathbf{G}$ matrix is constructed to correspond to the effects constituting $\mathbf{Z}$. The structure of $\mathbf{G}$ is defined by using the TYPE= option described on page 2831.

You can specify INTERCEPT (or INT) as a random effect. PROC HPMIXED does not include the intercept in the RANDOM statement by default, as it does in the MODEL statement.

You can specify the following options in the RANDOM statement after a slash (/).

ALPHA=*number*

requests that a *t*-type confidence interval with confidence level $1-$*number* be constructed for the predictors of random effects in this statement. The value of *number* must be between 0 and 1 exclusively; the default is 0.05. Specifying the ALPHA= option implies the CL option.

CL

requests that *t*-type confidence limits be constructed for each of the predictors of random effects in this statement. The confidence level is 0.95 by default; this can be changed with the ALPHA= option. The CL option implies the SOLUTION option.

GROUP=*effect*

defines an effect specifying heterogeneity in the covariance structure of $\mathbf{G}$. All observations having the same level of the group effect have the same covariance parameters. Each new level of the group effect produces a new set of covariance parameters with the same structure as the original group. You should exercise caution in defining the group effect, because strange covariance patterns can result from its misuse. Also, the group effect can greatly increase the number of estimated covariance parameters, which can adversely affect the optimization process.

Continuous variables are permitted as arguments to the GROUP= option. PROC HPMIXED does not sort by the values of the continuous variable; rather, it considers the data to be from a new group whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of groups and also prevents the production of a large "Class Levels Information" table.

NOFULLZ

eliminates the columns in $\mathbf{Z}$ corresponding to missing levels of random effects involving CLASS variables. By default, these columns are included in $\mathbf{Z}$. It is sufficient to specify the NOFULLZ option in any RANDOM statement.

SOLUTION

requests that the solution for the random-effects parameters be produced. Using notation from the section "Model Assumptions" on page 2832, these estimates are the empirical best linear unbiased predictors (BLUPs) $\widehat{\boldsymbol{\gamma}} = \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{V}}^{-1}(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})$. They can be useful for comparing the random effects from different experimental units and can also be treated as residuals in performing diagnostics for your mixed model.

The numbers displayed in the SE Pred column of the "Solution for Random Effects" table are not the standard errors of the $\widehat{\boldsymbol{\gamma}}$ displayed in the Estimate column; rather, they are the standard errors of predictions $\widehat{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i$, where $\widehat{\boldsymbol{\gamma}}_i$ is the ith BLUP and $\boldsymbol{\gamma}_i$ is the ith random-effect parameter.

SUBJECT=*effect*

identifies the subjects in your mixed model. Complete independence is assumed across subjects; thus, for the RANDOM statement, the SUBJECT= option produces a block-diagonal structure in **G** with identical blocks. The **Z** matrix is modified to accommodate this block-diagonality. In fact, specifying a subject effect is equivalent to nesting all other effects in the RANDOM statement within the subject effect.

Continuous variables are permitted as arguments to the SUBJECT= option. PROC HPMIXED does not sort by the values of the continuous variable; rather, it considers the data to be from a new subject whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of subjects and also prevents the production of a large "Class Levels Information" table.

TYPE=*covariance-structure*

specifies the covariance structure of **G** for G-side effects.

The TYPE=VC (variance components) option is the default structure. Another structure available in this experimental release is CHOL.

TEST Statement

TEST *fixed-effects* < / *options* > **;**

The TEST statement performs a hypothesis test on the fixed effects. You can specify multiple effects in one TEST statement or in multiple TEST statements, and all TEST statements must appear after the MODEL statement.

You can specify the following options in the TEST statement after a slash (/).

HTYPE=*value-list*

indicates the type of hypothesis test to perform on the specified effects. Valid entries for values in the *value-list* are 3, corresponding to a Type III test. The default value is 3. The ODS table name is "Tests3" for the Type III test.

E

requests that matrix coefficients associated with test types be displayed for specified effects.

E3 | EIII

requests that Type III matrix coefficients be displayed if a Type III test is performed.

CHISQ

requests that χ^2 tests be performed in addition to any F tests. A χ^2 statistic equals its corresponding F statistic times the associate numerator degree of freedom, and this same degree of freedom is used to compute the p-value for the χ^2 test. This p-value will always be less than that for the F test, because it effectively corresponds to an F test with infinite denominator degrees of freedom.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement replaces $\mathbf{R}$ with $\mathbf{W}^{-1/2}\mathbf{R}\mathbf{W}^{-1/2}$, where $\mathbf{W}$ is a diagonal matrix containing the weights. Observations with nonpositive or missing weights are not included in the resulting PROC HPMIXED analysis. If a WEIGHT statement is not included, all observations used in the analysis are assigned a weight of 1.

Details: HPMIXED Procedure

Model Assumptions

The following sections provide an overview of the approach used by the HPMIXED procedure for likelihood-based analysis of linear mixed models with sparse matrix technique. Additional theory and examples are provided in Littell et al. (1996), Verbeke and Molenberghs (1997, 2000), and Brown and Prescott (1999).

The HPMIXED procedure fits models generally of the form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

Models of this form contain both fixed-effects parameters, $\boldsymbol{\beta}$, and random-effects parameters, $\boldsymbol{\gamma}$; hence, they are called *mixed models*. Refer to Henderson (1990) and Searle, Casella, and McCulloch (1992) for historical developments of the mixed model. Note that the matrix $\mathbf{Z}$ can contain either continuous or dummy variables, just like $\mathbf{X}$.

So far this is the same general form of model fit by the MIXED procedure. The difference between the models handled by the two procedures lies in the assumptions about the distributions of $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$. For both procedures a key assumption is that $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$ are normally distributed with

$$\mathrm{E}\begin{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$
$$\mathrm{Var}\begin{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix}$$

The two procedures differ in their assumptions about the variance matrices $\mathbf{G}$ and $\mathbf{R}$ for $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$, respectively. The MIXED procedure allows a variety of different structures for both $\mathbf{G}$ and $\mathbf{R}$; while in HPMIXED procedure, $\mathbf{R}$ is always assumed to be of the form $\mathbf{R} = \mathbf{I}\sigma^2$, and the structures available for modeling $\mathbf{G}$ are only a small subset of the structures offered by the MIXED procedure.

Estimates of fixed effects and predictions for random effects are obtained by solving the so-called *mixed model equations*:

$$\begin{bmatrix} \mathbf{X}'\mathbf{X}/\sigma^2 & \mathbf{X}'\mathbf{Z}/\sigma^2 \\ \mathbf{Z}'\mathbf{X}/\sigma^2 & \mathbf{Z}'\mathbf{Z}/\sigma^2 + \mathbf{G}^{-1} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\mathbf{y}/\sigma^2 \\ \mathbf{Z}'\mathbf{y}/\sigma^2 \end{bmatrix}$$

Let $\mathbf{C}$ denote the coefficient matrix of the mixed model equations:

$$\mathbf{C} = \begin{bmatrix} \mathbf{X}'\mathbf{X}/\sigma^2 & \mathbf{X}'\mathbf{Z}/\sigma^2 \\ \mathbf{Z}'\mathbf{X}/\sigma^2 & \mathbf{Z}'\mathbf{Z}/\sigma^2 + \mathbf{G}^{-1} \end{bmatrix}$$

Under the assumptions given previously for the moments of $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$, the variance of $\mathbf{y}$ is $\mathbf{V} = \mathbf{Z}\mathbf{G}\mathbf{Z}' + \mathbf{I}\sigma^2$. You can model $\mathbf{V}$ by setting up the random-effects design matrix $\mathbf{Z}$ and by specifying covariance structures for $\mathbf{G}$. Let $\boldsymbol{\theta}$ be a vector of all unknown parameters in $\mathbf{G}$. Then the general form of the restricted likelihood function for the mixed models that the HPMIXED procedure can fit is

$$L(\boldsymbol{\theta}, \sigma^2) = -2 \log l = (n-p)\log(2\pi) + \log|\mathbf{C}| + \log|\mathbf{G}| + n\log(\sigma^2) + \mathbf{y}'\mathbf{P}\mathbf{y}$$

where

$$\mathbf{P} = \mathbf{V}^{-1} - \mathbf{V}^{-1}\mathbf{X}(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-}\mathbf{X}'\mathbf{V}^{-1}$$

and p is the rank of $\mathbf{X}$. The HPMIXED procedure minimizes $L(\boldsymbol{\theta}, \sigma^2)$ over all unknown parameters in $\boldsymbol{\theta}$ and σ^2 by using nonlinear optimization algorithms.

Computing and Maximizing the Likelihood

In computing the restricted likelihood function given previously, the determinants of the matrices $\mathbf{C}$ and $\mathbf{G}$ can be obtained effectively by using Cholesky decomposition. The quadratic term $\mathbf{y}'\mathbf{P}\mathbf{y}$ can be expressed in terms of solutions of mixed model equations as follows:

$$\mathbf{y}'\mathbf{P}\mathbf{y} = \frac{1}{\sigma^2}(\mathbf{y}'\mathbf{y} - \left[\widehat{\boldsymbol{\beta}}', \widehat{\boldsymbol{\gamma}}'\right] \begin{bmatrix} \mathbf{X}'\mathbf{y} \\ \mathbf{Z}'\mathbf{y} \end{bmatrix})$$

By default, the HPMIXED procedure profiles out the residual variance σ^2 from the parameter vector $\boldsymbol{\theta}$. Let $\boldsymbol{\theta}^*$ be the new parameter vector such that $\theta_i^* = \theta_i/\sigma^2$. The profiled objective function becomes

$$L(\boldsymbol{\theta}^*, \sigma^2) = (n-p)\log(2\pi) + \log|\mathbf{C}^*| + \log|\mathbf{G}^*| - (r_C - r_G - n)\log(\sigma^2) + (n-p)$$

where $\mathbf{C}^* = \mathbf{C}\sigma^2$ and $\mathbf{G}^* = \mathbf{G}\sigma^2$ are the profiled versions of $\mathbf{C}$ and $\mathbf{G}$, r_C and r_G are the ranks of $\mathbf{C}$ and $\mathbf{G}$. Minimizing analytically for σ^2 yields

$$\widehat{\sigma}^2 = \frac{1}{n-p}(\mathbf{y}'\mathbf{y} - \left[\widehat{\boldsymbol{\beta}}', \widehat{\boldsymbol{\gamma}}'\right] \begin{bmatrix} \mathbf{X}'\mathbf{y} \\ \mathbf{Z}'\mathbf{y} \end{bmatrix})$$

Optimizing the likelihood calls for derivatives with respect to the parameters. The first and second derivatives of the log-likelihood function L with respect to scalar variance components θ_i and θ_j are

$$\frac{\partial L}{\partial \theta_i} = \mathrm{tr}\left(\frac{\partial \mathbf{V}}{\partial \theta_i}\mathbf{P}\right) - \mathbf{y}'\mathbf{P}\frac{\partial \mathbf{V}}{\partial \theta_i}\mathbf{P}\mathbf{y}$$

and

$$\frac{\partial^2 L}{\partial\theta_i\theta_j} = -\mathrm{tr}\left(\frac{\partial \mathbf{V}}{\partial\theta_i}\mathbf{P}\frac{\partial \mathbf{V}}{\partial\theta_j}\mathbf{P}\right) + 2\mathbf{y}'\mathbf{P}\frac{\partial \mathbf{V}}{\partial\theta_i}\mathbf{P}\frac{\partial \mathbf{V}}{\partial\theta_j}\mathbf{P}\mathbf{y}$$

The default quasi-Newton method of optimization for the HPMIXED procedure requires only first derivatives of the log likelihood, and these are readily derived by solving the mixed model equations. For example, when $\mathbf{G} = \mathbf{I}\sigma_a$, the first derivative of the log likelihood with respect to the parameter σ_a^2 can be computed as follows:

$$\frac{\partial L}{\partial\sigma_a^2} = \frac{q}{\sigma_a^2} - \frac{\mathrm{tr}(\mathbf{C}^{aa})}{\sigma_a^4} - \frac{\widehat{\boldsymbol{\gamma}}'\widehat{\boldsymbol{\gamma}}}{\sigma_a^4}$$

where q is the size of $\boldsymbol{\gamma}$ vector and $\mathbf{C}^{aa}$ is the part of the g-inverse of the mixed model equation coefficient matrix $\mathbf{C}$ corresponding to the random effect $\boldsymbol{\gamma}$.

The second derivative of the log likelihood needs to be computed only if you specify certain nondefault optimization techniques in the NLOPTIONS statement, namely TECH=NEWRAP, TECH=NRRIDG, or TECH=TRUREG; see "Nonlinear Optimization: The NLOPTIONS Statement" on page 391 in Chapter 18, "Shared Concepts and Topics," for more information about optimization techniques. For these second-derivative-based optimization techniques, the HPMIXED procedure does not actually use the true second derivative matrix, or *observed information matrix*, as defined earlier. Instead, it uses an alternative matrix that is more efficient to compute for large problems and that can be more stable. This alternative is called the *average information* matrix, and it is defined as follows. The expected value of the second derivative is

$$\mathbf{E}\left(\frac{\partial^2 L}{\partial\theta_i\theta_j}\right) = \mathrm{tr}\left(\frac{\partial \mathbf{V}}{\partial\theta_i}\mathbf{P}\frac{\partial \mathbf{V}}{\partial\theta_j}\mathbf{P}\right)$$

It is this trace that is computationally inefficient to evaluate. But if you average the expected information matrix defined by this formula with the observed information matrix defined by the preceding formula for the true second derivative, then the trace term cancels, leaving just a quadratic expression in $\mathbf{y}$. This quadratic expression defines the average information (Johnson and Thompson 1995) with respect to θ_i and θ_j:

$$\mathrm{AI}(\theta_i,\theta_j) = \mathbf{y}'\mathbf{P}\frac{\partial \mathbf{V}}{\partial\theta_i}\mathbf{P}\frac{\partial \mathbf{V}}{\partial\theta_j}\mathbf{P}\mathbf{y}$$

Computing Starting Values by EM-REML

The EM-REML algorithm (Dempster, Laird, and Rubin 1977) iteratively alternates between an expectation step and a maximization step to maximize the restricted log likelihood. The algorithm is based on augmenting the observed data $\mathbf{y}$ with the unobservable random effects $\boldsymbol{\gamma}$, leading to a simplified form for the log likelihood. For example, if $\mathbf{G} = \mathbf{I}\sigma_a^2$ then given the realized values $\tilde{\boldsymbol{\gamma}}$ of the unobservable random effects $\boldsymbol{\gamma}$, the REML estimate of σ_a^2 satisfies

$$\widehat{\sigma}_a^2 = \frac{\tilde{\boldsymbol{\gamma}}'\tilde{\boldsymbol{\gamma}}}{q - \sigma^2/\sigma_a^2\mathrm{tr}(\mathbf{C}^{aa})}$$

This corresponds to the maximization step of EM-REML. However, the true realized values $\tilde{\boldsymbol{\gamma}}$ are unknown in practice. The expectation step of EM-REML replaces them with the conditional expected values $\widehat{\boldsymbol{\gamma}}$ of the random effects, given the observed data $\mathbf{y}$ and initial values for the parameters. The new estimate of σ_a^2 is used in turn to recalculate the conditional expected values, and the iteration is repeated until convergence.

It is well known that EM-REML is generally more robust against a poor choice of starting values than general nonlinear optimization methods such as Newton-Raphson, though it tends to converge slowly as it approaches the optimum. The Newton-Raphson method, on the other hand, converges much faster when it has a good set of starting values. The HPMIXED procedure, thus, employs a scheme that uses EM-REML initially in order to get good starting values, and after a few iterations, when the decrease in log likelihood has significantly slowed down, switching to a more general nonlinear optimization technique (by default, quasi-Newton).

Sparse Matrix Techniques

A key component of the HPMIXED procedure is the use of sparse matrix techniques for computing and optimizing the likelihood expression given in the section "Model Assumptions" on page 2832. There are two aspects to sparse matrix techniques, namely, sparse matrix storage and sparse matrix computations. Typically, computer programs represent an $N \times M$ matrix in a dense form as an array of size NM, making row-wise and column-wise arithmetic operations particularly efficient to compute. However, if many of these NM numbers are zeros, then correspondingly many of these operations are unnecessary or trivial. Sparse matrix techniques exploit this fact by representing a matrix not as a complete array, but as a set of nonzero elements and their location (row and column) within the matrix. Sparse matrix techniques are more efficient if there are enough zero-element operations in the dense form to make the extra time required to find and operate on matrix elements in the sparse form worthwhile.

The following discussion illustrates sparse techniques. Let the symmetric matrix $\mathbf{C}$ be the matrix of mixed model equations of size 5×5.

$$\mathbf{C} = \begin{bmatrix} 8.0 & 0 & 0 & 2.0 & 0 \\ 0 & 4.0 & 3.0 & 0 & 0 \\ 0 & 3.0 & 5.0 & 0 & 0 \\ 2.0 & 0 & 0 & 7.0 & 0 \\ 0 & 0 & 0 & 0 & 9.0 \end{bmatrix}$$

There are 15 elements in the upper triangle of $\mathbf{C}$, though eight of them are zeros. The row and column indices and the values of seven nonzero elements are listed as follows:

i	1	1	2	2	3	4	5
j	1	4	2	3	3	4	5
C_{ij}	8.0	2.0	4.0	3.0	5.0	7.0	9.0

The most elegant scheme to store these seven elements is to store them in a hash table with row and column indices as a hash key. However, this scheme is not efficient as the number of nonzero elements gets very large. The classical and widely used scheme, and the one the HPMIXED

procedure employs, is the (ic, jc, c) format, in which the nonzero elements are stored contiguously row by row in the vector c. To identify the individual nonzero elements in each row, you need to know the column index of an element. These column indices are stored in the vector jc; that is, if $c(k) = C_{ij}$, then $jc(k) = j$. To identify the individual rows, you need to know where each row starts and ends. These row starting positions are stored in the vector ic. For instance, if C_{ij} is the first nonzero element in the row i and $c(k) = C_{ij}$, then $ic(i) = k$. The row i ending position is one less than $ic(i+1)$. Thus, the number of nonzero elements in the row i is $ic(i+1) - ic(i)$, these elements in the row i are stored consecutively starting from the position $k_i = ic(i)$

$$c(k_i), c(k_i+1), c(k_i+2), ..., c(k_{i+1}-1)$$

and the corresponding columns indices are stored consecutively in

$$jc(k_i), jc(k_i+1), jc(k_i+2), ..., jc(k_{i+1}-1)$$

For example, the seven nonzero elements in matrix $\mathbf{C}$ are stored in (ic, jc, c) format as

ic	1	3	5	6	7	8	
jc	1	4	2	3	3	4	5
c	8.0	2.0	4.0	3.0	5.0	7.0	9.0

Note that since matrices are stored row by row in the (ic, jc, c) format, row-wise operations can be performed efficiently but it is inefficient to retrieve elements column-wise. Thus, this representation will be inefficient for matrix computations requiring column-wise operations. Fortunately, the likelihood calculations for mixed models can usually avoid column-wise operations.

In mixed models, sparse matrices typically arise from a large number of levels for fixed effects and/or random effects. If a linear model contains one or more large CLASS effects, then the mixed model equations are usually very sparse. Storing zeros in mixed model equations not only requires significantly more memory but also results in longer execution time and larger rounding error. As an illustration, the example in the "Getting Started: HPMIXED Procedure" on page 2808 has 3506 mixed model equations. Storing just the upper triangle of these equations in a dense form requires $(1 + 3506) \times 3506/2 = 6,147,771$ elements. However, there are only 60,944 nonzero elements—less than 1% of what dense storage requires.

Note that as the density of the mixed model equations increases, the advantage of sparse matrix techniques decreases. For instance, a classical regression model typically has a dense coefficient matrix, though the dimension of the matrix is relatively small.

The HPMIXED procedure employs sparse matrix techniques to store the nonzero elements in the mixed model equations and to compute a sparse Cholesky decomposition of these equations. A reordering of the mixed model equations is required in order to keep the minimum memory consumption during the factorization. This reordering process results in a different g-inverse from what is produced by most other SAS/STAT procedures, for which the g-inverse is defined by sequential sweeping in the order defined by the model. If mixed model equations are singular, this different g-inverse produces a different solution of mixed model equations. However, estimable functions and tests based on them are invariant to the choice of g-inverse, and are thus the same for the HPMIXED procedure as for other procedures.

Hypothesis Tests for Fixed Effects

Unlike most other SAS/STAT procedures for analyzing general linear models, the HPMIXED procedure does not by default provide F tests for the fixed effects. This is because, for the large mixed model problems that the HPMIXED procedure is designed to address, such tests are often computationally prohibitive to compute. The computation of Type III tests first constructs the Hermite matrix of the mixed model coefficient matrix $\mathbf{C}$ and then forms the $\mathbf{L}$ coefficient matrix to obtain the F value as follows:

$$F = \frac{\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}' \mathbf{L}'(\mathbf{L}\widehat{\mathbf{C}}^{-1}\mathbf{L}')^{-1}\mathbf{L} \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}}{r}$$

where $r = \text{rank}(\mathbf{L}\widehat{\mathbf{C}}^{-1}\mathbf{L}')$. The coefficient matrix $\mathbf{L}$ corresponding to fixed effects with many levels can be very large and dense, making them very difficult to work with. At the same time, Type III tests for effects with many levels are relatively unlikely to be statistically useful.

For this reason, you must use the TEST statement in PROC HPMIXED to specifically ask for Type III tests for any effects for which you want to compute them. An example of this is given in the section "Getting Started: HPMIXED Procedure" on page 2808. No other types of test (namely, I, II, or IV) are available in the HPMIXED procedure in this experimental release.

Default Output

The following sections describe the output PROC HPMIXED produces by default. This output is organized into various tables, and they are discussed in order of appearance.

Model Information

The "Model Information" table describes the model, some of the variables it involves, and the method used in fitting it. It also lists the method for computing the degrees of freedom.

For ODS purposes, the name of the "Model Information" table is "ModelInfo."

Class Level Information

The "Class Level Information" table lists the first 20 levels of every variable specified in the CLASS statement. You should check this information to make sure the data are correct. You can adjust the order of the CLASS variable levels with the ORDER= option in the PROC HPMIXED statement. For ODS purposes, the name of the "Class Level Information" table is "ClassLevels."

Dimensions

The "Dimensions" table lists the sizes of relevant matrices. This table can be useful in determining CPU time and memory requirements. For ODS purposes, the name of the "Dimensions" table is "Dimensions."

Number of Observations

The "Number of Observations" table shows the number of observations read from the data set and the number of observations used in fitting the model.

Descriptive Statistics

The "Descriptive Statistics" table lists simple statistics such as means and standard deviations for the dependent variable and for each covariate in the MODEL statement.

Iteration History

The "Iteration History" table describes the optimization of the residual log likelihood. The function to be minimized (the *objective function*) is $-2l$.

For ODS purposes, the name of the "Iteration History" table is "IterHistory."

Covariance Parameter Estimates

The "Covariance Parameter Estimates" table contains the estimates of the parameters in $\mathbf{G}$ and $\mathbf{R}$. Their values are labeled in the "Cov Parm" table along with Subject and Group information if applicable. The estimates are displayed in the Estimate column.

For ODS purposes, the name of the "Covariance Parameter Estimates" table is "CovParms."

Convergence Status

The "Convergence Status" table contains a status message that describes the reason the optimization terminated. The message is also written to the log. For ODS purposes, the name of the "Convergence Status" table is "ConvergenceStatus." You can query the nonprinting numeric variable Status to check for a successful optimization. This is useful in batch processing, or when processing BY groups, such as in simulations. Successful optimizations are indicated by the value 0 for the Status variable.

Fit Statistics

The "Fit Statistics" table provides some statistics about the estimated mixed model.

In addition, the "Fit Statistics" table lists three information criteria: AIC, AICC, and BIC, all in smaller-is-better form. Expressions for these criteria are described under the IC option on page 2811.

For ODS purposes, the name of the "Model Fitting Information" table is "FitStatistics."

ODS Table Names

Each table created by PROC HPMIXED has a name associated with it, and you must use this name to reference the table when using ODS statements. These names are listed in Table 43.3.

Table 43.3 ODS Tables Produced by PROC HPMIXED

Table Name	Description	Required Statement / Option
OverallANOVA	ANOVA table for model without random effect	default output for fixed models
ClassLevels	level information from the CLASS statement	default output
Coef	**L** matrix coefficients	E option in MODEL, CONTRAST, ESTIMATE, or LSMEANS
Contrasts	results from the CONTRAST statements	CONTRAST
ConvergenceStatus	convergence status	default
CovParms	estimated covariance parameters	default output
Dimensions	dimensions of the model	default output
Estimates	results from ESTIMATE statements	ESTIMATE
FitStatistics	fit statistics	default
IterHistory	iteration history	default output
LSMeans	LS-means	LSMEANS
MMEq	mixed model equations	PROC HPMIXED MMEQ
ModelInfo	model information	default output
NObs	number of observations read and used	default output
OptInfo	optimization information	default output
ParameterEstimates	fixed-effects solution	MODEL/SOLUTION
ParmSearch	parameter search values	PARMS
SimpleStatistics	descriptive statistics for dependent variable and covariate variables	default output
SolutionR	random-effect solution vector	RANDOM/SOLUTION
Tests3	Type III tests of fixed effects	TEST

Examples: HPMIXED Procedure

Example 43.1: Ranking Many Random-Effect Coefficients

In analyzing models with random effects that have many levels, a frequent goal is to estimate and rank the predicted values of the coefficients corresponding to these levels. For example, in mixed models for animal breeding, the predicted coefficient of the random effect for each animal is referred to as the *estimated breeding value* (EBV) and animals with relatively high EBVs are chosen for breeding. This example demonstrates the use of the HPMIXED procedure for computing EBVs and their precision. Although other mixed modeling tools in SAS/STAT can potentially compute EBVs, PROC HPMIXED is particularly suited for the large, sparse matrix calculations involved. The typical performance of the HPMIXED procedure and other tools for this problem is also discussed.

The data for this problem are generated by simulation. Suppose you are considering analyzing EBVs for animals on 15 farms, with about 100 animals of 5 different species on each farm. The following DATA step simulates data with this structure, where about 40 observations of the response variable Yield are made per animal:

```
%let NFarm = 15;
%let NAnimal = %eval(&NFarm*100);
data Sim;
   keep Species Farm Animal Yield;
   array BV{&NAnimal};
   array AnimalSpecies{&NAnimal};
   array AnimalFarm{&NAnimal};
   do i = 1 to &NAnimal;
      BV           {i} = sqrt(4.0)*rannor(12345);
      AnimalSpecies{i} = 1 + int(   5  *ranuni(12345));
      AnimalFarm   {i} = 1 + int(&NFarm*ranuni(12345));
      end;
   do i = 1 to 40*&NAnimal;
      Animal  = 1 + int(&NAnimal*ranuni(12345));
      Species = AnimalSpecies{Animal};
      Farm    = AnimalFarm   {Animal};
      Yield   = 1 + Species
                  + Farm
                  + BV{Animal}
                  + sqrt(8.0)*rannor(12345);
      output;
      end;
run;
```

In this simulation, the true breeding value (BV1-BV1500) for each animal has a variance component of 4.0, while the level of background variance is 8.0.

In this type of experiment, the effect of Species and the interaction between Species and Farm are typically modeled as fixed effects, while the effect of Animal is modeled as a random effect. The

following statements use the HPMIXED procedure to compute predictions for the Animal random effect and save them to the data set EBV. This data set is then sorted and the 10 animals with the highest EBVs are displayed.

```
ods listing close;
proc hpmixed data=Sim;
   class Species Farm Animal;
   model Yield = Species Farm*Species;
   random Animal/cl;
   ods output SolutionR=EBV;
run;
ods listing;
proc sort data=EBV;
   by descending estimate;
proc print data=EBV(obs=10) noobs;
   var Animal Estimate StdErrPred Lower Upper;
run;
```

The preceding statements close the ODS listing destination for the duration of the PROC HPMIXED run. This avoids displaying the long random-effects solution table, since only the top few EBVs are of interest. Output 43.1.1 displays the EBVs of the top 10 animals, along with their precision and confidence bounds.

Output 43.1.1 Estimated Breeding Values: Top 10 Animals

Animal	Estimate	StdErr Pred	Lower	Upper
1294	5.9703	0.6317	4.7321	7.2085
1219	5.0081	0.6396	3.7544	6.2618
1054	4.9452	0.5874	3.7939	6.0966
758	4.9340	0.6196	3.7195	6.1485
986	4.9329	0.5767	3.8025	6.0633
1150	4.7444	0.5806	3.6064	5.8824
962	4.6651	0.5794	3.5294	5.8008
225	4.5294	0.6137	3.3266	5.7322
1252	4.5012	0.5686	3.3868	5.6157
1033	4.4971	0.6080	3.3054	5.6889

Notice that animal 1294 is ranked as the top animal based on its EBV, but the precision of this estimate, as measured by the standard error of prediction, is lower than that of other animals.

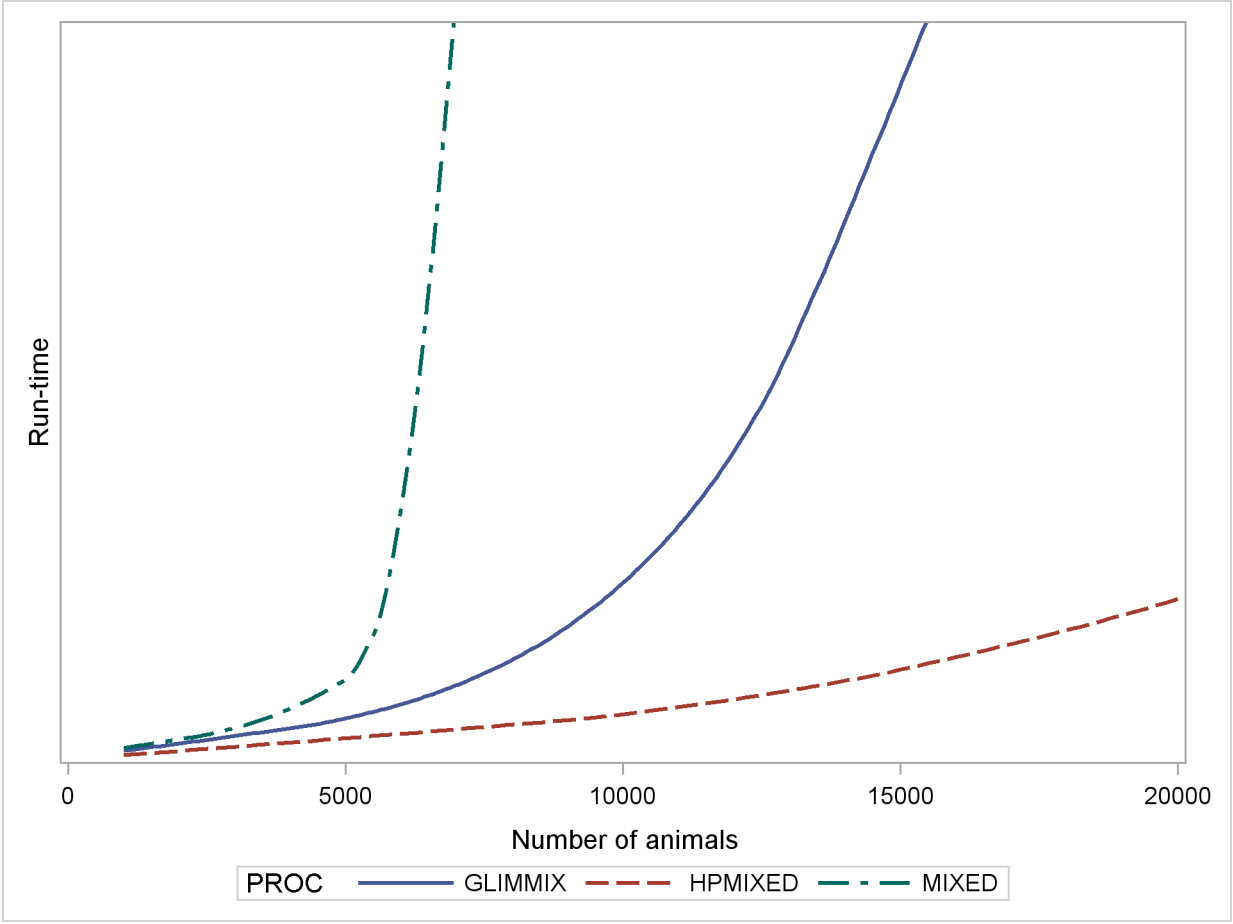
You can also use PROC MIXED and PROC GLIMMIX to compute EBVs, but the performance of these general mixed modeling procedures for this specialized kind of data and model is quite different from that of PROC HPMIXED. The MIXED and GLIMMIX procedures are engineered to have good performance properties across a broad class of models and analyses, a class much broader than what PROC HPMIXED can handle. The HPMIXED procedure, on the other hand, can have better performance, in terms of both memory and run time, for certain specialized models and analyses, of which the current example is one.

For this example, an equivalent PROC GLIMMIX approach can take twice as long to complete, and PROC MIXED three times as long. Precise relative timings are not feasible, since those of the MIXED and GLIMMIX procedures are sensitive to the speed of disk access for writing to and

reading from the utility file that holds the underlying matrices. But the results on any system would be similar: for the limited class of models to which it applies, the sparse matrix representation that the HPMIXED procedure employs should provide better computational performance than a dense representation, in terms of both run time and memory use.

Moreover, for a given analysis, if the size of the problem is increased in such a way that the underlying matrices become sparser, the relative performance of PROC HPMIXED gets even better. As an illustration of this, Output 43.1.2 shows relative performance of the three procedures for simulated data as the number of farms increases. For this plot, each additional farm adds 500 levels of the Animal random effect to the model—a substantial number.

Output 43.1.2 Comparing Mixed Model Tools for Increasingly Sparse Problems

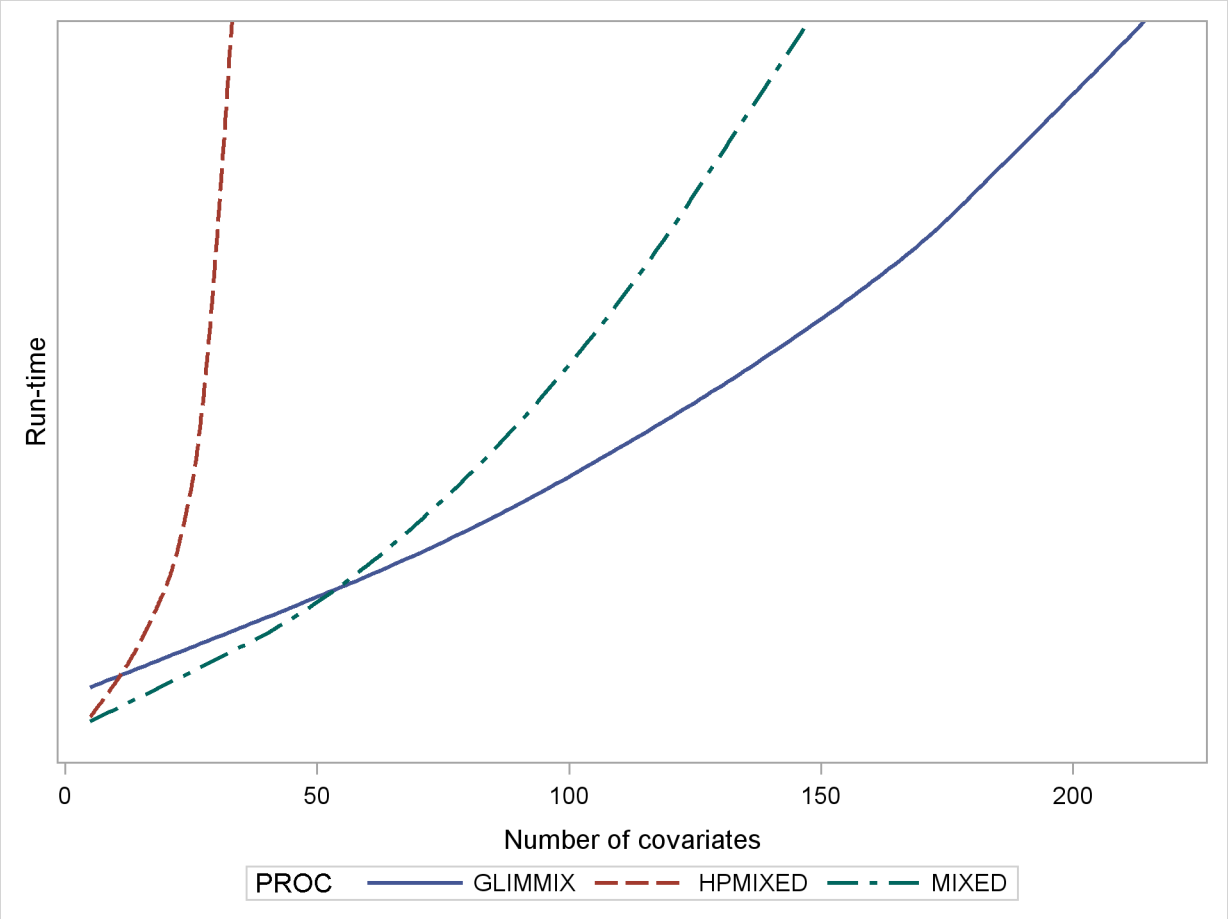


The vertical axis in Output 43.1.2 measures run time, but the units are omitted: relative performance is what counts, and that is expected to be fairly invariant to machine architecture. The output shows that while the performance of the MIXED and GLIMMIX procedures is relatively competitive with PROC HPMIXED for up to 3000 or 4000 animals, both procedures' relative performance decreases as the number of animals increases into the tens of thousands.

As a caveat, note that PROC HPMIXED can be *inefficient* relative to PROC MIXED and PROC GLIMMIX for models and data that are not sparse, because it can take many times longer to invert a large, dense matrix by sparse techniques. For example, Output 43.1.3 shows relative performance

of the three procedures for simulated data like the preceding, but where the fixed part of the model consists of an increasing number of continuous covariates and is thus dense.

Output 43.1.3 Comparing Mixed Model Tools for Increasingly Dense Problems



As before, the HPMIXED procedure is more efficient than the MIXED and GLIMMIX procedures for few covariates, but when the fixed-effect calculations dominate the run time, PROC HPMIXED rapidly becomes relatively inefficient as the size of the dense fixed-effect matrix increases. Also note that while PROC MIXED is more efficient than PROC GLIMMIX for small to moderate numbers of covariates, PROC GLIMMIX has the best performance as the number of covariates get very large.

Example 43.2: Comparing Results from PROC HPMIXED and PROC MIXED

This example revisits the mixed model problem from the section "Getting Started: MIXED Procedure" on page 3896, in Chapter 56, "The MIXED Procedure," with the data set shown in the following statements:

```
data heights;
  input Family Gender$ Height @@;
  datalines;
1 F 67   1 F 66   1 F 64   1 M 71   1 M 72   2 F 63
2 F 63   2 F 67   2 M 69   2 M 68   2 M 70   3 F 63
3 M 64   4 F 67   4 F 66   4 M 67   4 M 67   4 M 69
;
```

The response variable Height measures the heights (in inches) of 18 individuals. The individuals are classified according to Family and Gender. The following statements fit a mixed model with random effects for Family and the Family*Gender interaction with the MIXED procedure:

```
proc mixed;
   class Family Gender;
   model Height = Gender / s;
   random Family Family*Gender / s;
run;
```

The "Iteration History" and "Fit Statistics" tables for the optimization in PROC MIXED are shown in Output 43.2.1. The MIXED procedure converges after six iterations and achieves a –2 restricted log likelihood of 71.02245.

Output 43.2.1 Iteration History and Fit Statistics: MIXED Procedure

The Mixed Procedure

Iteration History

Iteration	Evaluations	-2 Res Log Like	Criterion
0	1	74.11074833	
1	2	71.51614003	0.01441208
2	1	71.13845990	0.00412226
3	1	71.03613556	0.00058188
4	1	71.02281757	0.00001689
5	1	71.02245904	0.00000002
6	1	71.02245869	0.00000000

Fit Statistics

-2 Res Log Likelihood	71.0
AIC (smaller is better)	77.0
AICC (smaller is better)	79.0
BIC (smaller is better)	75.2

Output 43.2.2 displays the covariance parameter estimates and the solutions for the fixed and random effects. Because the fixed-effect model contains a classification effect (Gender) and an intercept, the $\mathbf{X}'\mathbf{X}$ matrix is singular. Only two fixed-effect parameters can be estimated in this model. The MIXED procedure, relying on a sweep operation in the order in which effects enter the model, determines that the last column of the $\mathbf{X}'\mathbf{X}$ matrix is a linear function of previous columns. Consequently, the coefficient for the second level of the Gender variable is zero.

Output 43.2.2 Parameter Estimates and Solutions: MIXED Procedure

Covariance Parameter Estimates

Cov Parm	Estimate
Family	2.4010
Family*Gender	1.7657
Residual	2.1668

Solution for Fixed Effects

Effect	Gender	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept		68.2114	1.1477	3	59.43	<.0001
Gender	F	−3.3621	1.1923	3	−2.82	0.0667
Gender	M	0	.	.	.	.

Solution for Random Effects

Effect	Gender	Family	Estimate	Std Err Pred	DF	t Value	Pr > |t|
Family		1	1.2680	1.1201	10	1.13	0.2840
Family		2	0.08980	1.1121	10	0.08	0.9372
Family		3	−1.6660	1.1712	10	−1.42	0.1853
Family		4	0.3082	1.1201	10	0.28	0.7888
Family*Gender	F	1	−0.3198	1.0810	10	−0.30	0.7734
Family*Gender	M	1	1.2523	1.0933	10	1.15	0.2787
Family*Gender	F	2	−0.4299	1.0774	10	−0.40	0.6983
Family*Gender	M	2	0.4959	1.0774	10	0.46	0.6551
Family*Gender	F	3	−0.08229	1.1409	10	−0.07	0.9439
Family*Gender	M	3	−1.1429	1.1409	10	−1.00	0.3401
Family*Gender	F	4	0.8320	1.0933	10	0.76	0.4642
Family*Gender	M	4	−0.6053	1.0810	10	−0.56	0.5878

The "Type 3 Tests of Fixed Effects" table in Output 43.2.3 is produced by the MIXED procedure by default.

Output 43.2.3 Test of Gender Effect

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Gender	1	3	7.95	0.0667

The same linear mixed model is fit with the HPMIXED procedure with the following statements:

```
proc hpmixed;
   class Family Gender;
   model Height = Gender / s;
   random Family Family*Gender / s;
   test gender;
run;
```

Output 43.2.4 displays the "Iteration History" and "Fit Statistics" tables. The HPMIXED procedure, with its default quasi-Newton algorithm, achieves the same –2 restricted log likelihood as the MIXED procedure (71.02245; see Output 43.2.1).

Output 43.2.4 Iteration History and Fit Statistics: HPMIXED Procedure

The HPMIXED Procedure

Iteration History

Iteration	Evaluations	Objective Function	Change	Max Gradient
0	4	71.023177956	.	0.034074
1	3	71.022519936	0.00065802	0.007839
2	3	71.022477283	0.00004265	0.004674
3	2	71.0224587	0.00001858	0.000168
4	2	71.022458689	0.00000001	3.28E-6

Fit Statistics

-2 Res Log Likelihood	71.02246
AIC (smaller is better)	77.02246
AICC (smaller is better)	79.02246
BIC (smaller is better)	75.18134
CAIC (smaller is better)	78.18134
HQIC (smaller is better)	72.98226

Output 43.2.5 displays the results that correspond to those in Output 43.2.2 in the MIXED procedure.

Output 43.2.5 Parameter Estimates and Solutions: HPMIXED Procedure

Covariance Parameter Estimates

Cov Parm	Estimate
Family	2.4010
Family*Gender	1.7657
Residual	2.1668

Output 43.2.5 *continued*

Solution for Fixed Effects

Effect	Gender	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept		0	.	.	.	.
Gender	F	64.8493	1.1477	16	56.50	<.0001
Gender	M	68.2114	1.1477	16	59.43	<.0001

Solution for Random Effects

Effect	Gender	Family	Estimate	Std Err Pred	DF	t Value	Pr > |t|
Family		1	1.2680	1.1201	16	1.13	0.2743
Family		2	0.08980	1.1121	16	0.08	0.9366
Family		3	−1.6660	1.1712	16	−1.42	0.1741
Family		4	0.3082	1.1201	16	0.28	0.7867
Family*Gender	F	1	−0.3198	1.0810	16	−0.30	0.7712
Family*Gender	M	1	1.2523	1.0933	16	1.15	0.2689
Family*Gender	F	2	−0.4299	1.0774	16	−0.40	0.6951
Family*Gender	M	2	0.4959	1.0774	16	0.46	0.6515
Family*Gender	F	3	−0.08229	1.1409	16	−0.07	0.9434
Family*Gender	M	3	−1.1429	1.1409	16	−1.00	0.3314
Family*Gender	F	4	0.8320	1.0933	16	0.76	0.4577
Family*Gender	M	4	−0.6053	1.0810	16	−0.56	0.5832

A number of points are noteworthy in comparing the results from the procedures. The covariance parameter estimates are the same, yet the solutions for the fixed effects differ. In fact, both solutions are correct. Solving a sparse system of linear equations requires reordering of the mixed model equations to minimize memory consumption in the factorization process. As a consequence, the order in which singularities are detected can differ from the order in which effects enter the model. Mathematically, the two sets of solutions simply correspond to different choices for the generalized inverse in solving a singular linear system. See the sections "Generalized Inverse Matrices" on page 52 and "Linear Model Theory" on page 61, in Chapter 3, "Introduction to Statistical Modeling with SAS/STAT Software," for more information about the role and importance of generalized inverses in linear model analysis.

Although the two sets of solutions for the fixed effects correspond to different choices of generalized inverses, many important results are invariant to the choice of the g-inverse. For example, the solutions for the random effects in Output 43.2.5 and Output 43.2.2 are identical. Also, the test for the Gender effect yields the same F value in both analyses (compare Output 43.2.6 and Output 43.2.3). However, note that the p-values associated with both F tests and t tests differ between the two procedures. This is due to their different default methods for computing the degrees of freedom. For this model, the HPMIXED procedure use the residual method to determine the denominator degrees of freedom for tests of fixed effects, whereas the MIXED procedure uses the containment method. The containment method is order-dependent, and thus not available in the HPMIXED procedure.

Output 43.2.6 Parameter Estimates and Solutions: HPMIXED Procedure

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Gender	1	16	7.95	0.0123

Example 43.3: Using PROC GLIMMIX for Further Analysis of PROC HPMIXED Fit

The HPMIXED procedure handles only a subset of the analyses of the GLIMMIX procedure. However, you can use the HPMIXED procedure to accelerate your GLIMMIX procedure analyses for large problems. The idea is to use PROC HPMIXED to maximize the likelihood and produce parameter estimates more quickly than PROC GLIMMIX, and then to pass these parameter estimates to PROC GLIMMIX for some further analysis that is not available within PROC HPMIXED.

This example revisits the mixed model problem from the section "Getting Started: HPMIXED Procedure" on page 2808 to illustrate how to obtain the covariance estimates from the HPMIXED procedure and, in turn, how to use these estimates in PROC GLIMMIX's PARMS statement. The following statements again simulate data from animals of different species on different farms:

```
data Sim;
   keep Species Farm Animal Yield;
   array AnimalEffect{3000};
   array AnimalSpecies{3000};
   array AnimalFarm{3000};
   do i = 1 to 3000;
      AnimalEffect{i} = sqrt(4.0)*rannor(12345);
      AnimalSpecies{i} = 1 + int(5*ranuni(12345));
      AnimalFarm{i}    = 1 + int(10*ranuni(12345));
      end;
   do i = 1 to 40000;
      Animal  = 1 + int(3000*ranuni(12345));
      Species = AnimalSpecies{Animal};
      Farm    = AnimalFarm{Animal};
      Yield   = 1 + Species + int(Farm/2) + AnimalEffect{Animal}
                  + sqrt(8.0)*rannor(12345);
      output;
      end;
run;
```

Note that in the preceding DATA step program, certain pairs of farms are simulated to have the same effect on yield. Suppose that your goal is to determine which farms are significantly different. While the HPMIXED procedure has an LSMEANS statement, it has no options for multiple comparisons. The following statements first use the HPMIXED procedure to obtain the covariance estimates, saving them in the SAS data set HPMEstimate. Then the GLIMMIX procedure is executed with the

PARMS statement to initialize the parameter values from the data set HPMEstimate and with the HOLD= and NOITER options to prevent further optimization iterations. The LSMEANS statement is used in PROC GLIMMIX to perform multiple comparisons of the LS-means for farms, and the results are displayed as a so-called diffogram.

```
proc hpmixed data=Sim;
   class Species Farm Animal;
   model Yield = Farm|Species;
   random Animal;
   test Species Species*Farm;
   ods output CovParms=HPMEstimate;
run;

proc glimmix data=Sim;
   class Species Farm Animal;
   model Yield = Farm|Species;
   random int/sub=Animal;
   parms /pdata=HPMEstimate hold=1,2 noiter;
   lsmeans Farm / pdiff=all plot=diffplot;
run;
```

The iteration histories for the two procedures are shown in Output 43.3.1 and Output 43.3.2. Whereas PROC HPMIXED requires several iterations in order to converge, PROC GLIMMIX "converges" to the same value in one step, with no iteration since the options HOLD= and NOITER are used.

Output 43.3.1 Iteration History for the HPMIXED Procedure

The HPMIXED Procedure

Iteration History

Iteration	Evaluations	Objective Function	Change	Max Gradient
0	4	202516.66891	.	0.841954
1	6	202516.66887	0.00004385	0.000641
2	1	202516.66887	-0.00000000	0.000641

Output 43.3.2 Iteration History for the GLIMMIX Procedure

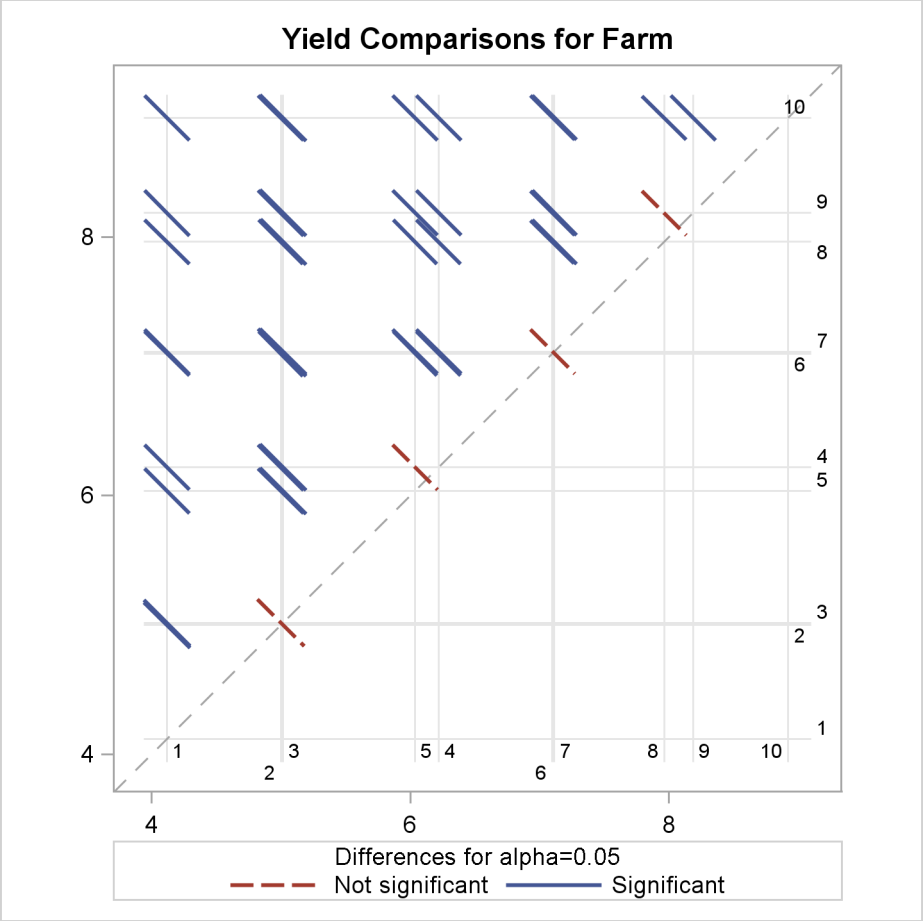
The GLIMMIX Procedure

Iteration History

Iteration	Restarts	Evaluations	Objective Function	Change	Max Gradient
0	0	4	202516.66887	.	0

The graphical multiple-comparisons analysis for the LS-means of farms is shown in Output 43.3.3. It confirms the pairwise equalities between farm effects with which the data were simulated.

Output 43.3.3 LS-Means Plot of Pairwise Farm Differences



For more information about the interpretation of the LS-means difference plot, see the section "ODS Graphics" on page 2282, in Chapter 38, "The GLIMMIX Procedure."

Example 43.4: Mixed Model Analysis of Microarray Data

Microarray experiments are an advanced genomic technique used in the discovery of new treatments for diseases. Microarray analysis allows for the detection of tens of thousands of genes in a single DNA sample. A microarray is a glass slide or membrane that has been spotted or "arrayed" with DNA fragments or oligonucleotides representing specific genes. The response of the gene detected by a spot is proportional to the intensity of fluorescence associated with that spot. These gene responses can indicate associations with disease conditions, but they can also be affected by systematic biases and different treatments such as sex and genotypes. Statistical models for microarray data attempt to assess the significance and magnitude of gene effects across treatments

while adjusting for these systematic biases and to evaluate the significance of differences between treatments.

There are two statistical approaches frequently used in mixed model analysis for microarray data. The first approach is to fit multiple gene-specific models to data normalized for systematic biases (Wolfinger et al. 2001; Gibson and Wolfinger 2004). This approach is based on assuming that the biases are independent from the gene effects. If this assumption is untenable, then a second approach fits a single model that combines both the systematic biases and the gene effects (Kerr, Martin, and Churchill 2000; Churchill 2002; Littell et al. 2006). When the number of genes is very large, several hundreds to tens of thousands, this is an analysis for which the sparse matrix approach implemented in the HPMIXED procedure is well suited.

The following SAS statements simulate a microarray experiment with a so-called loop design structure, which is commonly used in such studies. There are 500 genes, each gene occurs in 6 arrays, and each array has 2 dyes.

```
%let narray  = 6;
%let ndye    = 2;
%let nrow    = 4;
%let ngene   = 500;
%let ntrt    = 6;
%let npin    = 4;
%let ndip    = 4;
%let no      = %eval(&ndye*&nrow*&ngene);
%let tno     = %eval(&narray*&no);

data microarray;
   keep Gene MArray Dye Trt Pin Dip log2i;
   array PinDist{&tno};
   array DipDist{&tno};
   array GeneDist{&tno};

   array ArrayEffect{&narray};
   array ArrayGeneEffect{%eval(&narray*&ngene)};
   array ArrayDipEffect{%eval(&narray*&ndip)};
   array ArrayPinEffect{%eval(&narray*&npin)};

   do i = 1 to &tno;
      PinDist{i}  = 1 + int(&npin*ranuni(12345));
      DipDist{i}  = 1 + int(&ndip*ranuni(12345));
      GeneDist{i} = 1 + int(&ngene*ranuni(12345));
      end;

   igene = 0;
   idip = 0;
   ipin = 0;
   do i = 1 to &narray;
      ArrayEffect{i} = sqrt(0.014)*rannor(12345);
      do j = 1 to &ngene;
         igene = igene+1;
         ArrayGeneEffect{igene} = sqrt(0.0017)*rannor(12345);
         end;
      do j = 1 to &ndip;
```

```
      idip = idip + 1;
      ArrayDipEffect{idip} = sqrt(0.0033)*rannor(12345);
      end;
   do j = 1 to &npin;
      ipin = ipin + 1;
      ArrayPinEffect{ipin} = sqrt(0.037)*rannor(12345);
      end;
   end;

i = 0;
do MArray = 1 to &narray;
   do Dye = 1 to &ndye;
      do Row = 1 to &nrow;
         do k = 1 to &ngene;
            if MArray=1 and Dye = 1 then do;
               Trt = 0;
               trtc = 0;
               end;
            else do;
               if trtc >= &no then trtc = 0;
               if trtc = 0 then do;
                  Trt = Trt + 1;
                  if Trt >= &ntrt then do;
                     Trt = 0;
                     trtc = 0;
                     end;
                  end;
               trtc = trtc + 1;
               end;
            i = i + 1;
            Pin = PinDist{i};
            Dip = DipDist{i};
            Gene = GeneDist{i};
            a   = ArrayEffect{MArray};
            ag  = ArrayGeneEffect{(MArray-1)*&ngene+Gene};
            ad  = ArrayDipEffect{(MArray-1)*&ndip+Dip};
            ap  = ArrayPinEffect{(MArray-1)*&npin+Pin};
            log2i   = 1 +
                      + Dye
                      + Trt
                      + Gene/1000.0
                      + Dye*Gene/1000.0
                      + Trt*Gene/1000.0
                      + Pin
                      + a
                      + ag
                      + ad
                      + ap
                      + sqrt(0.02)*rannor(12345);
            output;
            end;
         end;
      end;
   end;
```

```
run;
```

A linear mixed model for fitting the log intensity data Y_{ijkmnr} from such a design is described by Littell et al. (2006) as follows:

$$
\begin{array}{rcll}
Y_{ijkmnr} & = & & \textbf{Fixed Effects} \\
 & & \mu & \text{Overall mean} \\
 & + & \lambda_i & \text{Gene} \\
 & + & \tau_j & \text{Treatment} \\
 & + & \delta_k & \text{Dye} \\
 & + & (\tau\lambda)_{ij} & \text{Treatment-by-gene} \\
 & + & (\delta\lambda)_{ik} & \text{Dye-by-gene} \\
 & + & p_r & \text{Pin} \\
 & & & \\
 & & & \textbf{Random Effects} \\
 & + & a_m & \text{Microarray} \\
 & + & (a\lambda)_{im} & \text{Microarray-by-gene} \\
 & + & d(a)_{mn} & \text{Dip-within-mircroarray} \\
 & + & (ap)_{mr} & \text{Microarray-by-pin} \\
 & + & e_{ijkmnr} & \text{Residual noise}
\end{array}
$$

You can use the HPMIXED procedure with the following statements to fit this model:

```
proc hpmixed data=microarray;
   class marray dye trt gene pin dip;
   model log2i = dye trt gene dye*gene trt*gene pin;
   random marray marray*gene dip(marray) pin*marray;
   test trt;
run;
```

The "Dimensions" table shown in Output 43.4.1 indicates that this is a very large model, with 4512 columns in **X** matrix and 3054 columns in **Z** matrix. It will be computationally very inefficient to fit this model by using dense matrix methods; the sparse matrix approach of the HPMIXED procedure is of critical importance.

Output 43.4.1 Mixed Model Dimensions

```
                    The HPMIXED Procedure

                          Dimensions

              G-side Cov. Parameters            4
              R-side Cov. Parameters            1
              Columns in X                   4513
              Columns in Z                   3054
              Subjects (Blocks in V)            1
```

The *p*-value in Output 43.4.2 indicates that there are significant differences between treatments.

Output 43.4.2 Type III Tests of Fixed Effects

Type III Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Trt	5	20497	370005	<.0001

References

Akaike, H. (1974), "A New Look at Statistical Model Identification," *IEEE Transactions on Automatic Control*, 19, 716–723.

Bozdogan, H. (1987), "Model Selection and Akaike's Information Criterion (AIC): The General Theory and Its Analytical Extensions," *Psychometrika*, 52, 345–370.

Brown, H. and Prescott, R. (1999), *Applied Mixed Models in Medicine*, New York: John Wiley & Sons.

Burnham, K. P. and Anderson, D. R. (1998), *Model Selection and Inference: A Practical Information-Theoretic Approach*, New York: Springer-Verlag.

Churchill, G. A. (2002), "Fundamentals of Experimental Design for cDNA Microarray," *Nature Genetics*, 32, 490–495.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society, Series B*, 39, 1–38.

George, J. A. and Liu, J. W. (1981), *Computer Solutions of Large Sparse Positive Definite Systems*, Englewood Cliffs, NJ: Prentice-Hall.

Gibson, G. and Wolfinger, R. D. (2004), "Gene Expression Profiling Using Mixed Models," in A. M. Saxton, ed., *Genetic Analysis of Complex Traits Using SAS*, 251–278, Cary, NC: SAS Publishing.

Gilmour, A. R., Thompson, R., and Cullis, B. R. (1995), "Average Information REML: An Efficient Algorithm for Variance Parameter Estimation in Linear Mixed Models," *Biometrics*, 51, 1440–1450.

Hannan, E. J. and Quinn, B. G. (1979), "The Determination of the Order of an Autoregression," *Journal of the Royal Statistical Society, Series B*, 41, 190–195.

Henderson, C. R. (1990), "Statistical Method in Animal Improvement: Historical Overview," in *Advances in Statistical Methods for Genetic Improvement of Livestock*, 1–14, New York: Springer-Verlag.

Hurvich, C. M. and Tsai, C.-L. (1989), "Regression and Time Series Model Selection in Small Samples," *Biometrika*, 76, 297–307.

Johnson, D. L. and Thompson, R. (1995), "Restricted Maximum Likelihood Estimation of Variance Components for Univariate Animal Models Using Sparse Matrix Techniques and Average Information," *Journal of Dairy Science*, 78, 449–456.

Kerr, M. K., Martin, M., and Churchill, G. A. (2000), "Analysis of Variance for Gene Expression Microarray Data," *Journal of Computational Biology*, 7, 819–837.

Littell, R. C., Milliken, G. A., Stroup, W. W., and Wolfinger, R. D. (1996), *SAS System for Mixed Models*, Cary, NC: SAS Institute Inc.

Littell, R. C., Milliken, G. A., Stroup, W. W., Wolfinger, R. D., and Schabenberger, O. (2006), *SAS for Mixed Models*, Second Edition, Cary, NC: SAS Press.

McLean, R. A., Sanders, W. L., and Stroup, W. W. (1991), "A Unified Approach to Mixed Linear Models," *The American Statistician*, 45, 54–64.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Searle, S. R., Casella, G., and McCulloch, C. E. (1992), *Variance Components*, New York: John Wiley & Sons.

Shewchuk, J. R. (1994), *An Introduction to the Conjugate Gradient Method without the Agonizing Pain*, Technical report, Carnegie Mellon University, Pittsburgh, PA.

Tsuruta, S., Misztal, I., and Stranden, I. (2001), "Use of the Preconditioned Conjugate Gradient Algorithm as a Generic Solver for Mixed-Model Equations in Animal Breeding Apllications," *Journal of Animal Science*, 79, 1166–1172.

Verbeke, G. and Molenberghs, G., eds. (1997), *Linear Mixed Models in Practice: A SAS-Oriented Approach*, New York: Springer.

Verbeke, G. and Molenberghs, G. (2000), *Linear Mixed Models for Longitudinal Data*, New York: Springer.

Wolfinger, R. D., Gibson, G., Wolfinger, E., Bennett, L., Hamadeh, H., Bushel, P., Afshari, C., and Paules, R. S. (2001), "Assessing Gene Significance from cDNA Microarray Expression Data via Mixed Models," *Journal of Computational Biology*, 8, 625–637.

Chapter 44

The INBREED Procedure

Contents

Overview: INBREED Procedure **2857**
Getting Started: INBREED Procedure **2858**
The Format of the Input Data Set 2858
Performing the Analysis 2860
Syntax: INBREED Procedure **2863**
PROC INBREED Statement 2864
BY Statement 2865
CLASS Statement 2866
GENDER Statement 2866
MATINGS Statement 2867
VAR Statement 2867
Details: INBREED Procedure **2867**
Missing Values 2867
DATA= Data Set 2868
Computational Details 2869
OUTCOV= Data Set 2875
Displayed Output 2876
ODS Table Names 2877
Examples: INBREED Procedure **2877**
Example 44.1: Monoecious Population Analysis 2877
Example 44.2: Pedigree Analysis 2880
Example 44.3: Pedigree Analysis with BY Groups 2882
References **2883**

Overview: INBREED Procedure

The INBREED procedure calculates the covariance or inbreeding coefficients for a pedigree. PROC INBREED is unique in that it handles very large populations.

The INBREED procedure has two modes of operation. One mode carries out analysis on the assumption that all the individuals belong to the same generation. The other mode divides the population into nonoverlapping generations and analyzes each generation separately, assuming that the parents of individuals in the current generation are defined in the previous generation.

PROC INBREED also computes averages of the covariance or inbreeding coefficients within sex categories if the sex of individuals is known.

Getting Started: INBREED Procedure

This section demonstrates how you can use the INBREED procedure to calculate the inbreeding or covariance coefficients for a pedigree, how you can control the analysis mode if the population consists of nonoverlapping generations, and how you can obtain averages within sex categories.

For you to use PROC INBREED effectively, your input data set must have a definite format. The following sections first introduce this format for a fictitious population and then demonstrate how you can analyze this population by using the INBREED procedure.

The Format of the Input Data Set

The SAS data set used as input to the INBREED procedure must contain an observation for each individual. Each observation must include one variable identifying the individual and two variables identifying the individual's parents. Optionally, an observation can contain a known covariance coefficient and a character variable defining the gender of the individual.

For example, consider the following data:

```
data Population;
   input Individual $ Parent1 $ Parent2 $
         Covariance Sex $ Generation;
   datalines;
Mark    George Lisa     .     M  1
Kelly   Scott  Lisa     .     F  1
Mike    George Amy      .     M  1
.       Mark   Kelly   0.50   .  1
David   Mark   Kelly    .     M  2
Merle   Mike   Jane     .     F  2
Jim     Mark   Kelly   0.50   M  2
Mark    Mike   Kelly    .     M  2
;
```

It is important to order the pedigree observations so that individuals are defined before they are used as parents of other individuals. The family relationships between individuals cannot be ascertained correctly unless you observe this ordering. Also, older individuals must precede younger ones. For example, 'Mark' appears as the first parent of 'David' at observation 5; therefore, his observation needs to be defined prior to observation 5. Indeed, this is the case (see observation 1). Also, 'David' is older than 'Jim', whose observation appears after the observation for 'David', as is appropriate.

In populations with distinct, nonoverlapping generations, the older generation (parents) must precede the younger generation. For example, the individuals defined in Generation=1 appear as parents of individuals defined in Generation=2.

PROC INBREED produces warning messages when a parent cannot be found. For example, 'Jane' appears as the second parent of the individual 'Merle' even though there are no previous observations defining her own parents. If the population is treated as an overlapping population, that is, if the generation grouping is ignored, then the procedure inserts an observation for 'Jane' with missing parents just before the sixth observation, which defines 'Merle' as follows:

```
Jane    .       .       .     F  2
Merle   Mike    Jane    .     F  2
```

However, if generation grouping is taken into consideration, then 'Jane' is defined as the last observation in Generation=1, as follows:

```
Mike    George  Amy     .     M  1
Jane    .       .       .     F  1
```

In this latter case, however, the observation for 'Jane' is inserted after the computations are reported for the first generation. Therefore, she does not appear in the covariance/inbreeding matrix, even though her observation is used in computations for the second generation (see Figure 44.2).

If the data for an individual are duplicated, only the first occurrence of the data is used by the procedure, and a warning message is displayed to note the duplication. For example, individual 'Mark' is defined twice, at observations 1 and 8. If generation grouping is ignored, then this is an error and observation 8 is skipped. However, if the population is processed with respect to two distinct generations, then 'Mark' refers to two different individuals, one in Generation=1 and the other in Generation=2.

If a covariance is to be assigned between two individuals, then those individuals must be defined prior to the assignment observation. For example, a covariance of 0.50 can be assigned between 'Mark' and 'Kelly' since they are previously defined. Note that assignment statements must have different formats depending on whether the population is processed with respect to generations (see the section "DATA= Data Set" on page 2868 for further information). For example, while observation 4 is valid for nonoverlapping generations, it is invalid for a processing mode that ignores generation grouping. In this latter case, observation 7 indicates a valid assignment, and observation 4 is skipped.

The latest covariance specification between any given two individuals overrides the previous one between the same individuals.

Performing the Analysis

To compute the covariance coefficients for the overlapping generation mode, use the following statements:

```
proc inbreed data=Population covar matrix init=0.25;
run;
```

Here, the DATA= option names the SAS data set to be analyzed, and the COVAR and MATRIX options tell the procedure to output the covariance coefficients matrix. If you omit the COVAR option, the inbreeding coefficients are output instead of the covariance coefficients.

Note that the PROC INBREED statement also contains the INIT= option. This option gives an initial covariance between any individual and unknown individuals. For example, the covariance between any individual and 'Jane' would be 0.25, since 'Jane' is unknown, except when 'Jane' appears as a parent (see Figure 44.4).

Figure 44.1 Analysis for an Overlapping Population

The INBREED Procedure

Covariance Coefficients

Individual	Parent1	Parent2	George	Lisa	Mark	Scott	Kelly
George			1.1250	0.2500	0.6875	0.2500	0.2500
Lisa			0.2500	1.1250	0.6875	0.2500	0.6875
Mark	George	Lisa	0.6875	0.6875	1.1250	0.2500	0.5000
Scott			0.2500	0.2500	0.2500	1.1250	0.6875
Kelly	Scott	Lisa	0.2500	0.6875	0.5000	0.6875	1.1250
Amy			0.2500	0.2500	0.2500	0.2500	0.2500
Mike	George	Amy	0.6875	0.2500	0.4688	0.2500	0.2500
David	Mark	Kelly	0.4688	0.6875	0.8125	0.4688	0.8125
Jane			0.2500	0.2500	0.2500	0.2500	0.2500
Merle	Mike	Jane	0.4688	0.2500	0.3594	0.2500	0.2500
Jim	Mark	Kelly	0.4688	0.6875	0.8125	0.4688	0.8125

Covariance Coefficients

Individual	Parent1	Parent2	Amy	Mike	David	Jane	Merle
George			0.2500	0.6875	0.4688	0.2500	0.4688
Lisa			0.2500	0.2500	0.6875	0.2500	0.2500
Mark	George	Lisa	0.2500	0.4688	0.8125	0.2500	0.3594
Scott			0.2500	0.2500	0.4688	0.2500	0.2500
Kelly	Scott	Lisa	0.2500	0.2500	0.8125	0.2500	0.2500
Amy			1.1250	0.6875	0.2500	0.2500	0.4688
Mike	George	Amy	0.6875	1.1250	0.3594	0.2500	0.6875
David	Mark	Kelly	0.2500	0.3594	1.2500	0.2500	0.3047
Jane			0.2500	0.2500	0.2500	1.1250	0.6875
Merle	Mike	Jane	0.4688	0.6875	0.3047	0.6875	1.1250
Jim	Mark	Kelly	0.2500	0.3594	0.8125	0.2500	0.3047

Covariance Coefficients

Individual	Parent1	Parent2	Jim
George			0.4688
Lisa			0.6875
Mark	George	Lisa	0.8125
Scott			0.4688
Kelly	Scott	Lisa	0.8125
Amy			0.2500
Mike	George	Amy	0.3594
David	Mark	Kelly	0.8125
Jane			0.2500
Merle	Mike	Jane	0.3047
Jim	Mark	Kelly	1.2500

Number of Individuals 11

In the previous example, PROC INBREED treats the population as a single generation. However, you might want to process the population with respect to distinct, nonoverlapping generations. To

accomplish this, you need to identify the generation variable in a CLASS statement, as shown by the following statements:

```
proc inbreed data=Population covar matrix init=0.25;
   class Generation;
run;
```

Note that, in this case, the covariance matrix is displayed separately for each generation (see Figure 44.5).

Figure 44.2 Analysis for a Nonoverlapping Population

The INBREED Procedure

Generation = 1

Covariance Coefficients

Individual	Parent1	Parent2	Mark	Kelly	Mike
Mark	George	Lisa	1.1250	0.5000	0.4688
Kelly	Scott	Lisa	0.5000	1.1250	0.2500
Mike	George	Amy	0.4688	0.2500	1.1250

Number of Individuals 3

The INBREED Procedure

Generation = 2

Covariance Coefficients

Individual	Parent1	Parent2	David	Merle	Jim	Mark
David	Mark	Kelly	1.2500	0.3047	0.8125	0.5859
Merle	Mike	Jane	0.3047	1.1250	0.3047	0.4688
Jim	Mark	Kelly	0.8125	0.3047	1.2500	0.5859
Mark	Mike	Kelly	0.5859	0.4688	0.5859	1.1250

Number of Individuals 4

You might also want to see covariance coefficient averages within sex categories. This is accomplished by indicating the variable defining the gender of individuals in a GENDER statement and by adding the AVERAGE option to the PROC INBREED statement. For example, the following statements produce the covariance coefficient averages shown in Figure 44.3:

```
proc inbreed data=Population covar average init=0.25;
   class Generation;
   gender Sex;
run;
```

Figure 44.3 Averages within Sex Categories for a Nonoverlapping Generation

```
                    The INBREED Procedure

                        Generation = 1

      Averages of Covariance Coefficient Matrix in Generation 1

                          On Diagonal      Below Diagonal

        Male X Male              1.1250              0.4688
        Male X Female             .                  0.3750
        Female X Female          1.1250              0.0000
        Over Sex                 1.1250              0.4063

                     Number of Males          2
                     Number of Females        1
                     Number of Individuals    3

                    The INBREED Procedure

                        Generation = 2

      Averages of Covariance Coefficient Matrix in Generation 2

                          On Diagonal      Below Diagonal

        Male X Male              1.2083              0.6615
        Male X Female             .                  0.3594
        Female X Female          1.1250              0.0000
        Over Sex                 1.1875              0.5104

                     Number of Males          3
                     Number of Females        1
                     Number of Individuals    4
```

Syntax: INBREED Procedure

The following statements are available in PROC INBREED:

PROC INBREED < *options* > ;
BY *variables* ;
CLASS *variable* ;
GENDER *variable* ;
MATINGS *individual-list1* / *mate-list1* <,. . . , *individual-listn* / *mate-listn* > ;
VAR *variables* ;

The PROC INBREED statement is required. Items within angle brackets (< >) are optional. The syntax of each statement is described in the following sections.

PROC INBREED Statement

PROC INBREED < *options* > **;**

The options listed in Table 44.1 are available in the PROC INBREED statement.

Table 44.1 INBREED Procedure Options

Task	Option
Specify Data Sets	DATA= OUTCOV=
Control Type of Coefficient	COVAR
Control Displayed Tables	AVERAGE IND MATRIX
Specify Default Covariance Value	INIT=
Suppress Output	INDL MATRIXL NOPRINT

AVERAGE

A

produces a table of averages of coefficients for each pedigree of offspring. The AVERAGE option is used together with the GENDER statement to average the inbreeding/covariance coefficients within sex categories.

COVAR

C

specifies that all coefficients output consist of covariance coefficients rather than inbreeding coefficients.

DATA=*SAS-data-set*

names the SAS data set to be used by PROC INBREED. If you omit the DATA= option, the most recently created SAS data set is used.

IND

I

displays the individuals' inbreeding coefficients (diagonal of the inbreeding coefficients matrix) for each pedigree of offspring.

If you also specify the COVAR option, the individuals' covariance coefficients (diagonal of the covariance coefficients matrix) are displayed.

INDL

displays individuals' coefficients for only the last generation of a multiparous population.

INIT=*cov*

specifies the covariance value *cov* if any of the parents are unknown; a value of 0 is assumed if you do not specify the INIT= option.

MATRIX

M

displays the inbreeding coefficient matrix for each pedigree of offspring.

If you also specify the COVAR option, the covariance matrices are displayed instead of inbreeding coefficients matrices.

MATRIXL

displays coefficients for only the last generation of a multiparous population.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information on ODS, see Chapter 20, "Using the Output Delivery System."

OUTCOV=*SAS-data-set*

names an output data set to contain the inbreeding coefficients. When the COVAR option is also specified, covariance estimates are output to the OUTCOV= data set instead of inbreeding coefficients.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC INBREED to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC INBREED. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide.*

CLASS Statement

CLASS *variable* **;**

To analyze the population within nonoverlapping generations, you must specify the variable that identifies generations in a CLASS statement. Values of the generation variable, called *generation numbers*, must be integers, but generations are assumed to occur in the order of their input in the input data set rather than in numerical order of the generation numbers. The name of an individual needs to be unique only within its generation.

When the MATRIXL option or the INDL option is specified, each generation requires a unique generation number in order for the specified option to work correctly. If generation numbers are not unique, all the generations with a generation number that is the same as the last generation's are output.

GENDER Statement

GENDER *variable* **;**

The GENDER statement specifies a variable that indicates the sex of the individuals. Values of the sex variable must be character beginning with 'M' or 'F', for male or female. The GENDER statement is needed only when you specify the AVERAGE option to average the inbreeding/covariance coefficients within sex categories or when you want to include a gender variable in the OUTCOV= data set.

PROC INBREED makes the following assumptions regarding the gender of individuals:

- The first parent is always assumed to be the male. See the section "VAR Statement" on page 2867.
- The second parent is always assumed to be the female. See the section "VAR Statement" on page 2867.
- If the gender of an individual is missing or invalid, this individual is assumed to be a female unless the population is overlapping and this individual appears as the first parent in a later observation.

Any contradictions to these rules are reported in the SAS log.

MATINGS Statement

MATINGS *individual-list1 / mate-list1 <,. . . , individual-listn / mate-listn>* **;**

You can specify the MATINGS statement with PROC INBREED to specify selected matings of individuals. Each individual given in *individual-list* is mated with each individual given in *mate-list*. You can write multiple mating specifications if you separate them by commas or asterisks. The procedure reports the inbreeding coefficients or covariances for each pair of mates. For example, you can use the following statement to specify the mating of an individual named 'David' with an individual named 'Jane':

```
matings david / jane;
```

VAR Statement

VAR *individual parent1 parent2 <covariance>* **;**

The VAR statement specifies three or four variables: the first variable contains an individual's name, the second variable contains the name of the individual's first parent, and the third variable contains the name of the individual's second parent. An optional fourth variable assigns a known value to the covariance of the individual's first and second parents in the current generation.

The first three variables in the VAR statement can be either numeric or character; however, only the first 12 characters of a character variable are recognized by the procedure. The fourth variable, if specified, must be numeric.

If you omit the VAR statement, then the procedure uses the first three unaddressed variables as the names of the individual and its parents. (Unaddressed variables are those that are not referenced in any other PROC INBREED statement.) If the input data set contains an unaddressed fourth variable, then it becomes the covariance variable.

Details: INBREED Procedure

Missing Values

A missing value for a parent implies that the parent is unknown. Unknown parents are assumed to be unrelated and not inbred unless you specify the INIT= option.

When the value of the variable identifying the individual is missing, the observation is not added to the list of individuals. However, for a multiparous population, an observation with a missing individual is valid and is used for assigning covariances.

Missing covariance values are determined from the INIT=*cov* option, if specified. Observations with missing generation variables are excluded.

If the gender of an individual is missing, it is determined from the order in which it is listed on the first observation defining its progeny for an overlapping population. If it appears as the first parent, it is set to 'M'; otherwise, it is set to 'F'. When the gender of an individual cannot be determined, it is assigned a default value of 'F'.

DATA= Data Set

Each observation in the input data set should contain necessary information such as the identification of an individual and the first and second parents of an individual. In addition, if a CLASS statement is specified, each observation should contain the generation identification; and, if a GENDER statement is specified, each observation should contain the gender of an individual. Optionally, each observation might also contain the covariance between the first and the second parents. Depending on how many statements are specified with the procedure, there should be enough variables in the input data set containing this information.

If you omit the VAR statement, then the procedure uses the first three *unaddressed variables* in the input data set as the names of the individual and his or her parents. Unaddressed variables in the input data set are those variables that are not referenced by the procedure in any other statements, such as CLASS, GENDER, or BY statements. If the input data set contains an unaddressed fourth variable, then the procedure uses it as the covariance variable.

If the individuals given by the variables associated with the first and second parents are not in the population, they are added to the population. However, if they are in the population, they must be defined prior to the observation that gives their progeny.

When there is a CLASS statement, the functions of defining new individuals and assigning covariances must be separated. This is necessary because the parents of any given individual are defined in the previous generation, while covariances are assigned between individuals in the current generation.

Therefore, there could be two types of observations for a multiparous population:

- one to define new individuals in the current generation whose parents have been defined in the previous generation, as in the following, where the missing value is for the covariance variable:

```
Mark   George Lisa    .    M  1
Kelly  Scott  Lisa    .    F  1
```

- one to assign covariances between two individuals in the current generation, as in the following, where the individual's name is missing, 'Mark' and 'Kelly' are in the current generation, and the covariance coefficient between these two individuals is 0.50:

```
.      Mark   Kelly  0.50  .  1
```

Note that the observations defining individuals must precede the observation assigning a covariance value between them. For example, if a covariance is to be assigned between 'Mark' and 'Kelly', then both of them should be defined prior to the assignment observation.

Computational Details

This section describes the rules that the INBREED procedure uses to compute the covariance and inbreeding coefficients. Each computational rule is explained by an example referring to the fictitious population introduced in the section "Getting Started: INBREED Procedure" on page 2858.

Coancestry (or Kinship Coefficient)

To calculate the inbreeding coefficient and the covariance coefficients, use the degree of relationship by descent between the two parents, which is called *coancestry* or *kinship coefficient* (Falconer and Mackay 1996, p.85), or *coefficient of parentage* (Kempthorne 1957, p.73). Denote the coancestry between individuals X and Y by f_{XY}. For information on how to calculate the coancestries among a population, see the section "Calculation of Coancestry" on page 2870.

Covariance Coefficient (or Coefficient of Relationship)

The covariance coefficient between individuals X and Y is defined by

$$\mathrm{Cov}(\mathrm{X},\mathrm{Y}) = 2f_{\mathrm{XY}}$$

where f_{XY} is the coancestry between X and Y. The covariance coefficient is sometimes called the *coefficient of relationship* or the *theoretical correlation* (Falconer and Mackay (1996, p.153); Crow and Kimura (1970, p.134)). If a covariance coefficient cannot be calculated from the individuals in the population, it is assigned to an initial value. The initial value is set to 0 if the INIT= option is not specified or to *cov* if INIT=*cov*. Therefore, the corresponding initial coancestry is set to 0 if the INIT= option is not specified or to $\frac{1}{2}cov$ if INIT=*cov*.

Inbreeding Coefficients

The inbreeding coefficient of an individual is the probability that the pair of alleles carried by the gametes that produced it are identical by descent (Falconer and Mackay (1996, Chapter 5), Kempthorne (1957, Chapter 5)). For individual X, denote its inbreeding coefficient by F_{X}. The inbreeding coefficient of an individual is equal to the coancestry between its parents. For example, if X has parents A and B, then the inbreeding coefficient of X is

$$F_{\mathrm{X}} = f_{\mathrm{AB}}$$

Calculation of Coancestry

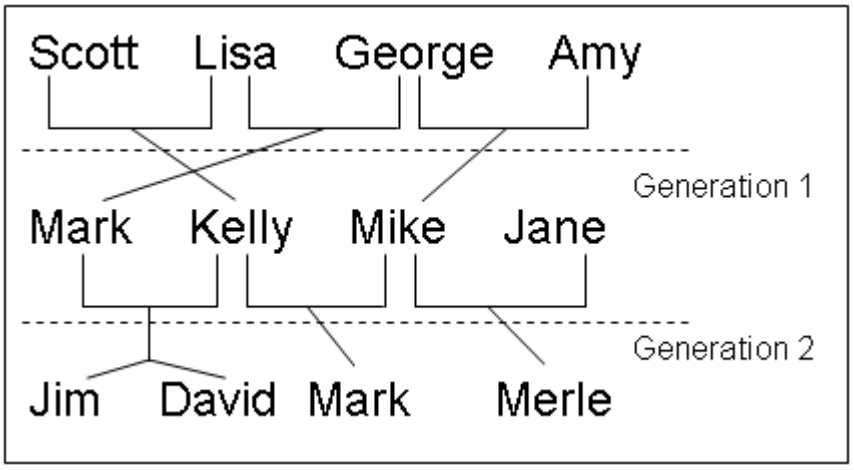
Given individuals X and Y, assume that X has parents A and B and that Y has parents C and D. For nonoverlapping generations, the basic rule to calculate the coancestry between X and Y is given by the following formula (Falconer and Mackay 1996, p.86):

$$f_{XY} = \frac{1}{4}\left(f_{AC} + f_{AD} + f_{BC} + f_{BD}\right)$$

And the inbreeding coefficient for an offspring of X and Y, called Z, is the coancestry between X and Y:

$$F_Z = f_{XY}$$

Figure 44.4 Inbreeding Relationship for Nonoverlapping Population



For example, in Figure 44.4, 'Jim' and 'Mark' from Generation 2 are progenies of 'Mark' and 'Kelly' and of 'Mike' and 'Kelly' from Generation 1, respectively. The coancestry between 'Jim' and 'Mark' is

$$f_{\text{Jim,Mark}} = \frac{1}{\left(f_{\text{Mark,Mike}} + f_{\text{Mark,Kelly}} + f_{\text{Kelly,Mike}} + f_{\text{Kelly,Kelly}}\right)}$$

From the covariance matrix for Generation=1 in Figure 44.4 and the relationship that coancestry is half of the covariance coefficient,

$$f_{\text{Jim,Mark}} = \frac{1}{4}\left(\frac{0.4688}{2} + \frac{0.5}{2} + \frac{0.25}{2} + \frac{1.125}{2}\right) = 0.29298$$

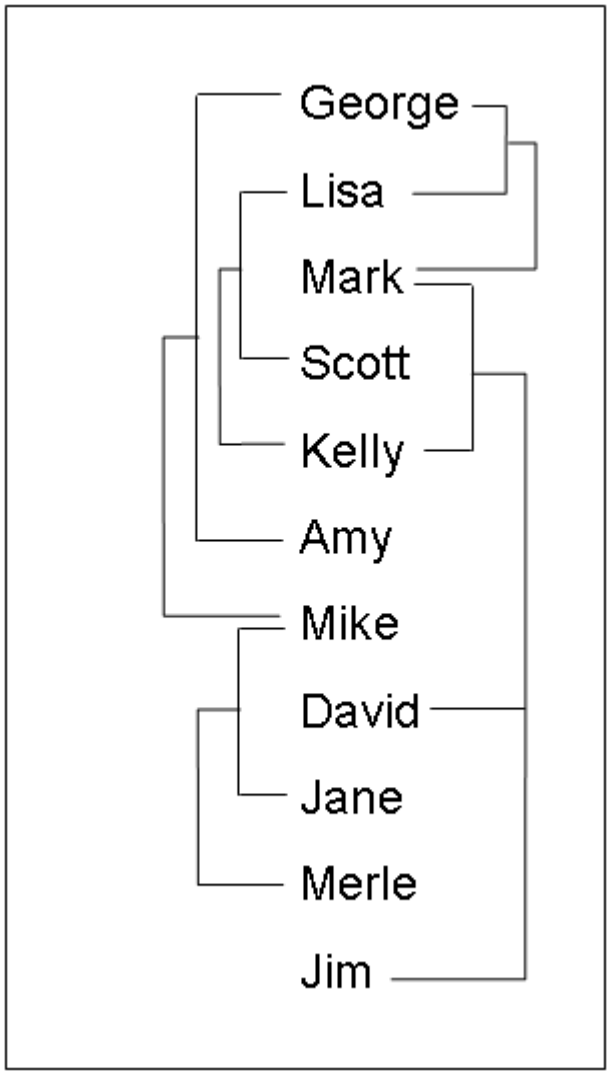
For overlapping generations, if X is older than Y, then the basic rule can be simplified to

$$F_Z = f_{XY} = \frac{1}{2}\left(f_{XC} + f_{XD}\right)$$

That is, the coancestry between X and Y is the average of coancestries between older X with younger Y's parents. For example, in Figure 44.5, the coancestry between 'Kelly' and 'David' is

$$f_{\text{Kelly,David}} = \frac{1}{2}\left(f_{\text{Kelly,Mark}} + f_{\text{Kelly,Kelly}}\right)$$

Figure 44.5 Inbreeding Relationship for Overlapping Population



This is so because 'Kelly' is defined before 'David'; therefore, 'Kelly' is not younger than 'David', and the parents of 'David' are 'Mark' and 'Kelly'. The covariance coefficient values Cov(Kelly,Mark) and Cov(Kelly,Kelly) from the matrix in Figure 44.5 yield that the coancestry between 'Kelly' and 'David' is

$$f_{\text{Kelly,David}} = \frac{1}{2}\left(\frac{0.5}{2} + \frac{1.125}{2}\right) = 0.40625$$

The numerical values for some initial coancestries must be known in order to use these rule. Either the parents of the first generation have to be unrelated, with $f = 0$ if the INIT= option is not specified in the PROC statement, or their coancestries must have an initial value of $\frac{1}{2}cov$, where cov is set by the INIT= option. Then the subsequent coancestries among their progenies and the inbreeding coefficients of their progenies in the rest of the generations are calculated by using these initial values.

Special rules need to be considered in the calculations of coancestries for the following cases.

Self-Mating

The coancestry for an individual X with itself, f_{XX}, is the inbreeding coefficient of a progeny that is produced by self-mating. The relationship between the inbreeding coefficient and the coancestry for self-mating is

$$f_{\text{XX}} = \frac{1}{2}(1 + F_{\text{X}})$$

The inbreeding coefficient F_{X} can be replaced by the coancestry between X's parents A and B, f_{AB}, if A and B are in the population:

$$f_{\text{XX}} = \frac{1}{2}(1 + f_{\text{AB}})$$

If X's parents are not in the population, then F_{X} is replaced by the initial value $\frac{1}{2}cov$ if cov is set by the INIT= option, or F_{X} is replaced by 0 if the INIT= option is not specified. For example, the coancestry of 'Jim' with himself is

$$f_{\text{Jim,Jim}} = \frac{1}{2}\left(1 + f_{\text{Mark,Kelly}}\right)$$

where 'Mark' and 'Kelly' are the parents of 'Jim'. Since the covariance coefficient Cov(Mark,Kelly) is 0.5 in Figure 44.5 and also in the covariance matrix for GENDER=1 in Figure 44.4, the coancestry of 'Jim' with himself is

$$f_{\text{Jim,Jim}} = \frac{1}{2}\left(1 + \frac{0.5}{2}\right) = 0.625$$

When INIT=0.25, then the coancestry of 'Jane' with herself is

$$f_{\text{Jane,Jane}} = \frac{1}{2}\left(1 + \frac{0.25}{2}\right) = 0.5625$$

because 'Jane' is not an offspring in the population.

Offspring and Parent Mating

Assuming that X's parents are A and B, the coancestry between X and A is

$$f_{\text{XA}} = \frac{1}{2}\left(f_{\text{AB}} + f_{\text{AA}}\right)$$

The inbreeding coefficient for an offspring of X and A, denoted by Z, is

$$F_{\text{Z}} = f_{\text{XA}} = \frac{1}{2}\left(f_{\text{AB}} + f_{\text{AA}}\right)$$

For example, 'Mark' is an offspring of 'George' and 'Lisa', so the coancestry between 'Mark' and 'Lisa' is

$$f_{\text{Mark,Lisa}} = \frac{1}{2}\left(f_{\text{Lisa,George}} + f_{\text{Lisa,Lisa}}\right)$$

From the covariance coefficient matrix in Figure 44.5, $f_{\text{Lisa,George}} = 0.25/2 = 0.125$, $f_{\text{Lisa,Lisa}} = 1.125/2 = 0.5625$, so that

$$f_{\text{Mark,Lisa}} = \frac{1}{2}\left(0.125 + 0.5625\right) = 0.34375$$

Thus, the inbreeding coefficient for an offspring of 'Mark' and 'Lisa' is 0.34375.

Full Sibs Mating

This is a special case for the basic rule given at the beginning of the section "Calculation of Coancestry" on page 2870. If X and Y are full sibs with same parents A and B, then the coancestry between X and Y is

$$f_{XY} = \frac{1}{4}(2f_{AB} + f_{AA} + f_{BB})$$

and the inbreeding coefficient for an offspring of A and B, denoted by Z, is

$$F_Z = f_{XY} = \frac{1}{4}(2f_{AB} + f_{AA} + f_{BB})$$

For example, 'David' and 'Jim' are full sibs with parents 'Mark' and 'Kelly', so the coancestry between 'David' and 'Jim' is

$$f_{\text{David,Jim}} = \frac{1}{4}\left(2f_{\text{Mark,Kelly}} + f_{\text{Mark,Mark}} + f_{\text{Kelly,Kelly}}\right)$$

Since the coancestry is half of the covariance coefficient, from the covariance matrix in Figure 44.5,

$$f_{\text{David,Jim}} = \frac{1}{4}\left(2 \times \frac{0.5}{2} + \frac{1.125}{2} + \frac{1.125}{2}\right) = 0.40625$$

Unknown or Missing Parents

When individuals or their parents are unknown in the population, their coancestries are assigned by the value $\frac{1}{2}cov$ if cov is set by the INIT= option or by the value 0 if the INIT= option is not specified. That is, if either A or B is unknown, then

$$f_{AB} = \frac{1}{2}cov$$

For example, 'Jane' is not in the population, and since 'Jane' is assumed to be defined just before the observation at which 'Jane' appears as a parent (that is, between observations 4 and 5), then 'Jane' is not older than 'Scott'. The coancestry between 'Jane' and 'Scott' is then obtained by using the simplified basic rule (see the section "Calculation of Coancestry" on page 2870):

$$f_{\text{Scott,Jane}} = \frac{1}{2}\left(f_{\text{Scott,}\cdot} + f_{\text{Scott,}\cdot}\right)$$

Here, dots (·) indicate Jane's unknown parents. Therefore, $f_{\text{Scott,}\cdot}$ is replaced by $\frac{1}{2}cov$, where cov is set by the INIT= option. If INIT=0.25, then

$$f_{\text{Scott,Jane}} = \frac{1}{2}\left(\frac{0.25}{2} + \frac{0.25}{2}\right) = 0.125$$

For a more detailed discussion on the calculation of coancestries, inbreeding coefficients, and covariance coefficients, refer to Falconer and Mackay (1996), Kempthorne (1957), and Crow and Kimura (1970).

OUTCOV= Data Set

The OUTCOV= data set has the following variables:

- a list of BY variables, if there is a BY statement
- the generation variable, if there is a CLASS statement
- the gender variable, if there is a GENDER statement
- _Type_, a variable indicating the type of observation. The valid values of the _Type_ variable are 'COV' for covariance estimates and 'INBREED' for inbreeding coefficients.
- _Panel_, a variable indicating the panel number used when populations delimited by BY groups contain different numbers of individuals. If there are n individuals in the first BY group and if any subsequent BY group contains a larger population, then its covariance/inbreeding matrix is divided into panels, with each panel containing n columns of data. If you put these panels side by side in increasing _Panel_ number order, then you can reconstruct the covariance or inbreeding matrix.
- _Col_, a variable used to name columns of the inbreeding or covariance matrix. The values of this variable start with 'COL', followed by a number indicating the column number. The names of the individuals corresponding to any given column i can be found by reading the individual's name across the row that has a _Col_ value of 'COLi'. When the inbreeding or covariance matrix is divided into panels, all the rows repeat for the first n columns, all the rows repeat for the next n columns, and so on.
- the variable containing the names of the individuals, that is, the first variable listed in the VAR statement
- the variable containing the names of the first parents, that is, the second variable listed in the VAR statement
- the variable containing the names of the second parents, that is, the third variable listed in the VAR statement
- a list of covariance variables Col1–Col$n$, where n is the maximum number of individuals in the first population

The functions of the variables _Panel_ and _Col_ can best be demonstrated by an example. Assume that there are three individuals in the first BY group and that, in the current BY group (Byvar=2), there are five individuals with the following covariance matrix.

COV	1	2	3	4	5
1	Cov(1,1)	Cov(1,2)	Cov(1,3)	Cov(1,4)	Cov(1,5)
2	Cov(2,1)	Cov(2,2)	Cov(2,3)	Cov(2,4)	Cov(2,5)
3	Cov(3,1)	Cov(3,2)	Cov(3,3)	Cov(3,4)	Cov(3,5)
4	Cov(4,1)	Cov(4,2)	Cov(4,3)	Cov(4,4)	Cov(4,5)
5	Cov(5,1)	Cov(5,2)	Cov(5,3)	Cov(5,4)	Cov(5,5)
	_____ Panel 1 _____			___ Panel 2 ___	

Then the OUTCOV= data set appears as follows.

Byvar	_Panel_	_Col_	Individual	Parent	Parent2	Col1	Col2	Col3
2	1	COL1	1			Cov(1,1)	Cov(1,2)	Cov(1,3)
2	1	COL2	2			Cov(2,1)	Cov(2,2)	Cov(2,3)
2	1	COL3	3			Cov(3,1)	Cov(3,2)	Cov(3,3)
2	1		4			Cov(4,1)	Cov(4,2)	Cov(4,3)
2	1		5			Cov(5,1)	Cov(5,2)	Cov(5,3)
2	2		1			Cov(1,4)	Cov(1,5)	.
2	2		2			Cov(2,4)	Cov(2,5)	.
2	2		3			Cov(3,4)	Cov(3,5)	.
2	2	COL1	4			Cov(4,4)	Cov(4,5)	.
2	2	COL2	5			Cov(5,4)	Cov(5,5)	.

Notice that the first three columns go to the first panel (_Panel_=1), and the remaining two go to the second panel (_Panel_=2). Therefore, in the first panel, 'COL1', 'COL2', and 'COL3' correspond to individuals 1, 2, and 3, respectively, while in the second panel, 'COL1' and 'COL2' correspond to individuals 4 and 5, respectively.

Displayed Output

The INBREED procedure can output either covariance coefficients or inbreeding coefficients. Note that the following items can be produced for each generation if generations do not overlap.

The output produced by PROC INBREED can be any or all of the following items:

- a matrix of coefficients
- coefficients of the individuals
- coefficients for selected matings

ODS Table Names

PROC INBREED assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 44.2. For more information on ODS, see Chapter 20, "Using the Output Delivery System."

Table 44.2 ODS Tables Produced by PROC INBREED

ODS Table Name	Description	Statement	Option
AvgCovCoef	Averages of covariance coefficient matrix	GENDER	COVAR and AVERAGE
AvgInbreedingCoef	Averages of inbreeding coefficient matrix	GENDER	AVERAGE
CovarianceCoefficient	Covariance coefficient table	PROC	COVAR and MATRIX
InbreedingCoefficient	Inbreeding coefficient table	PROC	MATRIX
IndividualCovCoef	Covariance coefficients of individuals	PROC	IND and COVAR
IndividualInbreedingCoef	Inbreeding coefficients of individuals	PROC	IND
MatingCovCoef	Covariance coefficients of matings	MATINGS	COVAR
MatingInbreedingCoef	Inbreeding coefficients of matings	MATINGS	
NumberOfObservations	Number of observations	PROC	

Examples: INBREED Procedure

Example 44.1: Monoecious Population Analysis

The following example shows a covariance analysis within nonoverlapping generations for a monoecious population. Parents of generation 1 are unknown and therefore assumed to be unrelated. The following statements produce Output 44.1.1 through Output 44.1.3:

```
data Monoecious;
   input Generation Individual Parent1 Parent2 Covariance @@;
   datalines;
1 1 . .  .     1 2 . .  .     1 3 . .  .
2 1 1 1  .     2 2 1 2  .     2 3 2 3  .
3 1 1 2  .     3 2 1 3  .     3 3 2 1  .
3 4 1 3  .     3 . 2 3 0.50   3 . 4 3 1.135
;

title 'Inbreeding within Nonoverlapping Generations';
proc inbreed ind covar matrix data=Monoecious;
   class Generation;
run;
```

Output 44.1.1 Monoecious Population Analysis, Generation 1

Inbreeding within Nonoverlapping Generations

The INBREED Procedure

Generation = 1

Covariance Coefficients

Individual	Parent1	Parent2	1	2	3
1			1.0000	.	.
2			.	1.0000	.
3			.	.	1.0000

Covariance Coefficients of Individuals

Individual	Parent1	Parent2	Coefficient
1			1.0000
2			1.0000
3			1.0000

Number of Individuals 3

Output 44.1.2 Monoecious Population Analysis, Generation 2

Inbreeding within Nonoverlapping Generations

The INBREED Procedure

Generation = 2

Covariance Coefficients

Individual	Parent1	Parent2	1	2	3
1	1	1	1.5000	0.5000	.
2	1	2	0.5000	1.0000	0.2500
3	2	3	.	0.2500	1.0000

Covariance Coefficients of Individuals

Individual	Parent1	Parent2	Coefficient
1	1	1	1.5000
2	1	2	1.0000
3	2	3	1.0000

Number of Individuals 3

Output 44.1.3 Monoecious Population Analysis, Generation 3

Inbreeding within Nonoverlapping Generations

The INBREED Procedure

Generation = 3

Covariance Coefficients

Individual	Parent1	Parent2	1	2	3	4
1	1	2	1.2500	0.5625	0.8750	0.5625
2	1	3	0.5625	1.0000	1.1349	0.6250
3	2	1	0.8750	1.1349	1.2500	1.1349
4	1	3	0.5625	0.6250	1.1349	1.0000

Covariance Coefficients of Individuals

Individual	Parent1	Parent2	Coefficient
1	1	2	1.2500
2	1	3	1.0000
3	2	1	1.2500
4	1	3	1.0000

Number of Individuals 4

Note that, since the parents of the first generation are unknown, off-diagonal elements of the covariance matrix are all 0s and on-diagonal elements are all 1s. If there is an INIT=*cov* value, then the off-diagonal elements would be equal to *cov*, while on-diagonal elements would be equal to $1 + cov/2$.

In the third generation, individuals 2 and 4 are full siblings, so they belong to the same family. Since PROC INBREED computes covariance coefficients between families, the second and fourth columns of inbreeding coefficients are the same, except that their intersections with the second and fourth rows are reordered. Notice that, even though there is an observation to assign a covariance of 0.50 between individuals 2 and 3 in the third generation, the covariance between 2 and 3 is set to 1.135, the same value assigned between 4 and 3. This is because families get the same covariances, and later specifications override previous ones.

Example 44.2: Pedigree Analysis

In the following example, an inbreeding analysis is performed for a complicated pedigree. This analysis includes computing selective matings of some individuals and inbreeding coefficients of all individuals. Also, inbreeding coefficients are averaged within sex categories. The following statements produce Output 44.2.1:

```
data Swine;
   input Swine_Number $ Sire $ Dam $ Sex $;
   datalines;
3504 2200 2501  M
3514 2521 3112  F
3519 2521 2501  F
2501 2200 3112  M
2789 3504 3514  F
3501 2521 3514  M
3712 3504 3514  F
3121 2200 3501  F
;

title 'Least Related Matings';
proc inbreed data=Swine ind average;
   var Swine_Number Sire Dam;
   matings 2501 / 3501 3504 ,
           3712 / 3121;
   gender Sex;
run;
```

Note the following from Output 44.2.1:

- Observation 4, which defines Swine_Number=2501, should precede the first and third observations where the progeny for 2501 are given. PROC INBREED ignores observation 4 since it is given out of order. As a result, the parents of 2501 are missing or unknown.

- The first column in the "Inbreeding Averages" table corresponds to the averages taken over the on-diagonal elements of the inbreeding coefficients matrix, and the second column gives averages over the off-diagonal elements.

Output 44.2.1 Pedigree Analysis

Least Related Matings

The INBREED Procedure

Inbreeding Coefficients of Individuals

Swine_ Number	Sire	Dam	Coefficient
2200			.
2501			.
3504	2200	2501	.
2521			.
3112			.
3514	2521	3112	.
3519	2521	2501	.
2789	3504	3514	.
3501	2521	3514	0.2500
3712	3504	3514	.
3121	2200	3501	.

Inbreeding Coefficients of Matings

Sire	Dam	Coefficient
2501	3501	.
2501	3504	0.2500
3712	3121	0.1563

Averages of Inbreeding Coefficient Matrix

	Inbreeding	Coancestry
Male X Male	0.0625	0.1042
Male X Female	.	0.1362
Female X Female	0.0000	0.1324
Over Sex	0.0227	0.1313

Number of Males	4
Number of Females	7
Number of Individuals	11

Example 44.3: Pedigree Analysis with BY Groups

This example demonstrates the structure of the OUTCOV= data set created by PROC INBREED. Note that the first BY group has three individuals, while the second has five. Therefore, the covariance matrix for the second BY group is broken up into two panels. The following statements produce Output 44.3.1.

```
data Swine;
   input Group Swine_Number $ Sire $ Dam $ Sex $;
   datalines;
1  2789 3504 3514  F
2  2501 2200 3112  .
2  3504 2501 3782  M
;
proc inbreed data=Swine covar noprint outcov=Covariance
             init=0.4;
   var Swine_Number Sire Dam;
   gender Sex;
   by Group;
run;

title 'Printout of OUTCOV= data set';
proc print data=Covariance;
   format Col1-Col3 4.2;
run;
```

Output 44.3.1 Pedigree Analysis with BY Groups

Printout of OUTCOV= data set

Obs	Group	Sex	_TYPE_	_PANEL_	_COL_	Swine_ Number	Sire	Dam	COL1	COL2	COL3
1	1	M	COV	1	COL1	3504			1.20	0.40	0.80
2	1	F	COV	1	COL2	3514			0.40	1.20	0.80
3	1	F	COV	1	COL3	2789	3504	3514	0.80	0.80	1.20
4	2	M	COV	1	COL1	2200			1.20	0.40	0.80
5	2	F	COV	1	COL2	3112			0.40	1.20	0.80
6	2	M	COV	1	COL3	2501	2200	3112	0.80	0.80	1.20
7	2	F	COV	1		3782			0.40	0.40	0.40
8	2	M	COV	1		3504	2501	3782	0.60	0.60	0.80
9	2	M	COV	2		2200			0.40	0.60	.
10	2	F	COV	2		3112			0.40	0.60	.
11	2	M	COV	2		2501	2200	3112	0.40	0.80	.
12	2	F	COV	2	COL1	3782			1.20	0.80	.
13	2	M	COV	2	COL2	3504	2501	3782	0.80	1.20	.

References

Crow, J. F. and Kimura, M. (1970), *An Introduction to Population Genetics Theory*, New York: Harper and Row.

Falconer, D. S. and Mackay, T. F. C. (1996), *Introduction to Quantitative Genetics*, Fourth Edition, London: Longman.

Kempthorne, O. (1957), *An Introduction to Genetic Statistics*, New York: John Wiley & Sons.

Chapter 45

The KDE Procedure

Contents

Overview: KDE Procedure **2885**
Getting Started: KDE Procedure **2886**
Syntax: KDE Procedure **2889**
PROC KDE Statement 2889
BIVAR Statement 2890
UNIVAR Statement 2893
BY Statement 2896
FREQ Statement 2896
WEIGHT Statement 2897
Details: KDE Procedure **2897**
Computational Overview 2897
Kernel Density Estimates 2897
Binning 2899
Convolutions 2900
Fast Fourier Transform 2902
Bandwidth Selection 2903
ODS Table Names 2904
ODS Graphics 2904
Examples: KDE Procedure **2907**
Example 45.1: Computing a Basic Kernel Density Estimate 2907
Example 45.2: Changing the Bandwidth 2909
Example 45.3: Changing the Bandwidth (Bivariate) 2911
Example 45.4: Requesting Additional Output Tables 2913
Example 45.5: Univariate KDE Graphics 2916
Example 45.6: Bivariate KDE Graphics 2920
References **2927**

Overview: KDE Procedure

The KDE procedure performs univariate and bivariate kernel density estimation. Statistical *density estimation* involves approximating a hypothesized probability density function from observed data.

Kernel density estimation is a nonparametric technique for density estimation in which a known density function (the *kernel*) is averaged across the observed data points to create a smooth approximation. PROC KDE uses a Gaussian density as the kernel, and its assumed variance determines the smoothness of the resulting estimate. Refer to Silverman (1986) for a thorough review and discussion.

You can use PROC KDE to compute a variety of common statistics, including estimates of the percentiles of the hypothesized probability density function. You can produce a variety of plots, including univariate and bivariate histograms, plots of the kernel density estimates, and contour plots. You can also save kernel density estimates into SAS data sets.

Getting Started: KDE Procedure

The following example illustrates the basic features of PROC KDE. Assume that 1000 observations are simulated from a bivariate normal density with means $(0, 0)$, variances $(10, 10)$, and covariance 9. The SAS DATA step to accomplish this is as follows:

```
data bivnormal;
   seed = 1283470;
   do i = 1 to 1000;
      z1 = rannor(seed);
      z2 = rannor(seed);
      z3 = rannor(seed);
      x = 3*z1+z2;
      y = 3*z1+z3;
      output;
   end;
   drop seed;
run;
```

The following statements request a bivariate kernel density estimate for the variables `x` and `y`, with contour and surface plots:

```
ods graphics on;
proc kde data=bivnormal;
   bivar x y / plots=(contour surface);
run;
ods graphics off;
```

The contour plot and the surface plot of the estimate are displayed in Figure 45.1 and Figure 45.2, respectively. Note that the correlation of 0.9 in the original data results in oval-shaped contours. These graphs are produced by specifying the ODS GRAPHICS statement prior to the PROC statement and the PLOTS= option in the BIVAR statement. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the KDE procedure, see the section "ODS Graphics" on page 2904.

Figure 45.1 Contour Plot of Estimated Density

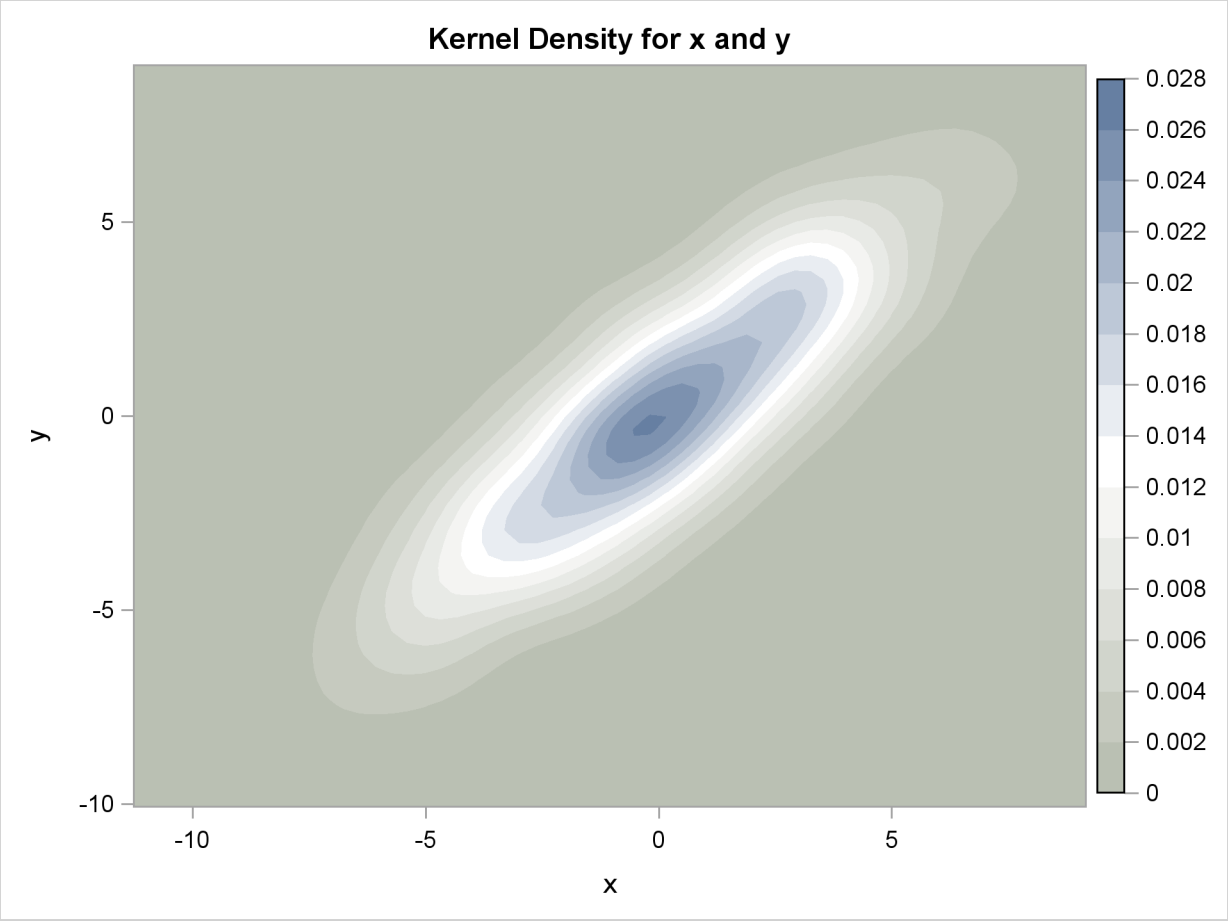
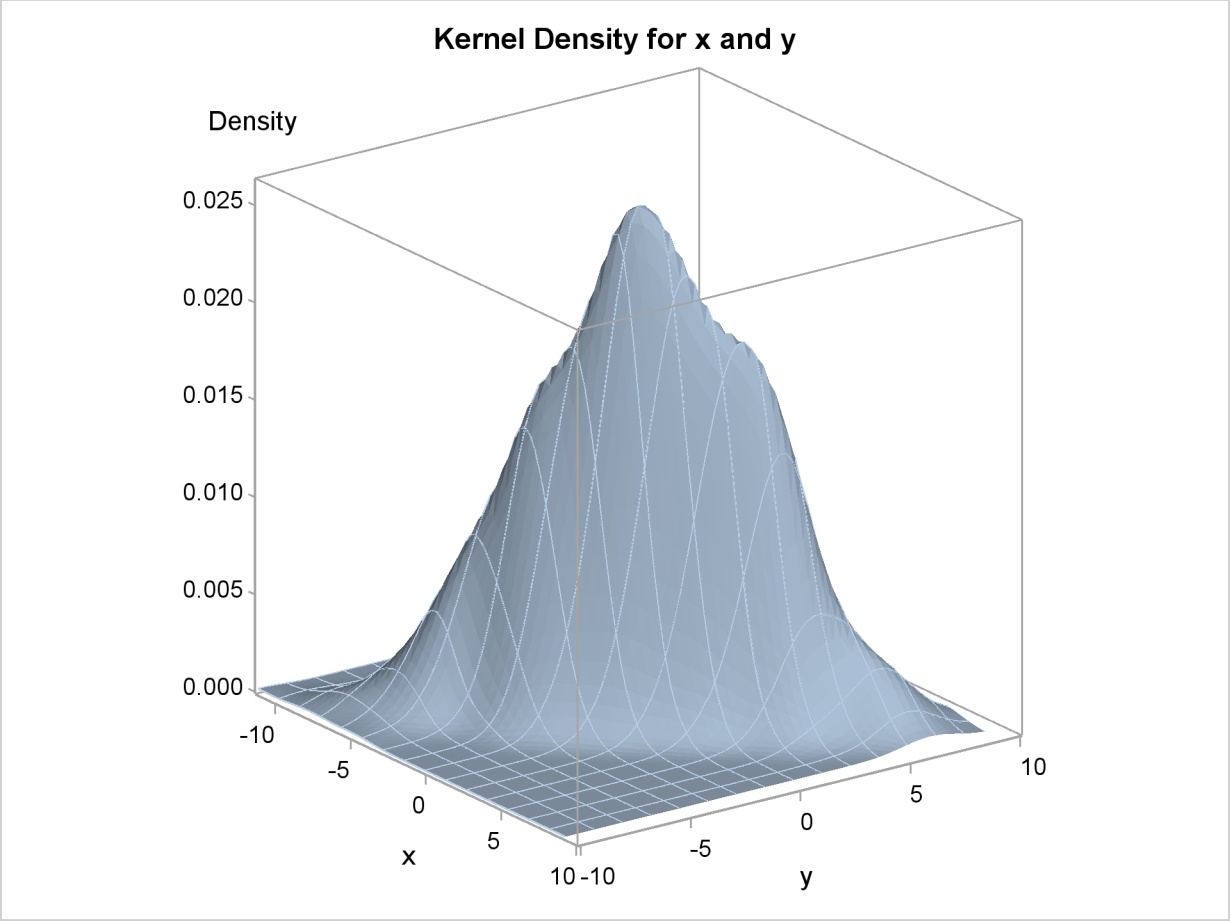


Figure 45.2 Surface Plot of Estimated Density



The default output tables for this analysis are shown in Figure 45.3.

Figure 45.3 Default Bivariate Tables

The KDE Procedure

Inputs

Data Set	WORK.BIVNORMAL
Number of Observations Used	1000
Variable 1	x
Variable 2	y
Bandwidth Method	Simple Normal Reference

Controls

	x	y
Grid Points	60	60
Lower Grid Limit	-11.25	-10.05
Upper Grid Limit	9.1436	9.0341
Bandwidth Multiplier	1	1

The "Inputs" table lists basic information about the density fit, including the input data set, the number of observations, and the variables. The bandwidth method is the technique used to select the amount of smoothing in the estimate. A simple normal reference rule is used for bivariate smoothing.

The "Controls" table lists the primary numbers controlling the kernel density fit. Here a 60×60 grid is fit to the entire range of the data, and no adjustment is made to the default bandwidth.

Syntax: KDE Procedure

You can use the following statements with the KDE procedure:

PROC KDE < *options* > **;**
BIVAR *variable-list* < / *options* > **;**
UNIVAR *variable-list* < / *options* > **;**
BY *variables* **;**
FREQ *variable* **;**
WEIGHT *variable* **;**

The PROC KDE statement invokes the procedure. The BIVAR statement requests that one or more bivariate kernel density estimates be computed. The UNIVAR statement requests one or more univariate kernel density estimates. You can specify any number of BIVAR and UNIVAR statements.

PROC KDE Statement

PROC KDE < *options* > **;**

The PROC KDE statement invokes the procedure and specifies the input data set.

DATA=*SAS-data-set*
specifies the input SAS data set to be used by PROC KDE. The default is the most recently created data set.

NOTE: The following options, which were available in the PROC KDE statement in SAS 8, are now obsolete. These options are now available in the UNIVAR and BIVAR statements.

SAS 8	SAS 9.2	
PROC KDE option	**UNIVAR option**	**BIVAR option**
BWM=*numlist*	BWM=*number*	BWM=*number*
GRIDL=*numlist*	GRIDL=*number*	GRIDL=*number*
GRIDU=*numlist*	GRIDU=*number*	GRIDU=*number*
LEVELS		LEVELS
METHOD	METHOD	
NGRID=*numlist*	NGRID=*number*	NGRID=*number*
OUT	OUT	OUT
PERCENTILES	PERCENTILES	PERCENTILES
SJPIMAX	SJPIMAX	
SJPIMIN	SJPIMIN	
SJPINUM	SJPINUM	
SJPITOL	SJPITOL	

BIVAR Statement

The BIVAR statement computes bivariate kernel density estimates. The basic syntax for the BIVAR statement specifies two variables:

BIVAR *v1 < (v-options) > v2 < (v-options) > < / options >* **;**

This statement requests a bivariate kernel density estimate for the variables v1 and v2. The *v-options* optionally specified in parentheses after a variable name apply only to that variable, and override corresponding global *options* specified following a slash (/).

You can specify a list of more than two variables:

BIVAR *v1 < (v-options) > v2 < (v-options) > . . . vN < (v-options) > < / options >* **;**

This statement requests a bivariate kernel density estimate for each distinct pair of variables in the list. For example, if you specify

```
bivar x y z;
```

then a bivariate kernel density estimate is computed for each of the variable pairs (x, y), (x, z), and (y, z).

Alternatively, you can specify an explicit list of variable pairs, with each pair enclosed in parentheses:

BIVAR *(v1 v2) (v3 v4) . . . (vN-1 vN) < / options >* **;**

(You can also specify *v-options* following a variable name appearing in an explicit pair, but they are omitted here for clarity.) This statement requests a bivariate kernel density estimate for each pair of variables. For example, if you specify

```
bivar (x y) (y z);
```

then bivariate kernel density estimates are computed for (x, y) and (y, z).

NOTE: The VAR statement supported by PROC KDE in SAS 8 and earlier releases is now obsolete. The VAR statement has been replaced by the UNIVAR and the BIVAR statements, which enable you to produce multiple kernel density estimates with a single invocation of the procedure.

You can specify the following options in the BIVAR statement. As noted, some options can be used as *v-options*.

BIVSTATS

produces a table for each density estimate containing the covariance and correlation between the two variables.

BWM=*number*

specifies the bandwidth multiplier applied to each variable in each kernel density estimate. The default value is 1. Larger multipliers produce a smoother estimate, and smaller ones produce a rougher estimate. To specify different bandwidth multipliers for different variables, specify BWM= as a *v-option*.

GRIDL=*number*

specifies the lower grid limit applied to each variable in each kernel density estimate. The default value for a given variable is the minimum observed value of that variable. To specify different lower grid limits for different variables, specify GRIDL= as a *v-option*.

GRIDU=*number*

specifies the upper grid limit applied to each variable in each kernel density estimate. The default value for a given variable is the maximum observed value of that variable. To specify different upper grid limits for different variables, specify GRIDU= as a *v-option*.

LEVELS

LEVELS=*numlist*

requests a table of levels for contours of the bivariate density. The contours are defined in such a way that the density has a constant level along each contour, and the volume enclosed by each contour corresponds to a specified percent. In other words, the contours correspond to slices or levels of the density surface taken along the density axis. You can specify the percents used to define the contours. The default values are 1, 5, 10, 50, 90, 95, 99, and 100. The "Levels" table also provides the minimum and maximum values for each contour along the directions of the two data variables.

NGRID=*number*

NG=*number*

specifies the number of grid points associated with each variable in each kernel density estimate. The default value is 60. To specify different numbers of grid points for different variables, specify NGRID= as a *v-option*.

NOPRINT

suppresses output tables produced by the BIVAR statement. You can use the NOPRINT option when you want to produce graphical output only.

OUT=*SAS-data-set*

specifies the name of the output data set in which kernel density estimates are saved. This output data set contains the following variables:

- var1, whose value is the name of the first variable in a bivariate kernel density estimate
- var2, whose value is the name of the second variable in a bivariate kernel density estimate
- value1, with values corresponding to grid coordinates for the first variable
- value2, with values corresponding to grid coordinates for the second variable
- density, with values equal to kernel density estimates at the associated grid point
- count, containing the number of original observations contained in the bin corresponding to a grid point

PERCENTILES

PERCENTILES=*numlist*

requests that a table of percentiles be computed for each BIVAR variable. You can specify a list of percentiles to be computed. The default percentiles are 0.5, 1, 2.5, 5, 10, 25, 50, 75, 90, 95, 97.5, 99, and 99.5.

PLOTS=*plot-request*< *(options)*> | **ALL** | **NONE**

PLOTS=(*plot-request*< *(options)*> **<** . . . *plot-request*< *(options)*> **>)**

requests one or more plots of the bivariate data and kernel density estimate. By default, if you enable ODS Graphics and you do not specify the PLOTS= option, then the BIVAR statement creates a contour plot. If you specify the PLOTS= option, you get only the requested plots.

The following *plot-requests* are available.

ALL

produces all bivariate plots.

CONTOUR

produces a contour plot of the bivariate density estimate.

CONTOURSCATTER

produces a contour plot of the bivariate density estimate overlaid with a scatter plot of the data.

HISTOGRAM <(*view-options***)>**

produces a bivariate histogram of the data. The following *view-options* can be specified:

ROTATE=*angle* rotates the histogram *angle* degrees, where $-180 < angle < 180$. By default, $angle = 54$.

TILT=*angle* tilts the histogram *angle* degrees, where $-180 < angle < 180$. By default, $angle = 20$.

HISTSURFACE <(*view-options***)>**

produces a bivariate histogram of the data overlaid with a surface plot of the bivariate kernel density estimate. The following *view-options* can be specified:

ROTATE=*angle* rotates the histogram and kernel density surface *angle* degrees, where –180 < *angle* < 180. By default, *angle* = 54.

TILT=*angle* tilts the histogram and kernel density surface *angle* degrees, where –180 < *angle* < 180. By default, *angle* = 20.

NONE
suppresses all plots, including the contour plot that is produced by default when ODS Graphics is enabled and the PLOTS= option is not specified.

SCATTER
produces a scatter plot of the data.

SURFACE < (*view-options***) >**
produces a surface plot of the bivariate kernel density estimate. The following *view-options* can be specified:

ROTATE=*angle* rotates the kernel density surface *angle* degrees, where –180 < *angle* < 180. By default, *angle* = 54.

TILT=*angle* tilts the kernel density surface *angle* degrees, where –180 < *angle* < 180. By default, *angle* = 20.

UNISTATS
produces a table for each density estimate containing standard univariate statistics for each of the two variables and the bandwidths used to compute the kernel density estimate. The statistics listed are the mean, variance, standard deviation, range, and interquartile range.

UNIVAR Statement

UNIVAR *variable* < *(v-options)* > < . . . *variable* < *(v-options)* > > < */ options* > **;**

The UNIVAR statement computes univariate kernel density estimates. You can specify various *v-options* for each variable by enclosing them in parentheses after the variable name. You can also specify global *options* among the UNIVAR statement options following a slash (/). Global *options* apply to all the variables specified in the UNIVAR statement. However, individual variable *v-options* override the global *options*.

NOTE: The VAR statement supported by PROC KDE in SAS 8 and earlier releases is now obsolete. The VAR statement has been replaced by the UNIVAR and BIVAR statements, which enable you to produce multiple kernel density estimates with a single invocation of the procedure.

You can specify the following options in the UNIVAR statement. As noted, some options can be used as *v-options*.

BWM=*number*
specifies a bandwidth multiplier used for each kernel density estimate. The default value is 1. Larger multipliers produce a smoother estimate, and smaller ones produce a rougher estimate. To specify different bandwidth multipliers for different variables, specify BWM= as a *v-option*.

GRIDL=*number*

specifies a lower grid limit used for each kernel density estimate. The default value for a given variable is the minimum observed value of that variable. To specify different lower grid limits for different variables, specify GRIDL= as a *v-option*.

GRIDU=*number*

specifies an upper grid limit used for each kernel density estimate. The default value for a given variable is the maximum observed value of that variable. To specify different upper grid limits for different variables, specify GRIDU= as a *v-option*.

METHOD=SJPI | SNR | SNRQ | SROT | OS

specifies the method used to compute the bandwidth. Available methods are Sheather-Jones plug-in (SJPI), simple normal reference (SNR), simple normal reference that uses the interquartile range (SNRQ), Silverman's rule of thumb (SROT), and oversmoothed (OS). See the section "Bandwidth Selection" on page 2903 and refer to Jones, Marron, and Sheather (1996) for a description of these methods. SJPI is the default method.

NGRID=*number*

NG=*number*

specifies a number of grid points used for each kernel density estimate. The default value is 401. To specify different numbers of grid points for different variables, specify NGRID= as a *v-option*.

NOPRINT

suppresses output tables produced by the UNIVAR statement. You can use the NOPRINT option when you want to produce graphical output only.

OUT=*SAS-data-set*

specifies the output SAS data set containing the kernel density estimate. This output data set contains the following variables:

- var, whose value is the name of the variable in the kernel density estimate
- value, with values corresponding to grid coordinates for the variable
- density, with values equal to kernel density estimates at the associated grid point
- count, containing the number of original observations contained in the bin corresponding to a grid point

PERCENTILES

PERCENTILES=*numlist*

requests that a table of percentiles be computed for each UNIVAR variable. You can specify a list of percentiles to be computed. The default percentiles are 0.5, 1, 2.5, 5, 10, 25, 50, 75, 90, 95, 97.5, 99, and 99.5.

PLOTS=*plot-request* **| ALL | NONE**

PLOTS=(*plot-request* **<** *. . . plot-request* **>)**

requests plots of the univariate kernel density estimate. The following table shows the available *plot-requests*.

Keyword	Description
ALL	produces all plots
DENSITY	univariate kernel density estimate curve
DENSITYOVERLAY	overlaid univariate kernel density estimate curves
HISTDENSITY	univariate histogram of data overlaid with kernel density estimate curve
HISTOGRAM	univariate histogram of data
NONE	suppresses all plots

By default, if you enable ODS Graphics and you do not specify the PLOTS= option, then the UNIVAR statement creates a histogram overlaid with a kernel density estimate. If you specify the PLOTS= option, you get only the requested plots.

If you specify more than one variable in the UNIVAR statement, the DENSITYOVERLAY keyword overlays the density curves for all the variables on a single plot. The other keywords each produce a separate plot for every variable listed in the UNIVAR statement.

SJPIMAX=*number*

specifies the maximum grid value in determining the Sheather-Jones plug-in bandwidth. The default value is two times the oversmoothed estimate.

SJPIMIN=*number*

specifies the minimum grid value in determining the Sheather-Jones plug-in bandwidth. The default value is the maximum value divided by 18.

SJPINUM=*number*

specifies the number of grid values used in determining the Sheather-Jones plug-in bandwidth. The default is 21.

SJPITOL=*number*

specifies the tolerance for termination of the bisection algorithm used in computing the Sheather-Jones plug-in bandwidth. The default value is 0.001.

UNISTATS

produces a table for each variable containing standard univariate statistics and the bandwidth used to compute its kernel density estimate. The statistics listed are the mean, variance, standard deviation, range, and interquartile range.

Examples

Suppose you have the variables x1, x2, x3, and x4 in the SAS data set MyData. You can request a univariate kernel density estimate for each of these variables with the following statements:

```
proc kde data=MyData;
   univar x1 x2 x3 x4;
run;
```

You can also specify different bandwidths and other options for each variable. For example, the following statements request kernel density estimates that use Silverman's rule of thumb (SROT) method for all variables:

```
proc kde data=MyData;
   univar x1 (bwm=2)
          x2 (bwm=0.5 ngrid=100)
          x3 x4 / ngrid=200 method=srot;
run;
```

The option NGRID=200 applies to the variables x1, x3, and x4, but the *v-option* NGRID=100 is applied to x2. Bandwidth multipliers of 2 and 0.5 are specified for the variables x1 and x2, respectively.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC KDE to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the KDE procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

FREQ Statement

FREQ *variable* **;**

The FREQ statement specifies a variable that provides frequencies for each observation in the DATA= data set. Specifically, if n is the value of the FREQ variable for a given observation, then that observation is used n times. If the value of the FREQ variable is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement specifies a variable that weights the observations in computing the kernel density estimate. Observations with higher weights have more influence in the computations. If an observation has a nonpositive or missing weight, then the entire observation is omitted from the analysis. You should be cautious in using data sets with extreme weights, because they can produce unreliable results.

Details: KDE Procedure

Computational Overview

The two main computational tasks of PROC KDE are automatic bandwidth selection and the construction of a kernel density estimate once a bandwidth has been selected. The primary computational tools used to accomplish these tasks are binning, convolutions, and the fast Fourier transform. The following sections provide analytical details on these topics, beginning with the density estimates themselves.

Kernel Density Estimates

A weighted univariate kernel density estimate involves a variable X and a weight variable W. Let $(X_i, W_i),\ i = 1, 2, \ldots, n$, denote a sample of X and W of size n. The weighted kernel density estimate of $f(x)$, the density of X, is as follows:

$$\hat{f}(x) = \frac{1}{\sum_{i=1}^{n} W_i} \sum_{i=1}^{n} W_i \varphi_h(x - X_i)$$

where h is the bandwidth and

$$\varphi_h(x) = \frac{1}{\sqrt{2\pi}h} \exp\left(-\frac{x^2}{2h^2}\right)$$

is the standard normal density rescaled by the bandwidth. If $h \to 0$ and $nh \to \infty$, then the optimal bandwidth is

$$h_{\text{AMISE}} = \left[\frac{1}{2\sqrt{\pi}n \int (f'')^2}\right]^{1/5}$$

This optimal value is unknown, and so approximations methods are required. For a derivation and discussion of these results, refer to Silverman (1986, Chapter 3) and Jones, Marron, and Sheather (1996).

For the bivariate case, let $\mathbf{X} = (X, Y)$ be a bivariate random element taking values in R^2 with joint density function

$$f(x, y),\ (x, y) \in R^2$$

and let $\mathbf{X}_i = (X_i, Y_i),\ i = 1, 2, \ldots, n$, be a sample of size n drawn from this distribution. The kernel density estimate of $f(x, y)$ based on this sample is

$$\begin{aligned}\hat{f}(x, y) &= \frac{1}{n}\sum_{i=1}^{n}\varphi_{\mathbf{h}}(x - X_i, y - Y_i) \\ &= \frac{1}{nh_Xh_Y}\sum_{i=1}^{n}\varphi\left(\frac{x - X_i}{h_X}, \frac{y - Y_i}{h_Y}\right)\end{aligned}$$

where $(x, y) \in R^2$, $h_X > 0$ and $h_Y > 0$ are the bandwidths, and $\varphi_{\mathbf{h}}(x, y)$ is the rescaled normal density

$$\varphi_{\mathbf{h}}(x, y) = \frac{1}{h_Xh_Y}\varphi\left(\frac{x}{h_X}, \frac{y}{h_Y}\right)$$

where $\varphi(x, y)$ is the standard normal density function

$$\varphi(x, y) = \frac{1}{2\pi}\exp\left(-\frac{x^2 + y^2}{2}\right)$$

Under mild regularity assumptions about $f(x, y)$, the mean integrated squared error (MISE) of $\hat{f}(x, y)$ is

$$\begin{aligned}\mathrm{MISE}(h_X, h_Y) &= \mathrm{E}\int(\hat{f} - f)^2 \\ &= \frac{1}{4\pi nh_Xh_Y} + \frac{h_X^4}{4}\int\left(\frac{\partial^2 f}{\partial X^2}\right)^2 dxdy \\ &\quad + \frac{h_Y^4}{4}\int\left(\frac{\partial^2 f}{\partial Y^2}\right)^2 dxdy + O\left(h_X^4 + h_Y^4 + \frac{1}{nh_Xh_Y}\right)\end{aligned}$$

as $h_X \to 0$, $h_Y \to 0$ and $nh_Xh_Y \to \infty$.

Now set

$$\begin{aligned}\mathrm{AMISE}(h_X, h_Y) &= \frac{1}{4\pi nh_Xh_Y} + \frac{h_X^4}{4}\int\left(\frac{\partial^2 f}{\partial X^2}\right)^2 dxdy \\ &\quad + \frac{h_Y^4}{4}\int\left(\frac{\partial^2 f}{\partial Y^2}\right)^2 dxdy\end{aligned}$$

which is the asymptotic mean integrated squared error (AMISE). For fixed n, this has a minimum at $(h_{\text{AMISE\_}X}, h_{\text{AMISE\_}Y})$ defined as

$$h_{\text{AMISE\_}X} = \left[\frac{\int(\frac{\partial^2 f}{\partial X^2})^2}{4n\pi}\right]^{1/6}\left[\frac{\int(\frac{\partial^2 f}{\partial X^2})^2}{\int(\frac{\partial^2 f}{\partial Y^2})^2}\right]^{2/3}$$

and

$$h_{\text{AMISE\_}Y} = \left[\frac{\int(\frac{\partial^2 f}{\partial Y^2})^2}{4n\pi}\right]^{1/6}\left[\frac{\int(\frac{\partial^2 f}{\partial Y^2})^2}{\int(\frac{\partial^2 f}{\partial X^2})^2}\right]^{2/3}$$

These are the optimal asymptotic bandwidths in the sense that they minimize MISE. However, as in the univariate case, these expressions contain the second derivatives of the unknown density f being estimated, and so approximations are required. Refer to Wand and Jones (1993) for further details.

Binning

Binning, or assigning data to discrete categories, is an effective and fast method for large data sets (Fan and Marron 1994). When the sample size n is large, direct evaluation of the kernel estimate $\hat{f}$ at any point would involve n kernel evaluations, as shown in the preceding formulas. To evaluate the estimate at each point of a grid of size g would thus require ng kernel evaluations. When you use $g = 401$ in the univariate case or $g = 60 \times 60 = 3600$ in the bivariate case and $n \geq 1000$, the amount of computation can be prohibitively large. With binning, however, the computational order is reduced to g, resulting in a much quicker algorithm that is nearly as accurate as direct evaluation.

To bin a set of weighted univariate data $X_1, X_2, \ldots, X_n$ to a grid $x_1, x_2, \ldots, x_g$, simply assign each sample X_i, together with its weight W_i, to the nearest grid point x_j (also called the bin center). When binning is completed, each grid point x_i has an associated number c_i, which is the sum total of all the weights that correspond to sample points that have been assigned to x_i. These c_is are known as the *bin counts.*

This procedure replaces the data (X_i, W_i), $i = 1, 2, \ldots, n$, with the smaller set (x_i, c_i), $i = 1, 2, \ldots, g$, and the estimation is carried out with these new data. This is so-called *simple binning,* versus the finer *linear binning* described in Wand (1994). PROC KDE uses simple binning for the sake of faster and easier implementation. Also, it is assumed that the bin centers $x_1, x_2, \ldots, x_g$ are equally spaced and in increasing order. In addition, assume for notational convenience that $\sum_{i=1}^{n} W_i = n$ and, therefore, $\sum_{i=1}^{g} c_i = n$.

If you replace the data $(X_i, W_i), i = 1, 2, \ldots, n$, with $(x_i, c_i), i = 1, 2, \ldots, g$, the weighted estimator $\hat{f}$ then becomes

$$\hat{f}(x) = \frac{1}{n}\sum_{i=1}^{g} c_i \varphi_h(x - x_i)$$

with the same notation as used previously. To evaluate this estimator at the g points of the same grid vector $grid = (x_1, x_2, \ldots, x_g)'$ is to calculate

$$\hat{f}(x_i) = \frac{1}{n} \sum_{j=1}^{g} c_j \varphi_h(x_i - x_j)$$

for $i = 1, 2, \ldots, g$. This can be rewritten as

$$\hat{f}(x_i) = \frac{1}{n} \sum_{j=1}^{g} c_j \varphi_h(|i - j|\delta)$$

where $\delta = x_2 - x_1$ is the increment of the grid.

The same idea of binning works similarly with bivariate data, where you estimate $\hat{f}$ over the grid matrix $grid = grid_X \times grid_Y$ as follows:

$$grid = \begin{bmatrix} \mathbf{x}_{1,1} & \mathbf{x}_{1,2} & \ldots & \mathbf{x}_{1,g_Y} \\ \mathbf{x}_{2,1} & \mathbf{x}_{2,2} & \ldots & \mathbf{x}_{2,g_Y} \\ \vdots & & & \\ \mathbf{x}_{g_X,1} & \mathbf{x}_{g_X,2} & \ldots & \mathbf{x}_{g_X,g_Y} \end{bmatrix}$$

where $\mathbf{x}_{i,j} = (x_i, y_i),\ i = 1, 2, \ldots, g_X,\ j = 1, 2, \ldots, g_Y$, and the estimates are

$$\hat{f}(\mathbf{x}_{i,j}) = \frac{1}{n} \sum_{k=1}^{g_X} \sum_{l=1}^{g_Y} c_{k,l} \varphi_{\mathbf{h}}(|i - k|\delta_X, |j - l|\delta_Y)$$

where $\delta_X = x_2 - x_1$ and $\delta_Y = y_2 - y_1$ are the increments of the grid.

Convolutions

The formulas for the binned estimator $\hat{f}$ in the previous subsection are in the form of a convolution product between two matrices, one of which contains the bin counts, the other of which contains the rescaled kernels evaluated at multiples of grid increments. This section defines these two matrices explicitly, and shows that $\hat{f}$ is their convolution.

Beginning with the weighted univariate case, define the following matrices:

$$\begin{aligned} K &= \frac{1}{n}(\varphi_h(0), \varphi_h(\delta), \ldots, \varphi_h((g-1)\delta))' \\ C &= (c_1, c_2, \ldots, c_g)' \end{aligned}$$

The first thing to note is that many terms in K are negligible. The term $\varphi_h(i\delta)$ is taken to be 0 when $|i\delta/h| \geq 5$, so you can define

$$l = \min(g - 1, \text{floor}(5h/\delta))$$

as the maximum integer multiple of the grid increment to get nonzero evaluations of the rescaled kernel. Here floor(x) denotes the largest integer less than or equal to x.

Next, let p be the smallest power of 2 that is greater than $g + l + 1$,

$$p = 2^{\text{ceil}(\log_2(g+l+1))}$$

where ceil(x) denotes the smallest integer greater than or equal to x.

Modify K as follows:

$$K = \frac{1}{n}(\varphi_h(0), \varphi_h(\delta), \ldots, \varphi_h(l\delta), \underbrace{0, \ldots, 0}_{p-2l-1}, \varphi_h(l\delta), \ldots, \varphi_h(\delta))'$$

Essentially, the negligible terms of K are omitted, and the rest are *symmetrized* (except for one term). The whole matrix is then padded to size $p \times 1$ with zeros in the middle. The dimension p is a highly composite number—that is, one that decomposes into many factors—leading to the most efficient fast Fourier transform operation (refer to Wand 1994).

The third operation is to pad the bin count matrix C with zeros to the same size as K:

$$C = (c_1, c_2, \ldots, c_g, \underbrace{0, \ldots, 0}_{p-g})'$$

The convolution $K * C$ is then a $p \times 1$ matrix, and the preceding formulas show that its first g entries are exactly the estimates $\hat{f}(x_i)$, $i = 1, 2, \ldots, g$.

For bivariate smoothing, the matrix K is defined similarly as

$$K = \begin{bmatrix} \kappa_{0,0} & \kappa_{0,1} & \ldots & \kappa_{0,l_Y} & \mathbf{0} & \kappa_{0,l_Y} & \ldots & \kappa_{0,1} \\ \kappa_{1,0} & \kappa_{1,1} & \ldots & \kappa_{1,l_Y} & \mathbf{0} & \kappa_{1,l_Y} & \ldots & \kappa_{1,1} \\ \vdots & & & & & & & \\ \kappa_{l_X,0} & \kappa_{l_X,1} & \ldots & \kappa_{l_X,l_Y} & \mathbf{0} & \kappa_{l_X,l_Y} & \ldots & \kappa_{l_X,1} \\ \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} & \mathbf{0} & \mathbf{0} & \ldots & \mathbf{0} \\ \kappa_{l_X,0} & \kappa_{l_X,1} & \ldots & \kappa_{l_X,l_Y} & \mathbf{0} & \kappa_{l_X,l_Y} & \ldots & \kappa_{l_X,1} \\ \vdots & & & & & & & \\ \kappa_{1,0} & \kappa_{1,1} & \ldots & \kappa_{1,l_Y} & \mathbf{0} & \kappa_{1,l_Y} & \ldots & \kappa_{1,1} \end{bmatrix}_{p_X \times p_Y}$$

where $l_X = \min(g_X - 1, \text{floor}(5h_X/\delta_X))$, $p_X = 2^{\text{ceil}(\log_2(g_X + l_X + 1))}$, and so forth, and $\kappa_{i,j} = \frac{1}{n}\varphi_{\mathbf{h}}(i\delta_X, j\delta_Y)$ $i = 0, 1, \ldots, l_X$, $j = 0, 1, \ldots, l_Y$.

The bin count matrix C is defined as

$$C = \begin{bmatrix} c_{1,1} & c_{1,2} & \ldots & c_{1,g_Y} & 0 & \ldots & 0 \\ c_{2,1} & c_{2,2} & \ldots & c_{2,g_Y} & 0 & \ldots & 0 \\ \vdots & & & & & & \\ c_{g_X,1} & c_{g_X,2} & \ldots & c_{g_X,g_Y} & 0 & \ldots & 0 \\ 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & & & & & & \\ 0 & 0 & \ldots & 0 & 0 & \ldots & 0 \end{bmatrix}_{p_X \times p_Y}$$

As with the univariate case, the $g_X \times g_Y$ upper-left corner of the convolution $K * C$ is the matrix of the estimates $\hat{f}(grid)$.

Most of the results in this subsection are found in Wand (1994).

Fast Fourier Transform

As shown in the last subsection, kernel density estimates can be expressed as a submatrix of a certain convolution. The fast Fourier transform (FFT) is a computationally effective method for computing such convolutions. For a reference on this material, see Press et al. (1988).

The *discrete Fourier transform* of a complex vector $\mathbf{z} = (z_0, \ldots, z_{N-1})$ is the vector $\mathbf{Z} = (Z_0, \ldots, Z_{N-1})$, where

$$Z_j = \sum_{l=0}^{N-1} z_l e^{2\pi i l j/N}, \quad j = 0, \ldots, N-1$$

and i is the square root of -1. The vector $\mathbf{z}$ can be recovered from $\mathbf{Z}$ by applying the *inverse discrete Fourier transform* formula

$$z_l = N^{-1} \sum_{j=0}^{N-1} Z_j e^{-2\pi i l j/N}, \quad l = 0, \ldots, N-1$$

Discrete Fourier transforms and their inverses can be computed quickly using the FFT algorithm, especially when N is *highly composite*; that is, it can be decomposed into many factors, such as a power of 2. By the *discrete convolution theorem*, the convolution of two vectors is the inverse Fourier transform of the element-by-element product of their Fourier transforms. This, however, requires certain periodicity assumptions, which explains why the vectors K and C require zero-padding. This is to avoid *wrap-around* effects (refer to Press et al. 1988, pp. 410–411). The vector K is actually mirror-imaged so that the convolution of C and K will be the vector of binned estimates. Thus, if S denotes the inverse Fourier transform of the element-by-element product of the Fourier transforms of K and C, then the first g elements of S are the estimates.

The bivariate Fourier transform of an $N_1 \times N_2$ complex matrix having $(l_1 + 1, l_2 + 1)$ entry equal to $z_{l_1 l_2}$ is the $N_1 \times N_2$ matrix with $(j_1 + 1, j_2 + 1)$ entry given by

$$Z_{j_1 j_2} = \sum_{l_1=0}^{N_1-1} \sum_{l_2=0}^{N_2-1} z_{l_1 l_2} e^{2\pi i (l_1 j_1/N_1 + l_2 j_2/N_2)}$$

and the formula of the inverse is

$$z_{l_1 l_2} = (N_1 N_2)^{-1} \sum_{j_1=0}^{N_1-1} \sum_{j_2=0}^{N_2-1} Z_{j_1 j_2} e^{-2\pi i (l_1 j_1/N_1 + l_2 j_2/N_2)}$$

The same discrete convolution theorem applies, and zero-padding is needed for matrices C and K. In the case of K, the matrix is mirror-imaged twice. Thus, if S denotes the inverse Fourier transform of the element-by-element product of the Fourier transforms of K and C, then the upper-left $g_X \times g_Y$ corner of S contains the estimates.

Bandwidth Selection

Several different bandwidth selection methods are available in PROC KDE in the univariate case. Following the recommendations of Jones, Marron, and Sheather (1996), the default method follows a plug-in formula of Sheather and Jones.

This method solves the fixed-point equation

$$h = \left[\frac{R(\varphi)}{nR\left(\hat{f}''_{g(h)}\right)\left(\int x^2\varphi(x)dx\right)^2} \right]^{1/5}$$

where $R(\varphi) = \int \varphi^2(x)dx$.

PROC KDE solves this equation by first evaluating it on a grid of values spaced equally on a log scale. The largest two values from this grid that bound a solution are then used as starting values for a bisection algorithm.

The simple normal reference rule works by assuming $\hat{f}$ is Gaussian in the preceding fixed-point equation. This results in

$$\begin{aligned} h &= \hat{\sigma}[4/(3n)]^{1/5} \\ &= 1.06\,\hat{\sigma}n^{-1/5} \end{aligned}$$

where $\hat{\sigma}$ is the sample standard deviation.

Alternatively, the bandwidth can be computed using the interquartile range, Q:

$$\begin{aligned} h &= 1.06\,\hat{\sigma}n^{-1/5} \\ &\approx 1.06\,(Q/1.34)n^{-1/5} \\ &\approx 0.785\,Qn^{-1/5} \end{aligned}$$

Silverman's rule of thumb (Silverman 1986, Section 3.4.2) is computed as

$$h = 0.9\min[\hat{\sigma}, Q/1.34]n^{-1/5}$$

The oversmoothed bandwidth is computed as

$$h = 3\hat{\sigma}[1/(70\sqrt{\pi}n)]^{1/5}$$

When you specify a WEIGHT variable, PROC KDE uses weighted versions of Q_3, Q_1, and $\hat{\sigma}$ in the preceding expressions. The weighted quartiles are computed as weighted order statistics, and the weighted variance takes the form

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} W_i(X_i - \bar{X})^2}{\sum_{i=1}^{n} W_i}$$

where $\bar{X} = (\sum_{i=1}^{n} W_i X_i)/(\sum_{i=1}^{n} W_i)$ is the weighted sample mean.

For the bivariate case, Wand and Jones (1993) note that automatic bandwidth selection is both difficult and computationally expensive. Their study of various ways of specifying a bandwidth matrix also shows that using two bandwidths, one in each coordinate's direction, is often adequate. PROC KDE enables you to adjust the two bandwidths by specifying a multiplier for the default bandwidths recommended by Bowman and Foster (1993):

$$\begin{aligned} h_X &= \hat{\sigma}_X n^{-1/6} \\ h_Y &= \hat{\sigma}_Y n^{-1/6} \end{aligned}$$

Here $\hat{\sigma}_X$ and $\hat{\sigma}_Y$ are the sample standard deviations of X and Y, respectively. These are the optimal bandwidths for two independent normal variables that have the same variances as X and Y. They are, therefore, conservative in the sense that they tend to oversmooth the surface.

You can specify the BWM= option to adjust the aforementioned bandwidths to provide the appropriate amount of smoothing for your application.

ODS Table Names

PROC KDE assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 45.1. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 45.1 ODS Tables Produced in PROC KDE

ODS Table Name	Description	Statement	Option
BivariateStatistics	Bivariate statistics	BIVAR	BIVSTATS
Controls	Control variables	default	
Inputs	Input information	default	
Levels	Levels of density estimate	BIVAR	LEVELS
Percentiles	Percentiles of data	BIVAR / UNIVAR	PERCENTILES
UnivariateStatistics	Basic statistics	BIVAR / UNIVAR	UNISTATS

ODS Graphics

This section describes the use of ODS for creating graphics with the KDE procedure. To request these graphs, you must specify the ODS GRAPHICS statement in addition to the following options. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Bivariate Plots

You can specify the PLOTS= option in the BIVAR statement to request graphical displays of bivariate kernel density estimates.

PLOTS= *option1 < option2 . . . >*

requests one or more plots of the bivariate kernel density estimate. The following table shows the available plot *options*.

Option	Description
ALL	all available displays
CONTOUR	contour plot of bivariate density estimate
CONTOURSCATTER	contour plot of bivariate density estimate overlaid with scatter plot of data
HISTOGRAM	bivariate histogram of data
HISTSURFACE	bivariate histogram overlaid with bivariate kernel density estimate
NONE	suppresses all plots
SCATTER	scatter plot of data
SURFACE	surface plot of bivariate kernel density estimate

By default, if you enable ODS Graphics and you do not specify the PLOTS= option, then the BIVAR statement creates a contour plot. If you specify the PLOTS= option, you get only the requested plots.

Univariate Plots

You can specify the PLOTS= option in the UNIVAR statement to request graphical displays of univariate kernel density estimates.

PLOTS= *option1 < option2 . . . >*

requests one or more plots of the univariate kernel density estimate. The following table shows the available plot *options*.

Option	Description
ALL	all available displays
DENSITY	univariate kernel density estimate curve
DENSITYOVERLAY	overlaid univariate kernel density estimate curves
HISTDENSITY	univariate histogram of data overlaid with kernel density estimate curve
HISTOGRAM	univariate histogram of data
NONE	suppresses all plots

By default, if you enable ODS Graphics and you do not specify the PLOTS= option, then the UNIVAR statement creates a histogram overlaid with a kernel density estimate. If you specify the PLOTS= option, you get only the requested plots.

ODS Graph Names

PROC KDE assigns a name to each graph it creates using the Output Delivery System (ODS). You can use these names to reference the graphs when using ODS. The names are listed in Table 45.2.

To request these graphs you must specify the ODS GRAPHICS statement in addition to the options indicated in Table 45.2. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Table 45.2 ODS Graphics Produced by PROC KDE

ODS Graph Name	Plot Description	Statement	PLOTS= Option
BivariateHistogram	Bivariate histogram of data	BIVAR	HISTOGRAM
ContourPlot	Contour plot of bivariate kernel density estimate	BIVAR	CONTOUR
ContourScatterPlot	Contour plot of bivariate kernel density estimate overlaid with scatter plot	BIVAR	CONTOURSCATTER
DensityPlot	Univariate kernel density estimate curve	UNIVAR	DENSITY
DensityOverlayPlot	Overlaid univariate kernel density estimate curves	UNIVAR	DENSITYOVERLAY
HistogramDensity	Univariate histogram overlaid with kernel density estimate curve	UNIVAR	HISTDENSITY
Histogram	Univariate histogram of data	UNIVAR	HISTOGRAM
HistogramSurface	Bivariate histogram overlaid with surface plot of bivariate kernel density estimate	BIVAR	HISTSURFACE
ScatterPlot	Scatter plot of data	BIVAR	SCATTER
SurfacePlot	Surface plot of bivariate kernel density estimate	BIVAR	SURFACE

Binning of Bivariate Histogram

Let (X_i, Y_i), $i = 1, 2, \ldots, n$, be a sample of size n drawn from a bivariate distribution. For the marginal distribution of X_i, $i = 1, 2, \ldots, n$, the number of bins (Nbins_X) in the bivariate histogram is calculated according to the formula

$$\text{Nbins}_X = \text{ceil}\left(\text{range}_X / \text{width}_X\right)$$

where $\text{ceil}(x)$ denotes the smallest integer greater than or equal to x,

$$\text{range}_X = \max_{1 \le i \le n}(X_i) - \min_{1 \le i \le n}(X_i)$$

and the optimal bin width is obtained, following Scott (1992, p. 84), as

$$\text{width}_X = 3.504\,\hat{\sigma}_X(1-\hat{\rho}^2)^{3/8}n^{-1/4}$$

Here, $\hat{\sigma}_X$ and $\hat{\rho}$ are the sample variance and the sample correlation coefficient, respectively. When you specify a WEIGHT variable, PROC KDE uses weighted versions of $\hat{\sigma}_X$ and $\hat{\rho}$ in the preceding expressions.

Similar formulas are used to compute the number of bins for the marginal distribution of Y_i, $i = 1, 2, \ldots, n$. Further details can be found in Scott (1992).

Notice that if $|\hat{\rho}| > 0.99$, then Nbins_X is calculated as in the univariate case (see Terrell and Scott 1985). In this case $\text{Nbins}_Y = \text{Nbins}_X$.

Examples: KDE Procedure

Example 45.1: Computing a Basic Kernel Density Estimate

This example illustrates the basic functionality of the UNIVAR statement. The effective channel length (in microns) is measured for 1225 field effect transistors. The channel lengths are saved as values of the variable length in a SAS data set named channel; refer to the file *kdex1.sas* in the SAS Sample Library. These statements create the channel data set:

```
data channel;
   input length @@;
datalines;
0.91 1.01 0.95 1.13 1.12 0.86 0.96 1.17 1.36 1.10
0.98 1.27 1.13 0.92 1.15 1.26 1.14 0.88 1.03 1.00
0.98 0.94 1.09 0.92 1.10 0.95 1.05 1.05 1.11 1.15

... more lines ...

2.13 2.05 1.90 2.07 2.15 1.96 2.15 1.89 2.15 2.04
1.95 1.93 2.22 1.74 1.91
;
```

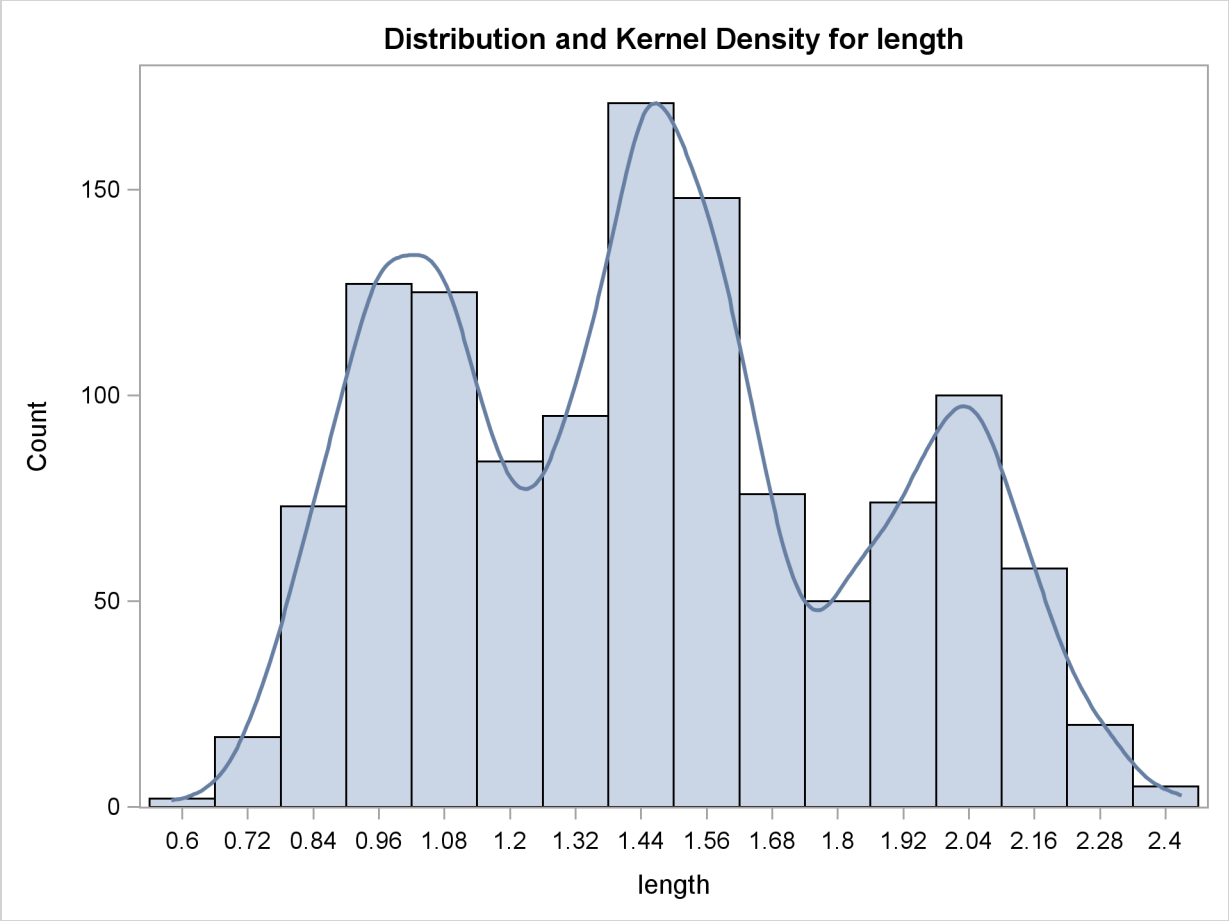
The following statements request a kernel density estimate of the variable length:

```
ods graphics on;
proc kde data=channel;
   univar length;
run;
```

Because ODS Graphics is enabled, PROC KDE produces a histogram with an overlaid kernel density estimate by default, although the PLOTS= option is not specified. The resulting graph is shown

in Output 45.1.1. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the KDE procedure, see the section "ODS Graphics" on page 2904.

Output 45.1.1 Histogram with Overlaid Kernel Density Estimate



The default output tables for this analysis are the "Inputs" and "Controls" tables, shown in Output 45.1.2.

Output 45.1.2 Univariate Inputs Table

The KDE Procedure

Inputs

Data Set	WORK.CHANNEL
Number of Observations Used	1225
Variable	length
Bandwidth Method	Sheather-Jones Plug In

Output 45.1.2 *continued*

Controls

	length
Grid Points	401
Lower Grid Limit	0.58
Upper Grid Limit	2.43
Bandwidth Multiplier	1

The "Inputs" table lists basic information about the density fit, including the input data set, the number of observations, the variable used, and the bandwidth method. The default bandwidth method is the Sheather-Jones plug-in.

The "Controls" table lists the primary numbers controlling the kernel density fit. Here the default number of grid points is used and no adjustment is made to the default bandwidth.

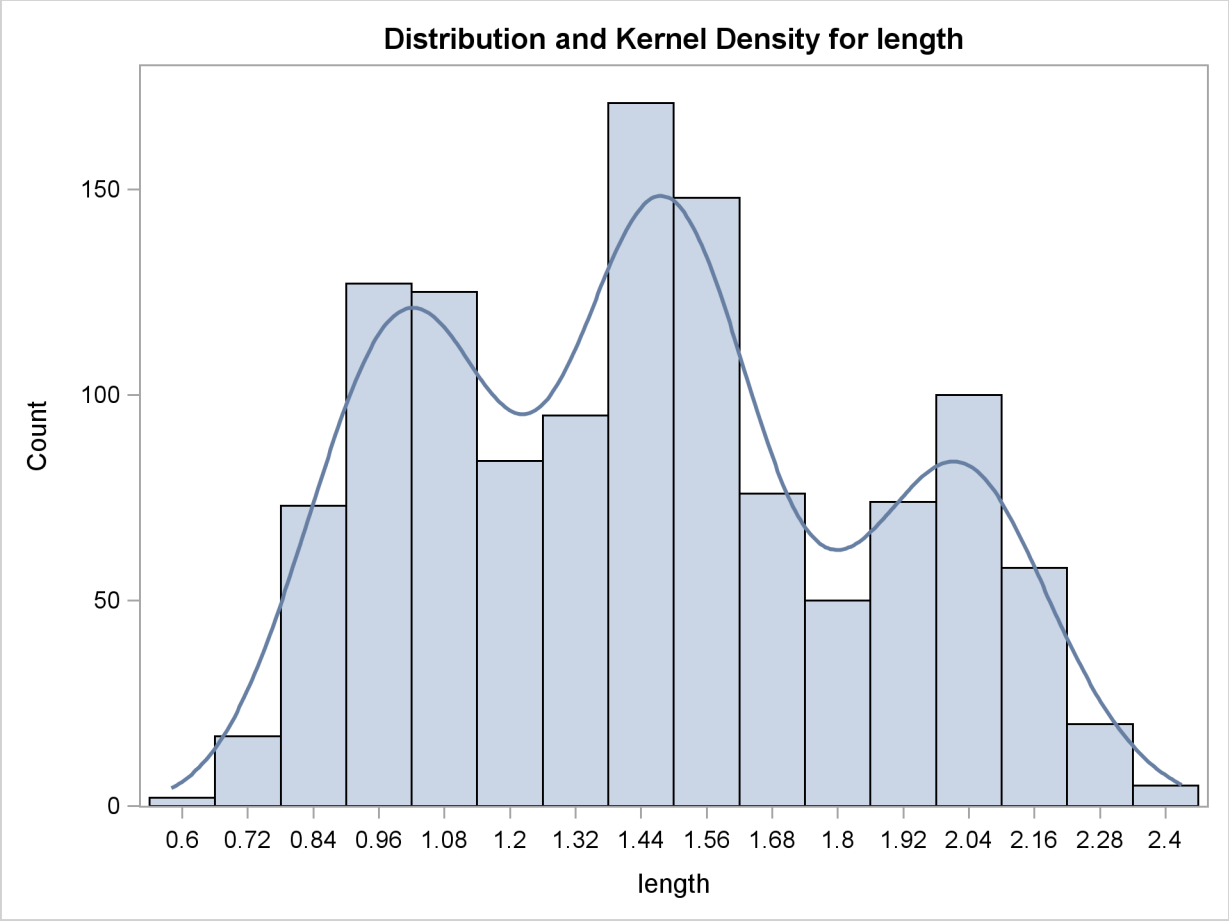
Example 45.2: Changing the Bandwidth

Continuing with Example 45.1, you can specify different bandwidth multipliers that determine the smoothness of the kernel density estimate. The following statements show kernel density estimates for the variable length by specifying two different bandwidth multipliers with the BWM= option:

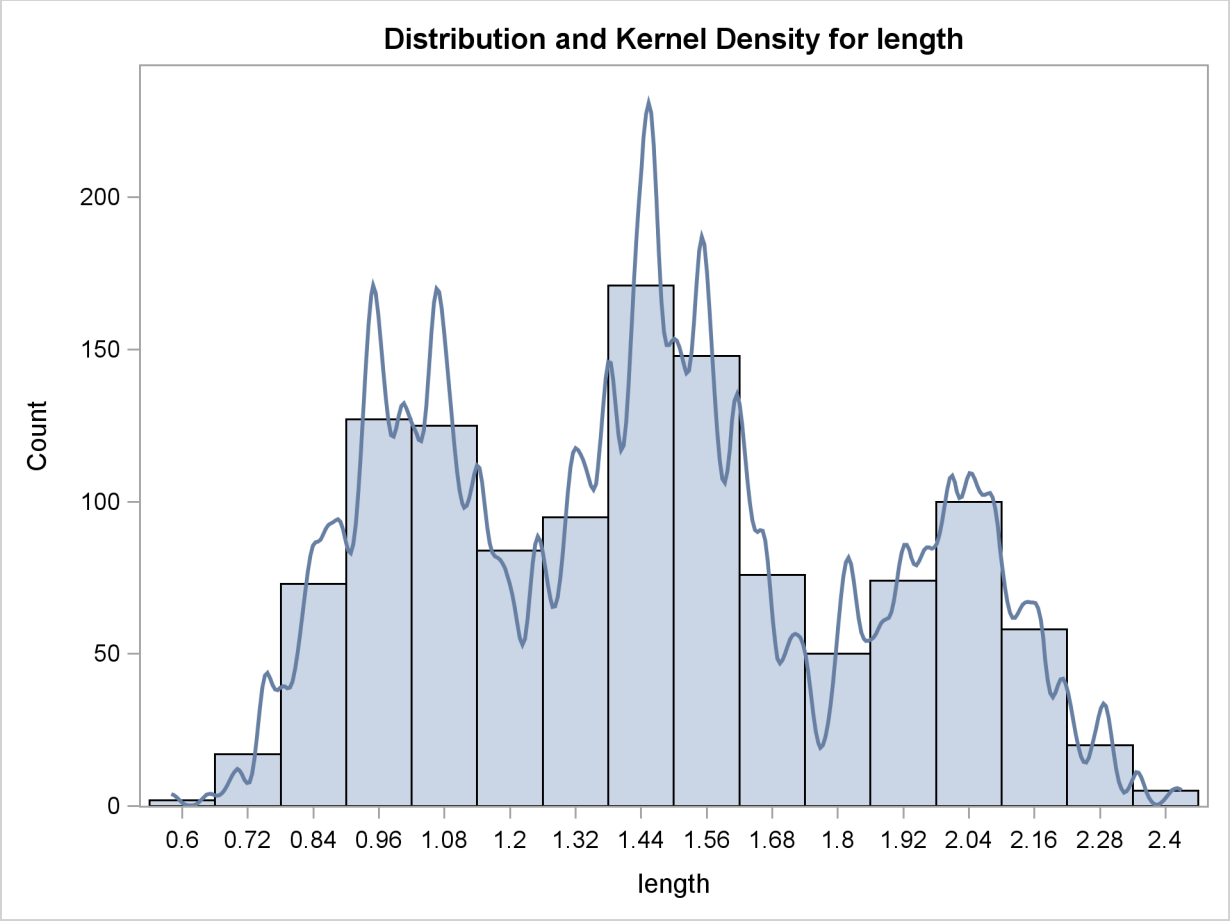
```
proc kde data=channel;
   univar length(bwm=2) length(bwm=0.25);
run;
ods graphics off;
```

Output 45.2.1 shows an oversmoothed estimate because the bandwidth multiplier is 2. Output 45.2.2 is created by specifying BWM=0.25, so it is an undersmoothed estimate.

Output 45.2.1 Histogram with Oversmoothed Kernel Density Estimate



Output 45.2.2 Histogram with Undersmoothed Kernel Density Estimate



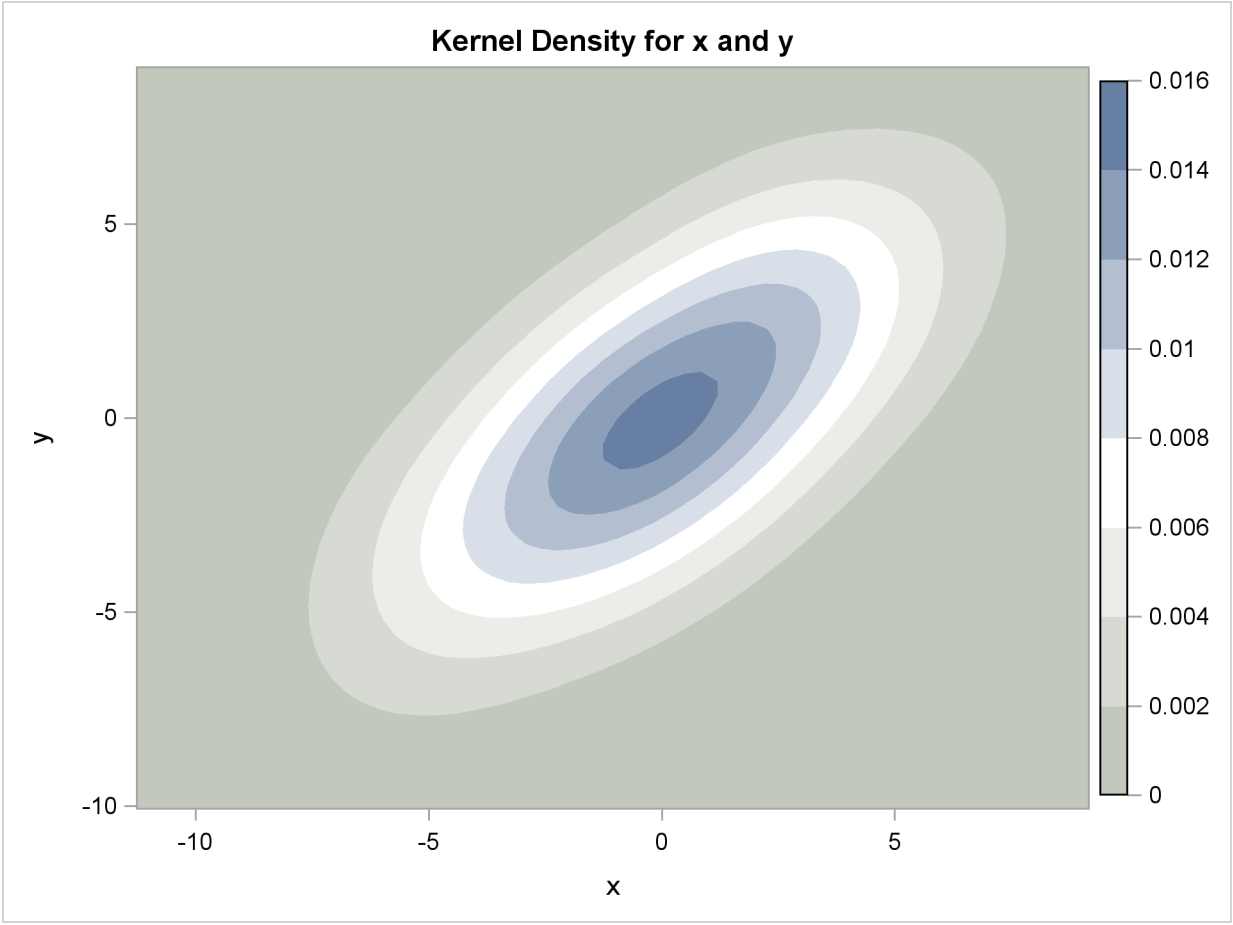
Example 45.3: Changing the Bandwidth (Bivariate)

Recall the analysis from the section "Getting Started: KDE Procedure" on page 2886. Suppose you would like a slightly smoother estimate. You could then rerun the analysis with a larger bandwidth:

```
ods graphics on;
proc kde data=bivnormal;
   bivar x y / bwm=2;
run;
```

The BWM= option requests bandwidth multipliers of 2 for both x and y. With ODS Graphics enabled, the BIVAR statement produces a contour plot, as shown in Output 45.3.1.

Output 45.3.1 Contour Plot of Estimated Density with Additional Smoothing



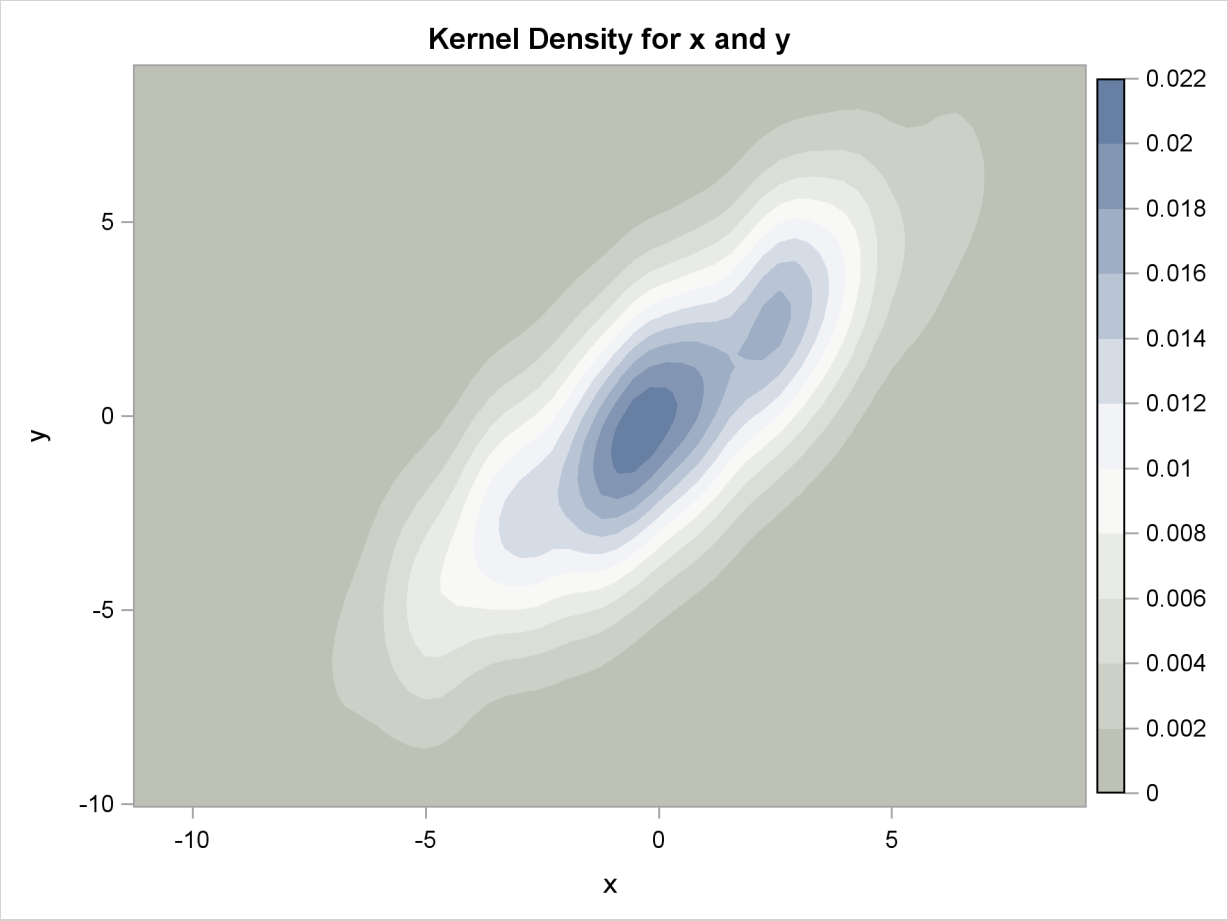
Multiple Bandwidths

You can also specify multiple bandwidths with only one run of the KDE procedure. Notice that by specifying pairs of variables inside parentheses, a kernel density estimate is computed for each pair. In the following statements the first kernel density is computed with the default bandwidth, but the second kernel density specifies a bandwidth multiplier of 0.5 for the variable x and a multiplier of 2 for the variable y:

```
proc kde data=bivnormal;
   bivar (x y) (x (bwm=0.5) y (bwm=2));
run;
ods graphics off;
```

The contour plot of the second kernel density estimate is shown in Output 45.3.2.

Output 45.3.2 Contour Plot of Estimated Density with Different Smoothing for x and y



Example 45.4: Requesting Additional Output Tables

This example illustrates how to request output tables with summary statistics in addition to the default output tables. Using the same data as in the section "Getting Started: KDE Procedure" on page 2886, the following statements request univariate and bivariate summary statistics, percentiles, and levels of the kernel density estimate:

```
proc kde data=bivnormal;
   bivar x y / bivstats levels percentiles unistats;
run;
```

The resulting output is shown in Output 45.4.1.

Output 45.4.1 Bivariate Kernel Density Estimate Tables

The KDE Procedure

Inputs

Data Set	WORK.BIVNORMAL
Number of Observations Used	1000
Variable 1	x
Variable 2	y
Bandwidth Method	Simple Normal Reference

Controls

	x	y
Grid Points	60	60
Lower Grid Limit	-11.25	-10.05
Upper Grid Limit	9.1436	9.0341
Bandwidth Multiplier	1	1

Univariate Statistics

	x	y
Mean	-0.075	-0.070
Variance	9.73	9.93
Standard Deviation	3.12	3.15
Range	20.39	19.09
Interquartile Range	4.46	4.51
Bandwidth	0.99	1.00

Bivariate Statistics

Covariance	8.88
Correlation	0.90

Percentiles

	x	y
0.5	-7.71	-8.44
1.0	-7.08	-7.46
2.5	-6.17	-6.31
5.0	-5.28	-5.23
10.0	-4.18	-4.11
25.0	-2.24	-2.30
50.0	-0.11	-0.058
75.0	2.22	2.21
90.0	3.81	3.94
95.0	4.88	5.22
97.5	6.03	5.94
99.0	6.90	6.77
99.5	7.71	7.07

Output 45.4.1 *continued*

		Levels			
Percent	Density	Lower for x	Upper for x	Lower for y	Upper for y
1	0.001181	−8.14	8.45	−8.76	8.39
5	0.003031	−7.10	7.07	−7.14	6.77
10	0.004989	−6.41	5.69	−6.49	6.12
50	0.01591	−3.64	3.96	−3.58	3.86
90	0.02388	−1.22	1.19	−1.32	0.95
95	0.02525	−0.88	0.50	−0.99	0.62
99	0.02608	−0.53	0.16	−0.67	0.30
100	0.02629	−0.19	−0.19	−0.35	−0.35

The "Univariate Statistics" table contains standard univariate statistics for each variable, as well as statistics associated with the density estimate. Note that the estimated variances for both x and y are fairly close to the true values of 10.

The "Bivariate Statistics" table lists the covariance and correlation between the two variables. Note that the estimated correlation is equal to its true value to two decimal places.

The "Percentiles" table lists percentiles for each variable.

The "Levels" table lists contours of the density corresponding to percentiles of the bivariate data, and the minimum and maximum values of each variable on those contours. For example, 5% of the observed data have a density value less than 0.0030. The minimum x and y values on this contour are −7.10 and −7.14, respectively (the Lower for x and Lower for y columns), and the maximum values are 7.07 and 6.77, respectively (the Upper for x and Upper for y columns).

You can also request "Percentiles" or "Levels" tables with specific percentiles:

```
proc kde data=bivnormal;
   bivar x y / levels=2.5, 50, 97.5
               percentiles=2.5, 25, 50, 75, 97.5;
run;
```

The resulting "Percentiles" and "Levels" tables are shown in Output 45.4.2.

Output 45.4.2 Customized Percentiles and Levels Tables

The KDE Procedure

Percentiles

	x	y
2.5	−6.17	−6.31
25.0	−2.24	−2.30
50.0	−0.11	−0.058
75.0	2.22	2.21
97.5	6.03	5.94

Output 45.4.2 *continued*

		Levels			
Percent	Density	Lower for x	Upper for x	Lower for y	Upper for y
2.5	0.001914	−7.79	8.11	−7.79	7.74
50.0	0.01591	−3.64	3.96	−3.58	3.86
97.5	0.02573	−0.88	0.50	−0.99	0.30

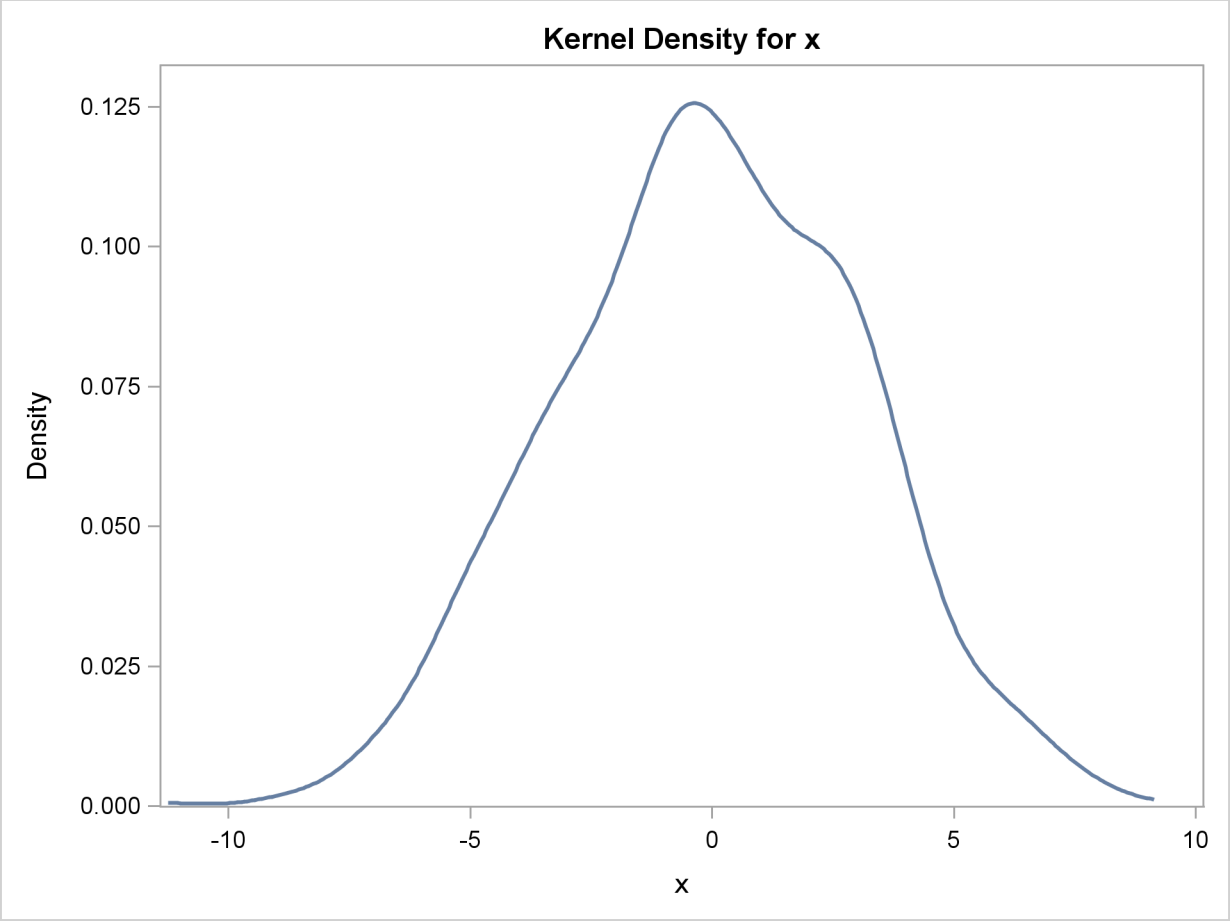
Example 45.5: Univariate KDE Graphics

This example uses data from the section "Getting Started: KDE Procedure" to illustrate the use of ODS Graphics. The following statements request the available univariate plots in PROC KDE:

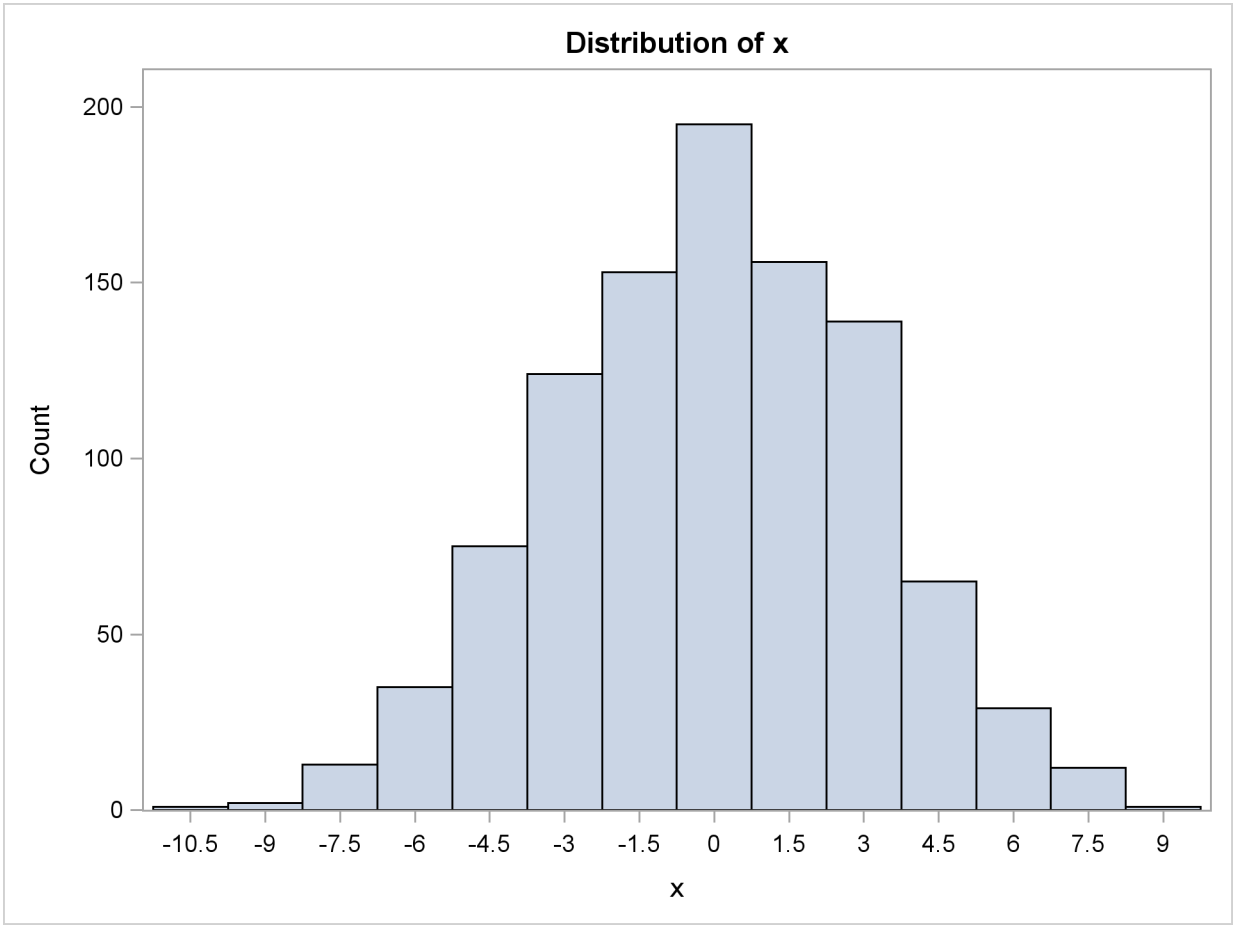
```
ods graphics on;
proc kde data=bivnormal;
   univar x / plots=(density histogram histdensity);
   univar x y / plots=densityoverlay;
run;
ods graphics off;
```

Graphs are requested by specifying the ODS GRAPHICS statement and the PLOTS= option in the UNIVAR statement. Output 45.5.1, Output 45.5.2, and Output 45.5.3 show the kernel density estimate, histogram, and histogram with kernel density estimate overlaid, respectively, produced by the first UNIVAR statement.

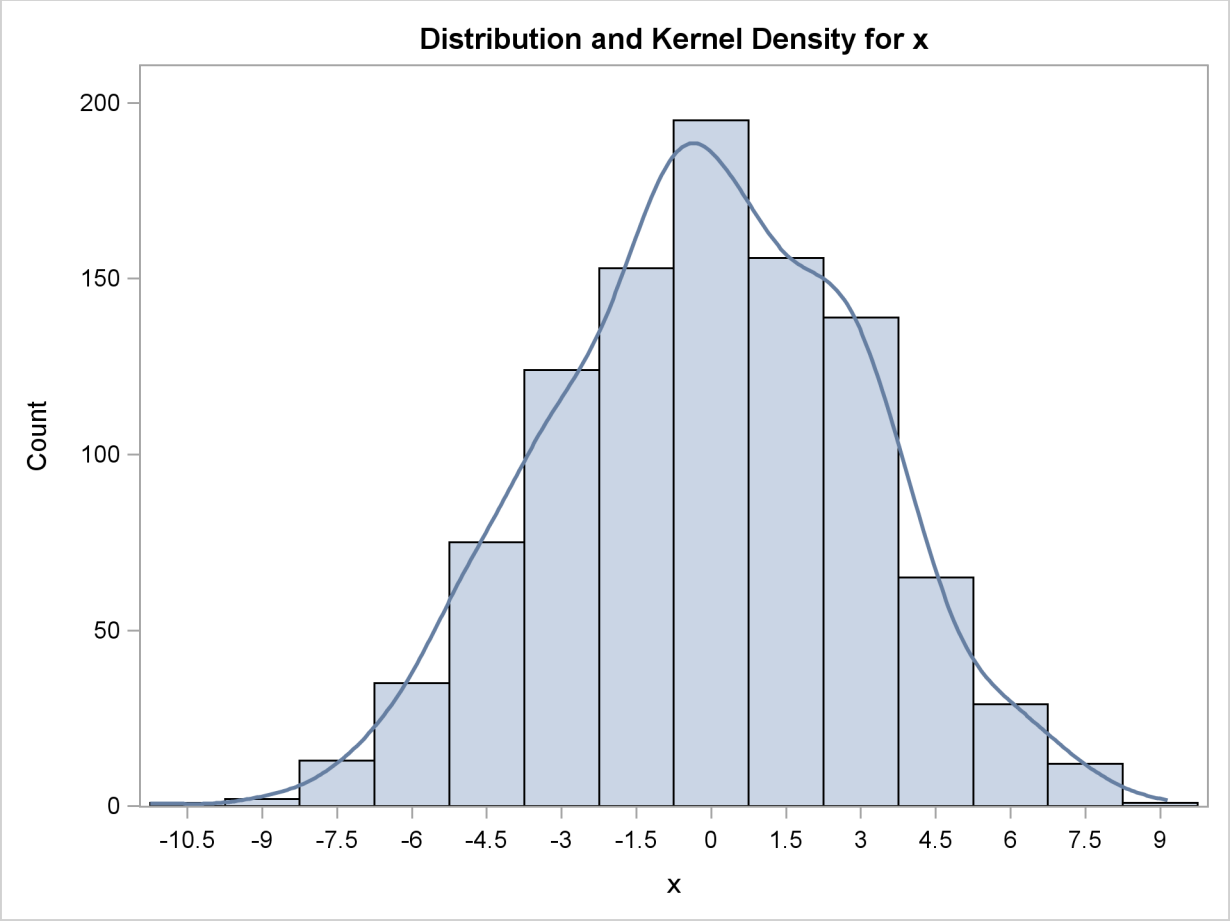
Output 45.5.1 Kernel Density Estimate



Output 45.5.2 Histogram

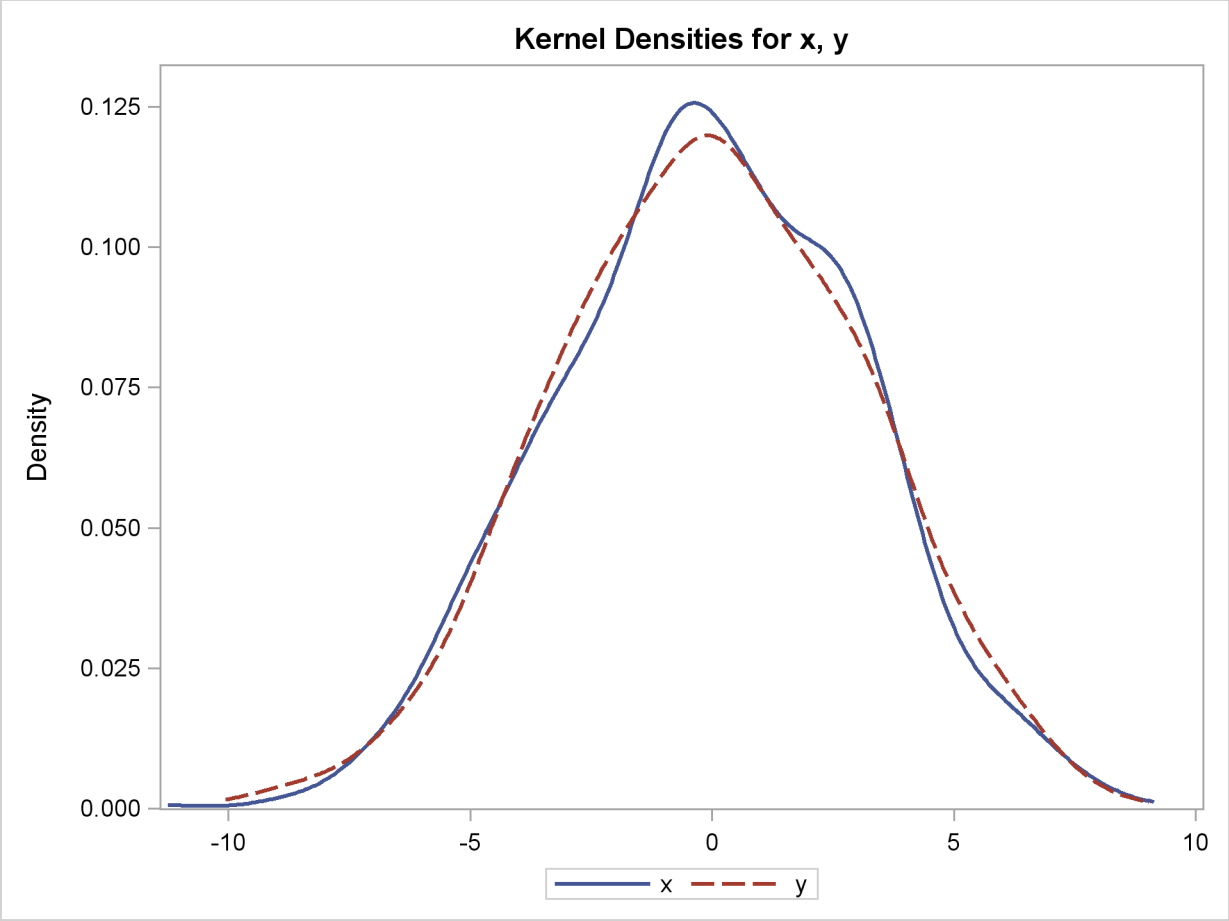


Output 45.5.3 Histogram with Overlaid Kernel Density Estimate



Output 45.5.4 shows the plot produced by the second UNIVAR statement, in which the kernel density estimates for x and y are overlaid.

Output 45.5.4 Overlaid Kernel Density Estimates



For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the KDE procedure, see the section "ODS Graphics" on page 2904.

Example 45.6: Bivariate KDE Graphics

This example illustrates the available bivariate graphics in PROC KDE. The octane data set comes from Rodriguez and Taniguchi (1980), where it is used for predicting customer octane satisfaction by using trained-rater observations. The variables in this data set are Rater and Customer. Either variable might have missing values. Refer to the file *kdex3.sas* in the SAS Sample Library. The following statements create the octane data set:

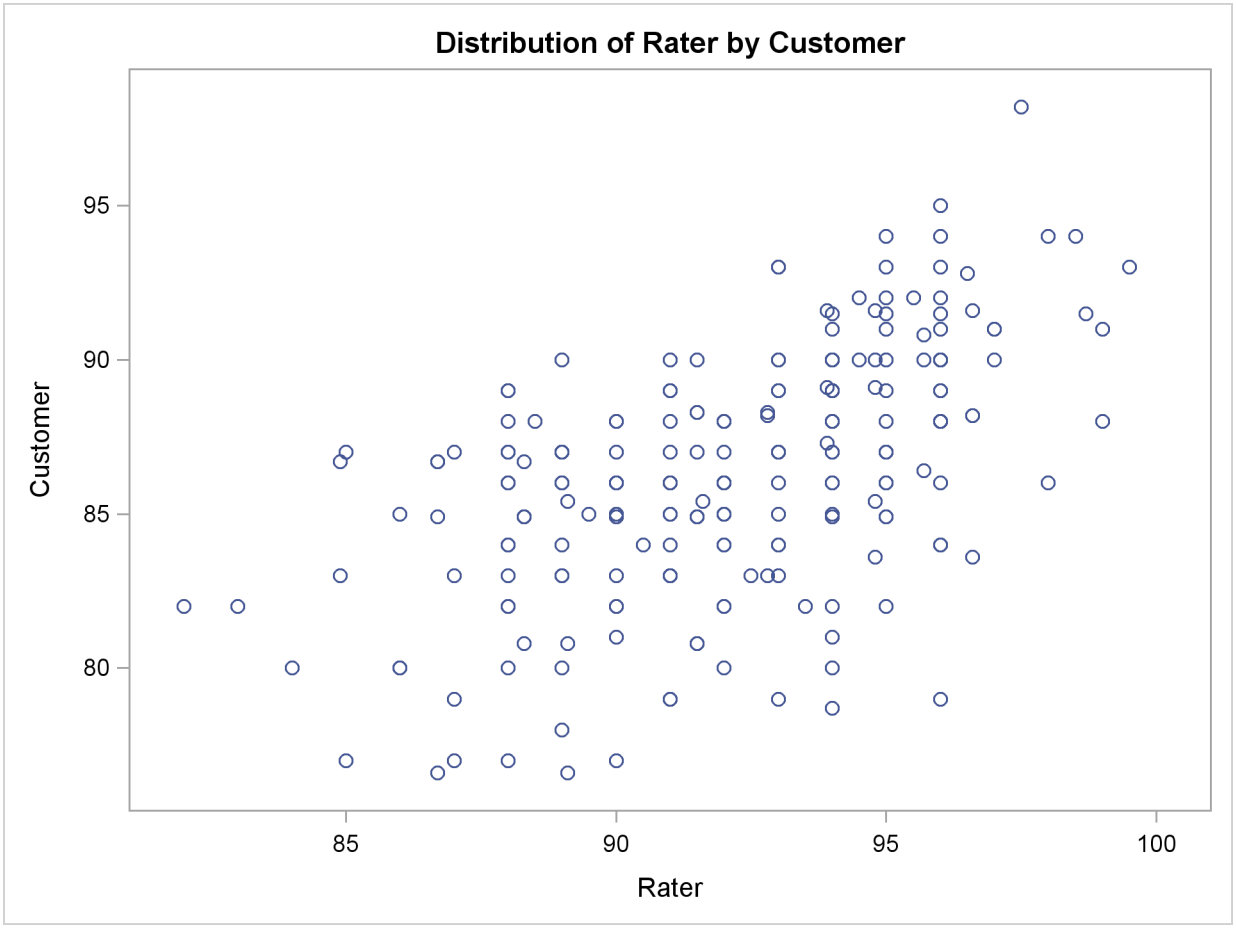
```
data octane;
   input Rater Customer;
   label Rater    = 'Rater'
         Customer = 'Customer';
datalines;
94.5 92.0
94.0 88.0
94.0 90.0

... more lines ...

88.0 84.0
  .H 90.0
;
```

The following statements request all the available bivariate plots in PROC KDE:

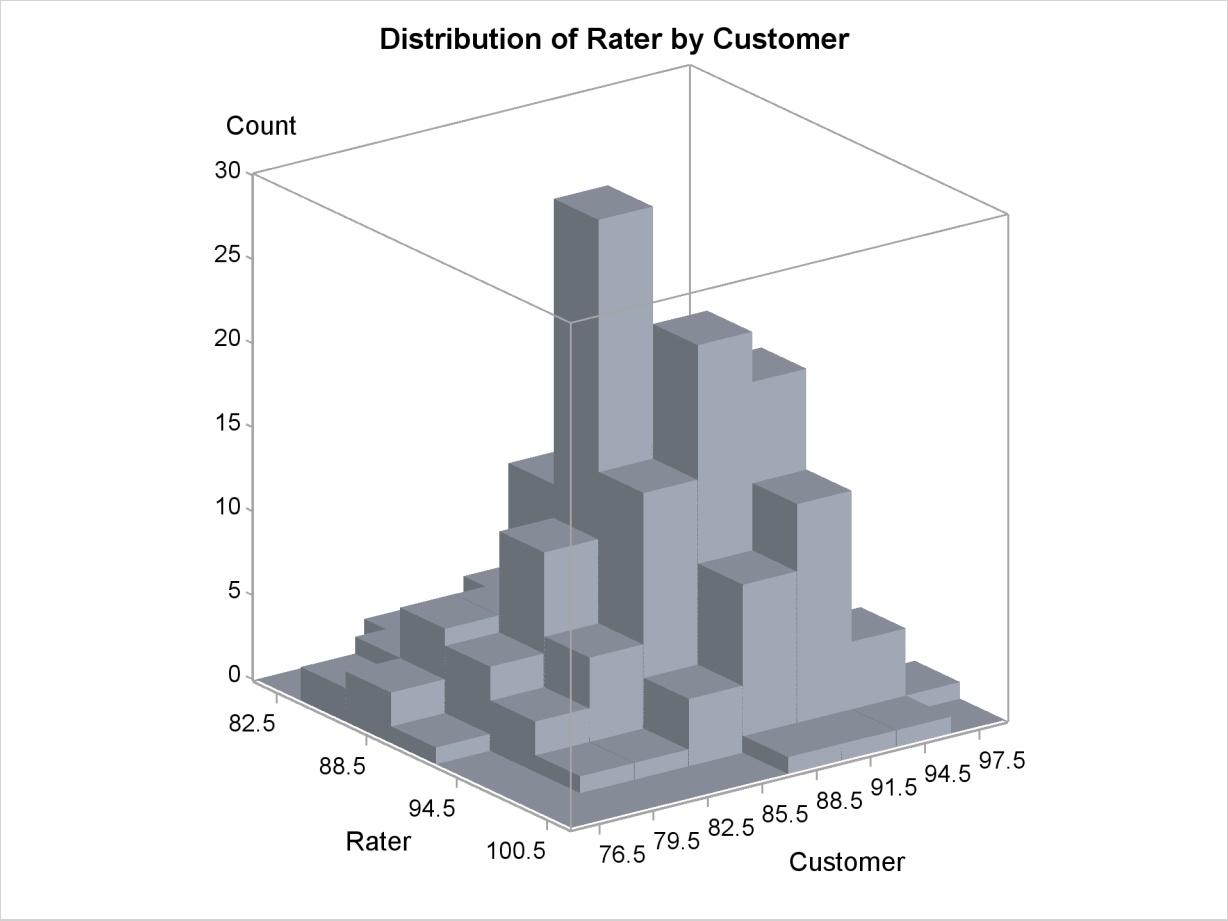
```
ods graphics on;
proc kde data=octane;
   bivar Rater Customer / plots=all;
run;
ods graphics off;
```

Output 45.6.1 shows a scatter plot of the data, Output 45.6.2 shows a bivariate histogram of the data, Output 45.6.3 shows a contour plot of bivariate density estimate, Output 45.6.4 shows a contour plot of bivariate density estimate overlaid with a scatter plot of data, Output 45.6.5 shows a surface plot of bivariate kernel density estimate, and Output 45.6.6 shows a bivariate histogram overlaid with a bivariate kernel density estimate. These graphical displays are requested by specifying the ODS GRAPHICS statement and the PLOTS= option in the BIVAR statement. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the KDE procedure, see the section "ODS Graphics" on page 2904.

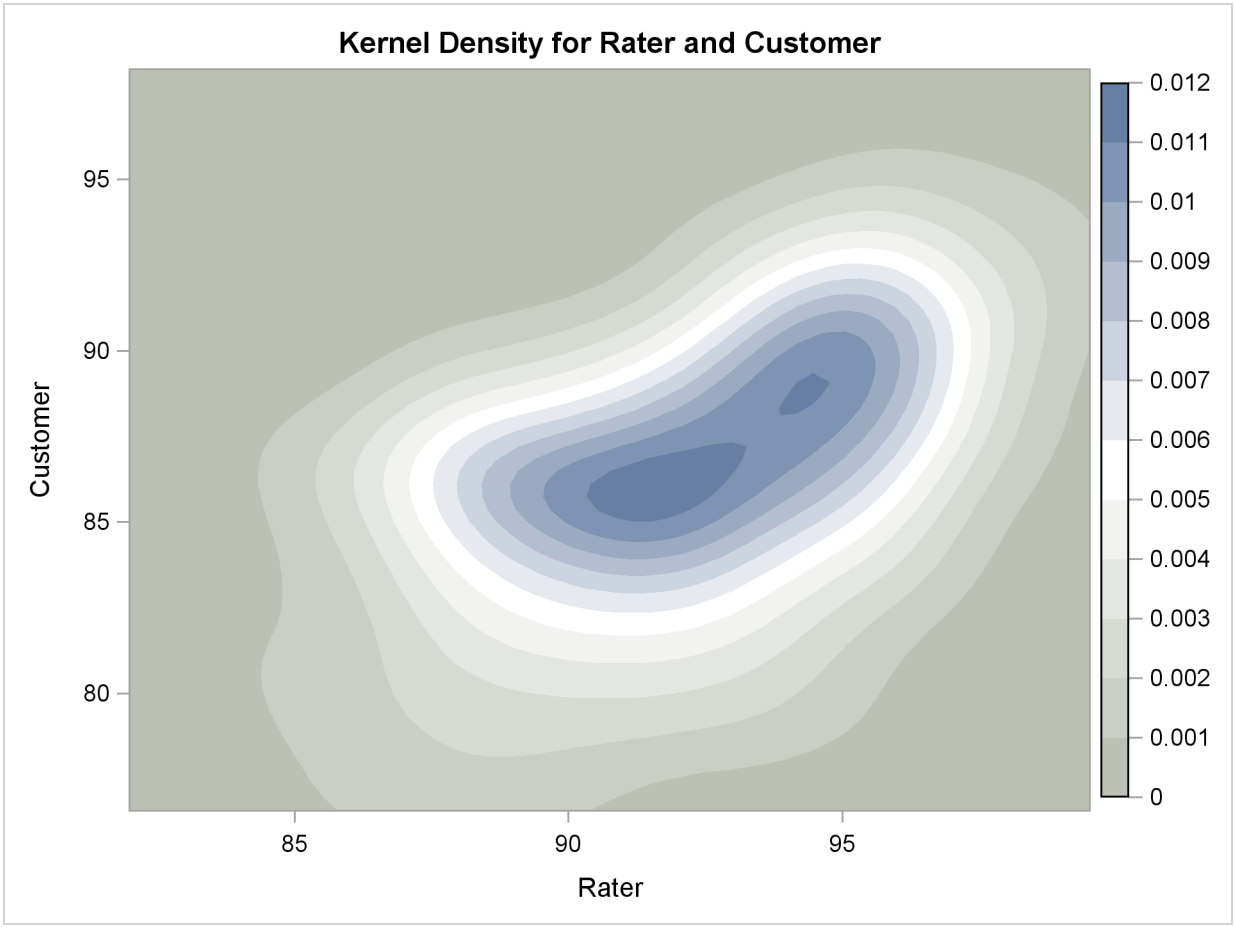
Output 45.6.1 Scatter Plot



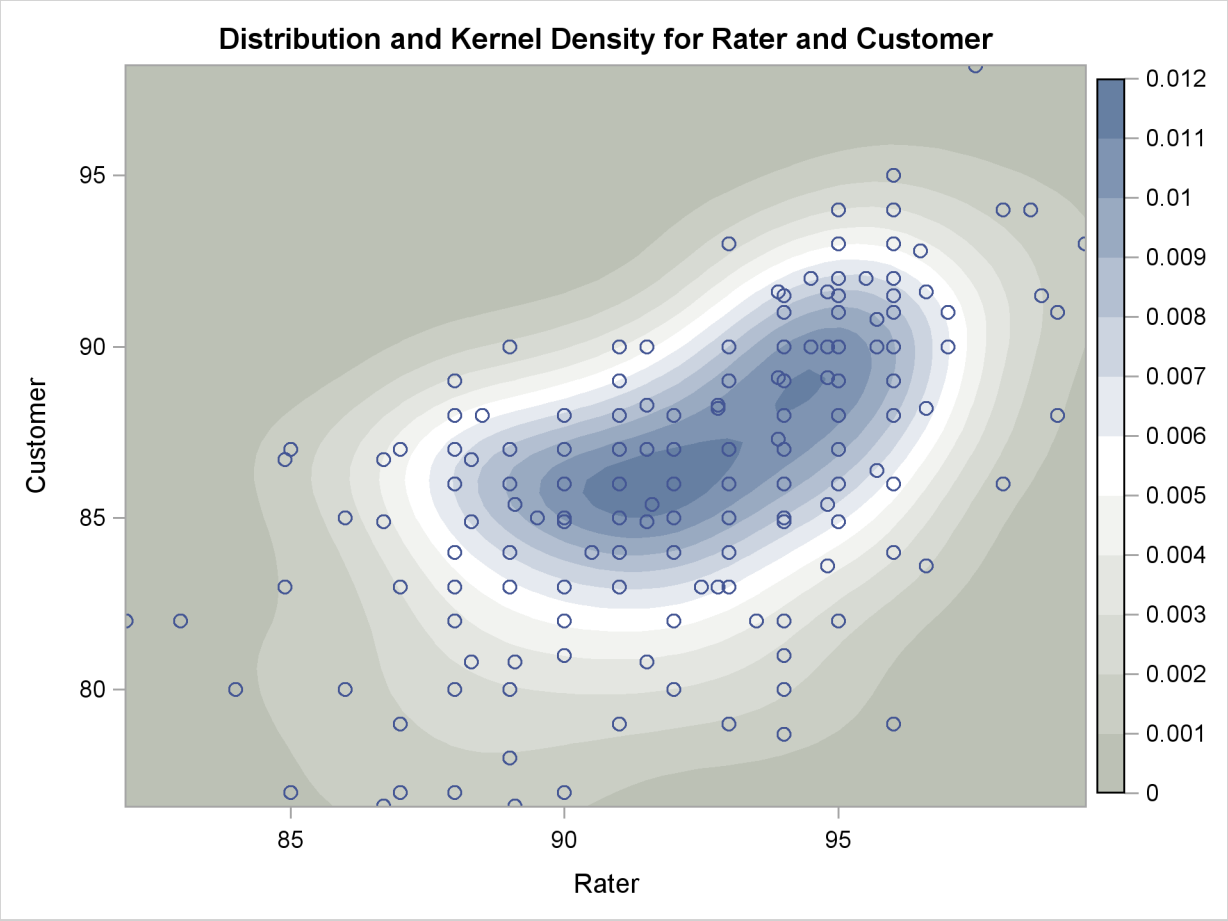
Output 45.6.2 Bivariate Histogram



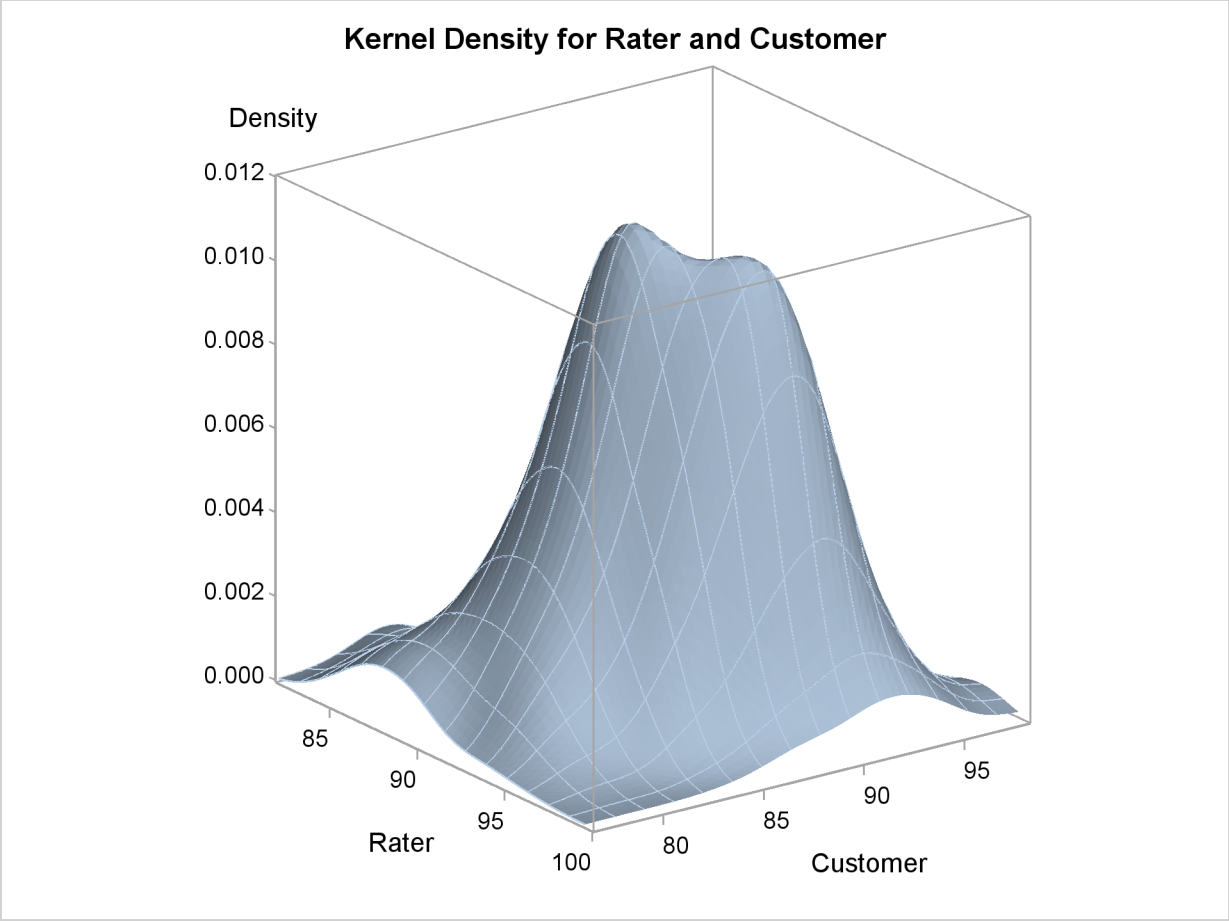
Output 45.6.3 Contour Plot



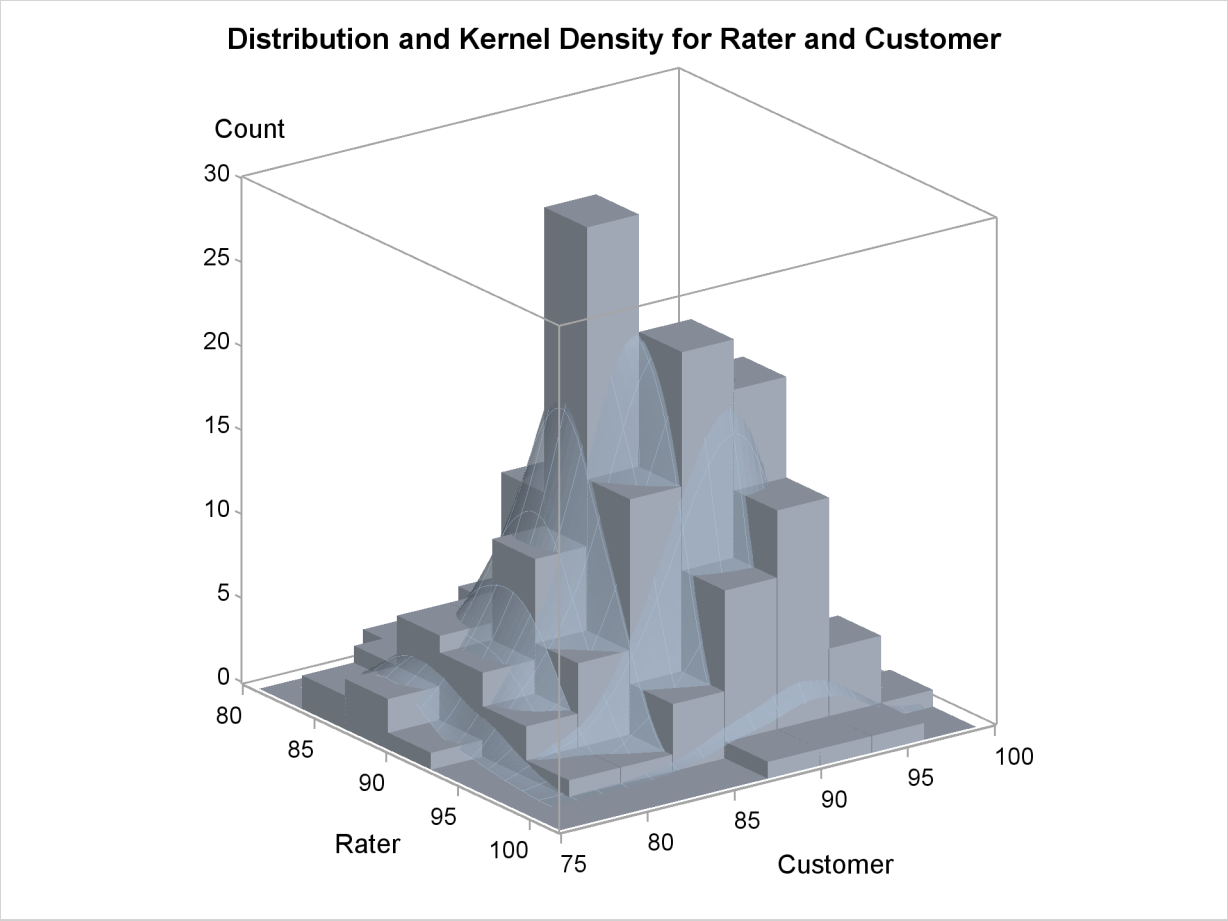
Output 45.6.4 Contour Plot with Overlaid Scatter Plot



Output 45.6.5 Surface Plot



Output 45.6.6 Bivariate Histogram with Overlaid Surface Plot



References

Bowman, A. W. and Foster, P. J. (1993), "Density Based Exploration of Bivariate Data," *Statistics and Computing*, 3, 171–177.

Fan, J. and Marron, J. S. (1994), "Fast Implementations of Nonparametric Curve Estimators," *Journal of Computational and Graphical Statistics*, 3, 35–56.

Jones, M. C., Marron, J. S., and Sheather, S. J. (1996), "A Brief Survey of Bandwidth Selection for Density Estimation," *Journal of the American Statistical Association*, 91, 401–407.

Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1988), *Numerical Recipes: The Art of Scientific Computing*, Cambridge: Cambridge University Press.

Rodriguez, R. N. and Taniguchi, B. Y. (1980), "A New Statistical Model for Predicting Customer Octane Satisfaction Using Trained-Rater Observations," *Transactions of the Society of Automotive Engineers*, 4213–4235.

Scott, D. W. (1992), *Multivariate Density Estimation: Theory, Practice, and Visualization*, New York: John Wiley & Sons.

Silverman, B. W. (1986), *Density Estimation*, New York: Chapman & Hall.

Terrell, G. R. and Scott, D. W. (1985), "Oversmoothed Nonparametric Density Estimates," *Journal of the American Statistical Association*, 80, 209–214.

Wand, M. P. (1994), "Fast Computation of Multivariate Kernel Estimators," *Journal of Computational and Graphical Statistics*, 3, 433–445.

Wand, M. P. and Jones, M. C. (1993), "Comparison of Smoothing Parameterizations in Bivariate Kernel Density Estimation," *Journal of the American Statistical Association*, 88, 520–528.

Chapter 46

The KRIGE2D Procedure

Contents

Overview: KRIGE2D Procedure **2930**
Introduction to Spatial Prediction 2930
Getting Started: KRIGE2D Procedure **2931**
Spatial Prediction Using Kriging, Contour Plots 2931
Syntax: KRIGE2D Procedure **2936**
PROC KRIGE2D Statement 2937
BY Statement 2940
COORDINATES Statement 2941
GRID Statement 2942
PREDICT Statement 2943
MODEL Statement 2944
Details: KRIGE2D Procedure **2947**
Theoretical Semivariogram Models 2947
The Nugget Effect 2954
Anisotropic Models 2955
Geometric Anisotropy 2956
Zonal Anisotropy 2959
Anisotropic Nugget Effect 2962
Details of Ordinary Kriging 2962
Introduction 2962
Spatial Random Fields 2963
Ordinary Kriging 2964
Computational Resources 2966
Output Data Sets 2967
Displayed Output 2968
ODS Table Names 2969
ODS Graphics 2969
Examples: KRIGE2D Procedure **2970**
Example 46.1: Investigating the Effect of Model Specification on Prediction 2970
Example 46.2: Data Quality and Prediction with Missing Values 2973
References **2977**

Overview: KRIGE2D Procedure

The KRIGE2D procedure performs ordinary kriging in two dimensions. PROC KRIGE2D can handle anisotropic and nested semivariogram models. Four semivariogram models are supported: the Gaussian, exponential, spherical, and power models. A single nugget effect is also supported.

You can specify the locations of kriging predictions in a GRID statement, or they can be read from a SAS data set. The grid specification is most suitable for a regular grid; the data set specification can handle any irregular pattern of points.

Local kriging is supported through the specification of a radius around a grid point or the specification of the number of nearest neighbors to use in the kriging system. When you perform local kriging, a separate kriging system is solved at each grid point by using a neighborhood of the data point established by the radius or number specification.

The KRIGE2D procedure writes the kriging predictions and associated standard errors for each grid to an output data set. When you perform local kriging, PROC KRIGE2D writes the neighborhood information for each grid point to an additional, optional data set. The KRIGE2D procedure does not produce any displayed output.

The KRIGE2D procedure now uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For more information about the graphics available in PROC KRIGE2D, see the section "ODS Graphics" on page 2969.

Introduction to Spatial Prediction

Many activities in science and technology involve measurements of one or more quantities at given spatial locations, with the goal of predicting the measured quantities at unsampled locations. Application areas include reservoir prediction in mining and petroleum exploration, as well as modeling in a broad spectrum of fields (for example, environmental health, environmental pollution, natural resources and energy, hydrology, risk analysis). Often, the unsampled locations are on a regular grid, and the predictions are used to produce surface plots or contour maps.

The preceding tasks fall within the scope of *spatial prediction*, which, in general, is any prediction method that incorporates spatial dependence. The study of these tasks involves naturally occurring uncertainties that cannot be ignored. Stochastic analysis frameworks and methods are used to account for these uncertainties. Hence, the terms *stochastic spatial prediction* and *stochastic modeling* are also used to characterize this type of analysis.

A popular method of spatial prediction is *ordinary kriging*, which produces both predicted values and associated standard errors. Ordinary kriging requires the complete specification (the form and parameter values) of the spatial dependence characterizing the spatial process. For this purpose, models for the spatial dependence are expressed in terms of the distance between any two locations in the spatial domain of interest. These models take the form of a covariance or semivariance function.

Spatial prediction, then, involves two steps. First, you model the covariance or semivariance of the spatial process. These measures are typically not known in advance. This step involves computing an empirical estimate, as well as determining both the mathematical form and the values of any parameters for a theoretical form of the dependence model. Second, you use this dependence model to solve the kriging system at a specified set of spatial points, resulting in predicted values and associated standard errors.

The KRIGE2D procedure performs the second of these steps by using ordinary kriging of two-dimensional data.

This introduction concludes with a note on terminology. You might commonly encounter the terms *estimation* and *prediction* used interchangeably by experts in different fields; this could be a source of confusion. A precise statistical vernacular uses the term *estimation* to refer to inferences about the value of fixed but unknown parameters, whereas *prediction* concerns inferences about the value of random variables—see, for example, Cressie (1993, p. 106). In light of these definitions, kriging methods are clearly predictive techniques, since they are concerned with making inferences about the value of a spatial random field at observed or unobserved locations. The SAS/STAT suite of procedures for spatial analysis and prediction (VARIOGRAM, KRIGE2D, and SIM2D) follows the statistical vernacular in the use of the terms *estimation* and *prediction*.

Getting Started: KRIGE2D Procedure

Spatial Prediction Using Kriging, Contour Plots

After an appropriate semivariogram model is chosen, there are a number of choices involved in producing the kriging surface. In order to illustrate these choices, you use the theoretical semivariogram model that was selected for the coal seam thickness data in "Example 95.1: Theoretical Semivariogram Model Fitting" on page 7570 in the VARIOGRAM procedure. This model is Gaussian,

$$\gamma_z(h) = c_0\left[1 - \exp\left(-\frac{h^2}{a_0^2}\right)\right]$$

with a scale of $c_0 = 7.2881$ (that is, the model sill) and a range of $a_0 = 30.6239$, based on the weighted least squares fitting results in the PROC VARIOGRAM example.

The first choice is whether to use local or global kriging. Local kriging uses only data points in the neighborhood of a grid point, and you choose this type of analysis by specifying a data search radius around the grid point. Global kriging uses all data points.

The most important consideration in this decision is the spatial covariance structure. Global kriging is appropriate when the correlation range ϵ is approximately equal to the length of the spatial domain. The correlation range ϵ is the distance r_ϵ (also known as *effective* or *practical* range, too) at which the covariance is 5% of its value at zero. That is,

$$C_Z(r_\epsilon) = 0.05C_z(0)$$

For a Gaussian model, r_ϵ is $\sqrt{3}a_0 \approx 52{,}000$ feet. The data points are scattered uniformly throughout a 100×100 (10^6 ft^2) area. Hence, the linear dimension of the data is nearly double the r_ϵ range. This indicates that local kriging rather than global kriging is appropriate because data that are farther away than r_ϵ essentially add to the computational burden without significant contribution to the prediction. The following DATA step is used to input the thickness data set:

```
data thick;
   input East North Thick @@;
   label Thick='Coal Seam Thickness';
   datalines;
    0.7  59.6  34.1   2.1  82.7  42.2   4.7  75.1  39.5
    4.8  52.8  34.3   5.9  67.1  37.0   6.0  35.7  35.9
    6.4  33.7  36.4   7.0  46.7  34.6   8.2  40.1  35.4
   13.3   0.6  44.7  13.3  68.2  37.8  13.4  31.3  37.8
   17.8   6.9  43.9  20.1  66.3  37.7  22.7  87.6  42.8
   23.0  93.9  43.6  24.3  73.0  39.3  24.8  15.1  42.3
   24.8  26.3  39.7  26.4  58.0  36.9  26.9  65.0  37.8
   27.7  83.3  41.8  27.9  90.8  43.3  29.1  47.9  36.7
   29.5  89.4  43.0  30.1   6.1  43.6  30.8  12.1  42.8
   32.7  40.2  37.5  34.8   8.1  43.3  35.3  32.0  38.8
   37.0  70.3  39.2  38.2  77.9  40.7  38.9  23.3  40.5
   39.4  82.5  41.4  43.0   4.7  43.3  43.7   7.6  43.1
   46.4  84.1  41.5  46.7  10.6  42.6  49.9  22.1  40.7
   51.0  88.8  42.0  52.8  68.9  39.3  52.9  32.7  39.2
   55.5  92.9  42.2  56.0   1.6  42.7  60.6  75.2  40.1
   62.1  26.6  40.1  63.0  12.7  41.8  69.0  75.6  40.1
   70.5  83.7  40.9  70.9  11.0  41.7  71.5  29.5  39.8
   78.1  45.5  38.7  78.2   9.1  41.7  78.4  20.0  40.8
   80.5  55.9  38.7  81.1  51.0  38.6  83.8   7.9  41.6
   84.5  11.0  41.5  85.2  67.3  39.4  85.5  73.0  39.8
   86.7  70.4  39.6  87.2  55.7  38.8  88.1   0.0  41.6
   88.4  12.1  41.3  88.4  99.6  41.2  88.8  82.9  40.5
   88.9   6.2  41.5  90.6   7.0  41.5  90.7  49.6  38.9
   91.5  55.4  39.0  92.9  46.8  39.1  93.4  70.9  39.7
   55.8  50.5  38.1  96.2  84.3  40.3  98.2  58.2  39.5
   ;
```

Local kriging is performed by using only data points within a specified radius of each grid point. In this example, a radius of 60,000 feet is used. Other choices involved in local kriging are the minimum and maximum number of data points in each neighborhood (around a grid point). The minimum number is left at the default value of 20; the maximum number defaults to all observations in the data set within the specified radius.

The last step in contouring the data is to define the prediction grid point (node) locations. The prediction grid is typically rectangular, and you decide on the grid points population and spacing based on your available data as well as your application needs. A convenient area that encompasses all the data points is a square of side length 100,000 feet. In the present analysis, a distance of 2,500 feet between nodes in the prediction grid is selected to obtain a smooth contour plot. Based on this choice, you will obtain predictions on a square grid with 41 nodes on each side, which yields a total of 1681 grid points.

You can visualize the outcome of your analysis by using the PLOTS= option in the PROC KRIGE2D statement. PROC KRIGE2D will produce by default one plot that displays the kriging prediction and its corresponding standard error at each output grid point. The locations of the Thick observations are displayed too, as outlines in the default plot. You can also ask for a plot of the thick data set observations and their values by specifying the OBSERV option in the PLOTS option.

The kriging analysis with the KRIGE2D procedure requires that you provide the prediction parameters in the PREDICT statement. You use the VAR= option to specify that you will be using the Thick variable in the kriging system, and the RADIUS= option to specify the radius of the local kriging regression. In this scenario you want to consider for your predictions all the neighboring data within a radius of 60,000 feet from each prediction location. You can specify more than one PREDICT statements; for example, you can do this when you want predictions for different variables in your DATA= data set.

The coordinates of your variable are specified in the COORDINATES statement. The MODEL statement contains the parameters that describe your data spatial correlation. Namely, the FORM= option specifies the model type, based on its mathematical form. The SCALE= and RANGE= options specify the model sill and range, respectively. You can specify more than one MODEL statement for the same PREDICT statement in order to obtain predictions based on different correlation models.

When you use the RADIUS= option to perform local kriging, as in the present example, it is suggested that the radius parameter is at least as large as your model range, so that you do not exclude data points that can contribute to your prediction.

Eventually, you specify the region of predictions with the GRID statement. The following SAS statements compute the kriged surface by using the preceding options and grid choice:

```
title 'Coal Seam Thickness Kriging';
ods graphics on;

proc krige2d data=thick;
   coordinates xc=East yc=North;
   predict var=Thick radius=60;
   model scale=7.2881 range=30.6239 form=gauss;
   grid x=0 to 100 by 2.5 y=0 to 100 by 2.5;
run;

ods graphics off;
```

The table in Figure 46.1 shows the number of observations read and used in the kriging prediction. This table provides you with useful information in case you have missing values in the input data.

Figure 46.1 Number of Observations for the thick Data Set

```
                    Coal Seam Thickness Kriging

                       The KRIGE2D Procedure
                     Dependent Variable: Thick

              Number of Observations Read          75
              Number of Observations Used          75
```

Figure 46.2 shows some general information about the kriging analysis. This includes the count of the output grid points. You have specified the RADIUS= option; therefore you also see that local kriging is requested. Because this is a local analysis, the table also displays the parameters related to the neighborhood search around the grid points.

Figure 46.2 Kriging Analysis Information

```
                         Kriging Information

          Prediction Grid Points                     1681
          Type of Analysis                          Local
          Neighborhood Search Radius                   60
          Minimum Neighbors                            20
          Maximum Neighbors             All Within Radius
```

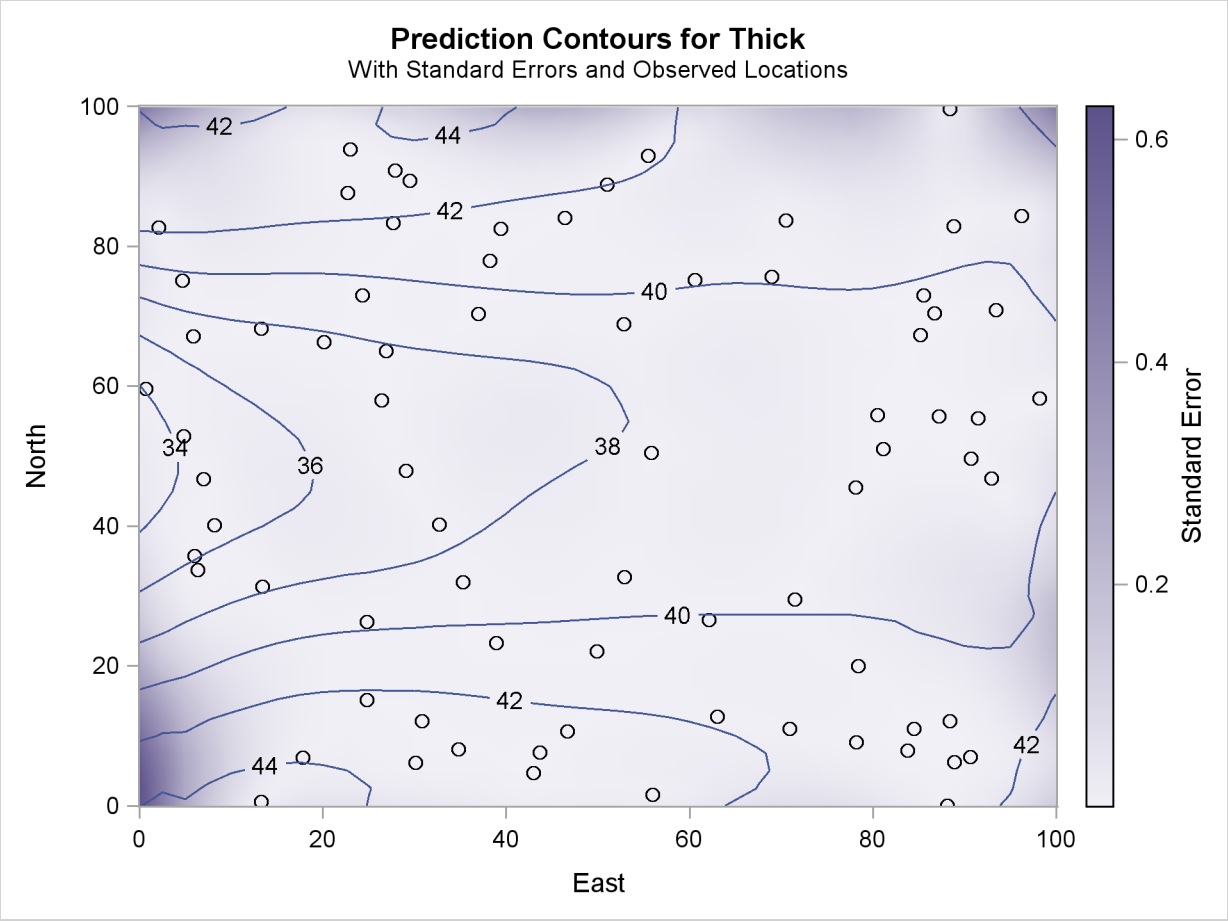
The covariance model parameters, including the effective range of the Gaussian model you specified, are shown in Figure 46.3.

Figure 46.3 Kriging Covariance Model Information

```
                    Coal Seam Thickness Kriging

                       The KRIGE2D Procedure
                     Dependent Variable: Thick
                  Prediction: Pred1, Model: Model1

                    Covariance Model Information

                    Type                Gaussian
                    Sill                  7.2881
                    Range                30.6239
                    Effective Range    53.042151
                    Nugget Effect              0
```

Figure 46.4 shows a map of the kriging prediction contours based on the Thick observations in the specified spatial domain. The prediction error is displayed as a surface in the background.

Figure 46.4 Contour Plot of Kriged Coal Seam Thickness



Note the locations of the observed data in Figure 46.4. The figure suggests that the Thick sampling locations are not ideally spread around the prediction area; however, there are no extended areas lacking measurements.

Based on the spatial distribution of the Thick data and the range r_ϵ of your covariance model, you can roughly see that for each prediction location there are at least several neighboring data points that will contribute to the prediction value. Except perhaps for the nodes close to the boundaries of the prediction grid, you can then expect the prediction errors to be reasonably low compared to the predicted Thick values.

The kriging outcome in Figure 46.4 indicates that the standard errors are smaller in the neighborhoods where data are available. The size of these neighborhoods depends on the range of the specified covariance model that characterizes the spatial continuity of the domain, and the prediction radius, if one is specified as in this example. The standard errors tend to increase toward the borders of the prediction area, beyond which no observations are available.

Syntax: KRIGE2D Procedure

The following statements are available in PROC KRIGE2D:

PROC KRIGE2D *options* ;
BY *variables* ;
COORDINATES | **COORD** *coordinate-variables* ;
GRID *grid-options* ;
PREDICT | **PRED** | **P** *predict-options* ;
MODEL *model-options* ;

The PREDICT and MODEL statements are hierarchical; the PREDICT statement is followed by a MODEL statement. If more than one MODEL statement is given, only the last one is used for the analysis. The MODEL statement following a PREDICT statement uses the variable and neighborhood specifications in that PREDICT statement.

You must specify at least one PREDICT statement and one MODEL statement. You must supply a single COORDINATES statement to identify the x and y coordinate variables in the input data set. You must also specify a single GRID statement to include the grid information.

The following table outlines the options available in PROC KRIGE2D classified by function.

Table 46.1 Options Available in the KRIGE2D Procedure

Task	Statement	Option
Data Set Options		
Specify input data set	PROC KRIGE2D	DATA=
Specify grid data set	GRID	GDATA=
Specify model data set	MODEL	MDATA=
Write kriging predictions and standard errors	PROC KRIGE2D	OUTEST=
Write neighborhood information for each grid point	PROC KRIGE2D	OUTNBHD=
Specify plot display and options	PROC KRIGE2D	PLOTS=
Declaring the Role of Variables		
Specify variables to define analysis subgroups	BY	
Specify the variables to be predicted (kriged)	PREDICT	VAR=
Specify the x and y coordinate variables in the DATA= data set	COORDINATES	XC= YC=
Specify the x and y coordinate variables in the GDATA= data set	GRID	XC= YC=
Controlling Kriging Neighborhoods		
Specify the radius of a neighborhood for all grid points	PREDICT	RADIUS=
Specify the number of neighbors for all grid points	PREDICT	NUMPOINTS=
Specify the maximum of neighbors for all grid points	PREDICT	MAXPOINTS=
Specify the minimum of neighbors for all grid points	PREDICT	MINPOINTS=
Specify action when maximum not met	PREDICT	NODECREMENT

Table 46.1 *continued*

Task	Statement	Option
Specify action when minimum not met	PREDICT	NOINCREMENT
Controlling the Semivariogram Model		
Specify a nugget effect	MODEL	NUGGET=
Specify a type with a functional form	MODEL	FORM=
Specify a range parameter	MODEL	RANGE=
Specify a scale parameter	MODEL	SCALE=
Specify an angle for an anisotropic model	MODEL	ANGLE=
Specify a minor-major axis ratio for an anisotropic model	MODEL	RATIO=

PROC KRIGE2D Statement

PROC KRIGE2D *options* **;**

You can specify the following options in the PROC KRIGE2D statement.

DATA=*SAS-data-set*
specifies a SAS data set containing the x and y coordinate variables and the VAR= variables in the PREDICT statement.

NOPRINT
suppresses the normal display of results. The NOPRINT option is useful when you want only to create one or more output data sets with the procedure. Note that this option temporarily disables the Output Delivery System (ODS); see the section "ODS Graphics" on page 2969 for more information.

OUTEST=*SAS-data-set*

OUTE=*SAS-data-set*
specifies a SAS data set in which to store the kriging predictions, standard errors, and grid location. For details, see the section "OUTEST=*SAS-data-set*" on page 2967.

OUTNBHD=*SAS-data-set*

OUTN=*SAS-data-set*
specifies a SAS data set in which to store the neighborhood information for each grid point. Information is written to this data set only if one or more PREDICT statements have options specifying local kriging. For details, see the section "OUTNBHD=*SAS-data-set*" on page 2967.

PLOTS <(*global-plot-option***)> <= ***plot-request* *< (options) >* **>**

PLOTS <(*global-plot-option***)> <= (***plot-request* *< (options) >* **<...** *plot-request* *< (options) >* **>>)>**
controls the plots produced through ODS Graphics. When you specify only one plot request, you can omit the parentheses around the plot request. Here are some examples:

```
plots=none
plots=observ
plots=(observ(outl) prediction)
plots=(prediction(fill=pred line=se obs=grad) prediction(fill=se))
```

You must enable ODS Graphics before requesting plots, for example, like this:

```
ods graphics on;

proc krige2d data=thick;
   coordinates xc=East yc=North;
   predict var=thick r=60;
   model scale=7.2881 range=30.6239 form=gauss;
   grid x=0 to 100 by 10 y=0 to 100 by 10;
run;

ods graphics off;
```

For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." If you have enabled ODS Graphics but do not specify the PLOTS= option or have specified PLOTS=ALL, then PROC KRIGE2D produces a default plot for each MODEL statement of every PREDICT statement that you specify. The default PROC KRIGE2D plot displays a contour plot of the kriging prediction and the gradient of the kriging prediction standard error at every location of the prediction grid, as well as empty circles that indicate the observation locations. See Figure 46.4 for an example of the default KRIGE2D plot.

The following *global-plot-option* is available:

ONLY
suppresses the default plot. Only plots specifically requested are displayed.

The following individual *plot-requests* and plot *options* are available:

ALL
produces all appropriate plots. You can specify other options with ALL. For example, to request the default plot and an additional plot of the predictions, specify PLOTS=(ALL PREDICTION).

EQUATE
specifies that all appropriate plots be produced in a way that the axes coordinates have equal size units.

NONE
suppresses all plots.

OBSERVATIONS < (*observations-plot-options*) >
OBSERV < (*observations-plot-options*) >
OBS < (*observations-plot-options*) >
produces the observed data plot. Only one observations plot will be created if you specify the OBSERVATIONS option more than once within a PLOTS option.

The OBSERVATIONS option has the following suboptions:

GRADIENT
specifies that observations be displayed as circles colored by the observed measurement.

OUTLINE
specifies that observations be displayed as circles with a border but with a completely transparent fill.

OUTLINEGRADIENT
is the same as OBSERVATIONS(GRADIENT) except that a border is shown around each observation.

SHOWMISSING
specifies that observations with missing values be displayed in addition to the observations with nonmissing values. By default, missing values locations are not shown on the plot. If you specify multiple instances of the OBSERVATIONS option, and you specify the SHOWMISSING suboption in any of those, then the resulting observations plot will display the observations with missing values.

For the GRADIENT, OUTLINE, and OUTLINEGRADIENT suboptions: The OUTLINEGRADIENT is the default suboption if you do not specify any of those three. If you specify multiple instances of the OBSERVATIONS option or multiple suboptions for OBSERVATIONS, then the resulting observations plot will honor the last specified GRADIENT, OUTLINE, or OUTLINEGRADIENT suboption.

PREDICTION < (*prediction-plot-options*) >
PRED < (*prediction-plot-options*) >
specifies that the kriging prediction plot be produced. You can specify the PREDICTION option multiple times in the same PLOTS option to request instances of plots with the following *prediction-plot-options*:

FILL=NONE | PRED | SE
produces a surface plot for either the predicted values or the standard errors. FILL=SE is the default; however, if you do not specify the FILL suboption the behavior depends on the LINE suboption as follows: If you specify LINE=NONE or do not specify the LINE suboption at all, then the FILL suboption will be set to its default value. If LINE=PRED or LINE=SE, then the FILL suboption will be set to the same value as the LINE suboption.

LINE=NONE | PRED | SE
produces a contour line plot for either the predicted values or the standard errors. LINE=PRED is the default; however, if you do not specify the LINE suboption the behavior depends on the FILL suboption as follows: If you specify FILL=NONE or do not specify the FILL suboption at all, then the LINE suboption will be set to its default value. If FILL=PRED or FILL=SE, then the LINE suboption will be set to the same value as the FILL suboption.

OBS=*obs-options*
produces an overlaid scatter plot of the observations in addition to the specified contour plots. The following *obs-options* are available:

GRAD
specifies that observations be displayed as circles colored by the observed measurement. The same color gradient is used to display the prediction surface and the observations. Observations where the prediction is close to the observed values have similar colors—the greater the contrast between the color of an observation and the surface, the larger the prediction standard error is at that point.

LINEGRAD
is the same as OBS=GRAD except that a border is shown around each observation. This option is useful to identify the location of observations where the standard errors are small, because at these points the color of the observations and the color of the surface are indistinguishable.

NONE
specifies that no observations be displayed.

OUTL
specifies that observations be displayed as circles with a border but with a completely transparent fill.

OBS=NONE is the default; however, the default plot in PROC KRIGE2D displays the observations locations as outlines.

If you specify multiple instances of the FILL, LINE, or OBS suboptions in the same PREDICTION option, then the resulting predictions plot will honor the last value specified for any of the suboptions.

Any combination where you specify FILL=NONE and LINE=NONE is not available.

The PREDICTION option requires that predictions be requested at more than one location on a rectangular grid. The plot is not available when you use the GRIDDATA= data set option in the GRID statement, because the GRIDDATA= option allows you to specify any coordinates for the prediction locations.

SINGULARMSG=*number*
SMSG=*number*
controls the number of warning messages displayed for a singular matrix. When local kriging is performed, a separate kriging system is solved for each grid point. Anytime a singular matrix is encountered, a warning message is displayed up to a total of SINGULARMSG=*n* times. The default is SINGULARMSG=10.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC KRIGE2D to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the

input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement options NOTSORTED or DESCENDING in the BY statement for the KRIGE2D procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

COORDINATES Statement

COORDINATES | COORD *coordinate-variables* **;**

The following two options specify the names of the variables in the DATA= data set containing the values of the x and y coordinates of the data.

Only one COORDINATES statement is allowed, and it is applied to all PREDICT statements. In other words, it is assumed that all the VAR= variables in all PREDICT statements have the same x and y coordinates.

This is not a limitation. Since each VAR= variable is processed separately, observations for which the current VAR= variable is missing are excluded. With the next VAR= variable, the entire data are read again, this time excluding missing values in this next variable. Hence, a single run of PROC KRIGE2D can be used for variables measured at different locations without overlap.

XCOORD= *(variable-name)*

XC= *(variable-name)*

specifies the name of the variable containing the x coordinate of the data locations in the DATA= data set.

YCOORD= *(variable-name)*

YC= *(variable-name)*

specifies the name of the variable containing the y coordinate of the data locations in the DATA= data set.

GRID Statement

GRID *grid-options* **;**

You can use the following options to specify the grid of spatial locations for the kriging predictions. The grid specification is applied to all PREDICT and MODEL statements.

There are two basic methods for specifying the grid. You can specify the x and y coordinates explicitly, or they can be read from a SAS data set. The options for the explicit specification of grid locations are as follows:

X=*number*

X=$x_1, \ldots, x_m$

X=x_1 *to* x_m

X=x_1 *to* x_m *by* δx

specifies the x coordinate of the grid locations.

Y=*number*

Y=$y_1, \ldots, y_m$

Y=y_1 *to* y_m

Y=y_1 *to* y_m *by* δy

specifies the y coordinate of the grid locations.

For example, the following two GRID statements are equivalent.

```
grid x=1,2,3,4,5  y=0,2,4,6,8,10;
grid x=1 to 5 y=0 to 10 by 2;
```

To specify grid locations from a SAS data set, you must provide the name of the data set and the variables containing the values of the x and y coordinates.

GRIDDATA=*SAS-data-set*

GDATA=*SAS-data-set*

specifies a SAS data set containing the x and y grid coordinates.

XCOORD= *(variable-name)*

XC= *(variable-name)*

specifies the name of the variable containing the x coordinate of the grid locations in the GRIDDATA= data set.

YCOORD= *(variable-name)*

YC= *(variable-name)*

specifies the name of the variable containing the y coordinate of the grid locations in the GRIDDATA= data set.

PREDICT Statement

PREDICT | PRED | P *predict-options* **;**

You can specify the following options in a PREDICT statement.

MAXPOINTS=*number*

MAXP=*number*

MAX=*number*

specifies the maximum number of data points in a neighborhood. You specify this option in conjunction with the RADIUS= option. When the number of data points in the neighborhood formed at a given grid point by the RADIUS= option is greater than the MAXPOINTS= value, the RADIUS= value is decreased just enough to honor the MAXPOINTS= value unless you specify the NODECREMENT option. The default is to include all data points within the specified RADIUS= value. Neighborhoods with very large numbers of data points might lead to unnecessarily slow execution times and potential lack of memory issues, depending on the problem setup and your computational resources. In that case, you could use the MAXPOINTS= option to set a cap for your neighborhood size. For details about numerical considerations see the section "Computational Resources" on page 2966. Unless the RADIUS= option is also specified, when the MAXPOINTS= and NUMPOINTS= options are specified in the same PREDICT statement the MAXPOINTS= option is ignored.

MINPOINTS=*number*

MINP=*number*

MIN=*number*

specifies the minimum number of data points in a neighborhood. You specify this option in conjunction with the RADIUS= option. When the number of data points in the neighborhood formed at a given grid point by the RADIUS= option is less than the MINPOINTS= value, the RADIUS= value is increased just enough to honor the MINPOINTS= value unless you specify the NOINCREMENT option. The default is MINPOINTS=20. When enough data are available, you might improve prediction if you increase this value. When the MINPOINTS= and NUMPOINTS= options are specified in the same PREDICT statement the MINPOINTS= option is set to the value of NUMPOINTS=.

NODECREMENT | NODECR

requests that the RADIUS= value not be decremented when the MAXPOINTS= value is exceeded at a grid point. This option is relevant only when you specify both a RADIUS= value and a MAXPOINTS= value. In this case, when the number of points in the neighborhood constructed from the RADIUS= specification is greater than the MAXPOINTS= value, the RADIUS= value is decremented enough to honor the MAXPOINTS= value, and the kriging system is solved for this grid point. If you specify the NODECREMENT option, no decrementing is done, prediction is skipped at this grid point, and a message is written to the log.

NOINCREMENT | NOINCR

requests that the RADIUS= value not be incremented when the MINPOINTS= value is not met at a grid point. This option is relevant only when you specify both a RADIUS= value and

a MINPOINTS= number. In this case, when the number of points in the neighborhood constructed from the RADIUS= specification is less than the MINPOINTS= value, the RADIUS= value is incremented enough to honor the MINPOINTS= value, and the kriging system is solved for this grid point. If you specify the NOINCREMENT option, no incrementing is done, prediction is skipped at this grid point, and a message is written to the log.

NUMPOINTS=*number*
NPOINTS=*number*
NPTS=*number*
NP=*number*

specifies the exact size of a neighborhood. This option is incompatible with all other PREDICT statement options that control the neighborhood; it must appear by itself. In particular, if you specify both NUMPOINTS= and the RADIUS= option in the same PREDICT statement, then RADIUS= is honored, instead. In this event the value of the MINPOINTS= option is set to NUMPOINTS=, and the value of the MAXPOINTS= option is set to default, regardless whether these options have been specified or not. If you specify any of the MINPOINTS= or MAXPOINTS= option without the RADIUS= option in the same PREDICT statement as NUMPOINTS=, then the NUMPOINTS= option will be honored.

RADIUS=*number*
R=*number*

specifies the radius to use in a local kriging regression. When you specify this option, a separate kriging system is solved at each grid point by finding the neighborhood of this grid point consisting of all data points within the distance specified by the RADIUS= value. Note that if you specify the RADIUS= value to include data points situated beyond the extent of your problem's spatial correlation, this could result in unnecessary computational burden in your analysis. For additional control on the neighborhood see the MAXPOINTS= and MINPOINTS= options. When you specify the RADIUS= and NUMPOINTS= options in the same PREDICT statement, then RADIUS= is honored.

VAR= *variable-name*

specifies the single numeric variable used in the kriging system.

MODEL Statement

MODEL *model-options* **;**

You can use the following options to specify a semivariogram or covariance model. The specified model is used in the kriging system defined by the most previous PREDICT statement.

There are two ways to specify a semivariogram or covariance model. In the first method, you specify the required parameters SCALE, RANGE, and FORM, and possibly the optional parameters NUGGET, ANGLE, and RATIO, explicitly in the MODEL statement. In the second method, you specify an MDATA= data set. This data set contains variables corresponding to the required SCALE, RANGE, and FORM parameters, and optionally variables for the NUGGET, ANGLE, and RATIO parameters. The two methods are exclusive; either you specify all parameters explicitly, or they all are read from the MDATA= data set.

ANGLE=*angle*

ANGLE=*(angle*$_1$*,...,angle*$_k$*)*

specifies the angle of the major axis for anisotropic models, measured in degrees clockwise from the N-S axis. In the case of a nested semivariogram model, you can specify an angle for each nesting. The default is ANGLE=0.

FORM=SPHERICAL | EXPONENTIAL | GAUSSIAN | POWER

FORM=SPH | EXP | GAUSS | POW

specifies the functional form (type) of the semivariogram model. All the supported models are two-parameter models (SCALE= and RANGE=). A FORM= value is required; in the case of a nested semivariogram model, you must specify a form for each nesting.

See the section "Theoretical Semivariogram Models" on page 2947 for details on how the FORM= forms are determined.

MDATA=*SAS-data-set*

specifies the input data set that contains parameter values for the covariance or semivariogram model. The MDATA= data set must contain variables named SCALE, RANGE, and FORM=, and it can optionally contain variables NUGGET, ANGLE, and RATIO.

The FORM variable must be a character variable, assuming only the values allowed in the explicit FORM= syntax described previously. The RANGE and SCALE variables must be numeric. The optional variables ANGLE, RATIO, and NUGGET must also be numeric if present.

The number of observations present in the MDATA= data set corresponds to the level of nesting of the semivariogram model.

For example, to specify a non-nested model that uses a spherical covariance, an MDATA= data set might be given by the following statement:

```
data md1;
   input scale range form $;
   datalines;
   25 10 SPH
run;
```

The PROC KRIGE2D statement to use the MDATA= specification is of the form shown in the following:

```
proc krige2d data=...;
   predict var=....;
   model mdata=md1;
run;
```

This is equivalent to the following explicit specification of the covariance model parameters:

```
proc krige2d data=...;
   predict var=....;
   model scale=25 range=10 form=sph;
run;
```

The following MDATA= data set is an example of an anisotropic nested model:

```
data md1;
   input scale range form $ nugget angle ratio;
   datalines;
   20 8 S  5 35 0.7
   12 3 G  5 0  0.8
   4  1 G  5 45 0.5
   ;
```

This is equivalent to the following explicit specification of the covariance model parameters:

```
proc krige2d data=...;
   predict var=....;
   model scale=(20,12,4) range=(8,3,1) form=(S,G,G)
         angle=(35,0,45) ratio=(0.7,0.8,0.5) nugget=5;
run;
```

This example is somewhat artificial in that it is usually hard to detect different anisotropy directions and ratios for different nestings using an experimental semivariogram. Note that the NUGGET value is the same for all nestings. This is always the case; the nugget effect is a single additive term for all models. For further details, see the section "Details: VARIOGRAM Procedure" on page 7540 in the VARIOGRAM procedure.

NUGGET=*number*

specifies the nugget effect for the model. The nugget effect is due to a discontinuity in the semivariogram as determined by plotting the sample semivariogram (see the chapter on the VARIOGRAM procedure for details). For models without any nugget effect, this option is left out; the default is NUGGET=0.

RANGE=*range*

RANGE=*(range$_1$,...,range$_k$)*

specifies the range parameter in semivariogram models. If you have anisotropy, you must specify the range of the major anisotropy axis. In the case of a nested semivariogram model, you must specify a range for each nested structure.

The range parameter is the divisor in the exponent in all supported models except the power model. It has the units of distance, and it is related to the correlation scale for the underlying spatial process.

See the section "Theoretical Semivariogram Models" on page 2947 for details on how the RANGE= values are determined.

RATIO=*ratio*

RATIO=*(ratio$_1$,...,ratio$_k$)*

specifies the ratio of the length of the minor axis to the length of the major axis for anisotropic models. The value of the RATIO= option must be between 0 and 1. In the case of a nested semivariogram model, you can specify a ratio for each nesting. The default is RATIO=1.

SCALE=*scale*

SCALE=*(scale$_1$,...,scale$_k$)*

specifies the scale parameter in semivariogram models. In the case of a nested semivariogram model, you must specify a scale for each nesting.

The scale parameter is the multiplicative factor in all supported models; it has the same units as the variance of the VAR= variable in the preceding PREDICT statement.

See the section "Theoretical Semivariogram Models" on page 2947 for details on how the SCALE= values are determined.

SINGULAR=*number*

gives the singularity criteria for solving kriging systems. The larger the value of the SINGULAR= option, the easier it is for a kriging system to be declared singular. The default is SINGULAR=1E-7. See the section "Ordinary Kriging" on page 2964 for more detailed information.

Details: KRIGE2D Procedure

Theoretical Semivariogram Models

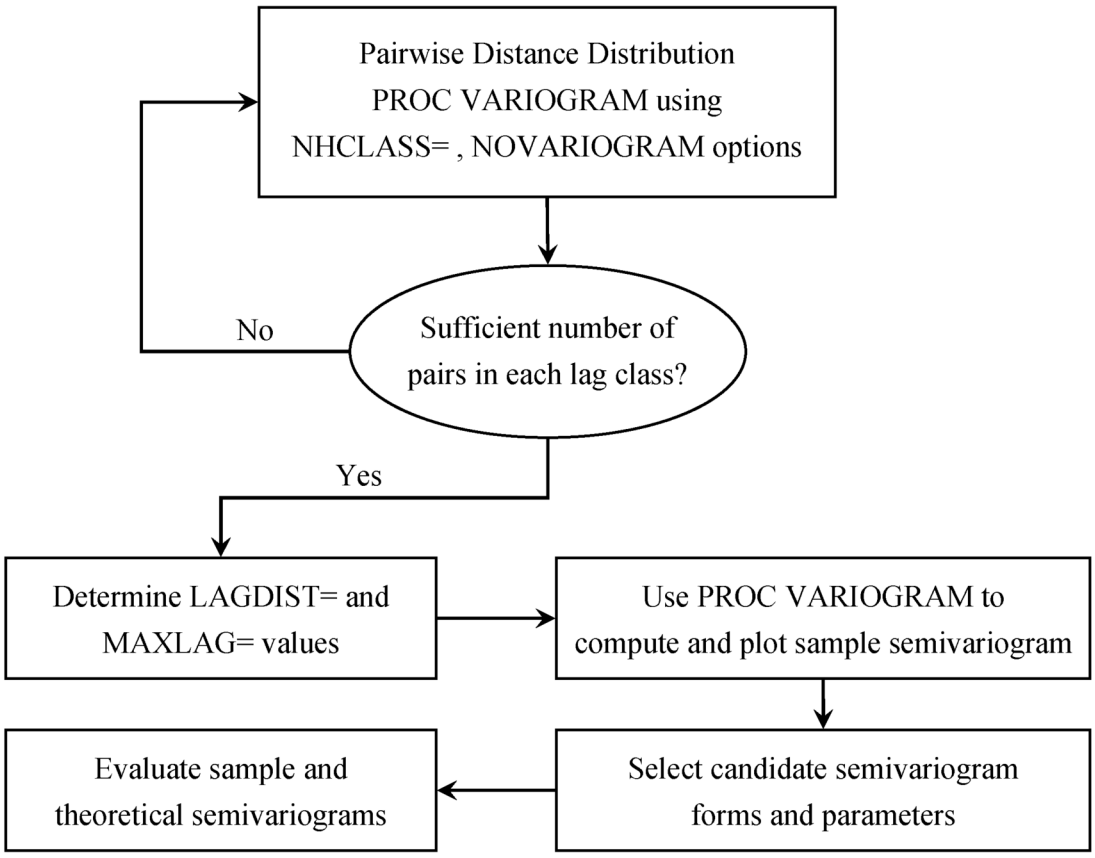
Consider a stochastic spatial process represented by the spatial random field (SRF; see Christakos 1992) $\{Z(\boldsymbol{s}), \boldsymbol{s} \in D \subset \mathcal{R}^2\}$. PROC VARIOGRAM computes the empirical (also known as sample or experimental) semivariance of $Z(\boldsymbol{s})$. Prediction of the spatial process $Z(\boldsymbol{s})$ at unsampled locations by techniques such as ordinary kriging requires a theoretical semivariogram or covariance.

When you use PROC VARIOGRAM and PROC KRIGE2D to perform spatial prediction, you must determine a suitable theoretical semivariogram based on the sample semivariogram. There are various methods of fitting semivariogram models, such as least squares, maximum likelihood, and robust methods (Cressie 1993, Section 2.6). A different approach is manual fitting, where a theoretical semivariogram is chosen based on visual inspection of the empirical estimate; see, for example, Hohn (1988, p. 25).

In some cases, a plot of the experimental semivariogram suggests that a single theoretical model is inadequate. Nested models, anisotropic models, and the nugget effect increase the scope of theoretical models available. All of these concepts are discussed in this section. The specification of the final theoretical model is provided by the syntax of PROC KRIGE2D.

Note the general flow of investigation. The empirical semivariogram is computed after a suitable choice is made for the LAGDISTANCE= and MAXLAGS= options in PROC VARIOGRAM, and possibly the NDIR= option or the DIRECTIONS statement for computations in more than one directions. Potential theoretical models (which can also incorporate nesting, anisotropy, and the nugget effect) are then plotted against the empirical semivariogram and evaluated. A suitable theoretical model is found by using the methodology presented in the section "Examples: VARIOGRAM Procedure" on page 7570 in the VARIOGRAM procedure. The flow of this process is illustrated in Figure 46.5.

Figure 46.5 Flowchart for Semivariogram Selection



Four theoretical models are supported by PROC KRIGE2D: the spherical, Gaussian, exponential, and power models—see also the section "Theoretical Semivariogram Models" on page 7540 in the VARIOGRAM procedure. For the first three types, the parameters a_0 and c_0, corresponding to the RANGE= and SCALE= options in the MODEL statement in PROC KRIGE2D, have the same dimensions and have similar effects on the shape of $\gamma_z(h)$, as illustrated in the following paragraph.

In particular, the dimension of c_0 is the same as the dimension of the variance of the spatial process $Z(\boldsymbol{s})$. The dimension of a_0 is length with the same units as h.

These four model forms are now examined in more detail.

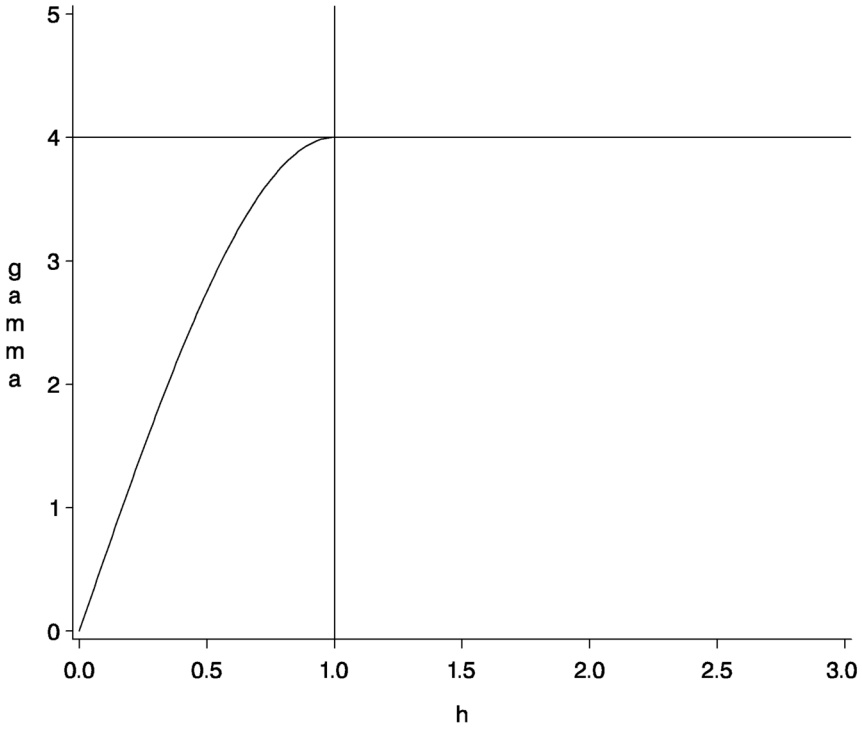
The Spherical Semivariogram Model

The form of the spherical model is

$$\gamma_z(h) = \begin{cases} c_0\left[\frac{3}{2}\frac{h}{a_0} - \frac{1}{2}\left(\frac{h}{a_0}\right)^3\right], & \text{for } h \le a_0 \\ c_0, & \text{for } h > a_0 \end{cases}$$

The shape is displayed in Figure 46.6, using range $a_0 = 1$ and scale $c_0 = 4$.

Figure 46.6 Spherical Semivariogram Model with Parameters $a_0 = 1$ and $c_0 = 4$



The vertical line at $h = 1$ shows the range a_0 of the model.

The horizontal line at 4.0 variance units (corresponding to $c_0 = 4$) is called the *sill.* In the case of the spherical model, $\gamma_z(h)$ actually reaches this value. For the following two model types, the sill is a horizontal asymptote.

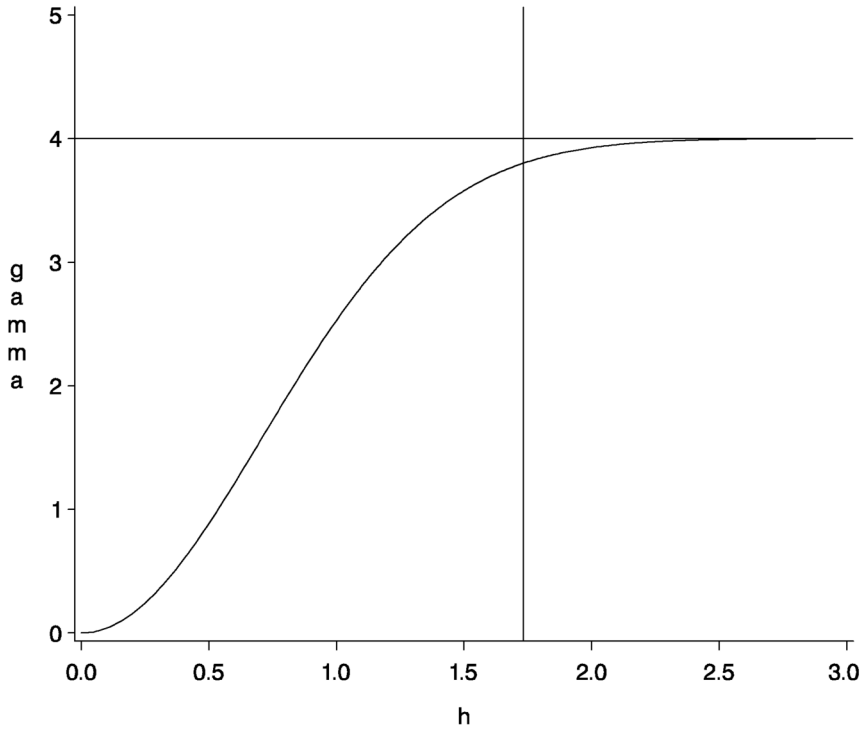
The Gaussian Semivariogram Model

The form of the Gaussian model is

$$\gamma_z(h) = c_0 \left[1 - \exp\left(-\frac{h^2}{a_0^2} \right) \right]$$

The shape is displayed in Figure 46.7, using range $a_0 = 1$ and scale $c_0 = 4$.

Figure 46.7 Gaussian Semivariogram Model with Parameters $a_0 = 1$ and $c_0 = 4$



The vertical line at $h = r_\epsilon = \sqrt{3}a_0$ shows the *effective* (or *practical*) *range* as defined by Deutsch and Journel (1992), or the *range* ϵ defined by Christakos (1992). The effective range is the h-value where the covariance is approximately 5% of its value at zero.

The horizontal line at 4.0 variance units (corresponding to $c_0 = 4$) is the sill; $\gamma_z(h)$ approaches the sill asymptotically.

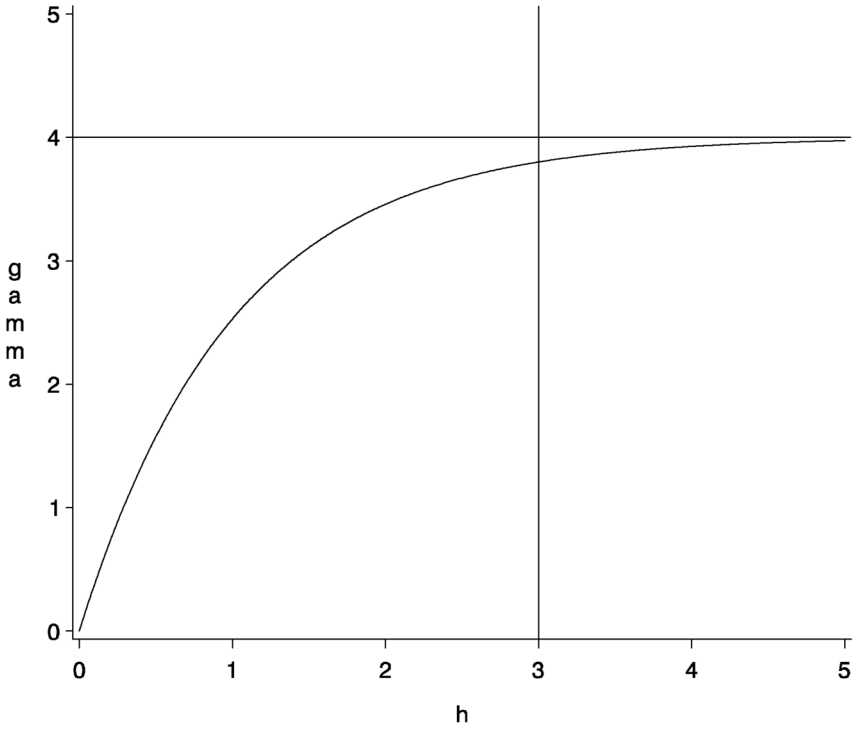
The Exponential Semivariogram Model

The form of the exponential model is

$$\gamma_z(h) = c_0\left[1 - \exp\left(-\frac{h}{a_0}\right)\right]$$

The shape is displayed in Figure 46.8, using range $a_0 = 1$ and scale $c_0 = 4$.

Figure 46.8 Exponential Semivariogram Model with Parameters $a_0 = 1$ and $c_0 = 4$



The vertical line at $h = r_\epsilon = 3a_0$ is the *effective* (or *practical*) *range*, or the *range* ϵ (that is, the h-value where the covariance is approximately 5% of its value at zero).

The horizontal line at 4.0 variance units (corresponding to $c_0 = 4$) is the sill, as in the previous model types.

It is noted from Figure 46.7 and Figure 46.8 that the major distinguishing feature of the Gaussian and exponential forms is the shape in the neighborhood of the origin $h = 0$. In general, small lags are important in determining an appropriate theoretical form based on a sample semivariogram.

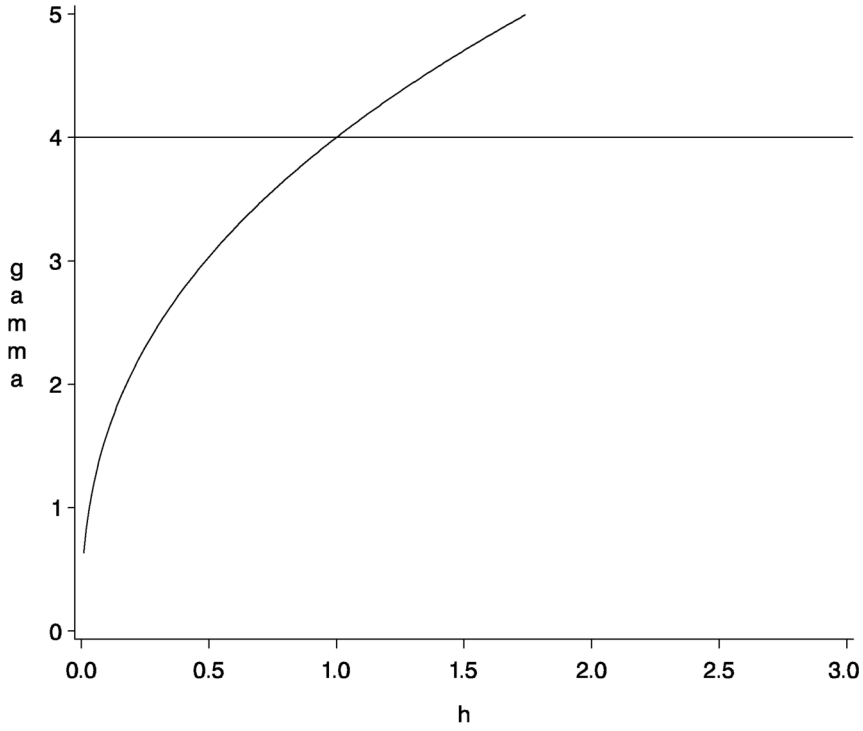
The Power Semivariogram Model

The form of the power model is

$$\gamma_z(h) = c_0 h^{a_0}$$

For this model, the parameter a_0 is a dimensionless quantity, with typical values $0 < a_0 < 2$. Note that the value of $a_0 = 1$ yields a straight line. The parameter c_0 has dimensions of the variance, as in the other models. There is no sill for the power model. The shape of the power model with $a_0 = 0.4$ and $c_0 = 4$ is displayed in Figure 46.9.

Figure 46.9 Power Semivariogram Model with Parameters $a_0 = 0.4$ and $c_0 = 4$



Nested Models

For a given set of spatial data, a plot of an experimental semivariogram might not seem to fit any of the individual theoretical models. In such a case, you might obtain a more accurate fit if you consider your covariance model to be the sum of two or more covariance structures. Such covariance models are called *nested* models. Nesting is common in geologic applications where there are correlations at different length scales. At small lag distances h, the smaller scale correlations dominate, while the large scale correlations dominate at larger lag distances.

Nested models are permissible covariances if they are the sum of permissible models. Therefore, you can include in a sum any combination of the models presented in the preceding subsections and produce permissible covariance models. As an illustration, consider two semivariogram models: an exponential and a spherical:

$$\gamma_{z,1}(h) = c_{0,1} \exp(-\frac{h}{a_{0,1}})$$

and

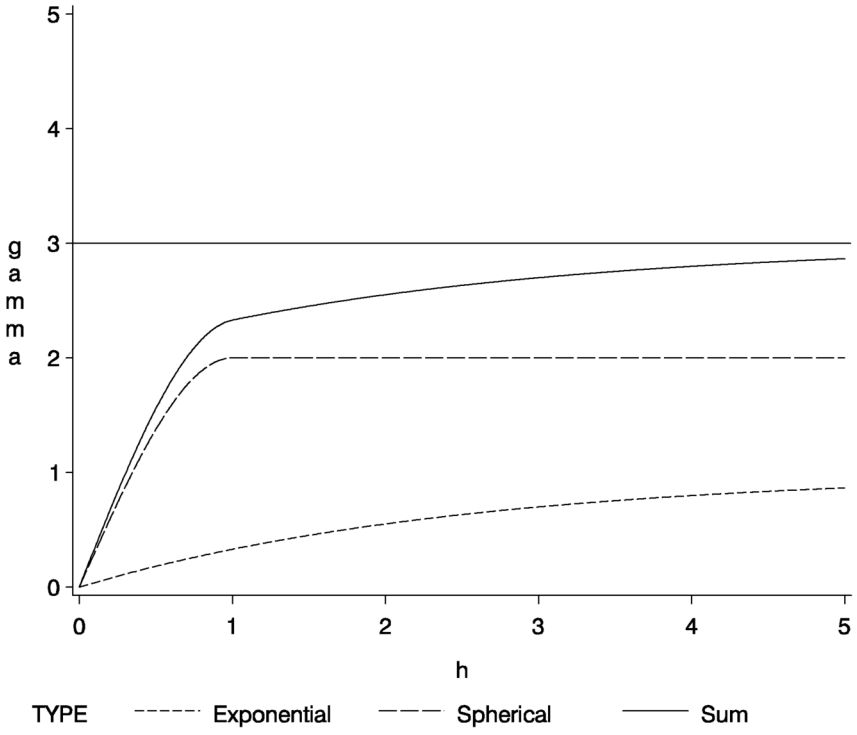
$$\gamma_{z,2}(h) = \left\{ \begin{array}{ll} c_{0,2}\left[\frac{3}{2}\frac{h}{a_{0,2}} - \frac{1}{2}(\frac{h}{a_{0,2}})^3\right], & \text{for } h \leq a_{0,2} \\ c_{0,2}, & \text{for } h > a_{0,2} \end{array} \right\}$$

with $c_{0,1} = 1, a_{0,1} = 2.5, c_{0,2} = 2$, and $a_{0,2} = 1$. If both of these correlation structures are present in a spatial process $\{Z(\boldsymbol{s}), \boldsymbol{s} \in D\}$, then the semivariance $\gamma_z(h)$ of this process can be expressed as

$$\gamma_z(h) = \gamma_{z,1}(h) + \gamma_{z,2}(h)$$

In this case, the experimental semivariogram $\gamma_z(h)$ for the process $Z(\boldsymbol{s})$ resembles the semivariogram of the sum of $\gamma_{z,1}(h)$ and $\gamma_{z,2}(h)$. This is illustrated in Figure 46.10.

Figure 46.10 Sum of Exponential and Spherical Structures at Different Scales



The sum of $\gamma_{z,1}(h)$ and $\gamma_{z,2}(h)$ in Figure 46.10 does not resemble any *single* theoretical semivariogram; however, its shape at $h = 1$ is similar to a spherical form. The asymptotic approach to a sill at three variance units, along with the shape around $h = 0$, indicates an exponential structure. Note that the sill value c_0 of the sum is the sum of the individual sills $c_{0,1} = 1$ and $c_{0,2} = 2$. In general, a nested model has a sill equal to the sum of the sills of its nested structures plus the nugget effect, if present.

Refer to Hohn (1988, p. 38ff) for further examples of nested correlation structures.

The Nugget Effect

For all the semivariogram models considered previously, the following property holds:

$$\gamma_z(0) = \lim_{h \downarrow 0} \gamma_z(h) = 0$$

However, a plot of the experimental semivariogram might indicate a discontinuity at $h = 0$; that is, $\gamma_z(h) \rightarrow c_n > 0$ as $h \rightarrow 0$, while $\gamma_z(0) = 0$. The quantity c_n is called the *nugget effect*; this term is from mining geostatistics where nuggets literally exist, and it represents variations at a much smaller scale than any of the measured pairwise distances—that is, at distances $h \ll h_{min}$, where

$$h_{min} = \min_{i,j} h_{ij} = \min_{i,j} | \boldsymbol{s}_i - \boldsymbol{s}_j |$$

There are conceptual and theoretical difficulties associated with a nonzero nugget effect; refer to Cressie (1993, Section 2.3.1) and Christakos (1992, Section 7.4.3) for details. There is no *practical* difficulty, however; you simply visually extrapolate the experimental semivariogram as $h \rightarrow 0$. The importance of availability of data at small lag distances is again illustrated.

As an example, an exponential semivariogram with a nugget effect c_n has the form

$$\gamma_z(h) = c_n + {\sigma_0}^2 \left[1 - \exp\left(-\frac{h}{a_0} \right) \right], h > 0$$

and

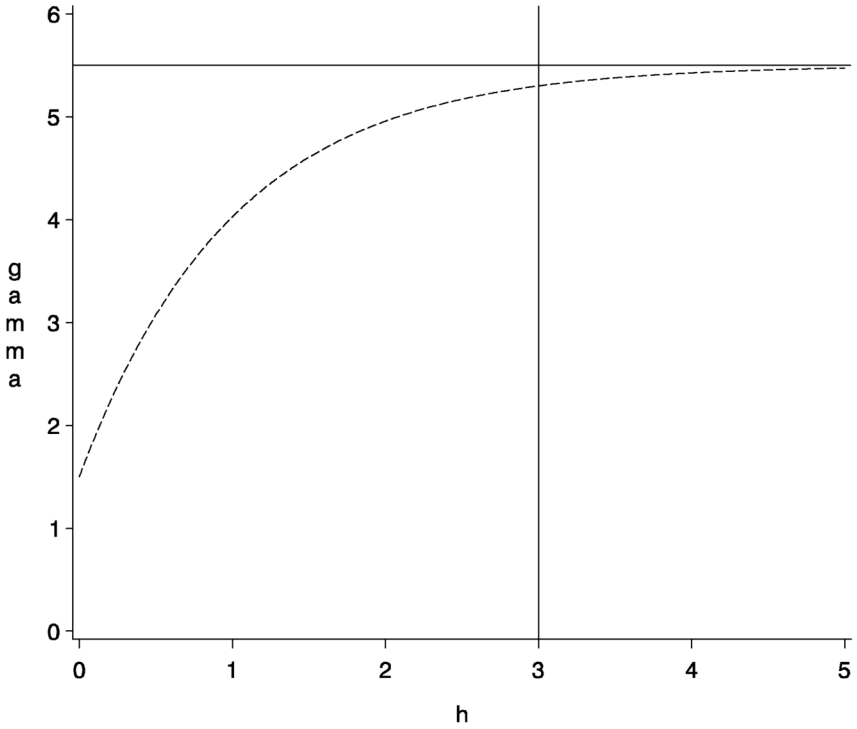
$$\gamma_z(0) = 0$$

where the factor ${\sigma_0}^2$ is called the *partial sill*, and the sill $c_0 = c_n + {\sigma_0}^2$.

This is illustrated in Figure 46.11 for the parameters $a_0 = 1$, ${\sigma_0}^2 = 4$, and nugget effect $c_n = 1.5$.

You can specify the nugget effect in PROC KRIGE2D with the NUGGET= option in the MODEL statement. It is a separate, additive term independent of direction; that is, it is isotropic. There is a way to approximate an anisotropic nugget effect; this is described in the following section.

Figure 46.11 Exponential Semivariogram Model with a Nugget Effect $c_n = 1.5$



Anisotropic Models

In all of the theoretical models considered previously, the lag distance h entered as a scalar value. This implies that the correlation between the spatial process at two point pairs P_1, P_2 is dependent *only* on the separation distance $h = | P_1 P_2 |$, not on the orientation of the vector $\boldsymbol{h}$. A spatial process described by an SRF $\{Z(\boldsymbol{s}), \boldsymbol{s} \in D \subset \mathcal{R}^2\}$ with this property is called isotropic, as is the associated covariance or semivariogram.

However, real spatial phenomena often show directional effects. Particularly in geologic applications, measurements along a particular direction might be highly correlated, while typically the perpendicular direction shows little or no correlation. Such processes are called anisotropic; see, for example, Journel and Huijbregts (1978, Section III.B.4).

When the correlation structure varies across different directions, you need different models for each direction so that you can account correctly for the continuity within the SRF. As you will see in the following subsections, techniques are applied to override the anisotropy effects for computational purposes. First, we examine characteristics of anisotropy.

The semivariogram sill is a measure of the process variability, hence the direction of the highest continuity is perpendicular to the direction where the highest sill occurs. If the sill is the same in all directions, then the direction with the highest range indicates highest continuity. The directions in which the spatial process $\{Z(\boldsymbol{s}), \boldsymbol{s} \in D\}$ is most and least correlated are called the *major* and *minor* axis of anisotropy, respectively.

In some cases, these directions are known a priori. This can occur in mining applications where the geology of a region is known in advance. In most cases, however, nothing is known about possible anisotropy. Depending on the amount of data available, using several directions is usually sufficient to determine the presence of anisotropy and to find the approximate major/minor axis directions; see the discussion in the section "Anisotropy" on page 7547 in the VARIOGRAM procedure documentation. You can find a detailed example of anisotropy investigation in the section "Example 95.2: An Anisotropic Case Study with Surface Trend in the Data" on page 7574 in the VARIOGRAM procedure documentation.

Once you explore an anisotropic process and you identify the minor and major axis directions, you can compute the *anisotropy factor* parameter R which is defined as

$$R = \frac{{a_0}^{min}}{{a_0}^{max}}$$

where ${a_0}^{min}$ is the semivariogram range in the direction of the minor axis and ${a_0}^{max}$ is the semivariogram range in the direction of the major axis.

There are two types of anisotropy, depending on which semivariogram characteristics change in different directions. These types are the *geometric* and the *zonal* anisotropy, and either or both can be present. Both are examined in detail in the following subsections.

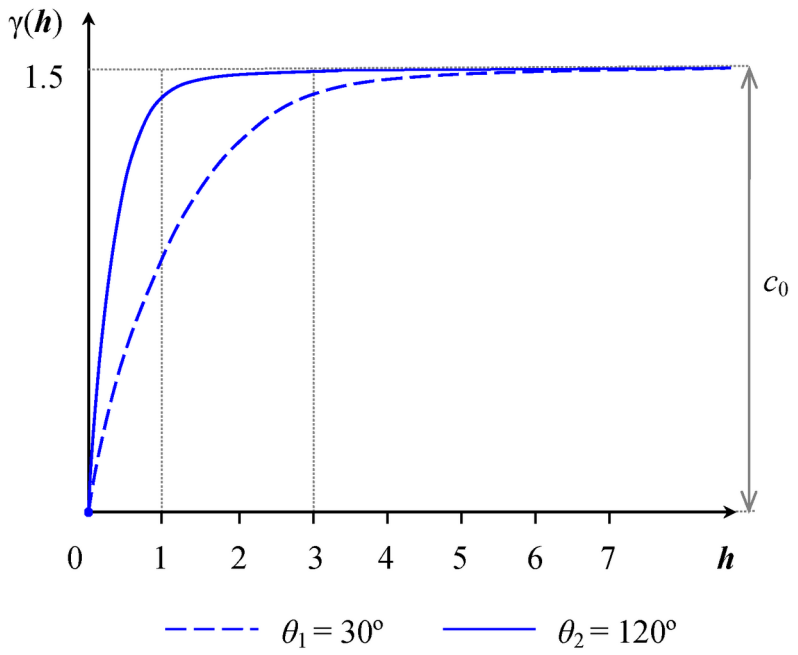
Geometric Anisotropy

Geometric anisotropy is the simplest type of anisotropy. It occurs when the same sill (or scale) parameter c_0 is present in all directions but the range a_0 changes with direction. Note also that in geometric anisotropy the same covariance form (or forms, if you use a nested model) is present in all directions.

Therefore, there is a single sill in geometric anisotropy, but the semivariogram reaches the sill in a shorter lag distance along a certain direction. This is illustrated in Figure 46.12, where an anisotropic exponential semivariogram is plotted. Assume that the two curves displayed in this figure have the same sill $c_0 = 1.5$ and are generated using the ranges $a_{0,1} = 3$ in the direction $\theta_1 = 30°$ (effective range is $r_{\epsilon,1} = 9$) and $a_{0,2} = 1$ in the direction $\theta_2 = 120°$ (effective range is $r_{\epsilon,2} = 3$).

As you can see from the figure, the ratio of the shorter to longer range is $R = 1/3$. The anisotropy factor R is the value to use in the RATIO= parameter in the MODEL statement in PROC KRIGE2D. Note that $R \leq 1$ for modeling geometric anisotropy. In fact, isotropy is a partial case of geometric anisotropy for which ${a_0}^{min} = {a_0}^{max}$ and $R = 1$.

Figure 46.12 Geometric Anisotropy with Major Axis in the Direction $\theta_1 = 30°$



The values of the RANGE= and ANGLE= parameters in the MODEL statement in PROC KRIGE2D are set based on the major anisotropy axis characteristics. Specifically, the RANGE= parameter is the value of the major axis range ${a_0}^{max} = a_{0,1}$, and the ANGLE= parameter is the angle θ_1 of the major axis measured clockwise from north (angles measured in this way are also known as *azimuths*). You can then specify the following MODEL statement in PROC KRIGE2D to approximate the covariance structure:

```
MODEL FORM=EXP RANGE=3 SCALE=1.5 ANGLE=30 RATIO=0.3333;
```

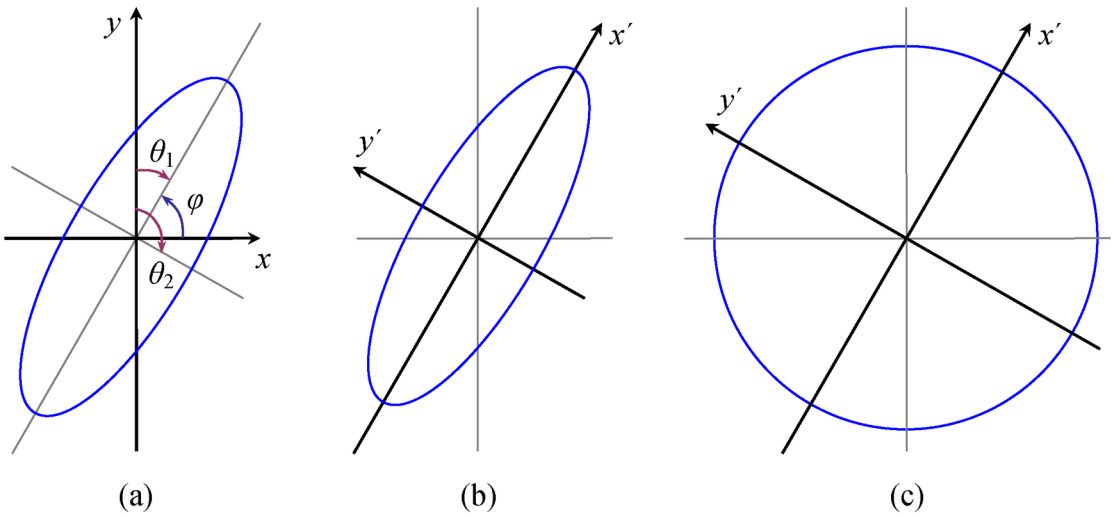
If you use a nested model, provide the type for each one of the nested structures with the FORM= option, and assign the individual SCALE= parameters so that they add up to the total sill (include in the sum the nugget effect, if present). In the typical case, all of your nested structures have the same anisotropy axes. This means that you specify the same ANGLE= parameter value for all structures. Each structure likely has its own values for the RANGE= and RATIO= parameters depending on the degree of its contribution to the nested model.

The terminology associated with geometric anisotropy is that of ellipses. To see how this comes about, consider the following hypothetical set of calculations. Let $\{Z(\boldsymbol{s}), \boldsymbol{s} \in D \subset \mathcal{R}^2\}$ be a geometrically anisotropic process, and assume there are sufficient data points to calculate an experimental semivariogram at a large number of angle classes $\theta \in \{0, \delta\theta, 2\delta\theta, \cdots, 180°\}$. At each of these angles θ, the experimental semivariogram is plotted and the range a_0 is recorded. A diagram in polar coordinates (a_0, θ) yields an ellipse with the major axis ${a_0}^{max}$ in the direction of the largest a_0 and the minor axis ${a_0}^{min}$ perpendicular to it. For the example in Figure 46.12, the ellipse is shown in Figure 46.13(a). Its major axis has size ${a_0}^{max}$ situated at angle θ_1 clockwise from north, and the minor axis has size ${a_0}^{min}$ oriented at angle θ_2 clockwise from north.

The KRIGE2D procedure handles geometric anisotropy by applying a reversible transformation in two steps that converts geometric anisotropy into isotropic conditions.

The first step is to align your coordinates axes with the anisotropy ellipse axes. Specifically, you choose to rotate by an angle φ the standard Cartesian orientation of the (x, y) coordinates system shown in Figure 46.13(a) so that the y axis coincides with the ellipse minor axis. The rotation result is illustrated in Figure 46.13(b). The second step is to elongate the minor axis so its length equals that of the major axis of the ellipse. You can see the result in Figure 46.13(c). The computational details are shown in the following.

Figure 46.13 Transformation Applied to Geometric Anisotropy



The transformation angle φ is measured in standard Cartesian orientation counterclockwise from the x axis (east). If the major axis azimuth is θ_1, then the Cartesian system of (x, y) needs to be rotated by $\varphi = 90° - \theta_1$ so that the y axis can coincide with the ellipse minor axis; see Figure 46.13(a). Let us call the ellipse major axis x' and the minor axis y'. The transformation that converts any coordinates in the (x, y) system into (x', y') coordinates in terms of φ is given by the matrix:

$$\mathbf{H} = \begin{pmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi) & \cos(\varphi) \end{pmatrix}$$

The elongation of the minor axis in the second step is performed with the matrix:

$$\mathbf{D}_R = \begin{pmatrix} 1 & 0 \\ 0 & 1/R \end{pmatrix}$$

Note that these two steps are sequential and their order cannot be reversed. For any point pair P_1 and P_2 with respective coordinates $\boldsymbol{s}_1 = (x_1, y_1)$ and $\boldsymbol{s}_2 = (x_2, y_2)$ in the (x, y) axes, their distance is given by:

$$| P_i P_j |_{(x,y)} = h = \sqrt{(\delta x)^2 + (\delta y)^2}$$

where the distance components $\delta x = x_2 - x_1$ and $\delta y = y_2 - y_1$. Based on the previous, the corresponding distances $\delta x'$ and $\delta y'$ in the (x', y') coordinates system are given by the vector:

$$\begin{pmatrix} \delta x' \\ \delta y' \end{pmatrix} = \mathbf{D}_R \, \mathbf{H} \begin{pmatrix} \delta x \\ \delta y \end{pmatrix} = \begin{pmatrix} \cos(\varphi) & \sin(\varphi) \\ -\sin(\varphi)/R & \cos(\varphi)/R \end{pmatrix} \begin{pmatrix} \delta x \\ \delta y \end{pmatrix}$$

The transformed interpair distance is then:

$$\mid P_i P_j \mid_{(x',y')} = h' = \sqrt{(\delta x')^2 + (\delta y')^2}$$

As a result, the original anisotropic semivariogram in Figure 46.12 that was a function $\gamma(\boldsymbol{h}) = \gamma(h, \theta)$ of both h and θ is then transformed to an equivalent function $\hat{\gamma}(h')$ only of h':

$$\hat{\gamma}(h') = \gamma(\boldsymbol{h})$$

This single isotropic semivariogram $\hat{\gamma}(h')$ is then used for kriging purposes.

Note that the two steps used by PROC KRIGE2D in the previous analysis can be performed in a different manner, as well. For instance, you might equivalently choose to rotate the (x, y) Cartesian coordinates so that the y axis coincides with the ellipse major axis, rather than with the minor axis as was shown earlier. Also, you might prefer to compress the major axis rather than elongating the short one. In any case, you need to perform the appropriate computations for the transformation of your choice.

Zonal Anisotropy

In zonal anisotropy, the sill (or scale) parameter c_0 is different for different directions. It is not possible to transform such a structure into an isotropic semivariogram. Instead, nesting and geometric anisotropy are used together to approximate zonal anisotropy. Due to this process, you might also have to use different covariance forms in different directions, as shown in the following discussion.

The main idea when you deal with zonal anisotropy is to specify structures in the covariance model that contribute exclusively in particular directions and thus increase the sill only in these directions. After you have identified such a structure, you assign it a suitable range in the contributing direction. The contributing direction is the one of maximum continuity for this structure; hence it is the structure's major anisotropy axis with a range of ${a_0}^{max}$. For all other specified directions you want to use an infinite range that suggests that the structure sill is never reached in these directions.

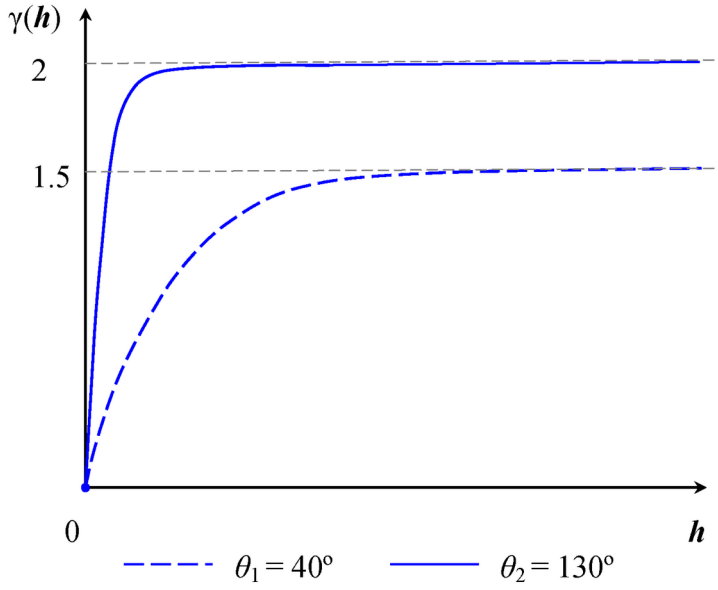
Practically, you can consider the structure range in any non-contributing direction to be a very large number compared to ${a_0}^{max}$. This means that the anisotropy factor R for this structure is a very large number, too. Accordingly, in PROC KRIGE2D you indicate the presence of zonal anisotropy and a covariance structure that contributes only in a particular direction by specifying the RATIO= parameter of this structure to a be large number. The two examples that follow illustrate different instances of zonal anisotropy and how to specify the corresponding covariance model parameters in PROC KRIGE2D.

Example 1

The first example shows that if you can model the direction with the highest sill as a nested model, you then treat the case as a composition of geometric anisotropy and an additional structure that acts only in the direction of the increased sill.

Consider a spatial process where the fitting of theoretical models in your experimental semivariogram produces a correlation structure like the one shown in Figure 46.14. In the direction $\theta_1 = 40°$, the covariance model has a single exponential structure $\gamma_1(\boldsymbol{h}) = \text{Exp}(a_{0,1E}, c_{0,1E})$ with range $a_{0,1E} = 2$ and sill $c_{0,1E} = 1.5$. In the direction $\theta_2 = 130°$, the covariance model $\gamma_2(\boldsymbol{h}) = \text{Exp}(a_{0,2E}, c_{0,2E}) + \text{Sph}(a_{0,2S}, c_{0,2S})$ has two nested structures: an exponential structure with range $a_{0,2E} = 0.5$ and sill $c_{0,2E} = 1.5$ and a spherical structure with range $a_{0,2S} = 1$ and sill $c_{0,2S} = 0.5$.

Figure 46.14 Zonal Anisotropy in Two Directions



The total sill in the direction θ_2 of the highest variance is the sum of the nested structures sills $c_{0,2E} + c_{0,2S} = 2$. You can consider that your process is characterized: (a) by a geometrically anisotropic exponential structure with common sill $c_{0,E} = 1.5$ across all directions and major axis range $a_{0,1E} = 2$ and (b) by a spherical structure which is a zonal anisotropy component that contributes only in the θ_2 direction. Based on the remarks in this section, the RATIO= parameter for the exponential structure is $R_E = 0.5/2 = 0.25$, whereas for the spherical structure you choose a large value, such as $R_S = 10^8$.

Then, you can approximate this structure in PROC KRIGE2D by specifying the two structures with the following MODEL statement:

```
MODEL FORM=(EXP,SPH) RANGE=(2,1) SCALE=(1.5,0.5)
      ANGLE=(40,130) RATIO=(0.25,1E8);
```

You can handle more elaborate cases in a similar way, where the covariance models in different directions might all be nested models. Your goal is to model the continuity by starting with a sum of isotropic or geometrically anisotropic structures whose total sill is the lowest sill in all directions. Then, in each of the directions with higher sills you add a zonal anisotropy component in the corresponding sum that compensates for the increased variability in that direction.

Example 2

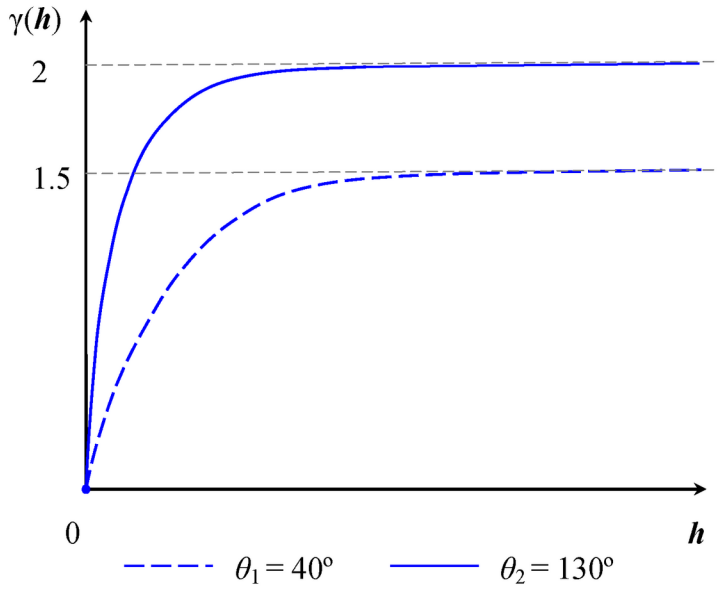
The second example provides important perspective of the physics in the zonal anisotropy analysis. It is an extreme case of the general guidelines for zonal anisotropy. You examine what happens when each of the directions can be modeled with a simple, nonnested model, and the sills for these models are clearly different.

Consider a spatial process with a spatial continuity description almost identical to the one in the previous example. In the direction θ_1, the covariance model has again a single exponential structure $\gamma_1(\boldsymbol{h}) = \text{Exp}(a_{0,1E}, c_{0,1E})$ with range $a_{0,1E} = 2$ and sill $c_{0,1E} = 1.5$. However, this time in the direction θ_2 you have fit the experimental semivariogram by using a single exponential structure $\gamma_2(\boldsymbol{h}) = \text{Exp}(a_{0,2E}, c_{0,2E})$ with range $a_{0,2E} = 1$ and sill $c_{0,2E} = 2$. These models are shown in Figure 46.15.

In this case you have a simplified situation with a single covariance structure in each direction, and neither structure contributes to the sill of the other direction. Hence, you could consider them both to be zonal anisotropy components, where there is no variation due to $\gamma_1(\boldsymbol{h})$ in the θ_2 direction and no variation due to $\gamma_2(\boldsymbol{h})$ in the θ_1 direction. As a result, both structures can be practically approximated by specifying two models with large RATIO= values. You could then use the following MODEL statement in PROC KRIGE2D to describe the covariance in this example:

```
MODEL FORM=(EXP,EXP) RANGE=(2,1) SCALE=(1.5,2)
      ANGLE=(40,130) RATIO=(1E8,1E8);
```

Figure 46.15 Zonal Anisotropy in Two Directions



However, note that the preceding representation of the spatial continuity accounts only for the specified directions of angles θ_1 and θ_2. Then, the lack of an underlying covariance structure in all other directions can be interpreted as zero continuity everywhere in the process except along the axes at angles θ_1 and θ_2. This would be a very unlikely occurrence in natural processes.

An acceptable alternative would be an attempt to represent the continuity by filling in the intermediate levels of the sill between its lowest and highest values. This would require that you provide zonal anisotropy components for as many directions as possible, thus monitoring closely the spatial continuity by means of the directional change of the process sill. However, in this way you are considering discretized directional intervals, so you are still missing a description for the spatial continuity in the intermediate directions within these intervals. This is also a more complicated approach which can be more tedious than the analysis in the previous example.

For that reason, it is strongly advised that in similar cases you should try to use the analysis illustrated in the previous example. In particular, you should try to model the highest sill direction as a nested structure, such that it comprises a geometrical anisotropy component with cumulative sill equal to the lower sill and a zonal anisotropy component that accounts for the sill difference.

Anisotropic Nugget Effect

Note that an isotropic nugget effect can be approximated by using nested models, with one of the nested structures having a small range. Applying a geometric anisotropy specification to this nested structure results in an anisotropic nugget effect.

Details of Ordinary Kriging

Introduction

There are three common characteristics often observed with spatial data (that is, data indexed by their spatial locations):

(i) slowly varying, large-scale variations in the measured values

(ii) irregular, small-scale variations

(iii) similarity of measurements at locations close together

As an illustration, consider a hypothetical example in which an organic solvent leaks from an industrial site and spreads over a large area. Assume the solvent is absorbed and immobilized into the subsoil above any groundwater level, so you can ignore any time dependence.

To find the areal extent and the concentration values of the solvent, you need measurements. Although the problem is inherently three-dimensional, if you measure total concentration in a column of soil or take a depth-averaged concentration, it can be handled reasonably well with two-dimensional techniques.

You usually assume that measured concentrations are higher closer to the source and decrease at larger distances from the source. On top of this smooth variation, there are small-scale variations in the measured concentrations, due perhaps to the inherent variability of soil properties.

You also tend to suspect that measurements made close together yield similar concentration values, while measurements made far apart can have very different values.

These physically reasonable qualitative statements have no explicit probabilistic content, and there are a number of numerical smoothing techniques, such as inverse distance weighting and splines, that make use of large-scale variations and "close distance-close value" characteristics of spatial data to interpolate the measured concentrations for contouring purposes.

While characteristics (i) and (iii) are handled by such smoothing methods, characteristic (ii), the small-scale residual variation in the concentration field, is not accounted for.

There can be situations, due to the use of the prediction map or the relative magnitude of the irregular fluctuations, where you cannot ignore these small-scale irregular fluctuations. In other words, the smoothed or predicted values of the concentration field alone are not a sufficient characterization; you also need the possible spread around these contoured values.

Spatial Random Fields

One method of incorporating characteristic (ii) into the construction of a contour map is to model the concentration field as a spatial random field (SRF). The mathematical details of SRF models are given in a number of texts, such as Cressie (1993) and Christakos (1992). The mathematics of SRFs is formidable. However, under certain simplifying assumptions, it produces classical linear predictors with very simple properties, allowing easy implementation for prediction purposes. These predictors, primarily ordinary kriging (OK), give both a prediction and a standard error of prediction at unsampled locations. This allows the construction of a map of both predicted values and level of uncertainty about the predicted values.

The key assumption in applying the SRF formalism is that the measurements come from a single realization of the SRF. However, in most geostatistical applications, the focus is on a single, unique realization. This is unlike most other situations in stochastic modeling in which there will be future experiments or observational activities (at least conceptually) under similar circumstances. This renders many traditional ideas of statistical inference ambiguous and somewhat counterintuitive.

There are additional logical and methodological problems in applying a stochastic model to a unique but partly unknown natural process; refer to the introduction in Matheron (1971) and Cressie (1993, Section 2.3). These difficulties have resulted in attempts to frame the prediction problem in a completely deterministic way (Isaaks and Srivastava 1988; Journel 1985). Also, some issues with kriging, and with spatial prediction methods in general, are related to the necessary assumption of ergodicity of the spatial process. This assumption is required to estimate the covariance or semivariogram from sample data. Details are provided in Cressie (1993, pp. 52–58).

Despite these difficulties, ordinary kriging remains a popular and widely used tool in modeling spatial data, especially in generating surface plots and contour maps. An abbreviated derivation of the OK predictor for point prediction and the associated standard error is discussed in the following section. Full details are given in Journel and Huijbregts (1978), Christakos (1992), and Cressie (1993).

Ordinary Kriging

Denote the SRF by $Z(\boldsymbol{s}), \boldsymbol{s} \in D \subset \mathcal{R}^2$. Following the notation in Cressie (1993), the following model for $Z(\boldsymbol{s})$ is assumed:

$$Z(\boldsymbol{s}) = \mu + \varepsilon(\boldsymbol{s})$$

Here, μ is the fixed, unknown mean of the process, and $\varepsilon(\boldsymbol{s})$ is a zero mean SRF representing the variation around the mean.

In most practical applications, an additional assumption is required in order to estimate the covariance C_z of the $Z(\boldsymbol{s})$ process. This assumption is second-order stationarity:

$$C_z(\boldsymbol{s}_1, \boldsymbol{s}_2) = \mathrm{E}[\varepsilon(\boldsymbol{s}_1)\varepsilon(\boldsymbol{s}_2)] = C_z(\boldsymbol{s}_1 - \boldsymbol{s}_2) = C_z(\boldsymbol{h})$$

This requirement can be relaxed slightly when you are using the semivariogram instead of the covariance. In this case, second-order stationarity is required of the differences $\varepsilon(\boldsymbol{s}_1) - \varepsilon(\boldsymbol{s}_2)$ rather than $\varepsilon(\boldsymbol{s})$:

$$\gamma_z(\boldsymbol{s}_1, \boldsymbol{s}_2) = \frac{1}{2}\mathrm{E}[(\varepsilon(\boldsymbol{s}_1) - \varepsilon(\boldsymbol{s}_2))^2] = \gamma_z(\boldsymbol{s}_1 - \boldsymbol{s}_2) = \gamma_z(\boldsymbol{h})$$

By performing local kriging, the spatial processes represented by the previous equation for $Z(\boldsymbol{s})$ are more general than they appear. In local kriging, at an unsampled location $\boldsymbol{s}_0$, a separate model is fit using only data in a neighborhood of $\boldsymbol{s}_0$. This has the effect of fitting a separate mean μ at each point, and it is similar to the *kriging with trend* (KT) method discussed in Journel and Rossi (1989).

Given the N measurements $Z(\boldsymbol{s}_1), \ldots, Z(\boldsymbol{s}_N)$ at known locations $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N$, you want to obtain a prediction of Z at an unsampled location $\boldsymbol{s}_0$. When the following three requirements are imposed on the predictor $\hat{Z}$, the OK predictor is obtained.

(i) $\hat{Z}$ is linear in $Z(\boldsymbol{s}_1), \cdots, Z(\boldsymbol{s}_N)$.

(ii) $\hat{Z}$ is unbiased.

(ii) $\hat{Z}$ minimizes the mean square prediction error $\mathrm{E}[(Z(\boldsymbol{s}_0) - \hat{Z}(\boldsymbol{s}_0))^2]$.

Linearity requires the following form for $\hat{Z}(\boldsymbol{s}_0)$:

$$\hat{Z}(\boldsymbol{s}_0) = \sum_{i=1}^{N} \lambda_i Z(\boldsymbol{s}_i)$$

Applying the unbiasedness condition to the preceding equation yields

$$\mathrm{E}[\hat{Z}(\boldsymbol{s}_0)] = \mu \Rightarrow \sum_{i=1}^{N} \lambda_i \mathrm{E}[Z(\boldsymbol{s}_i)] = \mu \Rightarrow \sum_{i=1}^{N} \lambda_i \mu = \mu \Rightarrow \sum_{i=1}^{N} \lambda_i = 1$$

Finally, the third condition requires a constrained linear optimization involving $\lambda_1, \cdots, \lambda_N$ and a Lagrange parameter $2m$. This constrained linear optimization can be expressed in terms of the function $L(\lambda_1, \cdots, \lambda_N, m)$ given by

$$L = \mathrm{E}\left[\left(Z(\boldsymbol{s}_0) - \sum_{i=1}^{N} \lambda_i Z(\boldsymbol{s}_i)\right)^2\right] - 2m\left(\sum_{i=1}^{N} \lambda_i - 1\right)$$

Define the $N \times 1$ column vector $\boldsymbol{\lambda}$ by

$$\boldsymbol{\lambda} = (\lambda_1, \cdots, \lambda_N)^T$$

and the $(N+1) \times 1$ column vector $\boldsymbol{\lambda_0}$ by

$$\boldsymbol{\lambda_0} = (\lambda_1, \cdots, \lambda_N, m)^T = \begin{pmatrix} \boldsymbol{\lambda} \\ m \end{pmatrix}$$

The optimization is performed by solving

$$\frac{\partial L}{\partial \boldsymbol{\lambda_0}} = \mathbf{0}$$

in terms of $\lambda_1, \cdots, \lambda_N$ and m.

The resulting matrix equation can be expressed in terms of either the covariance $C_z(\boldsymbol{h})$ or semivariogram $\gamma_z(\boldsymbol{h})$. In terms of the covariance, the preceding equation results in the following matrix equation:

$$\mathbf{C}\boldsymbol{\lambda_0} = \mathbf{C_0}$$

where

$$\mathbf{C} = \begin{pmatrix} C_z(\mathbf{0}) & C_z(\boldsymbol{s}_1 - \boldsymbol{s}_2) & \cdots & C_z(\boldsymbol{s}_1 - \boldsymbol{s}_N) & 1 \\ C_z(\boldsymbol{s}_2 - \boldsymbol{s}_1) & C_z(\mathbf{0}) & \cdots & C_z(\boldsymbol{s}_2 - \boldsymbol{s}_N) & 1 \\ & & \ddots & & \\ C_z(\boldsymbol{s}_N - \boldsymbol{s}_1) & C_z(\boldsymbol{s}_N - \boldsymbol{s}_2) & \cdots & C_z(\mathbf{0}) & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix}$$

and

$$\mathbf{C_0} = \begin{pmatrix} C_z(\boldsymbol{s}_0 - \boldsymbol{s}_1) \\ C_z(\boldsymbol{s}_0 - \boldsymbol{s}_2) \\ \vdots \\ C_z(\boldsymbol{s}_0 - \boldsymbol{s}_N) \\ 1 \end{pmatrix}$$

The solution to the previous matrix equation is

$$\hat{\boldsymbol{\lambda}}_\mathbf{0} = \mathbf{C}^{-1}\mathbf{C_0}$$

Using this solution for $\boldsymbol{\lambda}$ and m, the ordinary kriging prediction at r_0 is

$$\hat{Z}(\boldsymbol{s}_0) = \lambda_1 Z(\boldsymbol{s}_1) + \cdots + \lambda_N Z(\boldsymbol{s}_N)$$

with associated prediction error the square root of the variance

$${\sigma_z}^2(\boldsymbol{s}_0) = C_z(\mathbf{0}) - \boldsymbol{\lambda}'\mathbf{c_0} + m$$

where $\mathbf{c_0}$ is $\mathbf{C_0}$ with the 1 in the last row removed, making it an $N \times 1$ vector.

These formulas are used in the best linear unbiased prediction (BLUP) of random variables (Robinson 1991). Further details are provided in Cressie (1993, pp. 119–123).

Because of possible numeric problems when solving the previous matrix equation, Deutsch and Journel (1992) suggest replacing the last row and column of 1s in the preceding matrix $\mathbf{C}$ by $C_z(0)$, keeping the 0 in the $(N+1, N+1)$ position and similarly replacing the last element in the preceding right-hand vector $\mathbf{C_0}$ with $C_z(0)$. This results in an equivalent system but avoids numeric problems when $C_z(0)$ is large or small relative to 1.

Computational Resources

To generate a predicted value at a single grid point by using N data points, PROC KRIGE2D must solve the following kriging system:

$$\mathbf{C}\boldsymbol{\lambda_0} = \mathbf{C_0}$$

where the dimensions of $\mathbf{C}$ are $(N+1) \times (N+1)$, and the right-hand-side $\mathbf{C_0}$ has one column.

Holding the matrix and vector associated with this system in core requires approximately $8N^2/2$ bytes. The CPU time used in solving the system is proportional to N^3. For large N, this time dominates the $O(N^2)$ time to compute the elements of the covariance matrix $\mathbf{C}$ from the specified covariance or semivariogram model.

For local kriging, the kriging system is set up and solved for each grid point. Part of the setup process involves determining the neighborhood of each grid point. A fast K-D tree algorithm is used to determine neighborhoods. For G grid points, the dominant CPU time factor is setting up and solving the G kriging systems. The N in the algorithm of the section "Ordinary Kriging" on page 2964 is the number of data points in a given neighborhood, and it can differ for each grid point.

In global kriging, the entire input data set and all grid points are used to set up and solve the single system

$$\mathbf{C}\boldsymbol{\lambda_0} = \mathbf{C_0}$$

Again $\mathbf{C}$ has dimensions $(N+1) \times (N+1)$, but $\boldsymbol{\lambda_0}$ and $\mathbf{C_0}$ now have G columns, where G is the number of grid points. Memory requirements are approximately $8[(N^2/2) + GN]$ bytes. The CPU time used in solving the system is still dominated by the N^3 factorization of the left-hand side.

Output Data Sets

The KRIGE2D procedure produces two data sets: the OUTEST=*SAS-data-set* and the OUTNBHD=*SAS-data-set*. These data sets are described as follows.

OUTEST=*SAS-data-set*

The OUTEST= data set contains the kriging predictions and the associated standard errors. The OUTEST= data set contains the following variables:

- ESTIMATE, which is the kriging prediction for the current variable
- GXC, which is the x coordinate of the grid point at which the kriging prediction is made
- GYC, which is the y coordinate of the grid point at which the kriging prediction is made
- LABEL, which is the label for the current PREDICT/MODEL combination producing the kriging prediction. If you do not specify a label, default labels of the form Pred*j*.Model*k* are used.
- NPOINTS, which is the number of points used in the prediction. This number varies for each grid point if local kriging is performed.
- STDERR, which is the standard error of the kriging predict
- VARNAME, which is the variable name

OUTNBHD=*SAS-data-set*

When you specify the RADIUS= option or the NUMPOINTS= option in the PREDICT statement, local kriging is performed. Local kriging is simply ordinary kriging at a given grid location, using only those data points in a neighborhood defined by the RADIUS= value or the NUMPOINTS= value.

The OUTNBHD= data set contains one observation for each data point in each neighborhood. Hence, this data set can be large. For example, if the grid specification results in 1,000 grid points and each grid point has a neighborhood of 100 points, the resulting OUTNBHD= data set contains 100,000 points.

The OUTNBHD= data set contains the following variables:

- GXC, which is the x coordinate of the grid point
- GYC, which is the y coordinate of the grid point
- LABEL, which is the label for the current PREDICT/MODEL combination. If you do not specify a label, default labels of the form Pred*j*.Model*k* are used.

- NPOINTS, which is the number of points used in the prediction
- RADIUS, which is the radius used for each neighborhood
- VALUE, which is the value of the variable at the current data point
- VARNAME, which is the variable name of the current variable
- XC, which is the x coordinate of the current data point
- YC, which is the y coordinate of the current data point

Displayed Output

In addition to the output data sets, the KRIGE2D procedure produces output objects as well. The KRIGE2D procedure output objects are the following:

- a default "Number of Observations" table that displays the number of observations read from the input data set and the number of observations used in the analysis
- a map showing the spatial distribution of the observations of the current VAR variable in the PREDICT statement. The observations are displayed by default with circled markers whose color indicates the VAR value at the corresponding location.
- a default table for each PREDICT statement that sums up basic information about the kriging analysis
- a default table for each MODEL statement that shows the covariance model parameters for the corresponding PREDICT statement
- plots of the kriging prediction and the prediction standard error at each point of the specified output grid. The KRIGE2D procedure produces by default a plot of the kriging prediction and the corresponding prediction error for each MODEL statement of every PREDICT statement that you specify. You can produce more of these plots with styles that you can specify by using the available suboptions of the PLOTS=PREDICTION option.

ODS Table Names

Each table created by PROC KRIGE2D has a name associated with it, and you must use this name to reference the table when using ODS Graphics. These names are listed in Table 46.2.

Table 46.2 ODS Tables Produced by PROC KRIGE2D

ODS Table Name	Description	Statement	Option
KrigInfo	Kriging analysis general information	PROC	default output
ModelInfo	Parameters of the covariance model used in current kriging analysis	PROC	default output
NObs	Number of observations read and used	PROC	default output

ODS Graphics

This section describes the use of the Output Delivery System (ODS) for creating graphics with the KRIGE2D procedure.

To request these graphs, you must specify the ODS GRAPHICS statement. For additional control of the graphics that are displayed, see the PLOTS= option in the section "PROC KRIGE2D Statement" on page 2937. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

ODS Graph Names

PROC KRIGE2D assigns a name to each graph it creates by using ODS Graphics. You can use these names to reference the graphs when using ODS Graphics. The names are listed in Table 46.3.

Table 46.3 ODS Graphics Produced by PROC KRIGE2D

ODS Graph Name	Plot Description	Statement	Option
ObservationsPlot	Scatter plot of observed data and colored markers indicating observed values	PROC	PLOTS=OBSERV
PredictionPlot	Contour plots of the kriging prediction, surface of the prediction error, and outlines of the observation locations	PROC	PLOTS=PREDICTION

To request these graphs, you must specify the ODS GRAPHICS statement in addition to the statements indicated in Table 46.3. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Examples: KRIGE2D Procedure

Example 46.1: Investigating the Effect of Model Specification on Prediction

In the section "Example 95.1: Theoretical Semivariogram Model Fitting" on page 7570 in the VARIOGRAM procedure, a particular theoretical semivariogram is chosen for the coal seam thickness data. The chosen semivariogram is Gaussian with a scale (sill) of $c_0 = 7.2881$, and a range of $a_0 = 30.6239$. This choice of the semivariogram is based on an automated fit by using PROC NLIN.

Another possible choice of model is the spherical semivariogram with the same scale (sill) of $c_0 = 7.5$ but with a range of $a_0 = 60$. This choice of range comes from a visual fit, which is based on a comparison of the plots of the regular and robust sample semivariograms and the spherical semivariogram for various scale (sill) and range values. While not as good as the Gaussian model, the fit is reasonable.

It is generally held that spatial prediction is robust against model specification, while the standard error computation is not so robust.

This example investigates the effect of using these different models on the prediction and associated standard errors. First, you use a DATA step to input the thickness data as in the following:

```
data thick;
   input East North Thick @@;
   label Thick='Coal Seam Thickness';
   datalines;
 0.7  59.6  34.1   2.1  82.7  42.2   4.7  75.1  39.5
 4.8  52.8  34.3   5.9  67.1  37.0   6.0  35.7  35.9
 6.4  33.7  36.4   7.0  46.7  34.6   8.2  40.1  35.4
13.3   0.6  44.7  13.3  68.2  37.8  13.4  31.3  37.8
17.8   6.9  43.9  20.1  66.3  37.7  22.7  87.6  42.8
23.0  93.9  43.6  24.3  73.0  39.3  24.8  15.1  42.3
24.8  26.3  39.7  26.4  58.0  36.9  26.9  65.0  37.8
27.7  83.3  41.8  27.9  90.8  43.3  29.1  47.9  36.7
29.5  89.4  43.0  30.1   6.1  43.6  30.8  12.1  42.8
32.7  40.2  37.5  34.8   8.1  43.3  35.3  32.0  38.8
37.0  70.3  39.2  38.2  77.9  40.7  38.9  23.3  40.5
39.4  82.5  41.4  43.0   4.7  43.3  43.7   7.6  43.1
46.4  84.1  41.5  46.7  10.6  42.6  49.9  22.1  40.7
51.0  88.8  42.0  52.8  68.9  39.3  52.9  32.7  39.2
55.5  92.9  42.2  56.0   1.6  42.7  60.6  75.2  40.1
62.1  26.6  40.1  63.0  12.7  41.8  69.0  75.6  40.1
70.5  83.7  40.9  70.9  11.0  41.7  71.5  29.5  39.8
78.1  45.5  38.7  78.2   9.1  41.7  78.4  20.0  40.8
80.5  55.9  38.7  81.1  51.0  38.6  83.8   7.9  41.6
84.5  11.0  41.5  85.2  67.3  39.4  85.5  73.0  39.8
86.7  70.4  39.6  87.2  55.7  38.8  88.1   0.0  41.6
88.4  12.1  41.3  88.4  99.6  41.2  88.8  82.9  40.5
```

```
   88.9   6.2  41.5  90.6   7.0  41.5  90.7  49.6  38.9
   91.5  55.4  39.0  92.9  46.8  39.1  93.4  70.9  39.7
   55.8  50.5  38.1  96.2  84.3  40.3  98.2  58.2  39.5
   ;
```

You run the KRIGE2D procedure on the original Gaussian model, as the following statements show:

```
proc krige2d data=thick outest=pred1 noprint;
   coordinates xc=East yc=North;
   predict var=Thick r=60;
   model scale=7.2881 range=30.6239 form=gauss;
   grid x=0 to 100 by 10 y=0 to 100 by 10;
run;
```

Then, you run the KRIGE2D procedure by using a spherical model with a modified range, using the following statements:

```
proc krige2d data=thick outest=pred2 noprint;
   coordinates xc=East yc=North;
   predict var=Thick r=60;
   model scale=7.5 range=60 form=spherical;
   grid x=0 to 100 by 10 y=0 to 100 by 10;
run;
```

Eventually, you perform the comparison and obtain the comparison results by using the COMPARE and the PRINT procedures as the following statements show:

```
data compare;
   merge pred1(rename=(estimate=g_prd stderr=g_std))
         pred2(rename=(estimate=s_prd stderr=s_std));
   prd_dif=g_prd-s_prd;
   std_dif=g_std-s_std;
run;

proc print data=compare;
   title 'Comparison of Gaussian and Spherical Models';
   title2 'Differences of Predictions and Standard Errors';
   var gxc gyc npoints g_prd s_prd prd_dif g_std s_std std_dif;
run;
```

The predicted values at each of the grid locations do not differ greatly for the two semivariogram models. However, the standard error of prediction for the spherical model is substantially larger than for the Gaussian model. A partial outcome from the first 50 prediction grid points of the comparison analysis is shown in Output 46.1.1.

Output 46.1.1 Comparison of Gaussian and Spherical Models

Comparison of Gaussian and Spherical Models
Differences of Predictions and Standard Errors: First 50 Observations

Obs	GXC	GYC	NPOINTS	g_prd	s_prd	prd_dif	g_std	s_std	std_dif
1	0	0	23	43.9702	42.6700	1.30018	0.63046	2.05947	-1.42901
2	0	10	28	41.7145	41.6780	0.03657	0.50937	2.03464	-1.52526
3	0	20	31	38.9756	39.7285	-0.75291	0.27275	1.93478	-1.66203
4	0	30	33	36.1591	37.2816	-1.12253	0.11363	1.54521	-1.43157
5	0	40	40	33.8340	35.4018	-1.56771	0.04291	1.37653	-1.33361
6	0	50	39	32.8464	34.3835	-1.53711	0.02561	1.22559	-1.19997
7	0	60	36	33.9556	34.3140	-0.35842	0.00168	0.54120	-0.53952
8	0	70	31	36.9217	37.6517	-0.73009	0.03428	1.20363	-1.16935
9	0	80	31	41.1035	41.1016	0.00192	0.04180	0.99544	-0.95364
10	0	90	28	43.6723	42.5216	1.15068	0.09125	1.57357	-1.48232
11	0	100	23	41.8818	42.6511	-0.76939	0.48854	2.20792	-1.71938
12	10	0	25	44.6825	44.1959	0.48655	0.07061	1.09743	-1.02683
13	10	10	31	42.8441	42.7496	0.09449	0.09701	1.46686	-1.36984
14	10	20	35	40.3026	40.3557	-0.05301	0.04555	1.54876	-1.50321
15	10	30	40	37.7583	37.7659	-0.00754	0.00766	0.94135	-0.93369
16	10	40	45	35.6487	35.5495	0.09918	0.00497	0.75917	-0.75420
17	10	50	45	35.0798	34.7083	0.37154	0.01284	1.05027	-1.03743
18	10	60	42	36.0688	35.4784	0.59034	0.01123	1.18270	-1.17147
19	10	70	37	38.1205	38.1052	0.01527	0.00272	0.89156	-0.88884
20	10	80	34	41.2811	41.0803	0.20080	0.02097	1.22754	-1.20657
21	10	90	30	43.2213	42.8904	0.33089	0.05290	1.49438	-1.44148
22	10	100	26	40.9801	43.1350	-2.15488	0.17057	1.93434	-1.76377
23	20	0	29	44.4724	44.4359	0.03655	0.05490	1.23618	-1.18128
24	20	10	36	43.3401	43.2958	0.04428	0.00417	0.95510	-0.95092
25	20	20	40	41.1299	40.9923	0.13757	0.00547	1.18538	-1.17991
26	20	30	44	38.6046	38.5335	0.07118	0.00765	1.08968	-1.08203
27	20	40	50	36.5357	36.5331	0.00258	0.02509	1.33589	-1.31080
28	20	50	50	36.1273	35.8019	0.32543	0.02075	1.31950	-1.29875
29	20	60	50	36.8110	36.5435	0.26748	0.00684	1.11476	-1.10791
30	20	70	40	38.4321	38.5186	-0.08647	0.00217	0.89419	-0.89202
31	20	80	37	41.0639	41.0482	0.01561	0.00645	1.18542	-1.17898
32	20	90	34	43.1765	43.1070	0.06948	0.00524	0.94924	-0.94400
33	20	100	27	42.7637	43.4689	-0.70513	0.06070	1.52094	-1.46024
34	30	0	36	43.4020	43.9436	-0.54159	0.03891	1.32240	-1.28348
35	30	10	40	43.1542	43.1454	0.00879	0.00175	0.72412	-0.72237
36	30	20	45	41.2410	41.2158	0.02520	0.00397	1.10233	-1.09836
37	30	30	53	38.9347	39.0203	-0.08556	0.00391	1.04491	-1.04100
38	30	40	58	37.2798	37.3441	-0.06425	0.00654	0.89225	-0.88571
39	30	50	58	36.7224	36.7485	-0.02611	0.00534	0.83405	-0.82871
40	30	60	56	37.2079	37.3339	-0.12594	0.00561	1.00195	-0.99634
41	30	70	49	38.8800	38.8913	-0.01132	0.00225	1.01429	-1.01204
42	30	80	44	41.0573	41.0664	-0.00910	0.00192	0.97336	-0.97145
43	30	90	37	43.0975	43.0463	0.05114	0.00168	0.51312	-0.51144
44	30	100	30	44.6255	43.3247	1.30077	0.12062	1.57134	-1.45072
45	40	0	37	42.8223	43.5137	-0.69134	0.01588	1.25685	-1.24097
46	40	10	41	42.8953	42.9167	-0.02144	0.00247	0.95163	-0.94916
47	40	20	53	41.1033	41.1816	-0.07829	0.00154	0.96204	-0.96049
48	40	30	61	39.3295	39.2949	0.03453	0.00366	1.05527	-1.05161
49	40	40	68	38.1841	37.9297	0.25443	0.01252	1.27124	-1.25872
50	40	50	68	37.3330	37.4359	-0.10284	0.02764	1.44559	-1.41796

Example 46.2: Data Quality and Prediction with Missing Values

Kriging methods depend primarily on your data. The quantity and quality of your observations are important factors in minimizing prediction errors and increasing accuracy in your prediction analysis.

A typical aspect of data quality is measurement accuracy. In principle, the accuracy level of your data is not a parameter in kriging prediction; kriging assumes by definition that your data are perfectly accurate (hard) measurements. Whether you accept this assumption depends on your application. For example, an instrumentation error of $\pm 1\%$ in the data values might be regarded as considerable in one case, whereas the same level of uncertainty might be trivial within a different framework. Your experience and judgment are crucial when you consider whether observations in a data set might be too noisy for kriging predictions to be useful.

A second aspect of data quality involves the spatial arrangement of your observations. You need to have a sufficient number of observations in order to perform spatial prediction. Also, a key element in minimizing prediction errors is an adequate sampling density. Interpretation of the expressions "sufficient number" and "adequate sampling" is again case-specific. In any event, you want enough measurements so that you can deduce the underlying spatial correlation in the working domain; see also the discussion in the section "Choosing the Size of Classes" on page 7555 in the VARIOGRAM procedure.

This example focuses on the effects of different sampling densities on the prediction analysis. The demonstration is a slight variation of the example in the section "Getting Started: KRIGE2D Procedure" on page 2931. Specifically, you use the same correlation structure and prediction grid. However, the thick data set, is modified as follows: three values in the central area of the grid are assumed missing, namely the observation values at locations $\boldsymbol{s}_1 = (x_1, y_1) = (55.8, 50.5)$, $\boldsymbol{s}_2 = (x_2, y_2) = (52.8, 68.9)$, and $\boldsymbol{s}_3 = (x_3, y_3) = (52.9, 32.7)$. These locations have been selected so that an extended area without observations is created in the domain. The following DATA step is the input for the modified thick data set:

```
data thick;
   input East North Thick @@;
   label Thick='Coal Seam Thickness';
   datalines;
    0.7  59.6  34.1   2.1  82.7  42.2   4.7  75.1  39.5
    4.8  52.8  34.3   5.9  67.1  37.0   6.0  35.7  35.9
    6.4  33.7  36.4   7.0  46.7  34.6   8.2  40.1  35.4
   13.3   0.6  44.7  13.3  68.2  37.8  13.4  31.3  37.8
   17.8   6.9  43.9  20.1  66.3  37.7  22.7  87.6  42.8
   23.0  93.9  43.6  24.3  73.0  39.3  24.8  15.1  42.3
   24.8  26.3  39.7  26.4  58.0  36.9  26.9  65.0  37.8
   27.7  83.3  41.8  27.9  90.8  43.3  29.1  47.9  36.7
   29.5  89.4  43.0  30.1   6.1  43.6  30.8  12.1  42.8
   32.7  40.2  37.5  34.8   8.1  43.3  35.3  32.0  38.8
   37.0  70.3  39.2  38.2  77.9  40.7  38.9  23.3  40.5
   39.4  82.5  41.4  43.0   4.7  43.3  43.7   7.6  43.1
   46.4  84.1  41.5  46.7  10.6  42.6  49.9  22.1  40.7
   51.0  88.8  42.0  52.8  68.9     .  52.9  32.7     .
```

```
   55.5  92.9  42.2  56.0   1.6  42.7  60.6  75.2  40.1
   62.1  26.6  40.1  63.0  12.7  41.8  69.0  75.6  40.1
   70.5  83.7  40.9  70.9  11.0  41.7  71.5  29.5  39.8
   78.1  45.5  38.7  78.2   9.1  41.7  78.4  20.0  40.8
   80.5  55.9  38.7  81.1  51.0  38.6  83.8   7.9  41.6
   84.5  11.0  41.5  85.2  67.3  39.4  85.5  73.0  39.8
   86.7  70.4  39.6  87.2  55.7  38.8  88.1   0.0  41.6
   88.4  12.1  41.3  88.4  99.6  41.2  88.8  82.9  40.5
   88.9   6.2  41.5  90.6   7.0  41.5  90.7  49.6  38.9
   91.5  55.4  39.0  92.9  46.8  39.1  93.4  70.9  39.7
   55.8  50.5     .  96.2  84.3  40.3  98.2  58.2  39.5
   ;

title 'Coal Seam Thickness Kriging';
ods graphics on;
```

Note that here you assume prior knowledge of the correlation structure model, because its parameters are based on the complete thick data set. A covariance model extracted from the incomplete set with the missing values would be a covariance model coming from a different data set; hence, it is likely to have different parameters.

After you define the modified data set, you run PROC KRIGE2D and request the OBSERVATIONS plot with the SHOWMISSING suboption. You also request two instances of the PREDICTION plot: one that displays the prediction surface and contours, and another that plots the kriging standard error surface and contours. In both of these PREDICTION plots you specify that the observations be shown as gradient markers with outlines. The following statements compute the kriged predictions and produce the requested graphics:

```
proc krige2d data=thick outest=predictions
             plots(only)=(observ(showmissing)
                          pred(fill=pred line=pred obs=linegrad)
                          pred(fill=se line=se obs=linegrad));
   coordinates xc=East yc=North;
   predict var=Thick r=60;
   model scale=7.2881 range=30.6239 form=gauss;
   grid x=0 to 100 by 2.5 y=0 to 100 by 2.5;
run;

ods graphics off;
```

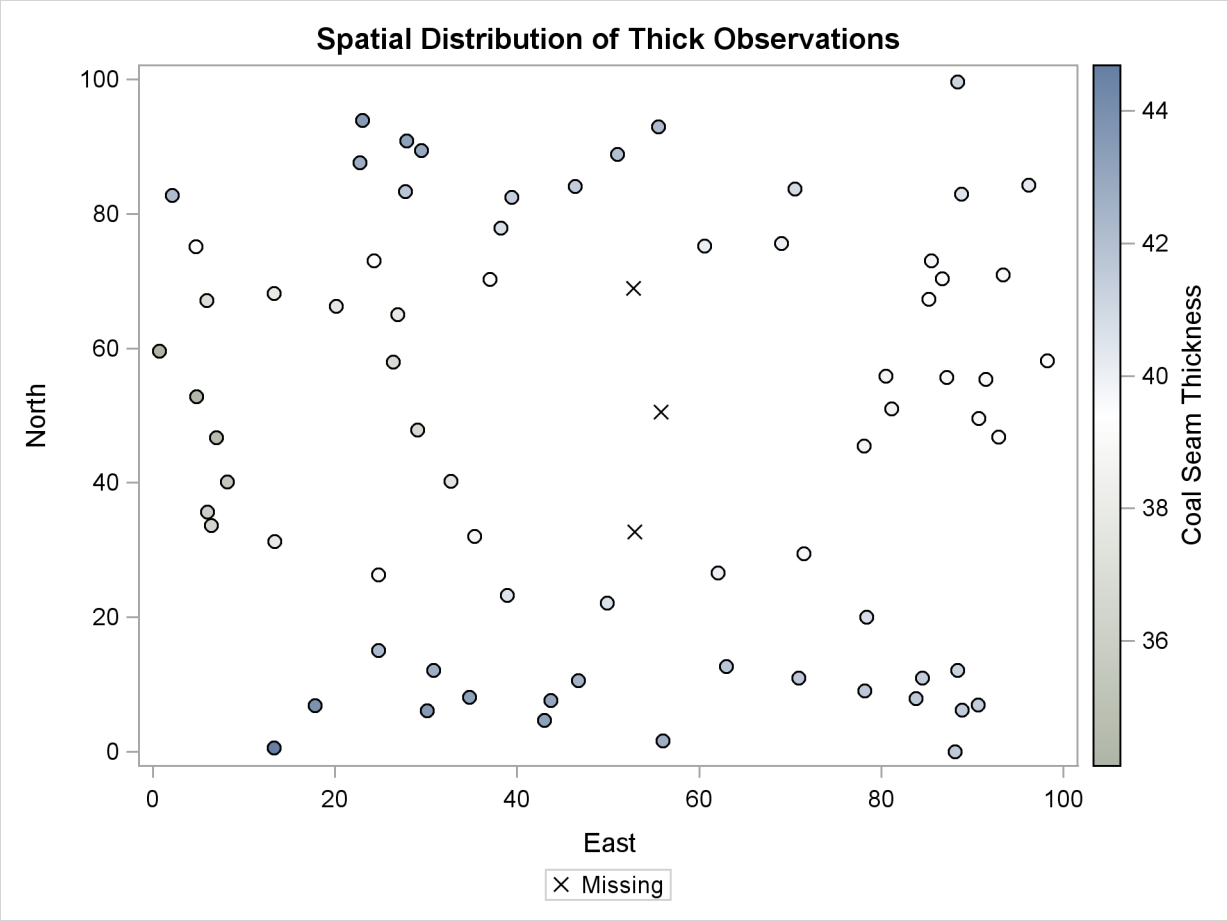
The number of observations table indicates the three missing values in Output 46.2.1.

Output 46.2.1 Number of Observations for the Modified thick Data Set

```
                    Coal Seam Thickness Kriging

                      The KRIGE2D Procedure
                    Dependent Variable: Thick

                Number of Observations Read          75
                Number of Observations Used          72
```

Output 46.2.2 is a map of the scatter plot of the modified observed data. The SHOWMISSING suboption produces marks in the observations plot that conveniently indicate the locations s_1, s_2, and s_3 of the missing values. Consequently, there is now an extended area without any Thick observed values in the central part of the domain.

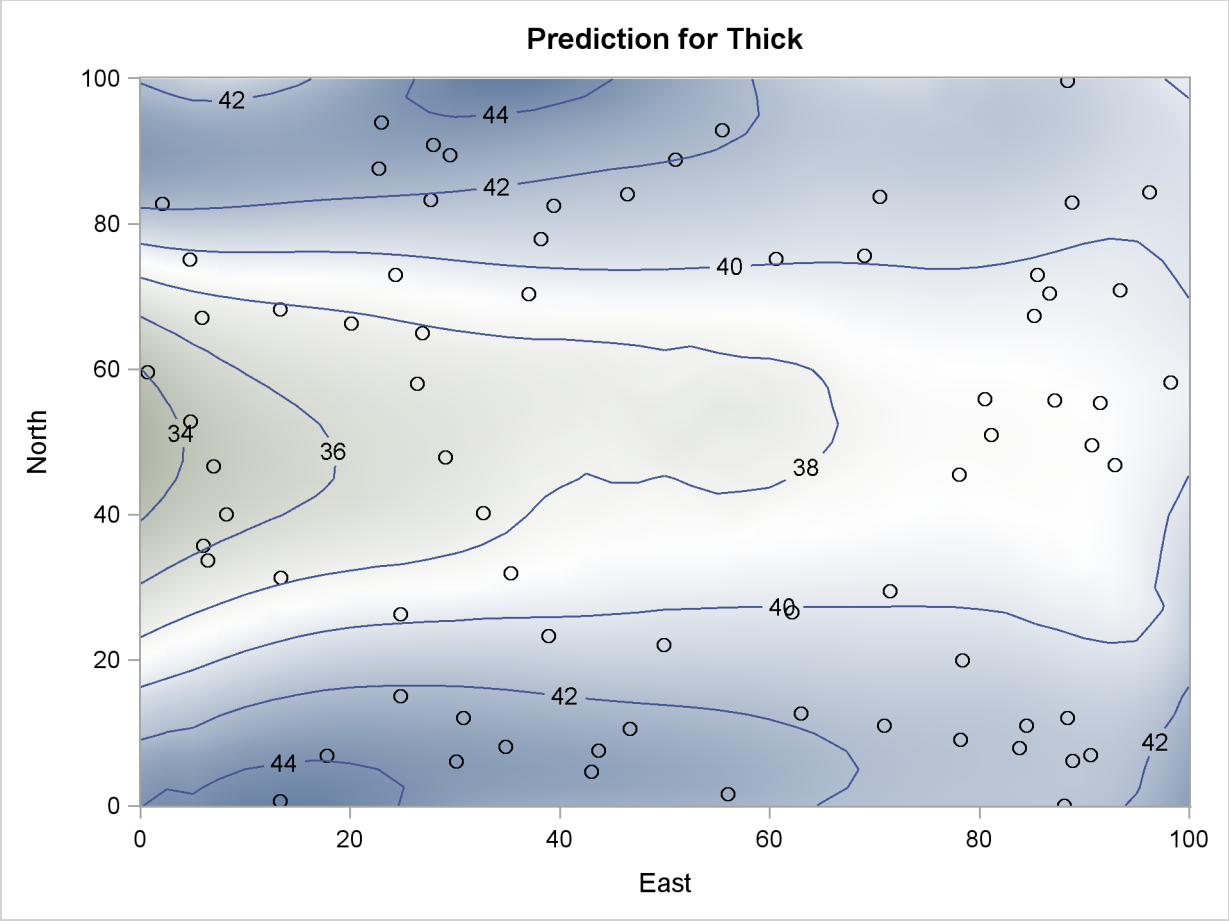
Output 46.2.2 Scatter Plot of the Observations Spatial Distribution



Predictions at grid points with few neighboring data points rely heavily on the underlying covariance structure. The covariance model has a range of about 30,000 feet, which suggests that within this range a grid point might have no data neighbors at all and still obtain a prediction value on the basis of the correlation structure alone. This type of behavior is demonstrated in the Output 46.2.3, which shows a circular region in the center of the plot where there are no data points. Predictions at the nodes in this area are mostly influenced by the covariance structure.

You can see the impact of this effect on the predictions if you compare the prediction contours in the Output 46.2.3 to the ones in Figure 46.4. Despite the contribution of the neighboring Thick data values to the predictions within the area of no observations, the outcome is clearly altered by the absence of observations at the locations s_1, s_2, and s_3.

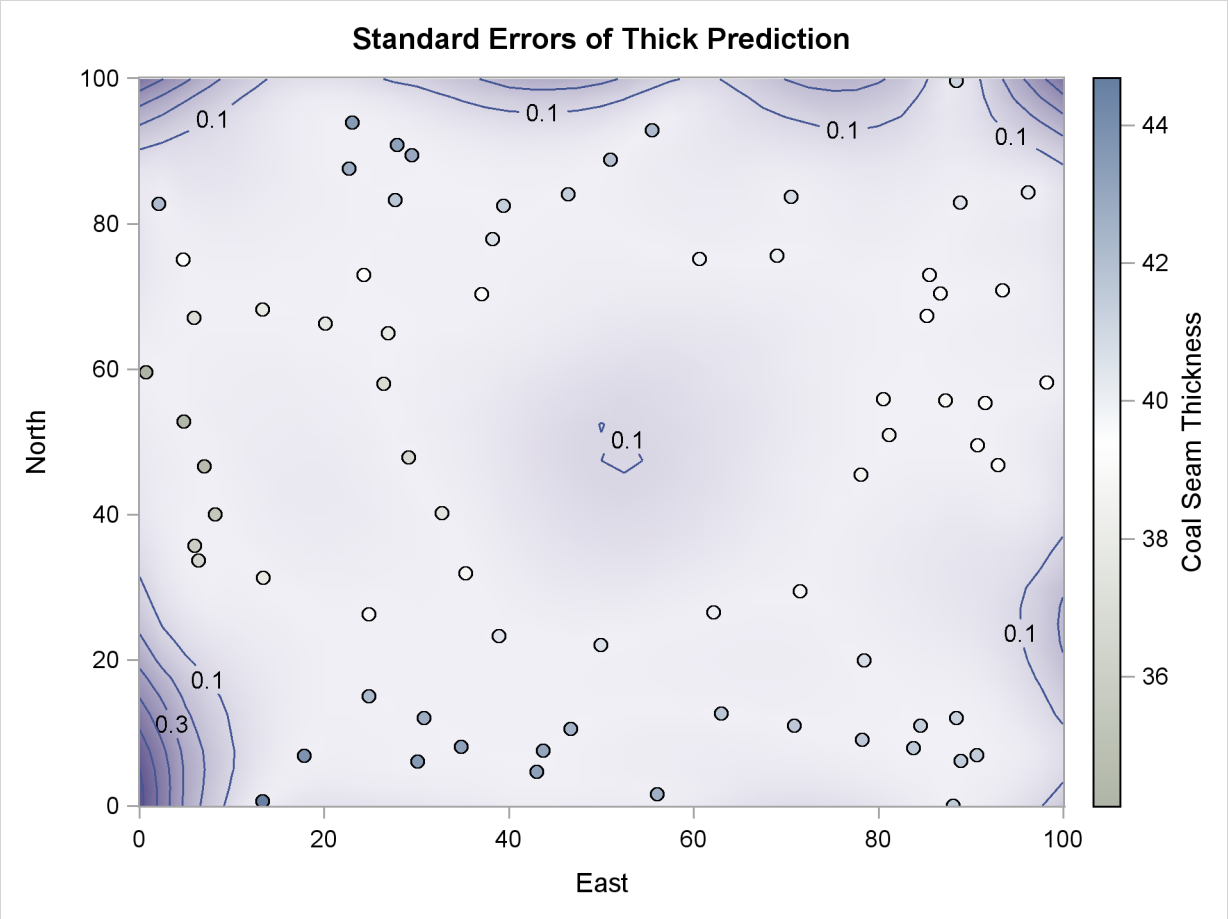
Output 46.2.3 Surface Plot and Contours of Kriged Coal Seam Thickness



A noticeable difference is also apparent in the plot of the prediction standard errors. Output 46.2.4 displays these errors, and you can compare it to the standard error surface in Figure 46.4. The comparison shows a slight difference in the color gradient within the area of the missing data values. Output 46.2.4 uses standard error contours to enhance the effect of this difference.

The lack of information from the removed data resulted in an increase of the prediction uncertainty at the grid nodes that are most remotely situated from any observation in the central part of the domain. According to Output 46.2.4, the standard error at these nodes is almost comparable to the error observed near the borders of the domain, where the nodes of the prediction grid have relatively fewer data neighbors than other nodes in the domain.

Output 46.2.4 Surface Plot and Contours of Prediction Standard Errors



Note that PREDICTION plots display only observations with nonmissing values, as the plots in Output 46.2.3 and Output 46.2.4 demonstrate.

References

Christakos, G. (1992), *Random Field Models in Earth Sciences*, New York: Academic Press.

Cressie, N. A. C. (1993), *Statistics for Spatial Data*, New York: John Wiley & Sons.

Deutsch, C. V. and Journel, A. G. (1992), *GSLIB: Geostatistical Software Library and User's Guide*, New York: Oxford University Press.

Hohn, M. (1988), *Geostatistics and Petroleum Geology*, New York: Van Nostrand Reinhold.

Isaaks, E. H. and Srivastava, R. M. (1988), "Spatial Continuity Measures for Probabilistic and Deterministic Geostatistics," *Mathematical Geology*, 20, 313–341.

Journel, A. G. (1985), "The Deterministic Side of Geostatistics," *Mathematical Geology*, 17, 1–15.

Journel, A. G. and Huijbregts, C. J. (1978), *Mining Geostatistics*, New York: Academic Press.

Journel, A. G. and Rossi, M. (1989), "When Do We Need a Trend Model in Kriging?" *Mathematical Geology*, 21, 715–739.

Matheron, G. (1971), *The Theory of Regionalized Variables and Its Applications*, Les Cahiers du Centre de Morphologie Mathematique de Fontainebleau.

Robinson, G. K. (1991), "That BLUP Is a Good Thing: The Estimation of Random Effects," *Statistical Science*, 6, 15–51.

Chapter 47

The LATTICE Procedure

Contents

Overview: LATTICE Procedure **2979**
Getting Started: LATTICE Procedure **2980**
Syntax: LATTICE Procedure **2982**
PROC LATTICE Statement 2982
BY Statement 2983
VAR Statement 2983
Input Data Set 2983
Missing Values 2984
Displayed Output 2984
ODS Table Names 2985
Example: LATTICE Procedure **2985**
Example 47.1: Analysis of Variance through PROC LATTICE 2985
References: LATTICE Procedure **2989**

Overview: LATTICE Procedure

The LATTICE procedure computes the analysis of variance and analysis of simple covariance for data from an experiment with a lattice design. PROC LATTICE analyzes balanced square lattices, partially balanced square lattices, and some rectangular lattices.

In balanced square lattices, the number of treatments is equal to the square of the number of units per block. Incomplete blocks are grouped to form mutually orthogonal replications. The number of replicates in the basic plan is always 1 plus the number of units per block.

Partially balanced square lattices are similar to balanced lattices, although the number of replicates can vary. Partially balanced designs are constructed of the replicates in the basic plan, but not all replicates are included the same number of times, and some might not be included at all.

In rectangular lattices, there are k units per block and k (k +1) treatments. As in square lattices, blocks are grouped to form mutually orthogonal replicates in the basic plan. PROC LATTICE can analyze simple rectangular lattices (two orthogonal replications) and triple rectangular lattices (three orthogonal replications). The experiment can include several repetitions of the basic plan.

The LATTICE procedure determines from the data set which type of design has been used. It also checks to see whether the design is valid and displays an appropriate message if it is not.

Getting Started: LATTICE Procedure

An example of a balanced square design is an experiment to investigate the effects of nine diets on the growth rate of pigs.

In some breeds of pigs, past experience has shown that a large part of the total variation in growth rates between animals can be attributed to the litter. Therefore, this experiment is planned so that litter differences do not contribute to the intrablock error.

First, the pigs are separated into sets of three litter-mates. Each block is assigned two sets of the three litter-mates. In a given block, one pig from each set receives a diet. Therefore, the experimental unit is a pair of pigs feeding in a particular pen on one of the nine diets. The response variable, growth rate, is the sum of the growth rates for the two pigs in a particular pen. To get the adjusted diet mean per pig, the adjusted treatment mean for the pen must be divided by 2.

The special numeric SAS variables named Group, Block, Treatmnt, and Rep must be used to define the design. In this example, the Treatmnt variable ranges from 1 to 9 and indicates the particular diet. The Block variable is 1, 2, or 3 and indicates the pen containing the two pigs. The Group variable ranges from 1 to 4 and specifies which replication within the basic plan includes the experimental unit. In this example, you would not use the Rep variable since the entire basic plan is not replicated.

You can use the following DATA step and PROC LATTICE statement to analyze this experiment. The response variable is Weight .

```
title 'Examining the Growth Rate of Pigs';
data Pigs;
   input Group Block Treatmnt Weight @@;
   datalines;
1 1 1 2.20  1 1 2 1.84  1 1 3 2.18
1 2 4 2.05  1 2 5 0.85  1 2 6 1.86
1 3 7 0.73  1 3 8 1.60  1 3 9 1.76
2 1 1 1.19  2 1 4 1.20  2 1 7 1.15
2 2 2 2.26  2 2 5 1.07  2 2 8 1.45
2 3 3 2.12  2 3 6 2.03  2 3 9 1.63
3 1 1 1.81  3 1 5 1.16  3 1 9 1.11
3 2 2 1.76  3 2 6 2.16  3 2 7 1.80
3 3 3 1.71  3 3 4 1.57  3 3 8 1.13
4 1 1 1.77  4 1 6 1.57  4 1 8 1.43
4 2 2 1.50  4 2 4 1.60  4 2 9 1.42
4 3 3 2.04  4 3 5 0.93  4 3 7 1.78
;

title 'Examining the Growth Rate of Pigs';
proc lattice data=Pigs;
   var Weight;
run;
```

The SAS code produces the output shown in Figure 47.1.

Figure 47.1 Output from Example LATTICE Procedure

Examining the Growth Rate of Pigs

The Lattice Procedure

Analysis of Variance for Weight

Source	DF	Sum of Squares	Mean Square
Replications	3	0.07739	0.02580
Blocks within Replications (Adj.)	8	1.4206	0.1776
Component B	8	1.4206	0.1776
Treatments (Unadj.)	8	3.2261	0.4033
Intra Block Error	16	1.2368	0.07730
Randomized Complete Block Error	24	2.6574	0.1107
Total	35	5.9609	0.1703

Additional Statistics for Weight

Variance of Means in Same Block	0.04593
LSD at .01 Level	0.6259
LSD at .05 Level	0.4543
Efficiency Relative to RCBD	120.55

Adjusted Treatment Means for Weight

Treatment	Mean
1	1.8035
2	1.7544
3	1.9643
4	1.7267
5	0.9393
6	1.8448
7	1.3870
8	1.4347
9	1.5004

Diet 3 yields the highest mean growth rate at 1.9643 pounds for the two pigs (0.9822 per pig), while diet 5 has the lowest rate at 0.9393 (0.4696 per pig). The efficiency of the experiment relative to a randomized complete block design is 120.55 percent, so using the lattice design increased precision, producing more accurate estimates of the treatment effects. The different elements of the LATTICE procedure's output are discussed in the "Displayed Output" on page 2984 section.

Syntax: LATTICE Procedure

The following statements are available in PROC LATTICE.

PROC LATTICE < *options* > **;**
BY *variables* **;**
VAR *variables* **;**

Three specific numeric SAS variables, Group, Block, and Treatmnt, *must* be present in the data set to which PROC LATTICE is applied. A fourth numeric variable named Rep must be present when the design involves repetition of the entire basic plan. (See the "Input Data Set" on page 2983 section for more information.)

Every numeric variable other than Group, Block, Treatmnt, or Rep in the input SAS data set may be considered a response variable. A VAR statement tells PROC LATTICE that only the variables listed in the VAR statement are to be considered response variables. If the VAR statement is omitted, then all numeric variables, excluding Group, Block, Treatmnt, and Rep, are considered response variables. PROC LATTICE performs an analysis for each response variable.

PROC LATTICE Statement

PROC LATTICE < *options* > **;**

You can specify the following options in the PROC LATTICE statement.

DATA=*SAS-data-set*
names the SAS data set to be used by PROC LATTICE. If you omit the DATA= option, the most recently created SAS data set is used.

COVARIANCE
COV
calculates sums of products for every possible pair of response variables. A sum of products is given for each source of variation in the analysis of variance table. For each pair of response variables, the one appearing later in the data set (or in the VAR statement) is the covariable.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC LATTICE to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in the order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the LATTICE procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the response variables. If you do not include a VAR statement, all numeric variables in the data set are considered response variables (except Group, Block, Treatmnt, and Rep).

Input Data Set

Four numeric SAS variables, Group, Block, Treatmnt, and Rep, are used in the input data set to define the lattice design. The Group, Block, and Treatmnt variables are required in the data set to which PROC LATTICE is applied. The Rep variable must be present when the design involves repetition of the entire basic plan.

Group specifies which orthogonal replication in the basic plan includes the experimental unit. Values of Group must be $1, 2, \ldots, n$, where n is the number of replicates in the basic plan.

Block specifies the block in which the experimental unit is present. Values of Block must be $1, 2, \ldots, m$, where m is the number of blocks in a replication.

Treatmnt specifies which treatment was applied to the experimental unit. Values of Treatmnt must be $1, 2, \ldots, i$, where i is the number of treatments in a replication.

Rep specifies which repetition of the basic plan includes the experimental unit. Values of Rep must be $1, 2, \ldots, p$, where p is the number of replications of the entire basic plan. Thus, the experiment has a total of np replicates.

Missing Values

If a value of Group, Block, Treatmnt, or Rep is missing, the analysis is not performed and an appropriate error message is displayed.

If a value of a response variable is missing, this entire variable is dropped from the analysis. If other response variables exist that do not have missing values, they are analyzed.

Displayed Output

For each response variable, PROC LATTICE displays the following

- an "Analysis of Variance" table and related statistics, including the following as separate sources of variations:
  - Replications
  - Blocks within Replications (adjusted for treatments)
  - Treatments (unadjusted)
  - Intra-block Error
  - Randomized Complete Block Error

  The Blocks within Replications sum of squares is further broken down into "Component A" and "Component B." If there is no repetition of the basic plan, the Component B sum of squares is the same as the Blocks within Replications sum of squares. If there is repetition of the basic plan, the Component A sum of squares reflects the variation among blocks that contain the same treatments.

  The source of variation called Randomized Complete Block Error is the sum of the Blocks within Replications sum of squares and the Intra-block Error sum of squares. It is the appropriate error term if the experimental design is a randomized complete block design, with the replications filling the roles of complete blocks.

- two values for the Variance of Means. For some lattice designs, these are only approximations. The first value is applicable when the two treatments appear in the same block; the other (when it appears) applies when the two treatments never appear in the same block (a possibility in partially balanced and rectangular designs).

- an Average of Variance. Except with small designs, it is sufficient to use this average variance of means for tests between treatments (whether the two treatments appear in the same block or not); see Cochran and Cox (1957).
- the Least Significant Differences (LSDs) at the 0.01 and 0.05 levels of significance, based on the Average of Variance
- Efficiency Relative to RCBD, the efficiency of the lattice design relative to a randomized complete block design. The efficiency is the ratio of the randomized complete block mean squared error to the effective error variance; see Cochran and Cox (1957).
- the Adjusted Treatment Means. These are adjusted for blocks if the relative precision is greater than 105%.

When you specify the COVARIANCE option, PROC LATTICE produces sums of products and the mean product for each source of variation in the analysis of variance table.

ODS Table Names

PROC LATTICE assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 47.1. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 47.1 ODS Tables Produced by PROC LATTICE

ODS Table Name	Description	PROC LATTICE Option
ANOVA	Analysis of variance	default
AdjTreatmentMeans	Adjusted treatment means	default
Statistics	Additional statistics	default

Example: LATTICE Procedure

Example 47.1: Analysis of Variance through PROC LATTICE

In the following example, from Cochran and Cox (1957, p. 406), the data are yields (Yield) in bushels per acre of 25 varieties (Treatmnt) of soybeans. The data are collected in two replications (Group) of 25 varieties in five blocks (Block) containing five varieties each. This is an example of a partially balanced square lattice design.

```
data Soy;
  do Group = 1 to 2;
    do Block = 1 to 5;
      do Plot = 1 to 5;
         input Treatmnt Yield @@;
         output;
      end;
    end;
  end;
  drop Plot;
  datalines;
 1   6   2   7   3   5   4   8   5   6
 6   16  7   12  8  12   9   13  10  8
11  17  12  7   13  7   14  9   15  14
16 18   17  16  18  13  19  13  20  14
21  14  22  15  23  11  24  14  25  14
 1   24  6   13  11  24  16  11  21  8
 2   21  7   11  12  14  17  11  22  23
 3   16  8   4   13  12  18  12  23  12
 4   17  9   10  14  30  19  9   24  23
 5   15  10  15  15  22  20  16  25  19
;

proc print data=Soy;
   id Treatmnt;
run;

proc lattice data=Soy;
run;
```

The results from these statements are shown in Output 47.1.1 and Output 47.1.2.

Output 47.1.1 Displayed Output from PROC PRINT

Obs	Group	Block	Treatmnt	Yield
1	1	1	1	6
2	1	1	2	7
3	1	1	3	5
4	1	1	4	8
5	1	1	5	6
6	1	2	6	16
7	1	2	7	12
8	1	2	8	12
9	1	2	9	13
10	1	2	10	8
11	1	3	11	17
12	1	3	12	7
13	1	3	13	7
14	1	3	14	9
15	1	3	15	14
16	1	4	16	18
17	1	4	17	16
18	1	4	18	13
19	1	4	19	13
20	1	4	20	14
21	1	5	21	14
22	1	5	22	15
23	1	5	23	11
24	1	5	24	14
25	1	5	25	14
26	2	1	1	24
27	2	1	6	13
28	2	1	11	24
29	2	1	16	11
30	2	1	21	8
31	2	2	2	21
32	2	2	7	11
33	2	2	12	14
34	2	2	17	11
35	2	2	22	23
36	2	3	3	16
37	2	3	8	4
38	2	3	13	12
39	2	3	18	12
40	2	3	23	12
41	2	4	4	17
42	2	4	9	10
43	2	4	14	30
44	2	4	19	9
45	2	4	24	23
46	2	5	5	15
47	2	5	10	15
48	2	5	15	22
49	2	5	20	16
50	2	5	25	19

Output 47.1.2 Displayed Output from PROC LATTICE

The Lattice Procedure

Analysis of Variance for Yield

Source	DF	Sum of Squares	Mean Square
Replications	1	212.18	212.18
Blocks within Replications (Adj.)	8	501.84	62.7300
Component B	8	501.84	62.7300
Treatments (Unadj.)	24	559.28	23.3033
Intra Block Error	16	218.48	13.6550
Randomized Complete Block Error	24	720.32	30.0133
Total	49	1491.78	30.4445

Additional Statistics for Yield

Variance of Means in Same Block	15.7915
Variance of Means in Different Bloc	17.9280
Average of Variance	17.2159
LSD at .01 Level	12.1189
LSD at .05 Level	8.7959
Efficiency Relative to RCBD	174.34

Adjusted Treatment Means for Yield

Treatment	Mean
1	19.0681
2	16.9728
3	14.6463
4	14.7687
5	12.8470
6	13.1701
7	9.0748
8	6.7483
9	8.3707
10	8.4489
11	23.5511
12	12.4558
13	12.6293
14	20.7517
15	19.3299
16	12.6224
17	10.5272
18	10.7007
19	7.3231
20	11.4013
21	11.6259
22	18.5306
23	12.2041
24	17.3265
25	15.4048

The efficiency of the experiment relative to a randomized complete block design is 174.34%. Precision is gained using the lattice design via the recovery of intra-block error information, enabling more accurate estimates of the treatment effects. Variety 8 of soybean had the lowest adjusted treatment mean (6.7483 bushels per acre), while variety 11 of soybean had the highest adjusted treatment mean (23.5511 bushels per acre).

References: LATTICE Procedure

Cochran, W.G. and Cox, G.M. (1957), *Experimental Designs* , Second Edition, New York: John Wiley & Sons, Inc.

Comstock, R.E., Peterson, W.J., and Stewart, H.A. (1948), "An Application of the Balanced Lattice Design in a Feeding Trial with Swine," *Journal of Animal Science* , 7, 320–331.

Cornelius, P.L. (1983), "Lattice Designs," *Encyclopedia of Statistical Sciences* , 4, 510–518.

Robinson, H.F. and Watson, G.S. (1949), "Analysis of Simple and Triple Rectangular Designs," *North Carolina Agricultural Experiment Station Technical Bulletin* , 88.

Chapter 48

The LIFEREG Procedure

Contents

Overview: LIFEREG Procedure **2992**
Getting Started: LIFEREG Procedure **2995**
Modeling Right-Censored Failure Time Data 2995
Bayesian Analysis of Right-Censored Data 2999
Syntax: LIFEREG Procedure **3005**
PROC LIFEREG Statement 3006
BAYES Statement 3007
BY Statement 3017
CLASS Statement 3017
INSET Statement 3018
MODEL Statement 3020
OUTPUT Statement 3025
PROBPLOT Statement 3027
WEIGHT Statement 3036
Details: LIFEREG Procedure **3036**
Missing Values 3036
Model Specification 3036
Computational Method 3037
Supported Distributions 3039
Predicted Values 3043
Confidence Intervals 3044
Fit Statistics 3045
Probability Plotting 3046
INEST= Data Set 3050
OUTEST= Data Set 3051
XDATA= Data Set 3051
Computational Resources 3052
Bayesian Analysis 3053
Displayed Output for Classical Analysis 3056
Displayed Output for Bayesian Analysis 3058
ODS Table Names 3060
ODS Graphics 3062
Examples: LIFEREG Procedure **3063**
Example 48.1: Motorette Failure 3063

Example 48.2: Computing Predicted Values for a Tobit Model 3069
Example 48.3: Overcoming Convergence Problems by Specifying Initial Values 3073
Example 48.4: Analysis of Arbitrarily Censored Data with Interaction Effects 3078
Example 48.5: Probability Plotting—Right Censoring 3083
Example 48.6: Probability Plotting—Arbitrary Censoring 3085
Example 48.7: Bayesian Analysis of Clinical Trial Data 3088
References **3096**

Overview: LIFEREG Procedure

The LIFEREG procedure fits parametric models to failure time data that can be uncensored, right censored, left censored, or interval censored. The models for the response variable consist of a linear effect composed of the covariates and a random disturbance term. The distribution of the random disturbance can be taken from a class of distributions that includes the extreme value, normal, logistic, and, by using a log transformation, the exponential, Weibull, lognormal, loglogistic, and three-parameter gamma distributions.

The model assumed for the response $\mathbf{y}$ is

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \sigma\boldsymbol{\epsilon}$$

where $\mathbf{y}$ is a vector of response values, often the log of the failure times, $\mathbf{X}$ is a matrix of covariates or independent variables (usually including an intercept term), $\boldsymbol{\beta}$ is a vector of unknown regression parameters, σ is an unknown scale parameter, and $\boldsymbol{\epsilon}$ is a vector of errors assumed to come from a known distribution (such as the standard normal distribution). If an offset variable O is specified, the form of the model is $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{O} + \sigma\boldsymbol{\epsilon}$, where $\mathbf{O}$ is a vector of values of the offset variable O. The distribution might also depend on additional shape parameters. These models are equivalent to accelerated failure time models when the log of the response is the quantity being modeled. The effect of the covariates in an accelerated failure time model is to change the scale, and not the location, of a baseline distribution of failure times.

The LIFEREG procedure estimates the parameters by maximum likelihood with a Newton-Raphson algorithm. PROC LIFEREG estimates the standard errors of the parameter estimates from the inverse of the observed information matrix.

The accelerated failure time model assumes that the effect of independent variables on an event time distribution is multiplicative on the event time. Usually, the scale function is $\exp(\mathbf{x}_c'\boldsymbol{\beta}_c)$, where $\mathbf{x}_c$ is the vector of covariate values (not including the intercept term) and $\boldsymbol{\beta}_c$ is a vector of unknown parameters. Thus, if T_0 is an event time sampled from the baseline distribution corresponding to values of zero for the covariates, then the accelerated failure time model specifies that, if the vector of covariates is $\mathbf{x}_c$, the event time is $T = \exp(\mathbf{x}_c'\boldsymbol{\beta}_c)T_0$. If $y = \log(T)$ and $y_0 = \log(T_0)$, then

$$y = \mathbf{x}_c'\boldsymbol{\beta}_c + y_0$$

This is a linear model with y_0 as the error term.

In terms of survival or exceedance probabilities, this model is

$$\Pr(T > t \mid \mathbf{x}_c) = \Pr(T_0 > \exp(-\mathbf{x}_c'\boldsymbol{\beta}_c)t)$$

The probability on the left-hand side of the equal sign is evaluated given the value $\mathbf{x}_c$ for the covariates, and the right-hand side is computed using the baseline probability distribution but at a scaled value of the argument. The right-hand side of the equation represents the value of the baseline survival function evaluated at $\exp(-\mathbf{x}_c'\boldsymbol{\beta}_c)t$.

Models usually have an intercept parameter and a scale parameter. In terms of the original untransformed event times, the effects of the intercept term and the scale term are to scale the event time and to raise the event time to a power, respectively. That is, if

$$\log(T_0) = \mu + \sigma \log(T_\epsilon)$$

then

$$T_0 = \exp(\mu) T_\epsilon^\sigma$$

Although it is possible to fit these models to the original response variable by using the NOLOG option, it is more common to model the log of the response variable. Because of this log transformation, zero values for the observed failure times are not allowed unless the NOLOG option is specified. Similarly, small values for the observed failure times lead to large negative values for the transformed response. The NOLOG option should be used only if you want to fit a distribution appropriate for the untransformed response, such as the extreme value instead of the Weibull. If you specify the normal or logistic distributions, the responses are not log transformed; that is, the NOLOG option is implicitly assumed.

Parameter estimates for the normal distribution are sensitive to large negative values, and care must be taken that the fitted model is not unduly influenced by them. Large negative values for the normal distribution can occur when fitting the lognormal distribution by log transforming the response, and some response values are near zero. Likewise, values that are extremely large after the log transformation have a strong influence in fitting the Weibull distribution (that is, the extreme value distribution for log responses). You should examine the residuals and check the effects of removing observations with large residuals or extreme values of covariates on the model parameters. The logistic distribution gives robust parameter estimates in the sense that the estimates have a bounded influence function.

The standard errors of the parameter estimates are computed from large sample normal approximations by using the observed information matrix. In small samples, these approximations might be poor. Refer to Lawless (2003) for additional discussion and references. You can sometimes construct better confidence intervals by transforming the parameters. For example, large sample theory is often more accurate for $\log(\sigma)$ than σ. Therefore, it might be more accurate to construct confidence intervals for $\log(\sigma)$ and transform these into confidence intervals for σ. The parameter estimates and their estimated covariance matrix are available in an output SAS data set and can be used to construct additional tests or confidence intervals for the parameters. Alternatively, tests of parameters can be based on log-likelihood ratios. Refer to Cox and Oakes (1984) for a discussion of the merits of some possible test methods including score, Wald, and likelihood ratio tests. Likelihood ratio tests are generally more reliable for small samples than tests based on the information matrix.

The log-likelihood function is computed using the log of the failure time as a response. This log likelihood differs from the log likelihood obtained using the failure time as the response by an additive term of $\sum \log(t_i)$, where the sum is over the uncensored failure times. This term does not depend on the unknown parameters and does not affect parameter or standard error estimates. However, many published values of log likelihoods use the failure time as the basic response variable and, hence, differ by the additive term from the value computed by the LIFEREG procedure.

The classic Tobit model also fits into this class of models but with data usually censored on the left. The data considered by Tobin (1958) in his original paper came from a survey of consumers where the response variable is the ratio of expenditures on durable goods to the total disposable income. The two explanatory variables are the age of the head of household and the ratio of liquid assets to total disposable income. Because many observations in this data set have a value of zero for the response variable, the model fit by Tobin is

$$\mathbf{y} = \max(\mathbf{x}'\boldsymbol{\beta} + \epsilon, 0)$$

which is a regression model with left censoring, where $\mathbf{x}' = (1, \mathbf{x}_c')$.

Bayesian analysis of parametric survival models can be requested by using the BAYES statement in the LIFEREG procedure. In Bayesian analysis, the model parameters are treated as random variables, and inference about parameters is based on the posterior distribution of the parameters, given the data. The posterior distribution is obtained using Bayes' theorem as the likelihood function of the data weighted with a prior distribution. The prior distribution enables you to incorporate knowledge or experience of the likely range of values of the parameters of interest into the analysis. If you have no prior knowledge of the parameter values, you can use a noninformative prior distribution, and the results of the Bayesian analysis will be very similar to a classical analysis based on maximum likelihood. A closed form of the posterior distribution is often not feasible, and a Markov chain Monte Carlo method by Gibbs sampling is used to simulate samples from the posterior distribution. See Chapter 7, "Introduction to Bayesian Analysis Procedures," for an introduction to the basic concepts of Bayesian statistics. Also see the section "Bayesian Analysis: Advantages and Disadvantages" on page 149 for a discussion of the advantages and disadvantages of Bayesian analysis. Refer to Ibrahim, Chen, and Sinha (2001) and Gilks, Richardson, and Spiegelhalter (1996) for more information about Bayesian analysis, including guidance in choosing prior distributions.

For Bayesian analysis, PROC LIFEREG generates a Gibbs chain for the posterior distribution of the model parameters. Summary statistics (mean, standard deviation, quartiles, HPD and credible intervals, correlation matrix) and convergence diagnostics (autocorrelations; Gelman-Rubin, Geweke, Raftery-Lewis, and Heidelberger and Welch tests; and the effective sample size) are computed for each parameter, as well as the correlation matrix of the posterior sample. Trace plots, posterior density plots, and autocorrelation function plots that are created using ODS Graphics are also provided for each parameter.

The LIFEREG procedure now uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

Getting Started: LIFEREG Procedure

The following examples demonstrate how you can use the LIFEREG procedure to fit a parametric model to failure time data.

Suppose you have a response variable y that represents failure time; a binary variable, censor, with censor=0 indicating censored values; and two linearly independent variables, x1 and x2. The following statements perform a typical accelerated failure time model analysis. Higher-order effects such as interactions and nested effects are allowed in the independent variables list, but they are not shown in this example.

```
proc lifereg;
   model y*censor(0) = x1 x2;
run;
```

PROC LIFEREG can fit models to interval-censored data. The syntax for specifying interval-censored data is as follows:

```
proc lifereg;
   model (begin, end) = x1 x2;
run;
```

You can also model binomial data by using the *events/trials* syntax for the response, as illustrated in the following statements:

```
proc lifereg;
   model r/n=x1 x2;
run;
```

The variable n represents the number of trials, and the variable r represents the number of events.

Modeling Right-Censored Failure Time Data

The following example demonstrates how you can use the LIFEREG procedure to fit a model to right-censored failure time data.

Suppose you conduct a study of two headache pain relievers. You divide patients into two groups, with each group receiving a different type of pain reliever. You record the time taken (in minutes) for each patient to report headache relief. Because some of the patients never report relief for the entire study, some of the observations are censored.

The following DATA step creates the SAS data set headache:

```
data Headache;
   input Minutes Group Censor @@;
   datalines;
11  1  0   12  1  0   19  1  0   19  1  0
19  1  0   19  1  0   21  1  0   20  1  0
21  1  0   21  1  0   20  1  0   21  1  0
20  1  0   21  1  0   25  1  0   27  1  0
30  1  0   21  1  1   24  1  1   14  2  0
16  2  0   16  2  0   21  2  0   21  2  0
23  2  0   23  2  0   23  2  0   23  2  0
25  2  1   23  2  0   24  2  0   24  2  0
26  2  1   32  2  1   30  2  1   30  2  0
32  2  1   20  2  1
;
```

The data set Headache contains the variable Minutes, which represents the reported time to headache relief; the variable Group, the group to which the patient is assigned; and the variable Censor, a binary variable indicating whether the observation is censored. Valid values of the variable Censor are 0 (no) and 1 (yes). Figure 48.1 shows the first five records of the data set Headache.

Figure 48.1 Headache Data

Obs	Minutes	Group	Censor
1	11	1	0
2	12	1	0
3	19	1	0
4	19	1	0
5	19	1	0

The following statements invoke the LIFEREG procedure:

```
proc lifereg data=Headache;
   class Group;
   model Minutes*Censor(1)=Group;
   output out=New cdf=Prob;
run;
```

The CLASS statement specifies the variable Group as the classification variable. The MODEL statement syntax indicates that the response variable Minutes is right censored when the variable Censor takes the value 1. The MODEL statement specifies the variable Group as the single explanatory variable. Because the MODEL statement does not specify the DISTRIBUTION= option, the LIFEREG procedure fits the default type 1 extreme-value distribution by using log(Minutes) as the response. This is equivalent to fitting the Weibull distribution.

The OUTPUT statement creates the output data set New. In addition to containing the variables in the original data set Headache, the SAS data set New also contains the variable Prob. This new variable is created by the CDF= option to contain the estimates of the cumulative distribution function evaluated at the observed response.

The results of this analysis are displayed in the following figures.

Figure 48.2 Model Fitting Information from the LIFEREG Procedure

The LIFEREG Procedure

Model Information

Data Set	WORK.HEADACHE
Dependent Variable	Log(Minutes)
Censoring Variable	Censor
Censoring Value(s)	1
Number of Observations	38
Noncensored Values	30
Right Censored Values	8
Left Censored Values	0
Interval Censored Values	0
Name of Distribution	Weibull
Log Likelihood	-9.37930239

Class Level Information

Name	Levels	Values
Group	2	1 2

Figure 48.2 displays the class level information and model fitting information. There are 30 uncensored observations and 8 right-censored observations. The log likelihood for the Weibull distribution is –9.3793. The log-likelihood value can be used to compare the goodness of fit for different models.

Figure 48.3 Model Parameter Estimates from the LIFEREG Procedure

Analysis of Maximum Likelihood Parameter Estimates

Parameter		DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept		1	3.3091	0.0589	3.1938	3.4245	3161.70	<.0001
Group	1	1	-0.1933	0.0786	-0.3473	-0.0393	6.05	0.0139
Group	2	0	0.0000	.	.	.	.	.
Scale		1	0.2122	0.0304	0.1603	0.2809		
Weibull Shape		1	4.7128	0.6742	3.5604	6.2381		

The table of parameter estimates is displayed in Figure 48.3. Both the intercept and the slope parameter for the variable group are significantly different from 0 at the 0.05 level. Because the variable group has only one degree of freedom, parameter estimates are given for only one level of the variable group (group=1). However, the estimate for the intercept parameter provides a baseline for group=2.

The resulting model is as follows:

$$\log(\text{minutes}) = \begin{cases} 3.30911843 - 0.1933025 & \text{for group} = 1 \\ 3.30911843 & \text{for group} = 2 \end{cases}$$

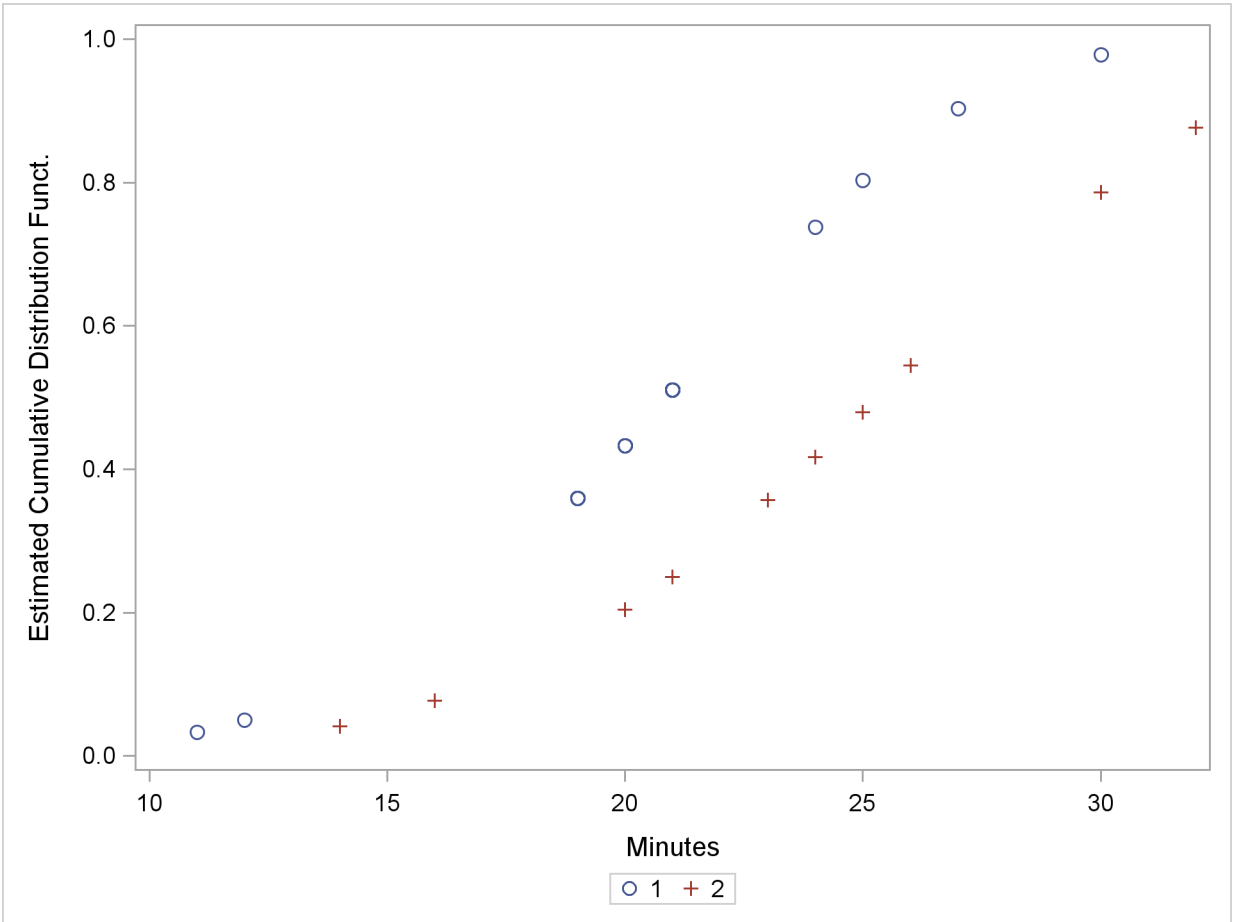
Note that the Weibull shape parameter for this model is the reciprocal of the extreme-value scale parameter estimate shown in Figure 48.3 ($1/0.21219 = 4.7128$).

The following statements produce a graph of the cumulative distribution values versus the variable Minutes.

```
proc sgplot data=New;
   scatter x=Minutes y=Prob / group=Group;
   discretelegend;
run;
```

Figure 48.4 displays the estimated cumulative distribution function values contained in the output data set New for each group.

Figure 48.4 Plot of the Estimated Cumulative Distribution Function



Bayesian Analysis of Right-Censored Data

Nelson (1982) describes a study of the lifetimes of locomotive engine fans. This example shows how to use PROC LIFEREG to carry out a Bayesian analysis of the engine fan data. In this example, a lognormal distribution is used to model the engine lifetimes, but other survival time distributions, such as the Weibull, can also be used.

The following SAS statements create the SAS data set Fan. This data set contains a censoring indicator variable and right-censored survival times for the 70 locomotive engine fans in the study.

```
data Fan;
   input Lifetime Censor@@;
   datalines;
 450 0    460 1   1150 0   1150 0   1560 1
1600 0   1660 1   1850 1   1850 1   1850 1
1850 1   1850 1   2030 1   2030 1   2030 1
2070 0   2070 0   2080 0   2200 1   3000 1
3000 1   3000 1   3000 1   3100 0   3200 1
3450 0   3750 1   3750 1   4150 1   4150 1
4150 1   4150 1   4300 1   4300 1   4300 1
4300 1   4600 0   4850 1   4850 1   4850 1
4850 1   5000 1   5000 1   5000 1   6100 1
6100 0   6100 1   6100 1   6300 1   6450 1
6450 1   6700 1   7450 1   7800 1   7800 1
8100 1   8100 1   8200 1   8500 1   8500 1
8500 1   8750 1   8750 0   8750 1   9400 1
9900 1  10100 1  10100 1  10100 1  11500 1
;
run;
```

Some of the fans had not failed at the time the data were collected, and the unfailed units have right-censored lifetimes. The variable Lifetime represents either a failure time or a censoring time. The variable Censor is equal to 0 if the value of Lifetime is a failure time, and it is equal to 1 if the value is a censoring time.

The following SAS statements specify a Bayesian analysis that uses a lognormal model for the engine lifetimes. There are no covariates, so the model is an intercept-only model. The OUTPOST= option saves the samples from the posterior distribution in the SAS data set Post for further processing.

```
ods graphics on;
proc lifereg data=Fan;
   model Lifetime*Censor( 1 )= / dist=lognormal;
   bayes seed=1  outpost=Post;
run;
ods graphics off;
```

The SEED= option is specified to maintain reproducibility; no other options are specified in the BAYES statement. By default, a uniform prior distribution is assumed for the intercept coefficient. The uniform prior is a flat prior on the real line with a distribution that reflects ignorance of the location of the parameter, placing equal probability on all possible values the regression coefficient can

take. Using the uniform prior in the following example, you would expect the Bayesian estimates to resemble the classical results of maximizing the likelihood. If you can elicit an informative prior on the regression coefficients, you should use the COEFFPRIOR= option to specify it. A default noninformative gamma prior is used for the lognormal scale parameter σ.

You should make sure that the posterior distribution samples have achieved convergence before using them for Bayesian inference. If you do not specify additional options, PROC LIFEREG produces by default three convergence diagnostics: autocorrelations of the posterior sample, effective sample size, and the Geweke statistic. See the section "Assessing Markov Chain Convergence" on page 156 for information about assessing the convergence of the chain of posterior samples. Trace plots, posterior density plots, and autocorrelation function plots that are created using ODS Graphics are also provided for each parameter. See the section "Visual Analysis via Trace Plots" on page 156 for help in interpreting these plots.

The "Analysis of Maximum Likelihood Parameter Estimates" table in Figure 48.5 summarizes maximum likelihood estimates of the lognormal intercept and scale parameters.

Figure 48.5 Maximum Likelihood Estimates from the LIFEREG Procedure

The LIFEREG Procedure

Bayesian Analysis

Analysis of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits	
Intercept	1	10.1432	0.5211	9.1219	11.1646
Scale	1	1.6796	0.3893	1.0664	2.6453

Since no prior distribution for the intercept was specified, the default uniform improper distribution shown in the "Uniform Prior for Regression Coefficients" table in Figure 48.6 is used.

Noninformative prior distributions are appropriate if you have no prior knowledge of the likely range of values of the parameters, and if you want to make probability statements about the parameters or functions of the parameters. Refer, for example, to Ibrahim, Chen, and Sinha (2001) for more information about choosing prior distributions.

The default noninformative gamma prior distribution for the lognormal scale parameter is shown in the "Independent Prior Distributions for Model Parameters" table in Figure 48.6.

Figure 48.6 Noninformative Prior Distributions

The LIFEREG Procedure

Bayesian Analysis

Uniform Prior for Regression Coefficients

Parameter	Prior
Intercept	Constant

Independent Prior Distributions for Model Parameters

Parameter	Prior Distribution	Hyperparameters			
Scale	Gamma	Shape	0.001	Inverse Scale	0.001

By default, posterior mode estimates of the model parameters are used as the starting value for the simulation. These are listed in the "Initial Values of the Chain" table in Figure 48.7.

Figure 48.7 Markov Chain Initial Values

Initial Values of the Chain

Chain	Seed	Intercept	Scale
1	1	10.0501	1.59544

Summary statistics for the posterior sample are displayed in the "Fit Statistics," "Descriptive Statistics for the Posterior Sample," "Interval Statistics for the Posterior Sample," and "Posterior Correlation Matrix" tables in Figure 48.8. Since noninformative prior distributions were used, these results are consistent with the maximum likelihood estimates shown in Figure 48.5.

Figure 48.8 Posterior Sample Summary Statistics

Fit Statistics

DIC (smaller is better)	87.245
pD (effective number of parameters)	1.823

Figure 48.8 *continued*

The LIFEREG Procedure

Bayesian Analysis

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
Intercept	10000	10.4196	0.6172	9.9670	10.3259	10.7959
Scale	10000	1.9196	0.4809	1.5675	1.8476	2.1931

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
Intercept	0.050	9.4477	11.8994	9.3216	11.6752
Scale	0.050	1.1906	3.0570	1.1104	2.8834

Posterior Correlation Matrix

Parameter	Intercept	Scale
Intercept	1.0000	0.8297
Scale	0.8297	1.0000

By default, PROC LIFEREG computes three convergence diagnostics: the lag1, lag5, lag10, and lag50 autocorrelations; the Geweke diagnostic; and the effective sample size. These are displayed in Figure 48.9. There is no indication that the Markov chain has not converged. See the section "Assessing Markov Chain Convergence" on page 156 for more information about convergence diagnostics and their interpretation.

Figure 48.9 Posterior Sample Summary Statistics

The LIFEREG Procedure

Bayesian Analysis

Posterior Autocorrelations

Parameter	Lag 1	Lag 5	Lag 10	Lag 50
Intercept	0.6973	0.1765	0.0190	-0.0017
Scale	0.6955	0.1713	0.0172	-0.0002

Geweke Diagnostics

Parameter	z	Pr > |z|
Intercept	-0.9183	0.3585
Scale	-0.9233	0.3559

Figure 48.9 *continued*

Effective Sample Sizes

Parameter	ESS	Correlation Time	Efficiency
Intercept	1772.8	5.6408	0.1773
Scale	1805.0	5.5400	0.1805

Summary statistics of the posterior distribution samples are produced by default. However, these statistics might not be sufficient for carrying out your Bayesian inference. The samples from the posterior distribution saved in the SAS data set Post created with the OUTPOST= option can be used for further analysis.

Trace, autocorrelation, and density plots for the three model parameters shown in Figure 48.10 and Figure 48.11 are useful in diagnosing whether the Markov chain of posterior samples has converged. These plots show no evidence that the chain has not converged. See the section "Visual Analysis via Trace Plots" on page 156 for more information about interpreting these types of diagnostic plots.

Figure 48.10 Diagnostic Plots

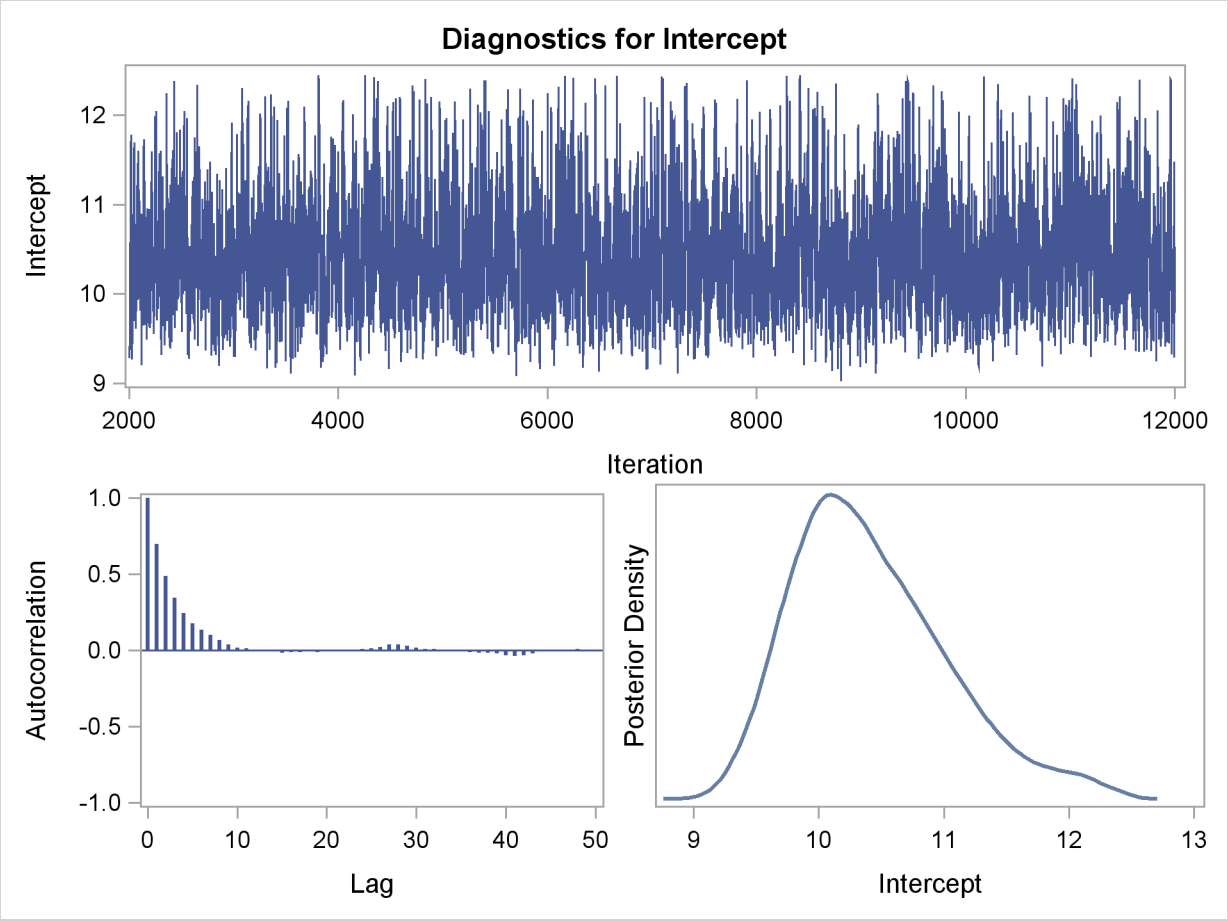
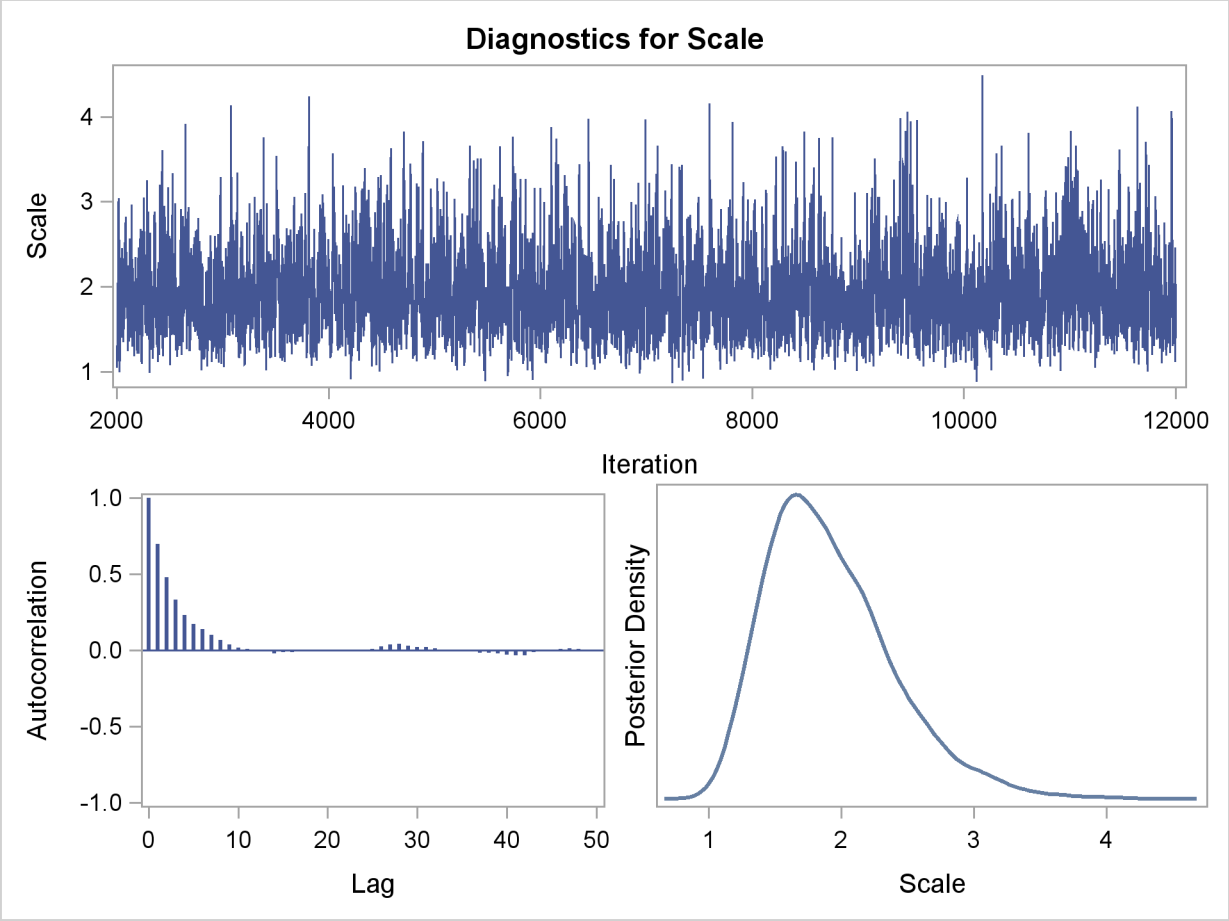


Figure 48.11 Diagnostic Plots



The fraction failing in the first 8000 hours of operation might be a quantity of interest. This kind of information could be useful, for example, in determining whether to improve the reliability of the engine components due to warranty considerations. The following SAS statements compute the mean and percentiles of the distribution of the fraction failing in the first 8000 hours from the posterior sample data set Post:

```
data Prob;
   set Post;
   Frac = ProbNorm(( log(8000) - Intercept ) / Scale );
   label Frac= 'Fraction Failing in 8000 Hours';
run;

proc means data = Prob(keep=Frac) n mean p10 p25 p50 p75 p90;
run;
```

The mean fraction of failures in the first 8000 hours, shown in Figure 48.12, is about 0.24, which could be used in further analysis of warranty costs. The 10th percentile is about 0.16 and the 90th percentile is about 0.32, which gives an assessment of the probable range of the fraction failing in the first 8000 hours.

Figure 48.12 Fraction Failing in 8000 Hours

```
                          The MEANS Procedure

            Analysis Variable : Frac Fraction Failing in 8000 Hours

    N           Mean      10th Pctl      25th Pctl      50th Pctl      75th Pctl
-------------------------------------------------------------------------------
10000      0.2381467      0.1628591      0.1953691      0.2336756      0.2766051
-------------------------------------------------------------------------------

            Analysis Variable : Frac Fraction Failing in 8000 Hours

                                   90th Pctl
                                ------------
                                   0.3190883
                                ------------
```

Syntax: LIFEREG Procedure

The following statements are available in PROC LIFEREG:

PROC LIFEREG < *options* > **;**
BY *variables* **;**
CLASS *variables* **;**
INSET < *keyword-list* > < / *options* > **;**
MODEL *response*=< *effects* > < / *options* > **;**
OUTPUT < **OUT=***SAS-data-set* > < *keyword=name . . . keyword=name* > < *options* > **;**
PROBPLOT < / *options* > **;**
WEIGHT *variable* **;**

The PROC LIFEREG statement invokes the procedure. The MODEL statement is required and specifies the variables used in the regression part of the model as well as the distribution used for the error, or random, component of the model. Only a single MODEL statement can be used with one invocation of the LIFEREG procedure. If multiple MODEL statements are present, only the last is used. Main effects and interaction terms can be specified in the MODEL statement, as in the GLM procedure. Initial values can be specified in the MODEL statement or in an INEST= data set. If no initial values are specified, the starting estimates are obtained by ordinary least squares. The CLASS statement determines which explanatory variables are treated as categorical. The WEIGHT statement identifies a variable with values that are used to weight the observations. Observations with zero or negative weights are not used to fit the model, although predicted values can be computed for them. The OUTPUT statement creates an output data set containing predicted values and residuals.

PROC LIFEREG Statement

PROC LIFEREG < *options* > **;**

The PROC LIFEREG statement invokes the procedure. You can specify the following options in the PROC LIFEREG statement.

COVOUT
writes the estimated covariance matrix to the OUTEST= data set if convergence is attained.

DATA=*SAS-data-set*
specifies the input SAS data set used by PROC LIFEREG. By default, the most recently created SAS data set is used.

GOUT=*graphics-catalog*
specifies a graphics catalog in which to save graphics output.

INEST=*SAS-data-set*
specifies an input SAS data set that contains initial estimates for all the parameters in the model. See the section "INEST= Data Set" on page 3050 for a detailed description of the contents of the INEST= data set.

NAMELEN=*n*
specifies the length of effect names in tables and output data sets to be *n* characters, where *n* is a value between 20 and 200. The default length is 20 characters.

NOPRINT
suppresses the display of the output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 20, "Using the Output Delivery System."

ORDER=DATA | FORMATTED | FREQ | INTERNAL
specifies the sorting order for the levels of the classification variables (specified in the CLASS statement). This ordering determines which parameters in the model correspond to each level in the data. The following table illustrates how PROC LIFEREG interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	formatted value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent. For more information about sorting order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide*, and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

OUTEST=*SAS-data-set*
specifies an output SAS data set containing the parameter estimates, the maximized log likelihood, and, if the COVOUT option is specified, the estimated covariance matrix. See the section "OUTEST= Data Set" on page 3051 for a detailed description of the contents of the OUTEST= data set.

PLOTS=NONE | PROBPLOT
specifies the following graphics options:

NONE	suppresses any graphics specified in other LIFEREG statements, such as the BAYES or PROBPLOT statement.
PROBPLOT	creates a default probability plot based on information in the MODEL statement. If a PROBPLOT option is also specified, the probability plot specified in the PROBPLOT statement is created, and this option is ignored.

XDATA=*SAS-data-set*
specifies an input SAS data set that contains values for all the independent variables in the MODEL statement and variables in the CLASS statement for probability plotting. If there are covariates specified in a MODEL statement and a probability plot is requested with a PROBPLOT statement, you specify fixed values for the effects in the MODEL statement with the XDATA= data set. See the section "XDATA= Data Set" on page 3051 for a detailed description of the contents of the XDATA= data set.

BAYES Statement

BAYES < *options* > **;**

The BAYES statement requests a Bayesian analysis of the regression model by using Gibbs sampling. The Bayesian posterior samples (also known as the chain) for the regression parameters are not tabulated. The Bayesian posterior samples (also known as the chain) for the model parameters can be output to a SAS data set.

Table 48.1 summarizes the options available in the BAYES statement.

Table 48.1 BAYES Statement Options

Option	Description
Monte Carlo Options	
INITIAL=	specifies initial values of the chain
INITIALMLE	specifies that maximum likelihood estimates be used as initial values of the chain
METROPOLIS=	specifies the use of a Metropolis step
NBI=	specifies the number of burn-in iterations
NMC=	specifies the number of iterations after burn-in
SEED=	specifies the random number generator seed
THINNING=	controls the thinning of the Markov chain

Table 48.1 (*continued*)

Option	Description
Model and Prior Options	
COEFFPRIOR=	specifies the prior of the regression coefficients
EXPONENTIALSCALEPRIOR=	specifies the prior of the exponential scale parameter
GAMMASHAPEPRIOR=	specifies the prior of the three-parameter gamma shape parameter
SCALEPRIOR=	specifies the prior of the scale parameter
WEIBULLSCALEPRIOR=	specifies the prior of the Weibull scale parameter
WEIBULLSHAPEPRIOR=	specifies the prior of the Weibull shape parameter
Summary Statistics and Convergence Diagnostics	
DIAGNOSTICS=	displays convergence diagnostics
PLOTS=	displays diagnostic plots
STATISTICS=	displays summary statistics of the posterior samples
Posterior Samples	
OUTPOST=	names a SAS data set for the posterior samples

The following list describes these options and their suboptions.

COEFFPRIOR=UNIFORM | NORMAL <(*normal-options*)>
CPRIOR=UNIFORM | NORMAL <(*option*)>
COEFF=UNIFORM | NORMAL <(*option*)>

specifies the prior distribution for the regression coefficients. The default is COEFFPRIOR=UNIFORM. The available prior distributions are as follows:

NORMAL<(*normal-option*)>

specifies a normal distribution. The *normal-options* include the following:

CONDITIONAL

specifies that the normal prior, conditional on the current Markov chain value of the location-scale model precision parameter $\tau = \frac{1}{\sigma^2}$, is $N(\boldsymbol{\mu}, \tau^{-1}\boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the mean and covariance of the normal prior specified by other normal options.

INPUT= *SAS-data-set*

specifies a SAS data set that contains the mean and covariance information of the normal prior. The data set must have a _TYPE_ variable to represent the type of each observation and a variable for each regression coefficient. If the data set also contains a _NAME_ variable, the values of this variable are used to identify the covariances for the _TYPE_='COV' observations; otherwise, the _TYPE_='COV' observations are assumed to be in the same order as the explanatory variables in the MODEL statement. PROC LIFEREG reads the mean vector from the observation with _TYPE_='MEAN' and reads the covariance matrix from observations with _TYPE_='COV'. For an independent normal prior, the variances can be specified with _TYPE_='VAR'; alternatively, the precisions (inverse of the variances) can be specified with _TYPE_='PRECISION'.

RELVAR< =c >
specifies the normal prior $N(\mathbf{0}, c\mathbf{J})$, where $\mathbf{J}$ is a diagonal matrix with diagonal elements equal to the variances of the corresponding ML estimator. By default, $c = 10^6$.

VAR< =c >
specifies the normal prior $N(\mathbf{0}, c\mathbf{I})$, where $\mathbf{I}$ is the identity matrix.

If you do not specify an option, the normal prior $N(\mathbf{0}, 10^6\mathbf{I})$, where $\mathbf{I}$ is the identity matrix, is used. See the section "Normal Prior" on page 3054 for more details.

UNIFORM
specifies a flat prior—that is, the prior that is proportional to a constant ($p(\beta_1, \ldots, \beta_k) \propto 1$ for all $-\infty < \beta_i < \infty$).

DIAGNOSTICS=ALL | NONE | *(keyword-list)*

DIAG=ALL | NONE | *(keyword-list)*
controls the number of diagnostics produced. You can request all the following diagnostics by specifying DIAGNOSTICS=ALL. If you do not want any of these diagnostics, specify DIAGNOSTICS=NONE. If you want some but not all of the diagnostics, or if you want to change certain settings of these diagnostics, specify a subset of the following keywords. The default is DIAGNOSTICS=(AUTOCORR ESS GEWEKE).

AUTOCORR < (LAGS= *numeric-list*) >
computes the autocorrelations of lags given by LAGS= list for each parameter. Elements in the list are truncated to integers and repeated values are removed. If the LAGS= option is not specified, autocorrelations of lags 1, 5, 10, and 50 are computed for each variable. See the section "Autocorrelations" on page 169 for details.

ESS
computes Carlin's estimate of the effective sample size, the correlation time, and the efficiency of the chain for each parameter. See the section "Effective Sample Size" on page 169 for details.

MCERROR
computes an estimate of the Monte Carlo standard error for each parameter. See the section "Standard Error of the Mean Estimate" on page 170 for details.

HEIDELBERGER < (*heidel-options***) >**
computes the Heidelberger and Welch diagnostic for each variable, which consists of a stationarity test of the null hypothesis that the sample values form a stationary process. If the stationarity test is not rejected, a halfwidth test is then carried out. Optionally, you can specify one or more of the following *heidel-options*:

SALPHA=*value*
specifies the α level ($0 < \alpha < 1$) for the stationarity test.

HALPHA=*value*
specifies the α level ($0 < \alpha < 1$) for the halfwidth test.

EPS=*value*

specifies a positive number ϵ such that if the halfwidth is less than ϵ times the sample mean of the retained iterates, the halfwidth test is passed.

See the section "Heidelberger and Welch Diagnostics" on page 165 for details.

GELMAN <(*gelman-options***)>**

computes the Gelman and Rubin convergence diagnostics. You can specify one or more of the following *gelman-options*:

NCHAIN=*number*

N=*number*

specifies the number of parallel chains used to compute the diagnostic, and must be 2 or larger. The default is NCHAIN=3. If an INITIAL= data set is used, NCHAIN defaults to the number of rows in the INITIAL= data set. If any number other than this is specified with the NCHAIN= option, the NCHAIN= value is ignored.

ALPHA=*value*

specifies the significance level for the upper bound. The default is ALPHA=0.05, resulting in a 97.5% bound.

See the section "Gelman and Rubin Diagnostics" on page 161 for details.

GEWEKE <(*geweke-options***)>**

computes the Geweke spectral density diagnostics, which are essentially a two-sample t test between the first f_1 portion and the last f_2 portion of the chain. The default is $f_1 = 0.1$ and $f_2 = 0.5$, but you can choose other fractions by using the following *geweke-options*:

FRAC1=*value*

specifies the fraction f_1 for the first window.

FRAC2=*value*

specifies the fraction f_2 for the second window.

See the section "Geweke Diagnostics" on page 163 for details.

RAFTERY<(*raftery-options***)>**

computes the Raftery and Lewis diagnostics that evaluate the accuracy of the estimated quantile ($\hat{\theta}_Q$ for a given $Q \in (0,1)$) of a chain. $\hat{\theta}_Q$ can achieve any degree of accuracy when the chain is allowed to run for a long time. A stopping criterion is when the estimated probability $\hat{P}_Q = \Pr(\theta \leq \hat{\theta}_Q)$ reaches within $\pm R$ of the value Q with probability S; that is, $\Pr(Q - R \leq \hat{P}_Q \leq Q + R) = S$. The following *raftery-options* enable you to specify Q, R, S, and a precision level ϵ for the test:

QUANTILE | Q=*value*

specifies the order (a value between 0 and 1) of the quantile of interest. The default is 0.025.

ACCURACY | R=*value*
specifies a small positive number as the margin of error for measuring the accuracy of estimation of the quantile. The default is 0.005.

PROBABILITY | S=*value*
specifies the probability of attaining the accuracy of the estimation of the quantile. The default is 0.95.

EPSILON | EPS=*value*
specifies the tolerance level (a small positive number) for the stationary test. The default is 0.001.

See the section "Raftery and Lewis Diagnostics" on page 166 for details.

EXPSCALEPRIOR=GAMMA< (*options***) > | IMPROPER**
ESCALEPRIOR=GAMMA< (*options***) > | IMPROPER**
ESCPRIOR=GAMMA< (*options***) > | IMPROPER**
specifies that Gibbs sampling be performed on the exponential distribution scale parameter and the prior distribution for the scale parameter. This prior distribution applies only when the exponential distribution and no covariates are specified.

A gamma prior $G(a, b)$ with density $f(t) = \frac{b(bt)^{a-1}e^{-bt}}{\Gamma(a)}$ is specified by EXPSCALEPRIOR=GAMMA, which can be followed by one of the following *gamma-options* enclosed in parentheses. The hyperparameters a and b are the shape and inverse-scale parameters of the gamma distribution, respectively. See the section "Gamma Prior" on page 3054 for more details. The default is $G(10^{-4}, 10^{-4})$.

RELSHAPE< =*c* **>**
specifies independent $G(c\hat{\alpha}, c)$ distribution, where $\hat{\alpha}$ is the MLE of the exponential scale parameter. With this choice of hyperparameters, the mean of the prior distribution is $\hat{\alpha}$ and the variance is $\frac{\hat{\alpha}}{c^2}$. By default, c=10^{-4}.

SHAPE=*a*
and

ISCALE=*b*
specify the $G(a, b)$ prior.

SHAPE=*c*
specifies the $G(c, c)$ prior.

ISCALE=*c*
specifies the $G(c, c)$ prior.

An improper prior with density $f(t)$ proportional to t^{-1} is specified with EXPSCALEPRIOR=IMPROPER.

GAMMASHAPEPRIOR=NORMAL< (*options*) >

GAMASHAPEPRIOR=NORMAL< (*options*) >

SHAPE1PRIOR=NORMAL< (*options*) >

specifies the prior distribution for the gamma distribution shape parameter. If you do not specify any options in a gamma model, the $N(0, 10^6)$ prior for the shape is used. You can specify MEAN= and VAR= or RELVAR= options, either alone or together, to specify the mean and variance of the normal prior for the gamma shape parameter.

MEAN=*a*

specifies a normal prior $N(a, 10^6)$. By default, a=0.

RELVAR< =*b* **>**

specifies the normal prior $N(0, bJ)$, where J is the variance of the MLE of the shape parameter. By default, b=10^6.

VAR=*c*

specifies the normal prior $N(0, c)$. By default, c=10^6.

INITIAL=*SAS-data-set*

specifies the SAS data set that contains the initial values of the Markov chains. The INITIAL= data set must contain all the variables of the model. You can specify multiple rows as the initial values of the parallel chains for the Gelman-Rubin statistics, but posterior summaries, diagnostics, and plots are computed only for the first chain. If the data set also contains the variable _SEED_, the value of the _SEED_ variable is used as the seed of the random number generator for the corresponding chain.

INITIALMLE

specifies that maximum likelihood estimates of the model parameters be used as initial values of the Markov chain. If this option is not specified, estimates of the mode of the posterior distribution obtained by optimization are used as initial values.

METROPOLIS=YES

METROPOLIS=NO

specifies the use of a Metropolis step to generate Gibbs samples for posterior distributions that are not log concave. The default value is METROPOLIS=YES.

NBI=*number*

specifies the number of burn-in iterations before the chains are saved. The default is 2000.

NMC=*number*

specifies the number of iterations after the burn-in. The default is 10000.

OUTPOST=*SAS-data-set*

OUT=*SAS-data-set*

names the SAS data set that contains the posterior samples. See the section "OUTPOST= Output Data Set" on page 3056 for more information. Alternatively, you can create the output data set by specifying an ODS OUTPUT statement as follows:

ODS OUTPUT PosteriorSample = *SAS-data-set* **;**

PLOTS< (*global-plot-options*) **>=** *plot-request*

PLOTS< (*global-plot-options*) **>=** *(plot-request < . . . plot-request >)*

controls the display of diagnostic plots. Three types of plots can be requested: trace plots, autocorrelation function plots, and kernel density plots. By default, the plots are displayed in panels unless the global plot option UNPACK is specified. Also, when specifying more than one type of plots, the plots are displayed by parameters unless the global plot option GROUPBY is specified. When you specify only one plot request, you can omit the parentheses around the plot request. For example:

```
plots=none
plots(unpack)=trace
plots=(trace autocorr)
```

You must enable ODS Graphics before requesting plots. For example, the following SAS statements enable ODS Graphics:

```
ods graphics on;
proc lifereg;
   model y=x;
   bayes plots=trace;
   run;
end;
ods graphics off;
```

The global plot options are as follows:

FRINGE

creates a fringe plot on the X axis of the density plot.

GROUPBY=PARAMETER

GROUPBY=TYPE

specifies how the plots are grouped when there is more than one type of plot.

GROUPBY=TYPE

specifies that the plots be grouped by type.

GROUPBY=PARAMETER

specifies that the plots be grouped by parameter.

GROUPBY=PARAMETER is the default.

LAGS=*n*

specifies that autocorrelations be plotted up to lag *n*. If this option is not specified, autocorrelations are plotted up to lag 50.

SMOOTH

displays a fitted penalized B-spline curve for each trace plot.

UNPACKPANEL

UNPACK

specifies that all paneled plots be unpacked, meaning that each plot in a panel is displayed separately.

The plot requests include the following:

ALL
specifies all types of plots. PLOTS=ALL is equivalent to specifying PLOTS=(TRACE AUTOCORR DENSITY).

AUTOCORR
displays the autocorrelation function plots for the parameters.

DENSITY
displays the kernel density plots for the parameters.

NONE
suppresses all diagnostic plots.

TRACE
displays the trace plots for the parameters. See the section "Visual Analysis via Trace Plots" on page 156 for details.

SCALEPRIOR=GAMMA< (*options*) >
specifies that Gibbs sampling be performed on the location-scale model scale parameter and the prior distribution for the scale parameter.

A gamma prior $G(a,b)$ with density $f(t) = \frac{b(bt)^{a-1}\mathrm{e}^{-bt}}{\Gamma(a)}$ is specified by SCALEPRIOR=GAMMA, which can be followed by one of the following *gamma-options* enclosed in parentheses. The hyperparameters a and b are the shape and inverse-scale parameters of the gamma distribution, respectively. See the section "Gamma Prior" on page 3054 for details. The default is $G(10^{-4}, 10^{-4})$.

RELSHAPE< =*c* >
specifies independent $G(c\hat{\sigma}, c)$ distribution, where $\hat{\sigma}$ is the MLE of the scale parameter. With this choice of hyperparameters, the mean of the prior distribution is $\hat{\sigma}$ and the variance is $\frac{\hat{\sigma}}{c}$. By default, c=10^{-4}.

SHAPE=*a*
and

ISCALE=*b*
specify the $G(a,b)$ prior.

SHAPE=*c*
specifies the $G(c,c)$ prior.

ISCALE=*c*
specifies the $G(c,c)$ prior.

SEED=*number*
specifies an integer seed in the range 1 to $2^{31}-1$ for the random number generator in the simulation. Specifying a seed enables you to reproduce identical Markov chains for the same specification. If the SEED= option is not specified, or if you specify a nonpositive seed, a random seed is derived from the time of day.

STATISTICS < (*global-options*) > = **ALL** | **NONE** | *keyword* | *(keyword-list)*

STATS < (*global-statoptions*) > = **ALL** | **NONE** | *keyword* | *(keyword-list)*

controls the number of posterior statistics produced. Specifying STATISTICS=ALL is equivalent to specifying STATISTICS= (SUMMARY INTERVAL COV CORR). If you do not want any posterior statistics, you specify STATISTICS=NONE. The default is STATISTICS=(SUMMARY INTERVAL). See the section "Summary Statistics" on page 170 for details. The *global-options* include the following:

ALPHA=*numeric-list*

controls the probabilities of the credible intervals. The ALPHA= values must be between 0 and 1. Each ALPHA= value produces a pair of 100(1–ALPHA)% equal-tail and HPD intervals for each parameters. The default is ALPHA=0.05, which yields the 95% credible intervals for each parameter.

PERCENT=*numeric-list*

requests the percentile points of the posterior samples. The PERCENT= values must be between 0 and 100. The default is PERCENT=25, 50, 75, which yields the 25th, 50th, and 75th percentile points, respectively, for each parameter.

The list of *keywords* includes the following:

CORR

produces the posterior correlation matrix.

COV

produces the posterior covariance matrix.

SUMMARY

produces the means, standard deviations, and percentile points for the posterior samples. The default is to produce the 25th, 50th, and 75th percentile points, but you can use the global PERCENT= option to request specific percentile points.

INTERVAL

produces equal-tail credible intervals and HPD intervals. The defult is to produce the 95% equal-tail credible intervals and 95% HPD intervals, but you can use the global ALPHA= option to request intervals of any probabilities.

NONE

suppresses printing all summary statistics.

THINNING=*number*

THIN=*number*

controls the thinning of the Markov chain. Only one in every k samples is used when THINNING=k, and if NBI=n_0 and NMC=n, the number of samples kept is

$$\left[\frac{n_0 + n}{k}\right] - \left[\frac{n_0}{k}\right]$$

where $[a]$ represents the integer part of the number a. The default is THINNING=1.

WEIBULLSCALEPRIOR=GAMMA< (*options*) >

WSCALEPRIOR=GAMMA< (*options*) >

WSCPRIOR=GAMMA< (*options*) >

specifies that Gibbs sampling be performed on the Weibull model scale parameter and the prior distribution for the scale parameter. This option applies only when a Weibull distribution and no covariates are specified. When this option is specified, PROC LIFEREG performs Gibbs sampling on the Weibull scale parameter, which is defined as $\exp(\mu)$, where μ is the intercept term.

A gamma prior $G(a, b)$ is specified by WEIBULLSCALEPRIOR=GAMMA, which can be followed by one of the following *gamma-options* enclosed in parentheses. The gamma probability density is given by $g(t) = \frac{b(bt)^{a-1}\mathrm{e}^{-bt}}{\Gamma(a)}$. The hyperparameters a and b are the shape and inverse-scale parameters of the gamma distribution, respectively. See the section "Gamma Prior" on page 3054 for details about the gamma prior. The default is $G(10^{-4}, 10^{-4})$.

RELSHAPE< =*c* >

specifies independent $G(c\hat{\alpha}, c)$ distribution, where $\hat{\alpha}$ is the MLE of the Weibull scale parameter. With this choice of hyperparameters, the mean of the prior distribution is $\hat{\alpha}$ and the variance is $\frac{\hat{\alpha}}{c}$. By default, c=10^{-4}.

SHAPE=*a*

and

ISCALE=*b*

specify the $G(a, b)$ prior.

SHAPE=*c*

specifies the $G(c, c)$ prior.

ISCALE=*c*

specifies the $G(c, c)$ prior.

WEIBULLSHAPEPRIOR=GAMMA< (*options*) >

WSHAPEPRIOR=GAMMA< (*options*) >

WSHPRIOR=GAMMA< (*options*) >

specifies that Gibbs sampling be performed on the Weibull model shape parameter and the prior distribution for the shape parameter. When this option is specified, PROC LIFEREG performs Gibbs sampling on the Weibull shape parameter, which is defined as σ^{-1}, where σ is the location-scale model scale parameter.

A gamma prior $G(a, b)$ with density $f(t) = \frac{b(bt)^{a-1}\mathrm{e}^{-bt}}{\Gamma(a)}$ is specified by WEIBULLSHAPEPRIOR=GAMMA, which can be followed by one of the following *gamma-options* enclosed in parentheses. The hyperparameters a and b are the shape and inverse-scale parameters of the gamma distribution, respectively. See the section "Gamma Prior" on page 3054 for details about the gamma prior. The default is $G(10^{-4}, 10^{-4})$.

RELSHAPE< =*c* >

specifies independent $G(c\hat{\beta}, c)$ distribution, where $\hat{\beta}$ is the MLE of the Weibull shape

parameter. With this choice of hyperparameters, the mean of the prior distribution is $\hat{\beta}$ and the variance is $\frac{\hat{\beta}}{c}$. By default, c=10^{-4}.

SHAPE<=*a***>**
and

ISCALE=*b*
specify the $G(a, b)$ prior.

SHAPE=*c*
specifies the $G(c, c)$ prior.

ISCALE=*c*
specifies the $G(c, c)$ prior.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC LIFEREG to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the LIFEREG procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

Variables that are classification variables rather than quantitative numeric variables must be listed in the CLASS statement. For each explanatory variable listed in the CLASS statement, indicator variables are generated for the levels assumed by the CLASS variable. If the CLASS statement is used, it must appear before the MODEL statement.

INSET Statement

INSET < *keyword-list* > < / *options* > **;**

The box or table of summary information produced on plots made with the PROBPLOT statement is called an *inset*. You can use the INSET statement to customize the information that is displayed in the inset box as well as to customize the appearance of the inset box. To supply the information that is displayed in the inset box, you specify *keywords* corresponding to the information that you want shown. For example, the following statements produce a probability plot with the number of observations, the number of right-censored observations, the name of the distribution, and the estimated Weibull shape parameter in the inset:

```
proc lifereg data=epidemic;
   model life = dose / dist = Weibull;
   probplot ;
   inset nobs right dist shape;
run;
```

By default, inset entries are identified with appropriate labels. However, you can provide a customized label by specifying the *keyword* for that entry followed by the equal sign (=) and the label in quotes. For example, the following INSET statement produces an inset containing the number of observations and the name of the distribution, labeled “Sample Size” and “Distribution” in the inset:

```
inset nobs='Sample Size' dist='Distribution';
```

If you specify a keyword that does not apply to the plot you are creating, then the keyword is ignored.

If you specify more than one INSET statement, only the first one is used.

Table 48.2 lists keywords available in the INSET statement to display summary statistics, distribution parameters, and distribution fitting information.

Table 48.2 INSET Statement Keywords

Keyword	Description
CONFIDENCE	confidence coefficient for all confidence intervals
DIST	name of the distribution
INTERVAL	number of interval-censored observations
LEFT	number of left-censored observations
NOBS	number of observations
NMISS	number of observations with missing values
RIGHT	number of right-censored observations
SCALE	value of the scale parameter
SHAPE	value of the shape parameter
UNCENSORED	number of uncensored observations

The following *options* control the appearance of the box when you use traditional graphics. These options are not available if ODS Graphics is enabled. All *options* are specified after the slash (/) in the INSET statement.

CFILL=*color*
specifies the color for the filling box.

CFILLH=*color*
specifies the color for the filling box header.

CFRAME=*color*
specifies the color for the frame.

CHEADER=*color*
specifies the color for text in the header.

CTEXT=*color*
specifies the color for the text.

FONT=*font*
specifies the software font for the text.

HEIGHT=*value*
specifies the height of the text.

HEADER=*'quoted string'*
specifies the text for the header or box title.

NOFRAME
omits the frame around the box.

POS=*value* **< DATA | PERCENT >**

determines the position of the inset. The *value* can be a compass point (N, NE, E, SE, S, SW, W, NW) or a pair of coordinates (x, y) enclosed in parentheses. The coordinates can be specified in screen percentage units or axis data units. The default is screen percentage units.

REFPOINT=*name*

specifies the reference point for an inset that is positioned by a pair of coordinates with the POS= option. You use the REFPOINT= option in conjunction with the POS= coordinates. The REFPOINT= option specifies which corner of the inset frame you have specified with coordinates (x, y), and it can take the value of BR (bottom right), BL (bottom left), TR (top right), or TL (top left). The default is REFPOINT=BL. If the inset position is specified as a compass point, then the REFPOINT= option is ignored.

MODEL Statement

<label:> **MODEL** *response< *censor(list) >=effects < / options >* **;**

<label:> **MODEL** *(lower,upper)=effects < / options >* **;**

<label:> **MODEL** *events/trials=effects < / options >* **;**

Only a single MODEL statement can be used with one invocation of the LIFEREG procedure. If multiple MODEL statements are present, only the last is used. The optional *label* is used to label the model estimates in the output SAS data set and OUTEST= data set.

The first MODEL syntax is appropriate for right censoring. The variable *response* is possibly right censored. If the *response* variable can be right censored, then a second variable, denoted *censor*, must appear after the *response* variable with a list of parenthesized values, separated by commas or blanks, to indicate censoring. That is, if the *censor* variable takes on a value given in the list, the *response* is a right-censored value; otherwise, it is an observed value.

The second MODEL syntax specifies two variables, *lower* and *upper*, that contain values of the endpoints of the censoring interval. If the two values are the same (and not missing), it is assumed that there is no censoring and the actual response value is observed. If the lower value is missing, then the upper value is used as a left-censored value. If the upper value is missing, then the lower value is taken as a right-censored value. If both values are present and the lower value is less than the upper value, it is assumed that the values specify a censoring interval. If the lower value is greater than the upper value or both values are missing, then the observation is not used in the analysis, although predicted values can still be obtained if none of the covariates are missing. The following table summarizes the ways of specifying censoring.

lower	*upper*	**Comparison**	**Interpretation**
not missing	not missing	equal	no censoring
not missing	not missing	lower < upper	censoring interval
missing	not missing		upper used as left-censoring value
not missing	missing		lower used as right-censoring value
not missing	not missing	lower > upper	observation not used
missing	missing		observation not used

The third MODEL syntax specifies two variables that contain count data for a binary response. The value of the first variable, *events*, is the number of successes. The value of the second variable, *trials*, is the number of tries. The values of both *events* and (*trials-events*) must be nonnegative, and *trials* must be positive for the response to be valid. The values of the two variables do not need to be integers and are not modified to be integers.

The *effects* following the equal sign are the covariates in the model. Higher-order effects, such as interactions and nested terms, are allowed in the list, similar to the GLM procedure. Variable names and combinations of variable names representing higher-order terms are allowed to appear in this list. Classification, or CLASS, variables can be used as effects, and indicator variables are generated for the class levels. If you do not specify any covariates following the equal sign, an intercept-only model is fit.

Examples of three valid MODEL statements follow:

```
a: model time*flag(1,3)=temp;

b: model (start, finish)=;

c: model r/n=dose;
```

MODEL statement a indicates that the response is contained in a variable named time and that, if the variable flag takes on the values 1 or 3, the observation is right censored. The explanatory variable is temp, which could be a CLASS variable. MODEL statement b indicates that the response is known to be in the interval between the values of the variables start and finish and that there are no covariates except for a default intercept term. MODEL statement c indicates a binary response, with the variable r containing the number of responses and the variable n containing the number of trials.

The following options can appear in the MODEL statement.

Task	Option
Model specification	
set the significance level	ALPHA=
specify distribution type for failure time	DISTRIBUTION=
request no log transformation of response	NOLOG
initial estimate for intercept term	INTERCEPT=
hold intercept term fixed	NOINT
initial estimates for regression parameters	INITIAL=
initialize scale parameter	SCALE=
hold scale parameter fixed	NOSCALE
initialize first shape parameter	SHAPE1=
hold first shape parameter fixed	NOSHAPE1
Model fitting	
set convergence criterion	CONVERGE=
set maximum iterations	MAXITER=
set tolerance for testing singularity	SINGULAR=
Output	
display estimated correlation matrix	CORRB
display estimated covariance matrix	COVB
display iteration history, final gradient, and second derivative matrix	ITPRINT

ALPHA=*value*

sets the significance level for the confidence intervals for regression parameters and estimated survival probabilities. The value must be between 0 and 1. By default, ALPHA=0.05.

CONVERGE=*value*

sets the convergence criterion. Convergence is declared when the maximum change in the parameter estimates between Newton-Raphson steps is less than the value specified. The change is a relative change if the parameter is greater than 0.01 in absolute value; otherwise, it is an absolute change. By default, CONVERGE=1E−8.

CONVG=*value*

sets the relative Hessian convergence criterion; *value* must be between 0 and 1. After convergence is determined with the change in parameter criterion specified with the CONVERGE= option, the quantity $tc = \frac{\mathbf{g}'\mathbf{H}^{-1}\mathbf{g}}{|f|}$ is computed and compared to *value*, where $\mathbf{g}$ is the gradient vector, $\mathbf{H}$ is the Hessian matrix for the model parameters, and f is the log-likelihood function. If tc is greater than *value*, a warning that the relative Hessian convergence criterion has been exceeded is displayed. This criterion detects the occasional case where the change in parameter convergence criterion is satisfied, but a maximum in the log-likelihood function has not been attained. By default, CONVG=1E−4.

CORRB

produces the estimated correlation matrix of the parameter estimates.

COVB

produces the estimated covariance matrix of the parameter estimates.

DISTRIBUTION=*distribution-type*

DIST=*distribution-type*

D=*distribution-type*

specifies the distribution type assumed for the failure time. By default, PROC LIFEREG fits a type 1 extreme-value distribution to the log of the response. This is equivalent to fitting the Weibull distribution, since the scale parameter for the extreme-value distribution is related to a Weibull shape parameter and the intercept is related to the Weibull scale parameter in this case. When the NOLOG option is specified, PROC LIFEREG models the untransformed response with a type 1 extreme-value distribution as the default. See the section "Supported Distributions" on page 3039 for descriptions of the distributions. The following are valid values for *distribution-type*:

EXPONENTIAL	the exponential distribution, which is treated as a restricted Weibull distribution
GAMMA	a generalized gamma distribution (Lawless 2003, p. 240). The standard two-parameter gamma distribution is not available in PROC LIFEREG.
LLOGISTIC	a loglogistic distribution
LNORMAL	a lognormal distribution
LOGISTIC	a logistic distribution (equivalent to LLOGISTIC when the NOLOG option is specified)
NORMAL	a normal distribution (equivalent to LNORMAL when the NOLOG option is specified)
WEIBULL	a Weibull distribution. If NOLOG is specified, it fits a type 1 extreme-value distribution to the raw, untransformed data.

By default, PROC LIFEREG transforms the response with the natural logarithm before fitting the specified model when you specify the GAMMA, LLOGISTIC, LNORMAL, or WEIBULL option. You can suppress the log transformation with the NOLOG option. The following table summarizes the resulting distributions when the preceding distribution options are used in combination with the NOLOG option.

DISTRIBUTION=	NOLOG Specified?	Resulting Distribution
EXPONENTIAL	No	Exponential
EXPONENTIAL	Yes	One-parameter extreme value
GAMMA	No	Generalized gamma
GAMMA	Yes	Generalized gamma with untransformed responses
LOGISTIC	No	Logistic
LOGISTIC	Yes	Logistic (NOLOG has no effect)
LLOGISTIC	No	Log-logistic
LLOGISTIC	Yes	Logistic
LNORMAL	No	Lognormal
LNORMAL	Yes	Normal
NORMAL	No	Normal
NORMAL	Yes	Normal (NOLOG has no effect)
WEIBULL	No	Weibull
WEIBULL	Yes	Extreme value

INITIAL=*values*

sets initial values for the regression parameters. This option can be helpful in the case of convergence difficulty. Specified values are used to initialize the regression coefficients for the covariates specified in the MODEL statement. The intercept parameter is initialized with the INTERCEPT= option and is not included here. The values are assigned to the variables in the MODEL statement in the same order in which they are listed in the MODEL statement. Note that a CLASS variable requires $k - 1$ values when the CLASS variable takes on k different levels. The order of the CLASS levels is determined by the ORDER= option. If there is no intercept term, the first CLASS variable requires k initial values. If a BY statement is used, all CLASS variables must take on the same number of levels in each BY group or no meaningful initial values can be specified. The INITIAL= option can be specified as follows.

Type of List	Specification
list separated by blanks	`initial=3 4 5`
list separated by commas	`initial=3,4,5`
x to y	`initial=3 to 5`
x to y by z	`initial=3 to 5 by 1`
combination of methods	`initial=1,3 to 5,9`

By default, PROC LIFEREG computes initial estimates with ordinary least squares. See the section "Computational Method" on page 3037 for details.

NOTE: The INITIAL= option is overwritten by the INEST= option. See the section "INEST= Data Set" on page 3050 for details.

INTERCEPT=*value*

initializes the intercept term to *value*. By default, the intercept is initialized by an ordinary least squares estimate.

ITPRINT

displays the iteration history for computing maximum likelihood estimates, the final evaluation of the gradient, and the final evaluation of the negative of the second derivative matrix—that is, the negative of the Hessian. If you perform a Bayesian analysis by specifying the BAYES statement, the iteration history for computing the mode of the posterior distribution is also displayed.

MAXITER=*n*

sets the maximum allowable number of iterations during the model estimation. By default, MAXITER=50.

NOINT

holds the intercept term fixed. Because of the usual log transformation of the response, the intercept parameter is usually a scale parameter for the untransformed response, or a location parameter for a transformed response.

NOLOG

requests that no log transformation of the response variable be performed. By default, PROC

LIFEREG models the log of the response variable for the GAMMA, LLOGISTIC, LOGNORMAL, and WEIBULL distribution options. NOLOG is implicitly assumed for the NORMAL and LOGISTIC distribution options.

NOSCALE
holds the scale parameter fixed. Note that if the log transformation has been applied to the response, the effect of the scale parameter is a power transformation of the original response. If no SCALE= value is specified, the scale parameter is fixed at the value 1.

NOSHAPE1
holds the first shape parameter, SHAPE1, fixed. If no SHAPE1= value is specified, SHAPE1 is fixed at a value that depends on the DISTRIBUTION type.

OFFSET=*variable*
specifies a variable in the input data set to be used as an offset variable. This variable cannot be a CLASS variable, and it cannot be the response variable or one of the explanatory variables.

SCALE=*value*
initializes the scale parameter to *value*. If the Weibull distribution is specified, this scale parameter is the scale parameter of the type 1 extreme-value distribution, not the Weibull scale parameter. Note that, with a log transformation, the exponential model is the same as a Weibull model with the scale parameter fixed at the value 1.

SHAPE1=*value*
initializes the first shape parameter to *value*. If the specified distribution does not depend on this parameter, then this option has no effect. The only distribution that depends on this shape parameter is the generalized gamma distribution. See the section "Supported Distributions" on page 3039 for descriptions of the parameterizations of the distributions.

SINGULAR=*value*
sets the tolerance for testing singularity of the information matrix and the crossproducts matrix for the initial least squares estimates. Roughly, the test requires that a pivot be at least this value times the original diagonal value. By default, SINGULAR=1E−12.

OUTPUT Statement

OUTPUT < *OUT=SAS-data-set* > < *keyword=name* > . . . < *keyword=name* > **;**

The OUTPUT statement creates a new SAS data set containing statistics calculated after fitting the model. At least one specification of the form *keyword=name* is required.

All variables in the original data set are included in the new data set, along with the variables created as options for the OUTPUT statement. These new variables contain fitted values and estimated quantiles. If you want to create a permanent SAS data set, you must specify a two-level name (see *SAS Language Reference: Concepts* for more information about permanent SAS data sets). Each OUTPUT statement applies to the preceding MODEL statement. See Example 48.1 for illustrations of the OUTPUT statement.

The following specifications can appear in the OUTPUT statement:

OUT=*SAS-data-set* specifies the new data set. By default, the procedure uses the DATA*n* convention to name the new data set.

keyword=name specifies the statistics to include in the output data set and gives names to the new variables. Specify a keyword for each desired statistic (see the following list of keywords), an equal sign, and the variable to contain the statistic.

The keywords allowed and the statistics they represent are as follows:

CENSORED specifies an indicator variable to signal censoring. The variable takes on the value 1 if the observation is censored; otherwise, it is 0.

CDF specifies a variable to contain the estimates of the cumulative distribution function evaluated at the observed response. See the section "Predicted Values" on page 3043 for more information.

CONTROL specifies a variable in the input data set to control the estimation of quantiles. See Example 48.1 for an illustration. If the specified variable has the value 1, estimates for all the values listed in the QUANTILE= list are computed for that observation in the input data set; otherwise, no estimates are computed. If no CONTROL= variable is specified, all quantiles are estimated for all observations. If the response variable in the MODEL statement is binomial, then this option has no effect.

CRESIDUAL | CRES specifies a variable to contain the Cox-Snell residuals

$$-\log(S(u_i))$$

where S is the standard survival function and

$$u_i = \frac{y_i - \mathbf{x}_i'\mathbf{b}}{\sigma}$$

If the response variable in the corresponding model statement is binomial, then the residuals are not computed, and this variable contains missing values.

SRESIDUAL | SRES specifies a variable to contain the standardized residuals

$$\frac{y_i - \mathbf{x}_i'\mathbf{b}}{\sigma}$$

If the response variable in the corresponding model statement is binomial, then the residuals are not computed, and this variable contains missing values.

PREDICTED | P specifies a variable to contain the quantile estimates. If the response variable in the corresponding model statement is binomial, then this variable contains the estimated probabilities, $1 - F(-\mathbf{x}'\mathbf{b})$.

QUANTILES | QUANTILE | Q gives a list of values for which quantiles are calculated. The values must be between 0 and 1, noninclusive. For each value, a corresponding quantile is estimated. This option is not used if the response variable in the corresponding MODEL statement is binomial. The QUANTILES option can be specified as follows.

Type of List	Specification
list separated by blanks	`.2 .4 .6 .8`
list separated by commas	`.2,.4,.6,.8`
x to y	`.2 to .8`
x to y by z	`.2 to .8 by .1`
combination of methods	`.1,.2 to .8 by .2`

By default, QUANTILES=0.5. When the response is not binomial, a numeric variable, _PROB_, is added to the OUTPUT data set whenever the QUANTILES= option is specified. The variable _PROB_ gives the probability value for the quantile estimates. These are the values taken from the QUANTILES= list and are given as values between 0 and 1, not as values between 0 and 100.

STD_ERR | STD specifies a variable to contain the estimates of the standard errors of the estimated quantiles or $\mathbf{x}'\mathbf{b}$. If the response used in the MODEL statement is a binomial response, then these are the standard errors of $\mathbf{x}'\mathbf{b}$. Otherwise, they are the standard errors of the quantile estimates. These estimates can be used to compute confidence intervals for the quantiles. However, if the model is fit to the log of the event time, better confidence intervals can usually be computed by transforming the confidence intervals for the log response. See Example 48.1 for such a transformation.

XBETA specifies a variable to contain the computed value of $\mathbf{x}'\mathbf{b}$, where $\mathbf{x}$ is the covariate vector and $\mathbf{b}$ is the vector of parameter estimates.

PROBPLOT Statement

PROBPLOT | PPLOT < / *options* > **;**

You can use the PROBPLOT statement to create a probability plot from lifetime data. The data can be uncensored, right censored, or arbitrarily censored. You can specify any number of PROBPLOT statements after a MODEL statement. The syntax used for the response in the MODEL statement determines the type of censoring assumed in creating the probability plot. The model fit with the MODEL statement is plotted along with the data. If there are covariates in the model, they are set to constant values specified in the XDATA= data set when creating the probability plot. If no XDATA= data set is specified, continuous variables are set to their overall mean values and categorical variables specified in the CLASS statement are set to their highest levels.

You can specify the following options to control the content, layout, and appearance of a probability plot.

Traditional Graphics

The following options are available if you use traditional graphics—that is, if ODS Graphics is not enabled.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an Annotate data set, as described in *SAS/GRAPH Software: Reference*, that enables you to add features to the probability plot. The data set you specify with the ANNOTATE= option in the PROBPLOT statement provides the Annotate data set for all plots created by the statement.

CAXIS=*color*

CAXES=*color*

specifies the color used for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement. The default is the first color in the device color list.

CCENSOR=*color*

specifies the color for filling the censor plot area. The default is the first color in the device color list.

CENBIN

plots censored data as frequency counts (rounding for noninteger frequency) rather than as individual points.

CENCOLOR=*color*

specifies the color for the censor symbol. The default is the first color in the device color list.

CENSYMBOL=*symbol* | **(***symbol list***)**

specifies symbols for censored values. The *symbol* is one of the symbol names (plus, star, square, diamond, triangle, hash, paw, point, dot, and circle) or a letter (A–Z). If you do not specify the CENSYMBOL= option, the symbol used for censored values is the same as for failures.

CFIT=*color*

specifies the color for the fitted probability line and confidence curves. The default is the first color in the device color list.

CFRAME=*color*

CFR=*color*

specifies the color for the area enclosed by the axes and frame. This area is not shaded by default.

CGRID=*color*

specifies the color for grid lines. The default is the first color in the device color list.

CHREF=*color*

CH=*color*

specifies the color for lines requested by the HREF= option. The default is the first color in the device color list.

CTEXT=*color*

specifies the color for tick mark values and axis labels. The default is the color specified for the CTEXT= option in the most recent GOPTIONS statement.

CVREF=*color*

CV=*color*

specifies the color for lines requested by the VREF= option. The default is the first color in the device color list.

DESCRIPTION='*string***'**

DES='*string***'**

specifies a description, up to 40 characters, that appears in the PROC GREPLAY master menu. The default is the variable name.

FONT=*font*

specifies a software font for reference line and axis labels. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the most recent GOPTIONS statement. Hardware characters are used by default.

HCL

computes and draws confidence limits for the predicted probabilities in the horizontal direction.

HEIGHT=*value*

specifies the height of text used outside framed areas. The default value is 3.846 (in percentage).

HLOWER=*value*

specifies the lower limit on the lifetime axis scale. The HLOWER= option specifies *value* as the lower lifetime axis tick mark. The tick mark interval and the upper axis limit are determined automatically.

HOFFSET=*value*

specifies the offset for the horizontal axis. The default value is 1.

HUPPER=*value*

specifies *value* as the upper lifetime axis tick mark. The tick mark interval and the lower axis limit are determined automatically.

HREF <(INTERSECT)> =*value-list*

requests reference lines perpendicular to the horizontal axis. If (INTERSECT) is specified, a second reference line perpendicular to the vertical axis is drawn that intersects the fit line at the same point as the horizontal axis reference line. If a horizontal axis reference line label is specified, the intersecting vertical axis reference line is labeled with the vertical axis value. See also the CHREF=, HREFLABELS=, and LHREF= options.

HREFLABELS=*'label1'* **...** *'labeln'*

HREFLABEL=*'label1'* **...** *'labeln'*

HREFLAB=*'label1'* **...** *'labeln'*

specifies labels for the lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

HREFLABPOS=*n*

specifies the vertical position of labels for HREF= lines. The following table shows the valid values for *n* and the corresponding label placements.

n	Label Placement
1	top
2	staggered from top
3	bottom
4	staggered from bottom
5	alternating from top
6	alternating from bottom

INBORDER

requests a border around probability plots.

INTERTILE=*value*

specifies the distance between tiles.

ITPRINTEM

displays the iteration history for the Turnbull algorithm.

JITTER=*value*

specifies the amount to jitter overlaying plot symbols, in units of symbol width.

LFIT=*linetype*

specifies a line style for fitted curves and confidence limits. By default, fitted curves are drawn by connecting solid lines (*linetype = 1*), and confidence limits are drawn by connecting dashed lines (*linetype = 3*).

LGRID=*linetype*

specifies a line style for all grid lines; *linetype* is between 1 and 46. The default is 35.

LHREF=*linetype*

LH=*linetype*

specifies the line type for lines requested by the HREF= option. The default is 2, which produces a dashed line.

LVREF=*linetype*

LV=*linetype*

specifies the line type for lines requested by the VREF= option. The default is 2, which produces a dashed line.

MAXITEM=*n1 <,n2> n1*

specifies the maximum number of iterations allowed for the Turnbull algorithm. Iteration history will be displayed in increments of *n2* if requested with the ITPRINTEM option. See the section "Arbitrarily Censored Data" on page 3048 for details.

NAME=*'string'*

specifies a name for the plot, up to eight characters, that appears in the PROC GREPLAY master menu. The default is 'LIFEREG'.

NOCENPLOT
suppresses the plotting of censored data points.

NOCONF
suppresses the default percentile confidence bands on the probability plot.

NODATA
suppresses plotting of the estimated empirical probability plot.

NOFIT
suppresses the fitted probability (percentile) line and confidence bands.

NOFRAME
suppresses the frame around plotting areas.

NOGRID
suppresses grid lines.

NOHLABEL
suppresses horizontal labels.

NOHTICK
suppresses horizontal tick marks.

NOPOLISH
suppresses setting small interval probabilities to zero in the Turnbull algorithm.

NOVLABEL
suppresses vertical labels.

NOVTICK
suppresses vertical tick marks.

NPINTERVALS=*interval type*
specifies one of the two kinds of confidence limits for the estimated cumulative probabilities, pointwise (NPINTERVALS=POINT) or simultaneous (NPINTERVALS=SIMUL), requested by the PPOUT option to be displayed in the tabular output.

PCTLIST=*value-list*
specifies the list of percentages for which to compute percentile estimates; *value-list* must be a list of values separated by blanks or commas. Each value in the list must be between 0 and 100.

PLOWER=*value*
specifies the lower limit on the probability axis scale. The PLOWER= option specifies *value* as the lower probability axis tick mark. The tick mark interval and the upper axis limit are determined automatically.

PRINTPROBS
displays intervals and associated probabilities for the Turnbull algorithm.

PUPPER=*value*

specifies the upper limit on the probability axis scale. The PUPPER= option specifies *value* as the upper probability axis tick mark. The tick mark interval and the lower axis limit are determined automatically.

PPOS=*character-list*

specifies the plotting position type. See the section "Probability Plotting" on page 3046 for details.

PPOS	Method
EXPRANK	expected ranks
MEDRANK	median ranks
MEDRANK1	median ranks (exact formula)
KM	Kaplan-Meier
MKM	modified Kaplan-Meier (default)

PPOUT

specifies that a table of the cumulative probabilities plotted on the probability plot be displayed. Kaplan-Meier estimates of the cumulative probabilities are also displayed, along with standard errors and confidence limits. The confidence limits can be pointwise or simultaneous, as specified by the NPINTERVALS= option.

PROBLIST=*value-list*

specifies the list of initial values for the Turnbull algorithm.

ROTATE

requests probability plots with probability scale on the horizontal axis.

SQUARE

makes the layout of the probability plots square.

TOLLIKE=*value*

specifies the criterion for convergence in the Turnbull algorithm.

TOLPROB=*value*

specifies the criterion for setting the interval probability to zero in the Turnbull algorithm.

VAXISLABEL='*string*'

specifies a label for the vertical axis.

VREF=*value-list*

requests reference lines perpendicular to the vertical axis. If (INTERSECT) is specified, a second reference line perpendicular to the horizontal axis is drawn that intersects the fit line at the same point as the vertical axis reference line. If a vertical axis reference line label is specified, the intersecting horizontal axis reference line is labeled with the horizontal axis value. See also the entries for the CVREF=, LVREF=, and VREFLABELS= options.

VREFLABELS=*'label1' . . . 'labeln'*

VREFLABEL=*'label1' . . . 'labeln'*

VREFLAB=*'label1' . . . 'labeln'*

specifies labels for the lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

VREFLABPOS=*n*

specifies the horizontal position of labels for VREF= lines. The valid values for *n* and the corresponding label placements are shown in the following table.

n	Label Placement
1	left
2	right

WAXIS=*n*

specifies line thickness for axes and frame. The default value is 1.

WFIT=*n*

specifies line thickness for fitted curves. The default value is 1.

WGRID=*n*

specifies line thickness for grids. The default value is 1.

WREFL=*n*

specifies line thickness for reference lines. The default value is 1.

ODS Graphics

The following options are available if ODS Graphics is enabled.

HCL

computes and draws confidence limits for the predicted probabilities in the horizontal direction.

HLOWER=*value*

specifies the lower limit on the lifetime axis scale. The HLOWER= option specifies *value* as the lower lifetime axis tick mark. The tick mark interval and the upper axis limit are determined automatically.

HUPPER=*value*

specifies *value* as the upper lifetime axis tick mark. The tick mark interval and the lower axis limit are determined automatically.

HREF <(INTERSECT)> =*value-list*

requests reference lines perpendicular to the horizontal axis. If (INTERSECT) is specified, a second reference line perpendicular to the vertical axis is drawn that intersects the fit line at the same point as the horizontal axis reference line. If a horizontal axis reference line label is specified, the intersecting vertical axis reference line is labeled with the vertical axis value. See also the CHREF=, HREFLABELS=, and LHREF= options.

HREFLABELS=*'label1'* **...** *'labeln'*

HREFLABEL=*'label1'* **...** *'labeln'*

HREFLAB=*'label1'* **...** *'labeln'*

specifies labels for the lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

ITPRINTEM

displays the iteration history for the Turnbull algorithm.

MAXITEM=*n1 <,n2> n1*

specifies the maximum number of iterations allowed for the Turnbull algorithm. Iteration history will be displayed in increments of *n2* if requested with the ITPRINTEM option. See the section "Arbitrarily Censored Data" on page 3048 for details.

NOCENPLOT

suppresses the plotting of censored data points.

NOCONF

suppresses the default percentile confidence bands on the probability plot.

NODATA

suppresses plotting of the estimated empirical probability plot.

NOFIT

suppresses the fitted probability (percentile) line and confidence bands.

NOFRAME

suppresses the frame around plotting areas.

NOGRID

suppresses grid lines.

NOPOLISH

suppresses setting small interval probabilities to zero in the Turnbull algorithm.

NPINTERVALS=*interval type*

specifies one of the two kinds of confidence limits for the estimated cumulative probabilities, pointwise (NPINTERVALS=POINT) or simultaneous (NPINTERVALS=SIMUL), requested by the PPOUT option to be displayed in the tabular output.

PCTLIST=*value-list*

specifies the list of percentages for which to compute percentile estimates; *value-list* must be a list of values separated by blanks or commas. Each value in the list must be between 0 and 100.

PLOWER=*value*

specifies the lower limit on the probability axis scale. The PLOWER= option specifies *value* as the lower probability axis tick mark. The tick mark interval and the upper axis limit are determined automatically.

PRINTPROBS
displays intervals and associated probabilities for the Turnbull algorithm.

PUPPER=*value*
specifies the upper limit on the probability axis scale. The PUPPER= option specifies *value* as the upper probability axis tick mark. The tick mark interval and the lower axis limit are determined automatically.

PPOS=*plotting-position-type*
specifies the plotting position type. See the section "Probability Plotting" on page 3046 for details.

PPOS	Method
EXPRANK	expected ranks
MEDRANK	median ranks
MEDRANK1	median ranks (exact formula)
KM	Kaplan-Meier
MKM	modified Kaplan-Meier (default)

PPOUT
specifies that a table of the cumulative probabilities plotted on the probability plot be displayed. Kaplan-Meier estimates of the cumulative probabilities are also displayed, along with standard errors and confidence limits. The confidence limits can be pointwise or simultaneous, as specified by the NPINTERVALS= option.

PROBLIST=*value-list*
specifies the list of initial values for the Turnbull algorithm.

ROTATE
requests probability plots with probability scale on the horizontal axis.

SQUARE
makes the layout of the probability plots square.

TOLLIKE=*value*
specifies the criterion for convergence in the Turnbull algorithm.

TOLPROB=*value*
specifies the criterion for setting the interval probability to zero in the Turnbull algorithm.

VREF=*value-list*
requests reference lines perpendicular to the vertical axis. If (INTERSECT) is specified, a second reference line perpendicular to the horizontal axis is drawn that intersects the fit line at the same point as the vertical axis reference line. If a vertical axis reference line label is specified, the intersecting horizontal axis reference line is labeled with the horizontal axis value. See also the entries for the CVREF=, LVREF=, and VREFLABELS= options.

VREFLABELS=*'label1' . . . 'labeln'*

VREFLABEL=*'label1' . . . 'labeln'*

VREFLAB=*'label1' . . . 'labeln'*

specifies labels for the lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

WEIGHT Statement

WEIGHT *variable* **;**

If you want to use weights for each observation in the input data set, place the weights in a variable in the data set and specify the name in a WEIGHT statement. The values of the WEIGHT variable can be nonintegral and are not truncated. Observations with nonpositive or missing values for the weight variable do not contribute to the fit of the model. The WEIGHT variable multiplies the contribution to the log likelihood for each observation.

Details: LIFEREG Procedure

Missing Values

Any observation with missing values for the dependent variable is not used in the model estimation unless it is one and only one of the values in an interval specification. Also, if one of the explanatory variables or the censoring variable is missing, the observation is not used. For any observation to be used in the estimation of a model, only the variables needed in that model have to be nonmissing. Predicted values are computed for all observations with no missing explanatory variable values. If the censoring variable is missing, the CENSORED= variable in the OUT= SAS data set is also missing.

Model Specification

Main effects as well as interaction terms are allowed in the model specification, similar to the GLM procedure. For numeric variables, a main effect is a linear term equal to the value of the variable unless the variable appears in the CLASS statement. For variables listed in the CLASS statement, PROC LIFEREG creates indicator variables (variables taking the values zero or one) for every level of the variable except the last level. If there is no intercept term, the first CLASS variable has indicator variables created for all levels including the last level. The levels are ordered according to the ORDER= option. Estimates of a main effect depend upon other effects in the model and, therefore, are adjusted for the presence of other effects in the model.

Computational Method

By default, the LIFEREG procedure computes initial values for the parameters by using ordinary least squares (OLS) and ignoring censoring. This might not be the best set of starting values for a given set of data. For example, if there are extreme values in your data, the OLS fit might be excessively influenced by the extreme observations, causing an overflow or convergence problems. See Example 48.3 for one way to deal with convergence problems.

You can specify the INITIAL= option in the MODEL statement to override these starting values. You can also specify the INTERCEPT=, SCALE=, and SHAPE= options to set initial values of the intercept, scale, and shape parameters. For models with multilevel interaction effects, it is a little difficult to use the INITIAL= option to provide starting values for all parameters. In this case, you can use the INEST= data set. See the section "INEST= Data Set" on page 3050 for details. The INEST= data set overrides all previous specifications for starting values of parameters.

The rank of the design matrix $\mathbf{X}$ is estimated before the model is fit. Columns of $\mathbf{X}$ that are judged linearly dependent on other columns have the corresponding parameters set to zero. The test for linear dependence is controlled by the SINGULAR= option in the MODEL statement. Variables are included in the model in the order in which they are listed in the MODEL statement with the continuous variables included in the model before any classification variables.

The log-likelihood function is maximized by means of a ridge-stabilized Newton-Raphson algorithm. The maximized value of the log likelihood can take positive or negative values, depending on the specified model and the values of the maximum likelihood estimates of the model parameters.

If convergence of the maximum likelihood estimates is attained, a Type III chi-square test statistic is computed for each effect, testing whether there is any contribution from any of the levels of the effect. This statistic is computed as a quadratic form in the appropriate parameter estimates by using the corresponding submatrix of the asymptotic covariance matrix estimate. See Chapter 39, "The GLM Procedure," and Chapter 15, "The Four Types of Estimable Functions," for more information about Type III estimable functions.

The asymptotic covariance matrix is computed as the inverse of the observed information matrix. Note that if the NOINT option is specified and CLASS variables are used, the first CLASS variable contains a contribution from an intercept term. The results are displayed in an ODS table named "Type3Analysis."

Chi-square tests for individual parameters are Wald tests based on the observed information matrix and the parameter estimates. If an effect has a single degree of freedom in the parameter estimates table, the chi-square test for this parameter is equivalent to the Type III test for this effect.

Before SAS 8.2, a multiple-degree-of-freedom statistic was computed for each effect to test for contribution from any level of the effect. In general, the Type III test statistic in a main-effect-only model (no interaction terms) will be equal to the previously computed effect statistic, unless there are collinearities among the effects. If there are collinearities, the Type III statistic will adjust for them, and the value of the Type III statistic and the number of degrees of freedom might not be equal to those of the previous effect statistic.

Suppose there are n observations from the model $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \sigma\boldsymbol{\epsilon}$ (or $\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{O} + \sigma\boldsymbol{\epsilon}$ if there

is an offset variable), where $\mathbf{X}$ is an $n \times k$ matrix of covariate values (including the intercept), $\mathbf{y}$ is a vector of responses, $\mathbf{O}$ is a vector of offset variable values, and $\boldsymbol{\epsilon}$ is a vector of errors with survival function S, cumulative distribution function F, and probability density function f. That is, $S(t) = \Pr(\epsilon_i > t)$, $F(t) = \Pr(\epsilon_i \leq t)$, and $f(t) = dF(t)/dt$, where ϵ_i is a component of the error vector. Then, if all the responses are observed, the log likelihood, L, can be written as

$$L = \sum \log\left(\frac{f(u_i)}{\sigma}\right)$$

where $u_i = \frac{1}{\sigma}(y_i - \mathbf{x}_i'\boldsymbol{\beta})$.

If some of the responses are left, right, or interval censored, the log likelihood can be written as

$$L = \sum \log\left(\frac{f(u_i)}{\sigma}\right) + \sum \log\left(S(u_i)\right) + \sum \log\left(F(u_i)\right) + \sum \log\left(F(u_i) - F(v_i)\right)$$

with the first sum over uncensored observations, the second sum over right-censored observations, the third sum over left-censored observations, the last sum over interval-censored observations, and

$$v_i = \frac{1}{\sigma}(z_i - \mathbf{x}_i'\boldsymbol{\beta})$$

where z_i is the lower end of a censoring interval.

If the response is specified in the binomial format, *events/trials*, then the log-likelihood function is

$$L = \sum r_i \log(P_i) + (n_i - r_i)\log(1 - P_i)$$

where r_i is the number of events and n_i is the number of trials for the ith observation. In this case, $P_i = 1 - F(-\mathbf{x}_i'\boldsymbol{\beta})$. For the symmetric distributions, logistic and normal, this is the same as $F(\mathbf{x}_i'\boldsymbol{\beta})$. Additional information about censored and limited dependent variable models can be found in Kalbfleisch and Prentice (1980) and Maddala (1983).

The estimated covariance matrix of the parameter estimates is computed as the negative inverse of $\mathbf{I}$, which is the information matrix of second derivatives of L with respect to the parameters evaluated at the final parameter estimates. If $\mathbf{I}$ is not positive definite, a positive-definite submatrix of $\mathbf{I}$ is inverted, and the remaining rows and columns of the inverse are set to zero. If some of the parameters, such as the scale and intercept, are restricted, the corresponding elements of the estimated covariance matrix are set to zero. The standard error estimates for the parameter estimates are taken as the square roots of the corresponding diagonal elements.

For restrictions placed on the intercept, scale, and shape parameters, one-degree-of-freedom Lagrange multiplier test statistics are computed. These statistics are computed as

$$\chi^2 = \frac{g^2}{V}$$

where g is the derivative of the log likelihood with respect to the restricted parameter at the restricted maximum and

$$V = \mathbf{I}_{11} - \mathbf{I}_{12}\mathbf{I}_{22}^{-1}\mathbf{I}_{21}$$

where the 1 subscripts refer to the restricted parameter and the 2 subscripts refer to the unrestricted parameters. The information matrix is evaluated at the restricted maximum. These statistics are

asymptotically distributed as chi-squares with one degree of freedom under the null hypothesis that the restrictions are valid, provided that some regularity conditions are satisfied. Refer to Rao (1973, p. 418) for a more complete discussion. It is possible for these statistics to be missing if the observed information matrix is not positive definite. Higher-degree-of-freedom tests for multiple restrictions are not currently computed.

A Lagrange multiplier test statistic is computed to test this constraint. Notice that this test statistic is comparable to the Wald test statistic for testing that the scale is one. The Wald statistic is the result of squaring the difference of the estimate of the scale parameter from one and dividing this by the square of its estimated standard error.

Supported Distributions

For most distributions, the baseline survival function (S) and the probability density function(f) are listed for the additive random disturbance (y_0 or $\log(T_0)$) with location parameter μ and scale parameter σ. See the section "Overview: LIFEREG Procedure" on page 2992 for more information. These distributions apply when the log of the response is modeled (this is the default analysis). The corresponding survival function (G) and its density function (g) are given for the untransformed baseline distribution (T_0).

For the normal and logistic distributions, the response is not log transformed by PROC LIFEREG, and the survival functions and probability density functions listed apply to the untransformed response.

For example, for the WEIBULL distribution, $S(w)$ and $f(w)$ are the survival function and the probability density function for the extreme-value distribution (distribution of the log of the response), while $G(t)$ and $g(t)$ are the survival function and the probability density function of a Weibull distribution (using the untransformed response).

The chosen baseline functions define the meaning of the intercept, scale, and shape parameters. Only the gamma distribution has a free shape parameter in the following parameterizations. Notice that some of the distributions do not have mean zero and that σ is not, in general, the standard deviation of the baseline distribution.

For the Weibull distribution, the accelerated failure time model is also a proportional-hazards model. However, the parameterization for the covariates differs by a multiple of the scale parameter from the parameterization commonly used for the proportional hazards model.

The distributions supported in the LIFEREG procedure follow. If there are no covariates in the model, μ = Intercept in the output; otherwise, $\mu = \mathbf{x}'\boldsymbol{\beta}$. σ = Scale in the output.

Exponential

$$
\begin{aligned}
S(w) &= \exp(-\exp(w-\mu)) \\
f(w) &= \exp(w-\mu)\exp(-\exp(w-\mu)) \\
G(t) &= \exp(-\alpha t) \\
g(t) &= \alpha \exp(-\alpha t)
\end{aligned}
$$

where $\exp(-\mu) = \alpha$.

Generalized Gamma

$S(w) = S'(u)$, $f(w) = \sigma^{-1} f'(u)$, $G(t) = G'(v)$, $g(t) = \frac{v}{t\sigma} g'(v)$, $u = \frac{w-\mu}{\sigma}$, $v = \exp(\frac{\log(t)-\mu}{\sigma})$, and

$$
\begin{aligned}
S'(u) &= \begin{cases} 1 - \frac{\Gamma(\delta^{-2}, \delta^{-2}\exp(\delta u))}{\Gamma(\delta^{-2})} & \text{if } \delta > 0 \\ \frac{\Gamma(\delta^{-2}, \delta^{-2}\exp(\delta u))}{\Gamma(\delta^{-2})} & \text{if } \delta < 0 \end{cases} \\
f'(u) &= \frac{|\delta|}{\Gamma\left(\delta^{-2}\right)} \left(\delta^{-2}\exp(\delta u)\right)^{\delta^{-2}} \exp\left(-\exp(\delta u)\delta^{-2}\right) \\
G'(v) &= \begin{cases} 1 - \frac{\Gamma(\delta^{-2}, \delta^{-2}v^{\delta})}{\Gamma(\delta^{-2})} & \text{if } \delta > 0 \\ \frac{\Gamma(\delta^{-2}, \delta^{-2}v^{\delta})}{\Gamma(\delta^{-2})} & \text{if } \delta < 0 \end{cases} \\
g'(v) &= \frac{|\delta|}{v\Gamma\left(\delta^{-2}\right)} \left(\delta^{-2}v^{\delta}\right)^{\delta^{-2}} \exp\left(-v^{\delta}\delta^{-2}\right)
\end{aligned}
$$

where $\Gamma(a)$ denotes the complete gamma function, $\Gamma(a, z)$ denotes the incomplete gamma function, and δ is a free shape parameter. The δ parameter is called Shape by PROC LIFEREG. See Lawless (2003, p. 240), and Klein and Moeschberger (1997, p. 386) for a description of the generalized gamma distribution.

Logistic

$$
\begin{aligned}
S(w) &= \left(1 + \exp\left(\frac{w-\mu}{\sigma}\right)\right)^{-1} \\
f(w) &= \frac{\exp\left(\frac{w-\mu}{\sigma}\right)}{\sigma\left(1 + \exp\left(\frac{w-\mu}{\sigma}\right)\right)^2}
\end{aligned}
$$

Loglogistic

$$
\begin{aligned}
S(w) &= \left(1 + \exp\left(\frac{w-\mu}{\sigma}\right)\right)^{-1} \\
f(w) &= \frac{\exp\left(\frac{w-\mu}{\sigma}\right)}{\sigma\left(1 + \exp\left(\frac{w-\mu}{\sigma}\right)\right)^2} \\
G(t) &= \frac{1}{1+\alpha t^{\gamma}} \\
g(t) &= \frac{\alpha\gamma t^{\gamma-1}}{(1+\alpha t^{\gamma})^2}
\end{aligned}
$$

where $\gamma = 1/\sigma$ and $\alpha = \exp(-\mu/\sigma)$.

Lognormal

$$
\begin{aligned}
S(w) &= 1 - \Phi\left(\frac{w-\mu}{\sigma}\right) \\
f(w) &= \frac{1}{\sqrt{2\pi}\sigma}\exp\left(-\frac{1}{2}\left(\frac{w-\mu}{\sigma}\right)^2\right) \\
G(t) &= 1 - \Phi\left(\frac{\log(t)-\mu}{\sigma}\right) \\
g(t) &= \frac{1}{\sqrt{2\pi}\sigma t}\exp\left(-\frac{1}{2}\left(\frac{\log(t)-\mu}{\sigma}\right)^2\right)
\end{aligned}
$$

where Φ is the cumulative distribution function for the normal distribution.

Normal

$$
\begin{aligned}
S(w) &= 1 - \Phi\left(\frac{w-\mu}{\sigma}\right) \\
f(w) &= \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{w-\mu}{\sigma}\right)^2\right)
\end{aligned}
$$

where Φ is the cumulative distribution function for the normal distribution.

Weibull

$$
\begin{aligned}
S(w) &= \exp\left(-\exp\left(\frac{w-\mu}{\sigma}\right)\right) \\
f(w) &= \frac{1}{\sigma} \exp\left(\frac{w-\mu}{\sigma}\right) \exp\left(-\exp\left(\frac{w-\mu}{\sigma}\right)\right) \\
G(t) &= \exp\left(-\alpha t^{\gamma}\right) \\
g(t) &= \gamma \alpha t^{\gamma-1} \exp\left(-\alpha t^{\gamma}\right)
\end{aligned}
$$

where $\sigma = 1/\gamma$ and $\alpha = \exp(-\mu/\sigma)$.

If your parameterization is different from the ones shown here, you can still use the procedure to fit your model. For example, a common parameterization for the Weibull distribution is

$$
\begin{aligned}
g(t;\lambda,\beta) &= \left(\frac{\beta}{\lambda}\right)\left(\frac{t}{\lambda}\right)^{\beta-1} \exp\left(-\left(\frac{t}{\lambda}\right)^{\beta}\right) \\
G(t;\lambda,\beta) &= \exp\left(-\left(\frac{t}{\lambda}\right)^{\beta}\right)
\end{aligned}
$$

so that $\lambda = \exp(\mu)$ and $\beta = 1/\sigma$.

Again note that the expected value of the baseline log response is, in general, not zero and that the distributions are not symmetric in all cases. Thus, for a given set of covariates, $\mathbf{x}$, the expected value of the log response is not always $\mathbf{x}'\boldsymbol{\beta}$.

Some relations among the distributions are as follows:

- The gamma with Shape=1 is a Weibull distribution.
- The gamma with Shape=0 is a lognormal distribution.
- The Weibull with Scale=1 is an exponential distribution.

Predicted Values

For a given set of covariates, $\mathbf{x}$ (including the intercept term), the pth quantile of the log response, y_p, is given by

$$y_p = \mathbf{x}'\boldsymbol{\beta} + \sigma u_p$$

if no offset variable has been specified, or

$$y_p = \mathbf{x}'\boldsymbol{\beta} + \mathrm{o} + \sigma u_p$$

for a given value o of an offset variable, where u_p is the pth quantile of the baseline distribution. The estimated quantile is computed by replacing the unknown parameters with their estimates, including any shape parameters on which the baseline distribution might depend. The estimated quantile of the original response is obtained by taking the exponential of the estimated log quantile unless the NOLOG option is specified in the preceding MODEL statement.

The following table shows how u_p is computed from the baseline distribution $F(u)$:

Table 48.3 Baseline Probability Functions and u_p

Distribution	$F(u)$		u_p
Exponential	$1 - \exp(-\exp(u))$		$\log(-\log(1-p))$
Generalized Gamma	$\frac{\Gamma(\delta^{-2},\delta^{-2}\exp(\delta u))}{\Gamma(\delta^{-2})}$	if $\delta > 0$	$F^{-1}(p)$
	$1 - \frac{\Gamma(\delta^{-2},\delta^{-2}\exp(\delta u))}{\Gamma(\delta^{-2})}$	if $\delta < 0$	
Logistic	$1 - (1+\exp(u))^{-1}$		$\log(p/(1-p))$
Loglogistic	$1 - (1+\exp(u))^{-1}$		$\log(p/(1-p))$
Lognormal	$\Phi(u)$		$\Phi^{-1}(p)$
Normal	$\Phi(u)$		$\Phi^{-1}(p)$
Weibull	$1 - \exp(-\exp(u))$		$\log(-\log(1-p))$

For the generalized gamma distribution, u_p is computed numerically.

The standard errors of the quantile estimates are computed using the estimated covariance matrix of the parameter estimates and a Taylor series expansion of the quantile estimate. The standard error is computed as

$$\mathrm{STD} = \sqrt{\mathbf{z}'\mathbf{V}\mathbf{z}}$$

where $\mathbf{V}$ is the estimated covariance matrix of the parameter vector $(\beta', \sigma, \delta)'$, and $\mathbf{z}$ is the vector

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \hat{u}_p \\ \hat{\sigma}\frac{\partial u_p}{\partial \delta} \end{bmatrix}$$

where δ is the vector of the shape parameters. Unless the NOLOG option is specified, this standard error estimate is converted into a standard error estimate for $\exp(y_p)$ as $\exp(\hat{y}_p)\mathrm{STD}$. It might be

more desirable to compute confidence limits for the log response and convert them back to the original response variable than to use the standard error estimates for $\exp(y_p)$ directly. See Example 48.1 for a 90% confidence interval of the response constructed by exponentiating a confidence interval for the log response.

The variable CDF is computed as

$$\text{CDF}_i = F(u_i)$$

where the residual is defined by

$$u_i = \left(\frac{y_i - \mathbf{x}_i' \mathbf{b}}{\hat{\sigma}} \right)$$

and F is the baseline cumulative distribution function.

Confidence Intervals

Confidence intervals are computed for all model parameters and are reported in the "Analysis of Parameter Estimates" table. The confidence coefficient can be specified with the ALPHA=α MODEL statement option, resulting in a $(1 - \alpha) \times 100\%$ two-sided confidence coefficient. The default confidence coefficient is 95%, corresponding to $\alpha = 0.05$.

Regression Parameters

A two-sided $(1 - \alpha) \times 100\%$ confidence interval $[\beta_{iL}, \quad \beta_{iU}]$ for the regression parameter β_i is based on the asymptotic normality of the maximum likelihood estimator $\hat{\beta}_i$ and is computed by

$$\beta_{iL} = \hat{\beta}_i - z_{1-\alpha/2}(\text{SE}_{\hat{\beta}_i})$$

$$\beta_{iU} = \hat{\beta}_i + z_{1-\alpha/2}(\text{SE}_{\hat{\beta}_i})$$

where $\text{SE}_{\hat{\beta}_i}$ is the estimated standard error of $\hat{\beta}_i$, and z_p is the $p \times 100\%$ percentile of the standard normal distribution.

Scale Parameter

A two-sided $(1-\alpha) \times 100\%$ confidence interval $[\sigma_L, \quad \sigma_U]$ for the scale parameter σ in the location-scale model is based on the asymptotic normality of the logarithm of the maximum likelihood estimator $\log(\hat{\sigma})$, and is computed by

$$\sigma_L = \hat{\sigma} / \exp[z_{1-\alpha/2}(\text{SE}_{\hat{\sigma}})/\hat{\sigma}]$$

$$\sigma_U = \hat{\sigma} \exp[z_{1-\alpha/2}(\mathrm{SE}_{\hat{\sigma}})/\hat{\sigma}]$$

Refer to Meeker and Escobar (1998) for more information.

Weibull Scale and Shape Parameters

The Weibull distribution scale parameter η and shape parameter β are obtained by transforming the extreme-value location parameter μ and scale parameter σ:

$$\eta = \exp(\mu)$$

$$\beta = 1/\sigma$$

Consequently, two-sided $(1 - \alpha) \times 100\%$ confidence intervals for the Weibull scale and shape parameters are computed as

$$[\eta_L, \ \eta_U] = [\exp(\mu_L), \ \exp(\mu_U)]$$

$$[\beta_L, \ \beta_U] = [1/\sigma_U, \ 1/\sigma_L]$$

Gamma Shape Parameter

A two-sided $(1 - \alpha) \times 100\%$ confidence interval for the three-parameter gamma shape parameter δ is computed by

$$[\delta_L, \ \delta_U] = [\hat{\delta} - z_{1-\alpha/2}(\mathrm{SE}_{\hat{\delta}}), \ \hat{\delta} + z_{1-\alpha/2}(\mathrm{SE}_{\hat{\delta}})]$$

Fit Statistics

Suppose that the model contains p parameters and that n observations are used in model fitting. The fit criteria displayed by the LIFEREG procedure are calculated as follows:

- −2 log likelihood:

  $$-2\log(\mathrm{L})$$

  where L is the maximized likelihood for the model.

- Akaike information criterion:

  $$\mathrm{AIC} = -2\log(\mathrm{L}) + 2p$$

- corrected Akaike information criterion:

$$\text{AICC} = \text{AIC} + \frac{2p(p+1)}{n-p-1}$$

- Bayesian information criterion:

$$\text{BIC} = -2\log(\text{L}) + p\log(n)$$

Refer to Akaike (1981, 1979) for details of AIC and BIC. Refer to Simonoff (2003) for a discussion of using AIC, AICC, and BIC in statistical modeling.

Probability Plotting

Probability plots are useful tools for the display and analysis of lifetime data. Probability plots use an inverse distribution scale so that a cumulative distribution function (CDF) plots as a straight line. A nonparametric estimate of the CDF of the lifetime data will plot approximately as a straight line, thus providing a visual assessment of goodness of fit.

You can use the PROBPLOT statement in PROC LIFEREG to create probability plots of data that are complete, right censored, interval censored, or a combination of censoring types (arbitrarily censored). A line representing the maximum likelihood fit from the MODEL statement and pointwise parametric confidence bands for the cumulative probabilities are also included in the plot.

A random variable Y belongs to a *location-scale* family of distributions if its CDF F is of the form

$$Pr\{Y \leq y\} = F(y) = G\left(\frac{y-\mu}{\sigma}\right)$$

where μ is the location parameter and σ is the scale parameter. Here, G is a CDF that cannot depend on any unknown parameters, and G is the CDF of Y if $\mu = 0$ and $\sigma = 1$. For example, if Y is a normal random variable with mean μ and standard deviation σ,

$$G(u) = \Phi(u) = \int_{-\infty}^{u} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{u^2}{2}\right) du$$

and

$$F(y) = \Phi\left(\frac{y-\mu}{\sigma}\right)$$

The normal, extreme-value, and logistic distributions are location-scale models. The three-parameter gamma distribution is a location-scale model if the shape parameter δ is fixed. If T has a lognormal, Weibull, or log-logistic distribution, then $\log(T)$ has a distribution that is a location-scale model. Probability plots are constructed for lognormal, Weibull, and log-logistic distributions by using $\log(T)$ instead of T in the plots.

Let $y_{(1)} \leq y_{(2)} \leq \ldots \leq y_{(n)}$ be ordered observations of a random sample with distribution function $F(y)$. A probability plot is a plot of the points $y_{(i)}$ against $m_i = G^{-1}(a_i)$, where $a_i = \hat{F}(y_i)$ is an estimate of the CDF $F(y_{(i)}) = G\left(\frac{y_{(i)}-\mu}{\sigma}\right)$. The nonparametric CDF estimates a_i are sometimes called *plotting positions*. The axis on which the points m_i are plotted is usually labeled with a probability scale (the scale of a_i).

If F is one of the location-scale distributions, then y is the lifetime; otherwise, the log of the lifetime is used to transform the distribution to a location-scale model.

If the data actually have the stated distribution, then $\hat{F} \approx F$,

$$m_i = G^{-1}(\hat{F}(y_i)) \approx G^{-1}\left(G\left(\frac{y_{(i)}-\mu}{\sigma}\right)\right) = \frac{y_{(i)}-\mu}{\sigma}$$

and points $(y_{(i)}, m_i)$ should fall approximately in a straight line.

There are several ways to compute the nonparametric CDF estimates used in probability plots from lifetime data. These are discussed in the next two sections.

Complete and Right-Censored Data

The censoring times must be taken into account when you compute plotting positions for right-censored data. The modified Kaplan-Meier method described in the following section is the default method for computing nonparametric CDF estimates for display on probability plots. Refer to Abernethy (1996), Meeker and Escobar (1998), and Nelson (1982) for discussions of the methods described in the following sections.

Expected Ranks, Kaplan-Meier, and Modified Kaplan-Meier Methods

Let $y_{(1)} \leq y_{(2)} \leq \ldots \leq y_{(n)}$ be ordered observations of a random sample including failure times and censor times. Order the data in increasing order. Label all the data with reverse ranks r_i, with $r_1 = n, \ldots, r_n = 1$. For the lifetime (not censoring time) corresponding to reverse rank r_i, compute the survival function estimate

$$S_i = \left[\frac{r_i}{r_i + 1}\right] S_{i-1}$$

with $S_0 = 1$. The expected rank plotting position is computed as $a_i = 1 - S_i$. The option PPOS=EXPRANK specifies the expected rank plotting position.

For the Kaplan-Meier method,

$$S_i = \left[\frac{r_i - 1}{r_i}\right] S_{i-1}$$

The Kaplan-Meier plotting position is then computed as $a'_i = 1 - S_i$. The option PPOS=KM specifies the Kaplan-Meier plotting position.

For the modified Kaplan-Meier method, use

$$S'_i = \frac{S_i + S_{i-1}}{2}$$

where S_i is computed from the Kaplan-Meier formula with $S_0 = 1$. The plotting position is then computed as $a_i'' = 1 - S_i'$. The option PPOS=MKM specifies the modified Kaplan-Meier plotting position. If the PPOS option is not specified, the modified Kaplan-Meier plotting position is used as the default method.

For complete samples, $a_i = i/(n+1)$ for the expected rank method, $a_i' = i/n$ for the Kaplan-Meier method, and $a_i'' = (i - 0.5)/n$ for the modified Kaplan-Meier method. If the largest observation is a failure for the Kaplan-Meier estimator, then $F_n = 1$ and the point is not plotted.

Median Ranks

Let $y_{(1)} \leq y_{(2)} \leq \ldots \leq y_{(n)}$ be ordered observations of a random sample including failure times and censor times. A failure order number j_i is assigned to the ith failure: $j_i = j_{i-1} + \Delta$, where $j_0 = 0$. The increment Δ is initially 1 and is modified when a censoring time is encountered in the ordered sample. The new increment is computed as

$$\Delta = \frac{(n+1) - \text{ previous failure order number}}{1 + \text{ number of items beyond previous censored item}}$$

The plotting position is computed for the ith failure time as

$$a_i = \frac{j_i - 0.3}{n + 0.4}$$

For complete samples, the failure order number j_i is equal to i, the order of the failure in the sample. In this case, the preceding equation for a_i is an approximation of the median plotting position computed as the median of the ith-order statistic from the uniform distribution on (0, 1). In the censored case, j_i is not necessarily an integer, but the preceding equation still provides an approximation to the median plotting position. The PPOS=MEDRANK option specifies the median rank plotting position.

Arbitrarily Censored Data

The LIFEREG procedure can create probability plots for data that consist of combinations of exact, left-censored, right-censored, and interval-censored lifetimes—that is, arbitrarily censored data. The LIFEREG procedure uses an iterative algorithm developed by Turnbull (1976) to compute a nonparametric maximum likelihood estimate of the cumulative distribution function for the data. Since the technique is maximum likelihood, standard errors of the cumulative probability estimates are computed from the inverse of the associated Fisher information matrix. This algorithm is an example of the expectation-maximization (EM) algorithm. The default initial estimate assigns equal probabilities to each interval. You can specify different initial values with the PROBLIST= option. Convergence is determined if the change in the log likelihood between two successive iterations is less than delta, where the default value of delta is 10^{-8}. You can specify a different value for delta with the TOLLIKE= option. Iterations will be terminated if the algorithm does not converge after a fixed number of iterations. The default maximum number of iterations is 1000. Some data might require more iterations for convergence. You can specify the maximum allowed number of iterations with the MAXITEM= option in the PROBPLOT statement. The iteration history of the log likelihood is displayed if you specify the ITPRINTEM option. The iteration history of the

estimated interval probabilities are also displayed if you specify both options ITPRINTEM and PRINTPROBS.

If an interval probability is smaller than a tolerance (10^{-6} by default) after convergence, the probability is set to zero, the interval probabilities are renormalized so that they add to one, and iterations are restarted. Usually the algorithm converges in just a few more iterations. You can change the default value of the tolerance with the TOLPROB= option. You can specify the NOPOLISH option to avoid setting small probabilities to zero and restarting the algorithm.

If you specify the ITPRINTEM option, a table summarizing the Turnbull estimate of the interval probabilities is displayed. The columns labeled "Reduced Gradient" and "Lagrange Multiplier" are used in checking final convergence of the maximum likelihood estimate. The Lagrange multipliers must all be greater than or equal to zero, or the solution is not maximum likelihood. Refer to Gentleman and Geyer (1994) for more details of the convergence checking. Also refer to Meeker and Escobar (1998, Chapter 3) for more information.

See Example 48.6 for an illustration.

Nonparametric Confidence Intervals

You can use the PPOUT option in the PROBPLOT statement to create a table containing the nonparametric CDF estimates computed by the selected method, Kaplan-Meier CDF estimates, standard errors of the Kaplan-Meier estimator, and nonparametric confidence limits for the CDF. The confidence limits are either pointwise or simultaneous, depending on the value of the NPINTERVALS= option in the PROBPLOT statement. The method used in the LIFEREG procedure for computation of approximate pointwise and simultaneous confidence intervals for cumulative failure probabilities relies on the Kaplan-Meier estimator of the cumulative distribution function of failure time and approximate standard deviation of the Kaplan-Meier estimator. For the case of arbitrarily censored data, the Turnbull algorithm, discussed previously, provides an extension of the Kaplan-Meier estimator. Both the Kaplan-Meier and the Turnbull estimators provide an estimate of the standard error of the CDF estimator, $\text{se}_{\hat{F}}$, that is used in computing confidence intervals.

Pointwise Confidence Intervals

Approximate $(1-\alpha)100\%$ pointwise confidence intervals are computed as in Meeker and Escobar (1998, Section 3.6) as

$$[F_L,\ \ F_U] = \left[\frac{\hat{F}}{\hat{F}+(1-\hat{F})w},\ \ \frac{\hat{F}}{\hat{F}+(1-\hat{F})/w}\right]$$

where

$$w = \exp\left[\frac{z_{1-\alpha/2}\text{se}_{\hat{F}}}{(\hat{F}(1-\hat{F}))}\right]$$

where z_p is the pth quantile of the standard normal distribution.

Simultaneous Confidence Intervals

Approximate $(1-\alpha)100\%$ simultaneous confidence bands valid over the lifetime interval (t_a, t_b) are computed as the "Equal Precision" case of Nair (1984) and Meeker and Escobar (1998, Section 3.8) as

$$[F_L, \quad F_U] = \left[\frac{\hat{F}}{\hat{F} + (1-\hat{F})w}, \quad \frac{\hat{F}}{\hat{F} + (1-\hat{F})/w} \right]$$

where

$$w = \exp\left[\frac{e_{a,b,1-\alpha/2} \mathrm{se}_{\hat{F}}}{(\hat{F}(1-\hat{F}))} \right]$$

where the factor $x = e_{a,b,1-\alpha/2}$ is the solution of

$$x \exp(-x^2/2) \log\left[\frac{(1-a)b}{(1-b)a} \right] / \sqrt{8\pi} = \alpha/2$$

The time interval (t_a, t_b) over which the bands are valid depends in a complicated way on the constants a and b defined in Nair (1984), $0 < a < b < 1$. The constants a and b are chosen by default so that the confidence bands are valid between the lowest and highest times corresponding to failures in the case of multiply censored data, or to the lowest and highest intervals for which probabilities are computed for arbitrarily censored data. You can optionally specify a and b directly with the NPINTERVALS=SIMULTANEOUS(a, b) option in the PROBPLOT statement.

INEST= Data Set

If specified, the INEST= data set specifies initial estimates for all the parameters in the model. The INEST= data set must contain the intercept variable (named Intercept) and all independent variables in the MODEL statement.

If BY processing is used, the INEST= data set should also include the BY variables, and there must be at least one observation for each BY group. If there is more than one observation in one BY group, the first observation read is used for that BY group.

If the INEST= data set also contains the _TYPE_ variable, only observations with _TYPE_ value 'PARMS' are used as starting values. Combining the INEST= data set and the MAXITER= option in the MODEL statement, partial scoring can be done, such as predicting on a validation data set by using the model built from a training data set.

You can specify starting values for the iterative algorithm in the INEST= data set. This data set overwrites the INITIAL= option in the MODEL statement, which is a little difficult to use for

models including multilevel interaction effects. The INEST= data set has the same structure as the OUTEST= data set but is not required to have all the variables or observations that appear in the OUTEST= data set. One simple use of the INEST= option is passing the previous OUTEST= data set directly to the next model as an INEST= data set, assuming that the two models have the same parameterization. See Example 48.3 for an illustration.

OUTEST= Data Set

The OUTEST= data set contains parameter estimates and the log likelihood for the model. You can specify a label in the MODEL statement to distinguish between the estimates for different models fit with the LIFEREG procedure. If the COVOUT option is specified, the OUTEST= data set also contains the estimated covariance matrix of the parameter estimates. Note that, if the LIFEREG procedure does not converge, the parameter estimates are set to missing in the OUTEST data set.

The OUTEST= data set contains all variables specified in the MODEL statement and the BY statement. One observation consists of parameter values for the model with the dependent variable having the value -1. If the COVOUT option is specified, there are additional observations containing the rows of the estimated covariance matrix. For these observations, the dependent variable contains the parameter estimate for the corresponding row variable. The following variables are also added to the data set:

MODEL	a character variable containing the label of the MODEL statement, if present. Otherwise, the variable's value is blank.
NAME	a character variable containing the name of the dependent variable for the parameter estimates observations or the name of the row for the covariance matrix estimates
TYPE	a character variable containing the type of the observation, either PARMS for parameter estimates or COV for covariance estimates
DIST	a character variable containing the name of the distribution modeled
LNLIKE	a numeric variable containing the last computed value of the log likelihood
INTERCEPT	a numeric variable containing the intercept parameter estimates and covariances
SCALE	a numeric variable containing the scale parameter estimates and covariances
SHAPE1	a numeric variable containing the first shape parameter estimates and covariances if the specified distribution has additional shape parameters

Any BY variables specified are also added to the OUTEST= data set.

XDATA= Data Set

The XDATA= data set is used for plotting the predicted probability when there are covariates specified in a MODEL statement and a probability plot is specified with a PROBPLOT statement. See Example 48.4 for an illustration.

The XDATA= data set is an input SAS data set that contains values for all the independent variables in the MODEL statement and variables in the CLASS statement.The XDATA= data set has the same structure as the DATA= data set but is not required to have all the variables or observations that appear in the DATA= data set.

The XDATA= data set must contain all the independent variables in the MODEL statement and variables in the CLASS statement. Even though variables in the CLASS statement might not be used, valid values are required for these variables in the XDATA= data set. Missing values are not allowed. Missing values are not allowed in the XDATA= data set for any of the independent variables, either. Missing values are allowed for the dependent variables and other variables if they are included in the XDATA= data set.

If BY processing is used, the XDATA= data set should also include the BY variables, and there must be at least one valid observation for each BY group. If there is more than one valid observation in a BY group, the last one read is used for that BY group.

If there is no XDATA= data set in the PROC LIFEREG statement, by default, the LIFEREG procedure will use the overall mean for effects containing a continuous variable (or variables) and the highest level of a single classification variable as reference level. The rules are summarized as follows:

- If the effect contains a continuous variable (or variables), the overall mean of this effect (not the variables) is used.
- If the effect is a single classification variable, the highest level of the variable is used.

Computational Resources

Let p be the number of parameters estimated in the model. The minimum working space (in bytes) needed is

$$16p^2 + 100p$$

However, if sufficient space is available, the input data set is also kept in memory; otherwise, the input data set is reread for each evaluation of the likelihood function and its derivatives, with the resulting execution time of the procedure substantially increased.

Let n be the number of observations used in the model estimation. Each evaluation of the likelihood function and its first and second derivatives requires $O(np^2)$ multiplications and additions, n individual function evaluations for the log density or log distribution function, and n evaluations of the first and second derivatives of the function. The calculation of each updating step from the gradient and Hessian requires $O(p^3)$ multiplications and additions. The $O(v)$ notation means that, for large values of the argument, v, $O(v)$ is approximately a constant times v.

Bayesian Analysis

Gibbs Sampling

This section provides details about Bayesian analysis by Gibbs sampling in the location-scale models for survival data available in PROC LIFEREG. See the section "Gibbs Sampler" on page 154 for a general discussion of Gibbs sampling. PROC LIFEREG fits parametric location-scale survival models. That is, the probability density of the response Y can expressed in the general form

$$f(y) = g\left(\frac{y-\mu}{\sigma}\right)$$

where $Y = \log(T)$ for lifetimes T. The function g determines the specific distribution. The location parameter μ_i is modeled through regression parameters as $\mu_i = \mathbf{x}_i'\boldsymbol{\beta}$. The LIFEREG procedure can provide Bayesian estimates of the regression parameters and σ. The OUTPUT and PROBPLOT statements, if specified, are ignored. The PLOTS=PROBPLOT option in the PROC LIFEREG statement and the CORRB and COVB options in the MODEL statement are also ignored.

For the Weibull distribution, you can specify that Gibbs sampling be performed on the Weibull shape parameter $\beta = \sigma^{-1}$ instead of the scale parameter σ by specifying a prior distribution for the shape parameter with the WEIBULLSHAPEPRIOR= option. In addition, if there are no covariates in the model, you can specify Gibbs sampling on the Weibull scale parameter $\alpha = \exp(\mu)$, where μ is the intercept term, with the WEIBULLSCALEPRIOR= option.

In the case of the exponential distribution with no covariates, you can specify Gibbs sampling on the exponential scale parameter $\alpha = \exp(\mu)$, where μ is the intercept term, with the EXPSCALEPRIOR= option.

Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)\prime$ be the parameter vector. For location-scale models, the θ_i's are the regression coefficients β_i's and the scale parameter σ. In the case of the three-parameter gamma distribution, there is an additional gamma shape parameter τ. Let $L(D|\boldsymbol{\theta})$ be the likelihood function, where D is the observed data. Let $\pi(\boldsymbol{\theta})$ be the prior distribution. The full conditional distribution of $[\theta_i|\theta_j, i \neq j]$ is proportional to the joint distribution; that is,

$$\pi(\theta_i|\theta_j, i \neq j, D) \propto L(D|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

For instance, the one-dimensional conditional distribution of θ_1 given $\theta_j = \theta_j^*, 2 \leq j \leq k$, is computed as

$$\pi(\theta_1|\theta_j = \theta_j^*, 2 \leq j \leq k, D) = L(D|(\boldsymbol{\theta} = (\theta_1, \theta_2^*, \ldots, \theta_k^*)')p(\boldsymbol{\theta} = (\theta_1, \theta_2^*, \ldots, \theta_k^*)')$$

Suppose you have a set of arbitrary starting values $\{\theta_1^{(0)}, \ldots, \theta_k^{(0)}\}$. Using the ARMS (adaptive rejection Metropolis sampling) algorithm of Gilks and Wild (1992) and Gilks, Best, and Tan (1995), you can do the following:

draw $\theta_1^{(1)}$ from $[\theta_1|\theta_2^{(0)}, \ldots, \theta_k^{(0)}]$

draw $\theta_2^{(1)}$ from $[\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, \ldots, \theta_k^{(0)}]$

...

draw $\theta_k^{(1)}$ from $[\theta_k|\theta_1^{(1)},\ldots,\theta_{k-1}^{(1)}]$

This completes one iteration of the Gibbs sampler. After one iteration, you have $\{\theta_1^{(1)},\ldots,\theta_k^{(1)}\}$. After n iterations, you have $\{\theta_1^{(n)},\ldots,\theta_k^{(n)}\}$. PROC LIFEREG implements the ARMS algorithm based on a program provided by Gilks (2003) to draw a sample from a full conditional distribution. See the section "Assessing Markov Chain Convergence" on page 156 for information about assessing the convergence of the chain of posterior samples.

You can output these posterior samples into a SAS data set. The following option in the BAYES statement outputs the posterior samples into the SAS data set Post:

```
OUTPOST=Post
```

The data set also includes the variable LogPost, representing the log of the posterior log likelihood.

Priors for Model Parameters

The model parameters are the regression coefficients and the dispersion parameter (or the precision or scale), if the model has one. The priors for the dispersion parameter and the priors for the regression coefficients are assumed to be independent, while you can have a joint multivariate normal prior for the regression coefficients.

Scale and Shape Parameters

Gamma Prior The gamma distribution $G(a,b)$ has a pdf

$$f_{a,b}(u) = \frac{b(bu)^{a-1}e^{-bu}}{\Gamma(a)}, \qquad u > 0$$

where a is the shape parameter and b is the inverse-scale parameter. The mean is $\frac{a}{b}$ and the variance is $\frac{a}{b^2}$.

Improper Prior The joint prior density is given by

$$p(u) \propto u^{-1}, \qquad u > 0$$

Regression Coefficients

Let $\boldsymbol{\beta}$ be the regression coefficients.

Normal Prior Assume $\boldsymbol{\beta}$ has a multivariate normal prior with mean vector $\boldsymbol{\beta}_0$ and covariance matrix $\boldsymbol{\Sigma}_0$. The joint prior density is given by

$$p(\boldsymbol{\beta}) \propto e^{-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}_0)'\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_0)}$$

Uniform Prior The joint prior density is given by

$$p(\boldsymbol{\beta}) \propto 1$$

Posterior Distribution

Denote the observed data by D.

The posterior distribution is

$$\pi(\boldsymbol{\theta}|D) \propto L_P(D|\boldsymbol{\theta})p(\boldsymbol{\theta})$$

where $L_P(D|\boldsymbol{\theta})$ is the likelihood function with regression coefficients and any additional parameters, such as scale or shape, $\boldsymbol{\theta}$ as parameters; and $p(\boldsymbol{\theta})$ is the joint prior distribution of the parameters.

Deviance Information Criterion

Let $\boldsymbol{\theta}_i$ be the model parameters at iteration i of the Gibbs sampler, and let $\mathrm{LL}(\boldsymbol{\theta}_i)$ be the corresponding model log likelihood. PROC LIFEREG computes the following fit statistics defined by Spiegelhalter et al. (2002):

- effective number of parameters:

$$p_D = \overline{\mathrm{LL}(\boldsymbol{\theta})} - \mathrm{LL}(\bar{\boldsymbol{\theta}})$$

- deviance information criterion (DIC):

$$\mathrm{DIC} = \overline{\mathrm{LL}(\boldsymbol{\theta})} + p_D$$

where

$$\overline{\mathrm{LL}(\boldsymbol{\theta})} = \frac{1}{n}\sum_{i=1}^{n}\mathrm{LL}(\boldsymbol{\theta}_i)$$

$$\bar{\boldsymbol{\theta}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{\theta}_i$$

and n is the number of Gibbs samples.

Starting Values of the Markov Chains

When the BAYES statement is specified, PROC LIFEREG generates one Markov chain containing the approximate posterior samples of the model parameters. Additional chains are produced when the Gelman-Rubin diagnostics are requested. Starting values (or initial values) can be specified in the INITIAL= data set in the BAYES statement. If INITIAL= option is not specified, PROC LIFEREG picks its own initial values for the chains.

Denote $[x]$ as the integral value of x. Denote $\hat{s}(X)$ as the estimated standard error of the estimator X.

Regression Coefficients and Gamma Shape Parameter

For the first chain that the summary statistics and regression diagnostics are based on, the default initial values are estimates of the mode of the posterior distribution. If the INITIALMLE option is specified, the initial values are the maximum likelihood estimates; that is,

$$\beta_i^{(0)} = \hat{\beta}_i$$

Initial values for the rth chain ($r \geq 2$) are given by

$$\beta_i^{(0)} = \hat{\beta}_i \pm \left(2 + \left[\frac{r}{2}\right]\right)\hat{s}(\hat{\beta}_i)$$

with the plus sign for odd r and minus sign for even r.

Scale, Exponential Scale, Weibull Scale, or Weibull Shape Parameter λ

Let λ be the parameter sampled.

For the first chain that the summary statistics and diagnostics are based on, the initial values are estimates of the mode of the posterior distribution; or the maximum likelihood estimates if the INITIALMLE option is specified; that is,

$$\lambda^{(0)} = \hat{\lambda}$$

The initial values of the rth chain ($r \geq 2$) are given by

$$\lambda^{(0)} = \hat{\lambda}\mathrm{e}^{\pm\left([\frac{r}{2}]+2\right)\hat{s}(\hat{\lambda})}$$

with the plus sign for odd r and minus sign for even r.

OUTPOST= Output Data Set

The OUTPOST= data set contains the generated posterior samples. There are 2+n variables, where n is the number of model parameters. The variable Iteration represents the iteration number and the variable LogPost contains the log posterior likelihood values. The other n variables represent the draws of the Markov chain for the model parameters.

Displayed Output for Classical Analysis

For each model, PROC LIFEREG displays the following.

Model Information

The "Model Information" table displays the two-level name of the input data set, the distribution name, and the name and label of the dependent variable; the name and label of the censor indicator variable, for right-censored data; if you specify the WEIGHT statement, the name and label of the weight variable; and the maximum value of the log likelihood.

Number of Observations

The "Number of Observations" table displays the number of observations read from the input data set, and the number of observations used in the analysis.

Class Level Information

The "Class Level Information" table displays the levels of classification variables if you specify a CLASS statement.

Fit Statistics

The "Fit Statistics" table displays the negative of twice the log likelihood, the Akaike information criterion (AIC), the corrected Akaike information criterion (AICC), and the Bayesian information criterion (BIC).

Type III Analysis of Effects

The "Type III Analysis of Effects" table displays, for each effect in the model, the effect name, the degrees of freedom associated with the type III contrast for the effect, the chi-square statistic for the contrast, and the *p*-value for the statistic.

Analysis of Maximum Likelihood Parameter Estimates

The "Analysis of Maximum Likelihood Parameter Estimates" table displays the parameter name, the degrees of freedom for each parameter, the maximum likelihood estimate of each parameter, the estimated standard error of the parameter estimator, confidence limits for each parameter, a chi-square statistic for testing whether the parameter is zero, and the associated *p*-value for the statistic.

Lagrange Multiplier Statistics

If there are constrained parameters in the model, such as the scale or intercept, then the "Lagrange Multiplier Statistics" table displays a Lagrange multiplier test for the constraint.

Displayed Output for Bayesian Analysis

If a Bayesian analysis is requested with a BAYES statement, the displayed output includes the following.

Model Information

The "Model Information" table displays the two-level name of the input data set, the number of burn-in iterations, the number of iterations after the burn-in, the number of thinning iterations, the distribution name, and the name and label of the dependent variable; the name and label of the censor indicator variable, for right-censored data; if you specify the WEIGHT statement, the name and label of the weight variable; and the maximum value of the log likelihood.

Class Level Information

The "Class Level Information" table displays the levels of classification variables if you specify a CLASS statement.

Maximum Likelihood Estimates

The "Analysis of Maximum Likelihood Parameter Estimates" table displays the maximum likelihood estimate of each parameter, the estimated standard error of the parameter estimator, and confidence limits for each parameter.

Coefficient Prior

The "Coefficient Prior" table displays the prior distribution of the regression coefficients.

Independent Prior Distributions for Model Parameters

The "Independent Prior Distributions for Model Parameters" table displays the prior distributions of additional model parameters (scale, exponential scale, Weibull scale, Weibull shape, gamma shape).

Initial Values and Seeds

The "Initial Values and Seeds" table displays the initial values and random number generator seeds for the Gibbs chains.

Fit Statistics

The "Fit Statistics" table displays the deviance information criterion (DIC) and the effective number of parameters.

Posterior Summaries

The "Posterior Summaries" table contains the size of the sample, the mean, the standard deviation, and the quartiles for each model parameter.

Posterior Intervals

The "Posterior Intervals" table contains the HPD intervals and the credible intervals for each model parameter.

Correlation Matrix of the Posterior Samples

The "Correlation Matrix of the Posterior Samples" table is produced if you include the CORR suboption in the SUMMARY= option in the BAYES statement. This table displays the sample correlation of the posterior samples.

Covariance Matrix of the Posterior Samples

The "Covariance Matrix of the Posterior Samples" table is produced if you include the COV suboption in the SUMMARY= option in the BAYES statement. This table displays the sample covariance of the posterior samples.

Autocorrelations of the Posterior Samples

The "Autocorrelations of the Posterior Samples" table displays the lag1, lag5, lag10, and lag50 autocorrelations for each parameter.

Gelman and Rubin Diagnostics

The "Gelman and Rubin Diagnostics" table is produced if you include the GELMAN suboption in the DIAGNOSTIC= option in the BAYES statement. This table displays the estimate of the potential scale reduction factor and its 97.5% upper confidence limit for each parameter.

Geweke Diagnostics

The "Geweke Diagnostics" table displays the Geweke statistic and its p-value for each parameter.

Raftery and Lewis Diagnostics

The "Raftery Diagnostics" tables is produced if you include the RAFTERY suboption in the DIAGNOSTIC= option in the BAYES statement. This table displays the Raftery and Lewis diagnostics for each variable.

Heidelberger and Welch Diagnostics

The "Heidelberger and Welch Diagnostics" table is displayed if you include the HEIDELBERGER suboption in the DIAGNOSTIC= option in the BAYES statement. This table shows the results of a stationary test and a halfwidth test for each parameter.

Effective Sample Size

The "Effective Sample Size" table displays, for each parameter, the effective sample size, the correlation time, and the efficiency.

Monte Carlo Standard Errors

The "Monte Carlo Standard Errors" table displays, for each parameter, the Monte Carlo standard error, the posterior sample standard deviation, and the ratio of the two.

ODS Table Names

PROC LIFEREG assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed separately in Table 48.4 for a maximum likelihood analysis and in Table 48.5 for a Bayesian analysis. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 48.4 ODS Tables Produced in PROC LIFEREG for a Classical Analysis

ODS Table Name	Description	Statement	Option
ClassLevels	Classification variable levels	CLASS	default*
ConvergenceStatus	Convergence status	MODEL	default
CorrB	Parameter estimate correlation matrix	MODEL	CORRB
CovB	Parameter estimate covariance matrix	MODEL	COVB
IterEM	Iteration history for Turnbull algorithm	PROBPLOT	ITPRINTEM
FitStatistics	Fit statistics	MODEL	default
IterHistory	Iteration history	MODEL	ITPRINT
LagrangeStatistics	Lagrange statistics	MODEL	NOINT | NOSCALE
LastGrad	Last evaluation of the gradient	MODEL	ITPRINT
LastHess	Last evaluation of the Hessian	MODEL	ITPRINT

Table 48.4 *continued*

ODS Table Name	Description	Statement	Option
ModelInfo	Model information	MODEL	default
NObs	Number of observations	MODEL	default
ParameterEstimates	Parameter estimates	MODEL	default
ParmInfo	Parameter indices	MODEL	default
ProbabilityEstimates	Nonparametric CDF estimates	PROBPLOT	PPOUT
TConvergenceStatus	Convergence status for Turnbull algorithm	PROBPLOT	default
Turnbull	Probability estimates from Turnbull algorithm	PROBPLOT	ITPRINTEM
Type3Analysis	Type 3 tests	MODEL	default*

* Depends on data.

Table 48.5 ODS Tables Produced in PROC LIFEREG for a Bayesian Analysis

ODS Table Name	Description	Statement	Option
AutoCorr	Autocorrelations of the posterior samples	BAYES	default
ClassLevels	Classification variable levels	CLASS	default*
CoeffPrior	Prior distribution of the regression coefficients	BAYES	default
ConvergenceStatus	Convergence status of maximum likelihood estimation	MODEL	default
Corr	Correlation matrix of the posterior samples	BAYES	SUMMARY=CORR
ESS	Effective sample size	BAYES	default
FitStatistics	Fit statistics	BAYES	default
Gelman	Gelman and Rubin convergence diagnostics	BAYES	DIAG=GELMAN
Geweke	Geweke convergence diagnostics	BAYES	default
Heidelberger	Heidelberger and Welch convergence diagnostics	BAYES	DIAG=HEIDELBERGER
InitialValues	Initial values of the Markov chains	BAYES	default
ModelInfo	Model information	MODEL	default
NObs	Number of observations	MODEL	default
ParameterEstimates	Maximum likelihood estimates of model parameters	MODEL	default
ParmPrior	Prior distribution for scale and shape	BAYES	default
PostIntervals	HPD and equal-tail intervals of the posterior samples	BAYES	default
PosteriorSample	Posterior samples (for output data set only)	BAYES	
PostSummaries	Summary statistics of the posterior samples	BAYES	default
Raftery	Raftery and Lewis convergence diagnostics	BAYES	DIAG=RAFTERY

* Depends on data.

ODS Graphics

To request graphics with PROC LIFEREG, you must first enable ODS Graphics by specifying the `ods graphics on;` statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. Some graphs are produced by default; other graphs are produced by using statements and options. You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC LIFEREG generates are listed in Table 48.6, along with the required statements and options.

ODS Graph Names

PROC LIFEREG assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 48.6.

To request these graphs, you must specify the `ods graphics on;` statement in addition to the options indicated in Table 48.6.

Table 48.6 ODS Graphics Produced by PROC LIFEREG

ODS Graph Name	Description	Statement	Option
ADPanel	Autocorrelation function and density panel	BAYES	PLOTS=(AUTOCORR DENSITY)
AutocorrPanel	Autocorrelation function panel	BAYES	PLOTS= AUTOCORR
AutocorrPlot	Autocorrelation function plot	BAYES	PLOTS(UNPACK)=AUTOCORR
ProbPlot	Probability plot	PROBPLOT	default
TAPanel	Trace and autocorrelation function panel	BAYES	PLOTS=(TRACE AUTOCORR)
TADPanel	Trace, autocorrelation, and density function panel	BAYES	default
TDPanel	Trace and density panel	BAYES	PLOTS=(TRACE DENSITY)
TracePanel	Trace panel	BAYES	PLOTS=TRACE
TracePlot	Trace plot	BAYES	PLOTS(UNPACK)=TRACE

Examples: LIFEREG Procedure

Example 48.1: Motorette Failure

This example fits a Weibull model and a lognormal model to the example given in Kalbfleisch and Prentice (1980, p. 5). An output data set called models is specified to contain the parameter estimates. By default, the natural log of the variable time is used by the procedure as the response. After this log transformation, the Weibull model is fit using the extreme-value baseline distribution, and the lognormal is fit using the normal baseline distribution.

Since the extreme-value and normal distributions do not contain any shape parameters, the variable SHAPE1 is missing in the models data set. An additional output data set, out, is created that contains the predicted quantiles and their standard errors for values of the covariate corresponding to temp=130 and temp=150. This is done with the control variable, which is set to 1 for only two observations.

Using the standard error estimates obtained from the output data set, approximate 90% confidence limits for the predicted quantities are then created in a subsequent DATA step for the log response. The logs of the predicted values are obtained because the values of the P= variable in the OUT= data set are in the same units as the original response variable, time. The standard errors of the quantiles of log(time) are approximated (using a Taylor series approximation) by the standard deviation of time divided by the mean value of time. These confidence limits are then converted back to the original scale by the exponential function.

The following statements produce Output 48.1.1:

```
title 'Motorette Failures With Operating Temperature as a Covariate';
data motors;
   input time censor temp @@;
   if _N_=1 then
      do;
         temp=130;
         time=.;
         control=1;
         z=1000/(273.2+temp);
         output;
         temp=150;
         time=.;
         control=1;
         z=1000/(273.2+temp);
         output;
      end;
   if temp>150;
   control=0;
   z=1000/(273.2+temp);
   output;
   datalines;
8064 0 150 8064 0 150 8064 0 150 8064 0 150 8064 0 150
8064 0 150 8064 0 150 8064 0 150 8064 0 150 8064 0 150
1764 1 170 2772 1 170 3444 1 170 3542 1 170 3780 1 170
4860 1 170 5196 1 170 5448 0 170 5448 0 170 5448 0 170
 408 1 190  408 1 190 1344 1 190 1344 1 190 1440 1 190
1680 0 190 1680 0 190 1680 0 190 1680 0 190 1680 0 190
 408 1 220  408 1 220  504 1 220  504 1 220  504 1 220
 528 0 220  528 0 220  528 0 220  528 0 220  528 0 220
;
run;
proc print data=motors;
run;
```

Output 48.1.1 Motorette Failure Data

Motorette Failures With Operating Temperature as a Covariate

Obs	time	censor	temp	control	z
1	.	0	130	1	2.48016
2	.	0	150	1	2.36295
3	1764	1	170	0	2.25632
4	2772	1	170	0	2.25632
5	3444	1	170	0	2.25632
6	3542	1	170	0	2.25632
7	3780	1	170	0	2.25632
8	4860	1	170	0	2.25632
9	5196	1	170	0	2.25632
10	5448	0	170	0	2.25632
11	5448	0	170	0	2.25632
12	5448	0	170	0	2.25632
13	408	1	190	0	2.15889
14	408	1	190	0	2.15889
15	1344	1	190	0	2.15889
16	1344	1	190	0	2.15889
17	1440	1	190	0	2.15889
18	1680	0	190	0	2.15889
19	1680	0	190	0	2.15889
20	1680	0	190	0	2.15889
21	1680	0	190	0	2.15889
22	1680	0	190	0	2.15889
23	408	1	220	0	2.02758
24	408	1	220	0	2.02758
25	504	1	220	0	2.02758
26	504	1	220	0	2.02758
27	504	1	220	0	2.02758
28	528	0	220	0	2.02758
29	528	0	220	0	2.02758
30	528	0	220	0	2.02758
31	528	0	220	0	2.02758
32	528	0	220	0	2.02758

The following statements produce Output 48.1.2 and Output 48.1.3:

```
proc lifereg data=motors outest=modela covout;
   a: model time*censor(0)=z;
      output out=outa quantiles=.1 .5 .9 std=std p=predtime
         control=control;
run;
proc lifereg data=motors outest=modelb covout;
   b: model time*censor(0)=z / dist=lnormal;
      output out=outb quantiles=.1 .5 .9 std=std p=predtime
         control=control;
run;
```

Output 48.1.2 Motorette Failure: Model A

Motorette Failures With Operating Temperature as a Covariate

The LIFEREG Procedure

Model Information

Data Set	WORK.MOTORS
Dependent Variable	Log(time)
Censoring Variable	censor
Censoring Value(s)	0
Number of Observations	30
Noncensored Values	17
Right Censored Values	13
Left Censored Values	0
Interval Censored Values	0
Name of Distribution	Weibull
Log Likelihood	-22.95148315

Type III Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
z	1	99.5239	<.0001

Analysis of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-11.8912	1.9655	-15.7435	-8.0389	36.60	<.0001
z	1	9.0383	0.9060	7.2626	10.8141	99.52	<.0001
Scale	1	0.3613	0.0795	0.2347	0.5561		
Weibull Shape	1	2.7679	0.6091	1.7982	4.2605		

Output 48.1.3 Motorette Failure: Model B

Motorette Failures With Operating Temperature as a Covariate

The LIFEREG Procedure

Model Information

Data Set	WORK.MOTORS
Dependent Variable	Log(time)
Censoring Variable	censor
Censoring Value(s)	0
Number of Observations	30
Noncensored Values	17
Right Censored Values	13
Left Censored Values	0
Interval Censored Values	0
Name of Distribution	Lognormal
Log Likelihood	-24.47381031

Type III Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
z	1	42.0001	<.0001

Analysis of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-10.4706	2.7719	-15.9034	-5.0377	14.27	0.0002
z	1	8.3221	1.2841	5.8052	10.8389	42.00	<.0001
Scale	1	0.6040	0.1107	0.4217	0.8652		

The following statements produce Output 48.1.4:

```
data models;
   set modela modelb;
run;
proc print data=models;
   id _model_;
   title 'fitted models';
run;
```

Output 48.1.4 Motorette Failure: Fitted Models

fitted models

MODEL	_NAME_	_TYPE_	_DIST_	_STATUS_	_LNLIKE_
a	time	PARMS	Weibull	0 Converged	-22.9515
a	Intercept	COV	Weibull	0 Converged	-22.9515
a	z	COV	Weibull	0 Converged	-22.9515
a	Scale	COV	Weibull	0 Converged	-22.9515
b	time	PARMS	Lognormal	0 Converged	-24.4738
b	Intercept	COV	Lognormal	0 Converged	-24.4738
b	z	COV	Lognormal	0 Converged	-24.4738
b	Scale	COV	Lognormal	0 Converged	-24.4738

MODEL	time	Intercept	z	_SCALE_
a	-1.0000	-11.8912	9.03834	0.36128
a	-11.8912	3.8632	-1.77878	0.03448
a	9.0383	-1.7788	0.82082	-0.01488
a	0.3613	0.0345	-0.01488	0.00632
b	-1.0000	-10.4706	8.32208	0.60403
b	-10.4706	7.6835	-3.55566	0.03267
b	8.3221	-3.5557	1.64897	-0.01285
b	0.6040	0.0327	-0.01285	0.01226

The following statements produce Output 48.1.5:

```
data out;
   set outa outb;
run;

data out1;
   set out;
   ltime=log(predtime);
   stde=std/predtime;
   upper=exp(ltime+1.64*stde);
   lower=exp(ltime-1.64*stde);
run;
title 'quantile estimates and confidence limits';
proc print data=out1;
   id temp;
run;
title;
```

Output 48.1.5 Motorette Failure: Quantile Estimates and Confidence Limits

quantile estimates and confidence limits

temp	tmper	censor	lotol	z	_PROB_	pred time	std	lmean	lstd	upper	lower
130	.	0	1	2.48016	0.1	16519.27	5999.85	9.7123	0.36320	29969.51	9105.47
130	.	0	1	2.48016	0.5	32626.65	9874.33	10.3929	0.30265	53595.71	19861.63
130	.	0	1	2.48016	0.9	50343.22	15044.35	10.8266	0.29884	82183.49	30838.80
150	.	0	1	2.36295	0.1	5726.74	1569.34	8.6529	0.27404	8976.12	3653.64
150	.	0	1	2.36295	0.5	11310.68	2299.92	9.3335	0.20334	15787.62	8103.28
150	.	0	1	2.36295	0.9	17452.49	3629.28	9.7672	0.20795	24545.37	12409.24
130	.	0	1	2.48016	0.1	12033.19	5482.34	9.3954	0.45560	25402.68	5700.09
130	.	0	1	2.48016	0.5	26095.68	11359.45	10.1695	0.43530	53285.36	12779.95
130	.	0	1	2.48016	0.9	56592.19	26036.90	10.9436	0.46008	120349.65	26611.42
150	.	0	1	2.36295	0.1	4536.88	1443.07	8.4200	0.31808	7643.71	2692.83
150	.	0	1	2.36295	0.5	9838.86	2901.15	9.1941	0.29487	15957.38	6066.36
150	.	0	1	2.36295	0.9	21336.97	7172.34	9.9682	0.33615	37029.72	12294.62

Example 48.2: Computing Predicted Values for a Tobit Model

The LIFEREG procedure can be used to perform a Tobit analysis. The Tobit model, described by Tobin (1958), is a regression model for left-censored data assuming a normally distributed error term. The model parameters are estimated by maximum likelihood. PROC LIFEREG provides estimates of the parameters of the distribution of the *uncensored* data. See Greene (1993) and Maddala (1983) for a more complete discussion of censored normal data and related distributions. This example shows how you can use PROC LIFEREG and the DATA step to compute two of the three types of predicted values discussed there.

Consider a continuous random variable Y and a constant C. If you were to sample from the distribution of Y but discard values less than (greater than) C, the distribution of the remaining observations would be *truncated* on the left (right). If you were to sample from the distribution of Y and report values less than (greater than) C as C, the distribution of the sample would be left (right) *censored*.

The probability density function of the truncated random variable Y' is given by

$$f_{\mathrm{Y}'}(y) = \frac{f_{\mathrm{Y}}(y)}{\Pr(\mathrm{Y} > \mathrm{C})} \quad \text{for} \quad y > \mathrm{C}$$

where $f_{\mathrm{Y}}(y)$ is the probability density function of Y. PROC LIFEREG cannot compute the proper likelihood function to estimate parameters or predicted values for a truncated distribution. Suppose the model being fit is specified as follows:

$$\mathrm{Y}_i^* = \mathbf{x}_i'\boldsymbol{\beta} + \epsilon_i$$

where ϵ_i is a normal error term with zero mean and standard deviation σ.

Define the censored random variable Y_i as

$$\begin{aligned} Y_i &= 0 \text{ if } Y_i^* \leq 0 \\ Y_i &= Y_i^* \text{ if } Y_i^* > 0 \end{aligned}$$

This is the Tobit model for left-censored normal data. Y_i^* is sometimes called the *latent variable*. PROC LIFEREG estimates parameters of the distribution of Y_i^* by maximum likelihood.

You can use the LIFEREG procedure to compute predicted values based on the mean functions of the latent and observed variables. The mean of the latent variable Y_i^* is $\mathbf{x}_i'\boldsymbol{\beta}$, and you can compute values of the mean for different settings of $\mathbf{x}_i$ by specifying XBETA=*variable-name* in an OUTPUT statement. Estimates of $\mathbf{x}_i'\boldsymbol{\beta}$ for each observation will be written to the OUT= data set. Predicted values of the observed variable Y_i can be computed based on the mean

$$E(Y_i) = \Phi\left(\frac{\mathbf{x}_i'\boldsymbol{\beta}}{\sigma}\right)(\mathbf{x}_i'\boldsymbol{\beta} + \sigma\lambda_i)$$

where

$$\lambda_i = \frac{\phi(\mathbf{x}_i'\boldsymbol{\beta}/\sigma)}{\Phi(\mathbf{x}_i'\boldsymbol{\beta}/\sigma)}$$

ϕ and Φ represent the normal probability density and cumulative distribution functions.

Although the distribution of ϵ_i in the Tobit model is often assumed normal, you can use other distributions for the Tobit model in the LIFEREG procedure by specifying a distribution with the DISTRIBUTION= option in the MODEL statement. One distribution that should be mentioned is the logistic distribution. For this distribution, the MLE has bounded influence function with respect to the response variable, but not the design variables. If you believe your data have outliers in the response direction, you might try this distribution for some robust estimation of the Tobit model.

With the logistic distribution, the predicted values of the observed variable Y_i can be computed based on the mean of Y_i^*,

$$E(Y_i) = \sigma \ln(1 + \exp(\mathbf{x}_i'\boldsymbol{\beta}/\sigma))$$

The following table shows a subset of the Mroz (1987) data set. In these data, Hours is the number of hours the wife worked outside the household in a given year, Yrs_Ed is the years of education, and Yrs_Exp is the years of work experience. A Tobit model will be fit to the hours worked with years of education and experience as covariates.

Hours	Yrs_Ed	Yrs_Exp
0	8	9
0	8	12
0	9	10
0	10	15
0	11	4
0	11	6
1000	12	1
1960	12	29
0	13	3
2100	13	36
3686	14	11
1920	14	38
0	15	14
1728	16	3
1568	16	19
1316	17	7
0	17	15

If the wife was not employed (worked 0 hours), her hours worked will be left censored at zero. In order to accommodate left censoring in PROC LIFEREG, you need two variables to indicate censoring status of observations. You can think of these variables as lower and upper endpoints of interval censoring. If there is no censoring, set both variables to the observed value of Hours. To indicate left censoring, set the lower endpoint to missing and the upper endpoint to the censored value, zero in this case.

The following statements create a SAS data set with the variables Hours, Yrs_Ed, and Yrs_Exp from the preceding data. A new variable, Lower, is created such that Lower=. if Hours=0 and Lower=Hours if Hours>0.

```
data subset;
   input Hours Yrs_Ed Yrs_Exp @@;
   if Hours eq 0
      then Lower=.;
      else Lower=Hours;
datalines;
0 8 9 0 8 12 0 9 10 0 10 15 0 11 4 0 11 6
1000 12 1 1960 12 29 0 13 3 2100 13 36
3686 14 11 1920 14 38 0 15 14 1728 16 3
1568 16 19 1316 17 7 0 17 15
;
run;
```

The following statements fit a normal regression model to the left-censored Hours data with Yrs_Ed and Yrs_Exp as covariates. You need the estimated standard deviation of the normal distribution to compute the predicted values of the censored distribution from the preceding formulas. The data set OUTEST contains the standard deviation estimate in a variable named _SCALE_. You also need estimates of $\mathbf{x}_i'\boldsymbol{\beta}$. These are contained in the data set OUT as the variable Xbeta.

```
proc lifereg data=subset outest=OUTEST(keep=_scale_);
   model (lower, hours) = yrs_ed yrs_exp / d=normal;
   output out=OUT xbeta=Xbeta;
run;
```

Output 48.2.1 shows the results of the model fit. These tables show parameter estimates for the uncensored, or latent variable, distribution.

Output 48.2.1 Parameter Estimates from PROC LIFEREG

```
                         The LIFEREG Procedure

                           Model Information

                  Data Set                     WORK.SUBSET
                  Dependent Variable                 Lower
                  Dependent Variable                 Hours
                  Number of Observations                17
                  Noncensored Values                     8
                  Right Censored Values                  0
                  Left Censored Values                   9
                  Interval Censored Values               0
                  Name of Distribution              Normal
                  Log Likelihood               -74.9369977

            Analysis of Maximum Likelihood Parameter Estimates

                          Standard   95% Confidence     Chi-
     Parameter DF Estimate    Error       Limits       Square Pr > ChiSq

     Intercept  1 -5598.64 2850.248 -11185.0 -12.2553    3.86     0.0495
     Yrs_Ed     1 373.1477 191.8872  -2.9442 749.2397    3.78     0.0518
     Yrs_Exp    1  63.3371  38.3632 -11.8533 138.5276    2.73     0.0987
     Scale      1 1582.870 442.6732 914.9433 2738.397
```

The following statements combine the two data sets created by PROC LIFEREG to compute predicted values for the censored distribution. The OUTEST= data set contains the estimate of the standard deviation from the uncensored distribution, and the OUT= data set contains estimates of $\mathbf{x}_i'\boldsymbol{\beta}$.

```
data predict;
   drop lambda _scale_ _prob_;
   set out;
   if _n_ eq 1 then set outest;
   lambda = pdf('NORMAL',Xbeta/_scale_)
            / cdf('NORMAL',Xbeta/_scale_);
   Predict = cdf('NORMAL', Xbeta/_scale_)
             * (Xbeta + _scale_*lambda);
   label Xbeta='MEAN OF UNCENSORED VARIABLE'
         Predict = 'MEAN OF CENSORED VARIABLE';
run;
```

Output 48.2.2 shows the original variables, the predicted means of the uncensored distribution, and the predicted means of the censored distribution.

Output 48.2.2 Predicted Means from PROC LIFEREG

Hours	Lower	Yrs_Ed	Yrs_Exp	MEAN OF UNCENSORED VARIABLE	MEAN OF CENSORED VARIABLE
0	.	8	9	-2043.42	73.46
0	.	8	12	-1853.41	94.23
0	.	9	10	-1606.94	128.10
0	.	10	15	-917.10	276.04
0	.	11	4	-1240.67	195.76
0	.	11	6	-1113.99	224.72
1000	1000	12	1	-1057.53	238.63
1960	1960	12	29	715.91	1052.94
0	.	13	3	-557.71	391.42
2100	2100	13	36	1532.42	1672.50
3686	3686	14	11	322.14	805.58
1920	1920	14	38	2032.24	2106.81
0	.	15	14	885.30	1170.39
1728	1728	16	3	561.74	951.69
1568	1568	16	19	1575.13	1708.24
1316	1316	17	7	1188.23	1395.61
0	.	17	15	1694.93	1809.97

Example 48.3: Overcoming Convergence Problems by Specifying Initial Values

This example illustrates the use of parameter initial value specification to help overcome convergence difficulties.

The following statements create a SAS data set.

```
data raw;
   input censor x c1 @@;
   datalines;
0 16 0.00   0 17 0.00   0 18 0.00
0 17 0.04   0 18 0.04   0 18 0.04
0 23 0.40   0 22 0.40   0 22 0.40
0 33 4.00   0 34 4.00   0 35 4.00
1 54 40.00  1 54 40.00  1 54 40.00
1 54 400.00 1 54 400.00 1 54 400.00
;
run;
```

Output 48.3.1 shows the contents of the data set raw.

Output 48.3.1 Contents of the Data Set

Obs	censor	x	c1
1	0	16	0.00
2	0	17	0.00
3	0	18	0.00
4	0	17	0.04
5	0	18	0.04
6	0	18	0.04
7	0	23	0.40
8	0	22	0.40
9	0	22	0.40
10	0	33	4.00
11	0	34	4.00
12	0	35	4.00
13	1	54	40.00
14	1	54	40.00
15	1	54	40.00
16	1	54	400.00
17	1	54	400.00
18	1	54	400.00

The following SAS statements request that a Weibull regression model be fit to the data:

```
title 'OLS (default) initial values';
proc lifereg data=raw;
   model x*censor(1) = c1 / distribution = Weibull itprint;
run;
```

Convergence was not attained in 50 iterations for this model, as the following messages to the log indicate:

```
WARNING: Convergence was not attained in 50 iterations. You might want to
         increase the maximum number of iterations (MAXITER= option) or
         change the convergence criteria (CONVERGE = value) in the MODEL
         statement.
WARNING: The procedure is continuing in spite of the above warning. Results
         shown are based on the last maximum likelihood iteration. Validity
         of the model fit is questionable.
```

The first line (iter=0) of the iteration history table, shown in Output 48.3.2, shows the default initial ordinary least squares (OLS) estimates of the parameters.

Output 48.3.2 Initial Least Squares

OLS (default) initial values

The LIFEREG Procedure

Iteration History for Parameter Estimates

Iter	Ridge	Loglikelihood	Intercept	c1	Scale
0	0	−22.891088	3.2324769714	0.0020664542	0.3995754195
1	0	−16.427074	3.5337141598	0.0028713635	0.3283544365
2	0	−13.216768	3.4480787541	0.0052801225	0.3816964358
3	0	−5.0786635	3.1966395335	0.0191439929	0.2325418958
4	0	−2.0018885	3.1848047525	0.0275425402	0.1963590539
5	0	−0.1814984	3.1478989655	0.0374731819	0.2103607621
6	0	2.90712131	3.0858183316	0.0659946149	0.1818245261
7	0.063	2.9991781	3.1014479187	0.0661096622	0.1648677081
8	0.063	3.01557837	3.0995493638	0.0662333056	0.1670552505
9	0.063	3.0301815	3.0992317977	0.0663580659	0.1669529486
10	0.063	3.0448013	3.0989901232	0.0664827053	0.1667371524
11	0.063	3.05941254	3.0987507448	0.0666071514	0.1665197313
12	0.063	3.07401474	3.0985118143	0.0667314052	0.1663026517
13	0.063	3.08860788	3.0982732928	0.066855467	0.1660859472
14	0.063	3.10319193	3.0980351787	0.0669793371	0.1658696184
15	0.063	3.11776689	3.0977974713	0.0671030156	0.1656536651
16	0.063	3.13233272	3.0975601698	0.0672265029	0.1654380873
17	0.063	3.1468894	3.0973232737	0.0673497993	0.165222885
18	0.063	3.16143692	3.0970867821	0.0674729049	0.1650080579
19	0.063	3.17597526	3.0968506943	0.06759582	0.1647936061
20	0.063	3.19050439	3.0966150098	0.0677185449	0.1645795293
21	0.063	3.2050243	3.0963797277	0.0678410799	0.1643658275
22	0.063	3.21953496	3.0961448474	0.0679634252	0.1641525006
23	0.063	3.23403635	3.0959103682	0.068085581	0.1639395483
24	0.063	3.24852845	3.0956762896	0.0682075476	0.1637269705
25	0.063	3.26301123	3.0954426107	0.0683293253	0.1635147672
26	0.063	3.27748468	3.095209331	0.0684509143	0.163302938
27	0.063	3.29194878	3.0949764498	0.0685723149	0.1630914829
28	0.063	3.3064035	3.0947439665	0.0686935273	0.1628804017
29	0.063	3.32084881	3.0945118805	0.0688145517	0.1626696942
30	0.063	3.3352847	3.0942801911	0.0689353885	0.1624593601
31	0.063	3.34971114	3.0940488977	0.0690560378	0.1622493994
32	0.063	3.36412812	3.0938179997	0.0691765	0.1620398118
33	0.063	3.3785356	3.0935874965	0.0692967752	0.1618305971
34	0.063	3.39293356	3.0933573875	0.0694168637	0.161621755
35	0.063	3.40732199	3.093127672	0.0695367658	0.1614132855
36	0.063	3.42170085	3.0928983495	0.0696564816	0.1612051882
37	0.063	3.43607013	3.0926694194	0.0697760116	0.1609974629
38	0.063	3.45042979	3.0924408811	0.0698953558	0.1607901095
39	0.063	3.46477983	3.092212734	0.0700145146	0.1605831276
40	0.063	3.4791202	3.0919849776	0.0701334882	0.160376517
41	0.063	3.4934509	3.0917576112	0.0702522768	0.1601702775
42	0.063	3.50777188	3.0915306343	0.0703708808	0.1599644088
43	0.063	3.52208314	3.0913040464	0.0704893002	0.1597589108
44	0.063	3.53638465	3.0910778468	0.0706075354	0.159553783
45	0.063	3.55067637	3.0908520349	0.0707255867	0.1593490254
46	0.063	3.5649583	3.0906266104	0.0708434542	0.1591446376
47	0.063	3.57923039	3.0904015725	0.0709611382	0.1589406193
48	0.063	3.59349263	3.0901769207	0.0710786389	0.1587369703
49	0.063	3.607745	3.0899526546	0.0711959567	0.1585336903
50	0.063	3.62198746	3.0897287734	0.0713130916	0.1583307791

The log-logistic distribution is more robust to large values of the response than the Weibull distribution, so one approach to improving the convergence performance is to fit a log-logistic distribution, and if this converges, use the resulting parameter estimates as initial values in a subsequent fit of a model with the Weibull distribution.

The following statements fit a log-logistic distribution to the data:

```
proc lifereg data=raw;
   model x*censor(1) = c1 / distribution = llogistic;
run;
```

The algorithm converges, and the maximum likelihood estimates for the log-logistic distribution are shown in Output 48.3.3

Output 48.3.3 Estimates from the Log-Logistic Distribution

OLS (default) initial values

The LIFEREG Procedure

Analysis of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	2.8983	0.0318	2.8360	2.9606	8309.43	<.0001
c1	1	0.1592	0.0133	0.1332	0.1852	143.85	<.0001
Scale	1	0.0498	0.0122	0.0308	0.0804		

The following statements refit the Weibull model by using the maximum likelihood estimates from the log-logistic fit as initial values:

```
proc lifereg data=raw outest=outest;
   model x*censor(1) = c1 / itprint distribution = weibull
                            intercept=2.898 initial=0.16 scale=0.05;
   output out=out xbeta=xbeta;
run;
```

Examination of the resulting output in Output 48.3.4 shows that the convergence problem has been solved by specifying different initial values.

Output 48.3.4 Final Estimates from the Weibull Distribution

```
                        OLS (default) initial values

                           The LIFEREG Procedure

                             Model Information

                    Data Set                          WORK.RAW
                    Dependent Variable                  Log(x)
                    Censoring Variable                  censor
                    Censoring Value(s)                       1
                    Number of Observations                  18
                    Noncensored Values                      12
                    Right Censored Values                    6
                    Left Censored Values                     0
                    Interval Censored Values                 0
                    Name of Distribution               Weibull
                    Log Likelihood                11.232023272

  Algorithm converged.

              Analysis of Maximum Likelihood Parameter Estimates

                                 Standard   95% Confidence     Chi-
    Parameter      DF Estimate     Error       Limits        Square Pr > ChiSq

    Intercept       1   2.9699    0.0326    2.9059    3.0338 8278.86     <.0001
    c1              1   0.1435    0.0165    0.1111    0.1758   75.43     <.0001
    Scale           1   0.0844    0.0189    0.0544    0.1308
    Weibull Shape   1  11.8526    2.6514    7.6455   18.3749
```

As an example of an alternative way of specifying initial values, the following invocation of PROC LIFEREG, using the INEST= data set to provide starting values for the three parameters, is equivalent to the previous invocation:

```
data in;
   input  intercept c1 scale;
   datalines;
2.898 0.16 0.05
;
proc lifereg data=raw inest=in outest=outest;
   model x*censor(1) = c1 / itprint distribution = weibull;
   output out=out xbeta=xbeta;
run;
```

Example 48.4: Analysis of Arbitrarily Censored Data with Interaction Effects

The artificial data in this example are from a study of the natural recovery time of mice after injection of a certain toxin. Twenty mice were grouped by sex (sex: 1 = Male, 2 = Female) with equal sizes. Their ages (in days) were recorded at the injection. Their recovery times (in minutes) were also recorded. Toxin density in blood was used to decide whether a mouse recovered. Mice were checked at two times for recovery. If a mouse had recovered at the first time, the observation is left censored, and no further measurement is made. The variable time1 is set to missing and time2 is set to the measurement time to indicate left censoring. If a mouse had not recovered at the first time, it was checked later at a second time. If it had recovered by the second measurement time, the observation is interval censored, and the variable time1 is set to the first measurement time and time2 is set to the second measurement time. If there was no recovery at the second measurement, the observation is right censored, and time1 is set to the second measurement time and time2 is set to missing to indicate right censoring.

The following statements create a SAS data set containing the data from the experiment:

```
title 'Natural Recovery Time';
data mice;
   input sex age time1 time2 ;
   datalines;
1  57  631   631
1  45  .     170
1  54  227   227
1  43  143   143
1  64  916   .
1  67  691   705
1  44  100   100
1  59  730   .
1  47  365   365
1  74  1916  1916
2  79  1326  .
2  75  837   837
2  84  1200  1235
2  54  .     365
2  74  1255  1255
2  71  1823  .
2  65  537   637
2  33  583   683
2  77  955   .
2  46  577   577
;
```

The following SAS statements create the SAS data sets xrow1 and xrow2:

```
data xrow1;
   input sex age time1 time2 ;
   datalines;
1  50   .   .
;

data xrow2;
   input sex age time1 time2 ;
   datalines;
2  60.6   .   .
;
```

The following SAS statements fit a Weibull model with age, sex, and an age-by-sex interaction term as covariates, and create a plot of predicted probabilities against recovery time for the fixed values of age and sex specified in the SAS data set xrow1:

```
ods graphics on;
proc lifereg data=mice xdata=xrow1;
   class sex  ;
   model (time1, time2) = age sex age*sex / dist=Weibull;

   probplot  / nodata
       plower=.5
       vref(intersect) = 75
       vreflab = '75 Percent'
       ;
   inset;
run;
```

Standard output is shown in Output 48.4.1. Tables containing general model information, Type III tests for the main effects and interaction terms, and parameter estimates are created.

Output 48.4.1 Parameter Estimates for the Interaction Model

Natural Recovery Time

The LIFEREG Procedure

Model Information

Data Set	WORK.MICE
Dependent Variable	Log(time1)
Dependent Variable	Log(time2)
Number of Observations	20
Noncensored Values	9
Right Censored Values	5
Left Censored Values	2
Interval Censored Values	4
Name of Distribution	Weibull
Log Likelihood	-25.91033295

Output 48.4.1 *continued*

Type III Analysis of Effects

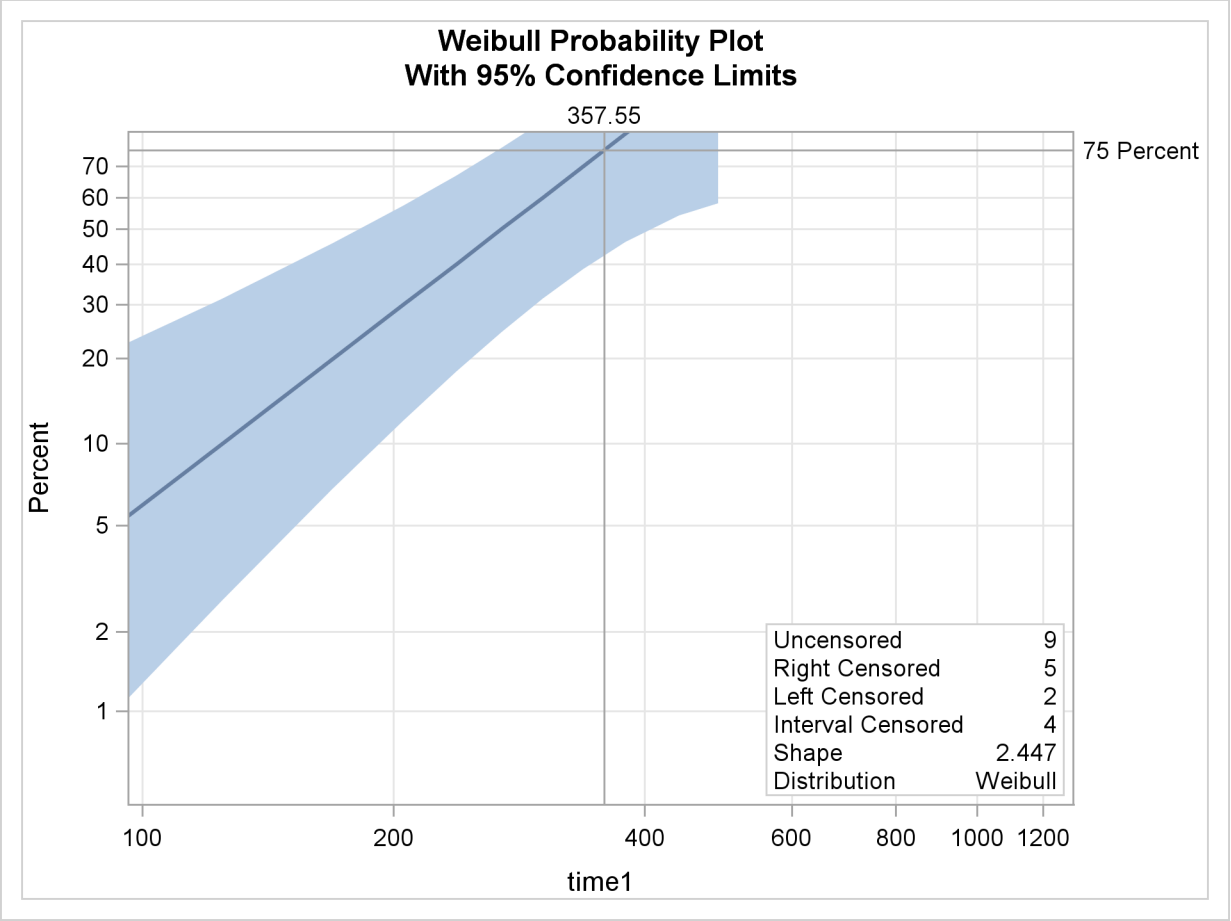
Effect	DF	Wald Chi-Square	Pr > ChiSq
age	1	33.8496	<.0001
sex	1	14.0245	0.0002
age*sex	1	10.7196	0.0011

Analysis of Maximum Likelihood Parameter Estimates

Parameter		DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept		1	5.4110	0.5549	4.3234	6.4986	95.08	<.0001
age		1	0.0250	0.0086	0.0081	0.0419	8.42	0.0037
sex	1	1	-3.9808	1.0630	-6.0643	-1.8974	14.02	0.0002
sex	2	0	0.0000	.	.	.	.	.
age*sex	1	1	0.0613	0.0187	0.0246	0.0980	10.72	0.0011
age*sex	2	0	0.0000	.	.	.	.	.
Scale		1	0.4087	0.0900	0.2654	0.6294		
Weibull Shape		1	2.4468	0.5391	1.5887	3.7682		

The following two plots display the predicted probability against the recovery time for two different populations. Output 48.4.2 is created with the PROBPLOT statement with the option XDATA= xrow1, which specifies the population with sex = 1, age = 50. Output 48.4.3 is created with the PROBPLOT statement with the option XDATA= xrow2, which specifies the population with sex = 2, age = 60.6. These are the default values that the LIFEREG procedure would use for the probability plot if the XDATA= option had not been specified. Reference lines are used to display specified predicted probability points and their relative locations in the plot.

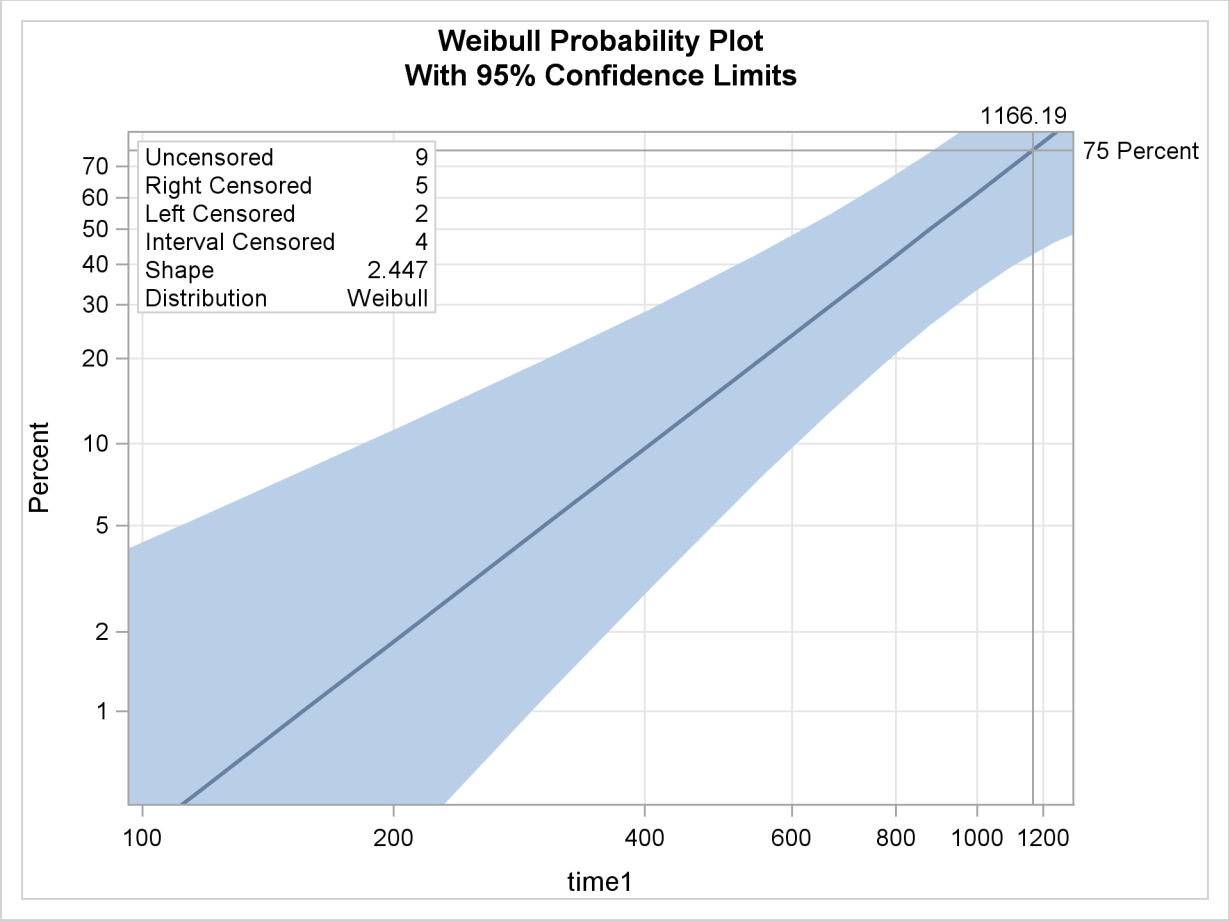
Output 48.4.2 Probability Plot for Recovery Time with sex = 1, age = 50



The following SAS statements fit a Weibull model with age, sex, and an age-by-sex interaction term as covariates, and create the plot of predicted probabilities against recovery time shown in Output 48.4.3, for the fixed values of age and sex specified in the SAS data set xrow2:

```
proc lifereg data=mice xdata=xrow2;
   class sex  ;
   model (time1, time2) = age sex age*sex / dist=Weibull;

   probplot  / nodata
       plower=.5
       vref(intersect) = 75
       vreflab = '75 Percent'
       ;
   inset;
run;
title;
ods graphics off;
```

Output 48.4.3 Probability Plot for Recovery Time with sex = 2, age = 60.6



Example 48.5: Probability Plotting—Right Censoring

The following statements create a SAS data set containing observed and right-censored lifetimes of 70 diesel engine fans (Nelson 1982):

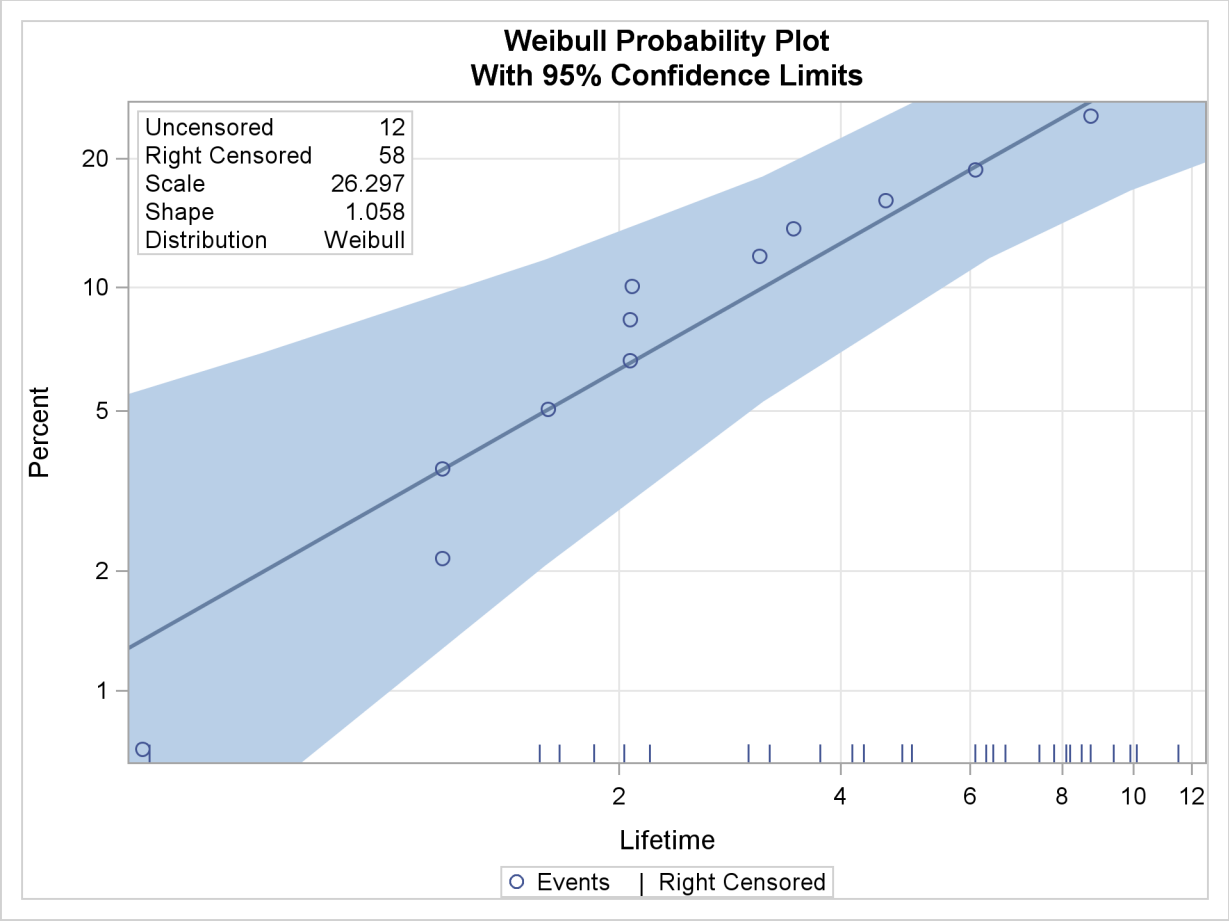
```
data Fan;
   input Lifetime Censor@@;
   Lifetime = Lifetime / 1000;
   datalines;
 450 0    460 1   1150 0   1150 0   1560 1
1600 0   1660 1   1850 1   1850 1   1850 1
1850 1   1850 1   2030 1   2030 1   2030 1
2070 0   2070 0   2080 0   2200 1   3000 1
3000 1   3000 1   3000 1   3100 0   3200 1
3450 0   3750 1   3750 1   4150 1   4150 1
4150 1   4150 1   4300 1   4300 1   4300 1
4300 1   4600 0   4850 1   4850 1   4850 1
4850 1   5000 1   5000 1   5000 1   6100 1
6100 0   6100 1   6100 1   6300 1   6450 1
6450 1   6700 1   7450 1   7800 1   7800 1
8100 1   8100 1   8200 1   8500 1   8500 1
8500 1   8750 1   8750 0   8750 1   9400 1
9900 1  10100 1  10100 1  10100 1  11500 1
;
run;
```

Some of the fans had not failed at the time the data were collected, and the unfailed units have right-censored lifetimes. The variable LIFETIME represents either a failure time or a censoring time, in thousands of hours. The variable CENSOR is equal to 0 if the value of LIFETIME is a failure time, and it is equal to 1 if the value is a censoring time. The following statements use the LIFEREG procedure to produce the probability plot with an inset for the engine lifetimes:

```
ods graphics on;
proc lifereg data=Fan;
   model Lifetime*Censor( 1 ) = / d = Weibull;
   probplot
   ppout
   npintervals=simul
   ;
   inset;
run;
ods graphics off;
```

The resulting graphical output is shown in Output 48.5.1. The estimated CDF, a line representing the maximum likelihood fit, and pointwise parametric confidence bands are plotted in the body of Output 48.5.1. The values of right-censored observations are plotted along the bottom of the graph. The "Cumulative Probability Estimates" table is also created in Output 48.5.2.

Output 48.5.1 Probability Plot for the Fan Data



Output 48.5.2 CDF Estimates

Cumulative Probability Estimates

Lifetime	Cumulative Probability	Simultaneous 95% Confidence Limits Lower	Upper	Kaplan-Meier Estimate	Kaplan-Meier Standard Error
0.45	0.0071	0.0007	0.2114	0.0143	0.0142
1.15	0.0215	0.0033	0.2114	0.0288	0.0201
1.15	0.0360	0.0073	0.2168	0.0433	0.0244
1.6	0.0506	0.0125	0.2304	0.0580	0.0282
2.07	0.0666	0.0190	0.2539	0.0751	0.0324
2.07	0.0837	0.0264	0.2760	0.0923	0.0361
2.08	0.1008	0.0344	0.2972	0.1094	0.0392
3.1	0.1189	0.0436	0.3223	0.1283	0.0427
3.45	0.1380	0.0535	0.3471	0.1477	0.0460
4.6	0.1602	0.0653	0.3844	0.1728	0.0510
6.1	0.1887	0.0791	0.4349	0.2046	0.0581
8.75	0.2488	0.0884	0.6391	0.2930	0.0980

Example 48.6: Probability Plotting—Arbitrary Censoring

Table 48.7 contains microprocessor failure data (Nelson 1990). Units were inspected at predetermined time intervals. The data consist of inspection interval endpoints (in hours) and the number of units failing in each interval. A missing (.) lower endpoint indicates left censoring, and a missing upper endpoint indicates right censoring. These can be thought of as semi-infinite intervals with a lower (upper) endpoint of negative (positive) infinity for left (right) censoring.

Table 48.7 Interval-Censored Data

Lower Endpoint	Upper Endpoint	Number Failed
.	6	6
6	12	2
24	48	2
24	.	1
48	168	1
48	.	839
168	500	1
168	.	150
500	1000	2
500	.	149
1000	2000	1
1000	.	147
2000	.	122

The following SAS statements create the SAS data set Micro:

```
data Micro;
   input t1 t2 f ;
   datalines;
. 6 6
6 12 2
12 24 0
24 48 2
24 .  1
48 168 1
48 .   839
168 500 1
168 .   150
500 1000 2
500 .    149
1000 2000 1
1000 . 147
2000 . 122
;
run;
```

The following SAS statements compute the nonparametric Turnbull estimate of the cumulative distribution function and create a lognormal probability plot:

```
ods graphics on;
proc lifereg data=Micro;
   model ( t1 t2 ) = / d=lognormal intercept=25 scale=5;
   weight f;
   probplot
   pupper = 10
   itprintem
   printprobs
   maxitem = (1000,25)
   ppout;
   inset;
run;
ods graphics off;
```

The two initial values INTERCEPT=25 and SCALE=5 in the MODEL statement are used to aid convergence in the model-fitting algorithm.

The following tables are created by the PROBPLOT statement in addition to the standard tabular output from the MODEL statement. Output 48.6.1 shows the iteration history for the Turnbull estimate of the CDF for the microprocessor data. With both options ITPRINTEM and PRINTPROBS specified in the PROBPLOT statement, this table contains the log likelihoods and interval probabilities for every 25th iteration and the last iteration. It would contain only the log likelihoods if the option PRINTPROBS were not specified.

Output 48.6.1 Iteration History for the Turnbull Estimate

The LIFEREG Procedure

Iteration History for the Turnbull Estimate of the CDF

Iteration	Loglikelihood	(., 6)	(6, 12)	(24, 48)	(48, 168)
		(168, 500)	(500, 1000)	(1000, 2000)	(2000, .)
0	−1133.4051	0.125	0.125	0.125	0.125
		0.125	0.125	0.125	0.125
25	−104.16622	0.00421644	0.00140548	0.00140648	0.00173338
		0.00237846	0.00846094	0.04565407	0.93474475
50	−101.15151	0.00421644	0.00140548	0.00140648	0.00173293
		0.00234891	0.00727679	0.01174486	0.96986811
75	−101.06641	0.00421644	0.00140548	0.00140648	0.00173293
		0.00234891	0.00727127	0.00835638	0.9732621
100	−101.06534	0.00421644	0.00140548	0.00140648	0.00173293
		0.00234891	0.00727125	0.00801814	0.97360037
125	−101.06533	0.00421644	0.00140548	0.00140648	0.00173293
		0.00234891	0.00727125	0.00798438	0.97363413
130	−101.06533	0.00421644	0.00140548	0.00140648	0.00173293
		0.00234891	0.00727125	0.007983	0.97363551

The table in Output 48.6.2 summarizes the Turnbull estimates of the interval probabilities, the reduced gradients, and Lagrange multipliers as described in the section "Arbitrarily Censored Data" on page 3048.

Output 48.6.2 Summary for the Turnbull Algorithm

Lower Lifetime	Upper Lifetime	Probability	Reduced Gradient	Lagrange Multiplier
.	6	0.0042	0	0
6	12	0.0014	0	0
24	48	0.0014	0	0
48	168	0.0017	0	0
168	500	0.0023	0	0
500	1000	0.0073	-7.219342E-9	0
1000	2000	0.0080	-0.037063236	0
2000	.	0.9736	0.0003038877	0

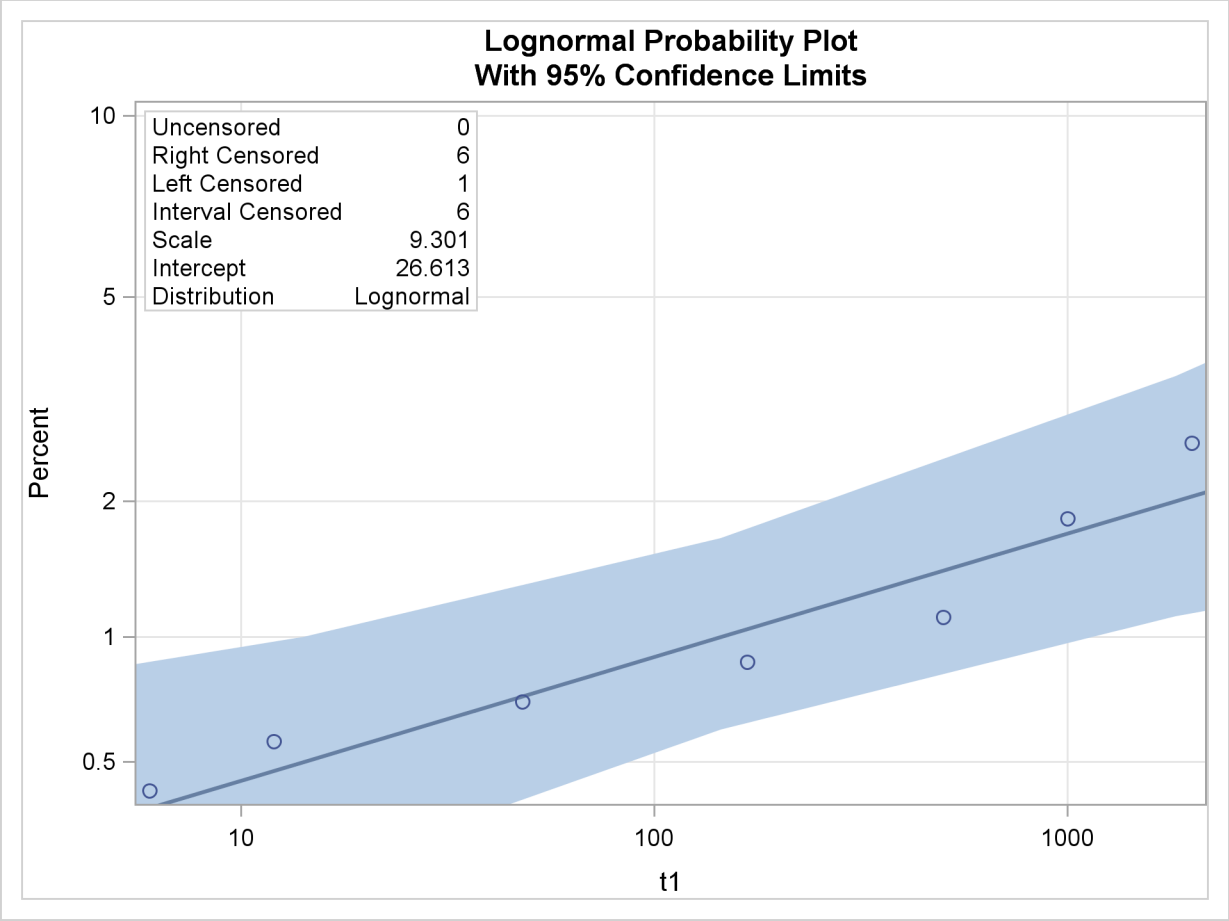
Output 48.6.3 shows the final estimate of the CDF, along with standard errors and nonparametric confidence limits. Two kinds of nonparametric confidence limits, pointwise or simultaneous, are available. The default is the pointwise nonparametric confidence limits. You can specify the simultaneous nonparametric confidence limits by using the NPINTERVALS=SIMUL option.

Output 48.6.3 Final CDF Estimates for Turnbull Algorithm

		Cumulative Probability Estimates			
			Pointwise 95% Confidence Limits		
Lower Lifetime	Upper Lifetime	Cumulative Probability	Lower	Upper	Standard Error
6	6	0.0042	0.0019	0.0094	0.0017
12	24	0.0056	0.0028	0.0112	0.0020
48	48	0.0070	0.0038	0.0130	0.0022
168	168	0.0088	0.0047	0.0164	0.0028
500	500	0.0111	0.0058	0.0211	0.0037
1000	1000	0.0184	0.0094	0.0357	0.0063
2000	2000	0.0264	0.0124	0.0553	0.0101

Output 48.6.4 shows the CDF estimates, maximum likelihood fit, and pointwise parametric confidence limits plotted on a lognormal probability plot.

Output 48.6.4 Lognormal Probability Plot for the Microprocessor Data



Example 48.7: Bayesian Analysis of Clinical Trial Data

Consider the data on melanoma patients from a clinical trial described in Ibrahim, Chen, and Sinha (2001). A partial listing of the data is shown in Output 48.7.1.

The survival time is modeled by a Weibull regression model with three covariates. An analysis of the right-censored survival data is performed with PROC LIFEREG to obtain Bayesian estimates of the regression coefficients by using the following SAS statements:

```
ods graphics on;
proc lifereg data=e1684;
   class Sex;
   model Survtime*Survcens(1)=Age Sex Perform / dist=Weibull;
   bayes WeibullShapePrior=gamma;
run;
ods graphics off;
```

Output 48.7.1 Clinical Trial Data

Obs	survtime	survcens	age	sex	perform
1	1.57808	2	35.9945	1	0
2	1.48219	2	41.9014	1	0
3	7.33425	1	70.2164	2	0
4	0.65479	2	58.1753	2	1
5	2.23288	2	33.7096	1	0
6	9.38356	1	47.9726	1	0
7	3.27671	2	31.8219	2	0
8	0.00000	1	72.3644	2	0
9	0.80274	2	40.7151	2	0
10	9.64384	1	32.9479	1	0
11	1.66575	2	35.9205	1	0
12	0.94247	2	40.5068	2	0
13	1.68767	2	57.0384	1	0
14	5.94247	2	63.1452	1	0
15	2.34247	2	62.0630	1	0
16	0.89863	2	56.5342	1	1
17	9.03288	1	22.9945	2	0
18	9.63014	1	18.4712	1	0
19	0.52603	2	41.2521	1	0
20	1.82192	2	29.5178	1	0

Maximum likelihood estimates of the model parameters shown in Output 48.7.2 are displayed by default.

Output 48.7.2 Maximum Likelihood Parameter Estimates

The LIFEREG Procedure

Bayesian Analysis

Analysis of Maximum Likelihood Parameter Estimates

Parameter		DF	Estimate	Standard Error	95% Confidence Limits	
Intercept		1	2.4402	0.3716	1.7119	3.1685
age		1	−0.0115	0.0070	−0.0253	0.0023
sex	1	1	−0.1170	0.1978	−0.5046	0.2707
sex	2	0	0.0000	.	.	.
perform		1	0.2905	0.3222	−0.3411	0.9220
Scale		1	1.2537	0.0824	1.1021	1.4260
Weibull Shape		1	0.7977	0.0524	0.7012	0.9073

Since no prior distributions for the regression coefficients were specified, the default uniform improper distributions shown in the "Uniform Prior for Regression Coefficients" table in Output 48.7.3 are used. The specified gamma prior for the Weibull shape parameter is also shown in Output 48.7.3.

Output 48.7.3 Model Parameter Priors

The LIFEREG Procedure

Bayesian Analysis

Uniform Prior for Regression Coefficients

Parameter	Prior
Intercept	Constant
age	Constant
sex1	Constant
perform	Constant

Independent Prior Distributions for Model Parameters

Parameter	Prior Distribution	Hyperparameters			
Weibull Shape	Gamma	Shape	0.001	Inverse Scale	0.001

Fit statistics, descriptive statistics, interval statistics, and the sample parameter correlation matrix for the posterior sample are displayed in the tables in Output 48.7.4. Since noninformative prior distributions for the regression coefficients were used, the mean and standard deviations of the posterior distributions for the model parameters are close to the maximum likelihood estimates and standard errors.

Output 48.7.4 Posterior Sample Statistics

Fit Statistics

DIC (smaller is better)	875.156
pD (effective number of parameters)	4.935

The LIFEREG Procedure

Bayesian Analysis

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
Intercept	10000	2.4762	0.3794	2.2140	2.4697	2.7303
age	10000	−0.0117	0.00717	−0.0165	−0.0117	−0.00693
sex1	10000	−0.1261	0.2024	−0.2622	−0.1253	0.0125
perform	10000	0.3279	0.3352	0.0966	0.3152	0.5407
WeibShape	10000	0.7826	0.0517	0.7467	0.7820	0.8167

Output 48.7.4 *continued*

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
Intercept	0.050	1.7530	3.2344	1.7558	3.2366
age	0.050	−0.0256	0.00231	−0.0255	0.00236
sex1	0.050	−0.5235	0.2654	−0.5217	0.2666
perform	0.050	−0.3079	1.0233	−0.3291	0.9937
WeibShape	0.050	0.6850	0.8864	0.6803	0.8807

Posterior Correlation Matrix

Parameter	Intercept	age	sex1	perform	Weib Shape
Intercept	1.0000	−.8983	−.2988	−.0759	−.1422
age	−.8983	1.0000	−.0467	−.0401	0.0685
sex1	−.2988	−.0467	1.0000	0.0874	0.0485
perform	−.0759	−.0401	0.0874	1.0000	−.0320
WeibShape	−.1422	0.0685	0.0485	−.0320	1.0000

The default diagnostic statistics are displayed in Output 48.7.5. See the section "Assessing Markov Chain Convergence" on page 156 for more details on Bayesian convergence diagnostics.

Output 48.7.5 Convergence Diagnostics

The LIFEREG Procedure

Bayesian Analysis

Posterior Autocorrelations

Parameter	Lag 1	Lag 5	Lag 10	Lag 50
Intercept	0.0660	0.0089	0.0194	−0.0107
age	0.0035	−0.0134	0.0180	−0.0175
sex1	0.6230	0.0645	−0.0057	−0.0121
perform	0.6594	0.1132	0.0199	−0.0100
WeibShape	0.0923	0.0322	0.0050	0.0063

Geweke Diagnostics

Parameter	z	Pr > |z|
Intercept	1.4526	0.1463
age	−1.5941	0.1109
sex1	−0.6555	0.5121
perform	−0.1008	0.9197
WeibShape	0.0127	0.9899

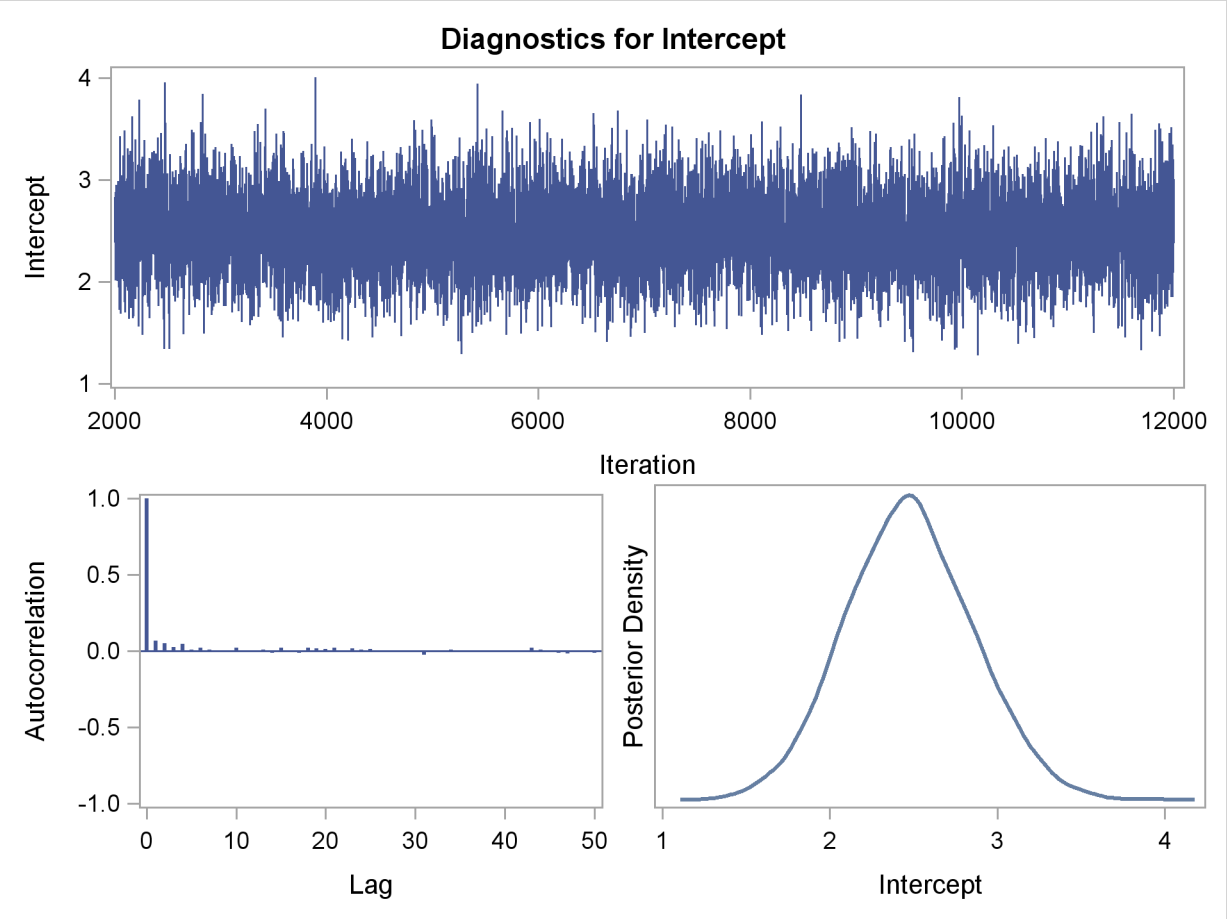
Output 48.7.5 *continued*

Effective Sample Sizes

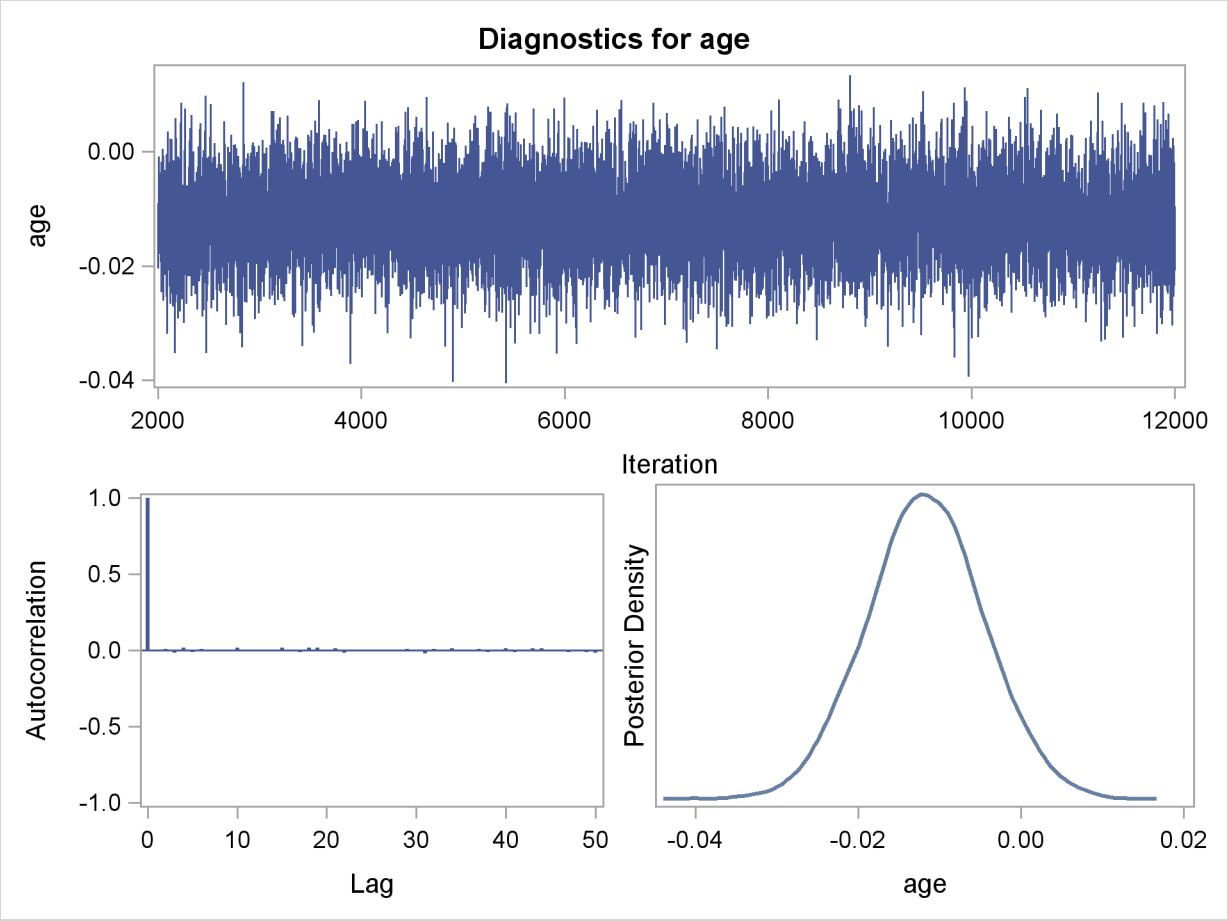
Parameter	ESS	Correlation Time	Efficiency
Intercept	7262.3	1.3770	0.7262
age	10000.0	1.0000	1.0000
sex1	2521.7	3.9656	0.2522
perform	2071.8	4.8266	0.2072
WeibShape	6585.9	1.5184	0.6586

Trace, autocorrelation, and density plots for the seven model parameters are shown in Output 48.7.6 through Output 48.7.10. These plots show no indication that the Markov chains have not converged. See the sections "Assessing Markov Chain Convergence" on page 156 and "Visual Analysis via Trace Plots" on page 156 for more information about assessing the convergence of the chain of posterior samples.

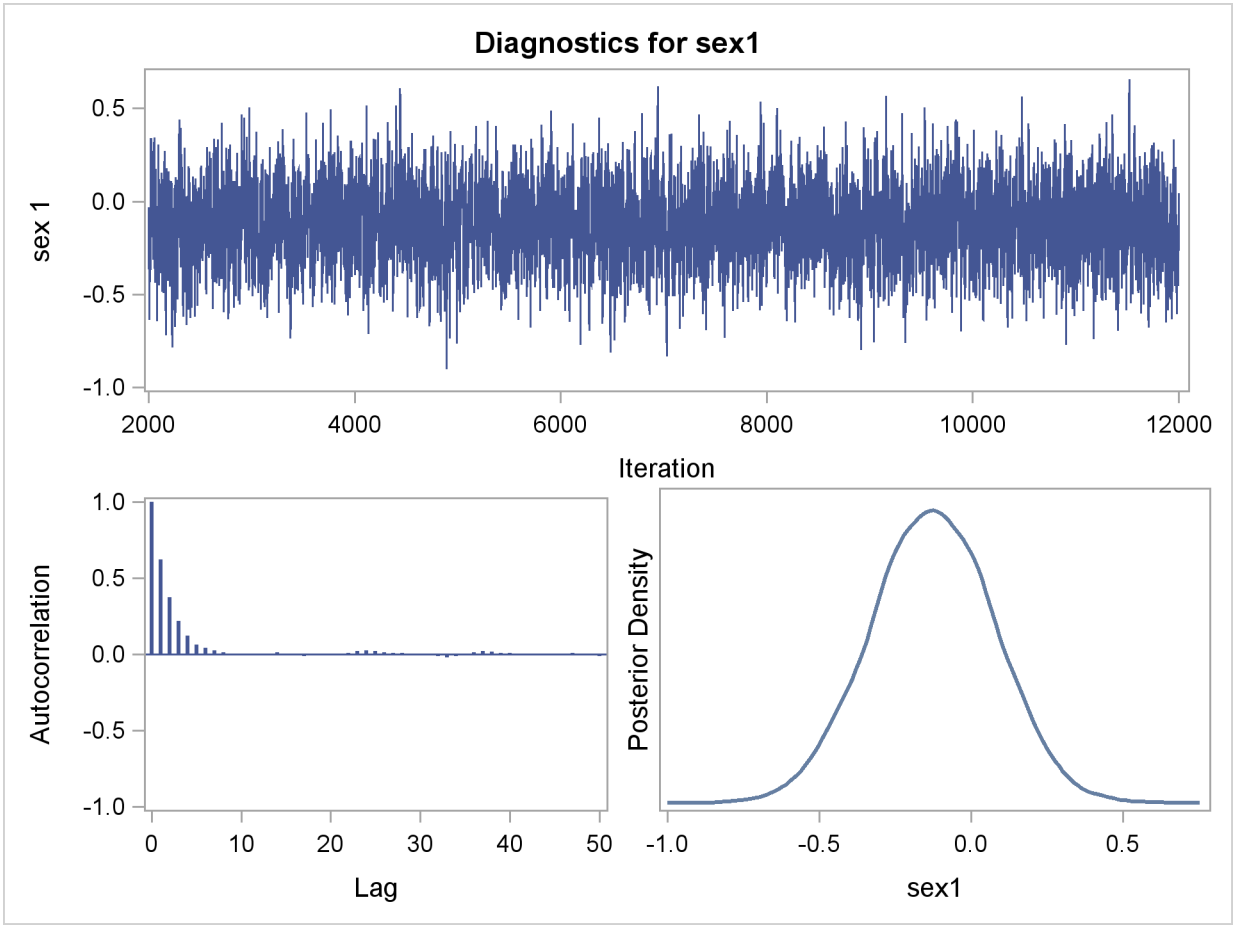
Output 48.7.6 Diagnostic Plots



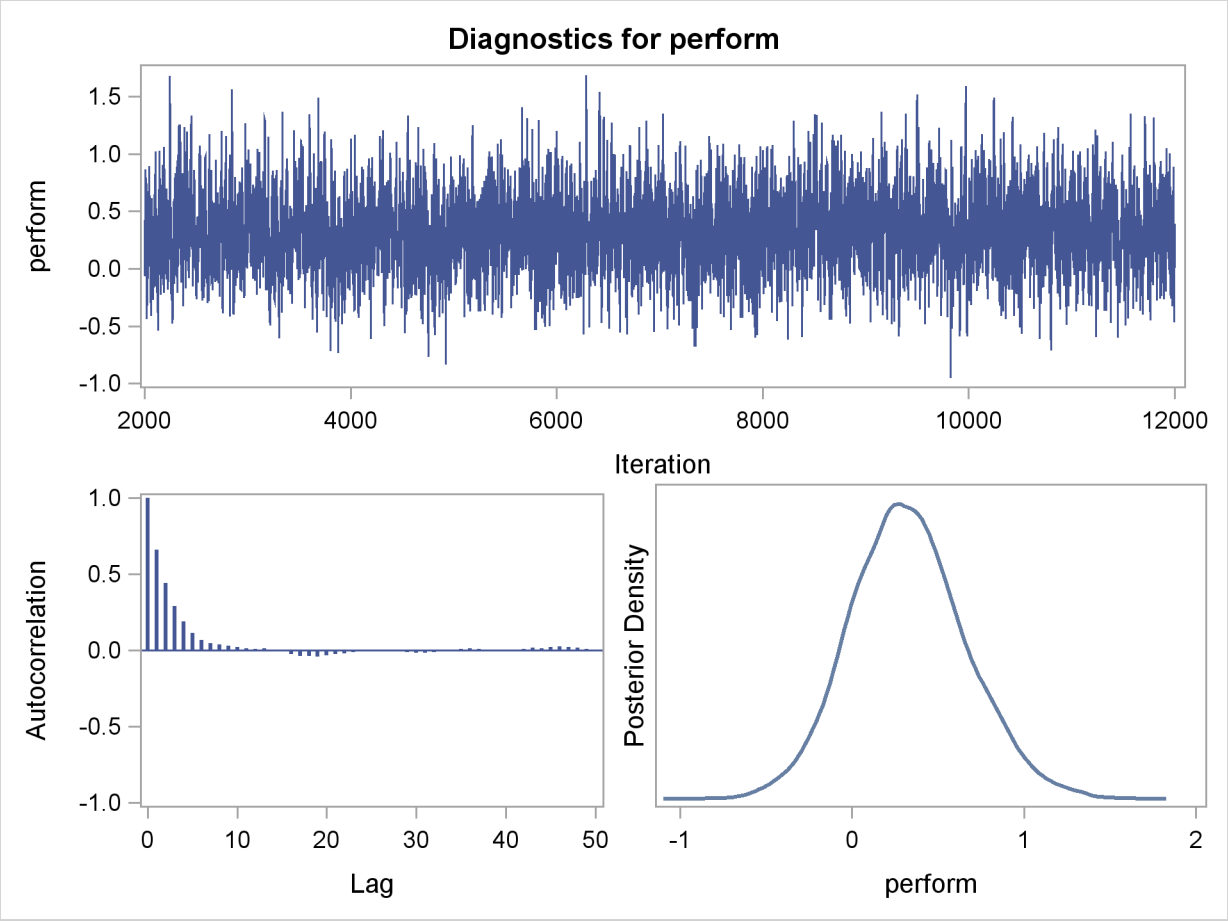
Output 48.7.7 Diagnostic Plots



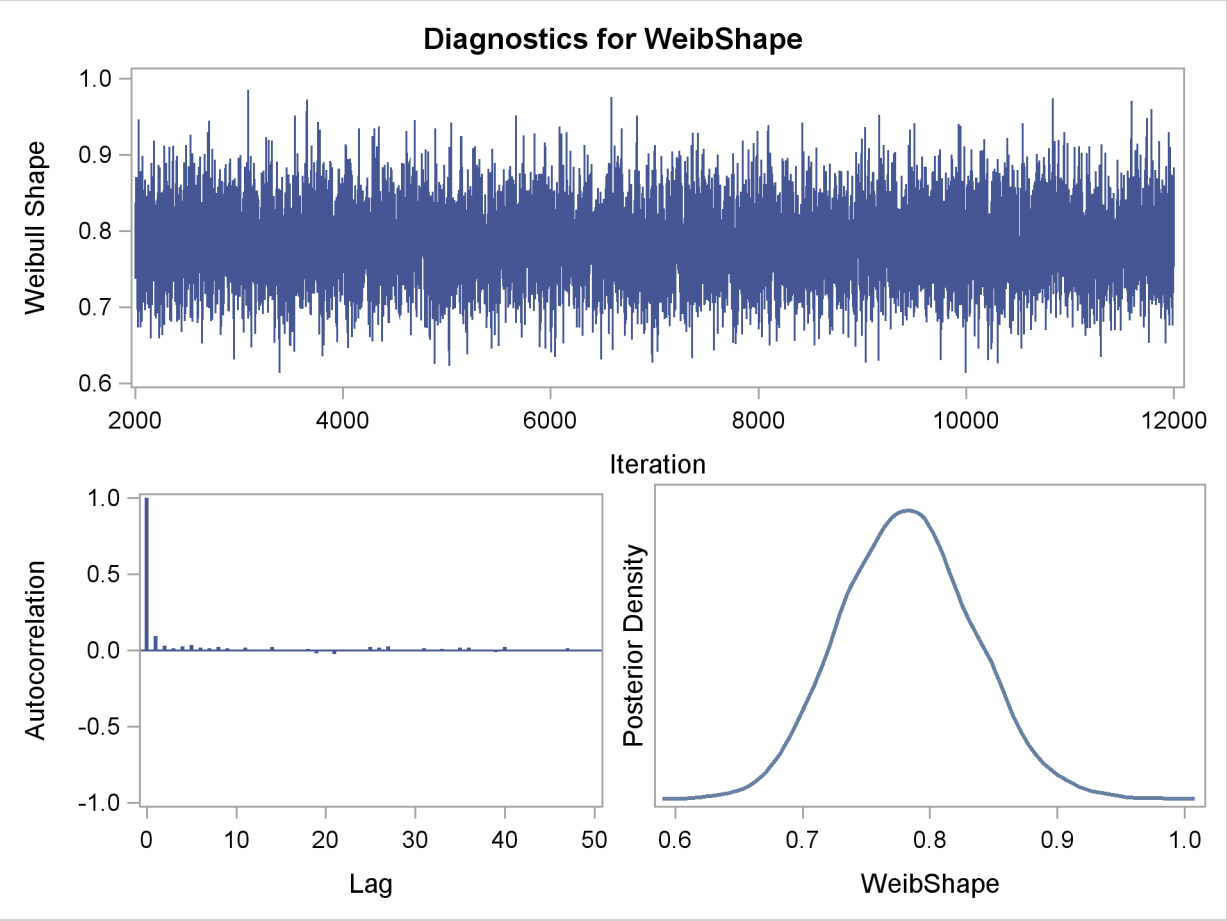
Output 48.7.8 Diagnostic Plots



Output 48.7.9 Diagnostic Plots



Output 48.7.10 Diagnostic Plots



References

Abernethy, R. B. (1996), *The New Weibull Handbook*, Second Edition, North Palm Beach, FL: Robert B. Abernethy.

Akaike, H. (1979), "A Bayesian Extension of the Minimum AIC Procedure of Autoregressive Model Fitting," *Biometrika*, 66, 237–242.

Akaike, H. (1981), "Likelihood of a Model and Information Criteria," *Journal of Econometrics*, 16, 3–14.

Cox, D. R. and Oakes, D. (1984), *Analysis of Survival Data*, London: Chapman & Hall.

Gentleman, R. and Geyer, C. J. (1994), "Maximum Likelihood for Interval Censored Data: Consistency and Computation," *Biometrika*, 81, 618–623.

Gilks, W. (2003), "Adaptive Metropolis Rejection Sampling (ARMS)," software from MRC Biostatistics Unit, Cambridge, UK, http://www.maths.leeds.ac.uk/~wally.gilks/adaptive.rejection/web_page/Welcome.html.

Gilks, W. R., Best, N. G., and Tan, K. K. C. (1995), "Adaptive Rejection Metropolis Sampling with Gibbs Sampling," *Applied Statistics*, 44, 455–472.

Gilks, W. R., Richardson, S., and Spiegelhalter, D. J. (1996), *Markov Chain Monte Carlo in Practice*, London: Chapman & Hall.

Gilks, W. R. and Wild, P. (1992), "Adaptive Rejection Sampling for Gibbs Sampling," *Applied Statistics*, 41, 337–348.

Greene, W. H. (1993), *Econometric Analysis*, Second Edition, New York: Macmillan.

Ibrahim, J. G., Chen, M. H., and Sinha, D. (2001), *Bayesian Survival Analysis*, New York: Springer-Verlag.

Kalbfleisch, J. D. and Prentice, R. L. (1980), *The Statistical Analysis of Failure Time Data*, New York: John Wiley & Sons.

Klein, J. P. and Moeschberger, M. L. (1997), *Survival Analysis: Techniques for Censored and Truncated Data*, New York: Springer-Verlag.

Lawless, J. F. (2003), *Statistical Model and Methods for Lifetime Data*, Second Edition, New York: John Wiley & Sons.

Maddala, G. S. (1983), *Limited-Dependent and Qualitative Variables in Econometrics*, New York: Cambridge University Press.

Meeker, W. Q. and Escobar, L. A. (1998), *Statistical Methods for Reliability Data*, New York: John Wiley & Sons.

Mroz, T. A. (1987), "The Sensitivity of an Empirical Model of Married Women's Work to Economic and Statistical Assumptions," *Econometrica*, 55, 765–799.

Nair, V. N. (1984), "Confidence Bands for Survival Functions with Censored Data: A Comparative Study," *Technometrics*, 26, 265–275.

Nelson, W. (1982), *Applied Life Data Analysis*, New York: John Wiley & Sons.

Nelson, W. (1990), *Accelerated Testing: Statistical Models, Test Plans, and Data Analyses*, New York: John Wiley & Sons.

Rao, C. R. (1973), *Linear Statistical Inference*, New York: John Wiley & Sons.

Simonoff, J. S. (2003), *Analyzing Categorical Data*, New York: Springer-Verlag.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van der Linde, A. (2002), "Bayesian Measures of Model Complexity and Fit," *Journal of the Royal Statistical Society, Series B*, 64(4), 583–616, with discussion.

Tobin, J. (1958), "Estimation of Relationships for Limited Dependent Variables," *Econometrica*, 26, 24–36.

Turnbull, B. W. (1976), "The Empirical Distribution Function with Arbitrarily Grouped, Censored and Truncated Data," *Journal of the Royal Statistical Society, Series B*, 38, 290–295.

Chapter 49

The LIFETEST Procedure

Contents

Overview: LIFETEST Procedure **3100**
Getting Started: LIFETEST Procedure **3101**
Syntax: LIFETEST Procedure **3110**
PROC LIFETEST Statement 3111
BY Statement 3124
FREQ Statement 3125
ID Statement 3125
STRATA Statement 3125
TEST Statement 3130
TIME Statement 3131
Details: LIFETEST Procedure **3131**
Missing Values 3131
Computational Formulas 3131
Product-Limit Method 3131
Life-Table Method 3134
Pointwise Confidence Limits in the OUTSURV= Data Set 3136
Simultaneous Confidence Intervals for Kaplan-Meier Curves 3137
Kernel-Smoothed Hazard Estimate 3140
Comparison of Two or More Groups of Survival Data 3142
Rank Tests for the Association of Survival Time with Covariates . . 3145
Computer Resources 3147
Output Data Sets 3148
OUTSURV= Data Set 3148
OUTTEST= Data Set 3150
Displayed Output 3150
ODS Table Names 3155
ODS Graphics 3156
Examples: LIFETEST Procedure **3158**
Example 49.1: Product-Limit Estimates and Tests of Association 3158
Example 49.2: Enhanced Survival Plot and Multiple-Comparison Adjustments 3172
Example 49.3: Life-Table Estimates for Males with Angina Pectoris 3176
References **3183**

Overview: LIFETEST Procedure

A common feature of lifetime or survival data is the presence of right-censored observations due either to withdrawal of experimental units or to termination of the experiment. For such observations, you know only that the lifetime exceeded a given value; the exact lifetime remains unknown. Such data cannot be analyzed by ignoring the censored observations because, among other considerations, the longer-lived units are generally more likely to be censored. The analysis methodology must correctly use the censored observations as well as the uncensored observations.

Texts that discuss the survival analysis methodology include Collett (1994), Cox and Oakes (1984), Kalbfleisch and Prentice (1980), Klein and Moeschberger (1997), Lawless (1982), and Lee (1992). Users interested in the theory should consult Fleming and Harrington (1991) and Andersen et al. (1992).

Usually, a first step in the analysis of survival data is the estimation of the distribution of the survival times. Survival times are often called *failure* times, and *event* times are uncensored survival times. The survival distribution function (SDF), also known as the survivor function, is used to describe the lifetimes of the population of interest. The SDF evaluated at t is the probability that an experimental unit from the population will have a lifetime exceeding t—that is,

$$S(t) = \Pr(T > t)$$

where $S(t)$ denotes the survivor function and T is the lifetime of a randomly selected experimental unit. The LIFETEST procedure can be used to compute nonparametric estimates of the survivor function either by the product-limit method (also called the Kaplan-Meier method) or by the life-table method (also called the actuarial method).

Some functions closely related to the SDF are the cumulative distribution function (CDF), the probability density function (PDF), and the hazard function. The CDF, denoted $F(t)$, is defined as $1 - S(t)$ and is the probability that a lifetime does not exceed t. The PDF, denoted $f(t)$, is defined as the derivative of $F(t)$, and the hazard function, denoted $h(t)$, is defined as $f(t)/S(t)$. If the life-table method is chosen, the estimates of the probability density function can also be computed. Plots of these estimates can be produced by a graphical or line printer device, or based on the output delivery system (ODS).

An important task in the analysis of survival data is the comparison of survival curves. It is of interest to determine whether the underlying populations of k ($k \geq 2$) samples have identical survivor functions. PROC LIFETEST provides nonparametric k-sample tests based on weighted comparisons of the estimated hazard rate of the individual population under the null and alternative hypotheses. Corresponding to various weight functions, a variety of tests can be specified, which include the log-rank test, Wilcoxon test, Tarone-Ware test, Peto-Peto test, modified Peto-Peto test, and Fleming-Harrington G_ρ family of tests. PROC LIFETEST also provides corresponding trend tests to detect ordered alternatives. Stratified tests can be specified to adjust for prognostic factors that affect the events rates in the various populations. A likelihood ratio test, based on an underlying exponential model, is also included to compare the survival curves of the samples.

There are other prognostic variables, called covariates, that are thought to be related to the failure time. These covariates can also be used to construct statistics to test for association between

the covariates and the lifetime variable. PROC LIFETEST can compute two such test statistics: censored data linear rank statistics based on the exponential scores and the Wilcoxon scores. The corresponding tests are known as the log-rank test and the Wilcoxon test, respectively. These tests are computed by pooling over any defined strata, thus adjusting for the stratum variables.

There have been a number of enhancements to PROC LIFETEST with this release, and a couple changes. The most noticeable enhancement is in the PLOTS= option through ODS Graphics. You can request the display of the number of subjects at risk in the survival plot, as well as the pointwise or simultaneous confidence limits for the survivor functions. For the product-limit method, you can also request a smoothed hazard function. For specific information about use of the ODS Graphics in PROC LIFETEST, see the PLOTS= option in the PROC LIFETEST statement. Another enhancement is to allow multiple-comparison adjustments for comparing survival curves through the ADJUST= and DIFF= options in the STRATA statement. See the section "Multiple-Comparison Adjustments" on page 3143 for the adjustment methods. Some of these new features are shown in Example 49.2.

One change in SAS 9.2 is that the calculation of confidence limits for the quartiles of survival time is based on the transformation specified by the CONFTYPE= option. Another change is that the SURVIVAL statement in SAS 9.1 is folded into the PROC LIFETEST statement; that is, options that were in the SURVIVAL statement can now be specified in the PROC LIFETEST statement. The SURVIVAL statement is no longer needed and it is not documented.

Getting Started: LIFETEST Procedure

You can use the LIFETEST procedure to compute nonparametric estimates of the survivor functions, to compare survival curves, and to compute rank tests for association of the failure time variable with covariates.

For simple analyses, only the PROC LIFETEST and TIME statements are required. Consider a sample of survival data. Suppose that the time variable is T and the censoring variable is C with value 1 indicating censored observations. The following statements compute the product-limit estimate for the sample:

```
proc lifetest;
   time t*c(1);
run;
```

You can use the STRATA statement to divide the data into various strata. A separate survivor function is then estimated for each stratum, and tests of the homogeneity of strata are performed. However, if the GROUP= option is also specified in the STRATA statement, the GROUP= variable is used to identify the samples whose survivor functions are to be compared, and the STRATA variables are used to define the strata for the stratified tests. You can specify covariates (prognostic variables) in the TEST statement, and PROC LIFETEST computes linear rank statistics to test the effects of these covariates on survival.

For example, consider the results of a small randomized trial on rats. Suppose you randomize 40

rats that have been exposed to a carcinogen into two treatment groups (Drug X and Placebo). The event of interest is death from cancer induced by the carcinogen. The response is the time from randomization to death. Four rats died of other causes; their survival times are regarded as censored observations. Interest lies in whether the survival distributions differ between the two treatments.

The following DATA step creates the data set Exposed, which contains four variables: Days (survival time in days from treatment to death), Status (censoring indicator variable: 0 if censored and 1 if not censored), Treatment (treatment indicator), and Sex (gender: F if female and M if male).

```
proc format;
   value Rx 1='Drug X' 0='Placebo';
data exposed;
  input Days  Status Treatment Sex $ @@;
  format Treatment Rx.;
  datalines;
179   1   1   F   378   0   1   M
256   1   1   F   355   1   1   M
262   1   1   M   319   1   1   M
256   1   1   F   256   1   1   M
255   1   1   M   171   1   1   F
224   0   1   F   325   1   1   M
225   1   1   F   325   1   1   M
287   1   1   M   217   1   1   F
319   1   1   M   255   1   1   F
264   1   1   M   256   1   1   F
237   0   0   F   291   1   0   M
156   1   0   F   323   1   0   M
270   1   0   M   253   1   0   M
257   1   0   M   206   1   0   F
242   1   0   M   206   1   0   F
157   1   0   F   237   1   0   M
249   1   0   M   211   1   0   F
180   1   0   F   229   1   0   F
226   1   0   F   234   1   0   F
268   0   0   M   209   1   0   F
;
```

PROC LIFETEST is invoked as follows to compute the product-limit estimate of the survivor function for each treatment and to compare the survivor functions between the two treatments.

```
ods graphics on;
proc lifetest data=Exposed plots=(survival(atrisk) logsurv);
   time Days*Status(0);
   strata Treatment;
run;
ods graphics off;
```

In the TIME statement, the survival time variable, Days, is crossed with the censoring variable, Status, with the value 0 indicating censoring. That is, the values of Days are considered censored if the corresponding values of Status are 0; otherwise, they are considered as event times. In the STRATA statement, the variable Treatment is specified, which indicates that the data are to be divided into strata based on the values of Treatment. Graphics results are enabled through ODS with the specifi-

cation of the **ods graphics on** statement. Two plots are requested through the PLOTS= option—a plot of the survival curves with at risk numbers and a plot of the negative log of the survival curves.

The results of the analysis are displayed in the following figures.

Figure 49.1 displays the product-limit survival estimate for the Drug X group (Treatment=1). The figure lists, for each observed time, the survival estimate, failure rate, standard error of the estimate, cumulative number of failures, and number of subjects remaining in the study.

Figure 49.1 Survivor Function Estimate for the Drug X-Treated Rats

The LIFETEST Procedure

Stratum 1: Treatment = Drug X

Product-Limit Survival Estimates

Days	Survival	Failure	Survival Standard Error	Number Failed	Number Left
0.000	1.0000	0	0	0	20
171.000	0.9500	0.0500	0.0487	1	19
179.000	0.9000	0.1000	0.0671	2	18
217.000	0.8500	0.1500	0.0798	3	17
224.000*	.	.	.	3	16
225.000	0.7969	0.2031	0.0908	4	15
255.000	.	.	.	5	14
255.000	0.6906	0.3094	0.1053	6	13
256.000	.	.	.	7	12
256.000	.	.	.	8	11
256.000	.	.	.	9	10
256.000	0.4781	0.5219	0.1146	10	9
262.000	0.4250	0.5750	0.1135	11	8
264.000	0.3719	0.6281	0.1111	12	7
287.000	0.3188	0.6813	0.1071	13	6
319.000	.	.	.	14	5
319.000	0.2125	0.7875	0.0942	15	4
325.000	.	.	.	16	3
325.000	0.1063	0.8938	0.0710	17	2
355.000	0.0531	0.9469	0.0517	18	1
378.000*	0.0531	.	.	18	0

NOTE: The marked survival times are censored observations.

Figure 49.2 displays summary statistics of survival times for the Drug X group. It contains estimates of the 25th, 50th, and 75th percentiles and the corresponding 95% confidence limits. The median survival time for rats in this treatment is 256 days. The mean and standard error are also displayed; however, it is noted that these values are underestimated because the largest observed time is censored and the estimation is restricted to the largest event time.

Figure 49.2 Summary Statistics of Survival Times for Drug X-Treated Rats

Quartile Estimates

Percent	Point Estimate	95% Confidence Interval Transform	[Lower	Upper)
75	319.000	LOGLOG	256.000	355.000
50	256.000	LOGLOG	255.000	319.000
25	255.000	LOGLOG	171.000	256.000

Mean	Standard Error
271.131	11.877

NOTE: The mean survival time and its standard error were underestimated because the largest observation was censored and the estimation was restricted to the largest event time.

Figure 49.3 and Figure 49.4 display the survival estimates and the summary statistics of the survival times for Placebo (Treatment=0). The median survival time for rats in this treatment is 235 days.

Figure 49.3 Survivor Function Estimate for Placebo-Treated Rats

The LIFETEST Procedure

Stratum 2: Treatment = Placebo

Product-Limit Survival Estimates

Days	Survival	Failure	Survival Standard Error	Number Failed	Number Left
0.000	1.0000	0	0	0	20
156.000	0.9500	0.0500	0.0487	1	19
157.000	0.9000	0.1000	0.0671	2	18
180.000	0.8500	0.1500	0.0798	3	17
206.000	.	.	.	4	16
206.000	0.7500	0.2500	0.0968	5	15
209.000	0.7000	0.3000	0.1025	6	14
211.000	0.6500	0.3500	0.1067	7	13
226.000	0.6000	0.4000	0.1095	8	12
229.000	0.5500	0.4500	0.1112	9	11
234.000	0.5000	0.5000	0.1118	10	10
237.000	0.4500	0.5500	0.1112	11	9
237.000*	.	.	.	11	8
242.000	0.3938	0.6063	0.1106	12	7
249.000	0.3375	0.6625	0.1082	13	6
253.000	0.2813	0.7188	0.1038	14	5
257.000	0.2250	0.7750	0.0971	15	4
268.000*	.	.	.	15	3
270.000	0.1500	0.8500	0.0891	16	2
291.000	0.0750	0.9250	0.0693	17	1
323.000	0	1.0000	.	18	0

NOTE: The marked survival times are censored observations.

Figure 49.4 Summary Statistics of Survival Times for Placebo-Treated Rats

Quartile Estimates

Percent	Point Estimate	Transform	95% Confidence Interval [Lower	Upper)
75	257.000	LOGLOG	237.000	323.000
50	235.500	LOGLOG	206.000	253.000
25	207.500	LOGLOG	156.000	229.000

Mean	Standard Error
235.156	10.211

A summary of the number of censored and event observations is shown in Figure 49.5. The figure lists, for each stratum, the number of event and censored observations, and the percentage of censored observations.

Figure 49.5 Number of Event and Censored Observations

Summary of the Number of Censored and Uncensored Values

Stratum	Treatment	Total	Failed	Censored	Percent Censored
1	Drug X	20	18	2	10.00
2	Placebo	20	18	2	10.00
Total		40	36	4	10.00

Figure 49.6 displays the graph of the product-limit survivor function estimates versus survival time. The two treatments differ primarily at larger survival times. Note the number of subjects at risk in the plot. You can display the number of subjects at risk at specific time points by using the ATRISK= option.

Figure 49.6 Plot of Estimated Survivor Functions

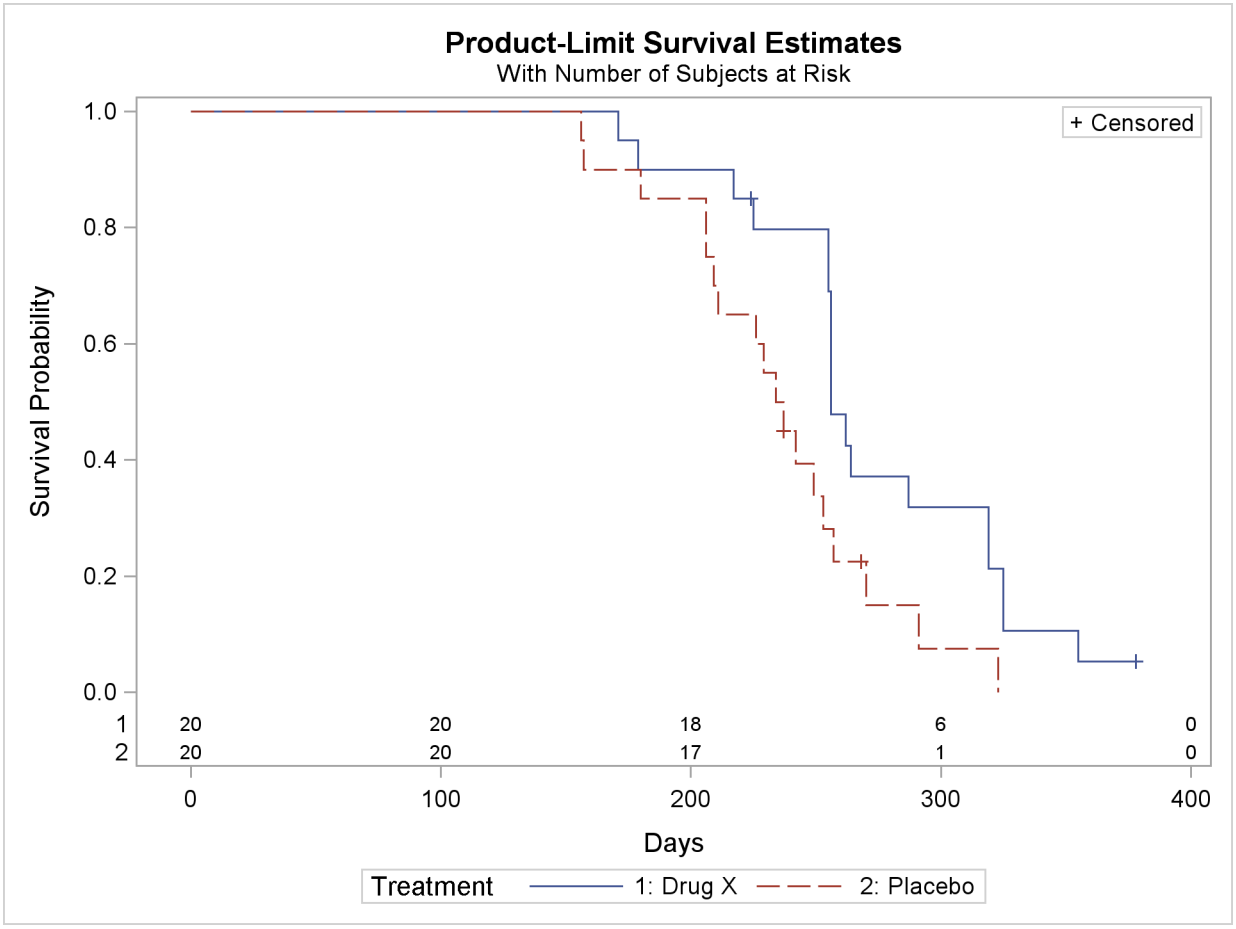
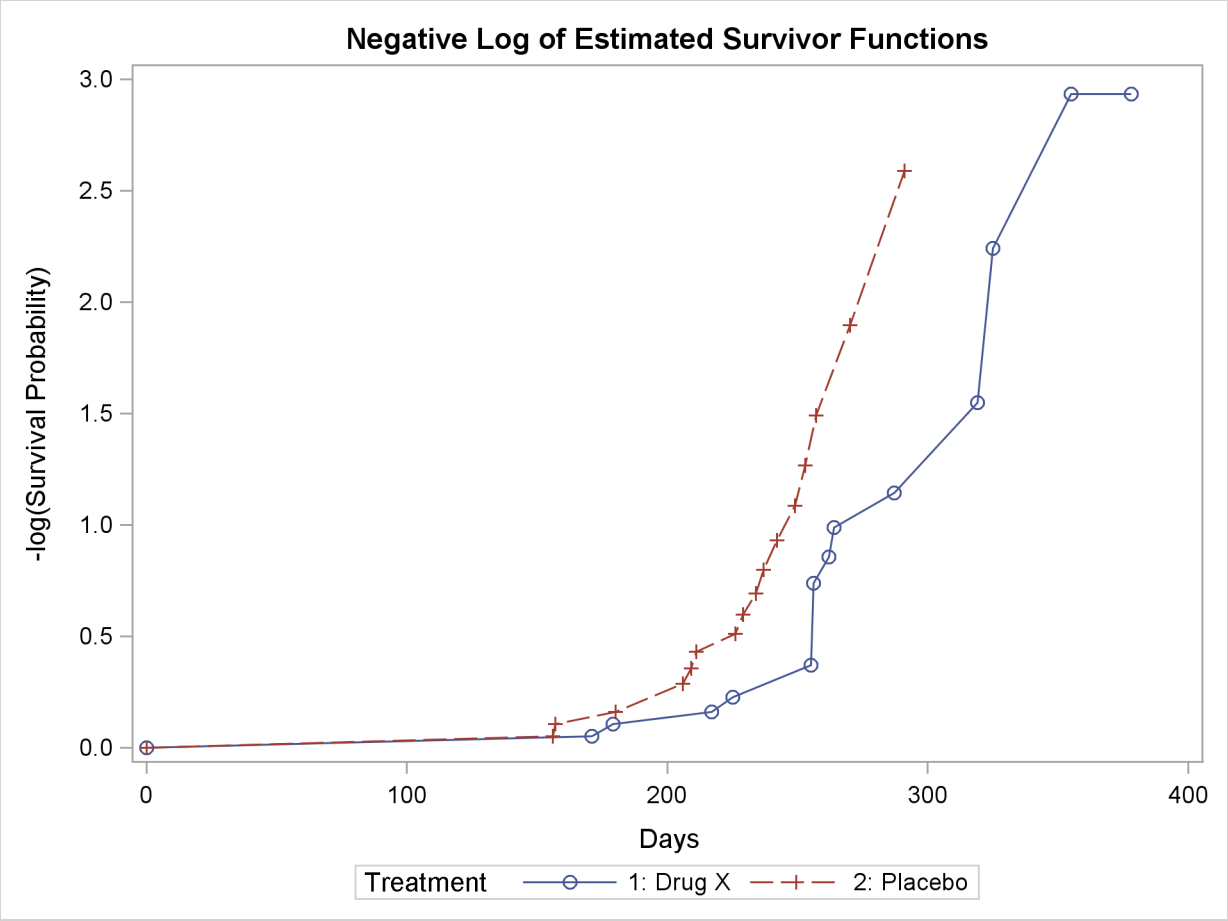


Figure 49.7 displays the graph of the log survivor function estimates versus survival time. Neither curve approximates a straight line through the origin—the exponential model is not appropriate for the survival data.

Figure 49.7 Plot of Estimated Negative Log Survivor Functions



Note that these graphical displays are generated through ODS. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

Results of the comparison of survival curves between the two treatments are shown in Figure 49.8. The rank tests for homogeneity indicate a significant difference between the treatments (p=0.0175 for the log-rank test and p=0.0249 for the Wilcoxon test). Rats treated with Drug X live significantly longer than those treated with Placebo. Since the survival curves for the two treatments differ primarily at longer survival times, the Wilcoxon test, which places more weight on shorter survival times, becomes less significant than the log-rank test. As noted earlier, the exponential model is not appropriate for the given survival data; consequently, the result of the likelihood ratio test should be ignored.

Figure 49.8 Results of the Two-sample Tests

Test of Equality over Strata

Test	Chi-Square	DF	Pr > Chi-Square
Log-Rank	5.6485	1	0.0175
Wilcoxon	5.0312	1	0.0249
-2Log(LR)	0.1983	1	0.6561

Next, suppose male rats and female rats are thought to have different survival rates, and you want to assess the treatment effect while adjusting for the gender differences. By specifying the variable Sex in the STRATA statement as a stratifying variable and by specifying the variable Treatment in the GROUP= option, you can carry out a stratified test to test Treatment while adjusting for Sex. The test statistics are computed by pooling over the strata defined by the values of Sex, thus controlling for the effect of Sex. The NOTABLE option is added to the PROC LIFETEST statement as follows to avoid estimating a survival curve for each gender.

```
proc lifetest data=Exposed notable;
   time Days*Status(0);
   strata Sex / group=Treatment;
run;
```

Results of the stratified tests are shown in Figure 49.9. The treatment effect is statistically significant for both the log-rank test (p=0.0071) and the Wilcoxon test (p=0.0150). As compared to the results of the unstratified tests in Figure 49.8, the significance of the treatment effect has been sharpened by controlling for the effect of the gender of the subjects.

Figure 49.9 Results of the Stratified Two-Sample Tests

The LIFETEST Procedure

Stratified Test of Equality over Group

Test	Chi-Square	DF	Pr > Chi-Square
Log-Rank	7.2466	1	0.0071
Wilcoxon	5.9179	1	0.0150

Since Treatment is a binary variable, another way to study the effect of Treatment is to carry out a censored linear rank test with Treatment as an independent variable. Although this test is less popular than the two-sample test, nevertheless, in situations where the independent variables are continuous and are difficult to discretize, it might be infeasible to perform a k-sample test. To compute the censored linear rank statistics to test the Treatment effect, Treatment is specified in the TEST statement as follows.

```
proc lifetest data=Exposed notable;
   time Days*Status(0);
   test Treatment;
run;
```

Results of the linear rank tests are shown Figure 49.10. The *p*-values are very similar to those of the two-sample tests in Figure 49.8.

Figure 49.10 Results of Linear Rank Tests of Treatment

The LIFETEST Procedure

Univariate Chi-Squares for the Wilcoxon Test

Variable	Test Statistic	Standard Error	Chi-Square	Pr > Chi-Square
Treatment	3.9525	1.7524	5.0875	0.0241

Univariate Chi-Squares for the Log-Rank Test

Variable	Test Statistic	Standard Error	Chi-Square	Pr > Chi-Square
Treatment	6.2708	2.6793	5.4779	0.0193

With Sex as a prognostic factor that you want to control, you can compute a stratified linear rank statistic to test the effect of Treatment by specifying Sex in the STRATA statement and Treatment in the TEST statement as in the following program. The TEST=NONE option is specified in the STRATA statement to suppress the two-sample tests for Sex.

```
proc lifetest data=Exposed notable;
   time Days*Status(0);
   strata Sex / test=none;
   test Treatment;
run;
```

Results of the stratified linear rank tests are shown in Figure 49.11. The *p*-values are very similar to those of the stratified tests in Figure 49.9.

Figure 49.11 Results of Stratified Linear Rank Tests of Treatment

The LIFETEST Procedure

Univariate Chi-Squares for the Wilcoxon Test

Variable	Test Statistic	Standard Error	Chi-Square	Pr > Chi-Square
Treatment	4.2372	1.7371	5.9503	0.0147

Figure 49.11 *continued*

Univariate Chi-Squares for the Log-Rank Test

Variable	Test Statistic	Standard Error	Chi-Square	Pr > Chi-Square
Treatment	6.8021	2.5419	7.1609	0.0075

Syntax: LIFETEST Procedure

The following statements are available in PROC LIFETEST:

PROC LIFETEST < *options* > ;
  BY *variables* ;
  FREQ *variable* ;
  ID *variables* ;
  STRATA *variable* < *(list)* > < . . . *variable* < *(list)* > > < */options* > ;
  TEST *variables* ;
  TIME *variable* < **censor(list)* > ;

The simplest use of PROC LIFETEST is to request the nonparametric estimates of the survivor function for a sample of survival times. In such a case, only the PROC LIFETEST statement and the TIME statement are required. You can use the STRATA statement to divide the data into various strata. A separate survivor function is then estimated for each stratum, and tests of the homogeneity of strata are performed. However, if the GROUP= option is also specified in the STRATA statement, stratified tests are carried out to test the k samples defined by the GROUP= variable while controlling for the effect of the STRATA variables. You can specify covariates in the TEST statement. PROC LIFETEST computes linear rank statistics to test the effects of these covariates on survival.

The PROC LIFETEST statement invokes the procedure. All statements except the TIME statement are optional, and there is no required order for the statements following the PROC LIFETEST statement. The TIME statement is used to specify the variables that define the survival time and censoring indicator. The STRATA statement specifies a variable or set of variables defining the strata for the analysis. The TEST statement specifies a list of numeric covariates to be tested for their association with the response survival time. Each variable is tested individually, and a joint test statistic is also computed. The ID statement provides a list of variables whose values are used to identify observations in the product-limit estimates of the survival function. When only the TIME statement appears, no strata are defined and no tests of homogeneity are performed.

PROC LIFETEST Statement

PROC LIFETEST < *options* > **;**

The PROC LIFETEST statement invokes the procedure. Optionally, this statement identifies an input and an OUTSURV= data set, and specifies the computation details of the survivor function estimation. The options listed in Table 49.1 are available in the PROC LIFETEST statement and are described in alphabetic order. If no options are requested, PROC LIFETEST computes and displays the product-limit estimate of the survivor function; and if an **ods graphics on** statement is specified, a plot of the estimated survivor function is also displayed.

Table 49.1 Options Available in the PROC LIFETEST Statement

Option	Description
Input and Output Data Sets	
DATA=	specifies the input SAS data set
OUTSURV=	names an output data set to contain survival estimates and confidence limits
OUTTEST=	names an output data set to contain rank test statistics for association of survival time with covariates
Nonparametric Estimation	
INTERVALS=	specifies interval endpoints for life-table estimates
NELSON	adds the Nelson-Aalen estimates
METHOD=	specifies the method to compute survivor function
NINTERVAL=	specifies the number of intervals for life-table estimates
WIDTH=	specifies the width of intervals for life-table estimates
Confidence Limits for Survivorship	
ALPHA=	sets the confidence level for interval estimation estimates
BANDMAXTIME=	specifies the maximum time for confidence band
BANDMINTIME=	specifies the minimum time for confidence band
CONFBAND=	specifies the type of confidence band in the OUTSURV= data set
CONFTYPE=	specifies the transformation applied to the survivor function to obtain confidence limits
Line Printer Plots	
FORMCHAR(1,2,7,9)=	defines the characters used for line printer plot axes
LINEPRINTER	specifies that plots be produced by a line printer
MAXTIME=	specifies the maximum time value for plotting
NOCENSPLOT	suppresses the plot of censored observations
PLOTS=	specifies the plots to display
ODS Graphics	
MAXTIME=	specifies the maximum time value for plotting
PLOTS=	specifies plots to display
Traditional Graphics	
ANNOTATE=	specifies an Annotate data set that adds features to plots
CENSOREDSYMBOL=	defines the symbol used for censored observations in plots

Table 49.1 *continued*

Option	Description
DESCRIPTION=	specifies the string that appears in the description field of the PROC GREPLAY master menu for the plots
EVENTSYMBOL=	specifies the symbol used for event observations in plots
GOUT=	specifies the graphics catalog name for saving graphics output
LANNOTATE=	specifies an input data set that contains variables for local annotation
MAXTIME=	specifies the maximum time value for plotting
PLOTS=	specifies the plots to display
Control Output	
NOPRINT	suppresses the display of printed output
NOTABLE	suppresses the display of survival function estimates
INTERVALS=	displays only the product-limit estimate for the smallest time within each specified interval
TIMELIST=	specifies a list of time points at which the Kaplan-Meier estimates are displayed
REDUCEOUT	specifies that only INTERVAL= or TIMELIST= observations be listed in the OUTSURV= data set
Miscellaneous	
ALPHAQT=	sets the confidence level for survival time quartiles
MISSING	allows missing values to be a stratum level
SINGULAR=	sets the tolerance for testing singularity of covariance matrix of rank statistics
STDERR	outputs the standard error for the survival estimators to the OUTSURV= data set
TIMELIM=	specifies the time limit used to estimate the mean survival time and its standard error

The PLOTS= option in the PROC LIFETEST statement specifies the plots to display. You can select one of the following three types of graphics in PROC LIFETEST: line printer, traditional, and ODS. If you specify the LINEPRINTER option, you get line printer plots; otherwise you get the traditional graphics if ODS Graphics is not enabled, and you get the ODS Graphics plots if the **`ods graphics on`** statement is specified. The following table depicts the behavior regarding the type of graphics produced:

ODS Graphics	PROC LIFETEST Options		Behavior
OFF			no graphics
OFF		LINEPRINTER	no graphics
OFF	PLOTS=SURVIVAL		traditional graphics
OFF	PLOTS=SURVIVAL	LINEPRINTER	line printer plot
ON			ODS Graphics survival plot
ON		LINEPRINTER	no graphics
ON	PLOTS=SURVIVAL		ODS Graphics
ON	PLOTS=SURVIVAL	LINEPRINTER	line printer plot

ODS Graphics is now the preferred method of requesting graphics. Many new features have been added to the ODS Graphics plots in PROC LIFETEST. For example, you can display the number of subjects at risk in a survival plot through ODS Graphics, but such a feature is not available in traditional graphics or line printer plots. The PLOTS= option syntax is documented separately for each type of graphics and is preceded by a heading that indicates the graphics type.

ALPHA=*α*

specifies the level of significance α for the $100(1-\alpha)\%$ confidence intervals for the survivor, hazard, and density functions. For example, the option ALPHA=0.05 requests the 95% confidence limits for the survivor function. The default value is 0.05.

ALPHAQT=*α*

specifies the significance level α for the $100(1-\alpha)\%$ confidence intervals for the quartiles of the survival time. For example, the option ALPHAQT=0.05 requests a 95% confidence interval for the quartiles of the survival time. The default value is 0.05.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an input data set that contains appropriate variables for annotation of the traditional graphics. The ANNOTATE= option enables you to add features (for example, labels explaining extreme observations) to plots produced on graphics devices. The ANNOTATE= option cannot be used if the LINEPRINTER option or the **ods graphics on** statement is specified. The data set specified must be an ANNOTATE= type data set, as described in *SAS/GRAPH Software: Reference*.

The data set specified with the ANNOTATE= option in the PROC LIFETEST statement is "global" in the sense that the information in this data set is displayed in every plot produced by a single invocation of PROC LIFETEST.

BANDMAXTIME=*value*

BANDMAX=*value*

specifies the maximum time for the confidence bands. The default is the largest observed event time. If the specified BANDMAX= time exceeds the largest observed event time, it is truncated to the largest observed event time.

BANDMINTIME=*value*

BANDMIN=*value*

specifies the minimum time for the confidence bands. The default is the smallest observed event time. For the equal-precision band, if the BANDMIN= value is less than the smallest observed event time, it is defaulted to the smallest observed event time.

CENSOREDSYMBOL=*name* | *'string'*

CS=*name* | *'string'*

specifies the symbol value for the censored observations in traditional graphics. The value, *name* or *'string'*, is the symbol value specification allowed in SAS/GRAPH software. The default is CS=CIRCLE. If you want to omit plotting the censored observations, specify CS=NONE. The CENSOREDSYMBOL= option cannot be used if the LINEPRINTER option or the **ods graphics on** statement is specified.

CONFBAND=*keyword*

specifies the confidence bands to be output to the OUTSURV= data set. Confidence bands are available only for the product-limit method. You can use the following *keywords*:

ALL	outputs both the Hall-Wellner and the equal-precision confidence bands.
EP	outputs the equal-precision confidence bands.
HW	outputs the Hall-Wellner confidence bands.

CONFTYPE=*keyword*

specifies the transformation applied to $S(t)$ to obtain the pointwise confidence intervals and the confidence bands for the survivor function as well as the confidence intervals for the quartiles of the survival times. The following *keywords* can be used and the default is CONFTYPE=LOGLOG.

ASINSQRT the arcsine-square root transformation

$$g(x) = \sin^{-1}(\sqrt{x})$$

LOGLOG the log-log transformation

$$g(x) = \log(-\log(x))$$

This is also referred to as the log cumulative hazard transformation since it is applying the logarithmic function log(.) to the cumulative hazard function. Collett (1994) and Lachin (2000) refer it as the complementary log-log transformation.

LINEAR the identity transformation

$$g(x) = x$$

LOG the logarithmic transformation

$$g(x) = \log(x)$$

LOGIT the logit transformation

$$g(x) = \log\left(\frac{x}{1-x}\right)$$

DATA=*SAS-data-set*

names the SAS data set used by PROC LIFETEST. By default, the most recently created SAS data set is used.

DESCRIPTION='*string* '

DES='*string* '

specifies a descriptive string of up to 256 characters that appears in the "Description" field of the traditional graphics catalog. The description does not appear in the plots. By default, PROC LIFETEST assigns a description of the form PLOT OF *vname* vs *hname*, where *vname* and *hname* are the names of the *y* variable and the *x* variable, respectively. The DESCRIPTION= option cannot be used if the LINEPRINTER option or the **ods graphics on** statement is specified.

EVENTSYMBOL=*name* | *'string'*

ES=*name* | *'string'*

specifies the symbol value for the event observations in traditional graphics. The value, *name* or *'string'*, is the symbol value specification allowed in SAS/GRAPH software. The default is ES=NONE. The EVENTSYMBOL= option cannot be used if the LINEPRINTER option or the **ods graphics on** statement is specified.

FORMCHAR(1,2,7,9)=*'string'*

defines the characters used for constructing the vertical and horizontal axes of the line printer plots. The string should be four characters. The first and second characters define the vertical and horizontal bars, respectively, which are also used in drawing the *steps* of the product-limit survival function. The third character defines the tick mark for the axes, and the fourth character defines the lower-left corner of the plot. If the FORMCHAR option in PROC LIFETEST is not specified, the value supplied, if any, with the system option FORMCHAR= is used. The default is FORMCHAR(1,2,7,9)='|-+-'. Any character or hexadecimal string can be used to customize the plot appearance. To send the plot output to a printer with the IBM graphics character set (1 or 2) or display it directly on your PC screen, you can use the hexadecimal representation

```
formchar(1,2,7,9)='B3C4C5C0'x
```

or the system option

```
formchar='B3C4DACBFC3C5B4C0C1D9'x
```

Refer to the chapter titled "The PLOT Procedure" in the *Base SAS Procedures Guide* or the section "System Options" in *SAS Language Reference: Dictionary* for further information.

GOUT=*graphics-catalog*

specifies the graphics catalog for saving traditional graphics output from PROC LIFETEST. The default is WORK.GSEG. The GOUT= option cannot be used if the LINEPRINTER option or the **ods graphics on** statement is specified. For more information, refer to the chapter titled "The GREPLAY Procedure" in *SAS/GRAPH Software: Reference*.

INTERVALS=*values*

specifies a list of interval endpoints for the life-table method. These endpoints must all be nonnegative numbers. The initial interval is assumed to start at zero whether or not zero is specified in the list. Each interval contains its lower endpoint but does not contain its upper endpoint. When this option is used with the product-limit method, it reduces the number of survival estimates displayed by displaying only the estimates for the smallest time within each specified interval. The INTERVALS= option can be specified in any of the following ways:

list separated by blanks	`intervals=1 3 5 7`
list separated by commas	`intervals=1,3,5,7`
x to *y*	`intervals=1 to 7`
x to *y* by *z*	`intervals=1 to 7 by 1`
combination of the above	`intervals=1,3 to 5,7`

For example, the specification

```
intervals=5,10 to 30 by 10
```

produces the set of intervals

$$\{[0, 5), [5, 10), [10, 20), [20, 30), [30, \infty)\}$$

LANNOTATE=*SAS-data-set*

LANN=*SAS-data-set*

specifies an input data set that contains variables for local annotation of traditional graphics. You can use the LANNOTATE= option to specify a different annotation for each BY group, in which case the BY variables must be included in the LANNOTATE= data set. The LANNOTATE= option cannot be used if the LINEPRINTER option or the `ods graphics on` statement is specified. The data set specified must be an ANNOTATE= type data set, as described in *SAS/GRAPH Software: Reference*.

If there is no BY-group processing, the ANNOTATE= and LANNOTATE= options have the same effects.

LINEPRINTER

LS

specifies that plots are produced by a line printer instead of by a graphical device.

MAXTIME=*value*

specifies the maximum value of the time variable allowed on the plots so that outlying points do not determine the scale of the time axis of the plots. This option affects only the displayed plots and has no effect on any calculations.

METHOD=*type*

specifies the method used to compute the survival function estimates. Valid values for *type* are as follows:

PL

KM

specifies that product-limit (PL) or Kaplan-Meier (KM) estimates be computed.

ACT

LIFE

LT

specifies that life-table (or actuarial) estimates be computed.

By default, METHOD=PL.

MISSING

allows missing values for numeric variables and blank values for character variables as valid stratum levels. See the section "Missing Values" on page 3131 for details.

By default, PROC LIFETEST does not use observations with missing values for any stratum variables.

NELSON

AALEN

produces the Nelson-Aalen estimates of the cumulative hazards and the corresponding standard errors. This option is ignored unless METHOD=PL.

NINTERVAL=*value*

specifies the number of intervals used to compute the life-table estimates of the survivor function. This parameter is overridden by the WIDTH= option or the INTERVALS= option. When you specify the NINTERVAL= option, PROC LIFETEST tries to find an interval that results in round numbers for the endpoints. Consequently, the number of intervals can be different from the number requested. Use the INTERVALS= option to control the interval endpoints. The default is NINTERVAL=10.

NOCENSPLOT

NOCENS

requests that the plot of censored observations be suppressed when the LINEPRINTER and PLOTS= options are specified. This option is not needed when the life-table method is used to compute the survival estimates, since the plot of censored observations is not produced.

NOPRINT

suppresses the display of output. This option is useful when only an output data set is needed. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System" for more information.

NOTABLE

suppresses the display of survival function estimates. Only the number of censored and event times, plots, and test results is displayed.

OUTSURV=*SAS-data-set*

OUTS=*SAS-data-set*

creates an output SAS data set to contain the estimates of the survival function and corresponding confidence limits for all strata. See the section "OUTSURV= Data Set" on page 3148 for more information about the contents of the OUTSURV= data set.

OUTTEST=*SAS-data-set*

OUTT=*SAS-data-set*

creates an output SAS data set to contain the overall chi-square test statistic for association with failure time for the variables in the TEST statement, the values of the univariate rank test statistics for each variable in the TEST statement, and the estimated covariance matrix of the univariate rank test statistics. See the section "OUTTEST= Data Set" on page 3150 for more information about the contents of the OUTTEST= data set.

Line Printer PLOTS= Option

PLOTS=*plot-request*

PLOTS=*(plot-requests)*

controls the line printer plots produced. You must also specify the LINEPRINTER option to obtain line printer plots. When you specify only one plot-request, you can omit the parentheses around the plot-request. Here are some examples:

```
plots=s
plots=(s ls lls)
```

The *plot-requests* include the following:

CENSORED

C

specifies a plot of censored observations (product-limit method only).

SURVIVAL

S

specifies a plot of the estimated SDF versus time.

LOGSURV

LS

specifies a plot of the $-\log$(estimated SDF) versus time.

LOGLOGS

LLS

specifies a plot of the $\log(-\log$(estimated SDF)) versus log(time).

HAZARD

H

specifies a plot of the estimated hazard function versus time (life-table method only).

PDF

P

specifies a plot of the estimated probability density function versus time (life-table method only).

ODS Graphics PLOTS= Option

PLOTS< *(global-plot-option)* > **=** *plot-request* < *(options)* >

PLOTS< *(global-plot-option)* > **=** *(plot-request* < *(options)* > < *. . . plot-request* <*(options)* > >*)*

controls the plots produced using ODS Graphics. When you specify only one *plot-request*, you can omit the parentheses around the *plot-request*. Here are some examples:

```
plots=none
plots=(survival(atrisk=100 to 350 by 50) logsurv)
plots(only)=hazard
```

You must enable ODS Graphics before requesting plots—for example, like this:

```
ods graphics on;

proc lifetest plots=survival(atrisk);
   time T*Status(0);
run;

ods graphics off;
```

For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." If you have enabled ODS Graphics but do not specify the PLOTS= option, then PROC LIFETEST produces a plot of the estimated survivor functions by default. The only global plot option follows:

ONLY
specifies that only the specified plots in the list be produced; otherwise, the default survivor function plot is also displayed.

The *plot-requests* and *plot-request* options include the following.

ALL
produces all appropriate plots. For the product-limit method, specifying PLOTS=ALL is equivalent to specifying PLOTS=(SURVIVAL LOGSURV LOGLOGLS HAZARD); for the life-table method, it is the same as specifying PLOTS=(SURVIVAL LOGSURV LOGLOGS DENSITY HAZARD).

HAZARD < (*hazard-options*) >

H < *hazard-options* >
plots the estimated hazard functions. Kernel-smoothed estimates are produced for the product-limit method. The following *hazard-options* can be specified for the product-limit method, but only the CL option can be used for the life-table method.

BANDWIDTH=*bandwidth-option*

BW=*bandwidth-option*
specifies what bandwidth is chosen for the kernel-smoothing and how it is chosen. You can specify one of the following *bandwidth-options*.

value
sets the bandwidth to the given *value*.

numeric-list
selects the bandwidth from the given *numeric-list* that minimizes the mean integrated squared error.

RANGE(*lower,upper***)**
selects the bandwidth from the interval *(lower, upper)* that minimizes the mean integrated squared error. PROC LIFETEST uses the golden section search algorithm to find the minimum. If there is more than one local minimum in the interval, there is no guarantee that the local minimum found is also the global minimum.

See the section "Optimal Bandwidth" on page 3141 for details about the mean integrated squared error. If the BANDWIDTH= option is not specified, the default is BANDWIDTH= RANGE(0.2b,20b), where $b = \frac{g_u - g_l}{8n^{.2}}$, g_l and g_u are the values of the GRIDL= and GRIDU= options, respectively, and n is the total number of noncensored observations.

GRIDL=*number*

specifies the lower grid limit for the kernel-smoothed estimate. The default value is the time origin.

GRIDU=*number*

specifies the upper grid limit for the kernel-smoothed estimate. The default value equals the maximum event time

KERNEL=*kernel-option*

specifies the kernel used. The choices are as follows:

BIWEIGHT

BW

$$K_{BW}(x) = \frac{15}{16}(1 - x^2)^2, \qquad -1 \le x \le 1$$

EPANECHNIKOV

E

$$K_E(x) = \frac{3}{4}(1 - x^2), \qquad -1 \le x \le 1$$

UNIFORM

U

$$K_U(x) = \frac{1}{2}, \qquad -1 \le x \le 1$$

The default is KERNEL=EPANECHNIKOV.

NMINGRID=*number*

specifies the number of grid points in determining the mean integrated square error (MISE). The default value is 51.

NGRID=*number*

specifies the number of grid points. The default is 101.

CL

displays the pointwise confidence limits for the smoothed hazard.

LOGLOGS

LLS

plots the log of negative log of estimated survivor functions versus the log of time.

LOGSURV

LS

plots the negative log of estimated survivor functions versus time.

NONE

suppresses all plots.

PDF < (CL) >

P < (CL) >

plots the estimated probability density functions (life-table method only). Pointwise confidence limits are displayed optionally by specifying the CL option.

SURVIVAL < (*survival-options*) >

S < (*survival-options*) >

plots the estimated survivor functions. Censored times are plotted as a plus sign on the product-limit curves unless the NOCENSOR option is specified. You can customized the display by using the following *survival-options*.

ATRISK < = *number-list* >

displays the numbers of subjects at risk at the given times. The *number-list* identifies the times at which the numbers at risk are displayed. If the *number-list* is not specified, PROC LIFETEST uses the default list $\{0, a, 2a, \ldots, n \times a\}$, where a and n are computed by the following algorithm. Let m be the MAXTIME= value or the largest observed time if the MAXTIME= option is not specified; let $b = 10^{\text{ceil}[\log_{10}(m)-1]}$, where ceil() is the ceiling function.

$$\begin{aligned} a &= \begin{cases} \frac{b}{2} & \text{if } m < 0.25b \\ 2b & \text{if } m > 0.75b \\ b & \text{otherwise} \end{cases} \\ n &= \text{integral value of } m/a \end{aligned}$$

CB < =*keyword* >

displays the confidence bands (that is, simulataneous confidence intervals) for the survivor functions. You can specify one of the following *keywords*. The default is CB=HW.

ALL

displays both the equal-precision and the Hall-Wellner bands.

EP

displays the equal-precision band.

HW

displays the Hall-Wellner confidence band.

CL

displays the pointwise confidence limits for the survivor functions.

FAILURE

F

changes all the displays for survivor functions to those for the failure functions. For example, if both the FAILURE and CL options are specified, the plot displays the failure curves as well as the pointwise confidence limits for the failure functions.

NOCENSOR
suppresses plotting the censored times on the product-limit survival curves (product-limit method only).

STRATA=*strata-option*
specifies how to display the survival/failure curves for multiple strata. This option has no effect if there is only one stratum. You can choose one of the following *strata options*:

INDIVIDUAL
UNPACK
specifies that a separate plot be displayed for each stratum.

OVERLAY
specifies that the survival/failure curves for the strata be overlaid in one plot.

PANEL
specifies that separate plots for the strata be organized into panels of two or four plots, depending on the number of strata.

The default is STRATA=OVERLAY.

TEST
displays the *p*-value of a homogeneity test specified in the STRATA statement. If more than one test is produced, the test is chosen in the following order: LOGRANK, WILCOXON, TARONE, PETO, MODPETO, FLEMING, and LR.

Traditional Graphics PLOTS= Option

PLOTS=*plot-request* **<(NAME=***name* | '*string*'**)>**
PLOTS=(*plot-request* **<(NAME=***name* | '*string*'**)> <, . . . ,** *plot-request* **<(NAME=***name* | '*string*'**)> >)**

controls plots produced in traditional graphics. To obtain traditional graphics, you must neither enable ODS Graphics nor specify the LINEPRINTER option. For each *plot-request*, you can use the NAME= option to specify a name to identify the plot. The name can be specified as a SAS name or as a quoted string of up to 256 characters. Only the first eight characters are used as the entry name in the GOUT= catalog. The *plot-requests* include the following:

SURVIVAL
S
plots the estimated survivor functions versus time.

LOGSURV
LS
plots the negative log of estimated survivor functions versus time.

LOGLOGS

LLS

plots the log of negative log of estimated survivor functions versus the log of time.

HAZARD

H

plots estimated hazard function versus time (life-table method only).

PDF

P

plots the estimated probability density function versus time (life-table method only).

When you specify only one *plot-request*, you can omit the parentheses around the *plot-request*. Here are some examples:

```
plots=s
plots=(s(name=Surv2), h(name=Haz2))
```

The latter requests a plot of the estimated survivor function versus time and a plot of the estimated hazard function versus time, with Surv2 and Haz2 as their names in the GOUT= catalog, respectively.

REDUCEOUT

specifies that the OUTSURV= data set contain only those observations that are included in the INTERVALS= or TIMELIST= option. This option has no effect if the OUTSURV= option is not specified. It also has no effect if neither the INTERVALS= option nor the TIMELIST= option is specified.

SINGULAR=*value*

specifies the tolerance for testing singularity of the covariance matrix for the rank test statistics. The test requires that a pivot for sweeping a covariance matrix be at least this number times a norm of the matrix. The default value is 1E–12.

STDERR

specifies that the standard error of the survivor function (SDF_STDERR) be output to the OUTSURV= data set. If the life-table method is used, the standard error of the density function (PDF_STDERR) and the standard error of the hazard function (HAZ_STDERR) are also output.

TIMELIM=*time-limit*

specifies the time limit used in the estimation of the mean survival time and its standard error. The mean survival time can be shown to be the area under the Kaplan-Meier survival curve. However, if the largest observed time in the data is censored, the area under the survival curve is not a closed area. In such a situation, you can choose a time limit L and estimate the mean survival curve limited to a time L (Lee 1992, pp. 72–76). This option is ignored if the largest observed time is an event time. Valid *time-limit* values are as follows:

EVENT

LET

specifies that the time limit L be the largest event time in the data. TIMELIM=EVENT is the default.

OBSERVED

LOT

specifies that the time limit L be the largest observed time in the data.

number

specifies that the time limit L be the given *number*. The *number* must be positive and at least as large as the largest event time in the data.

TIMELIST=*number-list*

specifies a list of time points at which the Kaplan-Meier estimates are displayed. The time points are listed in the column labeled Timelist. Since the Kaplan-Meier survival curve is a decreasing step function, each given time point falls in an interval that has a constant survival estimate. The event time that corresponds to the beginning of the time interval is displayed along with its survival estimate.

WIDTH=*value*

sets the width of the intervals used in the life-table calculation of the survival function. This parameter is overridden by the INTERVALS= option.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC LIFETEST to obtain separate analyses on observations in groups defined by the BY variables.

The BY statement is more efficient than the STRATA statement for defining strata in large data sets. However, if you use the BY statement to define strata, PROC LIFETEST does not pool over strata for testing the association of survival time with covariates, nor does it test for homogeneity across the BY groups.

Interval size is computed separately for each BY group. When intervals are determined by default, they can be different for each BY group. To make intervals the same for each BY group, use the INTERVALS= option in the PROC LIFETEST statement.

When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.

- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the LIFETEST procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

FREQ Statement

FREQ *variable* **;**

The *variable* in the FREQ statement identifies a variable containing the frequency of occurrence of each observation. PROC LIFETEST treats each observation as if it appeared *n* times, where *n* is the value of the FREQ variable for the observation. The FREQ statement is useful for producing life tables when the data are already in the form of a summary data set. If not an integer, the frequency value is truncated to an integer. If the frequency value is missing or less than one, the observation is not used.

ID Statement

ID *variables* **;**

The ID variable values are used to label the observations of the product-limit survival function estimates. SAS format statements can be used to format the values of the ID variables.

STRATA Statement

STRATA *variable* < *(list)* > < *. . . variable* < *(list)* > > < */options* > **;**

The STRATA statement identifies the variables that determine the strata levels. Strata are formed according to the nonmissing values of these variables. The MISSING option can be used to allow missing values as a valid stratum level. Other options enable you to specify various *k*-sample tests, stratified tests, or trend tests and to make multiple-comparison adjustments for paired differences.

In the preceding syntax, *variable* is a variable whose values determine the stratum levels, and *list* is a list of endpoints for a numeric variable. The values for *variable* can be formatted or unformatted. If the variable is a character variable, or if the variable is numeric and no list appears, then the strata are defined by the unique values of the strata variable. More than one variable can be specified in the STRATA statement, and each numeric variable can be followed by a list. Each interval

contains its lower endpoint but not its upper endpoint. The corresponding strata are formed by the combination of levels. If a variable is numeric and is followed by a list, then the levels for that variable correspond to the intervals defined by the list. The initial interval is assumed to start at $-\infty$, and the final interval is assumed to end at ∞.

The specification of STRATA variables can have any of the following forms:

list separated by blanks	`strata age(5 10 20 30)`
list separated by commas	`strata age(5,10,20,30)`
x to *y*	`strata age(5 to 10)`
x to *y* by *z*	`strata age(5 to 30 by 10)`
combination of the above	`strata age(5,10 to 50 by 10)`

For example, the specification

```
strata Age(5,20 to 50 by 10) Sex;
```

indicates the following levels for the Age variable:

$$\{(-\infty, 5), [5, 20), [20, 30), [30, 40), [40, 50), [50, \infty)\}$$

This statement also specifies that the Age strata be further subdivided by values of the variable Sex. In this example, there are 6 age groups by 2 sex groups, forming a total of 12 strata.

The specification of several strata variables, such as

```
strata A B C;
```

is equivalent to the A*B*C syntax of the TABLES statement in the FREQ procedure. The number of strata levels usually grows very rapidly with the number of STRATA variables, so you must be cautious when specifying the list of STRATA variables.

When comparing more than two survival curves, a *k*-sample test tells you whether the curves are significantly different from each other, but it does not identify which pairs of curves are different. A multiple-comparison adjustment of the *p*-values for the paired comparisons retains the same overall false positives as the *k*-sample test. Two types of paired comparisons can be made: comparisons between all pairs of curves and comparisons between a control curve and all other curves. You use the DIFF= option to specify the comparison type, and you use the ADJUST= option to select a method of multiple-comparison adjustments.

Table 49.2 summaries the options available in the STRATA statement.

Table 49.2 Options Available in the STRATA Statement

Option	Description
Homogeneity Tests	
GROUP=	specifies the group variable for stratified tests
NODETAIL	suppresses printing the test statistic and covariance matrix
NOTEST	suppresses any tests
TEST=	specifies tests corresponding to various weight functions

Table 49.2 *continued*

Option	Description
TREND	requests a trend test
Multiple Comparisons	
ADJUST=	requests a multiple-comparison adjustment
DIFF=	specifies the type of differences to consider
Missing Strata Value	
MISSING	allows missing values as valid stratum values

Options in the STRATA statement are specified after a slash ("/"). The following list describes these options.

ADJUST=*method*

specifies the multiple-comparison method for adjusting the p-values of the paired tests. See the section "Multiple-Comparison Adjustments" on page 3143 for mathematical details; also see Westfall et al. (1999). The adjustment methods include the following:

BONFERRONI
BON

applies the Bonferroni correction to the raw p-values.

DUNNETT

performs Dunnett's two-tailed comparisons of the control group with all other groups. PROC LIFETEST uses the factor-analytic covariance approximation described in Hsu (1992) and identifies the adjustment in the results as "Dunnett-Hsu." Note that ADJUST=DUNNETT is incompatible with DIFF=ALL.

SCHEFFE

performs Scheffé's multiple-comparison adjustment.

SIDAK

applies the Šidák correction to the raw p-values.

SMM
GTE

performs the paired comparisons based on the studentized maximum modulus test.

TUKEY

performs the paired comparisons based on Tukey's studentized range test. PROC LIFETEST uses the approximation described in Kramer (1956) and identifies the adjustment as "Tukey-Kramer" in the results. Note that ADJUST=TUKEY is incompatible with DIFF=CONTROL.

SIMULATE < (*simulate-options***) >**

computes the adjusted p-values from the simulated distribution of the maximum or maximum absolute value of a multivariate normal random vector. The simulation estimates q, the true $(1-\alpha)$th quantile, where α is the value of the ALPHA= *simulate-option*.

The number of samples for the SIMULATE adjustment is set so that the tail area for the simulated q is within a certain *accuracy radius* γ of $1-\alpha$ with an *accuracy confidence* of $100(1-\epsilon)\%$. In equation form,

$$\Pr(|F(\hat{q}) - (1-\alpha)| \leq \gamma) \quad = \quad 1-\epsilon$$

where $\hat{q}$ is the simulated q and F is the true distribution function of the maximum; see Edwards and Berry (1987) for details. By default, $\gamma = 0.005$ and $\epsilon = 0.01$ so that the tail area of $\hat{q}$ is within 0.005 of 0.95 with 99% confidence.

The *simulate-options* include the following:

ACC=*value*
specifies the target accuracy radius γ of a $100(1-\epsilon)\%$ confidence interval for the true probability content of the estimated $(1-\alpha)$th quantile. The default value is ACC=0.005.

ALPHA=*value*
specifies the value α for estimating the $(1-\alpha)$th quantile. The default value is the ALPHA= value in the PROC LIFETEST statement, or 0.05 if that option is not specified.

EPS=*value*
specifies the value ϵ for a $100(1-\epsilon)\%$ confidence interval for the true probability content of the estimated $(1-\alpha)$th quantile. The default value for the accuracy confidence is 99%, corresponding to EPS=0.01.

NSAMP=*n*
specifies the sample size for the simulation. By default, n is set based on the values of the target accuracy radius γ and accuracy confidence $100(1-\epsilon)\%$ for an interval for the true probability content of the estimated $(1-\alpha)$th quantile. With the default values for γ, ϵ, and α (0.005, 0.01, and 0.05, respectively), NSAMP=12604 by default.

REPORT
specifies that a report on the simulation should be displayed, including a listing of the parameters, such as γ, ϵ, and α, as well as an analysis of various methods for estimating or approximating the quantile.

SEED=*number*
specifies an integer used to start the pseudo-random number generator for the simulation. If you do not specify a seed, or if you specify a value less than or equal to zero, the seed is by default generated from reading the time of day from the computer's clock.

DIFF=ALL | CONTROL< (*'string' < . . . , 'string'* >) >
specifies which pairs of survival curves are considered for the multiple comparisons.

DIFF=ALL
requests all paired comparisons

DIFF=CONTROL <(*'string'* < *. . . 'string'* >**)>**
requests comparisons of the control curve with all other curves. To specify the control curve, you specify the quotes strings of formatted values that represent the curve in parentheses. For example, if Cell='large' identifies the control group, you specify

```
DIFF=CONTROL('large')
```

If more than one variable is used to identify the curves (for example, if Cell='large' and Sex='F' represent the control), you specify

```
DIFF=CONTROL('large' 'F')
```

The order of the quoted strings should correspond to the order of the stratum variables. If no specific curve is specified as the control, the first stratum or group value is used.

By default, DIFF=ALL unless you specify ADJUST=DUNNETT, in which case DIFF=CONTROL.

GROUP=*variable*
specifies the variable whose formatted values identify the various samples whose underlying survival curves are to be compared. The tests are stratified on the levels of the STRATA variables. For instance, in a multicenter trial in which two forms of therapy are to be compared, you specify the variable identifying therapies as the GROUP= variable and the variable identifying centers as the STRATA variable, in order to perform a stratified test to compare the therapies while controlling the effect of the centers.

MISSING
allows missing values to be a stratum level or a valid value of the GROUP= variable.

NODETAIL
suppresses the display of the rank statistics and the corresponding covariance matrices for various strata. If the TREND option is specified, the display of the scores for computing the trend tests is suppressed.

NOTEST
suppresses the k-sample tests, stratified tests, and trend tests.

TREND
computes the trend tests for testing the null hypothesis that the k population hazards rate are the same versus an ordered alternatives. If there is only one STRATA variable and the variable is numeric, the unformatted values of the variable are used as the scores; otherwise, the scores are $1, 2, \ldots,$ in the given order of the strata.

TEST=*test-request*

TEST=*(test-request* < *. . . test-request*> *)*
controls the tests produced. Each test corresponds to a different weight function (see the section "Nonparametric Tests" on page 3142 for the weight functions). The test requests include the following:

ALL — specifies all the nonparametric tests with $\rho 1=1$ and $\rho 2=0$ for the Fleming and Harrington test—FLEMING(1,0).

FLEMING(ρ1, ρ2)	specifies the family of tests in Harrington and Fleming (1982), where ρ1 and ρ2 are nonnegative numbers. FLEMING(ρ1,ρ2) reduces to the Fleming-Harrington G^{ρ} family (Fleming and Harrington 1981) when ρ2=0, which you can specify as FLEMING(ρ) with one argument. When ρ=0, the test becomes the log-rank test. When ρ=1, the test should be very close to the Peto-Peto test.
LOGRANK	specifies the log-rank test.
NONE	suppresses all comparison tests. Specifying TEST=NONE is equivalent to specify NOTEST.
LR	specifies the likelihood ratio test based on the exponential model.
MODPETO	specifies the modified Peto-Peto test.
PETO	specifies the Peto-Peto test. The test is also referred to as the Peto-Peto-Prentice test.
WILCOXON	specifies the Wilcoxon test. The test is also referred to as the Gehan test or the Breslow test.
TARONE	specifies the Tarone-Ware test.

By default, TEST=(LOGRANK WILCOXON LR) for the *k*-sample tests, and TEST=(LOGRANK WILCOXON) for stratified and trend tests.

TEST Statement

TEST *variables* **;**

The TEST statement specifies a list of numeric covariates (prognostic variables) that you want tested for association with the failure time.

Two sets of rank statistics are computed. These rank statistics and their variances are pooled over all strata. Univariate (marginal) test statistics are displayed for each of the covariates.

Additionally, a sequence of test statistics for joint effects of covariates is displayed. The first element of the sequence is the largest univariate test statistic. Other variables are then added on the basis of the largest increase in the joint test statistic. The process continues until all the variables have been added or until the remaining variables are linearly dependent on the previously added variables.

See the section "Rank Tests for the Association of Survival Time with Covariates" on page 3145 for more information.

TIME Statement

TIME *variable* < **censor(list)* > **;**

The TIME statement is required. It is used to indicate the failure time variable, where *variable* is the name of the failure time variable that can be optionally followed by an asterisk, the name of the censoring variable, and a parenthetical list of values that correspond to right censoring. The censoring values should be numeric, nonmissing values. For example, the statement

```
time T*Flag(1,2);
```

identifies the variable T as containing the observed failure times (event or censored). If the variable Flag has the value 1 or 2, the corresponding value of T is a right-censored value.

Details: LIFETEST Procedure

Missing Values

Observations with a missing value for either the failure time or the censoring variable are not used in the analysis. If a stratum variable value is missing, the observation is not used; however, the MISSING option can be used to request that missing values be treated as valid stratum values. If any variable specified in the TEST statement has a missing value, that observation is not used in the calculation of the rank statistics.

Computational Formulas

Product-Limit Method

Let $t_1 < t_2 < \cdots < t_D$ represent the distinct event times. For each $i = 1, \ldots, D$, let n_i be the number of surviving units, the size of the risk set, just prior to t_i. Let d_i be the number of units that fail at t_i, and let $s_i = n_i - d_i$.

The product-limit estimate of the SDF at t_i is the cumulative product

$$\hat{S}(t_i) = \prod_{j=1}^{i} \left(1 - \frac{d_j}{n_j}\right)$$

Notice that the estimator is defined to be right continuous; that is, the events at t_i are included in the estimate of $S(t_i)$. The corresponding estimate of the standard error is computed using Greenwood's

formula (Kalbfleisch and Prentice 1980) as

$$\hat{\sigma}\left(\hat{S}(t_i)\right) = \hat{S}(t_i)\sqrt{\sum_{j=1}^{i}\frac{d_j}{n_j s_j}}$$

The first quartile (or the 25th percentile) of the survival time is the time beyond which 75% of the subjects in the population under study are expected to survive. It is estimated by

$$q_{.25} = \min\{t_j | \hat{S}(t_j) < 0.75\}$$

If $\hat{S}(t)$ is exactly equal to 0.75 from t_j to t_{j+1}, the first quartile is taken to be $(t_j + t_{j+1})/2$. If it happens that $\hat{S}(t)$ is greater than 0.75 for all values of t, the first quartile cannot be estimated and is represented by a missing value in the printed output.

The general formula for estimating the $100p$th percentile point is

$$q_p = \min\{t_j | \hat{S}(t_j) < 1 - p\}$$

The second quartile (the median) and the third quartile of survival times correspond to p=0.5 and p=0.75, respectively.

Brookmeyer and Crowley (1982) have constructed the confidence interval for the median survival time based on the confidence interval for the $S(t)$. The methodology is generalized to construct the confidence interval for the $100p$th percentile based on a g-transformed confidence interval for $S(t)$ (Klein and Moeschberger 1997). You can use the CONFTYPE= option to specify the g-transformation. The $100(1-\alpha)\%$ confidence interval for the first quantile survival time is the set of all points t that satisfy

$$\left|\frac{g(\hat{S}(t)) - g(1-0.25)}{g'(\hat{S}(t))\hat{\sigma}(\hat{S}(t))}\right| \leq z_{1-\frac{\alpha}{2}}$$

where $g'(x)$ is the first derivative of $g(x)$ and $z_{1-\frac{\alpha}{2}}$ is the $100(1-\frac{\alpha}{2})$th percentile of the standard normal distribution.

Consider the bone marrow transplant data described in Example 49.2. The following table illustrates the construction of the confidence limits for the first quartile in the ALL group. Values of $\frac{g(\hat{S}(t))-g(1-0.25)}{g'(\hat{S}(t))\hat{\sigma}(\hat{S}(t))}$ that lie between $\pm z_{1-\frac{0.05}{2}} = \pm$ 1.965 are highlighted.

Constructing 95% Confidence Limits for the 25th Percentile							
			$\frac{g(\hat{S}(t))-g(1-0.25)}{g'(\hat{S}(t))\hat{\sigma}(\hat{S}(t))}$				
t	$\hat{S}(t)$	$\hat{\sigma}(\hat{S}(t))$	LINEAR	LOGLOG	LOG	ASINSQRT	LOGIT
1	0.97368	0.025967	8.6141	2.37831	9.7871	4.44648	2.47903
55	0.94737	0.036224	5.4486	2.36375	6.1098	3.60151	2.46635
74	0.92105	0.043744	3.9103	2.16833	4.3257	2.94398	2.25757
86	0.89474	0.049784	2.9073	**1.89961**	3.1713	2.38164	1.97023
104	0.86842	0.054836	2.1595	**1.59196**	2.3217	**1.87884**	**1.64297**
107	0.84211	0.059153	**1.5571**	**1.26050**	**1.6490**	**1.41733**	**1.29331**
109	0.81579	0.062886	**1.0462**	**0.91307**	**1.0908**	**0.98624**	**0.93069**
110	0.78947	0.066135	**0.5969**	**0.55415**	**0.6123**	**0.57846**	**0.56079**
122	0.73684	0.071434	**-0.1842**	**-0.18808**	**-0.1826**	**-0.18573**	**-0.18728**
129	0.71053	0.073570	**-0.5365**	**-0.56842**	**-0.5222**	**-0.54859**	**-0.56101**
172	0.68421	0.075405	**-0.8725**	**-0.95372**	**-0.8330**	**-0.90178**	**-0.93247**
192	0.65789	0.076960	**-1.1968**	**-1.34341**	**-1.1201**	**-1.24712**	**-1.30048**
194	0.63158	0.078252	**-1.5133**	**-1.73709**	**-1.3870**	**-1.58613**	**-1.66406**
230	0.60412	0.079522	**-1.8345**	-2.14672	**-1.6432**	**-1.92995**	-2.03291
276	0.57666	0.080509	-2.1531	-2.55898	**-1.8825**	-2.26871	-2.39408
332	0.54920	0.081223	-2.4722	-2.97389	-2.1070	-2.60380	-2.74691
383	0.52174	0.081672	-2.7948	-3.39146	-2.3183	-2.93646	-3.09068
418	0.49428	0.081860	-3.1239	-3.81166	-2.5177	-3.26782	-3.42460
466	0.46682	0.081788	-3.4624	-4.23445	-2.7062	-3.59898	-3.74781
487	0.43936	0.081457	-3.8136	-4.65971	-2.8844	-3.93103	-4.05931
526	0.41190	0.080862	-4.1812	-5.08726	-3.0527	-4.26507	-4.35795
609	0.38248	0.080260	-4.5791	-5.52446	-3.2091	-4.60719	-4.64271
662	0.35306	0.079296	-5.0059	-5.96222	-3.3546	-4.95358	-4.90900

Consider the LINEAR transformation where $g(x) = x$. The event times that satisfy $\left|\frac{g(\hat{S}(t))-g(1-p)}{g'(\hat{S}(t))\sqrt{\hat{V}(\hat{S}(t))}}\right| \leq 1.9599$ include 107, 109, 110, 122, 129, 172, 192, 194, and 230. The confidence of the interval [107, 230] is less than 95%. Brookmeyer and Crowley (1982) suggest extending the confidence interval to but not including the next event time. As such the 95% confidence interval for the first quartile based on the linear transform is [107, 276). The following table lists the confidence intervals for the various transforms.

95% CI's for the 25th Percentile		
CONFTYPE	[Lower	Upper)
LINEAR	107	276
LOGLOG	86	230
LOG	107	332
ASINSQRT	104	276
LOGIT	104	230

Sometimes, the confidence limits for the quartiles cannot be estimated. For convenience of explanation, consider the linear transform $g(x) = x$. If the curve representing the upper confidence limits for the survivor function lies above 0.75, the upper confidence limit for first quartile cannot be es-

timated. On the other hand, if the curve representing the lower confidence limits for the survivor function lies above 0.75, the lower confidence limit for the quartile cannot be estimated.

The estimated mean survival time is

$$\hat{\mu} = \sum_{i=1}^{D} \hat{S}(t_{i-1})(t_i - t_{i-1})$$

where t_0 is defined to be zero. When the largest observed time is censored, this sum underestimates the mean. The standard error of $\hat{\mu}$ is estimated as

$$\hat{\sigma}(\hat{\mu}) = \sqrt{\frac{m}{m-1} \sum_{i=1}^{D-1} \frac{A_i^2}{n_i s_i}}$$

where

$$A_i = \sum_{j=i}^{D-1} \hat{S}(t_j)(t_{j+1} - t_j)$$

$$m = \sum_{j=1}^{D} d_j$$

If the largest observed time is not an event, you can use the TIMELIM= option to specify a time limit L and estimate the mean survival time limited to the time L and its standard error by replacing k by $k+1$ with $t_{k+1} = L$.

Nelson-Aalen Estimate of the Cumulative Hazard Function

The Nelson-Aalen cumulative hazard estimator, defined up to the largest observed time on study, is

$$\tilde{H}(t) = \sum_{t_i \le t} \frac{d_i}{n_i}$$

and its estimated variance is

$$\hat{V}(\tilde{H}(t)) = \sum_{t_i \le t} \frac{d_i}{n_i^2}$$

Life-Table Method

The life-table estimates are computed by counting the numbers of censored and uncensored observations that fall into each of the time intervals $[t_{i-1}, t_i)$, $i = 1, 2, \ldots, k+1$, where $t_0 = 0$ and $t_{k+1} = \infty$. Let n_i be the number of units entering the interval $[t_{i-1}, t_i)$, and let d_i be the number of events occurring in the interval. Let $b_i = t_i - t_{i-1}$, and let $n_i' = n_i - w_i/2$, where w_i is the number of units censored in the interval. The *effective sample size* of the interval $[t_{i-1}, t_i)$ is denoted by n_i'. Let t_{mi} denote the midpoint of $[t_{i-1}, t_i)$.

The conditional probability of an event in $[t_{i-1}, t_i)$ is estimated by

$$\hat{q}_i = \frac{d_i}{n'_i}$$

and its estimated standard error is

$$\hat{\sigma}(\hat{q}_i) = \sqrt{\frac{\hat{q}_i \hat{p}_i}{n'_i}}$$

where $\hat{p}_i = 1 - \hat{q}_i$.

The estimate of the survival function at t_i is

$$\hat{S}(t_i) = \begin{cases} 1 & i = 0 \\ \hat{S}(t_{i-1}) p_{i-1} & i > 0 \end{cases}$$

and its estimated standard error is

$$\hat{\sigma}\left(\hat{S}(t_i)\right) = \hat{S}(t_i) \sqrt{\sum_{j=1}^{i-1} \frac{\hat{q}_j}{n'_j \hat{p}_j}}$$

The density function at t_{mi} is estimated by

$$\hat{f}(t_{mi}) = \frac{\hat{S}(t_i)\hat{q}_i}{b_i}$$

and its estimated standard error is

$$\hat{\sigma}\left(\hat{f}(t_{mi})\right) = \hat{f}(t_{mi}) \sqrt{\sum_{j=1}^{i-1} \frac{\hat{q}_j}{n'_j \hat{p}_j} + \frac{\hat{p}_i}{n'_i \hat{q}_i}}$$

The estimated hazard function at t_{mi} is

$$\hat{h}(t_{mi}) = \frac{2\hat{q}_i}{b_i(1 + \hat{p}_i)}$$

and its estimated standard error is

$$\hat{\sigma}\left(\hat{h}(t_{mi})\right) = \hat{h}(t_{mi}) \sqrt{\frac{1 - (b_i \hat{h}(t_{mi})/2)^2}{n'_i \hat{q}_i}}$$

Let $[t_{j-1}, t_j)$ be the interval in which $\hat{S}(t_{j-1}) \geq \hat{S}(t_i)/2 > \hat{S}(t_j)$. The median residual lifetime at t_i is estimated by

$$\hat{M}_i = t_{j-1} - t_i + b_j \frac{\hat{S}(t_{j-1}) - \hat{S}(t_i)/2}{\hat{S}(t_{j-1}) - \hat{S}(t_j)}$$

and the corresponding standard error is estimated by

$$\hat{\sigma}(\hat{M}_i) = \frac{\hat{S}(t_i)}{2\hat{f}(t_{mj})\sqrt{n'_i}}$$

Interval Determination

If you want to determine the intervals exactly, use the INTERVALS= option in the PROC LIFETEST statement to specify the interval endpoints. Use the WIDTH= option to specify the width of the intervals, thus indirectly determining the number of intervals. If neither the INTERVALS= option nor the WIDTH= option is specified in the life-table estimation, the number of intervals is determined by the NINTERVAL= option. The width of the time intervals is 2, 5, or 10 times an integer (possibly a negative integer) power of 10. Let $c = \log_{10}(\text{maximum observed time/number of intervals})$, and let b be the largest integer not exceeding c. Let $d = 10^{c-b}$ and let

$$a = 2 \times I(d \leq 2) + 5 \times I(2 < d \leq 5) + 10 \times I(d > 5)$$

with I being the indicator function. The width is then given by

$$\text{width} = a \times 10^b$$

By default, NINTERVAL=10.

Pointwise Confidence Limits in the OUTSURV= Data Set

Pointwise confidence limits are computed for the survivor function, and for the density function and hazard function when the life-table method is used. Let α be specified by the ALPHA= option. Let $z_{\alpha/2}$ be the critical value for the standard normal distribution. That is, $\Phi(-z_{\alpha/2}) = \alpha/2$, where Φ is the cumulative distribution function of the standard normal random variable.

Survivor Function

When the computation of confidence limits for the survivor function $S(t)$ is based on the asymptotic normality of the survival estimator $\hat{S}(t)$, the approximate confidence interval might include impossible values outside the range [0,1] at extreme values of t. This problem can be avoided by applying the asymptotic normality to a transformation of $S(t)$ for which the range is unrestricted. In addition, certain transformed confidence intervals for $S(t)$ perform better than the usual linear confidence intervals (Borgan and Liestøl 1990). The CONFTYPE= option enables you to pick one of the following transformations: the log-log function (Kalbfleisch and Prentice 1980), the arcsine-square root function (Nair 1984), the logit function (Meeker and Escobar 1998), the log function, and the linear function.

Let g be the transformation that is being applied to the survivor function $S(t)$. By the delta method, the standard error of $g(\hat{S}(t))$ is estimated by

$$\tau(t) = \hat{\sigma}(g(\hat{S}(t)) = g'(\hat{S}(t))\hat{\sigma}(\hat{S}(t))$$

where g' is the first derivative of the function g. The 100(1–α)% confidence interval for $S(t)$ is given by

$$g^{-1}\left(g(\hat{S}(t)) \pm z_{\frac{\alpha}{2}} g'(\hat{S}(t))\hat{\sigma}(\hat{S}(t))\right)$$

where g^{-1} is the inverse function of g.

Arcsine-Square Root Transformation The estimated variance of $\sin^{-1}(\sqrt{\hat{S}(t)})$ is $\hat{\tau}^2(t) = \frac{\hat{\sigma}^2[\hat{S}(t)]}{4\hat{S}(t)[1-\hat{S}(t)]}$. The 100(1–$\alpha$)% confidence interval for $S(t)$ is given by

$$\sin^2\{\max[0, \sin^{-1}(\sqrt{\hat{S}(t)}) - z_{\frac{\alpha}{2}}\hat{\tau}(t)]\} \leq S(t) \leq \sin^2\{\min[\frac{\pi}{2}, \sin^{-1}(\sqrt{\hat{S}(t)}) + z_{\frac{\alpha}{2}}\hat{\tau}(t)]\}$$

Linear Transformation This is the same as having no transformation in which g is the identity. The 100(1–α)% confidence interval for $S(t)$ is given by

$$\hat{S}(t) - z_{\frac{\alpha}{2}}\hat{\sigma}(S(t))) \leq S(t) \leq \hat{S}(t) + z_{\frac{\alpha}{2}}\hat{\sigma}(S(t)))$$

Log Transformation The estimated variance of $\log(\hat{S}(t))$ is $\hat{\tau}^2(t) = \frac{\hat{\sigma}^2(\hat{S}(t))}{\hat{S}^2(t)}$. The 100(1–$\alpha$)% confidence interval for $S(t)$ is given by

$$\hat{S}(t)\exp(-z_{\frac{\alpha}{2}}\hat{\tau}(t)) \leq S(t) \leq \hat{S}(t)\exp(z_{\frac{\alpha}{2}}\hat{\tau}(t))$$

Log-Log Transformation The estimated variance of $\log(-\log(\hat{S}(t))$ is $\hat{\tau}^2(t) = \frac{\hat{\sigma}^2[\hat{S}(t)]}{[\hat{S}(t)\log(\hat{S}(t))]^2}$. The 100(1–$\alpha$)% confidence interval for $S(t)$ is given by

$$[\hat{S}(t)]^{\exp(z_{\frac{\alpha}{2}}\hat{\tau}(t))} \leq S(t) \leq [\hat{S}(t)]^{\exp(-z_{\frac{\alpha}{2}}\hat{\tau}(t))}$$

Logit Transformation The estimated variance of $\log(\frac{\hat{S}(t)}{1-\hat{S}(t)})$ is $\hat{\tau}^2(t) = \frac{\hat{\sigma}^2(\hat{S}(t))}{\hat{S}^2(t)[1-\hat{S}(t)]^2}$. The 100(1–$\alpha$)% confidence limits for $S(t)$ are given by

$$\frac{\hat{S}(t)}{\hat{S}(t) + (1-\hat{S}(t))\exp(z_{\frac{\alpha}{2}}\hat{\tau}(t))} \leq S(t) \leq \frac{\hat{S}(t)}{\hat{S}(t) + (1-\hat{S}(t))\exp(-z_{\frac{\alpha}{2}}\hat{\tau}(t))}$$

Density and Hazard Functions

For the life-table method, a 100(1–α)% confidence interval for hazard function or density function at time t is computed as

$$\hat{g}(t) \pm z_{\alpha/2}\hat{\sigma}[\hat{g}(t)]$$

where $\hat{g}(t)$ is the estimate of either the hazard function or the density function at time t, and $\hat{\sigma}[\hat{g}(t)]$ is the corresponding standard error estimate.

Simultaneous Confidence Intervals for Kaplan-Meier Curves

The pointwise confidence interval for the survivor function $S(t)$ is valid for a single fixed time at which the inference is to be made. In some applications, it is of interest to find the upper and lower confidence bands that guarantee, with a given confidence level, that the survivor function falls within the band for all t in some interval. Hall and Wellner (1980) and Nair (1984) provide two different approaches for deriving the confidence bands. An excellent review can be found in Klein and

Moeschberger (1997). You can use the CONFBAND= option in the SURVIVAL statement to select the confidence bands. The EP confidence band provides confidence bounds that are proportional to the pointwise confidence interval, while those of the HW band are not proportional to the pointwise confidence bounds. The maximum time, t_U, for the bands can be specified by the BANDMAX= option; the minimum time, t_L, can be specified by the BANDMIN= option. Transformations that are used to improve the pointwise confidence intervals can be applied to improve the confidence bands. It might turn out that the upper and lower bounds of the confidence bands are not decreasing in $t_L < t < t_U$, which is contrary to the nonincreasing characteristic of survivor function. Meeker and Escobar (1998) suggest making an adjustment so that the bounds do not increase: if the upper bound is increasing on the right, it is made flat from the minimum to t_U; if the lower bound is increasing from the right, it is made flat from t_L to the maximum. PROC LIFETEST does not make any adjustment for the nondecreasing behavior of the confidence bands in the OUTSURV= data set. However, the adjustment was made in the display of the confidence bands by using ODS Graphics.

For Kaplan-Meier estimation, let $t_1 < t_2 < \ldots < t_D$ be the D distinct events times, and at time t_i, there are d_i events. Let Y_i be the number of individuals who are at risk at time t_i. The variance of $\hat{S}(t)$, given by the Greenwood formula, is $\hat{\sigma}^2[\hat{S}(t)] = \sigma_S^2(t)\hat{S}^2(t)$, where

$$\sigma_S^2(t) = \sum_{t_i \le t} \frac{d_i}{Y_i(Y_i - d_i)}$$

Let $t_L < t_U$ be the time range for the confidence band so that t_U is less than or equal to the largest event time. For the Hall-Wellner band, t_L can be zero, but for the equal-precision band, t_L is greater than or equal to the smallest event time. Let

$$a_L = \frac{n\sigma_S^2(t_L)}{1 + n\sigma_S^2(t_L)} \quad \text{and} \quad a_U = \frac{n\sigma_S^2(t_U)}{1 + n\sigma_S^2(t_U)}$$

Let $\{W^0(u), 0 \le u \le 1\}$ be a Brownian bridge.

Hall-Wellner Band

The 100(1–α)% HW band of Hall and Wellner (1980) is

$$\hat{S}(t) - h_\alpha(a_L, a_U)n^{-\frac{1}{2}}[1 + n\sigma_S^2(t)]\hat{S}(t) \le S(t) \le \hat{S}(t) + h_\alpha(a_L, a_U)n^{-\frac{1}{2}}[1 + n\sigma_S^2(t)]\hat{S}(t)$$

for all $t_L \le t \le t_U$, where $h_\alpha(a_L, a_U)$ is given by

$$\alpha = \Pr\{\sup_{a_L \le u \le a_U} |W^0(u)| > h_\alpha(a_L, a_U)\}$$

The critical values are computed from the results in Chung (1986).

Note that the given confidence band has a formula similar to that of the (linear) pointwise confidence interval, where $h_\alpha(a_L, a_U)$ and $n^{-\frac{1}{2}}[1 + n\sigma_S^2(t)]\hat{S}(t)$ in the former correspond to $z_{\frac{\alpha}{2}}$ and $\hat{\sigma}(\hat{S}(t))$ in the latter, respectively. You can obtain the other transformations (arcsine-square root, log-log, log, and logit) for the confidence bands by replacing $z_{\frac{\alpha}{2}}$ and $\hat{\tau}(t)$ in the corresponding pointwise confidence interval formula by $h_\alpha(a_L, a_U)$ and the following $\hat{\tau}(t)$, respectively.

Arcsine-Square Root Transformation

$$\hat{\tau}(t) = \frac{1 + n\sigma_S^2(t)}{2}\sqrt{\frac{S(t)}{n[1 - S(t)]}}$$

Log Transformation

$$\hat{\tau}(t) = \frac{1 + n\sigma_S^2(t)}{\sqrt{n}}$$

Log-Log Transformation

$$\hat{\tau}(t) = \frac{1 + n\sigma_S^2(t)}{\sqrt{n}|\log[\hat{S}(t)]|}$$

Logit Transformation

$$\hat{\tau}(t) = \frac{1 + n\sigma_S^2(t)}{\sqrt{n}[1 - \hat{S}(t)]}$$

Equal-Precision Band

The 100(1–α)% EP band of Nair (1984) is

$$\hat{S}(t) - e_\alpha(a_L, a_U)\hat{S}(t)\sigma_S(t) \le S(t) \le \hat{S}(t) + e_\alpha(a_L, a_U)\hat{S}(t)\sigma_S(t)$$

for all $t_L \le t \le t_U$, where $e_\alpha(a_L, a_U)$ is given by

$$\alpha = \Pr\{\sup_{a_L \le u \le a_U} \frac{|W^0(u)|}{[u(1-u)]^{\frac{1}{2}}} > e_\alpha(a_L, a_U)\}$$

PROC LIFETEST uses the approximation of Miller and Siegmund (1982, Equation 8) to approximate the tail probability in which $e_\alpha(a_L, a_U)$ is obtained by solving x in

$$\frac{4x\phi(x)}{x} + \phi(x)\left(x - \frac{1}{x}\right)\log\left[\frac{a_U(1-a_L)}{a_L(1-a_U)}\right] = \alpha$$

where $\phi()$ is the standard normal density. Note that the confidence bounds given are proportional to the pointwise confidence intervals. As a matter of fact, this confidence band and the (linear) pointwise confidence interval have the same formula except for the critical values ($z_{\frac{\alpha}{2}}$ for the pointwise confidence interval and $e_\alpha(a_L, a_U)$ for the band). You can obtain the other transformations (arcsine-square root, log-log, log, and logit) for the confidence bands by replacing $z_{\frac{\alpha}{2}}$ by $e_\alpha(a_L, a_U)$ in the formula of the pointwise confidence intervals.

Kernel-Smoothed Hazard Estimate

Kernel-smoothed estimators of the hazard function $h(t)$ are based on the Nelson-Aalen estimator $\tilde{H}(t)$ and its variance $\hat{V}(\tilde{H}(t))$. Consider the jumps of $\tilde{H}(t)$ and $\hat{V}(\tilde{H}(t))$ at the event times $t_1 < t_2 < \ldots < t_D$ as follows:

$$\begin{aligned} \Delta\tilde{H}(t_i) &= \tilde{H}(t_i) - \tilde{H}(t_{i-1}) \\ \hat{V}(\tilde{H}(t_i)) &= \hat{V}(\tilde{H}(t_i)) - \hat{V}(\tilde{H}(t_{i-1})) \end{aligned}$$

where t_0=0.

The kernel-smoothed estimator of $h(t)$ is a weighted average of $\Delta\tilde{H}(t)$ over event times that are within a bandwidth distance b of t. The weights are controlled by the choice of kernel function, $K()$, defined on the interval [–1,1]. The choices of $K()$ are as follows:

uniform kernel

$$K_U(x) = \frac{1}{2}, \quad -1 \le x \le 1$$

Epanechnikov kernel

$$K_E(x) = \frac{3}{4}(1 - x^2), \quad -1 \le x \le 1$$

biweight kernel

$$K_{BW}(x) = \frac{15}{16}(1 - x^2)^2, \quad -1 \le x \le 1$$

The kernel-smoothed hazard rate estimator is defined for all time points on $(0, t_D)$. For time points t for which $b \le t \le t_D - b$, the kernel-smoothed estimated of $h(t)$ based on the kernel $K()$ is given by

$$\hat{h}(t) = \frac{1}{b}\sum_{i=1}^{D} K\left(\frac{t - t_i}{b}\right)\Delta\tilde{H}(t_i)$$

The variance of $\hat{h}(t)$ is estimated by

$$\hat{\sigma^2}(\hat{h}(t)) = \frac{1}{b^2}\sum_{i=1}^{D} K\left(\frac{t - t_i}{b}\right)^2 \Delta\hat{V}(\tilde{H}(t_i))$$

For $t < b$, the symmetric kernels $K()$ are replaced by the corresponding asymmetric kernels of Gasser and Müller (1979). Let $q = \frac{t}{b}$. The modified kernels are as follows:

uniform kernel

$$K_{U,q}(x) = \frac{4(1 + q^3)}{(1 + q)^4} + \frac{6(1 - q)}{(1 + q)^3}x, \quad -1 \le x \le q$$

Epanechnikov kernel

$$K_{E,q}(x) = K_E(x)\frac{64(2-4q+6q^2-3q^3)+240(1-q)^2x}{(1+q)^4(19-18q+3q^2)}, \quad -1 \le x \le q$$

biweight kernel

$$K_{BW}(x) = K_{BW}\frac{64(8-24q+48q^2-45q^3+15q^4)+1120(1-q)^3x}{(1+q)^5(81-168q+126q^2-40q^3+5q^4)}, \quad -1 \le x \le q$$

For $t_D - b \le t \le t_D$, let $q = \frac{t_D - t}{b}$. The asymmetric kernels for $t < b$ are used with x replaced by $-x$.

Using the log transform on the smoothed hazard rate, the 100(1–α)% pointwise confidence interval for the smoothed hazard rate $h(t)$ is given by

$$\hat{h}(t) = \hat{h}(t)\exp\left[\pm\frac{z_{1-\alpha/2}\hat{\sigma}(\hat{h}(t))}{\hat{h}(t)}\right]$$

where $z_{1-\frac{\alpha}{2}}$ is the 100(1–$\frac{\alpha}{2}$)th percentile of the standard normal distribution.

Optimal Bandwidth

The following mean integrated squared error (MISE) over the range τ_L and τ_U is used as a measure of the global performance of the kernel function estimator

$$\begin{aligned} MISE(b) &= E\int_{\tau_L}^{\tau_U}(\hat{h}(i)-h(u))^2du \\ &= E\int_{\tau_L}^{\tau_U}\hat{h}^2(u)du - 2E\int_{\tau_L}^{\tau_U}\hat{h}(u)h(u)du + E\int_{\tau_L}^{\tau_U}h^2(u)du \end{aligned}$$

The last term is independent of the choice of the kernel and bandwidth and can be ignored when you are looking for the best value of b. The first integral can be approximated by using the trapezoid rule by evaluating $\hat{h}(t)$ at a grid of points $\tau_L = u_1 < \ldots < u_M = \tau_U$. You can specify τ_L, τ_R, and M by using the options MISEMIN=, MISEMAX=, and MISENUM=, respectively, of the HAZARD plot. The second integral can be estimated by Ramlau-Hansen (1983a, b) cross-validation estimate

$$\frac{1}{b}\sum_{i\neq j}K\left(\frac{t_i-t_j}{b}\right)\Delta\hat{H}(t_i)\Delta\hat{H}(t_j)$$

Therefore, for a fixed kernel, the optimal bandwidth is the quantity b that minimizes

$$g(b) = \sum_{i=1}^{M-1}\left[\frac{u_{i+1}-u_k}{2}\left(\hat{h}^2(u_i)+\hat{h}^2(u_{i+1})\right)\right] - \frac{2}{b}\sum_{i\neq j}K\left(\frac{t_i-t_j}{b}\right)\Delta\hat{H}(t_i)\Delta\hat{H}(t_j)$$

The minimization is carried out by the golden section search algorithm.

Comparison of Two or More Groups of Survival Data

Let k be the number of groups. Let $S_i(t)$ be the underlying survivor function ith group, $i = 1, \ldots, k$. The null and alternative hypotheses to be tested are

$$H_0 : S_1(t) = S_2(t) = \ldots = S_k(t) \text{ for all } t \leq \tau$$

versus

$$H_1 : \text{at least one of the } S_i(t)\text{'s is different for some } t \leq \tau$$

respectively, where τ is the largest observed time.

Likelihood Ratio Test

The likelihood ratio test statistic (Lawless 1982) for test H_0 versus H_1 assumes that the data in the various samples are exponentially distributed and tests that the scale parameters are equal. The test statistic is computed as

$$\chi^2 = 2N \log\left(\frac{T}{N}\right) - 2\sum_{j=1}^{k} N_j \log\left(\frac{T_j}{N_j}\right)$$

where N_j is the total number of events in the jth stratum, $N = \sum_{j=1}^{k} N_j$, T_j is the total time on test in the jth stratum, and $T = \sum_{j=1}^{k} T_j$. The approximate probability value is computed by treating χ^2 as having a chi-square distribution with $k-1$ degrees of freedom.

Nonparametric Tests

Let $t_1 < t_2 < \ldots < t_D$ be the distinct event times in the pooled sample. At time t_i, let $W(t_i)$ be a positive weight function, and let n_{ij} and d_{ij} be the size of the risk set and the number of events in the jth sample, respectively. Let $n_i = \sum_{j=1}^{k} n_{ij}$, $d_i = \sum_{j=1}^{k} d_{ij}$, and $s_i = n_i - d_i$.

The choices of the weight function $W(t_i)$ are given in Table 49.3.

Table 49.3 Weight Functions for Various Tests

Test	$W(t_i)$
log-rank	1.0
Wilcoxon	n_i
Tarone-Ware	$\sqrt{n_i}$
Peto-Peto	$\tilde{S}(t_i)$
modified Peto-Peto	$\tilde{S}(t_i)\frac{n_i}{n_i+1}$
Harrington-Fleming (p,q)	$[\hat{S}(t_i)]^p[1-\hat{S}(t_i)]^q,\ p \geq 0, q \geq 0$

where $\hat{S}(t)$ is the product-limit estimate at t for the pooled sample, and $\tilde{S}(t)$ is a survivor function

estimate close to $\hat{S}(t)$ given by

$$\tilde{S}(t) = \prod_{t_i \le t} \left(1 - \frac{d_i}{n_i + 1}\right)$$

Unstratified Tests The rank statistics (Klein and Moeschberger 1997, Section 7.3) for testing H_0 versus H_1 have the form of a k-vector $\mathbf{v} = (v_1, v_2, \ldots, v_k)'$ with

$$v_j = \sum_{i=1}^{D} W(t_i) \left\{ d_{ij} - \frac{n_{ij} d_i}{n_i} \right\}$$

and the estimated covariance matrix, $\mathbf{V} = (V_{jl})$, is given by

$$V_{jl} = \sum_{i=1}^{D} W^2(t_i) \left\{ \frac{d_i s_i (n_i n_{il} \delta_{jl} - n_{ij} n_{il})}{n_i^2 (n_i - 1)} \right\}$$

where δ_{jl} is 1 if $j = l$ and 0 otherwise. The term v_j can be interpreted as a weighted sum of observed minus expected numbers of failure under the null hypothesis of identical survival curves. The overall test statistic for homogeneity is $\mathbf{v}'\mathbf{V}^{-}\mathbf{v}$, where $\mathbf{V}^{-}$ denotes a generalized inverse of $\mathbf{V}$. This statistic is treated as having a chi-square distribution with degrees of freedom equal to the rank of $\mathbf{V}$ for the purposes of computing an approximate probability level.

Stratified Tests Suppose the test is to be stratified on M levels of a set of STRATA variables. Based only on the data of the sth stratum ($s = 1 \ldots M$), let $\mathbf{v}_s$ be the test statistic (Klein and Moeschberger 1997, Section 7.5) for the sth stratum, and let $\mathbf{V}_s$ be its covariance matrix. Let

$$\begin{aligned} \mathbf{v} &= \sum_{i=1}^{M} \mathbf{v}_s \\ \mathbf{V} &= \sum_{i=1}^{M} \mathbf{V}_s \end{aligned}$$

A global test statistic is constructed as

$$\chi^2 = \mathbf{v}'\mathbf{V}\mathbf{v}$$

Under the null hypothesis, the test statistic has a χ^2 distribution with the same degrees of freedom as the individual test for each stratum.

Multiple-Comparison Adjustments Let χ^2_r denote a chi-squared random variable with r degrees of freedom. Denote $\phi()$ and $\Phi()$ as the density function and the distribution function of a standard normal distribution, respectively. Let m be the number of comparisons; that is,

$$m = \begin{cases} \frac{k(k-1)}{2} & \text{DIFF} = \text{ALL} \\ k - 1 & \text{DIFF} = \text{CONTROL} \end{cases}$$

For a two-sided test comparing the survival of the jth group with that of lth group, $1 \le j \ne l \le r$, the test statistic is

$$z_{jl}^2 = \frac{(v_j - v_l)^2}{V_{jj} + V_{ll} - 2V_{jl}}$$

and the raw p-value is

$$p = \Pr(\chi_1^2 > z_{jl}^2)$$

Adjusted p-values for various multiple-comparison adjustments are computed as follows:

Bonferroni Adjustment

$$p = \min\{1, m\Pr(\chi_1^2 > z_{jl}^2)\}$$

Dunnett-Hsu Adjustment

With the first group being the control, let $\mathbf{C} = (c_{ij})$ be the $(r-1) \times r$ matrix of contrasts; that is,

$$c_{ij} = \begin{cases} 1 & i = 1, \ldots, r-1, j = 2, \ldots, r \\ -1 & j = i+1, i = 2, \ldots, r \\ 0 & \text{otherwise} \end{cases}$$

Let $\mathbf{\Sigma} \equiv (\sigma_{ij})$ and $\mathbf{R} \equiv (r_{ij})$ be covariance and correlation matrices of $\mathbf{Cv}$, respectively; that is,

$$\mathbf{\Sigma} = \mathbf{CVC}'$$

and

$$r_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}}$$

The factor-analytic covariance approximation of Hsu (1992) is to find $\lambda_1, \ldots, \lambda_{r-1}$ such that

$$\mathbf{R} = \mathbf{D} + \boldsymbol{\lambda}\boldsymbol{\lambda}'$$

where $\mathbf{D}$ is a diagonal matrix with the jth diagonal element being $1 - \lambda_j$ and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_{r-1})'$. The adjusted p-value is

$$p = 1 - \int_{-\infty}^{\infty} \phi(y) \prod_{i=1}^{r-1} \left[\Phi\left(\frac{\lambda_i y + z_{jl}}{\sqrt{1-\lambda_i^2}} \right) - \Phi\left(\frac{\lambda_i y - z_{jl}}{\sqrt{1-\lambda_i^2}} \right) \right] dy$$

which can be obtained in a DATA step as 1 – **PROBMC**("DUNNETT2", z_{ij},.,.,$r-1, \lambda_1, \ldots, \lambda_{r-1}$).

Scheffé Adjustment

$$p = \Pr(\chi_{r-1}^2 > z_{jl}^2)$$

Šidák Adjustment

$$p = 1 - \{1 - \Pr(\chi_1^2 > z_{jl}^2)\}^m$$

SMM Adjustment

$$p = 1 - [2\Phi(z_{jl}) - 1]^m$$

which can also be evaluated in a DATA step as 1 – **PROBMC**("MAXMOD",z_{jl}, ., ., m).

Tukey Adjustment

$$p = 1 - \int_{-\infty}^{\infty} r\phi(y)[\Phi(y) - \Phi(y - \sqrt{2}z_{jl})]^{r-1} dy$$

which can also be evaluated in a DATA step as 1 – **PROBMC**("RANGE",$\sqrt{2}z_{jl}$, ., ., r).

Trend Tests Trend tests (Klein and Moeschberger 1997, Section 7.4) have more power to detect ordered alternatives as

$$H_2 : S_1(t) \geq S_2(t) \geq \ldots \geq S_k(t), t \leq \tau, \text{ with at least one inequality}$$

or

$$H_2 : S_1(t) \leq S_2(t) \leq \ldots \leq S_k(t), t \leq \tau, \text{ with at least one inequality}$$

Let $a_1 < a_2 < \ldots < a_k$ be a sequence of scores associated with the k samples. The test statistic and its standard error are given by $\sum_{j=1}^{k} a_j v_j$ and $\sum_{j=1}^{k} \sum_{l=1}^{k} a_j a_l V_{jl}$, respectively. Under H_0, the z-score

$$Z = \frac{\sum_{j=1}^{k} a_j v_j}{\sqrt{\{\sum_{j=1}^{k} \sum_{l=1}^{k} a_j a_l V_{jl}\}}}$$

has, asymptotically, a standard normal distribution. PROC LIFETEST provides both one-tail and two-tail p-values for the test.

Rank Tests for the Association of Survival Time with Covariates

The rank tests for the association of covariates (Kalbfleisch and Prentice 1980, Chapter 6) are more general cases of the rank tests for homogeneity. In this section, the index α is used to label all observations, $\alpha = 1, 2, \ldots, n$, and the indices i, j range only over the observations that correspond to events, $i, j = 1, 2, \ldots, k$. The ordered event times are denoted as $t_{(i)}$, the corresponding vectors of covariates are denoted as $\mathbf{z}_{(i)}$, and the ordered times, both censored and event times, are denoted as t_α.

The rank test statistics have the form

$$\mathbf{v} = \sum_{\alpha=1}^{n} c_{\alpha,\delta_\alpha} \mathbf{z}_\alpha$$

where n is the total number of observations, c_{α,δ_α} are rank scores, which can be either log-rank or Wilcoxon rank scores, δ_α is 1 if the observation is an event and 0 if the observation is censored, and $\mathbf{z}_\alpha$ is the vector of covariates in the TEST statement for the αth observation. Notice that the scores, c_{α,δ_α}, depend on the censoring pattern and that the terms are summed up over all observations.

The log-rank scores are

$$c_{\alpha,\delta_\alpha} = \sum_{(j:t_{(j)} \le t_\alpha)} \left(\frac{1}{n_j} - \delta_\alpha \right)$$

and the Wilcoxon scores are

$$c_{\alpha,\delta_\alpha} = 1 - (1 + \delta_\alpha) \prod_{(j:t_{(j)} \le t_\alpha)} \frac{n_j}{n_j + 1}$$

where n_j is the number at risk just prior to $t_{(j)}$.

The estimates used for the covariance matrix of the log-rank statistics are

$$\mathbf{V} = \sum_{i=1}^{k} \frac{\mathbf{V}_i}{n_i}$$

where $\mathbf{V}_i$ is the corrected sum of squares and crossproducts matrix for the risk set at time $t_{(i)}$; that is,

$$\mathbf{V}_i = \sum_{(\alpha:t_\alpha \ge t_{(i)})} (\mathbf{z}_\alpha - \bar{\mathbf{z}}_i)'(\mathbf{z}_\alpha - \bar{\mathbf{z}}_i)$$

where

$$\bar{\mathbf{z}}_i = \sum_{(\alpha:t_\alpha \ge t_{(i)})} \frac{\mathbf{z}_\alpha}{n_i}$$

The estimate used for the covariance matrix of the Wilcoxon statistics is

$$\mathbf{V} = \sum_{i=1}^{k} \left[a_i(1 - a_i^*)(2\mathbf{z}_{(i)}\mathbf{z}'_{(i)} + \mathbf{S}_i) - (a_i^* - a_i) \left(a_i \mathbf{x}_i \mathbf{x}_i' + \sum_{j=i+1}^{k} a_j (\mathbf{x}_i \mathbf{x}'_j + \mathbf{x}_j \mathbf{x}_i') \right) \right]$$

where

$$\begin{aligned}
a_i &= \prod_{j=1}^{i} \frac{n_j}{n_j + 1} \\
a_i^* &= \prod_{j=1}^{i} \frac{n_j + 1}{n_j + 2} \\
\mathbf{S}_i &= \sum_{(\alpha:t_{(i+1)} > t_\alpha > t_{(i)})} \mathbf{z}_\alpha \mathbf{z}'_\alpha \\
\mathbf{x}_i &= 2\mathbf{z}_{(i)} + \sum_{(\alpha:t_{(i+1)} > t_\alpha > t_{(i)})} \mathbf{z}_\alpha
\end{aligned}$$

In the case of tied failure times, the statistics $\mathbf{v}$ are averaged over the possible orderings of the tied failure times. The covariance matrices are also averaged over the tied failure times. Averaging the covariance matrices over the tied orderings produces functions with appropriate symmetries for the tied observations; however, the actual variances of the $\mathbf{v}$ statistics would be smaller than the preceding estimates. Unless the proportion of ties is large, it is unlikely that this will be a problem.

The univariate tests for each covariate are formed from each component of $\mathbf{v}$ and the corresponding diagonal element of $\mathbf{V}$ as v_i^2 / V_{ii}. These statistics are treated as coming from a chi-square distribution for calculation of probability values.

The statistic $\mathbf{v}'\mathbf{V}^{-}\mathbf{v}$ is computed by sweeping each pivot of the $\mathbf{V}$ matrix in the order of greatest increase to the statistic. The corresponding sequence of partial statistics is tabulated. Sequential increments for including a given covariate and the corresponding probabilities are also included in the same table. These probabilities are calculated as the tail probabilities of a chi-square distribution with one degree of freedom. Because of the selection process, these probabilities should not be interpreted as p-values.

If desired for data screening purposes, the output data set requested by the OUTTEST= option can be treated as a sum of squares and crossproducts matrix and processed by the REG procedure by using the option METHOD=RSQUARE. Then the sets of variables of a given size can be found that give the largest test statistics. Output 49.1 illustrates this process.

Computer Resources

The data are first read and sorted into strata. If the data are originally sorted by failure time and censoring state, with smaller failure times coming first and event values preceding censored values in cases of ties, the data can be processed by strata without additional sorting. Otherwise, the data are read into memory by strata and sorted.

Memory Requirements

For a given BY group, define the following:

N	the total number of observations
V	the number of STRATA variables
C	the number of covariates listed in the TEST statement
L	total length of the ID variables in bytes
S	number of strata
n	maximum number of observations within strata

$$
\begin{array}{ll}
b & 12 + 8C + L \\
m1 & (112 + 16V) \times S \\
m2 & 50 \times b \times S \\
m3 & (50 + n) \times (b + 4) \\
m4 & 8(C + 4)^2 \\
m5 & 20N + 8S \times (S + 4)
\end{array}
$$

The memory, in bytes, required to process the BY group is at least

$$m1 + \max(m2, m3) + m4$$

The test of equality of survival functions across strata requires additional memory ($m5$ bytes). However, if this additional memory is not available, PROC LIFETEST skips the test for equality of survival functions and finishes the other computations. Additional memory is required for the PLOTS= option. Temporary storage of $16n$ bytes is required to store the product-limit estimates for plotting.

Output Data Sets

OUTSURV= Data Set

You can specify the OUTSURV= option in the PROC LIFETEST statement to create an output data set containing the following columns:

- any specified BY variables
- any specified STRATA variables, their values coming from either their original values or the midpoints of the stratum intervals if endpoints are used to define strata (semi-infinite intervals are labeled by their finite endpoint)
- STRATUM, a numeric variable that numbers the strata
- the time variable as given in the TIME statement. In the case of the product-limit estimates, it contains the observed failure or censored times. For the life-table estimates, it contains the lower endpoints of the time intervals.
- SURVIVAL, a variable containing the survivor function estimates
- CONFTYPE, a variable containing the name of the transformation applied to the survival time in the computation of confidence intervals (if the OUT= option is specified in the SURVIVAL statement)
- SDF_LCL, a variable containing the lower limits of the pointwise confidence intervals for the survivor function

- SDF_UCL, a variable containing the upper limits of the pointwise confidence intervals for the survivor function

If the estimation uses the product-limit method, then the data set also contains the following:

- _CENSOR_, an indicator variable that has a value 1 for a censored observation and a value 0 for an event observation
- SDF_STDERR, a variable containing the standard error of the survivor function estimator (if the STDERR option is specified in the PROC LIFETEST statement)
- HW_LCL, a variable containing the lower limits of the Hall-Wellner confidence bands (if the CONFBAND=HW option or the CONFBAND=ALL option is specified in the PROC LIFETEST statement)
- HW_UCL, a variable containing the upper limits of the Hall-Wellner confidence bands (if the CONFBAND=HW option or the CONFBAND=ALL option is specified in the PROC LIFETEST statement)
- EP_LCL, a variable containing the lower limits of the equal-precision confidence bands (if the CONFBAND=EP option or the CONFBAND=ALL option is specified in the PROC LIFETEST statement)
- EP_UCL, a variable containing the upper limits of the equal-precision confidence bands (if the CONFBAND=EP option or the CONFBAND=ALL option is specified in the PROC LIFETEST statement)

If the estimation uses the life-table method, then the data set also contains the following:

- MIDPOINT, a variable containing the value of the midpoint of the time interval
- PDF, a variable containing the density function estimates
- PDF_LCL, a variable containing the lower endpoints of the PDF confidence intervals
- PDF_UCL, a variable containing the upper endpoints of the PDF confidence intervals
- HAZARD, a variable containing the hazard estimates
- HAZ_LCL, a variable containing the lower endpoints of the hazard confidence intervals
- HAZ_UCL, a variable containing the upper endpoints of the hazard confidence intervals

Each survival function contains an initial observation with the value 1 for the SDF and the value 0 for the time. The output data set contains an observation for each distinct failure time if the product-limit method is used or an observation for each time interval if the life-table method is used. The product-limit survival estimates are defined to be right continuous; that is, the estimates at a given time include the factor for the failure events that occur at that time.

Labels are assigned to all the variables in the output data set except the BY variable and the STRATA variable.

OUTTEST= Data Set

The OUTTEST= option in the LIFETEST statement creates an output data set containing the rank statistics for testing the association of failure time with covariates. It contains the following:

- any specified BY variables
- _TYPE_, a character variable of length 8 that labels the type of rank test, either "LOG-RANK" or "WILCOXON"
- _NAME_, a character variable of length 8 that labels the rows of the covariance matrix and the test statistics
- the TIME variable, containing the overall test statistic in the observation that has _NAME_ equal to the name of the time variable and the univariate test statistics under their respective covariates.
- all variables listed in the TEST statement

The output is in the form of a symmetric matrix formed by the covariance matrix of the rank statistics bordered by the rank statistics and the overall chi-square statistic. If the value of _NAME_ is the name of a variable in the TEST statement, the observation contains a row of the covariance matrix and the value of the rank statistic in the time variable. If the value of _NAME_ is the name of the TIME variable, the observation contains the values of the rank statistics in the variables from the TEST list and the value of the overall chi-square test statistic in the TIME variable.

Two complete sets of statistics labeled by the _TYPE_ variable are produced, one for the log-rank test and one for the Wilcoxon test.

Displayed Output

If you use the NOPRINT option in the PROC LIFETEST statement, the procedure does not display any output.

Product-Limit Survival Estimates

The "Product-Limit Survival Estimates" table is displayed if you request the product-limit method of estimation. The table displays the observed (event or censored) time, the estimate of the survivor function, the estimate of the cumulative distribution function of the failure time, the standard error estimate of the estimated survivor function, the Nelson-Aalen cumulative hazard function estimate and its standard error (if the NELSON option is specified in the PROC LIFETEST statement), the number of event times that have been observed, and the number of event or censored times that remain to be observed. The table also includes the frequency of the observed event or censored times if you specify the FREQ statement, and the values of the ID variables if you specify the ID statement. For ODS purposes, the name of this table is "ProductLimitEstimates."

Quartile Estimates

The "Quartiles Estimates" table is displayed if you request the product-limit method of estimation. The table displays the point estimates of the quartiles of the survival times, and the lower and upper confidence limits for the quartiles. For ODS purposes, the name of this table is "Quartiles."

Mean Estimate

The "Mean Estimate" table is displayed if you request the product-limit method of estimation. The table displays the estimated mean survival time and its estimated standard error. For ODS purposes, the name of this table is "Means."

Life-Table Survival Estimates

The "Life-Table Survival Estimates" table is displayed if you request the life-table method of estimation. The table displays the time intervals into which the failure and censored times are distributed (each interval is from the lower limit, up to but not including the upper limit; if the upper limit is infinity, the missing value is printed), the number of events that occur in the interval, the number of censored observations that fall into the interval, the effective sample size for the interval, the estimate of conditional probability of events (failures) in the interval, the standard error of the estimated conditional probability of events, the estimate of the survival function at the beginning of the interval, the estimate of the cumulative distribution function of the failure time at the beginning of the interval, the standard error estimate of the estimated survival function, the estimate of the median residual lifetime (which is the amount of time elapsed before reducing the number of at-risk units to one-half; this is also known as the *median future lifetime* in Elandt-Johnson and Johnson (1980)), the estimated standard error of the estimated median residual lifetime, the density function estimated at the midpoint of the interval, the standard error estimate of the estimated density, the hazard rate estimated at the midpoint of the interval, and the standard error estimate of the estimated hazard. For ODS purposes, the name of this table is "LifetableEstimates."

Summary of the Number of Censored and Uncensored Values

The "Summary of the Number of Censored and Uncensored Values" table displays the stratum identification if the STRATA statement is specified, the total number of observations, the number of event observations, the number of censored observations, and the percentage of censored observations. For ODS purposes, the name of this table is "CensoredSummary."

Rank Statistics

The "Rank Statistics" table contains the test statistics of the nonparametric k-sample tests. For ODS purposes, the name of this table is "HomStats."

Covariance Matrix for the Log-Rank Statistics

The "Covariance Matrix for the Log-Rank Statistics" table is displayed if the log-rank k-sample test is requested. For ODS purposes, the name of this table is "LogrankHomCov."

Covariance Matrix for the Wilcoxon Statistics

The "Covariance Matrix for the Wilcoxon Statistics" table is displayed if the Wilcoxon k-sample test is requested. For ODS purposes, the name of this table is "WilHomCov."

Covariance Matrix for the Tarone Statistics

The "Covariance Matrix for the Tarone Statistics" table is displayed if the Tarone-Ware k-sample test is requested. For ODS purposes, the name of this table is "TaroneHomCov."

Covariance Matrix for the Peto Statistics

The "Covariance Matrix for the Peto Statistics" table is displayed if the Peto-Peto k-sample test is requested. For ODS purposes, the name of this table is "PetoHomCov."

Covariance Matrix for the ModPeto Statistics

The "Covariance Matrix for the ModPeto Statistics" table is displayed if the modified Peto-Peto k-sample test is requested. For ODS purposes, the name of this table is "ModPetoHomCov."

Covariance Matrix for the Fleming Statistics

The "Covariance Matrix for the Fleming Statistics" table is displayed if the Fleming-Harrington k-sample test is requested. For ODS purposes, the name of this table is "FlemingHomCov."

Test of Equality over Strata

The "Test of Equality over Strata" table is displayed if an unstratified k-sample test is carried out. The table contains the chi-square statistics, degrees of freedom, and p-values of the nonparametric tests and the likelihood ratio test (that is based on the exponential distribution). For ODS purposes, the name of this table is "HomTests."

Stratified Test of Equality over Group

The "Stratified Test of Equality over Group" table is displayed if a stratified test is carried out. The tables contains the chi-square statistics, degrees of freedom, and *p*-values of the stratified tests. For ODS purposes, the name of this table is "HomTests."

Scores for Trend Test

The "Scores for Trend Test" table is displayed if the TREND option is specified in the STRATA statement. The table contains the set of scores used to construct the trend tests. For ODS purposes, the name of this table is "TrendScores."

Trend Tests

The "Trend Tests" table is displayed if the TREND option is specified in the STRATA statement. The table contains the results of the trend tests. For ODS purposes, the name of this table is "TrendTests."

Adjustment for Multiple Comparisons for the Log-Rank Test

The "Adjustment for Multiple Comparisons for the Log-Rank Test" table is displayed if the log-rank test and a multiple-comparison adjustment method are specified. The table contains the chi-square statistics and the raw and adjusted *p*-values of the paired comparisons. For ODS purposes, the name of this table is "SurvDiff."

Adjustment for Multiple Comparisons for the Wilcoxon Test

The "Adjustment for Multiple Comparisons for the Wilcoxon Test" table is displayed if the Wilcoxon test and a multiple-comparison method are specified. The table contains the chi-square statistics and the raw and adjusted *p*-values of the paired comparisons. For ODS purposes, the name of this table is "SurvDiff."

Adjustment for Multiple Comparisons for the Tarone Test

The "Adjustment for Multiple Comparisons for the Tarone Test" table is displayed if the Tarone-Ware test and a multiple-comparison method are specified. The table contains the chi-square statistics and the raw and adjusted *p*-values of the paired comparisons. For ODS purposes, the name of this table is "SurvDiff."

Adjustment for Multiple Comparisons for the Peto Test

The "Adjustment for Multiple Comparisons for the Peto Test" table is displayed if the Peto-Peto test and a multiple-comparison method are specified. The table contains the chi-square statistics and the raw and adjusted *p*-values of the paired comparisons. For ODS purposes, the name of this table is "SurvDiff."

Adjustment for Multiple Comparisons for the ModPeto Test

The "Adjustment for Multiple Comparisons for the ModPeto Test" table is displayed if the modified Peto-Peto test and a multiple-comparison method are specified. The table contains the chi-square statistics and the raw and adjusted *p*-values of the paired comparisons. For ODS purposes, the name of this table is "SurvDiff."

Adjustment for Multiple Comparisons for the Fleming Test

The "Adjustment for Multiple Comparisons for the Fleming Test" table is displayed if the Fleming-Harrington test and a multiple-comparison method are specified. The table contains the chi-square statistics and the raw and adjusted *p*-values of the paired comparisons. For ODS purposes, the name of this table is "SurvDiff."

Univariate Chi-Squares for the Log-Rank Test

The "Univariate Chi-Squares for the Log-Rank Test" table is displayed if the TEST statement is specified. The table displays the log-rank test results for individual variables in the TEST statement. For ODS purposes, the name of this table is "LogUniChiSq."

Covariance Matrix of the Log-Rank Statistics

The "Covariance Matrix of the Log-Rank Statistics" table is displayed if the TEST statement is specified. The table displays the estimated covariance matrix of the log-rank statistics for association. For ODS purposes, the name of this table is "LogTestCov."

Forward Stepwise Sequence of Chi-Squares for the Log-Rank Test

The "Forward Stepwise Sequence of Chi-Squares for the Log-Rank Test" table is displayed if the TEST statement is specified. The table contains the sequence of partial chi-square statistics for the log-rank test in the order of the greatest increase to the overall test statistic, the degrees of freedom of the partial chi-square statistics, the approximate probability values of the partial chi-square statistics, the chi-square increments for including the given variables, and the probability values of the chi-square increments. For ODS purposes, the name of this table is "LogForStepSeq."

Univariate Chi-Squares for the Wilcoxon Test

The "Univariate Chi-Squares for the Wilcoxon Test" table displays the Wilcoxon test results for individual variables in the TEST statement. For ODS purposes, the name of this table is "WilUniChiSq."

Covariance Matrix of the Wilcoxon Statistics

The "Covariance Matrix of the Wilcoxon Statistics" table is displayed if the TEST statement is specified. The table displays the estimated covariance matrix of the Wilcoxon statistics for association. For ODS purposes, the name of this table is "WilTestCov."

Forward Stepwise Sequence of Chi-Squares for the Wilcoxon Test

The "Forward Stepwise Sequence of Chi-Squares for the Wilcoxon Test" table is displayed if the TEST statement is specified. The table contains the sequence of partial chi-square statistics for the Wilcoxon test in the order of the greatest increase to the overall test statistic, the degrees of freedom of the partial chi-square statistics, the approximate probability values of the partial chi-square statistics, the chi-square increments for including the given variables, and the probability values of the chi-square increments. For ODS purposes, the name of this table is "WilForStepSeq."

ODS Table Names

PROC LIFETEST assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 49.4. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 49.4 ODS Tables Produced by PROC LIFETEST

ODS Table Name	Description	Statement / Option
CensoredSummary	Number of event and censored observations	PROC LIFETEST METHOD=PL
FlemingHomCov	Covariance matrix for *k*-sample FLEMING statistics	STRATA / TEST=FLEMING
HomStats	Test statistics for *k*-sample tests	STRATA / TEST=
HomTests	Results of *k*-sample tests	STRATA / TEST=
LifetableEstimates	Life-table survival estimates	PROC LIFETEST METHOD=LT
LogForStepSeq	Forward stepwise sequence for the log-rank statistics for association	TEST /
LogHomCov	Covariance matrix for *k*-sample LOGRANK statistics	STRATA / TEST=LOGRANK

Table 49.4 *continued*

ODS Table Name	Description	Statement / Option
LogTestCov	Covariance matrix for log-rank statistics for association	TEST /
LogUniChisq	Univariate chi-squares for log-rank statistics for association	TEST /
Means	Mean and standard error of survival times	PROC LIFETEST METHOD=PL
ModPetoHomCov	Covariance matrix for *k*-sample MODPETO statistics	STRATA / TEST=MODPETO
PetoHomCov	Covariance matrix for *k*-sample PETO statistics	STRATA / TEST=PETO
ProductLimitEstimates	Product-limit survival estimates	PROC LIFETEST METHOD=PL
Quartiles	Quartiles of the survival times	PROC LIFETEST METHOD=PL
SurvDiff	Adjustments for multiple comparisons	STRATA / ADJUST= & DIFF=
TaroneHomCov	Covariance matrix for *k*-sample TARONE statistics	STRATA / TEST=TARONE
TrendScores	Scores used to construct trend tests	STRATA / TREND
TrendTests	Results of trend tests	STRATA / TREND
WilForStepSeq	Forward stepwise sequence for the log-rank statistics for association	TEST /
WilHomCov	Covariance matrix for *k*-sample WILCOXON statistics	STRATA / TEST=WILCOXON
WilTestCov	Covariance matrix for log-rank statistics for association	TEST /
WilUniChiSq	Univariate chi-squares for Wilcoxon statistics for association	TEST /

ODS Graphics

To request graphics with PROC LIFETEST, you must first enable ODS Graphics by specifying the **`ods graphics on`** statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. The survival plot is produced by default; other graphs are produced by using the PLOTS= option in the PROC LIFETEST statement. You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC LIFETEST generates are listed in Table 49.5, along with the required keywords for the PLOTS= option.

Table 49.5 ODS Graphics Produced by PROC LIFETEST

ODS Graph Name	Plot Description	PLOTS=Option
DensityPlot	Estimated density for life-table method	PDF
FailurePlot	Estimated failure function	SURVIVAL(FAILURE)
HazardPlot	Estimated hazard function for life-table method or smoothed hazard for product-limit method	HAZARD
LogNegLogSurvivalPlot	Log of negative log of the estimated survivor function	LOGLOGS
NegLogSurvivalPlot	Negative log of the estimated survivor function	LOGSURV
SurvivalPlot	Estimated survivor function	SURVIVAL
SurvivalPlot	Estimated survivor function with number of subjects at risk	SURVIVAL(ATRISK=)
SurvivalPlot	Estimated survivor function with pointwise confidence limits	SURVIVAL(CL)
SurvivalPlot	Estimated survivor function with equal-precision band	SURVIVAL(CB=EP)
SurvivalPlot	Estimated survivor function with Hall-Wellner band	SURVIVAL(CB=HW)
SurvivalPlot	Estimated survivor function with homogeneity test p-value	SURVIVAL(TEST)

Additional Dynamic Variables for Survival Plots Using ODS Graphics

PROC LIFETEST passes a number of summary statistics as dynamic variables to the ODS Graphics for survival plots. Table 49.6 and Table 49.7 list these additional dynamic variables for the Kaplan-Meier curves and the lifetable curves, respectively. These dynamic variables are not declared in the templates for the survival curves, but you can declare them and use them to enhance the default plots. The names of the dynamic variables depend on the STRATA= suboption of the PLOTS=SURVIVAL option: STRATA=INDIVIDUAL produces a separate plot for each stratum, and STRATA=OVERALL produces one plot with overlaid curves.

Table 49.6 Additional Dynamic Variables for `Stat.Graphics.ProductLimitSurvival`

STRATA=	Dynamic	Description
OVERLAY	StrVal$j$	Label for the jth stratum
	NObs$j$	Number of observations in the jth stratum
	NEvent$j$	Number of events in the jth stratum
	Median$j$	Median survival time of the jth stratum
	LowerMedian$j$	Lower median survival time of the jth stratum
	UpperMedian$j$	Upper median survival time of the jth stratum
	pctMedianConfid	Percent confidence of the median intervals
INDIVIDUAL	NObs	Number of observations
	NEvent	Number of events

Table 49.6 *continued*

STRATA=	Dynamic	Description
	Median	Median survival time
	LowerMedian	Lower median survival time
	UpperMedian	Upper median survival time
	pctMedianConfid	Percent confidence of the median interval

Table 49.7 Additional Dynamic Variables for `Stat.Graphics.LifetableSurvival`

STRATA=	Dynamic	Description
OVERLAY	StrVal*j*	Label for the *j*th stratum
	NObs*j*	Number of observations in the *j*th stratum
	NEvent*j*	Number of events in the *j*th stratum
INDIVIDUAL	NObs	Number of observations
	NEvent	Number of events

See the section "The Graph Template Language" on page 577 and Example 21.5 in Chapter 21, "Statistical Graphics Using ODS," for more information about dynamic variables and ODS Graphics.

Examples: LIFETEST Procedure

Example 49.1: Product-Limit Estimates and Tests of Association

The data presented in Appendix I of Kalbfleisch and Prentice (1980) are coded in the following DATA step. The response variable, SurvTime, is the survival time in days of a lung cancer patient. Negative values of SurvTime are censored values. The covariates are Cell (type of cancer cell), Therapy (type of therapy: standard or test), Prior (prior therapy: 0=no, 10=yes), Age (age in years), DiagTime (time in months from diagnosis to entry into the trial), and Kps (performance status). A censoring indicator variable Censor is created from the data, with the value 1 indicating a censored time and the value 0 indicating an event time. Since there are only two types of therapy, an indicator variable, Treatment, is constructed for therapy type, with value 0 for standard therapy and value 1 for test therapy.

```
data VALung;
   drop check m;
   retain Therapy Cell;
   infile cards column=column;
   length Check $ 1;
   label SurvTime='failure or censoring time'
      Kps='karnofsky index'
      DiagTime='months till randomization'
```

```
      Age='age in years'
      Prior='prior treatment?'
      Cell='cell type'
      Therapy='type of treatment'
      Treatment='treatment indicator';
   M=Column;
   input Check $ @@;
   if M>Column then M=1;
   if Check='s'|Check='t' then input @M Therapy $ Cell $ ;
   else input @M SurvTime Kps DiagTime Age Prior @@;
   if SurvTime > .;
   censor=(SurvTime<0);
   SurvTime=abs(SurvTime);
   Treatment=(Therapy='test');
   dataline;
standard squamous
 72 60  7 69  0   411 70  5 64 10   228 60  3 38  0   126 60  9 63 10
118 70 11 65 10    10 20  5 49  0    82 40 10 69 10   110 80 29 68  0
314 50 18 43  0  -100 70  6 70  0    42 60  4 81  0     8 40 58 63 10
144 30  4 63  0   -25 80  9 52 10    11 70 11 48 10
standard small
 30 60  3 61  0   384 60  9 42  0     4 40  2 35  0    54 80  4 63 10
 13 60  4 56  0  -123 40  3 55  0   -97 60  5 67  0   153 60 14 63 10
 59 30  2 65  0   117 80  3 46  0    16 30  4 53 10   151 50 12 69  0
 22 60  4 68  0    56 80 12 43 10    21 40  2 55 10    18 20 15 42  0
139 80  2 64  0    20 30  5 65  0    31 75  3 65  0    52 70  2 55  0
287 60 25 66 10    18 30  4 60  0    51 60  1 67  0   122 80 28 53  0
 27 60  8 62  0    54 70  1 67  0     7 50  7 72  0    63 50 11 48  0
392 40  4 68  0    10 40 23 67 10
standard adeno
  8 20 19 61 10    92 70 10 60  0    35 40  6 62  0   117 80  2 38  0
132 80  5 50  0    12 50  4 63 10   162 80  5 64  0     3 30  3 43  0
 95 80  4 34  0
standard large
177 50 16 66 10   162 80  5 62  0   216 50 15 52  0   553 70  2 47  0
278 60 12 63  0    12 40 12 68 10   260 80  5 45  0   200 80 12 41 10
156 70  2 66  0  -182 90  2 62  0   143 90  8 60  0   105 80 11 66  0
103 80  5 38  0   250 70  8 53 10   100 60 13 37 10
test squamous
999 90 12 54 10   112 80  6 60  0   -87 80  3 48  0  -231 50  8 52 10
242 50  1 70  0   991 70  7 50 10   111 70  3 62  0     1 20 21 65 10
587 60  3 58  0   389 90  2 62  0    33 30  6 64  0    25 20 36 63  0
357 70 13 58  0   467 90  2 64  0   201 80 28 52 10     1 50  7 35  0
 30 70 11 63  0    44 60 13 70 10   283 90  2 51  0    15 50 13 40 10
test small
 25 30  2 69  0  -103 70 22 36 10    21 20  4 71  0    13 30  2 62  0
 87 60  2 60  0     2 40 36 44 10    20 30  9 54 10     7 20 11 66  0
 24 60  8 49  0    99 70  3 72  0     8 80  2 68  0    99 85  4 62  0
 61 70  2 71  0    25 70  2 70  0    95 70  1 61  0    80 50 17 71  0
 51 30 87 59 10    29 40  8 67  0
test adeno
 24 40  2 60  0    18 40  5 69 10   -83 99  3 57  0    31 80  3 39  0
 51 60  5 62  0    90 60 22 50 10    52 60  3 43  0    73 60  3 70  0
  8 50  5 66  0    36 70  8 61  0    48 10  4 81  0     7 40  4 58  0
```

```
140 70  3 63  0   186 90  3 60  0    84 80  4 62 10    19 50 10 42  0
 45 40  3 69  0    80 40  4 63  0
test large
 52 60  4 45  0   164 70 15 68 10    19 30  4 39 10    53 60 12 66  0
 15 30  5 63  0    43 60 11 49 10   340 80 10 64 10   133 75  1 65  0
111 60  5 64  0   231 70 18 67 10   378 80  4 65  0    49 30  3 37  0
;
```

In the following statements, PROC LIFETEST is invoked to compute the product-limit estimate of the survivor function for each type of cancer cell and to analyze the effects of the variables Age, Prior, DiagTime, Kps, and Treatment on the survival of the patients. These prognostic factors are specified in the TEST statement, and the variable Cell is specified in the STRATA statement. Graphics results are enabled through ODS with the specification of the **ods graphics on** statement. Graphical display of the product-limit estimates (S), the negative log estimates (LS), and the log of negative log estimates (LLS) are requested through the PLOTS= option in the PROC LIFETEST statement. Because of a few large survival times, a MAXTIME of 600 is used to set the scale of the time axis; that is, the time scale extends from 0 to a maximum of 600 days in the plots. The variable Therapy is specified in the ID statement to identify the type of therapy for each observation in the product-limit estimates. The OUTTEST option specifies the creation of an output data set named Test to contain the rank test matrices for the covariates.

```
ods graphics on;
proc lifetest data=VALung plots=(s,ls,lls) outtest=Test maxtime=600;
   time SurvTime*Censor(1);
   id Therapy;
   strata Cell;
   test Age Prior DiagTime Kps Treatment;
run;
ods graphics off;
```

Output 49.1.1 through Output 49.1.4 display the product-limit estimates of the survivor functions for the four cell types. Summary statistics of the survival times are also shown. The median survival times are 51 days, 156 days, 51 days, and 118 days for patients with adeno cells, large cells, small cells, and squamous cells, respectively.

Output 49.1.1 Estimation Results for Cell=adeno

The LIFETEST Procedure

Stratum 1: Cell = adeno

Product-Limit Survival Estimates

SurvTime	Survival	Failure	Survival Standard Error	Number Failed	Number Left	Therapy
0.000	1.0000	0	0	0	27	
3.000	0.9630	0.0370	0.0363	1	26	standard
7.000	0.9259	0.0741	0.0504	2	25	test
8.000	.	.	.	3	24	standard
8.000	0.8519	0.1481	0.0684	4	23	test
12.000	0.8148	0.1852	0.0748	5	22	standard
18.000	0.7778	0.2222	0.0800	6	21	test
19.000	0.7407	0.2593	0.0843	7	20	test
24.000	0.7037	0.2963	0.0879	8	19	test
31.000	0.6667	0.3333	0.0907	9	18	test
35.000	0.6296	0.3704	0.0929	10	17	standard
36.000	0.5926	0.4074	0.0946	11	16	test
45.000	0.5556	0.4444	0.0956	12	15	test
48.000	0.5185	0.4815	0.0962	13	14	test
51.000	0.4815	0.5185	0.0962	14	13	test
52.000	0.4444	0.5556	0.0956	15	12	test
73.000	0.4074	0.5926	0.0946	16	11	test
80.000	0.3704	0.6296	0.0929	17	10	test
83.000*	.	.	.	17	9	test
84.000	0.3292	0.6708	0.0913	18	8	test
90.000	0.2881	0.7119	0.0887	19	7	test
92.000	0.2469	0.7531	0.0850	20	6	standard
95.000	0.2058	0.7942	0.0802	21	5	standard
117.000	0.1646	0.8354	0.0740	22	4	standard
132.000	0.1235	0.8765	0.0659	23	3	standard
140.000	0.0823	0.9177	0.0553	24	2	test
162.000	0.0412	0.9588	0.0401	25	1	standard
186.000	0	1.0000	.	26	0	test

NOTE: The marked survival times are censored observations.

Output 49.1.2 Estimation Results for Cell=large

The LIFETEST Procedure

Stratum 2: Cell = large

Product-Limit Survival Estimates

SurvTime	Survival	Failure	Survival Standard Error	Number Failed	Number Left	Therapy
0.000	1.0000	0	0	0	27	
12.000	0.9630	0.0370	0.0363	1	26	standard
15.000	0.9259	0.0741	0.0504	2	25	test
19.000	0.8889	0.1111	0.0605	3	24	test
43.000	0.8519	0.1481	0.0684	4	23	test
49.000	0.8148	0.1852	0.0748	5	22	test
52.000	0.7778	0.2222	0.0800	6	21	test
53.000	0.7407	0.2593	0.0843	7	20	test
100.000	0.7037	0.2963	0.0879	8	19	standard
103.000	0.6667	0.3333	0.0907	9	18	standard
105.000	0.6296	0.3704	0.0929	10	17	standard
111.000	0.5926	0.4074	0.0946	11	16	test
133.000	0.5556	0.4444	0.0956	12	15	test
143.000	0.5185	0.4815	0.0962	13	14	standard
156.000	0.4815	0.5185	0.0962	14	13	standard
162.000	0.4444	0.5556	0.0956	15	12	standard
164.000	0.4074	0.5926	0.0946	16	11	test
177.000	0.3704	0.6296	0.0929	17	10	standard
182.000*	.	.	.	17	9	standard
200.000	0.3292	0.6708	0.0913	18	8	standard
216.000	0.2881	0.7119	0.0887	19	7	standard
231.000	0.2469	0.7531	0.0850	20	6	test
250.000	0.2058	0.7942	0.0802	21	5	standard
260.000	0.1646	0.8354	0.0740	22	4	standard
278.000	0.1235	0.8765	0.0659	23	3	standard
340.000	0.0823	0.9177	0.0553	24	2	test
378.000	0.0412	0.9588	0.0401	25	1	test
553.000	0	1.0000	.	26	0	standard

NOTE: The marked survival times are censored observations.

Output 49.1.3 Estimation Results for Cell=small

The LIFETEST Procedure

Stratum 3: Cell = small

Product-Limit Survival Estimates

SurvTime	Survival	Failure	Survival Standard Error	Number Failed	Number Left	Therapy
0.000	1.0000	0	0	0	48	
2.000	0.9792	0.0208	0.0206	1	47	test
4.000	0.9583	0.0417	0.0288	2	46	standard
7.000	.	.	.	3	45	standard
7.000	0.9167	0.0833	0.0399	4	44	test
8.000	0.8958	0.1042	0.0441	5	43	test
10.000	0.8750	0.1250	0.0477	6	42	standard
13.000	.	.	.	7	41	standard
13.000	0.8333	0.1667	0.0538	8	40	test
16.000	0.8125	0.1875	0.0563	9	39	standard
18.000	.	.	.	10	38	standard
18.000	0.7708	0.2292	0.0607	11	37	standard
20.000	.	.	.	12	36	standard
20.000	0.7292	0.2708	0.0641	13	35	test
21.000	.	.	.	14	34	standard
21.000	0.6875	0.3125	0.0669	15	33	test
22.000	0.6667	0.3333	0.0680	16	32	standard
24.000	0.6458	0.3542	0.0690	17	31	test
25.000	.	.	.	18	30	test
25.000	0.6042	0.3958	0.0706	19	29	test
27.000	0.5833	0.4167	0.0712	20	28	standard
29.000	0.5625	0.4375	0.0716	21	27	test
30.000	0.5417	0.4583	0.0719	22	26	standard
31.000	0.5208	0.4792	0.0721	23	25	standard
51.000	.	.	.	24	24	standard
51.000	0.4792	0.5208	0.0721	25	23	test
52.000	0.4583	0.5417	0.0719	26	22	standard
54.000	.	.	.	27	21	standard
54.000	0.4167	0.5833	0.0712	28	20	standard
56.000	0.3958	0.6042	0.0706	29	19	standard
59.000	0.3750	0.6250	0.0699	30	18	standard
61.000	0.3542	0.6458	0.0690	31	17	test
63.000	0.3333	0.6667	0.0680	32	16	standard
80.000	0.3125	0.6875	0.0669	33	15	test
87.000	0.2917	0.7083	0.0656	34	14	test
95.000	0.2708	0.7292	0.0641	35	13	test
97.000*	.	.	.	35	12	standard
99.000	.	.	.	36	11	test
99.000	0.2257	0.7743	0.0609	37	10	test
103.000*	.	.	.	37	9	test
117.000	0.2006	0.7994	0.0591	38	8	standard
122.000	0.1755	0.8245	0.0567	39	7	standard
123.000*	.	.	.	39	6	standard
139.000	0.1463	0.8537	0.0543	40	5	standard
151.000	0.1170	0.8830	0.0507	41	4	standard
153.000	0.0878	0.9122	0.0457	42	3	standard
287.000	0.0585	0.9415	0.0387	43	2	standard
384.000	0.0293	0.9707	0.0283	44	1	standard
392.000	0	1.0000	.	45	0	standard

NOTE: The marked survival times are censored observations.

Output 49.1.4 Estimation Results for Cell=squamous

The LIFETEST Procedure

Stratum 4: Cell = squamous

Product-Limit Survival Estimates

SurvTime	Survival	Failure	Survival Standard Error	Number Failed	Number Left	Therapy
0.000	1.0000	0	0	0	35	
1.000	.	.	.	1	34	test
1.000	0.9429	0.0571	0.0392	2	33	test
8.000	0.9143	0.0857	0.0473	3	32	standard
10.000	0.8857	0.1143	0.0538	4	31	standard
11.000	0.8571	0.1429	0.0591	5	30	standard
15.000	0.8286	0.1714	0.0637	6	29	test
25.000	0.8000	0.2000	0.0676	7	28	test
25.000*	.	.	.	7	27	standard
30.000	0.7704	0.2296	0.0713	8	26	test
33.000	0.7407	0.2593	0.0745	9	25	test
42.000	0.7111	0.2889	0.0772	10	24	standard
44.000	0.6815	0.3185	0.0794	11	23	test
72.000	0.6519	0.3481	0.0813	12	22	standard
82.000	0.6222	0.3778	0.0828	13	21	standard
87.000*	.	.	.	13	20	test
100.000*	.	.	.	13	19	standard
110.000	0.5895	0.4105	0.0847	14	18	standard
111.000	0.5567	0.4433	0.0861	15	17	test
112.000	0.5240	0.4760	0.0870	16	16	test
118.000	0.4912	0.5088	0.0875	17	15	standard
126.000	0.4585	0.5415	0.0876	18	14	standard
144.000	0.4257	0.5743	0.0873	19	13	standard
201.000	0.3930	0.6070	0.0865	20	12	test
228.000	0.3602	0.6398	0.0852	21	11	standard
231.000*	.	.	.	21	10	test
242.000	0.3242	0.6758	0.0840	22	9	test
283.000	0.2882	0.7118	0.0820	23	8	test
314.000	0.2522	0.7478	0.0793	24	7	standard
357.000	0.2161	0.7839	0.0757	25	6	test
389.000	0.1801	0.8199	0.0711	26	5	test
411.000	0.1441	0.8559	0.0654	27	4	standard
467.000	0.1081	0.8919	0.0581	28	3	test
587.000	0.0720	0.9280	0.0487	29	2	test
991.000	0.0360	0.9640	0.0352	30	1	test
999.000	0	1.0000	.	31	0	test

NOTE: The marked survival times are censored observations.

The distribution of event and censored observations among the four cell types is summarized in Output 49.1.5.

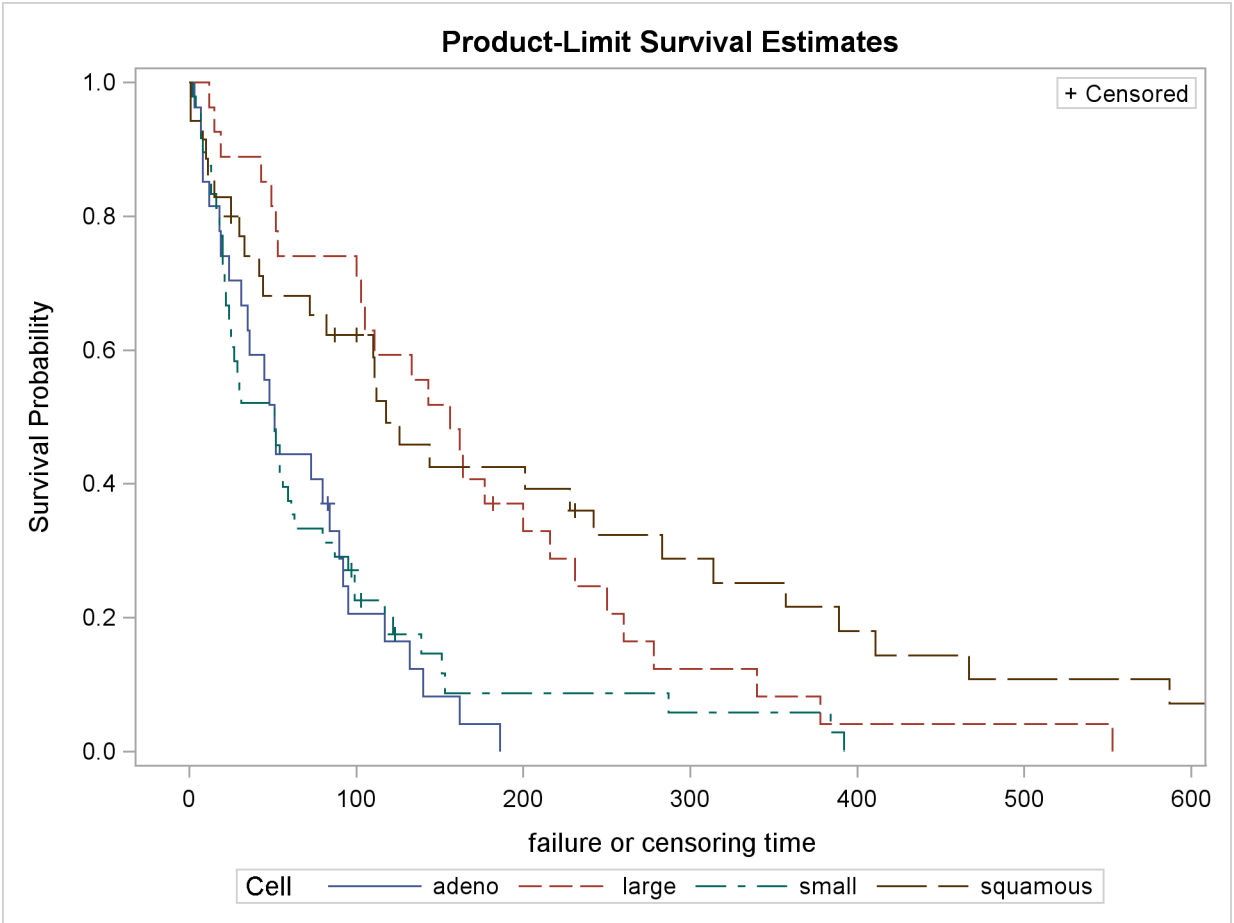
Output 49.1.5 Summary of Censored and Uncensored Values

Summary of the Number of Censored and Uncensored Values

Stratum	Cell	Total	Failed	Censored	Percent Censored
1	adeno	27	26	1	3.70
2	large	27	26	1	3.70
3	small	48	45	3	6.25
4	squamous	35	31	4	11.43
Total		137	128	9	6.57

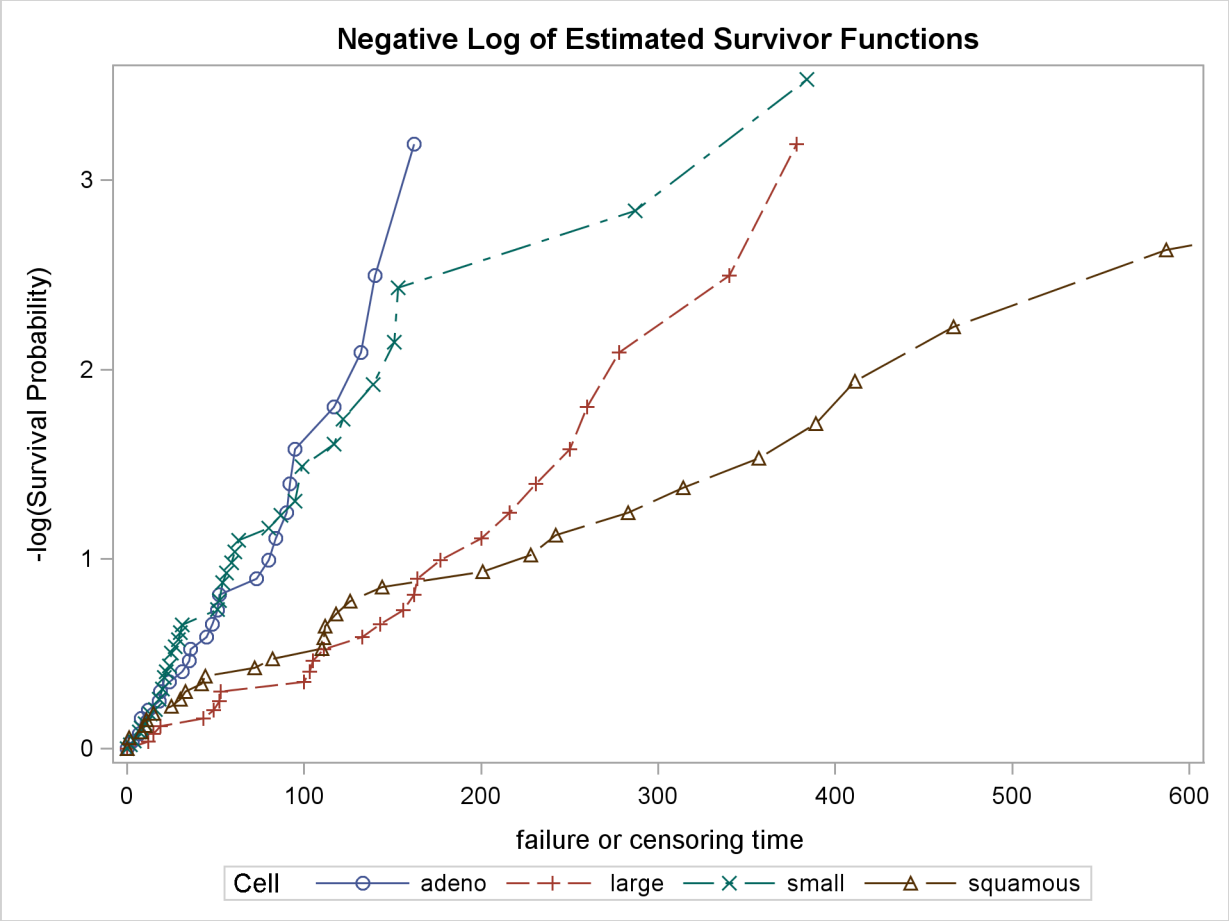
The graph of the estimated survivor functions is shown in Output 49.1.6. The adeno cell curve and the small cell curve are much closer to each other than they are to the large cell curve or the squamous cell curve. The survival rates of the adeno cell patients and the small cell patients decrease rapidly to approximately 29% in 90 days. Shapes of the large cell curve and the squamous cell curve are quite different, although both decrease less rapidly than those of the adeno and small cells. The squamous cell curve decreases more rapidly initially than the large cell curve, but the role is reversed in the later period.

Output 49.1.6 Graph of the Estimated Survivor Functions

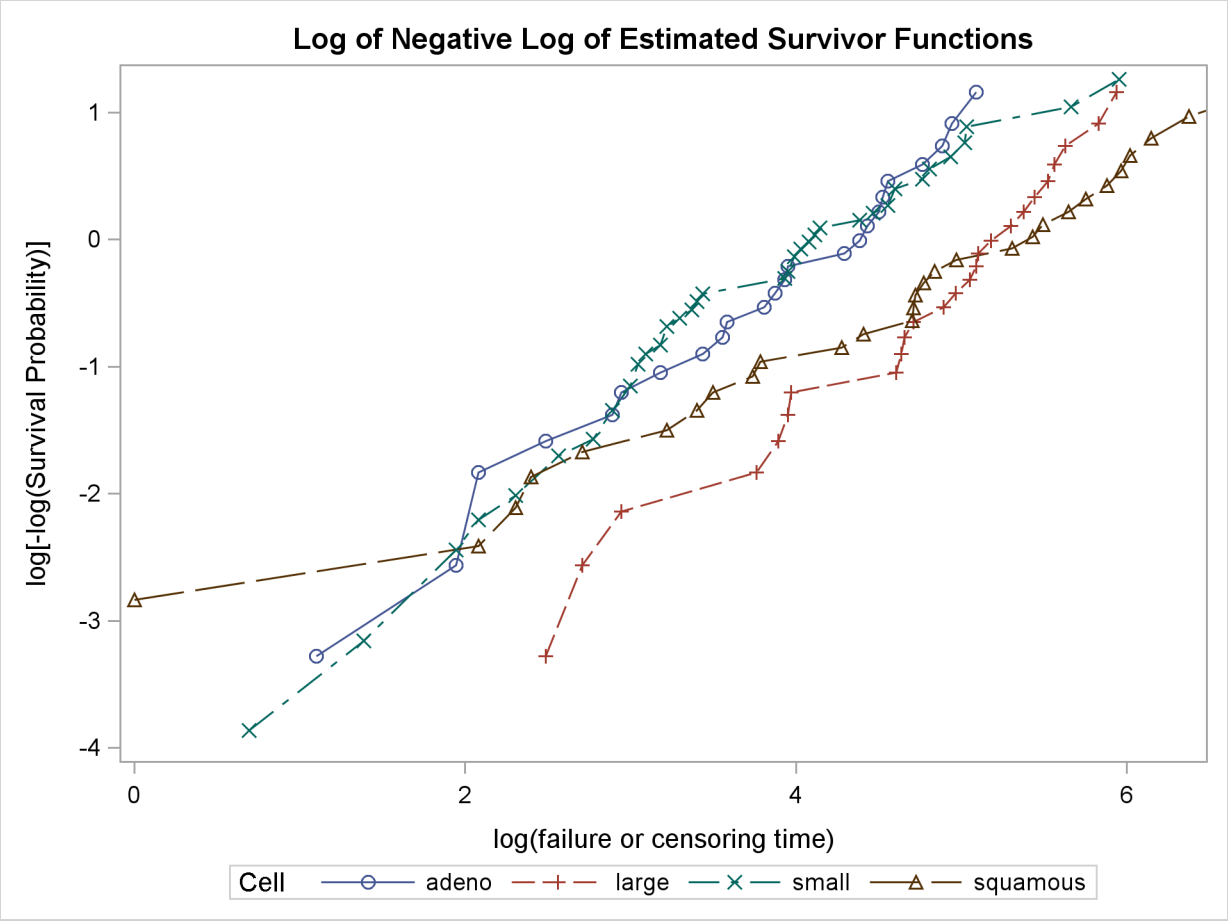


The graph of the negative log of the estimated survivor functions is displayed in Output 49.1.7. Output 49.1.8 displays the log of the negative log of the estimated survivor functions against the log of time.

Output 49.1.7 Graph of Negative Log of the Estimated Survivor Functions



Output 49.1.8 Graph of Log of the Negative Log of the Estimated Survivor Functions



Results of the homogeneity tests across cell types are given in Output 49.1.9. The log-rank and Wilcoxon statistics and their corresponding covariance matrices are displayed. Also given is a table that consists of the approximate chi-square statistics, degrees of freedom, and p-values for the log-rank, Wilcoxon, and likelihood ratio tests. All three tests indicate strong evidence of a significant difference among the survival curves for the four types of cancer cells (p<0.0001).

Output 49.1.9 Homogeneity Tests across Cell Types

Rank Statistics

Cell	Log-Rank	Wilcoxon
adeno	10.306	697.0
large	-8.549	-1085.0
small	14.898	1278.0
squamous	-16.655	-890.0

Output 49.1.9 *continued*

Covariance Matrix for the Log-Rank Statistics

Cell	adeno	large	small	squamous
adeno	12.9662	-4.0701	-4.4087	-4.4873
large	-4.0701	24.1990	-7.8117	-12.3172
small	-4.4087	-7.8117	21.7543	-9.5339
squamous	-4.4873	-12.3172	-9.5339	26.3384

Covariance Matrix for the Wilcoxon Statistics

Cell	adeno	large	small	squamous
adeno	121188	-34718	-46639	-39831
large	-34718	151241	-59948	-56576
small	-46639	-59948	175590	-69002
squamous	-39831	-56576	-69002	165410

Test of Equality over Strata

Test	Chi-Square	DF	Pr > Chi-Square
Log-Rank	25.4037	3	<.0001
Wilcoxon	19.4331	3	0.0002
-2Log(LR)	33.9343	3	<.0001

Results of the log-rank test of the prognostic variables are shown in Output 49.1.10. The univariate test results correspond to testing each prognostic factor marginally. The joint covariance matrix of these univariate test statistics is also displayed. In computing the overall chi-square statistic, the partial chi-square statistics following a forward stepwise entry approach are tabulated.

Consider the log-rank test in Output 49.1.10. Since the univariate test for Kps has the largest chi-square (43.4747) among all the covariates, Kps is entered first. At this stage, the partial chi-square and the chi-square increment for Kps are the same as the univariate chi-square. Among all the covariates not in the model (Age, Prior, DiagTime, Treatment), Treatment has the largest approximate chi-square increment (1.7261) and is entered next. The approximate chi-square for the model containing Kps and Treatment is 43.4747+1.7261=45.2008 with 2 degrees of freedom. The third covariate entered is Age. The fourth is Prior, and the fifth is DiagTime. The overall chi-square statistic in the last line of the output is the partial chi-square for including all the covariates. It has a value of 46.4200 with 5 degrees of freedom, which is highly significant (p<0.0001).

Output 49.1.10 Log-Rank Test of the Prognostic Factors

Univariate Chi-Squares for the Log-Rank Test

Variable	Test Statistic	Standard Error	Chi-Square	Pr > Chi-Square	Label
Age	-40.7383	105.7	0.1485	0.7000	age in years
Prior	-19.9435	46.9836	0.1802	0.6712	prior treatment?
DiagTime	-115.9	97.8708	1.4013	0.2365	months till randomization
Kps	1123.1	170.3	43.4747	<.0001	karnofsky index
Treatment	-4.2076	5.0407	0.6967	0.4039	treatment indicator

Covariance Matrix for the Log-Rank Statistics

Variable	Age	Prior	DiagTime	Kps	Treatment
Age	11175.4	-301.2	-892.2	-2948.4	119.3
Prior	-301.2	2207.5	2010.9	78.6	13.9
DiagTime	-892.2	2010.9	9578.7	-2295.3	21.9
Kps	-2948.4	78.6	-2295.3	29015.6	61.9
Treatment	119.3	13.9	21.9	61.9	25.4

Forward Stepwise Sequence of Chi-Squares for the Log-Rank Test

Variable	DF	Chi-Square	Pr > Chi-Square	Chi-Square Increment	Pr > Increment
Kps	1	43.4747	<.0001	43.4747	<.0001
Treatment	2	45.2008	<.0001	1.7261	0.1889
Age	3	46.3012	<.0001	1.1004	0.2942
Prior	4	46.4134	<.0001	0.1122	0.7377
DiagTime	5	46.4200	<.0001	0.00665	0.9350

Variable	Label
Kps	karnofsky index
Treatment	treatment indicator
Age	age in years
Prior	prior treatment?
DiagTime	months till randomization

You can establish this forward stepwise entry of prognostic factors by passing the matrix corresponding to the log-rank test to the RSQUARE method in the REG procedure, as follows. PROC REG finds the sets of variables that yield the largest chi-square statistics.

```
data RSq;
   set Test;
   if _type_='LOG RANK';
   _type_='cov';
 proc print data=RSq;
 run;
 proc reg data=RSq(type=COV);
    model SurvTime=Age Prior DiagTime Kps Treatment
       / selection=rsquare;
    title 'All Possible Subsets of Covariates for the log-rank Test';
run;
```

Output 49.1.11 displays the univariate statistics and their covariance matrix for the log-rank test.

Output 49.1.11 Log-Rank Statistics and Covariance Matrix

All Possible Subsets of Covariates for the log-rank Test

Obs	_TYPE_	_NAME_	SurvTime	Age	Prior	DiagTime	Kps	Treatment
1	cov	SurvTime	46.42	-40.74	-19.94	-115.86	1123.14	-4.208
2	cov	Age	-40.74	11175.44	-301.23	-892.24	-2948.45	119.297
3	cov	Prior	-19.94	-301.23	2207.46	2010.85	78.64	13.875
4	cov	DiagTime	-115.86	-892.24	2010.85	9578.69	-2295.32	21.859
5	cov	Kps	1123.14	-2948.45	78.64	-2295.32	29015.62	61.945
6	cov	Treatment	-4.21	119.30	13.87	21.86	61.95	25.409

Results of the best subset regression are shown in Output 49.1.12. The variable Kps generates the largest univariate test statistic among all the covariates, the pair Kps and Age generate the largest test statistic among any other pairs of covariates, and so on. The entry order of covariates is identical to that of PROC LIFETEST.

Output 49.1.12 Best Subset Regression from the REG Procedure

All Possible Subsets of Covariates for the log-rank Test

The REG Procedure
Model: MODEL1
Dependent Variable: SurvTime

R-Square Selection Method

Number in Model	R-Square	Variables in Model
1	0.9366	Kps
1	0.0302	DiagTime
1	0.0150	Treatment
1	0.0039	Prior
1	0.0032	Age
2	0.9737	Kps Treatment
2	0.9472	Age Kps
2	0.9417	Prior Kps
2	0.9382	DiagTime Kps
2	0.0434	DiagTime Treatment
2	0.0353	Age DiagTime
2	0.0304	Prior DiagTime
2	0.0181	Prior Treatment
2	0.0159	Age Treatment
2	0.0075	Age Prior
3	0.9974	Age Kps Treatment
3	0.9774	Prior Kps Treatment
3	0.9747	DiagTime Kps Treatment
3	0.9515	Age Prior Kps
3	0.9481	Age DiagTime Kps
3	0.9418	Prior DiagTime Kps
3	0.0456	Age DiagTime Treatment
3	0.0438	Prior DiagTime Treatment
3	0.0355	Age Prior DiagTime
3	0.0192	Age Prior Treatment
4	0.9999	Age Prior Kps Treatment
4	0.9976	Age DiagTime Kps Treatment
4	0.9774	Prior DiagTime Kps Treatment
4	0.9515	Age Prior DiagTime Kps
4	0.0459	Age Prior DiagTime Treatment
5	1.0000	Age Prior DiagTime Kps Treatment

Example 49.2: Enhanced Survival Plot and Multiple-Comparison Adjustments

This example highlights some of the new features of PROC LIFETEST for SAS 9.2, especially the survival plot with number of subjects at risk and multiple comparisons of survival curves. In the following DATA step, data of 137 bone marrow transplant patients extracted from Klein and Moeschberger (1997) are saved in the data set BMT . At the time of transplant, each patient is classified into one of three risk categories: ALL (acute lymphoblastic leukemia), AML (acute myeloctic leukemia)-Low Risk, and AML-High Risk. The endpoint of interest is the disease-free survival time, which is the time to death or relapse or the end of the study in days. In this data set, the variable Group represents the patient's risk category, the variable T represents the disease-free survival time, and the variable Status is the censoring indicator, with the value 1 indicating an event time and the value 0 a censored time.

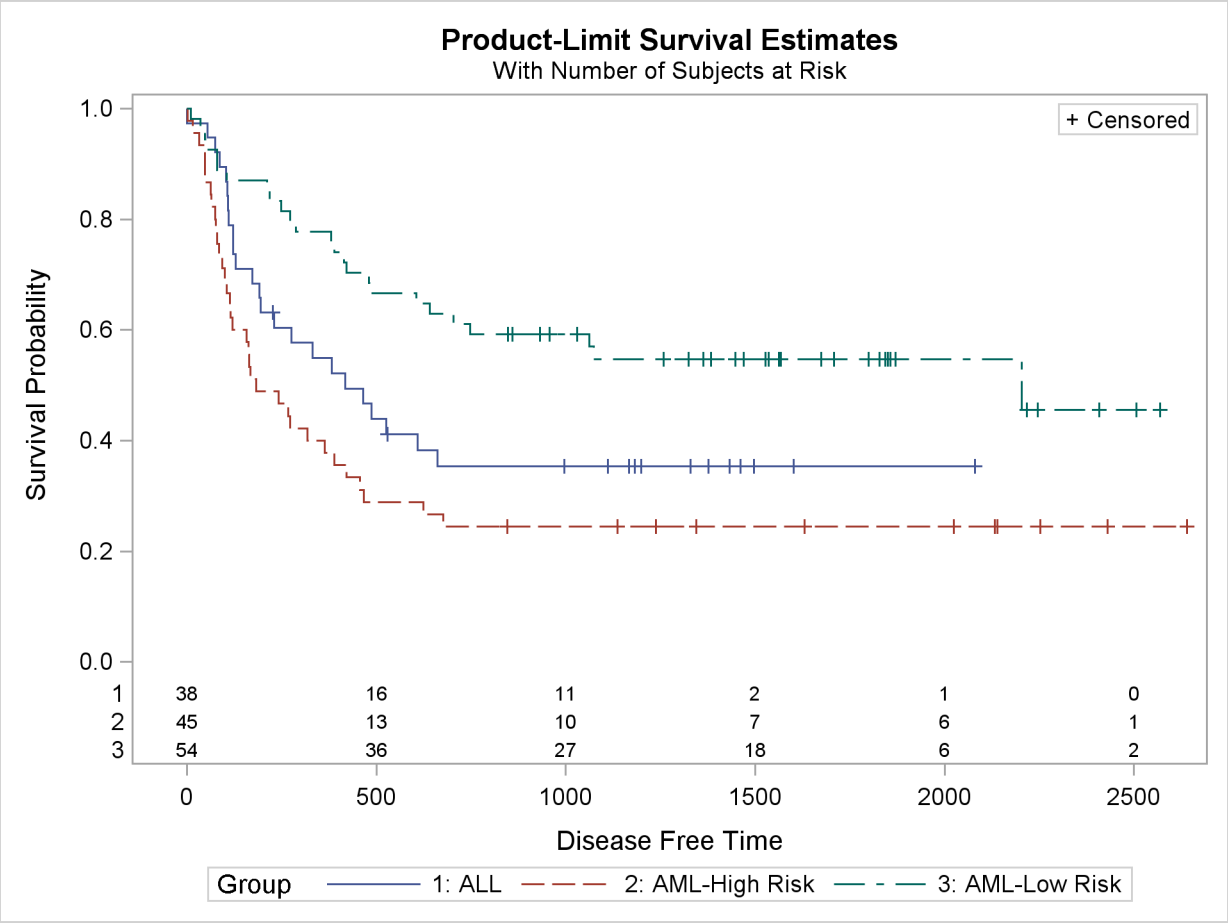
```
proc format;
   value risk 1='ALL' 2='AML-Low Risk' 3='AML-High Risk';

data BMT;
   input Group T Status @@;
   format Group risk.;
   label T='Disease Free Time';
   datalines;
1 2081 0 1 1602 0 1 1496 0 1 1462 0 1 1433 0
1 1377 0 1 1330 0 1  996 0 1  226 0 1 1199 0
1 1111 0 1  530 0 1 1182 0 1 1167 0 1  418 1
1  383 1 1  276 1 1  104 1 1  609 1 1  172 1
1  487 1 1  662 1 1  194 1 1  230 1 1  526 1
1  122 1 1  129 1 1   74 1 1  122 1 1   86 1
1  466 1 1  192 1 1  109 1 1   55 1 1    1 1
1  107 1 1  110 1 1  332 1 2 2569 0 2 2506 0
2 2409 0 2 2218 0 2 1857 0 2 1829 0 2 1562 0
2 1470 0 2 1363 0 2 1030 0 2  860 0 2 1258 0
2 2246 0 2 1870 0 2 1799 0 2 1709 0 2 1674 0
2 1568 0 2 1527 0 2 1324 0 2  957 0 2  932 0
2  847 0 2  848 0 2 1850 0 2 1843 0 2 1535 0
2 1447 0 2 1384 0 2  414 1 2 2204 1 2 1063 1
2  481 1 2  105 1 2  641 1 2  390 1 2  288 1
2  421 1 2   79 1 2  748 1 2  486 1 2   48 1
2  272 1 2 1074 1 2  381 1 2   10 1 2   53 1
2   80 1 2   35 1 2  248 1 2  704 1 2  211 1
2  219 1 2  606 1 3 2640 0 3 2430 0 3 2252 0
3 2140 0 3 2133 0 3 1238 0 3 1631 0 3 2024 0
3 1345 0 3 1136 0 3  845 0 3  422 1 3  162 1
3   84 1 3  100 1 3    2 1 3   47 1 3  242 1
3  456 1 3  268 1 3  318 1 3   32 1 3  467 1
3   47 1 3  390 1 3  183 1 3  105 1 3  115 1
3  164 1 3   93 1 3  120 1 3   80 1 3  677 1
3   64 1 3  168 1 3   74 1 3   16 1 3  157 1
3  625 1 3   48 1 3  273 1 3   63 1 3   76 1
3  113 1 3  363 1
;
```

In the following statements, PROC LIFETEST is invoked to compute the product-limit estimate of the survivor function for each risk category. Using ODS Graphics, you can display the number of subjects at risk in the survival plot. The PLOTS= option requests that the survival curves be plotted, and the ATRISK= suboption specifies the time points at which the at-risk numbers are displayed. In the STRATA statement, the ADJUST=SIDAK option requests the Šidák multiple-comparison adjustment, and by default, all paired comparisons are carried out.

```
ods graphics on;

proc lifetest data=BMT plots=survival(atrisk=0 to 2500 by 500);
   time T * Status(0);
   strata Group / test=logrank adjust=sidak;
   run;
```

Output 49.2.1 displays the estimated disease free survival for the three leukemia groups with the number of subjects at risk at 0, 500, 1000, 1500, 2000, and 2500 days. Patients in the AML-Low Risk group experience disease free longer than those in the ALL group, who in turn fare better than those in the AML-High Risk group.

Output 49.2.1 Estimated Disease-Free Survival for 137 Bone Marrow Transplant Patients



The log-rank test (Output 49.2.2) shows that the disease-free survival times for these three risk groups are significantly different (p=0.001).

Output 49.2.2 Log-Rank Test of Disease Group Homogeneity

Test of Equality over Strata

Test	Chi-Square	DF	Pr > Chi-Square
Log-Rank	13.8037	2	0.0010

The Šidák multiple-comparison results are shown in Output 49.2.3. There is no significant difference in disease-free survivor functions between the ALL and AML-High Risk groups (p=0.2779). The difference between the ALL and AML-Low Risk groups is marginal (p=0.0685), but the AML-Low Risk and AML-High Risk groups have significantly different disease-free survivor functions (p=0.0006).

Output 49.2.3 All Paired Comparsions

Adjustment for Multiple Comparisons for the Logrank Test

Strata Comparison Group	Group	Chi-Square	p-Values Raw	Sidak
ALL	AML-High Risk	2.6610	0.1028	0.2779
ALL	AML-Low Risk	5.1400	0.0234	0.0685
AML-High Risk	AML-Low Risk	13.8011	0.0002	0.0006

Suppose you consider the AML-Low Risk group as the reference group. You can use the DIFF= option in the STRATA statement to designate this risk group as the control and apply a multiple-comparison adjustment to the p-values for the paired comparison between the AML-Low Risk group with each of the other groups. Consider the Šidák correction again. You specify the ADJUST= and DIFF= options as in the following statements.

```
proc lifetest data=BMT notable plots=none;
   time T * Status(0);
   strata Group / test=logrank adjust=sidak diff=control('AML-Low Risk');
   run;
```

Output 49.2.4 shows that although both the ALL and AML-High Risk groups differ from the AML-Low Risk group at the 0.05 level, the difference between the AML-High Risk and the AML-Low Risk group is highly significant (p=0.0004).

Output 49.2.4 Comparisons with the Reference Group

The LIFETEST Procedure

Adjustment for Multiple Comparisons for the Logrank Test

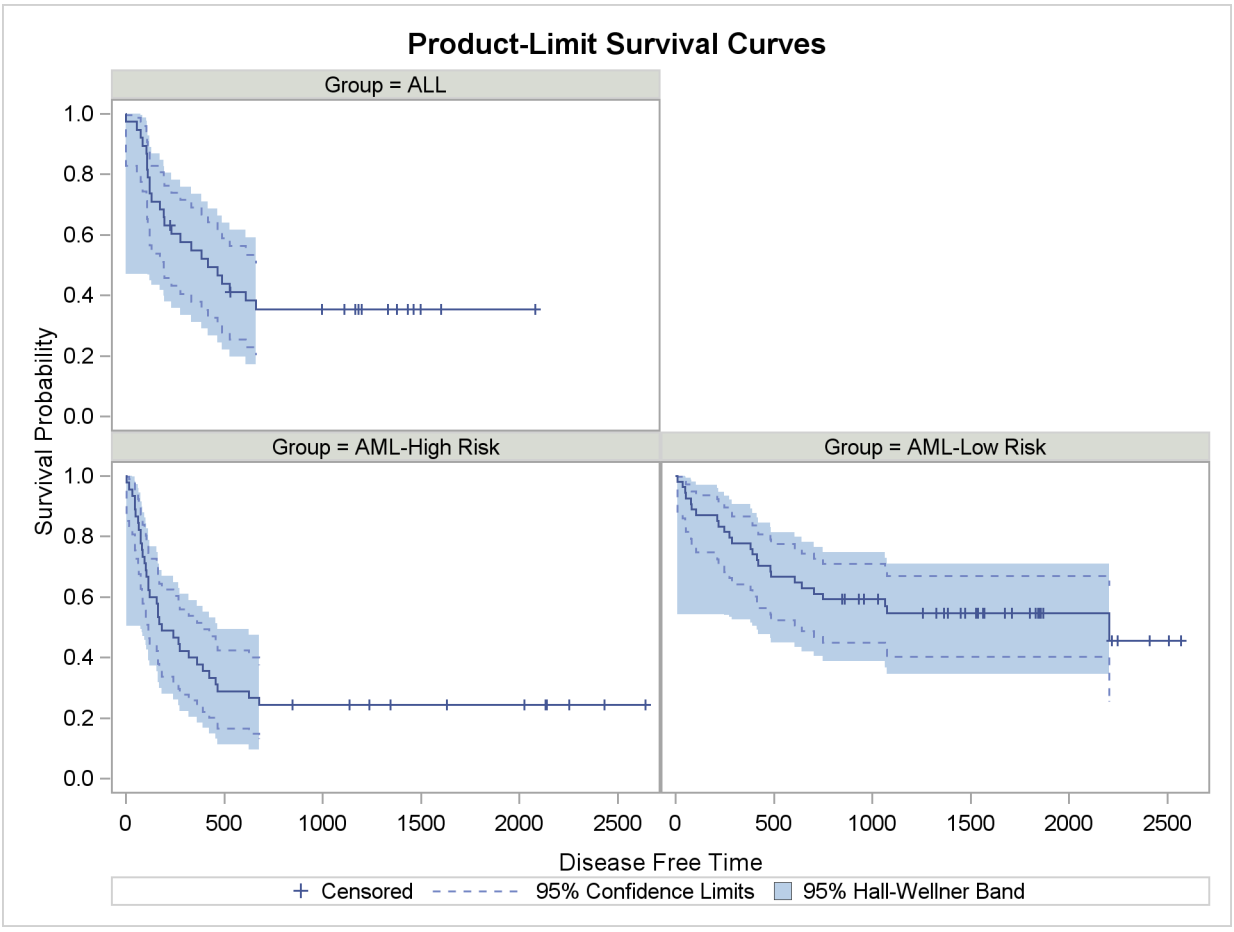
Strata Comparison Group	Group	Chi-Square	p-Values Raw	Sidak
ALL	AML-Low Risk	5.1400	0.0234	0.0462
AML-High Risk	AML-Low Risk	13.8011	0.0002	0.0004

Klein and Moeschberger (1997, Section 4.4) describe in detail how to compute the Hall-Wellner (HW) and equal-precision (EP) confidence bands for the survivor function. You can output these simultaneous confidence intervals to a SAS data set by using the CONFBAND= and OUTSURV= options in the PROC LIFETEST statement. You can display survival curves with pointwise and simultaneous confidence limits through ODS Graphics. When the survival data are stratified, displaying all the survival curves and their confidence limits in the same plot can make the plot appear cluttered. In the following statements, the PLOTS= specification requests that the survivor functions be displayed along with their pointwise confidence limits (CL) and Hall-Wellner confidence bands (CB=HW). The STRATA=PANEL specification requests that the survival curves be displayed in a panel of three plots, one for each risk group.

```
proc lifetest data=BMT plots=survival(cl cb=hw strata=panel);
   time T * Status(0);
   strata Group;
   run;

ods graphics off;
```

The panel plot is shown in Output 49.2.5.

Output 49.2.5 Estimated Disease-Free Survivor Functions with Confidence Limits



Example 49.3: Life-Table Estimates for Males with Angina Pectoris

The data in this example come from Lee (1992, p. 91) and represent the survival rates of males with angina pectoris. Survival time is measured as years from the time of diagnosis. In the following DATA step, the data are read as number of events and number of withdrawals in each one-year time interval for 16 intervals. Three variables are constructed from the data: Years (an artificial time variable with values that are the midpoints of the time intervals), Censored (a censoring indicator variable with the value 1 indicating censored observations and the value 0 indicating event observations), and Freq (the frequency variable). Two observations are created for each interval, one representing the event observations and the other representing the censored observations.

```
title 'Survival of Males with Angina Pectoris';
data Males;
   keep Freq Years Censored;
   retain Years -.5;
   input fail withdraw @@;
   Years + 1;
   Censored=0;
   Freq=fail;
   output;
   Censored=1;
   Freq=withdraw;
   output;
   datalines;
456   0 226  39 152  22 171  23 135 24 125 107
 83 133  74 102  51  68  42  64  43 45  34  53
 18  33   9  27   6  23   0  30
;
```

In the following statements, the **ods graphics on** specification enables ODS Graphics. PROC LIFETEST is invoked to compute the various life-table survival estimates, the median residual time, and their standard errors. The life-table method of computing estimates is requested by specifying METHOD=LT. The intervals are specified by the INTERVAL= option. Graphical display of the life-table survivor function estimate, negative log of the estimate, log of negative log of the estimate, estimated density function, and estimated hazard function are requested by the PLOTS= option. No tests for homogeneity are carried out because the data are not stratified.

```
ods graphics on;
proc lifetest data=Males  method=lt intervals=(0 to 15 by 1)
              plots=(s,ls,lls,h,p);
   time Years*Censored(1);
   freq Freq;
run;
ods graphics off;
```

Results of the life-table estimation are shown in Output 49.3.1. The five-year survival rate is 0.5193 with a standard error of 0.0103. The estimated median residual lifetime, which is 5.33 years initially, reaches a maximum of 6.34 years at the beginning of the second year and decreases gradually to a value lower than the initial 5.33 years at the beginning of the seventh year.

Output 49.3.1 Life-Table Survivor Function Estimate

Survival of Males with Angina Pectoris

The LIFETEST Procedure

Life Table Survival Estimates

Interval [Lower, Upper)		Number Failed	Number Censored	Effective Sample Size	Conditional Probability of Failure	Conditional Probability Standard Error
0	1	456	0	2418.0	0.1886	0.00796
1	2	226	39	1942.5	0.1163	0.00728
2	3	152	22	1686.0	0.0902	0.00698
3	4	171	23	1511.5	0.1131	0.00815
4	5	135	24	1317.0	0.1025	0.00836
5	6	125	107	1116.5	0.1120	0.00944
6	7	83	133	871.5	0.0952	0.00994
7	8	74	102	671.0	0.1103	0.0121
8	9	51	68	512.0	0.0996	0.0132
9	10	42	64	395.0	0.1063	0.0155
10	11	43	45	298.5	0.1441	0.0203
11	12	34	53	206.5	0.1646	0.0258
12	13	18	33	129.5	0.1390	0.0304
13	14	9	27	81.5	0.1104	0.0347
14	15	6	23	47.5	0.1263	0.0482
15	.	0	30	15.0	0	0

Interval [Lower, Upper)		Survival	Failure	Survival Standard Error	Median Residual Lifetime	Median Standard Error
0	1	1.0000	0	0	5.3313	0.1749
1	2	0.8114	0.1886	0.00796	6.2499	0.2001
2	3	0.7170	0.2830	0.00918	6.3432	0.2361
3	4	0.6524	0.3476	0.00973	6.2262	0.2361
4	5	0.5786	0.4214	0.0101	6.2185	0.1853
5	6	0.5193	0.4807	0.0103	5.9077	0.1806
6	7	0.4611	0.5389	0.0104	5.5962	0.1855
7	8	0.4172	0.5828	0.0105	5.1671	0.2713
8	9	0.3712	0.6288	0.0106	4.9421	0.2763
9	10	0.3342	0.6658	0.0107	4.8258	0.4141
10	11	0.2987	0.7013	0.0109	4.6888	0.4183
11	12	0.2557	0.7443	0.0111	.	.
12	13	0.2136	0.7864	0.0114	.	.
13	14	0.1839	0.8161	0.0118	.	.
14	15	0.1636	0.8364	0.0123	.	.

Output 49.3.1 *continued*

Survival of Males with Angina Pectoris

The LIFETEST Procedure

Interval [Lower,	Upper)	Survival	Failure	Survival Standard Error	Median Residual Lifetime	Median Standard Error
15	.	0.1429	0.8571	0.0133	.	.

Evaluated at the Midpoint of the Interval

Interval [Lower,	Upper)	PDF	PDF Standard Error	Hazard	Hazard Standard Error
0	1	0.1886	0.00796	0.208219	0.009698
1	2	0.0944	0.00598	0.123531	0.008201
2	3	0.0646	0.00507	0.09441	0.007649
3	4	0.0738	0.00543	0.119916	0.009154
4	5	0.0593	0.00495	0.108043	0.009285
5	6	0.0581	0.00503	0.118596	0.010589
6	7	0.0439	0.00469	0.1	0.010963
7	8	0.0460	0.00518	0.116719	0.013545
8	9	0.0370	0.00502	0.10483	0.014659
9	10	0.0355	0.00531	0.112299	0.017301
10	11	0.0430	0.00627	0.155235	0.023602
11	12	0.0421	0.00685	0.17942	0.030646
12	13	0.0297	0.00668	0.149378	0.03511
13	14	0.0203	0.00651	0.116883	0.038894
14	15	0.0207	0.00804	0.134831	0.054919
15	.	.	.	.	.

The breakdown of event and censored observation in the data is shown in Output 49.3.2. Note that 32.8% of the patients have withdrawn from the study.

Output 49.3.2 Summary of Censored and Event Observations

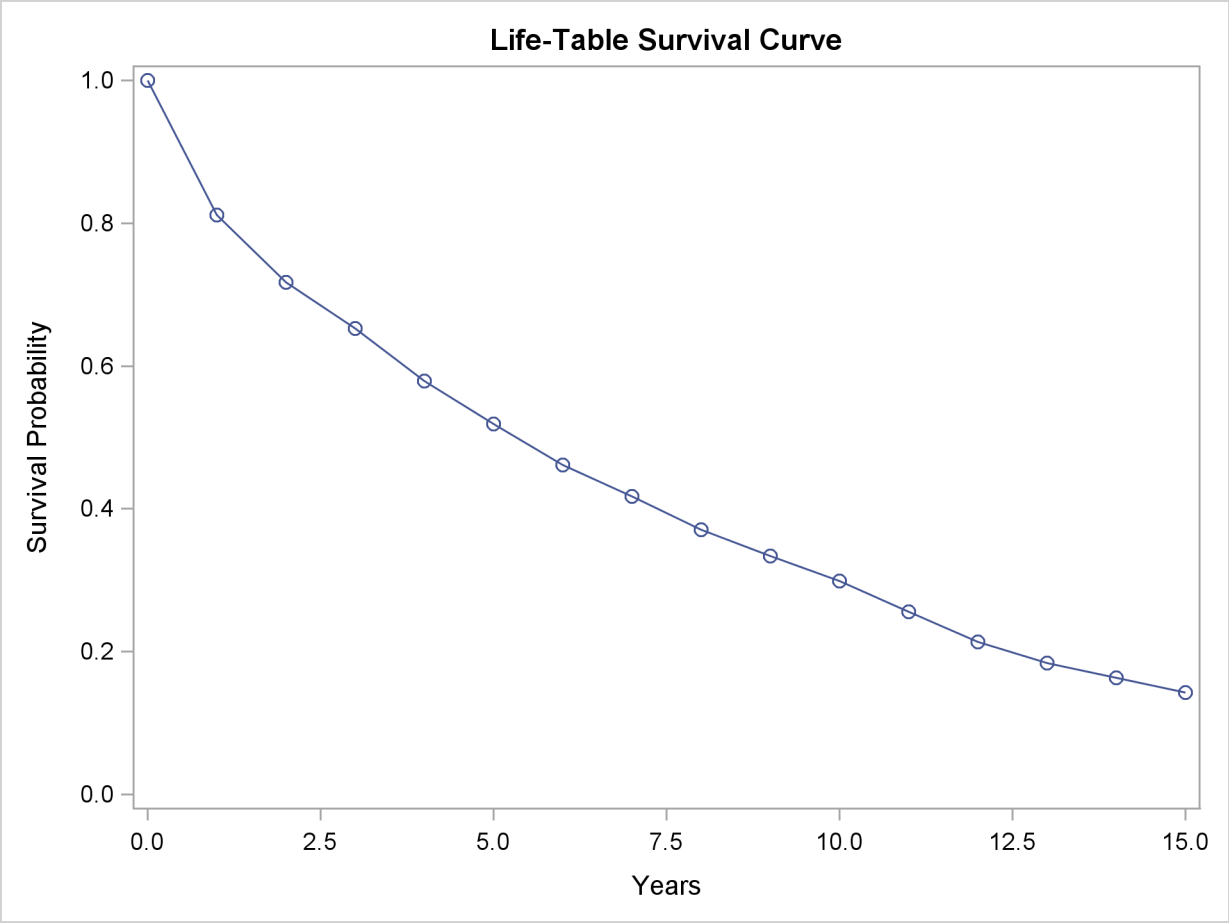
Summary of the Number of Censored and Uncensored Values

Total	Failed	Censored	Percent Censored
2418	1625	793	32.80

NOTE: 2 observations with invalid time, censoring, or frequency values were deleted.

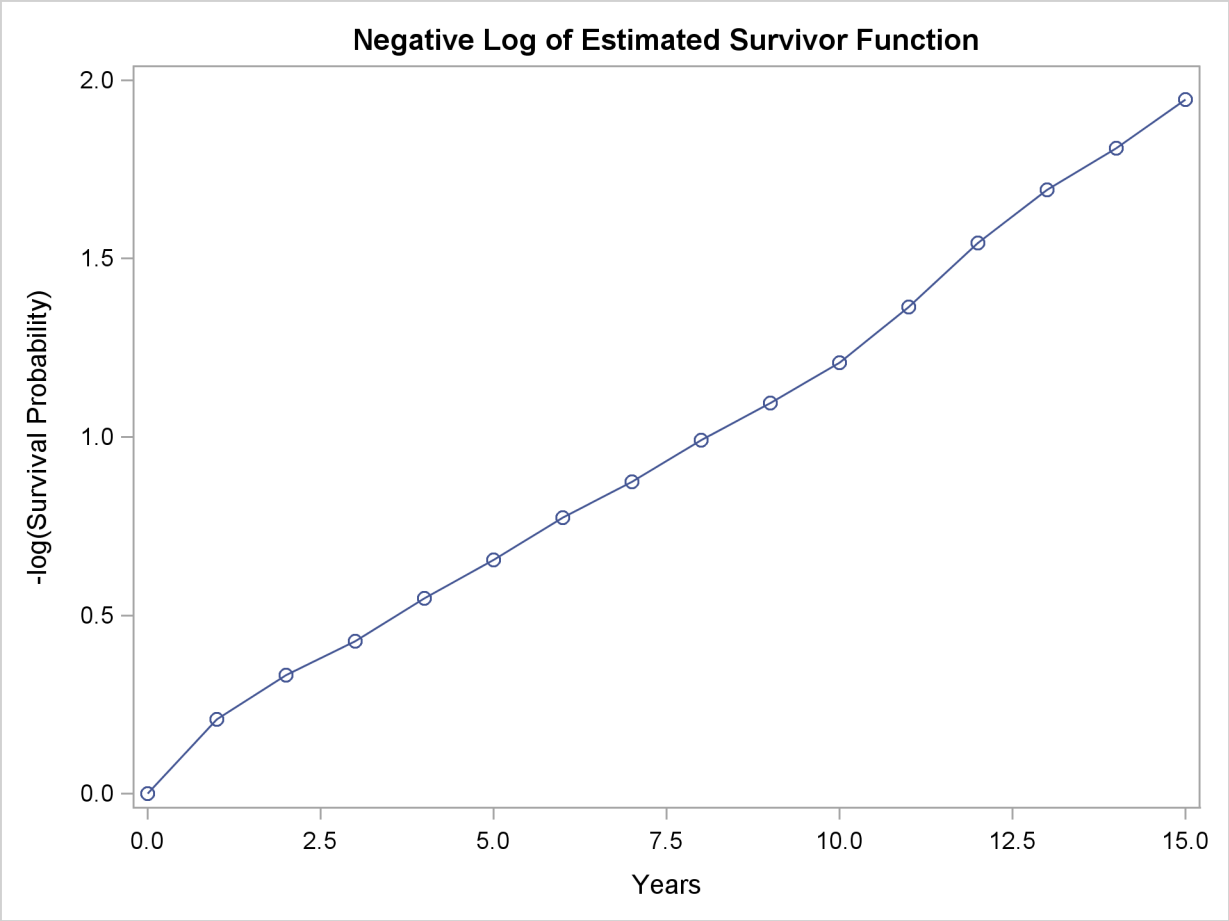
Output 49.3.3 displays the graph of the life-table survivor function estimate. The median survival time, read from the survivor function curve, is 5.33 years, and the 25th and 75th percentiles are 1.04 and 11.13 years, respectively.

Output 49.3.3 Life-Table Survivor Function Estimate



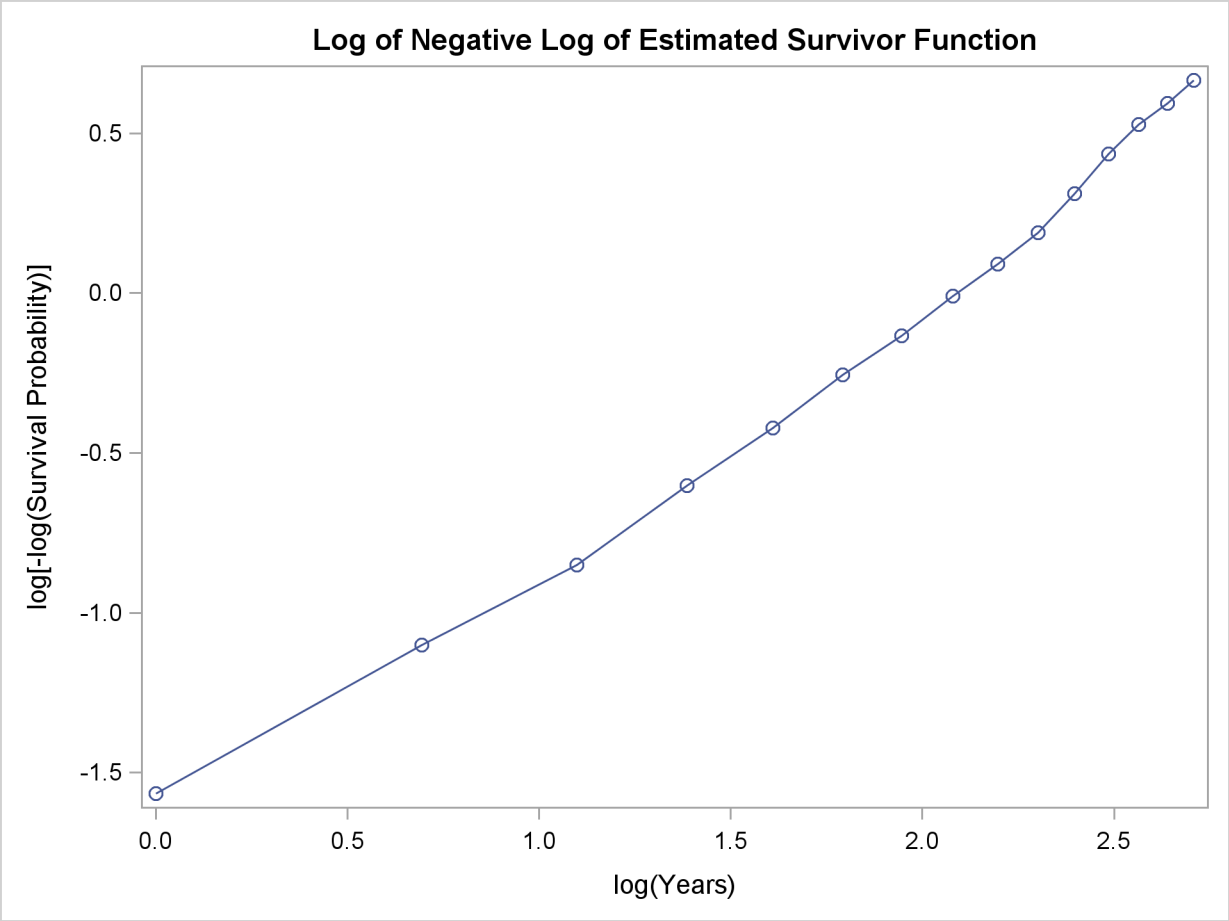
An exponential model might be appropriate for the survival of these male patients with angina pectoris since the curve of the negative log of the survivor function estimate versus the survival time (Output 49.3.4) approximates a straight line through the origin. Note that the graph of the log of the negative log of the survivor function estimate versus the log of time (Output 49.3.5) is practically a straight line.

Output 49.3.4 Negative Log of Survivor Function Estimate

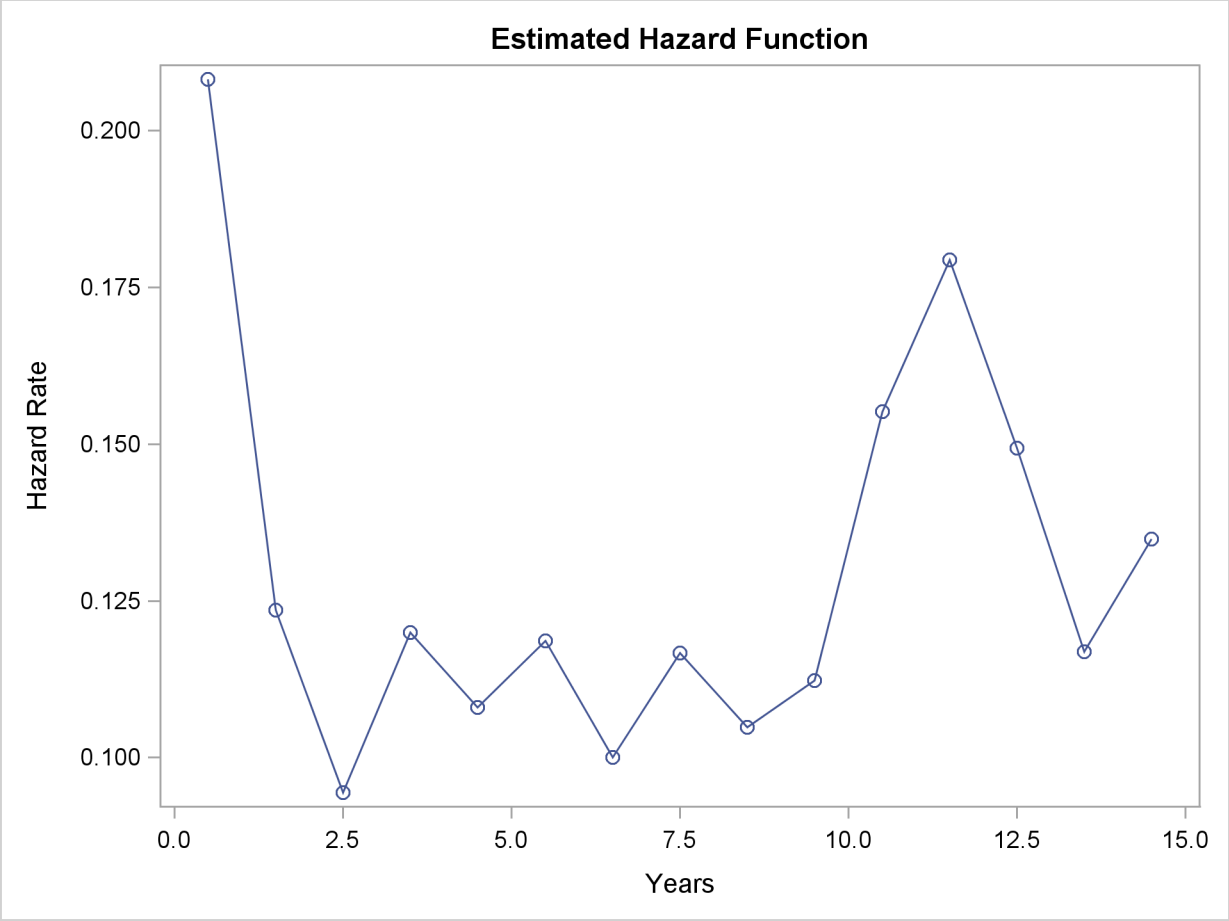


As discussed in Lee (1992), the graph of the estimated hazard function (Output 49.3.6) shows that the death rate is highest in the first year of diagnosis. From the end of the first year to the end of the tenth year, the death rate remains relatively constant, fluctuating between 0.09 and 0.12. The death rate is generally higher after the tenth year. This could indicate that a patient who has survived the first year has a better chance than a patient who has just been diagnosed. The profile of the median residual lifetimes also supports this interpretation.

Output 49.3.5 Log of Negative Log of Survivor Function Estimate

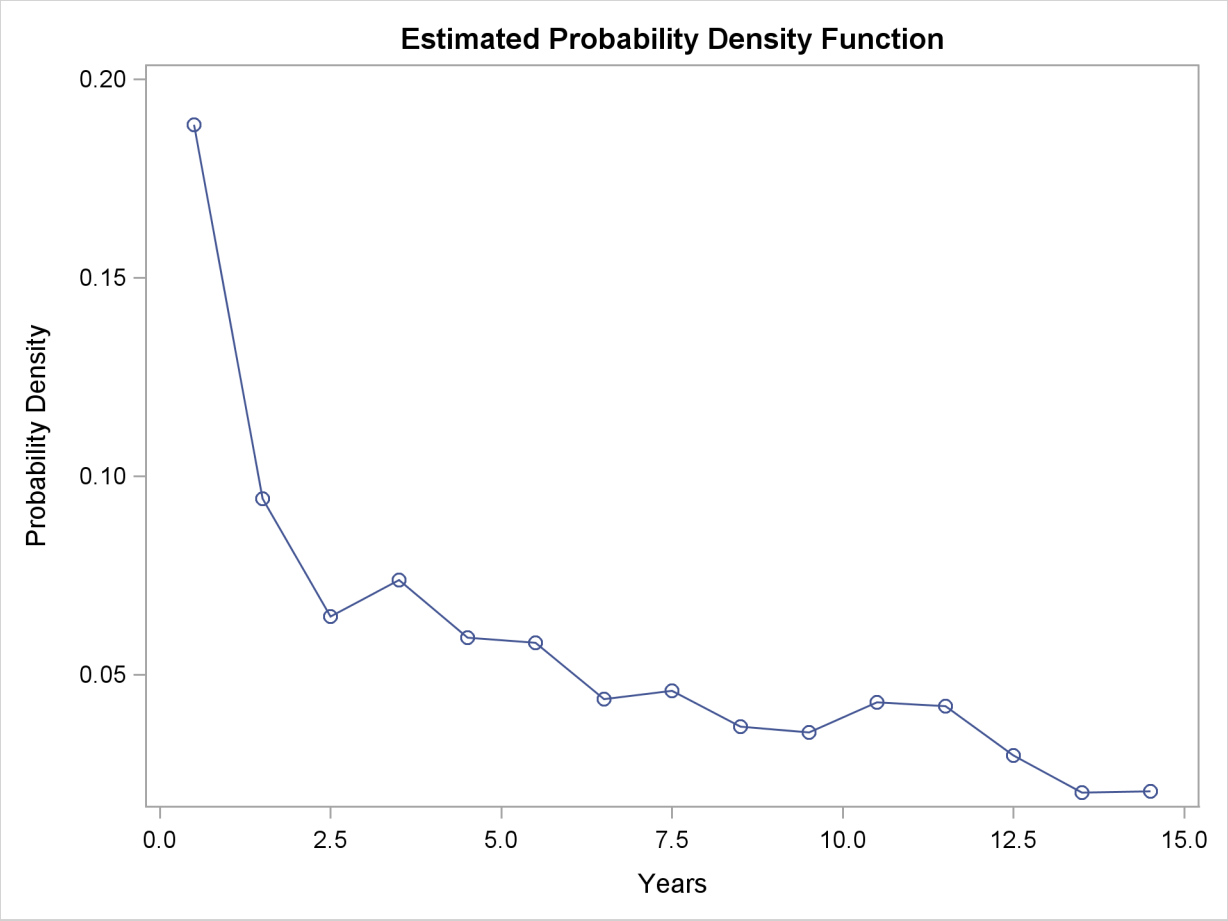


Output 49.3.6 Hazard Function Estimate



The density estimate is shown in (Output 49.3.7). Visually, it resembles the density function of an exponential distribution.

Output 49.3.7 Density Function Estimate



References

Andersen, P. K., Borgan, O., Gill, R. D., and Keiding, N. (1992), *Statistical Models Based on Counting Processes*, New York: Springer-Verlag.

Borgan, Ø. and Liestøl, K. (1990), "A Note on Confidence Interval and Bands for the Survival Curves Based on Transformations," *Scandinavian Journal of Statistics*, 18, 35–41.

Brookmeyer, R. and Crowley, J. (1982), "A Confidence Interval for the Median Survival Time," *Biometrics*, 38, 29–41.

Chung, C. F. (1986), *Formulae for Probabilities Associated with Wiener and Brownian Bridge Processes*, Technical Report 79, Laboratory for Research in Statistics and Probability, Ottawa, Canada: Carleton University.

Collett, D. (1994), *Modeling Survival Data in Medical Research*, London: Chapman & Hall.

Cox, D. R. and Oakes, D. (1984), *Analysis of Survival Data*, London: Chapman & Hall.

Edwards, D. and Berry, J. J. (1987), "The Efficiency of Simulation-Based Multiple Comparisons," *Biometrics*, 43, 913–928.

Elandt-Johnson, R. C. and Johnson, N. L. (1980), *Survival Models and Data Analysis*, New York: John Wiley & Sons.

Fleming, T. R. and Harrington, D. (1991), *Counting Processes and Survival Analysis*, New York: John Wiley & Sons.

Fleming, T. R. and Harrington, D. P. (1981), "A Class of Hypothesis Tests for One and Two Samples of Censored Survival Data," *Communications in Statistics*, 10, 763–794.

Gasser, T. and Müller, H. G. (1979), "Kernel Estimation of Regression Functions," in *Smoothing Techniques for Curve Estimation, Lecture Notes in Mathematics 757*, 23–68, Berlin: Springer-Verlag.

Hall, W. J. and Wellner, J. A. (1980), "Confidence Bands for a Survival Curve for Censored Data," *Biometrika 69*.

Harrington, D. P. and Fleming, T. R. (1982), "A Class of Rank Test Procedures for Censored Survival Data," *Biometrika*, 69, 133–143.

Hsu, J. C. (1992), "The Factor Analytic Approach to Simultaneous Inference in the General Linear Model," *Journal of Computational and Graphical Statistics*, 1, 151–168.

Kalbfleisch, J. D. and Prentice, R. L. (1980), *The Statistical Analysis of Failure Time Data*, New York: John Wiley & Sons.

Klein, J. P. and Moeschberger, M. L. (1997), *Survival Analysis: Techniques for Censored and Truncated Data*, New York: Springer-Verlag.

Kramer, C. Y. (1956), "Extension of Multiple Range Tests to Group Means with Unequal Numbers of Replications," *Biometrics*, 12, 307–310.

Lachin, J. M. (2000), *Biostatistical Methods: The Assessment of Relative Risks*, New York: John Wiley & Sons.

Lawless, J. F. (1982), *Statistical Methods and Methods for Lifetime Data*, New York: John Wiley & Sons.

Lee, E. T. (1992), *Statistical Methods for Survival Data Analysis*, Second Edition, New York: John Wiley & Sons.

Meeker, W. Q. and Escobar, L. A. (1998), *Statistical Methods for Reliability Data*, New York: John Wiley & Sons.

Miller, R. G. and Siegmund, D. (1982), "Maximally Selected Chi-Square Statistics," *Biometrics*, 1011–1016.

Nair, V. N. (1984), "Confidence Bands for Survival Functions with Censored Data: A Comparative Study," *Technometrics*, 26, 265–275.

Ramlau-Hansen, H. (1983a), "The Choice of a Kernel Function in the Graduation of Counting Process Intensities," *Scandinavian Actuarial Journal*, 165–182.

Ramlau-Hansen, H. (1983b), "Smoothing Counting Process Intensities by Means of Kernel Functions," *Annual of Statistics*, 11, 453–466.

Westfall, P. H., Tobias, R. D., Rom, D., Wolfinger, R. D., and Hochberg, Y. (1999), *Multiple Comparisons and Multiple Tests Using the SAS System*, Cary, NC: SAS Institute Inc.

Chapter 50

The LOESS Procedure

Contents

Overview: LOESS Procedure **3188**
Local Regression and the Loess Method 3188
Getting Started: LOESS Procedure **3189**
Scatter Plot Smoothing 3189
Syntax: LOESS Procedure **3201**
PROC LOESS Statement 3202
BY Statement 3207
ID Statement 3207
MODEL Statement 3208
SCORE Statement 3213
WEIGHT Statement 3214
Details: LOESS Procedure **3215**
Missing Values 3215
Output Data Sets 3215
Data Scaling 3216
Direct versus Interpolated Fitting 3217
kd Trees and Blending 3218
Local Weighting 3218
Iterative Reweighting 3219
Specifying the Local Polynomials 3219
Smoothing Matrix 3220
Model Degrees of Freedom 3220
Statistical Inference and Lookup Degrees of Freedom 3220
Automatic Smoothing Parameter Selection 3221
Sparse and Approximate Degrees of Freedom Computation 3224
Scoring Data Sets 3225
ODS Table Names 3225
ODS Graphics 3226
Examples: LOESS Procedure **3227**
Example 50.1: Engine Exhaust Emissions 3227
Example 50.2: Sulfate Deposits in the U.S. for 1990 3233
Example 50.3: Catalyst Experiment 3237
Example 50.4: El Niño Southern Oscillation 3245
References **3254**

Overview: LOESS Procedure

The LOESS procedure implements a nonparametric method for estimating regression surfaces pioneered by Cleveland, Devlin, and Grosse (1988), Cleveland and Grosse (1991), and Cleveland, Grosse, and Shyu (1992). The LOESS procedure allows great flexibility because no assumptions about the parametric form of the regression surface are needed.

The SAS System provides many regression procedures such as the GLM, REG, and NLIN procedures for situations in which you can specify a reasonable parametric model for the regression surface. You can use the LOESS procedure for situations in which you do not know a suitable parametric form of the regression surface. Furthermore, the LOESS procedure is suitable when there are outliers in the data and a robust fitting method is necessary.

The main features of the LOESS procedure are as follows:

- fits nonparametric models
- supports the use of multidimensional data
- supports multiple dependent variables
- supports both direct and interpolated fitting that uses kd trees
- performs statistical inference
- performs automatic smoothing parameter selection
- performs iterative reweighting to provide robust fitting when there are outliers in the data
- supports graphical displays produced through ODS Graphics

Local Regression and the Loess Method

Assume that for $i = 1$ to n, the ith measurement y_i of the response y and the corresponding measurement x_i of the vector x of p predictors are related by

$$y_i = g(x_i) + \epsilon_i$$

where g is the regression function and ϵ_i is a random error. The idea of local regression is that at a predictor x, the regression function $g(x)$ can be locally approximated by the value of a function in some specified parametric class. Such a local approximation is obtained by fitting a regression surface to the data points within a chosen neighborhood of the point x.

In the loess method, weighted least squares is used to fit linear or quadratic functions of the predictors at the centers of neighborhoods. The radius of each neighborhood is chosen so that the neighborhood contains a specified percentage of the data points. The fraction of the data, called the *smoothing parameter,* in each local neighborhood controls the smoothness of the estimated surface.

Data points in a given local neighborhood are weighted by a smooth decreasing function of their distance from the center of the neighborhood.

In a direct implementation, such fitting is done at each point at which the regression surface is to be estimated. A much faster computational procedure is to perform such local fitting at a selected sample of points in predictor space and then to blend these local polynomials to obtain a regression surface.

You can use the LOESS procedure to perform statistical inference provided that the error distribution satisfies some basic assumptions. In particular, such analysis is appropriate when the ϵ_i are i.i.d. normal random variables with mean 0. By using the iterative reweighting, the LOESS procedure can also provide statistical inference when the error distribution is symmetric but not necessarily normal. Furthermore, by doing iterative reweighting, you can use the LOESS procedure to perform robust fitting in the presence of outliers in the data.

While all output of the LOESS procedure can be optionally displayed, most often the LOESS procedure is used to produce output data sets that will be viewed and manipulated by other SAS procedures. PROC LOESS uses the Output Delivery System (ODS) to place results in output data sets. This is a departure from older SAS procedures that provide OUTPUT statements to create SAS data sets from analysis results.

Getting Started: LOESS Procedure

Scatter Plot Smoothing

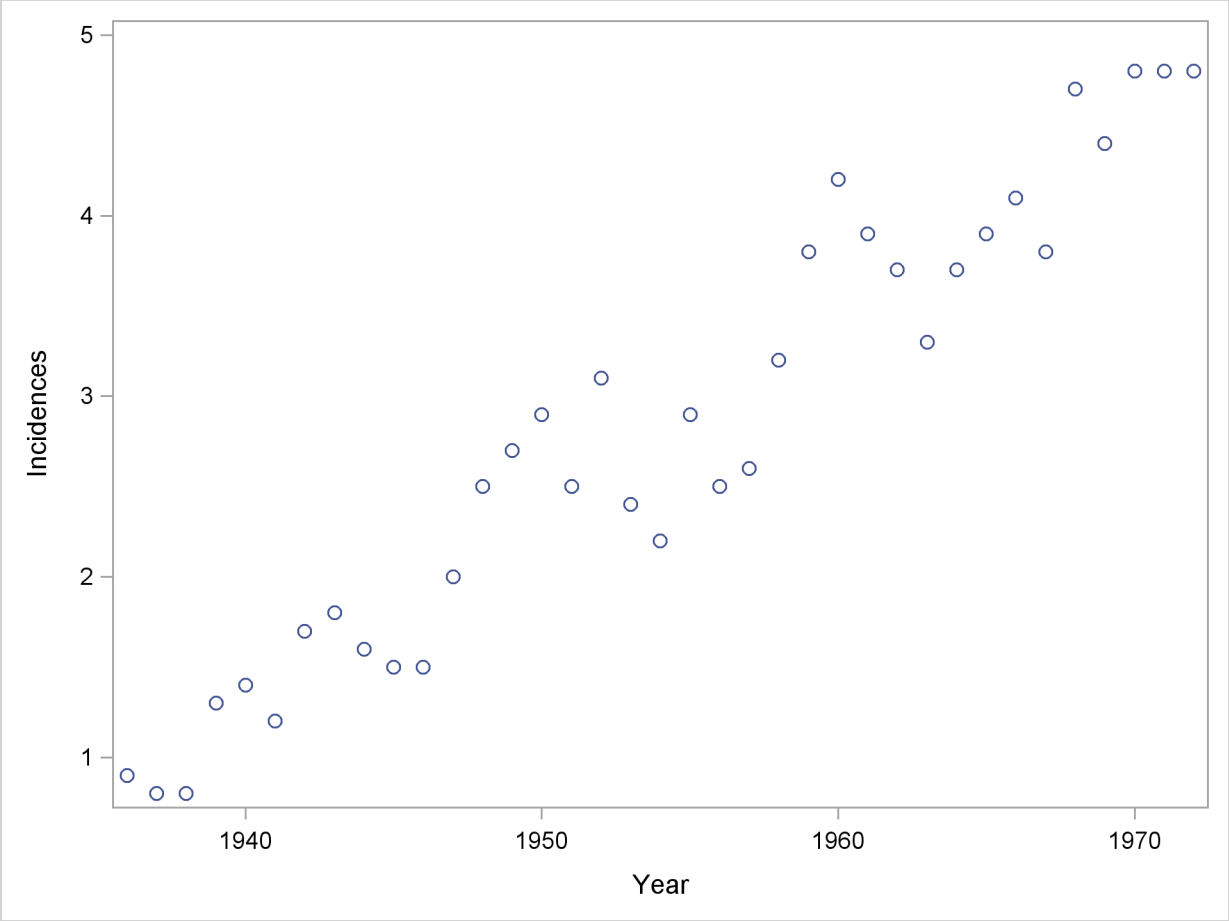
The following data from the Connecticut Tumor Registry presents age-adjusted numbers of melanoma incidences per 100,000 people for the 37 years from 1936 to 1972 (Houghton, Flannery, and Viola 1980).

```
data Melanoma;
   input  Year Incidences @@;
   format Year d4.0;
datalines;
1936    0.9   1937   0.8  1938   0.8  1939   1.3
1940    1.4   1941   1.2  1942   1.7  1943   1.8
1944    1.6   1945   1.5  1946   1.5  1947   2.0
1948    2.5   1949   2.7  1950   2.9  1951   2.5
1952    3.1   1953   2.4  1954   2.2  1955   2.9
1956    2.5   1957   2.6  1958   3.2  1959   3.8
1960    4.2   1961   3.9  1962   3.7  1963   3.3
1964    3.7   1965   3.9  1966   4.1  1967   3.8
1968    4.7   1969   4.4  1970   4.8  1971   4.8
1972    4.8
;
```

The following PROC SGPLOT statements produce the simple scatter plot of these data displayed in Figure 50.1.

```
proc sgplot data=Melanoma;
   scatter y=Incidences x=Year;
run;
```

Figure 50.1 Scatter Plot of the Melanoma Data

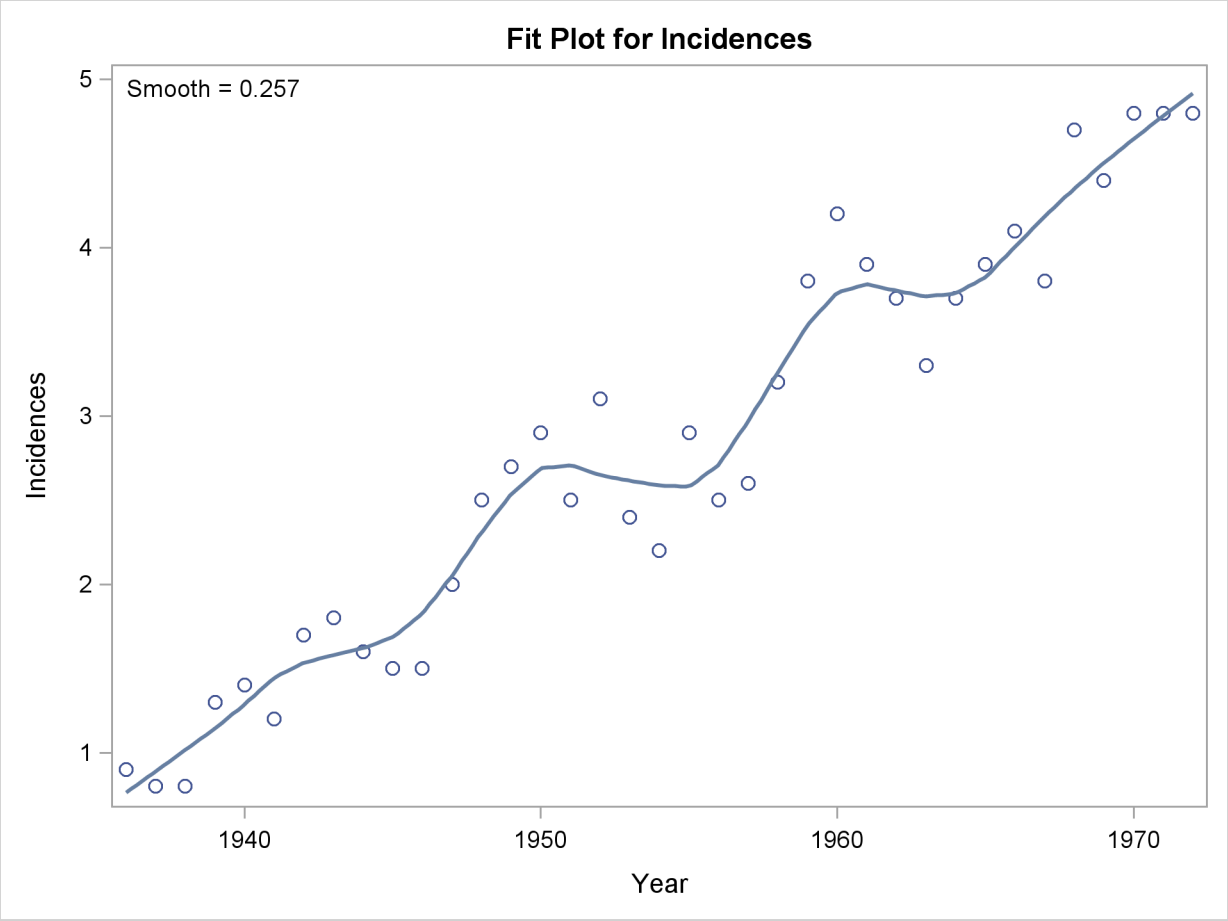


Suppose that you want to smooth the response variable Incidences as a function of the variable Year. The following PROC LOESS statements request this analysis with the default settings:

```
ods graphics on;

proc loess data=Melanoma;
   model Incidences=Year;
run;
```

You use the PROC LOESS statement to invoke the procedure and specify the data set. The MODEL statement names the dependent and independent variables.

Figure 50.2 Default Loess Fit for the Melanoma data



When you enable graphics with an ODS GRAPHICS statement, PROC LOESS produces several default plots. Figure 50.2 shows the "Fit Plot" that overlays the loess fit on a scatter plot of the data. You can see that the loess fit captures the increasing trend in the data as well as the periodic pattern in the data, which is related to an 11-year sunspot activity cycle.

Figure 50.3 Fit Summary

The LOESS Procedure
Selected Smoothing Parameter: 0.257
Dependent Variable: Incidences

Fit Summary

Fit Method	kd Tree
Blending	Linear
Number of Observations	37
Number of Fitting Points	37
kd Tree Bucket Size	1
Degree of Local Polynomials	1
Smoothing Parameter	0.25676
Points in Local Neighborhood	9
Residual Sum of Squares	2.03105
Trace[L]	8.62243
GCV	0.00252
AICC	-1.17277

Figure 50.3 shows the "Fit Summary" table. This table details the settings used and provides statistics about the fit that is produced. You can see that smoothing parameter value for this loess fit is 0.257. This smoothing parameter determines the fraction of the data in each local neighborhood. In this example, there are 37 data points and so the smoothing parameter value of 0.257 yields local neighborhoods containing 9 observations.

Figure 50.4 Smoothing Parameter Selection

Optimal Smoothing
Criterion

AICC	Smoothing Parameter
-1.17277	0.25676

The "Smoothing Criterion" table provides information about how this smoothing parameter value is selected. The default method implemented in PROC LOESS chooses the smoothing parameter that minimizes the AICC criterion (Hurvich, Simonoff, and Tsai 1998) that strikes a balance between the residual sum of squares and the complexity of the fit.

You use options in the MODEL statement to change the default settings and request optionally displayed tables. For example, the following statements request that the "Model Summary" and "Output Statistics" tables be included in the displayed output. By default, these tables are not displayed.

```
proc loess data=Melanoma;
   model Incidences=Year / details(ModelSummary OutputStatistics);
run;
```

Figure 50.5 Model Summary Table

The LOESS Procedure
Dependent Variable: Incidences

Model Summary

Smoothing Parameter	Local Points	Residual SS	GCV	AICC
0.41892	15	3.42229	0.00339	−0.96252
0.68919	25	4.05838	0.00359	−0.93459
0.31081	11	2.51054	0.00279	−1.12034
0.20270	7	1.58513	0.00239	−1.12221
0.17568	6	1.56896	0.00241	−1.09706
0.28378	10	2.50487	0.00282	−1.10402
0.20270	7	1.58513	0.00239	−1.12221
0.25676	9	2.03105	0.00252	−1.17277
0.22973	8	2.02965	0.00256	−1.15145
0.25676	9	2.03105	0.00252	−1.17277

The "Model Summary" table shown in Figure 50.5 provides information about all the models that PROC LOESS evaluated in choosing the smoothing parameter value.

Figure 50.6 AICC Criterion by Smoothing Parameter

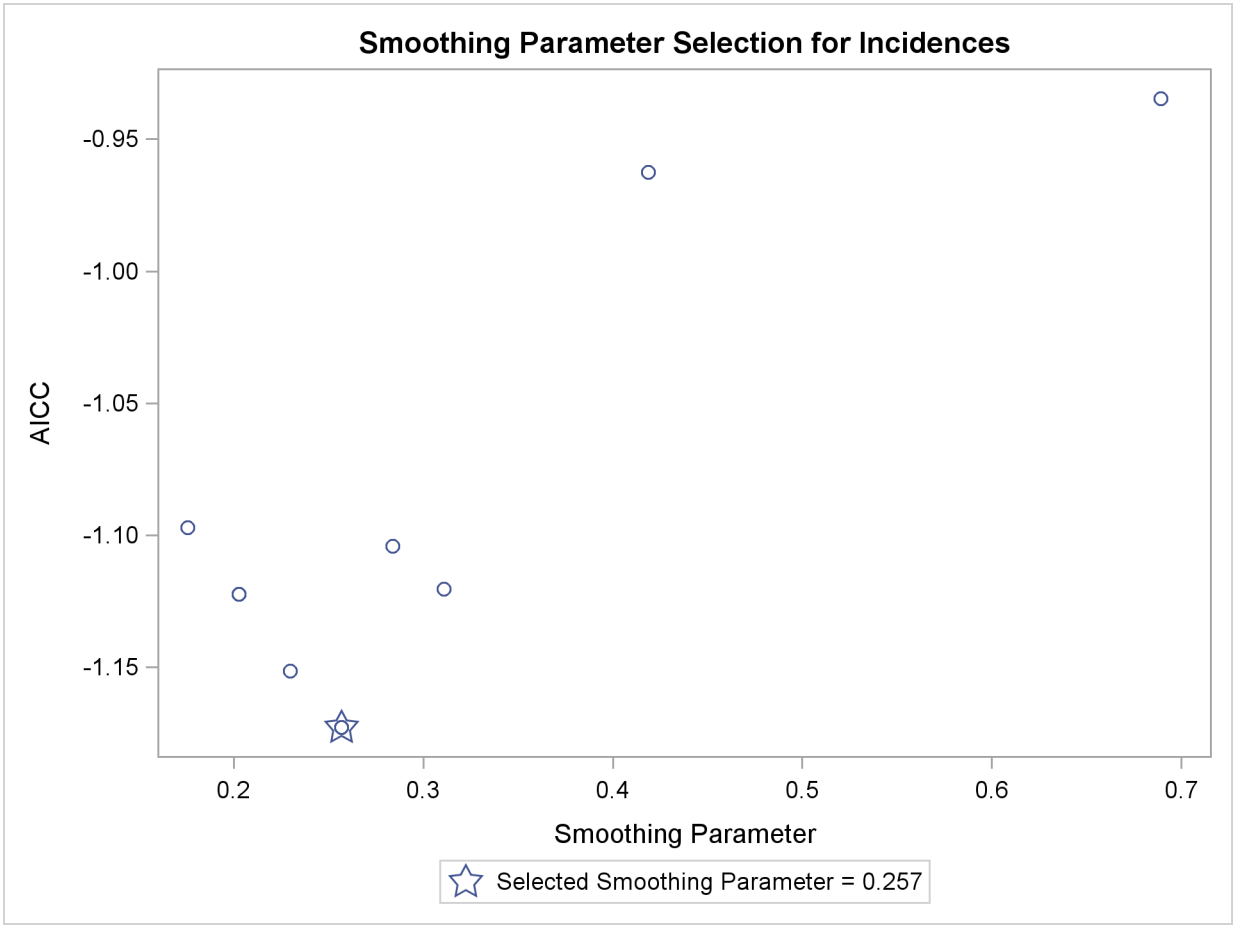


Figure 50.6 shows the "Criterion Plot" that provides a graphical display of the smoothing parameter selection process.

Figure 50.7 Output Statistics

The LOESS Procedure
Selected Smoothing Parameter: 0.257
Dependent Variable: Incidences

Output Statistics

Obs	Year	Incidences	Predicted Incidences	Residual
1	1936	0.90000	0.76235	0.13765
2	1937	0.80000	0.88992	-0.08992
3	1938	0.80000	1.01764	-0.21764
4	1939	1.30000	1.14303	0.15697
5	1940	1.40000	1.28654	0.11346
6	1941	1.20000	1.44528	-0.24528
7	1942	1.70000	1.53482	0.16518
8	1943	1.80000	1.57895	0.22105
9	1944	1.60000	1.62058	-0.02058
10	1945	1.50000	1.68627	-0.18627
11	1946	1.50000	1.82449	-0.32449
12	1947	2.00000	2.04976	-0.04976
13	1948	2.50000	2.30981	0.19019
14	1949	2.70000	2.53653	0.16347
15	1950	2.90000	2.68921	0.21079
16	1951	2.50000	2.70779	-0.20779
17	1952	3.10000	2.64837	0.45163
18	1953	2.40000	2.61468	-0.21468
19	1954	2.20000	2.58792	-0.38792
20	1955	2.90000	2.57877	0.32123
21	1956	2.50000	2.71078	-0.21078
22	1957	2.60000	2.96981	-0.36981
23	1958	3.20000	3.26005	-0.06005
24	1959	3.80000	3.54143	0.25857
25	1960	4.20000	3.73482	0.46518
26	1961	3.90000	3.78186	0.11814
27	1962	3.70000	3.74362	-0.04362
28	1963	3.30000	3.70904	-0.40904
29	1964	3.70000	3.72917	-0.02917
30	1965	3.90000	3.82382	0.07618
31	1966	4.10000	4.00515	0.09485
32	1967	3.80000	4.18573	-0.38573
33	1968	4.70000	4.35152	0.34848
34	1969	4.40000	4.50284	-0.10284
35	1970	4.80000	4.64413	0.15587
36	1971	4.80000	4.78291	0.01709
37	1972	4.80000	4.91602	-0.11602

Figure 50.7 show the "Output Statistics" table that contains the predicted loess fit value at each observation in the input data set.

Although the default method for selecting the smoothing parameter value is often satisfactory, it is often a good practice to examine how the loess fit varies with the smoothing parameter. In some cases, fits with different smoothing parameters might reveal important features of the data that cannot be discerned by looking at a fit with just a single "best" smoothing parameter. Example 50.4

provides such an example. You can produce the loess fits for a range of smoothing parameters by using the SMOOTH= option in the MODEL statement as follows:

```
proc loess data=Melanoma;
   model Incidences=Year/smooth=0.1 0.25 0.4 0.6 residual;
   ods output OutputStatistics=Results;
run;
```

The RESIDUAL option causes the residuals to be added to the "Output Statistics" table. Note that, even if you do not specify the DETAILS option in the MODEL statement to request the display of the "Output Statistics" table, you can use an ODS OUTPUT statement to output this and other optionally displayed tables as data sets.

PROC PRINT displays the first five observations of the Results data set:

```
proc print data=Results(obs=5);
  id obs;
run;
```

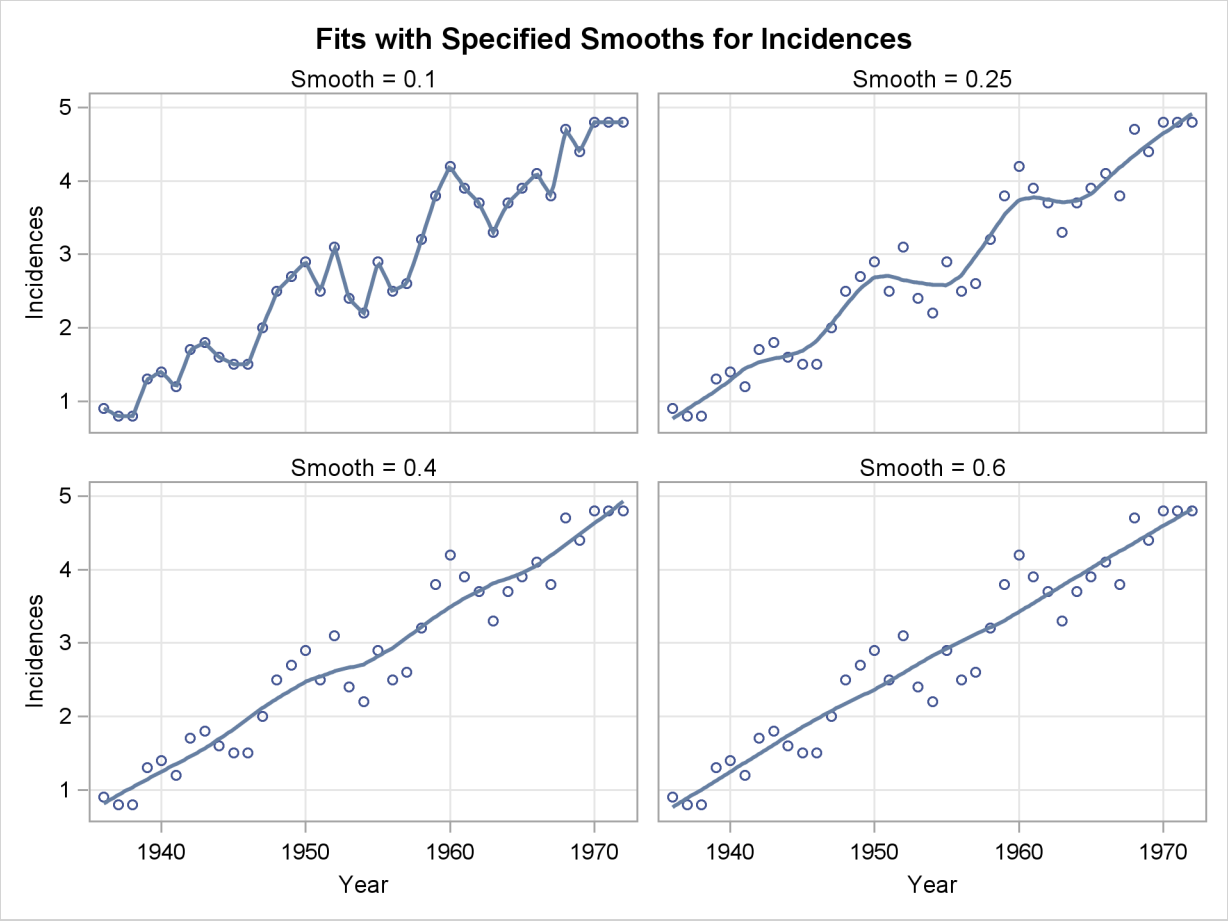
Figure 50.8 PROC PRINT Output of the Results Data Set

Obs	Smoothing Parameter	Year	Dep Var	Pred	Residual
1	0.1	1936	0.9	0.90000	0
2	0.1	1937	0.8	0.80000	0
3	0.1	1938	0.8	0.80000	0
4	0.1	1939	1.3	1.30000	0
5	0.1	1940	1.4	1.40000	0

Note that the fits for all the smoothing parameters are placed in single data set. A variable named SmoothingParameter that you use to distinguish each fit is included in this data set.

When you specify a list of smoothing parameters for a model and you have enabled ODS Graphics, PROC LOESS produces a panel containing up to six plots that show the fit obtained for each value of the smoothing parameter that you specify. If you specify more than six smoothing values, then multiple panels are produced. For each regressor, PROC LOESS also produces panels of the residuals versus each regressor by the smoothing parameters that you specify.

Figure 50.9 Loess Fits for a Range of Smoothing Parameters



If you examine the plots in Figure 50.9, you see that a visually reasonable fit is obtained with smoothing parameter values of 0.25. With smoothing parameter value 0.1, there is gross overfitting in the sense that the original data are exactly interpolated. When the smoothing parameter value is 0.4, you obtain an overly smooth fit where the contribution of the sunspot cycle has been mostly averaged away. At smoothing parameter value 0.6 the fit shows just the increasing trend in the data.

It is also instructive to look at scatter plots of the residuals for each of the fits. These are also produced by default by PROC LOESS when ODS Graphics is enabled.

Figure 50.10 Residuals of Loess Fits for a Range of Smoothing Parameters

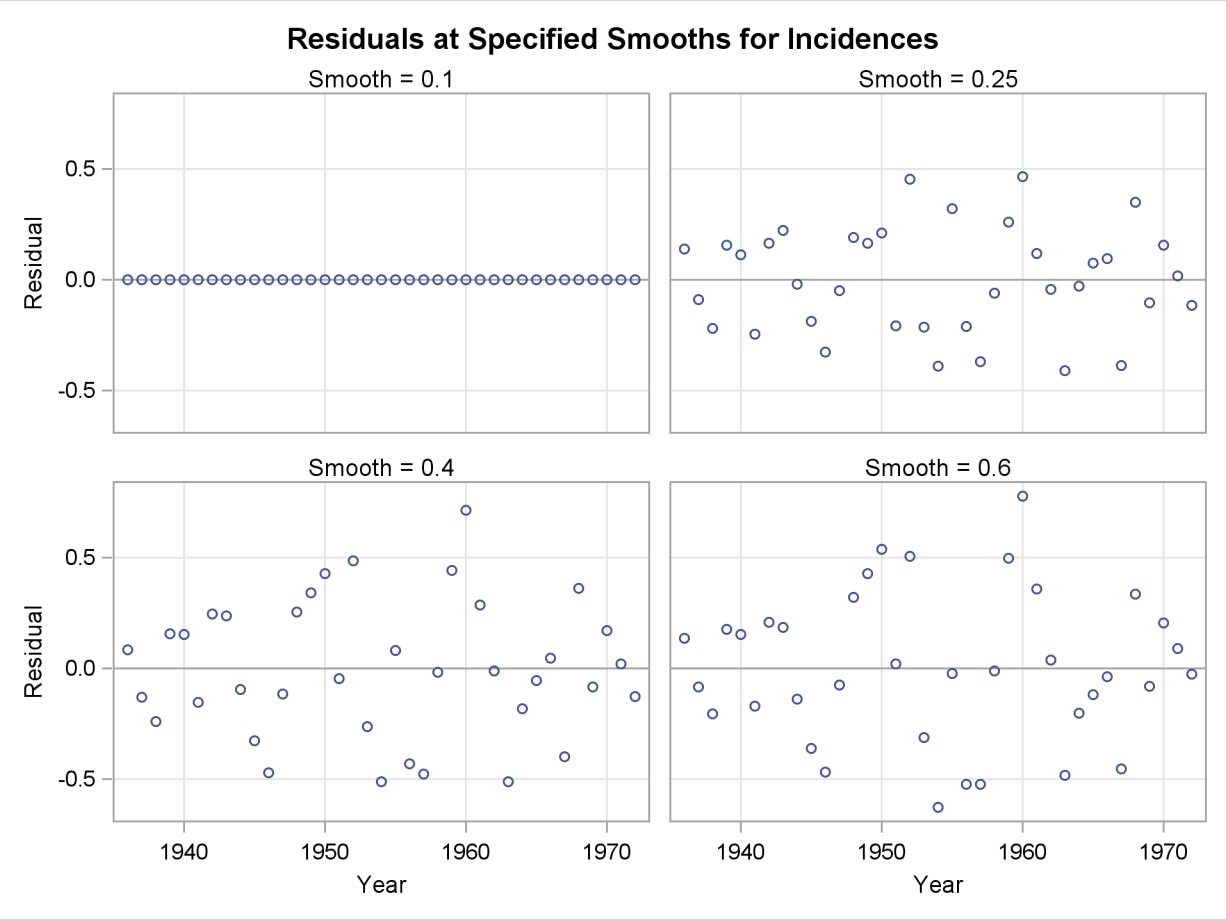
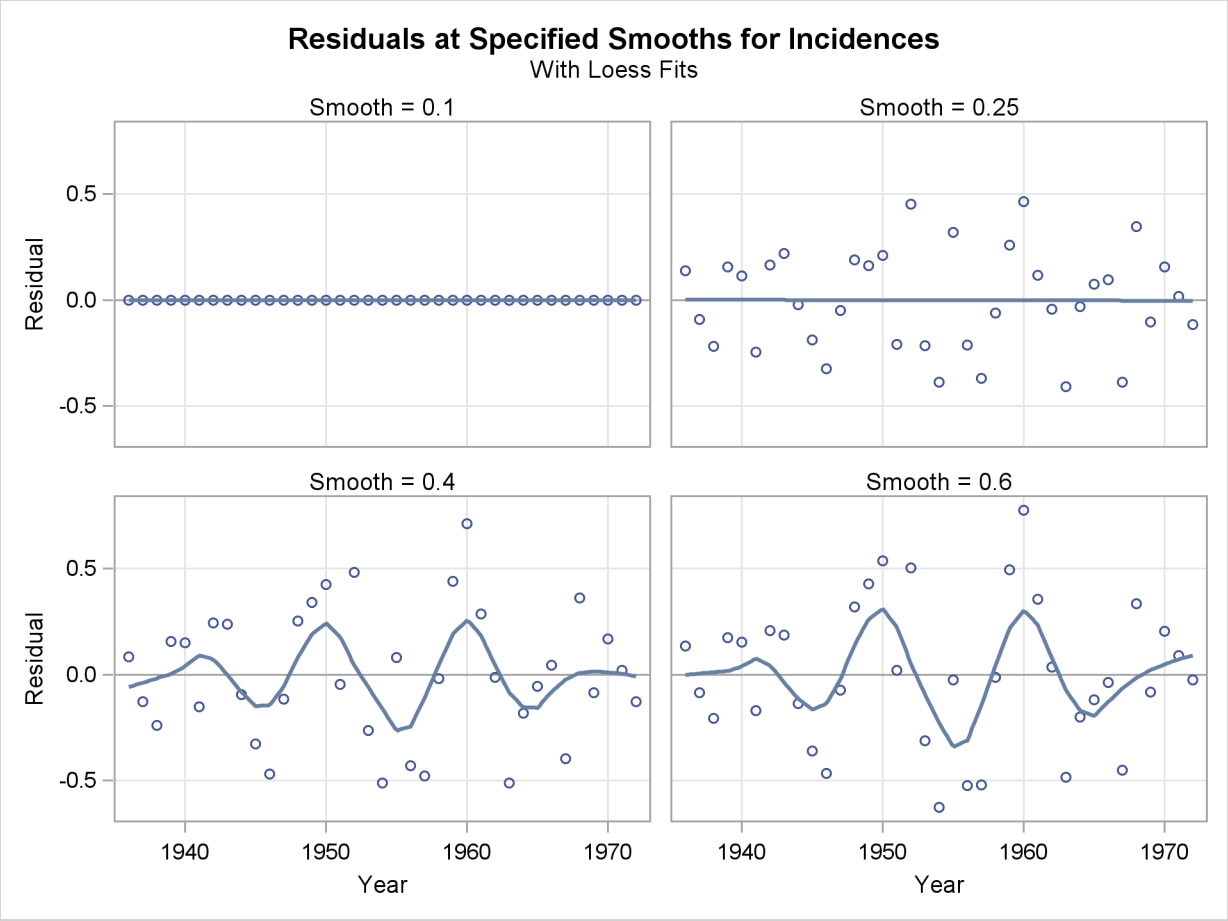


Figure 50.10 shows a scatter plot of the residuals by year for each smoothing parameter value. One way to discern patterns in these residuals is to superimpose a loess fit on each plot in the panel. You request loess fits on the residual plots in this panel by specifying the SMOOTH= suboption of the PLOTS=RESIDUALSBYSMOOTH option in the PROC LOESS statement. Note that the loess fits that are displayed on each of the residual plots are obtained independently of the loess fit that produces these residuals. The following statements show how you do this for the Melanoma data.

```
proc loess data=Melanoma plots=ResidualsBySmooth(smooth);
   model Incidences=Year/smooth=0.1 0.25 0.4 0.6;
run;
```

Figure 50.11 Residuals with Superimposed Loess Fits



The loess fits shown on the plots in Figure 50.11 help confirm the conclusions obtained when you look at Figure 50.9. Note that residuals for smoothing parameter value 0.25 do not exhibit any pattern, confirming that at this value the loess fit of the melanoma data has successfully modeled the variation in this data. By contrast, the residuals for the fit with smoothing parameter 0.6 retain the variation caused by the sunspot cycle.

The examination of the fits and residuals obtained with a range of smoothing parameter values confirms that the value of 0.257 that PROC LOESS selects automatically is appropriate for these data. The next step in this analysis is to examine fit diagnostics and produce confidence limit for the fit. If you enable ODS Graphics, then a panel of fit diagnostics is produced. Furthermore, you can request prediction confidence limits by adding the CLM option in the MODEL statement. By default 95% limits are produced, but you can use the ALPHA= option in the MODEL statement to change the significance level. The following statements request 90% confidence limits.

```
proc loess data=Melanoma;
   model Incidences=Year/clm alpha=0.1;
run;

ods graphics off;
```

Figure 50.12 Fit Diagnostics

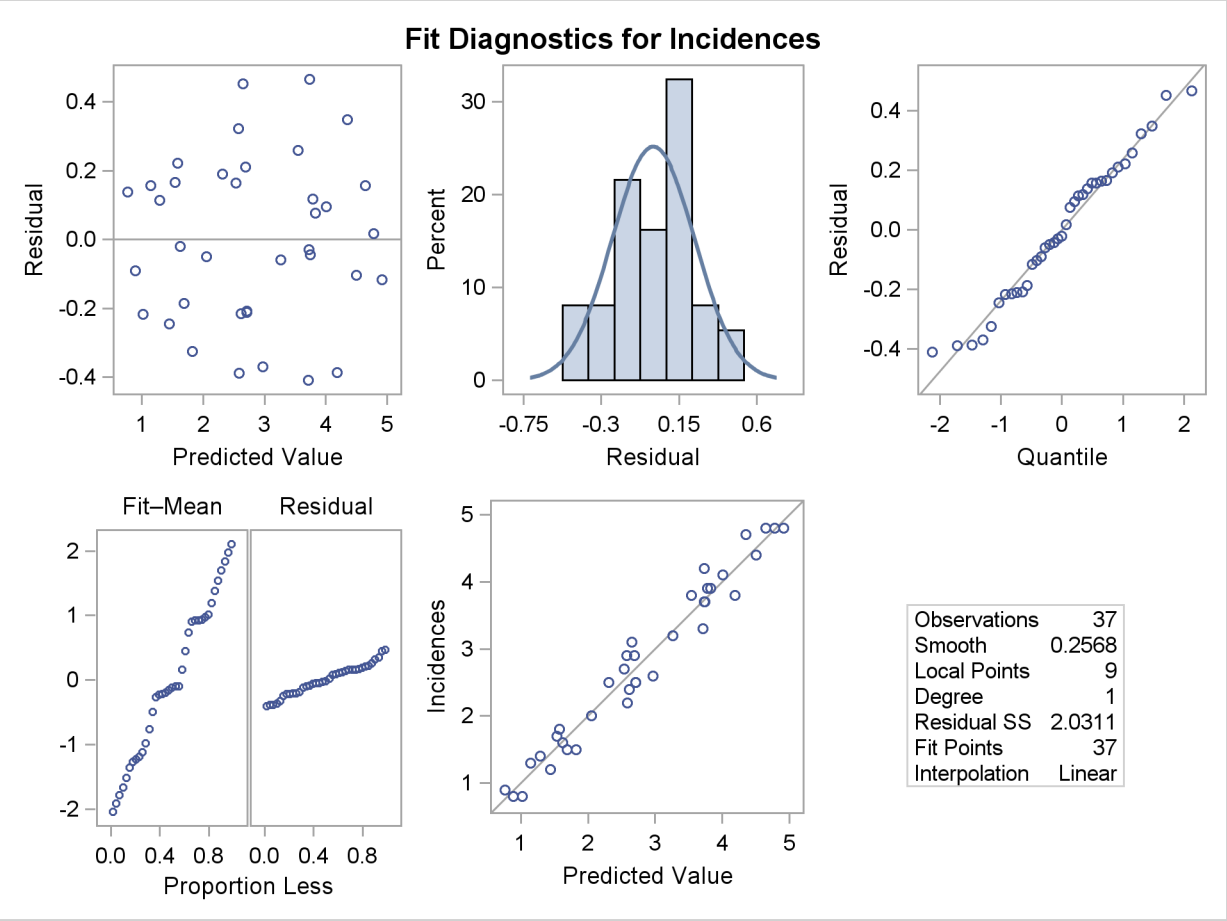
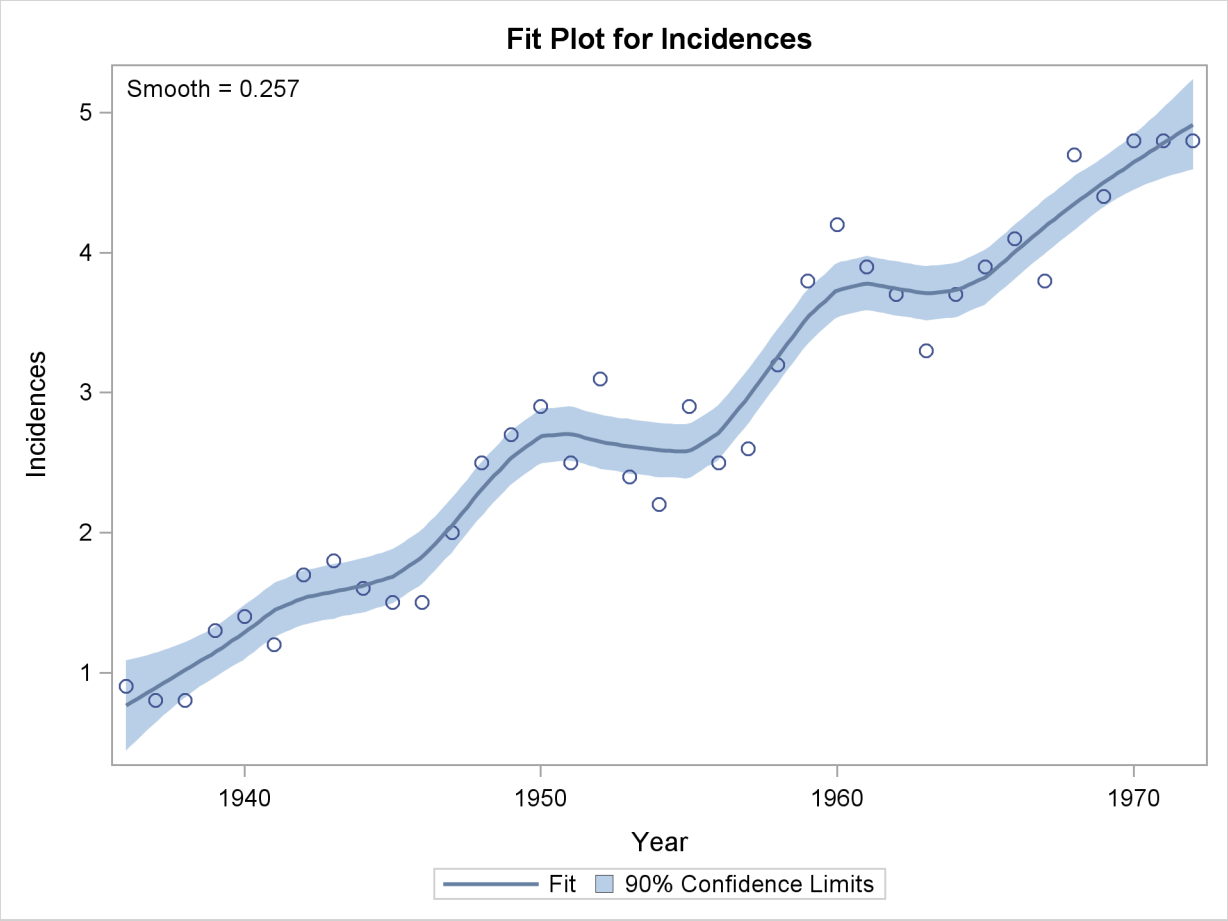


Figure 50.12 shows the fit diagnostics panel. The histogram of the residuals with overlaid normal density estimator and the normal quantile plot show that the residuals do exhibit some small departure from normality. The "Residual-Fit" spread plot shows that the spread in the centered fit is much wider that the spread in the residuals. This indicates that the fit has accounted for most of the variation in the incidences of melanoma in this data. This conclusion is supported by the absence of any clear pattern in the scatter plot of residuals by predicted values and the closeness of the points to the 45-degree reference line in the plot of observed by predicted values.

Figure 50.13 Loess Fit of Melanoma Data with 90% Confidence Limits



Finally, Figure 50.13 shows the selected loess fit with 90% confidence limits.

Syntax: LOESS Procedure

The following statements are available in PROC LOESS:

PROC LOESS < *DATA=SAS-data-set* > ;
MODEL *dependents=regressors* < / *options* > ;
ID *variables* ;
BY *variables* ;
WEIGHT *variable* ;
SCORE *DATA=SAS-data-set* < *ID=(variable list)* > < / *options* > ;

The PROC LOESS and MODEL statements are required. The BY, WEIGHT, and ID statements are optional. The SCORE statement is optional, and more than one SCORE statement can be used.

The statements used with the LOESS procedure, in addition to the PROC LOESS statement, are as follows.

BY	specifies variables to define subgroups for the analysis.
ID	names variables to identify observations in the displayed output.
MODEL	specifies the dependent and independent variables in the loess model, details and parameters for the computational algorithm, and the required output.
SCORE	specifies a data set containing observations to be scored.
WEIGHT	declares a variable to weight observations.

PROC LOESS Statement

PROC LOESS < *options* > **;**

The PROC LOESS statement is required. You can specify the following options in the PROC LOESS statement:

DATA=*SAS-data-set*
names the SAS data set to be used by PROC LOESS. If the DATA= option is not specified, PROC LOESS uses the most recently created SAS data set.

PLOTS < **(***global-plot-options***)** > < **=** *plot-request* < *(options)* > >

PLOTS < **(***global-plot-options***)** > < **=** **(***plot-request* < *(options)* > < **...** *plot-request* < *(options)* > **>)** >
controls the plots produced through ODS Graphics. When you specify only one plot request, you can omit the parentheses around the plot request. Here are some examples:

```
plots=none
plots=residuals(smooth)
plots(unpack)=diagnostics
plots(only)=(fit residualHistogram)
```

You must enable ODS Graphics before requesting plots as shown in the following example. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

```
ods graphics on;

proc loess;
   model y = x;
run;

proc loess plots=all;
   model y = x;
run;

ods graphics off;
```

The first PROC LOESS step does not specify the PLOTS= option, so the default plots comprising a panel of fit diagnostics, a fit plot, and a plot of residuals by x are produced. The PLOTS=ALL option in the second PROC LOESS step produces the default plots together with individual plots of the plots in the panel of fit diagnostics.

If you have enabled ODS Graphics but do not specify the PLOTS= option, then PROC LOESS produces a default set of plots. The following table lists the default set of plots produced.

Table 50.1 Default ODS Graphics Produced

Plot	Conditional On
ContourFitPanel	SMOOTH= option specified in the MODEL statement
ContourFit	Model with two regressors
CriterionPlot	Smoothing parameter selection performed
DiagnosticsPanel	Unconditional
ResidualsBySmooth	SMOOTH= option specified in the MODEL statement
ResidualPanel	Unconditional
FitPanel	SMOOTH= option specified in the MODEL statement
FitPlot	Model with one regressor
ScorePlot	One or more SCORE statements and a model with one regressor

For models with multiple dependent variables, separate plots are produced for each dependent variable. For models where multiple smoothing parameters are requested with the SMOOTH= option in the MODEL statement and smoothing parameter value selection is not requested, separate plots are produced for each smoothing parameter. If smoothing parameter value selection is requested with the SELECT= option in the MODEL statement, then the plots are produced for the selected model only. However, if you specify the STEPS suboption of the SELECT= option, then plots are produced for all smoothing parameters examined in the selection process.

The *global-plot-options* apply to all relevant plots generated by the LOESS procedure, unless they are overridden with a *specific-plot-option*. The *global-plot-options* supported by the LOESS procedure follow.

Global Plot Options

MAXPOINTS=NONE | *number*
specifies that plots with elements that require processing more than *number* points are suppressed. The default is MAXPOINTS=5000. This cutoff is ignored if you specify MAXPOINTS=NONE.

ONLY
suppresses the default plots. Only the plots specifically requested are produced.

UNPACK
suppresses paneling. By default, multiple plots can appear in some output panels. Specify UNPACK to get each plot individually. You can specify PLOTS(UNPACK) to unpack the default plots. You can also specify UNPACK as a suboption with CONTOURFITPANEL, DIAGNOSTICS, FITPANEL, RESIDUALS and RESIDUALSBYSMOOTH.

Specific Plot Options

The following listing describes the specific plots and their options.

ALL
requests that all plots appropriate for the particular analysis be produced. You can specify other options with ALL; for example, to request all plots and unpack only the residuals, specify PLOTS=(ALL RESIDUALS(UNPACK)).

CONTOURFIT < (*contour-options*) >
produces a contour plot of the fitted surface overlaid with a scatter plot of the data for models with two regressors. Contour plots are not produced if you specify the DIRECT option in the MODEL statement. You can use the following *contour-options* to control how the observations are displayed:

OBS=GRADIENT
specifies that observations be displayed as circles colored by the observed response. The same color gradient is used to display the fitted surface and the observations. Observations where the predicted response is close to the observed response have similar colors—the greater the contrast between the color of an observation and the surface, the larger the residual is at that point. OBS=GRADIENT is the default if you do not specify any *contour-options*.

OBS=NONE
suppresses the observations.

OBS=OUTLINE
specifies that observations be displayed as circles with a border but with a completely transparent fill.

OBS=OUTLINEGRADIENT
is the same as OBS=GRADIENT except that a border is shown around each observation. This option is useful to identify the location of observations where the residuals are small, because at these points the color of the observations and the color of the surface are indistinguishable.

CONTOURFITPANEL < (< UNPACK > < *contour-options* >) >
produces panels of contour plots overlaid with a scatter plot of the data for each smoothing parameter specified in the SMOOTH= option in the MODEL statement, for models with two regressors. This plot is not produced if you specify the DIRECT option in the MODEL statement. If you do not specify the SMOOTH= option or if the model does

not have two regressors, then this plot is not produced. If you specify the SELECT= option in addition to the SMOOTH= option in the MODEL statement, then you need to additionally specify the STEPS suboption of the SELECT= option to obtain this plot. Note that each panel contains at most six plots, and multiple panels are used in the case that there are more than six smoothing parameters in the SMOOTH= option in the MODEL statement. See the CONTOURFIT option for a description of the individual plots in this panel. The UNPACK option suppresses paneling, and the *contour-options* are the same as for the CONTOURFIT option.

CRITERIONPLOT | CRITERION

displays a scatter plot of the value of the SELECTION= criterion versus the smoothing parameter value for all smoothing parameter values examined in the selection process. This plot is not produced if smoothing parameter selection is not done.

DIAGNOSTICSPANEL | DIAGNOSTICS < (UNPACK) >

produces a summary panel of fit diagnostics consisting of the following:

- residuals versus the predicted values
- histogram of the residuals
- normal quantile plot of the residuals
- a "Residual-Fit" (or RF) plot consisting of side-by-side quantile plots of the centered fit and the residuals.
- dependent variable values versus the predicted values

You can request the five plots in this panel as individual plots by specifying the UNPACK option. You can also request individual plots in the panel by name without having to unpack the panel. Note that the fit diagnostics panel is produced by default whenever ODS Graphics is enabled.

FITPANEL < (UNPACK) >

produces panels of plots showing the fitted LOESS curve overlaid on a scatter plot of the input data for each smoothing parameter specified in the SMOOTH= option in the MODEL statement. If you do not specify the SMOOTH= option or the model has more than one regressor, then this plot is not produced. If you specify the SELECT= option in addition to the SMOOTH= option in the MODEL statement, then you need to additionally specify the STEPS suboption of the SELECT= option to obtain this plot. Note that each panel contains at most six plots, and multiple panels are used in the case that there are more than six smoothing parameters in the SMOOTH= option in the MODEL statement. If the CLM option is specified in the MODEL statement, then a confidence band at the significance level specified in the ALPHA= option is included in each plot in the panels. If you specify the UNPACK option, then all fit panels are unpacked.

FITPLOT | FIT

produces a scatter plot of the input data with the fitted LOESS curve overlaid for models with a single regressor. If the CLM option is specified in the MODEL statement, then a confidence band at the significance level specified in the ALPHA= option is included in the plot.

NONE
suppresses all plots.

OBSERVEDBYPREDICTED
produces a scatter plot of the dependent variable values by the predicted values.

QQPLOT | QQ
produces a normal quantile plot of the residuals.

RESIDUALSBYSMOOTH < (< UNPACK > < SMOOTH >) >
produces for each regressor panels of plots showing the residuals of the LOESS fit versus the regressor for each smoothing parameter specified in the SMOOTH= option in the MODEL statement. If you do not specify the SMOOTH= option, then this plot is not produced. If you specify the SELECT= option in addition to the SMOOTH= option in the MODEL statement, then you need to additionally specify the STEPS suboption of the SELECT= option to obtain this plot. Note that each panel contains at most six plots, and multiple panels are used in the case that there are more than six smoothing parameters in the SMOOTH= option in the MODEL statement. If you specify the UNPACK option, then all RESIDUALSBYSMOOTH panels are unpacked.

The SMOOTH option requests that a nonparametric fit line be shown in each plot in the panel. The type of nonparametric fit and the options used are controlled by the template that underlies this plot. In the standard template that is provided, the nonparametric smooth is specified to be a loess fit correponding to the default options of PROC LOESS, except that the PRESEARCH suboption is always used. It is important to note that the loess fit that is shown in each of the residual plots is computed independently of the loess fit that is used to obtain the residuals.

RESIDUALBYPREDICTED
produces a scatter plot of the residuals by the predicted values.

RESIDUALHISTOGRAM
produces a histogram of the residuals.

RESIDUALPANEL | RESIDUALS < (*residual-options*) >
produces panels of the residuals versus the regressors in the model. Note that each panel contains at most six plots, and multiple panels are used when there are more than six regressors in the model.

The following *residual-options* are available:

SMOOTH
requests that a nonparametric fit line be shown in each plot in the panel. The type of nonparametric fit and the options used are controlled by the template that underlies this plot. In the standard template that is provided, the nonparametric smooth is specified to be a loess fit correponding to the default options of PROC LOESS, except that the PRESEARCH suboption is always used. It is important to note that the loess fit that is shown in each of the residual plots is computed independently of the loess fit that is used to obtain the residuals.

UNPACK
suppresses paneling.

RFPLOT | RF
produces a "Residual-Fit" (or RF) plot consisting of side-by-side quantile plots of the centered fit and the residuals. This plot "shows how much variation in the data is explained by the fit and how much remains in the residuals" (Cleveland 1993).

SCOREPLOT | SCORE
produces a scatter plot of the scored values at the score points for each SCORE statement. SCORE plots are not produced for models with more than one regressor. If the CLM option is specified in the MODEL statement, then confidence bars at the significance level specified in the ALPHA= option are shown at score data points.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC LOESS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in the order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

ID Statement

ID *variables* **;**

The ID statement is optional, and more than one ID statement can be used. The variables listed in any of the ID statements are displayed in the "Output Statistics" table beside each observation. Any variables specified as a regressor or dependent variable in the MODEL statement already appear in

the "Output Statistics" table and are not treated as ID variables, even if they appear in the variable list of an ID statement.

MODEL Statement

The MODEL statement names the dependent variables and the independent variables. Variables specified in the MODEL statement must be numeric variables in the data set being analyzed.

Table 50.2 lists the options available in the MODEL statement.

Table 50.2 Summary of MODEL Statement Options

Option	Description
Fit Options	
BUCKET=	specifies the number of points in kd tree buckets
DEGREE=	specifies the degree of local polynomials (1 or 2)
DFMETHOD=	specifies the method of computing lookup degrees of freedom
DIRECT	specifies direct fitting at every data point
DROPSQUARE=	specifies the variables whose squares are to be dropped from local quadratic polynomials
INTERP=	specifies the interpolating polynomials (linear or cubic)
ITERATIONS=	specifies the number of reweighting iterations
SCALE=	specifies the method used to scale the regressor variables
SELECT=	specifies that automatic smoothing parameter selection be done
SMOOTH=	specifies the list of smoothing values
Output Statistics Table Options	
ALL	requests CLM, RESIDUAL, SCALEDINDEP, STD, and T options
CLM	displays confidence limits for mean predictions
RESIDUAL	displays residuals
SCALEDINDEP	displays scaled independent variable coordinates
STD	displays standard errors of the mean predicted values
T	displays t statistics
Other options	
ALPHA=	sets significance level for confidence intervals
DETAILS=	specifies which tables are to be displayed
TRACEL	displays the trace of the smoothing matrix

The following options are available in the MODEL statement after a slash (/).

ALL
requests all these options: CLM, RESIDUAL, SCALEDINDEP, STD, and T.

ALPHA=*number*
sets the significance level used for the construction of confidence intervals for the current

MODEL statement. The value must be between 0 and 1; the default value of 0.05 results in 95% intervals.

BUCKET=*number*

specifies the maximum number of points in the leaf nodes of the kd tree. The default value used is $s * n/5$, where s is a smoothing parameter value specified using the SMOOTH= option and n is the number of observations being used in the current BY group. The BUCKET= option is ignored if the DIRECT option is specified.

CLM

requests that $100(1-\alpha)\%$ confidence limits on the mean predicted value be added to the "Output Statistics" table. By default, 95% limits are computed; the ALPHA= option in the MODEL statement can be used to change the significance level. The use of this option implicitly selects the model option DFMETHOD=EXACT if the DFMETHOD= option has not been explicitly used.

DEGREE=1 | 2

sets the degree of the local polynomials to use for each local regression. The valid values are 1 for local linear fitting and 2 for local quadratic fitting, with 1 being the default.

DETAILS < (*tables*) >

selects which tables to display, where *tables* is one or more of the specifications KDTREE, MODELSUMMARY, OUTPUTSTATISTICS, and PREDATVERTICES:

- KDTREE displays the kd tree structure.
- MODELSUMMARY displays the fit criteria for all smoothing parameter values that are specified in the SMOOTH= option in the MODEL statement, or that are fit with automatic smoothing parameter selection.
- OUTPUTSTATISTICS displays the predicted values and other requested statistics at the points in the input data set.
- PREDATVERTICES displays fitted values and coordinates of the kd tree vertices where the local least squares fitting is done.

The KDTREE and PREDATVERTICES specifications are ignored if the DIRECT option is specified in the MODEL statement. Specifying the option DETAILS with no qualifying list outputs all tables.

DFMETHOD=NONE | EXACT | APPROX < (*approx-options*) >

specifies the method used to calculate the lookup degrees of freedom used in performing statistical inference. The default is DFMETHOD=NONE, unless you specify any of the MODEL statement options ALL, CLM, STD, and T, or any SCORE statement CLM option, in which case the default is DFMETHOD=EXACT.

You can specify the following *approx-options* in parentheses after the DFMETHOD=APPROX option:

QUANTILE=*number*

specifies that the smallest 100(*number*)% of the nonzero coefficients in the smoothing matrix be set to zero in computing the approximate lookup degrees of freedom. The default value is QUANTILE=0.9.

CUTOFF=*number*

specifies that coefficients in the smoothing matrix whose magnitude is less than the specified value be set to zero in computing the approximate lookup degrees of freedom. Using the CUTOFF= option overrides the QUANTILE= option.

See the section "Sparse and Approximate Degrees of Freedom Computation" on page 3224 for a description of the method used when the DFMETHOD=APPROX option is specified.

DIRECT

specifies that local least squares fits are to be done at every point in the input data set. When the direct option is not specified, a computationally faster method is used. This faster method performs local fitting at vertices of a kd tree decomposition of the predictor space followed by blending of the local polynomials to obtain a regression surface.

DROPSQUARE=(*variables***)**

specifies the quadratic monomials to exclude from the local quadratic fits. This option is ignored unless the DEGREE=2 option has been specified.

For example,

```
model z=x y / degree=2 dropsquare=(y)
```

uses the monomials 1, x, y, x^2, and xy in performing the local fitting.

INTERP=LINEAR | CUBIC

specifies the degree of the interpolating polynomials used for blending local polynomial fits at the kd tree vertices. This option is ignored if the DIRECT option is specified in the model statement. INTERP=CUBIC is not supported for models with more than two regressors. The default is INTERP=LINEAR.

ITERATIONS=*number*

specifies the total number of iterations to be done. The first iteration performs an initial LOESS fit. Subsequent iterations perform iterative reweighting. Such iterations are appropriate when there are outliers in the data or when the error distribution is a symmetric long-tailed distribution. The default number of iterations is 1.

RESIDUAL | R

specifies that residuals be included in the "Output Statistics" table.

SCALE=NONE | SD < (*number***) >**

specifies the scaling method to be applied to scale the regressors. The default is NONE, in which case no scaling is applied. A specification of SD(*number*) indicates that a trimmed standard deviation is to be used as a measure of scale, where *number* is the trimming fraction. A specification of SD with no qualification defaults to 10% trimmed standard deviation.

SCALEDINDEP

specifies that scaled regressor coordinates be included in the output tables. This option is ignored if the SCALE= model option is not used or if SCALE=NONE is specified.

SELECT=*criterion* **<(<GLOBAL> <PRESEARCH> <STEPS> <RANGE(***lower***,***upper***)>)>**
SELECT=*DFCriterion* **<(** *target* **<GLOBAL> <PRESEARCH> <STEPS> <RANGE(***lower***,***upper***)>)>**

specifies that automatic smoothing parameter selection be done using the named *criterion* or *DFCriterion*. Valid values for the *criterion* are as follows:

AICC specifies the AIC_C criterion (Hurvich, Simonoff, and Tsai 1998).

AICC1 specifies the AIC_{C_1} criterion (Hurvich, Simonoff, and Tsai 1998).

GCV specifies the generalized cross validation criterion (Craven and Wahba 1979).

The *DFCriterion* specifies the measure used to estimate the model degrees of freedom. The measures implemented in PROC LOESS all depend on prediction matrix L relating the observed and predicted values of the dependent variable. Valid values for the *DFCriterion* are as follows:

DF1 specifies Trace(L).

DF2 specifies Trace($L^T L$).

DF3 specifies 2Trace(L) − Trace($L^T L$).

For both types of selection, the smoothing parameter value is selected to yield a minimum of an optimization criterion. If you specify *criterion* as one of AICC, AICC1, or GCV, the optimization criterion is the specified *criterion*. If you specify *DFCriterion* as one of DF1, DF2, or DF3, the optimization criterion is | *DFCriterion* − *target* |, where *target* is a specified target degree of freedom value. Note that if you specify a *DFCriterion*, then you must also specify a target value. See the section "Automatic Smoothing Parameter Selection" on page 3221 for definitions and properties of the selection criteria.

The selection is done as follows:

- If you specify the SMOOTH=*value-list* option, then PROC LOESS selects the largest value in this list that yields the global minimum of the specified optimization criterion.
- If you do not specify the SMOOTH= option, then PROC LOESS finds a local minimum of the specified optimization criterion by using a golden section search of values less than or equal to one.

You can specify the following suboptions in parentheses after the specified criterion to alter the behavior of the SELECT= option:

GLOBAL

specifies that a global minimum be found within the range of smoothing parameter values examined. This suboption has no effect if you also specify the SMOOTH= option in the MODEL statement.

PRESEARCH

requests an initial grid search to find a smoothing parameter range within which the subsequent golden section search is done. The inital point in this grid is the smoothing parameter value corresponding to the smallest number of points, n, in the local neighborhoods that yields a fit that does not interpolate all the data points. Subsequent fits with number of local points $n+1$, $n+2$, $n+4$, $n+8$, ... are evaluated until either the number of local points exceeds the number of fitting points or the SELECT=criterion starts increasing. This suboption is ignored if you additionally specify the GLOBAL suboption of the SELECT= option or if you specify the SMOOTH= option in the MODEL statement. If you additionally specify the RANGE= suboption, then the golden section search is done on the intersection of the range found by this grid search and the range that you specify in the RANGE= suboption. This option is useful for data exhibiting features at multiple scales, because in such cases the SELECT= criterion often has multiple local minima. Using the PRESEARCH option increases the likelihood that the golden section search will find the global minimum of the SELECT= criterion. See Example 50.4 for such an example.

RANGE(*lower*,*upper*)

specifies that only smoothing parameter values greater than or equal to *lower* and less than or equal to *upper* be examined.

STEPS

specifies that all models evaluated in the selection process be displayed.

For models with one dependent variable, if you specify neither the SELECT= nor the SMOOTH= options in the MODEL statement, then PROC LOESS uses SELECT=AICC.

The following table summarizes how the smoothing parameter values are chosen for various combinations of the SMOOTH= option, the SELECT= option, and the SELECT= option modifiers.

Table 50.3 Smoothing Parameter Value(s) Used for Combinations of SMOOTH= and SELECT= OPTIONS for Models with One Dependent Variable

Syntax	Search Method	Search Domain
default	golden section using AICC	$(0, 1]$
SMOOTH=*list*	no selection	values in *list*
SMOOTH=*list* SELECT=*criterion*	global	values in *list*
SMOOTH=*list* SELECT=*criterion* (RANGE(l, u))	global	values in *list* within $[l, u]$
SELECT=*criterion*	golden section	$(0, 1]$
SELECT=*criterion* (RANGE(*l*,*u*))	golden section	$[l, u]$
SELECT=*criterion* (GLOBAL)	global	$(0, 1]$
SELECT=*criterion* (GLOBAL RANGE(l, u))	global	$[l, u]$

Some examples of using the SELECT= option follow:

SELECT=GCV	specifies selection that uses the GCV *criterion*.
SELECT=DF1(6.3)	specifies selection that uses the DF1 *DFCriterion* with target value 6.3.
SELECT=AICC(STEPS)	specifies selection that uses the AICC *criterion*, showing all step details.
SELECT=DF2(7 GLOBAL)	specifies selection that uses a global search algorithm to find the smoothing parameter that yields the DF2 *DFCriterion* closest to the target value 7.

NOTE: The SELECT= option cannot be used for models with more than one dependent variable.

SMOOTH=*value-list*

specifies a list of positive smoothing parameter values. If you do not specify the SELECT= option in the MODEL statement, then a separate fit is obtained for each SMOOTH= value specified. If you do specify the SELECT= option, then models with all values specified in the SMOOTH= list are examined, and PROC LOESS selects the value that minimizes the criterion specified in the SELECT= option.

For models with two or more dependent variables, if the SMOOTH= option is not specified in the MODEL statement, then SMOOTH=0.5 is used as a default.

STD

specifies that standard errors of the mean predicted values be included in the "Output Statistics" table. The use of this option implicitly selects the model option DFMETHOD=EXACT if the DFMETHOD= option has not been explicitly used.

T

specifies that *t* statistics are to be included in the "Output Statistics" table. The use of this option implicitly selects the model option DFMETHOD=EXACT if the DFMETHOD= option has not been explicitly used.

TRACEL

specifies that the trace of the prediction matrix as well as the GCV and AICC statistics be included in the "Fit Summary" table. The use of any of the MODEL statement options ALL, CLM, DFMETHOD=EXACT, DIRECT, SELECT=, STD, and T implicitly selects the TRACEL option.

SCORE Statement

SCORE < *DATA=SAS-data-set* > < **ID=***(variable list)* > < / *options* > **;**

The fitted loess model is used to score the data in the specified SAS data set. This data set must contain all the regressor variables specified in the MODEL statement. Furthermore, when a BY

statement is used, the score data set must also contain all the BY variables sorted in the order of the BY variables. A SCORE statement is optional, and more than one SCORE statement can be used. SCORE statements cannot be used if the DIRECT option is specified in the MODEL statement. The optional ID= (*variable list*) specifies ID variables to be included in the "Score Results" table.

You find the results of the SCORE statement in the "Score Results" table. This table contains all the data in the data set named in the SCORE statement, including observations with missing values. However, only those observations with nonmissing regressor variables are scored. If no data set is named in the SCORE statement, the data set named in the PROC LOESS statement is scored. You use the PRINT option in the SCORE statement to request that the "Score Results" table be displayed. You can place the "Score Results" table in an output data set by using an ODS OUTPUT statement even if this table is not displayed.

The following options are available in the SCORE statement after a slash (/).

CLM

requests that $100(1-\alpha)\%$ confidence limits on the mean predicted value be added to the "Score Results" table. By default the 95% limits are computed; the ALPHA= option in the MODEL statement can be used to change the significance level. The use of this option implicitly selects the model option DFMETHOD=EXACT if the DFMETHOD= option has not been explicitly used.

PRINT <(*variables* **)>**

specifies that the "Score Results" table be displayed. By default only the variables named in the MODEL statement, the variables listed in the ID list in the SCORE statement, and the scored dependent variables are displayed. The optional list in the PRINT option specifies additional variables in the score data set that are to be included in the displayed output. Note, however, that all columns in the SCORE data set are placed in the SCORE results table, even if you do not request that they be included in the displayed output.

SCALEDINDEP

specifies that scaled regressor coordinates be included in the SCORE results table. This option is ignored if the SCALE= model option is not used or if SCALE=NONE is specified.

STEPS

requests that all models evaluated during smoothing parameter value selection be scored, provided that the SELECT= option together with the STEPS modifier is specified in the MODEL statement. By default only the selected model is scored.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement specifies a variable in the input data set that contains values to be used as a priori weights for a loess fit.

The values of the weight variable must be nonnegative. If an observation's weight is zero, negative, or missing, the observation is deleted from the analysis.

Details: LOESS Procedure

Missing Values

PROC LOESS deletes any observation with missing values for any variable specified in the MODEL statement. This enables the procedure to reuse the kd tree for all the dependent variables that appear in the MODEL statement. If you have multiple dependent variables with different missing value structures for the same set of independent variables, you might want to use separate PROC LOESS steps for each dependent variable.

Output Data Sets

PROC LOESS assigns a name to each table it creates. You can use the ODS OUTPUT statement to place one or more of these tables in output data sets. See the section "ODS Table Names" on page 3225 for a list of the table names created by PROC LOESS. For detailed information about ODS, see Chapter 20, "Using the Output Delivery System."

For example, the following statements create an output data set named MyOutStats containing the "Output Statistics" table and an output data set named MySummary containing the "Fit Summary" table.

```
proc loess data=Melanoma;
   model Incidences=Year;
   ods output OutputStatistics = MyOutStats
              FitSummary       = MySummary;
run;
```

Often, a single MODEL statement describes more than one model. For example, the following statements fit eight different models (four smoothing parameter values for each dependent variable).

```
proc loess;
   model y1 y2 = x1 x2 x3/smooth =0.1 to 0.7 by 0.2;
   ods output OutputStatistics = MyOutStats;
run;
```

The eight "Output Statistics" tables for these models are stacked in a single data set called MyOutStats. The data set contains a column named DepVarName and a column named SmoothingParameter that distinguish each model (see Figure 50.8 for an example). If you want the "Output Statistics" table for each model to be in its own data set, you can use the MATCH_ALL option in the ODS OUTPUT statement. The following statements create eight data sets named MyOutStats, MyOutStats1, . . . , MyOutStats7.

```
proc loess;
   model y1 y2 = x1 x2 x3/smooth =0.1 to 0.7 by 0.2;
   ods output OutputStatistics(match_all) = MyOutStats;
run;
```

For further options available in the ODS OUTPUT statement, see Chapter 20, "Using the Output Delivery System."

Only the "Scale Details" and "Fit Summary" tables are displayed by default. The other tables are optionally displayed by using the DETAILS option in the MODEL statement and the PRINT option in the SCORE statement. Note that it is not necessary to display a table in order for that table to be used in an ODS OUTPUT statement. For example, the following statements display the "Output Statistics" and "kd Tree" tables but place the "Output Statistics" and "Prediction at Vertices" tables in output data sets.

```
proc loess data=Melanoma;
   model Incidences=Year/details(OutputStatistics kdTree);
   ods output OutputStatistics = MyOutStats
              PredAtVertices   = MyVerticesOut;
run;
```

Using the DETAILS option alone causes all tables to be displayed.

The MODEL statement options CLM, RESIDUAL, STD, SCALEDINDEP, and T control which optional columns are added to the OutputStatistics table. For example, to obtain an OutputStatistics output data set containing residuals and confidence limits in addition to the model variables and predicted value, you need to specify the RESIDUAL and CLM options in the MODEL statement as in the following example:

```
proc loess data=Melanoma;
   model Incidences=Year/residual clm;
   ods output OutputStatistics = MyOutStats;
run;
```

Finally, note that using the ALL option in the MODEL statement causes all optional columns to be included in the output. Also, ID columns can be added to the OutputStatistics table by using the ID statement.

Data Scaling

The loess algorithm to obtain a predicted value at a given point in the predictor space proceeds by doing a least squares fit that uses all data points close to the given point. Thus the algorithm depends critically on the metric used to define closeness. This has the consequence that if you have more than one predictor variable and these predictor variables have significantly different scales, then closeness depends almost entirely on the variable with the largest scaling. It also means that merely changing the units of one of your predictors can significantly change the loess model fit.

To circumvent this problem, it is necessary to standardize the scale of the independent variables in the loess model. The SCALE= option in the MODEL statement is provided for this purpose. PROC LOESS uses a symmetrically trimmed standard deviation as the scale estimate for each independent

variable of the loess model. This is a robust scale estimator in that extreme values of a variable are discarded before estimating the data scaling. For example, to compute a 10% trimmed standard deviation of a sample, you discard the smallest and largest 5% of the data and compute the standard deviation of the remaining 90% of the data points. In this case, the trimming fraction is 0.1.

For example, the following statement specifies that the variables Temperature and Catalyst are scaled before performing the loess fitting. In this case, because the trimming fraction is 0.1, the scale estimate used for each of these variables is a 10% trimmed standard deviation.

```
model Yield=Temperature Catalyst / scale = SD(0.1);
```

The default trimming fraction used by PROC LOESS is 0.1 and need not be specified by the SCALE= option. Thus the following MODEL statement is equivalent to the previous MODEL statement.

```
model Yield=Temperature Catalyst / scale = SD;
```

If the SCALE= option is not specified, no scaling of the independent variables is done. This is appropriate when there is only a single independent variable or when all the independent variables are a priori scaled similarly.

When the SCALE= option is specified, the scaling details for each independent variable are added to the ScaleDetails table (see Output 50.3.2 for an example). By default, this table contains only the minimum and maximum values of each independent variable in the model. Finally, note that when the SCALE= option is used, specifying the SCALEDINDEP option in the MODEL statement adds the scaled values of the independent variables to the OutputStatistics and PredAtVertices tables. If the SCALEDINDEP option is specified in the SCORE statement, then scaled values of the independent variables are included in the ScoreResults table. By default, only the unscaled values are placed in these tables.

Direct versus Interpolated Fitting

Local regression to obtain a predicted value at a given point in the predictor space is done by doing a least squares fit that uses all data points in a local neighborhood of the given point. This method is computationally expensive because a local neighborhood must be determined and a least squares problem must be solved for each point at which a fitted value is required. A faster method is to obtain such fits at a representative sample of points in the predictor space and to obtain fitted values at all other points by interpolation.

PROC LOESS can fit models by using either of these two methods. By default, PROC LOESS uses fitting at a sample of points and interpolation. The method fitting a local model at every data point is selected by specifying the DIRECT option in the MODEL statement.

kd Trees and Blending

PROC LOESS uses a kd tree to divide the box (also called the *initial cell* or *bucket*) enclosing all the predictor data points into rectangular cells. The vertices of these cells are the points at which local least squares fitting is done.

Starting from the initial cell, the direction of the longest cell edge is selected as the split direction. The median of this coordinate of the data in the cell is the split value. The data in the starting cell are partitioned into two child cells. The left child consists of all data from the parent cell whose coordinate in the split direction is less than the split value. This procedure is repeated for each child cell that has more than a prespecified number of points, called the *bucket size* of the kd tree.

You can specify the bucket size with the BUCKET= option in the MODEL statement. If you do not specify the BUCKET= option, the default value used is the largest integer less than or equal to $ns/5$, where n is the number of observations and s is the value of the smoothing parameter. Note that if fitting is being done for a range of smoothing parameter values, the bucket size can change for each value.

The set of vertices of all the cells of the kd tree are the points at which PROC LOESS performs its local fitting. The fitted value at an original data point (or at any other point within the original data cell) is obtained by blending the fitted values at the vertices of the kd tree cell that contains that data point.

The univariate blending methods available in PROC LOESS are linear and cubic polynomial interpolation, with linear interpolation being the default. You can request cubic interpolation by specifying the INTERP=CUBIC option in the MODEL statement. In this case, PROC LOESS uses the unique cubic polynomial whose values and first derivatives match those of the fitted local polynomials evaluated at the two endpoints of the kd tree cell edge.

In the multivariate case, such univariate interpolating polynomials are computed on each edge of the kd tree cells and are combined using blending functions (Gordon 1971). In the case of two regressors, if you specify INTERP=CUBIC in the MODEL statement, PROC LOESS uses Hermite cubic polynomials as blending functions. If you do not specify INTERP=CUBIC, or if you specify a model with more than two regressors, then PROC LOESS uses linear polynomials as blending functions. In these cases, the blending method reduces to tensor product interpolation from the 2^p vertices of each kd tree cell, where p is the number of regressors.

While the details of the kd tree and the fitted values at the vertices of the kd tree are implementation details that seldom need to be examined, PROC LOESS does provide options for their display. Each kd tree subdivision of the data used by PROC LOESS is placed in the "kdTree" table. The predicted values at the vertices of each kd tree are placed in the "PredAtVertices" table. You can request these tables by using the DETAILS option in the MODEL statement.

Local Weighting

The size of the local neighborhoods that PROC LOESS uses in performing local fitting is determined by the smoothing parameter value s. When $s < 1$, the local neighborhood used at a point x contains the s fraction of the data points closest to the point x. When $s \geq 1$, all data points are used.

Suppose q denotes the number of points in the local neighborhoods and $d_1, d_2, \ldots, d_q$ denote the distances in increasing order of the q points closest to x. The point at distance d_i from x is given a weight w_i in the local regression that decreases as the distance from x increases. PROC LOESS uses a tricube weight function to define

$$w_i = \frac{32}{5}\left(1 - (\frac{d_i}{d_q})^3\right)^3$$

If $s > 1$, then d_q is replaced by $d_q s^{1/p}$ in the previous formula, where p is the number of predictors in the model.

Finally, note that if a weight variable has been specified using a WEIGHT statement, then w_i is multiplied by the corresponding value of the specified weight variable.

Iterative Reweighting

PROC LOESS can do iterative reweighting to improve the robustness of the fit in the presence of outliers in the data. Iterative reweighting is also appropriate when statistical inference is requested and the error distribution is symmetric but not Gaussian.

The number of iterations is specified by the ITERATIONS= option in the MODEL statement. The default is ITERATIONS=1, which corresponds to no reweighting.

At iterations beyond the first iteration, the local weights w_i of the previous section are replaced by $r_i w_i$, where r_i is a weight that decreases as the residual of the fitted value at the previous iteration at the point corresponding to d_i increases. Refer to Cleveland and Grosse (1991) and Cleveland, Grosse, and Shyu (1992) for details.

Specifying the Local Polynomials

PROC LOESS uses linear or quadratic polynomials in doing the local least squares fitting. The option DEGREE = in the MODEL statement is used to specify the degree of the local polynomials used by PROC LOESS, with DEGREE = 1 being the default. In addition, when DEGREE = 2 is specified, the MODEL statement DROPSQUARE= option can be used to exclude specific monomials during the least squares fitting.

For example, the following statements use the monomials 1, x1, x2, x1*x2, and x2*x2 for the local least squares fitting.

```
proc loess;
   model y= x1 x2/ degree=2 dropsquare=(x1);
run;
```

Smoothing Matrix

When no iterative reweighting is done, the "Smoothing Matrix" denoted by L defines the linear relationship between the fitted and observed dependent variable values of a loess model. You can obtain the predicted values of a loess fit from the observed values via

$$\hat{y} = Ly$$

where y is the vector of observed values and $\hat{y}$ is the corresponding vector of predicted values of the dependent variable. Note that L is an n by n matrix, where n is the number of observations in the analysis. PROC LOESS does not explicitly form L if the DFMETHOD=EXACT option is not explicitly or implicitly selected.

Model Degrees of Freedom

The approximate model degrees of freedom in a nonparametric fit is a number that is analogous to the number of free parameters in a parametric model. There are three commonly used measures of model degrees of freedom in nonparametric models. These criteria are as follows:

$$\begin{aligned} \text{DF1} &\equiv \text{Trace}(L) \\ \text{DF2} &\equiv \text{Trace}(L^T L) \\ \text{DF3} &\equiv 2\text{Trace}L - \text{Trace}(L^T L) \end{aligned}$$

A discussion of their properties can be found in Hastie and Tibshirani (1990). DF2 is also referred to as the "Equivalent Number of Parameters," and this is the name that PROC LOESS uses for DF2 when it appears in the "Fit Summary" table.

Statistical Inference and Lookup Degrees of Freedom

If you denote the ith measurement of the response by y_i and the corresponding measurement of predictors by x_i, then

$$y_i = g(x_i) + \epsilon_i$$

where g is the regression function and ϵ_i are independent random errors with mean zero. If the errors are normally distributed with constant variance, then you can obtain confidence intervals for the predictions from PROC LOESS. You can also obtain confidence limits in the case where ϵ_i is heteroscedastic but $a_i\epsilon_i$ has constant variance and a_i are a priori weights that are specified using the WEIGHT statement of PROC LOESS. You can do inference in the case in which the error

distribution is symmetric by using iterative reweighting. Formulas for doing statistical inference under the preceding conditions can be found in Cleveland and Grosse (1991) and Cleveland, Grosse, and Shyu (1992). Cleveland and Grosse (1991) show that standardized residuals for a loess model follow a t distribution with ρ degrees of freedom where

$$\begin{aligned}
\delta_1 &\equiv \text{Trace}(I-L)^T(I-L) \\
\delta_2 &\equiv \text{Trace}\left((I-L)^T(I-L)\right)^2 \\
\rho &\equiv \text{Lookup Degrees of Freedom} \\
&\equiv \delta_1^2/\delta_2
\end{aligned}$$

The residual standard error that you find in the "Fit Summary" table is defined by

$$\text{Residual Standard Error} \equiv \sqrt{\text{Residual SS}/\delta_1}$$

The determination of ρ is computationally expensive and is not done by default. It is computed if you specify the DFMETHOD=EXACT or DFMETHOD=APPROX option in the MODEL statement. It is also computed if you specify any of the options CLM, STD, and T in the MODEL statement. Note that the values of δ_1, δ_2, and ρ are reported in the "Fit Summary" table.

If you specify the CLM option in the MODEL statement, confidence limits are added to the OutputStatistics table. By default, 95% limits are computed, but you can change this by using the ALPHA= option in the MODEL statement.

Automatic Smoothing Parameter Selection

There are several methodologies for automatic smoothing parameter selection. One class of methods chooses the smoothing parameter value to minimize a criterion that incorporates both the tightness of the fit and model complexity. Such a criterion can usually be written as a function of the error mean square, $\hat{\sigma}^2$, and a penalty function designed to decrease with increasing smoothness of the fit. This penalty function is usually defined in terms of the smoothing matrix L (see the section "Smoothing Matrix" on page 3220).

Examples of specific criteria are generalized cross validation (Craven and Wahba 1979) and the Akaike information criterion (Akaike 1973). These classical selectors have two undesirable properties when used with local polynomial and kernel estimators: they tend to undersmooth small data sets and tend to be nonrobust in the sense that small variations of the input data can change the choice of smoothing parameter value significantly. Hurvich, Simonoff, and Tsai (1998) obtained several corrected AIC criteria that address the small-sample bias and perform comparably with the *plug-in selectors* (Ruppert, Sheather, and Wand 1995). PROC LOESS provides automatic smoothing parameter selection that uses two of these corrected AIC criteria, named $AICC_1$ and $AICC$ in Hurvich, Simonoff, and Tsai (1998), and generalized cross validation, denoted by GCV.

The relevant formulas are

$$
\begin{aligned}
AICC_1 &= n\log(\hat{\sigma}^2) + n\frac{\delta_1/\delta_2(n+\nu_1)}{\delta_1^2/\delta_2 - 2} \\
AICC &= \log(\hat{\sigma}^2) + 1 + \frac{2\,(\mathrm{Trace}(L)+1)}{n - \mathrm{Trace}(L) - 2} \\
GCV &= \frac{n\hat{\sigma}^2}{(n-\mathrm{Trace}(L))^2}
\end{aligned}
$$

where n is the number of observations and

$$
\begin{aligned}
\delta_1 &\equiv \mathrm{Trace}(I-L)^T(I-L) \\
\delta_2 &\equiv \mathrm{Trace}\left((I-L)^T(I-L)\right)^2 \\
\nu_1 &\equiv \text{Equivalent Number of Parameters} \\
&\equiv \mathrm{Trace}(L^TL)
\end{aligned}
$$

You invoke these methods for automatic smoothing parameter selection by specifying the SELECT=*criterion* option in the MODEL statement, where *criterion* is AICC1, AICC, or GCV. The LOESS procedure evaluates the specified criterion for a sequence of smoothing parameter values and selects the value in this sequence that minimizes the specified criterion. If multiple values yield the optimum, then the largest of these values is selected.

A second class of methods seeks to set an approximate measure of model degrees of freedom to a specified target value. These methods are useful for making meaningful comparisons between loess fits and other nonparametric and parametric fits. Three approximate model degrees of freedom for a loess model are defined in the section "Model Degrees of Freedom" on page 3220. You invoke these methods by specifying the SELECT=*DFCriterion*(*target*) option in the MODEL statement, where *DFCriterion* is DF1, DF2, or DF3. The criterion that is minimized is given in the following table.

Table 50.4 Minimization Criteria

Syntax	Minimization Criterion
SELECT=DF1(*target*)	$\lvert\ \mathrm{Trace}(L) - \mathit{target}\ \rvert$
SELECT=DF2(*target*)	$\lvert\ \mathrm{Trace}(L^TL) - \mathit{target}\ \rvert$
SELECT=DF3(*target*)	$\lvert 2\mathrm{Trace}(L) - \mathrm{Trace}(L^TL) - \mathit{target}\ \rvert$

The results are summarized in the "Smoothing Criterion" table. This table is displayed whenever automatic smoothing parameter selection is performed. You can obtain details of the sequence of models examined by specifying the DETAILS(MODELSUMMARY) option in the MODEL statement to display the "Model Summary" table.

There are several ways in which you can control the sequence of models examined by PROC LOESS. If you specify the SMOOTH=*value-list* option in the MODEL statement, then only the values in this list are examined in performing the selection. For example, the following statements select the model that minimizes the AICC1 criterion among the three models with smoothing parameter values 0.1, 0.3, and 0.4:

```
proc loess;
   model y= x1/ smooth=0.1 0.3 0.4 select=AICC1;
run;
```

If you do not specify the SMOOTH= option in the MODEL statement, then by default PROC LOESS uses a golden section search method to find a local minimum of the specified criterion in the range $(0, 1]$. You can use the RANGE(*lower*,*upper*) modifier in the SELECT= option to change the interval in which the golden section search is performed. For example, the following statements request a golden section search to find a local minimizer of the GCV criterion for smoothing parameter values in the interval [0.1,0.5]:

```
proc loess;
   model y= x1/select=GCV( range(0.1,0.5) );
run;
```

If you want to be sure of obtaining a global minimum in the range of smoothing parameter values examined, you can specify the GLOBAL modifier in the SELECT= option. For example, the following statements request that a global minimizer of the AICC criterion be obtained for smoothing parameter values in the interval $[0.2, 0.8]$:

```
proc loess;
   model y= x1/select=AICC( global range(0.2,0.8) );
run;
```

Note that even though the smoothing parameter is a continuous variable, a given range of smoothing parameter values corresponds to a finite set of local models. For example, for a data set with 100 observations, the range $[0.2, 0.4]$ corresponds to models with $20, 21, 22, \ldots, 40$ points in the local neighborhoods. If the GLOBAL modifier is specified, all possible models in the range are evaluated sequentially.

Note that by default PROC LOESS displays a "Fit Summary" and other optionally requested tables only for the selected model. You can request that these tables be displayed for all models in the selection process by adding the STEPS modifier in the SELECT= option. Also note that by default scoring requested with SCORE statements is done only for the selected model. However, if you specify the STEPS in both the MODEL and SCORE statements, then all models evaluated in the selection process are scored.

In terms of computation, $AICC$, GCV, and DF1 depend on the smoothing matrix L only through its trace. In the direct method, this trace can be computed efficiently. In the interpolated method that uses kd trees, there is some additional computational cost but the overall work is not significant compared to the rest of the computation. In contrast, the quantities δ_1, δ_2, and ν_1 that appear in the $AICC_1$ criterion, and the DF2 and DF3 criteria, depend on the entire L matrix and for this reason, the time needed to compute these quantities dominates the time required for the model fitting. Hence SELECT=AICC1, SELECT=DF2, and SELECT=DF3 are much more computationally expensive than SELECT=AICC, SELECT=GCV, and SELECT=DF1, especially when combined with the GLOBAL modifier. Hurvich, Simonoff, and Tsai (1998) note that $AICC$ can be regarded as an approximation of $AICC_1$ and that "the $AICC$ selector generally performs well in all circumstances."

For models with one dependent variable, PROC LOESS uses SELECT=AICC as its default, if you specify neither the SMOOTH= nor the SELECT= option in the MODEL statement. With two or more dependent variables, automatic smoothing parameter selection needs to be done separately for each dependent variable. For this reason automatic smoothing parameter selection is not available for models with multiple dependent variables. In such cases you should use a separate PROC LOESS step for each dependent variable, if you want to use automatic smoothing parameter selection.

Sparse and Approximate Degrees of Freedom Computation

As noted in the section "Statistical Inference and Lookup Degrees of Freedom" on page 3220, obtaining confidence limits in loess models requires the computation of the lookup degrees of freedom. This in turn requires the computation of

$$\delta_2 \equiv \text{Trace}\left((I-L)^T(I-L)\right)^2$$

where L is the loess smoothing matrix (see the section "Smoothing Matrix" on page 3220).

The work in a direct implementation of this formula grows as n^3, where n is the number of observations in analysis. For large n, this work dominates the time needed to fit the loess model itself. To alleviate this computational bottleneck, Cleveland and Grosse (1991) and Cleveland, Grosse, and Shyu (1992) developed approximate methods for estimating this quantity in terms of more readily computable statistics. A different approach to obtaining a computationally cheap estimate of δ_2 has been implemented in PROC LOESS.

For large data sets with significant local structure, the loess model is often used with small values of the smoothing parameter. Recalling that the smoothing parameter defines the fraction of the data used in each local regression, this means that the loess fit at any point in regressor space depends on only a small fraction of the data. This is reflected in the smoothing matrix L whose (i,j)th entry is nonzero only if the ith and jth observations lie in at least one common local neighborhood. Hence the smoothing matrix is a sparse matrix (has mostly zero entries) in such cases. By exploiting this sparsity, PROC LOESS now computes δ_2 orders of magnitude faster than in previous implementations.

When each local neighborhood contains a large subset of the data—i.e., when the smoothing parameter is large—then it is no longer true that the smoothing matrix is sparse. However, since a point in a local neighborhood is given a local weight that decreases with its distance from the center of the neighborhood, many of the coefficients in the smoothing matrix turn out to be nonzero but with orders of magnitude smaller than that of the larger coefficients in the matrix. The approximate method for computing δ_2 that has been implemented in PROC LOESS exploits these disparities in magnitudes of the elements in the smoothing matrix by setting the small elements to zero. This creates a sparse approximation of the smoothing matrix to which the fast sparse methods can be applied.

In order to decide the threshold at which elements in the smoothing matrix are set to zero, PROC LOESS samples the elements in the smoothing matrix to obtain the value of the element in a speci-

fied lower quantile in this sample. The magnitude of the element at this quantile is used as a cutoff value, and all elements in the smoothing matrix whose magnitude is less than this cutoff are set to zero for the approximate computation. By default all elements in the lower 90th percentile are set to zero. You can use the DFMETHOD=APPROX(QUANTILE=) option in the MODEL statement to change this value. As you increase the value for the quantile to be zeroed, you speed up the degrees of freedom computation at the expense of increasing approximation errors. You can also use the DFMETHOD=APPROX(CUTOFF=) option in the MODEL statement to specify the cutoff value directly.

For small data sets, the approximate computation is not needed and would be rougher than for larger data sets. Hence PROC LOESS performs the exact computation for analyses with fewer than 500 points, even if DFMETHOD=APPROX is specified in the model statement. Also, for small values of the smoothing parameter, elements in the lower specified quantile might already all be zero. In such cases the approximate method is the same as the exact method. PROC LOESS labels as approximate any statistics that depend on the approximate computation of δ_2 only in the cases where the approximate computation was used and is different from the exact computation.

Scoring Data Sets

One or more SCORE statements can be used with PROC LOESS. A data set that includes all the variables specified in the MODEL and BY statements must be specified in each SCORE statement. Score results are placed in the ScoreResults table. This table is not displayed by default, but specifying the PRINT option in the SCORE statement produces the table. If you specify the CLM option in the SCORE statement, confidence intervals are included in the ScoreResults table.

Note that scoring is not supported when the DIRECT option is specified in the MODEL statement. Scoring at a point specified in a score data set is done by first finding the cell in the kd tree containing this point and then interpolating the scored value from the predicted values at the vertices of this cell. This methodology precludes scoring any points that are not contained in the box that surrounds the data used in fitting the loess model.

ODS Table Names

PROC LOESS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 50.5 ODS Tables Produced by PROC LOESS

ODS Table Name	Description	Statement	Option
FitSummary	Specified fit parameters and fit summary		default
kdTree	Structure of kd tree used	MODEL	DETAILS(kdTree)
ModelSummary	Summary of all models evaluated	MODEL	DETAILS(ModelSummary)
OutputStatistics	Coordinates and fit results at input data points	MODEL	DETAILS(OutputStatistics)
PredAtVertices	Coordinates and fitted values at kd tree vertices	MODEL	DETAILS(PredAtVertices)
ScaleDetails	Extent and scaling of the independent variables		default
ScoreResults	Coordinates and fit results at scoring points	SCORE	PRINT
SmoothingCriterion	Criterion value and selected smoothing parameter	MODEL	SELECT

ODS Graphics

To request graphics with PROC LOESS, you must first enable ODS Graphics by specifying the **ODS GRAPHICS ON** statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC LOESS generates are listed in Table 50.6, along with the relevant PLOTS= options.

Table 50.6 ODS Graphics Produced by PROC LOESS

ODS Graph Name	Plot Description	PLOTS Option
ContourFitPanel	Panel of loess contour surfaces overlaid on scatter plots of data	CONTOURFITPANEL
ContourFit	Loess contour surface overlaid on scatter plot of data	CONTOURFITPANEL
DiagnosticsPanel	Panel of fit diagnostics	DIAGNOSTICS
FitPanel	Panel of loess curves overlaid on scatter plots of data	FITPANEL
FitPlot	Loess curve overlaid on scatter plot of data	FIT
ObservedByPredicted	Dependent variable versus loess fit	OBSERVEDBYPREDICTED
QQPlot	Normal quantile plot residuals	QQPLOT
ResidualsBySmooth	Panel of residuals versus regressor by smoothing parameter values	RESIDUALSBYSMOOTH

Table 50.6 *continued*

ODS Graph Name	Plot Description	PLOTS Option
ResidualByPredicted	Residuals versus loess fit	RESIDUALBYPREDICTED
ResidualHistogram	Histogram of fit residuals	RESIDUALHISTOGRAM
ResidualPanel	Panel of residuals versus regressors for fixed smoothing parameter value	RESIDUALS
ResidualPlot	Plot of residuals versus regressor	RESIDUALS
RFPlot	Side-by-side plots of quantiles of centered fit and residuals	RFPLOT
ScorePlot	Loess fit evaluated at scoring points	SCOREPLOT
CriterionPlot	Selection criterion versus smoothing parameter	CRITERION

Examples: LOESS Procedure

Example 50.1: Engine Exhaust Emissions

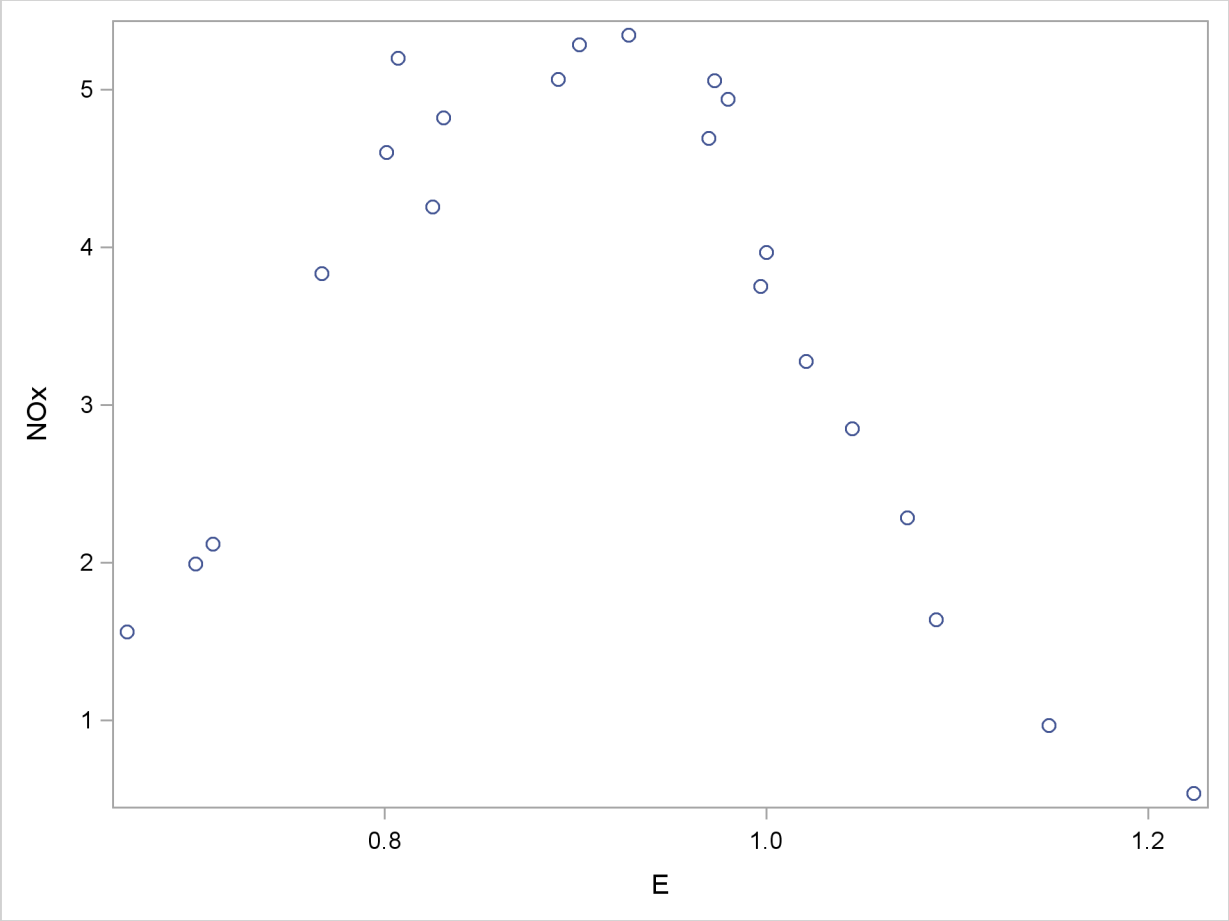
Investigators studied the exhaust emissions of a one-cylinder engine (Brinkman 1981). The SAS data set Gas contains the results data. The dependent variable, NOx, measures the concentration, in micrograms per joule, of nitric oxide and nitrogen dioxide normalized by the amount of work of the engine. The independent variable, E, is a measure of the richness of the air and fuel mixture.

```
data Gas;
   input NOx E @@;
   format NOx f3.1;
   format E f3.1;
datalines;
4.818  0.831   2.849  1.045
3.275  1.021   4.691  0.97
4.255  0.825   5.064  0.891
2.118  0.71    4.602  0.801
2.286  1.074   0.97   1.148
3.965  1       5.344  0.928
3.834  0.767   1.99   0.701
5.199  0.807   5.283  0.902
3.752  0.997   0.537  1.224
1.64   1.089   5.055  0.973
4.937  0.98    1.561  0.665
;
```

The following PROC SGPLOT statements produce the simple scatter plot of these data displayed in Output 50.1.1.

```
proc sgplot data=Gas;
   scatter x=E y=NOx;
run;
```

Output 50.1.1 Scatter Plot of the Gas Data



The following statements fit two loess models for these data. Because this is a small data set, it is reasonable to do direct fitting at every data point. As there is substantial curvature in the data, quadratic local polynomials are used. An ODS OUTPUT statement creates two output data sets containing the "Output Statistics" and "Fit Summary" tables.

```
ods graphics on;

proc loess data=Gas;
   ods output OutputStatistics = GasFit
              FitSummary=Summary;
   model NOx = E / degree=2 select=AICC(steps) smooth = 0.6 1.0
                   direct alpha=.01 all details;
run;

ods graphics off;
```

Output 50.1.2 Fit Summary Table

The LOESS Procedure
Selected Smoothing Parameter: 0.6
Dependent Variable: NOx

Fit Summary

Fit Method	Direct
Number of Observations	22
Degree of Local Polynomials	2
Smoothing Parameter	0.60000
Points in Local Neighborhood	13
Residual Sum of Squares	1.71852
Trace[L]	6.42184
GCV	0.00708
AICC	-0.45637
AICC1	-9.39715
Delta1	15.12582
Delta2	14.73089
Equivalent Number of Parameters	5.96950
Lookup Degrees of Freedom	15.53133
Residual Standard Error	0.33707

The "Fit Summary" table for smoothing parameter value 0.6, shown in Output 50.1.2, records the fitting parameters specified and some overall fit statistics. See the section "Smoothing Matrix" on page 3220 for a definition of the smoothing matrix L, and the sections "Model Degrees of Freedom" on page 3220 and "Statistical Inference and Lookup Degrees of Freedom" on page 3220 for definitions of the statistics that appear this table.

The "Output Statistics" table for smoothing parameter value 0.6 is shown in Output 50.1.3. Note that, because the ALL option is specified in the MODEL statement, this table includes all the relevant optional columns. Furthermore, because the ALPHA=0.01 option is specified in the MODEL statement, the confidence limits in this table are 99% limits.

Output 50.1.3 Output Statistics Table

The LOESS Procedure
Selected Smoothing Parameter: 0.6
Dependent Variable: NOx

Output Statistics

Obs	E	NOx	Predicted NOx	Estimated Prediction Std Deviation	Residual	t Value
1	0.8	4.8	4.87377	0.15528	−0.05577	−0.36
2	1.0	2.8	2.81984	0.15380	0.02916	0.19
3	1.0	3.3	3.48153	0.15187	−0.20653	−1.36
4	1.0	4.7	4.73249	0.13923	−0.04149	−0.30
5	0.8	4.3	4.82305	0.15278	−0.56805	−3.72
6	0.9	5.1	5.18561	0.19337	−0.12161	−0.63
7	0.7	2.1	2.51120	0.15528	−0.39320	−2.53
8	0.8	4.6	4.48267	0.15285	0.11933	0.78
9	1.1	2.3	2.12619	0.16683	0.15981	0.96
10	1.1	1.0	0.97120	0.18134	−0.00120	−0.01
11	1.0	4.0	4.09987	0.13477	−0.13487	−1.00
12	0.9	5.3	5.31258	0.17283	0.03142	0.18
13	0.8	3.8	3.84572	0.14929	−0.01172	−0.08
14	0.7	2.0	2.26578	0.16712	−0.27578	−1.65
15	0.8	5.2	4.58394	0.15363	0.61506	4.00
16	0.9	5.3	5.24741	0.19319	0.03559	0.18
17	1.0	3.8	4.16979	0.13478	−0.41779	−3.10
18	1.2	0.5	0.53059	0.32170	0.00641	0.02
19	1.1	1.6	1.83157	0.17127	−0.19157	−1.12
20	1.0	5.1	4.66733	0.13735	0.38767	2.82
21	1.0	4.9	4.52385	0.13556	0.41315	3.05
22	0.7	1.6	1.19888	0.26774	0.36212	1.35

Output Statistics

Obs	99% Confidence Limits	
1	4.41841	5.32912
2	2.36883	3.27085
3	3.03617	3.92689
4	4.32419	5.14079
5	4.37503	5.27107
6	4.61855	5.75266
7	2.05585	2.96655
8	4.03444	4.93089
9	1.63697	2.61541
10	0.43942	1.50298
11	3.70467	4.49507
12	4.80576	5.81940
13	3.40794	4.28350
14	1.77571	2.75584
15	4.13342	5.03445
16	4.68089	5.81393
17	3.77457	4.56502
18	−0.41278	1.47397
19	1.32933	2.33380
20	4.26456	5.07010
21	4.12632	4.92139
22	0.41375	1.98401

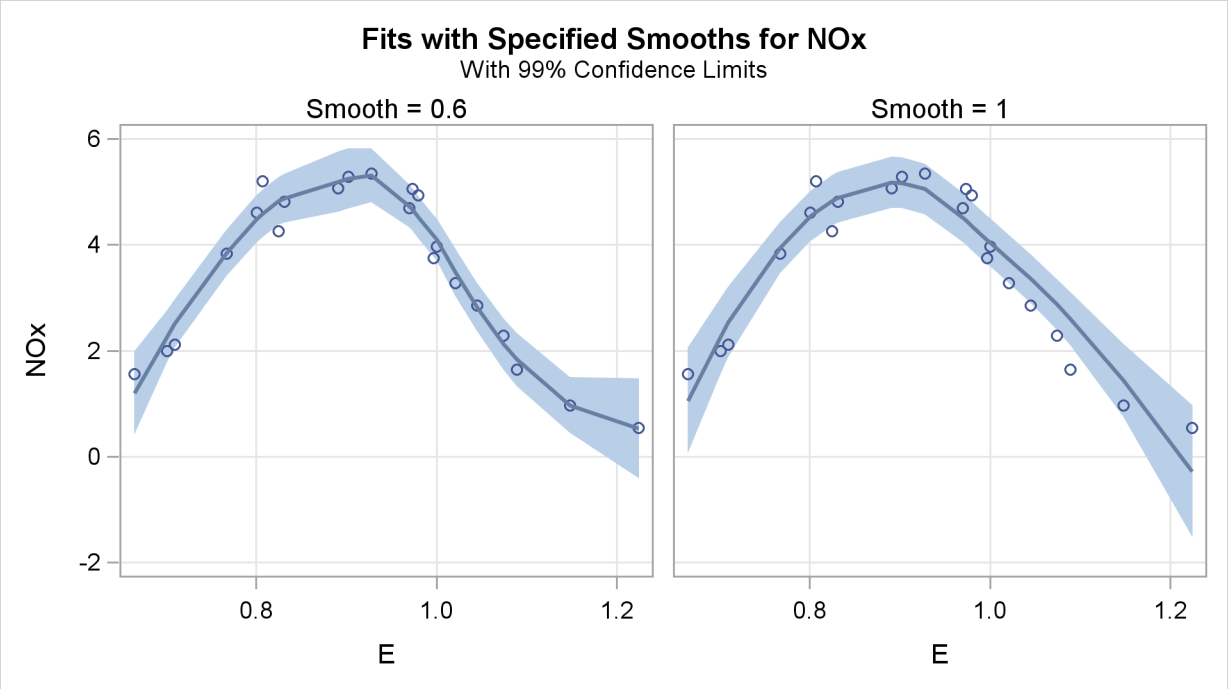
Output 50.1.4 Output Statistics Table

Optimal Smoothing Criterion	
AICC	Smoothing Parameter
−0.45637	0.60000

The combination of the options SELECT=AICC and SMOOTH=0.6 1 in the MODEL statement specifies that PROC LOESS fit models with smoothing parameters of 0.6 and 1 and select the model that yields the smaller value of the AICC statistic. The "Smoothing Criterion" shown in Output 50.1.4 shows that PROC LOESS selects the model with smoothing parameter value 0.6 as it yields the smaller value of the AICC statistic.

With ODS Graphics enabled, PROC LOESS produces a panel of fit plots whenever you specify the SMOOTH= option in the MODEL statement. These fit plots include confidence limits if you additionally specify the CLM option in the MODEL statement.

Output 50.1.5 Loess Fits with 99% Confidence Limits for the Gas Data



Output 50.1.5 shows the "Fit Panel" that displays the fitted models with 99% confidence limits overlaid on scatter plots of the data.

Based on the AICC criterion, the model with smoothing parameter 0.6 is preferred. You can address the question of whether the differences between these models are significant using analysis of variance. You do this by using the model with smoothing parameter value 1 as the null model.

The statistic

$$F = \frac{(\text{rss}^{(n)} - \text{rss})/(\delta_1^{(n)} - \delta_1)}{\text{rss}/\delta_1}$$

has a distribution that is well approximated by an F distribution with

$$\nu = \frac{(\delta_1^{(n)} - \delta_1)^2}{\delta_2^{(n)} - \delta_2}$$

numerator degrees of freedom and ρ denominator degrees of freedom (Cleveland and Grosse 1991). Here quantities with superscript n refer to the null model, rss is the residual sum of squares, and δ_1, δ_2, and ρ are defined in the section "Statistical Inference and Lookup Degrees of Freedom" on page 3220.

The "Fit Summary" tables contain the information needed to carry out such an analysis. These tables have been captured in the output data set named Summary by using an ODS OUTPUT statement. The following statements extract the relevant information from this data set and carry out the analysis of variance:

```
data h0 h1;
  set Summary(keep=SmoothingParameter Label1 nValue1
              where=(Label1 in ('Residual Sum of Squares','Delta1',
                     'Delta2','Lookup Degrees of Freedom')));
  if SmoothingParameter = 1 then output h0;
  else output h1;
run;

proc transpose data=h0(drop=SmoothingParameter Label1) out=h0;

data h0(drop=_NAME_); set h0;
  rename Col1 = RSSNull
         Col2 = delta1Null
         Col3 = delta2Null;

proc transpose data=h1(drop=SmoothingParameter Label1) out=h1;

data h1(drop=_NAME_); set h1;
  rename Col1 = RSS     Col2 = delta1
         Col3 = delta2  Col4 = rho;

data ftest; merge h0 h1;
  nu = (delta1Null - delta1)**2 / (delta2Null - delta2);
  Numerator = (RSSNull - RSS)/(delta1Null - delta1);
  Denominator = RSS/delta1;
  FValue = Numerator / Denominator;
  PValue = 1 - ProbF(FValue, nu, rho);
  label nu     = 'Num DF'   rho    = 'Den DF'
        FValue = 'F Value'  PValue = 'Pr > F';

proc print data=ftest label;
  var nu rho Numerator Denominator FValue PValue;
  format nu rho FValue 7.2 PValue 6.4;
run;
```

The results are shown in Output 50.1.6.

Output 50.1.6 Test ANOVA for Loess Models of Gas Data

Obs	Num DF	Den DF	Numerator	Denominator	F Value	Pr > F
1	2.67	15.53	1.05946	0.11362	9.32	0.0012

The small p-value confirms that the fit with smoothing parameter value 0.6 is significantly different from the loess model with smoothing parameter value 1.

Example 50.2: Sulfate Deposits in the U.S. for 1990

The following data set contains measurements in grams per square meter of sulfate (SO_4) deposits during 1990 at 179 sites throughout the 48 contiguous states.

```
data SO4;
   input Latitude Longitude SO4 @@;
   format Latitude f4.0;
   format Longitude f4.0;
   format SO4 f4.1;
datalines;
32.45833  87.24222 1.403 34.28778  85.96889 2.103
33.07139 109.86472 0.299 36.07167 112.15500 0.304
31.95056 112.80000 0.263 33.60500  92.09722 1.950

... more lines ...

43.87333 104.19222 0.306 44.91722 110.42028 0.210
45.07611  72.67556 2.646
;
```

As longitudes decrease from west to east in the western hemisphere, the roles of east and west get interchanged if you use these longitudes on the horizontal axis of a plot. You can address this by using negative values to represent longitudes in the western hemisphere. The following statements change the sign of longitude in the SO4 data set and define a format to display these negative values with a suffix of "W".

```
proc format;
   picture latitude  -90 - 0  = '000S'
                      0  - 90 = '000N';
   picture longitude -180  - 0    = '000W'
                      0    - 180  = '000E';
run;
data SO4;
  set SO4;
  format longitude longitude. latitude  latitude.;
  longitude = -longitude;
run;
```

The following statements use ODS Graphics to plot the locations of the sulfate measurements. The circles indicating the locations are colored using a gradient that denotes the value of SO4.

```
proc template;
   define statgraph gradientScatter;
      beginGraph;
         layout overlay;
            scatterPlot x=longitude y=latitude /
               markercolorgradient = SO4
               markerattrs         = (symbol=circleFilled)
               colormodel          =  ThreeColorLowHigh
               name                =  "Scatter";
            scatterPlot x=longitude y=latitude /
               markerattrs         = (symbol=circle);
            continuousLegend "Scatter"/title= "SO4";
         endlayout;
      endgraph;
   end;
run;

proc sgrender data=SO4 template=gradientScatter;run;
```

Output 50.2.1 Sulfate Measurements

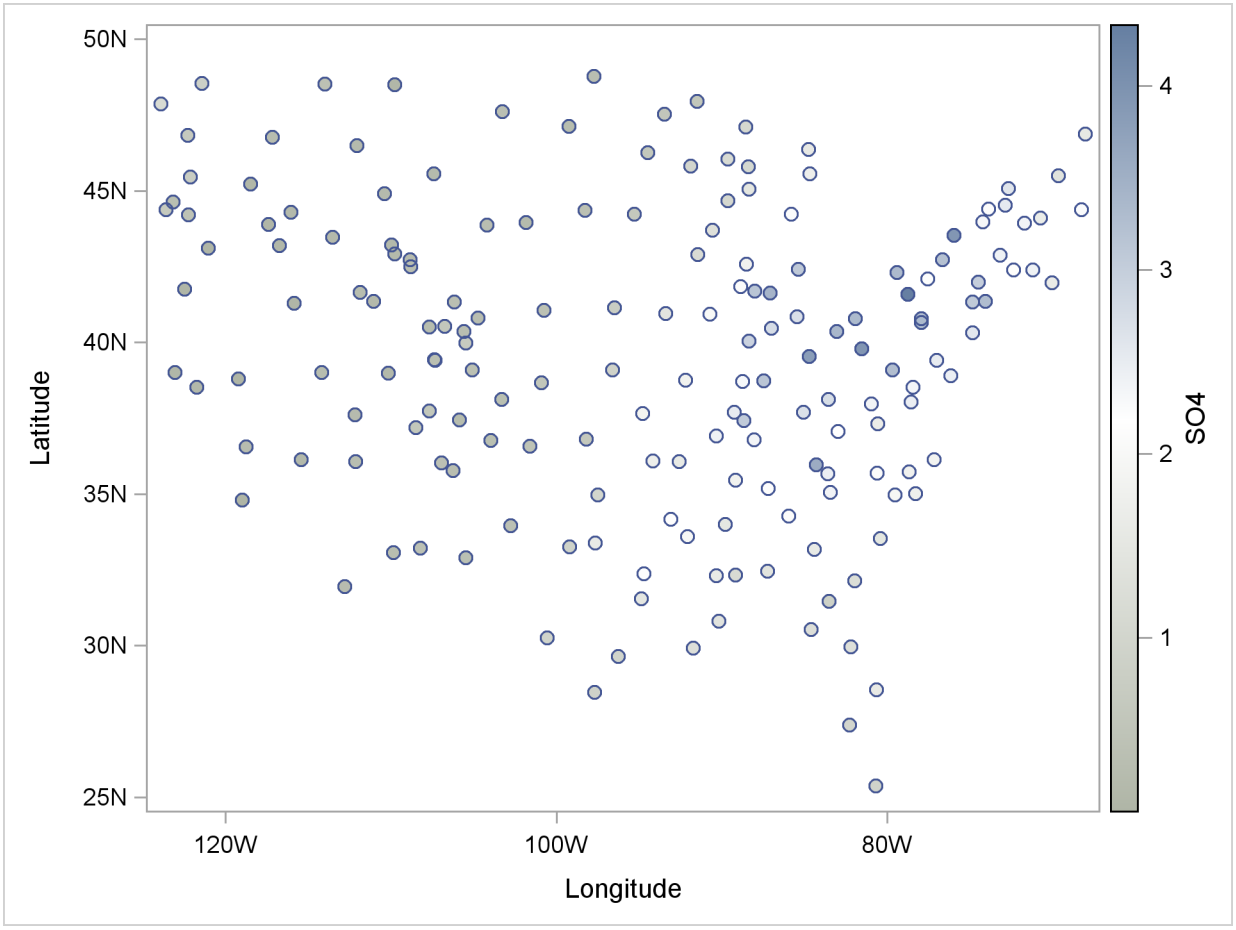


Figure 50.2.1 shows that the largest concentrations of sulfate deposits occur in the northeastern United States.

The following statements fit a loess model.

```
ods graphics on;

proc loess data=SO4;
   model SO4=Longitude Latitude / degree=2 interp=cubic;
run;

ods graphics off;
```

Output 50.2.2 Fit Plot for the SO4 Data

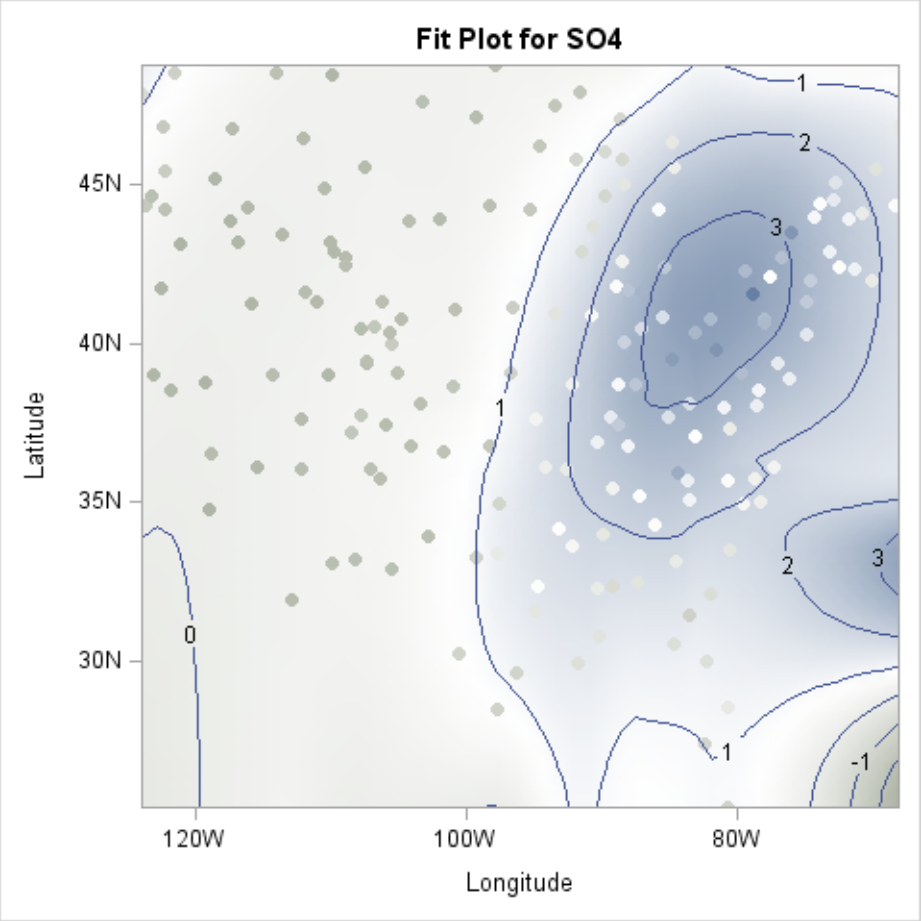


Figure 50.2.2 shows a contour plot of the fitted loess surface overlaid with a scatter plot of the data. The data are colored by the observed sulfate concentrations, using the same color gradient as the gradient-filled contour plot of the fitted surface. Note that for observations where the residual is small, the observations blend in with the contour plot. The greater the size of the residual, the greater the contrast between the observation color and the surface color.

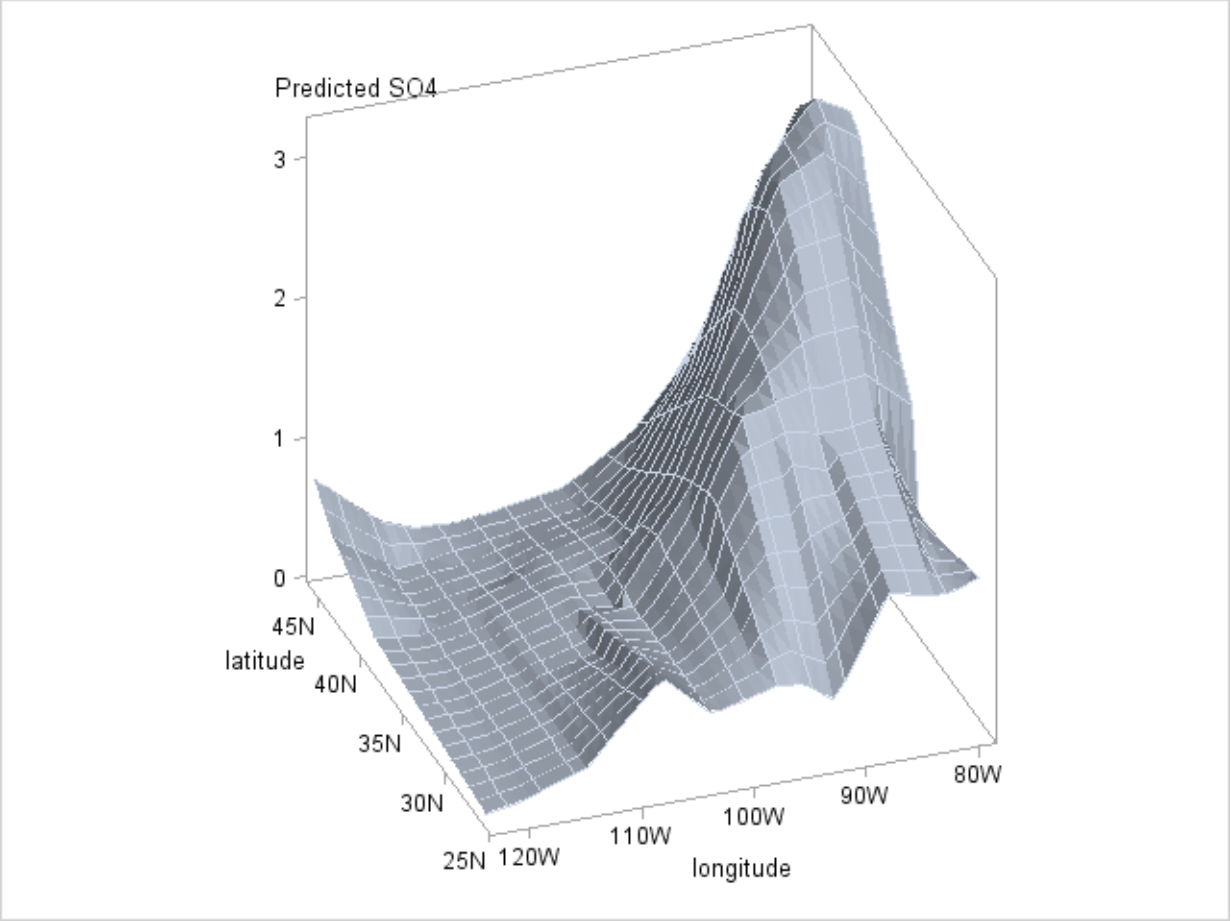
The sulfate measurements are irregularly spaced. To facilitate producing a plot of the fitted loess surface, you can create a data set containing a regular grid of longitudes and latitudes and then use the SCORE statement to evaluate the loess surface at these points. The following statements show how you do this:

```
data PredPoints;
  format longitude longitude.
         latitude  latitude.;
   do Latitude = 26 to 46 by 1;
      do Longitude = -79 to -123 by -1;
         output;
      end;
   end;
run;

proc loess data=SO4;
   model SO4=Longitude Latitude;
   score data=PredPoints / print;
   ods Output ScoreResults=ScoreOut;
run;
```

The PRINT option in the SCORE statement requests that the "Score Results" table be displayed as part of the PROC LOESS output. The ODS OUTPUT statement outputs this table to a data set named ScoreOut. If you do not want to display the score results but you do want the score results in an output data set, then you can omit the PRINT option from the SCORE statement. To plot the surface shown in Figure 50.2.3 by using ODS Graphics, use the following statements:

```
proc template;
   define statgraph surface;
      begingraph;
         layout overlay3d / rotate=340 tilt=30 cube=false;
            surfaceplotparm x=Longitude y=Latitude z=p_SO4;
         endlayout;
      endgraph;
   end;
run;

proc sgrender data=ScoreOut template=surface;
run;
```

Output 50.2.3 Loess Fit of SO4 Surface



Example 50.3: Catalyst Experiment

The following data set records the results of an experiment to determine how the yield of a chemical reaction varies with temperature and amount of a catalyst used.

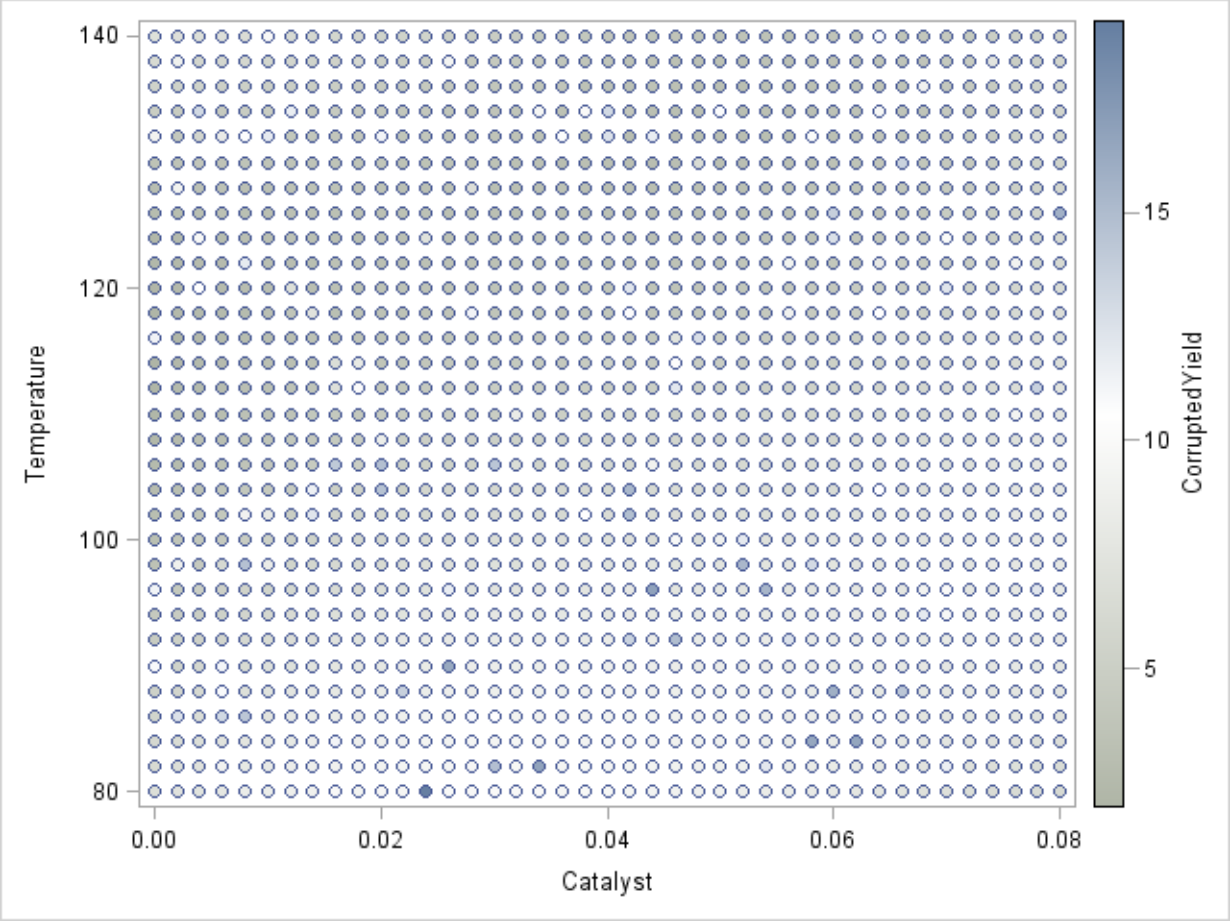
```
data Experiment;
   input Temperature Catalyst MeasuredYield;
   if ranuni(1) < 0.1
      then CorruptedYield = MeasuredYield + 10 * ranuni(1);
      else CorruptedYield = MeasuredYield;
datalines;
 80     0.000      6.85601
 80     0.002      7.26355
 80     0.004      7.41448

... more lines ...

140     0.078      5.20562
140     0.080      5.49371
;
```

The aim of this example is to show how you can use PROC LOESS for robust fitting in the presence of outliers. To simulate an intermittent equipment malfunction, the variable CorruptedYield is the same as the variable MeasuredYield except for about 10% of the observations where an offset has been added. This example shows how you can use PROC LOESS obtain a fit for CorruptedYield that is close to the fit you obtain for MeasuredYield.

The following statements produce a scatter plot of Temperature by Catalyst where the observations are colored by CorruptedYield:

```
proc template;
  define statgraph gradientScatter;
    beginGraph;
      layout overlay;
        scatterPlot x=Catalyst y=Temperature /
            markercolorgradient = CorruptedYield
            markerattrs         = (symbol=circleFilled)
            colormodel          =  ThreeColorLowHigh
            name                =  "Yield";

        scatterPlot x=Catalyst y=Temperature /
            markerattrs         = (symbol=circle);

         continuousLegend "Yield" / title= "CorruptedYield";
      endlayout;
    endgraph;
  end;
run;

proc sgrender data=Experiment template=gradientScatter;run;
```

Output 50.3.1 Scatter Plot of Experiment Data Colored by CorruptedYield



Output 50.3.1 shows a scatter plot of the data where the observations are shaded by the value of CorruptedYield. The darkly shaded points that are surrounded by lightly shaded points are points where the simulated incorrect measurements occur.

The following code fits a loess model to the measured data:

```
ods graphics on;

proc loess data=Experiment;
   model MeasuredYield = Temperature Catalyst / scale=sd(0.1);
run;
```

Output 50.3.2 Scale Details for the Experiment Data

The LOESS Procedure

Independent Variable Scaling

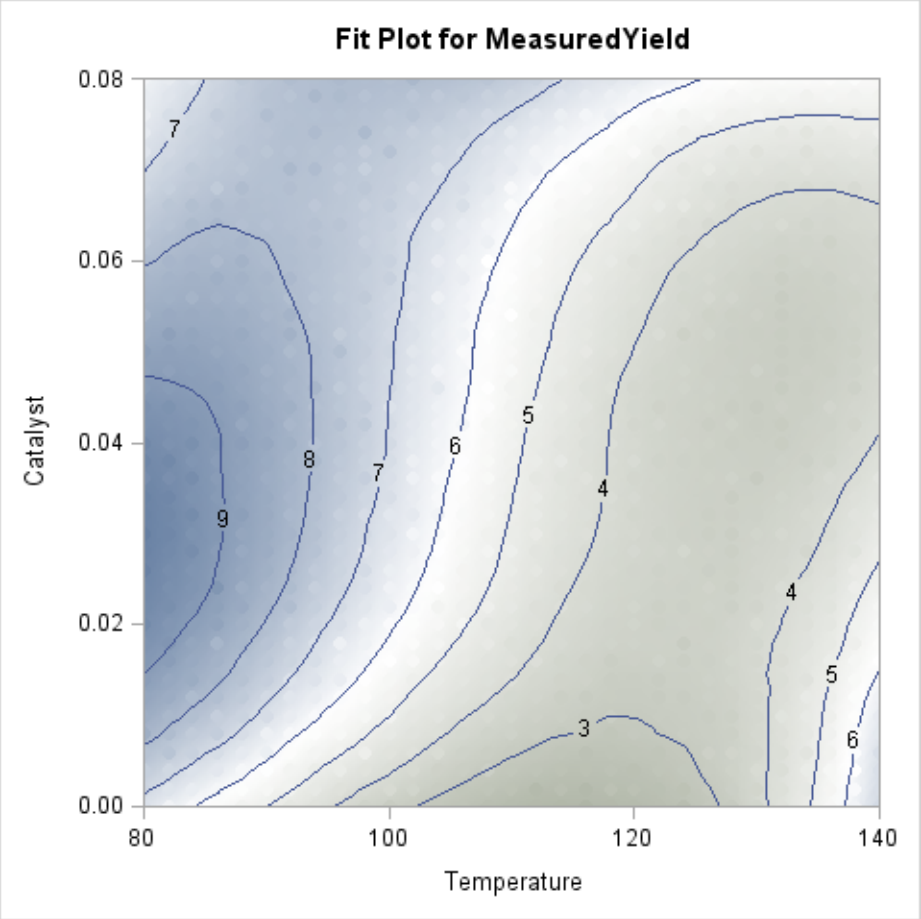
Scaling applied: 10% trimmed standard deviation

Statistic	Temperature	Catalyst
Minimum Value	80.00000	0
Maximum Value	140.00000	0.08000
Trimmed Mean	110.00000	0.04000
Trimmed Standard Deviation	14.32149	0.01894

The SCALE=SD(0.1) option in the MODEL statement specifies that the independent variables in the model are to be divided by their respective 10% trimmed standard deviations before the fitted model is computed. This is appropriate because the independent variables Temperature and Catalyst are not similarly scaled. The "Scale Details" table in Output 50.3.2 displays the details of ranges of the regressors and the scale factors applied to each regressor.

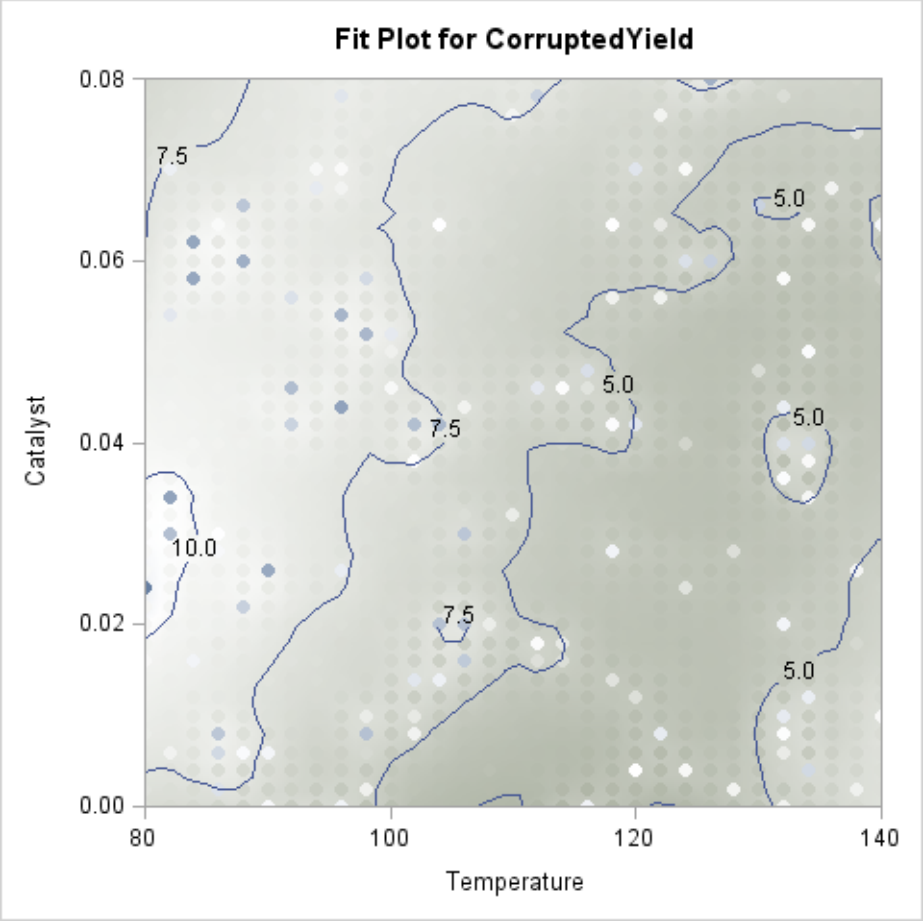
Output 50.3.3 displays the loess fit. Because the fitted surface is a good fit of the observed data, the observations on this plot are not clearly distinguishable from the fitted surface. The results are dramatically different when the outliers are included. The following statements fit a loess model to the corrupted response, using the same smoothing parameter that was selected for the measured response.

Output 50.3.3 Fit for MeasuredYield




```
proc loess data=Experiment;
   model CorruptedYield = Temperature Catalyst /
                  scale=sd(0.1) smooth=0.018;
run;
```

Output 50.3.4 Fit for CorruptedYield

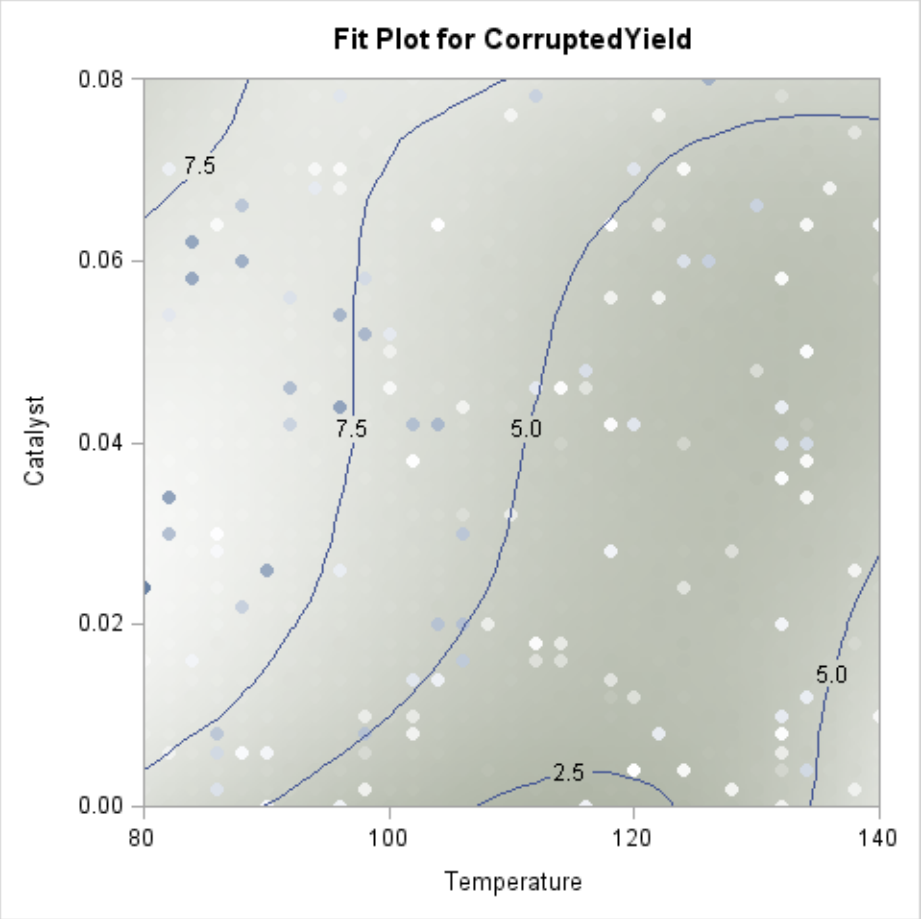


Output 50.3.4 displays the loess fit. The fit is pulled upward in the neighborhoods of these outliers. If you use a larger smoothing parameter value, then these local perturbations in the fit get smoothed out, but at the expense of smoothing away the information in the underlying measured response. In such cases a robust fitting method is indicated. The following statements show how you do this:

```
proc loess data=Experiment;
   model CorruptedYield = Temperature Catalyst /
                          scale  = sd(0.1)
                          smooth = 0.018
                          iterations=4;
run;
```

The ITERATIONS=4 option in the MODEL statement requests the initial loess fit followed by three iteratively reweighted iterations.

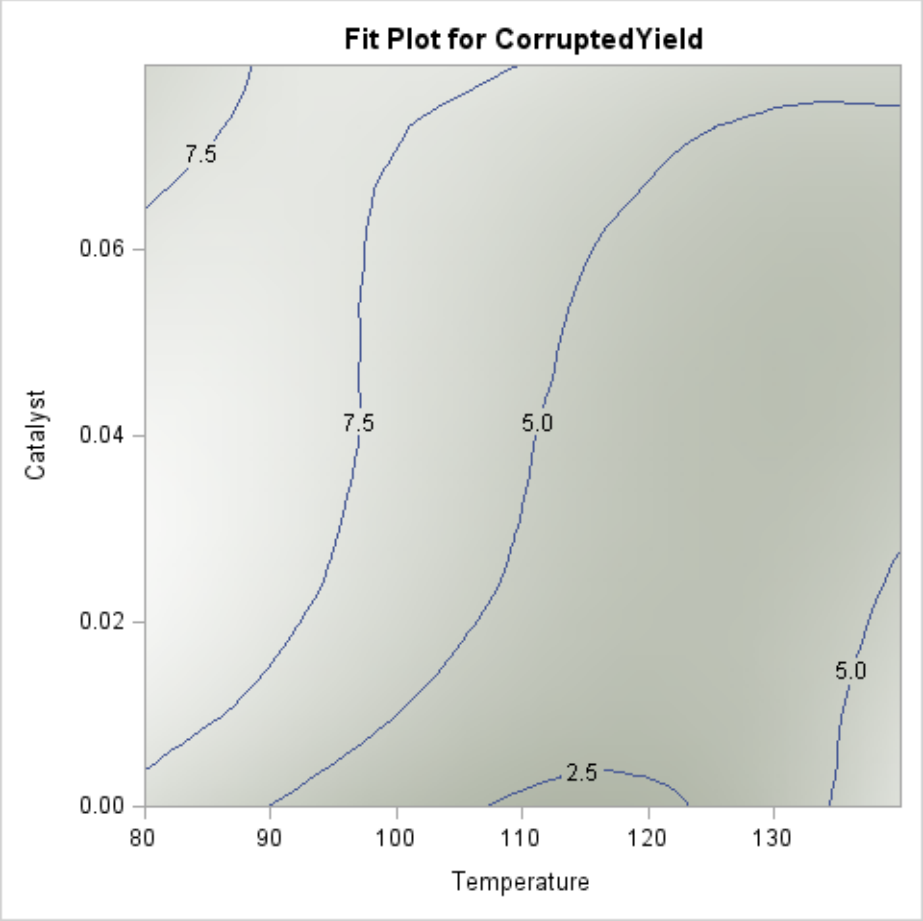
Output 50.3.5 Robust Fit for CorruptedYield



You can see the impact of the robust fitting by comparing the robust fit shown in Output 50.3.5 with the nonrobust fit in Output 50.3.4. In the robust fit you see that the local perturbations caused by the outliers have been eliminated as these the outlying observations get down-weighted during the robustness iterations. By comparing the labeled contours on the fit plot for the uncorrupted response shown in Output 50.3.3 with the labeled contours for the corrupted response shown in Output 50.3.4, you can see that the robust fit has produced a reasonable fit for the underlying measured data. The color gradient in Output 50.3.5 is chosen to accommodate the outliers that are present in the observed data, and so you cannot easily compare the color gradient in this plot with that in Output 50.3.3. The following statements repeat the robust analysis with an option added to suppress the display of the observations on the fit plot:

```
proc loess data=Experiment plots=contourFit(obs=none);
   model CorruptedYield = Temperature Catalyst /
                          scale  = sd(0.1)
                          smooth = 0.018
                          iterations=4;
run;

ods graphics off;
```

Output 50.3.6 Robust Fit for CorruptedYield with Observations Suppressed



Output 50.3.6 shows the robust fit with the observations suppressed. The range of the fitted surface values in this plot is similar to the range in Output 50.3.3. By comparing this contour plot with the contour plot in Output 50.3.3, you clearly see that the robust loess fit has successfully modeled the underlying surface despite the presence of the outliers.

Example 50.4: El Niño Southern Oscillation

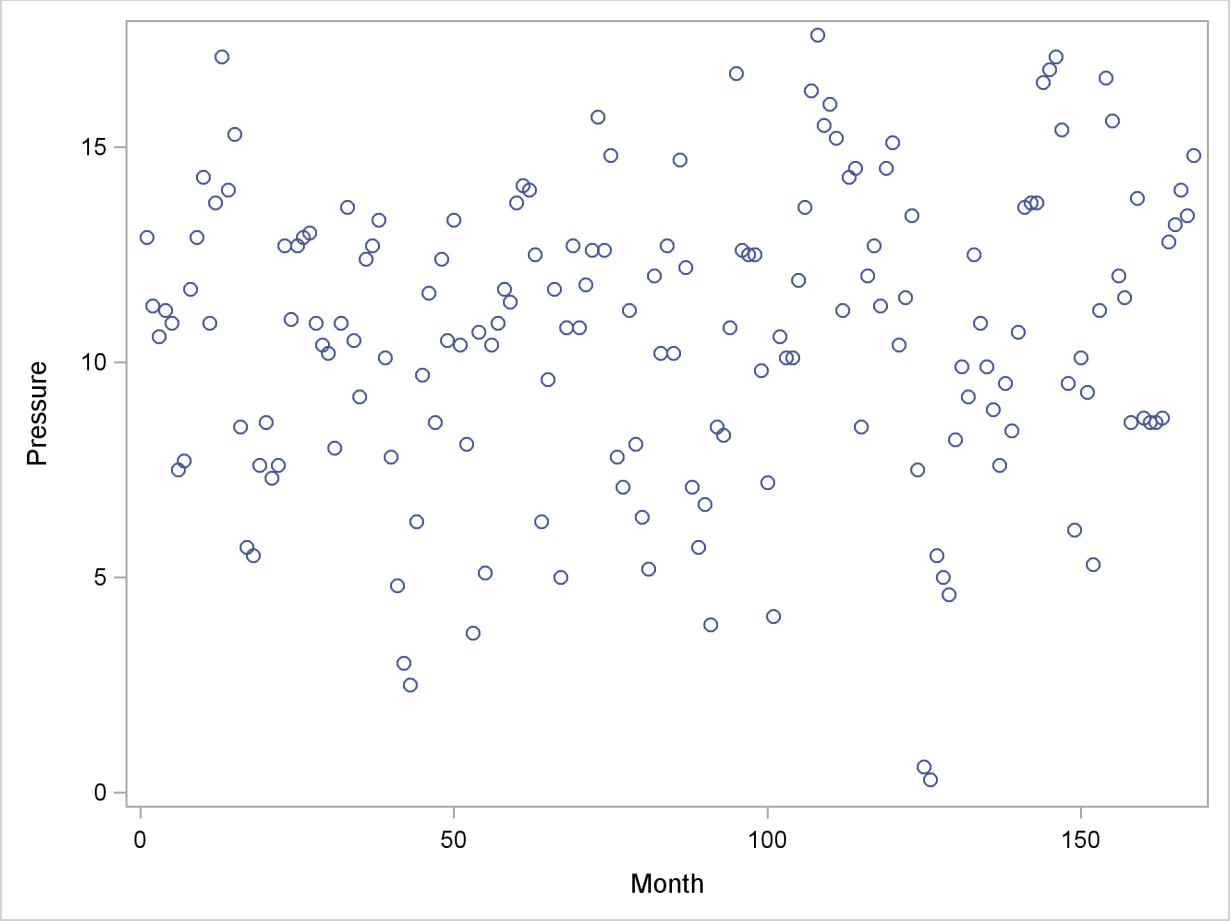
The following data set contains measurements of monthly averaged atmospheric pressure differences between Easter Island and Darwin, Australia, for a period of 168 months (National Institute of Standards and Technology 1998):

```
data ENSO;
  input Pressure @@;
  Month=_N_;
  format Pressure 4.1;
  format Month 3.0;
datalines;
12.9  11.3  10.6  11.2  10.9   7.5   7.7  11.7
12.9  14.3  10.9  13.7  17.1  14.0  15.3   8.5
 5.7   5.5   7.6   8.6   7.3   7.6  12.7  11.0
12.7  12.9  13.0  10.9  10.4  10.2   8.0  10.9
13.6  10.5   9.2  12.4  12.7  13.3  10.1   7.8
 4.8   3.0   2.5   6.3   9.7  11.6   8.6  12.4
10.5  13.3  10.4   8.1   3.7  10.7   5.1  10.4
10.9  11.7  11.4  13.7  14.1  14.0  12.5   6.3
 9.6  11.7   5.0  10.8  12.7  10.8  11.8  12.6
15.7  12.6  14.8   7.8   7.1  11.2   8.1   6.4
 5.2  12.0  10.2  12.7  10.2  14.7  12.2   7.1
 5.7   6.7   3.9   8.5   8.3  10.8  16.7  12.6
12.5  12.5   9.8   7.2   4.1  10.6  10.1  10.1
11.9  13.6  16.3  17.6  15.5  16.0  15.2  11.2
14.3  14.5   8.5  12.0  12.7  11.3  14.5  15.1
10.4  11.5  13.4   7.5   0.6   0.3   5.5   5.0
 4.6   8.2   9.9   9.2  12.5  10.9   9.9   8.9
 7.6   9.5   8.4  10.7  13.6  13.7  13.7  16.5
16.8  17.1  15.4   9.5   6.1  10.1   9.3   5.3
11.2  16.6  15.6  12.0  11.5   8.6  13.8   8.7
 8.6   8.6   8.7  12.8  13.2  14.0  13.4  14.8
;
```

The following PROC SGPLOT statements produce the simple scatter plot of these data, displayed in Output 50.4.1.

```
proc sgplot data=ENSO;
   scatter y=Pressure x=Month;
run;
```

Output 50.4.1 Scatter Plot of ENSO Data



You can compute a loess fit and obtain graphical results for these data by using the following statements:

```
ods graphics on;

proc loess data=ENSO plots=residuals(smooth);
   model Pressure=Month;
run;
```

The "Smoothing Criterion" and "Fit Summary" tables are shown in Output 50.4.2, and the fit plot is shown in Output 50.4.3.

Output 50.4.2 Output from PROC LOESS

The LOESS Procedure
Dependent Variable: Pressure

Optimal Smoothing Criterion

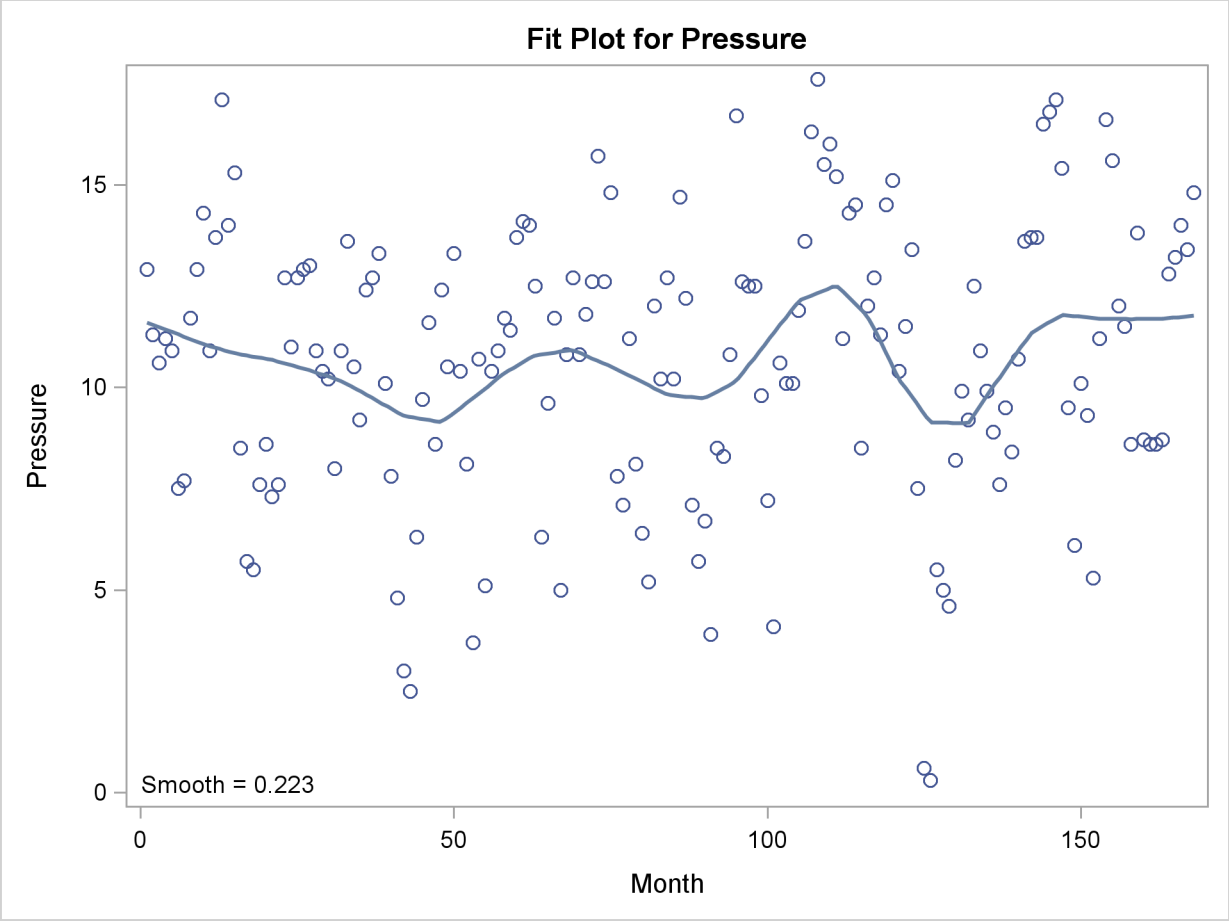
AICC	Smoothing Parameter
3.41105	0.22321

The LOESS Procedure
Selected Smoothing Parameter: 0.223
Dependent Variable: Pressure

Fit Summary

Fit Method	kd Tree
Blending	Linear
Number of Observations	168
Number of Fitting Points	33
kd Tree Bucket Size	7
Degree of Local Polynomials	1
Smoothing Parameter	0.22321
Points in Local Neighborhood	37
Residual Sum of Squares	1654.27725
Trace[L]	8.74180
GCV	0.06522
AICC	3.41105

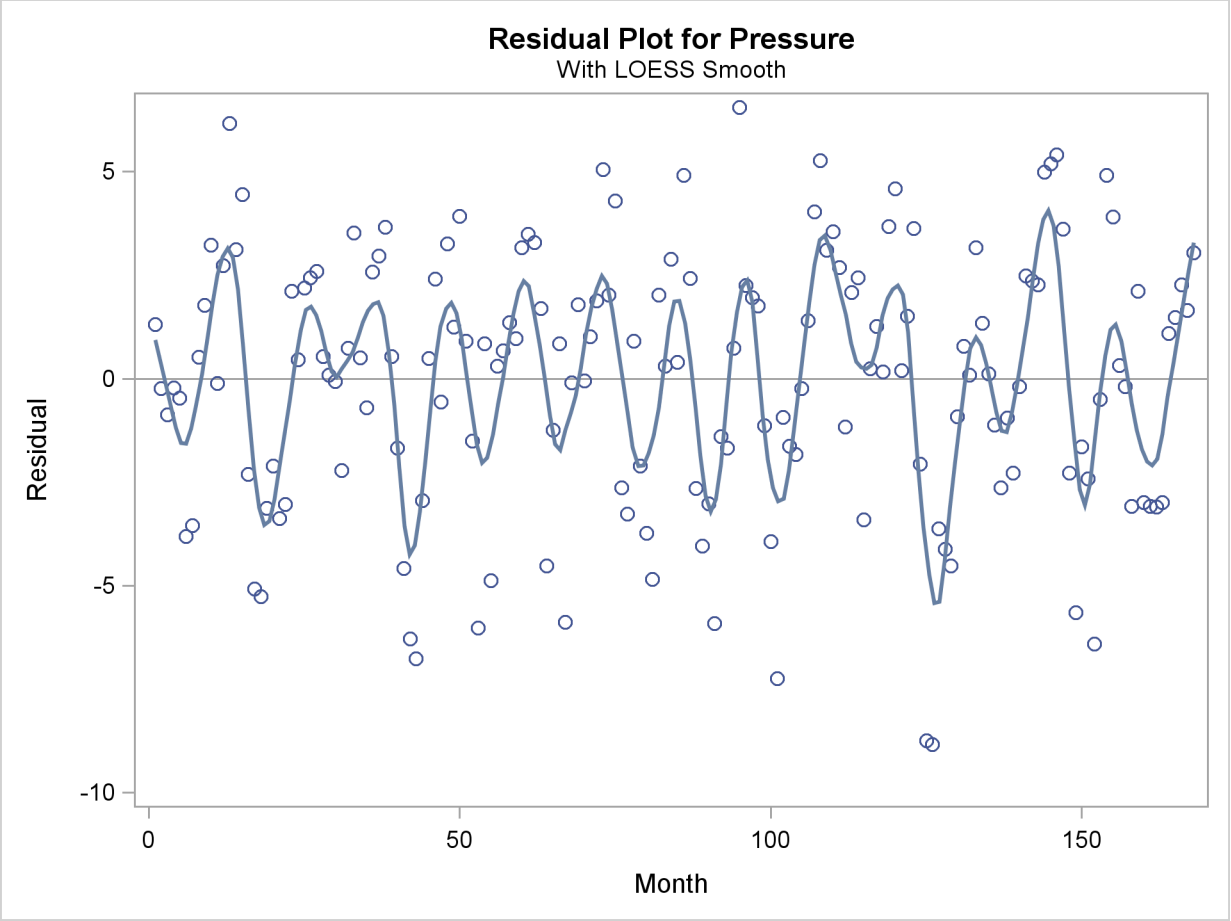
Output 50.4.3 Oversmoothed Loess Fit for the ENSO Data



This weather-related data should exhibit an annual cycle. However, the loess fit in Output 50.4.3 indicates a longer cycle but no annual cycle. This suggests that the loess fit is oversmoothed. One way to detect oversmoothing is to look for patterns in the fit residuals. With ODS Graphics enabled, PROC LOESS produces a scatter plot of the residuals versus each regressor in the model. To aid in visually detecting patterns in these scatter plots, it is useful to superimpose a nonparametric fit on these scatter plots. You can request this by specifying the SMOOTH suboption of the PLOTS=RESIDUALS option in the PROC LOESS statement. The nonparametric fit that is produced is again a loess fit that is produced independently of the loess fit used to obtain these residuals.

With the superimposed loess fit shown in Output 50.4.4, you can clearly identify an annual cycle in the residuals, which confirms that the loess fit for the ENSO is oversmoothed. What accounts for this poor fit?

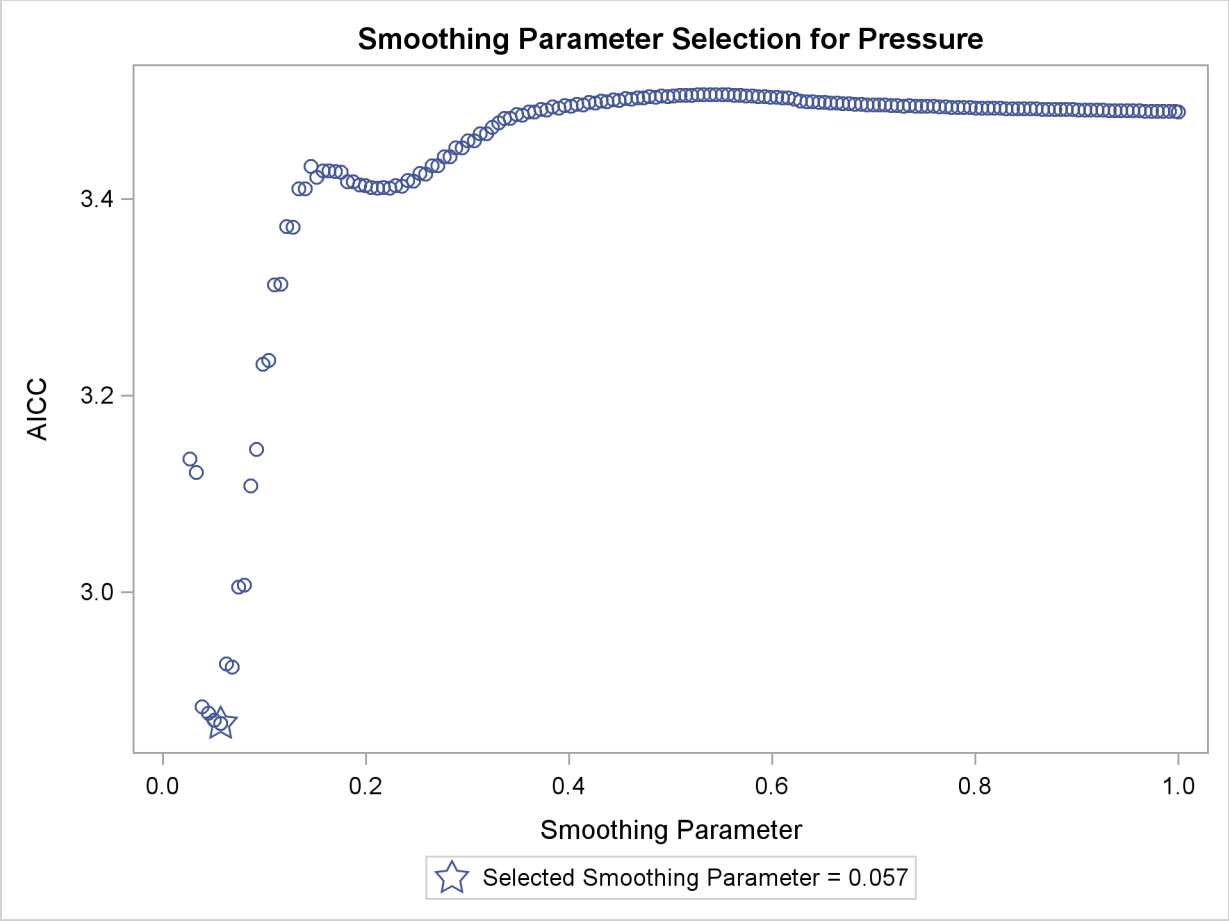
Output 50.4.4 Residuals for the Loess Fit for the ENSO Data



The smoothing parameter value used for the loess fit shown in Output 50.4.3 was chosen using the default method of PROC LOESS, namely a golden section minimization of the AICC criterion over the interval $(0, 1]$. One possibility is that the golden section search has found a local rather than a global minimum of the AICC criterion. You can test this by redoing the fit requesting a global minimum. You do this with the following statements:

```
proc loess data=ENSO;
   model Pressure=Month/select=AICC(global);
run;
```

Output 50.4.5 AICC versus Smoothing Parameter Showing Local Minima



The explanation for the oversmoothed fit in Output 50.4.3 is now apparent. Output 50.4.5 shows that the golden section search algorithm found the local minimum that occurs near the value 0.22 of the smoothing parameter rather than the global minimum that occurs near 0.06. Note that if you restrict the range of smoothing parameter values examined to lie below 0.2, then the golden section search finds the global minimum, as the following statements demonstrate:

```
proc loess data=ENSO;
   model Pressure=Month/select=AICC(range(0.03,0.2));
run;
```

Output 50.4.6 Selected Smoothing Parameter Value

The LOESS Procedure
Dependent Variable: Pressure

Optimal Smoothing Criterion

AICC	Smoothing Parameter
2.86660	0.05655

Output 50.4.6 shows that with the restricted range of smoothing parameter values examined, PROC LOESS finds the global minimum of the AICC criterion. Often you might not know an appropriate range of smoothing parameter values to examine. In such cases, you can use the PRESEARCH suboption of the SELECT= option in the MODEL statement. When you specify this option, PROC LOESS does a preliminary search to try to locate a smoothing parameter value range that contains just the first local minimum of the criterion being used for the selection. The following statements provide an example.

```
proc loess data=ENSO plots=residuals(smooth);
   model Pressure=Month/select=AICC(presearch);
run;

ods graphics off;
```

Output 50.4.7 Selected Smoothing Parameter Value When Presearch Is Specified

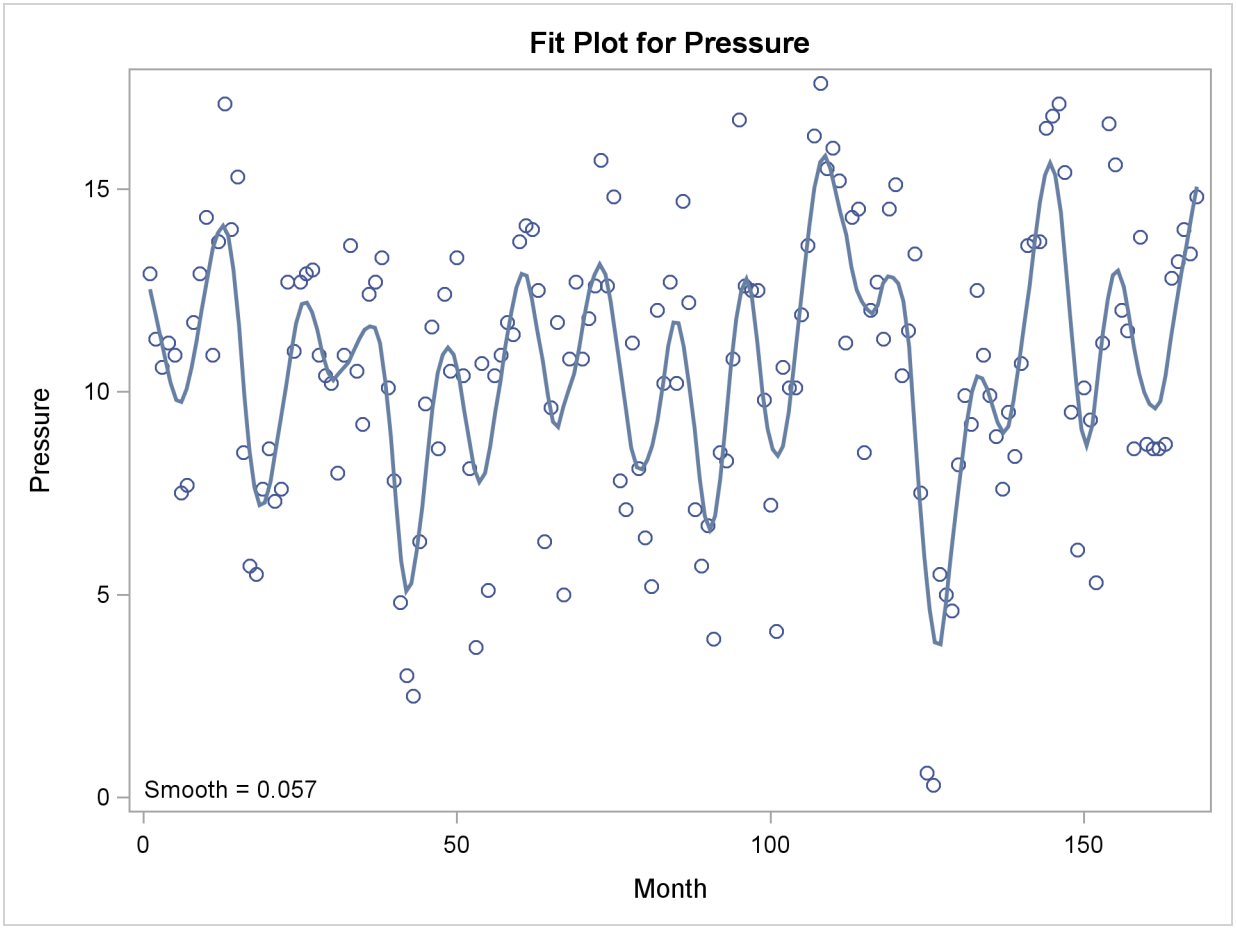
The LOESS Procedure
Dependent Variable: Pressure

Optimal Smoothing Criterion

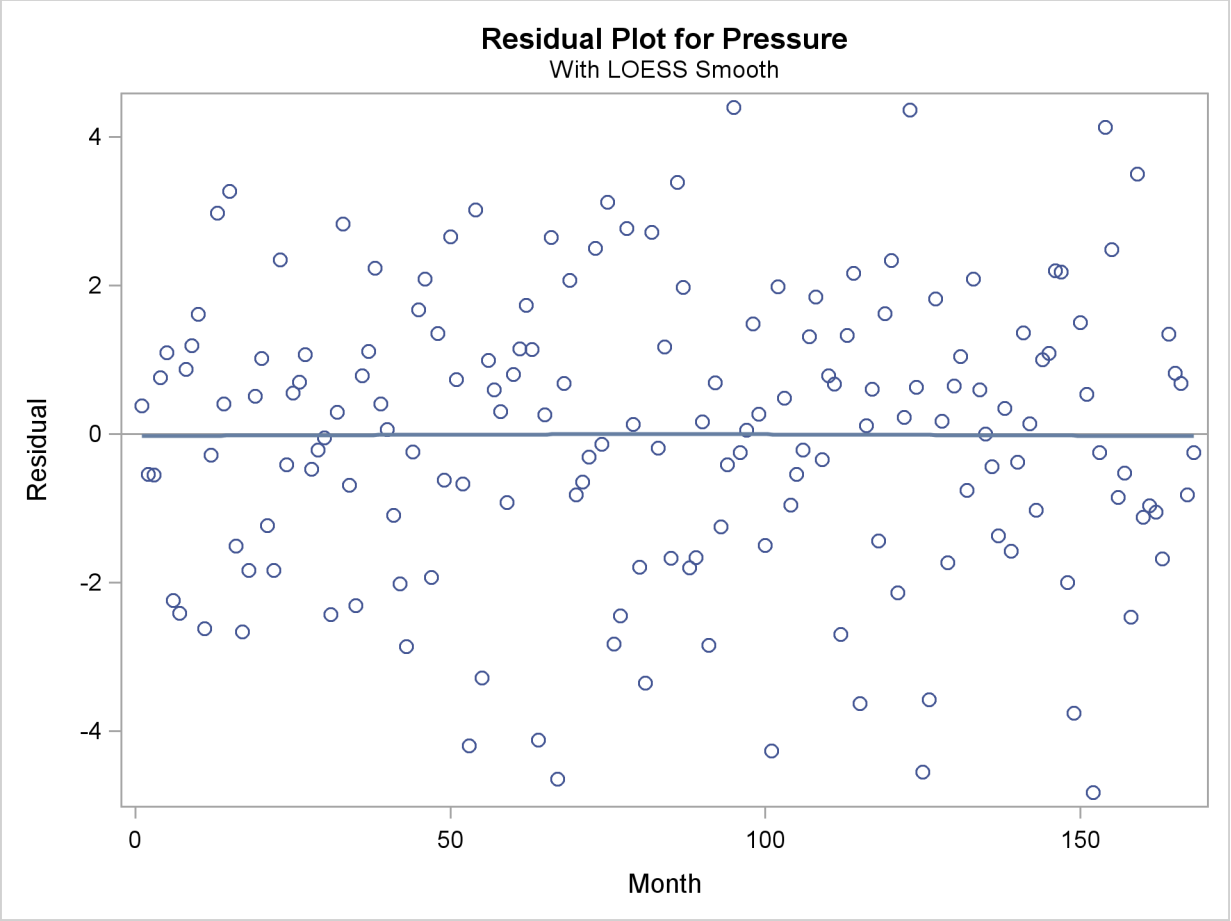
AICC	Smoothing Parameter
2.86660	0.05655

Output 50.4.7 shows that with the PRESEARCH suboption specified, PROC LOESS selects the smoothing parameter value that yields the global minimum of the AICC criterion. The fit obtained is shown in Output 50.4.8, and a plot of the residuals with a superimposed loess fit is shown in Output 50.4.9.

Output 50.4.8 Loess Fit Showing an Annual Cycle



Output 50.4.9 Residuals of the Selected Model



In contrast to the residual plot show in Output 50.4.4, the residuals plotted in Output 50.4.9 do not exhibit any pattern, indicating that the corresponding loess fit has captured all the systematic variation in the data.

An interesting question is whether there is some phenomenon captured in the data that would explain the presence of the local minimum near 0.22 in the AICC curve. Note that there is some evidence of a cycle of about 42 months in the oversmoothed fit in Output 50.4.3. You can see this cycle because the strong annual cycle in Output 50.4.8 has been smoothed out. The physical phenomenon that accounts for the existence of this cycle has been identified as the periodic warming of the Pacific Ocean known as "El Niño."

References

Akaike, H. (1973), "Information Theory and an Extension of the Maximum Likelihood Principle," in Petrov and Csaki, eds., *Proceedings of the Second International Symposium on Information Theory*, 267–281.

Brinkman, N. D. (1981), "Ethanol Fuel—A Single-Cylinder Engine Study of Efficiency and Exhaust Emissions," *Society of Automotive Engineers Transactions*, 90, 1410–1424.

Cleveland, W. S. (1993), *Visualizing Data*, Summit, NJ: Hobart Press.

Cleveland, W. S., Devlin, S. J., and Grosse, E. (1988), "Regression by Local Fitting," *Journal of Econometrics*, 37, 87–114.

Cleveland, W. S. and Grosse, E. (1991), "Computational Methods for Local Regression," *Statistics and Computing*, 1, 47–62.

Cleveland, W. S., Grosse, E., and Shyu, M.-J. (1992), "A Package of C and Fortran Routines for Fitting Local Regression Models," Unpublished.

Craven, P. and Wahba, G. (1979), "Smoothing Noisy Data with Spline Functions," *Numerical Mathematics*, 31, 377–403.

Gordon, W. J. (1971), "Blending-Function Methods of Bivariate and Multivariate Interpolation and Approximation," *SIAM Journal of Numerical Analysis*, 8, No. 1, 158–177.

Hastie, T. J. and Tibshirani, R. J. (1990), *Generalized Additive Models*, New York: Chapman & Hall.

Houghton, A. N., Flannery, J., and Viola, M. V. (1980), "Malignant Melanoma in Connecticut and Denmark," *International Journal of Cancer*, 25, 95–104.

Hurvich, C. M., Simonoff, J. S., and Tsai, C. L. (1998), "Smoothing Parameter Selection in Nonparametric Regression Using an Improved Akaike Information Criterion," *Journal of the Royal Statistical Society B*, 60, 271–293.

National Institute of Standards and Technology (1998), *Statistical Reference Data Sets*, last accessed April 4, 2007.
URL http://www.nist.gov/itl/div898/strd

Ruppert, D., Sheather, S. J., and Wand, M. P. (1995), "An Effective Bandwidth Selector for Local Least Squares Regression," *Journal of the American Statistical Association*, 90, 1257–1270.

Chapter 51

The LOGISTIC Procedure

Contents

Overview: LOGISTIC Procedure **3257**
Getting Started: LOGISTIC Procedure **3260**
Syntax: LOGISTIC Procedure **3266**
PROC LOGISTIC Statement 3267
BY Statement 3279
CLASS Statement 3280
CONTRAST Statement 3282
EXACT Statement 3285
FREQ Statement 3287
MODEL Statement 3288
ODDSRATIO Statement 3303
OUTPUT Statement 3305
ROC Statement 3310
ROCCONTRAST Statement 3311
SCORE Statement 3312
STRATA Statement 3314
TEST Statement 3315
UNITS Statement 3316
WEIGHT Statement 3317
Details: LOGISTIC Procedure **3318**
Missing Values 3318
Response Level Ordering 3318
CLASS Variable Parameterization 3319
Link Functions and the Corresponding Distributions 3322
Determining Observations for Likelihood Contributions 3324
Iterative Algorithms for Model Fitting 3324
Convergence Criteria 3327
Existence of Maximum Likelihood Estimates 3327
Effect-Selection Methods 3328
Model Fitting Information 3329
Generalized Coefficient of Determination 3330
Score Statistics and Tests 3331
Confidence Intervals for Parameters 3333
Odds Ratio Estimation 3335

Rank Correlation of Observed Responses and Predicted Probabilities 3338
Linear Predictor, Predicted Probability, and Confidence Limits 3339
Classification Table 3340
Overdispersion 3342
The Hosmer-Lemeshow Goodness-of-Fit Test 3344
Receiver Operating Characteristic Curves 3346
Testing Linear Hypotheses about the Regression Coefficients 3348
Regression Diagnostics 3349
Scoring Data Sets 3352
Conditional Logistic Regression 3355
Exact Conditional Logistic Regression 3359
Input and Output Data Sets 3364
Computational Resources 3369
Displayed Output 3371
ODS Table Names 3377
ODS Graphics 3379
Examples: LOGISTIC Procedure **3381**
Example 51.1: Stepwise Logistic Regression and Predicted Values 3381
Example 51.2: Logistic Modeling with Categorical Predictors 3396
Example 51.3: Ordinal Logistic Regression 3405
Example 51.4: Nominal Response Data: Generalized Logits Model 3412
Example 51.5: Stratified Sampling 3419
Example 51.6: Logistic Regression Diagnostics 3420
Example 51.7: ROC Curve, Customized Odds Ratios, Goodness-of-Fit Statistics, R-Square, and Confidence Limits 3429
Example 51.8: Comparing Receiver Operating Characteristic Curves 3434
Example 51.9: Goodness-of-Fit Tests and Subpopulations 3442
Example 51.10: Overdispersion 3445
Example 51.11: Conditional Logistic Regression for Matched Pairs Data . . 3449
Example 51.12: Firth's Penalized Likelihood Compared with Other Approaches 3454
Example 51.13: Complementary Log-Log Model for Infection Rates 3458
Example 51.14: Complementary Log-Log Model for Interval-Censored Survival Times 3462
Example 51.15: Scoring Data Sets with the SCORE Statement 3468
References **3472**

Overview: LOGISTIC Procedure

Binary responses (for example, success and failure), ordinal responses (for example, normal, mild, and severe), and nominal responses (for example, major TV networks viewed at a certain hour) arise in many fields of study. Logistic regression analysis is often used to investigate the relationship between these discrete responses and a set of explanatory variables. Texts that discuss logistic regression include Agresti (2002), Allison (1999), Collett (2003), Cox and Snell (1989), Hosmer and Lemeshow (2000), and Stokes, Davis, and Koch (2000).

For binary response models, the response, Y, of an individual or an experimental unit can take on one of two possible values, denoted for convenience by 1 and 2 (for example, $Y = 1$ if a disease is present, otherwise $Y = 2$). Suppose $\boldsymbol{x}$ is a vector of explanatory variables and $\pi = \Pr(Y = 1 \mid \boldsymbol{x})$ is the response probability to be modeled. The linear logistic model has the form

$$\text{logit}(\pi) \equiv \log\left(\frac{\pi}{1-\pi}\right) = \alpha + \boldsymbol{\beta}'\boldsymbol{x}$$

where α is the intercept parameter and $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_s)'$ is the vector of s slope parameters. Notice that the LOGISTIC procedure, by default, models the probability of the *lower* response levels.

The logistic model shares a common feature with a more general class of linear models: a function $g = g(\mu)$ of the mean of the response variable is assumed to be linearly related to the explanatory variables. Since the mean μ implicitly depends on the stochastic behavior of the response, and the explanatory variables are assumed to be fixed, the function g provides the link between the random (stochastic) component and the systematic (deterministic) component of the response variable Y. For this reason, Nelder and Wedderburn (1972) refer to $g(\mu)$ as a link function. One advantage of the logit function over other link functions is that differences on the logistic scale are interpretable regardless of whether the data are sampled prospectively or retrospectively (McCullagh and Nelder 1989, Chapter 4). Other link functions that are widely used in practice are the probit function and the complementary log-log function. The LOGISTIC procedure enables you to choose one of these link functions, resulting in fitting a broader class of binary response models of the form

$$g(\pi) = \alpha + \boldsymbol{\beta}'\boldsymbol{x}$$

For ordinal response models, the response, Y, of an individual or an experimental unit might be restricted to one of a (usually small) number of ordinal values, denoted for convenience by $1, \ldots, k, k+1$. For example, the severity of coronary disease can be classified into three response categories as 1=no disease, 2=angina pectoris, and 3=myocardial infarction. The LOGISTIC procedure fits a common slopes cumulative model, which is a parallel lines regression model based on the cumulative probabilities of the response categories rather than on their individual probabilities. The cumulative model has the form

$$g(\Pr(Y \le i \mid \boldsymbol{x})) = \alpha_i + \boldsymbol{\beta}'\boldsymbol{x}, \quad i = 1, \ldots, k$$

where $\alpha_1, \ldots, \alpha_k$ are k intercept parameters, and $\boldsymbol{\beta}$ is the vector of slope parameters. This model has been considered by many researchers. Aitchison and Silvey (1957) and Ashford (1959) employ a probit scale and provide a maximum likelihood analysis; Walker and Duncan (1967) and Cox and Snell (1989) discuss the use of the log odds scale. For the log odds scale, the cumulative logit model is often referred to as the *proportional odds* model.

For nominal response logistic models, where the $k+1$ possible responses have no natural ordering, the logit model can also be extended to a *multinomial* model known as a *generalized* or *baseline-category* logit model, which has the form

$$\log\left(\frac{\Pr(Y=i \mid \boldsymbol{x})}{\Pr(Y=k+1 \mid \boldsymbol{x})}\right) = \alpha_i + \boldsymbol{\beta}_i'\boldsymbol{x}, \quad i = 1,\ldots,k$$

where the $\alpha_1,\ldots,\alpha_k$ are k intercept parameters, and the $\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_k$ are k vectors of slope parameters. These models are a special case of the *discrete choice* or *conditional logit* models introduced by McFadden (1974).

The LOGISTIC procedure fits linear logistic regression models for discrete response data by the method of maximum likelihood. It can also perform conditional logistic regression for binary response data and exact conditional logistic regression for binary and nominal response data. The maximum likelihood estimation is carried out with either the Fisher scoring algorithm or the Newton-Raphson algorithm, and you can perform the bias-reducing penalized likelihood optimization as discussed by Firth (1993) and Heinze and Schemper (2002). You can specify starting values for the parameter estimates. The logit link function in the logistic regression models can be replaced by the probit function, the complementary log-log function, or the generalized logit function.

The LOGISTIC procedure enables you to specify categorical variables (also known as *classification* or CLASS variables) or continuous variables as explanatory variables. You can also specify more complex model terms such as interactions and nested terms in the same way as in the GLM procedure. Any term specified in the model is referred to as an *effect*, whether it is a continuous variable, a CLASS variable, an interaction, or a nested term. An effect in the model that is not an interaction or a nested term is referred to as a *main* effect.

The LOGISTIC procedure allows either a full-rank parameterization or a less-than-full-rank parameterization of the CLASS variables. The full-rank parameterization offers eight coding methods: effect, reference, ordinal, polynomial, and orthogonalizations of these. The effect coding is the same method that is used in the CATMOD procedure. The less-than-full-rank parameterization, often called *dummy coding*, is the same coding as that used in the GLM procedure.

The LOGISTIC procedure provides four effect selection methods: forward selection, backward elimination, stepwise selection, and best subset selection. The best subset selection is based on the likelihood score statistic. This method identifies a specified number of best models containing one, two, three effects, and so on, up to a single model containing effects for all the explanatory variables.

The LOGISTIC procedure has some additional options to control how to move effects in and out of a model with the forward selection, backward elimination, or stepwise selection model-building strategies. When there are no interaction terms, a main effect can enter or leave a model in a single step based on the p-value of the score or Wald statistic. When there are interaction terms, the selection process also depends on whether you want to preserve model hierarchy. These additional options enable you to specify whether model hierarchy is to be preserved, how model hierarchy is applied, and whether a single effect or multiple effects can be moved in a single step.

Odds ratio estimates are displayed along with parameter estimates. You can also specify the change in the continuous explanatory main effects for which odds ratio estimates are desired. Confidence intervals for the regression parameters and odds ratios can be computed based either on the profile-likelihood function or on the asymptotic normality of the parameter estimators. You can also pro-

duce odds ratios for effects that are involved in interactions or nestings, and for any type of parameterization of the CLASS variables.

Various methods to correct for overdispersion are provided, including Williams' method for grouped binary response data. The adequacy of the fitted model can be evaluated by various goodness-of-fit tests, including the Hosmer-Lemeshow test for binary response data.

Like many procedures in SAS/STAT software that enable the specification of CLASS variables, the LOGISTIC procedure provides a CONTRAST statement for specifying customized hypothesis tests concerning the model parameters. The CONTRAST statement also provides estimation of individual rows of contrasts, which is particularly useful for obtaining odds ratio estimates for various levels of the CLASS variables.

You can perform a conditional logistic regression on binary response data by specifying the STRATA statement. This enables you to perform matched-set and case-control analyses. The number of events and nonevents can vary across the strata. Many of the features available with the unconditional analysis are also available with a conditional analysis.

The LOGISTIC procedure enables you to perform exact conditional logistic regression by using the method of Hirji, Mehta, and Patel (1987) and Mehta, Patel, and Senchaudhuri (1992) by specifying one or more EXACT statements. You can test individual parameters or conduct a joint test for several parameters. The procedure computes two exact tests: the exact conditional score test and the exact conditional probability test. You can request exact estimation of specific parameters and corresponding odds ratios where appropriate. Point estimates, standard errors, and confidence intervals are provided.

Further features of the LOGISTIC procedure enable you to do the following:

- control the ordering of the response categories
- compute a generalized R^2 measure for the fitted model
- reclassify binary response observations according to their predicted response probabilities
- test linear hypotheses about the regression parameters
- create a data set for producing a receiver operating characteristic curve for each fitted model
- specify contrasts to compare several receiver operating characteristic curves
- create a data set containing the estimated response probabilities, residuals, and influence diagnostics
- score a data set by using a previously fitted model

The LOGISTIC procedure now uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For more information about the plots implemented in PROC LOGISTIC, see the section "ODS Graphics" on page 3379.

The remaining sections of this chapter describe how to use PROC LOGISTIC and discuss the underlying statistical methodology. The section "Getting Started: LOGISTIC Procedure" on page 3260 introduces PROC LOGISTIC with an example for binary response data. The section "Syntax: LOGISTIC Procedure" on page 3266 describes the syntax of the procedure. The section "Details:

LOGISTIC Procedure" on page 3318 summarizes the statistical technique employed by PROC LOGISTIC. The section "Examples: LOGISTIC Procedure" on page 3381 illustrates the use of the LOGISTIC procedure.

For more examples and discussion on the use of PROC LOGISTIC, see Stokes, Davis, and Koch (2000), Allison (1999), and SAS Institute Inc. (1995).

Getting Started: LOGISTIC Procedure

The LOGISTIC procedure is similar in use to the other regression procedures in the SAS System. To demonstrate the similarity, suppose the response variable y is binary or ordinal, and x1 and x2 are two explanatory variables of interest. To fit a logistic regression model, you can specify a MODEL statement similar to that used in the REG procedure. For example:

```
proc logistic;
   model y=x1 x2;
run;
```

The response variable y can be either character or numeric. PROC LOGISTIC enumerates the total number of response categories and orders the response levels according to the response variable option ORDER= in the MODEL statement.

The procedure also allows the input of binary response data that are grouped. In the following statements, n represents the number of trials and r represents the number of events:

```
proc logistic;
   model r/n=x1 x2;
run;
```

The following example illustrates the use of PROC LOGISTIC. The data, taken from Cox and Snell (1989, pp. 10–11), consist of the number, r, of ingots not ready for rolling, out of n tested, for a number of combinations of heating time and soaking time.

```
data ingots;
   input Heat Soak r n @@;
   datalines;
7 1.0 0 10  14 1.0 0 31  27 1.0 1 56  51 1.0 3 13
7 1.7 0 17  14 1.7 0 43  27 1.7 4 44  51 1.7 0  1
7 2.2 0  7  14 2.2 2 33  27 2.2 0 21  51 2.2 0  1
7 2.8 0 12  14 2.8 0 31  27 2.8 1 22  51 4.0 0  1
7 4.0 0  9  14 4.0 0 19  27 4.0 1 16
;
```

The following invocation of PROC LOGISTIC fits the binary logit model to the grouped data. The continous covariates Heat and Soak are specified as predictors, and the bar notation ("|") includes their interaction, Heat*Soak. The ODDSRATIO statement produces odds ratios in the presence of interactions, and the ODS GRAPHICS statements produces a graphical display of the requested odds ratios.

```
ods graphics on;
proc logistic data=ingots;
   model r/n = Heat | Soak;
   oddsratio Heat / at(Soak=1 2 3 4);
run;
ods graphics off;
```

The results of this analysis are shown in the following figures. PROC LOGISTIC first lists background information in Figure 51.1 about the fitting of the model. Included are the name of the input data set, the response variable(s) used, the number of observations used, and the link function used.

Figure 51.1 Binary Logit Model

```
                    The LOGISTIC Procedure

                      Model Information

          Data Set                        WORK.INGOTS
          Response Variable (Events)      r
          Response Variable (Trials)      n
          Model                           binary logit
          Optimization Technique          Fisher's scoring

              Number of Observations Read          19
              Number of Observations Used          19
              Sum of Frequencies Read             387
              Sum of Frequencies Used             387
```

The "Response Profile" table (Figure 51.2) lists the response categories (which are Event and Nonevent when grouped data are input), their ordered values, and their total frequencies for the given data.

Figure 51.2 Response Profile with Events/Trials Syntax

```
                    Response Profile

              Ordered     Binary           Total
                Value     Outcome      Frequency

                    1     Event               12
                    2     Nonevent           375

                  Model Convergence Status

        Convergence criterion (GCONV=1E-8) satisfied.
```

The "Model Fit Statistics" table (Figure 51.3) contains the Akaike information criterion (AIC), the Schwarz criterion (SC), and the negative of twice the log likelihood (–2 Log L) for the intercept-only model and the fitted model. AIC and SC can be used to compare different models, and the ones with smaller values are preferred. Results of the likelihood ratio test and the efficient score test for testing the joint significance of the explanatory variables (Soak, Heat, and their interaction) are included in the "Testing Global Null Hypothesis: BETA=0" table (Figure 51.3); the small p-values reject the hypothesis that all slope parameters are equal to zero.

Figure 51.3 Fit Statistics and Hypothesis Tests

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	108.988	103.222
SC	112.947	119.056
-2 Log L	106.988	95.222

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	11.7663	3	0.0082
Score	16.5417	3	0.0009
Wald	13.4588	3	0.0037

The "Analysis of Maximum Likelihood Estimates" table in Figure 51.4 lists the parameter estimates, their standard errors, and the results of the Wald test for individual parameters. Note that the Heat*Soak parameter is not significantly different from zero (p=0.727), nor is the Soak variable (p=0.6916).

Figure 51.4 Parameter Estimates

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-5.9901	1.6666	12.9182	0.0003
Heat	1	0.0963	0.0471	4.1895	0.0407
Soak	1	0.2996	0.7551	0.1574	0.6916
Heat*Soak	1	-0.00884	0.0253	0.1219	0.7270

The "Association of Predicted Probabilities and Observed Responses" table (Figure 51.5) contains four measures of association for assessing the predictive ability of a model. They are based on the number of pairs of observations with different response values, the number of concordant pairs, and the number of discordant pairs, which are also displayed. Formulas for these statistics are given in the section "Rank Correlation of Observed Responses and Predicted Probabilities" on page 3338.

Figure 51.5 Association Table

Association of Predicted Probabilities and Observed Responses

Percent Concordant	70.9	Somers' D	0.537
Percent Discordant	17.3	Gamma	0.608
Percent Tied	11.8	Tau-a	0.032
Pairs	4500	c	0.768

The ODDSRATIO statement produces the "Wald Confidence Interval for Odds Ratios" table (Figure 51.6), and the ODS GRAPHICS statements display these estimates in Figure 51.7. The differences between the odds ratios are small compared to the variability shown by their confidence intervals, which confirms the previous conclusion that the Heat*Soak parameter is not significantly different from zero.

Figure 51.6 Odds Ratios of Heat at Several Values of Soak

Wald Confidence Interval for Odds Ratios

Label	Estimate	95% Confidence Limits	
Heat at Soak=1	1.091	1.032	1.154
Heat at Soak=2	1.082	1.028	1.139
Heat at Soak=3	1.072	0.986	1.166
Heat at Soak=4	1.063	0.935	1.208

Figure 51.7 Plot of Odds Ratios of Heat at Several Values of Soak



Since the Heat*Soak interaction is nonsignificant, the following statements fit a main-effects model:

```
proc logistic data=ingots;
   model r/n = Heat Soak;
run;
```

The results of this analysis are shown in the following figures. The model information and response profiles are the same as those in Figure 51.1 and Figure 51.2 for the saturated model. The "Model Fit Statistics" table in Figure 51.8 shows that the AIC and SC for the main-effects model are smaller than for the saturated model, indicating that the main-effects model might be the preferred model. As in the preceding model, the "Testing Global Null Hypothesis: BETA=0" table indicates that the parameters are significantly different from zero.

Figure 51.8 Fit Statistics and Hypothesis Tests

The LOGISTIC Procedure

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	108.988	101.346
SC	112.947	113.221
-2 Log L	106.988	95.346

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	11.6428	2	0.0030
Score	15.1091	2	0.0005
Wald	13.0315	2	0.0015

The "Analysis of Maximum Likelihood Estimates" table in Figure 51.9 again shows that the Soak parameter is not significantly different from zero (p=0.8639). The odds ratio for each effect parameter, estimated by exponentiating the corresponding parameter estimate, is shown in the "Odds Ratios Estimates" table (Figure 51.4), along with 95% Wald confidence intervals. The confidence interval for the Soak parameter contains the value 1, which also indicates that this effect is not significant.

Figure 51.9 Parameter Estimates and Odds Ratios

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-5.5592	1.1197	24.6503	<.0001
Heat	1	0.0820	0.0237	11.9454	0.0005
Soak	1	0.0568	0.3312	0.0294	0.8639

Figure 51.9 *continued*

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
Heat	1.085	1.036	1.137
Soak	1.058	0.553	2.026

Association of Predicted Probabilities and Observed Responses

Percent Concordant	64.4	Somers' D	0.460
Percent Discordant	18.4	Gamma	0.555
Percent Tied	17.2	Tau-a	0.028
Pairs	4500	c	0.730

Using these parameter estimates, you can calculate the estimated logit of π as

$$-5.5592 + 0.082 \times \text{Heat} + 0.0568 \times \text{Soak}$$

For example, if Heat=7 and Soak=1, then $\text{logit}(\widehat{\pi}) = -4.9284$. Using this logit estimate, you can calculate $\widehat{\pi}$ as follows:

$$\widehat{\pi} = 1/(1 + \text{e}^{4.9284}) = 0.0072$$

This gives the predicted probability of the event (ingot not ready for rolling) for Heat=7 and Soak=1. Note that PROC LOGISTIC can calculate these statistics for you; use the OUTPUT statement with the PREDICTED= option, or use the SCORE statement.

To illustrate the use of an alternative form of input data, the following program creates the ingots data set with the new variables NotReady and Freq instead of n and r. The variable NotReady represents the response of individual units; it has a value of 1 for units not ready for rolling (event) and a value of 0 for units ready for rolling (nonevent). The variable Freq represents the frequency of occurrence of each combination of Heat, Soak, and NotReady. Note that, compared to the previous data set, NotReady=1 implies Freq=r, and NotReady=0 implies Freq=n–r.

```
data ingots;
   input Heat Soak NotReady Freq @@;
   datalines;
7 1.0 0 10  14 1.0 0 31  14 4.0 0 19  27 2.2 0 21  51 1.0 1  3
7 1.7 0 17  14 1.7 0 43  27 1.0 1  1  27 2.8 1  1  51 1.0 0 10
7 2.2 0  7  14 2.2 1  2  27 1.0 0 55  27 2.8 0 21  51 1.7 0  1
7 2.8 0 12  14 2.2 0 31  27 1.7 1  4  27 4.0 1  1  51 2.2 0  1
7 4.0 0  9  14 2.8 0 31  27 1.7 0 40  27 4.0 0 15  51 4.0 0  1
;
```

The following statements invoke PROC LOGISTIC to fit the main-effects model by using the alternative form of the input data set:

```
proc logistic data=ingots;
   model NotReady(event='1') = Heat Soak;
   freq Freq;
run;
```

Results of this analysis are the same as the preceding single-trial main-effects analysis. The displayed output for the two runs are identical except for the background information of the model fit and the "Response Profile" table shown in Figure 51.10.

Figure 51.10 Response Profile with Single-Trial Syntax

The LOGISTIC Procedure

Response Profile

Ordered Value	NotReady	Total Frequency
1	0	375
2	1	12

Probability modeled is NotReady=1.

By default, Ordered Values are assigned to the sorted response values in ascending order, and PROC LOGISTIC models the probability of the response level that corresponds to the Ordered Value 1. There are several methods to change these defaults; the preceding statements specify the response variable option EVENT= to model the probability of NotReady=1 as displayed in Figure 51.10. See the section "Response Level Ordering" on page 3318 for more details.

Syntax: LOGISTIC Procedure

The following statements are available in PROC LOGISTIC:

PROC LOGISTIC < *options* > ;
BY *variables* ;
CLASS *variable* < *(options)* >< *variable* < *(options)* >. . . >< / *options* > ;
CONTRAST *'label' effect values*< *, effect values,*. . . >< / *options* > ;
EXACT < *'label'* >< **INTERCEPT** >< *effects* >< / *options* > ;
FREQ *variable* ;
< *label:* > **MODEL** *events/trials=*< *effects* >< / *options* > ;
< *label:* > **MODEL** *variable* < *(variable_options)* >=< *effects* >< / *options* > ;
OUTPUT < **OUT=***SAS-data-set* >< *keyword=name* < *keyword=name*. . . > >< / *option* > ;
ROC < *'label'* > < *specification* > < / *options* > ;
ROCCONTRAST < *'label'* >< *contrast* >< / *options* > ;
SCORE < *options* > ;
STRATA *effects* < / *options* > ;
< *label:* > **TEST** *equation1* < *,equation2,*. . . >< / *option* > ;
UNITS *independent1=list1* < *independent2=list2* . . . >< / *option* > ;
WEIGHT *variable* < / *option* > ;

The PROC LOGISTIC and MODEL statements are required. The CLASS statement (if specified) must precede the MODEL statement, and the CONTRAST, EXACT, and ROC statements (if specified) must follow the MODEL statement.

The PROC LOGISTIC, MODEL, and ROCCONTRAST statements can be specified at most once. If a FREQ or WEIGHT statement is specified more than once, the variable specified in the first instance is used. If a BY, OUTPUT, or UNITS statement is specified more than once, the last instance is used.

The rest of this section provides detailed syntax information for each of the preceding statements, beginning with the PROC LOGISTIC statement. The remaining statements are covered in alphabetical order.

PROC LOGISTIC Statement

PROC LOGISTIC < *options* > **;**

The PROC LOGISTIC statement invokes the LOGISTIC procedure and optionally identifies input and output data sets, suppresses the display of results, and controls the ordering of the response levels. Table 51.1 summarizes the available options.

Table 51.1 PROC LOGISTIC Statement Options

Option	Description
Input/Output Data Set Options	
COVOUT	displays estimated covariance matrix in OUTEST= data set
DATA=	names the input SAS data set
INEST=	specifies inital estimates SAS data set
INMODEL=	specifies model information SAS data set
NOCOV	does not save covariance matrix in OUTMODEL= data set
OUTDESIGN=	specifies design matrix output SAS data set
OUTDESIGNONLY	outputs the design matrix only
OUTEST=	specifies parameter estimates output SAS data set
OUTMODEL=	specifies model output data set for scoring
Response and CLASS Variable Options	
DESCENDING	reverses sorting order of response variable
NAMELEN=	specifies maximum length of effect names
ORDER=	specifies sorting order of response variable
TRUNCATE	truncates class level names
Displayed Output Options	
ALPHA=	specifies significance level for confidence intervals
NOPRINT	suppresses all displayed output
PLOTS	specifies options for plots
SIMPLE	displays descriptive statistics
Large Data Set Option	
MULTIPASS	does not copy input SAS data set for internal computations

Table 51.1 *continued*

Option	Description
Control of Other Statement Options	
EXACTONLY	performs exact analysis only
EXACTOPTIONS	specifies global options for EXACT statements
ROCOPTIONS	specifies global options for ROC statements

ALPHA=*number*

specifies the level of significance α for $100(1-\alpha)\%$ confidence intervals. The value *number* must be between 0 and 1; the default value is 0.05, which results in 95% intervals. This value is used as the default confidence level for limits computed by the following options:

Statement	Options
CONTRAST	ESTIMATE=
EXACT	ESTIMATE=
MODEL	CLODDS= CLPARM=
ODDSRATIO	CL=
OUTPUT	LOWER= UPPER=
PROC LOGISTIC	PLOTS=EFFECT(CLBAR CLBAND)
ROCCONTRAST	ESTIMATE=
SCORE	CLM

You can override the default in most of these cases by specifying the ALPHA= option in the separate statements.

COVOUT

adds the estimated covariance matrix to the OUTEST= data set. For the COVOUT option to have an effect, the OUTEST= option must be specified. See the section "OUTEST= Output Data Set" on page 3364 for more information.

DATA=*SAS-data-set*

names the SAS data set containing the data to be analyzed. If you omit the DATA= option, the procedure uses the most recently created SAS data set. The INMODEL= option cannot be specified with this option.

DESCENDING

DESC

reverses the sorting order for the levels of the response variable. If both the DESCENDING and ORDER= options are specified, PROC LOGISTIC orders the levels according to the ORDER= option and then reverses that order. This option has the same effect as the response variable option DESCENDING in the MODEL statement. See the section "Response Level Ordering" on page 3318 for more detail.

EXACTONLY

requests only the exact analyses. The asymptotic analysis that PROC LOGISTIC usually performs is suppressed.

EXACTOPTIONS (*options*)

specifies options that apply to every EXACT statement in the program. The following options are available:

ADDTOBS adds the observed sufficient statistic to the sampled exact distribution if the statistic was not sampled. This option has no effect unless the METHOD=NETWORKMC option is specified and the ESTIMATE option is specified in the EXACT statement. If the observed statistic has not been sampled, then the parameter estimate does not exist; by specifying this option, you can produce (biased) estimates.

BUILDSUBSETS Some exact distributions are created by taking a subset of a previously generated exact distribution. When the METHOD=NETWORKMC option is invoked, this has the effect of using fewer than the desired n samples; see the N= option for more details. The BUILDSUBSETS option suppresses this subsetting behavior and instead builds every distribution for sampling.

EPSILON=*value* controls how the partial sums $\sum_{i=1}^{j} y_i \boldsymbol{x}_i$ are compared. *value* must be between 0 and 1; by default, *value*=1E–8.

MAXTIME=*seconds* specifies the maximum clock time (in seconds) that PROC LOGISTIC can use to calculate the exact distributions. If the limit is exceeded, the procedure halts all computations and prints a note to the LOG. The default maximum clock time is seven days.

METHOD=*keyword* specifies which exact conditional algorithm to use for every EXACT statement specified. You can specify one of the following *keywords*:

DIRECT invokes the multivariate shift algorithm of Hirji, Mehta, and Patel (1987). This method directly builds the exact distribution, but it can require an excessive amount of memory in its intermediate stages. METHOD=DIRECT is invoked by default when you are conditioning out at most the intercept, or when the LINK=GLOGIT option is specified in the MODEL statement.

NETWORK invokes an algorithm described in Mehta, Patel, and Senchaudhuri (1992). This method builds a network for each parameter that you are conditioning out, combines the networks, then uses the multivariate shift algorithm to create the exact distribution. The NETWORK method can be faster and require less memory than the DIRECT method. The NETWORK method is invoked by default for most analyses.

NETWORKMC invokes the hybrid network and Monte Carlo algorithm of Mehta, Patel, and Senchaudhuri (1992). This method creates a network, then samples from that network; this method does not reject any of the samples at the cost of using a large amount of memory to create the network. METHOD=NETWORKMC is most useful for producing parameter estimates for problems that are too large for the DIRECT and NETWORK methods to handle and for which asymptotic methods are invalid—for example, for sparse data on a large grid.

N=*n* specifies the number of Monte Carlo samples to take when the METHOD=NETWORKMC option is specified. By default, $n = 10,000$. If the procedure cannot obtain n samples due to a lack of memory, then a note is printed in the SAS log (the number of valid samples is also reported in the listing) and the analysis continues.

Note that the number of samples used to produce any particular statistic might be smaller than n. For example, let $X1$ and $X2$ be continuous variables, denote their joint distribution by $f(X1, X2)$, and let $f(X1|X2 = x2)$ denote the marginal distribution of $X1$ conditioned on the observed value of $X2$. If you request the JOINT test of $X1$ and $X2$, then n samples are used to generate the estimate $\hat{f}(X1, X2)$ of $f(X1, X2)$, from which the test is computed. However, the parameter estimate for $X1$ is computed from the subset of $\hat{f}(X1, X2)$ having $X2 = x2$, and this subset need not contain n samples. Similarly, the distribution for each level of a classification variable is created by extracting the appropriate subset from the joint distribution for the CLASS variable.

In some cases, the marginal sample size can be too small to admit accurate estimation of a particular statistic; a note is printed in the SAS log when a marginal sample size is less than 100. Increasing n will increase the number of samples used in a marginal distribution; however, if you want to control the sample size exactly, you can either specify the BUILDSUBSETS option or do both of the following:

- Remove the JOINT option from the EXACT statement.
- Create dummy variables in a DATA step to represent the levels of a CLASS variable, and specify them as independent variables in the MODEL statement.

ONDISK uses disk space instead of random access memory to build the exact conditional distribution. Use this option to handle larger problems at the cost of slower processing.

SEED=*seed* specifies the initial seed for the random number generator used to take the Monte Carlo samples when the METHOD=NETWORKMC option is specified. The value of the SEED= option must be an integer. If you do not specify a seed, or if you specify a value less than or equal to zero, then PROC LOGISTIC uses the time of day from the computer's clock to generate an initial seed. The seed is displayed in the "Model Information" table.

STATUSN=*number* prints a status line in the SAS log after every *number* Monte Carlo samples when the METHOD=NETWORKMC option is specified. The number of samples taken and the current exact p-value for testing the significance of the model are displayed. You can use this status line to track the progress of the computation of the exact conditional distributions.

STATUSTIME=*seconds* specifies the time interval (in seconds) for printing a status line in the LOG. You can use this status line to track the progress of the computation of the exact conditional distributions. The time interval you specify is approximate; the actual time interval will vary. By default, no status reports are produced.

INEST=*SAS-data-set*

names the SAS data set that contains initial estimates for all the parameters in the model. If BY-group processing is used, it must be accommodated in setting up the INEST= data set. See the section "INEST= Input Data Set" on page 3365 for more information.

INMODEL=*SAS-data-set*

specifies the name of the SAS data set that contains the model information needed for scoring new data. This INMODEL= data set is the OUTMODEL= data set saved in a previous PROC LOGISTIC call. Note that the OUTMODEL= data set should not be modified before its use as an INMODEL= data set.

The DATA= option in the PROC LOGISTIC statement cannot be specified with this option; instead, specify the data sets to be scored in the SCORE statements. FORMAT statements are not allowed when the INMODEL= data set is specified; variables in the DATA= and PRIOR= data sets in the SCORE statement should be formatted within the data sets.

You can specify the BY statement provided that the INMODEL= data set is created under the same BY-group processing.

The CLASS, EXACT, MODEL, OUTPUT, ROC, ROCCONTRAST, TEST, and UNIT statements are not available with the INMODEL= option.

MULTIPASS

forces the procedure to reread the DATA= data set as needed rather than require its storage in memory or in a temporary file on disk. By default, the data set is cleaned up and stored in memory or in a temporary file. This option can be useful for large data sets. All exact analyses are ignored in the presence of the MULTIPASS option. If a STRATA statement is specified, then the data set must first be grouped or sorted by the strata variables.

NAMELEN=*n*

specifies the maximum length of effect names in tables and output data sets to be n characters, where n is a value between 20 and 200. The default length is 20 characters.

NOCOV

specifies that the covariance matrix not be saved in the OUTMODEL= data set. The covariance matrix is needed for computing the confidence intervals for the posterior probabilities in the OUT= data set in the SCORE statement. Specifying this option will reduce the size of the OUTMODEL= data set.

NOPRINT

suppresses all displayed output. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

RORDER=DATA | FORMATTED | INTERNAL

specifies the sorting order for the levels of the response variable. See the response variable option ORDER= in the MODEL statement for more information. For ordering of CLASS variable levels, see the ORDER= option in the CLASS statement.

OUTDESIGN=*SAS-data-set*

specifies the name of the data set that contains the design matrix for the model. The data set contains the same number of observations as the corresponding DATA= data set and includes the response variable (with the same format as in the DATA= data set), the FREQ variable, the WEIGHT variable, the OFFSET= variable, and the design variables for the covariates,

including the Intercept variable of constant value 1 unless the NOINT option in the MODEL statement is specified.

OUTDESIGNONLY
suppresses the model fitting and creates only the OUTDESIGN= data set. This option is ignored if the OUTDESIGN= option is not specified.

OUTEST=*SAS-data-set*
creates an output SAS data set that contains the final parameter estimates and, optionally, their estimated covariances (see the preceding COVOUT option). The output data set also includes a variable named _LNLIKE_, which contains the log likelihood. See the section "OUTEST= Output Data Set" on page 3364 for more information.

OUTMODEL=*SAS-data-set*
specifies the name of the SAS data set that contains the information about the fitted model. This data set contains sufficient information to score new data without having to refit the model. It is solely used as the input to the INMODEL= option in a subsequent PROC LOGISTIC call. The OUTMODEL= option is not available with the STRATA statement. Information in this data set is stored in a very compact form, so you should not modify it manually.

PLOTS <(*global-plot-options***)>< =***plot-request*< *(options)*>>

PLOTS <(*global-plot-options***)> =(***plot-request*< *(options)*>< . . . *plot-request*< *(options)*>>)
controls the plots produced through ODS Graphics. When you specify only one *plot-request*, you can omit the parentheses from around the plot-request. For example:

```
PLOTS = ALL
PLOTS = (ROC EFFECT INFLUENCE(UNPACK))
PLOTS(ONLY) = EFFECT(CLBAR SHOWOBS)
```

You must enable ODS Graphics before requesting plots. For example:

```
ods graphics on;
proc logistic plots=all;
   model y=x;
run;
ods graphics off;
```

If the PLOTS option is not specified or is specified with no options, then graphics are produced by default in the following situations:

- If the INFLUENCE or IPLOTS option is specified in the MODEL statement, then the line-printer plots are suppressed and the INFLUENCE plots are produced.
- If you specify the OUTROC= option in the MODEL statement, then ROC curves are produced. If you also specify a SELECTION= method, then an overlaid plot of all the ROC curves for each step of the selection process is displayed.
- If the OUTROC= option is specified in a SCORE statement, then the ROC curve for the scored data set is displayed.
- If you specify ROC statements, then an overlaid plot of the ROC curves for the model (or the selected model if a SELECTION= method is specified) and for all the ROC statement models is displayed.

- If you specify the CLODDS= option in the MODEL statement, or specify an ODDSRATIO statement, then a plot of the odds ratios and their confidence limits is displayed.

For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

The following *global-plot-options* are available:

LABEL displays the case number on diagnostic plots, to aid in identifying the outlying observations. This option enhances the plots produced by the DFBETAS, DPC, INFLUENCE, LEVERAGE, and PHAT options.

ONLY suppresses the default plots. Only specifically requested *plot-requests* are displayed.

UNPACKPANELS | UNPACK suppresses paneling. By default, multiple plots can appear in some output *panels*. Specify UNPACKPANEL to display each plot separately.

The following *plot-requests* are available:

ALL produces all appropriate plots. You can specify other options with ALL. For example, to display all plots and unpack the DFBETAS plots you can specify `plots=(all dfbetas(unpack))`.

DFBETAS <(UNPACK)> displays plots of DFBETAS versus the case (observation) number. This displays the statistics generated by the DFBETAS=_ALL_ option in the OUTPUT statement. The UNPACK option displays the plots separately. See Output 51.6.5 for an example of this plot.

DPC <(UNPACK)> displays plots of DIFCHISQ and DIFDEV versus the predicted event probability, and colors the markers according to the value of the confidence interval displacement C. The UNPACK option displays the plots separately. See Output 51.6.8 for an example of this plot.

EFFECT<(*effect-options*)> displays and enhances the effect plots for the model. For more information about effect plots and the available *effect-options*, see the section "EFFECT Plots" on page 3275. See Outputs 51.2.11, 51.3.5, 51.4.8, 51.7.4, and 51.15.4 for examples of this plot.

INFLUENCE<(UNPACK)> displays index plots of RESCHI, RESDEV, leverage, confidence interval displacements C and CBar, DIFCHISQ, and DIFDEV. These plots are produced by default when `ods graphics on` is specified. The UNPACK option displays the plots separately. See Outputs 51.6.3 and 51.6.4 for examples of this plot.

LEVERAGE<(UNPACK)> displays plots of DIFCHISQ, DIFDEV, confidence interval displacement C, and the predicted probability versus the leverage. The UNPACK option displays the plots separately. See Output 51.6.7 for an example of this plot.

NONE suppresses all plots.

ODDSRATIO<(*oddsratio-options*)> displays and enhances the odds ratio plots for the model when the CLODDS= option or ODDSRATIO statements are also specified.

For more information about odds ratio plots and the available *oddsratio-options*, see the section "Odds Ratio Plots" on page 3278. See Outputs 51.7,51.2.9, 51.3.3, and 51.4.5 for examples of this plot.

PHAT< (UNPACK) > displays plots of DIFCHISQ, DIFDEV, confidence interval displacement C, and leverage versus the predicted event probability. The UNPACK option displays the plots separately. See Output 51.6.6 for an example of this plot.

ROC< (ID=*keyword***) >** displays the ROC curve. If you also specify a SELECTION= method, then an overlaid plot of all the ROC curves for each step of the selection process is displayed. If you specify ROC statements, then an overlaid plot of the model (or the selected model if a SELECTION= method is specified) and the ROC statement models will be displayed. If the OUTROC= option is specified in a SCORE statement, then the ROC curve for the scored data set is displayed.

The ID= option labels certain points on the ROC curve. Typically, the labeled points are closest to the upper-left corner of the plot, and points directly below or to the right of a labeled point are suppressed. Specifying ID=PROB | CUTPOINT displays the predicted probability of those points, while ID=CASENUM | OBS displays the observation number. In case of ties, only the last observation number is displayed.

See Output 51.7.3 and Example 51.8 for examples of these ROC plots.

ROCOPTIONS (*options***)**

specifies options that apply to every model specified in a ROC statement. The following *options* are available:

ALPHA=*number* sets the significance level for creating confidence limits of the areas and the pairwise differences. The ALPHA= value specified in the PROC LOGISTIC statement is the default. If neither ALPHA= value is specified, then ALPHA=0.05 by default.

EPS=*value* is an alias for the ROCEPS= option in the MODEL statement. This value is used to determine which predicted probabilities are equal. By default, EPS=1000*MACEPS (about 1E–12) for comparisons; however, EPS=0.0001 for computing *c* from the "Association of Predicted Probabilities and Observed Responses" table when ROC statements are not specified.

ID=*keyword-or-variable* displays labels on certain points on the individual ROC curves. This option is identical to, and overrides, the ID= suboption of the PLOTS=ROC option in the PROC statement. Specifying ID=PROB | CUTPOINT displays the predicted probability of an observation, while ID=CASENUM | OBS displays the observation number. In case of ties, the last observation number is displayed.

NODETAILS suppresses the display of the model fitting information for the models specified in the ROC statements.

OUT=*SAS-data-set-name* is an alias for the OUTROC= option in the MODEL statement.

WEIGHTED uses frequency×weight in the ROC computations (Izrael et al. 2002) instead of just frequency. Typically, weights are considered in the fit of the model only, and hence are accounted for in the parameter estimates. The "Association of Predicted Probabilities and Observed Responses" table uses frequency only, and is suppressed when ROC comparisons are performed.

SIMPLE
displays simple descriptive statistics (mean, standard deviation, minimum and maximum) for each continuous explanatory variable. For each CLASS variable involved in the modeling, the frequency counts of the classification levels are displayed. The SIMPLE option generates a breakdown of the simple descriptive statistics or frequency counts for the entire data set and also for individual response categories.

TRUNCATE
determines class levels by using no more than the first 16 characters of the formatted values of CLASS, response, and strata variables. When formatted values are longer than 16 characters, you can use this option to revert to the levels as determined in releases previous to SAS 9.0. This option invokes the same option in the CLASS statement.

EFFECT Plots

Only one EFFECT plot is produced by default; you must specify other *effect-options* to produce multiple plots. For binary response models, the following plots are produced when an EFFECT option is specified with no *effect-options*:

- If you only have continuous covariates in the model, then a plot of the predicted probability versus the first continuous covariate fixing all other continuous covariates at their means is displayed. See Output 51.7.4 for an example with one continuous covariate.
- If you only have classification covariates in the model, then a plot of the predicted probability versus the first CLASS covariate at each level of the second CLASS covariate, if any, holding all other CLASS covariates at their reference levels is displayed.
- If you have CLASS and continuous covariates, then a plot of the predicted probability versus the first continuous covariate at up to 10 cross-classifications of the CLASS covariate levels, while fixing all other continuous covariates at their means and all other CLASS covariates at their reference levels, is displayed. For example, if your model has four binary covariates, there are 16 cross-classifications of the CLASS covariate levels. The plot displays the 8 cross-classifications of the levels of the first three covariates while the fourth covariate is fixed at its reference level.

For polytomous response models, similar plots are produced by default, except that the response levels are used in place of the CLASS covariate levels. Plots for polytomous response models involving OFFSET= variables with multiple values are not available.

See Outputs 51.2.11, 51.3.5, 51.4.8, 51.7.4, and 51.15.4 for examples of effect plots.

The following *effect-options* specify the type of graphic to produce.

AT(*variable=value-list* | **ALL<** *...variable=value-list* | **ALL>)**
specifies fixed values for a covariate. For continuous covariates, you can specify one or more numbers in the *value-list*. For classification covariates, you can specify one or more formatted levels of the covariate enclosed in single quotes (for example, `A='cat' 'dog'`), or you can specify the keyword ALL to select all levels of the classification variable. You can specify a variable at most once in the AT option. By default, continuous covariates are set to their means when they are not used on an axis, while classification covariates are set

to their reference level when they are not used as an X=, SLICEBY=, or PLOTBY= effect. For example, for a model that includes a classification variable A={cat,dog} and a continuous covariate X, specifying **AT(A='cat' X=7 9)** will set A to cat when A does not appear in the plot. When X does not define an axis it first produces plots setting $X = 7$ and then produces plots setting $X = 9$. Note in this example that specifying **AT(A=ALL)** is the same as specifying the PLOTBY=A option.

FITOBSONLY

computes the predicted values only at the observed data. If the FITOBSONLY option is omitted and the X-axis variable is continuous, the predicted values are computed at a grid of points extending slightly beyond the range of the data (see the EXTEND= option for more information). If the FITOBSONLY option is omitted and the X-axis effect is categorical, the predicted values are computed at all possible categories.

INDIVIDUAL

displays the individual probabilities instead of the cumulative probabilities. This option is available only with cumulative models, and it is not available with the LINK option.

LINK

displays the linear predictors instead of the probabilities on the Y axis. For example, for a binary logistic regression, the Y axis will be displayed on the logit scale. The INDIVIDUAL and POLYBAR options are not available with the LINK option.

PLOTBY=*effect*

displays an effect plot at each unique level of the PLOTBY= effect. You can specify *effect* as one CLASS variable or as an interaction of classification covariates. For polytomous-response models, you can also specify the response variable as the lone SLICEBY= effect. For nonsingular parameterizations, the complete cross-classification of the CLASS variables specified in the effect define the different PLOTBY= levels. When the GLM parameterization is used, the PLOTBY= levels can depend on the model and the data.

SLICEBY=*effect*

displays predicted probabilities at each unique level of the SLICEBY= effect. You can specify *effect* as one CLASS variable or as an interaction of classification covariates. For polytomous-response models, you can also specify the response variable as the lone SLICEBY= effect. For nonsingular parameterizations, the complete cross-classification of the CLASS variables specified in the effect define the different SLICEBY= levels. When the GLM parameterization is used, the SLICEBY= levels can depend on the model and the data.

X=*effect*

X=(*effect...effect***)**

specifies effects to be used on the X axis of the effect plots. You can specify several different X axes: continuous variables must be specified as main effects, while CLASS variables can be crossed. For nonsingular parameterizations, the complete cross-classification of the CLASS variables specified in the effect define the axes. When the GLM parameterization is used, the X= levels can depend on the model and the data. The response variable is not allowed as an *effect*.

NOTE: Any variable not specified in a SLICEBY= or PLOTBY= option is available to be displayed on the X axis. A variable can be specified in at most one of the SLICEBY=, PLOTBY=, and X= options.

The following *effect-options* enhance the graphical output.

ALPHA=*number*

specifies the size of the confidence limits. The ALPHA= value specified in the PROC LOGISTIC statement is the default. If neither ALPHA= value is specified, then ALPHA=0.05 by default.

CLBAND<=YES | NO>

displays confidence limits on the plots. This option is not available with the INDIVIDUAL option. If you have CLASS covariates on the X axis, then error bars are displayed (see the CLBAR option) unless you also specify the CONNECT option.

CLBAR

displays the error bars on the plots when you have CLASS covariates on the X axis; if the X axis is continuous, then this invokes the CLBAND option. For polytomous-response models with CLASS covariates only and with the POLYBAR option specified, the stacked bar charts are replaced by side-by-side bar charts with error bars.

CONNECT<=YES | NO>

JOIN<=YES | NO>

connects the predicted values with a line. This option affects only X axes containing classification variables.

EXTEND=*value*

extends continuous X axes by a factor of $value/2$ in each direction. By default, EXTEND=0.2.

MAXATLEN=*length*

specifies the maximum number of characters used to display the levels of all the fixed variables. If the text is too long, it is truncated and ellipses ("...") are appended. By default, *length* is equal to its maximum allowed value, 256.

POLYBAR

replaces scatter plots of polytomous response models with bar charts. This option has no effect on binary-response models, and it is overridden by the CONNECT option.

SHOWOBS<=YES | NO>

displays observations on the plot. For event/trial notation, the observed proportions are displayed; for single-trial binary-response models, the observed events are displayed at $\hat{p} = 1$ and the observed nonevents are displayed at $\hat{p} = 0$. For polytomous response models the predicted probabilities at the observed values of the covariate are computed and displayed.

YRANGE=(<*min*><,*max*>)

displays the Y axis as [*min*,*max*]. Note that the axis might extend beyond your specified values. By default, the entire Y axis, [0,1], is displayed for the predicted probabilities. This option is useful if your predicted probabilities are all contained in some subset of this range.

Odds Ratio Plots

When either the CLODDS= option or the ODDSRATIO statement is specified, the resulting odds ratios and confidence limits can be displayed in a graphic. If you have many odds ratios, you can produce multiple graphics, or *panels*, by displaying subsets of the odds ratios. Odds ratios with duplicate labels are not displayed. See Outputs 51.2.9 and 51.3.3 for examples of odds ratio plots.

The following *oddsratio-options* modify the default odds ratio plot.

DOTPLOT
displays dotted gridlines on the plot.

GROUP
displays the odds ratios in panels defined by the ODDSRATIO statements. The NPANELPOS= option is ignored when this option is specified.

LOGBASE=2 | E | 10
displays the odds ratio axis on the specified log scale.

NPANELPOS=*n*
breaks the plot into multiple graphics having at most $|n|$ odds ratios per graphic. If n is positive, then the number of odds ratios per graphic is balanced; but if n is negative, then no balancing of the number of odds ratios takes place. By default, $n = 0$ and all odds ratios are displayed in a single plot. For example, suppose you want to display 21 odds ratios. Then specifying `NPANELPOS=20` displays two plots, the first with 11 odds ratios and the second with 10; but specifying `NPANELPOS=-20` displays 20 odds ratios in the first plot and only 1 odds ratio in the second.

ORDER=ASCENDING | DESCENDING
displays the odds ratios in sorted order. By default the odds ratios are displayed in the order in which they appear in the corresponding table.

RANGE=(< *min* **><** *,max* **>) | CLIP**
specifies the range of the displayed odds ratio axis. The RANGE=CLIP option has the same effect as specifying the minimum odds ratio as *min* and the maximum odds ratio as *max*. By default, all odds ratio confidence intervals are displayed.

TYPE=HORIZONTAL | HORIZONTALSTAT | VERTICAL | VERTICALBLOCK
controls the look of the graphic. The default TYPE=HORIZONTAL option places the odds ratio values on the X axis, while the TYPE=HORIZONTALSTAT option also displays the values of the odds ratios and their confidence limits on the right side of the graphic. The TYPE=VERTICAL option places the odds ratio values on the Y axis, while the TYPE=VERTICALBLOCK option (available only with the CLODDS= option) places the odds ratio values on the Y axis and puts boxes around the labels.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC LOGISTIC to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set. If you specify more than one BY statement, the last one specified is used.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify NOTSORTED or DESCENDING option in the BY statement for the LOGISTIC procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

If a SCORE statement is specified, then define the *training data set* to be the DATA= or the INMODEL=data set in the PROC LOGISTIC statement, and define the *scoring data set* to be the DATA= data set and PRIOR= data set in the SCORE statement. The training data set contains all of the BY variables, and the scoring data set must contain either all of them or none of them. If the scoring data set contains all the BY variables, matching is carried out between the training and scoring data sets. If the scoring data set does not contain any of the BY variables, the entire scoring data set is used for every BY group in the training data set and the BY variables are added to the output data sets specified in the SCORE statement.

CAUTION: The order of the levels in the response and classification variables is determined from all the data regardless of BY groups. However, different sets of levels might appear in different BY groups. This might affect the value of the reference level for these variables, and hence your interpretation of the model and the parameters.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variable* < *(options)* >< *variable* < *(options)* >. . . >< / *options* > **;**

The CLASS statement names the classification variables to be used in the analysis. The CLASS statement must precede the MODEL statement. You can specify various *options* for each variable by enclosing them in parentheses after the variable name. You can also specify global *options* for the CLASS statement by placing the *options* after a slash (/). Global *options* are applied to all the variables specified in the CLASS statement. If you specify more than one CLASS statement, the global *options* specified in any one CLASS statement apply to all CLASS statements. However, individual CLASS variable *options* override the global *options*. The following *options* are available:

CPREFIX=*n*

specifies that, at most, the first n characters of a CLASS variable name be used in creating names for the corresponding design variables. The default is $32-\min(32, \max(2, f))$, where f is the formatted length of the CLASS variable.

DESCENDING
DESC

reverses the sorting order of the classification variable. If both the DESCENDING and ORDER= options are specified, PROC LOGISTIC orders the categories according to the ORDER= option and then reverses that order.

LPREFIX=*n*

specifies that, at most, the first n characters of a CLASS variable label be used in creating labels for the corresponding design variables. The default is $256-\min(256, \max(2, f))$, where f is the formatted length of the CLASS variable.

MISSING

treats missing values ('.', '.A',...,'.Z' for numeric variables and blanks for character variables) as valid values for the CLASS variable.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of classification variables. By default, ORDER=FORMATTED. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine dependent. When ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option can be useful when you use the CONTRAST statement.

The following table shows how PROC LOGISTIC interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

For more information about sorting order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

PARAM=*keyword*

specifies the parameterization method for the classification variable or variables. Design matrix columns are created from CLASS variables according to the following coding schemes. You can use one of the following *keywords*; the default is PARAM=EFFECT coding.

EFFECT specifies effect coding.

GLM specifies less-than-full-rank reference cell coding. This option can be used only as a global option.

ORDINAL specifies the cumulative parameterization for an ordinal CLASS variable.

POLYNOMIAL | POLY specifies polynomial coding.

REFERENCE | REF specifies reference cell coding.

ORTHEFFECT orthogonalizes PARAM=EFFECT coding.

ORTHORDINAL orthogonalizes PARAM=ORDINAL coding.

ORTHPOLY orthogonalizes PARAM=POLYNOMIAL coding.

ORTHREF orthogonalizes PARAM=REFERENCE coding.

All parameterizations are full rank, except for the GLM parameterization. The REF= option in the CLASS statement determines the reference level for EFFECT and REFERENCE coding, and for their orthogonal parameterizations.

If PARAM=ORTHPOLY or PARAM=POLY, and the classification variable is numeric, then the ORDER= option in the CLASS statement is ignored, and the internal, unformatted values are used. See the section "CLASS Variable Parameterization" on page 3319 for further details.

Parameter names for a CLASS predictor variable are constructed by concatenating the CLASS variable name with the CLASS levels. However, for the POLYNOMIAL and orthogonal parameterizations, parameter names are formed by concatenating the CLASS variable name and keywords that reflect the parameterization. See the section "CLASS Variable Parameterization" on page 3319 for further details.

REF=*'level' | keyword*

specifies the reference level for PARAM=EFFECT, PARAM=REFERENCE, and their orthogonalizations. For an individual (but not a global) variable REF= *option*, you can specify the *level* of the variable to use as the reference level. Specify the formatted value of the variable if a format is assigned. For a global or individual variable REF= *option*, you can use one of the following *keywords*. The default is REF=LAST.

FIRST designates the first ordered level as reference.

LAST designates the last ordered level as reference.

TRUNCATE

determines class levels by using no more than the first 16 characters of the formatted values of CLASS, response, and strata variables. When formatted values are longer than 16 characters,

you can use this option to revert to the levels as determined in releases previous to SAS 9.0. The TRUNCATE option is available only as a global option. This option invokes the same option in the PROC LOGISTIC statement.

CONTRAST Statement

CONTRAST *'label' row-description<, . . .,row-description><< / options>* **;**

where a *row-description* is defined as follows:

effect values<, . . ., effect values>

The CONTRAST statement provides a mechanism for obtaining customized hypothesis tests. It is similar to the CONTRAST and ESTIMATE statements in other modeling procedures.

The CONTRAST statement enables you to specify a matrix, $\mathbf{L}$, for testing the hypothesis $\mathbf{L}\boldsymbol{\beta} = \mathbf{0}$, where $\boldsymbol{\beta}$ is the vector of intercept and slope parameters. You must be familiar with the details of the model parameterization that PROC LOGISTIC uses (for more information, see the PARAM= option in the section "CLASS Statement" on page 3280). Optionally, the CONTRAST statement enables you to estimate each row, $\boldsymbol{l}_i'\boldsymbol{\beta}$, of $\mathbf{L}\boldsymbol{\beta}$ and test the hypothesis $\boldsymbol{l}_i'\boldsymbol{\beta} = 0$. Computed statistics are based on the asymptotic chi-square distribution of the Wald statistic.

There is no limit to the number of CONTRAST statements that you can specify, but they must appear after the MODEL statement.

The following parameters are specified in the CONTRAST statement:

label identifies the contrast in the displayed output. A label is required for every contrast specified, and it must be enclosed in quotes.

effect identifies an effect that appears in the MODEL statement. The name INTERCEPT can be used as an effect when one or more intercepts are included in the model. You do not need to include all effects that are included in the MODEL statement.

values are constants that are elements of the $\mathbf{L}$ matrix associated with the effect. To correctly specify your contrast, it is crucial to know the ordering of parameters within each effect and the variable levels associated with any parameter. The "Class Level Information" table shows the ordering of levels within variables. The E option, described later in this section, enables you to verify the proper correspondence of *values* to parameters. If too many values are specified for an effect, the extra ones are ignored. If too few values are specified, the remaining ones are set to 0.

Multiple degree-of-freedom hypotheses can be tested by specifying multiple *row-descriptions*; the rows of $\mathbf{L}$ are specified in order and are separated by commas. The degrees of freedom is the number of linearly independent constraints implied by the CONTRAST statement—that is, the rank of $\mathbf{L}$.

More details for specifying contrasts involving effects with full-rank parameterizations are given in the section "Full-Rank Parameterized Effects" on page 3283, while details for less-than-full-rank parameterized effects are given in the section "Less-Than-Full-Rank Parameterized Effects" on page 3284.

You can specify the following options after a slash (/).

ALPHA=*number*

specifies the level of significance α for the $100(1-\alpha)\%$ confidence interval for each contrast when the ESTIMATE option is specified. The value of *number* must be between 0 and 1. By default, *number* is equal to the value of the ALPHA= option in the PROC LOGISTIC statement, or 0.05 if that option is not specified.

E

displays the $\mathbf{L}$ matrix.

ESTIMATE=*keyword*

estimates and tests each individual contrast (that is, each row, $\boldsymbol{l}_i'\boldsymbol{\beta}$, of $\mathbf{L}\boldsymbol{\beta}$), exponentiated contrast ($e^{\boldsymbol{l}_i'\boldsymbol{\beta}}$), or predicted probability for the contrast ($g^{-1}(\boldsymbol{l}_i'\boldsymbol{\beta})$). PROC LOGISTIC displays the point estimate, its standard error, a Wald confidence interval, and a Wald chi-square test. The significance level of the confidence interval is controlled by the ALPHA= option. You can estimate the individual contrast, the exponentiated contrast, or the predicted probability for the contrast by specifying one of the following *keywords*:

PARM estimates the individual contrast.

EXP estimates the exponentiated contrast.

BOTH estimates both the individual contrast and the exponentiated contrast.

PROB estimates the predicted probability of the contrast.

ALL estimates the individual contrast, the exponentiated contrast, and the predicted probability of the contrast.

For details about the computations of the standard errors and confidence limits, see the section "Linear Predictor, Predicted Probability, and Confidence Limits" on page 3339.

SINGULAR=*number*

tunes the estimability check. This option is ignored when a full-rank parameterization is specified. If $\boldsymbol{v}$ is a vector, define ABS($\boldsymbol{v}$) to be the largest absolute value of the elements of $\boldsymbol{v}$. For a row vector $\boldsymbol{l}'$ of the contrast matrix $\mathbf{L}$, define $c = \text{ABS}(\boldsymbol{l})$ if ABS($\boldsymbol{l}$) is greater than 0; otherwise, $c = 1$. If ABS($\boldsymbol{l}'-\boldsymbol{l}'\mathbf{T}$) is greater than $c*number$, then $\boldsymbol{l}$ is declared nonestimable. The $\mathbf{T}$ matrix is the Hermite form matrix $\mathbf{I}_0^-\mathbf{I}_0$, where $\mathbf{I}_0^-$ represents a generalized inverse of the information matrix $\mathbf{I}_0$ of the null model. The value for *number* must be between 0 and 1; the default value is 1E–4.

Full-Rank Parameterized Effects

If an effect involving a CLASS variable with a full-rank parameterization does not appear in the CONTRAST statement, then all of its coefficients in the $\mathbf{L}$ matrix are set to 0.

If you use effect coding by default or by specifying PARAM=EFFECT in the CLASS statement, then all parameters are directly estimable and involve no other parameters. For example, suppose an effect-coded CLASS variable A has four levels. Then there are three parameters $(\beta_1, \beta_2, \beta_3)$ representing the first three levels, and the fourth parameter is represented by

$$-\beta_1 - \beta_2 - \beta_3$$

To test the first versus the fourth level of A, you would test

$$\beta_1 = -\beta_1 - \beta_2 - \beta_3$$

or, equivalently,

$$2\beta_1 + \beta_2 + \beta_3 = 0$$

which, in the form $\mathbf{L}\boldsymbol{\beta} = 0$, is

$$\begin{bmatrix} 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \beta_3 \end{bmatrix} = 0$$

Therefore, you would use the following CONTRAST statement:

```
contrast '1 vs. 4' A 2 1 1;
```

To contrast the third level with the average of the first two levels, you would test

$$\frac{\beta_1 + \beta_2}{2} = \beta_3$$

or, equivalently,

$$\beta_1 + \beta_2 - 2\beta_3 = 0$$

Therefore, you would use the following CONTRAST statement:

```
contrast '1&2 vs. 3' A 1 1 -2;
```

Other CONTRAST statements are constructed similarly. For example:

```
contrast '1 vs. 2    '  A  1 -1  0;
contrast '1&2 vs. 4  '  A  3  3  2;
contrast '1&2 vs. 3&4'  A  2  2  0;
contrast 'Main Effect'  A  1  0  0,
                        A  0  1  0,
                        A  0  0  1;
```

Less-Than-Full-Rank Parameterized Effects

When you use the less-than-full-rank parameterization (by specifying PARAM=GLM in the CLASS statement), each row is checked for estimability; see the section "Estimable Functions" on page 66 for more information. If PROC LOGISTIC finds a contrast to be nonestimable, it displays missing values in corresponding rows in the results. PROC LOGISTIC handles missing level combinations of classification variables in the same manner as PROC GLM: parameters corresponding to missing level combinations are not included in the model. This convention can affect the way in which you specify the **L** matrix in your CONTRAST statement. If the elements of **L** are not specified for an effect that contains a specified effect, then the elements of the specified effect are distributed over the levels of the higher-order effect just as the GLM procedure does for its CONTRAST and ESTIMATE statements. For example, suppose that the model contains effects A and B and their interaction A*B. If you specify a CONTRAST statement involving A alone, the **L** matrix contains nonzero terms for both A and A*B, since A*B contains A. See rule 4 in the section "Construction of Least Squares Means" on page 2528 for more details.

EXACT Statement

EXACT < *'label'* >< **INTERCEPT** >< *effects* >< / *options* > **;**

The EXACT statement performs exact tests of the parameters for the specified *effects* and optionally estimates the parameters and outputs the exact conditional distributions. You can specify the keyword INTERCEPT and any effects in the MODEL statement. Inference on the parameters of the specified effects is performed by conditioning on the sufficient statistics of all the other model parameters (possibly including the intercept).

You can specify several EXACT statements, but they must follow the MODEL statement. Each statement can optionally include an identifying *label*. If several EXACT statements are specified, any statement without a label will be assigned a label of the form "Exact*n*", where "*n*" indicates the *n*th EXACT statement. The label is included in the headers of the displayed exact analysis tables.

If a STRATA statement is also specified, then a stratified exact conditional logistic regression is performed. The model contains a different intercept for each stratum, and these intercepts are conditioned out of the model along with any other nuisance parameters (parameters for effects specified in the MODEL statement that are not in the EXACT statement).

If the LINK=GLOGIT option is specified in the MODEL statement, then the EXACTOPTIONS option METHOD=DIRECT is invoked by default and a generalized logit model is fit. Since each effect specified in the MODEL statement adds k parameters to the model (where $k+1$ is the number of response levels), exact analysis of the generalized logit model by using this method is limited to rather small problems.

See the sections "Exact Conditional Logistic Regression" on page 3359 and "Computational Resources for Exact Conditional Logistic Regression" on page 3369 for more information about exact analyses.

The CONTRAST, ODDSRATIO, OUTPUT, ROC, ROCCONTRAST, SCORE, TEST, and UNITS statements are not available with an exact analysis. Exact analyses are not performed when you specify a WEIGHT statement, a link other than LINK=LOGIT or LINK=GLOGIT, an offset variable, the NOFIT option, or a model-selection method. Exact estimation is not available for ordinal response models.

For classification variables, use of the reference parameterization is recommended.

The following options can be specified in each EXACT statement after a slash (/):

ALPHA=*number*

specifies the level of significance α for $100(1-\alpha)\%$ confidence limits for the parameters or odds ratios. The value of *number* must be between 0 and 1. By default, *number* is equal to the value of the ALPHA= option in the PROC LOGISTIC statement, or 0.05 if that option is not specified.

ESTIMATE < **=***keyword* >

estimates the individual parameters (conditioned on all other parameters) for the effects specified in the EXACT statement. For each parameter, a point estimate, a standard error, a confidence interval, and a p-value for a two-sided test that the parameter is zero are displayed.

Note that the two-sided p-value is twice the one-sided p-value. You can optionally specify one of the following keywords:

PARM specifies that the parameters be estimated. This is the default.

ODDS specifies that the odds ratios be estimated. For classification variables, use of the reference parameterization is required.

BOTH specifies that the parameters and odds ratios be estimated.

JOINT

performs the joint test that all of the parameters are simultaneously equal to zero, performs individual hypothesis tests for the parameter of each continuous variable, and performs joint tests for the parameters of each classification variable. The joint test is indicated in the "Conditional Exact Tests" table by the label "Joint."

JOINTONLY

performs only the joint test of the parameters. The test is indicated in the "Conditional Exact Tests" table by the label "Joint." When this option is specified, individual tests for the parameters of each continuous variable and joint tests for the parameters of the classification variables are not performed.

CLTYPE=EXACT | MIDP

requests either the exact or mid-p confidence intervals for the parameter estimates. By default, the exact intervals are produced. The confidence coefficient can be specified with the ALPHA= option. The mid-p interval can be modified with the MIDPFACTOR= option. See the section "Inference for a Single Parameter" on page 3362 for details.

MIDPFACTOR=δ_1 | (δ_1, δ_2)

sets the tie factors used to produce the mid-p hypothesis statistics and the mid-p confidence intervals. δ_1 modifies both the hypothesis tests and confidence intervals, while δ_2 affects only the hypothesis tests. By default, $\delta_1 = 0.5$ and $\delta_2 = 1.0$. See the sections "Hypothesis Tests" on page 3361 and "Inference for a Single Parameter" on page 3362 for details.

ONESIDED

requests one-sided confidence intervals and p-values for the individual parameter estimates and odds ratios. The one-sided p-value is the smaller of the left- and right-tail probabilities for the observed sufficient statistic of the parameter under the null hypothesis that the parameter is zero. The two-sided p-values (default) are twice the one-sided p-values. See the section "Inference for a Single Parameter" on page 3362 for more details.

OUTDIST=*SAS-data-set*

names the SAS data set containing the exact conditional distributions. This data set contains all of the exact conditional distributions required to process the corresponding EXACT statement. This data set contains the possible sufficient statistics for the parameters of the effects specified in the EXACT statement, the counts, and, when hypothesis tests are performed on the parameters, the probability of occurrence and the score value for each sufficient statistic. When you request an OUTDIST= data set, the observed sufficient statistics are displayed in the "Sufficient Statistics" table. See the section "OUTDIST= Output Data Set" on page 3367 for more information.

EXACT Statement Examples

In the following example, two exact tests are computed: one for x1 and the other for x2. The test for x1 is based on the exact conditional distribution of the sufficient statistic for the x1 parameter given the observed values of the sufficient statistics for the intercept, x2, and x3 parameters; likewise, the test for x2 is conditional on the observed sufficient statistics for the intercept, x1, and x3.

```
proc logistic;
   model y= x1 x2 x3;
   exact x1 x2;
run;
```

PROC LOGISTIC determines, from all the specified EXACT statements, the distinct conditional distributions that need to be evaluated. For example, there is only one exact conditional distribution for the following two EXACT statements:

```
exact 'One' x1 / estimate=parm;
exact 'Two' x1 / estimate=parm onesided;
```

For each EXACT statement, individual tests for the parameters of the specified effects are computed unless the JOINTONLY option is specified. Consider the following EXACT statements:

```
exact 'E12' x1 x2 / estimate;
exact 'E1'  x1    / estimate;
exact 'E2'  x2    / estimate;
exact 'J12' x1 x2 / joint;
```

In the E12 statement, the parameters for x1 and x2 are estimated and tested separately. Specifying the E12 statement is equivalent to specifying both the E1 and E2 statements. In the J12 statement, the joint test for the parameters of x1 and x2 is computed as well as the individual tests for x1 and x2.

FREQ Statement

FREQ *variable* **;**

The FREQ statement identifies a *variable* that contains the frequency of occurrence of each observation. PROC LOGISTIC treats each observation as if it appears n times, where n is the value of the FREQ variable for the observation. If it is not an integer, the frequency value is truncated to an integer. If the frequency value is less than 1 or missing, the observation is not used in the model fitting. When the FREQ statement is not specified, each observation is assigned a frequency of 1. If you specify more than one FREQ statement, then the first statement is used.

If a SCORE statement is specified, then the FREQ variable is used for computing fit statistics and the ROC curve, but they are not required for scoring. If the DATA= data set in the SCORE statement does not contain the FREQ variable, the frequency values are assumed to be 1 and a warning message is issued in the LOG. If you fit a model and perform the scoring in the same run, the same FREQ variable is used for fitting and scoring. If you fit a model in a previous run and input it with the INMODEL= option in the current run, then the FREQ variable can be different from the one used in the previous run. However, if a FREQ variable was not specified in the previous run, you can still specify a FREQ variable in the current run.

MODEL Statement

< *label:* > **MODEL** *events/trials=< effects >< / options >* **;**

< *label:* > **MODEL** *variable< (variable_options) >=< effects >< / options >* **;**

The MODEL statement names the response variable and the explanatory effects, including covariates, main effects, interactions, and nested effects; see the section "Specification of Effects" on page 2488 of Chapter 39, "The GLM Procedure," for more information. If you omit the explanatory effects, the procedure fits an intercept-only model. You must specify exactly one MODEL statement.

Two forms of the MODEL statement can be specified. The first form, referred to as *single-trial* syntax, is applicable to binary, ordinal, and nominal response data. The second form, referred to as *events/trials* syntax, is restricted to the case of binary response data. The *single-trial* syntax is used when each observation in the DATA= data set contains information about only a single trial, such as a single subject in an experiment. When each observation contains information about multiple binary-response trials, such as the counts of the number of subjects observed and the number responding, then *events/trials* syntax can be used.

In the *events/trials* syntax, you specify two variables that contain count data for a binomial experiment. These two variables are separated by a slash. The value of the first variable, *events*, is the number of positive responses (or events). The value of the second variable, *trials*, is the number of trials. The values of both *events* and (*trials–events*) must be nonnegative and the value of *trials* must be positive for the response to be valid.

In the *single-trial* syntax, you specify one variable (on the left side of the equal sign) as the response variable. This variable can be character or numeric. Variable_options specific to the response variable can be specified immediately after the response variable with parentheses around them.

For both forms of the MODEL statement, explanatory *effects* follow the equal sign. Variables can be either continuous or classification variables. Classification variables can be character or numeric, and they must be declared in the CLASS statement. When an effect is a classification variable, the procedure inserts a set of coded columns into the design matrix instead of directly entering a single column containing the values of the variable.

Response Variable Options

DESCENDING | DESC

reverses the order of the response categories. If both the DESCENDING and ORDER= options are specified, PROC LOGISTIC orders the response categories according to the ORDER= option and then reverses that order. See the section "Response Level Ordering" on page 3318 for more detail.

EVENT='*category***' | *keyword***

specifies the event category for the binary response model. PROC LOGISTIC models the probability of the event category. The EVENT= option has no effect when there are more than two response categories. You can specify the value (formatted if a format is applied) of

the event category in quotes, or you can specify one of the following keywords. The default is EVENT=FIRST.

FIRST designates the first ordered category as the event.

LAST designates the last ordered category as the event.

One of the most common sets of response levels is {0,1}, with 1 representing the event for which the probability is to be modeled. Consider the example where Y takes the values 1 and 0 for event and nonevent, respectively, and Exposure is the explanatory variable. To specify the value 1 as the event category, use the following MODEL statement:

```
model Y(event='1') = Exposure;
```

ORDER= DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of the response variable. The following table displays the available ORDER= options:

ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine dependent. When ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values.

For more information about sorting order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

REFERENCE='*category***' | *keyword***

REF='*category***' | *keyword***

specifies the reference category for the generalized logit model and the binary response model. For the generalized logit model, each logit contrasts a nonreference category with the reference category. For the binary response model, specifying one response category as the reference is the same as specifying the other response category as the event category. You can specify the value (formatted if a format is applied) of the reference category in quotes, or you can specify one of the following keywords:

FIRST designates the first ordered category as the reference.

LAST designates the last ordered category as the reference. This is the default.

Model Options

Table 51.2 summarizes the options available in the MODEL statement, which can be specified after a slash (/).

Table 51.2 Model Statement Options

Option	Description
Model Specification Options	
LINK=	specifies link function
NOFIT	suppresses model fitting
NOINT	suppresses intercept
OFFSET=	specifies offset variable
SELECTION=	specifies effect selection method
Effect Selection Options	
BEST=	controls the number of models displayed for SCORE selection
DETAILS	requests detailed results at each step
FAST	uses fast elimination method
HIERARCHY=	specifies whether and how hierarchy is maintained and whether a single effect or multiple effects are allowed to enter or leave the model per step
INCLUDE=	specifies number of effects included in every model
MAXSTEP=	specifies maximum number of steps for STEPWISE selection
SEQUENTIAL	adds or deletes effects in sequential order
SLENTRY=	specifies significance level for entering effects
SLSTAY=	specifies significance level for removing effects
START=	specifies number of variables in first model
STOP=	specifies number of variables in final model
STOPRES	adds or deletes variables by residual chi-square criterion
Model-Fitting Specification Options	
ABSFCONV=	specifies absolute function convergence criterion
FCONV=	specifies relative function convergence criterion
FIRTH	specifies Firth's penalized likelihood method
GCONV=	specifies relative gradient convergence criterion
MAXFUNCTION=	specifies maximum number of function calls for the conditional analysis
MAXITER=	specifies maximum number of iterations
NOCHECK	suppresses checking for infinite parameters
RIDGING=	specifies the technique used to improve the log-likelihood function when its value is worse than that of the previous step
SINGULAR=	specifies tolerance for testing singularity
TECHNIQUE=	specifies iterative algorithm for maximization
XCONV=	specifies relative parameter convergence criterion

Table 51.2 *continued*

Option	Description
Confidence Interval Options	
ALPHA=	specifies α for the $100(1-\alpha)\%$ confidence intervals
CLODDS=	computes confidence intervals for odds ratios
CLPARM=	computes confidence intervals for parameters
PLCONV=	specifies profile-likelihood convergence criterion
Classification Options	
CTABLE	displays classification table
PEVENT=	specifies prior event probabilities
PPROB=	specifies probability cutpoints for classification
Overdispersion and Goodness-of-Fit Test Options	
AGGREGATE=	determines subpopulations for Pearson chi-square and deviance
LACKFIT	requests Hosmer and Lemeshow goodness-of-fit test
SCALE=	specifies method to correct overdispersion
ROC Curve Options	
OUTROC=	names the output ROC data set
ROCEPS=	specifies probability grouping criterion
Regression Diagnostics Options	
INFLUENCE	displays influence statistics
IPLOTS	requests index plots
Display Options	
CORRB	displays correlation matrix
COVB	displays covariance matrix
EXPB	displays exponentiated values of estimates
ITPRINT	displays iteration history
NODUMMYPRINT	suppresses "Class Level Information" table
PARMLABEL	displays parameter labels
RSQUARE	displays generalized R^2
STB	displays standardized estimates
Computational Options	
BINWIDTH=	specifies bin size for estimating association statistics
NOLOGSCALE	performs calculations by using normal scaling

The following list describes these options.

ABSFCONV=*value*

specifies the absolute function convergence criterion. Convergence requires a small change in the log-likelihood function in subsequent iterations,

$$|l_i - l_{i-1}| < value$$

where l_i is the value of the log-likelihood function at iteration i. See the section "Convergence Criteria" on page 3327 for more information.

AGGREGATE< =*(variable-list)*>

specifies the subpopulations on which the Pearson chi-square test statistic and the likelihood ratio chi-square test statistic (deviance) are calculated. Observations with common values in the given list of variables are regarded as coming from the same subpopulation. Variables in the list can be any variables in the input data set. Specifying the AGGREGATE option is equivalent to specifying the AGGREGATE= option with a variable list that includes all explanatory variables in the MODEL statement. The deviance and Pearson goodness-of-fit statistics are calculated only when the SCALE= option is specified. Thus, the AGGREGATE (or AGGREGATE=) option has no effect if the SCALE= option is not specified.

See the section "Rescaling the Covariance Matrix" on page 3342 for more information.

ALPHA=*number*

sets the level of significance α for $100(1-\alpha)\%$ confidence intervals for regression parameters or odds ratios. The value of *number* must be between 0 and 1. By default, *number* is equal to the value of the ALPHA= option in the PROC LOGISTIC statement, or 0.05 if the option is not specified. This option has no effect unless confidence limits for the parameters (CLPARM= option) or odds ratios (CLODDS= option or ODDSRATIO statement) are requested.

BEST=*n*

specifies that n models with the highest score chi-square statistics are to be displayed for each model size. It is used exclusively with the SCORE model selection method. If the BEST= option is omitted and there are no more than 10 explanatory variables, then all possible models are listed for each model size. If the option is omitted and there are more than 10 explanatory variables, then the number of models selected for each model size is, at most, equal to the number of explanatory variables listed in the MODEL statement.

BINWIDTH=*width*

specifies the size of the bins used for estimating the association statistics. See the section "Rank Correlation of Observed Responses and Predicted Probabilities" on page 3338 for details. Valid values are $0 \leq width < 1$ (for polytomous response models, $0 < width < 1$). The default *width* is 0.002. If the *width* does not evenly divide the unit interval, it is reduced to a valid value and a message is displayed in the SAS log. The width is also constrained by the amount of memory available on your machine; if you specify a *width* that is too small, it is adjusted to a value for which memory can be allocated and a note is displayed in the SAS log.

If you have a binary response and specify **`BINWIDTH=0`**, then no binning is performed and the exact values of the statistics are computed; this method is a bit slower and might require more memory than the binning approach.

The BINWIDTH= option is ignored when a ROC statement is specified, when ROC graphics are produced, and for the ROC area computations from a SCORE statement.

CLODDS=PL | WALD | BOTH

produces confidence intervals for odds ratios of main effects not involved in interactions or nestings. Computation of these confidence intervals is based on the profile likelihood (CLODDS=PL) or based on individual Wald tests (CLODDS=WALD). By specifying

CLODDS=BOTH, the procedure computes two sets of confidence intervals for the odds ratios, one based on the profile likelihood and the other based on the Wald tests. The confidence coefficient can be specified with the ALPHA= option. The CLODDS=PL option is not available with the STRATA statement. Classification main effects that use parameterizations other than REF, EFFECT, or GLM are ignored. If you need to compute odds ratios for an effect involved in interactions or nestings, or using some other parameterization, then you should specify an ODDSRATIO statement for that effect.

CLPARM=PL | WALD | BOTH

requests confidence intervals for the parameters. Computation of these confidence intervals is based on the profile likelihood (CLPARM=PL) or individual Wald tests (CLPARM=WALD). If you specify CLPARM=BOTH, the procedure computes two sets of confidence intervals for the parameters, one based on the profile likelihood and the other based on individual Wald tests. The confidence coefficient can be specified with the ALPHA= option. The CLPARM=PL option is not available with the STRATA statement.

See the section "Confidence Intervals for Parameters" on page 3333 for more information.

CORRB

displays the correlation matrix of the parameter estimates.

COVB

displays the covariance matrix of the parameter estimates.

CTABLE

classifies the input binary response observations according to whether the predicted event probabilities are above or below some cutpoint value z in the range $(0, 1)$. An observation is predicted as an event if the predicted event probability exceeds or equals z. You can supply a list of cutpoints other than the default list by specifying the PPROB= option (page 3299). Also, false positive and negative rates can be computed as posterior probabilities by using Bayes' theorem. You can use the PEVENT= option to specify prior probabilities for computing these rates. The CTABLE option is ignored if the data have more than two response levels. The CTABLE option is not available with the STRATA statement.

For more information, see the section "Classification Table" on page 3340.

DETAILS

produces a summary of computational details for each step of the effect selection process. It produces the "Analysis of Effects Eligible for Entry" table before displaying the effect selected for entry for forward or stepwise selection. For each model fitted, it produces the "Type 3 Analysis of Effects" table if the fitted model involves CLASS variables, the "Analysis of Maximum Likelihood Estimates" table, and measures of association between predicted probabilities and observed responses. For the statistics included in these tables, see the section "Displayed Output" on page 3371. The DETAILS option has no effect when SELECTION=NONE.

EXPB

EXPEST

displays the exponentiated values ($e^{\widehat{\beta}_i}$) of the parameter estimates $\widehat{\beta}_i$ in the "Analysis of Maximum Likelihood Estimates" table for the logit model. These exponentiated values are

the estimated odds ratios for parameters corresponding to the continuous explanatory variables, and for CLASS effects that use reference or GLM parameterizations.

FAST

uses a computational algorithm of Lawless and Singhal (1978) to compute a first-order approximation to the remaining slope estimates for each subsequent elimination of a variable from the model. Variables are removed from the model based on these approximate estimates. The FAST option is extremely efficient because the model is not refitted for every variable removed. The FAST option is used when SELECTION=BACKWARD and in the backward elimination steps when SELECTION=STEPWISE. The FAST option is ignored when SELECTION=FORWARD or SELECTION=NONE.

FCONV=*value*

specifies the relative function convergence criterion. Convergence requires a small relative change in the log-likelihood function in subsequent iterations,

$$\frac{|l_i - l_{i-1}|}{|l_{i-1}| + 1\text{E}{-}6} < value$$

where l_i is the value of the log likelihood at iteration i. See the section "Convergence Criteria" on page 3327 for more information.

FIRTH

performs Firth's penalized maximum likelihood estimation to reduce bias in the parameter estimates (Heinze and Schemper 2002; Firth 1993). This method is useful in cases of separability, as often occurs when the event is rare, and is an alternative to performing an exact logistic regression. See the section "Firth's Bias-Reducing Penalized Likelihood" on page 3326 for more information.

NOTE: The intercept-only log likelihood is modified by using the full-model Hessian, computed with the slope parameters equal to zero. When fitting a model and scoring a data set in the same PROC LOGISTIC step, the model is fit using Firth's penalty for parameter estimation purposes, but the penalty is not applied to the scored log likelihood.

GCONV=*value*

specifies the relative gradient convergence criterion. Convergence requires that the normalized prediction function reduction is small,

$$\frac{\boldsymbol{g}_i' \mathbf{I}_i \boldsymbol{g}_i}{|l_i| + 1\text{E}{-}6} < value$$

where l_i is the value of the log-likelihood function, $\boldsymbol{g}_i$ is the gradient vector, and $\mathbf{I}_i$ is the (expected) information matrix, all at iteration i. This is the default convergence criterion, and the default value is 1E–8. See the section "Convergence Criteria" on page 3327 for more information.

HIERARCHY=*keyword*

HIER=*keyword*

specifies whether and how the model hierarchy requirement is applied and whether a single effect or multiple effects are allowed to enter or leave the model in one step. You

can specify that only CLASS effects, or both CLASS and interval effects, be subject to the hierarchy requirement. The HIERARCHY= option is ignored unless you also specify one of the following options: SELECTION=FORWARD, SELECTION=BACKWARD, or SELECTION=STEPWISE.

Model hierarchy refers to the requirement that, for any term to be in the model, all effects contained in the term must be present in the model. For example, in order for the interaction A*B to enter the model, the main effects A and B must be in the model. Likewise, neither effect A nor B can leave the model while the interaction A*B is in the model.

The keywords you can specify in the HIERARCHY= option are as follows:

NONE indicates that the model hierarchy is not maintained. Any single effect can enter or leave the model at any given step of the selection process.

SINGLE indicates that only one effect can enter or leave the model at one time, subject to the model hierarchy requirement. For example, suppose that you specify the main effects A and B and the interaction A*B in the model. In the first step of the selection process, either A or B can enter the model. In the second step, the other main effect can enter the model. The interaction effect can enter the model only when both main effects have already been entered. Also, before A or B can be removed from the model, the A*B interaction must first be removed. All effects (CLASS and interval) are subject to the hierarchy requirement.

SINGLECLASS is the same as HIERARCHY=SINGLE except that only CLASS effects are subject to the hierarchy requirement.

MULTIPLE indicates that more than one effect can enter or leave the model at one time, subject to the model hierarchy requirement. In a forward selection step, a single main effect can enter the model, or an interaction can enter the model together with all the effects that are contained in the interaction. In a backward elimination step, an interaction itself, or the interaction together with all the effects that the interaction contains, can be removed. All effects (CLASS and continuous) are subject to the hierarchy requirement.

MULTIPLECLASS is the same as HIERARCHY=MULTIPLE except that only CLASS effects are subject to the hierarchy requirement.

The default value is HIERARCHY=SINGLE, which means that model hierarchy is to be maintained for all effects (that is, both CLASS and continuous effects) and that only a single effect can enter or leave the model at each step.

INCLUDE=*n*

includes the first *n* effects in the MODEL statement in every model. By default, INCLUDE=0. The INCLUDE= option has no effect when SELECTION=NONE.

Note that the INCLUDE= and START= options perform different tasks: the INCLUDE= option includes the first *n* effects variables in every model, whereas the START= option requires only that the first *n* effects appear in the first model.

INFLUENCE

displays diagnostic measures for identifying influential observations in the case of a binary response model. For each observation, the INFLUENCE option displays the case number

(which is the sequence number of the observation), the values of the explanatory variables included in the final model, and the regression diagnostic measures developed by Pregibon (1981).

For a discussion of these diagnostic measures, see the section "Regression Diagnostics" on page 3349. When a STRATA statement is specified, the diagnostics are computed following Storer and Crowley (1985); see the section "Regression Diagnostic Details" on page 3357 for details.

IPLOTS

produces an index plot for each regression diagnostic statistic. An index plot is a scatter plot with the regression diagnostic statistic represented on the Y axis and the case number on the X axis. See Example 51.6 for an illustration.

ITPRINT

displays the iteration history of the maximum-likelihood model fitting. The ITPRINT option also displays the last evaluation of the gradient vector and the final change in the –2 Log Likelihood.

LACKFIT< (*n*) >

performs the Hosmer and Lemeshow goodness-of-fit test (Hosmer and Lemeshow 2000) for the case of a binary response model. The subjects are divided into approximately 10 groups of roughly the same size based on the percentiles of the estimated probabilities. The discrepancies between the observed and expected number of observations in these groups are summarized by the Pearson chi-square statistic, which is then compared to a chi-square distribution with t degrees of freedom, where t is the number of groups minus n. By default, n=2. A small p-value suggests that the fitted model is not an adequate model. The LACKFIT option is not available with the STRATA statement. See the section "The Hosmer-Lemeshow Goodness-of-Fit Test" on page 3344 for more information.

LINK=*keyword*

L=*keyword*

specifies the link function linking the response probabilities to the linear predictors. You can specify one of the following keywords. The default is LINK=LOGIT.

CLOGLOG is the complementary log-log function. PROC LOGISTIC fits the binary complementary log-log model when there are two response categories and fits the cumulative complementary log-log model when there are more than two response categories. The aliases are CCLOGLOG, CCLL, and CUMCLOGLOG.

GLOGIT is the generalized logit function. PROC LOGISTIC fits the generalized logit model where each nonreference category is contrasted with the reference category. You can use the response variable option REF= to specify the reference category.

LOGIT is the log odds function. PROC LOGISTIC fits the binary logit model when there are two response categories and fits the cumulative logit model when there are more than two response categories. The aliases are CLOGIT and CUMLOGIT.

PROBIT is the inverse standard normal distribution function. PROC LOGISTIC fits the binary probit model when there are two response categories and fits the cumulative

probit model when there are more than two response categories. The aliases are NORMIT, CPROBIT, and CUMPROBIT.

The LINK= option is not available with the STRATA statement.

See the section "Link Functions and the Corresponding Distributions" on page 3322 for more details.

MAXFUNCTION=*number*

specifies the maximum number of function calls to perform when maximizing the conditional likelihood. This option is valid only when a STRATA statement is specified. The default values are as follows:

- 125 when the number of parameters $p < 40$
- 500 when $40 \leq p < 400$
- 1000 when $p \geq 400$

Since the optimization is terminated only after completing a full iteration, the number of function calls that are actually performed can exceed *number*. If convergence is not attained, the displayed output and all output data sets created by the procedure contain results based on the last maximum likelihood iteration.

MAXITER=*number*

specifies the maximum number of iterations to perform. By default, MAXITER=25. If convergence is not attained in *number* iterations, the displayed output and all output data sets created by the procedure contain results that are based on the last maximum likelihood iteration.

MAXSTEP=*n*

specifies the maximum number of times any explanatory variable is added to or removed from the model when SELECTION=STEPWISE. The default number is twice the number of explanatory variables in the MODEL statement. When the MAXSTEP= limit is reached, the stepwise selection process is terminated. All statistics displayed by the procedure (and included in output data sets) are based on the last model fitted. The MAXSTEP= option has no effect when SELECTION=NONE, FORWARD, or BACKWARD.

NOCHECK

disables the checking process to determine whether maximum likelihood estimates of the regression parameters exist. If you are sure that the estimates are finite, this option can reduce the execution time if the estimation takes more than eight iterations. For more information, see the section "Existence of Maximum Likelihood Estimates" on page 3327.

NODUMMYPRINT

NODESIGNPRINT

NODP

suppresses the "Class Level Information" table, which shows how the design matrix columns for the CLASS variables are coded.

NOINT

suppresses the intercept for the binary response model, the first intercept for the ordinal response model (which forces all intercepts to be nonnegative), or all intercepts for the generalized logit model. This can be particularly useful in conditional logistic analysis; see Example 51.11.

NOFIT

performs the global score test without fitting the model. The global score test evaluates the joint significance of the effects in the MODEL statement. No further analyses are performed. If the NOFIT option is specified along with other MODEL statement options, NOFIT takes effect and all other options except FIRTH, LINK=, NOINT, OFFSET=, ROC, and TECHNIQUE= are ignored. The NOFIT option is not available with the STRATA statement.

NOLOGSCALE

specifies that computations for the conditional and exact conditional logistic models should be computed by using normal scaling. Log scaling can handle numerically larger problems than normal scaling; however, computations in the log scale are slower than computations in normal scale.

OFFSET=*name*

names the offset variable. The regression coefficient for this variable will be fixed at 1. For an example that uses this option, see Example 51.13. You can also use the OFFSET= option to restrict parameters to a fixed value. For example, if you want to restrict the variable X1 to 1 and X2 to 2, compute Restrict= $X1 + 2 * X2$ in a DATA step, specify the option **`offset=Restrict`**, and leave X1 and X2 out of the model.

OUTROC=*SAS-data-set*

OUTR=*SAS-data-set*

creates, for binary response models, an output SAS data set that contains the data necessary to produce the receiver operating characteristic (ROC) curve. The OUTROC= option is not available with the STRATA statement. See the section "OUTROC= Output Data Set" on page 3368 for the list of variables in this data set.

PARMLABEL

displays the labels of the parameters in the "Analysis of Maximum Likelihood Estimates" table.

PEVENT=*value*

PEVENT=(*list***)**

specifies one prior probability or a list of prior probabilities for the event of interest. The false positive and false negative rates are then computed as posterior probabilities by Bayes' theorem. The prior probability is also used in computing the rate of correct prediction. For each prior probability in the given list, a classification table of all observations is computed. By default, the prior probability is the total sample proportion of events. The PEVENT= option is useful for stratified samples. It has no effect if the CTABLE option is not specified. For more information, see the section "False Positive and Negative Rates Using Bayes' Theorem" on page 3341. Also see the PPROB= option for information about how the *list* is specified.

PLCL

is the same as specifying CLPARM=PL.

PLCONV=*value*

controls the convergence criterion for confidence intervals based on the profile-likelihood function. The quantity *value* must be a positive number, with a default value of 1E–4. The PLCONV= option has no effect if profile-likelihood confidence intervals (CLPARM=PL) are not requested.

PLRL

is the same as specifying CLODDS=PL.

PPROB=*value*

PPROB=(*list***)**

specifies one critical probability value (or cutpoint) or a list of critical probability values for classifying observations with the CTABLE option. Each *value* must be between 0 and 1. A response that has a cross validated predicted probability greater than or equal to the current PPROB= value is classified as an event response. The PPROB= option is ignored if the CTABLE option is not specified.

A classification table for each of several cutpoints can be requested by specifying a list. For example, the following statement requests a classification of the observations for each of the cutpoints 0.3, 0.5, 0.6, 0.7, and 0.8:

```
pprob= (0.3, 0.5 to 0.8 by 0.1)
```

If the PPROB= option is not specified, the default is to display the classification for a range of probabilities from the smallest estimated probability (rounded down to the nearest 0.02) to the highest estimated probability (rounded up to the nearest 0.02) with 0.02 increments.

RIDGING=ABSOLUTE | RELATIVE | NONE

specifies the technique used to improve the log-likelihood function when its value in the current iteration is less than that in the previous iteration. If you specify the RIDGING=ABSOLUTE option, the diagonal elements of the negative (expected) Hessian are inflated by adding the ridge value. If you specify the RIDGING=RELATIVE option, the diagonal elements are inflated by a factor of 1 plus the ridge value. If you specify the RIDGING=NONE option, the crude line search method of taking half a step is used instead of ridging. By default, RIDGING=RELATIVE.

RISKLIMITS

RL

WALDRL

is the same as specifying CLODDS=WALD.

ROCEPS=*number*

specifies a criterion for the ROC curve used for grouping estimated event probabilities that are close to each other. In each group, the difference between the largest and the smallest estimated event probabilities does not exceed the given value. The value for *number* must be between 0 and 1; the default value is the square root of the machine epsilon, which is about

1E–8 (in prior releases, the default was 1E–4). The smallest estimated probability in each group serves as a cutpoint for predicting an event response. The ROCEPS= option has no effect if the OUTROC= option is not specified.

RSQUARE
RSQ

requests a generalized R^2 measure for the fitted model. For more information, see the section "Generalized Coefficient of Determination" on page 3330.

SCALE=*scale*

enables you to supply the value of the dispersion parameter or to specify the method for estimating the dispersion parameter. It also enables you to display the "Deviance and Pearson Goodness-of-Fit Statistics" table. To correct for overdispersion or underdispersion, the covariance matrix is multiplied by the estimate of the dispersion parameter. Valid values for *scale* are as follows:

D | DEVIANCE specifies that the dispersion parameter be estimated by the deviance divided by its degrees of freedom.

P | PEARSON specifies that the dispersion parameter be estimated by the Pearson chi-square statistic divided by its degrees of freedom.

WILLIAMS <(*constant*)> specifies that Williams' method be used to model overdispersion. This option can be used only with the *events/trials* syntax. An optional *constant* can be specified as the scale parameter; otherwise, a scale parameter is estimated under the full model. A set of weights is created based on this scale parameter estimate. These weights can then be used in fitting subsequent models of fewer terms than the full model. When fitting these submodels, specify the computed scale parameter as *constant*. See Example 51.10 for an illustration.

N | NONE specifies that no correction is needed for the dispersion parameter; that is, the dispersion parameter remains as 1. This specification is used for requesting the deviance and the Pearson chi-square statistic without adjusting for overdispersion.

constant sets the estimate of the dispersion parameter to be the square of the given *constant*. For example, SCALE=2 sets the dispersion parameter to 4. The value *constant* must be a positive number.

You can use the AGGREGATE (or AGGREGATE=) option to define the subpopulations for calculating the Pearson chi-square statistic and the deviance. In the absence of the AGGREGATE (or AGGREGATE=) option, each observation is regarded as coming from a different subpopulation. For the *events/trials* syntax, each observation consists of n Bernoulli trials, where n is the value of the *trials* variable. For *single-trial* syntax, each observation consists of a single response, and for this setting it is not appropriate to carry out the Pearson or deviance goodness-of-fit analysis. Thus, PROC LOGISTIC ignores specifications SCALE=P, SCALE=D, and SCALE=N when *single-trial* syntax is specified without the AGGREGATE (or AGGREGATE=) option.

The "Deviance and Pearson Goodness-of-Fit Statistics" table includes the Pearson chi-square statistic, the deviance, the degrees of freedom, the ratio of each statistic divided by its degrees of freedom, and the corresponding p-value. The SCALE= option is not available with the STRATA statement. For more information, see the section "Overdispersion" on page 3342.

SELECTION=BACKWARD | B
| FORWARD | F
| NONE | N
| STEPWISE | S
| SCORE

specifies the method used to select the variables in the model. BACKWARD requests backward elimination, FORWARD requests forward selection, NONE fits the complete model specified in the MODEL statement, and STEPWISE requests stepwise selection. SCORE requests best subset selection. By default, SELECTION=NONE.

For more information, see the section "Effect-Selection Methods" on page 3328.

SEQUENTIAL
SEQ

forces effects to be added to the model in the order specified in the MODEL statement or eliminated from the model in the reverse order of that specified in the MODEL statement. The model-building process continues until the next effect to be added has an insignificant adjusted chi-square statistic or until the next effect to be deleted has a significant Wald chi-square statistic. The SEQUENTIAL option has no effect when SELECTION=NONE.

SINGULAR=*value*

specifies the tolerance for testing the singularity of the Hessian matrix (Newton-Raphson algorithm) or the expected value of the Hessian matrix (Fisher scoring algorithm). The Hessian matrix is the matrix of second partial derivatives of the log-likelihood function. The test requires that a pivot for sweeping this matrix be at least this number times a norm of the matrix. Values of the SINGULAR= option must be numeric. By default, *value* is the machine epsilon times 1E7, which is approximately 1E–9.

SLENTRY=*value*
SLE=*value*

specifies the significance level of the score chi-square for entering an effect into the model in the FORWARD or STEPWISE method. Values of the SLENTRY= option should be between 0 and 1, inclusive. By default, SLENTRY=0.05. The SLENTRY= option has no effect when SELECTION=NONE, SELECTION=BACKWARD, or SELECTION=SCORE.

SLSTAY=*value*
SLS=*value*

specifies the significance level of the Wald chi-square for an effect to stay in the model in a backward elimination step. Values of the SLSTAY= option should be between 0 and 1, inclusive. By default, SLSTAY=0.05. The SLSTAY= option has no effect when SELECTION=NONE, SELECTION=FORWARD, or SELECTION=SCORE.

START=*n*

begins the FORWARD, BACKWARD, or STEPWISE effect selection process with the first *n* effects listed in the MODEL statement. The value of *n* ranges from 0 to *s*, where *s* is the total number of effects in the MODEL statement. The default value of *n* is *s* for the BACKWARD method and 0 for the FORWARD and STEPWISE methods. Note that START=*n* specifies only that the first *n* effects appear in the first model, while INCLUDE=*n* requires that the first

n effects be included in every model. For the SCORE method, START=*n* specifies that the smallest models contain *n* effects, where *n* ranges from 1 to *s*; the default value is 1. The START= option has no effect when SELECTION=NONE.

STB

displays the standardized estimates for the parameters for the continuous explanatory variables in the "Analysis of Maximum Likelihood Estimates" table. The standardized estimate of β_i is given by $\widehat{\beta}_i/(s/s_i)$, where s_i is the total sample standard deviation for the ith explanatory variable and

$$s = \begin{cases} \pi/\sqrt{3} & \text{Logistic} \\ 1 & \text{Normal} \\ \pi/\sqrt{6} & \text{Extreme-value} \end{cases}$$

For the intercept parameters and parameters associated with a CLASS variable, the standardized estimates are set to missing.

STOP=*n*

specifies the maximum (SELECTION=FORWARD) or minimum (SELECTION=BACKWARD) number of effects to be included in the final model. The effect selection process is stopped when *n* effects are found. The value of *n* ranges from 0 to *s*, where *s* is the total number of effects in the MODEL statement. The default value of *n* is *s* for the FORWARD method and 0 for the BACKWARD method. For the SCORE method, STOP=*n* specifies that the largest models contain *n* effects, where *n* ranges from 1 to *s*; the default value of *n* is *s*. The STOP= option has no effect when SELECTION=NONE or STEPWISE.

STOPRES

SR

specifies that the removal or entry of effects be based on the value of the residual chi-square. If SELECTION=FORWARD, then the STOPRES option adds the effects into the model one at a time until the residual chi-square becomes insignificant (until the *p*-value of the residual chi-square exceeds the SLENTRY= *value*). If SELECTION=BACKWARD, then the STOPRES option removes effects from the model one at a time until the residual chi-square becomes significant (until the *p*-value of the residual chi-square becomes less than the SLSTAY= *value*). The STOPRES option has no effect when SELECTION=NONE or SELECTION=STEPWISE.

TECHNIQUE=FISHER | NEWTON

TECH=FISHER | NEWTON

specifies the optimization technique for estimating the regression parameters. NEWTON (or NR) is the Newton-Raphson algorithm and FISHER (or FS) is the Fisher scoring algorithm. Both techniques yield the same estimates, but the estimated covariance matrices are slightly different except for the case when the LOGIT link is specified for binary response data. The default is TECHNIQUE=FISHER. See the section "Iterative Algorithms for Model Fitting" on page 3324 for details.

WALDCL

CL

is the same as specifying CLPARM=WALD.

XCONV=*value*

specifies the relative parameter convergence criterion. Convergence requires a small relative parameter change in subsequent iterations,

$$\max_j |\delta_j^{(i)}| < value$$

where

$$\delta_j^{(i)} = \begin{cases} \beta_j^{(i)} - \beta_j^{(i-1)} & |\beta_j^{(i-1)}| < 0.01 \\ \frac{\beta_j^{(i)} - \beta_j^{(i-1)}}{\beta_j^{(i-1)}} & \text{otherwise} \end{cases}$$

and $\beta_j^{(i)}$ is the estimate of the jth parameter at iteration i. See the section "Convergence Criteria" on page 3327 for more information.

ODDSRATIO Statement

ODDSRATIO < *'label'* > *variable* < */ options* > **;**

The ODDSRATIO statement produces odds ratios for variable even when variable is involved in interactions with other covariates, and for classification variables that use any parameterization. You can specify several ODDSRATIO statements.

If variable is continuous, then the odds ratios honor any values specified in the UNITS statement. If variable is a classification variable, then odds ratios comparing each pairwise difference between the levels of variable are produced. If variable interacts with a continuous variable, then the odds ratios are produced at the mean of the interacting covariate by default. If variable interacts with a classification variable, then the odds ratios are produced at each level of the interacting covariate by default. The computed odds ratios are independent of the parameterization of any classification variable.

The odds ratios are uniquely labeled by concatenating the following terms to variable:

1. If this is a polytomous response model, then prefix the response variable and the level describing the logit followed by a colon; for example, "Y 0:".
2. If variable is continuous and the UNITS statement provides a value that is not equal to 1, then append "Units=value"; otherwise, if variable is a classification variable, then append the levels being contrasted; for example, "cat vs dog".
3. Append all interacting covariates preceded by "At"; for example, "At X=1.2 A=cat".

If you are also creating odds ratio plots, then this label is displayed on the plots (see the PLOTS option for more information). If you specify a 'label' in the ODDSRATIO statement, then the odds ratios produced by this statement are also labeled: 'label', 'label 2', 'label 3',..., and these are the labels used in the plots. If there are any duplicated labels across all ODDSRATIO statements, then the corresponding odds ratios are not displayed on the plots.

The following *options* are available.

AT(*covariate=value-list* **| REF | ALL< ...***covariate=value-list* **| REF | ALL >)**
specifies fixed levels of the interacting covariates. If a specified *covariate* does not interact with the *variable*, then its AT list is ignored.

For continuous interacting covariates, you can specify one or more numbers in the *value-list*. For classification covariates, you can specify one or more formatted levels of the covariate enclosed in single quotes (for example, **A='cat' 'dog'**), you can specify the keyword REF to select the reference-level, or you can specify the keyword ALL to select all levels of the classification variable. By default, continuous covariates are set to their means, while CLASS covariates are set to ALL. For a model that includes a classification variable A={cat,dog} and a continuous covariate X, specifying **AT(A='cat' X=7 9)** will set A to 'cat', and X to 7 and then 9.

CL=WALD | PL | BOTH
specifies whether to create Wald or profile-likelihood confidence limits, or both. By default, Wald confidence limits are produced.

DIFF=REF | ALL
specifies whether the odds ratios for a classification variable are computed against the reference level, or all pairs of variable are compared. By default, DIFF=ALL. The DIFF= option is ignored when variable is continuous.

PLCONV=*value*
controls the convergence criterion for confidence intervals based on the profile-likelihood function. The quantity *value* must be a positive number, with a default value of 1E–4. The PLCONV= option has no effect if profile-likelihood confidence intervals (CL=PL) are not requested.

PLMAXITER=*n*
specifies the maximum number of iterations to perform. By default, PLMAXITER=25. If convergence is not attained in *n* iterations, the odds ratio or the confidence limits are set to missing. The PLMAXITER= option has no effect if profile-likelihood confidence intervals (CL=PL) are not requested.

PLSINGULAR=*value*
specifies the tolerance for testing the singularity of the Hessian matrix (Newton-Raphson algorithm) or the expected value of the Hessian matrix (Fisher scoring algorithm). The test requires that a pivot for sweeping this matrix be at least this number times a norm of the matrix. Values of the PLSINGULAR= option must be numeric. By default, *value* is the machine epsilon times 1E7, which is approximately 1E–9. The PLSINGULAR= option has no effect if profile-likelihood confidence intervals (CL=PL) are not requested.

OUTPUT Statement

OUTPUT < **OUT=***SAS-data-set*>< *options* > **;**

The OUTPUT statement creates a new SAS data set that contains all the variables in the input data set and, optionally, the estimated linear predictors and their standard error estimates, the estimates of the cumulative or individual response probabilities, and the confidence limits for the cumulative probabilities. Regression diagnostic statistics and estimates of cross validated response probabilities are also available for binary response models. If you specify more than one OUTPUT statement, only the last one is used. Formulas for the statistics are given in the sections "Linear Predictor, Predicted Probability, and Confidence Limits" on page 3339 and "Regression Diagnostics" on page 3349, and, for conditional logistic regression, in the section "Conditional Logistic Regression" on page 3355.

If you use the *single-trial* syntax, the data set also contains a variable named _LEVEL_, which indicates the level of the response that the given row of output is referring to. For instance, the value of the cumulative probability variable is the probability that the response variable is as large as the corresponding value of _LEVEL_. For details, see the section "OUT= Output Data Set in the OUTPUT Statement" on page 3366.

The estimated linear predictor, its standard error estimate, all predicted probabilities, and the confidence limits for the cumulative probabilities are computed for all observations in which the explanatory variables have no missing values, even if the response is missing. By adding observations with missing response values to the input data set, you can compute these statistics for new observations or for settings of the explanatory variables not present in the data without affecting the model fit. Alternatively, the SCORE statement can be used to compute predicted probabilities and confidence intervals for new observations.

Table 51.3 lists the available *options*, which can be specified after a slash (/). The statistic and diagnostic options specify the statistics to be included in the output data set and name the new variables that contain the statistics. If a STRATA statement is specified, only the PREDICTED=, DFBETAS=, and H= options are available; see the section "Regression Diagnostic Details" on page 3357 for details.

Table 51.3 OUTPUT Statement Options

Option	Description
ALPHA=	specifies α for the $100(1-\alpha)\%$ confidence intervals
OUT=	names the output data set
Statistic Options	
LOWER=	names the lower confidence limit
PREDICTED=	names the predicted probabilities
PREDPROBS=	requests the individual, cumulative, or cross validated predicted probabilities
STDXBETA=	names the standard error estimate of the linear predictor
UPPER=	names the upper confidence limit
XBETA=	names the linear predictor

Table 51.3 *continued*

Option	Description
Diagnostic Options for Binary Response	
C=	names the confidence interval displacement
CBAR=	names the confidence interval displacement
DFBETAS=	names the standardized deletion parameter differences
DIFCHISQ=	names the deletion chi-square goodness-of-fit change
DIFDEV=	names the deletion deviance change
H=	names the leverage
RESCHI=	names the Pearson chi-square residual
RESDEV=	names the deviance residual

The following list describes these options.

ALPHA=*number*

sets the level of significance α for $100(1-\alpha)\%$ confidence limits for the appropriate response probabilities. The value of *number* must be between 0 and 1. By default, *number* is equal to the value of the ALPHA= option in the PROC LOGISTIC statement, or 0.05 if that option is not specified.

C=*name*

specifies the confidence interval displacement diagnostic that measures the influence of individual observations on the regression estimates.

CBAR=*name*

specifies the confidence interval displacement diagnostic that measures the overall change in the global regression estimates due to deleting an individual observation.

DFBETAS=_ALL_

DFBETAS=*var-list*

specifies the standardized differences in the regression estimates for assessing the effects of individual observations on the estimated regression parameters in the fitted model. You can specify a list of up to $s+1$ variable names, where s is the number of explanatory variables in the MODEL statement, or you can specify just the keyword _ALL_. In the former specification, the first variable contains the standardized differences in the intercept estimate, the second variable contains the standardized differences in the parameter estimate for the first explanatory variable in the MODEL statement, and so on. In the latter specification, the DFBETAS statistics are named DFBETA_*xxx*, where *xxx* is the name of the regression parameter. For example, if the model contains two variables X1 and X2, the specification DFBETAS=_ALL_ produces three DFBETAS statistics: DFBETA_Intercept, DFBETA_X1, and DFBETA_X2. If an explanatory variable is not included in the final model, the corresponding output variable named in DFBETAS=*var-list* contains missing values.

DIFCHISQ=*name*

specifies the change in the chi-square goodness-of-fit statistic attributable to deleting the individual observation.

DIFDEV=*name*

specifies the change in the deviance attributable to deleting the individual observation.

H=*name*

specifies the diagonal element of the hat matrix for detecting extreme points in the design space.

LOWER=*name*

L=*name*

names the variable containing the lower confidence limits for π, where π is the probability of the event response if *events/trials* syntax or *single-trial* syntax with binary response is specified; for a cumulative model, π is cumulative probability (that is, the probability that the response is less than or equal to the value of _LEVEL_); for the generalized logit model, it is the individual probability (that is, the probability that the response category is represented by the value of _LEVEL_). See the ALPHA= option to set the confidence level.

OUT=*SAS-data-set*

names the output data set. If you omit the OUT= option, the output data set is created and given a default name by using the DATA*n* convention.

PREDICTED=*name*

PRED=*name*

PROB=*name*

P=*name*

names the variable containing the predicted probabilities. For the *events/trials* syntax or *single-trial* syntax with binary response, it is the predicted event probability. For a cumulative model, it is the predicted cumulative probability (that is, the probability that the response variable is less than or equal to the value of _LEVEL_); and for the generalized logit model, it is the predicted individual probability (that is, the probability of the response category represented by the value of _LEVEL_).

PREDPROBS=(*keywords***)**

requests individual, cumulative, or cross validated predicted probabilities. Descriptions of the *keywords* are as follows.

INDIVIDUAL | I requests the predicted probability of each response level. For a response variable Y with three levels, 1, 2, and 3, the individual probabilities are Pr(Y=1), Pr(Y=2), and Pr(Y=3).

CUMULATIVE | C requests the cumulative predicted probability of each response level. For a response variable Y with three levels, 1, 2, and 3, the cumulative probabilities are Pr(Y≤1), Pr(Y≤2), and Pr(Y≤3). The cumulative probability for the last response level always has the constant value of 1. For generalized logit models, the cumulative predicted probabilities are not computed and are set to missing.

CROSSVALIDATE | XVALIDATE | X requests the cross validated individual predicted probability of each response level. These probabilities are derived from the leave-one-out principle—that is, dropping the data of one subject and reestimating the parameter estimates. PROC LOGISTIC uses a less expensive one-step approximation to compute the parameter estimates. This option is valid only for binary

response models; for nominal and ordinal models, the cross validated probabilities are not computed and are set to missing.

See the section "Details of the PREDPROBS= Option" on page 3308 at the end of this section for further details.

RESCHI=*name*
specifies the Pearson (chi-square) residual for identifying observations that are poorly accounted for by the model.

RESDEV=*name*
specifies the deviance residual for identifying poorly fitted observations.

STDXBETA=*name*
names the variable containing the standard error estimates of XBETA. See the section "Linear Predictor, Predicted Probability, and Confidence Limits" on page 3339 for details.

UPPER=*name*

U=*name*
names the variable containing the upper confidence limits for π, where π is the probability of the event response if *events/trials* syntax or *single-trial* syntax with binary response is specified; for a cumulative model, π is cumulative probability (that is, the probability that the response is less than or equal to the value of _LEVEL_); for the generalized logit model, it is the individual probability (that is, the probability that the response category is represented by the value of _LEVEL_). See the ALPHA= option to set the confidence level.

XBETA=*name*
names the variable containing the estimates of the linear predictor $\alpha_i + \boldsymbol{\beta}'\boldsymbol{x}$, where i is the corresponding ordered value of _LEVEL_.

Details of the PREDPROBS= Option

You can request any of the three types of predicted probabilities. For example, you can request both the individual predicted probabilities and the cross validated probabilities by specifying PREDPROBS=(I X).

When you specify the PREDPROBS= option, two automatic variables, _FROM_ and _INTO_, are included for the *single-trial* syntax and only one variable, _INTO_, is included for the *events/trials* syntax. The variable _FROM_ contains the formatted value of the observed response. The variable _INTO_ contains the formatted value of the response level with the largest individual predicted probability.

If you specify PREDPROBS=INDIVIDUAL, the OUT= data set contains k additional variables representing the individual probabilities, one for each response level, where k is the maximum number of response levels across all BY groups. The names of these variables have the form IP_*xxx*, where *xxx* represents the particular level. The representation depends on the following situations:

- If you specify *events/trials* syntax, *xxx* is either 'Event' or 'Nonevent'. Thus, the variable containing the event probabilities is named IP_Event and the variable containing the nonevent probabilities is named IP_Nonevent.
- If you specify the *single-trial* syntax with more than one BY group, *xxx* is 1 for the first ordered level of the response, 2 for the second ordered level of the response, and so forth, as given in the "Response Profile" table. The variable containing the predicted probabilities Pr(Y=1) is named IP_1, where Y is the response variable. Similarly, IP_2 is the name of the variable containing the predicted probabilities Pr(Y=2), and so on.
- If you specify the *single-trial* syntax with no BY-group processing, *xxx* is the left-justified formatted value of the response level (the value might be truncated so that IP_*xxx* does not exceed 32 characters). For example, if Y is the response variable with response levels 'None', 'Mild', and 'Severe', the variables representing individual probabilities Pr(Y='None'), P(Y='Mild'), and P(Y='Severe') are named IP_None, IP_Mild, and IP_Severe, respectively.

If you specify PREDPROBS=CUMULATIVE, the OUT= data set contains k additional variables representing the cumulative probabilities, one for each response level, where k is the maximum number of response levels across all BY groups. The names of these variables have the form CP_*xxx*, where *xxx* represents the particular response level. The naming convention is similar to that given by PREDPROBS=INDIVIDUAL. The PREDPROBS=CUMULATIVE values are the same as those output by the PREDICT= option, but are arranged in variables on each output observation rather than in multiple output observations.

If you specify PREDPROBS=CROSSVALIDATE, the OUT= data set contains k additional variables representing the cross validated predicted probabilities of the k response levels, where k is the maximum number of response levels across all BY groups. The names of these variables have the form XP_*xxx*, where *xxx* represents the particular level. The representation is the same as that given by PREDPROBS=INDIVIDUAL except that for the *events/trials* syntax there are four variables for the cross validated predicted probabilities instead of two:

XP_EVENT_R1E is the cross validated predicted probability of an event when a current event trial is removed.

XP_NONEVENT_R1E is the cross validated predicted probability of a nonevent when a current event trial is removed.

XP_EVENT_R1N is the cross validated predicted probability of an event when a current nonevent trial is removed.

XP_NONEVENT_R1N is the cross validated predicted probability of a nonevent when a current nonevent trial is removed.

The cross validated predicted probabilities are precisely those used in the CTABLE option. See the section "Predicted Probability of an Event for Classification" on page 3341 for details of the computation.

ROC Statement

ROC < *'label'* > < *specification* > < / *options* > **;**

The ROC statements specify models to be used in the ROC comparisons. You can specify more than one ROC statement. ROC statements are identified by their *label*—if you do not specify a *label*, the *i*th ROC statement is labeled "ROC*i*". Additionally, the specified or selected model is labeled with the MODEL statement label or "Model" if the MODEL label is not present. The *specification* can be either a list of effects that have previously been specified in the MODEL statement, or PRED=*variable*, where the *variable* does not have to be specified in the MODEL statement. The PRED= option allows you to input a criterion produced outside PROC LOGISTIC; for example, you can fit a random-intercept model by using PROC GLIMMIX or use survey weights in PROC SURVEYLOGISTIC, then use the predicted values from those models to produce an ROC curve for the comparisons. If you do not make a *specification*, then an intercept-only model is fit to the data, resulting in a noninformative ROC curve that can be used for comparing the area under another ROC curve to 0.5.

You can specify a ROCCONTRAST statement and a ROCOPTIONS option in the PROC LOGISTIC statement to control how the models are compared, while the PLOTS=ROC option controls the ODS Graphics displays. See Example 51.8 for an example that uses the ROC statement.

If you specify any *options*, then a "ROC Model Information" table summarizing the new ROC model is displayed. The *options* are ignored for the PRED= specification. The following *options* are available:

NOOFFSET
does not include an offset variable if the OFFSET= option is specified in the MODEL statement. A constant offset has no effect on the ROC curve, although the cutpoints might be different, but a nonconstant offset can affect the parameter estimates and hence the ROC curve.

LINK=*keyword*
specifies the link function to be used in the model. The available keywords are LOGIT, NORMIT, and CLOGLOG.

ROCCONTRAST Statement

ROCCONTRAST < *'label'* >< *contrast* >< / *options* > **;**

The ROCCONTRAST statement compares the different ROC models. You can specify only one ROCCONTRAST statement. The ROCOPTIONS options in the PROC LOGISTIC statement control how the models are compared. You can specify one of the following *contrast* specifications:

REFERENCE< (MODEL | *'roc-label'*) >
produces a contrast matrix of differences between each ROC curve and a reference curve. The MODEL keyword specifies that the reference curve is that produced from the MODEL statement; the *roc-label* specifies the label of the ROC curve that is to be used as the reference curve. If neither the MODEL keyword nor the *roc-label* label is specified, then the reference ROC curve is either the curve produced from the MODEL statement, the selected model if a selection method is specified, or the model from the first ROC statement if the NOFIT option is specified.

ADJACENTPAIRS
produces a contrast matrix of each ROC curve minus the succeeding curve.

matrix
specifies the contrast in the form `row1,row2,...`, where each *row* contains the coefficients used to compare the ROC curves. Each *row* must contain the same number of entries as there are ROC curves being compared. The elements of each *row* refer to the ROC statements in the order in which they are specified. However, the first element of each *row* refers either to the fitted model, the selected model if a SELECTION= method is specified, or the first specified ROC statement if the NOFIT option is specified.

If no *contrast* is specified, then the REFERENCE contrast with the default reference curve is used. See the section "Comparing ROC Curves" on page 3347 for more information about comparing ROC curves, and see Example 51.8 for an example.

The following *options* are available:

E
displays the contrast.

ESTIMATE < = ROWS | ALLPAIRS >
produces estimates of each row of the contrast when ESTIMATE or ESTIMATE=ROWS is specified. If the ESTIMATE=ALLPAIRS option is specified, then estimates of every pairwise difference of ROC curves are produced.

The row contrasts are labeled "ModelLabel1 – ModelLabel2", where the model labels are as described in the ROC statement; in particular, for the REFERENCE contrast, ModelLabel2 is the reference model label. If you specify your own contrast matrix, then the *i*th contrast row estimate is labeled "Row*i*".

COV
displays covariance matrices used in the computations.

SCORE Statement

SCORE < *options* > **;**

The SCORE statement creates a data set that contains all the data in the DATA= data set together with posterior probabilities and, optionally, prediction confidence intervals. Fit statistics are displayed on request. If you have binary response data, the SCORE statement can be used to create a data set containing data for the ROC curve. You can specify several SCORE statements. FREQ, WEIGHT, and BY statements can be used with the SCORE statements. The SCORE statement is not available with the STRATA statement.

If a SCORE statement is specified in the same run as fitting the model, FORMAT statements should be specified after the SCORE statement in order for the formats to apply to all the DATA= and PRIOR= data sets in the SCORE statement.

See the section "Scoring Data Sets" on page 3352 for more information, and see Example 51.15 for an illustration of how to use this statement.

You can specify the following options:

ALPHA=*number*

specifies the significance level α for $100(1-\alpha)\%$ confidence intervals. By default, the value of *number* is equal to the ALPHA= option in the PROC LOGISTIC statement, or 0.05 if that option is not specified. This option has no effect unless the CLM option in the SCORE statement is requested.

CLM

outputs the Wald-test-based confidence limits for the predicted probabilities. This option is not available when the INMODEL= data set is created with the NOCOV option.

CUMULATIVE

outputs the cumulative predicted probabilities $\Pr(Y \leq i), i = 1, \ldots, k+1$, to the OUT= data set. This option is valid only when you have more than two response levels; otherwise, the option is ignored and a note is printed in the SAS log. These probabilities are named CP_*level_i*, where *level_i* is the ith response level.

If the CLM option is also specified in the SCORE statement, then the Wald-based confidence limits for the cumulative predicted probabilities are also output. The confidence limits are named CLCL_*level_i* and CUCL_*level_i*. In particular, for the lowest response level, the cumulative values (CP, CLCL, CUCL) should be identical to the individual values (P, LCL, UCL), and for the highest response level CP=CLCL=CUCL=1.

DATA=*SAS-data-set*

names the SAS data set that you want to score. If you omit the DATA= option in the SCORE statement, then scoring is performed on the DATA= input data set in the PROC LOGISTIC statement, if specified; otherwise, the DATA=_LAST_ data set is used.

It is not necessary for the DATA= data set in the SCORE statement to contain the response variable unless you are specifying the FITSTAT or OUTROC= option.

Only those variables involved in the fitted model effects are required in the DATA= data set in the SCORE statement. For example, the following statements use forward selection to select effects:

```
proc logistic data=Neuralgia outmodel=sasuser.Model;
   class Treatment Sex;
   model Pain(event='Yes')= Treatment|Sex Age
         / selection=forward sle=.01;
run;
```

Suppose Treatment and Age are the effects selected for the final model. You can score a data set that does not contain the variable Sex since the effect Sex is not in the model that the scoring is based on. For example, the following statements score the Neuralgia data set after dropping the Sex variable:

```
proc logistic inmodel=sasuser.Model;
   score data=Neuralgia(drop=Sex);
run;
```

FITSTAT

displays a table of fit statistics for the data set being scored. Four statistics are computed: total frequency= $\sum_i f_i$, total weight= $\sum_i f_i w_i$, log likelihood= $\sum_i f_i w_i \log(\widehat{\pi}_i)$, and misclassification rate= $\frac{\sum_i (1\{\text{F\_}Y_i \neq \text{I\_}Y_i\} f_i)}{\sum_i f_i}$, where the summations are over all observations in the data set being scored, and the values F_Y and I_Y are described in the section "OUT= Output Data Set in a SCORE Statement" on page 3366.

OUT=*SAS-data-set*

names the SAS data set that contains the predicted information. If you omit the OUT= option, the output data set is created and given a default name by using the DATA*n* convention.

OUTROC=*SAS-data-set*

names the SAS data set that contains the ROC curve for the DATA= data set. The ROC curve is computed only for binary response data. See the section "OUTROC= Output Data Set" on page 3368 for the list of variables in this data set.

PRIOR=*SAS-data-set*

names the SAS data set that contains the priors of the response categories. The priors can be values proportional to the prior probabilities; thus, they do not necessarily sum to one. This data set should include a variable named _PRIOR_ that contains the prior probabilities. For events/trials MODEL syntax, this data set should also include an _OUTCOME_ variable that contains the values EVENT and NONEVENT; for single-trial MODEL syntax, this data set should include the response variable that contains the unformatted response categories. See Example 51.15 for an example.

PRIOREVENT=*value*

specifies the prior event probability for a binary response model. If both PRIOR= and PRIOREVENT= options are specified, the PRIOR= option takes precedence.

ROCEPS=*value*

specifies the criterion for grouping estimated event probabilities that are close to each other for the ROC curve. In each group, the difference between the largest and the smallest estimated event probability does not exceed the given value. The *value* must be between 0 and 1; the

default value is 1E–4. The smallest estimated probability in each group serves as a cutpoint for predicting an event response. The ROCEPS= option has no effect if the OUTROC= option is not specified in the SCORE statement.

STRATA Statement

STRATA *variable* < *(option)*>< *variable* < *(option)*>. . . >< */ options*> **;**

The STRATA statement names the *variables* that define *strata* or *matched sets* to use in a *stratified conditional logistic regression* of binary response data. Observations having the same variable levels are in the same matched set. You can analyze 1: 1, 1: n, m: n and general $m_i : n_i$ matched sets where the number of cases and controls varies across strata. At least one variable must be specified to invoke the stratified analysis, and the usual unconditional asymptotic analysis is not performed. The stratified logistic model has the form

$$\text{logit}(\pi_{hi}) = \alpha_h + \boldsymbol{x}'_{hi}\boldsymbol{\beta}$$

where π_{hi} is the event probability for the ith observation in stratum h having covariates $\boldsymbol{x}_{hi}$, and where the stratum-specific intercepts α_h are the nuisance parameters that are to be conditioned out.

STRATA variables can also be specified in the MODEL statement as classification or continuous covariates; however, the effects are nondegenerate only when crossed with a nonstratification variable. Specifying several STRATA statements is the same as specifying one STRATA statement containing all the strata variables. The STRATA variables can be either character or numeric, and the formatted values of the STRATA variables determine the levels. Thus, you can also use formats to group values into levels; see the discussion of the FORMAT procedure in the *Base SAS Procedures Guide*.

If an EXACT statement is also specified, then a stratified *exact* conditional logistic regression is performed.

The SCORE and WEIGHT statements are not available with a STRATA statement. The following MODEL options are also not supported with a STRATA statement: CLPARM=PL, CLODDS=PL, CTABLE, FIRTH, LACKFIT, LINK=, NOFIT, OUTMODEL=, OUTROC=, ROC, and SCALE=.

The "Strata Summary" table is displayed by default. It displays the number of strata that have a specific number of events and nonevents. For example, if you are analyzing a 1:5 matched study, this table enables you to verify that every stratum in the analysis has exactly one event and five nonevents. Strata containing only events or only nonevents are reported in this table, but such strata are uninformative and are not used in the analysis. Note that you can use the response variable option EVENT= to identify the events; otherwise, the first ordered response category is the event.

The following option can be specified for a stratification variable by enclosing the option in parentheses after the variable name, or it can be specified globally for all STRATA variables after a slash (/).

MISSING
: treats missing values ('.','.A',. . . ,'.Z' for numeric variables and blanks for character variables) as valid STRATA variable values.

The following strata options are also available after the slash.

CHECKDEPENDENCY | CHECK=*keyword*

specifies which variables are to be tested for dependency before the analysis is performed. The available *keywords* are as follows:

NONE performs no dependence checking. Typically, a message about a singular information matrix will be displayed if you do have dependent variables. Dependent variables can be identified after the analysis by noting any missing parameter estimates.

COVARIATES checks dependence between covariates and an added intercept. Dependent covariates are removed from the analysis. However, covariates that are linear functions of the strata variable might not be removed, which will result in a singular information matrix message being displayed in the SAS log. This is the default.

ALL checks dependence between all the strata and covariates. This option can adversely affect performance if you have a large number of strata.

NOSUMMARY

suppresses the display of the "Strata Summary" table.

INFO

displays the "Strata Information" table, which includes the stratum number, levels of the STRATA variables that define the stratum, the number of events, the number of nonevents, and the total frequency for each stratum. Since the number of strata can be very large, this table is displayed only by request.

TEST Statement

< *label:* > **TEST** *equation1* < *, equation2, . . .* >< */ option* > **;**

The TEST statement tests linear hypotheses about the regression coefficients. The Wald test is used to perform a joint test of the null hypotheses H_0: $\mathbf{L}\boldsymbol{\beta} = \boldsymbol{c}$ specified in a single TEST statement, where $\boldsymbol{\beta}$ is the vector of intercept and slope parameters. When $\boldsymbol{c} = \mathbf{0}$ you should specify a CONTRAST statement instead.

Each *equation* specifies a linear hypothesis (a row of the $\mathbf{L}$ matrix and the corresponding element of the $\boldsymbol{c}$ vector). Multiple *equations* are separated by commas. The *label*, which must be a valid SAS name, is used to identify the resulting output and should always be included. You can submit multiple TEST statements.

The form of an *equation* is as follows:

term< ± term . . . > < = ± *term* < ± *term*. . . >>

where *term* is a parameter of the model, or a constant, or a constant times a parameter. Intercept and CLASS variable parameter names should be specified as described in the section "Parameter Names in the OUTEST= Data Set" on page 3364. Note for generalized logit models that this allows you to construct tests of parameters from specific logits. When no equal sign appears, the expression is set to 0. The following statements illustrate possible uses of the TEST statement:

```
proc logistic;
   model y= a1 a2 a3 a4;
   test1: test intercept + .5 * a2 = 0;
   test2: test intercept + .5 * a2;
   test3: test a1=a2=a3;
   test4: test a1=a2, a2=a3;
run;
```

Note that the first and second TEST statements are equivalent, as are the third and fourth TEST statements.

You can specify the following option in the TEST statement after a slash(/).

PRINT

displays intermediate calculations in the testing of the null hypothesis $H_0: \mathbf{L}\boldsymbol{\beta} = \boldsymbol{c}$. This includes $\mathbf{L}\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})\mathbf{L}'$ bordered by $(\mathbf{L}\widehat{\boldsymbol{\beta}}-\boldsymbol{c})$ and $[\mathbf{L}\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})\mathbf{L}']^{-1}$ bordered by $[\mathbf{L}\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})\mathbf{L}']^{-1}(\mathbf{L}\widehat{\boldsymbol{\beta}}-\boldsymbol{c})$, where $\widehat{\boldsymbol{\beta}}$ is the maximum likelihood estimator of $\boldsymbol{\beta}$ and $\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})$ is the estimated covariance matrix of $\widehat{\boldsymbol{\beta}}$.

For more information, see the section "Testing Linear Hypotheses about the Regression Coefficients" on page 3348.

UNITS Statement

UNITS *independent1=list1 < independent2 = list2. . . ></ option>* **;**

The UNITS statement enables you to specify units of change for the continuous explanatory variables so that customized odds ratios can be estimated. If you specify more than one UNITS statement, only the last one is used. An estimate of the corresponding odds ratio is produced for each unit of change specified for an explanatory variable. The UNITS statement is ignored for CLASS variables. Odds ratios are computed only for main effects that are not involved in interactions or nestings, unless an ODDSRATIO statement is also specified. If the CLODDS= option is specified in the MODEL statement, the corresponding confidence limits for the odds ratios are also displayed, as are odds ratios and confidence limits for any CLASS main effects that are not involved in interactions or nestings. The CLASS effects must use the GLM, reference, or effect coding.

The UNITS statement also enables you to customize the odds ratios for effects specified in ODDSRATIO statements, in which case interactions and nestings are allowed, and CLASS variables can be specified with any parameterization.

The term *independent* is the name of an explanatory variable and *list* represents a list of units of change, separated by spaces, that are of interest for that variable. Each unit of change in a list has one of the following forms:

- *number*
- SD or –SD
- *number* * SD

where *number* is any nonzero number, and SD is the sample standard deviation of the corresponding independent variable. For example, $X = -2$ requests an odds ratio that represents the change in the odds when the variable X is decreased by two units. $X = 2*\text{SD}$ requests an estimate of the change in the odds when X is increased by two sample standard deviations.

You can specify the following option in the UNITS statement after a slash(/).

DEFAULT=*list*

gives a list of units of change for all explanatory variables that are not specified in the UNITS statement. Each unit of change can be in any of the forms described previously. If the DEFAULT= option is not specified, PROC LOGISTIC does not produce customized odds ratio estimates for any continuous explanatory variable that is not listed in the UNITS statement.

For more information, see the section "Odds Ratio Estimation" on page 3335.

WEIGHT Statement

WEIGHT *variable* < / *option* > **;**

When a WEIGHT statement appears, each observation in the input data set is weighted by the value of the WEIGHT variable. Unlike a FREQ variable, the values of the WEIGHT variable can be non-integral and are not truncated. Observations with negative, zero, or missing values for the WEIGHT variable are not used in the model fitting. When the WEIGHT statement is not specified, each observation is assigned a weight of 1. The WEIGHT statement is not available with the STRATA statement. If you specify more than one WEIGHT statement, then the first WEIGHT variable is used.

If a SCORE statement is specified, then the WEIGHT variable is used for computing fit statistics and the ROC curve, but it is not required for scoring. If the DATA= data set in the SCORE statement does not contain the WEIGHT variable, the weights are assumed to be 1 and a warning message is issued in the SAS log. If you fit a model and perform the scoring in the same run, the same WEIGHT variable is used for fitting and scoring. If you fit a model in a previous run and input it with the INMODEL= option in the current run, then the WEIGHT variable can be different from the one used in the previous run; however, if a WEIGHT variable was not specified in the previous run, you can still specify a WEIGHT variable in the current run.

CAUTION: PROC LOGISTIC does not compute the proper variance estimators if you are analyzing survey data and specifying the sampling weights through the WEIGHT statement. The SURVEYLOGISTIC procedure is designed to perform the necessary, and correct, computations.

The following option can be added to the WEIGHT statement after a slash (/).

NORMALIZE

NORM

causes the weights specified by the WEIGHT variable to be normalized so that they add up to the actual sample size. Weights w_i are normalized by multiplying them by $\frac{n}{\sum_{i=1}^{n} w_i}$, where n is the sample size. With this option, the estimated covariance matrix of the parameter estimators is invariant to the scale of the WEIGHT variable.

Details: LOGISTIC Procedure

Missing Values

Any observation with missing values for the response, offset, strata, or explanatory variables is excluded from the analysis; however, missing values are valid for variables specified with the MISSING option in the CLASS or STRATA statement. The estimated linear predictor and its standard error estimate, the fitted probabilities and confidence limits, and the regression diagnostic statistics are not computed for any observation with missing offset or explanatory variable values. However, if only the response value is missing, the linear predictor, its standard error, the fitted individual and cumulative probabilities, and confidence limits for the cumulative probabilities can be computed and output to a data set by using the OUTPUT statement.

Response Level Ordering

Response level ordering is important because, by default, PROC LOGISTIC models the probability of response levels with *lower Ordered Value*. Ordered Values are assigned to response levels in ascending sorted order (that is, the lowest response level is assigned Ordered Value 1, the next lowest is assigned Ordered Value 2, and so on) and are displayed in the "Response Profiles" table. If your response variable Y takes values in $\{1,\ldots,k+1\}$, then, by default, the functions modeled with the binary or cumulative model are

$$\text{logit}(\Pr(Y \le i|\boldsymbol{x})), \quad i = 1,\ldots,k$$

and for the generalized logit model the functions modeled are

$$\log\left(\frac{\Pr(Y = i|\boldsymbol{x})}{\Pr(Y = k+1|\boldsymbol{x})}\right), \quad i = 1,\ldots,k$$

where the highest Ordered Value $Y = k+1$ is the reference level. You can change which probabilities are modeled by specifying the EVENT=, REF=, DESCENDING, or ORDER= response variable options in the MODEL statement.

For binary response data with event and nonevent categories, if your event category has a higher Ordered Value, then by default the nonevent is modeled. Since the default response function modeled is

$$\text{logit}(\pi) = \log\left(\frac{\pi}{1-\pi}\right)$$

where π is the probability of the response level assigned Ordered Value 1, and since

$$\text{logit}(\pi) = -\text{logit}(1-\pi)$$

the effect of modeling the nonevent is to change the signs of α and $\boldsymbol{\beta}$ in the model for the event, $\text{logit}(\pi) = \alpha + \boldsymbol{\beta}'\boldsymbol{x}$.

For example, suppose the binary response variable Y takes the values 1 and 0 for event and nonevent, respectively, and Exposure is the explanatory variable. By default, PROC LOGISTIC assigns Ordered Value 1 to response level Y=0, and Ordered Value 2 to response level Y=1. As a result, PROC LOGISTIC models the probability of the nonevent (Ordered Value=1) category, and your parameter estimates have the opposite sign from those in the model for the event. To model the event without using a DATA step to change the values of the variable Y, you can control the ordering of the response levels or select the event or reference level, as shown in the following list:

- Explicitly state which response level is to be modeled by using the response variable option EVENT= in the MODEL statement:

```
model Y(event='1') = Exposure;
```

- Specify the nonevent category for the response variable in the response variable option REF= in the MODEL statement. This option is most useful for generalized logit models where the EVENT= option cannot be used.

```
model Y(ref='0') = Exposure;
```

- Specify the response variable option DESCENDING in the MODEL statement to assign the lowest Ordered Value to Y=1:

```
model Y(descending)=Exposure;
```

- Assign a format to Y such that the first formatted value (when the formatted values are put in sorted order) corresponds to the event. In the following example, Y=1 is assigned the formatted value 'event' and Y=0 is assigned the formatted value 'nonevent'. Since ORDER=FORMATTED by default, Ordered Value 1 is assigned to response level Y=1, so the procedure models the event.

```
proc format;
   value Disease 1='event' 0='nonevent';
run;
proc logistic;
   format Y Disease.;
   model Y=Exposure;
run;
```

CLASS Variable Parameterization

Consider a model with one CLASS variable A with four levels, 1, 2, 5, and 7. This section provides details of the possible choices for the PARAM= option.

CAUTION: PROC LOGISTIC initially parameterizes the CLASS variables by looking at the levels of the variables across the complete data set. If you have an *unbalanced* replication of levels across variables or BY groups, then the design matrix and the parameter interpretation might be different from what you expect. See the EFFECT parameterization in the following list for a specific example.

EFFECT Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three design variables have a value of −1. For instance, if the reference level is 7 (REF='7'), the design matrix columns for A are as follows:

	Design Matrix		
A	**A1**	**A2**	**A5**
1	1	0	0
2	0	1	0
5	0	0	1
7	−1	−1	−1

Parameter estimates of CLASS main effects, using the effect coding scheme, estimate the difference in the effect of each nonreference level compared to the average effect over all 4 levels.

CAUTION: *Unbalanced* replication of levels across variables or BY groups might result in unexpected design matrices and parameter interpretations. For instance, suppose that in addition to the four-level variable A discussed earlier, you have another variable B with two levels, where the fourth level of A occurs only with the first level of B. If your model contains the effect A(B), then the design for A within the second level of B will not be a differential effect. In particular, the design will look like the following:

		Design Matrix					
		A(B=1)			A(B=2)		
B	**A**	**A1**	**A2**	**A5**	**A1**	**A2**	**A5**
1	1	1	0	0	0	0	0
1	2	0	1	0	0	0	0
1	5	0	0	1	0	0	0
1	7	−1	−1	−1	0	0	0
2	1	0	0	0	1	0	0
2	2	0	0	0	0	1	0
2	5	0	0	0	0	0	1

PROC LOGISTIC will then detect linear dependency among the last three design variables and set the parameter for A5(B=2) to zero, resulting in an interpretation of these parameters as if they were reference- or dummy-coded. The REFERENCE or GLM parameterization might be more appropriate for such problems.

GLM As in PROC GLM, four columns are created to indicate group membership. The design matrix columns for A are as follows:

	Design Matrix			
A	**A1**	**A2**	**A5**	**A7**
1	1	0	0	0
2	0	1	0	0
5	0	0	1	0
7	0	0	0	1

Parameter estimates of CLASS main effects, using the GLM coding scheme, estimate the difference in the effects of each level compared to the last level.

ORDINAL Three columns are created to indicate group membership of the higher levels of the effect. For the first level of the effect (which for A is 1), all three design variables have a value of 0. The design matrix columns for A are as follows:

	Design Matrix		
A	**A2**	**A5**	**A7**
1	0	0	0
2	1	0	0
5	1	1	0
7	1	1	1

The first level of the effect is a control or baseline level. Parameter estimates of CLASS main effects, using the ORDINAL coding scheme, estimate the differences between effects of successive levels. When the parameters have the same sign, the effect is monotonic across the levels.

POLYNOMIAL | POLY Three columns are created. The first represents the linear term (x), the second represents the quadratic term (x^2), and the third represents the cubic term (x^3), where x is the level value. If the CLASS levels are not numeric, they are translated into 1, 2, 3, ... according to their sorting order. The design matrix columns for A are as follows:

	Design Matrix		
A	**APOLY1**	**APOLY2**	**APOLY3**
1	1	1	1
2	2	4	8
5	5	25	125
7	7	49	343

REFERENCE | REF Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three design variables have a value of 0. For instance, if the reference level is 7 (REF='7'), the design matrix columns for A are as follows:

	Design Matrix		
A	**A1**	**A2**	**A5**
1	1	0	0
2	0	1	0
5	0	0	1
7	0	0	0

Parameter estimates of CLASS main effects, using the reference coding scheme, estimate the difference in the effect of each nonreference level compared to the effect of the reference level.

ORTHEFFECT The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=EFFECT. The design matrix columns for A are as follows:

	Design Matrix		
A	**AOEFF1**	**AOEFF2**	**AOEFF3**
1	1.41421	−0.81650	−0.57735
2	0.00000	1.63299	−0.57735
5	0.00000	0.00000	1.73205
7	−1.41421	−0.81649	−0.57735

ORTHORDINAL The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=ORDINAL. The design matrix columns for A are as follows:

	Design Matrix		
A	**AOORD1**	**AOORD2**	**AOORD3**
1	−1.73205	0.00000	0.00000
2	0.57735	−1.63299	0.00000
5	0.57735	0.81650	−1.41421
7	0.57735	0.81650	1.41421

ORTHPOLY The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=POLY. The design matrix columns for A are as follows:

	Design Matrix		
A	**AOPOLY1**	**AOPOLY2**	**AOPOLY5**
1	−1.153	0.907	−0.921
2	−0.734	−0.540	1.473
5	0.524	−1.370	−0.921
7	1.363	1.004	0.368

ORTHREF The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=REFERENCE. The design matrix columns for A are as follows:

	Design Matrix		
A	**AOREF1**	**AOREF2**	**AOREF3**
1	1.73205	0.00000	0.00000
2	−0.57735	1.63299	0.00000
5	−0.57735	−0.81650	1.41421
7	−0.57735	−0.81650	−1.41421

Link Functions and the Corresponding Distributions

Four link functions are available in the LOGISTIC procedure. The logit function is the default. To specify a different link function, use the LINK= option in the MODEL statement. The link functions and the corresponding distributions are as follows:

- The logit function

  $$g(p) = \log(p/(1-p))$$

  is the inverse of the cumulative logistic distribution function, which is

  $$F(x) = 1/(1+\exp(-x)) = \exp(x)/(1+\exp(x))$$

- The probit (or normit) function

  $$g(p) = \Phi^{-1}(p)$$

  is the inverse of the cumulative standard normal distribution function, which is

  $$F(x) = \Phi(x) = (2\pi)^{-1/2} \int_{-\infty}^{x} \exp(-z^2/2)dz$$

  Traditionally, the probit function contains the additive constant 5, but throughout PROC LOGISTIC, the terms probit and normit are used interchangeably.

- The complementary log-log function

  $$g(p) = \log(-\log(1-p))$$

  is the inverse of the cumulative extreme-value function (also called the Gompertz distribution), which is

  $$F(x) = 1 - \exp(-\exp(x))$$

- The generalized logit function extends the binary logit link to a vector of levels $(p_1, \ldots, p_{k+1})$ by contrasting each level with a fixed level

  $$g(p_i) = \log(p_i/p_{k+1}) \quad i = 1, \ldots, k$$

The variances of the normal, logistic, and extreme-value distributions are not the same. Their respective means and variances are shown in the following table:

Distribution	**Mean**	**Variance**
Normal	0	1
Logistic	0	$\pi^2/3$
Extreme-value	$-\gamma$	$\pi^2/6$

Here γ is the Euler constant. In comparing parameter estimates from different link functions, you need to take into account the different scalings of the corresponding distributions and, for the complementary log-log function, a possible shift in location. For example, if the fitted probabilities are in the neighborhood of 0.1 to 0.9, then the parameter estimates from the logit link function should be about $\pi/\sqrt{3}$ larger than the estimates from the probit link function.

Determining Observations for Likelihood Contributions

If you use *events/trials* MODEL syntax, each observation is split into two observations. One has response value 1 with a frequency equal to the frequency of the original observation (which is 1 if the FREQ statement is not used) times the value of the *events* variable. The other observation has response value 2 and a frequency equal to the frequency of the original observation times the value of (*trials–events*). These two observations will have the same explanatory variable values and the same FREQ and WEIGHT values as the original observation.

For either *single-trial* or *events/trials* syntax, let j index all observations. In other words, for *single-trial* syntax, j indexes the actual observations. And, for *events/trials* syntax, j indexes the observations after splitting (as described in the preceding paragraph). If your data set has 30 observations and you use *single-trial* syntax, j has values from 1 to 30; if you use *events/trials* syntax, j has values from 1 to 60.

Suppose the response variable in a cumulative response model can take on the ordered values $1, \ldots, k, k+1$, where k is an integer ≥ 1. The likelihood for the jth observation with ordered response value y_j and explanatory variables vector $\boldsymbol{x}_j$ is given by

$$L_j = \begin{cases} F(\alpha_1 + \boldsymbol{\beta}'\boldsymbol{x}_j) & y_j = 1 \\ F(\alpha_i + \boldsymbol{\beta}'\boldsymbol{x}_j) - F(\alpha_{i-1} + \boldsymbol{\beta}'\boldsymbol{x}_j) & 1 < y_j = i \leq k \\ 1 - F(\alpha_k + \boldsymbol{\beta}'\boldsymbol{x}_j) & y_j = k+1 \end{cases}$$

where $F(\cdot)$ is the logistic, normal, or extreme-value distribution function, $\alpha_1, \ldots, \alpha_k$ are ordered intercept parameters, and $\boldsymbol{\beta}$ is the common slope parameter vector.

For the generalized logit model, letting the $k+1$st level be the reference level, the intercepts $\alpha_1, \ldots, \alpha_k$ are unordered and the slope vector $\boldsymbol{\beta}_i$ varies with each logit. The likelihood for the jth observation with response value y_j and explanatory variables vector $\boldsymbol{x}_j$ is given by

$$L_j = \Pr(Y = y_j | \boldsymbol{x}_j) = \begin{cases} \dfrac{e^{\alpha_i + \boldsymbol{x}_j'\boldsymbol{\beta}_i}}{1 + \sum_{m=1}^{k} e^{\alpha_m + \boldsymbol{x}_j'\boldsymbol{\beta}_m}} & 1 \leq y_j = i \leq k \\ \dfrac{1}{1 + \sum_{m=1}^{k} e^{\alpha_m + \boldsymbol{x}_j'\boldsymbol{\beta}_m}} & y_j = k+1 \end{cases}$$

Iterative Algorithms for Model Fitting

Two iterative maximum likelihood algorithms are available in PROC LOGISTIC. The default is the Fisher scoring method, which is equivalent to fitting by iteratively reweighted least squares. The alternative algorithm is the Newton-Raphson method. Both algorithms give the same parameter estimates; however, the estimated covariance matrix of the parameter estimators can differ slightly. This is due to the fact that Fisher scoring is based on the expected information matrix while the Newton-Raphson method is based on the observed information matrix. In the case of a binary logit model, the observed and expected information matrices are identical, resulting in identical estimated covariance matrices for both algorithms. For a generalized logit model, only the Newton-Raphson technique is available. You can use the TECHNIQUE= option to select a fitting algorithm. Also, the

FIRTH option modifies these techniques to perform a bias-reducing penalized maximum likelihood fit.

Iteratively Reweighted Least Squares Algorithm (Fisher Scoring)

Consider the multinomial variable $\mathbf{Z}_j = (Z_{1j}, \ldots, Z_{k+1,j})'$ such that

$$Z_{ij} = \begin{cases} 1 & \text{if } Y_j = i \\ 0 & \text{otherwise} \end{cases}$$

With π_{ij} denoting the probability that the jth observation has response value i, the expected value of $\mathbf{Z}_j$ is $\boldsymbol{\pi}_j = (\pi_{1j}, \ldots, \pi_{k+1,j})'$ where $\pi_{k+1,j} = 1 - \sum_{i=1}^{k} \pi_{ij}$. The covariance matrix of $\mathbf{Z}_j$ is $\mathbf{V}_j$, which is the covariance matrix of a multinomial random variable for one trial with parameter vector $\boldsymbol{\pi}_j$. Let $\boldsymbol{\beta}$ be the vector of regression parameters; in other words, $\boldsymbol{\beta} = (\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_s)'$. Let $\mathbf{D}_j$ be the matrix of partial derivatives of $\boldsymbol{\pi}_j$ with respect to $\boldsymbol{\beta}$. The estimating equation for the regression parameters is

$$\sum_j \mathbf{D}_j' \mathbf{W}_j (\mathbf{Z}_j - \boldsymbol{\pi}_j) = \mathbf{0}$$

where $\mathbf{W}_j = w_j f_j \mathbf{V}_j^-$, w_j and f_j are the weight and frequency of the jth observation, and $\mathbf{V}_j^-$ is a generalized inverse of $\mathbf{V}_j$. PROC LOGISTIC chooses $\mathbf{V}_j^-$ as the inverse of the diagonal matrix with $\boldsymbol{\pi}_j$ as the diagonal.

With a starting value of $\boldsymbol{\beta}^{(0)}$, the maximum likelihood estimate of $\boldsymbol{\beta}$ is obtained iteratively as

$$\boldsymbol{\beta}^{(m+1)} = \boldsymbol{\beta}^{(m)} + \left(\sum_j \mathbf{D}_j' \mathbf{W}_j \mathbf{D}_j\right)^{-1} \sum_j \mathbf{D}_j' \mathbf{W}_j (\mathbf{Z}_j - \boldsymbol{\pi}_j)$$

where $\mathbf{D}_j$, $\mathbf{W}_j$, and $\boldsymbol{\pi}_j$ are evaluated at $\boldsymbol{\beta}^{(m)}$. The expression after the plus sign is the step size. If the likelihood evaluated at $\boldsymbol{\beta}^{(m+1)}$ is less than that evaluated at $\boldsymbol{\beta}^{(m)}$, then $\boldsymbol{\beta}^{(m+1)}$ is recomputed by step-halving or ridging as determined by the value of the RIDGING= option. The iterative scheme continues until convergence is obtained—that is, until $\boldsymbol{\beta}^{(m+1)}$ is sufficiently close to $\boldsymbol{\beta}^{(m)}$. Then the maximum likelihood estimate of $\boldsymbol{\beta}$ is $\widehat{\boldsymbol{\beta}} = \boldsymbol{\beta}^{(m+1)}$.

The covariance matrix of $\widehat{\boldsymbol{\beta}}$ is estimated by

$$\widehat{\mathrm{Cov}}(\widehat{\boldsymbol{\beta}}) = \left(\sum_j \widehat{\mathbf{D}}_j' \widehat{\mathbf{W}}_j \widehat{\mathbf{D}}_j\right)^{-1}$$

where $\widehat{\mathbf{D}}_j$ and $\widehat{\mathbf{W}}_j$ are, respectively, $\mathbf{D}_j$ and $\mathbf{W}_j$ evaluated at $\widehat{\boldsymbol{\beta}}$.

By default, starting values are zero for the slope parameters, and for the intercept parameters, starting values are the observed cumulative logits (that is, logits of the observed cumulative proportions of response). Alternatively, the starting values can be specified with the INEST= option.

Newton-Raphson Algorithm

For cumulative models, let the parameter vector be $\boldsymbol{\beta} = (\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_s)'$, and for the generalized logit model let $\boldsymbol{\beta} = (\alpha_1, \ldots, \alpha_k, \boldsymbol{\beta}_1', \ldots, \boldsymbol{\beta}_k')'$. The gradient vector and the Hessian matrix are given, respectively, by

$$\begin{aligned} \boldsymbol{g} &= \sum_j w_j f_j \frac{\partial l_j}{\partial \boldsymbol{\beta}} \\ \mathbf{H} &= \sum_j -w_j f_j \frac{\partial^2 l_j}{\partial \boldsymbol{\beta}^2} \end{aligned}$$

where $l_j = \log L_j$ is the log likelihood for the jth observation. With a starting value of $\boldsymbol{\beta}^{(0)}$, the maximum likelihood estimate $\widehat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ is obtained iteratively until convergence is obtained:

$$\boldsymbol{\beta}^{(m+1)} = \boldsymbol{\beta}^{(m)} + \mathbf{H}^{-1}\boldsymbol{g}$$

where $\mathbf{H}$ and $\boldsymbol{g}$ are evaluated at $\boldsymbol{\beta}^{(m)}$. If the likelihood evaluated at $\boldsymbol{\beta}^{(m+1)}$ is less than that evaluated at $\boldsymbol{\beta}^{(m)}$, then $\boldsymbol{\beta}^{(m+1)}$ is recomputed by step-halving or ridging.

The covariance matrix of $\widehat{\boldsymbol{\beta}}$ is estimated by

$$\widehat{\mathrm{Cov}}(\widehat{\boldsymbol{\beta}}) = \widehat{\mathbf{H}}^{-1}$$

where $\widehat{\mathbf{H}}$ is $\mathbf{H}$ evaluated at $\widehat{\boldsymbol{\beta}}$.

Firth's Bias-Reducing Penalized Likelihood

Firth's method is currently available only for binary logistic models. It replaces the usual score (gradient) equation

$$g(\beta_j) = \sum_{i=1}^{n}(y_i - \pi_i)x_{ij} = 0 \quad (j = 1, \ldots, p)$$

where p is the number of parameters in the model, with the modified score equation

$$g(\beta_j)^* = \sum_{i=1}^{n}\{y_i - \pi_i + h_i(0.5 - \pi_i)\}x_{ij} = 0 \quad (j = 1, \ldots, p)$$

where the h_is are the ith diagonal elements of the hat matrix $\mathbf{W}^{1/2}\mathbf{X}(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{X}'\mathbf{W}^{1/2}$ and $\mathbf{W} = \mathrm{diag}\{\pi_i(1 - \pi_i)\}$. The Hessian matrix is not modified by this penalty, and the optimization method is performed in the usual manner.

Convergence Criteria

Four convergence criteria are available: ABSFCONV=, FCONV=, GCONV=, and XCONV=. If you specify more than one convergence criterion, the optimization is terminated as soon as one of the criteria is satisfied. If none of the criteria is specified, the default is GCONV=1E–8.

If you specify a STRATA statement, then all unspecified (or nondefault) criteria are also compared to zero. For example, specifying only the criterion XCONV=1E–8 but attaining FCONV=0 terminates the optimization even if the XCONV= criterion is not satisfied, because the log likelihood has reached its maximum.

Existence of Maximum Likelihood Estimates

The likelihood equation for a logistic regression model does not always have a finite solution. Sometimes there is a nonunique maximum on the boundary of the parameter space, at infinity. The existence, finiteness, and uniqueness of maximum likelihood estimates for the logistic regression model depend on the patterns of data points in the observation space (Albert and Anderson 1984; Santner and Duffy 1986). Existence checks are not performed for conditional logistic regression.

Consider a binary response model. Let Y_j be the response of the ith subject, and let $\boldsymbol{x}_j$ be the vector of explanatory variables (including the constant 1 associated with the intercept). There are three mutually exclusive and exhaustive types of data configurations: complete separation, quasi-complete separation, and overlap.

Complete Separation There is a complete separation of data points if there exists a vector $\boldsymbol{b}$ that correctly allocates all observations to their response groups; that is,

$$\begin{cases} \boldsymbol{b}'\boldsymbol{x}_j > 0 & Y_j = 1 \\ \boldsymbol{b}'\boldsymbol{x}_j < 0 & Y_j = 2 \end{cases}$$

This configuration gives nonunique infinite estimates. If the iterative process of maximizing the likelihood function is allowed to continue, the log likelihood diminishes to zero, and the dispersion matrix becomes unbounded.

Quasi-complete Separation The data are not completely separable, but there is a vector $\boldsymbol{b}$ such that

$$\begin{cases} \boldsymbol{b}'\boldsymbol{x}_j \geq 0 & Y_j = 1 \\ \boldsymbol{b}'\boldsymbol{x}_j \leq 0 & Y_j = 2 \end{cases}$$

and equality holds for at least one subject in each response group. This configuration also yields nonunique infinite estimates. If the iterative process of maximizing the likelihood function is allowed to continue, the dispersion matrix becomes unbounded and the log likelihood diminishes to a nonzero constant.

Overlap If neither complete nor quasi-complete separation exists in the sample points, there is an overlap of sample points. In this configuration, the maximum likelihood estimates exist and are unique.

Complete separation and quasi-complete separation are problems typically encountered with small data sets. Although complete separation can occur with any type of data, quasi-complete separation is not likely with truly continuous explanatory variables.

The LOGISTIC procedure uses a simple empirical approach to recognize the data configurations that lead to infinite parameter estimates. The basis of this approach is that any convergence method of maximizing the log likelihood must yield a solution giving complete separation, if such a solution exists. In maximizing the log likelihood, there is no checking for complete or quasi-complete separation if convergence is attained in eight or fewer iterations. Subsequent to the eighth iteration, the probability of the observed response is computed for each observation. If the predicted response equals the observed response for every observation, there is a complete separation of data points and the iteration process is stopped. If the complete separation of data has not been determined and an observation is identified to have an extremely large probability ($\geq$0.95) of predicting the observed response, there are two possible situations. First, there is overlap in the data set, and the observation is an atypical observation of its own group. The iterative process, if allowed to continue, will stop when a maximum is reached. Second, there is quasi-complete separation in the data set, and the asymptotic dispersion matrix is unbounded. If any of the diagonal elements of the dispersion matrix for the standardized observations vectors (all explanatory variables standardized to zero mean and unit variance) exceeds 5000, quasi-complete separation is declared and the iterative process is stopped. If either complete separation or quasi-complete separation is detected, a warning message is displayed in the procedure output.

Checking for quasi-complete separation is less foolproof than checking for complete separation. The NOCHECK option in the MODEL statement turns off the process of checking for infinite parameter estimates. In cases of complete or quasi-complete separation, turning off the checking process typically results in the procedure failing to converge. The presence of a WEIGHT statement also turns off the checking process.

To address the separation issue, you can change your model, specify the FIRTH option to use Firth's penalized likelihood method, or for small data sets specify an EXACT statement to perform an exact conditional logistic regression.

Effect-Selection Methods

Five effect-selection methods are available by specifying the SELECTION= option in the MODEL statement. The simplest method (and the default) is SELECTION=NONE, for which PROC LOGISTIC fits the complete model as specified in the MODEL statement. The other four methods are FORWARD for forward selection, BACKWARD for backward elimination, STEPWISE for stepwise selection, and SCORE for best subsets selection. Intercept parameters are forced to stay in the model unless the NOINT option is specified.

When SELECTION=FORWARD, PROC LOGISTIC first estimates parameters for effects forced into the model. These effects are the intercepts and the first n explanatory effects in the MODEL statement, where n is the number specified by the START= or INCLUDE= option in the MODEL statement (n is zero by default). Next, the procedure computes the score chi-square statistic for each effect not in the model and examines the largest of these statistics. If it is significant at the SLENTRY= level, the corresponding effect is added to the model. Once an effect is entered in the

model, it is never removed from the model. The process is repeated until none of the remaining effects meet the specified level for entry or until the STOP= value is reached.

When SELECTION=BACKWARD, parameters for the complete model as specified in the MODEL statement are estimated unless the START= option is specified. In that case, only the parameters for the intercepts and the first n explanatory effects in the MODEL statement are estimated, where n is the number specified by the START= option. Results of the Wald test for individual parameters are examined. The least significant effect that does not meet the SLSTAY= level for staying in the model is removed. Once an effect is removed from the model, it remains excluded. The process is repeated until no other effect in the model meets the specified level for removal or until the STOP= value is reached. Backward selection is often less successful than forward or stepwise selection because the full model fit in the first step is the model most likely to result in a complete or quasi-complete separation of response values as described in the section "Existence of Maximum Likelihood Estimates" on page 3327.

The SELECTION=STEPWISE option is similar to the SELECTION=FORWARD option except that effects already in the model do not necessarily remain. Effects are entered into and removed from the model in such a way that each forward selection step can be followed by one or more backward elimination steps. The stepwise selection process terminates if no further effect can be added to the model or if the current model is identical to a previously visited model.

For SELECTION=SCORE, PROC LOGISTIC uses the branch-and-bound algorithm of Furnival and Wilson (1974) to find a specified number of models with the highest likelihood score (chi-square) statistic for all possible model sizes, from 1, 2, 3 effect models, and so on, up to the single model containing all of the explanatory effects. The number of models displayed for each model size is controlled by the BEST= option. You can use the START= option to impose a minimum model size, and you can use the STOP= option to impose a maximum model size. For instance, with BEST=3, START=2, and STOP=5, the SCORE selection method displays the best three models (that is, the three models with the highest score chi-squares) containing 2, 3, 4, and 5 effects. The SELECTION=SCORE option is not available for models with CLASS variables.

The options FAST, SEQUENTIAL, and STOPRES can alter the default criteria for entering or removing effects from the model when they are used with the FORWARD, BACKWARD, or STEPWISE selection method.

Model Fitting Information

For the jth observation, let $\hat{\pi}_j$ be the estimated probability of the observed response. The three criteria displayed by the LOGISTIC procedure are calculated as follows:

- –2 Log Likelihood:

$$-2 \text{ Log L} = -2 \sum_j \frac{w_j}{\sigma^2} f_j \log(\hat{\pi}_j)$$

  where w_j and f_j are the weight and frequency values of the jth observation, and σ^2 is the dispersion parameter, which equals 1 unless the SCALE= option is specified. For binary

response models that use *events/trials* MODEL syntax, this is equivalent to

$$-2 \text{ Log L} = -2 \sum_j \frac{w_j}{\sigma^2} f_j [r_j \log(\widehat{\pi}_j) + (n_j - r_j) \log(1 - \widehat{\pi}_j)]$$

where r_j is the number of events, n_j is the number of trials, and $\widehat{\pi}_j$ is the estimated event probability.

- Akaike Information Criterion:

  $$\text{AIC} = -2 \text{ Log L} + 2p$$

  where p is the number of parameters in the model. For cumulative response models, $p = k + s$, where k is the total number of response levels minus one and s is the number of explanatory effects. For the generalized logit model, $p = k(s + 1)$.

- Schwarz (Bayesian Information) Criterion:

  $$\text{SC} = -2 \text{ Log L} + p \log(\sum_j f_j)$$

  where p is the number of parameters in the model.

The AIC and SC statistics give two different ways of adjusting the –2 Log L statistic for the number of terms in the model and the number of observations used. These statistics can be used when comparing different models for the same data (for example, when you use the SELECTION=STEPWISE option in the MODEL statement). The models being compared do not have to be nested; lower values of the statistics indicate a more desirable model.

The difference in the –2 Log L statistics between the intercepts-only model and the specified model has a $p - k$ degree-of-freedom chi-square distribution under the null hypothesis that all the explanatory effects in the model are zero, where p is the number of parameters in the specified model and k is the number of intercepts. The likelihood ratio test in the "Testing Global Null Hypothesis: BETA=0" table displays this difference and the associated p-value for this statistic. The score and Wald tests in that table test the same hypothesis and are asymptotically equivalent; see the sections "Residual Chi-Square" on page 3331 and "Testing Linear Hypotheses about the Regression Coefficients" on page 3348 for details.

Generalized Coefficient of Determination

Cox and Snell (1989, pp. 208–209) propose the following generalization of the coefficient of determination to a more general linear model:

$$R^2 = 1 - \left\{ \frac{L(\mathbf{0})}{L(\widehat{\boldsymbol{\beta}})} \right\}^{\frac{2}{n}}$$

where $L(\mathbf{0})$ is the likelihood of the intercept-only model, $L(\widehat{\boldsymbol{\beta}})$ is the likelihood of the specified model, and n is the sample size. The quantity R^2 achieves a maximum of less than one for discrete models, where the maximum is given by

$$R^2_{\max} = 1 - \{L(\mathbf{0})\}^{\frac{2}{n}}$$

To take the frequency f_i and weight w_i of observation i into account, the sample size n is replaced in R^2 and $R^2_{\max}$ with $\sum_i f_i w_i$. Specifying the NORMALIZE option in the WEIGHT statement makes these coefficients invariant to the scale of the weights.

Nagelkerke (1991) proposes the following adjusted coefficient, which can achieve a maximum value of one:

$$\tilde{R}^2 = \frac{R^2}{R^2_{\max}}$$

Like the AIC and SC statistics described in the section "Model Fitting Information" on page 3329, R^2 and $\tilde{R}^2$ are most useful for comparing competing models that are not necessarily nested—larger values indicate better models. More properties and interpretation of R^2 and $\tilde{R}^2$ are provided in Nagelkerke (1991). In the "Testing Global Null Hypothesis: BETA=0" table, R^2 is labeled as "RSquare" and $\tilde{R}^2$ is labeled as "Max-rescaled RSquare." Use the RSQUARE option to request R^2 and $\tilde{R}^2$.

Score Statistics and Tests

To understand the general form of the score statistics, let $\boldsymbol{g}(\boldsymbol{\beta})$ be the vector of first partial derivatives of the log likelihood with respect to the parameter vector $\boldsymbol{\beta}$, and let $\mathbf{H}(\boldsymbol{\beta})$ be the matrix of second partial derivatives of the log likelihood with respect to $\boldsymbol{\beta}$. That is, $\boldsymbol{g}(\boldsymbol{\beta})$ is the gradient vector, and $\mathbf{H}(\boldsymbol{\beta})$ is the Hessian matrix. Let $\mathbf{I}(\boldsymbol{\beta})$ be either $-\mathbf{H}(\boldsymbol{\beta})$ or the expected value of $-\mathbf{H}(\boldsymbol{\beta})$. Consider a null hypothesis H_0. Let $\widehat{\boldsymbol{\beta}}_{H_0}$ be the MLE of $\boldsymbol{\beta}$ under H_0. The chi-square score statistic for testing H_0 is defined by

$$\boldsymbol{g}'(\widehat{\boldsymbol{\beta}}_{H_0})\mathbf{I}^{-1}(\widehat{\boldsymbol{\beta}}_{H_0})\boldsymbol{g}(\widehat{\boldsymbol{\beta}}_{H_0})$$

and it has an asymptotic χ^2 distribution with r degrees of freedom under H_0, where r is the number of restrictions imposed on $\boldsymbol{\beta}$ by H_0.

Residual Chi-Square

When you use SELECTION=FORWARD, BACKWARD, or STEPWISE, the procedure calculates a residual chi-square score statistic and reports the statistic, its degrees of freedom, and the p-value. This section describes how the statistic is calculated.

Suppose there are s explanatory effects of interest. The full cumulative response model has a parameter vector

$$\boldsymbol{\beta} = (\alpha_1, \ldots, \alpha_k, \beta_1, \ldots, \beta_s)'$$

where $\alpha_1, \ldots, \alpha_k$ are intercept parameters, and $\beta_1, \ldots, \beta_s$ are the common slope parameters for the s explanatory effects. The full generalized logit model has a parameter vector

$$\begin{aligned} \boldsymbol{\beta} &= (\alpha_1, \ldots, \alpha_k, \boldsymbol{\beta}_1', \ldots, \boldsymbol{\beta}_k')' \quad \text{with} \\ \boldsymbol{\beta}_i' &= (\beta_{i1}, \ldots, \beta_{is}), \quad i = 1, \ldots, k \end{aligned}$$

where β_{ij} is the slope parameter for the jth effect in the ith logit.

Consider the null hypothesis $H_0: \beta_{t+1} = \cdots = \beta_s = 0$, where $t < s$ for the cumulative response model, and $H_0: \beta_{i,t+1} = \cdots = \beta_{is} = 0, t < s, i = 1, \ldots, k$, for the generalized logit model. For the reduced model with t explanatory effects, let $\widehat{\alpha}_1, \ldots, \widehat{\alpha}_k$ be the MLEs of the unknown intercept parameters, let $\widehat{\beta}_1, \ldots, \widehat{\beta}_t$ be the MLEs of the unknown slope parameters, and let $\widehat{\boldsymbol{\beta}}_{i(t)}' = (\widehat{\beta}_{i1}, \ldots, \widehat{\beta}_{it}), i = 1, \ldots, k$, be those for the generalized logit model. The residual chi-square is the chi-square score statistic testing the null hypothesis H_0; that is, the residual chi-square is

$$\boldsymbol{g}'(\widehat{\boldsymbol{\beta}}_{H_0})\mathbf{I}^{-1}(\widehat{\boldsymbol{\beta}}_{H_0})\boldsymbol{g}(\widehat{\boldsymbol{\beta}}_{H_0})$$

where for the cumulative response model $\widehat{\boldsymbol{\beta}}_{H_0} = (\widehat{\alpha}_1, \ldots, \widehat{\alpha}_k, \widehat{\beta}_1, \ldots, \widehat{\beta}_t, 0, \ldots, 0)'$, and for the generalized logit model $\widehat{\boldsymbol{\beta}}_{H_0} = (\widehat{\alpha}_1, \ldots, \widehat{\alpha}_k, \widehat{\boldsymbol{\beta}}_{1(t)}', \mathbf{0}_{(s-t)}', \ldots \widehat{\boldsymbol{\beta}}_{k(t)}', \mathbf{0}_{(s-t)}')'$, where $\mathbf{0}_{(s-t)}$ denotes a vector of $s - t$ zeros.

The residual chi-square has an asymptotic chi-square distribution with $s - t$ degrees of freedom ($k(s - t)$ for the generalized logit model). A special case is the global score chi-square, where the reduced model consists of the k intercepts and no explanatory effects. The global score statistic is displayed in the "Testing Global Null Hypothesis: BETA=0" table. The table is not produced when the NOFIT option is used, but the global score statistic is displayed.

Testing Individual Effects Not in the Model

These tests are performed when you specify SELECTION=FORWARD or STEPWISE, and are displayed when the DETAILS option is specified. In the displayed output, the tests are labeled "Score Chi-Square" in the "Analysis of Effects Not in the Model" table and in the "Summary of Stepwise (Forward) Selection" table. This section describes how the tests are calculated.

Suppose that k intercepts and t explanatory variables (say $v_1, \ldots, v_t$) have been fit to a model and that v_{t+1} is another explanatory variable of interest. Consider a full model with the k intercepts and $t + 1$ explanatory variables $(v_1, \ldots, v_t, v_{t+1})$ and a reduced model with v_{t+1} excluded. The significance of v_{t+1} adjusted for $v_1, \ldots, v_t$ can be determined by comparing the corresponding residual chi-square with a chi-square distribution with one degree of freedom (k degrees of freedom for the generalized logit model).

Testing the Parallel Lines Assumption

For an ordinal response, PROC LOGISTIC performs a test of the parallel lines assumption. In the displayed output, this test is labeled "Score Test for the Equal Slopes Assumption" when the LINK= option is NORMIT or CLOGLOG. When LINK=LOGIT, the test is labeled as "Score Test for the

Proportional Odds Assumption" in the output. For small sample sizes, this test might be too liberal (Stokes, Davis, and Koch 2000). This section describes the methods used to calculate the test.

For this test the number of response levels, $k+1$, is assumed to be strictly greater than 2. Let Y be the response variable taking values $1, \ldots, k, k+1$. Suppose there are s explanatory variables. Consider the general cumulative model without making the parallel lines assumption

$$g(\Pr(Y \leq i \mid \boldsymbol{x})) = (1, \boldsymbol{x}')\boldsymbol{\beta}_i, \quad 1 \leq i \leq k$$

where $g(\cdot)$ is the link function, and $\boldsymbol{\beta}_i = (\alpha_i, \beta_{i1}, \ldots, \beta_{is})'$ is a vector of unknown parameters consisting of an intercept α_i and s slope parameters $\beta_{i1}, \ldots, \beta_{is}$. The parameter vector for this general cumulative model is

$$\boldsymbol{\beta} = (\boldsymbol{\beta}_1', \ldots, \boldsymbol{\beta}_k')'$$

Under the null hypothesis of parallelism H_0: $\beta_{1m} = \beta_{2m} = \cdots = \beta_{km}, 1 \leq m \leq s$, there is a single common slope parameter for each of the s explanatory variables. Let $\beta_1, \ldots, \beta_s$ be the common slope parameters. Let $\widehat{\alpha}_1, \ldots, \widehat{\alpha}_k$ and $\widehat{\beta}_1, \ldots, \widehat{\beta}_s$ be the MLEs of the intercept parameters and the common slope parameters. Then, under H_0, the MLE of $\boldsymbol{\beta}$ is

$$\widehat{\boldsymbol{\beta}}_{H_0} = (\widehat{\boldsymbol{\beta}}_1', \ldots, \widehat{\boldsymbol{\beta}}_k')' \quad \text{with} \quad \widehat{\boldsymbol{\beta}}_i = (\widehat{\alpha}_i, \widehat{\beta}_1, \ldots, \widehat{\beta}_s)' \quad 1 \leq i \leq k$$

and the chi-square score statistic $\boldsymbol{g}'(\widehat{\boldsymbol{\beta}}_{H_0})\mathbf{I}^{-1}(\widehat{\boldsymbol{\beta}}_{H_0})\boldsymbol{g}(\widehat{\boldsymbol{\beta}}_{H_0})$ has an asymptotic chi-square distribution with $s(k-1)$ degrees of freedom. This tests the parallel lines assumption by testing the equality of separate slope parameters simultaneously for all explanatory variables.

Confidence Intervals for Parameters

There are two methods of computing confidence intervals for the regression parameters. One is based on the profile-likelihood function, and the other is based on the asymptotic normality of the parameter estimators. The latter is not as time-consuming as the former, since it does not involve an iterative scheme; however, it is not thought to be as accurate as the former, especially with small sample size. You use the CLPARM= option to request confidence intervals for the parameters.

Likelihood Ratio-Based Confidence Intervals

The likelihood ratio-based confidence interval is also known as the profile-likelihood confidence interval. The construction of this interval is derived from the asymptotic χ^2 distribution of the generalized likelihood ratio test (Venzon and Moolgavkar 1988). Suppose that the parameter vector

is $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_s)'$ and you want to compute a confidence interval for β_j. The profile-likelihood function for $\beta_j = \gamma$ is defined as

$$l_j^*(\gamma) = \max_{\boldsymbol{\beta} \in \mathcal{B}_j(\gamma)} l(\boldsymbol{\beta})$$

where $\mathcal{B}_j(\gamma)$ is the set of all $\boldsymbol{\beta}$ with the jth element fixed at γ, and $l(\boldsymbol{\beta})$ is the log-likelihood function for $\boldsymbol{\beta}$. If $l_{\max} = l(\widehat{\boldsymbol{\beta}})$ is the log likelihood evaluated at the maximum likelihood estimate $\widehat{\boldsymbol{\beta}}$, then $2(l_{\max} - l_j^*(\beta_j))$ has a limiting chi-square distribution with one degree of freedom if β_j is the true parameter value. Let $l_0 = l_{\max} - 0.5\chi_1^2(1-\alpha)$, where $\chi_1^2(1-\alpha)$ is the $100(1-\alpha)$ percentile of the chi-square distribution with one degree of freedom. A $100(1-\alpha)\%$ confidence interval for β_j is

$$\{\gamma : l_j^*(\gamma) \geq l_0\}$$

The endpoints of the confidence interval are found by solving numerically for values of β_j that satisfy equality in the preceding relation. To obtain an iterative algorithm for computing the confidence limits, the log-likelihood function in a neighborhood of $\boldsymbol{\beta}$ is approximated by the quadratic function

$$\tilde{l}(\boldsymbol{\beta} + \boldsymbol{\delta}) = l(\boldsymbol{\beta}) + \boldsymbol{\delta}'\boldsymbol{g} + \frac{1}{2}\boldsymbol{\delta}'\mathbf{V}\boldsymbol{\delta}$$

where $\boldsymbol{g} = \boldsymbol{g}(\boldsymbol{\beta})$ is the gradient vector and $\mathbf{V} = \mathbf{V}(\boldsymbol{\beta})$ is the Hessian matrix. The increment $\boldsymbol{\delta}$ for the next iteration is obtained by solving the likelihood equations

$$\frac{d}{d\boldsymbol{\delta}}\{\tilde{l}(\boldsymbol{\beta} + \boldsymbol{\delta}) + \lambda(\boldsymbol{e}_j'\boldsymbol{\delta} - \gamma)\} = \mathbf{0}$$

where λ is the Lagrange multiplier, $\boldsymbol{e}_j$ is the jth unit vector, and γ is an unknown constant. The solution is

$$\boldsymbol{\delta} = -\mathbf{V}^{-1}(\boldsymbol{g} + \lambda \boldsymbol{e}_j)$$

By substituting this $\boldsymbol{\delta}$ into the equation $\tilde{l}(\boldsymbol{\beta} + \boldsymbol{\delta}) = l_0$, you can estimate λ as

$$\lambda = \pm\left(\frac{2(l_0 - l(\boldsymbol{\beta}) + \frac{1}{2}\boldsymbol{g}'\mathbf{V}^{-1}\boldsymbol{g})}{\boldsymbol{e}_j'\mathbf{V}^{-1}\boldsymbol{e}_j}\right)^{\frac{1}{2}}$$

The upper confidence limit for β_j is computed by starting at the maximum likelihood estimate of $\boldsymbol{\beta}$ and iterating with positive values of λ until convergence is attained. The process is repeated for the lower confidence limit by using negative values of λ.

Convergence is controlled by the value ϵ specified with the PLCONV= option in the MODEL statement (the default value of ϵ is 1E–4). Convergence is declared on the current iteration if the following two conditions are satisfied:

$$|l(\boldsymbol{\beta}) - l_0| \leq \epsilon$$

and

$$(\boldsymbol{g} + \lambda \boldsymbol{e}_j)'\mathbf{V}^{-1}(\boldsymbol{g} + \lambda \boldsymbol{e}_j) \leq \epsilon$$

Wald Confidence Intervals

Wald confidence intervals are sometimes called the normal confidence intervals. They are based on the asymptotic normality of the parameter estimators. The $100(1-\alpha)\%$ Wald confidence interval for β_j is given by

$$\widehat{\beta}_j \pm z_{1-\alpha/2}\widehat{\sigma}_j$$

where z_p is the $100p$th percentile of the standard normal distribution, $\widehat{\beta}_j$ is the maximum likelihood estimate of β_j, and $\widehat{\sigma}_j$ is the standard error estimate of $\widehat{\beta}_j$.

Odds Ratio Estimation

Consider a dichotomous response variable with outcomes *event* and *nonevent*. Consider a dichotomous risk factor variable X that takes the value 1 if the risk factor is present and 0 if the risk factor is absent. According to the logistic model, the log odds function, $\text{logit}(X)$, is given by

$$\text{logit}(X) \equiv \log\left(\frac{\Pr(event \mid X)}{\Pr(nonevent \mid X)}\right) = \alpha + X\beta$$

The odds ratio ψ is defined as the ratio of the odds for those with the risk factor ($X = 1$) to the odds for those without the risk factor ($X = 0$). The log of the odds ratio is given by

$$\log(\psi) \equiv \log(\psi(X = 1, X = 0)) = \text{logit}(X = 1) - \text{logit}(X = 0) = (\alpha + 1\times\beta) - (\alpha + 0\times\beta) = \beta$$

In general, the odds ratio can be computed by exponentiating the difference of the logits between any two population profiles. This is the approach taken by the ODDSRATIO statement, so the computations are available regardless of parameterization, interactions, and nestings. However, as shown in the preceding equation for $\log(\psi)$, odds ratios of main effects can be computed as functions of the parameter estimates, and the remainder of this section is concerned with this methodology.

The parameter, β, associated with X represents the change in the log odds from $X = 0$ to $X = 1$. So the odds ratio is obtained by simply exponentiating the value of the parameter associated with the risk factor. The odds ratio indicates how the odds of the *event* change as you change X from 0 to 1. For instance, $\psi = 2$ means that the odds of an event when $X = 1$ are twice the odds of an event when $X = 0$. You can also express this as follows: the percent change in the odds of an event from $X = 0$ to $X = 1$ is $(\psi - 1)100\% = 100\%$.

Suppose the values of the dichotomous risk factor are coded as constants a and b instead of 0 and 1. The odds when $X = a$ become $\exp(\alpha + a\beta)$, and the odds when $X = b$ become $\exp(\alpha + b\beta)$. The odds ratio corresponding to an increase in X from a to b is

$$\psi = \exp[(b-a)\beta] = [\exp(\beta)]^{b-a} \equiv [\exp(\beta)]^c$$

Note that for any a and b such that $c = b - a = 1$, $\psi = \exp(\beta)$. So the odds ratio can be interpreted as the change in the odds for any increase of one unit in the corresponding risk factor. However,

the change in odds for some amount other than one unit is often of greater interest. For example, a change of one pound in body weight might be too small to be considered important, while a change of 10 pounds might be more meaningful. The odds ratio for a change in X from a to b is estimated by raising the odds ratio estimate for a unit change in X to the power of $c = b - a$ as shown previously.

For a polytomous risk factor, the computation of odds ratios depends on how the risk factor is parameterized. For illustration, suppose that Race is a risk factor with four categories: White, Black, Hispanic, and Other.

For the effect parameterization scheme (PARAM=EFFECT) with White as the reference group (REF='White'), the design variables for Race are as follows:

	Design Variables		
Race	X_1	X_2	X_3
Black	1	0	0
Hispanic	0	1	0
Other	0	0	1
White	−1	−1	−1

The log odds for Black is

$$\begin{aligned} \text{logit(Black)} &= \alpha + (X_1 = 1)\beta_1 + (X_2 = 0)\beta_2 + (X_3 = 0)\beta_3 \\ &= \alpha + \beta_1 \end{aligned}$$

The log odds for White is

$$\begin{aligned} \text{logit(White)} &= \alpha + (X_1 = -1)\beta_1 + (X_2 = -1)\beta_2 + (X_3 = -1)\beta_3 \\ &= \alpha - \beta_1 - \beta_2 - \beta_3 \end{aligned}$$

Therefore, the log odds ratio of Black versus White becomes

$$\begin{aligned} \log(\psi(\text{Black, White})) &= \text{logit(Black)} - \text{logit(White)} \\ &= 2\beta_1 + \beta_2 + \beta_3 \end{aligned}$$

For the reference cell parameterization scheme (PARAM=REF) with White as the reference cell, the design variables for race are as follows:

	Design Variables		
Race	X_1	X_2	X_3
Black	1	0	0
Hispanic	0	1	0
Other	0	0	1
White	0	0	0

The log odds ratio of Black versus White is given by

$$\begin{aligned}\log(\psi(\text{Black, White})) &= \text{logit(Black)} - \text{logit(White)} \\ &= (\alpha + (X_1 = 1)\beta_1 + (X_2 = 0)\beta_2 + (X_3 = 0)\beta_3) - \\ &\quad (\alpha + (X_1 = 0)\beta_1 + (X_2 = 0)\beta_2 + (X_3 = 0)\beta_3) \\ &= \beta_1\end{aligned}$$

For the GLM parameterization scheme (PARAM=GLM), the design variables are as follows:

	Design Variables			
Race	X_1	X_2	X_3	X_4
Black	1	0	0	0
Hispanic	0	1	0	0
Other	0	0	1	0
White	0	0	0	1

The log odds ratio of Black versus White is

$$\begin{aligned}\log(\psi(\text{Black, White})) &= \text{logit(Black)} - \text{logit(White)} \\ &= (\alpha + (X_1 = 1)\beta_1 + (X_2 = 0)\beta_2 + (X_3 = 0)\beta_3 + (X_4 = 0)\beta_4) - \\ &\quad (\alpha + (X_1 = 0)\beta_1 + (X_2 = 0)\beta_2 + (X_3 = 0)\beta_3 + (X_4 = 1)\beta_4) \\ &= \beta_1 - \beta_4\end{aligned}$$

Consider the hypothetical example of heart disease among race in Hosmer and Lemeshow (2000, p. 56). The entries in the following contingency table represent counts:

	Race			
Disease Status	White	Black	Hispanic	Other
Present	5	20	15	10
Absent	20	10	10	10

The computation of odds ratio of Black versus White for various parameterization schemes is tabulated in Table 51.4.

Table 51.4 Odds Ratio of Heart Disease Comparing Black to White

	Parameter Estimates				
PARAM=	$\widehat{\beta}_1$	$\widehat{\beta}_2$	$\widehat{\beta}_3$	$\widehat{\beta}_4$	**Odds Ratio Estimates**
EFFECT	0.7651	0.4774	0.0719		$\exp(2 \times 0.7651 + 0.4774 + 0.0719) = 8$
REF	2.0794	1.7917	1.3863		$\exp(2.0794) = 8$
GLM	2.0794	1.7917	1.3863	0.0000	$\exp(2.0794) = 8$

Since the log odds ratio ($\log(\psi)$) is a linear function of the parameters, the Wald confidence interval for $\log(\psi)$ can be derived from the parameter estimates and the estimated covariance matrix.

Confidence intervals for the odds ratios are obtained by exponentiating the corresponding confidence limits for the log odd ratios. In the displayed output of PROC LOGISTIC, the "Odds Ratio Estimates" table contains the odds ratio estimates and the corresponding 95% Wald confidence intervals. For continuous explanatory variables, these odds ratios correspond to a unit increase in the risk factors.

To customize odds ratios for specific units of change for a continuous risk factor, you can use the UNITS statement to specify a list of relevant units for each explanatory variable in the model. Estimates of these customized odds ratios are given in a separate table. Let (L_j, U_j) be a confidence interval for $\log(\psi)$. The corresponding lower and upper confidence limits for the customized odds ratio $\exp(c\beta_j)$ are $\exp(cL_j)$ and $\exp(cU_j)$, respectively (for $c > 0$), or $\exp(cU_j)$ and $\exp(cL_j)$, respectively (for $c < 0$). You use the CLODDS= option or ODDSRATIO statement to request the confidence intervals for the odds ratios.

For a generalized logit model, odds ratios are computed similarly, except k odds ratios are computed for each effect, corresponding to the k logits in the model.

Rank Correlation of Observed Responses and Predicted Probabilities

The predicted mean score of an observation is the sum of the Ordered Values (shown in the "Response Profile" table) minus one, weighted by the corresponding predicted probabilities for that observation; that is, the predicted means score $= \sum_{i=1}^{k+1}(i-1)\widehat{\pi}_i$, where $k+1$ is the number of response levels and $\widehat{\pi}_i$ is the predicted probability of the ith (ordered) response.

A pair of observations with different observed responses is said to be *concordant* if the observation with the lower ordered response value has a lower predicted mean score than the observation with the higher ordered response value. If the observation with the lower ordered response value has a higher predicted mean score than the observation with the higher ordered response value, then the pair is *discordant*. If the pair is neither concordant nor discordant, it is a *tie*. Enumeration of the total numbers of concordant and discordant pairs is carried out by categorizing the predicted mean score into intervals of length $k/500$ and accumulating the corresponding frequencies of observations. Note that the length of these intervals can be modified by specification of the BINWIDTH= option in the MODEL statement.

Let N be the sum of observation frequencies in the data. Suppose there are a total of t pairs with different responses: n_c of them are concordant, n_d of them are discordant, and $t - n_c - n_d$ of them are tied. PROC LOGISTIC computes the following four indices of rank correlation for assessing the predictive ability of a model:

$$
\begin{array}{lcl}
c & = & (n_c + 0.5(t - n_c - n_d))/t \\
\text{Somers' } D \text{ (Gini coefficient)} & = & (n_c - n_d)/t \\
\text{Goodman-Kruskal Gamma} & = & (n_c - n_d)/(n_c + n_d) \\
\text{Kendall's Tau-}a & = & (n_c - n_d)/(0.5N(N-1))
\end{array}
$$

If there are no ties, then Somers' D (Gini's coefficient) $= 2c-1$. Note that the concordance index, c, also gives an estimate of the area under the receiver operating characteristic (ROC) curve when the response is binary (Hanley and McNeil 1982). See the section "ROC Computations" on page 3347 for more information about this area.

For binary responses, the predicted mean score is equal to the predicted probability for Ordered Value 2. As such, the preceding definition of concordance is consistent with the definition used in previous releases for the binary response model.

These statistics are not available when the STRATA statement is specified.

Linear Predictor, Predicted Probability, and Confidence Limits

This section describes how predicted probabilities and confidence limits are calculated by using the maximum likelihood estimates (MLEs) obtained from PROC LOGISTIC. For a specific example, see the section "Getting Started: LOGISTIC Procedure" on page 3260. Predicted probabilities and confidence limits can be output to a data set with the OUTPUT statement.

Binary and Cumulative Response Models

For a vector of explanatory variables $\boldsymbol{x}$, the linear predictor

$$\eta_i = g(\Pr(Y \le i \mid \boldsymbol{x})) = \alpha_i + \boldsymbol{x}'\boldsymbol{\beta} \quad 1 \le i \le k$$

is estimated by

$$\hat{\eta}_i = \widehat{\alpha}_i + \boldsymbol{x}'\widehat{\boldsymbol{\beta}}$$

where $\widehat{\alpha}_i$ and $\widehat{\boldsymbol{\beta}}$ are the MLEs of α_i and $\boldsymbol{\beta}$. The estimated standard error of η_i is $\hat{\sigma}(\hat{\eta}_i)$, which can be computed as the square root of the quadratic form $(1, \boldsymbol{x}')\widehat{\mathbf{V}}_{\boldsymbol{\beta}}(1, \boldsymbol{x}')'$, where $\widehat{\mathbf{V}}_{\boldsymbol{\beta}}$ is the estimated covariance matrix of the parameter estimates. The asymptotic $100(1-\alpha)\%$ confidence interval for η_i is given by

$$\hat{\eta}_i \pm z_{\alpha/2}\hat{\sigma}(\hat{\eta}_i)$$

where $z_{\alpha/2}$ is the $100(1-\alpha/2)$ percentile point of a standard normal distribution.

The predicted probability and the $100(1-\alpha)\%$ confidence limits for $\pi_i = \Pr(Y \le i \mid \boldsymbol{x})$ are obtained by back-transforming the corresponding measures for the linear predictor, as shown in the following table:

Link	Predicted Probability	100(1–α)% Confidence Limits
LOGIT	$1/(1+\exp(-\hat{\eta}_i))$	$1/(1+\exp(-\hat{\eta}_i \pm z_{\alpha/2}\hat{\sigma}(\hat{\eta}_i)))$
PROBIT	$\Phi(\hat{\eta}_i)$	$\Phi(\hat{\eta}_i \pm z_{\alpha/2}\hat{\sigma}(\hat{\eta}_i))$
CLOGLOG	$1-\exp(-\exp(\hat{\eta}_i))$	$1-\exp(-\exp(\hat{\eta}_i \pm z_{\alpha/2}\hat{\sigma}(\hat{\eta}_i)))$

The CONTRAST statement also enables you to estimate the exponentiated contrast, $e^{\hat{\eta}_i}$. The corresponding standard error is $e^{\hat{\eta}_i}\hat{\sigma}(\hat{\eta}_i)$, and the confidence limits are computed by exponentiating those for the linear predictor: $\exp\{\hat{\eta}_i \pm z_{\alpha/2}\hat{\sigma}(\hat{\eta}_i)\}$.

Generalized Logit Model

For a vector of explanatory variables $\boldsymbol{x}$, define the linear predictors $\eta_i = \alpha_i + \boldsymbol{x}'\boldsymbol{\beta}_i$, and let π_i denote the probability of obtaining the response value i:

$$\pi_i = \begin{cases} \pi_{k+1} e^{\eta_i} & 1 \le i \le k \\ \dfrac{1}{1 + \sum_{j=1}^{k} e^{\eta_j}} & i = k+1 \end{cases}$$

By the *delta method*,

$$\sigma^2(\pi_i) = \left(\frac{\partial \pi_i}{\partial \boldsymbol{\beta}}\right)' \mathbf{V}(\boldsymbol{\beta}) \frac{\partial \pi_i}{\partial \boldsymbol{\beta}}$$

A $100(1-\alpha)\%$ confidence level for π_i is given by

$$\widehat{\pi}_i \pm z_{\alpha/2} \hat{\sigma}(\widehat{\pi}_i)$$

where $\widehat{\pi}_i$ is the estimated expected probability of response i, and $\hat{\sigma}(\widehat{\pi}_i)$ is obtained by evaluating $\sigma(\pi_i)$ at $\boldsymbol{\beta} = \widehat{\boldsymbol{\beta}}$.

Note that the contrast $\hat{\eta}_i$ and exponentiated contrast $e^{\hat{\eta}_i}$, their standard errors, and their confidence intervals are computed in the same fashion as for the cumulative response models, replacing $\boldsymbol{\beta}$ with $\boldsymbol{\beta}_i$.

Classification Table

For binary response data, the response is either an *event* or a *nonevent*. In PROC LOGISTIC, the response with Ordered Value 1 is regarded as the *event*, and the response with Ordered Value 2 is the *nonevent*. PROC LOGISTIC models the probability of the *event*. From the fitted model, a predicted *event* probability can be computed for each observation. A method to compute a reduced-bias estimate of the predicted probability is given in the section "Predicted Probability of an Event for Classification" on page 3341. If the predicted *event* probability exceeds or equals some cutpoint value $z \in [0, 1]$, the observation is predicted to be an *event* observation; otherwise, it is predicted as a *nonevent*. A 2×2 frequency table can be obtained by cross-classifying the observed and predicted responses. The CTABLE option produces this table, and the PPROB= option selects one or more cutpoints. Each cutpoint generates a classification table. If the PEVENT= option is also specified, a classification table is produced for each combination of PEVENT= and PPROB= values.

The accuracy of the classification is measured by its *sensitivity* (the ability to predict an *event* correctly) and specificity (the ability to predict a *nonevent* correctly). *Sensitivity* is the proportion of *event* responses that were predicted to be *events*. *Specificity* is the proportion of *nonevent* responses that were predicted to be *nonevents*. PROC LOGISTIC also computes three other conditional probabilities: *false positive rate*, *false negative rate*, and *rate of cosrrect classification*. The *false positive rate* is the proportion of predicted *event* responses that were observed as *nonevents*. The *false negative rate* is the proportion of predicted *nonevent* responses that were observed as *events*. Given prior probabilities specified with the PEVENT= option, these conditional probabilities can be computed as posterior probabilities by using Bayes' theorem.

Predicted Probability of an Event for Classification

When you classify a set of binary data, if the same observations used to fit the model are also used to estimate the classification error, the resulting error-count estimate is biased. One way of reducing the bias is to remove the binary observation to be classified from the data, reestimate the parameters of the model, and then classify the observation based on the new parameter estimates. However, it would be costly to fit the model by leaving out each observation one at a time. The LOGISTIC procedure provides a less expensive one-step approximation to the preceding parameter estimates. Let $\widehat{\boldsymbol{\beta}}$ be the MLE of the parameter vector $(\alpha, \beta_1, \ldots, \beta_s)'$ based on all observations. Let $\widehat{\boldsymbol{\beta}}_{(j)}$ denote the MLE computed without the jth observation. The one-step estimate of $\widehat{\boldsymbol{\beta}}_{(j)}$ is given by

$$\widehat{\boldsymbol{\beta}}^1_{(j)} = \widehat{\boldsymbol{\beta}} - \frac{w_j(y_j - \widehat{\pi}_j)}{1 - h_{jj}} \widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}}) \begin{pmatrix} 1 \\ \boldsymbol{x}_j \end{pmatrix}$$

where

y_j	is 1 for an observed event response and 0 otherwise
w_j	is the weight of the observation
$\widehat{\pi}_j$	is the predicted event probability based on $\widehat{\boldsymbol{\beta}}$
h_{jj}	is the hat diagonal element (defined on page 3349) with $n_j = 1$ and $r_j = y_j$
$\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})$	is the estimated covariance matrix of $\widehat{\boldsymbol{\beta}}$

False Positive and Negative Rates Using Bayes' Theorem

Suppose n_1 of n individuals experience an event, such as a disease. Let this group be denoted by $\mathcal{C}_1$, and let the group of the remaining $n_2 = n - n_1$ individuals who do not have the disease be denoted by $\mathcal{C}_2$. The jth individual is classified as giving a positive response if the predicted probability of disease ($\widehat{\pi}^*_{(j)}$) is large. The probability $\widehat{\pi}^*_{(j)}$ is the reduced-bias estimate based on the one-step approximation given in the preceding section. For a given cutpoint z, the jth individual is predicted to give a positive response if $\widehat{\pi}^*_{(j)} \geq z$.

Let B denote the event that a subject has the disease, and let $\bar{B}$ denote the event of not having the disease. Let A denote the event that the subject responds positively, and let $\bar{A}$ denote the event of responding negatively. Results of the classification are represented by two conditional probabilities, $\Pr(A|B)$ and $\Pr(A|\bar{B})$, where $\Pr(A|B)$ is the sensitivity and $\Pr(A|\bar{B})$ is one minus the specificity.

These probabilities are given by

$$\Pr(A|B) = \frac{\sum_{j \in \mathcal{C}_1} I(\widehat{\pi}^*_{(j)} \geq z)}{n_1}$$
$$\Pr(A|\bar{B}) = \frac{\sum_{j \in \mathcal{C}_2} I(\widehat{\pi}^*_{(j)} \geq z)}{n_2}$$

where $I(\cdot)$ is the indicator function.

Bayes' theorem is used to compute the error rates of the classification. For a given prior probability $\Pr(B)$ of the disease, the false positive rate P_{F+} and the false negative rate P_{F-} are given by Fleiss (1981, pp. 4–5) as follows:

$$\begin{aligned} P_{F+} = \Pr(\bar{B}|A) &= \frac{\Pr(A|\bar{B})[1-\Pr(B)]}{\Pr(A|\bar{B}) + \Pr(B)[\Pr(A|B) - \Pr(A|\bar{B})]} \\ P_{F-} = \Pr(B|\bar{A}) &= \frac{[1-\Pr(A|B)]\Pr(B)}{1-\Pr(A|\bar{B}) - \Pr(B)[\Pr(A|B) - \Pr(A|\bar{B})]} \end{aligned}$$

The prior probability $\Pr(B)$ can be specified by the PEVENT= option. If the PEVENT= option is not specified, the sample proportion of diseased individuals is used; that is, $\Pr(B) = n_1/n$. In such a case, the false positive rate and the false negative rate reduce to

$$\begin{aligned} P_{F+} &= \frac{\sum_{j\in\mathcal{C}_2} I(\widehat{\pi}^*_{(j)} \geq z)}{\sum_{j\in\mathcal{C}_1} I(\widehat{\pi}^*_{(j)} \geq z) + \sum_{j\in\mathcal{C}_2} I(\widehat{\pi}^*_{(j)} \geq z)} \\ P_{F-} &= \frac{\sum_{j\in\mathcal{C}_1} I(\widehat{\pi}^*_{(j)} < z)}{\sum_{j\in\mathcal{C}_1} I(\widehat{\pi}^*_{(j)} < z) + \sum_{j\in\mathcal{C}_2} I(\widehat{\pi}^*_{(j)} < z)} \end{aligned}$$

Note that for a stratified sampling situation in which n_1 and n_2 are chosen a priori, n_1/n is not a desirable estimate of $\Pr(B)$. For such situations, the PEVENT= option should be specified.

Overdispersion

For a correctly specified model, the Pearson chi-square statistic and the deviance, divided by their degrees of freedom, should be approximately equal to one. When their values are much larger than one, the assumption of binomial variability might not be valid and the data are said to exhibit overdispersion. Underdispersion, which results in the ratios being less than one, occurs less often in practice.

When fitting a model, there are several problems that can cause the goodness-of-fit statistics to exceed their degrees of freedom. Among these are such problems as outliers in the data, using the wrong link function, omitting important terms from the model, and needing to transform some predictors. These problems should be eliminated before proceeding to use the following methods to correct for overdispersion.

Rescaling the Covariance Matrix

One way of correcting overdispersion is to multiply the covariance matrix by a dispersion parameter. This method assumes that the sample sizes in each subpopulation are approximately equal. You can supply the value of the dispersion parameter directly, or you can estimate the dispersion parameter based on either the Pearson chi-square statistic or the deviance for the fitted model.

The Pearson chi-square statistic χ^2_P and the deviance χ^2_D are given by

$$\chi^2_P = \sum_{i=1}^{m}\sum_{j=1}^{k+1}\frac{(r_{ij}-n_i\widehat{\pi}_{ij})^2}{n_i\widehat{\pi}_{ij}}$$

$$\chi^2_D = 2\sum_{i=1}^{m}\sum_{j=1}^{k+1} r_{ij}\log\left(\frac{r_{ij}}{n_i\widehat{\pi}_{ij}}\right)$$

where m is the number of subpopulation profiles, $k+1$ is the number of response levels, r_{ij} is the total weight (sum of the product of the frequencies and the weights) associated with jth level responses in the ith profile, $n_i = \sum_{j=1}^{k+1} r_{ij}$, and $\widehat{\pi}_{ij}$ is the fitted probability for the jth level at the ith profile. Each of these chi-square statistics has $mk-p$ degrees of freedom, where p is the number of parameters estimated. The dispersion parameter is estimated by

$$\widehat{\sigma^2} = \begin{cases} \chi^2_P/(mk-p) & \text{SCALE=PEARSON} \\ \chi^2_D/(mk-p) & \text{SCALE=DEVIANCE} \\ (constant)^2 & \text{SCALE=}constant \end{cases}$$

In order for the Pearson statistic and the deviance to be distributed as chi-square, there must be sufficient replication within the subpopulations. When this is not true, the data are sparse, and the p-values for these statistics are not valid and should be ignored. Similarly, these statistics, divided by their degrees of freedom, cannot serve as indicators of overdispersion. A large difference between the Pearson statistic and the deviance provides some evidence that the data are too sparse to use either statistic.

You can use the AGGREGATE (or AGGREGATE=) option to define the subpopulation profiles. If you do not specify this option, each observation is regarded as coming from a separate subpopulation. For *events/trials* syntax, each observation represents n Bernoulli trials, where n is the value of the *trials* variable; for *single-trial* syntax, each observation represents a single trial. Without the AGGREGATE (or AGGREGATE=) option, the Pearson chi-square statistic and the deviance are calculated only for *events/trials* syntax.

Note that the parameter estimates are not changed by this method. However, their standard errors are adjusted for overdispersion, affecting their significance tests.

Williams' Method

Suppose that the data consist of n binomial observations. For the ith observation, let r_i/n_i be the observed proportion and let $\boldsymbol{x}_i$ be the associated vector of explanatory variables. Suppose that the response probability for the ith observation is a random variable P_i with mean and variance

$$E(P_i) = \pi_i \quad \text{and} \quad V(P_i) = \phi\pi_i(1-\pi_i)$$

where p_i is the probability of the event, and ϕ is a nonnegative but otherwise unknown scale parameter. Then the mean and variance of r_i are

$$E(r_i) = n_i\pi_i \quad \text{and} \quad V(r_i) = n_i\pi_i(1-\pi_i)[1+(n_i-1)\phi]$$

Williams (1982) estimates the unknown parameter ϕ by equating the value of Pearson's chi-square statistic for the full model to its approximate expected value. Suppose w_i^* is the weight associated with the ith observation. The Pearson chi-square statistic is given by

$$\chi^2 = \sum_{i=1}^{n} \frac{w_i^*(r_i - n_i\widehat{\pi}_i)^2}{n_i\widehat{\pi}_i(1-\widehat{\pi}_i)}$$

Let $g'(\cdot)$ be the first derivative of the link function $g(\cdot)$. The approximate expected value of χ^2 is

$$E_{\chi^2} = \sum_{i=1}^{n} w_i^*(1 - w_i^* v_i d_i)[1 + \phi(n_i - 1)]$$

where $v_i = n_i/(\pi_i(1-\pi_i)[g'(\pi_i)]^2)$ and d_i is the variance of the linear predictor $\widehat{\alpha}_i + \boldsymbol{x}_i'\widehat{\boldsymbol{\beta}}$. The scale parameter ϕ is estimated by the following iterative procedure.

At the start, let $w_i^* = 1$ and let π_i be approximated by r_i/n_i, $i = 1, 2, \ldots, n$. If you apply these weights and approximated probabilities to χ^2 and E_{χ^2} and then equate them, an initial estimate of ϕ is

$$\hat{\phi}_0 = \frac{\chi^2 - (n-p)}{\sum_i (n_i - 1)(1 - v_i d_i)}$$

where p is the total number of parameters. The initial estimates of the weights become $\hat{w}_{i0}^* = [1 + (n_i - 1)\hat{\phi}_0]^{-1}$. After a weighted fit of the model, the $\widehat{\alpha}_i$ and $\widehat{\boldsymbol{\beta}}$ are recalculated, and so is χ^2. Then a revised estimate of ϕ is given by

$$\hat{\phi}_1 = \frac{\chi^2 - \sum_i w_i^*(1 - w_i^* v_i d_i)}{w_i^*(n_i - 1)(1 - w_i^* v_i d_i)}$$

The iterative procedure is repeated until χ^2 is very close to its degrees of freedom.

Once ϕ has been estimated by $\hat{\phi}$ under the full model, weights of $(1 + (n_i - 1)\hat{\phi})^{-1}$ can be used to fit models that have fewer terms than the full model. See Example 51.10 for an illustration.

NOTE: If the WEIGHT statement is specified with the NORMALIZE option, then the initial w_i^* values are set to the normalized weights, and the weights resulting from Williams' method will not add up to the actual sample size. However, the estimated covariance matrix of the parameter estimates remains invariant to the scale of the WEIGHT variable.

The Hosmer-Lemeshow Goodness-of-Fit Test

Sufficient replication within subpopulations is required to make the Pearson and deviance goodness-of-fit tests valid. When there are one or more continuous predictors in the model, the data are often too sparse to use these statistics. Hosmer and Lemeshow (2000) proposed a statistic that they show, through simulation, is distributed as chi-square when there is no replication in any of the subpopulations. This test is available only for binary response models.

First, the observations are sorted in increasing order of their estimated event probability. The event is the response level specified in the response variable option EVENT=, or the response level that is not specified in the REF= option, or, if neither of these options was specified, then the event is the response level identified in the "Response Profiles" table as "Ordered Value 1". The observations are then divided into approximately 10 groups according to the following scheme. Let N be the total number of subjects. Let M be the target number of subjects for each group given by

$$M = [0.1 \times N + 0.5]$$

where $[x]$ represents the integral value of x. If the *single-trial* syntax is used, blocks of subjects are formed of observations with identical values of the explanatory variables. Blocks of subjects are not divided when being placed into groups.

Suppose there are n_1 subjects in the first block and n_2 subjects in the second block. The first block of subjects is placed in the first group. Subjects in the second block are added to the first group if

$$n_1 < M \quad \text{and} \quad n_1 + [0.5 \times n_2] \leq M$$

Otherwise, they are placed in the second group. In general, suppose subjects of the $(j-1)$th block have been placed in the kth group. Let c be the total number of subjects currently in the kth group. Subjects for the jth block (containing n_j subjects) are also placed in the kth group if

$$c < M \quad \text{and} \quad c + [0.5 \times n_j] \leq M$$

Otherwise, the n_j subjects are put into the next group. In addition, if the number of subjects in the last group does not exceed $[0.05 \times N]$ (half the target group size), the last two groups are collapsed to form only one group.

Note that the number of groups, g, can be smaller than 10 if there are fewer than 10 patterns of explanatory variables. There must be at least three groups in order for the Hosmer-Lemeshow statistic to be computed.

The Hosmer-Lemeshow goodness-of-fit statistic is obtained by calculating the Pearson chi-square statistic from the $2 \times g$ table of observed and expected frequencies, where g is the number of groups. The statistic is written

$$\chi^2_{HL} = \sum_{i=1}^{g} \frac{(O_i - N_i \bar{\pi}_i)^2}{N_i \bar{\pi}_i (1 - \bar{\pi}_i)}$$

where N_i is the total frequency of subjects in the ith group, O_i is the total frequency of event outcomes in the ith group, and $\bar{\pi}_i$ is the average estimated predicted probability of an event outcome for the ith group. (Note that the predicted probabilities are computed as shown in the section "Linear Predictor, Predicted Probability, and Confidence Limits" on page 3339 and are not the cross validated estimates discussed in the section "Classification Table" on page 3340.) The Hosmer-Lemeshow statistic is then compared to a chi-square distribution with $(g-n)$ degrees of freedom, where the value of n can be specified in the LACKFIT option in the MODEL statement. The default is $n = 2$. Large values of χ^2_{HL} (and small p-values) indicate a lack of fit of the model.

Receiver Operating Characteristic Curves

ROC curves are used to evaluate and compare the performance of diagnostic tests; they can also be used to evaluate model fit. An ROC curve is just a plot of the proportion of true positives (events predicted to be events) versus the proportion of false positives (nonevents predicted to be events).

In a sample of n individuals, suppose n_1 individuals are observed to have a certain condition or event. Let this group be denoted by $\mathcal{C}_1$, and let the group of the remaining $n_2 = n - n_1$ individuals who do not have the condition be denoted by $\mathcal{C}_2$. Risk factors are identified for the sample, and a logistic regression model is fitted to the data. For the jth individual, an estimated probability $\widehat{\pi}_j$ of the event of interest is calculated. Note that the $\widehat{\pi}_j$ are computed as shown in the section "Linear Predictor, Predicted Probability, and Confidence Limits" on page 3339 and are not the cross validated estimates discussed in the section "Classification Table" on page 3340.

Suppose the n individuals undergo a test for predicting the event and the test is based on the estimated probability of the event. Higher values of this estimated probability are assumed to be associated with the event. A receiver operating characteristic (ROC) curve can be constructed by varying the cutpoint that determines which estimated event probabilities are considered to predict the event. For each cutpoint z, the following measures can be output to a data set by specifying the OUTROC= option in the MODEL statement or the OUTROC= option in the SCORE statement:

$$
\begin{aligned}
\text{\_POS\_}(z) &= \sum_{i \in \mathcal{C}_1} I(\widehat{\pi}_i \geq z) \\
\text{\_NEG\_}(z) &= \sum_{i \in \mathcal{C}_2} I(\widehat{\pi}_i < z) \\
\text{\_FALPOS\_}(z) &= \sum_{i \in \mathcal{C}_2} I(\widehat{\pi}_i \geq z) \\
\text{\_FALNEG\_}(z) &= \sum_{i \in \mathcal{C}_1} I(\widehat{\pi}_i < z) \\
\text{\_SENSIT\_}(z) &= \frac{\text{\_POS\_}(z)}{n_1} \\
\text{\_1MSPEC\_}(z) &= \frac{\text{\_FALPOS\_}(z)}{n_2}
\end{aligned}
$$

where $I(\cdot)$ is the indicator function.

Note that _POS_(z) is the number of correctly predicted event responses, _NEG_(z) is the number of correctly predicted nonevent responses, _FALPOS_(z) is the number of falsely predicted event responses, _FALNEG_(z) is the number of falsely predicted nonevent responses, _SENSIT_(z) is the sensitivity of the test, and _1MSPEC_(z) is one minus the specificity of the test.

The ROC curve is a plot of sensitivity (_SENSIT_) against 1–specificity (_1MSPEC_). The plot can be produced by using the PLOTS option or by using the GPLOT or SGPLOT procedure with the OUTROC= data set. See Example 51.7 for an illustration. The area under the ROC curve, as determined by the trapezoidal rule, is estimated by the concordance index, c, in the "Association of Predicted Probabilities and Observed Responses" table.

Comparing ROC Curves

ROC curves can be created from each model fit in a selection routine, from the specified model in the MODEL statement, from specified models in ROC statements, or from input variables which act as $\widehat{\pi}$ in the preceding discussion. Association statistics are computed for these models, and the models are compared when the ROCCONTRAST statement is specified. The ROC comparisons are performed by using a contrast matrix to take differences of the areas under the empirical ROC curves (DeLong, DeLong, and Clarke-Pearson 1988). For example, if you have three curves and the second curve is the reference, the contrast used for the overall test is

$$\mathbf{L}_1 = \begin{pmatrix} \boldsymbol{l}_1' \\ \boldsymbol{l}_2' \end{pmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix}$$

and you can optionally estimate and test each row of this contrast, in order to test the difference between the reference curve and each of the other curves. If you do not want to use a reference curve, the global test optionally uses the following contrast:

$$\mathbf{L}_2 = \begin{pmatrix} \boldsymbol{l}_1' \\ \boldsymbol{l}_2' \end{pmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & 1 & -1 \end{bmatrix}$$

You can also specify your own contrast matrix. Instead of estimating the rows of these contrasts, you can request that the difference between every pair of ROC curves be estimated and tested.

By default for the reference contrast, the specified or selected model is used as the reference unless the NOFIT option is specified in the MODEL statement, in which case the first ROC model is the reference.

In order to label the contrasts, a name is attached to every model. The name for the specified or selected model is the MODEL statement label, or "Model" if the MODEL label is not present. The ROC statement models are named with their labels, or as "ROC*i*" for the *i*th ROC statement if a label is not specified. The contrast $\mathbf{L}_1$ is labeled as "Reference = ModelName", where ModelName is the reference model name, while $\mathbf{L}_2$ is labeled "Adjacent Pairwise Differences". The estimated rows of the contrast matrix are labeled "ModelName1 – ModelName2". In particular, for the rows of $\mathbf{L}_1$, ModelName2 is the reference model name. If you specify your own contrast matrix, then the contrast is labeled "Specified" and the *i*th contrast row estimates are labeled "Row*i*".

If **ods graphics on** is specified, then all ROC curves are displayed individually and are also overlaid in a final display. If a selection method is specified, then the curves produced in each step of the model selection process are overlaid onto a single plot and are labeled "Step*i*", and the selected model is displayed on a separate plot and on a plot with curves from specified ROC statements. See Example 51.8 for an example.

ROC Computations

The trapezoidal area under an empirical ROC curve is equal to the Mann-Whitney two-sample rank measure of association statistic (a generalized U-statistic) applied to two samples, $\{X_i\}, i = 1, \ldots, n_1$, in $\mathcal{C}_1$ and $\{Y_i\}, i = 1, \ldots, n_2$, in $\mathcal{C}_2$. PROC LOGISTIC uses the predicted probabilities in place of $\mathbf{X}$ and $\mathbf{Y}$; however, in general any criterion could be used. Denote the frequency of observation i in $\mathcal{C}_k$ as f_{ki}, and denote the total frequency in $\mathcal{C}_k$ as F_k. The WEIGHTED option

replaces f_{ki} with $f_{ki}w_{ki}$, where w_{ki} is the weight of observation i in group C_k. The trapezoidal area under the curve is computed as

$$\hat{c} = \frac{1}{F_1 F_2} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \psi(X_i, Y_j) f_{1i} f_{2j}$$

$$\psi(X,Y) = \begin{cases} 1 & Y < X \\ \frac{1}{2} & Y = X \\ 0 & Y > X \end{cases}$$

so that $E(\hat{c}) = \Pr(Y < X) + \frac{1}{2}\Pr(Y = X)$. Note that the concordance index, c, in the "Association of Predicted Probabilities and Observed Responses" table is computed by creating 500 bins and binning the X_i and Y_j; this results in more ties than the preceding method (unless the BINWIDTH=0 or ROCEPS=0 option is specified), so c is not necessarily equal to $E(\hat{c})$.

To compare K empirical ROC curves, first compute the trapezoidal areas. Asymptotic normality of the estimated area follows from U-statistic theory, and a covariance matrix $\mathbf{S}$ can be computed; see DeLong, DeLong, and Clarke-Pearson (1988) for details. A Wald confidence interval for the rth area, $1 \le r \le K$, can be constructed as

$$\hat{c}_r \pm z_{1-\frac{\alpha}{2}} s_{r,r}$$

where $s_{r,r}$ is the rth diagonal of $\mathbf{S}$.

For a contrast of ROC curve areas, $\mathbf{L}\boldsymbol{c}$, the statistic

$$(\hat{\boldsymbol{c}} - \boldsymbol{c})'\mathbf{L}' \left[\mathbf{L}\mathbf{S}\mathbf{L}'\right]^{-1} \mathbf{L}(\hat{\boldsymbol{c}} - \boldsymbol{c})$$

has a chi-square distribution with df=rank($\mathbf{L}\mathbf{S}\mathbf{L}'$). For a row of the contrast, $\boldsymbol{l}'\boldsymbol{c}$,

$$\frac{\boldsymbol{l}'\hat{\boldsymbol{c}} - \boldsymbol{l}'\boldsymbol{c}}{[\boldsymbol{l}'\mathbf{S}\boldsymbol{l}]^{1/2}}$$

has a standard normal distribution. The corresponding confidence interval is

$$\boldsymbol{l}'\hat{\boldsymbol{c}} \pm z_{1-\frac{\alpha}{2}} \left[\boldsymbol{l}'\mathbf{S}\boldsymbol{l}\right]^{1/2}$$

Testing Linear Hypotheses about the Regression Coefficients

Linear hypotheses for $\boldsymbol{\beta}$ are expressed in matrix form as

$$H_0: \mathbf{L}\boldsymbol{\beta} = \boldsymbol{c}$$

where $\mathbf{L}$ is a matrix of coefficients for the linear hypotheses, and $\boldsymbol{c}$ is a vector of constants. The vector of regression coefficients $\boldsymbol{\beta}$ includes slope parameters as well as intercept parameters. The Wald chi-square statistic for testing H_0 is computed as

$$\chi^2_W = (\mathbf{L}\widehat{\boldsymbol{\beta}} - \boldsymbol{c})'[\mathbf{L}\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})\mathbf{L}']^{-1}(\mathbf{L}\widehat{\boldsymbol{\beta}} - \boldsymbol{c})$$

where $\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})$ is the estimated covariance matrix. Under H_0, χ^2_W has an asymptotic chi-square distribution with r degrees of freedom, where r is the rank of $\mathbf{L}$.

Regression Diagnostics

For binary response data, regression diagnostics developed by Pregibon (1981) can be requested by specifying the INFLUENCE option. For diagnostics available with conditional logistic regression, see the section "Regression Diagnostic Details" on page 3357. These diagnostics can also be obtained from the OUTPUT statement.

This section uses the following notation:

r_j, n_j	r_j is the number of event responses out of n_j trials for the jth observation. If *events/trials* syntax is used, r_j is the value of *events* and n_j is the value of *trials*. For *single-trial* syntax, $n_j = 1$, and $r_j = 1$ if the ordered response is 1, and $r_j = 0$ if the ordered response is 2.
w_j	is the weight of the jth observation.
π_j	is the probability of an event response for the jth observation given by $\pi_j = F(\alpha + \boldsymbol{\beta}'\boldsymbol{x}_j)$, where $F(\cdot)$ is the inverse link function defined on page 3323.
$\widehat{\boldsymbol{\beta}}$	is the maximum likelihood estimate (MLE) of $(\alpha, \beta_1, \ldots, \beta_s)'$.
$\widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}})$	is the estimated covariance matrix of $\widehat{\boldsymbol{\beta}}$.
$\hat{p}_j, \hat{q}_j$	$\hat{p}_j$ is the estimate of π_j evaluated at $\widehat{\boldsymbol{\beta}}$, and $\hat{q}_j = 1 - \hat{p}_j$.

Pregibon (1981) suggests using the index plots of several diagnostic statistics to identify influential observations and to quantify the effects on various aspects of the maximum likelihood fit. In an index plot, the diagnostic statistic is plotted against the observation number. In general, the distributions of these diagnostic statistics are not known, so cutoff values cannot be given for determining when the values are large. However, the IPLOTS and INFLUENCE options in the MODEL statement and the PLOTS option in the PROC LOGISTIC statement provide displays of the diagnostic values, allowing visual inspection and comparison of the values across observations. In these plots, if the model is correctly specified and fits all observations well, then no extreme points should appear.

The next five sections give formulas for these diagnostic statistics.

Hat Matrix Diagonal (Leverage)

The diagonal elements of the hat matrix are useful in detecting extreme points in the design space where they tend to have larger values. The jth diagonal element is

$$h_{jj} = \begin{cases} \widetilde{w}_j (1, \boldsymbol{x}_j') \widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}}) (1, \boldsymbol{x}_j')' & \text{Fisher scoring} \\ \widehat{w}_j (1, \boldsymbol{x}_j') \widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}}) (1, \boldsymbol{x}_j')' & \text{Newton-Raphson} \end{cases}$$

where

$$\begin{aligned}\widetilde{w}_j &= \frac{w_j n_j}{\hat{p}_j \hat{q}_j [g'(\hat{p}_j)]^2} \\ \widehat{w}_j &= \widetilde{w}_j + \frac{w_j (r_j - n_j \hat{p}_j)[\hat{p}_j \hat{q}_j g''(\hat{p}_j) + (\hat{q}_j - \hat{p}_j) g'(\hat{p}_j)]}{(\hat{p}_j \hat{q}_j)^2 [g'(\hat{p}_j)]^3}\end{aligned}$$

and $g'(\cdot)$ and $g''(\cdot)$ are the first and second derivatives of the link function $g(\cdot)$, respectively.

For a binary response logit model, the hat matrix diagonal elements are

$$h_{jj} = w_j n_j \hat{p}_j \hat{q}_j (1, \boldsymbol{x}'_j) \widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}}) \begin{pmatrix} 1 \\ \boldsymbol{x}_j \end{pmatrix}$$

If the estimated probability is extreme (less than 0.1 and greater than 0.9, approximately), then the hat diagonal might be greatly reduced in value. Consequently, when an observation has a very large or very small estimated probability, its hat diagonal value is not a good indicator of the observation's distance from the design space (Hosmer and Lemeshow 2000, p. 171).

Pearson Residuals and Deviance Residuals

Pearson and deviance residuals are useful in identifying observations that are not explained well by the model. Pearson residuals are components of the Pearson chi-square statistic and deviance residuals are components of the deviance. The Pearson residual for the jth observation is

$$\chi_j = \frac{\sqrt{w_j}(r_j - n_j \hat{p}_j)}{\sqrt{n_j \hat{p}_j \hat{q}_j}}$$

The Pearson chi-square statistic is the sum of squares of the Pearson residuals.

The deviance residual for the jth observation is

$$d_j = \begin{cases} -\sqrt{-2w_j n_j \log(\hat{q}_j)} & \text{if } r_j = 0 \\ \pm\sqrt{2w_j [r_j \log(\frac{r_j}{n_j \hat{p}_j}) + (n_j - r_j) \log(\frac{n_j - r_j}{n_j \hat{q}_j})]} & \text{if } 0 < r_j < n_j \\ \sqrt{-2w_j n_j \log(\hat{p}_j)} & \text{if } r_j = n_j \end{cases}$$

where the plus (minus) in $\pm$ is used if r_j/n_j is greater (less) than $\hat{p}_j$. The deviance is the sum of squares of the deviance residuals.

DFBETAS

For each parameter estimate, the procedure calculates a DFBETAS diagnostic for each observation. The DFBETAS diagnostic for an observation is the standardized difference in the parameter estimate due to deleting the observation, and it can be used to assess the effect of an individual observation on each estimated parameter of the fitted model. Instead of reestimating the parameter every time an observation is deleted, PROC LOGISTIC uses the one-step estimate. See the section "Predicted

Probability of an Event for Classification" on page 3341. For the jth observation, the DFBETAS are given by

$$\text{DFBETAS}i_j = \mathbf{\Delta}_i \widehat{\boldsymbol{\beta}}^1_j / \hat{\sigma}_i$$

where $i = 0, 1, \ldots, s, \hat{\sigma}_i$ is the standard error of the ith component of $\widehat{\boldsymbol{\beta}}$, and $\mathbf{\Delta}_i \widehat{\boldsymbol{\beta}}^1_j$ is the ith component of the one-step difference

$$\mathbf{\Delta}\widehat{\boldsymbol{\beta}}^1_j = \frac{w_j(r_j - n_j \hat{p}_j)}{1 - h_{jj}} \widehat{\mathbf{V}}(\widehat{\boldsymbol{\beta}}) \begin{pmatrix} 1 \\ \boldsymbol{x}_j \end{pmatrix}$$

$\mathbf{\Delta}\widehat{\boldsymbol{\beta}}^1_j$ is the approximate change $(\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}^1_j)$ in the vector of parameter estimates due to the omission of the jth observation. The DFBETAS are useful in detecting observations that are causing instability in the selected coefficients.

C and CBAR

C and CBAR are confidence interval displacement diagnostics that provide scalar measures of the influence of individual observations on $\widehat{\boldsymbol{\beta}}$. These diagnostics are based on the same idea as the Cook distance in linear regression theory (Cook and Weisberg 1982), but use the one-step estimate. C and CBAR for the jth observation are computed as

$$C_j = \chi^2_j h_{jj} / (1 - h_{jj})^2$$

and

$$\overline{C}_j = \chi^2_j h_{jj} / (1 - h_{jj})$$

respectively.

Typically, to use these statistics, you plot them against an index and look for outliers.

DIFDEV and DIFCHISQ

DIFDEV and DIFCHISQ are diagnostics for detecting ill-fitted observations; in other words, observations that contribute heavily to the disagreement between the data and the predicted values of the fitted model. DIFDEV is the change in the deviance due to deleting an individual observation while DIFCHISQ is the change in the Pearson chi-square statistic for the same deletion. By using the one-step estimate, DIFDEV and DIFCHISQ for the jth observation are computed as

$$\text{DIFDEV} = d^2_j + \overline{C}_j$$

and

$$\text{DIFCHISQ} = \overline{C}_j / h_{jj}$$

Scoring Data Sets

Scoring a data set, which is especially important for predictive modeling, means applying a previously fitted model to a new data set in order to compute the conditional, or *posterior*, probabilities of each response category given the values of the explanatory variables in each observation.

The SCORE statement enables you to score new data sets and output the scored values and, optionally, the corresponding confidence limits into a SAS data set. If the response variable is included in the new data set, then you can request fit statistics for the data, which is especially useful for test or validation data. If the response is binary, you can also create a SAS data set containing the *receiver operating characteristic* (ROC) curve. You can specify multiple SCORE statements in the same invocation of PROC LOGISTIC.

By default, the posterior probabilities are based on implicit prior probabilities that are proportional to the frequencies of the response categories in the *training data* (the data used to fit the model). Explicit prior probabilities should be specified when the sample proportions of the response categories in the training data differ substantially from the operational data to be scored. For example, to detect a rare category, it is common practice to use a training set in which the rare categories are overrepresented; without prior probabilities that reflect the true incidence rate, the predicted posterior probabilities for the rare category will be too high. By specifying the correct priors, the posterior probabilities are adjusted appropriately.

The model fit to the DATA= data set in the PROC LOGISTIC statement is the default model used for the scoring. Alternatively, you can save a model fit in one run of PROC LOGISTIC and use it to score new data in a subsequent run. The OUTMODEL= option in the PROC LOGISTIC statement saves the model information in a SAS data set. Specifying this data set in the INMODEL= option of a new PROC LOGISTIC run will score the DATA= data set in the SCORE statement without refitting the model.

Posterior Probabilities and Confidence Limits

Let F be the inverse link function. That is,

$$F(t) = \begin{cases} \frac{1}{1+\exp(-t)} & \text{logistic} \\ \Phi(t) & \text{normal} \\ 1-\exp(-\exp(t)) & \text{complementary log-log} \end{cases}$$

The first derivative of F is given by

$$F'(t) = \begin{cases} \frac{\exp(-t)}{(1+\exp(-t))^2} & \text{logistic} \\ \phi(t) & \text{normal} \\ \exp(t)\exp(-\exp(t)) & \text{complementary log-log} \end{cases}$$

Suppose there are $k+1$ response categories. Let Y be the response variable with levels $1,\ldots,k+1$. Let $\boldsymbol{x} = (x_0, x_1, \ldots, x_s)'$ be a $(s+1)$-vector of covariates, with $x_0 \equiv 1$. Let $\boldsymbol{\beta}$ be the vector of intercept and slope regression parameters.

Posterior probabilities are given by

$$p(Y=i|\boldsymbol{x}) = \frac{p_o(Y=i|\boldsymbol{x})\frac{\widetilde{p}(Y=i)}{p_o(Y=i)}}{\sum_j p_o(Y=j|\boldsymbol{x})\frac{\widetilde{p}(Y=j)}{p_o(Y=j)}} \quad i=1,\ldots,k+1$$

where the old posterior probabilities $(p_o(Y=i|\boldsymbol{x}), i=1,\ldots,k+1)$ are the conditional probabilities of the response categories given $\boldsymbol{x}$, and the old priors $(p_o(Y=i), i=1,\ldots,k+1)$ are the sample proportions of response categories of the training data. To simplify notation, absorb the old priors into the new priors; that is

$$p(Y=i) = \frac{\widetilde{p}(Y=i)}{p_o(Y=i)} \quad i=1,\ldots,k+1$$

The posterior probabilities are functions of $\boldsymbol{\beta}$ and their estimates are obtained by substituting $\boldsymbol{\beta}$ by its MLE $\widehat{\boldsymbol{\beta}}$. The variances of the estimated posterior probabilities are given by the *delta method* as follows:

$$\text{Var}(\widehat{p}(Y=i|\boldsymbol{x})) = \left[\frac{\partial p(Y=i|\boldsymbol{x})}{\partial \boldsymbol{\beta}}\right]' \text{Var}(\widehat{\boldsymbol{\beta}}) \left[\frac{\partial p(Y=i|\boldsymbol{x})}{\partial \boldsymbol{\beta}}\right]$$

where

$$\frac{\partial p(Y=i|\boldsymbol{x})}{\partial \boldsymbol{\beta}} = \frac{\frac{\partial p_o(Y=i|\boldsymbol{x})}{\partial \boldsymbol{\beta}} p(Y=i)}{\sum_j p_o(Y=j|\boldsymbol{x})p(Y=j)} - \frac{p_o(Y=i|\boldsymbol{x})p(Y=i)\sum_j \frac{\partial p_o(Y=j|\boldsymbol{x})}{\partial \boldsymbol{\beta}} p(Y=j)}{[\sum_j p_o(Y=j|\boldsymbol{x})p(Y=j)]^2}$$

and the old posterior probabilities $p_o(Y=i|\boldsymbol{x})$ are described in the following sections.

A $100(1-\alpha)\%$ confidence interval for $p(Y=i|\boldsymbol{x})$ is

$$\widehat{p}(Y=i|\boldsymbol{x}) \pm z_{1-\alpha/2}\sqrt{\widehat{\text{Var}}(\widehat{p}(Y=i|\boldsymbol{x}))}$$

where z_τ is the upper 100τ percentile of the standard normal distribution.

Binary and Cumulative Response Models

Let $\alpha_1,\ldots,\alpha_k$ be the intercept parameters and let $\boldsymbol{\beta}_s$ be the vector of slope parameters. Denote $\boldsymbol{\beta} = (\alpha_1,\ldots,\alpha_k,\boldsymbol{\beta}_s')'$. Let

$$\eta_i = \eta_i(\boldsymbol{\beta}) = \alpha_i + \boldsymbol{x}'\boldsymbol{\beta}_s, i=1,\ldots,k$$

Estimates of $\eta_1,\ldots,\eta_k$ are obtained by substituting the maximum likelihood estimate $\widehat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}$.

The predicted probabilities of the responses are

$$\widehat{p_o}(Y=i|\boldsymbol{x}) = \widehat{\Pr}(Y=i) = \begin{cases} F(\hat{\eta}_1) & i=1 \\ F(\hat{\eta}_i) - F(\hat{\eta}_{i-1}) & i=2,\ldots,k \\ 1-F(\hat{\eta}_k) & i=k+1 \end{cases}$$

For $i = 1, \ldots, k$, let $\boldsymbol{\delta}_i(\boldsymbol{x})$ be a $(k+1)$ column vector with ith entry equal to 1, $k+1$th entry equal to $\boldsymbol{x}$, and all other entries 0. The derivative of $p_o(Y = i|\boldsymbol{x})$ with respect to $\boldsymbol{\beta}$ are

$$\frac{\partial p_o(Y=i|\boldsymbol{x})}{\partial \boldsymbol{\beta}} = \begin{cases} F'(\alpha_1 + \boldsymbol{x}'\boldsymbol{\beta}_s)\boldsymbol{\delta}_1(\boldsymbol{x}) & i = 1 \\ F'(\alpha_i + \boldsymbol{x}'\boldsymbol{\beta}_s)\boldsymbol{\delta}_i(\boldsymbol{x}) - F'(\alpha_{i-1} + \boldsymbol{x}'\boldsymbol{\beta}_s)\boldsymbol{\delta}_{i-1}(\boldsymbol{x}) & i = 2, \ldots, k \\ -F'(\alpha_k + \boldsymbol{x}'\boldsymbol{\beta}_s)\boldsymbol{\delta}_k(\boldsymbol{x}) & i = k+1 \end{cases}$$

The cumulative posterior probabilities are

$$p(Y \le i|\boldsymbol{x}) = \frac{\sum_{j=1}^{i} p_o(Y=j|\boldsymbol{x})p(Y=j)}{\sum_{j=1}^{k+1} p_o(Y=j|\boldsymbol{x})p(Y=j)} = \sum_{j=1}^{i} p(Y=j|\boldsymbol{x}) \quad i = 1, \ldots, k+1$$

Their derivatives are

$$\frac{\partial p(Y \le i|\boldsymbol{x})}{\partial \boldsymbol{\beta}} = \sum_{j=1}^{i} \frac{\partial p(Y=j|\boldsymbol{x})}{\partial \boldsymbol{\beta}} \quad i = 1, \ldots, k+1$$

In the delta-method equation for the variance, replace $p(Y = \cdot|\boldsymbol{x})$ with $p(Y \le \cdot|\boldsymbol{x})$.

Finally, for the cumulative response model, use

$$\begin{aligned} \widehat{p_o}(Y \le i|\boldsymbol{x}) &= F(\hat{\eta}_i) \quad i = 1, \ldots, k \\ \widehat{p_o}(Y \le k+1|\boldsymbol{x}) &= 1 \\ \frac{\partial p_o(Y \le i|\boldsymbol{x})}{\partial \boldsymbol{\beta}} &= F'(\alpha_i + \boldsymbol{x}'\boldsymbol{\beta}_s)\delta_i(\boldsymbol{x}) \quad i = 1, \ldots, k \\ \frac{\partial p_o(Y \le k+1|\boldsymbol{x})}{\partial \boldsymbol{\beta}} &= 0 \end{aligned}$$

Generalized Logit Model

Consider the last response level (Y=k+1) as the reference. Let $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_k$ be the (intercept and slope) parameter vectors for the first k logits, respectively. Denote $\boldsymbol{\beta} = (\boldsymbol{\beta}_1', \ldots, \boldsymbol{\beta}_k')'$. Let $\eta = (\eta_1, \ldots, \eta_k)'$ with

$$\eta_i = \eta_i(\boldsymbol{\beta}) = \boldsymbol{x}'\boldsymbol{\beta}_i \quad i = 1, \ldots, k$$

Estimates of $\eta_1, \ldots, \eta_k$ are obtained by substituting the maximum likelihood estimate $\widehat{\boldsymbol{\beta}}$ for $\boldsymbol{\beta}$.

The predicted probabilities are

$$\begin{aligned} \widehat{p_o}(Y = k+1|\boldsymbol{x}) \equiv \Pr(Y = k+1|\boldsymbol{x}) &= \frac{1}{1 + \sum_{l=1}^{k} \exp(\hat{\eta}_l)} \\ \widehat{p_o}(Y = i|\boldsymbol{x}) \equiv \Pr(Y = i|\boldsymbol{x}) &= \widehat{p_o}(Y = k+1|\boldsymbol{x}) \exp(\eta_i), i = 1, \ldots, k \end{aligned}$$

The derivative of $p_o(Y = i|\boldsymbol{x})$ with respect to $\boldsymbol{\beta}$ are

$$\begin{aligned}\frac{\partial p_o(Y=i|\boldsymbol{x})}{\partial \boldsymbol{\beta}} &= \frac{\partial \eta}{\partial \boldsymbol{\beta}} \frac{\partial p_o(Y=i|\boldsymbol{x})}{\partial \eta} \\ &= (I_k \otimes \boldsymbol{x}) \left(\frac{\partial p_o(Y=i|\boldsymbol{x})}{\partial \eta_1}, \cdots, \frac{\partial p_o(Y=i|\boldsymbol{x})}{\partial \eta_k} \right)'\end{aligned}$$

where

$$\frac{\partial p_o(Y=i|\boldsymbol{x})}{\partial \eta_j} = \begin{cases} p_o(Y=i|\boldsymbol{x})(1-p_o(Y=i|\boldsymbol{x})) & j=i \\ -p_o(Y=i|\boldsymbol{x})p_o(Y=j|\boldsymbol{x}) & \text{otherwise} \end{cases}$$

Special Case of Binary Response Model with No Priors

Let $\boldsymbol{\beta}$ be the vector of regression parameters. Let

$$\eta = \eta(\boldsymbol{\beta}) = \boldsymbol{x}'\boldsymbol{\beta}$$

The variance of $\hat{\eta}$ is given by

$$\text{Var}(\hat{\eta}) = \boldsymbol{x}'\text{Var}(\widehat{\boldsymbol{\beta}})\boldsymbol{x}$$

A $100(1-\alpha)$ percent confidence interval for η is

$$\hat{\eta} \pm z_{1-\alpha/2}\sqrt{\widehat{\text{Var}}(\hat{\eta})}$$

Estimates of $p_o(Y = 1|\boldsymbol{x})$ and confidence intervals for the $p_o(Y = 1|\boldsymbol{x})$ are obtained by back-transforming $\hat{\eta}$ and the confidence intervals for η, respectively. That is,

$$\widehat{p_o}(Y = 1|\boldsymbol{x}) = F(\hat{\eta})$$

and the confidence intervals are

$$F\left(\hat{\eta} \pm z_{1-\alpha/2}\sqrt{\widehat{\text{Var}}(\hat{\eta})}\right)$$

Conditional Logistic Regression

The method of maximum likelihood described in the preceding sections relies on large-sample asymptotic normality for the validity of estimates and especially of their standard errors. When you do not have a large sample size compared to the number of parameters, this approach might be inappropriate and might result in biased inferences. This situation typically arises when your data are stratified and you fit intercepts to each stratum so that the number of parameters is of the same order as the sample size. For example, in a 1:1 matched pairs study with n pairs and p covariates, you would estimate $n-1$ intercept parameters and p slope parameters. Taking the stratification into account by "conditioning out" (and not estimating) the stratum-specific intercepts gives consistent and asymptotically normal MLEs for the slope coefficients. See Breslow and Day (1980) and Stokes, Davis, and Koch (2000) for more information. If your nuisance parameters are not just stratum-specific intercepts, you can perform an exact conditional logistic regression .

Computational Details

For each stratum h, $h = 1, \ldots, H$, number the observations as $i = 1, \ldots, n_h$ so that hi indexes the ith observation in the hth stratum. Denote the p covariates for observation hi as $\boldsymbol{x}_{hi}$ and its binary response as y_{hi}, and let $\boldsymbol{y} = (y_{11}, \ldots, y_{1n_1}, \ldots, y_{H1}, \ldots, y_{Hn_H})'$, $\mathbf{X}_h = (\boldsymbol{x}_{h1} \ldots \boldsymbol{x}_{hn_h})'$, and $\mathbf{X} = (\mathbf{X}_1' \ldots \mathbf{X}_H')'$. Let the dummy variables $z_h, h = 1, \ldots, H$, be indicator functions for the strata ($z_h = 1$ if the observation is in stratum h), and denote $\boldsymbol{z}_{hi} = (z_1, \ldots, z_H)$ for observation hi, $\mathbf{Z}_h = (\boldsymbol{z}_{h1} \ldots \boldsymbol{z}_{hn_h})'$, and $\mathbf{Z} = (\mathbf{Z}_1' \ldots \mathbf{Z}_H')'$. Denote $\mathbf{X}^* = (\mathbf{Z}|\mathbf{X})$ and $\boldsymbol{x}^*_{hi} = (\boldsymbol{z}'_{hi}|\boldsymbol{x}'_{hi})'$. Arrange the observations in each stratum h so that $y_{hi} = 1$ for $i = 1, \ldots, m_h$, and $y_{hi} = 0$ for $i = m_{h+1}, \ldots, n_h$. Suppose all observations have unit frequency.

Consider the binary logistic regression model on page 3257 written as

$$\text{logit}(\boldsymbol{\pi}) = \mathbf{X}^* \boldsymbol{\theta}$$

where the parameter vector $\boldsymbol{\theta} = (\boldsymbol{\alpha}', \boldsymbol{\beta}')'$ consists of $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_H)'$, α_h is the intercept for stratum $h, h = 1, \ldots, H$, and $\boldsymbol{\beta}$ is the parameter vector for the p covariates.

From the section "Determining Observations for Likelihood Contributions" on page 3324, you can write the likelihood contribution of observation $hi, i = 1, \ldots, n_h, h = 1, \ldots, H$, as

$$L_{hi}(\boldsymbol{\theta}) = \frac{e^{y_{hi} \boldsymbol{x}^{*\prime}_{hi} \boldsymbol{\theta}}}{1 + e^{\boldsymbol{x}^{*\prime}_{hi} \boldsymbol{\theta}}}$$

where $y_{hi} = 1$ when the response takes Ordered Value 1, and $y_{hi} = 0$ otherwise.

The full likelihood is

$$L(\boldsymbol{\theta}) = \prod_{h=1}^{H} \prod_{i=1}^{n_h} L_{hi}(\boldsymbol{\theta}) = \frac{e^{\boldsymbol{y}' \mathbf{X}^* \boldsymbol{\theta}}}{\prod_{h=1}^{H} \prod_{i=1}^{n_h} \left(1 + e^{\boldsymbol{x}^{*\prime}_{hi} \boldsymbol{\theta}}\right)}$$

Unconditional likelihood inference is based on maximizing this likelihood function.

When your nuisance parameters are the stratum-specific intercepts $(\alpha_1, \ldots, \alpha_H)'$, and the slopes $\boldsymbol{\beta}$ are your parameters of interest, "conditioning out" the nuisance parameters produces the conditional likelihood (Lachin 2000)

$$L(\boldsymbol{\beta}) = \prod_{h=1}^{H} L_h(\boldsymbol{\beta}) = \prod_{h=1}^{H} \frac{\prod_{i=1}^{m_h} \exp(\boldsymbol{x}'_{hi} \boldsymbol{\beta})}{\sum \prod_{j=j_1}^{j_{m_h}} \exp(\boldsymbol{x}'_{hj} \boldsymbol{\beta})}$$

where the summation is over all $\binom{n_h}{m_h}$ subsets $\{j_1, \ldots, j_{m_h}\}$ of m_h observations chosen from the n_h observations in stratum h. Note that the nuisance parameters have been factored out of this equation.

For conditional asymptotic inference, maximum likelihood estimates $\widehat{\boldsymbol{\beta}}$ of the regression parameters are obtained by maximizing the conditional likelihood, and asymptotic results are applied to the conditional likelihood function and the maximum likelihood estimators. A relatively fast method of computing this conditional likelihood and its derivatives is given by Gail, Lubin, and Rubinstein (1981) and Howard (1972). The default optimization techniques, which are the same as those implemented by the NLP procedure in SAS/OR software, are as follows:

- Newton-Raphson with ridging when the number of parameters $p < 40$
- quasi-Newton when $40 \leq p < 400$
- conjugate gradient when $p \geq 400$

Sometimes the log likelihood converges but the estimates diverge. This condition is flagged by having inordinately large standard errors for some of your parameter estimates, and can be monitored by specifying the ITPRINT option. Unfortunately, broad existence criteria such as those discussed in the section "Existence of Maximum Likelihood Estimates" on page 3327 do not exist for this model. It might be possible to circumvent such a problem by standardizing your independent variables before fitting the model.

Regression Diagnostic Details

Diagnostics are used to indicate observations that might have undue influence on the model fit or that might be outliers. Further investigation should be performed before removing such an observation from the data set.

The derivations in this section use an augmentation method described by Storer and Crowley (1985), which provides an estimate of the "one-step" DFBETAS estimates advocated by Pregibon (1984). The method also provides estimates of conditional stratum-specific predicted values, residuals, and leverage for each observation.

Following Storer and Crowley (1985), the log-likelihood contribution can be written as

$$\begin{aligned} l_h &= \log(L_h) = \boldsymbol{y}_h'\boldsymbol{\gamma}_h - a(\boldsymbol{\gamma}_h) \quad \text{where} \\ a(\boldsymbol{\gamma}_h) &= \log\left[\sum \prod_{j=j_1}^{j_{m_h}} \exp(\boldsymbol{\gamma}_{hj})\right] \end{aligned}$$

and the h subscript on matrices indicates the submatrix for the stratum, $\boldsymbol{\gamma}_h = (\gamma_{h1}, \ldots, \gamma_{hn_h})'$, and $\gamma_{hi} = \boldsymbol{x}_{hi}'\boldsymbol{\beta}$. Then the gradient and information matrix are

$$\begin{aligned} \boldsymbol{g}(\boldsymbol{\beta}) &= \left\{\frac{\partial l_h}{\partial \boldsymbol{\beta}}\right\}_{h=1}^{H} = \mathbf{X}'(\boldsymbol{y} - \boldsymbol{\pi}) \\ \boldsymbol{\Lambda}(\boldsymbol{\beta}) &= \left\{\frac{\partial^2 l_h}{\partial \boldsymbol{\beta}^2}\right\}_{h=1}^{H} = \mathbf{X}'\text{diag}(\mathbf{U}_1, \ldots, \mathbf{U}_H)\mathbf{X} \end{aligned}$$

where

$$\begin{aligned}
\pi_{hi} &= \frac{\partial a(\boldsymbol{\gamma}_h)}{\partial \gamma_{hi}} = \frac{\sum_{j(i)} \prod_{j=j_1}^{j_{m_h}} \exp(\gamma_{hj})}{\sum \prod_{j=j_1}^{j_{m_h}} \exp(\gamma_{hj})} \\
\boldsymbol{\pi}_h &= (\pi_{h1}, \ldots, \pi_{hn_h}) \\
\mathbf{U}_h &= \frac{\partial^2 a(\boldsymbol{\gamma}_h)}{\partial \boldsymbol{\gamma}_h^2} = \left\{ \frac{\partial^2 \boldsymbol{a}(\boldsymbol{\gamma}_h)}{\partial \gamma_{hi} \partial \gamma_{hj}} \right\} = \{a_{ij}\} \\
a_{ij} &= \frac{\sum_{k(i,j)} \prod_{k=k_1}^{k_{m_h}} \exp(\gamma_{hk})}{\sum \prod_{k=k_1}^{k_{m_h}} \exp(\gamma_{hk})} - \frac{\partial a(\boldsymbol{\gamma}_h)}{\partial \gamma_{hi}} \frac{\partial a(\boldsymbol{\gamma}_h)}{\partial \gamma_{hj}} = \pi_{hij} - \pi_{hi}\pi_{hj}
\end{aligned}$$

and where π_{hi} is the conditional stratum-specific probability that subject i in stratum h is a case, the summation on $j(i)$ is over all subsets from $\{1, \ldots, n_h\}$ of size m_h that contain the index i, and the summation on $k(i, j)$ is over all subsets from $\{1, \ldots, n_h\}$ of size m_h that contain the indices i and j.

To produce the true one-step estimate $\boldsymbol{\beta}_{hi}^1$, start at the MLE $\widehat{\boldsymbol{\beta}}$, delete the hith observation, and use this reduced data set to compute the next Newton-Raphson step. Note that if there is only one event or one nonevent in a stratum, deletion of that single observation is equivalent to deletion of the entire stratum. The augmentation method does not take this into account.

The augmented model is

$$\text{logit}(\Pr(y_{hi} = 1 | \boldsymbol{x}_{hi})) = \boldsymbol{x}'_{hi}\boldsymbol{\beta} + \boldsymbol{z}'_{hi}\gamma$$

where $\boldsymbol{z}_{hi} = (0, \ldots, 0, 1, 0, \ldots, 0)'$ has a 1 in the hith coordinate, and use $\boldsymbol{\beta}^0 = (\widehat{\boldsymbol{\beta}}', 0)'$ as the initial estimate for $(\boldsymbol{\beta}', \gamma)'$. The gradient and information matrix before the step are

$$\begin{aligned}
\boldsymbol{g}(\boldsymbol{\beta}^0) &= \begin{bmatrix} \mathbf{X}' \\ \boldsymbol{z}'_{hi} \end{bmatrix} (\boldsymbol{y} - \boldsymbol{\pi}) = \begin{bmatrix} \mathbf{0} \\ y_{hi} - \pi_{hi} \end{bmatrix} \\
\boldsymbol{\Lambda}(\boldsymbol{\beta}^0) &= \begin{bmatrix} \mathbf{X}' \\ \boldsymbol{z}'_{hi} \end{bmatrix} \mathbf{U} [\mathbf{X} \quad \boldsymbol{z}_{hi}] = \begin{bmatrix} \boldsymbol{\Lambda}(\boldsymbol{\beta}) & \mathbf{X}'\mathbf{U}\boldsymbol{z}_{hi} \\ \boldsymbol{z}'_{hi}\mathbf{U}\mathbf{X} & \boldsymbol{z}'_{hi}\mathbf{U}\boldsymbol{z}_{hi} \end{bmatrix}
\end{aligned}$$

Inserting the $\boldsymbol{\beta}^0$ and $(\mathbf{X}', \boldsymbol{z}'_{hi})'$ into the Gail, Lubin, and Rubinstein (1981) algorithm provides the appropriate estimates of $\boldsymbol{g}(\boldsymbol{\beta}^0)$ and $\boldsymbol{\Lambda}(\boldsymbol{\beta}^0)$. Indicate these estimates with $\widehat{\boldsymbol{\pi}} = \boldsymbol{\pi}(\widehat{\boldsymbol{\beta}})$, $\widehat{\mathbf{U}} = \mathbf{U}(\widehat{\boldsymbol{\beta}})$, $\widehat{\boldsymbol{g}}$, and $\widehat{\boldsymbol{\Lambda}}$.

DFBETA is computed from the information matrix as

$$\begin{aligned}
\Delta_{hi}\boldsymbol{\beta} &= \boldsymbol{\beta}^0 - \boldsymbol{\beta}_{hi}^1 \\
&= -\widehat{\boldsymbol{\Lambda}}^{-1}(\boldsymbol{\beta}^0)\widehat{\boldsymbol{g}}(\boldsymbol{\beta}^0) \\
&= -\widehat{\boldsymbol{\Lambda}}^{-1}(\widehat{\boldsymbol{\beta}})(\mathbf{X}'\widehat{\mathbf{U}}\boldsymbol{z}_{hi})\mathbf{M}^{-1}\boldsymbol{z}'_{hi}(\boldsymbol{y} - \widehat{\boldsymbol{\pi}})
\end{aligned}$$

where

$$\mathbf{M} = (\boldsymbol{z}'_{hi}\widehat{\mathbf{U}}\boldsymbol{z}_{hi}) - (\boldsymbol{z}'_{hi}\widehat{\mathbf{U}}\mathbf{X})\widehat{\boldsymbol{\Lambda}}^{-1}(\widehat{\boldsymbol{\beta}})(\mathbf{X}'\widehat{\mathbf{U}}\boldsymbol{z}_{hi})$$

For each observation in the data set, a DFBETA statistic is computed for each parameter β_j, $1 \leq j \leq p$, and standardized by the standard error of β_j from the full data set to produce the estimate of DFBETAS.

The estimated residuals $e_{hi} = y_{hi} - \widehat{\pi}_{hi}$ are obtained from $\widehat{\boldsymbol{g}}(\boldsymbol{\beta}^0)$, and the weights, or predicted probabilities, are then $\widehat{\pi}_{hi} = y_{hi} - e_{hi}$. The residuals are standardized and reported as (estimated) Pearson residuals:

$$\frac{r_{hi} - n_{hi}\widehat{\pi}_{hi}}{\sqrt{n_{hi}\widehat{\pi}_{hi}(1-\widehat{\pi}_{hi})}}$$

where r_{hi} is the number of events in the observation and n_{hi} is the number of trials.

The estimated leverage is defined as

$$h_{hi} = \frac{\text{trace}\{(\boldsymbol{z}'_{hi}\widehat{\mathbf{U}}\mathbf{X})\widehat{\boldsymbol{\Lambda}}^{-1}(\widehat{\boldsymbol{\beta}})(\mathbf{X}'\widehat{\mathbf{U}}\boldsymbol{z}_{hi})\}}{\text{trace}\{\boldsymbol{z}'_{hi}\widehat{\mathbf{U}}\boldsymbol{z}_{hi}\}}$$

This definition of leverage produces different values from those defined by Pregibon (1984), Moolgavkar, Lustbader, and Venzon (1985), and Hosmer and Lemeshow (2000); however, it has the advantage that no extra computations beyond those for the DFBETAS are required.

For events/trials MODEL syntax, treat each observation as two observations (the first for the nonevents and the second for the events) with frequencies $f_{h,2i-1} = n_{hi} - r_{hi}$ and $f_{h,2i} = r_{hi}$, and augment the model with a matrix $\mathbf{Z}_{hi} = [z_{h,2i-1}\boldsymbol{z}_{h,2i}]$ instead of a single $\boldsymbol{z}_{hi}$ vector. Writing $\gamma_{hi} = \boldsymbol{x}'_{hi}\boldsymbol{\beta}\, f_{hi}$ in the preceding section results in the following gradient and information matrix.

$$\begin{aligned}
\boldsymbol{g}(\boldsymbol{\beta}^0) &= \begin{bmatrix} \mathbf{0} \\ f_{h,2i-1}(y_{h,2i-1} - \pi_{h,2i-1}) \\ f_{h,2i}(y_{h,2i} - \pi_{h,2i}) \end{bmatrix} \\
\boldsymbol{\Lambda}(\boldsymbol{\beta}^0) &= \begin{bmatrix} \boldsymbol{\Lambda}(\boldsymbol{\beta}) & \mathbf{X}'\text{diag}(\boldsymbol{f})\mathbf{U}\text{diag}(\boldsymbol{f})\mathbf{Z}_{hi} \\ \mathbf{Z}'_{hi}\text{diag}(\boldsymbol{f})\mathbf{U}\text{diag}(\boldsymbol{f})\mathbf{X} & \mathbf{Z}'_{hi}\text{diag}(\boldsymbol{f})\mathbf{U}\text{diag}(\boldsymbol{f})\mathbf{Z}_{hi} \end{bmatrix}
\end{aligned}$$

The predicted probabilities are then $\widehat{\pi}_{hi} = y_{h,2i} - e_{h,2i}/r_{h,2i}$, while the leverage and the DFBETAS are produced from $\boldsymbol{\Lambda}(\boldsymbol{\beta}^0)$ in a fashion similar to that for the preceding single-trial equations.

Exact Conditional Logistic Regression

The theory of exact conditional logistic regression analysis was originally laid out by Cox (1970), and the computational methods employed in PROC LOGISTIC are described in Hirji, Mehta, and Patel (1987), Hirji (1992), and Mehta, Patel, and Senchaudhuri (1992). Other useful references for the derivations include Cox and Snell (1989), Agresti (1990), and Mehta and Patel (1995).

Exact conditional inference is based on generating the conditional distribution for the sufficient statistics of the parameters of interest. This distribution is called the *permutation* or *exact conditional* distribution. Using the notation in the section "Computational Details" on page 3356, follow

Mehta and Patel (1995) and first note that the sufficient statistics $\mathbf{T} = (T_1, \ldots, T_p)$ for the parameter vector of intercepts and slopes, $\boldsymbol{\beta}$, are

$$T_j = \sum_{i=1}^{n} y_i x_{ij}, \quad j = 1, \ldots, p$$

Denote a vector of observable sufficient statistics as $\boldsymbol{t} = (t_1, \ldots, t_p)'$.

The probability density function (pdf) for $\mathbf{T}$ can be created by summing over all binary sequences $\boldsymbol{y}$ that generate an observable $\boldsymbol{t}$ and letting $C(\boldsymbol{t}) = ||\{\boldsymbol{y} : \boldsymbol{y}'\mathbf{X} = \boldsymbol{t}'\}||$ denote the number of sequences $\boldsymbol{y}$ that generate $\boldsymbol{t}$

$$\Pr(\mathbf{T} = \boldsymbol{t}) = \frac{C(\boldsymbol{t}) \exp(\boldsymbol{t}'\boldsymbol{\beta})}{\prod_{i=1}^{n} [1 + \exp(\boldsymbol{x}_i'\boldsymbol{\beta})]}$$

In order to condition out the stratum parameters, partition the parameter vector $\boldsymbol{\beta} = (\boldsymbol{\beta}_{\mathrm{N}}', \boldsymbol{\beta}_{\mathrm{I}}')'$, where $\boldsymbol{\beta}_{\mathrm{N}}$ is a $p_{\mathrm{N}} \times 1$ vector of the nuisance parameters, and $\boldsymbol{\beta}_{\mathrm{I}}$ is the parameter vector for the remaining $p_{\mathrm{I}} = p - p_{\mathrm{N}}$ parameters of interest. Likewise, partition $\mathbf{X}$ into $\mathbf{X}_{\mathrm{N}}$ and $\mathbf{X}_{\mathrm{I}}$, $\mathbf{T}$ into $\mathbf{T}_{\mathrm{N}}$ and $\mathbf{T}_{\mathrm{I}}$, and $\boldsymbol{t}$ into $\boldsymbol{t}_{\mathrm{N}}$ and $\boldsymbol{t}_{\mathrm{I}}$. The nuisance parameters can be removed from the analysis by conditioning on their sufficient statistics to create the conditional likelihood of $\mathbf{T}_{\mathrm{I}}$ given $\mathbf{T}_{\mathrm{N}} = \boldsymbol{t}_{\mathrm{N}}$,

$$\begin{aligned} \Pr(\mathbf{T}_{\mathrm{I}} = \boldsymbol{t}_{\mathrm{I}} | \mathbf{T}_{\mathrm{N}} = \boldsymbol{t}_{\mathrm{N}}) &= \frac{\Pr(\mathbf{T} = \boldsymbol{t})}{\Pr(\mathbf{T}_{\mathrm{N}} = \boldsymbol{t}_{\mathrm{N}})} \\ = f_{\boldsymbol{\beta}_{\mathrm{I}}}(\boldsymbol{t}_{\mathrm{I}} | \boldsymbol{t}_{\mathrm{N}}) &= \frac{C(\boldsymbol{t}_{\mathrm{N}}, \boldsymbol{t}_{\mathrm{I}}) \exp(\boldsymbol{t}_{\mathrm{I}}'\boldsymbol{\beta}_{\mathrm{I}})}{\sum_u C(\boldsymbol{t}_{\mathrm{N}}, \boldsymbol{u}) \exp(\boldsymbol{u}'\boldsymbol{\beta}_{\mathrm{I}})} \end{aligned}$$

where $C(\boldsymbol{t}_{\mathrm{N}}, \boldsymbol{u})$ is the number of vectors $\boldsymbol{y}$ such that $\boldsymbol{y}'\mathbf{X}_{\mathrm{N}} = \boldsymbol{t}_{\mathrm{N}}$ and $\boldsymbol{y}'\mathbf{X}_{\mathrm{I}} = \boldsymbol{u}$. Note that the nuisance parameters have factored out of this equation, and that $C(\boldsymbol{t}_{\mathrm{N}}, \boldsymbol{t}_{\mathrm{I}})$ is a constant.

The goal of the exact conditional analysis is to determine how likely the observed response $\boldsymbol{y}_0$ is with respect to all 2^n possible responses $\boldsymbol{y} = (y_1, \ldots, y_n)'$. One way to proceed is to generate every $\boldsymbol{y}$ vector for which $\boldsymbol{y}'\mathbf{X}_{\mathrm{N}} = \boldsymbol{t}_{\mathrm{N}}$, and count the number of vectors $\boldsymbol{y}$ for which $\boldsymbol{y}'\mathbf{X}_{\mathrm{I}}$ is equal to each unique $\boldsymbol{t}_{\mathrm{I}}$. Generating the conditional distribution from complete enumeration of the joint distribution is conceptually simple; however, this method becomes computationally infeasible very quickly. For example, if you had only 30 observations, you would have to scan through 2^{30} different $\boldsymbol{y}$ vectors.

Several algorithms are available in PROC LOGISTIC to generate the exact distribution. All of the algorithms are based on the following observation. Given any $\boldsymbol{y} = (y_1, \ldots, y_n)'$ and a design $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)'$, let $\boldsymbol{y}_{(i)} = (y_1, \ldots, y_i)'$ and $\mathbf{X}_{(i)} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_i)'$ be the first i rows of each matrix. Write the sufficient statistic based on these i rows as $\boldsymbol{t}_{(i)}' = \boldsymbol{y}_{(i)}'\mathbf{X}_{(i)}$. A recursion relation results: $\boldsymbol{t}_{(i+1)} = \boldsymbol{t}_{(i)} + y_{i+1}\boldsymbol{x}_{i+1}$.

The following methods are available.

- The *multivariate shift algorithm* developed by Hirji, Mehta, and Patel (1987), which steps through the recursion relation by adding one observation at a time and building an intermediate distribution at each step. If it determines that $\boldsymbol{t}_{(i)}$ for the nuisance parameters could eventually equal $\boldsymbol{t}$, then $\boldsymbol{t}_{(i)}$ is added to the intermediate distribution.

- An extension of the multivariate shift algorithm to generalized logit models by Hirji (1992). Since the generalized logit model fits a new set of parameters to each logit, the number of parameters in the model can easily get too large for this algorithm to handle. Note for these models that the hypothesis tests for each effect are computed across the logit functions, while individual parameters are estimated for each logit function.
- A network algorithm described in Mehta, Patel, and Senchaudhuri (1992), which builds a network for each parameter that you are conditioning out in order to identify feasible y_i for the $\boldsymbol{y}$ vector. These networks are combined and the set of feasible y_i is further reduced, and then the multivariate shift algorithm uses this knowledge to build the exact distribution without adding as many intermediate $\boldsymbol{t}_{(i+1)}$ as the multivariate shift algorithm does.
- A hybrid Monte Carlo and network algorithm described by Mehta, Patel, and Senchaudhuri (2000), which extends their 1992 algorithm by sampling from the combined network to build the exact distribution.

The bulk of the computation time and memory for these algorithms is consumed by the creation of the networks and the exact joint distribution. After the joint distribution for a set of effects is created, the computational effort required to produce hypothesis tests and parameter estimates for any subset of the effects is (relatively) trivial. See the section "Computational Resources for Exact Conditional Logistic Regression" on page 3369 for more computational notes about exact analyses.

NOTE: An alternative to using these exact conditional methods is to perform Firth's bias-reducing penalized likelihood method (see the FIRTH option in the MODEL statement); this method has the advantage of being much faster and less memory intensive than exact algorithms, but it might not converge to a solution.

Hypothesis Tests

Consider testing the null hypothesis $H_0: \boldsymbol{\beta}_{\mathrm{I}} = \mathbf{0}$ against the alternative $H_A: \boldsymbol{\beta}_{\mathrm{I}} \neq \mathbf{0}$, conditional on $\mathbf{T}_{\mathrm{N}} = \boldsymbol{t}_{\mathrm{N}}$. Under the null hypothesis, the test statistic for the *exact probability test* is just $f_{\boldsymbol{\beta}_{\mathrm{I}}=\mathbf{0}}(\boldsymbol{t}_{\mathrm{I}}|\boldsymbol{t}_{\mathrm{N}})$, while the corresponding p-value is the probability of getting a less likely (more extreme) statistic,

$$p(\boldsymbol{t}_{\mathrm{I}}|\boldsymbol{t}_{\mathrm{N}}) = \sum_{u \in \Omega_p} f_0(\boldsymbol{u}|\boldsymbol{t}_{\mathrm{N}})$$

where $\Omega_p = \{\boldsymbol{u}$: there exist $\boldsymbol{y}$ with $\boldsymbol{y}'\mathbf{X}_{\mathrm{I}} = \boldsymbol{u}$, $\boldsymbol{y}'\mathbf{X}_{\mathrm{N}} = \boldsymbol{t}_{\mathrm{N}}$, and $f_0(\boldsymbol{u}|\boldsymbol{t}_{\mathrm{N}}) \leq f_0(\boldsymbol{t}_{\mathrm{I}}|\boldsymbol{t}_{\mathrm{N}})\}$.

For the *exact conditional scores test*, the conditional mean $\boldsymbol{\mu}_{\mathrm{I}}$ and variance matrix $\boldsymbol{\Sigma}_{\mathrm{I}}$ of the $\mathbf{T}_{\mathrm{I}}$ (conditional on $\mathbf{T}_{\mathrm{N}} = \boldsymbol{t}_{\mathrm{N}}$) are calculated, and the score statistic for the observed value,

$$s = (\boldsymbol{t}_{\mathrm{I}} - \boldsymbol{\mu}_{\mathrm{I}})' \boldsymbol{\Sigma}_{\mathrm{I}}^{-1} (\boldsymbol{t}_{\mathrm{I}} - \boldsymbol{\mu}_{\mathrm{I}})$$

is compared to the score for each member of the distribution

$$S(\mathbf{T}_{\mathrm{I}}) = (\mathbf{T}_{\mathrm{I}} - \boldsymbol{\mu}_{\mathrm{I}})' \boldsymbol{\Sigma}_{\mathrm{I}}^{-1} (\mathbf{T}_{\mathrm{I}} - \boldsymbol{\mu}_{\mathrm{I}})$$

The resulting p-value is

$$p(\boldsymbol{t}_{\mathrm{I}}|\boldsymbol{t}_{\mathrm{N}}) = Pr(S \geq s) = \sum_{u \in \Omega_s} f_0(\boldsymbol{u}|\boldsymbol{t}_{\mathrm{N}})$$

where $\Omega_s = \{\boldsymbol{u}$: there exist $\boldsymbol{y}$ with $\boldsymbol{y}'\mathbf{X}_{\mathrm{I}} = \boldsymbol{u}$, $\boldsymbol{y}'\mathbf{X}_{\mathrm{N}} = \boldsymbol{t}_{\mathrm{N}}$, and $S(\boldsymbol{u}) \geq s\}$.

The mid-p statistic, defined as

$$p(\boldsymbol{t}_{\mathrm{I}}|\boldsymbol{t}_{\mathrm{N}}) - \frac{1}{2} f_0(\boldsymbol{t}_{\mathrm{I}}|\boldsymbol{t}_{\mathrm{N}})$$

was proposed by Lancaster (1961) to compensate for the discreteness of a distribution. See Agresti (1992) for more information. However, to allow for more flexibility in handling ties, you can write the mid-p statistic as (based on a suggestion by Lamotte (2002) and generalizing Vollset, Hirji, and Afifi (1991))

$$\sum_{\boldsymbol{u}\in\Omega_<} f_0(\boldsymbol{u}|\boldsymbol{t}_{\mathrm{N}}) + \delta_1 f_0(\boldsymbol{t}_{\mathrm{I}}|\boldsymbol{t}_{\mathrm{N}}) + \delta_2 \sum_{\boldsymbol{u}\in\Omega_=} f_0(\boldsymbol{u}|\boldsymbol{t}_{\mathrm{N}})$$

where, for $i \in \{p, s\}$, $\Omega_<$ is Ω_i using strict inequalities, and $\Omega_=$ is Ω_i using equalities with the added restriction that $\boldsymbol{u} \neq \boldsymbol{t}_{\mathrm{I}}$. Letting $(\delta_1, \delta_2) = (0.5, 1.0)$ yields Lancaster's mid-p.

CAUTION: When the exact distribution has ties and METHOD=NETWORKMC is specified, the Monte Carlo algorithm estimates $p(\boldsymbol{t}|\boldsymbol{t}_{\mathrm{N}})$ with error, and hence it cannot determine precisely which values contribute to the reported p-values. For example, if the exact distribution has densities $\{0.2, 0.2, 0.2, 0.4\}$ and if the observed statistic has probability 0.2, then the exact probability p-value is exactly 0.6. Under Monte Carlo sampling, if the densities after N samples are $\{0.18, 0.21, 0.23, 0.38\}$ and the observed probability is 0.21, then the resulting p-value is 0.39. Therefore, the exact probability test p-value for this example fluctuates between 0.2, 0.4, and 0.6, and the reported p-values are actually lower bounds for the true p-values. If you need more precise values, you can specify the OUTDIST= option, determine appropriate cutoff values for the observed probability and score, and then construct the true p-value estimates from the OUTDIST= data set and display them in the SAS log by using the following statements:

```
data _null_;
   set outdist end=end;
   retain pvalueProb 0 pvalueScore 0;
   if prob < ProbCutOff then pvalueProb+prob;
   if score > ScoreCutOff then pvalueScore+prob;
   if end then put pvalueProb= pvalueScore=;
run;
```

Inference for a Single Parameter

Exact parameter estimates are derived for a single parameter β_i by regarding all the other parameters $\boldsymbol{\beta}_{\mathrm{N}} = (\beta_1, \ldots, \beta_{i-1}, \beta_{i+1}, \ldots, \beta_{p_{\mathrm{N}}+p_{\mathrm{I}}})'$ as nuisance parameters. The appropriate sufficient statistics are $\mathbf{T}_{\mathrm{I}} = T_i$ and $\mathbf{T}_{\mathrm{N}} = (T_1, \ldots, T_{i-1}, T_{i+1}, \ldots, T_{p_{\mathrm{N}}+p_{\mathrm{I}}})'$, with their observed values denoted by the lowercase t. Hence, the conditional pdf used to create the parameter estimate for β_i is

$$f_{\beta_i}(t_i|\boldsymbol{t}_{\mathrm{N}}) = \frac{C(\boldsymbol{t}_{\mathrm{N}}, t_i)\exp(t_i\beta_i)}{\sum_{u\in\Omega} C(\boldsymbol{t}_{\mathrm{N}}, u)\exp(u\beta_i)}$$

for $\Omega = \{u\colon$ there exist $\boldsymbol{y}$ with $T_i = u$ and $\mathbf{T}_{\mathrm{N}} = \boldsymbol{t}_{\mathrm{N}}\}$.

The maximum exact conditional likelihood estimate is the quantity $\widehat{\beta}_i$, which maximizes the conditional pdf. A Newton-Raphson algorithm is used to perform this search. However, if the observed t_i attains either its maximum or minimum value in the exact distribution (that is, either

$t_i = \min\{u : u \in \Omega\}$ or $t_i = \max\{u : u \in \Omega\}$), then the conditional pdf is monotonically increasing in β_i and cannot be maximized. In this case, a median unbiased estimate (Hirji, Tsiatis, and Mehta 1989) $\widehat{\beta}_i$ is produced that satisfies $f_{\widehat{\beta}_i}(t_i|\boldsymbol{t}_{\mathrm{N}}) = 0.5$, and a Newton-Raphson algorithm is used to perform the search.

The standard error of the exact conditional likelihood estimate is just the negative of the inverse of the second derivative of the exact conditional log likelihood (Agresti 2002).

Likelihood ratio tests based on the conditional pdf are used to test the null $H_0\colon \beta_i = 0$ against the alternative $H_A\colon \beta_i > 0$. The critical region for this UMP test consists of the upper tail of values for T_i in the exact distribution. Thus, the one-sided significance level $p_+(t_i;0)$ is

$$p_+(t_i;0) = \sum_{u \geq t_i} f_0(u|\boldsymbol{t}_{\mathrm{N}})$$

Similarly, the one-sided significance level $p_-(t_i;0)$ against $H_A\colon \beta_i < 0$ is

$$p_-(t_i;0) = \sum_{u \leq t_i} f_0(u|\boldsymbol{t}_{\mathrm{N}})$$

The two-sided significance level $p(t_i;0)$ against $H_A\colon \beta_i \neq 0$ is calculated as

$$p(t_i;0) = 2\min[p_-(t_i;0), p_+(t_i;0)]$$

An upper $100(1-2\epsilon)\%$ exact confidence limit for $\widehat{\beta}_i$ corresponding to the observed t_i is the solution $\beta_U(t_i)$ of $\epsilon = p_-(t_i, \beta_U(t_i))$, while the lower exact confidence limit is the solution $\beta_L(t_i)$ of $\epsilon = p_+(t_i, \beta_L(t_i))$. Again, a Newton-Raphson procedure is used to search for the solutions.

Specifying the ONESIDED option displays only one p-value and one confidence interval, because small values of $p_+(t_i;0)$ and $p_-(t_i;0)$ support different alternative hypotheses and only one of these p-values can be less than 0.50.

The mid-p confidence limits are the solutions to $\min\{p_-(t_i, \beta(t_i)), p_+(t_i, \beta(t_i))\} - (1 - \delta_1) f_{\beta(t_i)}(u|\boldsymbol{t}_{\mathrm{N}}) = \epsilon$ for $\epsilon = \alpha/2, 1-\alpha/2$ (Vollset, Hirji, and Afifi 1991). $\delta_1 = 1$ produces the usual exact (or *max-p*) confidence interval, $\delta_1 = 0.5$ yields the mid-p interval, and $\delta_1 = 0$ gives the *min-p* interval. The mean of the endpoints of the max-p and min-p intervals provides the *mean-p* interval as defined by Hirji, Mehta, and Patel (1988).

Estimates and confidence intervals for the odds ratios are produced by exponentiating the estimates and interval endpoints for the parameters.

Notes about Exact *p*-Values

In the "Conditional Exact Tests" table, the exact probability test is not necessarily a sum of tail areas and can be inflated if the distribution is skewed. The more robust exact conditional scores test is a sum of tail areas and is generally preferred over the exact probability test.

The p-value reported for a single parameter in the "Exact Parameter Estimates" table is twice the one-sided tail area of a likelihood ratio test against the null hypothesis of the parameter equaling zero.

Input and Output Data Sets

OUTEST= Output Data Set

The OUTEST= data set contains one observation for each BY group containing the maximum likelihood estimates of the regression coefficients. If you also use the COVOUT option in the PROC LOGISTIC statement, there are additional observations containing the rows of the estimated covariance matrix. If you specify SELECTION=FORWARD, BACKWARD, or STEPWISE, only the estimates of the parameters and covariance matrix for the final model are output to the OUTEST= data set.

Variables in the OUTEST= Data Set

The OUTEST= data set contains the following variables:

- any BY variables specified
- _LINK_, a character variable of length 8 with four possible values: CLOGLOG for the complementary log-log function, LOGIT for the logit function, NORMIT for the probit (alias normit) function, and GLOGIT for the generalized logit function
- _TYPE_, a character variable of length 8 with two possible values: PARMS for parameter estimates or COV for covariance estimates. If an EXACT statement is also specified, then two other values are possible: EPARMMLE for the exact maximum likelihood estimates and EPARMMUE for the exact median unbiased estimates.
- _NAME_, a character variable containing the name of the response variable when _TYPE_=PARMS, EPARMMLE, and EPARMMUE, or the name of a model parameter when _TYPE_=COV
- _STATUS_, a character variable that indicates whether the estimates have converged
- one variable for each intercept parameter
- one variable for each slope parameter and one variable for the offset variable if the OFFSET= option if specified. If an effect is not included in the final model in a model building process, the corresponding parameter estimates and covariances are set to missing values.
- _LNLIKE_, the log likelihood

Parameter Names in the OUTEST= Data Set

If there are only two response categories in the entire data set, the intercept parameter is named Intercept. If there are more than two response categories in the entire data set, the intercept parameters are named Intercept_xxx, where xxx is the value (formatted if a format is applied) of the corresponding response category.

For continuous explanatory variables, the names of the parameters are the same as the corresponding variables. For CLASS variables, the parameter names are obtained by concatenating the corresponding CLASS variable name with the CLASS category; see the PARAM= option in the CLASS statement and the section "CLASS Variable Parameterization" on page 3319 for more details. For interaction and nested effects, the parameter names are created by concatenating the names of each effect.

For the generalized logit model, names of parameters corresponding to each nonreference category contain _xxx as the suffix, where xxx is the value (formatted if a format is applied) of the corresponding nonreference category. For example, suppose the variable Net3 represents the television network (ABC, CBS, and NBC) viewed at a certain time. The following statements fit a generalized logit model with Age and Gender (a CLASS variable with values Female and Male) as explanatory variables:

```
proc logistic;
   class Gender;
   model Net3 = Age Gender / link=glogit;
run;
```

There are two logit functions, one contrasting ABC with NBC and the other contrasting CBS with NBC. For each logit, there are three parameters: an intercept parameter, a slope parameter for Age, and a slope parameter for Gender (since there are only two gender levels and the EFFECT parameterization is used by default). The names of the parameters and their descriptions are as follows:

Intercept_ABC	intercept parameter for the logit contrasting ABC with NBC
Intercept_CBS	intercept parameter for the logit contrasting CBS with NBC
Age_ABC	Age slope parameter for the logit contrasting ABC with NBC
Age_CBS	Age slope parameter for the logit contrasting CBS with NBC
GenderFemale_ABC	Gender=Female slope parameter for the logit contrasting ABC with NBC
GenderFemale_CBS	Gender=Female slope parameter for the logit contrasting CBS with NBC

INEST= Input Data Set

You can specify starting values for the iterative algorithm in the INEST= data set. The INEST= data set has the same structure as the OUTEST= data set but is not required to have all the variables or observations that appear in the OUTEST= data set. A previous OUTEST= data set can be used as, or modified for use as, an INEST= data set.

The INEST= data set must contain the intercept variables (named Intercept for binary response models and Intercept, Intercept_2, Intercept_3, and so forth, for ordinal and nominal response models) and all explanatory variables in the MODEL statement. If BY processing is used, the INEST= data set should also include the BY variables, and there must be one observation for each BY group. If the INEST= data set also contains the _TYPE_ variable, only observations with _TYPE_ value 'PARMS' are used as starting values.

OUT= Output Data Set in the OUTPUT Statement

The OUT= data set in the OUTPUT statement contains all the variables in the input data set along with statistics you request by specifying *keyword=name* options or the PREDPROBS= option in the OUTPUT statement. In addition, if you use the *single-trial* syntax and you request any of the XBETA=, STDXBETA=, PREDICTED=, LCL=, and UCL= options, the OUT= data set contains the automatic variable _LEVEL_. The value of _LEVEL_ identifies the response category upon which the computed values of XBETA=, STDXBETA=, PREDICTED=, LCL=, and UCL= are based.

When there are more than two response levels, only variables named by the XBETA=, STDXBETA=, PREDICTED=, LOWER=, and UPPER= options and the variables given by PREDPROBS=(INDIVIDUAL CUMULATIVE) have their values computed; the other variables have missing values. If you fit a generalized logit model, the cumulative predicted probabilities are not computed.

When there are only two response categories, each input observation produces one observation in the OUT= data set.

If there are more than two response categories and you specify only the PREDPROBS= option, then each input observation produces one observation in the OUT= data set. However, if you fit an ordinal (cumulative) model and specify options other than the PREDPROBS= options, each input observation generates as many output observations as one fewer than the number of response levels, and the predicted probabilities and their confidence limits correspond to the cumulative predicted probabilities. If you fit a generalized logit model and specify options other than the PREDPROBS= options, each input observation generates as many output observations as the number of response categories; the predicted probabilities and their confidence limits correspond to the probabilities of individual response categories.

For observations in which only the response variable is missing, values of the XBETA=, STDXBETA=, PREDICTED=, UPPER=, LOWER=, and the PREDPROBS= options are computed even though these observations do not affect the model fit. This enables, for instance, predicted probabilities to be computed for new observations.

OUT= Output Data Set in a SCORE Statement

The OUT= data set in a SCORE statement contains all the variables in the data set being scored. The data set being scored can be either the input DATA= data set in the PROC LOGISTIC statement or the DATA= data set in the SCORE statement. The DATA= data set in the SCORE statement does not need to contain the response variable.

If the data set being scored contains the response variable, then denote the *normalized* levels (left-justified, formatted values of 16 characters or less) of your response variable Y by $Y_1, \ldots, Y_{k+1}$. For each response level, the OUT= data set also contains the following:

- F_*Y*, the normalized levels of the response variable Y in the data set being scored. If the *events/trials* syntax is used, the F_*Y* variable is not created.
- I_*Y*, the normalized levels that the observations are classified into. Note that an observation is classified into the level with the largest probability. If the *events/trials* syntax is used, the _INTO_ variable is created instead, and it contains the values EVENT and NONEVENT.

- P_Y_i, the posterior probabilities of the normalized response level Y_i
- If the CLM option is specified in the SCORE statement, the OUT= data set also includes the following:
  - LCL_Y_i, the lower $100(1-\alpha)\%$ confidence limits for P_Y_i
  - UCL_Y_i, the upper $100(1-\alpha)\%$ confidence limits for P_Y_i

OUTDIST= Output Data Set

The OUTDIST= data set contains every exact conditional distribution necessary to process the corresponding EXACT statement. For example, the following statements create one distribution for the x1 parameter and another for the x2 parameters, and produce the data set dist shown in Figure 51.11:

```
proc logistic;
   class x2 / param=ref;
   model y=x1 x2;
   exact x1 x2/ outdist=dist;
proc print data=dist;
run;
```

Figure 51.11 OUTDIST= Data Set

Obs	x1	x20	x21	Count	Score	Prob
1	.	0	0	3	5.81151	0.03333
2	.	0	1	15	1.66031	0.16667
3	.	0	2	9	3.12728	0.10000
4	.	1	0	15	1.46523	0.16667
5	.	1	1	18	0.21675	0.20000
6	.	1	2	6	4.58644	0.06667
7	.	2	0	19	1.61869	0.21111
8	.	2	1	2	3.27293	0.02222
9	.	3	0	3	6.27189	0.03333
10	2	.	.	6	3.03030	0.12000
11	3	.	.	12	0.75758	0.24000
12	4	.	.	11	0.00000	0.22000
13	5	.	.	18	0.75758	0.36000
14	6	.	.	3	3.03030	0.06000

The first nine observations in the dist data set contain a exact distribution for the parameters of the x2 effect (hence the values for the x1 parameter are missing), and the remaining five observations are for the x1 parameter. If a joint distribution was created, there would be observations with values for both the x1 and x2 parameters. For CLASS variables, the corresponding parameters in the dist data set are identified by concatenating the variable name with the appropriate classification level.

The data set contains the possible sufficient statistics of the parameters for the effects specified in the EXACT statement, and the Count variable contains the number of different responses that yield these statistics. In particular, there were six possible response vectors $\boldsymbol{y}$ for which the dot product

$\boldsymbol{y'x1}$ was equal to 2, and for which $\boldsymbol{y'x20}$, $\boldsymbol{y'x21}$, and $\boldsymbol{y'1}$ were equal to their actual observed values (displayed in the "Sufficient Statistics" table).

When hypothesis tests are performed on the parameters, the Prob variable contains the probability of obtaining that statistic (which is just the count divided by the total count), and the Score variable contains the score for that statistic.

The OUTDIST= data set can contain a different exact conditional distribution for each specified EXACT statement. For example, consider the following EXACT statements:

```
exact 'O1'   x1    /           outdist=o1;
exact 'OJ12' x1 x2 / jointonly outdist=oj12;
exact 'OA12' x1 x2 / joint     outdist=oa12;
exact 'OE12' x1 x2 / estimate  outdist=oe12;
```

The O1 statement outputs a single exact conditional distribution. The OJ12 statement outputs only the joint distribution for x1 and x2. The OA12 statement outputs three conditional distributions: one for x1, one for x2, and one jointly for x1 and x2. The OE12 statement outputs two conditional distributions: one for x1 and the other for x2. Data set oe12 contains both the x1 and x2 variables; the distribution for x1 has missing values in the x2 column while the distribution for x2 has missing values in the x1 column.

OUTROC= Output Data Set

The OUTROC= data set contains data necessary for producing the ROC curve, and can be created by specifying the OUTROC= option in the MODEL statement or the OUTROC= option in the SCORE statement: It has the following variables:

- any BY variables specified
- _STEP_, the model step number. This variable is not included if model selection is not requested.
- _PROB_, the estimated probability of an event. These estimated probabilities serve as cutpoints for predicting the response. Any observation with an estimated event probability that exceeds or equals _PROB_ is predicted to be an event; otherwise, it is predicted to be a nonevent. Predicted probabilities that are close to each other are grouped together, with the maximum allowable difference between the largest and smallest values less than a constant that is specified by the ROCEPS= option. The smallest estimated probability is used to represent the group.
- _POS_, the number of correctly predicted event responses
- _NEG_, the number of correctly predicted nonevent responses
- _FALPOS_, the number of falsely predicted event responses
- _FALNEG_, the number of falsely predicted nonevent responses
- _SENSIT_, the sensitivity, which is the proportion of event observations that were predicted to have an event response

- _1MSPEC_, one minus specificity, which is the proportion of nonevent observations that were predicted to have an event response

Note that none of these statistics are affected by the bias-correction method discussed in the section "Classification Table" on page 3340. An ROC curve is obtained by plotting _SENSIT_ against _1MSPEC_.

For more information, see the section "Receiver Operating Characteristic Curves" on page 3346.

Computational Resources

The memory needed to fit an unconditional model is approximately $8n(p+2)+24(p+2)^2$ bytes, where p is the number of parameters estimated and n is the number of observations in the data set. For cumulative response models with more than two response levels, a test of the parallel lines assumption requires an additional memory of approximately $4k^2(m+1)^2+24(m+2)^2$ bytes, where k is the number of response levels and m is the number of slope parameters. However, if this additional memory is not available, the procedure skips the test and finishes the other computations. You might need more memory if you use the SELECTION= option for model building.

The data that consist of relevant variables (including the design variables for model effects) and observations for fitting the model are stored in a temporary utility file. If sufficient memory is available, such data will also be kept in memory; otherwise, the data are reread from the utility file for each evaluation of the likelihood function and its derivatives, with the resulting execution time of the procedure substantially increased. Specifying the MULTIPASS option in the MODEL statement avoids creating this utility file and also does not store the data in memory; instead, the DATA= data set is reread when needed. This saves approximately $8n(p+2)$ bytes of memory but increases the execution time.

If a conditional logistic regression is performed, then approximately $4(m^2+m+4)\max_h(m_h)+(8s_H+36)H+12s_H$ additional bytes of memory are needed, where m_h is the number of events in stratum h, H is the total number of strata, and s_H is the number of variables used to define the strata. If the CHECKDEPENDENCY=ALL option is specified in the STRATA statement, then an extra $4(m+H)(m+H+1)$ bytes are required, and the resulting execution time of the procedure might be substantially increased.

Computational Resources for Exact Conditional Logistic Regression

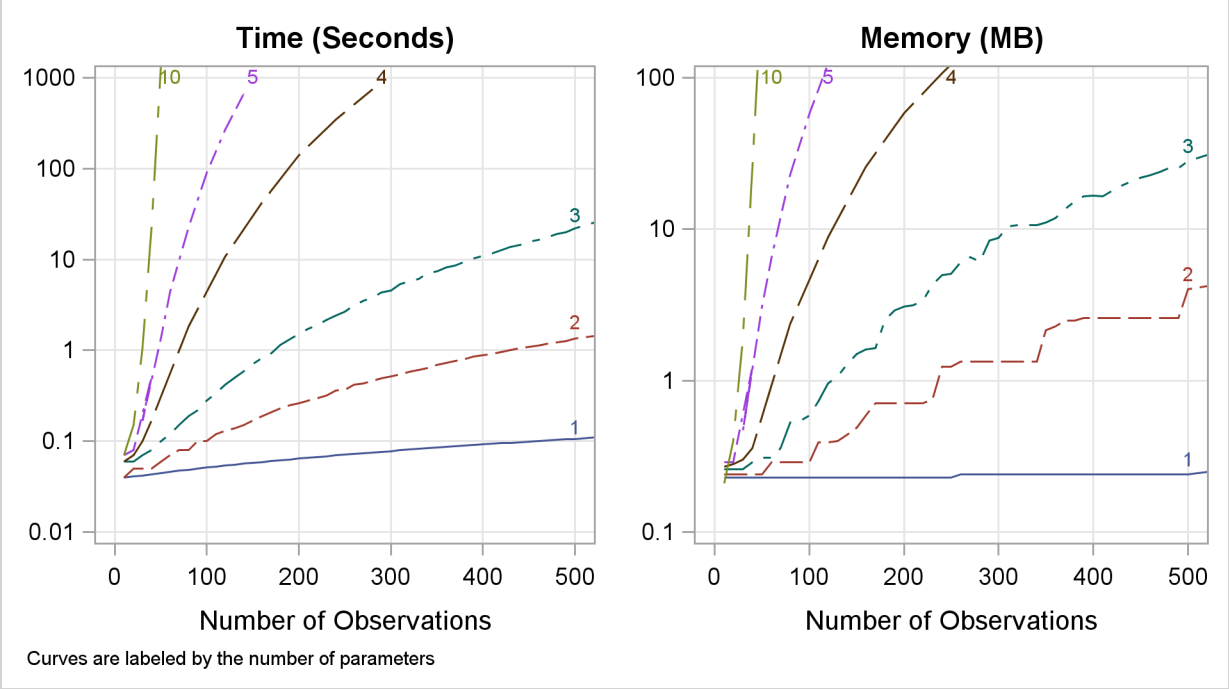
Many problems require a prohibitive amount of time and memory for exact computations, depending on the speed and memory available on your computer. For such problems, consider whether exact methods are really necessary. Stokes, Davis, and Koch (2000) suggest looking at exact p-values when the sample size is small and the approximate p-values from the unconditional analysis are less than 0.10, and they provide *rules of thumb* for determining when various models are valid.

A formula does not exist that can predict the amount of time and memory necessary to generate the exact conditional distributions for a particular problem. The time and memory required depends on several factors, including the total sample size, the number of parameters of interest, the number

of nuisance parameters, and the order in which the parameters are processed. To provide a feel for how these factors affect performance, 19 data sets containing Nobs $\in \{10, \ldots, 500\}$ observations consisting of up to 10 independent uniform binary covariates (X1,...,*XN*) and a binary response variable (Y), are generated, and the following statements create exact conditional distributions for X1 conditional on the other covariates by using the default METHOD=NETWORK. Figure 51.12 displays results obtained on a 400Mhz PC with 768MB RAM running Microsoft Windows NT.

```
data one;
   do obs=1 to HalfNobs;
      do Y=0 to 1;
         X1=round(ranuni(0));
         ...
         XN=round(ranuni(0));
         output;
      end;
   end;
options fullstimer;
proc logistic exactonly exactoptions(method=network maxtime=1200);
   class X1...XN / param=ref;
   model Y=X1...XN;
   exact X1 / outdist=dist;
run;
```

Figure 51.12 Mean Time and Memory Required



At any time while PROC LOGISTIC is deriving the distributions, you can terminate the computations by pressing the system interrupt key sequence (see the SAS Companion for your system) and

choosing to stop computations. If you run out of memory, see the SAS Companion for your system to see how to allocate more.

You can use the EXACTOPTIONS option MAXTIME= to limit the total amount of time PROC LOGISTIC uses to derive all of the exact distributions. If PROC LOGISTIC does not finish within that time, the procedure terminates.

Calculation of frequencies are performed in the log scale by default. This reduces the need to check for excessively large frequencies but can be slower than not scaling. You can turn off the log scaling by specifying the NOLOGSCALE option in the MODEL statement. If a frequency in the exact distribution is larger than the largest integer that can be held in double precision, a warning is printed to the SAS log. But since inaccuracies due to adding small numbers to these large frequencies might have little or no effect on the statistics, the exact computations continue.

You can monitor the progress of the procedure by submitting your program with the EXACTOPTIONS option STATUSTIME=. If the procedure is too slow, you can try another method by specifying the EXACTOPTIONS option METHOD=, you can try reordering the variables in the MODEL statement (note that CLASS variables are always processed before continuous covariates), or you can try reparameterizing your classification variables as in the following statement:

```
class class-variables / param=ref ref=first order=freq;
```

Displayed Output

If you use the NOPRINT option in the PROC LOGISTIC statement, the procedure does not display any output. Otherwise, the tables displayed by the LOGISTIC procedure are discussed in the following section in the order in which they appear in the output. Note that some of the tables appear only in conjunction with certain options or statements; see the section "ODS Table Names" on page 3377 for details.

Table Summary

Model Information and the Number of Observations

See the section "Missing Values" on page 3318 for information about missing-value handling, and the sections "FREQ Statement" on page 3287 and "WEIGHT Statement" on page 3317 for information about valid frequencies and weights.

Response Profile

Displays the Ordered Value assigned to each response level. See the section "Response Level Ordering" on page 3318 for details.

Class Level Information

Displays the design values for each CLASS explanatory variable. See the section "CLASS Variable Parameterization" on page 3319 for details.

Simple Statistics Tables

The following tables are displayed if you specify the SIMPLE option in the PROC LOGISTIC statement.

- **Descriptive Statistics for Continuous Explanatory Variables**
- **Frequency Distribution of Class Variables**
- **Weight Distribution of Class Variables**
  Displays if you also specify a WEIGHT statement.

Strata Tables for (Exact) Conditional Logistic Regression

The following tables are displayed if you specify a STRATA statement.

- **Strata Summary**
  Shows the pattern of the number of events and the number of nonevents in a stratum. See the section "STRATA Statement" on page 3314 for more information.
- **Strata Information**
  Displays if you specify the INFO option in a STRATA statement.

Maximum Likelihood Iteration History

Displays if you specify the ITPRINT option in the MODEL statement. See the sections "Iterative Algorithms for Model Fitting" on page 3324, "Convergence Criteria" on page 3327, and "Existence of Maximum Likelihood Estimates" on page 3327 for details.

Deviance and Pearson Goodness-of-Fit Statistics

Displays if you specify the SCALE= option in the MODEL statement. Small p-values reject the null hypothesis that the fitted model is adequate. See the section "Overdispersion" on page 3342 for details.

Score Test for the Equal Slopes (Proportional Odds) Assumption

Tests the parallel lines assumption if you fit an ordinal response model with the LINK=CLOGLOG or LINK=PROBIT options. If you specify LINK=LOGIT, this is called the "Proportional Odds" assumption. Small p-values reject the null hypothesis that the slope parameters for each explanatory variable are constant across all the response functions. See the section "Testing the Parallel Lines Assumption" on page 3332 for details.

Model Fit Statistics

Computes various fit criteria based on a model with intercepts only and a model with intercepts and explanatory variables. If you specify the NOINT option in the MODEL statement, these statistics are calculated without considering the intercept parameters. See the section "Model Fitting Information" on page 3329 for details.

Testing Global Null Hypothesis: BETA=0

Tests the joint effect of the explanatory variables included in the model. Small p-values reject the null hypothesis that all slope parameters are equal to zero, $H_0: \boldsymbol{\beta} = \mathbf{0}$. See the sections "Model Fitting Information" on page 3329, "Residual Chi-Square" on page 3331, and "Testing Linear Hypotheses about the Regression Coefficients" on page 3348 for details. If you also specify the RSQUARE option in the MODEL statement, two generalized R^2 measures are included; see the section "Generalized Coefficient of Determination" on page 3330 for details.

Score Test for Global Null Hypothesis

Displays instead of the "Testing Global Null Hypothesis: BETA=0" table if the NOFIT option is specified in the MODEL statement. The global score test evaluates the joint significance of the effects in the MODEL statement. Small p-values reject the null hypothesis that all slope parameters are equal to zero, $H_0: \boldsymbol{\beta} = \mathbf{0}$. See the section "Residual Chi-Square" on page 3331 for details.

Model Selection Tables

The tables in this section are produced when the SELECTION= option is specified in the MODEL statement. See the section "Effect-Selection Methods" on page 3328 for more information.

- **Residual Chi-Square Test**
  Displays if you specify SELECTION=FORWARD, BACKWARD, or STEPWISE in the MODEL statement. Small p-values reject the null hypothesis that the reduced model is adequate. See the section "Residual Chi-Square" on page 3331 for details.

- **Analysis of Effects Eligible for Entry**
  Displays if you specify the DETAILS option and the SELECTION=FORWARD or STEPWISE option in the MODEL statement. Small p-values reject $H_0: \beta_i \neq 0$. The score chi-square is used to determine entry; see the section "Testing Individual Effects Not in the Model" on page 3332 for details.

- **Analysis of Effects Eligible for Removal**
  Displays if you specify the SELECTION=BACKWARD or STEPWISE option in the MODEL statement. Small p-values reject $H_0: \beta_i = 0$. The Wald chi-square is used to determine removal; see the section "Testing Linear Hypotheses about the Regression Coefficients" on page 3348 for details.

- **Analysis of Effects Removed by Fast Backward Elimination**
  Displays if you specify the FAST option and the SELECTION=BACKWARD or STEPWISE option in the MODEL statement. This table gives the approximate chi-square statistic for the

variable removed, the corresponding p-value with respect to a chi-square distribution with one degree of freedom, the residual chi-square statistic for testing the joint significance of the variable and the preceding ones, the degrees of freedom, and the p-value of the residual chi-square with respect to a chi-square distribution with the corresponding degrees of freedom.

- **Summary of Forward, Backward, and Stepwise Selection**
  Displays if you specify SELECTION=FORWARD, BACKWARD, or STEPWISE in the MODEL statement. The score chi-square is used to determine entry; see the section "Testing Individual Effects Not in the Model" on page 3332 for details. The Wald chi-square is used to determine removal; see the section "Testing Linear Hypotheses about the Regression Coefficients" on page 3348 for details.

- **Regression Models Selected by Score Criterion**
  Displays the score chi-square for all models if you specify the SELECTION=SCORE option in the MODEL statement. Small p-values reject the null hypothesis that the fitted model is adequate. See the section "Effect-Selection Methods" on page 3328 for details.

Type 3 Analysis of Effect

Displays if the model contains a CLASS variable. Performs Wald chi-square tests of the joint effect of the parameters for each CLASS variable in the model. Small p-values reject H_0: $\boldsymbol{\beta}_i = 0$. See the section "Testing Linear Hypotheses about the Regression Coefficients" on page 3348 for details.

Analysis of Maximum Likelihood Estimates

CLASS effects are identified by their (nonreference) level. For generalized logit models, a response variable column displays the nonreference level of the logit. The table includes the following:

- the estimated standard error of the parameter estimate, computed as the square root of the corresponding diagonal element of the estimated covariance matrix
- the Wald chi-square statistic, computed by squaring the ratio of the parameter estimate divided by its standard error estimate. See the section "Testing Linear Hypotheses about the Regression Coefficients" on page 3348 for details.
- the p-value tests the null hypothesis H_0: $\beta_i = 0$; small values reject the null.
- the standardized estimate for the slope parameter, if you specify the STB option in the MODEL statement. See the STB option on page 3302 for details.
- exponentiated values of the estimates of the slope parameters, if you specify the EXPB option in the MODEL statement. See the EXPB option on page 3293 for details.
- the label of the variable, if you specify the PARMLABEL option in the MODEL statement and if space permits. Due to constraints on the line size, the variable label might be suppressed in order to display the table in one panel. Use the SAS system option LINESIZE= to specify a larger line size to accommodate variable labels. A shorter line size can break the table into two panels allowing labels to be displayed.

Odds Ratio Estimates

Displays the odds ratio estimates and the corresponding 95% Wald confidence intervals for variables that are not involved in nestings or interactions. For continuous explanatory variables, these odds ratios correspond to a unit increase in the risk factors. See the section "Odds Ratio Estimation" on page 3335 for details.

Association of Predicted Probabilities and Observed Responses

See the section "Rank Correlation of Observed Responses and Predicted Probabilities" on page 3338 for details.

Profile-Likelihood or Wald Confidence Intervals for Parameters

Displays if you specify the CLPARM= option in the MODEL statement. See the section "Confidence Intervals for Parameters" on page 3333 for details.

Profile-Likelihood or Wald Confidence Intervals for Odds Ratios

Displays if you specify the ODDSRATIO statement for any effects with any class parameterizations. Also displays if you specify the CLODDS= option in the MODEL statement, except odds ratios are computed only for main effects not involved in interactions or nestings, and if the main effect is a CLASS variable, the parameterization must be EFFECT, REFERENCE, or GLM. See the section "Odds Ratio Estimation" on page 3335 for details.

Estimated Covariance or Correlation Matrix

Displays if you specify the COVB or CORRB option in the MODEL statement. See the section "Iterative Algorithms for Model Fitting" on page 3324 for details.

Contrast Test Results

Displays the Wald test for each specified CONTRAST statement. Small p-values reject $H_0: \mathbf{L}\boldsymbol{\beta} = \mathbf{0}$. The "Coefficients of Contrast" table displays the contrast matrix if you specify the E option, and the "Contrast Rows Estimation and Testing Results" table displays estimates and Wald tests for each row of the contrast matrix if you specify the ESTIMATE= option. See the sections "CONTRAST Statement" on page 3282, "Testing Linear Hypotheses about the Regression Coefficients" on page 3348, and "Linear Predictor, Predicted Probability, and Confidence Limits" on page 3339 for details.

Linear Hypotheses Testing Results

Displays the Wald test for each specified TEST statement. See the sections "Testing Linear Hypotheses about the Regression Coefficients" on page 3348 and "TEST Statement" on page 3315 for details.

Hosmer and Lemeshow Goodness-of-Fit Test

Displays if you specify the LACKFIT option in the MODEL statement. Small p-values reject the null hypothesis that the fitted model is adequate. The "Partition for the Hosmer and Lemeshow Test" table displays the grouping used in the test. See the section "The Hosmer-Lemeshow Goodness-of-Fit Test" on page 3344 for details.

Classification Table

Displays if you use the CTABLE option in the MODEL statement. If you specify a list of cutpoints with the PPROB= option, then the cutpoints are displayed in the Prob Level column. If you specify the prior event probabilities with the PEVENT= option, then the probabilities are displayed in the Prob Event column. The Correct column displays the number of correctly classified events and nonevents, the Incorrect Event column displays the number of nonevents incorrectly classified as events, and the Incorrect Nonevent column gives the number of nonevents incorrectly classified as events. See the section "Classification Table" on page 3340 for more details.

Regression Diagnostics

Displays if you specify the INFLUENCE option in the MODEL statement. See the section "Regression Diagnostics" on page 3349 for more information about diagnostics from an unconditional analysis, and the section "Regression Diagnostic Details" on page 3357 for information about diagnostics from a conditional analysis.

Fit Statistics for SCORE Data

Displays if you specify the FITSTAT option in the SCORE statement. See the section "Scoring Data Sets" on page 3352 for other details.

ROC Association Statistic and Contrast Tables

Displayed if a ROC statement and/or a ROCCONTRAST statement is specified. See the section "ROC Computations" on page 3347 for details about the Mann-Whitney statistics and the test and estimation computations, and see the section "Rank Correlation of Observed Responses and Predicted Probabilities" on page 3338 for details about the other statistics.

Exact Conditional Logistic Regression Tables

The tables in this section are produced when the EXACT statement is specified. If the METHOD=NETWORKMC option is specified, the test and estimate tables are renamed "Monte Carlo" tables and a Monte Carlo standard error column ($\sqrt{p(1-p)/n}$) is displayed.

- **Sufficient Statistics**
  Displays if you request an OUTDIST= data set in an EXACT statement. The table lists the parameters and their observed sufficient statistics.

- **(Monte Carlo) Conditional Exact Tests**
  See the section "Hypothesis Tests" on page 3361 for details.
- **(Monte Carlo) Exact Parameter Estimates**
  Displays if you specify the ESTIMATE option in the EXACT statement. This table gives individual parameter estimates for each variable (conditional on the values of all the other parameters in the model), confidence limits, and a two-sided *p*-value (twice the one-sided *p*-value) for testing that the parameter is zero. See the section "Inference for a Single Parameter" on page 3362 for details.
- **(Monte Carlo) Exact Odds Ratios**
  Displays if you specify the ESTIMATE=ODDS or ESTIMATE=BOTH option in the EXACT statement. See the section "Inference for a Single Parameter" on page 3362 for details.

ODS Table Names

PROC LOGISTIC assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 51.5. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 51.5 ODS Tables Produced by PROC LOGISTIC

ODS Table Name	Description	Statement	Option
Association	Association of predicted probabilities and observed responses	MODEL (without STRATA)	default
BestSubsets	Best subset selection	MODEL	SELECTION=SCORE
ClassFreq	Frequency breakdown of CLASS variables	PROC	Simple (with CLASS vars)
ClassLevelInfo	CLASS variable levels and design variables	MODEL	default (with CLASS vars)
Classification	Classification table	MODEL	CTABLE
ClassWgt	Weight breakdown of CLASS variables	PROC, WEIGHT	Simple (with CLASS vars)
CLOddsPL	Profile-likelihood confidence limits for odds ratios	MODEL	CLODDS=PL
CLOddsWald	Wald's confidence limits for odds ratios	MODEL	CLODDS=WALD
CLParmPL	Profile-likelihood confidence limits for parameters	MODEL	CLPARM=PL
CLParmWald	Wald's confidence limits for parameters	MODEL	CLPARM=WALD

Table 51.5 *continued*

ODS Table Name	Description	Statement	Option
ContrastCoeff	L matrix from CONTRAST	CONTRAST	E
ContrastEstimate	Estimates from CONTRAST	CONTRAST	ESTIMATE=
ContrastTest	Wald test for CONTRAST	CONTRAST	default
ConvergenceStatus	Convergence status	MODEL	default
CorrB	Estimated correlation matrix of parameter estimators	MODEL	CORRB
CovB	Estimated covariance matrix of parameter estimators	MODEL	COVB
CumulativeModelTest	Test of the cumulative model assumption	MODEL	(ordinal response)
EffectNotInModel	Test for effects not in model	MODEL	SELECTION=S|F
ExactOddsRatio	Exact odds ratios	EXACT	ESTIMATE=ODDS, ESTIMATE=BOTH
ExactParmEst	Parameter estimates	EXACT	ESTIMATE, ESTIMATE=PARM, ESTIMATE=BOTH
ExactTests	Conditional exact tests	EXACT	default
FastElimination	Fast backward elimination	MODEL	SELECTION=B,FAST
FitStatistics	Model fit statistics	MODEL	default
GlobalScore	Global score test	MODEL	NOFIT
GlobalTests	Test for global null hypothesis	MODEL	default
GoodnessOfFit	Pearson and deviance goodness-of-fit tests	MODEL	SCALE
IndexPlots	Batch capture of the index plots	MODEL	IPLOTS
Influence	Regression diagnostics	MODEL	INFLUENCE
IterHistory	Iteration history	MODEL	ITPRINT
LackFitChiSq	Hosmer-Lemeshow chi-square test results	MODEL	LACKFIT
LackFitPartition	Partition for the Hosmer-Lemeshow test	MODEL	LACKFIT
LastGradient	Last evaluation of gradient	MODEL	ITPRINT
Linear	Linear combination	PROC	default
LogLikeChange	Final change in the log likelihood	MODEL	ITPRINT
ModelBuildingSummary	Summary of model building	MODEL	SELECTION=B|F|S
ModelInfo	Model information	PROC	default
NObs	Number of observations	PROC	default
OddsEst	Adjusted odds ratios	UNITS	default
OddsRatios	Odds ratios	MODEL	default
OddsRatiosWald	Odds ratios with Wald confidence limits	ODDSRATIOS	CL=WALD

Table 51.5 *continued*

ODS Table Name	Description	Statement	Option
OddsRatiosPL	Odds ratios with PL confidence limits	ODDSRATIOS	CL=PL
ParameterEstimates	Maximum likelihood estimates of model parameters	MODEL	default
RSquare	R-square	MODEL	RSQUARE
ResidualChiSq	Residual chi-square	MODEL	SELECTION=F|B
ResponseProfile	Response profile	PROC	default
ROCAssociation	Association table for ROC models	ROC	default
ROCContrastCoeff	L matrix from ROCCONTRAST	ROCCONTRAST	E
ROCContrastCov	Covariance of ROCCONTRAST rows	ROCCONTRAST	COV
ROCContrastEstimate	Estimates from ROCCONTRAST	ROCCONTRAST	ESTIMATE=
ROCContrastTest	Wald test from ROCCONTRAST	ROCCONTRAST	default
ROCCov	Covariance between ROC curves	ROCCONTRAST	COV
SimpleStatistics	Summary statistics for explanatory variables	PROC	SIMPLE
StrataSummary	Number of strata with specific response frequencies	STRATA	default
StrataInfo	Event and nonevent frequencies for each stratum	STRATA	INFO
SuffStats	Sufficient statistics	EXACT	OUTDIST=
TestPrint1	**L**[Cov(**b**)]**L**' and **Lb**-**c**	TEST	PRINT
TestPrint2	Ginv(**L**[Cov(**b**)]**L**') and Ginv(**L**[Cov(**b**)]**L**')(**Lb**-**c**)	TEST	PRINT
TestStmts	Linear hypotheses testing results	TEST	default
Type3	Type 3 tests of effects	MODEL	default (with CLASS variables)

ODS Graphics

PROC LOGISTIC assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 51.6.

To request these graphs you must specify the ODS GRAPHICS statement in addition to the op-

tions indicated in Table 51.6. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Table 51.6 ODS Graphics Produced by PROC LOGISTIC

ODS Graph Name	Plot Description	Statement or Option
DfBetasPlot	Panel of dfbetas by case number	PLOTS=DFBETAS or MODEL / INFLUENCE or IPLOTS
	Effect dfbetas by case number	PLOTS=DFBETAS(UNPACK)
DPCPlot	Difchisq and/or difdev by predicted probability by CI displacement C	PLOTS=DPC
EffectPlot	Predicted probability	PLOTS=EFFECT
InfluencePlots	Panel of influence statistics by case number	default or PLOTS=INFLUENCE or MODEL / INFLUENCE or IPLOTS
CBarPlot	CI displacement Cbar by case number	PLOTS=INFLUENCE (UNPACK)
CPlot	CI displacement C by case number	PLOTS=INFLUENCE (UNPACK)
DevianceResidualPlot	Deviance residual by case number	PLOTS=INFLUENCE (UNPACK)
DifChisqPlot	Difchisq by case number	PLOTS=INFLUENCE (UNPACK)
DifDeviancePlot	Difdev by case number	PLOTS=INFLUENCE (UNPACK)
LeveragePlot	Hat diagonal by case number	PLOTS=INFLUENCE (UNPACK)
PearsonResidualPlot	Pearson chi-square residual by case number	PLOTS=INFLUENCE (UNPACK)
LeveragePlots	Panel of influence statistics by leverage	PLOTS=LEVERAGE
LeverageCPlot	CI displacement C by leverage	PLOTS=LEVERAGE(UNPACK)
LeverageDifChisqPlot	Difchisq by leverage	PLOTS=LEVERAGE(UNPACK)
LeverageDifDevPlot	Difdev by leverage	PLOTS=LEVERAGE(UNPACK)
LeveragePhatPlot	Predicted probability by leverage	PLOTS=LEVERAGE(UNPACK)
ORPlot	Odds ratios	PLOTS=ODDSRATIO or MODEL / CLODDS= or ODDSRATIO
PhatPlots	Panel of influence by predicted probability	PLOTS=PHAT
PhatCPlot	CI displacement C by predicted probability	PLOTS=PHAT(UNPACK)
PhatDifChisqPlot	Difchisq by predicted probability	PLOTS=PHAT(UNPACK)
PhatDifDevPlot	Difdev by predicted probability	PLOTS=PHAT(UNPACK)
PhatLeveragePlot	Leverage by predicted probability	PLOTS=PHAT(UNPACK)
ROCCurve	Receiver operating characteristics curve	PLOTS=ROC or MODEL / OUTROC= or SCORE OUTROC= or ROC
ROCOverlay	ROC curves for comparisons	PLOTS=ROC and MODEL / SELECTION= or ROC

Examples: LOGISTIC Procedure

Example 51.1: Stepwise Logistic Regression and Predicted Values

Consider a study on cancer remission (Lee 1974). The data consist of patient characteristics and whether or not cancer remission occured. The following DATA step creates the data set Remission containing seven variables. The variable remiss is the cancer remission indicator variable with a value of 1 for remission and a value of 0 for nonremission. The other six variables are the risk factors thought to be related to cancer remission.

```
data Remission;
   input remiss cell smear infil li blast temp;
   label remiss='Complete Remission';
   datalines;
1   .8   .83  .66  1.9  1.1     .996
1   .9   .36  .32  1.4   .74    .992
0   .8   .88  .7    .8   .176   .982
0  1     .87  .87   .7  1.053   .986
1   .9   .75  .68  1.3   .519   .98
0  1     .65  .65   .6   .519   .982
1   .95  .97  .92  1    1.23    .992
0   .95  .87  .83  1.9  1.354  1.02
0  1     .45  .45   .8   .322   .999
0   .95  .36  .34   .5  0      1.038
0   .85  .39  .33   .7   .279   .988
0   .7   .76  .53  1.2   .146   .982
0   .8   .46  .37   .4   .38   1.006
0   .2   .39  .08   .8   .114   .99
0  1     .9   .9   1.1  1.037   .99
1  1     .84  .84  1.9  2.064  1.02
0   .65  .42  .27   .5   .114  1.014
0  1     .75  .75  1    1.322  1.004
0   .5   .44  .22   .6   .114   .99
1  1     .63  .63  1.1  1.072   .986
0  1     .33  .33   .4   .176  1.01
0   .9   .93  .84   .6  1.591  1.02
1  1     .58  .58  1     .531  1.002
0   .95  .32  .3   1.6   .886   .988
1  1     .6   .6   1.7   .964   .99
1  1     .69  .69   .9   .398   .986
0  1     .73  .73   .7   .398   .986
;
```

The following invocation of PROC LOGISTIC illustrates the use of stepwise selection to identify the prognostic factors for cancer remission. A significance level of 0.3 is required to allow a variable into the model (SLENTRY=0.3), and a significance level of 0.35 is required for a variable to stay in the model (SLSTAY=0.35). A detailed account of the variable selection process is requested by specifying the DETAILS option. The Hosmer and Lemeshow goodness-of-fit test for the final

selected model is requested by specifying the LACKFIT option. The OUTEST= and COVOUT options in the PROC LOGISTIC statement create a data set that contains parameter estimates and their covariances for the final selected model. The response variable option EVENT= chooses remiss=1 (remission) as the event so that the probability of remission is modeled. The OUTPUT statement creates a data set that contains the cumulative predicted probabilities and the corresponding confidence limits, and the individual and cross validated predicted probabilities for each observation.

```
title 'Stepwise Regression on Cancer Remission Data';
proc logistic data=Remission outest=betas covout;
   model remiss(event='1')=cell smear infil li blast temp
                / selection=stepwise
                  slentry=0.3
                  slstay=0.35
                  details
                  lackfit;
   output out=pred p=phat lower=lcl upper=ucl
          predprob=(individual crossvalidate);
run;

proc print data=betas;
   title2 'Parameter Estimates and Covariance Matrix';
run;

proc print data=pred;
   title2 'Predicted Probabilities and 95% Confidence Limits';
run;
```

In stepwise selection, an attempt is made to remove any insignificant variables from the model before adding a significant variable to the model. Each addition or deletion of a variable to or from a model is listed as a separate step in the displayed output, and at each step a new model is fitted. Details of the model selection steps are shown in Outputs 51.1.1 through 51.1.5.

Prior to the first step, the intercept-only model is fit and individual score statistics for the potential variables are evaluated (Output 51.1.1).

Output 51.1.1 Startup Model

```
                  Stepwise Regression on Cancer Remission Data

Step  0. Intercept entered:


                            Model Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.

                              -2 Log L = 34.372

                   Analysis of Maximum Likelihood Estimates

                                      Standard          Wald
      Parameter    DF    Estimate        Error    Chi-Square    Pr > ChiSq

      Intercept     1     -0.6931       0.4082        2.8827        0.0895
```

Output 51.1.1 *continued*

Residual Chi-Square Test

Chi-Square	DF	Pr > ChiSq
9.4609	6	0.1493

Analysis of Effects Eligible for Entry

Effect	DF	Score Chi-Square	Pr > ChiSq
cell	1	1.8893	0.1693
smear	1	1.0745	0.2999
infil	1	1.8817	0.1701
li	1	7.9311	0.0049
blast	1	3.5258	0.0604
temp	1	0.6591	0.4169

In Step 1 (Output 51.1.2), the variable li is selected into the model since it is the most significant variable among those to be chosen ($p = 0.0049 < 0.3$). The intermediate model that contains an intercept and li is then fitted. li remains significant ($p = 0.0146 < 0.35$) and is not removed.

Output 51.1.2 Step 1 of the Stepwise Analysis

Step 1. Effect li entered:

Model Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	36.372	30.073
SC	37.668	32.665
-2 Log L	34.372	26.073

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	8.2988	1	0.0040
Score	7.9311	1	0.0049
Wald	5.9594	1	0.0146

Output 51.1.2 *continued*

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-3.7771	1.3786	7.5064	0.0061
li	1	2.8973	1.1868	5.9594	0.0146

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
li	18.124	1.770	185.563

Association of Predicted Probabilities and Observed Responses

Percent Concordant	84.0	Somers' D	0.710
Percent Discordant	13.0	Gamma	0.732
Percent Tied	3.1	Tau-a	0.328
Pairs	162	c	0.855

Residual Chi-Square Test

Chi-Square	DF	Pr > ChiSq
3.1174	5	0.6819

Analysis of Effects Eligible for Removal

Effect	DF	Wald Chi-Square	Pr > ChiSq
li	1	5.9594	0.0146

NOTE: No effects for the model in Step 1 are removed.

Analysis of Effects Eligible for Entry

Effect	DF	Score Chi-Square	Pr > ChiSq
cell	1	1.1183	0.2903
smear	1	0.1369	0.7114
infil	1	0.5715	0.4497
blast	1	0.0932	0.7601
temp	1	1.2591	0.2618

In Step 2 (Output 51.1.3), the variable temp is added to the model. The model then contains an intercept and the variables li and temp. Both li and temp remain significant at 0.35 level; therefore, neither li nor temp is removed from the model.

Output 51.1.3 Step 2 of the Stepwise Analysis

Step 2. Effect temp entered:

Model Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	36.372	30.648
SC	37.668	34.535
-2 Log L	34.372	24.648

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	9.7239	2	0.0077
Score	8.3648	2	0.0153
Wald	5.9052	2	0.0522

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	47.8448	46.4381	1.0615	0.3029
li	1	3.3017	1.3593	5.9002	0.0151
temp	1	-52.4214	47.4897	1.2185	0.2697

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
li	27.158	1.892	389.856
temp	<0.001	<0.001	>999.999

Association of Predicted Probabilities and Observed Responses

Percent Concordant	87.0	Somers' D	0.747
Percent Discordant	12.3	Gamma	0.752
Percent Tied	0.6	Tau-a	0.345
Pairs	162	c	0.873

Residual Chi-Square Test

Chi-Square	DF	Pr > ChiSq
2.1429	4	0.7095

Output 51.1.3 *continued*

```
                    Analysis of Effects Eligible for Removal

                                          Wald
                   Effect      DF    Chi-Square    Pr > ChiSq

                   li           1        5.9002        0.0151
                   temp         1        1.2185        0.2697

NOTE: No effects for the model in Step 2 are removed.


                     Analysis of Effects Eligible for Entry

                                         Score
                   Effect      DF    Chi-Square    Pr > ChiSq

                   cell         1        1.4700        0.2254
                   smear        1        0.1730        0.6775
                   infil        1        0.8274        0.3630
                   blast        1        1.1013        0.2940
```

In Step 3 (Output 51.1.4), the variable cell is added to the model. The model then contains an intercept and the variables li, temp, and cell. None of these variables are removed from the model since all are significant at the 0.35 level.

Output 51.1.4 Step 3 of the Stepwise Analysis

```
Step  3. Effect cell entered:


                            Model Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.

                              Model Fit Statistics

                                                  Intercept
                                     Intercept            and
                     Criterion           Only     Covariates

                     AIC               36.372         29.953
                     SC                37.668         35.137
                     -2 Log L          34.372         21.953

                    Testing Global Null Hypothesis: BETA=0

            Test                  Chi-Square       DF     Pr > ChiSq

            Likelihood Ratio         12.4184        3         0.0061
            Score                     9.2502        3         0.0261
            Wald                      4.8281        3         0.1848
```

Output 51.1.4 *continued*

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	67.6339	56.8875	1.4135	0.2345
cell	1	9.6521	7.7511	1.5507	0.2130
li	1	3.8671	1.7783	4.7290	0.0297
temp	1	-82.0737	61.7124	1.7687	0.1835

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
cell	>999.999	0.004	>999.999
li	47.804	1.465	>999.999
temp	<0.001	<0.001	>999.999

Association of Predicted Probabilities and Observed Responses

Percent Concordant	88.9	Somers' D	0.778
Percent Discordant	11.1	Gamma	0.778
Percent Tied	0.0	Tau-a	0.359
Pairs	162	c	0.889

Residual Chi-Square Test

Chi-Square	DF	Pr > ChiSq
0.1831	3	0.9803

Analysis of Effects Eligible for Removal

Effect	DF	Wald Chi-Square	Pr > ChiSq
cell	1	1.5507	0.2130
li	1	4.7290	0.0297
temp	1	1.7687	0.1835

NOTE: No effects for the model in Step 3 are removed.

Analysis of Effects Eligible for Entry

Effect	DF	Score Chi-Square	Pr > ChiSq
smear	1	0.0956	0.7572
infil	1	0.0844	0.7714
blast	1	0.0208	0.8852

Finally, none of the remaining variables outside the model meet the entry criterion, and the stepwise selection is terminated. A summary of the stepwise selection is displayed in Output 51.1.5.

Output 51.1.5 Summary of the Stepwise Selection

Summary of Stepwise Selection

Step	Effect Entered	Effect Removed	DF	Number In	Score Chi-Square	Wald Chi-Square	Pr > ChiSq
1	li		1	1	7.9311		0.0049
2	temp		1	2	1.2591		0.2618
3	cell		1	3	1.4700		0.2254

Results of the Hosmer and Lemeshow test are shown in Output 51.1.6. There is no evidence of a lack of fit in the selected model ($p = 0.5054$).

Output 51.1.6 Display of the LACKFIT Option

Partition for the Hosmer and Lemeshow Test

Group	Total	remiss = 1 Observed	remiss = 1 Expected	remiss = 0 Observed	remiss = 0 Expected
1	3	0	0.00	3	3.00
2	3	0	0.01	3	2.99
3	3	0	0.19	3	2.81
4	3	0	0.56	3	2.44
5	4	1	1.09	3	2.91
6	3	2	1.35	1	1.65
7	3	2	1.84	1	1.16
8	3	3	2.15	0	0.85
9	2	1	1.80	1	0.20

Hosmer and Lemeshow Goodness-of-Fit Test

Chi-Square	DF	Pr > ChiSq
6.2983	7	0.5054

The data set betas created by the OUTEST= and COVOUT options is displayed in Output 51.1.7. The data set contains parameter estimates and the covariance matrix for the final selected model. Note that all explanatory variables listed in the MODEL statement are included in this data set; however, variables that are not included in the final model have all missing values.

Output 51.1.7 Data Set of Estimates and Covariances

Stepwise Regression on Cancer Remission Data
Parameter Estimates and Covariance Matrix

Obs	_LINK_	_TYPE_	_STATUS_	_NAME_	Intercept	cell
1	LOGIT	PARMS	0 Converged	remiss	67.63	9.652
2	LOGIT	COV	0 Converged	Intercept	3236.19	157.097
3	LOGIT	COV	0 Converged	cell	157.10	60.079
4	LOGIT	COV	0 Converged	smear	.	.
5	LOGIT	COV	0 Converged	infil	.	.
6	LOGIT	COV	0 Converged	li	64.57	6.945
7	LOGIT	COV	0 Converged	blast	.	.
8	LOGIT	COV	0 Converged	temp	-3483.23	-223.669

Obs	smear	infil	li	blast	temp	_LNLIKE_
1	.	.	3.8671	.	-82.07	-10.9767
2	.	.	64.5726	.	-3483.23	-10.9767
3	.	.	6.9454	.	-223.67	-10.9767
4	.	.	.	.	.	-10.9767
5	.	.	.	.	.	-10.9767
6	.	.	3.1623	.	-75.35	-10.9767
7	.	.	.	.	.	-10.9767
8	.	.	-75.3513	.	3808.42	-10.9767

The data set pred created by the OUTPUT statement is displayed in Output 51.1.8. It contains all the variables in the input data set, the variable phat for the (cumulative) predicted probability, the variables lcl and ucl for the lower and upper confidence limits for the probability, and four other variables (IP_1, IP_0, XP_1, and XP_0) for the PREDPROBS= option. The data set also contains the variable _LEVEL_, indicating the response value to which phat, lcl, and ucl refer. For instance, for the first row of the OUTPUT data set, the values of _LEVEL_ and phat, lcl, and ucl are 1, 0.72265, 0.16892, and 0.97093, respectively; this means that the estimated probability that remiss=1 is 0.723 for the given explanatory variable values, and the corresponding 95% confidence interval is (0.16892, 0.97093). The variables IP_1 and IP_0 contain the predicted probabilities that remiss=1 and remiss=0, respectively. Note that values of phat and IP_1 are identical since they both contain the probabilities that remiss=1. The variables XP_1 and XP_0 contain the cross validated predicted probabilities that remiss=1 and remiss=0, respectively.

Output 51.1.8 Predicted Probabilities and Confidence Intervals

Stepwise Regression on Cancer Remission Data
Predicted Probabilities and 95% Confidence Limits

Obs	remiss	cell	smear	infil	li	blast	temp	_FROM_	_INTO_	IP_0
1	1	0.80	0.83	0.66	1.9	1.100	0.996	1	1	0.27735
2	1	0.90	0.36	0.32	1.4	0.740	0.992	1	1	0.42126
3	0	0.80	0.88	0.70	0.8	0.176	0.982	0	0	0.89540
4	0	1.00	0.87	0.87	0.7	1.053	0.986	0	0	0.71742
5	1	0.90	0.75	0.68	1.3	0.519	0.980	1	1	0.28582
6	0	1.00	0.65	0.65	0.6	0.519	0.982	0	0	0.72911
7	1	0.95	0.97	0.92	1.0	1.230	0.992	1	0	0.67844
8	0	0.95	0.87	0.83	1.9	1.354	1.020	0	1	0.39277
9	0	1.00	0.45	0.45	0.8	0.322	0.999	0	0	0.83368
10	0	0.95	0.36	0.34	0.5	0.000	1.038	0	0	0.99843
11	0	0.85	0.39	0.33	0.7	0.279	0.988	0	0	0.92715
12	0	0.70	0.76	0.53	1.2	0.146	0.982	0	0	0.82714

Obs	IP_1	XP_0	XP_1	_LEVEL_	phat	lcl	ucl
1	0.72265	0.43873	0.56127	1	0.72265	0.16892	0.97093
2	0.57874	0.47461	0.52539	1	0.57874	0.26788	0.83762
3	0.10460	0.87060	0.12940	1	0.10460	0.00781	0.63419
4	0.28258	0.67259	0.32741	1	0.28258	0.07498	0.65683
5	0.71418	0.36901	0.63099	1	0.71418	0.25218	0.94876
6	0.27089	0.67269	0.32731	1	0.27089	0.05852	0.68951
7	0.32156	0.72923	0.27077	1	0.32156	0.13255	0.59516
8	0.60723	0.09906	0.90094	1	0.60723	0.10572	0.95287
9	0.16632	0.80864	0.19136	1	0.16632	0.03018	0.56123
10	0.00157	0.99840	0.00160	1	0.00157	0.00000	0.68962
11	0.07285	0.91723	0.08277	1	0.07285	0.00614	0.49982
12	0.17286	0.63838	0.36162	1	0.17286	0.00637	0.87206

Output 51.1.8 *continued*

Stepwise Regression on Cancer Remission Data
Predicted Probabilities and 95% Confidence Limits

Obs	remiss	cell	smear	infil	li	blast	temp	_FROM_	_INTO_	IP_0
13	0	0.80	0.46	0.37	0.4	0.380	1.006	0	0	0.99654
14	0	0.20	0.39	0.08	0.8	0.114	0.990	0	0	0.99982
15	0	1.00	0.90	0.90	1.1	1.037	0.990	0	1	0.42878
16	1	1.00	0.84	0.84	1.9	2.064	1.020	1	1	0.28530
17	0	0.65	0.42	0.27	0.5	0.114	1.014	0	0	0.99938
18	0	1.00	0.75	0.75	1.0	1.322	1.004	0	0	0.77711
19	0	0.50	0.44	0.22	0.6	0.114	0.990	0	0	0.99846
20	1	1.00	0.63	0.63	1.1	1.072	0.986	1	1	0.35089
21	0	1.00	0.33	0.33	0.4	0.176	1.010	0	0	0.98307
22	0	0.90	0.93	0.84	0.6	1.591	1.020	0	0	0.99378
23	1	1.00	0.58	0.58	1.0	0.531	1.002	1	0	0.74739
24	0	0.95	0.32	0.30	1.6	0.886	0.988	0	1	0.12989

Obs	IP_1	XP_0	XP_1	_LEVEL_	phat	lcl	ucl
13	0.00346	0.99644	0.00356	1	0.00346	0.00001	0.46530
14	0.00018	0.99981	0.00019	1	0.00018	0.00000	0.96482
15	0.57122	0.35354	0.64646	1	0.57122	0.25303	0.83973
16	0.71470	0.47213	0.52787	1	0.71470	0.15362	0.97189
17	0.00062	0.99937	0.00063	1	0.00062	0.00000	0.62665
18	0.22289	0.73612	0.26388	1	0.22289	0.04483	0.63670
19	0.00154	0.99842	0.00158	1	0.00154	0.00000	0.79644
20	0.64911	0.42053	0.57947	1	0.64911	0.26305	0.90555
21	0.01693	0.98170	0.01830	1	0.01693	0.00029	0.50475
22	0.00622	0.99348	0.00652	1	0.00622	0.00003	0.56062
23	0.25261	0.84423	0.15577	1	0.25261	0.06137	0.63597
24	0.87011	0.03637	0.96363	1	0.87011	0.40910	0.98481

Stepwise Regression on Cancer Remission Data
Predicted Probabilities and 95% Confidence Limits

Obs	remiss	cell	smear	infil	li	blast	temp	_FROM_	_INTO_	IP_0
25	1	1.00	0.60	0.60	1.7	0.964	0.990	1	1	0.06868
26	1	1.00	0.69	0.69	0.9	0.398	0.986	1	0	0.53949
27	0	1.00	0.73	0.73	0.7	0.398	0.986	0	0	0.71742

Obs	IP_1	XP_0	XP_1	_LEVEL_	phat	lcl	ucl
25	0.93132	0.08017	0.91983	1	0.93132	0.44114	0.99573
26	0.46051	0.62312	0.37688	1	0.46051	0.16612	0.78529
27	0.28258	0.67259	0.32741	1	0.28258	0.07498	0.65683

Next, a different variable selection method is used to select prognostic factors for cancer remission, and an efficient algorithm is employed to eliminate insignificant variables from a model. The following statements invoke PROC LOGISTIC to perform the backward elimination analysis:

```
title 'Backward Elimination on Cancer Remission Data';
proc logistic data=Remission;
   model remiss(event='1')=temp cell li smear blast
         / selection=backward fast slstay=0.2 ctable;
run;
```

The backward elimination analysis (SELECTION=BACKWARD) starts with a model that contains all explanatory variables given in the MODEL statement. By specifying the FAST option, PROC LOGISTIC eliminates insignificant variables without refitting the model repeatedly. This analysis uses a significance level of 0.2 to retain variables in the model (SLSTAY=0.2), which is different from the previous stepwise analysis where SLSTAY=.35. The CTABLE option is specified to produce classifications of input observations based on the final selected model.

Results of the fast elimination analysis are shown in Output 51.1.9 and Output 51.1.10. Initially, a full model containing all six risk factors is fit to the data (Output 51.1.9). In the next step (Output 51.1.10), PROC LOGISTIC removes blast, smear, cell, and temp from the model all at once. This leaves li and the intercept as the only variables in the final model. Note that in this analysis, only parameter estimates for the final model are displayed because the DETAILS option has not been specified.

Output 51.1.9 Initial Step in Backward Elimination

```
                 Backward Elimination on Cancer Remission Data

                              Model Information

     Data Set                      WORK.REMISSION
     Response Variable             remiss                Complete Remission
     Number of Response Levels     2
     Model                         binary logit
     Optimization Technique        Fisher's scoring

                    Number of Observations Read          27
                    Number of Observations Used          27

                               Response Profile

                       Ordered                       Total
                         Value       remiss     Frequency

                             1            0            18
                             2            1             9

                        Probability modeled is remiss=1.

                         Backward Elimination Procedure

Step  0. The following effects were entered:

Intercept  temp  cell  li  smear  blast


                            Model Convergence Status

                 Convergence criterion (GCONV=1E-8) satisfied.
```

Output 51.1.9 *continued*

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	36.372	33.857
SC	37.668	41.632
-2 Log L	34.372	21.857

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	12.5146	5	0.0284
Score	9.3295	5	0.0966
Wald	4.7284	5	0.4499

Output 51.1.10 Fast Elimination Step

Step 1. Fast Backward Elimination:

Analysis of Effects Removed by Fast Backward Elimination

Effect Removed	Chi-Square	DF	Pr > ChiSq	Residual Chi-Square	DF	Pr > Residual ChiSq
blast	0.0008	1	0.9768	0.0008	1	0.9768
smear	0.0951	1	0.7578	0.0959	2	0.9532
cell	1.5134	1	0.2186	1.6094	3	0.6573
temp	0.6535	1	0.4189	2.2628	4	0.6875

Model Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	36.372	30.073
SC	37.668	32.665
-2 Log L	34.372	26.073

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	8.2988	1	0.0040
Score	7.9311	1	0.0049
Wald	5.9594	1	0.0146

Output 51.1.10 *continued*

Residual Chi-Square Test

Chi-Square	DF	Pr > ChiSq
2.8530	4	0.5827

Summary of Backward Elimination

Step	Effect Removed	DF	Number In	Wald Chi-Square	Pr > ChiSq
1	blast	1	4	0.0008	0.9768
1	smear	1	3	0.0951	0.7578
1	cell	1	2	1.5134	0.2186
1	temp	1	1	0.6535	0.4189

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-3.7771	1.3786	7.5064	0.0061
li	1	2.8973	1.1868	5.9594	0.0146

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
li	18.124	1.770	185.563

Association of Predicted Probabilities and Observed Responses

Percent Concordant	84.0	Somers' D	0.710
Percent Discordant	13.0	Gamma	0.732
Percent Tied	3.1	Tau-a	0.328
Pairs	162	c	0.855

Note that you can also use the FAST option when SELECTION=STEPWISE. However, the FAST option operates only on backward elimination steps. In this example, the stepwise process only adds variables, so the FAST option would not be useful.

Results of the CTABLE option are shown in Output 51.1.11.

Output 51.1.11 Classifying Input Observations

Classification Table

Prob Level	Correct Event	Correct Non-Event	Incorrect Event	Incorrect Non-Event	Percentages Correct	Sensitivity	Specificity	False POS	False NEG
0.060	9	0	18	0	33.3	100.0	0.0	66.7	.
0.080	9	2	16	0	40.7	100.0	11.1	64.0	0.0
0.100	9	4	14	0	48.1	100.0	22.2	60.9	0.0
0.120	9	4	14	0	48.1	100.0	22.2	60.9	0.0
0.140	9	7	11	0	59.3	100.0	38.9	55.0	0.0
0.160	9	10	8	0	70.4	100.0	55.6	47.1	0.0
0.180	9	10	8	0	70.4	100.0	55.6	47.1	0.0
0.200	8	13	5	1	77.8	88.9	72.2	38.5	7.1
0.220	8	13	5	1	77.8	88.9	72.2	38.5	7.1
0.240	8	13	5	1	77.8	88.9	72.2	38.5	7.1
0.260	6	13	5	3	70.4	66.7	72.2	45.5	18.8
0.280	6	13	5	3	70.4	66.7	72.2	45.5	18.8
0.300	6	13	5	3	70.4	66.7	72.2	45.5	18.8
0.320	6	14	4	3	74.1	66.7	77.8	40.0	17.6
0.340	5	14	4	4	70.4	55.6	77.8	44.4	22.2
0.360	5	14	4	4	70.4	55.6	77.8	44.4	22.2
0.380	5	15	3	4	74.1	55.6	83.3	37.5	21.1
0.400	5	15	3	4	74.1	55.6	83.3	37.5	21.1
0.420	5	15	3	4	74.1	55.6	83.3	37.5	21.1
0.440	5	15	3	4	74.1	55.6	83.3	37.5	21.1
0.460	4	16	2	5	74.1	44.4	88.9	33.3	23.8
0.480	4	16	2	5	74.1	44.4	88.9	33.3	23.8
0.500	4	16	2	5	74.1	44.4	88.9	33.3	23.8
0.520	4	16	2	5	74.1	44.4	88.9	33.3	23.8
0.540	3	16	2	6	70.4	33.3	88.9	40.0	27.3
0.560	3	16	2	6	70.4	33.3	88.9	40.0	27.3
0.580	3	16	2	6	70.4	33.3	88.9	40.0	27.3
0.600	3	16	2	6	70.4	33.3	88.9	40.0	27.3
0.620	3	16	2	6	70.4	33.3	88.9	40.0	27.3
0.640	3	16	2	6	70.4	33.3	88.9	40.0	27.3
0.660	3	16	2	6	70.4	33.3	88.9	40.0	27.3
0.680	3	16	2	6	70.4	33.3	88.9	40.0	27.3
0.700	3	16	2	6	70.4	33.3	88.9	40.0	27.3
0.720	2	16	2	7	66.7	22.2	88.9	50.0	30.4
0.740	2	16	2	7	66.7	22.2	88.9	50.0	30.4
0.760	2	16	2	7	66.7	22.2	88.9	50.0	30.4
0.780	2	16	2	7	66.7	22.2	88.9	50.0	30.4
0.800	2	17	1	7	70.4	22.2	94.4	33.3	29.2
0.820	2	17	1	7	70.4	22.2	94.4	33.3	29.2
0.840	0	17	1	9	63.0	0.0	94.4	100.0	34.6
0.860	0	17	1	9	63.0	0.0	94.4	100.0	34.6
0.880	0	17	1	9	63.0	0.0	94.4	100.0	34.6
0.900	0	17	1	9	63.0	0.0	94.4	100.0	34.6
0.920	0	17	1	9	63.0	0.0	94.4	100.0	34.6
0.940	0	17	1	9	63.0	0.0	94.4	100.0	34.6
0.960	0	18	0	9	66.7	0.0	100.0	.	33.3

Each row of the "Classification Table" corresponds to a cutpoint applied to the predicted probabilities, which is given in the Prob Level column. The 2×2 frequency tables of observed and predicted responses are given by the next four columns. For example, with a cutpoint of 0.5, 4 events and 16 nonevents were classified correctly. On the other hand, 2 nonevents were incorrectly classified as events and 5 events were incorrectly classified as nonevents. For this cutpoint, the correct classification rate is 20/27 (=74.1%), which is given in the sixth column. Accuracy of the classification is summarized by the sensitivity, specificity, and false positive and negative rates, which are displayed in the last four columns. You can control the number of cutpoints used, and their values, by using the PPROB= option.

Example 51.2: Logistic Modeling with Categorical Predictors

Consider a study of the analgesic effects of treatments on elderly patients with neuralgia. Two test treatments and a placebo are compared. The response variable is whether the patient reported pain or not. Researchers recorded the age and gender of 60 patients and the duration of complaint before the treatment began. The following DATA step creates the data set Neuralgia:

```
Data Neuralgia;
   input Treatment $ Sex $ Age Duration Pain $ @@;
   datalines;
P  F  68   1  No   B  M  74  16  No  P  F  67  30  No
P  M  66  26  Yes  B  F  67  28  No  B  F  77  16  No
A  F  71  12  No   B  F  72  50  No  B  F  76   9  Yes
A  M  71  17  Yes  A  F  63  27  No  A  F  69  18  Yes
B  F  66  12  No   A  M  62  42  No  P  F  64   1  Yes
A  F  64  17  No   P  M  74   4  No  A  F  72  25  No
P  M  70   1  Yes  B  M  66  19  No  B  M  59  29  No
A  F  64  30  No   A  M  70  28  No  A  M  69   1  No
B  F  78   1  No   P  M  83   1  Yes B  F  69  42  No
B  M  75  30  Yes  P  M  77  29  Yes P  F  79  20  Yes
A  M  70  12  No   A  F  69  12  No  B  F  65  14  No
B  M  70   1  No   B  M  67  23  No  A  M  76  25  Yes
P  M  78  12  Yes  B  M  77   1  Yes B  F  69  24  No
P  M  66   4  Yes  P  F  65  29  No  P  M  60  26  Yes
A  M  78  15  Yes  B  M  75  21  Yes A  F  67  11  No
P  F  72  27  No   P  F  70  13  Yes A  M  75   6  Yes
B  F  65   7  No   P  F  68  27  Yes P  M  68  11  Yes
P  M  67  17  Yes  B  M  70  22  No  A  M  65  15  No
P  F  67   1  Yes  A  M  67  10  No  P  F  72  11  Yes
A  F  74   1  No   B  M  80  21  Yes A  F  69   3  No
;
```

The data set Neuralgia contains five variables: Treatment, Sex, Age, Duration, and Pain. The last variable, Pain, is the response variable. A specification of Pain=Yes indicates there was pain, and Pain=No indicates no pain. The variable Treatment is a categorical variable with three levels: A and B represent the two test treatments, and P represents the placebo treatment. The gender of the patients is given by the categorical variable Sex. The variable Age is the age of the patients, in years, when treatment began. The duration of complaint, in months, before the treatment began is given by the variable Duration.

The following statements use the LOGISTIC procedure to fit a two-way logit with interaction model for the effect of Treatment and Sex, with Age and Duration as covariates. The categorical variables Treatment and Sex are declared in the CLASS statement.

```
proc logistic data=Neuralgia;
   class Treatment Sex;
   model Pain= Treatment Sex Treatment*Sex Age Duration / expb;
run;
```

In this analysis, PROC LOGISTIC models the probability of no pain (Pain=No). By default, effect coding is used to represent the CLASS variables. Two design variables are created for Treatment and one for Sex, as shown in Output 51.2.1.

Output 51.2.1 Effect Coding of CLASS Variables

Class Level Information

Class	Value	Design Variables	
Treatment	A	1	0
	B	0	1
	P	-1	-1
Sex	F	1	
	M	-1	

PROC LOGISTIC displays a table of the Type 3 analysis of effects based on the Wald test (Output 51.2.2). Note that the Treatment*Sex interaction and the duration of complaint are not statistically significant (p =0.9318 and p =0.8752, respectively). This indicates that there is no evidence that the treatments affect pain differently in men and women, and no evidence that the pain outcome is related to the duration of pain.

Output 51.2.2 Wald Tests of Individual Effects

Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
Treatment	2	11.9886	0.0025
Sex	1	5.3104	0.0212
Treatment*Sex	2	0.1412	0.9318
Age	1	7.2744	0.0070
Duration	1	0.0247	0.8752

Parameter estimates are displayed in Output 51.2.3. The Exp(Est) column contains the exponentiated parameter estimates requested with the EXPB option. These values can, but do not necessarily, represent odds ratios for the corresponding variables. For continuous explanatory variables, the Exp(Est) value corresponds to the odds ratio for a unit increase of the corresponding variable. For

CLASS variables that use effect coding, the Exp(Est) values have no direct interpretation as a comparison of levels. However, when the reference coding is used, the Exp(Est) values represent the odds ratio between the corresponding level and the reference level. Following the parameter estimates table, PROC LOGISTIC displays the odds ratio estimates for those variables that are not involved in any interaction terms. If the variable is a CLASS variable, the odds ratio estimate comparing each level with the reference level is computed regardless of the coding scheme. In this analysis, since the model contains the Treatment*Sex interaction term, the odds ratios for Treatment and Sex were not computed. The odds ratio estimates for Age and Duration are precisely the values given in the Exp(Est) column in the parameter estimates table.

Output 51.2.3 Parameter Estimates with Effect Coding

Analysis of Maximum Likelihood Estimates

Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq	Exp(Est)
Intercept		1	19.2236	7.1315	7.2661	0.0070	2.232E8
Treatment	A	1	0.8483	0.5502	2.3773	0.1231	2.336
Treatment	B	1	1.4949	0.6622	5.0956	0.0240	4.459
Sex	F	1	0.9173	0.3981	5.3104	0.0212	2.503
Treatment*Sex	A F	1	−0.2010	0.5568	0.1304	0.7180	0.818
Treatment*Sex	B F	1	0.0487	0.5563	0.0077	0.9302	1.050
Age		1	−0.2688	0.0996	7.2744	0.0070	0.764
Duration		1	0.00523	0.0333	0.0247	0.8752	1.005

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
Age	0.764	0.629	0.929
Duration	1.005	0.942	1.073

The following PROC LOGISTIC statements illustrate the use of forward selection on the data set Neuralgia to identify the effects that differentiate the two Pain responses. The option SELECTION=FORWARD is specified to carry out the forward selection. The term Treatment|Sex@2 illustrates another way to specify main effects and two-way interactions. (Note that, in this case, the "@2" is unnecessary because no interactions besides the two-way interaction are possible).

```
proc logistic data=Neuralgia;
   class Treatment Sex;
   model Pain=Treatment|Sex@2 Age Duration
         /selection=forward expb;
run;
```

Results of the forward selection process are summarized in Output 51.2.4. The variable Treatment is selected first, followed by Age and then Sex. The results are consistent with the previous analysis (Output 51.2.2) in which the Treatment*Sex interaction and Duration are not statistically significant.

Output 51.2.4 Effects Selected into the Model

Summary of Forward Selection

Step	Effect Entered	DF	Number In	Score Chi-Square	Pr > ChiSq
1	Treatment	2	1	13.7143	0.0011
2	Age	1	2	10.6038	0.0011
3	Sex	1	3	5.9959	0.0143

Output 51.2.5 shows the Type 3 analysis of effects, the parameter estimates, and the odds ratio estimates for the selected model. All three variables, Treatment, Age, and Sex, are statistically significant at the 0.05 level (p=0.0018, p=0.0213, and p=0.0057, respectively). Since the selected model does not contain the Treatment*Sex interaction, odds ratios for Treatment and Sex are computed. The estimated odds ratio is 24.022 for treatment A versus placebo, 41.528 for Treatment B versus placebo, and 6.194 for female patients versus male patients. Note that these odds ratio estimates are not the same as the corresponding values in the Exp(Est) column in the parameter estimates table because effect coding was used. From Output 51.2.5, it is evident that both Treatment A and Treatment B are better than the placebo in reducing pain; females tend to have better improvement than males; and younger patients are faring better than older patients.

Output 51.2.5 Type 3 Effects and Parameter Estimates with Effect Coding

Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
Treatment	2	12.6928	0.0018
Sex	1	5.3013	0.0213
Age	1	7.6314	0.0057

Analysis of Maximum Likelihood Estimates

Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq	Exp(Est)
Intercept		1	19.0804	6.7882	7.9007	0.0049	1.9343E8
Treatment	A	1	0.8772	0.5274	2.7662	0.0963	2.404
Treatment	B	1	1.4246	0.6036	5.5711	0.0183	4.156
Sex	F	1	0.9118	0.3960	5.3013	0.0213	2.489
Age		1	-0.2650	0.0959	7.6314	0.0057	0.767

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
Treatment A vs P	24.022	3.295	175.121
Treatment B vs P	41.528	4.500	383.262
Sex F vs M	6.194	1.312	29.248
Age	0.767	0.636	0.926

Finally, the following statements refit the previously selected model, except that reference coding is used for the CLASS variables instead of effect coding. The ODDSRATIO statements compute the odds ratios for the covariates. The ODS GRAPHICS statement and the PLOTS option are specified to produce plots of the results; the PLOTS(ONLY)= option specifies that only the requested plots of the oddsratios and the fitted model are displayed. Two CONTRAST statements are specified, and provide another method of producing the odds ratios. The contrast labeled 'Pairwise' specifies three rows in the contrast matrix, L, for all the pairwise comparisons between the three levels of Treatment. The contrast labeled 'Female vs Male' compares female to male patients. The option ESTIMATE=EXP is specified in both CONTRAST statements to exponentiate the estimates of $\mathbf{L}'\boldsymbol{\beta}$. With the given specification of contrast coefficients, the first row of the 'Pairwise' CONTRAST statement corresponds to the odds ratio of A versus P, the second row corresponds to B versus P, and the third row corresponds to A versus B. There is only one row in the 'Female vs Male' CONTRAST statement, and it corresponds to the odds ratio comparing female to male patients.

```
ods graphics on;
proc logistic data=Neuralgia plots(only)=(oddsratio(range=clip) effect);
   class Treatment Sex /param=ref;
   model Pain= Treatment Sex age;
   oddsratio Treatment;
   oddsratio Sex;
   oddsratio age;
   contrast 'Pairwise' Treatment 1  0,
                       Treatment 0  1,
                       Treatment 1 -1 / estimate=exp;
   contrast 'Female vs Male' Sex 1 / estimate=exp;
run;
ods graphics off;
```

The reference coding is shown in Output 51.2.6. The Type 3 analysis of effects, the parameter estimates for the reference coding, and the odds ratio estimates are displayed in Output 51.2.7. Although the parameter estimates are different because of the different parameterizations, the "Type 3 Analysis of Effects" table and the "Odds Ratio" table remain the same as in Output 51.2.5. With effect coding, the treatment A parameter estimate (0.8772) estimates the effect of treatment A compared to the average effect of treatments A, B, and placebo. The treatment A estimate (3.1790) under the reference coding estimates the difference in effect of treatment A and the placebo treatment.

Output 51.2.6 Reference Coding of CLASS Variables

Class Level Information

Class	Value	Design Variables	
Treatment	A	1	0
	B	0	1
	P	0	0
Sex	F	1	
	M	0	

Output 51.2.7 Type 3 Effects and Parameter Estimates with Reference Coding

Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
Treatment	2	12.6928	0.0018
Sex	1	5.3013	0.0213
Age	1	7.6314	0.0057

Analysis of Maximum Likelihood Estimates

Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		1	15.8669	6.4056	6.1357	0.0132
Treatment A		1	3.1790	1.0135	9.8375	0.0017
Treatment B		1	3.7264	1.1339	10.8006	0.0010
Sex	F	1	1.8235	0.7920	5.3013	0.0213
Age		1	-0.2650	0.0959	7.6314	0.0057

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
Treatment A vs P	24.022	3.295	175.121
Treatment B vs P	41.528	4.500	383.262
Sex F vs M	6.194	1.312	29.248
Age	0.767	0.636	0.926

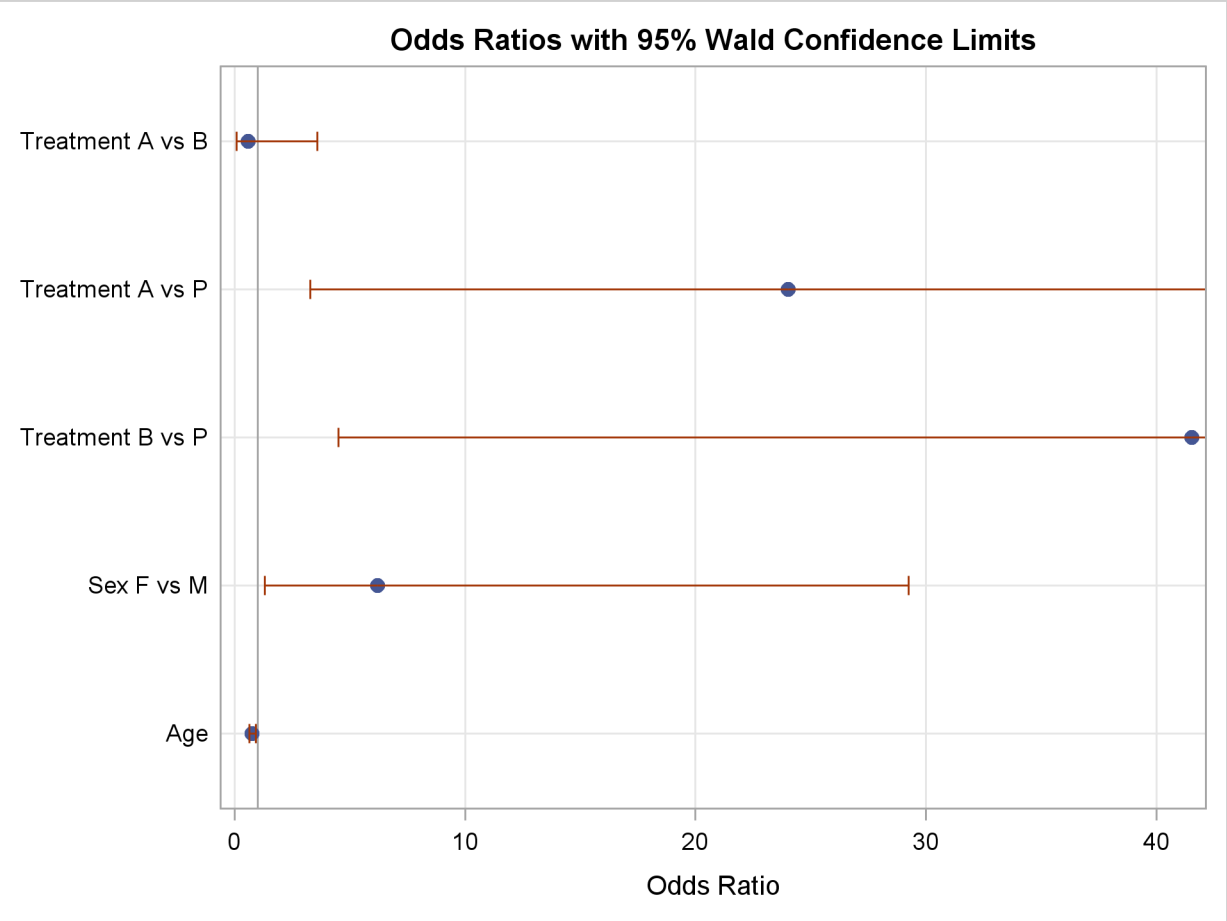
The ODDSRATIO statement results are shown in Output 51.2.8, and the resulting plot is displayed in Output 51.2.9. Note in Output 51.2.9 that the odds ratio confidence limits are truncated due to specifying the RANGE=CLIP option; this enables you to see which intervals contain "1" more clearly. The odds ratios are identical to those shown in the "Odds Ratio Estimates" table in Output 51.2.7 with the addition of the odds ratio for "Treatment A vs B". Both treatments A and B are highly effective over placebo in reducing pain, as can be seen from the odds ratios comparing treatment A against P and treatment B against P (the second and third rows in the table). However, the 95% confidence interval for the odds ratio comparing treatment A to B is (0.0932, 3.5889), indicating that the pain reduction effects of these two test treatments are not very different. Again, the 'Sex F vs M' odds ratio shows that female patients fared better in obtaining relief from pain than male patients. The odds ratio for age shows that a patient one year older is 0.77 times as likely to show no pain; that is, younger patients have more improvement than older patients.

Output 51.2.8 Results from the ODDSRATIO Statements

Wald Confidence Interval for Odds Ratios

Label	Estimate	95% Confidence Limits	
Treatment A vs B	0.578	0.093	3.589
Treatment A vs P	24.022	3.295	175.121
Treatment B vs P	41.528	4.500	383.262
Sex F vs M	6.194	1.312	29.248
Age	0.767	0.636	0.926

Output 51.2.9 Plot of the ODDSRATIO Statement Results



Output 51.2.10 contains two tables: the "Contrast Test Results" table and the "Contrast Rows Estimation and Testing Results" table. The former contains the overall Wald test for each CONTRAST statement. Although three rows are specified in the 'Pairwise' CONTRAST statement, there are only two degrees of freedom, and the Wald test result is identical to the Type 3 analysis of Treatment in Output 51.2.7. The latter table contains estimates and tests of individual contrast rows. The estimates for the first two rows of the 'Pairwise' CONTRAST statements are the same as those given in the two preceding odds ratio tables (Output 51.2.7 and Output 51.2.8). The third row estimates the odds ratio comparing A to B, agreeing with Output 51.2.8, and the last row computes the odds

ratio comparing pain relief for females to that for males.

Output 51.2.10 Results of CONTRAST Statements

Contrast Test Results

Contrast	DF	Wald Chi-Square	Pr > ChiSq
Pairwise	2	12.6928	0.0018
Female vs Male	1	5.3013	0.0213

Contrast Rows Estimation and Testing Results

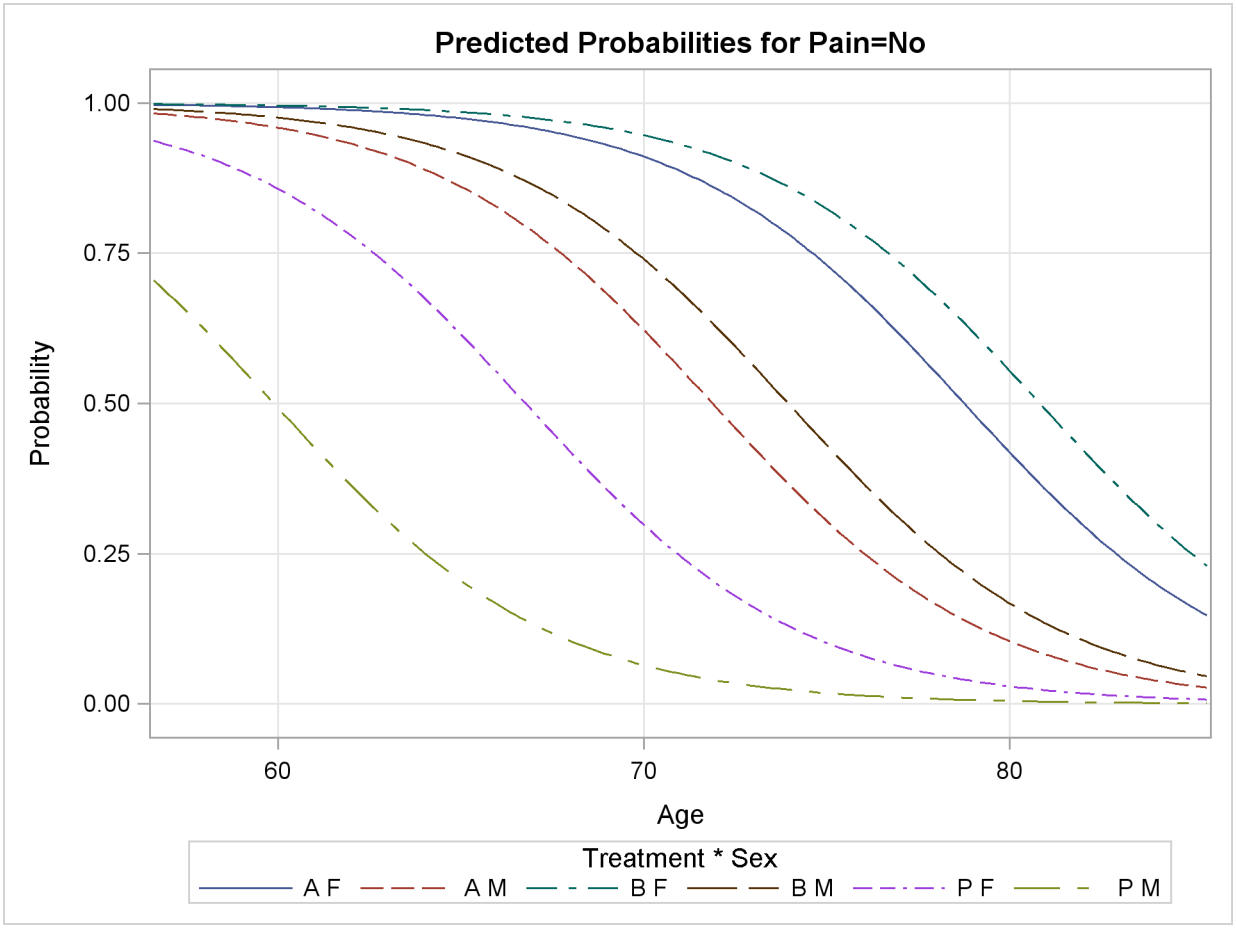
Contrast	Type	Row	Estimate	Standard Error	Alpha	Confidence Limits	
Pairwise	EXP	1	24.0218	24.3473	0.05	3.2951	175.1
Pairwise	EXP	2	41.5284	47.0877	0.05	4.4998	383.3
Pairwise	EXP	3	0.5784	0.5387	0.05	0.0932	3.5889
Female vs Male	EXP	1	6.1937	4.9053	0.05	1.3116	29.2476

Contrast Rows Estimation and Testing Results

Contrast	Type	Row	Wald Chi-Square	Pr > ChiSq
Pairwise	EXP	1	9.8375	0.0017
Pairwise	EXP	2	10.8006	0.0010
Pairwise	EXP	3	0.3455	0.5567
Female vs Male	EXP	1	5.3013	0.0213

An ANCOVA-style plot of the model-predicted probabilities against the Age variable for each combination of Treatment and Sex is displayed in Output 51.2.11. This plot confirms that females always have a higher probability of pain reduction in each treatment group, the placebo treatment has a lower probability of success than the other treatments, and younger patients respond to treatment better than older patients.

Output 51.2.11 Model-Predicted Probabilities



Example 51.3: Ordinal Logistic Regression

Consider a study of the effects on taste of various cheese additives. Researchers tested four cheese additives and obtained 52 response ratings for each additive. Each response was measured on a scale of nine categories ranging from strong dislike (1) to excellent taste (9). The data, given in McCullagh and Nelder (1989, p. 175) in the form of a two-way frequency table of additive by rating, are saved in the data set Cheese by using the following program. The variable y contains the response rating. The variable Additive specifies the cheese additive (1, 2, 3, or 4). The variable freq gives the frequency with which each additive received each rating.

```
data Cheese;
   do Additive = 1 to 4;
      do y = 1 to 9;
         input freq @@;
         output;
      end;
   end;
   label y='Taste Rating';
   datalines;
0  0  1  7  8  8 19  8  1
6  9 12 11  7  6  1  0  0
1  1  6  8 23  7  5  1  0
0  0  0  1  3  7 14 16 11
;
```

The response variable y is ordinally scaled. A cumulative logit model is used to investigate the effects of the cheese additives on taste. The following statements invoke PROC LOGISTIC to fit this model with y as the response variable and three indicator variables as explanatory variables, with the fourth additive as the reference level. With this parameterization, each Additive parameter compares an additive to the fourth additive. The COVB option displays the estimated covariance matrix. The ODDSRATIO statement computes odds ratios for all combinations of the Additive levels. The PLOTS(ONLY)= option produces a graphical display of the predicted probabilities and the odds ratios.

```
ods graphics on;
proc logistic data=Cheese plots(only)=(effect(polybar) oddsratio(range=clip));
   freq freq;
   class Additive (param=ref ref='4');
   model y=Additive / covb;
   oddsratio Additive;
   title 'Multiple Response Cheese Tasting Experiment';
run;
ods graphics off;
```

The "Response Profile" table in Output 51.3.1 shows that the strong dislike (y=1) end of the rating scale is associated with lower Ordered Values in the "Response Profile" table; hence the probability of disliking the additives is modeled.

The score chi-square for testing the proportional odds assumption is 17.287, which is not significant with respect to a chi-square distribution with 21 degrees of freedom ($p = 0.694$). This indicates

that the proportional odds assumption is reasonable. The positive value (1.6128) for the parameter estimate for Additive1 indicates a tendency toward the lower-numbered categories of the first cheese additive relative to the fourth. In other words, the fourth additive tastes better than the first additive. The second and third additives are both less favorable than the fourth additive. The relative magnitudes of these slope estimates imply the preference ordering: fourth, first, third, second.

Output 51.3.1 Proportional Odds Model Regression Analysis

Multiple Response Cheese Tasting Experiment

Model Information

Data Set	WORK.CHEESE	
Response Variable	y	Taste Rating
Number of Response Levels	9	
Frequency Variable	freq	
Model	cumulative logit	
Optimization Technique	Fisher's scoring	

Number of Observations Read	36
Number of Observations Used	28
Sum of Frequencies Read	208
Sum of Frequencies Used	208

Response Profile

Ordered Value	y	Total Frequency
1	1	7
2	2	10
3	3	19
4	4	27
5	5	41
6	6	28
7	7	39
8	8	25
9	9	12

Probabilities modeled are cumulated over the lower Ordered Values.

NOTE: 8 observations having nonpositive frequencies or weights were excluded since they do not contribute to the analysis.

Class Level Information

Class	Value	Design Variables		
Additive	1	1	0	0
	2	0	1	0
	3	0	0	1
	4	0	0	0

Output 51.3.1 *continued*

Model Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Score Test for the Proportional Odds Assumption

Chi-Square	DF	Pr > ChiSq
17.2866	21	0.6936

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	875.802	733.348
SC	902.502	770.061
-2 Log L	859.802	711.348

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	148.4539	3	<.0001
Score	111.2670	3	<.0001
Wald	115.1504	3	<.0001

Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
Additive	3	115.1504	<.0001

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept 1	1	-7.0801	0.5624	158.4851	<.0001
Intercept 2	1	-6.0249	0.4755	160.5500	<.0001
Intercept 3	1	-4.9254	0.4272	132.9484	<.0001
Intercept 4	1	-3.8568	0.3902	97.7087	<.0001
Intercept 5	1	-2.5205	0.3431	53.9704	<.0001
Intercept 6	1	-1.5685	0.3086	25.8374	<.0001
Intercept 7	1	-0.0669	0.2658	0.0633	0.8013
Intercept 8	1	1.4930	0.3310	20.3439	<.0001
Additive 1	1	1.6128	0.3778	18.2265	<.0001
Additive 2	1	4.9645	0.4741	109.6427	<.0001
Additive 3	1	3.3227	0.4251	61.0931	<.0001

Output 51.3.1 *continued*

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
Additive 1 vs 4	5.017	2.393	10.520
Additive 2 vs 4	143.241	56.558	362.777
Additive 3 vs 4	27.734	12.055	63.805

Association of Predicted Probabilities and Observed Responses

Percent Concordant	67.6	Somers' D	0.578
Percent Discordant	9.8	Gamma	0.746
Percent Tied	22.6	Tau-a	0.500
Pairs	18635	c	0.789

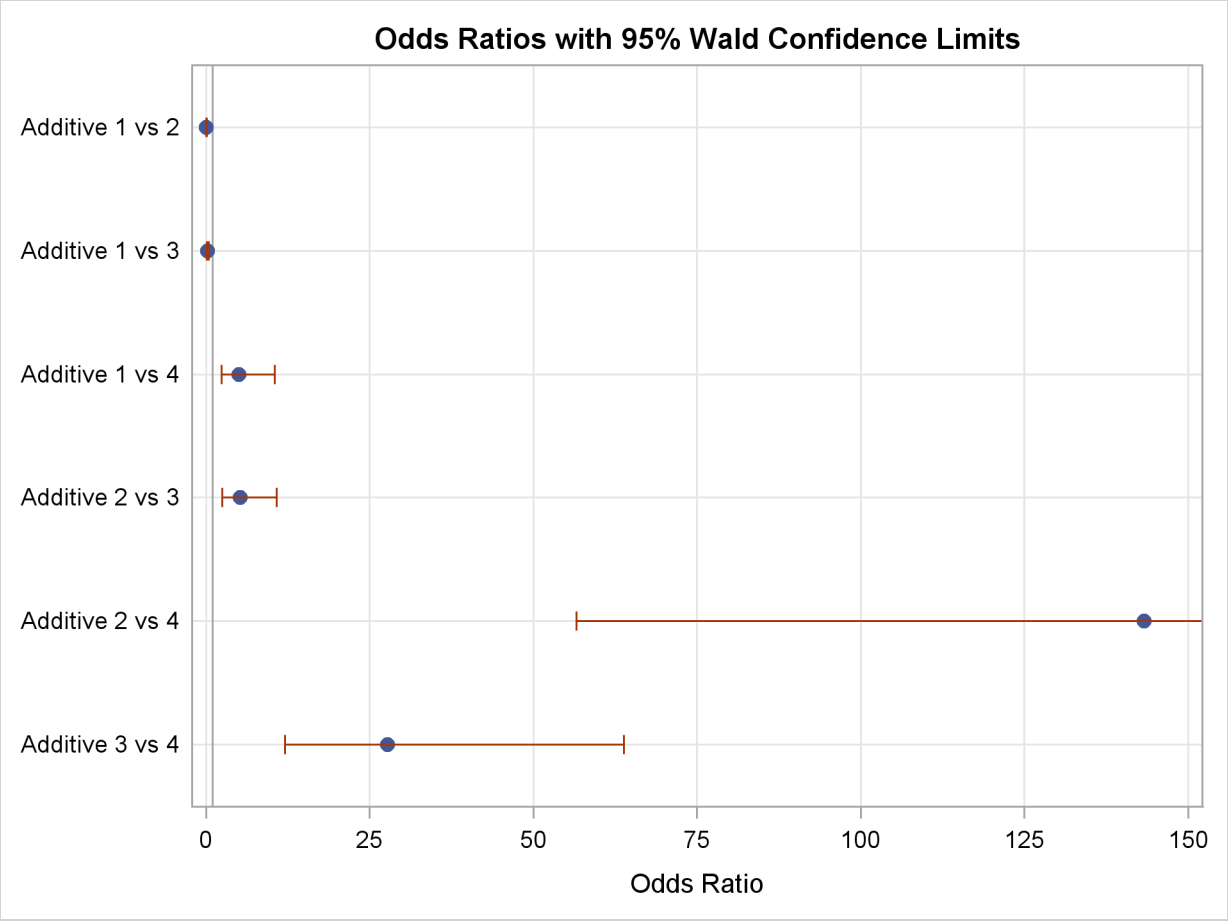
The odds ratio results in Output 51.3.2 show the preferences more clearly. For example, the "Additive 1 vs 4" odds ratio says that the first additive has 5.017 times the odds of receiving a lower score than the fourth additive; that is, the first additive is 5.017 times more likely than the fourth additive to receive a lower score. Output 51.3.3 displays the odds ratios graphically; the range of the confidence limits is truncated by the RANGE=CLIP option, so you can see that "1" is not contained in any of the intervals.

Output 51.3.2 Odds Ratios of All Pairs of Additive Levels

Wald Confidence Interval for Odds Ratios

Label	Estimate	95% Confidence Limits	
Additive 1 vs 2	0.035	0.015	0.080
Additive 1 vs 3	0.181	0.087	0.376
Additive 1 vs 4	5.017	2.393	10.520
Additive 2 vs 3	5.165	2.482	10.746
Additive 2 vs 4	143.241	56.558	362.777
Additive 3 vs 4	27.734	12.055	63.805

Output 51.3.3 Plot of Odds Ratios for Additive



The estimated covariance matrix of the parameters is displayed in Output 51.3.4.

Output 51.3.4 Estimated Covariance Matrix

Estimated Covariance Matrix

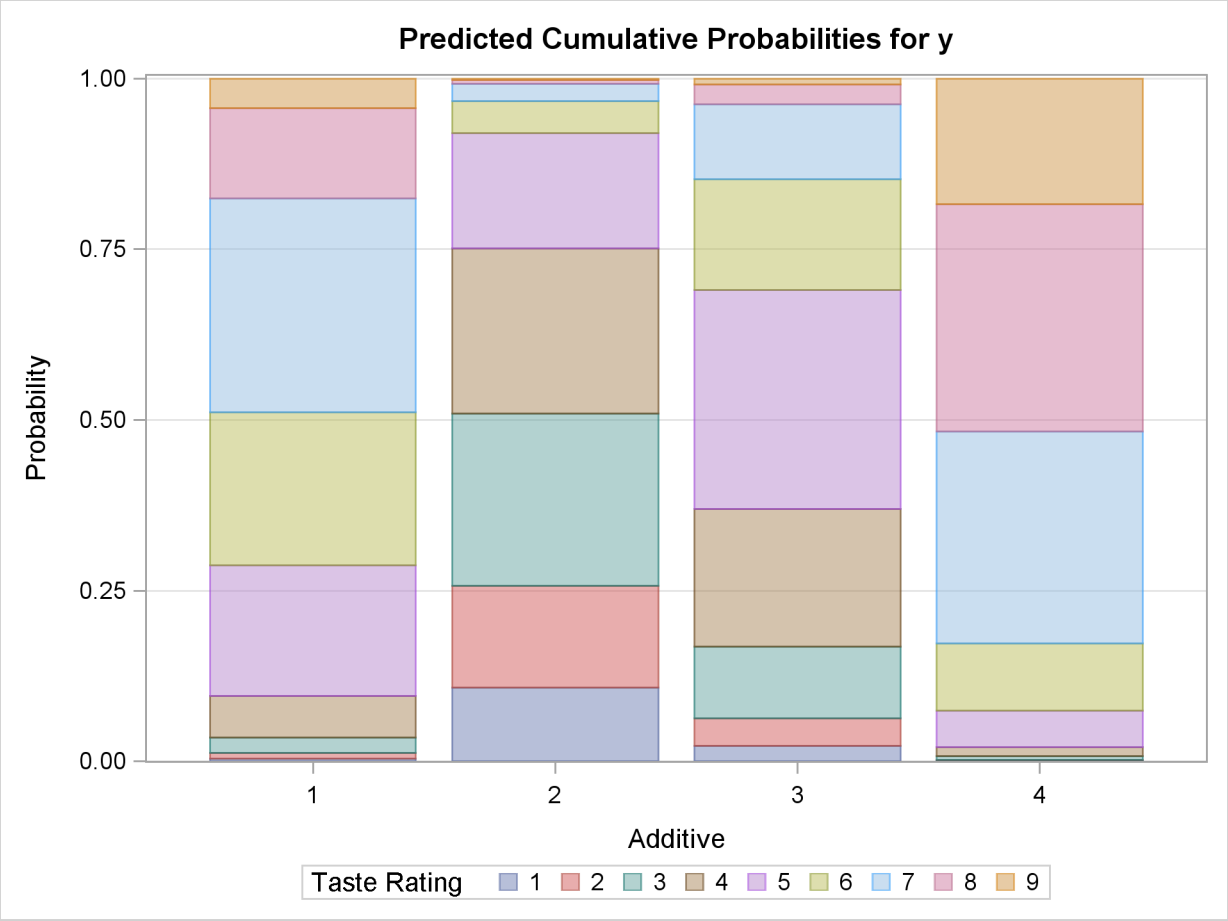
Parameter	Intercept_ 1	Intercept_ 2	Intercept_ 3	Intercept_ 4	Intercept_ 5
Intercept_1	0.316291	0.219581	0.176278	0.147694	0.114024
Intercept_2	0.219581	0.226095	0.177806	0.147933	0.11403
Intercept_3	0.176278	0.177806	0.182473	0.148844	0.114092
Intercept_4	0.147694	0.147933	0.148844	0.152235	0.114512
Intercept_5	0.114024	0.11403	0.114092	0.114512	0.117713
Intercept_6	0.091085	0.091081	0.091074	0.091109	0.091821
Intercept_7	0.057814	0.057813	0.057807	0.05778	0.057721
Intercept_8	0.041304	0.041304	0.0413	0.041277	0.041162
Additive1	−0.09419	−0.09421	−0.09427	−0.09428	−0.09246
Additive2	−0.18686	−0.18161	−0.1687	−0.14717	−0.11415
Additive3	−0.13565	−0.13569	−0.1352	−0.13118	−0.11207

Estimated Covariance Matrix

Parameter	Intercept_ 6	Intercept_ 7	Intercept_ 8	Additive1	Additive2	Additive3
Intercept_1	0.091085	0.057814	0.041304	−0.09419	−0.18686	−0.13565
Intercept_2	0.091081	0.057813	0.041304	−0.09421	−0.18161	−0.13569
Intercept_3	0.091074	0.057807	0.0413	−0.09427	−0.1687	−0.1352
Intercept_4	0.091109	0.05778	0.041277	−0.09428	−0.14717	−0.13118
Intercept_5	0.091821	0.057721	0.041162	−0.09246	−0.11415	−0.11207
Intercept_6	0.09522	0.058312	0.041324	−0.08521	−0.09113	−0.09122
Intercept_7	0.058312	0.07064	0.04878	−0.06041	−0.05781	−0.05802
Intercept_8	0.041324	0.04878	0.109562	−0.04436	−0.0413	−0.04143
Additive1	−0.08521	−0.06041	−0.04436	0.142715	0.094072	0.092128
Additive2	−0.09113	−0.05781	−0.0413	0.094072	0.22479	0.132877
Additive3	−0.09122	−0.05802	−0.04143	0.092128	0.132877	0.180709

Output 51.3.5 displays the probability of each taste rating y within each additive. You can see that Additive=1 mostly receives ratings of 5 to 7, Additive=2 mostly receives ratings of 2 to 5, Additive=3 mostly receives ratings of 4 to 6, and Additive=4 mostly receives ratings of 7 to 9, which also confirms the previously discussed preference orderings.

Output 51.3.5 Model-Predicted Probabilities



Example 51.4: Nominal Response Data: Generalized Logits Model

Over the course of one school year, third graders from three different schools are exposed to three different styles of mathematics instruction: a self-paced computer-learning style, a team approach, and a traditional class approach. The students are asked which style they prefer and their responses, classified by the type of program they are in (a regular school day versus a regular day supplemented with an afternoon school program), are displayed in Table 51.7. The data set is from Stokes, Davis, and Koch (2000), and is also analyzed in the section "Generalized Logits Model" on page 1103 of Chapter 28, "The CATMOD Procedure."

Table 51.7 School Program Data

		Learning Style Preference		
School	**Program**	**Self**	**Team**	**Class**
1	Regular	10	17	26
1	Afternoon	5	12	50
2	Regular	21	17	26
2	Afternoon	16	12	36
3	Regular	15	15	16
3	Afternoon	12	12	20

The levels of the response variable (self, team, and class) have no essential ordering, so a logistic regression is performed on the generalized logits. The model to be fit is

$$\log\left(\frac{\pi_{hij}}{\pi_{hir}}\right) = \alpha_j + \boldsymbol{x}'_{hi}\boldsymbol{\beta}_j$$

where π_{hij} is the probability that a student in school h and program i prefers teaching style j, $j \neq r$, and style r is the baseline style (in this case, class). There are separate sets of intercept parameters α_j and regression parameters $\boldsymbol{\beta}_j$ for each logit, and the vector $\boldsymbol{x}_{hi}$ is the set of explanatory variables for the hith population. Thus, two logits are modeled for each school and program combination: the logit comparing self to class and the logit comparing team to class.

The following statements create the data set school and request the analysis. The LINK=GLOGIT option forms the generalized logits. The response variable option ORDER=DATA means that the response variable levels are ordered as they exist in the data set: self, team, and class; thus, the logits are formed by comparing self to class and by comparing team to class. The ODDSRATIO statement produces odds ratios in the presence of interactions, and the ODS GRAPHICS statements produces a graphical display of the requested odds ratios.

```
data school;
   length Program $ 9;
   input School Program $ Style $ Count @@;
   datalines;
1 regular   self 10  1 regular   team 17  1 regular   class 26
1 afternoon self  5  1 afternoon team 12  1 afternoon class 50
2 regular   self 21  2 regular   team 17  2 regular   class 26
2 afternoon self 16  2 afternoon team 12  2 afternoon class 36
3 regular   self 15  3 regular   team 15  3 regular   class 16
3 afternoon self 12  3 afternoon team 12  3 afternoon class 20
;

ods graphics on;
proc logistic data=school;
   freq Count;
   class School Program(ref=first);
   model Style(order=data)=School Program School*Program / link=glogit;
   oddsratio program;
run;
ods graphics off;
```

Summary information about the model, the response variable, and the classification variables are displayed in Output 51.4.1.

Output 51.4.1 Analysis of Saturated Model

```
                              Model Information

                Data Set                      WORK.SCHOOL
                Response Variable             Style
                Number of Response Levels     3
                Frequency Variable            Count
                Model                         generalized logit
                Optimization Technique        Newton-Raphson

                    Number of Observations Read          18
                    Number of Observations Used          18
                    Sum of Frequencies Read             338
                    Sum of Frequencies Used             338

                               Response Profile

                       Ordered                      Total
                         Value     Style        Frequency

                             1     self                79
                             2     team                85
                             3     class              174

          Logits modeled use Style='class' as the reference category.
```

Output 51.4.1 *continued*

Class Level Information

Class	Value	Design Variables	
School	1	1	0
	2	0	1
	3	−1	−1
Program	afternoon	−1	
	regular	1	

Model Convergence Status

Convergence criterion (GCONV=1E−8) satisfied.

The "Testing Global Null Hypothesis: BETA=0" table in Output 51.4.2 shows that the parameters are significantly different from zero.

Output 51.4.2 Analysis of Saturated Model

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	699.404	689.156
SC	707.050	735.033
−2 Log L	695.404	665.156

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	30.2480	10	0.0008
Score	28.3738	10	0.0016
Wald	25.6828	10	0.0042

However, the "Type 3 Analysis of Effects" table in Output 51.4.3 shows that the interaction effect is clearly nonsignificant.

Output 51.4.3 Analysis of Saturated Model

Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
School	4	14.5522	0.0057
Program	2	10.4815	0.0053
School*Program	4	1.7439	0.7827

Analysis of Maximum Likelihood Estimates

Parameter			Style	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept			self	1	−0.8097	0.1488	29.5989	<.0001
Intercept			team	1	−0.6585	0.1366	23.2449	<.0001
School	1		self	1	−0.8194	0.2281	12.9066	0.0003
School	1		team	1	−0.2675	0.1881	2.0233	0.1549
School	2		self	1	0.2974	0.1919	2.4007	0.1213
School	2		team	1	−0.1033	0.1898	0.2961	0.5863
Program	regular		self	1	0.3985	0.1488	7.1684	0.0074
Program	regular		team	1	0.3537	0.1366	6.7071	0.0096
School*Program	1	regular	self	1	0.2751	0.2281	1.4547	0.2278
School*Program	1	regular	team	1	0.1474	0.1881	0.6143	0.4332
School*Program	2	regular	self	1	−0.0998	0.1919	0.2702	0.6032
School*Program	2	regular	team	1	−0.0168	0.1898	0.0079	0.9293

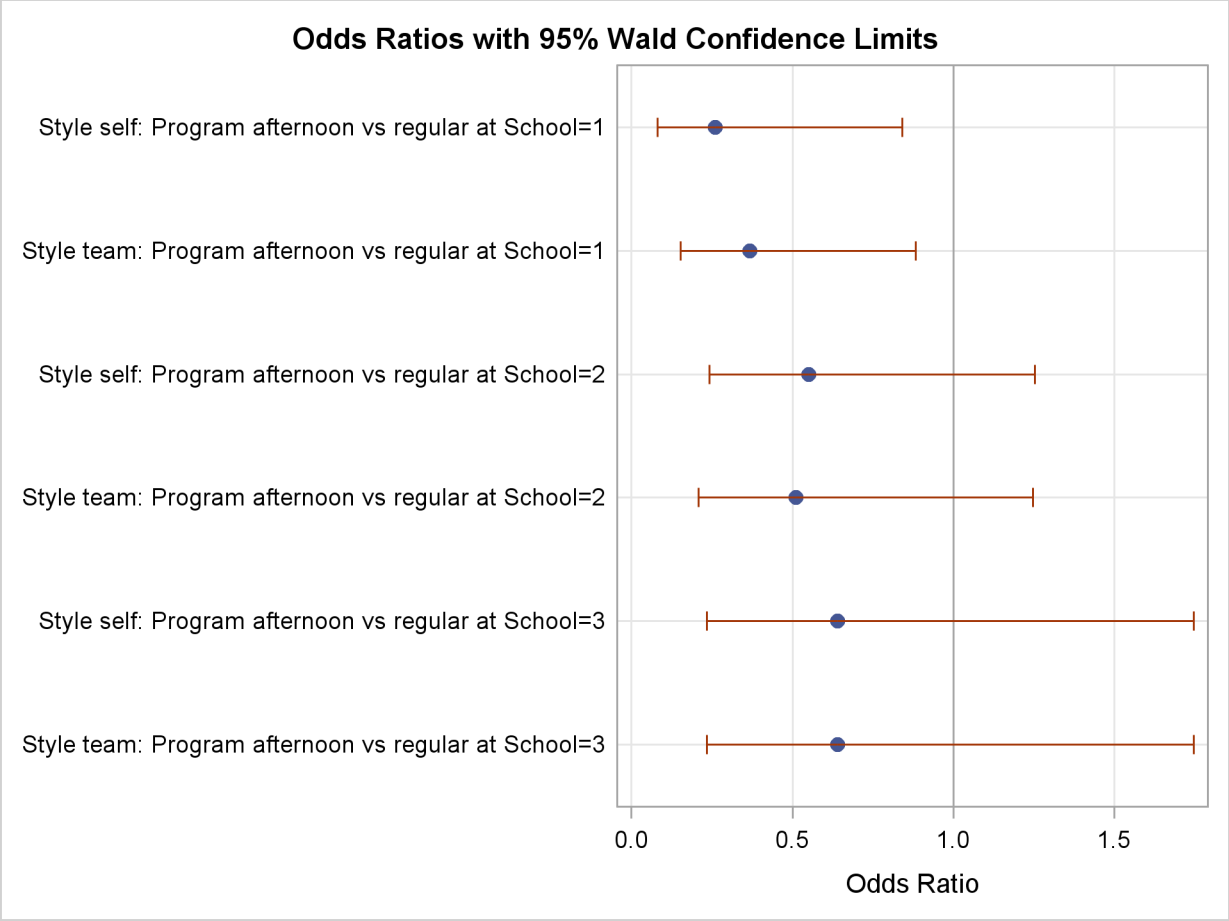
The table produced by the ODDSRATIO statement is displayed in Output 51.4.4. The differences between the program preferences are small across all the styles (logits) compared to their variability as displayed by the confidence limits in Output 51.4.5, confirming that the interaction effect is nonsignificant.

Output 51.4.4 Odds Ratios for Style

Wald Confidence Interval for Odds Ratios

Label	Estimate	95% Confidence Limits	
Style self: Program afternoon vs regular at School=1	0.260	0.080	0.841
Style team: Program afternoon vs regular at School=1	0.367	0.153	0.883
Style self: Program afternoon vs regular at School=2	0.550	0.242	1.253
Style team: Program afternoon vs regular at School=2	0.510	0.208	1.247
Style self: Program afternoon vs regular at School=3	0.640	0.234	1.747
Style team: Program afternoon vs regular at School=3	0.640	0.234	1.747

Output 51.4.5 Plot of Odds Ratios for Style



Since the interaction effect is clearly nonsignificant, a main-effects model is fit with the following statements:

```
ods graphics on;
proc logistic data=school plots(only)=effect(clbar connect);
   freq Count;
   class School Program(ref=first);
   model Style(order=data)=School Program / link=glogit;
run;
ods graphics off;
```

All of the global fit tests in Output 51.4.6 suggest the model is significant, and the Type 3 tests show that the school and program effects are also significant.

Output 51.4.6 Analysis of Main-Effects Model

Model Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	699.404	682.934
SC	707.050	713.518
-2 Log L	695.404	666.934

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	28.4704	6	<.0001
Score	27.1190	6	0.0001
Wald	25.5881	6	0.0003

Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
School	4	14.8424	0.0050
Program	2	10.9160	0.0043

The parameter estimates, tests for individual parameters, and odds ratios are displayed in Output 51.4.7. The Program variable has nearly the same effect on both logits, while School=1 has the largest effect of the schools.

Output 51.4.7 Estimates

Analysis of Maximum Likelihood Estimates

Parameter		Style	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		self	1	-0.7978	0.1465	29.6502	<.0001
Intercept		team	1	-0.6589	0.1367	23.2300	<.0001
School	1	self	1	-0.7992	0.2198	13.2241	0.0003
School	1	team	1	-0.2786	0.1867	2.2269	0.1356
School	2	self	1	0.2836	0.1899	2.2316	0.1352
School	2	team	1	-0.0985	0.1892	0.2708	0.6028
Program	regular	self	1	0.3737	0.1410	7.0272	0.0080
Program	regular	team	1	0.3713	0.1353	7.5332	0.0061

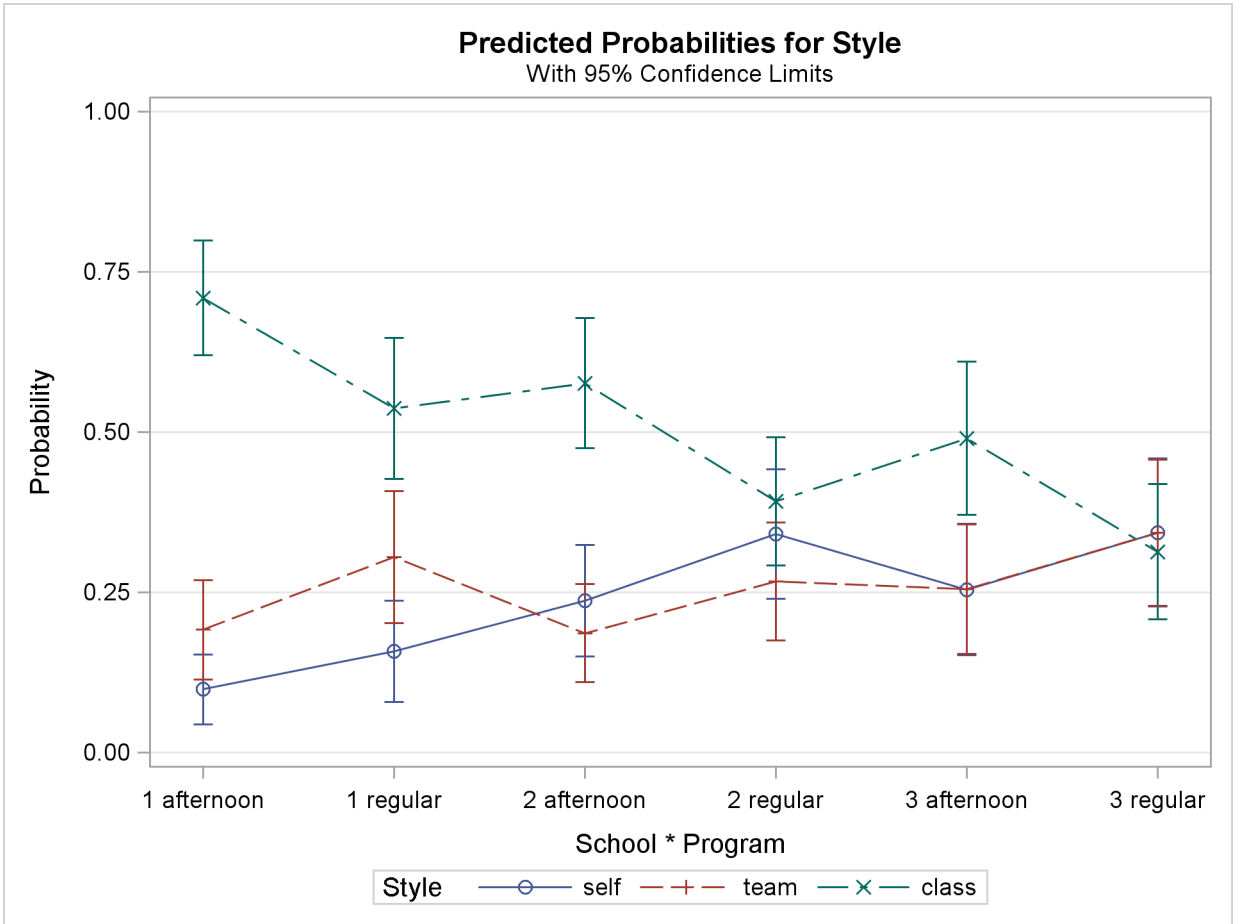
Output 51.4.7 *continued*

Odds Ratio Estimates

Effect	Style	Point Estimate	95% Wald Confidence Limits	
School 1 vs 3	self	0.269	0.127	0.570
School 1 vs 3	team	0.519	0.267	1.010
School 2 vs 3	self	0.793	0.413	1.522
School 2 vs 3	team	0.622	0.317	1.219
Program regular vs afternoon	self	2.112	1.215	3.670
Program regular vs afternoon	team	2.101	1.237	3.571

The interaction plot in Output 51.4.8 shows that School=1 and Program=afternoon has a preference for the traditional classroom style. Of course, since these are not simultaneous confidence intervals, the nonoverlapping 95% confidence limits do not take the place of an actual test.

Output 51.4.8 Model-Predicted Probabilities



Example 51.5: Stratified Sampling

Consider the hypothetical example in Fleiss (1981, pp. 6–7), in which a test is applied to a sample of 1,000 people known to have a disease and to another sample of 1,000 people known not to have the same disease. In the diseased sample, 950 test positive; in the nondiseased sample, only 10 test positive. If the true disease rate in the population is 1 in 100, specifying PEVENT=0.01 results in the correct false positive and negative rates for the stratified sampling scheme. Omitting the PEVENT= option is equivalent to using the overall sample disease rate (1000/2000 = 0.5) as the value of the PEVENT= option, which would ignore the stratified sampling.

The statements to create the data set and perform the analysis are as follows:

```
data Screen;
   do Disease='Present','Absent';
      do Test=1,0;
         input Count @@;
         output;
      end;
   end;
   datalines;
950  50
 10 990
;

proc logistic data=Screen;
   freq Count;
   model Disease(event='Present')=Test
         / pevent=.5 .01 ctable pprob=.5;
run;
```

The response variable option EVENT= indicates that Disease='Present' is the event. The CTABLE option is specified to produce a classification table. Specifying PPROB=0.5 indicates a cutoff probability of 0.5. A list of two probabilities, 0.5 and 0.01, is specified for the PEVENT= option; 0.5 corresponds to the overall sample disease rate, and 0.01 corresponds to a true disease rate of 1 in 100.

The classification table is shown in Output 51.5.1.

Output 51.5.1 False Positive and False Negative Rates

Classification Table

		Correct		Incorrect		Percentages				
Prob Event	Prob Level	Event	Non-Event	Event	Non-Event	Correct	Sensi-tivity	Speci-ficity	False POS	False NEG
0.500	0.500	950	990	10	50	97.0	95.0	99.0	1.0	4.8
0.010	0.500	950	990	10	50	99.0	95.0	99.0	51.0	0.1

In the classification table, the column "Prob Level" represents the cutoff values (the settings of the PPROB= option) for predicting whether an observation is an event. The "Correct" columns list the numbers of subjects that are correctly predicted as events and nonevents, respectively, and the "Incorrect" columns list the number of nonevents incorrectly predicted as events and the number of events incorrectly predicted as nonevents, respectively. For PEVENT=0.5, the false positive rate is 1% and the false negative rate is 4.8%. These results ignore the fact that the samples were stratified and incorrectly assume that the overall sample proportion of disease (which is 0.5) estimates the true disease rate. For a true disease rate of 0.01, the false positive rate and the false negative rate are 51% and 0.1%, respectively, as shown in the second line of the classification table.

Example 51.6: Logistic Regression Diagnostics

In a controlled experiment to study the effect of the rate and volume of air intake on a transient reflex vasoconstriction in the skin of the digits, 39 tests under various combinations of rate and volume of air intake were obtained (Finney 1947). The endpoint of each test is whether or not vasoconstriction occurred. Pregibon (1981) uses this set of data to illustrate the diagnostic measures he proposes for detecting influential observations and to quantify their effects on various aspects of the maximum likelihood fit.

The vasoconstriction data are saved in the data set vaso:

```
data vaso;
   length Response $12;
   input Volume Rate Response @@;
   LogVolume=log(Volume);
   LogRate=log(Rate);
   datalines;
3.70  0.825  constrict       3.50  1.09   constrict
1.25  2.50   constrict       0.75  1.50   constrict
0.80  3.20   constrict       0.70  3.50   constrict
0.60  0.75   no_constrict    1.10  1.70   no_constrict
0.90  0.75   no_constrict    0.90  0.45   no_constrict
0.80  0.57   no_constrict    0.55  2.75   no_constrict
0.60  3.00   no_constrict    1.40  2.33   constrict
0.75  3.75   constrict       2.30  1.64   constrict
3.20  1.60   constrict       0.85  1.415  constrict
1.70  1.06   no_constrict    1.80  1.80   constrict
0.40  2.00   no_constrict    0.95  1.36   no_constrict
1.35  1.35   no_constrict    1.50  1.36   no_constrict
1.60  1.78   constrict       0.60  1.50   no_constrict
1.80  1.50   constrict       0.95  1.90   no_constrict
1.90  0.95   constrict       1.60  0.40   no_constrict
2.70  0.75   constrict       2.35  0.03   no_constrict
1.10  1.83   no_constrict    1.10  2.20   constrict
1.20  2.00   constrict       0.80  3.33   constrict
0.95  1.90   no_constrict    0.75  1.90   no_constrict
1.30  1.625  constrict
;
```

In the data set vaso, the variable Response represents the outcome of a test. The variable LogVolume represents the log of the volume of air intake, and the variable LogRate represents the log of the rate of air intake.

The following statements invoke PROC LOGISTIC to fit a logistic regression model to the vasoconstriction data, where Response is the response variable, and LogRate and LogVolume are the explanatory variables. The ODS GRAPHICS statement is specified to display the regression diagnostics, and the INFLUENCE option is specified to display a table of the regression diagnostics.

```
ods graphics on;
title 'Occurrence of Vasoconstriction';
proc logistic data=vaso;
   model Response=LogRate LogVolume/influence iplots;
run;
ods graphics off;
```

Results of the model fit are shown in Output 51.6.1. Both LogRate and LogVolume are statistically significant to the occurrence of vasoconstriction ($p = 0.0131$ and $p = 0.0055$, respectively). Their positive parameter estimates indicate that a higher inspiration rate or a larger volume of air intake is likely to increase the probability of vasoconstriction.

Output 51.6.1 Logistic Regression Analysis for Vasoconstriction Data

```
                Occurrence of Vasoconstriction

                      Model Information

Data Set                          WORK.VASO
Response Variable                 Response
Number of Response Levels         2
Model                             binary logit
Optimization Technique            Fisher's scoring

   Number of Observations Read          39
   Number of Observations Used          39

                  Response Profile

    Ordered                          Total
      Value     Response         Frequency

          1     constrict               20
          2     no_constrict            19

 Probability modeled is Response='constrict'.

              Model Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.
```

Output 51.6.1 *continued*

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	56.040	35.227
SC	57.703	40.218
-2 Log L	54.040	29.227

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	24.8125	2	<.0001
Score	16.6324	2	0.0002
Wald	7.8876	2	0.0194

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-2.8754	1.3208	4.7395	0.0295
LogRate	1	4.5617	1.8380	6.1597	0.0131
LogVolume	1	5.1793	1.8648	7.7136	0.0055

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
LogRate	95.744	2.610	>999.999
LogVolume	177.562	4.592	>999.999

Association of Predicted Probabilities and Observed Responses

Percent Concordant	93.7	Somers' D	0.874
Percent Discordant	6.3	Gamma	0.874
Percent Tied	0.0	Tau-a	0.448
Pairs	380	c	0.937

The INFLUENCE option displays the values of the explanatory variables (LogRate and LogVolume) for each observation, a column for each diagnostic produced, and the *case number* that represents the sequence number of the observation (Output 51.6.2). Also produced (but suppressed by the ODS GRAPHICS statement) is a line-printer plot where the vertical axis represents the case number and the horizontal axis represents the value of the diagnostic statistic.

Output 51.6.2 Regression Diagnostics from the INFLUENCE Option

Regression Diagnostics

Case Number	Covariates LogRate	Log Volume	Pearson Residual	Deviance Residual	Hat Matrix Diagonal	Intercept DfBeta	LogRate DfBeta
1	−0.1924	1.3083	0.2205	0.3082	0.0927	−0.0165	0.0193
2	0.0862	1.2528	0.1349	0.1899	0.0429	−0.0134	0.0151
3	0.9163	0.2231	0.2923	0.4049	0.0612	−0.0492	0.0660
4	0.4055	−0.2877	3.5181	2.2775	0.0867	1.0734	−0.9302
5	1.1632	−0.2231	0.5287	0.7021	0.1158	−0.0832	0.1411
6	1.2528	−0.3567	0.6090	0.7943	0.1524	−0.0922	0.1710
7	−0.2877	−0.5108	−0.0328	−0.0464	0.00761	−0.00280	0.00274
8	0.5306	0.0953	−1.0196	−1.1939	0.0559	−0.1444	0.0613
9	−0.2877	−0.1054	−0.0938	−0.1323	0.0342	−0.0178	0.0173
10	−0.7985	−0.1054	−0.0293	−0.0414	0.00721	−0.00245	0.00246
11	−0.5621	−0.2231	−0.0370	−0.0523	0.00969	−0.00361	0.00358
12	1.0116	−0.5978	−0.5073	−0.6768	0.1481	−0.1173	0.0647
13	1.0986	−0.5108	−0.7751	−0.9700	0.1628	−0.0931	−0.00946
14	0.8459	0.3365	0.2559	0.3562	0.0551	−0.0414	0.0538
15	1.3218	−0.2877	0.4352	0.5890	0.1336	−0.0940	0.1408
16	0.4947	0.8329	0.1576	0.2215	0.0402	−0.0198	0.0234
17	0.4700	1.1632	0.0709	0.1001	0.0172	−0.00630	0.00701
18	0.3471	−0.1625	2.9062	2.1192	0.0954	0.9595	−0.8279
19	0.0583	0.5306	−1.0718	−1.2368	0.1315	−0.2591	0.2024
20	0.5878	0.5878	0.2405	0.3353	0.0525	−0.0331	0.0421
21	0.6931	−0.9163	−0.1076	−0.1517	0.0373	−0.0180	0.0158
22	0.3075	−0.0513	−0.4193	−0.5691	0.1015	−0.1449	0.1237
23	0.3001	0.3001	−1.0242	−1.1978	0.0761	−0.1961	0.1275
24	0.3075	0.4055	−1.3684	−1.4527	0.0717	−0.1281	0.0410
25	0.5766	0.4700	0.3347	0.4608	0.0587	−0.0403	0.0570
26	0.4055	−0.5108	−0.1595	−0.2241	0.0548	−0.0366	0.0329
27	0.4055	0.5878	0.3645	0.4995	0.0661	−0.0327	0.0496
28	0.6419	−0.0513	−0.8989	−1.0883	0.0647	−0.1423	0.0617
29	−0.0513	0.6419	0.8981	1.0876	0.1682	0.2367	−0.1950
30	−0.9163	0.4700	−0.0992	−0.1400	0.0507	−0.0224	0.0227
31	−0.2877	0.9933	0.6198	0.8064	0.2459	0.1165	−0.0996
32	−3.5066	0.8544	−0.00073	−0.00103	0.000022	−3.22E−6	3.405E−6
33	0.6043	0.0953	−1.2062	−1.3402	0.0510	−0.0882	−0.0137
34	0.7885	0.0953	0.5447	0.7209	0.0601	−0.0425	0.0877
35	0.6931	0.1823	0.5404	0.7159	0.0552	−0.0340	0.0755
36	1.2030	−0.2231	0.4828	0.6473	0.1177	−0.0867	0.1381
37	0.6419	−0.0513	−0.8989	−1.0883	0.0647	−0.1423	0.0617
38	0.6419	−0.2877	−0.4874	−0.6529	0.1000	−0.1395	0.1032
39	0.4855	0.2624	0.7053	0.8987	0.0531	0.0326	0.0190

Output 51.6.2 *continued*

Regression Diagnostics

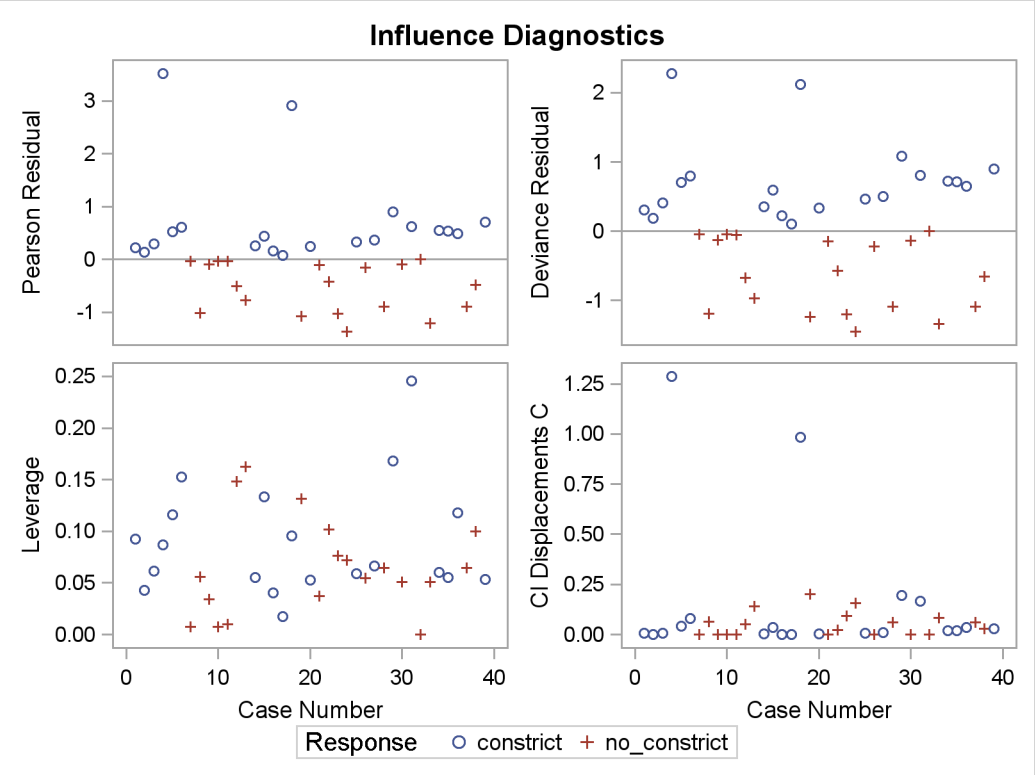
Case Number	Log Volume DfBeta	Confidence Interval Displacement C	Confidence Interval Displacement CBar	Delta Deviance	Delta Chi-Square
1	0.0556	0.00548	0.00497	0.1000	0.0536
2	0.0261	0.000853	0.000816	0.0369	0.0190
3	0.0589	0.00593	0.00557	0.1695	0.0910
4	-1.0180	1.2873	1.1756	6.3626	13.5523
5	0.0583	0.0414	0.0366	0.5296	0.3161
6	0.0381	0.0787	0.0667	0.6976	0.4376
7	0.00265	8.321E-6	8.258E-6	0.00216	0.00109
8	0.0570	0.0652	0.0616	1.4870	1.1011
9	0.0153	0.000322	0.000311	0.0178	0.00911
10	0.00211	6.256E-6	6.211E-6	0.00172	0.000862
11	0.00319	0.000014	0.000013	0.00274	0.00138
12	0.1651	0.0525	0.0447	0.5028	0.3021
13	0.1775	0.1395	0.1168	1.0577	0.7175
14	0.0527	0.00404	0.00382	0.1307	0.0693
15	0.0643	0.0337	0.0292	0.3761	0.2186
16	0.0307	0.00108	0.00104	0.0501	0.0259
17	0.00914	0.000089	0.000088	0.0101	0.00511
18	-0.8477	0.9845	0.8906	5.3817	9.3363
19	-0.00488	0.2003	0.1740	1.7037	1.3227
20	0.0518	0.00338	0.00320	0.1156	0.0610
21	0.0208	0.000465	0.000448	0.0235	0.0120
22	0.1179	0.0221	0.0199	0.3437	0.1956
23	0.0357	0.0935	0.0864	1.5212	1.1355
24	-0.1004	0.1558	0.1447	2.2550	2.0171
25	0.0708	0.00741	0.00698	0.2193	0.1190
26	0.0373	0.00156	0.00147	0.0517	0.0269
27	0.0788	0.0101	0.00941	0.2589	0.1423
28	0.1025	0.0597	0.0559	1.2404	0.8639
29	0.0286	0.1961	0.1631	1.3460	0.9697
30	0.0159	0.000554	0.000526	0.0201	0.0104
31	0.1322	0.1661	0.1253	0.7755	0.5095
32	2.48E-6	1.18E-11	1.18E-11	1.065E-6	5.324E-7
33	-0.00216	0.0824	0.0782	1.8744	1.5331
34	0.0671	0.0202	0.0190	0.5387	0.3157
35	0.0711	0.0180	0.0170	0.5295	0.3091
36	0.0631	0.0352	0.0311	0.4501	0.2641
37	0.1025	0.0597	0.0559	1.2404	0.8639
38	0.1397	0.0293	0.0264	0.4526	0.2639
39	0.0489	0.0295	0.0279	0.8355	0.5254

The index plots produced by the IPLOTS option are essentially the same line-printer plots as those produced by the INFLUENCE option, but with a 90-degree rotation and perhaps on a more refined scale. Since the ODS GRAPHICS statement is specified, the line-printer plots from the INFLUENCE and IPLOTS options are suppressed and ODS Graphics versions of the plots are displayed in Outputs 51.6.3 through 51.6.5. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the

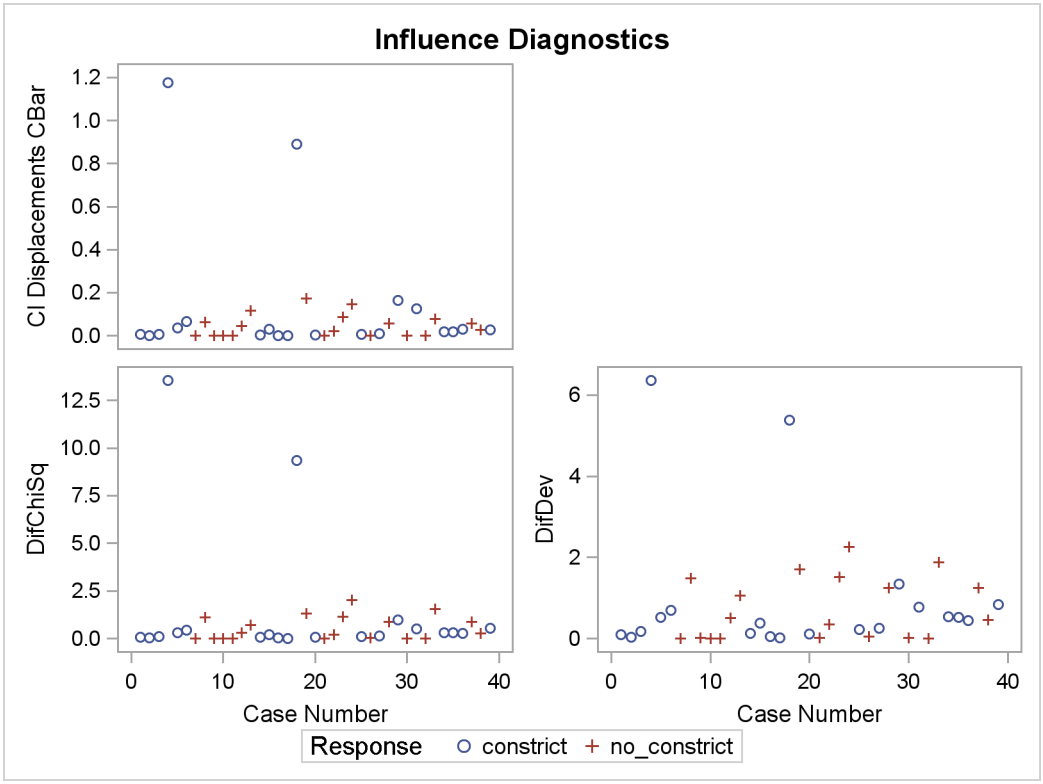
LOGISTIC procedure, see the section "ODS Graphics" on page 3379. The vertical axis of an index plot represents the value of the diagnostic, and the horizontal axis represents the sequence (case number) of the observation. The index plots are useful for identification of extreme values.

The index plots of the Pearson residuals and the deviance residuals (Output 51.6.3) indicate that case 4 and case 18 are poorly accounted for by the model. The index plot of the diagonal elements of the hat matrix (Output 51.6.3) suggests that case 31 is an extreme point in the design space. The index plots of DFBETAS (Outputs 51.6.4 and 51.6.5) indicate that case 4 and case 18 are causing instability in all three parameter estimates. The other four index plots in Outputs 51.6.3 and 51.6.4 also point to these two cases as having a large impact on the coefficients and goodness of fit.

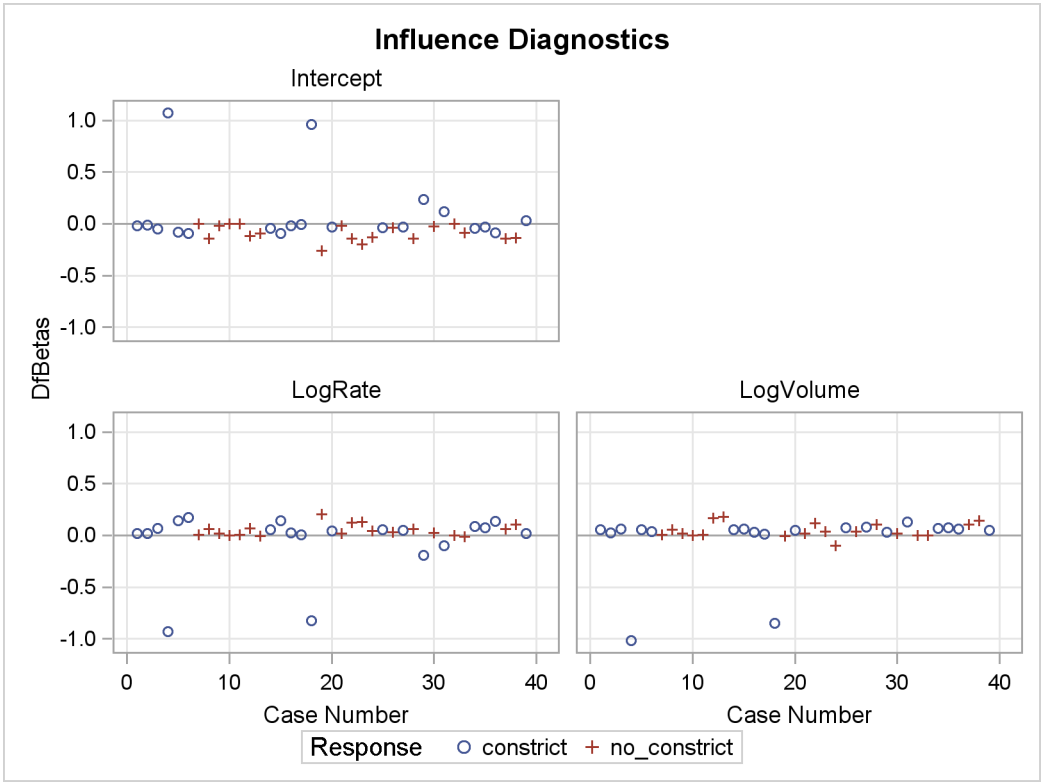
Output 51.6.3 Residuals, Hat Matrix, and CI Displacement C



Output 51.6.4 CI Displacement CBar, Change in Deviance and Pearson Chi-Square



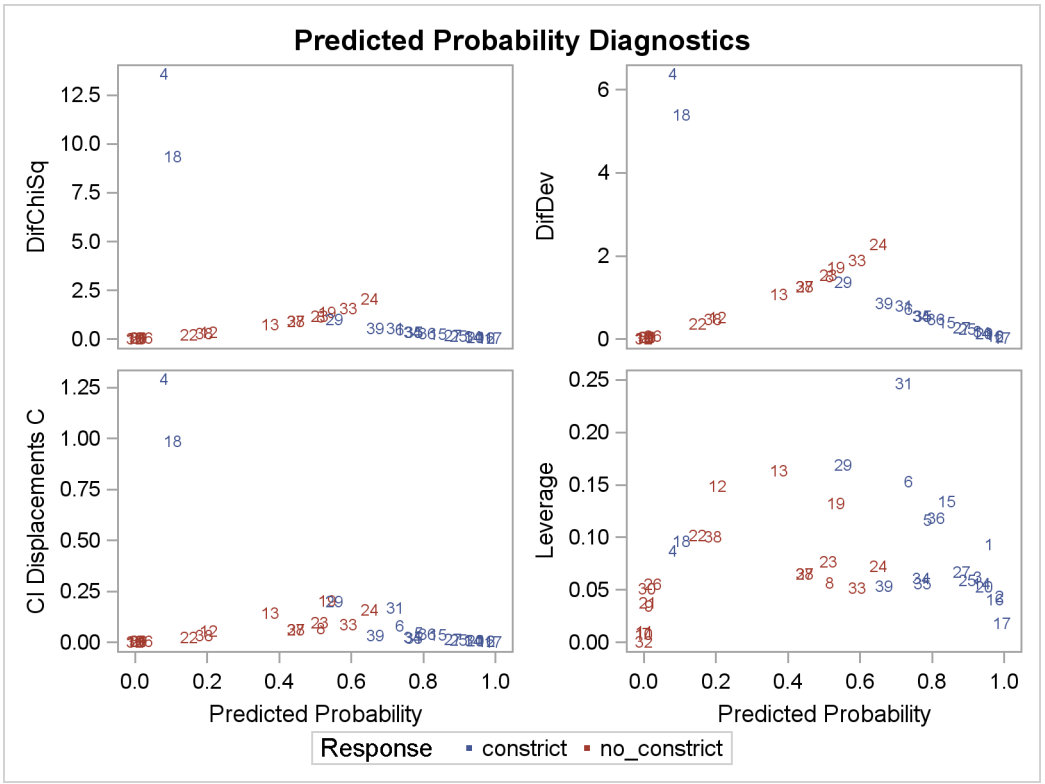
Output 51.6.5 DFBETAS Plots



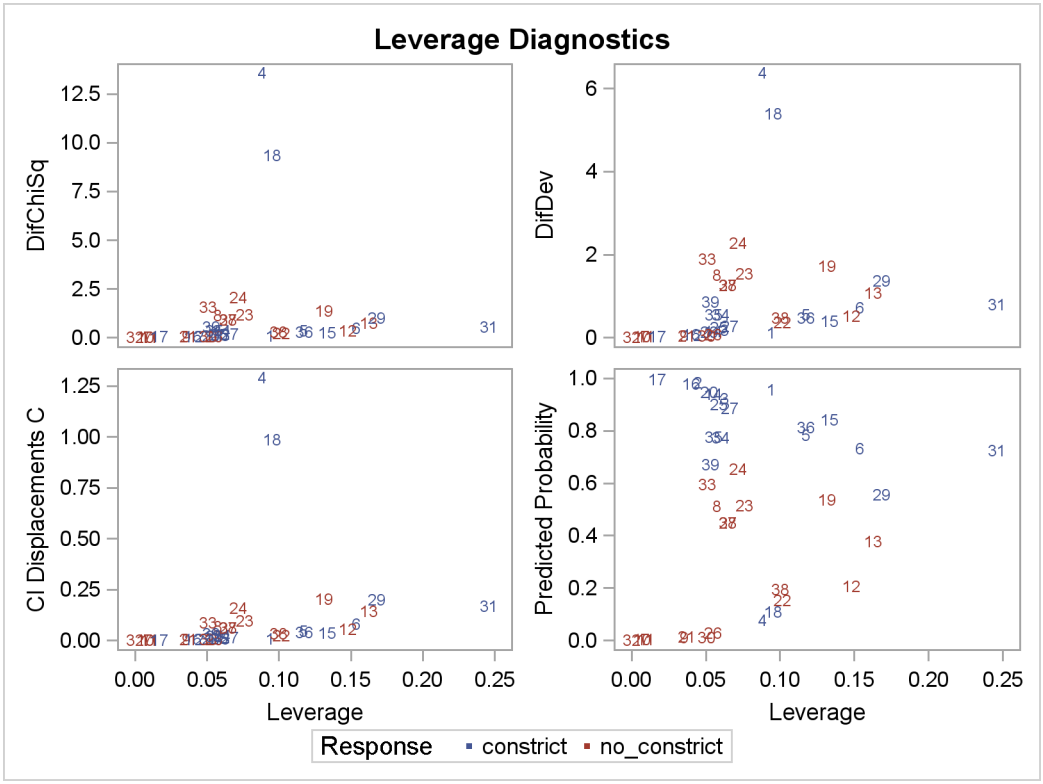
Other versions of diagnostic plots can be requested by specifying the appropriate options in the PLOTS= option. For example, the following statements produce three other sets of influence diagnostic plots: the PHAT option plots several diagnostics against the predicted probabilities (Output 51.6.6), the LEVERAGE option plots several diagnostics against the leverage (Output 51.6.7), and the DPC option plots the deletion diagnostics against the predicted probabilities and colors the observations according to the confidence interval displacement diagnostic (Output 51.6.8). The LABEL option displays the observation numbers on the plots. In all plots, you are looking for the outlying observations, and again cases 4 and 18 are noted.

```
ods graphics on;
proc logistic data=vaso plots(only label)=(phat leverage dpc);
   model Response=LogRate LogVolume;
run;
ods graphics off;
```

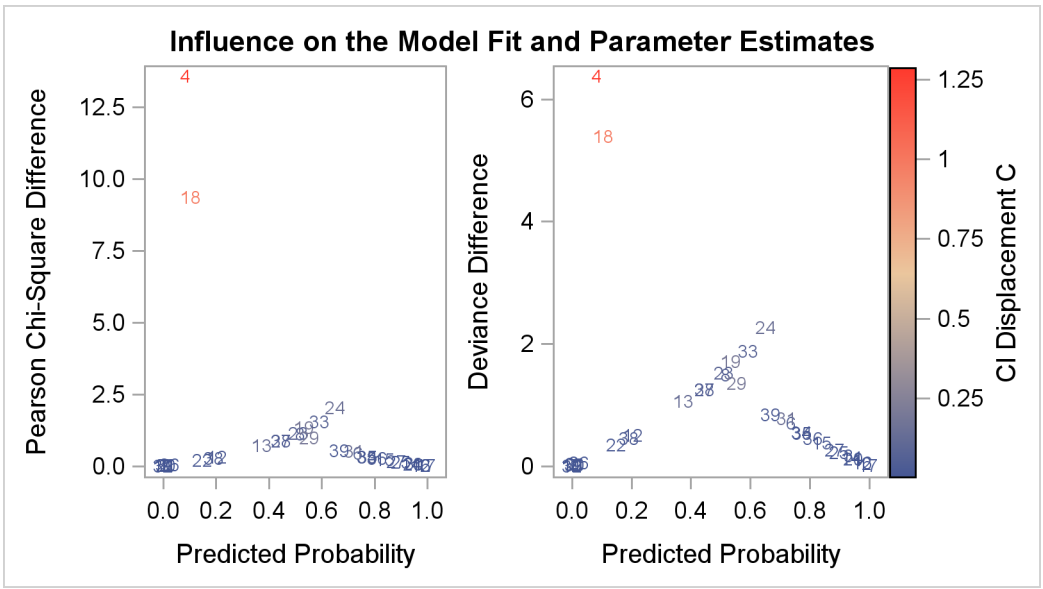
Output 51.6.6 Diagnostics versus Predicted Probability



Output 51.6.7 Diagnostics versus Leverage



Output 51.6.8 Three Diagnostics



Example 51.7: ROC Curve, Customized Odds Ratios, Goodness-of-Fit Statistics, R-Square, and Confidence Limits

This example plots an ROC curve, estimates a customized odds ratio, produces the traditional goodness-of-fit analysis, displays the generalized R^2 measures for the fitted model, calculates the normal confidence intervals for the regression parameters, and produces a display of the probability function and prediction curves for the fitted model. The data consist of three variables: n (number of subjects in the sample), disease (number of diseased subjects in the sample), and age (age for the sample). A linear logistic regression model is used to study the effect of age on the probability of contracting the disease. The statements to produce the data set and perform the analysis are as follows:

```
data Data1;
   input disease n age;
   datalines;
 0 14 25
 0 20 35
 0 19 45
 7 18 55
 6 12 65
17 17 75
;

ods graphics on;
proc logistic data=Data1 plots(only)=(roc(id=obs) effect);
   model disease/n=age / scale=none
                         clparm=wald
                         clodds=pl
                         rsquare;
   units age=10;
run;
ods graphics off;
```

The option SCALE=NONE is specified to produce the deviance and Pearson goodness-of-fit analysis without adjusting for overdispersion. The RSQUARE option is specified to produce generalized R^2 measures of the fitted model. The CLPARM=WALD option is specified to produce the Wald confidence intervals for the regression parameters. The UNITS statement is specified to produce customized odds ratio estimates for a change of 10 years in the age variable, and the CLODDS=PL option is specified to produce profile-likelihood confidence limits for the odds ratio. The ODS Graphics statement and the PLOTS= option produce graphical displays of the ROC curve of the model fit.

The results in Output 51.7.1 show that the deviance and Pearson statistics indicate no lack of fit in the model.

Output 51.7.1 Deviance and Pearson Goodness-of-Fit Analysis

Deviance and Pearson Goodness-of-Fit Statistics

Criterion	Value	DF	Value/DF	Pr > ChiSq
Deviance	7.7756	4	1.9439	0.1002
Pearson	6.6020	4	1.6505	0.1585

Number of events/trials observations: 6

Output 51.7.2 shows that the R-square for the model is 0.74. The odds of an event increases by a factor of 7.9 for each 10-year increase in age.

Output 51.7.2 R-Square, Confidence Intervals, and Customized Odds Ratio

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	124.173	52.468
SC	126.778	57.678
-2 Log L	122.173	48.468

R-Square 0.5215 Max-rescaled R-Square 0.7394

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	73.7048	1	<.0001
Score	55.3274	1	<.0001
Wald	23.3475	1	<.0001

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-12.5016	2.5555	23.9317	<.0001
age	1	0.2066	0.0428	23.3475	<.0001

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
age	1.229	1.131	1.337

Output 51.7.2 *continued*

Association of Predicted Probabilities and Observed Responses

Percent Concordant	92.6	Somers' D	0.906
Percent Discordant	2.0	Gamma	0.958
Percent Tied	5.4	Tau-a	0.384
Pairs	2100	c	0.953

Wald Confidence Interval for Parameters

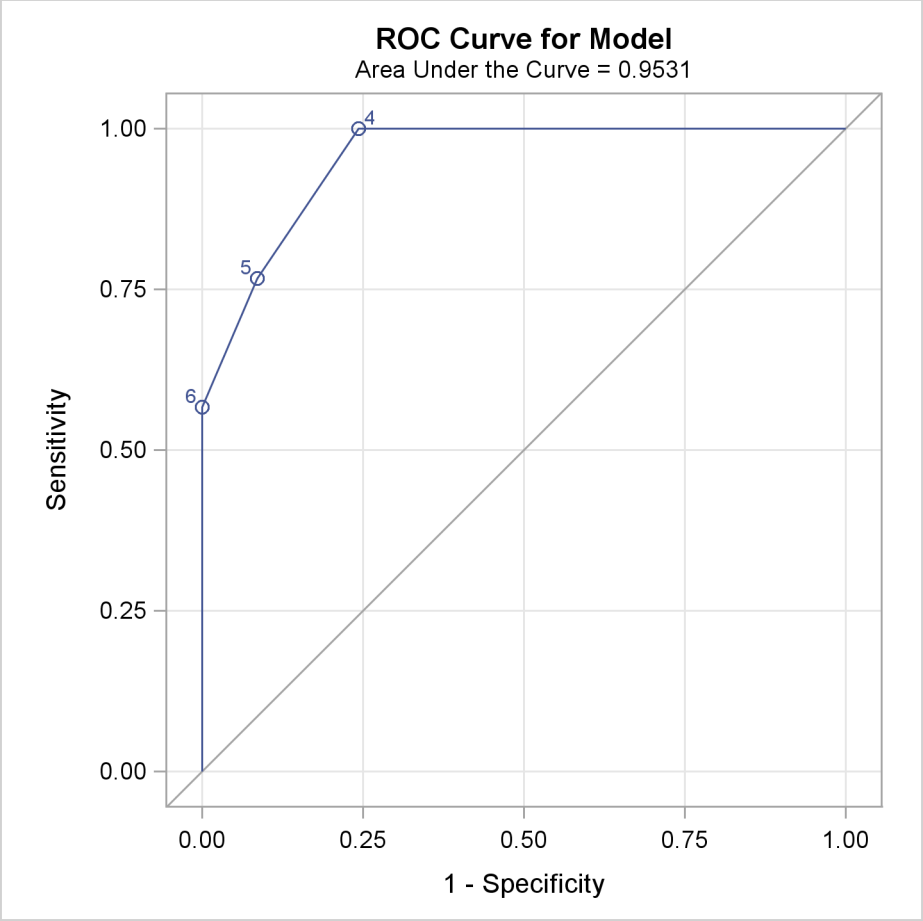
Parameter	Estimate	95% Confidence Limits	
Intercept	-12.5016	-17.5104	-7.4929
age	0.2066	0.1228	0.2904

Profile Likelihood Confidence Interval for Odds Ratios

Effect	Unit	Estimate	95% Confidence Limits	
age	10.0000	7.892	3.881	21.406

Since the ODS GRAPHICS statement is specified, a graphical display of the ROC curve is produced as shown in Output 51.7.3.

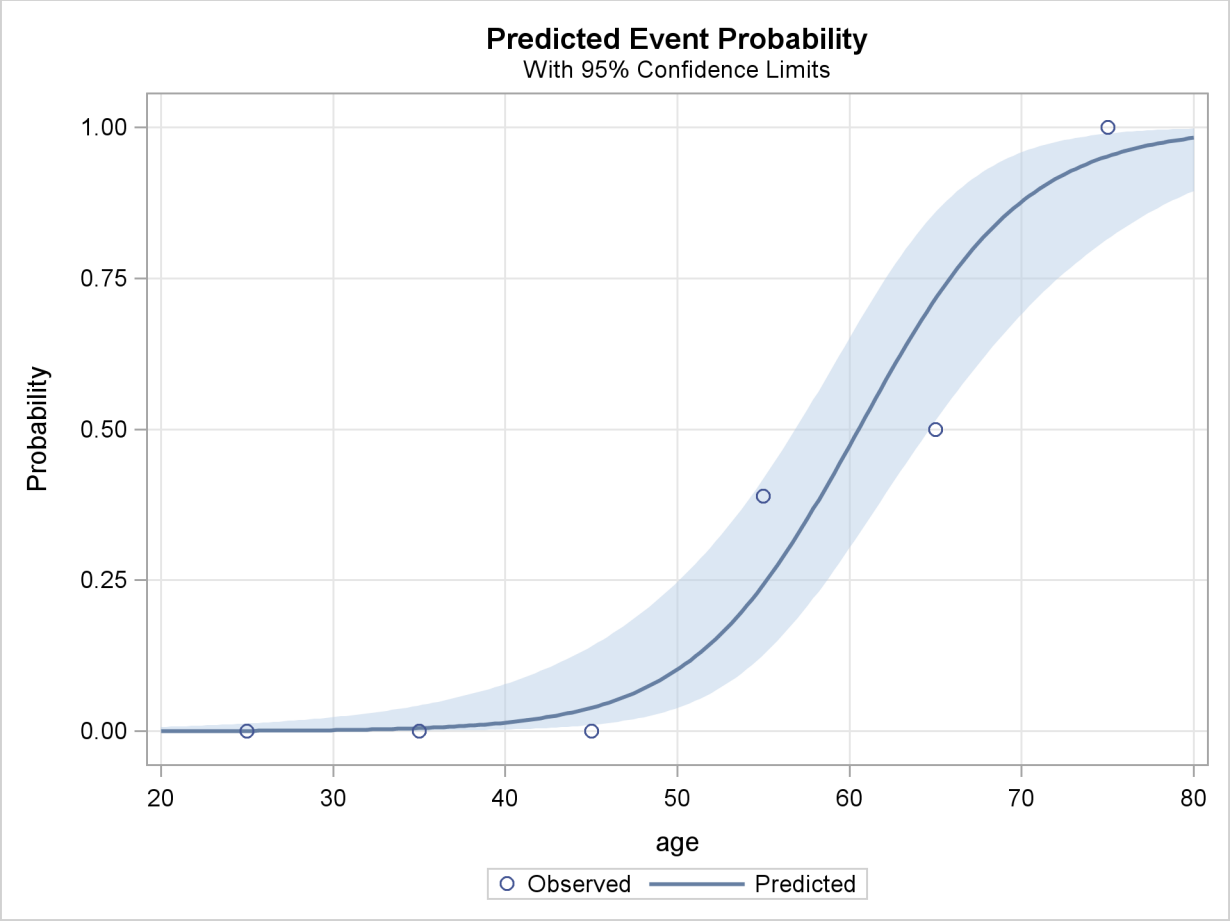
Output 51.7.3 Receiver Operating Characteristic Curve



Note that the area under the ROC curve is estimated by the statistic *c* in the "Association of Predicted Probabilities and Observed Responses" table. In this example, the area under the ROC curve is 0.953.

Because there is only one continuous covariate, the ODS GRAPHICS statement and the PLOTS=EFFECT option produce a graphical display of the predicted probability curve with bounding 95% confidence limits as shown in Output 51.7.4.

Output 51.7.4 Predicted Probability and 95% Prediction Limits



Example 51.8: Comparing Receiver Operating Characteristic Curves

DeLong, DeLong, and Clarke-Pearson (1988) report on 49 patients with ovarian cancer who also suffer from an intestinal obstruction. Three (correlated) screening tests are measured to determine whether a patient will benefit from surgery. The three tests are the K-G score and two measures of nutritional status: total protein and albumin. The data are as follows:

```
data roc;
   input alb tp totscore popind @@;
   totscore = 10 - totscore;
   datalines;
3.0 5.8 10 0   3.2 6.3  5 1   3.9 6.8  3 1   2.8 4.8  6 0
3.2 5.8  3 1   0.9 4.0  5 0   2.5 5.7  8 0   1.6 5.6  5 1
3.8 5.7  5 1   3.7 6.7  6 1   3.2 5.4  4 1   3.8 6.6  6 1
4.1 6.6  5 1   3.6 5.7  5 1   4.3 7.0  4 1   3.6 6.7  4 0
2.3 4.4  6 1   4.2 7.6  4 0   4.0 6.6  6 0   3.5 5.8  6 1
3.8 6.8  7 1   3.0 4.7  8 0   4.5 7.4  5 1   3.7 7.4  5 1
3.1 6.6  6 1   4.1 8.2  6 1   4.3 7.0  5 1   4.3 6.5  4 1
3.2 5.1  5 1   2.6 4.7  6 1   3.3 6.8  6 0   1.7 4.0  7 0
3.7 6.1  5 1   3.3 6.3  7 1   4.2 7.7  6 1   3.5 6.2  5 1
2.9 5.7  9 0   2.1 4.8  7 1   2.8 6.2  8 0   4.0 7.0  7 1
3.3 5.7  6 1   3.7 6.9  5 1   3.6 6.6  5 1
;
```

In the following statements, the NOFIT option is specified in the MODEL statement to prevent PROC LOGISTIC from fitting the model with three covariates. Each ROC statement lists one of the covariates, and PROC LOGISTIC then fits the model with that single covariate. Note that the original data set contains six more records with missing values for one of the tests, but PROC LOGISTIC ignores all records with missing values; hence there is a common sample size for each of the three models. The ROCCONTRAST statement implements the nonparameteric approach of DeLong, DeLong, and Clarke-Pearson (1988) to compare the three ROC curves, the REFERENCE option specifies that the K-G Score curve is used as the reference curve in the contrast, the E option displays the contrast coefficients, and the ESTIMATE option computes and tests each comparison. The ODS GRAPHICS statement and the **plots=roc(id=prob)** specification in the PROC LOGISTIC statement will display several plots, and the plots of individual ROC curves will have certain points labeled with their predicted probabilities.

```
ods graphics on;
proc logistic data=roc plots=roc(id=prob);
   model popind(event='0') = alb tp totscore / nofit;
   roc 'Albumin' alb;
   roc 'K-G Score' totscore;
   roc 'Total Protein' tp;
   roccontrast reference('K-G Score') / estimate e;
run;
ods graphics off;
```

The initial model information is displayed in Output 51.8.1.

Output 51.8.1 Initial LOGISTIC Output

```
                    Model Information

Data Set                      WORK.ROC
Response Variable             popind
Number of Response Levels     2
Model                         binary logit
Optimization Technique        Fisher's scoring

   Number of Observations Read          43
   Number of Observations Used          43

                 Response Profile

      Ordered                      Total
        Value       popind     Frequency

            1            0            12
            2            1            31

       Probability modeled is popind=0.

     Score Test for Global Null Hypothesis

      Chi-Square       DF     Pr > ChiSq

         10.7939        3         0.0129
```

For each ROC model, the model fitting details in Outputs 51.8.2, 51.8.4, and 51.8.6 can be suppressed with the ROCOPTIONS(NODETAILS) option; however, the convergence status is always displayed.

The ROC curves for the three models are displayed in Outputs 51.8.3, 51.8.5, and 51.8.7. Note that the labels on the ROC curve are produced by specifying the ID=PROB option, and are the predicted probabilities for the cutpoints.

Output 51.8.2 Fit Tables for Popind=Alb

```
                Model Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

                 Model Fit Statistics

                                   Intercept
                   Intercept            and
    Criterion           Only     Covariates

    AIC               52.918         49.384
    SC                54.679         52.907
    -2 Log L          50.918         45.384
```

Output 51.8.2 *continued*

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	5.5339	1	0.0187
Score	5.6893	1	0.0171
Wald	4.6869	1	0.0304

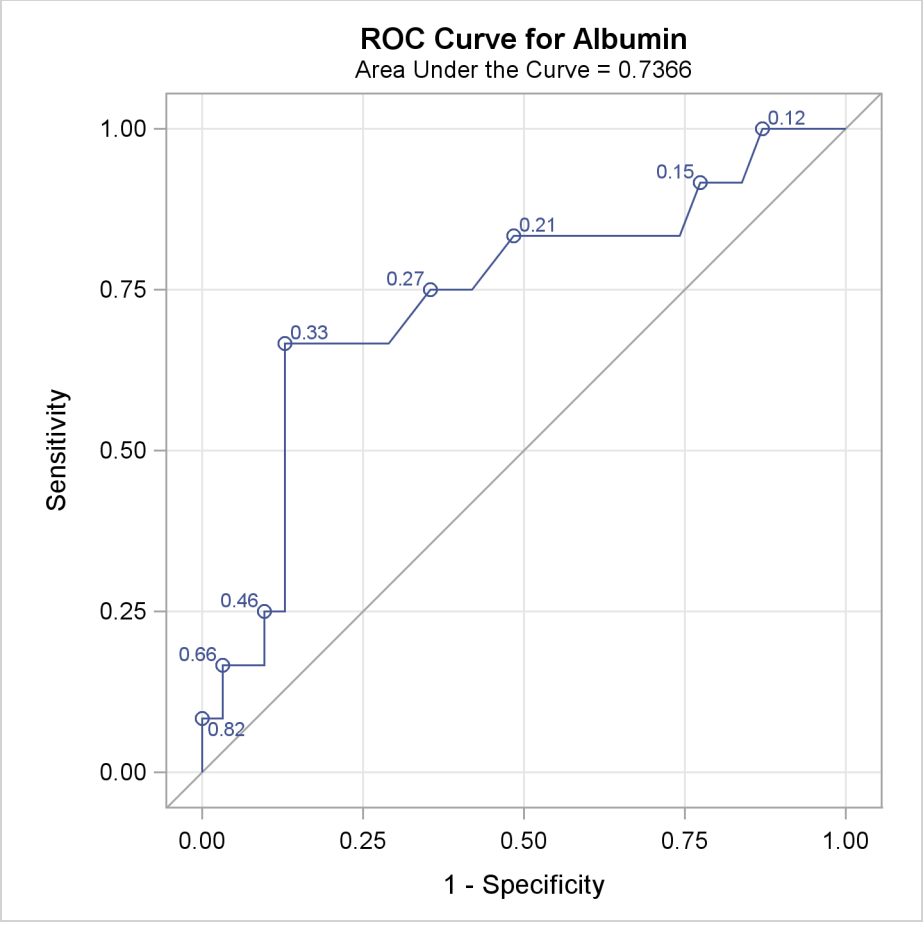
Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	2.4646	1.5913	2.3988	0.1214
alb	1	-1.0520	0.4859	4.6869	0.0304

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
alb	0.349	0.135	0.905

Output 51.8.3 ROC Curve for Popind=Alb



Output 51.8.4 Fit Tables for Popind=Totscore

Model Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	52.918	46.262
SC	54.679	49.784
-2 Log L	50.918	42.262

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	8.6567	1	0.0033
Score	8.3613	1	0.0038
Wald	6.3845	1	0.0115

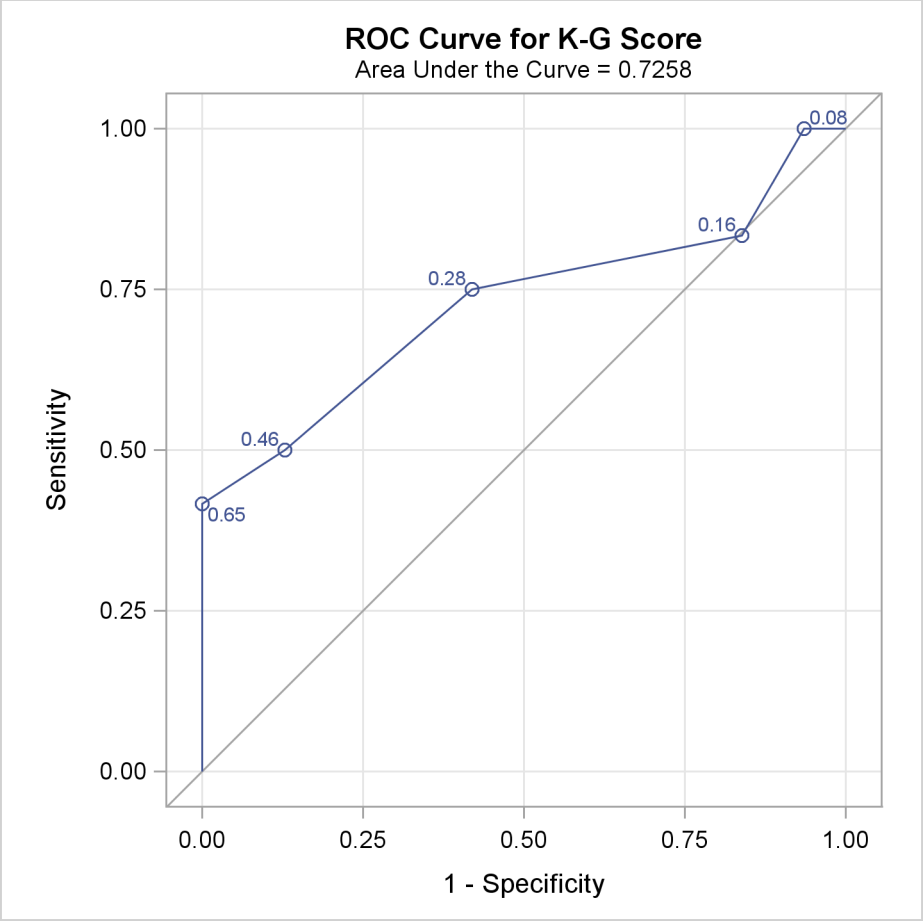
Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	2.1542	1.2477	2.9808	0.0843
totscore	1	-0.7696	0.3046	6.3845	0.0115

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
totscore	0.463	0.255	0.841

Output 51.8.5 ROC Curve for Popind=Totscore



Output 51.8.6 Fit Tables for Popind=Tp

Model Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	52.918	51.794
SC	54.679	55.316
-2 Log L	50.918	47.794

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	3.1244	1	0.0771
Score	3.1123	1	0.0777
Wald	2.9059	1	0.0883

Output 51.8.6 *continued*

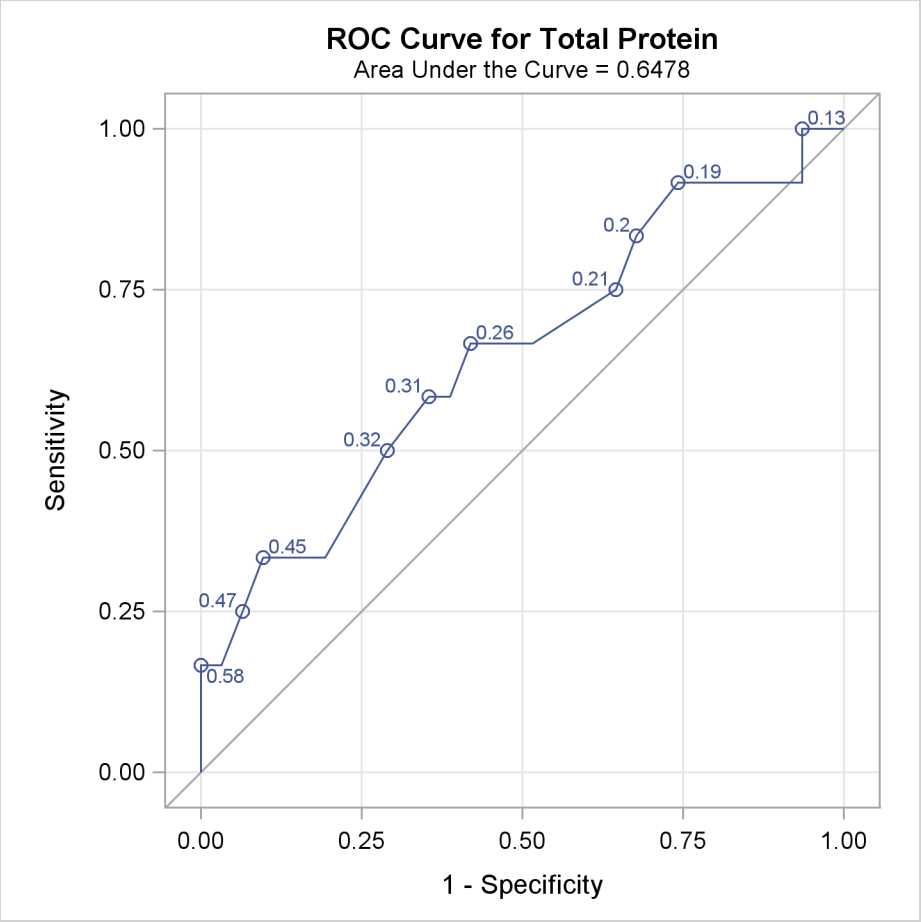
Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	2.8295	2.2065	1.6445	0.1997
tp	1	−0.6279	0.3683	2.9059	0.0883

Odds Ratio Estimates

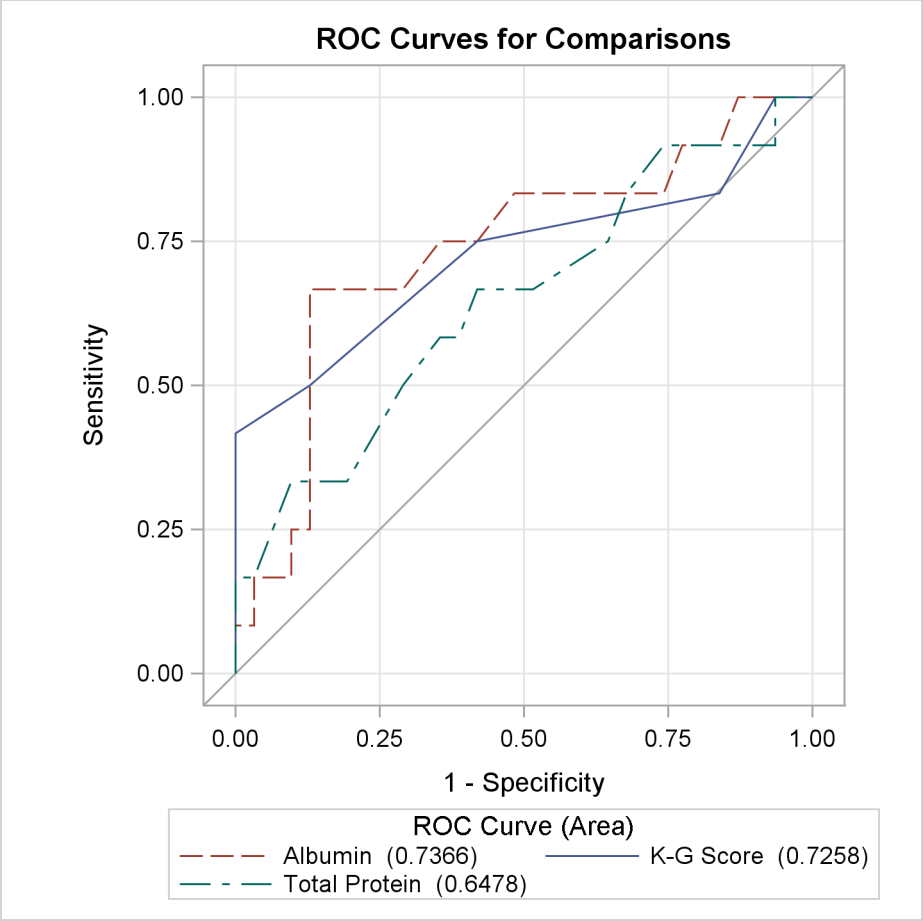
Effect	Point Estimate	95% Wald Confidence Limits	
tp	0.534	0.259	1.099

Output 51.8.7 ROC Curve for Popind=Tp



All ROC curves being compared are also overlaid on the same plot, as shown in Output 51.8.8.

Output 51.8.8 Overlay of All Models Being Compared



Output 51.8.9 displays the association statistics, and displays the area under the ROC curve (estimated by "c" in the "ROC Association Statistics" table) along with its standard error and a confidence interval for each model in the comparison. The confidence interval for Total Protein contains 0.50; hence it is not significantly different from random guessing, which is represented by the diagonal line in the preceding ROC plots.

Output 51.8.9 ROC Association Table

ROC Association Statistics

	Mann-Whitney						
		Standard	95% Wald		Somers' D		
ROC Model	Area	Error	Confidence Limits		(Gini)	Gamma	Tau-a
Albumin	0.7366	0.0927	0.5549	0.9182	0.4731	0.4809	0.1949
K-G Score	0.7258	0.1028	0.5243	0.9273	0.4516	0.5217	0.1860
Total Protein	0.6478	0.1000	0.4518	0.8439	0.2957	0.3107	0.1218

Output 51.8.10 shows that the contrast used 'K-G Score' as the reference level. This table is produced by specifying the E option in the ROCCONTRAST statement.

Output 51.8.10 ROC Contrast Coefficients

ROC Contrast Coefficients

ROC Model	Row1	Row2
Albumin	1	0
K-G Score	-1	-1
Total Protein	0	1

Output 51.8.11 shows that the 2-degrees-of-freedom test that the 'K-G Score' is different from at least one other test is not significant at the 0.05 level.

Output 51.8.11 ROC Test Results (2 Degrees of Freedom)

ROC Contrast Test Results

Contrast	DF	Chi-Square	Pr > ChiSq
Reference = K-G Score	2	2.5340	0.2817

Output 51.8.12 is produced by specifying the ESTIMATE option. Each row shows that the curves are not significantly different.

Output 51.8.12 ROC Contrast Row Estimates (1-Degree-of-Freedom Tests)

ROC Contrast Rows Estimation and Testing Results

Contrast	Estimate	Standard Error	95% Wald Confidence Limits		Chi-Square	Pr > ChiSq
Albumin - K-G Score	0.0108	0.0953	-0.1761	0.1976	0.0127	0.9102
Total Protein - K-G Score	-0.0780	0.1046	-0.2830	0.1271	0.5554	0.4561

Example 51.9: Goodness-of-Fit Tests and Subpopulations

A study is done to investigate the effects of two binary factors, A and B, on a binary response, Y. Subjects are randomly selected from subpopulations defined by the four possible combinations of levels of A and B. The number of subjects responding with each level of Y is recorded, and the following DATA step creates the data set One:

```
data One;
   do A=0,1;
      do B=0,1;
         do Y=1,2;
            input F @@;
            output;
         end;
      end;
   end;
   datalines;
23 63 31 70 67 100 70 104
;
```

The following statements fit a full model to examine the main effects of A and B as well as the interaction effect of A and B:

```
proc logistic data=One;
   freq F;
   model Y=A B A*B;
run;
```

Results of the model fit are shown in Output 51.9.1. Notice that neither the A*B interaction nor the B main effect is significant.

Output 51.9.1 Full Model Fit

```
                         Model Information

              Data Set                      WORK.ONE
              Response Variable             Y
              Number of Response Levels     2
              Frequency Variable            F
              Model                         binary logit
              Optimization Technique        Fisher's scoring

                 Number of Observations Read           8
                 Number of Observations Used           8
                 Sum of Frequencies Read             528
                 Sum of Frequencies Used             528
```

Output 51.9.1 *continued*

Response Profile

Ordered Value	Y	Total Frequency
1	1	191
2	2	337

Probability modeled is Y=1.

Model Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	693.061	691.914
SC	697.330	708.990
-2 Log L	691.061	683.914

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	7.1478	3	0.0673
Score	6.9921	3	0.0721
Wald	6.9118	3	0.0748

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-1.0074	0.2436	17.1015	<.0001
A	1	0.6069	0.2903	4.3714	0.0365
B	1	0.1929	0.3254	0.3515	0.5533
A*B	1	-0.1883	0.3933	0.2293	0.6321

Pearson and deviance goodness-of-fit tests cannot be obtained for this model since a full model containing four parameters is fit, leaving no residual degrees of freedom. For a binary response model, the goodness-of-fit tests have $m - q$ degrees of freedom, where m is the number of subpopulations and q is the number of model parameters. In the preceding model, $m = q = 4$, resulting in zero degrees of freedom for the tests.

The following statements fit a reduced model containing only the A effect, so two degrees of freedom become available for testing goodness of fit. Specifying the SCALE=NONE option requests the Pearson and deviance statistics. With *single-trial* syntax, the AGGREGATE= option is needed to define the subpopulations in the study. Specifying AGGREGATE=(A B) creates subpopulations of

the four combinations of levels of A and B. Although the B effect is being dropped from the model, it is still needed to define the original subpopulations in the study. If AGGREGATE=(A) were specified, only two subpopulations would be created from the levels of A, resulting in $m = q = 2$ and zero degrees of freedom for the tests.

```
proc logistic data=One;
   freq F;
   model Y=A / scale=none aggregate=(A B);
run;
```

The goodness-of-fit tests in Output 51.9.2 show that dropping the B main effect and the A*B interaction simultaneously does not result in significant lack of fit of the model. The tests' large p-values indicate insufficient evidence for rejecting the null hypothesis that the model fits.

Output 51.9.2 Reduced Model Fit

Deviance and Pearson Goodness-of-Fit Statistics

Criterion	Value	DF	Value/DF	Pr > ChiSq
Deviance	0.3541	2	0.1770	0.8377
Pearson	0.3531	2	0.1765	0.8382

Number of unique profiles: 4

Example 51.10: Overdispersion

In a seed germination test, seeds of two cultivars were planted in pots of two soil conditions. The following statements create the data set seeds, which contains the observed proportion of seeds that germinated for various combinations of cultivar and soil condition. The variable n represents the number of seeds planted in a pot, and the variable r represents the number germinated. The indicator variables cult and soil represent the cultivar and soil condition, respectively.

```
data seeds;
   input pot n r cult soil;
   datalines;
 1 16     8      0       0
 2 51    26      0       0
 3 45    23      0       0
 4 39    10      0       0
 5 36     9      0       0
 6 81    23      1       0
 7 30    10      1       0
 8 39    17      1       0
 9 28     8      1       0
10 62    23      1       0
11 51    32      0       1
12 72    55      0       1
13 41    22      0       1
14 12     3      0       1
15 13    10      0       1
16 79    46      1       1
17 30    15      1       1
18 51    32      1       1
19 74    53      1       1
20 56    12      1       1
;
```

PROC LOGISTIC is used as follows to fit a logit model to the data, with cult, soil, and cult × soil interaction as explanatory variables. The option SCALE=NONE is specified to display goodness-of-fit statistics.

```
proc logistic data=seeds;
   model r/n=cult soil cult*soil/scale=none;
   title 'Full Model With SCALE=NONE';
run;
```

Results of fitting the full factorial model are shown in Output 51.10.1. Both Pearson χ^2 and deviance are highly significant ($p < 0.0001$), suggesting that the model does not fit well.

Output 51.10.1 Results of the Model Fit for the Two-Way Layout

Full Model With SCALE=NONE

Deviance and Pearson Goodness-of-Fit Statistics

Criterion	Value	DF	Value/DF	Pr > ChiSq
Deviance	68.3465	16	4.2717	<.0001
Pearson	66.7617	16	4.1726	<.0001

Number of events/trials observations: 20

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	1256.852	1213.003
SC	1261.661	1232.240
-2 Log L	1254.852	1205.003

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	49.8488	3	<.0001
Score	49.1682	3	<.0001
Wald	47.7623	3	<.0001

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-0.3788	0.1489	6.4730	0.0110
cult	1	-0.2956	0.2020	2.1412	0.1434
soil	1	0.9781	0.2128	21.1234	<.0001
cult*soil	1	-0.1239	0.2790	0.1973	0.6569

If the link function and the model specification are correct and if there are no outliers, then the lack of fit might be due to overdispersion. Without adjusting for the overdispersion, the standard errors are likely to be underestimated, causing the Wald tests to be too sensitive. In PROC LOGISTIC, there are three SCALE= options to accommodate overdispersion. With unequal sample sizes for the observations, SCALE=WILLIAMS is preferred. The Williams model estimates a scale parameter ϕ by equating the value of Pearson χ^2 for the full model to its approximate expected value. The full model considered in the following statements is the model with cultivar, soil condition, and their interaction. Using a full model reduces the risk of contaminating ϕ with lack of fit due to incorrect model specification.

```
proc logistic data=seeds;
   model r/n=cult soil cult*soil / scale=williams;
   title 'Full Model With SCALE=WILLIAMS';
run;
```

Results of using Williams' method are shown in Output 51.10.2. The estimate of ϕ is 0.075941 and is given in the formula for the Weight Variable at the beginning of the displayed output.

Output 51.10.2 Williams' Model for Overdispersion

```
                    Full Model With SCALE=WILLIAMS

                          Model Information

        Data Set                          WORK.SEEDS
        Response Variable (Events)        r
        Response Variable (Trials)        n
        Weight Variable                   1 / ( 1 + 0.075941 * (n - 1) )
        Model                             binary logit
        Optimization Technique            Fisher's scoring

                 Number of Observations Read          20
                 Number of Observations Used          20
                 Sum of Frequencies Read             906
                 Sum of Frequencies Used             906
                 Sum of Weights Read            198.3216
                 Sum of Weights Used            198.3216

                          Response Profile

            Ordered     Binary           Total            Total
              Value     Outcome      Frequency           Weight

                1     Event              437         92.95346
                2     Nonevent           469        105.36819

                       Model Convergence Status

            Convergence criterion (GCONV=1E-8) satisfied.

           Deviance and Pearson Goodness-of-Fit Statistics

        Criterion          Value       DF     Value/DF     Pr > ChiSq

        Deviance         16.4402       16       1.0275         0.4227
        Pearson          16.0000       16       1.0000         0.4530

                 Number of events/trials observations: 20

NOTE: Since the Williams method was used to accommodate overdispersion, the
      Pearson chi-squared statistic and the deviance can no longer be used to
      assess the goodness of fit of the model.
```

Output 51.10.2 *continued*

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	276.155	273.586
SC	280.964	292.822
-2 Log L	274.155	265.586

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	8.5687	3	0.0356
Score	8.4856	3	0.0370
Wald	8.3069	3	0.0401

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-0.3926	0.2932	1.7932	0.1805
cult	1	-0.2618	0.4160	0.3963	0.5290
soil	1	0.8309	0.4223	3.8704	0.0491
cult*soil	1	-0.0532	0.5835	0.0083	0.9274

Since neither cult nor cult × soil is statistically significant ($p = 0.5290$ and $p = 0.9274$, respectively), a reduced model that contains only the soil condition factor is fitted, with the observations weighted by $1/(1 + 0.075941(N - 1))$. This can be done conveniently in PROC LOGISTIC by including the scale estimate in the SCALE=WILLIAMS option as follows:

```
proc logistic data=seeds;
   model r/n=soil / scale=williams(0.075941);
   title 'Reduced Model With SCALE=WILLIAMS(0.075941)';
run;
```

Results of the reduced model fit are shown in Output 51.10.3. Soil condition remains a significant factor ($p = 0.0064$) for the seed germination.

Output 51.10.3 Reduced Model with Overdispersion Controlled

Reduced Model With SCALE=WILLIAMS(0.075941)

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-0.5249	0.2076	6.3949	0.0114
soil	1	0.7910	0.2902	7.4284	0.0064

Example 51.11: Conditional Logistic Regression for Matched Pairs Data

In matched pairs, or *case-control*, studies, conditional logistic regression is used to investigate the relationship between an outcome of being an event (case) or a nonevent (control) and a set of prognostic factors.

The following data are a subset of the data from the Los Angeles Study of the Endometrial Cancer Data in Breslow and Day (1980). There are 63 matched pairs, each consisting of a case of endometrial cancer (Outcome=1) and a control (Outcome=0). The case and corresponding control have the same ID. Two prognostic factors are included: Gall (an indicator variable for gall bladder disease) and Hyper (an indicator variable for hypertension). The goal of the case-control analysis is to determine the relative risk for gall bladder disease, controlling for the effect of hypertension.

```
data Data1;
  do ID=1 to 63;
    do Outcome = 1 to 0 by -1;
      input Gall Hyper @@;
      output;
    end;
  end;
  datalines;
0 0  0 0    0 0  0 0    0 1  0 1    0 0  1 0    1 0  0 1
0 1  0 0    1 0  0 0    1 1  0 1    0 0  0 0    0 0  0 0
1 0  0 0    0 0  0 1    1 0  0 1    1 0  1 0    1 0  0 1
0 1  0 0    0 0  1 1    0 0  1 1    0 0  0 1    0 1  0 0
0 0  1 1    0 1  0 1    0 1  0 0    0 0  0 0    0 0  0 0
0 0  0 1    1 0  0 1    0 0  0 1    1 0  0 0    0 1  0 0
0 1  0 0    0 1  0 0    0 1  0 0    0 0  0 0    1 1  1 1
0 0  0 1    0 1  0 0    0 1  0 1    0 1  0 1    0 1  0 0
0 0  0 0    0 1  1 0    0 0  0 1    0 0  0 0    1 0  0 0
0 0  0 0    1 1  0 0    0 1  0 0    0 0  0 0    0 1  0 1
0 0  0 0    0 1  0 1    0 1  0 0    0 1  0 0    1 0  0 0
0 0  0 0    1 1  1 0    0 0  0 0    0 0  0 0    1 1  0 0
1 0  1 0    0 1  0 0    1 0  0 0
;
```

There are several ways to approach this problem with PROC LOGISTIC:

- Specify the STRATA statement to perform a conditional logistic regression.
- Specify EXACT and STRATA statements to perform an exact conditional logistic regression on the original data set, if you believe the data set is too small or too sparse for the usual asymptotics to hold.
- Transform each matched pair into a single observation, and then specify a PROC LOGISTIC statement on this transformed data without a STRATA statement; this also performs a conditional logistic regression and produces essentially the same results.
- Specify an EXACT statement on the transformed data.

SAS statements and selected results for these four approaches are given in the remainder of this example.

Conditional Analysis Using the STRATA Statement

In the following statements, PROC LOGISTIC is invoked with the ID variable declared in the STRATA statement to obtain the conditional logistic model estimates for a model containing Gall as the only predictor variable:

```
proc logistic data=Data1;
   strata ID;
   model outcome(event='1')=Gall;
run;
```

Results from the conditional logistic analysis are shown in Output 51.11.1. Note that there is no intercept term in the "Analysis of Maximum Likelihood Estimates" tables.

The odds ratio estimate for Gall is 2.60, which is marginally significant (p=0.0694) and which is an estimate of the relative risk for gall bladder disease. A 95% confidence interval for this relative risk is (0.927, 7.293).

Output 51.11.1 Conditional Logistic Regression (Gall as Risk Factor)

```
                    Conditional Analysis

                     Model Information

Data Set                          WORK.DATA1
Response Variable                 Outcome
Number of Response Levels         2
Number of Strata                  63
Model                             binary logit
Optimization Technique            Newton-Raphson ridge

      Number of Observations Read         126
      Number of Observations Used         126

                  Response Profile

          Ordered                       Total
            Value      Outcome     Frequency

                1            0            63
                2            1            63

          Probability modeled is Outcome=1.

                     Strata Summary

                Outcome
  Response      -------     Number of
   Pattern      0     1        Strata    Frequency

         1      1     1            63          126
```

Output 51.11.1 *continued*

Newton-Raphson Ridge Optimization

Without Parameter Scaling

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Without Covariates	With Covariates
AIC	87.337	85.654
SC	87.337	88.490
-2 Log L	87.337	83.654

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	3.6830	1	0.0550
Score	3.5556	1	0.0593
Wald	3.2970	1	0.0694

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Gall	1	0.9555	0.5262	3.2970	0.0694

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
Gall	2.600	0.927	7.293

Exact Analysis Using the STRATA Statement

When you believe there are not enough data or that the data are too sparse, you can perform a stratified exact conditional logistic regression. The following statements perform stratified exact conditional logistic regressions on the original data set by specifying both the STRATA and EXACT statements:

```
proc logistic data=Data1 exactonly;
   strata ID;
   model outcome(event='1')=Gall;
   exact Gall / estimate=both;
run;
```

Output 51.11.2 Exact Conditional Logistic Regression (Gall as Risk Factor)

Exact Conditional Analysis

Conditional Exact Tests

Effect	Test	Statistic	--- p-Value --- Exact	Mid
Gall	Score	3.5556	0.0963	0.0799
	Probability	0.0327	0.0963	0.0799

Exact Parameter Estimates

Parameter	Estimate	Standard Error	95% Confidence Limits		p-Value
Gall	0.9555	0.5262	-0.1394	2.2316	0.0963

Exact Odds Ratios

Parameter	Estimate	95% Confidence Limits		p-Value
Gall	2.600	0.870	9.315	0.0963

Note that the score statistic in the "Conditional Exact Tests" table in Output 51.11.2 is identical to the score statistic in Output 51.11.1 from the conditional analysis. The exact odds ratio confidence interval is much wider than its conditional analysis counterpart, but the parameter estimates are similar. The exact analysis confirms the marginal significance of Gall as a predictor variable.

Conditional Analysis Using Transformed Data

When each matched set consists of one event and one nonevent, the conditional likelihood is given by

$$\prod_i (1 + \exp(-\boldsymbol{\beta}'(\boldsymbol{x}_{i1} - \boldsymbol{x}_{i0}))^{-1}$$

where $\boldsymbol{x}_{i1}$ and $\boldsymbol{x}_{i0}$ are vectors representing the prognostic factors for the event and nonevent, respectively, of the ith matched set. This likelihood is identical to the likelihood of fitting a logistic regression model to a set of data with constant response, where the model contains no intercept term and has explanatory variables given by $\boldsymbol{d}_i = \boldsymbol{x}_{i1} - \boldsymbol{x}_{i0}$ (Breslow 1982).

To apply this method, the following DATA step transforms each matched pair into a single observation, where the variables Gall and Hyper contain the differences between the corresponding values for the case and the control (case–control). The variable Outcome, which will be used as the response variable in the logistic regression model, is given a constant value of 0 (which is the Outcome value for the control, although any constant, numeric or character, will suffice).

```
data Data2;
   set Data1;
   drop id1 gall1 hyper1;
   retain id1 gall1 hyper1 0;
   if (ID = id1) then do;
      Gall=gall1-Gall; Hyper=hyper1-Hyper;
      output;
   end;
   else do;
      id1=ID; gall1=Gall; hyper1=Hyper;
   end;
run;
```

Note that there are 63 observations in the data set, one for each matched pair. Since the number of observations n is halved, statistics that depend on n such as R^2 (the section "Generalized Coefficient of Determination" on page 3330) will be incorrect. The variable Outcome has a constant value of 0.

In the following statements, PROC LOGISTIC is invoked with the NOINT option to obtain the conditional logistic model estimates. Because the option CLODDS=PL is specified, PROC LOGISTIC computes a 95% profile-likelihood confidence interval for the odds ratio for each predictor variable; note that profile-likelihood confidence intervals are not currently available when a STRATA statement is specified.

```
proc logistic data=Data2;
   model outcome=Gall / noint clodds=PL;
run;
```

The results are not displayed here.

Exact Analysis Using Transformed Data

Sometimes the original data set in a matched-pairs study is too large for the exact methods to handle. In such cases it might be possible to use the transformed data set. The following statements perform exact conditional logistic regressions on the transformed data set. The results are not displayed here.

```
proc logistic data=Data2 exactonly;
   model outcome=Gall / noint;
   exact Gall / estimate=both;
run;
```

Example 51.12: Firth's Penalized Likelihood Compared with Other Approaches

Firth's penalized likelihood approach is a method of addressing issues of separability, small sample sizes, and bias of the parameter estimates. This example performs some comparisons between results from using the FIRTH option to results from the usual unconditional, conditional, and exact conditional logistic regression analyses. When the sample size is large enough, the unconditional estimates and the Firth penalized-likelihood estimates should be nearly the same. These examples show that Firth's penalized likelihood approach compares favorably with unconditional, conditional, and exact conditional logistic regression; however, this is not an exhaustive analysis of Firth's method. For more detailed analyses with separable data sets, see Heinze (2006, 1999) and Heinze and Schemper (2002).

Comparison on 2x2 Tables with One Zero Cell

A 2×2 table with one cell having zero frequency, where the rows of the table are the levels of a covariate while the columns are the levels of the response variable, is an example of a quasi-completely separated data set. The parameter estimate for the covariate under unconditional logistic regression will move off to infinity, although PROC LOGISTIC will stop the iterations at an earlier point in the process. An exact conditional logistic regression is sometimes performed to determine the importance of the covariate in describing the variation in the data, but the median-unbiased parameter estimate, while finite, might not be near the true value, and one confidence limit (for this example, the upper) is always infinite.

The following DATA step produces 1000 different 2×2 tables, all following an underlying probability structure, with one cell having a near zero probability of being observed:

```
%let beta0=-15;
%let beta1=16;
data one;
   keep sample X y pry;
   do sample=1 to 1000;
      do i=1 to 100;
         X=rantbl(987987,.4,.6)-1;
         xb= &beta0 + X*&beta1;
         exb=exp(xb);
         pry= exb/(1+exb);
         cut= ranuni(393993);
         if (pry < cut) then y=1; else y=0;
         output;
      end;
   end;
run;
```

The following statements perform the bias-corrected and exact logistic regression on each of the 1000 different data sets, output the odds ratio tables by using the ODS OUTPUT statement, and compute various statistics across the data sets by using the MEANS procedure:

```
ods exclude all;
proc logistic data=one;
   by sample;
   class X(param=ref);
   model y(event='1')=X / firth clodds=pl;
   ods output cloddspl=firth;
run;
proc logistic data=one exactonly;
   by sample;
   class X(param=ref);
   model y(event='1')=X;
   exact X / estimate=odds;
   ods output exactoddsratio=exact;
run;
ods select all;
proc means data=firth;
   var LowerCL OddsRatioEst UpperCL;
run;
proc means data=exact;
   var LowerCL Estimate UpperCL;
run;
```

The results of the PROC MEANS statements are summarized in Table 51.8. You can see that the odds ratios are all quite large; the confidence limits on every table suggest that the covariate X is a significant factor in explaining the variability in the data.

Table 51.8 Odds Ratio Results

Method	Mean Estimate	Standard Error	Minimum Lower CL	Maximum Upper CL
Firth	231.59	83.57	10.40	111317
Exact	152.02	52.30	8.82	∞

Comparison on Case-Control Data

Case-control models contain an intercept term for every case-control pair in the data set. This means that there are a large number of parameters compared to the number of observations. Breslow and Day (1980) note that the estimates from unconditional logistic regression are biased with the corresponding odds ratios off by a power of 2 from the true value; conditional logistic regression was developed to remedy this.

The following DATA step produces 1000 case-control data sets, with pair indicating the strata:

```
%let beta0=1;
%let beta1=2;
data one;
   do sample=1 to 1000;
      do pair=1 to 20;
         ran=ranuni(939393);
         a=3*ranuni(9384984)-1;
         pdf0= pdf('NORMAL',a,.4,1);
         pdf1= pdf('NORMAL',a,1,1);
         pry0= pdf0/(pdf0+pdf1);
         pry1= 1-pry0;
         xb= log(pry0/pry1);
         x= (xb-&beta0*pair/100) / &beta1;
         y=0;
         output;
         x= (-xb-&beta0*pair/100) / &beta1;
         y=1;
         output;
      end;
   end;
run;
```

Unconditional, conditional, exact conditional, and Firth-adjusted analyses are performed on the data sets, and the mean, minimum, and maximum odds ratios and the mean upper and lower limits for the odds ratios are displayed in Table 51.9. **WARNING:** Due to the exact analyses, this program takes a long time and a lot of resources to run. You might want to reduce the number of samples generated.

```
ods exclude all;
proc logistic data=one;
   by sample;
   class pair / param=ref;
   model y=x pair / clodds=pl;
   ods output cloddspl=oru;
run;
data oru;
   set oru;
   if Effect='x';
   rename lowercl=lclu uppercl=uclu oddsratioest=orestu;
run;
proc logistic data=one;
   by sample;
   strata pair;
   model y=x / clodds=wald;
   ods output cloddswald=orc;
run;
data orc;
   set orc;
   if Effect='x';
   rename lowercl=lclc uppercl=uclc oddsratioest=orestc;
run;
```

```
proc logistic data=one exactonly;
   by sample;
   strata pair;
   model y=x;
   exact x / estimate=both;
   ods output ExactOddsRatio=ore;
run;
proc logistic data=one;
   by sample;
   class pair / param=ref;
   model y=x pair / firth clodds=pl;
   ods output cloddspl=orf;
run;
data orf;
   set orf;
   if Effect='x';
   rename lowercl=lclf uppercl=uclf oddsratioest=orestf;
run;
data all;
   merge oru orc ore orf;
run;
ods select all;
proc means data=all;
run;
```

You can see from Table 51.9 that the conditional, exact conditional, and Firth-adjusted results are all comparable, while the unconditional results are several orders of magnitude different.

Table 51.9 Odds Ratio Estimates

Method	N	Minimum	Mean	Maximum
Unconditional	1000	0.00045	112.09	38038
Conditional	1000	0.021	4.20	195
Exact	1000	0.021	4.20	195
Firth	1000	0.018	4.89	71

Further examination of the data set all shows that the differences between the square root of the unconditional odds ratio estimates and the conditional estimates have mean –0.00019 and standard deviation 0.0008, verifying that the unconditional odds ratio is about the square of the conditional odds ratio. The conditional and exact conditional odds ratios are also nearly equal, with their differences having mean 3E–7 and standard deviation 6E–6. The differences between the Firth and the conditional odds ratios can be large (mean 0.69, standard deviation 5.40), but their relative differences, $\frac{Firth - Conditional}{Conditional}$, have mean 0.20 with standard deviation 0.19, so the largest differences occur with the larger estimates.

Example 51.13: Complementary Log-Log Model for Infection Rates

Antibodies produced in response to an infectious disease like malaria remain in the body after the individual has recovered from the disease. A serological test detects the presence or absence of such antibodies. An individual with such antibodies is called seropositive. In geographic areas where the disease is endemic, the inhabitants are at fairly constant risk of infection. The probability of an individual never having been infected in Y years is $\exp(-\mu Y)$, where μ is the mean number of infections per year (see the appendix of Draper, Voller, and Carpenter 1972). Rather than estimating the unknown μ, epidemiologists want to estimate the probability of a person living in the area being infected in one year. This infection rate γ is given by

$$\gamma = 1 - e^{-\mu}$$

The following statements create the data set sero, which contains the results of a serological survey of malarial infection. Individuals of nine age groups (Group) were tested. The variable A represents the midpoint of the age range for each age group. The variable N represents the number of individuals tested in each age group, and the variable R represents the number of individuals that are seropositive.

```
data sero;
   input Group A N R;
   X=log(A);
   label X='Log of Midpoint of Age Range';
   datalines;
1  1.5  123  8
2  4.0  132  6
3  7.5  182 18
4 12.5  140 14
5 17.5  138 20
6 25.0  161 39
7 35.0  133 19
8 47.0   92 25
9 60.0   74 44
;
```

For the ith group with the age midpoint A_i, the probability of being seropositive is $p_i = 1 - \exp(-\mu A_i)$. It follows that

$$\log(-\log(1 - p_i)) = \log(\mu) + \log(A_i)$$

By fitting a binomial model with a complementary log-log link function and by using X=log(A) as an offset term, you can estimate $\alpha = \log(\mu)$ as an intercept parameter. The following statements invoke PROC LOGISTIC to compute the maximum likelihood estimate of α. The LINK=CLOGLOG option is specified to request the complementary log-log link function. Also specified is the CLPARM=PL option, which requests the profile-likelihood confidence limits for α.

```
proc logistic data=sero;
   model R/N= / offset=X
                link=cloglog
                clparm=pl
                scale=none;
title 'Constant Risk of Infection';
run;
```

Results of fitting this constant risk model are shown in Output 51.13.1.

Output 51.13.1 Modeling Constant Risk of Infection

```
                          Constant Risk of Infection

                              Model Information

Data Set                          WORK.SERO
Response Variable (Events)        R
Response Variable (Trials)        N
Offset Variable                   X                    Log of Midpoint of Age Range
Model                             binary cloglog
Optimization Technique            Fisher's scoring

                    Number of Observations Read           9
                    Number of Observations Used           9
                    Sum of Frequencies Read            1175
                    Sum of Frequencies Used            1175

                                Response Profile

                       Ordered     Binary           Total
                         Value     Outcome      Frequency

                             1     Event              193
                             2     Nonevent           982

                     Intercept-Only Model Convergence Status

                  Convergence criterion (GCONV=1E-8) satisfied.

                                -2 Log L = 967.1158


               Deviance and Pearson Goodness-of-Fit Statistics

          Criterion          Value       DF     Value/DF     Pr > ChiSq

          Deviance         41.5032        8       5.1879         <.0001
          Pearson          50.6883        8       6.3360         <.0001

                    Number of events/trials observations: 9
```

Output 51.13.1 *continued*

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-4.6605	0.0725	4133.5626	<.0001
X	1	1.0000	0	.	.

Profile Likelihood Confidence Interval for Parameters

Parameter	Estimate	95% Confidence Limits	
Intercept	-4.6605	-4.8057	-4.5219

Output 51.13.1 shows that the maximum likelihood estimate of $\alpha = \log(\mu)$ and its estimated standard error are $\widehat{\alpha} = -4.6605$ and $\widehat{\sigma}_{\widehat{\alpha}} = 0.0725$, respectively. The infection rate is estimated as

$$\widehat{\gamma} = 1 - e^{-\widehat{\mu}} = 1 - e^{-e^{\widehat{\beta}_0}} = 1 - e^{-e^{-4.6605}} = 0.00942$$

The 95% confidence interval for γ, obtained by back-transforming the 95% confidence interval for α, is (0.0082, 0.0108); that is, there is a 95% chance that, in repeated sampling, the interval of 8 to 11 infections per thousand individuals contains the true infection rate.

The goodness-of-fit statistics for the constant risk model are statistically significant ($p < 0.0001$), indicating that the assumption of constant risk of infection is not correct. You can fit a more extensive model by allowing a separate risk of infection for each age group. Suppose μ_i is the mean number of infections per year for the ith age group. The probability of seropositive for the ith group with the age midpoint A_i is $p_i = 1 - \exp(-\mu_i A_i)$, so that

$$\log(-\log(1 - p_i)) = \log(\mu_i) + \log(A_i)$$

In the following statements, a complementary log-log model is fit containing Group as an explanatory classification variable with the GLM coding (so that a dummy variable is created for each age group), no intercept term, and X=log(A) as an offset term. The ODS OUTPUT statement saves the estimates and their 95% profile-likelihood confidence limits to the ClparmPL data set. Note that $\log(\mu_i)$ is the regression parameter associated with Group$= i$.

```
proc logistic data=sero;
   ods output ClparmPL=ClparmPL;
   class Group / param=glm;
   model R/N=Group / noint
                     offset=X
                     link=cloglog
                     clparm=pl;
   title 'Infectious Rates and 95% Confidence Intervals';
run;
```

Results of fitting the model with a separate risk of infection are shown in Output 51.13.2.

Output 51.13.2 Modeling Separate Risk of Infection

Infectious Rates and 95% Confidence Intervals

Analysis of Maximum Likelihood Estimates

Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Group	1	1	−3.1048	0.3536	77.0877	<.0001
Group	2	1	−4.4542	0.4083	119.0164	<.0001
Group	3	1	−4.2769	0.2358	328.9593	<.0001
Group	4	1	−4.7761	0.2674	319.0600	<.0001
Group	5	1	−4.7165	0.2238	443.9920	<.0001
Group	6	1	−4.5012	0.1606	785.1350	<.0001
Group	7	1	−5.4252	0.2296	558.1114	<.0001
Group	8	1	−4.9987	0.2008	619.4666	<.0001
Group	9	1	−4.1965	0.1559	724.3157	<.0001
X		1	1.0000	0	.	.

Profile Likelihood Confidence Interval for Parameters

Parameter		Estimate	95% Confidence Limits	
Group	1	−3.1048	−3.8880	−2.4833
Group	2	−4.4542	−5.3769	−3.7478
Group	3	−4.2769	−4.7775	−3.8477
Group	4	−4.7761	−5.3501	−4.2940
Group	5	−4.7165	−5.1896	−4.3075
Group	6	−4.5012	−4.8333	−4.2019
Group	7	−5.4252	−5.9116	−5.0063
Group	8	−4.9987	−5.4195	−4.6289
Group	9	−4.1965	−4.5164	−3.9037

For the first age group (Group=1), the point estimate of $\log(\mu_1)$ is -3.1048, which transforms into an infection rate of $1 - \exp(-\exp(-3.1048)) = 0.0438$. A 95% confidence interval for this infection rate is obtained by transforming the 95% confidence interval for $\log(\mu_1)$. For the first age group, the lower and upper confidence limits are $1 - \exp(-\exp(-3.8880) = 0.0203$ and $1 - \exp(-\exp(-2.4833)) = 0.0801$, respectively; that is, there is a 95% chance that, in repeated sampling, the interval of 20 to 80 infections per thousand individuals contains the true infection rate. The following statements perform this transformation on the estimates and confidence limits saved in the ClparmPL data set; the resulting estimated infection rates in one year's time for each age group are displayed in Table 51.10. Note that the infection rate for the first age group is high compared to that of the other age groups.

```
data ClparmPL;
   set ClparmPL;
   Estimate=round( 1000*( 1-exp(-exp(Estimate)) ) );
   LowerCL =round( 1000*( 1-exp(-exp(LowerCL )) ) );
   UpperCL =round( 1000*( 1-exp(-exp(UpperCL )) ) );
run;
```

Table 51.10 Infection Rate in One Year

Age Group	Number Infected per 1,000 People		
	Point Estimate	95% Confidence Limits	
		Lower	Upper
1	44	20	80
2	12	5	23
3	14	8	21
4	8	5	14
5	9	6	13
6	11	8	15
7	4	3	7
8	7	4	10
9	15	11	20

Example 51.14: Complementary Log-Log Model for Interval-Censored Survival Times

Often survival times are not observed more precisely than the interval (for instance, a day) within which the event occurred. Survival data of this form are known as grouped or interval-censored data. A discrete analog of the continuous proportional hazards model (Prentice and Gloeckler 1978; Allison 1982) is used to investigate the relationship between these survival times and a set of explanatory variables.

Suppose T_i is the discrete survival time variable of the ith subject with covariates $\boldsymbol{x}_i$. The discrete-time hazard rate λ_{it} is defined as

$$\lambda_{it} = \Pr(T_i = t \mid T_i \geq t, \boldsymbol{x}_i), \quad t = 1, 2, \ldots$$

Using elementary properties of conditional probabilities, it can be shown that

$$\Pr(T_i = t) = \lambda_{it} \prod_{j=1}^{t-1} (1 - \lambda_{ij}) \quad \text{and} \quad \Pr(T_i > t) = \prod_{j=1}^{t} (1 - \lambda_{ij})$$

Suppose t_i is the observed survival time of the ith subject. Suppose $\delta_i = 1$ if $T_i = t_i$ is an event time and 0 otherwise. The likelihood for the grouped survival data is given by

$$\begin{aligned} L &= \prod_i [\Pr(T_i = t_i)]^{\delta_i} [\Pr(T_i > t_i)]^{1-\delta_i} \\ &= \prod_i \left(\frac{\lambda_{it_i}}{1 - \lambda_{it_i}} \right)^{\delta_i} \prod_{j=1}^{t_i} (1 - \lambda_{ij}) \\ &= \prod_i \prod_{j=1}^{t_i} \left(\frac{\lambda_{ij}}{1 - \lambda_{ij}} \right)^{y_{ij}} (1 - \lambda_{ij}) \end{aligned}$$

where $y_{ij} = 1$ if the ith subject experienced an event at time $T_i = j$ and 0 otherwise.

Note that the likelihood L for the grouped survival data is the same as the likelihood of a binary response model with event probabilities λ_{ij}. If the data are generated by a continuous-time proportional hazards model, Prentice and Gloeckler (1978) have shown that

$$\lambda_{ij} = 1 - \exp(-\exp(\alpha_j + \boldsymbol{\beta}' \boldsymbol{x}_i))$$

which can be rewritten as

$$\log(-\log(1 - \lambda_{ij})) = \alpha_j + \boldsymbol{\beta}' \boldsymbol{x}_i$$

where the coefficient vector $\boldsymbol{\beta}$ is identical to that of the continuous-time proportional hazards model, and α_j is a constant related to the conditional survival probability in the interval defined by $T_i = j$ at $\boldsymbol{x}_i = \mathbf{0}$. The grouped data survival model is therefore equivalent to the binary response model with complementary log-log link function. To fit the grouped survival model by using PROC LOGISTIC, you must treat each discrete time unit for each subject as a separate observation. For each of these observations, the response is dichotomous, corresponding to whether or not the subject died in the time unit.

Consider a study of the effect of insecticide on flour beetles. Four different concentrations of an insecticide were sprayed on separate groups of flour beetles. The following DATA step saves the number of male and female flour beetles dying in successive intervals in the data set beetles:

```
data beetles(keep=time sex conc freq);
   input time m20 f20 m32 f32 m50 f50 m80 f80;
   conc=.20; freq= m20; sex=1; output;
             freq= f20; sex=2; output;
   conc=.32; freq= m32; sex=1; output;
             freq= f32; sex=2; output;
   conc=.50; freq= m50; sex=1; output;
             freq= f50; sex=2; output;
   conc=.80; freq= m80; sex=1; output;
             freq= f80; sex=2; output;
   datalines;
 1   3   0  7  1  5  0  4  2
 2  11   2 10  5  8  4 10  7
 3  10   4 11 11 11  6  8 15
 4   7   8 16 10 15  6 14  9
 5   4   9  3  5  4  3  8  3
 6   3   3  2  1  2  1  2  4
 7   2   0  1  0  1  1  1  1
 8   1   0  0  1  1  4  0  1
 9   0   0  1  1  0  0  0  0
10   0   0  0  0  0  0  1  1
11   0   0  0  0  1  1  0  0
12   1   0  0  0  0  1  0  0
13   1   0  0  0  0  1  0  0
14 101 126 19 47  7 17  2  4
;
```

The data set beetles contains four variables: time, sex, conc, and freq. The variable time represents the interval death time; for example, time=2 is the interval between day 1 and day 2. Insects surviving the duration (13 days) of the experiment are given a time value of 14. The variable sex

represents the sex of the insects (1=male, 2=female), conc represents the concentration of the insecticide (mg/cm^2), and freq represents the frequency of the observations.

To use PROC LOGISTIC with the grouped survival data, you must expand the data so that each beetle has a separate record for each day of survival. A beetle that died in the third day (time=3) would contribute three observations to the analysis, one for each day it was alive at the beginning of the day. A beetle that survives the 13-day duration of the experiment (time=14) would contribute 13 observations.

The following DATA step creates a new data set named days containing the beetle-day observations from the data set beetles. In addition to the variables sex, conc, and freq, the data set contains an outcome variable y and a classification variable day. The variable y has a value of 1 if the observation corresponds to the day that the beetle died, and it has a value of 0 otherwise. An observation for the first day will have a value of 1 for day; an observation for the second day will have a value of 2 for day, and so on. For instance, Output 51.14.1 shows an observation in the beetles data set with time=3, and Output 51.14.2 shows the corresponding beetle-day observations in the data set days.

```
data days;
   set beetles;
   do day=1 to time;
      if (day < 14) then do;
         y= (day=time);
         output;
      end;
   end;
run;
```

Output 51.14.1 An Observation with Time=3 in Beetles Data Set

Obs	time	conc	freq	sex
17	3	0.2	10	1

Output 51.14.2 Corresponding Beetle-Day Observations in Days

Obs	time	conc	freq	sex	day	y
25	3	0.2	10	1	1	0
26	3	0.2	10	1	2	0
27	3	0.2	10	1	3	1

The following statements invoke PROC LOGISTIC to fit a complementary log-log model for binary data with the response variable Y and the explanatory variables day, sex, and conc. Specifying the EVENT= option ensures that the event (y=1) probability is modeled. The GLM coding in the CLASS statement creates an indicator column in the design matrix for each level of day. The coefficients of the indicator effects for day can be used to estimate the baseline survival function. The NOINT option is specified to prevent any redundancy in estimating the coefficients of day. The Newton-Raphson algorithm is used for the maximum likelihood estimation of the parameters.

```
proc logistic data=days outest=est1;
   class day / param=glm;
   model y(event='1')= day sex conc
         / noint link=cloglog technique=newton;
   freq freq;
run;
```

Results of the model fit are given in Output 51.14.3. Both sex and conc are statistically significant for the survival of beetles sprayed by the insecticide. Female beetles are more resilient to the chemical than male beetles, and increased concentration of the insecticide increases its effectiveness.

Output 51.14.3 Parameter Estimates for the Grouped Proportional Hazards Model

Analysis of Maximum Likelihood Estimates

Parameter		DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
day	1	1	−3.9314	0.2934	179.5602	<.0001
day	2	1	−2.8751	0.2412	142.0596	<.0001
day	3	1	−2.3985	0.2299	108.8833	<.0001
day	4	1	−1.9953	0.2239	79.3960	<.0001
day	5	1	−2.4920	0.2515	98.1470	<.0001
day	6	1	−3.1060	0.3037	104.5799	<.0001
day	7	1	−3.9704	0.4230	88.1107	<.0001
day	8	1	−3.7917	0.4007	89.5233	<.0001
day	9	1	−5.1540	0.7316	49.6329	<.0001
day	10	1	−5.1350	0.7315	49.2805	<.0001
day	11	1	−5.1131	0.7313	48.8834	<.0001
day	12	1	−5.1029	0.7313	48.6920	<.0001
day	13	1	−5.0951	0.7313	48.5467	<.0001
sex		1	−0.5651	0.1141	24.5477	<.0001
conc		1	3.0918	0.2288	182.5665	<.0001

The coefficients of parameters for the day variable are the maximum likelihood estimates of $\alpha_1, \ldots, \alpha_{13}$, respectively. The baseline survivor function $S_0(t)$ is estimated by

$$\hat{S}_0(t) = \widehat{\Pr}(T > t) = \prod_{j \le t} \exp(-\exp(\widehat{\alpha}_j))$$

and the survivor function for a given covariate pattern (sex=x_1 and conc=x_2) is estimated by

$$\hat{S}(t) = [\hat{S}_0(t)]^{\exp(-0.5651x_1 + 3.0918x_2)}$$

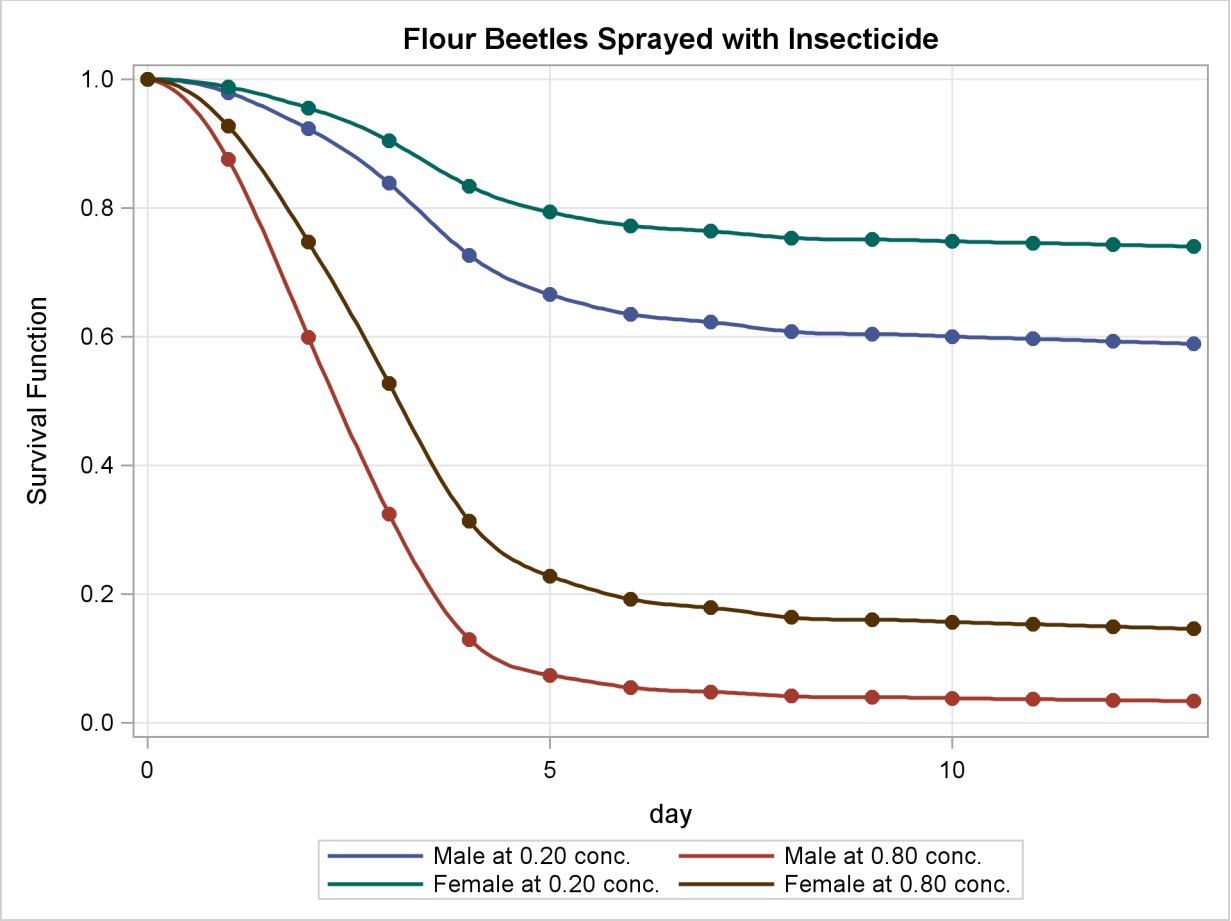
The following statements compute the survival curves for male and female flour beetles exposed to the insecticide in concentrations of 0.20 mg/cm^2 and 0.80 mg/cm^2.

```
data one (keep=day survival element s_m20 s_f20 s_m80 s_f80);
   array dd day1-day13;
   array sc[4] m20 f20 m80 f80;
   array s_sc[4] s_m20 s_f20 s_m80 s_f80 (1 1 1 1);
   set est1;
   m20= exp(sex + .20 * conc);
   f20= exp(2 * sex + .20 * conc);
   m80= exp(sex + .80 * conc);
   f80= exp(2 * sex + .80 * conc);
   survival=1;
   day=0;
   output;
   do over dd;
      element= exp(-exp(dd));
      survival= survival * element;
      do i=1 to 4;
        s_sc[i] = survival ** sc[i];
      end;
      day + 1;
      output;
   end;
run;
```

Instead of plotting the curves as step functions, the following statements use the PBSPLINE statement in the SGPLOT procedure to smooth the curves with a penalized B-spline. See Chapter 90, "The TRANSREG Procedure," for details about the implementation of the penalized B-spline method. The SAS autocall macro MODSTYLE is specified to change the default linestyles and marker symbols for the plot. For more information about autocall libraries, see *SAS Macro Language: Reference*. The smoothed survival curves are displayed in Output 51.14.4.

```
%modstyle(name=LogiStyle,parent=Statistical,markers=circlefilled,linestyles=solid);
ods listing style=LogiStyle;
proc sgplot data=one;
   title 'Flour Beetles Sprayed with Insecticide';
   xaxis grid integer;
   yaxis grid label='Survival Function';
   pbspline y=s_m20 x=day /
      legendlabel = "Male at 0.20 conc." name="pred1";
   pbspline y=s_m80 x=day /
      legendlabel = "Male at 0.80 conc." name="pred2";
   pbspline y=s_f20 x=day /
      legendlabel = "Female at 0.20 conc." name="pred3";
   pbspline y=s_f80 x=day /
      legendlabel = "Female at 0.80 conc." name="pred4";
   discretelegend "pred1" "pred2" "pred3" "pred4" / across=2;
run;
```

Output 51.14.4 Predicted Survival at Insecticide Concentrations of 0.20 and 0.80 mg/cm^2



The probability of survival is displayed on the vertical axis. Notice that most of the insecticide effect occurs by day 6 for both the high and low concentrations.

Example 51.15: Scoring Data Sets with the SCORE Statement

This example first illustrates the syntax used for scoring data sets, then uses a previously scored data set to score a new data set. A generalized logit model is fit to the remote-sensing data set used in the section "Example 31.4: Linear Discriminant Analysis of Remote-Sensing Data on Crops" on page 1469 of Chapter 31, "The DISCRIM Procedure," to illustrate discrimination and classification methods. In the following DATA step, the response variable is Crop and the prognostic factors are x1 through x4.

```
data Crops;
   length Crop $ 10;
   infile datalines truncover;
   input Crop $ @@;
   do i=1 to 3;
     input x1-x4 @@;
     if (x1 ^= .) then output;
   end;
   input;
   datalines;
Corn       16 27 31 33  15 23 30 30  16 27 27 26
Corn       18 20 25 23  15 15 31 32  15 32 32 15
Corn       12 15 16 73
Soybeans   20 23 23 25  24 24 25 32  21 25 23 24
Soybeans   27 45 24 12  12 13 15 42  22 32 31 43
Cotton     31 32 33 34  29 24 26 28  34 32 28 45
Cotton     26 25 23 24  53 48 75 26  34 35 25 78
Sugarbeets 22 23 25 42  25 25 24 26  34 25 16 52
Sugarbeets 54 23 21 54  25 43 32 15  26 54  2 54
Clover     12 45 32 54  24 58 25 34  87 54 61 21
Clover     51 31 31 16  96 48 54 62  31 31 11 11
Clover     56 13 13 71  32 13 27 32  36 26 54 32
Clover     53 08 06 54  32 32 62 16
;
```

In the following statements, you specify a SCORE statement to use the fitted model to score the Crops data. The data together with the predicted values are saved in the data set Score1. The output from the PLOTS option is discussed at the end of this section.

```
ods graphics on;
proc logistic data=Crops plots(only)=effect(x=x3);
   model Crop=x1-x4 / link=glogit;
   score out=Score1;
run;
ods graphics off;
```

In the following statements, the model is fit again, the data and the predicted values are saved into the data set Score2, and the OUTMODEL= option saves the fitted model information in the permanent SAS data set sasuser.CropModel:

```
proc logistic data=Crops outmodel=sasuser.CropModel;
   model Crop=x1-x4 / link=glogit;
   score data=Crops out=Score2;
run;
```

To score data without refitting the model, specify the INMODEL= option to identify a previously saved SAS data set of model information. In the following statements, the model is read from the sasuser.CropModel data set, and the data and the predicted values are saved in the data set Score3. Note that the data set being scored does not have to include the response variable.

```
proc logistic inmodel=sasuser.CropModel;
   score data=Crops out=Score3;
run;
```

To set prior probabilities on the responses, specify the PRIOR= option to identify a SAS data set containing the response levels and their priors. In the following statements, the Prior data set contains the values of the response variable (because this example uses single-trial MODEL syntax) and a _PRIOR_ variable containing values proportional to the default priors. The data and the predicted values are saved in the data set Score4.

```
data Prior;
   length Crop $10.;
   input Crop _PRIOR_;
   datalines;
Clover     11
Corn        7
Cotton      6
Soybeans    6
Sugarbeets  6
;
proc logistic inmodel=sasuser.CropModel;
   score data=Crops prior=prior out=Score4 fitstat;
run;
```

The "Fit Statistics for SCORE Data" table displayed in Output 51.15.1 shows that 47.22% of the observations are misclassified.

Output 51.15.1 Fit Statistics for Data Set Prior

Fit Statistics for SCORE Data

Data Set	Total Frequency	Log Likelihood	Misclassification Rate
WORK.CROPS	36	-32.2247	0.4722

The data sets Score1, Score2, Score3, and Score4 are identical. The following statements display the scoring results in Output 51.15.2:

```
proc freq data=Score1;
   table F_Crop*I_Crop / nocol nocum nopercent;
run;
```

Output 51.15.2 Classification of Data Used for Scoring

Table of F_Crop by I_Crop

F_Crop(From: Crop) / Frequency / Row Pct	Clover	Corn	Cotton	Soybeans	Sugarbeets	Total
Clover	6 54.55	0 0.00	2 18.18	2 18.18	1 9.09	11
Corn	0 0.00	7 100.00	0 0.00	0 0.00	0 0.00	7
Cotton	4 66.67	0 0.00	1 16.67	1 16.67	0 0.00	6
Soybeans	1 16.67	1 16.67	1 16.67	3 50.00	0 0.00	6
Sugarbeets	2 33.33	0 0.00	0 0.00	2 33.33	2 33.33	6
Total	13	8	4	8	3	36

The following statements use the previously fitted and saved model in the sasuser.CropModel data set to score the observations in a new data set, Test. The results of scoring the test data are saved in the ScoredTest data set and displayed in Output 51.15.3.

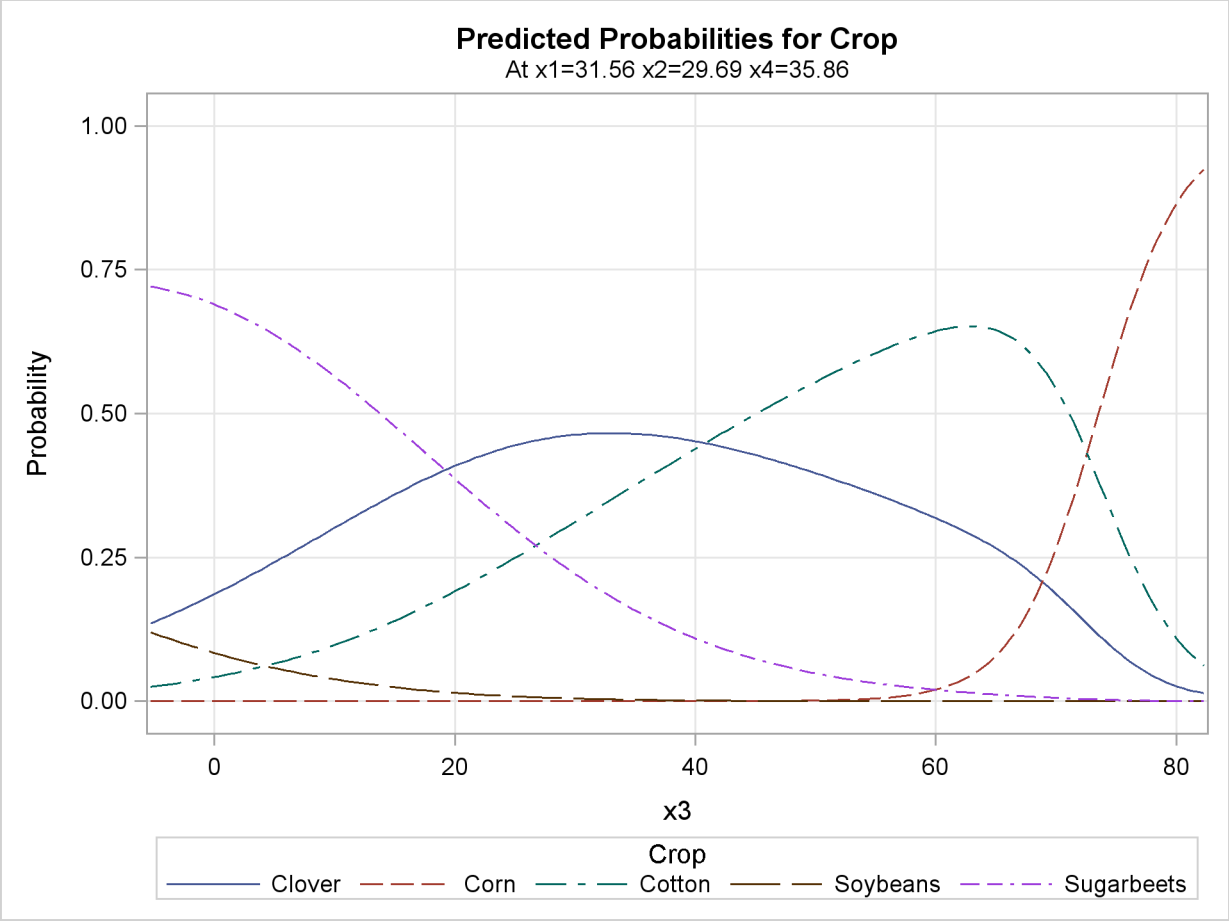
```
data Test;
   input Crop $ 1-10 x1-x4;
   datalines;
Corn       16 27 31 33
Soybeans   21 25 23 24
Cotton     29 24 26 28
Sugarbeets 54 23 21 54
Clover     32 32 62 16
;

proc logistic noprint inmodel=sasuser.CropModel;
   score data=Test out=ScoredTest;

proc print data=ScoredTest label noobs;
   var F_Crop I_Crop P_Clover P_Corn P_Cotton P_Soybeans P_Sugarbeets;
run;
```

Output 51.15.3 Classification of Test Data

From: Crop	Into: Crop	Predicted Probability: Crop=Clover	Predicted Probability: Crop=Corn	Predicted Probability: Crop=Cotton	Predicted Probability: Crop=Soybeans	Predicted Probability: Crop=Sugarbeets
Corn	Corn	0.00342	0.90067	0.00500	0.08675	0.00416
Soybeans	Soybeans	0.04801	0.03157	0.02865	0.82933	0.06243
Cotton	Clover	0.43180	0.00015	0.21267	0.07623	0.27914
Sugarbeets	Clover	0.66681	0.00000	0.17364	0.00000	0.15955
Clover	Cotton	0.41301	0.13386	0.43649	0.00033	0.01631

The PLOTS(ONLY)= option specified in the first PROC LOGISTIC invocation produces a plot of the model-predicted probabilities versus X3, holding the other three covariates fixed at their means (Output 51.15.4). This plot shows how the value of X3 affects the probabilities of the various crops when the other prognostic factors are fixed at their means. If you are interested in the effect of X3 when the other covariates are fixed at a certain level—say, 10—specify **effect(x=x3 at(x1=10 x2=10 x4=10))**.

Output 51.15.4 Model-Predicted Probabilities



References

Agresti, A. (1984), *Analysis of Ordinal Categorical Data*, New York: John Wiley & Sons.

Agresti, A. (1990), *Categorical Data Analysis*, New York: John Wiley & Sons.

Agresti, A. (1992), "A Survey of Exact Inference for Contingency Tables," *Statistical Science*, 7, 131–177.

Agresti, A. (2002), *Categorical Data Analysis*, Second Edition, New York: John Wiley & Sons.

Aitchison, J. and Silvey, S. (1957), "The Generalization of Probit Analysis to the Case of Multiple Responses," *Biometrika*, 44, 131–140.

Albert, A. and Anderson, J. A. (1984), "On the Existence of Maximum Likelihood Estimates in Logistic Regression Models," *Biometrika*, 71, 1–10.

Allison, P. D. (1982), "Discrete-Time Methods for the Analysis of Event Histories," in S. Leinhardt, ed., *Sociological Methods and Research*, volume 15, 61–98, San Francisco: Jossey-Bass.

Allison, P. D. (1999), *Logistic Regression Using the SAS System: Theory and Application*, Cary, NC: SAS Institute Inc.

Ashford, J. R. (1959), "An Approach to the Analysis of Data for Semi-Quantal Responses in Biology Response," *Biometrics*, 15, 573–581.

Bartolucci, A. A. and Fraser, M. D. (1977), "Comparative Step-Up and Composite Test for Selecting Prognostic Indicator Associated with Survival," *Biometrical Journal*, 19, 437–448.

Breslow, N. E. (1982), "Covariance Adjustment of Relative-Risk Estimates in Matched Studies," *Biometrics*, 38, 661–672.

Breslow, N. E. and Day, N. E. (1980), *Statistical Methods in Cancer Research, Volume I: The Analysis of Case-Control Studies*, IARC Scientific Publications, No. 32, Lyon, France: International Agency for Research on Cancer.

Collett, D. (2003), *Modelling Binary Data*, Second Edition, London: Chapman & Hall.

Cook, R. D. and Weisberg, S. (1982), *Residuals and Influence in Regression*, New York: Chapman & Hall.

Cox, D. R. (1970), *The Analysis of Binary Data*, New York: Chapman & Hall.

Cox, D. R. (1972), "Regression Models and Life Tables," *Journal of the Royal Statistical Society, Series B*, 20, 187–220, with discussion.

Cox, D. R. and Snell, E. J. (1989), *The Analysis of Binary Data*, Second Edition, London: Chapman & Hall.

DeLong, E. R., DeLong, D. M., and Clarke-Pearson, D. L. (1988), "Comparing the Areas under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach," *Biometrics*, 44, 837–845.

Draper, C. C., Voller, A., and Carpenter, R. G. (1972), "The Epidemiologic Interpretation of Serologic Data in Malaria," *American Journal of Tropical Medicine and Hygiene*, 21, 696–703.

Finney, D. J. (1947), "The Estimation from Individual Records of the Relationship between Dose and Quantal Response," *Biometrika*, 34, 320–334.

Firth, D. (1993), "Bias Reduction of Maximum Likelihood Estimates," *Biometrika*, 80, 27–38.

Fleiss, J. L. (1981), *Statistical Methods for Rates and Proportions*, Second Edition, New York: John Wiley & Sons.

Freeman, D. H., Jr. (1987), *Applied Categorical Data Analysis*, New York: Marcel Dekker.

Furnival, G. M. and Wilson, R. W. (1974), "Regression by Leaps and Bounds," *Technometrics*, 16, 499–511.

Gail, M. H., Lubin, J. H., and Rubinstein, L. V. (1981), "Likelihood Calculations for Matched Case-Control Studies and Survival Studies with Tied Death Times," *Biometrika*, 68, 703–707.

Hanley, J. A. and McNeil, B. J. (1982), "The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve," *Radiology*, 143, 29–36.

Harrell, F. E. (1986), "The LOGIST Procedure," *SUGI Supplemental Library Guide, Version 5 Edition*.

Heinze, G. (1999), *The Application of Firth's Procedure to Cox and Logistic Regression*, Technical Report 10/1999, update in January 2001, Section of Clinical Biometrics, Department of Medical Computer Sciences, University of Vienna.

Heinze, G. (2006), "A Comparative Investigation of Methods for Logistic Regression with Separated or Nearly Separated Data," *Statistics in Medicine*, 25, 4216–4226.

Heinze, G. and Schemper, M. (2002), "A Solution to the Problem of Separation in Logistic Regression," *Statistics in Medicine*, 21, 2409–2419.

Hirji, K. F. (1992), "Computing Exact Distributions for Polytomous Response Data," *Journal of the American Statistical Association*, 87, 487–492.

Hirji, K. F., Mehta, C. R., and Patel, N. R. (1987), "Computing Distributions for Exact Logistic Regression," *Journal of the American Statistical Association*, 82, 1110–1117.

Hirji, K. F., Mehta, C. R., and Patel, N. R. (1988), "Exact Inference for Matched Case-Control Studies," *Biometrics*, 44, 803–814.

Hirji, K. F., Tsiatis, A. A., and Mehta, C. R. (1989), "Median Unbiased Estimation for Binary Data," *American Statistician*, 43, 7–11.

Hosmer, D. W., Jr. and Lemeshow, S. (2000), *Applied Logistic Regression*, Second Edition, New York: John Wiley & Sons.

Howard, S. (1972), "Discussion on the Paper by Cox," in *Regression Models and Life Tables*, volume 34 of *Journal of the Royal Statistical Society, Series B*, 187–220, with discussion.

Izrael, D., Battaglia, A. A., Hoaglin, D. C., and Battaglia, M. P. (2002), "Use of the ROC Curve and the Bootstrap in Comparing Weighted Logistic Regression Models," in *Proceedings of the Twenty-seventh Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc., available at www2.sas.com/proceedings/sugi27/p248-27.pdf.

Lachin, J. M. (2000), *Biostatistical Methods: The Assessment of Relative Risks*, New York: John Wiley & Sons.

Lamotte, L. R. (2002), personal communication, June 2002.

Lancaster, H. O. (1961), "Significance Tests in Discrete Distributions," *Journal of the American Statistical Association*, 56, 223–234.

Lawless, J. F. and Singhal, K. (1978), "Efficient Screening of Nonnormal Regression Models," *Biometrics*, 34, 318–327.

Lee, E. T. (1974), "A Computer Program for Linear Logistic Regression Analysis," *Computer Programs in Biomedicine*, 80–92.

McCullagh, P. and Nelder, J. A. (1989), *Generalized Linear Models*, Second Edition, London: Chapman & Hall.

McFadden, D. (1974), "Conditional Logit Analysis of Qualitative Choice Behaviour," in P. Zarembka, ed., *Frontiers in Econometrics*, New York: Academic Press.

Mehta, C. R., Patel, N., and Senchaudhuri, P. (1992), "Exact Stratified Linear Rank Tests for Ordered Categorical and Binary Data," *Journal of Computational and Graphical Statistics*, 1, 21–40.

Mehta, C. R., Patel, N., and Senchaudhuri, P. (2000), "Efficient Monte Carlo Methods for Conditional Logistic Regression," *Journal of the American Statistical Association*, 95, 99–108.

Mehta, C. R. and Patel, N. R. (1995), "Exact Logistic Regression: Theory and Examples," *Statistics in Medicine*, 14, 2143–2160.

Moolgavkar, S. H., Lustbader, E. D., and Venzon, D. J. (1985), "Assessing the Adequacy of the Logistic Regression Model for Matched Case-Control Studies," *Statistics in Medicine*, 4, 425–435.

Naessens, J. M., Offord, K. P., Scott, W. F., and Daood, S. L. (1986), "The MCSTRAT Procedure," in *SUGI Supplemental Library User's Guide, Version 5 Edition*, 307–328, Cary, NC: SAS Institute Inc.

Nagelkerke, N. J. D. (1991), "A Note on a General Definition of the Coefficient of Determination," *Biometrika*, 78, 691–692.

Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized Linear Models," *Journal of the Royal Statistical Society, Series A*, 135, 370–384.

Pregibon, D. (1981), "Logistic Regression Diagnostics," *Annals of Statistics*, 9, 705–724.

Pregibon, D. (1984), "Data Analytic Methods for Matched Case-Control Studies," *Biometrics*, 40, 639–651.

Prentice, P. L. and Gloeckler, L. A. (1978), "Regression Analysis of Grouped Survival Data with Applications to Breast Cancer Data," *Biometrics*, 34, 57–67.

Press, S. J. and Wilson, S. (1978), "Choosing between Logistic Regression and Discriminant Analysis," *Journal of the American Statistical Association*, 73, 699–705.

Santner, T. J. and Duffy, E. D. (1986), "A Note on A. Albert and J. A. Anderson's Conditions for the Existence of Maximum Likelihood Estimates in Logistic Regression Models," *Biometrika*, 73, 755–758.

SAS Institute Inc. (1995), *Logistic Regression Examples Using the SAS System*, Cary, NC: SAS Institute Inc.

Stokes, M. E., Davis, C. S., and Koch, G. G. (2000), *Categorical Data Analysis Using the SAS System*, Second Edition, Cary, NC: SAS Institute Inc.

Storer, B. E. and Crowley, J. (1985), "A Diagnostic for Cox Regression and General Conditional Likelihoods," *Journal of the American Statistical Association*, 80, 139–147.

Venzon, D. J. and Moolgavkar, S. H. (1988), "A Method for Computing Profile-Likelihood Based Confidence Intervals," *Applied Statistics*, 37, 87–94.

Vollset, S. E., Hirji, K. F., and Afifi, A. A. (1991), "Evaluation of Exact and Asymptotic Interval Estimators in Logistic Analysis of Matched Case-Control Studies," *Biometrics*, 47, 1311–1325.

Walker, S. H. and Duncan, D. B. (1967), "Estimation of the Probability of an Event as a Function of Several Independent Variables," *Biometrika*, 54, 167–179.

Williams, D. A. (1982), "Extra-binomial Variation in Logistic Linear Models," *Applied Statistics*, 31, 144–148.

Chapter 52

The MCMC Procedure

Contents

Overview: MCMC Procedure **3478**
PROC MCMC Compared with Other SAS Procedures 3479
Getting Started: MCMC Procedure **3479**
Simple Linear Regression 3480
The Behrens-Fisher Problem 3488
Mixed-Effects Model 3492
Syntax: MCMC Procedure **3495**
PROC MCMC Statement 3496
ARRAY Statement 3508
BEGINCNST/ENDCNST Statement 3509
BEGINNODATA/ENDNODATA Statements 3511
BY Statement 3511
MODEL Statement 3512
PARMS Statement 3515
PRIOR/HYPERPRIOR Statement 3516
Programming Statements 3516
UDS Statement 3518
Details: MCMC Procedure **3522**
How PROC MCMC Works 3522
Blocking of Parameters 3523
Samplers 3524
Tuning the Proposal Distribution 3525
Initial Values of the Markov Chains 3528
Assignments of Parameters 3528
Standard Distributions 3530
Specifying a New Distribution 3541
Using Density Functions in the Programming Statements 3542
Truncation and Censoring 3544
Multivariate Density Functions 3546
Some Useful SAS Functions 3549
Matrix Functions in PROC MCMC 3551
Modeling Joint Likelihood 3556
Regenerating Diagnostics Plots 3557
Posterior Predictive Distribution 3560

Handling of Missing Data 3565
Floating Point Errors and Overflows 3565
Handling Error Messages 3568
Computational Resources 3570
Displayed Output 3571
ODS Table Names 3575
ODS Graphics 3577
Examples: MCMC Procedure **3578**
Example 52.1: Simulating Samples From a Known Density 3578
Example 52.2: Box-Cox Transformation 3583
Example 52.3: Generalized Linear Models 3592
Example 52.4: Nonlinear Poisson Regression Models 3605
Example 52.5: Random-Effects Models 3614
Example 52.6: Change Point Models 3630
Example 52.7: Exponential and Weibull Survival Analysis 3634
Example 52.8: Cox Models 3647
Example 52.9: Normal Regression with Interval Censoring 3664
Example 52.10: Constrained Analysis 3666
Example 52.11: Implement a New Sampling Algorithm 3672
Example 52.12: Using a Transformation to Improve Mixing 3683
Example 52.13: Gelman-Rubin Diagnostics 3693
References **3700**

Overview: MCMC Procedure

The MCMC procedure is a general purpose Markov chain Monte Carlo (MCMC) simulation procedure that is designed to fit Bayesian models. Bayesian statistics is different from traditional statistical methods such as frequentist or classical methods. For a short introduction to Bayesian analysis and related basic concepts, see Chapter 7, "Introduction to Bayesian Analysis Procedures." Also see the section "A Bayesian Reading List" on page 173 for a guide to Bayesian textbooks of varying degrees of difficulty.

In essence, Bayesian statistics treats parameters as unknown random variables, and it makes inferences based on the posterior distributions of the parameters. There are several advantages associated with this approach to statistical inference. Some of the advantages include its ability to use prior information and to directly answer specific scientific questions that can be easily understood. For further discussions of the relative advantages and disadvantages of Bayesian analysis, see the section "Bayesian Analysis: Advantages and Disadvantages" on page 149.

It follows from Bayes' theorem that a posterior distribution is the product of the likelihood function and the prior distribution of the parameter. In all but the simplest cases, it is very difficult to obtain the posterior distribution directly and analytically. Often, Bayesian methods rely on simulations to

generate sample from the desired posterior distribution and use the simulated draws to approximate the distribution and to make all of the inferences.

PROC MCMC is a flexible simulation-based procedure that is suitable for fitting a wide range of Bayesian models. To use the procedure, you need to specify a likelihood function for the data and a prior distribution for the parameters. You might also need to specify hyperprior distributions if you are fitting hierarchical models. PROC MCMC then obtains samples from the corresponding posterior distributions, produces summary and diagnostic statistics, and saves the posterior samples in an output data set that can be used for further analysis. You can analyze data that have any likelihood, prior, or hyperprior with PROC MCMC, as long as these functions are programmable using the SAS DATA step functions. The parameters can enter the model linearly or in any nonlinear functional form. The default algorithm that PROC MCMC uses is an adaptive blocked random walk Metropolis algorithm that uses a normal proposal distribution.

PROC MCMC Compared with Other SAS Procedures

PROC MCMC is unlike most other SAS/STAT procedures in that the nature of the statistical inference is Bayesian. You specify prior distributions for the parameters with PRIOR statements and the likelihood function for the data with MODEL statements. The procedure derives inferences from simulation rather than through analytic or numerical methods. You should expect slightly different answers from each run for the same problem, unless the same random number seed is used. The model specification is similar to PROC NLIN, and PROC MCMC shares much of the syntax of PROC NLMIXED.

Note that you can also carry out a Bayesian analysis with the GENMOD, PHREG, and LIFEREG procedures for generalized linear models, accelerated life failure models, Cox regression models, and piecewise constant baseline hazard models (also known as piecewise exponential models). See Chapter 37, "The GENMOD Procedure," Chapter 64, "The PHREG Procedure," and Chapter 48, "The LIFEREG Procedure."

Getting Started: MCMC Procedure

There are three examples in this "Getting Started" section: a simple linear regression, the Behrens-Fisher estimation problem, and a random effects model. The regression model is chosen for its simplicity; the Behrens-Fisher problem illustrates some advantages of the Bayesian approach; and the random effects model is one of the most prevalently used models.

Keep in mind that PARMS statements declare the parameters in the model, PRIOR statements declare the prior distributions, and MODEL statements declare the likelihood for the data. In most cases, you do not need to supply initial values. The procedure advises you if it is unable to generate starting values for the Markov chain.

Simple Linear Regression

This section illustrates some basic features of PROC MCMC by using a linear regression model. The model is as follows:

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

for the observations $i = 1, 2, \ldots, n$.

The following statements create a SAS data set with measurements of Height and Weight for a group of children:

```
title 'Simple Linear Regression';

data Class;
   input Name $ Height Weight @@;
   datalines;
Alfred  69.0 112.5   Alice  56.5  84.0   Barbara 65.3  98.0
Carol   62.8 102.5   Henry  63.5 102.5   James   57.3  83.0
Jane    59.8  84.5   Janet  62.5 112.5   Jeffrey 62.5  84.0
John    59.0  99.5   Joyce  51.3  50.5   Judy    64.3  90.0
Louise  56.3  77.0   Mary   66.5 112.0   Philip  72.0 150.0
Robert  64.8 128.0   Ronald 67.0 133.0   Thomas  57.5  85.0
William 66.5 112.0
;
```

The equation of interest is as follows:

$$\text{Weight}_i = \beta_0 + \beta_1 \text{Height}_i + \epsilon_i$$

The observation errors, ϵ_i, are assumed to be independent and identically distributed with a normal distribution with mean zero and variance σ^2.

$$\text{Weight}_i \sim \text{normal}(\beta_0 + \beta_1 \text{Height}_i, \sigma^2)$$

The likelihood function for each of the Weight, which is specified in the MODEL statement, is as follows:

$$p(\text{Weight}|\beta_0, \beta_1, \sigma^2, \text{Height}_i) = \phi(\beta_0 + \beta_1 \text{Height}_i, \sigma^2)$$

where $p(\cdot|\cdot)$ denotes a conditional probability density and ϕ is the normal density. There are three parameters in the likelihood: β_0, β_1, and σ^2. You use the PARMS statement to indicate that these are the parameters in the model.

Suppose that you want to use the following three prior distributions on each of the parameters:

$$\begin{aligned}
\pi(\beta_0) &= \phi(0, \text{var} = 1e6) \\
\pi(\beta_1) &= \phi(0, \text{var} = 1e6) \\
\pi(\sigma^2) &= f_{i\Gamma}(\text{shape} = 3/10, \text{scale} = 10/3)
\end{aligned}$$

where $\pi(\cdot)$ indicates a prior distribution and $f_{i\Gamma}$ is the density function for the inverse-gamma distribution. The normal priors on β_0 and β_1 have large variances, expressing your lack of knowledge about the regression coefficients. The priors correspond to an equal-tail 95% credible intervals of approximately $(-2000, 2000)$ for β_0 and β_1. Priors of this type are often called *vague* or *diffuse* priors. See the section "Prior Distributions" on page 144 for more information. Typically diffuse prior distributions have little influence on the posterior distribution and are appropriate when stronger prior information about the parameters is not available.

A frequently used diffuse prior for the variance parameter σ^2 is the *inverse-gamma* distribution. With a shape parameter of $3/10$ and a scale parameter of $10/3$, this prior corresponds to an equal-tail 95% credible interval of $(1.7, 1e6)$, with the mode at 2.5641 for σ^2. Alternatively, you can use any other positive prior, meaning that the density support is positive on this variance component. For example, you can use the gamma prior.

According to Bayes' theorem, the likelihood function and prior distributions determine the posterior (joint) distribution of β_0, β_1, and σ^2 as follows:

$$\pi(\beta_0, \beta_1, \sigma^2|\text{Weight}, \text{Height}) \quad \propto \quad \pi(\beta_0)\pi(\beta_1)\pi(\sigma^2)\, p(\text{Weight}|\beta_0, \beta_1, \sigma^2, \text{Height})$$

You do not need to know the form of the posterior distribution when you use PROC MCMC. PROC MCMC automatically obtains samples from the desired posterior distribution, which is determined by the prior and likelihood you supply.

The following statements fit this linear regression model with diffuse prior information:

```
ods graphics on;
proc mcmc data=class outpost=classout nmc=50000 thin=5 seed=246810;
   parms beta0 0 beta1 0;
   parms sigma2 1;
   prior beta0 beta1 ~ normal(mean = 0, var = 1e6);
   prior sigma2 ~ igamma(shape = 3/10, scale = 10/3);
   mu = beta0 + beta1*height;
   model weight ~ n(mu, var = sigma2);
run;
ods graphics off;
```

The ODS GRAPHICS ON statement invokes the ODS Graphics environment and displays the diagnostic plots, such as the trace and autocorrelation function plots of the posterior samples. For more information about ODS, see Chapter 21, "Statistical Graphics Using ODS."

The PROC MCMC statement invokes the procedure and specifies the input data set class. The output data set classout contains the posterior samples for all of the model parameters. The NMC= option specifies the number of posterior simulation iterations. The THIN= option controls the thinning of the Markov chain and specifies that one of every 5 samples is kept. Thinning is often used to reduce the correlations among posterior sample draws. In this example, 10,000 simulated values are saved in the classout data set. The SEED= option specifies a seed for the random number generator, which guarantees the reproducibility of the random stream. For more information about Markov chain sample size, burn-in, and thinning, see the section "Burn-in, Thinning, and Markov Chain Samples" on page 155.

The PARMS statements identify the three parameters in the model: beta0, beta1, and sigma2. Each statement also forms a block of parameters, where the parameters are updated simultaneously in each iteration. In this example, beta0 and beta1 are sampled jointly, conditional on sigma2; and sigma2 is sampled conditional on fixed values of beta0 and beta1. In simple regression models such as this, you expect the parameters beta0 and beta1 to have high posterior correlations, and placing them both in the same block improves the mixing of the chain—that is, the efficiency that the posterior parameter space is explored by the Markov chain. For more information, see the section "Blocking of Parameters" on page 3523. The PARMS statements also assign initial values to the parameters (see the section "Initial Values of the Markov Chains" on page 3528). The regression parameters are given 0 as their initial values, and the scale parameter sigma2 starts at value 1. If you do not provide initial values, the procedure chooses starting values for every parameter.

The PRIOR statements specify prior distributions for the parameters. The parameters beta0 and beta1 both share the same prior—a normal prior with mean 0 and variance $1e6$. The parameter sigma2 has an inverse-gamma distribution with a shape parameter of 3/10 and a scale parameter of 10/3. For a list of standard distributions that PROC MCMC supports, see the section "Standard Distributions" on page 3530.

The mu assignment statement calculates the expected value of Weight as a linear function of Height. The MODEL statement uses the shorthand notation, **n**, for the normal distribution to indicate that the response variable, Weight, is normally distributed with parameters mu and sigma2. The functional argument MEAN= in the normal distribution is optional, but you have to indicate whether sigma2 is a variance (VAR=), a standard deviation (SD=), or a precision (PRECISION=) parameter. See Table 52.2 in the section "MODEL Statement" on page 3512 for distribution specifications.

The distribution parameters can contain expressions. For example, you can write the MODEL statement as follows:

```
model weight ~ n(beta0 + beta1*height, var = sigma2);
```

Before you do any posterior inference, it is essential that you examine the convergence of the Markov chain (see the section "Assessing Markov Chain Convergence" on page 156). You cannot make valid inferences if the Markov chain has not converged. A very effective convergence diagnostic tool is the trace plot. Although PROC MCMC produces graphs at the end of the procedure output (see Figure 52.6), you should visually examine the convergence graph first.

The first table that PROC MCMC produces is the "Number of Observations" table, as shown in Figure 52.1. This table lists the number of observations read from the DATA= data set and the number of non-missing observations used in the analysis.

Figure 52.1 Observation Information

```
                    Simple Linear Regression

                       The MCMC Procedure

              Number of Observations Read          19
              Number of Observations Used          19
```

The "Parameters" table, shown in Figure 52.2, lists the names of the parameters, the blocking information (see the section "Blocking of Parameters" on page 3523), the sampling method used, the starting values (the section "Initial Values of the Markov Chains" on page 3528), and the prior distributions. You should to check this table to ensure that you have specified the parameters correctly, especially for complicated models.

Figure 52.2 Parameter Information

Parameters

Block	Parameter	Sampling Method	Initial Value	Prior Distribution
1	beta0	N-Metropolis	0	normal(mean = 0, var = 1e6)
1	beta1	N-Metropolis	0	normal(mean = 0, var = 1e6)
2	sigma2	N-Metropolis	1.0000	igamma(shape = 3/10, scale = 10/3)

The "Tuning History" table, shown in Figure 52.3, shows how the tuning stage progresses for the multivariate random walk Metropolis algorithm used by PROC MCMC to generate samples from the posterior distribution. An important aspect of the algorithm is the calibration of the proposal distribution. The tuning of the Markov chain is broken into a number of phases. In each phase, PROC MCMC generates trial samples and automatically modifies the proposal distribution as a result of the acceptance rate (see the section "Tuning the Proposal Distribution" on page 3525). In this example, PROC MCMC found an acceptable proposal distribution after 7 phases, and this distribution is used in both the burn-in and sampling stages of the simulation.

The "Burn-In History" table shows the burn-in phase, and the "Sampling History" table shows the main phase sampling.

Figure 52.3 Tuning, Burn-In and Sampling History

Tuning History

Phase	Block	Scale	Acceptance Rate
1	1	2.3800	0.0420
	2	2.3800	0.8860
2	1	1.0938	0.2180
	2	15.5148	0.3720
3	1	0.8299	0.4860
	2	15.5148	0.1260
4	1	1.1132	0.4840
	2	9.4767	0.0880
5	1	1.4866	0.5420
	2	5.1914	0.2000
6	1	2.2784	0.4600
	2	3.7859	0.3900
7	1	2.8820	0.3360
	2	3.7859	0.4020

Figure 52.3 *continued*

Burn-In History

Block	Scale	Acceptance Rate
1	2.8820	0.3400
2	3.7859	0.4150

Sampling History

Block	Scale	Acceptance Rate
1	2.8820	0.3284
2	3.7859	0.4008

For each posterior distribution, PROC MCMC also reports summary statistics (posterior means, standard deviations, and quantiles) and interval statistics (95% equal-tail and highest posterior density credible intervals), as shown in Figure 52.4. For more information about posterior statistics, see the section "Summary Statistics" on page 170.

Figure 52.4 MCMC Summary and Interval Statistics

Simple Linear Regression

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
beta0	10000	-142.6	33.9390	-164.5	-142.4	-120.5
beta1	10000	3.8917	0.5427	3.5406	3.8906	4.2402
sigma2	10000	136.8	51.7417	101.8	126.0	159.9

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
beta0	0.050	-209.3	-76.1692	-209.7	-77.1624
beta1	0.050	2.8317	4.9610	2.8280	4.9468
sigma2	0.050	69.2208	265.5	58.2627	233.8

By default, PROC MCMC also computes a number of convergence diagnostics to help you determine whether the chain has converged. These are the Monte Carlo standard errors, the autocorrelations at selected lags, the Geweke diagnostics, and the effective sample sizes. These statistics are shown in Figure 52.5. For details and interpretations of these diagnostics, see the section "Assessing Markov Chain Convergence" on page 156.

The "Monte Carlo Standard Errors" table indicates that the standard errors of the mean estimates for each of the parameters are relatively small, with respect to the posterior standard deviations. The values in the "MCSE/SD" column (ratios of the standard errors and the standard deviations) are small, around 0.01. This means that only a fraction of the posterior variability is due to the simulation. The "Autocorrelations of the Posterior Samples" table shows that the autocorrelations among posterior samples reduce quickly and become almost nonexistent after lag 5. The "Geweke Diagnostics" table indicates that no parameter failed the test, and the "Effective Sample Sizes" table reports the number of effective sample sizes of the Markov chain.

Figure 52.5 MCMC Convergence Diagnostics

Simple Linear Regression

The MCMC Procedure

Monte Carlo Standard Errors

Parameter	MCSE	Standard Deviation	MCSE/SD
beta0	0.4576	33.9390	0.0135
beta1	0.00731	0.5427	0.0135
sigma2	0.7151	51.7417	0.0138

Posterior Autocorrelations

Parameter	Lag 1	Lag 5	Lag 10	Lag 50
beta0	0.2986	-0.0008	0.0162	0.0193
beta1	0.2971	0.0000	0.0135	0.0161
sigma2	0.2966	0.0062	0.0008	-0.0068

Geweke Diagnostics

Parameter	z	Pr > |z|
beta0	0.1105	0.9120
beta1	-0.1701	0.8649
sigma2	-0.2175	0.8278

Effective Sample Sizes

Parameter	ESS	Correlation Time	Efficiency
beta0	5501.1	1.8178	0.5501
beta1	5514.8	1.8133	0.5515
sigma2	5235.4	1.9101	0.5235

PROC MCMC produces a number of graphs, shown in Figure 52.6, which also aid convergence diagnostic checks. With the trace plots, there are two important aspects to examine. First, you want to check whether the mean of the Markov chain has stabilized and appears constant over the graph. Second, you want to check whether the chain has good mixing and is "dense," in the sense

that it quickly traverses the support of the distribution to explore both the tails and the mode areas efficiently. The plots show that the chains appear to have reached their stationary distributions.

Next, you should examine the autocorrelation plots, which indicate the degree of autocorrelation for each of the posterior samples. High correlations usually imply slow mixing. Finally, the kernel density plots estimate the posterior marginal distributions for each parameter.

Figure 52.6 Diagnostic Plots for β_0, β_1 and σ^2

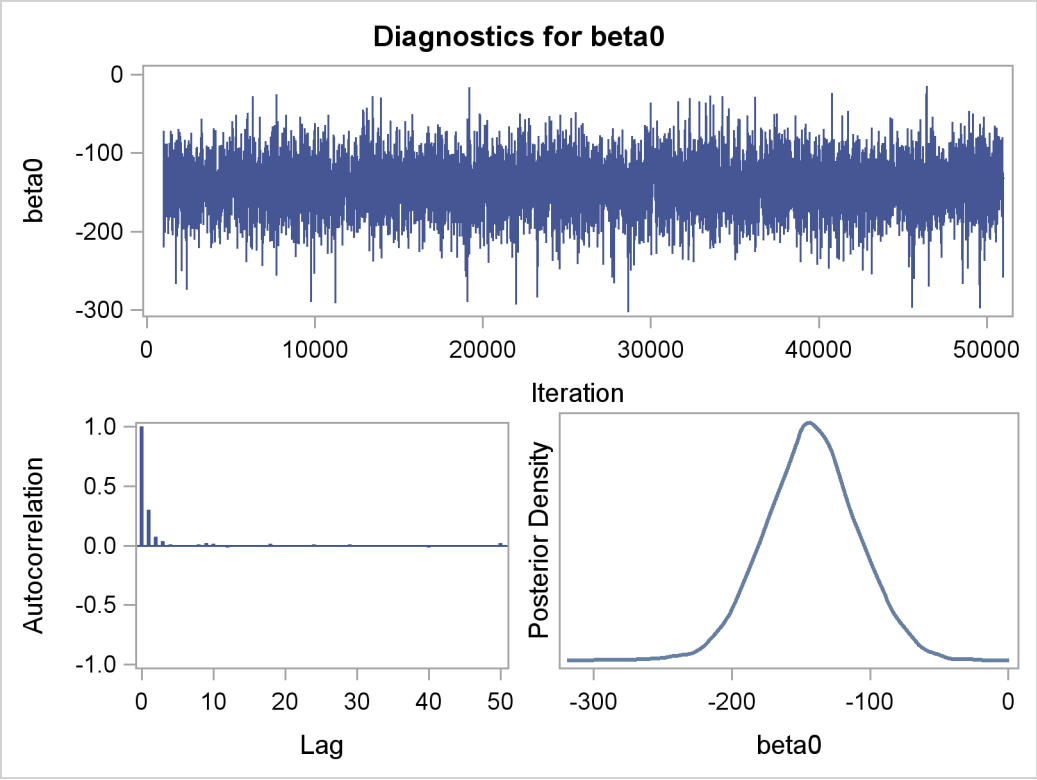
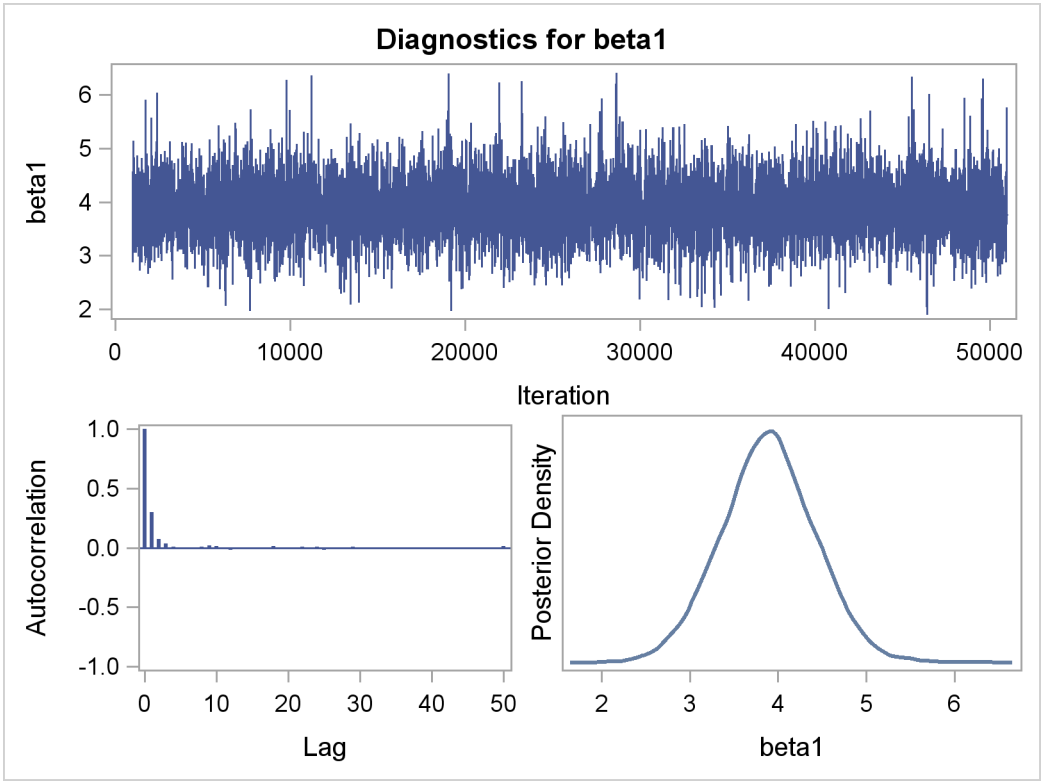
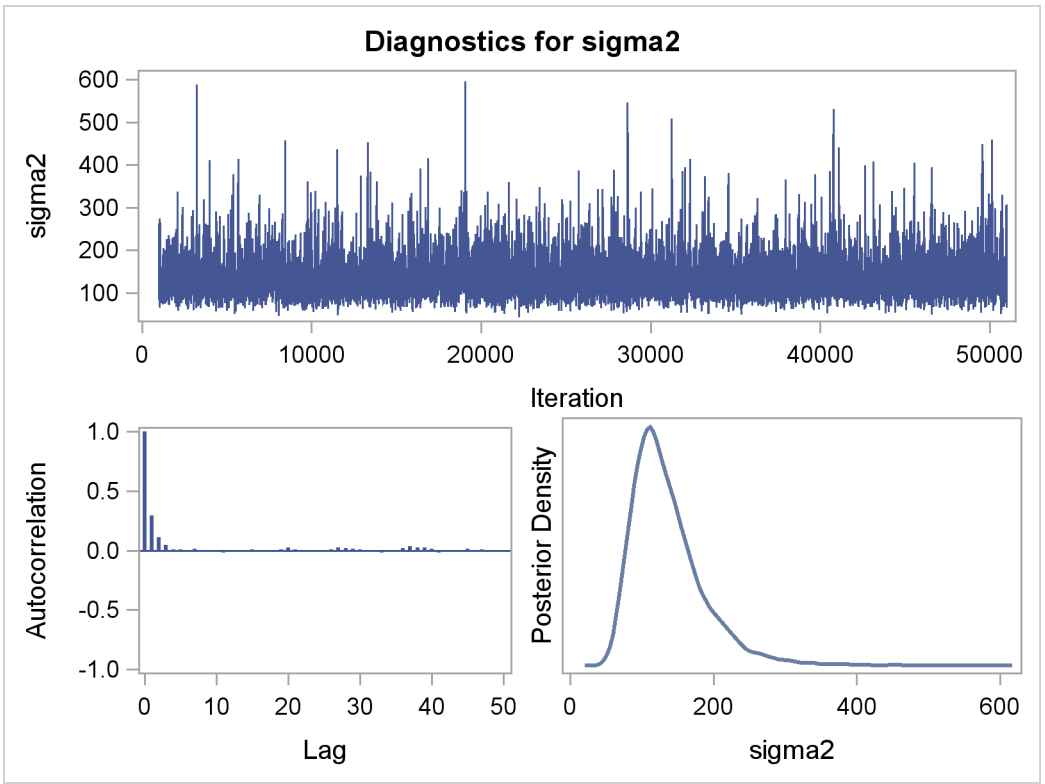


Figure 52.6 *continued*





In regression models such as this, you expect the posterior estimates to be very similar to the maximum likelihood estimators with noninformative priors on the parameters, The REG procedure produces the following fitted model (code not shown):

$$\text{Weight} = -143.0 + 3.9 \times \text{Height}$$

These are very similar to the means show in Figure 52.4. With PROC MCMC, you can carry out informative analysis that uses specifications to indicate prior knowledge on the parameters. Informative analysis is likely to produce different posterior estimates, which are the result of information from both the likelihood and the prior distributions. Incorporating additional information in the analysis is one major difference between the classical and Bayesian approaches to statistical inference.

The Behrens-Fisher Problem

One of the famous examples in the history of statistics is the Behrens-Fisher problem (Fisher 1935). Consider the situation where there are two independent samples from two different normal distributions:

$$y_{11}, y_{12}, \cdots, y_{1n_1} \sim \text{normal}(\mu_1, \sigma_1^2)$$
$$y_{21}, y_{22}, \cdots, y_{2n_2} \sim \text{normal}(\mu_2, \sigma_2^2)$$

Note that $n_1 \neq n_2$. When you do not want to assume that the variances are equal, testing the hypothesis $H_0 : \mu_1 = \mu 2$ is a difficult problem in the classical statistics framework, because the distribution under H_0 is not known. Within the Bayesian framework, this problem is straightforward because you can estimate the posterior distribution of $\mu_1 - \mu_2$ while taking into account the uncertainties in all of parameters by treating them as random variables.

Suppose that you have the following set of data:

```
title 'The Behrens-Fisher Problem';

data behrens;
   input y ind @@;
   datalines;
121 1  94 1 119 1 122 1 142 1 168 1 116 1
172 1 155 1 107 1 180 1 119 1 157 1 101 1
145 1 148 1 120 1 147 1 125 1 126 2 125 2
130 2 130 2 122 2 118 2 118 2 111 2 123 2
126 2 127 2 111 2 112 2 121 2
;
```

The response variable is `y`, and the `ind` variable is the group indicator, which takes two values: `1` and `2`. There are 19 observations that belong to group `1` and 14 that belong to group `2`.

The likelihood functions for the two samples are as follows:

$$\begin{aligned} p(y_{1i}|\mu_1,\sigma_1^2) &= \phi(y_{1i};\mu_1,\sigma_1^2) \quad \text{for} \quad i = 1,\cdots,19 \\ p(y_{2j}|\mu_2,\sigma_2^2) &= \phi(y_{2j};\mu_2,\sigma_2^2) \quad \text{for} \quad j = 1,\cdots,14 \end{aligned}$$

Berger (1985) showed that a uniform prior on the support of the location parameter is a noninformative prior. The distribution is invariant under location transformations—that is, $\theta = \mu + c$. You can use this prior for the mean parameters in the model:

$$\begin{aligned} \pi(\mu_1) &\propto 1 \\ \pi(\mu_2) &\propto 1 \end{aligned}$$

In addition, Berger (1985) showed that a prior of the form $1/\sigma^2$ is noninformative for the scale parameter, and it is invariant under scale transformations (that is $\tau = c\sigma^2$). You can use this prior for the variance parameters in the model:

$$\begin{aligned} \pi(\sigma_1^2) &\propto 1/\sigma_1^2 \\ \pi(\sigma_2^2) &\propto 1/\sigma_2^2 \end{aligned}$$

The log densities of the prior distributions on σ_1^2 and σ_2^2 are:

$$\begin{aligned} \log(\pi(\sigma_1^2)) &= -\log(\sigma_1^2) \\ \log(\pi(\sigma_2^2)) &= -\log(\sigma_2^2) \end{aligned}$$

The following statements generate posterior samples of $\mu_1, \mu_2, \sigma_1^2, \sigma_2^2$, and the difference in the means: $\mu_1 - \mu_2$:

```
proc mcmc data=behrens outpost=postout seed=123
          nmc=40000 thin=10 monitor=(_parms_ mudif)
          statistics(alpha=0.01)=(summary interval);
   ods select PostSummaries PostIntervals;
   parm mu1 0 mu2 0;
   parm sig21 1;
   parm sig22 1;
   prior mu: ~ general(0);
   prior sig21 ~ general(-log(sig21));
   prior sig22 ~ general(-log(sig22));
   mudif = mu1 - mu2;
   if ind = 1 then
      llike = lpdfnorm(y, mu1, sqrt(sig21));
   else
      llike = lpdfnorm(y, mu2, sqrt(sig22));
   model general(llike);
run;
```

The PROC MCMC statement specifies an input data set (behrens), an output data set containing the posterior samples (postout), a random number seed, the simulation size, and the thinning rate. The MONITOR= option specifies a list of symbols, which can be either parameters or functions of

the parameters in the model, for which inference is to be done. The symbol _parms_ is a shorthand for all model parameters—in this case, mu1, mu2, sig21, and sig22. The symbol mudif is defined in the program as the difference between μ_1 and μ_2.

The ODS SELECT statement displays the summary statistics and interval statistics tables while excluding all other output. For a complete list of ODS tables that PROC MCMC can produce, see the sections "Displayed Output" on page 3571 and "ODS Table Names" on page 3575.

The STATISTICS= option calculates summary and interval statistics. The global suboption ALPHA=0.01 specifies 99% equal-tail and highest posterior density (HPD) credible intervals for all parameters.

The PARMS statements assign the parameters mu1 and mu2 to the same block, and sig21 and sig22 each to their own separate blocks. There are a total of three blocks. The PARMS statements also assign an initial value to each parameter.

The PRIOR statements specify prior distributions for the parameters. Because the priors are all nonstandard (uniform on the real axis for μ_1 and μ_2 and $1/\sigma^2$ for σ_1^2 and σ_2^2), you must use the GENERAL function here. The argument in the GENERAL function is an expression for the log of the distribution, up to an additive constant. This distribution can have any functional form, as long as it is programmable using SAS functions and expressions. Note that the function specifies a distribution on the log scale, not the original scale. The log of the prior on mu1 and mu2 is 0, and the log of the priors on sig21 and sig22 are -log(sig21) and -log(sig22) respectively. See the section "Specifying a New Distribution" on page 3541 for more information about how to specify an arbitrary distribution.

The mudif assignment statement calculates the difference between mu1 and mu2. The IF-ELSE statements enable different y's to have different log-likelihood functions, depending on their group indicator ind. The function LPDFNORM is a PROC MCMC function that calculates the log density of a normal distribution. See the section "Using Density Functions in the Programming Statements" on page 3542 for more details. The MODEL statement specifies that llike is the log likelihood for each observation in the model.

Figure 52.7 displays the posterior summary and interval statistics.

Figure 52.7 Posterior Summary and Interval Statistics

The Behrens-Fisher Problem

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
mu1	4000	134.8	6.0065	130.9	134.7	138.7
mu2	4000	121.4	1.9150	120.2	121.4	122.7
sig21	4000	683.2	259.9	507.8	630.1	792.3
sig22	4000	51.3975	24.2881	35.0212	45.7449	61.2582
mudif	4000	13.3596	6.3335	9.1732	13.4078	17.6332

Figure 52.7 *continued*

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
mu1	0.010	118.7	150.6	119.3	151.0
mu2	0.010	115.9	126.6	116.2	126.7
sig21	0.010	292.0	1821.1	272.8	1643.7
sig22	0.010	18.5883	158.8	16.3730	140.5
mudif	0.010	-3.2537	29.9987	-3.1915	30.0558

The mean difference has a posterior mean value of 13.36, and the lower endpoints of the 99% credible intervals are negative. This suggests that the mean difference is positive with a high probability. However, if you want to estimate the probability that $\mu_1 - \mu_2 > 0$, you can do so as follows.

The following statements produce Figure 52.8:

```
proc format;
   value diffmt low-0 = 'mu1 - mu2 <= 0' 0<-high = 'mu1 - mu2 > 0';
run;

proc freq data = postout;
   tables mudif /nocum;
   format mudif diffmt.;
run;
```

The sample estimate of the posterior probability that $\mu_1 - \mu_2 > 0$ is 0.98. This example illustrates an advantage of Bayesian analysis. You are not limited to making inferences based on model parameters only. You can accurately quantify uncertainties with respect to any function of the parameters, and this allows for flexibility and easy interpretations in answering many scientific questions.

Figure 52.8 Estimated Probability of $\mu_1 - \mu_2 > 0$.

The Behrens-Fisher Problem

The FREQ Procedure

mudif	Frequency	Percent
mu1 - mu2 <= 0	77	1.93
mu1 - mu2 > 0	3923	98.08

Mixed-Effects Model

This example illustrates how you can fit a mixed-effects model in PROC MCMC. PROC MCMC offers you the ability to model beyond the normal likelihood (see "Example 52.5: Random-Effects Models" on page 3614), and you can model as many levels of random effects as are needed with this procedure.

Consider a scenario in which data are collected in groups and you wish to model group-specific effects. You can use a mixed-effects model (sometimes also known as a random-effects model or a variance-components model):

$$y_{ij} = \beta_0 + \beta_1 x_{ij} + \gamma_i + e_{ij}, \quad e_{ij} \sim \text{normal}(0, \sigma^2)$$

where $i = 1, 2, \cdots, I$ is the group index and $j = 1, 2, \cdots, n_i$ indexes the observations in the ith group. In the regression model, the fixed effects β_0 and β_1 are the intercept and the coefficient for variable x_{ij}, respectively. The random effects γ_i is the mean for the ith group, and e_{ij} are the error term.

Consider the following SAS data set:

```
title 'Mixed-Effects Model';

data heights;
   input Family G$ Height @@;
   datalines;
1 F 67   1 F 66   1 F 64   1 M 71   1 M 72   2 F 63
2 F 63   2 F 67   2 M 69   2 M 68   2 M 70   3 F 63
3 M 64   4 F 67   4 F 66   4 M 67   4 M 67   4 M 69
;

data input;
   set heights;
   if g eq 'F' then gender = 1;
   else gender = 0;
   drop g;
run;
```

The response variable Height measures the heights (in inches) of 18 individuals. The individuals are classified according to Family and Gender.

Height is assumed to be normally distributed:

$$y_{ij} \sim \text{normal}(\mu_{ij}, \sigma^2), \quad \mu_{ij} = \beta_0 + \beta_1 x_{ij} + \gamma_i$$

which corresponds to a normal likelihood as follows:

$$p(y_{ij} | \mu_{ij}, \sigma^2) = \phi(\mu_{ij}, \text{var} = \sigma^2)$$

The priors on the parameters β_0, β_1, γ_i are assumed to be normal as well:

$$\begin{aligned} \pi(\beta_0) &= \phi(0, \text{var} = 1e5) \\ \pi(\beta_1) &= \phi(0, \text{var} = 1e5) \\ \pi(\gamma_i) &= \phi(0, \text{var} = \sigma_\gamma^2) \end{aligned}$$

Priors on the variance terms, σ^2 and σ_γ^2, are inverse-gamma:

$$\begin{aligned}\pi(\sigma^2) &= f_{i\Gamma}(\text{shape} = 0.001, \text{scale} = 1000)\\ \pi(\sigma_\gamma^2) &= f_{i\Gamma}(\text{shape} = 0.001, \text{scale} = 1000)\end{aligned}$$

where $f_{i\Gamma}$ denotes the density function of an inverse-gamma distribution.

The following statements fit a linear random-effects model to the data and produce the output shown in Figure 52.9 and Figure 52.10:

```
ods graphics on;
proc mcmc data=input outpost=postout thin=10 nmc=50000 seed=7893
          monitor=(b0 b1);
   ods select PostSummaries PostIntervals tadpanel;
   array gamma[4];

   parms b0 0 b1 0 gamma: 0;
   parms s2 1 ;
   parms s2g 1;

   prior b: ~ normal(0, var = 10000);
   prior gamma: ~ normal(0, var = s2g);
   prior s2: ~ igamma(0.001, scale = 1000);

   mu = b0 + b1 * gender + gamma[family];
   model height ~ normal(mu, var = s2);
run;
ods graphics off;
```

The statements are very similar to those shown in the previous two examples. The ODS GRAPHICS ON statement requests ODS Graphics. The PROC MCMC statement specifies the input and output data sets, the simulation size, the thinning rate, and a random number seed. The MONITOR= option indicates that the model parameters b0 and b1 are the quantities of interest. The ODS SELECT statement displays the summary statistics table, the interval statistics table, and the diagnostics plots.

The ARRAY statement defines a one-dimensional array, gamma, with 4 elements. You can refer to the array elements with variable names (gamma1 to gamma4 by default) or with subscripts, such as gamma[2]. To indicate subscripts, you must use either brackets [ ] or braces { }, but not parentheses ( ). Note that this is different from the way subscripts are indicated in the DATA step. See the section "ARRAY Statement" on page 3508 for more information.

The PRIOR statements specify priors for all the parameters. The notation b: is a shorthand for all symbols that start with the letter 'b'. In this example, it includes b0 and b1. Similarly, gamma: stands for all four gamma parameters, and s2: stands for both s2 and s2g. This shorthand notation can save you some typing, and it keeps your statements tidy.

The mu assignment statement calculates the expected value of height in the random-effects model. The symbol family is a data set variable that indexes family. Here gamma[family] is the random effect for the value of family.

Finally, the MODEL statement specifies the likelihood function for height.

The posterior summary and interval statistics for b0 and b1 are shown in Figure 52.9.

Figure 52.9 Posterior Summary and Interval Statistics

Mixed-Effects Model

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
b0	5000	66.2685	19.1176	56.0024	66.7260	77.2356
b1	5000	−3.3492	6.3886	−7.4268	−3.2799	0.6078

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
b0	0.050	26.2226	103.3	27.1749	103.6
b1	0.050	−16.2018	9.6267	−17.0757	8.5265

Trace plots, autocorrelation plots, and posterior density plots for b1 and logpost are shown in Figure 52.10. The mixing of b1 looks good. The convergence plots for the other parameters also look reasonable, and are not shown here.

Figure 52.10 Plots for b_1 and Log of the Posterior Density

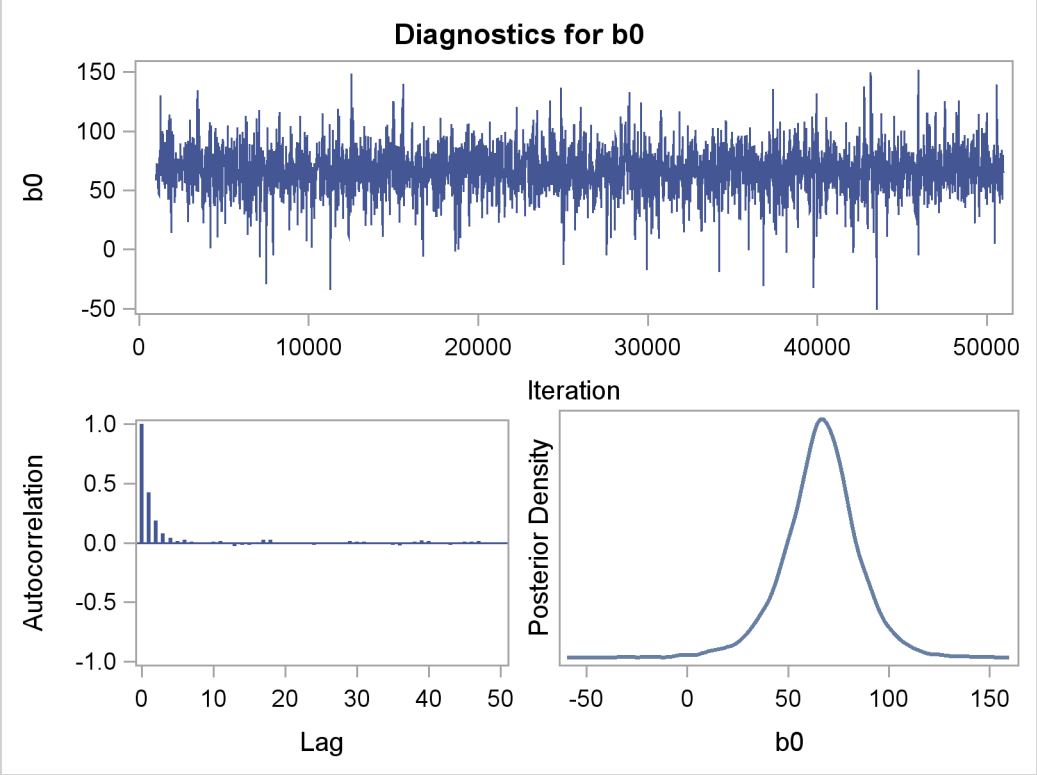
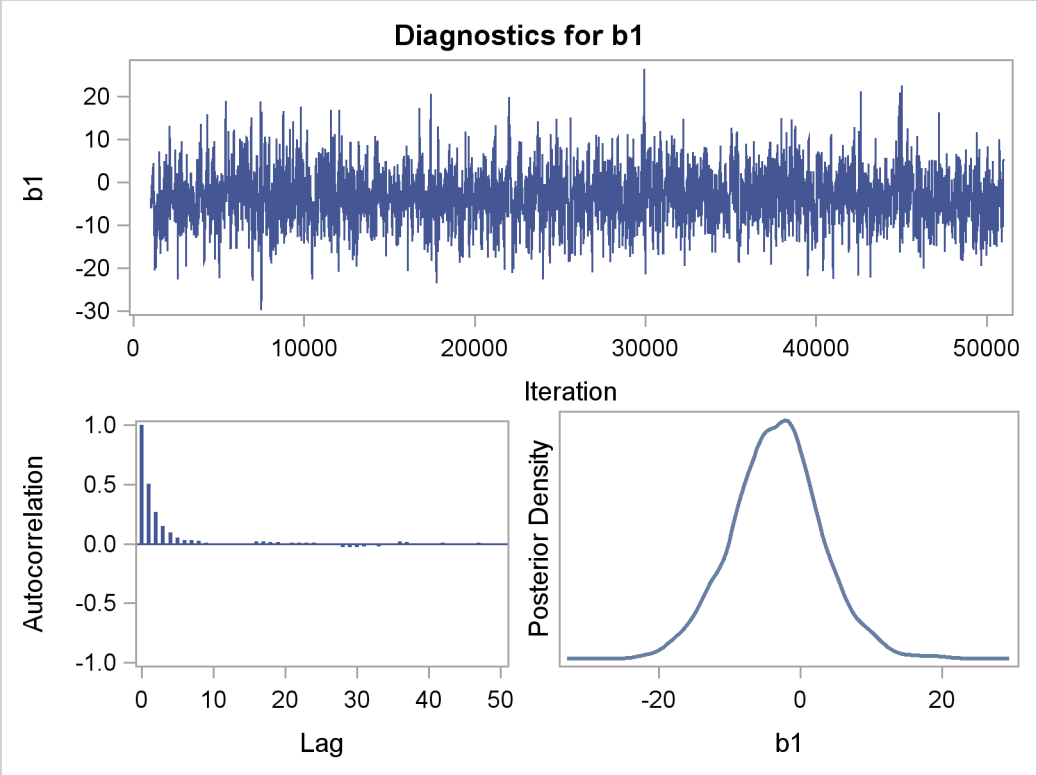


Figure 52.10 *continued*



From the interval statistics table, you see that both the equal-tail and HPD intervals for β_0 are positive, strongly indicating the positive effect of the parameter. On the other hand, both intervals for β_1 cover the value zero, indicating that gender does not have a strong impact on predicting height in this model.

Syntax: MCMC Procedure

The following statements can be used with PROC MCMC:

PROC MCMC *options* ;
  ARRAY *array specification* ;
  BEGINCNST/ENDCNST ;
  BEGINNODATA/ENDNODATA ;
  BY *variables* ;
  MODEL *statistical model specification* ;
  PARMS *parameters and starting values* ;
  PRIOR/HYPERPRIOR *prior or hyperprior specification* ;
  Program statements ;
  UDS *user defined sampler specification* ;

The PARMS statements declare parameters in the model and assign optional starting values for the Markov chain. The PRIOR/HYPERPRIOR statements specify the prior distributions of the parameters. The MODEL statements specify the log-likelihood functions for the response variables. These statements form the basis of every Bayesian model.

In addition, you can use the ARRAY statement to define constant or parameter arrays, the BEGINCNST/ENDCNST and similar statements to save unnecessary evaluation and reduce simulation time, the program statements to specify more complicated models that you wish to fit, and finally the UDS statements to define your own Gibbs samplers to sample any parameters in the model.

The following sections provide a description of each of these statements.

PROC MCMC Statement

PROC MCMC *options* **;**

This statement invokes PROC MCMC.

A number of options are available in the PROC MCMC statement; the following table categorizes them according to function.

Table 52.1 PROC MCMC Statement Options

Option	Description
Basic options	
DATA=	names the input data set
OUTPOST=	names the output data set for posterior samples of parameters
Debugging output	
LIST	displays model program and variables
LISTCODE	displays compiled model program
TRACE	displays detailed model execution messages
Frequently used MCMC options	
MAXTUNE=	specifies the maximum number of tuning loops
MINTUNE=	specifies the minimum number of tuning loops
NBI=	specifies the number of burn-in iterations
NMC=	specifies the number of MCMC iterations, excluding the burn-in iterations
NTU=	specifies the number of tuning iterations
PROPCOV=	controls options for constructing the initial proposal covariance matrix
SEED=	specifies the random seed for simulation
THIN=	specifies the thinning rate
Less frequently used MCMC options	
ACCEPTTOL=	specifies the acceptance rate tolerance
DISCRETE=	controls sampling discrete parameters
INIT=	controls generating initial values

Table 52.1 (continued)

Option	Description
PROPDIST=	specifies the proposal distribution
SCALE=	specifies the initial scale applied to the proposal distribution
TARGACCEPT=	specifies the target acceptance rate for random walk sampler
TARGACCEPTI=	specifies the target acceptance rate for independence sampler
TUNEWT=	specifies the weight used in covariance updating
Summary, diagnostics, and plotting options	
AUTOCORLAG=	specifies the number of autocorrelation lags used to compute effective sample sizes and Monte Carlo errors
DIAGNOSTICS=	controls the convergence diagnostics
DIC	computes deviance information criterion (DIC)
MONITOR=	outputs analysis for a list of symbols of interest
PLOTS=	controls plotting
STATISTICS=	controls posterior statistics
Other Options	
INF=	specifies the machine numerical limit for infinity
JOINTMODEL	specifies joint log-likelihood function
MISSING=	indicates how missing values are handled.
SIMREPORT=	controls the frequency of report for expected run time
SINGDEN=	specifies the singularity tolerance

These options are described in alphabetical order.

ACCEPTTOL=*n*

specifies a tolerance for acceptance probabilities. By default, ACCEPTTOL=0.075.

AUTOCORLAG=*n*

ACLAG=*n*

specifies the maximum number of autocorrelation lags used in computing the effective sample size; see the section "Effective Sample Size" on page 169 for more details. The value is used in the calculation of the Monte Carlo standard error; see the section "Standard Error of the Mean Estimate" on page 170. By default, AUTOCORLAG=MIN(500, MCsample/4), where MCsample is the Markov chain sample size kept after thinning—that is, MCsample $= \left[\frac{\text{NMC}}{\text{NTHIN}}\right]$. If AUTOCORLAG= is set too low, you might observe significant lags, and the effective sample size cannot be calculated accurately. A WARNING message appears, and you can either increase AUTOCORLAG= or NMC=, accordingly.

DISCRETE=*keyword*

specifies the proposal distribution used in sampling discrete parameters. The default is DISCRETE=BINNING.

The *keyword* values are as follows:

BINNING

uses continuous proposal distributions for all discrete parameter blocks. The proposed

sample is then discretized (binned) before further calculations. This sampling method approximates the correlation structure among the discrete parameters in the block and could improve mixing in some cases.

GEO

uses independent symmetric geometric proposal distributions for all discrete parameter blocks. This proposal does not take parameter correlations into account. However, it can work better than the BINNING option in cases where the range of the parameters is relatively small and a normal approximation can perform poorly.

DIAGNOSTICS=NONE | *(keyword-list)*

DIAG=NONE | *(keyword-list)*

specifies options for MCMC convergence diagnostics. By default, PROC MCMC computes the Geweke test, sample autocorrelations, effective sample sizes, and Monte Carlo errors. The Raftery-Lewis and Heidelberger-Welch tests are also available. See the section "Assessing Markov Chain Convergence" on page 156 for more details on convergence diagnostics. You can request all of the diagnostic tests by specifying DIAGNOSTICS=ALL. You can suppress all the tests by specifying DIAGNOSTICS=NONE.

The following options are available.

ALL

computes all diagnostic tests and statistics. You can combine the option ALL with any other specific tests to modify test options. For example DIAGNOSTICS=(ALL AUTOCORR(LAGS=(1 5 35))) computes all tests with default settings and autocorrelations at lags 1, 5, and 35.

AUTOCORR < *(autocorr-options)* **>**

computes default autocorrelations at lags 1, 5, 10, and 50 for each variable. You can choose other lags by using the following *autocorr-options*:

LAGS | **AC=***numeric-list*

specifies autocorrelation lags. The *numeric-list* must take positive integer values.

ESS

computes the effective sample sizes (Kass et al. (1998)) of the posterior samples of each parameter. It also computes the correlation time and the efficiency of the chain for each parameter. Small values of ESS might indicate a lack of convergence. See the section "Effective Sample Size" on page 169 for more details.

GEWEKE < *(Geweke-options)* **>**

computes the Geweke spectral density diagnostics; this is a two-sample t-test between the first f_1 portion and the last f_2 portion of the chain. See the section "Geweke Diagnostics" on page 163 for more details. The default is FRAC1=0.1 and FRAC2=0.5, but you can choose other fractions by using the following *Geweke-options*:

FRAC1 | **F1=***value*

specifies the beginning FRAC1 proportion of the Markov chain. By default, FRAC1=0.1.

FRAC2 | F2=*value*

specifies the end FRAC2 proportion of the Markov chain. By default, FRAC2=0.5.

HEIDELBERGER | HEIDEL < *(Heidel-options)* **>**

computes the Heidelberger and Welch diagnostic (which consists of a stationarity test and a halfwidth test) for each variable. The stationary diagnostic test tests the null hypothesis that the posterior samples are generated from a stationary process. If the stationarity test is passed, a halfwidth test is then carried out. See the section "Heidelberger and Welch Diagnostics" on page 165 for more details.

These diagnostics are not performed by default. You can specify the DIAGNOSTICS=HEIDELBERGER option to request these diagnostics, and you can also specify suboptions, such as DIAGNOSTICS=HEIDELBERGER(EPS=0.05), as follows:

SALPHA=*value*

specifies the α level ($0 < \alpha < 1$) for the stationarity test. By default, SALPHA=0.05.

HALPHA=*value*

specifies the α level ($0 < \alpha < 1$) for the halfwidth test. By default, HALPHA=0.05.

EPS=*value*

specifies a small positive number ϵ such that if the halfwidth is less than ϵ times the sample mean of the retaining iterates, the halfwidth test is passed. By default, EPS=0.1.

MCSE

MCERROR

computes the Monte Carlo standard error for the posterior samples of each parameter.

NONE

suppresses all of the diagnostic tests and statistics. This is not recommended.

RAFTERY | RL <(*Raftery-options***)>**

computes the Raftery and Lewis diagnostics, which evaluate the accuracy of the estimated quantile ($\hat{\theta}_Q$ for a given $Q \in (0,1)$) of a chain. $\hat{\theta}_Q$ can achieve any degree of accuracy when the chain is allowed to run for a long time. The algorithm stops when the estimated probability $\hat{P}_Q = \Pr(\theta \leq \hat{\theta}_Q)$ reaches within $\pm R$ of the value Q with probability S; that is, $\Pr(Q - R \leq \hat{P}_Q \leq Q + R) = S$. See the section "Raftery and Lewis Diagnostics" on page 166 for more details. The *Raftery-options* enable you to specify Q, R, S, and a precision level ϵ for a stationary test.

These diagnostics are not performed by default. You can specify the DIAGNOSTICS=RAFERTY option to request these diagnostics, and you can also specify suboptions, such as DIAGNOSTICS=RAFERTY(QUANTILE=0.05), as follows:

QUANTILE | Q=*value*

specifies the order (a value between 0 and 1) of the quantile of interest. By default, QUANTILE=0.025.

ACCURACY | R=*value*

specifies a small positive number as the margin of error for measuring the accuracy of estimation of the quantile. By default, ACCURACY=0.005.

PROB | S=*value*

specifies the probability of attaining the accuracy of the estimation of the quantile. By default, PROB=0.95.

EPS=*value*

specifies the tolerance level (a small positive number) for the stationary test. By default, EPS=0.001.

DIC

computes the Deviance Information Criterion (DIC). DIC is calculated using the posterior mean estimates of the parameters. See the section "Deviance Information Criterion (DIC)" on page 172 for more details.

DATA=*SAS-data-set*

specifies the input data set. Observations in this data set are used to compute the log-likelihood function that you specify with PROC MCMC statements.

INF=*value*

specifies the numerical definition of infinity in the procedure. The default is INF= 1E15. For example, PROC MCMC considers 1E16 to be outside of the support of the normal distribution and assigns a missing value to the log density evaluation. You can select a larger value with the INF= option. The minimum value allowed is 1E10.

INIT=*(keyword-list)*

specifies options for generating the initial values for the parameters. These options apply only to prior distributions that are recognized by PROC MCMC. See the section "Standard Distributions" on page 3530 for a list of these distributions. If either of the functions GENERAL or DGENERAL is used, you must supply explicit initial values for the parameters. By default, INIT=MODE. The following keywords are used:

MODE

uses the mode of the prior density as the initial value of the parameter, if you did not provide one. If the mode does not exist or if it is on the boundary of the support of the density, the mean value is used. If the mean is outside of the support or on the boundary, which can happen if the prior distribution is truncated, a random number drawn from the prior is used as the initial value.

PINIT
tabulates parameter values after the tuning phase. This option also tabulates the tuned proposal parameters used by the Metropolis algorithm. These proposal parameters include covariance matrices for continuous parameters and probability vectors for discrete parameters for each block. By default, PROC MCMC does not display the initial values or the tuned proposal parameters after the tuning phase.

RANDOM
generates a random number from the prior density and uses it as the initial value of the parameter, if you did not provide one.

REINIT
resets the parameters, after the tuning phase, with the initial values that you provided explicitly or that were assigned by the procedure. By default, PROC MCMC does not reset the parameters because the tuning phase usually moves the Markov chains to a more favorable place in the posterior distribution.

LIST
displays the model program and variable lists. The LIST option is a debugging feature and is not normally needed.

LISTCODE
displays the compiled program code. The LISTCODE option is a debugging feature and is not normally needed.

JOINTMODEL
JOINTLLIKE
specifies how the likelihood function is calculated. By default, PROC MCMC assumes that the observations in the data set are independent so that the joint log-likelihood function is the sum of the individual log-likelihood functions for the observations, where the individual log-likelihood function is specified in the MODEL statement. When your data are not independent, you can specify the JOINTMODEL option to modify the way that PROC MCMC computes the joint log-likelihood function. In this situation, PROC MCMC no longer steps through the input data set to sum the individual log likelihood.

To use this option correctly, you need to do the following two things:

- create ARRAY symbols to store all data set variables that are used in the program. This can be accomplished with the BEGINCNST and ENDCNST statements.
- program the joint log-likelihood function by using these ARRAY symbols only. The MODEL statement specifies the joint log-likelihood function for the entire data set. Typically, you use the function GENERAL in the MODEL statement.

See the sections "BEGINCNST/ENDCNST Statement" on page 3509 and "Modeling Joint Likelihood" on page 3556 for details.

MAXTUNE=*n*
specifies an upper limit for the number of proposal tuning loops. By default, MAXTUNE=24. See the section "Covariance Tuning" on page 3526 for more details.

MINTUNE=*n*

specifies a lower limit for the number of proposal tuning loops. By default, MINTUNE=2. See the section "Covariance Tuning" on page 3526 for more details.

MISSING=*keyword*

MISS=*keyword*

specifies how missing values are handled (see the section "Handling of Missing Data" on page 3565 for more details). The default is MISSING=COMPLETECASE.

ALLCASE | AC

gives you the option to model the missing values in an all-case analysis. You can use any techniques that you see fit, for example, fully Bayesian or multiple imputation.

COMPLETECASE | CC

assumes a complete case analysis, so all observations with missing variable values are discarded prior to the simulation.

MONITOR= (*symbol-list***)**

outputs analysis for selected symbols of interest in the program. The symbols can be any of the following: model parameters (symbols in the PARMS statement), secondary parameters (assigned using the operator "="), the log of the posterior density (LOGPOST), the log of the prior density (LOGPRIOR), the log of the hyperprior density (LOGHYPER) if the HYPER statement is used, or the log of the likelihood function (LOGLIKE). You can use the keyword _PARMS_ as a shorthand for all of the model parameters. PROC MCMC performs only posterior analyses (such as plotting, diagnostics, and summaries) on the symbols selected with the MONITOR= option. You can also choose to monitor an entire array by specifying the name of the array. By default MONITOR=_PARMS_.

Posterior samples of any secondary parameters listed in the MONITOR= option are saved in the OUTPOST= data set. Posterior samples of model parameters are always saved to the OUTPOST= data set, regardless of whether they appear in the MONITOR= option.

NBI=*n*

specifies the number of burn-in iterations to perform before beginning to save parameter estimate chains. By default, NBI=1000. See the section "Burn-in, Thinning, and Markov Chain Samples" on page 155 for more details.

NMC=*n*

specifies the number of iterations in the main simulation loop. This is the MCMC sample size if THIN=1. By default, NMC=1000.

NTU=*n*

specifies the number of iterations to use in each proposal tuning phase. By default, NTU=500.

OUTPOST=*SAS-data-set*

specifies an output data set that contains the posterior samples of all model parameters, the iteration numbers (variable name ITERATION), the log of the posterior density (LOGPOST), the log of the prior density (LOGPRIOR), the log of the hyperprior density (LOGHYPER), if the HYPER statement is used, and the log likelihood (LOGLIKE). Any secondary parameters

(assigned using the operator "=") listed in the MONITOR= option are saved to this data set. By default, no OUTPOST= data set is created.

PLOTS< *(global-plot-options)* >**=** *(plot-request* < *. . . plot-request* >*)*

PLOT< *(global-plot-options)* >**=** *(plot-request* < *. . . plot-request* >*)*

controls the display of diagnostic plots. Three types of plots can be requested: trace plots, autocorrelation function plots, and kernel density plots. By default, the plots are displayed in panels unless the global plot option UNPACK is specified. Also when more than one type of plot is specified, the plots are grouped by parameter unless the global plot option GROUPBY=TYPE is specified. When you specify only one plot request, you can omit the parentheses around the plot-request, as shown in the following example:

```
plots=none
plots(unpack)=trace
plots=(trace density)
```

You must enable ODS Graphics before requesting plots—for example, like this:

```
ods graphics on;

proc mcmc;
   ...;
run;

ods graphics off;
```

If you have enabled ODS Graphics but do not specify the PLOTS= option, then PROC MCMC produces, for each parameter, a panel that contains the trace plot, the autocorrelation function plot, and the density plot. This is equivalent to specifying PLOTS=(TRACE AUTOCORR DENSITY).

The *global-plot-options* include the following:

FRINGE

adds a fringe plot to the horizontal axis of the density plot.

GROUPBY|GROUP=PARAMETER | TYPE

specifies how the plots are grouped when there is more than one type of plot. GROUPBY=PARAMETER is the default. The choices are as follows:

TYPE

specifies that the plots are grouped by type.

PARAMETER

specifies that the plots are grouped by parameter.

LAGS=*n*

specifies the number of autocorrelation lags used in plotting the ACF graph. By default, LAGS=50.

SMOOTH

smoothes the trace plot with a fitted penalized B-spline curve (Eilers and Marx 1996).

UNPACKPANEL

UNPACK

specifies that all paneled plots are to be unpacked, so that each plot in a panel is displayed separately.

The *plot-requests* are as follows:

ALL

requests all types of plots. PLOTS=ALL is equivalent to specifying PLOTS=(TRACE AUTOCORR DENSITY).

AUTOCORR | ACF

displays the autocorrelation function plots for the parameters.

DENSITY | D | KERNEL | K

displays the kernel density plots for the parameters.

NONE

suppresses the display of all plots.

TRACE | T

displays the trace plots for the parameters.

Consider a model with four parameters, X1–X4. Displays for various specifications are depicted as follows.

- PLOTS=(TRACE AUTOCORR) displays the trace and autocorrelation plots for each parameter side by side with two parameters per panel:

Display 1	Trace(X1)	Autocorr(X1)
	Trace(X2)	Autocorr(X2)
Display 2	Trace(X3)	Autocorr(X3)
	Trace(X4)	Autocorr(X4)

- PLOTS(GROUPBY=TYPE)=(TRACE AUTOCORR) displays all the paneled trace plots, followed by panels of autocorrelation plots:

Display 1	Trace(X1)	
	Trace(X2)	
Display 2	Trace(X3)	
	Trace(X4)	
Display 3	Autocorr(X1)	Autocorr(X2)
	Autocorr(X3)	Autocorr(X4)

- PLOTS(UNPACK)=(TRACE AUTOCORR) displays a separate trace plot and a separate correlation plot, parameter by parameter:

Display 1	Trace(X1)
Display 2	Autocorr(X1)
Display 3	Trace(X2)
Display 4	Autocorr(X2)
Display 5	Trace(X3)
Display 6	Autocorr(X3)
Display 7	Trace(X4)
Display 8	Autocorr(X4)

- PLOTS(UNPACK GROUPBY=TYPE)=(TRACE AUTOCORR) displays all the separate trace plots followed by the separate autocorrelation plots:

Display 1	Trace(X1)
Display 2	Trace(X2)
Display 3	Trace(X3)
Display 4	Trace(X4)
Display 5	Autocorr(X1)
Display 6	Autocorr(X2)
Display 7	Autocorr(X3)
Display 8	Autocorr(X4)

PROPCOV=*value*

specifies the method used in constructing the initial covariance matrix for the Metropolis-Hastings algorithm. The QUANEW and NMSIMP methods find numerically approximated covariance matrices at the optimum of the posterior density function with respect to all continuous parameters. The optimization does not apply to discrete parameters. The tuning phase starts at the optimized values; in some problems, this can greatly increase convergence performance. If the approximated covariance matrix is not positive definite, then an identity matrix is used instead. Valid values are as follows:

IND

uses the identity covariance matrix. This is the default. See the section "Tuning the Proposal Distribution" on page 3525.

CONGRA< (*optimize-options*) >
performs a conjugate-gradient optimization.

DBLDOG< (*optimize-options*) >
performs a double-dogleg optimization.

QUANEW< (*optimize-options*) >
performs a quasi-Newton optimization.

NMSIMP | SIMPLEX< (*optimize-options*) >
performs a Nelder-Mead simplex optimization.

The *optimize-options* are as follows:

ITPRINT

prints optimization iteration steps and results.

PROPDIST=*value*
specifies a proposal distribution for the Metropolis algorithm. See the section "Metropolis and Metropolis-Hastings Algorithms" on page 152. You can also use PARMS statement option (see the section "PARMS Statement" on page 3515) to change the proposal distribution for a particular block of parameters. Valid values are as follows:

NORMAL
N
specifies a normal distribution as the proposal distribution. This is the default.

T< (*df*) >
specifies a *t*-distribution with the degrees of freedom *df*. By default, *df*=3. If $df > 100$, the normal distribution is used since the two distributions are almost identical.

SCALE=*value*
controls the initial multiplicative scale to the covariance matrix of the proposal distribution. By default, SCALE=2.38. See the section "Scale Tuning" on page 3526 for more details.

SEED=*n*
specifies the random number seed. By default, SEED=0, and PROC MCMC gets a random number seed from the clock.

SIMREPORT=*n*
controls the number of times that PROC MCMC reports the expected run time of the simulation. This can be useful for monitoring the progress of CPU-intensive programs. For example, with SIMREPORT=2, PROC MCMC reports the simulation progress twice. By default, SIMREPORT=0, and there is no reporting. The expected run times are displayed in the log file.

SINGDEN=*value*
defines the singularity criterion in the procedure. By default, SINGDEN=1E-11. The *value* indicates the exclusion of an endpoint in an interval. The mathematical notation "(0" is

equivalent to "[*value*" in PROC MCMC—that is, $x < 0$ is treated as $x \leq value$ in the procedure. The maximum SINGDEN allowed is $1E-6$.

STATISTICS< *(global-stats-options)* > **= NONE | ALL |***stats-request*

STATS< *(global-stats-options)* > **= NONE | ALL |***stats-request*

specifies options for posterior statistics. By default, PROC MCMC computes the posterior mean, standard deviation, quantiles, and two 95% credible intervals: equal-tail and highest posterior density (HPD). Other available statistics include the posterior correlation and covariance. See the section "Summary Statistics" on page 170 for more details. You can request all of the posterior statistics by specifying STATS=ALL. You can suppress all the calculations by specifying STATS=NONE.

The *global-stats-options* includes the following:

ALPHA=*numeric-list*

specifies the α level for the equal-tail and HPD intervals. The value α must be between 0 and 0.5. By default, ALPHA=0.05.

PERCENTAGE | PERCENT=*numeric-list*

calculates the posterior percentages. The *numeric-list* contains values between 0 and 100. By default, PERCENTAGE=(25 50 75).

The *stats-requests* include the following:

ALL

computes all posterior statistics. You can combine the option ALL with any other options. For example STATS(ALPHA=(0.02 0.05 0.1))=ALL computes all statistics with the default settings and intervals at α levels of 0.02, 0.05, and 0.1.

CORR

computes the posterior correlation matrix.

COV

computes the posterior covariance matrix.

SUMMARY

SUM

computes the posterior means, standard deviations, and percentile points for each variable. By default, the 25th, 50th, and 75th percentile points are produced, but you can use the global PERCENT= option to request specific percentile points.

INTERVAL

INT

computes the $100(1-\alpha)\%$ equal-tail and HPD credible intervals for each variable. See the sections "Equal-Tail Credible Interval" on page 171 and "Highest Posterior Density (HPD) Interval" on page 171 for details. By default, ALPHA=0.05, but you can use the global ALPHA= option to request other intervals of any probabilities.

NONE

suppresses all of the statistics.

TARGACCEPT=*value*

specifies the target acceptance rate for the random walk based Metropolis algorithm. See the section "Metropolis and Metropolis-Hastings Algorithms" on page 152. The numeric *value* must be between 0.01 and 0.99. By default, TARGACCEPT=0.45 for models with 1 parameter; TARGACCEPT=0.35 for models with 2, 3, or 4 parameters; and TARGACCEPT=0.234 for models with more than 4 parameters (Roberts, Gelman, and Gilks 1997; Roberts and Rosenthal 2001).

TARGACCEPTI=*value*

specifies the target acceptance rate for the independence sampler algorithm. The independence sampler is used for blocks of binary parameters. See the section "Independence Sampler" on page 153 for more details. The numeric *value* must be between 0 and 1. By default, TARGACCEPTI=0.6.

THIN=*n*

NTHIN=*n*

controls the thinning rate of the simulation. PROC MCMC keeps every *n*th simulation sample and discards the rest. All of the posterior statistics and diagnostics are calculated using the thinned samples. By default, THIN=1. See the section "Burn-in, Thinning, and Markov Chain Samples" on page 155 for more details.

TRACE

displays the result of each operation in each statement in the model program as it is executed. This debugging option is very rarely needed, and it produces voluminous output. If you use this option, also use small NMC=, NBI=, MAXTUNE=, and NTU= numbers.

TUNEWT=*value*

specifies the multiplicative weight used in updating the covariance matrix of the proposal distribution. The numeric *value* must be between 0 and 1. By default, TUNEWT=0.75. See the section "Covariance Tuning" on page 3526 for more details.

ARRAY Statement

ARRAY *arrayname <{ dimensions }> <$> <variables and constants>* **;**

The ARRAY statement associates a name (of no more than eight characters) with a list of variables and constants. The ARRAY statement is similar to, but not the same as, the ARRAY statement in the DATA step, and it is the same as the ARRAY statements in the NLIN, NLP, NLMIXED, and MODEL procedures. The array name is used with subscripts in the program to refer to the array elements, as illustrated in the following statements:

```
array r[8] r1-r8;

do i = 1 to 8;
   r[i] = 0;
end;
```

The ARRAY statement does not support all the features of the ARRAY statement in the DATA step. Implicit indexing of variables cannot be used; all array references must have explicit subscript expressions. Only exact array dimensions are allowed; lower-bound specifications are not supported. A maximum of six dimensions is allowed.

Both variables and constants can be array elements. Constant array elements cannot have values assigned to them while variables can. Both the dimension specification and the list of elements are optional, but at least one must be specified. When the list of elements is not specified or fewer elements than the size of the array are listed, array variables are created by appending element numbers to the array name to complete the element list. You can index array elements by enclosing a subscript in braces ({ }) or brackets ([ ]), but not in parentheses (( )). The parentheses are reserved for function calls only.

For example, the following statement names an array day:

```
array day[365];
```

By default, the variables names are day1 to day365. However, since **day** is a SAS function, any subscript that uses parentheses gives you the wrong results. The expression **day(4)** returns the value 5 and does not reference the array element day4.

BEGINCNST/ENDCNST Statement

BEGINCNST ;

ENDCNST ;

The BEGINCNST and ENDCNST statements define a block within which PROC MCMC processes the programming statements only during the setup stage of the simulation. You can use the BEGINCNST and ENDCNST statements to define constants or import data set variables into arrays. Storing data in arrays enables you to work with data that are not identically distributed (see the section "Modeling Joint Likelihood" on page 3556) or to implement your own Markov chain sampler (see the section "UDS Statement" on page 3518). You can also use the BEGINCNST and ENDCNST statements to assign initial values to the parameters (see the section "Assignments of Parameters" on page 3528).

Assign Constants

Whenever you have programming statements that calculate constants that do not need to be evaluated multiple times throughout the simulation, you should put them within the BEGINCNST and ENDCNST statements. Using these statements can reduce redundant processing. For example, you can assign a constant to a symbol or fill in an array with numbers:

```
array cnst[17];
begincnst;
   offset = 17;
```

```
   do i = 1 to 17;
      cnst[i] = i * i;
   end;
endcnst;
```

The MCMC procedure evaluates the programming statements with the BEGINCNST/ENDCNST block once and ignores them in the rest of the simulation.

READ_ARRAY Function

Sometimes you might need to store variables, either from the current input data set or from a different data set, in arrays and use these arrays to specify your model. The READ_ARRAY function is a convenient for that purpose.

The following two forms of the READ_ARRAY function are available:

rc = READ_ARRAY *(data_set, array)* **;**

rc = READ_ARRAY *(data_set, array <,"col_name_1"> <, "col_name_2"> <, ...>)* **;**

where

- *rc* returns 0 if the function is able to successfully read the data set.
- *data_set* specifies the name of the data set from which the array data is read. The value specified for *data_set* must be a character literal or a variable that contains the member name (libname.memname) of the data set to be read from.
- *array* specifies the PROC MCMC array variable into which the data is read. The value specified for *array* must be a local temporary array variable because the function might need to grow or shrink its size to accommodate the size of the data set.
- *col_name* specifies optional names for the specific columns of the data set that are read. If specified, *col_name* must be a literal string enclosed in quotation marks. In addition, *col_name* cannot be a PROC MCMC variable. If column names are not specified, PROC MCMC reads all of the columns in the data set.

When SAS translates between an array and a data set, the array is indexed as [row,column].

The READ_ARRAY function attempts to dynamically resize the array to match the dimensions of the input data set. Therefore, the array must be dynamic; that is, the array must be declared with the /NOSYMBOLS option.

For examples that use the READ_ARRAY function, see "Modeling Joint Likelihood" on page 3556, "Time Independent Model" on page 3649, and "Example 52.11: Implement a New Sampling Algorithm" on page 3672.

BEGINNODATA/ENDNODATA Statements

BEGINNODATA ;

ENDNODATA ;

BEGINPRIOR ;

ENDPRIOR ;

The BEGINNODATA and ENDNODATA statements define a block within which PROC MCMC processes the programming statements without stepping through the entire data set. The programming statements are executed only twice: at the first and the last observation of the data set. The BEGINNODATA and ENDNODATA statements are best used to reduce unnecessary observation-level computations. Any computations that are identical to every observation, such as transformation of parameters, should be enclosed in these statements.

The BEGINPRIOR and ENDPRIOR statements are aliases for the BEGINNODATA and ENDNODATA statements, respectively. You can enclose PRIOR statements in the BEGINNODATA and ENDNODATA statements.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MCMC to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC MCMC. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

MODEL Statement

MODEL *dependent-variable-list* ~ *distribution* **;**

The MODEL statement is used to specify the conditional distribution of the data given the parameters (the likelihood function). You must specify a single dependent variable or a list of dependent variables, a tilde (~), and then a distribution with its arguments. The dependent variables can be variables from the input data set or functions of the symbols in the program. The dependent variables must be specified unless the functions GENERAL or DGENERAL are used (see the section "Specifying a New Distribution" on page 3541 for more details). Multiple MODEL statements are allowed for defining models with multiple independent components. The log likelihood value is the sum of the log likelihood values from each MODEL statement.

PROC MCMC is a programming language that is similar to the DATA step, and the order of statement evaluation is important. For example, the MODEL statement must come after any SAS programming statements that define or modify arguments used in the construction of the log likelihood. In PROC MCMC, a symbol is allowed to be defined multiple times and used at different places. Using an expression out of order produces erroneous results that can also be hard to detect.

Standard distributions that the MODEL statement supports are listed in the Table 52.2 (see the section "Standard Distributions" on page 3530 for density specification). These distributions can also be used in the PRIOR and HYPERPRIOR statements. PROC MCMC allows some distributions to be parameterized in multiple ways. For example, you can specify a normal distribution with variance (VAR=), standard deviation (SD=), or precision (PRECISION=) parameter. For distributions that have different parameterizations, you must specify an option to clearly name the ambiguous parameter. In the normal distribution, for example, you must indicate whether the second argument is a variance, a standard deviation, or a precision.

All distributions, with the exception of binary and uniform, can have the optional arguments of LOWER= and UPPER=, which specify a truncated density. See the section "Truncation and Censoring" on page 3544 for more details.

Table 52.2 Valid Distributions

Distribution Name	Definition
beta(<**a=**>α, <**b=**>β)	beta distribution with shape parameters α and β
binary(<**prob**|**p=**> p)	binary (Bernoulli) distribution with probability of success p. You can use the alias **bern** for this distribution.
binomial (<**n=**> n, <**prob**|**p=**> p)	binomial distribution with count n and probability of success p
cauchy (<**location**|**loc**|**l=**>θ, <**scale**|**s=**>λ)	Cauchy distribution with location θ and scale λ
chisq(<**df=**> ν)	χ^2 distribution with ν degrees of freedom

Table 52.2 (continued)

Distribution Name	Definition
dgeneral(*ll*)	general log-likelihood function that you construct using SAS programming statements for single or multiple *discrete* variables. Also see the function **general**. The name **dlogden** is an alias for this function.
expchisq(<**df=**> ν)	log transformation of a χ^2 distribution with ν degrees of freedom: $\theta \sim$ **chisq**(ν) $\Leftrightarrow \log(\theta) \sim$ **expchisq**(ν). You can use the alias **echisq** for this distribution.
expexpon(**scale**|**s=** λ) **expexpon**(**iscale**|**is=** λ)	log transformation of an exponential distribution with scale or inverse-scale parameter λ: $\theta \sim$ **expon**(λ) $\Leftrightarrow \log(\theta) \sim$ **expexpon**(λ). You can use the alias **eexpon** for this distribution.
expGamma(<**shape**|**sp=**> a, **scale**|**s=** λ) **expGamma**(<**shape**|**sp=**> a, **iscale**|**is=** λ)	log transformation of a gamma distribution with shape a and scale or inverse-scale λ: $\theta \sim$ **gamma**(a, λ) $\Leftrightarrow \log(\theta) \sim$ **expgamma**(a, λ). You can use the alias **egamma** for this distribution.
expichisq(<**df=**> ν)	log transformation of an inverse χ^2 distribution with ν degrees of freedom: $\theta \sim$ **ichisq**(ν) $\Leftrightarrow \log(\theta) \sim$ **expichisq**(ν). You can use the alias **eichisq** for this distribution.
expiGamma(<**shape**|**sp=**> a, **scale**|**s=** λ) **expiGamma**(<**shape**|**sp=**> a, **iscale**|**is=** λ)	log transformation of an inverse-gamma distribution with shape a and scale or inverse-scale λ: $\theta \sim$ **igamma**(a, λ) $\Leftrightarrow \log(\theta) \sim$ **expigamma**(a, λ). You can use the alias **eigamma** for this distribution.
expsichisq(<**df=**> ν, <**scale**|**s=**> s)	log transformation of a scaled inverse χ^2 distribution with ν degrees of freedom and scale parameter s: $\theta \sim$ **sichisq**(ν) $\Leftrightarrow \log(\theta) \sim$ **expsichisq**(ν). You can use the alias **esichisq** for this distribution.
expon(**scale**|**s=** λ) **expon**(**iscale**|**is=** λ)	exponential distribution with scale or inverse-scale parameter λ
gamma(<**shape**|**sp=**> a, **scale**|**s=** λ) **gamma**(<**shape**|**sp=**> a, **iscale**|**is=** λ)	gamma distribution with shape a and scale or inverse-scale λ
geo(<**prob**|**p=**> p)	geometric distribution with probability p

Table 52.2 (continued)

Distribution Name	Definition
general(*ll*)	general log likelihood function that you construct using SAS programming statements for a single or multiple continuous variables. The argument *ll* is an expression for the log of the distribution. If there are multiple variables specified before the tilde in a MODEL, PRIOR, or HYPERPRIOR statement, *ll* is interpreted as the log of the joint distribution for these variables. Note that in the MODEL statement, the response variable specified before the tilde is just a place holder and is of no consequence; the variable must have appeared in the construction of *ll* in the programming statements. **general**(*constant*) is equivalent to a uniform distribution on the real line. You can use the alias **logden** for this distribution.
ichisq(<**df=**>ν)	inverse χ^2 distribution with ν degrees of freedom
igamma(<**shape**|**sp=**> a, **scale**|**s=** λ) **igamma**(<**shape**|**sp=**> a, **iscale**|**is=** λ)	inverse-gamma distribution with shape a and scale or inverse-scale λ
laplace(<**location**|**loc**|**l=**> θ, **scale**|**s=** λ) **laplace**(<**location**|**loc**|**l=**> θ, **iscale**|**is=** λ)	Laplace distribution with location θ and scale or inverse-scale λ. This is also known as the *double exponential* distribution. You can use the alias **dexpon** for this distribution.
logistic(<**location**|**loc**|**l=**> a, <**scale**|**s=**> b)	logistic distribution with location a and scale b
lognormal(<**mean**|**m=**> μ, **sd=** λ) **lognormal**(<**mean**|**m=**> μ, **var**|**v=** λ) **lognormal**(<**mean**|**m=**> μ, **prec=** λ)	log-normal distribution with mean μ and standard deviation or variance or precision λ. You can use the aliases **lognormal** or **lnorm** for this distribution.
negbin(<**n=**> n, <**prob**|**p=**> p)	negative binomial distribution with count n and probability of success p. You can use the alias **nb** for this distribution.
normal(<**mean**|**m=**> μ, **sd=** λ) **normal**(<**mean**|**m=**> μ, **var**|**v=** λ) **normal**(<**mean**|**m=**> μ, **prec=** λ)	normal (Gaussian) distribution with mean μ and standard deviation or variance or precision λ. You can use the aliases **gaussian**, **norm**, or **n** for this distribution.
pareto(<**shape**|**sp=**> a, <**scale**|**s=**> b)	Pareto distribution with shape a and scale b
poisson(<**mean**|**m=**> λ)	Poisson distribution with mean λ

Table 52.2 (continued)

Distribution Name	Definition
sichisq(<**df=**> ν, <**scale**|**s=**> s)	scaled inverse χ^2 distribution with ν degrees of freedom and scale parameter s
t(<**mean**|**m=**> μ, **sd=** λ, <**df=**> ν) **t**(<**mean**|**m=**> μ, **var**|**v=** λ, <**df=**> ν) **t**(<**mean**|**m=**> μ, **prec=** λ, <**df=**> ν)	t distribution with mean μ, standard deviation or variance or precision λ, and ν degrees of freedom
uniform(<**left**|**l=**> a, <**right**|**r=**> b)	uniform distribution with range a and b. You can use the alias **unif** for this distribution.
wald(<**mean**|**m=**> μ, <**iscale**|**is=**> λ)	Wald distribution with mean parameter μ and inverse scale parameter λ. This is also known as the *Inverse Gaussian* distribution. You can use the alias **igaussian** for this distribution.
weibull(μ, c, σ)	Weibull distribution with location (threshold) parameter μ, shape parameter c, and scale parameter σ.

PARMS Statement

PARMS *name | (name-list) < = > number < name | (name-list) <=> number . . . >< / NORMAL | T < (df) > | UDS >* **;**

The PARMS statement lists the names of the parameters in the model and specifies optional initial values for these parameters. Multiple PARMS statements are allowed. Each PARMS statement defines a block of parameters, and the blocked Metropolis algorithm updates the parameters in each block simultaneously. See the section "Blocking of Parameters" on page 3523 for more details. PROC MCMC generates missing initial values from the prior distributions whenever needed, as long as they are the standard distributions and not the functions GENERAL or DGENERAL.

Every parameter in the PARMS statement must have a corresponding prior distribution in the PRIOR statement. The program exits if the one-to-one requirement is not satisfied.

The optional arguments give you control over different samplers explicitly for that block of parameters. The normal proposal distribution in the random walk Metropolis is the default. You can also choose a *t*-distribution with *df* degrees of freedom. If $df > 100$, the normal distribution is used instead.

The user defined sampler (UDS, see the section "UDS Statement" on page 3518) option allows you to implement a new sampler for any of the parameters in the block. PROC MCMC does not use the Metropolis sampler on these parameters and incorporates your sampler to draw posterior samples. This can sometimes greatly improve the convergence and mixing of the Markov chain. This functionality is for advanced users, and you should proceed with caution.

PRIOR/HYPERPRIOR Statement

PRIOR *parameter-list* ~ *distribution* **;**

HYPERPRIOR *parameter-list* ~ *distribution* **;**

HYPER *parameter-list* ~ *distribution* **;**

The PRIOR statement is used to specify the prior distribution of the model parameters. You must specify a single parameter or a list of parameters, a tilde (~), and then a distribution with its parameters. Multiple PRIOR statements are allowed for defining models with multiple independent prior components. The log of the prior is the sum of the log prior values from each of the PRIOR statements. See the section "MODEL Statement" on page 3512 for the names of the standard distributions and the section "Standard Distributions" on page 3530 for density specification.

The PRIOR statements are processed twice at every Markov chain simulation—that is, twice per pass through the data set. The statements are called at the first and the last observation of the data set. This is the same as how the BEGINNODATA and ENDNODATA statements are processed.

The HYPERPRIOR statement is internally treated the same as the PRIOR statement. It provides a notational convenience in case you wish to fit a multilevel hierarchical model. It is used to specify the hyperprior distribution of the prior distribution parameters. The log of the hyperprior is the sum of the log hyperprior values from each of the HYPERPRIOR statements.

If you want to specify a multilevel hierarchical model, you can use either a PRIOR or a HYPERPRIOR statement as if it were a hyper-HYPERPRIOR statement. Your model can have as many hierarchical levels as desired.

Programming Statements

This section lists the programming statements available in PROC MCMC to compute the priors and log-likelihood functions. This section also documents the differences between programming statements in PROC MCMC and programming statements in the DATA step. The syntax of programming statements used in PROC MCMC is identical to that used in the NLMIXED procedure (see Chapter 61, "The NLMIXED Procedure") and the MODEL procedure (see Chapter 18, "The MODEL Procedure" (*SAS/ETS User's Guide*),). Most of the programming statements that can be used in the DATA step can also be used in PROC MCMC. Refer to *SAS Language Reference: Dictionary* for a description of SAS programming statements.

There are also a number of unique functions in PROC MCMC that calculate the log density of various distributions in the procedure. You can find them at the section "Using Density Functions in the Programming Statements" on page 3542.

For the list of matrix-based functions that is supported in PROC MCMC, see the section "Matrix Functions in PROC MCMC" on page 3551.

The following are valid statements:

ABORT;
CALL *name [(expression [, expression . . .])]*;
DELETE;
DO *[variable = expression*
*[***TO** *expression] [***BY** *expression]*
[, expression [**TO** *expression] [* **BY** *expression] . . .]*
]
[**WHILE** *expression] [* **UNTIL** *expression]*;
END;
GOTO *statement_label*;
IF *expression*;
IF *expression* **THEN** *program_statement*;
ELSE *program_statement*;
variable = expression;
variable + expression;
LINK *statement_label*;
PUT *[variable] [=] [...]*;
RETURN;
SELECT[(*expression* **)];**
STOP;
SUBSTR(*variable, index, length* **)=** *expression*;
WHEN *(expression) program_statement*;
OTHERWISE *program_statement*;

For the most part, the SAS programming statements work the same as they do in the DATA step, as documented in *SAS Language Reference: Concepts*. However, there are several differences:

- The ABORT statement does not allow any arguments.
- The DO statement does not allow a character index variable. Thus

  ```
  do i = 1,2,3;
  ```

  is supported; however, the following statement is not supported:

  ```
  do i = 'A','B','C';
  ```

- The PUT statement, used mostly for program debugging in PROC MCMC (see the section "Handling Error Messages" on page 3568), supports only some of the features of the DATA step PUT statement, and it has some features that are not available with the DATA step PUT statement:
  - The PROC MCMC PUT statement does not support line pointers, factored lists, iteration factors, overprinting, _INFILE_, _OBS_, the colon (:) format modifier, or "$".
  - The PROC MCMC PUT statement does support expressions, but the expression must be enclosed in parentheses. For example, the following statement displays the square root of x:

```
        put (sqrt(x));
```

- The WHEN and OTHERWISE statements enable you to specify more than one target statement. That is, DO/END groups are not necessary for multiple statement WHENs. For example, the following syntax is valid:

```
select;
   when (exp1) stmt1;
               stmt2;
   when (exp2) stmt3;
               stmt4;
end;
```

You should avoid defining variables that begin with an underscore (_). They might conflict with internal variables created by PROC MCMC. The MODEL statement must come after any SAS programming statements that define or modify terms used in the construction of the log likelihood.

UDS Statement

UDS *subroutine-name (subroutine-argument-list)* **;**

UDS stands for user defined sampler. The UDS statement allows you to use a separate algorithm, other than the default random walk Metropolis, to update parameters in the model. The purpose of the UDS statement is to give you a greater amount of flexibility and better control over the updating schemes of the Markov chain. Multiple UDS statements are allowed.

For the UDS statement to work properly, you have to do the following:

- write a subroutine by using PROC FCMP (see the FCMP Procedure in the *Base SAS Procedures Guide*) and save it to a SAS catalog (see the example in this section). The subroutine must update some parameters in the model. These are the UDS parameters. The subroutine is called the UDS subroutine.
- declare any UDS parameters in the PARMS statement with a sampling option, as in </ UDS > (see the section "PARMS Statement" on page 3515).
- specify the prior distributions for all UDS parameters, using the PRIOR statements.

NOTE: All UDS parameters must appear in three places: the UDS statement, the PARMS statement, and the PRIOR statement. Otherwise, PROC MCMC exits.

To obtain a valid Markov chain, a UDS subroutine must update a parameter from its full posterior conditional distribution and not the posterior marginal distribution. The posterior conditional is something that you need to provide. This conditional is implicitly based on a prior distribution. PROC MCMC has no means to verify that the implied prior in the UDS subroutine is the same as the prior that you specified in the PRIOR statement. You need to make sure that the two distributions agree; otherwise, you will get misleading results.

The priors in the PRIOR statements do not directly affect the sampling of the UDS parameters. They could affect the sampling of the other parameters in the model, which, in turn, changes the behavior of the Markov chain. You can see this by noting cases where the hyperparameters of the UDS parameters are model parameters; the priors should be part of the posterior conditional distributions of these hyperparameters, and they cannot be omitted.

Some additional information is listed to help you better understand the UDS statement:

- Most features of the SAS programming language can be used in subroutines processed by PROC FCMP (see the FCMP Procedure in the *Base SAS Procedures Guide*).
- The UDS statement does not support FCMP functions—a FCMP function returns a value, while a subroutine does not. A subroutine updates some of its subroutine arguments. These arguments are called OUTARGS arguments.
- The UDS parameters cannot be in the same block as other parameters. The optional argument </ UDS > in the PARMS statement prevents parameters that use the default Metropolis from being mixed with those that are updated by the UDS subroutines.
- You can put all the UDS parameters in the same PARMS statement or have a separate UDS statement for each of them.
- The same subroutine can be used in multiple UDS statements. This feature comes in handy if you have a generic sampler that can be applied to different parameters.
- PROC MCMC updates the UDS parameters by calling the UDS subroutines directly. At every iteration, PROC MCMC first samples parameters that use the Metropolis algorithm, then the UDS parameters. Sampling of the UDS parameters proceeds in the order in which the UDS statements are listed.
- A UDS subroutine accepts any symbols in the program as well as any input data set variables as its arguments.
- Only the OUTARGS arguments in a UDS subroutine are updated in PROC MCMC. You can modify other arguments in the subroutine, but the changes are not global in the procedure.
- If a UDS subroutine has an argument that is a SAS data set variable, PROC MCMC steps through the data set while updating the UDS parameters. The subroutine is called once per observation in the data set for every iteration.
- If a UDS subroutine does not have any arguments that are data set variables, PROC MCMC does not access the data set while executing the subroutine. The subroutine is called once per iteration.
- To reduce the overhead in calling the UDS subroutine and accessing the data set repeatedly, you might consider reading all the input data set variables into arrays and using the arrays as the subroutine arguments. See the section "BEGINCNST/ENDCNST Statement" on page 3509 about how to use the BEGINCNST and ENDCNST statements to store data set variables.

An Example that Uses the UDS Statement

Suppose that you are interested in modeling normal data with conjugate prior distributions. The data are as follows:

```
title 'An Example that uses the UDS Statement';

data a;
   input y @@;
   i = _n_;
   datalines;
-0.651  17.435  -5.943  -2.543 -10.444
-5.754  -5.002  -2.545  -1.743   0.998
;
```

The likelihood for each observation is as follows:

$$f(y_i|\mu,\sigma) = \phi(\mu, \text{var} = \sigma^2)$$

The prior distributions on μ and σ^2 are as follows:

$$\begin{aligned} \pi(\mu|\mu_0,\tau_0^2) &= \phi(\mu_0, \text{var} = \tau_0^2) \\ \pi(\sigma^2|\nu_0,\sigma_0^2) &= f_{si\chi^2}(\text{shape} = \nu_0, \text{scale} = \sigma_0^2) \end{aligned}$$

where $f_{si\chi^2}$ is the density function for a scaled inverse chi-square distribution. To sample μ and σ^2 without using any UDS statements, you can use the following program:

```
proc mcmc data=a seed=17;
   parm mu;
   parm s2;
   begincnst;
      mu0 = 0;  t0 = 20;
      nu0 = 10; s0 = 10;
   endcnst;

   prior mu ~ normal(mu0, var=t0);
   prior s2 ~ sichisq(nu0, s0);
   model y ~ normal(mu, var = s2);
run;
```

This is a case where the full posterior conditional distribution of μ given σ^2 and y has a closed form. It is also a normal distribution:

$$p(\mu|\sigma^2,y) = \phi\left(\frac{\frac{\mu_0}{\tau_0^2} + \frac{n\bar{y}}{\sigma^2}}{\frac{1}{\tau_0^2} + \frac{n}{\sigma^2}}, \frac{1}{\frac{1}{\tau_0} + \frac{n}{\sigma^2}}\right)$$

You can define a subroutine, **muupdater**, which generates a random normal sample from the posterior conditional distribution described previously.

```
proc fcmp outlib=sasuser.funcs.uds;
   subroutine muupdater(mu, s2, mu0, t0, n, sumy);
   outargs mu;
   sigma2 = 1 / (1/t0 + n/s2);
   mean = (mu0/t0 + sumy/s2) * sigma2;
   mu = rand("normal", mean, sqrt(sigma2));
   endsub;
run;
```

The subroutine is saved in the OUTLIB= library. The declaration of any subroutine begins with a SUBROUTINE statement and ends with an ENDSUB statement. The OUTARGS statement in the subroutine indicates that mu is updated. Others, such as sigma2, mu0, and so on, are arguments that are needed in the full conditional distribution. Here the rand and sqrt are two of the many SAS functions that you can use.

You specify a CMPLIB option to let SAS search each of the catalogs that are specified in the option for a package that contains **muupdater**.

```
options cmplib=sasuser.funcs;
```

To use the subroutine in the UDS statement, you can use the following statements:

```
proc mcmc data=a seed=17;
   UDS muupdater(mu, s2, mu0, t0, n, sumy);
   parm mu /uds;
   parm s2;
   begincnst;
      mu0 = 0;  t0 = 20;
      nu0 = 10; s0 = 10;
      n = 10;
      if i eq 1 then sumy = 0;
      sumy = sumy + y;
      call streaminit(1);
   endcnst;

   prior mu ~ normal(mu0, var=t0);
   prior s2 ~ sichisq(nu0, s0);
   model y ~ normal(mu, var = s2);
run;
```

These statements are very similar to the previous program. The differences are the UDS statement, the </ UDS > option in the PARMS statement, and a few lines that computes the values of sumy and n.

The symbol sumy is the sum of y. The value is obtained by taking advantage of the BEGINCNST and ENDCNST statements. See the example in the section "BEGINCNST/ENDCNST Statement" on page 3509. The symbol n is the sample size in the data set.

The CALL STREAMINIT routine ensures that the RAND function in **muupdater** creates a reproducible stream of random numbers. The SEED= option specifies a seed for the random number generator in PROC MCMC, which does not control the random number generator in the RAND function in the subroutine. You need to set both to reproduce the same stream of Markov chain

samples.

The two programs produce different but similar numbers (results not shown) for the posterior distributions of μ and σ^2.

For a more realistic example that uses the UDS statement, see "Example 52.11: Implement a New Sampling Algorithm" on page 3672.

Details: MCMC Procedure

How PROC MCMC Works

PROC MCMC uses a random walk Metropolis algorithm to obtain posterior samples. For details on the Metropolis algorithm, see the section "Metropolis and Metropolis-Hastings Algorithms" on page 152. For the actual implementation details of the Metropolis algorithm in PROC MCMC, such as the blocking of the parameters and tuning of the covariance matrices, see the section "Tuning the Proposal Distribution" on page 3525. By default, PROC MCMC assumes that all observations in the data set are independent, and the logarithm of the posterior density is calculated as follows:

$$\log(p(\theta|\mathbf{y})) = \log(\pi(\theta)) + \sum_{i=1}^{n} \log(f(y_i|\theta))$$

where θ is a parameter or a vector of parameters. The term $\log(\pi(\theta))$ is the sum of the log of the prior densities specified in the PRIOR and HYPERPRIOR statements. The term $\log(f(y_i|\theta))$ is the log likelihood specified in the MODEL statement. The MODEL statement specifies the log likelihood for a single observation in the data set.

The statements in PROC MCMC are in many ways like DATA step statements; PROC MCMC evaluates every statement in order for each observation. The procedure cumulatively adds the log likelihood for each observation. Statements between the BEGINNODATA and ENDNODATA statements are evaluated only at the first and the last observations. At the last observation, the log of the prior and hyperprior distributions is added to the sum of the log likelihood to obtain the log of the posterior distribution.

With multiple PARMS statements (multiple blocks of parameters), PROC MCMC updates each block of parameters while holding the others constants. The procedure still steps through all of the programming statements to calculate the log of the posterior distribution, given the current or the proposed values of the updating block of parameters. In other words, the procedure does not calculate the conditional distribution explicitly for each block of parameters, and it uses the full joint distribution in the Metropolis step for every block update. If you wish to model dependent data—that is, $\log(f(\mathbf{y}|\theta)) \neq \sum_i \log(f(y_i|\theta))$—you can use the PROC option JOINTMODEL. See the section "Modeling Joint Likelihood" on page 3556 for more details.

Blocking of Parameters

In a multivariate parameter model, if all k parameters are proposed with one joint distribution $q(\cdot|\cdot)$, acceptance or rejection would occur for all of them. This can be rather inefficient, especially when parameters have vastly different scales. A way to avoid this difficulty is to allocate the k parameters into d blocks and update them separately. The PARMS statement is used to specify model parameters. It also puts parameters in separate blocks, and each block of parameters is updated sequentially in the procedure.

Suppose that you wish to sample from a multivariate distribution with probability density function $p(\theta|\mathbf{y})$ where $\theta = \{\theta_1, \theta_2, \ldots, \theta_k\}$. Now suppose that these k parameters are separated into d blocks—for example, $p(\theta|\mathbf{x}) = f_d(z)$ where $z = \{z_1, z_2, \ldots, z_d\}$, where each z_j contains a nonempty subset of the $\{\theta_i\}$, and where each θ_i is contained in one and only one z_j. In the MCMC context, the z's are blocks of parameters. In the blocked algorithm, a proposal is composed of several parts. Instead of proposing a simultaneous move for all the θ's, a proposal is made for the θ_i's in z_1 only, then for the θ_i's in z_2, and so on for d subproposals. Any accepted proposal can involve any number of the blocks moving. Not necessarily all of the parameters move at once as in the all-at-once Metropolis algorithm.

Formally, the blocked Metropolis algorithm is as follows. Let w_j be the collection of θ_i that are in block z_j and let $q_j(\cdot|w_j)$ be a symmetric multivariate distribution centered at the current values of w_j.

1. Let $t = 0$. Choose points for all w_j^t. This can be an arbitrary point as long as $p(w_j^t|\mathbf{y}) > 0$.
2. For $j = 1, \cdots, d$:
   a) Generate a new sample, $w_{j,new}$, using the proposal distribution $q_j(\cdot|w_j^t)$.
   b) Calculate the following quantity:

$$r = \min\left\{\frac{p(w_{j,new}|w_1^t, \cdots, w_{j-1}^t, w_{j+1}^{t-1}, \cdots, w_d^t, \mathbf{y})}{p(w_j^t|w_1^t, \cdots, w_{j-1}^t, w_{j-1}^{t+1}, \cdots, w_d^t, \mathbf{y})}, 1\right\}.$$

   c) Sample u from the uniform distribution $U(0, 1)$.
   d) Set $w_j^{t+1} = w_{j,new}$ if $r < a$; $w_j^{t+1} = w_j^t$ otherwise.
3. Set $t = t + 1$. If $t < T$, the number of desired samples, go back to Step 2; otherwise, stop.

With PROC MCMC, you can sample all parameters simultaneously by putting them all in a single PARMS statement, you can sample parameters individually by putting each parameter in its own PARMS statement, or you can sample certain subsets of parameters together by grouping each subset in its own PARMS statements. For example, if the model you are interested in has five parameters, alpha, beta, gamma, phi, sigma, the all-at-once strategy is as follows:

```
parms alpha beta gamma phi sigma;
```

The one-at-a-time strategy is as follows:

```
parms alpha;
parms beta;
parms gamma;
parms phi;
parms sigma;
```

A two-block strategy could be as follows:

```
parms alpha beta gamma;
parms phi sigma;
```

One of the greatest challenges in MCMC sampling is achieving good mixing of the chains—the chains should quickly traverse the support of the stationary distribution. A number of factors determine the behavior of a Metropolis sampler; blocking is one of them, so you want to be extra careful when it comes to choosing a good design. Generally speaking, forming blocks of parameters has its advantages, but it is not true that the larger the block the faster the convergence.

When simultaneously sampling a large number of parameters, the algorithm might find it difficult to achieve good mixing. As the number of parameters gets large, it is much more likely to have (proposal) samples that fall well into the tails of the target distribution, producing too small a test ratio. As a result, few proposed values are accepted and convergence is slow. On the other hand, when sampling each parameter individually, the chain might mix far too slowly because the conditional distributions (of θ_i given all other θ's) might be very "narrow." Hence, it takes a long time for the chain to explore fully that dimension alone. There are no theoretical results that can help determine an optimal "blocking" for an arbitrary parametric model. A rule followed in practice is to form small groups of correlated parameters that belong to the same context in the formulation of the model. The best mixing is usually obtained with a blocking strategy somewhere between the all-at-once and one-at-a-time strategies.

Samplers

This section describes the sampling methods used in PROC MCMC. Each block of parameters is classified by the nature of the prior distributions. "Continuous" means all priors of the parameters in the same block are continuous distribution. "Discrete" means all priors are discrete. "Mixed" means that some parameters are continuous and others are discrete. Parameters that have binary priors are treated differently, as indicated in the table. MVN stands for the multivariate normal distribution, and MVT is short for the multivariate *t*-distribution.

Blocks	Default Method	Alternative Method
continuous	MVN	MVT
discrete (other than binary)	binned MVN	binned MVT or symmetric geometric
mixed	MVN	MVT
binary (single dimensional)	inverse CDF	
binary (multi-dimensional)	independence sampler	

For a block of continuous parameters, PROC MCMC uses a multivariate normal distribution as the default proposal distribution. In the tuning phase, the procedure finds an optimal scale c and a tuning covariance matrix Σ.

For a discrete block of parameters, PROC MCMC uses a discretized multivariate normal distribution as the default proposal distribution. The scale c and covariance matrix $\mathbf{\Sigma}$ are tuned. Alternatively, you can use an independent symmetric geometric proposal distribution. The density has form $\frac{p(1-p)^{|\theta|}}{2(1-p)}$ and has variance $\frac{(2-p)(1-p)}{p^2}$. In the tuning phase, the procedure finds an optimal proposal probability p for every parameter in the block.

You can change the proposal distribution, from the normal to a t-distribution. You can either use the PROC option PROPDIST=T(*df*) or PARMS statement option </ T(*df*)> to make the change. The t-distributions have thicker tails, and they can propose to the tail areas more efficiently than the normal distribution. It can help with the mixing of the Markov chain if some of the parameters have a skewed tails. See "Example 52.4: Nonlinear Poisson Regression Models" on page 3605. The independence sampler (see the section "Independence Sampler" on page 153) is used for a block of binary parameters. The inverse CDF method is used for a block that consists of a single binary parameter.

Tuning the Proposal Distribution

One key factor in achieving high efficiency of a Metropolis-based Markov chain is finding a good proposal distribution for each block of parameters. This process is referred to as tuning. The tuning phase consists of a number of loops. The minimum number of loops is controlled by the option MINTUNE=, with a default value of 2. The option MAXTUNE= controls the maximum number of tuning loops, with a default value of 24. Each loop lasts for NTU= iterations, where by default NTU= 500. At the end of every loop, PROC MCMC examines the acceptance probability for each block. The acceptance probability is the percentage of NTU= proposals that have been accepted. If the probability falls within the acceptance tolerance range (see the section "Scale Tuning" on page 3526), the current configuration of $c/\mathbf{\Sigma}$ or p is kept. Otherwise, these parameters are modified before the next tuning loop.

Continuous Distribution: Normal or *t*-Distribution

A good proposal distribution should resemble the actual posterior distribution of the parameters. Large sample theory states that the posterior distribution of the parameters approaches a multivariate normal distribution (see Gelman et al. 2004, Appendix B, and Schervish 1995, Section 7.4). That is why a normal proposal distribution often works well in practice. The default proposal distribution in PROC MCMC is the normal distribution: $q_j(\theta_{\text{new}}|\theta^t) = \text{MVN}(\theta_{\text{new}}|\theta^t, c^2\mathbf{\Sigma})$. As an alternative, you can choose a multivariate t-distribution as the proposal distribution. It is a good distribution to use if you think that the posterior distribution has thick tails and a t-distribution can improve the mixing of the Markov chain. See "Example 52.4: Nonlinear Poisson Regression Models" on page 3605.

Scale Tuning

The acceptance rate is closely related to the sampling efficiency of a Metropolis chain. For a random walk Metropolis, high acceptance rate means that most new samples occur right around the current data point. Their frequent acceptance means that the Markov chain is moving rather slowly and not exploring the parameter space fully. On the other hand, a low acceptance rate means that the proposed samples are often rejected; hence the chain is not moving much. An efficient Metropolis sampler has an acceptance rate that is neither too high nor too low. The scale c in the proposal distribution $q(\cdot|\cdot)$ effectively controls this acceptance probability. Roberts, Gelman, and Gilks (1997) showed that if both the target and proposal densities are normal, the optimal acceptance probability for the Markov chain should be around 0.45 in a single dimensional problem, and asymptotically approaches 0.234 in higher dimensions. The corresponding optimal scale is 2.38, which is the initial scale set for each block.

Due to the nature of stochastic simulations, it is impossible to fine-tune a set of variables such that the Metropolis chain has the exact desired acceptance rate. In addition, Roberts and Rosenthal (2001) empirically demonstrated that an acceptance rate between 0.15 and 0.5 is at least 80% efficient, so there is really no need to fine-tune the algorithms to reach acceptance probability that is within small tolerance of the optimal values. PROC MCMC works with a probability range, determined by the PROC options TARGACCEPT ± ACCEPTTOL. The default value of TARGACCEPT is a function of the number of parameters in the model, as outlined in Roberts, Gelman, and Gilks (1997). The default value of ACCEPTTOL is 0.075. If the observed acceptance rate in a given tuning loop is less than the lower bound of the range, the scale is reduced; if the observed acceptance rate is greater than the upper bound of the range, the scale is increased. During the tuning phase, a scale parameter in the normal distribution is adjusted as a function of the observed acceptance rate and the target acceptance rate. The following updating scheme is used in PROC MCMC [1]:

$$c_{\text{new}} = \frac{c_{\text{cur}} \cdot \Phi^{-1}(p_{\text{opt}}/2)}{\Phi^{-1}(p_{\text{cur}}/2)}$$

where c_{cur} is the current scale, p_{cur} is the current acceptance rate, p_{opt} is the optimal acceptance probability.

Covariance Tuning

To tune a covariance matrix, PROC MCMC takes a weighted average of the old proposal covariance matrix and the recent observed covariance matrix, based on NTU samples in the current loop. The TUNEWT=w option determines how much weight is put on the recently observed covariance matrix. The formula used to update the covariance matrix is as follows:

$$\text{COV}_{\text{new}} = w\,\text{COV}_{\text{cur}} + (1-w)\text{COV}_{\text{old}}$$

There are two ways to initialize the covariance matrix:

[1] Roberts, Gelman, and Gilks (1997) and Roberts and Rosenthal (2001) demonstrate that the relationship between acceptance probability and scale in a random walk Metropolis is $p = 2\Phi\left(-\sqrt{I}c/2\right)$, where c is the scale, p is the acceptance rate, Φ is the CDF of a standard normal, and $I \equiv E_f[(f'(x)/f(x))^2]$, $f(x)$ is the density function of samples. This relationship determines the updating scheme, with I being replaced by the identity matrix to simplify calculation.

- The default is an identity matrix multiplied by the initial scale of 2.38 (controlled by the PROC option SCALE=) and divided by the square root of the number of estimated parameters in the model. It can take a number of tuning phases before the proposal distribution is tuned to its optimal stage, since the Markov chain needs to spend time learning about the posterior covariance structure. If the posterior variances of your parameters vary by more than a few orders of magnitude, if the variances of your parameters are much different from 1, or if the posterior correlations are high, then the proposal tuning algorithm might have difficulty with forming an acceptable proposal distribution.

- Alternatively, you can use a numerical optimization routine, such as the quasi-Newton method, to find a starting covariance matrix. The optimization is performed on the joint posterior distribution, and the covariance matrix is a quadratic approximation at the posterior mode. In some cases this is a better and more efficient way of initializing the covariance matrix. However, there are cases, such as when the number of parameters is large, where the optimization could fail to find a matrix that is positive definite. In that case, the tuning covariance matrix is reset to the identity matrix.

A side product of the optimization routine is that it also finds the *maximum a posteriori* (MAP) estimates with respect to the posterior distribution. The MAP estimates are used as the initial values of the Markov chain.

If any of the parameters are discrete, then the optimization is performed conditional on these discrete parameters at their respective fixed initial values. On the other hand, if all parameters are continuous, you can in some cases skip the tuning phase (by setting MAXTUNE=0) or the burn-in phase (by setting NBI=0).

Discrete Distribution: Symmetric Geometric

By default, PROC MCMC uses the normal density as the proposal distribution in all Metropolis random walks. For parameters that have discrete prior distributions, PROC MCMC discretizes proposed samples. You can choose an alternative symmetric geometric proposal distribution by specifying the option DISCRETE=GEO.

The density of the symmetric geometric proposal distribution is as follows:

$$\frac{p_g(1-p_g)^{|\theta|}}{2(1-p_g)}$$

where the symmetry centers at θ. The distribution has a variance of

$$\sigma^2 = \frac{(2-p_g)(1-p_g)}{p_g^2}$$

Tuning for the proposal p_g uses the following formula:

$$\frac{\sigma_{\text{new}}}{\sigma_{\text{cur}}} = \frac{\Phi^{-1}(p_{\text{opt}}/2)}{\Phi^{-1}(p_{\text{cur}}/2)}$$

where σ_{new} is the standard deviation of the new proposal geometric distribution, σ_{cur} is the standard deviation of the current proposal distribution, p_{opt} is the target acceptance probability, and p_{cur} is the current acceptance probability for the discrete parameter block.

The updated p_g is the solution to the following equation that is between 0 and 1 :

$$\sqrt{\frac{(2-p_g)(1-p_g)}{p_g^2}} = \frac{\sigma_{\text{cur}} \cdot \Phi^{-1}(p_{\text{opt}}/2)}{\Phi^{-1}(p_{\text{cur}}/2)}$$

Binary Distribution: Independence Sampler

Blocks consisting of a single parameter with a binary prior do not require any tuning; the inverse-CDF method applies. Blocks that consist of multiple parameters with binary prior are sampled by using an independence sampler with binary proposal distributions. See the section "Independence Sampler" on page 153. During the tuning phase, the success probability p of the proposal distribution is taken to be the probability of acceptance in the current loop. Ideally, an independence sampler works best if the acceptance rate is 100%, but that is rarely achieved. The algorithm stops when the probability of success exceeds the TARGACCEPTI=*value*, which has a default value of 0.6.

Initial Values of the Markov Chains

You can assign initial values to any parameters. To assign initial values, you can either use the PARMS statements or use programming statements within the BEGINCNST and ENDCNST statements. For the latter approach, see the section "BEGINCNST/ENDCNST Statement" on page 3509.

When parameters have missing initial values, PROC MCMC tries to generate them from the respective prior distributions, as long as the distributions are listed in the section "Standard Distributions" on page 3530. PROC MCMC either uses the mode from the prior distribution or draws a random number from it. For distributions that do not have modes, such as the uniform distribution, PROC MCMC uses the mean instead. In general, PROC MCMC avoids using starting values that are close to the boundary of support of the prior distribution. For example, the exponential prior has a mode at 0, and PROC MCMC starts an initial value at the mean. This avoids some potential numerical problems. If you use the GENERAL or DGENERAL functions in the PRIOR statements, you must provide initial values for those parameters.

If you use the optimization option PROPCOV, PROC MCMC starts the tuning at the optimized values. The procedure overwrites the initial values that you provided unless you use the option INIT=REINIT.

Assignments of Parameters

In general, you cannot alter the values of any model parameters in PROC MCMC. For example, the following assignment statement produces an error:

```
parms alpha;
alpha = 27;
```

This restriction prevents incorrect calculation of the posterior density—assignments of parameters in the program would override the parameter values generated by the procedure and lead to a constant value of the density function.

However, you can modify parameter values and assign initial values to parameters within the block defined by the BEGINCNST and ENDCNST statements. The following syntax is allowed:

```
parms alpha;
begincnst;
   alpha = 27;
endcnst;
```

The initial value of alpha is 27. Assignments within the BEGINCNST/ENDCNST block override initial values specified in the PARMS statement. For example, with the following statements, the Markov chain starts at alpha $= 27$, not 23.

```
parms alpha 23;
begincnst;
   alpha = 27;
endcnst;
```

This feature enables you to systematically assign initial values. Suppose that z is an array parameter of the same length as the number of observations in the input data set. You want to start the Markov chain with each z_i having a different value depending on the data set variable y. The following statements set $z_i = |y|$ for the first half of the observations and $z_i = 2.3$ for the rest:

```
/* a rather artificial input data set. */
data inputdata;
   do ind = 1 to 10;
      y = rand('normal');
      output;
   end;
run;

proc mcmc data=inputdata;
   array z[10];
   begincnst;
      if ind <= 5 then z[ind] = abs(y);
      else z[ind] = 2.3;
   endcnst;
   parms z:;
   prior z: ~ normal(0, sd=1);
   model general(0);
run;
```

Elements of z are modified as PROC MCMC executes the programming statements between the BEGINCNST and ENDCNST statements. This feature could be useful when you use the GENERAL function and you find that the PARMS statements are too cumbersome for assigning starting values.

Standard Distributions

Table 52.4 through Table 52.31 show all densities that PROC MCMC recognizes. These densities can be used in the MODEL, PRIOR, and HYPERPRIOR statements. See the section "Using Density Functions in the Programming Statements" on page 3542 for information about how to use distributions in the programming statements. To specify an arbitrary distribution, you can use the functions GENERAL and DGENERAL. See the section "Specifying a New Distribution" on page 3541 for more details. See the section "Truncation and Censoring" on page 3544 for tips on how to work with truncated distributions and censoring data.

Table 52.4 Beta Distribution

PROC specification	**beta**(a, b)
density	$\frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)}\theta^{a-1}(1-\theta)^{b-1}$
parameter restriction	$a > 0, b > 0$
range	$\begin{cases} [0,1] & \text{when } a = 1, b = 1 \\ [0,1) & \text{when } a = 1, b \neq 1 \\ (0,1] & \text{when } a \neq 1, b = 1 \\ (0,1) & \text{otherwise} \end{cases}$
mean	$\frac{a}{a+b}$
variance	$\frac{ab}{(a+b)^2(a+b+1)}$
mode	$\begin{cases} \frac{a-1}{a+b-2} & a > 1, b > 1 \\ 0 \text{ and } 1 & a < 1, b < 1 \\ 0 & \begin{cases} a < 1, b \geq 1 \\ a = 1, b > 1 \end{cases} \\ 1 & \begin{cases} a \geq 1, b < 1 \\ a > 1, b = 1 \end{cases} \\ \text{does not exist uniquely} & a = b = 1 \end{cases}$
random number	if $\min(a,b) > 1$, see (Cheng 1978); if $\max(a,b) < 1$, see (Atkinson and Whittaker 1976) and (Atkinson 1979); if $\min(a,b) < 1$ and $\max(a,b) > 1$, see (Cheng 1978); if $a = 1$ or $b = 1$, inversion method; if $a = b = 1$, uniform random variable

Table 52.5 Binary Distribution

PROC specification	**binary**(p)
density	$p^{\theta}(1-p)^{1-\theta}$
parameter restriction	$0 \leq p \leq 1$
range	$\begin{cases} \{0\} & \text{when } p = 0 \\ \{1\} & \text{when } p = 1 \\ \{0,1\} & \text{otherwise} \end{cases}$
mean	round(p)
variance	$p(1-p)$
mode	$\begin{cases} \{1\} & \text{when } p = 1 \\ \{0\} & \text{otherwise} \end{cases}$
random number	generate $u \sim$ uniform$(0,1)$. If $u \leq p$, $\theta = 1$; else, $\theta = 0$

Table 52.6 Binomial Distribution

PROC specification	**binomial**$(n,\ p)$
density	$\binom{n}{\theta} p^{\theta}(1-p)^{n-\theta}$
parameter restriction	$n = 0, 1, 2, \cdots 0 \leq p \leq 1$
range	$\theta \in \{0, \cdots, n\}$
mean	$\lfloor np \rfloor$
variance	$np(1-p)$
mode	$\lfloor (n+1)p \rfloor$

Table 52.7 Cauchy Distribution

PROC specification	**cauchy**$(a,\ b)$
density	$\frac{1}{\pi}\left(\frac{b}{b^2+(\theta-a)^2}\right)$
parameter restriction	$b > 0$
range	$\theta \in (-\infty, \infty)$
mean	does not exist
variance	does not exist
mode	a
random number	generate $u_1, u_2 \sim$ uniform$(0,1)$, let $v = 2u_2 - 1$. Repeat the procedure until $u_1^2 + v^2 < 1$. $y = v/u_1$ is a draw from the standard Cauchy, and $\theta = a + by$ (Ripley 1987)

Table 52.8 χ^2 **Distribution**

PROC specification	**chisq**(ν)
density	$\frac{1}{\Gamma(\nu/2)2^{\nu/2}}\theta^{(\nu/2)-1}e^{-\theta/2}$
parameter restriction	$\nu > 0$
range	$\theta \in [0, \infty)$ if $\nu = 2$; $(0, \infty)$ otherwise
mean	ν
variance	2ν
mode	$\nu - 2$ if $\nu \geq 2$; does not exist otherwise
random number	χ^2 is a special case of the gamma distribution: $\theta \sim$ gamma($\nu/2$, scale=2) is a draw from the χ^2 distribution

Table 52.9 **Exponential** χ^2 **Distribution**

PROC specification	**expchisq**(ν)
density	$\frac{1}{\Gamma(\nu/2)2^{\nu/2}}\exp(\theta)^{\nu/2}\exp(-\exp(\theta)/2)$
parameter restriction	$\nu > 0$
range	$\theta \in (-\infty, \infty)$
mode	$\log(\nu)$
random number	generate $x_1 \sim \chi^2(\nu)$, and $\theta = \log(x_1)$ is a draw from the exponential χ^2 distribution
relationship to the χ^2 distribution	$\theta \sim \chi^2(\nu) \Leftrightarrow \log(\theta) \sim \exp \chi^2(\nu)$

Table 52.10 **Exponential Exponential Distribution**

PROC specification	**expexpon**(scale = b)	**expexpon**(iscale = β)
density	$\frac{1}{b}\exp(\theta)\exp(-\exp(\theta)/b)$	$\beta\exp(\theta)\exp(-\exp(\theta)\cdot\beta)$
parameter restriction	$b > 0$	$\beta > 0$
range	$\theta \in (-\infty, \infty)$	same
mode	$\log(b)$	$\log(1/\beta)$
random number	generate $x_1 \sim$ expon(scale=b), and $\theta = \log(x_1)$ is a draw from the exponential exponential distribution. Note that an exponential exponential distribution is not the same as the double exponential distribution.	
relationship to the Expon distribution	$\theta \sim \text{expon}(b) \Leftrightarrow \log(\theta) \sim \text{expExpon}(b)$	

Table 52.11 Exponential Gamma Distribution

PROC specification	**expgamma**(a, scale = b)	**expgamma**(a, iscale = β)
density	$\frac{1}{b^a\Gamma(a)}e^{a\theta}\exp(-e^{\theta}/b)$	$\frac{\beta^a}{\Gamma(a)}e^{a\theta}\exp(-e^{\theta}\cdot\beta)$
parameter restriction	$a>0, b>0$	$a>0, \beta>0$
range	$\theta\in(-\infty,\infty)$	same
mode	$\log(ab)$	$\log(a/\beta)$
random number	generate $x_1\sim$ gamma$(a,$ scale $=b)$, and $\theta=\log(x_1)$ is a draw from the exponential gamma distribution	
relationship to the Γ distribution	$\theta\sim$ gamma$(a,b)\Leftrightarrow\log(\theta)\sim$ expGamma(a,b)	

Table 52.12 Exponential Inverse χ^2 Distribution

PROC specification	**expichisq**(ν)
density	$\frac{1}{\Gamma(\frac{\nu}{2})2^{\nu/2}}\exp(-\nu\theta/2)\exp(-1/(2\exp(\theta)))$
parameter restriction	$\nu>0$
range	$\theta\in(-\infty,\infty)$
mode	$-\log(\nu)$
random number	generate $x_1\sim i\chi^2(\nu)$, and $\theta=\log(x_1)$ is a draw from the exponential inverse χ^2 distribution
relationship to the $i\chi^2$ distribution	$\theta\sim i\chi^2(\nu)\Leftrightarrow\log(\theta)\sim\exp i\chi^2(\nu)$

Table 52.13 Exponential Inverse-Gamma Distribution

PROC specification	**expigamma**(a, scale = b)	**expigamma**(a, iscale = β)
density	$\frac{b^a}{\Gamma(a)}\exp(-\alpha\theta)\exp(-b/\exp(\theta))$	$\frac{1}{\beta^{\alpha}\Gamma(a)}\exp(-\alpha\theta)\exp(-\frac{1}{\beta\exp(\theta)})$
parameter restriction	$a>0, b>0$	$a>0, \beta>0$
range	$\theta\in(-\infty,\infty)$	same
mode	$-\log(a/b)$	$-\log(a\beta)$
random number	generate $x_1\sim$ igamma$(a,$ scale $=b)$, and $\theta=\log(x_1)$ is a draw from the exponential inverse-gamma distribution	
relationship to the $i\Gamma$ distribution	$\theta\sim$ igamma$(a,b)\Leftrightarrow\log(\theta)\sim$ eigamma(a,b)	

Table 52.14 Exponential Scaled Inverse χ^2 Distribution

PROC specification	**expsichisq**(ν, s)
density	$\frac{(\frac{\nu}{2})^{\nu/2}}{\Gamma(\frac{\nu}{2})} s^{\nu} \exp(-\nu\theta/2) \exp(-\nu s^2/(2\exp(\theta)))$
parameter restriction	$\nu > 0, s > 0$
range	$\theta \in (-\infty, \infty)$
mode	$\log(s^2)$
random number	generate $x_1 \sim si\chi^2(\nu)$, and $\theta = \log(x_1)$ is a draw from the exponential scaled inverse χ^2 distribution
relationship to the $si\chi^2$ distribution	$\theta \sim si\chi^2(\nu) \Leftrightarrow \log(\theta) \sim \exp si\chi^2(\nu)$

Table 52.15 Exponential Distribution

PROC specification	**expon**(scale = b)	**expon**(iscale = β)
density	$\frac{1}{b} e^{-\theta/b}$	$\beta e^{-\beta\theta}$
parameter restriction	$b > 0$	$\beta > 0$
range	$\theta \in [0, \infty)$	same
mean	b	$1/\beta$
variance	b^2	$1/\beta^2$
mode	0	0
random number	the exponential distribution is a special case of the gamma distribution: $\theta \sim$ gamma(1, scale $= b$) is a draw from the exponential distribution	

Table 52.16 Gamma Distribution

PROC specification	**gamma**(a, scale = b)	**gamma**(a, iscale = β)
density	$\frac{1}{b^a \Gamma(a)} \theta^{a-1} e^{-\theta/b}$	$\frac{\beta^a}{\Gamma(a)} \theta^{a-1} e^{-\beta\theta}$
parameter restriction	$a > 0, b > 0$	$a > 0, \beta > 0$
range	$\theta \in [0, \infty)$ if $a = 1$; $(0, \infty)$ otherwise	same
mean	ab	a/β
variance	ab^2	a/β^2
mode	$(a-1)b$ if $a \geq 1$	$(a-1)/\beta$ if $a \geq 1$
random number	see (McGrath and Irving 1973)	

Table 52.17 Geometric Distribution

PROC specification	**geo**(p)
density [2]	$p(1-p)^{\theta}$
parameter restriction	$0 < p \leq 1$
range	$\theta \in \begin{cases} \{0,1,2,\ldots\} & 0 < p < 1 \\ \{0\} & p = 1 \end{cases}$
mean	round$(\frac{1-p}{p})$
variance	$\frac{1-p}{p^2}$
mode	0
random number	based on samples obtained from a Bernoulli distribution with probability p until the first success

Table 52.18 Inverse χ^2 Distribution

PROC specification	**ichisq**(ν)
density	$\frac{1}{\Gamma(\nu/2)2^{\nu/2}}\theta^{-(\nu/2+1)}e^{-1/(2\theta)}$
parameter restriction	$\nu > 0$
range	$\theta \in (0,\infty)$
mean	$\frac{1}{\nu-2}$ if $\nu > 2$
variance	$\frac{2}{(\nu-2)^2(\nu-4)}$ if $\nu > 4$
mode	$\frac{1}{\nu+2}$
random number	inverse χ^2 is a special case of the inverse-gamma distribution: $\theta \sim$ igamma$(\nu/2, \text{scale} = 2)$ is a draw from the inverse χ^2 distribution

[2] The random variable θ is the total number of failures in an experiment *before* the first success. This density function is not to be confused with another popular formulation, $p(1-p)^{\theta-1}$, which counts the total number of trials *until* the first success.

Table 52.19 Inverse-Gamma Distribution

PROC specification	**igamma**(a, scale = b)	**igamma**(a, iscale = β)
density	$\frac{b^a}{\Gamma(a)}\theta^{-(a+1)}e^{-b/\theta}$	$\frac{1}{\beta^a\Gamma(a)}\theta^{-(a+1)}e^{-1/\beta\theta}$
parameter restriction	$a > 0, b > 0$	$a > 0, \beta > 0$
range	$\theta \in (0, \infty)$	same
mean	$\frac{b}{a-1}$ if $a > 1$	$\frac{1}{\beta(a-1)}$ if $a > 1$
variance	$\frac{b^2}{(a-1)^2(a-2)}$	$\frac{1}{\beta^2(a-1)^2(a-2)}$
mode	$\frac{b}{a+1}$	$\frac{1}{\beta(a+1)}$
random number	generate $x_1 \sim$ gamma(a, scale $= b$), and $\theta = 1/x_1$ is a draw from the igamma(a, iscale $= b$) distribution	
relationship to the gamma distribution	$\theta \sim$ gamma(a, scale $= b$) $\Leftrightarrow 1/\theta \sim$ igamma(a, iscale $= b$)	

Table 52.20 Laplace (Double Exponential) Distribution

PROC specification	**laplace**(a, scale = b)	**laplace**(a, iscale = β)
density	$\frac{1}{2b}e^{-\lvert\theta-a\rvert/b}$	$\frac{\beta}{2}e^{-\beta\lvert\theta-a\rvert}$
parameter restriction	$b > 0$	$\beta > 0$
range	$\theta \in (-\infty, \infty)$	same
mean	a	a
variance	$2b^2$	$2/\beta^2$
mode	a	a
random number	inverse CDF. $F(\theta) = \begin{cases} \frac{1}{2}\exp\left(-\frac{a-\theta}{b}\right) & \theta < a \\ 1 - \frac{1}{2}\exp\left(-\frac{\theta-a}{b}\right) & \theta \geq a \end{cases}$. Generate $u_1, u_2 \sim$ uniform(0, 1). If $u_1 < 0.5, \theta = a + b\log(u_2)$; else $\theta = a - b\log(u_2)$. θ is a draw from the Laplace distribution	

Table 52.21 **Logistic Distribution**

PROC specification	**logistic**(a, b)
density	$\frac{\exp\left(-\frac{\theta-a}{b}\right)}{b\left(1+\exp\left(-\frac{\theta-a}{b}\right)\right)^2}$
parameter restriction	$b > 0$
range	$\theta \in (-\infty, \infty)$
mean	a
variance	$\frac{\pi^2 b^2}{3}$
mode	a
random number	inverse CDF method with $F(\theta) = \left(1 + \exp(-\frac{\theta-a}{b})\right)^{-1}$. Generate $u \sim$ uniform$(0, 1)$, and $\theta = a - b\log(1/u - 1)$ is a draw from the logistic distribution

Table 52.22 **LogNormal Distribution**

PROC specification	**lognormal**$(\mu,$ sd = $s)$	**lognormal**$(\mu,$ var = $v)$	**lognormal**$(\mu,$ prec = $\tau)$
density	$\frac{1}{\theta s\sqrt{2\pi}} \exp\left(-\frac{(\log\theta-\mu)^2}{2s^2}\right)$	$\frac{1}{\theta\sqrt{2\pi v}} \exp\left(-\frac{(\log\theta-\mu)^2}{2v}\right)$	$\frac{1}{\theta}\sqrt{\frac{\tau}{2\pi}} \exp\left(-\frac{\tau(\log\theta-\mu)^2}{2}\right)$
parameter restriction	$s > 0$	$v > 0$	$\tau > 0$
range	$\theta \in (0, \infty)$	same	same
mean	$\exp(\mu + s^2/2)$	$\exp(\mu + v/2)$	$\exp(\mu + 1/(2\tau))$
variance	$\exp(2(\mu + s^2))$ $-\exp(2\mu + s^2)$	$\exp(2(\mu + v))$ $-\exp(2\mu + v)$	$\exp(2(\mu + 1/\tau))$ $-\exp(2\mu + 1/\tau)$
mode	$\exp(\mu - s^2)$	$\exp(\mu - v)$	$\exp(\mu - 1/\tau)$
random number	generate $x_1 \sim$ normal$(0, 1)$, and $\theta = \exp(\mu + sx_1)$ is a draw from the lognormal distribution		

Table 52.23 Negative Binomial Distribution

PROC specification	**negbin**(n, p)
density	$\binom{\theta+n-1}{\theta} p^n (1-p)^\theta$
parameter restriction	$n = 1, 2, \cdots 0 < p \leq 1$
range	$\theta \in \begin{cases} \{0,1,2,\ldots\} & 0 < p < 1 \\ \{0\} & p = 1 \end{cases}$
mean	$\text{round}\left(\frac{n(1-p)}{p}\right)$
variance	$\frac{n(1-p)}{p^2}$
mode	$\begin{cases} 0 & n = 1 \\ \text{round}\left(\frac{(n-1)(1-p)}{p}\right) & n > 1 \end{cases}$
random number	generate $x_1 \sim \text{gamma}(n, 1)$, and $\theta \sim \text{Poisson}(x_1 \cdot (1-p)/p)$ (Fishman 1996).

Table 52.24 Normal Distribution

PROC specification	**normal**(μ, sd = s)	**normal**(μ, var = v)	**normal**(μ, prec = τ)
density	$\frac{1}{s\sqrt{2\pi}} \exp\left(-\frac{(\theta-\mu)^2}{2s^2}\right)$	$\frac{1}{\sqrt{2\pi v}} \exp\left(-\frac{(\theta-\mu)^2}{2v}\right)$	$\sqrt{\frac{\tau}{2\pi}} \exp\left(-\frac{\tau(\theta-\mu)^2}{2}\right)$
parameter restriction	$s > 0$	$v > 0$	$\tau > 0$
range	$\theta \in (-\infty, \infty)$	same	same
mean	μ	same	same
variance	s^2	v	$1/\tau$
mode	μ	same	same

Table 52.25 Pareto Distribution

PROC specification	**pareto**(a, b)
density	$\frac{a}{b}\left(\frac{b}{\theta}\right)^{a+1}$
parameter restriction	$a > 0, b > 0$
range	$\theta \in [b, \infty)$
mean	$\frac{ab}{a-1}$ if $a > 1$
variance	$\frac{b^2 a}{(a-1)^2(a-2)}$ if $a > 2$
mode	b
random number	inverse CDF method with $F(\theta) = 1 - (b/\theta)^a$. Generate $u \sim$ uniform$(0, 1)$, and $\theta = \frac{b}{u^{1/a}}$ is a draw from the Pareto distribution.
useful transformation	$x = 1/\theta$ is Beta$(a, 1)\mathbf{I}\{x < 1/b\}$.

Table 52.26 Poisson Distribution

PROC specification	**poisson**(λ)
density	$\frac{\lambda^\theta}{\theta!}\exp(-\lambda)$
parameter restriction	$\lambda \geq 0$
range	$\theta \in \begin{cases} \{0, 1, \ldots\} & \text{if } \lambda > 0 \\ \{0\} & \text{if } \lambda = 0 \end{cases}$
mean	λ
variance	λ, if $\lambda > 0$
mode	round(λ)

Table 52.27 Scaled Inverse χ^2 Distribution

PROC specification	**sichisq**(ν, s^2)
density	$\frac{(\nu/2)^{\nu/2}}{\Gamma(\nu/2)} s^\nu \theta^{-(\nu/2+1)} e^{-\nu s^2/(2\theta)}$
parameter restriction	$\nu > 0, s > 0$
range	$\theta \in (0, \infty)$
mean	$\frac{\nu}{\nu-2}s^2$ if $\nu > 2$
variance	$\frac{2\nu^2}{(\nu-2)^2(\nu-4)}s^4$ if $\nu > 4$
mode	$\frac{\nu}{\nu+2}s^2$
random number	scaled inverse χ^2 is a special case of the inverse-gamma distribution: $\theta \sim$ igamma$(\nu/2, \text{scale} = (\nu s^2)/2)$ is a draw from the scaled inverse χ^2 distribution.

Table 52.28 T Distribution

PROC specification	$\mathbf{t}(\mu, \text{sd} = s, \nu)$	$\mathbf{t}(\mu, \text{var} = v, \nu)$	$\mathbf{t}(\mu, \text{prec} = \tau, \nu)$
density	$\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})s\sqrt{\nu\pi}}(1+\frac{(\theta-\mu)^2}{\nu s^2})^{-\frac{\nu+1}{2}}$	$\frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})\sqrt{\nu\pi v}}(1+\frac{(\theta-\mu)^2}{\nu v})^{-\frac{\nu+1}{2}}$	$\frac{\Gamma(\frac{\nu+1}{2})\sqrt{\tau}}{\Gamma(\frac{\nu}{2})\sqrt{\nu\pi}}(1+\frac{\tau(\theta-\mu)^2}{\nu})^{-\frac{\nu+1}{2}}$
parm restriction	$s > 0, \nu > 0$	$v > 0, \nu > 0$	$\tau > 0, \nu > 0$
range	$\theta \in (-\infty, \infty)$	same	same
mean	μ if $\nu > 1$	same	same
variance	$\frac{\nu}{\nu-2}s^2$ if $\nu > 2$	$\frac{\nu}{\nu-2}v$ if $\nu > 2$	$\frac{\nu}{\nu-2}\frac{1}{\tau}$ if $\nu > 2$
mode	μ	same	same
random number	$x_1 \sim \text{normal}(0,1), x_2 \sim \chi^2(d)$, and $\theta = m + \sigma x_1\sqrt{d/x_2}$ is a draw from the t-distribution.		

Table 52.29 Uniform Distribution

PROC specification	**uniform**(a, b)
density	$\begin{cases} \frac{1}{a-b} & \text{if } a > b \\ \frac{1}{b-a} & \text{if } b > a \\ 1 & \text{if } a = b \end{cases}$
parameter restriction	none
range	$\theta \in [a, b]$
mean	$\frac{a+b}{2}$
variance	$\frac{\lvert b-a\rvert^2}{12}$
mode	does not exist
random number	Mersenne Twister (Matsumoto and Kurita 1992, 1994; Matsumoto and Nishimura 1998)

Table 52.30 Wald Distribution

PROC specification	**wald**(μ, λ)
density	$\sqrt{\frac{\lambda}{2\pi\theta^3}} \exp\left(\frac{-\lambda(\theta-\mu)^2}{2\mu^2\theta}\right)$
parameter restriction	$\mu > 0, \lambda > 0$
range	$\theta \in (0, \infty)$
mean	μ
variance	μ^3/λ
mode	$\mu\left[\left(1+\frac{9\mu^2}{4\lambda^2}\right)^{1/2} - \frac{3\mu}{2\lambda}\right]$
random number	generate $v_0 \sim \chi^2_{(1)}$. Let $x_1 = \mu + \frac{\mu^2 v_0}{2\lambda} - \frac{\mu}{2\lambda}\sqrt{4\mu\lambda v_0 + \mu^2 v_0^2}$ and $x_2 = \mu^2/x_1$. Perform a Bernoulli trial, $w \sim$ Bernoulli($\frac{\mu}{\mu+x_1}$). If $w = 1$, choose $\theta = x_1$; otherwise, choose $\theta = x_2$ (Michael, Schucany, and Haas 1976).

Table 52.31 Weibull Distribution

PROC specification	**weibull**(μ, c, σ)
density	$\exp\left(-\left(\frac{\theta-\mu}{\sigma}\right)^c\right) \frac{c}{\sigma}\left(\frac{\theta-\mu}{\sigma}\right)^{c-1}$
parameter restriction	$c > 0, \sigma > 0$
range	$\theta \in [\mu, \infty)$ if $c = 1$; (μ, ∞) otherwise
mean	$\mu + \sigma\Gamma(1 + 1/c)$
variance	$\sigma^2[\Gamma(1 + 2/c) - \Gamma^2(1 + 1/c)]$
mode	$\mu + \sigma(1 - 1/c)^{1/c}$ if $c > 1$
random number	inverse CDF method with $F(\theta) = 1 - \exp\left(-\left(-\frac{\theta-\mu}{\sigma}\right)^c\right)$. Generate $u \sim$ uniform(0, 1), and $\theta = \mu + \sigma \cdot (-\ln u)^{1/c}$ is a draw from the Weibull distribution.

Specifying a New Distribution

To work with a new density that is not listed in the section "Standard Distributions" on page 3530, you can use the GENERAL and DGENERAL functions. The letter "D" stands for discrete. The new distributions have to be specified on the logarithm scale.

Suppose that you want to use the inverse-beta distribution:

$$p(\alpha|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)+\Gamma(b)} \cdot \alpha^{(a-1)} \cdot (1+\alpha)^{-(a+b)}$$

The following statements in PROC MCMC define the density on its log scale:

```
a = 3; b = 5;
const = lgamma(a + b) - lgamma(a) - lgamma(b);
lp = const + (a - 1) * log(alpha) - (a + b) * log(1 + alpha);
prior alpha ~ general(lp);
```

The symbol lp is the expression for the log of an inverse-beta (a = 3, b = 5). The function **general(lp)** assigns that distribution to alpha. Note that the constant term, const, can be omitted as the Markov simulation requires only the log of the density kernel.

When you use the GENERAL function in the MODEL statement, you do not need to specify the dependent variable on the left of the tilde (~). The log-likelihood function takes the dependent variable into account; hence there is no need to explicitly state the dependent variable in the MODEL statement. However, in the PRIOR statements, you need to explicitly state the parameter names and a tilde with the GENERAL and DGENERAL functions.

You can specify any distribution function by using the GENERAL and DGENERAL functions as long as they are programmable with SAS statements. When the function is used in the PRIOR statements, you must supply initial values. This can be done in either the PARMS statement ("PARMS Statement" on page 3515) or within the BEGINCNST and ENDCNST statements ("BEGINCNST/ENDCNST Statement" on page 3509).

It is important to remember that PROC MCMC does not verify that the GENERAL function you specify is a valid distribution—that is, an integrable density. You must use the function with caution.

Using Density Functions in the Programming Statements

Density Functions in PROC MCMC

PROC MCMC also has a number of internally defined log-density functions. The functions have the basic form of lpdf*dist*(*x*, *parm-list*, < *lower* >, < *upper* >), where *dist* is the name of the distribution (see Table 52.32). The argument *x* is the random variable, *parm-list* is the list of parameters, and *lower* and *upper* are boundary arguments. The *lower* and *upper* arguments are optional but positional. With the exception of the Bernoulli and uniform distribution, you can specify limits on all distributions.

To set a lower bound on the normal density:

```
lpdfnorm(x, 0, 1, -2);
```

To set just an upper bound, specify a missing value for the lower bound argument:

```
lpdfnorm(x, 0, 1, ., 2);
```

Leaving both limits out gives you the unbounded density, and you can also specify both bounds:

```
lpdfnorm(x, 0, 1);
lpdfnorm(x, 0, 1, -3, 4);
```

See the following table for a list of distributions and their corresponding lpdf functions.

Table 52.32 Logarithm of Density Functions in PROC MCMC

Distribution Name	Function Call
beta	`lpdfbeta(x, a, b, <lower>, <upper>);`
binary	`lpdfbern(x, p);`
binomial	`lpdfbin(x, n, p, <lower>, <upper>);`
Cauchy	`lpdfcau(x, loc, scale, <lower>, <upper>);`
χ^2	`lpdfchisq(x, df, <lower>, <upper>);`
exponential χ^2	`lpdfechisq(x, df, <lower>, <upper>);`
exponential gamma	`lpdfegamma(x, sp, scale, <lower>, <upper>);`
exponential exponential	`lpdfeexpon(x, scale, <lower>, <upper>);`
exponential inverse χ^2	`lpdfeichisq(x, df, <lower>, <upper>);`
exponential inverse-gamma	`lpdfeigamma(x, sp, scale, <lower>, <upper>);`
exponential scaled inverse χ^2	`lpdfesichisq(x, df, scale, <lower>, <upper>);`
exponential	`lpdfexpon(x, scale, <lower>, <upper>);`
gamma	`lpdfgamma(x, sp, scale, <lower>, <upper>);`
geometric	`lpdfgeo(x, p, <lower>, <upper>);`
inverse χ^2	`lpdfichisq(x, df, <lower>, <upper>);`
inverse-gamma	`lpdfigamma(x, sp, scale, <lower>, <upper>);`
Laplace	`lpdfdexp(x, loc, scale, <lower>, <upper>);`
logistic	`lpdflogis(x, loc, scale, <lower>, <upper>);`
lognormal	`lpdflnorm(x, loc, sd, <lower>, <upper>);`
negative binomial	`lpdfnegbin(x, n, p, <lower>, <upper>);`
normal	`lpdfnorm(x, mu, sd, <lower>, <upper>);`
Pareto	`lpdfpareto(x, sp, scale, <lower>, <upper>);`
Poisson	`lpdfpoi(x, mean, <lower>, <upper>);`
scaled inverse χ^2	`lpdfsichisq(x, df, scale, <lower>, <upper>);`
T	`lpdft(x, mu, sd, df, <lower>, <upper>);`
uniform	`lpdfunif(x, a, b);`
Wald	`lpdfwald(x, mean, scale, <lower>, <upper>);`
Weibull	`lpdfwei(x, loc, sp, scale, <lower>, <upper>);`

Standard Distributions, the logpdf Functions, and the lpdf*dist* Functions

Standard distributions listed in the section "Standard Distributions" on page 3530 are *names* only, and they can only be used in the MODEL, PRIOR, and HYPERPRIOR statements to specify either a prior distribution or a conditional distribution of the data given parameters. They do not return

any values, and you cannot use them in the programming statements.

The LOGPDF functions are DATA step functions that compute the logarithm of various probability density (mass) functions. For example, **logpdf("beta", x, 2, 15)** returns the log of a beta density with parameters a = 2 and b = 15, evaluated at x. All the LOGPDF functions are supported in PROC MCMC.

The lpdf*dist* functions are unique to PROC MCMC. They compute the logarithm of various probability density (mass) functions. The functions are the same as the LOGPDF functions when it comes to calculating the log density. For example, **lpdfbeta(x, 2, 15)** returns the same value as **logpdf("beta", x, 2, 15)**. The lpdf*dist* functions cover a greater class of probability density functions, and they take the optional but positional boundary arguments. There are no corresponding **lcdf***dist* or **lsdf***dist* functions in PROC MCMC. To work with the cumulative probability function or the survival functions, you need to use the LOGCDF and the LOGSDF DATA step functions.

Truncation and Censoring

Truncated Distributions

To specify a truncated distribution, you can use the LOWER= and/or UPPER= options. Almost all of the standard distributions, including the GENERAL and DGENERALfunctions, take these optional truncation arguments. The exceptions are the binary and uniform distributions.

For example, you can specify the following:

```
prior alpha ~ normal(mean = 0, sd = 1, lower = 3, upper = 45);
```

or

```
parms beta;
a = 3; b = 7;
ll = (a + 1) * log(b / beta);
prior beta ~ general(ll, upper = b + 17);
```

The preceding statements state that if beta is less than b+17, the log of the prior density is ll, as calculated by the equation; otherwise, the log of the prior density is missing—the log of zero.

When the same distribution is applied to multiple parameters in a PRIOR statement, the LOWER= and UPPER= truncations apply to all parameters in that statement. For example, the following statements define a Poisson density for theta and gamma:

```
parms theta gamma;
lambda = 7;
l1 = theta * log(lambda) - lgamma(1 + theta);
l2 = gamma * log(lambda) - lgamma(1 + gamma);
ll = l1 + l2;
prior theta gamma ~ dgeneral(ll, lower = 1);
```

The LOWER=1 condition is applied to both theta and gamma, meaning that for the assignment to ll to be meaningful, both theta and gamma have to be greater than 1. If either of the parameters is less than 1, the log of the joint prior density becomes a missing value.

With the exceptions of the normal distribution and the GENERAL and DGENERAL functions, the LOWER= and UPPER= options cannot be parameters or functions of parameters. The reason is that most of the truncated distributions are not normalized. Unnormalized densities do not lead to wrong MCMC answers as long as the bounds are constants. However if the bounds involve model parameters, then the normalizing constant, which is a function of these parameters, must be taken into account in the posterior. Without specifying the normalizing constant, inferences on these boundary parameters are incorrect.

It is not difficult to construct a truncated distribution with a normalizing constant. Any truncated distribution has the probability distribution:

$$p(\theta|a < \theta < b) = \frac{p(\theta)}{F(a) - F(b)}$$

where $p(\cdot)$ is the density function and $F(\cdot)$ is the cumulative distribution function. In SAS functions, $p(\cdot)$ is probability density function and $F(\cdot)$ is cumulative distribution function. The following example shows how to construct a truncated gamma prior on theta, with SHAPE = 3, SCALE = 2, LOWER = a, and UPPER = b:

```
lp = logpdf('gamma', theta, 3, 2)
       - log(cdf('gamma', a, 3, 2) - cdf('gamma', b, 3, 2));
prior theta ~ general(lp);
```

Note the difference from a naive definition of the density, without taking into account of the normalizing constant:

```
lp = logpdf('gamma', theta, 3, 2);
prior theta ~ general(lp, lower=a, upper=b);
```

If a or b are parameters, you get very different results from the two formulations.

Censoring

There is no built-in mechanism in PROC MCMC that models censoring automatically. You need to construct the density function (using a combination of the LOGPDF, LOGCDF, and LOGSDF functions and IF-ELSE statements) for the censored data.

Suppose that you partition the data into four categories: uncensored (with observation x), left censored (with observation xl), right censored (with observation xr), and interval censored (with observations xl and xr). The likelihood is the normal with mean mu and standard deviation s. The following statements construct the corresponding log likelihood for the observed data:

```
if uncensored then
   ll = logpdf('normal', x, mu, s);
else if leftcensored then
   ll = logcdf('normal', xl, mu, s);
else if rightcensored then
   ll = logsdf('normal', xr, mu, s);
else /* this is the case of interval censored. */
```

```
   ll = log(cdf('normal', xr, mu, s) - cdf('normal', xl, mu, s));
model general(ll);
```

See "Example 52.9: Normal Regression with Interval Censoring" on page 3664.

Multivariate Density Functions

The DATA step has functions that compute the logarithm of the density of some multivariate distributions. You can use them in PROC MCMC. For a complete listing of multivariate functions, see *SAS Language Reference: Dictionary*.

Some commonly used multivariate functions in Bayesian analysis are as follows:

- LOGMPDFNORMAL, the logarithm of the multivariate normal
- LOGMPDFWISHART, the logarithm of the Wishart
- LOGMPDFIWISHART, the logarithm of the inverted-Wishart
- LOGMPDFDIR1, the logarithm of the Dirichlet distribution of Type I
- LOGMPDFDIR2, the logarithm of the Dirichlet distribution of Type II
- LOGMPDFMULTINOM, the logarithm of the multinomial

Other multivariate density functions include: LOGMPDFT (t-distribution), LOGMPDFGAMMA (gamma distribution), LOGMPDFBETA1 (beta of type I), and LOGMPDFBETA2 (beta of type II).

Density Function Definition

LOGMPDFNORMAL

Let x be an n-dimensional random vector with mean vector μ and covariance matrix $\mathbf{\Sigma}$. The density is

$$pdf(x;\mu,\mathbf{\Sigma}) = \frac{\exp(-\frac{1}{2}(x-\mu)^T\mathbf{\Sigma}^{-1}(x-\mu))}{\sqrt{(2\pi)^n\,|\mathbf{\Sigma}|}}$$

where $|\mathbf{\Sigma}|$ is the determinant of the covariance matrix $\mathbf{\Sigma}$.

The function has syntax:

$$y = \text{LOGMPDFNORMAL}(x\_list, \mu\_list, cov\_name);$$

WARNING: you must set up the cov_name covariance matrix before using the LOGMPDFNORMAL function and free the memory after PROC MCMC exits. See the section "Set Up the Covariance Matrices and Free Memory" on page 3548.

LOGMPDFWISHART and LOGMPDFIWISHART

The density function from the Wishart distribution is:

$$pdf(x;\mu,\mathbf{\Sigma}) = \frac{1}{C_n(\mu)}\,|\mathbf{\Sigma}|^{-\frac{\mu}{2}}\,|x|^{\frac{\mu-n-1}{2}}\exp\left(-\frac{1}{2}tr(\mathbf{\Sigma}^{-1}x)\right)$$

with $\mu > n$, and the trace of a square matrix A is given by:

$$tr(A) = \sum_i a_{ii} \quad C_n(\mu) = 2^{\frac{\mu n}{2}}\Gamma_n\left(\frac{\mu}{2}\right) \quad \Gamma_n(z) = \pi^{\frac{n(n-1)}{4}}\prod_{i=1}^{n}\Gamma\left(z-\frac{i-1}{2}\right)$$

The density function from the inverse-Wishart distribution is:

$$pdf(x;\mu,\mathbf{\Sigma}) = \frac{1}{D_n(\mu)}\,|\mathbf{\Sigma}|^{\frac{\mu-n-1}{2}}\,|x|^{-\frac{\mu}{2}}\exp\left(-\frac{1}{2}tr(\mathbf{\Sigma}x^{-1})\right)$$

for $\mu > 2n$, and

$$D_n(\mu) = 2^{\frac{(\mu-n-1)n}{2}}\Gamma_n\left(\frac{\mu-n-1}{2}\right)$$

If $V \sim IW_n(\mu,\mathbf{\Sigma})$ then $V^{-1} \sim W_n(\mu-n-1,\mathbf{\Sigma}^{-1})$

The functions have syntax:

$$y = \text{LOGMPDFWISHART}('name'_V,\mu,'name'_{\mathbf{\Sigma}});$$

and for the inverted Wishart:

$$y = \text{LOGMPDFIWISHART}('name'_V,\mu,'name'_{\mathbf{\Sigma}});$$

The three arguments are the multivariate matrix $'name'_V$, the degrees of freedom μ, and the covariance matrix $'name'_{\mathbf{\Sigma}}$k

WARNING: you must set up the *cov_name* covariance matrix before using these functions and free the memory after PROC MCMC exits. See the section "Set Up the Covariance Matrices and Free Memory" on page 3548.

LOGMPDFDIR1 and LOGMPDFDIR2

The random variables $u_1 ... u_k$, with $u_i > 0$ and $\sum_{i=1}^{k} u_i < 1$, are said to have a Dirichlet Type I distribution with parameters $a_1 ... a_{k+1}$ if their joint pdf is given by:

$$pdf_1(u_1,u_2,...,u_k,a_1,a_2,...,a_{k+1}) = \frac{\Gamma(\sum_{i=1}^{k+1}a_i)}{\prod_{i=1}^{r+1}\Gamma(a_i)}\left(\prod_{i=1}^{k}u_i^{a_i-1}\right)\left(1-\sum_{i=1}^{k}u_i\right)^{a_{k+1}-1}$$

The variables are said to have a Dirichlet type II distribution with parameters $a_1 ... a_{k+1}$ if their joint pdf is given by the following:

$$pdf_2(u_1,u_2,...,u_k,a_1,a_2,...,a_{k+1}) = \frac{\Gamma(\sum_{i=1}^{k+1}a_i)}{\prod_{i=1}^{r+1}\Gamma(a_i)}\left(\prod_{i=1}^{k}u_i^{a_i-1}\right)\left(1+\sum_{i=1}^{k}u_i\right)^{-\sum_{i=1}^{k+1}a_i}$$

The functions have syntax:

$$y = \text{LOGMPDFDIR1}(u\_list, a\_list);$$

and

$$y = \text{LOGMPDFDIR2}(u\_list, a\_list);$$

LOGMPDFMULTINOM

Let $n_1 ... n_k$ be random variables that denote the number of occurring of the events $E_1, ... E_k$ respectively occurring with probabilities $p_1 ... p_k$. Let $\sum_{i=1}^{k} p_i = 1$ and let $n = \sum_{i=1}^{k} n_i$. Then the joint distribution of $n_1, ...n_k$ is the following:

$$pdf(n_1, n_2, ...n_k, p_1, p_2, ..., p_k) = n! \prod_{i=1}^{k} \left(\frac{p_i^{n_i}}{n_i!} \right)$$

The function has syntax:

$$y = \text{LOGMPDFMULTINOM}(n\_list, p\_list);$$

Set Up the Covariance Matrices and Free Memory

For distributions that require symmetric positive definite matrices, such as the LOGMPDFNORMAL, LOGMPDFWISHART and LOGMPDFIWISHART functions, you need to set up these matrices by using the following functions:

- Use LOGMPDFSETSQ to set up a symmetric positive definite matrix from all its elements:

  $$rc = \text{LOGMPDFSETSQ}(name, num1, num2, ...);$$

  rc is set to 0 when the numeric arguments describe a symmetric positive definite matrix, otherwise it is set to a nonzero value.

- Use LOGMPDFSET to set up a symmetric positive definite matrix from its lower triangular elements:

  $$rc = \text{LOGMPDFSET}(name, num1, num2, ...);$$

  When the numeric arguments describe a symmetric positive definite matrix, the returned value rc is set to 0. Otherwise, a nonzero value for rc is returned.

- Use LOGMPFFREE to free the workspace previously allocated with either LOGMPDFSET or LOGMPDFSETSQ:

  $$rc = \text{LOGMPDFFREE}(< ... < 'name' >, 'name2' > ...);$$

  When called without arguments, the LOGMPDFFREE frees all the symbols previously allocated by LOGMPDFSETSQ or LOGMPDFSET. Each freed symbol is reported back in the SAS log.

The parameters used in these functions are defined as follows:

name is a string containing the name of the work space that stores the matrix by the numeric parameters $num1, ...$

$num1, ...$ are numeric arguments that represent the elements of a symmetric positive definite matrix.

You would set up this matrix under the DATA step by using the following syntax:

$$rc = \text{LOGMPDFSETSQ}(name, \sigma_{11}, \sigma_{12}, \sigma_{21}, \sigma_{22});$$

or the syntax:

$$rc = \text{LOGMPDFSET}(name, \sigma_{11}, \sigma_{21}, \sigma_{22});$$

If the matrix is positive definite, the returned value rc is zero.

Some Useful SAS Functions

Table 52.33 Some Useful SAS Functions

SAS Function	Definition
`abs(x)`	$\|x\|$
`airy(x)`	returns the value of the AIRY function.
`beta(x1, x2)`	$\int_0^1 z^{x1-1}(1-z)^{x2-1}dz$
`call logistic(x)`	$\ln(x/(1-x))$
`call softmax(x1,...,xn)`	each element is replaced by $\exp(x_j)/\sum \exp(x_j)$
`call stdize(x1,...,xn)`	standardize values
`cdf`	cumulative distribution function
`cdf('normal', x, 0, 1)`	standard normal cumulative distribution function
`comb(x1, x2)`	$\frac{x1!}{x2!(x1-x2)!}$
`constant('.')`	calculate commonly used constants
`cos(x)`	cosine(x)
`css(x1, ..., xn)`	$\sum_i (x_i - \bar{x})^2$
`cv(x1, ..., xn)`	std(x) / mean(x) * 100
`dairy(x)`	derivative of the AIRY function
`dimN(m)`	returns the numbers of elements in the Nth dim of array *m*
`(x1 eq x2)`	returns 1 if x1 = x2; 0 otherwise
`x1**x2`	$x1^{x2}$
`geomean(x1, ..., xn)`	$\exp\left(\frac{\log(x_1)+\cdots+\log(x_n)}{n}\right)$
`difN(x)`	returns differences between the argument and its Nth lag
`digamma(x1)`	$\frac{\Gamma'(x1)}{\Gamma(x1)}$
`erf(x)`	$\frac{2}{\sqrt{\pi}}\int_0^x exp(-z^2)dz$

Table 52.33 (continued)

SAS Function	Definition
`erfc(x)`	1 - erf(x)
`fact(x)`	$x!$
`floor(x)`	greatest integer $\leq x$
`gamma(x)`	$\int_0^\infty z^{x-1} \exp(-1) dz$
`harmean(x1, ..., xn)`	$\frac{n}{1/x_1 + \cdots 1/x_n}$
`ibessel(nu, x, kode)`	modified Bessel function of order *nu* evaluated at x
`jbessel(nu, x)`	Bessel function of order *nu* evaluated at x
`lagN(x)`	returns values from a queue
`largest(k, x1, ..., xn)`	the kth largest element
`lgamma(x)`	$\ln(\Gamma(x))$
`lgamma(x+1)`	$\ln(x!)$
`log(x), logN(x)`	$\ln(x)$
`logbeta(x1, x2)`	lgamma(x_1) + lgamma(x_2) - lgamma($x_1 + x_2$)
`logcdf`	log of a left cumulative distribution function
`logpdf`	log of a probability density (mass) function
`logsdf`	log of a survival function
`max(x1, x2)`	returns x_1 if $x_1 > x_2$; x_2 otherwise
`mean(of x1-xn)`	$\sum_i x_i / n$
`median(of x1-xn)`	returns the median of nonmissing values
`min(x1, x2)`	returns x_1 if $x_1 < x_2$; x_2 otherwise
`missing(x)`	returns 1 if x is missing; 0 otherwise
`mod(x1, x2)`	returns the remainder from x_1/x_2
`n(x1, ..., xn)`	returns number of nonmissing values
`nmiss(of y1-yn)`	number of missing values
`quantile`	computes the quantile from a specific distribution
`pdf`	probability density (mass) functions
`perm(n, r)`	$\frac{n!}{(n-r)!}$
`put`	returns a value that uses a specified format
`round(x)`	rounds x
`rms(of x1-xn)`	$\sqrt{\frac{x_1^2 + \cdots x_n^2}{n}}$
`sdf`	survival function
`sign(x)`	returns -1 if $x < 0$; 0 if $x = 0$; 1 if $x > 0$
`sin(x)`	sine(x)
`smallest(s, x1, ..., en )`	the sth smallest component of $x_1, \cdots, x_n$
`sortn(of x1-xn)`	sorts the values of the variables
`sqrt(x)`	$\sqrt{x}$
`std(x1, ..., xn)`	standard deviation of $x_1, \cdots, x_n$ (n-1 in denominator)
`sum(of x:)`	$\sum_i x_i$
`trigamma(x)`	derivative of the DIGAMMA(x) function
`uss(of x1-xn)`	uncorrected sum of squares

Here are examples of some commonly used transformations:

- logit

```
mu = beta0 + beta1 * z1;
call logistic(mu);
```

- log

```
w = beta0 + beta1 * z1;
mu = exp(w);
```

- probit

```
w = beta0 + beta1 * z1;
mu = cdf('normal', w, 0, 1);
```

- cloglog

```
w = beta0 + beta1 * z1;
mu = 1 - exp(-exp(w));
```

Matrix Functions in PROC MCMC

The MCMC procedure provides you with a number of CALL routines for performing simple matrix operations on declared arrays. With the exception of FILLMATRIX, IDENTITY, and ZEROMATRIX, the CALL routines listed in Table 52.34 do not support matrices or arrays that contain missing values.

Table 52.34 Matrix Functions in PROC MCMC

CALL Routine	Description
ADDMATRIX	Performs an element-wise addition of two matrices or of a matrix and a scalar.
CHOL	Calculates the Cholesky decomposition for a particular symmetric matrix.
DET	Calculates the determinant of a specified matrix, which must be square.
ELEMMULT	Performs an element-wise multiplication of two matrices.
FILLMATRIX	Replaces all of the element values of the input matrix with the specified value. You can use this routine with multidimensional numeric arrays.
IDENTITY	Converts the input matrix to an identity matrix. Diagonal element values of the matrix are set to 1, and the rest of the values are set to 0.
INV	Calculates a matrix that is the inverse of the input matrix. The input matrix must be a square, nonsingular matrix.
MULT	Calculates the matrix product of two input matrices.
SUBTRACTMATRIX	Performs an element-wide subtraction of two matrices or of a matrix and a scalar.
TRANSPOSE	Returns the transpose of a matrix.
ZEROMATRIX	Replaces all of the element values of the numeric input matrix with 0.

ADDMATRIX CALL Routine

The ADDMATRIX CALL routine performs an element-wise addition of two matrices or of a matrix and a scalar.

The syntax of the ADDMATRIX CALL routine is

CALL ADDMATRIX *(X, Y, Z)* **;**

where

X specifies a scalar or an input matrix with dimensions $m \times n$ (that is, $X[m, n]$)

Y specifies a scalar or an input matrix with dimensions $m \times n$ (that is, $Y[m, n]$)

Z specifies an output matrix with dimensions $m \times n$ (that is, $Z[m, n]$)

such that

$$Z = X + Y$$

CHOL CALL Routine

The CHOL CALL routine calculates the Cholesky decomposition for a particular symmetric matrix.

The syntax of the CHOL CALL routine is

CALL CHOL *(X, Y <, validate>)* **;**

where

X specifies a symmetric positive-definite input matrix with dimensions $m \times m$ (that is, X[m, m])

Y is a variable that contains the Cholesky decomposition and specifies an output matrix with dimensions $m \times m$ (that is, $Y[m, m]$)

validate specifies an optional argument that can increase the processing speed by avoiding error checking:

If *validate* = 0 or is not specified, then the matrix *X* is checked for symmetry.
If *validate* = 1, then the matrix *X* is assumed to be symmetric.

such that

$$X = YY^*$$

where *Y* is a lower triangular matrix with strictly positive diagonal entries and Y^* denotes the conjugate transpose of *Y*.

Both input and output matrices must be square and have the same dimensions. If *X* is symmetric positive-definite, *Y* is a lower triangle matrix. If *X* is not symmetric positive-definite, *Y* is filled with missing values.

DET CALL Routine

The determinant, the product of the eigenvalues, is a single numeric value. If the determinant of a matrix is zero, then that matrix is singular (that is, it does not have an inverse). The routine performs an LU decomposition and collects the product of the diagonals.

The syntax of the DET CALL routine is

CALL DET *(X, a)* **;**

where

X specifies an input matrix with dimensions $m \times m$ (that is, $X[m, m]$)

a specifies the returned determinate value

such that

$a = |X|$

ELEMMULT CALL Routine

The ELEMMULT CALL routine performs an element-wise multiplication of two matrices.

The syntax of the ELEMMULT CALL routine is

CALL ELEMMULT *(X, Y, Z)* **;**

where

X specifies an input matrix with dimensions $m \times n$ (that is, $X[m, n]$)

Y specifies an input matrix with dimensions $m \times n$ (that is, $Y[m, n]$)

Z specifies an output matrix with dimensions $m \times n$ (that is, $Z[m, n]$)

FILLMATRIX CALL Routine

The FILLMATRIX CALL routine replaces all of the element values of the input matrix with the specified value. You can use the FILLMATRIX CALL routine with multidimensional numeric arrays.

The syntax of the FILLMATRIX CALL routine is

CALL FILLMATRIX *(X, Y)* **;**

where

X specifies an input numeric matrix

Y specifies the numeric value that is used to fill the matrix

IDENTITY CALL Routine

The IDENTITY CALL routine converts the input matrix to an identity matrix. Diagonal element values of the matrix are set to 1, and the rest of the values are set to 0.

The syntax of the IDENTITY CALL routine is

CALL IDENTITY *(X)* **;**

where

X specifies an input matrix with dimensions $m \times m$ (that is, $X[m, m]$)

INV CALL Routine

The INV CALL routine calculates a matrix that is the inverse of the input matrix. The input matrix must be a square, nonsingular matrix.

The syntax of the INV CALL routine is

CALL INV *(X, Y)* **;**

where

X specifies an input matrix with dimensions $m \times m$ (that is, $X[m, m]$)

Y specifies an output matrix with dimensions $m \times m$ (that is, $Y[m, m]$)

MULT CALL Routine

The MULT CALL routine calculates the matrix product of two input matrices.

The syntax of the MULT CALL routine is

CALL MULT *(X, Y, Z)* **;**

where

X specifies an input matrix with dimensions $m \times n$ (that is, $X[m, n]$)

Y specifies an input matrix with dimensions $n \times p$ (that is, $Y[n, p]$)

Z specifies an output matrix with dimensions $m \times p$ (that is, $Z[m, p]$)

The number of columns for the first input matrix must be the same as the number of rows for the second matrix. The calculated matrix is the last argument.

SUBTRACTMATRIX CALL Routine

The SUBTRACTMATRIX CALL routine performs an element-wide subtraction of two matrices or of a matrix and a scalar.

The syntax of the SUBTRACTMATRIX CALL routine is

CALL SUBTRACTMATRIX *(X, Y, Z)* **;**

where

X specifies a scalar or an input matrix with dimensions $m \times n$ (that is, $X[m, n]$)

Y specifies a scalar or an input matrix with dimensions $m \times n$ (that is, $Y[m, n]$)

Z specifies an output matrix with dimensions $m \times n$ (that is, $Z[m, n]$)

such that

$$Z = X - Y$$

TRANSPOSE CALL Routine

The TRANSPOSE CALL routine returns the transpose of a matrix.

The syntax of the TRANSPOSE CALL routine is

CALL TRANSPOSE *(X, Y)* **;**

where

X specifies an input matrix with dimensions $m \times n$ (that is, $X[m, n]$)

Y specifies an output matrix with dimensions $n \times m$ (that is, $Y[n, m]$)

ZEROMATRIX CALL Routine

The ZEROMATRIX CALL routine replaces all of the element values of the numeric input matrix with 0. You can use the ZEROMATRIX CALL routine with multidimensional numeric arrays.

The syntax of the ZEROMATRIX CALL routine is

CALL ZEROMATRIX *(X)* **;**

where

X specifies a numeric input matrix.

Modeling Joint Likelihood

PROC MCMC assumes that the input observations are independent and that the joint log likelihood is the sum of individual log-likelihood functions. You specify the log likelihood of one observation in the MODEL statement. PROC MCMC evaluates that function for each observation in the data set and cumulatively sums them up. If observations are not independent of each other, this summation produces the incorrect log likelihood.

There are two ways to model dependent data. You can either use the DATA step LAG function or use the PROC option JOINTMODEL. The LAG function returns values of a variable from a queue. As PROC MCMC steps through the data set, the LAG function queues each data set variable, and you have access to the current value as well as to all previous values of any variable. If the log likelihood for observation x_i depends only on observations 1 to i in the data set, you can use this SAS function to construct the log-likelihood function for each observation. Note that the LAG function enables you to access observations from different rows, but the log-likelihood function in the MODEL statement must be generic enough that it applies to all observations. See "Example 52.8: Cox Models" on page 3647 for how to use this LAG function.

A second option is to create arrays, store all relevant variables in the arrays, and construct the joint log likelihood for the entire data set instead of for each observation. Following is a simple example that illustrates the usage of this option. For a more realistic example that models dependent data, see "Example 52.8: Cox Models" on page 3647.

```
/* allocate the sample size. */
data exi;
   call streaminit(17);
   do ind = 1 to 100;
      y = rand("normal", 2.3, 1);
      output;
   end;
run;
```

The log-likelihood function for each observation is as follows:

$$\log(f(y_i|\mu,\sigma)) = \log(\phi(y_i;\mu,\text{var}=\sigma^2))$$

The joint log-likelihood function is as follows:

$$\log(f(\mathbf{y}|\mu,\sigma)) = \sum_i \log(\phi(y_i;\mu,\text{var}=\sigma^2))$$

The following statements fit a simple model with an unknown mean (mu) in PROC MCMC, with the variance in the likelihood assumed known. The MODEL statement indicates a normal likelihood for each observation y.

```
proc mcmc data=exi seed=7 outpost=p1;
   parm mu;
   prior mu ~ normal(0, sd=10);
   model y ~ normal(mu, sd=1);
run;
```

The following statements show how you can specify the log-likelihood function for the entire data set:

```
data a;
run;

proc mcmc data=a seed=7 outpost=p2 jointmodel;
   array data[1] / nosymbols;
   begincnst;
      rc = read_array("exi", data, "y");
      n = dim(data, 1);
   endcnst;

   parm mu;
   prior mu ~ normal(0, sd=10);
   ll = 0;
   do i = 1 to n;
      ll = ll + lpdfnorm(data[i], mu, 1);
   end;
   model general(ll);
run;
```

The JOINTMODEL option indicates that the function used in the MODEL statement calculates the log likelihood for the entire data set, rather than just for one observation. Given this option, the procedure no longer steps through the input data during the simulation. Consequently, you can no longer use any data set variables to construct the log-likelihood function. Instead, you store the data set in arrays and use arrays instead of data set variables to calculate the log likelihood.

The ARRAY statement allocates a temporary array (data). The READ_ARRAY function selects the y variable from the exi data set and stores it in the data array. See the section "READ_ARRAY Function" on page 3510. In the programming statements, you use a DO loop to construct the joint log likelihood. The expression ll in the GENERAL function now takes the value of the joint log likelihood for all data.

You can run the following statements to see that two PROC MCMC runs produce identical results.

```
proc compare data=p1 compare=p2;
   var mu;
run;
```

Regenerating Diagnostics Plots

By default, PROC MCMC generates three plots: the trace plot, the autocorrelation plot and the kernel density plot. Unless you had requested the display of ODS Graphics (ODS GRAPHICS ON) before calling the procedure, it is hard to generate the same graph afterwards. Directly using the template (Stat.MCMC.Graphics.TraceAutocorrDensity) is not feasible. To regenerate the same graph with a Markov chain, you need to define a template and use PROC SGRENDER to create the graph. See the SGRENDER procedure in the *SAS/GRAPH: Statistical Graphics Procedures Guide*. The following PROC TEMPLATE (see Chapter 21, "Statistical Graphics Using ODS") statements

define a new graph template **mygraphs.mcmc**:

```
proc template;
   define statgraph mygraphs.mcmc;
   dynamic _sim _parm;
   BeginGraph;
      layout gridded /rows=2 columns=1 rowgutter=5;
         seriesplot x=_sim y=_parm;
         layout gridded /rows=1 columns=2 columngutter=15;
            layout overlay /
               yaxisopts=(linearopts=(viewmin=-1 viewmax=1
                                      tickvaluelist=(-1 -0.5 0 0.5 1))
                          label="Autocorrelation")
               xaxisopts=(linearopts=(integer=true)
                          label="Lag" offsetmin=.015);
               needleplot x=eval(lags(_parm,Max=50))
                          y=eval(acf(_parm, NLags=50));
            endlayout;

            layout overlay / xaxisopts=(label=_parm)
                             yaxisopts=(label="Density");
               densityplot _parm /kernel();
            endlayout;
         endlayout;
      endlayout;
   EndGraph;
end;
```

The DEFINE STATGRAPH statement tells PROC TEMPLATE that you are defining a new graph template (instead of a table or style template). The template is named **mygraphs.mcmc**. There are two dynamic variables: _sim and _parm. The variable _sim is the iteration number and the variable _parm is the variable in the data set that stores the posterior sample. All STATGRAPH template definitions must start with a BEGINGRAPH statement and conclude with a ENDGRAPH statement. The first LAYOUT GRIDDED statement assembles the results of nested STATGRAPH statements into a grid, with two rows and 1 column. The trace plot (SERIESPLOT) is shown in the first row of the graph. The second LAYOUT GRIDDED statement divides the second row of the graph into two graphs: one an autocorrelation plot (NEEDLEPLOT) and the other a kernel density plot (DENSITYPLOT). For details of other controls, such as the labels, line types, see Chapter 21, "Statistical Graphics Using ODS."

A simple regression example, with three parameters, is used here. For an explanation of the regression model and the data involved, see "Simple Linear Regression" on page 3480. The following statements generate a SAS data set and fit a regression model:

```
title 'Simple Linear Regression';

data Class;
   input Name $ Height Weight @@;
   datalines;
Alfred  69.0 112.5   Alice  56.5  84.0   Barbara 65.3  98.0
Carol   62.8 102.5   Henry  63.5 102.5   James   57.3  83.0
Jane    59.8  84.5   Janet  62.5 112.5   Jeffrey 62.5  84.0
```

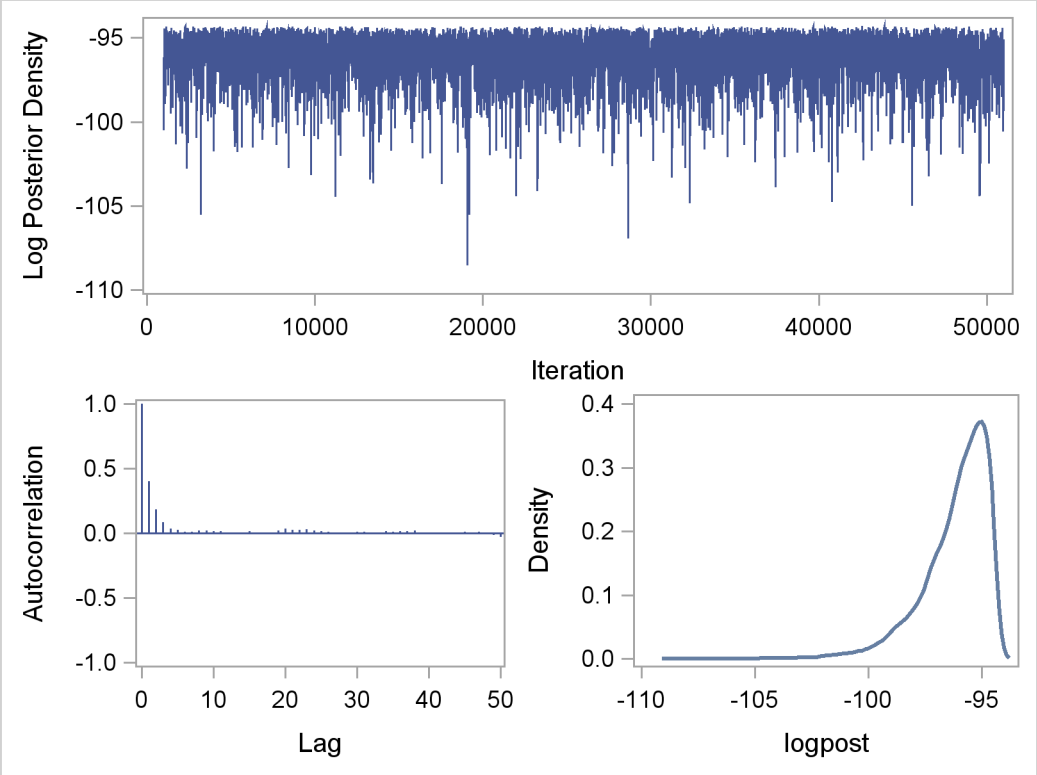
```
John    59.0  99.5   Joyce  51.3  50.5   Judy    64.3  90.0
Louise  56.3  77.0   Mary   66.5 112.0   Philip  72.0 150.0
Robert  64.8 128.0   Ronald 67.0 133.0   Thomas  57.5  85.0
William 66.5 112.0
;

proc mcmc data=class nmc=50000 thin=5 outpost=classout seed=246810;
   ods select none;
   parms beta0 0 beta1 0;
   parms sigma2 1;
   prior beta0 beta1 ~ normal(0, var = 1e6);
   prior sigma2 ~ igamma(3/10, scale = 10/3);
   mu = beta0 + beta1*height;
   model weight ~ normal(mu, var = sigma2);
run;
ods select all;
```

The output data set classout contains iteration number (Iteration) and posterior draws for beta0, beta1, and sigma2. It also stores the log of the prior density (LogPrior), log of the likelihood (LogLike), and the log of the posterior density (LogPost). If you want to examine the LogPost variable, you can use the following statements to generate the graphs:

```
proc sgrender data=classout template=mygraphs.mcmc;
   dynamic _sim='iteration' _parm='logpost';
run;
```

The SGRENDER procedure takes the classout data set and applies the template MYGRAPHS.MCMC that was defined previously. The DYNAMIC statement needs two arguments, iteration and logpost. The resulting graph is shown in Output 52.11.

Figure 52.11 Regenerate Diagnostics Plots for Log of the Posterior Density



Posterior Predictive Distribution

The posterior predictive distribution is the distribution of unobserved observations (prediction) conditional on the observed data. Let $\mathbf{y}$ be the observed data, θ be the parameter, and $\mathbf{y}_{\text{pred}}$ be the unobserved data; the posterior predictive distribution is defined to be the following:

$$\begin{aligned} p(\mathbf{y}_{\text{pred}}|\mathbf{y}) &= \int p(\mathbf{y}_{\text{pred}}, \theta|\mathbf{y}) d\theta \\ &= \int p(\mathbf{y}_{\text{pred}}|\theta, \mathbf{y}) p(\theta|\mathbf{y}) d\theta \end{aligned}$$

Given the assumption that the observed and unobserved data are conditional independent given θ, the posterior predictive distribution can be further simplified as the following:

$$p(\mathbf{y}_{\text{pred}}|\mathbf{y}) = \int p(\mathbf{y}_{\text{pred}}|\theta) p(\theta|\mathbf{y}) d\theta$$

The posterior predictive distribution is an integral of the likelihood function $p(\mathbf{y}_{\text{pred}}|\theta)$ with respect to the posterior distribution $p(\theta|\mathbf{y})$. You can use PROC MCMC to generate samples from a posterior predictive distribution based on draws from the posterior distribution of θ.

Note that the posterior predictive distribution is not the same as the prior predictive distribution. The prior predictive distribution is $p(\mathbf{y})$, which is also known as the marginal distribution of the

data. The prior predictive distribution is an integral of the likelihood function with respect to the prior distribution:

$$p(\mathbf{y}_{\text{pred}}) = \int p(\mathbf{y}_{\text{pred}}|\theta)p(\theta)d\theta$$

and the distribution is not conditional on observed data.

You can use the posterior predictive distribution to check whether the model is consistent with data. For more information about using predictive distribution as a model checking tool, see Gelman et al. 2004, Chapter 6 and the bibliography in that chapter. The idea is to generate replicate data from $p(\mathbf{y}_{\text{pred}}|\mathbf{y})$—call them $\mathbf{y}^i_{\text{pred}}$, for $i = 1, \cdots, M$, where M is the total number of replicates—compare them to the observed data, and see if there are any large and systematic differences. Large discrepancies suggest possible model misfit. One way to compare the replicate data to the observed data is to first summarize the data to some test quantities, such as the mean, standard deviation, order statistics, and so on. Then compute the tail-area probabilities of the test statistics (based on the observed data) with respect to the estimated posterior predictive distribution using the M replicate $\mathbf{y}_{\text{pred}}$ samples.

Let $T(\cdot)$ denote the function of the test quantity, $T(\mathbf{y})$ the test quantity using the observed data, and $T(\mathbf{y}^i_{\text{pred}})$ the test quantity using the ith replicate data from the posterior predictive distribution. You calculate the tail-area probability using the following formula:

$$\Pr(T(\boldsymbol{y}_{\text{pred}}) > T(\boldsymbol{y})|\theta)$$

The following example shows how you can estimate this probability using PROC MCMC.

An Example for Posterior Predictive Distribution

This example uses a normal mixed model to analyze the effects of coaching programs for the scholastic aptitude test (SAT) in eight high schools. For the original analysis of the data, see Rubin (1981). The presentation here follows the analysis and posterior predictive check presented in Gelman et al. (2004). The data are as follows:

```
title 'An Example for Posterior Predictive Distribution';

data SAT;
   input effect se @@;
   ind=_n_;
   datalines;
28.39 14.9  7.94 10.2 -2.75 16.3
 6.82 11.0 -0.64  9.4  0.63 11.4
18.01 10.4 12.16 17.6
;
```

The variable effect is the reported test score difference between coached and uncoached students in eight schools. The variable se is the corresponding estimated standard error for that school. In a normal mixed effect model, the variable effect is assumed to be normally distributed:

$$\text{effect}_i \sim \text{normal}(\mu_i, \text{se}^2) \quad \text{for } i = 1, \cdots, 8$$

The parameter μ_i has a normal prior with hyperparameters (m, v):

$$\mu_i \sim \text{normal}(m, \text{var} = v)$$

The hyperprior distribution on m is a uniform prior on the real axis, and the hyperprior distribution on v is a uniform prior from 0 to infinity.

The following statements fit a normal mixed model, general draws from the posterior predictive distribution, and calculate relevant test quantities.

```
ods listing close;
proc mcmc data=SAT outpost=pred nmc=50000 thin=10 seed=17
   monitor=(yrep mean sd max min);

   array theta[8];
   array yrep[8];
   begincnst;
   call streaminit(1);
   endcnst;

   parms theta: 0;
   parms m 0;
   parms v 1;

   hyper m ~ general(0);
   hyper v ~ general(1,lower=0);
   prior theta: ~ normal(m,var=v);
   mu = theta[ind];
   model effect ~ normal(mu,sd=se);

   /* generate predictive data and calculate test statistics. */
   yrep[ind] = rand('normal', mu, se);
   if (ind eq 8) then do;
      mean = mean(of yrep1-yrep8);
      sd = std(of yrep1-yrep8);
      max = max(of yrep1-yrep8);
      min = min(of yrep1-yrep8);
      end;
   run;
ods listing;
```

Four test quantities constructed are: the average (mean), the sample standard deviation (sd), the maximum effect (max), and the minimum effect (min). The MONITOR= option selects yrep (replicate samples) and the four test quantities and saves them to the OUTPOST= data set. The CALL STREAMINIT routine ensures that the RAND function, used here to generate posterior predictive samples, creates a reproducible stream of random numbers. The ODS LISTING CLOSE statement disables listing output because you are primarily interested only in the samples of the monitored quantities. The HYPER, PRIOR, and MODEL statements specify the Bayesian model of interest.

The yrep[ind] assignment statement generates a random normal sample for each predictive observation, indexed by ind, with ind $= 1, \cdots, 8$. Note that this normal distribution is the same as the likelihood function specified in the MODEL statement, with the same mean and standard deviation.

To calculate the test quantities, you want to wait until all $\mathbf{y}^i_{\text{pred}}$ is generated—that is at the last observation of the data set.

The following statements compute the corresponding test statistics, the mean, standard deviation, and the minimum and maximum statistics on the real data and store them in macro variables. You then calculate the tail-area probabilities by counting the number of samples in the data set pred that are greater than the observed test statistics based on the real data.

```
proc means data=SAT noprint;
   var effect;
   output out=stat mean=mean max=max min=min stddev=sd;
   run;

data _null_;
   set stat;
   call symputx('mean',mean);
   call symputx('sd',sd);
   call symputx('min',min);
   call symputx('max',max);
   run;

data _null_;
   set pred end=eof nobs=nobs;
   ctmean + (mean>&mean);
   ctmin + (min>&min);
   ctmax + (max>&max);
   ctsd + (sd>&sd);
   if eof then do;
      pmean = ctmean/nobs; call symputx('pmean',pmean);
      pmin = ctmin/nobs; call symputx('pmin',pmin);
      pmax = ctmax/nobs; call symputx('pmax',pmax);
      psd = ctsd/nobs; call symputx('psd',psd);
      end;
   run;
```

You can plot histograms of each test quantity to visualize the posterior predictive distributions. In addition, you can see where the estimated *p*-values fall on these densities. Figure 52.12 shows the histograms. To put all four histograms on the same panel, you need to use PROC TEMPLATE (see Chapter 21, "Statistical Graphics Using ODS") and define a new graph template. The following statements defines the template **twobytwo**:

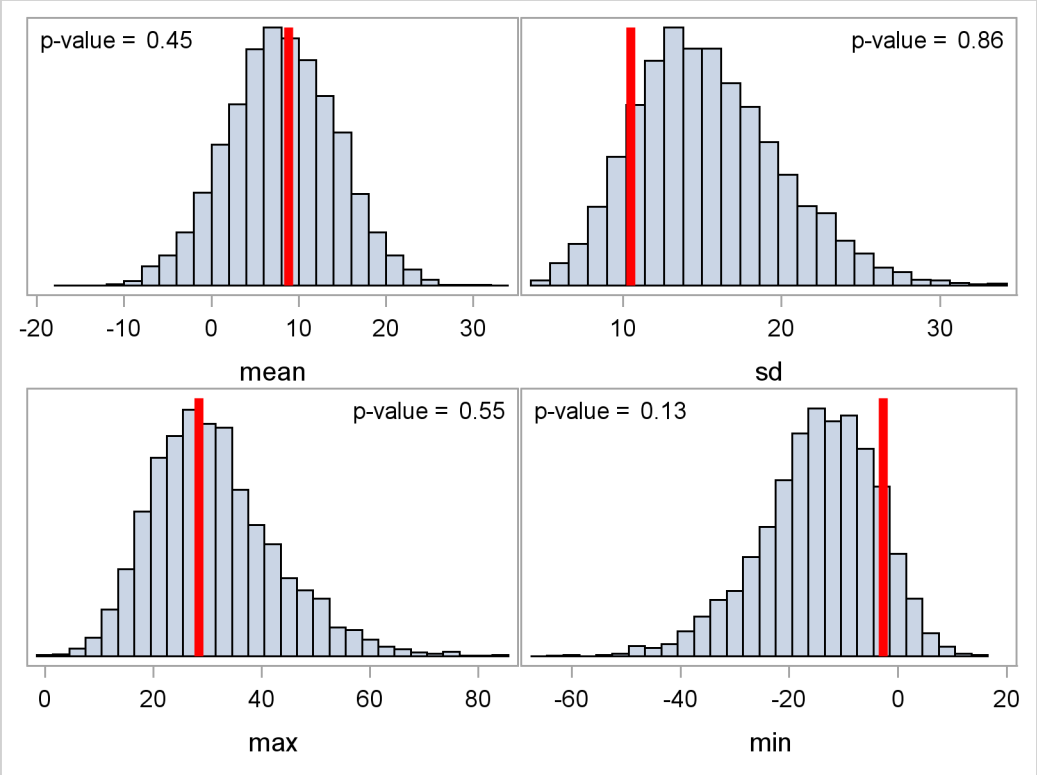
```
proc template;
   define statgraph twobytwo;
      begingraph;
         layout lattice / rows=2 columns=2;
            layout overlay / yaxisopts=(display=none)
                             xaxisopts=(label="mean");
               layout gridded / columns=2 border=false
                                autoalign=(topleft topright);
                  entry halign=right "p-value =";
                  entry halign=left eval(strip(put(&pmean, 12.2)));
               endlayout;
```

```
            histogram mean / binaxis=false;
            lineparm x=&mean y=0 slope=. /
                     lineattrs=(color=red thickness=5);
         endlayout;
         layout overlay / yaxisopts=(display=none)
                          xaxisopts=(label="sd");
            layout gridded / columns=2 border=false
                             autoalign=(topleft topright);
               entry halign=right "p-value =";
               entry halign=left eval(strip(put(&psd, 12.2)));
            endlayout;
            histogram sd / binaxis=false;
            lineparm x=&sd y=0 slope=. /
                     lineattrs=(color=red thickness=5);
         endlayout;
         layout overlay / yaxisopts=(display=none)
                          xaxisopts=(label="max");
            layout gridded / columns=2 border=false
                             autoalign=(topleft topright);
               entry halign=right "p-value =";
               entry halign=left eval(strip(put(&pmax, 12.2)));
            endlayout;
            histogram max / binaxis=false;
            lineparm x=&max y=0 slope=. /
                     lineattrs=(color=red thickness=5);
         endlayout;
         layout overlay / yaxisopts=(display=none)
                          xaxisopts=(label="min");
            layout gridded / columns=2 border=false
                             autoalign=(topleft topright);
               entry halign=right "p-value =";
               entry halign=left eval(strip(put(&pmin, 12.2)));
            endlayout;
            histogram min / binaxis=false;
            lineparm x=&min y=0 slope=. /
                     lineattrs=(color=red thickness=5);
         endlayout;
      endlayout;
   endgraph;
end;
run;
```

You call PROC SGRENDER (see the SGRENDER procedure in the *SAS/GRAPH: Statistical Graphics Procedures Guide*) to create the graph, which is shown in Figure 52.12. There are no extreme *p*-values observed; this supports the notion that the predicted results are similar to the actual observations and that the model fits the data.

```
proc sgrender data=pred template=twobytwo;
run;
```

Figure 52.12 Posterior Predictive Distribution Check for the SAT example



Handling of Missing Data

By default, PROC MCMC discards all observations that have missing values before carrying out the posterior sampling. This corresponds to the option MISSING=CC, where CC stands for complete cases. PROC MCMC does not automatically augment missing data. However, you can choose to model the missing values by using MISSING=AC. Given this option, PROC MCMC does not discard any missing values. It is up to you to specify how the missing values are handled in the program. You can choose to model the missing values as parameters (a fully Bayesian approach) or assign specific values to them (multiple imputation). In general, however, the handling of missing values largely depends on the assumptions you have about the missing mechanism, which is beyond the scope of this chapter.

Floating Point Errors and Overflows

When performing a Markov chain Monte Carlo simulation, you must calculate a proposed jump and an objective function (usually a posterior density). These calculations might lead to arithmetic exceptions and overflows. A typical cause of these problems is parameters with widely varying scales. If the posterior variances of your parameters vary by more than a few orders of magni-

tude, the numerical stability of the optimization problem can be severely reduced and can result in computational difficulties. A simple remedy is to rescale all the parameters so that their posterior variances are all approximately equal. Changing the SCALE= option might help if the scale of your parameters is much different than one. Another source of numerical instability is highly correlated parameters. Often a model can be reparameterized to reduce the posterior correlations between parameters.

If parameter rescaling does not help, consider the following actions:

- provide different initial values or try a different seed value
- use boundary constraints to avoid the region where overflows might happen
- change the algorithm (specified in programming statements) that computes the objective function

Problems Evaluating Code for Objective Function

The initial values must define a point for which the programming statements can be evaluated. However, during simulation, the algorithm might iterate to a point where the objective function cannot be evaluated. If you program your own likelihood, priors, and hyperpriors by using SAS statements and the GENERAL function in the MODEL, PRIOR, AND HYPERPRIOR statements, you can specify that an expression cannot be evaluated by setting the value you pass back through the GENERAL function to missing. This tells the PROC MCMC that the proposed set of parameters is invalid, and the proposal will not be accepted. If you use the shorthand notation that the MODEL, PRIOR, AND HYPERPRIOR statements provide, this error checking is done for you automatically.

Long Run Times

PROC MCMC can take a long time to run for problems with complex models, many parameters, or large input data sets. Although the techniques used by PROC MCMC are some of the best available, they are not guaranteed to converge or proceed quickly for all problems. Ill-posed or misspecified models can cause the algorithms to use more extensive calculations designed to achieve convergence, and this can result in longer run times. You should make sure that your model is specified correctly, that your parameters are scaled to the same order of magnitude, and that your data reasonably match the model that you are specifying.

To speed general computations, you should check over your programming statements to minimize the number of unnecessary operations. For example, you can use the proportional kernel in the priors or the likelihood and not add constants in the densities. You can also use the BEGINCNST and ENDCNST to reduce unnecessary computations on constants, and the BEGINNODATA and ENDNODATA statements to reduce observation-level calculations.

Reducing the number of blocks (the number of the PARMS statements) can speed up the sampling process. A single-block program is approximately three times faster than a three-block program for the same number of iterations. On the other hand, you do not want to put too many parameters in a single block, because blocks with large size tend not to produce well-mixed Markov chains.

Slow or No Convergence

There are a number of things to consider if the simulator is slow or fails to converge:

- Change the number of Monte Carlo iterations (NMC=), or the number of burn-in iterations (NBI=), or both. Perhaps the chain just needs to run a little longer. Note that after the simulation, you can always use the DATA step or the FIRSTOBS data set option to throw away initial observations where the algorithm has not yet burned in, so it is not always necessary to set NBI= to a large value.

- Increase the number of tuning. The proposal tuning can often work better in large models (models that have more parameters) with larger values of NTU=. The idea of tuning is to find a proposal distribution that is a good approximation to the posterior distribution. Sometimes 500 iterations per tuning phase (the default) is not sufficient to find a good approximating covariance.

- Change the initial values to more feasible starting values. Sometimes the proposal tuning starts badly if the initial values are too far away from the main mass of the posterior density, and it might not be able to recover.

- Use the PROPCOV= option to start the Markov chain at better starting values. With the PROPCOV=QUANEW option, PROC MCMC optimizes the object function and uses the posterior mode as the starting value of the Markov chain. In addition, a quadrature approximation to the posterior mode is used as the proposal covariance matrix. This option works well in many cases and can improve the mixing of the chain and shorten the tuning and burn-in time.

- Change the blocking by using the PARMS statements. Sometimes poor mixing and slow convergence can be attributed to highly correlated parameters being in different parameter blocks.

- Modify the target acceptance rate. A target acceptance rate of about 25% works well for many multi-parameter problems, but if the mixing is slow, a lower target acceptance rate might be better.

- Change the initial scaling or the TUNEWT= option to possibly help the proposal tuning.

- Consider using a different proposal distribution. If from a trace plot you see that a chain traverses to the tail area and sometimes takes quite a few simulations before it comes back, you can consider using a *t*-proposal distribution. You can do this by either using the PROC option PROPDIST=T or using a PARMS statement option T.

- Transform parameters and sample on a different scale. For example, if a parameter has a gamma distribution, sample on the logarithm scale instead. A parameter a that has a gamma distribution is equivalent to $\log(a)$ that has an egamma distribution, with the same distribution specification. For example, the following two formulations are equivalent:

```
parm a;
prior a ~ gamma(shape = 0.001, iscale = 0.001);
```

and

```
parm la;
prior la ~ egamma(shape = 0.001, iscale = 0.001);
a = exp(la);
```

See "Example 52.4: Nonlinear Poisson Regression Models" on page 3605 and "Example 52.12: Using a Transformation to Improve Mixing" on page 3683. You can also use the logit transformation on parameters that have uniform$(0, 1)$ priors. This prior is often used on probability parameters. The logit transformation is as follows: $q = \log(\frac{p}{1-p})$. The distribution on q is the Jacobian of the transformation: $\exp(-q)(1+\exp(-q))^{-2}$. Again, the following two formulations are equivalent:

```
parm p;
prior p ~ uniform(0, 1);
```

and

```
parm q;
lp = -q - 2 * log(1 + exp(-q));
prior q ~ general(lp);
p = 1/(1+exp(-q));
```

Precision of Solution

In some applications, PROC MCMC might produce parameter values that are not precise enough. Usually, this means that there were not enough iterations in the simulation. At best, the precision of MCMC estimates increases with the square of the simulation sample size. Autocorrelation in the parameter values deflate the precision of the estimates. For more information about autocorrelations in Markov chains, see the section "Autocorrelations" on page 169.

Handling Error Messages

PROC MCMC does not have a debugger. This section covers a few ways to debug and resolve error messages.

Using the PUT Statement

Adding the PUT statement often helps to find errors in a program. The following program produces an error:

```
data a;
run;

proc mcmc data=a seed=1;
   parms sigma lt w;
```

```
   beginnodata;
   prior sigma ~ unif(0.001,100);
   s2 = sigma*sigma;
   prior lt ~ gamma(shape=1, iscale=0.001);
   t = exp(lt);
   c = t/s2;
   d = 1/(s2);
   prior w ~ gamma(shape=c, iscale=d);
   endnodata;

   model general(0);
run;

ERROR: PROC MCMC is unable to generate an initial value for the
       parameter w.  The first parameter in the prior distribution is
       missing.
```

To find out why the shape parameter c is missing, you can add the put statement and examine all the calculations that lead up to the assignment of c:

```
proc mcmc data=a seed=1;
   parms sigma lt w;

   beginnodata;
   prior sigma ~ unif(0.001,100);
   s2 = sigma*sigma;
   prior lt ~ gamma(shape=1, iscale=0.001);
   t = exp(lt);
   c = t/s2;
   d = 1/(s2);
   put c= t= s2= lt=; /* display the values of these symbols. */
   prior w ~ gamma(shape=c, iscale=d);
   endnodata;

   model general(0);
run;
```

In the log file, you see the following:

```
c=. t=. s2=. lt=.
c=. t=. s2=2500.0500003 lt=1000
c=. t=. s2=2500.0500003 lt=1000
ERROR: PROC MCMC is unable to generate an initial value for the parameter w.
       The first parameter in the prior distribution is missing.
```

You can ignore the first few lines. They are the results of initial set up by PROC MCMC. The last line is important. The variable c is missing because t is the exponential of a very large number, 1000, in lt. The value 1000 is assigned to lt by PROC MCMC because none was given. The gamma prior with shape of 1 and inverse scale of 0.001 has mode 0 (see "Standard Distributions" on page 3530 for more details). PROC MCMC avoids starting the Markov chain at the boundary of the support of the distribution, and it uses the mean value here instead. The mean of the gamma

prior is 1000, hence the problem. You can change how the initial value is generated by using the PROC statement INIT=RANDOM. Do not forget to take out the put statement once you identify the problem. Otherwise, you will see a voluminous output in the log file.

Using the HYPER Statement

You can use the HYPER statement to narrow down possible errors in the prior distribution specification. With multiple PRIOR statements in a program, you might see the following error message if one of the prior distributions is not specified correctly:

```
ERROR: The initial prior parameter specifications must yield log
       of positive prior density values.
```

This message is displayed when PROC MCMC detects an error in the prior distribution calculation but cannot pinpoint the specific parameter at fault. It is frequently, although not necessarily, associated with parameters that have GENERAL or DGENERAL distributions. If you have a complicated model with many PRIOR statements, finding the parameter at fault can be time consuming. One way is to change a subset of the PRIOR statements to HYPER statements. The two statements are treated the same in PROC MCMC and the simulation is not affected, but you get a different message if the hyperprior distributions are calculated incorrectly:

```
ERROR: The initial hyperprior parameter specifications must yield
       log of positive hyperprior density values.
```

This message can help you identify more easily which distributions are producing the error, and you can then use the PUT statement to further investigate.

Computational Resources

It is not possible to estimate how long it will take for a general Markov chain to converge to its stationary distribution. It takes a skilled and thoughtful analysis of the chain to decide if it has converged to the target distribution and if the chain is mixing rapidly enough. It is easier, however, to estimate how long a particular simulation might take. The running time of a program is roughly linear to the following factors: the number of samples in the input data set (nsamples), the number of simulations (nsim), the number of blocks in the program (nblocks), and the speed of your computer. For an analysis that uses a data set of size nsamples, a simulation length of nsim, and a block design of nblocks, PROC MCMC evaluates the log-likelihood function the following number of times, excluding the tuning phase:

$$\text{nsamples} \times \text{nsim} \times \text{nblocks}$$

The faster your computer evaluates a single log-likelihood function, the faster this program runs. Suppose that you have nsamples equal to 200, nsim equal to 55,000, and nblocks equal to 3. PROC MCMC evaluates the log-likelihood function roughly a total number of 3.3×10^7 times. If your

computer can evaluate the log likelihood, for one observation, 10^6 times per second, this program will take approximately a half a minute to run. If you want to increase the number of simulations five-fold, the run time will approximately increase five-fold as well.

Of course, larger problems take longer than shorter ones, and if your model is amenable to frequentist treatment, then one of the other SAS procedures might be more suitable. With "regular" likelihoods and a lot of data, the results of standard frequentist analysis are often asymptotically equivalent to a Bayesian approach. If PROC MCMC requires too much CPU time, then perhaps another tool in SAS/STAT would be suitable.

Displayed Output

This section describes the displayed output from PROC MCMC. For a quick reference of all ODS table names, see the section "ODS Table Names" on page 3575. ODS tables are arranged under four groups, listed in the following sections: "Sampling Related ODS Tables" on page 3571, "Posterior Statistics Related ODS Tables" on page 3572, "Convergence Diagnostics Related ODS Tables" on page 3573, and "Optimization Related ODS Tables" on page 3574.

Sampling Related ODS Tables

Burn-In History

The "Burn-In History" table (ODS table name BurnInHistory) shows the scales and acceptance rates for each parameter block in the burn-in phase. The table is displayed by default.

Number of Observation Table

The "NObs" table (ODS table name NOBS) shows the number of observations that is in the data set and the number of observations that is used in the analysis. By default, observations with missing values are not used (see the section "Handling of Missing Data" on page 3565 for more details). This table is displayed by default.

Parameters

The "Parameters" table (ODS table name Parameters) shows the name of each parameter, the block number of each parameter, the sampling method used for the block, the initial values, and the prior or hyperprior distributions. This table is displayed by default.

Parameters Initial Value Table

The "Parameters Initial" table (ODS table name ParametersInit) shows the value of each parameter after the tuning phase. This table is not displayed by default and can be requested by specifying the option INIT=PINIT.

Posterior Samples

The "Posterior Samples" table (ODS table name PosteriorSample) stores posterior draws of all parameters. It is not printed by PROC MCMC. You can create an ODS output data set of the chain by specifying the following:

```
ODS OUTPUT PosteriorSample = SAS-data-set;
```

Sampling History

The "Sampling History" table (ODS table name SamplingHistory) shows the scales and acceptance rates for each parameter block in the main sampling phase. The table is displayed by default.

Tuning Covariance

The "Tuning Covariance" table (ODS table name TuneCov) shows the proposal covariance matrices for each parameter block after the tuning phase. The table is not displayed by default and can be requested by specifying the option INIT=PINIT. For more details about proposal tuning, see the section "Tuning the Proposal Distribution" on page 3525.

Tuning History

The "Tuning History" table (ODS table name TuningHistory) shows the number of tuning phases used in establishing the proposal distribution. The table also displays the scales and acceptance rates for each parameter block at each of the tuning phases. For more information about the self-adapting proposal tuning algorithm used by PROC MCMC, see the section "Tuning the Proposal Distribution" on page 3525. The table is displayed by default.

Tuning Probability Vector

The "Tuning Probability" table (ODS table name TuneP) shows the proposal probability vector for each discrete parameter block (when the option DISCRETE=GEO is specified and the geometric proposal distribution is used for discrete parameters) after the tuning phase. The table is not displayed by default and can be requested by specifying the option INIT=PINIT. For more information about proposal tuning, see the section "Tuning the Proposal Distribution" on page 3525.

Posterior Statistics Related ODS Tables

PROC MCMC calculates some essential posterior statistics and outputs them to a number of ODS tables that you can request and save individually. For details of the calculations, see the section "Summary Statistics" on page 170.

Summary Statistics

The "Posterior Summaries" table (ODS table name PostSummaries) contains basic statistics for each parameter. The table lists the number of posterior samples, the posterior mean and standard deviation estimates, and the percentile estimates. This table is displayed by default.

Correlation Matrix

The "Posterior Correlation Matrix" table (ODS table name Corr) contains the posterior correlation of model parameters. The table is not displayed by default and can be requested by specifying the option STATS=CORR.

Covariance Matrix

The "Posterior Covariance Matrix" table (ODS table name Cov) contains the posterior covariance of model parameters. The table is not displayed by default and can be requested by specifying the option STATISTICS=COV.

Deviance Information Criterion

The "Deviance Information Criterion" table (ODS table name DIC) contains the DIC of the model. The table is not displayed by default and can be requested by specifying the option DIC. For details of the calculations, see the section "Deviance Information Criterion (DIC)" on page 172.

Interval Statistics

The "Posterior Intervals" table (ODS table name PostIntervals) contains two the equal-tail and highest posterior density (HPD) interval estimates for each parameter. The default α value is 0.05, and you can change it to other levels by using the STATISTICS option. This table is displayed by default.

Convergence Diagnostics Related ODS Tables

PROC MCMC has convergence diagnostic tests that check for Markov chain convergence. The procedure produces a number of ODS tables that you can request and save individually. For details in calculation, see the section "Statistical Diagnostic Tests" on page 160.

Autocorrelation

The "Autocorrelations" table (ODS table name AUTOCORR) contains the first order autocorrelations of the posterior samples for each parameter. The "Parameter" column states the name of the parameter. By default, PROC MCMC displays lag 1, 5, 10, and 50 estimates of the autocorrelations. You can request different autocorrelations by using the DIAGNOSTICS = AUTOCORR(LAGS=) option. This table is displayed by default.

Effective Sample Size

The "Effective Sample Sizes" table (ODS table name ESS) calculates the effective sample size of each parameter. See the section "Effective Sample Size" on page 169 for more details. The table is displayed by default.

Monte Carlo Standard Errors

The "Monte Carlo Standard Errors" table (ODS table name MCSE) calculates the standard errors of the posterior mean estimate. See the section "Standard Error of the Mean Estimate" on page 170 for more details. The table is displayed by default.

Geweke Diagnostics

The "Geweke Diagnostics" table (ODS table name Geweke) lists the result of the Geweke diagnostic test. See the section "Geweke Diagnostics" on page 163 for more details. The table is displayed by default.

Heidelberger-Welch Diagnostics

The "Heidelberger-Welch Diagnostics" table (ODS table name Heidelberger) lists the result of the Heidelberger-Welch diagnostic test. The test is consisted of two parts: a stationary test and a half-width test. See the section "Heidelberger and Welch Diagnostics" on page 165 for more details. The table is not displayed by default and can be requested by specifying DIAGNOSTICS = HEIDEL.

Raftery-Lewis Diagnostics

The "Raftery-Lewis Diagnostics" table (ODS table name Raftery) lists the result of the Raftery-Lewis diagnostic test. See the section "Raftery and Lewis Diagnostics" on page 166 for more details. The table is not displayed by default and can be requested by specifying DIAGNOSTICS = RAFTERY.

Optimization Related ODS Tables

PROC MCMC can perform optimization on the joint posterior distribution. This is requested by the PROPCOV= option. The most commonly used optimization method is the quasi-Newton method: PROPCOV=QUANEW(ITPRINT). The ITPRINT option displays the ODS tables, listed as follows:

Input Options

The "Input Options" table (ODS table name InputOptions) lists optimization options used in the procedure.

Optimization Start

The "Optimization Start" table (ODS table name ProblemDescription) shows the initial state of the optimization.

Iteration History

The "Iteration History" table (ODS table name IterHist) shows iteration history of the optimization.

Optimization Results

The "Optimization Results" table (ODS table name IterStop) shows the results of the optimization, includes information about the number of function calls, and the optimized objective function, which is the joint log posterior density.

Convergence Status

The "Convergence Status" table (ODS table name ConvergenceStatus) shows whether the convergence criterion is satisfied.

Parameters Value After Optimization Table

The "Parameter Values After Optimization" table (ODS table name OptiEstimates) lists the parameter values that maximize the joint log posterior. These are the maximum a posteriori point estimates, and they are used to start the Markov chain.

Covariance Matrix After Optimization Table

The "Proposal Covariance" table (ODS table name OptiCov) lists covariance matrices for each block parameter by using quadrature approximation at the posterior mode. These covariance matrices are used in the proposal distribution.

ODS Table Names

PROC MCMC assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information about ODS, see Chapter 21, "Statistical Graphics Using ODS."

Table 52.35 ODS Tables Produced in PROC MCMC

ODS Table Name	Description	Statement or Option
AutoCorr	autocorrelation statistics for each parameter	default
PostSummaries	basic statistics for each parameter, including sample size, mean, standard deviation, and percentiles	default
ConvergenceStatus	optimization convergence status	PROPCOV=method(ITPRINT)
Corr	correlation matrix of the posterior samples	STATS=CORR
Cov	covariance matrix of the posterior samples	STATS=COV
DIC	deviance information criterion	DIC
ESS	effective sample size for each parameter	default
MCSE	Monte Carlo standard error for each parameter	default
Geweke	Geweke diagnostics for each parameter	default
Heidelberger	Heidelberger-Welch diagnostics for each parameter	DIAGNOSTICS=HEIDEL
InputOptions	optimization input table	PROPCOV=method(ITPRINT)
PostIntervals	equal-tail and HPD intervals for each parameter	default
IterHist	optimization iteration history	PROPCOV=method(ITPRINT)
IterStop	optimization results table	PROPCOV=method(ITPRINT)
NObs	number of observations	default
OptiEstimates	parameter values after either optimization	PROPCOV=method(ITPRINT)
OptiCov	covariance used in proposal distribution after optimization	PROPCOV=method(ITPRINT)
Parameters	summary of the PARMS, BLOCKING, PRIOR, sampling method, and initial value specification	default
ParametersInit	parameter values after the tuning phase	INIT=PINIT
PosteriorSample	posterior samples for each parameter	(for ODS output data set only)
ProblemDescription	optimization table	PROPCOV=method(ITPRINT)
Raftery	Raftery-Lewis diagnostics for each parameter	DIAGNOSTICS=RAFTERY
SamplingHistory	history of burn-in and main phase sampling	default

Table 52.35 (continued)

ODS Table Name	Description	Statement or Option
TuneCov	proposal covariance matrix (for continuous parameters) after the tuning phase	INIT=PINIT
TuneP	proposal probability vector (for discrete parameters) after the tuning phase	INIT=PINIT and DISCRETE=GEO
TuningHistory	history of proposal distribution tuning	default

ODS Graphics

To request graphics with PROC MCMC, you must first enable ODS Graphics by specifying the ODS GRAPHICS ON statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC MCMC generates are listed in Table 52.36.

Table 52.36 ODS Graphics Produced by PROC MCMC

ODS Graph Name	Plot Description	Statement & Option
ADPanel	autocorrelation function and density panel	PLOTS=(AUTOCORR DENSITY)
AutocorrPanel	autocorrelation function panel	PLOTS=AUTOCORR
AutocorrPlot	autocorrelation function plot	PLOTS(UNPACK)=AUTOCORR
DensityPanel	density panel	PLOTS=DENSITY
DensityPlot	density plot	PLOTS(UNPACK)=DENSITY
TAPanel	trace and autocorrelation function panel	PLOTS=(TRACE AUTOCORR)
TADPanel	trace, density, and autocorrelation function panel	PLOTS=(TRACE AUTOCORR DENSITY)
TDPanel	trace and density panel	PLOTS=(TRACE DENSITY)
TracePanel	trace panel	PLOTS=TRACE
TracePlot	trace plot	PLOTS(UNPACK)=TRACE

Examples: MCMC Procedure

Example 52.1: Simulating Samples From a Known Density

This example illustrates how you can obtain random samples from a known function. The target distributions are the normal distribution and a mixture of the normal distributions. You do not need any input data set to generate samples from a known density. You can set the likelihood function to a constant. The posterior distribution becomes identical to the prior distributions that you specify.

Sampling from a Normal Density

With a constant likelihood, there is no need to input a response variable since no data are relevant to a flat likelihood. However, PROC MCMC requires an input data set, so you can use an empty data set as the input data set. The following statements generate 10000 samples from a standard normal distribution:

```
data x;
run;

ods graphics on;
proc mcmc data=x outpost=simout seed=23 nmc=10000 maxtune=0
          nbi=0 statistics=(summary interval) diagnostics=none;
   ods exclude nobs parameters samplinghistory;
   parm alpha 0;
   prior alpha ~ normal(0, sd=1);
   model general(0);
run;
ods graphics off;
```

The ODS GRAPHICS ON statement requests ODS Graphics. The PROC MCMC statement specifies the input and output data sets, a random number seed, and the size of the simulation sample. There is no need for tuning (MAXTUNE=0) because the default scale and the proposal variance are optimal for a standard normal target distribution. For the same reason, no burn-in is needed (NBI=0). The STATISTICS= option is used to display only the summary and interval statistics. The ODS EXCLUDE statement excludes the display of the NObs, Parameters and SamplingHistory tables. The summary statistics (Output 52.1.1) are what you would expect from a standard normal distribution.

Output 52.1.1 MCMC Summary and Interval Statistics from a Normal Target Distribution

An Example for Posterior Predictive Distribution

The MCMC Procedure

Posterior Summaries

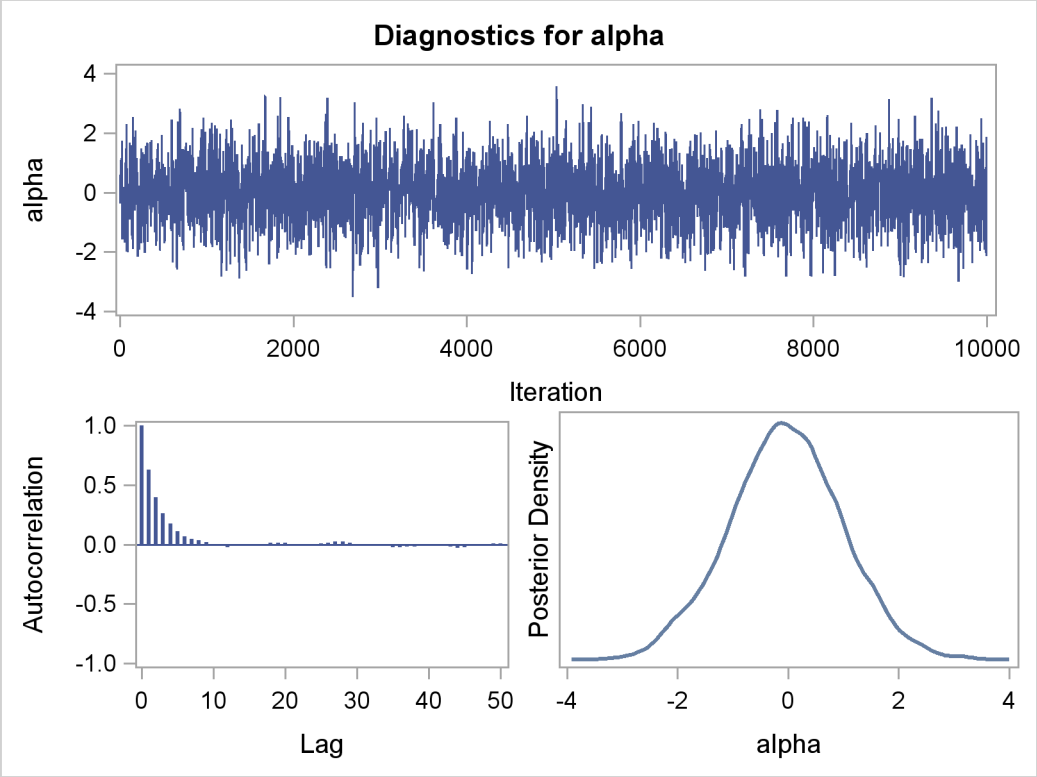
Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
alpha	10000	-0.0392	1.0194	-0.7198	-0.0403	0.6351

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
alpha	0.050	-2.0746	1.9594	-2.2197	1.7869

The trace plot (Output 52.1.2) shows good mixing of the Markov chain, and there is no significant autocorrelation in the lag plot.

Output 52.1.2 Diagnostics Plots for α



You can also overlay the estimated kernel density with the true density to get a visual comparison, as displayed in Output 52.1.3.

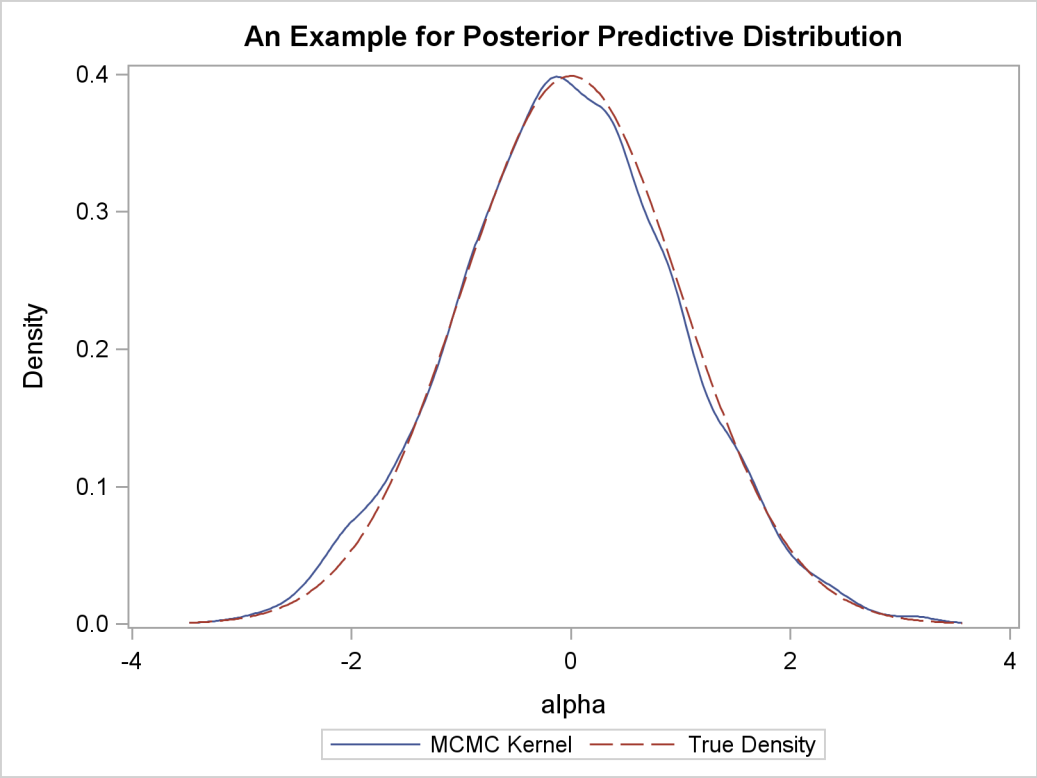
To create Output 52.1.3, you first use PROC KDE (see Chapter 45, "The KDE Procedure") to obtain a kernel density estimate of the posterior density on alpha, and then you evaluate a grid of alpha values by using PROC KDE output data set sample on a normal density. The following statements evaluate kernel density and compute corresponding normal density.

```
proc kde data=simout;
   ods exclude inputs controls;
   univar alpha /out=sample;
run;

data den;
   set sample;
   alpha = value;
   true  = pdf('normal', alpha, 0, 1);
   keep alpha density true;
run;
```

Finally, you plot the two curves on top of each other by using PROC SGPLOT (see Chapter 21, "Statistical Graphics Using ODS"); the resulting figure is in Output 52.1.3. You can see that the kernel estimate and the true density are very similar to one another. The following statements produce Output 52.1.3:

```
proc sgplot data=den;
   yaxis label="Density";
   series y=density x=alpha / legendlabel = "MCMC Kernel";
   series y=true x=alpha / legendlabel = "True Density";
   discretelegend;
run;
```

Output 52.1.3 Estimated Density versus the True Density



Sampling from a Mixture of Normal Densities

Suppose that you are interested in generating samples from a three-component mixture of normal distributions, with the density specified as follows:

$$p(\alpha) = 0.3 \cdot \phi(-3, \sigma = 2) + 0.4 \cdot \phi(2, \sigma = 1) + 0.3 \cdot \phi(10, \sigma = 4)$$

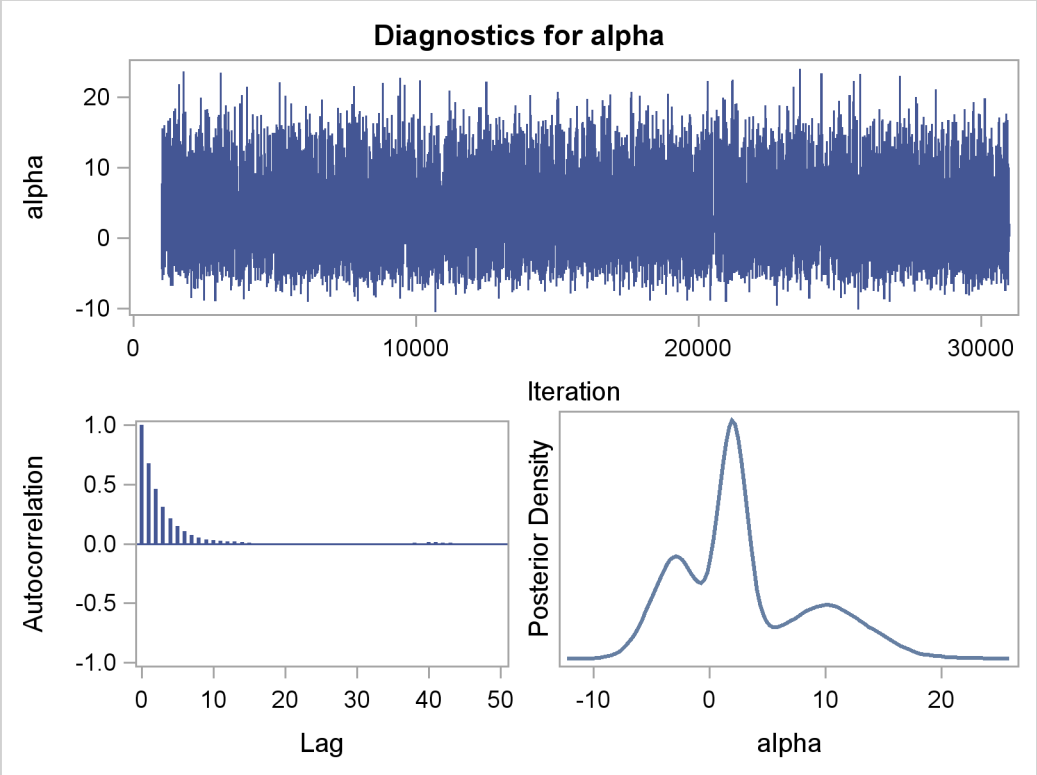
The following statements generate random samples from this mixture density:

```
data x;
run;

ods graphics on;
proc mcmc data=x outpost=simout seed=1234 nmc=30000;
   ods select TADpanel;
   parm alpha 0.3;
   lp = logpdf('normalmix', alpha, 3, 0.3, 0.4, 0.3, -3, 2, 10, 2, 1, 4);
   prior alpha  ~ general(lp);
   model general(0);
run;
ods graphics off;
```

The ODS SELECT statement displays the diagnostic plots. All other tables, such as the NObs tables, are excluded. The PROC MCMC statement uses the input data set x, saves output to the simout data set, sets a random number seed, and simulates 30,000 samples.

The lp assignment statement evaluates the log density of alpha at the mixture density, using the SAS function LOGPDF. The number 3 after alpha in the LOGPDF function indicates that the density is a three-component normal mixture. The following three numbers, 0.3, 0.4, and 0.3, are the weights in the mixture; −3, 2, and 10 are the means; 2, 1, and 4 are the standard deviations. The PRIOR statement assigns this log density function to alpha as its prior. Note that the GENERAL function interprets the density on the log scale, and not the original scale. Hence, you must use the LOGPDF function, not the PDF function. Output 52.1.4 displays the results. The kernel density clearly shows three modes.

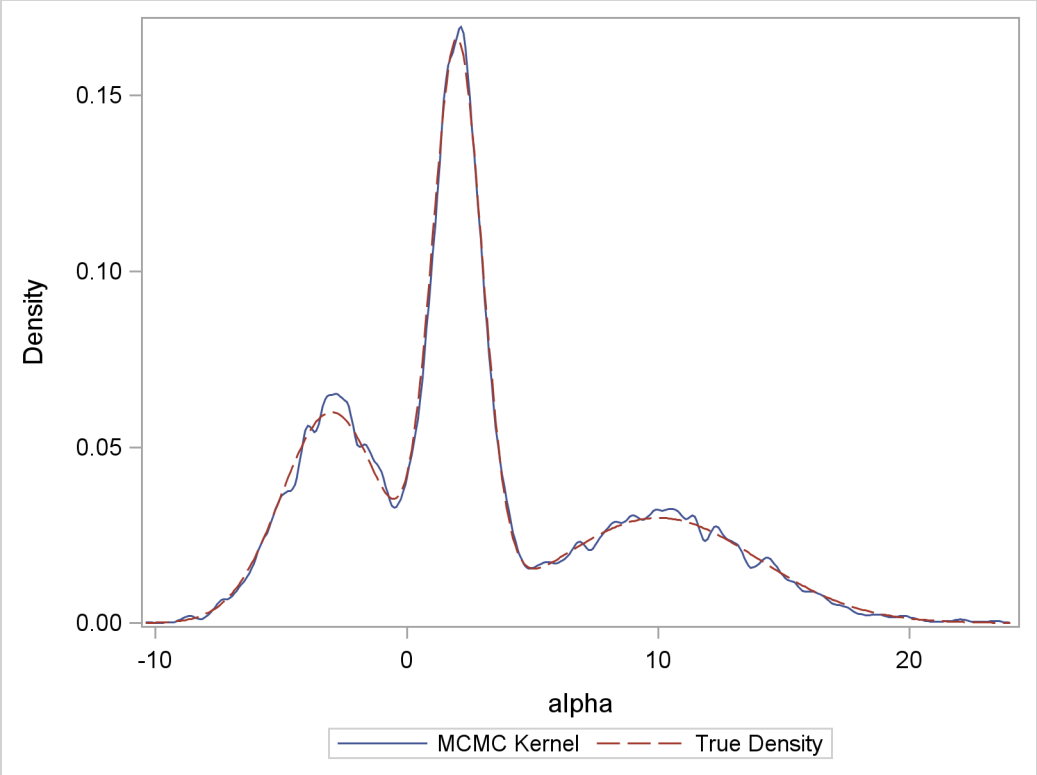
Output 52.1.4 Plots of Posterior Samples from a Mixture Normal Distribution



Using the following set of statements similar to the previous example, you can overlay the estimated kernel density with the true density. The comparison is shown in Output 52.1.5.

```
proc kde data=simout;
   ods exclude inputs controls;
   univar alpha /out=sample;
run;

data den;
   set sample;
   alpha = value;
   true  = pdf('normalmix', alpha, 3, 0.3, 0.4, 0.3, -3, 2, 10, 2, 1, 4);
   keep alpha density true;
run;

proc sgplot data=den;
   yaxis label="Density";
   series y=density x=alpha / legendlabel = "MCMC Kernel";
   series y=true x=alpha / legendlabel = "True Density";
   discretelegend;
run;
```

Output 52.1.5 Estimated Density versus the True Density



Example 52.2: Box-Cox Transformation

Box-Cox transformations (Box and Cox 1964) are often used to find a power transformation of a dependent variable to ensure the normality assumption in a linear regression model. This example illustrates how you can use PROC MCMC to estimate a Box-Cox transformation for a linear regression model. Two different priors on the transformation parameter λ are considered: a continuous prior and a discrete prior. You can estimate the probability of λ being 0 with a discrete prior but not with a continuous prior. The IF-ELSE statements are demonstrated in the example.

Using a Continuous Prior on λ

The following statements create a SAS data set with measurements of y (the response variable) and x (a single dependent variable):

```
title 'Box-Cox Transformation, with a Continuous Prior on Lambda';
data boxcox;
   input y x @@;
   datalines;
10.0  3.0  72.6  8.3  59.7  8.1  20.1  4.8  90.1  9.8   1.1  0.9
78.2  8.5  87.4  9.0   9.5  3.4   0.1  1.4   0.1  1.1  42.5  5.1

... more lines ...

 2.6  1.8  58.6  7.9  81.2  8.1  37.2  6.9
 ;
```

The Box-Cox transformation of y takes on the form of:

$$y(\lambda) = \begin{cases} \frac{y^{\lambda}-1}{\lambda} & \text{if } \lambda \neq 0; \\ \log(y) & \text{if } \lambda = 0. \end{cases}$$

The transformed response $y(\lambda)$ is assumed to be normally distributed:

$$y_i(\lambda) \sim \text{normal}(\beta_0 + \beta_1 x_i, \sigma^2)$$

The likelihood with respect to the original response y_i is as follows:

$$f(y_i|\lambda, \beta, \sigma^2, x_i) \propto \phi(y_i|\beta_0 + \beta_1 x_i, \sigma^2) \cdot J(\lambda, y_i)$$

where $J(\lambda, y_i)$ is the Jacobian:

$$J(\lambda, y) = \begin{cases} y_i^{\lambda-1} & \text{if } \lambda \neq 0; \\ 1/y_i & \text{if } \lambda = 0. \end{cases}$$

And on the log-scale, the Jacobian becomes:

$$\log(J(\lambda, y)) = \begin{cases} (\lambda - 1) \cdot \log(y_i) & \text{if } \lambda \neq 0; \\ -\log(y_i) & \text{if } \lambda = 0. \end{cases}$$

There are four model parameters: $\lambda, \beta = \{\beta_0, \beta_1\}$, and σ^2. You can considering using a flat prior on β and a gamma prior on σ^2.

To consider only power transformations ($\lambda \neq 0$), you can use a continuous prior (for example, a uniform prior from -2 to 2) on λ. One issue with using a continuous prior is that you cannot estimate the probability of $\lambda = 0$. To do so, you need to consider a discrete prior that places positive probability mass on the point 0. See "Modeling $\lambda = 0$" on page 3588.

The following statements fit a Box-Cox transformation model:

```
ods graphics on;
proc mcmc data=boxcox nmc=50000 thin=10 propcov=quanew seed=12567
          monitor=(lda);
   ods select PostSummaries PostIntervals TADpanel;

   parms beta0 0  beta1 0  lda 1 s2 1;

   beginnodata;
   prior beta: ~ general(0);
   prior s2 ~ gamma(shape=3, scale=2);
   prior lda ~ unif(-2,2);
   sd = sqrt(s2);
   endnodata;

   ys = (y**lda-1)/lda;
   mu = beta0+beta1*x;
   ll = (lda-1)*log(y)+lpdfnorm(ys, mu, sd);
   model general(ll);
run;
```

The PROPCOV option initializes the Markov chain at the posterior mode and uses the estimated inverse Hessian matrix as the initial proposal covariance matrix. The MONITOR= option selects λ as the variable to report. The ODS SELECT statement displays the summary statistics table, the interval statistics table, and the diagnostic plots.

The PARMS statement puts all four parameters, β_0, β_1, λ, and σ^2, in a single block and assigns initial values to each of them. Three PRIOR statements specify previously stated prior distributions for these parameters. The assignment to sd transforms a variance to a standard deviation. It is better to place the transformation inside the BEGINNODATA and ENDNODATA statements to save computational time.

The assignment to the symbol ys evaluates the Box-Cox transformation of y, where mu is the regression mean and ll is the log likelihood of the transformed variable ys. Note that the log of the Jacobian term is included in the calculation of ll.

Summary statistics and interval statistics for lda are listed in Output 52.2.1.

Output 52.2.1 Box-Cox Transformation

Box-Cox Transformation, with a Continuous Prior on Lambda

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
lda	5000	0.4702	0.0284	0.4515	0.4703	0.4884

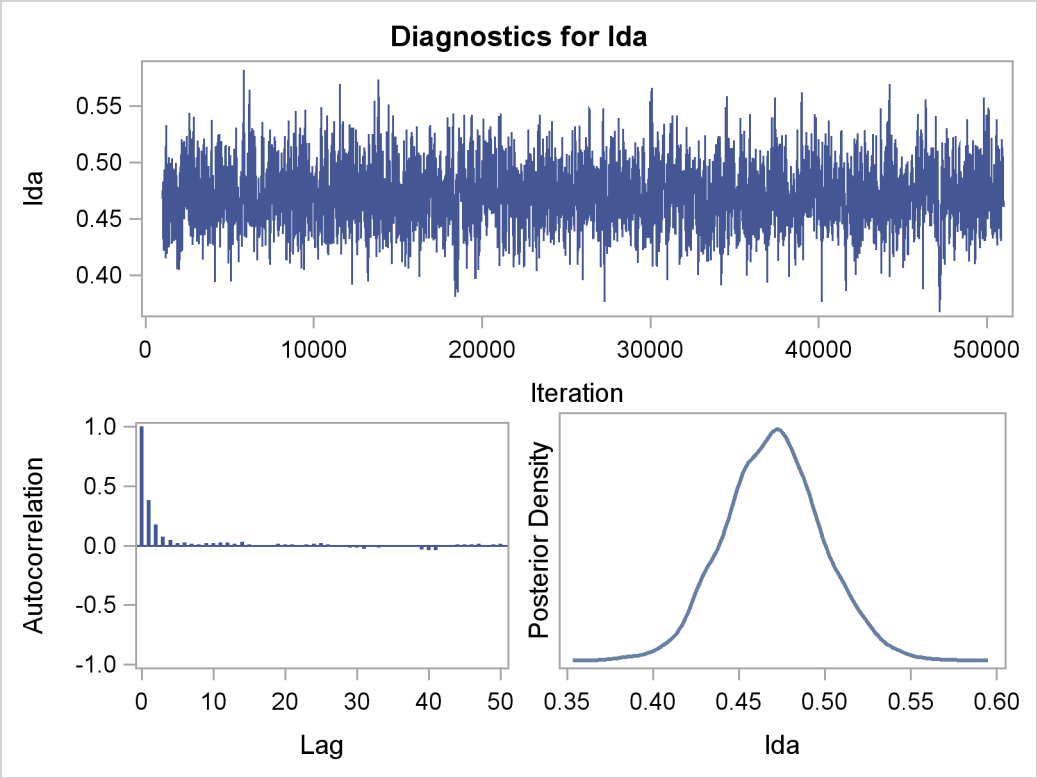
Output 52.2.1 *continued*

Posterior Intervals					
Parameter	Alpha	Equal-Tail Interval		HPD Interval	
lda	0.050	0.4162	0.5269	0.4197	0.5298

The posterior mean of λ is 0.47, with a 95% equal-tail interval of $[0.42, 0.53]$ and a similar HPD interval. The prefered power transformation would be 0.5 (rounding λ up to the square root transformation).

Output 52.2.2 shows diagnostics plots for lda. The chain appears to converge, and you can proceed to make inferences. The density plot shows that the posterior density is relatively symmetric around its mean estimate.

Output 52.2.2 Diagnostic Plots for λ

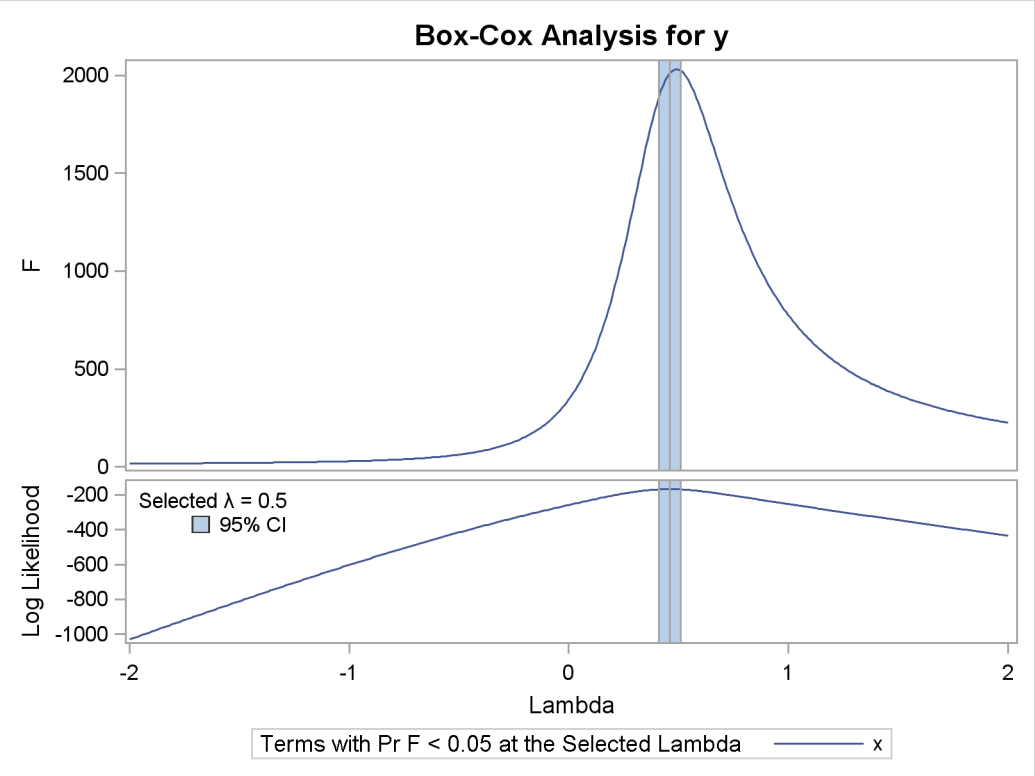


To verify the results, you can use PROC TRANSREG (see Chapter 90, "The TRANSREG Procedure") to find the estimate of λ.

```
proc transreg data=boxcox details pbo;
   ods output boxcox = bc;
   model boxcox(y / convenient lambda=-2 to 2 by 0.01) = identity(x);
   output out=trans;
run;
ods graphics off;
```

Output from PROC TRANSREG is shown in Output 52.2.5 and Output 52.2.4. PROC TRANSREG produces a similar point estimate of $\lambda = 0.46$, and the 95% confidence interval is shown in Output 52.2.5.

Output 52.2.3 Box-Cox Transformation Using PROC TRANSREG



Output 52.2.4 Estimates Reported by PROC TRANSREG

Box-Cox Transformation, with a Continuous Prior on Lambda

The TRANSREG Procedure

Model Statement Specification Details

Type	DF	Variable	Description	Value
Dep	1	BoxCox(y)	Lambda Used	0.5
			Lambda	0.46
			Log Likelihood	-167.0
			Conv. Lambda	0.5
			Conv. Lambda LL	-168.3
			CI Limit	-169.0
			Alpha	0.05
			Options	Convenient Lambda Used
Ind	1	Identity(x)	DF	1

The ODS data set bc contains the 95% confidence interval estimates produced by PROC TRANSREG. This ODS table is rather large, and you want to see only the relevant portion. The following statements generate the part of the table that is important and display Output 52.2.5:

```
proc print noobs label data=bc(drop=rmse);
   title2 'Confidence Interval';
   where ci ne ' ' or abs(lambda - round(lambda, 0.5)) < 1e-6;
   label convenient = '00'x ci = '00'x;
run;
```

The estimated 90% confidence interval is $[0.41, 0.51]$, which is very close to the reported Bayesian credible intervals. The resemblance of the intervals is probably due to the noninformative prior that you used in this analysis.

Output 52.2.5 Estimated Confidence Interval on λ

Box-Cox Transformation, with a Continuous Prior on Lambda
Confidence Interval

Dependent	Lambda		R-Square	Log Likelihood	
BoxCox(y)	-2.00		0.14	-1030.56	
BoxCox(y)	-1.50		0.17	-810.50	
BoxCox(y)	-1.00		0.22	-602.53	
BoxCox(y)	-0.50		0.39	-415.56	
BoxCox(y)	0.00		0.78	-257.92	
BoxCox(y)	0.41		0.95	-168.40	*
BoxCox(y)	0.42		0.95	-167.86	*
BoxCox(y)	0.43		0.95	-167.46	*
BoxCox(y)	0.44		0.95	-167.19	*
BoxCox(y)	0.45		0.95	-167.05	*
BoxCox(y)	0.46		0.95	-167.04	<
BoxCox(y)	0.47		0.95	-167.16	*
BoxCox(y)	0.48		0.95	-167.41	*
BoxCox(y)	0.49		0.95	-167.79	*
BoxCox(y)	0.50	+	0.95	-168.28	*
BoxCox(y)	0.51		0.95	-168.89	*
BoxCox(y)	1.00		0.89	-253.09	
BoxCox(y)	1.50		0.79	-345.35	
BoxCox(y)	2.00		0.70	-435.01	

Modeling $\lambda = 0$

With a continuous prior on λ, you can get only a continuous posterior distribution, and this makes the probability of $\Pr(\lambda = 0|\text{data})$ equal to 0 by definition. To consider $\lambda = 0$ as a viable solution to the Box-Cox transformation, you need to use a discrete prior that places some probability mass on the point 0 and allows for a meaningful posterior estimate of $\Pr(\lambda = 0|\text{data})$.

This example uses a simulation study where the data are generated from an exponential likelihood. The simulation implies that the correct transformation should be the logarithm and λ should be 0.

Consider the following exponential model:

$$y = \exp(x + \epsilon),$$

where $\epsilon \sim \text{normal}(0, 1)$. The transformed data can be fitted with a linear model:

$$\log(y) = x + \epsilon$$

The following statements generate a SAS data set with a gridded x and corresponding y:

```
title 'Box-Cox Transformation, Modeling Lambda = 0';
data boxcox;
   do x = 1 to 8 by 0.025;
      ly = x + normal(7);
      y = exp(ly);
      output;
   end;
run;
```

The log-likelihood function, after taking the Jacobian into consideration, is as follows:

$$\log p(y_i|\lambda, x_i) = \begin{cases} (\lambda-1)\log(y_i) - \frac{1}{2}\left(\log\sigma^2 + \frac{\left((y_i^\lambda - 1)/\lambda - x_i\right)^2}{\sigma^2}\right) + C_1 & \text{if } \lambda \neq 0; \\ -\log(y_i) - \frac{1}{2}\left(\log\sigma^2 + \frac{(\log(y_i) - x_i)^2}{\sigma^2}\right) + C_2 & \text{if } \lambda = 0. \end{cases}$$

where C_1 and C_2 are two constants.

You can use the function DGENERAL to place a discrete prior on λ. The function is similar to the function GENERAL, except that it indicates a discrete distribution. For example, you can specify a discrete uniform prior from -2 to 2 using

```
prior lda ~ dgeneral(1, lower=-2, upper=2);
```

This places equal probability mass on five points, -2, -1, 0, 1, and 2. This prior might not work well here because the grid is too coarse. To consider smaller values of λ, you can sample a parameter that takes a wider range of integer values and transform it back to the λ space. For example, set alpha as your model parameter and give it a discrete uniform prior from -200 to 200. Then define λ as alpha/100 so λ can take values between -2 and 2 but on a finer grid.

The following statements fit a Box-Cox transformation by using a discrete prior on λ:

```
proc mcmc data=boxcox outpost=simout nmc=50000 thin=10 seed=12567
          monitor=(lda);

   ods select PostSummaries PostIntervals;
   parms s2 1 alpha 10;

   beginnodata;
   prior s2 ~ gamma(shape=3, scale=2);
   if alpha=0 then lp = log(2);
      else lp = log(1);
   prior alpha ~ dgeneral(lp, lower=-200, upper=200);
```

```
   lda = alpha * 0.01;
   sd = sqrt(s2);
   endnodata;

   if alpha=0 then
      ll = -ly+lpdfnorm(ly, x, sd);
   else do;
      ys = (y**lda - 1)/lda;
      ll = (lda-1)*ly+lpdfnorm(ys, x, sd);
   end;
   model general(ll);
run;
```

There are two parameters, s2 and alpha, in the model. They are placed in a single PARMS statement so that they are sampled in the same block.

The parameter s2 takes a gamma distribution, and alpha takes a discrete prior. The IF-ELSE statements state that alpha takes twice as much prior density when it is 0 than otherwise. Note that on the original scale, $\Pr(\text{alpha} = 0) = 2 \cdot \Pr(\text{alpha} \neq 0)$. Translating that to the log scale, the densities become $\log(2)$ and $\log(1)$, respectively. The lda assignment statement transforms alpha to the parameter of interest: lda takes values between -2 and 2. You can model lda on a even smaller scale by dividing alpha by a larger constant. However, an increment of 0.01 in the Box-Cox transformation is usually sufficient. The sd assignment statement calculates the square root of the variance term.

The log-likelihood function uses another set of IF-ELSE statements, separating the case of $\lambda = 0$ from the others. The formulas are stated previously. The output of the program is shown in Output 52.2.6.

Output 52.2.6 Box-Cox Transformation

Box-Cox Transformation, Modeling Lambda = 0

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
lda	5000	-0.00002	0.00201	0	0	0

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
lda	0.050	0	0	0	0

From the summary statistics table, you see that the point estimate for λ is 0 and both of the 95% equal-tail and HPD credible intervals are 0. This strongly suggests that $\lambda = 0$ is the best estimate for this problem. In addition, you can also count the frequency of λ among posterior samples to get a more precise estimate on the posterior probability of λ being 0.

The following statements use PROC FREQ to produce Output 52.2.7 and Output 52.2.8:

```
ods graphics on;
proc freq data=simout;
   ods select onewayfreqs freqplot;
   tables lda /nocum plot=freqplot(scale=percent);
run;
ods graphics off;
```

Output 52.2.7 shows the frequency count table. An estimate of $\Pr(\lambda = 0|\text{data})$ is 96%. The conclusion is that the log transformation should be the appropriate transformation used here, which agrees with the simulation setup. Output 52.2.8 shows the histogram of λ.

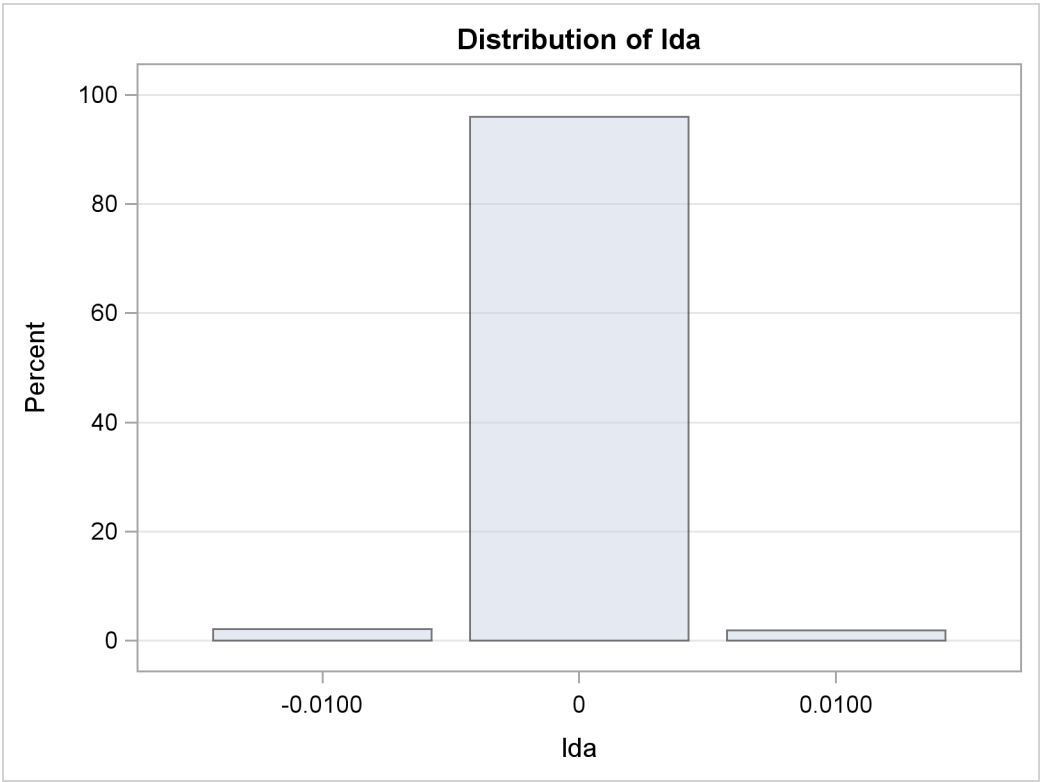
Output 52.2.7 Frequency Counts of λ

Box-Cox Transformation, Modeling Lambda = 0

The FREQ Procedure

lda	Frequency	Percent
-0.0100	106	2.12
0	4798	95.96
0.0100	96	1.92

Output 52.2.8 Histogram of λ



Example 52.3: Generalized Linear Models

This example discusses two examples of fitting generalized linear models (GLM) with PROC MCMC. One uses a logistic regression model and one uses a Poisson regression model. The logistic examples use both a diffuse prior and a Jeffreys' prior on the regression coefficients. You can also use the BAYES statement in PROC GENMOD. See Chapter 37, "The GENMOD Procedure."

Logistic Regression Model with a Diffuse Prior

The following statements create a SAS data set with measurements of the number of deaths, y, among n beetles that have been exposed to an environmental contaminant x:

```
title 'Logistic Regression Model with a Diffuse Prior';
data beetles;
   input n y x @@;
   datalines;
6  0  25.7   8  2  35.9   5  2  32.9   7  7  50.4   6  0  28.3
7  2  32.3   5  1  33.2   8  3  40.9   6  0  36.5   6  1  36.5
6  6  49.6   6  3  39.8   6  4  43.6   6  1  34.1   7  1  37.4
8  2  35.2   6  6  51.3   5  3  42.5   7  0  31.3   3  2  40.6
;
```

You can model the data points y_i with a binomial distribution:

$$y_i \,|\, p_i \sim \text{binomial}(n_i, p_i)$$

where p_i is the success probability and links to the regression covariate x_i through a logit transformation:

$$logit(p_i) = \log\left(\frac{p_i}{1-p_i}\right) = \alpha + \beta x_i$$

The priors on α and β are both diffuse normal:

$$\begin{aligned} \pi(\alpha) &= \phi(0, \text{var} = 10000) \\ \pi(\beta) &= \phi(0, \text{var} = 10000) \end{aligned}$$

These statements fit a logistic regression with PROC MCMC:

```
ods graphics on;
proc mcmc data=beetles ntu=1000 nmc=20000 nthin=2 propcov=quanew
          diag=(mcse ess) outpost=beetleout seed=246810;
   ods select PostSummaries PostIntervals mcse ess TADpanel;
   parms (alpha  beta) 0;
   prior alpha beta ~ normal(0, var = 10000);
   p = logistic(alpha + beta*x);
   model y ~ binomial(n,p);
run;
```

The key statement in the program is the assignment to p that calculates the probability of death. The SAS function LOGISTIC does the proper transformation. The MODEL statement specifies that the response variable, y, is binomially distributed with parameters n (from the input data set) and p. The summary statistics table, interval statistics table, the Monte Carlos standard error table, and the effective sample sizes table are shown in Output 52.3.1.

Output 52.3.1 MCMC Results

Logistic Regression Model with a Diffuse Prior

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
alpha	10000	−11.7707	2.0997	−13.1243	−11.6683	−10.3003
beta	10000	0.2920	0.0542	0.2537	0.2889	0.3268

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
alpha	0.050	−16.3332	−7.9675	−15.8822	−7.6673
beta	0.050	0.1951	0.4087	0.1901	0.4027

Logistic Regression Model with a Diffuse Prior

The MCMC Procedure

Monte Carlo Standard Errors

Parameter	MCSE	Standard Deviation	MCSE/SD
alpha	0.0422	2.0997	0.0201
beta	0.00110	0.0542	0.0203

Effective Sample Sizes

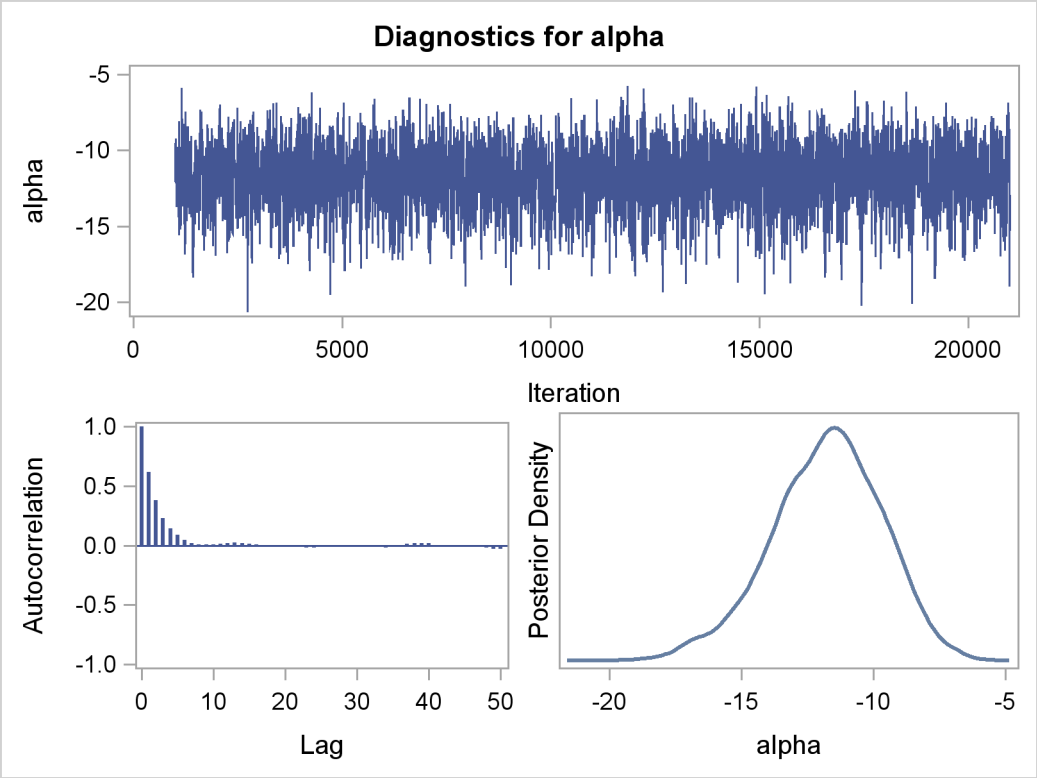
Parameter	ESS	Correlation Time	Efficiency
alpha	2470.1	4.0484	0.2470
beta	2435.4	4.1060	0.2435

The summary statistics table shows that the sample mean of the output chain for the parameter alpha is −11.7707. This is an estimate of the mean of the marginal posterior distribution for the intercept parameter alpha. The estimated posterior standard deviation for alpha is 2.0997. The two 95% credible intervals for alpha are both negative, which indicates with very high probability that the intercept term is negative. On the other hand, you observe a positive effect on the regression coefficient beta. Exposure to the environment contaminant increases the probability of death.

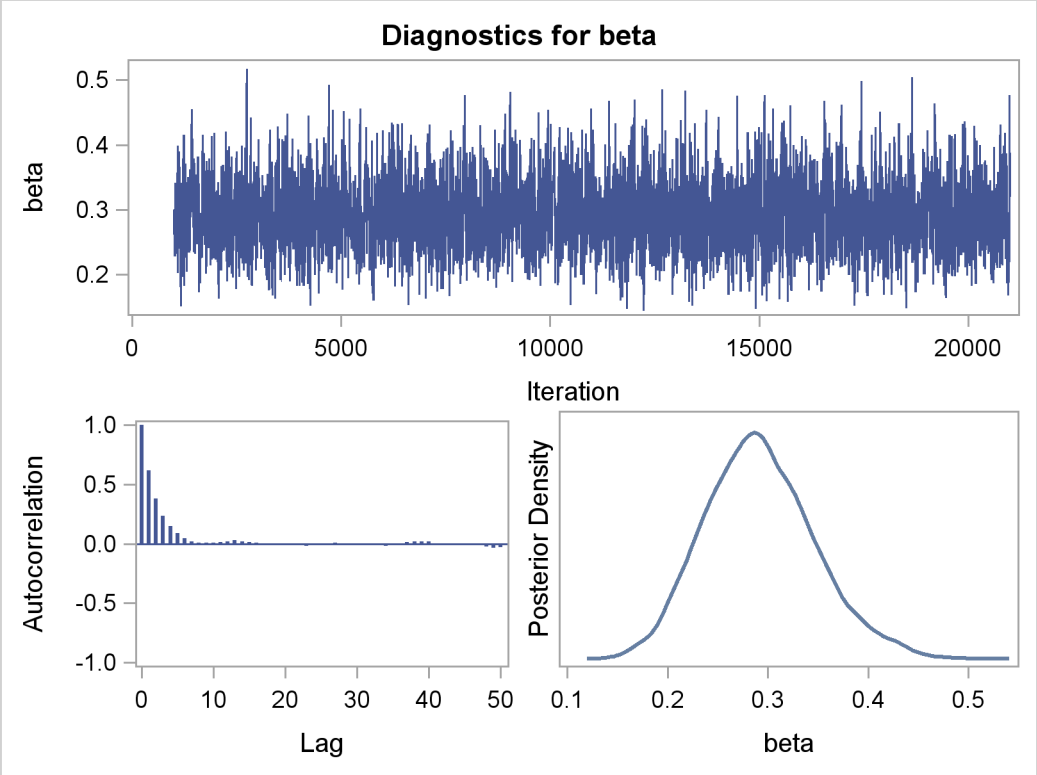
The Monte Carlo standard errors of each parameter are significantly small relative to the posterior standard deviations. A small MCSE/SD ratio indicates that the Markov chain has stabilized and the mean estimates do not vary much over time. Note that the precision in the parameter estimates increases with the square of the MCMC sample size, so if you want to double the precision, you must quadruple the MCMC sample size.

MCMC chains do not produce independent samples. Each sample point depends on the point before it. In this case, the correlation time estimate, read from the effective sample sizes table, is roughly 4. This means that it takes four observations from the MCMC output to make inferences about alpha with the same precision that you would get from using an independent sample. The effective sample size of 2470 reflects this loss of efficiency. The coefficient beta has similar efficiency. You can often observe that some parameters have significantly better mixing (better efficiency) than others, even in a single Markov chain run.

Output 52.3.2 Plots for Parameters in the Logistic Regression Example



Output 52.3.2 *continued*



Trace plots and autocorrelation plots of the posterior samples are shown in Output 52.3.2. Convergence looks good in both parameters; there is good mixing in the trace plot and quick drop-off in the ACF plot.

One advantage of Bayesian methods is the ability to directly answer scientific questions. In this example, you might want to find out the posterior probability that the environmental contaminant increases the probability of death—that is, $Pr(\beta > 0|\mathbf{y})$. This can be estimated using the following steps:

```
proc format;
   value betafmt low-0 = 'beta <= 0' 0<-high = 'beta > 0';
run;

proc freq data=beetleout;
   tables beta /nocum;
   format beta betafmt.;
run;
```

Output 52.3.3 Frequency Counts

Logistic Regression Model with a Diffuse Prior

The FREQ Procedure

beta	Frequency	Percent
beta > 0	10000	100.00

All of the simulated values for β are greater than zero, so the sample estimate of the posterior probability that $\beta > 0$ is 100%. The evidence overwhelmingly supports the hypothesis that increased levels of the environmental contaminant increase the probability of death.

If you are interested in making inference based on any quantities that are transformations of the random variables, you can either do it directly in PROC MCMC or by using the DATA step after you run the simulation. Transformations sometimes can make parameter inference quite formidable using direct analytical methods, but with simulated chains, it is easy to compute chains for any set of parameters. Suppose that you are interested in the lethal dose and want to estimate the level of the covariate x that corresponds to a probability of death, p. Abbreviate this quantity as ldp. In other words, you want to solve the logit transformation with a fixed value p. The lethal dose is as follows:

$$\text{ldp} = \frac{\log\left(\frac{p}{1-p}\right) - \alpha}{\beta}$$

You can obtain an estimate of any ldp by using the posterior mean estimates for α and β. For example, lp95, which corresponds to $p = 0.95$, is calculated as follows:

$$\text{lp95} = \frac{\log\left(\frac{0.95}{1-0.95}\right) + 11.77}{0.29} = 50.79$$

where -11.77 and 0.29 are the posterior mean estimates of α and β, respectively, and 50.79 is the estimated lethal dose that leads to a 95% death rate.

While it is easy to obtain the point estimates, it is harder to estimate other posterior quantities, such as the standard deviation directly. However, with PROC MCMC, you can trivially get estimates of any posterior quantities of lp95. Consider the following program in PROC MCMC:

```
proc mcmc data=beetles ntu=1000 nmc=20000 nthin=2 propcov=quanew
          outpost=beetleout seed=246810 plot=density
          monitor=(pi30 ld05 ld50 ld95);
   ods select PostSummaries PostIntervals densitypanel;
   parms (alpha  beta) 0;
   begincnst;
      c1 = log(0.05 / 0.95);
      c2 = -c1;
   endcnst;

   beginnodata;
   prior alpha beta ~ normal(0, var = 10000);
   pi30 = logistic(alpha + beta*30);
   ld05 = (c1 - alpha) / beta;
```

```
   ld50 = - alpha / beta;
   ld95 = (c2 - alpha) / beta;
   endnodata;
   pi = logistic(alpha + beta*x);
   model y ~ binomial(n,pi);
run;
ods graphics off;
```

The program estimates four additional posterior quantities. The three lpd quantities, ld05, ld50, and ld95, are the three levels of the covariate that kills 5%, 50%, and 95% of the population, respectively. The predicted probability when the covariate x takes the value of 30 is pi30. The MONITOR= option selects the quantities of interest. The PLOTS= option selects kernel density plots as the only ODS graphical output, excluding the trace plot and autocorrelation plot.

Programming statements between the BEGINCNST and ENDCNST statements define two constants. These statements are executed once at the beginning of the simulation. The programming statements between the BEGINNODATA and ENDNODATA statements evaluate the quantities of interest. The symbols, pi30, ld05, ld50, and ld95, are functions of the parameters alpha and beta only. Hence, they should not be processed at the observation level and should be included in the BEGINNODATA and ENDNODATA statements. Output 52.3.4 lists the posterior summary and Output 52.3.5 shows the density plots of these posterior quantities.

Output 52.3.4 PROC MCMC Results

Logistic Regression Model with a Diffuse Prior

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
pi30	10000	0.0524	0.0253	0.0340	0.0477	0.0662
ld05	10000	29.9281	1.8814	28.8430	30.1727	31.2563
ld50	10000	40.3745	0.9377	39.7271	40.3165	40.9612
ld95	10000	50.8210	2.5353	49.0372	50.5157	52.3100

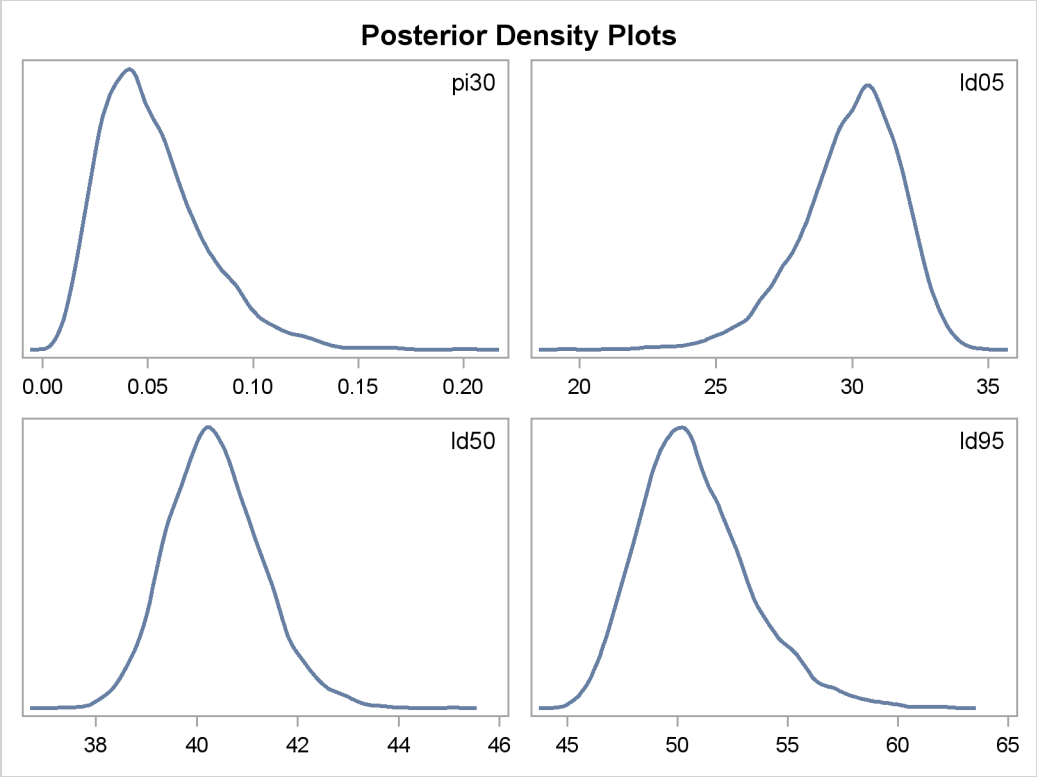
Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
pi30	0.050	0.0161	0.1133	0.0109	0.1008
ld05	0.050	25.6409	32.9660	26.2193	33.2774
ld50	0.050	38.6706	42.3718	38.6194	42.2811
ld95	0.050	46.7180	56.7667	46.3221	55.8774

The posterior mean estimate of lp95 is 50.82, which is close to the estimate of 50.79 by using the posterior mean estimates of the parameters. With PROC MCMC, in addition to the mean estimate, you can get the standard deviation, quantiles, and interval estimates at any level of significance.

From the density plots, you can see, for example, that the sample distribution for π_{30} is skewed to the right, and almost all of your posterior belief concerning π_{30} is concentrated in the region between zero and 0.15.

Output 52.3.5 Density Plots of Quantities of Interest in the Logistic Regression Example



It is easy to use the DATA step to calculate these quantities of interest. The following DATA step uses the simulated values of α and β to create simulated values from the posterior distributions of ld05, ld50, ld95, and π_{30}:

```
data transout;
   set beetleout;
   pi30 = logistic(alpha + beta*30);
   ld05 = (log(0.05 / 0.95) - alpha) / beta;
   ld50 = (log(0.50 / 0.50) - alpha) / beta;
   ld95 = (log(0.95 / 0.05) - alpha) / beta;
run;
```

Subsequently, you can use SAS/INSIGHT, or the UNIVARIATE, CAPABILITY, or KDE procedures to analyze the posterior sample. If you want to regenerate the default ODS graphs from PROC MCMC, see "Regenerating Diagnostics Plots" on page 3557.

Logistic Regression Model with Jeffreys' Prior

A controlled experiment was run to study the effect of the rate and volume of air inspired on a transient reflex vasoconstriction in the skin of the fingers. Thirty-nine tests under various combinations

of rate and volume of air inspired were obtained (Finney 1947). The result of each test is whether or not vasoconstriction occurred. Pregibon (1981) uses this set of data to illustrate the diagnostic measures he proposes for detecting influential observations and to quantify their effects on various aspects of the maximum likelihood fit. The following statements create the data set vaso:

```
title 'Logistic Regression Model with Jeffreys Prior';
data vaso;
   input vol rate resp @@;
   lvol = log(vol);
   lrate = log(rate);
   ind = _n_;
   cnst = 1;
   datalines;
3.7  0.825  1  3.5  1.09  1  1.25  2.5   1  0.75  1.5  1
0.8  3.2    1  0.7  3.5   1  0.6   0.75  0  1.1   1.7  0
0.9  0.75   0  0.9  0.45  0  0.8   0.57  0  0.55  2.75 0
0.6  3.0    0  1.4  2.33  1  0.75  3.75  1  2.3   1.64 1
3.2  1.6    1  0.85 1.415 1  1.7   1.06  0  1.8   1.8  1
0.4  2.0    0  0.95 1.36  0  1.35  1.35  0  1.5   1.36 0
1.6  1.78   1  0.6  1.5   0  1.8   1.5   1  0.95  1.9  0
1.9  0.95   1  1.6  0.4   0  2.7   0.75  1  2.35  0.03 0
1.1  1.83   0  1.1  2.2   1  1.2   2.0   1  0.8   3.33 1
0.95 1.9    0  0.75 1.9   0  1.3   1.625 1
;
```

The variable resp represents the outcome of a test. The variable lvol represents the log of the volume of air intake, and the variable lrate represents the log of the rate of air intake. You can model the data by using logistic regression. You can model the response with a binary likelihood:

$$\text{resp}_i \sim \text{binary}(p_i)$$

with

$$p_i = \frac{1}{1 + \exp(-(\beta_0 + \beta_1 \text{lvol}_i + \beta_2 \text{lrate}_i))}$$

Let X be the design matrix in the regression. Jeffreys' prior for this model is

$$p(\beta) \propto |X^\top \boldsymbol{M} X|^{1/2}$$

where $\boldsymbol{M}$ is a 39 by 39 matrix with off-diagonal elements being 0 and diagonal elements being $p_i(1 - p_i)$. For details on Jeffreys' prior, see "Jeffreys' Prior" on page 146. You can use a number of matrix functions, such as the determinant function, in PROC MCMC to construct Jeffreys' prior. The following statements illustrate how to fit a logistic regression with Jeffreys' prior:

```
/* fitting a logistic regression with Jeffreys' prior */
%let n = 39;
proc mcmc data=vaso nmc=10000 outpost=mcmcout seed=17;
   ods select PostSummaries PostIntervals;

   array beta[3] beta0 beta1 beta2;
   array m[&n, &n];
   array x[&n, 3];
```

```
   array xt[3, &n];
   array xtm[3, &n];
   array xmx[3, 3];
   array p[&n];

   parms beta0 1 beta1 1 beta2 1;

   begincnst;
      x[ind, 1] = 1;
      x[ind, 2] = lvol;
      x[ind, 3] = lrate;
      if (ind eq &n) then do;
         call transpose(x, xt);
         call zeromatrix(m);
      end;
   endcnst;

   beginnodata;
   call mult(x, beta, p);              /* p = x * beta */
   do i = 1 to &n;
      p[i] = 1 / (1 + exp(-p[i]));     /* p[i] = 1/(1+exp(-x*beta)) */
      m[i,i] = p[i] * (1-p[i]);
   end;
   call mult (xt, m, xtm);             /* xtm = xt * m        */
   call mult (xtm, x, xmx);            /* xmx = xtm * x       */
   call det (xmx, lp);                 /* lp = det(xmx)       */
   lp = 0.5 * log(lp);                 /* lp = -0.5 * log(lp) */
   prior beta: ~ general(lp);
   endnodata;

   model resp ~ bern(p[ind]);
run;
```

The first ARRAY statement defines an array beta with three elements: beta0, beta1, and beta2. The subsequent statements define arrays that are used in the construction of Jeffreys' prior. These include m (the $\boldsymbol{M}$ matrix), x (the design matrix), xt (the transpose of x), and some additional work spaces.

The explanatory variables lvol and lrate are saved in the array x in the BEGINCNST and ENDCNST statements. See "BEGINCNST/ENDCNST Statement" on page 3509 for details. After all the variables are read into x, you transpose the x matrix and store it to xt. The ZEROMATRIX function call assigns all elements in matrix m the value zero. To avoid redundant calculation, it is best to perform these calculations as the last observation of the data set is processed—that is, when ind is 39.

You calculate Jeffreys' prior in the BEGINNODATA and ENDNODATA statements. The probability vector p is the product of the design matrix x and parameter vector beta. The diagonal elements in the matrix m are $p_i(1 - p_i)$. The expression lp is the logarithm of Jeffreys' prior. The PRIOR statement assigns lp as the prior for the β regression coefficients. The MODEL statement assigns a binary likelihood to resp, with probability p[ind]. The p array is calculated earlier using the matrix function MULT. You use the ind variable to pick out the right probability value for each resp.

Posterior summary statistics are displayed in Output 52.3.6.

Output 52.3.6 PROC MCMC Results, Jeffreys' prior

Logistic Regression Model with Jeffreys Prior

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
beta0	10000	-2.9587	1.3258	-3.8117	-2.7938	-2.0007
beta1	10000	5.2905	1.8193	3.9861	5.1155	6.4145
beta2	10000	4.6889	1.8189	3.3570	4.4914	5.8547

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
beta0	0.050	-5.8247	-0.7435	-5.5936	-0.6027
beta1	0.050	2.3001	9.3789	1.8590	8.7222
beta2	0.050	1.6788	8.6643	1.3611	8.2490

You can also use PROC GENMOD to fit the same model by using the following statements:

```
proc genmod data=vaso descending;
   ods select PostSummaries PostIntervals;
   model resp = lvol lrate / d=bin link=logit;
   bayes seed=17 coeffprior=jeffreys nmc=20000 thin=2;
run;
```

The MODEL statement indicates that resp is the response variable and lvol and lrate are the covariates. The options in the MODEL statement specify a binary likelihood and a logit link function. The BAYES statement requests Bayesian capability. The SEED=, NMC=, and THIN= arguments work in the same way as in PROC MCMC. The COEFFPRIOR=JEFFREYS option requests Jeffreys' prior in this analysis.

The PROC GENMOD statements produce Output 52.3.7, with estimates very similar to those reported in Output 52.3.6. Note that you should not expect to see identical output from PROC GENMOD and PROC MCMC, even with the simulation setup and identical random number seed. The two procedures use different sampling algorithms. PROC GENMOD uses the adaptive rejection metropolis algorithm (ARMS) (Gilks and Wild 1992; Gilks 2003) while PROC MCMC uses a random walk Metropolis algorithm. The asymptotic answers, which means that you let both procedures run an very long time, would be the same as they both generate samples from the same posterior distribution.

Output 52.3.7 PROC GENMOD Results

Logistic Regression Model with Jeffreys Prior

The GENMOD Procedure

Bayesian Analysis

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
Intercept	10000	-2.8731	1.3088	-3.6754	-2.7248	-1.9253
lvol	10000	5.1639	1.8087	3.8451	4.9475	6.2613
lrate	10000	4.5501	1.8071	3.2250	4.3564	5.6810

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
Intercept	0.050	-5.8246	-0.7271	-5.5774	-0.6060
lvol	0.050	2.1844	9.2297	2.0112	8.9149
lrate	0.050	1.5666	8.6145	1.3155	8.1922

Poisson Regression

You can use the Poisson distribution to model the distribution of cell counts in a multiway contingency table. Aitkin et al. (1989) have used this method to model insurance claims data. Suppose the following hypothetical insurance claims data are classified by two factors: age group (with two levels) and car type (with three levels). The following statements create the data set:

```
title 'Poisson Regression';
data insure;
   input n c car $ age;
   ln = log(n);
   if car = 'large' then
      do car_dummy1=1;
         car_dummy2=0;
      end;
   else if car = 'medium' then
      do car_dummy1=0;
         car_dummy2=1;
      end;
   else
      do car_dummy1=0;
         car_dummy2=0;
      end;

   datalines;
   500   42  small  0
   1200  37  medium 0
```

```
100    1  large  0
400  101  small  1
500   73  medium 1
300   14  large  1
;
```

The variable n represents the number of insurance policy holders and the variable c represents the number of insurance claims. The variable car is the type of car involved (classified into three groups), and it is coded into two levels. The variable age is the age group of a policy holder (classified into two groups).

Assume that the number of claims c has a Poisson probability distribution and that its mean, μ_i, is related to the factors car and age for observation i by

$$\begin{aligned}\log(\mu_i) &= \log(\mathrm{n}_i) + \mathbf{x}'\boldsymbol{\beta} \\ &= \log(\mathrm{n}_i) + \beta_0 + \\ &\quad \mathrm{car}_i(1)\beta_1 + \mathrm{car}_i(2)\beta_2 + \mathrm{car}_i(3)\beta_3 + \\ &\quad \mathrm{age}_i(1)\beta_4 + \mathrm{age}_i(2)\beta_5\end{aligned}$$

The indicator variables $\mathrm{car}_i(j)$ is associated with the jth level of the variable car for observation i in the following way:

$$\mathrm{car}_i(j) = \begin{cases} 1 & \text{if car} = j \\ 0 & \text{if car} \neq j \end{cases}$$

A similar coding applies to age. The β's are parameters. The logarithm of the variable n is used as an offset—that is, a regression variable with a constant coefficient of 1 for each observation. Having the offset constant in the model is equivalent to fitting an expanded data set with 3000 observations, each with response variable y observed on an individual level. The log link is used to relate the mean and the factors car and age.

The following statements run PROC MCMC:

```
proc mcmc data=insure outpost=insureout nmc=5000 propcov=quanew
          maxtune=0 seed=7;
   ods select PostSummaries PostIntervals;
   parms alpha 0 beta_car1 0 beta_car2 0 beta_age 0;
   prior alpha beta: ~ normal(0, prec = 1e-6);

   mu = ln + alpha + beta_car1 * car_dummy1
      + beta_car2 * car_dummy2 + beta_age * age;
   model c ~ poisson(exp(mu));
run;
```

The analysis uses a relatively flat prior on all the regression coefficients, with mean at 0 and precision at 10^{-6}. The option MAXTUNE=0 skips the tuning phase because the optimization routine (PROPCOV=QUANEW) provides good initial values and proposal covariance matrix.

There are four parameters in the model: alpha is the intercept; beta_car1 and beta_car2 are coefficients for the class variable car, which has three levels; and beta_age is the coefficient for age. The symbol mu connects the regression model and the Poisson mean by using the log link. The MODEL statement specifies a Poisson likelihood for the response variable c.

Posterior summary and interval statistics are shown in Output 52.3.8.

Output 52.3.8 MCMC Results

Poisson Regression

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
alpha	5000	−2.6403	0.1344	−2.7261	−2.6387	−2.5531
beta_car1	5000	−1.8335	0.2917	−2.0243	−1.8179	−1.6302
beta_car2	5000	−0.6931	0.1255	−0.7775	−0.6867	−0.6118
beta_age	5000	1.3151	0.1386	1.2153	1.3146	1.4094

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
alpha	0.050	−2.9201	−2.3837	−2.9133	−2.3831
beta_car1	0.050	−2.4579	−1.3036	−2.4692	−1.3336
beta_car2	0.050	−0.9462	−0.4498	−0.9485	−0.4589
beta_age	0.050	1.0442	1.5898	1.0387	1.5812

To fit the same model by using PROC GENMOD, you can do the following. Note that the default normal prior on the coefficients β is $N(0, \text{prec} = 1e-6)$, the same as used in the PROC MCMC. The following statements run PROC GENMOD and create Output 52.3.9:

```
proc genmod data=insure;
   ods select PostSummaries PostIntervals;
   class car age(descending);
   model c = car age / dist=poisson link=log offset=ln;
   bayes seed=17 nmc=5000 coeffprior=normal;
run;
```

To compare, posterior summary and interval statistics from PROC GENMOD are reported in Output 52.3.9, and they are very similar to PROC MCMC results in Output 52.3.8.

Output 52.3.9 PROC GENMOD Results

Poisson Regression

The GENMOD Procedure

Bayesian Analysis

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
Intercept	5000	-2.6353	0.1299	-2.7243	-2.6312	-2.5455
carlarge	5000	-1.7996	0.2752	-1.9824	-1.7865	-1.6139
carmedium	5000	-0.6977	0.1269	-0.7845	-0.6970	-0.6141
age1	5000	1.3148	0.1348	1.2237	1.3138	1.4067

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
Intercept	0.050	-2.8952	-2.3867	-2.8755	-2.3730
carlarge	0.050	-2.3538	-1.2789	-2.3424	-1.2691
carmedium	0.050	-0.9494	-0.4487	-0.9317	-0.4337
age1	0.050	1.0521	1.5794	1.0624	1.5863

Note that the descending option in the CLASS statement reverses the sorting order of the class variable age so that the results agree with PROC MCMC. If this option is not used, the estimate for age has a reversed sign as compared to Output 52.3.9.

Example 52.4: Nonlinear Poisson Regression Models

This example illustrates how to fit a nonlinear Poisson regression with PROC MCMC. In addition, it shows how you can improve the mixing of the Markov chain by selecting a different proposal distribution or by sampling on the transformed scale of a parameter. This example shows how to analyze count data for calls to a technical support help line in the weeks immediately following a product release. This information could be used to decide upon the allocation of technical support resources for new products. You can model the number of daily calls as a Poisson random variable, with the average number of calls modeled as a nonlinear function of the number of weeks that have elapsed since the product's release. The data are input into a SAS data set as follows:

```
title 'Nonlinear Poisson Regression';
data calls;
   input weeks calls @@;
   datalines;
1   0   1   2   2   2   2   1   3   1   3   3
4   5   4   8   5   5   5   9   6  17   6   9
7  24   7  16   8  23   8  27
;
```

During the first several weeks after a new product is released, the number of questions that technical support receives concerning the product increases in a sigmoidal fashion. The expression for the mean value in the classic Poisson regression involves the log link. There is some theoretical justification for this link, but with MCMC methodologies, you are not constrained to exploring only models that are computationally convenient. The number of calls to technical support tapers off after the initial release, so in this example you can use a logistic-type function to model the mean number of calls received weekly for the time period immediately following the initial release. The mean function $\lambda(t)$ is modeled as follows:

$$\lambda_i = \frac{\gamma}{1 + \exp\left[-(\alpha + \beta t_i)\right]}$$

The likelihood for every observation calls_i is

$$\text{calls}_i \sim \text{Poisson}\left(\lambda_i\right)$$

Past experience with technical support data for similar products suggests using a gamma distribution with shape and scale parameters 3.5 and 12 as the prior distribution for γ, a normal distribution with mean -5 and variance 0.25 as the prior for α, and a normal distribution with mean 0.75 and variance 0.5 as the prior for β.

The following PROC MCMC statements fit this model:

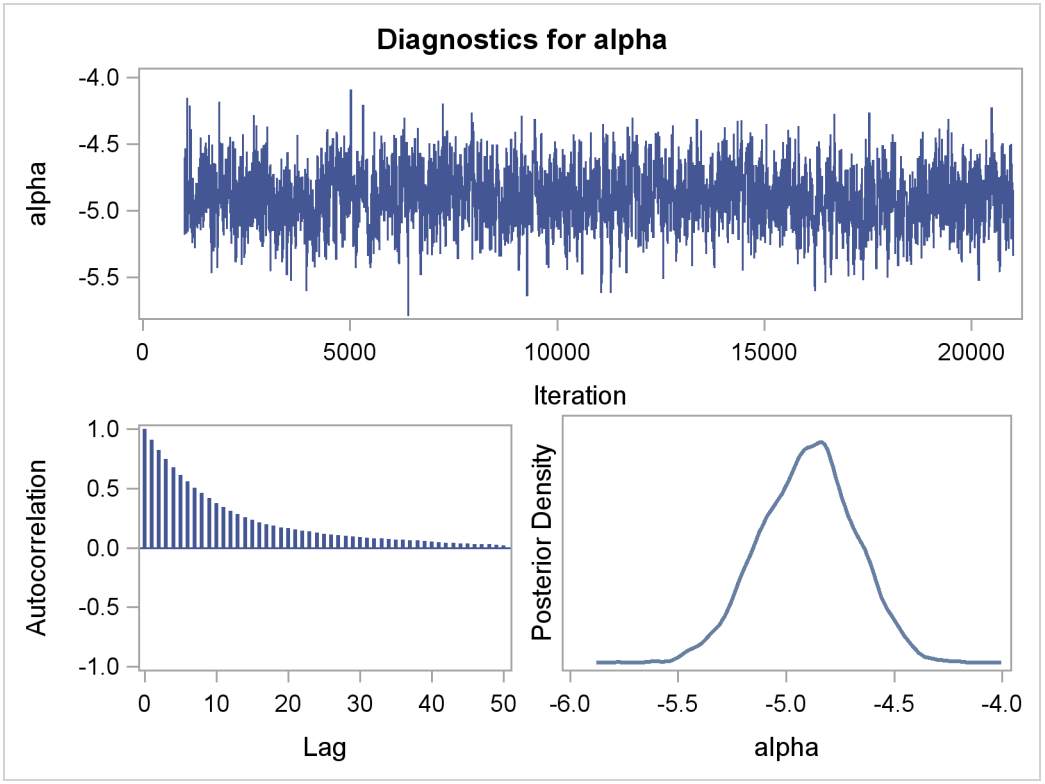
```
ods graphics on;
proc mcmc data=calls outpost=callout seed=53197 ntu=1000 nmc=20000
          propcov=quanew;
   ods select TADpanel;
   parms alpha -4 beta 1 gamma 2;
   prior alpha ~ normal(-5, sd=0.25);
   prior beta  ~ normal(0.75, sd=0.5);
   prior gamma ~ gamma(3.5, scale=12);
   lambda = gamma*logistic(alpha+beta*weeks);
   model calls ~ poisson(lambda);
run;
```

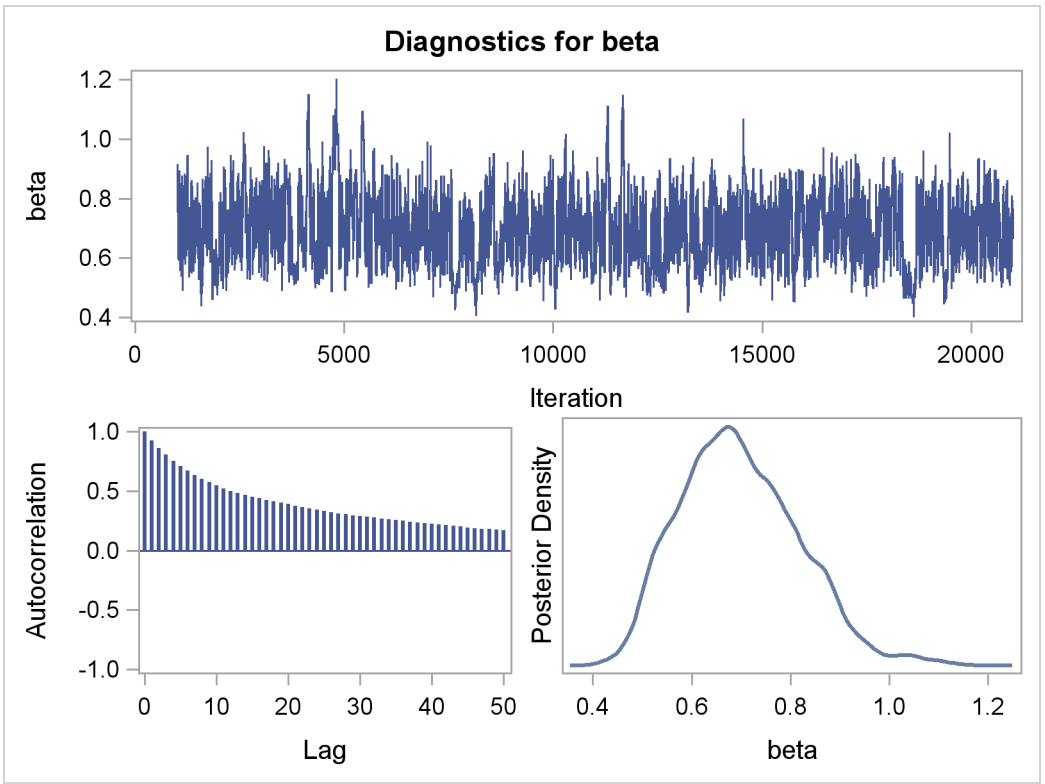
The one PARMS statement defines a block of all parameters and sets their initial values individually. The PRIOR statements specify the informative prior distributions for the three parameters. The assignment statement defines λ, the mean number of calls. Instead of using the SAS function LOGISTIC, you can use the following statement to calculate λ and get the same result:

```
lambda = gamma / (1 + exp(-(alpha+beta*weeks)));
```

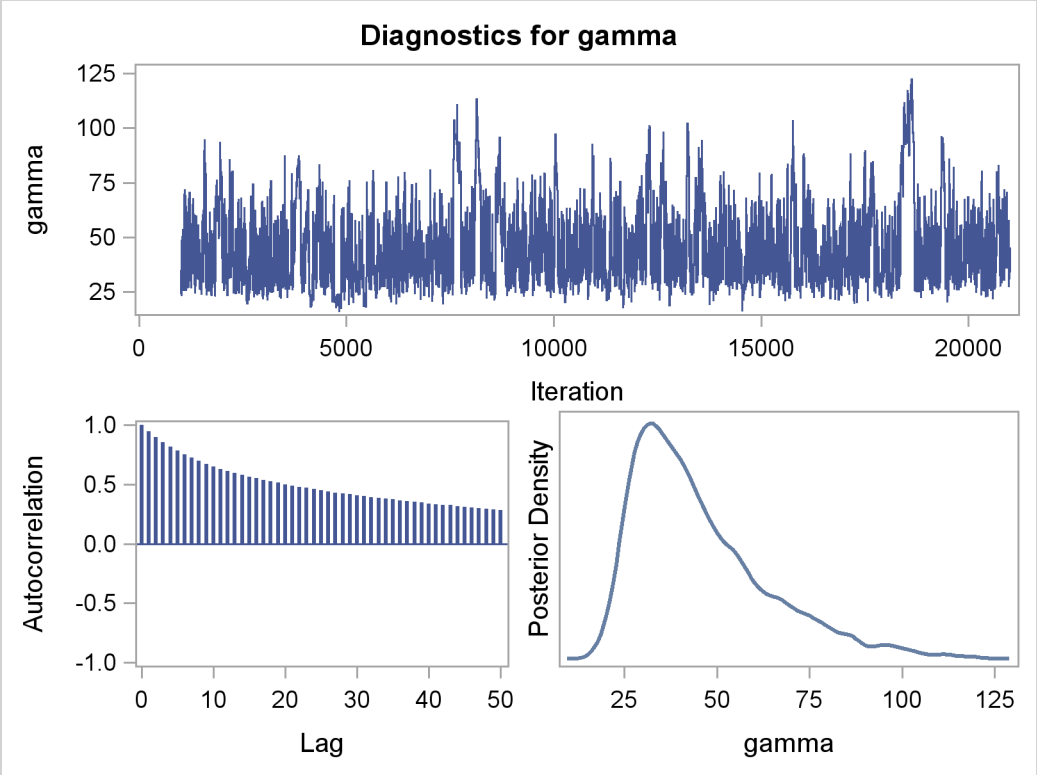
Mixing is not particularly good with this run of PROC MCMC. The ODS SELECT statement displays only the diagnostic graphs while excluding all other output. The graphical output is shown in Output 52.4.1.

Output 52.4.1 Plots for Parameters





Output 52.4.1 *continued*



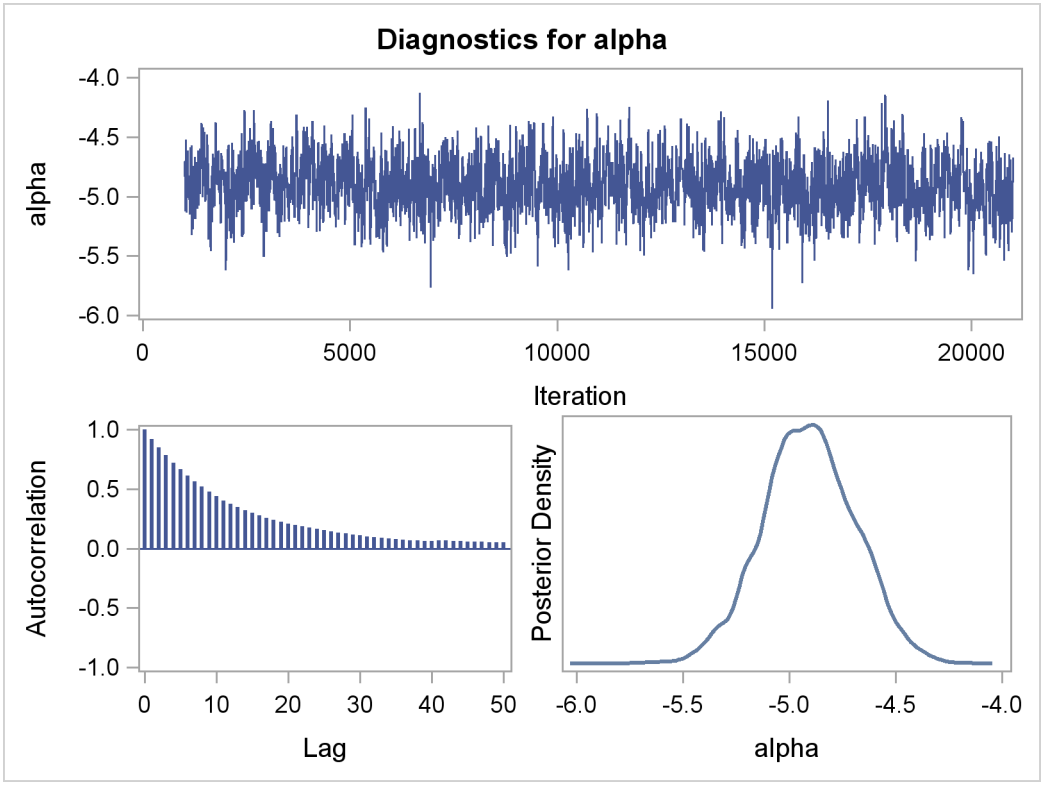
By examining the trace plot of the gamma parameter, you see that the Markov chain sometimes gets stuck in the far right tail and does not travel back to the high density area quickly. This effect can be seen around the simulations number 8000 and 18000. One possible explanation for this is that the random walk Metropolis is taking too small of steps in its proposal; therefore it takes more iterations for the Markov chain to explore the parameter space effectively. The step size in the random walk is controlled by the normal proposal distribution (with a multiplicative scale). A (good) proposal distribution is roughly an approximation to the joint posterior distribution at the mode. The curvature of the normal proposal distribution (the variance) does not take into account the thickness of the tail areas. As a result, a random walk Metropolis with normal proposal can have a hard time exploring distributions that have thick tails. This appears to be the case with the posterior distribution of the parameter gamma. You can improve the mixing by using a thicker-tailed proposal distribution, the *t*-distribution. The option PROPDIST controls the proposal distribution. PROPDIST=T(3) changes the proposal from a normal distribution to a *t*-distribution with three degrees of freedom.

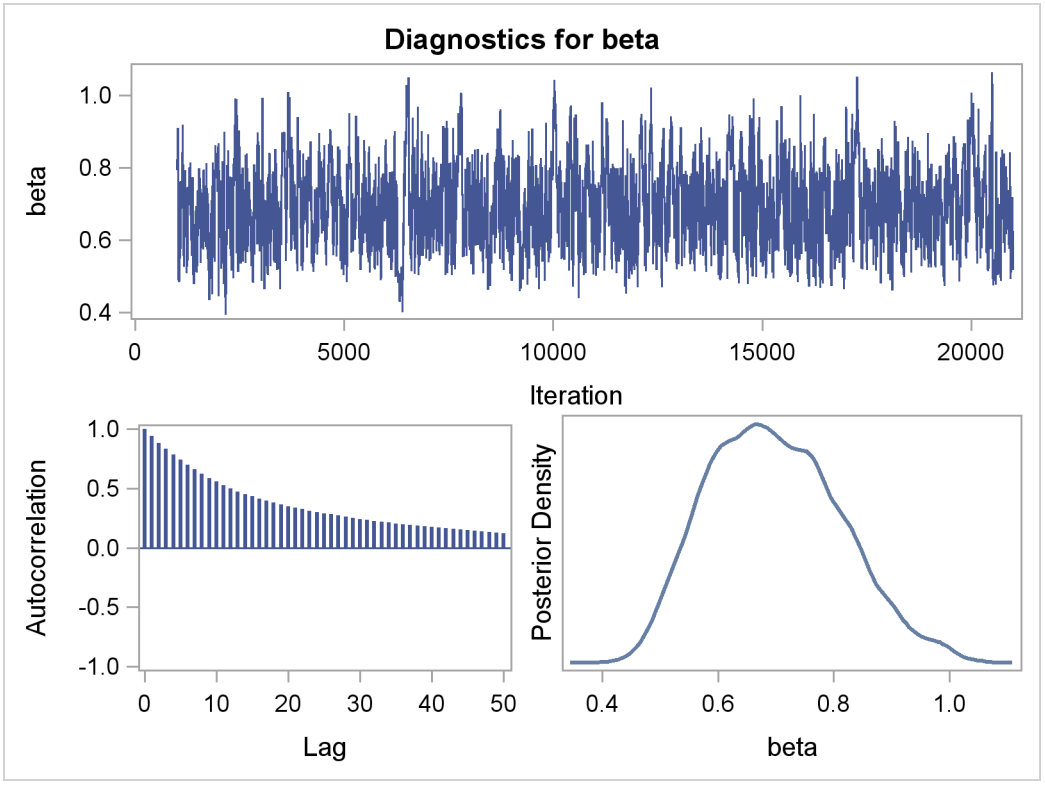
The following statements run PROC MCMC and produce Output 52.4.2:

```
proc mcmc data=calls outpost=callout seed=53197 ntu=1000 nmc=20000
          propcov=quanew stats=none propdist=t(3);
   ods select TADpanel;
   parms alpha -4 beta 1 gamma 2;
   prior alpha ~ normal(-5, sd=0.25);
   prior beta  ~ normal(0.75, sd=0.5);
   prior gamma ~ gamma(3.5, scale=12);
   lambda = gamma*logistic(alpha+beta*weeks);
   model calls ~ poisson(lambda);
run;
```

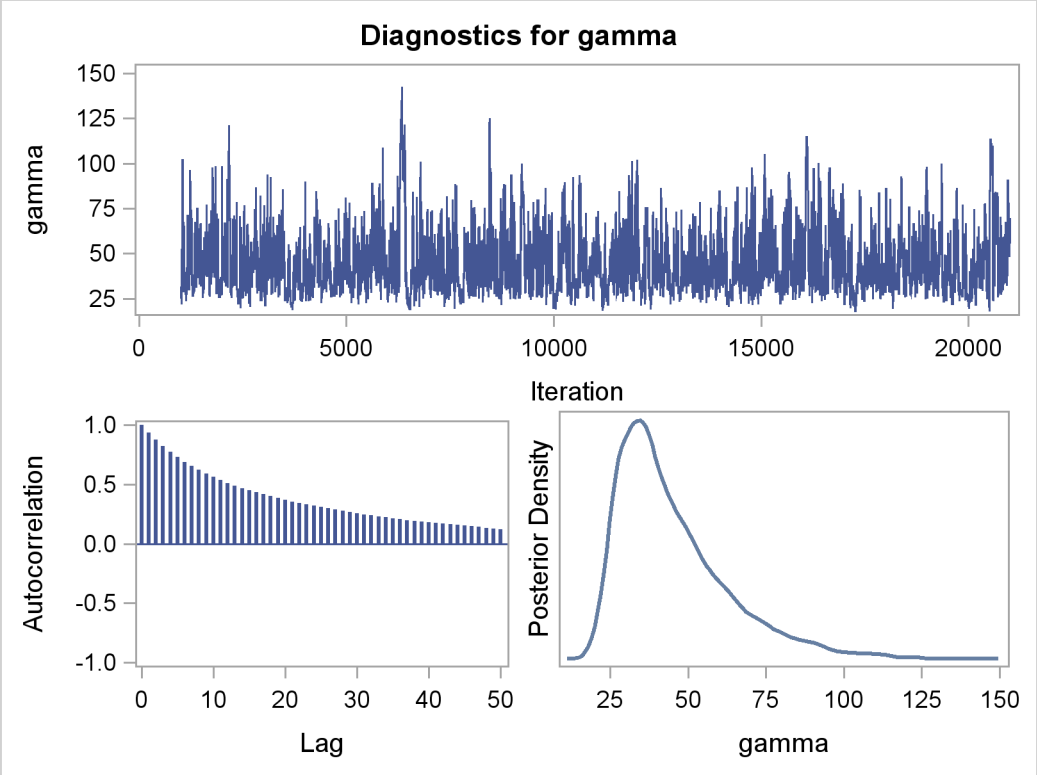
Output 52.4.2 displays the graphical output.

Output 52.4.2 Plots for Parameters, Using a t(3) Proposal Distribution





Output 52.4.2 *continued*



The trace plots are more dense and the ACF plots have faster drop-offs, and you see improved mixing by using a thicker-tailed proposal distribution. If you want to further improve the Markov chain, you can choose to sample the log transformation of the parameter gamma:

$$\text{lg} \sim \text{egamma}(3.5, \text{scale} = 12) \quad \text{is equivalent to} \quad gamma = \exp(\text{lg}) \sim \text{gamma}(3.5, \text{scale} = 12)$$

The parameter gamma has a positive support. Often in this case, it has right-skewed posterior. By taking the log transformation, you can sample on a parameter space that does not have a lower boundary and is more symmetric. This can lead to better mixing.

The following statements produce Output 52.4.4 and Output 52.4.3:

```
proc mcmc data=calls outpost=callout seed=53197 ntu=1000 nmc=20000
          propcov=quanew propdist=t(3)
          monitor=(alpha beta lgamma gamma);
   ods select PostSummaries PostIntervals TADpanel;
   parms alpha -4 beta 1 lgamma 2;
   prior alpha ~ normal(-5, sd=0.25);
   prior beta  ~ normal(0.75, sd=0.5);
   prior lgamma ~ egamma(3.5, scale=12);
   gamma = exp(lgamma);
   lambda = gamma*logistic(alpha+beta*weeks);
   model calls ~ poisson(lambda);
run;
ods graphics off;
```

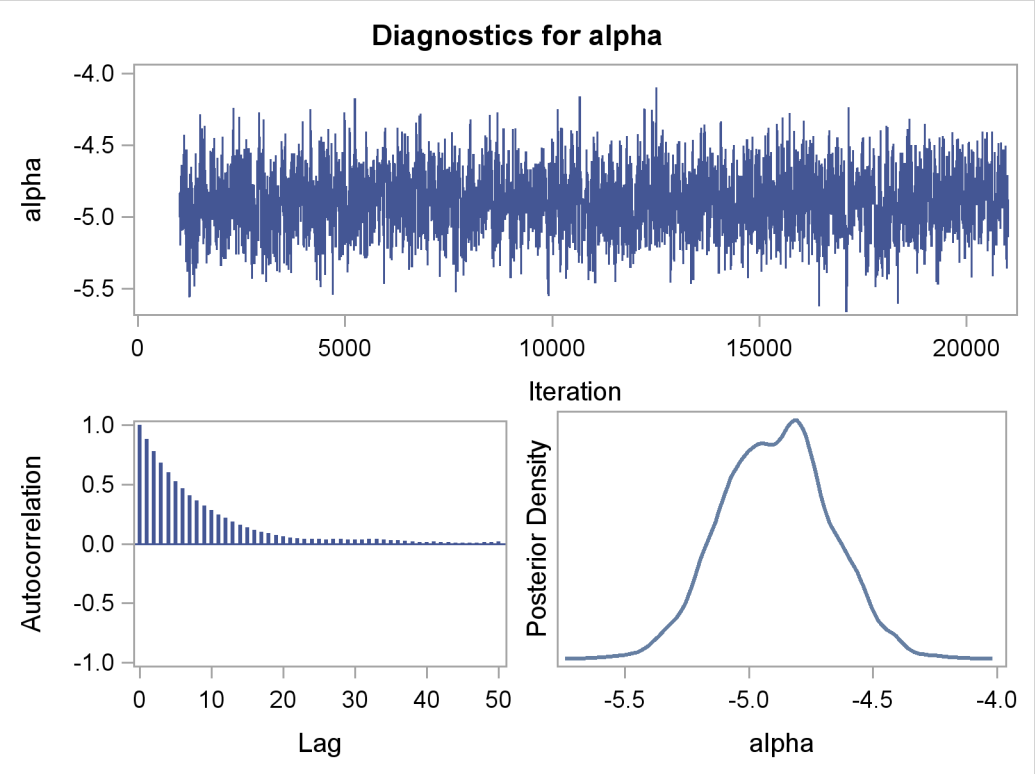
In the PARMS statement, instead of gamma, you have lgamma. Its prior distribution is egamma, as opposed to the gamma distribution. Note that the following two priors are equivalent to each other:

```
prior lgamma ~ egamma(3.5, scale=12);
prior gamma ~ gamma(3.5, scale=12);
```

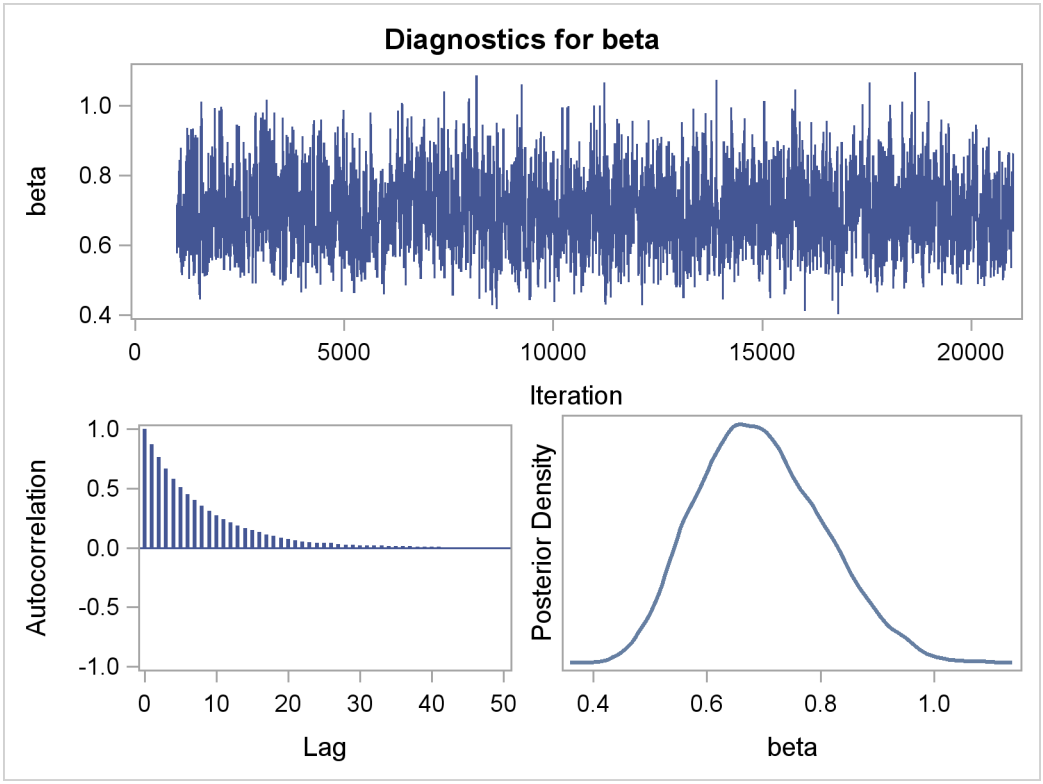
The gamma assignment statement transforms lgamma to gamma. The lambda assignment statement calculates the mean for the Poisson by using the gamma parameter. The MODEL statement specifies a Poisson likelihood for the calls response.

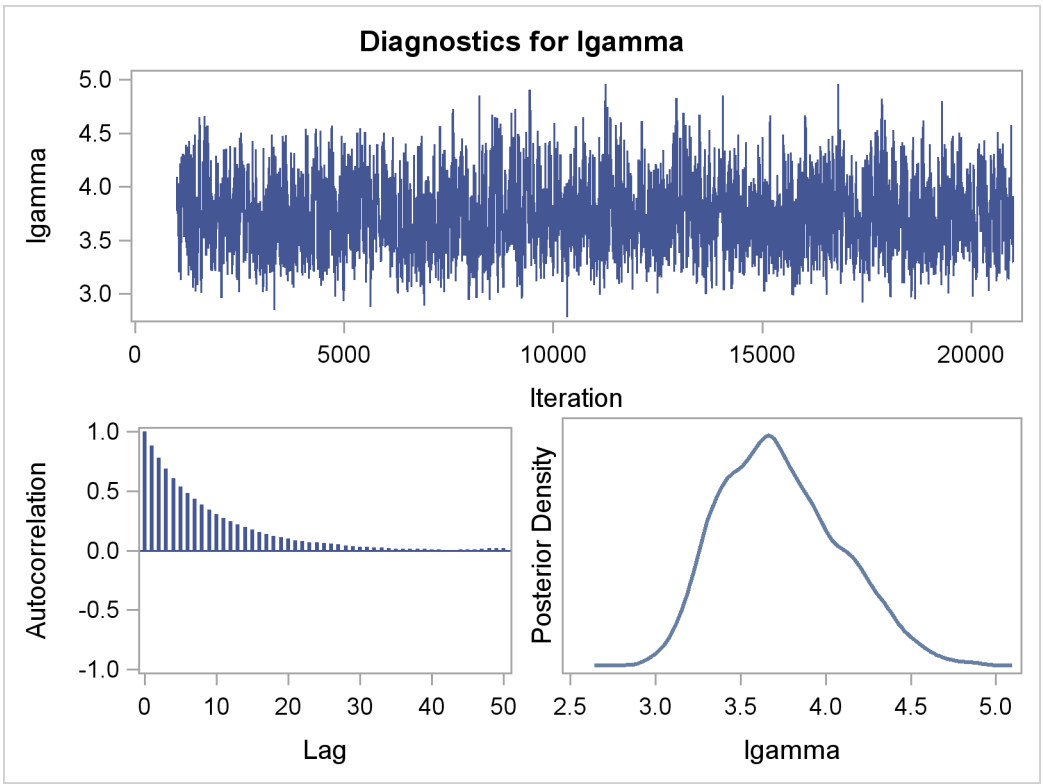
The trace plots and ACF plots in Output 52.4.3 show the best mixing seen so far in this example.

Output 52.4.3 Plots for Parameters, Sampling on the Log Scale of Gamma

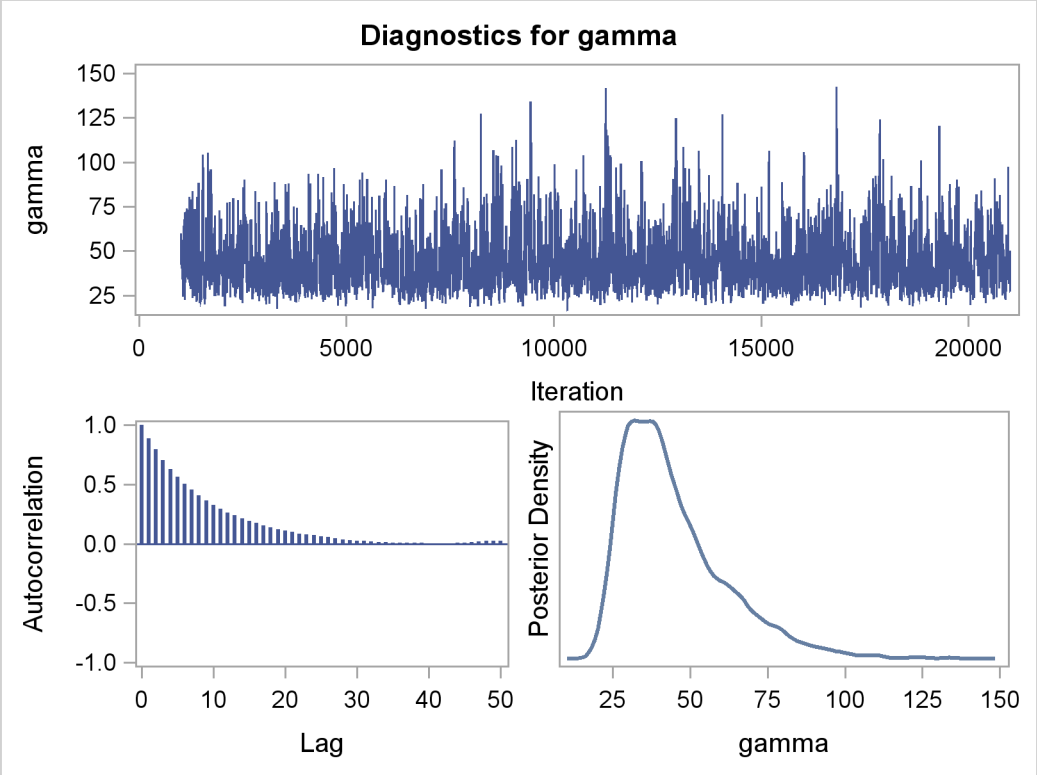


Output 52.4.3 *continued*





Output 52.4.3 *continued*



Output 52.4.4 shows the posterior summary statistics of the nonlinear Poisson regression. Note that the lgamma parameter has a more symmetric density than the skewed gamma parameter. The Metropolis algorithm always works better if the target distribution is approximately normal.

Output 52.4.4 MCMC Results, Sampling on the Log Scale of Gamma

Nonlinear Poisson Regression

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
alpha	20000	-4.8907	0.2160	-5.0435	-4.8872	-4.7461
beta	20000	0.6957	0.1089	0.6163	0.6881	0.7698
lgamma	20000	3.7391	0.3487	3.4728	3.7023	3.9696
gamma	20000	44.8136	17.0430	32.2263	40.5415	52.9647

Output 52.4.4 *continued*

Posterior Intervals					
Parameter	Alpha	Equal-Tail Interval		HPD Interval	
alpha	0.050	-5.3138	-4.4667	-5.3276	-4.4953
beta	0.050	0.5066	0.9253	0.4868	0.8996
lgamma	0.050	3.1580	4.4705	3.1222	4.4127
gamma	0.050	23.5225	87.3972	20.9005	79.4712

This example illustrates that PROC MCMC can fit Bayesian nonlinear models just as easily as Bayesian linear models. More importantly, transformations can sometimes improve the efficiency of the Markov chain, and that is something to always keep in mind. Also see "Example 52.12: Using a Transformation to Improve Mixing" on page 3683 for another example of how transformations can improve mixing of the Markov chains.

Example 52.5: Random-Effects Models

This example illustrates how you can use PROC MCMC to fit random effects models. In the example "Mixed-Effects Model" on page 3492 in "Getting Started: MCMC Procedure" on page 3479, you already saw PROC MCMC fit a linear random effects model. There are two more examples in this section. One is a logistic random effects model, and the second one is a nonlinear Poisson regression random effects model. In addition, this section illustrates how to construct prior distributions that depend on input data set variables. Such prior distributions appear frequently in random effects model, especially in cases of hierarchical centering. Although you can use PROC MCMC to analyze random effects models, you might want to first consider some other SAS procedures. For example, you can use PROC MIXED (see Chapter 56, "The MIXED Procedure") to analyze linear mixed effects models, PROC NLMIXED (see Chapter 61, "The NLMIXED Procedure") for nonlinear mixed effects models, and PROC GLIMMIX (see Chapter 38, "The GLIMMIX Procedure") for generalized linear mixed effects models. In addition, a sampling-based Bayesian analysis is available in the MIXED procedure through the PRIOR statement (see "PRIOR Statement" on page 3945).

Logistic Regression Random-Effects Model

This example shows how to fit a logistic random-effects model in PROC MCMC. The data are taken from Crowder (1978). The seeds data set is a 2×2 factorial layout, with two types of seeds, *O. aegyptiaca 75* and *O. aegyptiaca 73*, and two root extracts, *bean* and *cucumber*. You observe r, which is the number of germinated seeds, and n, which is the total number of seeds. The independent variables are seed and extract.

The following statements create the data set:

```
title 'Logistic Regression Random-Effects Model';
data seeds;
   input r n seed extract @@;
   ind = _N_;
   datalines;
10  39  0  0    23  62  0  0    23  81  0  0    26  51  0  0
17  39  0  0     5   6  0  1    53  74  0  1    55  72  0  1
32  51  0  1    46  79  0  1    10  13  0  1     8  16  1  0
10  30  1  0     8  28  1  0    23  45  1  0     0   4  1  0
 3  12  1  1    22  41  1  1    15  30  1  1    32  51  1  1
 3   7  1  1
;
```

You can model each observation r_i as having its own probability of success p_i, and the likelihood is as follows:

$$\mathrm{r}_i \sim \text{binomial}(\mathrm{n}_i, p_i)$$

You can use the logit link function to link the covariates of each observation, seed and extract, to the probability of success:

$$\begin{aligned} \mu_i &= \beta_0 + \beta_1 \cdot \text{seed}_i + \beta_2 \cdot \text{extract}_i + \beta_3 \cdot \text{seed}_i \cdot \text{extract}_i \\ p_i &= \text{logistic}(\mu_i + \eta_i) \end{aligned}$$

where η_i is assumed to be as i.i.d. random effect with a normal prior:

$$\eta_i \sim \text{normal}(0, \text{var} = \sigma^2)$$

The four β regression coefficients and the standard deviation σ^2 in the random effects are model parameters; they are given noninformative priors as follows:

$$\begin{aligned} \pi(\beta_0, \beta_1, \beta_2, \beta_3) &\propto 1 \\ \pi(\sigma^2) &\propto 1/\sigma^2 \end{aligned}$$

Another way of expressing the same model is as follows:

$$p_i = \text{logistic}(\delta_i)$$

where

$$\delta_i \sim \text{normal}(\beta_0 + \beta_1 \cdot \text{seed}_i + \beta_2 \cdot \text{extract}_i + \beta_3 \cdot \text{seed}_i \cdot \text{extract}_i, \sigma^2)$$

The two models are equivalent. In the first model, the random effects η_i centers at 0 in the normal distribution, and in the second model, δ_i centers at the regression mean. This hierarchical centering can sometimes improve mixing.

From a programming point of view, the second parameterization of the model is more difficult because the prior distribution on δ_i involves the data set variables seed and extract. Each prior distribution depends on a different set of observations in the input data set. Intuitively, you might think that the following statements would specify such a prior:

```
mu = beta0 + beta1*seed + beta2*extract + beta3*seed*extract;
prior delta ~ normal(mu, var = v);
```

However, this will not work. This is because the procedure is not able to match the observational level calculation (mu) with elements of a parameter array (there are 21 random effects in delta). Thus, the procedure cannot calculate the log of the prior density correctly. The solution is to cumulatively calculate the joint prior distribution for all $\delta_i, i = 1 \cdots 21$, and assign the prior distribution to all δ by using the GENERAL function.

The following statements generate Output 52.5.1:

```
proc mcmc data=seeds outpost=postout seed=332786 nmc=100000 thin=10
          ntu=3000 monitor=(beta0-beta3 v);
   ods select PostSummaries ess;
   array delta[21];

   parms delta: 0;
   parms beta0 0 beta1 0 beta2 0 beta3 0 ;
   parms v 1;

   beginnodata;
   sigma = sqrt(v);
   endnodata;
   w = beta0 + beta1*seed + beta2*extract + beta3*seed*extract;
   if ind eq 1 then
      lp = lpdfnorm(delta[ind], w, sigma);
   else
      lp = lp + lpdfnorm(delta[ind], w, sigma);

   prior v   ~ general(-log(v));
   prior beta: ~ general(0);
   prior delta: ~ general(lp);

   pi = logistic(delta[ind]);
   model r ~ binomial(n = n, p = pi);
run;
```

PROC MCMC statement specifies the input and output data sets, sets a seed for the random number generator, requests a very large simulation number, thins the Markov chain by 10, and specifies a tuning sample size of 3000. The MONITOR= option selects the parameters of interest. The ODS SELECT statement displays the summary statistics and effective sample size tables.

The ARRAY statement allocates an array of size 21 for the random effects parameter δ. There are three PARMS statements that place δ, β and σ^2 into three sampling blocks. Calculation of sigma does not involve any observations; hence, it is enclosed in the BEGINNODATA and ENDNODATA statements.

The next few lines of statements construct a joint prior distribution for all the δ parameters. The symbol w is the regression mean, whose value changes for every observation. The IF-ELSE statements add the log of the normal density to the symbol lp as PROC MCMC steps through the data set. When ind is 1, lp is the log of the normal density for delta[1] evaluated at the first regression mean w. As ind gradually increases to 21, lp becomes

$$\sum_i \log(\phi(\delta_i|\beta X_i, \sigma))$$

which is the joint prior distribution for all δ.

The PRIOR statements assign three priors to these parameters, with noninformative priors on σ^2 and β. All of the delta parameters share a joint prior, which is defined by lp. Recall that PROC MCMC adds the log of the prior density to the log of the posterior density at the last observation at every simulation, so the expression lp will have the correct value.

CAUTION: You must define the expression lp before the PRIOR statement for the delta parameters. Switching the order of the PRIOR statement and the programming statements that define lp leads to an incorrect prior distribution for delta. The following statements are wrong because the expression lp has not completed its calculation when lp is added to the log of the posterior density at the last observation of the input data set.

```
prior delta: ~ general(lp);
w = beta0 + beta1*seed + beta2*extract + beta3*seed*extract;
if ind eq 1 then
   lp = lpdfnorm(delta[ind], w, sigma);
else
   lp = lp + lpdfnorm(delta[ind], w, sigma);
```

The prior you specify in this case is:

$$\sum_{i=1}^{n-1} \log(\phi(\delta_i|\beta X_i, \sigma))$$

The correct log density is the following:

$$\sum_{i=1}^{n} \log(\phi(\delta_i|\beta X_i, \sigma))$$

The symbol pi is the logit transformation. The MODEL specifies the response variable r as a binomial distribution with parameters n and pi.

The mixing is poor in this example. You can see from the effective sample size table (Output 52.5.1) that the efficiency for all parameters is relatively low, even after a substantial amount of thinning. One possible solution is to break the random effects block of parameters (b) into multiple blocks with a smaller number of parameters.

Output 52.5.1 Logistic Regression Random-Effects Model

Logistic Regression Random-Effects Model

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
beta0	10000	−0.5503	0.2025	−0.6784	−0.5522	−0.4193
beta1	10000	0.0626	0.3292	−0.1512	0.0653	0.2760
beta2	10000	1.3546	0.2876	1.1732	1.3391	1.5349
beta3	10000	−0.8257	0.4498	−1.1044	−0.8255	−0.5344
v	10000	0.1145	0.1019	0.0472	0.0875	0.1503

Logistic Regression Random-Effects Model

The MCMC Procedure

Effective Sample Sizes

Parameter	ESS	Correlation Time	Efficiency
beta0	885.3	11.2952	0.0885
beta1	603.2	16.5771	0.0603
beta2	854.9	11.6970	0.0855
beta3	591.6	16.9021	0.0592
v	273.1	36.6182	0.0273

To fit the same model in PROC GLIMMIX, you can use the following statements, which produce Output 52.5.2:

```
proc glimmix data=seeds method=quad;
   ods select covparms parameterestimates;
   ods output covparms=cp parameterestimates=ps;
  class ind;
  model r/n = seed extract seed*extract/ dist=binomial link=logit solution;
  random intercept / subject=ind;
run;
```

Output 52.5.2 Estimates by PROC GLMMIX

Logistic Regression Random-Effects Model

The GLIMMIX Procedure

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error
Intercept	ind	0.05577	0.05196

Solutions for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	−0.5484	0.1666	17	−3.29	0.0043
seed	0.09701	0.2780	0	0.35	.
extract	1.3370	0.2369	0	5.64	.
seed*extract	−0.8104	0.3851	0	−2.10	.

It is hard to compare point estimates from these two procedures. However, you can visually compare the results by plotting a kernel density plot (by using the posterior sample from PROC MCMC output) on top of a normal approximation plot (by using the mean and standard error estimates from PROC GLIMMIX, for each parameter). This kernel comparison plot is shown in Output 52.5.3.

However, it takes some work to produce the kernel comparison plot. First, you must use PROC KDE to estimate the kernel density for each parameter from MCMC. Next, you want to get the point estimates from the PROC GLIMMIX output. Then, you generate a SAS data set that contains both the kernel density estimates and the gridded estimates based on normal approximations. Finally, you use PROC TEMPLATE (see Chapter 21, "Statistical Graphics Using ODS") to define an appropriate graphical template and produce the comparison plot by using PROC SGRENDER (see the SGRENDER Procedure in the *SAS/GRAPH: Statistical Graphics Procedures Guide*).

The following statements use PROC KDE on the posterior sample data set postout and estimate a kernel density for each parameter, saving the estimates to a SAS data set m1:

```
proc kde data=postout;
   univar beta0 beta1 beta2 beta3 v / out=m1 (drop=count);
run;
```

The following SAS statements take the estimates of all the parameters from the PROC GLIMMIX output, data sets ps and cp, and assign them to macro variables:

```
data gmxest(keep = parm mean sd);
   set ps cp;
   mean = estimate;
   sd = stderr;
   i = _n_-1;
   if(_n_ ne 5) then
      parm = "beta" || put(i, z1.);
```

```
   else
      parm = "var";
run;

data msd (keep=mean sd);
   set gmxest;
   do j = 1 to 401;
      output;
   end;
run;

data _null_;
   set ps;
   call symputx(compress(effect,'*'), estimate);
   call symputx(compress('s' || effect,'*'), stderr);
run;

data _null_;
   set cp;
   call symputx("var", estimate);
   call symputx("var_sd", stderr);
run;

%put &intercept &seed &extract &seedextract &var;
%put &sintercept &sseed &sextract &sseedextract &var_sd;
```

Specifically, the mean estimate of β_0 is assigned to intercept, and the standard error of β_0 is assigned to sintercept. The macro variables seed, extract, seedextract are the mean estimates for β_1, β_2 and β_3, respectively.

To create a SAS data set that contains both the kernel density estimates and the corresponding normal approximation, you can use the %REN and %RESHAPE macros. The %REN macro renames the variables of a SAS data set by appending the suffix name to each variable name, to avoid redundant variable names. The %RESHAPE macro takes an output data set from a PROC KDE run, and transposes it to the right format so that PROC SGRENDER can generate the right graph. The following statements define the %REN and %RESHAPE macros:

```
/* define macros */
%macro ren(in=, out=, suffix=);
   %local s;

   proc contents data=&in noprint out=__temp__(keep=name);
   run;

   data _null_;
      length s $ 32000;
      retain s;
      set __temp__ end=eof;
      s = trim(s)||' '||trim(name)||'='||compress(name||"&suffix");
      if eof then call symput('s', trim(s));
   run;

   proc datasets nolist;
```

```
      delete __temp__;
   run; quit;

   data &out;
      set &in(rename=(&s));
   run;
%mend;

%macro reshape(input, output, suffix1=, suffix2=);
   proc sort data=&input;
      by var;
   run;

   data tmp&input;
      set &input;
      by var;
      _n + 1;
      if first.var then _n = 0;
   run;

   proc sort;
      by _n var;
   run;

   proc transpose data=tmp&input out=_by_value_(drop=_n _name_ _label_);
      var value;
      by _n;
      id var;
   run;

   %ren(in=_by_value_, out=_by_value_, suffix=&suffix1)

   proc transpose data=tmp&input out=_by_den_(drop=_n _name_ _label_);
      var density;
      by _n;
      id var;
   run;

   %ren(in=_by_den_, out=_by_den_, suffix=&suffix2)

   data &output;
      merge _by_value_ _by_den_;
   run;

   proc datasets library=work;
      ods exclude all;
      delete tmp&input _by_value_ _by_den_;
   run;
   ods exclude none;
%mend;
```

When you apply the %RESHAPE macro to the data set m1, you create a SAS data set mcmc that has grid values of the β parameters and their corresponding kernel density estimates. Next, you evaluate these parameter grid values in a normal density with the macro variables taken from the

PROC GLIMMIX output:

```
/* create data set mcmc */
%reshape(m1, mcmc, suffix1=, suffix2=_kde);
data all;
   set mcmc;
   beta0_gmx = pdf('normal', beta0, &intercept, &sintercept);
   beta1_gmx = pdf('normal', beta1, &seed, &sseed);
   beta2_gmx = pdf('normal', beta2, &extract, &sextract);
   beta3_gmx = pdf('normal', beta3, &seedextract, &sseedextract);
   v_gmx = pdf('normal', v, &var, &var_sd);
run;
```

In the data set all, you have grid values on β and σ^2, their kernel density estimates from PROC MCMC, and the normal density evaluated by using estimates from PROC GLIMMIX. To create an overlaid plot, you first use PROC TEMPLATE to create a 2×3 template as demonstrated by the following statements:

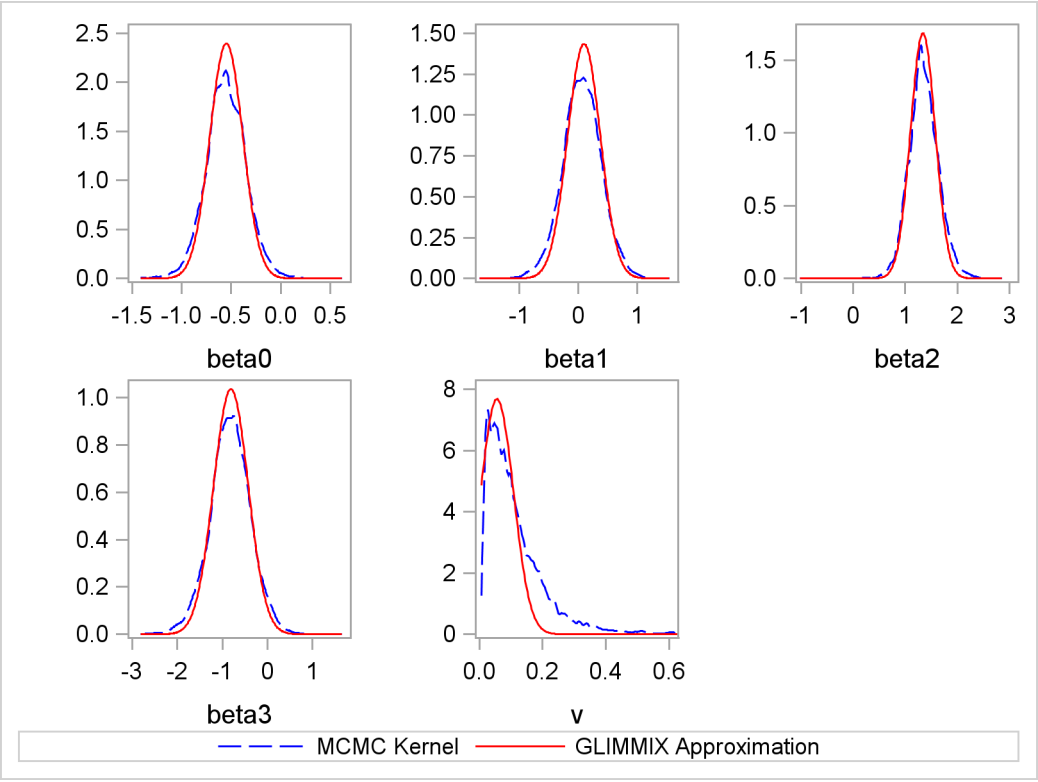
```
proc template;
   define statgraph twobythree;
      %macro plot;
         begingraph;
            layout lattice / rows=2 columns=3;
               %do i = 0 %to 3;
                  layout overlay /yaxisopts=(label=" ");
                     seriesplot y=beta&i._kde x=beta&i
                        / connectorder=xaxis
                          lineattrs=(pattern=mediumdash color=blue)
                        legendlabel = "MCMC Kernel" name="MCMC";
                     seriesplot y=beta&i._gmx x=beta&i
                        / connectorder=xaxis lineattrs=(color=red)
                        legendlabel="GLIMMIX Approximation" name="GLIMMIX";
                  endlayout;
               %end;
               layout overlay /yaxisopts=(label=" ")
                              xaxisopts=(linearopts=(viewmin=0 viewmax=0.6));
                  seriesplot y=v_kde x=v
                     / connectorder=xaxis
                       lineattrs=(pattern=mediumdash color=blue)
                     legendlabel = "MCMC Kernel" name="MCMC";
                  seriesplot y=v_gmx x=v
                     / connectorder=xaxis lineattrs=(color=red)
                     legendlabel="GLIMMIX Approximation" name="GLIMMIX";
               endlayout;
               Sidebar / align = bottom;
                  discretelegend "MCMC" "GLIMMIX";
               endsidebar;
            endlayout;
         endgraph;
      %mend; %plot;
   end;
run;
```

The kernel density comparison plot is produced by calling PROC SGRENDER (see the SGREN-

DER Procedure in the *SAS/GRAPH: Statistical Graphics Procedures Guide*):

```
proc sgrender data=all template=twobythree;
run;
```

Output 52.5.3 Comparing Estimates from PROC MCMC and PROC GLIMMIX.



The kernel densities are very similar to each other. Kernel densities from PROC MCMC are not as smooth, possibly due to bad mixing of the Markov chains.

Nonlinear Poisson Regression Random-Effects Model

This example uses the pump failure data of Gaver and O'Muircheartaigh (1987). The number of failures and the time of operation are recorded for 10 pumps. Each of the pumps is classified into one of two groups corresponding to either continuous or intermittent operation. The following statements generate the data set:

```
title 'Nonlinear Poisson Regression Random Effects Model';
data pump;
   input y t group @@;
   pump = _n_;
   logtstd = log(t) - 2.4564900;
   datalines;
 5  94.320 1   1  15.720 2    5  62.880 1
14 125.760 1   3   5.240 2   19  31.440 1
 1   1.048 2   1   1.048 2    4   2.096 2
22  10.480 2
;
```

Each row denotes data for a single pump, and the variable logtstd contains the centered operation times. Letting y_{ij} denote the number of failures for the jth pump in the ith group, Draper (1996) considers the following hierarchical model for these data:

$$\begin{aligned} y_{ij}|\lambda_{ij} &\sim \text{Poisson}(\lambda_{ij}) \\ \log\lambda_{ij} &= \alpha_i + \beta_i(\log t_{ij} - \overline{\log t}) + e_{ij} \\ e_{ij}|\sigma^2 &\sim \text{normal}(0, \sigma^2) \end{aligned}$$

The model specifies different intercepts and slopes for each group, and the random effect is a mechanism for accounting for over-dispersion. You can use noninformative priors on the parameters α_i, β_i, and σ^2.

$$\begin{aligned} \pi(\alpha_1, \alpha_2, \beta_1, \beta_2) &\propto 1 \\ \pi(\sigma^2) &\propto 1/\sigma^2 \end{aligned}$$

The following statements fit this nonlinear hierarchical model and produce Output 52.5.4:

```
proc mcmc data=pump outpost=postout seed=248601 nmc=100000
          ntu=2000 thin=10
          monitor=(logsig beta1 beta2 alpha1 alpha2 s2 adif bdif);
   ods select PostSummaries;
   array alpha[2];
   array beta[2];
   array llambda[10];

   parms (alpha: beta:) 1;
   parms llambda: 1;
   parms s2 1;

   beginnodata;
   sd = sqrt(s2);
   logsig = log(s2)/2;
   adif = alpha1 - alpha2;
   bdif = beta1 - beta2;
   endnodata;
   w = alpha[group] + beta[group] * logtstd;
   if pump eq 1 then
      lp = lpdfnorm(llambda[pump], w, sd);
   else
      lp = lp + lpdfnorm(llambda[pump], w, sd);
   prior alpha: beta: ~ general(0);
   prior s2 ~ general(-log(s2));
   prior llambda: ~ general(lp);

   lambda = exp(llambda[pump]);
   model y ~ poisson(lambda);
run;
```

The PROC MCMC statement specifies the input data set (pump), the output data set (postout), a seed for the random number generator, and an MCMC sample of 100000. It also requests a tuning

sample size of 2000 and a thinning rate of 10. The MONITOR= option keeps track of a number of parameters and symbols in the model. The five parameters are beta1, beta2, alpha1, alpha2, and s2. The symbol logsig is the log of the standard deviation, adif measures the difference between alpha1 and alpha2, and bdif measures the difference between beta1 and beta2. The ODS SELECT statement displays the summary statistics table.

Modeling the random effects e_{ij} with a normal distribution with mean 0 and variance σ^2 is equivalent to modeling $\log \lambda_{ij}$ with a normal distribution with mean $\alpha_i + \beta_i(\log t_{ij} - \overline{\log t})$ and variance σ^2. Here again, the prior distribution on $\log \lambda_{ij}$ depends on the data set variable logstd; hence, the construction of the prior must take place before the PRIOR statement for $\log \lambda_{ij}$. The symbol lp keeps track of the cumulative log prior density for $\log \lambda_{ij}$.

The symbol lambda is the exponential of the corresponding $\log \lambda_{ij}$, and the MODEL statement gives the response variable y a Poisson likelihood with a mean parameter lambda.

The posterior summary statistics table is shown in Output 52.5.4.

Output 52.5.4 Summary Statistics for the Nonlinear Poisson Regression

Nonlinear Poisson Regression Random Effects Model

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
logsig	10000	0.1045	0.3862	-0.1563	0.0883	0.3496
beta1	10000	-0.4467	1.2818	-1.1641	-0.4421	0.2832
beta2	10000	0.5858	0.5808	0.2376	0.5796	0.9385
alpha1	10000	2.9719	2.3658	1.6225	2.9612	4.3137
alpha2	10000	1.6406	0.8674	1.1429	1.6782	2.1673
s2	10000	1.7004	1.7995	0.7316	1.1931	2.0121
adif	10000	1.3313	2.4934	-0.1670	1.2669	2.8032
bdif	10000	-1.0325	1.4186	-1.8379	-1.0284	-0.2189

Draper (1996) reports a posterior mean and standard deviation as follows: $\log \sigma = (0.28, 0.42)$, $\beta_1 = (-0.45, 1.5)$, $\beta_2 = (0.63, 0.68)$, and $\alpha 1 - \alpha 2 = (1.3, 3.0)$. Most estimates from Output 52.5.4 agree with Draper's estimates, with the exception of $\log \sigma$. The difference might be attributed to the different set of prior distributions on α_i, β_i, and σ that are used in this analysis.

You can also use PROC NLMIXED to fit the same model. The following statements run PROC NLMIXED and produce Output 52.5.5:

```
proc nlmixed data=pump;
   ods select parameterestimates additionalestimates;
   ods output additionalestimates=cp parameterestimates=ps;
   parms logsig 0 beta1 1 beta2 1 alpha1 1 alpha2 1;
   if (group = 1) then eta = alpha1 + beta1*logtstd + e;
   else eta = alpha2 + beta2*logtstd + e;
   lambda = exp(eta);
   model y ~ poisson(lambda);
   random e ~ normal(0,exp(2*logsig)) subject=pump;
   estimate 'adif' alpha1-alpha2;
   estimate 'bdif' beta1-beta2;
   estimate 's2' exp(2*logsig);
run;
```

Output 52.5.5 Estimates by PROC NLMIXED

Nonlinear Poisson Regression Random Effects Model

The NLMIXED Procedure

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower
logsig	−0.3161	0.3213	9	−0.98	0.3508	0.05	−1.0429
beta1	−0.4256	0.7473	9	−0.57	0.5829	0.05	−2.1162
beta2	0.6097	0.3814	9	1.60	0.1443	0.05	−0.2530
alpha1	2.9644	1.3826	9	2.14	0.0606	0.05	−0.1632
alpha2	1.7992	0.5492	9	3.28	0.0096	0.05	0.5568

Parameter Estimates

Parameter	Upper	Gradient
logsig	0.4107	−0.00002
beta1	1.2649	−0.00002
beta2	1.4724	−1.61E−6
alpha1	6.0921	−5.25E−6
alpha2	3.0415	−5.73E−6

Additional Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
adif	1.1653	1.4855	9	0.78	0.4529	0.05	−2.1952	4.5257
bdif	−1.0354	0.8389	9	−1.23	0.2484	0.05	−2.9331	0.8623
s2	0.5314	0.3415	9	1.56	0.1541	0.05	−0.2410	1.3038

Again, the point estimates from PROC NLMIXED for the mean parameters agree relatively closely with the Bayesian posterior means. You can note that there are differences in the likelihood-based

standard errors. This is most likely due to the fact that the Bayesian standard deviations account for the uncertainty in estimating σ^2, whereas the likelihood approach plugs in its estimated value.

You can do a similar kernel density plot that compares the PROC MCMC results, the PROC NLMIXED results and those reported by Draper. The following statements generate Output 52.5.6:

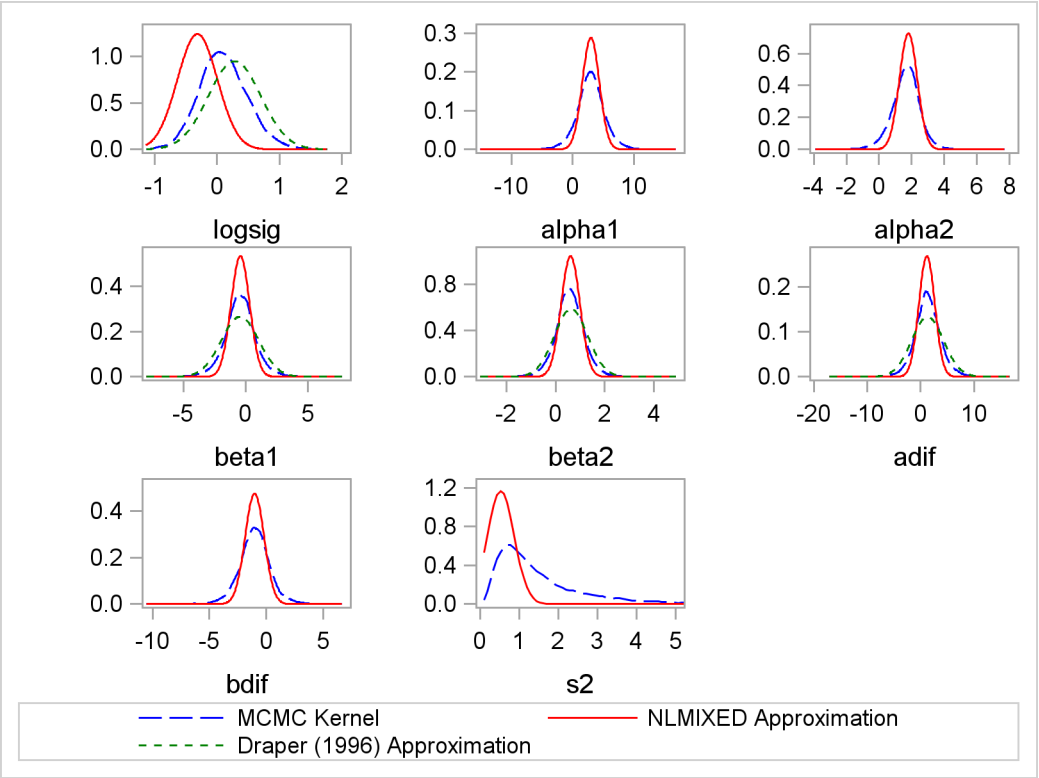
```
data nlmest(keep = parm mean sd);
   set ps cp;
   mean = estimate;
   sd = standarderror;
   if _n_ <= 5 then
      parm = parameter;
   else
      parm = label;
run;

data msd (keep=mean sd);
   set nlmest;
   do j = 1 to 401;
      output;
   end;
run;

data _null_;
   set ps;
   call symputx(compress('m' || parameter,'*'), estimate);
   call symputx(compress('s' || parameter,'*'), standarderror);
run;

data _null_;
   set cp;
   call symputx(compress('m' || label,'*'), estimate);
   call symputx(compress('s' || label,'*'), standarderror);
run;

%put &mlogsig &mbeta1 &mbeta2 &malpha1 &malpha2 &madif &mbdif &ms2;
%put &slogsig &sbeta1 &sbeta2 &salpha1 &salpha2 &sadif &sbdif &ss2;

proc kde data=postout;
   univar logsig beta1 beta2 alpha1 alpha2 adif bdif s2 / out=m1 (drop=count);
run;

%reshape(m1, mcmc, suffix1=, suffix2=_kde);

data all;
   set mcmc;
   logsig_nlm = pdf('normal', logsig, &mlogsig, &slogsig);
   alpha1_nlm = pdf('normal', alpha1, &malpha1, &salpha1);
   alpha2_nlm = pdf('normal', alpha2, &malpha2, &salpha2);
   beta1_nlm = pdf('normal', beta1, &mbeta1, &sbeta1);
   beta2_nlm = pdf('normal', beta2, &mbeta2, &sbeta2);
   adif_nlm = pdf('normal', adif, &madif, &sadif);
   bdif_nlm = pdf('normal', bdif, &mbdif, &sbdif);
```

```
   s2_nlm = pdf('normal', s2, &ms2, &ss2);
   logsig_draper = pdf('normal', logsig, 0.28, 0.42);
   beta1_draper = pdf('normal', beta1, -0.45, 1.5);
   beta2_draper = pdf('normal', beta2, 0.63, 0.68);
   adif_draper = pdf('normal', adif, 1.3, 3.0);
run;

proc template;
   define statgraph threebythree;
      %macro plot;
         begingraph;
            layout lattice / rows=3 columns=3;
               layout overlay /yaxisopts=(label=" ");
                  seriesplot y=logsig_kde x=logsig
                     / connectorder=xaxis
                       lineattrs=(pattern=mediumdash color=blue)
                     legendlabel = "MCMC Kernel" name="MCMC";
                  seriesplot y=logsig_nlm x=logsig
                     / connectorder=xaxis lineattrs=(color=red)
                     legendlabel = "NLMIXED Approximation" name="NLMIXED";
                  seriesplot y=logsig_draper x=logsig
                     / connectorder=xaxis
                       lineattrs=(pattern=shortdash color=green)
                     legendlabel = "Draper (1996) Approximation" name="Draper";
               endlayout;
               %do i = 1 %to 2;
                  layout overlay /yaxisopts=(label=" ");
                     seriesplot y=alpha&i._kde x=alpha&i
                        / connectorder=xaxis
                          lineattrs=(pattern=mediumdash color=blue)
                        legendlabel = "MCMC Kernel" name="MCMC";
                     seriesplot y=alpha&i._nlm x=alpha&i
                        / connectorder=xaxis lineattrs=(color=red)
                        legendlabel = "NLMIXED Approximation" name="NLMIXED";
                  endlayout;
               %end;
               %do i = 1 %to 2;
                  layout overlay /yaxisopts=(label=" ");
                     seriesplot y=beta&i._kde x=beta&i
                        / connectorder=xaxis
                          lineattrs=(pattern=mediumdash color=blue)
                        legendlabel = "MCMC Kernel" name="MCMC";
                     seriesplot y=beta&i._nlm x=beta&i
                        / connectorder=xaxis lineattrs=(color=red)
                        legendlabel = "NLMIXED Approximation" name="NLMIXED";
                     seriesplot y=beta&i._draper x=beta&i
                        / connectorder=xaxis
                          lineattrs=(pattern=shortdash color=green)
                        legendlabel = "Draper (1996) Approximation" name="Draper";
                  endlayout;
               %end;
               layout overlay /yaxisopts=(label=" ");
                  seriesplot y=adif_kde x=adif
                     / connectorder=xaxis
```

```
                    lineattrs=(pattern=mediumdash color=blue)
                 legendlabel = "MCMC Kernel" name="MCMC";
              seriesplot y=adif_nlm x=adif
                 / connectorder=xaxis lineattrs=(color=red)
                 legendlabel = "NLMIXED Approximation" name="NLMIXED";
              seriesplot y=adif_draper x=adif
                 / connectorder=xaxis
                   lineattrs=(pattern=shortdash color=green)
                 legendlabel = "Draper (1996) Approximation" name="Draper";
           endlayout;
           layout overlay /yaxisopts=(label=" ");
              seriesplot y=bdif_kde x=bdif
                 / connectorder=xaxis
                   lineattrs=(pattern=mediumdash color=blue)
                 legendlabel = "MCMC Kernel" name="MCMC";
              seriesplot y=bdif_nlm x=bdif
                 / connectorder=xaxis lineattrs=(color=red)
                 legendlabel = "NLMIXED Approximation" name="NLMIXED";
           endlayout;
           layout overlay /yaxisopts=(label=" ")
                          xaxisopts=(linearopts=(viewmin=0 viewmax=5));
              seriesplot y=s2_kde x=s2
                 / connectorder=xaxis
                   lineattrs=(pattern=mediumdash color=blue)
                 legendlabel = "MCMC Kernel" name="MCMC";
              seriesplot y=s2_nlm x=s2
                 / connectorder=xaxis lineattrs=(color=red)
                 legendlabel = "NLMIXED Approximation" name="NLMIXED";
           endlayout;
           Sidebar / align = bottom;
              discretelegend "MCMC" "NLMIXED" "Draper";
           endsidebar;
        endlayout;
     endgraph;
  %mend; %plot;
end;
run;

proc sgrender data=all template=threebythree;
run;
```

The macro %RESHAPE is defined in the example "Logistic Regression Random-Effects Model" on page 3614.

Output 52.5.6 Comparing Estimates from PROC MCMC (dashed blue), PROC NLMIXED (solid red) and Draper (dotted green)



Example 52.6: Change Point Models

Consider the data set from Bacon and Watts (1971), where y_i is the logarithm of the height of the stagnant surface layer and the covariate x_i is the logarithm of the flow rate of water. The following statements create the data set:

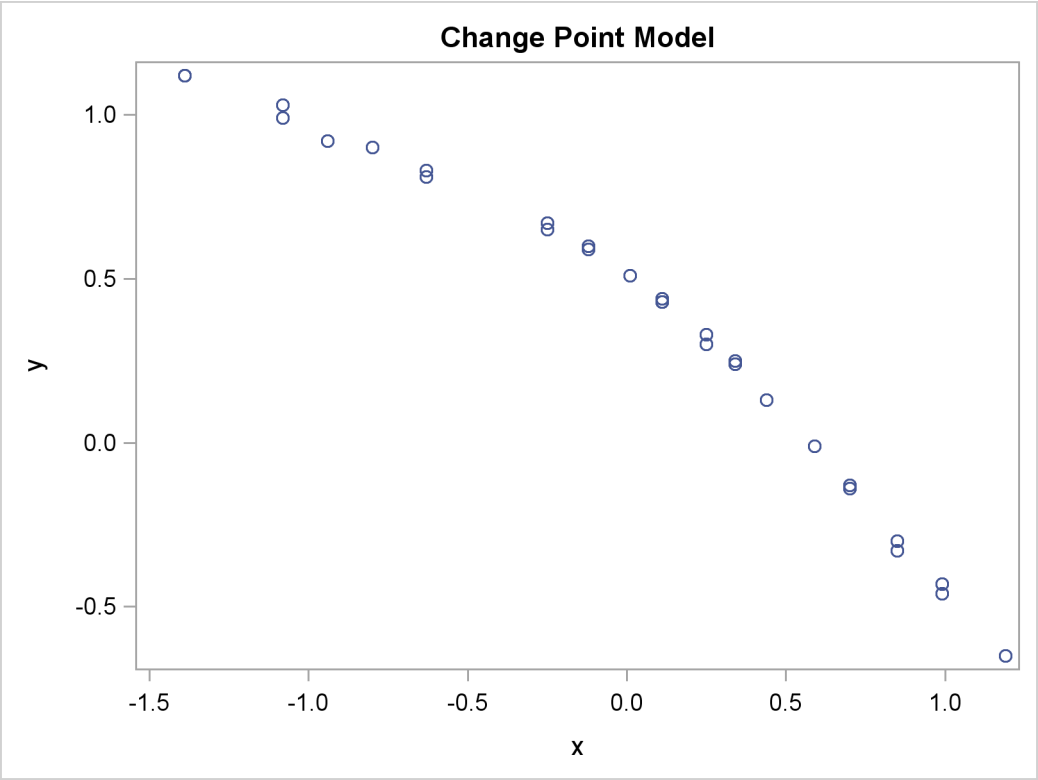
```
title 'Change Point Model';
data stagnant;
   input y x @@;
   ind = _n_;
   datalines;
 1.12  -1.39   1.12  -1.39   0.99  -1.08   1.03  -1.08
 0.92  -0.94   0.90  -0.80   0.81  -0.63   0.83  -0.63
 0.65  -0.25   0.67  -0.25   0.60  -0.12   0.59  -0.12
 0.51   0.01   0.44   0.11   0.43   0.11   0.43   0.11
 0.33   0.25   0.30   0.25   0.25   0.34   0.24   0.34
 0.13   0.44  -0.01   0.59  -0.13   0.70  -0.14   0.70
-0.30   0.85  -0.33   0.85  -0.46   0.99  -0.43   0.99
-0.65   1.19
;
```

A scatter plot (Output 52.6.1) shows the presence of a nonconstant slope in the data. This suggests a change point regression model (Carlin, Gelfand, and Smith 1992). The following statements

generate the scatter plot in Output 52.6.1:

```
proc sgplot data=stagnant;
   scatter x=x y=y;
run;
```

Output 52.6.1 Scatter Plot of the Stagnant Data Set



Let the change point be cp. Following formulation by Spiegelhalter et al. (1996), the regression model is as follows:

$$y_i \sim \begin{cases} \text{normal}(\alpha + \beta_1(x_i - \text{cp}), \sigma^2) & \text{if } x_i < \text{cp} \\ \text{normal}(\alpha + \beta_2(x_i - \text{cp}), \sigma^2) & \text{if } x_i >= \text{cp} \end{cases}$$

You might consider the following diffuse prior distributions:

$$\begin{aligned} \pi(\text{cp}) &\sim \text{uniform}(-1.3, 1.1) \\ \pi(\alpha, \beta_1, \beta_2) &\sim \text{normal}(0, \text{var} = 1e6) \\ \pi(\sigma^2) &\sim \text{uniform}(0, 5) \end{aligned}$$

The following statements generate Output 52.6.2:

```
proc mcmc data=stagnant outpost=postout seed=24860 ntu=1000
          nmc=20000;
   ods select PostSummaries;
   ods output PostSummaries=ds;

   array beta[2];
```

```
   parms alpha cp beta1 beta2;
   parms s2;

   prior cp ~ unif(-1.3, 1.1);
   prior s2  ~ uniform(0, 5);
   prior alpha beta:  ~ normal(0, v = 1e6);

   j = 1 + (x >= cp);
   mu = alpha + beta[j] * (x - cp);
   model y ~ normal(mu, var=s2);
run;
```

The PROC MCMC statement specifies the input data set (stagnant), the output data set (postout), a random number seed, a tuning sample of 1000, and an MCMC sample of 20000. The ODS SELECT statement displays only the summary statistics table. The ODS OUTPUT statement saves the summary statistics table to the data set ds.

The ARRAY statement allocates an array of size 2 for the beta parameters. You can use beta1 and beta2 as parameter names without allocating an array, but having the array makes it easier to construct the likelihood function. The two PARMS statements put the five model parameters in two blocks. The three PRIOR statements specify the prior distributions for these parameters.

The symbol j indicates the segment component of the regression. When x is less than the change point, (x >= cp) returns 0 and j is assigned the value 1; if x is greater than or equal to the change point, (x >= cp) returns 1 and j is 2. The symbol mu is the mean for the jth segment, and beta[j] changes between the two regression coefficients depending on the segment component. The MODEL statement assigns the normal model to the response variable y.

Posterior summary statistics are shown in Output 52.6.2.

Output 52.6.2 MCMC Estimates of the Change Point Regression Model

Change Point Model

The MCMC Procedure

Posterior Summaries

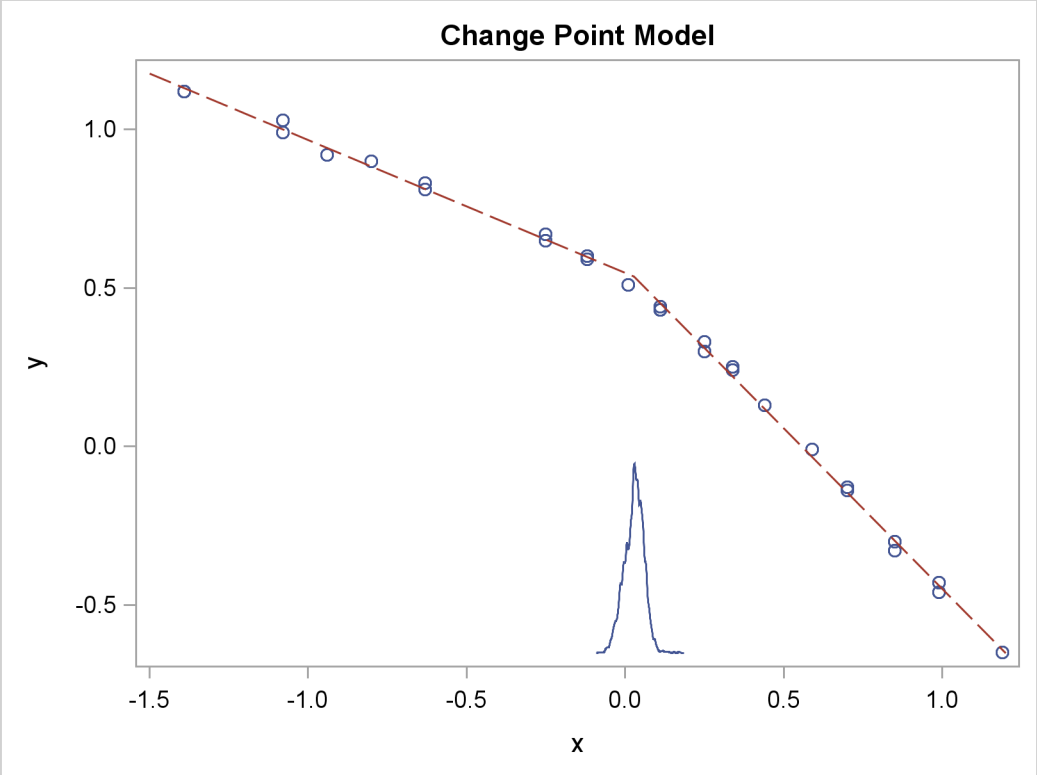
Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
alpha	20000	0.5349	0.0249	0.5188	0.5341	0.5509
cp	20000	0.0283	0.0314	0.00728	0.0303	0.0493
beta1	20000	-0.4200	0.0146	-0.4293	-0.4198	-0.4111
beta2	20000	-1.0136	0.0167	-1.0248	-1.0136	-1.0023
s2	20000	0.000451	0.000145	0.000348	0.000425	0.000522

You can use PROC SGPLOT to visualize the model fit. Output 52.6.3 shows the fitted regression lines over the original data. In addition, on the bottom of the plot is the kernel density of the posterior marginal distribution of cp, the change point. The kernel density plot shows the relative variability of the posterior distribution on the data plot. You can use the following statements to create the plot:

```
data _null_;
   set ds;
   call symputx(parameter, mean);
run;

data b;
   missing A;
   input x1 @@;
   if x1 eq .A then x1 = &cp;
   if _n_ <= 2 then y1 = &alpha + &beta1 * (x1 - &cp);
   else y1 = &alpha + &beta2 * (x1 - &cp);
   datalines;
   -1.5 A 1.2
   ;

proc kde data=postout;
   univar cp / out=m1 (drop=count);
run;

data m1;
   set m1;
   density = (density / 25) - 0.653;
run;

data all;
   set stagnant b m1;
run;

proc sgplot data=all noautolegend;
   scatter x=x y=y;
   series x=x1 y=y1 / lineattrs = graphdata2;
   series x=value y=density / lineattrs = graphdata1;
run;
```

The macro variables &alpha, &beta1, &beta2, and &cp store the posterior mean estimates from the data set ds. The data set predicted contains three predicted values, at the minimum and maximum values of x and the estimated change point &cp. These input values give you fitted values from the regression model. Data set m1 contains the kernel density estimates of the parameter cp. The density is scaled down so the curve would fit in the plot. Finally, you use PROC SGPLOT to overlay the scatter plot, regression line and kernel density plots in the same graph.

Output 52.6.3 Estimated Fit to the Stagnant Data Set



Example 52.7: Exponential and Weibull Survival Analysis

This example covers two commonly used survival analysis models: the exponential model and the Weibull model. The deviance information criterion (DIC) is used to do model selections, and you can also find programs that visualize posterior quantities. Exponential and Weibull models are widely used for survival analysis. This example shows you how to use PROC MCMC to analyze the treatment effect for the E1684 melanoma clinical trial data. These data were collected to assess the effectiveness of using interferon alpha-2b in chemotherapeutic treatment of melanoma. The following statements create the data set:

```
data e1684;
   input t t_cen treatment @@;
   if t = . then do;
     t = t_cen;
     v = 0;
   end;
   else
     v = 1;
   ifn = treatment - 1;
   et = exp(t);
   lt = log(t);
   drop t_cen;
   datalines;
```

```
 1.57808  0.00000  2   1.48219  0.00000  2    .       7.33425  1
 2.23288  0.00000  1    .       9.38356  2   3.27671  0.00000  1
  .       9.64384  1   1.66575  0.00000  2   0.94247  0.00000  1

... more lines ...

 3.39178  0.00000  1   .        4.36164  2   .        4.81918  2
;
```

The data set e1684 contains the following variables: t is the failure time that equals the censoring time whether the observation was censored, v indicates whether the observation is an actual failure time or a censoring time, treatment indicates two levels of treatments, and ifn indicates the use of interferon as a treatment. The variables et and lt are the exponential and logarithm transformation of the time t. The published data contains other potential covariates that are not listed here. This example concentrates on the effectiveness of the interferon treatment.

Exponential Survival Model

The density function for exponentially distributed survival times is as follows:

$$p(t_i|\lambda_i) = \lambda_i \exp(-\lambda_i t_i)$$

Note that this formulation of the exponential distribution is different from what is used in the SAS probability function PDF. The definition used in PDF for the exponential distributions is as follows:

$$p(t_i|\nu_i) = \frac{1}{\nu_i} \exp(-\frac{t_i}{\nu_i})$$

The relationship between λ and ν is as follows:

$$\lambda_i = \frac{1}{\nu_i}$$

The corresponding survival function, using the λ_i formulation, is as follows:

$$S(t_i|\lambda_i) = \exp(-\lambda_i t_i)$$

If you have a sample $\{t_i\}$ of n independent exponential survival times, each with mean λ_i, then the likelihood function in terms of λ is as follows:

$$\begin{aligned} L(\lambda|t) &= \Pi_{i=1}^{n} p(t_i|\lambda_i)^{\nu_i} S(t_i|\lambda_i)^{1-\nu_i} \\ &= \Pi_{i=1}^{n} (\lambda_i \exp(-\lambda_i t_i))^{\nu_i} (\exp(-\lambda_i t_i))^{1-\nu_i} \\ &= \Pi_{i=1}^{n} \lambda_i^{\nu_i} \exp(-\lambda_i t_i) \end{aligned}$$

If you link the covariates to λ with $\lambda_i = \exp x_i'\beta$, where x_i is the vector of covariates corresponding to the ith observation and β is a vector of regression coefficients, then the log-likelihood function is as follows:

$$l(\beta|t,x) = \sum_{i=1}^{n} \nu_i x_i'\beta - t_i \exp(x_i'\beta)$$

In the absence of prior information about the parameters in this model, you can choose diffuse normal priors for the β:

$$\beta \sim \text{normal}(0, \text{sd} = 10000)$$

There are two ways to program the log-likelihood function in PROC MCMC. You can use the SAS functions LOGPDF and LOGSDF. Alternatively, you can use the simplified log-likelihood function, which is more computationally efficient. You get identical results by using either approaches.

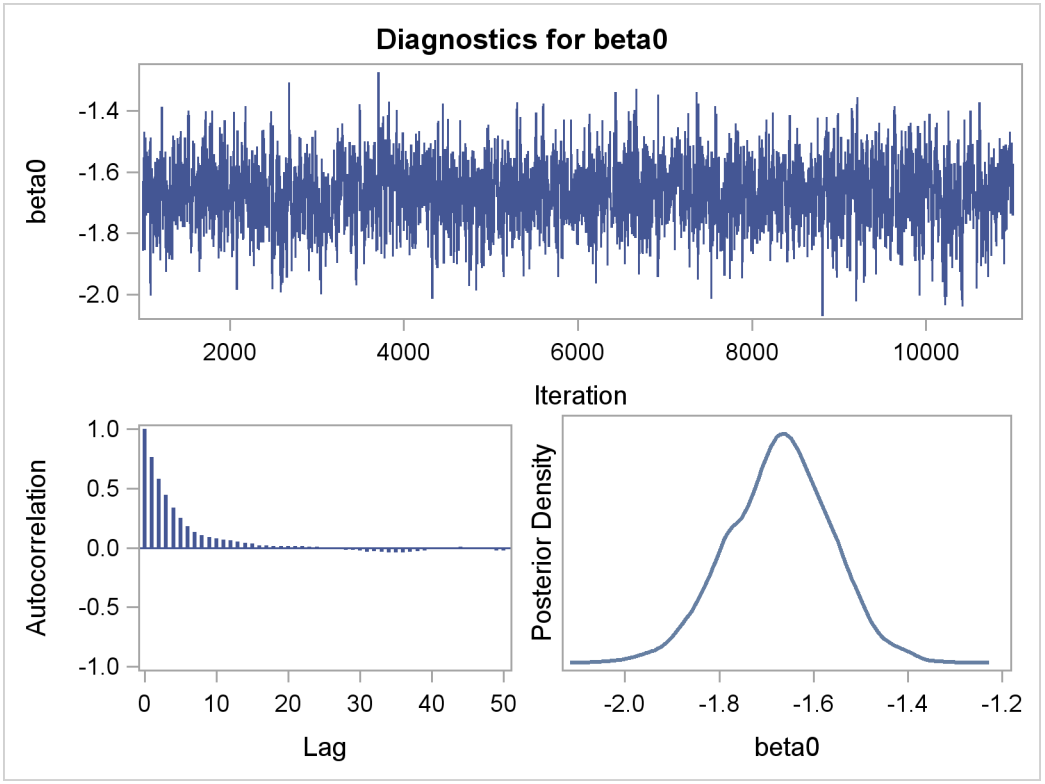
The following PROC MCMC statements fit an exponential model with simplified log-likelihood function:

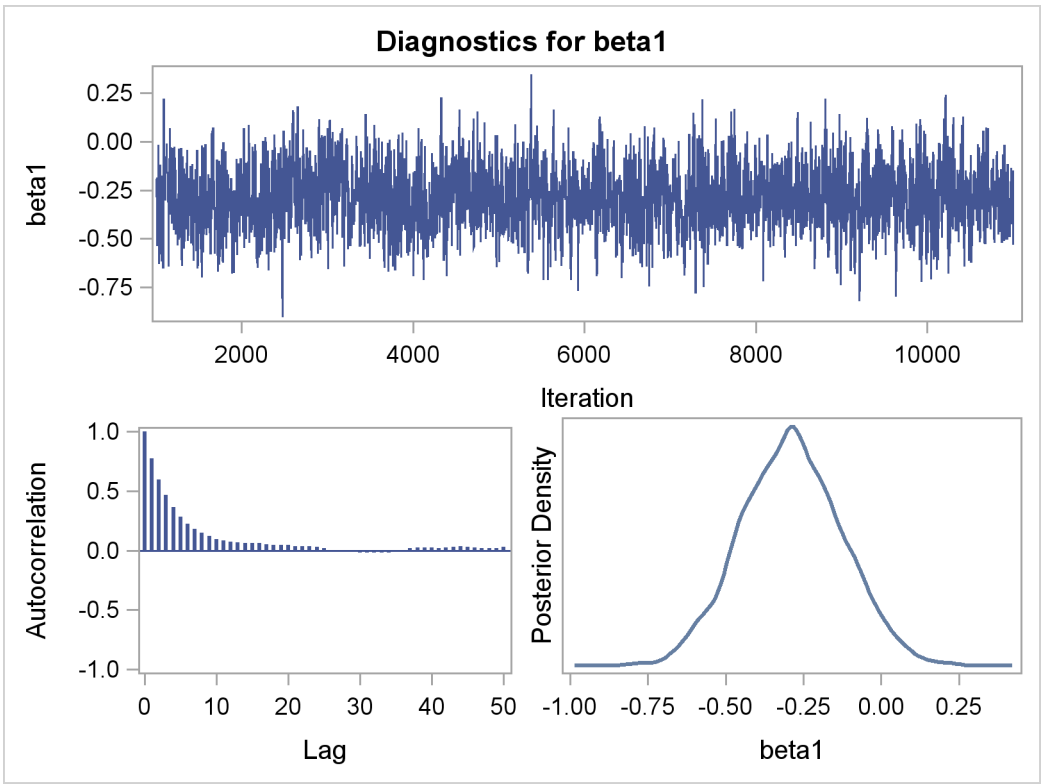
```
title 'Exponential Survival Model';
ods graphics on;
proc mcmc data=e1684 outpost=expsurvout nmc=10000 seed=4861;
    ods select PostSummaries PostIntervals TADpanel
               ess mcse;
   parms (beta0 beta1) 0;
   prior beta: ~ normal(0, sd = 10000);
   /*****************************************************/
   /* (1) the logpdf and logsdf functions are not used  */
   /*****************************************************/
   /*     nu = 1/exp(beta0 + beta1*ifn);
          llike = v*logpdf("exponential", t, nu) +
                  (1-v)*logsdf("exponential", t, nu);
   */
   /*****************************************************/
   /* (2) the simplified likelihood formula is used     */
   /*****************************************************/
   l_h = beta0 + beta1*ifn;
   llike = v*(l_h) -  t*exp(l_h);
   model general(llike);
run;
ods graphics off;
```

The two assignment statements that are commented out calculate the log-likelihood function by using the SAS functions LOGPDF and LOGSDF for the exponential distribution. The next two assignment statements calculate the log likelihood by using the simplified formula. The first approach is slower because of the redundant calculation involved in calling both LOGPDF and LOGSDF.

An examination of the trace plots for β_0 and β_1 (see Output 52.7.1) reveals that the sampling has gone well with no particular concerns about the convergence or mixing of the chains.

Output 52.7.1 Posterior Plots for β_0 and β_1 in the Exponential Survival Analysis





The MCMC results are shown in Output 52.7.2.

Output 52.7.2 Posterior Summary and Interval Statistics

Exponential Survival Model

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
beta0	10000	−1.6715	0.1091	−1.7426	−1.6684	−1.5964
beta1	10000	−0.2879	0.1615	−0.4001	−0.2892	−0.1803

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
beta0	0.050	−1.8907	−1.4639	−1.8930	−1.4673
beta1	0.050	−0.5985	0.0300	−0.6104	0.0169

The Monte Carlo standard errors and effective sample sizes are shown in Output 52.7.3. The posterior means for β_0 and β_1 are estimated with high precision, with small standard errors with respect to the standard deviation. This indicates that the mean estimates have stabilized and do not vary greatly in the course of the simulation. The effective sample sizes are roughly the same for both parameters.

Output 52.7.3 MCSE and ESS

Exponential Survival Model

The MCMC Procedure

Monte Carlo Standard Errors

Parameter	MCSE	Standard Deviation	MCSE/SD
beta0	0.00302	0.1091	0.0277
beta1	0.00485	0.1615	0.0301

Effective Sample Sizes

Parameter	ESS	Correlation Time	Efficiency
beta0	1304.1	7.6682	0.1304
beta1	1107.2	9.0319	0.1107

The next part of this example shows fitting a Weibull regression to the data and then comparing the two models with DIC to see which one provides a better fit to the data.

Weibull Survival Model

The density function for Weibull distributed survival times is as follows:

$$p(t_i|\alpha,\lambda_i) = \alpha t_i^{\alpha-1}\exp(\lambda_i - \exp(\lambda_i)t_i^{\alpha})$$

Note that this formulation of the Weibull distribution is different from what is used in the SAS probability function PDF. The definition used in PDF is as follows:

$$p(t_i|\alpha,\gamma_i) = \exp\left(-\left(\frac{t_i}{\gamma_i}\right)^{\alpha}\right)\frac{\alpha}{\gamma_i}\left(\frac{t_i}{\gamma_i}\right)^{\alpha-1}$$

The relationship between λ and γ in these two parameterizations is as follows:

$$\lambda_i = -\alpha\log\gamma_i$$

The corresponding survival function, using the λ_i formulation, is as follows:

$$S(t_i|\alpha,\lambda_i) = \exp(-\exp(\lambda_i)t_i^{\alpha})$$

If you have a sample $\{t_i\}$ of n independent Weibull survival times, with parameters α, and λ_i, then the likelihood function in terms of α and λ is as follows:

$$\begin{aligned} L(\alpha,\lambda|t) &= \Pi_{i=1}^{n} p(t_i|\alpha,\lambda_i)^{\nu_i} S(t_i|\alpha,\lambda_i)^{1-\nu_i} \\ &= \Pi_{i=1}^{n} (\alpha t_i^{\alpha-1}\exp(\lambda_i - \exp(\lambda_i)t_i^{\alpha}))^{\nu_i} (\exp(-\exp(\lambda_i)t_i^{\alpha}))^{1-\nu_i} \\ &= \Pi_{i=1}^{n} (\alpha t_i^{\alpha-1}\exp(\lambda_i))^{\nu_i} (\exp(-\exp(\lambda_i)t_i^{\alpha})) \end{aligned}$$

If you link the covariates to λ with $\lambda_i = x_i'\beta$, where x_i is the vector of covariates corresponding to the ith observation and β is a vector of regression coefficients, the log-likelihood function becomes this:

$$l(\alpha,\beta|t,x) = \sum_{i=1}^{n} \nu_i(\log(\alpha) + (\alpha-1)\log(t_i) + x_i'\beta) - \exp(x_i'\beta)t_i^{\alpha})$$

As with the exponential model, in the absence of prior information about the parameters in this model, you can use diffuse normal priors on β. You might wish to choose a diffuse gamma distribution for α. Note that when $\alpha = 1$, the Weibull survival likelihood reduces to the exponential survival likelihood. Equivalently, by looking at the posterior distribution of α, you can conclude whether fitting an exponential survival model would be more appropriate than the Weibull model.

PROC MCMC also allows you to make inference on any functions of the parameters. Quantities of interest in survival analysis include the value of the survival function at specific times for specific treatments and the relationship between the survival curves for different treatments. With PROC MCMC, you can compute a sample from the posterior distribution of the interested survival functions at any number of points. The data in this example range from about 0 to 10 years, and the treatment of interest is the use of interferon.

Like in the previous exponential model example, there are two ways to fit this model: using the SAS functions LOGPDF and LOGSDF, or using the simplified log likelihood functions. The example uses the latter method. The following statements run PROC MCMC and produce Output 52.7.4:

```
title 'Weibull Survival Model';
proc mcmc data=e1684 outpost=weisurvout nmc=10000 seed=1234
          monitor=(_parms_ surv_ifn surv_noifn);
   ods select PostSummaries;
   ods output PostSummaries=ds PostIntervals=is;
   array surv_ifn[10];
   array surv_noifn[10];
   parms alpha 1 (beta0 beta1) 0;
   prior beta: ~ normal(0, var=10000);
   prior alpha ~ gamma(0.001,is=0.001);

   beginnodata;
      do t = 1 to 10;
        surv_ifn[t] = exp(-exp(beta0+beta1)*t**alpha);
        surv_noifn[t] = exp(-exp(beta0)*t**alpha);
      end;
   endnodata;

   lambda = beta0 + beta1*ifn;
   /*****************************************************/
   /* (1) the logpdf and logsdf functions are not used  */
   /*****************************************************/
   /*     gamma =  exp(-lambda /alpha);
          llike = v*logpdf('weibull', t, alpha, gamma) +
                  (1-v)*logsdf('weibull', t, alpha, gamma);
   */
   /*****************************************************/
   /* (2) the simplified likelihood formula is used     */
   /*****************************************************/
   llike  = v*(log(alpha) + (alpha-1)*log(t) + lambda) -
            exp(lambda)*(t**alpha);
   model general(llike);
run;
```

The MONITOR= option indicates the parameters and quantities of interest that PROC MCMC tracks. The symbol _PARMS_ specifies all model parameters. The array surv_ifn stores the expected survival probabilities for patients who received interferon over a period of 10 years. Similarly, surv_noifn stores the expected survival probabilities for patients who did not received interferon.

The BEGINNODATA and ENDNODATA statements enclose the calculations for the survival probabilities. The assignment statements proceeding the MODEL statement calculate the log likelihood for the Weibull survival model. The MODEL statement specifies the log likelihood that you programmed.

An examination of the trace plots for α, β_0, and β_1 (not displayed here) reveals that the sampling has gone well, with no particular concerns about the convergence or mixing of the chains.

Output 52.7.4 displays the posterior summary statistics.

Output 52.7.4 Posterior Summary Statistics

Weibull Survival Model

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
alpha	10000	0.7856	0.0533	0.7488	0.7849	0.8225
beta0	10000	−1.3414	0.1389	−1.4321	−1.3424	−1.2463
beta1	10000	−0.2918	0.1683	−0.4050	−0.2919	−0.1711
surv_ifn1	10000	0.8212	0.0237	0.8054	0.8228	0.8374
surv_ifn2	10000	0.7128	0.0308	0.6919	0.7138	0.7337
surv_ifn3	10000	0.6283	0.0352	0.6039	0.6282	0.6522
surv_ifn4	10000	0.5588	0.0383	0.5326	0.5589	0.5852
surv_ifn5	10000	0.5001	0.0405	0.4728	0.4997	0.5276
surv_ifn6	10000	0.4497	0.0420	0.4204	0.4489	0.4786
surv_ifn7	10000	0.4060	0.0431	0.3760	0.4051	0.4350
surv_ifn8	10000	0.3677	0.0437	0.3375	0.3664	0.3972
surv_ifn9	10000	0.3340	0.0440	0.3035	0.3325	0.3638
surv_ifn10	10000	0.3041	0.0440	0.2736	0.3024	0.3337
surv_noifn1	10000	0.7685	0.0280	0.7501	0.7701	0.7876
surv_noifn2	10000	0.6360	0.0349	0.6131	0.6376	0.6598
surv_noifn3	10000	0.5372	0.0386	0.5119	0.5384	0.5637
surv_noifn4	10000	0.4593	0.0407	0.4330	0.4599	0.4878
surv_noifn5	10000	0.3960	0.0417	0.3686	0.3959	0.4256
surv_noifn6	10000	0.3437	0.0421	0.3159	0.3432	0.3727
surv_noifn7	10000	0.2999	0.0419	0.2715	0.2993	0.3282
surv_noifn8	10000	0.2629	0.0412	0.2349	0.2624	0.2909
surv_noifn9	10000	0.2313	0.0403	0.2037	0.2302	0.2581
surv_noifn10	10000	0.2041	0.0392	0.1767	0.2033	0.2302

An examination of the α parameter reveals that the exponential model might not be inappropriate here. The estimated posterior mean of α is 0.7856 with a posterior standard deviation of 0.0533. As noted previously, if $\alpha = 1$, then the Weibull survival distribution is the exponential survival distribution. With these data, you can see that the evidence is in favor of $\alpha < 1$. The value 1 is almost 4 posterior standard deviations away from the posterior mean. The following statements compute the posterior probability of the hypothesis that $\alpha < 1$.:

```
proc format;
   value alphafmt low-<1 = 'alpha < 1' 1-high = 'alpha >= 1';
run;

proc freq data=weisurvout;
   tables alpha /nocum;
   format alpha alphafmt.;
run;
```

The PROC FREQ results are shown in Output 52.7.5.

Output 52.7.5 Frequency Analysis of α

```
                    Weibull Survival Model

                      The FREQ Procedure

                 alpha    Frequency     Percent
              ----------------------------------
              alpha < 1        10000      100.00
```

The output from PROC FREQ shows that 100% of the 10000 simulated values for α are less than 1. This is a very strong indication that the exponential model is too restrictive to model these data well.

You can examine the estimated survival probabilities over time individually, either through the posterior summary statistics or by looking at the kernel density plots. Alternatively, you might find it more informative to examine these quantities in relation with each other. For example, you can use a side-by-side box plot to display these posterior distributions by using PROC SGPLOT (“Statistical Graphics Using ODS” on page 497). First you need to take the posterior output data set weisurvout and stack variables that you want to plot. For example, to plot all the survival times for patients who received interferon, you want to stack surv_inf1–surv_inf10. The macro %Stackdata takes an input data set dataset, stacks the wanted variables vars, and outputs them into the output data set.

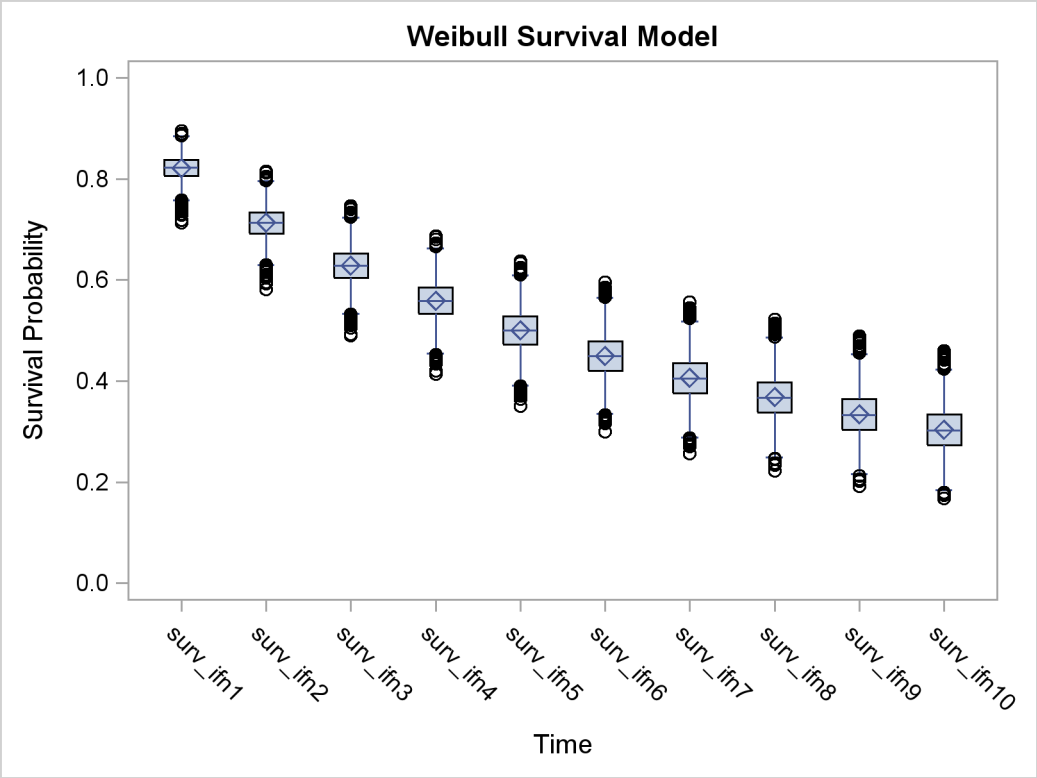
The following statements define the macro stackdata:

```
/* define macro stackdata */
%macro StackData(dataset,output,vars);
   data &output;
      length var $ 32;
      if 0 then set &dataset nobs=nnn;
      array lll[*] &vars;
      do jjj=1 to dim(lll);
         do iii=1 to nnn;
            set &dataset point=iii;
            value = lll[jjj];
            call vname(lll[jjj],var);
            output;
         end;
      end;
      stop;
      keep var value;
   run;
%mend;

/* stack the surv_ifn variables and saved them to survifn. */
%StackData(weisurvout, survifn, surv_ifn1-surv_ifn10);
```

Once you stack the data, use PROC SGPLOT to create the side-by-side box plots. The following statements generate Output 52.7.6:

```
proc sgplot data=survifn;
   yaxis label='Survival Probability' values=(0 to 1 by 0.2);
   xaxis label='Time' discreteorder=data;
   vbox value / category=var;
run;
```

Output 52.7.6 Side-by-Side Box Plots of Estimated Survival Probabilities



There is a clear decreasing trend over time of the survival probabilities for patients who receive the treatment. You might ask how does this group compare to those who did not receive the treatment? In this case, you want to overlay the two predicted curves for the two groups of patients and add the corresponding credible interval. See Output 52.7.7. To generate the graph, you first take the posterior mean estimates from the ODS output table ds and the lower and upper HPD interval estimates is, store them in the data set surv, and draw the figure by using PROC SGPLOT.

The following statements generate data set surv:

```
data surv;
   set ds;
   if _n_ >= 4 then do;
      set is point=_n_;
      group = 'with interferon   ';
      time = _n_ - 3;
      if time > 10 then do;
         time = time - 10;
         group = 'without interferon';
      end;
```

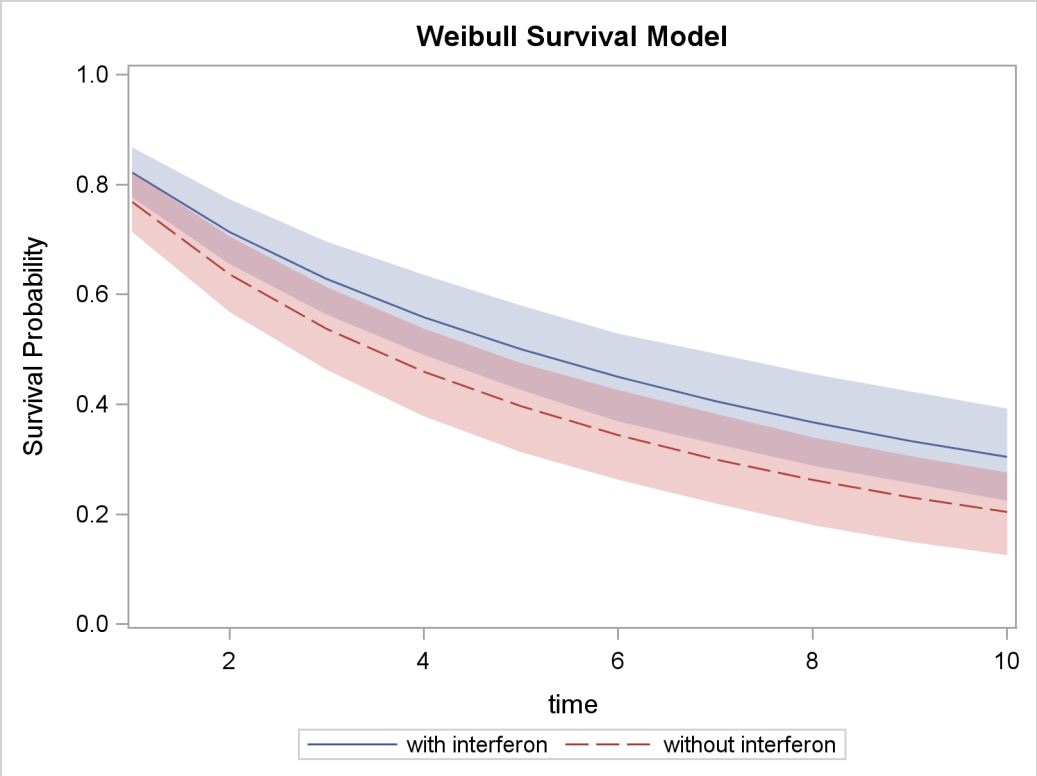
```
      output;
   end;
   keep time group mean hpdlower hpdupper;
run;
```

The following SGPLOT statements generate Output 52.7.7:

```
proc sgplot data=surv;
   yaxis label="Survival Probability" values=(0 to 1 by 0.2);
   series x=time y=mean / group = group name='i';
   band x=time lower=hpdlower upper=hpdupper / group = group transparency=0.7;
   keylegend 'i';
run;
```

In Output 52.7.7, the solid line is the survival curve for patients who received interferon; the shaded region centers at the solid line is the 95% HPD intervals; the medium-dashed line is the survival curve for patients who did not receive interferon; and the shaded region around the dashed line is the corresponding 95% HPD intervals.

Output 52.7.7 Predicted Survival Probability Curves with 95% HPD Intervals



The plot suggests that there is an effect of using interferon because patients who received interferon have sustained better survival probabilities than those who did not. However, the effect might not be very significant, as the 95% credible intervals of the two groups do overlap. For more on these interferon studies, refer to Ibrahim, Chen, and Sinha (2001).

Weibull or Exponential?

Although the evidence from the Weibull model fit shows that the posterior distribution of α has a significant amount of density mass less than 1, suggesting that the Weibull model is a better fit to the data than the exponential model, you might still be interested in comparing the two models more formally. You can use the Bayesian model selection criterion (see the section "Deviance Information Criterion (DIC)" on page 172) to determine which model fits the data better.

The PROC MCMC DIC option requests the calculation of DIC, and the procedure displays the ODS output table DIC. The table includes the posterior mean of the deviation, $\overline{D(\boldsymbol{\theta})}$, deviation at the estimate, $D(\overline{\boldsymbol{\theta}})$, effective number of parameters, p_D, and DIC. It is important to remember that the standardizing term, $p(\mathbf{y})$, which is a function of the data alone, is not taken into account in calculating the DIC. This term is irrelevant only if you compare two models that have the *same* likelihood function. If you do not have identical likelihood functions, using DIC for model selection purposes without taking this standardizing term into account can produce incorrect results. In addition, you want to be careful in interpreting the DIC whenever you use the GENERAL function to construct the log-likelihood, as the case in this example. Using the GENERAL function, you can obtain identical posterior samples with two log-likelihood functions that differ only by a constant. This difference translates to a difference in the DIC calculation, which could be very misleading.

If $\alpha = 1$, the Weibull likelihood is identical to the exponential likelihood. It is safe in this case to directly compare DICs from these two models. However, if you do not want to work out the mathematical detail or you are uncertain of the equivalence, a better way of comparing the DICs is to run the Weibull model twice: once with α being a parameter and once with $\alpha = 1$. This ensures that the likelihood functions are the same, and the DIC comparison is meaningful.

The following statements fit a Weibull model:

```
title 'Model Comparison between Weibull and Exponential';
proc mcmc data=e1684 outpost=weisurvout nmc=10000 seed=4861 dic;
   ods select dic;
   parms alpha 1 (beta0 beta1) 0;
   prior beta: ~ normal(0, var=10000);
   prior alpha ~ gamma(0.001,is=0.001);

   lambda = beta0 + beta1*ifn;
   llike  = v*(log(alpha) + (alpha-1)*log(t) + lambda) -
            exp(lambda)*(t**alpha);
   model general(llike);
run;
```

The DIC option requests the calculation of DIC, and the table is displayed is displayed in Output 52.7.8:

Output 52.7.8 DIC Table from the Weibull Model

```
                Model Comparison between Weibull and Exponential

                              The MCMC Procedure

                       Deviance Information Criterion

           Dbar (posterior mean of deviance)                 858.623
           Dmean (deviance evaluated at posterior mean)      855.633
           pD (effective number of parameters)                 2.990
           DIC (smaller is better)                           861.614

           The [D]GENERAL function is used in this program. To make
           meaningful comparisons, you must ensure that all
           [D]GENERAL functions include appropriate normalizing
           constants. Otherwise, DIC comparisons can be misleading.
```

The note in Output 52.7.8 reminds you of the importance of ensuring identical likelihood functions when you use the GENERAL function. The DIC value is 861.6.

Based on the same set of code, the following statements fit an exponential model by setting $\alpha = 1$:

```
proc mcmc data=e1684 outpost=expsurvout nmc=10000 seed=4861 dic;
   ods select dic;
   parms beta0 beta1 0;
   prior beta: ~ normal(0, var=10000);
   begincnst;
      alpha = 1;
   endcnst;

   lambda = beta0 + beta1*ifn;
   llike  = v*(log(alpha) + (alpha-1)*log(t) + lambda) -
            exp(lambda)*(t**alpha);
   model general(llike);
run;
```

Output 52.7.9 displays the DIC table.

Output 52.7.9 DIC Table from the Exponential Model

```
                Model Comparison between Weibull and Exponential

                                The MCMC Procedure

                        Deviance Information Criterion

           Dbar (posterior mean of deviance)                 870.133
           Dmean (deviance evaluated at posterior mean)      868.190
           pD (effective number of parameters)                 1.943
           DIC (smaller is better)                           872.075

           The [D]GENERAL function is used in this program. To make
           meaningful comparisons, you must ensure that all
           [D]GENERAL functions include appropriate normalizing
           constants. Otherwise, DIC comparisons can be misleading.
```

The DIC value of 872.075 is greater than 861. A smaller DIC indicates a better fit to the data; hence, you can conclude that the Weibull model is more appropriate for this data set. You can see the equivalencing of the exponential model you fitted in "Exponential Survival Model" on page 3635 by running the following comparison.

The following statements are taken from the section "Exponential Survival Model" on page 3635, and they fit the same exponential model:

```
proc mcmc data=e1684 outpost=expsurvout1 nmc=10000 seed=4861 dic;
   ods select none;
   parms (beta0 beta1) 0;
   prior beta: ~ normal(0, sd = 10000);
   l_h = beta0 + beta1*ifn;
   llike = v*(l_h) -  t*exp(l_h);
   model general(llike);
run;

proc compare data=expsurvout compare=expsurvout1;
   var beta0 beta1;
run;
```

The posterior samples of beta0 and beta1 in the data set expsurvout1 are identical to those in the data set expsurvout. The comparison results are not shown here.

Example 52.8: Cox Models

This example has two purposes. One is to illustrate how to use PROC MCMC to fit a Cox proportional hazard model. Specifically, two models are considered: time independent and time dependent models. However, note that it is much easier to fit a Bayesian Cox model by specifying the BAYES statement in PROC PHREG (see Chapter 64, "The PHREG Procedure"). If you are interested only in fitting a Cox regression survival model, you should use PROC PHREG.

The second objective of this example is to demonstrate how to model data that are not independent. That is the case where the likelihood for observation i depends on other observations in the data set. In other words, if you work with a likelihood function that cannot be broken down simply as $L(\mathbf{y}) = \prod_i^n L(y_i)$, you can use this example for illustrative purposes. By default, PROC MCMC assumes that the programming statements and model specification is intended for a single row of observations in the data set. The Cox model is chosen because the complexity in the data structure requires more elaborate coding.

The Cox proportional hazard model is widely used in the analysis of survival time, failure time, or other duration data to explain the effect of exogenous explanatory variables. The data set used in this example is taken from Krall, Uthoff, and Harley (1975), who analyzed data from a study on myeloma in which researchers treated 65 patients with alkylating agents. Of those patients, 48 died during the study and 17 survived. The following statements generate the data set that is used in this example:

```
data Myeloma;
   input Time Vstatus LogBUN HGB Platelet Age LogWBC Frac
         LogPBM Protein SCalc;
   label Time='survival time'
         VStatus='0=alive 1=dead';
   datalines;
 1.25  1  2.2175   9.4  1  67  3.6628  1  1.9542  12  10
 1.25  1  1.9395  12.0  1  38  3.9868  1  1.9542  20  18
 2.00  1  1.5185   9.8  1  81  3.8751  1  2.0000   2  15

... more lines ...

77.00  0  1.0792  14.0  1  60  3.6812  0  0.9542   0  12
;
proc sort data = Myeloma;
   by descending time;
run;

data _null_;
   set Myeloma nobs=_n;
   call symputx('N', _n);
   stop;
run;
```

The variable Time represents the survival time in months from diagnosis. The variable VStatus consists of two values, 0 and 1, indicating whether the patient was alive or dead, respectively, at the end of the study. If the value of VStatus is 0, the corresponding value of Time is censored. The variables thought to be related to survival are LogBUN (log(BUN) at diagnosis), HGB (hemoglobin at diagnosis), Platelet (platelets at diagnosis: 0=abnormal, 1=normal), Age (age at diagnosis in years), LogWBC (log(WBC) at diagnosis), Frac (fractures at diagnosis: 0=none, 1=present), LogPBM (log percentage of plasma cells in bone marrow), Protein (proteinuria at diagnosis), and SCalc (serum calcium at diagnosis). Interest lies in identifying important prognostic factors from these explanatory variables. In addition, there are 65 (&n) observations in the data set Myeloma. The likelihood

used in these examples is the Brewslow likelihood:

$$L(\beta) = \prod_{i=1}^{n} \left[\prod_{j=1}^{d_i} \frac{\exp(\beta' Z_j(t_i))}{\sum_{l \in \mathcal{R}_i} \exp(\beta' Z_l(t_i))} \right]^{v_i}$$

where

- β is the vector parameters
- n is the total number of observations in the data set
- t_i is the ith time, which can be either event time or censored time
- $Z_l(t)$ is the vector explanatory variables for the lth individual at time t
- d_i is the multiplicity of failures at t_i. If there are no ties in time, d_i is 1 for all i.
- $\mathcal{R}_i$ is the risk set for the ith time t_i, which includes all observations that have survival time greater than or equal to t_i
- v_i indicates whether the patient is censored. The value 0 corresponds to censoring. Note that the censored time t_i enters the likelihood function only through the formation of the risk set $\mathcal{R}_i$.

Priors on the coefficients are independent normal priors with very large variance (1e6). Throughout this example, the symbol bZ represents the regression term $\beta' Z_j(t_i)$ in the likelihood, and the symbol S represents the term $\sum_{l \in \mathcal{R}_i} \exp(\beta' Z_l(t_i))$.

Time Independent Model

The regression model considered in this example uses the following formula:

$$\begin{aligned} \beta' Z_j &= \beta_1 \text{logbun} + \beta_2 \text{hgb} + \beta_3 \text{platelet} + \beta_4 \text{age} + \\ &\quad \beta_5 \text{logwbc} + \beta_6 \text{frac} + \beta_7 \text{logpbm} + \beta_8 \text{protein} + \beta_9 \text{scalc} \end{aligned}$$

The hard part of coding this in PROC MCMC is the construction of the risk set $\mathcal{R}_i$. $\mathcal{R}_i$ contains all observations that have survival time greater than or equal to t_i. First suppose that there are no ties in time. Sorting the data set by the variable time into descending order gives you $\mathcal{R}_i$ that is in the right order. Observation i's risk set consists of all data points j such that $j <= i$ in the data set. You can cumulatively increment S in the SAS statements.

With potential ties in time, at observation i, you need to know whether any subsequent observations, $i + 1$ and so on, have the same survival time as t_i. Suppose that the ith, the $i + 1$th, and the $i + 2$th observations all have the same survival time; all three of them need to be included in the risk set calculation. This means that to calculate the likelihood for some observations, you need to access both the previous and subsequent observations in the data set. There are two ways to do this. One is to use the LAG function; the other is to use the option JOINTMODEL.

The LAG function returns values from a queue (see *SAS Language Reference: Dictionary*). So for the ith observation, you can use LAG1 to access variables from the previous row in the data set. You want to compare the lag1 value of time with the current time value. Depending on whether the two time values are equal, you can add correction terms in the calculation for the risk set S.

The following statements sort the data set by time into descending order, with the largest survival time on top:

```
title 'Cox Model with Time Independent Covariates';
proc freq data=myeloma;
   ods select none;
   tables time / out=freqs;
run;

proc sort data = freqs;
   by descending time;
run;

data myelomaM;
   set myeloma;
   ind = _N_;
run;
```

The following statements run PROC MCMC and produce Output 52.8.1:

```
proc mcmc data=myelomaM outpost=outi nmc=50000 ntu=3000 seed=1;
   ods select PostSummaries PostIntervals;
   array beta[9];
   parms beta: 0;
   prior beta: ~ normal(0, var=1e6);

   bZ = beta1 * LogBUN + beta2 * HGB + beta3 * Platelet
      + beta4 * Age + beta5 * LogWBC + beta6 * Frac +
      beta7 * LogPBM + beta8 * Protein + beta9 * SCalc;

   if ind = 1 then do;        /* first observation         */
      S = exp(bZ);
      l = vstatus * bZ;
      v = vstatus;
   end;
   else if (1 < ind < &N) then do;
      if (lag1(time) ne time) then do;
         l = vstatus * bZ;
         l = l - v * log(S);  /* correct the loglike value   */
         v = vstatus;         /* reset v count value         */
         S = S + exp(bZ);
      end;
      else do;                /* still a tie                 */
         l = vstatus * bZ;
         S = S + exp(bZ);
         v = v + vstatus;     /* add # of nonsensored values */
      end;
   end;
```

```
   else do;                    /* last observation            */
      if (lag1(time) ne time) then do;
         l = - v * log(S);     /* correct the loglike value   */
         S = S + exp(bZ);
         l = l + vstatus * (bZ - log(S));
      end;
      else do;
         S = S + exp(bZ);
         l = vstatus * bZ - (v + vstatus) * log(S);
      end;
   end;
   model general(l);
run;
```

The symbol bZ is the regression term, which is independent of the time variable. The symbol ind indexes observation numbers in the data set. The symbol S keeps track of the risk set term for every observation. The symbol l calculates the log likelihood for each observation. Note that the value of l for observation ind is not necessarily the correct log likelihood value for that observation, especially in cases where the observation ind is in the tied times. Correction terms are added to subsequent values of l when the time variable becomes different in order to make up the difference. The total sum of l calculated over the entire data set is correct. The symbol v keeps track of the sum of vstatus, as censored data do not enter the likelihood and need to be taken out.

You use the function LAG1 to detect if two adjacent time values are different. If they are, you know that the current observation is in a different risk set than the last one. You then need to add a correction term to the log likelihood value of l. The IF-ELSE statements break the observations into three parts: the first observation, the last observation and everything in the middle.

Output 52.8.1 Summary Statistics on Cox Model with Time Independent Explanatory Variables and Ties in the Survival Time, Using PROC MCMC

Cox Model with Time Independent Covariates

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
beta1	50000	1.7600	0.6441	1.3275	1.7651	2.1947
beta2	50000	-0.1308	0.0720	-0.1799	-0.1304	-0.0817
beta3	50000	-0.2017	0.5148	-0.5505	-0.1965	0.1351
beta4	50000	-0.0126	0.0194	-0.0257	-0.0128	0.000641
beta5	50000	0.3373	0.7256	-0.1318	0.3505	0.8236
beta6	50000	0.3992	0.4337	0.0973	0.3864	0.6804
beta7	50000	0.3749	0.4861	0.0464	0.3636	0.6989
beta8	50000	0.0106	0.0271	-0.00723	0.0118	0.0293
beta9	50000	0.1272	0.1064	0.0579	0.1300	0.1997

Output 52.8.1 *continued*

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
beta1	0.050	0.4649	3.0214	0.5117	3.0465
beta2	0.050	−0.2704	0.0114	−0.2746	0.00524
beta3	0.050	−1.2180	0.8449	−1.2394	0.7984
beta4	0.050	−0.0501	0.0257	−0.0512	0.0245
beta5	0.050	−1.1233	1.7232	−1.1124	1.7291
beta6	0.050	−0.4136	1.2970	−0.4385	1.2575
beta7	0.050	−0.5551	1.3593	−0.5423	1.3689
beta8	0.050	−0.0451	0.0618	−0.0451	0.0616
beta9	0.050	−0.0933	0.3272	−0.0763	0.3406

An alternative to using the LAG function is to use the PROC option JOINTMODEL. With this option, the log-likelihood function you specify applies not to a single observation but to the entire data set. See "Modeling Joint Likelihood" on page 3556 for details on how to properly use this option. The basic idea is that you store all necessary data set variables in arrays and use only the arrays to construct the log likelihood of the entire data set. This approach works here because for every observation i, you can use index to access different values of arrays to construct the risk set S. To use the JOINTMODEL option, you need to do some additional data manipulation. You want to create a stop variable for each observation, which indicates the observation number that should be included in S for that observation. For example, if observations 4, 5, 6 all have the same survival time, the stop value for all of them is 6.

The following statements generate a new data set myelomaM that contains the stop variable:

```
data myelomaM;
   merge myelomaM freqs(drop=percent);
   by descending time;
   retain stop;
   if first.time then do;
      stop = _n_ + count - 1;
   end;
run;
```

The following SAS program fits the same Cox model by using the JOINTMODEL option:

```
data a;
   run;

proc mcmc data=a outpost=outa nmc=50000 ntu=3000 seed=1 jointmodel;
   ods select none;
      array beta[9];
      array data[1] / nosymbols;
      array timeA[1] / nosymbols;
      array vstatusA[1] / nosymbols;
      array stopA[1] / nosymbols;
      array bZ[&n];
      array S[&n];
```

```
      begincnst;
      rc = read_array("myelomam", data, "logbun", "hgb", "platelet",
                      "age", "logwbc", "frac", "logpbm", "protein", "scalc");
      rc = read_array("myelomam", timeA, "time");
      rc = read_array("myelomam", vstatusA, "vstatus");
      rc = read_array("myelomam", stopA, "stop");
      endcnst;

      parms (beta:) 0;
      prior beta: ~ normal(0, var=1e6);

      jl = 0;
      /* calculate each bZ and cumulatively adding S as if there are no ties.*/
      call mult(data, beta, bZ);
      S[1] = exp(bZ[1]);
      do i = 2 to &n;
         S[i] = S[i-1] + exp(bZ[i]);
      end;

      do i = 1 to &n;
         /* correct the S[i] term, when needed. */
         if(stopA[i] > i) then do;
            do j = (i+1) to stopA[i];
               S[i] = S[i] + exp(bZ[j]);
            end;
         end;
         jl = jl + vstatusA[i] * (bZ[i] - log(S[i]));
      end;
      model general(jl);
run;
ods select all;
```

No output tables were produced because this PROC MCMC run produces identical posterior samples as does the previous example.

Because the JOINTMODEL option is specified here, you do not need to specify myelomaM as the input data set. An empty data set a is used to speed up the procedure run.

Multiple ARRAY statements allocate array symbols that are used to store the parameters (beta), the response and the covariates (data, timeA, vstatusA, and stopA), and the work space (bZ and S). The data, timeA, vstatusA, and stopA arrays are declared with the /NOSYMBOLS option. This option enables PROC MCMC to dynamically resize these arrays to match the dimensions of the input data set. See the section "READ_ARRAY Function" on page 3510. The bZ and S arrays store the regression term and the risk set term for every observation.

The BEGINCNST and ENDCNST statements enclose programming statements that read the data set variables into these arrays. The rest of the programming statements construct the log likelihood for the entire data set.

The CALL MULT function calculates the regression term in the model and stores the result in the array bZ. In the first DO loop, you sum the risk set term S as if there are no ties in time. This underevaluates some of the S elements. For observations that have a tied time, you make the necessary correction to the corresponding S values. The correction takes place in the second DO

loop. Any observation that has a tied time also has a stopA[i] that is different from i. You add the right terms to S and sum up the joint log likelihood jl. The MODEL statement specifies that the log likelihood takes on the value of jl.

To see that you get identical results from these two approaches, use PROC COMPARE to compare the posterior samples from two runs:

```
proc compare data=outi compare=outa;
   ods select comparesummary;
   var beta1-beta9;
run;
```

The output is not shown here.

Generally, the JOINTMODEL option can be slightly faster than using the default setup. The savings come from avoiding the overhead cost of accessing the data set repeatedly at every iteration. However, the speed gain is not guaranteed because it largely depends on the efficiency of your programs.

PROC PHREG fits the same model, and you get very similar results to PROC MCMC. The following statements run PROC PHREG and produce Output 52.8.2:

```
proc phreg data=Myeloma;
   ods select PostSummaries PostIntervals;
   ods output posteriorsample = phout;
   model Time*VStatus(0)=LogBUN HGB Platelet Age LogWBC
                         Frac LogPBM Protein Scalc;
   bayes seed=1 nmc=10000;
run;
```

Output 52.8.2 Summary Statistics for Cox Model with Time Independent Explanatory Variables and Ties in the Survival Time, Using PROC PHREG

Cox Model with Time Independent Covariates

The PHREG Procedure

Bayesian Analysis

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
LogBUN	10000	1.7610	0.6593	1.3173	1.7686	2.2109
HGB	10000	−0.1279	0.0727	−0.1767	−0.1287	−0.0789
Platelet	10000	−0.2179	0.5169	−0.5659	−0.2360	0.1272
Age	10000	−0.0130	0.0199	−0.0264	−0.0131	0.000492
LogWBC	10000	0.3150	0.7451	−0.1718	0.3321	0.8253
Frac	10000	0.3766	0.4152	0.0881	0.3615	0.6471
LogPBM	10000	0.3792	0.4909	0.0405	0.3766	0.7023
Protein	10000	0.0102	0.0267	−0.00745	0.0106	0.0283
SCalc	10000	0.1248	0.1062	0.0545	0.1273	0.1985

Output 52.8.2 *continued*

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
LogBUN	0.050	0.4418	3.0477	0.4107	2.9958
HGB	0.050	−0.2718	0.0150	−0.2801	0.00599
Platelet	0.050	−1.1952	0.8296	−1.1871	0.8341
Age	0.050	−0.0514	0.0259	−0.0519	0.0251
LogWBC	0.050	−1.2058	1.7228	−1.1783	1.7483
Frac	0.050	−0.3995	1.2316	−0.4273	1.2021
LogPBM	0.050	−0.5652	1.3671	−0.5939	1.3241
Protein	0.050	−0.0437	0.0611	−0.0405	0.0637
SCalc	0.050	−0.0935	0.3264	−0.0846	0.3322

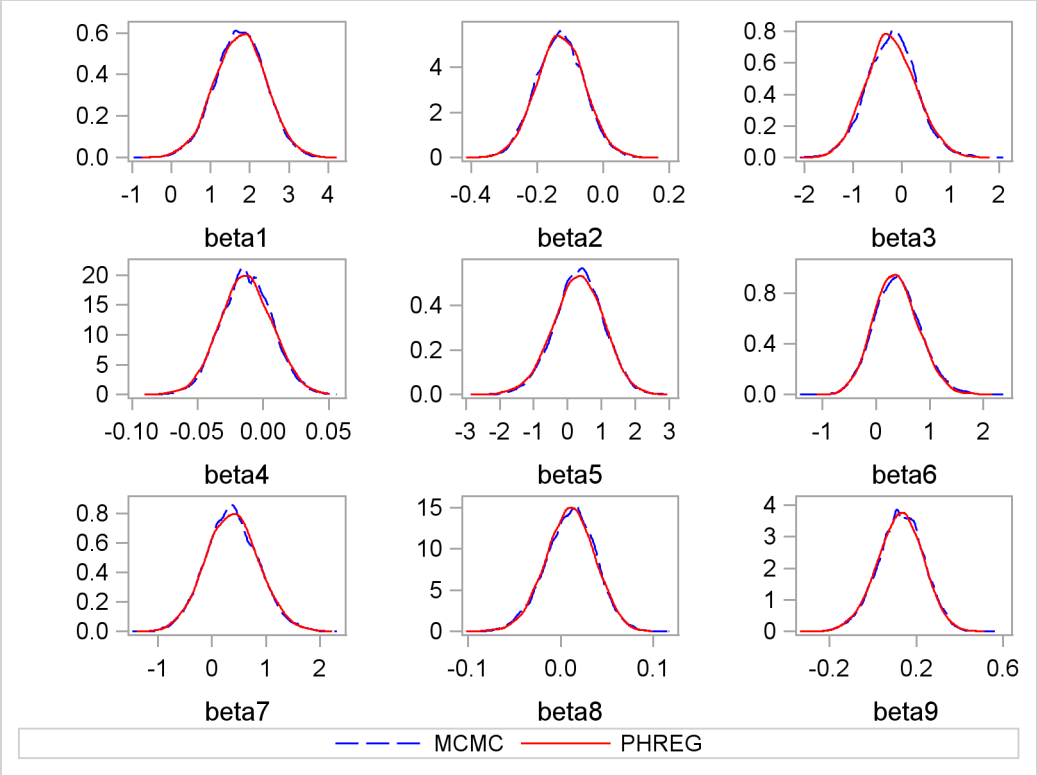
Output 52.8.3 shows kernel density plots that compare the posterior marginal distributions of all the beta parameters from the PROC MCMC run and the PROC PHREG run. The following statements generate the comparison:

```
proc kde data=outi;
   ods exclude all;
   univar beta1 beta2 beta3 beta4 beta5 beta6 beta7 beta8 beta9
      / out=m1 (drop=count);
run;
ods exclude none;

%reshape(m1, mcmc, suffix1=, suffix2=md);

data phout;
   set phout(drop = LogPost Iteration);
   beta1 = LogBUN;  beta2 = HGB;      beta3 = Platelet;
   beta4 = Age;     beta5 = LogWBC;   beta6 = Frac;
   beta7 = LogPBM;  beta8 = Protein;  beta9 = SCalc;
   drop LogBUN HGB Platelet Age LogWBC Frac LogPBM Protein SCalc;
run;

proc kde data=phout;
   ods exclude all;
   univar beta1 beta2 beta3 beta4 beta5 beta6 beta7 beta8 beta9
      / out=m2 (drop=count);
run;
ods exclude none;

%reshape(m2, phreg, suffix1=p, suffix2=pd);

data all;
   merge mcmc phreg;
run;

proc template;
   define statgraph threebythree;
      %macro plot;
```

```
            begingraph;
               layout lattice / rows=3 columns=3;
                  %do i = 1 %to 9;
                     layout overlay /yaxisopts=(label=" ");
                        seriesplot y=beta&i.md x=beta&i
                           / connectorder=xaxis
                             lineattrs=(pattern=mediumdash color=blue)
                           legendlabel = "MCMC" name="MCMC";
                        seriesplot y=beta&i.pd x=beta&i.p
                           / connectorder=xaxis lineattrs=(color=red)
                           legendlabel = "PHREG" name="PHREG";
                     endlayout;
                  %end;
                  Sidebar / align = bottom;
                     discretelegend "MCMC" "PHREG";
                  endsidebar;
               endlayout;
            endgraph;
         %mend; %plot;
      end;
   run;

   proc sgrender data=all template=threebythree;
      title "Kernel Density Comparison";
   run;
```

The macro %RESHAPE is defined in the example "Logistic Regression Random-Effects Model" on page 3614. The posterior densities are almost identical to one another.

Output 52.8.3 Comparing Estimates from PROC MCMC and PROC PHREG



Time Dependent Model

To model $Z_i(t_i)$ as a function of the survival time, you can relate time t_i to covariates by using this formula:

$$\beta' Z_j(t_i) = (\beta_1 + \beta_2 t_i)\text{logbun} + (\beta_3 + \beta_4 t_i)\text{hgb} + (\beta_5 + \beta_6 t_i)\text{platelet}$$

For illustrational purposes, only three explanatory variables, LOGBUN, HBG, and PLATELET, are used in this example.

Since $Z_j(t_i)$ depends on t_i, every term in the summation of $\sum_{l \in \mathcal{R}_i} \exp(\beta' Z_l(t_i))$ is a product of the current time t_i and all observations that are in the risk set. You can use the JOINTMODEL option, as in the last example, or you can modify the input data set such that every row contains not only the current observation but also all observations that are in the corresponding risk set. When you construct the log likelihood for each observation, you have all the relevant data at your disposal.

The following statements illustrate how you can create a new data set with different risk sets at different rows:

```
title 'Cox Model with Time Dependent Covariates';
ods select none;
proc freq data=myeloma;
   tables time / out=freqs;
run;
ods select all;
```

```
proc sort data = freqs;
   by descending time;
run;

data myelomaM;
   set myeloma;
   ind = _N_;
run;

data myelomaM;
   merge myelomaM freqs(drop=percent); by descending time;
   retain stop;
   if first.time then do;
      stop = _n_ + count - 1;
   end;
run;

%macro array(list);
   %global mcmcarray;
   %let mcmcarray = ;
   %do i = 1 %to 32000;
      %let v = %scan(&list, &i, %str( ));
      %if %nrbquote(&v) ne %then %do;
         array _&v[&n];
         %let mcmcarray = &mcmcarray array _&v[&n] _&v.1 - _&v.&n%str(;);
            do i = 1 to stop;
               set myelomaM(keep=&v) point=i;
               _&v[i] = &v;
            end;
      %end;
      %else %let i = 32001;
   %end;
%mend;

data z;
   set myelomaM;
   %array(logbun hgb platelet);
   drop vstatus logbun hgb platelet count stop;
run;

data myelomaM;
   merge myelomaM z; by descending time;
run;
```

The data set myelomaM contains 65 observations and 209 variables. For each observation, you see added variables stop, _logbun1 through _logbun65, _hgb1 through _hgb65, and _platelet1 through _platelet65. The variable stop indicates the number of observations that are in the risk set of the current observation. The rest are transposed values of model covariates of the entire data set. The data set contains a number of missing values. This is due to the fact that only the relevant observations are kept, such as _logbun*1* to _logbun*stop*. The rest of the cells are filled in with missing values. For example, the first observation has a unique survival time of 92 and stop is 1, making it a risk set of itself. You see nonmissing values only in _logbun1, _hgb1, and _platelet1.

The following statements fit the Cox model by using PROC MCMC:

```
proc mcmc data=myelomaM outpost=outi nmc=50000 ntu=3000 seed=17
          missing=ac;
   ods select PostSummaries PostIntervals;
   array beta[6];
   &mcmcarray
   parms (beta:) 0;
   prior beta: ~ normal(0, prec=1e-6);

   b = (beta1 + beta2 * time) * logbun +
       (beta3 + beta4 * time) * hgb +
       (beta5 + beta6 * time) * platelet;
   S = 0;
   do i = 1 to stop;
      S = S + exp( (beta1 + beta2 * time) * _logbun[i] +
                   (beta3 + beta4 * time) * _hgb[i] +
                   (beta5 + beta6 * time) * _platelet[i]);
   end;
   loglike = vstatus * (b - log(S));

   model general(loglike);
run;
```

Note that the option MISSING= is set to AC. This is due to missing cells in the input data set. You must use this option so that PROC MCMC retains observations that contain missing values.

The macro variable &mcmcarray is defined in the earlier part in this example. You can use a **%put** statement to print its value:

```
%put &mcmcarray;
```

This statement prints the following:

```
array _logbun[65] _logbun1 - _logbun65; array _hgb[65] _hgb1 - _hgb65; array
_platelet[65] _platelet1 - _platelet65;
```

The macro uses the ARRAY statement to allocate three arrays, each of which links their corresponding data set variables. This makes it easier to reference these data set variables in the program. The PARMS statement puts all the parameters in the same block. The PRIOR statement gives them normal priors with large variance. The symbol b is the regression term, and S is cumulatively added from 1 to stop for each observation in the DO loop. The symbol loglike completes the construction of log likelihood for each observation and the MODEL statement completes the model specification.

Posterior summary and interval statistics are shown in Output 52.8.4.

Output 52.8.4 Summary Statistics on Cox Model with Time Dependent Explanatory Variables and Ties in the Survival Time, Using PROC MCMC

Cox Model with Time Dependent Covariates

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
beta1	50000	3.2397	0.8226	2.6835	3.2413	3.7830
beta2	50000	−0.1411	0.0471	−0.1722	−0.1406	−0.1092
beta3	50000	−0.0369	0.1017	−0.1064	−0.0373	0.0315
beta4	50000	−0.00409	0.00360	−0.00656	−0.00408	−0.00167
beta5	50000	0.3548	0.7359	−0.1634	0.3530	0.8445
beta6	50000	−0.0417	0.0359	−0.0661	−0.0423	−0.0181

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
beta1	0.050	1.6399	4.8667	1.6664	4.8752
beta2	0.050	−0.2351	−0.0509	−0.2294	−0.0458
beta3	0.050	−0.2337	0.1642	−0.2272	0.1685
beta4	0.050	−0.0111	0.00282	−0.0112	0.00264
beta5	0.050	−1.0317	1.8202	−1.0394	1.8100
beta6	0.050	−0.1107	0.0295	−0.1122	0.0269

You can also use the option JOINTMODEL to get the same inference and avoid transposing the data for every observation:

```
proc mcmc data=myelomaM outpost=outa nmc=50000 ntu=3000 seed=17 jointmodel;
   ods select none;
   array beta[6];      array timeA[&n];      array vstatusA[&n];
   array logbunA[&n];  array hgbA[&n];       array plateletA[&n];
   array stopA[&n];    array bZ[&n];         array S[&n];

   begincnst;
      timeA[ind]=time;          vstatusA[ind]=vstatus;
      logbunA[ind]=logbun;      hgbA[ind]=hgb;
      plateletA[ind]=platelet;  stopA[ind]=stop;
   endcnst;

   parms (beta:) 0;
   prior beta: ~ normal(0, prec=1e-6);

   jl = 0;
   do i = 1 to &n;
      v1 = beta1 + beta2 * timeA[i];
      v2 = beta3 + beta4 * timeA[i];
      v3 = beta5 + beta6 * timeA[i];
      bZ[i] = v1 * logbunA[i] + v2 * hgbA[i] + v3 * plateletA[i];
```

```
      /* sum over risk set without considering ties in time. */
      S[i] = exp(bZ[i]);
      if (i > 1) then do;
         do j = 1 to (i-1);
            b1 = v1 * logbunA[j] + v2 * hgbA[j] + v3 * plateletA[j];
            S[i] = S[i] + exp(b1);
         end;
      end;
   end;

   /* make correction to the risk set due to ties in time. */
   do i = 1 to &n;
      if(stopA[i] > i) then do;
         v1 = beta1 + beta2 * timeA[i];
         v2 = beta3 + beta4 * timeA[i];
         v3 = beta5 + beta6 * timeA[i];
         do j = (i+1) to stopA[i];
            b1 = v1 * logbunA[j] + v2 * hgbA[j] + v3 * plateletA[j];
            S[i] = S[i] + exp(b1);
         end;
      end;
      jl = jl + vstatusA[i] * (bZ[i] - log(S[i]));
   end;
   model general(jl);
run;
```

The multiple ARRAY statements allocate array symbols that are used to store the parameters (beta), the response (timeA), the covariates (vstatusA, logbunA, hgbA, plateletA, and stopA), and work space (bZ and S). The bZ and S arrays store the regression term and the risk set term for every observation. Programming statements in the BEGINCNST and ENDCNST statements input the response and covariates from the data set to the arrays.

Using the same technique shown in the example "Time Independent Model" on page 3649, the next DO loop calculates the regression term and corresponding S for every observation, pretending that there are no ties in time. This means that the risk set for observation i involves only observation 1 to i. The correction terms are added to the corresponding S[i] in the second DO loop, conditional on whether the stop variable is greater than the observation count itself. The symbol jl cumulatively adds the log likelihood for the entire data set, and the MODEL statement specifies the joint log-likelihood function.

The following statements run PROC COMPARE and show that the output data set outa contains identical posterior samples as outi:

```
proc compare data=outi compare=outa;
   ods select comparesummary;
   var beta1-beta6;
run;
```

The results are not shown here.

The following statements use PROC PHREG to fit the same time dependent Cox model:

```
proc phreg data=Myeloma;
   ods select PostSummaries PostIntervals;
   ods output posteriorsample = phout;
   model Time*VStatus(0)=LogBUN z2 hgb z3 platelet z4;
   z2 = Time*logbun;
   z3 = Time*hgb;
   z4 = Time*platelet;
   bayes seed=1 nmc=10000;
run;
```

Coding is simpler than PROC MCMC. See Output 52.8.5 for posterior summary and interval statistics:

Output 52.8.5 Summary Statistics on Cox Model with Time Dependent Explanatory Variables and Ties in the Survival Time, Using PROC PHREG

Cox Model with Time Dependent Covariates

The PHREG Procedure

Bayesian Analysis

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
LogBUN	10000	3.2423	0.8311	2.6838	3.2445	3.7929
z2	10000	−0.1401	0.0482	−0.1723	−0.1391	−0.1069
HGB	10000	−0.0382	0.1009	−0.1067	−0.0385	0.0297
z3	10000	−0.00407	0.00363	−0.00652	−0.00404	−0.00162
Platelet	10000	0.3778	0.7524	−0.1500	0.3389	0.8701
z4	10000	−0.0419	0.0364	−0.0660	−0.0425	−0.0178

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
LogBUN	0.050	1.6059	4.8785	1.5925	4.8582
z2	0.050	−0.2361	−0.0494	−0.2354	−0.0492
HGB	0.050	−0.2343	0.1598	−0.2331	0.1603
z3	0.050	−0.0113	0.00297	−0.0109	0.00322
Platelet	0.050	−0.9966	1.9464	−1.1342	1.7968
z4	0.050	−0.1124	0.0296	−0.1142	0.0274

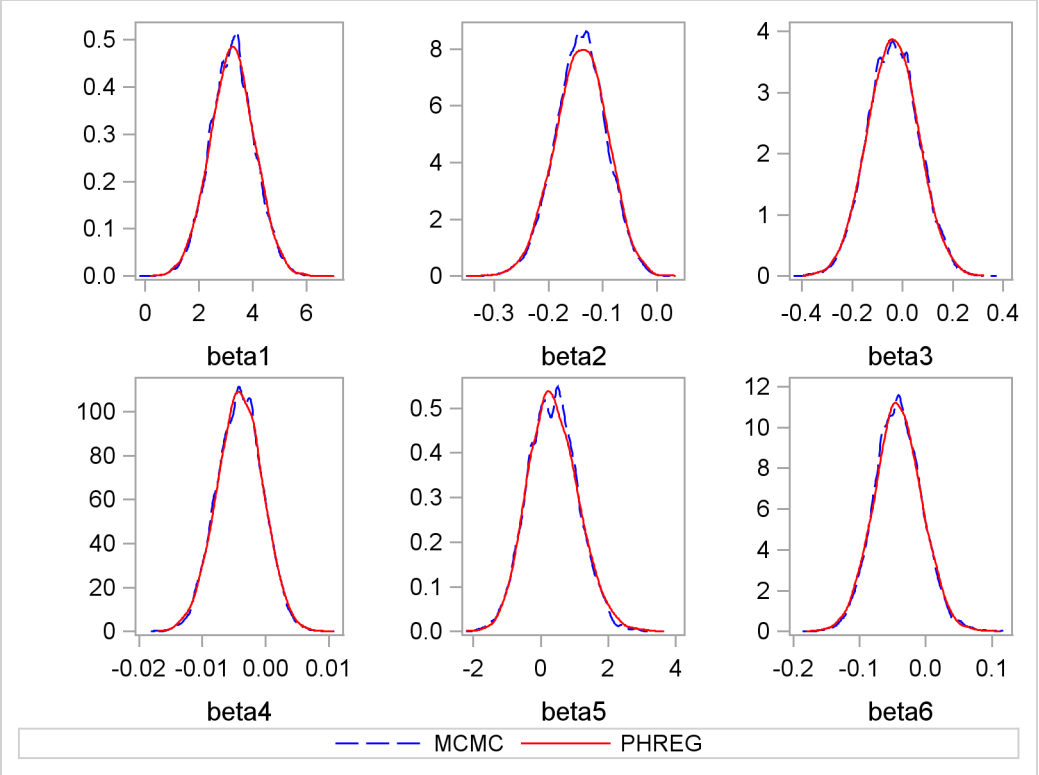
Output 52.8.6 shows a kernel density comparison plot that compares posterior marginal distributions of all the beta parameters from the PROC MCMC run and the PROC PHREG run. The following statements generate Output 52.8.6:

```
proc kde data=outi;
   ods exclude all;
   univar beta1 beta2 beta3 beta4 beta5 beta6 / out=m1 (drop=count);
run;
ods exclude none;

%reshape(m1, mcmc, suffix1=, suffix2=md);
```

```
data phout;
   set phout(drop = LogPost Iteration);
   beta1 = LogBUN;  beta2 = z2;       beta3 = HGB;
   beta4 = z3;      beta5 = Platelet; beta6 = z4;
   drop LogBUN HGB Platelet z2-z4;
run;

proc kde data=phout;
   ods exclude all;
   univar beta1 beta2 beta3 beta4 beta5 beta6 / out=m2 (drop=count);
run;
ods exclude none;

%reshape(m2, phreg, suffix1=p, suffix2=pd);

data all;
   merge mcmc phreg;
run;

proc template;
   define statgraph twobythree;
      %macro plot;
         begingraph;
            layout lattice / rows=2 columns=3;
                %do i = 1 %to 6;
                  layout overlay /yaxisopts=(label=" ");
                     seriesplot y=beta&i.md x=beta&i
                        / connectorder=xaxis
                          lineattrs=(pattern=mediumdash color=blue)
                        legendlabel = "MCMC" name="MCMC";
                     seriesplot y=beta&i.pd x=beta&i.p
                        / connectorder=xaxis lineattrs=(color=red)
                        legendlabel = "PHREG" name="PHREG";
                  endlayout;
                %end;
                Sidebar / align = bottom;
                   discretelegend "MCMC" "PHREG";
                endsidebar;
            endlayout;
         endgraph;
      %mend; %plot;
   end;
run;

proc sgrender data=all template=twobythree;
   title "Kernel Density Comparison";
run;
```

The macro %RESHAPE is defined in the example "Logistic Regression Random-Effects Model" on page 3614.

Output 52.8.6 Comparing Estimates from PROC MCMC and PROC PHREG



Example 52.9: Normal Regression with Interval Censoring

You can use PROC MCMC to fit failure time data that can be right, left, or interval censored. To illustrate, a normal regression model is used in this example.

Assume that you have the following simple regression model with no covariates:

$$\mathbf{y} = \mu + \sigma\boldsymbol{\epsilon}$$

where $\mathbf{y}$ is a vector of response values (the failure times), μ is the grand mean, σ is an unknown scale parameter, and $\boldsymbol{\epsilon}$ are errors from the standard normal distribution. Instead of observing y_i directly, you only observe a truncated value t_i. If the true y_i occurs after the censored time t_i, it is called *right censoring*. If y_i occurs before the censored time, it is called *left censoring*. A failure time y_i can be censored at both ends, and this is called *interval censoring*. The likelihood for y_i is as follows:

$$p(y_i|\mu) = \begin{cases} \phi(y_i|\mu,\sigma) & \text{if } y_i \text{ is uncensored} \\ S(t_{l,i}|\mu) & \text{if } y_i \text{ is right censored by } t_{l,i} \\ 1 - S(t_{r,i}|\mu) & \text{if } y_i \text{ is left censored by } t_{r,i} \\ S(t_{l,i}|\mu) - S(t_{r,i}|\mu) & \text{if } y_i \text{ is interval censored by } t_{l,i} \text{ and } t_{r,i} \end{cases}$$

where $S(\cdot)$ is the survival function, $S(t) = Pr(T > t)$.

Gentleman and Geyer (1994) uses the following data on cosmetic deterioration for early breast cancer patients treated with radiotherapy:

```
title 'Normal Regression with Interval Censoring';
data cosmetic;
   label tl = 'Time to Event (Months)';
   input tl tr @@;
   datalines;
45  .   6 10   .  7  46  .  46  .   7 16  17  .   7 14
37 44   .  8   4 11  15  .  11 15  22  .  46  .  46  .
25 37  46  .  26 40  46  .  27 34  36 44  46  .  36 48
37  .  40  .  17 25  46  .  11 18  38  .   5 12  37  .
 .  5  18  .  24  .  36  .   5 11  19 35  17 25  24  .
32  .  33  .  19 26  37  .  34  .  36  .
;
```

The data consist of time interval endpoints (in months). Nonmissing equal endpoints (tl = tr) indicates uncensoring; a nonmissing lower endpoint (tl $\neq$.) and a missing upper endpoint (tr = .) indicates right censoring; a missing lower endpoint (tl = .) and a nonmissing upper endpoint (tr $\neq$.) indicates left censoring; and nonmissing unequal endpoints (tl $\neq$ tr) indicates interval censoring.

With this data set, you can consider using proper but diffuse priors on both μ and σ, for example:

$$\begin{aligned} \pi(\mu) &\propto \phi(0, \text{sd} = 1000) \\ \pi(\sigma) &\propto f_{\Gamma}(0.001, \text{iscale} = 0.001) \end{aligned}$$

where f_{Γ} is the gamma density function.

The following SAS statements fit an interval censoring model and generate Output 52.9.1:

```
proc mcmc data=cosmetic outpost=postout seed=1 nmc=20000 missing=AC;
   ods select PostSummaries PostIntervals;
   parms mu 60 sigma 50;

   prior mu ~ normal(0, sd=1000);
   prior sigma ~ gamma(shape=0.001,iscale=0.001);

   if (tl^=. and tr^=. and tl=tr) then
      llike = logpdf('normal',tr,mu,sigma);
   else if (tl^=. and tr=.) then
      llike = logsdf('normal',tl,mu,sigma);
   else if (tl=. and tr^=.) then
      llike = logcdf('normal',tr,mu,sigma);
   else
      llike = log(sdf('normal',tl,mu,sigma) -
         sdf('normal',tr,mu,sigma));

   model general(llike);
run;
```

Because there are missing cells in the input data, you want to use the MISSING=AC option so that PROC MCMC does not delete any observations that contain missing values. The IF-ELSE statements distinguish different censoring cases for y_i, according to the likelihood. The SAS functions LOGCDF, LOGSDF, LOGPDF, and SDF are useful here. The MODEL statement assigns llike as

the log likelihood to the response. The Markov chain appears to have converged in this example (evidence not shown here), and the posterior estimates are shown in Output 52.9.1.

Output 52.9.1 Interval Censoring

Normal Regression with Interval Censoring

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
mu	20000	41.7807	5.7882	37.7220	41.3468	45.2249
sigma	20000	29.1122	6.0503	24.8774	28.2210	32.4250

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
mu	0.050	32.0499	54.6104	31.3604	53.6115
sigma	0.050	20.0889	43.1335	19.4041	41.6742

Example 52.10: Constrained Analysis

Conjoint analysis uses regression techniques to model consumer preferences and to estimate consumer utility functions. A problem with conventional conjoint analysis is that sometimes your estimated utilities do not make sense. Your results might suggest, for example, that the consumers would prefer to spend more on a product than to spend less. With PROC MCMC, you can specify constraints on the part-worth utilities (parameter estimates). Suppose that the consumer product being analyzed is an off-road motorcycle. The relevant attributes are how large each motorcycle is (less than 300cc, 301–550cc, and more than 551cc), how much it costs (less than $5000, $5001–$6000, $6001–$7000, and more than $7000), whether or not it has an electric starter, whether or not the engine is counter-balanced, and whether the bike is from Japan or Europe. The preference variable is a ranking of the bikes. You could perform an ordinary conjoint analysis with PROC TRANSREG (see Chapter 90, "The TRANSREG Procedure") as follows:

```
title 'Constrained Conjoint Analysis';

options validvarname=any;
proc format;
   value sizef  1 = '< 300cc' 2 = '300-550cc' 3 = '> 551cc';
   value pricef 1 = '< $5000' 2 = '$5000 - $6000'
                3 = '$6001 - $7000' 4 = '> $7000';
   value startf 1 = 'Electric Start' 2 = 'Kick Start';
   value balf   1 = 'Counter Balanced' 2 = 'Unbalanced';
   value orif   1 = 'Japanese' 2 = 'European';
run;

data bikes;
```

```
   input Size Price Start Balance Origin Rank @@;
   format size sizef. price pricef. start startf.
          balance balf. origin orif.;
   datalines;
2 1 2 1 2  3  1 4 2 2 2  7  1 2 1 1 2  6
3 3 1 1 2  1  1 3 2 1 1  5  3 4 2 2 2 12
2 3 2 2 1  9  1 1 1 2 1  8  2 2 1 2 2 10
2 4 1 1 1  4  3 1 1 2 1 11  3 2 2 1 1  2
;

proc transreg data=bikes utilities cprefix=0 lprefix=0;
   ods select Utilities;
   model identity(rank / reflect) =
         class(size price start balance origin / zero=sum);
   output out=coded(drop=intercept) replace;
run;
```

The DATA step reads the experimental design and dependent variable Rank and assigns formats to label the factor levels. PROC TRANSREG is run specifying UTILITIES, which requests a conjoint analysis. The rank variable is reflected around its mean ($1 \rightarrow 12$, $2 \rightarrow 11$, $\ldots$, $12 \rightarrow 1$) so that in the analysis, larger part-worth utilities correspond to higher preference. The OUT=CODED data set contains the reflected ranks and a binary coding of the factors that can be used in other analyses. Refer to Kuhfeld (2004) for more information about conjoint analysis and coding with PROC TRANSREG.

The Utilities table from the conjoint analysis is shown in Output 52.10.1. Notice the part-worth utilities for price. The part-worth utility for < \$5000 is 0.25. As price increases to the \$5000–\$6000 range, utility decreases to -0.5. Then as price increases to the \$6001–\$7000 range, part-worth utility *increases* to 0.5. Finally, for the most expensive bikes, utility decreases again to -0.25. In cases like this, you might want to impose constraints on the solution so that the part-worth utility for price never increases as prices go up.

Output 52.10.1 Ordinary Conjoint Analysis by PROC TRANSREG

Constrained Conjoint Analysis

The TRANSREG Procedure

Utilities Table Based on the Usual Degrees of Freedom

Label	Utility	Standard Error	Importance (% Utility Range)	Variable
Intercept	6.5000	0.95743		Intercept
< 300cc	−0.0000	1.35401	0.000	Class.< 300cc
300−550cc	−0.0000	1.35401		Class.300−550cc
> 551cc	0.0000	1.35401		Class.> 551cc
< $5000	0.2500	1.75891	13.333	Class.< $5000
$5000 − $6000	−0.5000	1.75891		Class.$5000 − $6000
$6001 − $7000	0.5000	1.75891		Class.$6001 − $7000
> $7000	−0.2500	1.75891		Class.> $7000
Electric Start	−0.1250	1.01550	3.333	Class.Electric Start
Kick Start	0.1250	1.01550		Class.Kick Start
Counter Balanced	3.0000	1.01550	80.000	Class.Counter Balanced
Unbalanced	−3.0000	1.01550		Class.Unbalanced
Japanese	−0.1250	1.01550	3.333	Class.Japanese
European	0.1250	1.01550		Class.European

You could run PROC TRANSREG again, specifying monotonicity constraints on the part-worth utilities for price:

```
proc transreg data=bikes utilities cprefix=0 lprefix=0;
   ods select ConservUtilities;
   model identity(rank / reflect) =
         monotone(price / tstandard=center)
         class(size start balance origin / zero=sum);
run;
```

The output from this PROC TRANSREG step is shown in Output 52.10.2.

Output 52.10.2 Constrained Conjoint Analysis by PROC TRANSREG

Constrained Conjoint Analysis

The TRANSREG Procedure

Utilities Table Based on Conservative Degrees of Freedom

Label	Utility	Standard Error	Importance (% Utility Range)	Variable
Intercept	6.5000	0.97658		Intercept
Price	−0.1581	.	7.143	Monotone(Price)
< $5000	0.2500	.		
$5000 – $6000	0.0000	.		
$6001 – $7000	0.0000	.		
> $7000	−0.2500	.		
< 300cc	−0.0000	1.38109	0.000	Class.< 300cc
300–550cc	0.0000	1.38109		Class.300–550cc
> 551cc	0.0000	1.38109		Class.> 551cc
Electric Start	−0.2083	1.00663	5.952	Class.Electric Start
Kick Start	0.2083	1.00663		Class.Kick Start
Counter Balanced	3.0000	0.97658	85.714	Class.Counter Balanced
Unbalanced	−3.0000	0.97658		Class.Unbalanced
Japanese	−0.0417	1.00663	1.190	Class.Japanese
European	0.0417	1.00663		Class.European

This monotonicity constraint is one of the few constraints on the part-worth utilities that you can specify in PROC TRANSREG. In contrast, PROC MCMC allows you to specify any constraint that can be written in the DATA step language. You can perform the restricted conjoint analysis with PROC MCMC by using the coded factors that were output from PROC TRANSREG. The data set is coded.

The likelihood is a simple regression model:

$$\text{rank}_i \sim \text{normal}(\mathbf{x}_i'\boldsymbol{\beta}, \sigma)$$

where rank is the response, the covariates are '< 300cc'n, '300-500cc'n, '< $5000'n, '$5000 - $6000'n, '$6001 - $7000'n, 'Electric Start'n, 'Counter Balanced'n, and Japanese. Note that OPTIONS VALIDVARNAME=ANY allows PROC TRANSREG to create names for the coded variables with blanks and special characters. That is why the name-literal notation ('*variable-name*'n) is used for the input data set variables.

Suppose that there are two constraints you want to put on some of the parameters: one is that the parameters for '< $5000'n, '$5000 - $6000'n, and '$6001 - $7000'n decrease in order, and the other is that the parameter for 'Counter Balanced'n is strictly positive. You can consider a truncated multivariate normal prior as follows:

$$\pi\left(\beta_{\text{'< \$5000'n}}, \beta_{\text{'\$5000 - \$6000'n}}, \beta_{\text{'\$6001 - \$7000'n}}, \beta_{\text{'Counter Balanced'n}}\right) \sim \text{MVN}(0, \sigma\boldsymbol{I})$$

with the following set of constraints:

$$\begin{aligned} \beta_{\text{'< \$5000'n}} &> \beta_{\text{'\$5000 - \$6000'n}} > \beta_{\text{'\$6001 - \$7000'n}} > 0 \\ \beta_{\text{'Counter Balanced'n}} &> 0 \end{aligned}$$

The condition that $\beta_{\text{'\$6001 - \$7000'n}} > 0$ reflects an implied constraint that, by definition, 0 is the utility for the highest price range, > $7000, which is the reference level for the binary coded price variable. The following statements fit the desired model:

```
proc mcmc data=coded outpost=bikesout ntu=3000 nmc=50000 thin=10
          seed=448;
   ods select PostSummaries;
   array sigma[4,4] sigma1-sigma16;
   array mu[4] mu1-mu4;

   begincnst;
      call identity(sigma);
      call mult(sigma, 100, sigma);
      call zeromatrix(mu);
      rc = logmpdfsetsq('v', of sigma1-sigma16);
   endcnst;

   parms intercept pw300cc pw300_550cc pwElectricStart pwJapanese ltau 1;
   parms pw5000 0.3 pw5000_6000 0.2 pw6001_7000 0.1 pwCounterBalanced 1;

   beginnodata;
   prior intercept pw300: pwE: pwJ: ~ normal(0, var=100);
   if (pw5000      >= pw5000_6000 & pw5000_6000 >= pw6001_7000 &
       pw6001_7000 >= 0           & pwCounterBalanced > 0) then
       lp = logmpdfnormal(of mu1-mu4, pw5000, pw5000_6000,
                          pw6001_7000, pwCounterBalanced, 'v');
   else
       lp = .;
   prior pw5000 pw5000_6000 pw6001_7000 pwC: ~ general(lp);
   prior ltau  ~ egamma(0.001, scale=1000);
   tau = exp(ltau);
   endnodata;

   mean = intercept +
          pw300cc           * '< 300cc'n          +
          pw300_550cc       * '300-550cc'n        +
          pw5000            * '< $5000'n          +
          pw5000_6000       * '$5000 - $6000'n    +
          pw6001_7000       * '$6001 - $7000'n    +
          pwElectricStart   * 'Electric Start'n   +
          pwCounterBalanced * 'Counter Balanced'n +
          pwJapanese        * Japanese;
   model rank ~ normal(mean, prec=tau);
run;

data _null_;
   rc = logmpdffree();
run;
```

The two ARRAY statements allocate a 4×4 dimensional array for the prior covariance and an array of size 4 for the prior means. In the BEGINCNST and ENDCNST statements, the CALL IDENTITY function sets sigma to be an identity matrix; the CALL MULT function sets sigma's diagonal elements to be 100 (the diagonal variance terms); the CALL ZEROMATRIX function sets mu to be a vector of zeros (the prior means); and the LOGMPDFSETSQ function sets up sigma to be called in a multivariate normal density function later. For matrix functions in PROC MCMC, see the section "Matrix Functions in PROC MCMC" on page 3551. For multivariate density functions, see the section "Multivariate Density Functions" on page 3546. It is important to note that if you used the LOGMPDFSET or the LOGMPDFSETSQ functions to set up covariance matrix, you must free the memory allocated by these functions after you exit PROC MCMC. To free the memory, use the function LOGMPDFFREE.

There are two PARMS statements, with each of them naming a block of parameters. The first PARMS statement blocks the following: the intercept, the two size parameters, the one start-type parameter, the one origin parameter, and the log of the precision. The second PARMS statement blocks the three price parameters and the one balance parameter, parameters that have the constraint multivariate normal prior. The second PARMS statement also specifies initial values for the parameter estimates. The initial values reflect the constraints on these parameters. The initial part-worth utilities all decrease from 0.3 to 0.2 to 0.1 to 0.0 (for the implicit reference level) as the prices increase. Also, the initial part-worth utility for the counter-balanced engine is set to a positive value, 1.

In the PRIOR statements, regression coefficients without constraints are given an independent normal prior with mean at 0 and variance of 100. The next IF-ELSE construction imposes the constraints. When these constraints are met, pw5000, pw5000_6000, pw6001_7000, pwCounterBalanced are jointly distributed as a multivariate normal prior with mean mu and covariance sigma (as defined via the symbol 'v' in the BEGINCNST and ENDCNST statements). Otherwise, the prior is not defined and lp is assigned a missing value.

The parameter ltau is given an egamma prior. It is an equivalent prior to placing a gamma prior, with the same configuration, on tau = exp(ltau). For the definition of the egamma distribution, see the section "Standard Distributions" on page 3530. This transformation often improves mixing (see "Example 52.4: Nonlinear Poisson Regression Models" on page 3605 and "Example 52.12: Using a Transformation to Improve Mixing" on page 3683). The next assignment statement transforms ltau back to tau.

The model specification is linear. The mean is comprised of an intercept and the sum of terms like pw300cc * '< 300cc'n, which is a parameter times an input data set variable. The MODEL statement specifies that the linear model for rank is normally distributed with mean mean and precision tau.

After the PROC MCMC run, you *must* run the memory clean up function LOGMPDFFREE, which should produce the following note in the log file:

```
NOTE: The matrix -  v - has been deleted.
```

The MCMC results are shown in Output 52.10.3.

Output 52.10.3 MCMC Results

Constrained Conjoint Analysis

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
intercept	5000	2.2052	2.6285	0.8089	2.3658	3.8732
pw300cc	5000	0.0780	2.5670	−1.4062	0.0717	1.5850
pw300_550cc	5000	−0.0173	2.5378	−1.5136	−0.00275	1.4536
pwElectricStart	5000	−1.2175	2.1805	−2.4933	−1.1041	0.1410
pwJapanese	5000	−0.4212	2.1485	−1.6575	−0.4102	0.7909
ltau	5000	−2.4440	0.7293	−2.9024	−2.3787	−1.9177
pw5000	5000	4.3724	2.4962	2.6418	3.9163	5.5202
pw5000_6000	5000	2.6649	1.8227	1.3878	2.2894	3.5162
pw6001_7000	5000	1.4880	1.3303	0.5077	1.1389	2.0849
pwCounterBalanced	5000	5.9056	2.0591	4.6440	5.9033	7.1036

The estimates of the part-worth utility for the price categories are ordered as expected. This agrees with the intuition that there is a higher preference for a less expensive motor bike when all other things are equal, and that is what you see when you look at the estimated posterior means for the price part-worths. The estimated standard deviations of the price part-worths in this model are of approximately the same order of magnitude as the posterior means. This indicates that the part-worth utilities for this subject are not significantly far from each other, and that this subject's ranking of the options was not significantly influenced by the difference in price.

One advantage of Bayesian analysis is that you can incorporate prior information in the data analysis. Constraints on the parameter space are one possible source of information that you might have before you examine the data. This example shows that it can easily be accomplished in PROC MCMC.

Example 52.11: Implement a New Sampling Algorithm

This example illustrates using the UDS statement to implement a new Markov chain sampler. The algorithm demonstrated here is proposed by Holmes and Held (2006), hereafter referred to as HH. They presented a Gibbs sampling algorithm for generating draws from the posterior distribution of the parameters in a probit regression model. The notation follows closely to HH.

The data used here is the remission data set from a PROC LOGISTIC example:

```
title 'Implement a New Sampling Algorithm';
data inputdata;
   input remiss cell smear infil li blast temp;
   ind = _n_;
   cnst = 1;
```

```
   label remiss='Complete Remission';
   datalines;

... more lines ...

   0  1     0.73  0.73  0.7  0.398  0.986
;
```

The variable remiss is the cancer remission indicator variable with a value of 1 for remission and a value of 0 for nonremission. There are six explanatory variables: cell, smear, infil, li, blast, and temp. These variables are the risk factors thought to be related to cancer remission. The binary regression model is as follows:

$$\text{remiss}_i \sim \text{binary}(p_i)$$

where the covariates are linked to p_i through a probit transformation:

$$\text{probit}(p_i) = \mathbf{x}'\boldsymbol{\beta}$$

$\boldsymbol{\beta}$ are the regression coefficients and $\mathbf{x}'$ the explanatory variables. Suppose that you want to use independent normal priors on the regression coefficients:

$$\beta_i \sim \text{normal}(0, \text{var} = 25)$$

Fitting a logistic model with PROC MCMC is straightforward. You can use the following statements:

```
proc mcmc data=inputdata nmc=100000 propcov=quanew seed=17
          outpost=mcmcout;
   ods select PostSummaries ess;
   parms beta0-beta6;
   prior beta: ~ normal(0,var=25);
   mu = beta0 + beta1*cell + beta2*smear +
        beta3*infil +  beta4*li + beta5*blast +  beta6*temp;
   p = cdf('normal', mu, 0, 1);
   model remiss ~ bern(p);
run;
```

The expression mu is the regression mean, and the CDF function links mu to the probability of remission p in the binary likelihood.

The summary statistics and effective sample sizes tables are shown in Output 52.11.1. There are high autocorrelations among the posterior samples, and efficiency is relatively low. The correlation time is reduced only after a large amount of thinning.

Output 52.11.1 Random Walk Metropolis

Implement a New Sampling Algorithm

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
beta0	100000	-2.0531	3.8299	-4.6418	-2.0354	0.5638
beta1	100000	2.6300	2.8270	0.6563	2.5272	4.4846
beta2	100000	-0.8426	3.2108	-3.0270	-0.8263	1.3429
beta3	100000	1.5933	3.5491	-0.7993	1.6190	3.9695
beta4	100000	2.0390	0.8796	1.4312	2.0028	2.6194
beta5	100000	-0.3184	0.9543	-0.9613	-0.3123	0.3418
beta6	100000	-3.2611	3.7806	-5.8050	-3.2736	-0.7243

Implement a New Sampling Algorithm

The MCMC Procedure

Effective Sample Sizes

Parameter	ESS	Correlation Time	Efficiency
beta0	4280.8	23.3602	0.0428
beta1	4496.5	22.2398	0.0450
beta2	3434.1	29.1199	0.0343
beta3	3856.6	25.9294	0.0386
beta4	3659.7	27.3245	0.0366
beta5	3229.9	30.9610	0.0323
beta6	4430.7	22.5696	0.0443

As an alternative to the random walk Metropolis, you can use the Gibbs algorithm to sample from the posterior distribution. The Gibbs algorithm is described in the section "Gibbs Sampler" on page 154. While the Gibbs algorithm generally applies to a wide range of statistical models, the actual implementation can be problem-specific. In this example, performing a Gibbs sampler involves introducing a class of auxiliary variables (also known as latent variables). You first reformulate the model by adding a z_i for each observation in the data set:

$$
\begin{aligned}
y_i &= \begin{cases} 1 & \text{if } z_i > 0 \\ 0 & \text{otherwise} \end{cases} \\
z_i &= \mathbf{x}_i'\boldsymbol{\beta} + \epsilon_i \\
\epsilon &\sim \text{normal}(0, 1) \\
\boldsymbol{\beta} &\sim \pi(\boldsymbol{\beta})
\end{aligned}
$$

If $\boldsymbol{\beta}$ has a normal prior, such as $\pi(\boldsymbol{\beta}) = N(\mathbf{b}, \mathbf{v})$, you can work out a closed form solution to the full conditional distribution of $\boldsymbol{\beta}$ given the data and the latent variables z_i. The full conditional

distribution is also a multivariate normal, due to the conjugacy of the problem. See the section "Conjugate Priors" on page 146. The formula is shown here:

$$\begin{aligned} \boldsymbol{\beta}|\mathbf{z},\mathbf{x} &\sim \text{normal}(\mathbf{B},\mathbf{V}) \\ \mathbf{B} &= \mathbf{V}((v)^{-1}\mathbf{b}+\mathbf{x}'\mathbf{z}) \\ \mathbf{V} &= (\mathbf{v}^{-1}+\mathbf{x}'\mathbf{x})^{-1} \end{aligned}$$

The advantage of creating the latent variables is that the full conditional distribution of $\mathbf{z}$ is also easy to work with. The distribution is a truncated normal distribution:

$$z_i|\boldsymbol{\beta},\boldsymbol{x}_i,y_i \sim \begin{cases} \text{normal}(\mathbf{x}_i\boldsymbol{\beta},1)I(z_i>0) & \text{if } y_i=1 \\ \text{normal}(\mathbf{x}_i\boldsymbol{\beta},1)I(z_i\le 0) & \text{otherwise} \end{cases}$$

You can sample $\boldsymbol{\beta}$ and z iteratively, by drawing $\boldsymbol{\beta}$ given z and vice verse. HH point out that a high degree of correlation could exist between $\boldsymbol{\beta}$ and z, and it makes this iterative way of sampling inefficient. As an improvement, HH proposed an algorithm that samples $\boldsymbol{\beta}$ and $\mathbf{z}$ jointly. At each iteration, you sample z_i from the posterior marginal distribution (this is the distribution that is conditional only on the data and not on any parameters) and then sample $\boldsymbol{\beta}$ from the same posterior full conditional distribution as described previously:

1. Sample z_i from its posterior marginal distribution:

$$\begin{aligned} z_i|\mathbf{z}_{-i},y_i &\sim \begin{cases} \text{normal}(m_i,v_i)I(z_i>0) & \text{if } y_i=1 \\ \text{normal}(m_i,v_i)I(z_i\le 0) & \text{otherwise} \end{cases} \\ m_i &= \mathbf{x}_i\mathbf{B}-w_i(z_i-\mathbf{x}_i\mathbf{B}) \\ v_i &= 1+w_i \\ w_i &= h_i/(1-h_i) \\ h_i &= (\mathbf{H})_i i, \mathbf{H}=\mathbf{x}\mathbf{V}\mathbf{x}' \end{aligned}$$

2. Sample $\boldsymbol{\beta}$ from the same posterior full conditional distribution described previously.

For a detailed description of each of the conditional terms, refer to the original paper.

PROC MCMC cannot sample from the probit model by using this sampling scheme but you can implement the algorithm by using the UDS statement. To sample z_i from its marginal, you need a function that draws random variables from a truncated normal distribution. The functions, RLTNORM and RRTNORM, generate left- and right-truncated normal variates, respectively. The algorithm is taken from Robert (1995).

The functions are written in PROC FCMP (see the FCMP Procedure in the *Base SAS Procedures Guide*):

```
proc fcmp outlib=sasuser.funcs.uds;
   /*****************************************/
   /* Generate left-truncated normal variate */
   /*****************************************/
   function rltnorm(mu,sig,lwr);
```

```
      if lwr<mu then do;
         ans = lwr-1;
         do while(ans<lwr);
            ans = rand('normal',mu,sig);
         end;
      end;
      else do;
         mul = (lwr-mu)/sig;
         alpha = (mul + sqrt(mul**2 + 4))/2;
         accept=0;
         do while(accept=0);
            z = mul + rand('exponential')/alpha;
            lrho = -(z-alpha)**2/2;
            u = rand('uniform');
            lu = log(u);
            if lu <= lrho then accept=1;
         end;
         ans = sig*z + mu;
      end;
      return(ans);
      endsub;

      /*****************************************/
      /* Generate right-truncated normal variate */
      /*****************************************/
      function rrtnorm(mu,sig,uppr);
      ans = 2*mu - rltnorm(mu,sig, 2*mu-uppr);
      return(ans);
      endsub;
   run;
```

The function call to RLTNORM(mu,sig,lwr) generates a random number from the left-truncated normal distribution:

$$\theta \sim \text{normal}(\text{mu}, \text{sd} = \text{sig}) I(\theta > \text{lwr})$$

Similarly, the function call to RRTNORM(mu,sig,uppr) generates a random number from the right-truncated normal distribution:

$$\theta \sim \text{normal}(\text{mu}, \text{sd} = \text{sig}) I(\theta < \text{lwr})$$

These functions are used to generate the latent variables z_i.

Using the algorithm A1 from the HH paper as an example, Output 52.37 lists a line-by-line implementation with the PROC MCMC coding style. The table is broken into three portions: set up the constants, initialize the parameters, and sample one draw from the posterior distribution. The left column of the table is identical to the A1 algorithm stated in the appendix of HH. The right column of the table lists SAS statements.

Table 52.37 Holmes and Held (2006), algorithm A1. Side-by-Side Comparison to SAS

Define Constants	In the BEGINCNST/ENDCNST Statements

$V \leftarrow (X^T X + v^{-1})^{-1}$	`call transpose(x,xt); /* xt = transpose(x) */`
	`call mult(xt,x,xtx);`
	`call inv(v,v); /* v = inverse(v) */`
	`call addmatrix(xtx,v,xtx); /* xtx = xtx+v */`
	`call inv(xtx,v); /* v = inverse(xtx) */`

$L \leftarrow \text{Chol}(V)$	`call chol(v,L);`
$S \leftarrow VX^T$	`call mult(v,xt,S);`

FOR $j = 1$ to n
$H[j] \leftarrow X[j,]S[,j]$
$W[j] \leftarrow H[j]/(1 - H[j])$
$Q[j] \leftarrow W[j] + 1$
END
	`call mult(x,S,HatMat);`
	`do j=1 to &n;`
	`H = HatMat[j,j];`
	`W[j] = H/(1-H);`
	`sQ[j] = sqrt(W[j] + 1); /* use s.d. in SAS */`
	`end;`

Initial Values	In the BEGINCNST/ENDCNST Statements

$Z \sim \text{normal}(0, I_n) Ind(Y, Z)$	`do j=1 to &n;`
	`if(y[j]=1) then`
	`Z[j] = rltnorm(0,1,0);`
	`else`
	`Z[j] = rrtnorm(0,1,0);`
	`end;`

$B \leftarrow SZ$	`call mult(S,Z,B);`

Table 52.37 (continued)

Draw One Sample	Subroutine HH

FOR $j = 1$ to n
$z_{old} \leftarrow Z[j]$
$m \leftarrow X[j,]B$
$m \leftarrow m - W[j](Z[j] - m)$
$Z[j] \sim \text{normal}(m, Q[j])Ind(Y[j], Z[j])$
$B \leftarrow B + (Z[j] - z_{old})S[, j]$
END
$T \sim \text{normal}(0, I_p)$
$\beta[, i] \leftarrow B + LT$
	`do j=1 to &n;`
	`  zold = Z[j];`
	`  m = 0;`
	`  do k= 1 to &p;`
	`  m = m + X[j,k] * B[k];`
	`  end;`
	`  m = m - W[j]*(Z[j]-m);`
	`  if (y[j]=1) then`
	`  Z[j] = rltnorm(m,sQ[j],0);`
	`  else`
	`  Z[j] = rrtnorm(m,sQ[j],0);`
	`  diff = Z[j] - zold;`
	`  do k= 1 to &p;`
	`  B[k] = B[k] + diff * S[k,j];`
	`  end;`
	`  end;`
	`do j = 1 to &p;`
	`  T[j] = rand('normal');`
	`  end;`
	`call mult(L,T,T);`
	`call addmatrix(B,T,beta);`

The following statements define the subroutine HH (algorithm A1) in PROC FCMP and store it in library **sasuser.funcs.uds**:

```
/* define the HH algorithm in PROC FCMP. */
%let n = 27;
%let p = 7;
options cmplib=sasuser.funcs;
proc fcmp outlib=sasuser.funcs.uds;
   subroutine HH(beta[*],Z[*],B[*],x[*,*],y[*],W[*],sQ[*],S[*,*],L[*,*]);
   outargs beta, Z, B;
   array T[&p] / nosym;
   do j=1 to &n;
      zold = Z[j];
      m = 0;
      do k = 1 to &p;
         m = m + X[j,k] * B[k];
      end;
      m = m - W[j]*(Z[j]-m);
      if (y[j]=1) then
         Z[j] = rltnorm(m,sQ[j],0);
      else
         Z[j] = rrtnorm(m,sQ[j],0);
      diff = Z[j] - zold;
      do k = 1 to &p;
```

```
            B[k] = B[k] + diff * S[k,j];
         end;
      end;
      do j=1 to &p;
         T[j] = rand('normal');
      end;
      call mult(L,T,T);
      call addmatrix(B,T,beta);
      endsub;
   run;
```

Note that one-dimensional array arguments take the form of *name*[*] and two-dimensional array arguments take the form of *name*[*,*]. Three variables, beta, Z, and B, are OUTARGS variables, making them the only arguments that can be modified in the subroutine. For the UDS statement to work, all OUTARGS variables have to be model parameters. Technically, only beta and Z are model parameters, and B is not. The reason that B is declared as an OUTARGS is because the array must be updated throughout the simulation, and this is the only way to modify its values. The input array x contains all of the explanatory variables, and the array y stores the response. The rest of the input arrays, W, sQ, S, and L, store constants as detailed in the algorithm. The following statements illustrate how to fit a Bayesian probit model by using the HH algorithm:

```
options cmplib=sasuser.funcs;

proc mcmc data=inputdata nmc=5000 monitor=(beta) outpost=hhout;
   ods select PostSummaries ess;
   array xtx[&p,&p];      /* work space                          */
   array xt[&p,&n];       /* work space                          */
   array v[&p,&p];        /* work space                          */
   array HatMat[&n,&n];   /* work space                          */
   array S[&p,&n];        /* V * Xt                              */
   array W[&n];
   array y[1]/ nosymbols; /* y stores the response variable      */
   array x[1]/ nosymbols; /* x stores the explanatory variables */
   array sQ[&n];          /* sqrt of the diagonal elements of Q */
   array B[&p];           /* conditional mean of beta            */
   array L[&p,&p];        /* Cholesky decomp of conditional cov */
   array Z[&n];           /* latent variables Z                  */
   array beta[&p] beta0-beta6;   /* regression coefficients      */

   begincnst;
      call streaminit(83101);
      if ind=1 then do;
         rc = read_array("inputdata", x, "cnst", "cell", "smear", "infil",
                         "li", "blast", "temp");
         rc = read_array("inputdata", y, "remiss");
         call identity(v);
         call mult(v, 25, v);
         call transpose(x,xt);
         call mult(xt,x,xtx);
         call inv(v,v);
         call addmatrix(xtx,v,xtx);
         call inv(xtx,v);
         call chol(v,L);
         call mult(v,xt,S);
```

```
            call mult(x,S,HatMat);
            do j=1 to &n;
               H = HatMat[j,j];
               W[j] = H/(1-H);
               sQ[j] = sqrt(W[j] + 1);
            end;

            do j=1 to &n;
               if(y[j]=1) then
                   Z[j] = rltnorm(0,1,0);
               else
                   Z[j] = rrtnorm(0,1,0);
            end;
            call mult(S,Z,B);
         end;
      endcnst;

      uds HH(beta,Z,B,x,y,W,sQ,S,L);
      parms z: beta: 0 B1-B7 / uds;
      prior z: beta: B1-B7 ~ general(0);

      model general(0);
   run;
```

The OPTIONS statement names the catalog of FCMP subroutines to use. The cmplib library stores the subroutine HH. You do not need to set a random number seed in the PROC MCMC statement because all random numbers are generated from the HH subroutine. The initialization of the rand function is controlled by the streaminit function, which is called in the program with a seed value of 83101.

A number of arrays are allocated. Some of them, such as xtx, xt, v, and HatMat, allocate work space that is used to construct constant arrays. Other arrays are used in the subroutine sampling. Explanations of the arrays are shown in comments in the statements.

In the BEGINCNST and ENDCNST statement block, you read data set variables in the arrays x and y, calculate all the constant terms, and assign initial values to Z and B. For the READ_ARRAY function, see the section "READ_ARRAY Function" on page 3510. For listings of all array functions and their definitions, see the section "Matrix Functions in PROC MCMC" on page 3551.

The UDS statement declares that the subroutine HH is used to sample the parameters beta, Z, and B. You also specify the UDS option in the PARMS statement. Because all parameters are updated through the UDS interface, it is not necessary to declare the actual form of the prior for any of the parameters. Each parameter is declared to have a prior of general(0). Similarly, it is not necessary to declare the actual form of the likelihood. The MODEL statement also takes a flat likelihood of the form `general(0)`.

Summary statistics and effective sample sizes are shown in Output 52.11.2. The posterior estimates are very close to what was shown in Output 52.11.1. The HH algorithm produces samples that are much less correlated.

Output 52.11.2 Holms and Held

Implement a New Sampling Algorithm

The MCMC Procedure

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
beta0	5000	-2.0567	3.8260	-4.6537	-2.0777	0.5495
beta1	5000	2.7254	2.8079	0.7812	2.6678	4.5370
beta2	5000	-0.8318	3.2017	-2.9987	-0.8626	1.2918
beta3	5000	1.6319	3.5108	-0.7481	1.6636	4.0302
beta4	5000	2.0567	0.8800	1.4400	2.0266	2.6229
beta5	5000	-0.3473	0.9490	-0.9737	-0.3267	0.2752
beta6	5000	-3.3787	3.7991	-5.9089	-3.3504	-0.7928

Implement a New Sampling Algorithm

The MCMC Procedure

Effective Sample Sizes

Parameter	ESS	Correlation Time	Efficiency
beta0	3651.3	1.3694	0.7303
beta1	1563.8	3.1973	0.3128
beta2	5005.9	0.9988	1.0012
beta3	4853.2	1.0302	0.9706
beta4	2611.2	1.9148	0.5222
beta5	3049.2	1.6398	0.6098
beta6	3503.2	1.4273	0.7006

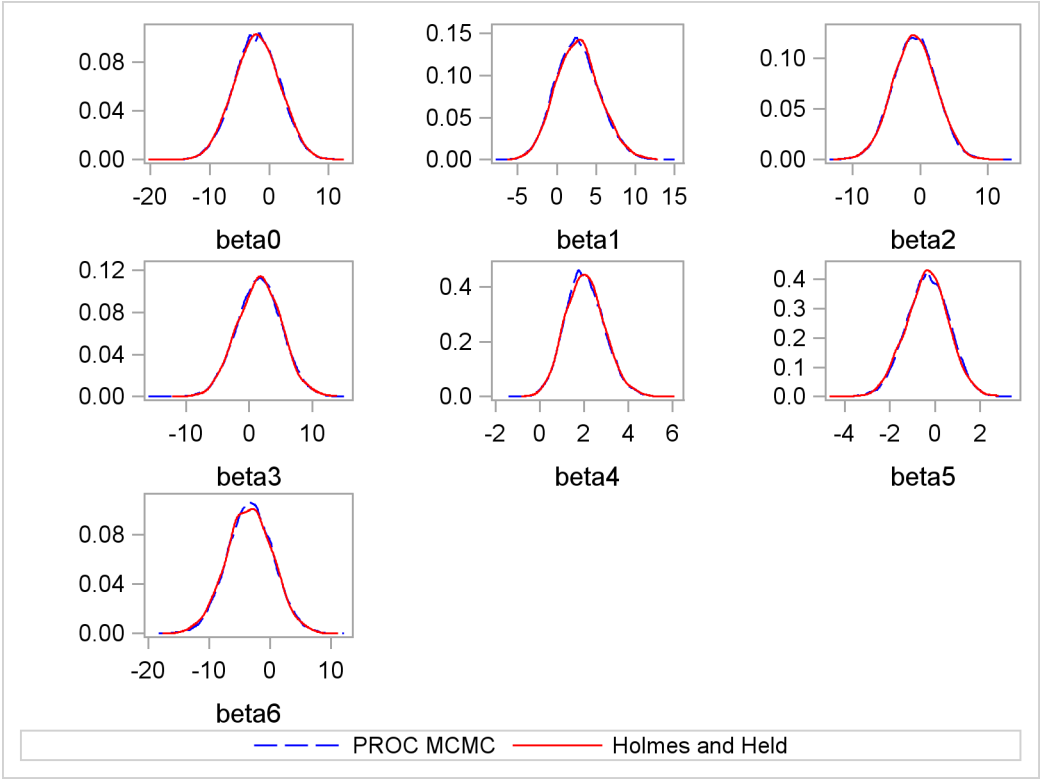
The following statements generate the kernel density comparison plots shown in Output 52.11.3:

```
proc kde data=mcmcout;
   ods exclude all;
   univar beta0 beta1 beta2 beta3 beta4 beta5 beta6 / out=m1(drop=count);
run;
ods exclude none;

%reshape(m1, mcmc, suffix1=, suffix2=md);

proc kde data=hhout(drop = LogPost logprior loglike Iteration z1-z27 b1-b7);;
   ods exclude all;
   univar beta0 beta1 beta2 beta3 beta4 beta5 beta6
          / out=m2 (drop=count);
run;
ods exclude none;

%reshape(m2, hh, suffix1=p, suffix2=pd);

data all;
```

```
   merge mcmc hh;
run;

proc template;
   define statgraph threebythree;
      %macro plot;
         begingraph;
            layout lattice / rows=3 columns=3;
               %do i = 0 %to 6;
                  layout overlay /yaxisopts=(label=" ");
                     seriesplot y=beta&i.md x=beta&i
                        / connectorder=xaxis
                          lineattrs=(pattern=mediumdash color=blue)
                        legendlabel = "PROC MCMC" name="MCMC";
                     seriesplot y=beta&i.pd x=beta&i.p
                        / connectorder=xaxis lineattrs=(color=red)
                        legendlabel = "Holmes and Held" name="HH";
                  endlayout;
               %end;
               Sidebar / align = bottom;
                  discretelegend "MCMC" "HH";
               endsidebar;
            endlayout;
         endgraph;
      %mend; %plot;
   end;
run;

proc sgrender data=all template=threebythree;
   title "Kernel Density Comparison";
run;
```

The macro %RESHAPE is defined in the example "Logistic Regression Random-Effects Model" on page 3614.

Output 52.11.3 Kernel Density Comparison



It is interesting to compare the two approaches of fitting a generalized linear model. The random walk Metropolis on a seven-dimensional parameter space produces autocorrelations that are substantially higher than the HH algorithm. A much longer chain is needed to produce roughly equivalent effective sample sizes. On the other hand, the Metropolis algorithm is faster to run. The running time of these two examples is roughly the same, with the random walk Metropolis with 100000 samples, a 20-fold increase over that in the HH algorithm example. The speed difference can be attributed to a number of factors, ranging from the implementation of the software and the overhead cost of calling PROC FCMP subroutine and functions. In addition, the HH algorithm requires more parameters by creating an equal number of latent variables as the sample size. Sampling more parameters takes time. A larger number of parameters also increases the challenge in convergence diagnostics, because it is imperative to have convergence in all parameters before you make valid posterior inferences. Finally, you might feel that coding in PROC MCMC is easier. However, this really is not a fair comparison to make here. Writing a Metropolis algorithm from scratch would have probably taken just as much, if not more, effort than the HH algorithm.

Example 52.12: Using a Transformation to Improve Mixing

Proper transformations of parameters can often improve the mixing in PROC MCMC. You already saw this in "Example 52.4: Nonlinear Poisson Regression Models" on page 3605, which sampled using the log scale of parameters that priors that are strictly positive, such as the gamma priors.

This example shows another useful transformation: the logit transformation on parameters that take a uniform prior on [0, 1].

The data set is taken from Sharples (1990). It is used in Chaloner and Brant (1988) and Chaloner (1994) to identify outliers in the data set in a two-level hierarchical model. Congdon (2003) also uses this data set to demonstrates the same technique. This example uses the data set to illustrate how mixing can be improved using transformation and does not address the question of outlier detection as in those papers. The following statements create the data set:

```
title 'Using Transformation to Improve Mixing';
data inputdata;
   input nobs grp y @@;
   ind = _n_;
   datalines;
1 1 24.80 2 1 26.90 3 1 26.65
4 1 30.93 5 1 33.77 6 1 63.31
1 2 23.96 2 2 28.92 3 2 28.19
4 2 26.16 5 2 21.34 6 2 29.46
1 3 18.30 2 3 23.67 3 3 14.47
4 3 24.45 5 3 24.89 6 3 28.95
1 4 51.42 2 4 27.97 3 4 24.76
4 4 26.67 5 4 17.58 6 4 24.29
1 5 34.12 2 5 46.87 3 5 58.59
4 5 38.11 5 5 47.59 6 5 44.67
;
```

There are five groups (grp, $j = 1, \cdots, 5$) with six observations (nobs, $i = 1, \cdots, 6$) in each. The two-level hierarchical model is specified as follows:

$$\begin{aligned} y_{ij} &\sim \text{normal}(\theta_j, \text{prec} = \tau_w) \\ \theta_j &\sim \text{normal}(\mu, \text{prec} = \tau_b) \\ \mu &\sim \text{normal}(0, \text{prec} = 1e-6) \\ \tau &\sim \text{gamma}(0.001, \text{iscale} = 0.001) \\ p &\sim \text{uniform}(0, 1) \end{aligned}$$

with the precision parameters related to each other in the following way:

$$\begin{aligned} \tau_b &= \tau/p \\ \tau_w &= \tau_b - \tau \end{aligned}$$

The total number of parameters in this model is eight: $\theta_1, \cdots, \theta_5, \mu, \tau$, and p.

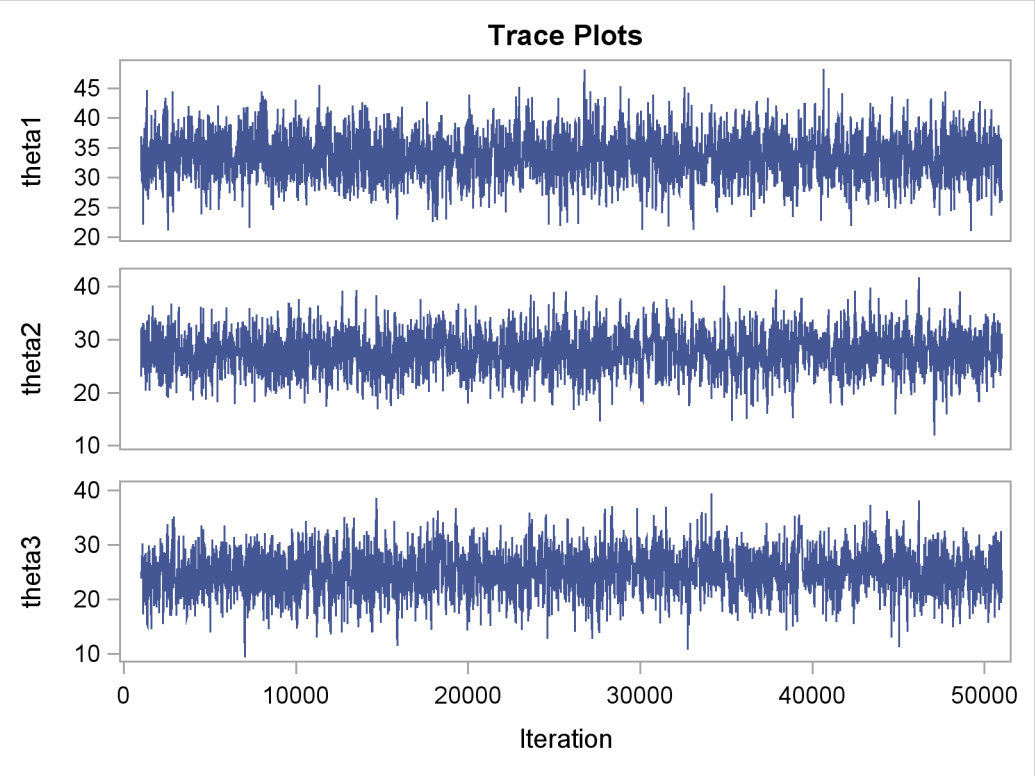
The following statements fit the model:

```
ods graphics on;
proc mcmc data=inputdata nmc=50000 thin=10 outpost=m1 seed=17
          plot=trace;
   ods select ess tracepanel;
   array theta[5];

   parms theta:;
```

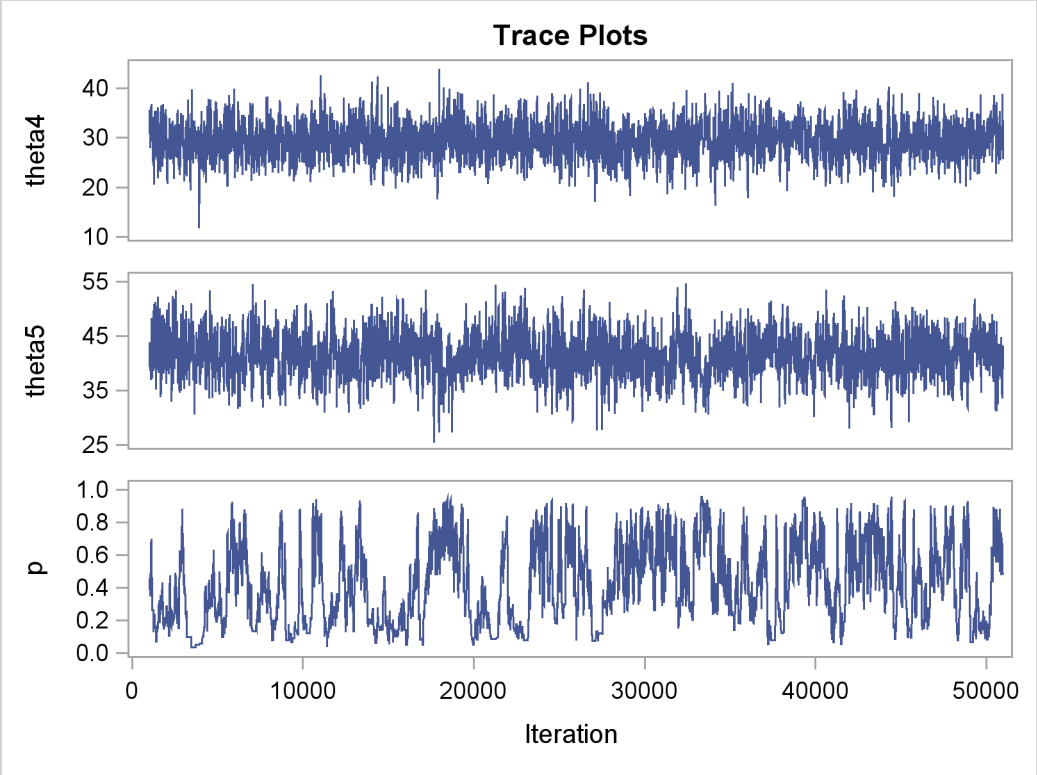
```
   parms p tau;
   parms mu ;

   beginnodata;
   hyper p ~ uniform(0,1);
   hyper tau ~ gamma(shape=0.001,iscale=0.001);
   hyper mu ~ normal(0,prec=0.00000001);
   taub = tau/p;
   prior theta: ~ normal(mu,prec=taub);
   tauw = taub-tau;
   endnodata;

   model y ~ normal(theta[grp],prec=tauw);
run;
```

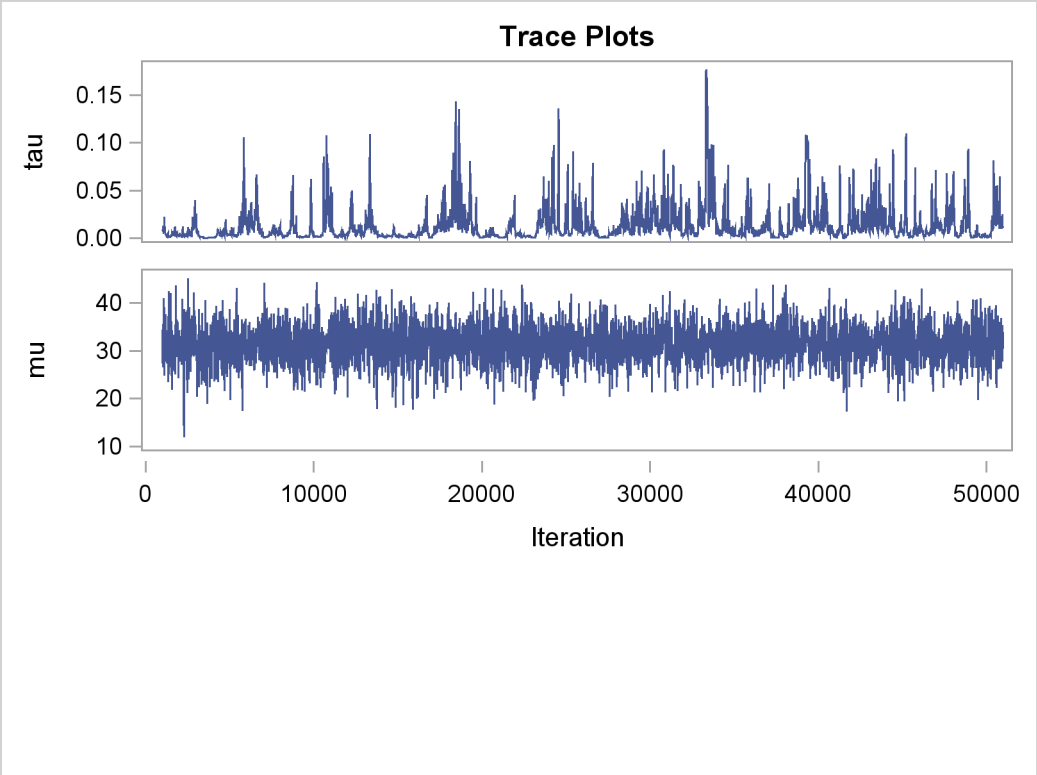
The ODS SELECT statement displays the effective sample size table and the trace plots. The ODS GRAPHICS ON statement requests ODS Graphics. The PROC MCMC statement specifies the usual options for the MCMC run and produces trace plots (PLOTS=TRACE). The ARRAY statement allocates an array of size 5 for theta. The three PARMS statements put parameters in three different blocks. The remaining statements specify the hyperprior, prior, and likelihood for the data, as described previously. The resulting trace plots are shown in Output 52.12.1, and the effective sample sizes table is shown in Output 52.12.2.

Output 52.12.1 Trace Plots



Output 52.12.1 *continued*





Output 52.12.2 Bad Effective Sample Sizes

Using Transformation to Improve Mixing

The MCMC Procedure

Effective Sample Sizes

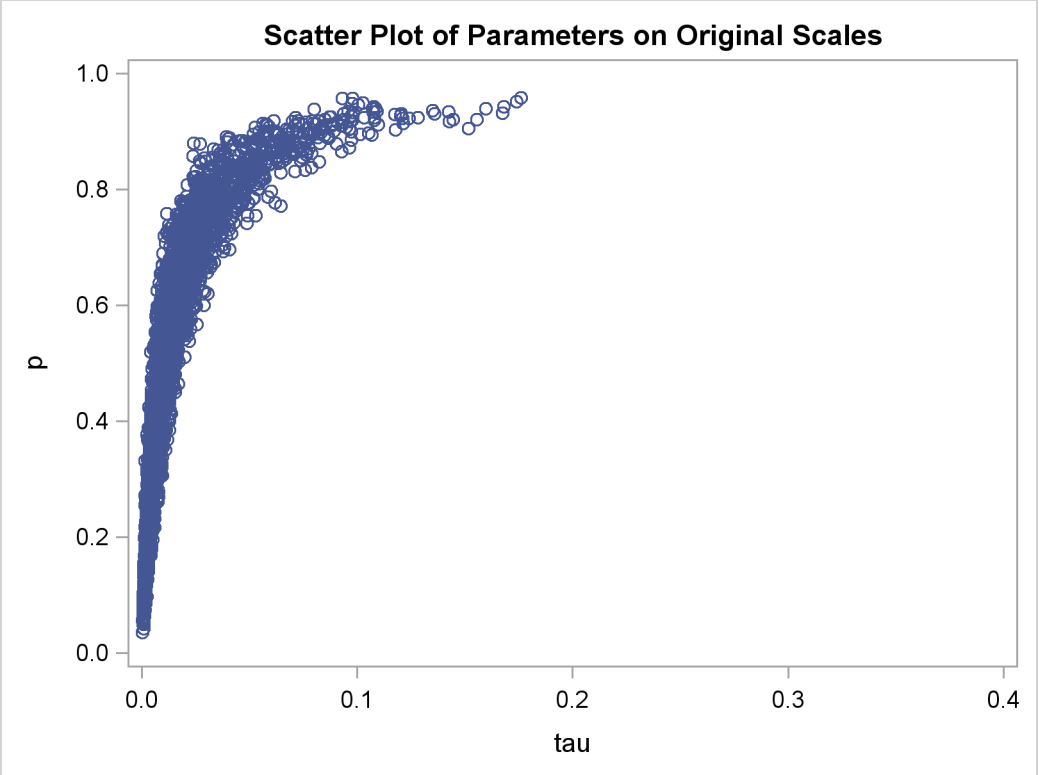
Parameter	ESS	Correlation Time	Efficiency
theta1	2207.5	2.2650	0.4415
theta2	1713.5	2.9180	0.3427
theta3	1458.5	3.4281	0.2917
theta4	1904.4	2.6255	0.3809
theta5	585.9	8.5345	0.1172
p	77.2	64.7758	0.0154
tau	140.8	35.5052	0.0282
mu	3340.3	1.4969	0.6681

The trace plots show that most parameters have relatively good mixing. Two exceptions appear to be p and τ. The trace plot of p shows a slow periodic movement. The τ parameter does not have good mixing either. When the values are close to zero, the chain stands there for long periods of time. An inspection of the effective sample sizes table reveals the same conclusion: p and τ have much smaller ESSs than the rest of the parameters.

A scatter plot of the posterior samples of p and τ reveals why mixing is bad in these two dimensions. The following statements generate the scatter plot in Output 52.12.3:

```
title 'Scatter Plot of Parameters on Original Scales';

proc sgplot data=m1;
   yaxis label = 'p';
   xaxis label = 'tau' values=(0 to 0.4 by 0.1);
   scatter x = tau y = p;
run;
```

Output 52.12.3 Scatter Plot of τ versus p



The two parameters clearly have a nonlinear relationship. It is not surprising that the Metropolis algorithm does not work well here. The algorithm is designed for cases where the parameters are linearly related with each other.

Instead of sampling on τ, you can sample on the log of τ. The formulation

$$\begin{aligned} \pi(\tau) &\propto f_{\Gamma}(0.001, \text{iscale} = 0.001) \\ \pi(\log(\tau)) &\propto f_{e\Gamma}(0.001, \text{iscale} = 0.001) \end{aligned}$$

where f_{Γ} and $f_{e\Gamma}$ are density functions for the gamma and egamma distributions. See the section "Standard Distributions" on page 3530 for the definitions of the distributions. In addition, you can sample on the logit of p. The formulation

$$\pi(p) \propto f_{\text{uniform}}(0, 1)$$

is equivalent to the following transformation:

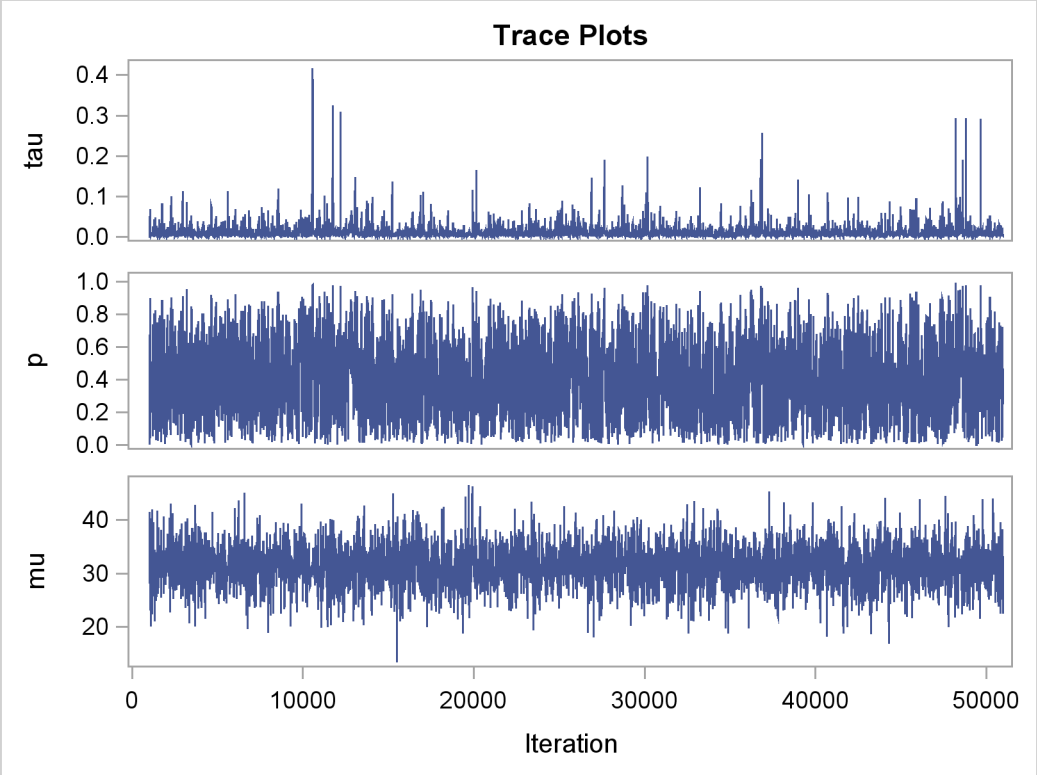
$$\begin{aligned} \text{lgp} &= \text{logit}(p) \\ \pi(\text{lgp}) &\propto \exp(-\text{lgp})(1 + \exp(-\text{lgp}))^{-2} \end{aligned}$$

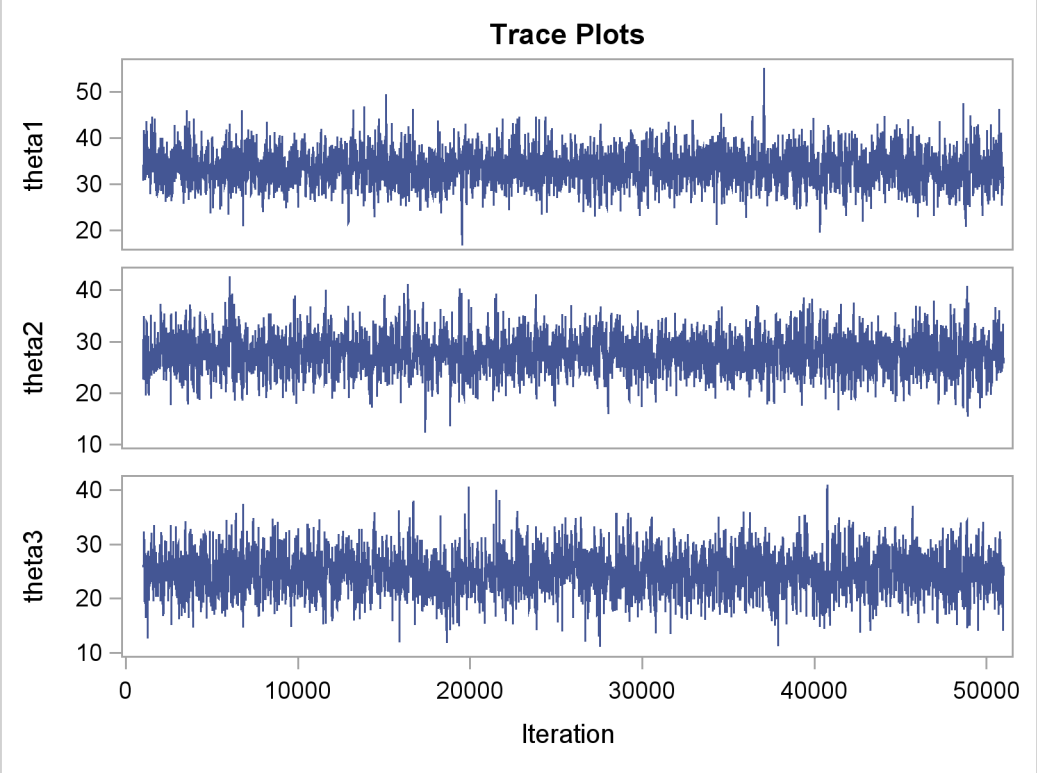
The following statements fit the same model by using transformed parameters:

```
proc mcmc data=inputdata nmc=50000 thin=10 outpost=m2 seed=17
          monitor=(tau p mu theta) plot=trace;
   ods select ess tracepanel;
   array theta[5];

   parms theta:;
   parms lgp 0 ltau ;
   parms mu ;

   beginnodata;
   prior ltau ~ egamma(shape=0.001,iscale=0.001);
   lp = -lgp - 2*log(1+exp(-lgp));
   prior lgp ~ general(lp);
   tau = exp(ltau);
   p = (1+exp(-lgp))**-1;
   prior mu ~ normal(0,prec=0.00000001);
   taub = tau/p;
   prior theta: ~ normal(mu,prec=taub);
   tauw = taub-tau;
   endnodata;

   model y ~ normal(theta[grp],prec=tauw);
run;
ods graphics off;
```

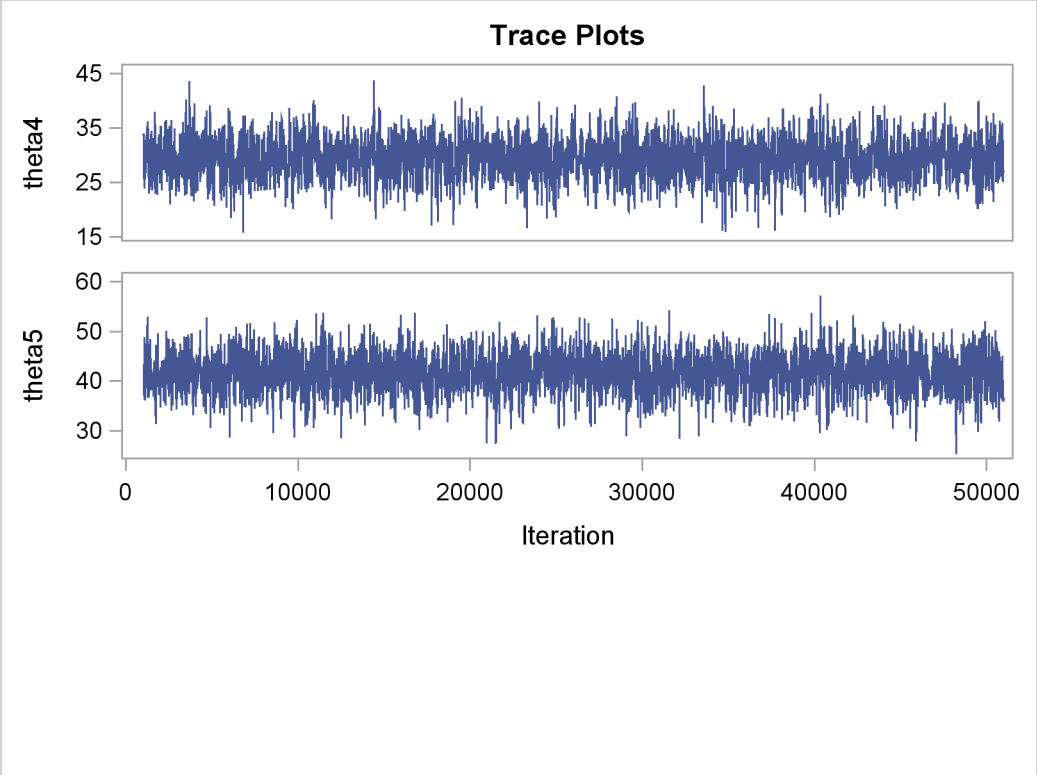
The variable lgp is the logit transformation of p, and ltau is the log transformation of τ. The prior for ltau is egamma. The lp assignment statement evaluates the log density of $\pi(\text{lgp})$. The tau and p assignment statements transform the parameters back to their original scales. The prior distributions for mu, theta, and the log likelihood in the MODEL statement remain unchanged. Trace plots (Output 52.12.4) and effective sample size (Output 52.12.5) both show significant improvements in the mixing for both p and τ.

Output 52.12.4 Trace Plots after Transformation





Output 52.12.4 *continued*



Output 52.12.5 Effective Sample Sizes after Transformation

Scatter Plot of Parameters on Original Scales

The MCMC Procedure

Effective Sample Sizes

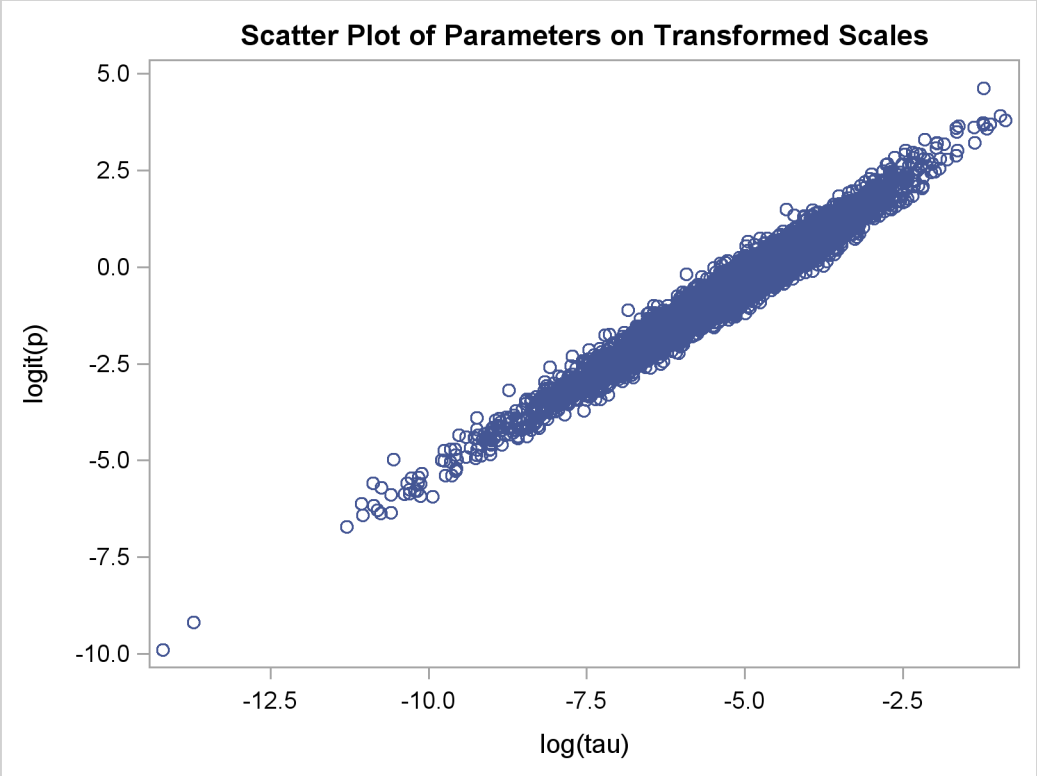
Parameter	ESS	Correlation Time	Efficiency
tau	1916.5	2.6089	0.3833
p	2468.7	2.0253	0.4937
mu	3273.9	1.5272	0.6548
theta1	2184.5	2.2888	0.4369
theta2	1938.1	2.5799	0.3876
theta3	1947.1	2.5679	0.3894
theta4	2115.8	2.3632	0.4232
theta5	2152.0	2.3235	0.4304

The following statements generate Output 52.12.6 and Output 52.12.7:

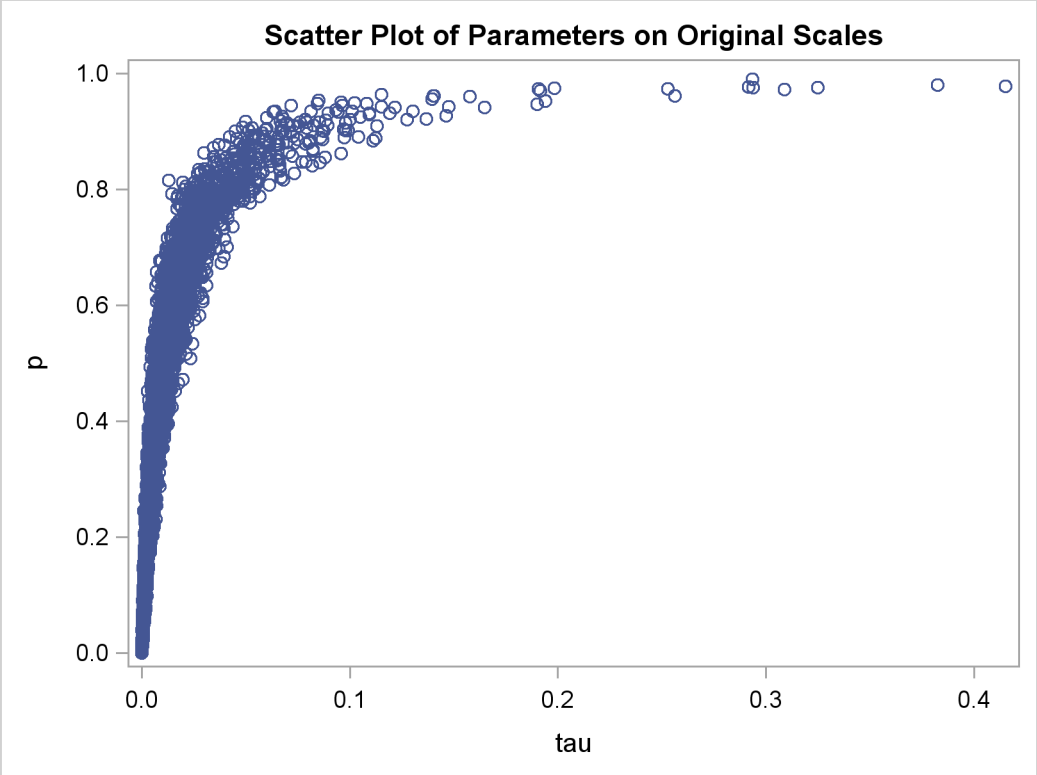
```
title 'Scatter Plot of Parameters on Transformed Scales';

proc sgplot data=m2;
   yaxis label = 'logit(p)';
```

```
   xaxis label = 'log(tau)';
   scatter x = ltau y = lgp;
run;

title 'Scatter Plot of Parameters on Original Scales';

proc sgplot data=m2;
   yaxis label = 'p';
   xaxis label = 'tau';
   scatter x = tau y = p;
run;
```

Output 52.12.6 Scatter Plot of $\log(\tau)$ versus logit(p), After Transformation



Output 52.12.7 Scatter Plot of τ versus p, After Transformation



The scatter plot of $\log(\tau)$ versus $\text{logit}(p)$ shows a linear relationship between the two transformed parameters, and this explains the improvement in mixing. In addition, the transformations also help the Markov chain better explore in the original parameter space. Output 52.12.7 shows a scatter plot of τ versus p. The plot is similar to Output 52.12.3. However, note that tau has a far longer tail in Output 52.12.7, extending all the way to 0.4 as opposed to 0.15 in Output 52.12.3. This means that the second Markov chain can explore this dimension of the parameter more efficiently, and as a result, you are able to draw more precise inference with an equal number of simulations.

Example 52.13: Gelman-Rubin Diagnostics

PROC MCMC does not have the Gelman-Rubin test (see the section "Gelman and Rubin Diagnostics" on page 161) as a part of its diagnostics. The Gelman-Rubin diagnostics rely on parallel chains to test whether they all converge to the same posterior distribution. This example demonstrates how you can carry out this convergence test. The regression model from the section "Simple Linear Regression" on page 3480 is used. The model has three parameters: β_0 and β_1 are the regression coefficients, and σ^2 is the variance of the error distribution.

The following statements generate the data set:

```
title 'Simple Linear Regression, Gelman-Rubin Diagnostics';
```

```
data Class;
   input Name $ Height Weight @@;
   datalines;
Alfred  69.0 112.5   Alice  56.5  84.0   Barbara 65.3  98.0
Carol   62.8 102.5   Henry  63.5 102.5   James   57.3  83.0
Jane    59.8  84.5   Janet  62.5 112.5   Jeffrey 62.5  84.0
John    59.0  99.5   Joyce  51.3  50.5   Judy    64.3  90.0
Louise  56.3  77.0   Mary   66.5 112.0   Philip  72.0 150.0
Robert  64.8 128.0   Ronald 67.0 133.0   Thomas  57.5  85.0
William 66.5 112.0
;
```

To run a Gelman-Rubin diagnostic test, you want to start Markov chains at different places in the parameter space. Suppose that you want to start β_0 at 10, -15, and 0; β_1 at -5, 10, and 0; and σ^2 at 1, 20, and 50. You can put these starting values in the following init SAS data set:

```
data init;
   input Chain beta0 beta1 sigma2;
   datalines;
   1   10  -5   1
   2  -15  10  20
   3    0   0  50
;
```

The following statements run PROC MCMC three times, each with starting values specified in the data set init:

```
/* define constants */
%let nchain = 3;
%let nparm = 3;
%let nsim = 50000;
%let var = beta0 beta1 sigma2;

%macro gmcmc;
   %do i=1 %to &nchain;
      data _null_;
         set init;
         if Chain=&i;
         %do j = 1 %to &nparm;
            call symputx("init&j", %scan(&var, &j));
         %end;
         stop;
      run;

      proc mcmc data=class outpost=out&i init=reinit nbi=0 nmc=&nsim
                stats=none seed=7;
         parms beta0 &init1 beta1 &init2;
         parms sigma2 &init3;
         prior beta0 beta1 ~ normal(0, var = 1e6);
         prior sigma2 ~ igamma(3/10, scale = 10/3);
         mu = beta0 + beta1*height;
         model weight ~ normal(mu, var = sigma2);
      run;
```

```
   %end;
%mend;

ods listing close;
%gmcmc;
ods listing;
```

The macro variables nchain, nparm, nsim, and var define the number of chains, the number of parameters, the number of Markov chain simulations, and the parameter names, respectively. The macro GMCMC gets initial values from the data set init, assigns them to the macro variables init1, init2 and init3, starts the Markov chain at these initial values, and stores the posterior draws to three output data sets: out1, out2, and out3.

In the PROC MCMC statement, the INIT=REINIT option restarts the Markov chain after tuning at the assigned initial values. No burn-in is requested.

You can use the autocall macro GELMAN to calculate the Gelman-Rubin statistics by using the three chains. The GELMAN macro has the following arguments:

```
%macro gelman(dset, nparm, var, nsim, nc=3, alpha=0.05);
```

The argument dset is the name of the data set that stores the posterior samples from all the runs, nparm is the number of parameters, var is the name of the parameters, nsim is the number of simulations, nc is the number of chains with a default value of 3, and alpha is the α significant level in the test with a default value of 0.05. This macro creates two data sets: _Gelman_Ests stores the diagnostic estimates and _Gelman_Parms stores the names of the parameters.

The following statements calculate the Gelman-Rubin diagnostics:

```
data all;
   set out1(in=in1) out2(in=in2) out3(in=in3);
   if in1 then Chain=1;
   if in2 then Chain=2;
   if in3 then Chain=3;
run;

%gelman(all, &nparm, &var, &nsim);

data GelmanRubin(label='Gelman-Rubin Diagnostics');
   merge _Gelman_Parms _Gelman_Ests;
run;

proc print data=GelmanRubin;
run;
```

The Gelman-Rubin statistics are shown in Output 52.13.1.

Output 52.13.1 Gelman-Rubin Diagnostics of the Regression Example

Simple Linear Regression, Gelman-Rubin Diagnostics

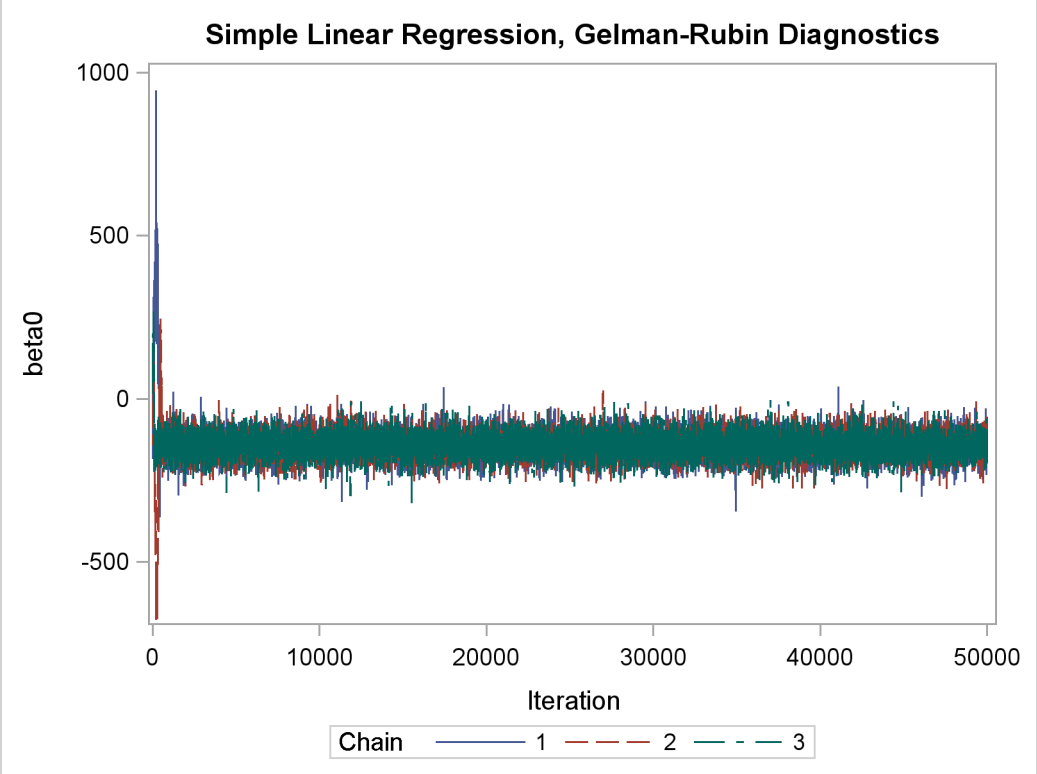
Obs	Parameter	Between-chain	Within-chain	Estimate	Upper Bound
1	beta0	5384.76	1168.64	1.0002	1.0001
2	beta1	1.20	0.30	1.0002	1.0002
3	sigma2	8034.41	2890.00	1.0010	1.0011

The Gelman-Rubin statistics do not reveal any concerns about the convergence or the mixing of the multiple chains. To get a better visual picture of the multiple chains, you can draw overlapping trace plots of these parameters from the three Markov chains runs.

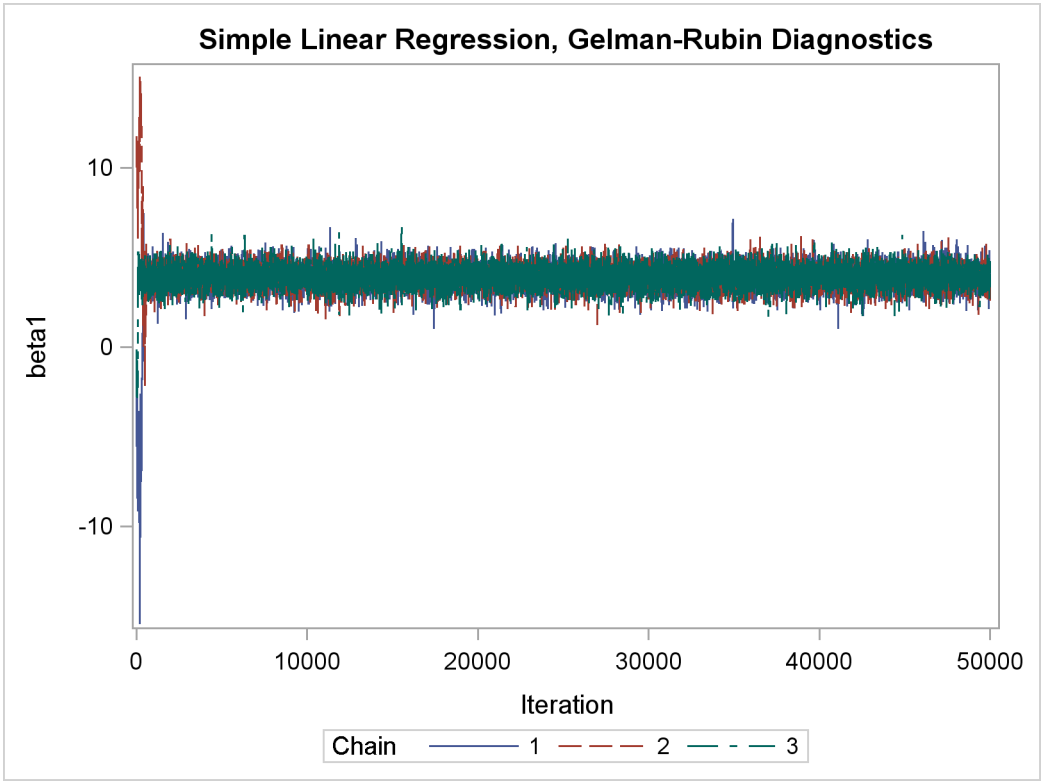
The following statements create Output 52.13.2:

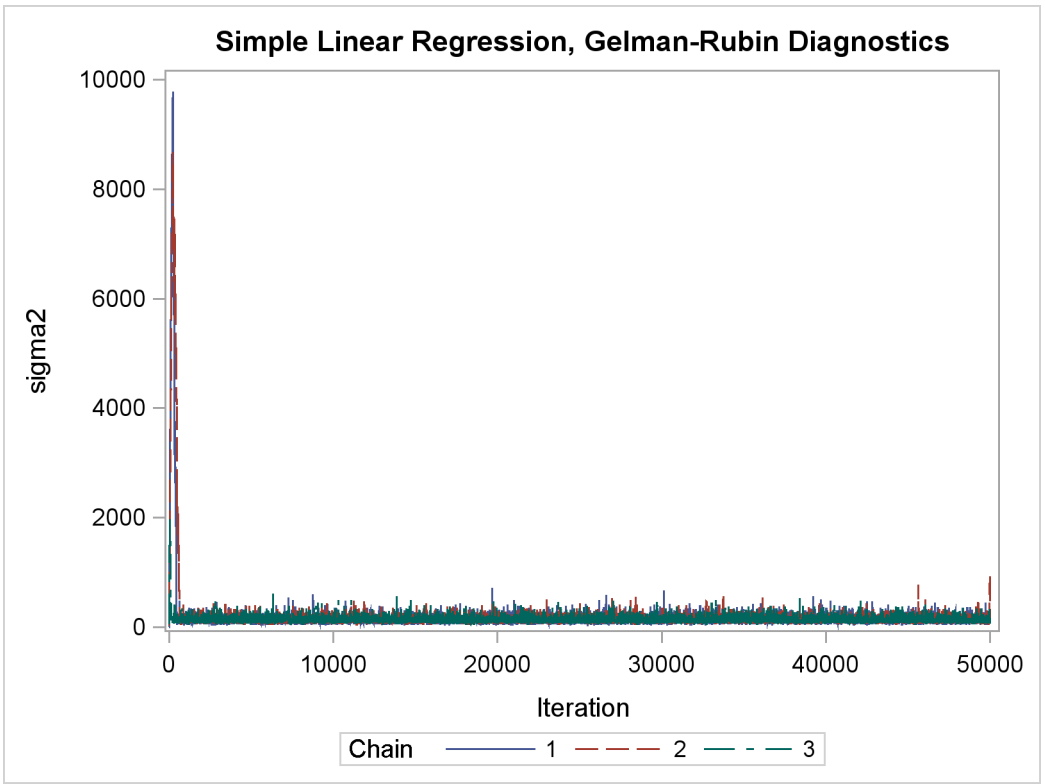
```
/* plot the trace plots of three Markov chains. */
%macro trace;
   %do i = 1 %to &nparm;
        proc sgplot data=all cycleattrs;
           series x=Iteration y=%scan(&var, &i) / group=Chain;
        run;
   %end;
%mend;
%trace;
```

Output 52.13.2 Trace Plots of Three Chains for Each of the Parameters



Output 52.13.2 *continued*





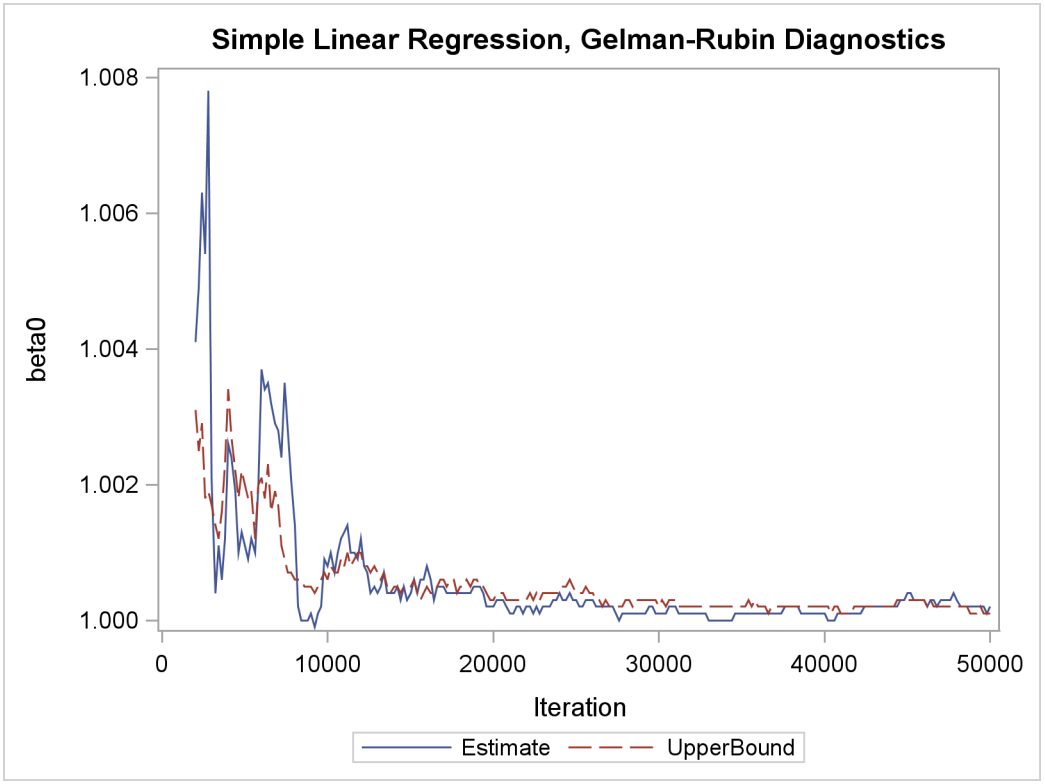
The trace plots show that three chains all eventually converge to the same regions even though they

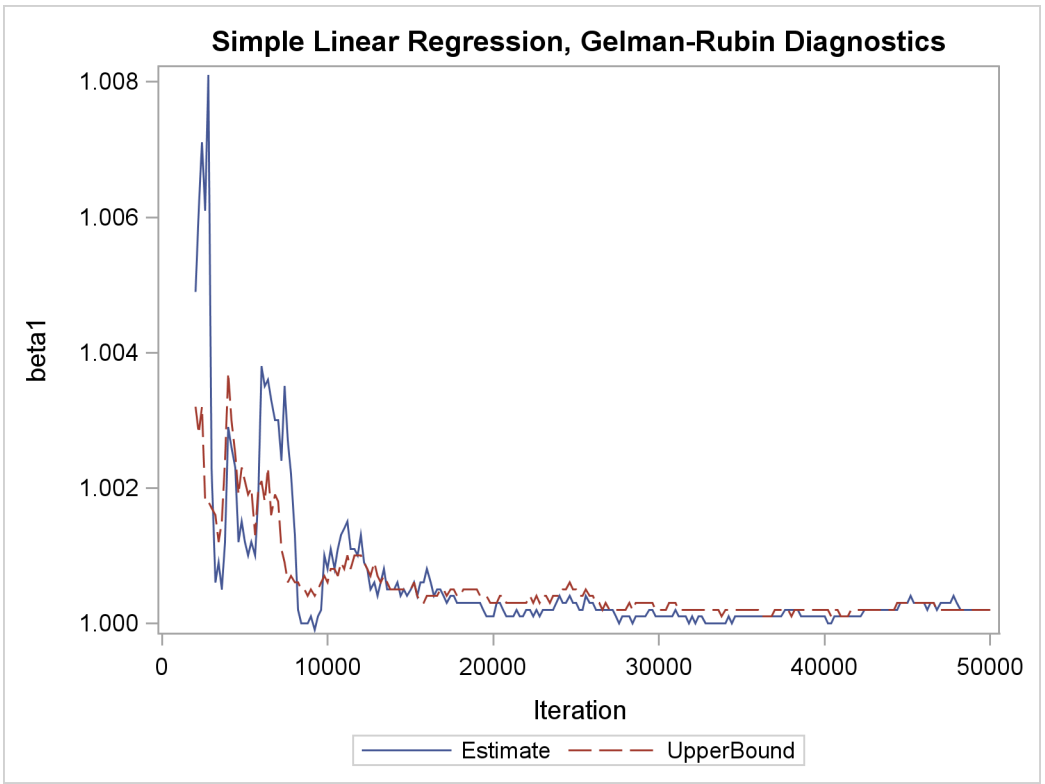
started at very different locations. In addition to the trace plots, you can also plot the potential scale reduction factor (PSRF). See the section "Gelman and Rubin Diagnostics" on page 161 for definition and details.

The following statements calculate PSRF for each parameter. They use the GELMAN macro repeatedly and can take a while to run:

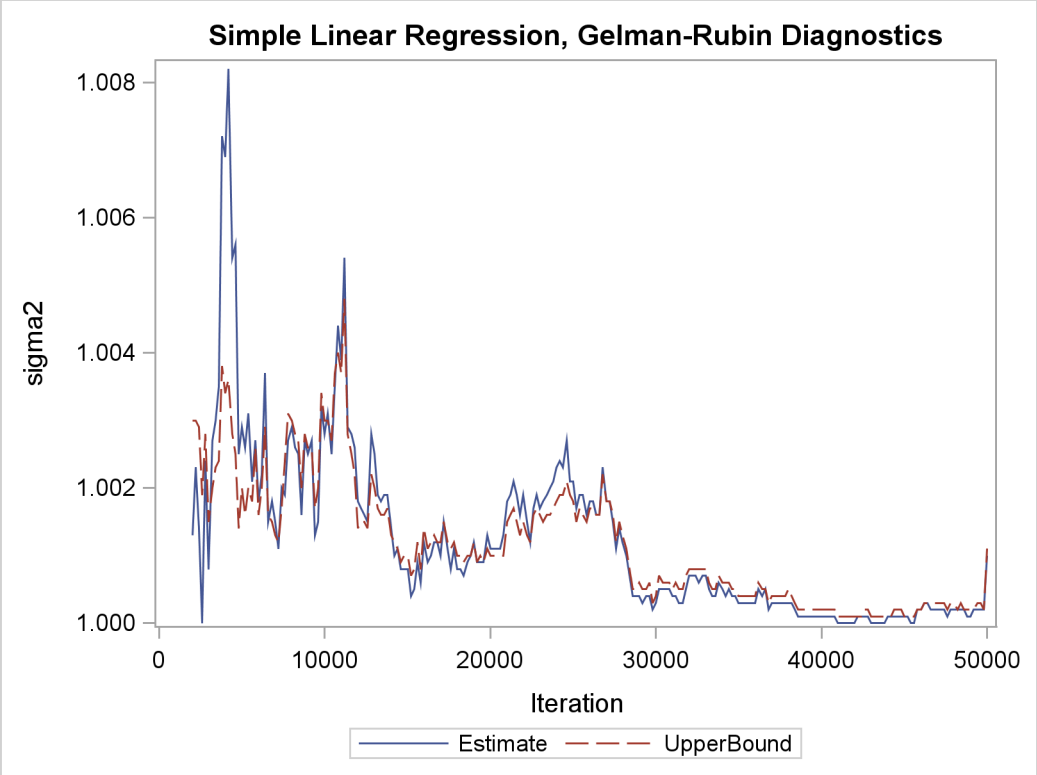
```
/* define sliding window size */
%let nwin = 200;
data PSRF;
run;

%macro PSRF(nsim);
   %do k = 1 %to %sysevalf(&nsim/&nwin, floor);
      %gelman(all, &nparm, &var, nsim=%sysevalf(&k*&nwin));
      data GelmanRubin;
          merge _Gelman_Parms _Gelman_Ests;
      run;

      data PSRF;
         set PSRF GelmanRubin;
      run;
   %end;
%mend PSRF;

options nonotes;
%PSRF(&nsim);
options notes;

data PSRF;
   set PSRF;
   if _n_ = 1 then delete;
run;

proc sort data=PSRF;
   by Parameter;
run;

%macro sepPSRF(nparm=, var=, nsim=);
   %do k = 1 %to &nparm;
       data save&k; set PSRF;
          if _n_ > %sysevalf(&k*&nsim/&nwin, floor) then delete;
          if _n_ < %sysevalf((&k-1)*&nsim/&nwin + 1, floor) then delete;
          Iteration + &nwin;
       run;

       proc sgplot data=save&k(firstobs=10) cycleattrs;
          series x=Iteration y=Estimate;
          series x=Iteration y=upperbound;
          yaxis label="%scan(&var, &k)";
       run;
   %end;
%mend sepPSRF;

%sepPSRF(nparm=&nparm, var=&var, nsim=&nsim);
```

Output 52.13.3 PSRF Plot for Each Parameter





Output 52.13.3 *continued*



PSRF is the square root of the ratio of the between-chain variance and the within-chain variance. A large PSRF indicates that the between-chain variance is substantially greater than the within-chain variance, so that longer simulation is needed. You want the PSRF to converge to 1 eventually, as it appears to be the case in this simulation study.

References

Aitkin, M., Anderson, D., Francis, B., and Hinde, J. (1989), *Statistical Modelling in GLIM*, Oxford: Oxford Science Publications.

Atkinson, A. C. (1979), "The Computer Generation of Poisson Random Variables," *Applied Statistics*, 28, 29–35.

Atkinson, A. C. and Whittaker, J. (1976), "A Switching Algorithm for the Generation of Beta Random Variables with at Least One Parameter Less Than One," *Proceedings of the Royal Society of London, Series A*, 139, 462–467.

Bacon, D. W. and Watts, D. G. (1971), "Estimating the Transition between Two Intersecting Straight Lines," *Biometrika*, 58, 525–534.

Berger, J. O. (1985), *Statistical Decision Theory and Bayesian Analysis*, Second Edition, New York: Springer-Verlag.

Box, G. E. P. and Cox, D. R. (1964), "An Analysis of Transformations," *Journal of the Royal Statistics Society, Series B*, 26, 211–234.

Carlin, B. P., Gelfand, A. E., and Smith, A. F. M. (1992), "Hierarchical Bayesian Analysis of Changepoint Problems," *Applied Statistics*, 41(2), 389–405.

Chaloner, K. (1994), "Residual Analysis and Outliers in Bayesian Hierarchical Models," in *Aspects of Uncertainty: A Tribute to D. V. Lindley*, 149–157, New York: Wiley.

Chaloner, K. and Brant, R. (1988), "A Bayesian Approach to Outlier Detection and Residual Analysis," *Biometrika*, 75(4), 651–659.

Cheng, R. C. H. (1978), "Generating Beta Variates with Non-integral Shape Parameters," *Communications ACM*, 28, 290–295.

Congdon, P. (2003), *Applied Bayesian Modeling*, John Wiley & Sons.

Crowder, M. J. (1978), "Beta-Binomial Anova for Proportions," *Applied Statistics*, 27, 34–37.

Draper, D. (1996), "Discussion of the Paper by Lee and Nelder," *Journal of the Royal Statistical Society, Series B*, 58, 662–663.

Eilers, P. H. C. and Marx, B. D. (1996), "Flexible Smoothing with *B*-Splines and Penalties," *Statistical Science*, 11, 89–121, with discussion.

Finney, D. J. (1947), "The Estimation from Individual Records of the Relationship between Dose and Quantal Response," *Biometrika*, 34, 320–334.

Fisher, R. A. (1935), "The Fiducial Argument in Statistical Inference," *Annals of Eugenics*, 6, 391–398.

Fishman, G. S. (1996), *Monte Carlo: Concepts, Algorithms, and Applications*, New York: John Wiley & Sons.

Gaver, D. P. and O'Muircheartaigh, I. G. (1987), "Robust Empirical Bayes Analysis of Event Rates," *Technometrics*, 29, 1–15.

Gelman, A., Carlin, J., Stern, H., and Rubin, D. (2004), *Bayesian Data Analysis*, Second Edition, London: Chapman & Hall.

Gentleman, R. and Geyer, C. J. (1994), "Maximum Likelihood for Interval Censored Data: Consistency and Computation," *Biometrika*, 81, 618–623.

Gilks, W. (2003), "Adaptive Metropolis Rejection Sampling (ARMS)," software from MRC Biostatistics Unit, Cambridge, UK, http://www.maths.leeds.ac.uk/~wally.gilks/adaptive.rejection/web_page/Welcome.html.

Gilks, W. R. and Wild, P. (1992), "Adaptive Rejection Sampling for Gibbs Sampling," *Applied Statistics*, 41, 337–348.

Holmes, C. C. and Held, L. (2006), "Bayesian Auxiliary Variable Models for Binary and Multinomial Regression," *Bayesian Analysis*, 1(1), 145–168, http://ba.stat.cmu.edu/journal/2006/vol01/issue01/held.pdf.

Ibrahim, J. G., Chen, M. H., and Sinha, D. (2001), *Bayesian Survival Analysis*, New York: Springer-Verlag.

Kass, R. E., Carlin, B. P., Gelman, A., and Neal, R. (1998), "Markov Chain Monte Carlo in Practice: A Roundtable Discussion," *The American Statistician*, 52, 93–100.

Krall, J. M., Uthoff, V. A., and Harley, J. B. (1975), "A Step-up Procedure for Selecting Variables Associated with Survival," *Biometrics*, 31, 49–57.

Kuhfeld, W. F. (2004), *Conjoint Analysis*, Technical report, SAS Institute Inc., http://support.sas.com/resources/papers/tnote/tnote_marketresearch.html.

Matsumoto, M. and Kurita, Y. (1992), "Twisted GFSR Generators," *ACM Transactions on Modeling and Computer Simulation*, 2(3), 179–194.

Matsumoto, M. and Kurita, Y. (1994), "Twisted GFSR Generators," *ACM Transactions on Modeling and Computer Simulation*, 4(3), 254–266.

Matsumoto, M. and Nishimura, T. (1998), "Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator," *ACM Transactions on Modeling and Computer Simulation*, 8, 3–30.

McGrath, E. J. and Irving, D. C. (1973), *Techniques for Efficient Monte Carlo Simulation, Volume II: Random Number Generation for Selected Probability Distributions*, Technical report, Science Applications Inc., La Jolla, CA.

Michael, J. R., Schucany, W. R., and Haas, R. W. (1976), "Generating Random Variates Using Transformations with Multiple Roots," *American Statistician*, 30(2), 88–90.

Pregibon, D. (1981), "Logistic Regression Diagnostics," *Annals of Statistics*, 9, 705–724.

Ripley, B. D. (1987), *Stochastic Simulation*, New York: John Wiley & Sons.

Robert, C. (1995), "Simulation of Truncated Normal Variables," *Statistics and Computing*, 5, 121–125.

Roberts, G. O., Gelman, A., and Gilks, W. R. (1997), "Weak Convergence and Optimal Scaling of Random Walk Metropolis Algorithms," *Annual of Applied Probability*, 7, 110–120.

Roberts, G. O. and Rosenthal, J. S. (2001), "Optimal Scaling for Various Metropolis-Hastings Algorithms," *Statistical Science*, 16, 351–367.

Rubin, D. B. (1981), "Estimation in Parallel Randomized Experiments," *Journal of Educational Statistics*, 6, 377–411.

Schervish, M. J. (1995), *Theory of Statistics*, New York: Springer-Verlag.

Sharples, L. (1990), "Identification and Accommodation of Outliers in General Hierarchical Models," *Biometrika*, 77, 445–453.

Spiegelhalter, D. J., Thomas, A., Best, N. G., and Gilks, W. R. (1996), "BUGS Examples, Volume 2, Version 0.5, (version ii)," .

Chapter 53

The MDS Procedure

Contents

Overview: MDS Procedure **3704**
Getting Started: MDS Procedure **3706**
Syntax: MDS Procedure **3709**
PROC MDS Statement 3710
BY Statement 3721
ID Statement 3721
INVAR Statement 3722
MATRIX Statement 3722
VAR Statement 3722
WEIGHT Statement 3723
Details: MDS Procedure **3723**
Formulas 3723
OUT= Data Set 3725
OUTFIT= Data Set 3726
OUTRES= Data Set 3727
INITIAL= Data Set 3728
Missing Values 3728
Normalization of the Estimates 3728
Comparison with Earlier Procedures 3729
Displayed Output 3730
ODS Table Names 3731
ODS Graphics 3731
Example: MDS Procedure **3732**
Example 53.1: Jacobowitz Body Parts Data from Children and Adults . . . 3732
References **3741**

Overview: MDS Procedure

Multidimensional scaling (MDS) refers to a class of methods. These methods estimate coordinates for a set of objects in a space of specified dimensionality. The input data are measurements of distances between pairs of objects. A variety of models can be used that include different ways of computing distances and various functions relating the distances to the actual data. The MDS procedure fits two- and three-way, metric and nonmetric multidimensional scaling models.

The data for the MDS procedure consist of one or more square symmetric or asymmetric matrices of similarities or dissimilarities between *objects* or *stimuli* (Kruskal and Wish 1978, pp. 7–11). Such data are also called *proximity* data. In psychometric applications, each matrix typically corresponds to a *subject*, and models that fit different parameters for each subject are called *individual difference* models.

Missing values are permitted. In particular, if the data are all missing except within some off-diagonal rectangle, the analysis is called *unfolding*. There are, however, many difficulties intrinsic to unfolding models (Heiser 1981). PROC MDS does not perform external unfolding; for analyses requiring external unfolding, use the TRANSREG procedure instead.

The MDS procedure estimates the following parameters by nonlinear least squares:

configuration
: the coordinates of each object in a Euclidean (Kruskal and Wish 1978, pp. 17–19) or weighted Euclidean space (Kruskal and Wish 1978, pp. 61–63) of one or more dimensions

dimension coefficients
: for each data matrix, the coefficients that multiply each coordinate of the *common* or *group* weighted Euclidean space to yield the *individual* unweighted Euclidean space. These coefficients are the square roots of the *subject weights* (Kruskal and Wish 1978, pp. 61–63). A plot of the dimension coefficients is directly interpretable in that it shows how a unit square in the group space is transformed to a rectangle in each individual space. A plot of subject weights has no such simple interpretation. The weighted Euclidean model is related to the INDSCAL model (Carroll and Chang 1970).

transformation parameters
: intercept, slope, or exponent in a linear, affine, or power transformation relating the distances to the data (Kruskal and Wish 1978, pp. 19–22). For a nonmetric analysis, monotone transformations involving no explicit parameters are used (Kruskal and Wish 1978, pp. 22–25). For a discussion of metric versus nonmetric transformations, see Kruskal and Wish (1978, pp. 76–78).

Depending on the LEVEL= option, PROC MDS fits either a regression model of the form

$$fit(datum) = fit(trans(distance)) + error$$

or a measurement model of the form

$$fit(trans(datum)) = fit(distance) + error$$

where

fit is a predetermined power or logarithmic transformation specified by the FIT= option.

trans is an estimated ("optimal") linear, affine, power, or monotone transformation specified by the LEVEL= option.

datum is a measure of the similarity or dissimilarity of two objects or stimuli.

distance is a distance computed from the estimated coordinates of the two objects and estimated dimension coefficients in a space of one or more dimensions. If there are no dimension coefficients (COEF=IDENTITY), this is an unweighted Euclidean distance. If dimension coefficients are used (COEF=DIAGONAL), this is a weighted Euclidean distance where the weights are the squares of the dimension coefficients; alternatively, you can multiply each dimension by its coefficient and compute an unweighted Euclidean distance.

error is an error term assumed to have an approximately normal distribution and to be independently and identically distributed for all data. Under these assumptions, least-squares estimation is statistically appropriate.

For an introduction to multidimensional scaling, see Kruskal and Wish (1978) and Arabie, Carroll, and DeSarbo (1987). A more advanced treatment is given by Young (1987). Many practical issues of data collection and analysis are discussed in Schiffman, Reynolds, and Young (1981). The fundamentals of psychological measurement, including both unidimensional and multidimensional scaling, are expounded by Torgerson (1958). Nonlinear least-squares estimation of PROC MDS models is discussed in Null and Sarle (1982).

Getting Started: MDS Procedure

The simplest application of PROC MDS is to reconstruct a map from a table of distances between points on the map (Kruskal and Wish 1978, pp. 7–9). For example, the following DATA step reads a table of flying mileages between 10 U.S. cities:

```
data city;
   title 'Analysis of Flying Mileages between Ten U.S. Cities';
   input (Atlanta Chicago Denver Houston LosAngeles
          Miami NewYork SanFrancisco Seattle WashingtonDC) (5.)
          @56 City $15.;
   datalines;
    0                                                    Atlanta
  587    0                                               Chicago
 1212  920    0                                          Denver
  701  940  879    0                                     Houston
 1936 1745  831 1374    0                                Los Angeles
  604 1188 1726  968 2339    0                           Miami
  748  713 1631 1420 2451 1092    0                      New York
 2139 1858  949 1645  347 2594 2571    0                San Francisco
 2182 1737 1021 1891  959 2734 2408  678    0           Seattle
  543  597 1494 1220 2300  923  205 2442 2329    0      Washington D.C.
;
```

Since the flying mileages are very good approximations to Euclidean distance, no transformation is needed to convert distances from the model to data. The analysis can therefore be done at the absolute level of measurement, as displayed in the following PROC MDS step (LEVEL=ABSOLUTE). The following statements produce Figure 53.1 and Figure 53.2:

```
ods graphics on;

proc mds data=city level=absolute;
   id city;
run;

ods graphics off;
```

PROC MDS first displays the iteration history. In this example, only one iteration is required. The badness-of-fit criterion 0.001689 indicates that the data fit the model extremely well. You can also see that the fit is excellent in the fit plot in Figure 53.2.

Figure 53.1 Iteration History from PROC MDS

```
          Analysis of Flying Mileages between Ten U.S. Cities

             Multidimensional Scaling:  Data=WORK.CITY.DATA
             Shape=TRIANGLE Condition=MATRIX Level=ABSOLUTE
               Coef=IDENTITY Dimension=2 Formula=1 Fit=1

             Gconverge=0.01 Maxiter=100 Over=1 Ridge=0.0001


                                 Badness-
                                   of-Fit    Change in    Convergence
      Iteration    Type         Criterion    Criterion        Measure
      ------------------------------------------------------------------
            0      Initial      0.003273             .         0.8562
            1      Lev-Mar      0.001689     0.001584       0.005128

     Convergence criterion is satisfied.
```

While PROC MDS can recover the relative positions of the cities, it cannot determine absolute location or orientation. In this case, north is toward the bottom of the plot. (See the first plot in Figure 53.2.)

Figure 53.2 Plot of Estimated Configuration and Fit

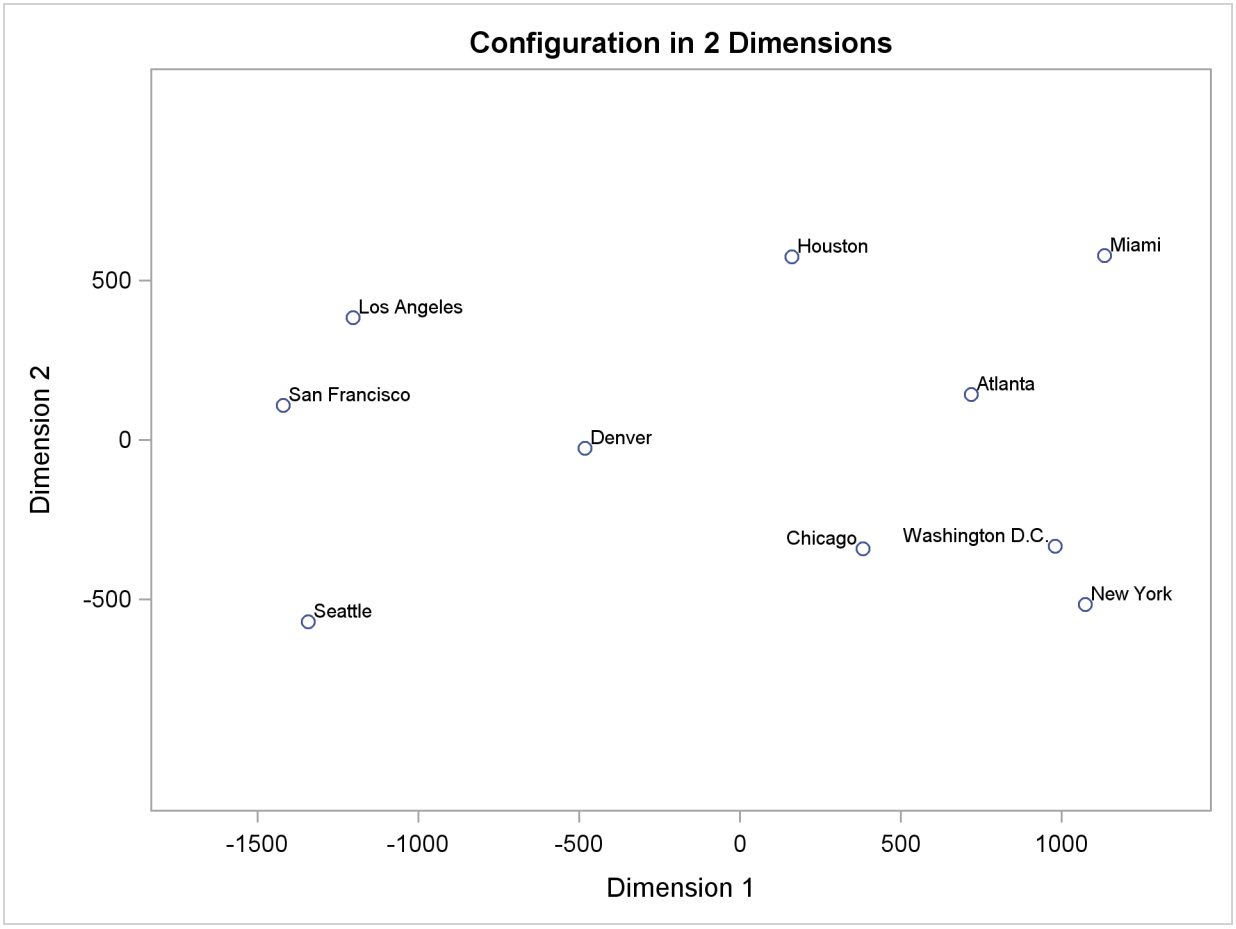
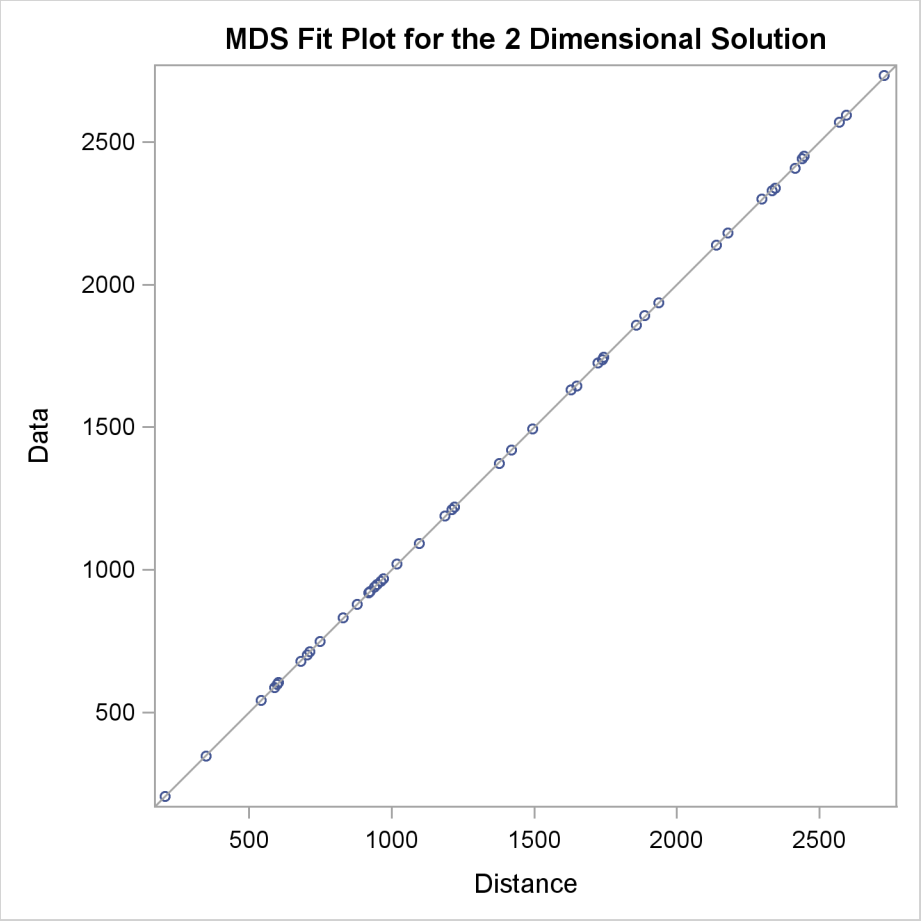


Figure 53.2 *continued*



Syntax: MDS Procedure

You can specify the following statements with the MDS procedure:

PROC MDS < *options* > ;
VAR *variables* ;
INVAR *variables* ;
ID | **OBJECT** *variable* ;
MATRIX | **SUBJECT** *variable* ;
WEIGHT *variables* ;
BY *variables* ;

The PROC MDS statement is required. All other statements are optional.

PROC MDS Statement

PROC MDS < *options* > **;**

PROC MDS produces an iteration history by default. Graphical displays are produced when you specify the ODS GRAPHICS ON statement before running PROC MDS. Additional displayed output is controlled by the interaction of the PCONFIG, PCOEF, PTRANS, PFIT, and PFITROW options with the PININ, PINIT, PITER, and PFINAL options. The PCONFIG, PCOEF, PTRANS, PFIT, and PFITROW options specify *which* estimates and fit statistics are to be displayed. The PININ, PINIT, PITER, and PFINAL options specify *when* the estimates and fit statistics are to be displayed. If you specify at least one of the PCONFIG, PCOEF, PTRANS, PFIT, and PFITROW options but none of the PININ, PINIT, PITER, and PFINAL options, the final results (PFINAL) are displayed. If you specify at least one of the PININ, PINIT, PITER, and PFINAL options but none of the PCONFIG, PCOEF, PTRANS, PFIT, and PFITROW options, all estimates (PCONFIG, PCOEF, PTRANS) and the fit statistics for each matrix and for the entire sample (PFIT) are displayed. If you do not specify any of these nine options, no estimates or fit statistics are displayed (except the badness-of-fit criterion in the iteration history).

The types of estimates written to the OUT= data set are determined by the OCONFIG, OCOEF, OTRANS, and OCRIT options. If you do not specify any of these four options, the estimates of all the parameters of the PROC MDS model and the value of the badness-of-fit criterion appear in the OUT= data set. If you specify one or more of these options, only the information requested by the specified options appears in the OUT= data set. Also, the OITER option causes these statistics to be written to the OUT= data set after initialization and on each iteration, as well as after the iterations have terminated.

See Table 53.2 for a list of options available in the PROC MDS statement.

Table 53.2 Summary of PROC MDS Statement Options

Option	Description
Data Set Options	
DATA=	specifies the input SAS data set
INITIAL=	specifies the input SAS data set containing initial values
OUT=	specifies the output data set
OUTFIT=	specifies the output fit data set
OUTRES=	specifies the output residual data set
Input Control	
CUTOFF=	replaces data values with missing values
SHAPE=	specifies the shape of the input data matrices
SIMILAR=	specifies that the data are similarity measurements
Model	
COEF=	specifies the type of matrix for the coefficients
CONDITION=	specifies the conditionality of the data
DIMENSION=	specifies the number of dimensions
LEVEL=	specifies the measurement level
NEGATIVE	permits slopes or powers to be negative

Table 53.2 *continued*

Option	Description
UNTIE	permits tied data to be untied
Initialization	
INAV=	affects the computation of initial coordinates
NOULB	specifies the missing data initialization
RANDOM=	specifies initial random coordinates
Estimation	
ALTERNATE=	specifies the alternating-least-squares algorithm
CONVERGE=	specifies the convergence criterion
EPSILON=	specifies the amount added to squared distances
FIT=	specifies a predetermined transformation
FORMULA=	specifies the badness-of-fit formula
GCONVERGE=	specifies the gradient convergence criterion
MAXITER=	specifies the maximum number of iterations
MCONVERGE=	specifies the monotone convergence criterion
MINCRIT=	specifies the minimum badness-of-fit criterion
NONORM	suppresses normalization of the initial and final estimates
OVER=	specifies the maximum overrelaxation factor
RIDGE=	specifies the initial ridge value
SINGULAR=	specifies the singularity criterion
Control Output Data Set Contents	
OCOEF	writes the dimension coefficients to the OUT= data set
OCONFIG	writes the coordinates of the objects to the OUT= data set
OCRIT	writes the badness-of-fit criterion to the OUT= data set
OITER	writes current values after initialization and on every iteration
OTRANS	writes the transformation parameter estimates to the OUT= data set
Control Displayed Output	
DECIMALS=	specifies how many decimal places to use
NOPHIST	suppresses the iteration history
PCOEF	displays the estimated dimension coefficients
PCONFIG	displays the estimated coordinates
PDATA	displays each data matrix
PFINAL	displays final estimates
PFIT	displays the badness-of-fit criterion
PFITROW	displays the badness-of-fit criterion for each row
PINAVDATA	displays INAV= data set information
PINEIGVAL	displays the initial eigenvalues
PINEIGVEC	displays the initial eigenvectors
PININ	displays values read from the INITIAL= data set
PINIT	displays initial values
PITER	displays estimates on each iteration
PLOTS=	controls the graphical displays
PTRANS	displays the estimated transformation parameters

ALTERNATE | ALT=NONE | NO | N

ALTERNATE | ALT=MATRIX | MAT | M | SUBJECT | SUB | S

ALTERNATE | ALT=ROW | R *<=n>*

determines what form of alternating-least-squares algorithm is used. The default depends on the amount of memory available. The following ALTERNATE= options are listed in order of decreasing memory requirements:

ALT=NONE causes all parameters to be adjusted simultaneously on each iteration. This option is usually best for a small number of subjects and objects.

ALT=MATRIX adjusts all the parameters for the first subject, then all the parameters for the second subject, and so on, and finally adjusts all parameters that do not correspond to a subject, such as coordinates and unconditional transformations. This option usually works best for a large number of subjects with a small number of objects.

ALT=ROW treats subject parameters the same way as the ALTERNATE=MATRIX option but also includes separate stages for unconditional parameters and for subsets of the objects. The ALT=ROW option usually works best for a large number of objects. Specifying ALT=ROW=*n* divides the objects into subsets of *n* objects each, except possibly for one subset when *n* does not divide the number of objects evenly. If you omit =*n*, the number of objects in the subsets is determined from the amount of memory available. The smaller the value of *n*, the less memory is required.

When you specify the LEVEL=ORDINAL option, the monotone transformation is always computed in a separate stage and is listed as a separate iteration in the iteration history. In this case, estimation is done by iteratively reweighted least squares. The weights are recomputed according to the FORMULA= option on each monotone iteration; hence, it is possible for the badness-of-fit criterion to increase after a monotone iteration.

COEF=IDENTITY | IDEN | I

COEF=DIAGONAL | DIAG | D

specifies the type of matrix for the dimension coefficients.

COEF=IDENTITY is the default, which yields Euclidean distances.

COEF=DIAGONAL produces weighted Euclidean distances, in which each subject can have different weights for the dimensions. The dimension coefficients that PROC MDS outputs are related to the square roots of what are called subject weights in PROC ALSCAL; the normalization in PROC MDS also differs from that in PROC ALSCAL. The weighted Euclidean model is related to the INDSCAL model (Carroll and Chang 1970).

CONDITION | COND=UN | U

CONDITION | COND=MATRIX | MAT | M | SUBJECT | SUB | S

CONDITION | COND=ROW | R

specifies the conditionality of the data (Young 1987, pp. 60–63). The data are divided into disjoint subsets called *partitions*. Within each partition, a separate transformation is applied, as specified by the LEVEL= option. The three types of conditionality are as follows:

COND=UN	(unconditional) puts all the data into a single partition.
COND=MATRIX	(matrix conditional) makes each data matrix a partition.
COND=ROW	(row conditional) makes each row of each data matrix a partition.

The default is CONDITION=MATRIX. The CONDITION= option also determines the default value for the SHAPE= option. If you specify the CONDITION=ROW option and omit the SHAPE= option, each data matrix is stored as a square and possibly asymmetric matrix. If you specify the CONDITION=UN or CONDITION=MATRIX option and omit the SHAPE= option, only one triangle is stored. See the SHAPE= option for details.

CONVERGE | CONV=*p*

sets both the gradient convergence criterion and the monotone convergence criterion to p, where $0 \leq p \leq 1$. The default is CONVERGE=0.01; smaller values might greatly increase the number of iterations required. Values less than 0.0001 might be impossible to satisfy because of the limits of machine precision. (See the GCONVERGE= and MCONVERGE= options.)

CUTOFF=*n*

replaces data values less than n with missing values. The default is CUTOFF=0.

DATA=*SAS-data-set*

specifies the SAS data set containing one or more square matrices to be analyzed. In typical psychometric data, each matrix contains judgments from one subject, so there is a one-to-one correspondence between data matrices and subjects.

The data matrices contain similarity or dissimilarity measurements to be modeled and, optionally, weights for these data. The data are generally assumed to be dissimilarities unless you use the SIMILAR option. However, if there are nonmissing diagonal values and these values are predominantly larger than the off-diagonal values, the data are assumed to be similarities and are treated as if the SIMILAR option is specified. The diagonal elements are not otherwise used in fitting the model.

Each matrix must have exactly the same number of observations as the number of variables specified by the VAR statement or determined by defaults. This number is the number of objects or stimuli.

The first observation and variable are assumed to contain data for the first object, the second observation and variable are assumed to contain data for the second object, and so on.

When there are two or more matrices, the observations in each matrix must correspond to the same objects in the same order as in the first matrix.

The matrices can be symmetric or asymmetric, as specified by the SHAPE= option.

DECIMALS | DEC=*n*

specifies how many decimal places to use when displaying the parameter estimates and fit statistics. The default is DECIMALS=2, which is generally reasonable except in conjunction with the LEVEL=ABSOLUTE option and very large or very small data.

DIMENSION | DIMENS | DIM=*n* **< TO** *m* **< BY=***i* **>>**

specifies the number of dimensions to use in the MDS model, where $1 \leq n, m <$ *number of objects*. The parameter *i* can be either positive or negative but not zero. If you specify different values for *n* and *m*, a separate model is fitted for each requested dimension. If you specify only DIMENSION=*n*, then only *n* dimensions are fitted. The default is DIMENSION=2 if there are three or more objects; otherwise, DIMENSION=1 is the only valid specification. The analyses for each number of dimensions are done independently. For information about choosing the dimensionality, see Kruskal and Wish (1978 , pp. 48–60).

EPSILON | EPS=*n*

specifies a number n, $0 < n < 1$, that determines the amount added to squared distances computed from the model to avoid numerical problems such as division by 0. This amount is computed as ϵ equal to n times the mean squared distance in the initial configuration. The distance in the MDS model is thus computed as

$$distance = \sqrt{sqdist + \epsilon}$$

where *sqdist* is the squared Euclidean distance or the weighted squared Euclidean distance.

The default is EPSILON=1E–12, which is small enough to have no practical effect on the estimates unless the FIT= value is nonpositive and there are dissimilarities that are very close to 0. Hence, when the FIT= value is nonpositive, dissimilarities less than n times 100 times the maximum dissimilarity are not permitted.

FIT=DISTANCE | DIS | D

FIT=SQUARED | SQU | S

FIT=LOG | L

FIT=*n*

specifies a predetermined (not estimated) transformation to apply to both sides of the MDS model before the error term is added.

The default is FIT=DISTANCE or, equivalently, FIT=1, which fits data to distances.

The option FIT=SQUARED or FIT=2 fits squared data to squared distances. This gives greater importance to large data and distances and lesser importance to small data and distances in fitting the model.

The FIT=LOG or FIT=0 option fits log data to log distances. This gives lesser importance to large data and distances and greater importance to small data and distances in fitting the model.

In general, the FIT=*n* option fits *n*th-power data to *n*th-power distances. Values of *n* that are large in absolute value can cause floating-point overflows.

If the FIT= value is 0 or negative, the data must be strictly positive (see the EPSILON= option). Negative data might produce strange results with any value other than FIT=1.

FORMULA | FOR=*0* | OLS | O
FORMULA | FOR=*1* | USS | U
FORMULA | FOR=*2* | CSS | C

determines how the badness-of-fit criterion is standardized in correspondence with stress formulas 1 and 2 (Kruskal and Wish 1978, pp. 24–26). The default is FORMULA=1 unless you specify FIT=LOG, in which case the default is FORMULA=2. Data partitions are defined by the CONDITION= option.

FORMULA=0 fits a regression model by ordinary least squares (Null and Sarle 1982) without standardizing the partitions; this option cannot be used with the LEVEL=ORDINAL option. The badness-of-fit criterion is the square root of the error sum of squares.

FORMULA=1 standardizes each partition by the uncorrected sum of squares of the (possibly transformed) data; this option should not be used with the FIT=LOG option. With the FIT=DISTANCE and LEVEL=ORDINAL options, this is equivalent to Kruskal's stress formula 1 or an obvious generalization thereof. With the FIT=SQUARED and LEVEL=ORDINAL options, this is equivalent to Young's s-stress formula 1 or an obvious generalization thereof. The badness-of-fit criterion is analogous to $\sqrt{1 - R^2}$, where R is a multiple correlation about the origin.

FORMULA=2 standardizes each partition by the corrected sum of squares of the (possibly transformed) data; this option is the recommended method for unfolding. With the FIT=DISTANCE and LEVEL=ORDINAL options, this is equivalent to Kruskal's stress formula 2 or an obvious generalization thereof. With the FIT=SQUARED and LEVEL=ORDINAL options, this is equivalent to Young's s-stress formula 2 or an obvious generalization thereof. The badness-of-fit criterion is analogous to $\sqrt{1 - R^2}$, where R is a multiple correlation computed with a denominator corrected for the mean.

GCONVERGE | GCONV=*p*

sets the gradient convergence criterion to p, where $0 \leq p \leq 1$. The default is GCONVERGE=0.01; smaller values might greatly increase the number of iterations required. Values less than 0.0001 might be impossible to satisfy because of the limits of machine precision.

The gradient convergence measure is the multiple correlation of the Jacobian matrix with the residual vector, uncorrected for the mean. (See the CONVERGE= and MCONVERGE= options.)

INAV=DATA | D
INAV=SSCP | S

affects the computation of initial coordinates. The default is INAV=DATA.

INAV=DATA computes a weighted average of the data matrices. Its value is estimated only if an element is missing from every data matrix. The weighted average of the data matrices with missing values filled in is then converted to

a scalar products matrix (or what would be a scalar products matrix if the fit were perfect), from which the initial coordinates are computed.

INAV=SSCP estimates missing values in each data matrix and converts each data matrix to a scalar products matrix. The initial coordinates are computed from the unweighted average of the scalar products matrices.

INITIAL | IN=*SAS-data-set*

specifies a SAS data set containing initial values for some or all of the parameters of the MDS model. If the INITIAL= option is omitted, the initial values are computed from the data.

LEVEL=ABSOLUTE | ABS | A

LEVEL=RATIO | RAT | R

LEVEL=INTERVAL | INT | I

LEVEL=LOGINTERVAL | LOG | L

LEVEL=ORDINAL | ORD | O

specifies the measurement level of the data and hence the type of estimated (optimal) transformations applied to the data or distances (Young 1987, pp. 57–60; Krantz et al. 1971, pp. 9–12) within each partition as specified by the CONDITION= option. LEVEL=ORDINAL specifies a nonmetric analysis, while all other LEVEL= options specify metric analyses. The default is LEVEL=ORDINAL.

LEVEL=ABSOLUTE permits no optimal transformations. Hence, the distinction between regression and measurement models is irrelevant.

LEVEL=RATIO fits a regression model in which the distances are multiplied by a slope parameter in each partition (a linear transformation). In this case, the regression model is equivalent to the measurement model with the slope parameter reciprocated.

LEVEL=INTERVAL fits a regression model in which the distances are multiplied by a slope parameter and added to an intercept parameter in each partition (an affine transformation). In this case, the regression and measurement models differ if there is more than one partition.

LEVEL=LOGINTERVAL fits a regression model in which the distances are raised to a power and multiplied by a slope parameter in each partition (a power transformation).

LEVEL=ORDINAL fits a measurement model in which a least-squares monotone increasing transformation is applied to the data in each partition. At the ordinal measurement level, the regression and measurement models differ.

MAXITER | ITER=*n*

specifies the maximum number of iterations, where $n \geq 0$. The default is MAXITER=100.

MCONVERGE | MCONV=*p*

sets the monotone convergence criterion to p, where $0 \leq p \leq 1$, for use with the

LEVEL=ORDINAL option. The default is MCONVERGE=0.01; if you want greater precision, MCONVERGE=0.001 is usually reasonable, but smaller values might greatly increase the number of iterations required.

The monotone convergence criterion is the Euclidean norm of the change in the optimally scaled data divided by the Euclidean norm of the optimally scaled data, averaged across partitions defined by the CONDITION= option. (See the CONVERGE= and GCONVERGE= options.)

MINCRIT | CRITMIN=*n*
causes iteration to terminate when the badness-of-fit criterion is less than or equal to n, where $n \geq 0$. The default is MINCRIT=1E–6.

NEGATIVE
permits slopes or powers to be negative with the LEVEL=RATIO, INTERVAL, or LOGINTERVAL option.

NONORM
suppresses normalization of the initial and final estimates.

NOPHIST | NOPRINT | NOP
suppresses the output of the iteration history.

NOULB
causes missing data to be estimated during initialization by the average nonmissing value, where the average is computed according to the FIT= option. Otherwise, missing data are estimated by interpolating between the Rabinowitz (1976) upper and lower bounds.

OCOEF
writes the dimension coefficients to the OUT= data set. See the OUT= option for interactions with other options.

OCONFIG
writes the coordinates of the objects to the OUT= data set. See the OUT= option for interactions with other options.

OCRIT
writes the badness-of-fit criterion to the OUT= data set. See the OUT= option for interactions with other options.

OITER | OUTITER
writes current values to the output data sets after initialization and on every iteration. Otherwise, only the final values are written to any output data sets. (See the OUT=, OUTFIT=, and OUTRES= options.)

OTRANS
writes the transformation parameter estimates to the OUT= data set if any such estimates are computed. There are no transformation parameters with the LEVEL=ORDINAL option. See the OUT= option for interactions with other options.

OUT=*SAS-data-set*
creates a SAS data set containing, by default, the estimates of all the parameters of the PROC MDS model and the value of the badness-of-fit criterion. However, if you specify one or more of the OCONFIG, OCOEF, OTRANS, and OCRIT options, only the information requested by the specified options appears in the OUT= data set. (See also the OITER option.)

OUTFIT=*SAS-data-set*
creates a SAS data set containing goodness-of-fit and badness-of-fit measures for each partition as well as for the entire data set. (See also the OITER option.)

OUTRES=*SAS-data-set*
creates a SAS data set containing one observation for each nonmissing data value from the DATA= data set. Each observation contains the original data value, the estimated distance computed from the MDS model, transformed data and distances, and the residual. (See also the OITER option.)

OVER=*n*
specifies the maximum overrelaxation factor, where $n \geq 1$. Values between 1 and 2 are generally reasonable. The default is OVER=2 with the LEVEL=ORDINAL, ALTERNATE=MATRIX, or ALTERNATE=ROW option; otherwise, the default is OVER=1. Use this option only if you have convergence problems.

PCOEF
produces the estimated dimension coefficients.

PCONFIG
produces the estimated coordinates of the objects in the configuration.

PDATA
displays each data matrix.

PFINAL
displays final estimates.

PFIT
displays the badness-of-fit criterion and various types of correlations between the data and fitted values for each data matrix, as well as for the entire sample.

PFITROW
displays the badness-of-fit criterion and various types of correlations between the data and fitted values for each row, as well as for each data matrix and for the entire sample. This option works only with the CONDITION=ROW option.

PINAVDATA
displays the sum of the weights and the weighted average of the data matrices computed during initialization with the INAV=DATA option.

PINEIGVAL
displays the eigenvalues computed during initialization.

PINEIGVEC
displays the eigenvectors computed during initialization.

PININ
displays values read from the INITIAL= data set. Since these values might be incomplete, the PFIT and PFITROW options do not apply.

PINIT
displays initial values.

PITER
displays estimates on each iteration.

PLOTS<(*global-plot-option***)> <=** *plot-request* **<(***options***)>>**
PLOTS<(*global-plot-options***)> <= (***plot-request* **<(***options***)> <...** *plot-request* **<(***options***)>>)>**

specifies options that control the details of the plots. When you specify only one plot request, you can omit the parentheses around the plot request. You must enable ODS Graphics before requesting plots, for example, like this:

```
ods graphics on;

proc mds plots(flip);
run;

ods graphics off;
```

The global plot option is as follows:

FLIP
flips or interchanges the X-axis and Y-axis dimensions for configuration and coefficient plots.

The plot requests include the following:

COEFFICIENTS(ONE)
combines the INDSCAL coefficients panel of plots into a single plot. By default, the display consists of a panel with two plots. The vectors are displayed in the left plot, and the labels are displayed in the right plot. The right plot provides a magnification of the region of the vector endpoints. In contrast, the single display, requested by COEFFICIENTS(ONE), is more compact, but there is less room for vector labels. It is often easier to identify the vectors in the default display.

NONE
suppresses all plots.

By default, a fit plot is produced. When more then one dimension is requested, plots of the configuration are also plotted. For individual differences models with more than one dimension, the subject weights or coefficients are plotted. When more than one value is specified for the DIMENSION= option, the badness-of-fit plot is produced.

PTRANS

displays the estimated transformation parameters if any are computed. There are no transformation parameters with the LEVEL=ORDINAL option.

RANDOM< =*seed* >

causes initial coordinate values to be pseudo-random numbers. In one dimension, the pseudo-random numbers are uniformly distributed on an interval. In two or more dimensions, the pseudo-random numbers are uniformly distributed on the circumference of a circle or the surface of a (hyper)sphere.

RIDGE=*n*

specifies the initial ridge value, where $n \geq 0$. The default is RIDGE=1E–4.

If you get a floating-point overflow in the first few iterations, specify a larger value such as RIDGE=0.01, RIDGE=1, or RIDGE=100.

If you know that the initial estimates are very good, using RIDGE=0 might speed convergence.

SHAPE=TRIANGULAR | TRIANGLE | TRI | T

SHAPE=SQUARE | SQU | S

determines whether the entire data matrix for each subject or only one triangle of the matrix is stored and analyzed. If you specify the CONDITION=ROW option, the default is SHAPE=SQUARE. Otherwise, the default is SHAPE=TRIANGLE.

SHAPE=SQUARE causes the entire matrix to be stored and analyzed. The matrix can be asymmetric.

SHAPE=TRIANGLE causes only one triangle to be stored. However, PROC MDS reads both upper and lower triangles to look for nonmissing values and to symmetrize the data if needed. If corresponding elements in the upper and lower triangles both contain nonmissing values, only the average of the two values is stored and analyzed (Kruskal and Wish 1978 , p. 74). Also, if an OUTRES= data set is requested, only the average of the two corresponding elements is output.

SIMILAR | SIM< =*max* >

causes the data to be treated as similarity measurements rather than dissimilarities. If =*max* is not specified, each data value is converted to a dissimilarity by subtracting it from the maximum value in the data set or BY group. If =*max* is specified, each data value is subtracted from the maximum of *max* and the data. The diagonal data are included in computing these maxima.

By default, the data are assumed to be dissimilarities unless there are nonmissing diagonal values and these values are predominantly larger than the off-diagonal values. In this case, the data are assumed to be similarities and are treated as if the SIMILAR option is specified.

SINGULAR=*p*

specifies the singularity criterion p, $0 \leq p \leq 1$. The default is SINGULAR=1E–8.

UNTIE

permits tied data to be assigned different optimally scaled values with the LEVEL=ORDINAL option. Otherwise, tied data are assigned equal optimally scaled values. The UNTIE option has no effect with values of the LEVEL= option other than LEVEL=ORDINAL.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MDS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC MDS. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

If the INITIAL= data set contains the BY variables, the BY groups must appear in the same order as in the DATA= data set. If the BY variables are not in the INITIAL= data set, the entire data set is used to provide initial values for each BY group in the DATA= data set.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

ID Statement

ID | OBJECT | OBJ *variable* **;**

The ID statement specifies a variable in the DATA= data set that contains descriptive labels for the objects. The labels are used in the output and are copied to the OUT= data set. If there is more than one data matrix, only the ID values from the observations containing the first data matrix are used.

The ID variable is not used to establish any correspondence between observations and variables.

If the ID statement is omitted, the variable labels or names are used as object labels.

INVAR Statement

INVAR *variables* **;**

The INVAR statement specifies the numeric variables in the INITIAL= data set that contain initial parameter estimates. The first variable corresponds to the first dimension, the second variable to the second dimension, and so on.

If the INVAR statement is omitted, the variables Dim1, . . . , Dim*m* are used, where *m* is the maximum number of dimensions.

MATRIX Statement

MATRIX | **MAT** | **SUBJECT** | **SUB** *variable* **;**

The MATRIX statement specifies a variable in the DATA= data set that contains descriptive labels for the data matrices or subjects. The labels are used in the output and are copied to the OUT= and OUTRES= data sets. Only the first observation from each data matrix is used to obtain the label for that matrix.

If the MATRIX statement is omitted, the matrices are labeled 1, 2, 3, and so on.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the numeric variables in the DATA= data set that contain similarity or dissimilarity measurements on a set of objects or stimuli. Each variable corresponds to one object.

If the VAR statement is omitted, all numeric variables that are not specified in another statement are used.

To analyze a subset of the objects in a data set, you can specify the variable names corresponding to the columns in the subset, but you must also use a DATA step or a WHERE clause to specify the rows in the subset. PROC MDS expects to read one or more square matrices, and you must ensure that the rows in the data set correctly correspond to the columns in number and order.

WEIGHT Statement

WEIGHT *variables* **;**

The WEIGHT statement specifies numeric variables in the DATA= data set that contain weights for each similarity or dissimilarity measurement. These weights are used to compute weighted least-squares estimates. The number of WEIGHT variables must be the same as the number of VAR variables, and the variables in the WEIGHT statement must be in the same order as the corresponding variables in the VAR statement.

If the WEIGHT statement is omitted, all data within a partition are assigned equal weights.

Data with zero or negative weights are ignored in fitting the model but are included in the OUTRES= data set and in monotone transformations.

Details: MDS Procedure

Formulas

The following notation is used:

A_p	intercept for partition p
B_p	slope for partition p
C_p	power for partition p
D_{rcs}	distance computed from the model between objects r and c for subject s
F_{rcs}	data weight for objects r and c for subject s obtained from the cth WEIGHT variable, or 1 if there is no WEIGHT statement
f	value of the FIT= option
N	number of objects
O_{rcs}	observed dissimilarity between objects r and c for subject s
P_{rcs}	partition index for objects r and c for subject s
Q_{rcs}	dissimilarity after applying any applicable estimated transformation for objects r and c for subject s
R_{rcs}	residual for objects r and c for subject s
S_p	standardization factor for partition p
$T_p(\cdot)$	estimated transformation for partition p
V_{sd}	coefficient for subject s on dimension d

X_{nd} coordinate for object n on dimension d

Summations are taken over nonmissing values.

Distances are computed from the model as

$$\begin{aligned} D_{rcs} &= \sqrt{\sum_d (X_{rd} - X_{cd})^2} && \text{for COEF=IDENTITY: Euclidean distance} \\ &= \sqrt{\sum_d V_{sd}^2 (X_{rd} - X_{cd})^2} && \text{for COEF=DIAGONAL: weighted Euclidean distance} \end{aligned}$$

Partition indexes are

$$\begin{aligned} P_{rcs} &= 1 && \text{for CONDITION=UN} \\ &= s && \text{for CONDITION=MATRIX} \\ &= (s-1)N + r && \text{for CONDITION=ROW} \end{aligned}$$

The estimated transformation for each partition is

$$\begin{aligned} T_p(d) &= d && \text{for LEVEL=ABSOLUTE} \\ &= B_p d && \text{for LEVEL=RATIO} \\ &= A_p + B_p d && \text{for LEVEL=INTERVAL} \\ &= B_p d^{C_p} && \text{for LEVEL=LOGINTERVAL} \end{aligned}$$

For LEVEL=ORDINAL, $T_p(\cdot)$ is computed as a least-squares monotone transformation.

For LEVEL=ABSOLUTE, RATIO, or INTERVAL, the residuals are computed as

$$\begin{aligned} Q_{rcs} &= O_{rcs} \\ R_{rcs} &= Q_{rcs}^f - [T_{P_{rcs}}(D_{rcs})]^f \end{aligned}$$

For LEVEL=ORDINAL, the residuals are computed as

$$\begin{aligned} Q_{rcs} &= T_{P_{rcs}}(O_{rcs}) \\ R_{rcs} &= Q_{rcs}^f - D_{rcs}^f \end{aligned}$$

If f is 0, then natural logarithms are used in place of the fth powers.

For each partition, let

$$U_p = \frac{\sum_{r,c,s} F_{rcs}}{\sum_{r,c,s|P_{rcs}=p} F_{rcs}}$$

and

$$\overline{Q}_p = \frac{\sum_{r,c,s|P_{rcs}=p} Q_{rcs} F_{rcs}}{\sum_{r,c,s|P_{rcs}=p} F_{rcs}}$$

Then the standardization factor for each partition is

$$\begin{aligned} S_p &= 1 && \text{for FORMULA=0} \\ &= U_p \sum_{r,c,s|P_{rcs}=p} Q_{rcs}^2 F_{rcs} && \text{for FORMULA=1} \\ &= U_p \sum_{r,c,s|P_{rcs}=p} (Q_{rcs} - \overline{Q}_p)^2 F_{rcs} && \text{for FORMULA=2} \end{aligned}$$

The badness-of-fit criterion that the MDS procedure tries to minimize is

$$\sqrt{\sum_{r,c,s} \frac{R_{rcs}^2 F_{rcs}}{S_{P_{rcs}}}}$$

OUT= Data Set

The OUT= data set contains the following variables:

- BY variables, if any
- _ITER_ (if the OUTITER option is specified), a numeric variable containing the iteration number
- _DIMENS_, a numeric variable containing the number of dimensions
- _MATRIX_ or the variable in the MATRIX statement, identifying the data matrix or subject to which the observation pertains. This variable contains a missing value for observations that pertain to the data set as a whole and not to a particular matrix, such as the coordinates (_TYPE_='CONFIG').
- _TYPE_, a character variable of length 10 identifying the type of information in the observation

  The values of _TYPE_ are as follows:

CONFIG	the estimated coordinates of the configuration of objects
DIAGCOEF	the estimated dimension coefficients for COEF=DIAGONAL
INTERCEPT	the estimated intercept parameters
SLOPE	the estimated slope parameters
POWER	the estimated power parameters
CRITERION	the badness-of-fit criterion

- _LABEL_ or the variable in the ID statement, containing the variable label or value of the ID variable of the object to which the observation pertains. This variable contains a missing value for observations that do not pertain to a particular object or dimension.
- _NAME_, a character variable containing the variable name of the object or dimension to which the observation pertains. This variable contains a missing value for observations that do not pertain to a particular object or dimension.
- Dim1, . . . , Dim*m*, where *m* is the maximum number of dimensions

OUTFIT= Data Set

The OUTFIT= data set contains various measures of goodness and badness of fit. There is one observation for the entire sample plus one observation for each matrix. For the CONDITION=ROW option, there is also one observation for each row.

The OUTFIT= data set contains the following variables:

- BY variables, if any
- _ITER_ (if the OUTITER option is specified), a numeric variable containing the iteration number
- _DIMENS_, a numeric variable containing the number of dimensions
- _MATRIX_ or the variable in the MATRIX statement, identifying the data matrix or subject to which the observation pertains
- _LABEL_ or the variable in the ID statement, containing the variable label or value of the ID variable of the object to which the observation pertains when CONDITION=ROW
- _NAME_, a character variable containing the variable name of the object or dimension to which the observation pertains when CONDITION=ROW
- N, the number of nonmissing data
- WEIGHT, the weight of the partition
- CRITER, the badness-of-fit criterion
- DISCORR, the correlation between the transformed data and the distances for LEVEL=ORDINAL or the correlation between the data and the transformed distances otherwise
- UDISCORR, the correlation uncorrected for the mean between the transformed data and the distances for LEVEL=ORDINAL or the correlation between the data and the transformed distances otherwise
- FITCORR, the correlation between the fit-transformed data and the fit-transformed distances
- UFITCORR, the correlation uncorrected for the mean between the fit-transformed data and the fit-transformed distances

OUTRES= Data Set

The OUTRES= data set has one observation for each nonmissing data value. It contains the following variables:

- BY variables, if any
- _ITER_ (if the OUTITER option is specified), a numeric variable containing the iteration number
- _DIMENS_, a numeric variable containing the number of dimensions
- _MATRIX_ or the variable in the MATRIX statement, identifying the data matrix or subject to which the observation pertains
- _ROW_, containing the variable label or value of the ID variable of the row to which the observation pertains
- _COL_, containing the variable label or value of the ID variable of the column to which the observation pertains
- DATA, the original data value
- TRANDATA, the optimally transformed data value when LEVEL=ORDINAL
- DISTANCE, the distance computed from the PROC MDS model
- TRANSDIST, the optimally transformed distance when the LEVEL= option is not ORDINAL or ABSOLUTE
- FITDATA, the data value further transformed according to the FIT= option
- FITDIST, the distance further transformed according to the FIT= option
- WEIGHT, the combined weight of the data value based on the WEIGHT variable(s), if any, and the standardization specified by the FORMULA= option
- RESIDUAL, FITDATA minus FITDIST

If you assign a nonmissing data value a weight of zero, PROC MDS will ignore it when the model is fit, but the value will still appear in the OUTRES= data set (see the section "WEIGHT Statement" on page 3723).

INITIAL= Data Set

The INITIAL= data set has the same structure as the OUT= data set but is not required to have all of the variables or observations that appear in the OUT= data set. You can use an OUT= data set previously created by PROC MDS (without the OUTITER option) as an INITIAL= data set in a subsequent invocation of the procedure.

The only variables that are required are Dim1, ..., Dim*m* (where *m* is the maximum number of dimensions) or equivalent variables specified in the INVAR statement. If these are the only variables, then all the observations are assumed to contain coordinates of the configuration; you cannot read dimension coefficients or transformation parameters.

To read initial values for the dimension coefficients or transformation parameters, the INITIAL= data set must contain the _TYPE_ variable and either the variable specified in the ID statement or, if no ID statement is used, the variable _NAME_. In addition, if there is more than one data matrix, either the variable specified in the MATRIX statement or, if no MATRIX statement is used, the variable _MATRIX_ or _MATNUM_ is required.

If the INITIAL= data set contains the variable _DIMENS_, initial values are obtained from observations with the corresponding number of dimensions. If there is no _DIMENS_ variable, the same observations are used for each number of dimensions analyzed.

If you want PROC MDS to read initial values from some but not all of the observations in the INITIAL= data set, use the WHERE= data set option to select the desired observations.

Missing Values

Missing data in the similarity or dissimilarity matrices are ignored in fitting the model and are omitted from the OUTRES= data set. Any matrix that is completely missing is omitted from the analysis.

Missing weights are treated as 0.

Missing values are also permitted in the INITIAL= data set, but a large number of missing values might yield a degenerate initial configuration.

Normalization of the Estimates

In multidimensional scaling models, the parameter estimates are not uniquely determined; the estimates can be transformed in various ways without changing their badness of fit. The initial and final estimates from PROC MDS are, therefore, normalized (unless you specify the NONORM option) to make it easier to compare results from different analyses.

The configuration always has a mean of 0 for each dimension.

With the COEF=IDENTITY option, the configuration is rotated to a principal-axis orientation. Unless you specify the LEVEL=ABSOLUTE option, the entire configuration is scaled so that the root-mean-square element is 1, and the transformations are adjusted to compensate.

With the COEF=DIAGONAL option, each dimension is scaled to a root-mean-square value of 1, and the dimension coefficients are adjusted to compensate. Unless you specify the LEVEL=ABSOLUTE option, the dimension coefficients are normalized as follows. If you specify the CONDITION=UN option, all of the dimension coefficients are scaled to a root-mean-square value of 1. For other values of the CONDITION= option, the dimension coefficients are scaled separately for each subject to a root-mean-square value of 1. In either case, the transformations are adjusted to compensate.

Each dimension is reflected to give a positive rank correlation with the order of the objects in the data set.

For the LEVEL=ORDINAL option, if the intercept, slope, or power parameters are fitted, the transformed data are normalized to eliminate these parameters if possible.

Comparison with Earlier Procedures

PROC MDS shares many of the features of the ALSCAL procedure (Young, Lewyckyj, and Takane 1986; Young 1982), as well as some features of the MLSCALE procedure (Ramsay 1986). Both PROC ALSCAL and PROC MLSCALE are no longer a part of SAS; however, they are described in the *SUGI Supplemental Library User's Guide, Version 5 Edition*. The MDS procedure generally produces results similar to those from the ALSCAL procedure (Young, Lewyckyj, and Takane 1986; Young 1982) if you use the following options in PROC MDS:

- FIT=SQUARED
- FORMULA=1 except for unfolding data, which require FORMULA=2
- PFINAL to get output similar to that from PROC ALSCAL

Running the MDS procedure with certain options generally produces results similar to those from using the MLSCALE procedure (Ramsay 1986) with other options. This is illustrated with the following statements:

```
proc mds fit=log level=loginterval ... ;

proc mlscale stvarnce=constant suvarnce=constant ... ;
```

Alternatively, using the FIT=DISTANCE option in the PROC MDS statement produces results similar to those from specifying the NORMAL option in the PROC MLSCALE statement.

Displayed Output

Unless you specify the NOPHIST option, PROC MDS displays the iteration history containing the following:

- Iteration number
- Type of iteration:

Initial	initial configuration
Monotone	monotone transformation
Gau-New	Gauss-Newton step
Lev-Mar	Levenberg-Marquardt step

- Badness-of-Fit Criterion
- Change in Criterion
- Convergence Measures:

Monotone	the Euclidean norm of the change in the optimally scaled data divided by the Euclidean norm of the optimally scaled data, averaged across partitions
Gradient	the multiple correlation of the Jacobian matrix with the residual vector, uncorrected for the mean

Depending on what options are specified, PROC MDS can also display the following tables:

- Data Matrix and possibly Weight Matrix for each subject
- Eigenvalues from the computation of the initial coordinates
- Sum of Data Weights and Pooled Data Matrix computed during initialization with INAV=DATA
- Configuration, the estimated coordinates of the objects
- Dimension Coefficients
- A table of transformation parameters, including one or more of the following:

  Intercept

  Slope

  Power

- A table of fit statistics for each matrix and possibly each row, including the following:

  Number of Nonmissing Data

Weight of the matrix or row, permitting both observation weights and standardization factors

Badness-of-Fit Criterion

Distance Correlation computed between the distances and data with optimal transformation

Uncorrected Distance Correlation not corrected for the mean

Fit Correlation computed after applying the FIT= transformation to both distances and data

Uncorrected Fit Correlation not corrected for the mean

ODS Table Names

PROC MDS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 53.3. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 53.3 ODS Tables Produced by PROC MDS

ODS Table Name	Description	Option
ConvergenceStatus	Convergence status	default
DimensionCoef	Dimension coefficients	PCOEF w/COEF= not IDENTITY
FitMeasures	Measures of fit	PFIT
IterHistory	Iteration history	default
PConfig	Configuration of coordinates	PCONFIG
PData	Data matrices	PDATA
PInAvData	INAV= data set information	PINAVDATA
PInEigval	Initial eigenvalues	PINEIGVAL
PInEigvec	Initial eigenvectors	PINEIGVEC
PInWeight	Initialization weights	PINWEIGHT
Transformations	Transformation parameters	PTRANS w/LEVEL=RATIO, INTERVAL, LOGINTERVAL

ODS Graphics

To request graphics with PROC MDS, you must first enable ODS Graphics by specifying the ODS GRAPHICS ON statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. All graphs are produced by default when they are appropriate. You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC MDS generates are listed in Table 53.4, along with the required options.

Table 53.4 ODS Graphics Produced by PROC MDS

ODS Graph Name	Plot Description	Option
BadnessPlot	Badness of fit	DIMENSION=*range*
CoefficientsPlot	Individual coefficients	DIMENSION=n, $n > 1$ COEF=DIAGONAL
ConfigPlot	Configuration	DIMENSION=n, $n > 1$
FitPlot	Fit	default

Example: MDS Procedure

Example 53.1: Jacobowitz Body Parts Data from Children and Adults

Jacobowitz (1975) collected conditional rank-order data regarding perceived similarity of parts of the body from children of ages 6, 8, and 10 years and from college sophomores. The data set includes data from 15 children (6-year-olds) and 15 sophomores. The method of data collection and some results of an analysis are also described by Young (1987, pp. 4–10). The following statements create the input data set:

```
data body;
   title  'Jacobowitz Body Parts Data from 6-Year-Olds and Adults';
   title2 'First 15 Subjects (obs 1-225) Are Children';
   title3 'Second 15 Subjects (obs 226-450) Are Adults';
   input  (Cheek Face Mouth Head Ear Body Arm Elbow Hand
           Palm Finger Leg Knee Foot Toe) (2.);
   if _n_ <= 225 then Subject='C'; else subject='A';
   datalines;
 0  2  1  3  4 10  5  9  6  7  8 11 12 13 14
 2  0 12  1 13  3  8 10 11  9  7  4  5  6 14
 3  2  0  1  4  9  5 11  6  7  8 10 13 12 14

... more lines ...

10 12 11 13  9 14  8  7  4  6  2  3  5  1  0
;
```

The data are analyzed as row conditional (CONDITION=ROW) at the ordinal level of measurement (LEVEL=ORDINAL) by using the weighted Euclidean model (COEF=DIAGONAL) in three dimensions (DIMENSION=3). The final estimates are displayed (PFINAL). The estimates (OUT=OUT) and fitted values (OUTRES=RES) are saved in output data sets. The following statements produce Output 53.1.1:

```
ods graphics on;
proc mds data=body condition=row level=ordinal coef=diagonal
         dimension=3 pfinal out=out outres=res;
   subject subject;
   title5 'Nonmetric Weighted MDS';
run;
ods graphics off;
```

Output 53.1.1 Analysis of Body Parts Data

```
          Jacobowitz Body Parts Data from 6-Year-Olds and Adults
               First 15 Subjects (obs 1-225) Are Children
              Second 15 Subjects (obs 226-450) Are Adults

                          Nonmetric Weighted MDS

              Multidimensional Scaling:  Data=WORK.BODY.DATA
                 Shape=SQUARE Condition=ROW Level=ORDINAL
                 Coef=DIAGONAL Dimension=3 Formula=1 Fit=1

Mconverge=0.01 Gconverge=0.01 Maxiter=100 Over=2 Ridge=0.0001 Alternate=MATRIX
```

Iteration	Type	Badness-of-Fit Criterion	Change in Criterion	Convergence Measures: Monotone	Convergence Measures: Gradient
0	Initial	0.5938	.	.	.
1	Monotone	0.2344	0.3594	0.4693	0.4028
2	Gau-New	0.2080	0.0264	.	.
3	Monotone	0.1963	0.0118	0.0556	0.2630
4	Gau-New	0.1927	0.003592	.	.
5	Monotone	0.1797	0.0130	0.0463	0.1544
6	Gau-New	0.1779	0.001809	.	.
7	Monotone	0.1744	0.003430	0.0225	0.1210
8	Gau-New	0.1736	0.000807	.	.
9	Monotone	0.1717	0.001929	0.0161	0.1128
10	Gau-New	0.1712	0.000474	.	.
11	Monotone	0.1698	0.001413	0.0135	0.1119
12	Gau-New	0.1696	0.000188	.	.
13	Monotone	0.1684	0.001261	0.0121	0.1121
14	Gau-New	0.1683	0.000117	.	.
15	Monotone	0.1672	0.001096	0.0111	0.1064
16	Gau-New	0.1670	0.000131	.	.
17	Monotone	0.1661	0.000902	0.0103	0.0965
18	Gau-New	0.1660	0.000160	.	.
19	Monotone	0.1652	0.000736	0.009740	0.0980
20	Gau-New	0.1651	0.000169	.	0.1062
21	Gau-New	0.1645	0.000542	.	0.0161
22	Gau-New	0.1645	4.2645E-6	.	0.009969

```
      Convergence criteria are satisfied.
```

Output 53.1.1 *continued*

Configuration

	Dim1	Dim2	Dim3
Cheek	0.25	1.47	2.06
Face	0.36	0.32	0.33
Mouth	0.78	0.08	1.08
Head	2.10	1.07	−0.01
Ear	0.26	0.72	−0.34
Body	1.51	−0.61	−0.68
Arm	−0.74	2.07	−0.59
Elbow	−0.51	−0.21	0.01
Hand	0.46	−1.50	−0.60
Palm	−1.44	−0.81	1.48
Finger	−0.24	−0.24	−0.81
Leg	−1.68	0.26	−0.05
Knee	−1.19	−1.19	−1.36
Foot	0.16	−0.03	−1.56
Toe	−0.10	−1.39	1.02

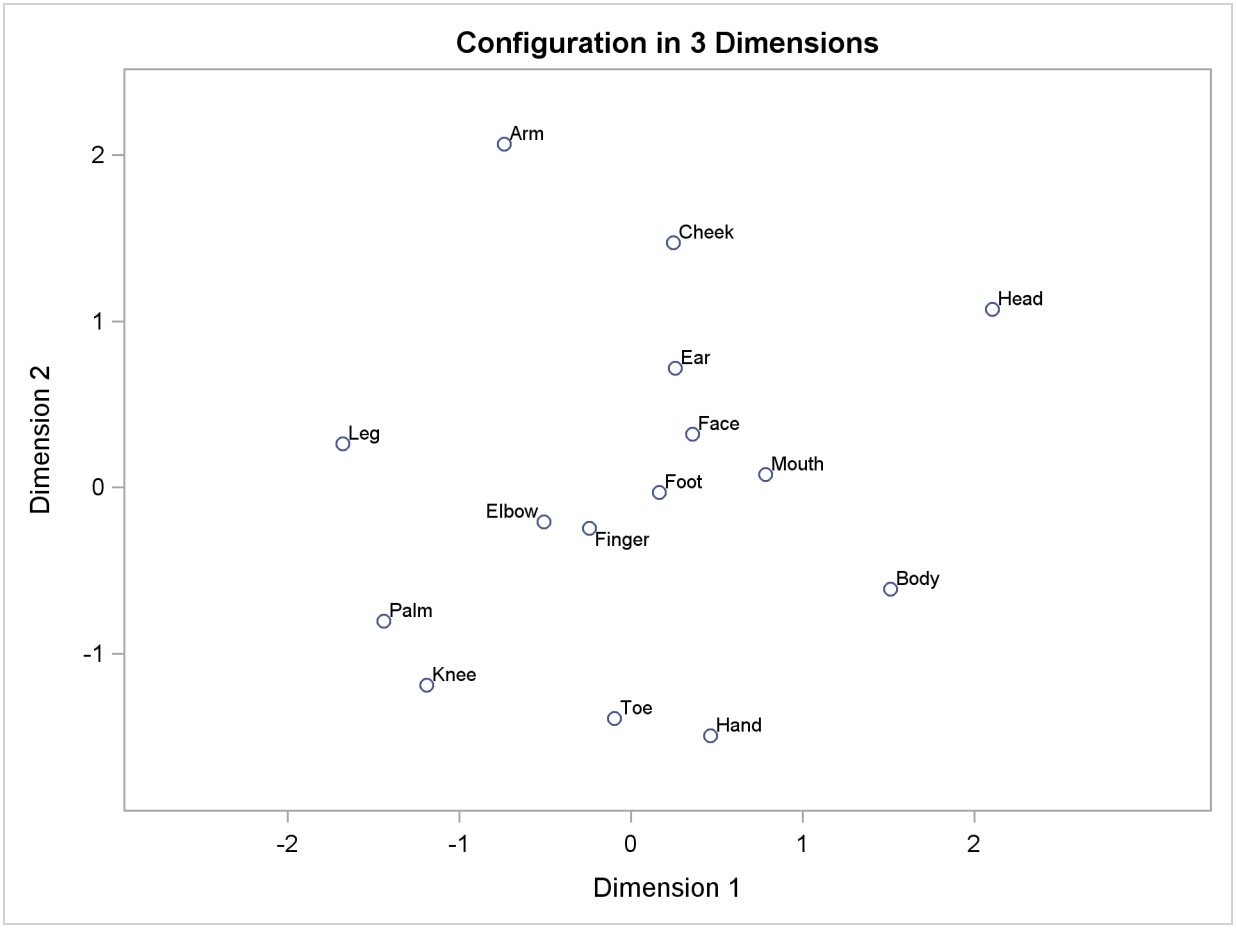
Dimension Coefficients

Subject	1	2	3
C	1.00	1.12	0.86
C	0.96	1.02	1.01
C	0.98	1.05	0.98
C	1.02	1.08	0.89
C	0.95	1.04	1.01
C	0.99	1.12	0.89
C	1.07	1.00	0.93
C	1.04	1.02	0.94
C	0.99	1.15	0.83
C	0.89	1.11	0.99
C	1.04	1.03	0.92
C	1.06	1.01	0.93
C	0.92	1.24	0.78
C	0.97	0.98	1.05
C	1.03	1.00	0.97
A	0.93	1.17	0.88
A	0.89	1.12	0.97
A	0.88	1.17	0.94
A	0.81	1.14	1.02
A	0.90	1.11	0.98
A	0.90	1.17	0.91
A	0.92	1.17	0.88
A	0.97	1.19	0.80
A	0.95	1.16	0.87
A	1.08	1.07	0.83
A	0.95	1.20	0.81
A	1.00	0.97	1.02
A	0.89	1.18	0.91
A	0.97	1.15	0.86
A	0.93	1.21	0.82

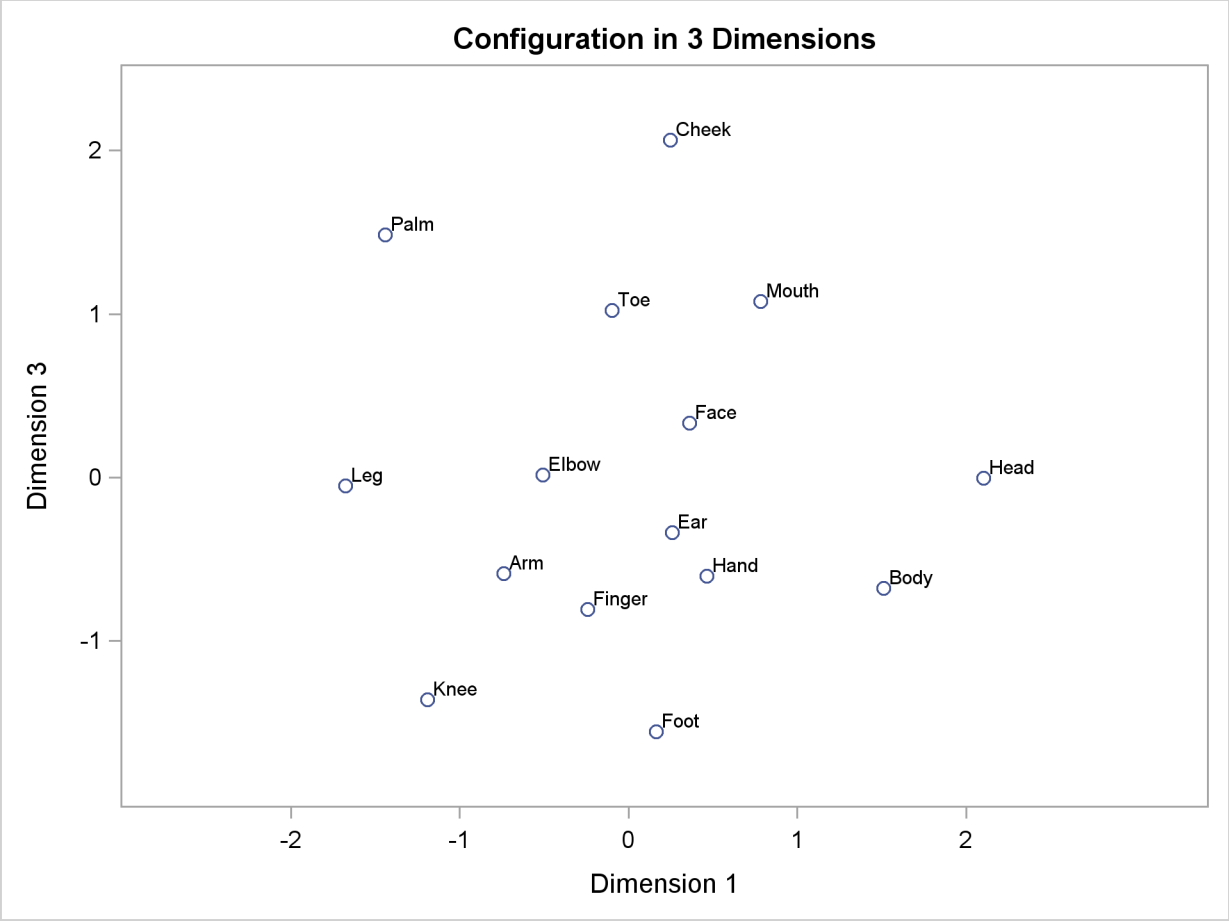
Output 53.1.1 *continued*

Subject	Number of Nonmissing Data	Weight	Badness-of-Fit Criterion	Distance Correlation	Uncorrected Distance Correlation
C	160	0.03	0.15	0.86	0.99
C	163	0.03	0.19	0.78	0.98
C	166	0.03	0.20	0.79	0.98
C	158	0.03	0.16	0.84	0.99
C	173	0.03	0.18	0.83	0.98
C	164	0.03	0.14	0.90	0.99
C	158	0.03	0.20	0.77	0.98
C	170	0.03	0.18	0.83	0.98
C	156	0.03	0.15	0.88	0.99
C	165	0.03	0.18	0.79	0.98
C	153	0.03	0.19	0.79	0.98
C	162	0.03	0.17	0.83	0.98
C	161	0.03	0.14	0.90	0.99
C	164	0.03	0.17	0.83	0.99
C	161	0.03	0.18	0.81	0.98
A	163	0.03	0.15	0.87	0.99
A	174	0.04	0.17	0.85	0.99
A	172	0.03	0.15	0.89	0.99
A	175	0.04	0.17	0.85	0.98
A	171	0.03	0.15	0.87	0.99
A	163	0.03	0.16	0.86	0.99
A	173	0.03	0.14	0.90	0.99
A	160	0.03	0.14	0.89	0.99
A	164	0.03	0.14	0.90	0.99
A	158	0.03	0.16	0.86	0.99
A	165	0.03	0.16	0.87	0.99
A	168	0.03	0.18	0.82	0.98
A	175	0.04	0.15	0.89	0.99
A	172	0.03	0.16	0.88	0.99
A	175	0.04	0.15	0.90	0.99
- All -	4962	1.00	0.16	0.85	0.99

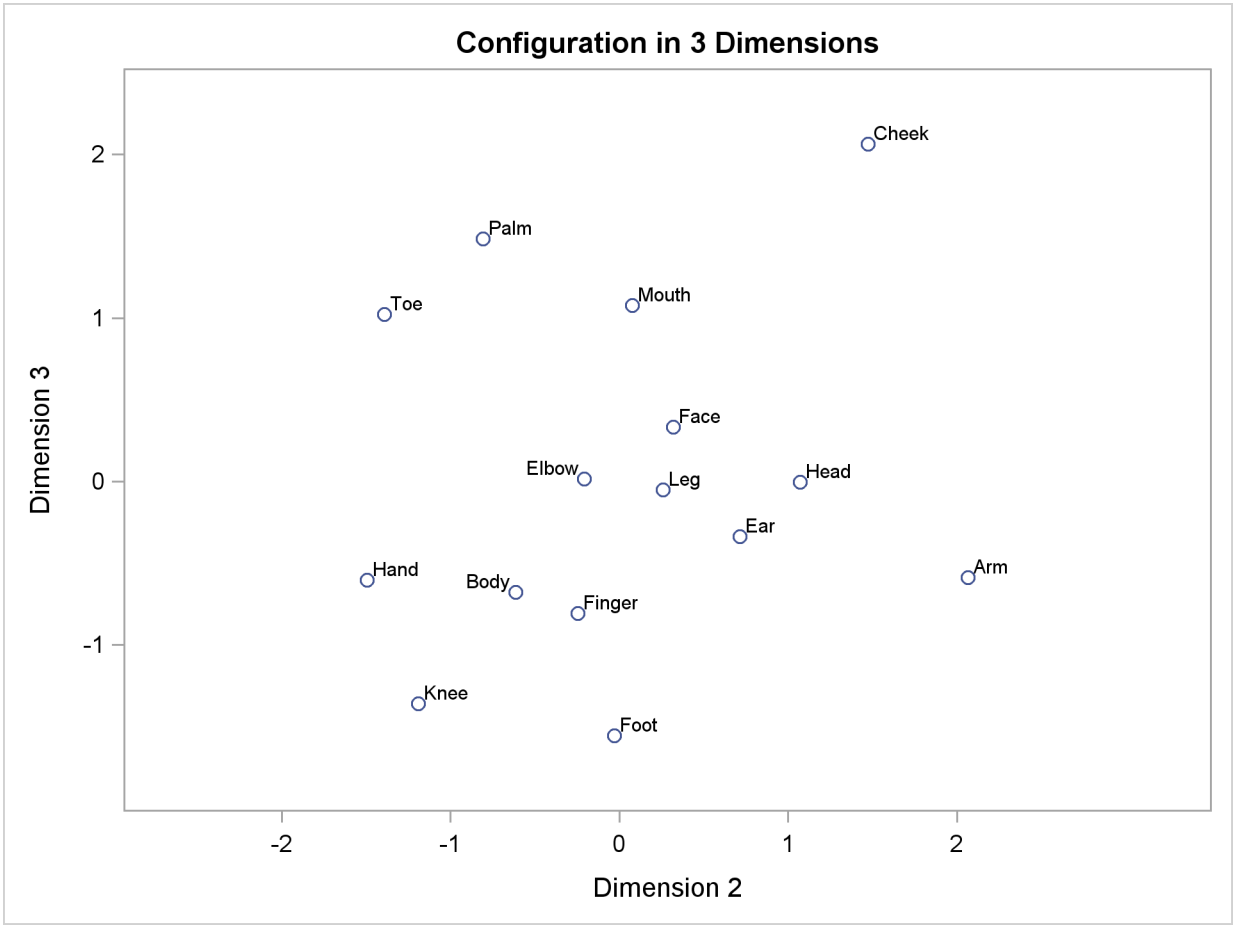
Output 53.1.1 *continued*

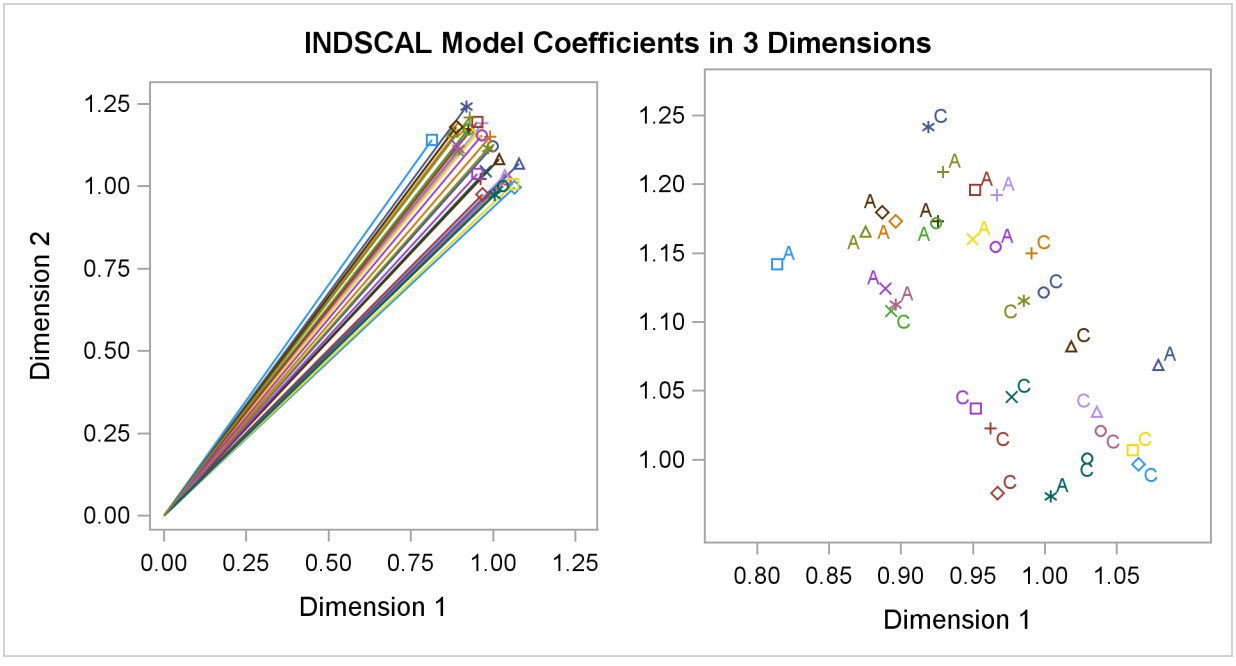


Output 53.1.1 *continued*

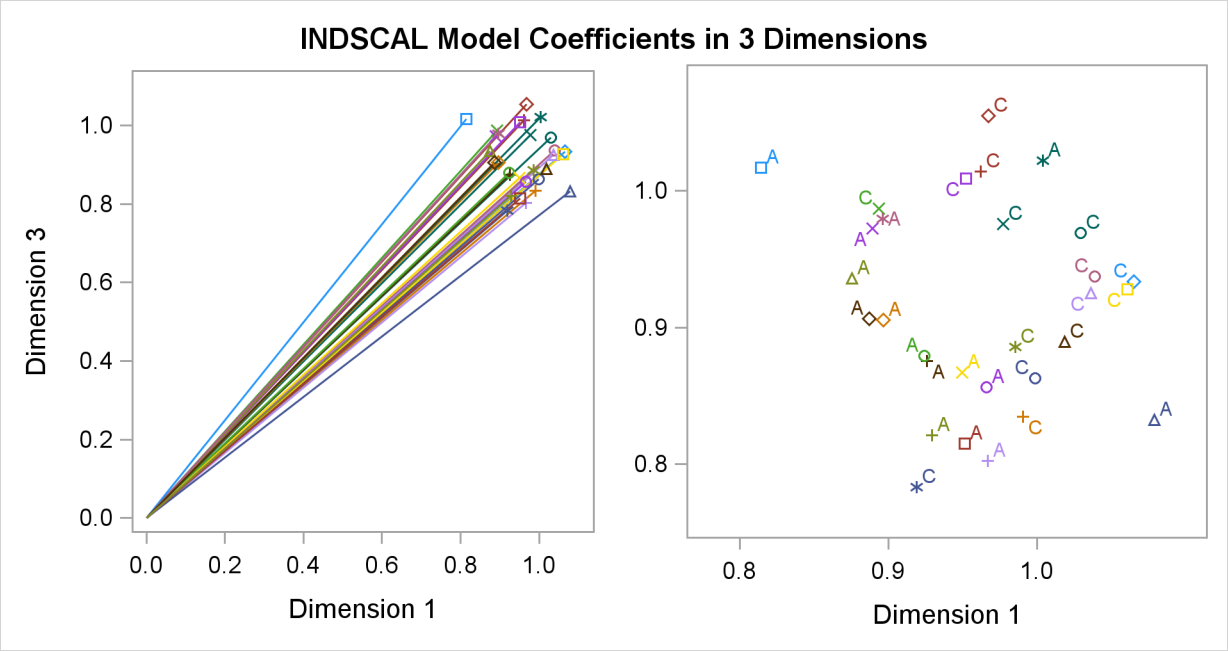


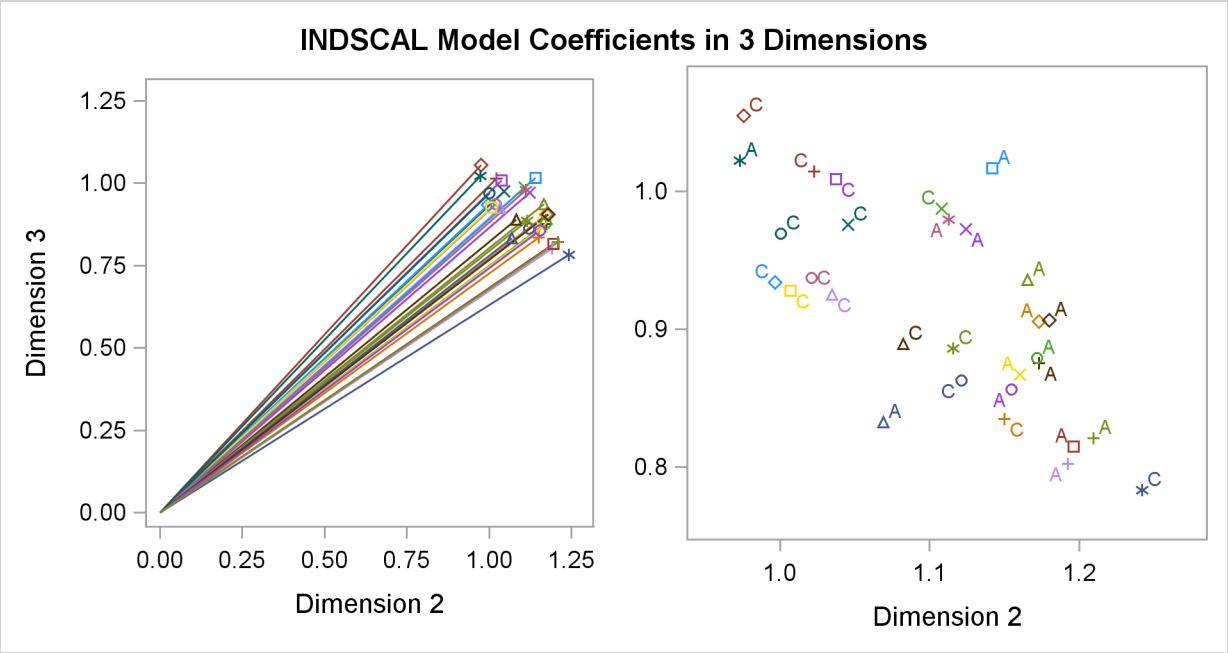
Output 53.1.1 *continued*



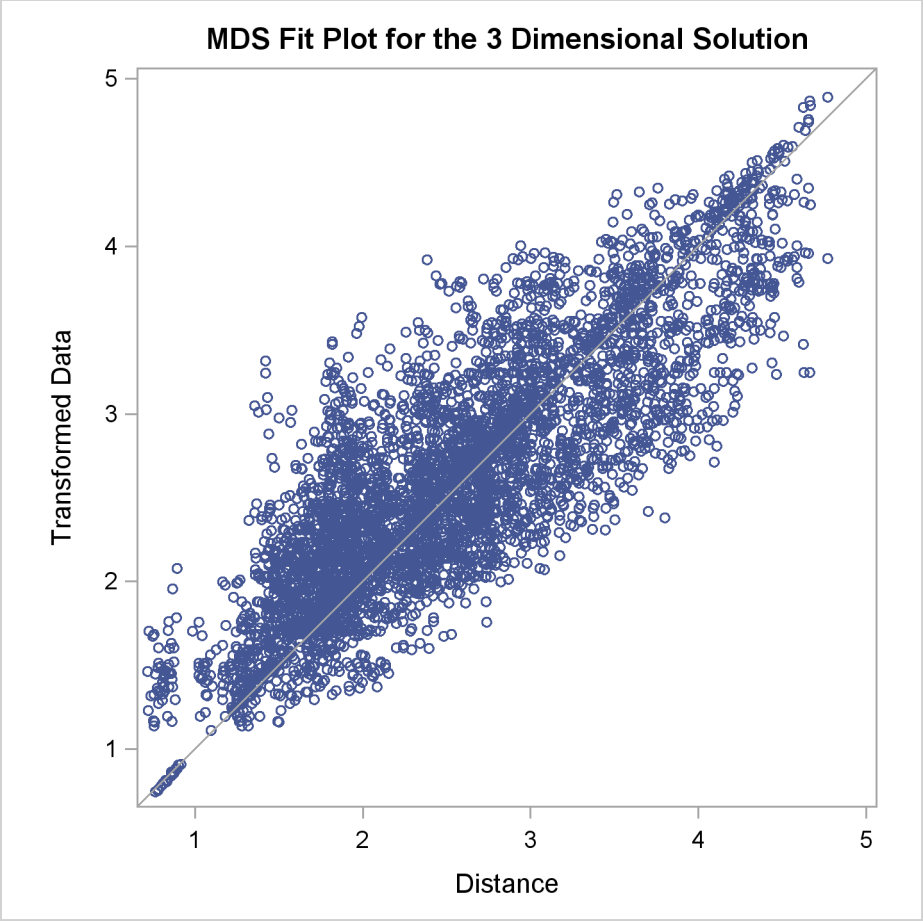


Output 53.1.1 *continued*





Output 53.1.1 *continued*



Often the output of greatest interest in an MDS analysis is the graphical output. The first plots show two-dimensional view of the three-dimensional configuration. Next, the coefficients are plotted. The last plot is the fit plot.

In the fit plot, the transformed data are plotted on the vertical axis, and the distances from the model are plotted on the horizontal axis. If the model fits perfectly, all points lie on a diagonal line from lower left to upper right. The vertical departure of each point from this diagonal line represents the residual of the corresponding observation.

The configuration has a tripodal shape with Body at the apex. The three legs of the tripod can be distinguished in the plot of dimension 2 by dimension 1, which shows three distinct clusters with Body in the center. Dimension 1 separates head parts from arm and leg parts. Dimension 2 separates arm parts from leg parts. The plot of dimension 3 by dimension 1 shows the tripod from the side. Dimension 3 distinguishes the more inclusive body parts (at the top) from the less inclusive body parts (at the bottom).

The plots of dimension coefficients show that children differ from adults primarily in the emphasis given to dimension 2. Children give about the same weight (approximately 1) to each dimension. Adults are much more variable than children, but all have coefficients less than 1.0 for dimension 2, with an average of about 0.7. Referring back to the configuration plot, you can see that adults

consider arm parts to be more similar to leg parts than children do. Many adults also give a high weight to dimension 1, indicating that they consider head parts to be more dissimilar from arm and leg parts than children do. Dimension 3 shows considerable variability for both children and adults.

References

Arabie, P., Carroll, J. D., and DeSarbo, W. S. (1987), *Three-Way Scaling and Clustering*, Sage University Paper Series on Quantitative Applications in the Social Sciences, 07-065, Beverly Hills and London: Sage Publications.

Carroll, J. D. and Chang, J. J. (1970), "Analysis of Individual Differences in Multidimensional Scaling via an N-way Generalization of the 'Eckart-Young' Decomposition," *Psychometrika*, 35, 283–319.

Heiser, W. J. (1981), *Unfolding Analysis of Proximity Data*, Leiden: Department of Data Theory, University of Leiden.

Jacobowitz, D. (1975), *The Acquisition of Semantic Structures*, Ph.D. dissertation, University of North Carolina at Chapel Hill.

Krantz, D. H., Luce, R. D., Suppes, P., and Tversky, A. (1971), *Foundations of Measurement*, New York: Academic Press.

Kruskal, J. B. and Wish, M. (1978), *Multidimensional Scaling*, Sage University Paper Series on Quantitative Applications in the Social Sciences, 07-011, Beverly Hills and London: Sage Publications.

Null, C. H. and Sarle, W. S. (1982), "Multidimensional Scaling by Least Squares," in *Proceedings of the Seventh Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Rabinowitz, G. (1976), "A Procedure for Ordering Object Pairs Consistent with the Multidimensional Unfolding Model," *Psychometrika*, 41, 349–373.

Ramsay, J. O. (1986), "The MLSCALE Procedure," in *SUGI Supplemental Library User's Guide, Version 5 Edition*, Cary, NC: SAS Institute Inc.

Schiffman, S. S., Reynolds, M. L., and Young, F. W. (1981), *Introduction to Multidimensional Scaling*, New York: Academic Press.

Torgerson, W. S. (1958), *Theory and Methods of Scaling*, New York: John Wiley & Sons.

Young, F. W. (1982), "Enhancements in ALSCAL-82," in *Proceedings of the Seventh Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Young, F. W. (1987), *Multidimensional Scaling: History, Theory, and Applications*, ed. R. M. Hamer, Hillsdale, NJ: Lawrence Erlbaum Associates.

Young, F. W., Lewyckyj, R., and Takane, Y. (1986), "The ALSCAL Procedure," in *SUGI Supplemental Library User's Guide, Version 5 Edition*, Cary, NC: SAS Institute Inc.

Chapter 54

The MI Procedure

Contents

Overview: MI Procedure **3744**
Getting Started: MI Procedure **3746**
Syntax: MI Procedure **3750**
PROC MI Statement 3751
BY Statement 3754
CLASS Statement 3754
EM Statement 3755
FREQ Statement 3756
MCMC Statement 3756
MONOTONE Statement 3764
TRANSFORM Statement 3767
VAR Statement 3768
Details: MI Procedure **3768**
Descriptive Statistics 3768
EM Algorithm for Data with Missing Values 3769
Statistical Assumptions for Multiple Imputation 3771
Missing Data Patterns 3772
Imputation Methods 3773
Regression Method for Monotone Missing Data 3774
Predictive Mean Matching Method for Monotone Missing Data 3775
Propensity Score Method for Monotone Missing Data 3776
Discriminant Function Method for Monotone Missing Data 3777
Logistic Regression Method for Monotone Missing Data 3779
MCMC Method for Arbitrary Missing Data 3780
Producing Monotone Missingness with the MCMC Method 3785
MCMC Method Specifications 3786
Checking Convergence in MCMC 3788
Input Data Sets 3791
Output Data Sets 3792
Combining Inferences from Multiply Imputed Data Sets 3794
Multiple Imputation Efficiency 3795
Imputer's Model Versus Analyst's Model 3796
Parameter Simulation versus Multiple Imputation 3797
Summary of Issues in Multiple Imputation 3798

ODS Table Names 3800
ODS Graphics 3800
Examples: MI Procedure **3801**
Example 54.1: EM Algorithm for MLE 3802
Example 54.2: Propensity Score Method 3806
Example 54.3: Regression Method 3809
Example 54.4: Logistic Regression Method for CLASS Variables 3812
Example 54.5: Discriminant Function Method for CLASS Variables 3815
Example 54.6: MCMC Method 3817
Example 54.7: Producing Monotone Missingness with MCMC 3820
Example 54.8: Checking Convergence in MCMC 3823
Example 54.9: Saving and Using Parameters for MCMC 3827
Example 54.10: Transforming to Normality 3829
Example 54.11: Multistage Imputation 3832
References **3835**

Overview: MI Procedure

The MI procedure performs multiple imputation of missing data. Missing values are an issue in a substantial number of statistical analyses. Most SAS statistical procedures exclude observations with any missing variable values from the analysis. These observations are called incomplete cases. Although analyzing only complete cases has the advantage of simplicity, the information contained in the incomplete cases is lost. This approach also ignores possible systematic differences between the complete cases and the incomplete cases, and the resulting inference might not be applicable to the population of all cases, especially with a small number of complete cases.

Some SAS procedures use all the available cases in an analysis—that is, cases with useful information. For example, the CORR procedure estimates a variable mean by using all cases with nonmissing values for this variable, ignoring the possible missing values in other variables. PROC CORR also estimates a correlation by using all cases with nonmissing values for this pair of variables. This makes better use of the available data than using only the complete cases does, but the resulting correlation matrix might not be positive definite.

Another strategy for handling missing data is single imputation, which substitutes a value for each missing value. Standard statistical procedures for complete data analysis can then be used with the filled-in data set. For example, each missing value can be imputed with the variable mean of the complete cases, or it can be imputed with the mean conditional on observed values of other variables. This approach treats missing values as if they were known in the complete-data analysis. However, single imputation does not reflect the uncertainty about the predictions of the unknown missing values, and the resulting estimated variances of the parameter estimates will be biased toward zero (Rubin 1987, p. 13).

Instead of filling in a single value for each missing value, multiple imputation (Rubin 1976, 1987) replaces each missing value with a set of plausible values that represent the uncertainty about the

right value to impute. The multiply imputed data sets are then analyzed by using standard procedures for complete data and combining the results from these analyses. No matter which complete-data analysis is used, the process of combining results from different data sets is essentially the same.

Multiple imputation does not attempt to estimate each missing value through simulated values. Instead, it draws a random sample of the missing values from its distribution. This process results in valid statistical inferences that properly reflect the uncertainty due to missing values—for example, confidence intervals with the correct probability coverage.

Multiple imputation inference involves three distinct phases:

1. The missing data are filled in m times to generate m complete data sets.
2. The m complete data sets are analyzed using standard statistical analyses.
3. The results from the m complete data sets are combined to produce inferential results.

The MI procedure creates multiply imputed data sets for incomplete multivariate data. It uses methods that incorporate appropriate variability across the m imputations. The method of choice depends on the patterns of missingness.

A data set with variables $Y_1, Y_2, \ldots, Y_p$ (in that order) is said to have a *monotone missing pattern* when the event that a variable Y_j is missing for a particular individual implies that all subsequent variables $Y_k, k > j$, are missing for that individual.

For data sets with monotone missing patterns, either a parametric method that assumes multivariate normality or a nonparametric method is appropriate to impute missing values for a continuous variable. Parametric methods available include the regression method (Rubin 1987, pp. 166–167) and the predictive mean matching method (Heitjan and Little 1991; Schenker and Taylor 1996). The nonparametric method is the propensity score method (Rubin 1987, pp. 124, 158).

To impute missing values for a classification variable in data sets with monotone missing patterns, you can use the logistic regression method when the classification variable has a binary or ordinal response, and the discriminant function method when the classification variable has a binary or nominal response.

For data sets with arbitrary missing patterns, a Markov chain Monte Carlo (MCMC) method (Schafer 1997) that assumes multivariate normality is used to impute all missing values or just enough missing values for continuous variables to make the imputed data sets have monotone missing patterns. When an imputed data set has a monotone missing pattern, methods for data sets with monotone missing patterns can then be used to impute remaining missing values.

Once the m complete data sets are analyzed using standard SAS procedures, the MIANALYZE procedure can be used to generate valid statistical inferences about these parameters by combining results from the m analyses.

Often, as few as three to five imputations are adequate in multiple imputation (Rubin 1996, p. 480). The relative efficiency of the small m imputation estimator is high for cases with little missing information (Rubin 1987, p. 114). (Also see the section "Multiple Imputation Efficiency" on page 3795.)

Multiple imputation inference assumes that the model (variables) you used to analyze the multiply imputed data (the analyst's model) is the same as the model used to impute missing values in multiple imputation (the imputer's model). But in practice, the two models might not be the same. The consequences for different scenarios (Schafer 1997, pp. 139–143) are discussed in the section "Imputer's Model Versus Analyst's Model" on page 3796.

When an MCMC method is used to used to impute missing values, the trace (time series) and autocorrelation function plots for parameters such as variable means and covariances can be displayed to check for convergence of the MCMC method. See the section "Checking Convergence in MCMC" on page 3788 for a detailed description of these plots. If the **ods graphics on** statement is specified, these statistical graphics are created via the Output Delivery System (ODS). Otherwise, the traditional graphics are created.

Getting Started: MI Procedure

Consider the following Fitness data set that has been altered to contain an arbitrary pattern of missingness:

```
*----------------- Data on Physical Fitness -----------------*
| These measurements were made on men involved in a physical |
| fitness course at N.C. State University.                   |
| Only selected variables of                                 |
| Oxygen (oxygen intake, ml per kg body weight per minute),  |
| Runtime (time to run 1.5 miles in minutes), and            |
| RunPulse (heart rate while running) are used.              |
| Certain values were changed to missing for the analysis.   |
*------------------------------------------------------------*;
data FitMiss;
   input Oxygen RunTime RunPulse @@;
   datalines;
44.609  11.37  178     45.313  10.07  185
54.297   8.65  156     59.571    .      .
49.874   9.22    .     44.811  11.63  176
  .     11.95  176          .  10.85    .
39.442  13.08  174     60.055   8.63  170
50.541    .      .     37.388  14.03  186
44.754  11.12  176     47.273    .      .
51.855  10.33  166     49.156   8.95  180
40.836  10.95  168     46.672  10.00    .
46.774  10.25    .     50.388  10.08  168
39.407  12.63  174     46.080  11.17  156
45.441   9.63  164       .      8.92    .
45.118  11.08    .     39.203  12.88  168
45.790  10.47  186     50.545   9.93  148
48.673   9.40  186     47.920  11.50  170
47.467  10.50  170
;
```

Suppose that the data are multivariate normally distributed and the missing data are missing at random (MAR). That is, the probability that an observation is missing can depend on the observed variable values of the individual, but not on the missing variable values of the individual. See the section "Statistical Assumptions for Multiple Imputation" on page 3771 for a detailed description of the MAR assumption.

The following statements invoke the MI procedure and impute missing values for the FitMiss data set:

```
proc mi data=FitMiss seed=501213 mu0=50 10 180 out=outmi;
   var Oxygen RunTime RunPulse;
run;
```

The "Model Information" table in Figure 54.1 describes the method used in the multiple imputation process. By default, the procedure uses the Markov chain Monte Carlo (MCMC) method with a single chain to create five imputations. The posterior mode, the highest observed-data posterior density, with a noninformative prior, is computed from the expectation-maximization (EM) algorithm and is used as the starting value for the chain.

Figure 54.1 Model Information

The MI Procedure

Model Information

Data Set	WORK.FITMISS
Method	MCMC
Multiple Imputation Chain	Single Chain
Initial Estimates for MCMC	EM Posterior Mode
Start	Starting Value
Prior	Jeffreys
Number of Imputations	5
Number of Burn-in Iterations	200
Number of Iterations	100
Seed for random number generator	501213

The MI procedure takes 200 burn-in iterations before the first imputation and 100 iterations between imputations. In a Markov chain, the information in the current iteration influences the state of the next iteration. The burn-in iterations are iterations in the beginning of each chain that are used both to eliminate the series of dependence on the starting value of the chain and to achieve the stationary distribution. The between-imputation iterations in a single chain are used to eliminate the series of dependence between the two imputations.

The "Missing Data Patterns" table in Figure 54.2 lists distinct missing data patterns with corresponding frequencies and percents. Here, an "X" means that the variable is observed in the corresponding group and a "." means that the variable is missing. The table also displays group-specific variable means. The MI procedure sorts the data into groups based on whether the analysis variables are observed or missing. For a detailed description of missing data patterns, see the section "Missing Data Patterns" on page 3772.

Figure 54.2 Missing Data Patterns

Missing Data Patterns

Group	Oxygen	Run Time	Run Pulse	Freq	Percent
1	X	X	X	21	67.74
2	X	X	.	4	12.90
3	X	.	.	3	9.68
4	.	X	X	1	3.23
5	.	X	.	2	6.45

Missing Data Patterns

	Group Means		
Group	Oxygen	RunTime	RunPulse
1	46.353810	10.809524	171.666667
2	47.109500	10.137500	.
3	52.461667	.	.
4	.	11.950000	176.000000
5	.	9.885000	.

After the completion of m imputations, the "Variance Information" table in Figure 54.3 displays the between-imputation variance, within-imputation variance, and total variance for combining complete-data inferences. It also displays the degrees of freedom for the total variance. The relative increase in variance due to missing values, the fraction of missing information, and the relative efficiency (in units of variance) for each variable are also displayed. A detailed description of these statistics is provided in the section "Combining Inferences from Multiply Imputed Data Sets" on page 3794.

Figure 54.3 Variance Information

Variance Information

	Variance			
Variable	Between	Within	Total	DF
Oxygen	0.056930	0.954041	1.022356	25.549
RunTime	0.000811	0.064496	0.065469	27.721
RunPulse	0.922032	3.269089	4.375528	15.753

Variance Information

Variable	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Oxygen	0.071606	0.068898	0.986408
RunTime	0.015084	0.014968	0.997015
RunPulse	0.338455	0.275664	0.947748

The "Parameter Estimates" table in Figure 54.4 displays the estimated mean and standard error of the mean for each variable. The inferences are based on the *t* distribution. The table also displays a 95% confidence interval for the mean and a *t* statistic with the associated *p*-value for the hypothesis that the population mean is equal to the value specified with the MU0= option. A detailed description of these statistics is provided in the section "Combining Inferences from Multiply Imputed Data Sets" on page 3794.

Figure 54.4 Parameter Estimates

Parameter Estimates

Variable	Mean	Std Error	95% Confidence Limits		DF
Oxygen	47.094040	1.011116	45.0139	49.1742	25.549
RunTime	10.572073	0.255870	10.0477	11.0964	27.721
RunPulse	171.787793	2.091776	167.3478	176.2278	15.753

Parameter Estimates

Variable	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
Oxygen	46.783898	47.395550	50.000000	-2.87	0.0081
RunTime	10.526392	10.599616	10.000000	2.24	0.0336
RunPulse	170.774818	173.122002	180.000000	-3.93	0.0012

In addition to the output tables, the procedure also creates a data set with imputed values. The imputed data sets are stored in the outmi data set, with the index variable _Imputation_ indicating the imputation numbers. The data set can now be analyzed using standard statistical procedures with _Imputation_ as a BY variable.

The following statements list the first 10 observations of data set outmi:

```
proc print data=outmi (obs=10);
  title 'First 10 Observations of the Imputed Data Set';
run;
```

The table in Figure 54.5 shows that the precision of the imputed values differs from the precision of the observed values. You can use the ROUND= option to make the imputed values consistent with the observed values.

Figure 54.5 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Oxygen	RunTime	Run Pulse
1	1	44.6090	11.3700	178.000
2	1	45.3130	10.0700	185.000
3	1	54.2970	8.6500	156.000
4	1	59.5710	8.0747	155.925
5	1	49.8740	9.2200	176.837
6	1	44.8110	11.6300	176.000
7	1	42.8857	11.9500	176.000
8	1	46.9992	10.8500	173.099
9	1	39.4420	13.0800	174.000
10	1	60.0550	8.6300	170.000

Syntax: MI Procedure

The following statements are available in PROC MI:

PROC MI < *options* > ;
  BY *variables* ;
  CLASS *variables* ;
  EM < *options* > ;
  FREQ *variable* ;
  MCMC < *options* > ;
  MONOTONE < *options* > ;
  TRANSFORM *transform (variables* < */ options* >*)* < *. . . transform (variables* < */ options* >*)* > ;
  VAR *variables* ;

The BY statement specifies groups in which separate multiple imputation analyses are performed.

The CLASS statement lists the classification variables in the VAR statement. Classification variables can be either character or numeric.

The EM statement uses the EM algorithm to compute the maximum likelihood estimate (MLE) of the data with missing values, assuming a multivariate normal distribution for the data.

The FREQ statement specifies the variable that represents the frequency of occurrence for other values in the observation.

The MCMC statement uses a Markov chain Monte Carlo method to impute values for a data set with an arbitrary missing pattern, assuming a multivariate normal distribution for the data.

The MONOTONE statement specifies monotone methods to impute continuous and classification variables for a data set with a monotone missing pattern. Note that you can use either an MCMC

statement or a MONOTONE statement, but not both. When neither of these two statements is specified, the MCMC method with its default options is used.

The TRANSFORM statement lists the variables to be transformed before the imputation process. The imputed values of these transformed variables are reverse-transformed to the original forms before the imputation.

The VAR statement lists the numeric variables to be analyzed. If you omit the VAR statement, all numeric variables not listed in other statements are used.

The PROC MI statement is the only required statement for the MI procedure. The rest of this section provides detailed syntax information for each of these statements, beginning with the PROC MI statement. The remaining statements are presented in alphabetical order.

PROC MI Statement

PROC MI < *options* > **;**

Table 54.1 summarizes the options available in the PROC MI statement.

Table 54.1 Summary of PROC MI Options

Option	Description
Data Sets	
DATA=	specifies input data set
OUT=	specifies output data set with imputed values
Imputation Details	
NIMPUTE=	specifies number of imputations
SEED=	specifies seed to begin random number generator
ROUND=	specifies units to round imputed variable values
MAXIMUM=	specifies maximum values for imputed variable values
MINIMUM=	specifies minimum values for imputed variable values
MINMAXITER=	specifies maximum number of iterations to impute values in the specified range
SINGULAR=	specifies singularity tolerance
Statistical Analysis	
ALPHA=	specifies level for the confidence interval, $(1 - \alpha)$
MU0=	specifies means under the null hypothesis
Printed Output	
NOPRINT	suppresses all displayed output
SIMPLE	displays univariate statistics and correlations

The following options can be used in the PROC MI statement. They are listed in alphabetical order.

ALPHA=α

specifies that confidence limits be constructed for the mean estimates with confidence level $100(1-\alpha)\%$, where $0 < \alpha < 1$. The default is ALPHA=0.05.

DATA=*SAS-data-set*

names the SAS data set to be analyzed by PROC MI. By default, the procedure uses the most recently created SAS data set.

MAXIMUM=*numbers*

specifies maximum values for imputed variables. When an intended imputed value is greater than the maximum, PROC MI redraws another value for imputation. If only one number is specified, that number is used for all variables. If more than one number is specified, you must use a VAR statement, and the specified numbers must correspond to variables in the VAR statement. The default number is a missing value, which indicates no restriction on the maximum for the corresponding variable

The MAXIMUM= option is related to the MINIMUM= and ROUND= options, which are used to make the imputed values more consistent with the observed variable values. These options are applicable only if you use the MCMC method or the monotone regression method.

When specifying a maximum for the first variable only, you must also specify a missing value after the maximum. Otherwise, the maximum is used for all variables.
For example, the "MAXIMUM= 100 ." option sets a maximum of 100 for the first analysis variable only and no maximum for the remaining variables. The "MAXIMUM= . 100" option sets a maximum of 100 for the second analysis variable only and no maximum for the other variables.

MINIMUM=*numbers*

specifies the minimum values for imputed variables. When an intended imputed value is less than the minimum, PROC MI redraws another value for imputation. If only one number is specified, that number is used for all variables. If more than one number is specified, you must use a VAR statement, and the specified numbers must correspond to variables in the VAR statement. The default number is a missing value, which indicates no restriction on the minimum for the corresponding variable

MINMAXITER=*number*

specifies the maximum number of iterations for imputed values to be in the specified range when the option MINIMUM or MAXIMUM is also specified. The default is MINMAXITER=100.

MU0=*numbers*

THETA0=*numbers*

specifies the parameter values $\boldsymbol{\mu}_0$ under the null hypothesis $\boldsymbol{\mu} = \boldsymbol{\mu}_0$ for the population means corresponding to the analysis variables. Each hypothesis is tested with a t test. If only one number is specified, that number is used for all variables. If more than one number is specified, you must use a VAR statement, and the specified numbers must correspond to variables in the VAR statement. The default is MU0=0.

If a variable is transformed as specified in a TRANSFORM statement, then the same transformation for that variable is also applied to its corresponding specified MU0= value in the *t* test. If the parameter values μ_0 for a transformed variable are not specified, then a value of zero is used for the resulting μ_0 after transformation.

NIMPUTE=*number*

specifies the number of imputations. The default is NIMPUTE=5. You can specify NIMPUTE=0 to skip the imputation. In this case, only tables of model information, missing data patterns, descriptive statistics (SIMPLE option), and MLE from the EM algorithm (EM statement) are displayed.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

OUT=*SAS-data-set*

creates an output SAS data set containing imputation results. The data set includes an index variable, _Imputation_, to identify the imputation number. For each imputation, the data set contains all variables in the input data set with missing values being replaced by the imputed values. See the section "Output Data Sets" on page 3792 for a description of this data set.

ROUND=*numbers*

specifies the units to round variables in the imputation. If only one number is specified, that number is used for all continuous variables. If more than one number is specified, you must use a VAR statement, and the specified numbers must correspond to variables in the VAR statement. When the classification variables are listed in the VAR statement, their corresponding roundoff units are not used. The default number is a missing value, which indicates no rounding for imputed variables.

When specifying a roundoff unit for the first variable only, you must also specify a missing value after the roundoff unit. Otherwise, the roundoff unit is used for all variables. For example, the option "ROUND= 10 ." sets a roundoff unit of 10 for the first analysis variable only and no rounding for the remaining variables. The option "ROUND= . 10" sets a roundoff unit of 10 for the second analysis variable only and no rounding for other variables.

The ROUND= option sets the precision of imputed values. For example, with a roundoff unit of 0.001, each value is rounded to the nearest multiple of 0.001. That is, each value has three significant digits after the decimal point. See Example 54.3 for an illustration of this option.

SEED=*number*

specifies a positive integer to start the pseudo-random number generator. The default is a value generated from reading the time of day from the computer's clock. However, in order to duplicate the results under identical situations, you must use the same value of the seed explicitly in subsequent runs of the MI procedure.

The seed information is displayed in the "Model Information" table so that the results can be reproduced by specifying this seed with the SEED= option. You need to specify the same seed number in the future to reproduce the results.

SIMPLE
displays simple descriptive univariate statistics and pairwise correlations from available cases. For a detailed description of these statistics, see the section "Descriptive Statistics" on page 3768.

SINGULAR=*p*
specifies the criterion for determining the singularity of a covariance matrix based on standardized variables, where $0 < p < 1$. The default is SINGULAR=1E−8.

Suppose that $\mathbf{S}$ is a covariance matrix and v is the number of variables in $\mathbf{S}$. Based on the spectral decomposition $\mathbf{S} = \mathbf{\Gamma}\mathbf{\Lambda}\mathbf{\Gamma}'$, where $\mathbf{\Lambda}$ is a diagonal matrix of eigenvalues λ_j, $j = 1, \ldots, v$, where $\lambda_i \geq \lambda_j$ when $i < j$, and $\mathbf{\Gamma}$ is a matrix with the corresponding orthonormal eigenvectors of $\mathbf{S}$ as columns, $\mathbf{S}$ is considered singular when an eigenvalue λ_j is less than $p\bar{\lambda}$, where the average $\bar{\lambda} = \sum_{k=1}^{v} \lambda_k / v$.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MI to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the MI procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

The CLASS statement specifies the classification variables in the VAR statement. Classification variables can be either character or numeric. The CLASS statement must be used in conjunction with the MONOTONE statement.

Classification levels are determined from the formatted values of the classification variables. See "The FORMAT Procedure" in the *Base SAS Procedures Guide* for details.

EM Statement

EM < *options* > **;**

The expectation-maximization (EM) algorithm is a technique for maximum likelihood estimation in parametric models for incomplete data. The EM statement uses the EM algorithm to compute the MLE for $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, the means and covariance matrix, of a multivariate normal distribution from the input data set with missing values. Either the means and covariances from complete cases or the means and standard deviations from available cases can be used as the initial estimates for the EM algorithm. You can also specify the correlations for the estimates from available cases.

You can also use the EM statement with the NIMPUTE=0 option in the PROC MI statement to compute the EM estimates without multiple imputation, as shown in Example 54.1.

The following seven options are available with the EM statement:

CONVERGE=*p*

XCONV=*p*

sets the convergence criterion. The value must be between 0 and 1. The iterations are considered to have converged when the change in the parameter estimates between iteration steps is less than p for each parameter—that is, for each of the means and covariances. For each parameter, the change is a relative change if the parameter is greater than 0.01 in absolute value; otherwise, it is an absolute change. By default, CONVERGE=1E−4.

INITIAL=CC | AC | AC(R=*r***)**

sets the initial estimates for the EM algorithm. The INITIAL=CC option uses the means and covariances from complete cases; the INITIAL=AC option uses the means and standard deviations from available cases and the correlations are set to zero; and the INITIAL=AC(R= r) option uses the means and standard deviations from available cases with correlation r, where $-1/(p-1) < r < 1$ and p is the number of variables to be analyzed. The default is INITIAL=AC.

ITPRINT

prints the iteration history in the EM algorithm.

MAXITER=*number*

specifies the maximum number of iterations used in the EM algorithm. The default is MAXITER=200.

OUT=*SAS-data-set*

creates an output SAS data set containing results from the EM algorithm. The data set contains all variables in the input data set, with missing values being replaced by the expected values from the EM algorithm. See the section "Output Data Sets" on page 3792 for a description of this data set.

OUTEM=*SAS-data-set*
creates an output SAS data set of TYPE=COV containing the MLE of the parameter vector $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. These estimates are computed with the EM algorithm. See the section "Output Data Sets" on page 3792 for a description of this output data set.

OUTITER < (*options*) > =*SAS-data-set*
creates an output SAS data set of TYPE=COV containing parameters for each iteration. The data set includes a variable named _Iteration_ to identify the iteration number. The parameters in the output data set depend on the options specified. You can specify the MEAN and COV options to output the mean and covariance parameters. When no options are specified, the output data set contains the mean parameters for each iteration. See the section "Output Data Sets" on page 3792 for a description of this data set.

FREQ Statement

FREQ *variable* **;**

If one variable in your input data set represents the frequency of occurrence of other values in the observation, specify the variable name in a FREQ statement. PROC MI then treats the data set as if each observation appears n times, where n is the value of the FREQ variable for the observation. If the value of the FREQ variable is less than one, the observation is not used in the analysis. Only the integer portion of the value is used. The total number of observations is considered to be equal to the sum of the FREQ variable when PROC MI calculates significance probabilities.

MCMC Statement

MCMC < *options* > **;**

The MCMC statement specifies the details of the MCMC method for imputation.

Table 54.2 summarizes the options available for the MCMC statement.

Table 54.2 Summary of Options in MCMC

Option	Description
Data Sets	
INEST=	inputs parameter estimates for imputations
OUTEST=	outputs parameter estimates used in imputations
OUTITER=	outputs parameter estimates used in iterations
Imputation Details	
IMPUTE=	specifies monotone/full imputation
CHAIN=	specifies single/multiple chain
NBITER=	specifies number of burn-in iterations for each chain
NITER=	specifies number of iterations between imputations in a chain

Table 54.2 *continued*

Option	Description
INITIAL=	specifies initial parameter estimates for MCMC
PRIOR=	specifies prior parameter information
START=	specifies starting parameters
ODS Output Graphics	
PLOTS=TRACE	displays trace plots
PLOTS=ACF	displays autocorrelation plots
Traditional Graphics	
TIMEPLOT	displays trace plots
ACFPLOT	displays autocorrelation plots
GOUT=	graphics catalog name for saving graphics output
Printed Output	
WLF	displays worst linear function
DISPLAYINIT	displays initial parameter values for MCMC

The following options are available for the MCMC statement (in alphabetical order).

ACFPLOT *< (options < / display-options >) >*

displays the traditional autocorrelation function plots of parameters from iterations. The ACFPLOT option is applicable only if the **ods graphics on** statement is not specified.

The available options are as follows.

COV *< (< variables > < variable1*variable2 > < . . . variable1*variable2 >) >*

displays plots of variances for variables in the list and covariances for pairs of variables in the list. When the option COV is specified without variables, variances for all variables and covariances for all pairs of variables are used.

MEAN *< (variables) >*

displays plots of means for variables in the list. When the option MEAN is specified without variables, all variables are used.

WLF

displays the plot for the worst linear function.

When the ACFPLOT is specified without the preceding options, the procedure displays plots of means for all variables that are used.

The display options provide additional information for the autocorrelation function plots. The available display options are as follows:

CCONF=*color*

specifies the color of the displayed confidence limits. The default is CCONF=BLACK.

CFRAME=*color*

specifies the color for filling the area enclosed by the axes and the frame. By default, this area is not filled.

CNEEDLES=*color*
specifies the color of the vertical line segments (needles) that connect autocorrelations to the reference line. The default is CNEEDLES=BLACK.

CREF=*color*
specifies the color of the displayed reference line. The default is CREF=BLACK.

CSYMBOL=*color*
specifies the color of the displayed data points. The default is CSYMBOL=BLACK.

HSYMBOL=*number*
specifies the height of data points in percentage screen units. The default is HSYMBOL=1.

LCONF=*linetype*
specifies the line type for the displayed confidence limits. The default is LCONF=1, a solid line.

LOG
requests that the logarithmic transformations of parameters be used to compute the autocorrelations; it is generally used for the variances of variables. When a parameter has values less than or equal to zero, the corresponding plot is not created.

LREF=*linetype*
specifies the line type for the displayed reference line. The default is LREF=3, a dashed line.

NAME=*'string'*
specifies a descriptive name, up to eight characters, that appears in the name field of the PROC GREPLAY master menu. The default is NAME='MI'.

NLAG=*number*
specifies the maximum lag of the series. The default is NLAG=20. The autocorrelations at each lag are displayed in the graph.

SYMBOL=*value*
specifies the symbol for data points in percentage screen units. The default is SYMBOL=STAR.

TITLE=*'string'*
specifies the title to be displayed in the autocorrelation function plots. The default is TITLE='Autocorrelation Plot'.

WCONF=*number*
specifies the width of the displayed confidence limits in percentage screen units. If you specify the WCONF=0 option, the confidence limits are not displayed. The default is WCONF=1.

WNEEDLES=*number*
specifies the width of the displayed needles that connect autocorrelations to the reference line, in percentage screen units. If you specify the WNEEDLES=0 option, the needles are not displayed. The default is WNEEDLES=1.

WREF=*number*
specifies the width of the displayed reference line in percentage screen units. If

you specify the WREF=0 option, the reference line is not displayed. The default is WREF=1.

For example, the following statement requests autocorrelation function plots for the means and variances of the variable y1, respectively:

```
acfplot( mean( y1) cov(y1) /log);
```

Logarithmic transformations of both the means and variances are used in the plots. For a detailed description of the autocorrelation function plot, see the section "Autocorrelation Function Plot" on page 3790; see also Schafer (1997, pp. 120–126) and the *SAS/ETS User's Guide*.

CHAIN=SINGLE | MULTIPLE
specifies whether a single chain is used for all imputations or a separate chain is used for each imputation. The default is CHAIN=SINGLE.

DISPLAYINIT
displays initial parameter values in the MCMC method for each imputation.

GOUT=*graphics-catalog*
specifies the graphics catalog for saving graphics output from PROC MI. The default is WORK.GSEG. For more information, see "The GREPLAY Procedure" in *SAS/GRAPH Software: Reference*.

IMPUTE=FULL | MONOTONE
specifies whether a full-data imputation is used for all missing values or a monotone-data imputation is used for a subset of missing values to make the imputed data sets have a monotone missing pattern. The default is IMPUTE=FULL. When IMPUTE=MONOTONE is specified, the order in the VAR statement is used to complete the monotone pattern.

INEST=*SAS-data-set*
names a SAS data set of TYPE=EST containing parameter estimates for imputations. These estimates are used to impute values for observations in the DATA= data set. A detailed description of the data set is provided in the section "Input Data Sets" on page 3791.

INITIAL=EM <(*options***)>**

INITIAL=INPUT=*SAS-data-set*
specifies the initial mean and covariance estimates for the MCMC method. The default is INITIAL=EM.

You can specify INITIAL=INPUT=*SAS-data-set* to read the initial estimates of the mean and covariance matrix for each imputation from a SAS data set. See the section "Input Data Sets" on page 3791 for a description of this data set.

With INITIAL=EM, PROC MI derives parameter estimates for a posterior mode, the highest observed-data posterior density, from the EM algorithm. The MLE from the EM algorithm is used to start the EM algorithm for the posterior mode, and the resulting EM estimates are used to begin the MCMC method. The prior information specified in the PRIOR= option is also used in the process to compute the posterior mode.

The following four options are available with INITIAL=EM:

BOOTSTRAP < =*number* >
requests bootstrap resampling, which uses a simple random sample with replacement from the input data set for the initial estimate. You can explicitly specify the number of observations in the random sample. Alternatively, you can implicitly specify the number of observations in the random sample by specifying the proportion $p, 0 < p <= 1$, to request $[np]$ observations in the random sample, where n is the number of observations in the data set and $[np]$ is the integer part of np. This produces an overdispersed initial estimate that provides different starting values for the MCMC method. If you specify the BOOTSTRAP option without the number, p=0.75 is used by default.

CONVERGE=*p*

XCONV=*p*
sets the convergence criterion. The value must be between 0 and 1. The iterations are considered to have converged when the change in the parameter estimates between iteration steps is less than p for each parameter—that is, for each of the means and covariances. For each parameter, the change is a relative change if the parameter is greater than 0.01 in absolute value; otherwise, it is an absolute change. By default, CONVERGE=1E−4.

ITPRINT
prints the iteration history in the EM algorithm for the posterior mode.

MAXITER=*number*
specifies the maximum number of iterations used in the EM algorithm. The default is MAXITER=200.

NBITER=*number*
specifies the number of burn-in iterations before the first imputation in each chain. The default is NBITER=200.

NITER=*number*
specifies the number of iterations between imputations in a single chain. The default is NITER=100.

OUTEST=*SAS-data-set*
creates an output SAS data set of TYPE=EST. The data set contains parameter estimates used in each imputation. The data set also includes a variable named _Imputation_ to identify the imputation number. See the section "Output Data Sets" on page 3792 for a description of this data set.

OUTITER < (*options*) > =*SAS-data-set*
creates an output SAS data set of TYPE=COV containing parameters used in the imputation step for each iteration. The data set includes variables named _Imputation_ and _Iteration_ to identify the imputation number and iteration number.

The parameters in the output data set depend on the options specified. You can specify the options MEAN, STD, COV, LR, LR_POST, and WLF to output parameters of means, standard deviations, covariances, −2 log LR statistic, −2 log LR statistic of the posterior mode,

and the worst linear function, respectively. When no options are specified, the output data set contains the mean parameters used in the imputation step for each iteration. See the section "Output Data Sets" on page 3792 for a description of this data set.

PLOTS < (LOG) > < = *plot-request* >

PLOTS < (LOG) > < = (*plot-request* < ... *plot-request* >) >

requests statistical graphics via the Output Delivery System (ODS). To request these graphs, you must specify the **ods graphics on** statement in addition to the following options in the MCMC statement. For more information about the **ods graphics** statement, see Chapter 21, "Statistical Graphics Using ODS."

The global plot option LOG requests that the logarithmic transformations of parameters be used. The plot request options include the following:

ACF < (*acf-options*) >

displays plots of the autocorrelation function of parameters from iterations. The default is ACF(MEAN).

ALL

produces all appropriate plots.

NONE

suppresses all plots.

TRACE < (*trace-options*) >

displays trace plots of parameters from iterations. The default is TRACE(MEAN).

The available *acf-options* are as follows:

NLAG=*n*

specifies the maximum lag of the series. The default is NLAG=20. The autocorrelations at each lag are displayed in the graph.

COV < (< variables > < variable1*variable2 > < ...variable1*variable2 >) >

displays plots of variances for variables in the list and covariances for pairs of variables in the list. When the option COV is specified without variables, variances for all variables and covariances for all pairs of variables are used.

MEAN < (variables) >

displays plots of means for variables in the list. When the option MEAN is specified without variables, all variables are used.

WLF

displays the plot for the worst linear function.

The available *trace-options* are as follows:

COV < **(** < **variables** > < **variable1*variable2** > < **...variable1*variable2** > **)** >

displays plots of variances for variables in the list and covariances for pairs of variables in the list. When the option COV is specified without variables, variances for all variables and covariances for all pairs of variables are used.

MEAN < **(** **variables** **)** >
displays plots of means for variables in the list. When the option MEAN is specified without variables, all variables are used.

WLF
displays the plot of the worst linear function.

PRIOR=*name*
specifies the prior information for the means and covariances. Valid values for *name* are as follows:

JEFFREYS	specifies a noninformative prior.
RIDGE=*number*	specifies a ridge prior.
INPUT=*SAS-data-set*	specifies a data set containing prior information.

For a detailed description of the prior information, see the section "Bayesian Estimation of the Mean Vector and Covariance Matrix" on page 3782 and the section "Posterior Step" on page 3783. If you do not specify the PRIOR= option, the default is PRIOR=JEFFREYS.

The PRIOR=INPUT= option specifies a TYPE=COV data set from which the prior information of the mean vector and the covariance matrix is read. See the section "Input Data Sets" on page 3791 for a description of this data set.

START=VALUE | DIST
specifies that the initial parameter estimates are used either as the starting value (START=VALUE) or as the starting distribution (START=DIST) in the first imputation step of each chain. If the IMPUTE=MONOTONE option is specified, then START=VALUE is used in the procedure. The default is START=VALUE.

TIMEPLOT < *(options* < */ display-options* > *)* >
displays the traditional trace (time series) plots of parameters from iterations. The TIMEPLOT option is applicable only if the **ods graphics on** statement is not specified.

The available options are as follows:

COV < *(< variables > < variable1*variable2 > < . . . variable1*variable2 >)* >
displays plots of variances for variables in the list and covariances for pairs of variables in the list. When the option COV is specified without variables, variances for all variables and covariances for all pairs of variables are used.

MEAN *< (variables) >*
displays plots of means for variables in the list. When the option MEAN is specified without variables, all variables are used.

WLF
displays the plot of the worst linear function.

When the TIMEPLOT is specified without the preceding options, the procedure displays plots of means for all variables that are used.

The display options provide additional information for the trace plots. The available display options are as follows:

CCONNECT=*color*
specifies the color of the line segments that connect data points in the trace plots. The default is CCONNECT=BLACK.

CFRAME=*color*
specifies the color for filling the area enclosed by the axes and the frame. By default, this area is not filled.

CSYMBOL=*color*
specifies the color of the data points to be displayed in the trace plots. The default is CSYMBOL=BLACK.

HSYMBOL=*number*
specifies the height of data points in percentage screen units. The default is HSYMBOL=1.

LCONNECT=*linetype*
specifies the line type for the line segments that connect data points in the trace plots. The default is LCONNECT=1, a solid line.

LOG
requests that the logarithmic transformations of parameters be used; it is generally used for the variances of variables. When a parameter value is less than or equal to zero, the value is not displayed in the corresponding plot.

NAME=*'string'*
specifies a descriptive name, up to eight characters, that appears in the name field of the PROC GREPLAY master menu. The default is NAME='MI'.

SYMBOL=*value*
specifies the symbol for data points in percentage screen units. The default is SYMBOL=PLUS.

TITLE=*'string'*
specifies the title to be displayed in the trace plots. The default is TITLE='Trace Plot'.

WCONNECT=*number*
specifies the width of the line segments that connect data points in the trace plots, in percentage screen units. If you specify the WCONNECT=0 option, the data points are not connected. The default is WCONNECT=1.

For a detailed description of the trace plot, see the section "Trace Plot" on page 3789 and Schafer (1997, pp. 120–126).

WLF

displays the worst linear function of parameters. This scalar function of parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ is "worst" in the sense that its values from iterations converge most slowly among parameters. For a detailed description of this statistic, see the section "Worst Linear Function of Parameters" on page 3789.

MONOTONE Statement

MONOTONE *< method < (< imputed < = effects > > </ options >) > > < . . . method < (< imputed < = effects > > </ options >) > > ;*

The MONOTONE statement specifies imputation methods for data sets with monotone missingness. You must also specify a VAR statement, and the data set must have a monotone missing pattern with variables ordered in the VAR list. When both MONOTONE and MCMC statements are specified, the MONOTONE statement is not used.

For each method, you can specify the imputed variables and, optionally, a set of the effects to impute these variables. Each effect is a variable or a combination of variables preceding the imputed variable in the VAR statement. The syntax for specification of effects is the same as for the GLM procedure. See Chapter 39, "The GLM Procedure," for more information.

One general form of an effect involving several variables is

X1 ∗ X2 ∗ A ∗ B ∗ C (D E)

where A, B, C, D, and E are classification variables and X1 and X2 are continuous variables.

If no covariates are specified, then all preceding variables are used as the covariates. That is, each preceding continuous variable is used as a regressor effect, and each preceding classification variable is used as a main effect. For the discriminant function method, only the continuous variables can be used as covariate effects.

When a method for continuous variables is specified without imputed variables, the method is used for all continuous variables in the VAR statement that are not specified in other methods. Similarly, when a method for classification variables is specified without imputed variables, the method is used for all classification variables in the VAR statement that are not specified in other methods.

When a MONOTONE statement is used without specifying any methods, the regression method is used for all continuous variables and the discriminant function method is used for all classification variables. The preceding variables of each imputed variable in the VAR statement are used as the covariates.

With a MONOTONE statement, the variables are imputed sequentially in the order given by the VAR statement. For a continuous variable, you can use a regression method, a regression predicted mean matching method, or a propensity score method to impute missing values.

For a nominal classification variable, you can use a discriminant function method to impute missing values without using the ordering of the class levels. For an ordinal classification variable, you can use a logistic regression method to impute missing values by using the ordering of the class levels. For a binary classification variable, either a discriminant function method or a logistic regression method can be used.

Note that except for the regression method, all other methods impute values from the observed observation values. You can specify the following methods in a MONOTONE statement.

DISCRIM < (*imputed* < = *effects* > < / *options* >) >

specifies the discriminant function method of classification variables. Only the continuous variables are allowed as covariate effects. The available options are DETAILS, PCOV=, and PRIOR=. The DETAILS option displays the group means and pooled covariance matrix used in each imputation. The PCOV= option specifies the pooled covariance used in the discriminant method. Valid values for the PCOV= option are as follows:

FIXED	uses the observed-data pooled covariance matrix for each imputation.
POSTERIOR	draws a pooled covariance matrix from its posterior distribution.

The default is PCOV=POSTERIOR. See the section "Discriminant Function Method for Monotone Missing Data" on page 3777 for a detailed description of the method.

The PRIOR= option specifies the prior probabilities of group membership. Valid values for the PRIOR= option are as follows:

EQUAL	sets the prior probabilities equal for all groups.
PROPORTIONAL	sets the prior probabilities proportion to the group sample sizes.
JEFFREYS < =*c* >	specifies a noninformative prior, $0 < c < 1$. If the number c is not specified, JEFFREYS=0.5.
RIDGE < =*d* >	specifies a ridge prior, $d > 0$. If the number d is not specified, RIDGE=0.25.

The default is PRIOR=JEFFREYS. See the section "Discriminant Function Method for Monotone Missing Data" on page 3777 for a detailed description of the method.

LOGISTIC < (*imputed* < = *effects* > < / *options* >) >

specifies the logistic regression method of classification variables. The available options are DETAILS, ORDER=, and DESCENDING. The DETAILS option displays the regression coefficients in the logistic regression model used in each imputation.

When the imputed variable has more than two response levels, the ordinal logistic regression method is used. The ORDER= option specifies the sorting order for the levels of the response variable. Valid values for the ORDER= option are as follows:

DATA	sorts by the order of appearance in the input data set.
FORMATTED	sorts by their external formatted values.

FREQ	sorts by the descending frequency counts.
INTERNAL	sorts by the unformatted values.

By default, ORDER=FORMATTED.

The option DESCENDING reverses the sorting order for the levels of the response variables.

See the section "Logistic Regression Method for Monotone Missing Data" on page 3779 for a detailed description of the method.

PROPENSITY < (*imputed* < = *effects* > < / *options* >) >
specifies the propensity scores method of variables. Each variable is either a classification variable or a continuous variable. The available options are DETAILS and NGROUPS=. The DETAILS option displays the regression coefficients in the logistic regression model for propensity scores. The NGROUPS= option specifies the number of groups created based on propensity scores. The default is NGROUPS=5.

See the section "Propensity Score Method for Monotone Missing Data" on page 3776 for a detailed description of the method.

REG | REGRESSION < (*imputed* < = *effects* > < / **DETAILS** >) >
specifies the regression method of continuous variables. The DETAILS option displays the regression coefficients in the regression model used in each imputation.

With a regression method, the MAXIMUM=, MINIMUM=, and ROUND= options can be used to make the imputed values more consistent with the observed variable values.

See the section "Regression Method for Monotone Missing Data" on page 3774 for a detailed description of the method.

REGPMM < (*imputed* < = *effects* > < *options* >) >

REGPREDMEANMATCH < (*imputed* < = *effects* > < *options* >) >
specifies the predictive mean matching method for continuous variables. This method is similar to the regression method except that it imputes a value randomly from a set of observed values whose predicted values are closest to the predicted value for the missing value from the simulated regression model (Heitjan and Little 1991; Schenker and Taylor 1996).

The available options are DETAILS and K=. The DETAILS option displays the regression coefficients in the regression model used in each imputation. The K= option specifies the number of closest observations to be used in the selection. The default is K=5.

See the section "Predictive Mean Matching Method for Monotone Missing Data" on page 3775 for a detailed description of the method.

With a MONOTONE statement, the missing values of a variable are imputed when the variable is either explicitly specified in the method or implicitly specified when a method is specified without imputed variables. These variables are imputed sequentially in the order specified in the VAR statement. For example, the following MI procedure statements use the logistic regression method to impute variable `c1` from effects `y1`, `y2`, and `y1 * y2` first, and then use the regression method to impute variable `y3` from effects `y1`, `y2`, and `c1`:

```
proc mi;
   class c1;
   var y1 y2 c1 y3;
   monotone reg(y3= y1 y2 c1) logistic(c1= y1 y2 y1*y2);
run;
```

The variables y1 and y2 are not imputed since y1 is the leading variable in the VAR statement and y2 is not specified as an imputed variable in the MONOTONE statement.

TRANSFORM Statement

TRANSFORM *transform (variables </ options>) < . . . transform (variables </ options>) >* **;**

The TRANSFORM statement lists the transformations and their associated variables to be transformed. The options are transformation options that provide additional information for the transformation.

The MI procedure assumes that the data are from a multivariate normal distribution when either the regression method or the MCMC method is used. When some variables in a data set are clearly nonnormal, it is useful to transform these variables to conform to the multivariate normality assumption. With a TRANSFORM statement, variables are transformed before the imputation process, and these transformed variable values are displayed in all of the results. When you specify an OUT= option, the variable values are back-transformed to create the imputed data set.

The following transformations can be used in the TRANSFORM statement:

BOXCOX
specifies the Box-Cox transformation of variables. The variable Y is transformed to $\frac{(Y+c)^{\lambda}-1}{\lambda}$, where c is a constant such that each value of $Y + c$ must be positive. If the specified constant $\lambda = 0$, the logarithmic transformation is used.

EXP
specifies the exponential transformation of variables. The variable Y is transformed to $e^{(Y+c)}$, where c is a constant.

LOG
specifies the logarithmic transformation of variables. The variable Y is transformed to $\log(Y+c)$, where c is a constant such that each value of $Y + c$ must be positive.

LOGIT
specifies the logit transformation of variables. The variable Y is transformed to $\log(\frac{Y/c}{1-Y/c})$, where the constant $c > 0$ and the values of Y/c must be between 0 and 1.

POWER
specifies the power transformation of variables. The variable Y is transformed to $(Y + c)^{\lambda}$, where c is a constant such that each value of $Y + c$ must be positive and the constant $\lambda \neq 0$.

The following options provide the constant c and λ values in the transformations.

C=*number*
specifies the c value in the transformation. The default is $c = 1$ for logit transformation and $c = 0$ for other transformations.

LAMBDA=*number*
specifies the λ value in the power and Box-Cox transformations. You must specify the λ value for these two transformations.

For example, the following statement requests that variables log(y1), a logarithmic transformation for the variable y1, and $\sqrt{y2 + 1}$, a power transformation for the variable y2, be used in the imputation:

```
transform log(y1) power(y2/c=1 lambda=.5);
```

If the MU0= option is used to specify a parameter value μ_0 for a transformed variable, the same transformation for the variable is also applied to its corresponding MU0= value in the t test. Otherwise, $\mu_0 = 0$ is used for the transformed variable. See Example 54.10 for a usage of the TRANSFORM statement.

VAR Statement

VAR *variables* **;**

The VAR statement lists the variables to be analyzed. The variables can be either character or numeric. If you omit the VAR statement, all continuous variables not mentioned in other statements are used. The VAR statement is required if you specify a MONOTONE statement, an IMPUTE=MONOTONE option in the MCMC statement, or more than one number in the MU0=, MAXIMUM=, MINIMUM=, or ROUND= option.

The character variables are allowed only when they are specified as CLASS variables and the MONOTONE statement is also specified.

Details: MI Procedure

Descriptive Statistics

Suppose $\mathbf{Y} = (\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n)'$ is the $(n \times p)$ matrix of complete data, which might not be fully observed, n_0 is the number of observations fully observed, and n_j is the number of observations with observed values for variable Y_j.

With complete cases, the sample mean vector is

$$\bar{\mathbf{y}} = \frac{1}{n_0} \sum \mathbf{y}_i$$

and the CSSCP matrix is

$$\sum (\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})'$$

where each summation is over the fully observed observations.

The sample covariance matrix is

$$\mathbf{S} = \frac{1}{n_0 - 1} \sum (\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})'$$

and is an unbiased estimate of the covariance matrix.

The correlation matrix **R** containing the Pearson product-moment correlations of the variables is derived by scaling the corresponding covariance matrix:

$$\mathbf{R} = \mathbf{D}^{-1}\mathbf{S}\,\mathbf{D}^{-1}$$

where **D** is a diagonal matrix whose diagonal elements are the square roots of the diagonal elements of **S**.

With available cases, the corrected sum of squares for variable Y_j is

$$\sum (y_{ji} - \bar{y}_j)^2$$

where $\bar{y}_j = \frac{1}{n_j} \sum y_{ji}$ is the sample mean and each summation is over observations with observed values for variable Y_j.

The variance is

$$s_{jj}^2 = \frac{1}{n_j - 1} \sum (y_{ji} - \bar{y}_j)^2$$

The correlations for available cases contain pairwise correlations for each pair of variables. Each correlation is computed from all observations that have nonmissing values for the corresponding pair of variables.

EM Algorithm for Data with Missing Values

The EM algorithm (Dempster, Laird, and Rubin 1977) is a technique that finds maximum likelihood estimates in parametric models for incomplete data. The books by Little and Rubin (2002), Schafer (1997), and McLachlan and Krishnan (1997) provide a detailed description and applications of the EM algorithm.

The EM algorithm is an iterative procedure that finds the MLE of the parameter vector by repeating the following steps:

1. The expectation E-step
Given a set of parameter estimates, such as a mean vector and covariance matrix for a multivariate normal distribution, the E-step calculates the conditional expectation of the complete-data log likelihood given the observed data and the parameter estimates.

2. The maximization M-step
Given a complete-data log likelihood, the M-step finds the parameter estimates to maximize the complete-data log likelihood from the E-step.

The two steps are iterated until the iterations converge.

In the EM process, the observed-data log likelihood is nondecreasing at each iteration. For multivariate normal data, suppose there are G groups with distinct missing patterns. Then the observed-data log likelihood being maximized can be expressed as

$$\log L(\theta|Y_{obs}) = \sum_{g=1}^{G} \log L_g(\boldsymbol{\theta}|Y_{obs})$$

where $\log L_g(\boldsymbol{\theta}|Y_{obs})$ is the observed-data log likelihood from the gth group, and

$$\log L_g(\boldsymbol{\theta}|Y_{obs}) = -\frac{n_g}{2}\log|\Sigma_g| - \frac{1}{2}\sum_{ig}(\mathbf{y}_{ig}-\boldsymbol{\mu}_g)'\boldsymbol{\Sigma}_g{}^{-1}(\mathbf{y}_{ig}-\boldsymbol{\mu}_g)$$

where n_g is the number of observations in the gth group, the summation is over observations in the gth group, $\mathbf{y}_{ig}$ is a vector of observed values corresponding to observed variables, $\boldsymbol{\mu}_g$ is the corresponding mean vector, and $\boldsymbol{\Sigma}_g$ is the associated covariance matrix.

A sample covariance matrix is computed at each step of the EM algorithm. If the covariance matrix is singular, the linearly dependent variables for the observed data are excluded from the likelihood function. That is, for each observation with linear dependency among its observed variables, the dependent variables are excluded from the likelihood function. Note that this can result in an unexpected change in the likelihood between iterations prior to the final convergence.

See Schafer (1997, pp. 163–181) for a detailed description of the EM algorithm for multivariate normal data.

PROC MI uses the means and standard deviations from available cases as the initial estimates for the EM algorithm. The correlations are set to zero. These initial estimates provide a good starting value with positive definite covariance matrix. For a discussion of suggested starting values for the algorithm, see Schafer (1997, p. 169).

You can specify the convergence criterion with the CONVERGE= option in the EM statement. The iterations are considered to have converged when the maximum change in the parameter estimates between iteration steps is less than the value specified. You can also specify the maximum number of iterations used in the EM algorithm with the MAXITER= option.

The MI procedure displays tables of the initial parameter estimates used to begin the EM process and the MLE parameter estimates derived from EM. You can also display the EM iteration history

with the ITPRINT option. PROC MI lists the iteration number, the likelihood −2 log L, and the parameter values $\boldsymbol{\mu}$ at each iteration. You can also save the MLE derived from the EM algorithm in a SAS data set by specifying the OUTEM= option.

Statistical Assumptions for Multiple Imputation

The MI procedure assumes that the data are from a continuous multivariate distribution and contain missing values that can occur for any of the variables. It also assumes that the data are from a multivariate normal distribution when either the regression method or the MCMC method is used.

Suppose $\mathbf{Y}$ is the $n \times p$ matrix of complete data, which is not fully observed, and denote the observed part of $\mathbf{Y}$ by $\mathbf{Y}_{obs}$ and the missing part by $\mathbf{Y}_{mis}$. The MI and MIANALYZE procedures assume that the missing data are missing at random (MAR); that is, the probability that an observation is missing can depend on $\mathbf{Y}_{obs}$, but not on $\mathbf{Y}_{mis}$ (Rubin 1976; 1987, p. 53).

To be more precise, suppose that $\mathbf{R}$ is the $n \times p$ matrix of response indicators whose elements are zero or one depending on whether the corresponding elements of Y are missing or observed. Then the MAR assumption is that the distribution of $\mathbf{R}$ can depend on Y_{obs} but not on Y_{mis}:

$$\text{pr}(\mathbf{R}|Y_{obs}, Y_{mis}) = \text{pr}(\mathbf{R}|Y_{obs})$$

For example, consider a trivariate data set with variables Y_1 and Y_2 fully observed, and a variable Y_3 that has missing values. MAR assumes that the probability that Y_3 is missing for an individual can be related to the individual's values of variables Y_1 and Y_2, but not to its value of Y_3. On the other hand, if a complete case and an incomplete case for Y_3 with exactly the same values for variables Y_1 and Y_2 have systematically different values, then there exists a response bias for Y_3, and MAR is violated.

The MAR assumption is not the same as missing completely at random (MCAR), which is a special case of MAR. Under the MCAR assumption, the missing data values are a simple random sample of all data values; the missingness does not depend on the values of any variables in the data set.

Although the MAR assumption cannot be verified with the data and it can be questionable in some situations, the assumption becomes more plausible as more variables are included in the imputation model (Schafer 1997, pp. 27–28; van Buuren, Boshuizen, and Knook, 1999, p. 687).

Furthermore, the MI and MIANALYZE procedures assume that the parameters $\boldsymbol{\theta}$ of the data model and the parameters $\boldsymbol{\phi}$ of the model for the missing-data indicators are distinct. That is, knowing the values of $\boldsymbol{\theta}$ does not provide any additional information about $\boldsymbol{\phi}$, and vice versa. If both the MAR and distinctness assumptions are satisfied, the missing-data mechanism is said to be ignorable (Rubin 1987, pp. 50–54; Schafer 1997, pp. 10–11) .

Missing Data Patterns

The MI procedure sorts the data into groups based on whether the analysis variables are observed or missing. Note that the input data set does not need to be sorted in any order.

For example, with variables Y_1, Y_2, and Y_3 (in that order) in a data set, up to eight groups of observations can be formed from the data set. Figure 54.6 displays the eight groups of observations and an unique missing pattern for each group:

Figure 54.6 Missing Data Patterns

Missing Data Patterns

Group	Y1	Y2	Y3
1	X	X	X
2	X	X	.
3	X	.	X
4	X	.	.
5	.	X	X
6	.	X	.
7	.	.	X
8	.	.	.

Here, an "X" means that the variable is observed in the corresponding group and a "." means that the variable is missing.

The variable order is used to derive the order of the groups from the data set, and thus determines the order of missing values in the data to be imputed. If you specify a different order of variables in the VAR statement, then the results are different even if the other specifications remain the same.

A data set with variables $Y_1, Y_2, \ldots, Y_p$ (in that order) is said to have a *monotone missing pattern* when the event that a variable Y_j is missing for a particular individual implies that all subsequent variables $Y_k, k > j$, are missing for that individual. Alternatively, when a variable Y_j is observed for a particular individual, it is assumed that all previous variables $Y_k, k < j$, are also observed for that individual.

For example, Figure 54.7 displays a data set of three variables with a monotone missing pattern.

Figure 54.7 Monotone Missing Patterns

Monotone Missing Data Patterns

Group	Y1	Y2	Y3
1	X	X	X
2	X	X	.
3	X	.	.

Figure 54.8 displays a data set of three variables with a non-monotone missing pattern.

Figure 54.8 Non-monotone Missing Patterns

Non-monotone Missing Data Patterns

Group	Y1	Y2	Y3
1	X	X	X
2	X	.	X
3	.	X	.
4	.	.	X

A data set with an *arbitrary missing pattern* is a data set with either a monotone missing pattern or a non-monotone missing pattern.

Imputation Methods

This section describes the methods for multiple imputation that are available in the MI procedure. The method of choice depends on the pattern of missingness in the data and the type of the imputed variable, as summarized in Table 54.3.

Table 54.3 Imputation Methods in PROC MI

Pattern of Missingness	Type of Imputed Variable	Recommended Methods
Monotone	Continuous	• Regression • Predicted mean matching • Propensity score
Monotone	Classification (Ordinal)	• Logistic regression
Monotone	Classification (Nominal)	• Discriminant function method
Arbitrary	Continuous	• MCMC full-data imputation • MCMC monotone-data imputation

To impute missing values for a continuous variable in data sets with monotone missing patterns, you should use either a parametric method that assumes multivariate normality or a nonparametric method that uses propensity scores (Rubin 1987, pp. 124, 158; Lavori, Dawson, and Shera 1995). Parametric methods available include the regression method (Rubin 1987, pp. 166–167) and the predictive mean matching method (Heitjan and Little 1991; Schenker and Taylor 1996).

To impute missing values for a classification variable in data sets with monotone missing patterns, you should use the logistic regression method or the discriminant function method. Use the logistic regression method when the classification variable has a binary or ordinal response, and use the discriminant function method when the classification variable has a binary or nominal response.

For continuous variables in data sets with arbitrary missing patterns, you can use the Markov chain Monte Carlo (MCMC) method (Schafer 1997) to impute either all the missing values or just enough missing values to make the imputed data sets have monotone missing patterns.

With a monotone missing data pattern, you have greater flexibility in your choice of imputation models. In addition to the MCMC method, you can implement other methods, such as the regression method, that do not use Markov chains. You can also specify a different set of covariates for each imputed variable.

With an arbitrary missing data pattern, you can often use the MCMC method, which creates multiple imputations by drawing simulations from a Bayesian predictive distribution for normal data. Another way to handle a data set with an arbitrary missing data pattern is to use the MCMC approach to impute just enough values to make the missing data pattern monotone. Then, you can use a more flexible imputation method. This approach is described in the section "Producing Monotone Missingness with the MCMC Method" on page 3785.

Note that all continuous variables are standardized before the imputation process and then are transformed back to the original scale after the imputation process.

Although the regression and MCMC methods assume multivariate normality, inferences based on multiple imputation can be robust to departures from multivariate normality if the amount of missing information is not large, because the imputation model is effectively applied not to the entire data set but only to its missing part (Schafer 1997, pp. 147–148).

You can also use a TRANSFORM statement to transform variables to conform to the multivariate normality assumption. Variables are transformed before the imputation process and then are reverse-transformed to create the imputed data set.

Li (1988) presents a theoretical argument for convergence of the MCMC method in the continuous case and uses it to create imputations for incomplete multivariate continuous data. In practice, however, it is not easy to check the convergence of a Markov chain, especially for a large number of parameters. PROC MI generates statistics and plots that you can use to check for convergence of the MCMC method. The details are described in the section "Checking Convergence in MCMC" on page 3788.

Regression Method for Monotone Missing Data

The regression method is the default imputation method for continuous variables in a data set with a monotone missing pattern.

In the regression method, a regression model is fitted for a continuous variable with the covariates constructed from a set of effects. Based on the fitted regression model, a new regression model is simulated from the posterior predictive distribution of the parameters and is used to impute the missing values for each variable (Rubin 1987, pp. 166–167). That is, for a continuous variable Y_j with missing values, a model

$$Y_j = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k$$

is fitted using observations with observed values for the variable Y_j and its covariates $X_1, X_2, \ldots, X_k$.

The fitted model includes the regression parameter estimates $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k)$ and the associated covariance matrix $\hat{\sigma}_j^2 \mathbf{V}_j$, where $\mathbf{V}_j$ is the usual $\mathbf{X}'\mathbf{X}$ inverse matrix derived from the intercept and covariates $X_1, X_2, \ldots, X_k$.

The following steps are used to generate imputed values for each imputation:

1. New parameters $\boldsymbol{\beta}_* = (\beta_{*0}, \beta_{*1}, \ldots, \beta_{*(k)})$ and σ_{*j}^2 are drawn from the posterior predictive distribution of the parameters. That is, they are simulated from $(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k)$, σ_j^2, and $\mathbf{V}_j$. The variance is drawn as

   $$\sigma_{*j}^2 = \hat{\sigma}_j^2 (n_j - k - 1)/g$$

   where g is a $\chi^2_{n_j - k - 1}$ random variate and n_j is the number of nonmissing observations for Y_j. The regression coefficients are drawn as

   $$\boldsymbol{\beta}_* = \hat{\boldsymbol{\beta}} + \sigma_{*j} \mathbf{V}'_{hj} \mathbf{Z}$$

   where $\mathbf{V}'_{hj}$ is the upper triangular matrix in the Cholesky decomposition, $\mathbf{V}_j = \mathbf{V}'_{hj} \mathbf{V}_{hj}$, and $\mathbf{Z}$ is a vector of $k + 1$ independent random normal variates.

2. The missing values are then replaced by

   $$\beta_{*0} + \beta_{*1}\, x_1 + \beta_{*2}\, x_2 + \ldots + \beta_{*(k)}\, x_k + z_i\, \sigma_{*j}$$

   where $x_1, x_2, \ldots, x_k$ are the values of the covariates and z_i is a simulated normal deviate.

Predictive Mean Matching Method for Monotone Missing Data

The predictive mean matching method is also an imputation method available for continuous variables. It is similar to the regression method except that for each missing value, it imputes a value randomly from a set of observed values whose predicted values are closest to the predicted value for the missing value from the simulated regression model (Heitjan and Little 1991; Schenker and Taylor 1996).

Following the description of the model in the section "Regression Method for Monotone Missing Data" on page 3774, the following steps are used to generate imputed values:

1. New parameters $\boldsymbol{\beta}_* = (\beta_{*0}, \beta_{*1}, \ldots, \beta_{*(k)})$ and σ_{*j}^2 are drawn from the posterior predictive distribution of the parameters. That is, they are simulated from $(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k)$, σ_j^2, and $\mathbf{V}_j$. The variance is drawn as

   $$\sigma_{*j}^2 = \hat{\sigma}_j^2 (n_j - k - 1)/g$$

   where g is a $\chi^2_{n_j - k - 1}$ random variate and n_j is the number of nonmissing observations for Y_j. The regression coefficients are drawn as

   $$\boldsymbol{\beta}_* = \hat{\boldsymbol{\beta}} + \sigma_{*j} \mathbf{V}'_{hj} \mathbf{Z}$$

where $\mathbf{V}'_{hj}$ is the upper triangular matrix in the Cholesky decomposition, $\mathbf{V}_j = \mathbf{V}'_{hj}\mathbf{V}_{hj}$, and $\mathbf{Z}$ is a vector of $k+1$ independent random normal variates.

2. For each missing value, a predicted value

$$y_{i*} = \beta_{*0} + \beta_{*1}\, x_1 + \beta_{*2}\, x_2 + \ldots + \beta_{*(k)}\, x_k$$

is computed with the covariate values $x_1, x_2, \ldots, x_k$.

3. A set of k_0 observations whose corresponding predicted values are closest to y_{i*} is generated. You can specify k_0 with the K= option.

4. The missing value is then replaced by a value drawn randomly from these k_0 observed values.

The predictive mean matching method requires the number of closest observations to be specified. A smaller k_0 tends to increase the correlation among the multiple imputations for the missing observation and results in a higher variability of point estimators in repeated sampling. On the other hand, a larger k_0 tends to lessen the effect from the imputation model and results in biased estimators (Schenker and Taylor 1996, p. 430).

The predictive mean matching method ensures that imputed values are plausible and might be more appropriate than the regression method if the normality assumption is violated (Horton and Lipsitz 2001, p. 246).

Propensity Score Method for Monotone Missing Data

The propensity score method is another imputation method available for continuous variables when the data set has a monotone missing pattern.

A propensity score is generally defined as the conditional probability of assignment to a particular treatment given a vector of observed covariates (Rosenbaum and Rubin 1983). In the propensity score method, for a variable with missing values, a propensity score is generated for each observation to estimate the probability that the observation is missing. The observations are then grouped based on these propensity scores, and an approximate Bayesian bootstrap imputation (Rubin 1987, p. 124) is applied to each group (Lavori, Dawson, and Shera 1995).

The propensity score method uses the following steps to impute values for variable Y_j with missing values:

1. Create an indicator variable R_j with the value 0 for observations with missing Y_j and 1 otherwise.

2. Fit a logistic regression model

$$\text{logit}(p_j) = \beta_0 + \beta_1\, X_1 + \beta_2\, X_2 + \ldots + \beta_k\, X_k$$

where $X_1, X_2, \ldots, X_k$ are covariates for Y_j, $p_j = Pr(R_j = 0 | X_1, X_2, \ldots, X_k)$, and $\text{logit}(p) = \log(p/(1-p))$.

3. Create a propensity score for each observation to estimate the probability that it is missing.
4. Divide the observations into a fixed number of groups (typically assumed to be five) based on these propensity scores.
5. Apply an approximate Bayesian bootstrap imputation to each group. In group k, suppose that Y_{obs} denotes the n_1 observations with nonmissing Y_j values and Y_{mis} denotes the n_0 observations with missing Y_j. The approximate Bayesian bootstrap imputation first draws n_1 observations randomly with replacement from Y_{obs} to create a new data set Y^*_{obs}. This is a nonparametric analog of drawing parameters from the posterior predictive distribution of the parameters. The process then draws the n_0 values for Y_{mis} randomly with replacement from Y^*_{obs}.

Steps 1 through 5 are repeated sequentially for each variable with missing values.

Note that the propensity score method was originally designed for a randomized experiment with repeated measures on the response variables. The goal was to impute the missing values on the response variables. The method uses only the covariate information that is associated with whether the imputed variable values are missing. It does not use correlations among variables. It is effective for inferences about the distributions of individual imputed variables, such as a univariate analysis, but it is not appropriate for analyses involving relationship among variables, such as a regression analysis (Schafer 1999, p. 11). It can also produce badly biased estimates of regression coefficients when data on predictor variables are missing (Allison 2000).

Discriminant Function Method for Monotone Missing Data

The discriminant function method is the default imputation method for classification variables in a data set with a monotone missing pattern.

For a nominal classification variable Y_j with responses $1, \ldots, g$ and a set of effects from its preceding variables, if the covariates $X_1, X_2, \ldots, X_k$ associated with these effects within each group are approximately multivariate normal and the within-group covariance matrices are approximately equal, the discriminant function method (Brand 1999, pp. 95–96) can be used to impute missing values for the variable Y_j.

Denote the group-specific means for covariates $X_1, X_2, \ldots, X_k$ by

$$\overline{\mathbf{X}}_t = (\overline{X}_{t1}, \overline{X}_{t2}, \ldots, \overline{X}_{tk}),\ t = 1, 2, \ldots, g$$

then the pooled covariance matrix is computed as

$$\mathbf{S} = \frac{1}{n-g}\sum_{t=1}^{g}(n_t - 1)\mathbf{S}_t$$

where $\mathbf{S}_t$ is the within-group covariance matrix, n_t is the group-specific sample size, and $n = \sum_{t=1}^{g} n_t$ is the total sample size.

In each imputation, new parameters of the group-specific means ($\mathbf{m}_{*t}$), pooled covariance matrix ($\mathbf{S}_*$), and prior probabilities of group membership (q_{*t}) can be drawn from their corresponding posterior distributions (Schafer 1997, p. 356).

Pooled Covariance Matrix and Group-Specific Means

For each imputation, the MI procedure uses either the fixed observed pooled covariance matrix (PCOV=FIXED) or a drawn pooled covariance matrix (PCOV=POSTERIOR) from its posterior distribution with a noninformative prior. That is,

$$\mathbf{\Sigma}|\mathbf{X} \quad \sim \quad W^{-1}\left(n-g,\ (n-g)\mathbf{S}\right)$$

where W^{-1} is an inverted Wishart distribution.

The group-specific means are then drawn from their posterior distributions with a noninformative prior

$$\boldsymbol{\mu}_t|(\mathbf{\Sigma}, \overline{\mathbf{X}}_t) \quad \sim \quad N\left(\overline{\mathbf{X}}_t,\ \frac{1}{n_t}\mathbf{\Sigma}\right)$$

See the section "Bayesian Estimation of the Mean Vector and Covariance Matrix" on page 3782 for a complete description of the inverted Wishart distribution and posterior distributions that use a noninformative prior.

Prior Probabilities of Group Membership

The prior probabilities are computed through the drawing of new group sample sizes. When the total sample size n is considered fixed, the group sample sizes $(n_1, n_2, \ldots, n_g)$ have a multinomial distribution. New multinomial parameters (group sample sizes) can be drawn from their posterior distribution by using a Dirichlet prior with parameters $(\alpha_1, \alpha_2, \ldots, \alpha_g)$.

After the new sample sizes are drawn from the posterior distribution of $(n_1, n_2, \ldots, n_g)$, the prior probabilities q_{*t} are computed proportionally to the drawn sample sizes.

See Schafer (1997, pp. 247–255) for a complete description of the Dirichlet prior.

Imputation Steps

The discriminant function method uses the following steps in each imputation to impute values for a nominal classification variable Y_j with g responses:

1. Draw a pooled covariance matrix $\mathbf{S}_*$ from its posterior distribution if the PCOV=POSTERIOR option is used.
2. For each group, draw group means $\mathbf{m}_{*t}$ from the observed group mean $\overline{\mathbf{X}}_t$ and either the observed pooled covariance matrix (PCOV=FIXED) or the drawn pooled covariance matrix $\mathbf{S}_*$ (PCOV=POSTERIOR).

3. For each group, compute or draw q_{*t}, prior probabilities of group membership, based on the PRIOR= option:

   - PRIOR=EQUAL, $q_{*t} = 1/g$, prior probabilities of group membership are all equal.
   - PRIOR=PROPORTIONAL, $q_{*t} = n_t/n$, prior probabilities are proportional to their group sample sizes.
   - PRIOR=JEFFREYS=*c*, a noninformative Dirichlet prior with $\alpha_t = c$ is used.
   - PRIOR=RIDGE=*d*, a ridge prior is used with $\alpha_t = d * n_t/n$ for $d \geq 1$ and $\alpha_t = d * n_t$ for $d < 1$.

4. With the group means $\mathbf{m}_{*t}$, the pooled covariance matrix $\mathbf{S}_*$, and the prior probabilities of group membership q_{*t}, the discriminant function method derives linear discriminant function and computes the posterior probabilities of an observation belonging to each group

$$p_t(\mathbf{x}) = \frac{\exp(-0.5 D_t^2(\mathbf{x}))}{\sum_{u=1}^{g} \exp(-0.5 D_u^2(\mathbf{x}))}$$

   where $D_t^2(\mathbf{x}) = (\mathbf{x} - \mathbf{m}_{*t})' \mathbf{S}_*^{-1} (\mathbf{x} - \mathbf{m}_{*t}) - 2\log(q_{*t})$ is the generalized squared distance from $\mathbf{x}$ to group t.

5. Draw a random uniform variate u, between 0 and 1, for each observation with missing group value. With the posterior probabilities, $p_1(\mathbf{x}) + p_2(\mathbf{x}) + \ldots, + p_g(\mathbf{x}) = 1$, the discriminant function method imputes $Y_j = 1$ if the value of u is less than $p_1(\mathbf{x})$, $Y_j = 2$ if the value is greater than or equal to $p_1(\mathbf{x})$ but less than $p_1(\mathbf{x}) + p_2(\mathbf{x})$, and so on.

Logistic Regression Method for Monotone Missing Data

The logistic regression method is another imputation method available for classification variables in a data set with a monotone missing pattern.

In the logistic regression method, a logistic regression model is fitted for a classification variable with a set of covariates constructed from the effects. For a binary classification variable, based on the fitted regression model, a new logistic regression model is simulated from the posterior predictive distribution of the parameters and is used to impute the missing values for each variable (Rubin 1987, pp. 169–170).

For a binary variable Y_j with responses 1 and 2, a logistic regression model is fitted using observations with observed values for the imputed variable Y_j and its covariates $X_1, X_2, \ldots, X_k$:

$$\text{logit}(p_j) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_k X_k$$

where $X_1, X_2, \ldots, X_k$ are covariates for Y_j, $p_j = \Pr(R_j = 1 | X_1, X_2, \ldots, X_k)$, and $\text{logit}(p) = \log(p/(1-p))$.

The fitted model includes the regression parameter estimates $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_k)$ and the associated covariance matrix $\mathbf{V}_j$.

The following steps are used to generate imputed values for a binary variable Y_j with responses 1 and 2:

1. New parameters $\boldsymbol{\beta}_* = (\beta_{*0}, \beta_{*1}, \ldots, \beta_{*(k)})$ are drawn from the posterior predictive distribution of the parameters.

   $$\boldsymbol{\beta}_* = \hat{\boldsymbol{\beta}} + \mathbf{V}'_{hj}\mathbf{Z}$$

   where $\mathbf{V}'_{hj}$ is the upper triangular matrix in the Cholesky decomposition, $\mathbf{V}_j = \mathbf{V}'_{hj}\mathbf{V}_{hj}$, and $\mathbf{Z}$ is a vector of $k+1$ independent random normal variates.

2. For an observation with missing Y_j and covariates $x_1, x_2, \ldots, x_k$, compute the expected probability that $Y_j = 1$:

   $$p_j = \frac{\exp(\mu_j)}{1 + \exp(\mu_j)}$$

   where $\mu_j = \beta_{*0} + \beta_{*1}\,x_1 + \beta_{*2}\,x_2 + \ldots + \beta_{*(k)}\,x_k$.

3. Draw a random uniform variate, u, between 0 and 1. If the value of u is less than p_j, impute $Y_j = 1$; otherwise impute $Y_j = 2$.

The preceding logistic regression method can be extended to include the ordinal classification variables with more than two levels of responses. The options ORDER= and DESCENDING can be used to specify the sorting order for the levels of the imputed variables.

MCMC Method for Arbitrary Missing Data

The Markov chain Monte Carlo (MCMC) method originated in physics as a tool for exploring equilibrium distributions of interacting molecules. In statistical applications, it is used to generate pseudo-random draws from multidimensional and otherwise intractable probability distributions via Markov chains. A Markov chain is a sequence of random variables in which the distribution of each element depends only on the value of the previous element.

In MCMC simulation, one constructs a Markov chain long enough for the distribution of the elements to stabilize to a stationary distribution, which is the distribution of interest. By repeatedly simulating steps of the chain, the method simulates draws from the distribution of interest. See Schafer (1997) for a detailed discussion of this method.

In Bayesian inference, information about unknown parameters is expressed in the form of a posterior probability distribution. This posterior distribution is computed using Bayes' theorem,

$$p(\boldsymbol{\theta}|y) = \frac{p(y|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int p(y|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}$$

MCMC has been applied as a method for exploring posterior distributions in Bayesian inference. That is, through MCMC, you can simulate the entire joint posterior distribution of the unknown quantities and obtain simulation-based estimates of posterior parameters that are of interest.

In many incomplete-data problems, the observed-data posterior $p(\boldsymbol{\theta}|Y_{obs})$ is intractable and cannot easily be simulated. However, when Y_{obs} is augmented by an estimated/simulated value of the missing data Y_{mis}, the complete-data posterior $p(\boldsymbol{\theta}|Y_{obs}, Y_{mis})$ is much easier to simulate. Assuming that the data are from a multivariate normal distribution, data augmentation can be applied to Bayesian inference with missing data by repeating the following steps:

1. The imputation I-step
Given an estimated mean vector and covariance matrix, the I-step simulates the missing values for each observation independently. That is, if you denote the variables with missing values for observation i by $Y_{i(mis)}$ and the variables with observed values by $Y_{i(obs)}$, then the I-step draws values for $Y_{i(mis)}$ from a conditional distribution for $Y_{i(mis)}$ given $Y_{i(obs)}$.

2. The posterior P-step
Given a complete sample, the P-step simulates the posterior population mean vector and covariance matrix. These new estimates are then used in the next I-step. Without prior information about the parameters, a noninformative prior is used. You can also use other informative priors. For example, a prior information about the covariance matrix can be helpful to stabilize the inference about the mean vector for a near singular covariance matrix.

The two steps are iterated long enough for the results to be reliable for a multiply imputed data set (Schafer 1997, p. 72). That is, with a current parameter estimate $\boldsymbol{\theta}^{(t)}$ at the tth iteration, the I-step draws $Y_{mis}^{(t+1)}$ from $p(Y_{mis}|Y_{obs}, \boldsymbol{\theta}^{(t)})$ and the P-step draws $\boldsymbol{\theta}^{(t+1)}$ from $p(\boldsymbol{\theta}|Y_{obs}, Y_{mis}^{(t+1)})$.

This creates a Markov chain $(Y_{mis}^{(1)}, \boldsymbol{\theta}^{(1)})$, $(Y_{mis}^{(2)}, \boldsymbol{\theta}^{(2)})$, ..., which converges in distribution to $p(Y_{mis}, \boldsymbol{\theta}|Y_{obs})$. Assuming the iterates converge to a stationary distribution, the goal is to simulate an approximately independent draw of the missing values from this distribution.

To validate the imputation results, you should repeat the process with different random number generators and starting values based on different initial parameter estimates.

The next three sections provide details for the imputation step, Bayesian estimation of the mean vector and covariance matrix, and the posterior step.

Imputation Step

In each iteration, starting with a given mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, the imputation step draws values for the missing data from the conditional distribution Y_{mis} given Y_{obs}.

Suppose $\boldsymbol{\mu} = \left[\boldsymbol{\mu}_1', \boldsymbol{\mu}_2'\right]'$ is the partitioned mean vector of two sets of variables, Y_{obs} and Y_{mis}, where $\boldsymbol{\mu}_1$ is the mean vector for variables Y_{obs} and $\boldsymbol{\mu}_2$ is the mean vector for variables Y_{mis}.

Also suppose

$$\boldsymbol{\Sigma} = \begin{bmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{12}' & \boldsymbol{\Sigma}_{22} \end{bmatrix}$$

is the partitioned covariance matrix for these variables, where $\boldsymbol{\Sigma}_{11}$ is the covariance matrix for variables Y_{obs}, $\boldsymbol{\Sigma}_{22}$ is the covariance matrix for variables Y_{mis}, and $\boldsymbol{\Sigma}_{12}$ is the covariance matrix between variables Y_{obs} and variables Y_{mis}.

By using the sweep operator (Goodnight 1979) on the pivots of the $\mathbf{\Sigma}_{11}$ submatrix, the matrix becomes

$$\begin{bmatrix} \mathbf{\Sigma}_{11}^{-1} & \mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12} \\ -\mathbf{\Sigma}_{12}'\mathbf{\Sigma}_{11}^{-1} & \mathbf{\Sigma}_{22.1} \end{bmatrix}$$

where $\mathbf{\Sigma}_{22.1} = \mathbf{\Sigma}_{22} - \mathbf{\Sigma}_{12}'\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12}$ can be used to compute the conditional covariance matrix of $\mathbf{Y}_{mis}$ after controlling for $\mathbf{Y}_{obs}$.

For an observation with the preceding missing pattern, the conditional distribution of Y_{mis} given $Y_{obs} = \mathbf{y}_1$ is a multivariate normal distribution with the mean vector

$$\boldsymbol{\mu}_{2.1} = \boldsymbol{\mu}_2 + \mathbf{\Sigma}_{12}'\mathbf{\Sigma}_{11}^{-1}(\mathbf{y}_1 - \boldsymbol{\mu}_1)$$

and the conditional covariance matrix

$$\mathbf{\Sigma}_{22.1} = \mathbf{\Sigma}_{22} - \mathbf{\Sigma}_{12}'\mathbf{\Sigma}_{11}^{-1}\mathbf{\Sigma}_{12}$$

Bayesian Estimation of the Mean Vector and Covariance Matrix

Suppose that $\mathbf{Y} = (\mathbf{y}_1', \mathbf{y}_2', \ldots, \mathbf{y}_n')'$ is an $(n \times p)$ matrix made up of n $(p \times 1)$ independent vectors $\mathbf{y}_i$, each of which has a multivariate normal distribution with mean zero and covariance matrix $\mathbf{\Lambda}$. Then the SSCP matrix

$$\mathbf{A} = \mathbf{Y}'\mathbf{Y} = \sum_i \mathbf{y}_i\mathbf{y}_i'$$

has a Wishart distribution $W(n, \mathbf{\Lambda})$.

When each observation $\mathbf{y}_i$ is distributed with a multivariate normal distribution with an unknown mean $\boldsymbol{\mu}$, then the CSSCP matrix

$$\mathbf{A} = \sum_i (\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})'$$

has a Wishart distribution $W(n-1, \mathbf{\Lambda})$.

If $\mathbf{A}$ has a Wishart distribution $W(n, \mathbf{\Lambda})$, then $\mathbf{B} = \mathbf{A}^{-1}$ has an inverted Wishart distribution $W^{-1}(n, \mathbf{\Psi})$, where n is the degrees of freedom and $\mathbf{\Psi} = \mathbf{\Lambda}^{-1}$ is the precision matrix (Anderson 1984).

Note that, instead of using the parameter $\mathbf{\Psi} = \mathbf{\Lambda}^{-1}$ for the inverted Wishart distribution, Schafer (1997) uses the parameter $\mathbf{\Lambda}$.

Suppose that each observation in the data matrix $\mathbf{Y}$ has a multivariate normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\mathbf{\Sigma}$. Then with a prior inverted Wishart distribution for $\mathbf{\Sigma}$ and a prior normal distribution for $\boldsymbol{\mu}$

$$\begin{aligned} \mathbf{\Sigma} &\sim W^{-1}(m, \mathbf{\Psi}) \\ \boldsymbol{\mu}|\mathbf{\Sigma} &\sim N\left(\boldsymbol{\mu}_0, \frac{1}{\tau}\mathbf{\Sigma}\right) \end{aligned}$$

where $\tau > 0$ is a fixed number.

The posterior distribution (Anderson 1984, p. 270; Schafer 1997, p. 152) is

$$\begin{aligned} \boldsymbol{\Sigma}|\mathbf{Y} &\sim W^{-1}\left(n+m,\ (n-1)\mathbf{S}+\boldsymbol{\Psi}+\frac{n\tau}{n+\tau}(\bar{\mathbf{y}}-\boldsymbol{\mu}_0)(\bar{\mathbf{y}}-\boldsymbol{\mu}_0)'\right) \\ \boldsymbol{\mu}|(\boldsymbol{\Sigma},\mathbf{Y}) &\sim N\left(\frac{1}{n+\tau}(n\bar{\mathbf{y}}+\tau\boldsymbol{\mu}_0),\ \frac{1}{n+\tau}\boldsymbol{\Sigma}\right) \end{aligned}$$

where $(n-1)\mathbf{S}$ is the CSSCP matrix.

Posterior Step

In each iteration, the posterior step simulates the posterior population mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ from prior information for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, and the complete sample estimates.

You can specify the prior parameter information by using one of the following methods:

- PRIOR=JEFFREYS, which uses a noninformative prior
- PRIOR=INPUT=, which provides a prior information for $\boldsymbol{\Sigma}$ in the data set. Optionally, it also provides a prior information for $\boldsymbol{\mu}$ in the data set.
- PRIOR=RIDGE=, which uses a ridge prior

The next four subsections provide details of the posterior step for different prior distributions.

1. A Noninformative Prior

Without prior information about the mean and covariance estimates, you can use a noninformative prior by specifying the PRIOR=JEFFREYS option. The posterior distributions (Schafer 1997, p. 154) are

$$\begin{aligned} \boldsymbol{\Sigma}^{(t+1)}|\mathbf{Y} &\sim W^{-1}\left(n-1,\ (n-1)\mathbf{S}\right) \\ \boldsymbol{\mu}^{(t+1)}|(\boldsymbol{\Sigma}^{(t+1)},\mathbf{Y}) &\sim N\left(\bar{\mathbf{y}},\ \frac{1}{n}\boldsymbol{\Sigma}^{(t+1)}\right) \end{aligned}$$

2. An Informative Prior for μ and Σ

When prior information is available for the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, you can provide it with a SAS data set that you specify with the PRIOR=INPUT= option:

$$\begin{aligned} \boldsymbol{\Sigma} &\sim W^{-1}\left(d^*,\ d^*\mathbf{S}^*\right) \\ \boldsymbol{\mu}|\boldsymbol{\Sigma} &\sim N\left(\boldsymbol{\mu}_0,\ \frac{1}{n_0}\boldsymbol{\Sigma}\right) \end{aligned}$$

To obtain the prior distribution for $\boldsymbol{\Sigma}$, PROC MI reads the matrix $\mathbf{S}^*$ from observations in the data set with _TYPE_='COV', and it reads $n^* = d^* + 1$ from observations with _TYPE_='N'.

To obtain the prior distribution for $\boldsymbol{\mu}$, PROC MI reads the mean vector $\boldsymbol{\mu}_0$ from observations with _TYPE_='MEAN', and it reads n_0 from observations with _TYPE_='N_MEAN'. When there are no observations with _TYPE_='N_MEAN', PROC MI reads n_0 from observations with _TYPE_='N'.

The resulting posterior distribution, as described in the section "Bayesian Estimation of the Mean Vector and Covariance Matrix" on page 3782, is given by

$$\begin{aligned} \boldsymbol{\Sigma}^{(t+1)}|\mathbf{Y} &\sim W^{-1}\left(n + d^*,\ (n-1)\mathbf{S} + d^*\mathbf{S}^* + \mathbf{S}_m\right) \\ \boldsymbol{\mu}^{(t+1)} \mid \left(\boldsymbol{\Sigma}^{(t+1)}, \mathbf{Y}\right) &\sim N\left(\frac{1}{n+n_0}(n\bar{\mathbf{y}} + n_0\boldsymbol{\mu}_0),\ \frac{1}{n+n_0}\boldsymbol{\Sigma}^{(t+1)}\right) \end{aligned}$$

where

$$\mathbf{S}_m = \frac{nn_0}{n+n_0}(\bar{\mathbf{y}} - \boldsymbol{\mu}_0)(\bar{\mathbf{y}} - \boldsymbol{\mu}_0)'$$

3. An Informative Prior for Σ

When the sample covariance matrix $\mathbf{S}$ is singular or near singular, prior information about $\boldsymbol{\Sigma}$ can also be used without prior information about $\boldsymbol{\mu}$ to stabilize the inference about $\boldsymbol{\mu}$. You can provide it with a SAS data set that you specify with the PRIOR=INPUT= option.

To obtain the prior distribution for $\boldsymbol{\Sigma}$, PROC MI reads the matrix $\mathbf{S}^*$ from observations in the data set with _TYPE_='COV', and it reads n^* from observations with _TYPE_='N'.

The resulting posterior distribution for $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ (Schafer 1997, p. 156) is

$$\begin{aligned} \boldsymbol{\Sigma}^{(t+1)}|\mathbf{Y} &\sim W^{-1}\left(n + d^*,\ (n-1)\mathbf{S} + d^*\mathbf{S}^*\right) \\ \boldsymbol{\mu}^{(t+1)} \mid \left(\boldsymbol{\Sigma}^{(t+1)}, \mathbf{Y}\right) &\sim N\left(\bar{\mathbf{y}},\ \frac{1}{n}\boldsymbol{\Sigma}^{(t+1)}\right) \end{aligned}$$

Note that if the PRIOR=INPUT= data set also contains observations with _TYPE_='MEAN', then a complete informative prior for both $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ will be used.

4. A Ridge Prior

A special case of the preceding adjustment is a ridge prior with $\mathbf{S}^*$ = Diag $(\mathbf{S})$ (Schafer 1997, p. 156). That is, $\mathbf{S}^*$ is a diagonal matrix with diagonal elements equal to the corresponding elements in $\mathbf{S}$.

You can request a ridge prior by using the PRIOR=RIDGE= option. You can explicitly specify the number $d^* \geq 1$ in the PRIOR=RIDGE=d^* option. Or you can implicitly specify the number by specifying the proportion p in the PRIOR=RIDGE=p option to request $d^* = (n-1)p$.

The posterior is then given by

$$\begin{aligned} \boldsymbol{\Sigma}^{(t+1)}|\mathbf{Y} &\sim W^{-1}\left(n+d^{*},\ (n-1)\mathbf{S}+d^{*}\mathrm{Diag}(\mathbf{S})\right) \\ \boldsymbol{\mu}^{(t+1)}\ \Big|\ \left(\boldsymbol{\Sigma}^{(t+1)},\mathbf{Y}\right) &\sim N\left(\bar{\mathbf{y}},\ \frac{1}{n}\boldsymbol{\Sigma}^{(t+1)}\right) \end{aligned}$$

Producing Monotone Missingness with the MCMC Method

The monotone data MCMC method was first proposed by Li (1988), and Liu (1993) described the algorithm. The method is useful especially when a data set is close to having a monotone missing pattern. In this case, the method needs to impute only a few missing values to the data set to have a monotone missing pattern in the imputed data set. Compared to a full data imputation that imputes all missing values, the monotone data MCMC method imputes fewer missing values in each iteration and achieves approximate stationarity in fewer iterations (Schafer 1997, p. 227).

You can request the monotone MCMC method by specifying the option IMPUTE=MONOTONE in the MCMC statement. The "Missing Data Patterns" table now denotes the variables with missing values by "." or "O". The value "." means that the variable is missing and will be imputed, and the value "O" means that the variable is missing and will not be imputed. The "Variance Information" and "Parameter Estimates" tables are not created.

You must specify the variables in the VAR statement. The variable order in the list determines the monotone missing pattern in the imputed data set. With a different order in the VAR list, the results will be different because the monotone missing pattern to be constructed will be different.

Assuming that the data are from a multivariate normal distribution, then like the MCMC method, the monotone MCMC method repeats the following steps:

1. The imputation I-step
Given an estimated mean vector and covariance matrix, the I-step simulates the missing values for each observation independently. Only a subset of missing values are simulated to achieve a monotone pattern of missingness.

2. The posterior P-step
Given a new sample with a monotone pattern of missingness, the P-step simulates the posterior population mean vector and covariance matrix with a noninformative Jeffreys prior. These new estimates are then used in the next I-step.

Imputation Step

The I-step is almost identical to the I-step described in the section "MCMC Method for Arbitrary Missing Data" on page 3780 except that only a subset of missing values need to be simulated. To state this precisely, denote the variables with observed values for observation i by $Y_{i(obs)}$ and the variables with missing values by $Y_{i(mis)} = (Y_{i(m1)}, Y_{i(m2)})$, where $Y_{i(m1)}$ is a subset of the missing variables that will cause a monotone missingness when their values are imputed. Then the I-step draws values for $Y_{i(m1)}$ from a conditional distribution for $Y_{i(m1)}$ given $Y_{i(obs)}$.

Posterior Step

The P-step is different from the P-step described in the section "MCMC Method for Arbitrary Missing Data" on page 3780. Instead of simulating the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ parameters from the full imputed data set, this P-step simulates the $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ parameters through simulated regression coefficients from regression models based on the imputed data set with a monotone pattern of missingness. The step is similar to the process described in the section "Regression Method for Monotone Missing Data" on page 3774.

That is, for the variable Y_j, a model

$$Y_j = \beta_0 + \beta_1 Y_1 + \beta_2 Y_2 + \ldots + \beta_{j-1} Y_{j-1}$$

is fitted using n_j nonmissing observations for variable Y_j in the imputed data sets.

The fitted model consists of the regression parameter estimates $\hat{\boldsymbol{\beta}} = (\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_{j-1})$ and the associated covariance matrix $\hat{\sigma}_j^2 \mathbf{V}_j$, where $\mathbf{V}_j$ is the usual $\mathbf{X}'\mathbf{X}$ inverse matrix from the intercept and variables $Y_1, Y_2, \ldots, Y_{j-1}$.

For each imputation, new parameters $\boldsymbol{\beta}_* = (\beta_{*0}, \beta_{*1}, \ldots, \beta_{*(j-1)})$ and σ_{*j}^2 are drawn from the posterior predictive distribution of the parameters. That is, they are simulated from $(\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_{j-1})$, σ_j^2, and $\mathbf{V}_j$. The variance is drawn as

$$\sigma_{*j}^2 = \hat{\sigma}_j^2 (n_j - j)/g$$

where g is a $\chi^2_{n_j - p + j - 1}$ random variate and n_j is the number of nonmissing observations for Y_j. The regression coefficients are drawn as

$$\boldsymbol{\beta}_* = \hat{\boldsymbol{\beta}} + \sigma_{*j} \mathbf{V}_{hj}' \mathbf{Z}$$

where $\mathbf{V}_{hj}'$ is the upper triangular matrix in the Cholesky decomposition, $\mathbf{V}_j = \mathbf{V}_{hj}' \mathbf{V}_{hj}$, and $\mathbf{Z}$ is a vector of j independent random normal variates.

These simulated values of $\boldsymbol{\beta}_*$ and σ_{*j}^2 are then used to re-create the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$. For a detailed description of how to produce monotone missingness with the MCMC method for a multivariate normal data, see Schafer (1997, pp. 226–235).

MCMC Method Specifications

With the MCMC method, you can impute either all missing values (IMPUTE=FULL) or just enough missing values to make the imputed data set have a monotone missing pattern (IMPUTE=MONOTONE). In the process, either a single chain for all imputations (CHAIN=SINGLE) or a separate chain for each imputation (CHAIN=MULTIPLE) is used. The single chain might be somewhat more precise for estimating a single quantity such as a posterior mean (Schafer 1997, p. 138). See Schafer (1997, pp. 137–138) for a discussion of single versus multiple chains.

You can specify the number of initial burn-in iterations before the first imputation with the NBITER= option. This number is also used for subsequent chains for multiple chains. For a single chain, you can also specify the number of iterations between imputations with the NITER= option.

You can explicitly specify initial parameter values for the MCMC method with the INITIAL=INPUT= data set option. Alternatively, you can use the EM algorithm to derive a set of initial parameter values for MCMC with the option INITIAL=EM. These estimates are used as either the starting value (START=VALUE) or the starting distribution (START=DIST) for the MCMC method. For multiple chains, these estimates are used again as either the starting value (START=VALUE) or the starting distribution (START=DIST) for the subsequent chains.

You can specify the prior parameter information in the PRIOR= option. You can use a noninformative prior (PRIOR=JEFFREYS), a ridge prior (PRIOR=RIDGE), or an informative prior specified in a data set (PRIOR=INPUT).

The parameter estimates used to generate imputed values in each imputation can be saved in a data set with the OUTEST= option. Later, this data set can be read with the INEST= option to provide the reference distribution for imputing missing values for a new data set.

By default, the MCMC method uses a single chain to produce five imputations. It completes 200 burn-in iterations before the first imputation and 100 iterations between imputations. The posterior mode computed from the EM algorithm with a noninformative prior is used as the starting values for the MCMC method.

INITIAL=EM Specifications

The EM algorithm is used to find the maximum likelihood estimates for incomplete data in the EM statement. You can also use the EM algorithm to find a posterior mode, the parameter estimates that maximize the observed-data posterior density. The resulting posterior mode provides a good starting value for the MCMC method.

With the INITIAL=EM option, PROC MI uses the MLE of the parameter vector as the initial estimates in the EM algorithm for the posterior mode. You can use the ITPRINT option within the INITIAL=EM option to display the iteration history for the EM algorithm.

You can use the CONVERGE= option to specify the convergence criterion in deriving the EM posterior mode. The iterations are considered to have converged when the maximum change in the parameter estimates between iteration steps is less than the value specified. By default, CONVERGE=1E−4.

You can also use the MAXITER= option to specify the maximum number of iterations of the EM algorithm. By default, MAXITER=200.

With the BOOTSTRAP option, you can use overdispersed starting values for the MCMC method. In this case, PROC MI applies the EM algorithm to a bootstrap sample, a simple random sample with replacement from the input data set, to derive the initial estimates for each chain (Schafer 1997, p. 128).

Checking Convergence in MCMC

The theoretical convergence of the MCMC method has been explored under various conditions, as described in Schafer (1997, p. 70). However, in practice, verification of convergence is not a simple matter.

The parameters used in the imputation step for each iteration can be saved in an output data set with the OUTITER= option. These include the means, standard deviations, covariances, worst linear function, and observed-data LR statistics. You can then monitor the convergence in a single chain by displaying trace plots and autocorrelations for those parameter values (Schafer 1997, p. 120). The trace and autocorrelation function plots for parameters such as variable means, covariances, and the worst linear function can be displayed by specifying the TIMEPLOT and ACFPLOT option.

You can apply the EM algorithm to a bootstrap sample to obtain overdispersed starting values for multiple chains (Gelman and Rubin 1992). This provides a conservative estimate of the number of iterations needed before each imputation.

The next four subsections describe useful statistics and plots that can be used to check the convergence of the MCMC method.

LR Statistics

You can save the observed-data likelihood ratio (LR) statistic in each iteration with the LR option in the OUTITER= data set. The statistic is based on the observed-data likelihood with parameter values used in the iteration and the observed-data maximum likelihood derived from the EM algorithm.

In each iteration, the LR statistic is given by

$$-2\log\left(\frac{f(\hat{\boldsymbol{\theta}}_i)}{f(\hat{\boldsymbol{\theta}})}\right)$$

where $f(\hat{\boldsymbol{\theta}})$ is the observed-data maximum likelihood derived from the EM algorithm and $f(\hat{\boldsymbol{\theta}}_i)$ is the observed-data likelihood for $\hat{\boldsymbol{\theta}}_i$ used in the iteration.

Similarly, you can also save the observed-data LR posterior mode statistic for each iteration with the LR_POST option. This statistic is based on the observed-data posterior density with parameter values used in each iteration and the observed-data posterior mode derived from the EM algorithm for posterior mode.

For large samples, these LR statistics tends to be approximately χ^2 distributed with degrees of freedom equal to the dimension of $\boldsymbol{\theta}$ (Schafer 1997, p. 131). For example, with a large number of iterations, if the values of the LR statistic do not behave like a random sample from the described χ^2 distribution, then there is evidence that the MCMC method has not converged.

Worst Linear Function of Parameters

The worst linear function (WLF) of parameters (Schafer 1997, pp. 129–131) is a scalar function of parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ that is "worst" in the sense that its function values converge most slowly among parameters in the MCMC method. The convergence of this function is evidence that other parameters are likely to converge as well.

For linear functions of parameters $\boldsymbol{\theta} = (\boldsymbol{\mu}, \boldsymbol{\Sigma})$, a worst linear function of $\boldsymbol{\theta}$ has the highest asymptotic rate of missing information. The function can be derived from the iterative values of $\boldsymbol{\theta}$ near the posterior mode in the EM algorithm. That is, an estimated worst linear function of $\boldsymbol{\theta}$ is

$$w(\boldsymbol{\theta}) = \mathbf{v}'(\boldsymbol{\theta} - \hat{\boldsymbol{\theta}})$$

where $\hat{\boldsymbol{\theta}}$ is the posterior mode and the coefficients $\mathbf{v} = \hat{\boldsymbol{\theta}}_{(-1)} - \hat{\boldsymbol{\theta}}$ are the difference between the estimated value of $\boldsymbol{\theta}$ one step prior to convergence and the converged value $\hat{\boldsymbol{\theta}}$.

You can display the coefficients of the worst linear function, $\mathbf{v}$, by specifying the WLF option in the MCMC statement. You can save the function value from each iteration in an OUTITER= data set by specifying the WLF option within the OUTITER option. You can also display the worst linear function values from iterations in an autocorrelation plot or a trace plot by specifying WLF as an ACFPLOT or TIMEPLOT option, respectively.

Note that when the observed-data posterior is nearly normal, the WLF is one of the slowest functions to approach stationarity. When the posterior is not close to normal, other functions might take much longer than the WLF to converge, as described in Schafer (1997, p. 130).

Trace Plot

A trace plot for a parameter ξ is a scatter plot of successive parameter estimates ξ_i against the iteration number i. The plot provides a simple way to examine the convergence behavior of the estimation algorithm for ξ. Long-term trends in the plot indicate that successive iterations are highly correlated and that the series of iterations has not converged.

You can display trace plots for worst linear function, variable means, variable variances, and covariances of variables. You can also request logarithmic transformations for positive parameters in the plots with the LOG option. When a parameter value is less than or equal to zero, the value is not displayed in the corresponding plot.

By default, the MI procedure uses solid line segments to connect data points in a trace plot. You can use the CCONNECT=, LCONNECT=, and WCONNECT= options to change the color, line type, and width of the line segments, respectively. When WCONNECT=0 is specified, the data points are not connected, and the procedure uses the plus sign (+) as the plot symbol to display the points with a height of one (percentage screen unit) in a trace plot. You can use the SYMBOL=, CSYMBOL=, and HSYMBOL= options to change the shape, color, and height of the plot symbol, respectively.

By default, the plot title "Trace Plot" is displayed in a trace plot. You can request another title by using the TITLE= option in the TIMEPLOT option. When another title is also specified in a TITLE statement, this title is displayed as the main title and the plot title is displayed as a subtitle in the plot.

You can use options in the GOPTIONS statement to change the color and height of the title. See the chapter "The SAS/GRAPH Statements" in *SAS/GRAPH Software: Reference* for an illustration of title options. See Example 54.8 for a usage of the trace plot.

Autocorrelation Function Plot

To examine relationships of successive parameter estimates ξ, the autocorrelation function (ACF) can be used. For a stationary series, $\xi_i, i \geq 1$, in trace data, the autocorrelation function at lag k is

$$\rho_k = \frac{\mathrm{Cov}(\xi_i, \xi_{i+k})}{\mathrm{Var}(\xi_i)}$$

The sample kth order autocorrelation is computed as

$$r_k = \frac{\sum_{i=1}^{n-k}(\xi_i - \bar{\xi})(\xi_{i+k} - \bar{\xi})}{\sum_{i=1}^{n}(\xi_i - \bar{\xi})^2}$$

You can display autocorrelation function plots for the worst linear function, variable means, variable variances, and covariances of variables. You can also request logarithmic transformations for parameters in the plots with the LOG option. When a parameter has values less than or equal to zero, the corresponding plot is not created.

You specify the maximum number of lags of the series with the NLAG= option. The autocorrelations at each lag less than or equal to the specified lag are displayed in the graph. In addition, the plot also displays approximate 95% confidence limits for the autocorrelations. At lag k, the confidence limits indicate a set of approximate 95% critical values for testing the hypothesis $\rho_j = 0, j \geq k$.

By default, the MI procedure uses the star (*) as the plot symbol to display the points with a height of one (percentage screen unit) in the plot, a solid line to display the reference line of zero autocorrelation, vertical line segments to connect autocorrelations to the reference line, and a pair of dashed lines to display approximately 95% confidence limits for the autocorrelations.

You can use the SYMBOL=, CSYMBOL=, and HSYMBOL= options to change the shape, color, and height of the plot symbol, respectively, and the CNEEDLES= and WNEEDLES= options to change the color and width of the needles, respectively. You can also use the LREF=, CREF=, and WREF= options to change the line type, color, and width of the reference line, respectively. Similarly, you can use the LCONF=, CCONF=, and WCONF= options to change the line type, color, and width of the confidence limits, respectively.

By default, the plot title "Autocorrelation Plot" is displayed in a autocorrelation function plot. You can request another title by using the TITLE= option within the ACFPLOT option. When another title is also specified in a TITLE statement, this title is displayed as the main title and the plot title is displayed as a subtitle in the plot.

You can use options in the GOPTIONS statement to change the color and height of the title. See the chapter "The SAS/GRAPH Statements" in *SAS/GRAPH Software: Reference* for a description of title options. See Example 54.8 for an illustration of the autocorrelation function plot.

Input Data Sets

You can specify the input data set with missing values by using the DATA= option in the PROC MI statement. When an MCMC method is used, you can specify the data set containing the reference distribution information for imputation with the INEST= option, the data set containing initial parameter estimates for the MCMC method with the INITIAL=INPUT= option, and the data set containing information for the prior distribution with the PRIOR=INPUT= option in the MCMC statement.

DATA=*SAS-data-set*

The input DATA= data set is an ordinary SAS data set containing multivariate data with missing values.

INEST=*SAS-data-set*

The input INEST= data set is a TYPE=EST data set and contains a variable _Imputation_ to identify the imputation number. For each imputation, PROC MI reads the point estimate from the observations with _TYPE_='PARM' or _TYPE_='PARMS' and the associated covariances from the observations with _TYPE_='COV' or _TYPE_='COVB'. These estimates are used as the reference distribution to impute values for observations in the DATA= data set. When the input INEST= data set also contains observations with _TYPE_='SEED', PROC MI reads the seed information for the random number generator from these observations. Otherwise, the SEED= option provides the seed information.

INITIAL=INPUT=*SAS-data-set*

The input INITIAL=INPUT= data set is a TYPE=COV or CORR data set and provides initial parameter estimates for the MCMC method. The covariances derived from the TYPE=COV/CORR data set are divided by the number of observations to get the correct covariance matrix for the point estimate (sample mean).

If TYPE=COV, PROC MI reads the number of observations from the observations with _TYPE_='N', the point estimate from the observations with _TYPE_='MEAN', and the covariances from the observations with _TYPE_='COV'.

If TYPE=CORR, PROC MI reads the number of observations from the observations with _TYPE_='N', the point estimate from the observations with _TYPE_='MEAN', the correlations from the observations with _TYPE_='CORR', and the standard deviations from the observations with _TYPE_='STD'.

PRIOR=INPUT=*SAS-data-set*

The input PRIOR=INPUT= data set is a TYPE=COV data set that provides information for the prior distribution. You can use the data set to specify a prior distribution for $\boldsymbol{\Sigma}$ of the form

$$\boldsymbol{\Sigma} \sim W^{-1}\left(d^*,\, d^*\mathbf{S}^*\right)$$

where $d^* = n^* - 1$ is the degrees of freedom. PROC MI reads the matrix $\mathbf{S}^*$ from observations with _TYPE_='COV' and reads n^* from observations with _TYPE_='N'.

You can also use this data set to specify a prior distribution for $\boldsymbol{\mu}$ of the form

$$\boldsymbol{\mu} \sim N\left(\boldsymbol{\mu}_0,\, \frac{1}{n_0}\boldsymbol{\Sigma}\right)$$

PROC MI reads the mean vector $\boldsymbol{\mu}_0$ from observations with _TYPE_='MEAN' and reads n_0 from observations with _TYPE_='N_MEAN'. When there are no observations with _TYPE_='N_MEAN', PROC MI reads n_0 from observations with _TYPE_='N'.

Output Data Sets

You can specify the output data set of imputed values with the OUT= option in the PROC MI statement. When an EM statement is used, you can specify the data set containing the original data set with missing values being replaced by the expected values from the EM algorithm by using the OUT= option in the EM statement. You can also specify the data set containing MLE computed with the EM algorithm by using the OUTEM= option.

When an MCMC method is used, you can specify the data set containing parameter estimates used in each imputation with the OUTEST= option in the MCMC statement, and you can specify the data set containing parameters used in the imputation step for each iteration with the OUTITER option in the MCMC statement.

OUT=*SAS-data-set* in the PROC MI statement

The OUT= data set contains all the variables in the original data set and a new variable named _Imputation_ that identifies the imputation. For each imputation, the data set contains all variables in the input DATA= data set with missing values being replaced by imputed values. Note that when the NIMPUTE=1 option is specified, the variable _Imputation_ is not created.

OUT=*SAS-data-set* in an EM statement

The OUT= data set contains the original data set with missing values being replaced by expected values from the EM algorithm.

OUTEM=*SAS-data-set*

The OUTEM= data set is a TYPE=COV data set and contains the MLE computed with the EM algorithm. The observations with _TYPE_='MEAN' contain the estimated mean and the observations with _TYPE_='COV' contain the estimated covariances.

OUTEST=*SAS-data-set*

The OUTEST= data set is a TYPE=EST data set and contains parameter estimates used in each imputation in the MCMC method. It also includes an index variable named _Imputation_, which identifies the imputation.

The observations with _TYPE_='SEED' contain the seed information for the random number generator. The observations with _TYPE_='PARM' or _TYPE_='PARMS' contain the point estimate, and the observations with _TYPE_='COV' or _TYPE_='COVB' contain the associated covariances. These estimates are used as the parameters of the reference distribution to impute values for observations in the DATA= dataset.

Note that these estimates are the values used in the I-step before each imputation. These are not the parameter values simulated from the P-step in the same iteration. See Example 54.9 for a usage of this option.

OUTITER < (*options*) > =*SAS-data-set* in an EM statement

The OUTITER= data set in an EM statement is a TYPE=COV data set and contains parameters for each iteration. It also includes a variable _Iteration_ that provides the iteration number.

The parameters in the output data set depend on the options specified. You can specify the MEAN and COV options for OUTITER. With the MEAN option, the output data set contains the mean parameters in observations with the variable _TYPE_='MEAN'. Similarly, with the MEAN option, the output data set contains the covariance parameters in observations with the variable _TYPE_='COV'. When no options are specified, the output data set contains the mean parameters for each iteration.

OUTITER < (*options*) > =*SAS-data-set* in an MCMC statement

The OUTITER= data set in an MCMC statement is a TYPE=COV data set and contains parameters used in the imputation step for each iteration. It also includes variables named _Imputation_ and _Iteration_, which provide the imputation number and iteration number.

The parameters in the output data set depend on the options specified. Table 54.4 summarizes the options available for OUTITER and the corresponding values for the output variable _TYPE_.

Table 54.4 Summary of Options for OUTITER in an MCMC statement

Option	Output Parameters	_TYPE_
MEAN	mean parameters	MEAN
STD	standard deviations	STD

Table 54.4 *continued*

Option	Output Parameters	_TYPE_
COV	covariances	COV
LR	−2 log LR statistic	LOG_LR
LR_POST	−2 log LR statistic of the posterior mode	LOG_POST
WLF	worst linear function	WLF

When no options are specified, the output data set contains the mean parameters used in the imputation step for each iteration. For a detailed description of the worst linear function and LR statistics, see the section "Checking Convergence in MCMC" on page 3788.

Combining Inferences from Multiply Imputed Data Sets

With m imputations, m different sets of the point and variance estimates for a parameter Q can be computed. Suppose $\hat{Q}_i$ and $\hat{W}_i$ are the point and variance estimates from the ith imputed data set, $i = 1, 2, \ldots, m$. Then the combined point estimate for Q from multiple imputation is the average of the m complete-data estimates:

$$\overline{Q} = \frac{1}{m}\sum_{i=1}^{m}\hat{Q}_i$$

Suppose $\overline{W}$ is the within-imputation variance, which is the average of the m complete-data estimates,

$$\overline{W} = \frac{1}{m}\sum_{i=1}^{m}\hat{W}_i$$

and B is the between-imputation variance

$$B = \frac{1}{m-1}\sum_{i=1}^{m}(\hat{Q}_i - \overline{Q})^2$$

Then the variance estimate associated with $\overline{Q}$ is the total variance (Rubin 1987)

$$T = \overline{W} + (1 + \frac{1}{m})B$$

The statistic $(Q - \overline{Q})T^{-(1/2)}$ is approximately distributed as t with v_m degrees of freedom (Rubin 1987), where

$$v_m = (m-1)\left[1 + \frac{\overline{W}}{(1+m^{-1})B}\right]^2$$

The degrees of freedom v_m depend on m and the ratio

$$r = \frac{(1 + m^{-1})B}{\overline{W}}$$

The ratio r is called the relative increase in variance due to nonresponse (Rubin 1987). When there is no missing information about Q, the values of r and B are both zero. With a large value of m or a small value of r, the degrees of freedom v_m will be large and the distribution of $(Q - \overline{Q})T^{-(1/2)}$ will be approximately normal.

Another useful statistic is the fraction of missing information about Q:

$$\hat{\lambda} = \frac{r + 2/(v_m + 3)}{r + 1}$$

Both statistics r and λ are helpful diagnostics for assessing how the missing data contribute to the uncertainty about Q.

When the complete-data degrees of freedom v_0 are small, and there is only a modest proportion of missing data, the computed degrees of freedom, v_m, can be much larger than v_0, which is inappropriate. For example, with $m = 5$ and $r = 10\%$, the computed degrees of freedom $v_m = 484$, which is inappropriate for data sets with complete-data degrees of freedom less than 484.

Barnard and Rubin (1999) recommend the use of adjusted degrees of freedom

$$v_m^* = \left[\frac{1}{v_m} + \frac{1}{\hat{v}_{obs}}\right]^{-1}$$

where $\hat{v}_{obs} = (1 - \gamma)\, v_0(v_0 + 1)/(v_0 + 3)$ and $\gamma = (1 + m^{-1})B/T$.

Note that the MI procedure uses the adjusted degrees of freedom, v_m^*, for inference.

Multiple Imputation Efficiency

The relative efficiency (RE) of using the finite m imputation estimator, rather than using an infinite number for the fully efficient imputation, in units of variance, is approximately a function of m and λ (Rubin 1987, p. 114):

$$RE = \left(1 + \frac{\lambda}{m}\right)^{-1}$$

Table 54.5 shows relative efficiencies with different values of m and λ.

Table 54.5 Relative Efficiencies

	λ				
m	10%	20%	30%	50%	70%
3	0.9677	0.9375	0.9091	0.8571	0.8108
5	0.9804	0.9615	0.9434	0.9091	0.8772
10	0.9901	0.9804	0.9709	0.9524	0.9346
20	0.9950	0.9901	0.9852	0.9756	0.9662

The table shows that for situations with little missing information, only a small number of imputations are necessary. In practice, the number of imputations needed can be informally verified by replicating sets of m imputations and checking whether the estimates are stable between sets (Horton and Lipsitz 2001, p. 246).

Imputer's Model Versus Analyst's Model

Multiple imputation inference assumes that the model you used to analyze the multiply imputed data (the analyst's model) is the same as the model used to impute missing values in multiple imputation (the imputer's model). But in practice, the two models might not be the same (Schafer 1997, p. 139).

Schafer (1997, pp. 139–143) provides comprehensive coverage of this topic, and the following example is based on his work.

Consider a trivariate data set with variables Y_1 and Y_2 fully observed, and a variable Y_3 with missing values. An imputer creates multiple imputations with the model $Y_3 = Y_1\ Y_2$. However, the analyst can later use the simpler model $Y_3 = Y_1$. In this case, the analyst assumes more than the imputer. That is, the analyst assumes there is no relationship between variables Y_3 and Y_2.

The effect of the discrepancy between the models depends on whether the analyst's additional assumption is true. If the assumption is true, the imputer's model still applies. The inferences derived from multiple imputations will still be valid, although they might be somewhat conservative because they reflect the additional uncertainty of estimating the relationship between Y_3 and Y_2.

On the other hand, suppose that the analyst models $Y_3 = Y_1$, and there is a relationship between variables Y_3 and Y_2. Then the model $Y_3 = Y_1$ will be biased and is inappropriate. Appropriate results can be generated only from appropriate analyst models.

Another type of discrepancy occurs when the imputer assumes more than the analyst. For example, suppose that an imputer creates multiple imputations with the model $Y_3 = Y_1$, but the analyst later fits a model $Y_3 = Y_1\ Y_2$. When the assumption is true, the imputer's model is a correct model and the inferences still hold.

On the other hand, suppose there is a relationship between Y_3 and Y_2. Imputations created under the incorrect assumption that there is no relationship between Y_3 and Y_2 will make the analyst's

estimate of the relationship biased toward zero. Multiple imputations created under an incorrect model can lead to incorrect conclusions.

Thus, generally you should include as many variables as you can when doing multiple imputation. The precision you lose with included unimportant predictors is usually a relatively small price to pay for the general validity of analyses of the resultant multiply imputed data set (Rubin 1996). But at the same time, you need to keep the model building and fitting feasible (Barnard and Meng, 1999, pp. 19–20).

To produce high-quality imputations for a particular variable, the imputation model should also include variables that are potentially related to the imputed variable and variables that are potentially related to the missingness of the imputed variable (Schafer 1997, p. 143).

Similar suggestions were also given by van Buuren, Boshuizen, and Knook (1999, p. 687). They recommend that the imputation model include three sets of covariates: variables in the analyst's model, variables associated with the missingness of the imputed variable, and variables correlated with the imputed variable. They also recommend the removal of the covariates not in the analyst's model if they have too many missing values for observations with missing imputed variables.

Note that it is good practice to include a description of the imputer's model with the multiply imputed data set (Rubin 1996, p. 479). That way, the analysts will have information about the variables involved in the imputation and which relationships among the variables have been implicitly set to zero.

Parameter Simulation versus Multiple Imputation

As an alternative to multiple imputation, parameter simulation can also be used to analyze the data for many incomplete-data problems. Although the MI procedure does not offer parameter simulation, the trade-offs between the two methods (Schafer 1997, pp. 89–90, 135–136) are examined in this section.

The parameter simulation method simulates random values of parameters from the observed-data posterior distribution and makes simple inferences about these parameters (Schafer 1997, p. 89). When a set of well-defined population parameters $\boldsymbol{\theta}$ are of interest, parameter simulation can be used to directly examine and summarize simulated values of $\boldsymbol{\theta}$. This usually requires a large number of iterations, and involves calculating appropriate summaries of the resulting dependent sample of the iterates of the $\boldsymbol{\theta}$. If only a small set of parameters are involved, parameter simulation is suitable (Schafer 1997).

Multiple imputation requires only a small number of imputations. Generating and storing a few imputations can be more efficient than generating and storing a large number of iterations for parameter simulation.

When fractions of missing information are low, methods that average over simulated values of the missing data, as in multiple imputation, can be much more efficient than methods that average over simulated values of $\boldsymbol{\theta}$ as in parameter simulation (Schafer 1997).

Summary of Issues in Multiple Imputation

This section summarizes issues that are encountered in applications of the MI procedure.

The MAR Assumption

The missing at random (MAR) assumption is needed for the imputation methods in the MI procedure. Although this assumption cannot be verified with the data, it becomes more plausible as more variables are included in the imputation model (Schafer 1997, pp. 27–28; van Buuren, Boshuizen, and Knook 1999, p. 687).

Number of Imputations

Based on the theory of multiple imputation, only a small number of imputations are needed for a data set with little missing information (Rubin 1987, p. 114). The number of imputations can be informally verified by replicating sets of m imputations and checking whether the estimates are stable (Horton and Lipsitz 2001, p. 246).

Imputation Model

Generally you should include as many variables as you can in the imputation model (Rubin 1996), At the same time, however, it is important to keep the number of variables in control, as discussed by Barnard and Meng (1999, pp. 19–20). For the imputation of a particular variable, the model should include variables in the complete-data model, variables that are correlated with the imputed variable, and variables that are associated with the missingness of the imputed variable (Schafer 1997, p. 143; van Buuren, Boshuizen, and Knook 1999, p. 687).

Multivariate Normality Assumption

Although the regression and MCMC methods assume multivariate normality, inferences based on multiple imputation can be robust to departures from the multivariate normality if the amount of missing information is not large (Schafer 1997, pp. 147–148).

You can use variable transformations to make the normality assumption more tenable. Variables are transformed before the imputation process and then back-transformed to create imputed values.

Monotone Regression Method

With the multivariate normality assumption, either the regression method or the predictive mean matching method can be used to impute continuous variables in data sets with monotone missing patterns.

The predictive mean matching method ensures that imputed values are plausible and might be more appropriate than the regression method if the normality assumption is violated (Horton and Lipsitz 2001, p. 246).

Monotone Propensity Score Method

The propensity score method can also be used to impute continuous variables in data sets with monotone missing patterns.

The propensity score method does not use correlations among variables and is not appropriate for analyses involving relationship among variables, such as a regression analysis (Schafer 1999, p. 11). It can also produce badly biased estimates of regression coefficients when data on predictor variables are missing (Allison 2000).

MCMC Monotone-Data Imputation

The MCMC method is used to impute continuous variables in data sets with arbitrary missing patterns, assuming a multivariate normal distribution for the data. It can also be used to impute just enough missing values to make the imputed data sets have a monotone missing pattern. Then, a more flexible monotone imputation method can be used for the remaining missing values.

Checking Convergence in MCMC

In an MCMC method, parameters are drawn after the MCMC is run long enough to converge to its stationary distribution. In practice, however, it is not simple to verify the convergence of the process, especially for a large number of parameters.

You can check for convergence by examining the observed-data likelihood ratio statistic and worst linear function of the parameters in each iteration. You can also check for convergence by examining a plot of autocorrelation function, as well as a trace plot of parameters (Schafer 1997, p. 120).

EM Estimates

The EM algorithm can be used to compute the MLE of the mean vector and covariance matrix of the data with missing values, assuming a multivariate normal distribution for the data. However, the covariance matrix associated with the estimate of the mean vector cannot be derived from the EM algorithm.

In the MI procedure, you can use the EM algorithm to compute the posterior mode, which provides a good starting value for the MCMC method (Schafer 1997, p. 169).

ODS Table Names

PROC MI assigns a name to each table it creates. You must use these names to reference tables when using the Output Delivery System (ODS). These names are listed in Table 54.6. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 54.6 ODS Tables Produced by PROC MI

ODS Table Name	Description	Statement	Option
Corr	Pairwise correlations		SIMPLE
EMEstimates	EM (MLE) estimates	EM	
EMInitEstimates	EM initial estimates	EM	
EMIterHistory	EM (MLE) iteration history	EM	ITPRINT
EMPostEstimates	EM (posterior mode) estimates	MCMC	INITIAL=EM
EMPostIterHistory	EM (posterior mode) iteration history	MCMC	INITIAL=EM (ITPRINT)
EMWLF	Worst linear function	MCMC	WLF
MCMCInitEstimates	MCMC initial estimates	MCMC	DISPLAYINIT
MissPattern	Missing data patterns		
ModelInfo	Model information		
MonoDiscrim	Discriminant model group means	MONOTONE	DISCRIM (/DETAILS)
MonoLogistic	Logistic model	MONOTONE	LOGISTIC (/DETAILS)
MonoModel	Multiple monotone models	MONOTONE	
MonoPropensity	Propensity score model logistic function	MONOTONE	PROPENSITY (/DETAILS)
MonoReg	Regression model	MONOTONE	REG (/DETAILS)
MonoRegPMM	Predicted mean matching model	MONOTONE	REGPMM (/DETAILS)
ParameterEstimates	Parameter estimates		
Transform	Variable transformations	TRANSFORM	
Univariate	Univariate statistics		SIMPLE
VarianceInfo	Between, within, and total variances		

ODS Graphics

PROC MI assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 54.7.

To request these graphs, you must specify the **ods graphics on** statement in addition to the options indicated in Table 54.7. For more information about the **ods graphics** statement, see Chapter 21, "Statistical Graphics Using ODS."

Table 54.7 ODS Graphics Produced by PROC MI

ODS Graph Name	Plot Description	Statement	Option
ACFPlot	ACF plot	MCMC	PLOTS=ACF
TracePlot	Trace plot	MCMC	PLOTS= TRACE

Examples: MI Procedure

The Fish data described in the STEPDISC procedure are measurements of 159 fish of seven species caught in Finland's lake Laengelmavesi. For each fish, the length, height, and width are measured. Three different length measurements are recorded: from the nose of the fish to the beginning of its tail (Length1), from the nose to the notch of its tail (Length2), and from the nose to the end of its tail (Length3). See Chapter 82, "The STEPDISC Procedure," for more information.

The Fish1 data set is constructed from the Fish data set and contains only one species of the fish and the three length measurements. Some values have been set to missing, and the resulting data set has a monotone missing pattern in the variables Length1, Length2, and Length3. The Fish1 data set is used in Example 54.2 with the propensity score method and in Example 54.3 with the regression method.

The Fish2 data set is also constructed from the Fish data set and contains two species of fish. Some values have been set to missing, and the resulting data set has a monotone missing pattern in the variables Length3, Height, Width, and Species. The Fish2 data set is used in Example 54.4 with the logistic regression method and in Example 54.5 with the discriminant function method. Note that some values of the variable Species have also been altered in the data set.

The FitMiss data set created in the section "Getting Started: MI Procedure" on page 3746 is used in other examples. The following statements create the Fish1 data set:

```
/*----------- Fish of Species Bream ----------*/
 data Fish1;
    title 'Fish Measurement Data';
    input Length1 Length2 Length3 @@;
    datalines;
 23.2 25.4 30.0    24.0 26.3 31.2    23.9 26.5 31.1
 26.3 29.0 33.5    26.5 29.0   .     26.8 29.7 34.7
 26.8   .    .     27.6 30.0 35.0    27.6 30.0 35.1
 28.5 30.7 36.2    28.4 31.0 36.2    28.7   .    .
 29.1 31.5   .     29.5 32.0 37.3    29.4 32.0 37.2
 29.4 32.0 37.2    30.4 33.0 38.3    30.4 33.0 38.5
 30.9 33.5 38.6    31.0 33.5 38.7    31.3 34.0 39.5
 31.4 34.0 39.2    31.5 34.5   .     31.8 35.0 40.6
 31.9 35.0 40.5    31.8 35.0 40.9    32.0 35.0 40.6
 32.7 36.0 41.5    32.8 36.0 41.6    33.5 37.0 42.6
 35.0 38.5 44.1    35.0 38.5 44.0    36.2 39.5 45.3
 37.4 41.0 45.9    38.0 41.0 46.5
 ;
```

The Fish2 data set contains two of the seven species in the Fish data set. For each of the two species (Bream and Roach), the length from the nose of the fish to the end of its tail, the height, and the width of each fish are measured. The height and width are recorded as percentages of the length variable.

The following statements create the Fish2 data set:

```
/*-------- Fish of Species Bream and Roach --------*/
 data Fish2 (drop=HtPct WidthPct);
 title 'Fish Measurement Data';
 input Species $ Length3 HtPct WidthPct @@;
 Height= HtPct*Length3/100;
 Width= WidthPct*Length3/100;
 datalines;
 Gp1  30.0 38.4 13.4   Gp1  31.2 40.0 13.8   Gp1  31.1 39.8 15.1
   .  33.5 38.0   .      .  34.0 36.6 15.1   Gp1  34.7 39.2 14.2
 Gp1  34.5 41.1 15.3   Gp1  35.0 36.2 13.4   Gp1  35.1 39.9 13.8
   .  36.2 39.3 13.7   Gp1  36.2 39.4 14.1     .  36.2 39.7 13.3
 Gp1  36.4 37.8 12.0     .  37.3 37.3 13.6   Gp1  37.2 40.2 13.9
 Gp1  37.2 41.5 15.0   Gp1  38.3 38.8 13.8   Gp1  38.5 38.8 13.5
 Gp1  38.6 40.5 13.3   Gp1  38.7 37.4 14.8   Gp1  39.5 38.3 14.1
 Gp1  39.2 40.8 13.7     .  39.7 39.1   .    Gp1  40.6 38.1 15.1
 Gp1  40.5 40.1 13.8   Gp1  40.9 40.0 14.8   Gp1  40.6 40.3 15.0
 Gp1  41.5 39.8 14.1   Gp2  41.6 40.6 14.9   Gp1  42.6 44.5 15.5
 Gp1  44.1 40.9 14.3   Gp1  44.0 41.1 14.3   Gp1  45.3 41.4 14.9
 Gp1  45.9 40.6 14.7   Gp1  46.5 37.9 13.7
 Gp2  16.2 25.6 14.0   Gp2  20.3 26.1 13.9   Gp2  21.2 26.3 13.7
 Gp2  22.2 25.3 14.3   Gp2  22.2 28.0 16.1   Gp2  22.8 28.4 14.7
 Gp2  23.1 26.7 14.7     .  23.7 25.8 13.9   Gp2  24.7 23.5 15.2
 Gp2  24.3 27.3 14.6   Gp2  25.3 27.8 15.1   Gp2  25.0 26.2 13.3
 Gp2  25.0 25.6 15.2   Gp2  27.2 27.7 14.1   Gp2  26.7 25.9 13.6
   .  26.8 27.6 15.4   Gp2  27.9 25.4 14.0   Gp2  29.2 30.4 15.4
 Gp2  30.6 28.0 15.6   Gp2  35.0 27.1 15.3
 ;
```

Example 54.1: EM Algorithm for MLE

This example uses the EM algorithm to compute the maximum likelihood estimates for parameters of multivariate normally distributed data with missing values. The following statements invoke the MI procedure and request the EM algorithm to compute the MLE for $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ of a multivariate normal distribution from the input data set FitMiss:

```
proc mi data=FitMiss seed=1518971 simple nimpute=0;
   em itprint outem=outem;
   var Oxygen RunTime RunPulse;
run;
```

Note that when you specify the NIMPUTE=0 option, the missing values are not imputed.

The "Model Information" table in Output 54.1.1 describes the method and options used in the procedure if a positive number is specified in the NIMPUTE= option.

Output 54.1.1 Model Information

The MI Procedure

Model Information

Data Set	WORK.FITMISS
Method	MCMC
Multiple Imputation Chain	Single Chain
Initial Estimates for MCMC	EM Posterior Mode
Start	Starting Value
Prior	Jeffreys
Number of Imputations	0
Number of Burn-in Iterations	200
Number of Iterations	100
Seed for random number generator	1518971

The "Missing Data Patterns" table in Output 54.1.2 lists distinct missing data patterns with corresponding frequencies and percents. Here, a value of "X" means that the variable is observed in the corresponding group and a value of "." means that the variable is missing. The table also displays group-specific variable means.

Output 54.1.2 Missing Data Patterns

Missing Data Patterns

Group	Oxygen	Run Time	Run Pulse	Freq	Percent
1	X	X	X	21	67.74
2	X	X	.	4	12.90
3	X	.	.	3	9.68
4	.	X	X	1	3.23
5	.	X	.	2	6.45

Missing Data Patterns

	Group Means		
Group	Oxygen	RunTime	RunPulse
1	46.353810	10.809524	171.666667
2	47.109500	10.137500	.
3	52.461667	.	.
4	.	11.950000	176.000000
5	.	9.885000	.

With the SIMPLE option, the procedure displays simple descriptive univariate statistics for available cases in the "Univariate Statistics" table in Output 54.1.3 and correlations from pairwise available cases in the "Pairwise Correlations" table in Output 54.1.4.

Output 54.1.3 Univariate Statistics

Univariate Statistics

Variable	N	Mean	Std Dev	Minimum	Maximum
Oxygen	28	47.11618	5.41305	37.38800	60.05500
RunTime	28	10.68821	1.37988	8.63000	14.03000
RunPulse	22	171.86364	10.14324	148.00000	186.00000

Univariate Statistics

	---Missing Values--	
Variable	Count	Percent
Oxygen	3	9.68
RunTime	3	9.68
RunPulse	9	29.03

Output 54.1.4 Pairwise Correlations

Pairwise Correlations

	Oxygen	RunTime	RunPulse
Oxygen	1.000000000	-0.849118562	-0.343961742
RunTime	-0.849118562	1.000000000	0.247258191
RunPulse	-0.343961742	0.247258191	1.000000000

When you use the EM statement, the MI procedure displays the initial parameter estimates for the EM algorithm in the "Initial Parameter Estimates for EM" table in Output 54.1.5.

Output 54.1.5 Initial Parameter Estimates for EM

Initial Parameter Estimates for EM

TYPE	_NAME_	Oxygen	RunTime	RunPulse
MEAN		47.116179	10.688214	171.863636
COV	Oxygen	29.301078	0	0
COV	RunTime	0	1.904067	0
COV	RunPulse	0	0	102.885281

When you use the ITPRINT option in the EM statement, the "EM (MLE) Iteration History" table in Output 54.1.6 displays the iteration history for the EM algorithm.

Output 54.1.6 EM (MLE) Iteration History

EM (MLE) Iteration History

Iteration	-2 Log L	Oxygen	RunTime	RunPulse
0	289.544782	47.116179	10.688214	171.863636
1	263.549489	47.116179	10.688214	171.863636
2	255.851312	47.139089	10.603506	171.538203
3	254.616428	47.122353	10.571685	171.426790
4	254.494971	47.111080	10.560585	171.398296
5	254.483973	47.106523	10.556768	171.389208
6	254.482920	47.104899	10.555485	171.385257
7	254.482813	47.104348	10.555062	171.383345
8	254.482801	47.104165	10.554923	171.382424
9	254.482800	47.104105	10.554878	171.381992
10	254.482800	47.104086	10.554864	171.381796
11	254.482800	47.104079	10.554859	171.381708
12	254.482800	47.104077	10.554858	171.381669

The "EM (MLE) Parameter Estimates" table in Output 54.1.7 displays the maximum likelihood estimates for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ of a multivariate normal distribution from the data set FitMiss.

Output 54.1.7 EM (MLE) Parameter Estimates

EM (MLE) Parameter Estimates

TYPE	_NAME_	Oxygen	RunTime	RunPulse
MEAN		47.104077	10.554858	171.381669
COV	Oxygen	27.797931	-6.457975	-18.031298
COV	RunTime	-6.457975	2.015514	3.516287
COV	RunPulse	-18.031298	3.516287	97.766857

You can also output the EM (MLE) parameter estimates to an output data set with the OUTEM= option. The following statements list the observations in the output data set outem:

```
proc print data=outem;
   title 'EM Estimates';
run;
```

The output data set outem in Output 54.1.8 is a TYPE=COV data set. The observation with _TYPE_='MEAN' contains the MLE for the parameter $\boldsymbol{\mu}$, and the observations with _TYPE_='COV' contain the MLE for the parameter $\boldsymbol{\Sigma}$ of a multivariate normal distribution from the data set FitMiss.

Output 54.1.8 EM Estimates

EM Estimates

Obs	_TYPE_	_NAME_	Oxygen	RunTime	RunPulse
1	MEAN		47.1041	10.5549	171.382
2	COV	Oxygen	27.7979	-6.4580	-18.031
3	COV	RunTime	-6.4580	2.0155	3.516
4	COV	RunPulse	-18.0313	3.5163	97.767

Example 54.2: Propensity Score Method

This example uses the propensity score method to impute missing values for variables in a data set with a monotone missing pattern. The following statements invoke the MI procedure and request the propensity score method. The resulting data set is named outex2.

```
proc mi data=Fish1 seed=899603 out=outex2;
   monotone propensity;
   var Length1 Length2 Length3;
run;
```

Note that the VAR statement is required and the data set must have a monotone missing pattern with variables as ordered in the VAR statement.

The "Model Information" table in Output 54.2.1 describes the method and options used in the multiple imputation process. By default, five imputations are created for the missing data.

Output 54.2.1 Model Information

The MI Procedure

Model Information

Data Set	WORK.FISH1
Method	Monotone
Number of Imputations	5
Seed for random number generator	899603

When monotone methods are used in the imputation, MONOTONE is displayed as the method. The "Monotone Model Specification" table in Output 54.2.2 displays the detailed model specification. By default, the observations are sorted into five groups based on their propensity scores.

Output 54.2.2 Monotone Model Specification

Monotone Model Specification

Method	Imputed Variables
Propensity(Groups= 5)	Length2 Length3

Without covariates specified for imputed variables Length2 and Length3, the variable Length1 is used as the covariate for Length2, and the variables Length1 and Length2 are used as covariates for Length3.

The "Missing Data Patterns" table in Output 54.2.3 lists distinct missing data patterns with corresponding frequencies and percents. Here, values of "X" and "." indicate that the variable is observed or missing, respectively, in the corresponding group. The table confirms a monotone missing pattern for these three variables.

Output 54.2.3 Missing Data Patterns

Missing Data Patterns

Group	Length1	Length2	Length3	Freq	Percent
1	X	X	X	30	85.71
2	X	X	.	3	8.57
3	X	.	.	2	5.71

Missing Data Patterns

	Group Means		
Group	Length1	Length2	Length3
1	30.603333	33.436667	38.720000
2	29.033333	31.666667	.
3	27.750000	.	.

For the imputation process, first, missing values of Length2 in group 3 are imputed using observed values of Length1. Then the missing values of Length3 in group 2 are imputed using observed values of Length1 and Length2. And finally, the missing values of Length3 in group 3 are imputed using observed values of Length1 and imputed values of Length2.

After the completion of m imputations, the "Variance Information" table in Output 54.2.4 displays the between-imputation variance, within-imputation variance, and total variance for combining complete-data inferences. It also displays the degrees of freedom for the total variance. The relative increase in variance due to missingness, the fraction of missing information, and the relative efficiency for each variable are also displayed. A detailed description of these statistics is provided in the section "Combining Inferences from Multiply Imputed Data Sets" on page 3794.

Output 54.2.4 Variance Information

Variance Information

Variable	Between	Within	Total	DF
	-----------Variance-----------			
Length2	0.001500	0.465422	0.467223	32.034
Length3	0.049725	0.547434	0.607104	27.103

Variance Information

Variable	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Length2	0.003869	0.003861	0.999228
Length3	0.108999	0.102610	0.979891

The "Parameter Estimates" table in Output 54.2.5 displays the estimated mean and standard error of the mean for each variable. The inferences are based on the t distributions. For each variable, the table also displays a 95% mean confidence interval and a t statistic with the associated p-value for the hypothesis that the population mean is equal to the value specified in the MU0= option, which is zero by default.

Output 54.2.5 Parameter Estimates

Parameter Estimates

Variable	Mean	Std Error	95% Confidence Limits		DF
Length2	33.006857	0.683537	31.61460	34.39912	32.034
Length3	38.361714	0.779169	36.76328	39.96015	27.103

Parameter Estimates

Variable	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
Length2	32.957143	33.060000	0	48.29	<.0001
Length3	38.080000	38.545714	0	49.23	<.0001

The following statements list the first 10 observations of the data set outex2, as shown in Output 54.2.6. The missing values are imputed from observed values with similar propensity scores.

```
proc print data=outex2(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 54.2.6 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Length1	Length2	Length3
1	1	23.2	25.4	30.0
2	1	24.0	26.3	31.2
3	1	23.9	26.5	31.1
4	1	26.3	29.0	33.5
5	1	26.5	29.0	38.6
6	1	26.8	29.7	34.7
7	1	26.8	29.0	35.0
8	1	27.6	30.0	35.0
9	1	27.6	30.0	35.1
10	1	28.5	30.7	36.2

Example 54.3: Regression Method

This example uses the regression method to impute missing values for all variables in a data set with a monotone missing pattern. The following statements invoke the MI procedure and request the regression method for the variable Length2 and the predictive mean matching method for variable Length3. The resulting data set is named outex3.

```
proc mi data=Fish1 round=.1  mu0= 0 35 45
        seed=13951639 out=outex3;
   monotone reg(Length2/ details)
            regpmm(Length3= Length1 Length2 Length1*Length2/ details);
   var Length1 Length2 Length3;
run;
```

The ROUND= option is used to round the imputed values to the same precision as observed values. The values specified with the ROUND= option are matched with the variables Length1, Length2, and Length3 in the order listed in the VAR statement. The MU0= option requests *t* tests for the hypotheses that the population means corresponding to the variables in the VAR statement are Length2=35 and Length3=45.

The "Missing Data Patterns" table lists distinct missing data patterns with corresponding frequencies and percents. It is identical to the table in Output 54.2.3 in Example 54.2.

The "Monotone Model Specification" table in Output 54.3.1 displays the model specification.

Output 54.3.1 Monotone Model Specification

The MI Procedure

Monotone Model Specification

Method	Imputed Variables
Regression	Length2
Regression-PMM(K= 5)	Length3

When you use the DETAILS option, the parameters estimated from the observed data and the parameters used in each imputation are displayed in Output 54.3.2 and Output 54.3.3.

Output 54.3.2 Regression Model

Regression Models for Monotone Method

Imputed Variable	Effect	Obs-Data	Imputation 1	Imputation 2	Imputation 3
Length2	Intercept	-0.04249	-0.049184	-0.055470	-0.051346
Length2	Length1	0.98587	1.001934	0.995275	0.992294

Regression Models for Monotone Method

Imputed Variable	Effect	Imputation 4	Imputation 5
Length2	Intercept	-0.064193	-0.030719
Length2	Length1	0.983122	0.995883

Output 54.3.3 Regression Predicted Mean Matching Model

Regression Models for Monotone Predicted Mean Matching Method

Imputed Variable	Effect	Obs Data	Imputation 1	Imputation 2	Imputation 3
Length3	Intercept	-0.01304	0.004134	-0.011417	-0.034177
Length3	Length1	-0.01332	0.025320	-0.037494	0.308765
Length3	Length2	0.98918	0.955510	1.025741	0.673374
Length3	Length1*Length2	-0.02521	-0.034964	-0.022017	-0.017919

Regression Models for Monotone Predicted Mean Matching Method

Imputed Variable	Effect	Imputation 4	Imputation 5
Length3	Intercept	-0.010532	0.004685
Length3	Length1	0.156606	-0.147118
Length3	Length2	0.828384	1.146440
Length3	Length1*Length2	-0.029335	-0.034671

After the completion of five imputations by default, the "Variance Information" table in Output 54.3.4 displays the between-imputation variance, within-imputation variance, and total variance for combining complete-data inferences. The relative increase in variance due to missingness, the fraction of missing information, and the relative efficiency for each variable are also displayed. These statistics are described in the section "Combining Inferences from Multiply Imputed Data Sets" on page 3794.

Output 54.3.4 Variance Information

Variance Information

Variable	Between	Within	Total	DF
Length2	0.000133	0.439512	0.439672	32.15
Length3	0.000386	0.486913	0.487376	32.131

Variance Information

Variable	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Length2	0.000363	0.000363	0.999927
Length3	0.000952	0.000951	0.999810

The "Parameter Estimates" table in Output 54.3.5 displays a 95% mean confidence interval and a t statistic with its associated p-value for each of the hypotheses requested with the MU0= option.

Output 54.3.5 Parameter Estimates

Parameter Estimates

Variable	Mean	Std Error	95% Confidence Limits		DF
Length2	33.104571	0.663078	31.75417	34.45497	32.15
Length3	38.424571	0.698123	37.00277	39.84637	32.131

Parameter Estimates

Variable	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
Length2	33.088571	33.117143	35.000000	-2.86	0.0074
Length3	38.397143	38.445714	45.000000	-9.42	<.0001

The following statements list the first 10 observations of the data set outex3 in Output 54.3.6. Note that the imputed values of Length2 are rounded to the same precision as the observed values.

```
proc print data=outex3(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 54.3.6 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Length1	Length2	Length3
1	1	23.2	25.4	30.0
2	1	24.0	26.3	31.2
3	1	23.9	26.5	31.1
4	1	26.3	29.0	33.5
5	1	26.5	29.0	34.7
6	1	26.8	29.7	34.7
7	1	26.8	28.8	34.7
8	1	27.6	30.0	35.0
9	1	27.6	30.0	35.1
10	1	28.5	30.7	36.2

Example 54.4: Logistic Regression Method for CLASS Variables

This example uses logistic regression method to impute values for a binary variable in a data set with a monotone missing pattern.

In the following statements, the logistic regression method is used for the binary CLASS variable Species:

```
proc mi data=Fish2 seed=1305417 out=outex4;
   class Species;
   monotone logistic( Species= Height Width Height*Width/ details);
   var Height Width Species;
run;
```

The "Model Information" table in Output 54.4.1 describes the method and options used in the multiple imputation process.

Output 54.4.1 Model Information

The MI Procedure

Model Information

Data Set	WORK.FISH2
Method	Monotone
Number of Imputations	5
Seed for random number generator	1305417

The "Monotone Model Specification" table in Output 54.4.2 describes methods and imputed variables in the imputation model. The procedure uses the logistic regression method to impute the variable Species in the model. Missing values in other variables are not imputed.

Output 54.4.2 Monotone Model Specification

Monotone Model Specification

Method	Imputed Variables
Logistic Regression	Species

The "Missing Data Patterns" table in Output 54.4.3 lists distinct missing data patterns with corresponding frequencies and percents. The table confirms a monotone missing pattern for these variables.

Output 54.4.3 Missing Data Patterns

Missing Data Patterns

Group	Height	Width	Species	Freq	Percent	Group Means: Height	Group Means: Width
1	X	X	X	47	85.45	12.097645	4.808204
2	X	X	.	6	10.91	11.411050	4.567050
3	X	.	.	2	3.64	14.126350	.

When you use the DETAILS option, parameters estimated from the observed data and the parameters used in each imputation are displayed in the "Logistic Models for Monotone Method" table in Output 54.4.4.

Output 54.4.4 Logistic Regression Model

Logistic Models for Monotone Method

Imputed Variable	Effect	Obs-Data	Imputation 1	Imputation 2	Imputation 3
Species	Intercept	2.14183	1.240681	5.018482	5.509416
Species	Height	9.08604	3.774512	11.322763	11.230355
Species	Width	-5.02065	0.674528	-6.245428	-5.785890
Species	Height*Width	-1.91634	-3.299450	-3.326538	-5.045058

Logistic Models for Monotone Method

Imputed Variable	Effect	Imputation 4	Imputation 5
Species	Intercept	-1.325099	6.069734
Species	Height	5.711366	12.766614
Species	Width	2.394018	-9.689260
Species	Height*Width	-2.570333	-2.214031

The following statements list the first 10 observations of the data set outex4 in Output 54.4.5:

```
proc print data=outex4(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 54.4.5 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Species	Length3	Height	Width
1	1	Gp1	30.0	11.5200	4.0200
2	1	Gp1	31.2	12.4800	4.3056
3	1	Gp1	31.1	12.3778	4.6961
4	1		33.5	12.7300	.
5	1	Gp1	34.0	12.4440	5.1340
6	1	Gp1	34.7	13.6024	4.9274
7	1	Gp1	34.5	14.1795	5.2785
8	1	Gp1	35.0	12.6700	4.6900
9	1	Gp1	35.1	14.0049	4.8438
10	1	Gp1	36.2	14.2266	4.9594

Note that a missing value of the variable Species is not imputed if the corresponding covariates are missing and not imputed, as shown by observation 4 in the table.

Example 54.5: Discriminant Function Method for CLASS Variables

This example uses discriminant monotone methods to impute values of a CLASS variable from the observed observation values in a data set with a monotone missing pattern.

The following statements impute the continuous variables Height and Width with the regression method and the classification variable Species with the discriminant function method:

```
proc mi data=Fish2 seed=7545417 nimpute=3 out=outex5;
   class Species;
   monotone reg( Height Width)
            discrim( Species= Length3 Height Width/ details);
   var Length3 Height Width Species;
run;
```

The "Model Information" table in Output 54.5.1 describes the method and options used in the multiple imputation process.

Output 54.5.1 Model Information

```
                          The MI Procedure

                         Model Information

        Data Set                              WORK.FISH2
        Method                                Monotone
        Number of Imputations                 3
        Seed for random number generator      7545417
```

The "Monotone Model Specification" table in Output 54.5.2 describes methods and imputed variables in the imputation model. The procedure uses the regression method to impute the variables Height and Width, and uses the logistic regression method to impute the variable Species in the model.

Output 54.5.2 Monotone Model Specification

```
                    Monotone Model Specification

                                          Imputed
                Method                    Variables

                Regression                Height Width
                Discriminant Function     Species
```

The "Missing Data Patterns" table in Output 54.5.3 lists distinct missing data patterns with corresponding frequencies and percents. The table confirms a monotone missing pattern for these variables.

Output 54.5.3 Missing Data Patterns

Missing Data Patterns

Group	Length3	Height	Width	Species	Freq	Percent
1	X	X	X	X	47	85.45
2	X	X	X	.	6	10.91
3	X	X	.	.	2	3.64

Missing Data Patterns

	Group Means		
Group	Length3	Height	Width
1	33.497872	12.097645	4.808204
2	32.366667	11.411050	4.567050
3	36.600000	14.126350	.

When you use the DETAILS option, the parameters estimated from the observed data and the parameters used in each imputation are displayed in Output 54.5.4.

Output 54.5.4 Discriminant Model

Group Means for Monotone Discriminant Method

			Imputation		
Species	Variable	Obs-Data	1	2	3
Gp1	Length3	0.68104	0.766779	0.724277	0.577304
Gp1	Height	0.74011	0.809770	0.794103	0.671612
Gp1	Width	0.63865	0.700122	0.725179	0.579870
Gp2	Length3	-1.00022	-0.809466	-0.999101	-0.908734
Gp2	Height	-1.09007	-0.965672	-1.089324	-1.024453
Gp2	Width	-0.88135	-0.710969	-0.827099	-0.746598

The following statements list the first 10 observations of the data set outex5 in Output 54.5.5. Note that all missing values of the variables Width and Species are imputed.

```
proc print data=outex5(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 54.5.5 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Species	Length3	Height	Width
1	1	Gp1	30.0	11.5200	4.02000
2	1	Gp1	31.2	12.4800	4.30560
3	1	Gp1	31.1	12.3778	4.69610
4	1	Gp1	33.5	12.7300	4.67966
5	1	Gp2	34.0	12.4440	5.13400
6	1	Gp1	34.7	13.6024	4.92740
7	1	Gp1	34.5	14.1795	5.27850
8	1	Gp1	35.0	12.6700	4.69000
9	1	Gp1	35.1	14.0049	4.84380
10	1	Gp1	36.2	14.2266	4.95940

Example 54.6: MCMC Method

This example uses the MCMC method to impute missing values for a data set with an arbitrary missing pattern. The following statements invoke the MI procedure and specify the MCMC method with six imputations:

```
proc mi data=FitMiss seed=21355417 nimpute=6 mu0=50 10 180 ;
   mcmc chain=multiple displayinit initial=em(itprint);
   var Oxygen RunTime RunPulse;
run;
```

The "Model Information" table in Output 54.6.1 describes the method used in the multiple imputation process. When you use the CHAIN=MULTIPLE option, the procedure uses multiple chains and completes the default 200 burn-in iterations before each imputation. The 200 burn-in iterations are used to make the iterations converge to the stationary distribution before the imputation.

Output 54.6.1 Model Information

The MI Procedure

Model Information

Data Set	WORK.FITMISS
Method	MCMC
Multiple Imputation Chain	Multiple Chains
Initial Estimates for MCMC	EM Posterior Mode
Start	Starting Value
Prior	Jeffreys
Number of Imputations	6
Number of Burn-in Iterations	200
Seed for random number generator	21355417

By default, the procedure uses a noninformative Jeffreys prior to derive the posterior mode from the EM algorithm as the starting values for the MCMC method.

The "Missing Data Patterns" table in Output 54.6.2 lists distinct missing data patterns with corresponding statistics.

Output 54.6.2 Missing Data Patterns

Missing Data Patterns

Group	Oxygen	Run Time	Run Pulse	Freq	Percent
1	X	X	X	21	67.74
2	X	X	.	4	12.90
3	X	.	.	3	9.68
4	.	X	X	1	3.23
5	.	X	.	2	6.45

Missing Data Patterns

	Group Means		
Group	Oxygen	RunTime	RunPulse
1	46.353810	10.809524	171.666667
2	47.109500	10.137500	.
3	52.461667	.	.
4	.	11.950000	176.000000
5	.	9.885000	.

When you use the ITPRINT option within the INITIAL=EM option, the procedure displays the "EM (Posterior Mode) Iteration History" table in Output 54.6.3.

Output 54.6.3 EM (Posterior Mode) Iteration History

EM (Posterior Mode) Iteration History

Iteration	-2 Log L	-2 Log Posterior	Oxygen	RunTime
0	254.482800	282.909549	47.104077	10.554858
1	255.081168	282.051584	47.104077	10.554857
2	255.271408	282.017488	47.104077	10.554857
3	255.318622	282.015372	47.104002	10.554523
4	255.330259	282.015232	47.103861	10.554388
5	255.333161	282.015222	47.103797	10.554341
6	255.333896	282.015222	47.103774	10.554325
7	255.334085	282.015222	47.103766	10.554320

EM (Posterior Mode) Iteration History

Iteration	RunPulse
0	171.381669
1	171.381652
2	171.381644
3	171.381842
4	171.382053
5	171.382150
6	171.382185
7	171.382196

When you use the DISPLAYINIT option in the MCMC statement, the "Initial Parameter Estimates for MCMC" table in Output 54.6.4 displays the starting mean and covariance estimates used in the MCMC method. The same starting estimates are used in the MCMC method for multiple chains because the EM algorithm is applied to the same data set in each chain. You can explicitly specify different initial estimates for different imputations, or you can use the bootstrap method to generate different parameter estimates from the EM algorithm for the MCMC method.

Output 54.6.4 Initial Parameter Estimates

Initial Parameter Estimates for MCMC

TYPE	_NAME_	Oxygen	RunTime	RunPulse
MEAN		47.103766	10.554320	171.382196
COV	Oxygen	24.549967	-5.726112	-15.926036
COV	RunTime	-5.726112	1.781407	3.124798
COV	RunPulse	-15.926036	3.124798	83.164045

Output 54.6.5 and Output 54.6.6 display variance information and parameter estimates, respectively, from the multiple imputation.

Output 54.6.5 Variance Information

Variance Information

Variable	Between	Within	Total	DF
	-----------Variance-----------			
Oxygen	0.051560	0.928170	0.988323	25.958
RunTime	0.003979	0.070057	0.074699	25.902
RunPulse	4.118578	4.260631	9.065638	7.5938

Variance Information

Variable	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Oxygen	0.064809	0.062253	0.989731
RunTime	0.066262	0.063589	0.989513
RunPulse	1.127769	0.575218	0.912517

Output 54.6.6 Parameter Estimates

Parameter Estimates

Variable	Mean	Std Error	95% Confidence Limits		DF
Oxygen	47.164819	0.994145	45.1212	49.2085	25.958
RunTime	10.549936	0.273312	9.9880	11.1118	25.902
RunPulse	170.969836	3.010920	163.9615	177.9782	7.5938

Parameter Estimates

Variable	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
Oxygen	46.858020	47.363540	50.000000	-2.85	0.0084
RunTime	10.476886	10.659412	10.000000	2.01	0.0547
RunPulse	168.252615	172.894991	180.000000	-3.00	0.0182

Example 54.7: Producing Monotone Missingness with MCMC

This example uses the MCMC method to impute just enough missing values for a data set with an arbitrary missing pattern so that each imputed data set has a monotone missing pattern based on the order of variables in the VAR statement.

The following statements invoke the MI procedure and specify the IMPUTE=MONOTONE option to create the imputed data set with a monotone missing pattern. You must specify a VAR statement to provide the order of variables in order for the imputed data to achieve a monotone missing pattern.

```
proc mi data=FitMiss seed=17655417 out=outex7;
   mcmc impute=monotone;
   var Oxygen RunTime RunPulse;
run;
```

The "Model Information" table in Output 54.7.1 describes the method used in the multiple imputation process.

Output 54.7.1 Model Information

The MI Procedure

Model Information

Data Set	WORK.FITMISS
Method	Monotone-data MCMC
Multiple Imputation Chain	Single Chain
Initial Estimates for MCMC	EM Posterior Mode
Start	Starting Value
Prior	Jeffreys
Number of Imputations	5
Number of Burn-in Iterations	200
Number of Iterations	100
Seed for random number generator	17655417

The "Missing Data Patterns" table in Output 54.7.2 lists distinct missing data patterns with corresponding statistics. Here, an "X" means that the variable is observed in the corresponding group, a "." means that the variable is missing and will be imputed to achieve the monotone missingness for the imputed data set, and an "O" means that the variable is missing and will not be imputed. The table also displays group-specific variable means.

Output 54.7.2 Missing Data Patterns

Missing Data Patterns

Group	Oxygen	Run Time	Run Pulse	Freq	Percent
1	X	X	X	21	67.74
2	X	X	O	4	12.90
3	X	O	O	3	9.68
4	.	X	X	1	3.23
5	.	X	O	2	6.45

Missing Data Patterns

	Group Means		
Group	Oxygen	RunTime	RunPulse
1	46.353810	10.809524	171.666667
2	47.109500	10.137500	.
3	52.461667	.	.
4	.	11.950000	176.000000
5	.	9.885000	.

As shown in the table in Output 54.7.2, the MI procedure needs to impute only three missing values from group 4 and group 5 to achieve a monotone missing pattern for the imputed data set.

When you use the MCMC method to produce an imputed data set with a monotone missing pattern, tables of variance information and parameter estimates are not created.

The following statements are used just to show the monotone missingness of the output data set outex7:

```
proc mi data=outex7  nimpute=0;
   var Oxygen RunTime RunPulse;
run;
```

The "Missing Data Patterns" table in Output 54.7.3 displays a monotone missing data pattern.

Output 54.7.3 Monotone Missing Data Patterns

The MI Procedure

Missing Data Patterns

Group	Oxygen	Run Time	Run Pulse	Freq	Percent
1	X	X	X	110	70.97
2	X	X	.	30	19.35
3	X	.	.	15	9.68

Missing Data Patterns

	Group Means		
Group	Oxygen	RunTime	RunPulse
1	46.152428	10.861364	171.863636
2	47.796038	10.053333	.
3	52.461667	.	.

The following statements impute one value for each missing value in the monotone missingness data set outex7:

```
proc mi data=outex7 nimpute=1 seed=51343672 out=outds;
   monotone method=reg;
   var Oxygen RunTime RunPulse;
   by _Imputation_;
run;
```

You can then analyze these data sets by using other SAS procedures and combine these results by using the MIANALYZE procedure. Note that the VAR statement is required with a MONOTONE statement to provide the variable order for the monotone missing pattern.

Example 54.8: Checking Convergence in MCMC

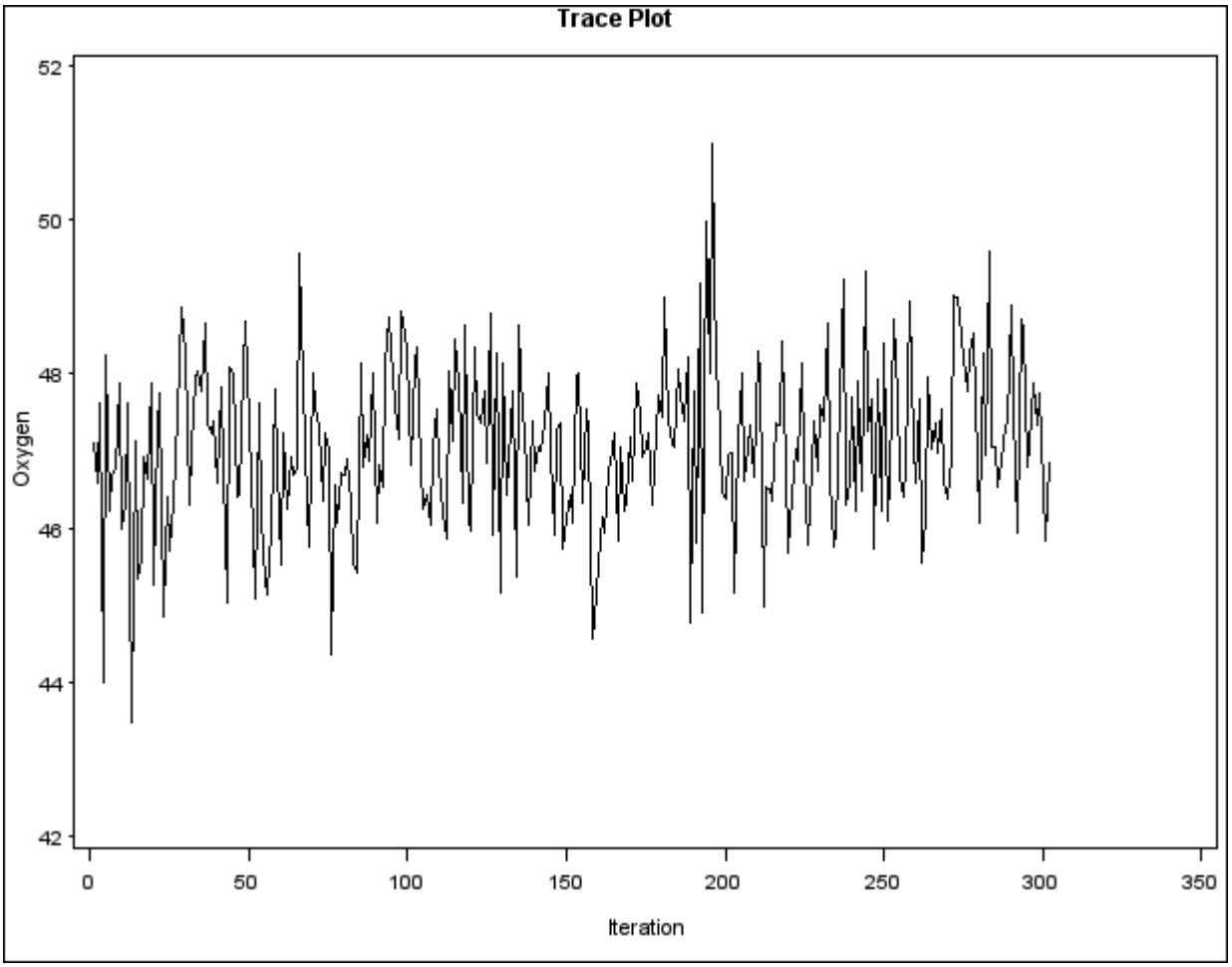
This example uses the MCMC method with a single chain. It also displays trace and autocorrelation plots to check convergence for the single chain.

The following statements use the MCMC method to create an iteration plot for the successive estimates of the mean of Oxygen. Note that iterations during the burn-in period are indicated with negative iteration numbers. These statements also create an autocorrelation function plot for the variable Oxygen.

```
proc mi data=FitMiss seed=42037921 noprint nimpute=2;
   mcmc timeplot(mean(Oxygen)) acfplot(mean(Oxygen));
   var Oxygen RunTime RunPulse;
run;
```

Using the TIMEPLOT(MEAN(OXYGEN)) option, the MI procedure displays a trace plot for the mean of Oxygen in Output 54.8.1.

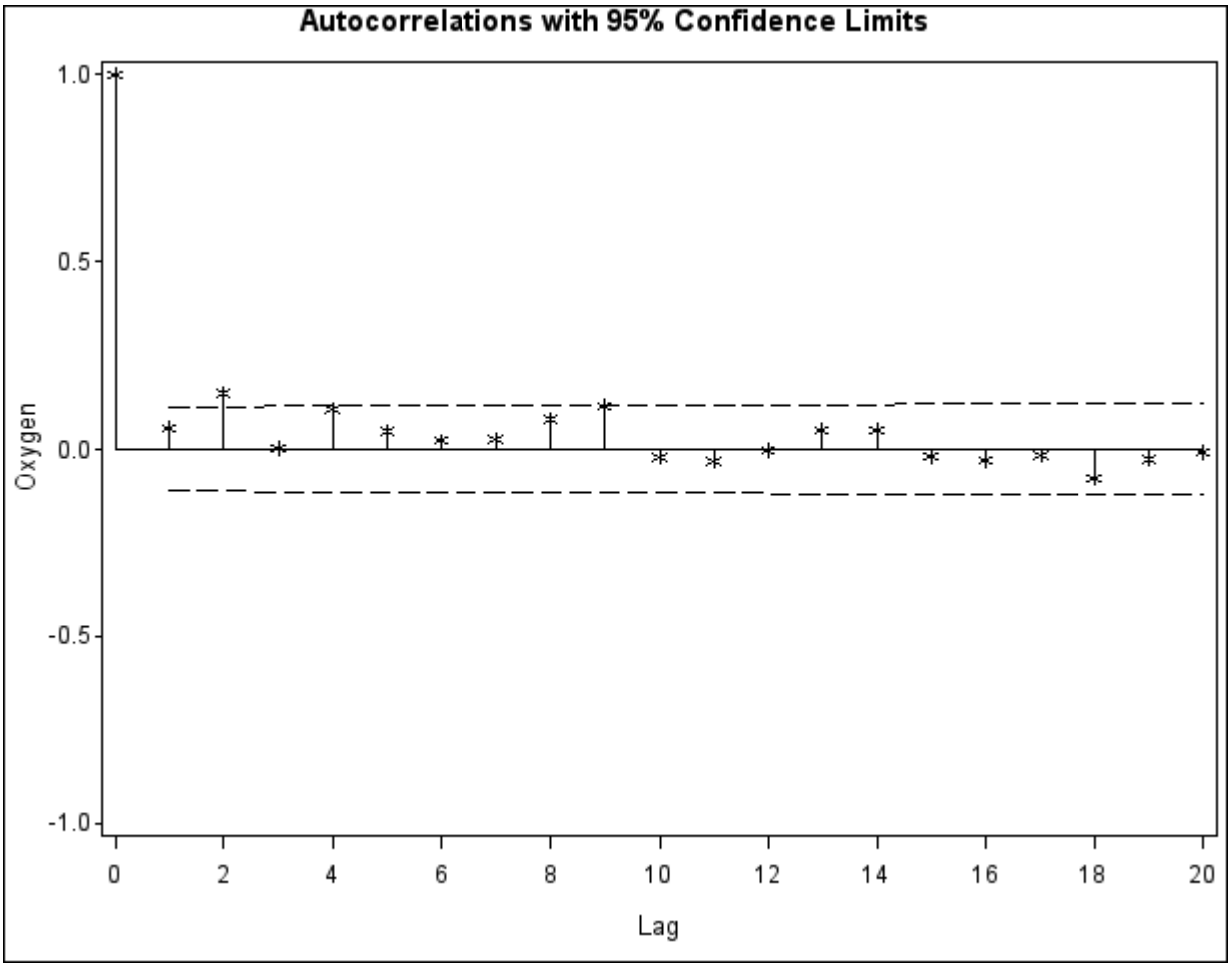
Output 54.8.1 Trace Plot for Oxygen



By default, the MI procedure displays solid line segments that connect data points in the trace plot. The plot shows no apparent trends for the variable Oxygen.

When you use the ACFPLOT(MEAN(OXYGEN)) option, the MI procedure displays an autocorrelation plot for the mean of Oxygen in Output 54.8.2.

Output 54.8.2 Autocorrelation Function Plot for Oxygen

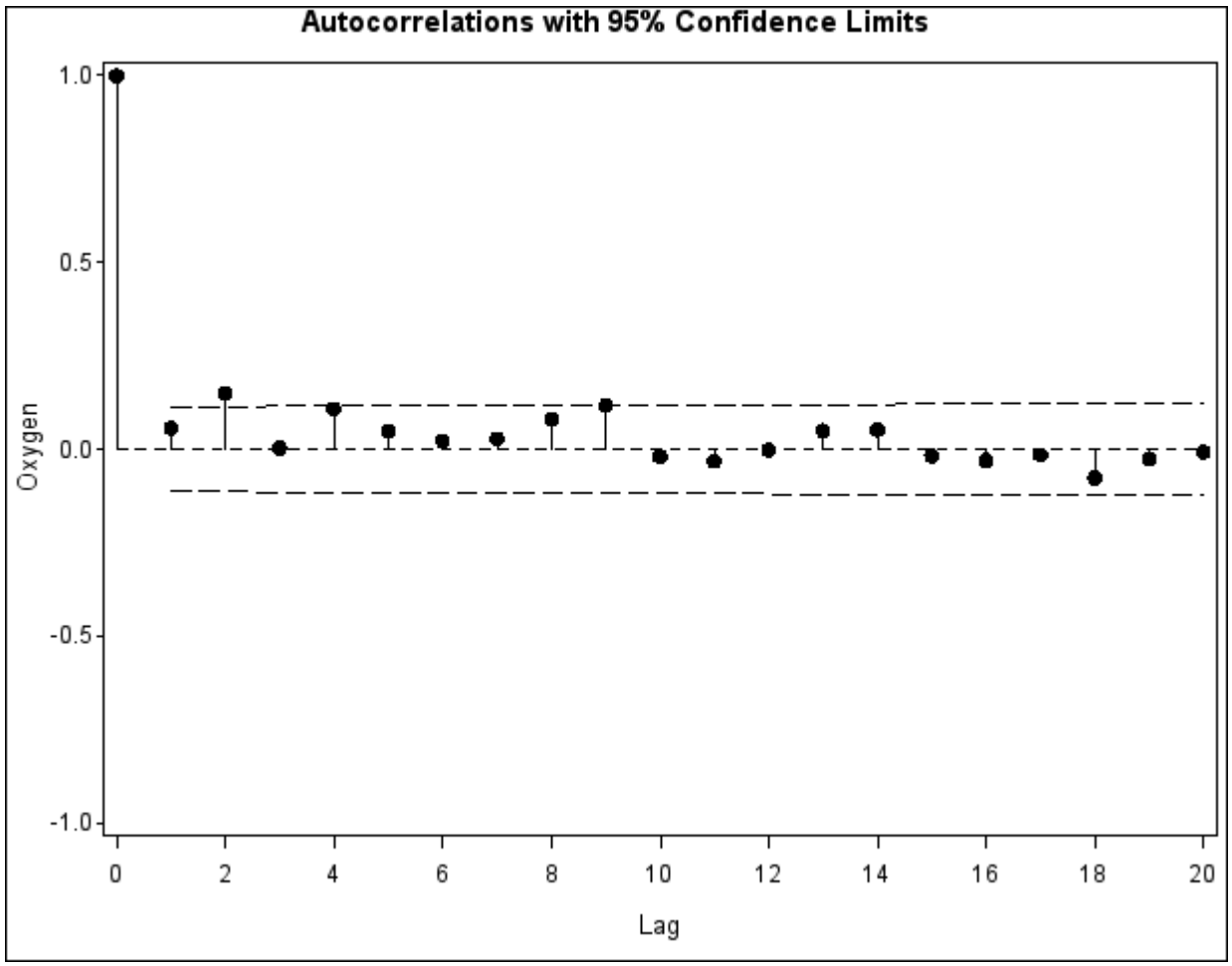


By default, the MI procedure uses the star (*) as the plot symbol to display the points in the plot, a solid line to display the reference line of zero autocorrelation, and a pair of dashed lines to display approximately 95% confidence limits for the autocorrelations. The autocorrelation function plot shows no significant positive or negative autocorrelation.

The following statements use display options to modify the autocorrelation function plot for Oxygen in Output 54.8.3:

```
proc mi data=FitMiss seed=42037921 noprint nimpute=2;
   mcmc acfplot(mean(Oxygen) / symbol=dot lref=2);
   var Oxygen RunTime RunPulse;
run;
```

Output 54.8.3 Autocorrelation Function Plot for Oxygen



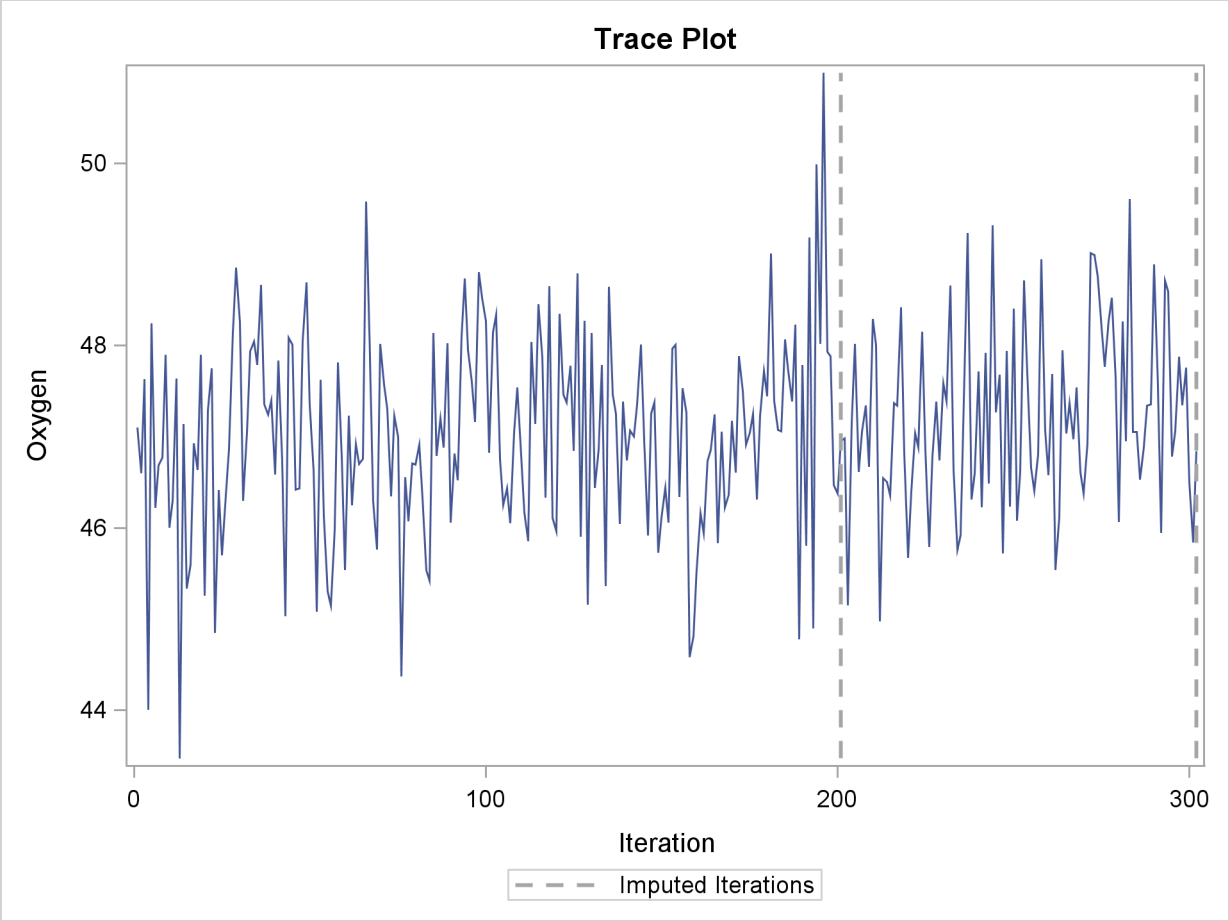
You can also create plots for the worst linear function, the means of other variables, the variances of variables, and the covariances between variables. Alternatively, you can use the OUTITER option to save statistics such as the means, standard deviations, covariances, −2 log LR statistic, −2 log LR statistic of the posterior mode, and worst linear function from each iteration in an output data set. Then you can do a more in-depth trace (time series) analysis of the iterations with other procedures, such as PROC AUTOREG and PROC ARIMA in the *SAS/ETS User's Guide*.

With the **ods graphics on** statement specified in the following statements, the MI procedure produces the ODS graphs in Output 54.8.4 and Output 54.8.5:

```
ods graphics on;
proc mi data=FitMiss seed=42037921 nimpute=2;
   mcmc plots=(trace(mean(Oxygen)) acf(mean(Oxygen)));
   var Oxygen RunTime RunPulse;
run;
ods graphics off;
```

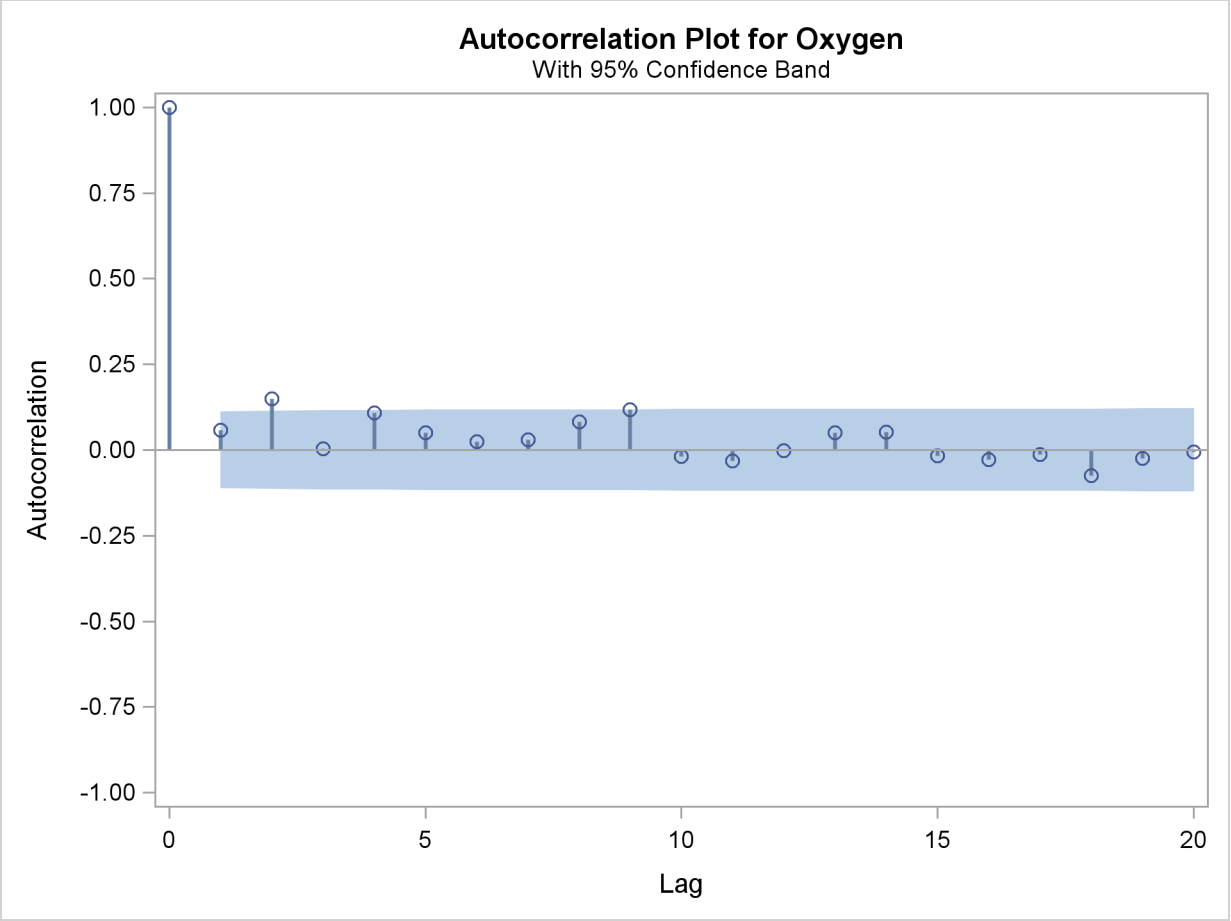
The TRACE(MEAN(OXYGEN)) option displays the trace plot of means for the variable Oxygen. The dashed vertical lines indicate the imputed iterations—that is, the Oxygen values used in the imputations.

Output 54.8.4 Trace Plot for Oxygen



The ACF(MEAN(OXYGEN)) option displays the autocorrelation plot of means for the variable Oxygen.

Output 54.8.5 Autocorrelation Function Plot for Oxygen



For general information about the **ods graphics** statement, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the MI procedure, see the section "ODS Graphics" on page 3800.

Example 54.9: Saving and Using Parameters for MCMC

This example uses the MCMC method with multiple chains as specified in Example 54.6. It saves the parameter values used for each imputation in an output data set of type EST called miest. This output data set can then be used to impute missing values in other similar input data sets. The following statements invoke the MI procedure and specify the MCMC method with multiple chains to create three imputations:

```
proc mi data=FitMiss seed=21355417 nimpute=6 mu0=50 10 180;
   mcmc chain=multiple initial=em outest=miest;
   var Oxygen RunTime RunPulse;
run;
```

The following statements list the parameters used for the imputations in Output 54.9.1. Note that the data set includes observations with _TYPE_='SEED' containing the seed to start the next random number generator.

```
proc print data=miest(obs=15);
   title 'Parameters for the Imputations';
run;
```

Output 54.9.1 OUTEST Data Set

Parameters for the Imputations

Obs	_Imputation_	_TYPE_	_NAME_	Oxygen	RunTime	RunPulse
1	1	SEED		825240167.00	825240167.00	825240167.00
2	1	PARM		46.77	10.47	169.41
3	1	COV	Oxygen	30.59	-8.32	-50.99
4	1	COV	RunTime	-8.32	2.90	17.03
5	1	COV	RunPulse	-50.99	17.03	200.09
6	2	SEED		1895925872.00	1895925872.00	1895925872.00
7	2	PARM		47.41	10.37	173.34
8	2	COV	Oxygen	22.35	-4.44	-21.18
9	2	COV	RunTime	-4.44	1.76	1.25
10	2	COV	RunPulse	-21.18	1.25	125.67
11	3	SEED		137653011.00	137653011.00	137653011.00
12	3	PARM		48.21	10.36	170.52
13	3	COV	Oxygen	23.59	-5.25	-19.76
14	3	COV	RunTime	-5.25	1.66	5.00
15	3	COV	RunPulse	-19.76	5.00	110.99

The following statements invoke the MI procedure and use the INEST= option in the MCMC statement:

```
proc mi data=FitMiss mu0=50 10 180;
   mcmc inest=miest;
   var Oxygen RunTime RunPulse;
run;
```

The "Model Information" table in Output 54.9.2 describes the method used in the multiple imputation process. The remaining tables for the example are identical to the tables in Output 54.6.2, Output 54.6.4, Output 54.6.5, and Output 54.6.6 in Example 54.6.

Output 54.9.2 Model Information

The MI Procedure

Model Information

Data Set	WORK.FITMISS
Method	MCMC
INEST Data Set	WORK.MIEST
Number of Imputations	6

Example 54.10: Transforming to Normality

This example applies the MCMC method to the FitMiss data set in which the variable Oxygen is transformed. Assume that Oxygen is skewed and can be transformed to normality with a logarithmic transformation. The following statements invoke the MI procedure and specify the transformation. The TRANSFORM statement specifies the log transformation for Oxygen. Note that the values displayed for Oxygen in all of the results correspond to transformed values.

```
proc mi data=FitMiss seed=32937921 mu0=50 10 180 out=outex10;
   transform log(Oxygen);
   mcmc chain=multiple displayinit;
   var Oxygen RunTime RunPulse;
run;
```

The "Missing Data Patterns" table in Output 54.10.1 lists distinct missing data patterns with corresponding statistics for the FitMiss data. Note that the values of Oxygen shown in the tables are transformed values.

Output 54.10.1 Missing Data Patterns

The MI Procedure

Missing Data Patterns

Group	Oxygen	Run Time	Run Pulse	Freq	Percent
1	X	X	X	21	67.74
2	X	X	.	4	12.90
3	X	.	.	3	9.68
4	.	X	X	1	3.23
5	.	X	.	2	6.45

Transformed Variables: Oxygen

Missing Data Patterns

Group	Group Means		
	Oxygen	RunTime	RunPulse
1	3.829760	10.809524	171.666667
2	3.851813	10.137500	.
3	3.955298	.	.
4	.	11.950000	176.000000
5	.	9.885000	.

Transformed Variables: Oxygen

The "Variable Transformations" table in Output 54.10.2 lists the variables that have been transformed.

Output 54.10.2 Variable Transformations

Variable Transformations

Variable	_Transform_
Oxygen	LOG

The "Initial Parameter Estimates for MCMC" table in Output 54.10.3 displays the starting mean and covariance estimates used in the MCMC method.

Output 54.10.3 Initial Parameter Estimates

Initial Parameter Estimates for MCMC

TYPE	_NAME_	Oxygen	RunTime	RunPulse
MEAN		3.846122	10.557605	171.382949
COV	Oxygen	0.010827	-0.120891	-0.328772
COV	RunTime	-0.120891	1.744580	3.011180
COV	RunPulse	-0.328772	3.011180	82.747609

Transformed Variables: Oxygen

Output 54.10.4 displays variance information from the multiple imputation.

Output 54.10.4 Variance Information

Variance Information

Variable	Variance: Between	Variance: Within	Variance: Total	DF
* Oxygen	0.000016175	0.000401	0.000420	26.499
RunTime	0.001762	0.065421	0.067536	27.118
RunPulse	0.205979	3.116830	3.364004	25.222

* Transformed Variables

Variance Information

Variable	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
* Oxygen	0.048454	0.047232	0.990642
RunTime	0.032318	0.031780	0.993684
RunPulse	0.079303	0.075967	0.985034

* Transformed Variables

Output 54.10.5 displays parameter estimates from the multiple imputation. Note that the parameter value of μ_0 has also been transformed using the logarithmic transformation.

Output 54.10.5 Parameter Estimates

Parameter Estimates

Variable	Mean	Std Error	95% Confidence Limits		DF
* Oxygen	3.845175	0.020494	3.8031	3.8873	26.499
RunTime	10.560131	0.259876	10.0270	11.0932	27.118
RunPulse	171.802181	1.834122	168.0264	175.5779	25.222

* Transformed Variables

Parameter Estimates

Variable	Minimum	Maximum	Mu0	t for H0: Mean=Mu0	Pr > |t|
* Oxygen	3.838599	3.848456	3.912023	-3.26	0.0030
RunTime	10.493031	10.600498	10.000000	2.16	0.0402
RunPulse	171.251777	172.498626	180.000000	-4.47	0.0001

* Transformed Variables

The following statements list the first 10 observations of the data set outmi in Output 54.10.6. Note that the values for Oxygen are in the original scale.

```
proc print data=outex10(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 54.10.6 Imputed Data Set in Original Scale

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Oxygen	RunTime	Run Pulse
1	1	44.6090	11.3700	178.000
2	1	45.3130	10.0700	185.000
3	1	54.2970	8.6500	156.000
4	1	59.5710	7.1440	167.012
5	1	49.8740	9.2200	170.092
6	1	44.8110	11.6300	176.000
7	1	38.5834	11.9500	176.000
8	1	43.7376	10.8500	158.851
9	1	39.4420	13.0800	174.000
10	1	60.0550	8.6300	170.000

Note that the results in Output 54.10.6 can also be produced from the following statements without using a TRANSFORM statement. A transformed value of log(50)=3.91202 is used in the MU0= option.

```
data temp;
   set FitMiss;
   LogOxygen= log(Oxygen);
run;
proc mi data=temp seed=14337921 mu0=3.91202 10 180 out=outtemp;
   mcmc chain=multiple displayinit;
   var LogOxygen RunTime RunPulse;
run;
data outex10;
   set outtemp;
   Oxygen= exp(LogOxygen);
run;
```

Example 54.11: Multistage Imputation

This example uses two separate imputation procedures to complete the imputation process. In the first case, the MI procedure statements use the MCMC method to impute just enough missing values for a data set with an arbitrary missing pattern so that each imputed data set has a monotone missing pattern. In the second case, the MI procedure statements use a MONOTONE statement to impute missing values for data sets with monotone missing patterns.

The following statements are identical to those in Example 54.7. The statements invoke the MI procedure and specify the IMPUTE=MONOTONE option to create the imputed data set with a monotone missing pattern.

```
proc mi data=FitMiss seed=17655417 out=outex11;
   mcmc impute=monotone;
   var Oxygen RunTime RunPulse;
run;
```

The "Missing Data Patterns" table in Output 54.11.1 lists distinct missing data patterns with corresponding statistics. Here, an "X" means that the variable is observed in the corresponding group, a "." means that the variable is missing and will be imputed to achieve the monotone missingness for the imputed data set, and an "O" means that the variable is missing and will not be imputed. The table also displays group-specific variable means.

Output 54.11.1 Missing Data Patterns

The MI Procedure

Missing Data Patterns

Group	Oxygen	Run Time	Run Pulse	Freq	Percent
1	X	X	X	21	67.74
2	X	X	O	4	12.90
3	X	O	O	3	9.68
4	.	X	X	1	3.23
5	.	X	O	2	6.45

Missing Data Patterns

	Group Means		
Group	Oxygen	RunTime	RunPulse
1	46.353810	10.809524	171.666667
2	47.109500	10.137500	.
3	52.461667	.	.
4	.	11.950000	176.000000
5	.	9.885000	.

As shown in the table, the MI procedure needs to impute only three missing values from group 4 and group 5 to achieve a monotone missing pattern for the imputed data set. When the MCMC method is used to produce an imputed data set with a monotone missing pattern, tables of variance information and parameter estimates are not created.

The following statements impute one value for each missing value in the monotone missingness data set outex11:

```
proc mi data=outex11
        nimpute=1 seed=51343672
        out=outex11a;
   monotone reg;
   var Oxygen RunTime RunPulse;
   by _Imputation_;
run;
```

You can then analyze these data sets by using other SAS procedures and combine these results by using the MIANALYZE procedure. Note that the VAR statement is required with a MONOTONE statement to provide the variable order for the monotone missing pattern.

The "Model Information" table in Output 54.11.2 shows that a monotone method is used to generate imputed values in the first BY group.

Output 54.11.2 Model Information

```
----------------------------- Imputation Number=1 ------------------------------

                                The MI Procedure

                               Model Information

          Data Set                              WORK.OUTEX11
          Method                                Monotone
          Number of Imputations                 1
          Seed for random number generator      51343672
```

The "Monotone Model Specification" table in Output 54.11.3 describes methods and imputed variables in the imputation model. The MI procedure uses the regression method to impute the variables RunTime and RunPulse in the model.

Output 54.11.3 Monotone Model Specification

```
----------------------------- Imputation Number=1 ------------------------------

                          Monotone Model Specification

                                        Imputed
                         Method         Variables

                         Regression     RunTime RunPulse
```

The "Missing Data Patterns" table in Output 54.11.4 lists distinct missing data patterns with corresponding statistics. It shows a monotone missing pattern for the imputed data set.

Output 54.11.4 Missing Data Patterns

```
----------------------------- Imputation Number=1 ------------------------------

                              Missing Data Patterns

                               Run     Run
            Group    Oxygen    Time    Pulse        Freq     Percent

                1    X         X       X              22       70.97
                2    X         X       .               6       19.35
                3    X         .       .               3        9.68

                             Missing Data Patterns

                      -----------------Group Means----------------
             Group          Oxygen         RunTime        RunPulse

                 1       46.057479       10.861364      171.863636
                 2       46.745227       10.053333               .
                 3       52.461667               .               .
```

The following statements list the first 10 observations of the data set outex11a in Output 54.11.5:

```
proc print data=outex11a(obs=10);
   title 'First 10 Observations of the Imputed Data Set';
run;
```

Output 54.11.5 Imputed Data Set

First 10 Observations of the Imputed Data Set

Obs	_Imputation_	Oxygen	RunTime	Run Pulse
1	1	44.6090	11.3700	178.000
2	1	45.3130	10.0700	185.000
3	1	54.2970	8.6500	156.000
4	1	59.5710	7.1569	169.914
5	1	49.8740	9.2200	159.315
6	1	44.8110	11.6300	176.000
7	1	39.8345	11.9500	176.000
8	1	45.3196	10.8500	151.252
9	1	39.4420	13.0800	174.000
10	1	60.0550	8.6300	170.000

This example presents an alternative to the full-data MCMC imputation, in which imputation of only a few missing values is needed to achieve a monotone missing pattern for the imputed data set. The example uses a monotone MCMC method that imputes fewer missing values in each iteration and achieves approximate stationarity in fewer iterations (Schafer 1997, p. 227). The example also demonstrates how to combine the monotone MCMC method with a method for monotone missing data, which does not rely on iterations of steps.

References

Anderson, T. W. (1984), *An Introduction to Multivariate Statistical Analysis,* Second Edition, New York: John Wiley & Sons.

Allison, P. D. (2000), "Multiple Imputation for Missing Data: A Cautionary Tale," *Sociological Methods and Research*, 28, 301–309.

Allison, P. D. (2001), "Missing Data," Thousand Oaks, CA: Sage Publications.

Barnard, J., and Meng, X. L. (1999), "Applications of Multiple Imputation in Medical Studies: From AIDS to NHANES," *Statistical Methods in Medical Research*, 8, 17–36.

Barnard, J. and Rubin, D. B. (1999), "Small-Sample Degrees of Freedom with Multiple Imputation," *Biometrika*, 86, 948–955.

Brand, J. P. L. (1999), "Development, Implementation and Evaluation of Multiple Imputation Strategies for the Statistical Analysis of Incomplete Data Sets," Ph.D. dissertation, Erasmus University, Rotterdam.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, Ser. B., 39, 1–38.

Gadbury, G. L., Coffey, C. S., and Allison, D. B. (2003), "Modern Statistical Methods for Handling Missing Repeated Measurements in Obesity Trial Data: Beyond LOCF," *Obesity Reviews*, 4, 175–184.

Gelman, A. and Rubin, D. B. (1992), "Inference from Iterative Simulation Using Multiple Sequences," *Statistical Science*, 7, 457–472.

Goodnight, J. H. (1979), "A Tutorial on the Sweep Operator," *American Statistician*, 33, 149–158.

Heitjan, F. and Little, R. J. A. (1991), "Multiple Imputation for the Fatal Accident Reporting System," *Applied Statistics*, 40, 13–29.

Horton, N. J. and Lipsitz, S. R. (2001), "Multiple Imputation in Practice: Comparison of Software Packages for Regression Models with Missing Variables," *The American Statistician*, 55, 244–254.

Lavori, P. W., Dawson, R., and Shera, D. (1995), "A Multiple Imputation Strategy for Clinical Trials with Truncation of Patient Data," *Statistics in Medicine*, 14, 1913–1925.

Li, K. H. (1988), "Imputation Using Markov Chains," *Journal of Statistical Computation and Simulation*, 30, 57–79.

Li, K. H., Raghunathan, T. E., and Rubin, D. B. (1991), "Large-Sample Significance Levels from Multiply Imputed Data Using Moment-Based Statistics and an F Reference Distribution," *Journal of the American Statistical Association*, 86, 1065–1073.

Little, R. J. A. and Rubin, D. B. (2002), *Statistical Analysis with Missing Data*, Second Edition, New York: John Wiley & Sons.

Liu, C. (1993), "Bartlett's Decomposition of the Posterior Distribution of the Covariance for Normal Monotone Ignorable Missing Data," *Journal of Multivariate Analysis*, 46, 198–206.

McLachlan, G. J. and Krishnan, T. (1997), *The EM Algorithm and Extensions*, New York: John Wiley & Sons.

Rosenbaum, P. R. and Rubin, D. B. (1983), "The Central Role of the Propensity Score in Observational Studies for Causal Effects," *Biometrika*, 70, 41–55.

Rubin, D. B. (1976), "Inference and Missing Data," *Biometrika*, 63, 581–592.

Rubin, D. B. (1987), *Multiple Imputation for Nonresponse in Surveys*, New York: John Wiley & Sons.

Rubin, D. B. (1996), "Multiple Imputation after 18+ Years," *Journal of the American Statistical Association*, 91, 473–489.

Schafer, J. L. (1997), *Analysis of Incomplete Multivariate Data*, New York: Chapman and Hall.

Schafer, J. L. (1999), "Multiple Imputation: A Primer," *Statistical Methods in Medical Research*, 8, 3–15.

Schenker, N. and Taylor, J. M. G. (1996), "Partially Parametric Techniques for Multiple Imputation," *Computational Statistics and Data Analysis*, 22, 425–446.

Tanner, M. A. and Wong, W. H. (1987), "The Calculation of Posterior Distributions by Data Augmentation," *Journal of the American Statistical Association*, 82, 528–540.

van Buuren, S., Boshuizen, H. C., and Knook, D. L. (1999), "Multiple Imputation of Missing Blood Pressure Covariates in Survival Analysis," *Statistics in Medicine*, 18, 681–694.

Chapter 55

The MIANALYZE Procedure

Contents

Overview: MIANALYZE Procedure **3840**
Getting Started: MIANALYZE Procedure **3841**
Syntax: MIANALYZE Procedure **3844**
PROC MIANALYZE Statement 3845
BY Statement 3847
CLASS Statement 3848
MODELEFFECTS Statement 3848
STDERR Statement 3849
TEST Statement 3849
Details: MIANALYZE Procedure **3851**
Input Data Sets 3851
Combining Inferences from Imputed Data Sets 3855
Multiple Imputation Efficiency 3857
Multivariate Inferences 3857
Testing Linear Hypotheses about the Parameters 3859
Examples of the Complete-Data Inferences 3859
ODS Table Names 3861
Examples: MIANALYZE Procedure **3862**
Example 55.1: Reading Means and Standard Errors from Variables in a DATA= Data Set 3863
Example 55.2: Reading Means and Covariance Matrices from a DATA= COV Data Set 3866
Example 55.3: Reading Regression Results from a DATA= EST Data Set . . 3869
Example 55.4: Reading Mixed Model Results from PARMS= and COVB= Data Sets 3871
Example 55.5: Reading Generalized Linear Model Results 3874
Example 55.6: Reading GLM Results from PARMS= and XPXI= Data Sets 3876
Example 55.7: Reading Logistic Model Results from PARMS= and COVB= Data Sets 3878
Example 55.8: Reading Mixed Model Results with Classification Variables . 3881
Example 55.9: Using a TEST statement 3884
Example 55.10: Combining Correlation Coefficients 3886
References **3889**

Overview: MIANALYZE Procedure

The MIANALYZE procedure combines the results of the analyses of imputations and generates valid statistical inferences. Multiple imputation provides a useful strategy for analyzing data sets with missing values. Instead of filling in a single value for each missing value, Rubin's (1976, 1987) multiple imputation strategy replaces each missing value with a set of plausible values that represent the uncertainty about the right value to impute.

Multiple imputation inference involves three distinct phases:

1. The missing data are filled in *m* times to generate *m* complete data sets.
2. The *m* complete data sets are analyzed using standard statistical analyses.
3. The results from the *m* complete data sets are combined to produce inferential results.

A companion procedure, PROC MI, creates multiply imputed data sets for incomplete multivariate data. It uses methods that incorporate appropriate variability across the *m* imputations.

The analyses of imputations are obtained by using standard SAS procedures (such as PROC REG) for complete data. No matter which complete-data analysis is used, the process of combining results from different imputed data sets is essentially the same and results in valid statistical inferences that properly reflect the uncertainty due to missing values. These results of analyses are combined in the MIANALYZE procedure to derive valid inferences.

The MIANALYZE procedure reads parameter estimates and associated standard errors or covariance matrix that are computed by the standard statistical procedure for each imputed data set. The MIANALYZE procedure then derives valid univariate inference for these parameters. With an additional assumption about the population between and within imputation covariance matrices, multivariate inference based on Wald tests can also be derived.

The MODELEFFECTS statement lists the effects to be analyzed, and the CLASS statement lists the classification variables in the MODELEFFECTS statement. The variables in the MODELEFFECTS statement that are not specified in a CLASS statement are assumed to be continuous.

When each effect in the MODELEFFECTS statement is a continuous variable by itself, a STDERR statement specifies the standard errors when both parameter estimates and associated standard errors are stored as variables in the same data set.

For some parameters of interest, you can use TEST statements to test linear hypotheses about the parameters. For others, it is not straightforward to compute estimates and associated covariance matrices with standard statistical SAS procedures. Examples include correlation coefficients between two variables and ratios of variable means. These special cases are described in the section "Examples of the Complete-Data Inferences" on page 3859.

Getting Started: MIANALYZE Procedure

Consider the following Fitness data set that has been altered to contain an arbitrary pattern of missingness:

```
*----------------- Data on Physical Fitness -----------------*
| These measurements were made on men involved in a physical |
| fitness course at N.C. State University.                   |
| Only selected variables of                                 |
| Oxygen (oxygen intake, ml per kg body weight per minute),  |
| Runtime (time to run 1.5 miles in minutes), and            |
| RunPulse (heart rate while running) are used.              |
| Certain values were changed to missing for the analysis.   |
*------------------------------------------------------------*;
data FitMiss;
   input Oxygen RunTime RunPulse @@;
   datalines;
44.609  11.37  178     45.313  10.07  185
54.297   8.65  156     59.571    .      .
49.874   9.22    .     44.811  11.63  176
  .     11.95  176          .  10.85    .
39.442  13.08  174     60.055   8.63  170
50.541    .      .     37.388  14.03  186
44.754  11.12  176     47.273    .      .
51.855  10.33  166     49.156   8.95  180
40.836  10.95  168     46.672  10.00    .
46.774  10.25    .     50.388  10.08  168
39.407  12.63  174     46.080  11.17  156
45.441   9.63  164       .      8.92    .
45.118  11.08    .     39.203  12.88  168
45.790  10.47  186     50.545   9.93  148
48.673   9.40  186     47.920  11.50  170
47.467  10.50  170
;
```

Suppose that the data are multivariate normally distributed and that the missing data are missing at random (see the "Statistical Assumptions for Multiple Imputation" section in the chapter "The MI Procedure" for a description of these assumptions). The following statements use the MI procedure to impute missing values for the FitMiss data set:

```
proc mi data=FitMiss noprint out=outmi seed=3237851;
   var Oxygen RunTime RunPulse;
run;
```

The MI procedure creates imputed data sets, which are stored in the outmi data set. A variable named _Imputation_ indicates the imputation numbers. Based on m imputations, m different sets of the point and variance estimates for a parameter can be computed. In this example, $m = 5$ is the default.

The following statements generate regression coefficients for each of the five imputed data sets:

```
proc reg data=outmi outest=outreg covout noprint;
   model Oxygen= RunTime RunPulse;
   by _Imputation_;
run;
```

The following statements display (in Figure 55.1) output parameter estimates and covariance matrices from PROC REG for the first two imputed data sets:

```
proc print data=outreg(obs=8);
   var _Imputation_ _Type_ _Name_
      Intercept RunTime RunPulse;
   title 'Parameter Estimates from Imputed Data Sets';
run;
```

Figure 55.1 Parameter Estimates

Parameter Estimates from Imputed Data Sets

Obs	_Imputation_	_TYPE_	_NAME_	Intercept	RunTime	RunPulse
1	1	PARMS		86.544	-2.82231	-0.05873
2	1	COV	Intercept	100.145	-0.53519	-0.55077
3	1	COV	RunTime	-0.535	0.10774	-0.00345
4	1	COV	RunPulse	-0.551	-0.00345	0.00343
5	2	PARMS		83.021	-3.00023	-0.02491
6	2	COV	Intercept	79.032	-0.66765	-0.41918
7	2	COV	RunTime	-0.668	0.11456	-0.00313
8	2	COV	RunPulse	-0.419	-0.00313	0.00264

The following statements combine the five sets of regression coefficients:

```
proc mianalyze data=outreg;
   modeleffects Intercept RunTime RunPulse;
run;
```

The "Model Information" table in Figure 55.2 lists the input data set(s) and the number of imputations.

Figure 55.2 Model Information Table

The MIANALYZE Procedure

Model Information

Data Set	WORK.OUTREG
Number of Imputations	5

The "Variance Information" table in Figure 55.3 displays the between-imputation, within-imputation, and total variances for combining complete-data inferences. It also displays the degrees

of freedom for the total variance, the relative increase in variance due to missing values, the fraction of missing information, and the relative efficiency for each parameter estimate.

Figure 55.3 Variance Information Table

Variance Information

Parameter	Variance: Between	Variance: Within	Variance: Total	DF
Intercept	45.529229	76.543614	131.178689	23.059
RunTime	0.019390	0.106220	0.129487	123.88
RunPulse	0.001007	0.002537	0.003746	38.419

Variance Information

Parameter	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Intercept	0.713777	0.461277	0.915537
RunTime	0.219051	0.192620	0.962905
RunPulse	0.476384	0.355376	0.933641

The "Parameter Estimates" table in Figure 55.4 displays a combined estimate and standard error for each regression coefficient (parameter). Inferences are based on t distributions. The table displays a 95% confidence interval and a t test with the associated p-value for the hypothesis that the parameter is equal to the value specified with the THETA0= option (in this case, zero by default). The minimum and maximum parameter estimates from the imputed data sets are also displayed.

Figure 55.4 Parameter Estimates

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF
Intercept	90.837440	11.453327	67.14779	114.5271	23.059
RunTime	-3.032870	0.359844	-3.74511	-2.3206	123.88
RunPulse	-0.068578	0.061204	-0.19243	0.0553	38.419

Parameter Estimates

Parameter	Minimum	Maximum
Intercept	83.020730	100.839807
RunTime	-3.204426	-2.822311
RunPulse	-0.112840	-0.024910

Parameter Estimates

Parameter	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Intercept	0	7.93	<.0001
RunTime	0	-8.43	<.0001
RunPulse	0	-1.12	0.2695

Syntax: MIANALYZE Procedure

The following statements are available in PROC MIANALYZE:

PROC MIANALYZE < *options* > ;
  BY *variables* ;
  CLASS *variables* ;
  MODELEFFECTS *effects* ;
  < *label:* > **TEST** *equation*1 < , . . . , < *equationk* > > < */options* > ;
  STDERR *variables* ;

The BY statement specifies groups in which separate analyses are performed.

The CLASS statement lists the classification variables in the MODELEFFECTS statement. Classification variables can be either character or numeric.

The required MODELEFFECTS statement lists the effects to be analyzed. The variables in the statement that are not specified in a CLASS statement are assumed to be continuous.

The STDERR statement lists the standard errors associated with the effects in the MODELEFFECTS statement when both parameter estimates and standard errors are saved as variables in the same DATA= data set. The STDERR statement can be used only when each effect in the MODELEFFECTS statement is a continuous variable by itself.

The TEST statement tests linear hypotheses about the parameters. An F statistic is used to jointly test the null hypothesis ($H_0 : \mathbf{L}_{\text{ˮ}} = \mathbf{c}$) specified in a single TEST statement. Several TEST statements can be used.

The PROC MIANALYZE and MODELEFFECTS statements are required for the MIANALYZE procedure. The rest of this section provides detailed syntax information for each of these statements, beginning with the PROC MIANALYZE statement. The remaining statements are in alphabetical order.

PROC MIANALYZE Statement

PROC MIANALYZE < *options* > **;**

Table 55.1 summarizes the options in the PROC MIANALYZE statement.

Table 55.1 Summary of PROC MIANALYZE Options

Option	Description
Input Data Sets	
DATA=	specifies COV, CORR, or EST type data set
DATA=	specifies data set for parameter estimates and standard errors
PARMS=	specifies data set for parameter estimates
PARMINFO=	specifies data set for parameter information
COVB=	specifies data set for covariance matrices
XPXI=	specifies data set for $(X'X)^{-1}$ matrices
Statistical Analysis	
THETA0=	specifies parameters under the null hypothesis
ALPHA=	specifies level for the confidence interval
EDF=	specifies complete-data degrees of freedom
Printed Output	
WCOV	displays within-imputation covariance matrix
BCOV	displays between-imputation covariance matrix
TCOV	displays total covariance matrix
MULT	displays multivariate inferences

The following options can be used in the PROC MIANALYZE statement. They are listed in alphabetical order.

ALPHA=α

specifies that confidence limits are to be constructed for the parameter estimates with confidence level $100(1-\alpha)\%$, where $0 < \alpha < 1$. The default is ALPHA=0.05.

BCOV

displays the between-imputation covariance matrix.

COVB <(EFFECTVAR=STACKING | ROWCOL)> =*SAS-data-set*
names an input SAS data set that contains covariance matrices of the parameter estimates from imputed data sets. If you provide a COVB= data set, you must also provide a PARMS= data set.

The EFFECTVAR= option identifies the variables for parameters displayed in the covariance matrix and is used only when the PARMINFO= option is not specified. The default is EFFECTVAR= STACKING.

See the section "Input Data Sets" on page 3851 for a detailed description of the COVB= option.

DATA=*SAS-data-set*
names an input SAS data set.

If the input DATA= data set is not a specially structured SAS data set, the data set contains both the parameter estimates and associated standard errors. The parameter estimates are specified in the MODELEFFECTS statement and the standard errors are specified in the STDERR statement.

If the data set is a specially structured input SAS data set, it must have a TYPE of EST, COV, or CORR that contains estimates from imputed data sets:

- If TYPE=EST, the data set contains the parameter estimates and associated covariance matrices.
- If TYPE=COV, the data set contains the sample means, sample sizes, and covariance matrices. Each covariance matrix for variables is divided by the sample size n to create the covariance matrix for parameter estimates.
- If TYPE=CORR, the data set contains the sample means, sample sizes, standard errors, and correlation matrices. The covariance matrices are computed from the correlation matrices and associated standard errors. Each covariance matrix for variables is divided by the sample size n to create the covariance matrix for parameter estimates.

If you do not specify an input data set with the DATA= or PARMS= option, then the most recently created SAS data set is used as an input DATA= data set. See the section "Input Data Sets" on page 3851 for a detailed description of the input data sets.

EDF=*number*
specifies the complete-data degrees of freedom for the parameter estimates. This is used to compute an adjusted degrees of freedom for each parameter estimate. By default, EDF=∞ and the degrees of freedom for each parameter estimate are not adjusted.

MULT

MULTIVARIATE
requests multivariate inference for the parameters. It is based on Wald tests and is a generalization of the univariate inference. See the section "Multivariate Inferences" on page 3857 for a detailed description of the multivariate inference.

PARMINFO=*SAS-data-set*

names an input SAS data set that contains parameter information associated with variables PRM1, PRM2,..., and so on. These variables are used as variables for parameters in a COVB= data set. See the section "Input Data Sets" on page 3851 for a detailed description of the PARMINFO= option.

PARMS < (CLASSVAR= *ctype*) > =*SAS-data-set*

names an input SAS data set that contains parameter estimates computed from imputed data sets. When a COVB= data set is not specified, the input PARMS= data set also contains standard errors associated with these parameter estimates. If multivariate inference is requested, you must also provide a COVB= or XPXI= data set.

When the effects contain classification variables, the option CLASSVAR= *ctype* can be used to identify the associated classification variables when reading the classification levels from observations. The available types are FULL, LEVEL, and CLASSVAL. The default is CLASSVAR= FULL. See the section "Input Data Sets" on page 3851 for a detailed description of the PARMS= option.

TCOV

displays the total covariance matrix derived by assuming that the population between-imputation and within-imputation covariance matrices are proportional to each other.

THETA0=*numbers*

MU0=*numbers*

specifies the parameter values $\boldsymbol{\theta}_0$ under the null hypothesis $\boldsymbol{\theta} = \boldsymbol{\theta}_0$ in the t tests for location for the effects. If only one number $\boldsymbol{\theta}_0$ is specified, that number is used for all effects. If more than one number is specified, the specified numbers correspond to effects in the MODELEFFECTS statement in the order in which they appear in the statement. When an effect contains classification variables, the corresponding value is not used and the test is not performed.

WCOV

displays the within-imputation covariance matrices.

XPXI=*SAS-data-set*

names an input SAS data set that contains the $(X'X)^{-1}$ matrices associated with the parameter estimates computed from imputed data sets. If you provide an XPXI= data set, you must also provide a PARMS= data set. In this case, PROC MIANALYZE reads the standard errors of the estimates from the PARMS= data. The standard errors and $(X'X)^{-1}$ matrices are used to derive the covariance matrices.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MIANALYZE to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the MI procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

The CLASS statement specifies the classification variables in the MODELEFFECTS statement. Classification variables can be either character or numeric. Classification levels are determined from the formatted values of the classification variables. See "The FORMAT Procedure" in the *Base SAS Procedures Guide* for details.

MODELEFFECTS Statement

MODELEFFECTS *effects* **;**

The MODELEFFECTS statement lists the effects in the data set to be analyzed. Each effect is a variable or a combination of variables, and is specified with a special notation that uses variable names and operators.

Each variable is either a classification (or CLASS) variable or a continuous variable. If a variable is not declared in the CLASS statement, it is assumed to be continuous. Crossing and nesting operators can be used in an effect to create crossed and nested effects.

One general form of an effect involving several variables is

X1 ∗ *X2* ∗ *A* ∗ *B* ∗ *C* (*D E*)

where A, B, C, D, and E are classification variables and X1 and X2 are continuous variables.

When the input DATA= data set is not a specially structured SAS data set, you must also specify standard errors of the parameter estimates in an STDERR statement.

STDERR Statement

STDERR *variables* **;**

The STDERR statement lists standard errors associated with effects in the MODELEFFECTS statement, when the input DATA= data set contains both parameter estimates and standard errors as variables in the data set.

With the STDERR statement, only continuous effects are allowed in the MODELEFFECTS statement. The specified standard errors correspond to parameter estimates in the order in which they appear in the MODELEFFECTS statement.

For example, you can use the following MODELEFFECTS and STDERR statements to identify both the parameter estimates and associated standard errors in a SAS data set:

```
proc mianalyze;
   modeleffects y1-y3;
   stderr sy1-sy3;
run;
```

TEST Statement

< *label:* > **TEST** $equation1 <, \ldots, < equationk > > < / options >$ **;**

The TEST statement tests linear hypotheses about the parameters $\boldsymbol{\beta}$. An F test is used to jointly test the null hypotheses ($H_0 : \mathbf{L}\boldsymbol{\beta} = \mathbf{c}$) specified in a single TEST statement in which the MULT option is specified.

Each *equation* specifies a linear hypothesis (a row of the **L** matrix and the corresponding element of the **c** vector); multiple *equations* are separated by commas. The label, which must be a valid SAS name, is used to identify the resulting output. You can submit multiple TEST statements. When a label is not included in a TEST statement, a label of "Test j" is used for the jth TEST statement.

The form of an *equation* is as follows:

term < ± *term* ... > < = ± *term* < ± *term* ... > >

where *term* is a parameter of the model, or a constant, or a constant times a parameter. When no equal sign appears, the expression is set to 0. Only parameters for regressor effects (continuous variables by themselves) are allowed.

For each TEST statement, PROC MIANALYZE displays a "Test Specification" table of the **L** matrix and the **c** vector. The procedure also displays a "Variance Information" table of the between-imputation, within-imputation, and total variances for combining complete-data inferences, and a "Parameter Estimates" table of a combined estimate and standard error for each linear component. The linear components are labeled TestPrm1, TestPrm2, ... in the tables.

The following statements illustrate possible uses of the TEST statement:

```
proc mianalyze;
   modeleffects intercept a1 a2 a3;
   test1: test intercept + a2 = 0;
   test2: test intercept + a2;
   test3: test a1=a2=a3;
   test4: test a1=a2, a2=a3;
run;
```

The first and second TEST statements are equivalent and correspond to the specification in Figure 55.5.

Figure 55.5 Test Specification for test1 and test2

The MIANALYZE Procedure
Test: test1

Test Specification

	L Matrix			
Parameter	intercept	a1	a2	a3	C
TestPrm1	1.000000	0	1.000000	0	0

The third and fourth TEST statements are also equivalent and correspond to the specification in Figure 55.6.

Figure 55.6 Test Specification for test3 and test4

The MIANALYZE Procedure
Test: test3

Test Specification

	L Matrix				
Parameter	intercept	a1	a2	a3	C
TestPrm1	0	1.000000	-1.000000	0	0
TestPrm2	0	0	1.000000	-1.000000	0

The ALPHA= and EDF options specified in the PROC MIANALYZE statement are also applied to the TEST statement. You can specify the following options in the TEST statement after a slash(/):

BCOV
displays the between-imputation covariance matrix.

MULT
displays the multivariate inference for parameters.

TCOV
displays the total covariance matrix.

WCOV

displays the within-imputation covariance matrix.

For more information, see the section "Testing Linear Hypotheses about the Parameters" on page 3859.

Details: MIANALYZE Procedure

Input Data Sets

You specify input data sets based on the type of inference you requested. For univariate inference, you can use one of the following options:

- a DATA= data set, which provides both parameter estimates and the associated standard errors
- a DATA=EST, COV, or CORR data set, which provides both parameter estimates and the associated standard errors either explicitly (type CORR) or through the covariance matrix (type EST, COV)
- PARMS= data set, which provides both parameter estimates and the associated standard errors

For multivariate inference, which includes the testing of linear hypotheses about parameters, you can use one of the following option combinations:

- a DATA=EST, COV, or CORR data set, which provides parameter estimates and the associated covariance matrix either explicitly (type EST, COV) or through the correlation matrix and standard errors (type CORR) in a single data set
- PARMS= and COVB= data sets, which provide parameter estimates in a PARMS= data set and the associated covariance matrix in a COVB= data set
- PARMS=, COVB=, and PARMINFO= data sets, which provide parameter estimates in a PARMS= data set, the associated covariance matrix in a COVB= data set with variables named PRM1, PRM2, ..., and the effects associated with these variables in a PARMINFO= data set
- PARMS= and XPXI= data sets, which provide parameter estimates and the associated standard errors in a PARMS= data set and the associated $(X'X)^{-1}$ matrix in an XPXI= data set

The appropriate combination depends on the type of inference and the SAS procedure you used to create the data sets. For instance, if you used PROC REG to create an OUTEST= data set

containing the parameter estimates and covariance matrix, you would use the DATA= option to read the OUTEST= data set.

When the input DATA= data set is a specially structured SAS data set, the data set must contain the variable _Imputation_ to identify the imputation by number. Otherwise, each observation corresponds to an imputation and contains both parameter estimates and associated standard errors.

If you do not specify an input data set with the DATA= or PARMS= option, then the most recently created SAS data set is used as an input DATA= data set. Note that with a DATA= data set, each effect represents a continuous variable; only regressor effects (continuous variables by themselves) are allowed in the MODELEFFECTS statement.

DATA= SAS Data Set

The DATA= data set provides both parameter estimates and the associated standard errors computed from imputed data sets. Such data sets are typically created with an OUTPUT statement in procedures such as PROC MEANS and PROC UNIVARIATE.

The MIANALYZE procedure reads parameter estimates from observations with variables in the MODELEFFECTS statement, and standard errors for parameter estimates from observations with variables in the STDERR statement. The order of the variables for standard errors must match the order of the variables for parameter estimates.

DATA=EST, COV, or CORR SAS Data Set

The specially structured DATA= data set provides both parameter estimates and the associated covariance matrix computed from imputed data sets. Such data sets are created by procedures such as PROC CORR (type COV, CORR) and PROC REG (type EST).

With TYPE=EST, the MIANALYZE procedure reads parameter estimates from observations with _TYPE_='PARM', _TYPE_='PARMS', _TYPE_='OLS', or _TYPE_='FINAL', and covariance matrices for parameter estimates from observations with _TYPE_='COV' or _TYPE_='COVB'.

With TYPE=COV, the procedure reads sample means from observations with _TYPE_='MEAN', sample size n from observations with _TYPE_='N', and covariance matrices for variables from observations with _TYPE_='COV'.

With TYPE=CORR, the procedure reads sample means from observations with _TYPE_='MEAN', sample size n from observations with _TYPE_='N', correlation matrices for variables from observations with _TYPE_='CORR', and standard errors for variables from observations with _TYPE_='STD'. The standard errors and correlation matrix are used to generate a covariance matrix for the variables.

Note that with TYPE=COV or CORR, each covariance matrix for the variables is divided by n to create the covariance matrix for the sample means.

PARMS <(CLASSVAR= *ctype*)> = Data Set

The PARMS= data set contains both parameter estimates and the associated standard errors computed from imputed data sets. Such data sets are typically created with an ODS OUTPUT statement in procedures such as PROC GENMOD, PROC GLM, PROC LOGISTIC, and PROC MIXED.

The MIANALYZE procedure reads effect names from observations with the variable Parameter, Effect, Variable, or Parm. It then reads parameter estimates from observations with the variable Estimate and standard errors for parameter estimates from observations with the variable StdErr.

When the effects contain classification variables, the option CLASSVAR= *ctype* can be used to identify associated classification variables when reading the classification levels from observations. The available types are FULL, LEVEL, and CLASSVAL. The default is CLASSVAR= FULL.

With CLASSVAR=FULL, the data set contains the classification variables explicitly. PROC MIANALYZE reads the classification levels from observations with their corresponding classification variables. PROC MIXED generates this type of table.

With CLASSVAR=LEVEL, PROC MIANALYZE reads the classification levels for the effect from observations with variables Level1, Level2, and so on, where the variable Level1 contains the classification level for the first classification variable in the effect, and the variable Level2 contains the classification level for the second classification variable in the effect. For each effect, the variables in the crossed list are displayed before the variables in the nested list. The variable order in the CLASS statement is used for variables inside each list. PROC GENMOD generates this type of table.

For example, with the following statements, the variable Level1 has the classification level of the variable c2 for the effect c2:

```
proc mianalyze parms(classvar=Level)= dataparm;
   class c1 c2 c3;
   modeleffects c2 c3(c2 c1);
run;
```

For the effect c3(c2 c1), the variable Level1 has the classification level of the variable c3, Level2 has the level of c1, and Level3 has the level of c2.

Similarly, with CLASSVAR=CLASSVAL, PROC MIANALYZE reads the classification levels for the effect from observations with variables ClassVal0, ClassVal1, and so on, where the variable ClassVal0 contains the classification level for the first classification variable in the effect, and the variable ClassVal1 contains the classification level for the second classification variable in the effect. For each effect, the variables in the crossed list are displayed before the variables in the nested list. The variable order in the CLASS statement is used for variables inside each list. PROC LOGISTIC generates this type of tables.

PARMS <(CLASSVAR= *ctype*)> = and COVB= Data Sets

The PARMS= data set contains parameter estimates, and the COVB= data set contains associated covariance matrices computed from imputed data sets. Such data sets are typically created with

an ODS OUTPUT statement in procedures such as PROC LOGISTIC, PROC MIXED, and PROC REG.

With a PARMS= data set, the MIANALYZE procedure reads effect names from observations with the variable Parameter, Effect, Variable, or Parm. It then reads parameter estimates from observations with the variable Estimate.

When the effects contain classification variables, the option CLASSVAR= *ctype* can be used to identify the associated classification variables when reading the classification levels from observations. The available types are FULL, LEVEL, and CLASSVAL, and they are described in the section "PARMS <(CLASSVAR= *ctype*)> = Data Set" on page 3853. The default is CLASSVAR= FULL.

The option EFFECTVAR=*etype* identifies the variables for parameters displayed in the covariance matrix. The available types are STACKING and ROWCOL. The default is EFFECTVAR=STACKING.

With EFFECTVAR=STACKING, each parameter is displayed by stacking variables in the effect. Begin with the variables in the crossed list, followed by the continuous list, then followed by the nested list. Each classification variable is displayed with its classification level attached. PROC LOGISTIC generates this type of table.

When each effect is a continuous variable by itself, each stacked parameter name reduces to the effect name. PROC REG generates this type of table.

With EFFECTVAR=STACKING, the MIANALYZE procedure reads parameter names from observations with the variable Parameter, Effect, Variable, Parm, or RowName. It then reads covariance matrices from observations with the stacked variables in a COVB= data set.

With EFFECTVAR=ROWCOL, parameters are displayed by the variables Col1, Col2, ... The parameter associated with the variable Col1 is identified by the observation with value 1 for the variable Row. The parameter associated with the variable Col2 is identified by the observation with value 2 for the variable Row. PROC MIXED generates this type of table.

With EFFECTVAR=ROWCOL, the MIANALYZE procedure reads the parameter indices from observations with the variable Row and the effect names from observations with the variable Parameter, Effect, Variable, Parm, or RowName. It then reads covariance matrices from observations with the variables Col1, Col2, and so on in a COVB= data set.

When the effects contain classification variables, the data set contains the classification variables explicitly and the MIANALYZE procedure also reads the classification levels from their corresponding classification variables.

PARMS <(CLASSVAR= *ctype*)> =, PARMINFO=, and COVB= Data Sets

The input PARMS= data set contains parameter estimates and the input COVB= data set contains associated covariance matrices computed from imputed data sets. Such data sets are typically created with an ODS OUTPUT statement using procedure such as PROC GENMOD.

With a PARMS= data set, the MIANALYZE procedure reads effect names from observations with the variable Parameter, Effect, Variable, or Parm. It then reads parameter estimates from observations with the variable Estimate.

When the effects contain classification variables, the option CLASSVAR= *ctype* can be used to identify the associated classification variables when reading the classification levels from observations. The available types are FULL, LEVEL, and CLASSVAL, and they are described in the section "PARMS <(CLASSVAR= *ctype*)> = Data Set" on page 3853. The default is CLASSVAR= FULL.

With a COVB= data set, the MIANALYZE procedure reads parameter names from observations with the variable Parameter, Effect, Variable, Parm, or RowName. It then reads covariance matrices from observations with the variables Prm1, Prm2, and so on.

The parameters associated with the variables Prm1, Prm2, and so on are identified in the PARMINFO= data set. PROC MIANALYZE reads the parameter names from observations with the variable Parameter and the corresponding effect from observations with the variable Effect. When the effects contain classification variables, the data set contains the classification variables explicitly and the MIANALYZE procedure also reads the classification levels from observations with their corresponding classification variables.

PARMS= and XPXI= Data Sets

The input PARMS= data set contains parameter estimates, and the input XPXI= data set contains associated $(X'X)^{-1}$ matrices computed from imputed data sets. Such data sets are typically created with an ODS OUTPUT statement in a procedure such as PROC GLM.

With a PARMS= data set, the MIANALYZE procedure reads parameter names from observations with the variable Parameter, Effect, Variable, or Parm. It then reads parameter estimates from observations with the variable Estimate and standard errors for parameter estimates from observations with the variable StdErr.

With a XPXI= data set, the MIANALYZE procedure reads parameter names from observations with the variable Parameter and $(X'X)^{-1}$ matrices from observations with the parameter variables in the data set.

Note that this combination can be used only when each effect is a continuous variable by itself.

Combining Inferences from Imputed Data Sets

With m imputations, m different sets of the point and variance estimates for a parameter Q can be computed. Suppose that $\hat{Q}_i$ and $\hat{W}_i$ are the point and variance estimates, respectively, from the ith imputed data set, $i = 1, 2, \ldots, m$. Then the combined point estimate for Q from multiple imputation is the average of the m complete-data estimates:

$$\overline{Q} = \frac{1}{m}\sum_{i=1}^{m}\hat{Q}_i$$

Suppose that $\overline{W}$ is the within-imputation variance, which is the average of the m complete-data estimates:

$$\overline{W} = \frac{1}{m}\sum_{i=1}^{m}\hat{W}_i$$

And suppose that B is the between-imputation variance:

$$B = \frac{1}{m-1}\sum_{i=1}^{m}(\hat{Q}_i - \overline{Q})^2$$

Then the variance estimate associated with $\overline{Q}$ is the total variance (Rubin 1987)

$$T = \overline{W} + (1 + \frac{1}{m})B$$

The statistic $(Q - \overline{Q})T^{-(1/2)}$ is approximately distributed as t with v_m degrees of freedom (Rubin 1987), where

$$v_m = (m-1)\left[1 + \frac{\overline{W}}{(1+m^{-1})B}\right]^2$$

The degrees of freedom v_m depend on m and the ratio

$$r = \frac{(1+m^{-1})B}{\overline{W}}$$

The ratio r is called the relative increase in variance due to nonresponse (Rubin 1987). When there is no missing information about Q, the values of r and B are both zero. With a large value of m or a small value of r, the degrees of freedom v_m will be large and the distribution of $(Q - \overline{Q})T^{-(1/2)}$ will be approximately normal.

Another useful statistic is the fraction of missing information about Q:

$$\hat{\lambda} = \frac{r + 2/(v_m + 3)}{r + 1}$$

Both statistics r and λ are helpful diagnostics for assessing how the missing data contribute to the uncertainty about Q.

When the complete-data degrees of freedom v_0 are small, and there is only a modest proportion of missing data, the computed degrees of freedom, v_m, can be much larger than v_0, which is inappropriate. For example, with $m = 5$ and $r = 10\%$, the computed degrees of freedom $v_m = 484$, which is inappropriate for data sets with complete-data degrees of freedom less than 484.

Barnard and Rubin (1999) recommend the use of adjusted degrees of freedom

$$v_m^* = \left[\frac{1}{v_m} + \frac{1}{\hat{v}_{obs}}\right]^{-1}$$

where $\hat{v}_{obs} = (1-\gamma)\, v_0(v_0+1)/(v_0+3)$ and $\gamma = (1+m^{-1})B/T$.

If you specify the complete-data degrees of freedom v_0 with the EDF= option, the MIANALYZE procedure uses the adjusted degrees of freedom, v_m^*, for inference. Otherwise, the degrees of freedom v_m are used.

Multiple Imputation Efficiency

The relative efficiency (RE) of using the finite m imputation estimator, rather than using an infinite number for the fully efficient imputation, in units of variance, is approximately a function of m and λ (Rubin 1987, p. 114):

$$RE = (1 + \frac{\lambda}{m})^{-1}$$

Table 55.2 shows relative efficiencies with different values of m and λ.

Table 55.2 Relative Efficiencies

	λ				
m	10%	20%	30%	50%	70%
3	0.9677	0.9375	0.9091	0.8571	0.8108
5	0.9804	0.9615	0.9434	0.9091	0.8772
10	0.9901	0.9804	0.9709	0.9524	0.9346
20	0.9950	0.9901	0.9852	0.9756	0.9662

The table shows that for situations with little missing information, only a small number of imputations are necessary. In practice, the number of imputations needed can be informally verified by replicating sets of m imputations and checking whether the estimates are stable between sets (Horton and Lipsitz 2001, p. 246).

Multivariate Inferences

Multivariate inference based on Wald tests can be done with m imputed data sets. The approach is a generalization of the approach taken in the univariate case (Rubin 1987, p. 137; Schafer 1997, p. 113). Suppose that $\hat{\mathbf{Q}}_i$ and $\hat{\mathbf{W}}_i$ are the point and covariance matrix estimates for a p-dimensional parameter $\mathbf{Q}$ (such as a multivariate mean) from the ith imputed data set, $i = 1, 2, \ldots, m$. Then the combined point estimate for $\mathbf{Q}$ from the multiple imputation is the average of the m complete-data estimates:

$$\overline{\mathbf{Q}} = \frac{1}{m} \sum_{i=1}^{m} \hat{\mathbf{Q}}_i$$

Suppose that $\overline{\mathbf{W}}$ is the within-imputation covariance matrix, which is the average of the m complete-data estimates:

$$\overline{\mathbf{W}} = \frac{1}{m} \sum_{i=1}^{m} \hat{\mathbf{W}}_i$$

And suppose that $\mathbf{B}$ is the between-imputation covariance matrix:

$$\mathbf{B} = \frac{1}{m-1}\sum_{i=1}^{m}(\hat{\mathbf{Q}}_i - \overline{\mathbf{Q}})(\hat{\mathbf{Q}}_i - \overline{\mathbf{Q}})'$$

Then the covariance matrix associated with $\overline{\mathbf{Q}}$ is the total covariance matrix

$$\mathbf{T}_0 = \overline{\mathbf{W}} + (1 + \frac{1}{m})\mathbf{B}$$

The natural multivariate extension of the t statistic used in the univariate case is the F statistic

$$F_0 = (\mathbf{Q} - \overline{\mathbf{Q}})'\mathbf{T}_0^{-1}(\mathbf{Q} - \overline{\mathbf{Q}})$$

with degrees of freedom p and

$$v = (m-1)(1 + 1/r)^2$$

where

$$r = (1 + \frac{1}{m})\,\mathrm{trace}(\mathbf{B}\overline{\mathbf{W}}^{-1})/p$$

is an average relative increase in variance due to nonresponse (Rubin 1987, p. 137; Schafer 1997, p. 114).

However, the reference distribution of the statistic F_0 is not easily derived. Especially for small m, the between-imputation covariance matrix $\mathbf{B}$ is unstable and does not have full rank for $m \leq p$ (Schafer 1997, p. 113).

One solution is to make an additional assumption that the population between-imputation and within-imputation covariance matrices are proportional to each other (Schafer 1997, p. 113). This assumption implies that the fractions of missing information for all components of $\mathbf{Q}$ are equal. Under this assumption, a more stable estimate of the total covariance matrix is

$$\mathbf{T} = (1 + r)\overline{\mathbf{W}}$$

With the total covariance matrix $\mathbf{T}$, the F statistic (Rubin 1987, p. 137)

$$F = (\mathbf{Q} - \overline{\mathbf{Q}})'\mathbf{T}^{-1}(\mathbf{Q} - \overline{\mathbf{Q}})/p$$

has an F distribution with degrees of freedom p and v_1, where

$$v_1 = \frac{1}{2}(p+1)(m-1)(1 + \frac{1}{r})^2$$

For $t = p(m-1) \leq 4$, PROC MIANALYZE uses the degrees of freedom v_1 in the analysis. For $t = p(m-1) > 4$, PROC MIANALYZE uses v_2, a better approximation of the degrees of freedom given by Li, Raghunathan, and Rubin (1991):

$$v_2 = 4 + (t-4)\left[1 + \frac{1}{r}(1 - \frac{2}{t})\right]^2$$

Testing Linear Hypotheses about the Parameters

Linear hypotheses for parameters $\boldsymbol{\beta}$ are expressed in matrix form as

$$H_0 : \mathbf{L}\boldsymbol{\beta} = \mathbf{c}$$

where $\mathbf{L}$ is a matrix of coefficients for the linear hypotheses and $\mathbf{c}$ is a vector of constants.

Suppose that $\hat{\mathbf{Q}}_i$ and $\hat{\mathbf{U}}_i$ are the point and covariance matrix estimates, respectively, for a p-dimensional parameter $\mathbf{Q}$ from the ith imputed data set, i=1, 2, ..., m. Then for a given matrix $\mathbf{L}$, the point and covariance matrix estimates for the linear functions $\mathbf{LQ}$ in the ith imputed data set are, respectively,

$$\mathbf{L}\hat{\mathbf{Q}}_i$$

$$\mathbf{L}\hat{\mathbf{U}}_i\mathbf{L}'$$

The inferences described in the section "Combining Inferences from Imputed Data Sets" on page 3855 and the section "Multivariate Inferences" on page 3857 are applied to these linear estimates for testing the null hypothesis $H_0 : \mathbf{L}\boldsymbol{\beta} = \mathbf{c}$.

For each TEST statement, the "Test Specification" table displays the $\mathbf{L}$ matrix and the $\mathbf{c}$ vector, the "Variance Information" table displays the between-imputation, within-imputation, and total variances for combining complete-data inferences, and the "Parameter Estimates" table displays a combined estimate and standard error for each linear component.

With the WCOV and BCOV options in the TEST statement, the procedure displays the within-imputation and between-imputation covariance matrices, respectively.

With the TCOV option, the procedure displays the total covariance matrix derived under the assumption that the population between-imputation and within-imputation covariance matrices are proportional to each other.

With the MULT option in the TEST statement, the "Multivariate Inference" table displays an F test for the null hypothesis $\mathbf{L}\boldsymbol{\beta} = \mathbf{c}$ of the linear components.

Examples of the Complete-Data Inferences

For a given parameter of interest, it is not always possible to compute the estimate and associated covariance matrix directly from a SAS procedure. This section describes examples of parameters with their estimates and associated covariance matrices, which provide the input to the MIANALYZE procedure. Some are straightforward, and others require special techniques.

Means

For a population mean vector $\boldsymbol{\mu}$, the usual estimate is the sample mean vector

$$\bar{\mathbf{y}} = \frac{1}{n}\sum \mathbf{y}_i$$

A variance estimate for $\bar{\mathbf{y}}$ is $\frac{1}{n}\mathbf{S}$, where $\mathbf{S}$ is the sample covariance matrix

$$\mathbf{S} = \frac{1}{n-1}\sum(\mathbf{y}_i - \bar{\mathbf{y}})(\mathbf{y}_i - \bar{\mathbf{y}})'$$

These statistics can be computed from a procedure such as PROC CORR. This approach is illustrated in Example 55.2.

Regression Coefficients

Many SAS procedures are available for regression analysis. Among them, PROC REG provides the most general analysis capabilities, and others like PROC LOGISTIC and PROC MIXED provide more specialized analyses.

Some regression procedures, such as REG and LOGISTIC, create an EST type data set that contains both the parameter estimates for the regression coefficients and their associated covariance matrix. You can read an EST type data set in the MIANALYZE procedure with the DATA= option. This approach is illustrated in Example 55.3.

Other procedures, such as GLM, MIXED, and GENMOD, do not generate EST type data sets for regression coefficients. For PROC MIXED and PROC GENMOD, you can use ODS OUTPUT statement to save parameter estimates in a data set and the associated covariance matrix in a separate data set. These data sets are then read in the MIANALYZE procedure with the PARMS= and COVB= options, respectively. This approach is illustrated in Example 55.4 for PROC MIXED and in Example 55.5 for PROC GENMOD.

PROC GLM does not display tables for covariance matrices. However, you can use the ODS OUTPUT statement to save parameter estimates and associated standard errors in a data set and the associated $(X'X)^{-1}$ matrix in a separate data set. These data sets are then read in the MIANALYZE procedure with the PARMS= and XPXI= options, respectively. This approach is illustrated in Example 55.6.

For univariate inference, only parameter estimates and associated standard errors are needed. You can use the ODS OUTPUT statement to save parameter estimates and associated standard errors in a data set. This data set is then read in the MIANALYZE procedure with the PARMS= option. This approach is illustrated in Example 55.4.

Correlation Coefficients

For the population correlation coefficient ρ, a point estimate is the sample correlation coefficient r. However, for nonzero ρ, the distribution of r is skewed.

The distribution of r can be normalized through Fisher's z transformation

$$z(r) = \frac{1}{2}\log\left(\frac{1+r}{1-r}\right)$$

$z(r)$ is approximately normally distributed with mean $z(\rho)$ and variance $1/(n-3)$.

With a point estimate $\hat{z}$ and an approximate 95% confidence interval (z_1, z_2) for $z(\rho)$, a point estimate $\hat{r}$ and a 95% confidence interval (r_1, r_2) for ρ can be obtained by applying the inverse transformation

$$r = \tanh(z) = \frac{e^{2z} - 1}{e^{2z} + 1}$$

to $z = \hat{z}, z_1$, and z_2.

This approach is illustrated in Example 55.10.

Ratios of Variable Means

For the ratio μ_1/μ_2 of means for variables Y_1 and Y_2, the point estimate is $\overline{y}_1/\overline{y}_2$, the ratio of the sample means. The Taylor expansion and delta method can be applied to the function y_1/y_2 to obtain the variance estimate (Schafer 1997, p. 196)

$$\frac{1}{n}\left[\left(\frac{\overline{y}_1}{\overline{y}_2^2}\right)^2 s_{22} - 2\left(\frac{\overline{y}_1}{y_2^2}\right)\left(\frac{1}{\overline{y}_2}\right)s_{12} + \left(\frac{1}{\overline{y}_2}\right)^2 s_{11}\right]$$

where s_{11} and s_{22} are the sample variances of Y_1 and Y_2, respectively, and s_{12} is the sample covariance between Y_1 and Y_2.

A ratio of sample means will be approximately unbiased and normally distributed if the coefficient of variation of the denominator (the standard error for the mean divided by the estimated mean) is 10% or less (Cochran 1977, p. 166; Schafer 1997, p. 196).

ODS Table Names

PROC MIANALYZE assigns a name to each table it creates. You must use these names to reference tables when using the Output Delivery System (ODS). These names are listed in Table 55.3. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 55.3 ODS Tables Produced by PROC MIANALYZE

ODS Table Name	Description	Statement	Option
BCov	Between-imputation covariance matrix		BCOV
ModelInfo	Model information		
MultStat	Multivariate inference		MULT
ParameterEstimates	Parameter estimates		
TCov	Total covariance matrix		TCOV
TestBCov	Between-imputation covariance matrix for $\mathbf{L}\boldsymbol{\beta}$	TEST	BCOV
TestMultStat	Multivariate inference for $\mathbf{L}\boldsymbol{\beta}$	TEST	MULT
TestParameterEstimates	Parameter estimates for $\mathbf{L}\boldsymbol{\beta}$	TEST	
TestSpec	Test specification, $\mathbf{L}$ and $\mathbf{c}$	TEST	
TestTCov	Total covariance matrix for $\mathbf{L}\boldsymbol{\beta}$	TEST	TCOV

Table 55.3 *continued*

ODS Table Name	Description	Statement	Option
TestVarianceInfo	Variance information for $\mathbf{L}\boldsymbol{\beta}$	TEST	
TestWCov	Within-imputation covariance matrix for $\mathbf{L}\boldsymbol{\beta}$	TEST	WCOV
VarianceInfo	Variance information		
WCov	Within-imputation covariance matrix		WCOV

Examples: MIANALYZE Procedure

The following statements generate five imputed data sets to be used in this section. The data set FitMiss was created in the section "Getting Started: MIANALYZE Procedure" on page 3841. See "The MI Procedure" chapter for details concerning the MI procedure.

```
proc mi data=FitMiss seed=3237851 noprint out=outmi;
   var Oxygen RunTime RunPulse;
run;
```

The Fish data described in the STEPDISC procedure are measurements of 159 fish of seven species caught in Finland's lake Laengelmavesi. For each fish, the length, height, and width are measured. Three different length measurements are recorded: from the nose of the fish to the beginning of its tail (Length1), from the nose to the notch of its tail (Length2), and from the nose to the end of its tail (Length3). See Chapter 82, "The STEPDISC Procedure," for more information.

The Fish2 data set is constructed from the Fish data set and contains two species of fish. Some values have been set to missing, and the resulting data set has a monotone missing pattern in the variables Length3, Height, Width, and Species. Note that some values of the variable Species have also been altered in the data set.

The following statements create the Fish2 data set. It contains two species of fish in the Fish data set.

```
/*-------- Fish of Species Bream and Roach --------*/
 data Fish2 (drop=HtPct WidthPct);
 title 'Fish Measurement Data';
 input Species $ Length3 HtPct WidthPct @@;
 Height= HtPct*Length3/100;
 Width= WidthPct*Length3/100;
 datalines;
 Gp1  30.0 38.4 13.4   Gp1  31.2 40.0 13.8   Gp1  31.1 39.8 15.1
   .  33.5 38.0   .       .  34.0 36.6 15.1   Gp1  34.7 39.2 14.2
 Gp1  34.5 41.1 15.3   Gp1  35.0 36.2 13.4   Gp1  35.1 39.9 13.8
   .  36.2 39.3 13.7   Gp1  36.2 39.4 14.1     .  36.2 39.7 13.3
 Gp1  36.4 37.8 12.0     .  37.3 37.3 13.6   Gp1  37.2 40.2 13.9
 Gp1  37.2 41.5 15.0   Gp1  38.3 38.8 13.8   Gp1  38.5 38.8 13.5
 Gp1  38.6 40.5 13.3   Gp1  38.7 37.4 14.8   Gp1  39.5 38.3 14.1
 Gp1  39.2 40.8 13.7     .  39.7 39.1   .    Gp1  40.6 38.1 15.1
```

```
Gp1  40.5 40.1 13.8   Gp1  40.9 40.0 14.8   Gp1  40.6 40.3 15.0
Gp1  41.5 39.8 14.1   Gp2  41.6 40.6 14.9   Gp1  42.6 44.5 15.5
Gp1  44.1 40.9 14.3   Gp1  44.0 41.1 14.3   Gp1  45.3 41.4 14.9
Gp1  45.9 40.6 14.7   Gp1  46.5 37.9 13.7
Gp2  16.2 25.6 14.0   Gp2  20.3 26.1 13.9   Gp2  21.2 26.3 13.7
Gp2  22.2 25.3 14.3   Gp2  22.2 28.0 16.1   Gp2  22.8 28.4 14.7
Gp2  23.1 26.7 14.7     .  23.7 25.8 13.9   Gp2  24.7 23.5 15.2
Gp2  24.3 27.3 14.6   Gp2  25.3 27.8 15.1   Gp2  25.0 26.2 13.3
Gp2  25.0 25.6 15.2   Gp2  27.2 27.7 14.1   Gp2  26.7 25.9 13.6
  .  26.8 27.6 15.4   Gp2  27.9 25.4 14.0   Gp2  29.2 30.4 15.4
Gp2  30.6 28.0 15.6   Gp2  35.0 27.1 15.3
;
```

The following statements generate five imputed data sets to be used in this section. The regression method is used to impute missing values in the variable Width and the discriminant function method is used to impute the variable Species.

```
proc mi data=Fish2 seed=1305417 out=outfish;
   class Species;
   monotone reg (Width)
            discrim( Species= Length3 Height Width);
   var Length3 Height Width Species;
run;
```

Example 55.1 through Example 55.6 use different input option combinations to combine parameter estimates computed from different procedures. Example 55.7 and Example 55.8 combine parameter estimates with classification variables. Example 55.9 shows the use of a TEST statement, and Example 55.10 combines statistics that are not directly derived from procedures.

Example 55.1: Reading Means and Standard Errors from Variables in a DATA= Data Set

This example creates an ordinary SAS data set that contains sample means and standard errors computed from imputed data sets. These estimates are then combined to generate valid univariate inferences about the population means.

The following statements use the UNIVARIATE procedure to generate sample means and standard errors for the variables in each imputed data set:

```
proc univariate data=outmi noprint;
   var Oxygen RunTime RunPulse;
   output out=outuni mean=Oxygen RunTime RunPulse
                     stderr=SOxygen SRunTime SRunPulse;
   by _Imputation_;
run;
```

The following statements display the output data set from PROC UNIVARIATE shown in Output 55.1.1:

```
proc print data=outuni;
   title 'UNIVARIATE Means and Standard Errors';
run;
```

Output 55.1.1 UNIVARIATE Output Data Set

UNIVARIATE Means and Standard Errors

Obs	_Imputation_	Oxygen	RunTime	Run Pulse	SOxygen	SRun Time	SRun Pulse
1	1	47.0120	10.4441	171.216	0.95984	0.28520	1.59910
2	2	47.2407	10.5040	171.244	0.93540	0.26661	1.75638
3	3	47.4995	10.5922	171.909	1.00766	0.26302	1.85795
4	4	47.1485	10.5279	171.146	0.95439	0.26405	1.75011
5	5	47.0042	10.4913	172.072	0.96528	0.27275	1.84807

The following statements combine the means and standard errors from imputed data sets, The EDF= option requests that the adjusted degrees of freedom be used in the analysis. For sample means based on 31 observations, the complete-data error degrees of freedom is 30.

```
proc mianalyze data=outuni edf=30;
   modeleffects Oxygen RunTime RunPulse;
   stderr SOxygen SRunTime SRunPulse;
run;
```

The "Model Information" table in Output 55.1.2 lists the input data set(s) and the number of imputations. The "Variance Information" table in Output 55.1.2 displays the between-imputation variance, within-imputation variance, and total variance for each univariate inference. It also displays the degrees of freedom for the total variance. The relative increase in variance due to missing values, the fraction of missing information, and the relative efficiency for each imputed variable are also displayed. A detailed description of these statistics is provided in the section "Combining Inferences from Imputed Data Sets" on page 3855 and the section "Multiple Imputation Efficiency" on page 3857.

Output 55.1.2 Variance Information

The MIANALYZE Procedure

Model Information

Data Set	WORK.OUTUNI
Number of Imputations	5

Output 55.1.2 *continued*

Variance Information

Parameter	Variance: Between	Within	Total	DF
Oxygen	0.041478	0.930853	0.980626	26.298
RunTime	0.002948	0.073142	0.076679	26.503
RunPulse	0.191086	3.114442	3.343744	25.463

Variance Information

Parameter	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Oxygen	0.053471	0.051977	0.989712
RunTime	0.048365	0.047147	0.990659
RunPulse	0.073626	0.070759	0.986046

The "Parameter Estimates" table in Output 55.1.3 displays the estimated mean and corresponding standard error for each variable. The table also displays a 95% confidence interval for the mean and a t statistic with the associated p-value for testing the hypothesis that the mean is equal to the value specified. You can use the THETA0= option to specify the value for the null hypothesis, which is zero by default. The table also displays the minimum and maximum parameter estimates from the imputed data sets.

Output 55.1.3 Parameter Estimates

The MIANALYZE Procedure

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF
Oxygen	47.180993	0.990266	45.1466	49.2154	26.298
RunTime	10.511906	0.276910	9.9432	11.0806	26.503
RunPulse	171.517500	1.828591	167.7549	175.2801	25.463

Parameter Estimates

Parameter	Minimum	Maximum
Oxygen	47.004201	47.499541
RunTime	10.444149	10.592244
RunPulse	171.146171	172.071730

Parameter Estimates

Parameter	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Oxygen	0	47.64	<.0001
RunTime	0	37.96	<.0001
RunPulse	0	93.80	<.0001

Note that the results in this example could also have been obtained with the MI procedure.

Example 55.2: Reading Means and Covariance Matrices from a DATA= COV Data Set

This example creates a COV-type data set that contains sample means and covariance matrices computed from imputed data sets. These estimates are then combined to generate valid statistical inferences about the population means.

The following statements use the CORR procedure to generate sample means and a covariance matrix for the variables in each imputed data set:

```
proc corr data=outmi cov nocorr noprint out=outcov(type=cov);
   var Oxygen RunTime RunPulse;
   by _Imputation_;
run;
```

The following statements display (in Output 55.2.1) output sample means and covariance matrices from PROC CORR for the first two imputed data sets:

```
proc print data=outcov(obs=12);
   title 'CORR Means and Covariance Matrices'
         ' (First Two Imputations)';
run;
```

Output 55.2.1 COV Data Set

CORR Means and Covariance Matrices (First Two Imputations)

Obs	_Imputation_	_TYPE_	_NAME_	Oxygen	RunTime	RunPulse
1	1	COV	Oxygen	28.5603	-7.2652	-11.812
2	1	COV	RunTime	-7.2652	2.5214	2.536
3	1	COV	RunPulse	-11.8121	2.5357	79.271
4	1	MEAN		47.0120	10.4441	171.216
5	1	STD		5.3442	1.5879	8.903
6	1	N		31.0000	31.0000	31.000
7	2	COV	Oxygen	27.1240	-6.6761	-10.217
8	2	COV	RunTime	-6.6761	2.2035	2.611
9	2	COV	RunPulse	-10.2170	2.6114	95.631
10	2	MEAN		47.2407	10.5040	171.244
11	2	STD		5.2081	1.4844	9.779
12	2	N		31.0000	31.0000	31.000

Note that the covariance matrices in the data set outcov are estimated covariance matrices of variables, $V(\mathbf{y})$. The estimated covariance matrix of the sample means is $V(\bar{\mathbf{y}}) = V(\mathbf{y})/n$, where n is the sample size, and is not the same as an estimated covariance matrix for variables.

The following statements combine the results for the imputed data sets, and derive both univariate and multivariate inferences about the means. The EDF= option is specified to request that the adjusted degrees of freedom be used in the analysis. For sample means based on 31 observations, the complete-data error degrees of freedom is 30.

```
proc mianalyze data=outcov edf=30;
   modeleffects Oxygen RunTime RunPulse;
run;
```

The "Variance Information" and "Parameter Estimates" tables display the same results as in Output 55.1.2 and Output 55.1.3, respectively, in Example 55.1.

With the WCOV, BCOV, and TCOV options, as in the following statements, the procedure displays the between-imputation covariance matrix, within-imputation covariance matrix, and total covariance matrix assuming that the between-imputation covariance matrix is proportional to the within-imputation covariance matrix in Output 55.2.2.

```
proc mianalyze data=outcov edf=30 wcov bcov tcov mult;
   modeleffects Oxygen RunTime RunPulse;
run;
```

Output 55.2.2 Covariance Matrices

The MIANALYZE Procedure

Within-Imputation Covariance Matrix

	Oxygen	RunTime	RunPulse
Oxygen	0.930852655	-0.226506411	-0.461022083
RunTime	-0.226506411	0.073141598	0.080316017
RunPulse	-0.461022083	0.080316017	3.114441784

Between-Imputation Covariance Matrix

	Oxygen	RunTime	RunPulse
Oxygen	0.0414778123	0.0099248946	0.0183701754
RunTime	0.0099248946	0.0029478891	0.0091684769
RunPulse	0.0183701754	0.0091684769	0.1910855259

Total Covariance Matrix

	Oxygen	RunTime	RunPulse
Oxygen	1.202882661	-0.292700068	-0.595750001
RunTime	-0.292700068	0.094516313	0.103787365
RunPulse	-0.595750001	0.103787365	4.024598310

With the MULT option, the procedure assumes that the between-imputation covariance matrix is proportional to the within-imputation covariance matrix and displays a multivariate inference for all the parameters taken jointly.

Output 55.2.3 Multivariate Inference

Multivariate Inference
Assuming Proportionality of Between/Within Covariance Matrices

Avg Relative Increase in Variance	Num DF	Den DF	F for H0: Parameter=Theta0	Pr > F
0.292237	3	122.68	12519.7	<.0001

The "Multivariate Inference" table in Output 55.2.3 shows a significant p-value for the null hypothesis that the population means are all equal to zero.

Example 55.3: Reading Regression Results from a DATA= EST Data Set

This example creates an EST-type data set that contains regression coefficients and their corresponding covariance matrices computed from imputed data sets. These estimates are then combined to generate valid statistical inferences about the regression model.

The following statements use the REG procedure to generate regression coefficients:

```
proc reg data=outmi outest=outreg covout noprint;
   model Oxygen= RunTime RunPulse;
   by _Imputation_;
run;
```

The following statements display (in Output 55.3.1) output regression coefficients and their covariance matrices from PROC REG for the first two imputed data sets:

```
proc print data=outreg(obs=8);
   var _Imputation_ _Type_ _Name_
      Intercept RunTime RunPulse;
   title 'REG Model Coefficients and Covariance Matrices'
         ' (First Two Imputations)';
run;
```

Output 55.3.1 EST-Type Data Set

REG Model Coefficients and Covariance Matrices (First Two Imputations)

Obs	_Imputation_	_TYPE_	_NAME_	Intercept	RunTime	RunPulse
1	1	PARMS		86.544	-2.82231	-0.05873
2	1	COV	Intercept	100.145	-0.53519	-0.55077
3	1	COV	RunTime	-0.535	0.10774	-0.00345
4	1	COV	RunPulse	-0.551	-0.00345	0.00343
5	2	PARMS		83.021	-3.00023	-0.02491
6	2	COV	Intercept	79.032	-0.66765	-0.41918
7	2	COV	RunTime	-0.668	0.11456	-0.00313
8	2	COV	RunPulse	-0.419	-0.00313	0.00264

The following statements combine the results for the imputed data sets. The EDF= option is specified to request that the adjusted degrees of freedom be used in the analysis. For a regression model with three independent variables (including the Intercept) and 31 observations, the complete-data error degrees of freedom is 28.

```
proc mianalyze data=outreg edf=28;
   modeleffects Intercept RunTime RunPulse;
run;
```

Output 55.3.2 Variance Information

The MIANALYZE Procedure

Variance Information

Parameter	Variance: Between	Within	Total	DF
Intercept	45.529229	76.543614	131.178689	9.1917
RunTime	0.019390	0.106220	0.129487	18.311
RunPulse	0.001007	0.002537	0.003746	12.137

Variance Information

Parameter	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Intercept	0.713777	0.461277	0.915537
RunTime	0.219051	0.192620	0.962905
RunPulse	0.476384	0.355376	0.933641

The "Variance Information" table in Output 55.3.2 displays the between-imputation, within-imputation, and total variances for combining complete-data inferences.

The "Parameter Estimates" table in Output 55.3.3 displays the estimated mean and standard error of the regression coefficients. The inferences are based on the t distribution. The table also displays a 95% mean confidence interval and a t test with the associated p-value for the hypothesis that the regression coefficient is equal to zero. Since the p-value for RunPulse is 0.1597, this variable can be removed from the regression model.

Output 55.3.3 Parameter Estimates

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF
Intercept	90.837440	11.453327	65.01034	116.6645	9.1917
RunTime	-3.032870	0.359844	-3.78795	-2.2778	18.311
RunPulse	-0.068578	0.061204	-0.20176	0.0646	12.137

Parameter Estimates

Parameter	Minimum	Maximum
Intercept	83.020730	100.839807
RunTime	-3.204426	-2.822311
RunPulse	-0.112840	-0.024910

Parameter Estimates

Parameter	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Intercept	0	7.93	<.0001
RunTime	0	-8.43	<.0001
RunPulse	0	-1.12	0.2842

Example 55.4: Reading Mixed Model Results from PARMS= and COVB= Data Sets

This example creates data sets containing parameter estimates and covariance matrices computed by a mixed model analysis for a set of imputed data sets. These estimates are then combined to generate valid statistical inferences about the parameters.

The following PROC MIXED statements generate the fixed-effect parameter estimates and covariance matrix for each imputed data set:

```
proc mixed data=outmi;
   model Oxygen= RunTime RunPulse RunTime*RunPulse/solution covb;
   by _Imputation_;
   ods output SolutionF=mixparms CovB=mixcovb;
run;
```

The following statements display (in Output 55.4.1) output parameter estimates from PROC MIXED for the first two imputed data sets:

```
proc print data=mixparms (obs=8);
   var _Imputation_ Effect Estimate StdErr;
   title 'MIXED Model Coefficients (First Two Imputations)';
run;
```

Output 55.4.1 PROC MIXED Model Coefficients

MIXED Model Coefficients (First Two Imputations)

Obs	_Imputation_	Effect	Estimate	StdErr
1	1	Intercept	148.09	81.5231
2	1	RunTime	-8.8115	7.8794
3	1	RunPulse	-0.4123	0.4684
4	1	RunTime*RunPulse	0.03437	0.04517
5	2	Intercept	64.3607	64.6034
6	2	RunTime	-1.1270	6.4307
7	2	RunPulse	0.08160	0.3688
8	2	RunTime*RunPulse	-0.01069	0.03664

The following statements display (in Output 55.4.2) the output covariance matrices associated with the parameter estimates from PROC MIXED for the first two imputed data sets:

```
proc print data=mixcovb (obs=8);
  var _Imputation_ Row Effect Col1 Col2 Col3 Col4;
  title 'Covariance Matrices (First Two Imputations)';
run;
```

Output 55.4.2 PROC MIXED Covariance Matrices

Covariance Matrices (First Two Imputations)

Obs	_Imputation_	Row	Effect	Col1	Col2	Col3	Col4
1	1	1	Intercept	6646.01	-637.40	-38.1515	3.6542
2	1	2	RunTime	-637.40	62.0842	3.6548	-0.3556
3	1	3	RunPulse	-38.1515	3.6548	0.2194	-0.02099
4	1	4	RunTime*RunPulse	3.6542	-0.3556	-0.02099	0.002040
5	2	1	Intercept	4173.59	-411.46	-23.7889	2.3441
6	2	2	RunTime	-411.46	41.3545	2.3414	-0.2353
7	2	3	RunPulse	-23.7889	2.3414	0.1360	-0.01338
8	2	4	RunTime*RunPulse	2.3441	-0.2353	-0.01338	0.001343

Note that the variables Col1, Col2, Col3, and Col4 are used to identify the effects Intercept, RunTime, RunPulse, and RunTime*RunPulse, respectively, through the variable Row.

For univariate inference, only parameter estimates and their associated standard errors are needed. The following statements use the MIANALYZE procedure with the input PARMS= data set to produce univariate results:

```
proc mianalyze parms=mixparms edf=28;
   modeleffects Intercept RunTime RunPulse RunTime*RunPulse;
run;
```

The "Variance Information" table in Output 55.4.3 displays the between-imputation, within-imputation, and total variances for combining complete-data inferences.

Output 55.4.3 Variance Information

The MIANALYZE Procedure

Variance Information

Parameter	Variance: Between	Within	Total	DF
Intercept	1972.654530	4771.948777	7139.134213	11.82
RunTime	14.712602	45.549686	63.204808	13.797
RunPulse	0.062941	0.156717	0.232247	12.046
RunTime*RunPulse	0.000470	0.001490	0.002055	13.983

Variance Information

Parameter	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Intercept	0.496063	0.365524	0.931875
RunTime	0.387601	0.305893	0.942348
RunPulse	0.481948	0.358274	0.933136
RunTime*RunPulse	0.378863	0.300674	0.943276

The "Parameter Estimates" table in Output 55.4.4 displays the estimated mean and standard error of the regression coefficients.

Output 55.4.4 Parameter Estimates

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF
Intercept	136.071356	84.493397	-48.3352	320.4779	11.82
RunTime	-7.457186	7.950145	-24.5322	9.6178	13.797
RunPulse	-0.328104	0.481920	-1.3777	0.7215	12.046
RunTime*RunPulse	0.025364	0.045328	-0.0719	0.1226	13.983

Parameter Estimates

Parameter	Minimum	Maximum
Intercept	64.360719	186.549814
RunTime	-11.514341	-1.127010
RunPulse	-0.602162	0.081597
RunTime*RunPulse	-0.010690	0.047429

Parameter Estimates

Parameter	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Intercept	0	1.61	0.1337
RunTime	0	-0.94	0.3644
RunPulse	0	-0.68	0.5089
RunTime*RunPulse	0	0.56	0.5846

Since each covariance matrix contains variables Row, Col1, Col2, Col3, and Col4 for parameters, the EFFECTVAR=ROWCOL option is needed when you specify the COVB= option. The following statements illustrate the use of the MIANALYZE procedure with input PARMS= and COVB(EFFECTVAR=ROWCOL)= data sets:

```
proc mianalyze parms=mixparms edf=28
               covb(effectvar=rowcol)=mixcovb;
   modeleffects Intercept RunTime RunPulse RunTime*RunPulse;
run;
```

Example 55.5: Reading Generalized Linear Model Results

This example creates data sets containing parameter estimates and corresponding covariance matrices computed by a generalized linear model analysis for a set of imputed data sets. These estimates are then combined to generate valid statistical inferences about the model parameters.

The following statements use PROC GENMOD to generate the parameter estimates and covariance matrix for each imputed data set:

```
proc genmod data=outmi;
   model Oxygen= RunTime RunPulse/covb;
   by _Imputation_;
   ods output ParameterEstimates=gmparms
              ParmInfo=gmpinfo
              CovB=gmcovb;
run;
```

The following statements print (in Output 55.5.1) the output parameter estimates and covariance matrix from PROC GENMOD for the first two imputed data sets:

```
proc print data=gmparms (obs=8);
   var _Imputation_ Parameter Estimate StdErr;
   title 'GENMOD Model Coefficients (First Two Imputations)';
run;
```

Output 55.5.1 PROC GENMOD Model Coefficients

GENMOD Model Coefficients (First Two Imputations)

Obs	_Imputation_	Parameter	Estimate	StdErr
1	1	Intercept	86.5440	9.5107
2	1	RunTime	−2.8223	0.3120
3	1	RunPulse	−0.0587	0.0556
4	1	Scale	2.6692	0.3390
5	2	Intercept	83.0207	8.4489
6	2	RunTime	−3.0002	0.3217
7	2	RunPulse	−0.0249	0.0488
8	2	Scale	2.5727	0.3267

The following statements display the parameter information table in Output 55.5.2. The table identifies parameter names used in the covariance matrices. The parameters Prm1, Prm2, and Prm3 are used for the effects Intercept, RunTime, and RunPulse, respectively, in each covariance matrix.

```
proc print data=gmpinfo (obs=6);
   title 'GENMOD Parameter Information (First Two Imputations)';
run;
```

Output 55.5.2 PROC GENMOD Model Information

GENMOD Parameter Information (First Two Imputations)

Obs	_Imputation_	Parameter	Effect
1	1	Prm1	Intercept
2	1	Prm2	RunTime
3	1	Prm3	RunPulse
4	2	Prm1	Intercept
5	2	Prm2	RunTime
6	2	Prm3	RunPulse

The following statements display (in Output 55.5.3) the output covariance matrices from PROC GENMOD for the first two imputed data sets. Note that the GENMOD procedure computes maximum likelihood estimates for each covariance matrix.

```
proc print data=gmcovb (obs=8);
   var _Imputation_ RowName Prm1 Prm2 Prm3;
   title 'GENMOD Covariance Matrices (First Two Imputations)';
run;
```

Output 55.5.3 PROC GENMOD Covariance Matrices

GENMOD Covariance Matrices (First Two Imputations)

Obs	_Imputation_	Row Name	Prm1	Prm2	Prm3
1	1	Prm1	90.453923	-0.483394	-0.497473
2	1	Prm2	-0.483394	0.0973159	-0.003113
3	1	Prm3	-0.497473	-0.003113	0.0030954
4	1	Scale	1.344E-15	-1.09E-17	-6.12E-18
5	2	Prm1	71.383332	-0.603037	-0.378616
6	2	Prm2	-0.603037	0.1034766	-0.002826
7	2	Prm3	-0.378616	-0.002826	0.0023843
8	2	Scale	1.602E-14	1.755E-16	-1.02E-16

The following statements use the MIANALYZE procedure with input PARMS=, PARMINFO=, and COVB= data sets:

```
proc mianalyze parms=gmparms covb=gmcovb parminfo=gmpinfo;
   modeleffects Intercept RunTime RunPulse;
run;
```

Since the GENMOD procedure computes maximum likelihood estimates for the covariance matrix, the EDF= option is not used. The resulting model coefficients are identical to the estimates in Output 55.3.3 in Example 55.3. However, the standard errors are slightly different because in this example, maximum likelihood estimates for the standard errors are combined without the EDF= option, whereas in Example 55.3, unbiased estimates for the standard errors are combined with the EDF= option.

Example 55.6: Reading GLM Results from PARMS= and XPXI= Data Sets

This example creates data sets containing parameter estimates and corresponding $(X'X)^{-1}$ matrices computed by a general linear model analysis for a set of imputed data sets. These estimates are then combined to generate valid statistical inferences about the model parameters.

The following statements use PROC GLM to generate the parameter estimates and $(X'X)^{-1}$ matrix for each imputed data set:

```
proc glm data=outmi;
   model Oxygen= RunTime RunPulse/inverse;
   by _Imputation_;
   ods output ParameterEstimates=glmparms
              InvXPX=glmxpxi;
quit;
```

The following statements display (in Output 55.6.1) the output parameter estimates and standard errors from PROC GLM for the first two imputed data sets:

```
proc print data=glmparms (obs=6);
   var _Imputation_ Parameter Estimate StdErr;
   title 'GLM Model Coefficients (First Two Imputations)';
run;
```

Output 55.6.1 PROC GLM Model Coefficients

GLM Model Coefficients (First Two Imputations)

Obs	_Imputation_	Parameter	Estimate	StdErr
1	1	Intercept	86.5440339	10.00726811
2	1	RunTime	-2.8223108	0.32824165
3	1	RunPulse	-0.0587292	0.05854109
4	2	Intercept	83.0207303	8.88996885
5	2	RunTime	-3.0002288	0.33847204
6	2	RunPulse	-0.0249103	0.05137859

The following statements display (in Output 55.6.2) $(X'X)^{-1}$ matrices from PROC GLM for the first two imputed data sets:

```
proc print data=glmxpxi (obs=8);
   var _Imputation_ Parameter Intercept RunTime RunPulse;
   title 'GLM X''X Inverse Matrices (First Two Imputations)';
run;
```

Output 55.6.2 PROC GLM $(X'X)^{-1}$ Matrices

GLM X'X Inverse Matrices (First Two Imputations)

Obs	_Imputation_	Parameter	Intercept	RunTime	RunPulse
1	1	Intercept	12.696250656	-0.067849956	-0.069826009
2	1	RunTime	-0.067849956	0.0136594055	-0.000436938
3	1	RunPulse	-0.069826009	-0.000436938	0.0004344762
4	1	Oxygen	86.544033929	-2.822310769	-0.058729234
5	2	Intercept	10.784620785	-0.091107072	-0.057201387
6	2	RunTime	-0.091107072	0.0156332765	-0.000426902
7	2	RunPulse	-0.057201387	-0.000426902	0.0003602208
8	2	Oxygen	83.020730343	-3.000228818	-0.024910305

The standard errors for the estimates in the output glmparms data set are needed to create the covariance matrix from the $(X'X)^{-1}$ matrix. The following statements use the MIANALYZE procedure

with input PARMS= and XPXI= data sets to produce the same results as displayed in Output 55.3.2 and Output 55.3.3 in Example 55.3:

```
proc mianalyze parms=glmparms xpxi=glmxpxi edf=28;
   modeleffects Intercept RunTime RunPulse;
run;
```

Example 55.7: Reading Logistic Model Results from PARMS= and COVB= Data Sets

This example creates data sets containing parameter estimates and corresponding covariance matrices computed by a logistic regression analysis for a set of imputed data sets. These estimates are then combined to generate valid statistical inferences about the model parameters.

The following statements use PROC LOGISTIC to generate the parameter estimates and covariance matrix for each imputed data set:

```
proc logistic data=outfish;
   class Species;
   model Species= Height Width Height*Width/ covb;
   by _Imputation_;
   ods output ParameterEstimates=lgsparms
              CovB=lgscovb;
run;
```

The following statements display (in Output 55.7.1) the output logistic regression coefficients from PROC LOGISTIC for the first two imputed data sets:

```
proc print data=lgsparms (obs=8);
   title 'LOGISTIC Model Coefficients (First Two Imputations)';
run;
```

Output 55.7.1 PROC LOGISTIC Model Coefficients

LOGISTIC Model Coefficients (First Two Imputations)

Obs	_Imputation_	Variable	DF	Estimate	StdErr	WaldChiSq	Prob ChiSq
1	1	Intercept	1	-22.7507	45.7309	0.2475	0.6188
2	1	Height	1	4.3267	3.7473	1.3332	0.2482
3	1	Width	1	-1.7027	7.8049	0.0476	0.8273
4	1	Height*Width	1	-0.3517	0.5542	0.4028	0.5256
5	2	Intercept	1	-27.2116	26.3744	1.0645	0.3022
6	2	Height	1	4.1833	2.0347	4.2271	0.0398
7	2	Width	1	1.1045	6.1278	0.0325	0.8570
8	2	Height*Width	1	-0.4495	0.3251	1.9116	0.1668

The following statements displays the covariance matrices associated with parameter estimates derived from the first two imputations in Output 55.7.2:

```
proc print data=lgscovb (obs=8);
   title 'LOGISTIC Model Covariance Matrices (First Two Imputations)';
run;
```

Output 55.7.2 PROC LOGISTIC Covariance Matrices

LOGISTIC Model Covariance Matrices (First Two Imputations)

Obs	_Imputation_	Parameter	Intercept	Height	Width	Height Width
1	1	Intercept	2091.316	-158.686	-330.408	24.9984
2	1	Height	-158.686	14.04241	21.1609	-1.98425
3	1	Width	-330.408	21.1609	60.91617	-3.84737
4	1	HeightWidth	24.9984	-1.98425	-3.84737	0.307092
5	2	Intercept	695.6072	-36.8743	-148.034	7.997264
6	2	Height	-36.8743	4.139861	4.497508	-0.57405
7	2	Width	-148.034	4.497508	37.55025	-1.52686
8	2	HeightWidth	7.997264	-0.57405	-1.52686	0.105678

The following statements use the MIANALYZE procedure with input PARMS= and COVB= data sets:

```
proc mianalyze parms=lgsparms
               covb(effectvar=stacking)=lgscovb;
   modeleffects Intercept Height Width Height*Width;
run;
```

The "Variance Information" table in Output 55.7.3 displays the between-imputation, within-imputation, and total variances for combining complete-data inferences.

Output 55.7.3 Variance Information

The MIANALYZE Procedure

Variance Information

Parameter	Variance: Between	Variance: Within	Variance: Total	DF
Intercept	5.034765	1668.718613	1674.760331	307358
Height	0.024662	11.102796	11.132390	565996
Width	2.258894	53.686948	56.397621	1731.5
Height*Width	0.002655	0.245967	0.249153	24467

Variance Information

Parameter	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Intercept	0.003621	0.003614	0.999278
Height	0.002666	0.002662	0.999468
Width	0.050490	0.049161	0.990263
Height*Width	0.012952	0.012867	0.997433

The "Parameter Estimates" table in Output 55.7.4 displays the combined parameter estimates with associated standard errors.

Output 55.7.4 Parameter Estimates

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF
Intercept	-25.952920	40.923836	-106.162	54.25664	307358
Height	4.367841	3.336524	-2.172	10.90732	565996
Width	-0.321990	7.509835	-15.051	14.40731	1731.5
Height*Width	-0.408057	0.499152	-1.386	0.57031	24467

Parameter Estimates

Parameter	Minimum	Maximum
Intercept	-28.627325	-22.750660
Height	4.183264	4.570938
Width	-1.702651	1.500299
Height*Width	-0.473804	-0.351718

Parameter Estimates

Parameter	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Intercept	0	-0.63	0.5260
Height	0	1.31	0.1905
Width	0	-0.04	0.9658
Height*Width	0	-0.82	0.4137

Example 55.8: Reading Mixed Model Results with Classification Variables

This example creates data sets containing parameter estimates and corresponding covariance matrices with classification variables computed by a mixed regression model analysis for a set of imputed data sets. These estimates are then combined to generate valid statistical inferences about the model parameters.

The following statements use PROC MIXED to generate the parameter estimates and covariance matrix for each imputed data set:

```
proc mixed data=outfish;
   class Species;
   model Length3= Species Height Width/ solution covb;
   by _Imputation_;
   ods output SolutionF=mxparms CovB=mxcovb;
run;
```

The following statements display (in Output 55.8.1) the output mixed model coefficients from PROC MIXED for the first two imputed data sets:

```
proc print data=mxparms (obs=10);
   var _Imputation_ Effect Species Estimate StdErr;
   title 'MIXED Model Coefficients (First Two Imputations)';
run;
```

Output 55.8.1 PROC MIXED Model Coefficients

MIXED Model Coefficients (First Two Imputations)

Obs	_Imputation_	Effect	Species	Estimate	StdErr
1	1	Intercept		6.8207	1.0248
2	1	Species	Gp1	-0.00467	0.8550
3	1	Species	Gp2	0	.
4	1	Height		0.9085	0.1964
5	1	Width		3.2744	0.5643
6	2	Intercept		6.9259	0.9823
7	2	Species	Gp1	-0.3542	0.8605
8	2	Species	Gp2	0	.
9	2	Height		0.9630	0.1910
10	2	Width		3.1548	0.5370

The following statements use the MIANALYZE procedure with an input PARMS= data set:

```
proc mianalyze parms(classvar=full)=mxparms;
   class Species;
   modeleffects Intercept Species Height Width;
run;
```

The "Variance Information" table in Output 55.8.2 displays the between-imputation, within-imputation, and total variances for combining complete-data inferences.

Output 55.8.2 Variance Information

The MIANALYZE Procedure

Variance Information

Parameter	Species	Variance: Between	Variance: Within	Variance: Total	DF
Intercept		0.013439	1.008137	1.024263	16136
Species	Gp1	0.131130	0.719737	0.877093	124.28
Species	Gp2	0	.	.	.
Height		0.005162	0.037108	0.043302	195.48
Width		0.025787	0.302173	0.333117	463.55

Variance Information

Parameter	Species	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
Intercept		0.015996	0.015866	0.996837
Species	Gp1	0.218629	0.192300	0.962964
Species	Gp2	.	.	.
Height		0.166925	0.151682	0.970557
Width		0.102406	0.096781	0.981011

The "Parameter Estimates" table in Output 55.8.3 displays the combined parameter estimates with associated standard errors.

Output 55.8.3 Parameter Estimates

Parameter Estimates

Parameter	Species	Estimate	Std Error	95% Confidence Limits		DF
Intercept		6.825324	1.012059	4.84158	8.809072	16136
Species	Gp1	−0.174302	0.936532	−2.02792	1.679317	124.28
Species	Gp2	0	.	.	.	.
Height		0.927390	0.208091	0.51700	1.337782	195.48
Width		3.242346	0.577163	2.10817	4.376527	463.55

Parameter Estimates

Parameter	Species	Minimum	Maximum
Intercept		6.692058	6.956116
Species	Gp1	−0.726082	0.129023
Species	Gp2	0	0
Height		0.860775	1.034011
Width		3.011865	3.400082

Parameter Estimates

Parameter	Species	Theta0	t for H0: Parameter=Theta0	Pr > |t|
Intercept		0	6.74	<.0001
Species	Gp1	0	−0.19	0.8527
Species	Gp2	0	.	.
Height		0	4.46	<.0001
Width		0	5.62	<.0001

Example 55.9: Using a TEST statement

This example creates an EST-type data set that contains regression coefficients and their corresponding covariance matrices computed from imputed data sets. These estimates are then combined to generate valid statistical inferences about the regression model. A TEST statement is used to test linear hypotheses about the parameters.

The following statements use the REG procedure to generate regression coefficients:

```
proc reg data=outmi outest=outreg covout noprint;
   model Oxygen= RunTime RunPulse;
   by _Imputation_;
run;
```

The following statements combine the results for the imputed data sets. A TEST statement is used to test linear hypotheses of Intercept=0 and RunTime=RunPulse.

```
proc mianalyze data=outreg edf=28;
   modeleffects Intercept RunTime RunPulse;
   test Intercept, RunTime=RunPulse / mult;
run;
```

The "Test Specification" table in Output 55.9.1 displays the **L** matrix and the **c** vector in a TEST statement. Since there is no label specified for the TEST statement, "Test 1" is used as the label.

Output 55.9.1 Test Specification

The MIANALYZE Procedure
Test: Test 1

Test Specification

	L Matrix			
Parameter	Intercept	RunTime	RunPulse	C
TestPrm1	1.000000	0	0	0
TestPrm2	0	1.000000	-1.000000	0

The "Variance Information" table in Output 55.9.2 displays the between-imputation variance, within-imputation variance, and total variance for each univariate inference. A detailed description of these statistics is provided in the section "Combining Inferences from Imputed Data Sets" on page 3855 and the section "Multiple Imputation Efficiency" on page 3857.

Output 55.9.2 Variance Information

Variance Information

	Variance			
Parameter	Between	Within	Total	DF
TestPrm1	45.529229	76.543614	131.178689	9.1917
TestPrm2	0.014715	0.114324	0.131983	20.598

Variance Information

Parameter	Relative Increase in Variance	Fraction Missing Information	Relative Efficiency
TestPrm1	0.713777	0.461277	0.915537
TestPrm2	0.154459	0.141444	0.972490

The "Parameter Estimates" table in Output 55.9.3 displays the estimated mean and standard error of the linear components. The inferences are based on the t distribution. The table also displays a 95% mean confidence interval and a t test with the associated p-value for the hypothesis that each linear component of $\mathbf{L}\boldsymbol{\beta}$ is equal to zero.

Output 55.9.3 Parameter Estimates

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF
TestPrm1	90.837440	11.453327	65.01034	116.6645	9.1917
TestPrm2	-2.964292	0.363294	-3.72070	-2.2079	20.598

Parameter Estimates

Parameter	Minimum	Maximum	C	t for H0: Parameter=C	Pr > |t|
TestPrm1	83.020730	100.839807	0	7.93	<.0001
TestPrm2	-3.091586	-2.763582	0	-8.16	<.0001

With the MULT option, the procedure assumes that the between-imputation covariance matrix is proportional to the within-imputation covariance matrix and displays a multivariate inference for all the linear components taken jointly in Output 55.9.4.

Output 55.9.4 Multivariate Inference

Multivariate Inference
Assuming Proportionality of Between/Within Covariance Matrices

Avg Relative Increase in Variance	Num DF	Den DF	F for H0: Parameter=Theta0	Pr > F
0.419868	2	35.053	60.34	<.0001

Example 55.10: Combining Correlation Coefficients

This example combines sample correlation coefficients computed from a set of imputed data sets by using Fisher's z transformation.

Fisher's z transformation of the sample correlation r is

$$z = \frac{1}{2}\log\left(\frac{1+r}{1-r}\right)$$

The statistic z is approximately normally distributed with mean

$$\log\left(\frac{1+\rho}{1-\rho}\right)$$

and variance $1/(n-3)$, where ρ is the population correlation coefficient and n is the number of observations.

The following statements use the CORR procedure to compute the correlation r and its associated Fisher's z statistic between variables Oxygen and RunTime for each imputed data set. The ODS statement is used to save Fisher's z statistic in an output data set.

```
proc corr data=outmi fisher(biasadj=no);
   var Oxygen RunTime;
   by _Imputation_;
   ods output FisherPearsonCorr= outz;
run;
```

The following statements display the number of observations and Fisher's z statistic for each imputed data set in Output 55.10.1:

```
proc print data=outz;
   title 'Fisher''s Correlation Statistics';
   var _Imputation_ NObs ZVal;
run;
```

Output 55.10.1 Output z Statistics

Fisher's Correlation Statistics

Obs	_Imputation_	NObs	ZVal
1	1	31	-1.27869
2	2	31	-1.30715
3	3	31	-1.27922
4	4	31	-1.39243
5	5	31	-1.40146

The following statements generate the standard error associated with the z statistic, $1/\sqrt{n-3}$:

```
data outz;
   set outz;
   StdZ= 1. / sqrt(NObs-3);
run;
```

The following statements use the MIANALYZE procedure to generate a combined parameter estimate $\hat{z}$ and its variance, as shown in Output 55.10.2. The ODS statement is used to save the parameter estimates in an output data set.

```
proc mianalyze data=outz;
   ods output ParameterEstimates=parms;
   modeleffects ZVal;
   stderr StdZ;
run;
```

Output 55.10.2 Combining Fisher's z Statistics

The MIANALYZE Procedure

Parameter Estimates

Parameter	Estimate	Std Error	95% Confidence Limits		DF
ZVal	-1.331787	0.200327	-1.72587	-0.93771	330.23

Parameter Estimates

Parameter	Minimum	Maximum
ZVal	-1.401459	-1.278686

Parameter Estimates

Parameter	Theta0	t for H0: Parameter=Theta0	Pr > |t|
ZVal	0	-6.65	<.0001

In addition to the estimate for z, PROC MIANALYZE also generates 95% confidence limits for z, $\hat{z}_{.025}$ and $\hat{z}_{.975}$. The following statements print the estimate and 95% confidence limits for z in Output 55.10.3:

```
proc print data=parms;
   title 'Parameter Estimates with 95% Confidence Limits';
   var Estimate LCLMean UCLMean;
run;
```

Output 55.10.3 Parameter Estimates with 95% Confidence Limits

Parameter Estimates with 95% Confidence Limits

Obs	Estimate	LCLMean	UCLMean
1	-1.331787	-1.72587	-0.93771

An estimate of the correlation coefficient with its corresponding 95% confidence limits is then generated from the following inverse transformation as described in the section "Correlation Coefficients" on page 3860:

$$r = tanh(z) = \frac{e^{2z} - 1}{e^{2z} + 1}$$

for $z = \hat{z}$, $\hat{z}_{.025}$, and $\hat{z}_{.975}$.

The following statements generate and display an estimate of the correlation coefficient and its 95% confidence limits, as shown in Output 55.10.4:

```
data corr_ci;
   set parms;
   r=       tanh( Estimate);
   r_lower= tanh( LCLMean);
   r_upper= tanh( UCLMean);
run;
proc print data=corr_ci;
   title 'Estimated Correlation Coefficient'
         ' with 95% Confidence Limits';
   var r r_lower r_upper;
run;
```

Output 55.10.4 Estimated Correlation Coefficient

Estimated Correlation Coefficient with 95% Confidence Limits

Obs	r	r_lower	r_upper
1	-0.86969	-0.93857	-0.73417

References

Allison, P. D. (2000), "Multiple Imputation for Missing Data: A Cautionary Tale," *Sociological Methods and Research*, 28, 301–309.

Allison, P. D. (2001), "Missing Data," Thousand Oaks, CA: Sage Publications.

Barnard, J. and Rubin, D. B. (1999), "Small-Sample Degrees of Freedom with Multiple Imputation," *Biometrika*, 86, 948–955.

Cochran, W. J. (1977), *Sampling Techniques,* Second Edition, New York: John Wiley & Sons.

Gadbury, G. L., Coffey, C. S., and Allison, D. B. (2003), "Modern Statistical Methods for Handling Missing Repeated Measurements in Obesity Trial Data: Beyond LOCF," *Obesity Reviews*, 4, 175–184.

Horton, N. J. and Lipsitz, S. R. (2001), "Multiple Imputation in Practice: Comparison of Software Packages for Regression Models with Missing Variables," *Journal of the American Statistical Association*, 55, 244–254.

Li, K. H., Raghunathan, T. E., and Rubin, D. B. (1991), "Large-Sample Significance Levels from Multiply Imputed Data Using Moment-Based Statistics and an F Reference Distribution," *Journal of the American Statistical Association*, 86, 1065–1073.

Little, R. J. A. and Rubin, D. B. (2002), *Statistical Analysis with Missing Data*, Second Edition, New York: John Wiley & Sons.

Rubin, D. B. (1976), "Inference and Missing Data," *Biometrika*, 63, 581–592.

Rubin, D. B. (1987), *Multiple Imputation for Nonresponse in Surveys*, New York: John Wiley & Sons.

Rubin, D. B. (1996), "Multiple Imputation after 18+ Years," *Journal of the American Statistical Association*, 91, 473–489.

Schafer, J. L. (1997), *Analysis of Incomplete Multivariate Data*, New York: Chapman and Hall.

Chapter 56

The MIXED Procedure

Contents

Overview: MIXED Procedure **3892**
- Basic Features 3893
- Notation for the Mixed Model 3894
- PROC MIXED Contrasted with Other SAS Procedures 3895

Getting Started: MIXED Procedure **3896**
- Clustered Data Example 3896

Syntax: MIXED Procedure **3902**
- PROC MIXED Statement 3904
- BY Statement 3916
- CLASS Statement 3916
- CONTRAST Statement 3917
- ESTIMATE Statement 3920
- ID Statement 3922
- LSMEANS Statement 3922
- MODEL Statement 3928
- PARMS Statement 3943
- PRIOR Statement 3945
- RANDOM Statement 3949
- REPEATED Statement 3954
- WEIGHT Statement 3968

Details: MIXED Procedure **3968**
- Mixed Models Theory 3968
- Parameterization of Mixed Models 3981
- Residuals and Influence Diagnostics 3986
- Default Output 3995
- ODS Table Names 3999
- ODS Graphics 4004
- Computational Issues 4010

Examples: Mixed Procedure **4014**
- Example 56.1: Split-Plot Design 4014
- Example 56.2: Repeated Measures 4019
- Example 56.3: Plotting the Likelihood 4032
- Example 56.4: Known G and R 4039
- Example 56.5: Random Coefficients 4047

Example 56.6: Line-Source Sprinkler Irrigation 4055
Example 56.7: Influence in Heterogeneous Variance Model 4061
Example 56.8: Influence Analysis for Repeated Measures Data 4070
Example 56.9: Examining Individual Test Components 4079
References **4084**

Overview: MIXED Procedure

The MIXED procedure fits a variety of mixed linear models to data and enables you to use these fitted models to make statistical inferences about the data. A *mixed linear model* is a generalization of the standard linear model used in the GLM procedure, the generalization being that the data are permitted to exhibit correlation and nonconstant variability. The mixed linear model, therefore, provides you with the flexibility of modeling not only the means of your data (as in the standard linear model) but their variances and covariances as well.

The primary assumptions underlying the analyses performed by PROC MIXED are as follows:

- The data are normally distributed (Gaussian).
- The means (expected values) of the data are linear in terms of a certain set of parameters.
- The variances and covariances of the data are in terms of a different set of parameters, and they exhibit a structure matching one of those available in PROC MIXED.

Since Gaussian data can be modeled entirely in terms of their means and variances/covariances, the two sets of parameters in a mixed linear model actually specify the complete probability distribution of the data. The parameters of the mean model are referred to as *fixed-effects parameters*, and the parameters of the variance-covariance model are referred to as *covariance parameters*.

The fixed-effects parameters are associated with known explanatory variables, as in the standard linear model. These variables can be either qualitative (as in the traditional analysis of variance) or quantitative (as in standard linear regression). However, the covariance parameters are what distinguishes the mixed linear model from the standard linear model.

The need for covariance parameters arises quite frequently in applications, the following being the two most typical scenarios:

- The experimental units on which the data are measured can be grouped into clusters, and the data from a common cluster are correlated.
- Repeated measurements are taken on the same experimental unit, and these repeated measurements are correlated or exhibit variability that changes.

The first scenario can be generalized to include one set of clusters nested within another. For example, if students are the experimental unit, they can be clustered into classes, which in turn can be

clustered into schools. Each level of this hierarchy can introduce an additional source of variability and correlation. The second scenario occurs in longitudinal studies, where repeated measurements are taken over time. Alternatively, the repeated measures could be spatial or multivariate in nature.

PROC MIXED provides a variety of covariance structures to handle the previous two scenarios. The most common of these structures arises from the use of *random-effects parameters*, which are additional unknown random variables assumed to affect the variability of the data. The variances of the random-effects parameters, commonly known as *variance components*, become the covariance parameters for this particular structure. Traditional mixed linear models contain both fixed- and random-effects parameters, and, in fact, it is the combination of these two types of effects that led to the name *mixed model*. PROC MIXED fits not only these traditional variance component models but numerous other covariance structures as well.

PROC MIXED fits the structure you select to the data by using the method of *restricted maximum likelihood (REML)*, also known as *residual maximum likelihood*. It is here that the Gaussian assumption for the data is exploited. Other estimation methods are also available, including *maximum likelihood* and *MIVQUE0*. The details behind these estimation methods are discussed in subsequent sections.

After a model has been fit to your data, you can use it to draw statistical inferences via both the fixed-effects and covariance parameters. PROC MIXED computes several different statistics suitable for generating hypothesis tests and confidence intervals. The validity of these statistics depends upon the mean and variance-covariance model you select, so it is important to choose the model carefully. Some of the output from PROC MIXED helps you assess your model and compare it with others.

Basic Features

PROC MIXED provides easy accessibility to numerous mixed linear models that are useful in many common statistical analyses. In the style of the GLM procedure, PROC MIXED fits the specified mixed linear model and produces appropriate statistics.

Here are some basic features of PROC MIXED:

- covariance structures, including variance components, compound symmetry, unstructured, AR(1), Toeplitz, spatial, general linear, and factor analytic
- GLM-type grammar, by using MODEL, RANDOM, and REPEATED statements for model specification and CONTRAST, ESTIMATE, and LSMEANS statements for inferences
- appropriate standard errors for all specified estimable linear combinations of fixed and random effects, and corresponding t and F tests
- subject and group effects that enable blocking and heterogeneity, respectively
- REML and ML estimation methods implemented with a Newton-Raphson algorithm
- capacity to handle unbalanced data
- ability to create a SAS data set corresponding to any table

PROC MIXED uses the Output Delivery System (ODS), a SAS subsystem that provides capabilities for displaying and controlling the output from SAS procedures. ODS enables you to convert any of the output from PROC MIXED into a SAS data set. See the section "ODS Table Names" on page 3999.

The MIXED procedure now uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the statistical graphics available with the MIXED procedure, see the PLOTS option in the PROC MIXED statement and the section "ODS Graphics" on page 4004.

Notation for the Mixed Model

This section introduces the mathematical notation used throughout this chapter to describe the mixed linear model. You should be familiar with basic matrix algebra (see Searle 1982). A more detailed description of the mixed model is contained in the section "Mixed Models Theory" on page 3968.

A statistical model is a mathematical description of how data are generated. The standard linear model, as used by the GLM procedure, is one of the most common statistical models:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

In this expression, $\mathbf{y}$ represents a vector of observed data, $\boldsymbol{\beta}$ is an unknown vector of fixed-effects parameters with known design matrix $\mathbf{X}$, and $\boldsymbol{\epsilon}$ is an unknown random error vector modeling the statistical noise around $\mathbf{X}\boldsymbol{\beta}$. The focus of the standard linear model is to model the mean of $\mathbf{y}$ by using the fixed-effects parameters $\boldsymbol{\beta}$. The residual errors $\boldsymbol{\epsilon}$ are assumed to be independent and identically distributed Gaussian random variables with mean 0 and variance σ^2.

The mixed model generalizes the standard linear model as follows:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

Here, $\boldsymbol{\gamma}$ is an unknown vector of random-effects parameters with known design matrix $\mathbf{Z}$, and $\boldsymbol{\epsilon}$ is an unknown random error vector whose elements are no longer required to be independent and homogeneous.

To further develop this notion of variance modeling, assume that $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$ are Gaussian random variables that are uncorrelated and have expectations $\mathbf{0}$ and variances $\mathbf{G}$ and $\mathbf{R}$, respectively. The variance of $\mathbf{y}$ is thus

$$\mathbf{V} = \mathbf{Z}\mathbf{G}\mathbf{Z}' + \mathbf{R}$$

Note that, when $\mathbf{R} = \sigma^2\mathbf{I}$ and $\mathbf{Z} = \mathbf{0}$, the mixed model reduces to the standard linear model.

You can model the variance of the data, $\mathbf{y}$, by specifying the structure (or form) of $\mathbf{Z}$, $\mathbf{G}$, and $\mathbf{R}$. The model matrix $\mathbf{Z}$ is set up in the same fashion as $\mathbf{X}$, the model matrix for the fixed-effects parameters. For $\mathbf{G}$ and $\mathbf{R}$, you must select some *covariance structure*. Possible covariance structures include the following:

- variance components
- compound symmetry (common covariance plus diagonal)
- unstructured (general covariance)
- autoregressive
- spatial
- general linear
- factor analytic

By appropriately defining the model matrices **X** and **Z**, as well as the covariance structure matrices **G** and **R**, you can perform numerous mixed model analyses.

PROC MIXED Contrasted with Other SAS Procedures

PROC MIXED is a generalization of the GLM procedure in the sense that PROC GLM fits standard linear models, and PROC MIXED fits the wider class of mixed linear models. Both procedures have similar CLASS, MODEL, CONTRAST, ESTIMATE, and LSMEANS statements, but their RANDOM and REPEATED statements differ (see the following paragraphs). Both procedures use the non-full-rank model parameterization, although the sorting of classification levels can differ between the two. PROC MIXED computes only Type I–Type III tests of fixed effects, while PROC GLM computes Types I–IV.

The RANDOM statement in PROC MIXED incorporates random effects constituting the $\boldsymbol{\gamma}$ vector in the mixed model. However, in PROC GLM, effects specified in the RANDOM statement are still treated as fixed as far as the model fit is concerned, and they serve only to produce corresponding expected mean squares. These expected mean squares lead to the traditional ANOVA estimates of variance components. PROC MIXED computes REML and ML estimates of variance parameters, which are generally preferred to the ANOVA estimates (Searle 1988; Harville 1988; Searle, Casella, and McCulloch 1992). Optionally, PROC MIXED also computes MIVQUE0 estimates, which are similar to ANOVA estimates.

The REPEATED statement in PROC MIXED is used to specify covariance structures for repeated measurements on subjects, while the REPEATED statement in PROC GLM is used to specify various transformations with which to conduct the traditional univariate or multivariate tests. In repeated measures situations, the mixed model approach used in PROC MIXED is more flexible and more widely applicable than either the univariate or multivariate approach. In particular, the mixed model approach provides a larger class of covariance structures and a better mechanism for handling missing values (Wolfinger and Chang 1995).

PROC MIXED subsumes the VARCOMP procedure. PROC MIXED provides a wide variety of covariance structures, while PROC VARCOMP estimates only simple random effects. PROC MIXED carries out several analyses that are absent in PROC VARCOMP, including the estimation and testing of linear combinations of fixed and random effects.

The ARIMA and AUTOREG procedures provide more time series structures than PROC MIXED, although they do not fit variance component models. The CALIS procedure fits general covariance matrices, but the fixed effects structure of the model is formed differently than in PROC MIXED. The LATTICE and NESTED procedures fit special types of mixed linear models that can also be handled in PROC MIXED, although PROC MIXED might run slower because of its more general algorithm. The TSCSREG procedure analyzes time series cross-sectional data, and it fits some structures not available in PROC MIXED.

The GLIMMIX procedure fits generalized linear mixed models (GLMMs). Linear mixed models—where the data are normally distributed, given the random effects—are in the class of GLMMs. The MIXED procedure can estimate covariance parameters with ANOVA methods that are not available in the GLIMMIX procedure (see METHOD=TYPE1, METHOD=TYPE2, and METHOD=TYPE3 in the PROC MIXED statement). Also, PROC MIXED can perform a sampling-based Bayesian analysis through the PRIOR statement, and the procedure supports certain Kronecker-type covariance structures. These features are not available in the GLIMMIX procedure. The GLIMMIX procedure, on the other hand, accommodates nonnormal data and offers a broader array of post-processing features than the MIXED procedure.

Getting Started: MIXED Procedure

Clustered Data Example

Consider the following SAS data set as an introductory example:

```
data heights;
  input Family Gender$ Height @@;
  datalines;
1 F 67   1 F 66   1 F 64   1 M 71   1 M 72   2 F 63
2 F 63   2 F 67   2 M 69   2 M 68   2 M 70   3 F 63
3 M 64   4 F 67   4 F 66   4 M 67   4 M 67   4 M 69
;
```

The response variable Height measures the heights (in inches) of 18 individuals. The individuals are classified according to Family and Gender. You can perform a traditional two-way analysis of variance of these data with the following PROC MIXED statements:

```
proc mixed data=heights;
   class Family Gender;
   model Height = Gender Family Family*Gender;
run;
```

The PROC MIXED statement invokes the procedure. The CLASS statement instructs PROC MIXED to consider both Family and Gender as classification variables. Dummy (indicator) variables are, as a result, created corresponding to all of the distinct levels of Family and Gender. For these data, Family has four levels and Gender has two levels.

The MODEL statement first specifies the response (dependent) variable Height. The explanatory (independent) variables are then listed after the equal (=) sign. Here, the two explanatory variables are Gender and Family, and these are the main effects of the design. The third explanatory term, Family*Gender, models an interaction between the two main effects.

PROC MIXED uses the dummy variables associated with Gender, Family, and Family*Gender to construct the **X** matrix for the linear model. A column of 1s is also included as the first column of **X** to model a global intercept. There are no **Z** or **G** matrices for this model, and **R** is assumed to equal $\sigma^2\mathbf{I}$, where **I** is an 18×18 identity matrix.

The RUN statement completes the specification. The coding is precisely the same as with the GLM procedure. However, much of the output from PROC MIXED is different from that produced by PROC GLM.

The output from PROC MIXED is shown in Figure 56.1–Figure 56.7.

The "Model Information" table in Figure 56.1 describes the model, some of the variables that it involves, and the method used in fitting it. This table also lists the method (profile, factor, parameter, or none) for handling the residual variance.

Figure 56.1 Model Information

The Mixed Procedure

Model Information

Data Set	WORK.HEIGHTS
Dependent Variable	Height
Covariance Structure	Diagonal
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Residual

The "Class Level Information" table in Figure 56.2 lists the levels of all variables specified in the CLASS statement. You can check this table to make sure that the data are correct.

Figure 56.2 Class Level Information

Class Level Information

Class	Levels	Values
Family	4	1 2 3 4
Gender	2	F M

The "Dimensions" table in Figure 56.3 lists the sizes of relevant matrices. This table can be useful in determining CPU time and memory requirements.

Figure 56.3 Dimensions

Dimensions	
Covariance Parameters	1
Columns in X	15
Columns in Z	0
Subjects	1
Max Obs Per Subject	18

The "Number of Observations" table in Figure 56.4 displays information about the sample size being processed.

Figure 56.4 Number of Observations

Number of Observations	
Number of Observations Read	18
Number of Observations Used	18
Number of Observations Not Used	0

The "Covariance Parameter Estimates" table in Figure 56.5 displays the estimate of σ^2 for the model.

Figure 56.5 Covariance Parameter Estimates

Covariance Parameter Estimates	
Cov Parm	Estimate
Residual	2.1000

The "Fit Statistics" table in Figure 56.6 lists several pieces of information about the fitted mixed model, including values derived from the computed value of the restricted/residual likelihood.

Figure 56.6 Fit Statistics

Fit Statistics	
-2 Res Log Likelihood	41.6
AIC (smaller is better)	43.6
AICC (smaller is better)	44.1
BIC (smaller is better)	43.9

The "Type 3 Tests of Fixed Effects" table in Figure 56.7 displays significance tests for the three effects listed in the MODEL statement. The Type 3 F statistics and p-values are the same as those produced by the GLM procedure. However, because PROC MIXED uses a likelihood-based esti-

mation scheme, it does not directly compute or display sums of squares for this analysis.

Figure 56.7 Tests of Fixed Effects

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Gender	1	10	17.63	0.0018
Family	3	10	5.90	0.0139
Family*Gender	3	10	2.89	0.0889

The Type 3 test for Family*Gender effect is not significant at the 5% level, but the tests for both main effects are significant.

The important assumptions behind this analysis are that the data are normally distributed and that they are independent with constant variance. For these data, the normality assumption is probably realistic since the data are observed heights. However, since the data occur in clusters (families), it is very likely that observations from the same family are statistically correlated—that is, not independent.

The methods implemented in PROC MIXED are still based on the assumption of normally distributed data, but you can drop the assumption of independence by modeling statistical correlation in a variety of ways. You can also model variances that are heterogeneous—that is, nonconstant.

For the height data, one of the simplest ways of modeling correlation is through the use of *random effects*. Here the family effect is assumed to be normally distributed with zero mean and some unknown variance. This is in contrast to the previous model in which the family effects are just constants, or *fixed effects*. Declaring Family as a random effect sets up a common correlation among all observations having the same level of Family.

Declaring Family*Gender as a random effect models an additional correlation between all observations that have the same level of both Family and Gender. One interpretation of this effect is that a female in a certain family exhibits more correlation with the other females in that family than with the other males, and likewise for a male. With the height data, this model seems reasonable.

The statements to fit this correlation model in PROC MIXED are as follows:

```
proc mixed;
   class Family Gender;
   model Height = Gender;
   random Family Family*Gender;
run;
```

Note that Family and Family*Gender are now listed in the RANDOM statement. The dummy variables associated with them are used to construct the **Z** matrix in the mixed model. The **X** matrix now consists of a column of 1s and the dummy variables for Gender.

The **G** matrix for this model is diagonal, and it contains the variance components for both Family and Family*Gender. The **R** matrix is still assumed to equal $\sigma^2\mathbf{I}$, where **I** is an identity matrix.

The output from this analysis is as follows.

Figure 56.8 Model Information

The Mixed Procedure

Model Information

Data Set	WORK.HEIGHTS
Dependent Variable	Height
Covariance Structure	Variance Components
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

The "Model Information" table in Figure 56.8 shows that the containment method is used to compute the degrees of freedom for this analysis. This is the default method when a RANDOM statement is used; see the description of the DDFM= option for more information.

Figure 56.9 Class Level Information

Class Level Information

Class	Levels	Values
Family	4	1 2 3 4
Gender	2	F M

The "Class Level Information" table in Figure 56.9 is the same as before. The "Dimensions" table in Figure 56.10 displays the new sizes of the **X** and **Z** matrices.

Figure 56.10 Dimensions and Number of Observations

Dimensions

Covariance Parameters	3
Columns in X	3
Columns in Z	12
Subjects	1
Max Obs Per Subject	18

Number of Observations

Number of Observations Read	18
Number of Observations Used	18
Number of Observations Not Used	0

The "Iteration History" table in Figure 56.11 displays the results of the numerical optimization of the restricted/residual likelihood. Six iterations are required to achieve the default convergence

criterion of 1E−8.

Figure 56.11 REML Estimation Iteration History

Iteration History

Iteration	Evaluations	−2 Res Log Like	Criterion
0	1	74.11074833	
1	2	71.51614003	0.01441208
2	1	71.13845990	0.00412226
3	1	71.03613556	0.00058188
4	1	71.02281757	0.00001689
5	1	71.02245904	0.00000002
6	1	71.02245869	0.00000000

Convergence criteria met.

The "Covariance Parameter Estimates" table in Figure 56.12 displays the results of the REML fit. The Estimate column contains the estimates of the variance components for Family and Family*Gender, as well as the estimate of σ^2.

Figure 56.12 Covariance Parameter Estimates (REML)

Covariance Parameter Estimates

Cov Parm	Estimate
Family	2.4010
Family*Gender	1.7657
Residual	2.1668

The "Fit Statistics" table in Figure 56.13 contains basic information about the REML fit.

Figure 56.13 Fit Statistics

Fit Statistics

−2 Res Log Likelihood	71.0
AIC (smaller is better)	77.0
AICC (smaller is better)	79.0
BIC (smaller is better)	75.2

The "Type 3 Tests of Fixed Effects" table in Figure 56.14 contains a significance test for the lone fixed effect, Gender. Note that the associated p-value is not nearly as significant as in the previous analysis. This illustrates the importance of correctly modeling correlation in your data.

Figure 56.14 Type 3 Tests of Fixed Effects

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Gender	1	3	7.95	0.0667

An additional benefit of the random effects analysis is that it enables you to make inferences about gender that apply to an entire population of families, whereas the inferences about gender from the analysis where Family and Family*Gender are fixed effects apply only to the particular families in the data set.

PROC MIXED thus offers you the ability to model correlation directly and to make inferences about fixed effects that apply to entire populations of random effects.

Syntax: MIXED Procedure

The following statements are available in PROC MIXED.

PROC MIXED < *options* > ;
BY *variables* ;
CLASS *variables* ;
ID *variables* ;
MODEL *dependent* **=** < *fixed-effects* > < / *options* > ;
RANDOM *random-effects* < / *options* > ;
REPEATED < *repeated-effect* >< / *options* > ;
PARMS (*value-list*) . . . < / *options* > ;
PRIOR < *distribution* >< / *options* > ;
CONTRAST *'label'* < *fixed-effect values . . .* >
< | *random-effect values . . .* >, . . . < / *options* > ;
ESTIMATE *'label'* < *fixed-effect values . . .* >
< | *random-effect values . . .* >< / *options* > ;
LSMEANS *fixed-effects* < / *options* > ;
WEIGHT *variable* ;

Items within angle brackets (< >) are optional. The CONTRAST, ESTIMATE, LSMEANS, and RANDOM statements can appear multiple times; all other statements can appear only once.

The PROC MIXED and MODEL statements are required, and the MODEL statement must appear after the CLASS statement if a CLASS statement is included. The CONTRAST, ESTIMATE, LSMEANS, RANDOM, and REPEATED statements must follow the MODEL statement. The CONTRAST and ESTIMATE statements must also follow any RANDOM statements.

Table 56.1 summarizes the basic functions and important options of each PROC MIXED statement.

The syntax of each statement in Table 56.1 is described in the following sections in alphabetical order after the description of the PROC MIXED statement.

Table 56.1 Summary of PROC MIXED Statements

Statement	Description	Important Options
PROC MIXED	invokes the procedure	DATA= specifies input data set, METHOD= specifies estimation method
BY	performs multiple PROC MIXED analyses in one invocation	none
CLASS	declares qualitative variables that create indicator variables in design matrices	none
ID	lists additional variables to be included in predicted values tables	none
MODEL	specifies dependent variable and fixed effects, setting up **X**	S requests solution for fixed-effects parameters, DDFM= specifies denominator degrees of freedom method, OUTP= outputs predicted values to a data set, INFLUENCE computes influence diagnostics
RANDOM	specifies random effects, setting up **Z** and **G**	SUBJECT= creates block-diagonality, TYPE= specifies covariance structure, S requests solution for random-effects parameters, G displays estimated **G**
REPEATED	sets up **R**	SUBJECT= creates block-diagonality, TYPE= specifies covariance structure, R displays estimated blocks of **R**, GROUP= enables between-subject heterogeneity, LOCAL adds a diagonal matrix to **R**
PARMS	specifies a grid of initial values for the covariance parameters	HOLD= and NOITER hold the covariance parameters or their ratios constant, PARMSDATA= reads the initial values from a SAS data set
PRIOR	performs a sampling-based Bayesian analysis for variance component models	NSAMPLE= specifies the sample size, SEED= specifies the starting seed
CONTRAST	constructs custom hypothesis tests	E displays the **L** matrix coefficients
ESTIMATE	constructs custom scalar estimates	CL produces confidence limits
LSMEANS	computes least squares means for classification fixed effects	DIFF computes differences of the least squares means, ADJUST= performs multiple comparisons adjustments, AT changes covariates, OM changes weighting, CL produces confidence limits, SLICE= tests simple effects

Table 56.1 *continued*

Statement	Description	Important Options
WEIGHT	specifies a variable by which to weight **R**	none

PROC MIXED Statement

PROC MIXED < *options* > **;**

The PROC MIXED statement invokes the procedure. Table 56.2 summarizes important options in the PROC MIXED statement by function. These and other options in the PROC MIXED statement are then described fully in alphabetical order.

Table 56.2 PROC MIXED Statement Options

Option	Description
Basic Options	
DATA=	specifies input data set
METHOD=	specifies the estimation method
NOPROFILE	includes scale parameter in optimization
ORDER=	determines the sort order of CLASS variables
Displayed Output	
ASYCORR	displays asymptotic correlation matrix of covariance parameter estimates
ASYCOV	displays asymptotic covariance matrix of covariance parameter estimates
CL	requests confidence limits for covariance parameter estimates
COVTEST	displays asymptotic standard errors and Wald tests for covariance parameters
IC	displays a table of information criteria
ITDETAILS	displays estimates and gradients added to "Iteration History"
LOGNOTE	writes periodic status notes to the log
MMEQ	displays mixed model equations
MMEQSOL	displays the solution to the mixed model equations
NOCLPRINT	suppresses "Class Level Information" completely or in parts
NOITPRINT	suppresses "Iteration History" table
PLOTS	produces ODS statistical graphics
RATIO	produces ratio of covariance parameter estimates with residual variance
Optimization Options	
MAXFUNC=	specifies the maximum number of likelihood evaluations
MAXITER=	specifies the maximum number of iterations

Table 56.2 *continued*

Option	Description
Computational Options	
CONVF	requests and tunes the relative function convergence criterion
CONVG	requests and tunes the relative gradient convergence criterion
CONVH	requests and tunes the relative Hessian convergence criterion
DFBW	selects between-within degree of freedom method
EMPIRICAL	computes empirical ("sandwich") estimators
NOBOUND	unbounds covariance parameter estimates
RIDGE=	specifies starting value for minimum ridge value
SCORING=	applies Fisher scoring where applicable

You can specify the following options.

ABSOLUTE
makes the convergence criterion absolute. By default, it is relative (divided by the current objective function value). See the CONVF, CONVG, and CONVH options in this section for a description of various convergence criteria.

ALPHA=*number*
requests that confidence limits be constructed for the covariance parameter estimates with confidence level $1 - number$. The value of *number* must be between 0 and 1; the default is 0.05.

ANOVAF
The ANOVAF option computes F tests in models with REPEATED statement and without RANDOM statement by a method similar to that of Brunner, Domhof, and Langer (2002). The method consists of computing special F statistics and adjusting their degrees of freedom. The technique is a generalization of the Greenhouse-Geiser adjustment in MANOVA models (Greenhouse and Geiser 1959). For more details, see the section "*F* Tests With the ANOVAF Option" on page 3979.

ASYCORR
produces the asymptotic correlation matrix of the covariance parameter estimates. It is computed from the corresponding asymptotic covariance matrix (see the description of the ASYCOV option, which follows). For ODS purposes, the name of the "Asymptotic Correlation" table is "AsyCorr."

ASYCOV
requests that the asymptotic covariance matrix of the covariance parameters be displayed. By default, this matrix is the observed inverse Fisher information matrix, which equals $2\mathbf{H}^{-1}$, where $\mathbf{H}$ is the Hessian (second derivative) matrix of the objective function. See the section "Covariance Parameter Estimates" on page 3997 for more information about this matrix. When you use the SCORING= option and PROC MIXED converges without stopping the scoring algorithm, PROC MIXED uses the expected Hessian matrix to compute the covariance matrix instead of the observed Hessian. For ODS purposes, the name of the "Asymptotic Covariance" table is "AsyCov."

CL<=WALD>
requests confidence limits for the covariance parameter estimates. A Satterthwaite approximation is used to construct limits for all parameters that have a lower boundary constraint of zero. These limits take the form

$$\frac{\nu\widehat{\sigma}^2}{\chi^2_{\nu,1-\alpha/2}} \leq \sigma^2 \leq \frac{\nu\widehat{\sigma}^2}{\chi^2_{\nu,\alpha/2}}$$

where $\nu = 2Z^2$, Z is the Wald statistic $\widehat{\sigma}^2/\mathrm{se}(\widehat{\sigma}^2)$, and the denominators are quantiles of the χ^2-distribution with ν degrees of freedom. See Milliken and Johnson (1992) and Burdick and Graybill (1992) for similar techniques.

For all other parameters, Wald Z-scores and normal quantiles are used to construct the limits. Wald limits are also provided for variance components if you specify the NOBOUND option. The optional =WALD specification requests Wald limits for all parameters.

The confidence limits are displayed as extra columns in the "Covariance Parameter Estimates" table. The confidence level is $1-\alpha = 0.95$ by default; this can be changed with the ALPHA= option.

CONVF<=*number***>**
requests the relative function convergence criterion with tolerance *number*. The relative function convergence criterion is

$$\frac{|f_k - f_{k-1}|}{|f_k|} \leq number$$

where f_k is the value of the objective function at iteration k. To prevent the division by $|f_k|$, use the ABSOLUTE option. The default convergence criterion is CONVH, and the default tolerance is 1E−8.

CONVG <=*number***>**
requests the relative gradient convergence criterion with tolerance *number*. The relative gradient convergence criterion is

$$\frac{\max_j |g_{jk}|}{|f_k|} \leq number$$

where f_k is the value of the objective function, and g_{jk} is the jth element of the gradient (first derivative) of the objective function, both at iteration k. To prevent division by $|f_k|$, use the ABSOLUTE option. The default convergence criterion is CONVH, and the default tolerance is 1E−8.

CONVH<=*number***>**
requests the relative Hessian convergence criterion with tolerance *number*. The relative Hessian convergence criterion is

$$\frac{\mathbf{g}_k{}'\mathbf{H}_k^{-1}\mathbf{g}_k}{|f_k|} \leq number$$

where f_k is the value of the objective function, $\mathbf{g}_k$ is the gradient (first derivative) of the objective function, and $\mathbf{H}_k$ is the Hessian (second derivative) of the objective function, all at iteration k.

If $\mathbf{H}_k$ is singular, then PROC MIXED uses the following relative criterion:

$$\frac{\mathbf{g}_k' \mathbf{g}_k}{|f_k|} \leq number$$

To prevent the division by $|f_k|$, use the ABSOLUTE option. The default convergence criterion is CONVH, and the default tolerance is 1E−8.

COVTEST

produces asymptotic standard errors and Wald Z-tests for the covariance parameter estimates.

DATA=*SAS-data-set*

names the SAS data set to be used by PROC MIXED. The default is the most recently created data set.

DFBW

has the same effect as the DDFM=BW option in the MODEL statement.

EMPIRICAL

computes the estimated variance-covariance matrix of the fixed-effects parameters by using the asymptotically consistent estimator described in Huber (1967), White (1980), Liang and Zeger (1986), and Diggle, Liang, and Zeger (1994). This estimator is commonly referred to as the "sandwich" estimator, and it is computed as follows:

$$(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}\left(\sum_{i=1}^{S}\mathbf{X}_i'\widehat{\mathbf{V}_i}^{-1}\widehat{\boldsymbol{\epsilon}_i}\widehat{\boldsymbol{\epsilon}_i}'\widehat{\mathbf{V}_i}^{-1}\mathbf{X}_i\right)(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}$$

Here, $\widehat{\boldsymbol{\epsilon}_i} = y_i - \mathbf{X}_i\widehat{\boldsymbol{\beta}}$, S is the number of subjects, and matrices with an i subscript are those for the ith subject. You must include the SUBJECT= option in either a RANDOM or REPEATED statement for this option to take effect.

When you specify the EMPIRICAL option, PROC MIXED adjusts all standard errors and test statistics involving the fixed-effects parameters. This changes output in the following tables (listed in Table 56.22): Contrast, CorrB, CovB, Diffs, Estimates, InvCovB, LSMeans, Slices, SolutionF, Tests1–Tests3. The OUTP= and OUTPM= data sets are also affected. Finally, the Satterthwaite and Kenward-Roger degrees of freedom methods are not available if you specify the EMPIRICAL option.

IC

displays a table of various information criteria. The criteria are all in smaller-is-better form, and are described in Table 56.3.

Table 56.3 Information Criteria

Criterion	Formula	Reference
AIC	$-2\ell + 2d$	Akaike (1974)
AICC	$-2\ell + 2dn^*/(n^* - d - 1)$	Hurvich and Tsai (1989) Burnham and Anderson (1998)
HQIC	$-2\ell + 2d \log\log n$	Hannan and Quinn (1979)
BIC	$-2\ell + d \log n$	Schwarz (1978)
CAIC	$-2\ell + d(\log n + 1)$	Bozdogan (1987)

Here ℓ denotes the maximum value of the (possibly restricted) log likelihood, d the dimension of the model, and n the number of observations. In SAS 6 of SAS/STAT software, n equals the number of valid observations for maximum likelihood estimation and $n - p$ for restricted maximum likelihood estimation, where p equals the rank of $\mathbf{X}$. In later versions, n equals the number of effective subjects as displayed in the "Dimensions" table, unless this value equals 1, in which case n equals the number of levels of the first random effect you specify in a RANDOM statement. If the number of effective subjects equals 1 and you have no RANDOM statements, then n reverts to the SAS 6 values. For AICC (a finite-sample corrected version of AIC), n^* equals the SAS 6 values of n, unless this number is less than $d + 2$, in which case it equals $d + 2$.

For restricted likelihood estimation, d equals q, the effective number of estimated covariance parameters. In SAS 6, when a parameter estimate lies on a boundary constraint, then it is still included in the calculation of d, but in later versions it is not. The most common example of this behavior is when a variance component is estimated to equal zero. For maximum likelihood estimation, d equals $q + p$.

For ODS purposes, the name of the "Information Criteria" table is "InfoCrit."

INFO

is a default option. The creation of the "Model Information," "Dimensions," and "Number of Observations" tables can be suppressed by using the NOINFO option.

Note that in SAS 6 this option displays the "Model Information" and "Dimensions" tables.

ITDETAILS

displays the parameter values at each iteration and enables the writing of notes to the SAS log pertaining to "infinite likelihood" and "singularities" during Newton-Raphson iterations.

LOGNOTE

writes periodic notes to the log describing the current status of computations. It is designed for use with analyses requiring extensive CPU resources.

MAXFUNC=*number*

specifies the maximum number of likelihood evaluations in the optimization process. The default is 150.

MAXITER=*number*

specifies the maximum number of iterations. The default is 50.

METHOD=REML

METHOD=ML

METHOD=MIVQUE0

METHOD=TYPE1

METHOD=TYPE2

METHOD=TYPE3

specifies the estimation method for the covariance parameters. The REML specification performs residual (restricted) maximum likelihood, and it is the default method. The ML specification performs maximum likelihood, and the MIVQUE0 specification performs minimum variance quadratic unbiased estimation of the covariance parameters.

The METHOD=TYPE*n* specifications apply only to variance component models with no SUBJECT= effects and no REPEATED statement. An analysis of variance table is included in the output, and the expected mean squares are used to estimate the variance components (see Chapter 39, "The GLM Procedure," for further explanation). The resulting method-of-moment variance component estimates are used in subsequent calculations, including standard errors computed from ESTIMATE and LSMEANS statements. For ODS purposes, the new table names are "Type1," "Type2," and "Type3," respectively.

MMEQ

requests that coefficients of the mixed model equations be displayed. These are

$$\begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{Z} \\ \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}+\widehat{\mathbf{G}}^{-1} \end{bmatrix}, \begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \\ \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \end{bmatrix}$$

assuming that $\widehat{\mathbf{G}}$ is nonsingular. If $\widehat{\mathbf{G}}$ is singular, PROC MIXED produces the following coefficients:

$$\begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}\widehat{\mathbf{G}} \\ \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}\widehat{\mathbf{G}}+\widehat{\mathbf{G}} \end{bmatrix}, \begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \\ \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \end{bmatrix}$$

See the section "Estimating Fixed and Random Effects in the Mixed Model" on page 3976 for further information about these equations.

MMEQSOL

requests that a solution to the mixed model equations be produced, as well as the inverted coefficients matrix. Formulas for these equations are provided in the preceding description of the MMEQ option.

When $\widehat{\mathbf{G}}$ is singular, $\widehat{\boldsymbol{\tau}}$ and a generalized inverse of the left-hand-side coefficient matrix are transformed by using $\widehat{\mathbf{G}}$ to produce $\widehat{\boldsymbol{\gamma}}$ and $\widehat{\mathbf{C}}$, respectively, where $\widehat{\mathbf{C}}$ is a generalized inverse of the left-hand-side coefficient matrix of the original equations.

NAMELEN<=*number*>

specifies the length to which long effect names are shortened. The default and minimum value is 20.

NOBOUND

has the same effect as the NOBOUND option in the PARMS statement.

NOCLPRINT<=*number*>

suppresses the display of the "Class Level Information" table if you do not specify *number*. If you do specify *number*, only levels with totals that are less than *number* are listed in the table.

NOINFO

suppresses the display of the "Model Information," "Dimensions," and "Number of Observations" tables.

NOITPRINT

suppresses the display of the "Iteration History" table.

NOPROFILE

includes the residual variance as part of the Newton-Raphson iterations. This option applies only to models that have a residual variance parameter. By default, this parameter is profiled out of the likelihood calculations, except when you have specified the HOLD= option in the PARMS statement.

ORD

displays ordinates of the relevant distribution in addition to *p*-values. The ordinate can be viewed as an approximate odds ratio of hypothesis probabilities.

ORDER=DATA

ORDER=FORMATTED

ORDER=FREQ

ORDER=INTERNAL

specifies the sorting order for the levels of all CLASS variables. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option can be useful when you use CONTRAST or ESTIMATE statements.

The default is ORDER=FORMATTED, and its behavior has been modified for SAS 8. When the default ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. In releases previous to SAS 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values. In order to revert to the previous method you can specify this format explicitly for the CLASS variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and required you to use an explicit format or ORDER=INTERNAL to get the more natural ordering.

Table 56.4 shows how PROC MIXED interprets values of the ORDER= option.

Table 56.4 Sort Order and Value of ORDER= Option

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

For FORMATTED and INTERNAL, the sort order is machine dependent.

For more information about sort order, see the chapter on the SORT procedure in the *SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

PLOTS < (*global-plot-options*) > < =*plot-request* < (*options*) > >

PLOTS < (*global-plot-options*) > < = (*plot-request*< (*options*) >< . . . *plot-request*< (*options*) > >) >

requests that the MIXED procedure produce statistical graphics via the Output Delivery System, provided that the ODS GRAPHICS statement has been specified. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For examples of the basic statistical graphics produced by the MIXED procedure and aspects of their computation and interpretation, see the section "ODS Graphics" on page 4004.

The *global-plot-options* apply to all relevant plots generated by the MIXED procedure. The *global-plot-options* supported by the MIXED procedure follow.

Global Plot Options

OBSNO

uses the data set observation number to identify observations in tooltips, provided that the observation number can be determined. Otherwise, the number displayed in tooltips is the index of the observation as it is used in the analysis within the BY group.

ONLY

suppresses the default plots. Only the plots specifically requested are produced.

UNPACK

breaks a graphic that is otherwise paneled into individual component plots.

Specific Plot Options

The following listing describes the specific plots and their options.

ALL

requests that all plots appropriate for the particular analysis be produced.

BOXPLOT < (*boxplot-options*) >

requests box plots for the effects in your model that consist of classification effects only. Note that these effects can involve more than one classification variable (interaction and nested effects), but they cannot contain any continuous variables. By default, the BOXPLOT request produces box plots based on (conditional) raw residuals for the qualifying effects in the MODEL, RANDOM, and REPEATED statements. See the discussion of the *boxplot-options* in a later section for information about how to tune your box plot request.

DISTANCE< (USEINDEX) >

requests a plot of the likelihood or restricted likelihood distance. When influence diagnostics are requested with set selection according to an effect, the USEINDEX option enables you to replace the formatted tick values on the horizontal axis with integer indices of the effect levels in order to reduce the space taken up by the horizontal plot axis.

INFLUENCEESTPLOT< (*options*) >

requests panels of the deletiob estimates in an influence analysis, provided that the INFLUENCE option is specified in the MODEL statement. No plots are produced for fixed-effects parameters associated with singular columns in the **X** matrix or for covariance parameters associated with singularities in the ASYCOV matrix. By default, separate panels are produced for the fixed-effects and covariance parameters delete estimates. The FIXED and RANDOM options enable you to select these specific panels. The UNPACK option produces separate plots for each of the parameter estimates. The USEINDEX option replaces formatted tick values for the horizontal axis with integer indices.

INFLUENCESTATPANEL< (*options*) >

requests panels of influence statistics. For iterative influence analysis (see the INFLUENCE option in the MODEL statement), the panel shows the Cook's *D* and CovRatio statistics for fixed-effects and covariance parameters, enabling you to gauge impact on estimates and precision for both types of estimates. In noniterative analysis, only statistics for the fixed effects are plotted. The UNPACK option produces separate plots from the elements in the panel. The USEINDEX option replaces formatted tick values for the horizontal axis with integer indices.

RESIDUALPANEL < (*residualplot-options*) >

requests a panel of raw residuals. By default, the conditional residuals are produced. See the discussion of *residualplot-options* in a later section for information about how to tune this panel.

STUDENTPANEL < (*residualplot-options*) >

requests a panel of studentized residuals. By default, the conditional residuals are produced. See the discussion of *residualplot-options* in a later section for information about how to tune this panel.

PEARSONPANEL < (*residualplot-options*) >

requests a panel of Pearson residuals. By default, the conditional residuals are produced. See the discussion of *residualplot-options* in a later section for information about how to tune this panel.

PRESS< (USEINDEX) >

requests a plot of PRESS residuals or PRESS statistics. These are based on "leave-one-out" or "leave-set-out" prediction of the marginal mean. When influence diagnostics are requested with set selection according to an effect, the USEINDEX option enables you to replace the formatted tick values on the horizontal axis with integer indices of the effect levels in order to reduce the space taken up by the horizontal plot axis.

VCIRYPANEL < (*residualplot-options*) >

requests a panel of residual graphics based on the scaled residuals. See the VCIRY option in the MODEL statement for details about these scaled residuals. Only the UNPACK and BOX options of the *residualplot-options* are available for this type of residual panel.

NONE

suppresses all plots.

Residual Plot Options

The *residualplot-options* determine both the composition of the panels and the type of residuals being plotted.

BOX
BOXPLOT
: replaces the inset of summary statistics in the lower-right corner of the panel with a box plot of the residual (the "PROC GLIMMIX look").

CONDITIONAL
BLUP
: constructs plots from conditional residuals.

MARGINAL
NOBLUP
: constructs plots from marginal residuals.

UNPACK
: produces separate plots from the elements of the panel. The inset statistics are not part of the unpack operation.

Box Plot Options

The *boxplot-options* determine whether box plots are produced for residuals or for residuals and observed values, and for which model effects the box plots are constructed. The available *boxplot-options* are as follows.

CONDITIONAL
BLUP
: constructs box plots from conditional residuals—that is, residuals using the estimated BLUPs of random effects.

FIXED
: produces box plots for all fixed effects (MODEL statement) consisting entirely of classification variables

GROUP
: produces box plots for all GROUP= effects (RANDOM and REPEATED statement) consisting entirely of classification variables

MARGINAL
NOBLUP
: constructs box plots from marginal residuals.

NPANEL=*number*
provides the ability to break a box plot into multiple graphics. If *number* is negative, no balancing of the number of boxes takes place and *number* is the maximum number of boxes per graphic. If *number* is positive, the number of boxes per graphic is balanced. For example, suppose variable A has 125 levels, and consider the following statements:

```
ods graphics on;
proc mixed plots=boxplot(npanel=20);
    class A;
    model y = A;
run;
```

The box balancing results in six plots with 18 boxes each and one plot with 17 boxes. If *number* is zero, and this is the default, all levels of the effect are displayed in a single plot.

OBSERVED
adds box plots of the observed data for the selected effects.

RANDOM
produces box plots for all random effects (RANDOM statement) consisting entirely of classification variables. This does not include effects specified in the GROUP= or SUBJECT= options of the RANDOM statement.

REPEATED
produces box plots for the repeated effects (REPEATED statement). This does not include effects specified in the GROUP= or SUBJECT= options of the REPEATED statement.

STUDENT
constructs box plots from studentized residuals rather than from raw residuals.

SUBJECT
produces box plots for all SUBJECT= effects (RANDOM and REPEATED statement) consisting entirely of classification variables.

USEINDEX
uses as the horizontal axis label the index of the effect level rather than the formatted value(s). For classification variables with many levels or model effects that involve multiple classification variables, the formatted values identifying the effect levels can take up too much space as axis tick values, leading to extensive thinning. The USEINDEX option replaces tick values constructed from formatted values with the internal level number.

Multiple Plot Request

You can list a plot request one or more times with different options. For example, the following statements request a panel of marginal raw residuals, individual plots generated from a panel of the conditional raw residuals, and a panel of marginal studentized residuals:

```
ods graphics on;
proc mixed plots(only)=(
       ResidualPanel(marginal)
       ResidualPanel(unpack conditional)
       StudentPanel(marginal box));
```

The inset of residual statistics is replaced in this last panel by a box plot of the studentized residuals. Similarly, if you specify the INFLUENCE option in the MODEL statement, then the following statements request statistical graphics of fixed-effects deletion estimates (in a panel), covariance parameter deletion estimates (unpacked in individual plots), and box plots for the SUBJECT= and fixed classification effects based on residuals and observed values:

```
ods graphics on / imagefmt=staticmap;
proc mixed plots(only)=(
       InfluenceEstPlot(fixed)
       InfluenceEstPlot(random unpack)
       BoxPlot(observed fixed subject);
```

The STATICMAP image format enables tooltips that show, for example, values of influence diagnostics associated with a particular delete estimate.

This concludes the syntax section for the PLOTS= option in the PROC MIXED statement.

RATIO
produces the ratio of the covariance parameter estimates to the estimate of the residual variance when the latter exists in the model.

RIDGE=*number*
specifies the starting value for the minimum ridge value used in the Newton-Raphson algorithm. The default is 0.3125.

SCORING< *=number*>
requests that Fisher scoring be used in association with the estimation method up to iteration *number*, which is 0 by default. When you use the SCORING= option and PROC MIXED converges without stopping the scoring algorithm, PROC MIXED uses the expected Hessian matrix to compute approximate standard errors for the covariance parameters instead of the observed Hessian. The output from the ASYCOV and ASYCORR options is similarly adjusted.

SIGITER
is an alias for the NOPROFILE option.

UPDATE
is an alias for the LOGNOTE option.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MIXED to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement options NOTSORTED or DESCENDING in the BY statement for the MIXED procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

Because sorting the data changes the order in which PROC MIXED reads observations, the sorting order for the levels of the CLASS variable might be affected if you have specified ORDER=DATA in the PROC MIXED statement. This, in turn, affects specifications in the CONTRAST or ESTIMATE statement.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

The CLASS statement names the classification variables to be used in the analysis. If the CLASS statement is used, it must appear before the MODEL statement.

Classification variables can be either character or numeric. By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. In releases prior to SAS® 9, class levels were determined by using no more than the first 16 characters of the formatted values. If you want to revert to this previous behavior, you can use the TRUNCATE option in the CLASS statement. In any case, you can use formats to group values into levels. See the discussion of the FORMAT procedure in the *Base SAS Procedures Guide* and the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*. You can adjust the order of CLASS variable levels with the ORDER= option in the PROC MIXED statement.

You can specify the following option in the CLASS statement after a slash (/):

TRUNCATE

specifies that class levels should be determined by using no more than the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases previous to SAS® 9.

CONTRAST Statement

CONTRAST *'label'* < *fixed-effect values . . .* >
< | *random-effect values . . .* >, *. . .* < / *options* > **;**

The CONTRAST statement provides a mechanism for obtaining custom hypothesis tests. It is patterned after the CONTRAST statement in PROC GLM, although it has been extended to include random effects. This enables you to select an appropriate inference space (McLean, Sanders, and Stroup 1991).

You can test the hypothesis $\mathbf{L}'\boldsymbol{\phi} = \mathbf{0}$, where $\mathbf{L}' = (\mathbf{K}'\,\mathbf{M}')$ and $\boldsymbol{\phi}' = (\boldsymbol{\beta}'\,\boldsymbol{\gamma}')$, in several inference spaces. The inference space corresponds to the choice of $\mathbf{M}$. When $\mathbf{M} = \mathbf{0}$, your inferences apply to the entire population from which the random effects are sampled; this is known as the *broad* inference space. When all elements of $\mathbf{M}$ are nonzero, your inferences apply only to the observed levels of the random effects. This is known as the *narrow* inference space, and you can also choose it by specifying all of the random effects as fixed. The GLM procedure uses the narrow inference space. Finally, by setting to zero the portions of $\mathbf{M}$ corresponding to selected main effects and interactions, you can choose *intermediate* inference spaces. The broad inference space is usually the most appropriate, and it is used when you do not specify any random effects in the CONTRAST statement.

The CONTRAST statement has the following arguments:

label identifies the contrast in the table. A label is required for every contrast specified. Labels can be up to 200 characters and must be enclosed in quotes.

fixed-effect identifies an effect that appears in the MODEL statement. The keyword INTERCEPT can be used as an effect when an intercept is fitted in the model. You do not need to include all effects that are in the MODEL statement.

random-effect identifies an effect that appears in the RANDOM statement. The first random effect must follow a vertical bar (|); however, random effects do not have to be specified.

values are constants that are elements of the $\mathbf{L}$ matrix associated with the fixed and random effects.

The rows of $\mathbf{L}'$ are specified in order and are separated by commas. The rows of the $\mathbf{K}'$ component of $\mathbf{L}'$ are specified on the left side of the vertical bars (|). These rows test the fixed effects and are, therefore, checked for estimability. The rows of the $\mathbf{M}'$ component of $\mathbf{L}'$ are specified on the right side of the vertical bars. They test the random effects, and no estimability checking is necessary.

If PROC MIXED finds the fixed-effects portion of the specified contrast to be nonestimable (see the SINGULAR= option), then it displays a message in the log.

The following CONTRAST statement reproduces the *F* test for the effect A in the split-plot example (see Example 56.1):

```
contrast 'A broad'
          A  1 -1 0    A*B  .5 .5 -.5 -.5 0 0 ,
          A  1 0 -1    A*B  .5 .5 0 0 -.5 -.5 / df=6;
```

Note that no random effects are specified in the preceding contrast; thus, the inference space is broad. The resulting *F* test has two numerator degrees of freedom because $\mathbf{L}'$ has two rows. The denominator degrees of freedom is, by default, the residual degrees of freedom (9), but the DF= option changes the denominator degrees of freedom to 6.

The following CONTRAST statement reproduces the *F* test for A when Block and A*Block are considered fixed effects (the narrow inference space):

```
contrast 'A narrow'
          A         1 -1 0
          A*B      .5 .5 -.5 -.5 0 0 |
          A*Block .25  .25  .25  .25
                 -.25 -.25 -.25 -.25
                   0    0    0    0 ,
          A         1 0 -1
          A*B      .5 .5 0 0 -.5 -.5 |
          A*Block .25  .25  .25  .25
                   0    0    0    0
                  -.25 -.25 -.25 -.25 ;
```

The preceding contrast does not contain coefficients for B and Block, because they cancel out in estimated differences between levels of A. Coefficients for B and Block are necessary to estimate the mean of one of the levels of A in the narrow inference space (see Example 56.1).

If the elements of **L** are not specified for an effect that contains a specified effect, then the elements of the specified effect are automatically "filled in" over the levels of the higher-order effect. This feature is designed to preserve estimability for cases where there are complex higher-order effects. The coefficients for the higher-order effect are determined by equitably distributing the coefficients of the lower-level effect, as in the construction of least squares means. In addition, if the intercept is specified, it is distributed over all classification effects that are not contained by any other specified effect. If an effect is not specified and does not contain any specified effects, then all of its coefficients in **L** are set to 0. You can override this behavior by specifying coefficients for the higher-order effect.

If too many values are specified for an effect, the extra ones are ignored; if too few are specified, the remaining ones are set to 0. If no random effects are specified, the vertical bar can be omitted; otherwise, it must be present. If a SUBJECT effect is used in the RANDOM statement, then the coefficients specified for the effects in the RANDOM statement are equitably distributed across the levels of the SUBJECT effect. You can use the E option to see exactly which **L** matrix is used.

The SUBJECT and GROUP options in the CONTRAST statement are useful for the case when a SUBJECT= or GROUP= variable appears in the RANDOM statement, and you want to contrast

different subjects or groups. By default, CONTRAST statement coefficients on random effects are distributed equally across subjects and groups.

PROC MIXED handles missing level combinations of classification variables similarly to the way PROC GLM does. Both procedures delete fixed-effects parameters corresponding to missing levels in order to preserve estimability. However, PROC MIXED does not delete missing level combinations for random-effects parameters because linear combinations of the random-effects parameters are always estimable. These conventions can affect the way you specify your CONTRAST coefficients.

The CONTRAST statement computes the statistic

$$F = \frac{\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}' \mathbf{L}(\mathbf{L}'\widehat{\mathbf{C}}\mathbf{L})^{-1}\mathbf{L}' \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}}{r}$$

where $r = \text{rank}(\mathbf{L}'\widehat{\mathbf{C}}\mathbf{L})$, and approximates its distribution with an F distribution. In this expression, $\widehat{\mathbf{C}}$ is an estimate of the generalized inverse of the coefficient matrix in the mixed model equations. See the section "Inference and Test Statistics" on page 3978 for more information about this F statistic.

The numerator degrees of freedom in the F approximation are $r = \text{rank}(\mathbf{L}'\widehat{\mathbf{C}}\mathbf{L})$, and the denominator degrees of freedom are taken from the "Tests of Fixed Effects" table and corresponds to the final effect you list in the CONTRAST statement. You can change the denominator degrees of freedom by using the DF= option.

You can specify the following options in the CONTRAST statement after a slash (/).

CHISQ

requests that chi-square tests be performed in addition to any F tests. A chi-square statistic equals its corresponding F statistic times the associate numerator degrees of freedom, and the same degrees of freedom are used to compute the p-value for the chi-square test. This p-value is always less than that for the F -test, as it effectively corresponds to an F test with infinite denominator degrees of freedom.

DF=*number*

specifies the denominator degrees of freedom for the F test. The default is the denominator degrees of freedom taken from the "Tests of Fixed Effects" table and corresponds to the final effect you list in the CONTRAST statement.

E

requests that the $\mathbf{L}$ matrix coefficients for the contrast be displayed. For ODS purposes, the label of this "L Matrix Coefficients" table is "Coef."

GROUP *coeffs*

GRP *coeffs*

sets up random-effect contrasts between different groups when a GROUP= variable appears in the RANDOM statement. By default, CONTRAST statement coefficients on random effects are distributed equally across groups.

SINGULAR=*number*

tunes the estimability checking. If **v** is a vector, define ABS(**v**) to be the absolute value of the element of **v** with the largest absolute value. If ABS($\mathbf{K}' - \mathbf{K}'\mathbf{T}$) is greater than C**number* for any row of $\mathbf{K}'$ in the contrast, then **K** is declared nonestimable. Here **T** is the Hermite form matrix $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$, and C is ABS($\mathbf{K}'$) except when it equals 0, and then C is 1. The value for *number* must be between 0 and 1; the default is 1E−4.

SUBJECT *coeffs*

SUB *coeffs*

sets up random-effect contrasts between different subjects when a SUBJECT= variable appears in the RANDOM statement. By default, CONTRAST statement coefficients on random effects are distributed equally across subjects.

ESTIMATE Statement

ESTIMATE *'label' < fixed-effect values . . . >*
< | random-effect values . . . >< / options > **;**

The ESTIMATE statement is exactly like a CONTRAST statement, except only one-row **L** matrices are permitted. The actual estimate, $\mathbf{L}'\widehat{\mathbf{p}}$, is displayed along with its approximate standard error. An approximate *t* test that $\mathbf{L}'\widehat{\mathbf{p}} = 0$ is also produced.

PROC MIXED selects the degrees of freedom to match those displayed in the "Tests of Fixed Effects" table for the final effect you list in the ESTIMATE statement. You can modify the degrees of freedom by using the DF= option.

If PROC MIXED finds the fixed-effects portion of the specified estimate to be nonestimable, then it displays "Non-est" for the estimate entries.

The following examples of ESTIMATE statements compute the mean of the first level of A in the split-plot example (see Example 56.1) for various inference spaces:

```
estimate 'A1 mean narrow'    intercept 1
                             A 1 B .5 .5 A*B .5 .5 |
                             block   .25 .25 .25 .25
                             A*Block .25 .25 .25 .25
                                      0   0   0   0
                                      0   0   0   0;
estimate 'A1 mean intermed'  intercept 1
                             A 1 B .5 .5 A*B .5 .5 |
                             Block .25 .25 .25 .25;
estimate 'A1 mean broad'     intercept 1
                             A 1 B .5 .5 A*B .5 .5;
```

The construction of the **L** vector for an ESTIMATE statement follows the same rules as listed under the CONTRAST statement.

You can specify the following options in the ESTIMATE statement after a slash (/).

ALPHA=*number*

requests that a *t*-type confidence interval be constructed with confidence level 1−*number*. The value of *number* must be between 0 and 1; the default is 0.05.

CL

requests that *t*-type confidence limits be constructed. The confidence level is 0.95 by default; this can be changed with the ALPHA= option.

DF=*number*

specifies the degrees of freedom for the *t* test and confidence limits. The default is the denominator degrees of freedom taken from the "Tests of Fixed Effects" table and corresponds to the final effect you list in the ESTIMATE statement.

DIVISOR=*number*

specifies a value by which to divide all coefficients so that fractional coefficients can be entered as integer numerators.

E

requests that the **L** matrix coefficients be displayed. For ODS purposes, the name of this "L Matrix Coefficients" table is "Coef."

GROUP *coeffs*

GRP *coeffs*

sets up random-effect contrasts between different groups when a GROUP= variable appears in the RANDOM statement. By default, ESTIMATE statement coefficients on random effects are distributed equally across groups.

LOWER

LOWERTAILED

requests that the *p*-value for the *t* test be based only on values less than the *t* statistic. A two-tailed test is the default. A lower-tailed confidence limit is also produced if you specify the CL option.

SINGULAR=*number*

tunes the estimability checking as documented for the SINGULAR= option in the CONTRAST statement.

SUBJECT *coeffs*

SUB *coeffs*

sets up random-effect contrasts between different subjects when a SUBJECT= variable appears in the RANDOM statement. By default, ESTIMATE statement coefficients on random effects are distributed equally across subjects. For example, the ESTIMATE statement in the following code from Example 56.5 constructs the difference between the random slopes of the first two batches.

```
proc mixed data=rc;
   class batch;
   model y = month / s;
   random int month / type=un sub=batch s;
   estimate 'slope b1 - slope b2' | month 1 / subject 1 -1;
run;
```

UPPER

UPPERTAILED

requests that the *p*-value for the *t* test be based only on values greater than the *t* statistic. A two-tailed test is the default. An upper-tailed confidence limit is also produced if you specify the CL option.

ID Statement

ID *variables* **;**

The ID statement specifies which variables from the input data set are to be included in the OUTP= and OUTPM= data sets from the MODEL statement. If you do not specify an ID statement, then all variables are included in these data sets. Otherwise, only the variables you list in the ID statement are included. Specifying an ID statement with no variables prevents any variables from being included in these data sets.

LSMEANS Statement

LSMEANS *fixed-effects* < / *options* > **;**

The LSMEANS statement computes least squares means (LS-means) of fixed effects. As in the GLM procedure, LS-means are *predicted population margins*—that is, they estimate the marginal means over a balanced population. In a sense, LS-means are to unbalanced designs as class and subclass arithmetic means are to balanced designs. The **L** matrix constructed to compute them is the same as the **L** matrix formed in PROC GLM; however, the standard errors are adjusted for the covariance parameters in the model.

Each LS-mean is computed as $\mathbf{L}\widehat{\boldsymbol{\beta}}$, where **L** is the coefficient matrix associated with the least squares mean and $\widehat{\boldsymbol{\beta}}$ is the estimate of the fixed-effects parameter vector (see the section "Estimating Fixed and Random Effects in the Mixed Model" on page 3976). The approximate standard errors for the LS-mean is computed as the square root of $\mathbf{L}(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}\mathbf{L}'$.

LS-means can be computed for any effect in the MODEL statement that involves CLASS variables. You can specify multiple effects in one LSMEANS statement or in multiple LSMEANS statements, and all LSMEANS statements must appear after the MODEL statement. As in the ESTIMATE statement, the **L** matrix is tested for estimability, and if this test fails, PROC MIXED displays "Non-est" for the LS-means entries.

Assuming the LS-mean is estimable, PROC MIXED constructs an approximate *t* test to test the null hypothesis that the associated population quantity equals zero. By default, the denominator degrees of freedom for this test are the same as those displayed for the effect in the "Tests of Fixed Effects" table (see the section "Default Output" on page 3995).

Table 56.5 summarizes important options in the LSMEANS statement. All LSMEANS options are subsequently discussed in alphabetical order.

Table 56.5 Summary of Important LSMEANS Statement Options

Option	Description
Construction and Computation of LS-Means	
AT	modifies covariate value in computing LS-means
BYLEVEL	computes separate margins
DIFF	requests differences of LS-means
OM	specifies weighting scheme for LS-mean computation
SINGULAR=	tunes estimability checking
SLICE=	partitions *F* tests (simple effects)
Degrees of Freedom and *P*-values	
ADJDFE=	determines whether to compute row-wise denominator degrees of freedom with DDFM=SATTERTHWAITE or DDFM=KENWARDROGER
ADJUST=	determines the method for multiple comparison adjustment of LS-mean differences
ALPHA=α	determines the confidence level $(1-\alpha)$
DF=	assigns specific value to degrees of freedom for tests and confidence limits
Statistical Output	
CL	constructs confidence limits for means and or mean differences
CORR	displays correlation matrix of LS-means
COV	displays covariance matrix of LS-means
E	prints the **L** matrix

You can specify the following options in the LSMEANS statement after a slash (/).

ADJDFE=SOURCE

ADJDFE=ROW

specifies how denominator degrees of freedom are determined when p-values and confidence limits are adjusted for multiple comparisons with the ADJUST= option. When you do not specify the ADJDFE= option, or when you specify ADJDFE=SOURCE, the denominator degrees of freedom for multiplicity-adjusted results are the denominator degrees of freedom for the LS-mean effect in the "Type 3 Tests of Fixed Effects" table. When you specify ADJDFE=ROW, the denominator degrees of freedom for multiplicity-adjusted results correspond to the degrees of freedom displayed in the DF column of the "Differences of Least Squares Means" table.

The ADJDFE=ROW setting is particularly useful if you want multiplicity adjustments to take into account that denominator degrees of freedom are not constant across LS-mean differences. This can be the case, for example, when the DDFM=SATTERTHWAITE or DDFM=KENWARDROGER degrees-of-freedom method is in effect.

In one-way models with heterogeneous variance, combining certain ADJUST= options with the ADJDFE=ROW option corresponds to particular methods of performing multiplicity adjustments in the presence of heteroscedasticity. For example, the following statements fit a

heteroscedastic one-way model and perform Dunnett's T3 method (Dunnett 1980), which is based on the studentized maximum modulus (ADJUST=SMM):

```
proc mixed;
   class A;
   model y = A / ddfm=satterth;
   repeated / group=A;
   lsmeans A / adjust=smm adjdfe=row;
run;
```

If you combine the ADJDFE=ROW option with ADJUST=SIDAK, the multiplicity adjustment corresponds to the T2 method of Tamhane (1979), while ADJUST=TUKEY corresponds to the method of Games-Howell (Games and Howell 1976). Note that ADJUST=TUKEY gives the exact results for the case of fractional degrees of freedom in the one-way model, but it does not take into account that the degrees of freedom are subject to variability. A more conservative method, such as ADJUST=SMM, might protect the overall error rate better.

Unless the ADJUST= option of the LSMEANS statement is specified, the ADJDFE= option has no effect.

ADJUST=BON

ADJUST=DUNNETT

ADJUST=SCHEFFE

ADJUST=SIDAK

ADJUST=SIMULATE< **(***sim-options***)** >

ADJUST=SMM | GT2

ADJUST=TUKEY

requests a multiple comparison adjustment for the p-values and confidence limits for the differences of LS-means. By default, PROC MIXED adjusts all pairwise differences unless you specify ADJUST=DUNNETT, in which case PROC MIXED analyzes all differences with a control level. The ADJUST= option implies the DIFF option.

The BON (Bonferroni) and SIDAK adjustments involve correction factors described in Chapter 39, "The GLM Procedure," and Chapter 58, "The MULTTEST Procedure;" also see Westfall and Young (1993) and Westfall et al. (1999). When you specify ADJUST=TUKEY and your data are unbalanced, PROC MIXED uses the approximation described in Kramer (1956). Similarly, when you specify ADJUST=DUNNETT and the LS-means are correlated, PROC MIXED uses the factor-analytic covariance approximation described in Hsu (1992). The preceding references also describe the SCHEFFE and SMM adjustments.

The SIMULATE adjustment computes adjusted p-values and confidence limits from the simulated distribution of the maximum or maximum absolute value of a multivariate t random vector. All covariance parameters except the residual variance are fixed at their estimated values throughout the simulation, potentially resulting in some underdispersion. The simulation estimates q, the true $(1-\alpha)$th quantile, where $1-\alpha$ is the confidence coefficient. The default α is 0.05, and you can change this value with the ALPHA= option in the LSMEANS statement.

The number of samples is set so that the tail area for the simulated q is within γ of $1-\alpha$ with $100(1-\epsilon)\%$ confidence. In equation form,

$$P(|F(\widehat{q}) - (1-\alpha)| \leq \gamma) = 1 - \epsilon$$

where $\hat{q}$ is the simulated q and F is the true distribution function of the maximum; see Edwards and Berry (1987) for details. By default, $\gamma = 0.005$ and $\epsilon = 0.01$, placing the tail area of $\hat{q}$ within 0.005 of 0.95 with 99% confidence. The ACC= and EPS= *sim-options* reset γ and ϵ, respectively; the NSAMP= *sim-option* sets the sample size directly; and the SEED= *sim-option* specifies an integer used to start the pseudo-random number generator for the simulation. If you do not specify a seed, or if you specify a value less than or equal to zero, the seed is generated from reading the time of day from the computer clock. For additional descriptions of these and other simulation options, see the section "LSMEANS Statement" on page 2458 in Chapter 39, "The GLM Procedure."

ALPHA=*number*

requests that a *t*-type confidence interval be constructed for each of the LS-means with confidence level $1-$*number*. The value of *number* must be between 0 and 1; the default is 0.05.

AT *variable* **=** *value*

AT (*variable-list***) = (***value-list***)**

AT MEANS

enables you to modify the values of the covariates used in computing LS-means. By default, all covariate effects are set equal to their mean values for computation of standard LS-means. The AT option enables you to assign arbitrary values to the covariates. Additional columns in the output table indicate the values of the covariates.

If there is an effect containing two or more covariates, the AT option sets the effect equal to the product of the individual means rather than the mean of the product (as with standard LS-means calculations). The AT MEANS option sets covariates equal to their mean values (as with standard LS-means) and incorporates this adjustment to crossproducts of covariates.

As an example, consider the following invocation of PROC MIXED:

```
proc mixed;
   class A;
   model Y = A X1 X2 X1*X2;
   lsmeans A;
   lsmeans A / at means;
   lsmeans A / at X1=1.2;
   lsmeans A / at (X1 X2)=(1.2 0.3);
run;
```

For the first two LSMEANS statements, the LS-means coefficient for X1 is $\overline{x_1}$ (the mean of X1) and for X2 is $\overline{x_2}$ (the mean of X2). However, for the first LSMEANS statement, the coefficient for X1*X2 is $\overline{x_1 x_2}$, but for the second LSMEANS statement, the coefficient is $\overline{x_1} \cdot \overline{x_2}$. The third LSMEANS statement sets the coefficient for X1 equal to 1.2 and leaves it at $\overline{x_2}$ for X2, and the final LSMEANS statement sets these values to 1.2 and 0.3, respectively.

If a WEIGHT variable is present, it is used in processing AT variables. Also, observations with missing dependent variables are included in computing the covariate means, unless these

observations form a missing cell and the FULLX option in the MODEL statement is not in effect. You can use the E option in conjunction with the AT option to check that the modified LS-means coefficients are the ones you want.

The AT option is disabled if you specify the BYLEVEL option.

BYLEVEL

requests PROC MIXED to process the OM data set by each level of the LS-mean effect (LSMEANS effect) in question. For more details, see the OM option later in this section.

CL

requests that *t*-type confidence limits be constructed for each of the LS-means. The confidence level is 0.95 by default; this can be changed with the ALPHA= option.

CORR

displays the estimated correlation matrix of the least squares means as part of the "Least Squares Means" table.

COV

displays the estimated covariance matrix of the least squares means as part of the "Least Squares Means" table.

DF=*number*

specifies the degrees of freedom for the *t* test and confidence limits. The default is the denominator degrees of freedom taken from the "Tests of Fixed Effects" table corresponding to the LS-means effect unless the DDFM=SATTERTHWAITE or DDFM=KENWARDROGER option is in effect in the MODEL statement. For these DDFM= methods, degrees of freedom are determined separately for each test; see the DDFM= option for more information.

DIFF<=*difftype***>**

PDIFF<=*difftype***>**

requests that differences of the LS-means be displayed. The optional *difftype* specifies which differences to produce, with possible values being ALL, CONTROL, CONTROLL, and CONTROLU. The *difftype* ALL requests all pairwise differences, and it is the default. The *difftype* CONTROL requests the differences with a control, which, by default, is the first level of each of the specified LSMEANS effects.

To specify which levels of the effects are the controls, list the quoted formatted values in parentheses after the keyword CONTROL. For example, if the effects A, B, and C are classification variables, each having two levels, 1 and 2, the following LSMEANS statement specifies the (1,2) level of A*B and the (2,1) level of B*C as controls:

```
lsmeans A*B B*C / diff=control('1' '2' '2' '1');
```

For multiple effects, the results depend upon the order of the list, and so you should check the output to make sure that the controls are correct.

Two-tailed tests and confidence limits are associated with the CONTROL *difftype*. For one-tailed results, use either the CONTROLL or CONTROLU *difftype*. The CONTROLL *difftype* tests whether the noncontrol levels are significantly smaller than the control; the upper confidence limits for the control minus the noncontrol levels are considered to be infinity and

are displayed as missing. Conversely, the CONTROLU *difftype* tests whether the noncontrol levels are significantly larger than the control; the upper confidence limits for the noncontrol levels minus the control are considered to be infinity and are displayed as missing.

If you want to perform multiple comparison adjustments on the differences of LS-means, you must specify the ADJUST= option.

The differences of the LS-means are displayed in a table titled "Differences of Least Squares Means." For ODS purposes, the table name is "Diffs."

E

requests that the **L** matrix coefficients for all LSMEANS effects be displayed. For ODS purposes, the name of this "**L** Matrix Coefficients" table is "Coef."

OM< =*OM-data-set* >

OBSMARGINS< =*OM-data-set* >

specifies a potentially different weighting scheme for the computation of LS-means coefficients. The standard LS-means have equal coefficients across classification effects; however, the OM option changes these coefficients to be proportional to those found in *OM-data-set*. This adjustment is reasonable when you want your inferences to apply to a population that is not necessarily balanced but has the margins observed in *OM-data-set*.

By default, *OM-data-set* is the same as the analysis data set. You can optionally specify another data set that describes the population for which you want to make inferences. This data set must contain all model variables except for the dependent variable (which is ignored if it is present). In addition, the levels of all CLASS variables must be the same as those occurring in the analysis data set. Specifying an *OM-data-set* enables you to construct arbitrarily weighted LS-means.

In computing the observed margins, PROC MIXED uses all observations for which there are no missing or invalid independent variables, including those for which there are missing dependent variables. Also, if *OM-data-set* has a WEIGHT variable, PROC MIXED uses weighted margins to construct the LS-means coefficients. If *OM-data-set* is balanced, the LS-means are unchanged by the OM option.

The BYLEVEL option modifies the observed-margins LS-means. Instead of computing the margins across all of the *OM-data-set*, PROC MIXED computes separate margins for each level of the LSMEANS effect in question. In this case the resulting LS-means are actually equal to raw means for fixed-effects models and certain balanced random-effects models, but their estimated standard errors account for the covariance structure that you have specified. If the AT option is specified, the BYLEVEL option disables it.

You can use the E option in conjunction with either the OM or BYLEVEL option to check that the modified LS-means coefficients are the ones you want. It is possible that the modified LS-means are not estimable when the standard ones are, or vice versa. Nonestimable LS-means are noted as "Non-est" in the output.

PDIFF

is the same as the DIFF option.

SINGULAR=*number*

tunes the estimability checking as documented for the SINGULAR= option in the CONTRAST statement.

SLICE= *fixed-effect*

SLICE= (*fixed-effects*)

specifies effects by which to partition interaction LSMEANS effects. This can produce what are known as tests of simple effects (Winer 1971). For example, suppose that A*B is significant, and you want to test the effect of A for each level of B. The appropriate LSMEANS statement is as follows:

```
lsmeans A*B / slice=B;
```

This code tests for the simple main effects of A for B, which are calculated by extracting the appropriate rows from the coefficient matrix for the A*B LS-means and by using them to form an F test. See the section "Inference and Test Statistics" for more information about this F test.

The SLICE option produces a table titled "Tests of Effect Slices." For ODS purposes, the table name is "Slices."

MODEL Statement

MODEL *dependent* **=** < *fixed-effects* >< / *options* > **;**

The MODEL statement names a single dependent variable and the fixed effects, which determine the **X** matrix of the mixed model (see the section "Parameterization of Mixed Models" on page 3981 for details). The specification of effects is the same as in the GLM procedure; however, unlike PROC GLM, you do not specify random effects in the MODEL statement. The MODEL statement is required.

An intercept is included in the fixed-effects model by default. If no fixed effects are specified, only this intercept term is fit. The intercept can be removed by using the NOINT option.

Table 56.6 summarizes options in the MODEL statement. These are subsequently discussed in detail in alphabetical order.

Table 56.6 Summary of Important MODEL Statement Options

Option	Description
Model Building	
NOINT	excludes fixed-effect intercept from model
Statistical Computations	
ALPHA=α	determines the confidence level $(1-\alpha)$ for fixed effects
ALPHAP=α	determines the confidence level $(1-\alpha)$ for predicted values
CHISQ	requests chi-square tests
DDF=	specifies denominator degrees of freedom (list)

Table 56.6 *continued*

Option	Description
DDFM=	specifies the method for computing denominator degrees of freedom
HTYPE=	selects the type of hypothesis test
INFLUENCE	requests influence and case-deletion diagnostics
NOTEST	suppresses hypothesis tests for the fixed effects
OUTP=	specifies output data set for predicted values and related quantities
OUTPM=	specifies output data set for predicted values and related quantities
RESIDUAL	adds Pearson-type and studentized residuals to output data sets
VCIRY	adds scaled marginal residual to output data sets
Statistical Output	
CL	displays confidence limits for fixed-effects parameter estimates
CORRB	displays correlation matrix of fixed-effects parameter estimates
COVB	displays covariance matrix of fixed-effects parameter estimates
COVBI	displays inverse covariance matrix of fixed-effects parameter estimates
E, E1, E2, E3	displays **L** matrix coefficients
INTERCEPT	adds a row for the intercept to test tables
SOLUTION	displays fixed-effects parameter estimates (and scale parameter in GLM models)
Singularity Tolerances	
SINGCHOL=	tunes sensitivity in computing Cholesky roots
SINGRES=	tunes singularity criterion for residual variance
SINGULAR=	tunes the sensitivity in sweeping
ZETA=	tunes the sensitivity in forming Type 3 functions

You can specify the following options in the MODEL statement after a slash (/).

ALPHA=*number*

requests that a *t*-type confidence interval be constructed for each of the fixed-effects parameters with confidence level 1−*number*. The value of *number* must be between 0 and 1; the default is 0.05.

ALPHAP=*number*

requests that a *t*-type confidence interval be constructed for the predicted values with confidence level 1−*number*. The value of *number* must be between 0 and 1; the default is 0.05.

CHISQ

requests that chi-square tests be performed for all specified effects in addition to the *F* tests. Type 3 tests are the default; you can produce the Type 1 and Type 2 tests by using the HTYPE= option.

CL

requests that *t*-type confidence limits be constructed for each of the fixed-effects parameter

estimates. The confidence level is 0.95 by default; this can be changed with the ALPHA= option.

CONTAIN
has the same effect as the DDFM=CONTAIN option.

CORRB
produces the approximate correlation matrix of the fixed-effects parameter estimates. For ODS purposes, the name of this table is "CorrB."

COVB
produces the approximate variance-covariance matrix of the fixed-effects parameter estimates $\widehat{\boldsymbol{\beta}}$. By default, this matrix equals $(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}$ and results from sweeping $(\mathbf{X}\ \mathbf{y})'\widehat{\mathbf{V}}^{-1}(\mathbf{X}\ \mathbf{y})$ on all but its last pivot and removing the y border. The EMPIRICAL option in the PROC MIXED statement changes this matrix into "empirical sandwich" form. For ODS purposes, the name of this table is "CovB." If the degrees-of-freedom method of Kenward and Roger (1997) is in effect (DDFM=KENWARDROGER), the COVB matrix changes because the method entails an adjustment of the variance-covariance matrix of the fixed effects by the method proposed by Prasad and Rao (1990) and Harville and Jeske (1992); see also Kackar and Harville (1984).

COVBI
produces the inverse of the approximate variance-covariance matrix of the fixed-effects parameter estimates. For ODS purposes, the name of this table is "InvCovB."

DDF=*value-list*
enables you to specify your own denominator degrees of freedom for the fixed effects. The *value-list* specification is a list of numbers or missing values (.) separated by commas. The degrees of freedom should be listed in the order in which the effects appear in the "Tests of Fixed Effects" table. If you want to retain the default degrees of freedom for a particular effect, use a missing value for its location in the list. For example, the following statement assigns 3 denominator degrees of freedom to A and 4.7 to A*B, while those for B remain the same:

```
model Y = A B A*B / ddf=3,.,4.7;
```

If you specify DDFM=SATTERTHWAITE or DDFM=KENWARDROGER, the DDF= option has no effect.

DDFM=CONTAIN
DDFM=BETWITHIN
DDFM=RESIDUAL
DDFM=SATTERTHWAITE
DDFM=KENWARDROGER<(FIRSTORDER)>
specifies the method for computing the denominator degrees of freedom for the tests of fixed effects resulting from the MODEL, CONTRAST, ESTIMATE, and LSMEANS statements.

Table 56.7 lists syntax aliases for the degrees-of-freedom methods.

Table 56.7 Aliases for DDFM= Option

DDFM= Option	Alias
BETWITHIN	BW
CONTAIN	CON
KENWARDROGER	KENROG, KR
RESIDUAL	RES
SATTERTHWAITE	SATTERTH, SAT

The DDFM=CONTAIN option invokes the *containment method* to compute denominator degrees of freedom, and it is the default when you specify a RANDOM statement. The containment method is carried out as follows: Denote the fixed effect in question A, and search the RANDOM effect list for the effects that *syntactically* contain A. For example, the random effect B(A) contains A, but the random effect C does not, even if it has the same levels as B(A).

Among the random effects that contain A, compute their rank contribution to the $(\mathbf{X}\ \mathbf{Z})$ matrix. The DDF assigned to A is the smallest of these rank contributions. If no effects are found, the DDF for A is set equal to the residual degrees of freedom, $N - \text{rank}(\mathbf{X}\ \mathbf{Z})$. This choice of DDF matches the tests performed for balanced split-plot designs and should be adequate for moderately unbalanced designs.

CAUTION: If you have a $\mathbf{Z}$ matrix with a large number of columns, the overall memory requirements and the computing time after convergence can be substantial for the containment method. If it is too large, you might want to use the DDFM=BETWITHIN option.

The DDFM=BETWITHIN option is the default for REPEATED statement specifications (with no RANDOM statements). It is computed by dividing the residual degrees of freedom into between-subject and within-subject portions. PROC MIXED then checks whether a fixed effect changes within any subject. If so, it assigns within-subject degrees of freedom to the effect; otherwise, it assigns the between-subject degrees of freedom to the effect (see Schluchter and Elashoff 1990). If there are multiple within-subject effects containing classification variables, the within-subject degrees of freedom are partitioned into components corresponding to the subject-by-effect interactions.

One exception to the preceding method is the case where you have specified no RANDOM statements and a REPEATED statement with the TYPE=UN option. In this case, all effects are assigned the between-subject degrees of freedom to provide for better small-sample approximations to the relevant sampling distributions. DDFM=KENWARDROGER might be a better option to try for this case.

The DDFM=RESIDUAL option performs all tests by using the residual degrees of freedom, $n - \text{rank}(\mathbf{X})$, where n is the number of observations.

The DDFM=SATTERTHWAITE option performs a general Satterthwaite approximation for the denominator degrees of freedom, computed as follows. Suppose $\boldsymbol{\theta}$ is the vector of unknown parameters in $\mathbf{V}$, and suppose $\mathbf{C} = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-}$, where $^{-}$ denotes a generalized inverse. Let $\widehat{\mathbf{C}}$ and $\widehat{\boldsymbol{\theta}}$ be the corresponding estimates.

Consider the one-dimensional case, and consider ℓ to be a vector defining an estimable linear combination of $\boldsymbol{\beta}$. The Satterthwaite degrees of freedom for the t statistic

$$t = \frac{\ell\widehat{\boldsymbol{\beta}}}{\sqrt{\ell\hat{\mathbf{C}}\ell'}}$$

is computed as

$$\nu = \frac{2(\ell\hat{\mathbf{C}}\ell')^2}{\mathbf{g}'\mathbf{A}\mathbf{g}}$$

where $\mathbf{g}$ is the gradient of $\ell\mathbf{C}\ell'$ with respect to $\boldsymbol{\theta}$, evaluated at $\widehat{\boldsymbol{\theta}}$, and $\mathbf{A}$ is the asymptotic variance-covariance matrix of $\widehat{\boldsymbol{\theta}}$ obtained from the second derivative matrix of the likelihood equations.

For the multidimensional case, let $\mathbf{L}$ be an estimable contrast matrix and denote the rank of $\mathbf{L}\widehat{\mathbf{C}}\mathbf{L}'$ as $q > 1$. The Satterthwaite denominator degrees of freedom for the F statistic

$$F = \frac{\widehat{\boldsymbol{\beta}}'\mathbf{L}'(\mathbf{L}\widehat{\mathbf{C}}\mathbf{L}')^{-1}\mathbf{L}\widehat{\boldsymbol{\beta}}}{q}$$

are computed by first performing the spectral decomposition $\mathbf{L}\widehat{\mathbf{C}}\mathbf{L}' = \mathbf{P}'\mathbf{D}\mathbf{P}$, where $\mathbf{P}$ is an orthogonal matrix of eigenvectors and $\mathbf{D}$ is a diagonal matrix of eigenvalues, both of dimension $q \times q$. Define ℓ_m to be the mth row of $\mathbf{PL}$, and let

$$\nu_m = \frac{2(D_m)^2}{\mathbf{g}_m'\mathbf{A}\mathbf{g}_m}$$

where D_m is the mth diagonal element of $\mathbf{D}$ and $\mathbf{g}_m$ is the gradient of $\ell_m\mathbf{C}\ell_m'$ with respect to $\boldsymbol{\theta}$, evaluated at $\widehat{\boldsymbol{\theta}}$. Then let

$$E = \sum_{m=1}^{q} \frac{\nu_m}{\nu_m - 2} I(\nu_m > 2)$$

where the indicator function eliminates terms for which $\nu_m \leq 2$. The degrees of freedom for F are then computed as

$$\nu = \frac{2E}{E - q}$$

provided $E > q$; otherwise ν is set to zero.

This method is a generalization of the techniques described in Giesbrecht and Burns (1985), McLean and Sanders (1988), and Fai and Cornelius (1996). The method can also include estimated random effects. In this case, append $\widehat{\boldsymbol{\gamma}}$ to $\widehat{\boldsymbol{\beta}}$ and change $\widehat{\mathbf{C}}$ to be the inverse of the coefficient matrix in the mixed model equations. The calculations require extra memory to hold c matrices that are the size of the mixed model equations, where c is the number of covariance parameters. In the notation of Table 56.25, this is approximately $8q(p+g)(p+g)/2$ bytes. Extra computing time is also required to process these matrices. The Satterthwaite method implemented here is intended to produce an accurate F approximation; however, the results can differ from those produced by PROC GLM. Also, the small sample properties of this

approximation have not been extensively investigated for the various models available with PROC MIXED.

The DDFM=KENWARDROGER option performs the degrees of freedom calculations detailed by Kenward and Roger (1997). This approximation involves inflating the estimated variance-covariance matrix of the fixed and random effects by the method proposed by Prasad and Rao (1990) and Harville and Jeske (1992); see also Kackar and Harville (1984). Satterthwaite-type degrees of freedom are then computed based on this adjustment. By default, the observed information matrix of the covariance parameter estimates is used in the calculations. For covariance structures that have nonzero second derivatives with respect to the covariance parameters, the Kenward-Roger covariance matrix adjustment includes a second-order term. This term can result in standard error shrinkage. Also, the resulting adjusted covariance matrix can then be indefinite and is not invariant under reparameterization. The FIRSTORDER suboption of the DDFM=KENWARDROGER option eliminates the second derivatives from the calculation of the covariance matrix adjustment. For the case of scalar estimable functions, the resulting estimator is referred to as the Prasad-Rao estimator $\widetilde{m}^{@}$ in Harville and Jeske (1992). The following are examples of covariance structures that generally lead to nonzero second derivatives: TYPE=ANTE(1), TYPE=AR(1), TYPE=ARH(1), TYPE=ARMA(1,1), TYPE=CSH, TYPE=FA, TYPE=FA0(q), TYPE=TOEPH, TYPE=UNR, and all TYPE=SP() structures.

When the asymptotic variance matrix of the covariance parameters is found to be singular, a generalized inverse is used. Covariance parameters with zero variance then do not contribute to the degrees-of-freedom adjustment for DDFM=SATTERTHWAITE and DDFM=KENWARDROGER, and a message is written to the log.

This method changes output in the following tables (listed in Table 56.22): Contrast, CorrB, CovB, Diffs, Estimates, InvCovB, LSMeans, Slices, SolutionF, SolutionR, Tests1–Tests3. The OUTP= and OUTPM= data sets are also affected.

E

requests that Type 1, Type 2, and Type 3 **L** matrix coefficients be displayed for all specified effects. For ODS purposes, the name of the table is "Coef."

E1

requests that Type 1 **L** matrix coefficients be displayed for all specified effects. For ODS purposes, the name of the table is "Coef."

E2

requests that Type 2 **L** matrix coefficients be displayed for all specified effects. For ODS purposes, the name of the table is "Coef."

E3

requests that Type 3 **L** matrix coefficients be displayed for all specified effects. For ODS purposes, the name of the table is "Coef."

FULLX

requests that columns of the **X** matrix that consist entirely of zeros not be eliminated from **X**; otherwise, they are eliminated by default. For a column corresponding to a missing cell to be added to **X**, its particular levels must be present in at least one observation in the analysis

data set along with a missing dependent variable. The use of the FULLX option can affect coefficient specifications in the CONTRAST and ESTIMATE statements, as well as covariate coefficients from LSMEANS statements specified with the AT MEANS option.

HTYPE=*value-list*

indicates the type of hypothesis test to perform on the fixed effects. Valid entries for *value* are 1, 2, and 3; the default value is 3. You can specify several types by separating the values with a comma or a space. The ODS table names are "Tests1" for the Type 1 tests, "Tests2" for the Type 2 tests, and "Tests3" for the Type 3 tests.

INFLUENCE< (*influence-options***) >**

specifies that influence and case deletion diagnostics are to be computed.

The INFLUENCE option computes influence diagnostics by noniterative or iterative methods. The noniterative diagnostics rely on recomputation formulas under the assumption that covariance parameters or their ratios remain fixed. With the possible exception of a profiled residual variance, no covariance parameters are updated. This is the default behavior because of its computational efficiency. However, the impact of an observation on the overall analysis can be underestimated if its effect on covariance parameters is not assessed. Toward this end, iterative methods can be applied to gauge the overall impact of observations and to obtain influence diagnostics for the covariance parameter estimates.

If you specify the INFLUENCE option without further suboptions, PROC MIXED computes single-case deletion diagnostics and influence statistics for each observation in the data set by updating estimates for the fixed-effects parameter estimates, and also the residual variance, if it is profiled. The EFFECT=, SELECT=, ITER=, SIZE=, and KEEP= suboptions provide additional flexibility in the computation and reporting of influence statistics. Table 56.8 briefly describes important suboptions and their effect on the influence analysis.

Table 56.8 Summary of INFLUENCE Default and Suboptions

Description	Suboption
Compute influence diagnostics for individual observations	default
Measure influence of sets of observations chosen according to a classification variable or effect	EFFECT=
Remove pairs of observations and report the results sorted by degree of influence	SIZE=2
Remove triples, quadruples of observations, etc.	SIZE=
Allow selection of individual observations, observations sharing specific levels of effects, and construction of tuples from specified subsets of observations	SELECT=
Update fixed effects and covariance parameters by refitting the mixed model, adding up to n iterations	ITER=$n > 0$
Compute influence diagnostics for the covariance parameters	ITER=$n > 0$
Update only fixed effects and the residual variance, if it is profiled	ITER=0
Add the reduced-data estimates to the data set created with ODS OUTPUT	ESTIMATES

The modifiers and their default values are discussed in the following paragraphs. The set of computed influence diagnostics varies with the suboptions. The most extensive set of influence diagnostics is obtained when ITER=*n* with $n > 0$.

You can produce statistical graphics of influence diagnostics when the ODS GRAPHICS statement is specified. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the MIXED procedure, see the section "ODS Graphics" on page 4004.

You can specify the following *influence-options* in parentheses:

EFFECT=*effect*

specifies an effect according to which observations are grouped. Observations sharing the same level of the *effect* are removed from the analysis as a group. The *effect* must contain only classification variables, but they need not be contained in the model.

Removing observations can change the rank of the $(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-}$ matrix. This is particularly likely to happen when multiple observations are eliminated from the analysis. If the rank of the estimated variance-covariance matrix of $\widehat{\boldsymbol{\beta}}$ changes or its singularity pattern is altered, no influence diagnostics are computed.

ESTIMATES

EST

specifies that the updated parameter estimates should be written to the ODS output data set. The values are not displayed in the "Influence" table, but if you use ODS OUTPUT to create a data set from the listing, the estimates are added to the data set. If ITER=0, only the fixed-effects estimates are saved. In iterative influence analyses, fixed-effects and covariance parameters are stored. The p fixed-effects parameter estimates are named Parm1–Parm$p$, and the q covariance parameter estimates are named CovP1–CovPq. The order corresponds to that in the "Solution for Fixed Effects" and "Covariance Parameter Estimates" tables. If parameter updates fail—for example, because of a loss of rank or a nonpositive definite Hessian—missing values are reported.

ITER=*n*

controls the maximum number of additional iterations PROC MIXED performs to update the fixed-effects and covariance parameter estimates following data point removal. If you specify $n > 0$, then statistics such as DFFITS, MDFFITS, and the likelihood distances measure the impact of observation(s) on all aspects of the analysis. Typically, the influence will grow compared to values at ITER=0. In models without RANDOM or REPEATED effects, the ITER= option has no effect.

This documentation refers to analyses when $n > 0$ simply as iterative influence analysis, even if final covariance parameter estimates can be updated in a single step (for example, when METHOD=MIVQUE0 or METHOD=TYPE3). This nomenclature reflects the fact that only if $n > 0$ are all model parameters updated, which can require additional iterations. If $n > 0$ and METHOD=REML (default) or METHOD=ML, the procedure updates fixed effects *and* variance-covariance parameters after removing the selected observations with additional Newton-Raphson iterations, starting from the converged estimates for the entire data. The process stops for each observation or set of observations if the convergence criterion is satisfied or the number of further iterations

exceeds *n*. If $n > 0$ and METHOD=TYPE1, TYPE2, or TYPE3, ANOVA estimates of the covariance parameters are recomputed in a single step.

Compared to noniterative updates, the computations are more involved. In particular for large data sets and/or a large number of random effects, iterative updates require considerably more resources. A one-step (ITER=1) or two-step update might be a good compromise. The output includes the number of iterations performed, which is less than n if the iteration converges. If the process does not converge in n iterations, you should be careful in interpreting the results, especially if n is fairly large.

Bounds and other restrictions on the covariance parameters carry over from the full-data model. Covariance parameters that are not iterated in the model fit to the full data (the NOITER or HOLD option in the PARMS statement) are likewise not updated in the refit. In certain models, such as random-effects models, the ratios between the covariance parameters and the residual variance are maintained rather than the actual value of the covariance parameter estimate (see the section "Influence Diagnostics" on page 3988).

KEEP=*n*

determines how many observations are retained for display and in the output data set or how many tuples if you specify SIZE=. The output is sorted by an influence statistic as discussed for the SIZE= suboption.

SELECT=*value-list*

specifies which observations or effect levels are chosen for influence calculations. If the SELECT= suboption is not specified, diagnostics are computed as follows:

- for all observations, if EFFECT= or SIZE= are not given
- for all levels of the specified effect, if EFFECT= is specified
- for all tuples of size k formed from the observations in *value-list*, if SIZE=k is specified

When you specify an effect with the EFFECT= option, the values in *value-list* represent indices of the levels in the order in which PROC MIXED builds classification effects. Which observations in the data set correspond to this index depends on the order of the variables in the CLASS statement, not the order in which the variables appear in the interaction effect. See the section "Parameterization of Mixed Models" on page 3981 to understand precisely how the procedure indexes nested and crossed effects and how levels of classification variables are ordered. The actual values of the classification variables involved in the effect are shown in the output so you can determine which observations were removed.

If the EFFECT= suboption is not specified, the SELECT= value list refers to the sequence in which observations are read from the input data set or from the current BY group if there is a BY statement. This indexing is not necessarily the same as the observation numbers in the input data set, for example, if a WHERE clause is specified or during BY processing.

SIZE=*n*

instructs PROC MIXED to remove groups of observations formed as tuples of size n. For example, SIZE=2 specifies all $n \times (n-1)/2$ unique pairs of observations. The number of tuples for SIZE=k is $n!/(k!(n-k)!)$ and grows quickly with n and k. Using the SIZE= option can result in considerable computing time. The MIXED procedure displays by default only the 50 tuples with the greatest influence. Use the KEEP= option to override this default and to retain a different number of tuples in the listing or ODS output data set. Regardless of the KEEP= specification, all tuples are evaluated and the results are ordered according to an influence statistic. This statistic is the (restricted) likelihood distance as a measure of overall influence if ITER= $n > 0$ or when a residual variance is profiled. When likelihood distances are unavailable, the results are ordered by the PRESS statistic.

To reduce computational burden, the SIZE= option can be combined with the SELECT=*value-list* modifier. For example, the following statements evaluate all $15 = 6 \times 5/2$ pairs formed from observations 13, 14, 18, 30, 31, and 33 and display the five pairs with the greatest influence:

```
proc mixed;
   class a m f;
   model penetration = a m /
                  influence(size=2 keep=5
                            select=13,14,18,30,31,33);
   random f(m);
run;
```

If any observation in a tuple contains missing values or has otherwise not contributed to the analysis, the tuple is not evaluated. This guarantees that the displayed results refer to the same number of observations, so that meaningful statistics are available by which to order the results. If computations fail for a particular tuple—for example, because the $(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-}$ matrix changes rank or the $\mathbf{G}$ matrix is not positive definite—no results are produced. Results are retained when the maximum number of iterative updates is exceeded in iterative influence analyses.

The SIZE= suboption cannot be combined with the EFFECT= suboption. As in the case of the EFFECT= suboption, the statistics being computed are those appropriate for removal of multiple data points, even if SIZE=1.

For ODS purposes the name of the "Influence Diagnostics" table is "Influence." The variables in this table depend on whether you specify the EFFECT=, SIZE=, or KEEP= suboption and whether covariance parameters are iteratively updated. When ITER=0 (the default), certain influence diagnostics are meaningful only if the residual variance is profiled. Table 56.9 and Table 56.10 summarize the statistics obtained depending on the model and modifiers. The last column in these tables gives the variable name in the ODS OUTPUT INFLUENCE= data set. Restricted likelihood distances are reported instead of the likelihood distance unless METHOD=ML. See the section "Influence Diagnostics" on page 3988 for details about the individual statistics.

Table 56.9 Statistics Computed with INFLUENCE Option, Noniterative Analysis (ITER=0)

Suboption	σ^2 Profiled	Statistic	Variable Name
Default	Yes	Observed value	Observed
		Predicted value	Predicted
		Residual	Residual
		Leverage	Leverage
		PRESS residual	PRESSRes
		Internally studentized residual	Student
		Externally studentized residual	RStudent
		RMSE without deleted obs	RMSE
		Cook's D	CookD
		DFFITS	DFFITS
		CovRatio	COVRATIO
		(Restricted) likelihood distance	RLD, LD
Default	No	Observed value	Observed
		Predicted value	Predicted
		Residual	Residual
		Leverage	Leverage
		PRESS residual	PRESSRes
		Internally studentized residual	Student
		Cook's D	CookD
EFFECT=, SIZE=, or KEEP=	Yes	Observations in level (tuple)	Nobs
		PRESS statistic	PRESS
		Cook's D	CookD
		MDFFITS	MDFFITS
		CovRatio	COVRATIO
		COVTRACE	COVTRACE
		RMSE without deleted level (tuple)	RMSE
		(Restricted) likelihood distance	RLD, LD
EFFECT=, SIZE=, or KEEP=	No	Observations in level (tuple)	Nobs
		PRESS statistic	PRESS
		Cook's D	CookD

Table 56.10 Statistics Computed with INFLUENCE Option, Iterative Analysis (ITER=$n > 0$)

Suboption	Statistic	Variable Name
Default	Number of iterations	Iter
	Observed value	Observed
	Predicted value	Predicted
	Residual	Residual
	Leverage	Leverage

Table 56.10 *continued*

Suboption	Statistic	Variable Name
	PRESS residual	PRESSres
	Internally studentized residual	Student
	Externally studentized residual	RStudent
	RMSE without deleted obs (if possible)	RMSE
	Cook's D	CookD
	DFFITS	DFFITS
	CovRatio	COVRATIO
	Cook's D CovParms	CookDCP
	CovRatio CovParms	COVRATIOCP
	MDFFITS CovParms	MDFFITSCP
	(Restricted) likelihood distance	RLD, LD
EFFECT=, SIZE=, or KEEP=	Observations in level (tuple)	Nobs
	Number of iterations	Iter
	PRESS statistic	PRESS
	RMSE without deleted level (tuple)	RMSE
	Cook's D	CookD
	MDFFITS	MDFFITS
	CovRatio	COVRATIO
	COVTRACE	COVTRACE
	Cook's D CovParms	CookDCP
	CovRatio CovParms	COVRATIOCP
	MDFFITS CovParms	MDFFITSCP
	(Restricted) likelihood distance	RLD, LD

INTERCEPT

adds a row to the tables for Type 1, 2, and 3 tests corresponding to the overall intercept.

LCOMPONENTS

requests an estimate for each row of the **L** matrix used to form tests of fixed effects. Components corresponding to Type 3 tests are the default; you can produce the Type 1 and Type 2 component estimates with the HTYPE= option.

Tests of fixed effects involve testing of linear hypotheses of the form $\mathbf{L}\boldsymbol{\beta} = \mathbf{0}$. The matrix **L** is constructed from Type 1, 2, or 3 estimable functions. By default the MIXED procedure constructs Type 3 tests. In many situations, the individual rows of the matrix **L** represent contrasts of interest. For example, in a one-way classification model, the Type 3 estimable functions define differences of factor-level means. In a balanced two-way layout, the rows of **L** correspond to differences of cell means.

For example, suppose factors A and B have a and b levels, respectively. The following statements produce $(a-1)$ one degree of freedom tests for the rows of **L** associated with the Type 1 and Type 3 estimable functions for factor A, $(b-1)$ tests for the rows of **L** associated with factor B, and a single test for the Type 1 and Type 3 coefficients associated with regressor X:

```
class A B;
model y = A B x / htype=1,3 lcomponents;
```

The denominator degrees of freedom associated with a row of **L** are the same as those in the corresponding "Tests of Fixed Effects" table, except for DDFM=KENWARDROGER and DDFM=SATTERTHWAITE. For these degree of freedom methods, the denominator degrees of freedom are computed separately for each row of **L**.

For ODS purposes, the name of the table containing all requested component tests is "LComponents." See Example 56.9 for applications of the LCOMPONENTS option.

NOCONTAIN
has the same effect as the DDFM=RESIDUAL option.

NOINT
requests that no intercept be included in the model. An intercept is included by default.

NOTEST
specifies that no hypothesis tests be performed for the fixed effects.

OUTP=*SAS-data-set*

OUTPRED=*SAS-data-set*
specifies an output data set containing predicted values and related quantities. This option replaces the P option from SAS 6.

Predicted values are formed by using the rows from (**X Z**) as **L** matrices. Thus, predicted values from the original data are $\mathbf{X}\widehat{\boldsymbol{\beta}} + \mathbf{Z}\widehat{\boldsymbol{\gamma}}$. Their approximate standard errors of prediction are formed from the quadratic form of **L** with $\widehat{\mathbf{C}}$ defined in the section "Statistical Properties" on page 3977. The L95 and U95 variables provide a *t*-type confidence interval for the predicted values, and they correspond to the L95M and U95M variables from the GLM and REG procedures for fixed-effects models. The residuals are the observed minus the predicted values. Predicted values for data points other than those observed can be obtained by using missing dependent variables in your input data set.

Specifications that have a REPEATED statement with the SUBJECT= option and missing dependent variables compute predicted values by using empirical best linear unbiased prediction (EBLUP). Using hats ($\hat{}$) to denote estimates, the EBLUP formula is

$$\hat{\mathbf{m}} = \mathbf{X}_\mathbf{m}\hat{\boldsymbol{\beta}} + \hat{\mathbf{C}}_\mathbf{m}\hat{\mathbf{V}}^{-1}(\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}})$$

where **m** represents a hypothetical realization of a missing data vector with associated design matrix $\mathbf{X}_\mathbf{m}$. The matrix $\mathbf{C}_\mathbf{m}$ is the model-based covariance matrix between **m** and the observed data **y**, and other notation is as presented in the section "Mixed Models Theory" on page 3968.

The estimated prediction variance is as follows:

$$\widehat{\mathrm{Var}}(\hat{\mathbf{m}} - \mathbf{m}) = \hat{\mathbf{V}}_\mathbf{m} - \hat{\mathbf{C}}_\mathbf{m}\hat{\mathbf{V}}^{-1}\hat{\mathbf{C}}_\mathbf{m}^T + [\mathbf{X}_\mathbf{m} - \hat{\mathbf{C}}_\mathbf{m}\hat{\mathbf{V}}^{-1}\mathbf{X}](\mathbf{X}^T\hat{\mathbf{V}}^{-1}\mathbf{X})^-[\mathbf{X}_\mathbf{m} - \hat{\mathbf{C}}_\mathbf{m}\hat{\mathbf{V}}^{-1}\mathbf{X}]^T$$

where $\mathbf{V_m}$ is the model-based variance matrix of $\mathbf{m}$. For further details, see Henderson (1984) and Harville (1990). This feature can be useful for forecasting time series or for computing spatial predictions.

By default, all variables from the input data set are included in the OUTP= data set. You can select a subset of these variables by using the ID statement.

OUTPM=*SAS-data-set*

OUTPREDM=*SAS-data-set*

specifies an output data set containing predicted means and related quantities. This option replaces the PM option from SAS 6.

The output data set is of the same form as that resulting from the OUTP= option, except that the predicted values do not incorporate the EBLUP values $\mathbf{Z}\widehat{\boldsymbol{\gamma}}$. They also do not use the EBLUPs for specifications that have a REPEATED statement with the SUBJECT= option and missing dependent variables. The predicted values are formed as $\mathbf{X}\widehat{\boldsymbol{\beta}}$ in the OUTPM= data set, and standard errors are quadratic forms in the approximate variance-covariance matrix of $\widehat{\boldsymbol{\beta}}$ as displayed by the COVB option.

By default, all variables from the input data set are included in the OUTPM= data set. You can select a subset of these variables by using the ID statement.

RESIDUAL

requests that Pearson-type and (internally) studentized residuals be added to the OUTP= and OUTPM= data sets. Studentized residuals are raw residuals standardized by their estimated standard error. When residuals are internally studentized, the data point in question has contributed to the estimation of the covariance parameter estimates on which the standard error of the residual is based. Externally studentized residuals can be computed with the INFLUENCE option. Pearson-type residuals scale the residual by the standard deviation of the response.

The option has no effect unless the OUTP= or OUTPM= option is specified or unless you request statistical graphics with the ODS GRAPHICS statement. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the MIXED procedure, see the section "ODS Graphics" on page 4004. For computational details about studentized and Pearson residuals in MIXED, see the section "Residual Diagnostics" on page 3986.

SINGCHOL=*number*

tunes the sensitivity in computing Cholesky roots. If a diagonal pivot element is less than *D*number* as PROC MIXED performs the Cholesky decomposition on a matrix, the associated column is declared to be linearly dependent upon previous columns and is set to $\mathbf{0}$. The value D is the original diagonal element of the matrix. The default for *number* is 1E4 times the machine epsilon; this product is approximately 1E−12 on most computers.

SINGRES=*number*

sets the tolerance for which the residual variance is considered to be zero. The default is 1E4 times the machine epsilon; this product is approximately 1E−12 on most computers.

SINGULAR=*number*

tunes the sensitivity in sweeping. If a diagonal pivot element is less than *D*number* as PROC MIXED sweeps a matrix, the associated column is declared to be linearly dependent upon previous columns, and the associated parameter is set to 0. The value *D* is the original diagonal element of the matrix. The default is 1E4 times the machine epsilon; this product is approximately 1E−12 on most computers.

SOLUTION

S

requests that a solution for the fixed-effects parameters be produced. Using notation from the section "Mixed Models Theory" on page 3968, the fixed-effects parameter estimates are $\widehat{\boldsymbol{\beta}}$ and their approximate standard errors are the square roots of the diagonal elements of $(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}$. You can output this approximate variance matrix with the COVB option or modify it with the EMPIRICAL option in the PROC MIXED statement or the DDFM=KENWARDROGER option in the MODEL statement.

Along with the estimates and their approximate standard errors, a *t* statistic is computed as the estimate divided by its standard error. The degrees of freedom for this *t* statistic matches the one appearing in the "Tests of Fixed Effects" table under the effect containing the parameter. The "Pr > |t|" column contains the two-tailed *p*-value corresponding to the *t* statistic and associated degrees of freedom. You can use the CL option to request confidence intervals for all of the parameters; they are constructed around the estimate by using a radius of the standard error times a percentage point from the *t* distribution.

VCIRY

requests that responses and marginal residuals be scaled by the inverse Cholesky root of the marginal variance-covariance matrix. The variables ScaledDep and ScaledResid are added to the OUTPM= data set. These quantities can be important in bootstrapping of data or residuals. Examination of the scaled residuals is also helpful in diagnosing departures from normality. Notice that the results of this scaling operation can depend on the order in which the MIXED procedure processes the data.

The VCIRY option has no effect unless you also use the OUTPM= option or unless you request statistical graphics with the ODS GRAPHICS statement. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the MIXED procedure, see the section "ODS Graphics" on page 4004.

XPVIX

is an alias for the COVBI option.

XPVIXI

is an alias for the COVB option.

ZETA=*number*

tunes the sensitivity in forming Type 3 functions. Any element in the estimable function basis with an absolute value less than *number* is set to 0. The default is 1E−8.

PARMS Statement

PARMS (*value-list***) . . . <** */ options* **> ;**

The PARMS statement specifies initial values for the covariance parameters, or it requests a grid search over several values of these parameters. You must specify the values in the order in which they appear in the "Covariance Parameter Estimates" table.

The *value-list* specification can take any of several forms:

m	a single value
$m_1, m_2, \ldots, m_n$	several values
m to n	a sequence where m equals the starting value, n equals the ending value, and the increment equals 1
m to n by i	a sequence where m equals the starting value, n equals the ending value, and the increment equals i
m_1, m_2 to m_3	mixed values and sequences

You can use the PARMS statement to input known parameters. Referring to the split-plot example (Example 56.1), suppose the three variance components are known to be 60, 20, and 6. The SAS statements to fix the variance components at these values are as follows:

```
proc mixed data=sp noprofile;
   class Block A B;
   model Y = A B A*B;
   random Block A*Block;
   parms (60) (20) (6) / noiter;
run;
```

The NOPROFILE option requests PROC MIXED to refrain from profiling the residual variance parameter during its calculations, thereby enabling its value to be held at 6 as specified in the PARMS statement. The NOITER option prevents any Newton-Raphson iterations so that the subsequent results are based on the given variance components. You can also specify known parameters of **G** by using the GDATA= option in the RANDOM statement.

If you specify more than one set of initial values, PROC MIXED performs a grid search of the likelihood surface and uses the best point on the grid for subsequent analysis. Specifying a large number of grid points can result in long computing times. The grid search feature is also useful for exploring the likelihood surface. (See Example 56.3.)

The results from the PARMS statement are the values of the parameters on the specified grid (denoted by CovP1–CovP*n*), the residual variance (possibly estimated) for models with a residual variance parameter, and various functions of the likelihood.

For ODS purposes, the name of the "Parameter Search" table is "ParmSearch."

You can specify the following options in the PARMS statement after a slash (/).

HOLD=*value-list*

EQCONS=*value-list*

specifies which parameter values PROC MIXED should hold to equal the specified values. For example, the following statement constrains the first and third covariance parameters to equal 5 and 2, respectively:

```
parms (5) (3) (2) (3) / hold=1,3;
```

LOGDETH

evaluates the log determinant of the Hessian matrix for each point specified in the PARMS statement. A Log Det H column is added to the "Parameter Search" table.

LOWERB=*value-list*

enables you to specify lower boundary constraints on the covariance parameters. The *value-list* specification is a list of numbers or missing values (.) separated by commas. You must list the numbers in the order that PROC MIXED uses for the covariance parameters, and each number corresponds to the lower boundary constraint. A missing value instructs PROC MIXED to use its default constraint, and if you do not specify numbers for all of the covariance parameters, PROC MIXED assumes the remaining ones are missing.

An example for which this option is useful is when you want to constrain the **G** matrix to be positive definite in order to avoid the more computationally intensive algorithms required when **G** becomes singular. The corresponding statements for a random coefficients model are as follows:

```
proc mixed;
   class person;
   model y = time;
   random int time / type=fa0(2) sub=person;
   parms / lowerb=1e-4,.,1e-4;
run;
```

Here the TYPE=FA0(2) structure is used in order to specify a Cholesky root parameterization for the 2×2 unstructured blocks in **G**. This parameterization ensures that the **G** matrix is nonnegative definite, and the PARMS statement then ensures that it is positive definite by constraining the two diagonal terms to be greater than or equal to 1E−4.

NOBOUND

requests the removal of boundary constraints on covariance parameters. For example, variance components have a default lower boundary constraint of 0, and the NOBOUND option allows their estimates to be negative.

NOITER

requests that no Newton-Raphson iterations be performed and that PROC MIXED use the best value from the grid search to perform inferences. By default, iterations begin at the best value from the PARMS grid search.

NOPROFILE

specifies a different computational method for the residual variance during the grid search. By default, PROC MIXED estimates this parameter by using the profile likelihood when

appropriate. This estimate is displayed in the Variance column of the "Parameter Search" table. The NOPROFILE option suppresses the profiling and uses the actual value of the specified variance in the likelihood calculations.

OLS

requests starting values corresponding to the usual general linear model. Specifically, all variances and covariances are set to zero except for the residual variance, which is set equal to its ordinary least squares (OLS) estimate. This option is useful when the default MIVQUE0 procedure produces poor starting values for the optimization process.

PARMSDATA=*SAS-data-set*

PDATA=*SAS-data-set*

reads in covariance parameter values from a SAS data set. The data set should contain the Est or Covp1–Covp*n* variables.

RATIOS

indicates that ratios with the residual variance are specified instead of the covariance parameters themselves. The default is to use the individual covariance parameters.

UPPERB=*value-list*

enables you to specify upper boundary constraints on the covariance parameters. The *value-list* specification is a list of numbers or missing values (.) separated by commas. You must list the numbers in the order that PROC MIXED uses for the covariance parameters, and each number corresponds to the upper boundary constraint. A missing value instructs PROC MIXED to use its default constraint, and if you do not specify numbers for all of the covariance parameters, PROC MIXED assumes that the remaining ones are missing.

PRIOR Statement

PRIOR < *distribution* >< / *options* > **;**

The PRIOR statement enables you to carry out a sampling-based Bayesian analysis in PROC MIXED. It currently operates only with variance component models. The analysis produces a SAS data set containing a pseudo-random sample from the joint posterior density of the variance components and other parameters in the mixed model.

The posterior analysis is performed after all other PROC MIXED computations. It begins with the "Posterior Sampling Information" table, which provides basic information about the posterior sampling analysis, including the prior densities, sampling algorithm, sample size, and random number seed. For ODS purposes, the name of this table is "Posterior."

By default, PROC MIXED uses an independence chain algorithm in order to generate the posterior sample (Tierney 1994). This algorithm works by generating a pseudo-random proposal from a convenient base distribution, chosen to be as close as possible to the posterior. The proposal is then retained in the sample with probability proportional to the ratio of weights constructed by taking the ratio of the true posterior to the base density. If a proposal is not accepted, then a duplicate of the previous observation is added to the chain.

In selecting the base distribution, PROC MIXED makes use of the fact that the fixed-effects parameters can be analytically integrated out of the joint posterior, leaving the marginal posterior density of the variance components. In order to better approximate the marginal posterior density of the variance components, PROC MIXED transforms them by using the MIVQUE(0) equations. You can display the selected transformation with the PTRANS option or specify your own with the TDATA= option. The density of the transformed parameters is then approximated by a product of inverted gamma densities (see Gelfand et al. 1990).

To determine the parameters for the inverted gamma densities, PROC MIXED evaluates the logarithm of the posterior density over a grid of points in each of the transformed parameters, and you can display the results of this search with the PSEARCH option. PROC MIXED then performs a linear regression of these values on the logarithm of the inverted gamma density. The resulting base densities are displayed in the "Base Densities" table; for ODS purposes, the name of this table is "BaseDen." You can input different base densities with the BDATA= option.

At the end of the sampling, the "Acceptance Rates" table displays the acceptance rate computed as the number of accepted samples divided by the total number of samples generated. For ODS purposes, the name of the "Acceptance Rates" table is "AcceptanceRates."

The OUT= option specifies the output data set containing the posterior sample. PROC MIXED automatically includes all variance component parameters in this data set (labeled COVP1–COVP*n*), the Type 3 *F* statistics constructed as in Ghosh (1992) discussing Schervish (1992) (labeled T3F*n*), the log values of the posterior (labeled LOGF), the log of the base sampling density (labeled LOGG), and the log of their ratio (labeled LOGRATIO). If you specify the SOLUTION option in the MODEL statement, the data set also contains a random sample from the posterior density of the fixed-effects parameters (labeled BETA*n*); and if you specify the SOLUTION option in the RANDOM statement, the table contains a random sample from the posterior density of the random-effects parameters (labeled GAM*n*). PROC MIXED also generates additional variables corresponding to any CONTRAST, ESTIMATE, or LSMEANS statement that you specify.

Subsequently, you can use SAS/INSIGHT or the UNIVARIATE, CAPABILITY, or KDE procedure to analyze the posterior sample.

The prior density of the variance components is, by default, a noninformative version of Jeffreys' prior (Box and Tiao 1973). You can also specify informative priors with the DATA= option or a flat (equal to 1) prior for the variance components. The prior density of the fixed-effects parameters is assumed to be flat (equal to 1), and the resulting posterior is conditionally multivariate normal (conditioning on the variance component parameters) with mean $(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-}\mathbf{X}'\mathbf{V}^{-1}\mathbf{y}$ and variance $(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-}$.

The *distribution* argument in the PRIOR statement determines the prior density for the variance component parameters of your mixed model. Valid values are as follows.

DATA=

enables you to input the prior densities of the variance components used by the sampling algorithm. This data set must contain the Type and Parm1–Parm*n* variables, where *n* is the largest number of parameters among each of the base densities. The format of the DATA= data set matches that created by PROC MIXED in the "Base Densities" table, so you can output the densities from one run and use them as input for a subsequent run.

JEFFREYS
specifies a noninformative reference version of Jeffreys' prior constructed by using the square root of the determinant of the expected information matrix as in (1.3.92) of Box and Tiao (1973). This is the default prior.

FLAT
specifies a prior density equal to 1 everywhere, making the likelihood function the posterior.

You can specify the following options in the PRIOR statement after a slash (/).

ALG=IC | INDCHAIN
ALG=IS | IMPSAMP
ALG=RS | REJSAMP
ALG=RWC | RWCHAIN
specifies the algorithm used for generating the posterior sample. The ALG=IC option requests an independence chain algorithm, and it is the default. The option ALG=IS requests importance sampling, ALG=RS requests rejection sampling, and ALG=RWC requests a random walk chain. For more information about these techniques, see Ripley (1987), Smith and Gelfand (1992), and Tierney (1994).

BDATA=
enables you to input the base densities used by the sampling algorithm. This data set must contain the Type and Parm1–Parm$n$ variables, where n is the largest number of parameters among each of the base densities. The format of the BDATA= data set matches that created by PROC MIXED in the "Base Densities" table, so you can output the densities from one run and use them as input for a subsequent run.

GRID=(*value-list***)**
specifies a grid of values over which to evaluate the posterior density. The *value-list* syntax is the same as in the PARMS statement, and you must specify an output data set name with the OUTG= option.

GRIDT=(*value-list***)**
specifies a transformed grid of values over which to evaluate the posterior density. The *value-list* syntax is the same as in the PARMS statement, and you must specify an output data set name with the OUTGT= option.

IFACTOR=*number*
is an alias for the SFACTOR= option.

LOGNOTE=*number*
instructs PROC MIXED to write a note to the SAS log after it generates the sample corresponding to each multiple of *number*. This is useful for monitoring the progress of CPU-intensive runs.

LOGRBOUND=*number*
specifies the bounding constant for rejection sampling. The value of *number* equals the maximum of $\log\{f/g\}$ over the variance component parameter space, where f is the posterior density and g is the product inverted gamma densities used to perform rejection sampling.

When performing the rejection sampling, you might encounter the following message:

```
WARNING: The log ratio bound of LL was violated at sample XX.
```

When this occurs, PROC MIXED reruns an optimization algorithm to determine a new log upper bound and then restarts the rejection sampling. The resulting OUT= data set contains all observations that have been generated; therefore, assuming that you have requested N samples, you should retain only the final N observations in this data set for analysis purposes.

NSAMPLE=*number*
specifies the number of posterior samples to generate. The default is 1000, but more accurate results are obtained with larger samples such as 10000.

NSEARCH=*number*
specifies the number of posterior evaluations PROC MIXED makes for each transformed parameter in determining the parameters for the inverted gamma densities. The default is 20.

OUT=*SAS-data-set*
creates an output data set containing the sample from the posterior density.

OUTG=*SAS-data-set*
creates an output data set from the grid evaluations specified in the GRID= option.

OUTGT=*SAS-data-set*
creates an output data set from the transformed grid evaluations specified in the GRIDT= option.

PSEARCH
displays the search used to determine the parameters for the inverted gamma densities. For ODS purposes, the name of the table is "Search."

PTRANS
displays the transformation of the variance components. For ODS purposes, the name of the table is "Trans."

SEED=*number*
specifies an integer used to start the pseudo-random number generator for the simulation. If you do not specify a seed, or if you specify a value less than or equal to zero, the seed is by default generated from reading the time of day from the computer clock. You should use a positive seed (less than $2^{31} - 1$) whenever you want to duplicate the sample in another run of PROC MIXED.

SFACTOR=*number*
enables you to adjust the range over which PROC MIXED searches the transformed parameters in order to determine the parameters for the inverted gamma densities. PROC MIXED determines the range by first transforming the estimates from the standard PROC MIXED analysis (REML, ML, or MIVQUE0, depending upon which estimation method you select). It then multiplies and divides the transformed estimates by $2*number$ to obtain upper and lower bounds, respectively. Transformed values that produce negative variance components in the original scale are not included in the search. The default value is 1; *number* must be greater than 0.5.

TDATA=
enables you to input the transformation of the covariance parameters used by the sampling algorithm. This data set should contain the CovP1–CovP*n* variables. The format of the TDATA= data set matches that created by PROC MIXED in the "Trans" table, so you can output the transformation from one run and use it as input for a subsequent run.

TRANS=EXPECTED
TRANS=MIVQUE0
TRANS=OBSERVED
specifies the particular algorithm used to determine the transformation of the covariance parameters. The default is MIVQUE0, indicating a transformation based on the MIVQUE(0) equations. The other two options indicate the type of Hessian matrix used in constructing the transformation via a Cholesky root.

UPDATE=*number*
is an alias for the LOGNOTE= option.

RANDOM Statement

RANDOM *random-effects* < / *options* > **;**

The RANDOM statement defines the random effects constituting the $\boldsymbol{\gamma}$ vector in the mixed model. It can be used to specify traditional variance component models (as in the VARCOMP procedure) and to specify random coefficients. The random effects can be classification or continuous, and multiple RANDOM statements are possible.

Using notation from the section "Mixed Models Theory" on page 3968, the purpose of the RANDOM statement is to define the $\mathbf{Z}$ matrix of the mixed model, the random effects in the $\boldsymbol{\gamma}$ vector, and the structure of $\mathbf{G}$. The $\mathbf{Z}$ matrix is constructed exactly as the $\mathbf{X}$ matrix for the fixed effects, and the $\mathbf{G}$ matrix is constructed to correspond with the effects constituting $\mathbf{Z}$. The structure of $\mathbf{G}$ is defined by using the TYPE= option.

You can specify INTERCEPT (or INT) as a random effect to indicate the intercept. PROC MIXED does not include the intercept in the RANDOM statement by default as it does in the MODEL statement.

Table 56.11 summarizes important options in the RANDOM statement. All options are subsequently discussed in alphabetical order.

Table 56.11 Summary of Important RANDOM Statement Options

Option	Description
Construction of Covariance Structure	
GDATA=	requests that the $\mathbf{G}$ matrix be read from a SAS data set
GROUP=	varies covariance parameters by groups
LDATA=	specifies data set with coefficient matrices for TYPE= LIN
NOFULLZ	eliminates columns in $\mathbf{Z}$ corresponding to missing values

Table 56.11 *continued*

Option	Description
RATIOS	indicates that ratios are specified in the GDATA= data set
SUBJECT=	identifies the subjects in the model
TYPE=	specifies the covariance structure
Statistical Output	
ALPHA=α	determines the confidence level $(1-\alpha)$
CL	requests confidence limits for predictors of random effects
G	displays the estimated **G** matrix
GC	displays the Cholesky root (lower) of estimated **G** matrix
GCI	displays the inverse Cholesky root (lower) of estimated **G** matrix
GCORR	displays the correlation matrix corresponding to estimated **G** matrix
GI	displays the inverse of the estimated **G** matrix
SOLUTION	displays solutions $\widehat{\boldsymbol{\gamma}}$ of the G-side random effects
V	displays blocks of the estimated **V** matrix
VC	displays the lower-triangular Cholesky root of blocks of the estimated **V** matrix
VCI	displays the inverse Cholesky root of blocks of the estimated **V** matrix
VCORR	displays the correlation matrix corresponding to blocks of the estimated **V** matrix
VI	displays the inverse of the blocks of the estimated **V** matrix

You can specify the following options in the RANDOM statement after a slash (/).

ALPHA=*number*

requests that a *t*-type confidence interval be constructed for each of the random-effect estimates with confidence level 1−*number*. The value of *number* must be between 0 and 1; the default is 0.05.

CL

requests that *t*-type confidence limits be constructed for each of the random-effect estimates. The confidence level is 0.95 by default; this can be changed with the ALPHA= option.

G

requests that the estimated **G** matrix be displayed. PROC MIXED displays blanks for values that are 0. If you specify the SUBJECT= option, then the block of the **G** matrix corresponding to the first subject is displayed. For ODS purposes, the name of the table is "G."

GC

displays the lower-triangular Cholesky root of the estimated **G** matrix according to the rules listed under the G option. For ODS purposes, the name of the table is "CholG."

GCI

displays the inverse Cholesky root of the estimated **G** matrix according to the rules listed under the G option. For ODS purposes, the name of the table is "InvCholG."

GCORR

displays the correlation matrix corresponding to the estimated **G** matrix according to the rules listed under the G option. For ODS purposes, the name of the table is "GCorr."

GDATA=*SAS-data-set*

requests that the **G** matrix be read in from a SAS data set. This **G** matrix is assumed to be known; therefore, only **R**-side parameters from effects in the REPEATED statement are included in the Newton-Raphson iterations. If no REPEATED statement is specified, then only a residual variance is estimated.

The information in the GDATA= data set can appear in one of two ways. The first is a sparse representation for which you include Row, Col, and Value variables to indicate the row, column, and value of **G**, respectively. All unspecified locations are assumed to be 0. The second representation is for dense matrices. In it you include Row and Col1–Col*n* variables to indicate, respectively, the row and columns of **G**, which is a symmetric matrix of order *n*. For both representations, you must specify effects in the RANDOM statement that generate a **Z** matrix that contains *n* columns. (See Example 56.4.)

If you have more than one RANDOM statement, only one GDATA= option is required in any one of them, and the data set you specify must contain the entire **G** matrix defined by all of the RANDOM statements.

If the GDATA= data set contains variance ratios instead of the variances themselves, then use the RATIOS option.

Known parameters of **G** can also be input by using the PARMS statement with the HOLD= option.

GI

displays the inverse of the estimated **G** matrix according to the rules listed under the G option. For ODS purposes, the name of the table is "InvG."

GROUP=*effect*

GRP=*effect*

defines an effect specifying heterogeneity in the covariance structure of **G**. All observations having the same level of the group effect have the same covariance parameters. Each new level of the group effect produces a new set of covariance parameters with the same structure as the original group. You should exercise caution in defining the group effect, because strange covariance patterns can result from its misuse. Also, the group effect can greatly increase the number of estimated covariance parameters, which can adversely affect the optimization process.

Continuous variables are permitted as arguments to the GROUP= option. PROC MIXED does not sort by the values of the continuous variable; rather, it considers the data to be from a new subject or group whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of subjects or groups and also prevents the production of a large "Class Level Information" table.

LDATA=*SAS-data-set*

reads the coefficient matrices associated with the TYPE=LIN(*number*) option. The data set

must contain the variables Parm, Row, Col1–Col*n* or Parm, Row, Col, Value. The Parm variable denotes which of the *number* coefficient matrices is currently being constructed, and the Row, Col1–Col*n*, or Row, Col, Value variables specify the matrix values, as they do with the GDATA= option. Unspecified values of these matrices are set equal to 0.

NOFULLZ

eliminates the columns in $\mathbf{Z}$ corresponding to missing levels of random effects involving CLASS variables. By default, these columns are included in $\mathbf{Z}$.

RATIOS

indicates that ratios with the residual variance are specified in the GDATA= data set instead of the covariance parameters themselves. The default GDATA= data set contains the individual covariance parameters.

SOLUTION

S

requests that the solution for the random-effects parameters be produced. Using notation from the section "Mixed Models Theory" on page 3968, these estimates are the empirical best linear unbiased predictors (EBLUPs) $\widehat{\boldsymbol{\gamma}} = \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{V}}^{-1}(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})$. They can be useful for comparing the random effects from different experimental units and can also be treated as residuals in performing diagnostics for your mixed model.

The numbers displayed in the SE Pred column of the "Solution for Random Effects" table are not the standard errors of the $\widehat{\boldsymbol{\gamma}}$ displayed in the Estimate column; rather, they are the standard errors of predictions $\widehat{\boldsymbol{\gamma}}_i - \boldsymbol{\gamma}_i$, where $\widehat{\boldsymbol{\gamma}}_i$ is the *i*th EBLUP and $\boldsymbol{\gamma}_i$ is the *i*th random-effect parameter.

SUBJECT=*effect*

SUB=*effect*

identifies the subjects in your mixed model. Complete independence is assumed across subjects; thus, for the RANDOM statement, the SUBJECT= option produces a block-diagonal structure in $\mathbf{G}$ with identical blocks. The $\mathbf{Z}$ matrix is modified to accommodate this block diagonality. In fact, specifying a subject effect is equivalent to nesting all other effects in the RANDOM statement within the subject effect.

Continuous variables are permitted as arguments to the SUBJECT= option. PROC MIXED does not sort by the values of the continuous variable; rather, it considers the data to be from a new subject or group whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of subjects or groups and also prevents the production of a large "Class Level Information" table.

When you specify the SUBJECT= option and a classification random effect, computations are usually much quicker if the levels of the random effect are duplicated within each level of the SUBJECT= effect.

TYPE=*covariance-structure*

specifies the covariance structure of $\mathbf{G}$. Valid values for *covariance-structure* and their descriptions are listed in Table 56.13 and Table 56.14. Although a variety of structures are

available, most applications call for either TYPE=VC or TYPE=UN. The TYPE=VC (variance components) option is the default structure, and it models a different variance component for each random effect.

The TYPE=UN (unstructured) option is useful for correlated random coefficient models. For example, the following statement specifies a random intercept-slope model that has different variances for the intercept and slope and a covariance between them:

```
random intercept age / type=un subject=person;
```

You can also use TYPE=FA0(2) here to request a **G** estimate that is constrained to be nonnegative definite.

If you are constructing your own columns of **Z** with continuous variables, you can use the TYPE=TOEP(1) structure to group them together to have a common variance component. If you want to have different covariance structures in different parts of **G**, you must use multiple RANDOM statements with different TYPE= options.

V<**=***value-list*>
requests that blocks of the estimated **V** matrix be displayed. The first block determined by the SUBJECT= effect is the default displayed block. PROC MIXED displays entries that are 0 as blanks in the table.

You can optionally use the *value-list* specification, which indicates the subjects for which blocks of **V** are to be displayed. For example, the following statement displays block matrices for the first, third, and seventh persons:

```
random int time / type=un subject=person v=1,3,7;
```

The table name for ODS purposes is "V."

VC<**=***value-list*>
displays the Cholesky root of the blocks of the estimated **V** matrix. The *value-list* specification is the same as in the V= option. The table name for ODS purposes is "CholV."

VCI<**=***value-list*>
displays the inverse of the Cholesky root of the blocks of the estimated **V** matrix. The *value-list* specification is the same as in the V= option. The table name for ODS purposes is "InvCholV."

VCORR<**=***value-list*>
displays the correlation matrix corresponding to the blocks of the estimated **V** matrix. The *value-list* specification is the same as in the V= option. The table name for ODS purposes is "VCorr."

VI<**=***value-list*>
displays the inverse of the blocks of the estimated **V** matrix. The *value-list* specification is the same as in the V= option. The table name for ODS purposes is "InvV."

REPEATED Statement

REPEATED < *repeated-effect* >< / *options* > **;**

The REPEATED statement is used to specify the **R** matrix in the mixed model. Its syntax is different from that of the REPEATED statement in PROC GLM. If no REPEATED statement is specified, **R** is assumed to be equal to $\sigma^2\mathbf{I}$.

For many repeated measures models, no repeated effect is required in the REPEATED statement. Simply use the SUBJECT= option to define the blocks of **R** and the TYPE= option to define their covariance structure. In this case, the repeated measures data must be similarly ordered for each subject, and you must indicate all missing response variables with periods in the input data set unless they all fall at the end of a subject's repeated response profile. These requirements are necessary in order to inform PROC MIXED of the proper location of the observed repeated responses.

Specifying a repeated effect is useful when you do not want to indicate missing values with periods in the input data set. The repeated effect must contain only classification variables. Make sure that the levels of the repeated effect are different for each observation within a subject; otherwise, PROC MIXED constructs identical rows in **R** corresponding to the observations with the same level. This results in a singular **R** and an infinite likelihood.

Whether you specify a REPEATED effect or not, the rows of **R** for each subject are constructed in the order in which they appear in the input data set.

Table 56.12 summarizes important options in the REPEATED statement. All options are subsequently discussed in alphabetical order.

Table 56.12 Summary of Important REPEATED Statement Options

Option	Description
Construction of Covariance Structure	
GROUP=	defines an effect specifying heterogeneity in the R-side covariance structure
LDATA=	specifies data set with coefficient matrices for TYPE= LIN
LOCAL	requests that a diagonal matrix be added to **R**
LOCALW	specifies that only the local effects are weighted
NOLOCALW	specifies that only the nonlocal effects are weighted
SUBJECT=	identifies the subjects in the R-side model
TYPE=	specifies the R-side covariance structure
Statistical Output	
HLM	produces a table of Hotelling-Lawley-McKeon statistics (McKeon 1974)
HLPS	produces a table of Hotelling-Lawley-Pillai-Samson statistics (Pillai and Samson 1959)
R	displays blocks of the estimated **R** matrix
RC	display the Cholesky root (lower) of blocks of the estimated **R** matrix

Table 56.12 *continued*

Option	Description
RCI	displays the inverse Cholesky root (lower) of blocks of the estimated **R** matrix
RCORR	displays the correlation matrix corresponding to blocks of the estimated **R** matrix
RI	displays the inverse of blocks of the estimated **R** matrix

You can specify the following options in the REPEATED statement after a slash (/).

GROUP=*effect*

GRP=*effect*

defines an effect specifying heterogeneity in the covariance structure of **R**. All observations having the same level of the GROUP effect have the same covariance parameters. Each new level of the GROUP effect produces a new set of covariance parameters with the same structure as the original group. You should exercise caution in properly defining the GROUP effect, because strange covariance patterns can result with its misuse. Also, the GROUP effect can greatly increase the number of estimated covariance parameters, which can adversely affect the optimization process.

Continuous variables are permitted as arguments to the GROUP= option. PROC MIXED does not sort by the values of the continuous variable; rather, it considers the data to be from a new subject or group whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of subjects or groups and also prevents the production of a large "Class Level Information" table.

HLM

produces a table of Hotelling-Lawley-McKeon statistics (McKeon 1974) for all fixed effects whose levels change across data having the same level of the SUBJECT= effect (the *within-subject* fixed effects). This option applies only when you specify a REPEATED statement with the TYPE=UN option and no RANDOM statements. For balanced data, this model is equivalent to the multivariate model for repeated measures in PROC GLM.

The Hotelling-Lawley-McKeon statistic has a slightly better F approximation than the Hotelling-Lawley-Pillai-Samson statistic (see the description of the HLPS option, which follows). Both of the Hotelling-Lawley statistics can perform much better in small samples than the default F statistic (Wright 1994).

Separate tables are produced for Type 1, 2, and 3 tests, according to the ones you select. For ODS purposes, the table names are "HLM1," "HLM2," and "HLM3," respectively.

HLPS

produces a table of Hotelling-Lawley-Pillai-Samson statistics (Pillai and Samson 1959) for all fixed effects whose levels change across data having the same level of the SUBJECT= effect (the *within-subject* fixed effects). This option applies only when you specify a REPEATED statement with the TYPE=UN option and no RANDOM statements. For balanced data, this

model is equivalent to the multivariate model for repeated measures in PROC GLM, and this statistic is the same as the Hotelling-Lawley Trace statistic produced by PROC GLM.

Separate tables are produced for Type 1, 2, and 3 tests, according to the ones you select. For ODS purposes, the table names are "HLPS1," "HLPS2," and "HLPS3," respectively.

LDATA=*SAS-data-set*

reads the coefficient matrices associated with the TYPE=LIN(*number*) option. The data set must contain the variables Parm, Row, Col1–Col*n* or Parm, Row, Col, Value. The Parm variable denotes which of the *number* coefficient matrices is currently being constructed, and the Row, Col1–Col*n*, or Row, Col, Value variables specify the matrix values, as they do with the RANDOM statement option GDATA=. Unspecified values of these matrices are set equal to 0.

LOCAL

LOCAL=POM(*POM-data-set***)**

requests that a diagonal matrix be added to $\mathbf{R}$. With just the LOCAL option, this diagonal matrix equals $\sigma^2\mathbf{I}$, and σ^2 becomes an additional variance parameter that PROC MIXED profiles out of the likelihood provided that you do not specify the NOPROFILE option in the PROC MIXED statement. The LOCAL option is useful if you want to add an observational error to a time series structure (Jones and Boadi-Boateng 1991) or a nugget effect to a spatial structure (Cressie 1993).

The LOCAL=EXP(<*effects*>) option produces exponential local effects, also known as dispersion effects, in a log-linear variance model. These local effects have the form

$$\sigma^2\text{diag}[\exp(\mathbf{U}\boldsymbol{\delta})]$$

where $\mathbf{U}$ is the full-rank design matrix corresponding to the effects that you specify and $\boldsymbol{\delta}$ are the parameters that PROC MIXED estimates. An intercept is not included in $\mathbf{U}$ because it is accounted for by σ^2. PROC MIXED constructs the full-rank $\mathbf{U}$ in terms of 1s and -1s for classification effects. Be sure to scale continuous effects in $\mathbf{U}$ sensibly.

The LOCAL=POM(*POM-data-set*) option specifies the power-of-the-mean structure. This structure possesses a variance of the form $\sigma^2|\mathbf{x}_i'\boldsymbol{\beta}^*|^\theta$ for the ith observation, where $\mathbf{x}_i$ is the ith row of $\mathbf{X}$ (the design matrix of the fixed effects) and $\boldsymbol{\beta}^*$ is an estimate of the fixed-effects parameters that you specify in *POM-data-set*.

The SAS data set specified by *POM-data-set* contains the numeric variable Estimate (in previous releases, the variable name was required to be EST), and it has at least as many observations as there are fixed-effects parameters. The first p observations of the Estimate variable in *POM-data-set* are taken to be the elements of $\boldsymbol{\beta}^*$, where p is the number of columns of $\mathbf{X}$. You must order these observations according to the non-full-rank parameterization of the MIXED procedure. One easy way to set up *POM-data-set* for a $\boldsymbol{\beta}^*$ corresponding to ordinary least squares is illustrated by the following statements:

```
ods output SolutionF=sf;
proc mixed;
   class a;
   model y = a x / s;
run;
```

```
proc mixed;
   class a;
   model y = a x;
   repeated / local=pom(sf);
run;
```

Note that the generalized least squares estimate of the fixed-effects parameters from the second PROC MIXED step usually is not the same as your specified $\boldsymbol{\beta}^*$. However, you can iterate the POM fitting until the two estimates agree. Continuing from the previous example, the statements for performing one step of this iteration are as follows:

```
ods output SolutionF=sf1;
proc mixed;
   class a;
   model y = a x / s;
   repeated / local=pom(sf);
run;

proc compare brief data=sf compare=sf1;
   var estimate;
run;

data sf;
   set sf1;
run;
```

Unfortunately, this iterative process does not always converge. For further details, refer to the description of pseudo-likelihood in Chapter 3 of Carroll and Ruppert (1988).

LOCALW

specifies that only the local effects and no others be weighted. By default, all effects are weighted. The LOCALW option is used in connection with the WEIGHT statement and the LOCAL option in the REPEATED statement.

NONLOCALW

specifies that only the nonlocal effects and no others be weighted. By default, all effects are weighted. The NONLOCALW option is used in connection with the WEIGHT statement and the LOCAL option in the REPEATED statement.

R<**=***value-list*>

requests that blocks of the estimated **R** matrix be displayed. The first block determined by the SUBJECT= effect is the default displayed block. PROC MIXED displays blanks for value-lists that are 0.

The *value-list* indicates the subjects for which blocks of **R** are to be displayed. For example, the following statement displays block matrices for the first, third, and fifth persons:

```
repeated / type=cs subject=person r=1,3,5;
```

See the PARMS statement for the possible forms of *value-list*. The table name for ODS purposes is "R."

RC<**=***value-list*>
produces the Cholesky root of blocks of the estimated **R** matrix. The *value-list* specification is the same as with the R option. The table name for ODS purposes is "CholR."

RCI<**=***value-list*>
produces the inverse Cholesky root of blocks of the estimated **R** matrix. The *value-list* specification is the same as with the R option. The table name for ODS purposes is "InvCholR."

RCORR<**=***value-list*>
produces the correlation matrix corresponding to blocks of the estimated **R** matrix. The *value-list* specification is the same as with the R option. The table name for ODS purposes is "RCorr."

RI<**=***value-list*>
produces the inverse of blocks of the estimated **R** matrix. The *value-list* specification is the same as with the R option. The table name for ODS purposes is "InvR."

SSCP
requests that an unstructured **R** matrix be estimated from the sum-of-squares-and-crossproducts matrix of the residuals. It applies only when you specify TYPE=UN and have no RANDOM statements. Also, you must have a sufficient number of subjects for the estimate to be positive definite.

This option is useful when the size of the blocks of **R** is large (for example, greater than 10) and you want to use or inspect an unstructured estimate that is much quicker to compute than the default REML estimate. The two estimates will agree for certain balanced data sets when you have a classification fixed effect defined across all time points within a subject.

SUBJECT=*effect*
SUB=*effect*
identifies the subjects in your mixed model. Complete independence is assumed across subjects; therefore, the SUBJECT= option produces a block-diagonal structure in **R** with identical blocks. When the SUBJECT= effect consists entirely of classification variables, the blocks of **R** correspond to observations sharing the same level of that effect. These blocks are sorted according to this effect as well.

Continuous variables are permitted as arguments to the SUBJECT= option. PROC MIXED does not sort by the values of the continuous variable; rather, it considers the data to be from a new subject or group whenever the value of the continuous variable changes from the previous observation. Using a continuous variable decreases execution time for models with a large number of subjects or groups and also prevents the production of a large "Class Level Information" table.

If you want to model nonzero covariance among all of the observations in your SAS data set, specify SUBJECT=INTERCEPT to treat the data as if they are all from one subject. However, be aware that in this case PROC MIXED manipulates an **R** matrix with dimensions equal to the number of observations. If no SUBJECT= effect is specified, then every observation is assumed to be from a different subject and **R** is assumed to be diagonal. For this reason, you usually want to use the SUBJECT= option in the REPEATED statement.

TYPE=*covariance-structure*

specifies the covariance structure of the **R** matrix. The SUBJECT= option defines the blocks of **R**, and the TYPE= option specifies the structure of these blocks. Valid values for *covariance-structure* and their descriptions are provided in Table 56.13 and Table 56.14. The default structure is VC.

Table 56.13 Covariance Structures

Structure	Description	Parms	(i, j)th element
ANTE(1)	Ante-dependence	$2t-1$	$\sigma_i\sigma_j\prod_{k=i}^{j-1}\rho_k$
AR(1)	Autoregressive(1)	2	$\sigma^2\rho^{\lvert i-j\rvert}$
ARH(1)	Heterogeneous AR(1)	$t+1$	$\sigma_i\sigma_j\rho^{\lvert i-j\rvert}$
ARMA(1,1)	ARMA(1,1)	3	$\sigma^2[\gamma\rho^{\lvert i-j\rvert-1}1(i\neq j)+1(i=j)]$
CS	Compound Symmetry	2	$\sigma_1+\sigma^2 1(i=j)$
CSH	Heterogeneous CS	$t+1$	$\sigma_i\sigma_j[\rho 1(i\neq j)+1(i=j)]$
FA(q)	Factor Analytic	$\frac{q}{2}(2t-q+1)+t$	$\Sigma_{k=1}^{\min(i,j,q)}\lambda_{ik}\lambda_{jk}+\sigma_i^2 1(i=j)$
FA0(q)	No Diagonal FA	$\frac{q}{2}(2t-q+1)$	$\Sigma_{k=1}^{\min(i,j,q)}\lambda_{ik}\lambda_{jk}$
FA1(q)	Equal Diagonal FA	$\frac{q}{2}(2t-q+1)+1$	$\Sigma_{k=1}^{\min(i,j,q)}\lambda_{ik}\lambda_{jk}+\sigma^2 1(i=j)$
HF	Huynh-Feldt	$t+1$	$(\sigma_i^2+\sigma_j^2)/2+\lambda 1(i\neq j)$
LIN(q)	General Linear	q	$\Sigma_{k=1}^{q}\theta_k\mathbf{A}_{ij}$
TOEP	Toeplitz	t	$\sigma_{\lvert i-j\rvert+1}$
TOEP(q)	Banded Toeplitz	q	$\sigma_{\lvert i-j\rvert+1}1(\lvert i-j\rvert<q)$
TOEPH	Heterogeneous TOEP	$2t-1$	$\sigma_i\sigma_j\rho_{\lvert i-j\rvert}$
TOEPH(q)	Banded Hetero TOEP	$t+q-1$	$\sigma_i\sigma_j\rho_{\lvert i-j\rvert}1(\lvert i-j\rvert<q)$
UN	Unstructured	$t(t+1)/2$	σ_{ij}
UN(q)	Banded	$\frac{q}{2}(2t-q+1)$	$\sigma_{ij}1(\lvert i-j\rvert<q)$
UNR	Unstructured Corrs	$t(t+1)/2$	$\sigma_i\sigma_j\rho_{\max(i,j)\min(i,j)}$
UNR(q)	Banded Correlations	$\frac{q}{2}(2t-q+1)$	$\sigma_i\sigma_j\rho_{\max(i,j)\min(i,j)}$
UN@AR(1)	Direct Product AR(1)	$t_1(t_1+1)/2+1$	$\sigma_{i_1j_1}\rho^{\lvert i_2-j_2\rvert}$
UN@CS	Direct Product CS	$t_1(t_1+1)/2+1$	$\begin{cases}\sigma_{i_1j_1} & i_2=j_2\\ \sigma^2\sigma_{i_1j_1} & i_2\neq j_2\\ 0\le\sigma^2\le 1\end{cases}$
UN@UN	Direct Product UN	$t_1(t_1+1)/2+$ $t_2(t_2+1)/2-1$	$\sigma_{1,i_1j_1}\sigma_{2,i_2j_2}$
VC	Variance Components	q	$\sigma_k^2 1(i=j)$ and i corresponds to kth effect

In Table 56.13, "Parms" is the number of covariance parameters in the structure, t is the overall dimension of the covariance matrix, and $1(A)$ equals 1 when A is true and 0 otherwise. For example, $1(i=j)$ equals 1 when $i=j$ and 0 otherwise, and $1(|i-j|<q)$ equals 1 when $|i-j|<q$ and 0 otherwise. For the TOEPH structures, $\rho_0=1$, and for the UNR

structures, $\rho_{ii} = 1$ for all i. For the direct product structures, the subscripts "1" and "2" refer to the first and second structure in the direct product, respectively, and $i_1 = \text{int}((i + t_2 - 1)/t_2)$, $j_1 = \text{int}((j + t_2 - 1)/t_2)$, $i_2 = \text{mod}(i - 1, t_2) + 1$, and $j_2 = \text{mod}(j - 1, t_2) + 1$.

Table 56.14 Spatial Covariance Structures

Structure	Description	Parms	(i, j)th element
SP(EXP)(*c-list*)	Exponential	2	$\sigma^2 \exp\{-d_{ij}/\theta\}$
SP(EXPA)(*c-list*)	Anisotropic Exponential	$2c + 1$	$\sigma^2 \prod_{k=1}^{c} \exp\{-\theta_k d(i, j, k)^{p_k}\}$
SP(EXPGA)(c_1 c_2)	2D Exponential, Geometrically Anisotropic	4	$\sigma^2 \exp\{-d_{ij}(\theta, \lambda)/\rho\}$
SP(GAU)(*c-list*)	Gaussian	2	$\sigma^2 \exp\{-d_{ij}^2/\rho^2\}$
SP(GAUGA)(c_1 c_2)	2D Gaussian, Geometrically Anisotropic	4	$\sigma^2 \exp\{-d_{ij}(\theta, \lambda)^2/\rho^2\}$
SP(LIN)(*c-list*)	Linear	2	$\sigma^2(1 - \rho d_{ij})\ 1(\rho d_{ij} \leq 1)$
SP(LINL)(*c-list*)	Linear Log	2	$\sigma^2(1 - \rho \log(d_{ij}))$ $\times 1(\rho \log(d_{ij}) \leq 1)$
SP(MATERN)(*c-list*)	Matérn	3	$\sigma^2 \frac{1}{\Gamma(\nu)} \left(\frac{d_{ij}}{2\rho}\right)^{\nu} 2K_\nu(d_{ij}/\rho)$
SP(MATHSW)(*c-list*)	Matérn (Handcock-Stein-Wallis)	3	$\sigma^2 \frac{1}{\Gamma(\nu)} \left(\frac{d_{ij}\sqrt{\nu}}{\rho}\right)^{\nu} 2K_\nu \left(\frac{2d_{ij}\sqrt{\nu}}{\rho}\right)$
SP(POW)(*c-list*)	Power	2	$\sigma^2 \rho^{d_{ij}}$
SP(POWA)(*c-list*)	Anisotropic Power	$c + 1$	$\sigma^2 \rho_1^{d(i,j,1)} \rho_2^{d(i,j,2)} \ldots \rho_c^{d(i,j,c)}$
SP(SPH)(*c-list*)	Spherical	2	$\sigma^2[1 - (\frac{3d_{ij}}{2\rho}) + (\frac{d_{ij}^3}{2\rho^3})]\ 1(d_{ij} \leq \rho)$
SP(SPHGA)(c_1 c_2)	2D Spherical, Geometrically Anisotropic	4	$\sigma^2[1 - (\frac{3d_{ij}(\theta,\lambda)}{2\rho}) + (\frac{d_{ij}(\theta,\lambda)^3}{2\rho^3})]$ $\times 1(d_{ij}(\theta, \lambda) \leq \rho)$

In Table 56.14, *c-list* contains the names of the numeric variables used as coordinates of the location of the observation in space, and d_{ij} is the Euclidean distance between the *i*th and *j*th vectors of these coordinates, which correspond to the *i*th and *j*th observations in the input data set. For SP(POWA) and SP(EXPA), c is the number of coordinates, and $d(i, j, k)$ is the absolute distance between the *k*th coordinate, $k = 1, \ldots, c$, of the *i*th and *j*th observations in the input data set. For the geometrically anisotropic structures SP(EXPGA), SP(GAUGA), and SP(SPHGA), exactly two spatial coordinate variables must be specified as c_1 and c_2. Geometric anisotropy is corrected by applying a rotation θ and scaling λ to the coordinate system, and $d_{ij}(\theta, \lambda)$ represents the Euclidean distance between two points in the transformed space. SP(MATERN) and SP(MATHSW) represent covariance structures in a class defined by Matérn (see Matérn 1986, Handcock and Stein 1993, Handcock and Wallis 1994). The function K_ν is the modified Bessel function of the second kind of (real) order $\nu > 0$; the parameter ν governs the smoothness of the process (see below for more details).

Table 56.15 lists some examples of the structures in Table 56.13 and Table 56.14.

Table 56.15 Covariance Structure Examples

Description	Structure	Example
Variance Components	VC (default)	$\begin{bmatrix} \sigma_B^2 & 0 & 0 & 0 \\ 0 & \sigma_B^2 & 0 & 0 \\ 0 & 0 & \sigma_{AB}^2 & 0 \\ 0 & 0 & 0 & \sigma_{AB}^2 \end{bmatrix}$
Compound Symmetry	CS	$\begin{bmatrix} \sigma^2+\sigma_1 & \sigma_1 & \sigma_1 & \sigma_1 \\ \sigma_1 & \sigma^2+\sigma_1 & \sigma_1 & \sigma_1 \\ \sigma_1 & \sigma_1 & \sigma^2+\sigma_1 & \sigma_1 \\ \sigma_1 & \sigma_1 & \sigma_1 & \sigma^2+\sigma_1 \end{bmatrix}$
Unstructured	UN	$\begin{bmatrix} \sigma_1^2 & \sigma_{21} & \sigma_{31} & \sigma_{41} \\ \sigma_{21} & \sigma_2^2 & \sigma_{32} & \sigma_{42} \\ \sigma_{31} & \sigma_{32} & \sigma_3^2 & \sigma_{43} \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_4^2 \end{bmatrix}$
Banded Main Diagonal	UN(1)	$\begin{bmatrix} \sigma_1^2 & 0 & 0 & 0 \\ 0 & \sigma_2^2 & 0 & 0 \\ 0 & 0 & \sigma_3^2 & 0 \\ 0 & 0 & 0 & \sigma_4^2 \end{bmatrix}$
First-Order Autoregressive	AR(1)	$\sigma^2 \begin{bmatrix} 1 & \rho & \rho^2 & \rho^3 \\ \rho & 1 & \rho & \rho^2 \\ \rho^2 & \rho & 1 & \rho \\ \rho^3 & \rho^2 & \rho & 1 \end{bmatrix}$
Toeplitz	TOEP	$\begin{bmatrix} \sigma^2 & \sigma_1 & \sigma_2 & \sigma_3 \\ \sigma_1 & \sigma^2 & \sigma_1 & \sigma_2 \\ \sigma_2 & \sigma_1 & \sigma^2 & \sigma_1 \\ \sigma_3 & \sigma_2 & \sigma_1 & \sigma^2 \end{bmatrix}$
Toeplitz with Two Bands	TOEP(2)	$\begin{bmatrix} \sigma^2 & \sigma_1 & 0 & 0 \\ \sigma_1 & \sigma^2 & \sigma_1 & 0 \\ 0 & \sigma_1 & \sigma^2 & \sigma_1 \\ 0 & 0 & \sigma_1 & \sigma^2 \end{bmatrix}$
Spatial Power	SP(POW)(c)	$\sigma^2 \begin{bmatrix} 1 & \rho^{d_{12}} & \rho^{d_{13}} & \rho^{d_{14}} \\ \rho^{d_{21}} & 1 & \rho^{d_{23}} & \rho^{d_{24}} \\ \rho^{d_{31}} & \rho^{d_{32}} & 1 & \rho^{d_{34}} \\ \rho^{d_{41}} & \rho^{d_{42}} & \rho^{d_{43}} & 1 \end{bmatrix}$
Heterogeneous AR(1)	ARH(1)	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho & \sigma_1\sigma_3\rho^2 & \sigma_1\sigma_4\rho^3 \\ \sigma_2\sigma_1\rho & \sigma_2^2 & \sigma_2\sigma_3\rho & \sigma_2\sigma_4\rho^2 \\ \sigma_3\sigma_1\rho^2 & \sigma_3\sigma_2\rho & \sigma_3^2 & \sigma_3\sigma_4\rho \\ \sigma_4\sigma_1\rho^3 & \sigma_4\sigma_2\rho & \sigma_4\sigma_3\rho & \sigma_4^2 \end{bmatrix}$
First-Order Autoregressive Moving-Average	ARMA(1,1)	$\sigma^2 \begin{bmatrix} 1 & \gamma & \gamma\rho & \gamma\rho^2 \\ \gamma & 1 & \gamma & \gamma\rho \\ \gamma\rho & \gamma & 1 & \gamma \\ \gamma\rho^2 & \gamma\rho & \gamma & 1 \end{bmatrix}$

Table 56.15 *continued*

Description	Structure	Example
Heterogeneous CS	CSH	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho & \sigma_1\sigma_3\rho & \sigma_1\sigma_4\rho \\ \sigma_2\sigma_1\rho & \sigma_2^2 & \sigma_2\sigma_3\rho & \sigma_2\sigma_4\rho \\ \sigma_3\sigma_1\rho & \sigma_3\sigma_2\rho & \sigma_3^2 & \sigma_3\sigma_4\rho \\ \sigma_4\sigma_1\rho & \sigma_4\sigma_2\rho & \sigma_4\sigma_3\rho & \sigma_4^2 \end{bmatrix}$
First-Order Factor Analytic	FA(1)	$\begin{bmatrix} \lambda_1^2+d_1 & \lambda_1\lambda_2 & \lambda_1\lambda_3 & \lambda_1\lambda_4 \\ \lambda_2\lambda_1 & \lambda_2^2+d_2 & \lambda_2\lambda_3 & \lambda_2\lambda_4 \\ \lambda_3\lambda_1 & \lambda_3\lambda_2 & \lambda_3^2+d_3 & \lambda_3\lambda_4 \\ \lambda_4\lambda_1 & \lambda_4\lambda_2 & \lambda_4\lambda_3 & \lambda_4^2+d_4 \end{bmatrix}$
Huynh-Feldt	HF	$\begin{bmatrix} \sigma_1^2 & \frac{\sigma_1^2+\sigma_2^2}{2}-\lambda & \frac{\sigma_1^2+\sigma_3^2}{2}-\lambda \\ \frac{\sigma_2^2+\sigma_1^2}{2}-\lambda & \sigma_2^2 & \frac{\sigma_2^2+\sigma_3^2}{2}-\lambda \\ \frac{\sigma_3^2+\sigma_1^2}{2}-\lambda & \frac{\sigma_3^2+\sigma_2^2}{2}-\lambda & \sigma_3^2 \end{bmatrix}$
First-Order Ante-dependence	ANTE(1)	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_1 & \sigma_1\sigma_3\rho_1\rho_2 \\ \sigma_2\sigma_1\rho_1 & \sigma_2^2 & \sigma_2\sigma_3\rho_2 \\ \sigma_3\sigma_1\rho_2\rho_1 & \sigma_3\sigma_2\rho_2 & \sigma_3^2 \end{bmatrix}$
Heterogeneous Toeplitz	TOEPH	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_1 & \sigma_1\sigma_3\rho_2 & \sigma_1\sigma_4\rho_3 \\ \sigma_2\sigma_1\rho_1 & \sigma_2^2 & \sigma_2\sigma_3\rho_1 & \sigma_2\sigma_4\rho_2 \\ \sigma_3\sigma_1\rho_2 & \sigma_3\sigma_2\rho_1 & \sigma_3^2 & \sigma_3\sigma_4\rho_1 \\ \sigma_4\sigma_1\rho_3 & \sigma_4\sigma_2\rho_2 & \sigma_4\sigma_3\rho_1 & \sigma_4^2 \end{bmatrix}$
Unstructured Correlations	UNR	$\begin{bmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_{21} & \sigma_1\sigma_3\rho_{31} & \sigma_1\sigma_4\rho_{41} \\ \sigma_2\sigma_1\rho_{21} & \sigma_2^2 & \sigma_2\sigma_3\rho_{32} & \sigma_2\sigma_4\rho_{42} \\ \sigma_3\sigma_1\rho_{31} & \sigma_3\sigma_2\rho_{32} & \sigma_3^2 & \sigma_3\sigma_4\rho_{43} \\ \sigma_4\sigma_1\rho_{41} & \sigma_4\sigma_2\rho_{42} & \sigma_4\sigma_3\rho_{43} & \sigma_4^2 \end{bmatrix}$
Direct Product AR(1)	UN@AR(1)	$\begin{bmatrix} \sigma_1^2 & \sigma_{21} \\ \sigma_{21} & \sigma_2^2 \end{bmatrix} \otimes \begin{bmatrix} 1 & \rho & \rho^2 \\ \rho & 1 & \rho \\ \rho^2 & \rho & 1 \end{bmatrix} =$ $\begin{bmatrix} \sigma_1^2 & \sigma_1^2\rho & \sigma_1^2\rho^2 & \sigma_{21} & \sigma_{21}\rho & \sigma_{21}\rho^2 \\ \sigma_1^2\rho & \sigma_1^2 & \sigma_1^2\rho & \sigma_{21}\rho & \sigma_{21} & \sigma_{21}\rho \\ \sigma_1^2\rho^2 & \sigma_1^2\rho & \sigma_1^2 & \sigma_{21}\rho^2 & \sigma_{21}\rho & \sigma_{21} \\ \sigma_{21} & \sigma_{21}\rho & \sigma_{21}\rho^2 & \sigma_2^2 & \sigma_2^2\rho & \sigma_2^2\rho^2 \\ \sigma_{21}\rho & \sigma_{21} & \sigma_{21}\rho & \sigma_2^2\rho & \sigma_2^2 & \sigma_2^2\rho \\ \sigma_{21}\rho^2 & \sigma_{21}\rho & \sigma_{21} & \sigma_2^2\rho^2 & \sigma_2^2\rho & \sigma_2^2 \end{bmatrix}$

The following provides some further information about these covariance structures:

TYPE=ANTE(1) specifies the first-order antedependence structure (see Kenward 1987, Patel 1991, and Macchiavelli and Arnold 1994). In Table 56.13, σ_i^2 is the ith variance parameter, and ρ_k is the kth autocorrelation parameter satisfying $|\rho_k| < 1$.

TYPE=AR(1) specifies a first-order autoregressive structure. PROC MIXED imposes the constraint $|\rho| < 1$ for stationarity.

TYPE=ARH(1) specifies a heterogeneous first-order autoregressive structure. As with TYPE=AR(1), PROC MIXED imposes the constraint $|\rho| < 1$ for stationarity.

TYPE=ARMA(1,1) specifies the first-order autoregressive moving-average structure. In Table 56.13, ρ is the autoregressive parameter, γ models a moving-average component, and σ^2 is the residual variance. In the notation of Fuller (1976, p. 68), $\rho = \theta_1$ and

$$\gamma = \frac{(1 + b_1\theta_1)(\theta_1 + b_1)}{1 + b_1^2 + 2b_1\theta_1}$$

The example in Table 56.15 and $|b_1| < 1$ imply that

$$b_1 = \frac{\beta - \sqrt{\beta^2 - 4\alpha^2}}{2\alpha}$$

where $\alpha = \gamma - \rho$ and $\beta = 1 + \rho^2 - 2\gamma\rho$. PROC MIXED imposes the constraints $|\rho| < 1$ and $|\gamma| < 1$ for stationarity, although for some values of ρ and γ in this region the resulting covariance matrix is not positive definite. When the estimated value of ρ becomes negative, the computed covariance is multiplied by $\cos(\pi d_{ij})$ to account for the negativity.

TYPE=CS specifies the compound-symmetry structure, which has constant variance and constant covariance.

TYPE=CSH specifies the heterogeneous compound-symmetry structure. This structure has a different variance parameter for each diagonal element, and it uses the square roots of these parameters in the off-diagonal entries. In Table 56.13, σ_i^2 is the ith variance parameter, and ρ is the correlation parameter satisfying $|\rho| < 1$.

TYPE=FA(q) specifies the factor-analytic structure with q factors (Jennrich and Schluchter 1986). This structure is of the form $\mathbf{\Lambda}\mathbf{\Lambda}' + \mathbf{D}$, where $\mathbf{\Lambda}$ is a $t \times q$ rectangular matrix and $\mathbf{D}$ is a $t \times t$ diagonal matrix with t different parameters. When $q > 1$, the elements of $\mathbf{\Lambda}$ in its upper-right corner (that is, the elements in the ith row and jth column for $j > i$) are set to zero to fix the rotation of the structure.

TYPE=FA0(q) is similar to the FA(q) structure except that no diagonal matrix $\mathbf{D}$ is included. When $q < t$—that is, when the number of factors is less than the dimension of the matrix—this structure is nonnegative definite but not of full rank. In this situation, you can use it for approximating an unstructured $\mathbf{G}$ matrix in the RANDOM statement or for combining with the LOCAL option in the REPEATED statement. When $q = t$, you can use this structure to constrain $\mathbf{G}$ to be nonnegative definite in the RANDOM statement.

TYPE=FA1(q) is similar to the FA(q) structure except that all of the elements in $\mathbf{D}$ are constrained to be equal. This offers a useful and more parsimonious alternative to the full factor-analytic structure.

TYPE=HF specifies the Huynh-Feldt covariance structure (Huynh and Feldt 1970). This structure is similar to the CSH structure in that it has the same number of parameters and heterogeneity along the main diagonal. However, it constructs the off-diagonal elements by taking arithmetic rather than geometric means.

You can perform a likelihood ratio test of the Huynh-Feldt conditions by running PROC MIXED twice, once with TYPE=HF and once with TYPE=UN, and then subtracting their respective values of -2 times the maximized likelihood.

If PROC MIXED does not converge under your Huynh-Feldt model, you can specify your own starting values with the PARMS statement. The default MIVQUE(0) starting values can sometimes be poor for this structure. A good choice for starting values is often the parameter estimates corresponding to an initial fit that uses TYPE=CS.

TYPE=LIN(q) specifies the general linear covariance structure with q parameters. This structure consists of a linear combination of known matrices that are input with the LDATA= option. This structure is very general, and you need to make sure that the variance matrix is positive definite. By default, PROC MIXED sets the initial values of the parameters to 1. You can use the PARMS statement to specify other initial values.

TYPE=SIMPLE is an alias for TYPE=VC.

TYPE=SP(EXPA)(*c-list*) specifies the spatial anisotropic exponential structure, where *c-list* is a list of variables indicating the coordinates. This structure has (i, j)th element equal to

$$\sigma^2 \prod_{k=1}^{c} \exp\{-\theta_k d(i,j,k)^{p_k}\}$$

where c is the number of coordinates and $d(i, j, k)$ is the absolute distance between the kth coordinate ($k = 1, \ldots, c$) of the ith and jth observations in the input data set. There are $2c + 1$ parameters to be estimated: θ_k, p_k ($k = 1, \ldots, c$), and σ^2.

You might want to constrain some of the EXPA parameters to known values. For example, suppose you have three coordinate variables C1, C2, and C3 and you want to constrain the powers p_k to equal 2, as in Sacks et al. (1989). Suppose further that you want to model covariance across the entire input data set and you suspect the θ_k and σ^2 estimates are close to 3, 4, 5, and 1, respectively. Then specify the following statements:

```
repeated / type=sp(expa)(c1 c2 c3)
   subject=intercept;
parms (3) (4) (5) (2) (2) (2) (1) /
   hold=4,5,6;
```

TYPE=SP(EXPGA)(c_1 c_2)

TYPE=SP(GAUGA)(c_1 c_2)

TYPE=SP(SPHGA)(c_1 c_2) specify modifications of the isotropic SP(EXP), SP(SPH), and SP(GAU) covariance structures that allow for geometric anisotropy in two dimensions. The coordinates are specified by the variables c1 and c2.

If the spatial process is geometrically anisotropic in $\mathbf{c} = [c_{i1}, c_{i2}]$, then it is isotropic in the coordinate system

$$\mathbf{Ac} = \begin{bmatrix} 1 & 0 \\ 0 & \lambda \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \mathbf{c} = \mathbf{c}^*$$

for a properly chosen angle θ and scaling factor λ. Elliptical isocorrelation contours are thereby transformed to spherical contours, adding two parameters to the respective isotropic covariance structures. Euclidean distances (see Table 56.14) are expressed in terms of $\mathbf{c}^*$.

The angle θ of the clockwise rotation is reported in radians, $0 \leq \theta \leq 2\pi$. The scaling parameter λ represents the ratio of the range parameters in the direction of the major and minor axis of the correlation contours. In other words, following a rotation of the coordinate system by angle θ, isotropy is achieved by compressing or magnifying distances in one coordinate by the factor λ.

Fixing $\lambda = 1.0$ reduces the models to isotropic ones for any angle of rotation. If the scaling parameter is held constant at 1.0, you should also hold constant the angle of rotation, as in the following statements:

```
repeated / type=sp(expga)(gxc gyc)
           subject=intercept;
parms (6) (1.0) (0.0) (1) / hold=2,3;
```

If λ is fixed at any other value than 1.0, the angle of rotation can be estimated. Specifying a starting grid of angles and scaling factors can considerably improve the convergence properties of the optimization algorithm for these models. Only a single random effect with geometrically anisotropic structure is permitted.

TYPE=SP(MATERN)(*c-list*)

TYPE=SP(MATHSW)(*c-list*) specifies covariance structures in the Matérn class of covariance functions (Matérn 1986). Two observations for the same subject (block of $\mathbf{R}$) that are Euclidean distance d_{ij} apart have covariance

$$\sigma^2 \frac{1}{\Gamma(\nu)} \left(\frac{d_{ij}}{2\rho} \right)^{\nu} 2K_\nu(d_{ij}/\rho) \qquad \nu > 0,\ \rho > 0$$

where K_ν is the modified Bessel function of the second kind of (real) order $\nu > 0$. The smoothness (continuity) of a stochastic process with covariance function in this class increases with ν. The Matérn class thus enables data-driven estimation of the smoothness properties. The covariance is identical to the exponential model for $\nu = 0.5$ (TYPE=SP(EXP)(*c-list*)), while for $\nu = 1$ the model advocated by Whittle (1954) results.

As $\nu \rightarrow \infty$ the model approaches the gaussian covariance structure (TYPE=SP(GAU)(*c-list*)).

The MATHSW structure represents the Matérn class in the parameterization of Handcock and Stein (1993) and Handcock and Wallis (1994),

$$\sigma^2 \frac{1}{\Gamma(\nu)} \left(\frac{d_{ij}\sqrt{\nu}}{\rho} \right)^{\nu} 2K_{\nu} \left(\frac{2d_{ij}\sqrt{\nu}}{\rho} \right)$$

Since computation of the function K_{ν} and its derivatives is numerically very intensive, fitting models with Matérn covariance structures can be more time-consuming than with other spatial covariance structures. Good starting values are essential.

TYPE=SP(POW)(*c-list*)

TYPE=SP(POWA)(*c-list*) specifies the spatial power structures. When the estimated value of ρ becomes negative, the computed covariance is multiplied by $\cos(\pi d_{ij})$ to account for the negativity.

TYPE=TOEP<(q)> specifies a banded Toeplitz structure. This can be viewed as a moving-average structure with order equal to $q - 1$. The TYPE=TOEP option is a full Toeplitz matrix, which can be viewed as an autoregressive structure with order equal to the dimension of the matrix. The specification TYPE=TOEP(1) is the same as $\sigma^2 I$, where I is an identity matrix, and it can be useful for specifying the same variance component for several effects.

TYPE=TOEPH<(q)> specifies a heterogeneous banded Toeplitz structure. In Table 56.13, σ_i^2 is the ith variance parameter and ρ_j is the jth correlation parameter satisfying $|\rho_j| < 1$. If you specify the order parameter q, then PROC MIXED estimates only the first q bands of the matrix, setting all higher bands equal to 0. The option TOEPH(1) is equivalent to both the UN(1) and UNR(1) options.

TYPE=UN<(q)> specifies a completely general (unstructured) covariance matrix parameterized directly in terms of variances and covariances. The variances are constrained to be nonnegative, and the covariances are unconstrained. This structure is not constrained to be nonnegative definite in order to avoid nonlinear constraints; however, you can use the FA0 structure if you want this constraint to be imposed by a Cholesky factorization. If you specify the order parameter q, then PROC MIXED estimates only the first q bands of the matrix, setting all higher bands equal to 0.

TYPE=UNR<(q)> specifies a completely general (unstructured) covariance matrix parameterized in terms of variances and correlations. This structure fits the same model as the TYPE=UN(q) option but with a different parameterization. The ith variance parameter is σ_i^2. The parameter ρ_{jk} is the correlation between the jth and kth measurements; it satisfies $|\rho_{jk}| < 1$. If you specify the order parameter r, then PROC MIXED estimates only the first q bands of the matrix, setting all higher bands equal to zero.

TYPE=UN@AR(1)

TYPE=UN@CS

TYPE=UN@UN specify direct (Kronecker) product structures designed for multivariate repeated measures (see Galecki 1994). These structures are constructed by taking the Kronecker product of an unstructured matrix (modeling covariance across the multivariate observations) with an additional covariance matrix (modeling covariance across time or another factor). The upper-left value in the second matrix is constrained to equal 1 to identify the model. See the *SAS/IML User's Guide* for more details about direct products.

To use these structures in the REPEATED statement, you must specify two distinct REPEATED effects, both of which must be included in the CLASS statement. The first effect indicates the multivariate observations, and the second identifies the levels of time or some additional factor. Note that the input data set must still be constructed in "univariate" format; that is, all dependent observations are still listed observation-wise in one single variable. Although this construction provides for general modeling possibilities, it forces you to construct variables indicating both dimensions of the Kronecker product.

For example, suppose your observed data consist of heights and weights of several children measured over several successive years. Your input data set should then contain variables similar to the following:

- Y, all of the heights and weights, with a separate observation for each
- Var, indicating whether the measurement is a height or a weight
- Year, indicating the year of measurement
- Child, indicating the child on which the measurement was taken

Your PROC MIXED statements for a Kronecker AR(1) structure across years would then be as follows:

```
proc mixed;
   class Var Year Child;
   model Y = Var Year Var*Year;
   repeated Var Year / type=un@ar(1)
                       subject=Child;
run;
```

You should nearly always want to model different means for the multivariate observations; hence the inclusion of Var in the MODEL statement. The preceding mean model consists of cell means for all combinations of VAR and YEAR.

TYPE=VC specifies standard variance components and is the default structure for both the RANDOM and REPEATED statements. In the RANDOM statement, a distinct variance component is assigned to each effect. In the REPEATED statement, this structure is usually used only with the GROUP= option to specify a heterogeneous variance model.

Jennrich and Schluchter (1986) provide general information about the use of covariance structures, and Wolfinger (1996) presents details about many of the heterogeneous structures.

Modeling with spatial covariance structures is discussed in many sources, for example, Marx and Thompson (1987), Zimmerman and Harville (1991), Cressie (1993), Brownie, Bowman, and Burton (1993), Stroup, Baenziger, and Mulitze (1994), Brownie and Gumpertz (1997), Gotway and Stroup (1997), Chilès and Delfiner (1999), Schabenberger and Gotway (2005), and Littell et al. (2006).

WEIGHT Statement

WEIGHT *variable* **;**

If you do not specify a REPEATED statement, the WEIGHT statement operates exactly like the one in PROC GLM. In this case PROC MIXED replaces $\mathbf{X}'\mathbf{X}$ and $\mathbf{Z}'\mathbf{Z}$ with $\mathbf{X}'\mathbf{W}\mathbf{X}$ and $\mathbf{Z}'\mathbf{W}\mathbf{Z}$, where $\mathbf{W}$ is the diagonal weight matrix. If you specify a REPEATED statement, then the WEIGHT statement replaces $\mathbf{R}$ with $\mathbf{LRL}$, where $\mathbf{L}$ is a diagonal matrix with elements $\mathbf{W}^{-1/2}$. Observations with nonpositive or missing weights are not included in the PROC MIXED analysis.

Details: MIXED Procedure

Mixed Models Theory

This section provides an overview of a likelihood-based approach to general linear mixed models. This approach simplifies and unifies many common statistical analyses, including those involving repeated measures, random effects, and random coefficients. The basic assumption is that the data are linearly related to unobserved multivariate normal random variables. For extensions to nonlinear and nonnormal situations see the documentation of the GLIMMIX and NLMIXED procedures. Additional theory and examples are provided in Littell et al. (2006), Verbeke and Molenberghs (1997, 2000), and Brown and Prescott (1999).

Matrix Notation

Suppose that you observe n data points $y_1, \ldots, y_n$ and that you want to explain them by using n values for each of p explanatory variables $x_{11}, \ldots, x_{1p}$, $x_{21}, \ldots, x_{2p}$, $\ldots$, $x_{n1}, \ldots, x_{np}$. The x_{ij} values can be either regression-type continuous variables or dummy variables indicating class membership. The standard linear model for this setup is

$$y_i = \sum_{j=1}^{p} x_{ij}\beta_j + \epsilon_i \quad i = 1, \ldots, n$$

where $\beta_1, \ldots, \beta_p$ are unknown *fixed-effects* parameters to be estimated and $\epsilon_1, \ldots, \epsilon_n$ are unknown independent and identically distributed normal (Gaussian) random variables with mean 0 and variance σ^2.

The preceding equations can be written simultaneously by using vectors and a matrix, as follows:

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} x_{11} & x_{12} & \ldots & x_{1p} \\ x_{21} & x_{22} & \ldots & x_{2p} \\ \vdots & \vdots & & \vdots \\ x_{n1} & x_{n2} & \ldots & x_{np} \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_p \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \vdots \\ \epsilon_n \end{bmatrix}$$

For convenience, simplicity, and extendability, this entire system is written as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$$

where $\mathbf{y}$ denotes the vector of observed y_i's, $\mathbf{X}$ is the known matrix of x_{ij}'s, $\boldsymbol{\beta}$ is the unknown fixed-effects parameter vector, and $\boldsymbol{\epsilon}$ is the unobserved vector of independent and identically distributed Gaussian random errors.

In addition to denoting data, random variables, and explanatory variables in the preceding fashion, the subsequent development makes use of basic matrix operators such as transpose ($'$), inverse ($^{-1}$), generalized inverse ($^{-}$), determinant ($|\cdot|$), and matrix multiplication. See Searle (1982) for details about these and other matrix techniques.

Formulation of the Mixed Model

The previous general linear model is certainly a useful one (Searle 1971), and it is the one fitted by the GLM procedure. However, many times the distributional assumption about ϵ is too restrictive. The mixed model extends the general linear model by allowing a more flexible specification of the covariance matrix of ϵ. In other words, it allows for both correlation and heterogeneous variances, although you still assume normality.

The mixed model is written as

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

where everything is the same as in the general linear model except for the addition of the known design matrix, $\mathbf{Z}$, and the vector of unknown *random-effects parameters,* $\boldsymbol{\gamma}$. The matrix $\mathbf{Z}$ can contain either continuous or dummy variables, just like $\mathbf{X}$. The name *mixed model* comes from the fact that the model contains both fixed-effects parameters, $\boldsymbol{\beta}$, and random-effects parameters, $\boldsymbol{\gamma}$. See Henderson (1990) and Searle, Casella, and McCulloch (1992) for historical developments of the mixed model.

A key assumption in the foregoing analysis is that $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$ are normally distributed with

$$\mathrm{E}\begin{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

$$\mathrm{Var}\begin{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\epsilon} \end{bmatrix} = \begin{bmatrix} \mathbf{G} & \mathbf{0} \\ \mathbf{0} & \mathbf{R} \end{bmatrix}$$

The variance of $\mathbf{y}$ is, therefore, $\mathbf{V} = \mathbf{ZGZ}' + \mathbf{R}$. You can model $\mathbf{V}$ by setting up the random-effects design matrix $\mathbf{Z}$ and by specifying covariance structures for $\mathbf{G}$ and $\mathbf{R}$.

Note that this is a general specification of the mixed model, in contrast to many texts and articles that discuss only simple random effects. Simple random effects are a special case of the general specification with $\mathbf{Z}$ containing dummy variables, $\mathbf{G}$ containing variance components in a diagonal structure, and $\mathbf{R} = \sigma^2\mathbf{I}_n$, where $\mathbf{I}_n$ denotes the $n \times n$ identity matrix. The general linear model is a further special case with $\mathbf{Z} = \mathbf{0}$ and $\mathbf{R} = \sigma^2\mathbf{I}_n$.

The following two examples illustrate the most common formulations of the general linear mixed model.

Example: Growth Curve with Compound Symmetry

Suppose that you have three growth curve measurements for s individuals and that you want to fit an overall linear trend in time. Your $\mathbf{X}$ matrix is as follows:

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \\ \vdots & \vdots \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$$

The first column (coded entirely with 1s) fits an intercept, and the second column (coded with times of $1, 2, 3$) fits a slope. Here, $n = 3s$ and $p = 2$.

Suppose further that you want to introduce a common correlation among the observations from a single individual, with correlation being the same for all individuals. One way of setting this up in the general mixed model is to eliminate the $\mathbf{Z}$ and $\mathbf{G}$ matrices and let the $\mathbf{R}$ matrix be block diagonal with blocks corresponding to the individuals and with each block having the *compound-symmetry* structure. This structure has two unknown parameters, one modeling a common covariance and the other modeling a residual variance. The form for $\mathbf{R}$ would then be as follows:

$$\mathbf{R} = \begin{bmatrix} \sigma_1^2+\sigma^2 & \sigma_1^2 & \sigma_1^2 & & & \\ \sigma_1^2 & \sigma_1^2+\sigma^2 & \sigma_1^2 & & & \\ \sigma_1^2 & \sigma_1^2 & \sigma_1^2+\sigma^2 & & & \\ & & & \ddots & & \\ & & & & \sigma_1^2+\sigma^2 & \sigma_1^2 & \sigma_1^2 \\ & & & & \sigma_1^2 & \sigma_1^2+\sigma^2 & \sigma_1^2 \\ & & & & \sigma_1^2 & \sigma_1^2 & \sigma_1^2+\sigma^2 \end{bmatrix}$$

where blanks denote zeros. There are $3s$ rows and columns altogether, and the common correlation is $\sigma_1^2/(\sigma_1^2 + \sigma^2)$.

The PROC MIXED statements to fit this model are as follows:

```
proc mixed;
   class indiv;
   model y = time;
   repeated / type=cs subject=indiv;
run;
```

Here, indiv is a classification variable indexing individuals. The MODEL statement fits a straight line for time ; the intercept is fit by default just as in PROC GLM. The REPEATED statement models the $\mathbf{R}$ matrix: TYPE=CS specifies the compound symmetry structure, and SUBJECT=INDIV specifies the blocks of $\mathbf{R}$.

An alternative way of specifying the common intra-individual correlation is to let

$$\mathbf{Z} = \begin{bmatrix} 1 & & \\ 1 & & \\ 1 & & \\ & 1 & \\ & 1 & \\ & 1 & \\ & & \ddots \\ & & 1 \\ & & 1 \\ & & 1 \end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix} \sigma_1^2 & & & \\ & \sigma_1^2 & & \\ & & \ddots & \\ & & & \sigma_1^2 \end{bmatrix}$$

and $\mathbf{R} = \sigma^2 \mathbf{I}_n$. The $\mathbf{Z}$ matrix has $3s$ rows and s columns, and $\mathbf{G}$ is $s \times s$.

You can set up this model in PROC MIXED in two different but equivalent ways:

```
proc mixed;
   class indiv;
   model y = time;
   random indiv;
run;

proc mixed;
   class indiv;
   model y = time;
   random intercept / subject=indiv;
run;
```

Both of these specifications fit the same model as the previous one that used the REPEATED statement; however, the RANDOM specifications constrain the correlation to be positive, whereas the REPEATED specification leaves the correlation unconstrained.

Example: Split-Plot Design

The split-plot design involves two experimental treatment factors, A and B, and two different sizes of experimental units to which they are applied (see Winer 1971, Snedecor and Cochran 1980, Milliken and Johnson 1992, and Steel, Torrie, and Dickey 1997). The levels of A are randomly assigned to the larger-sized experimental unit, called *whole plots*, whereas the levels of B are assigned to the smaller-sized experimental unit, the *subplots*. The subplots are assumed to be nested within the whole plots, so that a whole plot consists of a cluster of subplots and a level of A is applied to the entire cluster.

Such an arrangement is often necessary by nature of the experiment, the classical example being the application of fertilizer to large plots of land and different crop varieties planted in subdivisions of the large plots. For this example, fertilizer is the whole-plot factor A and variety is the subplot factor B.

The first example is a split-plot design for which the whole plots are arranged in a randomized block design. The appropriate PROC MIXED statements are as follows:

```
proc mixed;
   class a b block;
   model y = a|b;
   random block a*block;
run;
```

Here

$$\mathbf{R} = \sigma^2 \mathbf{I}_{24}$$

and $\mathbf{X}$, $\mathbf{Z}$, and $\mathbf{G}$ have the following form:

$$\mathbf{X} = \begin{bmatrix}
1 & 1 & & & 1 & & 1 & & & & & \\
1 & 1 & & & & 1 & & 1 & & & & \\
1 & & 1 & & 1 & & & & 1 & & & \\
1 & & 1 & & & 1 & & & & 1 & & \\
1 & & & 1 & 1 & & & & & & 1 & \\
1 & & & 1 & & 1 & & & & & & 1 \\
\vdots & & \vdots & & \vdots & & & & \vdots & & & \\
1 & 1 & & & 1 & & 1 & & & & & \\
1 & 1 & & & & 1 & & 1 & & & & \\
1 & & 1 & & 1 & & & & 1 & & & \\
1 & & 1 & & & 1 & & & & 1 & & \\
1 & & & 1 & 1 & & & & & & 1 & \\
1 & & & 1 & & 1 & & & & & & 1
\end{bmatrix}$$

$$\mathbf{Z} = \begin{bmatrix}
1 & & 1 & & & & & & & & & & \\
1 & & 1 & & & & & & & & & & \\
1 & & & 1 & & & & & & & & & \\
1 & & & 1 & & & & & & & & & \\
1 & & & & 1 & & & & & & & & \\
1 & & & & 1 & & & & & & & & \\
 & 1 & & & & 1 & & & & & & & \\
 & 1 & & & & 1 & & & & & & & \\
 & 1 & & & & & 1 & & & & & & \\
 & 1 & & & & & 1 & & & & & & \\
 & 1 & & & & & & 1 & & & & & \\
 & 1 & & & & & & 1 & & & & & \\
 & & 1 & & & & & & 1 & & & & \\
 & & 1 & & & & & & 1 & & & & \\
 & & 1 & & & & & & & 1 & & & \\
 & & 1 & & & & & & & 1 & & & \\
 & & 1 & & & & & & & & 1 & & \\
 & & 1 & & & & & & & & 1 & & \\
 & & & 1 & & & & & & & & 1 & \\
 & & & 1 & & & & & & & & 1 & \\
 & & & 1 & & & & & & & & & 1 \\
 & & & 1 & & & & & & & & & 1
\end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix}
\sigma^2_B & & & & & & & \\
 & \sigma^2_B & & & & & & \\
 & & \sigma^2_B & & & & & \\
 & & & \sigma^2_B & & & & \\
 & & & & \sigma^2_{AB} & & & \\
 & & & & & \sigma^2_{AB} & & \\
 & & & & & & \ddots & \\
 & & & & & & & \sigma^2_{AB}
\end{bmatrix}$$

where σ^2_B is the variance component for Block and σ^2_{AB} is the variance component for A*Block. Changing the RANDOM statement as follows fits the same model, but with **Z** and **G** sorted differently:

```
random int a / subject=block;
```

$$\mathbf{Z} = \begin{bmatrix}
1 & 1 & & & & & & & & & & & & & & & & & & \\
1 & 1 & & & & & & & & & & & & & & & & & & \\
1 & & 1 & & & & & & & & & & & & & & & & & \\
1 & & 1 & & & & & & & & & & & & & & & & & \\
1 & & & 1 & & & & & & & & & & & & & & & & \\
1 & & & 1 & & & & & & & & & & & & & & & & \\
 & & & & 1 & 1 & & & & & & & & & & & & & & \\
 & & & & 1 & 1 & & & & & & & & & & & & & & \\
 & & & & 1 & & 1 & & & & & & & & & & & & & \\
 & & & & 1 & & 1 & & & & & & & & & & & & & \\
 & & & & 1 & & & 1 & & & & & & & & & & & & \\
 & & & & 1 & & & 1 & & & & & & & & & & & & \\
 & & & & & & & & 1 & 1 & & & & & & & & & & \\
 & & & & & & & & 1 & 1 & & & & & & & & & & \\
 & & & & & & & & 1 & & 1 & & & & & & & & & \\
 & & & & & & & & 1 & & 1 & & & & & & & & & \\
 & & & & & & & & 1 & & & 1 & & & & & & & & \\
 & & & & & & & & 1 & & & 1 & & & & & & & & \\
 & & & & & & & & & & & & 1 & 1 & & & & & & \\
 & & & & & & & & & & & & 1 & 1 & & & & & & \\
 & & & & & & & & & & & & 1 & & 1 & & & & & \\
 & & & & & & & & & & & & 1 & & 1 & & & & & \\
 & & & & & & & & & & & & 1 & & & 1 & & & & \\
 & & & & & & & & & & & & 1 & & & 1 & & & & \\
 & & & & & & & & & & & & & & & & 1 & 1 & & \\
 & & & & & & & & & & & & & & & & 1 & 1 & &
\end{bmatrix}$$

$$\mathbf{G} = \begin{bmatrix}
\sigma^2_B & & & & & & & & & & & & & & & & & & & \\
 & \sigma^2_{AB} & & & & & & & & & & & & & & & & & & \\
 & & \sigma^2_{AB} & & & & & & & & & & & & & & & & & \\
 & & & \sigma^2_{AB} & & & & & & & & & & & & & & & & \\
 & & & & \ddots & & & & & & & & & & & & & & & \\
 & & & & & \sigma^2_B & & & & & & & & & & & & & & \\
 & & & & & & \sigma^2_{AB} & & & & & & & & & & & & & \\
 & & & & & & & \sigma^2_{AB} & & & & & & & & & & & & \\
 & & & & & & & & \sigma^2_{AB} & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & & \\
 & & & & & & & & & & & & & & & & & & &
\end{bmatrix}$$

Estimating Covariance Parameters in the Mixed Model

Estimation is more difficult in the mixed model than in the general linear model. Not only do you have $\boldsymbol{\beta}$ as in the general linear model, but you have unknown parameters in $\boldsymbol{\gamma}$, $\mathbf{G}$, and $\mathbf{R}$ as well. Least squares is no longer the best method. *Generalized least squares* (GLS) is more appropriate, minimizing

$$(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})'\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta})$$

However, it requires knowledge of $\mathbf{V}$ and, therefore, knowledge of $\mathbf{G}$ and $\mathbf{R}$. Lacking such information, one approach is to use *estimated* GLS, in which you insert some reasonable estimate for $\mathbf{V}$ into the minimization problem. The goal thus becomes finding a reasonable estimate of $\mathbf{G}$ and $\mathbf{R}$.

In many situations, the best approach is to use *likelihood-based* methods, exploiting the assumption that $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$ are normally distributed (Hartley and Rao 1967; Patterson and Thompson 1971; Harville 1977; Laird and Ware 1982; Jennrich and Schluchter 1986). PROC MIXED implements two likelihood-based methods: *maximum likelihood* (ML) and *restricted/residual maximum likelihood* (REML). A favorable theoretical property of ML and REML is that they accommodate data that are missing at random (Rubin 1976; Little 1995).

PROC MIXED constructs an objective function associated with ML or REML and maximizes it over all unknown parameters. Using calculus, it is possible to reduce this maximization problem to one over only the parameters in $\mathbf{G}$ and $\mathbf{R}$. The corresponding log-likelihood functions are as follows:

$$\text{ML}: \quad l(\mathbf{G},\mathbf{R}) = -\frac{1}{2}\log|\mathbf{V}| - \frac{1}{2}\mathbf{r}'\mathbf{V}^{-1}\mathbf{r} - \frac{n}{2}\log(2\pi)$$

$$\text{REML}: \quad l_R(\mathbf{G},\mathbf{R}) = -\frac{1}{2}\log|\mathbf{V}| - \frac{1}{2}\log|\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}| - \frac{1}{2}\mathbf{r}'\mathbf{V}^{-1}\mathbf{r} - \frac{n-p}{2}\log(2\pi)\}$$

where $\mathbf{r} = \mathbf{y} - \mathbf{X}(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-}\mathbf{X}'\mathbf{V}^{-1}\mathbf{y}$ and p is the rank of $\mathbf{X}$. PROC MIXED actually minimizes -2 times these functions by using a ridge-stabilized Newton-Raphson algorithm. Lindstrom and Bates (1988) provide reasons for preferring Newton-Raphson to the Expectation-Maximum (EM) algorithm described in Dempster, Laird, and Rubin (1977) and Laird, Lange, and Stram (1987), as well as analytical details for implementing a QR-decomposition approach to the problem. Wolfinger, Tobias, and Sall (1994) present the sweep-based algorithms that are implemented in PROC MIXED.

One advantage of using the Newton-Raphson algorithm is that the second derivative matrix of the objective function evaluated at the optima is available upon completion. Denoting this matrix $\mathbf{H}$, the asymptotic theory of maximum likelihood (see Serfling 1980) shows that $2\mathbf{H}^{-1}$ is an asymptotic variance-covariance matrix of the estimated parameters of $\mathbf{G}$ and $\mathbf{R}$. Thus, tests and confidence intervals based on asymptotic normality can be obtained. However, these can be unreliable in small samples, especially for parameters such as variance components that have sampling distributions that tend to be skewed to the right.

If a residual variance σ^2 is a part of your mixed model, it can usually be *profiled* out of the likelihood. This means solving analytically for the optimal σ^2 and plugging this expression back into the likelihood formula (see Wolfinger, Tobias, and Sall 1994). This reduces the number of optimization parameters by one and can improve convergence properties. PROC MIXED profiles the residual variance out of the log likelihood whenever it appears reasonable to do so. This includes the case when $\mathbf{R}$ equals $\sigma^2\mathbf{I}$ and when it has blocks with a compound symmetry, time series, or spatial structure. PROC MIXED does not profile the log likelihood when $\mathbf{R}$ has unstructured blocks, when you use the HOLD= or NOITER option in the PARMS statement, or when you use the NOPROFILE option in the PROC MIXED statement.

Instead of ML or REML, you can use the noniterative MIVQUE0 method to estimate $\mathbf{G}$ and $\mathbf{R}$ (Rao 1972; LaMotte 1973; Wolfinger, Tobias, and Sall 1994). In fact, by default PROC MIXED uses MIVQUE0 estimates as starting values for the ML and REML procedures. For variance component models, another estimation method involves equating Type 1, 2, or 3 expected mean squares to their observed values and solving the resulting system. However, Swallow and Monahan (1984) present simulation evidence favoring REML and ML over MIVQUE0 and other method-of-moment estimators.

Estimating Fixed and Random Effects in the Mixed Model

ML, REML, MIVQUE0, or Type1–Type3 provide estimates of $\mathbf{G}$ and $\mathbf{R}$, which are denoted $\widehat{\mathbf{G}}$ and $\widehat{\mathbf{R}}$, respectively. To obtain estimates of $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$, the standard method is to solve the *mixed model equations* (Henderson 1984):

$$\begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{Z} \\ \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}+\widehat{\mathbf{G}}^{-1} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \\ \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \end{bmatrix}$$

The solutions can also be written as

$$\widehat{\boldsymbol{\beta}} = (\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{y}$$
$$\widehat{\boldsymbol{\gamma}} = \widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{V}}^{-1}(\mathbf{y}-\mathbf{X}\widehat{\boldsymbol{\beta}})$$

and have connections with empirical Bayes estimators (Laird and Ware 1982, Carlin and Louis 1996).

Note that the mixed model equations are extended normal equations and that the preceding expression assumes that $\widehat{\mathbf{G}}$ is nonsingular. For the extreme case where the eigenvalues of $\widehat{\mathbf{G}}$ are very large, $\widehat{\mathbf{G}}^{-1}$ contributes very little to the equations and $\widehat{\boldsymbol{\gamma}}$ is close to what it would be if $\boldsymbol{\gamma}$ actually contained fixed-effects parameters. On the other hand, when the eigenvalues of $\widehat{\mathbf{G}}$ are very small, $\widehat{\mathbf{G}}^{-1}$ dominates the equations and $\widehat{\boldsymbol{\gamma}}$ is close to 0. For intermediate cases, $\widehat{\mathbf{G}}^{-1}$ can be viewed as shrinking the fixed-effects estimates of $\boldsymbol{\gamma}$ toward 0 (Robinson 1991).

If $\widehat{\mathbf{G}}$ is singular, then the mixed model equations are modified (Henderson 1984) as follows:

$$\begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}\widehat{\mathbf{L}} \\ \widehat{\mathbf{L}}'\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \widehat{\mathbf{L}}'\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{Z}\widehat{\mathbf{L}}+\mathbf{I} \end{bmatrix} \begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\tau}} \end{bmatrix} = \begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \\ \widehat{\mathbf{L}}'\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{y} \end{bmatrix}$$

where $\widehat{\mathbf{L}}$ is the lower-triangular Cholesky root of $\widehat{\mathbf{G}}$, satisfying $\widehat{\mathbf{G}} = \widehat{\mathbf{L}}\widehat{\mathbf{L}}'$. Both $\widehat{\boldsymbol{\tau}}$ and a generalized inverse of the left-hand-side coefficient matrix are then transformed by using $\widehat{\mathbf{L}}$ to determine $\widehat{\boldsymbol{\gamma}}$.

An example of when the singular form of the equations is necessary is when a variance component estimate falls on the boundary constraint of 0.

Model Selection

The previous section on estimation assumes the specification of a mixed model in terms of $\mathbf{X}$, $\mathbf{Z}$, $\mathbf{G}$, and $\mathbf{R}$. Even though $\mathbf{X}$ and $\mathbf{Z}$ have known elements, their specific form and construction are flexible, and several possibilities can present themselves for a particular data set. Likewise, several different covariance structures for $\mathbf{G}$ and $\mathbf{R}$ might be reasonable.

Space does not permit a thorough discussion of model selection, but a few brief comments and references are in order. First, subject matter considerations and objectives are of great importance when selecting a model; see Diggle (1988) and Lindsey (1993).

Second, when the data themselves are looked to for guidance, many of the graphical methods and diagnostics appropriate for the general linear model extend to the mixed model setting as well (Christensen, Pearson, and Johnson 1992).

Finally, a likelihood-based approach to the mixed model provides several statistical measures for model adequacy as well. The most common of these are the likelihood ratio test and Akaike's and Schwarz's criteria (Bozdogan 1987; Wolfinger 1993; Keselman et al. 1998, 1999).

Statistical Properties

If $\mathbf{G}$ and $\mathbf{R}$ are known, $\widehat{\boldsymbol{\beta}}$ is the *best linear unbiased estimator* (BLUE) of $\boldsymbol{\beta}$, and $\widehat{\boldsymbol{\gamma}}$ is the *best linear unbiased predictor* (BLUP) of $\boldsymbol{\gamma}$ (Searle 1971; Harville 1988, 1990; Robinson 1991; McLean, Sanders, and Stroup 1991). Here, "best" means minimum mean squared error. The covariance matrix of $(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}, \widehat{\boldsymbol{\gamma}} - \boldsymbol{\gamma})$ is

$$\mathbf{C} = \begin{bmatrix} \mathbf{X}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{X}'\mathbf{R}^{-1}\mathbf{Z} \\ \mathbf{Z}'\mathbf{R}^{-1}\mathbf{X} & \mathbf{Z}'\mathbf{R}^{-1}\mathbf{Z} + \mathbf{G}^{-1} \end{bmatrix}^{-}$$

where $^-$ denotes a generalized inverse (see Searle 1971).

However, $\mathbf{G}$ and $\mathbf{R}$ are usually unknown and are estimated by using one of the aforementioned methods. These estimates, $\widehat{\mathbf{G}}$ and $\widehat{\mathbf{R}}$, are therefore simply substituted into the preceding expression to obtain

$$\widehat{\mathbf{C}} = \begin{bmatrix} \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{X}'\widehat{\mathbf{R}}^{-1}\mathbf{Z} \\ \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{X} & \mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{Z} + \widehat{\mathbf{G}}^{-1} \end{bmatrix}^{-}$$

as the approximate variance-covariance matrix of $(\widehat{\boldsymbol{\beta}} - \boldsymbol{\beta}, \widehat{\boldsymbol{\gamma}} - \boldsymbol{\gamma})$. In this case, the BLUE and BLUP acronyms no longer apply, but the word *empirical* is often added to indicate such an approximation. The appropriate acronyms thus become EBLUE and EBLUP.

McLean and Sanders (1988) show that $\widehat{\mathbf{C}}$ can also be written as

$$\widehat{\mathbf{C}} = \begin{bmatrix} \widehat{\mathbf{C}}_{11} & \widehat{\mathbf{C}}_{21}' \\ \widehat{\mathbf{C}}_{21} & \widehat{\mathbf{C}}_{22} \end{bmatrix}$$

where

$$\begin{aligned} \widehat{\mathbf{C}}_{11} &= (\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-} \\ \widehat{\mathbf{C}}_{21} &= -\widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{V}}^{-1}\mathbf{X}\widehat{\mathbf{C}}_{11} \\ \widehat{\mathbf{C}}_{22} &= (\mathbf{Z}'\widehat{\mathbf{R}}^{-1}\mathbf{Z} + \widehat{\mathbf{G}}^{-1})^{-1} - \widehat{\mathbf{C}}_{21}\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{Z}\widehat{\mathbf{G}} \end{aligned}$$

Note that $\widehat{\mathbf{C}}_{11}$ is the familiar estimated generalized least squares formula for the variance-covariance matrix of $\widehat{\boldsymbol{\beta}}$.

As a cautionary note, $\widehat{\mathbf{C}}$ tends to underestimate the true sampling variability of $(\widehat{\boldsymbol{\beta}}\ \widehat{\boldsymbol{\gamma}})$ because no account is made for the uncertainty in estimating $\mathbf{G}$ and $\mathbf{R}$. Although inflation factors have been proposed (Kackar and Harville 1984; Kass and Steffey 1989; Prasad and Rao 1990), they tend to be small for data sets that are fairly well balanced. PROC MIXED does not compute any inflation factors by default, but rather accounts for the downward bias by using the approximate t and F statistics described subsequently. The DDFM=KENWARDROGER option in the MODEL statement prompts PROC MIXED to compute a specific inflation factor along with Satterthwaite-based degrees of freedom.

Inference and Test Statistics

For inferences concerning the covariance parameters in your model, you can use likelihood-based statistics. One common likelihood-based statistic is the *Wald* Z, which is computed as the parameter estimate divided by its asymptotic standard error. The asymptotic standard errors are computed from the inverse of the second derivative matrix of the likelihood with respect to each of the covariance parameters. The Wald Z is valid for large samples, but it can be unreliable for small data sets and for parameters such as variance components, which are known to have a skewed or bounded sampling distribution.

A better alternative is the likelihood ratio χ^2 statistic. This statistic compares two covariance models, one a special case of the other. To compute it, you must run PROC MIXED twice, once for each of the two models, and then subtract the corresponding values of -2 times the log likelihoods. You can use either ML or REML to construct this statistic, which tests whether the full model is necessary beyond the reduced model.

As long as the reduced model does not occur on the boundary of the covariance parameter space, the χ^2 statistic computed in this fashion has a large-sample χ^2 distribution that is χ^2 with degrees of freedom equal to the difference in the number of covariance parameters between the two models. If the reduced model does occur on the boundary of the covariance parameter space, the asymptotic distribution becomes a mixture of χ^2 distributions (Self and Liang 1987). A common example of this is when you are testing that a variance component equals its lower boundary constraint of 0.

A final possibility for obtaining inferences concerning the covariance parameters is to simulate or resample data from your model and construct empirical sampling distributions of the parameters. The SAS macro language and the ODS system are useful tools in this regard.

F *and* t *Tests for Fixed- and Random-Effects Parameters*

For inferences concerning the fixed- and random-effects parameters in the mixed model, consider estimable linear combinations of the following form:

$$\mathbf{L}\begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\gamma} \end{bmatrix}$$

The estimability requirement (Searle 1971) applies only to the $\boldsymbol{\beta}$ portion of $\mathbf{L}$, because any linear combination of $\boldsymbol{\gamma}$ is estimable. Such a formulation in terms of a general $\mathbf{L}$ matrix encompasses a wide variety of common inferential procedures such as those employed with Type 1–Type 3 tests and LS-means. The CONTRAST and ESTIMATE statements in PROC MIXED enable you to specify your own $\mathbf{L}$ matrices. Typically, inference on fixed effects is the focus, and, in this case, the $\boldsymbol{\gamma}$ portion of $\mathbf{L}$ is assumed to contain all 0s.

Statistical inferences are obtained by testing the hypothesis

$$H : \mathbf{L}\begin{bmatrix} \boldsymbol{\beta} \\ \boldsymbol{\gamma} \end{bmatrix} = 0$$

or by constructing point and interval estimates.

When $\mathbf{L}$ consists of a single row, a general t statistic can be constructed as follows (see McLean and Sanders 1988, Stroup 1989a):

$$t = \frac{\mathbf{L}\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}}{\sqrt{\mathbf{L}\widehat{\mathbf{C}}\mathbf{L}'}}$$

Under the assumed normality of $\boldsymbol{\gamma}$ and $\boldsymbol{\epsilon}$, t has an exact t distribution only for data exhibiting certain types of balance and for some special unbalanced cases. In general, t is only approximately t-distributed, and its degrees of freedom must be estimated. See the DDFM= option for a description of the various degrees-of-freedom methods available in PROC MIXED.

With $\widehat{\nu}$ being the approximate degrees of freedom, the associated confidence interval is

$$\mathbf{L}\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix} \pm t_{\widehat{\nu},\alpha/2}\sqrt{\mathbf{L}\widehat{\mathbf{C}}\mathbf{L}'}$$

where $t_{\widehat{\nu},\alpha/2}$ is the $(1-\alpha/2)100$th percentile of the $t_{\widehat{\nu}}$ distribution.

When the rank of $\mathbf{L}$ is greater than 1, PROC MIXED constructs the following general F statistic:

$$F = \frac{\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}' \mathbf{L}'(\mathbf{L}\widehat{\mathbf{C}}\mathbf{L}')^{-1}\mathbf{L}\begin{bmatrix} \widehat{\boldsymbol{\beta}} \\ \widehat{\boldsymbol{\gamma}} \end{bmatrix}}{r}$$

where $r = \text{rank}(\mathbf{L}\widehat{\mathbf{C}}\mathbf{L}')$. Analogous to t, F in general has an approximate F distribution with r numerator degrees of freedom and $\widehat{\nu}$ denominator degrees of freedom.

The t and F statistics enable you to make inferences about your fixed effects, which account for the variance-covariance model you select. An alternative is the χ^2 statistic associated with the likelihood ratio test. This statistic compares two fixed-effects models, one a special case of the other. It is computed just as when comparing different covariance models, although you should use ML and not REML here because the penalty term associated with restricted likelihoods depends upon the fixed-effects specification.

F *Tests With the ANOVAF Option*

The ANOVAF option computes F tests by the following method in models with REPEATED statement and without RANDOM statement. Let $\mathbf{L}$ denote the matrix of estimable functions for the hypothesis $H{:}\mathbf{L}\boldsymbol{\beta} = \mathbf{0}$, where $\boldsymbol{\beta}$ are the fixed-effects parameters. Let $\mathbf{M} = \mathbf{L}'(\mathbf{L}\mathbf{L}')^{-}\mathbf{L}$, and suppose that $\widehat{\mathbf{C}}$ denotes the estimated variance-covariance matrix of $\widehat{\boldsymbol{\beta}}$ (see the section "Statistical Properties" for the construction of $\widehat{\mathbf{C}}$).

The ANOVAF F statistics are computed as

$$F_A = \widehat{\boldsymbol{\beta}}'\mathbf{L}'\left(\mathbf{L}\mathbf{L}'\right)^{-1}\mathbf{L}\widehat{\boldsymbol{\beta}}\Big/t_1 = \widehat{\boldsymbol{\beta}}'\mathbf{M}\widehat{\boldsymbol{\beta}}\Big/t_1$$

Notice that this is a modification of the usual F statistic where $(\mathbf{L}\widehat{\mathbf{C}}\mathbf{L}')^{-1}$ is replaced with $(\mathbf{L}\mathbf{L}')^{-1}$ and rank$(\mathbf{L})$ is replaced with $t_1 = \text{trace}(\mathbf{M}\widehat{\mathbf{C}})$; see, for example, Brunner, Domhof, and Langer

(2002, Sec. 5.4). The p-values for this statistic are computed from either an F_{ν_1,ν_2} or an $F_{\nu_1,\infty}$ distribution. The respective degrees of freedom are determined by the MIXED procedure as follows:

$$\nu_1 = \frac{t_1^2}{\text{trace}(\mathbf{M}\widehat{\mathbf{C}}\mathbf{M}\widehat{\mathbf{C}})}$$

$$\nu_2^* = \frac{2t_1^2}{\mathbf{g}'\mathbf{A}\mathbf{g}}$$

$$\nu_2 = \begin{cases} \max\{\min\{\nu_2^*, df_e\}, 1\} & \mathbf{g}'\mathbf{A}\mathbf{g} > 1\text{E}3 \times \text{MACEPS} \\ 1 & \text{otherwise} \end{cases}$$

The term $\mathbf{g}'\mathbf{A}\mathbf{g}$ in the term ν_2^* for the denominator degrees of freedom is based on approximating Var[trace($\mathbf{M}\widehat{\mathbf{C}}$)] based on a first-order Taylor series about the true covariance parameters. This generalizes results in the appendix of Brunner, Dette, and Munk (1997) to a broader class of models. The vector $\mathbf{g} = [g_1, \cdots, g_q]$ contains the partial derivatives

$$\text{trace}\left(\mathbf{L}'\left(\mathbf{L}\mathbf{L}'\right)^{-1}\mathbf{L}\frac{\partial\widehat{\mathbf{C}}}{\partial\theta_i}\right)$$

and $\mathbf{A}$ is the asymptotic variance-covariance matrix of the covariance parameter estimates (ASYCOV option).

PROC MIXED reports ν_1 and ν_2 as "NumDF" and "DenDF" under the "ANOVA F" heading in the output. The corresponding p-values are denoted as "Pr > F(DDF)" for F_{ν_1,ν_2} and "Pr > F(infty)" for $F_{\nu_1,\infty}$, respectively.

P-values computed with the ANOVAF option can be identical to the nonparametric tests in Akritas, Arnold, and Brunner (1997) and in Brunner, Domhof, and Langer (2002), provided that the response data consist of properly created (and sorted) ranks and that the covariance parameters are estimated by MIVQUE0 in models with REPEATED statement and properly chosen SUBJECT= and/or GROUP= effects.

If you model an unstructured covariance matrix in a longitudinal model with one or more repeated factors, the ANOVAF results are identical to a multivariate MANOVA where degrees of freedom are corrected with the Greenhouse-Geiser adjustment (Greenhouse and Geiser 1959). For example, suppose that factor `A` has 2 levels and factor `B` has 4 levels. The following two sets of statements produce the same p-values:

```
proc mixed data=Mydata anovaf method=mivque0;
   class id A B;
   model score = A | B / chisq;
   repeated / type=un subject=id;
   ods select Tests3;
run;

proc transpose data=MyData out=tdata;
   by id;
   var score;
proc glm data=tdata;
   model col: = / nouni;
```

```
   repeated A 2, B 4;
   ods output ModelANOVA=maov epsilons=eps;
run;
proc transpose data=eps(where=(substr(statistic,1,3)='Gre')) out=teps;
  var cvalue1;
run;

data aov; set maov;
  if (_n_ = 1) then merge teps;
  if (Source='A') then do;
      pFddf = ProbF;
      pFinf = 1 - probchi(df*Fvalue,df);
      output;
  end; else if (Source='B') then do;
      pFddf = ProbFGG;
      pFinf = 1 - probchi(df*col1*Fvalue,df*col1);
      output;
  end; else if (Source='A*B') then do;
      pfddF = ProbFGG;
      pFinf = 1 - probchi(df*col2*Fvalue,df*col2);
      output;
  end;
proc print data=aov label noobs;
  label Source = 'Effect'
        df     = 'NumDF'
        Fvalue = 'Value'
        pFddf  = 'Pr > F(DDF)'
        pFinf  = 'Pr > F(infty)';
  var Source df Fvalue pFddf pFinf;
  format pF: pvalue6.;
run;
```

The PROC GLM code produces *p*-values that correspond to the ANOVAF *p*-values shown as Pr > F(DDF) in the MIXED output. The subsequent DATA step computes the *p*-values that correspond to Pr > F(infty) in the PROC MIXED output.

Parameterization of Mixed Models

Recall that a mixed model is of the form

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma} + \boldsymbol{\epsilon}$$

where $\mathbf{y}$ represents univariate data, $\boldsymbol{\beta}$ is an unknown vector of fixed effects with known model matrix $\mathbf{X}$, $\boldsymbol{\gamma}$ is an unknown vector of random effects with known model matrix $\mathbf{Z}$, and $\boldsymbol{\epsilon}$ is an unknown random error vector.

PROC MIXED constructs a mixed model according to the specifications in the MODEL, RANDOM, and REPEATED statements. Each effect in the MODEL statement generates one or more columns in the model matrix $\mathbf{X}$, and each effect in the RANDOM statement generates one or more columns in the model matrix $\mathbf{Z}$. Effects in the REPEATED statement do not generate model

matrices; they serve only to index observations within subjects. This section shows precisely how PROC MIXED builds **X** and **Z**.

Intercept

By default, all models automatically include a column of 1s in **X** to estimate a fixed-effect intercept parameter μ. You can use the NOINT option in the MODEL statement to suppress this intercept. The NOINT option is useful when you are specifying a classification effect in the MODEL statement and you want the parameter estimate to be in terms of the mean response for each level of that effect, rather than in terms of a deviation from an overall mean.

By contrast, the intercept is not included by default in **Z**. To obtain a column of 1s in **Z**, you must specify in the RANDOM statement either the INTERCEPT effect or some effect that has only one level.

Regression Effects

Numeric variables, or polynomial terms involving them, can be included in the model as regression effects (covariates). The actual values of such terms are included as columns of the model matrices **X** and **Z**. You can use the bar operator with a regression effect to generate polynomial effects. For instance, X|X|X expands to X X*X X*X*X, a cubic model.

Main Effects

If a classification variable has m levels, PROC MIXED generates m columns in the model matrix for its main effect. Each column is an indicator variable for a given level. The order of the columns is the sort order of the values of their levels and can be controlled with the ORDER= option in the PROC MIXED statement. Table 56.16 is an example.

Table 56.16 Example of Main Effects

Data		**I**	A		B		
A	B	μ	A1	A2	B1	B2	B3
1	1	1	1	0	1	0	0
1	2	1	1	0	0	1	0
1	3	1	1	0	0	0	1
2	1	1	0	1	1	0	0
2	2	1	0	1	0	1	0
2	3	1	0	1	0	0	1

Typically, there are more columns for these effects than there are degrees of freedom for them. In other words, PROC MIXED uses an overparameterized model.

Interaction Effects

Often a model includes interaction (crossed) effects. With an interaction, PROC MIXED first reorders the terms to correspond to the order of the variables in the CLASS statement. Thus, B*A becomes A*B if A precedes B in the CLASS statement. Then, PROC MIXED generates columns for all combinations of levels that occur in the data. The order of the columns is such that the rightmost variables in the cross index faster than the leftmost variables (Table 56.17). Empty columns (that would contain all 0s) are not generated for **X**, but they are for **Z**.

Table 56.17 Example of Interaction Effects

Data		**I**	A		B			A*B					
A	B	μ	A1	A2	B1	B2	B3	A1B1	A1B2	A1B3	A2B1	A2B2	A2B3
1	1	1	1	0	1	0	0	1	0	0	0	0	0
1	2	1	1	0	0	1	0	0	1	0	0	0	0
1	3	1	1	0	0	0	1	0	0	1	0	0	0
2	1	1	0	1	1	0	0	0	0	0	1	0	0
2	2	1	0	1	0	1	0	0	0	0	0	1	0
2	3	1	0	1	0	0	1	0	0	0	0	0	1

In the preceding matrix, main-effects columns are not linearly independent of crossed-effects columns; in fact, the column space for the crossed effects contains the space of the main effect.

When your model contains many interaction effects, you might be able to code them more parsimoniously by using the bar operator (|). The bar operator generates all possible interaction effects. For example, A|B|C expands to A B A*B C A*C B*C A*B*C. To eliminate higher-order interaction effects, use the at sign (@) in conjunction with the bar operator. For instance, A|B|C|D @2 expands to A B A*B C A*C B*C D A*D B*D C*D.

Nested Effects

Nested effects are generated in the same manner as crossed effects. Hence, the design columns generated by the following two statements are the same (but the ordering of the columns is different):

```
model Y=A B(A);

model Y=A A*B;
```

The nesting operator in PROC MIXED is more a notational convenience than an operation distinct from crossing. Nested effects are typically characterized by the property that the nested variables never appear as main effects. The order of the variables within nesting parentheses is made to correspond to the order of these variables in the CLASS statement. The order of the columns is such that variables outside the parentheses index faster than those inside the parentheses, and the rightmost nested variables index faster than the leftmost variables (Table 56.18).

Table 56.18 Example of Nested Effects

Data		I	A		B(A)					
A	B	μ	A1	A2	B1A1	B2A1	B3A1	B1A2	B2A2	B3A2
1	1	1	1	0	1	0	0	0	0	0
1	2	1	1	0	0	1	0	0	0	0
1	3	1	1	0	0	0	1	0	0	0
2	1	1	0	1	0	0	0	1	0	0
2	2	1	0	1	0	0	0	0	1	0
2	3	1	0	1	0	0	0	0	0	1

Note that nested effects are often distinguished from interaction effects by the implied randomization structure of the design. That is, they usually indicate random effects within a fixed-effects framework. The fact that random effects can be modeled directly in the RANDOM statement might make the specification of nested effects in the MODEL statement unnecessary.

Continuous-Nesting-Class Effects

When a continuous variable nests with a classification variable, the design columns are constructed by multiplying the continuous values into the design columns for the class effect (Table 56.19).

Table 56.19 Example of Continuous-Nesting-Class Effects

Data		I	A		X(A)	
X	A	μ	A1	A2	X(A1)	X(A2)
21	1	1	1	0	21	0
24	1	1	1	0	24	0
22	1	1	1	0	22	0
28	2	1	0	1	0	28
19	2	1	0	1	0	19
23	2	1	0	1	0	23

This model estimates a separate slope for X within each level of A.

Continuous-by-Class Effects

Continuous-by-class effects generate the same design columns as continuous-nesting-class effects. The two models are made different by the presence of the continuous variable as a regressor by itself, as well as a contributor to a compound effect. Table 56.20 shows an example.

Table 56.20 Example of Continuous-by-Class Effects

Data		**I**	**X**	A		X*A	
X	A	μ	X	A1	A2	X*A1	X*A2
21	1	1	21	1	0	21	0
24	1	1	24	1	0	24	0
22	1	1	22	1	0	22	0
28	2	1	28	0	1	0	28
19	2	1	19	0	1	0	19
23	2	1	23	0	1	0	23

You can use continuous-by-class effects to test for homogeneity of slopes.

General Effects

An example that combines all the effects is X1*X2*A*B*C (D E). The continuous list comes first, followed by the crossed list, followed by the nested list in parentheses. You should be aware of the sequencing of parameters when you use the CONTRAST or ESTIMATE statement to compute some function of the parameter estimates.

Effects might be renamed by PROC MIXED to correspond to ordering rules. For example, B*A(E D) might be renamed A*B(D E) to satisfy the following:

- Classification variables that occur outside parentheses (crossed effects) are sorted in the order in which they appear in the CLASS statement.
- Variables within parentheses (nested effects) are sorted in the order in which they appear in the CLASS statement.

The sequencing of the parameters generated by an effect can be described by which variables have their levels indexed faster:

- Variables in the crossed list index faster than variables in the nested list.
- Within a crossed or nested list, variables to the right index faster than variables to the left.

For example, suppose a model includes four effects—A, B, C, and D—each having two levels, 1 and 2. Suppose the CLASS statement is as follows:

```
class A B C D;
```

Then the order of the parameters for the effect B*A(C D), which is renamed A*B (C D), is

$$\begin{array}{llll} A_1B_1C_1D_1 \rightarrow & A_1B_2C_1D_1 \rightarrow & A_2B_1C_1D_1 \rightarrow & A_2B_2C_1D_1 \rightarrow \\ A_1B_1C_1D_2 \rightarrow & A_1B_2C_1D_2 \rightarrow & A_2B_1C_1D_2 \rightarrow & A_2B_2C_1D_2 \rightarrow \\ A_1B_1C_2D_1 \rightarrow & A_1B_2C_2D_1 \rightarrow & A_2B_1C_2D_1 \rightarrow & A_2B_2C_2D_1 \rightarrow \\ A_1B_1C_2D_2 \rightarrow & A_1B_2C_2D_2 \rightarrow & A_2B_1C_2D_2 \rightarrow & A_2B_2C_2D_2 \end{array}$$

Note that first the crossed effects B and A are sorted in the order in which they appear in the CLASS statement so that A precedes B in the parameter list. Then, for each combination of the nested effects in turn, combinations of A and B appear. The B effect moves fastest because it is rightmost in the cross list. Then A moves next fastest, and D moves next fastest. The C effect is the slowest since it is leftmost in the nested list.

When numeric levels are used, levels are sorted by their character format, which might not correspond to their numeric sort sequence (for example, noninteger levels). Therefore, it is advisable to include a desired format for numeric levels or to use the ORDER=INTERNAL option in the PROC MIXED statement to ensure that levels are sorted by their internal values.

Implications of the Non-Full-Rank Parameterization

For models with fixed effects involving classification variables, there are more design columns in $\mathbf{X}$ constructed than there are degrees of freedom for the effect. Thus, there are linear dependencies among the columns of $\mathbf{X}$. In this event, all of the parameters are not estimable; there is an infinite number of solutions to the mixed model equations. PROC MIXED uses a generalized inverse (a g_2-inverse, Pringle and Rayner, 1971) to obtain values for the estimates (Searle 1971). The solution values are not displayed unless you specify the SOLUTION option in the MODEL statement. The solution has the characteristic that estimates are 0 whenever the design column for that parameter is a linear combination of previous columns. With this parameterization, hypothesis tests are constructed to test linear functions of the parameters that are estimable.

Some procedures (such as the CATMOD procedure) reparameterize models to full rank by using restrictions on the parameters. PROC GLM and PROC MIXED do not reparameterize, making the hypotheses that are commonly tested more understandable. See Goodnight (1978) for additional reasons for not reparameterizing.

Missing Level Combinations

PROC MIXED handles missing level combinations of classification variables similarly to the way PROC GLM does. Both procedures delete fixed-effects parameters corresponding to missing levels in order to preserve estimability. However, PROC MIXED does not delete missing level combinations for random-effects parameters because linear combinations of the random-effects parameters are always estimable. These conventions can affect the way you specify your CONTRAST and ESTIMATE coefficients.

Residuals and Influence Diagnostics

Residual Diagnostics

Consider a residual vector of the form $\widetilde{\mathbf{e}} = \mathbf{PY}$, where $\mathbf{P}$ is a projection matrix, possibly an oblique projector. A typical element $\widetilde{e}_i$ with variance v_i and estimated variance $\widehat{v}_i$ is said to be *standardized*

as

$$\frac{\widetilde{e}_i}{\sqrt{\text{Var}[\widetilde{e}_i]}} = \frac{\widetilde{e}_i}{\sqrt{v_i}}$$

and *studentized* as

$$\frac{\widetilde{e}_i}{\sqrt{\widehat{v}_i}}$$

External studentization uses an estimate of $\text{Var}[\widetilde{e}_i]$ that does not involve the ith observation. Externally studentized residuals are often preferred over studentized residuals because they have well-known distributional properties in standard linear models for independent data.

Residuals that are scaled by the estimated variance of the response, i.e., $\widetilde{e}_i / \sqrt{\widehat{\text{Var}}[Y_i]}$, are referred to as Pearson-type residuals.

Marginal and Conditional Residuals

The marginal and conditional means in the linear mixed model are $\text{E}[\mathbf{Y}] = \mathbf{X}\boldsymbol{\beta}$ and $\text{E}[\mathbf{Y}|\boldsymbol{\gamma}] = \mathbf{X}\boldsymbol{\beta} + \mathbf{Z}\boldsymbol{\gamma}$, respectively. Accordingly, the vector $\mathbf{r}_m$ of marginal residuals is defined as

$$\mathbf{r}_m = \mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}$$

and the vector $\mathbf{r}_c$ of conditional residuals is

$$\mathbf{r}_c = \mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}} - \mathbf{Z}\widehat{\boldsymbol{\gamma}} = \mathbf{r}_m - \mathbf{Z}\widehat{\boldsymbol{\gamma}}$$

Following Gregoire, Schabenberger, and Barrett (1995), let $\mathbf{Q} = \mathbf{X}(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}\mathbf{X}'$ and $\mathbf{K} = \mathbf{I} - \mathbf{Z}\widehat{\mathbf{G}}\mathbf{Z}'\widehat{\mathbf{V}}^{-1}$. Then

$$\widehat{\text{Var}}[\mathbf{r}_m] = \widehat{\mathbf{V}} - \mathbf{Q}$$
$$\widehat{\text{Var}}[\mathbf{r}_c] = \mathbf{K}(\widehat{\mathbf{V}} - \mathbf{Q})\mathbf{K}'$$

For an individual observation the raw, studentized, and Pearson-type residuals computed by the MIXED procedure are given in Table 56.21.

Table 56.21 Residual Types Computed by the MIXED Procedure

Type of Residual	Marginal	Conditional
Raw	$r_{mi} = Y_i - \mathbf{x}_i'\widehat{\boldsymbol{\beta}}$	$r_{ci} = r_{mi} - \mathbf{z}_i'\widehat{\boldsymbol{\gamma}}$
Studentized	$r_{mi}^{student} = \frac{r_{mi}}{\sqrt{\widehat{\text{Var}}[r_{mi}]}}$	$r_{ci}^{student} = \frac{r_{ci}}{\sqrt{\widehat{\text{Var}}[r_{ci}]}}$
Pearson	$r_{mi}^{pearson} = \frac{r_{mi}}{\sqrt{\widehat{\text{Var}}[Y_i]}}$	$r_{ci}^{pearson} = \frac{r_{ci}}{\sqrt{\widehat{\text{Var}}[Y_i\|\boldsymbol{\gamma}]}}$

When the OUTPM= option is specified in addition to the RESIDUAL option in the MODEL statement, $r_{mi}^{student}$ and $r_{mi}^{pearson}$ are added to the data set as variables Resid, StudentResid, and PearsonResid, respectively. When the OUTP= option is specified, $r_{ci}^{student}$ and $r_{ci}^{pearson}$ are added to the data set. Raw residuals are part of the OUTPM= and OUTP= data sets without the RESIDUAL option.

Scaled Residuals

For correlated data, a set of scaled quantities can be defined through the Cholesky decomposition of the variance-covariance matrix. Since fitted residuals in linear models are rank-deficient, it is customary to draw on the variance-covariance matrix of the data. If $\text{Var}[\mathbf{Y}] = \mathbf{V}$ and $\mathbf{C}'\mathbf{C} = \mathbf{V}$, then $\mathbf{C}'^{-1}\mathbf{Y}$ has uniform dispersion and its elements are uncorrelated.

Scaled residuals in a mixed model are meaningful for quantities based on the marginal distribution of the data. Let $\widehat{\mathbf{C}}$ denote the Cholesky root of $\widehat{\mathbf{V}}$, so that $\widehat{\mathbf{C}}'\widehat{\mathbf{C}} = \widehat{\mathbf{V}}$, and define

$$\begin{aligned}\mathbf{Y}_c &= \widehat{\mathbf{C}}'^{-1}\mathbf{Y} \\ \mathbf{r}_{m(c)} &= \widehat{\mathbf{C}}'^{-1}\mathbf{r}_m\end{aligned}$$

By analogy with other scalings, the inverse Cholesky decomposition can also be applied to the residual vector, $\widehat{\mathbf{C}}'^{-1}\mathbf{r}_m$, although $\mathbf{V}$ is not the variance-covariance matrix of $\mathbf{r}_m$.

To diagnose whether the covariance structure of the model has been specified correctly can be difficult based on $\mathbf{Y}_c$, since the inverse Cholesky transformation affects the expected value of $\mathbf{Y}_c$. You can draw on $\mathbf{r}_{m(c)}$ as a vector of (approximately) uncorrelated data with constant mean.

When the OUTPM= option in the MODEL statement is specified in addition to the VCIRY option, $\mathbf{Y}_c$ is added as variable ScaledDep and $\mathbf{r}_{m(c)}$ is added as ScaledResid to the data set.

Influence Diagnostics

Basic Idea and Statistics

The general idea of quantifying the influence of one or more observations relies on computing parameter estimates based on all data points, removing the cases in question from the data, refitting the model, and computing statistics based on the change between full-data and reduced-data estimation. Influence statistics can be coarsely grouped by the aspect of estimation that is their primary target:

- overall measures compare changes in objective functions: (restricted) likelihood distance (Cook and Weisberg 1982, Ch. 5.2)
- influence on parameter estimates: Cook's D (Cook 1977, 1979), MDFFITS (Belsley, Kuh, and Welsch 1980, p. 32)
- influence on precision of estimates: CovRatio and CovTrace
- influence on fitted and predicted values: PRESS residual, PRESS statistic (Allen 1974), DFFITS (Belsley, Kuh, and Welsch 1980, p. 15)

- outlier properties: internally and externally studentized residuals, leverage

For linear models for uncorrelated data, it is not necessary to refit the model after removing a data point in order to measure the impact of an observation on the model. The change in fixed effect estimates, residuals, residual sums of squares, and the variance-covariance matrix of the fixed effects can be computed based on the fit to the full data alone. By contrast, in mixed models several important complications arise. Data points can affect not only the fixed effects but also the covariance parameter estimates on which the fixed-effects estimates depend. Furthermore, closed-form expressions for computing the change in important model quantities might not be available.

This section provides background material for the various influence diagnostics available with the MIXED procedure. See the section "Mixed Models Theory" on page 3968 for relevant expressions and definitions. The parameter vector $\boldsymbol{\theta}$ denotes all unknown parameters in the $\mathbf{R}$ and $\mathbf{G}$ matrix.

The observations whose influence is being ascertained are represented by the set U and referred to simply as "the observations in U." The estimate of a parameter vector, such as $\boldsymbol{\beta}$, obtained from all observations except those in the set U is denoted $\widehat{\boldsymbol{\beta}}_{(U)}$. In case of a matrix $\mathbf{A}$, the notation $\mathbf{A}_{(U)}$ represents the matrix with the rows in U removed; these rows are collected in $\mathbf{A}_U$. If $\mathbf{A}$ is symmetric, then notation $\mathbf{A}_{(U)}$ implies removal of rows and columns. The vector Y_U comprises the responses of the data points being removed, and $\mathbf{V}_{(U)}$ is the variance-covariance matrix of the remaining observations. When $k = 1$, lowercase notation emphasizes that single points are removed, such as $\mathbf{A}_{(u)}$.

Managing the Covariance Parameters

An important component of influence diagnostics in the mixed model is the estimated variance-covariance matrix $\mathbf{V} = \mathbf{ZGZ}' + \mathbf{R}$. To make the dependence on the vector of covariance parameters explicit, write it as $\mathbf{V}(\boldsymbol{\theta})$. If one parameter, σ^2, is profiled or factored out of $\mathbf{V}$, the remaining parameters are denoted as $\boldsymbol{\theta}^*$. Notice that in a model where $\mathbf{G}$ is diagonal and $\mathbf{R} = \sigma^2\mathbf{I}$, the parameter vector $\boldsymbol{\theta}^*$ contains the ratios of each variance component and σ^2 (see Wolfinger, Tobias, and Sall 1994). When ITER=0, two scenarios are distinguished:

1. If the residual variance is not profiled, either because the model does not contain a residual variance or because it is part of the Newton-Raphson iterations, then $\widehat{\boldsymbol{\theta}}_{(U)} \equiv \widehat{\boldsymbol{\theta}}$.

2. If the residual variance is profiled, then $\widehat{\boldsymbol{\theta}}^*_{(U)} \equiv \widehat{\boldsymbol{\theta}}^*$ and $\widehat{\sigma}^2_{(U)} \neq \widehat{\sigma}^2$. Influence statistics such as Cook's D and internally studentized residuals are based on $\mathbf{V}(\widehat{\boldsymbol{\theta}})$, whereas externally studentized residuals and the DFFITS statistic are based on $\mathbf{V}(\widehat{\boldsymbol{\theta}}_U) = \sigma^2_{(U)}\mathbf{V}(\widehat{\boldsymbol{\theta}}^*)$. In a random components model with uncorrelated errors, for example, the computation of $\mathbf{V}(\widehat{\boldsymbol{\theta}}_U)$ involves scaling of $\widehat{\mathbf{G}}$ and $\widehat{\mathbf{R}}$ by the full-data estimate $\widehat{\sigma}^2$ and multiplying the result with the reduced-data estimate $\widehat{\sigma}^2_{(U)}$.

Certain statistics, such as MDFFITS, CovRatio, and CovTrace, require an estimate of the variance of the fixed effects that is based on the reduced number of observations. For example, $\mathbf{V}(\widehat{\boldsymbol{\theta}}_U)$ is evaluated at the reduced-data parameter estimates but computed for the entire data set. The matrix $\mathbf{V}_{(U)}(\widehat{\boldsymbol{\theta}}_{(U)})$, on the other hand, has rows and columns corresponding to the points in U removed. The resulting matrix is evaluated at the delete-case estimates.

When influence analysis is iterative, the entire vector $\boldsymbol{\theta}$ is updated, whether the residual variance is profiled or not. The matrices to be distinguished here are $\mathbf{V}(\widehat{\boldsymbol{\theta}})$, $\mathbf{V}(\widehat{\boldsymbol{\theta}}_{(U)})$, and $\mathbf{V}_{(U)}(\widehat{\boldsymbol{\theta}}_{(U)})$, with unambiguous notation.

Predicted Values, PRESS Residual, and PRESS Statistic

An unconditional predicted value is $\widehat{y}_i = \mathbf{x}_i'\widehat{\boldsymbol{\beta}}$, where the vector $\mathbf{x}_i$ is the ith row of $\mathbf{X}$. The (raw) residual is given as $\widehat{\epsilon}_i = y_i - \widehat{y}_i$, and the PRESS *residual* is

$$\widehat{\epsilon}_{i(U)} = y_i - \mathbf{x}_i'\widehat{\boldsymbol{\beta}}_{(U)}$$

The PRESS *statistic* is the sum of the squared PRESS residuals,

$$PRESS = \sum_{i \in U} \widehat{\epsilon}^2_{i(U)}$$

where the sum is over the observations in U.

If EFFECT=, SIZE=, or KEEP= is not specified, PROC MIXED computes the PRESS residual for each observation selected through SELECT= (or all observations if SELECT= is not given). If EFFECT=, SIZE=, or KEEP= is specified, the procedure computes $PRESS$.

Leverage

For the general mixed model, leverage can be defined through the projection matrix that results from a transformation of the model with the inverse of the Cholesky decomposition of $\mathbf{V}$, or through an oblique projector. The MIXED procedure follows the latter path in the computation of influence diagnostics. The leverage value reported for the ith observation is the ith diagonal entry of the matrix

$$\mathbf{H} = \mathbf{X}(\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}\mathbf{X})^{-}\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}$$

which is the weight of the observation in contributing to its own predicted value, $\mathbf{H} = d\widehat{\mathbf{Y}}/d\mathbf{Y}$.

While $\mathbf{H}$ is idempotent, it is generally not symmetric and thus not a projection matrix in the narrow sense.

The properties of these leverages are generalizations of the properties in models with diagonal variance-covariance matrices. For example, $\widehat{\mathbf{Y}} = \mathbf{HY}$, and in a model with intercept and $\mathbf{V} = \sigma^2\mathbf{I}$, the leverage values

$$h_{ii} = \mathbf{x}_i'(\mathbf{X}'\mathbf{X})^{-}\mathbf{x}_i$$

are $h^l_{ii} = 1/n \leq h_{ii} \leq 1 = h^u_{ii}$ and $\sum_{i=1}^n h_{ii} = \text{rank}(\mathbf{X})$. The lower bound for h_{ii} is achieved in an intercept-only model, and the upper bound is achieved in a saturated model. The trace of $\mathbf{H}$ equals the rank of $\mathbf{X}$.

If v_{ij} denotes the element in row i, column j of $\mathbf{V}^{-1}$, then for a model containing only an intercept the diagonal elements of $\mathbf{H}$ are

$$h_{ii} = \frac{\sum_{j=1}^n v_{ij}}{\sum_{i=1}^n \sum_{j=1}^n v_{ij}}$$

Because $\sum_{j=1}^{n} v_{ij}$ is a sum of elements in the ith row of the *inverse* variance-covariance matrix, h_{ii} can be negative, even if the correlations among data points are nonnegative. In case of a saturated model with $\mathbf{X} = \mathbf{I}$, $h_{ii} = 1.0$.

Internally and Externally Studentized Residuals

See the section "Residual Diagnostics" on page 3986 for the distinction between standardization, studentization, and scaling of residuals. Internally studentized marginal and conditional residuals are computed with the RESIDUAL option of the MODEL statement. The INFLUENCE option computes internally and externally studentized marginal residuals.

The computation of internally studentized residuals relies on the diagonal entries of $\mathbf{V}(\widehat{\boldsymbol{\theta}}) - \mathbf{Q}(\widehat{\boldsymbol{\theta}})$, where $\mathbf{Q}(\widehat{\boldsymbol{\theta}}) = \mathbf{X}(\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}\mathbf{X})^{-}\mathbf{X}'$. Externally studentized residuals require iterative influence analysis or a profiled residual variance. In the former case the studentization is based on $\mathbf{V}(\widehat{\boldsymbol{\theta}}_U)$; in the latter case it is based on $\sigma^2_{(U)}\mathbf{V}(\widehat{\boldsymbol{\theta}}^*)$.

Cook's D

Cook's D statistic is an invariant norm that measures the influence of observations in U on a vector of parameter estimates (Cook 1977). In case of the fixed-effects coefficients, let

$$\boldsymbol{\delta}_{(U)} = \widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_{(U)}$$

Then the MIXED procedure computes

$$D(\boldsymbol{\beta}) = \boldsymbol{\delta}'_{(U)}\widehat{\mathrm{Var}}[\widehat{\boldsymbol{\beta}}]^{-}\boldsymbol{\delta}_{(U)}/\mathrm{rank}(\mathbf{X})$$

where $\widehat{\mathrm{Var}}[\widehat{\boldsymbol{\beta}}]^{-}$ is the matrix that results from sweeping $(\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}})^{-1}\mathbf{X})^{-}$.

If $\mathbf{V}$ is known, Cook's D can be calibrated according to a chi-square distribution with degrees of freedom equal to the rank of $\mathbf{X}$ (Christensen, Pearson, and Johnson 1992). For estimated $\mathbf{V}$ the calibration can be carried out according to an $F(\mathrm{rank}(\mathbf{X}), n - \mathrm{rank}(\mathbf{X}))$ distribution. To interpret D on a familiar scale, Cook (1979) and Cook and Weisberg (1982, p. 116) refer to the 50th percentile of the reference distribution. If D is equal to that percentile, then removing the points in U moves the fixed-effects coefficient vector from the center of the confidence region to the 50% confidence ellipsoid (Myers 1990, p. 262).

In the case of iterative influence analysis, the MIXED procedure also computes a D-type statistic for the covariance parameters. If $\boldsymbol{\Gamma}$ is the asymptotic variance-covariance matrix of $\widehat{\boldsymbol{\theta}}$, then MIXED computes

$$D_{\boldsymbol{\theta}} = (\widehat{\boldsymbol{\theta}} - \widehat{\boldsymbol{\theta}}_{(U)})'\widehat{\boldsymbol{\Gamma}}^{-1}(\widehat{\boldsymbol{\theta}} - \widehat{\boldsymbol{\theta}}_{(U)})$$

DFFITS and MDFFITS

A DFFIT measures the change in predicted values due to removal of data points. If this change is standardized by the externally estimated standard error of the predicted value in the full data, the DFFITS statistic of Belsley, Kuh, and Welsch (1980, p. 15) results:

$$\mathrm{DFFITS}_i = (\widehat{y}_i - \widehat{y}_{i(u)})/\mathrm{ese}(\widehat{y}_i)$$

The MIXED procedure computes DFFITS when the EFFECT= or SIZE= modifier of the INFLUENCE option is not in effect. In general, an external estimate of the estimated standard error is used. When ITER > 0, the estimate is

$$\text{ese}(\widehat{y}_i) = \sqrt{\mathbf{x}_i'(\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}}_{(u)})^{-}\mathbf{X})^{-1}\mathbf{x}_i}$$

When ITER=0 and σ^2 is profiled, then

$$\text{ese}(\widehat{y}_i) = \widehat{\sigma}_{(u)}\sqrt{\mathbf{x}_i'(\mathbf{X}'\mathbf{V}(\widehat{\boldsymbol{\theta}}^*)^{-1}\mathbf{X})^{-}\mathbf{x}_i}$$

When the EFFECT=, SIZE=, or KEEP= modifier is specified, the MIXED procedure computes a multivariate version suitable for the deletion of multiple data points. The statistic, termed MDFFITS after the MDFFIT statistic of Belsley, Kuh, and Welsch (1980, p. 32), is closely related to Cook's D. Consider the case $\mathbf{V} = \sigma^2\mathbf{V}(\boldsymbol{\theta}^*)$ so that

$$\text{Var}[\widehat{\boldsymbol{\beta}}] = \sigma^2(\mathbf{X}'\mathbf{V}(\boldsymbol{\theta}^*)^{-1}\mathbf{X})^{-}$$

and let $\widetilde{\text{Var}}[\widehat{\boldsymbol{\beta}}_{(U)}]$ be an estimate of $\text{Var}[\widehat{\boldsymbol{\beta}}_{(U)}]$ that does not use the observations in U. The MDFFITS statistic is then computed as

$$\text{MDFFITS}(\boldsymbol{\beta}) = \boldsymbol{\delta}_{(U)}'\widetilde{\text{Var}}[\widehat{\boldsymbol{\beta}}_{(U)}]^{-}\boldsymbol{\delta}_{(U)}/\text{rank}(\mathbf{X})$$

If ITER=0 and σ^2 is profiled, then $\widetilde{\text{Var}}[\widehat{\boldsymbol{\beta}}_{(U)}]^{-}$ is obtained by sweeping

$$\widehat{\sigma}^2_{(U)}(\mathbf{X}'_{(U)}\mathbf{V}_{(U)}(\widehat{\boldsymbol{\theta}}^*)^{-}\mathbf{X}_{(U)})^{-}$$

The underlying idea is that if $\boldsymbol{\theta}^*$ were known, then

$$(\mathbf{X}'_{(U)}\mathbf{V}_{(U)}(\boldsymbol{\theta}^*)^{-1}\mathbf{X}_{(U)})^{-}$$

would be $\text{Var}[\widehat{\boldsymbol{\beta}}]/\sigma^2$ in a generalized least squares regression with all but the data in U.

In the case of iterative influence analysis, $\widetilde{\text{Var}}[\widehat{\boldsymbol{\beta}}_{(U)}]$ is evaluated at $\widehat{\boldsymbol{\theta}}_{(U)}$. Furthermore, a MDFFITS-type statistic is then computed for the covariance parameters:

$$\text{MDFFITS}(\boldsymbol{\theta}) = (\widehat{\boldsymbol{\theta}} - \widehat{\boldsymbol{\theta}}_{(U)})'\widehat{\text{Var}}[\widehat{\boldsymbol{\theta}}_{(U)}]^{-1}(\widehat{\boldsymbol{\theta}} - \widehat{\boldsymbol{\theta}}_{(U)})$$

Covariance Ratio and Trace

These statistics depend on the availability of an external estimate of $\mathbf{V}$, or at least of σ^2. Whereas Cook's D and MDFFITS measure the impact of data points on a vector of parameter estimates, the covariance-based statistics measure impact on their precision. Following Christensen, Pearson, and Johnson (1992), the MIXED procedure computes

$$\text{CovTrace}(\boldsymbol{\beta}) = |\text{trace}(\widehat{\text{Var}}[\widehat{\boldsymbol{\beta}}]^{-}\ \widetilde{\text{Var}}[\widehat{\boldsymbol{\beta}}_{(U)}]) - \text{rank}(\mathbf{X})|$$

$$\text{CovRatio}(\boldsymbol{\beta}) = \frac{\det_{ns}(\widetilde{\text{Var}}[\widehat{\boldsymbol{\beta}}_{(U)}])}{\det_{ns}(\widehat{\text{Var}}[\widehat{\boldsymbol{\beta}}])}$$

where $\det_{ns}(\mathbf{M})$ denotes the determinant of the nonsingular part of matrix $\mathbf{M}$.

In the case of iterative influence analysis these statistics are also computed for the covariance parameter estimates. If q denotes the rank of $\text{Var}[\widehat{\boldsymbol{\theta}}]$, then

$$\text{CovTrace}(\boldsymbol{\theta}) = |\text{trace}(\widehat{\text{Var}}[\widehat{\boldsymbol{\theta}}]^{-}\ \widehat{\text{Var}}[\widehat{\boldsymbol{\theta}}_{(U)}]) - q|$$

$$\text{CovRatio}(\boldsymbol{\theta}) = \frac{\det_{ns}(\widehat{\text{Var}}[\widehat{\boldsymbol{\theta}}_{(U)}])}{\det_{ns}(\widehat{\text{Var}}[\widehat{\boldsymbol{\theta}}])}$$

Likelihood Distances

The log-likelihood function l and restricted log-likelihood function l_R of the linear mixed model are given in the section "Estimating Covariance Parameters in the Mixed Model" on page 3974. Denote as $\boldsymbol{\psi}$ the collection of all parameters, i.e., the fixed effects $\boldsymbol{\beta}$ and the covariance parameters $\boldsymbol{\theta}$. Twice the difference between the (restricted) log-likelihood evaluated at the full-data estimates $\widehat{\boldsymbol{\psi}}$ and at the reduced-data estimates $\widehat{\boldsymbol{\psi}}_{(U)}$ is known as the (restricted) likelihood distance:

$$RLD_{(U)} = 2\{l_R(\widehat{\boldsymbol{\psi}}) - l_R(\widehat{\boldsymbol{\psi}}_{(U)})\}$$

$$LD_{(U)} = 2\{l(\widehat{\boldsymbol{\psi}}) - l(\widehat{\boldsymbol{\psi}}_{(U)})\}$$

Cook and Weisberg (1982, Ch. 5.2) refer to these differences as *likelihood distances*, Beckman, Nachtsheim, and Cook (1987) call the measures *likelihood displacements*. If the number of elements in $\boldsymbol{\psi}$ that are subject to updating following point removal is q, then likelihood displacements can be compared against cutoffs from a chi-square distribution with q degrees of freedom. Notice that this reference distribution does not depend on the number of observations removed from the analysis, but rather on the number of model parameters that are updated. The likelihood displacement gives twice the amount by which the log likelihood of the full data changes if one were to use an estimate based on fewer data points. It is thus a global, summary measure of the influence of the observations in U jointly on all parameters.

Unless METHOD=ML, the MIXED procedure computes the likelihood displacement based on the residual (=restricted) log likelihood, even if METHOD=MIVQUE0 or METHOD=TYPE1, TYPE2, or TYPE3.

Noniterative Update Formulas

Update formulas that do not require refitting of the model are available for the cases where $\mathbf{V} = \sigma^2\mathbf{I}$, $\mathbf{V}$ is known, or $\mathbf{V}^*$ is known. When ITER=0 and these update formulas can be invoked, the MIXED procedure uses the computational devices that are outlined in the following paragraphs. It is then assumed that the variance-covariance matrix of the fixed effects has the form $(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-}$. When DDFM=KENWARDROGER, this is not the case; the estimated variance-covariance matrix is then inflated to better represent the uncertainty in the estimated covariance parameters. Influence statistics when DDFM=KENWARDROGER should iteratively update the covariance parameters (ITER > 0). The dependence of $\mathbf{V}$ on $\boldsymbol{\theta}$ is suppressed in the sequel for brevity.

Updating the Fixed Effects Denote by $\mathbf{U}$ the $(n \times k)$ matrix that is assembled from k columns of the identity matrix. Each column of $\mathbf{U}$ corresponds to the removal of one data point. The point being targeted by the ith column of $\mathbf{U}$ corresponds to the row in which a 1 appears. Furthermore,

define

$$\boldsymbol{\Omega} = (\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-}$$
$$\mathbf{Q} = \mathbf{X}\boldsymbol{\Omega}\mathbf{X}'$$
$$\mathbf{P} = \mathbf{V}^{-1}(\mathbf{V} - \mathbf{Q})\mathbf{V}^{-1}$$

The change in the fixed-effects estimates following removal of the observations in U is

$$\widehat{\boldsymbol{\beta}} - \widehat{\boldsymbol{\beta}}_{(U)} = \boldsymbol{\Omega}\mathbf{X}'\mathbf{V}^{-1}\mathbf{U}(\mathbf{U}'\mathbf{P}\mathbf{U})^{-1}\mathbf{U}'\mathbf{V}^{-1}(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})$$

Using results in Cook and Weisberg (1982, A2) you can further compute

$$\widetilde{\boldsymbol{\Omega}} = (\mathbf{X}'_{(U)}\mathbf{V}^{-1}_{(U)}\mathbf{X}_{(U)})^{-} = \boldsymbol{\Omega} + \boldsymbol{\Omega}\mathbf{X}'\mathbf{V}^{-1}\mathbf{U}(\mathbf{U}'\mathbf{P}\mathbf{U})^{-1}\mathbf{U}'\mathbf{V}^{-1}\mathbf{X}\boldsymbol{\Omega}$$

If $\mathbf{X}$ is $(n \times p)$ of rank $m < p$, then $\boldsymbol{\Omega}$ is deficient in rank and the MIXED procedure computes needed quantities in $\widetilde{\boldsymbol{\Omega}}$ by sweeping (Goodnight 1979). If the rank of the $(k \times k)$ matrix $\mathbf{U}'\mathbf{P}\mathbf{U}$ is less than k, the removal of the observations introduces a new singularity, whether $\mathbf{X}$ is of full rank or not. The solution vectors $\widehat{\boldsymbol{\beta}}$ and $\widehat{\boldsymbol{\beta}}_{(U)}$ then do not have the same expected values and should not be compared. When the MIXED procedure encounters this situation, influence diagnostics that depend on the choice of generalized inverse are not computed. The procedure also monitors the singularity criteria when sweeping the rows of $(\mathbf{X}'\mathbf{V}^{-1}\mathbf{X})^{-}$ and of $(\mathbf{X}'_{(U)}\mathbf{V}^{-1}_{(U)}\mathbf{X}_{(U)})^{-}$. If a new singularity is encountered or a former singularity disappears, no influence statistics are computed.

Residual Variance When σ^2 is profiled out of the marginal variance-covariance matrix, a closed-form estimate of σ^2 that is based on only the remaining observations can be computed provided $\mathbf{V}^* = \mathbf{V}(\widehat{\boldsymbol{\theta}}^*)$ is known. Hurtado (1993, Thm. 5.2) shows that

$$(n - q - r)\widehat{\sigma}^2_{(U)} = (n - q)\widehat{\sigma}^2 - \widehat{\boldsymbol{\epsilon}}'_U(\widehat{\sigma}^2\mathbf{U}'\mathbf{P}\mathbf{U})^{-1}\widehat{\boldsymbol{\epsilon}}_U$$

and $\widehat{\boldsymbol{\epsilon}}_U = \mathbf{U}'\mathbf{V}^{*-1}(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})$. In the case of maximum likelihood estimation $q = 0$ and for REML estimation $q = \text{rank}(\mathbf{X})$. The constant r equals the rank of $(\mathbf{U}'\mathbf{P}\mathbf{U})$ for REML estimation and the number of effective observations that are removed if METHOD=ML.

Likelihood Distances For noniterative methods the following computational devices are used to compute (restricted) likelihood distances provided that the residual variance σ^2 is profiled.

The log likelihood function $l(\widehat{\boldsymbol{\theta}})$ evaluated at the full-data and reduced-data estimates can be written as

$$l(\widehat{\boldsymbol{\psi}}) = -\frac{n}{2}\log(\widehat{\sigma}^2) - \frac{1}{2}\log|\mathbf{V}^*| - \frac{1}{2}(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})'\mathbf{V}^{*-1}(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}})/\widehat{\sigma}^2 - \frac{n}{2}\log(2\pi)$$
$$l(\widehat{\boldsymbol{\psi}}_{(U)}) = -\frac{n}{2}\log(\widehat{\sigma}^2_{(U)}) - \frac{1}{2}\log|\mathbf{V}^*| - \frac{1}{2}(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}_{(U)})'\mathbf{V}^{*-1}(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}_{(U)})/\widehat{\sigma}^2_{(U)} - \frac{n}{2}\log(2\pi)$$

Notice that $l(\widehat{\boldsymbol{\theta}}_{(U)})$ evaluates the log likelihood for n data points at the reduced-data estimates. It is not the log likelihood obtained by fitting the model to the reduced data. The likelihood distance is then

$$LD_{(U)} = n\log\left\{\frac{\widehat{\sigma}^2_{(U)}}{\widehat{\sigma}^2}\right\} - n + \left(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}_{(U)}\right)'\mathbf{V}^{*-1}\left(\mathbf{y} - \mathbf{X}\widehat{\boldsymbol{\beta}}_{(U)}\right)/\widehat{\sigma}^2_{(U)}$$

Expressions for $RLD_{(U)}$ in noniterative influence analysis are derived along the same lines.

Default Output

The following sections describe the output PROC MIXED produces by default. This output is organized into various tables, and they are discussed in order of appearance.

Model Information

The "Model Information" table describes the model, some of the variables it involves, and the method used in fitting it. It also lists the method (profile, factor, parameter, or none) for handling the residual variance in the model. The *profile* method concentrates the residual variance out of the optimization problem, whereas the *parameter* method retains it as a parameter in the optimization. The *factor* method keeps the residual fixed, and *none* is displayed when a residual variance is not part of the model.

The "Model Information" table also has a row labeled Fixed Effects SE Method. This row describes the method used to compute the approximate standard errors for the fixed-effects parameter estimates and related functions of them. The two possibilities for this row are Model-Based, which is the default method, and Empirical, which results from using the EMPIRICAL option in the PROC MIXED statement.

For ODS purposes, the name of the "Model Information" table is "ModelInfo."

Class Level Information

The "Class Level Information" table lists the levels of every variable specified in the CLASS statement. You should check this information to make sure the data are correct. You can adjust the order of the CLASS variable levels with the ORDER= option in the PROC MIXED statement. For ODS purposes, the name of the "Class Level Information" table is "ClassLevels."

Dimensions

The "Dimensions" table lists the sizes of relevant matrices. This table can be useful in determining CPU time and memory requirements. For ODS purposes, the name of the "Dimensions" table is "Dimensions."

Number of Observations

The "Number of Observations" table shows the number of observations read from the data set and the number of observations used in fitting the model.

Iteration History

The "Iteration History" table describes the optimization of the residual log likelihood or log likelihood. The function to be minimized (the *objective function*) is $-2l$ for ML and $-2l_R$ for REML; the column name of the objective function in the "Iteration History" table is "-2 Log Like" for ML and "-2 Res Log Like" for REML. The minimization is performed by using a ridge-stabilized Newton-Raphson algorithm, and the rows of this table describe the iterations that this algorithm takes in order to minimize the objective function.

The Evaluations column of the "Iteration History" table tells how many times the objective function is evaluated during each iteration.

The Criterion column of the "Iteration History" table is, by default, a relative Hessian convergence quantity given by

$$\frac{\mathbf{g}_k' \mathbf{H}_k^{-1} \mathbf{g}_k}{|f_k|}$$

where f_k is the value of the objective function at iteration k, $\mathbf{g}_k$ is the gradient (first derivative) of f_k, and $\mathbf{H}_k$ is the Hessian (second derivative) of f_k. If $\mathbf{H}_k$ is singular, then PROC MIXED uses the following relative quantity:

$$\frac{\mathbf{g}_k' \mathbf{g}_k}{|f_k|}$$

To prevent the division by $|f_k|$, use the ABSOLUTE option in the PROC MIXED statement. To use a relative function or gradient criterion, use the CONVF or CONVG option, respectively.

The Hessian criterion is considered superior to function and gradient criteria because it measures orthogonality rather than lack of progress (Bates and Watts 1988). Provided the initial estimate is feasible and the maximum number of iterations is not exceeded, the Newton-Raphson algorithm is considered to have converged when the criterion is less than the tolerance specified with the CONVF, CONVG, or CONVH option in the PROC MIXED statement. The default tolerance is 1E−8. If convergence is not achieved, PROC MIXED displays the estimates of the parameters at the last iteration.

A convergence criterion that is missing indicates that a boundary constraint has been dropped; it is usually not a cause for concern.

If you specify the ITDETAILS option in the PROC MIXED statement, then the covariance parameter estimates at each iteration are included as additional columns in the "Iteration History" table.

For ODS purposes, the name of the "Iteration History" table is "IterHistory."

Convergence Status

The "Convergence Status" table informs about the status of the iterative estimation process at the end of the Newton-Raphson optimization. It appears as a message in the listing, and this message is repeated in the log. The ODS object "ConvergenceStatus" also contains several nonprinting columns that can be helpful in checking the success of the iterative process, in particular during

batch processing or when analyzing BY groups. The Status variable takes on the value 0 for a successful convergence (even if the Hessian matrix might not be positive definite). The values 1 and 2 of the Status variable indicate lack of convergence and infeasible initial parameter values, respectively. The variables pdG and pdH can be used to check whether the **G** and **R** matrices are positive definite.

For models that are not fit iteratively, such as models without random effects or when the NOITER option is in effect, the "Convergence Status" is not produced.

Covariance Parameter Estimates

The "Covariance Parameter Estimates" table contains the estimates of the parameters in **G** and **R** (see the section "Estimating Covariance Parameters in the Mixed Model" on page 3974). Their values are labeled in the table along with Subject and Group information if applicable. The estimates are displayed in the Estimate column and are the results of one of the following estimation methods: REML, ML, MIVQUE0, SSCP, Type1, Type2, or Type3.

If you specify the RATIO option in the PROC MIXED statement, the Ratio column is added to the table listing the ratio of each parameter estimate to that of the residual variance.

Specifying the COVTEST option in the PROC MIXED statement produces the "Std Error," "Z Value," and "Pr Z" columns. The "Std Error" column contains the approximate standard errors of the covariance parameter estimates. These are the square roots of the diagonal elements of the observed inverse Fisher information matrix, which equals $2\mathbf{H}^{-1}$, where **H** is the Hessian matrix. The **H** matrix consists of the second derivatives of the objective function with respect to the covariance parameters; see Wolfinger, Tobias, and Sall (1994) for formulas. When you use the SCORING= option and PROC MIXED converges without stopping the scoring algorithm, PROC MIXED uses the expected Hessian matrix to compute the covariance matrix instead of the observed Hessian. The observed or expected inverse Fisher information matrix can be viewed as an asymptotic covariance matrix of the estimates.

The "Z Value" column is the estimate divided by its approximate standard error, and the "Pr Z" column is the one- or two-tailed area of the standard Gaussian density outside of the Z-value. The MIXED procedure computes one-sided p-values for the residual variance and for covariance parameters with a lower bound of 0. The procedure computes two-sided p-values otherwise. These statistics constitute Wald tests of the covariance parameters, and they are valid only asymptotically.

CAUTION: Wald tests can be unreliable in small samples.

For ODS purposes, the name of the "Covariance Parameter Estimates" table is "CovParms."

Fit Statistics

The "Fit Statistics" table provides some statistics about the estimated mixed model. Expressions for the -2 times the log likelihood are provided in the section "Estimating Covariance Parameters in the Mixed Model" on page 3974. If the log likelihood is an extremely large number, then PROC MIXED has deemed the estimated **V** matrix to be singular. In this case, all subsequent results should be viewed with caution.

In addition, the "Fit Statistics" table lists three information criteria: AIC, AICC, and BIC, all in smaller-is-better form. Expressions for these criteria are described under the IC option.

For ODS purposes, the name of the "Model Fitting Information" table is "FitStatistics."

Null Model Likelihood Ratio Test

If one covariance model is a submodel of another, you can carry out a likelihood ratio test for the significance of the more general model by computing -2 times the difference between their log likelihoods. Then compare this statistic to the χ^2 distribution with degrees of freedom equal to the difference in the number of parameters for the two models.

This test is reported in the "Null Model Likelihood Ratio Test" table to determine whether it is necessary to model the covariance structure of the data at all. The "Chi-Square" value is -2 times the log likelihood from the null model minus -2 times the log likelihood from the fitted model, where the null model is the one with only the fixed effects listed in the MODEL statement and $\mathbf{R} = \sigma^2\mathbf{I}$. This statistic has an asymptotic χ^2 distribution with $q-1$ degrees of freedom, where q is the effective number of covariance parameters (those not estimated to be on a boundary constraint). The "Pr > ChiSq" column contains the upper-tail area from this distribution. This p-value can be used to assess the significance of the model fit.

This test is not produced for cases where the null hypothesis lies on the boundary of the parameter space, which is typically for variance component models. This is because the standard asymptotic theory does not apply in this case (Self and Liang 1987, Case 5).

If you specify a PARMS statement, PROC MIXED constructs a likelihood ratio test between the best model from the grid search and the final fitted model and reports the results in the "Parameter Search" table.

For ODS purposes, the name of the "Null Model Likelihood Ratio Test" table is "LRT."

Type 3 Tests of Fixed Effects

The "Type 3 Tests of Fixed Effects" table contains hypothesis tests for the significance of each of the fixed effects—that is, those effects you specify in the MODEL statement. By default, PROC MIXED computes these tests by first constructing a Type 3 $\mathbf{L}$ matrix (see Chapter 15, "The Four Types of Estimable Functions") for each effect. This $\mathbf{L}$ matrix is then used to compute the following F statistic:

$$F = \frac{\widehat{\boldsymbol{\beta}}'\mathbf{L}'[\mathbf{L}(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}\mathbf{L}']^{-}\mathbf{L}\widehat{\boldsymbol{\beta}}}{r}$$

where $r = \text{rank}(\mathbf{L}(\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X})^{-}\mathbf{L}')$. A p-value for the test is computed as the tail area beyond this statistic from an F distribution with NDF and DDF degrees of freedom. The numerator degrees of freedom (NDF) are the row rank of $\mathbf{L}$, and the denominator degrees of freedom are computed by using one of the methods described under the DDFM= option. Small values of the p-value (typically less than 0.05 or 0.01) indicate a significant effect.

You can use the HTYPE= option in the MODEL statement to obtain tables of Type 1 (sequential) tests and Type 2 (adjusted) tests in addition to or instead of the table of Type 3 (partial) tests.

You can use the CHISQ option in the MODEL statement to obtain Wald χ^2 tests of the fixed effects. These are carried out by using the numerator of the F statistic and comparing it with the χ^2 distribution with NDF degrees of freedom. It is more liberal than the F test because it effectively assumes infinite denominator degrees of freedom.

For ODS purposes, the names of the "Type 1 Tests of Fixed Effects" through the "Type 3 Tests of Fixed Effects" tables are "Tests1" through "Tests3," respectively.

ODS Table Names

Each table created by PROC MIXED has a name associated with it, and you must use this name to reference the table when using ODS statements. These names are listed in Table 56.22.

Table 56.22 ODS Tables Produced by PROC MIXED

Table Name	Description	Required Statement / Option
AccRates	acceptance rates for posterior sampling	PRIOR
AsyCorr	asymptotic correlation matrix of covariance parameters	PROC MIXED ASYCORR
AsyCov	asymptotic covariance matrix of covariance parameters	PROC MIXED ASYCOV
Base	base densities used for posterior sampling	PRIOR
Bound	computed bound for posterior rejection sampling	PRIOR
CholG	Cholesky root of the estimated **G** matrix	RANDOM / GC
CholR	Cholesky root of blocks of the estimated **R** matrix	REPEATED / RC
CholV	Cholesky root of blocks of the estimated **V** matrix	RANDOM / VC
ClassLevels	level information from the CLASS statement	default output
Coef	**L** matrix coefficients	E option in MODEL, CONTRAST, ESTIMATE, or LSMEANS
Contrasts	results from the CONTRAST statements	CONTRAST
ConvergenceStatus	convergence status	default
CorrB	approximate correlation matrix of fixed-effects parameter estimates	MODEL / CORRB
CovB	approximate covariance matrix of fixed-effects parameter estimates	MODEL / COVB
CovParms	estimated covariance parameters	default output
Diffs	differences of LS-means	LSMEANS / DIFF (or PDIFF)

Table 56.22 *continued*

Table Name	Description	Required Statement / Option
Dimensions	dimensions of the model	default output
Estimates	results from ESTIMATE statements	ESTIMATE
FitStatistics	fit statistics	default
G	estimated **G** matrix	RANDOM / G
GCorr	correlation matrix from the estimated **G** matrix	RANDOM / GCORR
HLM1	Type 1 Hotelling-Lawley-McKeon tests of fixed effects	MODEL / HTYPE=1 and REPEATED / HLM TYPE=UN
HLM2	Type 2 Hotelling-Lawley-McKeon tests of fixed effects	MODEL / HTYPE=2 and REPEATED / HLM TYPE=UN
HLM3	Type 3 Hotelling-Lawley-McKeon tests of fixed effects	REPEATED / HLM TYPE=UN
HLPS1	Type 1 Hotelling-Lawley-Pillai-Samson tests of fixed effects	MODEL / HTYPE=1 and REPEATED / HLPS TYPE=UN
HLPS2	Type 2 Hotelling-Lawley-Pillai-Samson tests of fixed effects	MODEL / HTYPE=1 and REPEATED / HLPS TYPE=UN
HLPS3	Type 3 Hotelling-Lawley-Pillai-Samson tests of fixed effects	REPEATED / HLPS TYPE=UN
Influence	influence diagnostics	MODEL / INFLUENCE
InfoCrit	information criteria	PROC MIXED IC
InvCholG	inverse Cholesky root of the estimated **G** matrix	RANDOM / GCI
InvCholR	inverse Cholesky root of blocks of the estimated **R** matrix	REPEATED / RCI
InvCholV	inverse Cholesky root of blocks of the estimated **V** matrix	RANDOM / VCI
InvCovB	inverse of approximate covariance matrix of fixed-effects parameter estimates	MODEL / COVBI
InvG	inverse of the estimated **G** matrix	RANDOM / GI
InvR	inverse of blocks of the estimated **R** matrix	REPEATED / RI
InvV	inverse of blocks of the estimated **V** matrix	RANDOM / VI
IterHistory	iteration history	default output
LComponents	single-degree-of-freedom estimates corresponding to rows of the **L** matrix for fixed effects	MODEL / LCOMPONENTS
LRT	likelihood ratio test	default output
LSMeans	LS-means	LSMEANS
MMEq	mixed model equations	PROC MIXED MMEQ
MMEqSol	mixed model equations solution	PROC MIXED MMEQSOL
ModelInfo	model information	default output

Table 56.22 *continued*

Table Name	Description	Required Statement / Option
NObs	number of observations read and used	default output
ParmSearch	parameter search values	PARMS
Posterior	posterior sampling information	PRIOR
R	blocks of the estimated **R** matrix	REPEATED / R
RCorr	correlation matrix from blocks of the estimated **R** matrix	REPEATED / RCORR
Search	posterior density search table	PRIOR / PSEARCH
Slices	tests of LS-means slices	LSMEANS / SLICE=
SolutionF	fixed-effects solution vector	MODEL / S
SolutionR	random-effects solution vector	RANDOM / S
Tests1	Type 1 tests of fixed effects	MODEL / HTYPE=1
Tests2	Type 2 tests of fixed effects	MODEL / HTYPE=2
Tests3	Type 3 tests of fixed effects	default output
Type1	Type 1 analysis of variance	PROC MIXED METHOD=TYPE1
Type2	Type 2 analysis of variance	PROC MIXED METHOD=TYPE2
Type3	Type 3 analysis of variance	PROC MIXED METHOD=TYPE3
Trans	transformation of covariance parameters	PRIOR / PTRANS
V	blocks of the estimated **V** matrix	RANDOM / V
VCorr	correlation matrix from blocks of the estimated **V** matrix	RANDOM / VCORR

In Table 56.22, "Coef" refers to multiple tables produced by the E, E1, E2, or E3 option in the MODEL statement and the E option in the CONTRAST, ESTIMATE, and LSMEANS statements. You can create one large data set of these tables with a statement similar to the following:

```
ods output Coef=c;
```

To create separate data sets, use the following statement:

```
ods output Coef(match_all)=c;
```

Here the resulting data sets are named C, C1, C2, etc. The same principles apply to data sets created from the "R," "CholR," "InvCholR," "RCorr," "InvR," "V," "CholV," "InvCholV," "VCorr," and "InvV" tables.

In Table 56.22, the following changes have occurred from SAS 6. The "Predicted," "PredMeans," and "Sample" tables from SAS 6 no longer exist and have been replaced by output data sets; see descriptions of the MODEL statement options OUTPRED= and OUTPREDM= and the PRIOR statement option OUT= for more details. The "ML" and "REML" tables from SAS 6 have been replaced by the "IterHistory" table. The "Tests," "HLM," and "HLPS" tables from SAS 6 have been renamed "Tests3," "HLM3," and "HLPS3," respectively.

Table 56.23 lists the variable names associated with the data sets created when you use the ODS OUTPUT option in conjunction with the preceding tables. In Table 56.23, *n* is used to denote a

generic number that depends on the particular data set and model you select, and it can assume a different value each time it is used (even within the same table). The phrase *model specific* appears in rows of the affected tables to indicate that columns in these tables depend on the variables you specify in the model.

CAUTION: There is a danger of name collisions with the variables in the *model specific* tables in Table 56.23 and variables in your input data set. You should avoid using input variables with the same names as the variables in these tables.

Table 56.23 Variable Names for the ODS Tables Produced in PROC MIXED

Table Name	Variables
AsyCorr	Row, CovParm, CovP1–CovP*n*
AsyCov	Row, CovParm, CovP1–CovP*n*
BaseDen	Type, Parm1–Parm*n*
Bound	Technique, Converge, Iterations, Evaluations, LogBound, CovP1–CovP*n*, TCovP1–TCovP*n*
CholG	*model specific*, Effect, Subject, Sub1–Sub*n*, Group, Group1–Group*n*, Row, Col1–Col*n*
CholR	Index, Row, Col1–Col*n*
CholV	Index, Row, Col1–Col*n*
ClassLevels	Class, Levels, Values
Coef	*model specific*, LMatrix, Effect, Subject, Sub1–Sub*n*, Group, Group1–Group*n*, Row1–Row*n*
Contrasts	Label, NumDF, DenDF, ChiSquare, FValue, ProbChiSq, ProbF
CorrB	*model specific*, Effect, Row, Col1–Col*n*
CovB	*model specific*, Effect, Row, Col1–Col*n*
CovParms	CovParm, Subject, Group, Estimate, StandardError, ZValue, ProbZ, Alpha, Lower, Upper
Diffs	*model specific*, Effect, Margins, ByLevel, AT variables, Diff, StandardError, DF, tValue, Tails, Probt, Adjustment, Adjp, Alpha, Lower, Upper, AdjLow, AdjUpp
Dimensions	Descr, Value
Estimates	Label, Estimate, StandardError, DF, tValue, Tails, Probt, Alpha, Lower, Upper
FitStatistics	Descr, Value
G	*model specific*, Effect, Subject, Sub1–Sub*n*, Group, Group1–Group*n*, Row, Col1–Col*n*
GCorr	*model specific*, Effect, Subject, Sub1–Sub*n*, Group, Group1–Group*n*, Row, Col1–Col*n*
HLM1	Effect, NumDF, DenDF, FValue, ProbF
HLM2	Effect, NumDF, DenDF, FValue, ProbF
HLM3	Effect, NumDF, DenDF, FValue, ProbF
HLPS1	Effect, NumDF, DenDF, FValue, ProbF
HLPS2	Effect, NumDF, DenDF, FValue, ProbF
HLPS3	Effect, NumDF, DenDF, FValue, ProbF

Table 56.23 *continued*

Table Name	Variables
Influence	*dependent on option modifiers*, Effect, Tuple, Obs1–Obs*k*, Level, Iter, Index, Predicted, Residual, Leverage, PressRes, PRESS, Student, RMSE, RStudent, CookD, DFFITS, MDFFITS, CovRatio, CovTrace, CookDCP, MDFFITSCP, CovRatioCP, CovTraceCP, LD, RLD, Parm1–Parm*p*, CovP1–CovP*q*, Notes
InfoCrit	Neg2LogLike, Parms, AIC, AICC, HQIC, BIC, CAIC
InvCholG	*model specific*, Effect, Subject, Sub1–Sub*n*, Group, Group1–Group*n*, Row, Col1–Col*n*
InvCholR	Index, Row, Col1–Col*n*
InvCholV	Index, Row, Col1–Col*n*
InvCovB	*model specific*, Effect, Row, Col1–Col*n*
InvG	*model specific*, Effect, Subject, Sub1–Sub*n*, Group, Group1–Group*n*, Row, Col1–Col*n*
InvR	Index, Row, Col1–Col*n*
InvV	Index, Row, Col1–Col*n*
IterHistory	CovP1–CovP*n*, Iteration, Evaluations, M2ResLogLike, M2LogLike, Criterion
LComponents	Effect, TestType, LIndex, Estimate, StdErr, DF, tValue, Probt
LRT	DF, ChiSquare, ProbChiSq
LSMeans	*model specific*, Effect, Margins, ByLevel, AT variables, Estimate, StandardError, DF, tValue, Probt, Alpha, Lower, Upper, Cov1–Cov*n*, Corr1–Corr*n*
MMEq	*model specific*, Effect, Subject, Sub1–Sub*n*, Group, Group1–Group*n*, Row, Col1–Col*n*
MMEqSol	*model specific*, Effect, Subject, Sub1–Sub*n*, Group, Group1–Group*n*, Row, Col1–Col*n*
ModelInfo	Descr, Value
Nobs	Label, N, NObsRead, NObsUsed, SumFreqsRead, SumFreqsUsed
ParmSearch	CovP1–CovP*n*, Var, ResLogLike, M2ResLogLike2, LogLike, M2LogLike, LogDetH
Posterior	Descr, Value
R	Index, Row, Col1–Col*n*
RCorr	Index, Row, Col1–Col*n*
Search	Parm, TCovP1–TCovP*n*, Posterior
Slices	*model specific*, Effect, Margins, ByLevel, AT variables, NumDF, DenDF, FValue, ProbF
SolutionF	*model specific*, Effect, Estimate, StandardError, DF, tValue, Probt, Alpha, Lower, Upper
SolutionR	*model specific*, Effect, Subject, Sub1–Sub*n*, Group, Group1–Group*n*, Estimate, StdErrPred, DF, tValue, Probt, Alpha, Lower, Upper
Tests1	Effect, NumDF, DenDF, ChiSquare, FValue, ProbChiSq, ProbF
Tests2	Effect, NumDF, DenDF, ChiSquare, FValue, ProbChiSq, ProbF
Tests3	Effect, NumDF, DenDF, ChiSquare, FValue, ProbChiSq, ProbF
Type1	Source, DF, SS, MS, EMS, ErrorTerm, ErrorDF, FValue, ProbF

Table 56.23 *continued*

Table Name	Variables
Type2	Source, DF, SS, MS, EMS, ErrorTerm, ErrorDF, FValue, ProbF
Type3	Source, DF, SS, MS, EMS, ErrorTerm, ErrorDF, FValue, ProbF
Trans	Prior, TCovP, CovP1–CovP*n*
V	Index, Row, Col1–Col*n*
VCorr	Index, Row, Col1–Col*n*

Some of the variables listed in Table 56.23 are created only when you specify certain options in the relevant PROC MIXED statements.

ODS Graphics

This section describes the use of ODS for creating diagnostic plots with the MIXED procedure. To request these graphs you must specify the ODS GRAPHICS statement and the relevant options of the PROC MIXED or MODEL statement (Table 56.24). For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS." ODS names of the various graphics are given in the section "ODS Graph Names" on page 4008.

Residual Plots

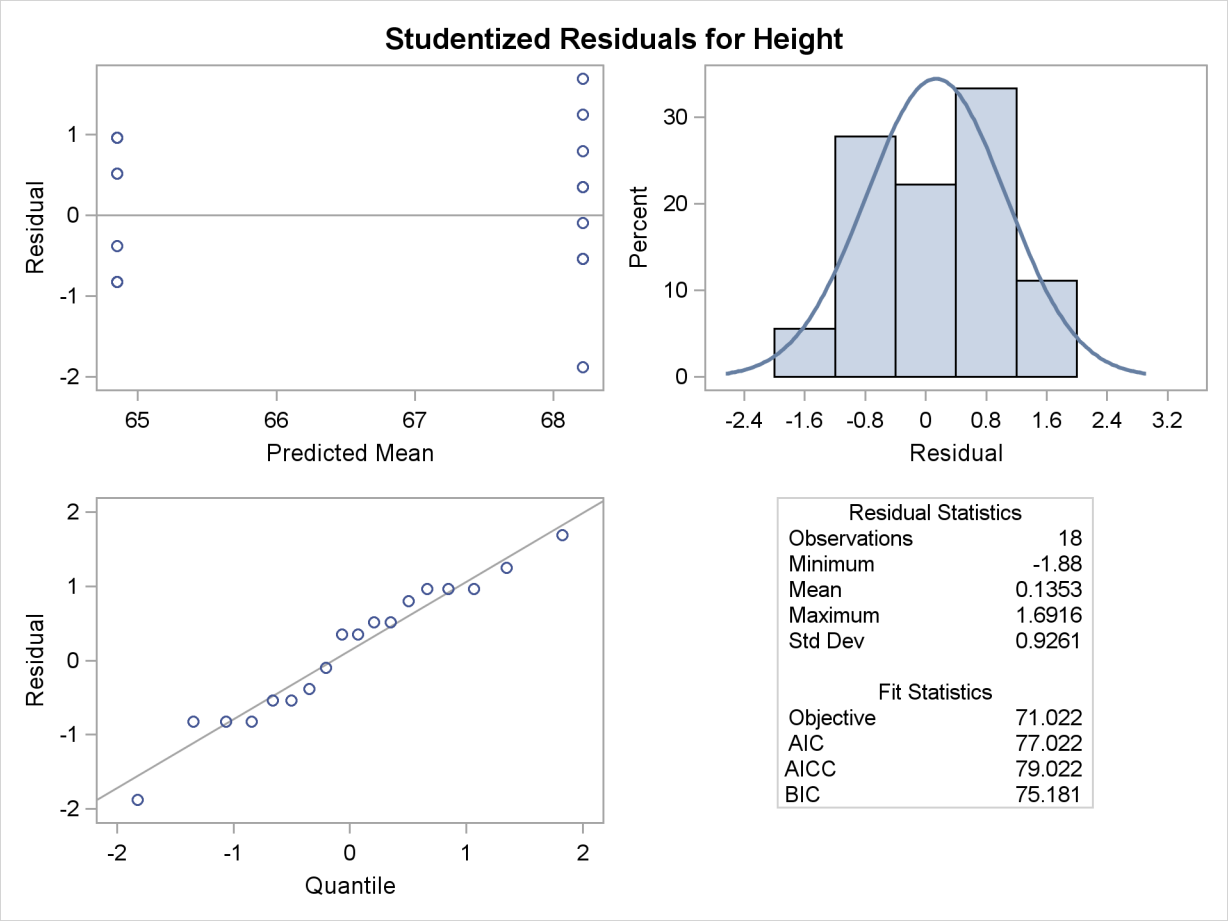
The MIXED procedure can generate panels of residual diagnostics. Each panel consists of a plot of residuals versus predicted values, a histogram with normal density overlaid, a Q-Q plot, and summary residual and fit statistics (Figure 56.15). The plots are produced even if the OUTP= and OUTPM= options in the MODEL statement are not specified. Residual panels can be generated for marginal and conditional raw, studentized, and Pearson residuals as well as for scaled residuals (see the section "Residual Diagnostics" on page 3986).

Recall the example in the section "Getting Started: MIXED Procedure" on page 3896. The following statements generate several 2×2 panels of residual graphs:

```
ods graphics on;
proc mixed data=heights;
   class Family Gender;
   model Height = Gender / residual;
   random Family Family*Gender;
run;
ods graphics off;
```

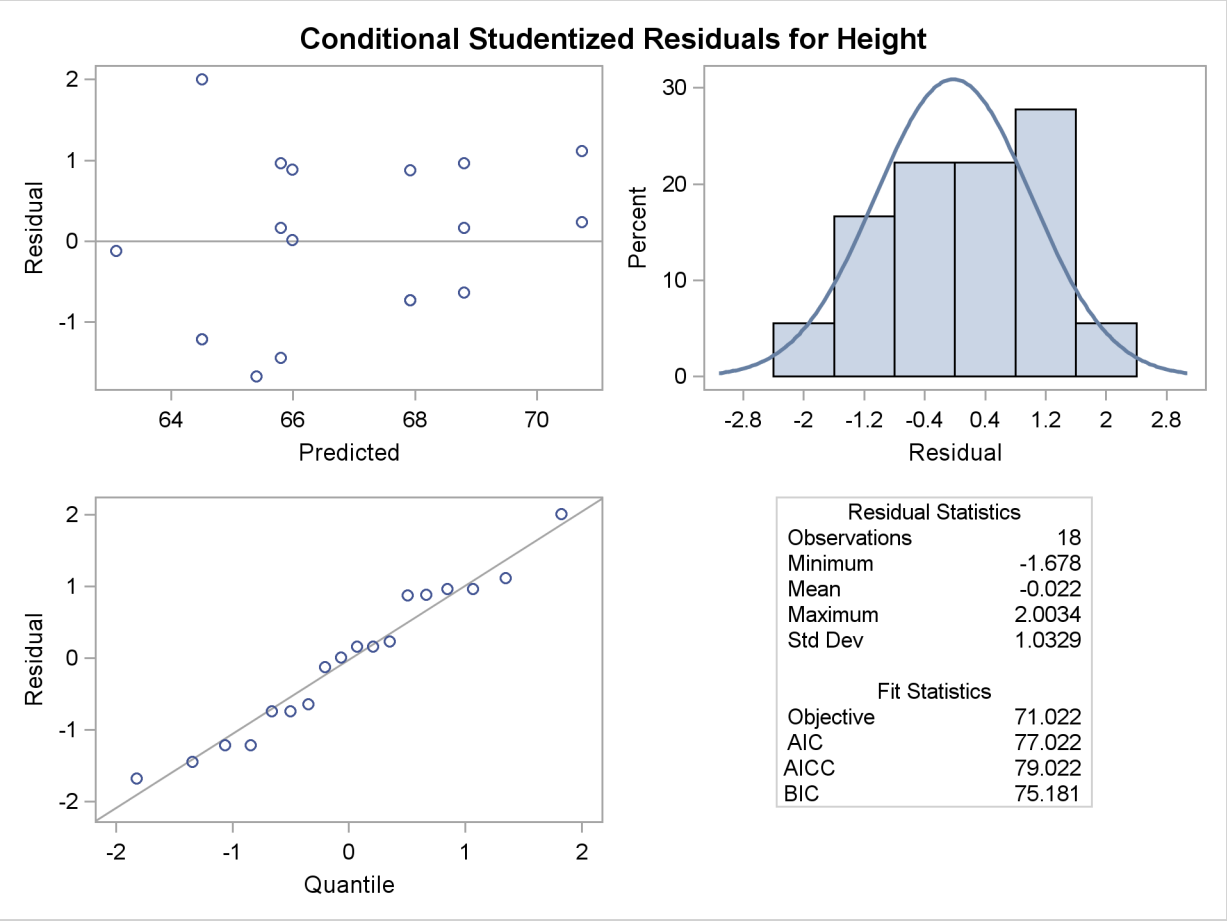
The graphical displays are requested by specifying the ODS GRAPHICS statement. The panel of the studentized marginal residuals is shown in Figure 56.15, and the panel of the studentized conditional residuals is shown in Figure 56.16.

Figure 56.15 Panel of the Studentized (Marginal) Residuals



Since the fixed-effects part of the model comprises only an intercept and the gender effect, the marginal mean takes on only two values, one for each gender. The "Residual Statistics" inset in the lower-right corner provides descriptive statistics for the set of residuals that is displayed. Note that residuals in a mixed model do not necessarily sum to zero, even if the model contains an intercept.

Figure 56.16 Panel of the Conditional Studentized Residuals



Influence Plots

The graphical features of the MIXED procedure enable you to generate plots of influence diagnostics and of deletion estimates. The type and number of plots produced depend on your modifiers of the INFLUENCE option in the MODEL statement and on the PLOTS= option in the PROC MIXED statement. Plots related to covariance parameters are produced only when diagnostics are computed by iterative methods (ITER=). The estimates of the fixed effects—and covariance parameters when updates are iterative—are plotted when you specify the ESTIMATES modifier or when you request PLOTS=INFLUENCEESTPLOT.

Two basic types of influence panels are shown in Figure 56.17 and Figure 56.18. The diagnostics panel shows Cook's D and CovRatio statistics for the fixed effects and the covariance parameters. For the SAS statements that produce these influence panels, see Example 56.8. In this example, the impact of subjects (Person) on the analysis is assessed. The Cook's D statistic measures a subject's impact on the estimates, and the CovRatio statistic measures a subject's impact on the precision of the estimates. Separate statistics are computed for the fixed effects and the covariance parameters. The CovRatio statistic has a threshold of 1.0. Values larger than 1.0 indicate that precision of the estimates is lost by exclusion of the observations in question. Values smaller than 1.0 indicate that

precision is gained by exclusion of the observations from the analysis. For example, it is evident from Output 56.17 that person 20 has considerable impact on the covariance parameter estimates and moderate influence on the fixed-effects estimates. Furthermore, exclusion of this subject from the analysis increases the precision of the covariance parameters, whereas the effect on the precision of the fixed effects is minor.

Output 56.18 shows another type of influence plot, a panel of the deletion estimates. Each plot within the panel corresponds to one of the model parameters. A reference line is drawn at the estimate based on the full data.

Figure 56.17 Influence Diagnostics

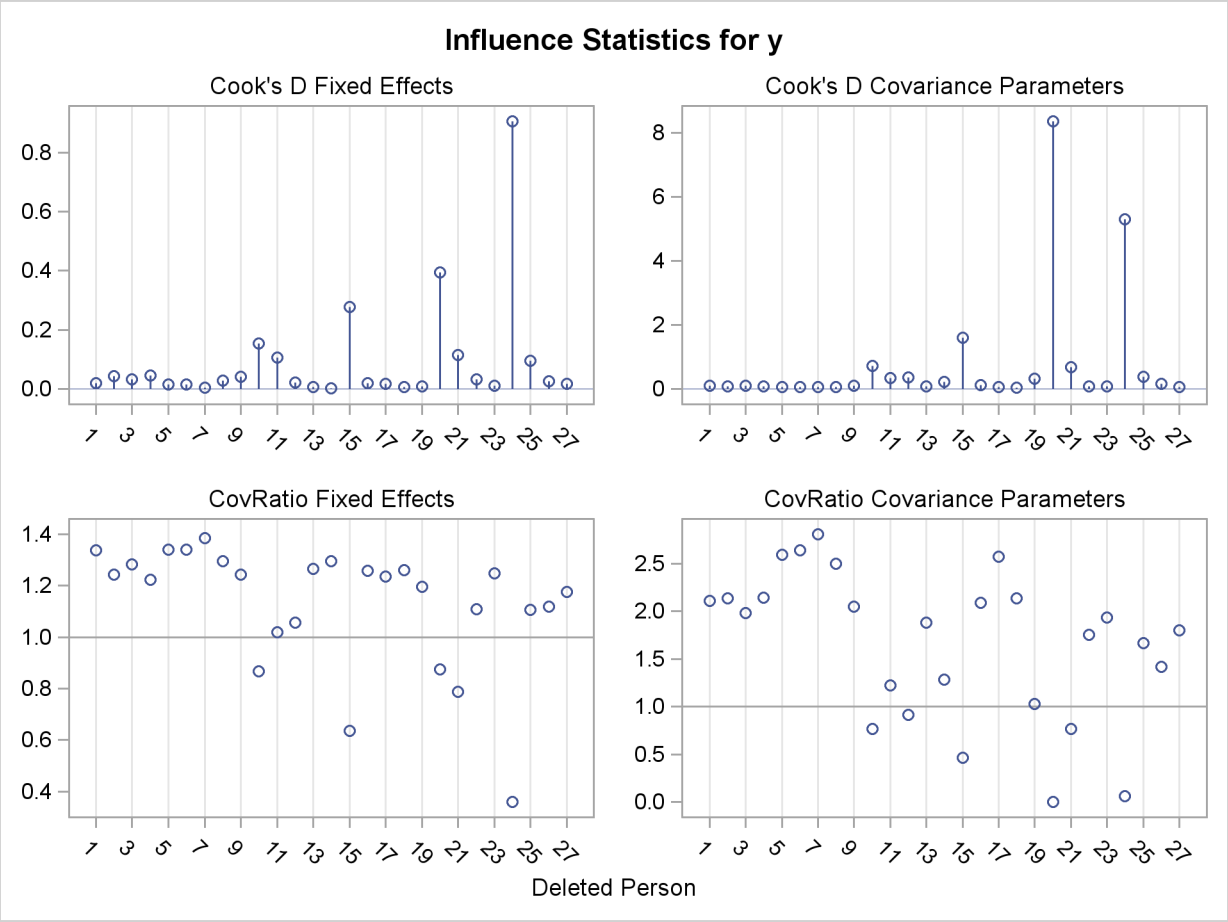
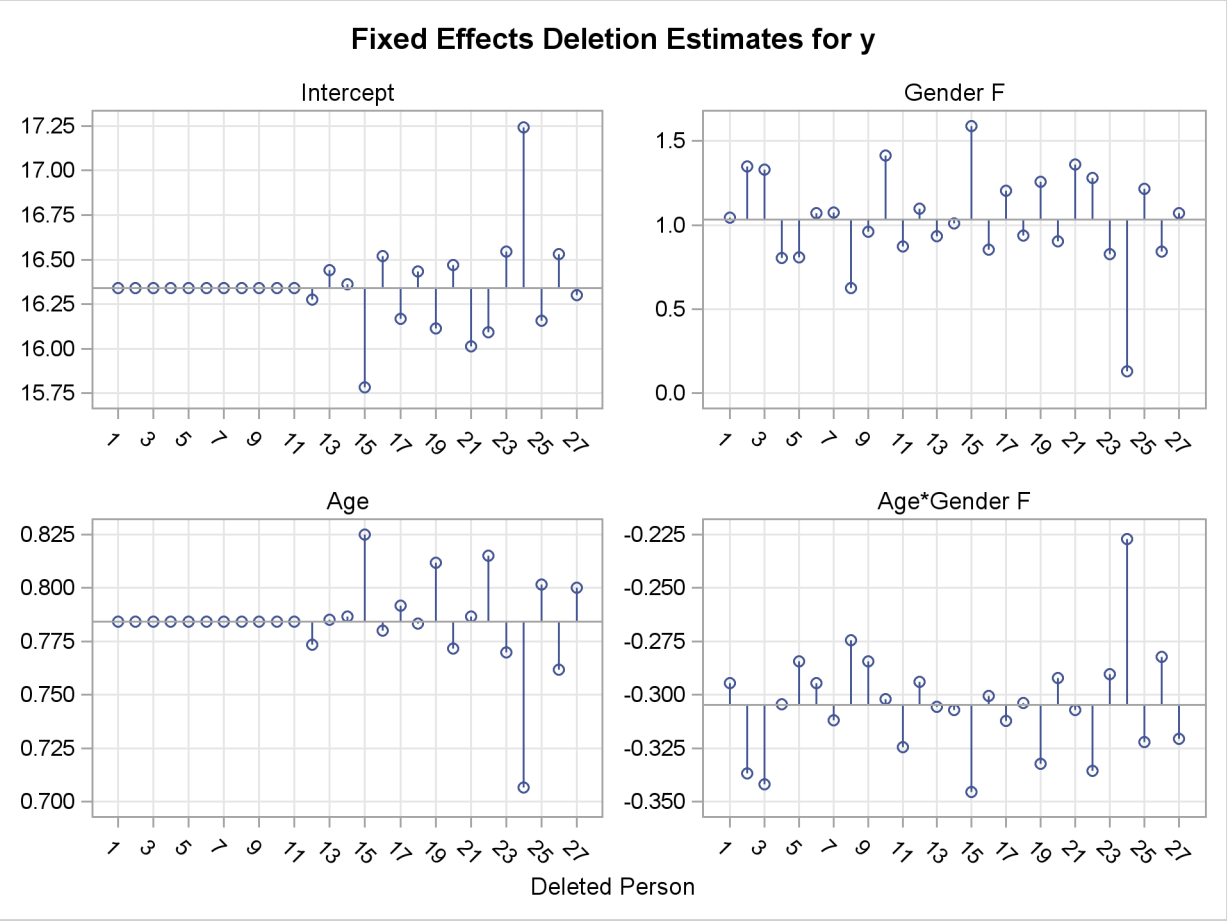


Figure 56.18 Deletion Estimates



ODS Graph Names

To request graphics with PROC MIXED, you must first enable ODS Graphics by specifying the **ODS GRAPHICS ON** statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. Some graphs are produced by default; other graphs are produced by using statements and options. You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC MIXED generates are listed in Table 56.24, along with the required statements and options.

Table 56.24 ODS Graphics Produced by PROC MIXED

ODS Graph Name	Plot Description	Statement or Option
Boxplot	Box plots	PLOTS=BOXPLOT
CovRatioPlot	CovRatio statistics for fixed effects or covariance parameters	PLOTS=INFLUENCESTATPANEL(UNPACK) and MODEL / INFLUENCE
CooksDPlot	Cook's *D* for fixed effects or covariance parameters	PLOTS=INFLUENCESTATPANEL(UNPACK) and MODEL / INFLUENCE

Table 56.24 *continued*

ODS Graph Name	Plot Description	Statement or Option
DistancePlot	Likelihood or restricted likelihood distance	MODEL / INFLUENCE
InfluenceEstPlot	Panel of deletion estimates	MODEL / INFLUENCE(EST) or PLOTS=INFLUENCEESTPLOT and MODEL / INFLUENCE
InfluenceEstPlot	Parameter estimates after removing observation or sets of observations	PLOTS=INFLUENCEESTPLOT(UNPACK) and MODEL / INFLUENCE
InfluenceStatPanel	Panel of influence statistics	MODEL / INFLUENCE
PearsonBoxPlot	Box plot of Pearson residuals	PLOTS=PEARSONPANEL(UNPACK BOX)
PearsonByPredicted	Pearson residuals vs. predicted	PLOTS=PEARSONPANEL(UNPACK)
PearsonHistogram	Histogram of Pearson residuals	PLOTS=PEARSONPANEL(UNPACK)
PearsonPanel	Panel of Pearson residuals	MODEL / RESIDUAL
PearsonQQplot	*Q*-*Q* plot of Pearson residuals	PLOTS=PEARSONPANEL(UNPACK)
PressPlot	Plot of PRESS residuals or PRESS statistic	PLOTS=PRESS and MODEL / INFLUENCE
ResidualBoxplot	Box plot of (raw) residuals	PLOTS=RESIDUALPANEL(UNPACK BOX)
ResidualByPredicted	Residuals vs. predicted	PLOTS=RESIDUALPANEL(UNPACK)
ResidualHistogram	Histogram of raw residuals	PLOTS=RESIDUALPANEL(UNPACK)
ResidualPanel	Panel of (raw) residuals	MODEL / RESIDUAL
ResidualQQplot	*Q*-*Q* plot of raw residuals	PLOTS=RESIDUALPANEL(UNPACK)
ScaledBoxplot	Box plot of scaled residuals	PLOTS=VCIRYPANEL(UNPACK BOX)
ScaledByPredicted	Scaled residuals vs. predicted	PLOTS=VCIRYPANEL(UNPACK)
ScaledHistogram	Histogram of scaled residuals	PLOTS=VCIRYPANEL(UNPACK)
ScaledQQplot	*Q*-*Q* plot of scaled residuals	PLOTS=VCIRYPANEL(UNPACK)
StudentBoxplot	Box plot of studentized residuals	PLOTS=STUDENTPANEL(UNPACK BOX)
StudentByPredicted	Studentized residuals vs. predicted	PLOTS=STUDENTPANEL(UNPACK)
StudentHistogram	Histogram of studentized residuals	PLOTS=STUDENTPANEL(UNPACK)
StudentPanel	Panel of studentized residuals	MODEL / RESIDUAL
StudentQQplot	*Q*-*Q* plot of studentized residuals	PLOTS=STUDENTPANEL(UNPACK)
VCIRYPanel	Panel of scaled residuals	MODEL / VCIRY

Computational Issues

Computational Method

In addition to numerous matrix-multiplication routines, PROC MIXED frequently uses the sweep operator (Goodnight 1979) and the Cholesky root (Golub and Van Loan 1989). The routines perform a modified W transformation (Goodnight and Hemmerle 1979) for **G**-side likelihood calculations and a direct method for **R**-side likelihood calculations. For the Type 3 F tests, PROC MIXED uses the algorithm described in Chapter 39, "The GLM Procedure."

PROC MIXED uses a ridge-stabilized Newton-Raphson algorithm to optimize either a full (ML) or residual (REML) likelihood function. The Newton-Raphson algorithm is preferred to the EM algorithm (Lindstrom and Bates 1988). PROC MIXED profiles the likelihood with respect to the fixed effects and also with respect to the residual variance whenever it appears reasonable to do so. The residual profiling can be avoided by using the NOPROFILE option of the PROC MIXED statement. PROC MIXED uses the MIVQUE0 method (Rao 1972; Giesbrecht 1989) to compute initial values.

The likelihoods that PROC MIXED optimizes are usually well-defined continuous functions with a single optimum. The Newton-Raphson algorithm typically performs well and finds the optimum in a few iterations. It is a quadratically converging algorithm, meaning that the error of the approximation near the optimum is squared at each iteration. The quadratic convergence property is evident when the convergence criterion drops to zero by factors of 10 or more.

Table 56.25 Notation for Order Calculations

Symbol	Number
p	columns of **X**
g	columns of **Z**
N	observations
q	covariance parameters
t	maximum observations per subject
S	subjects

Using the notation from Table 56.25, the following are estimates of the computational speed of the algorithms used in PROC MIXED. For likelihood calculations, the crossproducts matrix construction is of order $N(p+g)^2$ and the sweep operations are of order $(p+g)^3$. The first derivative calculations for parameters in **G** are of order qg^3 for ML and $q(g^3+pg^2+p^2g)$ for REML. If you specify a subject effect in the RANDOM statement and if you are not using the REPEATED statement, then replace g with g/S and q with qS in these calculations. The first derivative calculations for parameters in **R** are of order $qS(t^3+gt^2+g^2t)$ for ML and $qS(t^3+(p+g)t^2+(p^2+g^2)t)$ for REML. For the second derivatives, replace q with $q(q+1)/2$ in the first derivative expressions. When you specify both **G**- and **R**-side parameters (that is, when you use both the RANDOM and REPEATED statements), then additional calculations are required of an order equal to the sum of the orders for **G** and **R**. Considerable execution times can result in this case.

For further details about the computational techniques used in PROC MIXED, see Wolfinger, Tobias, and Sall (1994).

Parameter Constraints

By default, some covariance parameters are assumed to satisfy certain boundary constraints during the Newton-Raphson algorithm. For example, variance components are constrained to be nonnegative, and autoregressive parameters are constrained to be between -1 and 1. You can remove these constraints with the NOBOUND option in the PARMS statement (or with the NOBOUND option in the PROC MIXED statement), but this can lead to estimates that produce an infinite likelihood. You can also introduce or change boundary constraints with the LOWERB= and UPPERB= options in the PARMS statement.

During the Newton-Raphson algorithm, a parameter might be set equal to one of its boundary constraints for a few iterations and then it might move away from the boundary. You see a missing value in the Criterion column of the "Iteration History" table whenever a boundary constraint is dropped.

For some data sets the final estimate of a parameter might equal one of its boundary constraints. This is usually not a cause for concern, but it might lead you to consider a different model. For instance, a variance component estimate can equal zero; in this case, you might want to drop the corresponding random effect from the model. However, be aware that changing the model in this fashion can affect degrees-of-freedom calculations.

Convergence Problems

For some data sets, the Newton-Raphson algorithm can fail to converge. Nonconvergence can result from a number of causes, including flat or ridged likelihood surfaces and ill-conditioned data.

It is also possible for PROC MIXED to converge to a point that is not the global optimum of the likelihood, although this usually occurs only with the spatial covariance structures.

If you experience convergence problems, the following points might be helpful:

- One useful tool is the PARMS statement, which lets you input initial values for the covariance parameters and performs a grid search over the likelihood surface.
- Sometimes the Newton-Raphson algorithm does not perform well when two of the covariance parameters are on a different scale—that is, when they are several orders of magnitude apart. This is because the Hessian matrix is processed jointly for the two parameters, and elements of it corresponding to one of the parameters can become close to internal tolerances in PROC MIXED. In this case, you can improve stability by rescaling the effects in the model so that the covariance parameters are on the same scale.
- Data that are extremely large or extremely small can adversely affect results because of the internal tolerances in PROC MIXED. Rescaling it can improve stability.
- For stubborn problems, you might want to specify ODS OUTPUT COVPARMS=*data-set-name* to output the "Covariance Parameter Estimates" table as a precautionary measure. That

way, if the problem does not converge, you can read the final parameter values back into a new run with the PARMSDATA= option in the PARMS statement.

- Fisher scoring can be more robust than Newton-Raphson with poor MIVQUE(0) starting values. Specifying a SCORING= value of 5 or so might help to recover from poor starting values.
- Tuning the singularity options SINGULAR=, SINGCHOL=, and SINGRES= in the MODEL statement can improve the stability of the optimization process.
- Tuning the MAXITER= and MAXFUNC= options in the PROC MIXED statement can save resources. Also, the ITDETAILS option displays the values of all the parameters at each iteration.
- Using the NOPROFILE and NOBOUND options in the PROC MIXED statement might help convergence, although they can produce unusual results.
- Although the CONVH convergence criterion usually gives the best results, you might want to try CONVF or CONVG, possibly along with the ABSOLUTE option.
- If the convergence criterion reaches a relatively small value such as 1E−7 but never gets lower than 1E−8, you might want to specify CONVH=1E−6 in the PROC MIXED statement to get results; however, interpret the results with caution.
- An infinite likelihood during the iteration process means that the Newton-Raphson algorithm has stepped into a region where either the **R** or **V** matrix is nonpositive definite. This is usually no cause for concern as long as iterations continue. If PROC MIXED stops because of an infinite likelihood, recheck your model to make sure that no observations from the same subject are producing identical rows in **R** or **V** and that you have enough data to estimate the particular covariance structure you have selected. Any time that the final estimated likelihood is infinite, subsequent results should be interpreted with caution.
- A nonpositive definite Hessian matrix can indicate a surface saddlepoint or linear dependencies among the parameters.
- A warning message about the singularities of **X** changing indicates that there is some linear dependency in the estimate of $\mathbf{X}'\widehat{\mathbf{V}}^{-1}\mathbf{X}$ that is not found in $\mathbf{X}'\mathbf{X}$. This can adversely affect the likelihood calculations and optimization process. If you encounter this problem, make sure that your model specification is reasonable and that you have enough data to estimate the particular covariance structure you have selected. Rearranging effects in the MODEL statement so that the most significant ones are first can help, because PROC MIXED sweeps the estimate of $\mathbf{X}'\mathbf{V}^{-1}\mathbf{X}$ in the order of the MODEL effects and the sweep is more stable if larger pivots are dealt with first. If this does not help, specifying starting values with the PARMS statement can place the optimization on a different and possibly more stable path.
- Lack of convergence can indicate model misspecification or a violation of the normality assumption.

Memory

Let p be the number of columns in $\mathbf{X}$, and let g be the number of columns in $\mathbf{Z}$. For large models, most of the memory resources are required for holding symmetric matrices of order p, g, and $p+g$. The approximate memory requirement in bytes is

$$40(p^2 + g^2) + 32(p + g)^2$$

If you have a large model that exceeds the memory capacity of your computer, see the suggestions listed under "Computing Time."

Computing Time

PROC MIXED is computationally intensive, and execution times can be long. In addition to the CPU time used in collecting sums and crossproducts and in solving the mixed model equations (as in PROC GLM), considerable CPU time is often required to compute the likelihood function and its derivatives. These latter computations are performed for every Newton-Raphson iteration.

If you have a model that takes too long to run, the following suggestions can be helpful:

- Examine the "Model Information" table to find out the number of columns in the $\mathbf{X}$ and $\mathbf{Z}$ matrices. A large number of columns in either matrix can greatly increase computing time. You might want to eliminate some higher-order effects if they are too large.
- If you have a $\mathbf{Z}$ matrix with a lot of columns, use the DDFM=BW option in the MODEL statement to eliminate the time required for the containment method.
- If possible, "factor out" a common effect from the effects in the RANDOM statement and make it the SUBJECT= effect. This creates a block-diagonal $\mathbf{G}$ matrix and can often speed calculations.
- If possible, use the same or nested SUBJECT= effects in all RANDOM and REPEATED statements.
- If your data set is very large, you might want to analyze it in pieces. The BY statement can help implement this strategy.
- In general, specify random effects with a lot of levels in the REPEATED statement and those with a few levels in the RANDOM statement.
- The METHOD=MIVQUE0 option runs faster than either the METHOD=REML or METHOD=ML option because it is noniterative.
- You can specify known values for the covariance parameters by using the HOLD= or NOITER option in the PARMS statement or the GDATA= option in the RANDOM statement. This eliminates the need for iteration.
- The LOGNOTE option in the PROC MIXED statement writes periodic messages to the SAS log concerning the status of the calculations. It can help you diagnose where the slowdown is occurring.

Examples: Mixed Procedure

The following are basic examples of the use of PROC MIXED. More examples and details can be found in Littell et al. (2006), Wolfinger (1997), Verbeke and Molenberghs (1997, 2000), Murray (1998), Singer (1998), Sullivan, Dukes, and Losina (1999), and Brown and Prescott (1999).

Example 56.1: Split-Plot Design

PROC MIXED can fit a variety of mixed models. One of the most common mixed models is the split-plot design. The split-plot design involves two experimental factors, A and B. Levels of A are randomly assigned to whole plots (main plots), and levels of B are randomly assigned to split plots (subplots) within each whole plot. The design provides more precise information about B than about A, and it often arises when A can be applied only to large experimental units. An example is where A represents irrigation levels for large plots of land and B represents different crop varieties planted in each large plot.

Consider the following data from Stroup (1989a), which arise from a balanced split-plot design with the whole plots arranged in a randomized complete-block design. The variable A is the whole-plot factor, and the variable B is the subplot factor. A traditional analysis of these data involves the construction of the whole-plot error (A*Block) to test A and the pooled residual error (B*Block and A*B*Block) to test B and A*B. To carry out this analysis with PROC GLM, you must use a TEST statement to obtain the correct *F* test for A.

Performing a mixed model analysis with PROC MIXED eliminates the need for the error term construction. PROC MIXED estimates variance components for Block, A*Block, and the residual, and it automatically incorporates the correct error terms into test statistics.

The following statements create a DATA set for a split-plot design with four blocks, three whole-plot levels, and two subplot levels:

```
data sp;
  input Block A B Y @@;
  datalines;
1 1 1  56  1 1 2  41
1 2 1  50  1 2 2  36
1 3 1  39  1 3 2  35
2 1 1  30  2 1 2  25
2 2 1  36  2 2 2  28
2 3 1  33  2 3 2  30
3 1 1  32  3 1 2  24
3 2 1  31  3 2 2  27
3 3 1  15  3 3 2  19
4 1 1  30  4 1 2  25
4 2 1  35  4 2 2  30
4 3 1  17  4 3 2  18
;
```

The following statements fit the split-plot model assuming random block effects:

```
proc mixed;
   class A B Block;
   model Y = A B A*B;
   random Block A*Block;
run;
```

The variables A, B, and Block are listed as classification variables in the CLASS statement. The columns of model matrix **X** consist of indicator variables corresponding to the levels of the fixed effects A, B, and A*B listed on the right side of the MODEL statement. The dependent variable Y is listed on the left side of the MODEL statement.

The columns of the model matrix **Z** consist of indicator variables corresponding to the levels of the random effects Block and A*Block. The **G** matrix is diagonal and contains the variance components of Block and A*Block. The **R** matrix is also diagonal and contains the residual variance.

The SAS statements produce Output 56.1.1–Output 56.1.8.

The "Model Information" table in Output 56.1.1 lists basic information about the split-plot model. REML is used to estimate the variance components, and the residual variance is profiled from the optimization.

Output 56.1.1 Results for Split-Plot Analysis

```
                        The Mixed Procedure

                         Model Information

Data Set                          WORK.SP
Dependent Variable                Y
Covariance Structure              Variance Components
Estimation Method                 REML
Residual Variance Method          Profile
Fixed Effects SE Method           Model-Based
Degrees of Freedom Method         Containment
```

The "Class Level Information" table in Output 56.1.2 lists the levels of all variables specified in the CLASS statement. You can check this table to make sure that the data are correct.

Output 56.1.2 Split-Plot Example (*continued*)

```
                      Class Level Information

   Class     Levels    Values

   A              3    1 2 3
   B              2    1 2
   Block          4    1 2 3 4
```

The "Dimensions" table in Output 56.1.3 lists the magnitudes of various vectors and matrices. The **X** matrix is seen to be 24×12, and the **Z** matrix is 24×16.

Output 56.1.3 Split-Plot Example (*continued*)

Dimensions	
Covariance Parameters	3
Columns in X	12
Columns in Z	16
Subjects	1
Max Obs Per Subject	24

The "Number of Observations" table in Output 56.1.4 shows that all observations read from the data set are used in the analysis.

Output 56.1.4 Split-Plot Example (*continued*)

Number of Observations	
Number of Observations Read	24
Number of Observations Used	24
Number of Observations Not Used	0

PROC MIXED estimates the variance components for Block, A*Block, and the residual by REML. The REML estimates are the values that maximize the likelihood of a set of linearly independent error contrasts, and they provide a correction for the downward bias found in the usual maximum likelihood estimates. The objective function is -2 times the logarithm of the restricted likelihood, and PROC MIXED minimizes this objective function to obtain the estimates.

The minimization method is the Newton-Raphson algorithm, which uses the first and second derivatives of the objective function to iteratively find its minimum. The "Iteration History" table in Output 56.1.5 records the steps of that optimization process. For this example, only one iteration is required to obtain the estimates. The Evaluations column reveals that the restricted likelihood is evaluated once for each of the iterations. A criterion of 0 indicates that the Newton-Raphson algorithm has converged.

Output 56.1.5 Split-Plot Analysis (*continued*)

Iteration History

Iteration	Evaluations	-2 Res Log Like	Criterion
0	1	139.81461222	
1	1	119.76184570	0.00000000

Convergence criteria met.

The REML estimates for the variance components of Block, A*Block, and the residual are 62.40, 15.38, and 9.36, respectively, as listed in the Estimate column of the "Covariance Parameter Estimates" table in Output 56.1.6.

Output 56.1.6 Split-Plot Analysis (*continued*)

Covariance Parameter Estimates

Cov Parm	Estimate
Block	62.3958
A*Block	15.3819
Residual	9.3611

The "Fit Statistics" table in Output 56.1.7 lists several pieces of information about the fitted mixed model, including the residual log likelihood. The Akaike (AIC) and Bayesian (BIC) information criteria can be used to compare different models; the ones with smaller values are preferred. The AICC information criteria is a small-sample bias-adjusted form of the Akaike criterion (Hurvich and Tsai 1989).

Output 56.1.7 Split-Plot Analysis (*continued*)

Fit Statistics

-2 Res Log Likelihood	119.8
AIC (smaller is better)	125.8
AICC (smaller is better)	127.5
BIC (smaller is better)	123.9

Finally, the fixed effects are tested by using Type 3 estimable functions (Output 56.1.8).

Output 56.1.8 Split-Plot Analysis (*continued*)

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
A	2	6	4.07	0.0764
B	1	9	19.39	0.0017
A*B	2	9	4.02	0.0566

The tests match the one obtained from the following PROC GLM statements:

```
proc glm data=sp;
   class A B Block;
   model Y = A B A*B Block A*Block;
   test h=A e=A*Block;
run;
```

You can continue this analysis by producing solutions for the fixed and random effects and then testing various linear combinations of them by using the CONTRAST and ESTIMATE statements. If you use the same CONTRAST and ESTIMATE statements with PROC GLM, the test statistics

correspond to the fixed-effects-only model. The test statistics from PROC MIXED incorporate the random effects.

The various "inference space" contrasts given by Stroup (1989a) can be implemented via the ESTIMATE statement. Consider the following examples:

```
proc mixed data=sp;
   class A B Block;
   model Y = A B A*B;
   random Block A*Block;
   estimate 'a1 mean narrow'
            intercept 1 A 1 B .5 .5 A*B .5 .5 |
            Block     .25 .25 .25 .25
            A*Block   .25 .25 .25 .25 0 0 0 0 0 0 0 0;

   estimate 'a1 mean intermed'
            intercept 1 A 1 B .5 .5 A*B .5 .5 |
            Block     .25 .25 .25 .25;
   estimate 'a1 mean broad'
            intercept 1 a 1 b .5 .5 A*B .5 .5;
run;
```

These statements result in Output 56.1.9.

Output 56.1.9 Inference Space Results

The Mixed Procedure

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|
a1 mean narrow	32.8750	1.0817	9	30.39	<.0001
a1 mean intermed	32.8750	2.2396	9	14.68	<.0001
a1 mean broad	32.8750	4.5403	9	7.24	<.0001

Note that all the estimates are equal, but their standard errors increase with the size of the inference space. The narrow inference space consists of the observed levels of Block and A*Block, and the *t*-statistic value of 30.39 applies only to these levels. This is the same *t* statistic computed by PROC GLM, because it computes standard errors from the narrow inference space. The intermediate inference space consists of the observed levels of Block and the entire population of levels from which A*Block are sampled. The *t*-statistic value of 14.68 applies to this intermediate space. The broad inference space consists of arbitrary random levels of both Block and A*Block, and the *t*-statistic value of 7.24 is appropriate. Note that the larger the inference space, the weaker the conclusion. However, the broad inference space is usually the one of interest, and even in this space conclusive results are common. The highly significant *p*-value for 'a1 mean broad' is an example. You can also obtain the 'a1 mean broad' result by specifying A in an LSMEANS statement. For more discussion of the inference space concept, see McLean, Sanders, and Stroup (1991).

The following statements illustrate another feature of the RANDOM statement. Recall that the basic statements for a split-plot design with whole plots arranged in randomized blocks are as follows.

```
proc mixed;
   class A B Block;
   model Y = A B A*B;
   random Block A*Block;
run;
```

An equivalent way of specifying this model is as follows:

```
proc mixed data=sp;
   class A B Block;
   model Y = A B A*B;
   random intercept A / subject=Block;
run;
```

In general, if all of the effects in the RANDOM statement can be nested within one effect, you can specify that one effect by using the SUBJECT= option. The subject effect is, in a sense, "factored out" of the random effects. The specification that uses the SUBJECT= effect can result in faster execution times for large problems because PROC MIXED is able to perform the likelihood calculations separately for each subject.

Example 56.2: Repeated Measures

The following data are from Pothoff and Roy (1964) and consist of growth measurements for 11 girls and 16 boys at ages 8, 10, 12, and 14. Some of the observations are suspect (for example, the third observation for person 20); however, all of the data are used here for comparison purposes.

The analysis strategy employs a linear growth curve model for the boys and girls as well as a variance-covariance model that incorporates correlations for all of the observations arising from the same person. The data are assumed to be Gaussian, and their likelihood is maximized to estimate the model parameters. See Jennrich and Schluchter (1986), Louis (1988), Crowder and Hand (1990), Diggle, Liang, and Zeger (1994), and Everitt (1995) for overviews of this approach to repeated measures. Jennrich and Schluchter present results for the Pothoff and Roy data from various covariance structures. The PROC MIXED statements to fit an unstructured variance matrix (their Model 2) are as follows:

```
data pr;
  input Person Gender $ y1 y2 y3 y4;
  y=y1; Age=8;  output;
  y=y2; Age=10; output;
  y=y3; Age=12; output;
  y=y4; Age=14; output;
  drop y1-y4;
  datalines;
 1   F   21.0    20.0    21.5    23.0
 2   F   21.0    21.5    24.0    25.5
 3   F   20.5    24.0    24.5    26.0
 4   F   23.5    24.5    25.0    26.5
 5   F   21.5    23.0    22.5    23.5
```

```
 6   F   20.0   21.0   21.0   22.5
 7   F   21.5   22.5   23.0   25.0
 8   F   23.0   23.0   23.5   24.0
 9   F   20.0   21.0   22.0   21.5
10   F   16.5   19.0   19.0   19.5
11   F   24.5   25.0   28.0   28.0
12   M   26.0   25.0   29.0   31.0
13   M   21.5   22.5   23.0   26.5
14   M   23.0   22.5   24.0   27.5
15   M   25.5   27.5   26.5   27.0
16   M   20.0   23.5   22.5   26.0
17   M   24.5   25.5   27.0   28.5
18   M   22.0   22.0   24.5   26.5
19   M   24.0   21.5   24.5   25.5
20   M   23.0   20.5   31.0   26.0
21   M   27.5   28.0   31.0   31.5
22   M   23.0   23.0   23.5   25.0
23   M   21.5   23.5   24.0   28.0
24   M   17.0   24.5   26.0   29.5
25   M   22.5   25.5   25.5   26.0
26   M   23.0   24.5   26.0   30.0
27   M   22.0   21.5   23.5   25.0
;

proc mixed data=pr method=ml covtest;
   class Person Gender;
   model y = Gender Age Gender*Age / s;
   repeated / type=un subject=Person r;
run;
```

To follow Jennrich and Schluchter, this example uses maximum likelihood (METHOD=ML) instead of the default REML to estimate the unknown covariance parameters. The COVTEST option requests asymptotic tests of all the covariance parameters.

The MODEL statement first lists the dependent variable Y. The fixed effects are then listed after the equal sign. The variable Gender requests a different intercept for the girls and boys, Age models an overall linear growth trend, and Gender*Age makes the slopes different over time. It is actually not necessary to specify Age separately, but doing so enables PROC MIXED to carry out a test for heterogeneous slopes. The S option requests the display of the fixed-effects solution vector.

The REPEATED statement contains no effects, taking advantage of the default assumption that the observations are ordered similarly for each subject. The TYPE=UN option requests an unstructured block for each SUBJECT=Person. The **R** matrix is, therefore, block diagonal with 27 blocks, each block consisting of identical 4×4 unstructured matrices. The 10 parameters of these unstructured blocks make up the covariance parameters estimated by maximum likelihood. The R option requests that the first block of **R** be displayed.

The results from this analysis are shown in Output 56.2.1–Output 56.2.9.

Output 56.2.1 Repeated Measures Analysis with Unstructured Covariance Matrix

The Mixed Procedure

Model Information

Data Set	WORK.PR
Dependent Variable	y
Covariance Structure	Unstructured
Subject Effect	Person
Estimation Method	ML
Residual Variance Method	None
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Between-Within

In Output 56.2.1, the covariance structure is listed as "Unstructured," and no residual variance is used with this structure. The default degrees-of-freedom method here is "Between-Within."

Output 56.2.2 Repeated Measures Analysis (*continued*)

Class Level Information

Class	Levels	Values
Person	27	1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
Gender	2	F M

In Output 56.2.2, note that Person has 27 levels and Gender has 2.

Output 56.2.3 Repeated Measures Analysis (*continued*)

Dimensions

Covariance Parameters	10
Columns in X	6
Columns in Z	0
Subjects	27
Max Obs Per Subject	4

In Output 56.2.3, the 10 covariance parameters result from the 4×4 unstructured blocks of $\mathbf{R}$. There is no $\mathbf{Z}$ matrix for this model, and each of the 27 subjects has a maximum of 4 observations.

Output 56.2.4 Repeated Measures Analysis (*continued*)

Number of Observations		
Number of Observations Read		108
Number of Observations Used		108
Number of Observations Not Used		0

Iteration History

Iteration	Evaluations	-2 Log Like	Criterion
0	1	478.24175986	
1	2	419.47721707	0.00000152
2	1	419.47704812	0.00000000

Convergence criteria met.

Three Newton-Raphson iterations are required to find the maximum likelihood estimates (Output 56.2.4). The default relative Hessian criterion has a final value less than 1E−8, indicating the convergence of the Newton-Raphson algorithm and the attainment of an optimum.

Output 56.2.5 Repeated Measures Analysis (*continued*)

Estimated R Matrix for Person 1

Row	Col1	Col2	Col3	Col4
1	5.1192	2.4409	3.6105	2.5222
2	2.4409	3.9279	2.7175	3.0624
3	3.6105	2.7175	5.9798	3.8235
4	2.5222	3.0624	3.8235	4.6180

The 4×4 matrix in Output 56.2.5 is the estimated unstructured covariance matrix. It is the estimate of the first block of $\mathbf{R}$, and the other 26 blocks all have the same estimate.

Output 56.2.6 Repeated Measures Analysis (*continued*)

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr Z
UN(1,1)	Person	5.1192	1.4169	3.61	0.0002
UN(2,1)	Person	2.4409	0.9835	2.48	0.0131
UN(2,2)	Person	3.9279	1.0824	3.63	0.0001
UN(3,1)	Person	3.6105	1.2767	2.83	0.0047
UN(3,2)	Person	2.7175	1.0740	2.53	0.0114
UN(3,3)	Person	5.9798	1.6279	3.67	0.0001
UN(4,1)	Person	2.5222	1.0649	2.37	0.0179
UN(4,2)	Person	3.0624	1.0135	3.02	0.0025
UN(4,3)	Person	3.8235	1.2508	3.06	0.0022
UN(4,4)	Person	4.6180	1.2573	3.67	0.0001

The "Covariance Parameter Estimates" table in Output 56.2.6 lists the 10 estimated covariance parameters in order; note their correspondence to the first block of **R** displayed in Output 56.2.5. The parameter estimates are labeled according to their location in the block in the Cov Parm column, and all of these estimates are associated with Person as the subject effect. The Std Error column lists approximate standard errors of the covariance parameters obtained from the inverse Hessian matrix. These standard errors lead to approximate Wald Z statistics, which are compared with the standard normal distribution The results of these tests indicate that all the parameters are significantly different from 0; however, the Wald test can be unreliable in small samples.

To carry out Wald tests of various linear combinations of these parameters, use the following procedure. First, run the statements again, adding the ASYCOV option and an ODS statement:

```
ods output CovParms=cp AsyCov=asy;
proc mixed data=pr method=ml covtest asycov;
   class Person Gender;
   model y = Gender Age Gender*Age / s;
   repeated / type=un subject=Person r;
run;
```

This creates two data sets, cp and asy, which contain the covariance parameter estimates and their asymptotic variance covariance matrix, respectively. Then read these data sets into the SAS/IML matrix programming language as follows:

```
proc iml;
   use cp;
   read all var {Estimate} into est;
   use asy;
   read all var ('CovP1':'CovP10') into asy;
```

You can then construct your desired linear combinations and corresponding quadratic forms with the asy matrix.

Output 56.2.7 Repeated Measures Analysis (*continued*)

Fit Statistics	
-2 Log Likelihood	419.5
AIC (smaller is better)	447.5
AICC (smaller is better)	452.0
BIC (smaller is better)	465.6

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
9	58.76	<.0001

The null model likelihood ratio test (LRT) in Output 56.2.7 is highly significant for this model, indicating that the unstructured covariance matrix is preferred to the diagonal matrix of the ordinary least squares null model. The degrees of freedom for this test is 9, which is the difference between 10 and the 1 parameter for the null model's diagonal matrix.

Output 56.2.8 Repeated Measures Analysis (*continued*)

Solution for Fixed Effects

Effect	Gender	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept		15.8423	0.9356	25	16.93	<.0001
Gender	F	1.5831	1.4658	25	1.08	0.2904
Gender	M	0	.	.	.	.
Age		0.8268	0.07911	25	10.45	<.0001
Age*Gender	F	-0.3504	0.1239	25	-2.83	0.0091
Age*Gender	M	0	.	.	.	.

The "Solution for Fixed Effects" table in Output 56.2.8 lists the solution vector for the fixed effects. The estimate of the boys' intercept is 15.8423, while that for the girls is $15.8423 + 1.5831 = 17.0654$. Similarly, the estimate for the boys' slope is 0.8268, while that for the girls is $0.8268 - 0.3504 = 0.4764$. Thus the girls' starting point is larger than that for the boys, but their growth rate is about half that of the boys.

Note that two of the estimates equal 0; this is a result of the overparameterized model used by PROC MIXED. You can obtain a full-rank parameterization by using the following MODEL statement:

```
model y = Gender Gender*Age / noint s;
```

Here, the NOINT option causes the different intercepts to be fit directly as the two levels of Gender. However, this alternative specification results in different tests for these effects.

Output 56.2.9 Repeated Measures Analysis (*continued*)

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Gender	1	25	1.17	0.2904
Age	1	25	110.54	<.0001
Age*Gender	1	25	7.99	0.0091

The "Type 3 Tests of Fixed Effects" table in Output 56.2.9 displays Type 3 tests for all of the fixed effects. These tests are partial in the sense that they account for all of the other fixed effects in the model. In addition, you can use the HTYPE= option in the MODEL statement to obtain Type 1 (sequential) or Type 2 (also partial) tests of effects.

It is usually best to consider higher-order terms first, and in this case the Age*Gender test reveals a difference between the slopes that is statistically significant at the 1% level. Note that the p-value for this test (0.0091) is the same as the p-value in the "Age*Gender F" row in the "Solution for Fixed Effects" table (Output 56.2.8) and that the F statistic (7.99) is the square of the t statistic (-2.83), ignoring rounding error. Similar connections are evident among the other rows in these two tables.

The Age test is one for an overall growth curve accounting for possible heterogeneous slopes, and it is highly significant. Finally, the Gender row tests the null hypothesis of a common intercept, and this hypothesis cannot be rejected from these data.

As an alternative to the F tests shown here, you can carry out likelihood ratio tests of various hypotheses by fitting the reduced models, subtracting -2 log likelihoods, and comparing the resulting statistics with χ^2 distributions.

Since the different levels of the repeated effect represent different years, it is natural to try fitting a time series model to the data within each subject. To obtain time series structures in **R**, you can replace TYPE=UN with TYPE=AR(1) or TYPE=TOEP to obtain the first- or nth-order autoregressive covariance matrices, respectively. For example, the statements to fit an AR(1) structure are as follows:

```
proc mixed data=pr method=ml;
   class Person Gender;
   model y = Gender Age Gender*Age / s;
   repeated / type=ar(1) sub=Person r;
run;
```

To fit a random coefficients model, use the following statements:

```
proc mixed data=pr method=ml;
   class Person Gender;
   model y = Gender Age Gender*Age / s;
   random intercept Age / type=un sub=Person g;
run;
```

This specifies an unstructured covariance matrix for the random intercept and slope. In mixed model notation, **G** is block diagonal with identical 2×2 unstructured blocks for each person. By default, **R** becomes $\sigma^2\mathbf{I}$. See Example 56.5 for further information about this model.

Finally, you can fit a compound symmetry structure by using TYPE=CS, as follows:

```
proc mixed data=pr method=ml covtest;
   class Person Gender;
   model y = Gender Age Gender*Age / s;
   repeated / type=cs subject=Person r;
run;
```

The results from this analysis are shown in Output 56.2.10–Output 56.2.17.

The "Model Information" table in Output 56.2.10 is the same as before except for the change in "Covariance Structure."

Output 56.2.10 Repeated Measures Analysis with Compound Symmetry Structure

```
                        The Mixed Procedure

                         Model Information

Data Set                          WORK.PR
Dependent Variable                y
Covariance Structure              Compound Symmetry
Subject Effect                    Person
Estimation Method                 ML
Residual Variance Method          Profile
Fixed Effects SE Method           Model-Based
Degrees of Freedom Method         Between-Within
```

The "Dimensions" table in Output 56.2.11 shows that there are only two covariance parameters in the compound symmetry model; this covariance structure has common variance and common covariance.

Output 56.2.11 Analysis with Compound Symmetry (*continued*)

```
                     Class Level Information

Class     Levels    Values

Person        27    1 2 3 4 5 6 7 8 9 10 11 12 13
                    14 15 16 17 18 19 20 21 22 23
                    24 25 26 27
Gender         2    F M
```

Output 56.2.11 *continued*

Dimensions	
Covariance Parameters	2
Columns in X	6
Columns in Z	0
Subjects	27
Max Obs Per Subject	4

Number of Observations	
Number of Observations Read	108
Number of Observations Used	108
Number of Observations Not Used	0

Since the data are balanced, only one step is required to find the estimates (Output 56.2.12).

Output 56.2.12 Analysis with Compound Symmetry (*continued*)

Iteration History

Iteration	Evaluations	-2 Log Like	Criterion
0	1	478.24175986	
1	1	428.63905802	0.00000000

Convergence criteria met.

Output 56.2.13 displays the estimated **R** matrix for the first subject. Note the compound symmetry structure here, which consists of a common covariance with a diagonal enhancement.

Output 56.2.13 Analysis with Compound Symmetry (*continued*)

Estimated R Matrix for Person 1

Row	Col1	Col2	Col3	Col4
1	4.9052	3.0306	3.0306	3.0306
2	3.0306	4.9052	3.0306	3.0306
3	3.0306	3.0306	4.9052	3.0306
4	3.0306	3.0306	3.0306	4.9052

The common covariance is estimated to be 3.0306, as listed in the CS row of the "Covariance Parameter Estimates" table in Output 56.2.14, and the residual variance is estimated to be 1.8746, as listed in the Residual row. You can use these two numbers to estimate the intraclass correlation coefficient (ICC) for this model. Here, the ICC estimate equals $3.0306/(3.0306 + 1.8746) = 0.6178$. You can also obtain this number by adding the RCORR option to the REPEATED statement.

Output 56.2.14 Analysis with Compound Symmetry (*continued*)

Covariance Parameter Estimates

Cov Parm	Subject	Estimate	Standard Error	Z Value	Pr Z
CS	Person	3.0306	0.9552	3.17	0.0015
Residual		1.8746	0.2946	6.36	<.0001

In the case shown in Output 56.2.15, the null model LRT has only one degree of freedom, corresponding to the common covariance parameter. The test indicates that modeling this extra covariance is superior to fitting the simple null model.

Output 56.2.15 Analysis with Compound Symmetry (*continued*)

Fit Statistics

−2 Log Likelihood	428.6
AIC (smaller is better)	440.6
AICC (smaller is better)	441.5
BIC (smaller is better)	448.4

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
1	49.60	<.0001

Note that the fixed-effects estimates and their standard errors (Output 56.2.16) are not very different from those in the preceding unstructured example (Output 56.2.8).

Output 56.2.16 Analysis with Compound Symmetry (*continued*)

Solution for Fixed Effects

Effect	Gender	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept		16.3406	0.9631	25	16.97	<.0001
Gender	F	1.0321	1.5089	25	0.68	0.5003
Gender	M	0	.	.	.	.
Age		0.7844	0.07654	79	10.25	<.0001
Age*Gender	F	−0.3048	0.1199	79	−2.54	0.0130
Age*Gender	M	0	.	.	.	.

The F tests shown in Output 56.2.17 are also similar to those from the preceding unstructured example (Output 56.2.9). Again, the slopes are significantly different but the intercepts are not.

Output 56.2.17 Analysis with Compound Symmetry (*continued*)

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Gender	1	25	0.47	0.5003
Age	1	79	111.10	<.0001
Age*Gender	1	79	6.46	0.0130

You can fit the same compound symmetry model with the following specification by using the RANDOM statement:

```
proc mixed data=pr method=ml;
   class Person Gender;
   model y = Gender Age Gender*Age / s;
   random Person;
run;
```

Compound symmetry is the structure that Jennrich and Schluchter deemed best among the ones they fit. To carry the analysis one step further, you can use the GROUP= option as follows to specify heterogeneity of this structure across girls and boys:

```
proc mixed data=pr method=ml;
   class Person Gender;
   model y = Gender Age Gender*Age / s;
   repeated / type=cs subject=Person group=Gender;
run;
```

The results from this analysis are shown in Output 56.2.18–Output 56.2.24. Note that in Output 56.2.18 Gender is listed as a "Group Effect."

Output 56.2.18 Repeated Measures Analysis with Heterogeneous Structures

The Mixed Procedure

Model Information

Data Set	WORK.PR
Dependent Variable	y
Covariance Structure	Compound Symmetry
Subject Effect	Person
Group Effect	Gender
Estimation Method	ML
Residual Variance Method	None
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Between-Within

The four covariance parameters listed in Output 56.2.19 result from the two compound symmetry structures corresponding to the two levels of Gender.

Output 56.2.19 Analysis with Heterogeneous Structures (*continued*)

```
                Class Level Information

Class      Levels    Values

Person         27    1 2 3 4 5 6 7 8 9 10 11 12 13
                     14 15 16 17 18 19 20 21 22 23
                     24 25 26 27
Gender          2    F M


                    Dimensions

          Covariance Parameters              4
          Columns in X                       6
          Columns in Z                       0
          Subjects                          27
          Max Obs Per Subject                4


                Number of Observations

       Number of Observations Read           108
       Number of Observations Used           108
       Number of Observations Not Used         0
```

As Output 56.2.20 shows, even with the heterogeneity, only one iteration is required for convergence.

Output 56.2.20 Analysis with Heterogeneous Structures (*continued*)

```
                        Iteration History

Iteration    Evaluations        -2 Log Like       Criterion

        0              1       478.24175986
        1              1       408.81297228      0.00000000

                  Convergence criteria met.
```

The "Covariance Parameter Estimates" table in Output 56.2.21 lists the heterogeneous estimates. Note that both the common covariance and the diagonal enhancement differ between girls and boys.

Output 56.2.21 Analysis with Heterogeneous Structures (*continued*)

```
                Covariance Parameter Estimates

      Cov Parm     Subject    Group       Estimate

      Variance     Person     Gender F      0.5900
      CS           Person     Gender F      3.8804
      Variance     Person     Gender M      2.7577
      CS           Person     Gender M      2.4463
```

As Output 56.2.22 shows, both Akaike's information criterion (424.8) and Schwarz's Bayesian information criterion (435.2) are smaller for this model than for the homogeneous compound symmetry model (440.6 and 448.4, respectively). This indicates that the heterogeneous model is more appropriate. To construct the likelihood ratio test between the two models, subtract the -2 log likelihood values: $428.6 - 408.8 = 19.8$. Comparing this value with the χ^2 distribution with two degrees of freedom yields a p-value less than 0.0001, again favoring the heterogeneous model.

Output 56.2.22 Analysis with Heterogeneous Structures (*continued*)

Fit Statistics

-2 Log Likelihood	408.8
AIC (smaller is better)	424.8
AICC (smaller is better)	426.3
BIC (smaller is better)	435.2

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
3	69.43	<.0001

Note that the fixed-effects estimates shown in Output 56.2.23 are the same as in the homogeneous case, but the standard errors are different.

Output 56.2.23 Analysis with Heterogeneous Structures (*continued*)

Solution for Fixed Effects

Effect	Gender	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept		16.3406	1.1130	25	14.68	<.0001
Gender	F	1.0321	1.3890	25	0.74	0.4644
Gender	M	0	.	.	.	.
Age		0.7844	0.09283	79	8.45	<.0001
Age*Gender	F	-0.3048	0.1063	79	-2.87	0.0053
Age*Gender	M	0	.	.	.	.

The fixed-effects tests shown in Output 56.2.24 are similar to those from previous models, although the p-values do change as a result of specifying a different covariance structure. It is important for you to select a reasonable covariance structure in order to obtain valid inferences for your fixed effects.

Output 56.2.24 Analysis with Heterogeneous Structures (*continued*)

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Gender	1	25	0.55	0.4644
Age	1	79	141.37	<.0001
Age*Gender	1	79	8.22	0.0053

Example 56.3: Plotting the Likelihood

The data for this example are from Hemmerle and Hartley (1973) and are also used for an example in the VARCOMP procedure. The response variable consists of measurements from an oven experiment, and the model contains a fixed effect A and random effects B and A*B.

The SAS statements are as follows:

```
data hh;
  input a b y @@;
  datalines;
1 1 237   1 1 254    1 1 246
1 2 178   1 2 179
2 1 208   2 1 178    2 1 187
2 2 146   2 2 145    2 2 141
3 1 186   3 1 183
3 2 142   3 2 125    3 2 136
;

ods output ParmSearch=parms;
proc mixed data=hh asycov mmeq mmeqsol covtest;
   class a b;
   model y = a / outp=predicted;
   random b a*b;
   lsmeans a;
   parms (17 to 20 by .1) (.3 to .4 by .005) (1.0);
run;
proc print data=predicted;
run;
```

The ASYCOV option in the PROC MIXED statement requests the asymptotic variance matrix of the covariance parameter estimates. This matrix is the observed inverse Fisher information matrix, which equals $2\mathbf{H}^{-1}$, where $\mathbf{H}$ is the Hessian matrix of the objective function evaluated at the final covariance parameter estimates. The MMEQ and MMEQSOL options in the PROC MIXED statement request that the mixed model equations and their solution be displayed.

The OUTP= option in the MODEL statement produces the data set predicted, containing the predicted values. Least squares means (LSMEANS) are requested for A. The PARMS and ODS statements are used to construct a data set containing the likelihood surface.

The results from this analysis are shown in Output 56.3.1–Output 56.3.13.

The "Model Information" table in Output 56.3.1 lists details about this variance components model.

Output 56.3.1 Model Information

The Mixed Procedure

Model Information

Data Set	WORK.HH
Dependent Variable	y
Covariance Structure	Variance Components
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

The "Class Level Information" table in Output 56.3.2 lists the levels for A and B.

Output 56.3.2 Class Level Information

Class Level Information

Class	Levels	Values
a	3	1 2 3
b	2	1 2

The "Dimensions" table in Output 56.3.3 reveals that $\mathbf{X}$ is 16×4 and $\mathbf{Z}$ is 16×8. Since there are no SUBJECT= effects, PROC MIXED considers the data to be effectively from one subject with 16 observations.

Output 56.3.3 Model Dimensions and Number of Observations

Dimensions

Covariance Parameters	3
Columns in X	4
Columns in Z	8
Subjects	1
Max Obs Per Subject	16

Number of Observations

Number of Observations Read	16
Number of Observations Used	16
Number of Observations Not Used	0

Only a portion of the "Parameter Search" table is shown in Output 56.3.4 because the full listing has 651 rows.

Output 56.3.4 Selected Results of Parameter Search

The Mixed Procedure

CovP1	CovP2	CovP3	Variance	Res Log Like	-2 Res Log Like
17.0000	0.3000	1.0000	80.1400	-52.4699	104.9399
17.0000	0.3050	1.0000	80.0466	-52.4697	104.9393
17.0000	0.3100	1.0000	79.9545	-52.4694	104.9388
17.0000	0.3150	1.0000	79.8637	-52.4692	104.9384
17.0000	0.3200	1.0000	79.7742	-52.4691	104.9381
17.0000	0.3250	1.0000	79.6859	-52.4690	104.9379
17.0000	0.3300	1.0000	79.5988	-52.4689	104.9378
17.0000	0.3350	1.0000	79.5129	-52.4689	104.9377
17.0000	0.3400	1.0000	79.4282	-52.4689	104.9377
17.0000	0.3450	1.0000	79.3447	-52.4689	104.9378
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
20.0000	0.3550	1.0000	78.2003	-52.4683	104.9366
20.0000	0.3600	1.0000	78.1201	-52.4684	104.9368
20.0000	0.3650	1.0000	78.0409	-52.4685	104.9370
20.0000	0.3700	1.0000	77.9628	-52.4687	104.9373
20.0000	0.3750	1.0000	77.8857	-52.4689	104.9377
20.0000	0.3800	1.0000	77.8096	-52.4691	104.9382
20.0000	0.3850	1.0000	77.7345	-52.4693	104.9387
20.0000	0.3900	1.0000	77.6603	-52.4696	104.9392
20.0000	0.3950	1.0000	77.5871	-52.4699	104.9399
20.0000	0.4000	1.0000	77.5148	-52.4703	104.9406

As Output 56.3.5 shows, convergence occurs quickly because PROC MIXED starts from the best value from the grid search.

Output 56.3.5 Iteration History and Convergence Status

Iteration History

Iteration	Evaluations	-2 Res Log Like	Criterion
1	2	104.93416367	0.00000000

Convergence criteria met.

The "Covariance Parameter Estimates" table in Output 56.3.6 lists the variance components estimates. Note that B is much more variable than A*B.

Output 56.3.6 Estimated Covariance Parameters

Covariance Parameter Estimates

Cov Parm	Estimate	Standard Error	Z Value	Pr > Z
b	1464.36	2098.01	0.70	0.2426
a*b	26.9581	59.6570	0.45	0.3257
Residual	78.8426	35.3512	2.23	0.0129

The asymptotic covariance matrix in Output 56.3.7 also reflects the large variability of B relative to A*B.

Output 56.3.7 Asymptotic Covariance Matrix of Covariance Parameters

Asymptotic Covariance Matrix of Estimates

Row	Cov Parm	CovP1	CovP2	CovP3
1	b	4401640	1.2831	-273.32
2	a*b	1.2831	3558.96	-502.84
3	Residual	-273.32	-502.84	1249.71

As Output 56.3.8 shows, the PARMS likelihood ratio test (LRT) compares the best model from the grid search with the final fitted model. Since these models are nearly the same, the LRT is not significant.

Output 56.3.8 Fit Statistics and Likelihood Ratio Test

Fit Statistics

-2 Res Log Likelihood	104.9
AIC (smaller is better)	110.9
AICC (smaller is better)	113.6
BIC (smaller is better)	107.0

PARMS Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
2	0.00	1.0000

The mixed model equations are analogous to the normal equations in the standard linear model. As Output 56.3.9 shows, for this example, rows 1–4 correspond to the fixed effects, rows 5–12 correspond to the random effects, and Col13 corresponds to the dependent variable.

Output 56.3.9 Mixed Model Equations

Mixed Model Equations

Row	Effect	a	b	Col1	Col2	Col3	Col4	Col5	Col6	Col7
1	Intercept			0.2029	0.06342	0.07610	0.06342	0.1015	0.1015	0.03805
2	a	1		0.06342	0.06342			0.03805	0.02537	0.03805
3	a	2		0.07610		0.07610		0.03805	0.03805	
4	a	3		0.06342			0.06342	0.02537	0.03805	
5	b		1	0.1015	0.03805	0.03805	0.02537	0.1022		0.03805
6	b		2	0.1015	0.02537	0.03805	0.03805		0.1022	
7	a*b	1	1	0.03805	0.03805			0.03805		0.07515
8	a*b	1	2	0.02537	0.02537				0.02537	
9	a*b	2	1	0.03805		0.03805		0.03805		
10	a*b	2	2	0.03805		0.03805			0.03805	
11	a*b	3	1	0.02537			0.02537	0.02537		
12	a*b	3	2	0.03805			0.03805		0.03805	

Mixed Model Equations

Row	Col8	Col9	Col10	Col11	Col12	Col13
1	0.02537	0.03805	0.03805	0.02537	0.03805	36.4143
2	0.02537					13.8757
3		0.03805	0.03805			12.7469
4				0.02537	0.03805	9.7917
5		0.03805		0.02537		21.2956
6	0.02537		0.03805		0.03805	15.1187
7						9.3477
8	0.06246					4.5280
9		0.07515				7.2676
10			0.07515			5.4793
11				0.06246		4.6802
12					0.07515	5.1115

The solution matrix in Output 56.3.10 results from sweeping all but the last row of the mixed model equations matrix. The final column contains a solution vector for the fixed and random effects. The first four rows correspond to fixed effects and the last eight correspond to random effects.

Output 56.3.10 Solutions of the Mixed Model Equations

Mixed Model Equations Solution

Row	Effect	a	b	Col1	Col2	Col3	Col4	Col5	Col6	Col7
1	Intercept			761.84	-29.7718	-29.6578		-731.14	-733.22	-0.4680
2	a	1		-29.7718	59.5436	29.7718		-2.0764	2.0764	-14.0239
3	a	2		-29.6578	29.7718	56.2773		-1.0382	1.0382	0.4680
4	a	3								
5	b		1	-731.14	-2.0764	-1.0382		741.63	722.73	-4.2598
6	b		2	-733.22	2.0764	1.0382		722.73	741.63	4.2598
7	a*b	1	1	-0.4680	-14.0239	0.4680		-4.2598	4.2598	22.8027
8	a*b	1	2	0.4680	-12.9342	-0.4680		4.2598	-4.2598	4.1555
9	a*b	2	1	-0.5257	1.0514	-12.9534		-4.7855	4.7855	2.1570
10	a*b	2	2	0.5257	-1.0514	-14.0048		4.7855	-4.7855	-2.1570
11	a*b	3	1	-12.4663	12.9342	12.4663		-4.2598	4.2598	1.9200
12	a*b	3	2	-14.4918	14.0239	14.4918		4.2598	-4.2598	-1.9200

Mixed Model Equations Solution

Row	Col8	Col9	Col10	Col11	Col12	Col13
1	0.4680	-0.5257	0.5257	-12.4663	-14.4918	159.61
2	-12.9342	1.0514	-1.0514	12.9342	14.0239	53.2049
3	-0.4680	-12.9534	-14.0048	12.4663	14.4918	7.8856
4						
5	4.2598	-4.7855	4.7855	-4.2598	4.2598	26.8837
6	-4.2598	4.7855	-4.7855	4.2598	-4.2598	-26.8837
7	4.1555	2.1570	-2.1570	1.9200	-1.9200	3.0198
8	22.8027	-2.1570	2.1570	-1.9200	1.9200	-3.0198
9	-2.1570	22.5560	4.4021	2.1570	-2.1570	-1.7134
10	2.1570	4.4021	22.5560	-2.1570	2.1570	1.7134
11	-1.9200	2.1570	-2.1570	22.8027	4.1555	-0.8115
12	1.9200	-2.1570	2.1570	4.1555	22.8027	0.8115

The A factor is significant at the 5% level (Output 56.3.11).

Output 56.3.11 Tests of Fixed Effects

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
a	2	2	28.00	0.0345

Output 56.3.12 shows that the significance of A appears to be from the difference between its first level and its other two levels.

Output 56.3.12 Least Squares Means for A Effect

Least Squares Means

Effect	a	Estimate	Standard Error	DF	t Value	Pr > |t|
a	1	212.82	27.6014	2	7.71	0.0164
a	2	167.50	27.5463	2	6.08	0.0260
a	3	159.61	27.6014	2	5.78	0.0286

Output 56.3.13 lists the predicted values from the model. These values are the sum of the fixed-effects estimates and the empirical best linear unbiased predictors (EBLUPs) of the random effects.

Output 56.3.13 Predicted Values

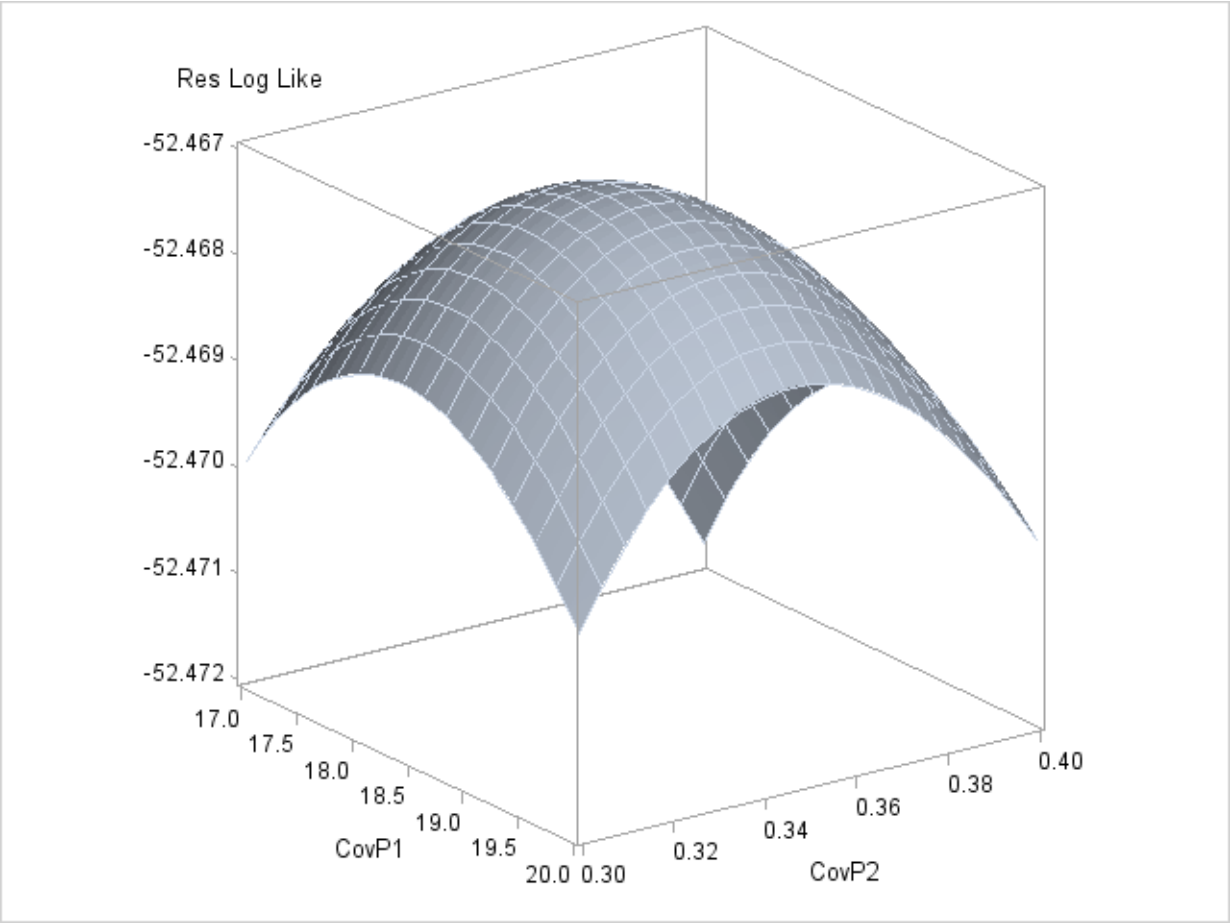
Obs	a	b	y	Pred	StdErr Pred	DF	Alpha	Lower	Upper	Resid
1	1	1	237	242.723	4.72563	10	0.05	232.193	253.252	-5.7228
2	1	1	254	242.723	4.72563	10	0.05	232.193	253.252	11.2772
3	1	1	246	242.723	4.72563	10	0.05	232.193	253.252	3.2772
4	1	2	178	182.916	5.52589	10	0.05	170.603	195.228	-4.9159
5	1	2	179	182.916	5.52589	10	0.05	170.603	195.228	-3.9159
6	2	1	208	192.670	4.70076	10	0.05	182.196	203.144	15.3297
7	2	1	178	192.670	4.70076	10	0.05	182.196	203.144	-14.6703
8	2	1	187	192.670	4.70076	10	0.05	182.196	203.144	-5.6703
9	2	2	146	142.330	4.70076	10	0.05	131.856	152.804	3.6703
10	2	2	145	142.330	4.70076	10	0.05	131.856	152.804	2.6703
11	2	2	141	142.330	4.70076	10	0.05	131.856	152.804	-1.3297
12	3	1	186	185.687	5.52589	10	0.05	173.374	197.999	0.3134
13	3	1	183	185.687	5.52589	10	0.05	173.374	197.999	-2.6866
14	3	2	142	133.542	4.72563	10	0.05	123.013	144.072	8.4578
15	3	2	125	133.542	4.72563	10	0.05	123.013	144.072	-8.5422
16	3	2	136	133.542	4.72563	10	0.05	123.013	144.072	2.4578

To plot the likelihood surface by using ODS Graphics, use the following statements:

```
proc template;
   define statgraph surface;
   begingraph;
      layout overlay3d;
         surfaceplotparm x=CovP1 y=CovP2 z=ResLogLike;
      endlayout;
   endgraph;
   end;
run;
proc sgrender data=parms template=surface;
run;
```

The results from this plot are shown in Output 56.3.14. The peak of the surface is the REML estimates for the B and A*B variance components.

Output 56.3.14 Plot of Likelihood Surface



Example 56.4: Known G and R

This animal breeding example from Henderson (1984, p. 48) considers multiple traits. The data are artificial and consist of measurements of two traits on three animals, but the second trait of the third animal is missing. Assuming an additive genetic model, you can use PROC MIXED to predict the breeding value of both traits on all three animals and also to predict the second trait of the third animal. The data are as follows:

```
data h;
  input Trait Animal Y;
  datalines;
1 1 6
1 2 8
1 3 7
2 1 9
2 2 5
2 3 .
;
```

Both **G** and **R** are known.

$$\mathbf{G} = \begin{bmatrix} 2 & 1 & 1 & 2 & 1 & 1 \\ 1 & 2 & .5 & 1 & 2 & .5 \\ 1 & .5 & 2 & 1 & .5 & 2 \\ 2 & 1 & 1 & 3 & 1.5 & 1.5 \\ 1 & 2 & .5 & 1.5 & 3 & .75 \\ 1 & .5 & 2 & 1.5 & .75 & 3 \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 4 & 0 & 0 & 1 & 0 & 0 \\ 0 & 4 & 0 & 0 & 1 & 0 \\ 0 & 0 & 4 & 0 & 0 & 1 \\ 1 & 0 & 0 & 5 & 0 & 0 \\ 0 & 1 & 0 & 0 & 5 & 0 \\ 0 & 0 & 1 & 0 & 0 & 5 \end{bmatrix}$$

In order to read **G** into PROC MIXED by using the GDATA= option in the RANDOM statement, perform the following DATA step:

```
data g;
   input Row Col1-Col6;
   datalines;
1  2  1  1  2   1     1
2  1  2 .5  1   2    .5
3  1 .5  2  1    .5  2
4  2  1  1  3   1.5  1.5
5  1  2 .5  1.5 3     .75
6  1 .5  2  1.5  .75 3
;
```

The preceding data are in the dense representation for a GDATA= data set. You can also construct a data set with the sparse representation by using Row, Col, and Value variables, although this would require 21 observations instead of 6 for this example.

The PROC MIXED statements are as follows:

```
proc mixed data=h mmeq mmeqsol;
   class Trait Animal;
   model Y = Trait / noint s outp=predicted;
   random Trait*Animal / type=un gdata=g g gi s;
   repeated / type=un sub=Animal r ri;
   parms (4) (1) (5) / noiter;
run;
proc print data=predicted;
run;
```

The MMEQ and MMEQSOL options request the mixed model equations and their solution. The variables Trait and Animal are classification variables, and Trait defines the entire **X** matrix for the fixed-effects portion of the model, since the intercept is omitted with the NOINT option. The fixed-effects solution vector and predicted values are also requested by using the S and OUTP= options, respectively.

The random effect Trait*Animal leads to a $\mathbf{Z}$ matrix with six columns, the first five corresponding to the identity matrix and the last consisting of 0s. An unstructured $\mathbf{G}$ matrix is specified by using the TYPE=UN option, and it is read into PROC MIXED from a SAS data set by using the GDATA=G specification. The G and GI options request the display of $\mathbf{G}$ and $\mathbf{G}^{-1}$, respectively. The S option requests that the random-effects solution vector be displayed.

Note that the preceding $\mathbf{R}$ matrix is block diagonal if the data are sorted by animals. The REPEATED statement exploits this fact by requesting $\mathbf{R}$ to have unstructured 2×2 blocks corresponding to animals, which are the subjects. The R and RI options request that the estimated 2×2 blocks for the first animal and its inverse be displayed. The PARMS statement lists the parameters of this 2×2 matrix. Note that the parameters from $\mathbf{G}$ are not specified in the PARMS statement because they have already been assigned by using the GDATA= option in the RANDOM statement. The NOITER option prevents PROC MIXED from computing residual (restricted) maximum likelihood estimates; instead, the known values are used for inferences.

The results from this analysis are shown in Output 56.4.1–Output 56.4.12.

The "Unstructured" covariance structure (Output 56.4.1) applies to both $\mathbf{G}$ and $\mathbf{R}$ here. The levels of Trait and Animal have been specified correctly.

Output 56.4.1 Model and Class Level Information

```
                    The Mixed Procedure

                     Model Information

Data Set                        WORK.H
Dependent Variable              Y
Covariance Structure            Unstructured
Subject Effect                  Animal
Estimation Method               REML
Residual Variance Method        None
Fixed Effects SE Method         Model-Based
Degrees of Freedom Method       Containment

                  Class Level Information

  Class     Levels    Values

  Trait          2    1 2
  Animal         3    1 2 3
```

The three covariance parameters indicated in Output 56.4.2 correspond to those from the $\mathbf{R}$ matrix. Those from $\mathbf{G}$ are considered fixed and known because of the GDATA= option.

Output 56.4.2 Model Dimensions and Number of Observations

Dimensions	
Covariance Parameters	3
Columns in X	2
Columns in Z	6
Subjects	1
Max Obs Per Subject	6

Number of Observations	
Number of Observations Read	6
Number of Observations Used	5
Number of Observations Not Used	1

Because starting values for the covariance parameters are specified in the PARMS statement, the MIXED procedure prints the residual (restricted) log likelihood at the starting values. Because of the NOITER option in the PARMS statement, this is also the final log likelihood in this analysis (Output 56.4.3).

Output 56.4.3 REML Log Likelihood

Parameter Search

CovP1	CovP2	CovP3	Res Log Like	-2 Res Log Like
4.0000	1.0000	5.0000	-7.3731	14.7463

The block of **R** corresponding to the first animal and the inverse of this block are shown in Output 56.4.4.

Output 56.4.4 Inverse R Matrix

Estimated R Matrix for Animal 1

Row	Col1	Col2
1	4.0000	1.0000
2	1.0000	5.0000

Estimated Inv(R) Matrix for Animal 1

Row	Col1	Col2
1	0.2632	-0.05263
2	-0.05263	0.2105

The **G** matrix as specified in the GDATA= data set and its inverse are shown in Output 56.4.5 and Output 56.4.6.

Output 56.4.5 G Matrix

Estimated G Matrix

Row	Effect	Trait	Animal	Col1	Col2	Col3	Col4
1	Trait*Animal	1	1	2.0000	1.0000	1.0000	2.0000
2	Trait*Animal	1	2	1.0000	2.0000	0.5000	1.0000
3	Trait*Animal	1	3	1.0000	0.5000	2.0000	1.0000
4	Trait*Animal	2	1	2.0000	1.0000	1.0000	3.0000
5	Trait*Animal	2	2	1.0000	2.0000	0.5000	1.5000
6	Trait*Animal	2	3	1.0000	0.5000	2.0000	1.5000

Estimated G Matrix

Row	Col5	Col6
1	1.0000	1.0000
2	2.0000	0.5000
3	0.5000	2.0000
4	1.5000	1.5000
5	3.0000	0.7500
6	0.7500	3.0000

Output 56.4.6 Inverse G Matrix

Estimated Inv(G) Matrix

Row	Effect	Trait	Animal	Col1	Col2	Col3	Col4
1	Trait*Animal	1	1	2.5000	-1.0000	-1.0000	-1.6667
2	Trait*Animal	1	2	-1.0000	2.0000		0.6667
3	Trait*Animal	1	3	-1.0000		2.0000	0.6667
4	Trait*Animal	2	1	-1.6667	0.6667	0.6667	1.6667
5	Trait*Animal	2	2	0.6667	-1.3333		-0.6667
6	Trait*Animal	2	3	0.6667		-1.3333	-0.6667

Estimated Inv(G) Matrix

Row	Col5	Col6
1	0.6667	0.6667
2	-1.3333	
3		-1.3333
4	-0.6667	-0.6667
5	1.3333	
6		1.3333

The table of covariance parameter estimates in Output 56.4.7 displays only the parameters in **R**. Because of the GDATA= option in the RANDOM statement, the G-side parameters do not partici-

pate in the parameter estimation process. Because of the NOITER option in the PARMS statement, however, the R-side parameters in this output are identical to their starting values.

Output 56.4.7 R-Side Covariance Parameters

Covariance Parameter Estimates

Cov Parm	Subject	Estimate
UN(1,1)	Animal	4.0000
UN(2,1)	Animal	1.0000
UN(2,2)	Animal	5.0000

The coefficients of the mixed model equations in Output 56.4.8 agree with Henderson (1984, p. 55). Recall from Output 56.4.1 that there are 2 columns in **X** and 6 columns in **Z**. The first 8 columns of the mixed model equations correspond to the **X** and **Z** components. Column 9 represents the Y border.

Output 56.4.8 Mixed Model Equations with Y Border

Mixed Model Equations

Row	Effect	Trait	Animal	Col1	Col2	Col3	Col4
1	Trait	1		0.7763	-0.1053	0.2632	0.2632
2	Trait	2		-0.1053	0.4211	-0.05263	-0.05263
3	Trait*Animal	1	1	0.2632	-0.05263	2.7632	-1.0000
4	Trait*Animal	1	2	0.2632	-0.05263	-1.0000	2.2632
5	Trait*Animal	1	3	0.2500		-1.0000	
6	Trait*Animal	2	1	-0.05263	0.2105	-1.7193	0.6667
7	Trait*Animal	2	2	-0.05263	0.2105	0.6667	-1.3860
8	Trait*Animal	2	3			0.6667	

Mixed Model Equations

Row	Col5	Col6	Col7	Col8	Col9
1	0.2500	-0.05263	-0.05263		4.6974
2		0.2105	0.2105		2.2105
3	-1.0000	-1.7193	0.6667	0.6667	1.1053
4		0.6667	-1.3860		1.8421
5	2.2500	0.6667		-1.3333	1.7500
6	0.6667	1.8772	-0.6667	-0.6667	1.5789
7		-0.6667	1.5439		0.6316
8	-1.3333	-0.6667		1.3333	

The solution to the mixed model equations also matches that given by Henderson (1984, p. 55). After solving the augmented mixed model equations, you can find the solutions for fixed and random effects in the last column (Output 56.4.9).

Output 56.4.9 Solutions of the Mixed Model Equations with Y Border

Mixed Model Equations Solution

Row	Effect	Trait	Animal	Col1	Col2	Col3	Col4
1	Trait	1		2.5508	1.5685	-1.3047	-1.1775
2	Trait	2		1.5685	4.5539	-1.4112	-1.3534
3	Trait*Animal	1	1	-1.3047	-1.4112	1.8282	1.0652
4	Trait*Animal	1	2	-1.1775	-1.3534	1.0652	1.7589
5	Trait*Animal	1	3	-1.1701	-0.9410	1.0206	0.7085
6	Trait*Animal	2	1	-1.3002	-2.1592	1.8010	1.0900
7	Trait*Animal	2	2	-1.1821	-2.1055	1.0925	1.7341
8	Trait*Animal	2	3	-1.1678	-1.3149	1.0070	0.7209

Mixed Model Equations Solution

Row	Col5	Col6	Col7	Col8	Col9
1	-1.1701	-1.3002	-1.1821	-1.1678	6.9909
2	-0.9410	-2.1592	-2.1055	-1.3149	6.9959
3	1.0206	1.8010	1.0925	1.0070	0.05450
4	0.7085	1.0900	1.7341	0.7209	-0.04955
5	1.7812	1.0095	0.7197	1.7756	0.02230
6	1.0095	2.7518	1.6392	1.4849	0.2651
7	0.7197	1.6392	2.6874	0.9930	-0.2601
8	1.7756	1.4849	0.9930	2.7645	0.1276

The solutions for the fixed and random effects in Output 56.4.10 correspond to the last column in Output 56.4.9. Note that the standard errors for the fixed effects and the prediction standard errors for the random effects are the square root values of the diagonal entries in the solution of the mixed model equations (Output 56.4.9).

Output 56.4.10 Solutions for Fixed and Random Effects

Solution for Fixed Effects

Effect	Trait	Estimate	Standard Error	DF	t Value	Pr > |t|
Trait	1	6.9909	1.5971	3	4.38	0.0221
Trait	2	6.9959	2.1340	3	3.28	0.0465

Solution for Random Effects

Effect	Trait	Animal	Estimate	Std Err Pred	DF	t Value	Pr > |t|
Trait*Animal	1	1	0.05450	1.3521	0	0.04	.
Trait*Animal	1	2	-0.04955	1.3262	0	-0.04	.
Trait*Animal	1	3	0.02230	1.3346	0	0.02	.
Trait*Animal	2	1	0.2651	1.6589	0	0.16	.
Trait*Animal	2	2	-0.2601	1.6393	0	-0.16	.
Trait*Animal	2	3	0.1276	1.6627	0	0.08	.

The estimates for the two traits are nearly identical, but the standard error of the second trait is larger because of the missing observation.

The Estimate column in the "Solution for Random Effects" table lists the best linear unbiased predictions (BLUPs) of the breeding values of both traits for all three animals. The *p*-values are missing because the default containment method for computing degrees of freedom results in zero degrees of freedom for the random effects parameter tests.

Output 56.4.11 Significance Test Comparing Traits

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Trait	2	3	10.59	0.0437

The two estimated traits are significantly different from zero at the 5% level (Output 56.4.11).

Output 56.4.12 displays the predicted values of the observations based on the trait and breeding value estimates—that is, the fixed and random effects.

Output 56.4.12 Predicted Observations

Obs	Trait	Animal	Y	Pred	StdErr Pred	DF	Alpha	Lower	Upper	Resid
1	1	1	6	7.04542	1.33027	0	0.05	.	.	-1.04542
2	1	2	8	6.94137	1.39806	0	0.05	.	.	1.05863
3	1	3	7	7.01321	1.41129	0	0.05	.	.	-0.01321
4	2	1	9	7.26094	1.72839	0	0.05	.	.	1.73906
5	2	2	5	6.73576	1.74077	0	0.05	.	.	-1.73576
6	2	3	.	7.12015	2.99088	0	0.05	.	.	.

The predicted values are not the predictions of future records in the sense that they do not contain a component corresponding to a new observational error. See Henderson (1984) for information about predicting future records. The Lower and Upper columns usually contain confidence limits for the predicted values; they are missing here because the random-effects parameter degrees of freedom equals 0.

Example 56.5: Random Coefficients

This example comes from a pharmaceutical stability data simulation performed by Obenchain (1990). The observed responses are replicate assay results, expressed in percent of label claim, at various shelf ages, expressed in months. The desired mixed model involves three batches of product that differ randomly in intercept (initial potency) and slope (degradation rate). This type of model is also known as a hierarchical or multilevel model (Singer 1998; Sullivan, Dukes, and Losina 1999).

The SAS statements are as follows:

```
data rc;
  input Batch Month @@;
  Monthc = Month;
  do i = 1 to 6;
     input Y @@;
     output;
  end;
  datalines;
 1   0  101.2 103.3 103.3 102.1 104.4 102.4
 1   1   98.8  99.4  99.7  99.5    .     .
 1   3   98.4  99.0  97.3  99.8    .     .
 1   6  101.5 100.2 101.7 102.7    .     .
 1   9   96.3  97.2  97.2  96.3    .     .
 1  12   97.3  97.9  96.8  97.7  97.7  96.7
 2   0  102.6 102.7 102.4 102.1 102.9 102.6
 2   1   99.1  99.0  99.9 100.6    .     .
 2   3  105.7 103.3 103.4 104.0    .     .
 2   6  101.3 101.5 100.9 101.4    .     .
 2   9   94.1  96.5  97.2 95.6     .     .
 2  12   93.1  92.8  95.4 92.2   92.2  93.0
 3   0  105.1 103.9 106.1 104.1 103.7 104.6
 3   1  102.2 102.0 100.8  99.8    .     .
 3   3  101.2 101.8 100.8 102.6    .     .
 3   6  101.1 102.0 100.1 100.2    .     .
 3   9  100.9  99.5 102.2 100.8    .     .
 3  12   97.8  98.3  96.9  98.4  96.9  96.5
 ;

proc mixed data=rc;
   class Batch;
   model Y = Month / s;
   random Int Month / type=un sub=Batch s;
run;
```

In the DATA step, Monthc is created as a duplicate of Month in order to enable both a continuous and a classification version of the same variable. The variable Monthc is used in a subsequent analysis

In the PROC MIXED statements, Batch is listed as the only classification variable. The fixed effect Month in the MODEL statement is not declared as a classification variable; thus it models a linear trend in time. An intercept is included as a fixed effect by default, and the S option requests that the

fixed-effects parameter estimates be produced.

The two random effects are Int and Month, modeling random intercepts and slopes, respectively. Note that Intercept and Month are used as both fixed and random effects. The TYPE=UN option in the RANDOM statement specifies an unstructured covariance matrix for the random intercept and slope effects. In mixed model notation, **G** is block diagonal with unstructured 2×2 blocks. Each block corresponds to a different level of Batch, which is the SUBJECT= effect. The unstructured type provides a mechanism for estimating the correlation between the random coefficients. The S option requests the production of the random-effects parameter estimates.

The results from this analysis are shown in Output 56.5.1–Output 56.5.9. The "Unstructured" covariance structure in Output 56.5.1 applies to **G** here.

Output 56.5.1 Model Information in Random Coefficients Analysis

The Mixed Procedure

Model Information

Data Set	WORK.RC
Dependent Variable	Y
Covariance Structure	Unstructured
Subject Effect	Batch
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

Batch is the only classification variable in this analysis, and it has three levels (Output 56.5.2).

Output 56.5.2 Random Coefficients Analysis (*continued*)

Class Level Information

Class	Levels	Values
Batch	3	1 2 3

The "Dimensions" table in Output 56.5.3 indicates that there are three subjects (corresponding to batches). The 24 observations not used correspond to the missing values of Y in the input data set.

Output 56.5.3 Random Coefficients Analysis (*continued*)

Dimensions

Covariance Parameters	4
Columns in X	2
Columns in Z Per Subject	2
Subjects	3
Max Obs Per Subject	36

Output 56.5.3 *continued*

Number of Observations	
Number of Observations Read	108
Number of Observations Used	84
Number of Observations Not Used	24

As Output 56.5.4 shows, only one iteration is required for convergence.

Output 56.5.4 Random Coefficients Analysis (*continued*)

Iteration History

Iteration	Evaluations	−2 Res Log Like	Criterion
0	1	367.02768461	
1	1	350.32813577	0.00000000

Convergence criteria met.

The Estimate column in Output 56.5.5 lists the estimated elements of the unstructured 2×2 matrix comprising the blocks of $\mathbf{G}$. Note that the random coefficients are negatively correlated.

Output 56.5.5 Random Coefficients Analysis (*continued*)

Covariance Parameter Estimates

Cov Parm	Subject	Estimate
UN(1,1)	Batch	0.9768
UN(2,1)	Batch	−0.1045
UN(2,2)	Batch	0.03717
Residual		3.2932

The null model likelihood ratio test indicates a significant improvement over the null model consisting of no random effects and a homogeneous residual error (Output 56.5.6).

Output 56.5.6 Random Coefficients Analysis (*continued*)

Fit Statistics

−2 Res Log Likelihood	350.3
AIC (smaller is better)	358.3
AICC (smaller is better)	358.8
BIC (smaller is better)	354.7

Output 56.5.6 *continued*

Null Model Likelihood Ratio Test

DF	Chi-Square	Pr > ChiSq
3	16.70	0.0008

The fixed-effects estimates represent the estimated means for the random intercept and slope, respectively (Output 56.5.7).

Output 56.5.7 Random Coefficients Analysis (*continued*)

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	102.70	0.6456	2	159.08	<.0001
Month	−0.5259	0.1194	2	−4.41	0.0478

The random-effects estimates represent the estimated deviation from the mean intercept and slope for each batch (Output 56.5.8). Therefore, the intercept for the first batch is close to $102.7 - 1 = 101.7$, while the intercepts for the other two batches are greater than 102.7. The second batch has a slope less than the mean slope of -0.526, while the other two batches have slopes greater than -0.526.

Output 56.5.8 Random Coefficients Analysis (*continued*)

Solution for Random Effects

Effect	Batch	Estimate	Std Err Pred	DF	t Value	Pr > |t|
Intercept	1	−1.0010	0.6842	78	−1.46	0.1474
Month	1	0.1287	0.1245	78	1.03	0.3047
Intercept	2	0.3934	0.6842	78	0.58	0.5669
Month	2	−0.2060	0.1245	78	−1.65	0.1021
Intercept	3	0.6076	0.6842	78	0.89	0.3772
Month	3	0.07731	0.1245	78	0.62	0.5365

The F statistic in the "Type 3 Tests of Fixed Effects" table in Output 56.5.9 is the square of the t statistic used in the test of Month in the preceding "Solution for Fixed Effects" table (compare Output 56.5.7 and Output 56.5.9). Both statistics test the null hypothesis that the slope assigned to Month equals 0, and this hypothesis can barely be rejected at the 5% level.

Output 56.5.9 Random Coefficients Analysis (*continued*)

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Month	1	2	19.41	0.0478

It is also possible to fit a random coefficients model with error terms that follow a nested structure (Fuller and Battese 1973). The following SAS statements represent one way of doing this:

```
proc mixed data=rc;
   class Batch Monthc;
   model Y = Month / s;
   random Int Month Monthc / sub=Batch s;
run;
```

The variable Monthc is added to the CLASS and RANDOM statements, and it models the nested errors. Note that Month and Monthc are continuous and classification versions of the same variable. Also, the TYPE=UN option is dropped from the RANDOM statement, resulting in the default variance components model instead of correlated random coefficients. The results from this analysis are shown in Output 56.5.10.

Output 56.5.10 Random Coefficients with Nested Errors Analysis

The Mixed Procedure

Model Information

Data Set	WORK.RC
Dependent Variable	Y
Covariance Structure	Variance Components
Subject Effect	Batch
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

Class Level Information

Class	Levels	Values
Batch	3	1 2 3
Monthc	6	0 1 3 6 9 12

Dimensions

Covariance Parameters	4
Columns in X	2
Columns in Z Per Subject	8
Subjects	3
Max Obs Per Subject	36

Output 56.5.10 *continued*

Number of Observations

Number of Observations Read	108
Number of Observations Used	84
Number of Observations Not Used	24

Iteration History

Iteration	Evaluations	-2 Res Log Like	Criterion
0	1	367.02768461	
1	4	277.51945360	.
2	1	276.97551718	0.00104208
3	1	276.90304909	0.00003174
4	1	276.90100316	0.00000004
5	1	276.90100092	0.00000000

Convergence criteria met.

Covariance Parameter Estimates

Cov Parm	Subject	Estimate
Intercept	Batch	0
Month	Batch	0.01243
Monthc	Batch	3.7411
Residual		0.7969

For this analysis, the Newton-Raphson algorithm requires five iterations and nine likelihood evaluations to achieve convergence. The missing value in the Criterion column in iteration 1 indicates that a boundary constraint has been dropped.

The estimate for the Intercept variance component equals 0. This occurs frequently in practice and indicates that the restricted likelihood is maximized by setting this variance component equal to 0. Whenever a zero variance component estimate occurs, the following note appears in the SAS log:

```
NOTE: Estimated G matrix is not positive definite.
```

The remaining variance component estimates are positive, and the estimate corresponding to the nested errors (MONTHC) is much larger than the other two.

A comparison of AIC and BIC for this model with those of the previous model favors the nested error model (compare Output 56.5.11 and Output 56.5.6). Strictly speaking, a likelihood ratio test cannot be carried out between the two models because one is not contained in the other; however, a cautious comparison of likelihoods can be informative.

Output 56.5.11 Random Coefficients with Nested Errors Analysis (*continued*)

Fit Statistics

−2 Res Log Likelihood	276.9
AIC (smaller is better)	282.9
AICC (smaller is better)	283.2
BIC (smaller is better)	280.2

The better-fitting covariance model affects the standard errors of the fixed-effects parameter estimates more than the estimates themselves (Output 56.5.12).

Output 56.5.12 Random Coefficients with Nested Errors Analysis (*continued*)

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	102.56	0.7287	2	140.74	<.0001
Month	−0.5003	0.1259	2	−3.97	0.0579

The random-effects solution provides the empirical best linear unbiased predictions (EBLUPs) for the realizations of the random intercept, slope, and nested errors (Output 56.5.13). You can use these values to compare batches and months.

Output 56.5.13 Random Coefficients with Nested Errors Analysis (*continued*)

Solution for Random Effects

Effect	Batch	Monthc	Estimate	Std Err Pred	DF	t Value	Pr > |t|
Intercept	1		0	.	.	.	.
Month	1		-0.00028	0.09268	66	-0.00	0.9976
Monthc	1	0	0.2191	0.7896	66	0.28	0.7823
Monthc	1	1	-2.5690	0.7571	66	-3.39	0.0012
Monthc	1	3	-2.3067	0.6865	66	-3.36	0.0013
Monthc	1	6	1.8726	0.7328	66	2.56	0.0129
Monthc	1	9	-1.2350	0.9300	66	-1.33	0.1888
Monthc	1	12	0.7736	1.1992	66	0.65	0.5211
Intercept	2		0	.	.	.	.
Month	2		-0.07571	0.09268	66	-0.82	0.4169
Monthc	2	0	-0.00621	0.7896	66	-0.01	0.9938
Monthc	2	1	-2.2126	0.7571	66	-2.92	0.0048
Monthc	2	3	3.1063	0.6865	66	4.53	<.0001
Monthc	2	6	2.0649	0.7328	66	2.82	0.0064
Monthc	2	9	-1.4450	0.9300	66	-1.55	0.1250
Monthc	2	12	-2.4405	1.1992	66	-2.04	0.0459
Intercept	3		0	.	.	.	.
Month	3		0.07600	0.09268	66	0.82	0.4152
Monthc	3	0	1.9574	0.7896	66	2.48	0.0157
Monthc	3	1	-0.8850	0.7571	66	-1.17	0.2466
Monthc	3	3	0.3006	0.6865	66	0.44	0.6629
Monthc	3	6	0.7972	0.7328	66	1.09	0.2806
Monthc	3	9	2.0059	0.9300	66	2.16	0.0347
Monthc	3	12	0.002293	1.1992	66	0.00	0.9985

Output 56.5.14 Random Coefficients with Nested Errors Analysis (*continued*)

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Month	1	2	15.78	0.0579

The test of Month is similar to that from the previous model, although it is no longer significant at the 5% level (Output 56.5.14).

Example 56.6: Line-Source Sprinkler Irrigation

These data appear in Hanks et al. (1980), Johnson, Chaudhuri, and Kanemasu (1983), and Stroup (1989b). Three cultivars (Cult) of winter wheat are randomly assigned to rectangular plots within each of three blocks (Block). The nine plots are located side by side, and a line-source sprinkler is placed through the middle. Each plot is subdivided into twelve subplots—six to the north of the line source, six to the south (Dir). The two plots closest to the line source represent the maximum irrigation level (Irrig=6), the two next-closest plots represent the next-highest level (Irrig=5), and so forth.

This example is a case where both **G** and **R** can be modeled. One of Stroup's models specifies a diagonal **G** containing the variance components for Block, Block*Dir, and Block*Irrig, and a Toeplitz **R** with four bands. The SAS statements to fit this model and carry out some further analyses follow.

CAUTION: This analysis can require considerable CPU time.

```
data line;
  length Cult$ 8;
  input Block Cult$ @;
  row = _n_;
  do Sbplt=1 to 12;
    if Sbplt le 6 then do;
      Irrig = Sbplt;
      Dir = 'North';
    end; else do;
      Irrig = 13 - Sbplt;
      Dir = 'South';
    end;
    input Y @; output;
  end;
  datalines;
 1 Luke     2.4 2.7 5.6 7.5 7.9 7.1 6.1 7.3 7.4 6.7 3.8 1.8
 1 Nugaines 2.2 2.2 4.3 6.3 7.9 7.1 6.2 5.3 5.3 5.2 5.4 2.9
 1 Bridger  2.9 3.2 5.1 6.9 6.1 7.5 5.6 6.5 6.6 5.3 4.1 3.1
 2 Nugaines 2.4 2.2 4.0 5.8 6.1 6.2 7.0 6.4 6.7 6.4 3.7 2.2
 2 Bridger  2.6 3.1 5.7 6.4 7.7 6.8 6.3 6.2 6.6 6.5 4.2 2.7
 2 Luke     2.2 2.7 4.3 6.9 6.8 8.0 6.5 7.3 5.9 6.6 3.0 2.0
 3 Nugaines 1.8 1.9 3.7 4.9 5.4 5.1 5.7 5.0 5.6 5.1 4.2 2.2
 3 Luke     2.1 2.3 3.7 5.8 6.3 6.3 6.5 5.7 5.8 4.5 2.7 2.3
 3 Bridger  2.7 2.8 4.0 5.0 5.2 5.2 5.9 6.1 6.0 4.3 3.1 3.1
 ;

proc mixed;
   class Block Cult Dir Irrig;
   model Y = Cult|Dir|Irrig@2;
   random Block Block*Dir Block*Irrig;
   repeated / type=toep(4) sub=Block*Cult r;
   lsmeans Cult|Irrig;
   estimate 'Bridger vs Luke' Cult 1 -1 0;
   estimate 'Linear Irrig' Irrig -5 -3 -1 1 3 5;
   estimate 'B vs L x Linear Irrig' Cult*Irrig
            -5 -3 -1 1 3 5 5 3 1 -1 -3 -5;
run;
```

The preceding statements use the bar operator (|) and the at sign (@) to specify all two-factor interactions between Cult, Dir, and Irrig as fixed effects.

The RANDOM statement sets up the **Z** and **G** matrices corresponding to the random effects Block, Block*Dir, and Block*Irrig.

In the REPEATED statement, the TYPE=TOEP(4) option sets up the blocks of the **R** matrix to be Toeplitz with four bands below and including the main diagonal. The subject effect is Block*Cult, and it produces nine 12×12 blocks. The R option requests that the first block of **R** be displayed.

Least squares means (LSMEANS) are requested for Cult, Irrig, and Cult*Irrig, and a few ESTIMATE statements are specified to illustrate some linear combinations of the fixed effects.

The results from this analysis are shown in Output 56.6.1.

The "Covariance Structures" row in Output 56.6.1 reveals the two different structures assumed for **G** and **R**.

Output 56.6.1 Model Information in Line-Source Sprinkler Analysis

The Mixed Procedure

Model Information

Data Set	WORK.LINE
Dependent Variable	Y
Covariance Structures	Variance Components, Toeplitz
Subject Effect	Block*Cult
Estimation Method	REML
Residual Variance Method	Profile
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Containment

The levels of each classification variable are listed as a single string in the Values column, regardless of whether the levels are numeric or character (Output 56.6.2).

Output 56.6.2 Class Level Information

Class Level Information

Class	Levels	Values
Block	3	1 2 3
Cult	3	Bridger Luke Nugaines
Dir	2	North South
Irrig	6	1 2 3 4 5 6

Even though there is a SUBJECT= effect in the REPEATED statement, the analysis considers all of the data to be from one subject because there is no corresponding SUBJECT= effect in the RANDOM statement (Output 56.6.3).

Output 56.6.3 Model Dimensions and Number of Observations

Dimensions	
Covariance Parameters	7
Columns in X	48
Columns in Z	27
Subjects	1
Max Obs Per Subject	108

Number of Observations	
Number of Observations Read	108
Number of Observations Used	108
Number of Observations Not Used	0

The Newton-Raphson algorithm converges successfully in seven iterations (Output 56.6.4).

Output 56.6.4 Iteration History and Convergence Status

Iteration	Evaluations	-2 Res Log Like	Criterion
0	1	226.25427252	
1	4	187.99336173	.
2	3	186.62579299	0.10431081
3	1	184.38218213	0.04807260
4	1	183.41836853	0.00886548
5	1	183.25111475	0.00075353
6	1	183.23809997	0.00000748
7	1	183.23797748	0.00000000

Convergence criteria met.

The first block of the estimated **R** matrix has the TOEP(4) structure, and the observations that are three plots apart exhibit a negative correlation (Output 56.6.5).

Output 56.6.5 Estimated R Matrix for the First Subject

Estimated R Matrix for Block*Cult 1 Bridger

Row	Col1	Col2	Col3	Col4	Col5	Col6	Col7
1	0.2850	0.007986	0.001452	-0.09253			
2	0.007986	0.2850	0.007986	0.001452	-0.09253		
3	0.001452	0.007986	0.2850	0.007986	0.001452	-0.09253	
4	-0.09253	0.001452	0.007986	0.2850	0.007986	0.001452	-0.09253
5		-0.09253	0.001452	0.007986	0.2850	0.007986	0.001452
6			-0.09253	0.001452	0.007986	0.2850	0.007986
7				-0.09253	0.001452	0.007986	0.2850
8					-0.09253	0.001452	0.007986
9						-0.09253	0.001452
10							-0.09253
11							
12							

Estimated R Matrix for Block*Cult 1 Bridger

Row	Col8	Col9	Col10	Col11	Col12
1					
2					
3					
4					
5	-0.09253				
6	0.001452	-0.09253			
7	0.007986	0.001452	-0.09253		
8	0.2850	0.007986	0.001452	-0.09253	
9	0.007986	0.2850	0.007986	0.001452	-0.09253
10	0.001452	0.007986	0.2850	0.007986	0.001452
11	-0.09253	0.001452	0.007986	0.2850	0.007986
12		-0.09253	0.001452	0.007986	0.2850

Output 56.6.6 lists the estimated covariance parameters from both **G** and **R**. The first three are the variance components making up the diagonal **G**, and the final four make up the Toeplitz structure in the blocks of **R**. The Residual row corresponds to the variance of the Toeplitz structure, and it represents the parameter profiled out during the optimization process.

Output 56.6.6 Estimated Covariance Parameters

Covariance Parameter Estimates

Cov Parm	Subject	Estimate
Block		0.2194
Block*Dir		0.01768
Block*Irrig		0.03539
TOEP(2)	Block*Cult	0.007986
TOEP(3)	Block*Cult	0.001452
TOEP(4)	Block*Cult	-0.09253
Residual		0.2850

The "−2 Res Log Likelihood" value in Output 56.6.7 is the same as the final value listed in the "Iteration History" table (Output 56.6.4).

Output 56.6.7 Fit Statistics Based on the Residual Log Likelihood

Fit Statistics

−2 Res Log Likelihood	183.2
AIC (smaller is better)	197.2
AICC (smaller is better)	198.8
BIC (smaller is better)	190.9

Every fixed effect except for Dir and Cult*Irrig is significant at the 5% level (Output 56.6.8).

Output 56.6.8 Tests for Fixed Effects

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Cult	2	68	7.98	0.0008
Dir	1	2	3.95	0.1852
Cult*Dir	2	68	3.44	0.0379
Irrig	5	10	102.60	<.0001
Cult*Irrig	10	68	1.91	0.0580
Dir*Irrig	5	68	6.12	<.0001

The "Estimates" table lists the results from the various linear combinations of fixed effects specified in the ESTIMATE statements (Output 56.6.9). Bridger is not significantly different from Luke, and Irrig possesses a strong linear component. This strength appears to be influencing the significance of the interaction.

Output 56.6.9 Estimates

Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|
Bridger vs Luke	−0.03889	0.09524	68	−0.41	0.6843
Linear Irrig	30.6444	1.4412	10	21.26	<.0001
B vs L x Linear Irrig	−9.8667	2.7400	68	−3.60	0.0006

The least squares means shown in Output 56.6.10 are useful in comparing the levels of the various fixed effects. For example, it appears that irrigation levels 5 and 6 have virtually the same effect.

Output 56.6.10 Least Squares Means for Cult, Irrig, and Their Interaction

Least Squares Means

Effect	Cult	Irrig	Estimate	Standard Error	DF	t Value	Pr > |t|
Cult	Bridger		5.0306	0.2874	68	17.51	<.0001
Cult	Luke		5.0694	0.2874	68	17.64	<.0001
Cult	Nugaines		4.7222	0.2874	68	16.43	<.0001
Irrig		1	2.4222	0.3220	10	7.52	<.0001
Irrig		2	3.1833	0.3220	10	9.88	<.0001
Irrig		3	5.0556	0.3220	10	15.70	<.0001
Irrig		4	6.1889	0.3220	10	19.22	<.0001
Irrig		5	6.4000	0.3140	10	20.38	<.0001
Irrig		6	6.3944	0.3227	10	19.81	<.0001
Cult*Irrig	Bridger	1	2.8500	0.3679	68	7.75	<.0001
Cult*Irrig	Bridger	2	3.4167	0.3679	68	9.29	<.0001
Cult*Irrig	Bridger	3	5.1500	0.3679	68	14.00	<.0001
Cult*Irrig	Bridger	4	6.2500	0.3679	68	16.99	<.0001
Cult*Irrig	Bridger	5	6.3000	0.3463	68	18.19	<.0001
Cult*Irrig	Bridger	6	6.2167	0.3697	68	16.81	<.0001
Cult*Irrig	Luke	1	2.1333	0.3679	68	5.80	<.0001
Cult*Irrig	Luke	2	2.8667	0.3679	68	7.79	<.0001
Cult*Irrig	Luke	3	5.2333	0.3679	68	14.22	<.0001
Cult*Irrig	Luke	4	6.5500	0.3679	68	17.80	<.0001
Cult*Irrig	Luke	5	6.8833	0.3463	68	19.87	<.0001
Cult*Irrig	Luke	6	6.7500	0.3697	68	18.26	<.0001
Cult*Irrig	Nugaines	1	2.2833	0.3679	68	6.21	<.0001
Cult*Irrig	Nugaines	2	3.2667	0.3679	68	8.88	<.0001
Cult*Irrig	Nugaines	3	4.7833	0.3679	68	13.00	<.0001
Cult*Irrig	Nugaines	4	5.7667	0.3679	68	15.67	<.0001
Cult*Irrig	Nugaines	5	6.0167	0.3463	68	17.37	<.0001
Cult*Irrig	Nugaines	6	6.2167	0.3697	68	16.81	<.0001

An interesting exercise is to fit other variance-covariance models to these data and to compare them to this one by using likelihood ratio tests, Akaike's information criterion, or Schwarz's Bayesian information criterion. In particular, some spatial models are worth investigating (Marx and Thompson 1987; Zimmerman and Harville 1991). The following is one example of spatial model statements:

```
proc mixed;
   class Block Cult Dir Irrig;
   model Y = Cult|Dir|Irrig@2;
   repeated / type=sp(pow)(Row Sbplt) sub=intercept;
run;
```

The TYPE=SP(POW)(Row Sbplt) option in the REPEATED statement requests the spatial power structure, with the two defining coordinate variables being Row and Sbplt. The SUB=INTERCEPT option indicates that the entire data set is to be considered as one subject, thereby modeling **R** as a dense 108×108 covariance matrix. See Wolfinger (1993) for further discussion of this example and additional analyses.

Example 56.7: Influence in Heterogeneous Variance Model

In this example from Snedecor and Cochran (1976, p. 256), a one-way classification model with heterogeneous variances is fit. The data, shown in the following DATA step, represent amounts of different types of fat absorbed by batches of doughnuts during cooking, measured in grams.

```
data absorb;
  input FatType Absorbed @@;
  datalines;
 1 164  1 172  1 168  1 177  1 156  1 195
 2 178  2 191  2 197  2 182  2 185  2 177
 3 175  3 193  3 178  3 171  3 163  3 176
 4 155  4 166  4 149  4 164  4 170  4 168
;
```

The statistical model for these data can be written as

$$
\begin{aligned}
Y_{ij} &= \mu + \tau_i + \epsilon_{ij} \\
i &= 1, \cdots, t = 4 \\
j &= 1, \cdots, r = 6 \\
\epsilon_{ij} &= N(0, \sigma_i^2)
\end{aligned}
$$

where Y_{ij} is the amount of fat absorbed by the jth batch of the ith fat type, and τ_i denotes the fat-type effects. A quick glance at the data suggests that observations 6, 9, 14, and 21 might be influential on the analysis, because these are extreme observations for the respective fat types.

The following SAS statements fit this model and request influence diagnostics for the fixed effects and covariance parameters. The ODS GRAPHICS statement requests plots of the influence diagnostics in addition to the tabular output. The ESTIMATES suboption requests plots of "leave-one-out" estimates for the fixed effects and group variances.

```
ods graphics on;

proc mixed data=absorb asycov;
   class FatType;
   model Absorbed = FatType / s
                    influence(iter=10 estimates);
   repeated / group=FatType;
   ods output Influence=inf;
run;

ods graphics off;
```

The "Influence" table is output to the SAS data set inf so that parameter estimates can be printed subsequently. Results from this analysis are shown in Output 56.7.1.

Output 56.7.1 Heterogeneous Variance Analysis

The Mixed Procedure

Model Information

Data Set	WORK.ABSORB
Dependent Variable	Absorbed
Covariance Structure	Variance Components
Group Effect	FatType
Estimation Method	REML
Residual Variance Method	None
Fixed Effects SE Method	Model-Based
Degrees of Freedom Method	Between-Within

Covariance Parameter Estimates

Cov Parm	Group	Estimate
Residual	FatType 1	178.00
Residual	FatType 2	60.4000
Residual	FatType 3	97.6000
Residual	FatType 4	67.6000

Solution for Fixed Effects

Effect	Fat Type	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept		162.00	3.3566	20	48.26	<.0001
FatType	1	10.0000	6.3979	20	1.56	0.1337
FatType	2	23.0000	4.6188	20	4.98	<.0001
FatType	3	14.0000	5.2472	20	2.67	0.0148
FatType	4	0	.	.	.	.

The fixed-effects solutions correspond to estimates of the following parameters:

$$\text{Intercept} : \mu + \tau_4$$
$$\text{FatType1} : \tau_1 - \tau_4$$
$$\text{FatType2} : \tau_2 - \tau_4$$
$$\text{FatType3} : \tau_3 - \tau_4$$
$$\text{FatType4} : 0$$

You can easily verify that these estimates are simple functions of the arithmetic means $\overline{y}_{i.}$ in the groups. For example, $\widehat{\mu + \tau_4} = \overline{y}_{4.} = 162.0$, $\widehat{\tau_1 - \tau_4} = \overline{y}_{1.} - \overline{y}_{4.} = 10.0$, and so forth. The covariance parameter estimates are the sample variances in the groups and are uncorrelated.

The variances in the four groups are shown in the "Covariance Parameter Estimates" table (Output 56.7.1). The estimated variance in the first group is two to three times larger than the variance in the other groups.

Output 56.7.2 Asymptotic Variances of Group Variance Estimates

Asymptotic Covariance Matrix of Estimates

Row	Cov Parm	CovP1	CovP2	CovP3	CovP4
1	Residual	12674			
2	Residual		1459.26		
3	Residual			3810.30	
4	Residual				1827.90

In groups where the residual variance estimate is large, the precision of the estimate is also small (Output 56.7.2).

The following statements print the "leave-one-out" estimates for fixed effects and covariance parameters that were written to the inf data set with the ESTIMATES suboption (Output 56.7.3):

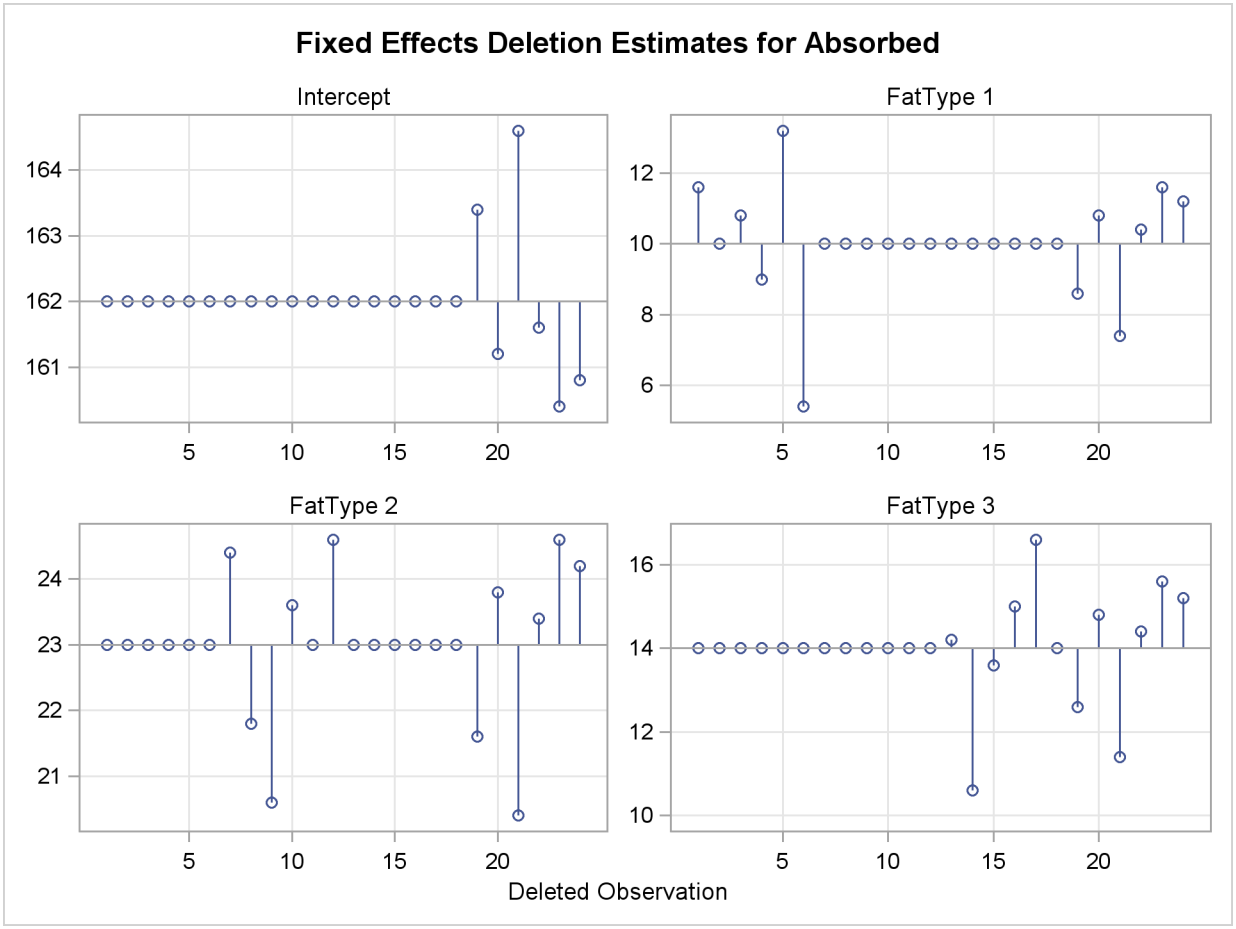
```
proc print data=inf label;
   var parm1-parm5 covp1-covp4;
run;
```

Output 56.7.3 Leave-One-Out Estimates

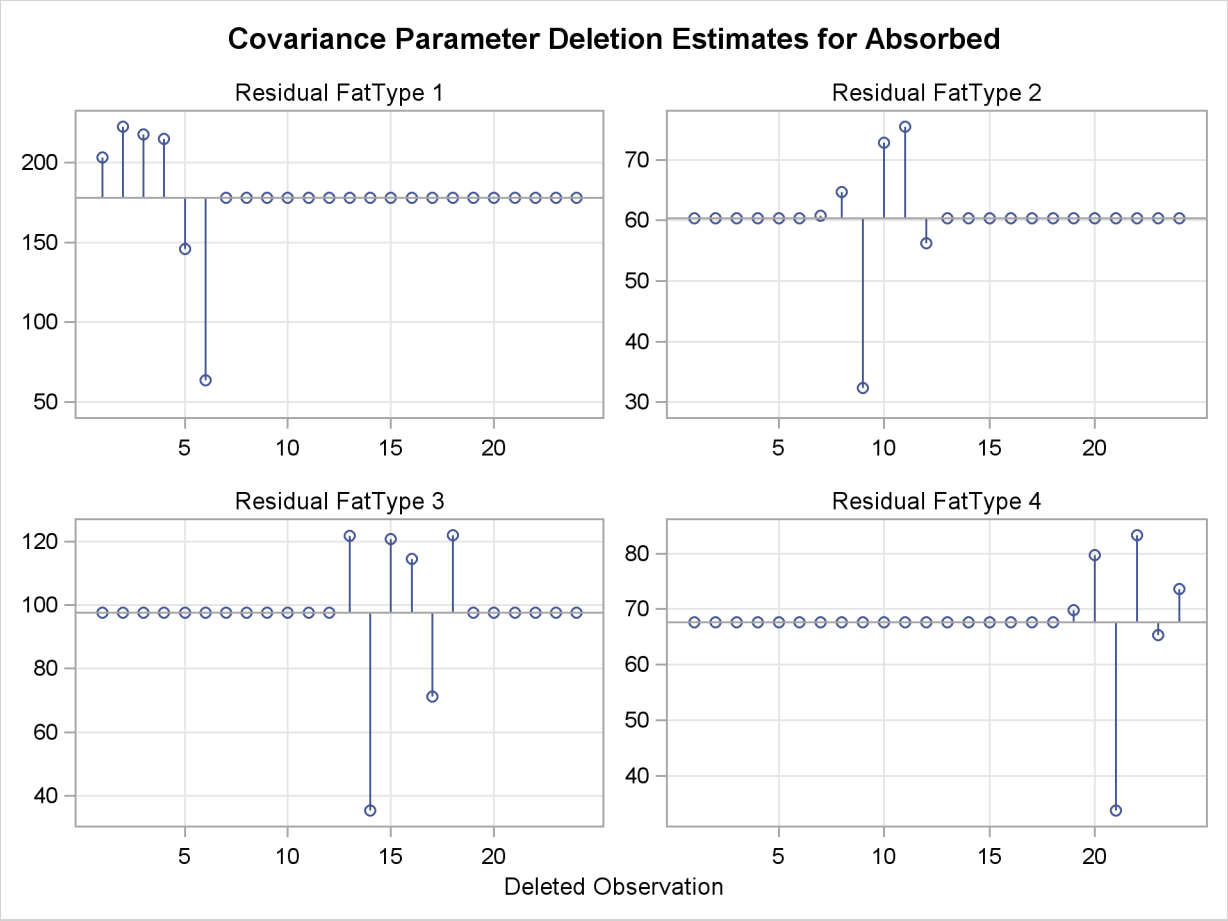
Obs	Intercept	Fat Type 1	Fat Type 2	Fat Type 3	Fat Type 4	Residual FatType 1	Residual FatType 2	Residual FatType 3	Residual FatType 4
1	162.00	11.600	23.000	14.000	0	203.30	60.400	97.60	67.600
2	162.00	10.000	23.000	14.000	0	222.47	60.400	97.60	67.600
3	162.00	10.800	23.000	14.000	0	217.68	60.400	97.60	67.600
4	162.00	9.000	23.000	14.000	0	214.99	60.400	97.60	67.600
5	162.00	13.200	23.000	14.000	0	145.70	60.400	97.60	67.600
6	162.00	5.400	23.000	14.000	0	63.80	60.400	97.60	67.600
7	162.00	10.000	24.400	14.000	0	178.00	60.795	97.60	67.600
8	162.00	10.000	21.800	14.000	0	178.00	64.691	97.60	67.600
9	162.00	10.000	20.600	14.000	0	178.00	32.296	97.60	67.600
10	162.00	10.000	23.600	14.000	0	178.00	72.797	97.60	67.600
11	162.00	10.000	23.000	14.000	0	178.00	75.490	97.60	67.600
12	162.00	10.000	24.600	14.000	0	178.00	56.285	97.60	67.600
13	162.00	10.000	23.000	14.200	0	178.00	60.400	121.68	67.600
14	162.00	10.000	23.000	10.600	0	178.00	60.400	35.30	67.600
15	162.00	10.000	23.000	13.600	0	178.00	60.400	120.79	67.600
16	162.00	10.000	23.000	15.000	0	178.00	60.400	114.50	67.600
17	162.00	10.000	23.000	16.600	0	178.00	60.400	71.30	67.600
18	162.00	10.000	23.000	14.000	0	178.00	60.400	121.98	67.600
19	163.40	8.600	21.600	12.600	0	178.00	60.400	97.60	69.799
20	161.20	10.800	23.800	14.800	0	178.00	60.400	97.60	79.698
21	164.60	7.400	20.400	11.400	0	178.00	60.400	97.60	33.800
22	161.60	10.400	23.400	14.400	0	178.00	60.400	97.60	83.292
23	160.40	11.600	24.600	15.600	0	178.00	60.400	97.60	65.299
24	160.80	11.200	24.200	15.200	0	178.00	60.400	97.60	73.677

The graphical displays in Output 56.7.4 and Output 56.7.5 are requested by specifying the ODS GRAPHICS statement. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the MIXED procedure, see the section "ODS Graphics" on page 4004.

Output 56.7.4 Fixed-Effects Deletion Estimates



Output 56.7.5 Covariance Parameter Deletion Estimates



The estimate of the intercept is affected only when observations from the last group are removed. The estimate of the "FatType 1" effect reacts to removal of observations in the first and last group (Output 56.7.4).

While observations can affect one or more fixed-effects solutions in this model, they can affect only one covariance parameter, the variance in their group (Output 56.7.5). Observations 6, 9, 14, and 21, which are extreme in their group, reduce the group variance considerably.

Diagnostics related to residuals and predicted values are printed with the following statements:

```
proc print data=inf label;
   var observed predicted residual pressres
       student Rstudent;
run;
```

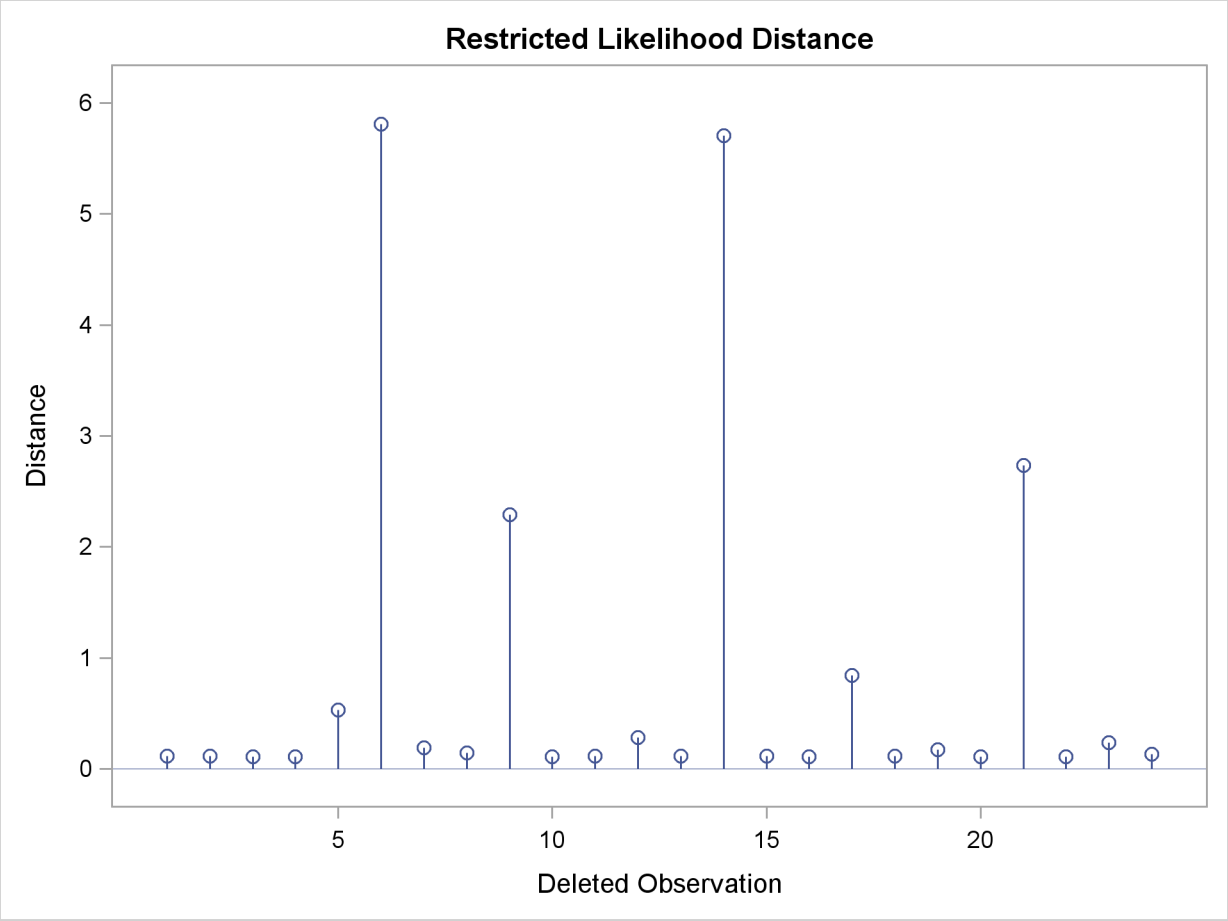
Output 56.7.6 Residual Diagnostics

Obs	Observed Value	Predicted Mean	Residual	PRESS Residual	Internally Studentized Residual	Externally Studentized Residual
1	164	172.0	-8.000	-9.600	-0.6569	-0.6146
2	172	172.0	0.000	0.000	0.0000	0.0000
3	168	172.0	-4.000	-4.800	-0.3284	-0.2970
4	177	172.0	5.000	6.000	0.4105	0.3736
5	156	172.0	-16.000	-19.200	-1.3137	-1.4521
6	195	172.0	23.000	27.600	1.8885	3.1544
7	178	185.0	-7.000	-8.400	-0.9867	-0.9835
8	191	185.0	6.000	7.200	0.8457	0.8172
9	197	185.0	12.000	14.400	1.6914	2.3131
10	182	185.0	-3.000	-3.600	-0.4229	-0.3852
11	185	185.0	0.000	-0.000	0.0000	0.0000
12	177	185.0	-8.000	-9.600	-1.1276	-1.1681
13	175	176.0	-1.000	-1.200	-0.1109	-0.0993
14	193	176.0	17.000	20.400	1.8850	3.1344
15	178	176.0	2.000	2.400	0.2218	0.1993
16	171	176.0	-5.000	-6.000	-0.5544	-0.5119
17	163	176.0	-13.000	-15.600	-1.4415	-1.6865
18	176	176.0	0.000	0.000	0.0000	0.0000
19	155	162.0	-7.000	-8.400	-0.9326	-0.9178
20	166	162.0	4.000	4.800	0.5329	0.4908
21	149	162.0	-13.000	-15.600	-1.7321	-2.4495
22	164	162.0	2.000	2.400	0.2665	0.2401
23	170	162.0	8.000	9.600	1.0659	1.0845
24	168	162.0	6.000	7.200	0.7994	0.7657

Observations 6, 9, 14, and 21 have large studentized residuals (Output 56.7.6). That the externally studentized residuals are much larger than the internally studentized residuals for these observations indicates that the variance estimate in the group shrinks when the observation is removed. Also important to note is that comparisons based on raw residuals in models with heterogeneous variance can be misleading. Observation 5, for example, has a larger residual but a smaller studentized residual than observation 21. The variance for the first fat type is much larger than the variance in the fourth group. A "large" residual is more "surprising" in the groups with small variance.

A measure of the overall influence on the analysis is the (restricted) likelihood distance, shown in Output 56.7.7. Observations 6, 9, 14, and 21 clearly displace the REML solution more than any other observations.

Output 56.7.7 Restricted Likelihood Distance



The following statements list the restricted likelihood distance and various diagnostics related to the fixed-effects estimates (Output 56.7.8):

```
proc print data=inf label;
   var leverage observed CookD DFFITS CovRatio RLD;
run;
```

Output 56.7.8 Restricted Likelihood Distance and Fixed-Effects Diagnostics

Obs	Leverage	Observed Value	Cook's D	DFFITS	COVRATIO	Restr. Likelihood Distance
1	0.167	164	0.02157	-0.27487	1.3706	0.1178
2	0.167	172	0.00000	-0.00000	1.4998	0.1156
3	0.167	168	0.00539	-0.13282	1.4675	0.1124
4	0.167	177	0.00843	0.16706	1.4494	0.1117
5	0.167	156	0.08629	-0.64938	0.9822	0.5290
6	0.167	195	0.17831	1.41069	0.4301	5.8101
7	0.167	178	0.04868	-0.43982	1.2078	0.1935
8	0.167	191	0.03576	0.36546	1.2853	0.1451
9	0.167	197	0.14305	1.03446	0.6416	2.2909
10	0.167	182	0.00894	-0.17225	1.4463	0.1116
11	0.167	185	0.00000	-0.00000	1.4998	0.1156
12	0.167	177	0.06358	-0.52239	1.1183	0.2856
13	0.167	175	0.00061	-0.04441	1.4961	0.1151
14	0.167	193	0.17766	1.40175	0.4340	5.7044
15	0.167	178	0.00246	0.08915	1.4851	0.1139
16	0.167	171	0.01537	-0.22892	1.4078	0.1129
17	0.167	163	0.10389	-0.75423	0.8766	0.8433
18	0.167	176	0.00000	0.00000	1.4998	0.1156
19	0.167	155	0.04349	-0.41047	1.2390	0.1710
20	0.167	166	0.01420	0.21950	1.4148	0.1124
21	0.167	149	0.15000	-1.09545	0.6000	2.7343
22	0.167	164	0.00355	0.10736	1.4786	0.1133
23	0.167	170	0.05680	0.48500	1.1592	0.2383
24	0.167	168	0.03195	0.34245	1.3079	0.1353

In this example, observations with large likelihood distances also have large values for Cook's D and values of CovRatio far less than one (Output 56.7.8). The latter indicates that the fixed effects are estimated more precisely when these observations are removed from the analysis.

The following statements print the values of the D statistic and the CovRatio for the covariance parameters:

```
proc print data=inf label;
   var iter CookDCP CovRatioCP;
run;
```

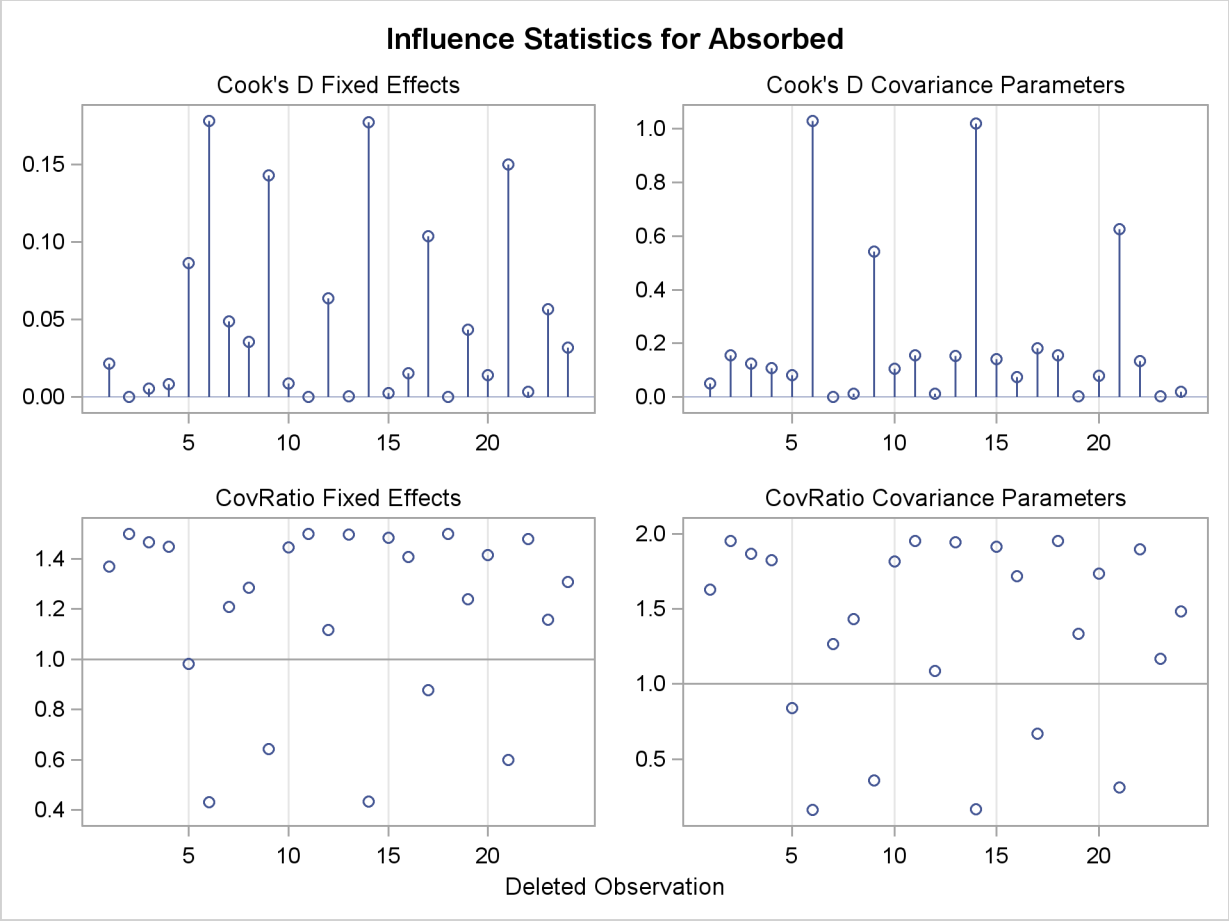
The same conclusions as for the fixed-effects estimates hold for the covariance parameter estimates. Observations 6, 9, 14, and 21 change the estimates and their precision considerably (Output 56.7.9, Output 56.7.10). All iterative updates converged within at most four iterations.

Output 56.7.9 Covariance Parameter Diagnostics

Obs	Iterations	Cook's D CovParms	COVRATIO CovParms
1	3	0.05050	1.6306
2	3	0.15603	1.9520
3	3	0.12426	1.8692
4	3	0.10796	1.8233
5	4	0.08232	0.8375
6	4	1.02909	0.1606
7	1	0.00011	1.2662
8	2	0.01262	1.4335
9	3	0.54126	0.3573
10	3	0.10531	1.8156
11	3	0.15603	1.9520
12	2	0.01160	1.0849
13	3	0.15223	1.9425
14	4	1.01865	0.1635
15	3	0.14111	1.9141
16	3	0.07494	1.7203
17	3	0.18154	0.6671
18	3	0.15603	1.9520
19	2	0.00265	1.3326
20	3	0.08008	1.7374
21	1	0.62500	0.3125
22	3	0.13472	1.8974
23	2	0.00290	1.1663
24	2	0.02020	1.4839

Output 56.7.10 displays the standard panel of influence diagnostics that is obtained when influence analysis is iterative. The Cook's D and CovRatio statistics are displayed for each deletion set for both fixed-effects and covariance parameter estimates. This provides a convenient summary of the impact on the analysis for each deletion set, since Cook's D statistic measures impact on the estimates and the CovRatio statistic measures impact on the precision of the estimates.

Output 56.7.10 Influence Diagnostics



Observations 6, 9, 14, and 21 have considerable impact on estimates and precision of fixed effects and covariance parameters. This is not necessarily the case. Observations can be influential on only some aspects of the analysis, as shown in the next example.

Example 56.8: Influence Analysis for Repeated Measures Data

This example revisits the repeated measures data of Pothoff and Roy (1964) that were analyzed in Example 56.2. Recall that the data consist of growth measurements at ages 8, 10, 12, and 14 for 11 girls and 16 boys. The model being fit contains fixed effects for Gender and Age and their interaction.

The earlier analysis of these data indicated some unusual observations in this data set. Because of the clustered data structure, it is of interest to study the influence of clusters (children) on the analysis rather than the influence of individual observations. A cluster comprises the repeated measurements for each child.

The repeated measures are first modeled with an unstructured within-child variance-covariance matrix. A residual variance is not profiled in this model. A noniterative influence analysis will update the fixed effects only. The following statements request this noniterative maximum likelihood analysis and produce Output 56.8.1:

```
proc mixed data=pr method=ml;
  class person gender;
  model y = gender age gender*age /
                  influence(effect=person);
  repeated / type=un subject=person;
  ods select influence;
run;
```

Output 56.8.1 Default Influence Statistics in Noniterative Analysis

The Mixed Procedure

Influence Diagnostics for Levels of Person

Person	Number of Observations in Level	PRESS Statistic	Cook's D
1	4	10.1716	0.01539
2	4	3.8187	0.03988
3	4	10.8448	0.02891
4	4	24.0339	0.04515
5	4	1.6900	0.01613
6	4	11.8592	0.01634
7	4	1.1887	0.00521
8	4	4.6717	0.02742
9	4	13.4244	0.03949
10	4	85.1195	0.13848
11	4	67.9397	0.09728
12	4	40.6467	0.04438
13	4	13.0304	0.00924
14	4	6.1712	0.00411
15	4	24.5702	0.12727
16	4	20.5266	0.01026
17	4	9.9917	0.01526
18	4	7.9355	0.01070
19	4	15.5955	0.01982
20	4	42.6845	0.01973
21	4	95.3282	0.10075
22	4	13.9649	0.03778
23	4	4.9656	0.01245
24	4	37.2494	0.15094
25	4	4.3756	0.03375
26	4	8.1448	0.03470
27	4	20.2913	0.02523

Each observation in the "Influence Diagnostics for Levels of Person" table in Output 56.8.1 represents the removal of four observations. The subjects 10, 15, and 24 have the greatest impact on the fixed effects (Cook's D), and subject 10 and 21 have large PRESS statistics. The 21st child has a

large PRESS statistic, and its D statistic is not that extreme. This is an indication that the model fits rather poorly for this child, whether it is part of the data or not.

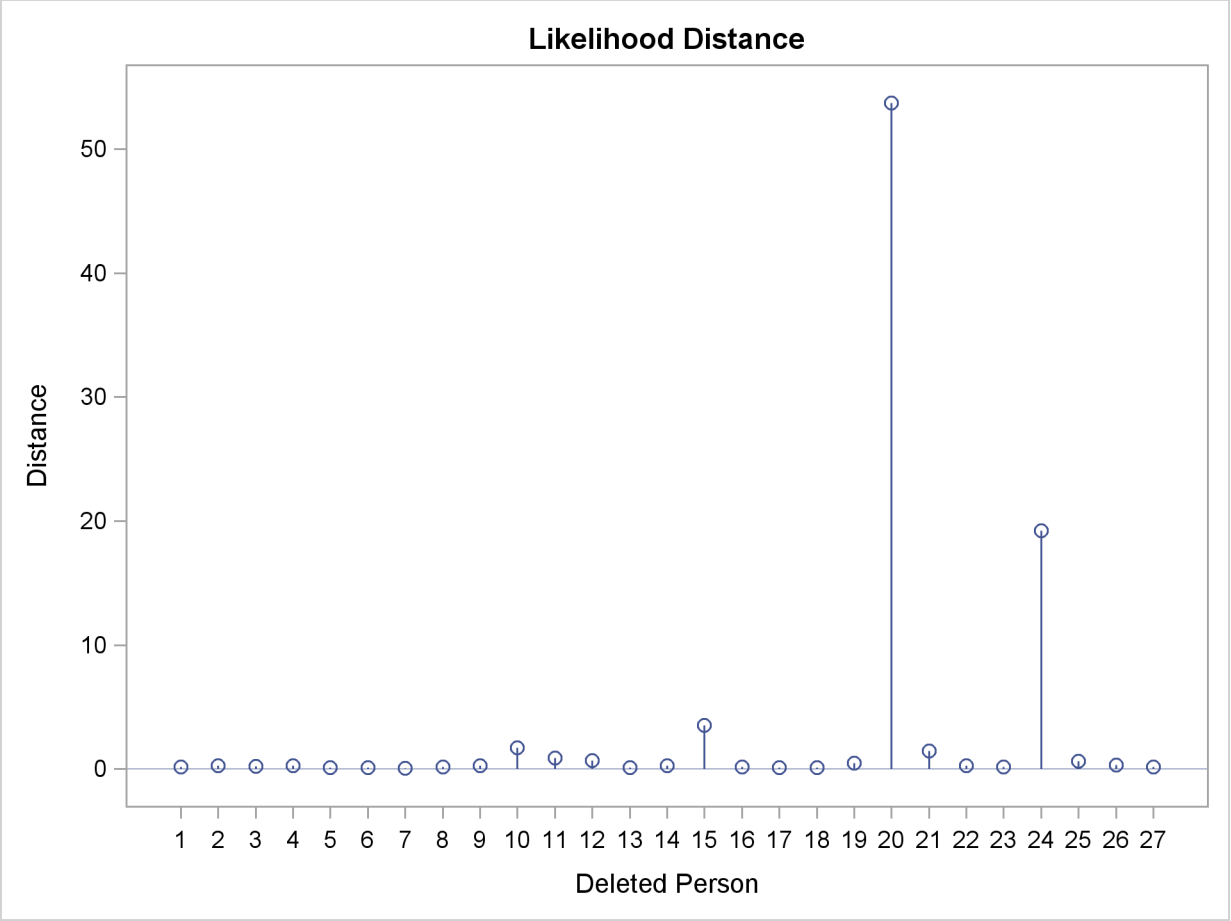
The previous analysis does not take into account the effect on the covariance parameters when a subject is removed from the analysis. If you also update the covariance parameters, the impact of observations on these can amplify or allay their effect on the fixed effects. To assess the overall influence of subjects on the analysis and to compute separate statistics for the fixed effects and covariance parameters, an iterative analysis is obtained by adding the INFLUENCE suboption ITER=, as follows:

```
ods graphics on;

proc mixed data=pr method=ml;
  class person gender;
  model y = gender age gender*age /
                   influence(effect=person iter=5);
  repeated / type=un subject=person;
run;
```

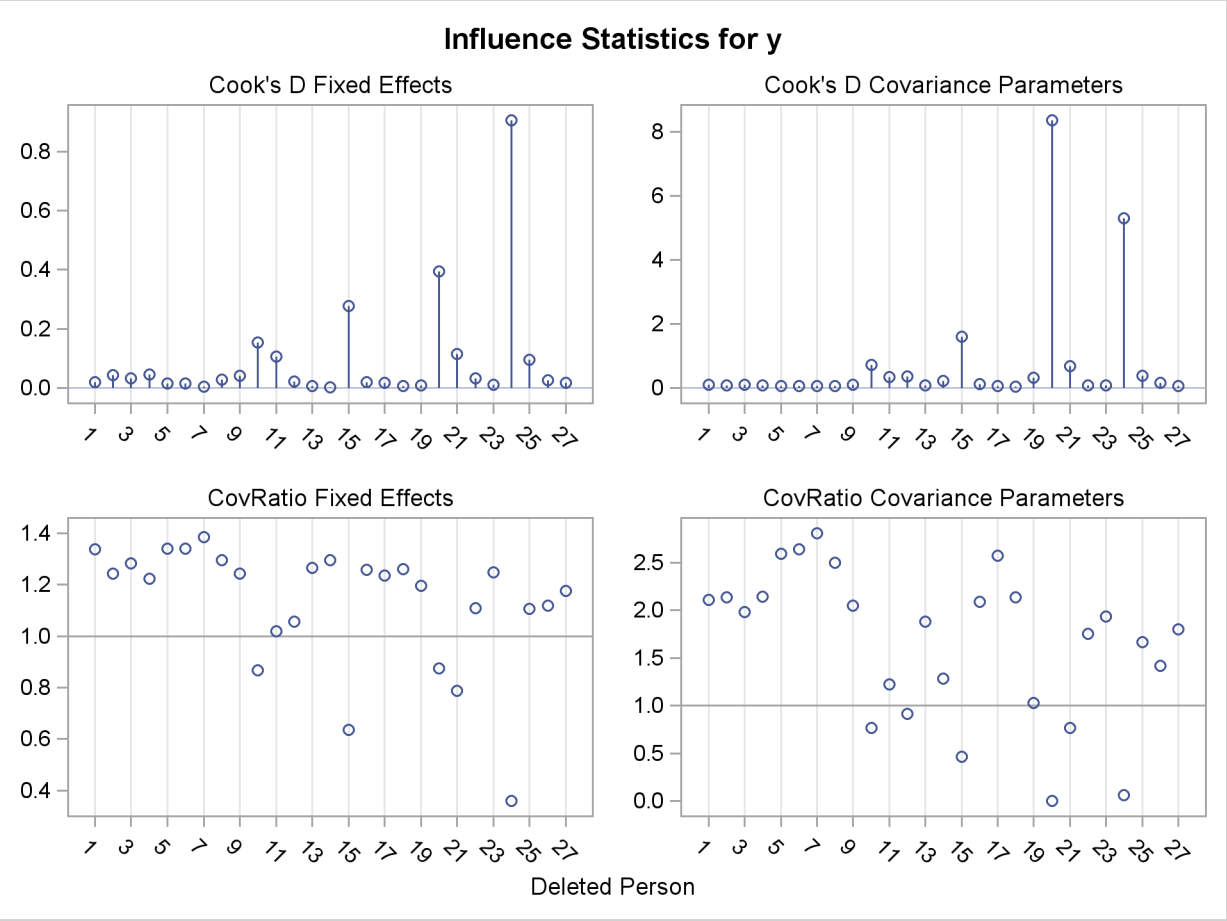
The number of additional iterations following removal of the observations for a particular subject is limited to five. Graphical displays of influence diagnostics are requested by specifying the ODS GRAPHICS statement. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the MIXED procedure, see the section "ODS Graphics" on page 4004.

The MIXED procedure produces a plot of the restricted likelihood distance (Output 56.8.2) and a panel of diagnostics for fixed effects and covariance parameters (Output 56.8.3).

Output 56.8.2 Restricted Likelihood Distance



Output 56.8.3 Influence Diagnostics Panel



As judged by the restricted likelihood distance, subjects 20 and 24 clearly have the most influence on the overall analysis (Output 56.8.2).

Output 56.8.3 displays Cook's D and CovRatio statistics for the fixed effects and covariance parameters. Clearly, subject 20 has a dramatic effect on the estimates of variances and covariances. This subject also affects the precision of the covariance parameter estimates more than any other subject in Output 56.8.3 (CovRatio near 0).

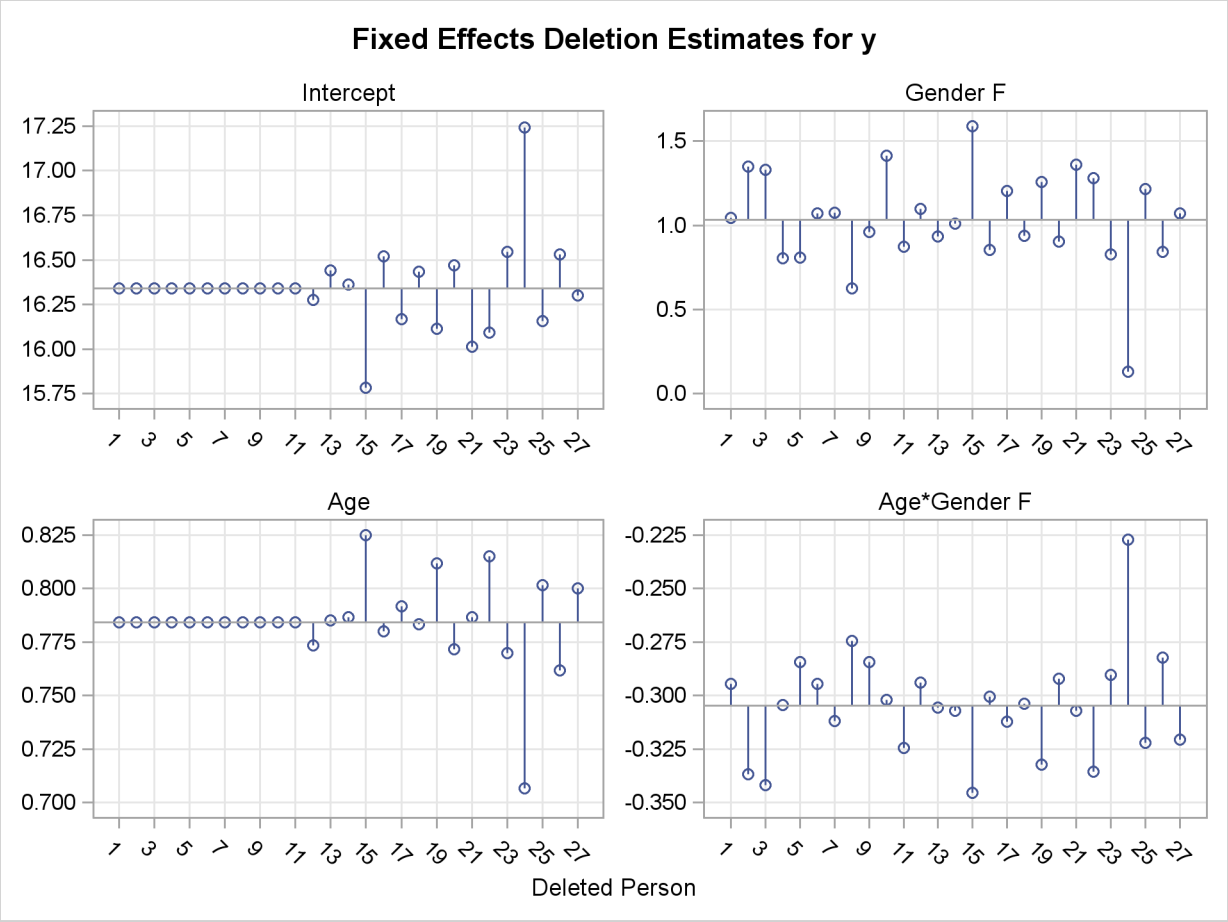
The child who exerts the greatest influence on the fixed effects is subject 24. Maybe surprisingly, this subject affects the variance-covariance matrix of the fixed effects more than subject 20 (small CovRatio in Output 56.8.3).

The final model investigated for these data is a random coefficient model as in Stram and Lee (1994) with random effects for the intercept and age effect. The following statements examine the estimates for fixed effects and the entries of the unstructured 2×2 variance matrix of the random coefficients graphically:

```
proc mixed data=pr method=ml
           plots(only)=InfluenceEstPlot;
   class person gender;
   model y = gender age gender*age /
                    influence(iter=5 effect=person est);
   random intercept age / type=un subject=person;
run;
```

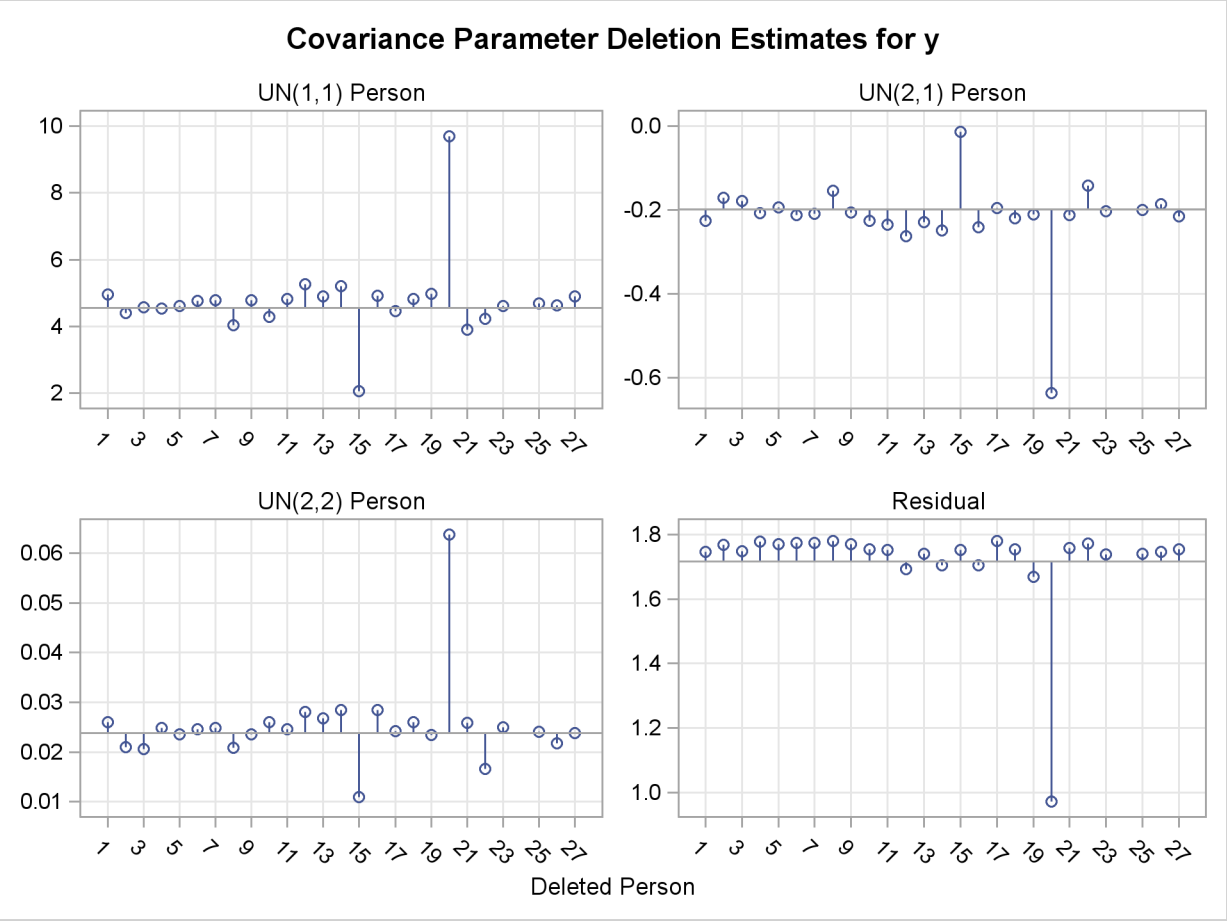
The PLOTS(ONLY)=INFLUENCEESTPLOT option restricts the graphical output from this PROC MIXED run to only the panels of deletion estimates (Output 56.8.4 and Output 56.8.5).

Output 56.8.4 Fixed-Effects Deletion Estimates



In Output 56.8.4 the graphs on the left side of the panel represent the intercept and slope estimate for boys; the graphs on the right side represent the difference in intercept and slope between boys and girls. Removing any one of the first eleven children, who are girls, does not alter the intercept or slope in the group of boys. The difference in these parameters between boys and girls is altered by the removal of any child. Subject 24 changes the fixed effects considerably, subject 20 much less so.

Output 56.8.5 Covariance Parameter Deletion Estimates



The covariance parameter deletion estimates in Output 56.8.5 show several important features.

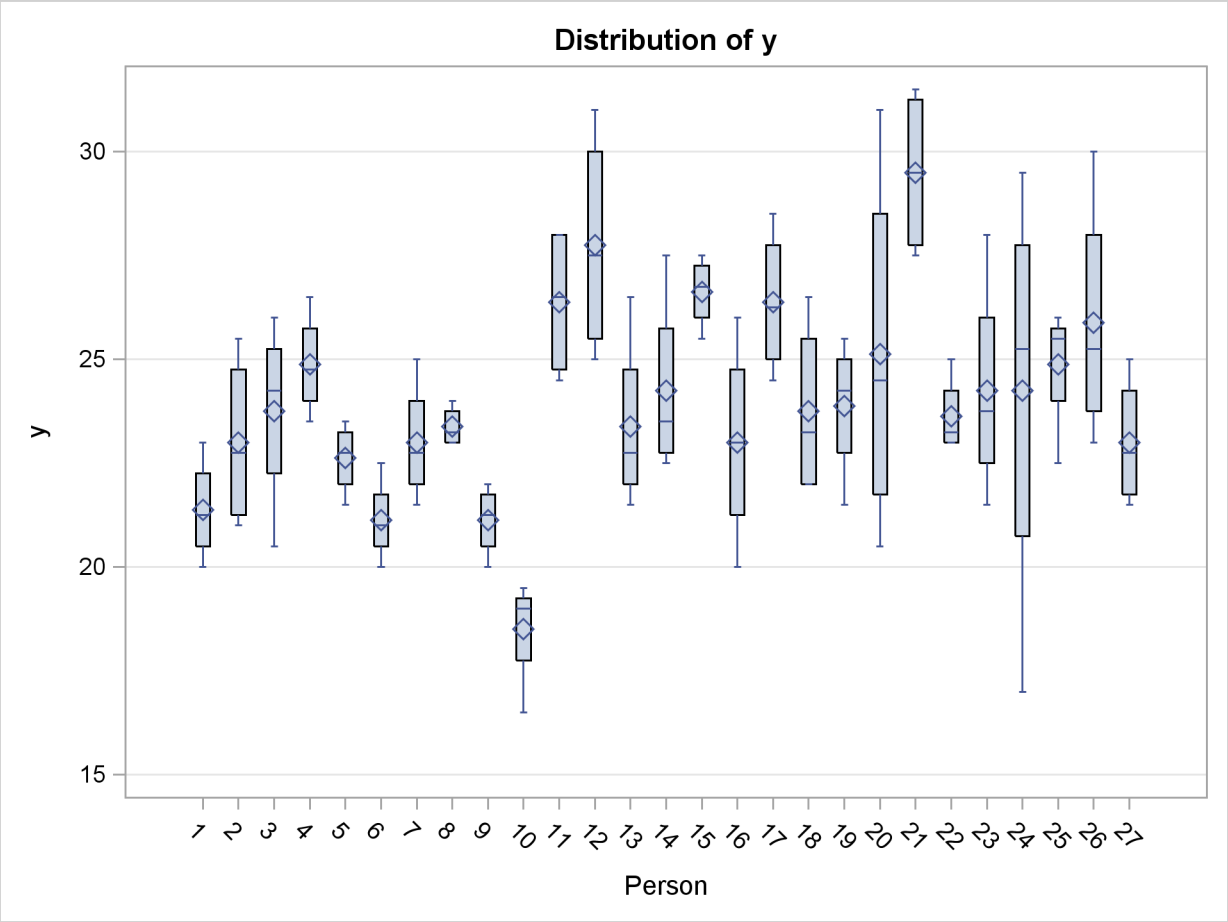
- The panels do not contain information about subject 24. Estimation of the **G** matrix following removal of that child did not yield a positive definite matrix. As a consequence, covariance parameter diagnostics are not produced for this subject.
- Subject 20 has great impact on the four covariance parameters. Removing this child from the analysis increases the variance of the random intercept and random slope and reduces the residual variance by almost 80%. The repeated measurements of this child exhibit an up-and-down behavior.
- The variance of the random intercept and slope are reduced when child 15 is removed from the analysis. This child's growth measurements oscillate about 27.0 from age 10 on.

Examining observed and residual values by levels of classification variables is also a useful tool to diagnose the adequacy of the model and unusual observations. Box plots for effects in the model that consist of only classification variables can be requested with the BOXPLOT option of the PLOT= option in the PROC MIXED statement. For example, the following statements produce box plots for the SUBJECT= effects in the model:

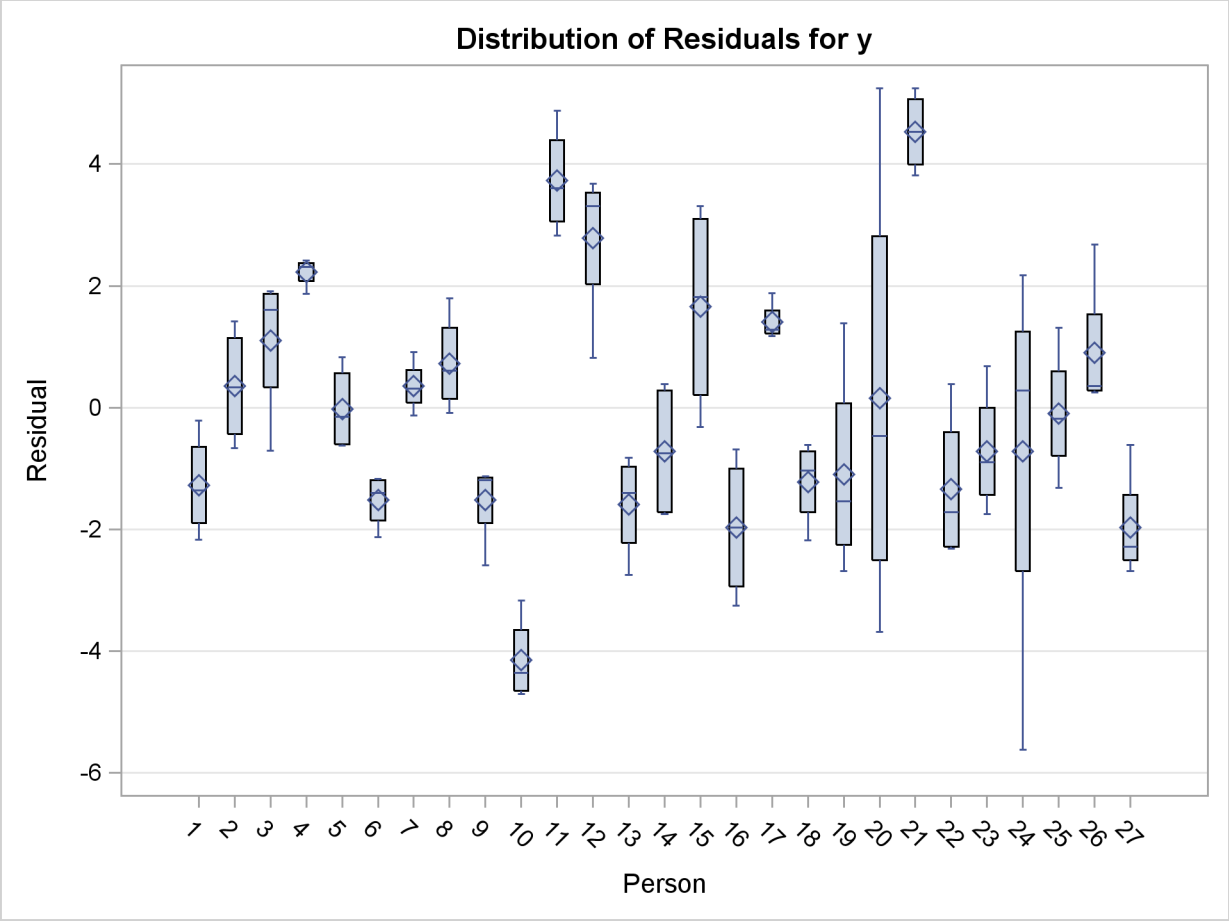
```
proc mixed data=pr method=ml
       plot=boxplot(observed marginal conditional subject);
   class person gender;
   model y = gender age gender*age;
   random intercept age / type=un subject=person;
run;
```

The specific boxplot options request a plot of the observed data (Output 56.8.6), the marginal residuals (Output 56.8.7), and the conditional residuals (Output 56.8.8). Box plots of the observed values show the variation within and between children clearly. The group of girls (subjects 1–11) is distinguishable from the group of boys by somewhat lesser average growth and lesser within-child variation (Output 56.8.6). After adjusting for overall (population-averaged) gender and age effects, the residual within-child variation is reduced but substantial differences in the means remain (Output 56.8.7). If child-specific inferences are desired, a model accounting for only Gender, Age, and Gender*Age effects is not adequate for these data.

Output 56.8.6 Distribution of Observed Values

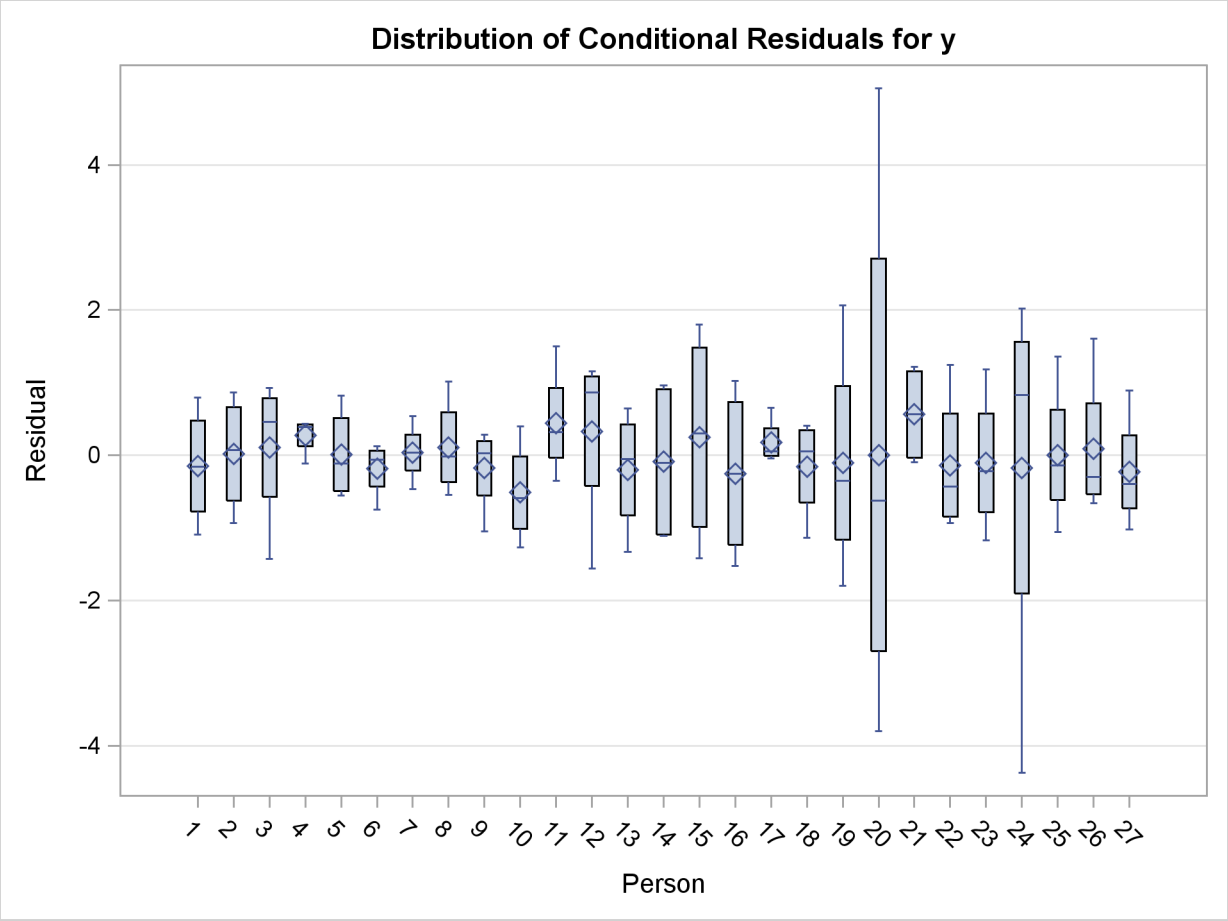


Output 56.8.7 Distribution of Marginal Residuals



The conditional residuals incorporate the EBLUPs for each child and enable you to examine whether the subject-specific model is adequate (Output 56.8.8). By using each child "as its own control," the residuals are now centered near zero. Subjects 20 and 24 stand out as unusual in all three sets of box plots.

Output 56.8.8 Distribution of Conditional Residuals



Example 56.9: Examining Individual Test Components

The LCOMPONENTS option in the MODEL statement enables you to perform single-degree-of-freedom tests for individual rows of the **L** matrix. Such tests are useful to identify interaction patterns. In a balanced layout, Type 3 components of **L** associated with A*B interactions correspond to simple contrasts of cell mean differences.

The first example revisits the data from the split-plot design by Stroup (1989a) that was analyzed in Example 56.1. Recall that variables A and B in the following statements represent the whole-plot and subplot factors, respectively:

```
proc mixed data=sp;
   class a b block;
   model y = a b a*b / LComponents e3;
   random block a*block;
run;
```

The MIXED procedure constructs a separate **L** matrix for each of the three fixed-effects components. The matrices are displayed in Output 56.9.1. The tests for fixed effects are shown in

Output 56.9.2.

Output 56.9.1 Coefficients of Type 3 Estimable Functions

The Mixed Procedure

Type 3 Coefficients for A

Effect	A	B	Row1	Row2
Intercept				
A	1		1	
A	2			1
A	3		-1	-1
B		1		
B		2		
A*B	1	1	0.5	
A*B	1	2	0.5	
A*B	2	1		0.5
A*B	2	2		0.5
A*B	3	1	-0.5	-0.5
A*B	3	2	-0.5	-0.5

Type 3 Coefficients for B

Effect	A	B	Row1
Intercept			
A	1		
A	2		
A	3		
B		1	1
B		2	-1
A*B	1	1	0.3333
A*B	1	2	-0.333
A*B	2	1	0.3333
A*B	2	2	-0.333
A*B	3	1	0.3333
A*B	3	2	-0.333

Type 3 Coefficients for A*B

Effect	A	B	Row1	Row2
Intercept				
A	1			
A	2			
A	3			
B		1		
B		2		
A*B	1	1	1	
A*B	1	2	-1	
A*B	2	1		1
A*B	2	2		-1
A*B	3	1	-1	-1
A*B	3	2	1	1

Output 56.9.2 Type 3 Tests in Split-Plot Example

Type 3 Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
A	2	6	4.07	0.0764
B	1	9	19.39	0.0017
A*B	2	9	4.02	0.0566

If $\mu_{i.}$ denotes a whole-plot main effect mean, $\mu_{.j}$ denotes a subplot main effect mean, and μ_{ij} denotes a cell mean, the five components shown in Output 56.9.3 correspond to tests of the following:

- $H_0 : \mu_{1.} = \mu_{2.}$
- $H_0 : \mu_{2.} = \mu_{3.}$
- $H_0 : \mu_{.1} = \mu_{.2}$
- $H_0 : \mu_{11} - \mu_{12} = \mu_{31} - \mu_{32}$
- $H_0 : \mu_{21} - \mu_{22} = \mu_{31} - \mu_{32}$

Output 56.9.3 Type 3 L Components Table

L Components of Type 3 Tests of Fixed Effects

Effect	L Index	Estimate	Standard Error	DF	t Value	Pr > |t|
A	1	7.1250	3.1672	6	2.25	0.0655
A	2	8.3750	3.1672	6	2.64	0.0383
B	1	5.5000	1.2491	9	4.40	0.0017
A*B	1	7.7500	3.0596	9	2.53	0.0321
A*B	2	7.2500	3.0596	9	2.37	0.0419

The first three components are comparisons of marginal means. The fourth component compares the effect of factor B at the first whole-plot level against the effect of B at the third whole-plot level. Finally, the last component tests whether the factor B effect changes between the second and third whole-plot level.

The Type 3 component tests can also be produced with these corresponding ESTIMATE statements:

```
proc mixed data=sp;
   class a b block ;
   model y = a b a*b;
   random block a*block;
   estimate 'a    1' a 1 0 -1;
   estimate 'a    2' a 0 1 -1;
```

```
   estimate 'b    1' b   1 -1;
   estimate 'a*b  1' a*b 1 -1 0  0 -1 1;
   estimate 'a*b  2' a*b 0  0 1 -1 -1 1;
   ods select Estimates;
run;
```

The results are shown in Output 56.9.4.

Output 56.9.4 Results from ESTIMATE Statements

The Mixed Procedure

Estimates

Label		Estimate	Standard Error	DF	t Value	Pr > |t|
a	1	7.1250	3.1672	6	2.25	0.0655
a	2	8.3750	3.1672	6	2.64	0.0383
b	1	5.5000	1.2491	9	4.40	0.0017
a*b	1	7.7500	3.0596	9	2.53	0.0321
a*b	2	7.2500	3.0596	9	2.37	0.0419

A second useful application of the LCOMPONENTS option is in polynomial models, where Type 1 tests are often used to test the entry of model terms sequentially. The SOLUTION option in the MODEL statement displays the regression coefficients that correspond to a Type 3 analysis. That is, the coefficients represent the partial coefficients you would get by adding the regressor variable last in a model containing all other effects, and the tests are identical to those in the "Type 3 Tests of Fixed Effects" table.

Consider the following DATA step and the fit of a third-order polynomial regression model.

```
data polynomial;
  do x=1 to 20; input y@@; output; end;
  datalines;
1.092   1.758   1.997   3.154   3.880
3.810   4.921   4.573   6.029   6.032
6.291   7.151   7.154   6.469   7.137
6.374   5.860   4.866   4.155   2.711
;

proc mixed data=polynomial;
  model y = x x*x x*x*x / s lcomponents htype=1,3;
run;
```

The *t* tests displayed in the "Solution for Fixed Effects" table are Type 3 tests, sometimes referred to as partial tests. They measure the contribution of a regressor in the presence of all other regressor variables in the model.

Output 56.9.5 Parameter Estimates in Polynomial Model

The Mixed Procedure

Solution for Fixed Effects

Effect	Estimate	Standard Error	DF	t Value	Pr > |t|
Intercept	0.7837	0.3545	16	2.21	0.0420
x	0.3726	0.1426	16	2.61	0.0189
x*x	0.04756	0.01558	16	3.05	0.0076
x*x*x	−0.00306	0.000489	16	−6.27	<.0001

The Type 3 L components are identical to the tests in the "Solutions for Fixed Effects" table shown in Output 56.9.5. The Type 1 table yields the following:

- sequential (Type 1) tests of regression variables that test the significance of a regressor given all other variables preceding it in the model list
- the regression coefficients for sequential submodels

Output 56.9.6 Type 1 and Type 3 L Components

L Components of Type 1 Tests of Fixed Effects

Effect	L Index	Estimate	Standard Error	DF	t Value	Pr > |t|
x	1	0.1763	0.01259	16	14.01	<.0001
x*x	1	−0.04886	0.002449	16	−19.95	<.0001
x*x*x	1	−0.00306	0.000489	16	−6.27	<.0001

L Components of Type 3 Tests of Fixed Effects

Effect	L Index	Estimate	Standard Error	DF	t Value	Pr > |t|
x	1	0.3726	0.1426	16	2.61	0.0189
x*x	1	0.04756	0.01558	16	3.05	0.0076
x*x*x	1	−0.00306	0.000489	16	−6.27	<.0001

The estimate of 0.1763 is the regression coefficient in a simple linear regression of Y on X. The estimate of −0.04886 is the partial coefficient for the quadratic term when it is added to a model containing only a linear component. Similarly, the value −0.00306 is the partial coefficient for the cubic term when it is added to a model containing a linear and quadratic component. The last Type 1 component is always identical to the corresponding Type 3 component.

References

Akritas, M. G., Arnold, S. F., and Brunner, E. (1997), "Nonparametric Hypotheses and Rank Statistics for Unbalanced Factorial Designs," *Journal of the American Statistical Association,* 92: 258–265.

Akaike, H. (1974), "A New Look at the Statistical Model Identification," *IEEE Transaction on Automatic Control*, AC–19, 716–723.

Allen, D. M. (1974), "The Relationship between Variable Selection and Data Augmentation and a Method of Prediction," *Technometrics*, 16, 125–127.

Bates, D. M. and Watts, D. G. (1988), *Nonlinear Regression Analysis and Its Applications*, New York: John Wiley & Sons.

Beckman, R. J., Nachtsheim, C. J., and Cook, D. R. (1987), "Diagnostics for Mixed-Model Analysis of Variance," *Technometrics*, 29, 413–426

Belsley, D. A., Kuh, E., and Welsch, R. E. (1980), *Regression Diagnostics; Identifying Influential Data and Sources of Collinearity*, New York: John Wiley & Sons.

Box, G. E. P. and Tiao, G. C. (1973), *Bayesian Inference in Statistical Analysis*, Wiley Classics Library Edition Published 1992, New York: John Wiley & Sons.

Bozdogan, H. (1987), "Model Selection and Akaike's Information Criterion (AIC): The General Theory and Its Analytical Extensions," *Psychometrika*, 52, 345–370.

Brown, H. and Prescott, R. (1999), *Applied Mixed Models in Medicine*, New York: John Wiley & Sons.

Brownie, C., Bowman, D. T., and Burton, J. W. (1993), "Estimating Spatial Variation in Analysis of Data from Yield Trials: A Comparison of Methods," *Agronomy Journal*, 85, 1244–1253.

Brownie, C., and Gumpertz, M. L. (1997), "Validity of Spatial Analysis of Large Field Trials," *Journal of Agricultural, Biological, and Environmental Statistics,* 2, 1–23.

Brunner, E., Dette, H., Munk, A. (1997), "Box-Type Approximations in Nonparametric Factorial Designs," *Journal of the American Statistical Association,* 92, 1494–1502.

Brunner, E., Domhof, S., and Langer, F. (2002), *Nonparametric Analysis of Longitudinal Data in Factorial Experiments*, New York: John Wiley & Sons.

Burdick, R. K. and Graybill, F. A. (1992), *Confidence Intervals on Variance Components,* New York: Marcel Dekker.

Burnham, K. P. and Anderson, D. R. (1998), *Model Selection and Inference: A Practical Information-Theoretic Approach,* New York: Springer-Verlag.

Carlin, B. P. and Louis, T. A. (1996), *Bayes and Empirical Bayes Methods for Data Analysis,* London: Chapman and Hall.

Carroll, R. J. and Ruppert, D. (1988), *Transformation and Weighting in Regression*, London: Chapman and Hall.

Chilès, J. P. and Delfiner, P. (1999), *Geostatistics. Modeling Spatial Uncertainty*, New York: John Wiley & Sons.

Christensen, R., Pearson, L. M., and Johnson, W. (1992), "Case-Deletion Diagnostics for Mixed Models," *Technometrics*, 34, 38–45.

Cook, R. D. (1977), "Detection of Influential Observations in Linear Regression," *Technometrics*, 19, 15–18.

Cook, R. D. (1979), "Influential Observations in Linear Regression," *Journal of the American Statistical Association*, 74, 169–174.

Cook, R. D. and Weisberg, S. (1982), *Residuals and Influence in Regression*, New York: Chapman and Hall.

Cressie, N. (1993), *Statistics for Spatial Data*, Revised Edition, New York: John Wiley & Sons.

Crowder, M. J. and Hand, D. J. (1990), *Analysis of Repeated Measures*, New York: Chapman and Hall.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977), "Maximum Likelihood from Incomplete Data via the EM Algorithm," *Journal of the Royal Statistical Society*, Ser. B., 39, 1–38.

Diggle, P. J. (1988), "An Approach to the Analysis of Repeated Measurements," *Biometrics*, 44, 959–971.

Diggle, P. J., Liang, K. Y., and Zeger, S. L. (1994), *Analysis of Longitudinal Data*, Oxford: Clarendon Press.

Dunnett, C. W. (1980), "Pairwise Multiple Comparisons in the Unequal Variance Case," *Journal of the American Statistical Association*, 75, 796–800.

Edwards, D. and Berry, J. J. (1987), "The Efficiency of Simulation-based Multiple Comparisons," *Biometrics*, 43, 913–928.

Everitt, B. S. (1995), "The Analysis of Repeated Measures: A Practical Review with Examples," *The Statistician,* 44, 113–135.

Fai, A. H. T. and Cornelius, P. L. (1996), "Approximate F-tests of Multiple Degree of Freedom Hypotheses in Generalized Least Squares Analyses of Unbalanced Split-plot Experiments," *Journal of Statistical Computation and Simulation,* 54, 363–378.

Federer, W. T. and Wolfinger, R. D. (1998), "SAS Code for Recovering Intereffect Information in Experiments with Incomplete Block and Lattice Rectangle Designs," *Agronomy Journal,* 90, 545–551.

Fuller, W. A. (1976), *Introduction to Statistical Time Series,* New York: John Wiley & Sons.

Fuller, W. A. and Battese, G. E. (1973), "Transformations for Estimation of Linear Models with Nested Error Structure," *Journal of the American Statistical Association*, 68, 626–632.

Galecki, A. T. (1994), "General Class of Covariance Structures for Two or More Repeated Factors in Longitudinal Data Analysis," *Communications in Statistics–Theory and Methods*, 23(11), 3105–3119.

Games, P. A., and Howell, J. F. (1976), "Pairwise Multiple Comparison Procedures With Unequal *n*'s and/or Variances: A Monte Carlo Study," *Journal of Educational Statistics*, 1, 113–125.

Gelfand, A. E., Hills, S. E., Racine-Poon, A., and Smith, A. F. M. (1990), "Illustration of Bayesian Inference in Normal Data Models Using Gibbs Sampling," *Journal of the American Statistical Association*, 85, 972–985.

Ghosh, M. (1992), Discussion of Schervish, M., "Bayesian Analysis of Linear Models," *Bayesian Statistics 4*, eds. J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, Oxford: University Press, 432–433.

Giesbrecht, F. G. (1989), "A General Structure for the Class of Mixed Linear Models," *Applications of Mixed Models in Agriculture and Related Disciplines*, Southern Cooperative Series Bulletin No. 343, Louisiana Agricultural Experiment Station, Baton Rouge, 183–201.

Giesbrecht, F. G. and Burns, J. C. (1985), "Two-Stage Analysis Based on a Mixed Model: Large-sample Asymptotic Theory and Small-Sample Simulation Results," *Biometrics*, 41, 477–486.

Golub, G. H. and Van Loan, C. F. (1989), *Matrix Computations*, Second Edition, Baltimore: Johns Hopkins University Press.

Goodnight, J. H. (1978), SAS Technical Report R-101, *Tests of Hypotheses in Fixed-Effects Linear Models*, Cary, NC: SAS Institute Inc.

Goodnight, J. H. (1979), "A Tutorial on the Sweep Operator," *American Statistician*, 33, 149–158.

Goodnight, J. H. and Hemmerle, W. J. (1979), "A Simplified Algorithm for the W-Transformation in Variance Component Estimation," *Technometrics*, 21, 265–268.

Gotway, C. A. and Stroup, W. W. (1997), "A Generalized Linear Model Approach to Spatial Data and Prediction," *Journal of Agricultural, Biological, and Environmental Statistics,* 2, 157–187.

Greenhouse, S. W. and Geisser, S. (1959), "On Methods in the Analysis of Profile Data," *Psychometrika,* 32, 95–112.

Gregoire, T. G., Schabenberger, O., and Barrett, J. P. (1995), "Linear Modelling of Irregularly Spaced, Unbalanced, Longitudinal Data from Permanent Plot Measurements," *Canadian Journal of Forest Research*, 25, 137–156.

Handcock, M. S. and Stein, M. L. (1993), "A Bayesian Analysis of Kriging," *Technometrics*, 35(4), 403–410

Handcock, M. S. and Wallis, J. R. (1994), "An Approach to Statistical Spatial-Temporal Modeling of Meteorological Fields (with Discussion)," *Journal of the American Statistical Association*, 89, 368–390.

Hanks, R.J., Sisson, D.V., Hurst, R.L, and Hubbard K.G. (1980), "Statistical Analysis of Results from Irrigation Experiments Using the Line-Source Sprinkler System," *Soil Science Society American Journal*, 44, 886–888.

Hannan, E.J. and Quinn, B.G. (1979), "The Determination of the Order of an Autoregression," *Journal of the Royal Statistical Society*, Series B, 41, 190–195.

Hartley, H. O. and Rao, J. N. K. (1967), "Maximum-Likelihood Estimation for the Mixed Analysis of Variance Model," *Biometrika*, 54, 93–108.

Harville, D. A. (1977), "Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems," *Journal of the American Statistical Association*, 72, 320–338.

Harville, D. A. (1988), "Mixed-Model Methodology: Theoretical Justifications and Future Directions," *Proceedings of the Statistical Computing Section*, American Statistical Association, New Orleans, 41–49.

Harville, D. A. (1990), "BLUP (Best Linear Unbiased Prediction), and Beyond," in *Advances in Statistical Methods for Genetic Improvement of Livestock*, Springer-Verlag, 239–276.

Harville, D. A. and Jeske, D. R. (1992), "Mean Squared Error of Estimation or Prediction under a General Linear Model," *Journal of the American Statistical Association*, 87, 724–731.

Hemmerle, W. J. and Hartley, H. O. (1973), "Computing Maximum Likelihood Estimates for the Mixed AOV Model Using the W-Transformation," *Technometrics*, 15, 819–831.

Henderson, C. R. (1984), *Applications of Linear Models in Animal Breeding*, University of Guelph.

Henderson, C. R. (1990), "Statistical Method in Animal Improvement: Historical Overview," in *Advances in Statistical Methods for Genetic Improvement of Livestock*, New York: Springer-Verlag, 1–14.

Hsu, J. C. (1992), "The Factor Analytic Approach to Simultaneous Inference in the General Linear Model," *Journal of Computational and Graphical Statistics*, 1, 151–168.

Huber, P. J. (1967), "The Behavior of Maximum Likelihood Estimates under Nonstandard Conditions," *Proc. Fifth Berkeley Symp. Math. Statist. Prob.*, 1, 221–233.

Hurtado, G. I. H. (1993), *Detection of Influential Observations in Linear Mixed Models*, Ph.D. dissertation, Department of Statistics, North Carolina State University, Raleigh, NC.

Hurvich, C. M. and Tsai, C.-L. (1989), "Regression and Time Series Model Selection in Small Samples," *Biometrika*, 76, 297–307.

Huynh, H. and Feldt, L. S. (1970), "Conditions Under Which Mean Square Ratios in Repeated Measurements Designs Have Exact F-Distributions," *Journal of the American Statistical Association*, 65, 1582–1589.

Jennrich, R. I. and Schluchter, M. D. (1986), "Unbalanced Repeated-Measures Models with Structured Covariance Matrices," *Biometrics*, 42, 805–820.

Johnson, D. E., Chaudhuri, U. N., and Kanemasu, E. T. (1983), "Statistical Analysis of Line-Source Sprinkler Irrigation Experiments and Other Nonrandomized Experiments Using Multivariate Methods," *Soil Science Society American Journal*, 47, 309–312.

Jones, R. H. and Boadi-Boateng, F. (1991), "Unequally Spaced Longitudinal Data with AR(1) Serial Correlation," *Biometrics*, 47, 161–175.

Kackar, R. N. and Harville, D. A. (1984), "Approximations for Standard Errors of Estimators of Fixed and Random Effects in Mixed Linear Models," *Journal of the American Statistical Association*, 79, 853–862.

Kass, R. E. and Steffey, D. (1989), "Approximate Bayesian Inference in Conditionally Independent Hierarchical Models (Parametric Empirical Bayes Models)," *Journal of the American Statistical Association*, 84, 717–726.

Kenward, M. G. (1987), "A Method for Comparing Profiles of Repeated Measurements," *Applied Statistics*, 36, 296–308.

Kenward, M. G. and Roger, J. H. (1997), "Small Sample Inference for Fixed Effects from Restricted Maximum Likelihood," *Biometrics*, 53, 983–997.

Keselman, H. J., Algina, J., Kowalchuk, R. K., and Wolfinger, R. D. (1998), "A Comparison of Two Approaches for Selecting Covariance Structures in the Analysis of Repeated Measures," *Communications in Statistics–Computation and Simulation,* 27(3), 591–604.

Keselman, H. J., Algina, J., Kowalchuk, R. K., and Wolfinger, R. D. (1999). "A Comparison of Recent Approaches to the Analysis of Repeated Measurements," *British Journal of Mathematical and Statistical Psychology,* 52, 63–78.

Kramer, C. Y. (1956), "Extension of Multiple Range Tests to Group Means with Unequal Numbers of Replications," *Biometrics*, 12, 309–310.

Laird, N. M. and Ware, J. H. (1982), "Random-Effects Models for Longitudinal Data," *Biometrics*, 38, 963–974.

Laird, N. M., Lange, N., and Stram, D. (1987), "Maximum Likelihood Computations with Repeated Measures: Application of the EM Algorithm," *Journal of the American Statistical Association*, 82, 97–105.

LaMotte, L. R. (1973), "Quadratic Estimation of Variance Components," *Biometrics*, 29, 311–330.

Liang, K.Y. and Zeger, S.L. (1986), "Longitudinal Data Analysis Using Generalized Linear Models," *Biometrika*, 73, 13–22.

Lindsey, J. K. (1993), *Models for Repeated Measurements*, Oxford: Clarendon Press.

Lindstrom, M. J. and Bates, D. M. (1988), "Newton-Raphson and EM Algorithms for Linear Mixed-Effects Models for Repeated-Measures Data," *Journal of the American Statistical Association*, 83, 1014–1022.

Littell, R. C., Milliken, G. A., Stroup, W. W., Wolfinger, R. D., and Schabenberger, O. (2006), *SAS for Mixed Models,* Second Edition, Cary, NC: SAS Institute Inc.

Little, R. J. A. (1995), "Modeling the Drop-Out Mechanism in Repeated-Measures Studies," *Journal of the American Statistical Association*, 90, 1112–1121.

Louis, T. A. (1988), "General Methods for Analyzing Repeated Measures," *Statistics in Medicine*, 7, 29–45.

Macchiavelli, R. E. and Arnold, S. F. (1994), "Variable Order Ante-dependence Models," *Communications in Statistics–Theory and Methods*, 23(9), 2683–2699.

Marx, D. and Thompson, K. (1987), "Practical Aspects of Agricultural Kriging," Bulletin 903, Arkansas Agricultural Experiment Station, Fayetteville.

Matérn, B. (1986), *Spatial Variation*, Second Edition, Lecture Notes in Statistics, New York: Springer-Verlag.

McKeon, J. J. (1974), "F Approximations to the Distribution of Hotelling's T_0^2," *Biometrika*, 61, 381–383.

McLean, R. A. and Sanders, W. L. (1988), "Approximating Degrees of Freedom for Standard Errors in Mixed Linear Models," *Proceedings of the Statistical Computing Section*, American Statistical Association, New Orleans, 50–59.

McLean, R. A., Sanders, W. L., and Stroup, W. W. (1991), "A Unified Approach to Mixed Linear Models," *The American Statistician*, 45, 54–64.

Milliken, G. A. and Johnson, D. E. (1992), *Analysis of Messy Data, Volume 1: Designed Experiments*, New York: Chapman and Hall.

Murray, D. M. (1998), *Design and Analysis of Group-Randomized Trials,* New York: Oxford University Press.

Myers, R. H. (1990), *Classical and Modern Regression with Applications*, Second Edition, Belmont, CA: PWS-Kent.

Obenchain, R. L. (1990), *STABLSIM.EXE*, Version 9010, Eli Lilly and Company, Indianapolis, Indiana, unpublished C code.

Patel, H. I. (1991), "Analysis of Incomplete Data from a Clinical Trial with Repeated Measurements," *Biometrika*, 78, 609–619.

Patterson, H. D. and Thompson, R. (1971), "Recovery of Inter-block Information When Block Sizes Are Unequal," *Biometrika*, 58, 545–554.

Pillai, K. C. and Samson, P. (1959), "On Hotelling's Generalization of T^2," *Biometrika*, 46, 160–168.

Pothoff, R. F. and Roy, S. N. (1964), "A Generalized Multivariate Analysis of Variance Model Useful Especially for Growth Curve Problems," *Biometrika*, 51, 313–326.

Prasad, N. G. N. and Rao, J. N. K. (1990), "The Estimation of Mean Squared Error of Small-Area Estimators," *Journal of the American Statistical Association*, 85, 163–171.

Pringle, R. M. and Rayner, A. A. (1971), *Generalized Inverse Matrices with Applications to Statistics*, New York: Hafner Publishing Co.

Rao, C. R. (1972), "Estimation of Variance and Covariance Components in Linear Models," *Journal of the American Statistical Association*, 67, 112–115.

Ripley, B. D. (1987), *Stochastic Simulation*, New York: John Wiley & Sons.

Robinson, G. K. (1991), "That BLUP Is a Good Thing: The Estimation of Random Effects," *Statistical Science*, 6, 15–51.

Rubin, D. B. (1976), "Inference and Missing Data," *Biometrika*, 63, 581–592.

Sacks, J., Welch, W. J., Mitchell, T. J. and Wynn, H. P. (1989), "Design and Analysis of Computer Experiments," *Statistical Science* 4, 409–435.

Schabenberger, O. and Gotway, C. A. (2005), *Statistical Methods for Spatial Data Analysis,* Boca Raton, FL: CRC Press.

Schluchter, M. D. and Elashoff, J. D. (1990), "Small-Sample Adjustments to Tests with Unbalanced Repeated Measures Assuming Several Covariance Structures," *Journal of Statistical Computation and Simulation*, 37, 69–87.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Schervish, M. J. (1992), "Bayesian Analysis of Linear Models," *Bayesian Statistics 4*, eds. J.M. Bernardo, J.O. Berger, A.P. Dawid, and A.F.M. Smith, Oxford: University Press, 419–434 (with discussion).

Searle, S. R. (1971), *Linear Models*, New York: John Wiley & Sons.

Searle, S. R. (1982), *Matrix Algebra Useful for Statisticians*, New York: John Wiley & Sons.

Searle, S. R. (1988), "Mixed Models and Unbalanced Data: Wherefrom, Whereat, and Whereto?" *Communications in Statistics–Theory and Methods*, 17(4), 935–968.

Searle, S. R., Casella, G., and McCulloch, C. E. (1992), *Variance Components*, New York: John Wiley & Sons.

Self, S. G. and Liang, K. Y. (1987), "Asymptotic Properties of Maximum Likelihood Estimators and Likelihood Ratio Tests under Nonstandard Conditions," *Journal of the American Statistical Association*, 82, 605–610.

Serfling, R. J. (1980), *Approximation Theorems of Mathematical Statistics*, New York: John Wiley & Sons.

Singer, J. D. (1998), "Using SAS PROC MIXED to Fit Multilevel Models, Hierarchical Models, and Individual Growth Models," *Journal of Educational and Behavioral Statistics*, 23(4), 323–355.

Smith, A. F. M. and Gelfand, A. E. (1992), "Bayesian Statistics without Tears: A Sampling-Resampling Perspective," *American Statistician*, 46, 84–88.

Snedecor, G. W. and Cochran, W. G. (1976), *Statistical Methods*, Sixth Edition, Ames: Iowa State University Press.

Snedecor, G. W. and Cochran, W. G. (1980), *Statistical Methods*, Ames: Iowa State University Press.

Steel, R. G. D., Torrie, J. H., and Dickey D. (1997), *Principles and Procedures of Statistics: A Biometrical Approach*, Third Edition, New York: McGraw-Hill, Inc.

Stram, D. O. and Lee, J. W. (1994), "Variance Components Testing in the Longitudinal Mixed Effects Model," *Biometrics*, 50, 1171–1177.

Stroup, W. W. (1989a), "Predictable Functions and Prediction Space in the Mixed Model Procedure," in *Applications of Mixed Models in Agriculture and Related Disciplines*, Southern Cooperative Series Bulletin No. 343, Louisiana Agricultural Experiment Station, Baton Rouge, 39–48.

Stroup, W. W. (1989b), "Use of Mixed Model Procedure to Analyze Spatially Correlated Data: An Example Applied to a Line-Source Sprinkler Irrigation Experiment," *Applications of Mixed Models in Agriculture and Related Disciplines*, Southern Cooperative Series Bulletin No. 343, Louisiana Agricultural Experiment Station, Baton Rouge, 104–122.

Stroup, W. W., Baenziger, P. S., and Mulitze, D. K. (1994), "Removing Spatial Variation from Wheat Yield Trials: A Comparison of Methods," *Crop Science*, 86, 62–66.

Sullivan, L. M., Dukes, K. A., and Losina, E. (1999), "An Introduction to Hierarchical Linear Modelling," *Statistics in Medicine*, 18, 855–888.

Swallow, W. H. and Monahan, J. F. (1984), "Monte Carlo Comparison of ANOVA, MIVQUE, REML, and ML Estimators of Variance Components," *Technometrics*, 28, 47–57.

Tamhane, A. C. (1979), "A Comparison of Procedures for Multiple Comparisons of Means With Unequal Variances," *Journal of the American Statistical Association*, 74, 471–480.

Tierney, L. (1994), "Markov Chains for Exploring Posterior Distributions" (with discussion), *Annals of Statistics*, 22, 1701–1762.

Verbeke, G. and Molenberghs, G., eds. (1997), *Linear Mixed Models in Practice: A SAS-Oriented Approach,* New York: Springer.

Verbeke, G. and Molenberghs, G. (2000), *Linear Mixed Models for Longitudinal Data,* New York: Springer.

Westfall, P. J. and Young, S. S. (1993), *Resampling-based Multiple Testing*, New York: John Wiley & Sons.

Westfall, P. H., Tobias, R. D., Rom, D., Wolfinger, R. D., and Hochberg, Y. (1999), *Multiple Comparisons and Multiple Tests Using the SAS System*, Cary, NC: SAS Institute Inc.

White, H. (1980), "A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity," *Econometrica*, 48, 817–838.

Whittle, P. (1954), "On Stationary Processes in the Plane," *Biometrika*, 41, 434–449.

Winer, B. J. (1971), *Statistical Principles in Experimental Design*, Second Edition, New York: McGraw-Hill, Inc.

Wolfinger, R. D. (1993), "Covariance Structure Selection in General Mixed Models," *Communications in Statistics, Simulation and Computation*, 22(4), 1079–1106.

Wolfinger, R. D. (1996), "Heterogeneous Variance-Covariance Structures for Repeated Measures," *Journal of Agricultural, Biological, and Environmental Statistics,* 1, 205-230.

Wolfinger, R. D. (1997), "An Example of Using Mixed Models and PROC MIXED for Longitudinal Data," *Journal of Biopharmaceutical Statistics*, 7(4), 481–500.

Wolfinger, R. D. and Chang, M. (1995), "Comparing the SAS GLM and MIXED Procedures for Repeated Measures," *Proceedings of the Twentieth Annual SAS Users Group Conference*.

Wolfinger, R. D., Tobias, R. D., and Sall, J. (1991), "Mixed Models: A Future Direction," *Proceedings of the Sixteenth Annual SAS Users Group Conference*, 1380–1388.

Wolfinger, R. D., Tobias, R. D., and Sall, J. (1994), "Computing Gaussian Likelihoods and Their Derivatives for General Linear Mixed Models," *SIAM Journal on Scientific Computing*, 15(6), 1294–1310.

Wright, P. S. (1994), "Adjusted F Tests for Repeated Measures with the MIXED Procedure," 328 SMC-Statistics Department, University of Tennessee.

Zimmerman, D. L. and Harville, D. A. (1991), "A Random Field Approach to the Analysis of Field-Plot Experiments and Other Spatial Experiments," *Biometrics*, 47, 223–239.

Chapter 57

The MODECLUS Procedure

Contents

Overview: MODECLUS Procedure **4093**
Getting Started: MODECLUS Procedure **4095**
Syntax: MODECLUS Procedure **4101**
PROC MODECLUS Statement 4101
BY Statement 4108
FREQ Statement 4109
ID Statement 4109
VAR Statement 4109
Details: MODECLUS Procedure **4109**
Density Estimation 4109
Clustering Methods 4113
Significance Tests 4115
Computational Resources 4121
Missing Values 4121
Output Data Sets 4121
Displayed Output 4124
ODS Table Names 4127
Examples: MODECLUS Procedure **4128**
Example 57.1: Cluster Analysis of Samples from Univariate Distributions . 4128
Example 57.2: Cluster Analysis of Flying Mileages between Ten American Cities 4151
Example 57.3: Cluster Analysis with Significance Tests 4162
Example 57.4: Cluster Analysis: Hertzsprung-Russell Plot 4170
Example 57.5: Using the TRACE Option When METHOD=6 4174
References **4178**

Overview: MODECLUS Procedure

The MODECLUS procedure clusters observations in a SAS data set by using any of several algorithms based on nonparametric density estimates. The data can be numeric coordinates or distances. PROC MODECLUS can perform approximate significance tests for the number of clusters and can

hierarchically join nonsignificant clusters. The significance tests are empirically validated by simulations with sample sizes ranging from 20 to 2000.

PROC MODECLUS produces output data sets containing density estimates and cluster membership, various cluster statistics including approximate p-values, and a summary of the number of clusters generated by various algorithms, smoothing parameters, and significance levels.

Most clustering methods are biased toward finding clusters possessing certain characteristics related to size (number of members), shape, or dispersion. Methods based on the least squares criterion (Sarle 1982), such as k-means and Ward's minimum variance method, tend to find clusters with roughly the same number of observations in each cluster. Average linkage (see Chapter 29, "The CLUSTER Procedure") is somewhat biased toward finding clusters of equal variance. Many clustering methods tend to produce compact, roughly hyperspherical clusters and are incapable of detecting clusters with highly elongated or irregular shapes. The methods with the least bias are those based on nonparametric density estimation (Silverman 1986, pp. 130–146; Scott 1992, pp. 125–190) such as density linkage (see Chapter 29, "The CLUSTER Procedure"), Wong and Lane 1983, and Wong and Schaack 1982). The biases of many commonly used clustering methods are discussed in Chapter 11, "Introduction to Clustering Procedures."

PROC MODECLUS implements several clustering methods by using nonparametric density estimation. Such clustering methods are referred to hereafter as *nonparametric clustering methods*. The methods in PROC MODECLUS are related to, but not identical to, methods developed by Gitman (1973), Huizinga (1978), Koontz and Fukunaga (1972a, 1972b), Koontz, Narendra, and Fukunaga (1976), Mizoguchi and Shimura (1980), and Wong and Lane (1983). Details of the algorithms are provided in the section "Clustering Methods" on page 4113.

For nonparametric clustering methods, a cluster is loosely defined as a region surrounding a local maximum of the probability density function (see the section "Significance Tests" on page 4115 for a more rigorous definition). Given a sufficiently large sample, nonparametric clustering methods are capable of detecting clusters of unequal size and dispersion and with highly irregular shapes. Nonparametric methods can also obtain good results for compact clusters of equal size and dispersion, but they naturally require larger sample sizes for good recovery than clustering methods that are biased toward finding such "nice" clusters.

For coordinate data, nonparametric clustering methods are less sensitive to changes in scale of the variables or to affine transformations of the variables than are most other commonly used clustering methods. Nevertheless, it is necessary to consider questions of scaling and transformation, since variables with large variances tend to have more of an effect on the resulting clusters than those with small variances. If two or more variables are not measured in comparable units, some type of standardization or scaling is necessary; otherwise, the distances used by the procedure might be based on inappropriate apples-and-oranges computations. For variables with comparable units of measurement, standardization or scaling might still be desirable if the scale estimates of the variables are not related to their expected importance for defining clusters. If you want two variables to have equal importance in the analysis, they should have roughly equal scale estimates. If you want one variable to have more of an effect than another, the former should be scaled to have a greater scale estimate than the latter. The STD option in the PROC MODECLUS statement scales all variables to equal variance. However, the variance is not necessarily the most appropriate scale estimate for cluster analysis. In particular, outliers should be removed before using PROC MODECLUS with the STD option. A variety of scale estimators including robust estimators are provided in the

STDIZE procedure (for detailed information, see Chapter 81, "The STDIZE Procedure"). Additionally, the ACECLUS procedure provides another way to transform the variables to try to improve the separation of clusters.

Since clusters are defined in terms of local maxima of the probability density function, nonlinear transformations of the data can change the number of population clusters. The variables should be transformed so that equal differences are of equal practical importance. An interval scale of measurement is required. Ordinal or ranked data are generally inappropriate, since monotone transformations can produce any arbitrary number of modes.

Unlike the methods in the CLUSTER procedure, the methods in the MODECLUS procedure are not inherently hierarchical. However, PROC MODECLUS can do approximate nonparametric significance tests for the number of clusters by obtaining an approximate p-value for each cluster, and it can hierarchically join nonsignificant clusters.

Another important difference between the MODECLUS procedure and many other clustering methods is that you do not tell PROC MODECLUS how many clusters you want. Instead, you specify a *smoothing parameter* (see the section "Density Estimation" on page 4109) and, optionally, a significance level, and PROC MODECLUS determines the number of clusters. You can specify a list of smoothing parameters, and PROC MODECLUS performs a separate cluster analysis for each value in the list.

Getting Started: MODECLUS Procedure

This section illustrates how PROC MODECLUS can be used to examine the clusters of data in the following artificial data set.

```
data example;
   input x y @@;
   datalines;

18 18  20 22  21 20  12 23  17 12  23 25  25 20  16 27
20 13  28 22  80 20  75 19  77 23  81 26  55 21  64 24
72 26  70 35  75 30  78 42  18 52  27 57  41 61  48 64
59 72  69 72  80 80  31 53  51 69  72 81
;
```

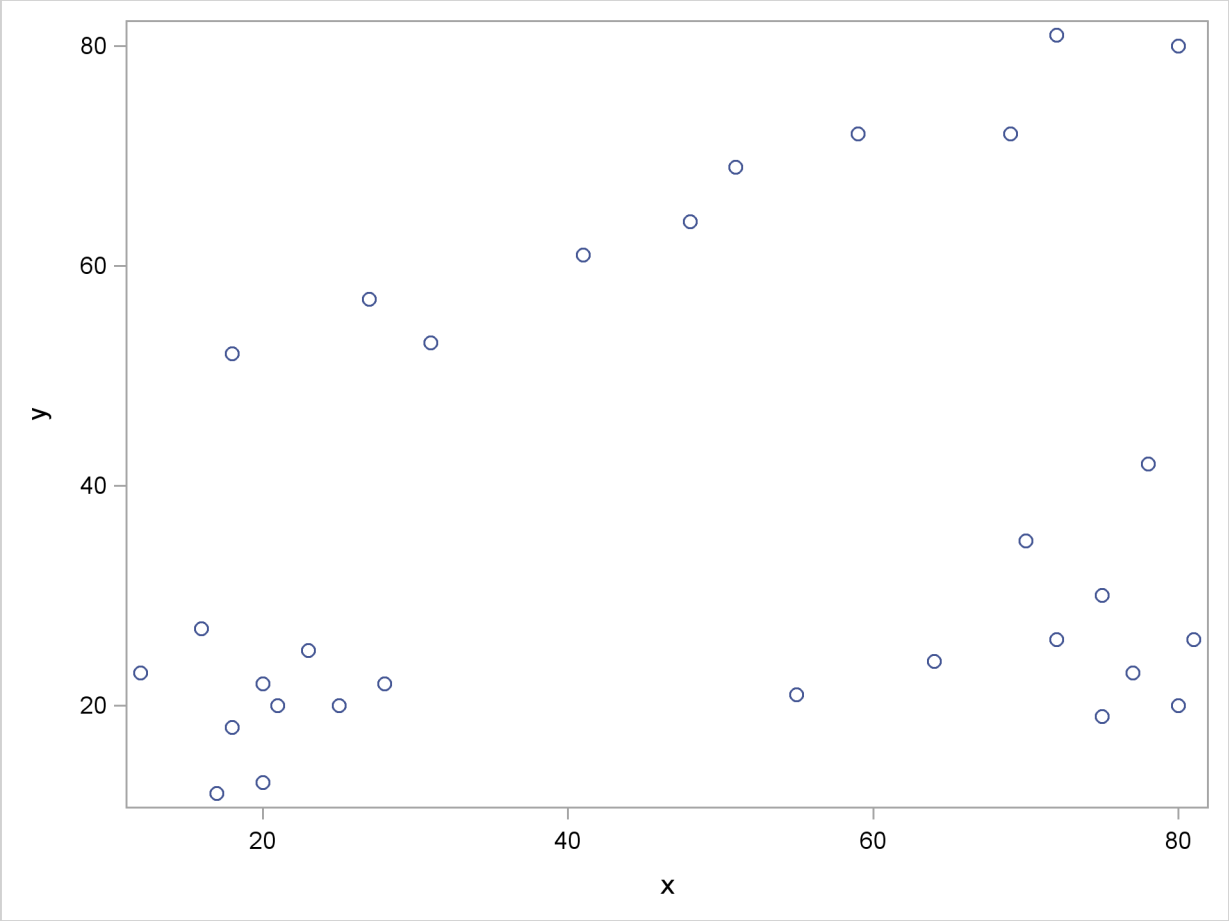
It is a good practice to plot the data to check for obvious clusters or pathologies prior to the analysis. In this example, with only two variables and a small sample size, the SGPLOT procedure in the following statements produces a scatter plot:

```
proc sgplot;
   scatter y=y x=x;
run;
```

Figure 57.1 suggests three clusters. Of these clusters, the one in the lower-left corner is the most compact, while the lower-right cluster is more dispersed.

The upper cluster is elongated and would be difficult for most clustering algorithms to identify as a single cluster. The plot also suggests that a Euclidean distance of 10 or 20 is a good initial guess for the neighborhood size in density estimation and clustering.

Figure 57.1 Scatter Plot of Data



To obtain a cluster analysis in PROC MODECLUS, you must specify the METHOD= option; for most purposes, METHOD=1 is recommended. The cluster analysis can be performed with a list of radii (R=10 15 35), as shown in the following PROC MODECLUS statement. An output data set containing the cluster membership is created with the OUT= option. The following statements produce Figure 57.2 through Figure 57.5:

```
proc modeclus data=example method=1 r=10 15 35 out=out;
run;
```

For each cluster solution, PROC MODECLUS produces a table of cluster statistics including the cluster number, the number of observations in the cluster, the maximum estimated density within the cluster, the number of observations in the cluster having a neighbor that belongs to a different cluster, and the estimated saddle density of the cluster. The results are displayed in Figure 57.2, Figure 57.3, and Figure 57.4 for three different radii. A smaller radius (R=10) yields a larger number of clusters (6), as displayed in Figure 57.2; a larger radius (R=35) includes all observations in a single cluster, as displayed in Figure 57.4. Note that all clusters in these three figures are “isolated”

since their corresponding boundary frequencies are all zeros. Consequently, all the estimated saddle densities are missing. A table summarizing each cluster solution is then produced at the end, as displayed in Figure 57.5.

Figure 57.2 Results from PROC MODECLUS for METHOD=1 and R=10

The MODECLUS Procedure
R=10 METHOD=1

Cluster Statistics

Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density
1	10	0.00106103	0	.
2	9	0.00084883	0	.
3	7	0.00031831	0	.
4	2	0.00021221	0	.
5	1	0.0001061	0	.
6	1	0.0001061	0	.

Figure 57.3 Results from PROC MODECLUS for METHOD=1 and R=15

The MODECLUS Procedure
R=15 METHOD=1

Cluster Statistics

Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density
1	10	0.00047157	0	.
2	10	0.00042441	0	.
3	10	0.00023579	0	.

Figure 57.4 Results from PROC MODECLUS for METHOD=1 and R=35

The MODECLUS Procedure
R=35 METHOD=1

Cluster Statistics

Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density
1	30	0.00012126	0	.

Figure 57.5 Summary Table

The MODECLUS Procedure

Cluster Summary

R	Number of Clusters	Frequency of Unclassified Objects
10	6	0
15	3	0
35	1	0

The OUT= data set contains a complete copy of the input data set for each cluster solution. By using a BY statement in the following PROC SGPLOT statement, you can examine the differences in cluster memberships for each radius as shown in Figure 57.6 through Figure 57.8:

```
proc sgplot data=out noautolegend;
   scatter y=y x=x / group=cluster markerchar=cluster;
by _r_;
run;
```

Figure 57.6 Scatter Plots of Cluster Memberships with _R_=10

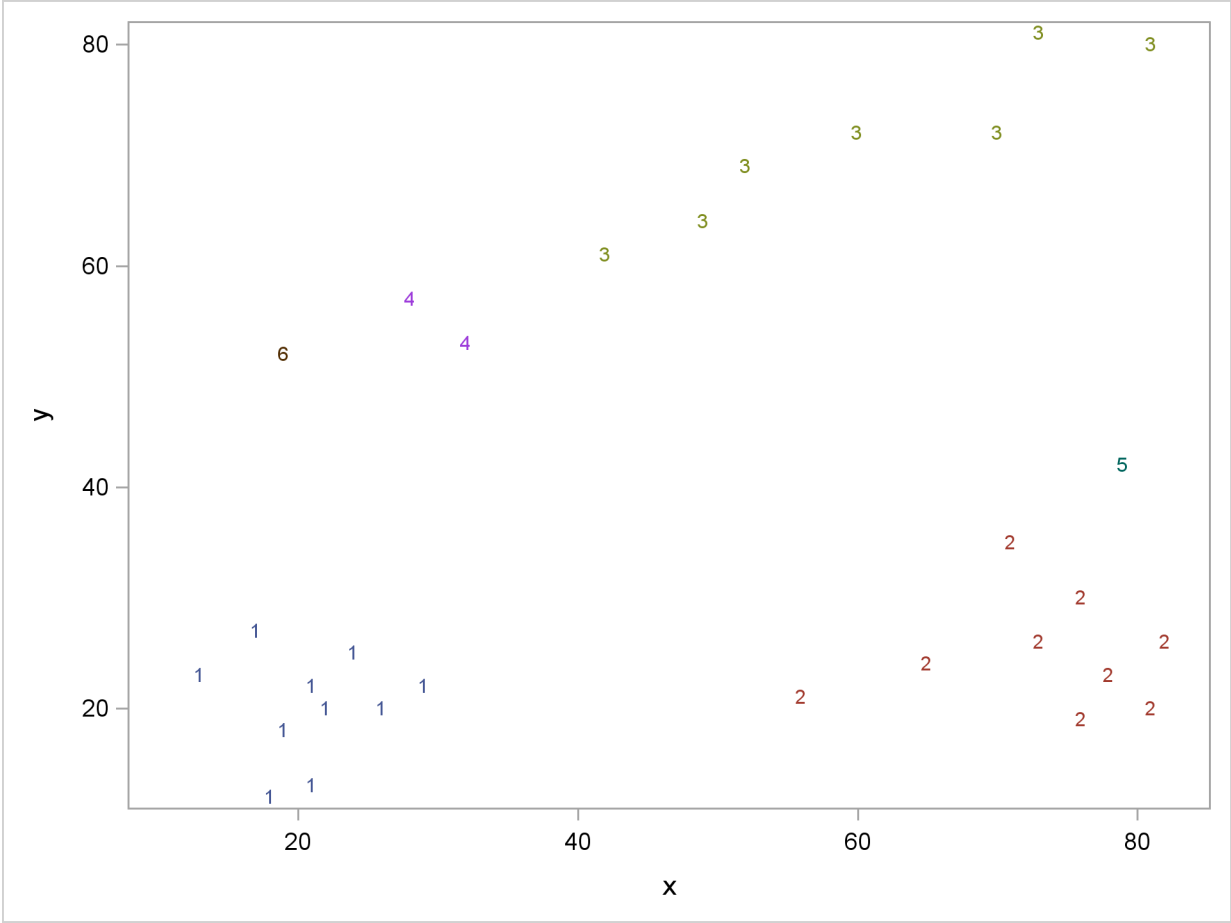


Figure 57.7 Scatter Plots of Cluster Memberships with _R_=15

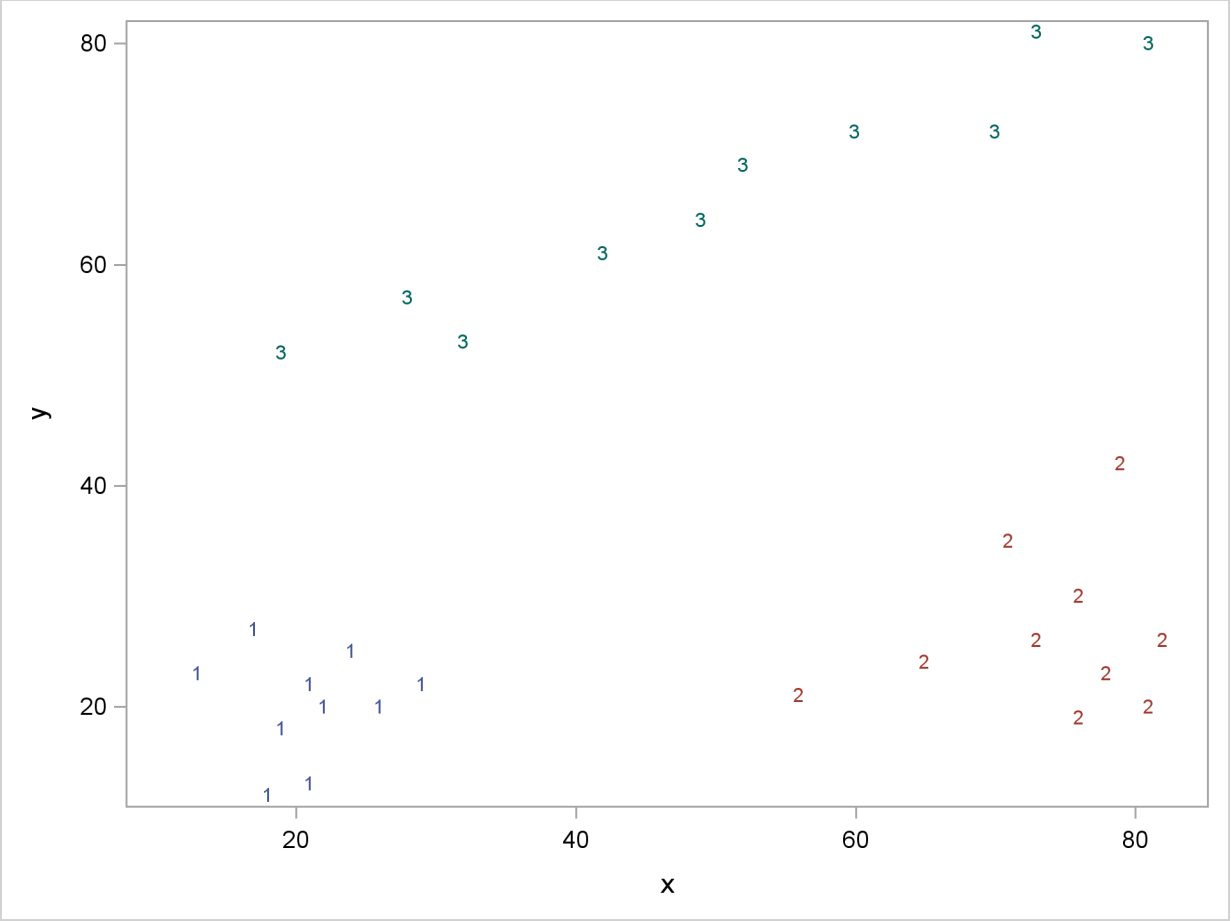
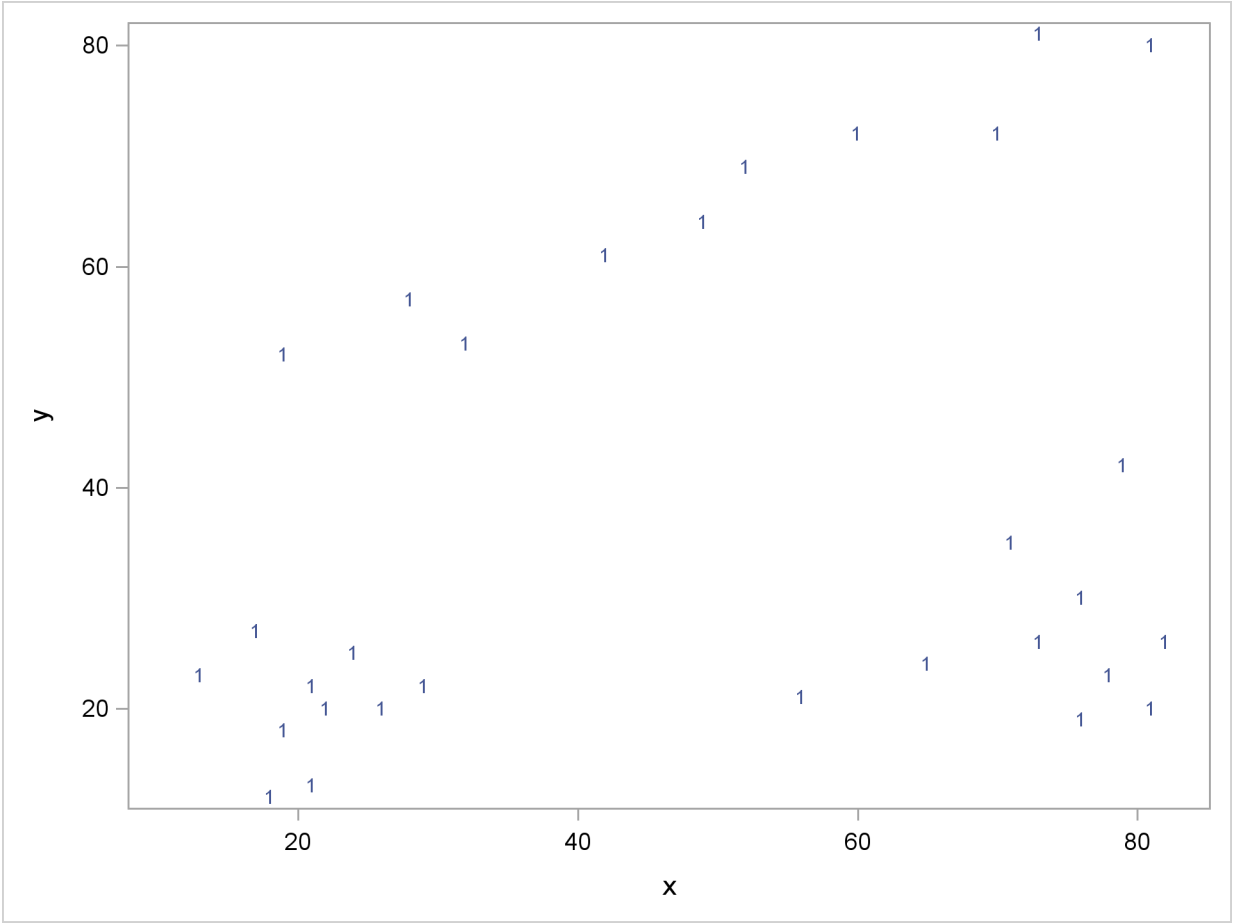


Figure 57.8 Scatter Plots of Cluster Memberships with _R_=35



Syntax: MODECLUS Procedure

The following statements invoke the MODECLUS procedure:

PROC MODECLUS < *options* > ;
BY *variables* ;
FREQ *variable* ;
ID *variable* ;
VAR *variables* ;

The PROC MODECLUS statement is required. All other statements are optional.

PROC MODECLUS Statement

PROC MODECLUS < *options* > ;

The PROC MODECLUS statement starts the MODECLUS procedure. The options available with the PROC MODECLUS statement are summarized in Table 57.1 and discussed in the following sections.

Table 57.1 Summary of PROC MODECLUS Statement Options

Option	Description
Specify input and output data sets	
DATA=	specifies input data set name
OUT=	specifies output data set name for observations
OUTC=	specifies output data set name for clusters
OUTS=	specifies output data set name for cluster solutions
Specify variables in output data sets	
CLUSTER=	specifies variable in the OUT= and OUTCLUS= data sets identifying clusters
DENSITY=	specifies variable in the OUT= data set containing density estimates
OUTLENGTH=	specifies length of variables in the output data sets
Summarize and process coordinate data before clustering	
SIMPLE	requests simple statistics
STANDARD	standardizes the variables to mean 0 and standard deviation 1
Specify smoothing parameters	
DK=	specifies number of neighbors to use for kth-nearest-neighbor density estimation
CK=	specifies number of neighbors to use for clustering
K=	specifies number of neighbors to use for kth-nearest-neighbor density estimation and clustering

Table 57.1 *continued*

Option	Description
DR=	specifies radius of the sphere of support for uniform-kernel density estimation
CR=	specifies radius of the neighborhood for clustering
R=	specifies radius of the sphere of support for uniform-kernel density estimation and the neighborhood clustering
Specify density estimation options	
CASCADE=	specifies number of times the density estimates are to be cascaded
CROSS or CROSSLIST	computes the likelihood cross validation criterion
DIMENSION=	specifies dimensionality to be used when computing density estimates
AM	uses arithmetic means for cascading density estimates
HM	uses harmonic means for cascading density estimates
SUM	uses sums for cascading density estimates
Specify clustering methods and options	
DOCK	dissolves clusters with *n* or fewer members
EARLY	stops the analysis after obtaining a solution with either no cluster or a single cluster
JOIN(=)	requests that nonsignificant clusters be hierarchically joined
MAXCLUSTERS=	specifies maximum number of clusters to be obtained with METHOD=6
METHOD=	specifies clustering method to use
MODE=	specifies minimum members for either cluster to be designated a modal cluster when two clusters are joined using METHOD=5
POWER=	specifies power of the density used with METHOD=6
TEST	specifies approximate significance tests for the number of clusters
THRESHOLD=	specifies assignment threshold used with METHOD=6
Specify the output display options	
ALL	produces all optional output
BOUNDARY	displays the density and cluster membership of observations with neighbors belonging to a different cluster
CORE	retains the neighbor lists for each observation in memory
CORSS	displays the estimated cross validated log density of each observation
CROSSLIST	displays the estimated density and cluster membership of each observation
LOCAL	displays estimates of local dimensionality and writes them to the OUT=data set
NEIGHBOR	displays the neighbors of each observation
NOPRINT	suppresses the display of the output

Table 57.1 *continued*

Option	Description
NOSUMMARY	suppresses the display of the summary of the number of clusters, number of unassigned observations, and maximum *p*-value for each analysis
SHORT	suppresses the display of statistics for each cluster
TRACE	traces the cluster assignments when METHOD=6

You can specify at least one of the following options for smoothing parameters for density estimation: DK=, K=, DR=, or R=. To obtain a cluster analysis, you can specify the METHOD= option and at least one of the following smoothing parameters for clustering: CK=, K=, CR=, or R=. If you want significance tests for the number of clusters, you should specify either the DR= or R= option. If none of the smoothing parameters is specified, the MODECLUS procedure provides a default value for the R= option. See the section "Density Estimation" on page 4109 for the formula of a reasonable first guess for R= and a discussion of smoothing parameters.

You can specify lists of values for the DK=, CK=, K=, DR=, CR=, and R= options. Numbers in the lists can be separated by blanks or commas. You can include in the lists one or more items of the form *start* TO *stop* BY *increment*. Each list can contain either one value or the same number of values as in every other list that contains more than one value. If a list has only one value, that value is used in combination with all the values in longer lists. If two or more lists have more than one value, then one analysis is done by using the first value in each list, another analysis is done by using the second value in each list, and so on.

You can specify the following options in the PROC MODECLUS statement.

ALL

produces all optional output.

AM

specifies arithmetic means for cascading density estimates. See the description of the CASCADE= option.

BOUNDARY

displays the density and cluster membership of observations with neighbors belonging to a different cluster.

CASCADE=*n*

CASC=*n*

specifies the number of times the density estimates are to be cascaded (see the section "Density Estimation" on page 4109). The default value 0 performs no cascading.

You can specify a list of values for the CASCADE= option. Each value in the list is combined with each combination of smoothing parameters to produce a separate analysis.

CK=*n*

specifies the number of neighbors to use for clustering. The number of neighbors should be at least two but less than the number of observations. See the section "Density Estimation" on page 4109 for details.

CLUSTER=*name*

provides a name for the variable in the OUT= and OUTCLUS= data sets identifying clusters. The default name is CLUSTER.

CORE

keeps the neighbor lists for each observation in the computer memory to make small problems run faster.

CR=*n*

specifies the radius of the neighborhood for clustering. See the section "Density Estimation" on page 4109 for details.

CROSS

computes the likelihood cross validation criterion (Silverman 1986, pp. 52–55). This option appears to be of limited usefulness. See the section "Density Estimation" on page 4109 for details.

CROSSLIST

displays the cross validated log density of each observation.

DATA=*SAS-data-set*

specifies the input data set containing observations to be clustered. If you omit the DATA= option, the most recently created SAS data set is used.

If the data set is TYPE=DISTANCE, the data are interpreted as a distance matrix. The number of variables must equal the number of observations in the data set or in each BY group. The distances are assumed to be Euclidean, but the procedure accepts other types of distances or dissimilarities. Unlike the CLUSTER procedure, PROC MODECLUS uses the entire distance matrix, not just the lower triangle; the distances are not required to be symmetric. The neighbors of a given observation are determined solely from the distances in that observation. Missing values are considered infinite. Various distance measures can be computed from coordinate data by using the DISTANCE procedure (for detailed information, see Chapter 32, "The DISTANCE Procedure").

If the data set is not TYPE=DISTANCE, the data are interpreted as coordinates in a Euclidean space, and Euclidean distances are computed. The variables can be discrete or continuous and should be at the interval level of measurement.

DENSITY=*name*

provides a name for the variable in the OUT= data set containing density estimates. The default name is DENSITY.

DIMENSION=*n*

DIM=*n*

specifies the dimensionality to be used when computing density estimates. The default is the number of variables if the data are coordinates; the default is 1 if the data are distances.

DK=*n*

specifies the number of neighbors to use for kth-nearest-neighbor density estimation. The number of neighbors should be at least two but less than the number of observations. See the section "Density Estimation" on page 4109 for details.

DOCK=*n*

dissolves clusters with n or fewer members by making the members unassigned.

DR=*n*

specifies the radius of the sphere of support for uniform-kernel density estimation. See the section "Density Estimation" on page 4109 for details.

EARLY

stops the cluster analysis after obtaining either a solution with no cluster or a solution with one cluster to which all observations are assigned. The smoothing parameters should be specified in increasing order. This can reduce the computer time required for the analysis but might occasionally miss some multiple-cluster solutions.

HM

uses harmonic means for cascading density estimates. See the description of the CASCADE= option for details.

JOIN<=*p*>

requests that nonsignificant clusters be hierarchically joined. The JOIN option implies the TEST option. After each solution is obtained, the cluster with the largest approximate p-value is either joined to a neighboring cluster or, if there is no neighboring cluster, dissolved by making all of its members unassigned. After two clusters are joined, an analysis of the remaining clusters is displayed.

If you do not specify a p-value with the JOIN= option, joining continues until only one cluster remains, and the results are written to the output data sets after each analysis. If you specify a p-value with the JOIN= option, joining continues until the greatest approximate p-value is less than the value given in the JOIN= option, and only if there is more than one cluster are the results for that analysis written to the output data sets.

Any value of p less than 1E−8 is set to 1E−8.

K=*n*

specifies the number of neighbors to use for kth-nearest-neighbor density estimation and clustering. The number of neighbors should be at least two but less than the number of observations. Specifying K=n is equivalent to specifying both DK=n and CK=n. See the section "Density Estimation" on page 4109 for details.

LIST

displays the estimated density and cluster membership of each observation.

LOCAL

requests estimates of local dimensionality (Tukey and Tukey 1981, pp. 236–237).

MAXCLUSTERS=*n*

MAXC=*n*

specifies the maximum number of clusters to be obtained with the METHOD=6 option. By default, there is no fixed limit.

METHOD=*n*

MET=*n*

M=*n*

specifies what clustering method to use. Since these methods do not have widely recognized names, the methods are indicated by numbers from 0 to 6. The methods are described in the section "Clustering Methods" on page 4113. For most purposes, METHOD=1 is recommended, although METHOD=6 might occasionally produce better results in return for considerably greater computer time and space requirements. METHOD=1 is not good for discrete coordinate data with only a few equally spaced values. In this case, METHOD=6 or METHOD=3 works better. METHOD=4 or METHOD=5 is less desirable than other methods when there are ties, since a general characteristic of agglomerative hierarchical clustering methods is that the results are indeterminate in the presence of ties.

You must specify the METHOD= option to obtain a cluster analysis.

You can specify a list of values for the METHOD= option. Each value in the list is combined with each combination of smoothing and cascading parameters to produce a separate cluster analysis.

MODE=*n*

specifies that when two clusters are joined using the METHOD=5 option (no other methods are affected by the MODE= option), each must have at least n members for either cluster to be designated a modal cluster. In any case, each cluster must also have a maximum density greater than the fusion density for either cluster to be designated a modal cluster. If you specify the K= option, the default value of the MODE= option is the same as the value of the K= option because the use of kth-nearest-neighbor density estimation limits the resolution that can be obtained for clusters with fewer than k members. If you do not specify the K= option, the default is MODE=2. If you specify MODE=0, the default value is used instead of 0. If you specify a FREQ statement, the MODE= value is compared to the number of observations in each cluster, not to the sum of the frequencies.

NEIGHBOR

displays the neighbors of each observation in a table called "Nearest Neighbor List." See Nearest Neighbor List on page 4124 for information displayed in the table.

NOPRINT

suppresses the display of the output. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

NOSUMMARY

suppresses the display of the summary of the number of clusters, number of unassigned observations, and maximum p-value for each analysis.

OUT=*SAS-data-set*

specifies the output data set containing the input data plus density estimates, cluster membership, and variables identifying the type of solution. There is an output observation corresponding to each input observation for each solution. Therefore, the OUT= data set can be very large.

If you want to create a permanent SAS data set, you must specify a two-level name. For details about OUT= data sets, see the section "Output Data Sets" on page 4121. See *SAS Language Reference: Concepts* for more information about permanent SAS data sets.

OUTCLUS=*SAS-data-set*

OUTC=*SAS-data-set*

specifies the output data set containing an observation corresponding to each cluster in each solution. The variables identify the solution and contain statistics describing the clusters.

If you want to create a permanent SAS data set, you must specify a two-level name. For details about OUTCLUS= data sets, see the section "Output Data Sets" on page 4121. See *SAS Language Reference: Concepts* for more information about permanent SAS data sets.

OUTSUM=*SAS-data-set*

OUTS=*SAS-data-set*

specifies the output data set containing an observation corresponding to each cluster solution, giving the number of clusters and the number of unclassified observations for that solution.

If you want to create a permanent SAS data set, you must specify a two-level name. For details about OUTSUM= data sets, see the section "Output Data Sets" on page 4121. See *SAS Language Reference: Concepts* for more information about permanent SAS data sets.

OUTLENGTH=*n*

OUTL=*n*

specifies the length of those output variables that are not copied from the input data set but are created by PROC MODECLUS.

The OUTLENGTH= option applies only to the following variables that appear in all of the output data sets: _K_, _DK_, _CK_, _R_, _DR_, _CR_, _CASCAD_, _METHOD_, _NJOIN_, and _LOCAL_.

The minimum value is 2 or 3, depending on the operating system. The maximum value is 8. The default value is 8.

POWER=*n*

POW=*n*

specifies the power of the density used with the METHOD=6 option. The default value is 2.

R=*n*

specifies the radius of the sphere of support for uniform-kernel density estimation and the neighborhood for clustering. Specifying R=*n* is equivalent to specifying both DR=*n* and CR=*n*. See the section "Density Estimation" on page 4109 for details.

SHORT

suppresses the display of statistics for each cluster.

SIMPLE

S

displays means, standard deviations, skewness, kurtosis, and a coefficient of bimodality. The SIMPLE option applies only to coordinate data.

STANDARD

STD

standardizes the variables to mean 0 and standard deviation 1. The STANDARD option applies only to coordinate data.

SUM

uses sums for cascading density estimates. See the description of the CASCADE= option on page 4103 for details.

TEST

performs approximate significance tests for the number of clusters. The R= or DR= option must also be specified with a nonzero value to obtain significance tests.

The significance tests performed by PROC MODECLUS are valid only for simple random samples, and they require at least 20 observations per cluster to have enough power to be of any use. See the section "Significance Tests" on page 4115 for details.

THRESHOLD=*n*

THR=*n*

specifies the assignment threshold used with the METHOD=6 option. The default is 0.5.

TRACE

traces the process of cluster assignments when METHOD=6 is specified.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MODECLUS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC MODECLUS. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide.*

FREQ Statement

FREQ *variable* **;**

If one variable in the input data set represents the frequency of occurrence for other values in the observation, specify the variable's name in a FREQ statement. PROC MODECLUS then treats the data set as if each observation appeared *n* times, where *n* is the value of the FREQ variable for the observation. Nonintegral values of the FREQ variable are truncated to the largest integer less than the FREQ value.

ID Statement

ID *variable* **;**

The values of the ID variable identify observations in the displayed results and in the OUT= data set. If you omit the ID statement, each observation is identified by its observation number, and a variable called _OBS_ is written to the OUT= data set containing the original observation numbers.

VAR Statement

VAR *variables* **;**

The VAR statement specifies numeric variables to be used in the cluster analysis. If you omit the VAR statement, all numeric variables not specified in other statements are used.

Details: MODECLUS Procedure

Density Estimation

See Silverman (1986) or Scott (1992) for an introduction to nonparametric density estimation.

PROC MODECLUS uses hyperspherical uniform kernels of fixed or variable radius. The density estimate at a point is computed by dividing the number of observations within a sphere centered at

the point by the product of the sample size and the volume of the sphere. The size of the sphere is determined by the smoothing parameters that you are required to specify.

For fixed-radius kernels, specify the radius as a Euclidean distance with either the DR= or R= option. For variable-radius kernels, specify the number of neighbors desired within the sphere with either the DK= or K= option; the radius is then the smallest radius that contains at least the specified number of observations including the observation at which the density is being estimated. If you specify both the DR= or R= option and the DK= or K= option, the radius used is the maximum of the two indicated radii; this is useful for dealing with outliers.

It is convenient to refer to the sphere of support of the kernel at observation x_i as the *neighborhood* of x_i. The observations within the neighborhood of x_i are the *neighbors* of x_i. In some contexts, x_i is considered a neighbor of itself, but in other contexts it is not. The following notation is used in this chapter:

x_i	the ith observation
$\mathrm{d}(x,y)$	the distance between points x and y
n	the total number of observations in the sample
n_i	the number of observations within the neighborhood of x_i, including x_i itself
n_i^-	the number of observations within the neighborhood of x_i, not including x_i itself
N_i	the set of indices of neighbors of x_i, including i
N_i^-	the set of indices of neighbors of x_i, not including i
v_i	the volume of the neighborhood of x_i
$\hat{f}_i$	the estimated density at x_i
$\hat{f}_i^-$	the cross validated density estimate at x_i
C_k	the set of indices of observations assigned to cluster k
v	the number of variables or the dimensionality
s_l	standard deviation of the lth variable

The estimated density at x_i is

$$\hat{f}_i = \frac{n_i}{n v_i}$$

which indicates the number of neighbors of x_i divided by the product of the sample size and the volume of the neighborhood at x_i.

The density estimates provided by uniform kernels are not quite as good as those provided by some other types of kernels, but they are quite satisfactory for clustering. The significance tests for the number of clusters require the use of fixed-size uniform kernels.

There is no simple answer to the question of which smoothing parameter to use (Silverman 1986, pp. 43–61, 84–88, 98–99). It is usually necessary to try several different smoothing parameters. A reasonable first guess for the K= option is in the range of 0.1 to 1 times $n^{4/(v+4)}$, smaller values

being suitable for higher dimensionalities. A reasonable first guess for the R= option in many coordinate data sets is given by

$$\left[\frac{2^{v+2}(v+2)\Gamma(0.5v+1)}{nv^2}\right]^{1/(v+4)}\sqrt{\sum_{l=1}^{v}s_l^2}$$

which can be computed in a DATA step by using the GAMMA function for Γ. The MODECLUS procedure also provides this first guess as a default smoothing parameter if none of the options (DR=, CR=, R=, DK=, CK=, and K=) is specified. This formula is derived under the assumption that the data are sampled from a multivariate normal distribution and, therefore, tend to be too large (oversmooth) if the true distribution is multimodal. Robust estimates of the standard deviations might be preferable if there are outliers. If the data are distances, the factor $\sqrt{\sum {s_l}^2}$ can be replaced by an average root-mean-squared Euclidean distance divided by $\sqrt{2}$. To prevent outliers from appearing as separate clusters, you can also specify K=2 or CK=2 or, more generally, K=m or CK=m, $m \geq 2$, which in most cases forces clusters to have at least m members.

If the variables all have unit variance (for example, if you specify the STD option), you can use Table 57.2 to obtain an initial guess for the R= option.

Table 57.2 Reasonable First Guess for R= for Standardized Data

Number of Obs	Number of Variables									
	1	2	3	4	5	6	7	8	9	10
20	1.01	1.36	1.77	2.23	2.73	3.25	3.81	4.38	4.98	5.60
35	0.91	1.24	1.64	2.08	2.56	3.08	3.62	4.18	4.77	5.38
50	0.84	1.17	1.56	1.99	2.46	2.97	3.50	4.06	4.64	5.24
75	0.78	1.09	1.47	1.89	2.35	2.85	3.38	3.93	4.50	5.09
100	0.73	1.04	1.41	1.82	2.28	2.77	3.29	3.83	4.40	4.99
150	0.68	0.97	1.33	1.73	2.18	2.66	3.17	3.71	4.27	4.85
200	0.64	0.93	1.28	1.67	2.11	2.58	3.09	3.62	4.17	4.75
350	0.57	0.85	1.18	1.56	1.98	2.44	2.93	3.45	4.00	4.56
500	0.53	0.80	1.12	1.49	1.91	2.36	2.84	3.35	3.89	4.45
750	0.49	0.74	1.06	1.42	1.82	2.26	2.74	3.24	3.77	4.32
1000	0.46	0.71	1.01	1.37	1.77	2.20	2.67	3.16	3.69	4.23
1500	0.43	0.66	0.96	1.30	1.69	2.11	2.57	3.06	3.57	4.11
2000	0.40	0.63	0.92	1.25	1.63	2.05	2.50	2.99	3.49	4.03

One data-based method for choosing the smoothing parameter is likelihood cross validation (Silverman 1986, pp. 52–55). The cross validated density estimate at an observation is obtained by omitting the observation from the computations:

$$\hat{f}_i^- = \frac{n_i^-}{nv_i}$$

The (log) likelihood cross validation criterion is then computed as

$$\sum_{i=1}^{n}\log \hat{f}_i^-$$

The suggested smoothing parameter is the one that maximizes this criterion. With fixed-radius kernels, likelihood cross validation oversmooths long-tailed distributions; for purposes of clustering, it tends to undersmooth short-tailed distributions. With k-nearest-neighbor density estimation, likelihood cross validation is useless because it almost always indicates k=2.

Cascaded density estimates are obtained by computing initial kernel density estimates and then, at each observation, taking the arithmetic mean, harmonic mean, or sum of the initial density estimates of the observations within the neighborhood. The cascaded density estimates can, in turn, be cascaded, and so on. Let ${}_k\hat{f}_i$ be the density estimate at x_i cascaded k times. For all types of cascading, ${}_0\hat{f}_i = \hat{f}_i$. If the cascading is done by arithmetic means, then, for $k \geq 0$,

$$ {}_{k+1}\hat{f}_i = \sum_{j \in N_i} {}_k\hat{f}_j / n_i $$

For harmonic means,

$$ {}_{k+1}\hat{f}_i = \left(\sum_{j \in N_i} {}_k\hat{f}_j^{-1} / n_i \right)^{-1} $$

and for sums,

$$ {}_{k+1}\hat{f}_i = \left(\sum_{j \in N_i} {}_k\hat{f}_j^{k+1} \right)^{\frac{1}{k+2}} $$

To avoid cluttering formulas, the symbol $\hat{f}_i$ is used in the rest of the chapter to denote the density estimate at x_i whether cascaded or not, since the clustering methods and significance tests do not depend on the degree of cascading.

Cascading increases the smoothness of the estimates with less computation than would be required by increasing the smoothing parameters to yield a comparable degree of smoothness. For population densities with bounded support and discontinuities at the boundaries, cascading improves estimates near the boundaries. Cascaded estimates, especially using sums, might be more sensitive to the local covariance structure of the distribution than are the uncascaded kernel estimates. Cascading seems to be useful for detecting very nonspherical clusters. Cascading was suggested by Tukey and Tukey (1981, p. 237). Additional research into the properties of cascaded density estimates is needed.

Clustering Methods

The number of clusters is a function of the smoothing parameters. The number of clusters tends to decrease as the smoothing parameters increase, but the relationship is not strictly monotonic. Generally, you should specify several different values of the smoothing parameters to see how the number of clusters varies.

The clustering methods used by PROC MODECLUS use spherical clustering neighborhoods of fixed or variable radius that are similar to the spherical kernels used for density estimation. For fixed-radius neighborhoods, specify the radius as a Euclidean distance with either the CR= or R= option. For variable-radius neighborhoods, specify the number of neighbors desired within the sphere with either the CK= or K= option; the radius is then the smallest radius that contains at least the specified number of observations including the observation for which the neighborhood is being determined. However, in the following descriptions of clustering methods, an observation is not considered to be one of its own neighbors. If you specify both the CR= or R= option and the CK= or K= option, the radius used is the maximum of the two indicated radii; this is useful for dealing with outliers. In this section, the symbols N_i, N_i^-, n_i, and n_i^- refer to clustering neighborhoods, not density estimation neighborhoods.

METHOD=0

Begin with each observation in a separate cluster. For each observation and each of its neighbors, join the cluster to which the observation belongs with the cluster to which the neighbor belongs. This method does not use density estimates. With a fixed clustering radius, the clusters are those obtained by cutting the single linkage tree at the specified radius (see Chapter 29, "The CLUSTER Procedure").

METHOD=1

Begin with each observation in a separate cluster. For each observation, find the nearest neighbor with a greater estimated density. If such a neighbor exists, join the cluster to which the observation belongs with the cluster to which the specified neighbor belongs.

Next, consider each observation with density estimates equal to that of one or more neighbors but not less than the estimate at any neighbor. Join the cluster containing the observation with (1) each cluster containing a neighbor of the observation such that the maximum density estimate in the cluster equals the density estimate at the observation and (2) the cluster containing the nearest neighbor of the observation such that the maximum density estimate in the cluster exceeds the density estimate at the observation.

This method is similar to the classification or assignment stage of algorithms described by Gitman (1973) and Huizinga (1978).

METHOD=2

Begin with each observation in a separate cluster. For each observation, find the neighbor with the greatest estimated density exceeding the estimated density of the observation. If such a neighbor exists, join the cluster to which the observation belongs with the cluster to which the specified neighbor belongs.

Observations with density estimates equal to that of one or more neighbors but not less than the estimate at any neighbor are treated the same way as they are in METHOD=1.

This method is similar to the first stage of an algorithm proposed by Mizoguchi and Shimura (1980).

METHOD=3

Begin with each observation in a separate cluster. For each observation, find the neighbor with greater estimated density such that the slope of the line connecting the point on the estimated density surface at the observation with the point on the estimated density surface at the neighbor is a maximum. That is, for observation x_i, find a neighbor x_j such that $(\hat{f}_j - \hat{f}_i)/\mathrm{d}(x_j, x_i)$ is a maximum. If this slope is positive, join the cluster to which observation x_i belongs with the cluster to which the specified neighbor x_j belongs. This method was invented by Koontz, Narendra, and Fukunaga (1976).

Observations with density estimates equal to that of one or more neighbors but not less than the estimate at any neighbor are treated the same way as they are in METHOD=1. The algorithm suggested for this situation by Koontz, Narendra, and Fukunaga (1976) might fail for flat areas in the estimated density that contain four or more observations.

METHOD=4

This method is equivalent to the first stage of two-stage density linkage (see Chapter 29, "The CLUSTER Procedure") without the use of the MODE=option.

METHOD=5

This method is equivalent to the first stage of two-stage density linkage (see Chapter 29, "The CLUSTER Procedure") with the use of the MODE=option.

METHOD=6

Begin with all observations unassigned.

Step 1: Form a list of seeds, each seed being a single observation such that the estimated density of the observation is not less than the estimated density of any of its neighbors. If you specify the MAXCLUSTERS=*n* option, retain only the *n* seeds with the greatest estimated densities.

Step 2: Consider each seed in decreasing order of estimated density, as follows:

1. If the current seed has already been assigned, proceed to the next seed. Otherwise, form a new cluster consisting of the current seed.
2. Add to the cluster any unassigned seed that is a neighbor of a member of the cluster or that shares a neighbor with a member of the cluster; repeat until no unassigned seed satisfies these conditions.
3. Add to the cluster all neighbors of seeds that belong to the cluster.
4. Consider each unassigned observation. Compute the ratio of the sum of the $p-1$ powers of the estimated density of the neighbors that belong to the current cluster to the sum of the $p-1$ powers of the estimated density of all of its neighbors, where p is specified by the POWER= option and is 2 by default. Let x_i be the current observation, and let k be the index of the current cluster. Then this ratio is

$$r_{ik} = \frac{\sum_{j \in N_i \cap C_k} \hat{f}_j^{p-1}}{\sum_{j \in N_i} \hat{f}_j^{p-1}}$$

(The sum of the $p-1$ powers of the estimated density of the neighbors of an observation is an estimate of the integral of the pth power of the density over the neighborhood.) If r_{ik} exceeds the maximum of 0.5 and the value of the THRESHOLD= option, add the observation x_i to the current cluster k. Repeat until no more observations can be added to the current cluster.

Step 3: (This step is performed only if the value of the THRESHOLD= option is less than 0.5.) Form a list of unassigned observations in decreasing order of estimated density. Repeat the following actions until the list is empty:

1. Remove the first observation from the list, such as observation x_i.
2. For each cluster k, compute r_{ik}.
3. If the maximum over clusters of r_{ik} exceeds the value of the THRESHOLD= option, assign observation x_i to the corresponding cluster and insert all observations of which the current observation is a neighbor into the list, keeping the list in decreasing order of estimated density.

METHOD=6 is related to a method invented by Koontz and Fukunaga (1972a) and discussed by Koontz and Fukunaga (1972b).

Significance Tests

Significance tests require that a fixed-radius kernel be specified for density estimation via the DR= or R= option. You can also specify the DK= or K= option, but only the fixed radius is used for the significance tests.

The purpose of the significance tests is as follows: given a simple random sample of objects from a population, obtain an estimate of the number of clusters in the population such that the probability

in repeated sampling that the estimate exceeds the true number of clusters is not much greater than α, $1\% \leq \alpha \leq 10\%$. In other words, a sequence of null hypotheses of the form

$$H_0^{(i)}: \text{The number of population clusters is } i \text{ or less}$$

where $i = 1, 2, \cdots, n$, is tested against the alternatives such as

$$H_a^{(i)}: \text{The number of population clusters exceeds } i$$

with a maximum experimentwise error rate of approximately α. The tests protect you from overestimating the number of population clusters. It is impossible to protect against underestimating the number of population clusters without introducing much stronger assumptions than are used here, since the number of population clusters could conceivably exceed the sample size.

The method for conducting significance tests is as follows:

1. Estimate densities by using fixed-radius uniform kernels.
2. Obtain preliminary clusters by a "valley-seeking" method. Other clustering methods could be used but would yield less power.
3. Compute an approximate p-value for each cluster by comparing the estimated maximum density in the cluster with the estimated maximum density on the cluster boundary.
4. Repeatedly join the least significant cluster with a neighboring cluster until all remaining clusters are significant.
5. Estimate the number of population clusters as the number of significant sample clusters.
6. The preceding steps can be repeated for any number of different radii, and the estimate of the number of population clusters can be taken to be the maximum number of significant sample clusters for any radius.

This method has the following useful features:

- No distributional assumptions are required.
- The choice of smoothing parameter is not critical since you can try any number of different values.
- The data can be coordinates or distances.
- Time and space requirements for the significance tests are no worse than those for obtaining the clusters.
- The power is high enough to be useful for practical purposes.

The method for computing the p-values is based on a series of plausible approximations. There are as yet no rigorous proofs that the method is infallible. Neither are there any asymptotic results. However, simulations for sample sizes ranging from 20 to 2000 indicate that the p-values are almost always conservative. The only case discovered so far in which the p-values are liberal is a uniform distribution in one dimension for which the simulated error rates exceed the nominal significance level only slightly for a limited range of sample sizes.

To make inferences regarding population clusters, it is first necessary to define what is meant by a cluster. For clustering methods that use nonparametric density estimation, a cluster is usually loosely defined as a region surrounding a local maximum of the probability density function or a maximal connected set of local maxima. This definition might not be satisfactory for very rough densities with many local maxima. It is not applicable at all to discrete distributions for which the density does not exist. As another example in which this definition is not intuitively reasonable, consider a uniform distribution in two dimensions with support in the shape of a figure eight (including the interior). This density might be considered to contain two clusters even though it does not have two distinct modes.

These difficulties can be avoided by defining clusters in terms of the local maxima of a smoothed probability density or mass function. For example, define the neighborhood distribution function (NDF) with radius r at a point x as the probability that a randomly selected point will lie within a radius r of x—that is, the probability integral over a hypersphere of radius r centered at x:

$$s(x) \quad = \quad P(\mathrm{d}(x, X) <= r)$$

where X is the random variable being sampled, r is a user-specified radius, and $\mathrm{d}(x,y)$ is the distance between points x and y.

The NDF exists for all probability distributions. You can select the radius according to the degree of resolution required. The minimum-variance unbiased estimate of the NDF at a point x is proportional to the uniform-kernel density estimate with corresponding support.

You can define a *modal region* as a maximal connected set of local maxima of the NDF. A cluster is a connected set containing exactly one modal region. This definition seems to give intuitively reasonable results in most cases. An exception is a uniform density on the perimeter of a square. The NDF has four local maxima. There are eight local maxima along the perimeter, but running PROC MODECLUS with the R= option would yield four clusters since the two local maxima at each corner are separated by a distance equal to the radius. While this density does indeed have four distinctive features (the corners), it is not obvious that each corner should be considered a cluster.

The number of population clusters depends on the radius of the NDF. The significance tests in PROC MODECLUS protect against overestimating the number of clusters at any specified radius. It is often useful to look at the clustering results across a range of radii. A plot of the number of sample clusters as a function of the radius is a useful descriptive display, especially for high-dimensional data (Wong and Schaack 1982).

If a population has two clusters, it must have two modal regions. If there are two modal regions, there must be a "valley" between them. It seems intuitively desirable that the boundary between the two clusters should follow the bottom of this valley. All the clustering methods in PROC MODECLUS are designed to locate the estimated cluster boundaries in this way, although methods 1

and 6 seem to be much more successful at this than the others. Regardless of the precise location of the cluster boundary, it is clear that the maximum of the NDF along the boundary between two clusters must be strictly less than the value of the NDF in either modal region; otherwise, there would be only a single modal region; according to Hartigan and Hartigan (1985), there must be a "dip" between the two modes. PROC MODECLUS assesses the significance of a sample cluster by comparing the NDF in the modal region with the maximum of the NDF along the cluster boundary. If the NDF has second-order derivatives in the region of interest and if the boundary between the two clusters is indeed at the bottom of the valley, then the maximum value of the NDF along the boundary occurs at a saddle point. Hence, this test is called a *saddle test*. This term is intended to describe any test for clusters that compares modal densities with saddle densities, not just the test currently implemented in the MODECLUS procedure.

The obvious estimate of the maximum NDF in a sample cluster is the maximum estimated NDF at an observation in the cluster. Let $m(k)$ be the index of the observation for which the maximum is attained in cluster k.

Estimating the maximum NDF on the cluster boundary is more complicated. One approach is to take the maximum NDF estimate at an observation in the cluster that has a neighbor belonging to another cluster. This method yields excessively large estimates when the neighborhood is large. Another approach is to try to choose an object closer to the boundary by taking the observation with the maximum sum of estimated densities of neighbors belonging to a different cluster. After some experimentation, it is found that a combination of these two methods works well. Let B_k be the set of indices of observations in cluster k that have neighbors belonging to a different cluster, and compute

$$\max_{i \in B_k} \left(0.2 \hat{f}_i n_i + \sum_{j \in N_i - C_k} \hat{f}_j \right)$$

Let $s(k)$ be the index of the observation for which the maximum is attained.

Using the notation $\#(S)$ for the cardinality of set S, let

$$\begin{aligned}
n_{ij}^{-} &= \#(N_i^{-} \cap N_j^{-}) \\
c_m(k) &= n_{m(k)}^{-} - n_{m(k)s(k)}^{-} \\
c_s(k) &= n_{s(k)}^{-} - n_{m(k)s(k)}^{-} \text{ if } B_k \neq \emptyset, \\
&= 0 \text{ otherwise} \\
q_k &= 1/2 \text{ if } B_k \neq \emptyset, \\
&= 2/3 \text{ otherwise} \\
z_k &= \frac{c_m(k) - q_k(c_m(k) + c_s(k)) - 1/2}{\sqrt{q_k(1-q_k)(c_m(k) + c_s(k))}} \\
u &= \left\lceil (0.2 + 0.05\sqrt{n}) \sum_{i:n_i > 1} \frac{1}{n_i + 1} \right\rceil
\end{aligned}$$

Let $R(u)$ be a random variable distributed as the range of a random sample of u observations from a standard normal distribution. Then the approximate p-value p_k for cluster k is

$$p_k = Pr(z_k > R(u)/\sqrt{2})$$

If points $m(k)$ and $s(k)$ are fixed a priori, z_k would be the usual approximately normal test statistic for comparing two binomial random variables. In fact, $m(k)$ and $s(k)$ are selected in such a way that $c_m(k)$ tends to be large and $c_s(k)$ tends to be small. For this reason, and because there might be a large number of clusters, each with its own z_k to be tested, each z_k is referred to the distribution of $R(u)$ instead of a standard normal distribution. If the tests are conducted for only one radius and if u is chosen equal to n, then the p-values are very conservative because (1) you are not making all possible pairwise comparisons of observations in the sample and (2) n_i^- and n_j^- are positively correlated if the neighborhoods overlap. In the formula for u, the summation overcorrects somewhat for the conservativeness due to correlated n_i^-'s. The factor $0.2 + 0.05\sqrt{n}$ is empirically estimated from simulation results to adjust for the use of more than one radius.

If the JOIN option is specified, the least significant cluster (the cluster with the smallest z_k) is either dissolved or joined with a neighboring cluster. If no members of the cluster have neighbors belonging to a different cluster, all members of the cluster are unassigned. Otherwise, the cluster is joined to the neighboring cluster such that the sum of density estimates of neighbors of the estimated saddle point belonging to it is a maximum. Joining clusters increases the power of the saddle test. For example, consider a population with two well-separated clusters. Suppose that, for a certain radius, each population cluster is divided into two sample clusters. None of the four sample clusters is likely to be significant, but after the two sample clusters corresponding to each population cluster are joined, the remaining two clusters can be highly significant.

The saddle test implemented in PROC MODECLUS has been evaluated by simulation from known distributions. Some results are given in the following three tables. In Table 57.3, samples of 20 to 2000 observations are generated from a one-dimensional uniform distribution. For sample sizes of 1000 or less, 2000 samples are generated and analyzed by PROC MODECLUS. For a sample size of 2000, only 1000 samples are generated. The analysis is done with at least 20 different values of the R= option spread across the range of radii most likely to yield significant results. The six central columns of the table give the observed error rates at the nominal error rates (α) at the head of each column. The standard errors of the observed error rates are given at the bottom of the table. The observed error rates are conservative for $\alpha \leq 5\%$, but they increase with α and become slightly liberal for sample sizes in the middle of the range tested.

Table 57.3 Observed Error Rates (%) for Uniform Distribution

Sample Size	Nominal Type 1 Error Rate						Number of Simulations
	1	2	5	10	15	20	
20	0.00	0.00	0.00	0.60	11.65	27.05	2000
50	0.35	0.70	4.50	10.95	20.55	29.80	2000
100	0.35	0.85	3.90	11.05	18.95	28.05	2000
200	0.30	1.35	4.00	10.50	18.60	27.05	2000
500	0.45	1.05	4.35	9.80	16.55	23.55	2000
1000	0.70	1.30	4.65	9.55	15.45	19.95	2000
2000	0.40	1.10	3.00	7.40	11.50	16.70	1000
Standard Error	0.22	0.31	0.49	0.67	0.80	0.89	2000
	0.31	0.44	0.69	0.95	1.13	1.26	1000

All unimodal distributions other than the uniform that have been tested, including normal, Cauchy, and exponential distributions and uniform mixtures, have produced much more conservative results. Table 57.4 displays results from a unimodal mixture of two normal distributions with equal variances and equal sampling probabilities and with means separated by two standard deviations. Any greater separation would produce a bimodal distribution. The observed error rates are quite conservative.

Table 57.4 Observed Error Rates (%) for Normal Mixture with 2σ Separation

Sample Size	Nominal Type 1 Error Rate						Number of Simulations
	1	2	5	10	15	20	
100	0.0	0.0	0.0	1.0	2.0	4.0	200
200	0.0	0.0	0.0	2.0	3.0	3.0	200
500	0.0	0.0	0.5	0.5	0.5	0.5	200

All distributions in two or more dimensions that have been tested yield extremely conservative results. For example, a uniform distribution on a circle yields observed error rates that are never more than one-tenth of the nominal error rates for sample sizes up to 1000. This conservatism is due to the fact that, as the dimensionality increases, more and more of the probability lies in the tails of the distribution (Silverman 1986, p. 92), and the saddle test used by PROC MODECLUS is more conservative for distributions with pronounced tails. This applies even to a uniform distribution on a hypersphere because, although the density does not have tails, the NDF does.

Since the formulas for the significance tests do not involve the dimensionality, no problems are created when the data are linearly dependent. Simulations of data in nonlinear subspaces (the circumference of a circle or surface of a sphere) have also yielded conservative results.

Table 57.5 displays results in terms of power for identifying two clusters in samples from a bimodal mixture of two normal distributions with equal variances and equal sampling probabilities separated by four standard deviations. In this simulation, PROC MODECLUS never indicated more than two significant clusters.

Table 57.5 Power (%) for Normal Mixture with 4σ Separation

Sample Size	Nominal Type 1 Error Rate						Number of Simulations
	1	2	5	10	15	20	
20	0.0	0.0	0.0	2.0	37.5	68.5	200
35	0.0	13.5	38.5	48.5	64.0	75.5	200
50	17.5	26.0	51.5	67.0	78.5	84.0	200
75	25.5	36.0	58.5	77.5	85.5	89.5	200
100	40.0	54.5	72.5	84.5	91.5	92.5	200
150	70.5	80.0	92.0	97.0	100.0	100.0	200
200	89.0	96.0	99.5	100.0	100.0	100.0	200

The saddle test is not as efficient as excess-mass tests for multimodality (Müller and Sawitzki 1991; Polonik 1993). However, there is not yet a general approximation for the distribution of excess-mass statistics to circumvent the need for simulations to do significance tests. See Minnotte (1992) for a review of tests for multimodality.

Computational Resources

The MODECLUS procedure stores coordinate data in memory if there is enough space. For distance data, only one observation at a time is in memory.

PROC MODECLUS constructs lists of the neighbors of each observation. The total space required is $12\sum n_i$ bytes, where n_i is based on the largest neighborhood required by any analysis. The lists are stored in a SAS utility data set unless you specify the CORE option. You might get an error message from the SAS System or from the operating system if there is not enough disk space for the utility data set. Clustering method 6 requires a second list that is always stored in memory.

For coordinate data, the time required to construct the neighbor lists is roughly proportional to $v(\log n)(\sum n_i)\log(\sum n_i/n)$. For distance data, the time is roughly proportional to $n^2\log(\sum n_i/n)$.

The time required for density estimation is proportional to $\sum n_i$ and is usually small compared to the time required for constructing the neighbor lists.

Clustering methods 0 through 3 are quite efficient, requiring time proportional to $\sum n_i$. Methods 4 and 5 are slower, requiring time roughly proportional to $(\sum n_i)\log(\sum n_i)$. Method 6 can also be slow, but the time requirements depend very much on the data and the particular options specified. Methods 4, 5, and 6 also require more memory than the other methods.

The time required for significance tests is roughly proportional to $g\sum n_i$, where g is the number of clusters.

PROC MODECLUS can process data sets of several thousand observations if you specify reasonable smoothing parameters. Very small smoothing values produce many clusters, whereas very large values produce many neighbors; either case can require excessive time or space.

Missing Values

If the data are coordinates, observations with missing values are excluded from the analysis.

If the data are distances, missing values are treated as infinite. The neighbors of each observation are determined solely by the distances in that observation. The distances are not required to be symmetric, and there is no check for symmetry; the neighbors of each observation are determined only from the distances in that observation. This treatment of missing values is quite different from that of the CLUSTER procedure, which ignores the upper triangle of the distance matrix.

Output Data Sets

The OUT= data set contains one complete copy of the input data set for each cluster solution. There are additional variables identifying each solution and giving information about individual

observations. Solutions with only one remaining cluster when JOIN=*p* is specified are omitted from the OUT= data set (see the description of the JOIN= option on page 4105). The OUT= data set can be extremely large, so it is advisable to specify the DROP= data set option to exclude unnecessary variables.

The OUTCLUS= or OUTC= data set contains one observation for each cluster in each cluster solution. The variables identify the solution and provide statistics describing the cluster.

The OUTSUM= or OUTS= data set contains one observation for each cluster solution. The variables identify the solution and provide information about the solution as a whole.

The following variables can appear in all of the output data sets:

- _K_, which is the value of the K= option for the current solution. This variable appears only if you specify the K= option.
- _DK_, which is the value of the DK= option for the current solution. This variable appears only if you specify the DK= option.
- _CK_, which is the value of the CK= option for the current solution. This variable appears only if you specify the CK= option.
- _R_, which is the value of the R= option for the current solution. This variable appears only if you specify the R= option.
- _DR_, which is the value of the DR= option for the current solution. This variable appears only if you specify the DR= option.
- _CR_, which is the value of the CR= option for the current solution. This variable appears only if you specify the CR= option.
- _CASCAD_, which is the number of times the density estimates have been cascaded for the current solution. This variable appears only if you specify the CASCADE= option.
- _METHOD_, which is the value of the METHOD= option for the current solution. This variable appears only if you specify the METHOD= option.
- _NJOIN_, which is the number of clusters that are joined or dissolved in the current solution. This variable appears only if you specify the JOIN option.
- _LOCAL_, which is the local dimensionality estimate of the observation. This variable appears only if you specify the LOCAL option.

The OUT= data set contains the following variables:

- the variables from the input data set
- _OBS_, which is the observation number from the input data set. This variable appears only if you omit the ID statement.
- DENSITY, which is the estimated density at the observation. This variable can be renamed by the DENSITY= option.

- CLUSTER, which is the number of the cluster to which the observation is assigned. This variable can be renamed by the CLUSTER= option.

The OUTC= data set contains the following variables:

- the BY variables, if any
- _NCLUS_, which is the number of clusters in the solution
- CLUSTER, which is the number of the current cluster
- _FREQ_, which is the number of observations in the cluster
- _MODE_, which is the maximum estimated density in the cluster
- _BFREQ_, which is the number of observations in the cluster with neighbors belonging to a different cluster
- _SADDLE_, which is the estimated saddle density for the cluster
- _MC_, which is the number of observations within the fixed-radius density-estimation neighborhood of the modal observation. This variable appears only if you specify the TEST or JOIN option.
- _SC_, which is the number of observations within the fixed-radius density-estimation neighborhood of the saddle observation. This variable appears only if you specify the TEST or JOIN option.
- _OC_, which is the number of observations within the overlap of the two previous neighborhoods. This variable appears only if you specify the TEST or JOIN option.
- _Z_, which is the approximate z statistic for the cluster. This variable appears only if you specify the TEST or JOIN option.
- _P_, which is the approximate p-value for the cluster. This variable appears only if you specify the TEST or JOIN option.

The OUTS= data set contains the following variables:

- the BY variables, if any
- _NCLUS_, which is the number of clusters in the solution
- _UNCL_, which is the number of unclassified observations
- _CROSS_, which is the likelihood cross validation criterion if you specify the CROSS or CROSSLIST option

Displayed Output

If you specify the SIMPLE option and the data are coordinates, PROC MODECLUS displays the following simple descriptive statistics for each variable:

- the mean
- the standard deviation
- the skewness
- the kurtosis
- a coefficient of bimodality (see Chapter 29, "The CLUSTER Procedure")

If you specify the NEIGHBOR option, PROC MODECLUS displays a list of neighbors for each observation. The table contains the following items:

- the observation number or ID value of the observation
- the observation number or ID value of each of its neighbors
- the distance to each neighbor

If you specify the CROSSLIST option, PROC MODECLUS produces a table of information regarding cross validation of the density estimates. Each table has a row for each observation. For each observation, the following are displayed:

- the observation number or ID value of the observation
- the radius of the neighborhood
- the number of neighbors
- the estimated log density
- the estimated cross validated log density

If you specify the LOCAL option, PROC MODECLUS produces a table of information regarding estimates of local dimensionality. Each table has a row for each observation. For each observation, the following are displayed:

- the observation number or ID value of the observation
- the radius of the neighborhood
- the estimated local dimensionality

If you specify the LIST option, PROC MODECLUS produces a table listing the observations within each cluster. The table includes the following items:

- the cluster number
- the observation number or ID value of the observation
- the estimated density
- the sum of the density estimates of observations within the neighborhood that belong to the same cluster
- the sum of the density estimates of observations within the neighborhood that belong to a different cluster
- the sum of the density estimates of all the observations within the neighborhood
- the ratio of the sum of the density estimates for the same cluster to the sum of all the density estimates in the neighborhood

If you specify the LIST option and there are unassigned objects, PROC MODECLUS produces a table listing those observations. The table includes the following items:

- the observation number or ID value of the observation
- the estimated density
- the ratio of the sum of the density estimates for the same cluster to the sum of the density estimates in the neighborhood for all other clusters

If you specify the BOUNDARY option, PROC MODECLUS produces a table listing the observations in each cluster that have a neighbor belonging to a different cluster. The table includes the following items:

- the observation number or ID value of the observation
- the estimated density
- the cluster number
- the ratio of the sum of the density estimates for the same cluster to the sum of the density estimates in the neighborhood for all other clusters

If you do not specify the SHORT option, PROC MODECLUS produces a table of cluster statistics including the following items:

- the cluster number
- the cluster frequency (the number of observations in the cluster)

- the maximum estimated density within the cluster
- the number of observations in the cluster having a neighbor that belongs to a different cluster
- the estimated saddle density of the cluster

If you specify the TEST or JOIN option, the table of cluster statistics includes the following items pertaining to the saddle test:

- the number of observations within the fixed-radius density-estimation neighborhood of the modal observation
- the number of observations within the fixed-radius density-estimation neighborhood of the saddle observation
- the number of observations within the overlap of the two preceding neighborhoods
- the z statistic for comparing the preceding counts
- the approximate p-value

If you do not specify the NOSUMMARY option, PROC MODECLUS produces a table summarizing each cluster solution containing the following items:

- the smoothing parameters and cascade value
- the number of clusters
- the frequency of unclassified objects
- the likelihood cross validation criterion if you specify the CROSS or CROSSLIST option

If you specify the JOIN option, the summary table also includes the following items:

- the number of clusters joined
- the maximum p-value of any cluster in the solution

If you specify the TRACE option, PROC MODECLUS produces a table for each cluster solution that lists each observation along with its cluster membership as it is reassigned from the "Old" cluster to the "New" cluster. This reassignment is described in **Step 1** through **Step 3** of the section "METHOD=6" on page 4114. Each table has a row for each observation. For each observation, the following are displayed:

- the observation number or ID value of the observation
- the estimated density

- the "Old" cluster membership. 0 represents an unassigned observation and –1 represents a seed.
- the "New" cluster membership
- "Ratio," which is documented in the section "METHOD=6" on page 4114. The following character values can also be displayed:

  "M" means the observation is a mode

  "S" means the observation is a seed

  "N" means the neighbor of a mode or seed, for which the ratio is not computed

ODS Table Names

PROC MODECLUS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 57.6.

For more information about ODS, see Chapter 20, "Using the Output Delivery System."

All of the ODS tables in Table 57.6 are created by specifying the PROC MODECLUS statement.

Table 57.6 ODS Tables Produced by PROC MODECLUS

ODS Table Name	Description	Option
BoundaryFreq	Boundary objects information	BOUNDARY or ALL
ClusterList	Cluster listing, cluster ID, frequency, density etc.	LIST or ALL
ClusterStats	Cluster statistics	default
	Cluster statistics, significance test statistics	TEST, JOIN, or ALL
ClusterSummary	Cluster summary	default
	Cluster summary, crossvalidation criterion	CROSS, CROSSLIS, or ALL
	Cluster summary, clusters joined information	JOIN or ALL
CrossList	Cross validated log density	CROSSLIST
ListLocal	Local dimensionality estimates	LOCAL
Neighbor	Nearest neighbor list	NEIGHBOR or ALL
SimpleStatistics	Simple statistics	SIMPLE or ALL
Trace	Trace of clustering algorithm (METHOD=6 only)	TRACE or ALL when METHOD=6
UnassignObjects	Information about unassigned objects	LIST or ALL

Examples: MODECLUS Procedure

Example 57.1: Cluster Analysis of Samples from Univariate Distributions

This example uses pseudo-random samples from a uniform distribution, an exponential distribution, and a bimodal mixture of two normal distributions. Results are presented in Output 57.1.1 through Output 57.1.18 as plots displaying both the true density and the estimated density, as well as cluster membership.

The following statements produce Output 57.1.1 through Output 57.1.4:

```
data uniform;
   title 'Modeclus Example with Univariate Distributions';
   title2 'Uniform Distribution';

   drop n;
   true=1;
   do n=1 to 100;
      x=ranuni(123);
      output;
   end;

proc modeclus data=uniform m=1 k=10 20 40 60 out=out short;
   var x;
run;
proc sgplot data=out noautolegend;
   y2axis label='True' values=(0 to 2 by 1.);
   yaxis values=(0 to 3 by 0.5);
   scatter y=density x=x / markerchar=cluster group=cluster;
   pbspline y=true x=x / y2axis nomarkers lineattrs=(thickness= 1);
   by _K_;
run;

proc modeclus data=uniform m=1 r=.05 .10 .20 .30
              out=out short;
   var x;
run;
proc sgplot data=out noautolegend;
   y2axis label='True' values=(0 to 2 by 1.);
   yaxis values=(0 to 2 by 0.5);
   scatter y=density x=x / markerchar=cluster group=cluster;
   pbspline y=true x=x / y2axis nomarkers lineattrs=(thickness= 1);
   by _R_;
run;
```

Output 57.1.1 Cluster Analysis of Sample from a Uniform Distribution

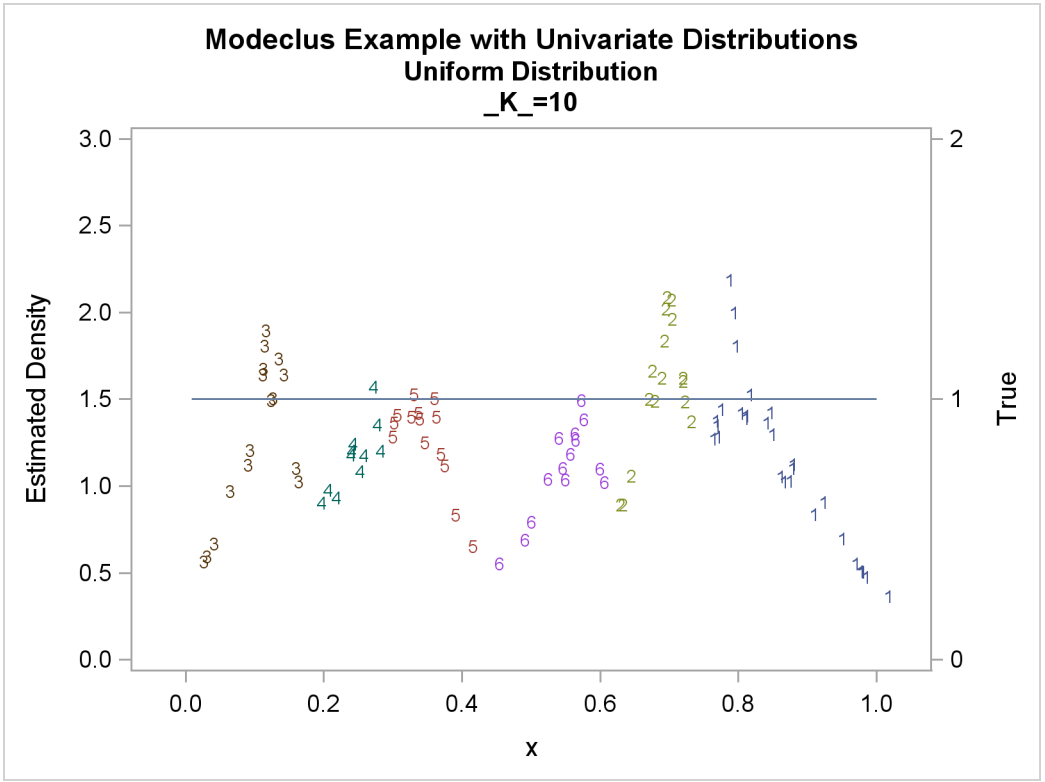
Modeclus Example with Univariate Distributions
Uniform Distribution

The MODECLUS Procedure

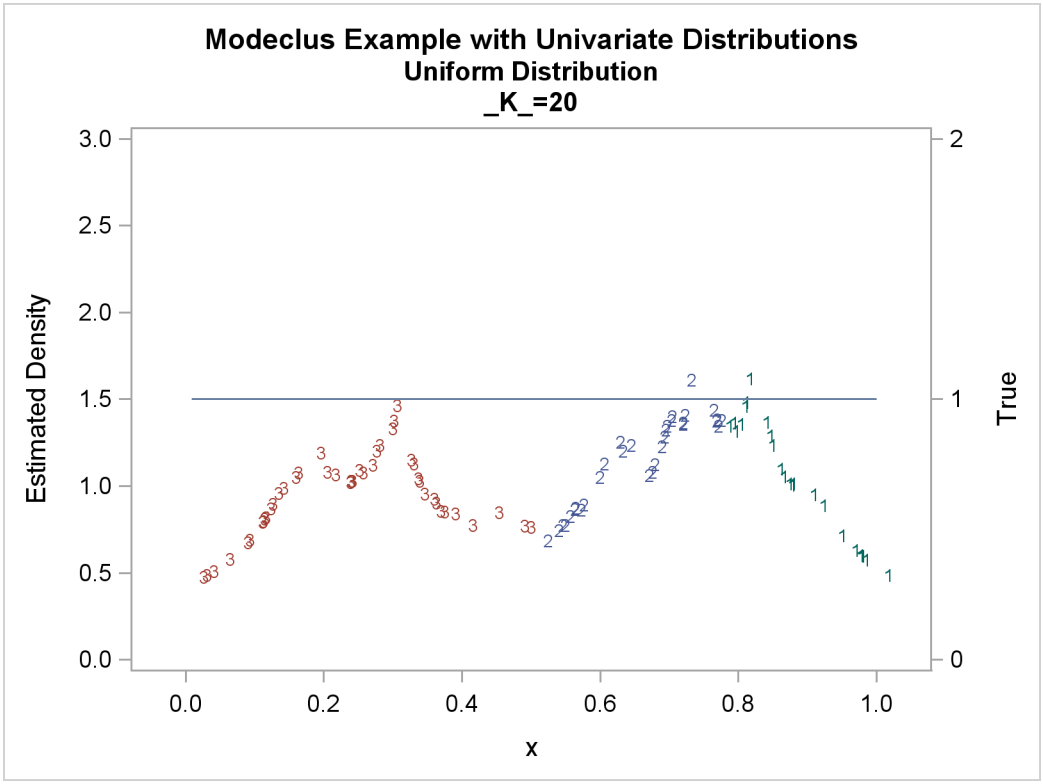
Cluster Summary

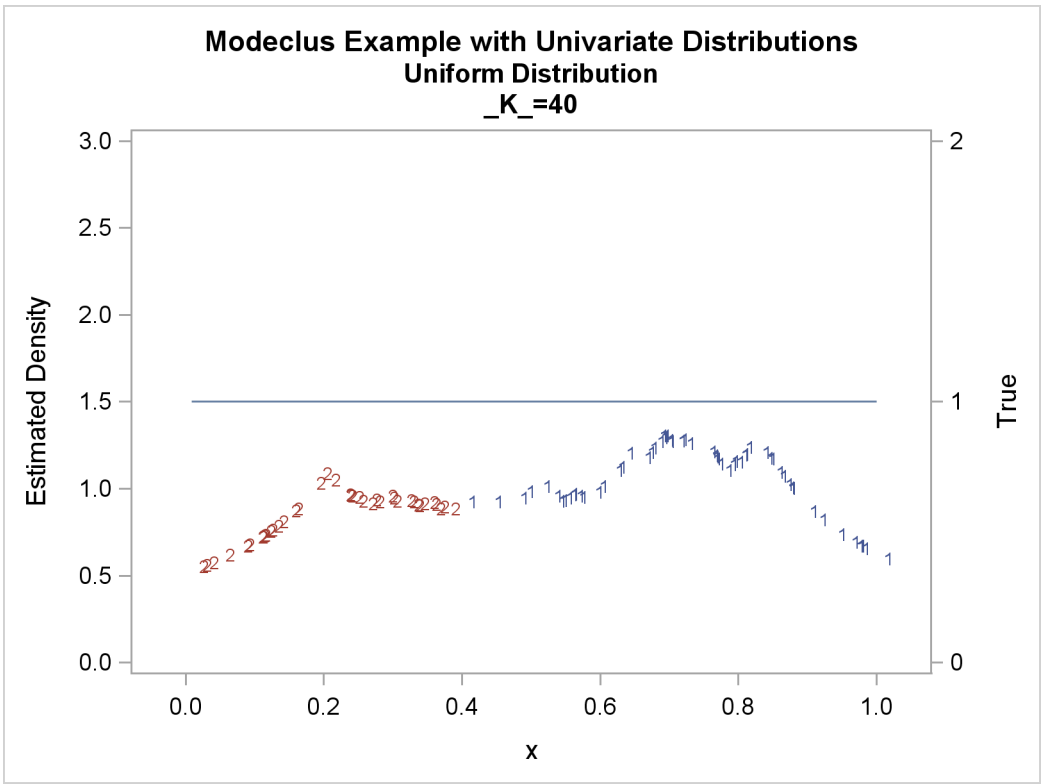
K	Number of Clusters	Frequency of Unclassified Objects
10	6	0
20	3	0
40	2	0
60	1	0

Output 57.1.2 True Density, Estimated Density, and Cluster Membership by Various _K_ Values

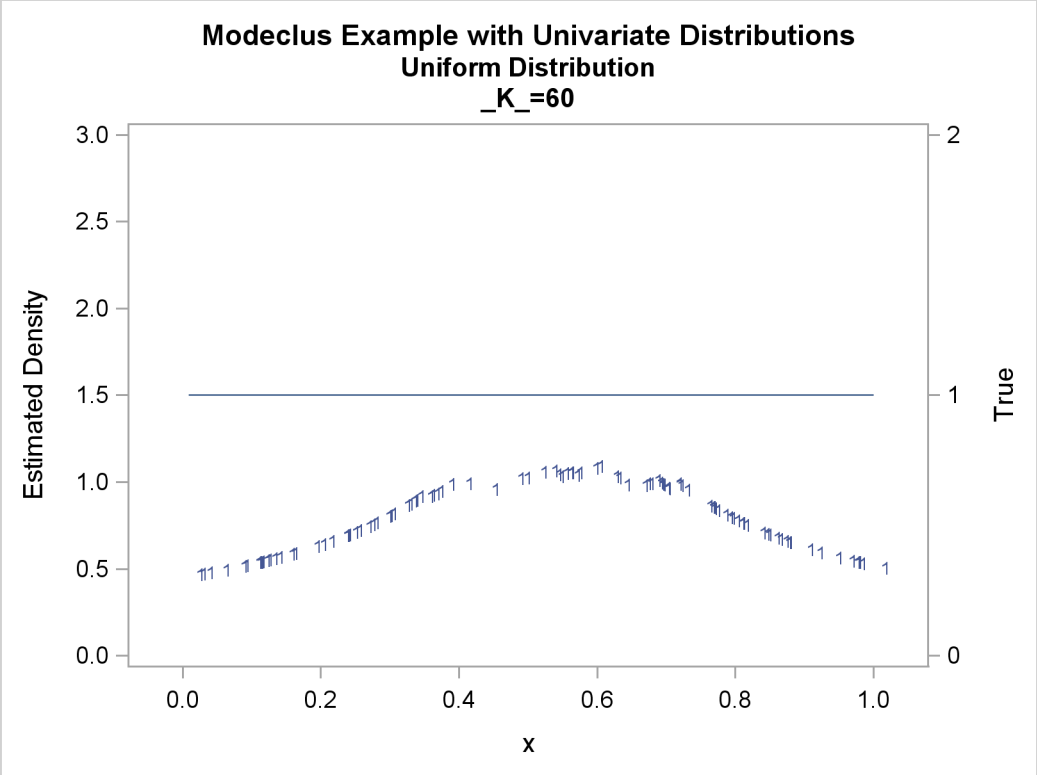


Output 57.1.2 *continued*





Output 57.1.2 *continued*



Output 57.1.3 Cluster Analysis of Sample from a Uniform Distribution

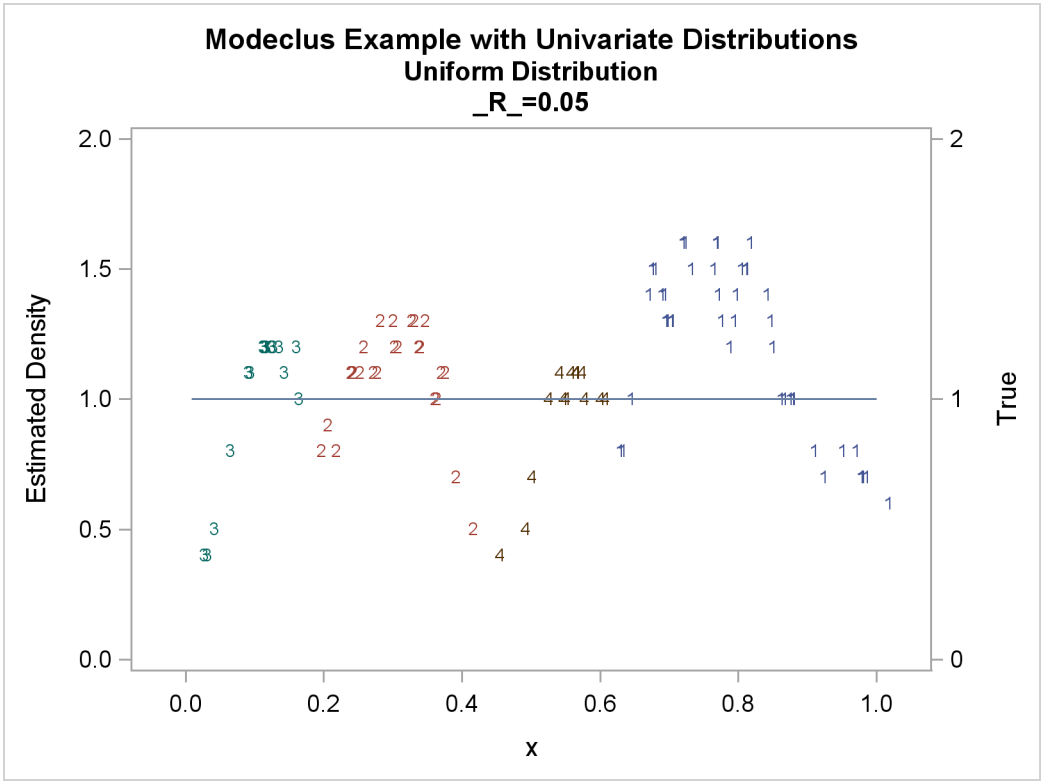
Modeclus Example with Univariate Distributions
Uniform Distribution

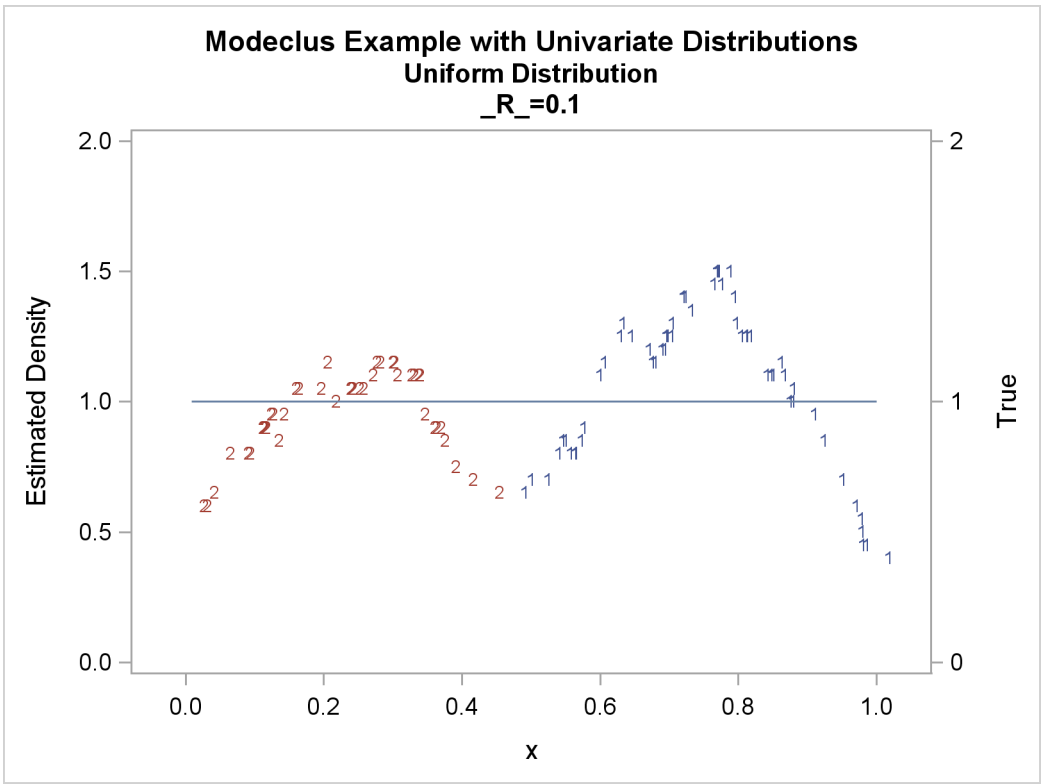
The MODECLUS Procedure

Cluster Summary

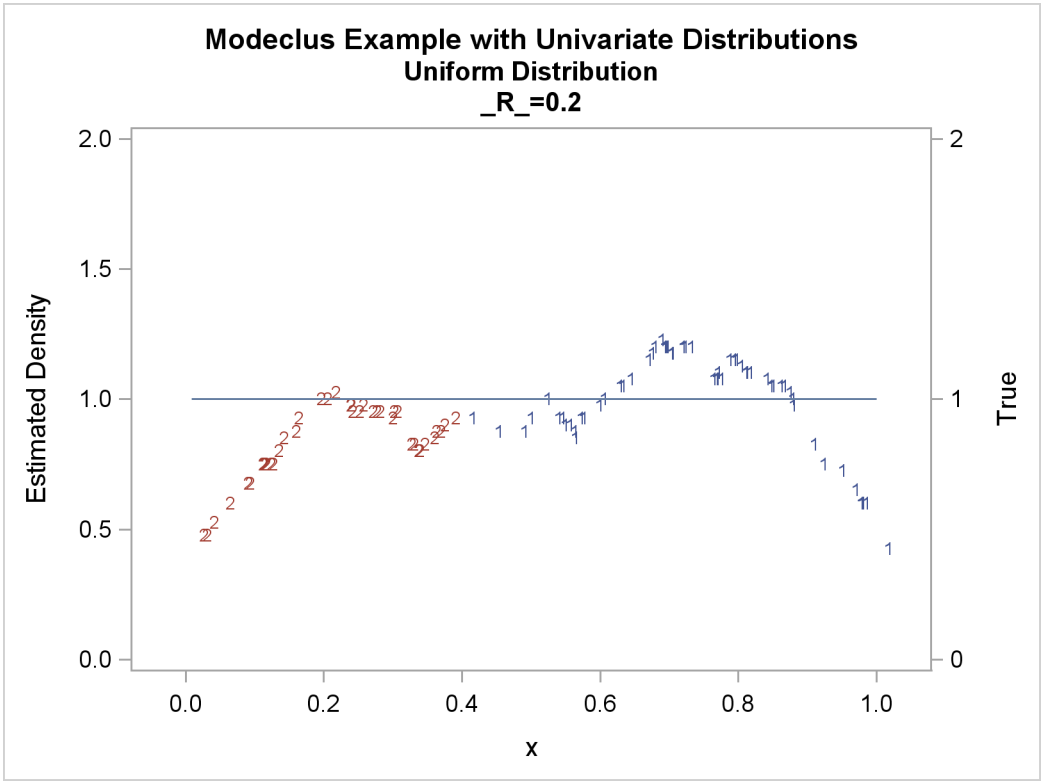
R	Number of Clusters	Frequency of Unclassified Objects
0.05	4	0
0.1	2	0
0.2	2	0
0.3	1	0

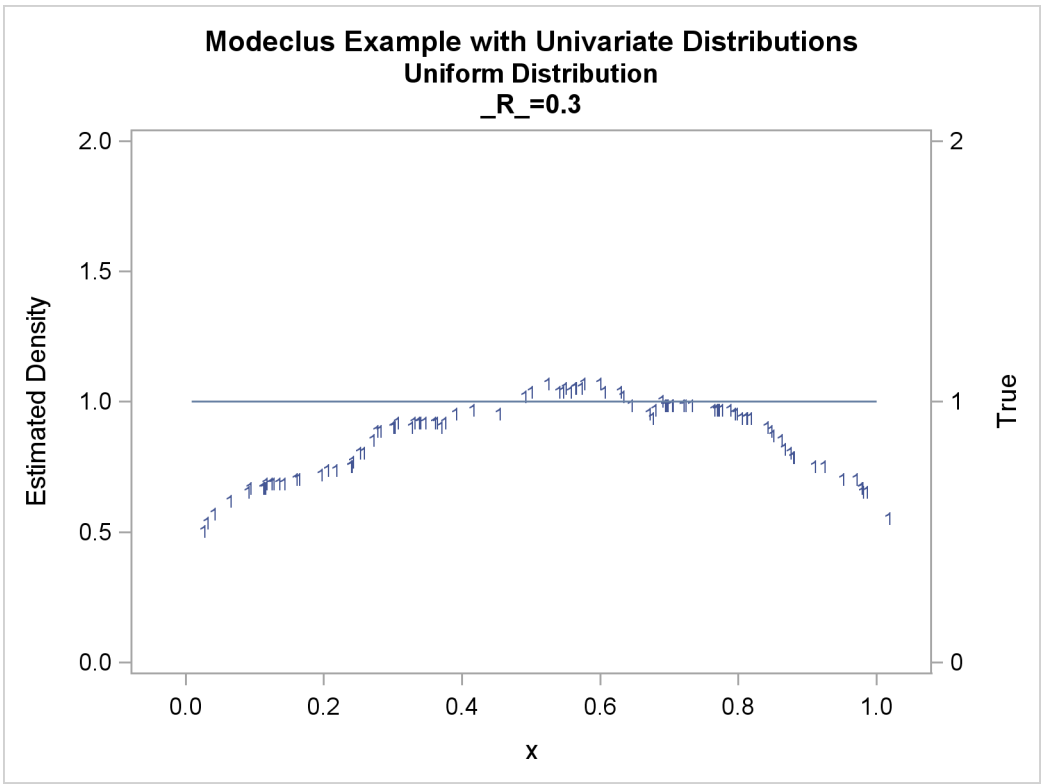
Output 57.1.4 True Density, Estimated Density, and Cluster Membership by Various _R_ Values





Output 57.1.4 *continued*





The following statements produce Output 57.1.5 through Output 57.1.12:

```
data expon;
   title2 'Exponential Distribution';
   drop n;
   do n=1 to 100;
      x=ranexp(123);
      true=exp(-x);
      output;
   end;

proc modeclus data=expon m=1 k=10 20 40 out=out short;
   var x;
run;
proc sgplot data=out noautolegend;
   y2axis label='True' values=(0 to 1 by .5);
   yaxis values=(0 to 2 by 0.5);
   scatter y=density x=x / markerchar=cluster group=cluster;
   pbspline y=true x=x / y2axis nomarkers lineattrs=(thickness= 1);
   by _K_;
run;


/*******************************************/

proc modeclus data=expon m=1 r=.20 .40 .80 out=out short;
   var x;
run;
proc sgplot data=out noautolegend;
   y2axis label='True' values=(0 to 1 by .5);
   yaxis values=(0 to 1 by 0.5);
   scatter y=density x=x / markerchar=cluster group=cluster;
   pbspline y=true x=x / y2axis nomarkers lineattrs=(thickness= 1);
   by _R_;
run;

/*******************************************/

title3 'Different Density-Estimation and Clustering Windows';
proc modeclus data=expon m=1 r=.20 ck=10 20 40
              out=out short;
   var x;
run;
proc sgplot data=out noautolegend;
   y2axis label='True' values=(0 to 1 by .5);
   yaxis values=(0 to 1 by 0.5);
   scatter y=density x=x / markerchar=cluster group=cluster;
   pbspline y=true x=x / y2axis nomarkers lineattrs=(thickness= 1);

   by _CK_;
run;

/*******************************************/
```

```
title3 'Cascaded Density Estimates Using Arithmetic Means';
proc modeclus data=expon m=1 r=.20 cascade=1 2 4 am out=out short;
   var x;
run;
proc sgplot data=out noautolegend;
   y2axis label='True' values=(0 to 1 by .5);
   yaxis values=(0 to 1 by 0.5);
   scatter y=density x=x / markerchar=cluster group=cluster;
   pbspline y=true x=x / y2axis nomarkers lineattrs=(thickness= 1);
   by _R_ _CASCAD_;
run;
```

Output 57.1.5 Cluster Analysis of Sample from an Exponential Distribution

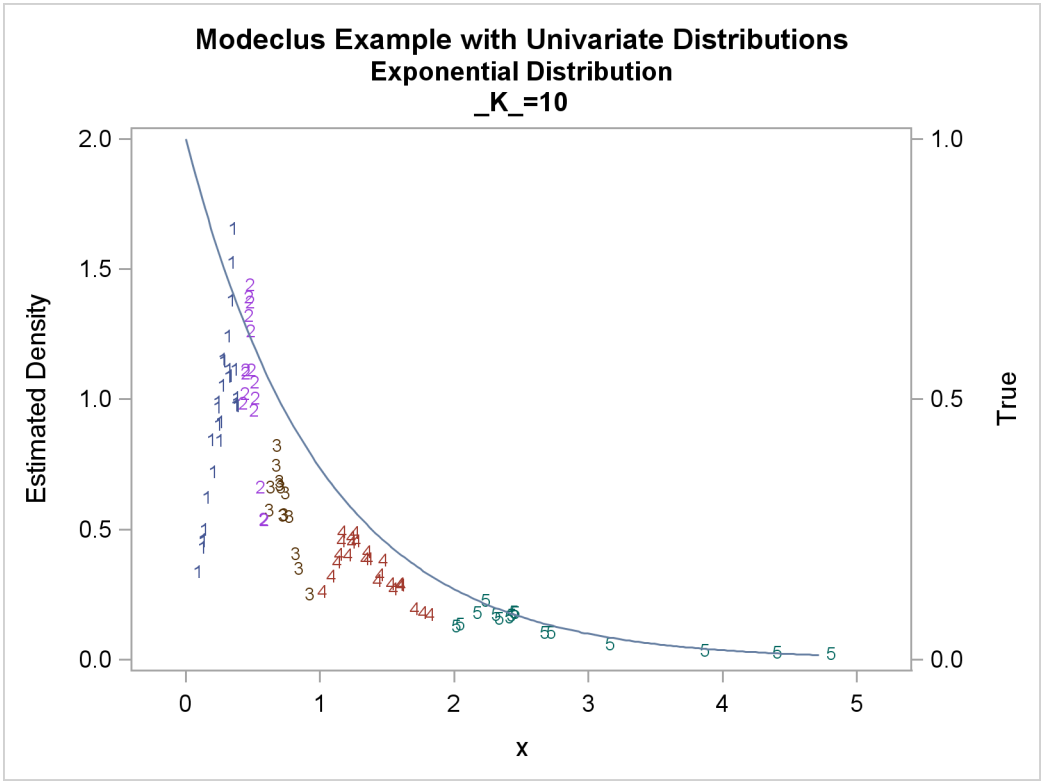
Modeclus Example with Univariate Distributions
Exponential Distribution

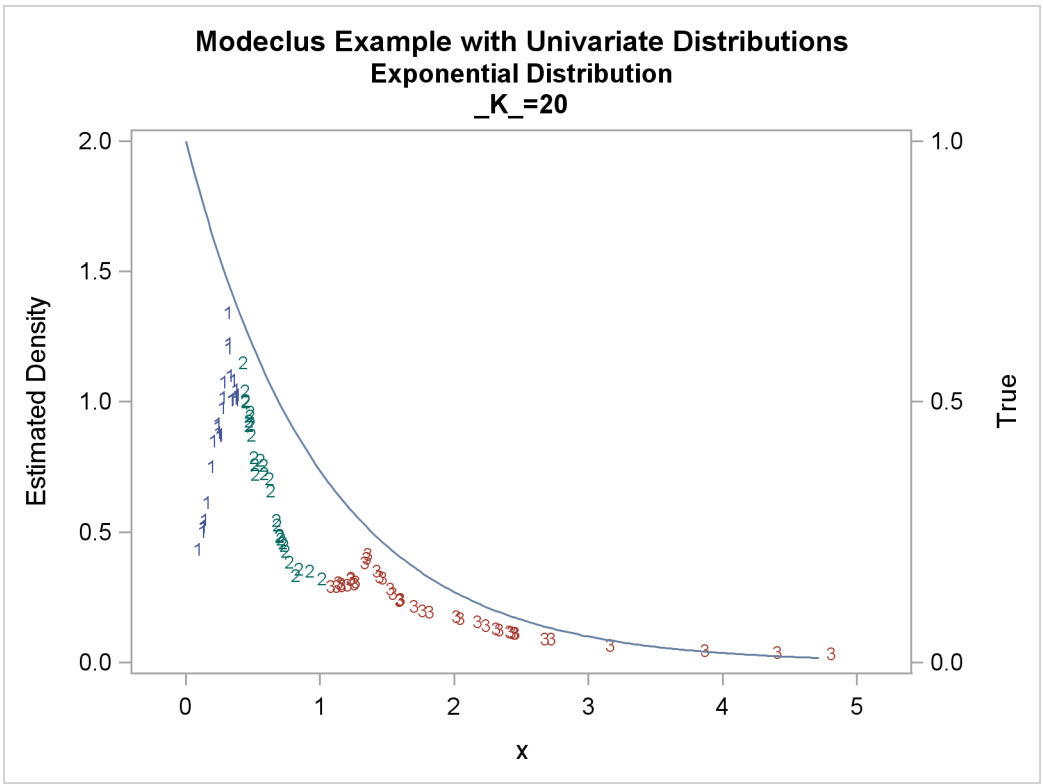
The MODECLUS Procedure

Cluster Summary

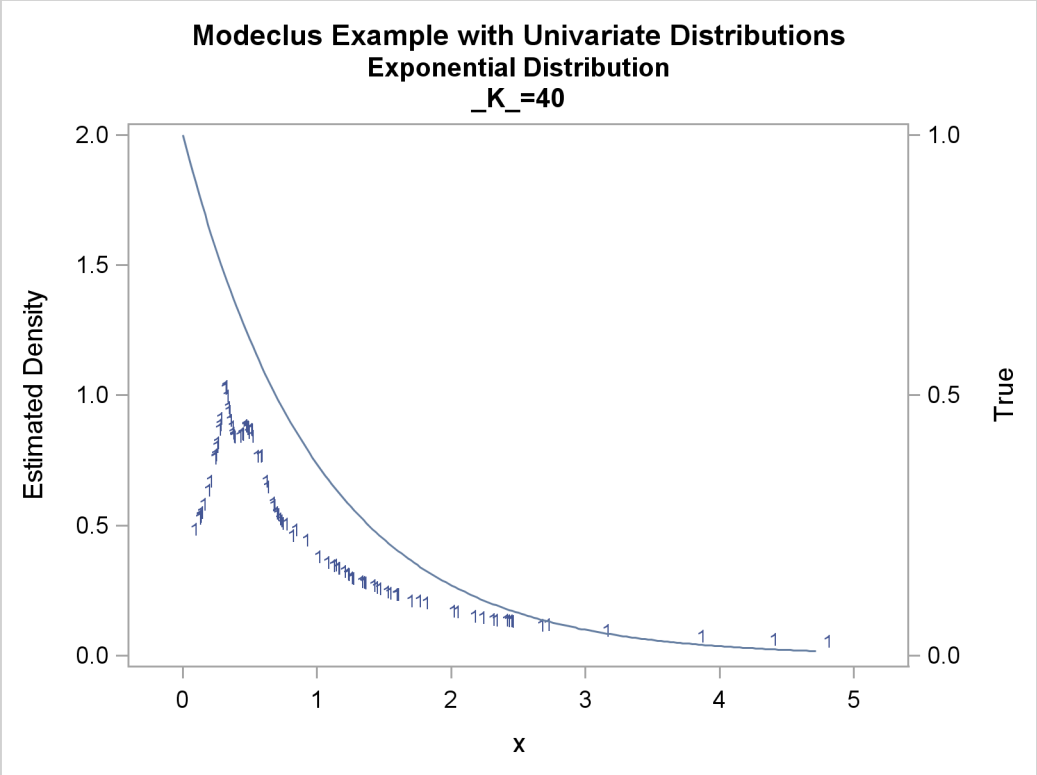
K	Number of Clusters	Frequency of Unclassified Objects
10	5	0
20	3	0
40	1	0

Output 57.1.6 True Density, Estimated Density, and Cluster Membership by Various _K_ Values





Output 57.1.6 *continued*



Output 57.1.7 Cluster Analysis of Sample from an Exponential Distribution

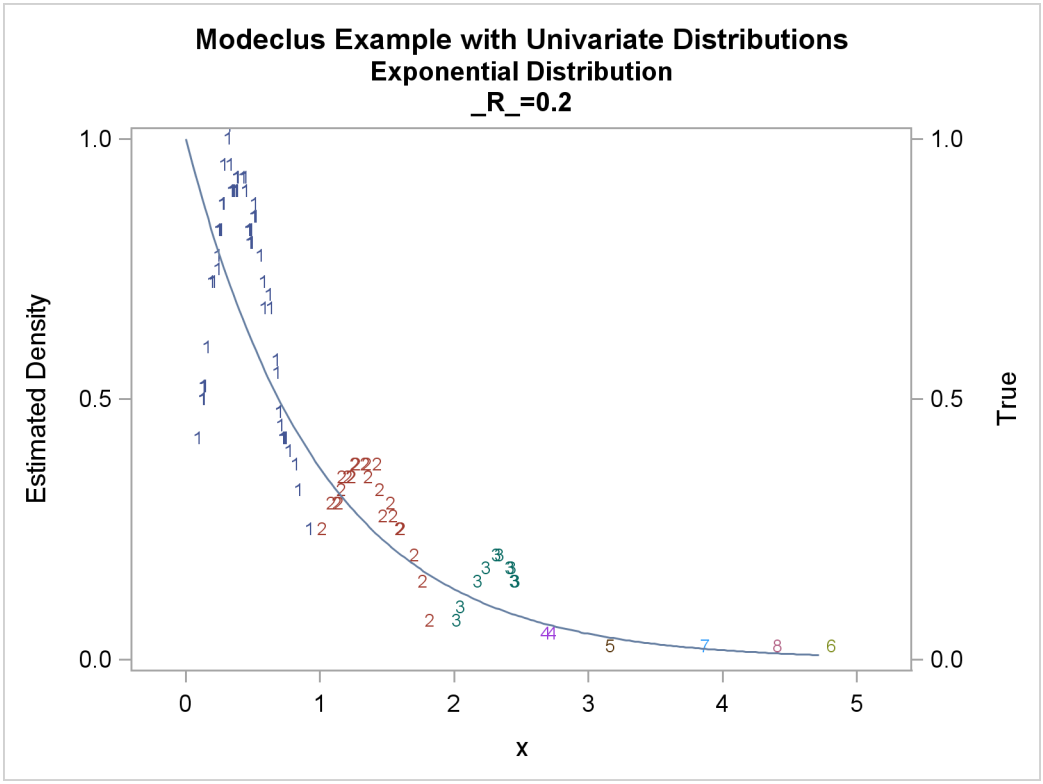
Modeclus Example with Univariate Distributions
Exponential Distribution

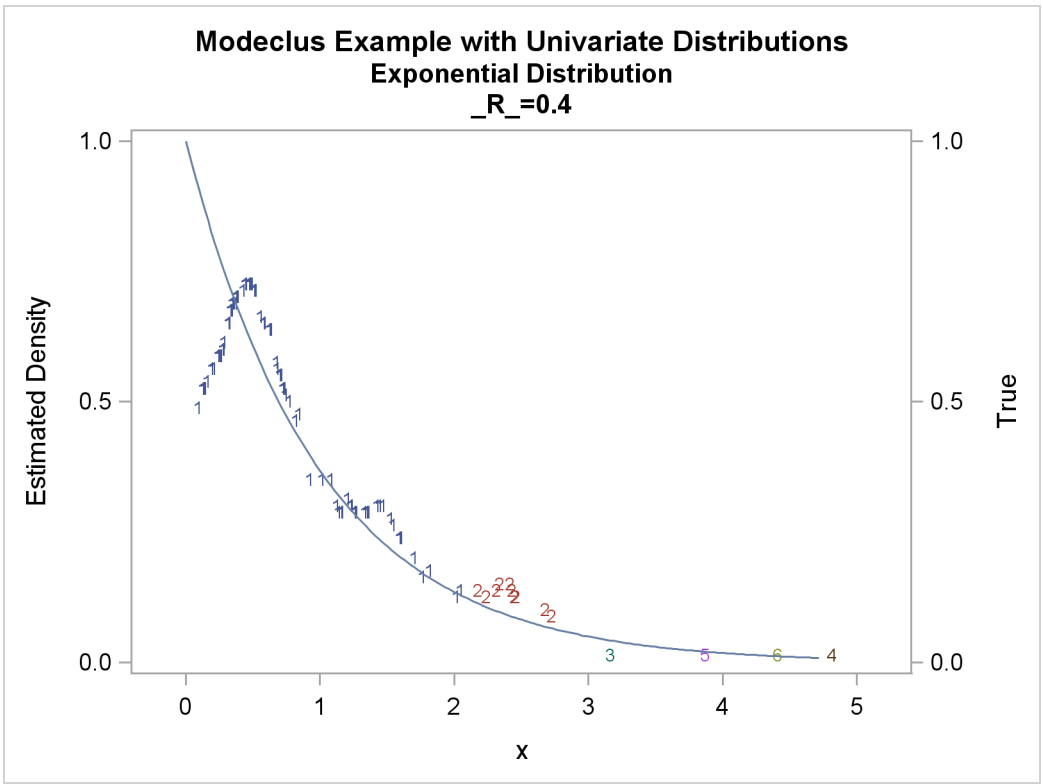
The MODECLUS Procedure

Cluster Summary

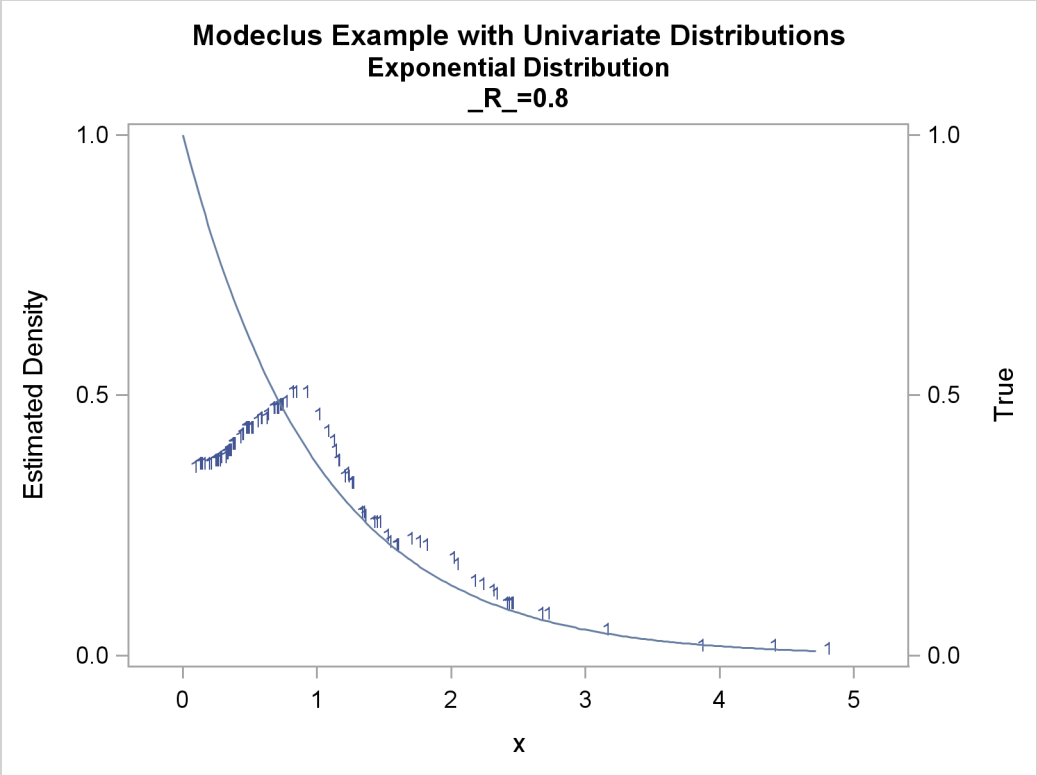
R	Number of Clusters	Frequency of Unclassified Objects
0.2	8	0
0.4	6	0
0.8	1	0

Output 57.1.8 True Density, Estimated Density, and Cluster Membership by Various _R_ Values





Output 57.1.8 *continued*



Output 57.1.9 Cluster Analysis of Sample from an Exponential Distribution

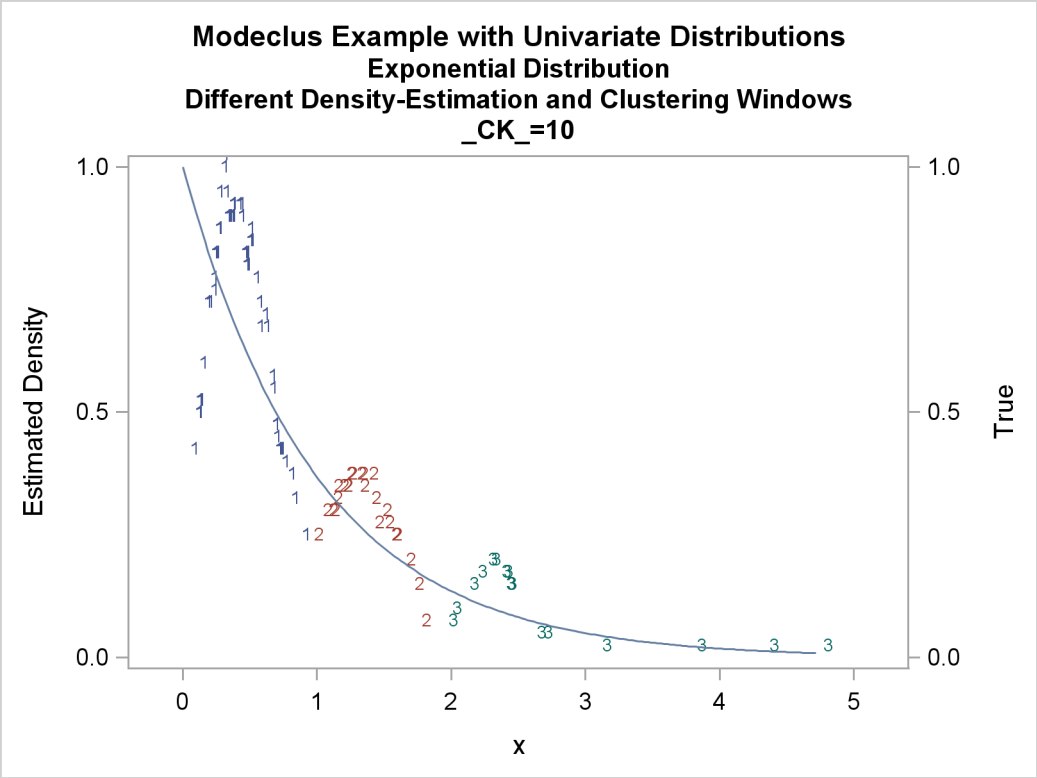
Modeclus Example with Univariate Distributions
Exponential Distribution
Different Density-Estimation and Clustering Windows

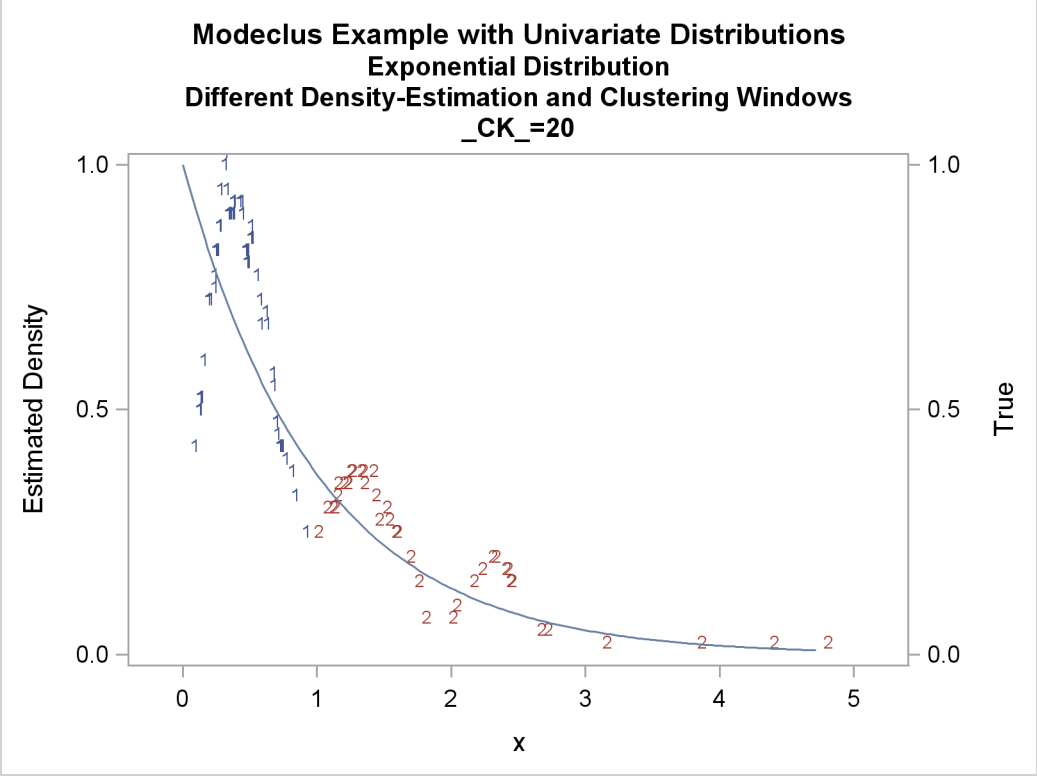
The MODECLUS Procedure

Cluster Summary

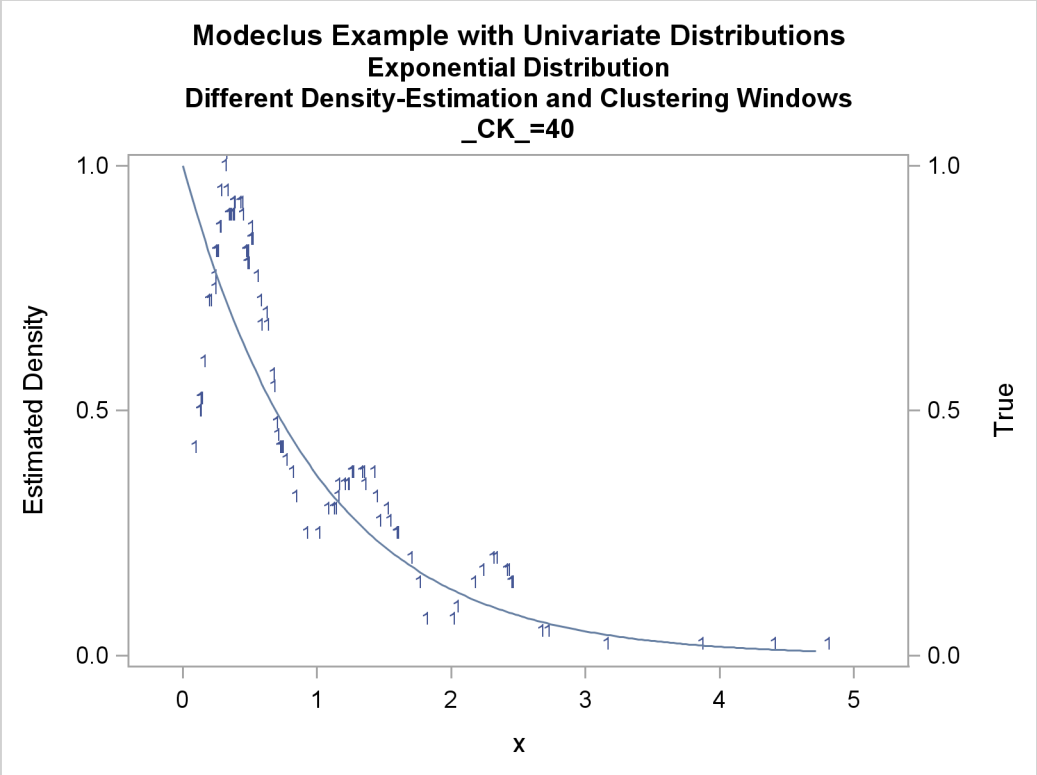
R	CK	Number of Clusters	Frequency of Unclassified Objects
0.2	10	3	0
0.2	20	2	0
0.2	40	1	0

Output 57.1.10 True Density, Estimated Density, and Cluster Membership by _R_=0.2 with Various _CK_ Values





Output 57.1.10 *continued*



Output 57.1.11 Cluster Analysis of Sample from an Exponential Distribution

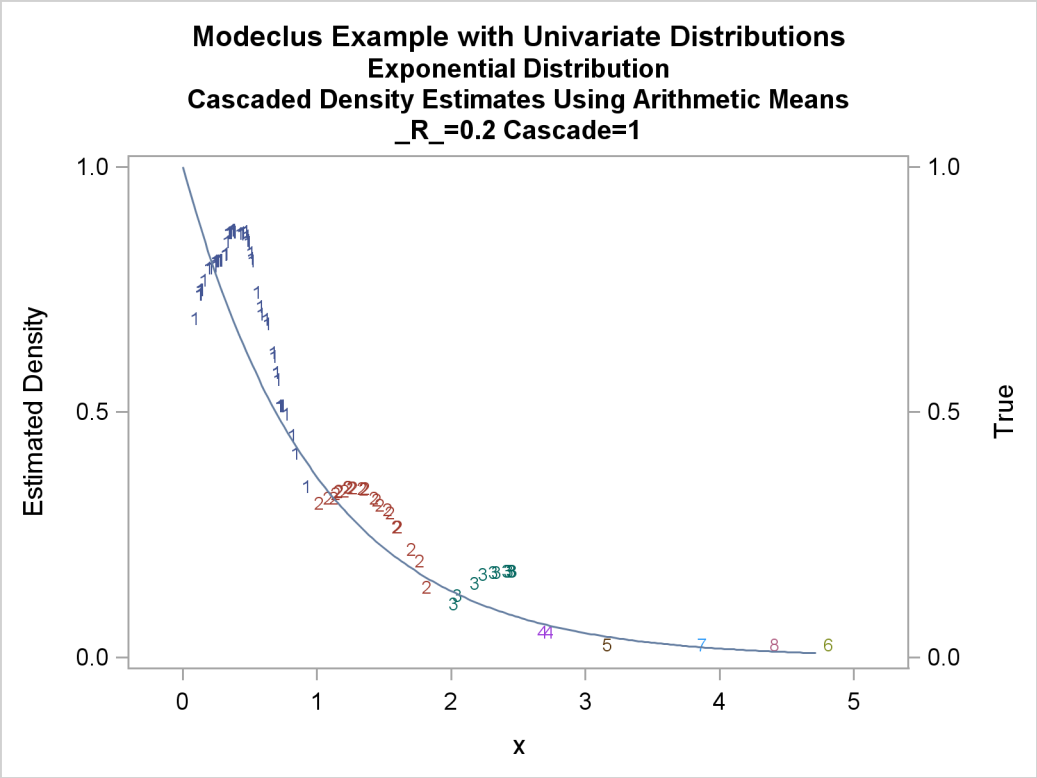
Modeclus Example with Univariate Distributions
Exponential Distribution
Cascaded Density Estimates Using Arithmetic Means

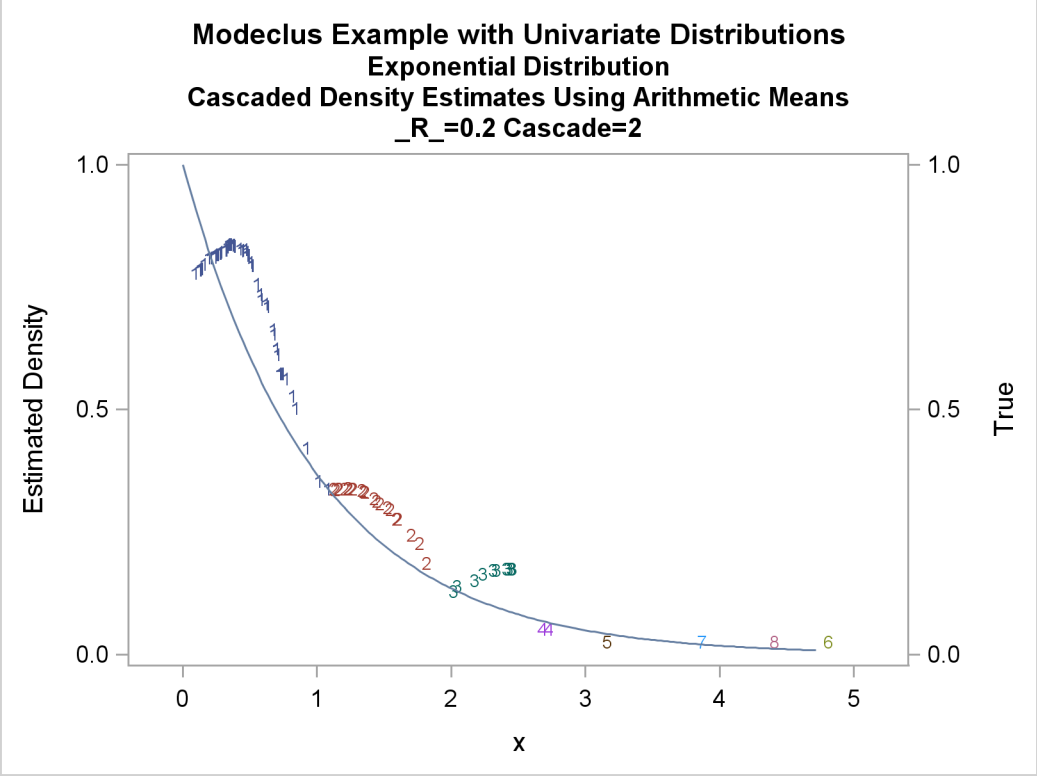
The MODECLUS Procedure

Cluster Summary

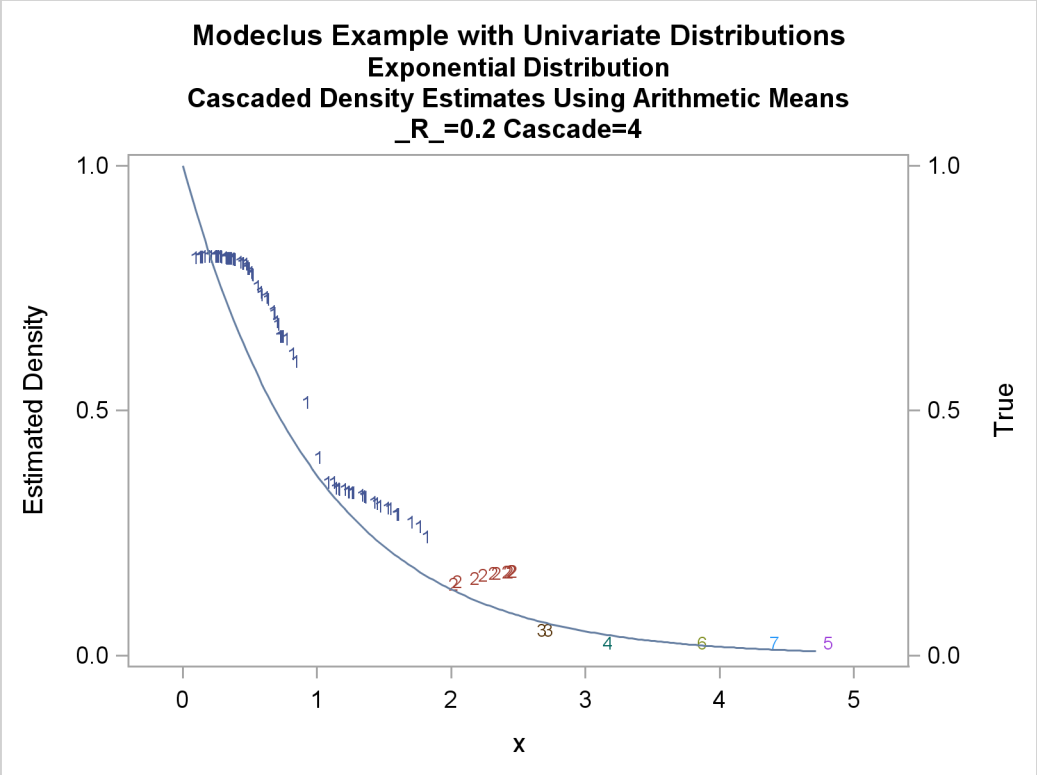
R	Cascade	Number of Clusters	Frequency of Unclassified Objects
0.2	1	8	0
0.2	2	8	0
0.2	4	7	0

Output 57.1.12 True Density, Estimated Density, and Cluster Membership by _R_=0.2 with Various _CASCAD_ Values





Output 57.1.12 *continued*



The following statements produce Output 57.1.13 through Output 57.1.18:

```
data normix;
   title2 'Normal Mixture Distribution';
   drop n sigma;
   sigma=.125;
   do n=1 to 100;
      x=rannor(456)*sigma+mod(n,2)/2;
      true=exp(-.5*(x/sigma)**2)+exp(-.5*((x-.5)/sigma)**2);
      true=.5*true/(sigma*sqrt(2*3.1415926536));
      output;
   end;

proc modeclus data=normix m=1 k=10 20 40 60 out=out short;
   var x;
run;
proc sgplot data=out noautolegend;
   y2axis label='True' values=(0 to 1.6 by .1);
   yaxis values=(0 to 3 by 0.5);
   scatter y=density x=x / markerchar=cluster group=cluster;
   pbspline y=true x=x / y2axis nomarkers lineattrs=(thickness= 1);
   by _K_;
run;

/*******************************************/
```

```
proc modeclus data=normix m=1 r=.05 .10 .20 .30 out=out short;
   var x;
run;
proc sgplot data=out noautolegend;
   y2axis label='True' values=(0 to 1.6 by .1);
   yaxis values=(0 to 3 by 0.5);
   scatter y=density x=x / markerchar=cluster group=cluster;
   pbspline y=true x=x / y2axis nomarkers lineattrs=(thickness= 1);
   by _R_;
run;

/*******************************************/

title3 'Cascaded Density Estimates Using Arithmetic Means';
proc modeclus data=normix m=1 r=.05 cascade=1 2 4 am out=out short;
   var x;
run;
proc sgplot data=out noautolegend;
   y2axis label='True' values=(0 to 1.6 by .1);
   yaxis values=(0 to 2 by 0.5);
   scatter y=density x=x / markerchar=cluster group=cluster;
   pbspline y=true x=x / y2axis nomarkers lineattrs=(thickness= 1);
   by _R_ _CASCAD_;
run;
```

Output 57.1.13 Cluster Analysis of Sample from a Bimodal Mixture of Two Normal Distributions

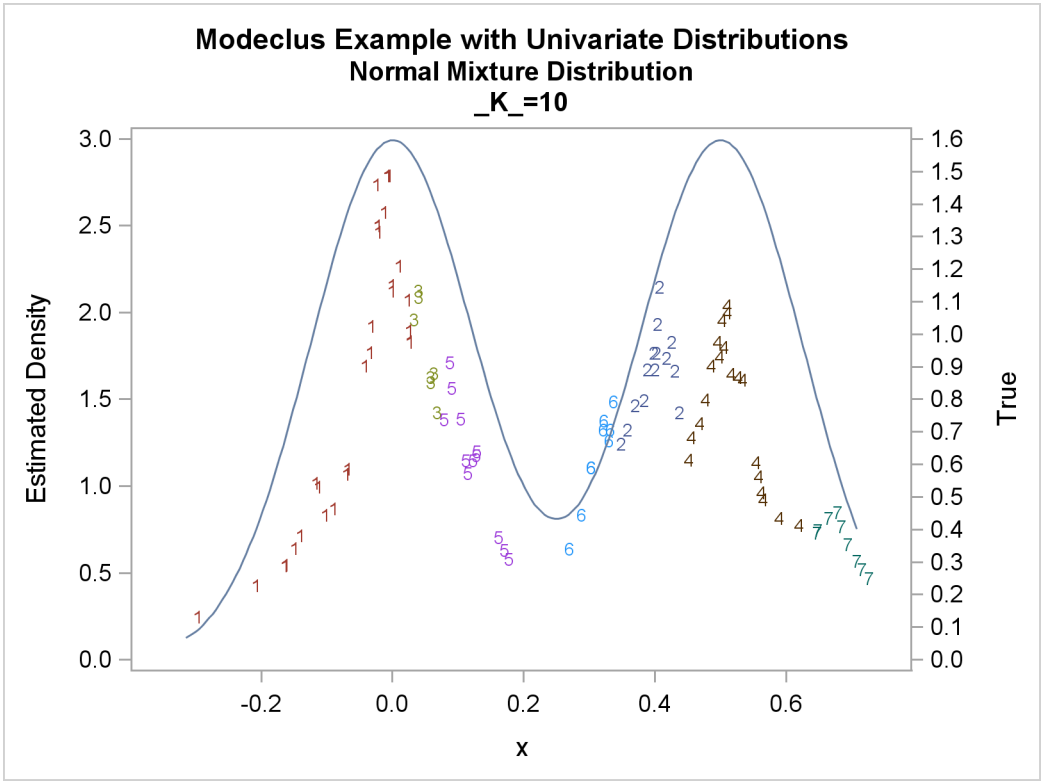
Modeclus Example with Univariate Distributions
Normal Mixture Distribution

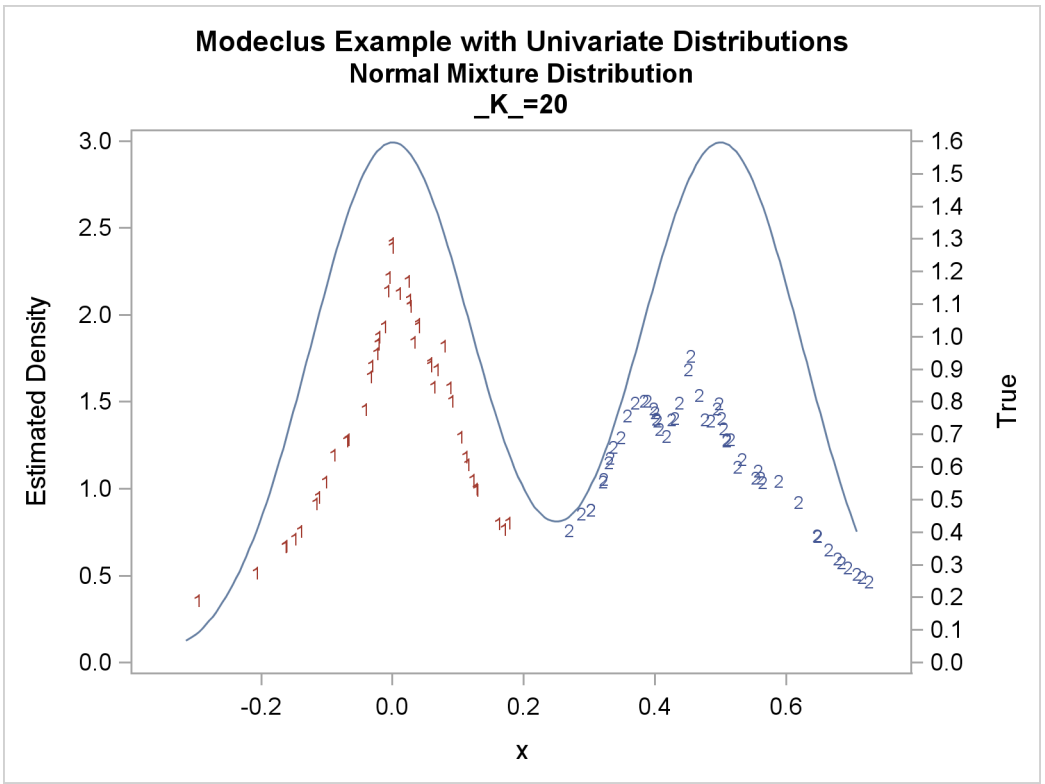
The MODECLUS Procedure

Cluster Summary

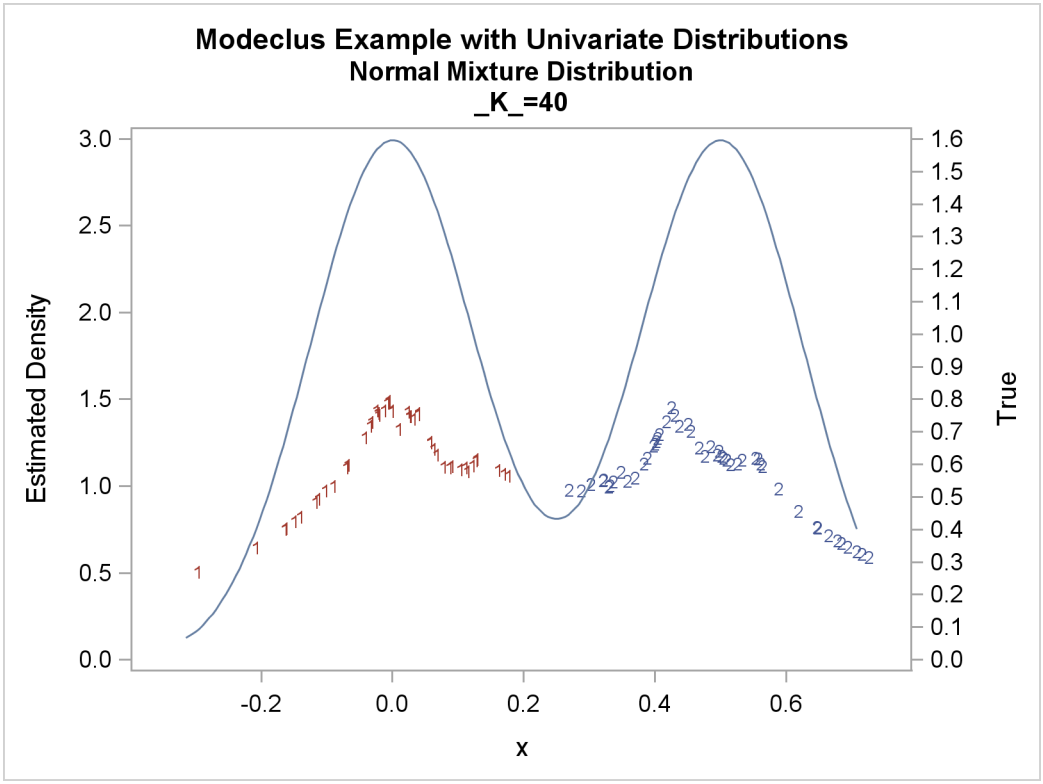
K	Number of Clusters	Frequency of Unclassified Objects
10	7	0
20	2	0
40	2	0
60	1	0

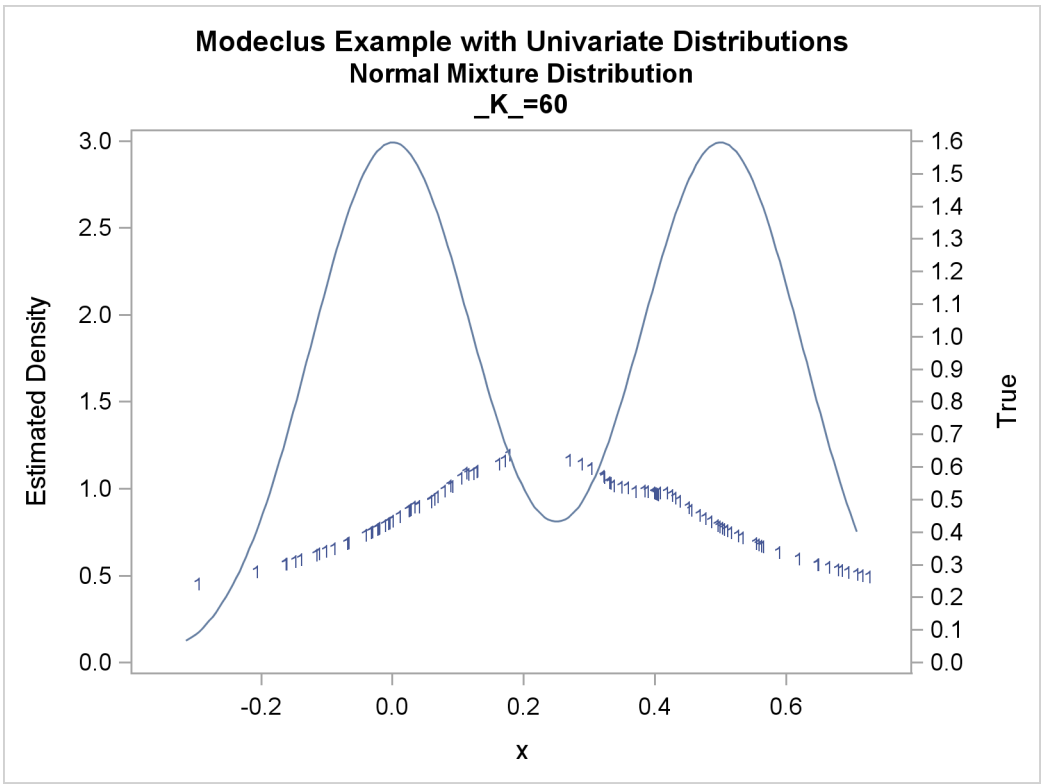
Output 57.1.14 True Density, Estimated Density, and Cluster Membership by Various _K_ Values





Output 57.1.14 *continued*





Output 57.1.15 Cluster Analysis of Sample from a Bimodal Mixture of Two Normal Distributions

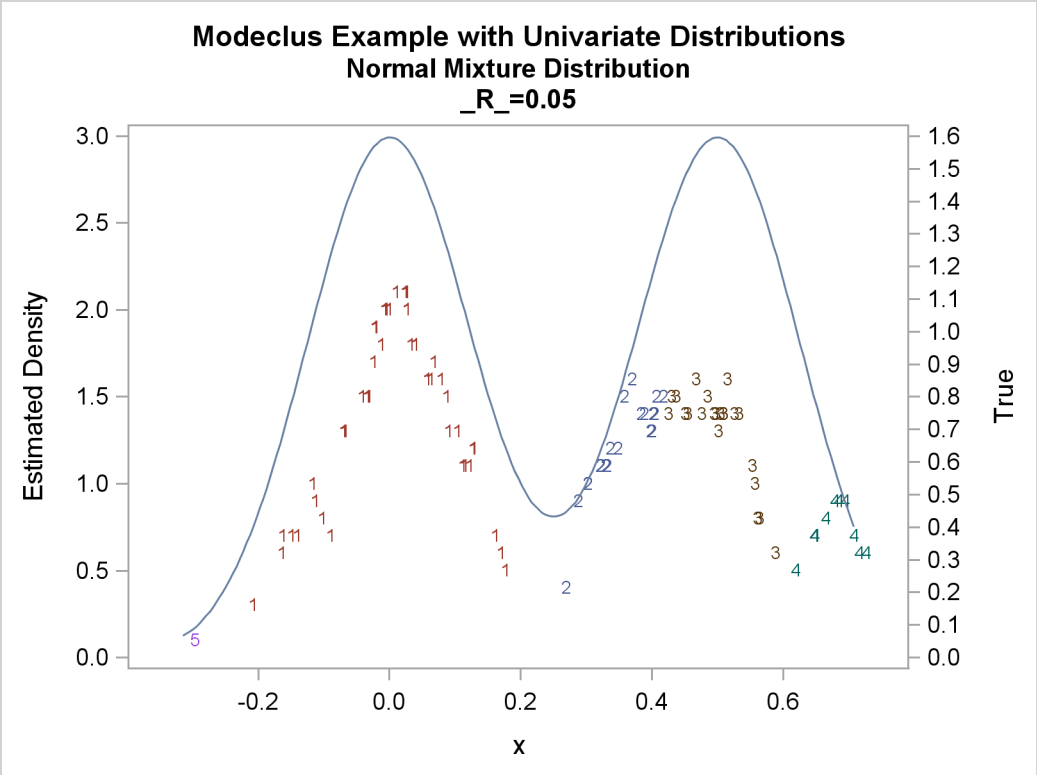
Modeclus Example with Univariate Distributions
Normal Mixture Distribution

The MODECLUS Procedure

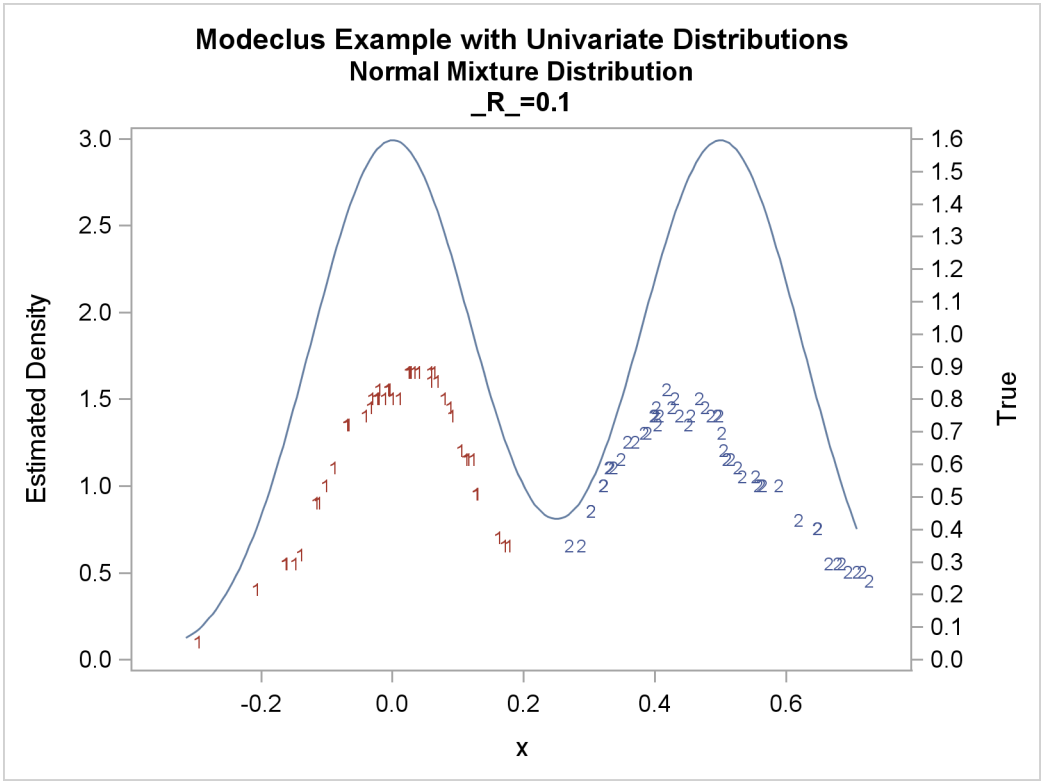
Cluster Summary

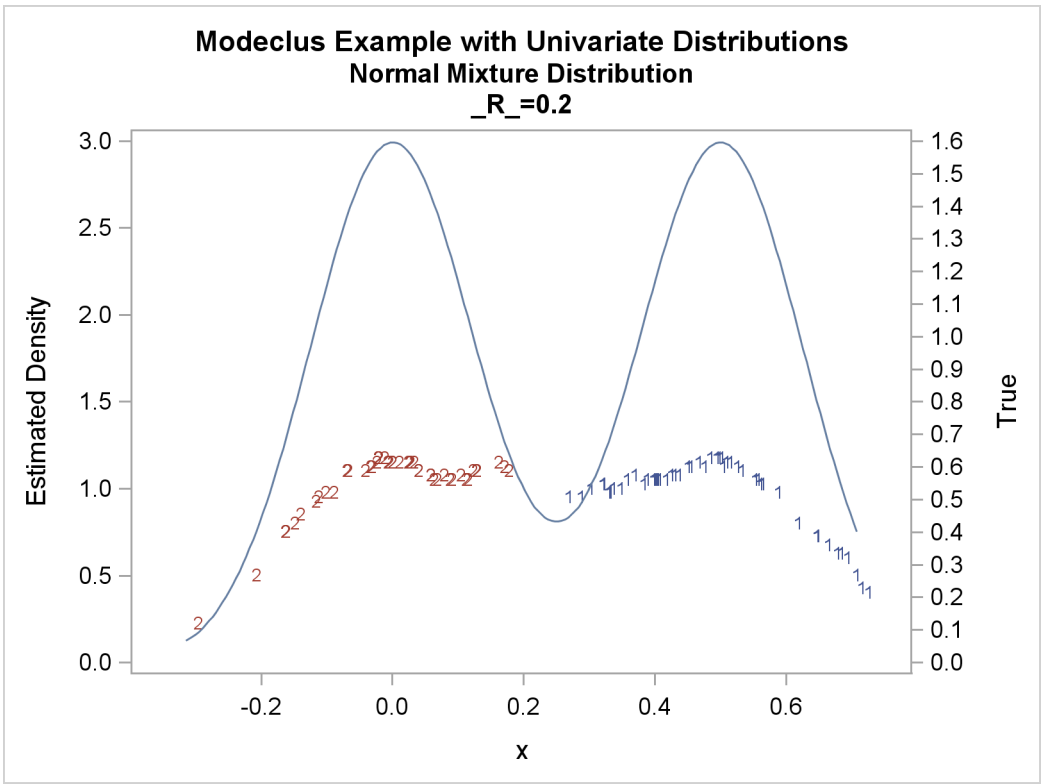
R	Number of Clusters	Frequency of Unclassified Objects
0.05	5	0
0.1	2	0
0.2	2	0
0.3	1	0

Output 57.1.16 True Density, Estimated Density, and Cluster Membership by Various _R_= Values

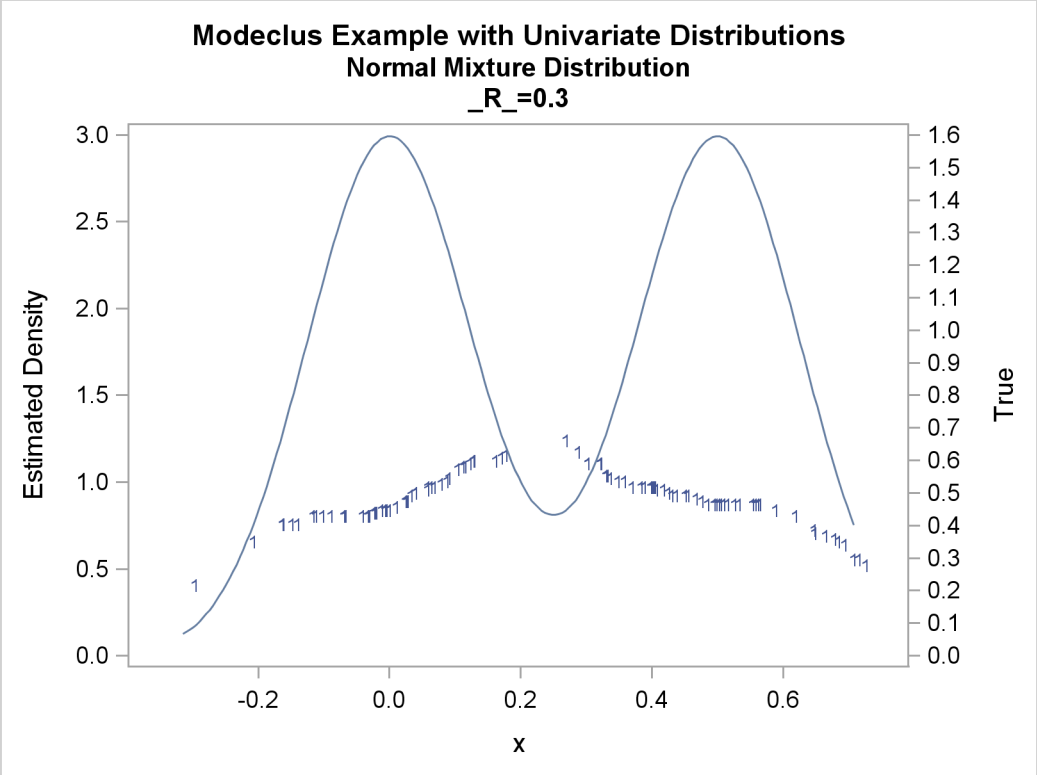


Output 57.1.16 *continued*





Output 57.1.16 *continued*



Output 57.1.17 Cluster Analysis of Sample from a Bimodal Mixture of Two Normal Distributions

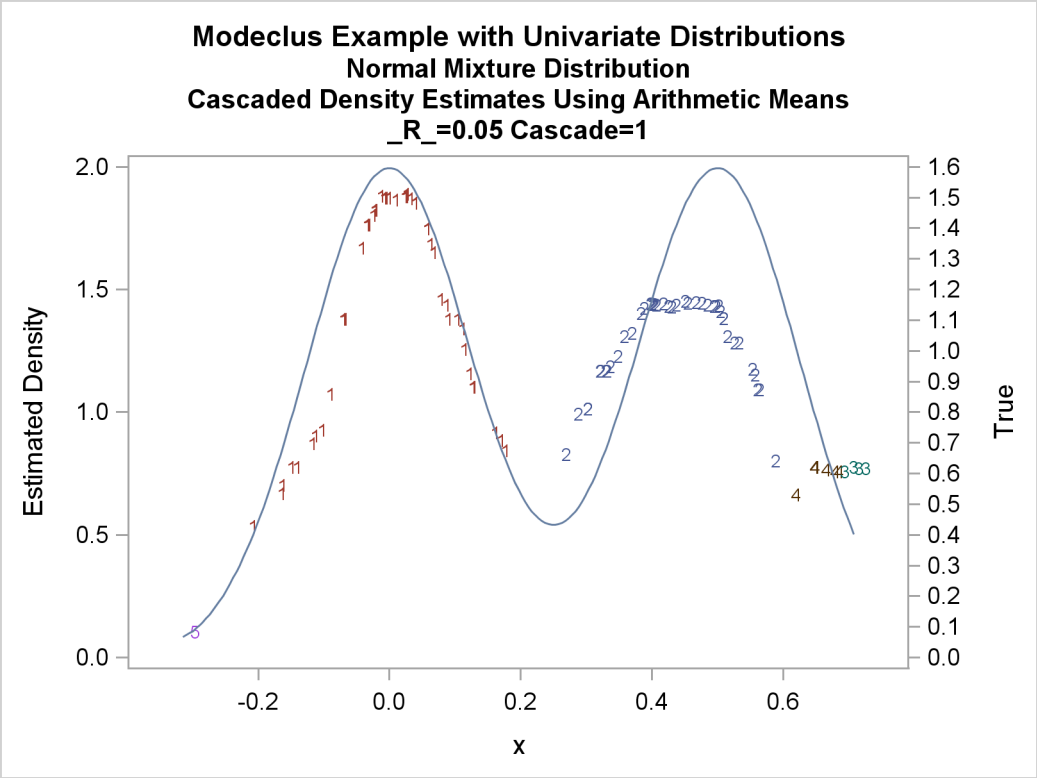
Modeclus Example with Univariate Distributions
Normal Mixture Distribution
Cascaded Density Estimates Using Arithmetic Means

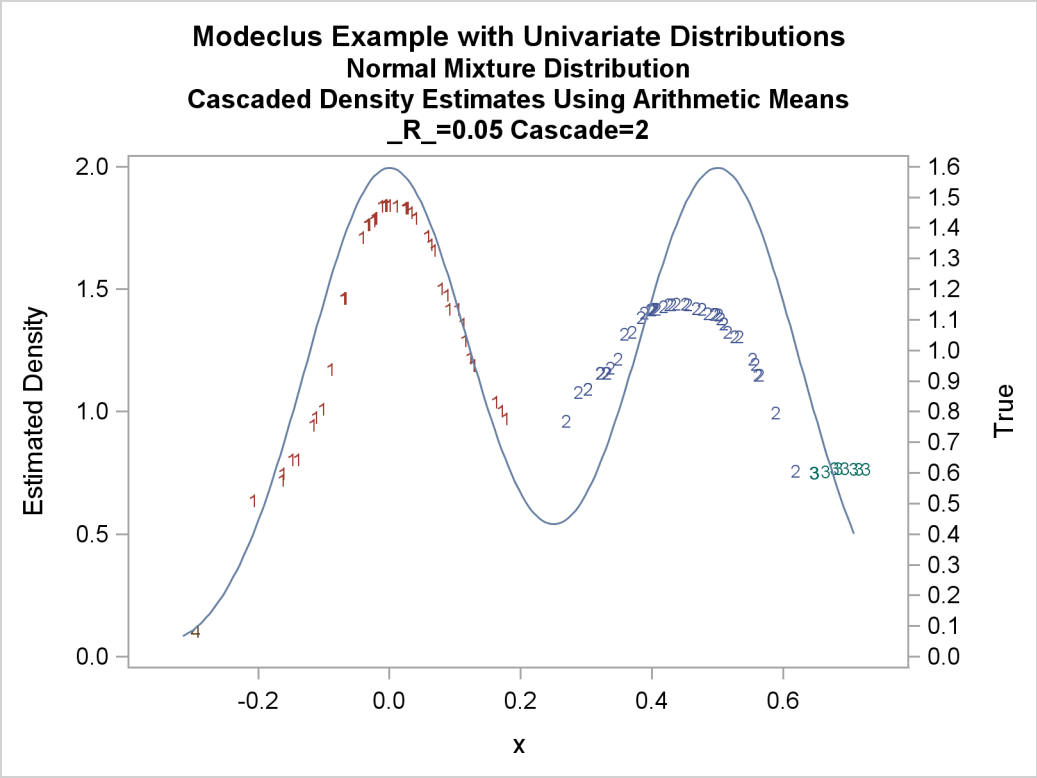
The MODECLUS Procedure

Cluster Summary

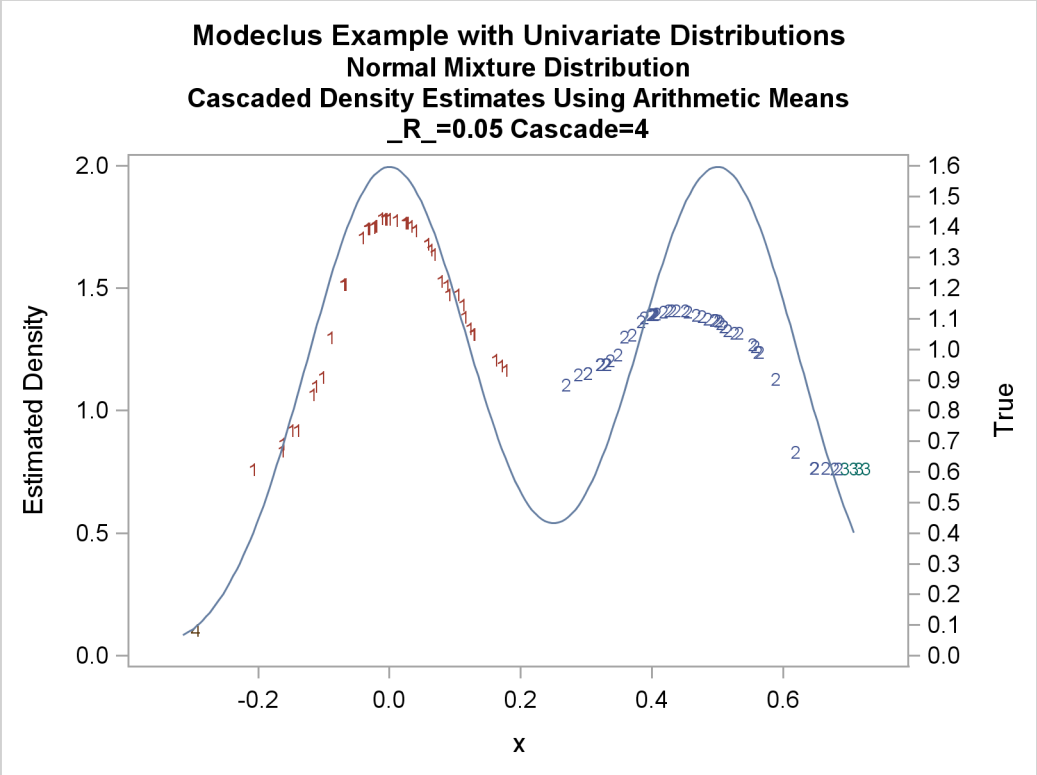
R	Cascade	Number of Clusters	Frequency of Unclassified Objects
0.05	1	5	0
0.05	2	4	0
0.05	4	4	0

Output 57.1.18 True Density, Estimated Density, and Cluster Membership by _R_=0.05 with Various _CASCAD_ Values





Output 57.1.18 *continued*



Example 57.2: Cluster Analysis of Flying Mileages between Ten American Cities

This example uses distance data and illustrates the use of the TRANSPOSE procedure and the DATA step to fill in the upper triangle of the distance matrix. The results are displayed in Output 57.2.1 through Output 57.2.3.

The following statements produce Output 57.2.1:

```
data mileages(type=distance);
   title 'Modeclus Analysis of 10 American Cities';
   title2 'Based on Flying Mileages';

   input (ATLANTA CHICAGO DENVER HOUSTON LOSANGELES
   MIAMI NEWYORK SANFRAN SEATTLE WASHDC) (5.)
   @53 CITY $15.;
   datalines;
   0                                                  ATLANTA
 587    0                                             CHICAGO
1212  920    0                                        DENVER
 701  940  879    0                                   HOUSTON
1936 1745  831 1374    0                             LOS ANGELES
 604 1188 1726  968 2339    0                        MIAMI
 748  713 1631 1420 2451 1092    0                   NEW YORK
2139 1858  949 1645  347 2594 2571    0              SAN FRANCISCO
2182 1737 1021 1891  959 2734 2408  678    0         SEATTLE
 543  597 1494 1220 2300  923  205 2442 2329    0   WASHINGTON D.C.
;

*-----Fill in Upper Triangle of Distance Matrix---------------;
proc transpose out=tran;
   copy CITY;
run;

data mileages(type=distance);
   merge mileages tran;
   array var ATLANTA--WASHDC;
   array col col1-col10;
   drop col1-col10 _name_;
   do over var;
      var=sum(var,col);
   end;

*-----Clustering with K-Nearest-Neighbor Density Estimates-----;
proc modeclus data=mileages all m=1 k=3;
   id CITY;
run;
```

Output 57.2.1 Clustering with K-Nearest-Neighbor Density Estimates

Modeclus Analysis of 10 American Cities
Based on Flying Mileages

The MODECLUS Procedure

Nearest Neighbor List

CITY	Neighbor	Distance
ATLANTA	WASHINGTON D.C.	543.0000000
	CHICAGO	587.0000000
CHICAGO	ATLANTA	587.0000000
	WASHINGTON D.C.	597.0000000
DENVER	LOS ANGELES	831.0000000
	HOUSTON	879.0000000
HOUSTON	ATLANTA	701.0000000
	DENVER	879.0000000
LOS ANGELES	SAN FRANCISCO	347.0000000
	DENVER	831.0000000
MIAMI	ATLANTA	604.0000000
	WASHINGTON D.C.	923.0000000
NEW YORK	WASHINGTON D.C.	205.0000000
	CHICAGO	713.0000000
SAN FRANCISCO	LOS ANGELES	347.0000000
	SEATTLE	678.0000000
SEATTLE	SAN FRANCISCO	678.0000000
	LOS ANGELES	959.0000000
WASHINGTON D.C.	NEW YORK	205.0000000
	ATLANTA	543.0000000

Output 57.2.1 *continued*

Modeclus Analysis of 10 American Cities
Based on Flying Mileages

The MODECLUS Procedure
K=3 METHOD=1

Sums of Density Estimates Within Neighborhood

Cluster	CITY	Estimated Density	Same Cluster	Other Clusters	Total
1	ATLANTA	0.00025554	0.0005275	0	0.0005275
	CHICAGO	0.00025126	0.00053178	0	0.00053178
	HOUSTON	0.00017065	0.00025554	0.00017065	0.00042619
	MIAMI	0.00016251	0.00053178	0	0.00053178
	NEW YORK	0.00021038	0.0005275	0	0.0005275
	WASHINGTON D.C.	0.00027624	0.00046592	0	0.00046592
2	DENVER	0.00017065	0.00018051	0.00017065	0.00035115
	LOS ANGELES	0.00018051	0.00039189	0	0.00039189
	SAN FRANCISCO	0.00022124	0.00033692	0	0.00033692
	SEATTLE	0.00015641	0.00040174	0	0.00040174

Sums of Density Estimates
Within Neighborhood

Cluster	CITY	Cluster Proportion Same/Total
1	ATLANTA	1.000
	CHICAGO	1.000
	HOUSTON	0.600
	MIAMI	1.000
	NEW YORK	1.000
	WASHINGTON D.C.	1.000
2	DENVER	0.514
	LOS ANGELES	1.000
	SAN FRANCISCO	1.000
	SEATTLE	1.000

Boundary Objects			-Cluster Proportions-	
CITY	Density	Cluster	1	2
DENVER	0.0001706485	2	0.486	0.514
HOUSTON	0.0001706485	1	0.600	0.400

Cluster Statistics

Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density
1	6	0.00027624	1	0.00017065
2	4	0.00022124	1	0.00017065

Output 57.2.1 *continued*

Modeclus Analysis of 10 American Cities
Based on Flying Mileages

The MODECLUS Procedure

Cluster Summary

K	Number of Clusters	Frequency of Unclassified Objects
3	2	0

The following statements produce Output 57.2.2:

```
*------Clustering with Uniform-Kernel Density Estimates--------;
proc modeclus data=mileages all m=1 r=600 800;
   id CITY;
run;
```

Output 57.2.2 Clustering with Uniform-Kernel Density Estimates

Modeclus Analysis of 10 American Cities
Based on Flying Mileages

The MODECLUS Procedure

Nearest Neighbor List

CITY	Neighbor	Distance
ATLANTA	WASHINGTON D.C.	543.0000000
	CHICAGO	587.0000000
	MIAMI	604.0000000
	HOUSTON	701.0000000
	NEW YORK	748.0000000
CHICAGO	ATLANTA	587.0000000
	WASHINGTON D.C.	597.0000000
	NEW YORK	713.0000000
HOUSTON	ATLANTA	701.0000000
LOS ANGELES	SAN FRANCISCO	347.0000000
MIAMI	ATLANTA	604.0000000
NEW YORK	WASHINGTON D.C.	205.0000000
	CHICAGO	713.0000000
	ATLANTA	748.0000000
SAN FRANCISCO	LOS ANGELES	347.0000000
	SEATTLE	678.0000000
SEATTLE	SAN FRANCISCO	678.0000000
WASHINGTON D.C.	NEW YORK	205.0000000
	ATLANTA	543.0000000
	CHICAGO	597.0000000

Output 57.2.2 *continued*

Modeclus Analysis of 10 American Cities
Based on Flying Mileages

The MODECLUS Procedure
R=600 METHOD=1

Sums of Density Estimates Within Neighborhood

Cluster	CITY	Estimated Density	Same Cluster	Other Clusters	Total
1	ATLANTA	0.00025	0.00058333	0	0.00058333
	CHICAGO	0.00025	0.00058333	0	0.00058333
	NEW YORK	0.00016667	0.00033333	0	0.00033333
	WASHINGTON D.C.	0.00033333	0.00066667	0	0.00066667
2	LOS ANGELES	0.00016667	0.00016667	0	0.00016667
	SAN FRANCISCO	0.00016667	0.00016667	0	0.00016667
3	DENVER	0.00008333	0	0	0
4	HOUSTON	0.00008333	0	0	0
5	MIAMI	0.00008333	0	0	0
6	SEATTLE	0.00008333	0	0	0

Sums of Density Estimates
Within Neighborhood

Cluster	CITY	Cluster Proportion Same/Total
1	ATLANTA	1.000
	CHICAGO	1.000
	NEW YORK	1.000
	WASHINGTON D.C.	1.000
2	LOS ANGELES	1.000
	SAN FRANCISCO	1.000
3	DENVER	.
4	HOUSTON	.
5	MIAMI	.
6	SEATTLE	.

Output 57.2.2 *continued*

No Boundary Objects

Cluster Statistics

Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density
1	4	0.00033333	0	.
2	2	0.00016667	0	.
3	1	0.00008333	0	.
4	1	0.00008333	0	.
5	1	0.00008333	0	.
6	1	0.00008333	0	.

Modeclus Analysis of 10 American Cities
Based on Flying Mileages

The MODECLUS Procedure
R=800 METHOD=1

Sums of Density Estimates Within Neighborhood

Cluster	CITY	Estimated Density	Same Cluster	Other Clusters	Total
1	ATLANTA	0.000375	0.001	0	0.001
	CHICAGO	0.00025	0.000875	0	0.000875
	HOUSTON	0.000125	0.000375	0	0.000375
	MIAMI	0.000125	0.000375	0	0.000375
	NEW YORK	0.00025	0.000875	0	0.000875
	WASHINGTON D.C.	0.00025	0.000875	0	0.000875
2	LOS ANGELES	0.000125	0.0001875	0	0.0001875
	SAN FRANCISCO	0.0001875	0.00025	0	0.00025
	SEATTLE	0.000125	0.0001875	0	0.0001875
3	DENVER	0.0000625	0	0	0

Sums of Density Estimates
Within Neighborhood

Cluster	CITY	Cluster Proportion Same/Total
1	ATLANTA	1.000
	CHICAGO	1.000
	HOUSTON	1.000
	MIAMI	1.000
	NEW YORK	1.000
	WASHINGTON D.C.	1.000
2	LOS ANGELES	1.000
	SAN FRANCISCO	1.000
	SEATTLE	1.000
3	DENVER	.

Output 57.2.2 *continued*

No Boundary Objects

Cluster Statistics

Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density
1	6	0.000375	0	.
2	3	0.0001875	0	.
3	1	0.0000625	0	.

Modeclus Analysis of 10 American Cities
Based on Flying Mileages

The MODECLUS Procedure

Cluster Summary

R	Number of Clusters	Frequency of Unclassified Objects
600	6	0
800	3	0

The following statements produce Output 57.2.3:

```
*------Clustering Neighborhoods Extended to Nearest Neighbor--------;
proc modeclus data=mileages list m=1 ck=2 r=600 800;
   id CITY;
run;
```

Output 57.2.3 Uniform-Kernel Density Estimates, Clustering Neighborhoods Extended to Nearest Neighbor

Modeclus Analysis of 10 American Cities
Based on Flying Mileages

The MODECLUS Procedure
CK=2 R=600 METHOD=1

Sums of Density Estimates Within Neighborhood

Cluster	CITY	Estimated Density	Same Cluster	Other Clusters	Total
1	ATLANTA	0.00025	0.00058333	0	0.00058333
	CHICAGO	0.00025	0.00058333	0	0.00058333
	HOUSTON	0.00008333	0.00025	0	0.00025
	MIAMI	0.00008333	0.00025	0	0.00025
	NEW YORK	0.00016667	0.00033333	0	0.00033333
	WASHINGTON D.C.	0.00033333	0.00066667	0	0.00066667
2	DENVER	0.00008333	0.00016667	0	0.00016667
	LOS ANGELES	0.00016667	0.00016667	0	0.00016667
	SAN FRANCISCO	0.00016667	0.00016667	0	0.00016667
	SEATTLE	0.00008333	0.00016667	0	0.00016667

Sums of Density Estimates Within Neighborhood

Cluster	CITY	Cluster Proportion Same/Total
1	ATLANTA	1.000
	CHICAGO	1.000
	HOUSTON	1.000
	MIAMI	1.000
	NEW YORK	1.000
	WASHINGTON D.C.	1.000
2	DENVER	1.000
	LOS ANGELES	1.000
	SAN FRANCISCO	1.000
	SEATTLE	1.000

Cluster Statistics

Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density
1	6	0.00033333	0	.
2	4	0.00016667	0	.

Output 57.2.3 *continued*

Modeclus Analysis of 10 American Cities
Based on Flying Mileages

The MODECLUS Procedure
CK=2 R=800 METHOD=1

Sums of Density Estimates Within Neighborhood

Cluster	CITY	Estimated Density	Same Cluster	Other Clusters	Total
1	ATLANTA	0.000375	0.001	0	0.001
	CHICAGO	0.00025	0.000875	0	0.000875
	HOUSTON	0.000125	0.000375	0	0.000375
	MIAMI	0.000125	0.000375	0	0.000375
	NEW YORK	0.00025	0.000875	0	0.000875
	WASHINGTON D.C.	0.00025	0.000875	0	0.000875
2	DENVER	0.0000625	0.000125	0	0.000125
	LOS ANGELES	0.000125	0.0001875	0	0.0001875
	SAN FRANCISCO	0.0001875	0.00025	0	0.00025
	SEATTLE	0.000125	0.0001875	0	0.0001875

Sums of Density Estimates
Within Neighborhood

Cluster	CITY	Cluster Proportion Same/Total
1	ATLANTA	1.000
	CHICAGO	1.000
	HOUSTON	1.000
	MIAMI	1.000
	NEW YORK	1.000
	WASHINGTON D.C.	1.000
2	DENVER	1.000
	LOS ANGELES	1.000
	SAN FRANCISCO	1.000
	SEATTLE	1.000

Cluster Statistics

Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density
1	6	0.000375	0	.
2	4	0.0001875	0	.

Output 57.2.3 *continued*

```
                    Modeclus Analysis of 10 American Cities
                            Based on Flying Mileages

                             The MODECLUS Procedure

                                Cluster Summary
                                                    Frequency of
                                       Number of    Unclassified
                      R         CK     Clusters         Objects
                ----------------------------------------------
                    600          2            2               0
                    800          2            2               0
```

Example 57.3: Cluster Analysis with Significance Tests

This example uses artificial data containing two clusters. One cluster is from a circular bivariate normal distribution. The other is a ring-shaped cluster that completely surrounds the first cluster. Without significance tests, the ring is divided into several sample clusters for any degree of smoothing that yields reasonable density estimates. The JOIN= option puts the ring back together. Output 57.3.1 displays a short summary generated from the first PROC MODECLUS statement. Output 57.3.2 contains a series of tables produced from the second PROC MODECLUS statement. The lack of p-value in the JOIN= option makes joining continue until only one cluster remains (see the description of the JOIN= option on page 4105). The cluster memberships are then plotted as displayed in Output 57.3.1 through Output 57.3.8.

The following statements produce Output 57.3.1 through Output 57.3.8:

```
   title  'Modeclus Analysis with the JOIN= option';
   title2 'A Normal Cluster Surrounded by a Ring Cluster';

data circle; keep x y;
   c=1;
   do n=1 to 30;
      x=rannor(5);
      y=rannor(5);
      output;
   end;

   c=2;
   do n=1 to 300;
      x=rannor(5);
      y=rannor(5);
      z=rannor(5)+8;
      l=z/sqrt(x**2+y**2);
      x=x*l;
      y=y*l;
      output;
   end;

proc modeclus data=circle m=1 r=1 to 3.5 by .25 join=20 short; run;
```

```
proc modeclus data=circle m=1 r=2.5 join out=out; run;

proc sgplot data=out noautolegend;
   yaxis values=(-10 to 10 by 5);
   xaxis values=(-15 to 15 by 5);
   scatter y=y x=x / group=cluster Markerchar=cluster;
   by _NJOIN_;
run;
```

Output 57.3.1 Significance Tests with the JOIN=20 and SHORT Options

Modeclus Analysis with the JOIN= option
A Normal Cluster Surrounded by a Ring Cluster

The MODECLUS Procedure

Cluster Summary

R	Number of Clusters Joined	Maximum P-value	Number of Clusters	Frequency of Unclassified Objects
1	36	0.9339	1	301
1.25	20	0.7131	1	301
1.5	10	0.3296	1	300
1.75	5	0.1990	2	0
2	5	0.0683	2	0
2.25	3	0.0504	2	0
2.5	4	0.0301	2	0
2.75	3	0.0585	2	0
3	5	0.0003	1	0
3.25	4	0.1923	2	0
3.5	4	0.0000	1	0

Output 57.3.2 Significance Tests with the JOIN Option

Modeclus Analysis with the JOIN= option
A Normal Cluster Surrounded by a Ring Cluster

The MODECLUS Procedure
R=2.5 METHOD=1

Cluster Statistics

Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density
1	103	0.00617328	22	0.00308664
2	71	0.00571029	20	0.0043213
3	53	0.00509296	18	0.00401263
4	45	0.00478429	19	0.00354964
5	30	0.00462996	0	.
6	28	0.00370397	17	0.00354964

Saddle Test: Version 92.7

Cluster	Mode Count	Saddle Count	Overlap Count	Z	Approx P-value
1	39	19	0	2.495	0.5055
2	36	27	9	1.193	0.999
3	32	25	10	0.986	0.9999
4	30	22	14	1.429	0.9924
5	29	0	.	3.611	0.0301
6	23	22	9	0.000	1

Cluster 6 with P-value 1.0000 will be joined to cluster 4.

Cluster Statistics

Cluster	Frequency	Maximum Estimated Density	Boundary Frequency	Estimated Saddle Density
1	103	0.00617328	22	0.00308664
2	71	0.00571029	20	0.0043213
3	53	0.00509296	18	0.00401263
4	73	0.00478429	13	0.00293231
5	30	0.00462996	0	.

Saddle Test: Version 92.7

Cluster	Mode Count	Saddle Count	Overlap Count	Z	Approx P-value
1	39	19	0	2.495	0.5055
2	36	27	9	1.193	0.999
3	32	25	10	0.986	0.9999
4	30	18	0	1.588	0.9778
5	29	0	.	3.611	0.0301

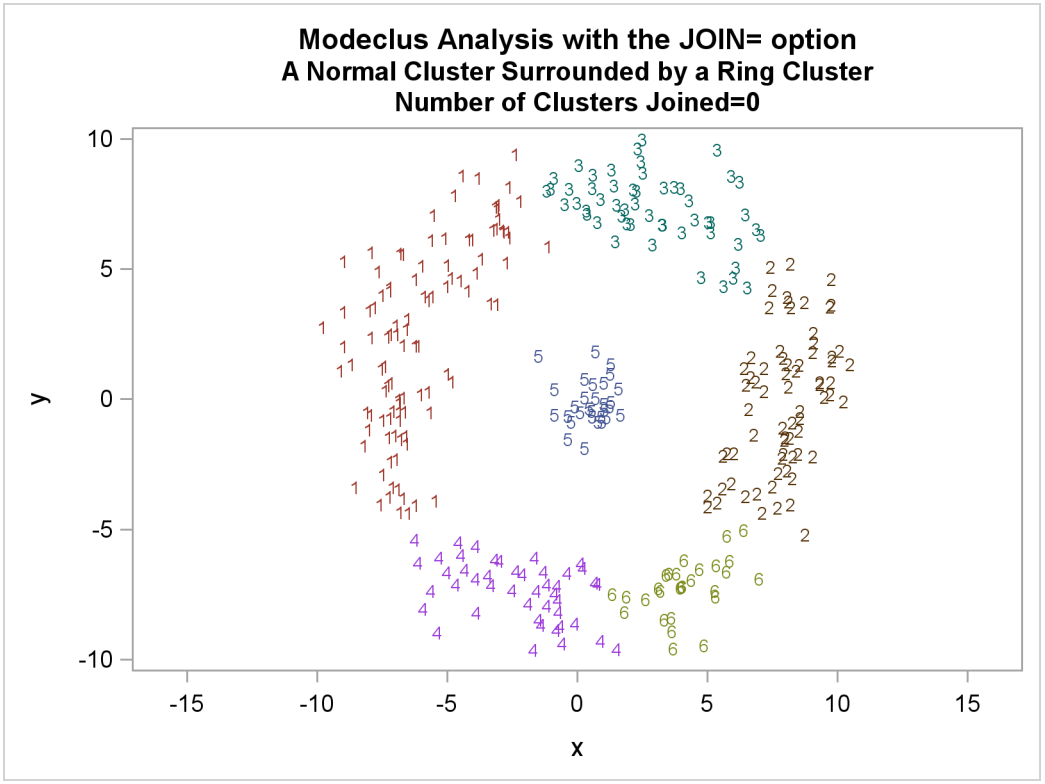
Output 57.3.2 *continued*

```
   Cluster 3 with P-value 0.9999 will be joined to cluster 1.


                    Cluster Statistics
                         Maximum                    Estimated
                       Estimated      Boundary        Saddle
Cluster     Frequency    Density     Frequency       Density
-------------------------------------------------------------
1                 156 0.00617328            17    0.00246931
2                  71 0.00571029            20     0.0043213
3                  73 0.00478429            13    0.00293231
4                  30 0.00462996             0             .

            -------------Saddle Test: Version 92.7------------
              Mode    Saddle    Overlap                  Approx
 Cluster     Count     Count      Count           Z     P-value
 ------------------------------------------------------------
 1              39        15          0       3.130      0.1318
 2              36        27          9       1.193       0.999
 3              30        18          0       1.588      0.9778
 4              29         0          .       3.611      0.0301

   Cluster 2 with P-value 0.9990 will be joined to cluster 3.


                    Cluster Statistics
                         Maximum                    Estimated
                       Estimated      Boundary        Saddle
Cluster     Frequency    Density     Frequency       Density
-------------------------------------------------------------
1                 156 0.00617328            17    0.00246931
2                 144 0.00571029            14    0.00293231
3                  30 0.00462996             0             .

            -------------Saddle Test: Version 92.7------------
              Mode    Saddle    Overlap                  Approx
 Cluster     Count     Count      Count           Z     P-value
 ------------------------------------------------------------
 1              39        15          0       3.130      0.1318
 2              36        18          0       2.313      0.6447
 3              29         0          .       3.611      0.0301
```

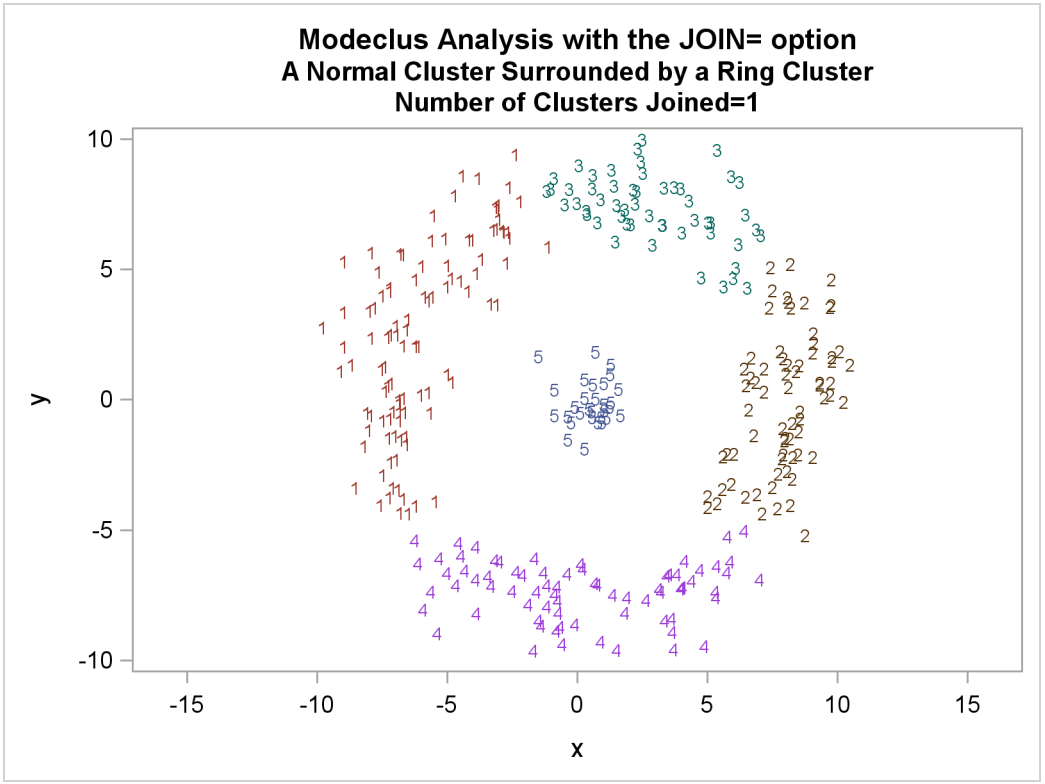
Output 57.3.2 *continued*

```
   Cluster 2 with P-value 0.6447 will be joined to cluster 1.


                    Cluster Statistics
                          Maximum                    Estimated
                        Estimated      Boundary        Saddle
Cluster     Frequency     Density     Frequency       Density
-------------------------------------------------------------
1                 300   0.00617328             0             .
2                  30   0.00462996             0             .

            -------------Saddle Test: Version 92.7------------
              Mode     Saddle     Overlap                 Approx
 Cluster     Count      Count       Count           Z     P-value
 ------------------------------------------------------------
 1              39          0           .       4.246      0.0026
 2              29          0           .       3.611      0.0301

        Cluster 2 with P-value 0.0301 will be dissolved.


                    Cluster Statistics
                          Maximum                    Estimated
                        Estimated      Boundary        Saddle
Cluster     Frequency     Density     Frequency       Density
-------------------------------------------------------------
1                 300   0.00617328             0             .

            -------------Saddle Test: Version 92.7------------
              Mode     Saddle     Overlap                 Approx
 Cluster     Count      Count       Count           Z     P-value
 ------------------------------------------------------------
 1              39          0           .       4.246      0.0026

               30 observations were unassigned.

        Cluster 1 with P-value 0.0026 will be dissolved.

           Modeclus Analysis with the JOIN= option
         A Normal Cluster Surrounded by a Ring Cluster

                    The MODECLUS Procedure

                       Cluster Summary
             Number of                              Frequency of
              Clusters     Maximum     Number of     Unclassified
       R       Joined     P-value      Clusters          Objects
  ----------------------------------------------------------------
     2.5            0      1.0000             6                0
     2.5            1      0.9999             5                0
     2.5            2      0.9990             4                0
     2.5            3      0.6447             3                0
     2.5            4      0.0301             2                0
     2.5            5      0.0026             1               30
```

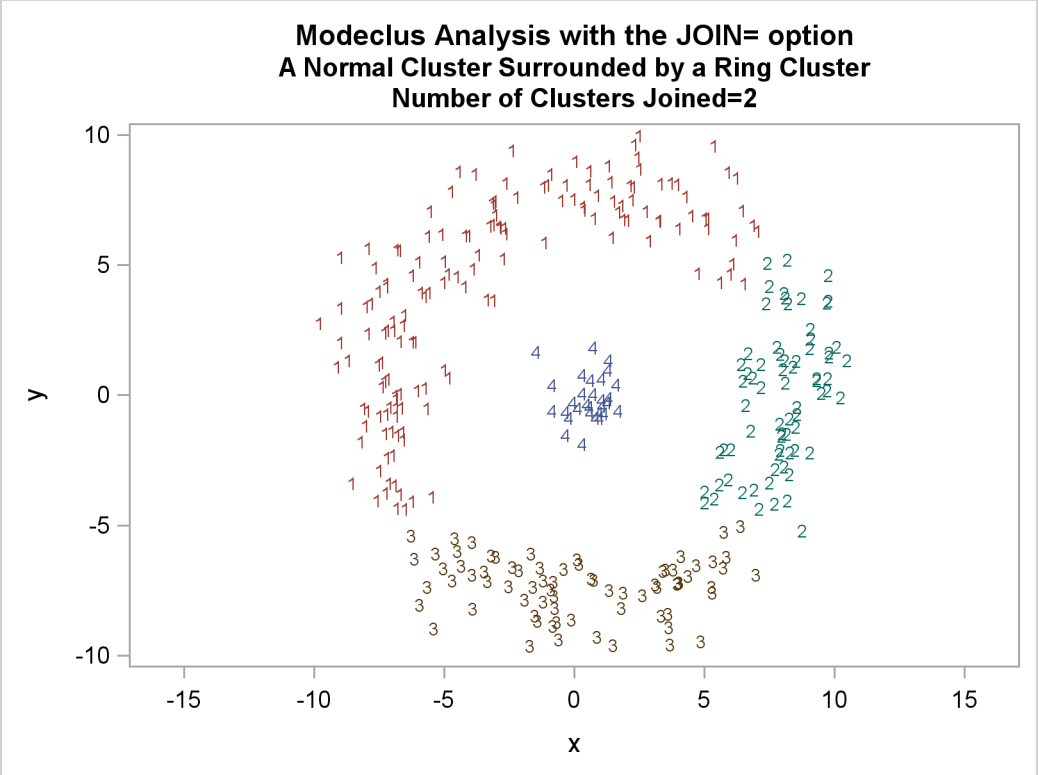
Output 57.3.3 Cluster Memberships When Number of Clusters Joined=0



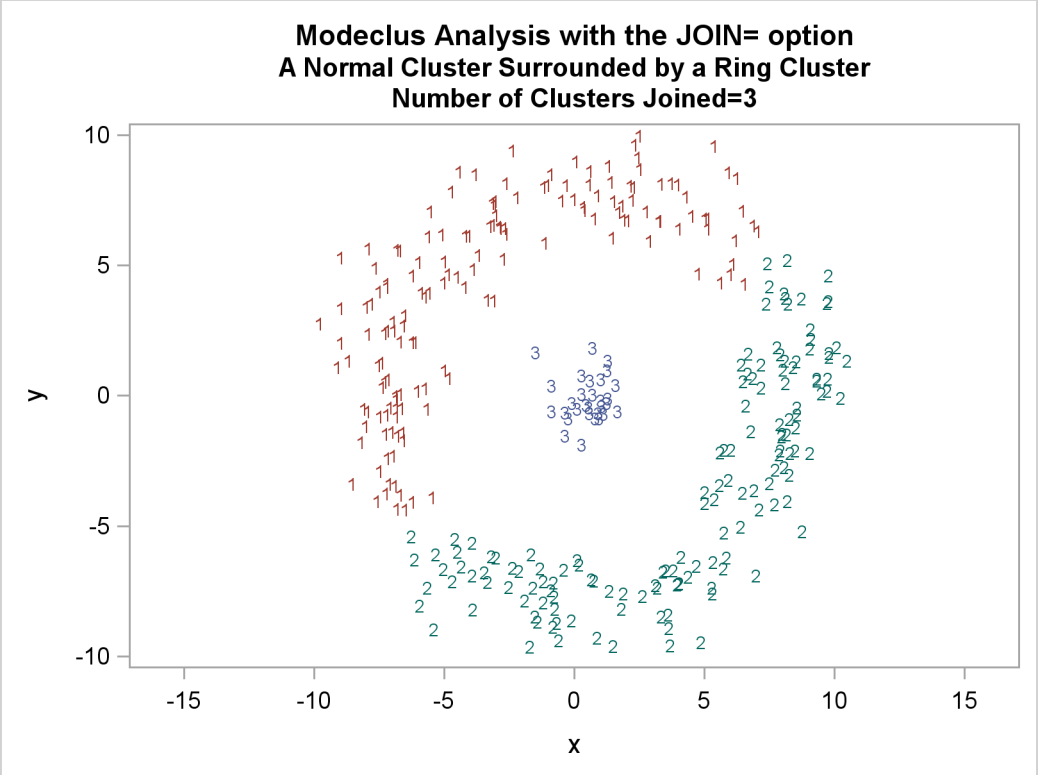
Output 57.3.4 Cluster Memberships When Number of Clusters Joined=1



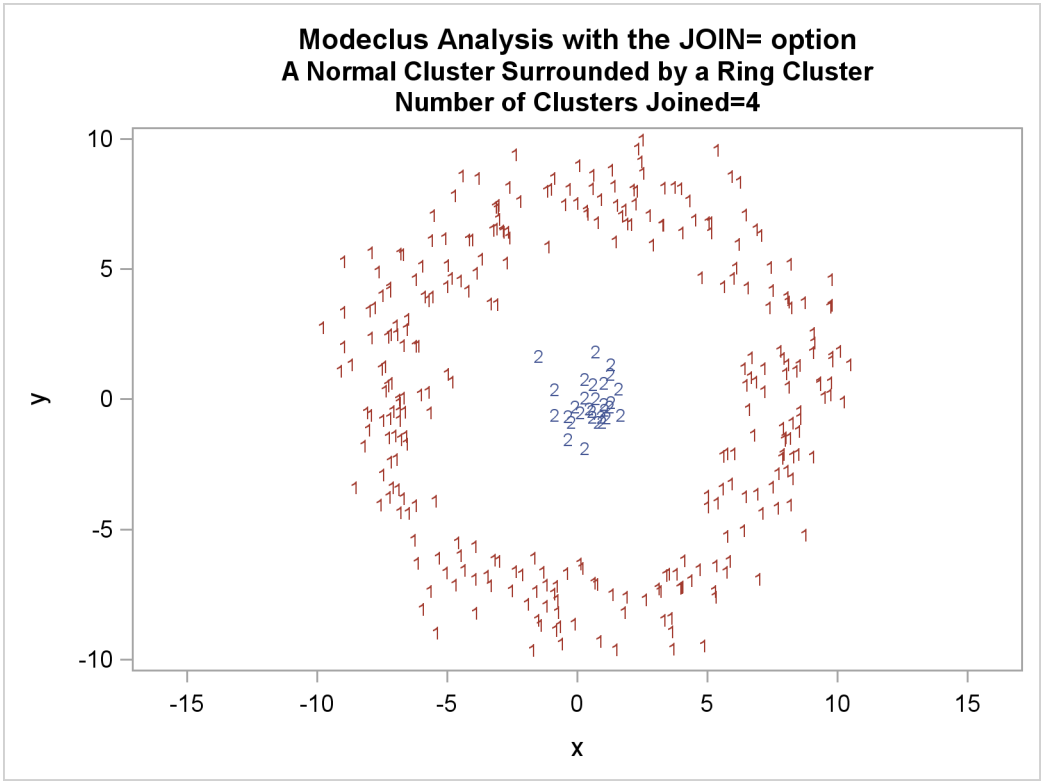
Output 57.3.5 Cluster Memberships When Number of Clusters Joined=2



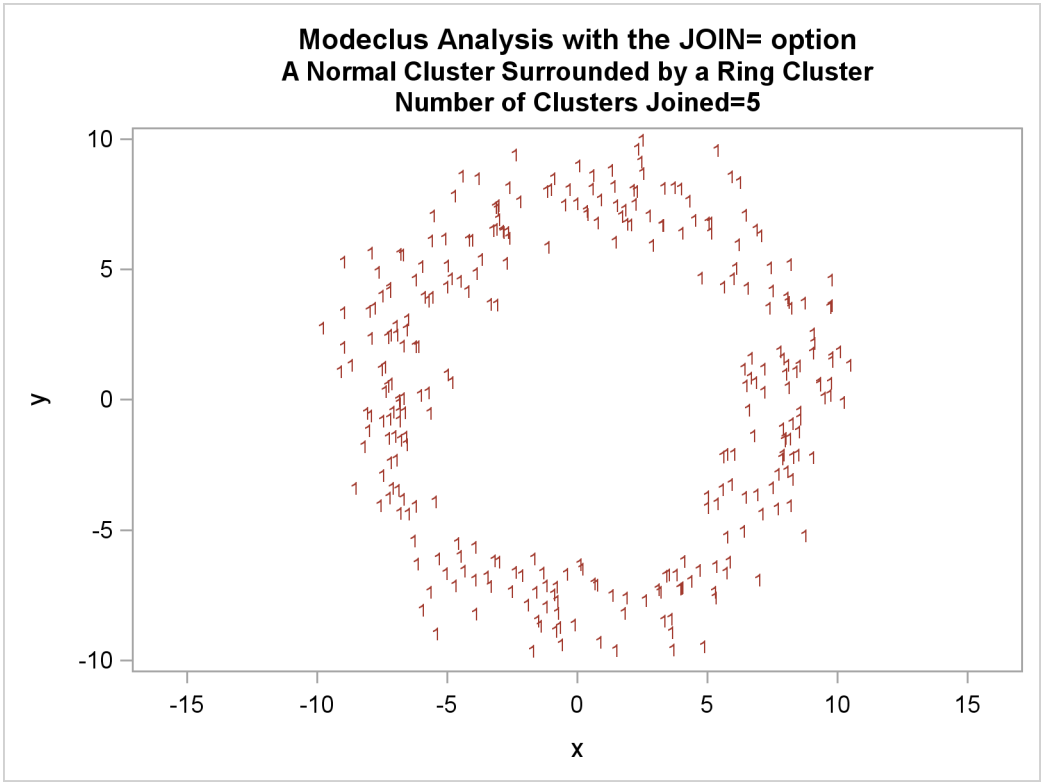
Output 57.3.6 Cluster Memberships When Number of Clusters Joined=3



Output 57.3.7 Cluster Memberships When Number of Clusters Joined=4



Output 57.3.8 Cluster Memberships When Number of Clusters Joined=5



Example 57.4: Cluster Analysis: Hertzsprung-Russell Plot

This example uses computer-generated data to mimic a Hertzsprung-Russell plot (Struve and Zebergs 1962, p. 259) of the temperature and luminosity of stars. The data are plotted and displayed in Output 57.4.1. It appears that there are two main groups of stars and a collection of isolated stars. The long straggling group of points appearing diagonally across the figure represents the main group of stars; the more compact group in the top-right corner contains giant stars. The JOIN= option is specified at a 0.05 significance level with various smoothing parameters. The CK=5 option is specified in order to prevent the numerous outliers from forming separate clusters. The results from PROC MODECLUS is displayed in Output 57.4.2. The cluster memberships are then plotted by PROC SGPLOT, as displayed in Output 57.4.3 through Output 57.4.5.

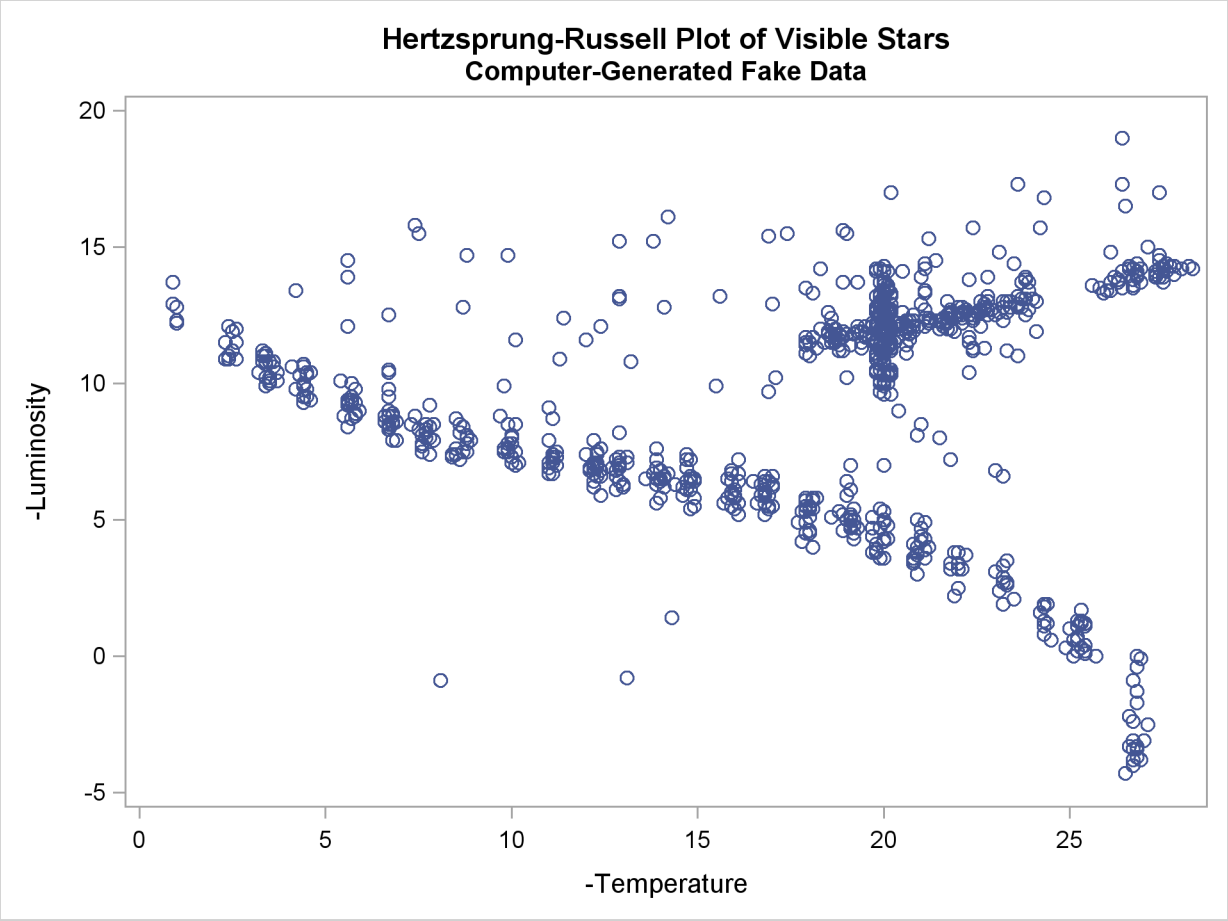
Note that the graphic output from PROC SGPLOT in Output 57.4.3 is not available when _R_ = 2.5 because only one cluster remains after joining at a 5% significance level, and the results are not written to the OUT= data set. See the description of the JOIN= option on page 4105 for more information.

The following statements produce Output 57.4.1 through Output 57.4.5:

```
   title 'Hertzsprung-Russell Plot of Visible Stars';
   title2 'Computer-Generated Fake Data';
data hr;
   input x y @@;
   label x='-Temperature'
         y='-Luminosity';
   datalines;
1.0   12.8   0.9  13.7   0.9  12.9   1.0  12.3   1.0  12.2   2.6  10.9
2.4   10.9   2.5  11.2   2.3  11.5   2.6  12.0   2.4  12.1   2.3  10.9
2.6   11.5   2.5  11.9   2.4  11.0   3.4  11.1   3.3  11.2   3.4  11.1
3.4    9.9   3.2  10.4   3.5  10.8   3.4  11.0   3.3  11.2   3.3  10.8
3.5   10.0   3.5  10.2   3.4  10.2   3.6  10.6   3.7  10.4   3.7  10.1
3.4   10.7   3.4  10.8   3.3  11.0   3.6  10.8   3.5  10.1   4.5  10.3
4.6    9.4   4.3  10.3   4.6   9.4   4.4   9.9   4.5  10.4   4.4   9.9
4.6    9.4   4.4  10.7   4.4   9.3   4.4   9.5   4.1  10.6   4.4  10.6
4.5   10.3   4.4  10.0   4.2   9.8   4.5   9.5   4.2  13.4   4.6  10.4
4.5    9.8   5.8   8.8   5.6   8.4   5.6  13.9   5.7   9.5   5.6  14.5
5.6    9.2   5.7   8.7   5.7   9.4   5.7   9.3   5.6   9.4   5.8   9.8
5.5    8.8   5.8   8.9   5.7   9.4   5.6  12.1   5.4  10.1   5.8   9.3
5.9    9.0   5.7  10.0   5.6   9.3   6.6   8.6   6.7   8.5   6.7  12.5

... more lines ...

26.4  14.1  26.6  14.2  27.5  13.7  27.6  14.4  27.8  14.0  27.4  14.7
25.8  13.5  25.6  13.6  26.8  14.4  26.4  19.0  26.0  13.4  27.3  14.0
27.5  14.3  27.4  14.5  26.3  13.8  26.9  13.7  26.3  13.7  27.7  14.3
27.3  14.1  28.3  14.2  17.4  15.5  13.8  15.2  12.0  11.6  14.1  12.8
17.1  10.2  16.9  15.4  18.5  12.6  14.2  16.1  23.2   6.6  11.4  12.4
20.4  11.7  20.9   8.1  18.9  13.7  16.9   9.7  15.5   9.9  18.3  14.2
19.3  13.7  17.0  12.9  10.1  11.6  17.9  13.5  14.3   1.4  13.1  -0.8
8.1  -0.9  20.0   7.0  21.0   8.5  15.6  13.2
;
```

```
proc sgplot data=hr;
   scatter y=y x=x;
run;

proc modeclus data=hr m=1 r=1 1.5 2 2.5 ck=5
              join=.05 short out=out;
run;

title2 'MODECLUS Analysis';
proc sgplot data=out;
   scatter y=y x=x/group=cluster;
   by _R_;
run;
```

Output 57.4.1 Scatter Plot of Data



Output 57.4.2 Results from PROC MODECLUS

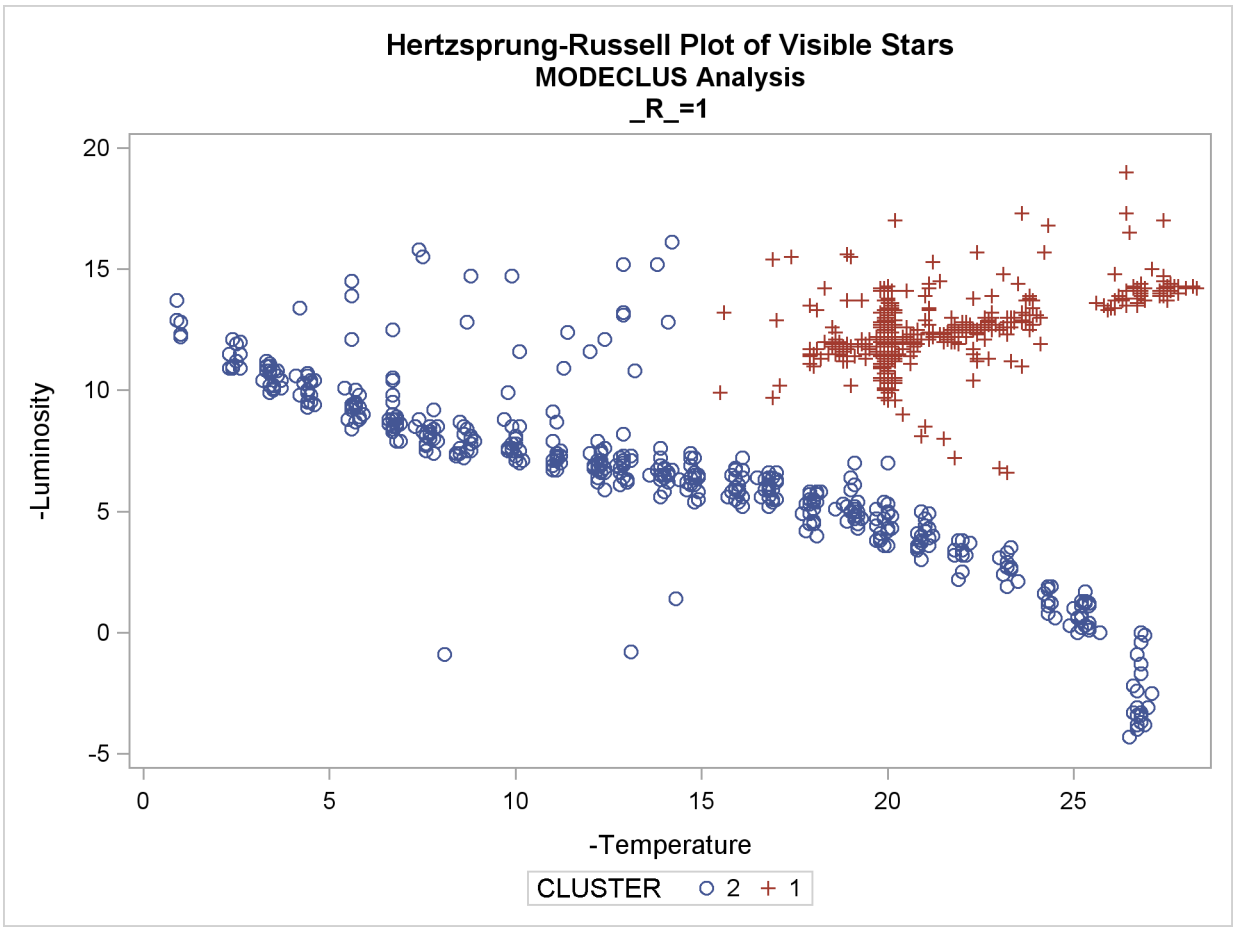
Hertzsprung-Russell Plot of Visible Stars
Computer-Generated Fake Data

The MODECLUS Procedure

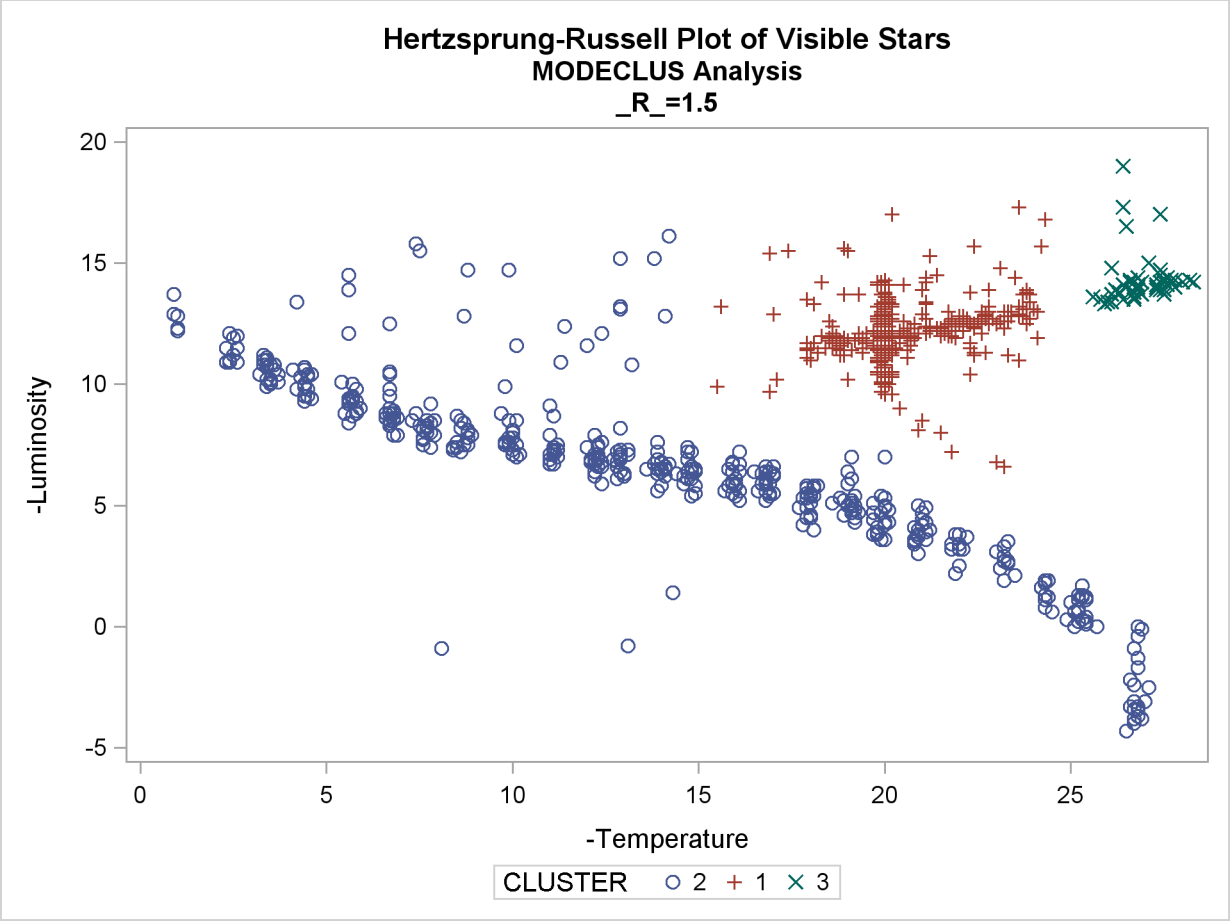
Cluster Summary

R	CK	Number of Clusters Joined	Maximum P-value	Number of Clusters	Frequency of Unclassified Objects
1	5	14	0.0001	2	0
1.5	5	6	0.0000	3	0
2	5	4	0.0000	2	0
2.5	5	2	0.0000	1	0

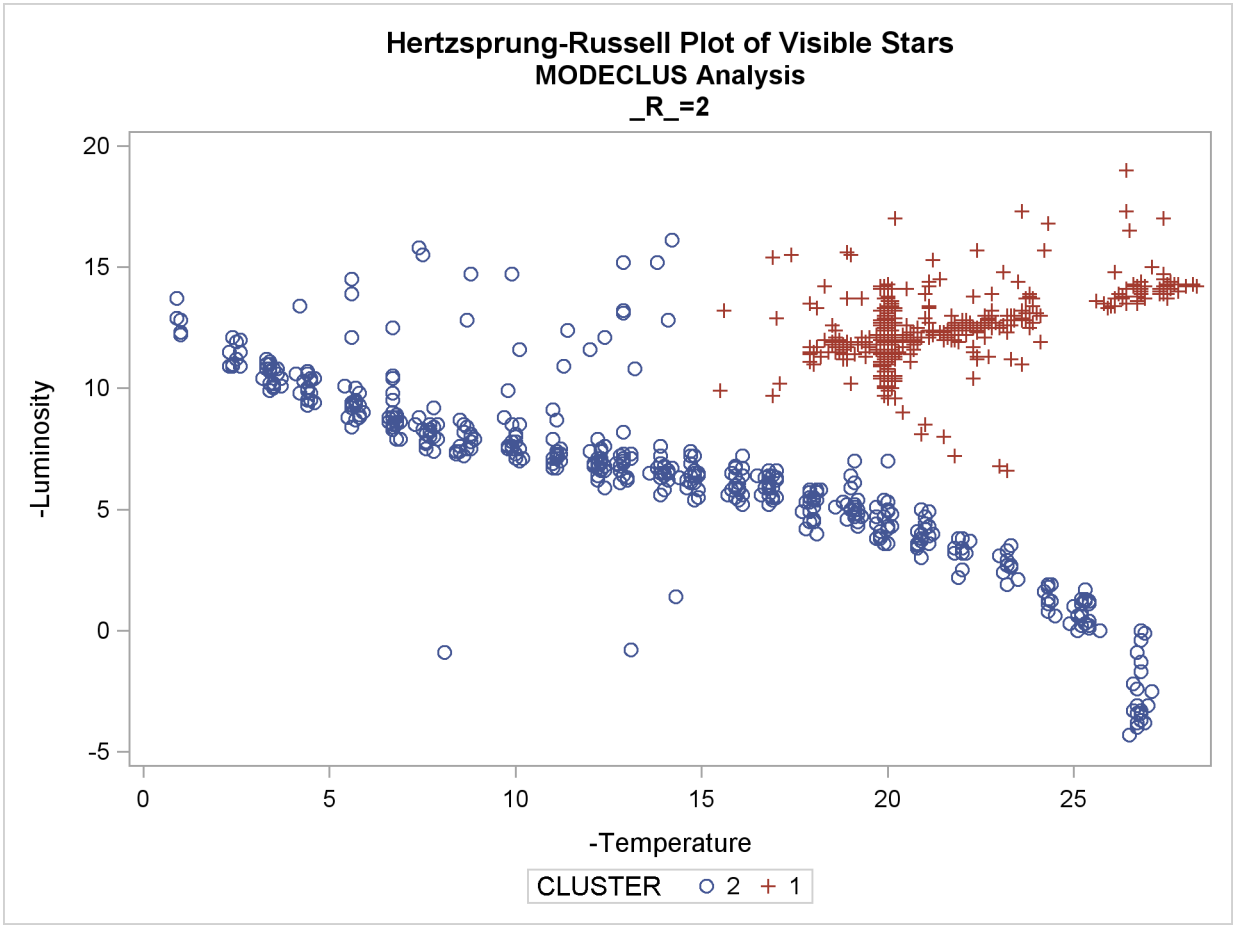
Output 57.4.3 Scatter Plots of Cluster Memberships by _R_= 1



Output 57.4.4 Scatter Plots of Cluster Memberships by _R_= 1.5



Output 57.4.5 Scatter Plots of Cluster Memberships by _R_=2



Example 57.5: Using the TRACE Option When METHOD=6

To illustrate how the TRACE option can help you to understand the clustering process when METHOD=6 is specified, the following data set is created with 12 observations:

```
data test;
   input x@@;
   datalines;
1 2 3 4 5 7.5 9 11.5 13 14.5 15 16
;
```

The first five observations seem to be close to each other, and the last five observations seem to be close to each other. Observation 6 is separated from the first five observations with a (Euclidean) distance of 2.5, and the same distance separates observation 7 from the last five observations. Observations 6 and 7 differ by 1.5.

Suppose METHOD=6 with a radius of 2.5 is chosen for the cluster analysis. You can specify the TRACE option to understand how each observation is assigned.

The following statements produce Output 57.5.1 and Output 57.5.2:

```
/*-- METHOD=6 with TRACE and THRESHOLD=0.5 (default) --*/
title 'METHOD=6 with TRACE and THRESHOLD=0.5 (default)';
proc modeclus data=test method=6 r=2.5 trace short out=out;
   var x;
run;

title2 'Plot of DENSITY*X=CLUSTER';
proc sgplot data=out;
   scatter y=density x=x / group=cluster datalabel=_obs_;
run;
```

Output 57.5.1 Partial Output of METHOD=6 with TRACE and Default THRESHOLD=

METHOD=6 with TRACE and THRESHOLD=0.5 (default)

The MODECLUS Procedure
R=2.5 METHOD=6

Trace of Clustering Algorithm

Obs	Density	Cluster Old	Cluster New	Ratio
3	0.0833333	-1	1	M
2	0.0666667	0	1	N
4	0.0666667	0	1	N
5	0.0666667	0	1	N
1	0.0500000	0	1	N
6	0.0500000	0	1	0.571
7	0.0500000	-1	1	0.500
9	0.0666667	-1	2	M
8	0.0500000	0	2	N
10	0.0666667	-1	2	S
12	0.0500000	0	2	N
11	0.0666667	-1	2	S

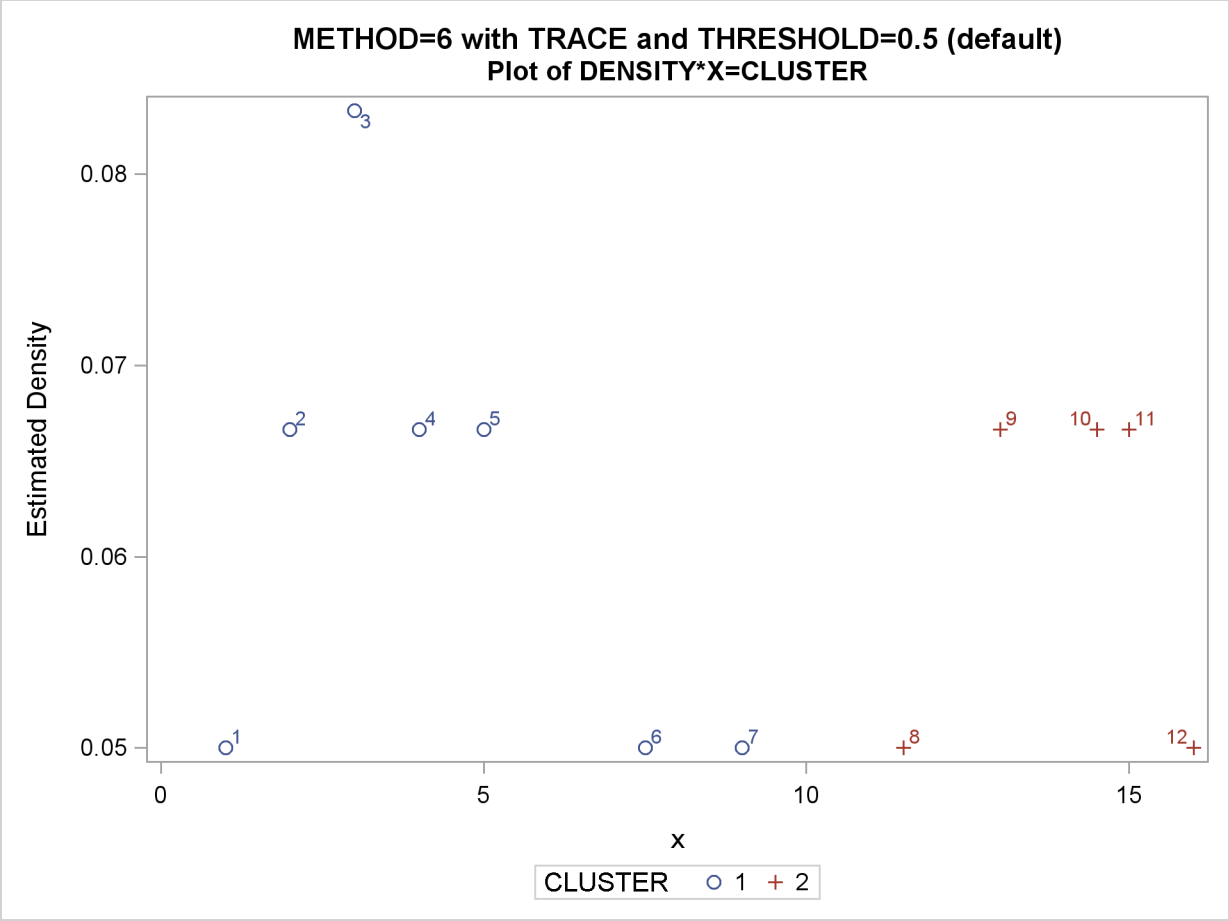
METHOD=6 with TRACE and THRESHOLD=0.5 (default)

The MODECLUS Procedure

Cluster Summary

R	Number of Clusters	Frequency of Unclassified Objects
2.5	2	0

Output 57.5.2 Density Plot



Note that in Output 57.5.1, observation 7 is originally a seed (indicated by a value of –1 in the "Old" column) and then assigned to cluster 1. This is because the ratio of observation 7 to cluster 1 is 0.5 and is not less than the default value of the THRESHOLD= option (0.5).

If the value of the THRESHOLD= option is increased to 0.55, observation 7 should be excluded from cluster 1 and the cluster membership of observation 7 is changed.

The following statements produce Output 57.5.3 and Output 57.5.4:

```
/*-- METHOD=6 with TRACE and THRESHOLD=0.55 --*/
title 'METHOD=6 with TRACE and THRESHOLD=0.55';
proc modeclus data=test method=6 r=2.5 trace threshold=0.55 short
              out=out;
   var x;
run;

title2 'Plot of DENSITY*X=CLUSTER with TRACE and THRESHOLD=0.55';
proc sgplot data=out;
   scatter y=density x=x / group=cluster datalabel=_obs_;
run;
```

Output 57.5.3 Partial Output of METHOD=6 with TRACE and THRESHOLD=.55

METHOD=6 with TRACE and THRESHOLD=0.55

The MODECLUS Procedure
R=2.5 METHOD=6

Trace of Clustering Algorithm

Obs	Density	Cluster Old	Cluster New	Ratio
3	0.0833333	-1	1	M
2	0.0666667	0	1	N
4	0.0666667	0	1	N
5	0.0666667	0	1	N
1	0.0500000	0	1	N
6	0.0500000	0	1	0.571
9	0.0666667	-1	2	M
8	0.0500000	0	2	N
10	0.0666667	-1	2	S
12	0.0500000	0	2	N
11	0.0666667	-1	2	S
7	0.0500000	-1	2	S

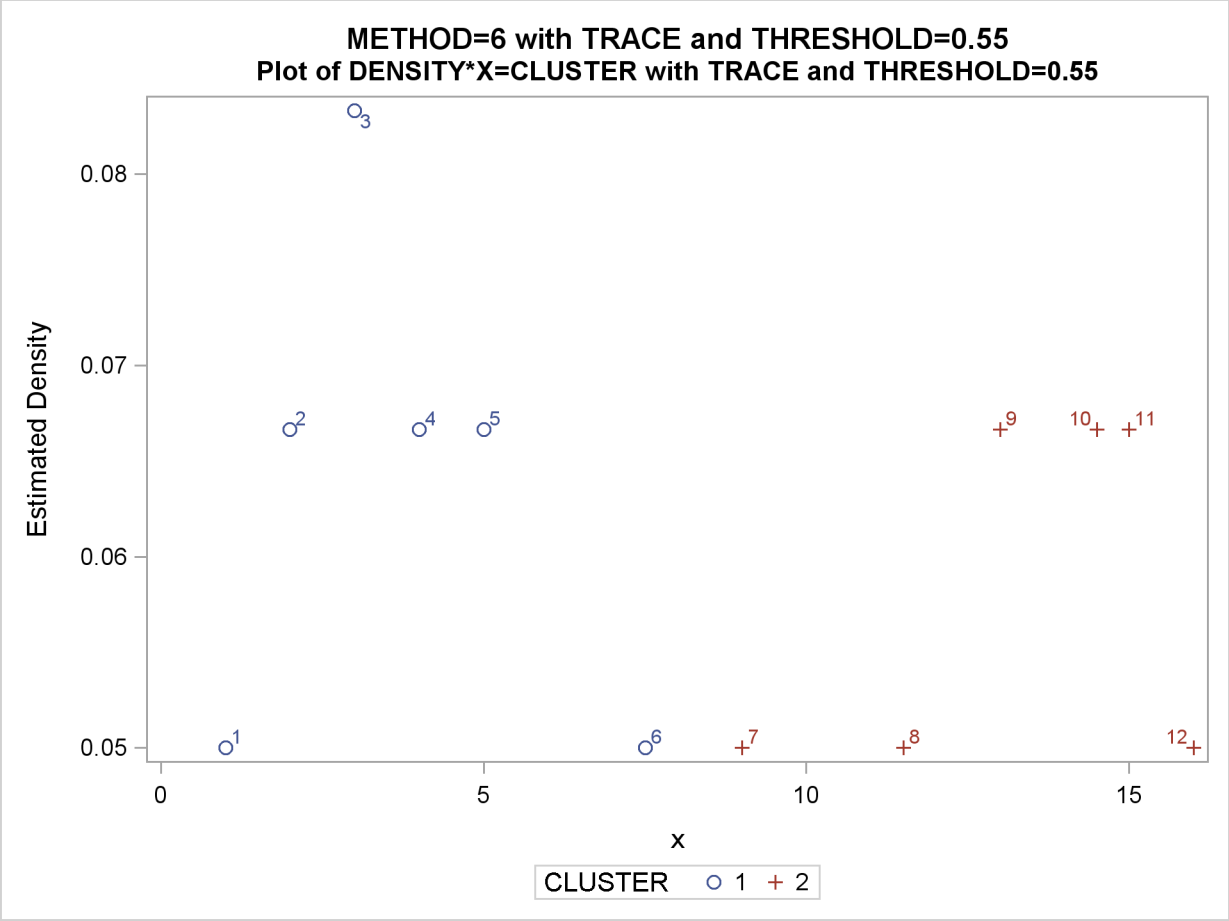
METHOD=6 with TRACE and THRESHOLD=0.55

The MODECLUS Procedure

Cluster Summary

R	Number of Clusters	Frequency of Unclassified Objects
2.5	2	0

Output 57.5.4 Density Plot



In Output 57.5.3, observation 7 is a seed that is excluded by cluster 1 because its ratio to cluster 1 is less than 0.55. Being a neighbor of a member (observation 8) of cluster 2, observation 7 eventually joins cluster 2 even though it remains a "SEED." (See **Step 2.2** in the section "METHOD=6" on page 4114.)

References

Barnett, V., ed. (1981), *Interpreting Multivariate Data*, New York: John Wiley & Sons.

Gitman, I. (1973), "An Algorithm for Nonsupervised Pattern Classification," *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-3, 66–74.

Hartigan, J. A. and Hartigan, P. M. (1985), "The Dip Test of Unimodality," *Annals of Statistics*, 13, 70–84.

Huizinga, D. H. (1978), "A Natural or Mode Seeking Cluster Analysis Algorithm," Technical Report 78-1, Behavioral Research Institute, 2305 Canyon Blvd., Boulder, CO 80302.

Koontz, W. L. G. and Fukunaga, K. (1972a), "A Nonparametric Valley-Seeking Technique for Cluster Analysis," *IEEE Transactions on Computers*, C-21, 171–178.

Koontz, W. L. G. and Fukunaga, K. (1972b), "Asymptotic Analysis of a Nonparametric Clustering Technique," *IEEE Transactions on Computers*, C-21, 967–974.

Koontz, W. L. G., Narendra, P. M., and Fukunaga, K. (1976), "A Graph-Theoretic Approach to Nonparametric Cluster Analysis," *IEEE Transactions on Computers*, C-25, 936–944.

Minnotte, M. C. (1992), "A Test of Mode Existence with Applications to Multimodality," Ph.D. dissertation, Rice University, Department of Statistics.

Mizoguchi, R. and Shimura, M. (1980), "A Nonparametric Algorithm for Detecting Clusters Using Hierarchical Structure," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-2, 292–300.

Müller, D. W. and Sawitzki, G. (1991), "Excess Mass Estimates and Tests for Multimodality," *JASA* 86, 738–746.

Polonik, W. (1993), "Measuring Mass Concentrations and Estimating Density Contour Clusters—An Excess Mass Approach," Technical Report, Beitraege zur Statistik Nr. 7, University of Heidelberg.

Sarle, W. S. (1982), "Cluster Analysis by Least Squares," *Proceedings of the Seventh Annual SAS Users Group International Conference*, 651–653.

Scott, D. W. (1992), *Multivariate Density Estimation: Theory, Practice, and Visualization*, New York: John Wiley & Sons.

Silverman, B. W. (1986), *Density Estimation*, New York: Chapman & Hall.

Struve, O. and Zebergs, V. (1962), *Astronomy of the Twentieth Century*, New York: Macmillan.

Tukey, P. A. and Tukey, J. W. (1981), "Data-Driven View Selection; Agglomeration and Sharpening," in Barnett (1981).

Wong, M. A. and Lane, T. (1983), "A kth Nearest Neighbor Clustering Procedure," *Journal of the Royal Statistical Society*, Series B, 45, 362–368.

Wong, M. A. and Schaack, C. (1982), "Using the kth Nearest Neighbor Clustering Procedure to Determine the Number of Subpopulations," *American Statistical Association 1982 Proceedings of the Statistical Computing Section*, 40–48.

Chapter 58

The MULTTEST Procedure

Contents

Overview: MULTTEST Procedure **4182**
Getting Started: MULTTEST Procedure **4184**
Drug Example 4184
Syntax: MULTTEST Procedure **4187**
PROC MULTTEST Statement 4187
BY Statement 4197
CLASS Statement 4198
CONTRAST Statement 4199
FREQ Statement 4200
STRATA Statement 4200
TEST Statement 4201
Details: MULTTEST Procedure **4203**
Statistical Tests 4203
p-Value Adjustments 4211
Missing Values 4219
Computational Resources 4220
Output Data Sets 4220
Displayed Output 4222
ODS Table Names 4223
ODS Graphics 4224
Examples: MULTTEST Procedure **4225**
Example 58.1: Cochran-Armitage Test with Permutation Resampling 4225
Example 58.2: Freeman-Tukey and *t* Tests with Bootstrap Resampling . . . 4229
Example 58.3: Peto Mortality-Prevalence Test 4233
Example 58.4: Fisher Test with Permutation Resampling 4236
Example 58.5: Inputting Raw *p*-Values 4240
Example 58.6: Adaptive Adjustments and ODS Graphics 4242
References **4248**

Overview: MULTTEST Procedure

The MULTTEST procedure addresses the multiple testing problem. This problem arises when you perform many hypothesis tests on the same data set. Carrying out multiple tests is often reasonable because of the cost of obtaining data, the discovery of new aspects of the data, and the many alternative statistical methods. However, a disadvantage of multiple testing is the greatly increased probability of declaring false significances.

For example, suppose you carry out 10 hypothesis tests at the 5% level, and you assume that the distributions of the p-values from these tests are uniform and independent. Then, the probability of declaring a particular test significant under its null hypothesis is 0.05, but the probability of declaring at least 1 of the 10 tests significant is 0.401. If you perform 20 hypothesis tests, the latter probability increases to 0.642. These high chances illustrate the danger of multiple testing.

PROC MULTTEST approaches the multiple testing problem by adjusting the p-values from a family of hypothesis tests. An adjusted p-value is defined as the smallest significance level for which the given hypothesis would be rejected, when the entire family of tests is considered. The decision rule is to reject the null hypothesis when the adjusted p-value is less then α. For most methods, this decision rule controls the *familywise error rate* at or below the α level. However, the *false discovery rate* controlling procedures control the false discovery rate at or below the α level.

PROC MULTTEST provides the following p-value adjustments:

- Bonferroni
- Šidák
- step-down methods
- Hochberg
- Hommel
- Fisher combination
- bootstrap
- permutation
- adaptive methods
- false discovery rate
- positive FDR

The Bonferroni and Šidák adjustments are simple functions of the raw p-values. They are computationally quick, but they can be too conservative. Step-down methods remove some conservativeness, as do the step-up methods of Hochberg (1988), and the adaptive methods. The bootstrap and permutation adjustments resample the data with and without replacement, respectively, to approximate the distribution of the minimum p-value of all tests. This distribution is then used to adjust the individual raw p-values. The bootstrap and permutation methods are computationally intensive but appealing in that, unlike the other methods, correlations and distributional characteristics are incorporated into the adjustments (Westfall and Young 1989; Westfall et al. 1999).

PROC MULTTEST handles data arising from a multivariate one-way ANOVA model, possibly stratified, with continuous and discrete response variables; it can also accept raw p-values as input

data. You can perform a *t* test for the mean for continuous data with or without a homogeneity assumption, and the following statistical tests for discrete data:

- Cochran-Armitage linear trend test
- Freeman-Tukey double arcsine test
- Peto mortality-prevalence (log-rank) test
- Fisher exact test

The Cochran-Armitage and Peto tests have exact versions that use permutation distributions and asymptotic versions that use an optional continuity correction. Also, with the exception of the Fisher exact test, you can use a stratification variable to construct Mantel-Haenszel-type tests. All of the previously mentioned tests can be one- or two-sided.

As in the GLM procedure, you can specify linear contrasts that compare means or proportions of the treated groups. The output contains summary statistics and regular and multiplicity-adjusted *p*-values. You can create output data sets containing raw and adjusted *p*-values, test statistics and other intermediate calculations, permutation distributions, and resampling information.

The MULTTEST procedure now uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

The GLIMMIX, GLM, MIXED, and LIFETEST procedures also adjust their results for multiple tests. For more information, see the documentation for these procedures, and Westfall et al. (1999).

Getting Started: MULTTEST Procedure

Drug Example

Suppose you conduct a small study to test the effect of a drug on 15 subjects. You randomly divide the subjects into three balanced groups receiving 0 mg, 1 mg, and 2 mg of the drug, respectively. You carry out the experiment and record the presence or absence of 10 side effects for each subject. Your data set is as follows:

```
data Drug;
   input Dose$ SideEff1-SideEff10;
   datalines;
0MG  0   0   1   0   0   1   0   0   0   0
0MG  0   0   0   0   0   0   0   0   0   1
0MG  0   0   0   0   0   0   0   0   1   0
0MG  0   0   0   0   0   0   0   0   0   0
0MG  0   1   0   0   0   0   0   0   0   0
1MG  1   0   0   1   0   1   0   0   1   0
1MG  0   0   0   1   1   0   0   1   0   1
1MG  0   1   0   0   0   0   1   0   0   0
1MG  0   0   1   0   0   0   0   0   0   1
1MG  1   0   1   0   0   0   0   1   0   0
2MG  0   1   1   1   0   1   1   1   0   1
2MG  1   1   1   1   1   1   0   1   1   0
2MG  1   0   0   1   0   1   1   0   1   0
2MG  0   1   1   1   1   0   1   1   1   1
2MG  1   0   1   0   1   1   1   0   0   1
;
```

The increasing incidence of 1s for higher dosages in the preceding data set provides an initial visual indication that the drug has an effect. To explore this statistically, you perform an analysis in which the possibility of side effects increases linearly with drug level. You can analyze the data for each side effect separately, but you are concerned that, with so many tests, there might be a high probability of incorrectly declaring some drug effects significant. You want to correct for this multiplicity problem in a way that accounts for the discreteness of the data and for the correlations between observations on the same unit.

PROC MULTTEST addresses these concerns by processing all of the data simultaneously and adjusting the *p*-values. The following statements perform a typical analysis:

```
ods graphics on;
proc multtest bootstrap nsample=20000 seed=41287 notables
              plots=PByTest(vref=0.05 0.1);
   class Dose;
   test ca(SideEff1-SideEff10);
   contrast 'Trend' 0 1 2;
run;
ods graphics off;
```

This analysis uses the BOOTSTRAP option to adjust the *p*-values. The NSAMPLE= option requests 20,000 samples for the bootstrap analysis, and the starting seed for the random number generator is 41287. The NOTABLES option suppresses the display of summary statistics for each side effect and drug level combination. The PLOTS= option displays a visual summary of the unadjusted and adjusted *p*-values against each test, and the VREF= option adds reference lines to the display.

The CLASS statement is used to specify the grouping variable, Dose. The **ca(sideeff1-sideeff10)** specification in the TEST statement requests a Cochran-Armitage linear trend test for all 10 characteristics. The CONTRAST statement gives the coefficients for the linear trend test.

The "Model Information" table in Figure 58.1 describes the statistical tests performed by PROC MULTTEST. For this example, PROC MULTTEST carries out a two-tailed Cochran-Armitage linear trend test with no continuity correction or strata adjustment. This test is performed on the raw data and on 20,000 bootstrap samples.

Figure 58.1 Output Summary for the MULTTEST Procedure

The Multtest Procedure

Model Information

Test for discrete variables	Cochran-Armitage
Z-score approximation used	Everywhere
Continuity correction	0
Tails for discrete tests	Two-tailed
Strata weights	None
P-value adjustment	Bootstrap
Number of resamples	20000
Seed	41287

The "Contrast Coefficients" table in Figure 58.2 displays the coefficients for the Cochran-Armitage test. They are 0, 1, and 2, as specified in the CONTRAST statement.

Figure 58.2 Coefficients Used in the MULTTEST Procedure

Contrast Coefficients

	Dose		
Contrast	0MG	1MG	2MG
Trend	0	1	2

The "p-Values" table in Figure 58.3 lists the *p*-values for the drug example. The Raw column lists the *p*-values for the Cochran-Armitage test on the original data, and the Bootstrap column provides the bootstrap adjustment of the raw *p*-values.

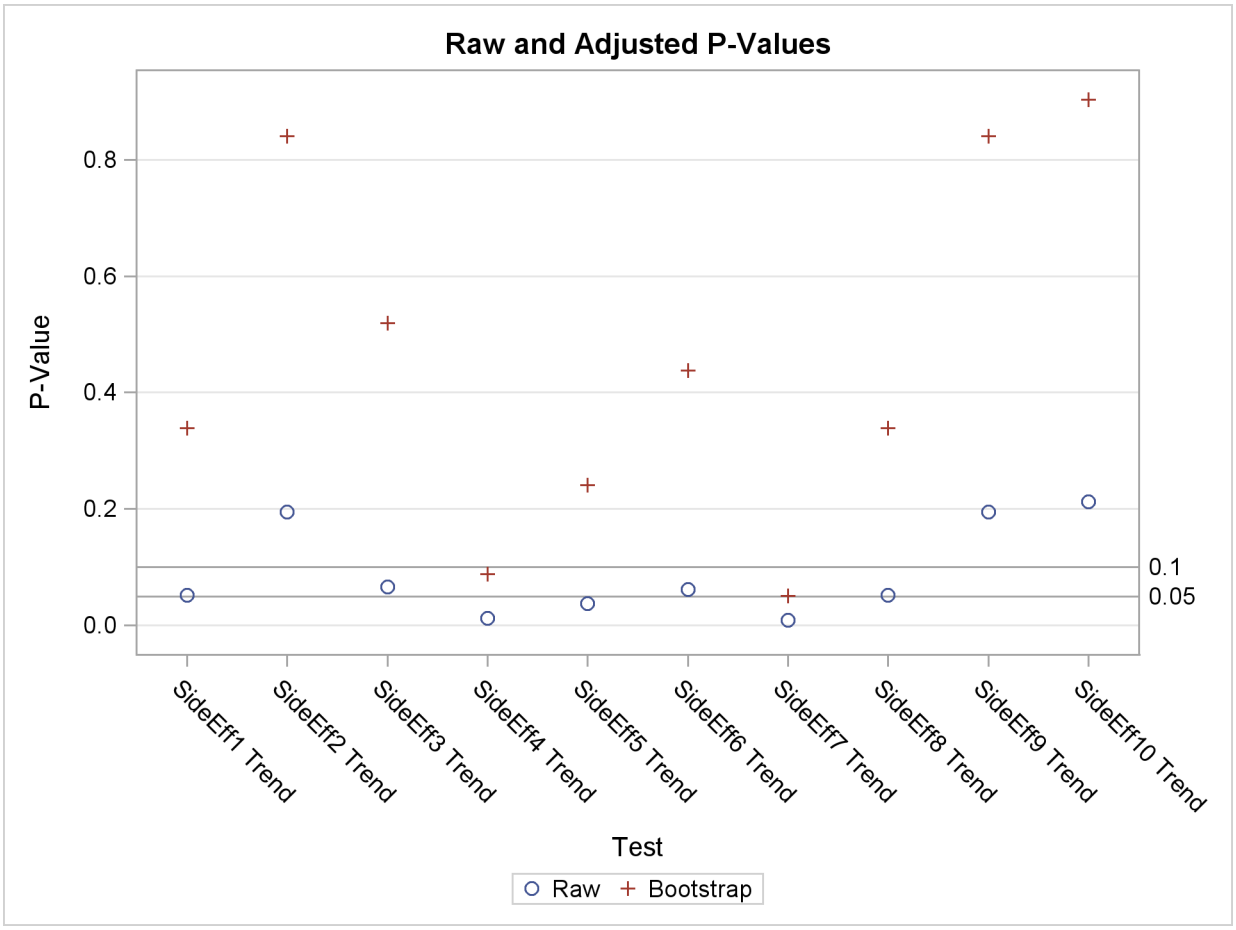
Note that the raw *p*-values lead you to reject the null hypothesis of no linear trend for 3 of the 10 characteristics at the 5% level and 7 of the 10 characteristics at the 10% level. The bootstrap *p*-values, however, lead to this conclusion for 0 of the 10 characteristics at the 5% level and only 2 of the 10 characteristics at the 10% level; you can also see this in Figure 58.4.

Figure 58.3 Summary of *p*-Values for the MULTTEST Procedure

p-Values

Variable	Contrast	Raw	Bootstrap
SideEff1	Trend	0.0519	0.3388
SideEff2	Trend	0.1949	0.8403
SideEff3	Trend	0.0662	0.5190
SideEff4	Trend	0.0126	0.0884
SideEff5	Trend	0.0382	0.2408
SideEff6	Trend	0.0614	0.4383
SideEff7	Trend	0.0095	0.0514
SideEff8	Trend	0.0519	0.3388
SideEff9	Trend	0.1949	0.8403
SideEff10	Trend	0.2123	0.9030

Figure 58.4 Adjusted *p*-Values



The bootstrap adjustment gives the probability of observing a *p*-value as extreme as each given *p*-value, considering all 10 tests simultaneously. This adjustment incorporates the correlation of the raw *p*-values, the discreteness of the data, and the multiple testing problem. Failure to account for these issues can certainly lead to misleading inferences for these data.

Syntax: MULTTEST Procedure

The following statements are available in PROC MULTTEST:

PROC MULTTEST < *options* > **;**
BY *variables* **;**
CLASS *variable* **;**
CONTRAST *'label' values* **;**
FREQ *variable* **;**
STRATA *variable* **;**
TEST *name (variables* < */ options* >*)* **;**

Statements following the PROC MULTTEST statement can appear in any order. The CLASS and TEST statements are required unless the INPVALUES= option is specified.

The syntax of each statement is described in the following section in alphabetical order after the description of the PROC MULTTEST statement.

PROC MULTTEST Statement

PROC MULTTEST < *options* > **;**

The PROC MULTTEST statement invokes the MULTTEST procedure and specifies the p-value adjustments. The options available in the PROC MULTTEST statement are listed in Table 58.1 grouped by their function, and are described in alphabetical order following the table.

Table 58.1 PROC MULTTEST Statement Options by Function

Option	Description
FWE-Controlling *p*-Value Adjustments	
ADAPTIVEHOLM	computes the adaptive step-down Bonferroni adjustment
ADAPTIVEHOCHBERG	computes the adaptive step-up Bonferroni adjustment
BONFERRONI	computes the Bonferroni adjustment
BOOTSTRAP	computes the bootstrap min-p adjustment
FISHER_C	computes Fisher's combination adjustment
HOCHBERG	computes the step-up Bonferroni adjustment
HOMMEL	computes Hommel's adjustment
HOLM	computes the step-down Bonferroni adjustment
PERMUTATION	computes the permutation min-p adjustment
SIDAK	computes Šidák's adjustment
STEPBON	computes the step-down Bonferroni adjustment
STEPBOOT	computes the step-down bootstrap adjustment
STEPPERM	computes the step-down permutation adjustment
STEPSID	computes the step-down Šidák adjustment

Table 58.1 *continued*

Option	Description
FDR-Controlling *p*-Value Adjustments	
ADAPTIVEFDR	computes the adaptive linear step-up adjustment
DEPENDENTFDR	computes the linear step-up adjustment under dependence
FDR	computes the linear step-up adjustment
FDRBOOT	computes the linear step-up bootstrap min-*p* adjustment
FDRPERM	computes the linear step-up permutation min-*p* adjustment
PFDR	computes the positive FDR adjustment
Input/Output Data Sets	
DATA=	names the input data set
INPVALUES=	names the input data set of raw *p*-values
OUT=	names the output data set
OUTPERM=	names the output permutation data set
OUTSAMP=	names the output resample data set
Displayed Output Options	
NOPRINT	suppresses all tables
NOTABLES	suppresses variable tables
NOZEROS	suppresses zero tables for CLASS variables
NOPVALUE	suppresses "p-Values" table
PLOTS	requests ODS Graphics
Resampling Options	
CENTER	mean-centers continuous variables before resampling
NOCENTER	does not mean-center continuous variables before resampling
NSAMPLE=	specifies the number of resamples
RANUNI	specifies a different random number generator
SEED=	specifies seed for resampling
CLASS Variable Options	
NOZEROS	suppresses zero tables for CLASS variables
ORDER=	specifies CLASS variable order
Computational Options	
EPSILON=	specifies the comparison value
NTRUENULL=	specifies the estimation method for the number of true nulls
PTRUENULL=	specifies the estimation method for the proportion of true nulls

You can specify the following options in the PROC MULTTEST statement.

ADAPTIVEHOCHBERG

AHOC

requests adjusted *p*-values by using the Hochberg and Benjamini (1990) adaptive step-up Bonferroni method. See the section "Adaptive Adjustments" on page 4215 for more details.

ADAPTIVEHOLM

AHOLM

requests adjusted *p*-values by using the Hochberg and Benjamini (1990) adaptive step-down Bonferroni method. See the section "Adaptive Adjustments" on page 4215 for more details.

ADAPTIVEFDR

AFDR

requests adjusted *p*-values by using the Benjamini and Hochberg (2000) adaptive linear step-up method. See the section "Adaptive False Discovery Rate" on page 4218 for more details.

BONFERRONI

BON

specifies that the Bonferroni adjustments (number of tests $\times$ *p*-value) be computed for each test. These adjustments can be extremely conservative and should be viewed with caution. When exact tests are specified via the PERMUTATION= option in the TEST statement, the actual permutation distributions are used, resulting in a much less conservative version of this procedure (Westfall and Wolfinger 1997). See the section "Bonferroni" on page 4212 for more details.

BOOTSTRAP

BOOT

specifies that the *p*-values be adjusted by using the bootstrap method to resample vectors (Westfall and Young 1993). Resampling is performed with replacement and independently within levels of the STRATA variable. Continuous variables are mean-centered by default prior to resampling; specify the NOCENTER option to change this. See the section "Bootstrap" on page 4213 for more details. The BOOTSTRAP option is not allowed with the Peto test.

If the PERMUTATION= suboption is used with the CA test in the TEST statement, the exact permutation distribution is recomputed for each bootstrap sample. **CAUTION:** This can be very time-consuming. It is preferable to use permutation resampling when permutation base tests are used.

CENTER

requests that continuous variables be mean-centered prior to resampling. The default action is to mean-center for bootstrap resampling and not to mean-center for permutation resampling.

DATA=*SAS-data-set*

names the input SAS data set to be used by PROC MULTTEST. The default is to use the most recently created data set. The DATA= and INPVALUES= options cannot both be specified.

DEPENDENTFDR

DFDR

requests adjusted *p*-values by using the method of Benjamini and Yekateuli (2001). See the section "Dependent False Discovery Rate" on page 4217 for more details.

EPSILON=*number*

specifies the amount by which two p-values must differ to be declared unequal. The value *number* must be between 0 and 1; the default value is 1000 times the machine epsilon, which is approximately 1E–12. For SAS 9.1 and earlier releases the default value was 1E–8. See Westfall and Young (1993, pp. 165–166) for more information.

FDR

LSU

requests adjusted p-values by using the linear step-up method of Benjamini and Hochberg (1995). These p-values do not control the familywise error rate, but they do control the false discovery rate in some cases. See the section "False Discovery Rate Controlling Adjustments" on page 4216 for more details.

FDRBOOT<(β)>

A bootstrap-resampling false discovery rate controlling method due to Yekateuli and Benjamini (1999). This method uses the same resampling algorithm as the BOOTSTRAP option. Every resample is saved in order to compute a quantile of the resampled p-values; therefore, this method can use a lot of memory. The parameter β designates that a $100(1-\beta)$th quantile is used in the computations for determining the adjustments; by default, $\beta = 0.05$. See the section "False Discovery Rate Resampling Adjustments" on page 4217 for details.

FDRPERM<(β)>

A permutation-resampling false discovery rate controlling method due to Yekateuli and Benjamini (1999). This method uses the same resampling algorithm as the PERMUTATION option. Every resample is saved in order to compute a quantile of the resampled p-values; therefore, this method can use a lot of memory. The parameter β designates that a $100(1-\beta)$th quantile is used in the computations for determining the adjustments; by default, $\beta = 0.05$. See the section "False Discovery Rate Resampling Adjustments" on page 4217 for details.

FISHER_C

FIC

requests adjusted p-values by using Fisher's combination method. See the section "Fisher Combination" on page 4215 for more details.

HOCHBERG

HOC

requests adjusted p-values by using the step-up Bonferroni method due to Hochberg (1988). See the section "Hochberg" on page 4215 for more details.

HOMMEL

HOM

requests adjusted p-values by using the method of Hommel (1988). See the section "Hommel" on page 4215 for more details.

HOLM

is an alias for the STEPBON adjustment.

INPVALUES< *(pvalue-name)* **>=***SAS-data-set*

names an input SAS data set that includes a variable containing raw p-values. The MULTTEST procedure adjusts the collection of raw p-values for multiplicity. Resampling-based adjustments are not permitted with this type of data input. The CLASS, CONTRAST, FREQ, STRATA, and TEST statements are ignored when an INPVALUES= data set is specified. The INPVALUES= and DATA= options cannot both be specified. The *pvalue-name* enables you to specify the name of the p-value column from your data set. By default, *pvalue-name*='raw_p'. The INPVALUES= data set can contain variables in addition to the raw p-values variable; see Example 58.5 for an example.

NOCENTER

requests that continuous variables not be mean-centered prior to resampling. The default action is to mean-center for bootstrap resampling and not to mean-center for permutation resampling.

NOPRINT

suppresses the normal display of results. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

NOPVALUE

suppresses the display of the "p-Values" table of raw and adjusted p-values. This option is most useful when you are adjusting many tests and need to create only an OUT= data set or display graphics.

NOTABLES

suppresses display of the "Discrete Variable Tabulations" and "Continuous Variable Tabulations" tables.

NOZEROS

suppresses display of tables having zero occurrences for all CLASS levels.

NSAMPLE=*number*

N=*number*

specifies the number of resamples for use with the resampling methods. The value *number* must be a positive integer; by default, 20,000 resamples are used. Large values of *number* (20,000 or more) are usually recommended for accuracy, but long execution times can result, particularly with large data sets.

NTRUENULL=*keyword* | *value*

M0=*keyword* | *value*

Controls the method used to estimate the number of true NULL hypotheses (m_0) for the adaptive methods. This option is ignored unless one of the adaptive methods is specified. By default, PROC MULTTEST uses the DECREASESLOPE method for the ADAPTIVEHOLM and ADAPTIVEHOCHBERG adjustments, and the LOWESTSLOPE method for ADAPTIVEFDR adjustment. For the PFDR adjustment, the SPLINE method is attempted first. If the estimate is nonpositive or if the slope of the spline at the last λ is greater than 0.1 times the range of the fitted spline values, then the BOOTSTRAP method is used.

You can specify a positive integer as the *value*, or you can specify one of the *keywords* in the following list. Alternatively, you can specify the proportion of true NULL hypotheses by using the PTRUENULL= option. Suppose you have m tests with ordered p-values $p_{(1)} \leq \ldots \leq p_{(m)}$, and define $q_{(i)} = 1 - p_{(i)}$.

BOOTSTRAP<(*bootstrap-options*)> uses the bootstrap method of Storey and Tibshirani (2003). Compute the proportion of true null hypotheses $\hat{\pi}_0(\lambda) = \frac{m - N(\lambda) + f}{(1-\lambda)m}$ for $\lambda \in L = \{0, 0.05, \ldots, 0.95\}$, where $N(\lambda)$ is the number of p-values less than or equal to λ, and $f = 1$ for the finite-sample case; otherwise $f = 0$. For each λ, bootstrap on the p-values to form B bootstrap versions $\hat{\pi}_0^b(\lambda), b = 1, \ldots, B$, and choose the λ that yields the minimum $\widehat{\mathrm{MSE}}(\lambda) = \frac{1}{B}\sum_{b=1}^{B}(\hat{\pi}_0^b(\lambda) - \min_{\lambda' \in L}\hat{\pi}_0(\lambda'))^2$. The available *bootstrap-options* are as follows:

FINITE modifies the computations for the finite-sample case of the PFDR option, described on page 4194.

NBOOT=*B* specifies the number of bootstrap resamples of the raw p-values for the λ computations. NBOOT= 10,000 by default; B must be a positive integer.

NLAMBDA=*n* specifies that the "optimal" λ is the value in $\{0, \frac{1}{n}, \ldots, \frac{n-1}{n}\}$ that minimizes the MSE. NLAMBDA= 20 by default; n must be an integer greater than 1.

DECREASESLOPE uses the method of Schweder and Spjøtvoll (1982) as modified by Hochberg and Benjamini (1990). Let b_i be the slope of the least squares line fit to $\{q_{(m)}, \ldots, q_{(m-i+1)}\}$ and through the origin, for $i = 1, \ldots, m$. Find the first $i = m-1, m-2, \ldots, 1$ such that $b_i < b_{i+1}$. Then $\hat{m}_0 = \mathrm{ceil}(\frac{1}{b_{i+1}} - 1)$.

KSTEST<(β)> uses the Kolmogov-Smirnov uniformity test method of Turkheimer, Smith, and Schmidt (2001). Let $k_{min} = 1, k_{max} = m$, and the Kolmogorov-Smirnov statistic $D = \max(q_{(i)} - i/(m+1)(\sqrt{k} + 0.12 + 0.11/\sqrt{k}))$. If D is greater than the upper-tail probability (Press et al. 1992), then $k_{max} = k, k = \mathrm{floor}((k_{min} + k)/2)$; otherwise, let $k_{min} = k, k = \mathrm{floor}((k + k_{max})/2)$. Repeat until $k = k_{min}$. Next compute the slope b of the weighted least squares regression line on the k smallest $q_{(i)}$ by using weights $w_i = i(k - i + 1)/((k+1)^2(k+2))$. Then $\hat{m}_0 = \mathrm{ceil}(\frac{1}{b} - 1)$.

LEASTSQUARES uses a linear least squares method to search for the correct cutpoint. For each $i = 0, \ldots, m$ compute the SSE of the least squares line through the origin fitting $\{q_{(m)}, \ldots, q_{(m-i+1)}\}$, let b_i be the slope of this line, and add the SSE of the unconstrained least squares line through the rest of the qs. For $i = 0$ compute the SSE for the unconstrained line. The argument i that minimizes the SSE is the cutpoint: if $i = 0$ then $\hat{m}_0 = 0$; if $i = m$ then $\hat{m}_0 = m$; otherwise $\hat{m}_0 = \mathrm{ceil}(\frac{1}{b_i} - 1)$.

LOWESTSLOPE uses the lowest slope method of Benjamini and Hochberg (2000). Find the first $i = 1, \ldots, m$ such that $b_i = q_{(i)}/(m - i + 1)$ decreases. Then $\hat{m}_0 = \mathrm{floor}(\min(\frac{1}{b_i} + 1, m))$.

MEANDIFF uses the mean of differences method of Hsueh, Chen, and Kodell (2003). Let $\bar{d}_i = \frac{q_{(m-i+1)}}{i}$ and estimate $\hat{m}_0^i = \frac{1}{\bar{d}_i} - 1$. Start from $i = m$ and proceed downward until the first time $\hat{m}_0^{i-1} \geq \hat{m}_0^i$ occurs.

SPLINE< (*spline-options*) > uses the cubic spline method of Storey and Tibshirani (2003). For each $\lambda \in \{0, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{n-1}{n}\}$ compute $\hat{\pi}_0(\lambda) = \frac{\#\{p_i > \lambda\}}{m(1-\lambda)}$. Let $\widehat{f}(\lambda)$ be the natural cubic spline with 3 degrees of freedom of $\hat{\pi}_0(\lambda)$ versus λ. Estimate $\hat{\pi}_0$ by taking the spline value at the last λ: $\hat{\pi}_0 = \hat{\pi}_0(\frac{n-1}{n})$, so that $\hat{m}_0 = m\hat{\pi}_0$. The available *spline-options* are as follows:

DF=*df* sets the degrees of freedom of the spline, where *df* is a nonnegative integer. The default is DF=3.

DFCONV=*number* specifies the absolute change in spline degrees of freedom value for concluding convergence. If $|df_i - df_{i+1}| < number$ (or if the SPCONV= criterion is satisfied), then convergence is declared. *number* must be between 0 and 1; by default, *number* is 1000 times the square root of machine epsilon, which is about 1E–5.

FINITE modifies the computations for the finite-sample case of the PFDR option, described on page 4194.

MAXITER=*n* specifies the maximum number of golden-search iterations used to find a spline with DF=*df* degrees of freedom. By default, MAXITER= 100; *number* must be a nonnegative integer.

NLAMBDA=*n* computes $\hat{\pi}_0(\lambda)$ for $\lambda \in \{0, \frac{1}{n}, \frac{2}{n}, \ldots, \frac{n-1}{n}\}$ for the spline fit. By default, NLAMBDA= 20; *number* must be an integer greater than 1.

SPCONV=*number* specifies the absolute change in smoothing parameter value for concluding convergence of the spline. If $|sp_i - sp_{i+1}| < number$ (or if the DFCONV= criterion is satisfied), then convergence is declared. By default, *number* equals the square root of the machine epsilon, which is about 1E–8.

In all cases $\hat{m}_0$ is constrained to lie between 0 and m; if the computed $\hat{m}_0 = 0$, then the adaptive adjustments do not produce results. If you specify $\hat{m}_0 > m$, then it is reduced to m. Values of $\hat{m}_0$ are displayed in the "Estimated Number of True Null Hypotheses" table.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of the CLASS variable. By default, ORDER=FORMATTED. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine dependent. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option can be useful when you use CONTRAST statements.

When ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. The following table shows how PROC MULTTEST interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

For more information about sorting order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

OUT=*SAS-data-set*

names the output SAS data set containing variable names, contrast names, intermediate calculations, and all associated p-values. See "OUT= Data Set" on page 4220 for more information.

OUTPERM=*SAS-data-set*

names the output SAS data set containing entire permutation distributions (upper-tail probabilities) for all tests when the PERMUTATION= option is specified. See "OUTPERM= Data Set" on page 4221 for more information. **CAUTION:** This data set can be very large.

OUTSAMP=*SAS-data-set*

names the output SAS data set containing information from the resampled data sets when resampling is performed. See "OUTSAMP= Data Set" on page 4222 for more information. **CAUTION:** This data set can be very large.

PDATA=*SAS-data-set*

is an alias for the INPVALUES= option.

PERMUTATION

PERM

computes adjusted p-values in identical fashion as the BOOTSTRAP option, with the exception that PROC MULTTEST resamples without replacement rather than with replacement. Resampling is performed independently within levels of the STRATA variable. Continuous variables are not mean-centered prior to resampling; specify the CENTER to change this. See the section "Bootstrap" on page 4213 for more details. The PERMUTATION option is not allowed with the Peto test.

PFDR<(*options***)>**

computes the "q-values" $\hat{q}_\lambda(p_i)$ of Storey (2002) and Storey, Taylor, and Siegmund (2004). PROC MULTTEST treats these "q-values" as adjusted p-values. The computations depend on selecting a parameter λ and an estimation method for the false discovery rate; see the section "Positive False Discovery Rate" on page 4218 for computational details. The available *options* for choosing the method are as follows:

FINITE estimates the false discovery rate with $\widehat{\text{pFDR}}$ or $\widehat{\text{FDR}}$ for the finite-sample case with independent null p-values.

POSITIVE estimates the false discovery rate with $\widehat{\text{pFDR}}$ instead of the default $\widehat{\text{FDR}}$.

The available options for controlling the λ search are the *bootstrap-options* (page 4192), the *spline-options* (page 4193), and the following options:

LAMBDA=*number* specifies a $\lambda \in [0, 1)$ and does not perform the bootstrap or spline searches for an "optimal" λ.

MAXLAMBDA=*number* stops the NLAMBDA= search sequence for the bootstrap and spline searches when this *number* is reached. The *number* must be in $[0, 1]$. This option is ignored if the LAMBDA= option is specified.

PLOTS<(*global-plot-options***)>=***plot-request*<*(options)*>

PLOTS<(*global-plot-options***)>=(***plot-request*<*(options)*>< . . . *plot-request*<*(options)*>>**)**

controls the plots produced through ODS Graphics. If you specify only one *plot-request*, you can omit the parentheses. For example, the following statements are valid specifications of the PLOTS= option:

```
plots = all
plots = (rawprob adjusted)
plots(sigonly) = (rawprob adjusted(unpack))
```

In order to produce plots, you must enable ODS Graphics and specify a *plot-request*, as shown in the following example. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

```
ods graphics on;
proc multtest plots=adjusted inpvalues=a pfdr;
run;
ods graphics off;
```

You need at least two tests to produce a graph. If you are not using an INPVALUES= data set, then each test is given a name constructed as "variable-name contrast-label". If you specify a MEAN test in the TEST statement, the *t*-test names are prefixed with "Mean:". See Example 58.6 for examples of the ODS graphical displays.

The following *global-plot-options* are available:

UNPACKPANELS | UNPACK suppresses paneling. By default, the plots produced with the ADJUSTED and RAWPROB options are grouped in a single display, called a *panel*. Specify UNPACK to display each plot separately.

SIGONLY<=*number***>** displays only those tests with adjusted *p*-values $\leq$ *number*, where $0 \leq number \leq 1$. By default, $number = 0.05$.

The following *plot-requests* are available:

ADJUSTED<(UNPACK)> displays a 2×2 panel of adjusted *p*-value plots similar to those Storey and Tibshirani (2003) developed for use with the PFDR *p*-value adjustment method. The plots of the adjusted *p*-values by the raw *p*-values and the adjusted *p*-values by their rank show the effect of the adjustments. The plot of the proportion of adjusted *p*-values $\leq$ each adjusted *p*-value and the plot of the expected number

of false positives (the proportion significant multiplied by the adjusted *p*-value) versus the proportion significant show the effect of choosing different significance levels. The UNPACK option unpanels the display.

ALL produces all appropriate plots. You can specify other options with ALL; for example, to display all plots and unpack the RAWPROB plots you can specify `plots=(all rawprob(unpack))`.

LAMBDA displays plots of the MSE and the estimated number of true nulls against the λ parameter when the NTRUENULL=SPLINE or NTRUENULL=BOOTSTRAP option is in effect.

NONE suppresses all plots.

PBYTEST<(*options***)>** displays the adjusted *p*-values for each test. The available options are as follows:

NOTESTNAME displays the number of the test instead of the test name on the axis, which is useful when you have many tests.

VREF=*number-list* displays reference lines at the *p*-values specified in the *number-list*. The values in the *number-list* must be between 0 and 1; otherwise they are ignored. You can specify a single value or a list of values; for example, `vref=0.1 0 to 0.05 by 0.01` displays reference lines at each of the values {0.01, 0.02, 0.03, 0.04, 0.05, and 0.1}.

RAWPROB<(UNPACK)> displays a uniform probability plot of 1 minus the raw *p*-values (Schweder and Spjøtvoll 1982) along with a histogram. If m_0 is the number of true null hypotheses among the m tests, the points on the left side of the plot should be approximately linear with slope $\frac{1}{m_0+1}$. This graphic is displayed when an adaptive *p*-value adjustment method is requested in order to see if the NTRUENULL= estimate is appropriate. The UNPACK option unpanels the display.

PTRUENULL=*keyword* | *value*

PI0=*keyword* | *value*

is alias for the NTRUENULL= option, except that you can specify the proportion of true null hypotheses as a *value* between 0 and 1, instead of specifying the number of true null hypotheses. The available *keywords* are also the NTRUENULL= options described on page 4191.

RANUNI

requests the random number generator used in releases prior to SAS 9.2. Beginning with SAS 9.2, the random number generator is the Mersenne Twister, which has better performance when bootstrapping. Changes in the bootstrap- or permutation-adjusted *p*-values from prior releases are due to unimportant sampling differences.

SEED=*number*

S=*number*

specifies the initial seed for the random number generator used for resampling. The value for *number* must be an integer. If you do not specify a seed, or if you specify a value less than or equal to zero, then PROC MULTTEST uses the time of day from the computer's clock to generate an initial seed. For more details about seed values, see *SAS Language Reference: Concepts*.

SIDAK

SID

computes the Šidák adjustment for each test. These adjustments take the form

$$1 - (1 - p)^m$$

where p is the raw p-value and m is the number of tests. These are slightly less conservative than the Bonferroni adjustments, but they still should be viewed with caution. When exact tests are specified via the PERMUTATION= option in the TEST statement, the actual permutation distributions are used, resulting in a much less conservative version of this procedure (Westfall and Wolfinger 1997). See the section "Šidák" on page 4213 for more details.

STEPBON

requests adjusted p-values by using the step-down Bonferroni method of Holm (1979). See the section "Step-Down Methods" on page 4214 for more details.

STEPBOOT

requests that adjusted p-values be computed by using bootstrap resampling as described under the BOOTSTRAP option, but in step-down fashion. See the section "Step-Down Methods" on page 4214 for more details.

STEPPERM

requests that adjusted p-values be computed by using permutation resampling as described under the PERMUTATION option, but in step-down fashion. See the section "Step-Down Methods" on page 4214 for more details.

STEPSID

requests adjusted p-values by using the Šidák method as described in the SIDAK option, but in step-down fashion. See the section "Step-Down Methods" on page 4214 for more details.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC MULTTEST to obtain separate analyses on observations in groups defined by the BY variables. The *variables* are one or more variables in the input data set. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by specifying the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the MULTTEST procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.

- Create an index on the BY variables specifying the DATASETS procedure (in Base SAS software).

Since sorting the data changes the order in which PROC MULTTEST reads observations, this can affect the sorting order for the levels of the CLASS variable if you have specified ORDER=DATA in the PROC MULTTEST statement. This, in turn, affects specifications in the CONTRAST statements.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variable* **;**

The CLASS statement is required unless the INPVALUES= option is specified. The CLASS statement specifies a single variable (character or numeric) used to identify the groups for the analysis. For example, if the variable Treatment defines different levels of a treatment that you want to compare, then you would specify the following statements:

```
class Treatment;
```

The CLASS variable can be either character or numeric. By default, its levels are determined from entire formatted values. Prior to SAS 9, class levels were determined by using no more than the first 16 characters of the formatted values. If you want to revert to this previous behavior, you can specify the TRUNCATE option in the CLASS statement. In any case, you can use formats to group values into levels. See the discussion of the FORMAT procedure in the *Base SAS Procedures Guide* and the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*. You can adjust the order of CLASS variable levels with the ORDER= option in the PROC MULTTEST statement. You need to be aware of the order when using the CONTRAST statement, and you should check the "Contrast Coefficients" table to verify that it is suitable.

You can specify the following option in the CLASS statement after a slash(/):

TRUNCATE

specifies that class levels should be determined by using no more than the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases prior to SAS 9.

The order of the class levels used by PROC MULTTEST corresponds to the order of their formatted values; this order can be changed with the ORDER= option in the PROC MULTTEST statement.

CONTRAST Statement

CONTRAST *'label' values* **;**

This statement is used to identify tests between the levels of the CLASS variable; in particular, it is used to specify the coefficients for the trend tests. The *label* is a string naming the contrast; it contains a maximum of 21 characters. The *values* are scoring coefficients across the CLASS variable levels.

You can specify multiple CONTRAST statements, thereby specifying multiple contrasts for each variable. Multiplicity adjustments are computed for all contrasts and all variables simultaneously. The coefficients are applied to the ordered CLASS variables; this order can be changed with the ORDER= option in the PROC MULTTEST statement. For example, consider a four-group experiment with CLASS variable levels A1, A2, B1, and B2 denoting two levels of two treatments. The following statements produce three linear trend tests for each variable identified in the TEST statement. PROC MULTTEST computes the multiplicity adjustments over the entire collection of tests, which is three times the number of variables.

```
contrast 'a vs b'    -1 -1  1  1;
contrast 'a linear'  -1  1  0  0;
contrast 'b linear'   0  0 -1  1;
```

As another example, consider an animal carcinogenicity experiment with dose levels 0, 4, 8, 16, and 50. You can specify a trend test with the indicated scoring coefficients by using the following statement:

```
contrast 'arithmetic trend' 0 4 8 16 50;
```

Multiplicity-adjusted *p*-values are then computed over the collection of variables identified in the TEST statement. See Lagakos and Louis (1985) for guidelines on the selection of contrast-scoring values.

When a Fisher test is specified in the TEST statement, the CONTRAST statement coefficients are used to group the CLASS variable's levels. Groups with a -1 contrast coefficient are combined and compared with groups with a 1 contrast coefficient for each test, and groups with a 0 coefficient are not included in the contrast. For example, the following statements compute Fisher exact tests for (a) control versus the combined treatment groups, (b) control versus the first treatment group, and (c) control versus the third treatment group.

```
contrast 'c vs all' 1 -1 -1 -1;
contrast 'c vs t1'  1 -1  0  0;
contrast 'c vs t3'  1  0  0 -1;
```

Multiplicity adjustments are then computed over the entire collection of tests and variables. Only -1, 1, and 0 are acceptable CONTRAST coefficients when the Fisher test is specified; PROC MULTTEST ignores the CONTRAST statement if any other coefficients appear.

If you specify the FISHER test and no CONTRAST statements, then all contrasts of control versus treatment are automatically generated, with the first level of the CLASS variable deemed to be

the control. In this case, the control level is assigned the value 1 in each contrast and the other treatment levels are assigned −1. You should therefore use the LOWERTAILED option to test for higher success rates in the treatment groups.

For tests other than FISHER, CONTRAST values are 0, 1, 2, . . . by default. If you specify the CA or PETO test with the PERMUTATION= option, then your CONTRAST coefficients must be integer valued.

For *t* tests for the mean of continuous data (and for the FT tests), the contrast coefficients are centered to have mean = 0. The resulting centered scoring coefficients are then applied to the sample means (or to the double-arcsine-transformed proportions in the case of the FT tests).

FREQ Statement

FREQ *variable* **;**

The FREQ statement names a variable that provides frequencies for each observation in the DATA= data set. Specifically, if *n* is the value of the FREQ variable for a given observation, then that observation is used *n* times.

If the value of the FREQ variable is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

STRATA Statement

STRATA *variable* **;**

The STRATA statement identifies a single variable to use as a stratification variable in the analysis. This yields tests similar to those discussed in Mantel and Haenszel (1959) and Hoel and Walburg (1972) for binary data and pooled-means tests for continuous data. For example, when you test for prevalence in a carcinogenicity study, it is common to stratify on intervals of the time of death; the first level of the stratification variable might represent weeks 0−52, the second might represent weeks 53−80, and so on. In multicenter clinical studies, each level of the stratification variable might represent a particular center.

The following option is available in the STRATA statement after a slash (/):

WEIGHT=*keyword*

specifies the type of strata weighting to use when computing the Freeman-Tukey and *t* tests. Valid *keywords* are SAMPLESIZE, HARMONIC, and EQUAL. SAMPLESIZE requests weights proportional to the within-stratum sample sizes, and is the default method even if the WEIGHT= option is not specified. HARMONIC sets up weights equal to the harmonic mean of the nonmissing within-stratum CLASS sizes, and is similar to a Type 2 analysis in PROC GLM. EQUAL specifies equal weights, and is similar to a Type 3 analysis in PROC GLM.

TEST Statement

TEST *name (variables </ options>)* **;**

The TEST statement is required unless the INPVALUES= option is specified. The TEST statement identifies statistical tests to be performed and the discrete and continuous variables to be tested. The following tests are permitted as *name* in the TEST statement.

CA
requests the Cochran-Armitage linear trend tests for group comparisons. The test variables should take the value 0 for a failure and 1 for a success. PERMUTATION= option can be used to request an exact permutation test; otherwise, a Z-score approximation is used. The CONTINUITY= option can be used to specify a continuity correction for the Z-score approximation.

FISHER
requests Fisher exact tests for comparing two treatment groups. The test variables should take the value 0 for a failure and 1 for a success.

FT
requests Z-score CA tests based upon the Freeman-Tukey double arcsine transformation of the frequencies. The test variables should take the value 0 for a failure and 1 for a success.

MEAN
requests the t test for the mean. The test variables can take on any numeric values.

PETO
requests the Peto mortality-prevalence test. The test variables should take the value 0 for a nonoccurrence, 1 for an incidental occurrence, and 2 for a fatal occurrence. The TIME= option should be used with the Peto test to specify an integer-valued variable giving the age at death. The CONTINUITY= option can be used to specify a continuity correction for the test.

If the value of a TEST variable is invalid, the observation is not used in the analysis. You can specify two tests only if one of them is MEAN. For example, the following statement is valid:

```
test ca(d1-d2) mean(c1-c2);
```

But specifying both CA and FT, as shown in the following statement, is invalid:

```
test ca(d1-d2) ft(d1-d2);
```

You can specify the following options in the TEST statement (some apply to only one test).

BINOMIAL
uses the binomial variance estimate for CA and Peto tests in their asymptotic normal approximations. The default is to use the hypergeometric variance.

CONTINUITY=*number*

C=*number*

specifies *number* as a particular continuity correction for the Z-score approximation in the CA and Peto tests. The default is 0.

LOWERTAILED

LOWER

is used to make all tests lower-tailed. All tests are two-tailed by default.

PERMUTATION=*number*

PERM=*number*

computes p-values for the CA and Peto tests by using exact permutation distributions when marginal success or failure totals within a stratum are *number* or less. You can specify *number* as a nonnegative integer. For totals greater than *number* (or when the PERMUTATION= option is omitted), PROC MULTTEST uses standard normal approximations with a continuity correction chosen to approximate the permutation distribution. PROC MULTTEST computes the appropriate convolution distributions when you use the STRATA statement along with the PERMUTATION= option.

DDFM= POOLED | SATTERTHWAITE

specifies whether the MEAN test uses a homogeneity assumption (DDFM=POOLED, the default) or deals with heterogeneous variances (DDFM=SATTERTHWAITE). See "*t* Test for the Mean" on page 4209 for more information.

TIME=*variable*

identifies the Peto test variable containing the age at death, which must be integer valued. If the TIME= option is omitted, all ages are assumed to equal 1.

UPPERTAILED

UPPER

is used to make all tests upper-tailed. All tests are two-tailed by default.

Details: MULTTEST Procedure

Statistical Tests

The following section discusses the statistical tests performed in the MULTTEST procedure. For continuous data, a t test for the mean (MEAN) is available. For discrete variables, available tests are the Cochran-Armitage linear trend test (CA), the Freeman-Tukey double arcsine test (FT), the Peto mortality-prevalence test (PETO), and the Fisher exact test (FISHER).

Throughout this section, the discrete and continuous variables are denoted by S_{vgsr} and X_{vgsr}, respectively, where v is the variable, g is the treatment group, s is the stratum, and r is the replication. Let m_{vgs} denote the sample size for a binary variable v within group g and stratum s. A plus sign $(+)$ subscript denotes summation over an index. Note that the tests are invariant to the location and scale of the contrast coefficients t_g.

Cochran-Armitage Linear Trend Test

The Cochran-Armitage linear trend test (Cochran 1954; Armitage 1955; Agresti 2002) is implemented by using a Z-score approximation, an exact permutation distribution, or a combination of both.

Z-Score Approximation

The pooled probability estimate for variable v and stratum s is

$$p_{vs} = \frac{S_{v+s+}}{m_{v+s}}$$

The expected value (under constant within-stratum treatment probabilities) for variable v, group g, and stratum s is

$$E_{vgs} = m_{vgs}\, p_{vs}$$

Letting t_g denote the contrast trend coefficients specified by the CONTRAST statement, the test statistic for variable v has numerator

$$N_v = \sum_s \sum_g t_g (S_{vgs+} - E_{vgs})$$

The binomial variance estimate for this statistic is

$$V_v = \sum_s p_{vs}(1 - p_{vs}) \sum_g m_{vgs}(t_g - \bar{t}_{vs})^2$$

where

$$\bar{t}_{vs} = \sum_g \frac{m_{vgs} t_g}{m_{v+s}}$$

The hypergeometric variance estimate (the default) is

$$V_v = \sum_s \{m_{v+s}/(m_{v+s}-1)\} p_{vs}(1-p_{vs}) \sum_g m_{vgs}(t_g - \bar{t}_{vs})^2$$

For any strata s with $m_{v+s} \leq 1$, the contribution to the variance is taken to be zero.

PROC MULTTEST computes the Z-score statistic

$$Z_v = \frac{N_v}{\sqrt{V_v}}$$

The p-value for this statistic comes from the standard normal distribution. Whenever a 0 is computed for the denominator, the p-value is set to 1. This p-value approximates the probability obtained from the exact permutation distribution, discussed in the following text.

The Z-score statistic can be continuity-corrected to better approximate the permutation distribution. With continuity correction c, the upper-tailed p-value is computed from

$$Z_v = \frac{N_v - c}{\sqrt{V_v}}$$

For two-tailed, noncontinuity-corrected tests, PROC MULTTEST reports the p-value as $2\min(p, 1-p)$, where p is the upper-tailed p-value. The same formula holds for the continuity-corrected test, with the exception that when the noncontinuity-corrected Z and the continuity-corrected Z have opposite signs, the two-tailed p-value is 1.

When the PERMUTATION= option is specified and no STRATA variable is specified, PROC MULTTEST uses a continuity correction selected to optimally approximate the upper-tail probability of permutation distributions with smaller marginal totals (Westfall and Lin 1988). Otherwise, the continuity correction is specified by the CONTINUITY= option in the TEST statement.

The CA Z-score statistic is the Hoel-Walburg (Mantel-Haenszel) statistic reported by Dinse (1985).

Exact Permutation Test

When you use the PERMUTATION= option for CA in the TEST statement, PROC MULTTEST computes the exact permutation distribution of the trend score

$$T_v = \sum_s \sum_g t_g S_{vgs+}$$

where the contrast trend coefficients t_g must be integer valued. The observed value of this trend is compared to the permutation distribution to obtain the p-value

$$p_v = \Pr(X \geq \text{ observed } T_v)$$

where X is a random variable from the permutation distribution and where upper-tailed tests are requested. This probability can be viewed as a binomial probability, where the within-stratum probabilities are constant and where the probability is conditional with respect to the marginal totals S_{v+s+}. It also can be considered a rerandomization probability.

Because the computations can be quite time-consuming with large data sets, specifying the PERMUTATION=*number* option in the TEST statement limits the situations where PROC

MULTTEST computes the exact permutation distribution. When marginal total success or total failure frequencies exceed *number* for a particular stratum, the permutation distribution is approximated by a continuity-corrected normal distribution. You should be cautious when using the PERMUTATION= option in conjunction with bootstrap resampling because the permutation distribution is recomputed for each bootstrap sample. This recomputation is not necessary with permutation resampling.

The permutation distribution is computed in two steps:

1. The permutation distributions of the trend scores are computed within each stratum.
2. The distributions are convolved to obtain the distribution of the total trend.

As long as the total success or failure frequency does not exceed *number* for any stratum, the computed distributions are exact. In other words, if $S_{v+s+} \leq$ *number* or $(m_{v+s} - S_{v+s+}) \leq$ *number* for all s, then the permutation trend distribution for variable v is computed exactly.

In step 1, the distribution of the within-stratum trend

$$\sum_g t_g S_{vgs+}$$

is computed by using the multivariate hypergeometric distribution of the S_{vgs+}, provided *number* is not exceeded. This distribution can be written as

$$\Pr(S_{v1s+}, S_{v2s+}, \ldots, S_{vGs+}) = \prod_{g=1}^{G} \frac{\begin{pmatrix} m_{vgs} \\ S_{vgs+} \end{pmatrix}}{\begin{pmatrix} m_{v+s} \\ S_{v+s+} \end{pmatrix}}$$

The distribution of the within-stratum trend is then computed by summing these probabilities over appropriate configurations. For further information about this technique, see Bickis and Krewski (1986) and Westfall and Lin (1988). In step 2, the exact convolution distribution is obtained for the trend statistic summed over all strata having totals that meet the threshold criterion. This distribution is obtained by applying the fast Fourier transform to the exact within-stratum distributions. A description of this general method can be found in Pagano and Tritchler (1983) and Good (1987).

The convolution distribution of the overall trend is then computed by convolving the exact distribution with the distribution of the continuity-corrected standard normal approximation. To be more specific, let S_1 denote the subset of stratum indices that satisfy the threshold criterion, and let S_2 denote the subset of indices that do not satisfy the criterion. Let T_{v1} denote the combined trend statistic from the set S_1, which has an exact distribution obtained from Fourier analysis as previously outlined, and let T_{v1} denote the combined trend statistic from the set S_2. Then the distribution of the overall trend $T_v = T_{v1} + T_{v2}$ is obtained by convolving the analytic distribution of T_{v1} with the continuity-corrected normal approximation for T_{v2}. Using the notation from the section "Z-Score Approximation" on page 4203, this convolution can be written as

$$\begin{aligned} \Pr(T_{v1} + T_{v2} \geq u) &= \sum_{u1} \Pr(T_{v1} + T_{v2} \geq u \mid T_{v1} = u1) \Pr(T_{v1} = u1) \\ &\approx \sum_{u1} \Pr(Z \geq z) \Pr(T_{v1} = u1) \end{aligned}$$

where Z is a standard normal random variable, and

$$z = \frac{1}{\sqrt{V_v}} \left(u - u1 - \sum_{S_2} p_{vs} \sum_g t_g m_{vgs} - c \right)$$

In this expression, the summation of s in V_v is over S_2, and c is the continuity correction discussed under the Z-score approximation.

When a two-tailed test is requested, the expected trend is computed

$$E_v = \sum_s \sum_g t_g E_{vgs}$$

The two-tailed p-value is reported as the permutation tail probability for the observed trend T_v plus the permutation tail probability for $2E_v - T_v$, the reflected trend.

Freeman-Tukey Double Arcsine Test

For this test, the contrast trend coefficients $t_1, \ldots, t_G$ are centered to the values $c_1, \ldots, c_G$, where $c_g = t_g - \bar{t}, \bar{t} = \sum_g t_g / G$, and G is the number of groups. The numerator of this test statistic is

$$N_v = \sum_s w_{vs} \sum_g c_g f(S_{vgs+}, m_{vgs})$$

where the weights w_{vs} take on three different types of values depending upon your specification of the WEIGHT= option in the STRATA statement. The default value is the within-strata sample size m_{v+s}, ensuring comparability with the ordinary CA trend statistic. WEIGHT=HARMONIC sets w_{vs} equal to the harmonic mean

$$\left[\left(\sum_g \frac{1}{m_{vgs}} \right) / G^* \right]^{-1}$$

where G^* is the number of nonmissing groups and the summation is over only the nonmissing elements. The harmonic means analysis places more weight on the smaller sample sizes than does the default sample size method, and is similar to a Type 2 analysis in PROC GLM. WEIGHT=EQUAL sets $w_{vs} = 1$ for all v and s, and is similar to a Type 3 analysis in PROC GLM.

The function $f(r, n)$ is the double arcsine transformation:

$$f(r, n) = \arcsin\left(\sqrt{\frac{r}{n+1}} \right) + \arcsin\left(\sqrt{\frac{r+1}{n+1}} \right)$$

The variance estimate is

$$V_v = \sum_s w_{vs}^2 \sum_g \frac{c_g^2}{m_{vgs} + \frac{1}{2}}$$

The test statistic is

$$Z_v = \frac{N_v}{\sqrt{V_v}}$$

The Freeman-Tukey transformation and its variance are described by Freeman and Tukey (1950) and Miller (1978). Since its variance is not weighted by the pooled probabilities, as is the CA test, the FT test can be more useful than the CA test for tests involving only a subset of the groups.

Peto Mortality-Prevalence Trend Test

The Peto test is a modified Cochran-Armitage procedure incorporating mortality and prevalence information. The Peto test is computed like two Cochran-Armitage Z-score approximations, one for prevalence and one for mortality (Peto, Pike, and Day 1980). It represents a special case in PROC MULTTEST because the data structure requirements are different, and the resampling methods used for adjusting p-values are not valid. The TIME= option variable is required to specify "death" times or, more generally, times of occurrence. In addition, the test variables must assume one of the following three values:

- 0 = no occurrence
- 1 = incidental occurrence
- 2 = fatal occurrence

Use the TIME= option variable to define the mortality strata, and use the STRATA statement variable to define the prevalence strata.

In the following notation, the subscript v represents the variable, g represents the treatment group, s represents the stratum, and t represents the time. Recall that a plus sign $(+)$ in a subscript location denotes summation over that subscript.

Let S^P_{vgs} be the number of incidental occurrences, and let m^P_{vgs} be the total sample size for variable v in group g, stratum s, excluding fatal tumors.

Let S^F_{vgt} be the number of fatal occurrences in time period t, and let m^F_{vgt} be the number of patients alive at the end of time $t-1$.

The pooled probability estimates are given by

$$\begin{aligned} p^P_{vs} &= \frac{S^P_{v+s}}{m^P_{v+s}} \\ p^F_{vt} &= \frac{S^F_{v+t}}{m^F_{v+t}} \end{aligned}$$

The expected values are

$$\begin{aligned} E^P_{vgs} &= m^P_{vgs} p^P_{vs} \\ E^F_{vgt} &= m^F_{vgt} p^F_{vt} \end{aligned}$$

Let t_g denote a contrast trend coefficient, and define the numerator terms as follows:

$$\begin{aligned} N^P_v &= \sum_s \sum_g t_g \left(S^P_{vgs} - E^P_{vgs} \right) \\ N^F_v &= \sum_t \sum_g t_g \left(S^F_{vgt} - E^F_{vgt} \right) \end{aligned}$$

Define the denominator variance terms by using the binomial variance:

$$V_v^P = \sum_s p_{vs}^P \left(1 - p_{vs}^P\right) \left[\left(\sum_g m_{vgs}^P {t_g}^2 \right) - \frac{1}{m_{v+s}^P} \left(\sum_g m_{vgs}^P t_g \right)^2 \right]$$

$$V_v^F = \sum_s p_{vt}^F \left(1 - p_{vt}^F\right) \left[\left(\sum_g m_{vgt}^F {t_g}^2 \right) - \frac{1}{m_{v+t}^F} \left(\sum_g m_{vgt}^F t_g \right)^2 \right]$$

The hypergeometric variances (the default) are calculated by weighting the within-strata variances as discussed in the section "Z-Score Approximation" on page 4203.

The Peto statistic is computed as

$$Z_v = \frac{N_v^P + N_v^F - c}{\sqrt{V_v^P + V_v^F}}$$

where c is a continuity correction. The p-value is determined from the standard normal distribution unless the PERMUTATION=*number* option is used. When you use the PERMUTATION= option for PETO in the TEST statement, PROC MULTTEST computes the "discrete approximation" permutation distribution described by Mantel (1980) and Soper and Tonkonoh (1993). Specifically, the permutation distribution of $\sum_s \sum_g t_g S_{vgs}^P + \sum_t \sum_g t_g S_{vgt}^F$ is computed, assuming that $\{\sum_g t_g S_{vgs}^P\}$ and $\{\sum_g t_g S_{vgt}^F\}$ are independent over all s and t. Note that the contrast trend coefficients t_g must be integer valued. The p-values are exact under this independence assumption. However, the independence assumption is valid only asymptotically, which is why these p-values are called "approximate."

An exact permutation distribution is available only under the assumption of equal risk of censoring in all treatment groups; even then, computing this distribution can be cumbersome. Soper and Tonkonoh (1993) describe situations where the discrete approximation distribution closely fits the exact permutation distribution.

Fisher Exact Test

The CONTRAST statement in PROC MULTTEST enables you to compute Fisher exact tests for two-group comparisons. No stratification variable is allowed for this test. Note, however, that the FISHER exact test is a special case of the exact permutation tests performed by PROC MULTTEST and that these permutation tests allow a stratification variable. Recall that contrast coefficients can be -1, 0, or 1 for the Fisher test. The frequencies and sample sizes of the groups scored as -1 are combined, as are the frequencies and sample sizes of the groups scored as 1. Groups scored as 0 are excluded. The -1 group is then compared with the 1 group by using the Fisher exact test.

Letting x and m denote the frequency and sample size of the 1 group, and letting y and n denote those of the -1 group, the p-value is calculated as

$$\Pr(X \geq x \mid X + Y = x + y) = \sum_{i=x}^{m} \frac{\binom{m}{i}\binom{n}{x+y-i}}{\binom{m+n}{x+y}}$$

where X and Y are independent binomially distributed random variables with sample sizes m and n and common probability parameters. The hypergeometric distribution is used to determine the stated probability; Yates (1984) discusses this technique. PROC MULTTEST computes the two-tailed p-values by adding probabilities from both tails of the hypergeometric distribution. The first tail is from the observed x and y, and the other tail is chosen so that the resulting probability is as large as possible without exceeding the probability from the first tail.

t Test for the Mean

For continuous variables, PROC MULTTEST automatically centers the contrast trend coefficients, as in the Freeman-Tukey test. These centered coefficients c_g are then used to form a t statistic contrasting the within-group means. Let n_{vgs} denote the sample size within group g and stratum s; it depends on variable v only when there are missing values. Determine the weights w_{vs} as in the Freeman-Tukey test with n_{vgs} replacing m_{vgs}. Define

$$\bar{X}_{vgs+} = \frac{1}{n_{vgs}} \sum_{r} X_{vgsr}$$

as the sample mean within a group-and-stratum combination, and let μ_{vgs} denote the treatment means. Write the null hypothesis as

$$\sum_{s} w_{vs} \sum_{g} c_g \mu_{vgs} = 0$$

Also define

$$s_v^2 = \frac{\sum_{s}\sum_{g}\sum_{r} \left(X_{vgsr} - \bar{X}_{vgs+}\right)^2}{\sum_{s}\sum_{g} \left(n_{vgs} - 1\right)}$$

as the pooled sample variance.

Homogeneous Variance

Assuming constant variance for all group-and-stratum combinations, the t statistic for the mean is

$$M_v = \frac{\sum_s w_{vs} \sum_g c_g \bar{X}_{vgs+}}{\sqrt{s_v^2 \left(\sum_s w_{vs}^2 \sum_g \frac{c_g^2}{n_{vgs}} \right)}}$$

Then under the null hypothesis and assuming normality, independence, and homoscedasticity, M_v follows a t distribution with $df_p = \sum_s \sum_g \left(n_{vgs} - 1 \right)$ degrees of freedom.

Whenever a denominator of 0 is computed, the p-value is set to 1. When missing data force $n_{vgs} = 0$, the contribution to the denominator of the pooled variance is 0 and not -1. This is also true for the degrees of freedom.

Heterogeneous Variance

If you do not assume constant variance for all group-and-stratum combinations, then the approximate t test is

$$M_v = \frac{\sum_s w_{vs} \sum_g c_g \bar{X}_{vgs+}}{\sqrt{\sum_s w_{vs}^2 \sum_g c_g^2 \frac{s_{vgs}^2}{n_{vgs}}}}$$

Under the null hypothesis and assuming normality and independence, the Satterthwaite (1946) approximation for the degrees of freedom of the t test is given by

$$df_s = \frac{\left(\sum_s w_{vs}^2 \sum_g c_g^2 \frac{s_{vgs}^2}{n_{vgs}} \right)^2}{\sum_s \sum_g \frac{\left(w_{vs}^2 c_g^2 \frac{s_{vgs}^2}{n_{vgs}} \right)^2}{n_{vgs} - 1}}$$

under the restriction $1 \leq df_s \leq \sum_s \sum_g n_{vgs}$.

Whenever a denominator of 0 for M_v is computed, the p-value is set to 1. If the denominator for df_s is computed as 0, then set $df_s = df_p$. When missing data force $n_{vgs} = 0$, that group-and-stratum combination does not contribute to the df_s computation.

p-Value Adjustments

Suppose you test m null hypotheses, $H_{01}, \ldots, H_{0m}$, and obtain the p-values $p_1, \ldots, p_m$. Denote the ordered p-values as $p_{(1)} \leq \ldots \leq p_{(m)}$ and order the tests appropriately: $H_{0(1)}, \ldots, H_{0(m)}$. Suppose you know m_0 of the null hypotheses are true and $m_1 = m - m_0$ are false. Let R indicate the number of null hypotheses rejected by the tests, where V of these are incorrectly rejected (that is, V tests are Type I errors) and $R - V$ are correctly rejected (so $m_1 - R + V$ tests are Type II errors). This information is summarized in the following table:

	Null Is Rejected	Null Is Not Rejected	Total
Null Is True	V	$m_0 - V$	m_0
Null Is False	$R - V$	$m_1 - R + V$	m_1
Total	R	$m - R$	m

The *familywise error rate* (FWE) is the overall Type I error rate for all the comparisons (possibly under some restrictions); that is, it is the maximum probability of incorrectly rejecting one or more null hypotheses:

$$\text{FWE} = \Pr(V > 0)$$

The FWE is also known as the *maximum experimentwise error rate* (MEER), as discussed in the section "Pairwise Comparisons" on page 2515 of Chapter 39, "The GLM Procedure."

The *false discovery rate* (FDR) is the expected proportion of incorrectly rejected hypotheses among all rejected hypotheses:

$$\begin{aligned}\text{FDR} &= E\left(\frac{V}{R}\right) \quad \text{where } \frac{V}{R} = 0 \text{ when } V = R = 0 \\ &= E\left(\frac{V}{R} \mid R > 0\right) \Pr(R > 0)\end{aligned}$$

Under the overall null hypothesis (all the null hypotheses are true), the FDR=FWE since $V = R$ gives $E\left(\frac{V}{R}\right) = 1 \times \Pr\left(\frac{V}{R} = 1\right) = \Pr(V > 0)$. Otherwise, FDR is always less than FWE, and an FDR-controlling adjustment also controls the FWE. Another definition used is the *positive* false discovery rate:

$$\text{pFDR} = E\left(\frac{V}{R} \mid R > 0\right)$$

The p-value adjustment methods discussed in the following sections attempt to correct the raw p-values while controlling either the FWE or the FDR. Note that the methods might impose some restrictions in order to achieve this; restrictions are discussed along with the methods in the following sections. Discussions and comparisons of some of these methods are given in Dmitrienko et al. (2005), Dudoit, Shaffer, and Boldrick (2003), Westfall et al. (1999), and Brown and Russell (1997).

Familywise Error Rate Controlling Adjustments

PROC MULTTEST provides several p-value adjustments to control the familywise error rate. *Single-step* adjustment methods are computed without reference to the other hypothesis tests under consideration. The available single-step methods are the Bonferroni and Šidák adjustments, which are simple functions of the raw p-values that try to distribute the significance level α across all the tests, and the bootstrap and permutation resampling adjustments, which require the raw data. The Bonferroni and Šidák methods are calculated from the permutation distributions when exact permutation tests are used with the CA or Peto test.

Stepwise tests, or *sequentially rejective* tests, order the hypotheses in *step-up* (least significant to most significant) or *step-down* fashion, then sequentially determine acceptance or rejection of the nulls. These tests are more powerful than the single-step tests, and they do not always require you to perform every test. However, PROC MULTTEST still adjusts every p-value. PROC MULTTEST provides the following stepwise p-value adjustments: step-down Bonferroni (Holm), step-down Šidák, step-down bootstrap and permutation resampling, Hochberg's (1988) step-up, Hommel's (1988), and Fisher's combination method. Adaptive versions of Holm's step-down Bonferroni and Hochberg's step-up Bonferroni methods, which require an estimate of the number of true null hypotheses, are also available.

Liu (1996) shows that all single-step and stepwise tests based on marginal p-values can be used to construct a *closed* test (Marcus, Peritz, and Gabriel 1976; Dmitrienko et al. 2005). Closed testing methods not only control the familywise error rate at size α, but are also more powerful than the tests on which they are based. Westfall and Wolfinger (2000) note that several of the methods available in PROC MULTTEST are closed—namely, the step-down methods, Hommel's method, and Fisher's combination; see that reference for conditions and exceptions.

All methods except the resampling methods are calculated by simple functions of the raw p-values or marginal permutation distributions; the permutation and bootstrap adjustments require the raw data. Because the resampling techniques incorporate distributional and correlational structures, they tend to be less conservative than the other methods.

When a resampling (bootstrap or permutation) method is used with only one test, the adjusted p-value is the bootstrap or permutation p-value for that test, with no adjustment for multiplicity, as described by Westfall and Soper (1994).

Bonferroni

The Bonferroni p-value for test $i, i = 1, \ldots, m$ is simply $\tilde{p}_i = mp_i$. If the adjusted p-value exceeds 1, it is set to 1. The Bonferroni test is conservative but always controls the familywise error rate.

If the unadjusted p-values are computed by using exact permutation distributions, then the Bonferroni adjustment for p_i is $\tilde{p}_i = p_1^* + \cdots + p_m^*$, where p_j^* is the largest p-value from the permutation distribution of test j satisfying $p_j^* \leq p_i$, or 0 if all permutational p-values of test j are greater than p_i. These adjustments are much less conservative than the ordinary Bonferroni adjustments because they incorporate the discrete distributional characteristics. However, they remain conservative in that they do not incorporate correlation structures between multiple contrasts and multiple variables (Westfall and Wolfinger 1997).

Šidák

A technique slightly less conservative than Bonferroni is the Šidák p-value (Šidák 1967), which is $\tilde{p}_i = 1 - (1 - p_i)^m$. It is exact when all of the p-values are uniformly distributed and independent, and it is conservative when the test statistics satisfy the positive orthant dependence condition (Holland and Copenhaver 1987).

If the unadjusted p-values are computed by using exact permutation distributions, then the Šidák adjustment for p_i is $\tilde{p}_i = 1-(1-p_1^*)\cdots(1-p_m^*)$, where the p_j^* are as described previously. These adjustments are less conservative than the corresponding Bonferroni adjustments, but they do not incorporate correlation structures between multiple contrasts and multiple variables (Westfall and Wolfinger 1997).

Bootstrap

The bootstrap method creates pseudo-data sets by sampling observations with replacement from each within-stratum pool of observations. An entire data set is thus created, and p-values for all tests are computed on this pseudo-data set. A counter records whether the minimum p-value from the pseudo-data set is less than or equal to the actual p-value for each base test. (If there are m tests, then there are m such counters.) This process is repeated a large number of times, and the proportion of resampled data sets where the minimum pseudo-p-value is less than or equal to an actual p-value is the adjusted p-value reported by PROC MULTTEST. The algorithms are described in Westfall and Young (1993).

In the case of continuous data, the pooling of the groups is not likely to re-create the shape of the null hypothesis distribution, since the pooled data are likely to be multimodal. For this reason, PROC MULTTEST automatically mean-centers all continuous variables prior to resampling. Such mean-centering is akin to resampling residuals in a regression analysis, as discussed by Freedman (1981). You can specify the NOCENTER option if you do not want to center the data.

The bootstrap method implicitly incorporates all sources of correlation, from both the multiple contrasts and the multivariate structure. The adjusted p-values incorporate all correlations and distributional characteristics. This method always provides weak control of the familywise error rate, and it provides strong control when the *subset pivotality* condition holds; that is, for any subset of the null hypotheses, the joint distribution of the p-values for the subset is identical to that under the complete null (Westfall and Young 1993).

Permutation

The permutation-style-adjusted p-values are computed in identical fashion as the bootstrap-adjusted p-values, with the exception that the within-stratum resampling is performed without replacement instead of with replacement. This produces a rerandomization analysis such as in Brown and Fears (1981) and Heyse and Rom (1988). In the spirit of rerandomization analyses, the continuous variables are not centered prior to resampling. This default can be overridden by using the CENTER option.

The permutation method implicitly incorporates all sources of correlation, from both the multiple contrasts and the multivariate structure. The adjusted p-values incorporate all correlations and distributional characteristics. This method always provides weak control of the familywise error rate, and it provides strong control of the familywise error rate under the *subset pivotality* condition, as described in the preceding section.

Step-Down Methods

Step-down testing is available for the Bonferroni, Šidák, bootstrap, and permutation methods. The benefit of using step-down methods is that the tests are made more powerful (smaller adjusted p-values) while, in most cases, maintaining strong control of the familywise error rate. The step-down method was pioneered by Holm (1979) and further developed by Shaffer (1986), Holland and Copenhaver (1987), and Hochberg and Tamhane (1987).

The Bonferroni step-down (Holm) p-values $\tilde{p}_{(1)}, \ldots, \tilde{p}_{(m)}$ are obtained from

$$\tilde{p}_{(i)} = \begin{cases} m p_{(1)} & \text{for } i = 1 \\ \max\left(\tilde{p}_{(i-1)}, (m-i+1)\, p_{(i)}\right) & \text{for } i = 2, \ldots, m \end{cases}$$

As always, if any adjusted p-value exceeds 1, it is set to 1.

The Šidák step-down p-values are determined similarly:

$$\tilde{p}_{(i)} = \begin{cases} 1 - (1 - p_{(1)})^m & \text{for } i = 1 \\ \max\left(\tilde{p}_{(i-1)}, 1 - (1 - p_{(i)})^{m-i+1}\right) & \text{for } i = 2, \ldots, m \end{cases}$$

Step-down Bonferroni adjustments that use exact tests are defined as

$$\tilde{p}_{(i)} = \begin{cases} p^*_{(1)} + \cdots + p^*_{(m)} & \text{for } i = 1 \\ \max\left(\tilde{p}_{(i-1)}, p^*_{(i)} + \cdots + p^*_{(m)},\right) & \text{for } i = 2, \ldots, m \end{cases}$$

where the p^*_j are defined as before. Note that p^*_j is taken from the permutation distribution corresponding to the jth-smallest unadjusted p-value. Also, any $\tilde{p}_j$ greater than 1.0 is reduced to 1.0.

Step-down Šidák adjustments for exact tests are defined analogously by substituting $1 - (1 - p^*_{(j)}) \cdots (1 - p^*_{(m)})$ for $p^*_{(j)} + \cdots + p^*_{(m)}$.

The resampling-style step-down methods are analogous to the preceding step-down methods; the most extreme p-value is adjusted according to all m tests, the second-most extreme p-value is adjusted according to $(m-1)$ tests, and so on. The difference is that all correlational and distributional characteristics are incorporated when you use resampling methods. More specifically, assuming the same ordering of p-values as discussed previously, the resampling-style step-down-adjusted p-value for test i is the probability that the minimum pseudo-p-value of tests $i, \ldots, m$ is less than or equal to p_i.

This probability is evaluated by using Monte Carlo simulation, as are the previously described resampling-style-adjusted p-values. In fact, the computations for step-down-adjusted p-values are essentially no more time-consuming than the computations for the non-step-down-adjusted p-values. After Monte Carlo, the step-down-adjusted p-values are corrected to ensure monotonicity; this correction leaves the first adjusted p-values alone, then corrects the remaining ones as needed. The step-down method approximately controls the familywise error rate, and it is described in more detail by Westfall and Young (1993), Westfall et al. (1999), and Westfall and Wolfinger (2000).

Hommel

Hommel's (1988) method is a closed testing procedure based on Simes' test (Simes 1986). The Simes p-value for a joint test of any set of S hypotheses with p-values $p_{(1)} \leq p_{(2)} \leq \ldots \leq p_{(S)}$ is $\min((S/1)p_{(1)}, (S/2)p_{(2)}, \ldots, (S/S)p_{(S)})$. The Hommel-adjusted p-value for test j is the maximum of all such Simes p-values, taken over all joint tests that include j as one of their components.

Hochberg-adjusted p-values are always as large or larger than Hommel-adjusted p-values. Sarkar and Chang (1997) showed that Simes' method is valid under independent or positively dependent p-values, so Hommel's and Hochberg's methods are also valid in such cases by the closure principle.

Hochberg

Assuming p-values are independent and uniformly distributed under their respective null hypotheses, Hochberg (1988) demonstrates that Holm's step-down adjustments control the familywise error rate even when calculated in *step-up* fashion. Since the adjusted p-values are uniformly smaller for Hochberg's method than for Holm's method, the Hochberg method is more powerful. However, this improved power comes at the cost of having to make the assumption of independence. Hochberg's method can be derived from Hommel's (Liu 1996), and is thus also derived from Simes' test (Simes 1986).

Hochberg-adjusted p-values are always as large or larger than Hommel-adjusted p-values. Sarkar and Chang (1997) showed that Simes' method is valid under independent or positively dependent p-values, so Hommel's and Hochberg's methods are also valid in such cases by the closure principle.

The Hochberg-adjusted p-values are defined in reverse order of the step-down Bonferroni:

$$\tilde{p}_{(i)} = \begin{cases} p_{(m)} & \text{for } i = m \\ \min\left(\tilde{p}_{(i+1)}, (m-i+1)p_{(i)}\right) & \text{for } i = m-1, \ldots, 1 \end{cases}$$

Fisher Combination

The FISHER_C option requests adjusted p-values by using closed tests, based on the idea of Fisher's combination test. The Fisher combination test for a joint test of any set of S hypotheses with p-values uses the chi-square statistic $\chi^2 = -2\sum \log(p_i)$, with $2S$ degrees of freedom. The FISHER_C adjusted p-value for test j is the maximum of all p-values for the combination tests, taken over all joint tests that include j as one of their components. Independence of p-values is required for the validity of this method.

Adaptive Adjustments

Adaptive adjustments modify the FWE- and FDR-controlling procedures by taking an estimate of the number m_0 or proportion π_0 of true null hypotheses into account. The adjusted p-values for Holm's and Hochberg's methods involve the number of unadjusted p-values larger than (i), $m-i+1$. So the minimal significance level at which the ith ordered p-value is rejected implies that the number of true null hypotheses is $m-i+1$. However, if you know m_0, then you can replace $m-i+1$ with $\min(m_0, m-i+1)$, thereby obtaining more power while maintaining the original α-level significance.

Since m_0 is unknown, there are several methods used to estimate the value—see the NTRUENULL= option for more information. The estimation method described by Hochberg and Benjamini (1990) considers the graph of $1 - p_{(i)}$ versus i, where the $p_{(i)}$ are the ordered p-values of your tests. See Output 58.6.4 for an example. If all null hypotheses are actually true ($m_0 = m$), then the p-values behave like a sample from a uniform distribution and this graph should be a straight line through the origin. However, if points in the upper-right corner of this plot do not follow the initial trend, then some of these null hypotheses are probably false and $0 < m_0 < m$.

The ADAPTIVEHOLM option uses this estimate of m_0 to adjust the step-up Bonferroni method while the ADAPTIVEHOCHBERG option adjusts the step-down Bonferroni method. Both of these methods are due to Hochberg and Benjamini (1990). When m_0 is known, these procedures control the familywise error rate in the same manner as their nonadaptive versions but with more power; however, since m_0 must be estimated, the FWE control is only approximate. The ADAPTIVEFDR and PFDR options also use $\hat{m}_0$, and are described in the following section.

The adjusted p-values for the ADAPTIVEHOLM method are computed by

$$\tilde{p}_{(i)} = \begin{cases} \min(m, \hat{m}_0)\, p_{(1)} & \text{for } i = 1 \\ \max\left[\tilde{p}_{(i-1)}, \min(m - i + 1, \hat{m}_0)\, p_{(i)}\right] & \text{for } i = 2, \ldots, m \end{cases}$$

The adjusted p-values for the ADAPTIVEHOCHBERG method are computed by

$$\tilde{p}_{(i)} = \begin{cases} \min(1, \hat{m}_0)\, p_{(m)} & \text{for } i = m \\ \min\left[\tilde{p}_{(i+1)}, \min(m - i + 1, \hat{m}_0)\, p_{(i)}\right] & \text{for } i = m - 1, \ldots, 1 \end{cases}$$

False Discovery Rate Controlling Adjustments

Methods that control the *false discovery rate* (FDR) were described by Benjamini and Hochberg (1995). These adjustments do not necessarily control the familywise error rate (FWE). However, FDR-controlling methods are more powerful and more liberal, and hence reject more null hypotheses, than adjustments protecting the FWE. FDR-controlling methods are often used when you have a large number of null hypotheses. To control the FDR, Benjamini and Hochberg's (1995) linear step-up method is provided, as well as an adaptive version, a dependence version, and bootstrap and permutation resampling versions. Storey's (2002) pFDR methods are also provided.

The FDR option requests p-values that control the "false discovery rate" described by Benjamini and Hochberg (1995). These *linear step-up* adjustments are potentially much less conservative than the Hochberg adjustments.

The FDR-adjusted p-values are defined in step-up fashion, like the Hochberg adjustments, but with less conservative multipliers:

$$\tilde{p}_{(i)} = \begin{cases} p_{(m)} & \text{for } i = m \\ \min\left(\tilde{p}_{(i+1)}, \frac{m}{i}\, p_{(i)}\right) & \text{for } i = m - 1, \ldots, 1 \end{cases}$$

The FDR method is guaranteed to control the false discovery rate at level $\leq \frac{m_0}{m}\alpha \leq \alpha$ when you have independent p-values that are uniformly distributed under their respective null hypotheses. Benjamini and Yekateuli (2001) show that the false discovery rate is also controlled at level $\leq \frac{m_0}{m}\alpha$

when the *positive regression dependent* condition holds on the set of the true null hypotheses, and they provide several examples where this condition is true.

NOTE: The positive regression dependent condition on the set of the true null hypotheses holds if the joint distribution of the test statistics $\mathbf{X} = (X_1, \ldots, X_m)$ for the null hypotheses $H_{01}, \ldots, H_{0m}$ satisfies: $\Pr(\mathbf{X} \in A | X_i = x)$ is nondecreasing in x for each X_i where H_{0i} is true, for any increasing set A. The set A is increasing if $\boldsymbol{x} \in A$ and $\boldsymbol{y} \geq \boldsymbol{x}$ implies $\boldsymbol{y} \in A$.

Dependent False Discovery Rate

The DEPENDENTFDR option requests a false discovery rate controlling method that is always valid for p-values under any kind of dependency (Benjamini and Yekateuli 2001), but is thus quite conservative. Let $\gamma = \sum_{i=1}^{m} \frac{1}{i}$. The DEPENDENTFDR procedure always controls the false discovery rate at level $\leq \frac{m_0}{m}\alpha\gamma$. The adjusted p-values are computed as

$$\tilde{p}_{(i)} = \begin{cases} \gamma p_{(m)} & \text{for } i = m \\ \min\left(\tilde{p}_{(i+1)}, \gamma \frac{m}{i} p_{(i)}\right) & \text{for } i = m-1, \ldots, 1 \end{cases}$$

False Discovery Rate Resampling Adjustments

Bootstrap and permutation resampling methods to control the false discovery rate are available with the FDRBOOT and FDRPERM options (Yekateuli and Benjamini 1999). These methods approximately control the false discovery rate when the *subset pivotality* condition holds, as discussed in the section "Bootstrap" on page 4213, and when the p-values corresponding to the true null hypotheses are independent of those for the false null hypotheses.

The resampling methodology for the BOOTSTRAP and PERMUTATION methods is used to create B resamples. For the bth resample, let $R^b(p)$ denote the number of p-values that are less than or equal to the observed p-value p. Let $r_\beta(p)$ be the $100(1-\beta)$th quantile of $\{R^1(p) \ldots R^b(p) \ldots R^B(p)\}$, and let $r(p)$ be the number of observed p-values less than or equal to p. Compute one of the following estimators:

$$\text{local estimator} \qquad Q_1(p) = \begin{cases} \dfrac{1}{B}\displaystyle\sum_{b=1}^{B} \dfrac{R^b(p)}{R^b(p) + r(p) - pm} & \text{if } r(p) - r_\beta(p) \geq pm \\ \#\{R^b(p) \geq 1\}/B & \text{otherwise} \end{cases}$$

$$\text{upper limit estimator} \qquad Q_\beta(p) = \begin{cases} \displaystyle\sup_{x \in [0,p]} \left(\frac{1}{B}\sum_{b=1}^{B} \frac{R^b(x)}{R^b(x) + r(x) - r_\beta(x)} \right) & \text{if } r(x) - r_\beta(x) \geq 0 \\ \#\{R^b(p) \geq 1\}/B & \text{otherwise} \end{cases}$$

where m is the number of tests and B is the number of resamples. Then for $Q = Q_1$ or Q_β, the adjusted p-values are computed as

$$\tilde{p}_{(i)} = \begin{cases} Q(p_{(m)}) & \text{for } i = m \\ \min\left(\tilde{p}_{(i+1)}, Q(p_{(i)})\right) & \text{for } i = m-1, \ldots, 1 \end{cases}$$

Adaptive False Discovery Rate

Since the FDR method controls the false discovery rate at $\leq \frac{m_0}{m}\alpha \leq \alpha$, knowledge of m_0 allows improvement of the power of the adjustment while still maintaining control of the false discovery rate. The ADAPTIVEFDR option requests adaptive adjusted p-values for approximate control of the false discovery rate, as discussed in Benjamini and Hochberg (2000). See the section "Adaptive Adjustments" on page 4215 for more details. These adaptive adjustments are also defined in step-up fashion but use an estimate $\hat{m}_0$ of the number of true null hypotheses:

$$\tilde{p}_{(i)} = \begin{cases} \frac{\hat{m}_0}{m} p_{(m)} & \text{for } i = m \\ \min\left(\tilde{p}_{(i+1)}, \frac{\hat{m}_0}{i} p_{(i)}\right) & \text{for } i = m-1, \ldots, 1 \end{cases}$$

Since $\hat{m}_0 \leq m$, the larger p-values are adjusted down. This means that controlling the false discovery rate allows you to reject these tests at a level less than the observed p-value. You can modify these results by outputting the raw and adjusted p-values with the OUT= option, then use a DATA step to set $\tilde{p}_i = \max\{\tilde{p}_i, p_i\}$.

To use this adjustment, Benjamini and Hochberg (2000) suggest first specifying the FDR option—if at least one test is rejected at your level, then apply the ADAPTIVEFDR adjustment. Alternatively, Benjamini, Krieger, and Yekutieli (2006) apply the FDR adjustment at level $\frac{\alpha}{\alpha+1}$, then specify the resulting number of true hypotheses with the NTRUENULL= option and apply the ADAPTIVEFDR adjustment; they show that this *two-stage linear step-up* procedure controls the false discovery rate at level α for independent test statistics.

Positive False Discovery Rate

The PFDR option computes the "q-values" $\hat{q}_\lambda(p_i)$ (Storey 2002; Storey, Taylor, and Siegmund 2004), which are adaptive adjusted p-values for strong control of the false discovery rate when the p-values corresponding to the true null hypotheses are independent and uniformly distributed. There are four versions of the PFDR available. Let $N(\lambda)$ be the number of observed p-values that are less than or equal to λ; let m be the number of tests; let $f = 1$ if the FINITE option is specified, and otherwise set $f = 0$; and denote the estimated proportion of true null hypotheses by

$$\hat{\pi}_0(\lambda) = \frac{m - N(\lambda) + f}{(1-\lambda)m}$$

The default estimate of FDR is

$$\widehat{\mathrm{FDR}}_\lambda(p) = \frac{\hat{\pi}_0(\lambda)p}{\max(N(p),1)/m}$$

If you set $\lambda = 0$, then this is identical to the FDR adjustment.

The positive FDR is estimated by

$$\widehat{\mathrm{pFDR}}_\lambda(p) = \frac{\widehat{\mathrm{FDR}}_\lambda(p)}{1-(1-p)^m}$$

The finite-sample versions of these two estimators for independent null p-values are given by

$$\widehat{\mathrm{FDR}}^*_\lambda(p) = \begin{cases} \dfrac{\hat{\pi}^*_0(\lambda)\, p}{\max(N(p),1)/m} & \text{if } p \le \lambda \\ 1 & \text{if } p > \lambda \end{cases}$$

$$\widehat{\mathrm{pFDR}}^*_\lambda(p) = \frac{\widehat{\mathrm{FDR}}^*_\lambda(p)}{1-(1-p)^m}$$

Finally, the adjusted p-values are computed as

$$\tilde{p}_i = \hat{q}_\lambda(p_i) = \inf_{p \ge p_i} \mathrm{FDR}_\lambda(p) \quad i = 1, \ldots, m$$

This method can produce adjusted p-values that are smaller than the raw p-values. This means that controlling the false discovery rate allows you to reject these tests at a level less than the observed p-value. You can modify these results by outputting the raw and adjusted p-values with the OUT= option, then use a DATA step to set $\tilde{p}_i = \max\{\tilde{p}_i, p_i\}$.

Missing Values

If a CLASS or STRATA variable has a missing value, then PROC MULTTEST removes that observation from the analysis.

When there are missing values for test variables, the within-group-and-stratum sample sizes can differ from variable to variable. In most cases this is not a problem; however, it is possible for all data to be missing for a particular group within a particular stratum. For continuous variables and Freeman-Tukey tests, PROC MULTTEST re-centers the contrast trend coefficients within strata where all data for a particular group are missing. Re-centering the MEAN tests could redefine your t tests in an undesirable fashion; for example, if you specify coefficients to contrast the first and third groups (`contrast -1 0 1`) but the third group is missing, then the re-centered coefficients become –0.5 and 0.5, thus contrasting the first and second groups. If this is the case, you can run your t tests in separate PROC MULTTEST invocations, then combine the data and adjust the p-values by using the INPVALUES= option. However, you might find this re-centering acceptable for the Freeman-Tukey trend tests, since the contrast still tests for an increasing trend. The Cochran-Armitage and Peto tests are unaffected by this situation.

PROC MULTTEST uses missing values for resampling if they exist in the original data set. If all variables have missing values for any observation, then PROC MULTTEST removes the observation prior to resampling. Otherwise, PROC MULTTEST treats all missing values as ordinary observations in the resampling. This means that different resampled data sets can have different group sizes. In some cases it means that a resampled data set can have all missing values for a particular variable in a particular group/stratum combination, even when values exist for that combination in the original data. For this reason, PROC MULTTEST recomputes all quantities within each pseudo-data set, including such items as centered scoring coefficients and degrees of freedom for p-values.

While PROC MULTTEST does provide analyses in missing value cases, you should not feel that it completely solves the missing-value problem. If you are concerned about the adverse effects of missing data on a particular analysis, you should consider using imputation and sensitivity analyses to assess the effects of the missing data.

Computational Resources

PROC MULTTEST keeps all of the data in memory to expedite resampling. A large portion of the memory requirement is thus 8*NOBS*NVAR bytes, where NOBS is the number of observations in the data set, and NVAR is the number of variables analyzed, including CLASS, FREQ, and STRATA variables.

If you specify PERMUTATION=*number* (for exact permutation distributions), then PROC MULTTEST requires additional memory. This requirement is approximately 4*NTEST*NSTRATA*CMAX**number**(*number*+1) bytes, where NTEST is the number of contrasts, NSTRATA is the number of STRATA levels, and CMAX is the maximum contrast coefficient.

If you specify the FDRBOOT or FDRPERM option, then saving all the resamples in memory requires 8*NSAMPLE*NOBS bytes, where NSAMPLE is the number of resamples used.

The execution time is linear in the number of resamples; that is, 10,000 resamples will take 10 times longer than 1,000 resamples. The execution time is also linear in the sample size; that is, 100 resamples of size N will take 10 times longer than 100 resamples of size $10N$.

Output Data Sets

OUT= Data Set

The OUT= data set contains contrast names (_test_), variable names (_var_), the contrast label (_contrast_), raw p-values (raw_p or the value specified in the INPVALUES= option), and all requested adjusted p-values (bon_p, sid_p, aholm_p, stpbon_p, stpsid_p, boot_p, perm_p, stpbootp, stppermp, fdrbootp, fdrpermp, ufdbootp, ufdpermp, hoc_p, ahoc_p, fdr_p, afdr_p, dfdr_p, or pfdr_p).

If a resampling-based adjusted p-value is requested, then the simulation standard error is included as either sim_se, stpsimse, fdrsimse, or ufdsimse, depending on whether single-step, step-down, or FDR adjustments are requested. The simulation standard errors are used to bound the true resampling-based adjusted p-value. For example, if the resampling-based estimate is 0.0312 and the simulation standard error is 0.00123, then a 95% confidence interval for the true adjusted p-value is $0.0312 \pm 1.96(0.00123)$, or 0.0288 to 0.0336.

Intermediate statistics used to calculate the p-values are also written to the OUT= data set. The statistics are separated by the _strat_ level. When _strat_ is reported as missing, the statistics refer to the pooled analysis over all _strat_ levels. The p-values are provided only for the pooled analyses and are therefore reported as missing for the strata-specific statistics.

For the Peto test, an additional variable, _tstrat_, is included to indicate whether the stratum is an incidental occurrence stratum (_tstrat_=0) or a fatal occurrence stratum (_tstrat_=1).

The statistic _value_ is the per-strata contribution to the numerator of the overall test statistic. In the case of the MEAN test, this is the contrast function of the sample means multiplied by the total

number of observations within the stratum. For the FT test, _value_ is the contrast function of the double-arcsine transformed proportions, again multiplied by the total number of observations within the stratum. For the CA and Peto tests, _value_ is the observed value of the trend statistic within that stratum.

When either PETO or CA is requested, the variable _exp_ is included; this variable contains the expected value of the trend statistic for the given stratum.

The statistic _se_ is the square root of the variance of the per-strata _value_ statistic for any of the tests.

For MEAN tests, the variable _nval_ is included. When reported with an individual stratum level (that is, when the _strat_ value is nonmissing), the value _nval_ refers to the within-stratum sample size. For the combined analysis (that is, the value of the _strat_ is missing), the value _nval_ contains degrees of freedom of the *t* distribution used to compute the unadjusted *p*-value.

When the FISHER test is requested, the OUT= data set contains the variables _xval_, _mval_, _yval_, and _nval_, which define observations and sample sizes in the two groups defined by the CONTRAST statement.

For example, the OUT= data set from the drug example in the section "Getting Started: MULTTEST Procedure" on page 4184 is displayed in Figure 58.5.

Figure 58.5 Output Data for the MULTTEST Procedure

Obs	_test_	_var_	_contrast_	_value_	_exp_	_se_	raw_p	boot_p	sim_se
1	CA	SideEff1	Trend	8	5	1.54303	0.05187	0.33880	.003346749
2	CA	SideEff2	Trend	7	5	1.54303	0.19492	0.84030	.002590327
3	CA	SideEff3	Trend	10	7	1.63299	0.06619	0.51895	.003532994
4	CA	SideEff4	Trend	10	6	1.60357	0.01262	0.08840	.002007305
5	CA	SideEff5	Trend	7	4	1.44749	0.03821	0.24080	.003023370
6	CA	SideEff6	Trend	9	6	1.60357	0.06137	0.43825	.003508468
7	CA	SideEff7	Trend	9	5	1.54303	0.00953	0.05135	.001560660
8	CA	SideEff8	Trend	8	5	1.54303	0.05187	0.33880	.003346749
9	CA	SideEff9	Trend	7	5	1.54303	0.19492	0.84030	.002590327
10	CA	SideEff10	Trend	8	6	1.60357	0.21232	0.90300	.002092737

OUTPERM= Data Set

The OUTPERM= data set contains contrast names (_contrast_), variable names (_var_), and the associated permutation distributions (_value_ and upper_p). PROC MULTTEST computes the permutation distributions when you use the PERMUTATION= option with the CA or Peto test. The _value_ variable represents the support of the distributions, and upper_p represents their cumulative upper-tail probabilities. The size of this data set depends on the number of variables and the support of their permutation distributions.

For information about how this distribution is computed, see the section "Exact Permutation Test" on page 4204. For an illustration, see Example 58.1.

OUTSAMP= Data Set

The OUTSAMP= data set contains the data sets used in the resampling analysis, if such an analysis is requested. The variable _sample_ indicates the number of the resampled data set. This variable ranges from 1 to the value of the NSAMPLE= option. For each value of the _sample_ variable, an entire resampled data set is included, with _stratum_, _class_, and all other variables in the original data set. The values of the original variables are mean-centered for the mean test, if requested. The variable _obs_ indicates the observation's position in the original data set.

Each new data set is randomly drawn from the original data set, either with (bootstrap) or without (permutation) replacement. The size of this data set is, thus, the number of observations in the original data set times the number of samples.

Displayed Output

The output produced by PROC MULTTEST is divided into several tables. If the DATA= data set is specified, then the following tables are displayed:

- The "Model Information" table provides a list of the options and settings used for that particular invocation of the procedure. This table is not displayed if the INPVALUES= data set is specified. Included in this list are the following items:
  - statistical tests
  - support of the exact permutation distribution for the CA and Peto tests
  - continuity corrections used for the CA test
  - test tails
  - strata adjustment
  - *p*-value adjustments and specified suboptions
  - centering of continuous variables
  - number of samples and seed
- The "Contrast Coefficients" table lists the coefficients used in constructing the statistical tests. These coefficients are either specified in CONTRAST statements or generated by default. The coefficients apply to the levels of the CLASS statement variable. If a MEAN or FT test is specified in the TEST statement, the centered coefficients are displayed. Patterns of missing values in your data set might affect the coefficients used in your analysis; the displayed contrasts take missing value patterns into account. See the section "Missing Values" on page 4219 for more information.
- The "Variable Tabulations" tables provide summary statistics for each variable listed in the TEST statement. Included for discrete variables are the count, sample size, and percentage of occurrences. For continuous variables, the mean, sample standard deviation, and sample size are displayed. All of the previously mentioned statistics are computed for distinct combinations of the CLASS and STRATA statement variables.

If the INPVALUES= data set is specified, then the following tables are displayed:

- The "P-Value Adjustment Information" table provides a list of the specified p-value adjustments. If an adaptive adjustment is specified (see section "Adaptive Adjustments" on page 4215), then the following settings are also displayed when appropriate:
  - whether the finite-sample version of the PFDR is used (FINITE)
  - the number of tuning parameters to check (NLAMBDA=), the maximum tuning parameter (MAXLAMBDA=), or the specified tuning parameter (LAMBDA=)
  - the degrees of freedom of the spline (DF=) and the smoothing parameter
  - the number of bootstrap resamples (NBOOT=) and the seed (SEED=)
- If the bootstrap or spline method for estimating the number of true null hypotheses m_0 is used and the PLOTS= option is specified, the "Lambda Values" table displays the m_0 estimates as a function of the tuning parameter λ. If the bootstrap method is used, the table also displays the mean-squared errors, the minimum of which is used to select a specific λ. This table contains the values used in the "Lambda Functions" plot.
- The "Estimated Number of True Null Hypotheses" table displays the p-value adjustment, the method used to estimate the number of true nulls, and an estimate of the number and proportion of true null hypotheses in the data set.

The following table is displayed unless the NOPVALUE option is specified:

- The "p-Values" table is a collection of the raw and adjusted p-values from the run of PROC MULTTEST. The p-values are identified by variable and test.

ODS Table Names

PROC MULTTEST assigns a name to each table it creates, and you must use this name to reference the table when using the Output Delivery System (ODS). These names are listed in the following table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 58.2 ODS Tables Produced by PROC MULTTEST

ODS Table Name	Description	Statement or Option
Continuous	Continuous variable tabulations	TEST with MEAN
Contrasts	Contrast coefficients	default
Discrete	Discrete variable tabulations	TEST with CA, FT, PETO, or FISHER
LambdaValues	True null estimates	AHOLM, AHOC, AFDR, or PFDR
ModelInfo	Model information	default
NumTrueNull	Estimates of number of true nulls	AHOLM, AHOC, AFDR, or PFDR
pValues	p-values from the tests	default
pValueInfo	p-value adjustment information	INPVALUES=

ODS Graphics

PROC MULTTEST assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 58.3.

To request these graphs you must specify the ODS GRAPHICS statement in addition to the options in the PROC MULTTEST statement as indicated in Table 58.3. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Table 58.3 ODS Graphics Produced by PROC MULTTEST

ODS Graph Name	Plot Description	Option
AdjPlots	Panel of adjusted *p*-value plots	PLOTS=ADJUSTED
AdjByRawRank	Adjusted by rank of raw *p*-values	PLOTS=ADJUSTED(UNPACK)
AdjbyRawP	Adjusted by raw *p*-values	PLOTS=ADJUSTED(UNPACK)
AdjBySignificant	Proportion significant by adjusted	PLOTS=ADJUSTED(UNPACK)
FalsePosBySignificant	Expected number of false positives by proportion significant	PLOTS=ADJUSTED(UNPACK)
PByTest	*p*-values by test	PLOTS=PBYTEST
LambdaPlot	MSE or NTRUENULL by lambda	PLOTS=LAMBDA and (NTRUENULL=BOOTSTRAP or NTRUENULL=SPLINE or PFDR)
RawUniformPlot	Raw *p*-values by rank and histogram	PLOTS=RAWPROB or AHOLM or AHOC or AFDR or PFDR
RawUniformPlot	Raw *p*-values by rank	PLOTS=RAWPROB(UNPACK) and AHOLM or AHOC or AFDR or PFDR
RawUniformHist	Histogram of raw *p*-values	PLOTS=RAWPROB(UNPACK) and AHOLM or AHOC or AFDR or PFDR

Examples: MULTTEST Procedure

Example 58.1: Cochran-Armitage Test with Permutation Resampling

This example, from Keith Soper at Merck, illustrates the exact permutation Cochran-Armitage test carried out on permutation resamples. In the following data set, each observation represents an animal. The binary variables S1 and S2 indicate two tumor types, with 0s indicating no tumor (failure) and 1 indicating a tumor (success); note that they have perfect negative association. The grouping variable is Dose.

```
data a;
   input S1 S2 Dose @@;
   datalines;
0 1 1   1 0 1   0 1 1
0 1 1   0 1 1   1 0 1
1 0 2   1 0 2   0 1 2
1 0 2   0 1 2   1 0 2
1 0 3   1 0 3   1 0 3
0 1 3   0 1 3   1 0 3
;
proc multtest data=a permutation nsample=10000 seed=36607 outperm=pmt;
   test ca(S1 S2 / permutation=10 uppertailed);
   class Dose;
   contrast 'CA Linear Trend' 0 1 2;
run;
proc print data=pmt;
run;
```

The PROC MULTTEST statement requests 10,000 permutation resamples. The OUTPERM= option creates an output SAS data set pmt used for the exact permutation distribution computed for the CA test.

The TEST statement specifies an upper-tailed Cochran-Armitage linear trend test for S1 and S2. The cutoff for exact permutation calculations is 10, as specified with the PERMUTATION= option in the TEST statement. Since S1 and S2 have 10 and 8 successes, respectively, PROC MULTTEST uses exact permutation distributions to compute the *p*-values for both variables.

The groups for the CA test are the levels of Dose from the CLASS statement. The trend coefficients applied to these groups are 0, 1, and 2, respectively, as specified in the CONTRAST statement.

Finally, the PROC PRINT statement displays the SAS data set pmt, which contains the permutation distributions.

The results from this analysis are displayed in Output 58.1.1 through Output 58.1.5. You should check the following "Model Information" table to verify that the analysis specifications are correct.

Output 58.1.1 Cochran-Armitage Test with Permutation Resampling

The Multtest Procedure

Model Information

Test for discrete variables	Cochran-Armitage
Exact permutation distribution used	Everywhere
Tails for discrete tests	Upper-tailed
Strata weights	None
P-value adjustment	Permutation
Number of resamples	10000
Seed	36607

The label and coefficients from the CONTRAST statement are shown in Output 58.1.2.

Output 58.1.2 Contrast Coefficients

Contrast Coefficients

	Dose		
Contrast	1	2	3
CA Linear Trend	0	1	2

Output 58.1.3 displays summary statistics for the two test variables, S1 and S2. The Count column lists the number of successes for each level of the CLASS variable, Dose. The NumObs column lists the sample size, and the Percent column lists the percentage of successes in the sample.

Output 58.1.3 Summary Statistics

Discrete Variable Tabulations

Variable	Dose	Count	NumObs	Percent
S1	1	2	6	33.33
S1	2	4	6	66.67
S1	3	4	6	66.67
S2	1	4	6	66.67
S2	2	2	6	33.33
S2	3	2	6	33.33

The Raw column in Output 58.1.4 contains the p-values from the CA test, and the Permutation column contains the permutation-adjusted p-values.

Output 58.1.4 Resulting *p*-Values

p-Values

Variable	Contrast	Raw	Permutation
S1	CA Linear Trend	0.1993	0.4009
S2	CA Linear Trend	0.9220	1.0000

This table shows that, for S1, the adjusted *p*-value is approximately twice the raw *p*-value. In fact, resamples with small (large) *p*-values for S1 have large (small) *p*-values for S2 due to the perfect negative association of the variables, and hence the permutation-adjusted *p*-value for S1 should be $2 \times 0.1993 = 0.3986$; the difference is due to resampling error. For the same reason, since the raw *p*-value for S2 is 0.9220, the adjusted *p*-value equals 1. The permutation *p*-values for S1 and S2 also happen to be the Bonferroni-adjusted *p*-values for this example.

The OUTPERM= data set is displayed in Output 58.1.5, which contains the exact permutation distributions for S1 and S2 in terms of cumulative probabilities.

Output 58.1.5 Exact Permutation Distribution

Obs	_contrast_	_var_	_value_	upper_p
1	CA Linear Trend	S1	0	1.00000
2	CA Linear Trend	S1	1	1.00000
3	CA Linear Trend	S1	2	1.00000
4	CA Linear Trend	S1	3	1.00000
5	CA Linear Trend	S1	4	1.00000
6	CA Linear Trend	S1	5	0.99966
7	CA Linear Trend	S1	6	0.99609
8	CA Linear Trend	S1	7	0.97827
9	CA Linear Trend	S1	8	0.92205
10	CA Linear Trend	S1	9	0.80070
11	CA Linear Trend	S1	10	0.61011
12	CA Linear Trend	S1	11	0.38989
13	CA Linear Trend	S1	12	0.19930
14	CA Linear Trend	S1	13	0.07795
15	CA Linear Trend	S1	14	0.02173
16	CA Linear Trend	S1	15	0.00391
17	CA Linear Trend	S1	16	0.00034
18	CA Linear Trend	S1	17	0.00000
19	CA Linear Trend	S1	18	0.00000
20	CA Linear Trend	S1	19	0.00000
21	CA Linear Trend	S1	20	0.00000
22	CA Linear Trend	S2	0	1.00000
23	CA Linear Trend	S2	1	1.00000
24	CA Linear Trend	S2	2	1.00000
25	CA Linear Trend	S2	3	0.99966
26	CA Linear Trend	S2	4	0.99609
27	CA Linear Trend	S2	5	0.97827
28	CA Linear Trend	S2	6	0.92205
29	CA Linear Trend	S2	7	0.80070
30	CA Linear Trend	S2	8	0.61011
31	CA Linear Trend	S2	9	0.38989
32	CA Linear Trend	S2	10	0.19930
33	CA Linear Trend	S2	11	0.07795
34	CA Linear Trend	S2	12	0.02173
35	CA Linear Trend	S2	13	0.00391
36	CA Linear Trend	S2	14	0.00034
37	CA Linear Trend	S2	15	0.00000
38	CA Linear Trend	S2	16	0.00000

Example 58.2: Freeman-Tukey and *t* Tests with Bootstrap Resampling

The data for this example are the same as for Example 58.1, except that a continuous variable T, which indicates the time of death of the animal, has been added.

```
data a;
   input S1 S2 T Dose @@;
   datalines;
0 1 104 1   1 0  80 1   0 1 104 1
0 1 104 1   0 1 100 1   1 0 104 1
1 0  85 2   1 0  60 2   0 1  89 2
1 0  96 2   0 1  96 2   1 0  99 2
1 0  60 3   1 0  50 3   1 0  80 3
0 1  98 3   0 1  99 3   1 0  50 3
;
proc multtest data=a bootstrap nsample=10000 seed=37081 outsamp=res;
   test ft(S1 S2 / lowertailed) mean(T / lowertailed);
   class Dose;
   contrast 'Linear Trend' 0 1 2;
run;
proc print data=res(obs=36);
run;
```

The BOOTSTRAP option in the PROC MULTTEST statement requests bootstrap resampling, and NSAMPLE=10000 requests 10,000 bootstrap samples. The SEED=37081 option provides a starting value for the random number generator. The OUTSAMP=res option creates an output SAS data set res containing the 10,000 bootstrap samples.

The TEST statement specifies the Freeman-Tukey test for S1 and S2 and specifies the *t* test for T. Both tests are lower-tailed. The grouping variable in the CLASS statement is Dose, and the coefficients across the levels of Dose are 0, 1, and 2, as specified in the CONTRAST statement. The PROC PRINT statement displays the first 36 observations of the res data set containing the bootstrap samples.

The results from this analysis are listed in Output 58.2.1 through Output 58.2.5.

The "Model Information" table in Output 58.2.1 corresponds to the specifications in the invocation of PROC MULTTEST.

Output 58.2.1 FT and *t* tests with Bootstrap Resampling

The Multtest Procedure

Model Information

Test for discrete variables	Freeman-Tukey
Test for continuous variables	Mean t-test
Degrees of Freedom Method	Pooled
Tails for discrete tests	Lower-tailed
Tails for continuous tests	Lower-tailed
Strata weights	None
P-value adjustment	Bootstrap
Center continuous variables	Yes
Number of resamples	10000
Seed	37081

The "Contrast Coefficients" table in Output 58.2.2 shows the coefficients from the CONTRAST statement after centering, and they model a linear trend.

Output 58.2.2 Contrast Coefficients

Contrast Coefficients

		Dose		
Contrast		1	2	3
Linear Trend	Centered	-1	0	1

The summary statistics are displayed in Output 58.2.3. The values for the discrete variables S1 and S2 are the same as those from Example 58.1. The mean, standard deviation, and sample size for the continuous variable T at each level of Dose are displayed in the "Continuous Variable Tabulations" table.

Output 58.2.3 Summary Statistics

Discrete Variable Tabulations

Variable	Dose	Count	NumObs	Percent
S1	1	2	6	33.33
S1	2	4	6	66.67
S1	3	4	6	66.67
S2	1	4	6	66.67
S2	2	2	6	33.33
S2	3	2	6	33.33

Output 58.2.3 *continued*

Continuous Variable Tabulations

Variable	Dose	NumObs	Mean	Standard Deviation
T	1	6	99.3333	9.6056
T	2	6	87.5000	14.4326
T	3	6	72.8333	22.7017

The *p*-values, displayed in Output 58.2.4, are from the Freeman-Tukey test for S1 and S2, and are from the *t* test for T.

Output 58.2.4 *p*-Values

p-Values

Variable	Contrast	Raw	Bootstrap
S1	Linear Trend	0.8547	1.0000
S2	Linear Trend	0.1453	0.4605
T	Linear Trend	0.0070	0.0281

The Raw column in Output 58.2.4 contains the results from the tests on the original data, while the Bootstrap column contains the bootstrap resampled adjustment to raw_p. Note that the adjusted *p*-values are larger than the raw *p*-values for all three variables. The adjusted *p*-values more accurately reflect the correlation of the raw *p*-values, the small size of the data, and the multiple testing.

Output 58.2.5 displays the first 36 observations of the SAS data set resulting from the OUTSAMP=RES option in the PROC MULTTEST statement. The entire data set has 180,000 observations, which is 10,000 times the number of observations in the data set.

Output 58.2.5 Resampling Data Set

Obs	_sample_	_class_	_obs_	S1	S2	T
1	1	1	17	0	1	26.1667
2	1	1	8	1	0	-27.5000
3	1	1	5	0	1	0.6667
4	1	1	9	0	1	1.5000
5	1	1	7	1	0	-2.5000
6	1	1	3	0	1	4.6667
7	1	2	12	1	0	11.5000
8	1	2	12	1	0	11.5000
9	1	2	14	1	0	-22.8333
10	1	2	17	0	1	26.1667
11	1	2	1	0	1	4.6667
12	1	2	15	1	0	7.1667
13	1	3	4	0	1	4.6667
14	1	3	17	0	1	26.1667
15	1	3	14	1	0	-22.8333
16	1	3	15	1	0	7.1667
17	1	3	15	1	0	7.1667
18	1	3	6	1	0	4.6667
19	2	1	6	1	0	4.6667
20	2	1	17	0	1	26.1667
21	2	1	8	1	0	-27.5000
22	2	1	13	1	0	-12.8333
23	2	1	9	0	1	1.5000
24	2	1	8	1	0	-27.5000
25	2	2	9	0	1	1.5000
26	2	2	18	1	0	-22.8333
27	2	2	15	1	0	7.1667
28	2	2	14	1	0	-22.8333
29	2	2	9	0	1	1.5000
30	2	2	17	0	1	26.1667
31	2	3	16	0	1	25.1667
32	2	3	11	0	1	8.5000
33	2	3	14	1	0	-22.8333
34	2	3	18	1	0	-22.8333
35	2	3	18	1	0	-22.8333
36	2	3	10	1	0	8.5000

The _sample_ variable is the sample indicator and _class_ indicates the resampling group—that is, the level of the CLASS variable Dose assigned to the new observation. The number of the observation in the original data set is represented by _obs_. Also listed are the values of the original test variables, S1 and S2, and the mean-centered values of T.

Example 58.3: Peto Mortality-Prevalence Test

This example illustrates the use of the Peto mortality-prevalence test. The test is a combination of analyses about the prevalence of incidental tumors in the population and mortality due to fatal tumors.

In the following data set, each observation represents an animal. The variables S1−S3 are three tumor types, with a value of 0 indicating no tumor, 1 indicating an incidental (nonlethal) tumor, and 2 indicating a lethal tumor. The time variable T indicates the time of death of the animal, a strata variable B is constructed from T, and the grouping variable Dose is drug dosage.

```
data a;
   input S1-S3 T Dose @@;
   if T<=90 then B=1; else B=2;
   datalines;
0 0 0 104 0   2 0 1  80 0   0 0 1 104 0
0 0 0 104 0   0 2 0 100 0   1 0 0 104 0
2 0 0  85 1   2 1 0  60 1   0 1 0  89 1
2 0 1  96 1   0 0 0  96 1   2 0 1  99 1
2 1 1  60 2   2 0 0  50 2   2 0 1  80 2
0 0 2  98 2   0 0 1  99 2   2 1 1  50 2
;

proc multtest data=a notables out=p stepsid;
   test peto(S1-S3 / permutation=20 time=T uppertailed);
   class Dose;
   strata B;
   contrast 'mort-prev' 0  1  2;
run;
proc print data=p;
run;
```

The NOTABLES option in the PROC MULTTEST statement suppresses the display of the summary statistics for each variable. The OUT= option creates an output SAS data set p containing all *p*-values and intermediate statistics. The STEPSID option is used to adjust the *p*-values.

The TEST statement specifies an upper-tailed Peto test for S1−S3. The mortality strata are defined by TIME=T, the death times. The CLASS statement contains the grouping variable Dose. The prevalence strata are defined by the STRATA statement as the blocking variable B. The CONTRAST statement lists the default linear trend coefficients. The PROC PRINT statement displays the requested *p*-value data set.

The results from this analysis are listed in Output 58.3.1 through Output 58.3.4.

The "Model Information" table in Output 58.3.1 displays information corresponding to the PROC MULTTEST invocation. In this case the totals for all prevalence and fatality strata are less than 20, so exact permutation tests are used everywhere, and the STEPSID adjustments are computed from these permutation distributions.

Output 58.3.1 Peto Test

The Multtest Procedure

Model Information

Test for discrete variables	Peto
Exact permutation distribution used	Everywhere
Tails for discrete tests	Upper-tailed
Strata weights	Sample size
P-value adjustment	Stepdown Sidak

The contrast trend coefficients are listed in Output 58.3.2. They happen to be the same as the levels of the Dose variable.

Output 58.3.2 Contrast Coefficients

Contrast Coefficients

	Dose		
Contrast	0	1	2
mort-prev	0	1	2

In the "*p*-Values" table in Output 58.3.3, the *p*-values for the Peto tests are listed in the Raw column, and the step-down Šidák adjusted *p*-values are in the Stepdown Šidák column.

Output 58.3.3 *p*-Values

p-Values

Variable	Contrast	Raw	Stepdown Sidak
S1	mort-prev	0.0681	0.0814
S2	mort-prev	0.5000	0.5000
S3	mort-prev	0.0363	0.0781

Significant *p*-values in the preceding table support the claim that higher dosage levels lead to higher mortality and prevalence. The raw Peto test is significant at the 5% level for S3, but the adjusted S3 test is no longer significant at 5%. The raw and adjusted *p*-values for S2 are the same because of the step-down technique.

The OUT= data set is displayed in Output 58.3.4.

Output 58.3.4 OUT= Data Set

Obs	_test_	_var_	_contrast_	_strat_	_tstrat_	_value_	_exp_	_se_	raw_p	stpsid_p
1	PETO	S1	mort-prev	1	0	0	0.00000	0.00000	.	.
2	PETO	S1	mort-prev	2	0	0	0.62500	0.85696	.	.
3	PETO	S1	mort-prev	50	1	4	2.00000	1.12022	.	.
4	PETO	S1	mort-prev	60	1	3	1.75000	1.06654	.	.
5	PETO	S1	mort-prev	80	1	2	1.57143	1.04978	.	.
6	PETO	S1	mort-prev	85	1	1	0.75000	0.72169	.	.
7	PETO	S1	mort-prev	96	1	1	0.70000	0.78102	.	.
8	PETO	S1	mort-prev	98	1	0	0.00000	0.00000	.	.
9	PETO	S1	mort-prev	99	1	1	0.42857	0.72843	.	.
10	PETO	S1	mort-prev	100	1	0	0.00000	0.00000	.	.
11	PETO	S2	mort-prev	1	0	6	5.50000	1.05221	.	.
12	PETO	S2	mort-prev	2	0	0	0.00000	0.00000	.	.
13	PETO	S2	mort-prev	50	1	0	0.00000	0.00000	.	.
14	PETO	S2	mort-prev	60	1	0	0.00000	0.00000	.	.
15	PETO	S2	mort-prev	80	1	0	0.00000	0.00000	.	.
16	PETO	S2	mort-prev	85	1	0	0.00000	0.00000	.	.
17	PETO	S2	mort-prev	96	1	0	0.00000	0.00000	.	.
18	PETO	S2	mort-prev	98	1	0	0.00000	0.00000	.	.
19	PETO	S2	mort-prev	99	1	0	0.00000	0.00000	.	.
20	PETO	S2	mort-prev	100	1	0	0.00000	0.00000	.	.
21	PETO	S3	mort-prev	1	0	6	5.50000	1.05221	.	.
22	PETO	S3	mort-prev	2	0	4	2.22222	1.08298	.	.
23	PETO	S3	mort-prev	50	1	0	0.00000	0.00000	.	.
24	PETO	S3	mort-prev	60	1	0	0.00000	0.00000	.	.
25	PETO	S3	mort-prev	80	1	0	0.00000	0.00000	.	.
26	PETO	S3	mort-prev	85	1	0	0.00000	0.00000	.	.
27	PETO	S3	mort-prev	96	1	0	0.00000	0.00000	.	.
28	PETO	S3	mort-prev	98	1	2	0.62500	0.85696	.	.
29	PETO	S3	mort-prev	99	1	0	0.00000	0.00000	.	.
30	PETO	S3	mort-prev	100	1	0	0.00000	0.00000	.	.
31	PETO	S1	mort-prev	.	.	12	7.82500	2.42699	0.06808	0.08140
32	PETO	S2	mort-prev	.	.	6	5.50000	1.05221	0.50000	0.50000
33	PETO	S3	mort-prev	.	.	12	8.34722	1.73619	0.03627	0.07811

The first 30 observations correspond to intermediate statistics used to compute the Peto p-values. The _test_ variable lists the name of the test, the _var_ variable lists the name of the TEST variables, and the _contrast_ variable lists the CONTRAST label. The _strat_ variable lists the level of the STRATA variable, and the _tstrat_ variable indicates whether or not the stratum corresponds to values of the TIME= variable. The _value_ variable is the observed contrast for a stratum, and the _exp_ variable is its expected value. The variable _se_ contains the square root of the variance terms summed to form the denominator of the Peto statistics.

The final three observations correspond to the three Peto tests, with their p-values listed under the raw_p variable. The stpsid_p variable contains the step-down Šidák-adjusted p-values.

Example 58.4: Fisher Test with Permutation Resampling

The following data, from Brown and Fears (1981), are the results of an 80-week carcinogenesis bioassay with female mice. Six tissue sites are examined at necropsy; 1 indicates the presence of a tumor and 0 the absence. A frequency variable Freq is included. A control and four different doses of a drug (in parts per milliliter) make up the levels of the grouping variable Dose.

```
data a;
   input Liver Lung Lymph Cardio Pitui Ovary Freq Dose$ @@;
   datalines;
1 0 0 0 0 0 8  CTRL   0 1 0 0 0 0 7  CTRL   0 0 1 0 0 0 6  CTRL
0 0 0 1 0 0 1  CTRL   0 0 0 0 0 1 2  CTRL   1 1 0 0 0 0 4  CTRL
1 0 1 0 0 0 1  CTRL   1 0 0 0 0 1 1  CTRL   0 1 1 0 0 0 1  CTRL
0 0 0 0 0 0 18 CTRL
1 0 0 0 0 0 9  4PPM   0 1 0 0 0 0 4  4PPM   0 0 1 0 0 0 7  4PPM
0 0 0 1 0 0 1  4PPM   0 0 0 0 1 0 2  4PPM   0 0 0 0 0 1 1  4PPM
1 1 0 0 0 0 4  4PPM   1 0 1 0 0 0 3  4PPM   1 0 0 0 1 0 1  4PPM
0 1 1 0 0 0 1  4PPM   0 1 0 1 0 0 1  4PPM   1 0 1 1 0 0 1  4PPM
0 0 0 0 0 0 15 4PPM
1 0 0 0 0 0 8  8PPM   0 1 0 0 0 0 3  8PPM   0 0 1 0 0 0 6  8PPM
0 0 0 1 0 0 3  8PPM   1 1 0 0 0 0 1  8PPM   1 0 1 0 0 0 2  8PPM
1 0 0 1 0 0 1  8PPM   1 0 0 0 1 0 1  8PPM   1 1 0 1 0 0 2  8PPM
1 1 0 0 0 1 2  8PPM   0 0 0 0 0 0 19 8PPM
1 0 0 0 0 0 4  16PPM  0 1 0 0 0 0 2  16PPM  0 0 1 0 0 0 9  16PPM
0 0 0 0 1 0 1  16PPM  0 0 0 0 0 1 1  16PPM  1 1 0 0 0 0 4  16PPM
1 0 1 0 0 0 1  16PPM  0 1 1 0 0 0 1  16PPM  0 1 0 1 0 0 1  16PPM
0 1 0 0 0 1 1  16PPM  0 0 1 1 0 0 1  16PPM  0 0 1 0 1 0 1  16PPM
1 1 1 0 0 0 2  16PPM  0 0 0 0 0 0 14 16PPM
1 0 0 0 0 0 8  50PPM  0 1 0 0 0 0 4  50PPM  0 0 1 0 0 0 8  50PPM
0 0 0 1 0 0 1  50PPM  0 0 0 0 0 1 4  50PPM  1 1 0 0 0 0 3  50PPM
1 0 1 0 0 0 1  50PPM  0 1 1 0 0 0 1  50PPM  0 1 0 0 1 1 1  50PPM
0 0 0 0 0 0 19 50PPM
;
proc multtest data=a order=data notables out=p
              permutation nsample=1000 seed=764511;
   test fisher(Liver Lung Lymph Cardio Pitui Ovary /
               lowertailed);
   class Dose;
   freq Freq;
run;
proc print data=p;
run;
```

In the PROC MULTTEST statement, the ORDER=DATA option is required to keep the levels of Dose in the order in which they appear in the data set. Without this option, the levels are sorted by their formatted value, resulting in an alphabetic ordering. The NOTABLES option suppresses the display of summary statistics, and the OUT= option produces an output data set p containing the *p*-values. The PERMUTATION option specifies permutation resampling, NSAMPLE=1000 requests 1000 samples, and SEED=764511 option provides a starting value for the random number generator. You should specify a seed if you need to duplicate resampling results.

To test for higher rates of tumor occurrence in the treatment groups compared to the control group, the LOWERTAILED option is specified in the FISHER option of the TEST statement to produce a lower-tailed Fisher exact test for the six tissue sites. The Fisher test is appropriate for comparing a treatment and a control, but multiple testing can be a problem. Brown and Fears (1981) use a multivariate permutation to evaluate the entire collection of tests. PROC MULTTEST adjusts the *p*-values by simulation.

The treatments make up the levels of the grouping variable Dose, listed in the CLASS statement. Since no CONTRAST statement is specified, PROC MULTTEST uses the default pairwise contrasts with the first level of Dose. The FREQ statement is used since these are summary data containing frequency counts of occurrences.

The results from this analysis are listed in Output 58.4.1 through Output 58.4.4. First, the PROC MULTTEST specifications are displayed in Output 58.4.1.

Output 58.4.1 Fisher Test with Permutation Resampling

The Multtest Procedure

Model Information

Test for discrete variables	Fisher
Tails for discrete tests	Lower-tailed
Strata weights	None
P-value adjustment	Permutation
Number of resamples	1000
Seed	764511

The default contrasts for the Fisher test are displayed in Output 58.4.2. Note that each dose is compared with the control.

Output 58.4.2 Default Contrast Coefficients

Contrast Coefficients

	Dose				
Contrast	CTRL	4PPM	8PPM	16PPM	50PPM
CTRL vs. 4PPM	1	-1	0	0	0
CTRL vs. 8PPM	1	0	-1	0	0
CTRL vs. 16PPM	1	0	0	-1	0
CTRL vs. 50PPM	1	0	0	0	-1

The "p-Values" table in Output 58.4.3 displays *p*-values for the Fisher exact tests and their permutation-based adjustments.

Output 58.4.3 *p*-Values

p-Values

Variable	Contrast	Raw	Permutation
Liver	CTRL vs. 4PPM	0.2828	0.9610
Liver	CTRL vs. 8PPM	0.3069	0.9670
Liver	CTRL vs. 16PPM	0.7102	1.0000
Liver	CTRL vs. 50PPM	0.7718	1.0000
Lung	CTRL vs. 4PPM	0.7818	1.0000
Lung	CTRL vs. 8PPM	0.8858	1.0000
Lung	CTRL vs. 16PPM	0.5469	0.9990
Lung	CTRL vs. 50PPM	0.8498	1.0000
Lymph	CTRL vs. 4PPM	0.2423	0.9280
Lymph	CTRL vs. 8PPM	0.5898	1.0000
Lymph	CTRL vs. 16PPM	0.0350	0.2680
Lymph	CTRL vs. 50PPM	0.4161	0.9930
Cardio	CTRL vs. 4PPM	0.3163	0.9710
Cardio	CTRL vs. 8PPM	0.0525	0.3710
Cardio	CTRL vs. 16PPM	0.4506	0.9960
Cardio	CTRL vs. 50PPM	0.7576	1.0000
Pitui	CTRL vs. 4PPM	0.1250	0.7540
Pitui	CTRL vs. 8PPM	0.4948	0.9970
Pitui	CTRL vs. 16PPM	0.2157	0.9080
Pitui	CTRL vs. 50PPM	0.5051	0.9970
Ovary	CTRL vs. 4PPM	0.9437	1.0000
Ovary	CTRL vs. 8PPM	0.8126	1.0000
Ovary	CTRL vs. 16PPM	0.7760	1.0000
Ovary	CTRL vs. 50PPM	0.3689	0.9930

As noted by Brown and Fears, only one of the 24 tests is significant at the 5% level (Lymph, CTRL vs. 16PPM). Brown and Fears report a 12% chance of observing at least one significant raw *p*-value for 16PPM and a 9% chance of observing at least one significant raw *p*-value for Lymph (both at the 5% level). Adjusted *p*-values exhibit much lower chances of false significances. For this example, none of the adjusted *p*-values are close to significant.

The OUT= data set is displayed in Output 58.4.4.

Output 58.4.4 OUT= Data Set

Obs	_test_	_var_	_contrast_	_xval_	_mval_	_yval_	_nval_	raw_p	perm_p	sim_se
1	FISHER	Liver	CTRL vs. 4PPM	14	49	18	50	0.28282	0.961	0.006122
2	FISHER	Liver	CTRL vs. 8PPM	14	49	17	48	0.30688	0.967	0.005649
3	FISHER	Liver	CTRL vs. 16PPM	14	49	11	43	0.71022	1.000	0.000000
4	FISHER	Liver	CTRL vs. 50PPM	14	49	12	50	0.77175	1.000	0.000000
5	FISHER	Lung	CTRL vs. 4PPM	12	49	10	50	0.78180	1.000	0.000000
6	FISHER	Lung	CTRL vs. 8PPM	12	49	8	48	0.88581	1.000	0.000000
7	FISHER	Lung	CTRL vs. 16PPM	12	49	11	43	0.54685	0.999	0.000999
8	FISHER	Lung	CTRL vs. 50PPM	12	49	9	50	0.84978	1.000	0.000000
9	FISHER	Lymph	CTRL vs. 4PPM	8	49	12	50	0.24228	0.928	0.008174
10	FISHER	Lymph	CTRL vs. 8PPM	8	49	8	48	0.58977	1.000	0.000000
11	FISHER	Lymph	CTRL vs. 16PPM	8	49	15	43	0.03498	0.268	0.014006
12	FISHER	Lymph	CTRL vs. 50PPM	8	49	10	50	0.41607	0.993	0.002636
13	FISHER	Cardio	CTRL vs. 4PPM	1	49	3	50	0.31631	0.971	0.005307
14	FISHER	Cardio	CTRL vs. 8PPM	1	49	6	48	0.05254	0.371	0.015276
15	FISHER	Cardio	CTRL vs. 16PPM	1	49	2	43	0.45061	0.996	0.001996
16	FISHER	Cardio	CTRL vs. 50PPM	1	49	1	50	0.75758	1.000	0.000000
17	FISHER	Pitui	CTRL vs. 4PPM	0	49	3	50	0.12496	0.754	0.013619
18	FISHER	Pitui	CTRL vs. 8PPM	0	49	1	48	0.49485	0.997	0.001729
19	FISHER	Pitui	CTRL vs. 16PPM	0	49	2	43	0.21572	0.908	0.009140
20	FISHER	Pitui	CTRL vs. 50PPM	0	49	1	50	0.50505	0.997	0.001729
21	FISHER	Ovary	CTRL vs. 4PPM	3	49	1	50	0.94372	1.000	0.000000
22	FISHER	Ovary	CTRL vs. 8PPM	3	49	2	48	0.81260	1.000	0.000000
23	FISHER	Ovary	CTRL vs. 16PPM	3	49	2	43	0.77596	1.000	0.000000
24	FISHER	Ovary	CTRL vs. 50PPM	3	49	5	50	0.36889	0.993	0.002636

The _test_, _var_, and _contrast_ variables provide the TEST name, TEST variable, and CONTRAST label, respectively. The _xval_, _mval_, _yval_, and _nval_ variables contain the components used to compute the Fisher exact tests from the hypergeometric distribution. The raw_p variable contains the *p*-values from the Fisher exact tests, and the perm_p variable contains their permutation-based adjustments. The variable sim_se is the simulation standard error from the permutation resampling.

Example 58.5: Inputting Raw *p*-Values

This example illustrates how to use PROC MULTTEST to multiplicity-adjust a collection of raw *p*-values obtained from some other source. This is a valuable option for those cases where PROC MULTTEST cannot compute the raw *p*-values directly. The data set a, which follows, contains the unadjusted *p*-values computed in Example 58.4. Note that the data set needs to have one variable containing the *p*-values, but the data set can contain other variables as well.

```
data a;
   input Test$ Raw_P @@;
   datalines;
test01  0.28282    test02  0.30688    test03  0.71022
test04  0.77175    test05  0.78180    test06  0.88581
test07  0.54685    test08  0.84978    test09  0.24228
test10  0.58977    test11  0.03498    test12  0.41607
test13  0.31631    test14  0.05254    test15  0.45061
test16  0.75758    test17  0.12496    test18  0.49485
test19  0.21572    test20  0.50505    test21  0.94372
test22  0.81260    test23  0.77596    test24  0.36889
;

proc multtest inpvalues=a holm hoc fdr;
run;
```

Note that the PROC MULTTEST statement is the only statement that can be specified with the *p*-value input mode. In this example, the raw *p*-values are adjusted by the Holm, Hochberg, and FDR methods.

The "P-Value Adjustment Information" table, displayed in Output 58.5.1, provides information about the requested adjustments and replaces the usual "Model Information" table. The adjusted *p*-values are displayed in Output 58.5.2

Output 58.5.1 Inputting Raw *p*-Values

```
                    The Multtest Procedure

              P-Value Adjustment Information

        P-Value Adjustment     Stepdown Bonferroni
        P-Value Adjustment     Hochberg
        P-Value Adjustment     False Discovery Rate
```

Output 58.5.2 *p*-Values

p-Values

Test	Raw	Stepdown Bonferroni	Hochberg	False Discovery Rate
1	0.2828	1.0000	0.9437	0.9243
2	0.3069	1.0000	0.9437	0.9243
3	0.7102	1.0000	0.9437	0.9243
4	0.7718	1.0000	0.9437	0.9243
5	0.7818	1.0000	0.9437	0.9243
6	0.8858	1.0000	0.9437	0.9243
7	0.5469	1.0000	0.9437	0.9243
8	0.8498	1.0000	0.9437	0.9243
9	0.2423	1.0000	0.9437	0.9243
10	0.5898	1.0000	0.9437	0.9243
11	0.0350	0.8395	0.8395	0.6305
12	0.4161	1.0000	0.9437	0.9243
13	0.3163	1.0000	0.9437	0.9243
14	0.0525	1.0000	0.9437	0.6305
15	0.4506	1.0000	0.9437	0.9243
16	0.7576	1.0000	0.9437	0.9243
17	0.1250	1.0000	0.9437	0.9243
18	0.4949	1.0000	0.9437	0.9243
19	0.2157	1.0000	0.9437	0.9243
20	0.5051	1.0000	0.9437	0.9243
21	0.9437	1.0000	0.9437	0.9437
22	0.8126	1.0000	0.9437	0.9243
23	0.7760	1.0000	0.9437	0.9243
24	0.3689	1.0000	0.9437	0.9243

Note that the adjusted *p*-values for the Hochberg method are less than or equal to those for the Holm (Step-down Bonferroni) method. In turn, the adjusted *p*-values for the FDR method (False Discovery Rate) are less than or equal to those for the Hochberg method. These comparisons hold generally for all *p*-value configurations. The FDR method controls the false discovery rate and not the familywise error rate. The Hochberg method controls the familywise error rate under independence. The Holm method controls the familywise error rate without assuming independence.

Example 58.6: Adaptive Adjustments and ODS Graphics

An experiment was performed using Affymetrix® gene chips on the CD4 lymphocyte white blood cells of patients with and without a hereditary allergy (atopy) and possibly with asthma. The Asthma-Atopy microarray data set and analysis are discussed in Gibson and Wolfinger (2004): a one-way ANOVA model of the log2mas5 variable ($\log_2$(MAS 5.0 summary statistics) is fit against a classification variable trt describing different asthma-atopy combinations in the patients, and the least squares means of the trt variable are computed.

For this example, a 1% random sample of least squares means having p-values exceeding 1E–6 is taken. The resulting data are recorded in the test data set, where the Probe_Set_ID variable identifies the probe and the Probt variable contains the p-values for the $m = 121$ tests, as follows:

```
data test;
   length Probe_Set_ID $9.;
   input Probe_Set_ID $ Probt @@;
   datalines;
200973_s_ .963316  201059_at .462754  201563_at .000409  201733_at .000819
201951_at .000252  202944_at .106550  203107_x_ .040396  203372_s_ .010911
203469_s_ .987234  203641_s_ .019296  203795_s_ .002276  204055_s_ .002328
205020_s_ .008628  205199_at .608129  205373_at .005209  205384_at .742381
205428_s_ .870533  205653_at .621671  205686_s_ .396440  205760_s_ .000002
206032_at .024661  206159_at .997627  206223_at .003702  206398_s_ .191682
206623_at .010030  206852_at .000004  207072_at .000214  207371_at .000013
207789_s_ .023623  207861_at .000002  207897_at .000007  208022_s_ .251999
208086_s_ .000361  208406_s_ .040182  208464_at .161468  209055_s_ .529824
209125_at .142276  209369_at .240079  209748_at .071750  209894_at .000042
209906_at .223282  210130_s_ .192187  210199_at .101623  210477_x_ .300038
210491_at .000078  210531_at .000784  210734_x_ .202931  210755_at .009644
210782_x_ .000011  211320_s_ .022896  211329_x_ .486869  211362_s_ .881798
211369_at .000030  211399_at .000008  211572_s_ .269788  211647_x_ .001301
213072_at .005019  213143_at .008711  213238_at .004824  213391_at .316133
213468_at .000172  213636_at .097133  213823_at .001678  213854_at .001921
213976_at .000299  214006_s_ .014616  214063_s_ .000361  214407_x_ .609880
214445_at .000009  214570_x_ .000002  214648_at .001255  214684_at .288156
214991_s_ .006695  215012_at .000499  215117_at .000136  215201_at .045235
215304_at .000816  215342_s_ .973786  215392_at .112937  215557_at .000007
215608_at .006204  215935_at .000027  215980_s_ .037382  216010_x_ .000354
216051_x_ .000003  216086_at .002310  216092_s_ .000056  216511_s_ .294776
216733_s_ .004823  216747_at .002902  216874_at .000117  216969_s_ .001614
217133_x_ .056851  217198_x_ .169196  217557_s_ .002966  217738_at .000005
218601_at .023817  218818_at .027554  219302_s_ .000039  219441_s_ .000172
219574_at .193737  219612_s_ .000075  219697_at .046476  219700_at .003049
219945_at .000066  219964_at .000684  220234_at .130064  220473_s_ .000017
220575_at .030223  220633_s_ .058460  220925_at .252465  221256_s_ .721731
221314_at .002307  221589_s_ .001810  221995_s_ .350859  222071_s_ .000062
222113_s_ .000023  222208_s_ .100961  222303_at .049265  37226_at   .000749
60474_at  .000423
run;
```

The following statements adjust the p-values in the test data set by using the adaptive adjustments (ADAPTIVEHOLM, ADAPTIVEHOCHBERG, ADAPTIVEFDR, and PFDR), which require an estimate of the number of true null hypotheses ($\hat{m}_0$) or proportion of true null hypotheses ($\hat{\pi}_0$). This example illustrates some of the features and graphics for computing and evaluating these estimates. The NOPVALUE option is specified to suppress the display of the "p-Values" table.

```
ods graphics on;
proc multtest inpvalues(Probt)=test plots=all seed=518498000
              aholm ahoc afdr pfdr(positive) nopvalue;
run;
ods graphics off;
```

Output 58.6.1 lists the requested p-value adjustments, along with the selected value of the "Lambda" tuning parameter and the seed (specified with the SEED= option) used in the bootstrap method of estimating the number of true null hypotheses. The "Lambda Values" table lists the estimated number of true nulls for each value of λ, where you can see that the minimum MSE (0.002315) occurs at $\lambda = 0.4$. Output 58.6.2 shows that the SPLINE method failed due to a large slope at $\lambda = 0.95$, so the bootstrap method is used and the MSE plot is displayed.

Output 58.6.1 *p* and Lambda Values

The Multtest Procedure

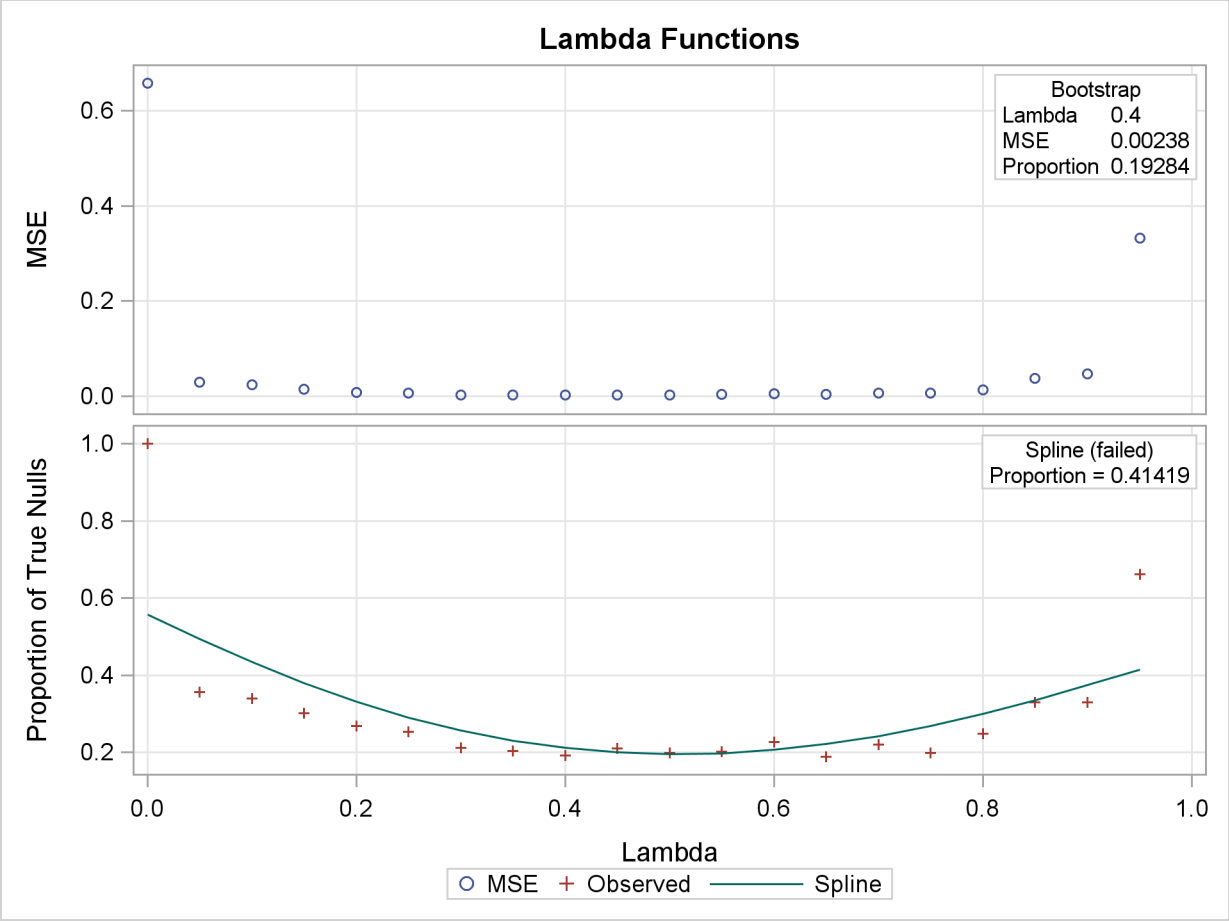
P-Value Adjustment Information

P-Value Adjustment	Adaptive Holm
P-Value Adjustment	Adaptive Hochberg
P-Value Adjustment	Adaptive FDR
P-Value Adjustment	pFDR Q-Value
Lambda	0.4
Seed	518498000

Lambda Values

Lambda	MSE	NTrueNull Observed	NTrueNull Spline
0	0.657880	121.000000	67.318707
0.050000	0.030212	43.157895	59.812885
0.100000	0.024897	41.111111	52.636271
0.150000	0.014904	36.470588	46.033846
0.200000	0.008580	32.500000	40.172642
0.250000	0.006476	30.666667	35.157768
0.300000	0.002719	25.714286	31.046105
0.350000	0.002471	24.615385	27.861153
0.400000	0.002378	23.333333	25.595089
0.450000	0.003285	25.454545	24.217908
0.500000	0.003036	24.000000	23.687690
0.550000	0.003567	24.444444	23.965745
0.600000	0.005813	27.500000	25.016579
0.650000	0.004118	22.857143	26.809774
0.700000	0.006647	26.666667	29.321876
0.750000	0.006260	24.000000	32.512203
0.800000	0.013242	30.000000	36.315191
0.850000	0.037624	40.000000	40.618909
0.900000	0.046906	40.000000	45.274355
0.950000	0.332183	80.000000	50.117369

Output 58.6.2 Tuning Parameter Plots



Output 58.6.3 also shows that the bootstrap estimate is used for the PFDR adjustment. The other adjustments have different default methods for estimating the number of true nulls.

Output 58.6.3 Adjustments and Their Default Estimation Method

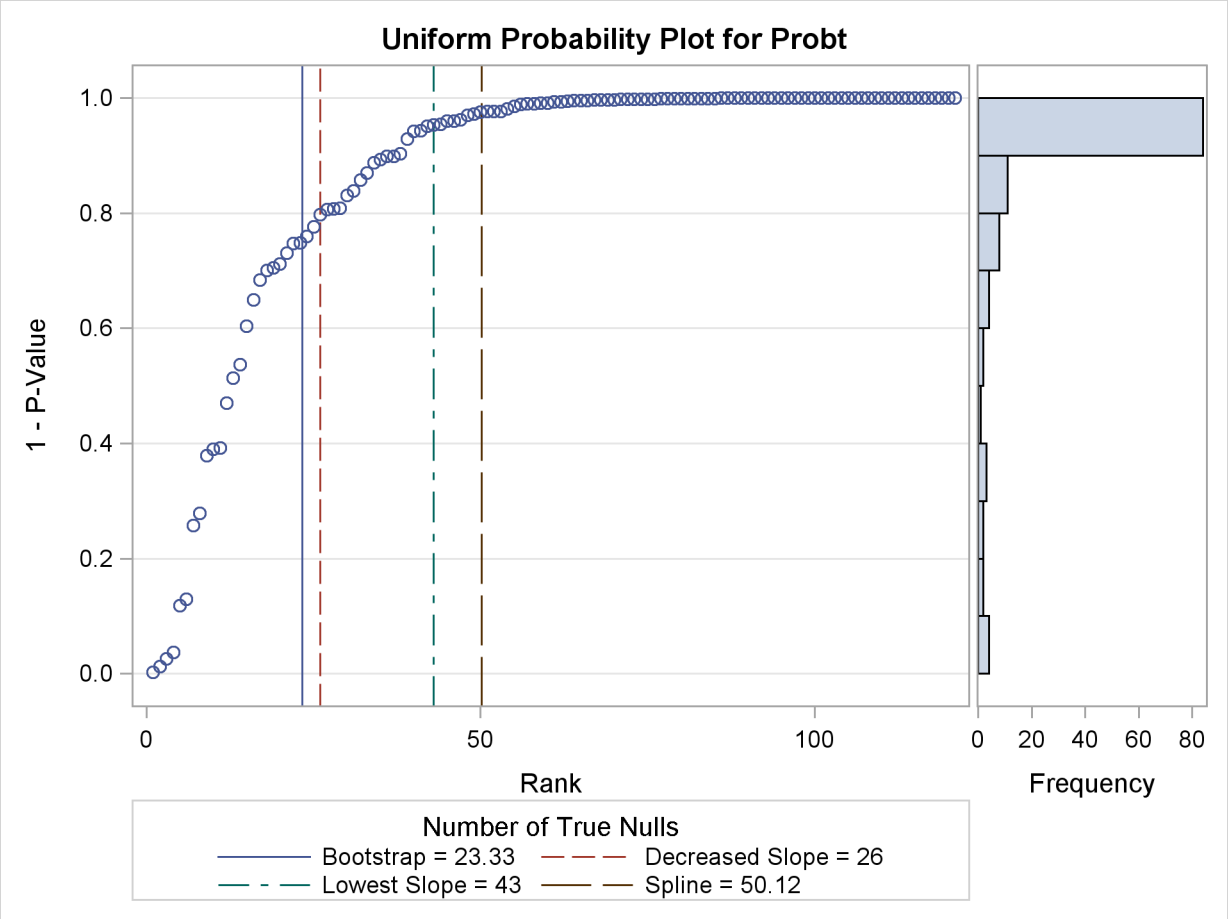
Estimated Number of True Null Hypotheses

P-Value Adjustment	Method	Estimate	Proportion
Adaptive Holm	Decreased Slope	26	0.21488
Adaptive Hochberg	Decreased Slope	26	0.21488
Adaptive FDR	Lowest Slope	43	0.35537
Positive FDR	Bootstrap	23.3333	0.19284

Output 58.6.4 displays the estimated number of true nulls $\hat{m}_0$ against a uniform probability plot of the unadjusted p-values (if the p-values are distributed uniformly, the points on the plot will all lie on a straight line). According to Schweder and Spjøtvoll (1982) and Hochberg and Benjamini

(1990), the points on the left side of the plot should be approximately linear with slope $\frac{1}{m_0+1}$, so you can use this plot to evaluate whether your estimate of $\hat{m}_0$ seems reasonable.

Output 58.6.4 *p*-Value Distribution



The NTRUENULL= option provides several methods for estimating the number of true null hypotheses; the following table displays each method and its estimate for this example:

NTRUENULL=	Estimate
BOOTSTRAP	23.3
DECREASESLOPE	26
KSTEST	35
LEASTSQUARES	28
LOWESTSLOPE	43
MEANDIFF	42
SPLINE	50.1

Another method of estimating the number of true null hypotheses fits a finite mixture model (mixing a uniform with a beta) to the distribution of the unadjusted *p*-values (Allison et al. 2002). Osborne (2006) provides the following PROC NLMIXED statements to fit this model:

```
proc nlmixed data=test;
   parameters pi0=0.5 a=.1 b=.1;
   pi1= 1-pi0;
   bounds 0 <= pi0 <= 1;
   loglikelihood= log(pi0+pi1*pdf('beta',Probt,a,b));
   model Probt ~ general(loglikelihood);
run;
```

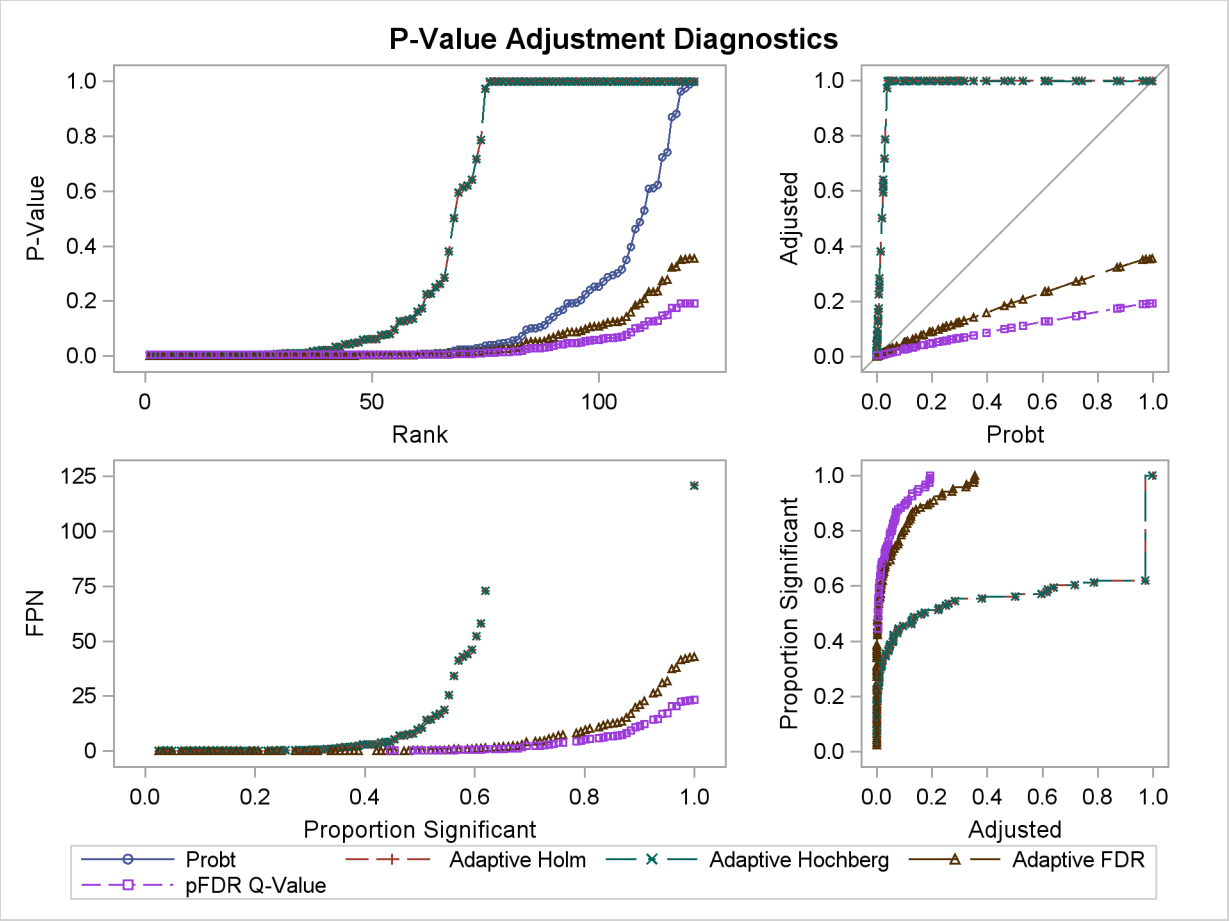
You might have to change the initial parameter values in the PARAMETERS statement to achieve convergence; see Chapter 61, "The NLMIXED Procedure," for more information. This mixture model estimates $\hat{\pi}_0 = 0$, meaning that the distribution of p-values is completely specified by a single beta distribution. If the estimate were, say, $\hat{\pi}_0 = 0.10$, you could then specify it as follows:

```
proc multtest inpvalues(Probt)=test ptruenull=0.10
              aholm ahoc afdr pfdr(positive) nopvalue;
run;
```

A plot of the unadjusted and adjusted p-values for each test is also produced. Due to the large number of tests and adjustments, the plot is not very informative and is not displayed here.

The top two plots in Output 58.6.5 show how the adjusted values compare with each other and the unadjusted p-values. The PFDR and AFDR adjustments are eventually smaller than the unadjusted p-values since they control the false discovery rate. The adaptive Holm and Hochberg adjustments are almost identical, so the adaptive Holm values are mostly obscured in all four plots. The plot of the Proportion Significant versus the Adjusted p-values tells you how many of the tests are significant for each cutoff, while the plot of the number of false positives (FPN) versus the Proportion Significant tells you how many false positives you can expect for that cutoff.

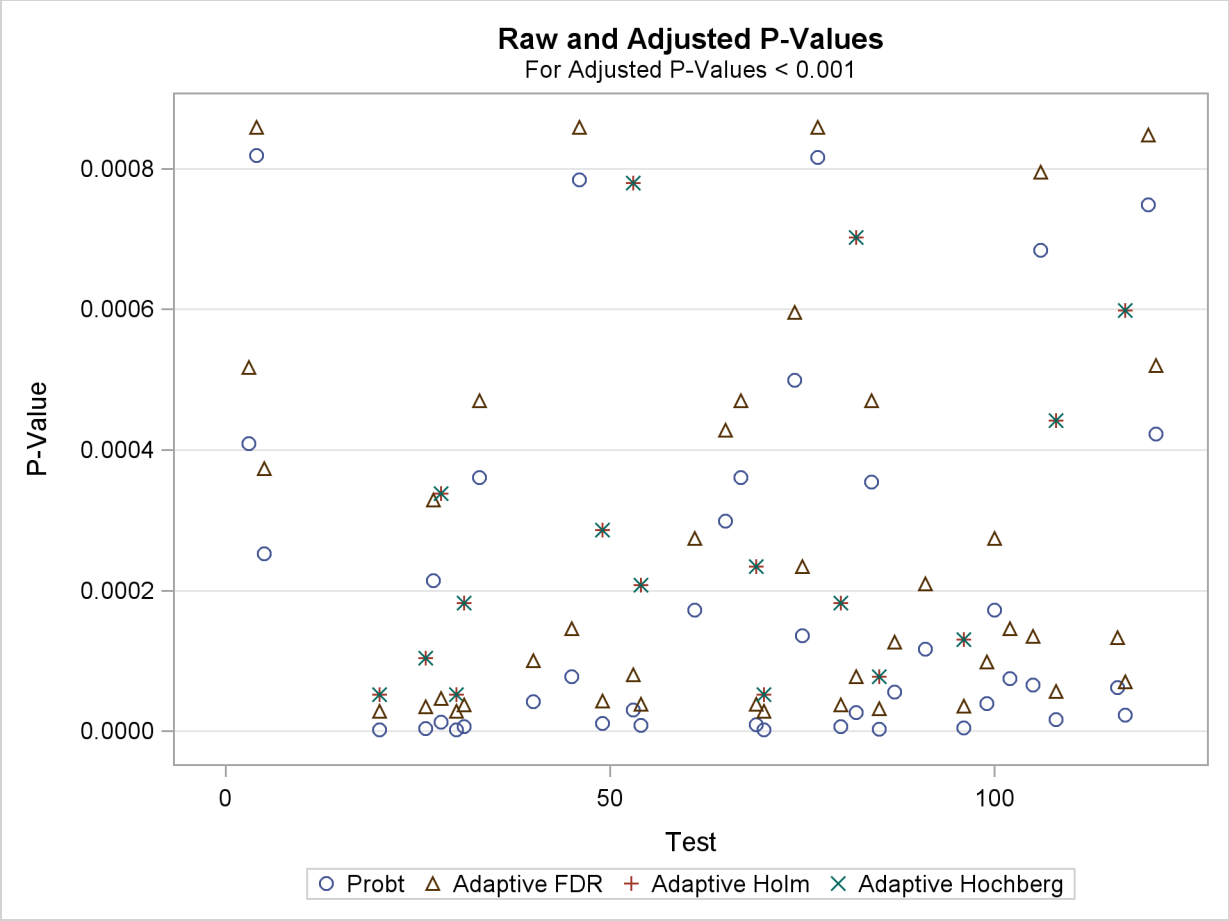
Output 58.6.5 Adjustment Diagnostics



If you have a lot of tests, the "Raw and Adjusted p-Values" and "P-Value Adjustment Diagnostics" plots can be more informative if you suppress some of the tests. In the following statements, the SIGONLY=0.001 option selects tests with adjusted p-values < 0.001 for display. Output 58.6.6 displays tests with their "significant" adjusted p-values.

```
ods graphics on;
proc multtest inpvalues(Probt)=test plots(sigonly=0.001)=PByTest
              aholm ahoc afdr pfdr(positive) nopvalue;
run;
ods graphics off;
```

Output 58.6.6 Raw and Adjusted *p*-Values



References

Agresti, A. (2002), *Categorical Data Analysis*, Second Edition, New York: John Wiley & Sons.

Allison, D. B., Gadbury, G. L., Moonseong, H., Fernández, J. R., Lee, C., Prolla, T. A., and Weindruch, R. (2002), "A Mixture Model Approach for the Analysis of Microarray Gene Expression Data," *Computational Statistics & Data Analysis*, 39, 1–20.

Armitage, P. (1955), "Tests for Linear Trend in Proportions and Frequencies," *Biometrics*, 11, 375–386.

Benjamini, Y. and Hochberg, Y. (1995), "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing," *Journal of the Royal Statistical Society, B*, 57, 289–300.

Benjamini, Y. and Hochberg, Y. (2000), "On the Adaptive Control of the False Discovery Rate in Multiple Testing with Independent Statistics," *Journal of Educational and Behavioral Statistics*, 25, 60–83.

Benjamini, Y., Krieger, A. M., and Yekutieli, D. (2006), "Adaptive Linear Step-up False Discovery Rate Controlling Procedures," *Biometrika*, 93, 491–507.

Benjamini, Y. and Yekateuli, D. (2001), "The Control of the False Discovery Rate in Multiple Testing under Dependency," *Annals of Statistics*, 29, 1165–1188.

Bickis, M. and Krewski, D. (1986), "Statistical Issues in the Analysis of the Long Term Carcinogenicity Bioassay in Small Rodents: An Empirical Evaluation of Statistical Decision Rules," *Environmental Health Directorate*.

Brown, B. W. and Russell, K. (1997), "Methods Correcting for Multiple Testing: Operating Characteristics," *Statistics in Medicine*, 16, 2511–2528.

Brown, C. C. and Fears, T. R. (1981), "Exact Significance Levels for Multiple Binomial Testing with Application to Carcinogenicity Screens," *Biometrics*, 37, 763–774.

Cochran, W. G. (1954), "Some Methods for Strengthening the Common χ^2 Tests," *Biometrics*, 10, 417–451.

Dinse, G. E. (1985), "Testing for Trend in Tumor Prevalence Rates: I. Nonlethal Tumors," *Biometrics*, 41, 751–770.

Dmitrienko, A., Molenberghs, G., Chuang-Stein, C., and Offen, W. (2005), *Analysis of Clinical Trials Using SAS: A Practical Guide*, Cary, NC: SAS Institute Inc.

Dudoit, S., Shaffer, J. P., and Boldrick, J. C. (2003), "Multiple Hypothesis Testing in Microarray Experiments," *Statistical Science*, 18, 71–103.

Freedman, D. A. (1981), "Bootstrapping Regression Models," *Annals of Statistics*, 9, 1218–1228.

Freeman, M. F. and Tukey, J. W. (1950), "Transformations Related to the Angular and the Square Root," *Annals of Mathematical Statistics*, 21, 607–611.

Gibson, G. and Wolfinger, R. D. (2004), "Gene Expression Profiling Using Mixed Models," in A. M. Saxton, ed., *Genetic Analysis of Complex Traits Using SAS*, 251–278, Cary, NC: SAS Publishing.

Good, I. J. (1987), "A Survey of the Use of the Fast Fourier Transform for Computing Distributions," *Journal of Statistical Computation and Simulation*, 28, 87–93.

Heyse, J. and Rom, D. (1988), "Adjusting for Multiplicity of Statistical Tests in the Analysis of Carcinogenicity Studies," *Biometrical Journal*, 30, 883–896.

Hochberg, Y. (1988), "A Sharper Bonferroni Procedure for Multiple Significance Testing," *Biometrika*, 75, 800–803.

Hochberg, Y. and Benjamini, Y. (1990), "More Powerful Procedures for Multiple Significance Testing," *Statistics in Medicine*, 9, 811–818.

Hochberg, Y. and Tamhane, A. C. (1987), *Multiple Comparison Procedures*, New York: John Wiley & Sons.

Hoel, D. G. and Walburg, H. E. (1972), "Statistical Analysis of Survival Experiments," *Journal of the National Cancer Institute*, 49, 361–372.

Holland, B. S. and Copenhaver, M. D. (1987), "An Improved Sequentially Rejective Bonferroni Test Procedure," *Biometrics*, 43, 417–424.

Holm, S. (1979), "A Simple Sequentially Rejective Bonferroni Test Procedure," *Scandinavian Journal of Statistics*, 6, 65–70.

Hommel, G. (1988), "A Comparison of Two Modified Bonferroni Procedures," *Biometrika*, 75, 383–386.

Hsueh, H., Chen, J. J., and Kodell, R. L. (2003), "Comparison of Methods for Estimating the Number of True Null Hypotheses in Multiplicity Testing," *Journal of Biopharmaceutical Statistics*, 13, 675–689.

Lagakos, S. W. and Louis, T. A. (1985), "The Statistical Analysis of Rodent Tumorigenicity Experiments," in D. B. Clayson, D. Krewski, and I. Munro, eds., *Toxicological Risk Assessment*, 144–163, Boca Raton, FL: CRC Press.

Liu, W. (1996), "Multiple Tests of a Non-hierarchical Finite Family of Hypotheses," *Journal of the Royal Statistical Society, Series B*, 58, 455–461.

Mantel, N. (1980), "Assessing Laboratory Evidence for Neoplastic Activity," *Biometrics*, 36, 381–399.

Mantel, N. and Haenszel, W. (1959), "Statistical Aspects of Analysis of Data from Retrospective Studies of Disease," *Journal of the National Cancer Institute*, 22, 719–748.

Marcus, R., Peritz, E., and Gabriel, K. R. (1976), "On Closed Testing Procedures with Special Reference to Ordered Analysis of Variance," *Biometrika*, 63, 655–660.

Miller, J. J. (1978), "The Inverse of the Freeman-Tukey Double Arcsine Transformation," *The American Statistician*, 32, 138.

Osborne, J. A. (2006), "Estimating the False Discovery Rate Using SAS," in *Proceedings of the Thirty-first Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Pagano, M. and Tritchler, D. (1983), "On Obtaining Permutation Distributions in Polynomial Time," *Journal of the American Statistical Association*, 78, 435–440.

Peto, R., Pike, M. C., and Day, N. E. (1980), "Guidelines for Simple, Sensitive Significance Tests for Carcinogenic Effects in Long-Term Animal Experiments," *Long-Term and Short-Term Screening Assays for Carcinogens: A Critical Appraisal*.

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. (1992), *Numerical Recipes in C: The Art of Scientific Computing*, Second Edition, Cambridge, UK: Cambridge University Press.

Sarkar, S. K. and Chang, C.-K. (1997), "The Simes Method for Multiple Hypothesis Testing with Positively Dependent Test Statistics," *Journal of the American Statistical Association*, 92, 1601–1608.

Satterthwaite, F. E. (1946), "An Approximate Distribution of Estimates of Variance Components," *Biometrics Bulletin*, 2, 110–114.

Schweder, T. and Spjøtvoll, E. (1982), "Plots of P-Values to Evaluate Many Tests Simultaneously," *Biometrika*, 69, 493–502.

Shaffer, J. P. (1986), "Modified Sequentially Rejective Multiple Test Procedures," *Journal of the American Statistical Association*, 81, 826–831.

Šidák (1967), "Rectangular Confidence Regions for the Means of Multivariate Normal Distributions," *Journal of the American Statistical Association*, 62, 626–633.

Simes, R. J. (1986), "An Improved Bonferroni Procedure for Multiple Tests of Significance," *Biometrika*, 73, 751–754.

Soper, K. A. and Tonkonoh, N. (1993), "The Discrete Distribution Used for the Log-Rank Test Can Be Inaccurate," *Biometrical Journal*, 35, 291–298.

Storey, J. D. (2002), "A Direct Approach to False Discovery Rates," *JRSS-B*, 64, 479–498.

Storey, J. D., Taylor, J. E., and Siegmund, D. (2004), "Strong Control, Conservative Point Estimation, and Simultaneous Conservative Consistency of False Discovery Rates: A Unified Approach," *JRSS-B*, 66, 187–205.

Storey, J. D. and Tibshirani, R. (2003), "Statistical Significance for Genomewide Studies," in *Proceedings of the National Academy of Sciences of the United States of America*, volume 100, 9440–9445.

Turkheimer, F. E., Smith, C. B., and Schmidt, K. (2001), "Estimation of the Number of 'True' Null Hypotheses in Multivariate Analysis of Neuroimaging Data," *NeuroImage*, 13, 920–930.

Westfall, P. H. and Lin, Y. (1988), "Estimating Optimal Continuity Corrections in Run Time," *Proceedings of the Statistical Computing Section*.

Westfall, P. H. and Soper, K. A. (1994), "Nonstandard Uses of PROC MULTTEST: Permutational Peto Tests; Permutational and Unconditional t and Binomial Tests," in *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Westfall, P. H., Tobias, R. D., Rom, D., Wolfinger, R. D., and Hochberg, Y. (1999), *Multiple Comparisons and Multiple Tests Using the SAS System*, Cary, NC: SAS Institute Inc.

Westfall, P. H. and Wolfinger, R. D. (1997), "Multiple Tests with Discrete Distributions," *The American Statistician*, 51, 3–8.

Westfall, P. H. and Wolfinger, R. D. (2000), "Closed Multiple Testing Procedures and PROC MULTTEST," *Observations*.

Westfall, P. H. and Young, S. S. (1989), "P-Value Adjustments for Multiple Tests in Multivariate Binomial Models," *Journal of the American Statistical Association*, 84, 780–786.

Westfall, P. J. and Young, S. S. (1993), *Resampling-Based Multiple Testing*, New York: John Wiley & Sons.

Yates, F. (1984), "Tests of Significance for 2×2 Contingency Tables," *Journal of the Royal Statistical Society*.

Yekateuli, D. and Benjamini, Y. (1999), "Resampling-Based False Discovery Rate Controlling Multiple Test Procedures for Correlated Test Statistics," *Journal of Statistical Planning and Inference*, 82, 171–196.

Chapter 59

The NESTED Procedure

Contents

Overview: NESTED Procedure **4253**
Contrasted with Other SAS Procedures 4254
Getting Started: NESTED Procedure **4255**
Reliability of Automobile Models 4255
Syntax: NESTED Procedure **4256**
PROC NESTED Statement 4257
BY Statement 4257
CLASS Statement 4258
VAR Statement 4258
Details: NESTED Procedure **4259**
Missing Values 4259
Unbalanced Data 4259
General Random-Effects Model 4259
Analysis of Covariation 4259
Error Terms in *F* Tests 4260
Computational Method 4260
Displayed Output 4261
ODS Table Names 4263
Example: NESTED Procedure **4263**
Example 59.1: Variability of Calcium Concentration in Turnip Greens . . . 4263
References **4265**

Overview: NESTED Procedure

The NESTED procedure performs random-effects analysis of variance for data from an experiment with a nested (hierarchical) structure.[1] A random-effects model for data from a completely nested design with two factors has the form

$$y_{ijr} = \mu + \alpha_i + \beta_{ij} + \epsilon_{ijr}$$

where

[1] PROC NESTED is modeled after the General Purpose Nested Analysis of Variance program of the Dairy Cattle Research Branch of the United States Department of Agriculture. That program was originally written by M. R. Swanson, Statistical Reporting Service, United States Department of Agriculture.

y_{ijr} is the value of the dependent variable observed at the rth replication with the first factor at its ith level and the second factor at its jth level.

μ is the overall (fixed) mean of the sampling population.

$\alpha_i, \beta_{ij}, \epsilon_{ijr}$ are mutually uncorrelated random effects with zero means and respective variances σ_α^2, σ_β^2, and σ_ϵ^2 (the variance components).

This model is appropriate for an experiment with a multistage nested sampling design. An example of this is given in Example 59.1, where four turnip plants are randomly chosen (the first factor), then three leaves are randomly chosen from each plant (the second factor nested within the first), and then two samples are taken from each leaf (the different replications at fixed levels of the two factors).

Note that PROC NESTED is appropriate for models with only classification effects; it does not handle models that contain continuous covariates. For random effects models with covariates, use either the GLM or MIXED procedure.

Contrasted with Other SAS Procedures

The NESTED procedure performs a computationally efficient analysis of variance for data with a nested design, estimating the different components of variance and also testing for their significance if the design is balanced (see the section "Unbalanced Data" on page 4259). Although other procedures (such as GLM and MIXED) provide similar analyzes, PROC NESTED is both easier to use and computationally more efficient for this special type of design. This is especially true when the design involves a large number of factors, levels, or observations.

For example, to specify a four-factor completely nested design in the GLM procedure, you use the following form:

```
class a b c d;
model y=a b(a) c(a b) d(a b c);
```

However, to specify the same design in PROC NESTED, you simply use the following form:

```
class a b c d;
var y;
```

In addition, other procedures require TEST statements to perform appropriate tests, whereas the NESTED procedure produces the appropriate tests automatically. However, PROC NESTED makes one assumption about the input data that the other procedures do not: **PROC NESTED assumes that the input data set is sorted by the classification (CLASS) variables defining the effects.** If you use PROC NESTED on data that are not sorted by the CLASS variables, then the results might not be valid.

Getting Started: NESTED Procedure

Reliability of Automobile Models

A study is performed to compare the reliability of several models of automobiles. Three different automobile models (Model) from each of four U.S. automobile manufacturers (Make) are tested. Three different cars of each make and model are subjected to a reliability test and given a score between 1 and 100 (Score), where higher scores indicate greater reliability.

The following statements create the SAS data set auto.

```
title1 'Reliability of Automobile Models';
data auto;
   input Make $ Model Score @@;
   datalines;
a 1 62  a 2 77  a 3 59
a 1 67  a 2 73  a 3 64
a 1 60  a 2 79  a 3 60
b 1 72  b 2 58  b 3 80
b 1 75  b 2 63  b 3 84
b 1 69  b 2 57  b 3 89
c 1 94  c 2 76  c 3 81
c 1 90  c 2 75  c 3 85
c 1 88  c 2 78  c 3 85
d 1 69  d 2 73  d 3 90
d 1 72  d 2 88  d 3 87
d 1 76  d 2 87  d 3 92
;
```

The Make variable contains the make of the automobile, represented here by 'a', 'b', 'c', or 'd', while the Model variable represents the automobile model with a '1', '2', or '3'. The Score variable contains the reliability scores given to the three sampled cars from each Make-Model group. Since the automobile models are nested within their makes, the NESTED procedure is used to analyze these data. The NESTED procedure requires the data to be sorted by Make and, within Make, by Model, so the following statements execute a PROC SORT before completing the analysis.

```
proc sort data=auto;
   by Make Model;

title1 'Reliability of Automobile Models';
proc nested data=auto;
   classes Make Model;
   var Score;
run;
```

The Model variable appears after the Make variable in the CLASS statement because it is nested within Make. The VAR statement specifies the response variable. The output is displayed in Figure 59.1.

Figure 59.1 Output from PROC NESTED

Reliability of Automobile Models

The NESTED Procedure

Coefficients of Expected Mean Squares

Source	Make	Model	Error
Make	9	3	1
Model	0	3	1
Error	0	0	1

Nested Random Effects Analysis of Variance for Variable Score

Variance Source	DF	Sum of Squares	F Value	Pr > F	Error Term	Mean Square	Variance Component	Percent of Total
Total	35	4177.888889				119.368254	131.876543	100.0000
Make	3	1709.000000	2.15	0.1719	Model	569.666667	33.867284	25.6811
Model	8	2118.888889	18.16	<.0001	Error	264.861111	83.425926	63.2606
Error	24	350.000000				14.583333	14.583333	11.0583

Score Mean	75.94444444
Standard Error of Score Mean	3.97794848

Figure 59.1 first displays the coefficients of the variance components that make up each of the expected mean squares, and then it displays the ANOVA table. The results do not indicate significant variation between the different automobile makes ($F = 2.15, p = 0.1719$). However, they do suggest that there is significant variation between the different models within the makes ($F = 18.16, p < 0.0001$). This is evident in the fact that the make of car accounts for only 25.7% of the total variation in the data, while the car model accounts for 63.3% (as shown in the Percent of Total column). The estimated variance components are shown in the Variance Component column.

Syntax: NESTED Procedure

The following statements are available in PROC NESTED:

PROC NESTED < *options* > ;
  CLASS *variables* < / *option* > ;
  VAR *variables* ;
  BY *variables* ;

The PROC NESTED and CLASS statements are required. The BY, CLASS, and VAR statements are described after the PROC NESTED statement.

PROC NESTED Statement

PROC NESTED < *options* > **;**

The PROC NESTED statement has the following options:

AOV
displays only the analysis of variance statistics when there is more than one dependent variable. The "analysis of covariation" statistics are suppressed (see the section "Analysis of Covariation" on page 4259).

DATA=*SAS-data-set*
names the SAS data set to be used by PROC NESTED. By default, the procedure uses the most recently created SAS data set.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC NESTED to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in the order of the BY variables.

NOTE: When you use the NESTED procedure, your data must be sorted first by the BY variables and, within the BY variables, by the CLASS variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the NESTED procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* < / *option* > **;**

You must include a CLASS statement with PROC NESTED specifying the classification variables for the analysis.

Values of a variable in the CLASS statement denote the levels of an effect. The name of that variable is also the name of the corresponding effect. The second effect is assumed to be nested within the first effect, the third effect is assumed to be nested within the second effect, and so on.

By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. Prior to SAS 9, class levels were determined using no more than the first eight characters of the formatted values, except for numeric variables with no explicit format, for which class levels were determined from the raw numeric values. If you want to revert to this previous behavior, you can use the TRUNCATE option in the CLASS statement. In any case, you can use formats to group values into levels. Refer to the discussion of the FORMAT procedure in the *Base SAS Procedures Guide*, and the discussions for the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

NOTE: The data set must be sorted by the classification variables in the order in which they are given in the CLASS statement. Use PROC SORT to sort the data if they are not already sorted.

You can specify the following option in the CLASS statement after a slash(/):

TRUNCATE
specifies that class levels should be determined using only up to the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option to revert to the levels as determined in releases prior to SAS 9.

VAR Statement

VAR *variables* **;**

The VAR statement lists the dependent variables for the analysis. The dependent variables must be numeric variables. If you do not specify a VAR statement, PROC NESTED performs an analysis of variance for all numeric variables in the data set, except those already specified in the CLASS statement.

Details: NESTED Procedure

Missing Values

An observation with missing values for any of the variables used by PROC NESTED is omitted from the analysis. Blank values of CLASS character variables are treated as missing values.

Unbalanced Data

A completely nested design is defined to be unbalanced if the groups corresponding to the levels of some classification variable are not all of the same size. The NESTED procedure can compute unbiased estimates for the variance components in an unbalanced design, but because the sums of squares on which these estimates are based no longer have χ^2 distributions under a Gaussian model for the data, F tests for the significance of the variance components cannot be computed. PROC NESTED checks to see that the design is balanced. If it is not, a warning to that effect appears in the log, and the columns corresponding to the F tests in the analysis of variance are left blank.

General Random-Effects Model

A random-effects model for data from a completely nested design with n factors has the general form

$$y_{i_1 i_2 \cdots i_n r} = \mu + \alpha_{i_1} + \beta_{i_1 i_2} + \cdots + \epsilon_{i_1 i_2 \cdots i_n r}$$

where

$y_{i_1 i_2 \cdots i_n r}$	is the value of the dependent variable observed at the rth replication with factor j at level i_j, for $j = 1, \ldots, n$.
μ	is the overall (fixed) mean of the sampled population.
$\alpha_{i_1}, \beta_{i_1 i_2}, \ldots, \epsilon_{i_1 i_2 \cdots i_n r}$	are mutually uncorrelated random effects with zero means and respective variances $\sigma^2_\alpha, \sigma^2_\beta, \ldots, \sigma^2_\epsilon$.

Analysis of Covariation

When you specify more than one dependent variable, the NESTED procedure produces a descriptive analysis of the covariance between each pair of dependent variables in addition to a separate analysis

of variance for each variable. The analysis of covariation is computed under the basic random-effects model for each pair of dependent variables:

$$\begin{aligned} y_{i_1 i_2 \cdots i_n r} &= \mu + \alpha_{i_1} + \beta_{i_1 i_2} + \cdots + \epsilon_{i_1 i_2 \cdots i_n r} \\ y'_{i_1 i_2 \cdots i_n r} &= \mu' + \alpha'_{i_1} + \beta'_{i_1 i_2} + \cdots + \epsilon'_{i_1 i_2 \cdots i_n r} \end{aligned}$$

where the notation is the same as that used in the preceding general random-effects model.

There is an additional assumption that all the random effects in the two models are mutually uncorrelated except for corresponding effects, for which

$$\begin{aligned} \mathrm{Corr}(\alpha_{i_1}, \alpha'_{i_1}) &= \rho_\alpha \\ \mathrm{Corr}(\beta_{i_1 i_2}, \beta'_{i_1 i_2}) &= \rho_\beta \\ &\vdots \\ \mathrm{Corr}(\epsilon_{i_1 i_2 \cdots i_n r}, \epsilon'_{i_1 i_2 \cdots i_n r}) &= \rho_\epsilon \end{aligned}$$

Error Terms in *F* Tests

Random-effects ANOVAs are distinguished from fixed-effects ANOVAs by which error mean squares are used as the denominator for F tests. Under a fixed-effects model, there is only one true error term in the model, and the corresponding mean square is used as the denominator for all tests. This is how the usual analysis is computed in PROC ANOVA, for example. However, in a random-effects model for a nested experiment, mean squares are compared sequentially. The correct denominator in the test for the first factor is the mean square due to the second factor; the correct denominator in the test for the second factor is the mean square due to the third factor; and so on. Only the mean square due to the last factor, the one at the bottom of the nesting order, should be compared to the error mean square.

Computational Method

The building blocks of the analysis are the sums of squares for the dependent variables for each classification variable within the factors that precede it in the model, corrected for the factors that

follow it. For example, for a two-factor nested design, PROC NESTED computes the following sums of squares:

$$\text{Total SS} \qquad \sum_{ijr}(y_{ijr} - y_{...})^2$$

$$\text{SS for Factor 1} \qquad \sum_{i} n_{i\cdot}\left(\frac{y_{i\cdot\cdot}}{n_{i\cdot}} - \frac{y_{...}}{n_{\cdot\cdot}}\right)^2$$

$$\text{SS for Factor 2 within Factor 1} \qquad \sum_{ij} n_{ij}\left(\frac{y_{ij\cdot}}{n_{ij}} - \frac{y_{i\cdot\cdot}}{n_{i\cdot}}\right)^2$$

$$\text{Error SS} \qquad \sum_{ijr}\left(y_{ijr} - \frac{y_{ij\cdot}}{n_{ij}}\right)^2$$

where y_{ijr} is the rth replication, n_{ij} is the number of replications at level i of the first factor and level j of the second, and a dot as a subscript indicates summation over the corresponding index. If there is more than one dependent variable, PROC NESTED also computes the corresponding sums of crossproducts for each pair. The expected value of the sum of squares for a given classification factor is a linear combination of the variance components corresponding to this factor and to the factors that are nested within it. For each factor, the coefficients of this linear combination are computed. (The efficiency of PROC NESTED is partly due to the fact that these various sums can be accumulated with just one pass through the data, assuming that the data have been sorted by the classification variables.) Finally, estimates of the variance components are derived as the solution to the set of linear equations that arise from equating the mean squares to their expected values.

Displayed Output

PROC NESTED displays the following items for each dependent variable:

- Coefficients of Expected Mean Squares, which are the coefficients of the $n + 1$ variance components making up the expected mean square. Denoting the element in the ith row and jth column of this matrix by C_{ij}, the expected value of the mean square due to the ith classification factor is

  $$C_{i1}\sigma_1^2 + \cdots + C_{in}\sigma_n^2 + C_{i,n+1}\sigma_\epsilon^2$$

  C_{ij} is always zero for $i > j$, and if the design is balanced, C_{ij} is equal to the common size of all classification groups of the jth factor for $i \leq j$. Finally, the mean square for error is always an unbiased estimate of σ_ϵ^2. In other words, $C_{n+1,n+1} = 1$.

For every dependent variable, PROC NESTED displays an analysis of variance table. Each table contains the following:

- each Variance Source in the model (the different components of variance) and the total variance
- degrees of freedom (DF) for the corresponding sum of squares
- Sum of Squares for each classification factor. The sum of squares for a given classification factor is the sum of squares in the dependent variable within the factors that precede it in the model, corrected for the factors that follow it. (See the section "Computational Method" on page 4260.)
- F Value for a factor, which is the ratio of its mean square to the appropriate error mean square. The next column, labeled PR > F, gives the significance levels that result from testing the hypothesis that each variance component equals zero.
- the appropriate Error Term for an F test, which is the mean square due to the next classification factor in the nesting order. (See the section "Error Terms in F Tests" on page 4260.)
- Mean Square due to a factor, which is the corresponding sum of squares divided by the degrees of freedom
- estimates of the Variance Components. These are computed by equating the mean squares to their expected values and solving for the variance terms. (See the section "Computational Method" on page 4260.)
- Percent of Total, the proportion of variance due to each source. For the ith factor, the value is

$$100 \times \frac{\text{source variance component}}{\text{total variance component}}$$

- Mean, the overall average of the dependent variable. This gives an unbiased estimate of the mean of the population. Its variance is estimated by a certain linear combination of the estimated variance components, which is identical to the mean square due to the first factor in the model divided by the total number of observations when the design is balanced.

If there is more than one dependent variable, then the NESTED procedure displays an "analysis of covariation" table for each pair of dependent variables (unless the AOV option is specified in the PROC NESTED statement). See the section "Analysis of Covariation" on page 4259 for details. For each source of variation, this table includes the following:

- Degrees of Freedom
- Sum of Products
- Mean Products
- Covariance Component, the estimate of the covariance component

Items in the analysis of covariation table are computed analogously to their counterparts in the analysis of variance table. The analysis of covariation table also includes the following:

- Variance Component Correlation for a given factor. This is an estimate of the correlation between corresponding effects due to this factor. This correlation is the ratio of the covariance component for this factor to the square root of the product of the variance components for the factor for the two different dependent variables. (See the section "Analysis of Covariation" on page 4259.)
- Mean Square Correlation for a given classification factor. This is the ratio of the Mean Products for this factor to the square root of the product of the Mean Squares for the factor for the two different dependent variables.

ODS Table Names

PROC NESTED assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 59.1 ODS Tables Produced by PROC NESTED

ODS Table Name	Description	Statement
ANCOVA	Analysis of covariance	default with more than one dependent variable
ANOVA	Analysis of variance	default
EMSCoef	Coefficients of expected mean squares	default
Statistics	Overall statistics for fit	default

Example: NESTED Procedure

Example 59.1: Variability of Calcium Concentration in Turnip Greens

In the following example from Snedecor and Cochran (1976), an experiment is conducted to study the variability of calcium concentration in turnip greens. Four plants are selected at random; then three leaves are randomly selected from each plant. Two 100-mg samples are taken from each leaf. The amount of calcium is determined by microchemical methods.

Because the data are read in sorted order, it is not necessary to use PROC SORT on the CLASS variables. Leaf is nested in Plant; Sample is nested in Leaf and is left for the residual term. All the effects are random effects. The following statements read the data and invoke PROC NESTED. These statements produce Output 59.1.1.

```
title 'Calcium Concentration in Turnip Leaves'
      '--Nested Random Model';
title2 'Snedecor and Cochran, ''Statistical Methods'''
       ', 1976, p. 286';
data Turnip;
   do Plant=1 to 4;
      do Leaf=1 to 3;
         do Sample=1 to 2;
            input Calcium @@;
            output;
         end;
      end;
   end;
   datalines;
3.28 3.09 3.52 3.48 2.88 2.80 2.46 2.44
1.87 1.92 2.19 2.19 2.77 2.66 3.74 3.44
2.55 2.55 3.78 3.87 4.07 4.12 3.31 3.31
;

proc nested data=Turnip;
   classes plant leaf;
   var calcium;
run;
```

Output 59.1.1 Analysis of Calcium Concentration in Turnip Greens Using PROC NESTED

Calcium Concentration in Turnip Leaves--Nested Random Model
Snedecor and Cochran, 'Statistical Methods', 1976, p. 286

The NESTED Procedure

Coefficients of Expected Mean Squares

Source	Plant	Leaf	Error
Plant	6	2	1
Leaf	0	2	1
Error	0	0	1

Nested Random Effects Analysis of Variance for Variable Calcium

Variance Source	DF	Sum of Squares	F Value	Pr > F	Error Term	Mean Square	Variance Component	Percent of Total
Total	23	10.270396				0.446539	0.532938	100.0000
Plant	3	7.560346	7.67	0.0097	Leaf	2.520115	0.365223	68.5302
Leaf	8	2.630200	49.41	<.0001	Error	0.328775	0.161060	30.2212
Error	12	0.079850				0.006654	0.006654	1.2486

Calcium Mean	3.01208333
Standard Error of Calcium Mean	0.32404445

The results indicate that there is significant (nonzero) variation from plant to plant (Pr > F is 0.0097) and from leaf to leaf within a plant (Pr > F is less than 0.0001). Notice that the variance component for Plant uses the Leaf mean square as an error term in the model rather than the error mean square.

References

Snedecor, G. W. and Cochran, W. G. (1976), *Statistical Methods*, Sixth Edition, Ames: Iowa State University Press.

Steel, R. G. D. and Torrie, J. H. (1980), *Principles and Procedures of Statistics*, Second Edition, New York: McGraw-Hill.

Chapter 60

The NLIN Procedure

Contents

Overview: NLIN Procedure **4268**
Getting Started: NLIN Procedure **4270**
Nonlinear or Linear Model 4270
Notation for Nonlinear Regression Models 4271
Estimating the Parameters in the Nonlinear Model 4272
Syntax: NLIN Procedure **4277**
PROC NLIN Statement 4278
BOUNDS Statement 4283
BY Statement 4284
CONTROL Statement 4285
DER Statements 4285
ID Statement 4285
MODEL Statement 4286
OUTPUT Statement 4286
PARAMETERS Statement 4289
RETAIN Statement 4292
Other Programming Statements 4292
Details: NLIN Procedure **4293**
Automatic Derivatives 4293
Hougaard's Measure of Skewness 4295
Missing Values 4297
Special Variables 4297
Troubleshooting 4299
Computational Methods 4301
Output Data Sets 4306
Confidence Intervals 4306
Covariance Matrix of Parameter Estimates 4307
Convergence Measures 4308
Displayed Output 4309
Incompatibilities with SAS 6.11 and Earlier Versions of PROC NLIN 4310
ODS Table Names 4311
Examples: NLIN Procedure **4313**
Example 60.1: Segmented Model 4313
Example 60.2: Iteratively Reweighted Least Squares 4318

Example 60.3: Probit Model with Likelihood Function 4321
Example 60.4: Affecting Curvature through Parameterization 4324
Example 60.5: Comparing Nonlinear Trends among Groups 4331
References **4341**

Overview: NLIN Procedure

The NLIN procedure fits nonlinear regression models and estimates the parameters by nonlinear least squares or weighted nonlinear least squares. You specify the model with programming statements. This gives you great flexibility in modeling the relationship between the response variable and independent (regressor) variables. It does, however, require additional coding compared to model specifications in linear modeling procedures such as the REG, GLM, and MIXED procedures.

Estimating parameters in a nonlinear model is an iterative process that commences from starting values. You need to declare the parameters in your model and supply their initial values for the NLIN procedure. You do not need to specify derivatives of the model equation with respect to the parameters. Although facilities for specifying first and second derivatives exist in the NLIN procedure, it is not recommended that you specify derivatives this way. Obtaining derivatives from user-specified expressions predates the high-quality automatic differentiator that is now used by the NLIN procedure.

Nonlinear least-squares estimation involves finding those values in the parameter space that minimize the (weighted) residual sum of squares. In a sense, this is a "distribution-free" estimation criterion since the distribution of the data does not need to be fully specified. Instead, the assumption of homoscedastic and uncorrelated model errors with zero mean is sufficient. You can relax the homoscedasticity assumption by using a weighted residual sum of squares criterion. The assumption of uncorrelated errors (independent observations) cannot be relaxed in the NLIN procedure. In summary, the primary assumptions for analyses with the NLIN procedure are as follows:

- The structure in the response variable can be decomposed additively into a mean function and an error component.
- The model errors are uncorrelated and have zero mean. Unless a weighted analysis is performed, the errors are also assumed to be homoscedastic (have equal variance).
- The mean function consists of known regressor (independent) variables and unknown constants (the parameters).

Fitting nonlinear models can be a difficult undertaking. There is no closed-form solution for the parameter estimates, and the process is iterative. There can be one or more local minima in the residual sum of squares surface, and the process depends on the starting values supplied by the user. You can reduce the dependence on the starting values and reduce the chance to arrive at a local minimum by specifying a grid of starting values. The NLIN procedure then computes the residual

sum of squares at each point on the grid and starts the iterative process from the point that yields the lowest sum of squares. Even in this case, however, convergence does not guarantee that a global minimum has been found.

The numerical behavior of a model and a model–data combination can depend on the way in which you parameterize the model—for example, whether parameters are expressed on the logarithmic scale or not. Parameterization also has bearing on the interpretation of the estimated quantities and the statistical properties of the parameter estimators. Inferential procedures in nonlinear regression models are typically approximate in that they rely on the asymptotic properties of the parameter estimators that are obtained as the sample size grows without bound. Such asymptotic inference can be questionable in small samples, especially if the behavior of the parameter estimators is "far-from-linear." Reparameterization of the model can yield parameters whose behavior is akin to that of estimators in linear models. These parameters exhibit close-to-linear behavior.

The NLIN procedure solves the nonlinear least squares problem by one of the following four algorithms (methods):

- steepest-descent or gradient method
- Newton method
- modified Gauss-Newton method
- Marquardt method

These methods use derivatives or approximations to derivatives of the SSE with respect to the parameters to guide the search for the parameters producing the smallest SSE. Derivatives computed automatically by the NLIN procedure are analytic, unless the model contains functions for which an analytic derivative is not available.

Using PROC NLIN, you can also do the following:

- confine the estimation procedure to a certain range of values of the parameters by imposing bounds on the estimates
- produce new SAS data sets containing predicted values, parameter estimates, residuals and other model diagnostics, estimates at each iteration, and so forth.

You can use the NLIN procedure for segmented models (see Example 60.1) or robust regression (see Example 60.2). You can also use it to compute maximum-likelihood estimates for certain models (see Jennrich and Moore 1975; Charnes, Frome, and Yu 1976). For maximum likelihood estimation in a model with a linear predictor and binomial error distribution, see the LOGISTIC, PROBIT, GENMOD, GLIMMIX, and CATMOD procedures. For a linear model with a Poisson, gamma, or inverse gaussian error distribution, see the GENMOD and GLIMMIX procedures. For likelihood estimation in a linear model with a normal error distribution, see the MIXED, GENMOD, and GLIMMIX procedures. The PHREG and LIFEREG procedures fit survival models by maximum likelihood. For general maximum likelihood estimation, see the NLP procedure in the *SAS/OR User's Guide* and the NLMIXED procedure in the *SAS/STAT User's Guide*. These procedures are recommended over the NLIN procedure for solving maximum likelihood problems.

PROC NLIN uses the Output Delivery System (ODS), a SAS subsystem that provides capabilities for displaying and controlling the output from SAS procedures. ODS enables you to convert any of the output from PROC NLIN into a SAS data set. See the section "ODS Table Names" on page 4311 for a listing of the ODS tables that are produced by the NLIN procedure.

Getting Started: NLIN Procedure

Nonlinear or Linear Model

The NLIN procedure performs univariate nonlinear regression by using the least squares method. Nonlinear regression analysis is indicated when the functional relationship between the response variable and the predictor variables is nonlinear. Nonlinearity in this context refers to a nonlinear relationship in the *parameters*. Many linear regression models exhibit a relationship in the regressor (predictor) variables that is not simply a straight line. This does not make the models nonlinear. A model is nonlinear in the parameters if the derivative of the model with respect to a parameter depends on this or other parameters.

Consider, for example the models

$$\mathrm{E}[Y|x] = \beta_0 + \beta_1 x$$
$$\mathrm{E}[Y|x] = \beta_0 + \beta_1 x + \beta_2 x^2$$
$$\mathrm{E}[Y|x] = \beta + x/\alpha$$

In these expressions, $\mathrm{E}[Y|x]$ denotes the expected value of the response variable Y at the fixed value of x. (The conditioning on x simply indicates that the predictor variables are assumed to be non-random in models fit by the NLIN procedure. Conditioning is often omitted for brevity in this chapter.)

Only the third model is a nonlinear model. The first model is a simple linear regression. It is linear in the parameters β_0 and β_1 since the model derivatives do not depend on unknowns:

$$\frac{\partial}{\beta_0}(\beta_0 + \beta_1 x) = 1$$
$$\frac{\partial}{\beta_1}(\beta_0 + \beta_1 x) = x$$

The model is also linear in its relationship with x (a straight line). The second model is also linear

in the parameters, since

$$\frac{\partial}{\beta_0}\left(\beta_0 + \beta_1 x + \beta_2 x^2\right) = 1$$
$$\frac{\partial}{\beta_1}\left(\beta_0 + \beta_1 x + \beta_2 x^2\right) = x$$
$$\frac{\partial}{\beta_2}\left(\beta_0 + \beta_1 x + \beta_2 x^2\right) = x^2$$

It is a *curvilinear* model since it exhibits a curved relationship when plotted against x. The third model, finally, is a nonlinear model since

$$\frac{\partial}{\beta}\left(\beta + x/\alpha\right) = 1$$
$$\frac{\partial}{\alpha}\left(\beta + x/\alpha\right) = -\frac{x}{\alpha^2}$$

The second of these derivatives depends on a parameter α. A model is nonlinear if it is not linear in at least one parameter.

Notation for Nonlinear Regression Models

This section briefly introduces the basic notation for nonlinear regression models that applies in this chapter. Additional notation is introduced throughout as needed.

The $(n \times 1)$ vector of observed responses is denoted as $\mathbf{y}$. This vector is the realization of an $(n \times 1)$ random vector $\mathbf{Y}$. The NLIN procedure assumes that the variance matrix of this random vector is $\sigma^2\mathbf{I}$. In other words, the observations have equal variance (are homoscedastic) and are uncorrelated. By defining the special variable _WEIGHT_ in your NLIN programming statements, you can introduce heterogeneous variances. If a _WEIGHT_ variable is present, then $\mathrm{Var}[\mathbf{Y}] = \sigma^2\mathbf{W}^{-1}$, where $\mathbf{W}$ is a diagonal matrix containing the values of the _WEIGHT_ variable.

The mean of the random vector is represented by a nonlinear model that depends on parameters $\beta_1, \cdots, \beta_p$ and regressor (independent) variables $z_1, \cdots, z_k$:

$$\mathrm{E}[Y_i] = f\left(\beta_1, \beta_2, \cdots, \beta_p; z_{i1}, \cdots, z_{ik}\right)$$

In contrast to linear models, the number of regressor variables (k) does not necessarily equal the number of parameters (p) in the mean function $f(\,)$. For example, the model fitted in the next subsection contains a single regressor and two parameters.

To represent the mean of the vector of observations, boldface notation is used in an obvious extension of the previous equation:

$$\mathrm{E}[\mathbf{Y}] = \mathbf{f}(\boldsymbol{\beta}; \mathbf{z}_1, \cdots, \mathbf{z}_k)$$

The vector $\mathbf{z}_1$, for example, is an $(n \times 1)$ vector of the values for the first regressor variables. The explicit dependence of the mean function on $\boldsymbol{\beta}$ and/or the $\mathbf{z}$ vectors is often omitted for brevity.

In summary, the stochastic structure of models fit with the NLIN procedure is mathematically captured by

$$\begin{aligned} \mathbf{Y} &= \mathbf{f}(\boldsymbol{\beta}; \mathbf{z}_1, \cdots, \mathbf{z}_k) + \boldsymbol{\epsilon} \\ \mathrm{E}[\boldsymbol{\epsilon}] &= \mathbf{0} \\ \mathrm{Var}[\boldsymbol{\epsilon}] &= \sigma^2 \mathbf{I} \end{aligned}$$

Note that the residual variance σ^2 is typically also unknown. Since it is not estimated in the same fashion as the other p parameters, it is often not counted in the number of parameters of the nonlinear regression. An estimate of σ^2 is obtained after the model fit by the method of moments based on the residual sum of squares.

A matrix that plays an important role in fitting nonlinear regression models is the $(n \times p)$ matrix of the first partial derivatives of the mean function $\mathbf{f}$ with respect to the p model parameters. It is frequently denoted as

$$\mathbf{X} = \frac{\partial \mathbf{f}\,(\boldsymbol{\beta}; \mathbf{z}_1, \cdots, \mathbf{z}_k)}{\partial \boldsymbol{\beta}}$$

The use of the symbol $\mathbf{X}$—common in linear statistical modeling—is no accident here. The first derivative matrix plays a similar role in nonlinear regression to that of the $\mathbf{X}$ matrix in a linear model. For example, the asymptotic variance of the nonlinear least-squares estimators is proportional to $(\mathbf{X}'\mathbf{X})^{-1}$, and projection-type matrices in nonlinear regressions are based on $\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'$. Also, fitting a nonlinear regression model can be cast as an iterative process where a nonlinear model is approximated by a series of linear models in which the derivative matrix is the regressor matrix. An important difference between linear and nonlinear models is that the derivatives in a linear model do not depend on any parameters (see previous subsection). In contrast, the derivative matrix $\partial \mathbf{f}(\boldsymbol{\beta})/\partial \boldsymbol{\beta}$ is a function of at least one element of $\boldsymbol{\beta}$. It is this dependence that lies at the core of the fact that estimating the parameters in a nonlinear model cannot be accomplished in closed form, but it is an iterative process that commences with user-supplied starting values and attempts to continually improve on the parameter estimates.

Estimating the Parameters in the Nonlinear Model

As an example of a nonlinear regression analysis, consider the following theoretical model of enzyme kinetics. The model relates the initial velocity of an enzymatic reaction to the substrate concentration.

$$f(\mathbf{x}, \boldsymbol{\theta}) = \frac{\theta_1 x_i}{\theta_2 + x_i}, \quad \text{for } i = 1, 2, \ldots, n$$

where x_i represents the amount of substrate for n trials and $f(\mathbf{x}, \boldsymbol{\theta})$ is the velocity of the reaction. The vector $\boldsymbol{\theta}$ contains the rate parameters. This model is known as the Michaelis-Menten model in biochemistry (Ratkowsky, 1990, p. 59). The model exists in many parameterizations. In the

form shown here, θ_1 is the maximum velocity of the reaction that is theoretically attainable. The parameter θ_2 is the substrate concentration at which the velocity is 50% of the maximum.

Suppose that you want to study the relationship between concentration and velocity for a particular enzyme/substrate pair. You record the reaction rate (velocity) observed at different substrate concentrations. A SAS data set is created for this experiment in the following DATA step:

```
data Enzyme;
   input Concentration Velocity @@;
   datalines;
0.26 124.7   0.30 126.9
0.48 135.9   0.50 137.6
0.54 139.6   0.68 141.1
0.82 142.8   1.14 147.6
1.28 149.8   1.38 149.4
1.80 153.9   2.30 152.5
2.44 154.5   2.48 154.7
;
```

The SAS data set Enzyme contains the two variables Concentration (substrate concentration) and Velocity (reaction rate). The following statements fit the Michaelis-Menten model by nonlinear least squares:

```
proc nlin data=Enzyme method=marquardt hougaard;
   parms theta1=155
         theta2=0 to 0.07 by 0.01;
   model Velocity = theta1*Concentration / (theta2 + Concentration);
run;
```

The DATA= option specifies that the SAS data set Enzyme be used in the analysis. The METHOD= option directs PROC NLIN to use the MARQUARDT iterative method. The HOUGAARD option requests that a skewness measure be calculated for the parameters.

The PARMS statement declares the parameters and specifies their initial values. Suppose that V represents the velocity and C represents the substrate concentration. In this example, the initial estimates listed in the PARMS statement for θ_1 and θ_2 are obtained as follows:

θ_1: Because the model is a monotonic increasing function in C, and because

$$\lim_{C\to\infty}\left(\frac{\theta_1 C}{\theta_2 + C}\right) = \theta_1$$

you can take the largest observed value of the variable Velocity (154.7) as the initial value for the parameter Theta1. Thus, the PARMS statement specifies 155 as the initial value for Theta1, which is approximately equal to the maximum observed velocity.

θ_2: To obtain an initial value for the parameter θ_2, first rearrange the model equation to solve for θ_2:

$$\theta_2 = \frac{\theta_1 C}{V} - C$$

By substituting the initial value of Theta1 for θ_1 and taking each pair of observed values of Concentration and Velocity for C and V, respectively, you obtain a set of possible starting values for Theta2 ranging from about 0.01 to 0.07.

You can choose any value within this range as a starting value for Theta2, or you can direct PROC NLIN to perform a preliminary search for the best initial Theta2 value within that range of values. The PARMS statement specifies a range of values for Theta2, resulting in a search over the grid points from 0 to 0.07 in increments of 0.01.

The MODEL statement specifies the enzymatic reaction model

$$V = \frac{\theta_1 C}{\theta_2 + C}$$

in terms of the data set variables Velocity and Concentration and in terms of the parameters in the PARMS statement.

The results from this PROC NLIN invocation are displayed in the following figures.

PROC NLIN evaluates the model at each point on the specified grid for the Theta2 parameter. Figure 60.1 displays the calculations resulting from the grid search.

Figure 60.1 Nonlinear Least-Squares Grid Search

The NLIN Procedure
Dependent Variable Velocity

Grid Search

theta1	theta2	Sum of Squares
155.0	0	3075.4
155.0	0.0100	2074.1
155.0	0.0200	1310.3
155.0	0.0300	752.0
155.0	0.0400	371.9
155.0	0.0500	147.2
155.0	0.0600	58.1130
155.0	0.0700	87.9662

The parameter Theta1 is held constant at its specified initial value of 155, the grid is traversed, and the residual sum of squares is computed at each point. The "best" starting value is the point that corresponds to the smallest value of the residual sum of squares. The best set of starting values is obtained for $\theta_1 = 155, \theta_2 = 0.06$ (Figure 60.1). PROC NLIN uses this point from which to start the following, iterative phase of nonlinear least-squares estimation.

Figure 60.2 displays the iteration history. Note that the first entry in the "Iterative Phase" table echoes the starting values and the residual sum of squares for the best value combination in Figure 60.1. The subsequent rows of the table show the updates of the parameter estimates and the improvement (decrease) in the residual sum of squares. For this data-and-model combination, the first iteration yielded a large improvement in the sum of squares (from 58.113 to 19.7017). Further steps were necessary to improve the estimates in order to achieve the convergence criterion. The NLIN procedure by default determines convergence by using R, the relative offset measure of Bates and Watts (1981). Convergence is declared when this measure is less than 10^{-5}—in this example, after three iterations.

Figure 60.2 Iteration History and Convergence Status

The NLIN Procedure
Dependent Variable Velocity
Method: Marquardt

Iterative Phase

Iter	theta1	theta2	Sum of Squares
0	155.0	0.0600	58.1130
1	158.0	0.0736	19.7017
2	158.1	0.0741	19.6606
3	158.1	0.0741	19.6606

NOTE: Convergence criterion met.

Figure 60.3 Estimation Summary

Estimation Summary

Method	Marquardt
Iterations	3
R	5.861E-6
PPC(theta2)	8.569E-7
RPC(theta2)	0.000078
Object	2.902E-7
Objective	19.66059
Observations Read	14
Observations Used	14
Observations Missing	0

A summary of the estimation including several convergence measures (R, PPC, RPC, and Object) is displayed in Figure 60.3.

The "R" measure in Figure 60.3 is the relative offset convergence measure of Bates and Watts. A "PPC" value of $8.569\text{E}-7$ indicates that the parameter Theta2 (which has the largest PPC value of the parameters) would change by that relative amount, if PROC NLIN were to take an additional iteration step. The "RPC" value indicates that Theta2 changed by 0.000078, relative to its value in the last iteration. These changes are measured before step length adjustments are made. The "Object" measure indicates that the objective function changed by $2.902\text{E}-7$ in relative value from the last iteration.

Figure 60.4 displays the analysis of variance table for the model. The table displays the degrees of freedom, sums of squares, and mean squares along with the model F test.

Figure 60.4 Nonlinear Least-Squares Analysis of Variance

NOTE: An intercept was not specified for this model.

Source	DF	Sum of Squares	Mean Square	F Value	Approx Pr > F
Model	2	290116	145058	88537.2	<.0001
Error	12	19.6606	1.6384		
Uncorrected Total	14	290135			

Figure 60.5 Parameter Estimates and Approximate 95% Confidence Intervals

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits		Skewness
theta1	158.1	0.6737	156.6	159.6	0.0152
theta2	0.0741	0.00313	0.0673	0.0809	0.0362

Figure 60.5 displays the estimates for each parameter, the associated asymptotic standard error, and the upper and lower values for the asymptotic 95% confidence interval. PROC NLIN also displays the asymptotic correlations between the estimated parameters (not shown).

The skewness measures of 0.0152 and 0.0362 indicate that the parameter estimators exhibit close-to-linear behavior and that their standard errors and confidence intervals can be safely used for inferences.

Thus, the estimated nonlinear model relating reaction velocity and substrate concentration can be written as

$$\widehat{V} = \frac{158.1C}{0.0741 + C}$$

where $\widehat{V}$ represents the predicted velocity or rate of the reaction, and C represents the substrate concentration.

Syntax: NLIN Procedure

PROC NLIN < *options* > ;
BOUNDS *inequality* < , . . . , *inequality* > ;
BY *variables* ;
CONTROL *variable* < =*values* > < . . . *variable* < =*values* > > ;
DER. *parameter=expression* ;
DER. *parameter.parameter=expression* ;
ID *variables* ;
MODEL *dependent=expression* ;
OUTPUT OUT=*SAS-data-set keyword=names* < . . . *keyword=names* > ;
PARAMETERS < *parameter-specification* > < ,. . . , *parameter-specification* >
< / **PDATA=***SAS-data-set* > ;
RETAIN *variable* < =*values* > < . . . *variable* < =*values* > > ;
Programming Statements ;

The statements in the NLIN procedure, in addition to the PROC NLIN statement, are as follows:

BOUNDS	constrains the parameter estimates within specified bounds
BY	specifies variables to define subgroups for the analysis
DER	specifies the first or second partial derivatives
ID	specifies additional variables to add to the output data set
MODEL	defines the relationship between the dependent and independent variables (the mean function)
OUTPUT	creates an output data set containing observation-wise statistics
PARAMETERS	identifies parameters to be estimated and their starting values
Programming Statements	includes, for example, assignment statements, ARRAY statements, DO loops, and other program control statements. These are valid SAS expressions that can appear in the DATA step. PROC NLIN enables you to create new variables within the procedure and use them in the nonlinear analysis. These programming statements can appear anywhere in the PROC NLIN code, but new variables must be created before they appear in other statements. The NLIN procedure automatically creates several variables that are also available for use in the analysis. See the section "Special Variables" on page 4297 for more information.

The PROC NLIN, PARAMETERS, and MODEL statements are required.

PROC NLIN Statement

PROC NLIN < *options* > **;**

The PROC NLIN statement invokes the procedure.

The following table lists important options available in the PROC NLIN statement. All options are subsequently discussed in alphabetical order.

Table 60.1 Summary of Important Options in PROC NLIN Statement

Option	Description
Options Related to Data Sets	
DATA=	specifies input data set
OUTEST=	specifies output data set for parameter estimates, covariance matrix, etc.
SAVE	requests that final estimates be added to the OUTEST= data set
Optimization Options	
BEST=	limits display of grid search
METHOD=	chooses the optimization method
MAXITER=	specifies maximum number of iterations
MAXSUBIT=	specifies maximum number of step halvings
NOHALVE	allows objective function to increase between iterations
RHO=	controls the step-size search
SMETHOD=	specifies the step-size search method
TAU=	controls the step-size search
G4	uses Moore-Penrose inverse
UNCORRECTEDDF	does not expense degrees of freedom when bounds are active
SIGSQ=	specifies fixed value for residual variance
Singularity and Convergence Criteria	
CONVERGE=	tunes the convergence criterion
CONVERGEOBJ=	uses the change in loss function as the convergence criterion and tunes its value
CONVERGEPARM=	uses the maximum change in parameter estimates as the convergence criterion and tunes its value
SINGULAR=	tunes the singularity criterion used in matrix inversions
Displayed Output	
HOUGAARD	adds Hougaard's skewness measure to the "Parameter Estimates" table
NOITPRINT	suppresses the "Iteration History" table
NOPRINT	suppresses displayed output
LIST	displays model program and variable list
LISTALL	selects the LIST, LISTDEP, LISTDER, and LISTCODE options
LISTCODE	displays the model program code

Table 60.1 *continued*

Option	Description
LISTDEP	displays dependencies of model variables
LISTDER	displays the derivative table
TOTALSS	adds uncorrected or corrected total sum of squares to the analysis of variance table
XREF	displays cross-reference of variables
Trace Model Execution	
FLOW	displays execution messages for program statements
PRINT	displays results of statements in model program
TRACE	displays results of operations in model program

ALPHA=*α*

specifies the level of significance α used in the construction of $100(1-\alpha)\%$ confidence intervals. The value must be strictly between 0 and 1; the default value of $\alpha = 0.05$ results in 95% intervals. This value is used as the default confidence level for limits computed in the "Parameter Estimates" table and with the LCLM, LCL, UCLM, and UCL options in the OUTPUT statement.

BEST=*n*

requests that PROC NLIN display the residual sums of squares only for the best n combinations of possible starting values from the grid. If you do not specify the BEST= option, PROC NLIN displays the residual sum of squares for every combination of possible parameter starting values.

CONVERGE=*c*

specifies the convergence criterion for PROC NLIN. For all iterative methods the relative offset convergence measure of Bates and Watts is used by default to determine convergence. This measure is labeled "R" in the "Estimation Summary" table. The iterations are said to have converged for CONVERGE=c if

$$\sqrt{\frac{\mathbf{r}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{r}}{\mathrm{LOSS}^{(i)}}} < c$$

where $\mathbf{r}$ is the residual vector and $\mathbf{X}$ is the $(n \times p)$ matrix of first derivatives with respect to the parameters. The default LOSS function is the sum of squared errors (SSE), and $\mathrm{LOSS}^{(i)}$ denotes the value of the loss function at the ith iteration. By default, CONVERGE=10^{-5}. The R convergence measure cannot be computed accurately in the special case of a perfect fit (residuals close to zero). When the SSE is less than the value of the SINGULAR= criterion, convergence is assumed.

CONVERGEOBJ=*c*

uses the change in the LOSS function as the convergence criterion and tunes the criterion. The iterations are said to have converged for CONVERGEOBJ=c if

$$\frac{|\mathrm{LOSS}^{(i-1)} - \mathrm{LOSS}^{(i)}|}{|\mathrm{LOSS}^{(i-1)} + 10^{-6}|} < c$$

where $\text{LOSS}^{(i)}$ is the LOSS for the ith iteration. The default LOSS function is the sum of squared errors (SSE), the residual sum of squares. The constant c should be a small positive number. For more details about the LOSS function, see the section "Special Variable Used to Determine Convergence Criteria" on page 4298. For more details about the computational methods in the NIN procedure, see the section "Computational Methods" on page 4301.

Note that in SAS 6 the CONVERGE= and CONVERGEOBJ= options both requested that convergence be tracked by the relative change in the loss function. If you specify the CONVERGEOBJ= option in newer releases, the CONVERGE= option is disabled. This enables you to track convergence as in SAS 6.

CONVERGEPARM=*c*

uses the maximum change among parameter estimates as the convergence criterion and tunes the criterion. The iterations are said to have converged for CONVERGEPARM=c if

$$\max_j \left(\frac{|\beta_j^{(i-1)} - \beta_j^{(i)}|}{|\beta_j^{(i-1)}|} \right) < c$$

where $\beta_j^{(i)}$ is the value of the jth parameter at the ith iteration.

The default convergence criterion is CONVERGE. If you specify CONVERGEPARM=c, the maximum change in parameters is used as the convergence criterion. If you specify both the CONVERGEOBJ= and CONVERGEPARM= options, PROC NLIN continues to iterate until the decrease in LOSS is sufficiently small (as determined by the CONVERGEOBJ= option) and the maximum change among the parameters is sufficiently small (as determined by the CONVERGEPARM= option).

DATA=*SAS-data-set*

specifies the input SAS data set to be analyzed by PROC NLIN. If you omit the DATA= option, the most recently created SAS data set is used.

FLOW

displays a message for each statement in the model program as it is executed. This debugging option is rarely needed, and it produces large amounts of output.

G4

uses a Moore-Penrose inverse (g_4-inverse) in parameter estimation. See Kennedy and Gentle (1980) for details.

HOUGAARD

adds Hougaard's measure of skewness to the "Parameter Estimates" table (Hougaard 1982, 1985). The skewness measure is one method of assessing a parameter estimator's close-to-linear behavior in the sense of Ratkowsky (1983, 1990). The behavior of estimators that are close to linear approaches that of least-squares estimators in linear models, which are unbiased and have minimum variance. When you specify the HOUGAARD option, the standardized skewness measure of Hougaard (1985) is added for each parameter to the "Parameter Estimates" table. Because of the linkage between nonlinear behavior of a parameter estimator in nonlinear regression and the nonnormality of the estimator's sampling distribution, Ratkowsky (1990, p. 28) provides the following rules to interpret the (standardized) Hougaard skewness measure:

- Values less than 0.1 in absolute value indicate very close-to-linear behavior.
- Values between 0.1 and 0.25 in absolute value indicate reasonably close-to-linear behavior.
- The nonlinear behavior is apparent for absolute values above 0.25 and is considerable for absolute values above 1.

See the section "Hougaard's Measure of Skewness" on page 4295 for further details. Example 60.4 shows how to use this measure to evaluate changes in the parameterization of a nonlinear model. Computation of the Hougaard skewness measure requires first and second derivatives. If you do not provide derivatives with the DER statement—and it is recommended that you do not—the analytic derivatives are computed for you.

LIST

displays the model program and variable lists, including the statements added by macros. Note that the expressions displayed by the LIST option do not necessarily represent the way the expression is actually calculated—because intermediate results for common subexpressions can be reused—but are shown in expanded form. To see how the expression is actually evaluated, use the LISTCODE option.

LISTALL

selects the LIST, LISTDEP, LISTDER, and LISTCODE options.

LISTCODE

displays the derivative tables and the compiled model program code. The LISTCODE option is a debugging feature and is not normally needed.

LISTDEP

produces a report that lists, for each variable in the model program, the variables that depend on it and the variables on which it depends.

LISTDER

displays a table of derivatives. The derivatives table lists each nonzero derivative computed for the problem. The derivative listed can be a constant, a variable in the model program, or a special derivative variable created to hold the result of an expression.

MAXITER=*n*

specifies the maximum number n of iterations in the optimization process. The default is $n = 100$.

MAXSUBIT=*n*

places a limit on the number of step halvings. The value of MAXSUBIT must be a positive integer and the default value is $n = 30$.

METHOD=GAUSS

METHOD=MARQUARDT

METHOD=NEWTON

METHOD=GRADIENT

specifies the iterative method employed by the NLIN procedure in solving the nonlinear least-squares problem. The GAUSS, MARQUARDT, and NEWTON methods are more robust than

the GRADIENT method. If you omit the METHOD= option, METHOD=GAUSS is used. See the section "Computational Methods" on page 4301 for more information.

NOITPRINT
suppresses the display of the "Iteration History" table.

NOHALVE
removes the restriction that the objective value must decrease at every iteration. Step halving is still used to satisfy BOUNDS and to ensure that the number of observations that can be evaluated does not decrease. The NOHALVE option can be useful in weighted nonlinear least-squares problems where the weights depend on the parameters, such as in iteratively reweighted least-squares (IRLS) fitting. See Example 60.2 for an application of IRLS fitting.

NOPRINT
suppresses the display of the output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 20, "Using the Output Delivery System."

OUTEST=*SAS-data-set*
specifies an output data set that contains the parameter estimates produced at each iteration. See the section "Output Data Sets" for details. If you want to create a permanent SAS data set, you must specify a two-level name. See the chapter "SAS Files" in *SAS Language Reference: Concepts* for more information about permanent SAS data sets.

PRINT
displays the result of each statement in the program as it is executed. This option is a debugging feature that produces large amounts of output and is normally not needed.

RHO=*value*
specifies a value that controls the step-size search. By default RHO=0.1, except when METHOD=MARQUARDT. In that case, RHO=10. See the section "Step-Size Search" on page 4306 for more details.

SAVE
specifies that, when the iteration limit is exceeded, the parameter estimates from the final iteration be output to the OUTEST= data set. These parameter estimates are associated with the observation for which _TYPE_="FINAL". If you omit the SAVE option, the parameter estimates from the final iteration are not output to the data set unless convergence has been attained.

SIGSQ=*value*
specifies a value to use as the estimate of the residual variance in lieu of the estimated mean-squared error. This value is used in computing the standard errors of the estimates. Fixing the value of the residual variance can be useful, for example, in maximum likelihood estimation.

SINGULAR=*s*
specifies the singularity criterion, s, which is the absolute magnitude of the smallest pivot value allowed when inverting the Hessian or the approximation to the Hessian. The default value is 1E4 times the machine epsilon; this product is approximately $1E-12$ on most computers.

SMETHOD=HALVE

SMETHOD=GOLDEN

SMETHOD=CUBIC

specifies the step-size search method. The default is SMETHOD=HALVE. See the section "Step-Size Search" on page 4306 for details.

TAU=*value*

specifies a value that is used to control the step-size search. The default is TAU=1, except when METHOD=MARQUARDT. In that case the default is TAU=0.01. See the section "Step-Size Search" on page 4306 for details.

TOTALSS

adds to the analysis of variance table the uncorrected total sum of squares in models that have an (implied) intercept, and adds the corrected total sum of squares in models that do not have an (implied) intercept.

TRACE

displays the result of each operation in each statement in the model program as it is executed, in addition to the information displayed by the FLOW and PRINT options. This debugging option is needed very rarely, and it produces even more output than the FLOW and PRINT options.

XREF

displays a cross-reference of the variables in the model program showing where each variable is referenced or given a value. The XREF listing does not include derivative variables.

UNCORRECTEDDF

specifies that no degrees of freedom be lost when a bound is active. When the UNCORRECTEDDF option is not specified, an active bound is treated as if a restriction were applied to the set of parameters, so one parameter degree of freedom is deducted.

BOUNDS Statement

BOUNDS *inequality* < , . . . , *inequality* > **;**

The BOUNDS statement restricts the parameter estimates so that they lie within specified regions. In each BOUNDS statement, you can specify a series of boundary values separated by commas. The series of bounds is applied simultaneously. Each boundary specification consists of a list of parameters, an inequality comparison operator, and a value. In a single-bounded expression, these three elements follow one another in the order described. The following are examples of valid single-bounded expressions:

```
bounds a1-a10 <= 20;
bounds c > 30;
bounds a b c > 0;
```

Multiple-bounded expressions are also permitted. For example:

```
bounds 0 <= B<= 10;
bounds 15 < x1 <= 30;
bounds r <= s <= p < q;
```

If you need to restrict an expression involving several parameters (for example, $\alpha + \beta < 1$), you can reparameterize the model so that the expression becomes a parameter or so that the boundary constraint can be expressed as a simple relationship between two parameters. For example, the boundary constraint $\alpha + \beta < 1$ in the model

```
model y = alpha + beta*x;
```

can be achieved by parameterizing $\theta = 1 - \beta$ as follows:

```
bounds alpha < theta;
model y = alpha + (1-theta)*x;
```

Starting with SAS 7.01, Lagrange multipliers are reported for all bounds that are enforced (active) when the estimation terminates. In the "Parameter Estimates" table, the Lagrange multiplier estimates are identified with names *Bound1*, *Bound2*, and so forth. An active bound is treated as if a restriction were applied to the set of parameters so that one parameter degree of freedom is deducted. You can use the UNCORRECTEDDF option to prevent the loss of degrees of freedom when bounds are active.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC NLIN to obtain separate analyses for observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the NLIN procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CONTROL Statement

CONTROL *variable* <**=***values*> <*. . . variable* <**=***values*>> **;**

The CONTROL statement declares control variables and specifies their values. A control variable is like a retained variable (see the section "RETAIN Statement" on page 4292) except that it is retained *across* iterations, and the derivative of the model with respect to a control variable is always zero.

DER Statements

DER. *parameter***=***expression* **;**

DER. *parameter.parameter***=***expression* **;**

The DER statement specifies first or second partial derivatives. By default, analytical derivatives are automatically computed. However, you can specify the derivatives yourself by using the DER.parm syntax. Use the first form shown to specify first partial derivatives, and use the second form to specify second partial derivatives. Note that the DER.parm syntax is retained for backward compatibility. The automatic analytical derivatives are, in general, a better choice. For additional information about automatic analytical derivatives, see the section "Automatic Derivatives" on page 4293.

For most of the computational methods, you need only specify the first partial derivative with respect to each parameter to be estimated. For the NEWTON method, specify both the first and the second derivatives. If any needed derivatives are not specified, they are automatically computed.

The expression can be an algebraic representation of the partial derivative of the expression in the MODEL statement with respect to the parameter or parameters that appear on the left side of the DER statement. Numerical derivatives can also be used. The expression in the DER statement must conform to the rules for a valid SAS expression, and it can include any quantities that the MODEL statement expression contains.

ID Statement

ID *variables* **;**

The ID statement specifies additional variables to place in the output data set created by the OUTPUT statement. Any variable on the left side of any assignment statement is eligible. Also, the special variables created by the procedure can be specified. Variables in the input data set do not need to be specified in the ID statement since they are automatically included in the output data set.

MODEL Statement

MODEL *dependent*=*expression* **;**

The MODEL statement defines the prediction equation by declaring the dependent variable and defining an expression that evaluates predicted values. The expression can be any valid SAS expression that yields a numeric result. The expression can include parameter names, variables in the data set, and variables created by programming statements in the NLIN procedure. Any operators or functions that can be used in a DATA step can also be used in the MODEL statement.

A statement such as

```
model y=expression;
```

is translated into the form

```
model.y=expression;
```

using the compound variable name model.y to hold the predicted value. You can use this assignment directly as an alternative to the MODEL statement. Either a MODEL statement or an assignment to a compound variable such as model.y must appear.

OUTPUT Statement

OUTPUT OUT= *SAS-data-set keyword*=*names* < . . . *keyword*=*names* > < / *options* > **;**

The OUTPUT statement specifies an output data set to contain statistics calculated for each observation. For each statistic, specify the keyword, an equal sign, and a variable name for the statistic in the output data set. All of the names appearing in the OUTPUT statement must be valid SAS names, and none of the new variable names can match a variable already existing in the data set to which PROC NLIN is applied.

If an observation includes a missing value for one of the independent variables, both the predicted value and the residual value are missing for that observation. If the iterations fail to converge, all the values of all the variables named in the OUTPUT statement are missing values.

You can specify the following options in the OUTPUT statement. For a description of computational formulas, see Chapter 4, "Introduction to Regression Procedures."

OUT=*SAS-data-set*
specifies the SAS data set to be created by PROC NLIN when an OUTPUT statement is included. The new data set includes the variables in the input data set. Also included are any ID variables specified in the ID statement, plus new variables with names that are specified in the OUTPUT statement.

The following values can be calculated and output to the new data set.

H=*name*

specifies a variable that contains the leverage, $\mathbf{x}_i(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_i'$, where $\mathbf{X} = \partial\mathbf{f}/\partial\boldsymbol{\beta}$ and $\mathbf{x}_i$ is the ith row of $\mathbf{X}$. If you specify the special variable _WEIGHT_, the leverage is computed as $w_i\mathbf{x}_i(\mathbf{X}'\mathbf{W}\mathbf{X})^{-1}\mathbf{x}_i'$.

L95=*name*

specifies a variable that contains the lower bound of an approximate 95% confidence interval for an individual prediction. This includes the variance of the error as well as the variance of the parameter estimates. See also the description for the U95= option later in this section.

L95M=*name*

specifies a variable that contains the lower bound of an approximate 95% confidence interval for the expected value (mean). See also the description for the U95M= option later in this section.

LCL=*name*

specifies a variable that contains the lower bound of an approximate $100(1-\alpha)\%$ confidence interval for an individual prediction. The α level is equal to the value of the ALPHA= option in the OUTPUT statement or, if this option is not specified, to the value of the ALPHA= option in the PROC NLIN statement. If neither of these options is specified, then $\alpha = 0.05$ by default, resulting in a lower bound for an approximate 95% confidence interval. For the corresponding upper bound, see the UCL keyword.

LCLM=*name*

specifies a variable that contains the lower bound of an approximate $100(1-\alpha)\%$ confidence interval for the expected value (mean). The α level is equal to the value of the ALPHA= option in the OUTPUT statement or, if this option is not specified, to the value of the ALPHA= option in the PROC NLIN statement. If neither of these options is specified, then $\alpha = 0.05$ by default, resulting in a lower bound for an approximate 95% confidence interval. For the corresponding lower bound, see the UCLM keyword.

PARMS=*names*

specifies variables in the output data set that contains parameter estimates. These can be the same variable names that are listed in the PARAMETERS statement; however, you can choose new names for the parameters identified in the sequence from the parameter estimates table. A note in the log indicates which variable in the output data set is associated with which parameter name. Note that, for each of these new variables, the values are the same for every observation in the new data set.

PREDICTED=*name*

P=*name*

specifies a variable in the output data set that contains the predicted values of the dependent variable.

RESIDUAL=*name*

R=*name*

specifies a variable in the output data set that contains the residuals (observed minus predicted values).

SSE=*name*

ESS=*name*

specifies a variable in the output data set that contains the residual sum of squares finally determined by the procedure. The value of the variable is the same for every observation in the new data set.

STDI=*name*

specifies a variable that contains the standard error of the individual predicted value.

STDP=*name*

specifies a variable that contains the standard error of the mean predicted value.

STDR=*name*

specifies a variable that contains the standard error of the residual.

STUDENT=*name*

specifies a variable that contains the studentized residuals. These are residuals divided by their estimated standard deviation.

U95=*name*

specifies a variable that contains the upper bound of an approximate 95% confidence interval for an individual prediction. See also the description for the L95= option.

U95M=*name*

specifies a variable that contains the upper bound of an approximate 95% confidence interval for the expected value (mean). See also the description for the L95M= option.

UCL=*name*

specifies a variable that contains the upper bound of an approximate $100(1-\alpha)\%$ confidence interval an individual prediction. The α level is equal to the value of the ALPHA= option in the OUTPUT statement or, if this option is not specified, to the value of the ALPHA= option in the PROC NLIN statement. If neither of these options is specified, then $\alpha = 0.05$ by default, resulting in an upper bound for an approximate 95% confidence interval. For the corresponding lower bound, see the LCL keyword.

UCLM=*name*

specifies a variable that contains the upper bound of an approximate $100(1-\alpha)\%$ confidence interval for the expected value (mean). The α level is equal to the value of the ALPHA= option in the OUTPUT statement or, if this option is not specified, to the value of the ALPHA= option in the PROC NLIN statement. If neither of these options is specified, then $\alpha = 0.05$ by default, resulting in an upper bound for an approximate 95% confidence interval. For the corresponding lower bound, see the LCLM keyword.

WEIGHT=*name*

specifies a variable in the output data set that contains values of the special variable _WEIGHT_.

You can specify the following options in the OUTPUT statement after a slash (/) :

ALPHA=α

specifies the level of significance α for $100(1-\alpha)\%$ confidence intervals. By default, α is equal to the value of the ALPHA= option in the PROC NLIN statement or 0.05 if that option is not specified. You can supply values that are strictly between 0 and 1.

DER

saves the first derivatives of the model with respect to the parameters to the OUTPUT data set. The derivatives are named DER_parmname, where "parmname" is the name of the model parameter in your NLIN statements. You can use the DER option to extract the $\mathbf{X} = \partial\mathbf{f}/\partial\boldsymbol{\beta}$ matrix into a SAS data set. For example, the following statements create the data set nlinX, which contains the $\mathbf{X}$ matrix:

```
proc nlin;
   parms theta1=155 theta3=0.04;
   model V = theta1*c / (theta3 + c);
   output out=nlinout / der;
run;
data nlinX; set nlinout(keep=DER_:);
run;
```

The derivatives are evaluated at the final parameter estimates.

PARAMETERS Statement

PARAMETERS < *parameter-specification* > < ,. . . , *parameter-specification* >
< / **PDATA=***SAS-data-set* > **;**

PARMS < *parameter-specification* > < ,. . . , *parameter-specification* >
< / **PDATA=***SAS-data-set* > **;**

A PARAMETERS (or PARMS) statement is required. The purpose of this statement is to provide starting values for the NLIN procedure. You can provide values that define a point in the parameter space or a set of points.

All parameters must be assigned starting values through the PARAMETERS statement. The NLIN procedure does not assign default starting values to parameters in your model that do not appear in the PARAMETERS statement. However, it is not necessary to supply parameters and starting values explicitly through a *parameter-specification*. Starting values can also be provided through a data set. The names assigned to parameters must be valid SAS names and must not coincide with names of variables in the input data set (see the DATA= option in the PROC NLIN statement). Parameters that are assigned starting values through the PARAMETERS statement can be omitted from the estimation, for example, if the expression in the MODEL statement does not depend on them.

Assigning Starting Values with Parameter-Specification

A *parameter-specification* has the general form

name **=** *value-list*

where *name* identifies the parameter and *value-list* provides the set of starting values for the parameter.

Very often, the *value-list* contains only a single value, but more general and flexible list specifications are possible:

m	a single value
$m1, m2, \ldots, mn$	several values
m TO n	a sequence where m equals the starting value, n equals the ending value, and the increment equals 1
m TO n BY i	a sequence where m equals the starting value, n equals the ending value, and the increment is i
$m1, m2$ TO $m3$	mixed values and sequences

When you specify more than one value for a parameter, PROC NLIN sorts the values in ascending sequence and removes duplicate values from the parameter list before forming the grid for the parameter search. If you specify several values for each parameter, PROC NLIN evaluates the model at each point on the grid. The iterations then commence from the point on the grid that yields the smallest objective function value.

For example, the following PARMS statement specifies five parameters and sets their possible starting values as shown in the table:

```
parms  b0 = 0
       b1 = 4 to 8
       b2 = 0 to .6 by .2
       b3 = 1, 10, 100
       b4 = 0, .5, 1 to 4;
```

Possible starting values				
B0	B1	B2	B3	B4
0	4	0.0	1	0.0
	5	0.2	10	0.5
	6	0.4	100	1.0
	7	0.6		2.0
	8			3.0
				4.0

Residual sums of squares are calculated for each of the $1 \times 5 \times 4 \times 3 \times 6 = 360$ combinations of possible starting values. (Specifying a large grid of values can take considerable computing time.)

If you specify a starting value with a *parameter-specification*, any starting values provided for this parameter through the PDATA= data set are ignored. The *parameter-specification* overrides the information in the PDATA= data set. When you specify a BY statement, the same *parameter-specification* is applied in each BY group. To vary starting values with BY groups, use the PDATA= option in the PARAMETERS statement as described in the following paragraphs.

Assigning Starting Values from a SAS Data Set

The PDATA= option in the PARAMETERS statement enables you to assign starting values for parameters through a SAS data set. The data set must contain at least two variables that identify the parameter and contain starting values, respectively. The *character* variable identifying the parameters must be named Parameter or Parm. The *numeric* variable containing the starting value must be named Estimate or Est. This enables you, for example, to use the contents of the "ParameterEstimates" table from one PROC NLIN run to supply starting values for a subsequent run, as in the following example:

```
proc nlin data=FirstData;
   parameters alpha=100 beta=3 gamma=4;
   < other NLIN statements >
   model y = ... ;
   ods output ParameterEstimates=pest;
run;

proc nlin data=SecondData;
   parameters / pdata=pest;
   Switch = 1/(1+gamma*exp(beta*log(dose)));
   model y = alpha*Switch;
run;
```

You can specify multiple values for a parameter in the PDATA= data set, and the parameters can appear in any order. The starting values are collected by parameter and arranged in ascending order, and duplicate values are removed. The parameter names in the PDATA= data set are not case sensitive. For example, the following DATA step defines starting values for three parameters and a starting grid with $1 \times 3 \times 1 = 3$ points:

```
data test;
   input Parameter $ Estimate;
   datalines;
alpha  100
 BETA   4
 beta   4.1
beta    4.2
beta    4.1
 gamma 30
;
```

If starting values for a parameter are provided through the PDATA= data set and through an explicit *parameter-specification*, the latter takes precedence.

When you specify a BY statement, you can control whether the same starting values are applied to each BY group or whether the starting values are varied. If the BY variables are not present in the PDATA= data set, the entire contents of the PDATA= data set are applied in each BY group.

If the BY variables are present in the PDATA= data set, then BY-group-specific starting values are assigned.

RETAIN Statement

RETAIN *variable* <**=***values*> <*. . . variable* <**=***values*>> **;**

The RETAIN statement declares retained variables and specifies their values. A retained variable is like a control variable (see the section "CONTROL Statement" on page 4285) except that it is retained only *within* iterations. An iteration involves a single pass through the data set.

Other Programming Statements

PROC NLIN supports many statements that are similar to SAS programming statements used in a DATA step. However, there are some differences in capabilities; for additional information, see also the section "Incompatibilities with SAS 6.11 and Earlier Versions of PROC NLIN" on page 4310.

Several SAS programming statements can be used after the PROC NLIN statement. These statements can appear anywhere in the PROC NLIN code, but new variables must be created before they appear in other statements. For example, the following statements are valid since they create the variable temp before it is used in the MODEL statement:

```
proc nlin;
   parms b0=0 to 2 by 0.5 b1=0.01 to 0.09 by 0.01;
   temp = exp(-b1*x);
   model y=b0*(1-temp);
```

The following statements result in missing values for y because the variable temp is undefined before it is used:

```
proc nlin;
   parms b0=0 to 2 by 0.5 b1=0.01 to 0.09 by 0.01;
   model y = b0*(1-temp);
   temp = exp(-b1*x);
```

PROC NLIN can process assignment statements, explicitly or implicitly subscripted ARRAY statements, explicitly or implicitly subscripted array references, IF statements, SAS functions, and program control statements. You can use programming statements to create new SAS variables for the duration of the procedure. These variables are not included in the data set to which PROC NLIN is applied. Program statements can include variables in the DATA= data set, parameter names, variables created by preceding programming statements within PROC NLIN, and special variables used by PROC NLIN. The following SAS programming statements can be used in PROC NLIN:

ARRAY;
variable = expression;
variable + expression;
CALL *name [(expression [, expression . . .])]*;
DO *[variable = expression*
*[***TO** *expression] [***BY** *expression]*
[, expression [**TO** *expression] [* **BY** *expression] . . .]*
]
[**WHILE** *expression] [* **UNTIL** *expression]*;
END;
FILE;
GOTO *statement_label*;
IF *expression*;
IF *expression* **THEN** *program_statement*;
ELSE *program_statement*;
variable = expression;
variable + expression;
LINK *statement_label*;
PUT *[variable] [=] [...]*;
RETURN;
SELECT[(*expression* **)];**

These statements can use the special variables created by PROC NLIN. See the section "Special Variables" on page 4297 for more information.

Details: NLIN Procedure

Automatic Derivatives

Depending on the optimization method you select, analytical first- and second-order derivatives are computed automatically. Derivatives can still be supplied using the DER.parm syntax. These DER.parm derivatives are not verified by the differentiator. If any needed derivatives are not supplied, they are computed and added to the programming statements. To view the computed derivatives, use the LISTDER or LIST option.

The following model is solved using Newton's method. Analytical first- and second-order derivatives are automatically computed. The compiled program code is shown in Figure 60.6.

```
proc nlin data=Enzyme method=newton list;
   parms x1=4 x2=2;
   model Velocity = x1 * exp (x2 * Concentration);
run;
```

Figure 60.6 Model and Derivative Code Output

```
                          The NLIN Procedure

                 Listing of Compiled Program Code
Stmt    Line:Col      Statement as Parsed

   1    1057:4        MODEL.Velocity = x1 * EXP(x2
                      * Concentration);
   1    1057:4        @MODEL.Velocity/@x1 = EXP(x2
                      * Concentration);
   1    1057:4        @MODEL.Velocity/@x2 = x1 * Concentration
                      * EXP(x2 * Concentration);
   1    1057:4        @@MODEL.Velocity/@x1/@x2 = Concentration
                      * EXP(x2 * Concentration);
   1    1057:4        @@MODEL.Velocity/@x2/@x1 = Concentration
                      * EXP(x2 * Concentration);
   1    1057:4        @@MODEL.Velocity/@x2/@x2 = x1
                      * Concentration * Concentration
                      * EXP(x2 * Concentration);
```

Note that all the derivatives require the evaluation of EXP(X2 * Concentration). The actual machine-level code is displayed if you specify the LISTCODE option, as in the following statements:

```
proc nlin data=Enzyme method=newton listcode;
   parms x1=4 x2=2;
   model Velocity = x1 * exp (x2 * Concentration);
run;
```

Note that, in the generated code, only one exponentiation is performed (Figure 60.7). The generated code reuses previous operations to be more efficient.

Figure 60.7 LISTCODE Output

```
                          The NLIN Procedure

                            Code Listing


1 Stmt MODEL      line 1064 column
                  4. (1)
                  arg=MODEL.Velocity
                  argsave=MODEL.
                  Velocity
                  Source Text:          model Velocity = x1 * exp
                                        (x2 * Concentration);
    Oper *        at 1064:34 (30,0,2).  * : _temp1 <- x2 Concentration
    Oper EXP      at 1064:30            EXP : _temp2 <- _temp1
                  (103,0,1).
    Oper *        at 1064:24 (30,0,2).  * : MODEL.Velocity <- x1 _temp2
    Oper eeocf    at 1064:24 (18,0,1).  eeocf : _DER_ <- _DER_
    Oper =        at 1064:24 (1,0,1).   = : @MODEL.Velocity/@x1 <- _temp2
    Oper *        at 1064:30 (30,0,2).  * : @1dt1_1 <- Concentration _temp2
    Oper *        at 1064:24 (30,0,2).  * : @MODEL.Velocity/@x2
                                        <- x1 @1dt1_1
    Oper =        at 1064:24 (1,0,1).   = : @@MODEL.Velocity/@x1/@x2
                                        <- @1dt1_1
    Oper =        at 1064:24 (1,0,1).   = : @@MODEL.Velocity/@x2/@x1
                                        <- @1dt1_1
    Oper *        at 1064:30 (30,0,2).  * : @2dt1_1 <- Concentration
                                        @1dt1_1
    Oper *        at 1064:24 (30,0,2).  * : @@MODEL.Velocity/@x2/@x2
                                        <- x1 @2dt1_1
```

Hougaard's Measure of Skewness

A "close-to-linear" nonlinear regression model, in the sense of Ratkowsky (1983, 1990), is a model in which parameter estimators have properties similar to those in a linear regression model. That is, the least-squares estimators of the parameters are close to being unbiased and normally distributed, and they have minimum variance.

A nonlinear regression model sometimes fails to be close to linear due to the properties of a single parameter. When this occurs, bias in the parameter estimates can render inferences that use the reported standard errors and confidence limits invalid. Ratkowsky (1990) notes that of the two curvature components in a nonlinear model—intrinsic curvature and parameter-effects curvature (Bates and Watts 1981)—the parameter-effects curvature is typically larger. It is this component that you can affect by changing the parameterization of a model. One motivation for fitting a nonlinear model in a different parameterization is to obtain a particular interpretation and to give parameter estimators more close-to-linear behavior. Example 60.4 shows how changing the parameterization of a four-parameter logistic model can reduce the parameter-effects curvature and can yield a useful parameter interpretation at the same time.

The degree to which a parameter estimator exhibits close-to-linear behavior can be assessed with Hougaard's measure of skewness, g_{1i} (Hougaard 1982, 1985). This measure is available through the HOUGAARD option in the PROC NLIN statement. Hougaard's skewness measure for the ith parameter is defined as follows:

$$\mathrm{E}\left[\widehat{\theta}_i - \mathrm{E}\left(\widehat{\theta}_i\right)\right]^3 = -\left(\sigma^2\right)^2 \sum_{jkl} [\mathbf{L}]_{ij}[\mathbf{L}]_{ik}[\mathbf{L}]_{il}\left(\left[\mathbf{W}_j\right]_{kl} + [\mathbf{W}_k]_{jl} + [\mathbf{W}_l]_{jk}\right)$$

where the sum is a triple sum over the number of parameters and

$$\mathbf{L} = \left(\mathbf{X}'\mathbf{X}\right)^{-1} = \left(\frac{\partial \mathbf{f}}{\partial \boldsymbol{\beta}'}\frac{\partial \mathbf{f}}{\partial \boldsymbol{\beta}}\right)^{-1}$$

The term $[\mathbf{L}]_{ij}$ denotes the value in row i, column j of the matrix $\mathbf{L}$. (Hougaard (1985) uses superscript notation to denote elements in this inverse.) The matrix $\mathbf{W}$ is a three-dimensional ($p \times p \times p$) array

$$\mathbf{W} = \left(\frac{\partial \mathbf{f}}{\partial \boldsymbol{\beta}'}\right)\mathbf{H}$$

$$\mathbf{H} = \frac{\partial^2 \mathbf{f}}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}'}$$

$$\left[\mathbf{W}_j\right]_{kl} = \sum_{m=1}^{n} \frac{\partial F_m}{\partial \beta_j}\frac{\partial^2 F_m}{\partial \beta_k \partial \beta_l}$$

The third central moment is then normalized using the standard error as

$$G_{1i} = \mathrm{E}\left[\widehat{\theta}_i - E(\widehat{\theta}_i)\right]^3 / \left(\sigma^2 \times [\mathbf{L}]_{ii}\right)^{3/2}$$

The previous expressions depend on the unknown values of the parameters and on the residual variance σ^2. In order to evaluate the Hougaard measure in a particular data set, the NLIN procedure computes

$$g_{1i} = \widehat{\mathrm{E}}\left[\widehat{\theta}_i - \mathrm{E}(\widehat{\theta}_i)\right]^3 / \left(\mathrm{mse} \times [\widehat{\mathbf{L}}]_{ii}\right)^{3/2}$$

$$\widehat{\mathrm{E}}\left[\widehat{\theta}_i - \mathrm{E}(\widehat{\theta}_i)\right]^3 = -\mathrm{mse}^2 \sum_{jkl} [\widehat{\mathbf{L}}]_{ij}[\widehat{\mathbf{L}}]_{ik}[\widehat{\mathbf{L}}]_{il}\left([\widehat{\mathbf{W}}_j]_{kl} + [\widehat{\mathbf{W}}_k]_{jl} + [\widehat{\mathbf{W}}_l]_{jk}\right)$$

Following Ratkowsky (1990, p. 28), the following rules can be applied to the standardized Hougaard measure computed by the NLIN procedure:

$\lvert g_{1i}\rvert < 0.1$	the estimator $\widehat{\theta}_i$ of parameter θ_i is very close to linear in behavior
$0.1 < \lvert g_{1i}\rvert < .25$	the estimator is reasonably close to linear
$\lvert g_{1i}\rvert > .25$	the skewness is very apparent
$\lvert g_{1i}\rvert > 1$	the nonlinear behavior is considerable

Missing Values

If the value of any one of the SAS variables involved in the model is missing from an observation, that observation is omitted from the analysis. If only the value of the dependent variable is missing, that observation has a predicted value calculated for it when you use an OUTPUT statement and specify the PREDICTED= option.

If an observation includes a missing value for one of the independent variables, both the predicted value and the residual value are missing for that observation. If the iterations fail to converge, the values for all variables named in the OUTPUT statement are set to missing.

Special Variables

Several special variables are created automatically and can be used in PROC NLIN programming statements.

Special Variables with Values That Are Set by PROC NLIN

The values of the following six special variables are set by PROC NLIN and should not be reset to a different value by programming statements:

ERROR is set to 1 if a numerical error or invalid argument to a function occurs during the current execution of the program. It is reset to 0 before each new execution.

ITER represents the current iteration number. The variable _ITER_ is set to −1 during the grid search phase.

MODEL is set to 1 for passes through the data when only the predicted values are needed, and not the derivatives. It is 0 when both predicted values and derivatives are needed. If your derivative calculations consume a lot of time, you can save resources by using the following statement after your MODEL statement but before your derivative calculations:

```
if _model_ then return;
```

The derivatives generated by PROC NLIN do this automatically.

_N_ indicates the number of times the PROC NLIN step has been executed. It is never reset for successive passes through the data set.

OBS indicates the observation number in the data set for the current program execution. It is reset to 1 to start each pass through the data set (unlike the _N_ variable).

SSE has the error sum of squares of the last iteration. During the grid search phase, the _SSE_ variable is set to 0. For iteration 0, the _SSE_ variable is set to the SSE associated with the point chosen from the grid search.

Special Variable Used to Determine Convergence Criteria

The special variable _LOSS_ can be used to determine the criterion function for convergence and step shortening. PROC NLIN looks for the variable _LOSS_ in the programming statements and, if it is defined, uses the (weighted) sum of this value instead of the residual sum of squares to determine the criterion function for convergence and step shortening. This feature is useful in certain types of maximum-likelihood estimation.

CAUTION: Even if you specify the _LOSS_ variable in your programming statements, the NLIN procedure continues to solve a least-squares problem. The specified _LOSS_ function does **not** define or alter the objective function for parameter estimation.

Weighted Regression with the Special Variable _WEIGHT_

To obtain weighted nonlinear least-squares estimates of parameters, make an assignment to the _WEIGHT_ variable as in the following statement:

```
_weight_ = expression;
```

When this statement is included, the expression on the right side of the assignment statement is evaluated for each observation in the data set. The values are then taken as inverse elements of the diagonal variance-covariance matrix of the dependent variable.

When a variable name is given after the equal sign, the values of the variable are taken as the inverse elements of the variance-covariance matrix. The larger the _WEIGHT_ value, the more importance the observation is given.

The _WEIGHT_ variable can be a function of the estimated parameters. For estimation purposes, the derivative of the _WEIGHT_ variable with respect to the parameters is not included in the gradient and the Hessian of the loss function. This is normally the desired approach for iteratively reweighted least-squares estimation. When the _WEIGHT_ variable is a function of the parameters, the gradient and the Hessian used can lead to poor convergence or nonconvergence of the requested estimation. To have the derivative of the _WEIGHT_ variable with respect to the parameters included in the gradient and the Hessian of the loss function, do not use the _WEIGHT_ variable. Instead, redefine the model as

$$(y - f(x, \beta)) \times \sqrt{wgt(\beta)}$$

where y is the original dependent variable, $f(x, \beta)$ is the nonlinear model, and $wgt(\beta)$ is the weight that is a function of the parameters.

If the _WEIGHT_= statement is not used, the default value of 1 is used, and regular least-squares estimates are obtained.

Troubleshooting

This section describes a number of problems that might occur in your analysis with PROC NLIN.

Excessive Computing Time

If you specify a grid of starting values that contains many points, the analysis might take excessive time since the procedure must go through the entire data set for each point on the grid.

The analysis might also take excessive time if your problem takes many iterations to converge, since each iteration requires as much time as a linear regression with predicted values and residuals calculated.

Dependencies

The matrix of partial derivatives can be singular, possibly indicating an overparameterized model. For example, if b0 starts at zero in the following model, the derivatives for b1 are all zero for the first iteration:

```
parms b0=0 b1=.022;
model pop=b0*exp(b1*(year-1790));
der.b0=exp(b1*(year-1790));
der.b1=(year-1790)*b0*exp(b1*(year-1790));
```

The first iteration changes a subset of the parameters; then the procedure can make progress in succeeding iterations. This singularity problem is local. The next example displays a global problem. The term b2 in the exponent is not identifiable since it trades roles with b0.

```
parms b0=3.9 b1=.022 b2=0;
model pop=b0*exp(b1*(year-1790)+b2);
der.b0 = exp(b1*(year-1790)+b2);
der.b1 = (year-1790)*b0*exp(b1*(year-1790)+b2);
der.b2 = b0*exp(b1*(year-1790)+b2);
```

Unable to Improve

The method can lead to steps that do not improve the estimates even after a series of step halvings. If this happens, the procedure issues a message stating that it is unable to make further progress, but it then displays the following warning message:

```
PROC NLIN failed to converge
```

Then it displays the results. This often means that the procedure has not converged at all. If you provided your own derivatives, check them carefully and then examine the residual sum of squares surface. If PROC NLIN has not converged, try a different set of starting values, a different METHOD= specification, the G4 option, or a different model.

Divergence

The iterative process might diverge, resulting in overflows in computations. It is also possible that parameters enter a space where arguments to such functions as LOG and SQRT become invalid. For example, consider the following model:

```
parms b=0;
model y = x / b;
```

Suppose that y contains only zeros, and suppose that the values for variable x are not zero. There is no least-squares estimate for b since the SSE declines as b approaches infinity or minus infinity. To avoid the problem, the same model could be parameterized as **y = a*x**.

If you have divergence problems, try reparameterizing the model, selecting different starting values, increasing the maximum allowed number of iterations (the MAXITER= option), specifying an alternative METHOD= option, or including a BOUNDS statement.

Local Minimum

The program might converge to a local rather than a global minimum. For example, consider the following model:

```
parms a=1 b=-1;
model y=(1-a*x)*(1-b*x);
```

Once a solution is found, an equivalent solution with the same SSE can be obtained by swapping the values of a and b.

Discontinuities

The computational methods assume that the model is a continuous and smooth function of the parameters. If this is not true, the method does not work. For example, the following models do not work:

```
model y=a+int(b*x);
model y=a+b*x+4*(z>c);
```

Responding to Trouble

PROC NLIN does not necessarily produce a good solution the first time. Much depends on specifying good initial values for the parameters. You can specify a grid of values in the PARMS statement to search for good starting values. While most practical models should give you no trouble, other models can require switching to a different iteration method or a different computational method for matrix inversion. Specifying the option METHOD=MARQUARDT sometimes works when the default method (Gauss-Newton) does not work.

Computational Methods

Nonlinear Least Squares

Recall the basic notation for the nonlinear regression model from the section "Notation for Nonlinear Regression Models" on page 4271. The parameter vector $\boldsymbol{\beta}$ belongs to $\boldsymbol{\Omega}$, a subset of R^p. Two points of this set are of particular interest: the true value $\widetilde{\boldsymbol{\beta}}$ and the least-squares estimate $\widehat{\boldsymbol{\beta}}$. The general nonlinear model fit with the NLIN procedure is represented by the equation

$$\mathbf{Y} = \mathbf{f}(\widetilde{\beta_0}, \widetilde{\beta_1}, \ldots, \widetilde{\beta_p}; \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k) + \boldsymbol{\epsilon} = \mathbf{f}(\widetilde{\boldsymbol{\beta}}; \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k) + \boldsymbol{\epsilon}$$

where $\mathbf{z}_j$ denotes the $(n \times 1)$ vector of the jth regressor (independent) variable, $\widetilde{\boldsymbol{\beta}}$ is the true value of the parameter vector, and $\boldsymbol{\epsilon}$ is the $(n \times 1)$ vector of homoscedastic and uncorrelated model errors with zero mean.

To write the model for the ith observation, the ith elements of $\mathbf{z}_1, \cdots, \mathbf{z}_k$ are collected in the row vector $\mathbf{z}_i'$, and the model equation becomes

$$Y_i = f(\boldsymbol{\beta}; \mathbf{z}_i') + \epsilon_i$$

The shorthand $f_i(\boldsymbol{\beta})$ will also be used throughout to denote the mean of the ith observation.

For any given value $\boldsymbol{\beta}$ we can compute the residual sum of squares

$$\begin{aligned} \text{SSE}(\boldsymbol{\beta}) &= \sum_{i=1}^{n} \left(y_i - f\left(\boldsymbol{\beta}; \mathbf{z}_i'\right)\right)^2 \\ &= \sum_{i=1}^{n} \left(y_i - f_i\left(\boldsymbol{\beta}\right)\right)^2 = \mathbf{r}(\boldsymbol{\beta})'\mathbf{r}(\boldsymbol{\beta}) \end{aligned}$$

The aim of nonlinear least-squares estimation is to find the value $\widehat{\boldsymbol{\beta}}$ that minimizes $\text{SSE}(\boldsymbol{\beta})$. Because $\mathbf{f}$ is a nonlinear function of $\boldsymbol{\beta}$, a closed-form solution does not exist for this minimization problem. An iterative process is used instead. The iterative techniques that PROC NLIN uses are similar to a series of linear regressions involving the matrix $\mathbf{X}$ and the residual vector $\mathbf{r} = \mathbf{y} - \mathbf{f}(\boldsymbol{\beta})$, evaluated at the current values of $\boldsymbol{\beta}$.

It is more insightful, however, to describe the algorithms in terms of their approach to minimizing the residual sum of squares and in terms of their updating formulas. If $\widehat{\boldsymbol{\beta}}^{(u)}$ denotes the value of the parameter estimates at the uth iteration, and $\widehat{\boldsymbol{\beta}}^{(0)}$ are your starting values, then the NLIN procedure attempts to find values k and $\boldsymbol{\Delta}$ such that

$$\widehat{\boldsymbol{\beta}}^{(u+1)} = \widehat{\boldsymbol{\beta}}^{(u)} + k\boldsymbol{\Delta}$$

and

$$\text{SSE}\left(\widehat{\boldsymbol{\beta}}^{(u+1)}\right) < \text{SSE}\left(\widehat{\boldsymbol{\beta}}^{(u)}\right)$$

The various methods to fit a nonlinear regression model—which you can select with the METHOD= option in the PROC NLIN statement—differ in the calculation of the update vector $\boldsymbol{\Delta}$.

The gradient and Hessian of the residual sum of squares with respect to individual parameters and pairs of parameters are, respectively,

$$\mathbf{g}(\beta_j) = \frac{\partial \text{SSE}(\boldsymbol{\beta})}{\partial \beta_j} = -2\sum_{i=1}^{n}(y_i - f_i(\boldsymbol{\beta}))\frac{\partial f_i(\boldsymbol{\beta})}{\partial \beta_j}$$

$$[\mathbf{H}(\boldsymbol{\beta})]_{jk} = \frac{\partial \text{SSE}(\boldsymbol{\beta})}{\partial \beta_j \partial \beta_k} = 2\sum_{i=1}^{n}\frac{\partial f_i(\boldsymbol{\beta})}{\partial \beta_j}\frac{\partial f_i(\boldsymbol{\beta})}{\partial \beta_k} - (y_i - f_i(\boldsymbol{\beta}))\frac{\partial^2 f_i(\boldsymbol{\beta})}{\partial \beta_j \partial \beta_k}$$

Denote as $\mathbf{H}_i^*(\boldsymbol{\beta})$ the Hessian matrix of the mean function,

$$\left[\mathbf{H}_i^*(\boldsymbol{\beta})\right]_{jk} = \left[\frac{\partial^2 f_i(\boldsymbol{\beta})}{\partial \beta_j \partial \beta_k}\right]_{jk}$$

Collecting the derivatives across all parameters leads to the expressions

$$\mathbf{g}(\boldsymbol{\beta}) = \frac{\partial \text{SSE}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -2\mathbf{X}'\mathbf{r}(\boldsymbol{\beta})$$

$$\mathbf{H}(\boldsymbol{\beta}) = \frac{\partial^2 \text{SSE}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}\partial \boldsymbol{\beta}'} = 2\left(\mathbf{X}'\mathbf{X} - \sum_{i=1}^{n} r_i(\boldsymbol{\beta})\mathbf{H}_i^*(\boldsymbol{\beta})\right)$$

The change in the vector of parameter estimates is computed as follows, depending on the estimation method:

$$\text{Gauss-Newton: } \boldsymbol{\Delta} = (-\text{E}[\mathbf{H}(\boldsymbol{\beta})])^{-}\mathbf{g}(\boldsymbol{\beta}) = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{r}$$

$$\text{Marquardt: } \boldsymbol{\Delta} = \left(\mathbf{X}'\mathbf{X} + \lambda \text{diag}(\mathbf{X}'\mathbf{X})\right)^{-}\mathbf{X}'\mathbf{r}$$

$$\text{Newton: } \boldsymbol{\Delta} = -\mathbf{H}(\boldsymbol{\beta})^{-}\mathbf{g}(\boldsymbol{\beta}) = \mathbf{H}(\boldsymbol{\beta})^{-}\mathbf{X}'\mathbf{r}$$

$$\text{Steepest descent: } \boldsymbol{\Delta} = -\frac{1}{2}\frac{\partial \text{SSE}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \mathbf{X}'\mathbf{r}$$

The Gauss-Newton and Marquardt iterative methods regress the residuals onto the partial derivatives of the model with respect to the parameters until the estimates converge. You can view the Marquardt algorithm as a Gauss-Newton algorithm with a ridging penalty. The Newton iterative method regresses the residuals onto a function of the first and second derivatives of the model with respect to the parameters until the estimates converge. Analytical first- and second-order derivatives are automatically computed as needed.

The default method used to compute $(\mathbf{X}'\mathbf{X})^{-}$ is the sweep (Goodnight 1979). It produces a reflexive generalized inverse (a g_2-inverse, Pringle and Rayner, 1971). In some cases it might be preferable to use a Moore-Penrose inverse (a g_4-inverse) instead. If you specify the G4 option in the PROC NLIN statement, a g_4-inverse is used to calculate $\boldsymbol{\Delta}$ on each iteration.

The four algorithms are now described in greater detail.

Algorithmic Details

Gauss-Newton and Newton Methods

From the preceding set of equations you can see that the Marquardt method is a ridged version of the Gauss-Newton method. If the ridge parameter λ equals zero, the Marquardt step is identical to the Gauss-Newton step. An important difference between the Newton methods and the Gauss-Newton-type algorithms lies in the use of second derivatives. To motivate this distinctive element between Gauss-Newton and the Newton method, focus first on the objective function in nonlinear least squares. To numerically find the minimum of

$$\text{SSE}(\boldsymbol{\beta}) = \mathbf{r}(\boldsymbol{\beta})'\mathbf{r}(\boldsymbol{\beta})$$

you can approach the problem by approximating the sum of squares criterion by a criterion for which you can compute a closed-form solution. Following Seber and Wild (1989, Sect. 2.1.3), we can achieve that by doing the following:

- approximating the model and substituting the approximation into $\text{SSE}(\boldsymbol{\beta})$
- approximating $\text{SSE}(\boldsymbol{\beta})$ directly

The first method, approximating the nonlinear model with a first-order Taylor series, is the purview of the Gauss-Newton method. Approximating the residual sum of squares directly is the realm of the Newton method.

The first-order Taylor series of the residual $\mathbf{r}(\boldsymbol{\beta})$ at the point $\widehat{\boldsymbol{\beta}}$ is

$$\mathbf{r}(\boldsymbol{\beta}) \approx \mathbf{r}(\widehat{\boldsymbol{\beta}}) - \widehat{\mathbf{X}}\left(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\right)$$

Substitution into $\text{SSE}(\boldsymbol{\beta})$ leads to the objective function for the Gauss-Newton step:

$$\text{SSE}\,(\boldsymbol{\beta}) \approx S_G(\boldsymbol{\beta}) = \mathbf{r}(\widehat{\boldsymbol{\beta}})' - 2\mathbf{r}'(\widehat{\boldsymbol{\beta}})\widehat{\mathbf{X}}\left(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\right) + \left(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\right)'\left(\widehat{\mathbf{X}}'\widehat{\mathbf{X}}\right)\left(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\right)$$

"Hat" notation is used here to indicate that the quantity in question is evaluated at $\widehat{\boldsymbol{\beta}}$.

To motivate the Newton method, take a second-order Taylor series of $\text{SSE}(\boldsymbol{\beta})$ around the value $\widehat{\boldsymbol{\beta}}$:

$$\text{SSE}\,(\boldsymbol{\beta}) \approx S_N(\boldsymbol{\beta}) = \text{SSE}\left(\widehat{\boldsymbol{\beta}}\right) + \mathbf{g}(\widehat{\boldsymbol{\beta}})'\left(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\right) + \frac{1}{2}\left(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\right)'\mathbf{H}(\widehat{\boldsymbol{\beta}})\left(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\right)$$

Both $S_G(\boldsymbol{\beta})$ and $S_N(\boldsymbol{\beta})$ are quadratic functions in $\boldsymbol{\beta}$ and are easily minimized. The minima occur when

$$\text{Gauss-Newton: } \boldsymbol{\beta} - \widehat{\boldsymbol{\beta}} = \left(\widehat{\mathbf{X}}'\widehat{\mathbf{X}}\right)^{-}\widehat{\mathbf{X}}'\mathbf{r}(\widehat{\boldsymbol{\beta}})$$

$$\text{Newton : } \boldsymbol{\beta} - \widehat{\boldsymbol{\beta}} = -\mathbf{H}(\widehat{\boldsymbol{\beta}})^{-}\mathbf{g}(\widehat{\boldsymbol{\beta}})$$

and these terms define the preceding $\boldsymbol{\Delta}$ update vectors.

Gauss-Newton Method

Since the Gauss-Newton method is based on an approximation of the model, you can also derive the update vector by first considering the "normal" equations of the nonlinear model

$$\mathbf{X}'\mathbf{f}(\boldsymbol{\beta}) = \mathbf{X}'\mathbf{Y}$$

and then substituting the Taylor approximation

$$\mathbf{f}(\boldsymbol{\beta}) \approx \mathbf{f}(\widehat{\boldsymbol{\beta}}) + \mathbf{X}\left(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}\right)$$

for $\mathbf{f}(\boldsymbol{\beta})$. This leads to

$$\begin{aligned}
\mathbf{X}'\left(\mathbf{f}(\widehat{\boldsymbol{\beta}}) + \mathbf{X}(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}})\right) &= \mathbf{X}'\mathbf{Y} \\
(\mathbf{X}'\mathbf{X})(\boldsymbol{\beta} - \widehat{\boldsymbol{\beta}}) &= \mathbf{X}'\mathbf{Y} - \mathbf{X}'\mathbf{f}(\widehat{\boldsymbol{\beta}}) \\
(\mathbf{X}'\mathbf{X})\boldsymbol{\Delta} &= \mathbf{X}'\mathbf{r}(\widehat{\boldsymbol{\beta}})
\end{aligned}$$

and the update vector becomes

$$\boldsymbol{\Delta} = \left(\mathbf{X}'\mathbf{X}\right)^{-}\mathbf{X}'\mathbf{r}(\widehat{\boldsymbol{\beta}})$$

CAUTION: If $\mathbf{X}'\mathbf{X}$ is singular or becomes singular, PROC NLIN computes $\boldsymbol{\Delta}$ by using a generalized inverse for the iterations after singularity occurs. If $\mathbf{X}'\mathbf{X}$ is still singular for the last iteration, the solution should be examined.

Newton Method

The Newton method uses the second derivatives and solves the equation

$$\boldsymbol{\Delta} = \mathbf{H}^{-}\mathbf{X}'\mathbf{r}$$

If the automatic variables _WEIGHT_, _WGTJPJ_, and _RESID_ are used, then

$$\boldsymbol{\Delta} = \mathbf{H}^{-}\mathbf{X}'\mathbf{W}^{SSE}\mathbf{r}^{*}$$

is the direction, where

$$\mathbf{H} = \mathbf{X}'\mathbf{W}^{XPX}\mathbf{X} - \sum_{i=1}^{n}\mathbf{H}_i^{*}(\boldsymbol{\beta})w_i^{XPX}r_i^{*}$$

and

$\mathbf{W}^{SSE}$ is an $n \times n$ diagonal matrix with elements w_i^{SSE} of weights from the _WEIGHT_ variable. Each element w_i^{SSE} contains the value of _WEIGHT_ for the ith observation.

$\mathbf{W}^{XPX}$ is an $n \times n$ diagonal matrix with elements w_i^{XPX} of weights from the _WGTJPJ_ variable. Each element w_i^{XPX} contains the value of _WGTJPJ_ for the ith observation.

$\mathbf{r}^{*}$ is a vector with elements r_i^{*} from the _RESID_ variable. Each element r_i^{*} contains the value of _RESID_ evaluated for the ith observation.

Marquardt Method

The updating formula for the Marquardt method is as follows:

$$\mathbf{\Delta} = (\mathbf{X}'\mathbf{X} + \lambda \text{diag}(\mathbf{X}'\mathbf{X}))^{-}\mathbf{X}'\mathbf{e}$$

The Marquardt method is a compromise between the Gauss-Newton and steepest descent methods (Marquardt 1963). As $\lambda \to 0$, the direction approaches Gauss-Newton. As $\lambda \to \infty$, the direction approaches steepest descent.

Marquardt's studies indicate that the average angle between Gauss-Newton and steepest descent directions is about 90°. A choice of λ between 0 and infinity produces a compromise direction.

By default, PROC NLIN chooses $\lambda = 10^{-7}$ to start and computes $\mathbf{\Delta}$. If $\text{SSE}(\boldsymbol{\beta}_0 + \mathbf{\Delta}) < \text{SSE}(\boldsymbol{\beta}_0)$, then $\lambda = \lambda/10$ for the next iteration. Each time $\text{SSE}(\boldsymbol{\beta}_0 + \mathbf{\Delta}) > \text{SSE}(\boldsymbol{\beta}_0)$, then $\lambda = 10\lambda$.

NOTE: If the SSE decreases on each iteration, then $\lambda \to 0$, and you are essentially using the Gauss-Newton method. If SSE does not improve, then λ is increased until you are moving in the steepest descent direction.

Marquardt's method is equivalent to performing a series of ridge regressions, and it is useful when the parameter estimates are highly correlated or the objective function is not well approximated by a quadratic.

Steepest Descent (Gradient) Method

The steepest descent method is based directly on the gradient of $0.5\mathbf{r}(\boldsymbol{\beta})'\mathbf{r}(\boldsymbol{\beta})$:

$$\frac{1}{2}\frac{\partial \text{SSE}(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = -\mathbf{X}'\mathbf{r}$$

The quantity $-\mathbf{X}'\mathbf{r}$ is the gradient along which $\boldsymbol{\epsilon}'\boldsymbol{\epsilon}$ increases. Thus $\mathbf{\Delta} = \mathbf{X}'\mathbf{r}$ is the direction of steepest descent.

If the automatic variables _WEIGHT_ and _RESID_ are used, then

$$\mathbf{\Delta} = \mathbf{X}'\mathbf{W}^{SSE}\mathbf{r}^*$$

is the direction, where

$\mathbf{W}^{SSE}$ is an $n \times n$ diagonal matrix with elements w_i^{SSE} of weights from the _WEIGHT_ variable. Each element w_i^{SSE} contains the value of _WEIGHT_ for the ith observation.

$\mathbf{r}^*$ is a vector with elements r_i^* from _RESID_. Each element r_i^* contains the value of _RESID_ evaluated for the ith observation.

Using the method of steepest descent, let

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \alpha\mathbf{\Delta}$$

where the scalar α is chosen such that

$$\text{SSE}(\boldsymbol{\beta}_i + \alpha\mathbf{\Delta}) < \text{SSE}(\boldsymbol{\beta}_i)$$

CAUTION: The steepest-descent method can converge very slowly and is therefore not generally recommended. It is sometimes useful when the initial values are poor.

Step-Size Search

The default method of finding the step size k is step halving by using SMETHOD=HALVE. If $\text{SSE}(\boldsymbol{\beta}^{(u)} + \boldsymbol{\Delta}) > \text{SSE}(\boldsymbol{\beta}^{(u)})$, compute $\text{SSE}(\boldsymbol{\beta}^{(u)} + 0.5\boldsymbol{\Delta})$, $\text{SSE}(\boldsymbol{\beta}^{(u)} + 0.25\boldsymbol{\Delta}), \ldots$, until a smaller SSE is found.

If you specify SMETHOD=GOLDEN, the step size k is determined by a golden section search. The parameter TAU determines the length of the initial interval to be searched, with the interval having length TAU (or $2 \times$ TAU), depending on $\text{SSE}(\boldsymbol{\beta}^{(u)} + \boldsymbol{\Delta})$. The RHO parameter specifies how fine the search is to be. The SSE at each endpoint of the interval is evaluated, and a new subinterval is chosen. The size of the interval is reduced until its length is less than RHO. One pass through the data is required each time the interval is reduced. Hence, if RHO is very small relative to TAU, a large amount of time can be spent determining a step size. For more information about the golden section search, see Kennedy and Gentle (1980).

If you specify SMETHOD=CUBIC, the NLIN procedure performs a cubic interpolation to estimate the step size. If the estimated step size does not result in a decrease in SSE, step halving is used.

Output Data Sets

The data set produced by the OUTEST= option in the PROC NLIN statement contains the parameter estimates on each iteration, including the grid search.

The variable _ITER_ contains the iteration number. The variable _TYPE_ denotes whether the observation contains iteration parameter estimates ("ITER"), final parameter estimates ("FINAL"), or covariance estimates ("COVB"). The variable _NAME_ contains the parameter name for covariances, and the variable _SSE_ contains the objective function value for the parameter estimates. The variable _STATUS_ indicates whether the estimates have converged.

The data set produced by the OUTPUT statement contains statistics calculated for each observation. In addition, the data set contains the variables from the input data set and any ID variables that are specified in the ID statement.

Confidence Intervals

Parameter Confidence Intervals

The parameter confidence intervals are computed using the Wald-based formula:

$$\widehat{\beta}_i \pm \text{stderr}_i \times t(n - p, 1 - \alpha/2)$$

where $\widehat{\beta}_i$ is the ith parameter estimate, stderr_i is its estimated approximate standard error, $t(n - p, 1 - \alpha/2)$ is a t statistic with $n - p$ degrees of freedom, n is the number of observations, and p is the number of parameters. The confidence intervals are only asymptotically valid. The significance

level α used in the construction of these confidence limits can be set with the ALPHA= option in the PROC NLIN statement; the default value is $\alpha = 0.05$.

Model Confidence Intervals

Model confidence intervals are output when an OUT= data set is specified and one or more of the keywords LCLM, UCLM, LCL, UCL, L95M=, U95M=, L95=, and U95= is specified. The expressions for these terms are as follows:

$$
\begin{aligned}
\text{LCLM} &= f(\boldsymbol{\beta}, \mathbf{z}_i) - \sqrt{MSE \times h_i / w_i} \times t(n-p, 1-\alpha/2) \\
\text{UCLM} &= f(\boldsymbol{\beta}, \mathbf{z}_i) + \sqrt{MSE \times h_i / w_i} \times t(n-p, 1-\alpha/2) \\
\text{LCL} &= f(\boldsymbol{\beta}, \mathbf{z}_i) - \sqrt{MSE(h_i + 1/w_i)} \times t(n-p, 1-\alpha/2) \\
\text{UCL} &= f(\boldsymbol{\beta}, \mathbf{z}_i) + \sqrt{MSE(h_i + 1/w_i)} \times t(n-p, 1-\alpha/2) \\
\text{L95M} &= f(\boldsymbol{\beta}, \mathbf{z}_i) - \sqrt{MSE \times h_i / w_i} \times t(n-p, 1-0.05/2) \\
\text{U95M} &= f(\boldsymbol{\beta}, \mathbf{z}_i) + \sqrt{MSE \times h_i / w_i} \times t(n-p, 1-0.05/2) \\
\text{L95} &= f(\boldsymbol{\beta}, \mathbf{z}_i) - \sqrt{MSE(h_i + 1/w_i)} \times t(n-p, 1-0.05/2) \\
\text{U95} &= f(\boldsymbol{\beta}, \mathbf{z}_i) + \sqrt{MSE(h_i + 1/w_i)} \times t(n-p, 1-0.05/2)
\end{aligned}
$$

where $h_i = w_i \mathbf{x}_i (\mathbf{X}'\mathbf{W}\mathbf{X})^{-1} \mathbf{x}_i'$ is the leverage, $\mathbf{X} = \partial \mathbf{f} / \partial \boldsymbol{\beta}$, and $\mathbf{x}_i$ is the ith row of $\mathbf{X}$. These results are derived for linear systems. The intervals are approximate for nonlinear models. The value α in the preceding formulas for LCLM, UCLM, LCL, and UCL can be set with the ALPHA= option in the PROC NLIN statement or with the ALPHA= option in the OUTPUT statement. If both ALPHA= options are specified, the option in the OUTPUT takes precedence.

Covariance Matrix of Parameter Estimates

For unconstrained estimates (no active bounds), the covariance matrix of the parameter estimates is

$$\text{mse} \times (\mathbf{X}'\mathbf{X})^{-1}$$

for the gradient, Marquardt, and Gauss methods and

$$\text{mse} \times \mathbf{H}^{-1}$$

for the Newton method. Recall that $\mathbf{X}$ is the matrix of the first partial derivatives of the nonlinear model with respect to the parameters. The matrices are evaluated at the final parameter estimates. The mean squared error, the estimate of the residual variance σ^2, is computed as

$$\text{mse} = \mathbf{r}'\mathbf{r}/(n-p)$$

where n is the number of nonmissing (used) observations and p is the number of estimable parameters. The standard error reported for the parameter estimates is the square root of the corresponding

diagonal element of this matrix. If you specify a value for the residual variance with the SIGSQ= option, then that value replaces mse in the preceding expressions.

Now suppose that restrictions or bounds are active. Equality restrictions can be written as a vector function, $h(\boldsymbol{\theta}) = \mathbf{0}$. Inequality restrictions are either active or inactive. When an inequality restriction is active, it is treated as an equality restriction.

Assume that the vector $h(\boldsymbol{\theta})$ contains the current active restrictions. The constraint matrix $\mathbf{A}$ is then

$$\mathbf{A}(\widehat{\boldsymbol{\theta}}) = \frac{\partial h(\widehat{\boldsymbol{\theta}})}{\partial \widehat{\boldsymbol{\theta}}}$$

The covariance matrix for the restricted parameter estimates is computed as

$$\mathbf{Z}(\mathbf{Z}'\mathbf{H}\mathbf{Z})^{-1}\mathbf{Z}'$$

where $\mathbf{H}$ is the Hessian (or approximation to the Hessian) and $\mathbf{Z}$ collects the last $(p-n_c)$ columns of $\mathbf{Q}$ from an LQ factorization of the constraint matrix. Further, n_c is the number of active constraints, and p denotes the number of parameters. See Gill, Murray, and Wright (1981) for more details about the LQ factorization. The covariance matrix for the Lagrange multipliers is computed as

$$\left(\mathbf{A}\mathbf{H}^{-1}\mathbf{A}'\right)^{-1}$$

Convergence Measures

The NLIN procedure computes and reports four convergence measures, labeled R, PPC, RPC, and OBJECT.

R is the primary convergence measure for the parameters. It measures the degree to which the residuals are orthogonal to the columns of $\mathbf{X}$, and it approaches 0 as the gradient of the objective function becomes small. R is defined as

$$\sqrt{\frac{\mathbf{r}'\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{r}}{\text{LOSS}^i}}$$

PPC is the prospective parameter change measure. PPC measures the maximum relative change in the parameters implied by the parameter-change vector computed for the next iteration. At the kth iteration, PPC is the maximum over the parameters

$$\frac{|\widehat{\theta}_i^{(k+1)} - \widehat{\theta}_i^{(k)}|}{|\widehat{\theta}_i^{(k)}| + 1\text{E}-6}$$

where $\widehat{\theta}_i^{(k)}$ is the current value of the ith parameter and $\widehat{\theta}_i^{(k+1)}$ is the prospective value of this parameter after adding the change vector computed for the next iteration. These changes are measured before step length adjustments are made. The parameter with the maximum prospective relative change is displayed with the value of PPC, unless the PPC is nearly 0.

RPC is the retrospective parameter change measure. RPC measures the maximum relative change in the parameters from the previous iteration. At the kth iteration, RPC is the maximum over i of

$$\frac{|\widehat{\theta}_i^{(k)} - \widehat{\theta}_i^{(k-1)}|}{|\widehat{\theta}_i^{(k-1)}| + 1\mathrm{E}-6}$$

where $\widehat{\theta}_i^{(k)}$ is the current value of the ith parameter and $\widehat{\theta}_i^{k-1}$ is the previous value of this parameter. These changes are measured before step length adjustments are made. The name of the parameter with the maximum retrospective relative change is displayed with the value of RPC, unless the RPC is nearly 0.

OBJECT measures the relative change in the objective function value between iterations:

$$\frac{|O^{(k)} - O^{(k-1)}|}{|O^{(k-1)} + 1\mathrm{E}-6|}$$

where $O^{(k-1)}$ is the value of the objective function ($O^{(k)}$) from the previous iteration. This is the old CONVERGEOBJ= criterion.

Displayed Output

In addition to the output data sets, PROC NLIN also produces the following output objects:

- the residual sums of squares associated with all or some of the combinations of possible starting values of the parameters
- the estimates of the parameters and the residual sums of squares at each iteration
- the estimation summary table, which displays information about the estimation method, the number of observations in the analysis, the objective function, and convergence measures
- the analysis of variance table, including sums of squares for the "Model," "Residual," and "Total" sources of variation ("Corrected Total" or "Uncorrected Total"), and the model F test. Note that beginning in SAS® 9, only the uncorrected total SS is reported and the respective F test is based on the uncorrected total SS if PROC NLIN determines the model does not include an intercept. If PROC NLIN determines the model does include an intercept, only the corrected total SS is reported and the respective F test is based on the corrected total SS.
- the table of parameter estimates, which contains for each parameter in the model its estimate, the approximate standard error of the estimate, and a 95% confidence interval based on the approximate standard error. The confidence level can be changed with the ALPHA= option in the PROC NLIN statement. The HOUGAARD option in the PROC NLIN statement requests that Hougaard's skewness measure be added for each parameter. The standard errors and confidence limits are labeled approximate because they are valid asymptotically as the number of observations grows. If your model is linear in the parameters, the standard errors and confidence intervals are not approximate.

- the approximate correlation matrix of the parameter estimates. This correlation matrix is labeled approximate because it is computed from the approximate covariance matrix of the parameter estimates. If your model is linear in the parameters, the correlation matrix is not approximate.

Incompatibilities with SAS 6.11 and Earlier Versions of PROC NLIN

The NLIN procedure now uses a compiler that is different from the DATA step compiler. The compiler was changed so that analytical derivatives could be computed automatically. For the most part, the syntax accepted by the old NLIN procedure can be used in the new NLIN procedure. However, there are several differences that should be noted:

- You cannot specify a character index variable in the DO statement, and you cannot specify a character test in the IF statement. Thus **`do i=1,2,3;`** is supported, but **`do i='ONE','TWO','THREE';`** is not supported. And **`if 'THIS' < 'THAT' then ...;`** is supported, but **`if 'THIS' THEN ...;`** is not supported.
- The PUT statement, which is used mostly for program debugging in PROC NLIN, supports only some of the features of the DATA step PUT statement, and it has some new features that the DATA step PUT statement does not.
  - The PUT statement does not support line pointers, factored lists, iteration factors, overprinting, the _INFILE_ option, the ':' format modifier, or the symbol '$'.
  - The PUT statement does support expressions inside of parentheses. For example, **`put (sqrt(X));`** produces the square root of X.
  - The PUT statement also supports the option _PDV_ to display a formatted listing of all the variables in the program. The statement **`put _pdv_;`** prints a much more readable listing of the variables than **`put _all_;`** does.
- You cannot use the '*' subscript, but you can specify an array name in a PUT statement without subscripts. Thus, **`array a ...; put a;`** is acceptable, but **`put a[*];`** is not. The statement **`put a;`** displays all the elements of the array a. The **`put a=;`** statement displays all the elements of A with each value labeled by the name of the element variable.
- You cannot specify arguments in the ABORT statement.
- You can specify more than one target statement in the WHEN and OTHERWISE statements. That is, DO/END groups are not necessary for multiple WHEN statements, such as **`select; when(exp1); stmt1; stmt2; when(exp2); stmt3; stmt4; end;`**.
- You can specify only the options LOG, PRINT, and LIST in the FILE statement.
- The RETAIN statement retains only values across one pass through the data set. If you need to retain values across iterations, use the CONTROL statement to make a control variable.

The ARRAY statement in PROC NLIN is similar to, but not the same as, the ARRAY statement in the DATA step. The ARRAY statement is used to associate a name (of no more than 8 characters) with a list of variables and constants. The array name can then be used with subscripts in the program to refer to the items in the list.

The ARRAY statement supported by PROC NLIN does not support all the features of the DATA step ARRAY statement. You cannot specify implicit indexing variables; all array references must have explicit subscript expressions. You can specify simple array dimensions; lower bound specifications are not supported. A maximum of six dimensions are accepted.

On the other hand, the ARRAY statement supported by PROC NLIN does accept both variables and constants as array elements. In the following statements, `b` is a constant array and `c` is a variable array. Note that the constant array elements cannot be changed with assignment statements.

```
proc nlin data=nld;
array b[4] 1 2 3 4;      /* Constant array */
array c[4] ( 1 2 3 4 ); /* Numeric array with initial values */

b[1] = 2;                /* This is an ERROR, b is a constant array*/
c[2] = 7.5;              /* This is allowed */
```

Both dimension specification and the list of elements are optional, but at least one must be specified. When the list of elements is not specified, or fewer elements than the size of the array are listed, array variables are created by suffixing element numbers to the array name to complete the element list.

If the array is used as a pure array in the program rather than a list of symbols (the individual symbols of the array are not referenced in the code), the array is converted to a numerical array. A pure array is literally a vector of numbers that are accessed only by index. Using these types of arrays results in faster derivatives and compiled code. The assignment to `c1` in the following statements forces the array to be treated as a list of symbols:

```
proc nlin data=nld;
array c[4] ( 1 2 3 4 ); /* Numeric array with initial values */

c[2] = 7.5;              /* This is C used as a pure array  */
c1 = -92.5;              /* This forces C to be a list of symbols */
```

ODS Table Names

PROC NLIN assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 60.3. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 60.3 ODS Tables Produced by PROC NLIN

ODS Table Name	Description	Statement
ANOVA	Analysis of variance	default
CodeDependency	Variable cross reference	LISTDEP
CodeList	Listing of program statements	LISTCODE
ConvergenceStatus	Convergence status	default
CorrB	Correlation of the parameters	default
EstSummary	Summary of the estimation	default
FirstDerivatives	First derivative table	LISTDER
IterHistory	Iteration output	default
MissingValues	Missing values generated by the program	default
ParameterEstimates	Parameter estimates	default
ProgList	Listing of the compiled program	LIST

Convergence Status Table

The "Convergence Status" table can be used to programmatically check the status of an estimation. This table contains the Status variable that takes on the value 0, 1, 2, or 3. If Status takes on a value less than 3, the convergence criterion was met. Specifically, the values mean the following:

Status=0 indicates that the convergence criterion was met and no warning or error messages were issued during the PROC NLIN run. Also, no notes that could indicate a problem with the model were issued.

Status=1 indicates that the convergence criterion was met and notes were written to the log that might indicate a problem with the model.

Status=2 indicates that the convergence criterion was met and one or more warning messages were produced during the PROC NLIN run.

Status=3 indicates that the convergence criterion was not met.

The following sample program demonstrates how the "Convergence Status" table can be used:

```
ods output ConvergenceStatus=ConvStatus;
proc nlin data=YourData;
   parameters a=1 b=1 c=1;
   model wgt = a + x / (b*y+c*z);
run;

data _NULL_; set ConvStatus;
  if status > 0 then put "A problem occurred";
run;
```

Examples: NLIN Procedure

Example 60.1: Segmented Model

Suppose you are interested in fitting a model that consists of two segments that connect in a smooth fashion. For example, the following model states that for values of x less than x_0 the mean of Y is a quadratic function in x, and for values of x greater than x_0 the mean of Y is constant:

$$\mathrm{E}[Y|x] = \begin{cases} \alpha + \beta x + \gamma x^2 & \text{if } x < x_0 \\ c & \text{if } x \geq x_0 \end{cases}$$

In this model equation α, β, and γ are the coefficients of the quadratic segment, and c is the plateau of the mean function. The NLIN procedure can fit such a segmented model even when the join point, x_0, is unknown.

We also want to impose conditions on the two segments of the model. First, the curve should be continuous—that is, the quadratic and the plateau section need to meet at x_0. Second, the curve should be smooth—that is, the first derivative of the two segments with respect to x need to coincide at x_0.

The continuity condition requires that

$$p = \mathrm{E}[Y|x_0] = \alpha + \beta x_0 + \gamma x_0^2$$

The smoothness condition requires that

$$\frac{\partial \mathrm{E}[Y|x_0]}{\partial x} = \beta + 2\gamma x_0 \equiv 0$$

If you solve for x_0 and substitute into the expression for c, the two conditions jointly imply that

$$x_0 = -\beta/2\gamma$$
$$c = \alpha - \beta^2/4\gamma$$

Although there are apparently four unknowns, the model contains only three parameters. The continuity and smoothness restrictions together completely determine one parameter given the other three.

The following DATA step creates the SAS data set for this example:

```
data a;
   input y x @@;
   datalines;
.46 1  .47  2 .57  3 .61  4 .62  5 .68  6 .69  7
.78 8  .70  9 .74 10 .77 11 .78 12 .74 13 .80 13
.80 15 .78 16
 ;
```

The following PROC NLIN statements fit this segmented model:

```
title 'Quadratic Model with Plateau';
proc nlin data=a;
   parms alpha=.45 beta=.05 gamma=-.0025;

   x0 = -.5*beta / gamma;

   if (x < x0) then
        mean = alpha + beta*x  + gamma*x*x;
   else mean = alpha + beta*x0 + gamma*x0*x0;
   model y = mean;

   if _obs_=1 and _iter_ =.  then do;
      plateau =alpha + beta*x0 + gamma*x0*x0;
      put /  x0= plateau=  ;
   end;
   output out=b predicted=yp;
run;
```

The parameters of the model are α, β, and γ, respectively. They are represented in the PROC NLIN statements by the variables alpha, beta, and gamma, respectively. In order to model the two segments, a conditional statement is used that assigns the appropriate expression to the mean function depending on the value of x_0. A PUT statement is used to print the constrained parameters every time the program is executed for the first observation. The OUTPUT statement computes predicted values for plotting and saves them to data set b.

Note that there are other ways in which you can write the conditional expressions for this model. For example, you could formulate a condition with two model statements, as follows:

```
proc nlin data=a;
   parms alpha=.45 beta=.05 gamma=-.0025;
   x0 = -.5*beta / gamma;
   if (x < x0) then
        model y = alpha+beta*x+gamma*x*x;
   else model y = alpha+beta*x0+gamma*x0*x0;
run;
```

Or you could use a single expression with a conditional evaluation, as in the following statements:

```
proc nlin data=a;
   parms alpha=.45 beta=.05 gamma=-.0025;
   x0 = -.5*beta / gamma;
   model y = (x <x0)*(alpha+beta*x +gamma*x*x) +
             (x>=x0)*(alpha+beta*x0+gamma*x0*x0);
run;
```

The results from fitting this model with PROC NLIN are shown in Output 60.1.1–Output 60.1.3. The iterative optimization converges after six iterations (Output 60.1.1). Output 60.1.1 indicates that the join point is 12.747 and the plateau value is 0.777.

Output 60.1.1 Nonlinear Least-Squares Iterative Phase

Quadratic Model with Plateau

The NLIN Procedure
Dependent Variable y
Method: Gauss-Newton

Iterative Phase

Iter	alpha	beta	gamma	Sum of Squares
0	0.4500	0.0500	−0.00250	0.0562
1	0.3881	0.0616	−0.00234	0.0118
2	0.3930	0.0601	−0.00234	0.0101
3	0.3922	0.0604	−0.00237	0.0101
4	0.3921	0.0605	−0.00237	0.0101
5	0.3921	0.0605	−0.00237	0.0101
6	0.3921	0.0605	−0.00237	0.0101

NOTE: Convergence criterion met.

Output 60.1.2 Results from Put Statement

```
x0=12.747669162 plateau=0.7774974276
```

Output 60.1.3 Least-Squares Analysis for the Quadratic Model

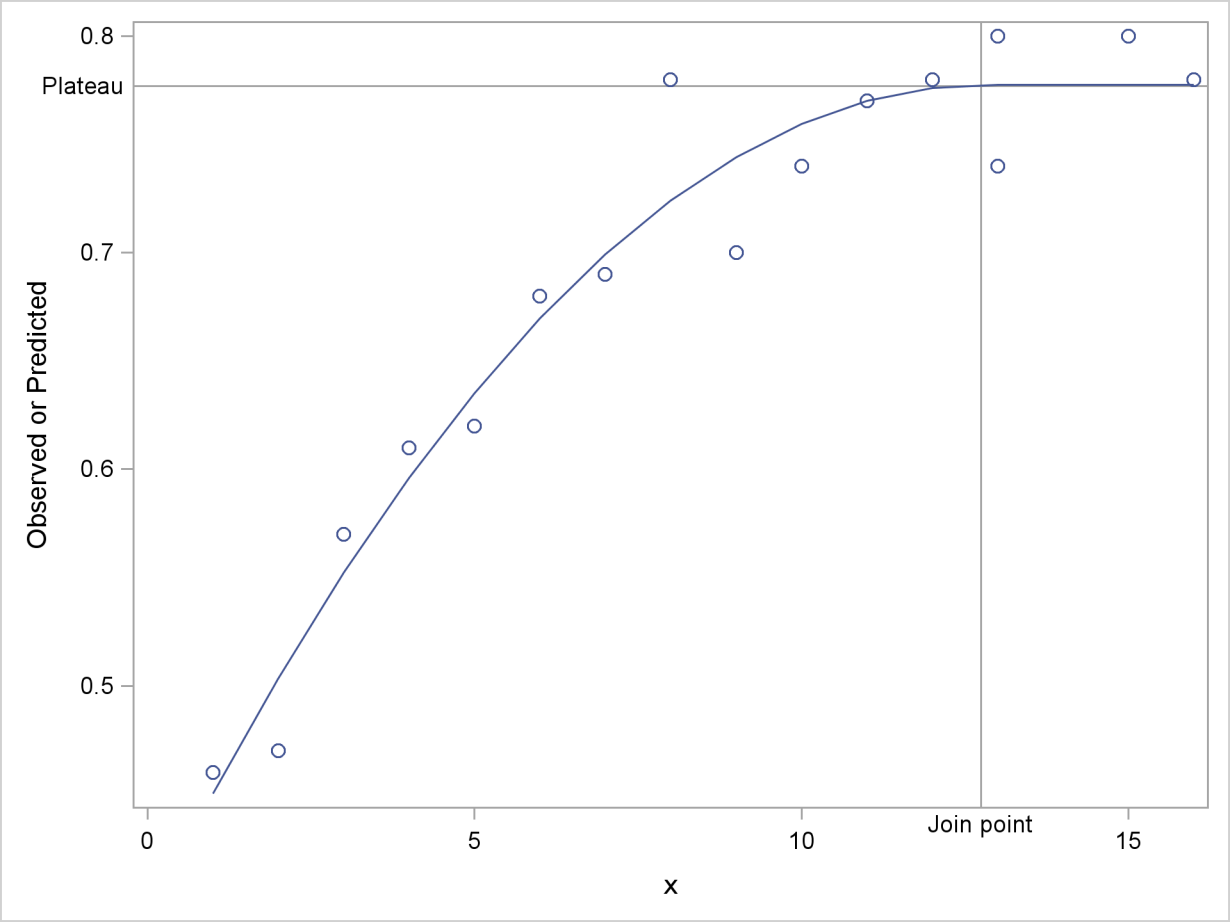
Source	DF	Sum of Squares	Mean Square	F Value	Approx Pr > F
Model	2	0.1769	0.0884	114.22	<.0001
Error	13	0.0101	0.000774		
Corrected Total	15	0.1869			

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits	
alpha	0.3921	0.0267	0.3345	0.4497
beta	0.0605	0.00842	0.0423	0.0787
gamma	−0.00237	0.000551	−0.00356	−0.00118

The following statements produce a graph of the observed and predicted values with reference lines for the join point and plateau estimates (Output 60.1.4):

```
proc sgplot data=b noautolegend;
   yaxis label='Observed or Predicted';
   refline 0.777  / axis=y label="Plateau"    labelpos=min;
   refline 12.747 / axis=x label="Join point" labelpos=min;
   scatter y=y  x=x;
   series  y=yp x=x;
run;
```

Output 60.1.4 Observed and Predicted Values for the Quadratic Model



If you want to estimate the join point directly, you can use the relationship between the parameters to change the parameterization of the model in such a way that the mean function depends directly on x_0. Using the smoothness condition that relates x_0 to γ,

$$x_0 = -\beta/2\gamma$$

you can express γ as a function of β and x_0:

$$\gamma = -\beta/(2x_0)$$

Substituting for γ in the model equation

$$\mathrm{E}[Y|x] = \begin{cases} \alpha + \beta x + \gamma x^2 & \text{if } x < x_0 \\ \alpha - \beta^2/(4\gamma) & \text{if } x \geq x_0 \end{cases}$$

yields the reparameterized model

$$\mathrm{E}[Y|x] = \begin{cases} \alpha + \beta x(1 - x/(2x_0)) & \text{if } x < x_0 \\ \alpha + \beta x_0/2 & \text{if } x \geq x_0 \end{cases}$$

This model is fit with the following PROC NLIN statements:

```
proc nlin data=a;
   parms alpha=.45 beta=.05 x0=10;
   if (x<x0) then
        mean = alpha + beta*x *(1-x/(2*x0));
   else mean = alpha + beta*x0/2;
   model y = mean;
run;
```

Output 60.1.5 Results from Reparameterized Model

```
                          The NLIN Procedure
                         Dependent Variable y
                         Method: Gauss-Newton

      NOTE: Convergence criterion met.

                                  Sum of        Mean                Approx
Source                    DF     Squares      Square    F Value     Pr > F

Model                      2      0.1769      0.0884     114.22     <.0001
Error                     13      0.0101    0.000774
Corrected Total           15      0.1869

                                    Approx      Approximate 95% Confidence
       Parameter      Estimate    Std Error                Limits

       alpha            0.3921       0.0267      0.3345      0.4497
       beta             0.0605      0.00842      0.0423      0.0787
       x0              12.7477       1.2781      9.9864     15.5089
```

The analysis of variance table in the reparameterized model is the same as in the earlier analysis (compare Output 60.1.5 and Output 60.1.3). Changing the parameterization of a model does not affect the fit. The "Parameter Estimates" table now shows x0 as a parameter in the model. The estimate agrees with the earlier result that uses the PUT statement (Output 60.1.2). Since x_0 is now a model parameter, the NLIN procedure also reports its asymptotic standard error and its approximate 95% confidence interval.

Example 60.2: Iteratively Reweighted Least Squares

With the NLIN procedure you can perform weighted nonlinear least-squares regression in situations where the weights are functions of the parameters. To minimize a weighted sum of squares, you assign an expression to the _WEIGHT_ variable in your PROC NLIN statements. When the _WEIGHT_ variable depends on the model parameters, the estimation technique is known as iteratively reweighted least squares (IRLS). In this situation you should employ the NOHALVE option in the PROC NLIN statement. Because the weights change from iteration to iteration, it is not reasonable to expect the *weighted* residual sum of squares to decrease between iterations. The NOHALVE option removes that restriction.

Examples where IRLS estimation is used include robust regression via M-estimation (Huber 1964, 1973), generalized linear models (McCullagh and Nelder 1989), and semivariogram fitting in spatial statistics (Schabenberger and Pierce 2002, Sect. 9.2). There are dedicated SAS/STAT procedures for robust regression (the ROBUSTREG procedure) and generalized linear models (the GENMOD and GLIMMIX procedures). Examples of weighted least-squares fitting of a semivariogram function can be found in Chapter 95, "The VARIOGRAM Procedure."

In this example we show an application of PROC NLIN for M-estimation only to illustrate the connection between robust regression and weighted least squares. The ROBUSTREG procedure is the appropriate tool to fit these models with SAS/STAT software.

M-estimation was introduced by Huber (1964, 1973) to estimate location parameters robustly. Beaton and Tukey (1974) applied the idea of M-estimation in regression models and introduced the biweight (or bisquare) weight function. See Holland and Welsch (1977) for this and other robust methods. Consider a linear regression model of the form

$$\mathrm{E}[Y_i|x] = \mathbf{x}_i'\boldsymbol{\beta} + \epsilon_i$$

In weighted least-squares estimation you seek the parameters $\widehat{\boldsymbol{\beta}}$ that minimize

$$\sum_{i=1}^{n} w_i \left(y_i - \mathbf{x}_i'\boldsymbol{\beta}\right)^2 = \sum_{i=1}^{n} w_i e_i^2$$

where w_i is the weight associated with the ith observation. The *normal* equations of this minimization problem can be written as

$$\sum_{i=1}^{n} w_i e_i \mathbf{x}_i = \mathbf{0}$$

In M-estimation the corresponding equations take on the form

$$\sum_{i=1}^{n} \psi(e_i)\mathbf{x}_i = \mathbf{0}$$

where $\psi(\cdot)$ is a weighing function. The Beaton-Tukey biweight, for example, can be written as

$$\psi(e_i) = \begin{cases} e_i\left(1 - \left(\frac{e_i}{\sigma k}\right)^2\right)^2 & |e_i/\sigma| \le k \\ 0 & |e_i/\sigma| > k \end{cases}$$

Substitution into the estimating equation for M-estimation yields weighted least-squares equations

$$\sum_{i=1}^{n} \psi(e_i)\mathbf{x}_i = \sum_{i=1}^{n} w_i e_i \mathbf{x}_i = \mathbf{0}$$

$$w_i = \begin{cases} \left(1 - \left(\frac{e_i}{\sigma k}\right)^2\right)^2 & |e_i/\sigma| \le k \\ 0 & |e_i/\sigma| > k \end{cases}$$

The biweight function involves two constants, σ and k. The scale σ can be fixed or estimated from the fit in the previous iteration. If σ is estimated, a robust estimator of scale is typically used. In this example σ is fixed at 2. A common value for the constant k is $k = 4.685$.

The following DATA step creates a SAS data set of the population of the United States (in millions), recorded at 10-year intervals starting in 1790 and ending in 1990. The aim is to fit a quadratic linear model to the population over time.

```
title 'U.S. Population Growth';
data uspop;
   input pop :6.3 @@;
   retain year 1780;
   year   = year+10;
   yearsq = year*year;
   datalines;
3929 5308 7239 9638 12866 17069 23191 31443 39818 50155
62947 75994 91972 105710 122775 131669 151325 179323 203211
226542 248710
;
```

The PROC NLIN code that follows fits this linear model by M-estimation and IRLS. The weight function is set to a zero or nonzero value depending on the value of the scaled residual. The NOHALVE option removes the requirement that the (weighted) residual sum of squares must decrease between iterations.

```
title 'Beaton/Tukey Biweight Robust Regression using IRLS';
proc nlin data=uspop nohalve;
   parms b0=20450.43 b1=-22.7806 b2=.0063456;
   model pop=b0+b1*year+b2*year*year;
   resid = pop-model.pop;
   sigma = 2;
   k     = 4.685;
   if abs(resid/sigma)<=k then _weight_=(1-(resid / (sigma*k))**2)**2;
   else _weight_=0;
   output out=c r=rbi;
run;
```

Parameter estimates from this fit are shown in Output 60.2.1, and the computed weights at the final iteration are displayed in Output 60.2.2. The observations for 1940 and 1950 are highly discounted because of their large residuals.

Output 60.2.1 Nonlinear Least-Squares Analysis

Beaton/Tukey Biweight Robust Regression using IRLS

The NLIN Procedure

Source	DF	Sum of Squares	Mean Square	F Value	Approx Pr > F
Model	2	113564	56782.0	49454.5	<.0001
Error	18	20.6670	1.1482		
Corrected Total	20	113585			

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits	
b0	20828.7	259.4	20283.8	21373.6
b1	-23.2004	0.2746	-23.7773	-22.6235
b2	0.00646	0.000073	0.00631	0.00661

Output 60.2.2 Listing of Computed Weights from PROC NLIN

Obs	pop	year	yearsq	rbi	sigma	k	_weight_
1	3.929	1790	3204100	-0.93711	2	4.685	0.98010
2	5.308	1800	3240000	0.46091	2	4.685	0.99517
3	7.239	1810	3276100	1.11853	2	4.685	0.97170
4	9.638	1820	3312400	0.95176	2	4.685	0.97947
5	12.866	1830	3348900	0.32159	2	4.685	0.99765
6	17.069	1840	3385600	-0.62597	2	4.685	0.99109
7	23.191	1850	3422500	-0.94692	2	4.685	0.97968
8	31.443	1860	3459600	-0.43027	2	4.685	0.99579
9	39.818	1870	3496900	-1.08302	2	4.685	0.97346
10	50.155	1880	3534400	-1.06615	2	4.685	0.97427
11	62.947	1890	3572100	0.11332	2	4.685	0.99971
12	75.994	1900	3610000	0.25539	2	4.685	0.99851
13	91.972	1910	3648100	2.03607	2	4.685	0.90779
14	105.710	1920	3686400	0.28436	2	4.685	0.99816
15	122.775	1930	3724900	0.56725	2	4.685	0.99268
16	131.669	1940	3763600	-8.61325	2	4.685	0.02403
17	151.325	1950	3802500	-8.32415	2	4.685	0.04443
18	179.323	1960	3841600	-0.98543	2	4.685	0.97800
19	203.211	1970	3880900	0.95088	2	4.685	0.97951
20	226.542	1980	3920400	1.03780	2	4.685	0.97562
21	248.710	1990	3960100	-1.33067	2	4.685	0.96007

You can obtain this analysis more conveniently with PROC ROBUSTREG. The procedure re-estimates the scale parameter robustly between iterations. To obtain an analysis with a fixed scale parameter as in this example, use the following PROC ROBUSTREG statements:

```
proc robustreg data=uspop method=m(scale=2);
   model pop = year year*year;
   output out=weights weight=w;
run;
```

```
proc print data=weights;
run;
```

Note that the computation of standard errors in the ROBUSTREG procedure is different from the calculations in the NLIN procedure.

Example 60.3: Probit Model with Likelihood Function

The data in this example, taken from Lee (1974), consist of patient characteristics and a variable indicating whether cancer remission occurred. This example demonstrates how to use PROC NLIN with a likelihood function. In this case, twice the negative of the log-likelihood function is to be minimized. This is the objective function for the analysis:

$$-2\log L = -2\sum_{i=1}^{n}\log\{\pi_i(y_i,\mathbf{x}_i)\}$$

In this expression, π_i denotes the success probability of the n Bernoulli trials, and $\log L$ is the log likelihood of n independent binary (Bernoulli) observations. The probability π_i depends on the observations through the linear predictor η_i,

$$\pi_i(y_i,\mathbf{x}_i) = \begin{cases} 1-\Phi(\eta_i) & y_i = 1 \\ \Phi(\eta_i) & y_i = 0 \end{cases}$$

The linear predictor takes the form of a regression equation that is linear in the parameters,

$$\eta_i = \beta_0 + \beta_1 z_{1i} + \beta_2 z_{2i} + \cdots \beta_k z_{ki} = \mathbf{z}_i\boldsymbol{\beta}$$

Despite this linearity of η in the z variables, the probit model is nonlinear, because the linear predictor appears inside the nonlinear probit function.

In order to use the NLIN procedure to minimize the function, the estimation problem must be cast in terms of a nonlinear least-squares problem with objective function

$$\sum_{i=1}^{n}(y_i - f(\boldsymbol{\beta},\mathbf{z}_i')^2$$

This can be accomplished by setting $y_i = 0$ and $f(\boldsymbol{\beta},\mathbf{z}_i') = \sqrt{(}-2\log\{\pi_i\})$. Because $0 \le \pi \le 1$, the function $-2\log\{\pi_i\}$ is strictly positive and the square root can be taken safely.

The following DATA step creates the data for the probit analysis. The variable like is created in the DATA step, and it contains the value 0 throughout. This variable serves as the "dummy" response variable in the PROC NLIN step. The variable remiss indicates whether cancer remission occurred. It is the binary outcome variable of interest and is used to determine the relevant probability for observation i as the success or failure probability of a Bernoulli experiment.

```
data remiss;
   input remiss cell smear infil li blast temp;
   label remiss = 'complete remission';
   like = 0;
   label like = 'dummy variable for nlin';
   datalines;
 1 0.8  .83 .66 1.9 1.10   .996
 1 0.9  .36 .32 1.4 0.74   .992
 0 0.8  .88 .70 0.8 0.176  .982
 0 1    .87 .87 0.7 1.053  .986
 1 0.9  .75 .68 1.3 0.519  .980
 0 1    .65 .65 0.6 0.519  .982
 1 0.95 .97 .92 1   1.23   .992
 0 0.95 .87 .83 1.9 1.354 1.020
 0 1    .45 .45 0.8 0.322  .999
 0 0.95 .36 .34 0.5 0     1.038
 0 0.85 .39 .33 0.7 0.279  .988
 0 0.7  .76 .53 1.2 0.146  .982
 0 0.8  .46 .37 0.4 0.38  1.006
 0 0.2  .39 .08 0.8 0.114  .990
 0 1    .90 .90 1.1 1.037  .990
 1 1    .84 .84 1.9 2.064 1.020
 0 0.65 .42 .27 0.5 0.114 1.014
 0 1    .75 .75 1   1.322 1.004
 0 0.5  .44 .22 0.6 0.114  .990
 1 1    .63 .63 1.1 1.072  .986
 0 1    .33 .33 0.4 0.176 1.010
 0 0.9  .93 .84 0.6 1.591 1.020
 1 1    .58 .58 1   0.531 1.002
 0 0.95 .32 .30 1.6 0.886  .988
 1 1    .60 .60 1.7 0.964  .990
 1 1    .69 .69 0.9 0.398  .986
 0 1    .73 .73 0.7 0.398  .986
 ;
```

The following NLIN statements fit the probit model:

```
proc nlin data=remiss method=newton sigsq=1;
   parms int=-10 a = -2 b = -1 c=6;

   linp = int + a*cell + b*li + c*temp;
   p   = probnorm(linp);

   if (remiss = 1) then pi = 1-p;
                   else pi = p;

   model.like = sqrt(- 2 * log(pi));
   output out=p p=predict;
run;
```

The assignment to the variable linp creates the linear predictor of the generalized linear model,

$$\eta = \beta_0 + \beta_1 \text{cell}_i + \beta_2 \text{li}_i + \beta_3 * \text{temp}_i$$

In this example, the variables cell, li, and temp are used as regressors.

By default, the NLIN procedure computes the covariance matrix of the parameter estimates based on the nonlinear least-squares assumption. That is, the procedure computes the estimate of the residual variance as the mean squared error and uses that to multiply the inverse crossproduct matrix or the inverse Hessian matrix. (See the section "Covariance Matrix of Parameter Estimates" on page 4307 for details.) In the probit model, there is no residual variance. In addition, standard errors in maximum likelihood estimation are based on the inverse Hessian matrix. The METHOD=NEWTON option in the PROC NLIN statement is used to employ the Hessian matrix in computing the covariance matrix of the parameter estimates. The SIGSQ=1 option replaces the residual variance estimate that PROC NLIN would use by default as a multiplier of the inverse Hessian with the value 1.0.

Output 60.3.1 shows the results of this analysis. The analysis of variance table shows an apparently strange result. The total sum of squares is zero, and the model sum of squares is negative. Recall that the values of the response variable were set to zero and the mean function was constructed as $-2\log\{\pi_i\}$ in order for the NLIN procedure to minimize a log-likelihood function in terms of a nonlinear least-squares problem. The value 21.9002 shown as the "Error" sum of squares is the value of the function $-2\log L$.

Output 60.3.1 Nonlinear Least-Squares Analysis from PROC NLIN

The NLIN Procedure

NOTE: An intercept was not specified for this model.

Source	DF	Sum of Squares	Mean Square	F Value	Approx Pr > F
Model	4	-21.9002	-5.4750	-5.75	.
Error	23	21.9002	0.9522		
Uncorrected Total	27	0			

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits	
int	-36.7548	32.3607	-103.7	30.1885
a	-5.6298	4.6376	-15.2235	3.9639
b	-2.2513	0.9790	-4.2764	-0.2262
c	45.1815	34.9095	-27.0343	117.4

The problem can be more simply solved using dedicated procedures for generalized linear models:

```
proc glimmix data=remiss;
   model remiss = cell li temp / dist=binary link=probit s;
run;

proc genmod data=remiss;
   model remiss = cell li temp / dist=bin link=probit;
run;
```

```
proc logistic data=remiss;
   model remiss = cell li temp / link=probit technique=newton;
run;
```

Example 60.4: Affecting Curvature through Parameterization

The work of Ratkowksy (1983, 1990) has brought into focus the importance of close-to-linear behavior of parameters in nonlinear regression models. The curvature in a nonlinear model consists of two components, the intrinsic curvature and the parameter-effects curvature. Intrinsic curvature expresses the degree to which the nonlinear model bends as values of the *parameters* change. This is not the same as the curviness of the model as a function of the covariates (the x variables). Intrinsic curvature is a function of the type of model you are fitting and the data. This curvature component cannot be affected by reparameterization of the model. According to Ratkowsky (1983), the intrinsic curvature component is typically smaller than the parameter-effects curvature, which can be affected by altering the parameterization of the model.

In models with low curvature, the nonlinear least-squares parameter estimators behave similarly to least-squares estimators in linear regression models, which have a number of desirable properties. If the model is correct, they are best linear unbiased estimators and are normally distributed if the model errors are normal (otherwise they are asymptotically normal). As you lower the curvature of a nonlinear model, you can expect that the parameter estimators approach the behavior of the linear regression model estimators: they behave "close to linear."

This example uses a simple data set and a commonly applied model for dose-response relationships to examine how the parameter-effects curvature can be reduced. The statistic by which an estimator's behavior is judged is Hougaard's measure of skewness (Hougaard 1982, 1985).

The log-logistic model

$$\mathrm{E}[Y|x] = \delta + \frac{\alpha - \delta}{1 + \gamma \exp\{\beta \ln(x)\}}$$

is a popular model to express the response Y as a function of dose x. The response is bounded between the asymptotes α and δ. The term in the denominator governs the transition between the asymptotes and depends on two parameters, γ and β. The log-logistic model can be viewed as a member of a broader class of dose-response functions, those relying on *switch-on* or *switch-off* mechanisms (see, for example, Schabenberger and Pierce 2002, Sect. 5.8.6). A switch function is usually a monotonic function $S(x, \boldsymbol{\theta})$ that takes values between 0 and 1. A switch-on function increases in x; a switch-off function decreases in x. In the log-logistic case, the function

$$S(x, [\beta, \gamma]) = \frac{1}{1 + \gamma \exp\{\beta \ln(x)\}}$$

is a switch-off function for $\beta > 0$ and a switch-on function for $\beta < 0$. You can write general dose-response functions with asymptotes simply as

$$\mathrm{E}[Y|x] = \mu_{\min} + (\mu_{\max} - \mu_{\min}) S(x, \boldsymbol{\theta})$$

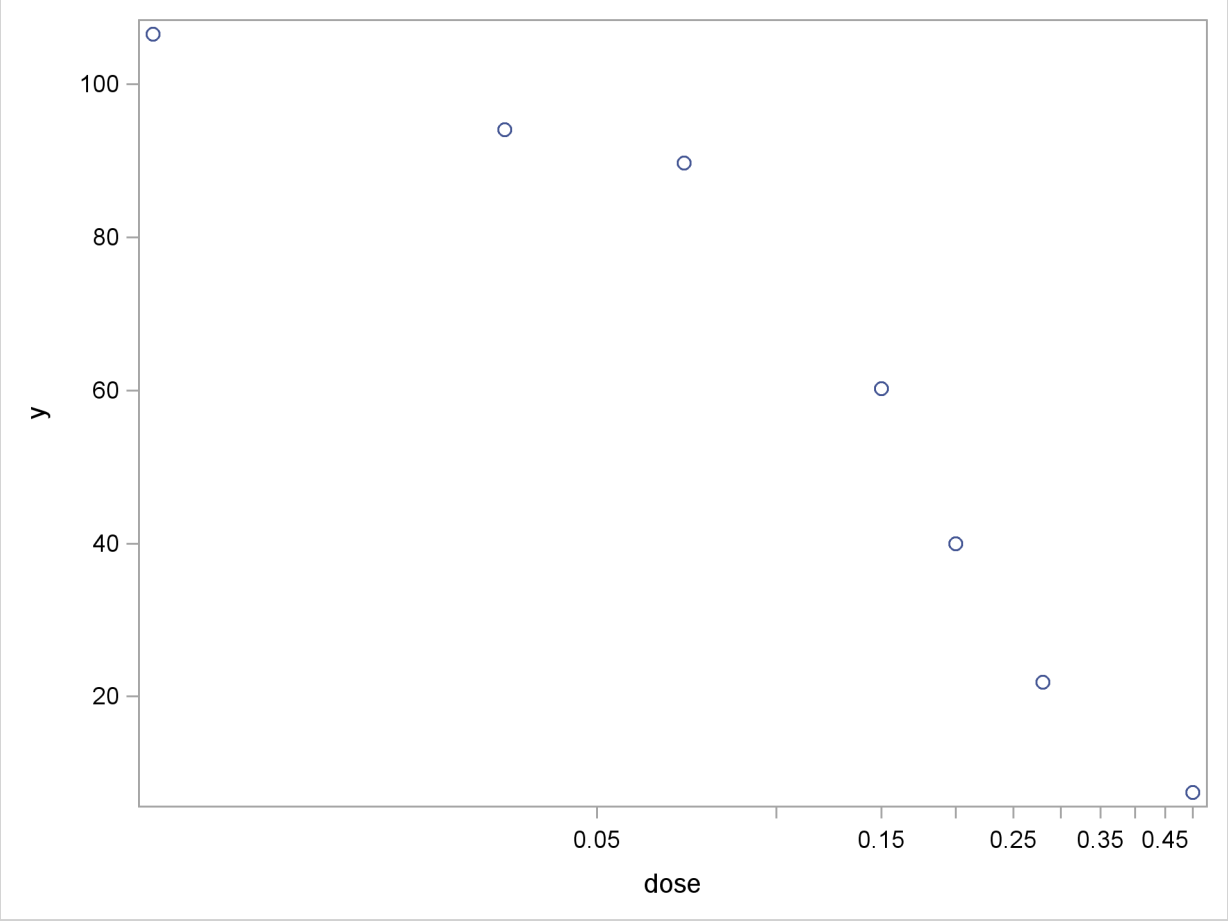
The following DATA step creates a small data set from a dose-response experiment with response y:

```
data logistic;
   input dose y;
   logdose = log(dose);
   datalines;
0.009  106.56
0.035   94.12
0.07    89.76
0.15    60.21
0.20    39.95
0.28    21.88
0.50     7.46
;
```

A graph of these data is produced with the following statements:

```
proc sgplot data=logistic;
   scatter y=y x=dose;
   xaxis type=log logstyle=linear;
run;
```

Output 60.4.1 Observed Data in Dose-Response Experiment



When dose is expressed on the log scale, the sigmoidal shape of the dose-response relationship is clearly visible (Output 60.4.1). The log-logistic switching model in the preceding parameterization is fit with the following statements in the NLIN procedure:

```
ods select IterHistory ConvergenceStatus Anova ParameterEstimates;
proc nlin data=logistic hougaard;
   parameters alpha=100 beta=3 gamma=300;
   delta = 0;
   Switch = 1/(1+gamma*exp(beta*log(dose)));
   model y = delta + (alpha - delta)*Switch;
run;
```

The lower asymptote δ was assumed to be 0 in this case. Since δ is not listed in the PARAMETERS statement and is assigned a value in the program, it is assumed to be constant. Note that the term Switch is the switch-off function in the log-logistic model. The HOUGAARD option in the PROC NLIN statement requests that Hougaard's skewness measure is added to the table of parameter estimates.

The NLIN procedure converges after 10 iteration and achieves a residual mean squared error of 15.1869 (Output 60.4.2). This value is not that important by itself, but it is worth noting since this model fit is compared to the fit with other parameterizations later on.

Output 60.4.2 Iteration History and Analysis of Variance

The NLIN Procedure
Dependent Variable y
Method: Gauss-Newton

Iterative Phase

Iter	alpha	beta	gamma	Sum of Squares
0	100.0	3.0000	300.0	386.4
1	100.4	2.8011	162.8	129.1
2	100.8	2.6184	101.4	69.2710
3	101.3	2.4266	69.7579	68.2167
4	101.7	2.3790	69.0358	60.8223
5	101.8	2.3621	67.3709	60.7516
6	101.8	2.3582	67.0044	60.7477
7	101.8	2.3573	66.9150	60.7475
8	101.8	2.3571	66.8948	60.7475
9	101.8	2.3570	66.8902	60.7475
10	101.8	2.3570	66.8892	60.7475

NOTE: Convergence criterion met.

NOTE: An intercept was not specified for this model.

Source	DF	Sum of Squares	Mean Square	F Value	Approx Pr > F
Model	3	33965.4	11321.8	745.50	<.0001
Error	4	60.7475	15.1869		
Uncorrected Total	7	34026.1			

The table of parameter estimates displays the estimates of the three model parameters, their approximate standard errors, 95% confidence limits, and Hougaard's skewness measure (Output 60.4.3). Parameters for which this measure is less than 0.1 in absolute value exhibit very close-to-linear behavior, and values less than 0.25 in absolute value indicate reasonably close-to-linear behavior (Ratkowsky 1990). According to these rules, the estimators $\widehat{\beta}$ and $\widehat{\gamma}$ suffer from substantial curvature. The estimator $\widehat{\gamma}$ is especially "far from linear." Inferences involving $\widehat{\gamma}$ that rely on the reported standard errors and/or confidence limits for this parameter might be questionable.

Output 60.4.3 Parameter Estimates and Hougaard's Skewness

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits		Skewness
alpha	101.8	3.0034	93.4751	110.2	0.1415
beta	2.3570	0.2928	1.5440	3.1699	0.4987
gamma	66.8892	31.6146	-20.8870	154.7	1.9200

One method of reducing the parameter-effects curvature is to replace a parameter with its expected-value parameterization. Schabenberger et al. (1999) and Schabenberger and Pierce (2002, Sect. 5.7.2) refer to this method as *reparameterization through defining relationships*. A defining relationship is obtained by equating the mean response at a chosen value of x (say, x^*) to the model:

$$\mathrm{E}[Y|x^*] = \delta + \frac{\alpha - \delta}{1 + \gamma \exp\{\beta \ln(x^*)\}}$$

This equation is then solved for a parameter that is subsequently replaced in the original equation. This method is particularly useful if x^* has an interesting interpretation. For example, let λ_K denote the value that reduces the response by $K \times 100\%$,

$$\mathrm{E}[Y|\lambda_K] = \delta + \left(\frac{100 - K}{100}\right)(\alpha - \delta)$$

Because γ exhibits large skewness, it is the target in the first round of reparameterization. Setting the expression for the conditional mean at λ_K equal to the mean function when $x = \lambda_K$ yields the following expression:

$$\delta + \left(\frac{100 - K}{100}\right)(\alpha - \delta) = \delta + \frac{\alpha - \delta}{1 + \gamma \exp\{\beta \ln(\lambda_K)\}}$$

This expression is solved for γ, and the result is substituted back into the model equation. This leads to a log-logistic model in which γ is replaced by the parameter λ_K, the dose at which the response was reduced by $K \times 100\%$. The new model equation is

$$\mathrm{E}[Y|x] = \delta + \frac{\alpha - \delta}{1 + K/(100 - K)\exp\{\beta \ln(x/\lambda_K)\}}$$

A particularly interesting choice is $K = 50$, since λ_{50} is the dose at which the response is halved. In studies of mortality, this concentration is also known as the LD50. For the special case of λ_{50} the model equation becomes

$$\mathrm{E}[Y|x] = \delta + \frac{\alpha - \delta}{1 + \exp\{\beta \ln(x/\lambda_{50})\}}$$

You can fit the model in the LD50 parameterization with the following statements:

```
ods select ConvergenceStatus Anova ParameterEstimates;
proc nlin data=logistic hougaard;
   parameters alpha=100 beta=3 LD50=0.15;
   delta = 0;
   Switch = 1/(1+exp(beta*log(dose/LD50)));
   model y = delta + (alpha - delta)*Switch;
   output out=nlinout pred=p lcl=lcl ucl=ucl;
run;
```

Partial results from this NLIN run are shown in Output 60.4.4. The analysis of variance tables in Output 60.4.2 and Output 60.4.4 are identical. Changing the parameterization of a model does not affect the model fit. It does, however, affect the interpretation of the parameters and the statistical properties (close-to-linear behavior) of the parameter estimators. The skewness measure of the parameter LD50 is considerably reduced compared to the skewness of the parameter γ in the previous parameterization. Also, γ has been replaced by a parameter with a useful interpretation, the dose that yields a 50% reduction in mean response. Also notice that the skewness measures of α and β are not affected by the $\gamma \rightarrow$ LD50 reparameterization.

Output 60.4.4 ANOVA Table and Parameter Estimates in LD50 Parameterization

The NLIN Procedure
Dependent Variable y
Method: Gauss-Newton

NOTE: Convergence criterion met.

NOTE: An intercept was not specified for this model.

Source	DF	Sum of Squares	Mean Square	F Value	Approx Pr > F
Model	3	33965.4	11321.8	745.50	<.0001
Error	4	60.7475	15.1869		
Uncorrected Total	7	34026.1			

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits		Skewness
alpha	101.8	3.0034	93.4752	110.2	0.1415
beta	2.3570	0.2928	1.5440	3.1699	0.4987
LD50	0.1681	0.00915	0.1427	0.1935	−0.0605

To reduce the parameter-effects curvature of the β parameter, you can use the technique of defining relationships again. This can be done generically, by solving

$$\mu^* = \delta + \frac{\alpha - \delta}{1 + \exp\{\beta \ln(x/\lambda)\}}$$

for β, treating μ^* as the new parameter (in lieu of β), and choosing a value for x^* that leads to low skewness. This results in the expected-value parameterization of β. Solving for β yields

$$\beta = \frac{\log\left(\frac{\alpha-\mu^*}{\mu^*-\delta}\right)}{\log\left(x^*/\lambda\right)}$$

The interpretation of the parameter μ^* that replaces β in the model equation is simple: it is the mean dose response when the dose is x^*. Fixing $x^* = 0.3$, the following PROC NLIN statements fit this model:

```
ods select ConvergenceStatus Anova ParameterEstimates;
proc nlin data=logistic hougaard;
   parameters alpha=100 mustar=20 LD50=0.15;
   delta   = 0;
   xstar   = 0.3;
   beta    = log((alpha - mustar)/(mustar - delta)) / log(xstar/LD50);
   Switch  = 1/(1+exp(beta*log(dose/LD50)));
   model y = delta + (alpha - delta)*Switch;
   output out=nlinout pred=p lcl=lcl ucl=ucl;
run;
```

Note that the switch-off function continues to be written in terms of β and the LD50. The only difference from the previous model is that β is now expressed as a function of the parameter μ^*. Using expected-value parameterizations is a simple mechanism to lower the curvature in a model and to arrive at starting values. The starting value for μ^* can be gleaned from Output 60.4.1 at $x = 0.3$.

Output 60.4.5 shows selected results from this NLIN run. The ANOVA table is again unaffected by the change in parameterization. The curvature for μ^* is greatly reduced in comparison to the curvature for the β parameter in the previous model (compare Output 60.4.5 and Output 60.4.4).

Output 60.4.5 ANOVA Table and Parameter Estimates in Expected-Value Parameterization

The NLIN Procedure
Dependent Variable y
Method: Gauss-Newton

NOTE: Convergence criterion met.

NOTE: An intercept was not specified for this model.

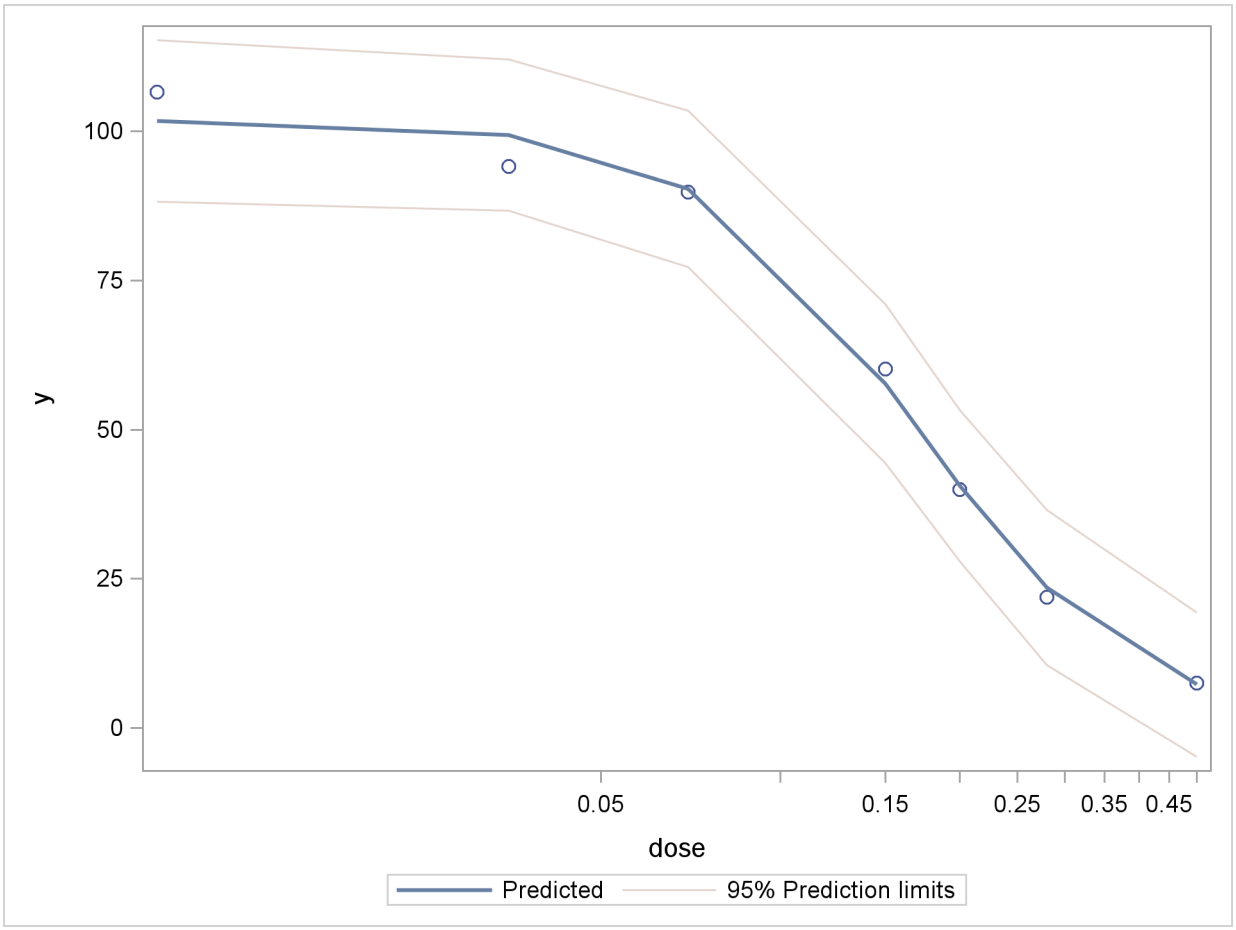
Source	DF	Sum of Squares	Mean Square	F Value	Approx Pr > F
Model	3	33965.4	11321.8	745.50	<.0001
Error	4	60.7475	15.1869		
Uncorrected Total	7	34026.1			

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits		Skewness
alpha	101.8	3.0034	93.4752	110.2	0.1415
mustar	20.7073	2.6430	13.3693	28.0454	−0.0572
LD50	0.1681	0.00915	0.1427	0.1935	−0.0605

The following statements produce a graph of the observed and fitted values along with the 95% prediction limits (Output 60.4.6). The graph is based on the data set created with the OUTPUT statement in the most recent PROC NLIN run.

```
proc sgplot data=nlinout;
   scatter y=y   x=dose;
   series  y=p   x=dose / name="fit"
                          lineattrs=GraphFit
                          LegendLabel="Predicted";
   series  y=lcl x=dose / lineattrs=GraphConfidence
                          name="lcl"
                          legendlabel="95% Prediction limits";
   series  y=ucl x=dose / lineattrs=GraphConfidence;
   keylegend "fit" "lcl";
   xaxis type=log logstyle=linear;
run;
```

Output 60.4.6 Fit Plot for Log-Logistic Model with 95% Prediction Limits



Example 60.5: Comparing Nonlinear Trends among Groups

When you model nonlinear trends in the presence of group (classification) variables, two questions often arise: whether the trends should be varied by group, and how to decide which parameters should be varied across groups. A large battery of tools is available on linear statistical models to test hypotheses involving the model parameters, especially to test linear hypotheses. To test similar hypotheses in nonlinear models, you can draw on analogous tools. Especially important in this regard are comparisons of nested models by contrasting their residual sums of squares.

In this example, a two-group model from a pharmacokinetic application is fit to data that are in part based on the theophylline data from Pinheiro and Bates (1995) and the first example in the documentation for the NLMIXED procedure. In a pharmacokinetic application you study how a drug is dispersed through a living organism. The following data represent concentrations of the drug theophylline over a 25-hour period following oral administration. The data are derived by collapsing and averaging the subject-specific data from Pinheiro and Bates (1995) in a particular, yet unimportant, way. The purpose of arranging the data in this way is purely to demonstrate the methodology.

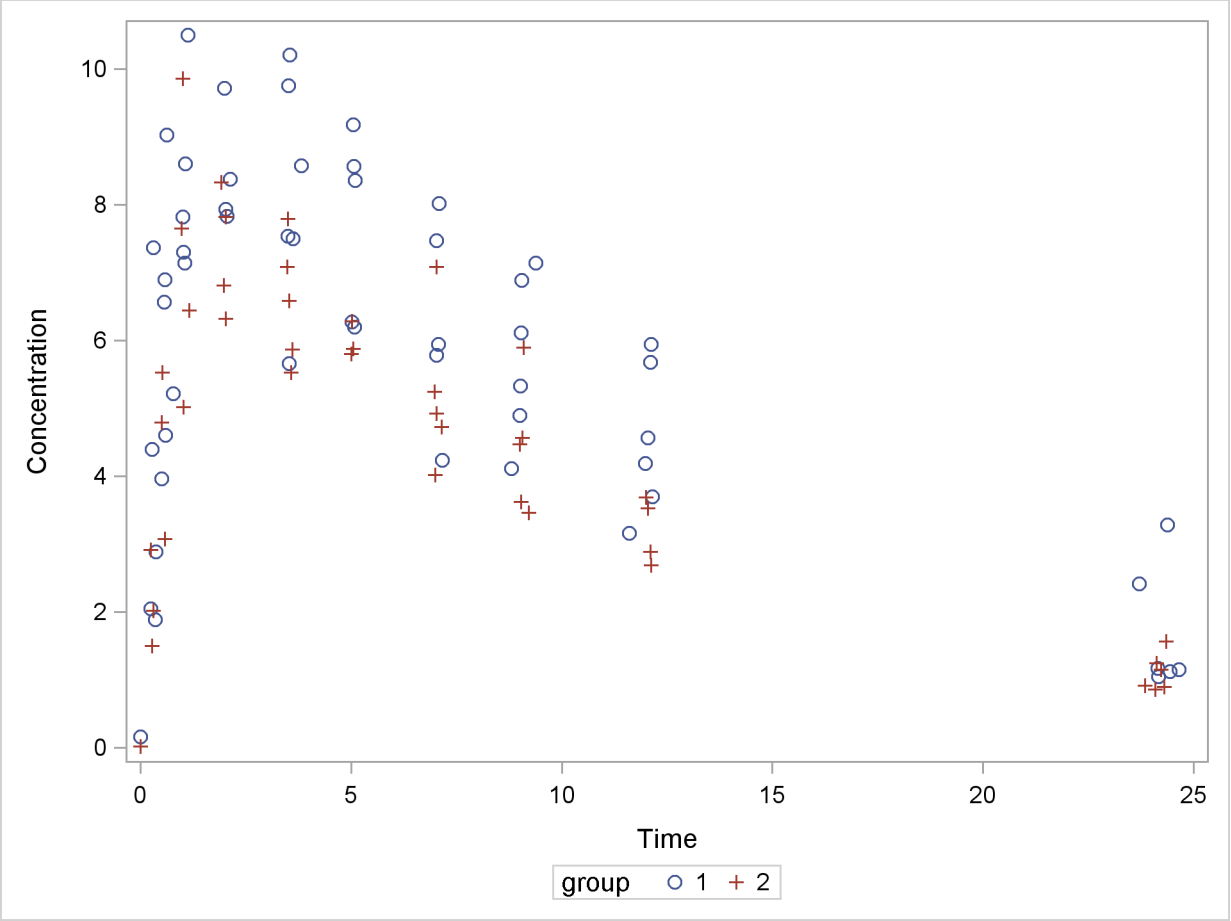
```
data theop;
  input time dose conc @@;
  if (dose = 4) then group=1; else group=2;
  datalines;
 0.00   4  0.1633  0.25   4   2.045
 0.27   4     4.4  0.30   4    7.37
 0.35   4    1.89  0.37   4    2.89
 0.50   4    3.96  0.57   4    6.57
 0.58   4     6.9  0.60   4     4.6
 0.63   4    9.03  0.77   4    5.22
 1.00   4    7.82  1.02   4   7.305
 1.05   4    7.14  1.07   4     8.6
 1.12   4    10.5  2.00   4    9.72
 2.02   4    7.93  2.05   4    7.83
 2.13   4    8.38  3.50   4    7.54
 3.52   4    9.75  3.53   4    5.66
 3.55   4   10.21  3.62   4     7.5
 3.82   4    8.58  5.02   4   6.275
 5.05   4    9.18  5.07   4    8.57
 5.08   4     6.2  5.10   4    8.36
 7.02   4    5.78  7.03   4    7.47
 7.07   4   5.945  7.08   4    8.02
 7.17   4    4.24  8.80   4    4.11
 9.00   4     4.9  9.02   4    5.33
 9.03   4    6.11  9.05   4    6.89
 9.38   4    7.14 11.60   4    3.16
11.98   4    4.19 12.05   4    4.57
12.10   4    5.68 12.12   4    5.94
12.15   4     3.7 23.70   4    2.42
24.15   4    1.17 24.17   4    1.05
24.37   4    3.28 24.43   4    1.12
24.65   4    1.15  0.00   5   0.025
```

```
 0.25   5    2.92  0.27   5   1.505
 0.30   5    2.02  0.50   5   4.795
 0.52   5    5.53  0.58   5    3.08
 0.98   5   7.655  1.00   5   9.855
 1.02   5    5.02  1.15   5    6.44
 1.92   5    8.33  1.98   5    6.81
 2.02   5  7.8233  2.03   5    6.32
 3.48   5    7.09  3.50   5   7.795
 3.53   5    6.59  3.57   5    5.53
 3.60   5    5.87  5.00   5     5.8
 5.02   5  6.2867  5.05   5    5.88
 6.98   5    5.25  7.00   5    4.02
 7.02   5    7.09  7.03   5   4.925
 7.15   5    4.73  9.00   5    4.47
 9.03   5    3.62  9.07   5    4.57
 9.10   5     5.9  9.22   5    3.46
12.00   5    3.69 12.05   5    3.53
12.10   5    2.89 12.12   5    2.69
23.85   5    0.92 24.08   5    0.86
24.12   5    1.25 24.22   5    1.15
24.30   5     0.9 24.35   5    1.57
;
```

The following code plots the theophylline concentration data over time for the two groups (Output 60.5.1). In each group the concentration tends to rise sharply right after the drug is administered, followed by a prolonged tapering of the concentration.

```
proc sgplot data=theop;
   scatter x=time y=conc / group=group;
   yaxis label='Concentration';
   xaxis label='Time';
run;
```

Output 60.5.1 Observed Responses in Two Groups



In the context of nonlinear mixed models, Pinheiro and Bates (1995) consider a first-order compartment model for these data. In terms of two fixed treatment groups, the model can be written as

$$C_{it} = \frac{Dk_{e_i}k_{a_i}}{Cl_i(k_{a_i} - k_{e_i})}[\exp(-k_{e_i}t) - \exp(-k_{a_i}t)] + \epsilon_{it}$$

where C_{it} is the observed concentration in group i at time t, D is the dose of theophylline, k_{e_i} is the elimination rate in group i, k_{a_i} is the absorption rate in group i, Cl_i is the clearance in group i, and ϵ_{it} denotes the model error. Because the rates and the clearance must be positive, you can parameterize the model in terms of log rates and the log clearance:

$$\begin{aligned} Cl_i &= \exp\{\beta_{1i}\} \\ k_{a_i} &= \exp\{\beta_{2i}\} \\ k_{e_i} &= \exp\{\beta_{3i}\} \end{aligned}$$

In this parameterization the model contains six parameters, and the rates and clearance vary by group. This produces two separate response profiles, one for each group. On the other extreme, you

could model the trends as if there were no differences among the groups:

$$\begin{aligned} Cl_i &= \exp\{\beta_1\} \\ k_{a_i} &= \exp\{\beta_2\} \\ k_{e_i} &= \exp\{\beta_3\} \end{aligned}$$

In between these two extremes lie other models, such as a model where both groups have the same absorption and elimination rate, but different clearances. The question then becomes how to go about building a model in an organized manner.

To test hypotheses about nested nonlinear models, you can apply the idea of a "Sum of Squares Reduction Test." A reduced model is nested within a full model if you can impose q constraints on the full model to obtain the reduced model. Then, if SSE_r and SSE_f denote the residual sum of squares in the reduced and the full model, respectively, the test statistic is

$$F_R = \frac{\left(\text{SSE}_r - \text{SSE}_f\right)/q}{\text{SSE}_f/(n-p)} = \frac{\left(\text{SSE}_r - \text{SSE}_f\right)/q}{\text{MSE}_f}$$

where n are the number of observations used and p are the number of parameters in the full model. The numerator of the F_R statistic is the average reduction in residual sum of squares per constraint. The mean squared error of the full model is used to scale this average because it is less likely to be a biased estimator of the residual variance than the variance estimate from a constrained (reduced) model. The F_R statistic is then compared against quantiles from an F distribution with q numerator and $n-p$ denominator degrees of freedom. Schabenberger and Pierce (2002) discuss the justification for this test and compare it to other tests in nonlinear models.

In the present application we might phrase the initial question akin to the overall F test for a factor in a linear model: Should any parameters be varied between the two groups? The corresponding null hypothesis is

$$H: \begin{cases} \beta_{11} = \beta_{12} \\ \beta_{21} = \beta_{22} \\ \beta_{31} = \beta_{32} \end{cases}$$

where the first subscript identifies the type of the parameter and the second subscript identifies the group. Note that this hypothesis implies

$$H: \begin{cases} Cl_1 = Cl_2 \\ k_{a_1} = k_{a_2} \\ k_{e_1} = k_{e_2} \end{cases}$$

If you fail to reject this hypothesis, there is no need to further examine individual parameter differences.

The reduced model—the model subject to the null hypothesis—is fit with the following PROC NLIN statements:

```
ods select ConvergenceStatus Anova ParameterEstimates;
proc nlin data=theop;
   parms beta1=-3.22 beta2=0.47 beta3=-2.45;
   cl   = exp(beta1);
   ka   = exp(beta2);
   ke   = exp(beta3);
   mean = dose*ke*ka*(exp(-ke*time)-exp(-ka*time))/cl/(ka-ke);
   model conc = mean;
   ods output Anova=aovred(rename=(ss=ssred ms=msred df=dfred));
run;
```

The clearance, the rates, and the mean function are formed independently of the group membership. The analysis of variance table is saved to the data set aovred, and some of its variables are renamed. This is done so that the data set can be merged easily with the analysis of variance table for the full model (see following).

The converged model has a residual sum of square of $\text{SSE}_r = 286.4$ and a mean squared error of 3.0142 (Output 60.5.2). The table of parameter estimates gives the values for the estimated log clearance ($\widehat{\beta}_1 = -3.2991$), the estimated log absorption rate ($\widehat{\beta}_2 = 0.4769$), and the estimated log elimination rate ($\widehat{\beta}_3 = -2.5555$).

Output 60.5.2 Fit Results for the Reduced Model

The NLIN Procedure
Dependent Variable conc
Method: Gauss-Newton

NOTE: Convergence criterion met.

NOTE: An intercept was not specified for this model.

Source	DF	Sum of Squares	Mean Square	F Value	Approx Pr > F
Model	3	3100.5	1033.5	342.87	<.0001
Error	95	286.4	3.0142		
Uncorrected Total	98	3386.8			

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits	
beta1	-3.2991	0.0956	-3.4888	-3.1094
beta2	0.4769	0.1640	0.1512	0.8025
beta3	-2.5555	0.1410	-2.8354	-2.2755

The full model, in which all three parameters are varied by group, can be fit with the following statements in the NLIN procedure:

```
ods select ConvergenceStatus Anova ParameterEstimates;
proc nlin data=theop;
   parms beta1_1=-3.22 beta2_1=0.47 beta3_1=-2.45
         beta1_2=-3.22 beta2_2=0.47 beta3_2=-2.45;
   if (group=1) then do;
      cl   = exp(beta1_1);
      ka   = exp(beta2_1);
      ke   = exp(beta3_1);
   end; else do;
      cl   = exp(beta1_2);
      ka   = exp(beta2_2);
      ke   = exp(beta3_2);
   end;
   mean = dose*ke*ka*(exp(-ke*time)-exp(-ka*time))/cl/(ka-ke);
   model conc = mean;
   ods output Anova=aovfull;
run;
```

Separate parameters for the groups are now specified in the PARMS statement, and the value of the model variables cl, ka, and ke is assigned conditional on the group membership of an observation. Notice that the same expression as in the previous run can be used to model the mean function.

The results from this PROC NLIN run are shown in Output 60.5.3. The residual sum of squares in the full model is only $\text{SSE}_f = 138.9$, compared to $\text{SSE}_r = 286.4$ in the reduced model (Output 60.5.3).

Output 60.5.3 Fit Results for the Full Model

```
                          The NLIN Procedure
                       Dependent Variable conc
                         Method: Gauss-Newton

     NOTE: Convergence criterion met.

           NOTE: An intercept was not specified for this model.

                                    Sum of        Mean                Approx
Source                      DF     Squares      Square    F Value    Pr > F

Model                        6      3247.9       541.3     358.56    <.0001
Error                       92       138.9      1.5097
Uncorrected Total           98      3386.8

                                       Approx      Approximate 95% Confidence
      Parameter      Estimate     Std Error                 Limits

      beta1_1         -3.5671        0.0864     -3.7387     -3.3956
      beta2_1          0.4421        0.1349      0.1742      0.7101
      beta3_1         -2.6230        0.1265     -2.8742     -2.3718
      beta1_2         -3.0111        0.1061     -3.2219     -2.8003
      beta2_2          0.3977        0.1987     0.00305      0.7924
      beta3_2         -2.4442        0.1618     -2.7655     -2.1229
```

Whether this reduction in sum of squares is sufficient to declare that the full model provides a significantly better fit than the reduced model depends on the number of constraints imposed on the full model and on the variability in the data. In other words, before drawing any conclusions, you have to take into account how many parameters have been dropped from the model and how much variation around the regression trends the data exhibit. The F_R statistic sets these quantities into relation. The following macro merges the analysis of variance tables from the full and reduced model, and computes F_R and its p-value:

```
%macro SSReductionTest;
   data aov; merge aovred aovfull;
     if (Source='Error') then do;
        Fstat  = ((SSred-SS)/(dfred-df))/ms;
        pvalue = 1-Probf(Fstat,dfred-df,df);
        output;
     end;
   run;
   proc print data=aov label noobs;
      label Fstat  = 'F Value'
            pValue = 'Prob > F';
      format pvalue pvalue8.;
      var Fstat pValue;
   run;
%mend;
%SSReductionTest;
```

Output 60.5.4 F Statistic and P-value for Hypothesis of Equal Trends

F Value	Prob > F
32.5589	<.000001

There is clear evidence that the model with separate trends fits these data significantly better (Output 60.5.4). To decide whether all parameters should be varied between the groups or only one or two of them, we first refit the model in a slightly different parameterization:

```
ods select ConvergenceStatus Anova ParameterEstimates;
proc nlin data=theop;
   parms beta1_1=-3.22 beta2_1=0.47 beta3_1=-2.45
         beta1_diff=0 beta2_diff=0 beta3_diff=0;
   if (group=1) then do;
      cl   = exp(beta1_1);
      ka   = exp(beta2_1);
      ke   = exp(beta3_1);
   end; else do;
      cl   = exp(beta1_1 + beta1_diff);
      ka   = exp(beta2_1 + beta2_diff);
      ke   = exp(beta3_1 + beta3_diff);
   end;
   mean = dose*ke*ka*(exp(-ke*time)-exp(-ka*time))/cl/(ka-ke);
   model conc = mean;
run;
```

In the preceding statements, the parameters in the second group were expressed using offsets from parameters in the first group. For example, the parameter beta1_diff measures the change in log clearance between group 2 and group 1.

This simple reparameterization does not affect the model fit. The analysis of variance tables in Output 60.5.5 and Output 60.5.3 are identical. It does, however, affect the interpretation of the estimated quantities. Since the parameter beta1_diff measures the change in the log clearance rates between the groups, you can use the approximate 95% confidence limits in Output 60.5.5 to assess whether that quantity in the pharmacokinetic equation varies between groups. Only the confidence interval for the difference in the log clearances excludes 0. The intervals for beta2_diff and beta3_diff include 0.

Output 60.5.5 Fit Results for the Full Model in Difference Parameterization

The NLIN Procedure
Dependent Variable conc
Method: Gauss-Newton

NOTE: Convergence criterion met.

NOTE: An intercept was not specified for this model.

Source	DF	Sum of Squares	Mean Square	F Value	Approx Pr > F
Model	6	3247.9	541.3	358.56	<.0001
Error	92	138.9	1.5097		
Uncorrected Total	98	3386.8			

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits	
beta1_1	-3.5671	0.0864	-3.7387	-3.3956
beta2_1	0.4421	0.1349	0.1742	0.7101
beta3_1	-2.6230	0.1265	-2.8742	-2.3718
beta1_diff	0.5560	0.1368	0.2842	0.8278
beta2_diff	-0.0444	0.2402	-0.5214	0.4326
beta3_diff	0.1788	0.2054	-0.2291	0.5866

This suggests as the final model one where the absorption and elimination rates are the same for both groups and only the clearances are varied. The following statements fit this model and perform the sum of squares reduction test:

```
ods select ConvergenceStatus Anova ParameterEstimates;
proc nlin data=theop;
   parms beta1_1=-3.22 beta2_1=0.47 beta3_1=-2.45
         beta1_diff=0;
   ka = exp(beta2_1);
   ke = exp(beta3_1);
   if (group=1) then do;
      cl = exp(beta1_1);
   end; else do;
      cl = exp(beta1_1 + beta1_diff);
   end;
   mean = dose*ke*ka*(exp(-ke*time)-exp(-ka*time))/cl/(ka-ke);
   model conc = mean;
   ods output Anova=aovred(rename=(ss=ssred ms=msred df=dfred));
   output out=predvals predicted=p;
run;
%SSReductionTest;
```

The results for this model with common absorption and elimination rates are shown in Output 60.5.6. The sum-of-squares reduction test comparing this model against the full model with six parameters shows—as expected—that the full model does not fit the data significantly better ($p = 0.6193$, Output 60.5.7).

Output 60.5.6 Fit Results for Model with Common Rates

The NLIN Procedure
Dependent Variable conc
Method: Gauss-Newton

NOTE: Convergence criterion met.

NOTE: An intercept was not specified for this model.

Source	DF	Sum of Squares	Mean Square	F Value	Approx Pr > F
Model	4	3246.5	811.6	543.60	<.0001
Error	94	140.3	1.4930		
Uncorrected Total	98	3386.8			

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits	
beta1_1	-3.5218	0.0681	-3.6570	-3.3867
beta2_1	0.4226	0.1107	0.2028	0.6424
beta3_1	-2.5571	0.0988	-2.7532	-2.3610
beta1_diff	0.4346	0.0454	0.3444	0.5248

Output 60.5.7 F Statistic and P-value for Hypothesis of Common Rates

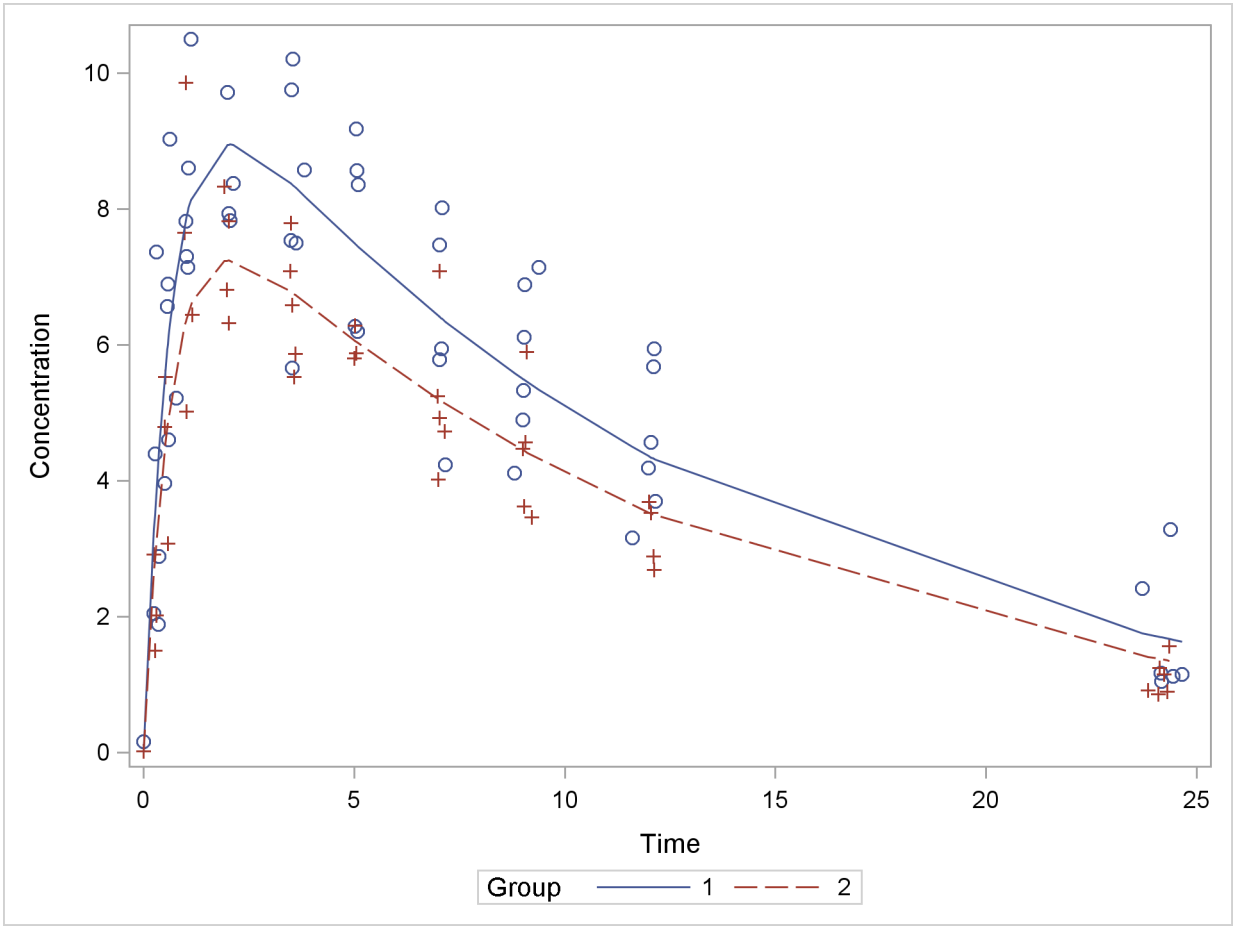
F Value	Prob > F
0.48151	0.619398

A plot of the observed and predicted values for this final model is obtained with the following statements:

```
proc sgplot data=predvals;
   scatter x=time y=conc / group=group;
   series  x=time y=p    / group=group name='fit';
   keylegend 'fit'       / across=2 title='Group';
   yaxis label='Concentration';
   xaxis label='Time';
run;
```

The plot is shown in Output 60.5.8.

Output 60.5.8 Observed and Fitted Values for Theophylline Data



The sum-of-squares reduction test is not the only possibility of performing linear-model style hy-

pothesis testing in nonlinear models. You can also perform Wald-type testing of linear hypotheses about the parameter estimates. See Example 38.17 in Chapter 38, "The GLIMMIX Procedure," for an application of this example that uses the NLIN and GLIMMIX procedures to compare the parameters across groups and adjusts *p*-values for multiplicity.

References

Bard, J. (1970), "Comparison of Gradient Methods for the Solution of the Nonlinear Parameter Estimation Problem," *SIAM Journal of Numerical Analysis*, 7, 157–186.

Bard, J. (1974), *Nonlinear Parameter Estimation*, New York: Academic Press.

Bates, D. M., and Watts, D. L. (1981), "A Relative Offset Orthogonality Convergence Criterion for Nonlinear Least Squares" *Technometrics*, 123, 179–183.

Beaton, A. E. and Tukey, J. W. (1974), "The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data," *Technometrics*, 16, 147–185.

Charnes, A., Frome, E. L., and Yu, P. L. (1976), "The Equivalence of Generalized Least Squares and Maximum Likelihood Estimation in the Exponential Family," *Journal of the American Statistical Association*, 71, 169–172.

Cox, D. R. (1970), *Analysis of Binary Data*, London: Chapman & Hall.

Finney, D. J. (1971), *Probit Analysis*, Third Edition, Cambridge: Cambridge University Press.

Gallant, A. R. (1975), "Nonlinear Regression," *American Statistician*, 29, 73–81.

Gill, P. E., Murray, W., and Wright, M. H. (1981), *Practical Optimization*, New York: Academic Press.

Goodnight, J. H. (1979), "A Tutorial on the Sweep Operator," *American Statistician*, 33, 149–158.

Hartley, H. O. (1961), "The Modified Gauss-Newton Method for the Fitting of Non-linear Regression Functions by Least Squares," *Technometrics*, 3, 269–280.

Holland, P. H. and Welsch, R. E. (1977), "Robust Regression Using Iteratively Reweighted Least-Squares," *Communications Statistics: Theory and Methods*, 6, 813–827.

Hougaard, P. (1982), "Parameterizations of Non-linear Models," *Journal of the Royal Statistical Society, Series B*, 244–252.

Hougaard, P. (1985), "The Appropriateness of the Asymptotic Distribution in a Nonlinear Regression Model in Relation to Curvature," *Journal of the Royal Statistical Society, Series B*, 103–114.

Huber, P. J. (1964), "Robust Estimation of a Location Parameter," *Annals of Mathematical Statistics*, 35, 73–101.

Huber, P. J. (1973), "Robust Regression: Asymptotics, Conjectures, and Monte Carlo," *Annals of Statistics*, 1, 799–821.

Jennrich, R. I. (1969), "Asymptotic Properties of Nonlinear Least Squares Estimators," *Annals of Mathematical Statistics*, 40, 633–643.

Jennrich, R. I. and Moore, R. H. (1975), "Maximum Likelihood Estimation by Means of Nonlinear Least Squares," *American Statistical Association*, *1975 Proceedings of the Statistical Computing Section*, 57–65.

Jennrich, R. I. and Sampson, P. F. (1968), "Application of Stepwise Regression to Non-linear Estimation," *Technometrics*, 10, 63–72.

Judge, G. G., Griffiths, W. E., Hill, R. C., and Lee, T.-C. (1980), *The Theory and Practice of Econometrics*, New York: John Wiley & Sons.

Kennedy, W. J. and Gentle, J. E. (1980), *Statistical Computing*, New York: Marcel Dekker.

Lee, E. T. (1974), "A Computer Program for Linear Logistic Regression Analysis," *Computer Programs in Biomedicine*, 80–92.

Marquardt, D. W. (1963), "An Algorithm for Least-Squares Estimation of Nonlinear Parameters," *Journal for the Society of Industrial and Applied Mathematics*, 11, 431–441.

McCullagh, P. and Nelder, J. A. (1989), *Generalized Linear Models*, Second Edition, New York: Chapman & Hall.

Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized Linear Models," *Journal of the Royal Statistical Society, Series A*, 135, 370–384.

Pinheiro, J. C. and Bates, D. M. (1995), "Approximations to the Log-Likelihood Function in the Nonlinear Mixed-Effects Model," *Journal of Computational and Graphical Statistics,* 4, 12–35.

Pringle, R. M. and Rayner, A. A. (1971), *Generalized Inverse Matrices with Applications to Statistics*, New York: Hafner Publishing Co.

Ratkowsky, D. (1983), *Nonlinear Regression Modeling*, New York and Basel: Marcel Dekker.

Ratkowsky, D. (1990), *Handbook of Nonlinear Regression Models*, New York and Basel: Marcel Dekker.

Schabenberger, O. and Pierce, F. J. (2002), *Contemporary Statistical Models for the Plant and Soil Sciences*, Boca Raton, FL: CRC Press.

Schabenberger, O., Tharp, B. E., Kells, J. J., and Penner, D. (1999), "Statistical Tests for Hormesis and Effective Dosages in Herbicide Dose Response," *Agronomy Journal*, 91, 713–721.

Seber, G. A. F. and Wild, C. J. (1989) *Nonlinear Regression*, New York: John Wiley & Sons.

Chapter 61

The NLMIXED Procedure

Contents

Overview: NLMIXED Procedure **4344**
Introduction 4344
Literature on Nonlinear Mixed Models 4345
PROC NLMIXED Compared with Other SAS Procedures and Macros . . . 4345
Getting Started: NLMIXED Procedure **4346**
Nonlinear Growth Curves with Gaussian Data 4346
Logistic-Normal Model with Binomial Data 4351
Syntax: NLMIXED Procedure **4354**
PROC NLMIXED Statement 4355
ARRAY Statement 4373
BOUNDS Statement 4373
BY Statement 4374
CONTRAST Statement 4374
ESTIMATE Statement 4375
ID Statement 4375
MODEL Statement 4376
PARMS Statement 4376
PREDICT Statement 4377
RANDOM Statement 4378
REPLICATE Statement 4379
Programming Statements 4379
Details: NLMIXED Procedure **4381**
Modeling Assumptions and Notation 4381
Integral Approximations 4382
Built-in Log-Likelihood Functions 4384
Optimization Algorithms 4387
Finite-Difference Approximations of Derivatives 4392
Hessian Scaling 4394
Active Set Methods 4395
Line-Search Methods 4397
Restricting the Step Length 4397
Computational Problems 4399
Covariance Matrix 4402
Prediction 4403

Computational Resources 4404
Displayed Output 4405
ODS Table Names 4408
Examples: NLMIXED Procedure **4409**
Example 61.1: One-Compartment Model with Pharmacokinetic Data 4409
Example 61.2: Probit-Normal Model with Binomial Data 4415
Example 61.3: Probit-Normal Model with Ordinal Data 4419
Example 61.4: Poisson-Normal Model with Count Data 4424
Example 61.5: Failure Time and Frailty Model 4427
References **4437**

Overview: NLMIXED Procedure

Introduction

The NLMIXED procedure fits nonlinear mixed models—that is, models in which both fixed and random effects enter nonlinearly. These models have a wide variety of applications, two of the most common being pharmacokinetics and overdispersed binomial data. PROC NLMIXED enables you to specify a conditional distribution for your data (given the random effects) having either a standard form (normal, binomial, Poisson) or a general distribution that you code using SAS programming statements.

PROC NLMIXED fits nonlinear mixed models by maximizing an approximation to the likelihood integrated over the random effects. Different integral approximations are available, the principal ones being adaptive Gaussian quadrature and a first-order Taylor series approximation. A variety of alternative optimization techniques are available to carry out the maximization; the default is a dual quasi-Newton algorithm.

Successful convergence of the optimization problem results in parameter estimates along with their approximate standard errors based on the second derivative matrix of the likelihood function. PROC NLMIXED enables you to use the estimated model to construct predictions of arbitrary functions by using empirical Bayes estimates of the random effects. You can also estimate arbitrary functions of the nonrandom parameters, and PROC NLMIXED computes their approximate standard errors by using the delta method.

Literature on Nonlinear Mixed Models

Davidian and Giltinan (1995) and Vonesh and Chinchilli (1997) provide good overviews as well as general theoretical developments and examples of nonlinear mixed models. Pinheiro and Bates (1995) is a primary reference for the theory and computational techniques of PROC NLMIXED. They describe and compare several different integrated likelihood approximations and provide evidence that adaptive Gaussian quadrature is one of the best methods. Davidian and Gallant (1993) also use Gaussian quadrature for nonlinear mixed models, although the smooth nonparametric density they advocate for the random effects is currently not available in PROC NLMIXED.

Traditional approaches to fitting nonlinear mixed models involve Taylor series expansions, expanding around either zero or the empirical best linear unbiased predictions of the random effects. The former is the basis for the well-known first-order method of Beal and Sheiner (1982, 1988) and Sheiner and Beal (1985), and it is optionally available in PROC NLMIXED. The latter is the basis for the estimation method of Lindstrom and Bates (1990), and it is not available in PROC NLMIXED. However, the closely related Laplacian approximation is an option; it is equivalent to adaptive Gaussian quadrature with only one quadrature point. The Laplacian approximation and its relationship to the Lindstrom-Bates method are discussed by Beal and Sheiner (1992), Wolfinger (1993), Vonesh (1992, 1996), Vonesh and Chinchilli (1997), and Wolfinger and Lin (1997).

A parallel literature exists in the area of generalized linear mixed models, in which random effects appear as a part of the linear predictor inside a link function. Taylor-series methods similar to those just described are discussed in articles such as Harville and Mee (1984), Stiratelli, Laird, and Ware (1984), Gilmour, Anderson, and Rae (1985), Goldstein (1991), Schall (1991), Engel and Keen (1992), Breslow and Clayton (1993), Wolfinger and O'Connell (1993), and McGilchrist (1994), but such methods have not been implemented in PROC NLMIXED because they can produce biased results in certain binary data situations (Rodriguez and Goldman 1995, Lin and Breslow 1996). Instead, a numerical quadrature approach is available in PROC NLMIXED, as discussed in Pierce and Sands (1975), Anderson and Aitkin (1985), Crouch and Spiegelman (1990), Hedeker and Gibbons (1994), Longford (1994), McCulloch (1994), Liu and Pierce (1994), and Diggle, Liang, and Zeger (1994).

Nonlinear mixed models have important applications in pharmacokinetics, and Roe (1997) provides a wide-ranging comparison of many popular techniques. Yuh et al. (1994) provide an extensive bibliography on nonlinear mixed models and their use in pharmacokinetics.

PROC NLMIXED Compared with Other SAS Procedures and Macros

The models fit by PROC NLMIXED can be viewed as generalizations of the random coefficient models fit by the MIXED procedure. This generalization allows the random coefficients to enter the model nonlinearly, whereas in PROC MIXED they enter linearly. With PROC MIXED you can perform both maximum likelihood and restricted maximum likelihood (REML) estimation, whereas PROC NLMIXED implements only maximum likelihood. This is because the analog to the REML method in PROC NLMIXED would involve a high-dimensional integral over all of the fixed-effects parameters, and this integral is typically not available in closed form. Finally, PROC

MIXED assumes the data to be normally distributed, whereas PROC NLMIXED enables you to analyze data that are normal, binomial, or Poisson or that have any likelihood programmable with SAS statements.

PROC NLMIXED does not implement the same estimation techniques available with the NLINMIX macro or the default estimation method of the GLIMMIX procedure. These are based on the estimation methods of Lindstrom and Bates (1990), Breslow and Clayton (1993), and Wolfinger and O'Connell (1993), and they iteratively fit a set of generalized estimating equations (see Chapters 14 and 15 of Littell et al. 2006 and Wolfinger 1997). In contrast, PROC NLMIXED directly maximizes an approximate integrated likelihood. This remark also applies to the SAS/IML macros MIXNLIN (Vonesh and Chinchilli 1997) and NLMEM (Galecki 1998).

The GLIMMIX procedure also fits mixed models for nonnormal data with nonlinearity in the conditional mean function. In contrast to the NLMIXED procedure, PROC GLIMMIX assumes that the model contains a linear predictor that links covariates to the conditional mean of the response. The NLMIXED procedure is designed to handle general conditional mean functions, whether they contain a linear component or not. As mentioned earlier, the GLIMMIX procedure by default estimates parameters in generalized linear mixed models by pseudo-likelihood techniques, whereas PROC NLMIXED by default performs maximum likelihood estimation by adaptive Gauss-Hermite quadrature. This estimation method is also available with the GLIMMIX procedure (METHOD=QUAD in the PROC GLIMMIX statement).

PROC NLMIXED has close ties with the NLP procedure in SAS/OR software. PROC NLMIXED uses a subset of the optimization code underlying PROC NLP and has many of the same optimization-based options. Also, the programming statement functionality used by PROC NLMIXED is the same as that used by PROC NLP and the MODEL procedure in SAS/ETS software.

Getting Started: NLMIXED Procedure

Nonlinear Growth Curves with Gaussian Data

As an introductory example, consider the orange tree data of Draper and Smith (1981). These data consist of seven measurements of the trunk circumference (in millimeters) on each of five orange trees. You can input these data into a SAS data set as follows:

```
data tree;
  input tree day y;
  datalines;
1  118   30
1  484   58
1  664   87
1 1004  115
1 1231  120
1 1372  142
```

```
1 1582  145
2  118   33
2  484   69
2  664  111
2 1004  156
2 1231  172
2 1372  203
2 1582  203
3  118   30
3  484   51
3  664   75
3 1004  108
3 1231  115
3 1372  139
3 1582  140
4  118   32
4  484   62
4  664  112
4 1004  167
4 1231  179
4 1372  209
4 1582  214
5  118   30
5  484   49
5  664   81
5 1004  125
5 1231  142
5 1372  174
5 1582  177
;
```

Lindstrom and Bates (1990) and Pinheiro and Bates (1995) propose the following logistic nonlinear mixed model for these data:

$$y_{ij} = \frac{b_1 + u_{i1}}{1 + \exp[-(d_{ij} - b_2)/b_3]} + e_{ij}$$

Here, y_{ij} represents the jth measurement on the ith tree ($i = 1, \ldots, 5$; $j = 1, \ldots, 7$), d_{ij} is the corresponding day, b_1, b_2, b_3 are the fixed-effects parameters, u_{i1} are the random-effect parameters assumed to be iid $N(0, \sigma_u^2)$, and e_{ij} are the residual errors assumed to be iid $N(0, \sigma_e^2)$ and independent of the u_{i1}. This model has a logistic form, and the random-effect parameters u_{i1} enter the model linearly.

The statements to fit this nonlinear mixed model are as follows:

```
proc nlmixed data=tree;
   parms b1=190 b2=700 b3=350 s2u=1000 s2e=60;
   num = b1+u1;
   ex  = exp(-(day-b2)/b3);
   den = 1 + ex;
   model y ~ normal(num/den,s2e);
   random u1 ~ normal(0,s2u) subject=tree;
run;
```

The PROC NLMIXED statement invokes the procedure and inputs the tree data set. The PARMS statement identifies the unknown parameters and their starting values. Here there are three fixed-effects parameters (b1, b2, b3) and two variance components (s2u, s2e).

The next three statements are SAS programming statements specifying the logistic mixed model. A new variable u1 is included to identify the random effect. These statements are evaluated for every observation in the data set when the NLMIXED procedure computes the log likelihood function and its derivatives.

The MODEL statement defines the dependent variable and its conditional distribution given the random effects. Here a normal (Gaussian) conditional distribution is specified with mean num/den and variance s2e.

The RANDOM statement defines the single random effect to be u1, and specifies that it follow a normal distribution with mean 0 and variance s2u. The SUBJECT= argument in the RANDOM statement defines a variable indicating when the random effect obtains new realizations; in this case, it changes according to the values of the tree variable. PROC NLMIXED assumes that the input data set is clustered according to the levels of the tree variable; that is, all observations from the same tree occur sequentially in the input data set.

The output from this analysis is as follows.

Figure 61.1 Model Specifications

The NLMIXED Procedure

Specifications

Data Set	WORK.TREE
Dependent Variable	y
Distribution for Dependent Variable	Normal
Random Effects	u1
Distribution for Random Effects	Normal
Subject Variable	tree
Optimization Technique	Dual Quasi-Newton
Integration Method	Adaptive Gaussian Quadrature

The "Specifications" table lists basic information about the nonlinear mixed model you have specified (Figure 61.1). Included are the input data set, the dependent and subject variables, the random effects, the relevant distributions, and the type of optimization. The "Dimensions" table lists various counts related to the model, including the number of observations, subjects, and parameters (Figure 61.2). These quantities are useful for checking that you have specified your data set and model correctly. Also listed is the number of quadrature points that PROC NLMIXED has selected based on the evaluation of the log likelihood at the starting values of the parameters. Here, only one quadrature point is necessary because the random-effect parameters u_{i1} enter the model linearly. (The Gauss-Hermite quadrature with a single quadrature point results in the Laplace approximation of the log likelihood.)

Figure 61.2 Dimensions Table for Growth Curve Model

Dimensions

Observations Used	35
Observations Not Used	0
Total Observations	35
Subjects	5
Max Obs Per Subject	7
Parameters	5
Quadrature Points	1

Figure 61.3 Starting Values of Parameter Estimates and Negative Log Likelihood

Parameters

b1	b2	b3	s2u	s2e	NegLogLike
190	700	350	1000	60	132.491787

The "Parameters" table lists the parameters to be estimated, their starting values, and the negative log likelihood evaluated at the starting values (Figure 61.3).

Figure 61.4 Iteration History for Growth Curve Model

Iteration History

Iter	Calls	NegLogLike	Diff	MaxGrad	Slope
1	4	131.686742	0.805045	0.010269	-0.633
2	6	131.64466	0.042082	0.014783	-0.0182
3	8	131.614077	0.030583	0.009809	-0.02796
4	10	131.572522	0.041555	0.001186	-0.01344
5	11	131.571895	0.000627	0.0002	-0.00121
6	13	131.571889	5.549E-6	0.000092	-7.68E-6
7	15	131.571888	1.096E-6	6.097E-6	-1.29E-6

NOTE: GCONV convergence criterion satisfied.

The "Iteration History" table records the history of the minimization of the negative log likelihood (Figure 61.4). For each iteration of the quasi-Newton optimization, values are listed for the number of function calls, the value of the negative log likelihood, the difference from the previous iteration, the absolute value of the largest gradient, and the slope of the search direction. The note at the bottom of the table indicates that the algorithm has converged successfully according to the GCONV convergence criterion, a standard criterion computed using a quadratic form in the gradient and the inverse Hessian.

The final maximized value of the log likelihood as well as the information criterion of Akaike (AIC), its small sample bias corrected version (AICC), and the Bayesian information criterion (BIC) in the

"smaller is better" form appear in the "Fit Statistics" table (Figure 61.5). These statistics can be used to compare different nonlinear mixed models.

Figure 61.5 Fit Statistics for Growth Curve Model

Fit Statistics

-2 Log Likelihood	263.1
AIC (smaller is better)	273.1
AICC (smaller is better)	275.2
BIC (smaller is better)	271.2

Figure 61.6 Parameter Estimates at Convergence

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower
b1	192.05	15.6473	4	12.27	0.0003	0.05	148.61
b2	727.90	35.2472	4	20.65	<.0001	0.05	630.04
b3	348.07	27.0790	4	12.85	0.0002	0.05	272.88
s2u	999.88	647.44	4	1.54	0.1974	0.05	-797.70
s2e	61.5139	15.8831	4	3.87	0.0179	0.05	17.4153

Parameter Estimates

Parameter	Upper	Gradient
b1	235.50	1.154E-6
b2	825.76	5.289E-6
b3	423.25	-6.1E-6
s2u	2797.45	-3.84E-6
s2e	105.61	2.892E-6

The maximum likelihood estimates of the five parameters and their approximate standard errors computed using the final Hessian matrix are displayed in the "Parameter Estimates" table (Figure 61.6). Approximate t-values and Wald-type confidence limits are also provided, with degrees of freedom equal to the number of subjects minus the number of random effects. You should interpret these statistics cautiously for variance parameters like s2u and s2e. The final column in the output shows the gradient vector at the optimization solution. Each element appears to be sufficiently small to indicate a stationary point.

Since the random-effect parameters u_{i1} enter the model linearly, you can obtain equivalent results by using the first-order method (specify METHOD=FIRO in the PROC NLMIXED statement).

Logistic-Normal Model with Binomial Data

This example analyzes the data from Beitler and Landis (1985), which represent results from a multi-center clinical trial investigating the effectiveness of two topical cream treatments (active drug, control) in curing an infection. For each of eight clinics, the number of trials and favorable cures are recorded for each treatment. The SAS data set is as follows.

```
data infection;
  input clinic t x n;
  datalines;
1 1 11 36
1 0 10 37
2 1 16 20
2 0 22 32
3 1 14 19
3 0  7 19
4 1  2 16
4 0  1 17
5 1  6 17
5 0  0 12
6 1  1 11
6 0  0 10
7 1  1  5
7 0  1  9
8 1  4  6
8 0  6  7
;
```

Suppose n_{ij} denotes the number of trials for the ith clinic and the jth treatment ($i = 1, \ldots, 8$; $j = 0, 1$), and x_{ij} denotes the corresponding number of favorable cures. Then a reasonable model for the preceding data is the following logistic model with random effects:

$$x_{ij}|u_i \sim \text{Binomial}(n_{ij}, p_{ij})$$

and

$$\eta_{ij} = \log\left(\frac{p_{ij}}{1-p_{ij}}\right) = \beta_0 + \beta_1 t_j + u_i$$

The notation t_j indicates the jth treatment, and the u_i are assumed to be iid $N(0, \sigma_u^2)$.

The PROC NLMIXED statements to fit this model are as follows:

```
proc nlmixed data=infection;
   parms beta0=-1 beta1=1 s2u=2;
   eta    = beta0 + beta1*t + u;
   expeta = exp(eta);
   p      = expeta/(1+expeta);
   model x ~ binomial(n,p);
   random u ~ normal(0,s2u) subject=clinic;
   predict eta out=eta;
   estimate '1/beta1' 1/beta1;
run;
```

The PROC NLMIXED statement invokes the procedure, and the PARMS statement defines the parameters and their starting values. The next three statements define p_{ij}, and the MODEL statement defines the conditional distribution of x_{ij} to be binomial. The RANDOM statement defines u to be the random effect with subjects defined by the clinic variable.

The PREDICT statement constructs predictions for each observation in the input data set. For this example, predictions of η_{ij} and approximate standard errors of prediction are output to a data set named eta. These predictions include empirical Bayes estimates of the random effects u_i.

The ESTIMATE statement requests an estimate of the reciprocal of β_1.

The output for this model is as follows.

Figure 61.7 Model Information and Dimensions for Logistic-Normal Model

The NLMIXED Procedure

Specifications

Data Set	WORK.INFECTION
Dependent Variable	x
Distribution for Dependent Variable	Binomial
Random Effects	u
Distribution for Random Effects	Normal
Subject Variable	clinic
Optimization Technique	Dual Quasi-Newton
Integration Method	Adaptive Gaussian Quadrature

Dimensions

Observations Used	16
Observations Not Used	0
Total Observations	16
Subjects	8
Max Obs Per Subject	2
Parameters	3
Quadrature Points	5

The "Specifications" table provides basic information about the nonlinear mixed model (Figure 61.7). For example, the distribution of the response variable, conditional on normally distributed random effects, is binomial. The "Dimensions" table provides counts of various variables. You should check this table to make sure the data set and model have been entered properly. PROC NLMIXED selects five quadrature points to achieve the default accuracy in the likelihood calculations.

Figure 61.8 Starting Values of Parameter Estimates

Parameters

beta0	beta1	s2u	NegLogLike
-1	1	2	37.5945925

The "Parameters" table lists the starting point of the optimization and the negative log likelihood at the starting values (Figure 61.8).

Figure 61.9 Iteration History and Fit Statistics for Logistic-Normal Model

Iteration History

Iter	Calls	NegLogLike	Diff	MaxGrad	Slope
1	2	37.3622692	0.232323	2.882077	-19.3762
2	3	37.1460375	0.216232	0.921926	-0.82852
3	5	37.0300936	0.115944	0.315897	-0.59175
4	6	37.0223017	0.007792	0.01906	-0.01615
5	7	37.0222472	0.000054	0.001743	-0.00011
6	9	37.0222466	6.57E-7	0.000091	-1.28E-6
7	11	37.0222466	5.38E-10	2.078E-6	-1.1E-9

NOTE: GCONV convergence criterion satisfied.

Fit Statistics

-2 Log Likelihood	74.0
AIC (smaller is better)	80.0
AICC (smaller is better)	82.0
BIC (smaller is better)	80.3

The "Iteration History" table indicates successful convergence in seven iterations (Figure 61.9). The "Fit Statistics" table lists some useful statistics based on the maximized value of the log likelihood.

Figure 61.10 Parameter Estimates for Logistic-Normal Model

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower
beta0	-1.1974	0.5561	7	-2.15	0.0683	0.05	-2.5123
beta1	0.7385	0.3004	7	2.46	0.0436	0.05	0.02806
s2u	1.9591	1.1903	7	1.65	0.1438	0.05	-0.8554

Parameter Estimates

Parameter	Upper	Gradient
beta0	0.1175	-3.1E-7
beta1	1.4488	-2.08E-6
s2u	4.7736	-2.48E-7

The "Parameter Estimates" table indicates marginal significance of the two fixed-effects parameters (Figure 61.10). The positive value of the estimate of β_1 indicates that the treatment significantly increases the chance of a favorable cure.

Figure 61.11 Table of Additional Estimates

Additional Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
1/beta1	1.3542	0.5509	7	2.46	0.0436	0.05	0.05146	2.6569

The "Additional Estimates" table displays results from the ESTIMATE statement (Figure 61.11). The estimate of $1/\beta_1$ equals $1/0.7385 = 1.3542$ and its standard error equals $0.3004/0.7385^2 = 0.5509$ by the delta method (Billingsley 1986, Cox 1998). Note that this particular approximation produces a t-statistic identical to that for the estimate of β_1.

Not shown is the eta data set, which contains the original 16 observations and predictions of the η_{ij}.

Syntax: NLMIXED Procedure

The following statements can be used with the NLMIXED procedure:

PROC NLMIXED < *options* > ;
  ARRAY *array specification* ;
  BOUNDS *boundary constraints* ;
  BY *variables* ;
  CONTRAST *'label' expression* < *,expression* >< *options* > ;
  ESTIMATE *'label' expression* < *options* > ;
  ID *names* ;
  MODEL *model specification* ;
  PARMS *parameters and starting values* ;
  PREDICT *expression* **OUT=***SAS-data-set* < *options* > ;
  RANDOM *random effects specification* ;
  REPLICATE *variable* ;
  Program statements ;

The following sections provide a detailed description of each of these statements.

PROC NLMIXED Statement

PROC NLMIXED < *options* > **;**

This statement invokes the NLMIXED procedure. A large number of options are available in the PROC NLMIXED statement, and Table 61.1 categorizes them according to function.

Table 61.1 PROC NLMIXED Statement Options

Option	Description
Basic Options	
DATA=	input data set
METHOD=	integration method
Displayed Output Specifications	
START	gradient at starting values
HESS	Hessian matrix
ITDETAILS	iteration details
CORR	correlation matrix
COV	covariance matrix
ECORR	correlation matrix of additional estimates
ECOV	covariance matrix of additional estimates
EDER	derivatives of additional estimates
EMPIRICAL	empirical (“sandwich”) estimator of covariance matrix
ALPHA=	alpha for confidence limits
DF=	degrees of freedom for p-values and confidence limits
Debugging Output	
LIST	model program, variables
LISTCODE	compiled model program
LISTDEP	model dependency listing
LISTDER	model derivatives
XREF	model cross references
FLOW	model execution messages
TRACE	detailed model execution messages
Quadrature Options	
NOAD	no adaptive centering
NOADSCALE	no adaptive scaling
OUTQ=	output data set
QFAC=	search factor
QMAX=	maximum points
QPOINTS=	number of points
QSCALEFAC=	scale factor
QTOL=	tolerance

Table 61.1 *continued*

Option	Description
Empirical Bayes Options	
EBSTEPS=	number of Newton steps
EBSUBSTEPS=	number of substeps
EBSSFRAC=	step-shortening fraction
EBSSTOL=	step-shortening tolerance
EBTOL=	convergence tolerance
EBOPT	comprehensive optimization
EBZSTART	zero starting values
Optimization Specifications	
TECHNIQUE=	minimization technique
UPDATE=	update technique
LINESEARCH=	line-search method
LSPRECISION=	line-search precision
HESCAL=	type of Hessian scaling
INHESSIAN<=>	start for approximated Hessian
RESTART=	iteration number for update restart
OPTCHECK<=>	check optimality in neighborhood
Derivatives Specifications	
FD<=>	finite-difference derivatives
FDHESSIAN<=>	finite-difference second derivatives
DIAHES	use only diagonal of Hessian
Constraint Specifications	
LCEPSILON=	range for active constraints
LCDEACT=	LM tolerance for deactivating
LCSINGULAR=	tolerance for dependent constraints
Termination Criteria Specifications	
MAXFUNC=	maximum number of function calls
MAXITER=	maximum number of iterations
MINITER=	minimum number of iterations
MAXTIME=	upper limit seconds of CPU time
ABSCONV=	absolute function convergence criterion
ABSFCONV=	absolute function convergence criterion
ABSGCONV=	absolute gradient convergence criterion
ABSXCONV=	absolute parameter convergence criterion
FCONV=	relative function convergence criterion
FCONV2=	relative function convergence criterion
GCONV=	relative gradient convergence criterion
XCONV=	relative parameter convergence criterion
FDIGITS=	number accurate digits in objective function
FSIZE=	used in FCONV, GCONV criterion

Table 61.1 *continued*

Option	Description
XSIZE=	used in XCONV criterion
Step Length Specifications	
DAMPSTEP<=>	damped steps in line search
MAXSTEP=	maximum trust-region radius
INSTEP=	initial trust-region radius
Singularity Tolerances	
SINGCHOL=	tolerance for Cholesky roots
SINGHESS=	tolerance for Hessian
SINGSWEEP=	tolerance for sweep
SINGVAR=	tolerance for variances
Covariance Matrix Tolerances	
ASINGULAR=	absolute singularity for inertia
MSINGULAR=	relative M singularity for inertia
VSINGULAR=	relative V singularity for inertia
G4=	threshold for Moore-Penrose inverse
COVSING=	tolerance for singular COV matrix
CFACTOR=	multiplication factor for COV matrix

These options are described in alphabetical order. For a description of the mathematical notation used in the following sections, see the section "Modeling Assumptions and Notation" on page 4381.

ABSCONV=*r*

ABSTOL=*r*

specifies an absolute function convergence criterion. For minimization, termination requires $f(\boldsymbol{\theta}^{(k)}) \leq r$. The default value of r is the negative square root of the largest double-precision value, which serves only as a protection against overflows.

ABSFCONV=*r*<**[***n***]**>

ABSFTOL=*r*<**[***n***]**>

specifies an absolute function convergence criterion. For all techniques except NMSIMP, termination requires a small change of the function value in successive iterations:

$$|f(\boldsymbol{\theta}^{(k-1)}) - f(\boldsymbol{\theta}^{(k)})| \leq r$$

The same formula is used for the NMSIMP technique, but $\boldsymbol{\theta}^{(k)}$ is defined as the vertex with the lowest function value, and $\boldsymbol{\theta}^{(k-1)}$ is defined as the vertex with the highest function value in the simplex. The default value is $r = 0$. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

ABSGCONV=*r*< [*n*] >

ABSGTOL=*r*< [*n*] >

specifies an absolute gradient convergence criterion. Termination requires the maximum absolute gradient element to be small:

$$\max_j |g_j(\boldsymbol{\theta}^{(k)})| \leq r$$

This criterion is not used by the NMSIMP technique. The default value is $r = 1\text{E}{-5}$. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

ABSXCONV=*r*< [*n*] >

ABSXTOL=*r*< [*n*] >

specifies an absolute parameter convergence criterion. For all techniques except NMSIMP, termination requires a small Euclidean distance between successive parameter vectors,

$$\| \boldsymbol{\theta}^{(k)} - \boldsymbol{\theta}^{(k-1)} \|_2 \leq r$$

For the NMSIMP technique, termination requires either a small length $\alpha^{(k)}$ of the vertices of a restart simplex,

$$\alpha^{(k)} \leq r$$

or a small simplex size,

$$\delta^{(k)} \leq r$$

where the simplex size $\delta^{(k)}$ is defined as the L1 distance from the simplex vertex $\boldsymbol{\xi}^{(k)}$ with the smallest function value to the other n simplex points $\boldsymbol{\theta}_l^{(k)} \neq \boldsymbol{\xi}^{(k)}$:

$$\delta^{(k)} = \sum_{\boldsymbol{\theta}_l \neq y} \| \boldsymbol{\theta}_l^{(k)} - \boldsymbol{\xi}^{(k)} \|_1$$

The default is $r = 1\text{E}{-8}$ for the NMSIMP technique and $r = 0$ otherwise. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

ALPHA=α

specifies the alpha level to be used in computing confidence limits. The default value is 0.05.

ASINGULAR=*r*

ASING=*r*

specifies an absolute singularity criterion for the computation of the inertia (number of positive, negative, and zero eigenvalues) of the Hessian and its projected forms. The default value is the square root of the smallest positive double-precision value.

CFACTOR=*f*

specifies a multiplication factor f for the estimated covariance matrix of the parameter estimates.

COV

requests the approximate covariance matrix for the parameter estimates.

CORR

requests the approximate correlation matrix for the parameter estimates.

COVSING=*r* > 0

specifies a nonnegative threshold that determines whether the eigenvalues of a singular Hessian matrix are considered to be zero.

DAMPSTEP<=*r***>**

DS<=*r***>**

specifies that the initial step-size value $\alpha^{(0)}$ for each line search (used by the QUANEW, CONGRA, or NEWRAP technique) cannot be larger than r times the step-size value used in the former iteration. If you specify the DAMPSTEP option without factor r, the default value is $r = 2$. The DAMPSTEP=r option can prevent the line-search algorithm from repeatedly stepping into regions where some objective functions are difficult to compute or where they could lead to floating-point overflows during the computation of objective functions and their derivatives. The DAMPSTEP=r option can save time-costly function calls that result in very small step sizes α. For more details on setting the start values of each line search, see the section "Restricting the Step Length" on page 4397.

DATA=*SAS-data-set*

specifies the input data set. Observations in this data set are used to compute the log likelihood function that you specify with PROC NLMIXED statements.

NOTE: If you are using a RANDOM statement, the input data set must be clustered according to the SUBJECT= variable. One easy way to accomplish this is to sort your data by the SUBJECT= variable prior to calling PROC NLMIXED. PROC NLMIXED does not sort the input data set for you.

DF=*d*

specifies the degrees of freedom to be used in computing p values and confidence limits. The default value is the number of subjects minus the number of random effects for random effects models, and the number of observations otherwise.

DIAHES

specifies that only the diagonal of the Hessian is used.

EBOPT

requests that a more comprehensive optimization be carried out if the default empirical Bayes optimization fails to converge.

EBSSFRAC=*r* > 0

specifies the step-shortening fraction to be used while computing empirical Bayes estimates of the random effects. The default value is 0.8.

EBSSTOL=*r* ≥ 0

specifies the objective function tolerance for determining the cessation of step-shortening while computing empirical Bayes estimates of the random effects. The default value is $r =$ 1E−8.

EBSTEPS=*n* ≥ 0
specifies the maximum number of Newton steps for computing empirical Bayes estimates of random effects. The default value is $n = 50$.

EBSUBSTEPS=*n* ≥ 0
specifies the maximum number of step-shortenings for computing empirical Bayes estimates of random effects. The default value is $n = 20$.

EBTOL=*r* ≥ 0
specifies the convergence tolerance for empirical Bayes estimation. The default value is $r = \epsilon$E4, where ϵ is the machine precision. This default value equals approximately 1E−12 on most machines.

EBZSTART
requests that a zero be used as starting values during empirical Bayes estimation. By default, the starting values are set equal to the estimates from the previous iteration (or zero for the first iteration).

ECOV
requests the approximate covariance matrix for all expressions specified in ESTIMATE statements.

ECORR
requests the approximate correlation matrix for all expressions specified in ESTIMATE statements.

EDER
requests the derivatives of all expressions specified in ESTIMATE statements with respect to each of the model parameters.

EMPIRICAL
requests that the covariance matrix of the parameter estimates be computed as a likelihood-based empirical ("sandwich") estimator (White 1982). If $f(\boldsymbol{\theta}) = -\log\{m(\boldsymbol{\theta})\}$ is the objective function for the optimization and $m(\boldsymbol{\theta})$ denotes the marginal log likelihood (see the section "Modeling Assumptions and Notation" on page 4381 for notation and further definitions) the empirical estimator is computed as

$$\mathbf{H}(\hat{\boldsymbol{\theta}})^{-1}\left(\sum_{i=1}^{s}\mathbf{g}_i(\hat{\boldsymbol{\theta}})\mathbf{g}_i(\hat{\boldsymbol{\theta}})'\right)\mathbf{H}(\hat{\boldsymbol{\theta}})^{-1}$$

where $\mathbf{H}$ is the second derivative matrix of f and $\mathbf{g}_i$ is the first derivative of the contribution to f by the ith subject. If you choose the EMPIRICAL option, this estimator of the covariance matrix of the parameter estimates replaces the model-based estimator $\mathbf{H}(\hat{\boldsymbol{\theta}})^{-1}$ in subsequent calculations. You can output the subject-specific gradients $\mathbf{g}_i$ to a SAS data set with the SUBGRADIENT option in the PROC NLMIXED statement.

The EMPIRICAL option requires the presence of a RANDOM statement and is available for METHOD=GAUSS and METHOD=ISAMP only.

FCONV=*r*< [*n*] >

FTOL=*r*< [*n*] >

specifies a relative function convergence criterion. For all techniques except NMSIMP, termination requires a small relative change of the function value in successive iterations,

$$\frac{|f(\boldsymbol{\theta}^{(k)}) - f(\boldsymbol{\theta}^{(k-1)})|}{\max(|f(\boldsymbol{\theta}^{(k-1)})|, \text{FSIZE})} \leq r$$

where FSIZE is defined by the FSIZE= option. The same formula is used for the NMSIMP technique, but $\boldsymbol{\theta}^{(k)}$ is defined as the vertex with the lowest function value, and $\boldsymbol{\theta}^{(k-1)}$ is defined as the vertex with the highest function value in the simplex. The default is $r = 10^{-\text{FDIGITS}}$, where FDIGITS is the value of the FDIGITS= option. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

FCONV2=*r*< [*n*] >

FTOL2=*r*< [*n*] >

specifies another function convergence criterion. For all techniques except NMSIMP, termination requires a small predicted reduction

$$df^{(k)} \approx f(\boldsymbol{\theta}^{(k)}) - f(\boldsymbol{\theta}^{(k)} + \mathbf{s}^{(k)})$$

of the objective function. The predicted reduction

$$\begin{aligned} df^{(k)} &= -\mathbf{g}^{(k)\prime}\mathbf{s}^{(k)} - \frac{1}{2}\mathbf{s}^{(k)\prime}\mathbf{H}^{(k)}\mathbf{s}^{(k)} \\ &= -\frac{1}{2}\mathbf{s}^{(k)\prime}\mathbf{g}^{(k)} \\ &\leq r \end{aligned}$$

is computed by approximating the objective function f by the first two terms of the Taylor series and substituting the Newton step:

$$\mathbf{s}^{(k)} = -[\mathbf{H}^{(k)}]^{-1}\mathbf{g}^{(k)}$$

For the NMSIMP technique, termination requires a small standard deviation of the function values of the $n+1$ simplex vertices $\boldsymbol{\theta}_l^{(k)}$, $l = 0, \ldots, n$,

$$\sqrt{\frac{1}{n+1}\sum_l \left[f(\boldsymbol{\theta}_l^{(k)}) - \overline{f}(\boldsymbol{\theta}^{(k)})\right]^2} \leq r$$

where $\overline{f}(\boldsymbol{\theta}^{(k)}) = \frac{1}{n+1}\sum_l f(\boldsymbol{\theta}_l^{(k)})$. If there are n_{act} boundary constraints active at $\boldsymbol{\theta}^{(k)}$, the mean and standard deviation are computed only for the $n + 1 - n_{act}$ unconstrained vertices. The default value is $r = 1\text{E}{-6}$ for the NMSIMP technique and $r = 0$ otherwise. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

FD < = FORWARD | CENTRAL |*r* >

specifies that all derivatives be computed using finite difference approximations. The following specifications are permitted:

FD is equivalent to FD=100.

FD=CENTRAL uses central differences.

FD=FORWARD uses forward differences.

FD=*r* uses central differences for the initial and final evaluations of the gradient and for the Hessian. During iteration, start with forward differences and switch to a corresponding central-difference formula during the iteration process when one of the following two criteria is satisfied:

- The absolute maximum gradient element is less than or equal to r times the ABSGCONV= threshold.
- The normalized predicted function reduction (see the GTOL option) is less than or equal to $\max(1E-6,\ r \times \text{GTOL})$. The 1E−6 ensures that the switch is done, even if you set the GTOL threshold to zero.

Note that the FD and FDHESSIAN options cannot apply at the same time. The FDHESSIAN option is ignored when only first-order derivatives are used. See the section "Finite-Difference Approximations of Derivatives" on page 4392 for more information.

FDHESSIAN< =**FORWARD | CENTRAL** >

FDHES< =**FORWARD | CENTRAL** >

FDH< =**FORWARD | CENTRAL** >

specifies that second-order derivatives be computed using finite difference approximations based on evaluations of the gradients.

FDHESSIAN=FORWARD uses forward differences.

FDHESSIAN=CENTRAL uses central differences.

FDHESSIAN uses forward differences for the Hessian except for the initial and final output.

Note that the FD and FDHESSIAN options cannot apply at the same time. See the section "Finite-Difference Approximations of Derivatives" on page 4392 for more information.

FDIGITS=*r*

specifies the number of accurate digits in evaluations of the objective function. Fractional values such as FDIGITS=4.7 are allowed. The default value is $r = -\log_{10}\epsilon$, where ϵ is the machine precision. The value of r is used to compute the interval size h for the computation of finite-difference approximations of the derivatives of the objective function and for the default value of the FCONV= option.

FLOW

displays a message for each statement in the model program as it is executed. This debugging option is very rarely needed and produces voluminous output.

FSIZE=*r*

specifies the FSIZE parameter of the relative function and relative gradient termination criteria. The default value is $r = 0$. For more details, see the FCONV= and GCONV= options.

G4=*n* > 0

specifies a dimension to determine the type of generalized inverse to use when the approximate covariance matrix of the parameter estimates is singular. The default value of n is 60. See the section "Covariance Matrix" on page 4402 for more information.

GCONV=*r* **<** [*n*] **>**

GTOL=*r* **<** [*n*] **>**

specifies a relative gradient convergence criterion. For all techniques except CONGRA and NMSIMP, termination requires that the normalized predicted function reduction is small,

$$\frac{\mathbf{g}(\boldsymbol{\theta}^{(k)})'[\mathbf{H}^{(k)}]^{-1}\mathbf{g}(\boldsymbol{\theta}^{(k)})}{\max(|f(\boldsymbol{\theta}^{(k)})|, \text{FSIZE})} \leq r$$

where FSIZE is defined by the FSIZE= option. For the CONGRA technique (where a reliable Hessian estimate H is not available), the following criterion is used:

$$\frac{\| \mathbf{g}(\boldsymbol{\theta}^{(k)}) \|_2^2 \quad \| \mathbf{s}(\boldsymbol{\theta}^{(k)}) \|_2}{\| \mathbf{g}(\boldsymbol{\theta}^{(k)}) - \mathbf{g}(\boldsymbol{\theta}^{(k-1)}) \|_2 \max(|f(\boldsymbol{\theta}^{(k)})|, \text{FSIZE})} \leq r$$

This criterion is not used by the NMSIMP technique.

The default value is $r = 1\text{E}{-8}$. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can terminate.

HESCAL=0|1|2|3

HS=0|1|2|3

specifies the scaling version of the Hessian matrix used in NRRIDG, TRUREG, NEWRAP, or DBLDOG optimization.

If HS is not equal to 0, the first iteration and each restart iteration sets the diagonal scaling matrix $\mathbf{D}^{(0)} = \text{diag}(d_i^{(0)})$:

$$d_i^{(0)} = \sqrt{\max(|H_{i,i}^{(0)}|, \epsilon)}$$

where $H_{i,i}^{(0)}$ are the diagonal elements of the Hessian. In every other iteration, the diagonal scaling matrix $\mathbf{D}^{(0)} = \text{diag}(d_i^{(0)})$ is updated depending on the HS option:

HS=0 specifies that no scaling is done.

HS=1 specifies the Moré (1978) scaling update:

$$d_i^{(k+1)} = \max\left[d_i^{(k)}, \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)}\right]$$

HS=2 specifies the Dennis, Gay, and Welsch (1981) scaling update:

$$d_i^{(k+1)} = \max\left[0.6 * d_i^{(k)}, \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)}\right]$$

HS=3 specifies that d_i is reset in each iteration:

$$d_i^{(k+1)} = \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)}$$

In each scaling update, ϵ is the relative machine precision. The default value is HS=0. Scaling of the Hessian can be time-consuming in the case where general linear constraints are active.

HESS

requests the display of the final Hessian matrix after optimization. If you also specify the START option, then the Hessian at the starting values is also printed.

INHESSIAN< =*r* >

INHESS< =*r* >

specifies how the initial estimate of the approximate Hessian is defined for the quasi-Newton techniques QUANEW and DBLDOG. There are two alternatives:

- If you do not use the r specification, the initial estimate of the approximate Hessian is set to the Hessian at $\boldsymbol{\theta}^{(0)}$.
- If you do use the r specification, the initial estimate of the approximate Hessian is set to the multiple of the identity matrix, $r\mathbf{I}$.

By default, if you do not specify the option INHESSIAN=*r*, the initial estimate of the approximate Hessian is set to the multiple of the identity matrix $r\mathbf{I}$, where the scalar r is computed from the magnitude of the initial gradient.

INSTEP=*r*

reduces the length of the first trial step during the line search of the first iterations. For highly nonlinear objective functions, such as the EXP function, the default initial radius of the trust-region algorithm TRUREG or DBLDOG or the default step length of the line-search algorithms can result in arithmetic overflows. If this occurs, you should specify decreasing values of $0 < r < 1$ such as INSTEP=1E−1, INSTEP=1E−2, INSTEP=1E−4, and so on, until the iteration starts successfully.

- For trust-region algorithms (TRUREG, DBLDOG), the INSTEP= option specifies a factor $r > 0$ for the initial radius $\Delta^{(0)}$ of the trust region. The default initial trust-region radius is the length of the scaled gradient. This step corresponds to the default radius factor of $r = 1$.
- For line-search algorithms (NEWRAP, CONGRA, QUANEW), the INSTEP= option specifies an upper bound for the initial step length for the line search during the first five iterations. The default initial step length is $r = 1$.
- For the Nelder-Mead simplex algorithm, using TECH=NMSIMP, the INSTEP=*r* option defines the size of the start simplex.

For more details, see the section "Computational Problems" on page 4399.

ITDETAILS

requests a more complete iteration history, including the current values of the parameter estimates, their gradients, and additional optimization statistics. For further details, see the section "Iterations" on page 4405.

LCDEACT=*r*

LCD=*r*

specifies a threshold r for the Lagrange multiplier that determines whether an active inequality constraint remains active or can be deactivated. During minimization, an active inequality constraint can be deactivated only if its Lagrange multiplier is less than the threshold value $r < 0$. The default value is

$$r = -\min(0.01, \max(0.1 \times \text{ABSGCONV}, 0.001 \times \text{gmax}^{(k)}))$$

where ABSGCONV is the value of the absolute gradient criterion, and $\text{gmax}^{(k)}$ is the maximum absolute element of the (projected) gradient $\mathbf{g}^{(k)}$ or $\mathbf{Z}'\mathbf{g}^{(k)}$. (See the section "Active Set Methods" for a definition of $\mathbf{Z}$.)

LCEPSILON=*r* > 0

LCEPS=*r* > 0

LCE=*r* > 0

specifies the range for active and violated boundary constraints. The default value is $r =$ 1E−8. During the optimization process, the introduction of rounding errors can force PROC NLMIXED to increase the value of r by a factor of $10, 100, \ldots$. If this happens, it is indicated by a message displayed in the log.

LCSINGULAR=*r* > 0

LCSING=*r* > 0

LCS=*r* > 0

specifies a criterion r, used in the update of the QR decomposition, that determines whether an active constraint is linearly dependent on a set of other active constraints. The default value is $r =$ 1E−8. The larger r becomes, the more the active constraints are recognized as being linearly dependent. If the value of r is larger than 0.1, it is reset to 0.1.

LINESEARCH=*i*

LIS=*i*

specifies the line-search method for the CONGRA, QUANEW, and NEWRAP optimization techniques. See Fletcher (1987) for an introduction to line-search techniques. The value of i can be $1, \ldots, 8$. For CONGRA, QUANEW and NEWRAP, the default value is $i = 2$.

LIS=1 specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is similar to one used by the Harwell subroutine library.

LIS=2 specifies a line-search method that needs more function than gradient calls for quadratic and cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the LSPRECISION= option.

LIS=3 specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the LSPRECISION= option.

LIS=4 specifies a line-search method that needs the same number of function and gradient calls for stepwise extrapolation and cubic interpolation.

LIS=5 specifies a line-search method that is a modified version of LIS=4.

LIS=6 specifies golden section line search (Polak 1971), which uses only function values for linear approximation.

LIS=7 specifies bisection line search (Polak 1971), which uses only function values for linear approximation.

LIS=8 specifies the Armijo line-search technique (Polak 1971), which uses only function values for linear approximation.

LIST

displays the model program and variable lists. The LIST option is a debugging feature and is not normally needed.

LISTCODE

displays the derivative tables and the compiled program code. The LISTCODE option is a debugging feature and is not normally needed.

LISTDEP

produces a report that lists, for each variable in the program, the variables that depend on it and on which it depends. The LISTDEP option is a debugging feature and is not normally needed.

LISTDER

displays a table of derivatives. This table lists each nonzero derivative computed for the problem. The LISTDER option is a debugging feature and is not normally needed.

LOGNOTE<=*n*>

writes periodic notes to the log that describe the current status of computations. It is designed for use with analyses requiring extensive CPU resources. The optional integer value n specifies the desired level of reporting detail. The default is $n = 1$. Choosing $n = 2$ adds information about the objective function values at the end of each iteration. The most detail is obtained with $n = 3$, which also reports the results of function evaluations within iterations.

LSPRECISION=*r*

LSP=*r*

specifies the degree of accuracy that should be obtained by the line-search algorithms LIS=2 and LIS=3. Usually an imprecise line search is inexpensive and successful. For more difficult optimization problems, a more precise and expensive line search might be necessary (Fletcher 1987). The second line-search method (which is the default for the NEWRAP, QUANEW, and CONGRA techniques) and the third line-search method approach exact line search for small LSPRECISION= values. If you have numerical problems, you should try to decrease the LSPRECISION= value to obtain a more precise line search. The default values are shown in the following table.

TECH=	UPDATE=	LSP default
QUANEW	DBFGS, BFGS	$r = 0.4$
QUANEW	DDFP, DFP	$r = 0.06$
CONGRA	all	$r = 0.1$
NEWRAP	no update	$r = 0.9$

For more details, see Fletcher (1987).

MAXFUNC=*i*

MAXFU=*i*

specifies the maximum number i of function calls in the optimization process. The default values are as follows:

- TRUREG, NRRIDG, NEWRAP: 125
- QUANEW, DBLDOG: 500
- CONGRA: 1000
- NMSIMP: 3000

Note that the optimization can terminate only after completing a full iteration. Therefore, the number of function calls that is actually performed can exceed the number that is specified by the MAXFUNC= option.

MAXITER=*i*

MAXIT=*i*

specifies the maximum number i of iterations in the optimization process. The default values are as follows:

- TRUREG, NRRIDG, NEWRAP: 50
- QUANEW, DBLDOG: 200
- CONGRA: 400
- NMSIMP: 1000

These default values are also valid when i is specified as a missing value.

MAXSTEP=*r*<**[***n*]>

specifies an upper bound for the step length of the line-search algorithms during the first n iterations. By default, r is the largest double-precision value and n is the largest integer available. Setting this option can improve the speed of convergence for the CONGRA, QUANEW, and NEWRAP techniques.

MAXTIME=*r*

specifies an upper limit of r seconds of CPU time for the optimization process. The default value is the largest floating-point double representation of your computer. Note that the time specified by the MAXTIME= option is checked only once at the end of each iteration. Therefore, the actual running time can be much longer than that specified by the MAXTIME= option. The actual running time includes the rest of the time needed to finish the iteration and the time needed to generate the output of the results.

METHOD=*value*

specifies the method for approximating the integral of the likelihood over the random effects. Valid values are as follows:

FIRO

specifies the first-order method of Beal and Sheiner (1982). When using METHOD=FIRO, you must specify the NORMAL distribution in the MODEL statement and you must also specify a RANDOM statement.

GAUSS

specifies adaptive Gauss-Hermite quadrature (Pinheiro and Bates 1995). You can prevent the adaptation with the NOAD option or prevent adaptive scaling with the NOADSCALE option. This is the default integration method.

HARDY

specifies Hardy quadrature based on an adaptive trapezoidal rule. This method is available only for one-dimensional integrals; that is, you must specify only one random effect.

ISAMP

specifies adaptive importance sampling (Pinheiro and Bates 1995). You can prevent the adaptation with the NOAD option or prevent adaptive scaling with the NOADSCALE option. You can use the SEED= option to specify a starting seed for the random number generation used in the importance sampling. If you do not specify a seed, or if you specify a value less than or equal to zero, the seed is generated from reading the time of day from the computer clock.

MINITER=*i*

MINIT=*i*

specifies the minimum number of iterations. The default value is 0. If you request more iterations than are actually needed for convergence to a stationary point, the optimization algorithms can behave strangely. For example, the effect of rounding errors can prevent the algorithm from continuing for the required number of iterations.

MSINGULAR=*r* > 0

MSING=*r* > 0

specifies a relative singularity criterion for the computation of the inertia (number of positive, negative, and zero eigenvalues) of the Hessian and its projected forms. The default value is 1E−12 if you do not specify the SINGHESS= option; otherwise, the default value is $\max(10\epsilon, (1\text{E} - 4) \times \text{SINGHESS})$. See the section "Covariance Matrix" on page 4402 for more information.

NOAD

requests that the Gaussian quadrature be nonadaptive; that is, the quadrature points are centered at zero for each of the random effects and the current random-effects variance matrix is used as the scale matrix.

NOADSCALE

requests nonadaptive scaling for adaptive Gaussian quadrature; that is, the quadrature points

are centered at the empirical Bayes estimates for the random effects, but the current random-effects variance matrix is used as the scale matrix. By default, the observed Hessian from the current empirical Bayes estimates is used as the scale matrix.

OPTCHECK<=$r > 0$>
computes the function values $f(\boldsymbol{\theta}_l)$ of a grid of points $\boldsymbol{\theta}_l$ in a ball of radius of r about $\boldsymbol{\theta}^*$. If you specify the OPTCHECK option without factor r, the default value is $r = 0.1$ at the starting point and $r = 0.01$ at the terminating point. If a point $\boldsymbol{\theta}_l^*$ is found with a better function value than $f(\boldsymbol{\theta}^*)$, then optimization is restarted at $\boldsymbol{\theta}_l^*$.

OUTQ=*SAS-data-set*
specifies an output data set containing the quadrature points used for numerical integration.

QFAC=$r > 0$
specifies the additive factor used to adaptively search for the number of quadrature points. For METHOD=GAUSS, the search sequence is 1, 3, 5, 7, 9, 11, $11 + r$, $11 + 2r$, ..., where the default value of r is 10. For METHOD=ISAMP, the search sequence is 10, $10 + r$, $10 + 2r$, ..., where the default value of r is 50.

QMAX=$r > 0$
specifies the maximum number of quadrature points permitted before the adaptive search is aborted. The default values are 31 for adaptive Gaussian quadrature, 61 for nonadaptive Gaussian quadrature, 160 for adaptive importance sampling, and 310 for nonadaptive importance sampling.

QPOINTS=$n > 0$
specifies the number of quadrature points to be used during evaluation of integrals. For METHOD=GAUSS, n equals the number of points used in each dimension of the random effects, resulting in a total of n^r points, where r is the number of dimensions. For METHOD=ISAMP, n specifies the total number of quadrature points regardless of the dimension of the random effects. By default, the number of quadrature points is selected adaptively, and this option disables the adaptive search.

QSCALEFAC=$r > 0$
specifies a multiplier for the scale matrix used during quadrature calculations. The default value is 1.0.

QTOL=$r > 0$
specifies the tolerance used to adaptively select the number of quadrature points. When the relative difference between two successive likelihood calculations is less than r, then the search terminates and the lesser number of quadrature points is used during the subsequent optimization process. The default value is 1E−4.

RESTART=$i > 0$
REST=$i > 0$
specifies that the QUANEW or CONGRA algorithm is restarted with a steepest descent/ascent search direction after, at most, i iterations. Default values are as follows:

- CONGRA: UPDATE=PB: restart is performed automatically, i is not used.

- CONGRA: UPDATE≠PB: $i = \min(10n, 80)$, where n is the number of parameters.
- QUANEW: i is the largest integer available.

SEED=*i*

specifies the random number seed for METHOD=ISAMP. If you do not specify a seed, or if you specify a value less than or equal to zero, the seed is generated from reading the time of day from the computer clock. The value must be less than $2^{31} - 1$.

SINGCHOL=*r* > 0

specifies the singularity criterion r for Cholesky roots of the random-effects variance matrix and scale matrix for adaptive Gaussian quadrature. The default value is 1E4 times the machine epsilon; this product is approximately 1E−12 on most computers.

SINGHESS=*r* > 0

specifies the singularity criterion r for the inversion of the Hessian matrix. The default value is 1E−8. See the ASINGULAR, MSINGULAR=, and VSINGULAR= options for more information.

SINGSWEEP=*r* > 0

specifies the singularity criterion r for inverting the variance matrix in the first-order method and the empirical Bayes Hessian matrix. The default value is 1E4 times the machine epsilon; this product is approximately 1E−12 on most computers.

SINGVAR=*r* > 0

specifies the singularity criterion r below which statistical variances are considered to equal zero. The default value is 1E4 times the machine epsilon; this product is approximately 1E−12 on most computers.

START

requests that the gradient of the log likelihood at the starting values be displayed. If you also specify the HESS option, then the starting Hessian is displayed as well.

SUBGRADIENT=*SAS-data-set*

SUBGRAD=*SAS-data-set*

specifies a SAS data set to save in models with RANDOM statement the subject-specific gradients of the integrated, marginal log-likelihood with respect to all parameters. The sum of the subject-specific gradients equals the gradient reported in the "Parameter Estimates" table. The data set contains a variable identifying the subjects.

In models without RANDOM statement the SUBGRADIENT= data set contains the observation-wise gradient. The variable identifying the SUBJECT= is then replaced with the Observation. This observation counter includes observations not used in the analysis and is reset in each BY-group.

Saving disaggregated gradient information with the SUBGRADIENT= option requires METHOD=GAUSS or METHOD=ISAMP.

TECHNIQUE=*value*

TECH=*value*

specifies the optimization technique. Valid values are as follows:

- CONGRA
  performs a conjugate-gradient optimization, which can be more precisely specified with the UPDATE= option and modified with the LINESEARCH= option. When you specify this option, UPDATE=PB by default.
- DBLDOG
  performs a version of double-dogleg optimization, which can be more precisely specified with the UPDATE= option. When you specify this option, UPDATE=DBFGS by default.
- NMSIMP
  performs a Nelder-Mead simplex optimization.
- NONE
  does not perform any optimization. This option can be used as follows:
  - to perform a grid search without optimization
  - to compute estimates and predictions that cannot be obtained efficiently with any of the optimization techniques
- NEWRAP
  performs a Newton-Raphson optimization combining a line-search algorithm with ridging. The line-search algorithm LIS=2 is the default method.
- NRRIDG
  performs a Newton-Raphson optimization with ridging.
- QUANEW
  performs a quasi-Newton optimization, which can be defined more precisely with the UPDATE= option and modified with the LINESEARCH= option. This is the default estimation method.
- TRUREG
  performs a trust region optimization.

TRACE

displays the result of each operation in each statement in the model program as it is executed. This debugging option is very rarely needed, and it produces voluminous output.

UPDATE=*method*

UPD=*method*

specifies the update method for the quasi-Newton, double-dogleg, or conjugate-gradient optimization technique. Not every update method can be used with each optimizer. See the section "Optimization Algorithms" on page 4387 for more information.

Valid methods are as follows:

- BFGS
  performs the original Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update of the inverse Hessian matrix.
- DBFGS
  performs the dual BFGS update of the Cholesky factor of the Hessian matrix. This is the default update method.

- DDFP
  performs the dual Davidon, Fletcher, and Powell (DFP) update of the Cholesky factor of the Hessian matrix.
- DFP
  performs the original DFP update of the inverse Hessian matrix.
- PB
  performs the automatic restart update method of Powell (1977) and Beale (1972).
- FR
  performs the Fletcher-Reeves update (Fletcher 1987).
- PR
  performs the Polak-Ribiere update (Fletcher 1987).
- CD
  performs a conjugate-descent update of Fletcher (1987).

VSINGULAR=*r* > 0

VSING=*r* > 0

specifies a relative singularity criterion for the computation of the inertia (number of positive, negative, and zero eigenvalues) of the Hessian and its projected forms. The default value is $r = 1\text{E}{-8}$ if the SINGHESS= option is not specified, and it is the value of SINGHESS= option otherwise. See the section "Covariance Matrix" on page 4402 for more information.

XCONV=*r***<** **[***n***]** **>**

XTOL=*r***<** **[***n***]** **>**

specifies the relative parameter convergence criterion. For all techniques except NMSIMP, termination requires a small relative parameter change in subsequent iterations:

$$\frac{\max_j |\theta_j^{(k)} - \theta_j^{(k-1)}|}{\max(|\theta_j^{(k)}|, |\theta_j^{(k-1)}|, \text{XSIZE})} \leq r$$

For the NMSIMP technique, the same formula is used, but $\boldsymbol{\theta}^{(k)}$ is defined as the vertex with the lowest function value and $\boldsymbol{\theta}^{(k-1)}$ is defined as the vertex with the highest function value in the simplex.

The default value is $r = 1\text{E}{-8}$ for the NMSIMP technique and $r = 0$ otherwise. The optional integer value n specifies the number of successive iterations for which the criterion must be satisfied before the process can be terminated.

XREF

displays a cross-reference of the variables in the program showing where each variable is referenced or given a value. The XREF listing does not include derivative variables. This option is a debugging feature and is not normally needed.

XSIZE=*r* > 0

specifies the XSIZE parameter of the relative parameter termination criterion. The default value is $r = 0$. For more details, see the XCONV= option.

ARRAY Statement

ARRAY *arrayname* **[{** *dimensions* **}] [$] [***variables and constants***] ;**

The ARRAY statement is similar to, but not exactly the same as, the ARRAY statement in the SAS DATA step, and it is exactly the same as the ARRAY statements in the NLIN, NLP, and MODEL procedures. The ARRAY statement is used to associate a name (of no more than eight characters) with a list of variables and constants. The array name is used with subscripts in the program to refer to the array elements. The following statements illustrate this:

```
array r[8] r1-r8;

do i = 1 to 8;
   r[i] = 0;
end;
```

The ARRAY statement does not support all the features of the ARRAY statement in the DATA step. It cannot be used to assign initial values to array elements. Implicit indexing of variables cannot be used; all array references must have explicit subscript expressions. Only exact array dimensions are allowed; lower-bound specifications are not supported. A maximum of six dimensions is allowed.

On the other hand, the ARRAY statement does allow both variables and constants to be used as array elements. (Constant array elements cannot have values assigned to them.) Both dimension specification and the list of elements are optional, but at least one must be specified. When the list of elements is not specified or fewer elements than the size of the array are listed, array variables are created by suffixing element numbers to the array name to complete the element list.

BOUNDS Statement

BOUNDS *b_con [, b_con. . .]* **;**

where	*b_con* :=	number *operator* parameter_list *operator* number
or	*b_con* :=	number *operator* parameter_list
or	*b_con* :=	parameter_list *operator* number
and	*operator* :=	<=, <, >=, or >

Boundary constraints are specified with a BOUNDS statement. One- or two-sided boundary constraints are allowed. The list of boundary constraints are separated by commas. For example:

```
bounds 0 <= a1-a9 X <= 1, -1 <= c2-c5;
bounds b1-b10 y >= 0;
```

You can specify more than one BOUNDS statement. If you specify more than one lower (upper) bound for the same parameter, the maximum (minimum) of these is taken.

If the maximum l_j of all lower bounds is larger than the minimum of all upper bounds u_j for the same parameter θ_j, the boundary constraint is replaced by $\theta_j := l_j := \min(u_j)$ defined by the minimum of all upper bounds specified for θ_j.

BY Statement

BY *variables* **;**

You can use a BY statement with the NLMIXED procedure to obtain separate analyses on DATA= data set observations in groups defined by the BY variables. This means that, unless TECH=NONE, an optimization problem is solved for each BY group separately. When a BY statement appears, the procedure expects the input DATA= data set to be sorted in the order of the BY variables. If your input data set is not sorted in ascending order, use one of the following alternatives:

- Use the SORT procedure with a similar BY statement to sort the data.
- Use the BY statement option NOTSORTED or DESCENDING in the BY statement for the NLMIXED procedure. As a cautionary note, the NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Use the DATASETS procedure (in Base SAS software) to create an index on the BY variables.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CONTRAST Statement

CONTRAST *'label' expression* < *, expression* > < *options* > **;**

The CONTRAST statement enables you to conduct a statistical test that several expressions simultaneously equal zero. The expressions are typically contrasts—that is, differences whose expected values equal zero under the hypothesis of interest.

In the CONTRAST statement you must provide a quoted string to identify the contrast and then a list of valid SAS expressions separated by commas. Multiple CONTRAST statements are permitted, and results from all statements are listed in a common table. PROC NLMIXED constructs approximate F tests for each statement using the delta method (Cox 1998) to approximate the variance-covariance matrix of the constituent expressions.

The following option is available in the CONTRAST statement:

DF=*d*

specifies the denominator degrees of freedom to be used in computing p values for the F statistics. The default value corresponds to the DF= option in the PROC NLMIXED statement.

ESTIMATE Statement

ESTIMATE *'label' expression < options >* **;**

The ESTIMATE statement enables you to compute an additional estimate that is a function of the parameter values. You must provide a quoted string to identify the estimate and then a valid SAS expression. Multiple ESTIMATE statements are permitted, and results from all statements are listed in a common table. PROC NLMIXED computes approximate standard errors for the estimates using the delta method (Billingsley 1986). It uses these standard errors to compute corresponding t statistics, p-values, and confidence limits.

The ECOV option in the PROC NLMIXED statement produces a table containing the approximate covariance matrix of all the additional estimates you specify. The ECORR option produces the corresponding correlation matrix. The EDER option produces a table of the derivatives of the additional estimates with respect to each of the model parameters.

The following options are available in the ESTIMATE statement:

ALPHA=*α*

specifies the alpha level to be used in computing confidence limits. The default value corresponds to the ALPHA= option in the PROC NLMIXED statement.

DF=*d*

specifies the degrees of freedom to be used in computing p-values and confidence limits. The default value corresponds to the DF= option in the PROC NLMIXED statement.

ID Statement

ID *names* **;**

The ID statement identifies additional quantities to be included in the OUT= data set of the PREDICT statement. These can be any symbols you have defined with SAS programming statements.

MODEL Statement

MODEL *dependent-variable* ∼ *distribution* **;**

The MODEL statement is the mechanism for specifying the conditional distribution of the data given the random effects. You must specify a single dependent variable from the input data set, a tilde (∼), and then a distribution with its parameters. Valid distributions are as follows.

- *normal(m,v)* specifies a normal (Gaussian) distribution with mean m and variance v.
- *binary(p)* specifies a binary (Bernoulli) distribution with probability p.
- *binomial(n,p)* specifies a binomial distribution with count n and probability p.
- *gamma(a,b)* specifies a gamma distribution with shape a and scale b.
- *negbin(n,p)* specifies a negative binomial distribution with count n and probability p.
- *poisson(m)* specifies a Poisson distribution with mean m.
- *general(ll)* specifies a general log likelihood function that you construct using SAS programming statements.

The MODEL statement must follow any SAS programming statements you specify for computing parameters of the preceding distributions. See the section "Built-in Log-Likelihood Functions" on page 4384 for expressions of the built-in conditional log-likelihood functions.

PARMS Statement

PARMS *<name_list [=numbers] [, name_list [=numbers] ...]>*
</ options> **;**

The PARMS statement lists names of parameters and specifies initial values, possibly over a grid. You can specify the parameters and values directly in a list, or you can provide the name of a SAS data set that contains them by using the DATA= option.

While the PARMS statement is not required, you are encouraged to use it to provide PROC NLMIXED with accurate starting values. Parameters not listed in the PARMS statement are assigned an initial value of 1. PROC NLMIXED considers all symbols not assigned values to be parameters, so you should specify your modeling statements carefully and check the output from the "Parameters" table to make sure the proper parameters are identified.

A list of parameter names in the PARMS statement is not separated by commas and is followed by an equal sign and a list of numbers. If the number list consists of only one number, this number defines the initial value for all the parameters listed to the left of the equal sign.

If the number list consists of more than one number, these numbers specify the grid locations for each of the parameters listed to the left of the equal sign. You can use the TO and BY keywords to

specify a number list for a grid search. If you specify a grid of points in a PARMS statement, PROC NLMIXED computes the objective function value at each grid point and chooses the best (feasible) grid point as an initial point for the optimization process. You can use the BEST= option to save memory for the storing and sorting of all grid point information.

The following options are available in the PARMS statement after a slash (/):

BEST=*i* > 0
specifies the maximum number of points displayed in the "Parameters" table, selected as the points with the maximum likelihood values. By default, all grid values are displayed.

BYDATA
enables you to assign different starting values for each BY group by using the DATA=*SAS-data-set* option during BY processing. By default, BY groups are ignored in the PARMS data set. For the BYDATA option to be effective, the DATA= data set must contain the BY variables and the same BY groups as the primary input data set. When you supply a grid of starting values with the DATA= data set and the BYDATA option is in effect, the size of the grid is determined by the first BY group.

DATA=*SAS-data-set*
specifies a SAS data set containing parameter names and starting values. The data set should be in one of two forms: narrow or wide. The narrow-form data set contains the variables Parameter and Estimate, with parameters and values listed as distinct observations. The wide-form data set has the parameters themselves as variables, and each observation provides a different set of starting values. By default, BY groups are ignored in this data set, so the same starting grid is evaluated for each BY group. You can vary the starting values for BY groups by using the BYDATA option.

PREDICT Statement

PREDICT *expression OUT=SAS-data-set* < *options* > **;**

The PREDICT statement enables you to construct predictions of an expression across all of the observations in the input data set. Any valid SAS programming expression involving the input data set variables, parameters, and random effects is valid. Predicted values are computed using the parameter estimates and empirical Bayes estimates of the random effects. Standard errors of prediction are computed using the delta method (Billingsley 1986, Cox 1998). Results are placed in an output data set that you specify with the OUT= option. Besides all variables from the input data set, the OUT= data set contains the following variables: Pred, StdErrPred, DF, tValue, Probt, Alpha, Lower, Upper. You can also add other computed quantities to this data set with the ID statement.

The following options are available in the PREDICT statement:

ALPHA=α
specifies the alpha level to be used in computing t statistics and intervals. The default value corresponds to the ALPHA= option in the PROC NLMIXED statement.

DER

requests that derivatives of the predicted expression with respect to all parameters be included in the OUT= data set. The variable names for the derivatives are the same as the parameter names with the prefix "Der_" appended. All of the derivatives are evaluated at the final estimates of the parameters and the empirical Bayes estimates of the random effects.

DF=*d*

specifies the degrees of freedom to be used in computing t statistics and intervals in the OUT= data set. The default value corresponds to the DF= option in the PROC NLMIXED statement.

RANDOM Statement

RANDOM *random-effects* ~ *distribution* **SUBJECT**=*variable* < *options* > **;**

The RANDOM statement defines the random effects and their distribution. The random effects must be represented by symbols that appear in your SAS programming statements. They typically influence the mean value of the distribution specified in the MODEL statement. The RANDOM statement consists of a list of the random effects (usually just one or two symbols), a tilde (~), the distribution for the random effects, and then a SUBJECT= variable.

NOTE: The input data set must be clustered according to the SUBJECT= variable. One easy way to accomplish this is to sort your data by the SUBJECT= variable prior to calling PROC NLMIXED. PROC NLMIXED does not sort the input data set for you; rather, it processes the data sequentially and considers an observation to be from a new subject whenever the value of its SUBJECT= variable changes from the previous observation.

The only distribution available for the random effects is normal(m,v) with mean m and variance v.

This syntax is illustrated as follows for one effect:

```
random u ~ normal(0,s2u) subject=clinic;
```

For multiple effects, you should specify bracketed vectors for m and v, the latter consisting of the lower triangle of the random-effects variance matrix listed in row order. This is illustrated for two and three random effects as follows:

```
random b1 b2 ~ normal([0,0],[g11,g21,g22]) subject=person;
random b1 b2 b3 ~ normal([0,0,0],[g11,g21,g22,g31,g32,g33])
   subject=person;
```

The SUBJECT= variable determines when new realizations of the random effects are assumed to occur. PROC NLMIXED assumes that a new realization occurs whenever the value of the SUBJECT= variable changes from the previous observation, so your input data set should be clustered according to this variable. One easy way to accomplish this is to run PROC SORT prior to calling PROC NLMIXED by using the SUBJECT= variable as the BY variable.

Only one RANDOM statement is permitted, so multilevel nonlinear mixed models are not accommodated. However, you can specify certain nested random effects structure with a single RANDOM statement (see Chapter 15 of Littell et al. (2006) for an example).

The following options are available in the RANDOM statement:

ALPHA=*α*

specifies the alpha level to be used in computing t statistics and intervals. The default value corresponds to the ALPHA= option in the PROC NLMIXED statement.

DF=*d*

specifies the degrees of freedom to be used in computing t statistics and intervals in the OUT= data set. The default value corresponds to the DF= option in the PROC NLMIXED statement.

OUT=*SAS-data-set*

requests an output data set containing empirical Bayes estimates of the random effects and their approximate standard errors of prediction.

REPLICATE Statement

REPLICATE *variable* **;**

The REPLICATE statement provides a way to accommodate models in which different subjects have identical data. This occurs most commonly when the dependent variable is binary. When you specify a REPLICATE variable, PROC NLMIXED assumes that its value indicates the number of subjects having data identical to those for the current value of the SUBJECT= variable (specified in the RANDOM statement). Only the last observation of the REPLICATE variable for each subject is used, and the replicate variable must have only positive integer values.

Note that the REPLICATE mechanism is different from using a FREQ statement in other statistical modeling procedures, such as PROC GLM, GENMOD, GLIMMIX, and LOGISTIC. A FREQ variable is used to identify grouped values for observations, essentially multiplying the log likelihood or sum of squares contribution for the observation. A REPLICATE variable is used to multiply the contribution of a subject that comprises one or more observations.

Programming Statements

This section lists the programming statements used to code the log-likelihood function in PROC NLMIXED. It also documents the differences between programming statements in PROC NLMIXED and programming statements in the SAS DATA step. The syntax of programming statements used in PROC NLMIXED is identical to that used in the CALIS and GENMOD procedures (see Chapter 25 and Chapter 37, respectively), and the MODEL procedure (see the). Most of the programming statements that can be used in the SAS DATA step can also be used in the NLMIXED procedure. See *SAS Language Reference: Dictionary* for a description of SAS programming statements. The following are valid statements:

ABORT;
CALL *name [(expression [, expression . . .])]*;
DELETE;
DO *[variable = expression*
*[***TO** *expression] [***BY** *expression]*
[, expression [**TO** *expression] [* **BY** *expression] . . .]*
]
[**WHILE** *expression] [* **UNTIL** *expression]*;
END;
GOTO *statement_label*;
IF *expression*;
IF *expression* **THEN** *program_statement*;
ELSE *program_statement*;
variable = expression;
variable + expression;
LINK *statement_label*;
PUT *[variable] [=] [...]*;
RETURN;
SELECT[(*expression* **)];**
STOP;
SUBSTR(*variable, index, length* **)=** *expression*;
WHEN *(expression) program_statement*;
OTHERWISE *program_statement*;

For the most part, the SAS programming statements work the same as they do in the SAS DATA step, as documented in *SAS Language Reference: Concepts*; however, there are the following differences:

- The ABORT statement does not allow any arguments.
- The DO statement does not allow a character index variable. Thus

```
do i = 1,2,3;
```

  is supported, but the following statement is not supported:

```
do i = 'A','B','C';
```

- The LAG function does work appropriately with PROC NLMIXED, but you can use the ZLAG function instead.
- The PUT statement, used mostly for program debugging in PROC NLMIXED, supports only some of the features of the DATA step PUT statement, and it has some new features that the DATA step PUT statement does not.
  - The PROC NLMIXED PUT statement does not support line pointers, factored lists, iteration factors, overprinting, _INFILE_, the colon (:) format modifier, or "$".
  - The PROC NLMIXED PUT statement does support expressions, but the expression must be enclosed in parentheses. For example, the following statement displays the square root of x:

```
put  (sqrt(x));
```

  - The PROC NLMIXED PUT statement supports the item _PDV_ to display a formatted listing of all variables in the program. For example, the following statement displays a much more readable listing of the variables than the _ALL_ print item:

```
put  _pdv_;
```

- The WHEN and OTHERWISE statements enable you to specify more than one target statement. That is, DO/END groups are not necessary for multiple statement WHENs. For example, the following syntax is valid:

```
select;
   when (exp1) stmt1;
               stmt2;
   when (exp2) stmt3;
               stmt4;
end;
```

When coding your programming statements, you should avoid defining variables that begin with an underscore (_), because they might conflict with internal variables created by PROC NLMIXED. The MODEL statement must come after any SAS programming statements that define or modify terms used in the construction of the log-likelihood.

Details: NLMIXED Procedure

This section contains details about the underlying theory and computations of PROC NLMIXED.

Modeling Assumptions and Notation

PROC NLMIXED operates under the following general framework for nonlinear mixed models. Assume that you have an observed data vector $\mathbf{y}_i$ for each of i subjects, $i = 1, \ldots, s$. The $\mathbf{y}_i$ are assumed to be independent across i, but within-subject covariance is likely to exist because each of the elements of $\mathbf{y}_i$ is measured on the same subject. As a statistical mechanism for modeling this within-subject covariance, assume that there exist latent random-effect vectors $\mathbf{u}_i$ of small dimension (typically one or two) that are also independent across i. Assume also that an appropriate model linking $\mathbf{y}_i$ and $\mathbf{u}_i$ exists, leading to the joint probability density function

$$p(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{u}_i)q(\mathbf{u}_i|\boldsymbol{\xi})$$

where $\mathbf{X}_i$ is a matrix of observed explanatory variables and $\boldsymbol{\phi}$ and $\boldsymbol{\xi}$ are vectors of unknown parameters.

Let $\boldsymbol{\theta} = [\boldsymbol{\phi}, \boldsymbol{\xi}]$ and assume that it is of dimension n. Then inferences about $\boldsymbol{\theta}$ are based on the marginal likelihood function

$$m(\boldsymbol{\theta}) = \prod_{i=1}^{s} \int p(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{u}_i) q(\mathbf{u}_i|\boldsymbol{\xi}) d\mathbf{u}_i$$

In particular, the function

$$f(\boldsymbol{\theta}) = -\log m(\boldsymbol{\theta})$$

is minimized over $\boldsymbol{\theta}$ numerically in order to estimate $\boldsymbol{\theta}$, and the inverse Hessian (second derivative) matrix at the estimates provides an approximate variance-covariance matrix for the estimate of $\boldsymbol{\theta}$. The function $f(\boldsymbol{\theta})$ is referred to both as the negative log likelihood function and as the objective function for optimization.

As an example of the preceding general framework, consider the nonlinear growth curve example in the section "Getting Started: NLMIXED Procedure" on page 4346. Here, the conditional distribution $p(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\phi}, u_i)$ is normal with mean

$$\frac{b_1 + u_{i1}}{1 + \exp[-(d_{ij} - b_2)/b_3]}$$

and variance σ_e^2; thus $\boldsymbol{\phi} = [b_1, b_2, b_3, \sigma_e^2]$. Also, u_i is a scalar and $q(u_i|\xi)$ is normal with mean 0 and variance σ_u^2; thus $\xi = \sigma_u^2$.

The following additional notation is also found in this chapter. The quantity $\boldsymbol{\theta}^{(k)}$ refers to the parameter vector at the kth iteration, the vector $\mathbf{g}(\boldsymbol{\theta})$ refers to the gradient vector $\nabla f(\boldsymbol{\theta})$, and the matrix $\mathbf{H}(\boldsymbol{\theta})$ refers to the Hessian $\nabla^2 f(\boldsymbol{\theta})$. Other symbols are used to denote various constants or option values.

Integral Approximations

An important part of the marginal maximum likelihood method described previously is the computation of the integral over the random effects. The default method in PROC NLMIXED for computing this integral is adaptive Gaussian quadrature as described in Pinheiro and Bates (1995). Another approximation method is the first-order method of Beal and Sheiner (1982, 1988). A description of these two methods follows.

Adaptive Gaussian Quadrature

A quadrature method approximates a given integral by a weighted sum over predefined abscissas for the random effects. A good approximation can usually be obtained with an adequate number of quadrature points as well as appropriate centering and scaling of the abscissas. Adaptive Gaussian quadrature for the integral over $\mathbf{u}_i$ centers the integral at the empirical Bayes estimate of $\mathbf{u}_i$, defined as the vector $\widehat{\mathbf{u}}_i$ that minimizes

$$-\log\left[p(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{u}_i) q(\mathbf{u}_i|\boldsymbol{\xi})\right]$$

with $\boldsymbol{\phi}$ and $\boldsymbol{\xi}$ set equal to their current estimates. The final Hessian matrix from this optimization can be used to scale the quadrature abscissas.

Suppose $(z_j, w_j; j = 1, \ldots, p)$ denote the standard Gauss-Hermite abscissas and weights (Golub and Welsch 1969, or Table 25.10 of Abramowitz and Stegun 1972). The adaptive Gaussian quadrature integral approximation is as follows:

$$\int p(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{u}_i) q(\mathbf{u}_i|\boldsymbol{\xi}) d\mathbf{u}_i \approx$$
$$2^{r/2} |\boldsymbol{\Gamma}(\mathbf{X}_i, \boldsymbol{\theta})|^{-1/2} \sum_{j_1=1}^{p} \cdots \sum_{j_r=1}^{p} \left[p(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{a}_{j_1,\ldots,j_r}) q(\mathbf{a}_{j_1,\ldots,j_r}|\boldsymbol{\xi}) \prod_{k=1}^{r} w_{j_k} \exp z_{j_k}^2 \right]$$

where r is the dimension of u_i, $\boldsymbol{\Gamma}(\mathbf{X}_i, \boldsymbol{\theta})$ is the Hessian matrix from the empirical Bayes minimization, $\mathbf{z}_{j_1,\ldots,j_r}$ is a vector with elements $(z_{j_1}, \ldots, z_{j_r})$, and

$$\mathbf{a}_{j_1,\ldots,j_r} = \widehat{\mathbf{u}}_i + 2^{1/2} \boldsymbol{\Gamma}(\mathbf{X}_i, \boldsymbol{\theta})^{-1/2} \mathbf{z}_{j_1,\ldots,j_r}$$

PROC NLMIXED selects the number of quadrature points adaptively by evaluating the log-likelihood function at the starting values of the parameters until two successive evaluations have a relative difference less than the value of the QTOL= option. The specific search sequence is described under the QFAC= option. Using the QPOINTS= option, you can adjust the number of quadrature points p to obtain different levels of accuracy. Setting $p = 1$ results in the Laplacian approximation as described in Beal and Sheiner (1992), Wolfinger (1993), Vonesh (1992, 1996), Vonesh and Chinchilli (1997), and Wolfinger and Lin (1997).

The NOAD option in the PROC NLMIXED statement requests nonadaptive Gaussian quadrature. Here all $\widehat{\mathbf{u}}_i$ are set equal to zero, and the Cholesky root of the estimated variance matrix of the random effects is substituted for $\boldsymbol{\Gamma}(\mathbf{X}_i, \boldsymbol{\theta})^{-1/2}$ in the preceding expression for $\mathbf{a}_{j_1,\ldots,j_r}$. In this case derivatives are computed using the algorithm of Smith (1995). The NOADSCALE option requests the same scaling substitution but with the empirical Bayes $\widehat{\mathbf{u}}_i$.

PROC NLMIXED computes the derivatives of the adaptive Gaussian quadrature approximation when carrying out the default dual quasi-Newton optimization.

First-Order Method

Another integral approximation available in PROC NLMIXED is the first-order method of Beal and Sheiner (1982, 1988) and Sheiner and Beal (1985). This approximation is used only in the case where $p(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{u}_i)$ is normal—that is,

$$p(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{u}_i) = (2\pi)^{-n_i/2} |\mathbf{R}_i(\mathbf{X}_i, \boldsymbol{\phi})|^{-1/2}$$
$$\exp\left\{-(1/2)\left[\mathbf{y}_i - \mathbf{m}_i(\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{u}_i)\right]' \mathbf{R}_i(\mathbf{X}_i, \boldsymbol{\phi})^{-1} \left[\mathbf{y}_i - \mathbf{m}_i(\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{u}_i)\right]\right\}$$

where n_i is the dimension of $\mathbf{y}_i$, $\mathbf{R}_i$ is a diagonal variance matrix, and $\mathbf{m}_i$ is the conditional mean vector of $\mathbf{y}_i$.

The first-order approximation is obtained by expanding $\mathbf{m}_i(\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{u}_i)$ with a one-term Taylor series expansion about $\mathbf{u}_i = \mathbf{0}$, resulting in the approximation

$$p(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{u}_i) \approx (2\pi)^{-n_i/2} |\mathbf{R}_i(\mathbf{X}_i, \boldsymbol{\phi})|^{-1/2} \exp\left(-(1/2)\left[\mathbf{y}_i - \mathbf{m}_i(\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{0}) - \mathbf{Z}_i(\mathbf{X}_i, \boldsymbol{\phi})\mathbf{u}_i\right]' \mathbf{R}_i(\mathbf{X}_i, \boldsymbol{\phi})^{-1}\left[\mathbf{y}_i - \mathbf{m}_i(\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{0}) - \mathbf{Z}_i(\mathbf{X}_i, \boldsymbol{\phi})\mathbf{u}_i\right]\right)$$

where $\mathbf{Z}_i(\mathbf{X}_i, \boldsymbol{\phi})$ is the Jacobian matrix $\partial \mathbf{m}_i(\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{u}_i)/\partial \mathbf{u}_i$ evaluated at $\mathbf{u}_i = \mathbf{0}$.

Assuming that $q(\mathbf{u}_i|\boldsymbol{\xi})$ is normal with mean $\mathbf{0}$ and variance matrix $\mathbf{G}(\boldsymbol{\xi})$, the first-order integral approximation is computable in closed form after completing the square:

$$\int p(\mathbf{y}_i|\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{u}_i) q(\mathbf{u}_i|\boldsymbol{\xi})\, d\mathbf{u}_i \approx (2\pi)^{-n_i/2} |\mathbf{V}_i(\mathbf{X}_i, \boldsymbol{\theta})|^{-1/2} \exp\left(-(1/2)\left[\mathbf{y}_i - \mathbf{m}_i(\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{0})\right]' \mathbf{V}_i(\mathbf{X}_i, \boldsymbol{\theta})^{-1}\left[\mathbf{y}_i - \mathbf{m}_i(\mathbf{X}_i, \boldsymbol{\phi}, \mathbf{0})\right]\right)$$

where $\mathbf{V}_i(\mathbf{X}_i, \boldsymbol{\theta}) = \mathbf{Z}_i(\mathbf{X}_i, \boldsymbol{\phi})\mathbf{G}(\boldsymbol{\xi})\mathbf{Z}_i(\mathbf{X}_i, \boldsymbol{\phi})' + \mathbf{R}_i(\mathbf{X}_i, \boldsymbol{\phi})$. The resulting approximation for $f(\boldsymbol{\theta})$ is then minimized over $\boldsymbol{\theta} = [\boldsymbol{\phi}, \boldsymbol{\xi}]$ to obtain the first-order estimates. PROC NLMIXED uses finite-difference derivatives of the first-order integral approximation when carrying out the default dual quasi-Newton optimization.

Built-in Log-Likelihood Functions

This section displays the basic formulas used by the NLMIXED procedure to compute the conditional log-likelihood functions of the data given the random effects. Note, however, that in addition to these basic equations, the NLMIXED procedure employs a number of checks for missing values and floating-point arithmetic. You can see the entire program used by the NLMIXED procedure to compute the conditional log-likelihood functions $l(\boldsymbol{\phi}; y)$ by adding the LIST debugging option to the PROC NLMIXED statement.

$Y \sim \text{normal}(m, v)$

$$l(m, v; y) = -\frac{1}{2}\left(\log\{2\pi\} + \frac{(y-m)^2}{v} + \log\{v\}\right)$$
$$\text{E}[Y] = m$$
$$\text{Var}[Y] = v$$
$$v > 0$$

$Y \sim \text{binary}(p)$

$$
\begin{aligned}
l_1(p;y) &= \begin{cases} y \ \log\{p\} & y > 0 \\ 0 & \text{otherwise} \end{cases} \\
l_2(p;y) &= \begin{cases} (1-y) \ \log\{1-p\} & y < 1 \\ 0 & \text{otherwise} \end{cases} \\
l(p;y) &= l_1(p;y) + l_2(p;y) \\
\mathrm{E}[Y] &= p \\
\mathrm{Var}[Y] &= p\,(1-p) \\
0 &< p < 1
\end{aligned}
$$

$Y \sim \text{binomial}(n, p)$

$$
\begin{aligned}
l_c &= \log\{\Gamma(n+1)\} - \log\{\Gamma(y+1)\} - \log\{\Gamma(n-y+1)\} \\
l_1(n,p;y) &= \begin{cases} y \ \log\{p\} & y > 0 \\ 0 & \text{otherwise} \end{cases} \\
l_2(n,p;y) &= \begin{cases} (n-y) \ \log\{1-p\} & n-y > 0 \\ 0 & \text{otherwise} \end{cases} \\
l(n,p;y) &= l_c + l_1(n,p;y) + l_2(n,p;y) \\
\mathrm{E}[Y] &= n\,p \\
\mathrm{Var}[Y] &= n\,p\,(1-p) \\
0 &< p < 1
\end{aligned}
$$

$Y \sim \text{gamma}(a, b)$

$$
\begin{aligned}
l(a,b;y) &= -a\log\{b\} - \log\{\Gamma(a)\} + (a-1)\log\{y\} - y/b \\
\mathrm{E}[Y] &= ab \\
\mathrm{Var}[Y] &= ab^2 \\
a &> 0 \\
b &> 0
\end{aligned}
$$

This parameterization of the gamma distribution differs from the parameterization used in the GLIMMIX and GENMOD procedures. The following statements show the equivalent reparameterization in the NLMIXED procedure that fits a generalized linear model for gamma-distributed data in the parameterization of the GLIMMIX procedure:

```
proc glimmix;
   model y = x / dist=gamma s;
run;

proc nlmixed;
   parms b0=1 b1=0 scale=14;
   linp = b0 + b1*x;
   mu   = exp(linp);
   b    = mu/scale;
   model y ~ gamma(scale,b);
run;
```

$Y \sim \text{negbin}(n, p)$

$$
\begin{aligned}
l(n, p; y) &= \log\{\Gamma(n + y)\} - \log\{\Gamma(n)\} - \log\{\Gamma(y + 1)\} \\
&\quad + n \log\{p\} + y \log\{1 - p\} \\
\text{E}[Y] &= kP = k\left(\frac{1-p}{p}\right) \\
\text{Var}[Y] &= kP(1 - P) = k\left(\frac{1-p}{p}\right)\frac{1}{p} \\
n &\geq 0 \\
0 &< p < 1
\end{aligned}
$$

This form of the negative binomial distribution is one of the many parameterizations in which the mass function or log-likelihood function appears. Another common parameterization uses

$$
\begin{aligned}
l(n, p; y) &= \log\{\Gamma(n + y)\} - \log\{\Gamma(n)\} - \log\{\Gamma(y + 1)\} \\
&\quad + n \log\{1 - P/(1 + P)\} + y \log\{P/(1 + P)\}
\end{aligned}
$$

with $P = (1 - p)/p$, $P > 0$.

Note that the parameter n can be real-numbered; it does not have to be integer-valued. The parameterization of the negative binomial distribution in the NLMIXED procedure differs from that in the GLIMMIX and GENMOD procedures. The following statements show the equivalent formulations for maximum likelihood estimation in the GLIMMIX and NLMIXED procedures in a negative binomial regression model:

```
proc glimmix;
  model y = x / dist=negbin s;
run;

proc nlmixed;
   parms b0=3, b1=1, k=0.8;
   linp = b0 + b1*x;
   mu = exp(linp);
   p  = 1/(1+mu*k);
   model y ~ negbin(1/k,p);
run;
```

$Y \sim \text{Poisson}(m)$

$$
\begin{aligned}
l(m; y) &= y \log\{m\} - m - \log\{\Gamma(y + 1) \\
\text{E}[Y] &= m \\
\text{Var}[Y] &= m \\
m &> 0
\end{aligned}
$$

Optimization Algorithms

There are several optimization techniques available in PROC NLMIXED. You can choose a particular optimizer with the TECH= option in the PROC NLMIXED statement.

Algorithm	TECH=
trust region method	TRUREG
Newton-Raphson method with line search	NEWRAP
Newton-Raphson method with ridging	NRRIDG
quasi-Newton methods (DBFGS, DDFP, BFGS, DFP)	QUANEW
double-dogleg method (DBFGS, DDFP)	DBLDOG
conjugate gradient methods (PB, FR, PR, CD)	CONGRA
Nelder-Mead simplex method	NMSIMP

No algorithm for optimizing general nonlinear functions exists that always finds the global optimum for a general nonlinear minimization problem in a reasonable amount of time. Since no single optimization technique is invariably superior to others, PROC NLMIXED provides a variety of optimization techniques that work well in various circumstances. However, you can devise problems for which none of the techniques in PROC NLMIXED can find the correct solution. Moreover, nonlinear optimization can be computationally expensive in terms of time and memory, so you must be careful when matching an algorithm to a problem.

All optimization techniques in PROC NLMIXED use $O(n^2)$ memory except the conjugate gradient methods, which use only $O(n)$ of memory and are designed to optimize problems with many parameters. Since the techniques are iterative, they require the repeated computation of the following:

- the function value (optimization criterion)
- the gradient vector (first-order partial derivatives)
- for some techniques, the (approximate) Hessian matrix (second-order partial derivatives)

However, since each of the optimizers requires different derivatives, some computational efficiencies can be gained. The following table shows, for each optimization technique, which derivatives are required (FOD: first-order derivatives; SOD: second-order derivatives).

Algorithm	FOD	SOD
TRUREG	x	x
NEWRAP	x	x
NRRIDG	x	x
QUANEW	x	-
DBLDOG	x	-
CONGRA	x	-
NMSIMP	-	-

Each optimization method employs one or more convergence criteria that determine when it has converged. The various termination criteria are listed and described in the "PROC NLMIXED Statement" section. An algorithm is considered to have converged when any one of the convergence criteria is satisfied. For example, under the default settings, the QUANEW algorithm will converge if ABSGCONV< 1E−5, FCONV< $10^{-FDIGITS}$, or GCONV< 1E−8.

Choosing an Optimization Algorithm

The factors that go into choosing a particular optimization technique for a particular problem are complex and can involve trial and error.

For many optimization problems, computing the gradient takes more computer time than computing the function value, and computing the Hessian sometimes takes *much* more computer time and memory than computing the gradient, especially when there are many decision variables. Unfortunately, optimization techniques that do not use some kind of Hessian approximation usually require many more iterations than techniques that do use a Hessian matrix, and as a result the total run time of these techniques is often longer. Techniques that do not use the Hessian also tend to be less reliable. For example, they can more easily terminate at stationary points rather than at global optima.

A few general remarks about the various optimization techniques follow:

- The second-derivative methods TRUREG, NEWRAP, and NRRIDG are best for small problems where the Hessian matrix is not expensive to compute. Sometimes the NRRIDG algorithm can be faster than the TRUREG algorithm, but TRUREG can be more stable. The NRRIDG algorithm requires only one matrix with $n(n+1)/2$ double words; TRUREG and NEWRAP require two such matrices.

- The first-derivative methods QUANEW and DBLDOG are best for medium-sized problems where the objective function and the gradient are much faster to evaluate than the Hessian. The QUANEW and DBLDOG algorithms, in general, require more iterations than TRUREG, NRRIDG, and NEWRAP, but each iteration can be much faster. The QUANEW and DBLDOG algorithms require only the gradient to update an approximate Hessian, and they require slightly less memory than TRUREG or NEWRAP (essentially one matrix with $n(n+1)/2$ double words). QUANEW is the default optimization method.

- The first-derivative method CONGRA is best for large problems where the objective function and the gradient can be computed much faster than the Hessian and where too much memory is required to store the (approximate) Hessian. The CONGRA algorithm, in general, requires more iterations than QUANEW or DBLDOG, but each iteration can be much faster. Since CONGRA requires only a factor of n double-word memory, many large applications of PROC NLMIXED can be solved only by CONGRA.

- The no-derivative method NMSIMP is best for small problems where derivatives are not continuous or are very difficult to compute.

Algorithm Descriptions

Some details about the optimization techniques follow.

Trust Region Optimization (TRUREG)

The trust region method uses the gradient $\mathbf{g}(\boldsymbol{\theta}^{(k)})$ and the Hessian matrix $\mathbf{H}(\boldsymbol{\theta}^{(k)})$; thus, it requires that the objective function $f(\boldsymbol{\theta})$ have continuous first- and second-order derivatives inside the feasible region.

The trust region method iteratively optimizes a quadratic approximation to the nonlinear objective function within a hyperelliptic trust region with radius Δ that constrains the step size corresponding to the quality of the quadratic approximation. The trust region method is implemented using Dennis, Gay, and Welsch (1981), Gay (1983), and Moré and Sorensen (1983).

The trust region method performs well for small- to medium-sized problems, and it does not need many function, gradient, and Hessian calls. However, if the computation of the Hessian matrix is computationally expensive, one of the (dual) quasi-Newton or conjugate gradient algorithms might be more efficient.

Newton-Raphson Optimization with Line Search (NEWRAP)

The NEWRAP technique uses the gradient $\mathbf{g}(\boldsymbol{\theta}^{(k)})$ and the Hessian matrix $\mathbf{H}(\boldsymbol{\theta}^{(k)})$; thus, it requires that the objective function have continuous first- and second-order derivatives inside the feasible region. If second-order derivatives are computed efficiently and precisely, the NEWRAP method can perform well for medium-sized to large problems, and it does not need many function, gradient, and Hessian calls.

This algorithm uses a pure Newton step when the Hessian is positive definite and when the Newton step reduces the value of the objective function successfully. Otherwise, a combination of ridging and line search is performed to compute successful steps. If the Hessian is not positive definite, a multiple of the identity matrix is added to the Hessian matrix to make it positive definite (Eskow and Schnabel 1991).

In each iteration, a line search is performed along the search direction to find an approximate optimum of the objective function. The default line-search method uses quadratic interpolation and cubic extrapolation (LINESEARCH=2).

Newton-Raphson Ridge Optimization (NRRIDG)

The NRRIDG technique uses the gradient $\mathbf{g}(\boldsymbol{\theta}^{(k)})$ and the Hessian matrix $\mathbf{H}(\boldsymbol{\theta}^{(k)})$; thus, it requires that the objective function have continuous first- and second-order derivatives inside the feasible region.

This algorithm uses a pure Newton step when the Hessian is positive definite and when the Newton step reduces the value of the objective function successfully. If at least one of these two conditions is not satisfied, a multiple of the identity matrix is added to the Hessian matrix.

The NRRIDG method performs well for small- to medium-sized problems, and it does not require many function, gradient, and Hessian calls. However, if the computation of the Hessian matrix is computationally expensive, one of the (dual) quasi-Newton or conjugate gradient algorithms might be more efficient.

Since the NRRIDG technique uses an orthogonal decomposition of the approximate Hessian, each iteration of NRRIDG can be slower than that of the NEWRAP technique, which works with Cholesky decomposition. Usually, however, NRRIDG requires fewer iterations than NEWRAP.

Quasi-Newton Optimization (QUANEW)

The (dual) quasi-Newton method uses the gradient $\mathbf{g}(\boldsymbol{\theta}^{(k)})$, and it does not need to compute second-order derivatives since they are approximated. It works well for medium to moderately large optimization problems where the objective function and the gradient are much faster to compute than the Hessian; but, in general, it requires more iterations than the TRUREG, NEWRAP, and NRRIDG techniques, which compute second-order derivatives. QUANEW is the default optimization algorithm because it provides an appropriate balance between the speed and stability required for most nonlinear mixed model applications.

The QUANEW technique is one of the following, depending on the value of the UPDATE= option.

- the original quasi-Newton algorithm, which updates an approximation of the inverse Hessian
- the dual quasi-Newton algorithm, which updates the Cholesky factor of an approximate Hessian (default)

You can specify four update formulas with the UPDATE= option:

- DBFGS performs the dual Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update of the Cholesky factor of the Hessian matrix. This is the default.
- DDFP performs the dual Davidon, Fletcher, and Powell (DFP) update of the Cholesky factor of the Hessian matrix.
- BFGS performs the original BFGS update of the inverse Hessian matrix.
- DFP performs the original DFP update of the inverse Hessian matrix.

In each iteration, a line search is performed along the search direction to find an approximate optimum. The default line-search method uses quadratic interpolation and cubic extrapolation to obtain a step size α satisfying the Goldstein conditions. One of the Goldstein conditions can be violated if the feasible region defines an upper limit of the step size. Violating the left-side Goldstein condition can affect the positive definiteness of the quasi-Newton update. In that case, either the update is skipped or the iterations are restarted with an identity matrix, resulting in the steepest descent or ascent search direction. You can specify line-search algorithms other than the default with the LINESEARCH= option.

The QUANEW algorithm uses its own line-search technique. No options and parameters (except the INSTEP= option) controlling the line search in the other algorithms apply here. In several

applications, large steps in the first iterations are troublesome. You can use the INSTEP= option to impose an upper bound for the step size α during the first five iterations. You can also use the INHESSIAN=r option to specify a different starting approximation for the Hessian. If you specify only the INHESSIAN option, the Cholesky factor of a (possibly ridged) finite difference approximation of the Hessian is used to initialize the quasi-Newton update process. The values of the LCSINGULAR=, LCEPSILON=, and LCDEACT= options, which control the processing of linear and boundary constraints, are valid only for the quadratic programming subroutine used in each iteration of the QUANEW algorithm.

Double-Dogleg Optimization (DBLDOG)

The double-dogleg optimization method combines the ideas of the quasi-Newton and trust region methods. In each iteration, the double-dogleg algorithm computes the step $\mathbf{s}^{(k)}$ as the linear combination of the steepest descent or ascent search direction $\mathbf{s}_1^{(k)}$ and a quasi-Newton search direction $\mathbf{s}_2^{(k)}$:

$$\mathbf{s}^{(k)} = \alpha_1 \mathbf{s}_1^{(k)} + \alpha_2 \mathbf{s}_2^{(k)}$$

The step is requested to remain within a prespecified trust region radius; see Fletcher (1987, p. 107). Thus, the DBLDOG subroutine uses the dual quasi-Newton update but does not perform a line search. You can specify two update formulas with the UPDATE= option:

- DBFGS performs the dual Broyden, Fletcher, Goldfarb, and Shanno update of the Cholesky factor of the Hessian matrix. This is the default.
- DDFP performs the dual Davidon, Fletcher, and Powell update of the Cholesky factor of the Hessian matrix.

The double-dogleg optimization technique works well for medium to moderately large optimization problems where the objective function and the gradient are much faster to compute than the Hessian. The implementation is based on Dennis and Mei (1979) and Gay (1983), but it is extended for dealing with boundary and linear constraints. The DBLDOG technique generally requires more iterations than the TRUREG, NEWRAP, and NRRIDG techniques, which require second-order derivatives; however, each of the DBLDOG iterations is computationally cheap. Furthermore, the DBLDOG technique requires only gradient calls for the update of the Cholesky factor of an approximate Hessian.

Conjugate Gradient Optimization (CONGRA)

Second-order derivatives are not required by the CONGRA algorithm and are not even approximated. The CONGRA algorithm can be expensive in function and gradient calls, but it requires only $O(n)$ memory for unconstrained optimization. In general, many iterations are required to obtain a precise solution, but each of the CONGRA iterations is computationally cheap. You can specify four different update formulas for generating the conjugate directions by using the UPDATE= option:

- PB performs the automatic restart update method of Powell (1977) and Beale (1972). This is the default.

- FR performs the Fletcher-Reeves update (Fletcher 1987).
- PR performs the Polak-Ribiere update (Fletcher 1987).
- CD performs a conjugate-descent update of Fletcher (1987).

The default, UPDATE=PB, behaved best in most test examples. You are advised to avoid the option UPDATE=CD, which behaved worst in most test examples.

The CONGRA subroutine should be used for optimization problems with large n. For the unconstrained or boundary constrained case, CONGRA requires only $O(n)$ bytes of working memory, whereas all other optimization methods require order $O(n^2)$ bytes of working memory. During n successive iterations, uninterrupted by restarts or changes in the working set, the conjugate gradient algorithm computes a cycle of n conjugate search directions. In each iteration, a line search is performed along the search direction to find an approximate optimum of the objective function. The default line-search method uses quadratic interpolation and cubic extrapolation to obtain a step size α satisfying the Goldstein conditions. One of the Goldstein conditions can be violated if the feasible region defines an upper limit for the step size. Other line-search algorithms can be specified with the LINESEARCH= option.

Nelder-Mead Simplex Optimization (NMSIMP)

The Nelder-Mead simplex method does not use any derivatives and does not assume that the objective function has continuous derivatives. The objective function itself needs to be continuous. This technique is quite expensive in the number of function calls, and it might be unable to generate precise results for $n \gg 40$.

The original Nelder-Mead simplex algorithm is implemented and extended to boundary constraints. This algorithm does not compute the objective for infeasible points, but it changes the shape of the simplex adapting to the nonlinearities of the objective function, which contributes to an increased speed of convergence. It uses a special termination criterion.

Finite-Difference Approximations of Derivatives

The FD= and FDHESSIAN= options specify the use of finite-difference approximations of the derivatives. The FD= option specifies that all derivatives are approximated using function evaluations, and the FDHESSIAN= option specifies that second-order derivatives are approximated using gradient evaluations.

Computing derivatives by finite-difference approximations can be very time-consuming, especially for second-order derivatives based only on values of the objective function (FD= option). If analytical derivatives are difficult to obtain (for example, if a function is computed by an iterative process), you might consider one of the optimization techniques that use first-order derivatives only (QUANEW, DBLDOG, or CONGRA). In the expressions that follow, $\boldsymbol{\theta}$ denotes the parameter vector, h_i denotes the step size for the ith parameter, and $\mathbf{e}_i$ is a vector of zeros with a 1 in the ith position.

Forward-Difference Approximations

The forward-difference derivative approximations consume less computer time, but they are usually not as precise as approximations that use central-difference formulas.

- For first-order derivatives, n additional function calls are required:

$$g_i = \frac{\partial f}{\partial \theta_i} \approx \frac{f(\boldsymbol{\theta} + h_i \mathbf{e}_i) - f(\boldsymbol{\theta})}{h_i}$$

- For second-order derivatives based on function calls only (Dennis and Schnabel 1983, p. 80), $n + n^2/2$ additional function calls are required for dense Hessian:

$$\frac{\partial^2 f}{\partial \theta_i \partial \theta_j} \approx \frac{f(\boldsymbol{\theta} + h_i \mathbf{e}_i + h_j \mathbf{e}_j) - f(\boldsymbol{\theta} + h_i \mathbf{e}_i) - f(\boldsymbol{\theta} + h_j \mathbf{e}_j) + f(\boldsymbol{\theta})}{h_i h_j}$$

- For second-order derivatives based on gradient calls (Dennis and Schnabel 1983, p. 103), n additional gradient calls are required:

$$\frac{\partial^2 f}{\partial \theta_i \partial \theta_j} \approx \frac{g_i(\boldsymbol{\theta} + h_j \mathbf{e}_j) - g_i(\boldsymbol{\theta})}{2h_j} + \frac{g_j(\boldsymbol{\theta} + h_i \mathbf{e}_i) - g_j(\boldsymbol{\theta})}{2h_i}$$

Central-Difference Approximations

Central-difference approximations are usually more precise, but they consume more computer time than approximations that use forward-difference derivative formulas.

- For first-order derivatives, $2n$ additional function calls are required:

$$g_i = \frac{\partial f}{\partial \theta_i} \approx \frac{f(\boldsymbol{\theta} + h_i \mathbf{e}_i) - f(\boldsymbol{\theta} - h_i \mathbf{e}_i)}{2h_i}$$

- For second-order derivatives based on function calls only (Abramowitz and Stegun 1972, p. 884), $2n + 4n^2/2$ additional function calls are required.

$$\frac{\partial^2 f}{\partial \theta_i^2} \approx \frac{-f(\boldsymbol{\theta} + 2h_i \mathbf{e}_i) + 16 f(\boldsymbol{\theta} + h_i \mathbf{e}_i) - 30 f(\boldsymbol{\theta}) + 16 f(\boldsymbol{\theta} - h_{\mathbf{i}} e_i) - f(\boldsymbol{\theta} - 2h_i \mathbf{e}_i)}{12 h_i^2}$$

$$\frac{\partial^2 f}{\partial \theta_i \partial \theta_j} \approx \frac{f(\boldsymbol{\theta} + h_i \mathbf{e}_i + h_j \mathbf{e}_j) - f(\boldsymbol{\theta} + h_i \mathbf{e}_i - h_j \mathbf{e}_j) - f(\boldsymbol{\theta} - h_i \mathbf{e}_i + h_j \mathbf{e}_j) + f(\boldsymbol{\theta} - h_i \mathbf{e}_i - h_j \mathbf{e}_j)}{4 h_i h_j}$$

- For second-order derivatives based on gradient calls, $2n$ additional gradient calls are required:

$$\frac{\partial^2 f}{\partial \theta_i \partial \theta_j} \approx \frac{g_i(\boldsymbol{\theta} + h_j \mathbf{e}_j) - g_i(\boldsymbol{\theta} - h_j \mathbf{e}_j)}{4h_j} + \frac{g_j(\boldsymbol{\theta} + h_i \mathbf{e}_i) - g_j(\boldsymbol{\theta} - h_i \mathbf{e}_i)}{4h_i}$$

You can use the FDIGITS= option to specify the number of accurate digits in the evaluation of the objective function. This specification is helpful in determining an appropriate interval size h to be used in the finite-difference formulas.

The step sizes h_j, $j = 1, \ldots, n$ are defined as follows:

- For the forward-difference approximation of first-order derivatives that use function calls and second-order derivatives that use gradient calls, $h_j = \sqrt[2]{\eta}(1 + |\theta_j|)$.
- For the forward-difference approximation of second-order derivatives that use only function calls and all central-difference formulas, $h_j = \sqrt[3]{\eta}(1 + |\theta_j|)$.

The value of η is defined by the FDIGITS= option:

- If you specify the number of accurate digits by using FDIGITS=r, η is set to 10^{-r}.
- If you do not specify the FDIGITS= option, η is set to the machine precision ϵ.

Hessian Scaling

The rows and columns of the Hessian matrix can be scaled when you use the trust region, Newton-Raphson, and double-dogleg optimization techniques. Each element $H_{i,j}$, $i, j = 1, \ldots, n$ is divided by the scaling factor $d_i d_j$, where the scaling vector $d = (d_1, \ldots, d_n)$ is iteratively updated in a way specified by the HESCAL=i option, as follows:

$i = 0$: No scaling is done (equivalent to $d_i = 1$).

$i \neq 0$: First iteration and each restart iteration sets:

$$d_i^{(0)} = \sqrt{\max(|H_{i,i}^{(0)}|, \epsilon)}$$

$i = 1$: Refer to Moré (1978):

$$d_i^{(k+1)} = \max\left[d_i^{(k)}, \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)}\right]$$

$i = 2$: Refer to Dennis, Gay, and Welsch (1981):

$$d_i^{(k+1)} = \max\left[0.6d_i^{(k)}, \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)}\right]$$

$i = 3$: d_i is reset in each iteration:

$$d_i^{(k+1)} = \sqrt{\max(|H_{i,i}^{(k)}|, \epsilon)}$$

In the preceding equations, ϵ is the relative machine precision or, equivalently, the largest double-precision value that, when added to 1, results in 1.

Active Set Methods

The parameter vector $\boldsymbol{\theta} \in \mathcal{R}^n$ can be subject to a set of m linear equality and inequality constraints:

$$\sum_{j=1}^{n} a_{ij}\theta_j = b_i \qquad i = 1, \ldots, m_e$$

$$\sum_{j=1}^{n} a_{ij}\theta_j \geq b_i \qquad i = m_e + 1, \ldots, m$$

The coefficients a_{ij} and right-hand sides b_i of the equality and inequality constraints are collected in the $m \times n$ matrix $\mathbf{A}$ and the m vector $\mathbf{b}$.

The m linear constraints define a feasible region $\mathcal{G}$ in $\mathcal{R}^n$ that must contain the point $\boldsymbol{\theta}_*$ that minimizes the problem. If the feasible region $\mathcal{G}$ is empty, no solution to the optimization problem exists.

In PROC NLMIXED, all optimization techniques use *active set methods*. The iteration starts with a feasible point $\boldsymbol{\theta}^{(0)}$, which you can provide or which can be computed by the Schittkowski and Stoer (1979) algorithm implemented in PROC NLMIXED. The algorithm then moves from one feasible point $\boldsymbol{\theta}^{(k-1)}$ to a better feasible point $\boldsymbol{\theta}^{(k)}$ along a feasible search direction $\mathbf{s}^{(k)}$,

$$\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k-1)} + \alpha^{(k)}\mathbf{s}^{(k)} \quad , \quad \alpha^{(k)} > 0$$

Theoretically, the path of points $\boldsymbol{\theta}^{(k)}$ never leaves the feasible region $\mathcal{G}$ of the optimization problem, but it can reach its boundaries. The active set $\mathcal{A}^{(k)}$ of point $\boldsymbol{\theta}^{(k)}$ is defined as the index set of all linear equality constraints and those inequality constraints that are satisfied at $\boldsymbol{\theta}^{(k)}$. If no constraint is active $\boldsymbol{\theta}^{(k)}$, the point is located in the interior of $\mathcal{G}$, and the active set $\mathcal{A}^{(k)} = \emptyset$ is empty. If the point $\boldsymbol{\theta}^{(k)}$ in iteration k hits the boundary of inequality constraint i, this constraint i becomes active and is added to $\mathcal{A}^{(k)}$. Each equality constraint and each active inequality constraint reduce the dimension (degrees of freedom) of the optimization problem.

In practice, the active constraints can be satisfied only with finite precision. The LCEPSILON=*r* option specifies the range for active and violated linear constraints. If the point $\boldsymbol{\theta}^{(k)}$ satisfies the condition

$$\left| \sum_{j=1}^{n} a_{ij}\theta_j^{(k)} - b_i \right| \leq t$$

where $t = r(|b_i| + 1)$, the constraint i is recognized as an active constraint. Otherwise, the constraint i is either an inactive inequality or a violated inequality or equality constraint. Due to rounding errors in computing the projected search direction, error can be accumulated so that an iterate $\boldsymbol{\theta}^{(k)}$ steps out of the feasible region.

In those cases, PROC NLMIXED might try to pull the iterate $\boldsymbol{\theta}^{(k)}$ back into the feasible region. However, in some cases the algorithm needs to increase the feasible region by increasing the LCEPSILON=*r* value. If this happens, a message is displayed in the log output.

If the algorithm cannot improve the value of the objective function by moving from an active constraint back into the interior of the feasible region, it makes this inequality constraint an equality constraint in the next iteration. This means that the active set $\mathcal{A}^{(k+1)}$ still contains the constraint i. Otherwise, it releases the active inequality constraint and increases the dimension of the optimization problem in the next iteration.

A serious numerical problem can arise when some of the active constraints become (nearly) linearly dependent. PROC NLMIXED removes linearly dependent equality constraints before starting optimization. You can use the LCSINGULAR= option to specify a criterion r used in the update of the QR decomposition that determines whether an active constraint is linearly dependent relative to a set of other active constraints.

If the solution $\boldsymbol{\theta}^*$ is subjected to n_{act} linear equality or active inequality constraints, the QR decomposition of the $n \times n_{act}$ matrix $\hat{\mathbf{A}}'$ of the linear constraints is computed by $\hat{\mathbf{A}}' = \mathbf{QR}$, where $\mathbf{Q}$ is an $n \times n$ orthogonal matrix and $\mathbf{R}$ is an $n \times n_{act}$ upper triangular matrix. The n columns of matrix $\mathbf{Q}$ can be separated into two matrices, $\mathbf{Q} = [\mathbf{Y}, \mathbf{Z}]$, where $\mathbf{Y}$ contains the first n_{act} orthogonal columns of $\mathbf{Q}$ and $\mathbf{Z}$ contains the last $n - n_{act}$ orthogonal columns of $\mathbf{Q}$. The $n \times (n - n_{act})$ column-orthogonal matrix $\mathbf{Z}$ is also called the *null-space matrix* of the active linear constraints $\hat{\mathbf{A}}'$. The $n - n_{act}$ columns of the $n \times (n - n_{act})$ matrix $\mathbf{Z}$ form a basis orthogonal to the rows of the $n_{act} \times n$ matrix $\hat{\mathbf{A}}$.

At the end of the iterating, PROC NLMIXED computes the *projected gradient* $\mathbf{g}_Z$,

$$\mathbf{g}_Z = \mathbf{Z}'\mathbf{g}$$

In the case of boundary-constrained optimization, the elements of the projected gradient correspond to the gradient elements of the free parameters. A necessary condition for $\boldsymbol{\theta}^*$ to be a local minimum of the optimization problem is

$$\mathbf{g}_Z(\boldsymbol{\theta}^*) = \mathbf{Z}'\mathbf{g}(\boldsymbol{\theta}^*) = \mathbf{0}$$

The symmetric $n_{act} \times n_{act}$ matrix $\mathbf{G}_Z$,

$$\mathbf{G}_Z = \mathbf{Z}'\mathbf{G}\mathbf{Z}$$

is called a *projected Hessian matrix*. A second-order necessary condition for $\boldsymbol{\theta}^*$ to be a local minimizer requires that the projected Hessian matrix is positive semidefinite.

Those elements of the n_{act} vector of first-order estimates of *Lagrange multipliers*,

$$\lambda = (\hat{\mathbf{A}}\hat{\mathbf{A}}')^{-1}\hat{\mathbf{A}}\mathbf{Z}\mathbf{Z}'\mathbf{g}$$

that correspond to active inequality constraints indicate whether an improvement of the objective function can be obtained by releasing this active constraint. For minimization, a significant negative Lagrange multiplier indicates that a possible reduction of the objective function can be achieved by releasing this active linear constraint. The LCDEACT=r option specifies a threshold r for the Lagrange multiplier that determines whether an active inequality constraint remains active or can be deactivated. (In the case of boundary-constrained optimization, the Lagrange multipliers for active lower (upper) constraints are the negative (positive) gradient elements corresponding to the active parameters.)

Line-Search Methods

In each iteration k, the (dual) quasi-Newton, conjugate gradient, and Newton-Raphson minimization techniques use iterative line-search algorithms that try to optimize a linear, quadratic, or cubic approximation of f along a feasible descent search direction $\mathbf{s}^{(k)}$,

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \alpha^{(k)}\mathbf{s}^{(k)}, \quad \alpha^{(k)} > 0$$

by computing an approximately optimal scalar $\alpha^{(k)}$.

Therefore, a line-search algorithm is an iterative process that optimizes a nonlinear function $f(\alpha)$ of one parameter (α) within each iteration k of the optimization technique. Since the outside iteration process is based only on the approximation of the objective function, the inside iteration of the line-search algorithm does not have to be perfect. Usually, it is satisfactory that the choice of α significantly reduces (in a minimization) the objective function. Criteria often used for termination of line-search algorithms are the Goldstein conditions (see Fletcher 1987).

You can select various line-search algorithms by specifying the LINESEARCH= option. The line-search method LINESEARCH=2 seems to be superior when function evaluation consumes significantly less computation time than gradient evaluation. Therefore, LINESEARCH=2 is the default method for Newton-Raphson, (dual) quasi-Newton, and conjugate gradient optimizations.

You can modify the line-search methods LINESEARCH=2 and LINESEARCH=3 to be exact line searches by using the LSPRECISION= option and specifying the σ parameter described in Fletcher (1987). The line-search methods LINESEARCH=1, LINESEARCH=2, and LINESEARCH=3 satisfy the left-side and right-side Goldstein conditions (see Fletcher 1987). When derivatives are available, the line-search methods LINESEARCH=6, LINESEARCH=7, and LINESEARCH=8 try to satisfy the right-side Goldstein condition; if derivatives are not available, these line-search algorithms use only function calls.

Restricting the Step Length

Almost all line-search algorithms use iterative extrapolation techniques that can easily lead them to (feasible) points where the objective function f is no longer defined or is difficult to compute. Therefore, PROC NLMIXED provides options restricting the step length α or trust region radius Δ, especially during the first main iterations.

The inner product $\mathbf{g}'\mathbf{s}$ of the gradient $\mathbf{g}$ and the search direction $\mathbf{s}$ is the slope of $f(\alpha) = f(\boldsymbol{\theta} + \alpha\mathbf{s})$ along the search direction $\mathbf{s}$. The default starting value $\alpha^{(0)} = \alpha^{(k,0)}$ in each line-search algorithm ($\min_{\alpha>0} f(\boldsymbol{\theta} + \alpha\mathbf{s})$) during the main iteration k is computed in three steps:

1. The first step uses either the difference $df = |f^{(k)} - f^{(k-1)}|$ of the function values during the last two consecutive iterations or the final step-size value α^- of the last iteration $k-1$ to compute a first value of $\alpha_1^{(0)}$.

- If the DAMPSTEP option is not used,

$$\alpha_1^{(0)} = \begin{cases} step & \text{if } 0.1 \leq step \leq 10 \\ 10 & \text{if } step > 10 \\ 0.1 & \text{if } step < 0.1 \end{cases}$$

with

$$step = \begin{cases} df/|\mathbf{g}'\mathbf{s}| & \text{if } |\mathbf{g}'\mathbf{s}| \geq \epsilon \max(100df, 1) \\ 1 & \text{otherwise} \end{cases}$$

This value of $\alpha_1^{(0)}$ can be too large and can lead to a difficult or impossible function evaluation, especially for highly nonlinear functions such as the EXP function.

- If the DAMPSTEP=r option is used,

$$\alpha_1^{(0)} = \min(1, r\alpha^-)$$

The initial value for the new step length can be no larger than r times the final step length α^- of the former iteration. The default value is $r = 2$.

2. During the first five iterations, the second step enables you to reduce $\alpha_1^{(0)}$ to a smaller starting value $\alpha_2^{(0)}$ by using the INSTEP=r option:

$$\alpha_2^{(0)} = \min(\alpha_1^{(0)}, r)$$

After more than five iterations, $\alpha_2^{(0)}$ is set to $\alpha_1^{(0)}$.

3. The third step can further reduce the step length by

$$\alpha_3^{(0)} = \min(\alpha_2^{(0)}, \min(10, u))$$

where u is the maximum length of a step inside the feasible region.

The INSTEP=r option enables you to specify a smaller or larger radius Δ of the trust region used in the first iteration of the trust region and double-dogleg algorithms. The default initial trust region radius $\Delta^{(0)}$ is the length of the scaled gradient (Moré 1978). This step corresponds to the default radius factor of $r = 1$. In most practical applications of the TRUREG and DBLDOG algorithms, this choice is successful. However, for bad initial values and highly nonlinear objective functions (such as the EXP function), the default start radius can result in arithmetic overflows. If this happens, you can try decreasing values of INSTEP=r, $0 < r < 1$, until the iteration starts successfully. A small factor r also affects the trust region radius $\Delta^{(k+1)}$ of the next steps because the radius is changed in each iteration by a factor $0 < c \leq 4$, depending on the ratio ρ expressing the goodness of quadratic function approximation. Reducing the radius Δ corresponds to increasing the ridge parameter λ, producing smaller steps aimed more closely toward the (negative) gradient direction.

Computational Problems

Floating-Point Errors and Overflows

Numerical optimization of a numerically integrated function is a difficult task, and the computation of the objective function and its derivatives can lead to arithmetic exceptions and overflows. A typical cause of these problems is parameters with widely varying scales. If the scaling of your parameters varies by more than a few orders of magnitude, the numerical stability of the optimization problem can be seriously reduced and can result in computational difficulties. A simple remedy is to rescale each parameter so that its final estimated value has a magnitude near 1.

If parameter rescaling does not help, consider the following actions:

- Specify the ITDETAILS option in the PROC NLMIXED statement to obtain more detailed information about when and where the problem is occurring.
- Provide different initial values or try a grid search of values.
- Use boundary constraints to avoid the region where overflows can happen.
- Delete outlying observations or subjects from the input data, if this is reasonable.
- Change the algorithm (specified in programming statements) that computes the objective function.

The line-search algorithms that work with cubic extrapolation are especially sensitive to arithmetic overflows. If an overflow occurs during a line search, you can use the INSTEP= option to reduce the length of the first trial step during the first five iterations, or you can use the DAMPSTEP or MAXSTEP option to restrict the step length of the initial α in subsequent iterations. If an arithmetic overflow occurs in the first iteration of the trust region or double-dogleg algorithm, you can use the INSTEP= option to reduce the default trust region radius of the first iteration. You can also change the optimization technique or the line-search method.

Long Run Times

PROC NLMIXED can take a long time to run for problems involving complex models, many parameters, or large input data sets. Although the optimization techniques used by PROC NLMIXED are some of the best ones available, they are not guaranteed to converge quickly for all problems. Ill-posed or misspecified models can cause the algorithms to use more extensive calculations designed to achieve convergence, and this can result in longer run times. So first make sure that your model is specified correctly, that your parameters are scaled to be of the same order of magnitude, and that your data reasonably match the model you are contemplating.

If you are using the default adaptive Gaussian quadrature algorithm and no iteration history is printing at all, then PROC NLMIXED might be bogged down trying to determine the number of quadrature points at the first set of starting values. Specifying the QPOINTS= option will bypass

this stage and proceed directly to iterations; however, be aware that the likelihood approximation might not be accurate if there are too few quadrature points.

PROC NLMIXED can also have difficulty determining the number of quadrature points if the initial starting values are far from the optimum values. To obtain more accurate starting values for the model parameters, one easy method is to fit a model with no RANDOM statement. You can then use these estimates as starting values, although you will still need to specify values for the random-effects distribution. For normal-normal models, another strategy is to use METHOD=FIRO. If you can obtain estimates by using this approximate method, then they can be used as starting values for more accurate likelihood approximations.

If you are running PROC NLMIXED multiple times, you will probably want to include a statement like the following in your program:

```
ods output ParameterEstimates=pe;
```

This statement creates a SAS data set named PE upon completion of the run. In your next invocation of PROC NLMIXED, you can then specify

```
parms / data=pe;
```

to read in the previous estimates as starting values.

To speed general computations, you should double-check your programming statements to minimize the number of floating-point operations. Using auxiliary variables and factoring amenable expressions can be useful changes in this regard.

Problems Evaluating Code for Objective Function

The starting point $\boldsymbol{\theta}^{(0)}$ must be a point for which the programming statements can be evaluated. However, during optimization, the optimizer might iterate to a point $\boldsymbol{\theta}^{(k)}$ where the objective function or its derivatives cannot be evaluated. In some cases, the specification of boundary for parameters can avoid such situations. In many other cases, you can indicate that the point $\boldsymbol{\theta}^{(0)}$ is a bad point simply by returning an extremely large value for the objective function. In these cases, the optimization algorithm reduces the step length and stays closer to the point that has been evaluated successfully in the former iteration.

No Convergence

There are a number of things to try if the optimizer fails to converge.

- Change the initial values by using a grid search specification to obtain a set of good feasible starting values.
- Change or modify the update technique or the line-search algorithm.

  This method applies only to TECH=QUANEW and TECH=CONGRA. For example, if you use the default update formula and the default line-search algorithm, you can do the following:

  - change the update formula with the UPDATE= option
  - change the line-search algorithm with the LINESEARCH= option
  - specify a more precise line search with the LSPRECISION= option, if you use LINESEARCH=2 or LINESEARCH=3

- Change the optimization technique.

  For example, if you use the default option, TECH=QUANEW, you can try one of the second-derivative methods if your problem is small or the conjugate gradient method if it is large.

- Adjust finite-difference derivatives.

  The forward-difference derivatives specified with the FD= or FDHESSIAN= option might not be precise enough to satisfy strong gradient termination criteria. You might need to specify the more expensive central-difference formulas. The finite-difference intervals might be too small or too big, and the finite-difference derivatives might be erroneous.

- Double-check the data entry and program specification.

Convergence to Stationary Point

The gradient at a stationary point is the null vector, which always leads to a zero search direction. This point satisfies the first-order termination criterion. Search directions that are based on the gradient are zero, so the algorithm terminates. There are two ways to avoid this situation:

- Use the PARMS statement to specify a grid of feasible initial points.
- Use the OPTCHECK=*r* option to avoid terminating at the stationary point.

The signs of the eigenvalues of the (reduced) Hessian matrix contain the following information regarding a stationary point:

- If all of the eigenvalues are positive, the Hessian matrix is positive definite, and the point is a minimum point.
- If some of the eigenvalues are positive and all remaining eigenvalues are zero, the Hessian matrix is positive semidefinite, and the point is a minimum or saddle point.
- If all of the eigenvalues are negative, the Hessian matrix is negative definite, and the point is a maximum point.
- If some of the eigenvalues are negative and all remaining eigenvalues are zero, the Hessian matrix is negative semidefinite, and the point is a maximum or saddle point.
- If all of the eigenvalues are zero, the point can be a minimum, maximum, or saddle point.

Precision of Solution

In some applications, PROC NLMIXED can result in parameter values that are not precise enough. Usually, this means that the procedure terminated at a point too far from the optimal point. The termination criteria define the size of the termination region around the optimal point. Any point inside this region can be accepted for terminating the optimization process. The default values of the termination criteria are set to satisfy a reasonable compromise between the computational effort (computer time) and the precision of the computed estimates for the most common applications. However, there are a number of circumstances in which the default values of the termination criteria specify a region that is either too large or too small.

If the termination region is too large, then it can contain points with low precision. In such cases, you should determine which termination criterion stopped the optimization process. In many applications, you can obtain a solution with higher precision simply by using the old parameter estimates as starting values in a subsequent run in which you specify a smaller value for the termination criterion that was satisfied at the former run.

If the termination region is too small, the optimization process might take longer to find a point inside such a region, or it might not even find such a point due to rounding errors in function values and derivatives. This can easily happen in applications in which finite-difference approximations of derivatives are used and the GCONV and ABSGCONV termination criteria are too small to respect rounding errors in the gradient values.

Covariance Matrix

The estimated covariance matrix of the parameter estimates is computed as the inverse Hessian matrix, and for unconstrained problems it should be positive definite. If the final parameter estimates are subjected to $n_{act} > 0$ active linear inequality constraints, the formulas of the covariance matrices are modified similar to Gallant (1987) and Cramer (1986, p. 38) and additionally generalized for applications with singular matrices.

There are several steps available that enable you to tune the rank calculations of the covariance matrix.

1. You can use the ASINGULAR=, MSINGULAR=, and VSINGULAR= options to set three singularity criteria for the inversion of the Hessian matrix **H**. The singularity criterion used for the inversion is

$$|d_{j,j}| \leq \max(\text{ASING}, \text{VSING} * |H_{j,j}|, \text{MSING} * max(|H_{1,1}|, \ldots, |H_{n,n}|))$$

   where $d_{j,j}$ is the diagonal pivot of the matrix **H**, and ASING, VSING, and MSING are the specified values of the ASINGULAR=, VSINGULAR=, and MSINGULAR= options, respectively. The default values are as follows:

   - ASING: the square root of the smallest positive double-precision value
   - MSING: 1E−12 if you do not specify the SINGHESS= option and $\max(10\epsilon, 1\text{E} - 4 \times$ SINGHESS) otherwise, where ϵ is the machine precision

- VSING: 1E−8 if you do not specify the SINGHESS= option and the value of SINGHESS otherwise

Note that, in many cases, a normalized matrix $\mathbf{D}^{-1}\mathbf{A}\mathbf{D}^{-1}$ is decomposed, and the singularity criteria are modified correspondingly.

2. If the matrix $\mathbf{H}$ is found to be singular in the first step, a generalized inverse is computed. Depending on the G4= option, either a generalized inverse satisfying all four Moore-Penrose conditions is computed (a g_4-inverse) or a generalized inverse satisfying only two Moore-Penrose conditions is computed (a g_2-inverse, Pringle and Rayner, 1971). If the number of parameters n of the application is less than or equal to G4=i, a g_4-inverse is computed; otherwise, only a g_2-inverse is computed. The g_4-inverse is computed by the (computationally very expensive but numerically stable) eigenvalue decomposition, and the g_2-inverse is computed by Gauss transformation. The g_4-inverse is computed using the eigenvalue decomposition $\mathbf{A} = \mathbf{Z}\mathbf{\Lambda}\mathbf{Z}'$, where $\mathbf{Z}$ is the orthogonal matrix of eigenvectors and $\mathbf{\Lambda}$ is the diagonal matrix of eigenvalues, $\mathbf{\Lambda} = diag(\lambda_1, \ldots, \lambda_n)$. The g_4-inverse of $\mathbf{H}$ is set to

$$\mathbf{A}^- = \mathbf{Z}\mathbf{\Lambda}^-\mathbf{Z}'$$

where the diagonal matrix $\mathbf{\Lambda}^- = diag(\lambda_1^-, \ldots, \lambda_n^-)$ is defined using the COVSING= option:

$$\lambda_i^- = \begin{cases} 1/\lambda_i & \text{if } |\lambda_i| > \text{COVSING} \\ 0 & \text{if } |\lambda_i| \leq \text{COVSING} \end{cases}$$

If you do not specify the COVSING= option, the nr smallest eigenvalues are set to zero, where nr is the number of rank deficiencies found in the first step.

For optimization techniques that do not use second-order derivatives, the covariance matrix is computed using finite-difference approximations of the derivatives.

Prediction

The nonlinear mixed model is a useful tool for statistical prediction. Assuming a prediction is to be made regarding the ith subject, suppose that $f(\boldsymbol{\theta}, \mathbf{u}_i)$ is a differentiable function predicting some quantity of interest. Recall that $\boldsymbol{\theta}$ denotes the vector of unknown parameters and $\mathbf{u}_i$ denotes the vector of random effects for the ith subject. A natural point prediction is $f(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{u}}_i)$, where $\widehat{\boldsymbol{\theta}}$ is the maximum likelihood estimate of $\boldsymbol{\theta}$ and $\widehat{\mathbf{u}}_i$ is the empirical Bayes estimate of $\mathbf{u}_i$ described previously in the section "Integral Approximations" on page 4382.

An approximate prediction variance matrix for $(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{u}}_i)$ is

$$\mathbf{P} = \begin{bmatrix} \widehat{\mathbf{H}}^{-1} & \widehat{\mathbf{H}}^{-1}\left(\frac{\partial \widehat{\mathbf{u}}_i}{\partial \boldsymbol{\theta}}\right)' \\ \left(\frac{\partial \widehat{\mathbf{u}}_i}{\partial \boldsymbol{\theta}}\right)\widehat{\mathbf{H}}^{-1} & \widehat{\mathbf{\Gamma}}^{-1} + \left(\frac{\partial \widehat{\mathbf{u}}_i}{\partial \boldsymbol{\theta}}\right)\widehat{\mathbf{H}}^{-1}\left(\frac{\partial \widehat{\mathbf{u}}_i}{\partial \boldsymbol{\theta}}\right)' \end{bmatrix}$$

where $\widehat{\mathbf{H}}$ is the approximate Hessian matrix from the optimization for $\widehat{\boldsymbol{\theta}}$, $\widehat{\mathbf{\Gamma}}$ is the approximate Hessian matrix from the optimization for $\widehat{\mathbf{u}}_i$, and $(\partial \widehat{\mathbf{u}}_i / \partial \boldsymbol{\theta})$ is the derivative of $\widehat{\mathbf{u}}_i$ with respect to

$\boldsymbol{\theta}$, evaluated at $(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{u}}_i)$. The approximate variance matrix for $\widehat{\boldsymbol{\theta}}$ is the standard one discussed in the previous section, and that for $\widehat{\mathbf{u}}_i$ is an approximation to the conditional mean squared error of prediction described by Booth and Hobert (1998).

The prediction variance for a general scalar function $f(\boldsymbol{\theta}, \mathbf{u}_i)$ is defined as the expected squared difference $E[f(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{u}}_i) - f(\boldsymbol{\theta}, \mathbf{u}_i)]^2$. PROC NLMIXED computes an approximation to it as follows. The derivative of $f(\boldsymbol{\theta}, \mathbf{u}_i)$ is computed with respect to each element of $(\boldsymbol{\theta}, \mathbf{u}_i)$ and evaluated at $(\widehat{\boldsymbol{\theta}}, \widehat{\mathbf{u}}_i)$. If $\mathbf{a}_i$ is the resulting vector, then the approximate prediction variance is $\mathbf{a}_i' \mathbf{P} \mathbf{a}_i$. This approximation is known as the delta method (Billingsley 1986, Cox 1998).

Computational Resources

Since nonlinear optimization is an iterative process that depends on many factors, it is difficult to estimate how much computer time is necessary to find an optimal solution satisfying one of the termination criteria. You can use the MAXTIME=, MAXITER=, and MAXFUNC= options to restrict the amount of CPU time, the number of iterations, and the number of function calls in a single run of PROC NLMIXED.

In each iteration k, the NRRIDG technique uses a symmetric Householder transformation to decompose the $n \times n$ Hessian matrix $\mathbf{H}$,

$$\mathbf{H} = \mathbf{V}'\mathbf{T}\mathbf{V}, \qquad \mathbf{V}\text{: orthogonal}, \quad \mathbf{T}\text{: tridiagonal}$$

to compute the (Newton) search direction $\mathbf{s}$,

$$\mathbf{s}^{(k)} = -[\mathbf{H}^{(k)}]^{-1}\mathbf{g}^{(k)} \qquad k = 1, 2, 3, \ldots$$

The TRUREG and NEWRAP techniques use the Cholesky decomposition to solve the same linear system while computing the search direction. The QUANEW, DBLDOG, CONGRA, and NMSIMP techniques do not need to invert or decompose a Hessian matrix; thus, they require less computational resources than the other techniques.

The larger the problem, the more time is needed to compute function values and derivatives. Therefore, you might want to compare optimization techniques by counting and comparing the respective numbers of function, gradient, and Hessian evaluations.

Finite-difference approximations of the derivatives are expensive because they require additional function or gradient calls:

- forward-difference formulas
  - For first-order derivatives, n additional function calls are required.
  - For second-order derivatives based on function calls only, for a dense Hessian, $n + n^2/2$ additional function calls are required.
  - For second-order derivatives based on gradient calls, n additional gradient calls are required.

- central-difference formulas
  - For first-order derivatives, $2n$ additional function calls are required.
  - For second-order derivatives based on function calls only, for a dense Hessian, $2n+2n^2$ additional function calls are required.
  - For second-order derivatives based on gradient calls, $2n$ additional gradient calls are required.

Many applications need considerably more time for computing second-order derivatives (Hessian matrix) than for computing first-order derivatives (gradient). In such cases, a dual quasi-Newton technique is recommended, which does not require second-order derivatives.

Displayed Output

This section describes the displayed output from PROC NLMIXED. See the section "ODS Table Names" on page 4408 for details about how this output interfaces with the Output Delivery System.

Specifications

The NLMIXED procedure first displays the "Specifications" table, listing basic information about the nonlinear mixed model that you have specified. It includes the principal variables and estimation methods.

Dimensions

The "Dimensions" table lists counts of important quantities in your nonlinear mixed model, including the number of observations, subjects, parameters, and quadrature points.

Parameters

The "Parameters" table displays the information you provided with the PARMS statement and the value of the negative log-likelihood function evaluated at the starting values.

Starting Gradient and Hessian

The START option in the PROC NLMIXED statement displays the gradient of the negative log-likelihood function at the starting values of the parameters. If you also specify the HESS option, then the starting Hessian is displayed as well.

Iterations

The iteration history consists of one line of output for each iteration in the optimization process. The iteration history is displayed by default because it is important that you check for possible convergence problems. The default iteration history includes the following variables:

- Iter, the iteration number
- Calls, the number of function calls
- NegLogLike, the value of the objective function
- Diff, the difference between adjacent function values
- MaxGrad, the maximum of the absolute (projected) gradient components (except NMSIMP)
- Slope, the slope $\mathbf{g}'\mathbf{s}$ of the search direction $\mathbf{s}$ at the current parameter iterate $\boldsymbol{\theta}^{(k)}$ (QUANEW only)
- Rho, the ratio between the achieved and predicted values of Diff (NRRIDG only)
- Radius, the radius of the trust region (TRUREG only)
- StdDev, the standard deviation of the simplex values (NMSIMP only)
- Delta, the vertex length of the simplex (NMSIMP only)
- Size, the size of the simplex (NMSIMP only)

For the QUANEW method, the value of Slope should be significantly negative. Otherwise, the line-search algorithm has difficulty reducing the function value sufficiently. If this difficulty is encountered, an asterisk (*) appears after the iteration number. If there is a tilde (~) after the iteration number, the BFGS update is skipped, and very high values of the Lagrange function are produced. A backslash (\ ) after the iteration number indicates that Powell's correction for the BFGS update is used.

For methods using second derivatives, an asterisk (*) after the iteration number means that the computed Hessian approximation was singular and had to be ridged with a positive value.

For the NMSIMP method, only one line is displayed for several internal iterations. This technique skips the output for some iterations because some of the termination tests (StdDev and Size) are rather time-consuming compared to the simplex operations, and they are performed only every five simplex operations.

The ITDETAILS option in the PROC NLMIXED statement provides a more detailed iteration history. Besides listing the current values of the parameters and their gradients, the ITDETAILS option provides the following values in addition to the default output:

- Restart, the number of iteration restarts
- Active, the number of active constraints
- Lambda, the value of the Lagrange multiplier (TRUREG and DBLDOG only)
- Ridge, the ridge value (NRRIDG only)
- Alpha, the line-search step size (QUANEW only)

An apostrophe (') trailing the number of active constraints indicates that at least one of the active constraints was released from the active set due to a significant Lagrange multiplier.

Convergence Status

The "Convergence Status" table contains a status message describing the reason for termination of the optimization. For ODS purposes, the name of this table is "ConvergenceStatus," and you can query the nonprinting numeric variable Status to check for a successful optimization. This is useful in batch processing, or when processing BY groups, for example, in simulations. Successful convergence is indicated by Status= 0.

Fitting Information

The "Fitting Information" table lists the final minimized value of −2 times the log likelihood as well as the information criteria of Akaike (AIC) and Schwarz (BIC), as well as a finite-sample corrected version of AIC (AICC). The criteria are computed as follows:

$$\begin{aligned} AIC &= 2f(\widehat{\boldsymbol{\theta}}) + 2p \\ AICC &= 2f(\widehat{\boldsymbol{\theta}}) + 2pn/(n-p-1) \\ BIC &= 2f(\widehat{\boldsymbol{\theta}}) + p\log(s) \end{aligned}$$

where $f()$ is the negative of the marginal log-likelihood function, $\widehat{\boldsymbol{\theta}}$ is the vector of parameter estimates, p is the number of parameters, n is the number of observations, and s is the number of subjects. Refer to Hurvich and Tsai (1989) and Burnham and Anderson (1998) for additional details.

Parameter Estimates

The "Parameter Estimates" table lists the estimates of the parameter values after successful convergence of the optimization problem or the final values of the parameters under nonconvergence. If the problem did converge, standard errors are computed from the final Hessian matrix. The ratio of the estimate with its standard error produces a t value, with approximate degrees of freedom computed as the number of subjects minus the number of random effects. A p-value and confidence limits based on this t distribution are also provided. Finally, the gradient of the negative log-likelihood function is displayed for each parameter, and you should verify that they each are sufficiently small for nonconstrained parameters.

Covariance and Correlation Matrices

Following standard maximum likelihood theory (for example, Serfling 1980), the asymptotic variance-covariance matrix of the parameter estimates equals the inverse of the Hessian matrix. You can display this matrix with the COV option in the PROC NLMIXED statement. The corresponding correlation form is available with the CORR option.

Additional Estimates

The "Additional Estimates" table displays the results of all ESTIMATE statements that you specify, with the same columns as the "Parameter Estimates" table. The ECOV and ECORR options in the PROC NLMIXED statement produce tables displaying the approximate covariance and correlation

matrices of the additional estimates. They are computed using the delta method (Billingsley 1986; Cox 1998). The EDER option in the PROC NLMIXED statement produces a table that displays the derivatives of the additional estimates with respect to the model parameters evaluated at their final estimated values.

ODS Table Names

PROC NLMIXED assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 61.2. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 61.2 ODS Tables Produced by PROC NLMIXED

ODS Table Name	Description	Statement or Option
AdditionalEstimates	Results from ESTIMATE statements	ESTIMATE
Contrasts	Results from CONTRAST statements	CONTRAST
ConvergenceStatus	Convergence status	default
CorrMatAddEst	Correlation matrix of additional estimates	ECORR
CorrMatParmEst	Correlation matrix of parameter estimates	CORR
CovMatAddEst	Covariance matrix of additional estimates	ECOV
CovMatParmEst	Covariance matrix of parameter estimates	COV
DerAddEst	Derivatives of additional estimates	EDER
Dimensions	Dimensions of the problem	default
FitStatistics	Fit statistics	default
Hessian	Second derivative matrix	HESS
IterHistory	Iteration history	default
Parameters	Parameters	default
ParameterEstimates	Parameter estimates	default
Specifications	Model specifications	default
StartingHessian	Starting Hessian matrix	START HESS
StartingValues	Starting values and gradient	START

Examples: NLMIXED Procedure

Example 61.1: One-Compartment Model with Pharmacokinetic Data

A popular application of nonlinear mixed models is in the field of pharmacokinetics, which studies how a drug disperses through a living individual. This example considers the theophylline data from Pinheiro and Bates (1995). Serum concentrations of the drug theophylline are measured in 12 subjects over a 25-hour period after oral administration. The data are as follows.

```
data theoph;
  input subject time conc dose wt;
  datalines;
 1  0.00  0.74 4.02 79.6
 1  0.25  2.84 4.02 79.6
 1  0.57  6.57 4.02 79.6
 1  1.12 10.50 4.02 79.6
 1  2.02  9.66 4.02 79.6
 1  3.82  8.58 4.02 79.6
 1  5.10  8.36 4.02 79.6
 1  7.03  7.47 4.02 79.6
 1  9.05  6.89 4.02 79.6
 1 12.12  5.94 4.02 79.6
 1 24.37  3.28 4.02 79.6
 2  0.00  0.00 4.40 72.4
 2  0.27  1.72 4.40 72.4
 2  0.52  7.91 4.40 72.4
 2  1.00  8.31 4.40 72.4
 2  1.92  8.33 4.40 72.4
 2  3.50  6.85 4.40 72.4
 2  5.02  6.08 4.40 72.4
 2  7.03  5.40 4.40 72.4
 2  9.00  4.55 4.40 72.4
 2 12.00  3.01 4.40 72.4
 2 24.30  0.90 4.40 72.4
 3  0.00  0.00 4.53 70.5
 3  0.27  4.40 4.53 70.5
 3  0.58  6.90 4.53 70.5
 3  1.02  8.20 4.53 70.5
 3  2.02  7.80 4.53 70.5
 3  3.62  7.50 4.53 70.5
 3  5.08  6.20 4.53 70.5
 3  7.07  5.30 4.53 70.5
 3  9.00  4.90 4.53 70.5
 3 12.15  3.70 4.53 70.5
 3 24.17  1.05 4.53 70.5
 4  0.00  0.00 4.40 72.7
 4  0.35  1.89 4.40 72.7
 4  0.60  4.60 4.40 72.7
```

```
4  1.07  8.60 4.40 72.7
4  2.13  8.38 4.40 72.7
4  3.50  7.54 4.40 72.7
4  5.02  6.88 4.40 72.7
4  7.02  5.78 4.40 72.7
4  9.02  5.33 4.40 72.7
4 11.98  4.19 4.40 72.7
4 24.65  1.15 4.40 72.7
5  0.00  0.00 5.86 54.6
5  0.30  2.02 5.86 54.6
5  0.52  5.63 5.86 54.6
5  1.00 11.40 5.86 54.6
5  2.02  9.33 5.86 54.6
5  3.50  8.74 5.86 54.6
5  5.02  7.56 5.86 54.6
5  7.02  7.09 5.86 54.6
5  9.10  5.90 5.86 54.6
5 12.00  4.37 5.86 54.6
5 24.35  1.57 5.86 54.6
6  0.00  0.00 4.00 80.0
6  0.27  1.29 4.00 80.0
6  0.58  3.08 4.00 80.0
6  1.15  6.44 4.00 80.0
6  2.03  6.32 4.00 80.0
6  3.57  5.53 4.00 80.0
6  5.00  4.94 4.00 80.0
6  7.00  4.02 4.00 80.0
6  9.22  3.46 4.00 80.0
6 12.10  2.78 4.00 80.0
6 23.85  0.92 4.00 80.0
7  0.00  0.15 4.95 64.6
7  0.25  0.85 4.95 64.6
7  0.50  2.35 4.95 64.6
7  1.02  5.02 4.95 64.6
7  2.02  6.58 4.95 64.6
7  3.48  7.09 4.95 64.6
7  5.00  6.66 4.95 64.6
7  6.98  5.25 4.95 64.6
7  9.00  4.39 4.95 64.6
7 12.05  3.53 4.95 64.6
7 24.22  1.15 4.95 64.6
8  0.00  0.00 4.53 70.5
8  0.25  3.05 4.53 70.5
8  0.52  3.05 4.53 70.5
8  0.98  7.31 4.53 70.5
8  2.02  7.56 4.53 70.5
8  3.53  6.59 4.53 70.5
8  5.05  5.88 4.53 70.5
8  7.15  4.73 4.53 70.5
8  9.07  4.57 4.53 70.5
8 12.10  3.00 4.53 70.5
8 24.12  1.25 4.53 70.5
9  0.00  0.00 3.10 86.4
9  0.30  7.37 3.10 86.4
```

```
 9  0.63  9.03 3.10 86.4
 9  1.05  7.14 3.10 86.4
 9  2.02  6.33 3.10 86.4
 9  3.53  5.66 3.10 86.4
 9  5.02  5.67 3.10 86.4
 9  7.17  4.24 3.10 86.4
 9  8.80  4.11 3.10 86.4
 9 11.60  3.16 3.10 86.4
 9 24.43  1.12 3.10 86.4
10  0.00  0.24 5.50 58.2
10  0.37  2.89 5.50 58.2
10  0.77  5.22 5.50 58.2
10  1.02  6.41 5.50 58.2
10  2.05  7.83 5.50 58.2
10  3.55 10.21 5.50 58.2
10  5.05  9.18 5.50 58.2
10  7.08  8.02 5.50 58.2
10  9.38  7.14 5.50 58.2
10 12.10  5.68 5.50 58.2
10 23.70  2.42 5.50 58.2
11  0.00  0.00 4.92 65.0
11  0.25  4.86 4.92 65.0
11  0.50  7.24 4.92 65.0
11  0.98  8.00 4.92 65.0
11  1.98  6.81 4.92 65.0
11  3.60  5.87 4.92 65.0
11  5.02  5.22 4.92 65.0
11  7.03  4.45 4.92 65.0
11  9.03  3.62 4.92 65.0
11 12.12  2.69 4.92 65.0
11 24.08  0.86 4.92 65.0
12  0.00  0.00 5.30 60.5
12  0.25  1.25 5.30 60.5
12  0.50  3.96 5.30 60.5
12  1.00  7.82 5.30 60.5
12  2.00  9.72 5.30 60.5
12  3.52  9.75 5.30 60.5
12  5.07  8.57 5.30 60.5
12  7.07  6.59 5.30 60.5
12  9.03  6.11 5.30 60.5
12 12.05  4.57 5.30 60.5
12 24.15  1.17 5.30 60.5
;
```

Pinheiro and Bates (1995) consider the following first-order compartment model for these data:

$$C_{it} = \frac{D k_{e_i} k_{a_i}}{Cl_i (k_{a_i} - k_{e_i})} [\exp(-k_{e_i} t) - \exp(-k_{a_i} t)] + e_{it}$$

where C_{it} is the observed concentration of the ith subject at time t, D is the dose of theophylline, k_{e_i} is the elimination rate constant for subject i, k_{a_i} is the absorption rate constant for subject i, Cl_i is the clearance for subject i, and e_{it} are normal errors. To allow for random variability between

subjects, they assume

$$Cl_i = \exp(\beta_1 + b_{i1})$$
$$k_{a_i} = \exp(\beta_2 + b_{i2})$$
$$k_{e_i} = \exp(\beta_3)$$

where the βs denote fixed-effects parameters and the b_is denote random-effects parameters with an unknown covariance matrix.

The PROC NLMIXED statements to fit this model are as follows:

```
proc nlmixed data=theoph;
   parms beta1=-3.22 beta2=0.47 beta3=-2.45
         s2b1 =0.03  cb12 =0    s2b2 =0.4 s2=0.5;
   cl   = exp(beta1 + b1);
   ka   = exp(beta2 + b2);
   ke   = exp(beta3);
   pred = dose*ke*ka*(exp(-ke*time)-exp(-ka*time))/cl/(ka-ke);
   model conc ~ normal(pred,s2);
   random b1 b2 ~ normal([0,0],[s2b1,cb12,s2b2]) subject=subject;
run;
```

The PARMS statement specifies starting values for the three βs and four variance-covariance parameters. The clearance and rate constants are defined using SAS programming statements, and the conditional model for the data is defined to be normal with mean pred and variance s2. The two random effects are b1 and b2, and their joint distribution is defined in the RANDOM statement. Brackets are used in defining their mean vector (two zeros) and the lower triangle of their variance-covariance matrix (a general 2×2 matrix). The SUBJECT= variable is subject.

The results from this analysis are as follows.

Output 61.1.1 Model Specification for One-Compartment Model

The NLMIXED Procedure

Specifications

Data Set	WORK.THEOPH
Dependent Variable	conc
Distribution for Dependent Variable	Normal
Random Effects	b1 b2
Distribution for Random Effects	Normal
Subject Variable	subject
Optimization Technique	Dual Quasi-Newton
Integration Method	Adaptive Gaussian Quadrature

The "Specifications" table lists the setup of the model (Output 61.1.1). The "Dimensions" table indicates that there are 132 observations, 12 subjects, and 7 parameters. PROC NLMIXED selects 5 quadrature points for each random effect, producing a total grid of 25 points over which quadrature is performed (Output 61.1.2).

Output 61.1.2 Dimensions Table for One-Compartment Model

Dimensions

Observations Used	132
Observations Not Used	0
Total Observations	132
Subjects	12
Max Obs Per Subject	11
Parameters	7
Quadrature Points	5

The "Parameters" table lists the 7 parameters, their starting values, and the initial evaluation of the negative log likelihood using adaptive Gaussian quadrature (Output 61.1.3). The "Iteration History" table indicates that 10 steps are required for the dual quasi-Newton algorithm to achieve convergence.

Output 61.1.3 Starting Values and Iteration History

Parameters

beta1	beta2	beta3	s2b1	cb12	s2b2	s2	NegLogLike
-3.22	0.47	-2.45	0.03	0	0.4	0.5	177.789945

Iteration History

Iter	Calls	NegLogLike	Diff	MaxGrad	Slope
1	5	177.776248	0.013697	2.873367	-63.0744
2	8	177.7643	0.011948	1.698144	-4.75239
3	10	177.757264	0.007036	1.297439	-1.97311
4	12	177.755688	0.001576	1.441408	-0.49772
5	14	177.7467	0.008988	1.132279	-0.8223
6	17	177.746401	0.000299	0.831293	-0.00244
7	19	177.746318	0.000083	0.724198	-0.00789
8	21	177.74574	0.000578	0.180018	-0.00583
9	23	177.745736	3.88E-6	0.017958	-8.25E-6
10	25	177.745736	3.222E-8	0.000143	-6.51E-8

NOTE: GCONV convergence criterion satisfied.

Output 61.1.4 Fit Statistics for One-Compartment Model

Fit Statistics

-2 Log Likelihood	355.5
AIC (smaller is better)	369.5
AICC (smaller is better)	370.4
BIC (smaller is better)	372.9

The "Fit Statistics" table lists the final optimized values of the log-likelihood function and information criteria in the "smaller is better" form (Output 61.1.4).

Output 61.1.5 Parameter Estimates for One-Compartment Model

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower
beta1	-3.2268	0.05950	10	-54.23	<.0001	0.05	-3.3594
beta2	0.4806	0.1989	10	2.42	0.0363	0.05	0.03745
beta3	-2.4592	0.05126	10	-47.97	<.0001	0.05	-2.5734
s2b1	0.02803	0.01221	10	2.30	0.0445	0.05	0.000833
cb12	-0.00127	0.03404	10	-0.04	0.9710	0.05	-0.07712
s2b2	0.4331	0.2005	10	2.16	0.0560	0.05	-0.01353
s2	0.5016	0.06837	10	7.34	<.0001	0.05	0.3493

Parameter Estimates

Parameter	Upper	Gradient
beta1	-3.0942	-0.00009
beta2	0.9238	3.645E-7
beta3	-2.3449	0.000039
s2b1	0.05523	-0.00014
cb12	0.07458	-0.00007
s2b2	0.8798	-6.98E-6
s2	0.6540	6.133E-6

The "Parameter Estimates" table contains the maximum likelihood estimates of the parameters (Output 61.1.5). Both s2b1 and s2b2 are marginally significant, indicating between-subject variability in the clearances and absorption rate constants, respectively. There does not appear to be a significant covariance between them, as seen by the estimate of cb12.

The estimates of β_1, β_2, and β_3 are close to the adaptive quadrature estimates listed in Table 3 of Pinheiro and Bates (1995). However, Pinheiro and Bates use a Cholesky-root parameterization for the random-effects variance matrix and a logarithmic parameterization for the residual variance. The PROC NLMIXED statements using their parameterization are as follows, and results are similar.

```
proc nlmixed data=theoph;
   parms l11=-1.5 l2=0 l13=-0.1 beta1=-3 beta2=0.5 beta3=-2.5
      ls2=-0.7;
   s2   = exp(ls2);
   l1   = exp(l11);
   l3   = exp(l13);
   s2b1 = l1*l1*s2;
   cb12 = l2*l1*s2;
   s2b2 = (l2*l2 + l3*l3)*s2;
   cl   = exp(beta1 + b1);
   ka   = exp(beta2 + b2);
   ke   = exp(beta3);
   pred = dose*ke*ka*(exp(-ke*time)-exp(-ka*time))/cl/(ka-ke);
   model conc   ~ normal(pred,s2);
   random b1 b2 ~ normal([0,0],[s2b1,cb12,s2b2]) subject=subject;
run;
```

Example 61.2: Probit-Normal Model with Binomial Data

For this example, consider the data from Weil (1970), also studied by Williams (1975), Ochi and Prentice (1984), and McCulloch (1994). In this experiment 16 pregnant rats receive a control diet and 16 receive a chemically treated diet, and the litter size for each rat is recorded after 4 and 21 days. The SAS data set follows:

```
data rats;
  input trt$ m x;
  if (trt='c') then do;
     x1 = 1;
     x2 = 0;
  end; else do;
     x1 = 0;
     x2 = 1;
  end;
  litter = _n_;
  datalines;
c 13 13
c 12 12
c  9  9
c  9  9
c  8  8
c  8  8
c 13 12
c 12 11
c 10  9
c 10  9
c  9  8
c 13 11
c  5  4
c  7  5
c 10  7
c 10  7
t 12 12
t 11 11
t 10 10
t  9  9
t 11 10
t 10  9
t 10  9
t  9  8
t  9  8
t  5  4
t  9  7
t  7  4
t 10  5
t  6  3
t 10  3
t  7  0
;
```

Here, m represents the size of the litter after 4 days, and x represents the size of the litter after 21 days. Also, indicator variables x1 and x2 are constructed for the two treatment levels.

Following McCulloch (1994), assume a latent survival model of the form

$$y_{ijk} = t_i + \alpha_{ij} + e_{ijk}$$

where i indexes treatment, j indexes litter, and k indexes newborn rats within a litter. The t_i represent treatment means, the α_{ij} represent random litter effects assumed to be iid $N(0, s_i^2)$, and the e_{ijk} represent iid residual errors, all on the latent scale.

Instead of observing the survival times y_{ijk}, assume that only the binary variable indicating whether y_{ijk} exceeds 0 is observed. If x_{ij} denotes the sum of these binary variables for the ith treatment and the jth litter, then the preceding assumptions lead to the following generalized linear mixed model:

$$x_{ij}|\alpha_{ij} \sim \text{Binomial}(m_{ij}, p_{ij})$$

where m_{ij} is the size of each litter after 4 days and

$$p_{ij} = \Phi(t_i + \alpha_{ij})$$

The PROC NLMIXED statements to fit this model are as follows:

```
proc nlmixed data=rats;
   parms t1=1 t2=1 s1=.05 s2=1;
   eta = x1*t1 + x2*t2 + alpha;
   p   = probnorm(eta);
   model x ~ binomial(m,p);
   random alpha ~ normal(0,x1*s1*s1+x2*s2*s2) subject=litter;
   estimate 'gamma2' t2/sqrt(1+s2*s2);
   predict p out=p;
run;
```

As in Example 61.1, the PROC NLMIXED statement invokes the procedure and the PARMS statement defines the parameters. The parameters for this example are the two treatment means, t1 and t2, and the two random-effect standard deviations, s1 and s2.

The indicator variables x1 and x2 are used in the program to assign the proper mean to each observation in the input data set as well as the proper variance to the random effects. Note that programming expressions are permitted inside the distributional specifications, as illustrated by the random-effects variance specified here.

The ESTIMATE statement requests an estimate of $\gamma_2 = t_2/\sqrt{1+s_2^2}$, which is a location-scale parameter from Ochi and Prentice (1984).

The PREDICT statement constructs predictions for each observation in the input data set. For this example, predictions of p and approximate standard errors of prediction are output to a SAS data set named P. These predictions are functions of the parameter estimates and the empirical Bayes estimates of the random effects α_i.

The output for this model is as follows.

Output 61.2.1 Specifications, Dimensions, and Starting Values

The NLMIXED Procedure

Specifications

Data Set	WORK.RATS
Dependent Variable	x
Distribution for Dependent Variable	Binomial
Random Effects	alpha
Distribution for Random Effects	Normal
Subject Variable	litter
Optimization Technique	Dual Quasi-Newton
Integration Method	Adaptive Gaussian Quadrature

Dimensions

Observations Used	32
Observations Not Used	0
Total Observations	32
Subjects	32
Max Obs Per Subject	1
Parameters	4
Quadrature Points	7

Parameters

t1	t2	s1	s2	NegLogLike
1	1	0.05	1	54.9362323

The "Specifications" table provides basic information about this nonlinear mixed model (Output 61.2.1). The "Dimensions" table provides counts of various variables. Note that each observation in the data comprises a separate subject. Using the starting values in the "Parameters" table, PROC NLMIXED determines that the log-likelihood function can be approximated with sufficient accuracy with a seven-point quadrature rule.

Output 61.2.2 Iteration History for Probit-Normal Model

Iteration History

Iter	Calls	NegLogLike	Diff	MaxGrad	Slope
1	2	53.9933934	0.942839	11.03261	-81.9428
2	3	52.875353	1.11804	2.148952	-2.86277
3	5	52.6350386	0.240314	0.329957	-1.05049
4	6	52.6319939	0.003045	0.122926	-0.00672
5	8	52.6313583	0.000636	0.028246	-0.00352
6	11	52.6313174	0.000041	0.013551	-0.00023
7	13	52.6313115	5.839E-6	0.000603	-0.00001
8	15	52.6313115	9.45E-9	0.000022	-1.68E-8

NOTE: GCONV convergence criterion satisfied.

The "Iteration History" table indicates successful convergence in 8 iterations (Output 61.2.2).

Output 61.2.3 Fit Statistics for Probit-Normal Model

Fit Statistics

-2 Log Likelihood	105.3
AIC (smaller is better)	113.3
AICC (smaller is better)	114.7
BIC (smaller is better)	119.1

The "Fit Statistics" table lists useful statistics based on the maximized value of the log likelihood (Output 61.2.3).

Output 61.2.4 Parameter Estimates for Probit-Normal Model

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower
t1	1.3063	0.1685	31	7.75	<.0001	0.05	0.9626
t2	0.9475	0.3055	31	3.10	0.0041	0.05	0.3244
s1	0.2403	0.3015	31	0.80	0.4315	0.05	-0.3746
s2	1.0292	0.2988	31	3.44	0.0017	0.05	0.4198

Parameter Estimates

Parameter	Upper	Gradient
t1	1.6499	-0.00002
t2	1.5705	9.283E-6
s1	0.8552	0.000014
s2	1.6385	-3.16E-6

The "Parameter Estimates" table indicates significance of all the parameters except S1 (Output 61.2.4).

Output 61.2.5 Additional Estimates

Additional Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
gamma2	0.6603	0.2165	31	3.05	0.0047	0.05	0.2186	1.1019

The "Additional Estimates" table displays results from the ESTIMATE statement (Output 61.2.5). The estimate of γ_2 equals 0.66, agreeing with that obtained by McCulloch (1994). The standard error 0.22 is computed using the delta method (Billingsley 1986; Cox, 1998).

Not shown is the P data set, which contains the original 32 observations and predictions of the p_{ij}.

Example 61.3: Probit-Normal Model with Ordinal Data

The data for this example are from Ezzet and Whitehead (1991), who describe a crossover experiment on two groups of patients using two different inhaler devices (A and B). Patients from group 1 used device A for one week and then device B for another week. Patients from group 2 used the devices in reverse order. The data entered as a SAS data set are as follows:

```
data inhaler;
   input clarity group time freq;
   gt = group*time;
   sub = floor((_n_+1)/2);
   datalines;
1 0 0 59
1 0 1 59
1 0 0 35
2 0 1 35
1 0 0  3
3 0 1  3
1 0 0  2
4 0 1  2
2 0 0 11
1 0 1 11
2 0 0 27
2 0 1 27
2 0 0  2
3 0 1  2
2 0 0  1
4 0 1  1
4 0 0  1
1 0 1  1
4 0 0  1
2 0 1  1
1 1 0 63
1 1 1 63
1 1 0 13
2 1 1 13
2 1 0 40
1 1 1 40
2 1 0 15
2 1 1 15
3 1 0  7
1 1 1  7
3 1 0  2
2 1 1  2
3 1 0  1
3 1 1  1
4 1 0  2
1 1 1  2
4 1 0  1
3 1 1  1
;
```

The response measurement, clarity, is the patients' assessment on the clarity of the leaflet instructions for the devices. The clarity variable is on an ordinal scale, with 1=easy, 2=only clear after rereading, 3=not very clear, and 4=confusing. The group variable indicates the treatment group, and the time variable indicates the time of measurement. The freq variable indicates the number of patients with exactly the same responses. A variable gt is created to indicate a group-by-time interaction, and a variable sub is created to indicate patients.

As in the previous example and in Hedeker and Gibbons (1994), assume an underlying latent continuous variable, here with the form

$$y_{ij} = \beta_0 + \beta_1 g_i + \beta_2 t_j + \beta_3 g_i t_j + u_i + e_{ij}$$

where i indexes patient and j indexes the time period, g_i indicates groups, t_j indicates time, u_i is a patient-level normal random effect, and e_{ij} are iid normal errors. The βs are unknown coefficients to be estimated.

Instead of observing y_{ij}, however, you observe only whether it falls in one of the four intervals: $(-\infty, 0)$, $(0, I1)$, $(I1, I1+I2)$, or $(I1+I2, \infty)$, where $I1$ and $I2$ are both positive. The resulting category is the value assigned to the clarity variable.

The following code sets up and fits this ordinal probit model:

```
proc nlmixed data=inhaler corr ecorr;
   parms b0=0 b1=0 b2=0 b3=0 sd=1 i1=1 i2=1;
   bounds i1 > 0, i2 > 0;
   eta = b0 + b1*group + b2*time + b3*gt + u;
   if (clarity=1) then p = probnorm(-eta);
   else if (clarity=2) then
      p = probnorm(i1-eta) - probnorm(-eta);
   else if (clarity=3) then
      p = probnorm(i1+i2-eta) - probnorm(i1-eta);
   else p = 1 - probnorm(i1+i2-eta);
   if (p > 1e-8) then ll = log(p);
   else ll = -1e100;
   model clarity ~ general(ll);
   random u ~ normal(0,sd*sd) subject=sub;
   replicate freq;
   estimate 'thresh2' i1;
   estimate 'thresh3' i1 + i2;
   estimate 'icc' sd*sd/(1+sd*sd);
run;
```

The PROC NLMIXED statement specifies the input data set and requests correlations both for the parameter estimates (CORR option) and for the additional estimates specified with ESTIMATE statements (ECORR option).

The parameters as defined in the PARMS statement are as follows: b0 (overall intercept), b1 (group main effect), B2 (time main effect), b3 (group-by-time interaction), sd (standard deviation of the random effect), i1 (increment between first and second thresholds), and i2 (increment between second and third thresholds). The BOUNDS statement restricts i1 and i2 to be positive.

The SAS programming statements begin by defining the linear predictor eta, which is a linear combination of the b parameters and a single random effect u. The next statements define the ordinal

likelihood according to the clarity variable, eta, and the increment variables. An error trap is included in case the likelihood becomes too small.

A general log-likelihood specification is used in the MODEL statement, and the RANDOM statement defines the random effect u to have standard deviation sd and subject variable sub. The REPLICATE statement indicates that data for each subject should be replicated according to the freq variable.

The ESTIMATE statements specify the second and third thresholds in terms of the increment variables (the first threshold is assumed to equal zero for model identifiability). Also computed is the intraclass correlation.

The output is as follows.

Output 61.3.1 Specifications for Ordinal Data Model

The NLMIXED Procedure

Specifications

Data Set	WORK.INHALER
Dependent Variable	clarity
Distribution for Dependent Variable	General
Random Effects	u
Distribution for Random Effects	Normal
Subject Variable	sub
Replicate Variable	freq
Optimization Technique	Dual Quasi-Newton
Integration Method	Adaptive Gaussian Quadrature

The "Specifications" table echoes some primary information specified for this nonlinear mixed model (Output 61.3.1). Because the log-likelihood function was expressed with SAS programming statements, the distribution is displayed as *General* in the "Specifications" table.

The "Dimensions" table reveals a total of 286 subjects, which is the sum of the values of the FREQ variable for the second time point. Five quadrature points are selected for log-likelihood evaluation (Output 61.3.2).

Output 61.3.2 Dimensions Table for Ordinal Data Model

Dimensions

Observations Used	38
Observations Not Used	0
Total Observations	38
Subjects	286
Max Obs Per Subject	2
Parameters	7
Quadrature Points	5

Output 61.3.3 Parameter Starting Values and Negative Log Likelihood

Parameters							
b0	b1	b2	b3	sd	i1	i2	NegLogLike
0	0	0	0	1	1	1	538.484276

The "Parameters" table lists the simple starting values for this problem (Output 61.3.3). The "Iteration History" table indicates successful convergence in 13 iterations (Output 61.3.4).

Output 61.3.4 Iteration History

Iteration History					
Iter	Calls	NegLogLike	Diff	MaxGrad	Slope
1	2	476.382511	62.10176	43.75062	-1431.4
2	4	463.228197	13.15431	14.24648	-106.753
3	5	458.528118	4.70008	48.31316	-33.0389
4	6	450.975735	7.552383	22.60098	-40.9954
5	8	448.012701	2.963033	14.86877	-16.7453
6	10	447.245153	0.767549	7.774189	-2.26743
7	11	446.72767	0.517483	3.793533	-1.59278
8	13	446.518273	0.209396	0.868638	-0.37801
9	16	446.514528	0.003745	0.328568	-0.02356
10	18	446.513341	0.001187	0.056778	-0.00183
11	20	446.513314	0.000027	0.010785	-0.00004
12	22	446.51331	3.956E-6	0.004922	-5.41E-6
13	24	446.51331	1.989E-7	0.00047	-4E-7

NOTE: GCONV convergence criterion satisfied.

Output 61.3.5 Fit Statistics for Ordinal Data Model

Fit Statistics	
-2 Log Likelihood	893.0
AIC (smaller is better)	907.0
AICC (smaller is better)	910.8
BIC (smaller is better)	932.6

The "Fit Statistics" table lists the log likelihood and information criteria for model comparisons (Output 61.3.5).

Output 61.3.6 Parameter Estimates at Convergence

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower
b0	−0.6364	0.1342	285	−4.74	<.0001	0.05	−0.9006
b1	0.6007	0.1770	285	3.39	0.0008	0.05	0.2523
b2	0.6015	0.1582	285	3.80	0.0002	0.05	0.2900
b3	−1.4817	0.2385	285	−6.21	<.0001	0.05	−1.9512
sd	0.6599	0.1312	285	5.03	<.0001	0.05	0.4017
i1	1.7450	0.1474	285	11.84	<.0001	0.05	1.4548
i2	0.5985	0.1427	285	4.19	<.0001	0.05	0.3177

Parameter Estimates

Parameter	Upper	Gradient
b0	−0.3722	0.00047
b1	0.9491	0.000265
b2	0.9129	0.00008
b3	−1.0122	0.000102
sd	0.9181	−0.00009
i1	2.0352	0.000202
i2	0.8794	0.000087

The "Parameter Estimates" table indicates significance of all the parameters (Output 61.3.6).

Output 61.3.7 Threshold and Intraclass Correlation Estimates

Additional Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower	Upper
thresh2	1.7450	0.1474	285	11.84	<.0001	0.05	1.4548	2.0352
thresh3	2.3435	0.2073	285	11.31	<.0001	0.05	1.9355	2.7515
icc	0.3034	0.08402	285	3.61	0.0004	0.05	0.1380	0.4687

The "Additional Estimates" table displays results from the ESTIMATE statements (Output 61.3.7).

Example 61.4: Poisson-Normal Model with Count Data

This example uses the pump failure data of Gaver and O'Muircheartaigh (1987). The number of failures and the time of operation are recorded for 10 pumps. Each of the pumps is classified into one of two groups corresponding to either continuous or intermittent operation. The data are as follows:

```
data pump;
  input y t group;
  pump = _n_;
  logtstd = log(t) - 2.4564900;
  datalines;
 5  94.320 1
 1  15.720 2
 5  62.880 1
14 125.760 1
 3   5.240 2
19  31.440 1
 1   1.048 2
 1   1.048 2
 4   2.096 2
22  10.480 2
;
```

Each row denotes data for a single pump, and the variable logtstd contains the centered operation times.

Letting y_{ij} denote the number of failures for the jth pump in the ith group, Draper (1996) considers the following hierarchical model for these data:

$$
\begin{aligned}
y_{ij}|\lambda_{ij} &\sim \text{Poisson}(\lambda_{ij}) \\
\log \lambda_{ij} &= \alpha_i + \beta_i(\log t_{ij} - \overline{\log t}) + e_{ij} \\
e_{ij}|\sigma^2 &\sim \text{Normal}(0, \sigma^2)
\end{aligned}
$$

The model specifies different intercepts and slopes for each group, and the random effect is a mechanism for accounting for overdispersion.

The corresponding PROC NLMIXED statements are as follows:

```
proc nlmixed data=pump;
   parms logsig 0 beta1 1 beta2 1 alpha1 1 alpha2 1;
   if (group = 1) then eta = alpha1 + beta1*logtstd + e;
   else eta = alpha2 + beta2*logtstd + e;
   lambda = exp(eta);
   model y ~ poisson(lambda);
   random e ~ normal(0,exp(2*logsig)) subject=pump;
   estimate 'alpha1-alpha2' alpha1-alpha2;
   estimate 'beta1-beta2' beta1-beta2;
run;
```

The selected output is as follows.

Output 61.4.1 Dimensions Table for Poisson-Normal Model

The NLMIXED Procedure

Dimensions

Observations Used	10
Observations Not Used	0
Total Observations	10
Subjects	10
Max Obs Per Subject	1
Parameters	5
Quadrature Points	5

The "Dimensions" table indicates that data for 10 pumps are used with one observation for each (Output 61.4.1).

Output 61.4.2 Iteration History for Poisson-Normal Model

Iteration History

Iter	Calls	NegLogLike	Diff	MaxGrad	Slope
1	2	30.6986932	2.162768	5.107253	-91.602
2	5	30.0255468	0.673146	2.761738	-11.0489
3	7	29.726325	0.299222	2.990401	-2.36048
4	9	28.7390263	0.987299	2.074431	-3.93678
5	10	28.3161933	0.422833	0.612531	-0.63084
6	12	28.09564	0.220553	0.462162	-0.52684
7	14	28.0438024	0.051838	0.405047	-0.10018
8	16	28.0357134	0.008089	0.135059	-0.01875
9	18	28.033925	0.001788	0.026279	-0.00514
10	20	28.0338744	0.000051	0.00402	-0.00012
11	22	28.0338727	1.681E-6	0.002864	-5.09E-6
12	24	28.0338724	3.199E-7	0.000147	-6.87E-7
13	26	28.0338724	2.532E-9	0.000017	-5.75E-9

NOTE: GCONV convergence criterion satisfied.

The "Iteration History" table indicates successful convergence in 13 iterations (Output 61.4.2).

Output 61.4.3 Fit Statistics for Poisson-Normal Model

Fit Statistics

-2 Log Likelihood	56.1
AIC (smaller is better)	66.1
AICC (smaller is better)	81.1
BIC (smaller is better)	67.6

The "Fit Statistics" table lists the final log likelihood and associated information criteria (Output 61.4.3).

Output 61.4.4 Parameter Estimates and Additional Estimates

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower
logsig	-0.3161	0.3213	9	-0.98	0.3508	0.05	-1.0429
beta1	-0.4256	0.7473	9	-0.57	0.5829	0.05	-2.1162
beta2	0.6097	0.3814	9	1.60	0.1443	0.05	-0.2530
alpha1	2.9644	1.3826	9	2.14	0.0606	0.05	-0.1632
alpha2	1.7992	0.5492	9	3.28	0.0096	0.05	0.5568

Parameter Estimates

Parameter	Upper	Gradient
logsig	0.4107	-0.00002
beta1	1.2649	-0.00002
beta2	1.4724	-1.61E-6
alpha1	6.0921	-5.25E-6
alpha2	3.0415	-5.73E-6

Additional Estimates

Label	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower
alpha1-alpha2	1.1653	1.4855	9	0.78	0.4529	0.05	-2.1952
beta1-beta2	-1.0354	0.8389	9	-1.23	0.2484	0.05	-2.9331

Additional Estimates

Label	Upper
alpha1-alpha2	4.5257
beta1-beta2	0.8623

The "Parameter Estimates" and "Additional Estimates" tables list the maximum likelihood estimates for each of the parameters and two differences (Output 61.4.4). The point estimates for the mean parameters agree fairly closely with the Bayesian posterior means reported by Draper (1996); however, the likelihood-based standard errors are roughly half the Bayesian posterior standard deviations. This is most likely due to the fact that the Bayesian standard deviations account for the uncertainty in estimating σ^2, whereas the likelihood values plug in its estimated value. This downward bias can be corrected somewhat by using the t_9 distribution shown here.

Example 61.5: Failure Time and Frailty Model

In this example an accelerated failure time model with proportional hazard is fitted with and without random effects. The data are from the "Getting Started" example of PROC LIFEREG; see Chapter 48, "The LIFEREG Procedure." Thirty-eight patients are divided into two groups of equal size, and different pain relievers are assigned to each group. The outcome reported is the time in minutes until headache relief. The variable censor indicates whether relief was observed during the course of the observation period (censor = 0) or whether the observation is censored (censor = 1). The SAS DATA step for these data is as follows:

```
data headache;
  input minutes group censor @@;
  patient = _n_;
  datalines;
11  1  0   12  1  0   19  1  0   19  1  0
19  1  0   19  1  0   21  1  0   20  1  0
21  1  0   21  1  0   20  1  0   21  1  0
20  1  0   21  1  0   25  1  0   27  1  0
30  1  0   21  1  1   24  1  1   14  2  0
16  2  0   16  2  0   21  2  0   21  2  0
23  2  0   23  2  0   23  2  0   23  2  0
25  2  1   23  2  0   24  2  0   24  2  0
26  2  1   32  2  1   30  2  1   30  2  0
32  2  1   20  2  1
;
```

In modeling survival data, censoring of observations must be taken into account carefully. In this example, only right censoring occurs. If $g(t,\boldsymbol{\beta})$, $h(t,\boldsymbol{\beta})$, and $G(t,\boldsymbol{\beta})$ denote the density of failure, hazard function, and survival distribution function at time t, respectively, the log likelihood can be written as

$$\begin{aligned} l(\boldsymbol{\beta};\mathbf{t}) &= \sum_{i\in U_u} \log f(t_i,\boldsymbol{\beta}) + \sum_{i\in U_c} \log G(t_i,\boldsymbol{\beta}) \\ &= \sum_{i\in U_u} \log h(t_i,\boldsymbol{\beta}) + \sum_{i=1}^{n} \log G(t_i,\boldsymbol{\beta}) \end{aligned}$$

(See Cox and Oakes 1984, Ch. 3.) In these expressions U_u is the set of uncensored observations, U_c is the set of censored observations, and n denotes the total sample size.

The proportional hazards specification expresses the hazard in terms of a baseline hazard, multiplied by a constant. In this example the hazard is that of a Weibull model and is parameterized as $h(t,\boldsymbol{\beta}) = \gamma\alpha(\alpha t)^{\gamma-1}$ and $\alpha = \exp\{-\mathbf{x}'\boldsymbol{\beta}\}$.

The linear predictor is set equal to the intercept in the reference group (group = 2); this defines the baseline hazard. The corresponding distribution of survival past time t is $G(t,\boldsymbol{\beta}) = \exp\{-(\alpha t)^{\gamma}\}$. See Cox and Oakes (1984, Table 2.1) and the section "Supported Distributions" in Chapter 48, "The LIFEREG Procedure," for this and other survival distribution models and various parameterizations.

The following NLMIXED statements fit this accelerated failure time model and estimate the cumulative distribution function of time to headache relief:

```
proc nlmixed data=headache;
    bounds gamma > 0;
    linp  = b0 - b1*(group-2);
    alpha = exp(-linp);
    G_t   = exp(-(alpha*minutes)**gamma);
    g     = gamma*alpha*((alpha*minutes)**(gamma-1))*G_t;
    ll    = (censor=0)*log(g) + (censor=1)*log(G_t);
    model minutes ~ general(ll);
    predict 1-G_t out=cdf;
run;
```

Output 61.5.1 Specifications Table for Fixed-Effects Failure Time Model

The NLMIXED Procedure

Specifications

Data Set	WORK.HEADACHE
Dependent Variable	minutes
Distribution for Dependent Variable	General
Optimization Technique	Dual Quasi-Newton
Integration Method	None

The "Specifications" table shows that no integration is required, since the model does not contain random effects (Output 61.5.1).

Output 61.5.2 Negative Log Likelihood with Default Starting Values

Parameters

gamma	b0	b1	NegLogLike
1	1	1	263.990327

No starting values were given for the three parameters. The NLMIXED procedure assigns the default value of 1.0 in this case. The negative log likelihood based on these starting values is shown in Output 61.5.2.

Output 61.5.3 Iteration History for Fixed-Effects Failure Time Model

Iteration History

Iter	Calls	NegLogLike	Diff	MaxGrad	Slope
1	2	169.244311	94.74602	22.5599	−2230.83
2	4	142.873508	26.3708	14.88631	−3.64643
3	6	140.633695	2.239814	11.25234	−9.49454
4	8	122.890659	17.74304	19.44959	−2.50807
5	9	121.396959	1.493699	13.85584	−4.55427
6	11	120.623843	0.773116	13.67062	−1.38064
7	12	119.278196	1.345647	15.78014	−1.69072
8	14	116.271325	3.006871	26.94029	−3.2529
9	16	109.427401	6.843925	19.88382	−6.9289
10	19	103.298102	6.129298	12.15647	−4.96054
11	22	101.686239	1.611863	14.24868	−4.34059
12	23	100.027875	1.658364	11.69853	−13.2049
13	26	99.9189048	0.108971	3.602552	−0.55176
14	28	99.8738836	0.045021	0.170712	−0.16645
15	30	99.8736392	0.000244	0.050822	−0.00041
16	32	99.8736351	4.071E−6	0.000705	−6.9E−6
17	34	99.8736351	6.1E−10	4.768E−6	−1.23E−9

NOTE: GCONV convergence criterion satisfied.

The "Iteration History" table shows that the procedure converges after 17 iterations and 34 evaluations of the objective function (Output 61.5.3).

Output 61.5.4 Fit Statistics and Parameter Estimates

Fit Statistics

−2 Log Likelihood	199.7
AIC (smaller is better)	205.7
AICC (smaller is better)	206.5
BIC (smaller is better)	210.7

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower
gamma	4.7128	0.6742	38	6.99	<.0001	0.05	3.3479
b0	3.3091	0.05885	38	56.23	<.0001	0.05	3.1900
b1	−0.1933	0.07856	38	−2.46	0.0185	0.05	−0.3523

Parameter Estimates

Parameter	Upper	Gradient
gamma	6.0777	5.327E−8
b0	3.4283	−4.77E−6
b1	−0.03426	−1.22E−6

The parameter estimates and their standard errors shown in Output 61.5.4 are identical to those obtained with the LIFEREG procedure and the following statements:

```
proc lifereg data=headache;
   class group;
   model minutes*censor(1) = group / dist=weibull;
   output out=new cdf=prob;
run;
```

The t statistic and confidence limits are based on 38 degrees of freedom. The LIFEREG procedure computes z-intervals for the parameter estimates.

For the two groups you obtain

$$\widehat{\alpha}(\text{group} = 1) = \exp\{-3.3091 + 0.1933\} = 0.04434$$
$$\widehat{\alpha}(\text{group} = 2) = \exp\{-3.3091\} = 0.03655$$

The probabilities of headache relief by t minutes are estimated as

$$1 - G(t, \text{group} = 1) = 1 - \exp\{-(0.04434 * t)^{4.7128}\}$$
$$1 - G(t, \text{group} = 2) = 1 - \exp\{-(0.03655 * t)^{4.7128}\}$$

These probabilities, calculated at the observed times, are shown for the two groups in Output 61.5.5 and printed with the following statements:

```
proc print data=cdf;
   var group censor patient minutes pred;
run;
```

Since the slope estimate is negative with p-value of 0.0185, you can infer that pain reliever 1 leads to overall significantly faster relief, but the estimated probabilities give no information about patient-to-patient variation within and between groups. For example, while pain reliever 1 provides faster relief overall, some patients in group 2 might respond more quickly than some patients in group 1. A frailty model enables you to accommodate and estimate patient-to-patient variation in health status by introducing random effects into a subject's hazard function.

Output 61.5.5 Estimated Cumulative Distribution Function

Obs	group	censor	patient	minutes	Pred
1	1	0	1	11	0.03336
2	1	0	2	12	0.04985
3	1	0	3	19	0.35975
4	1	0	4	19	0.35975
5	1	0	5	19	0.35975
6	1	0	6	19	0.35975
7	1	0	7	21	0.51063
8	1	0	8	20	0.43325
9	1	0	9	21	0.51063
10	1	0	10	21	0.51063
11	1	0	11	20	0.43325
12	1	0	12	21	0.51063
13	1	0	13	20	0.43325
14	1	0	14	21	0.51063
15	1	0	15	25	0.80315
16	1	0	16	27	0.90328
17	1	0	17	30	0.97846
18	1	1	18	21	0.51063
19	1	1	19	24	0.73838
20	2	0	20	14	0.04163
21	2	0	21	16	0.07667
22	2	0	22	16	0.07667
23	2	0	23	21	0.24976
24	2	0	24	21	0.24976
25	2	0	25	23	0.35674
26	2	0	26	23	0.35674
27	2	0	27	23	0.35674
28	2	0	28	23	0.35674
29	2	1	29	25	0.47982
30	2	0	30	23	0.35674
31	2	0	31	24	0.41678
32	2	0	32	24	0.41678
33	2	1	33	26	0.54446
34	2	1	34	32	0.87656
35	2	1	35	30	0.78633
36	2	0	36	30	0.78633
37	2	1	37	32	0.87656
38	2	1	38	20	0.20414

The following statements model the hazard for patient i in terms of $\alpha_i = \exp\{-\mathbf{x}_i'\boldsymbol{\beta} - z_i\}$, where z_i is a (normal) random patient effect. Notice that the only difference from the previous NLMIXED statements are the RANDOM statement and the addition of z in the linear predictor. The empirical Bayes estimates of the random effect (RANDOM statement), the parameter estimates (ODS OUTPUT statement), and the estimated cumulative distribution function (PREDICT statement) are saved to subsequently graph the patient-specific distribution functions.

```
ods output ParameterEstimates=est;
proc nlmixed data=headache;
    bounds gamma > 0;
    linp  = b0 - b1*(group-2) + z;
    alpha = exp(-linp);
    G_t   = exp(-(alpha*minutes)**gamma);
    g     = gamma*alpha*((alpha*minutes)**(gamma-1))*G_t;
    ll = (censor=0)*log(g) + (censor=1)*log(G_t);
    model minutes ~ general(ll);
    random z ~ normal(0,exp(2*logsig)) subject=patient out=EB;
    predict 1-G_t out=cdf;
run;
```

Output 61.5.6 Specifications for Random Frailty Model

The NLMIXED Procedure

Specifications

Data Set	WORK.HEADACHE
Dependent Variable	minutes
Distribution for Dependent Variable	General
Random Effects	z
Distribution for Random Effects	Normal
Subject Variable	patient
Optimization Technique	Dual Quasi-Newton
Integration Method	Adaptive Gaussian Quadrature

The "Specifications" table shows that the objective function is computed by adaptive Gaussian quadrature because of the presence of random effects (compare Output 61.5.6 and Output 61.5.1). The "Dimensions" table reports that nine quadrature points are being used to integrate over the random effects (Output 61.5.7).

Output 61.5.7 Dimensions Table for Random Frailty Model

Dimensions

Observations Used	38
Observations Not Used	0
Total Observations	38
Subjects	38
Max Obs Per Subject	1
Parameters	4
Quadrature Points	9

Output 61.5.8 Iteration History for Random Frailty Model

Iteration History

Iter	Calls	NegLogLike	Diff	MaxGrad	Slope
1	5	142.121411	28.82225	12.14484	−88.8664
2	7	136.440369	5.681042	25.93096	−65.7217
3	9	122.972041	13.46833	46.56546	−146.887
4	11	120.904825	2.067216	23.77936	−94.2862
5	13	109.224144	11.68068	57.65493	−92.4075
6	15	105.064733	4.159411	4.824649	−19.5879
7	16	101.902207	3.162526	14.1287	−6.33767
8	18	99.6907395	2.211468	7.676822	−3.42364
9	20	99.3654033	0.325336	5.689204	−0.93978
10	22	99.2602178	0.105185	0.317643	−0.23408
11	24	99.254434	0.005784	1.17351	−0.00556
12	25	99.2456973	0.008737	0.247412	−0.00871
13	27	99.2445445	0.001153	0.104942	−0.00218
14	29	99.2444958	0.000049	0.005646	−0.0001
15	31	99.2444957	9.147E−8	0.000271	−1.84E−7

NOTE: GCONV convergence criterion satisfied.

The procedure converges after 15 iterations (Output 61.5.8). The achieved −2 log likelihood is only 1.2 less than that in the model without random effects (compare Output 61.5.9 and Output 61.5.4). Compared to a chi-square distribution with one degree of freedom, the addition of the random effect appears not to improve the model significantly. You must exercise care, however, in interpreting likelihood ratio tests when the value under the null hypothesis falls on the boundary of the parameter space (see, for example, Self and Liang 1987).

Output 61.5.9 Fit Statistics for Random Frailty Model

Fit Statistics

−2 Log Likelihood	198.5
AIC (smaller is better)	206.5
AICC (smaller is better)	207.7
BIC (smaller is better)	213.0

Output 61.5.10 Parameter Estimates

Parameter Estimates

Parameter	Estimate	Standard Error	DF	t Value	Pr > |t|	Alpha	Lower
gamma	6.2867	2.1334	37	2.95	0.0055	0.05	1.9641
b0	3.2786	0.06576	37	49.86	<.0001	0.05	3.1453
b1	−0.1761	0.08264	37	−2.13	0.0398	0.05	−0.3436
logsig	−1.9027	0.5273	37	−3.61	0.0009	0.05	−2.9711

Parameter Estimates

Parameter	Upper	Gradient
gamma	10.6093	−1.89E−7
b0	3.4118	0.000271
b1	−0.00868	0.000111
logsig	−0.8343	0.000027

The estimate of the Weibull parameter has changed drastically from the model without random effects (compare Output 61.5.10 and Output 61.5.4). The variance of the patient random effect is $\exp\{-2 \times 1.9027\} = 0.02225$. The listing in Output 61.5.11 shows the empirical Bayes estimates of the random effects. These are the adjustments made to the linear predictor in order to obtain a patient's survival distribution. The listing is produced with the following statements:

```
proc print data=eb;
   var Patient Effect Estimate StdErrPred;
run;
```

Output 61.5.11 Empirical Bayes Estimates of Random Effects

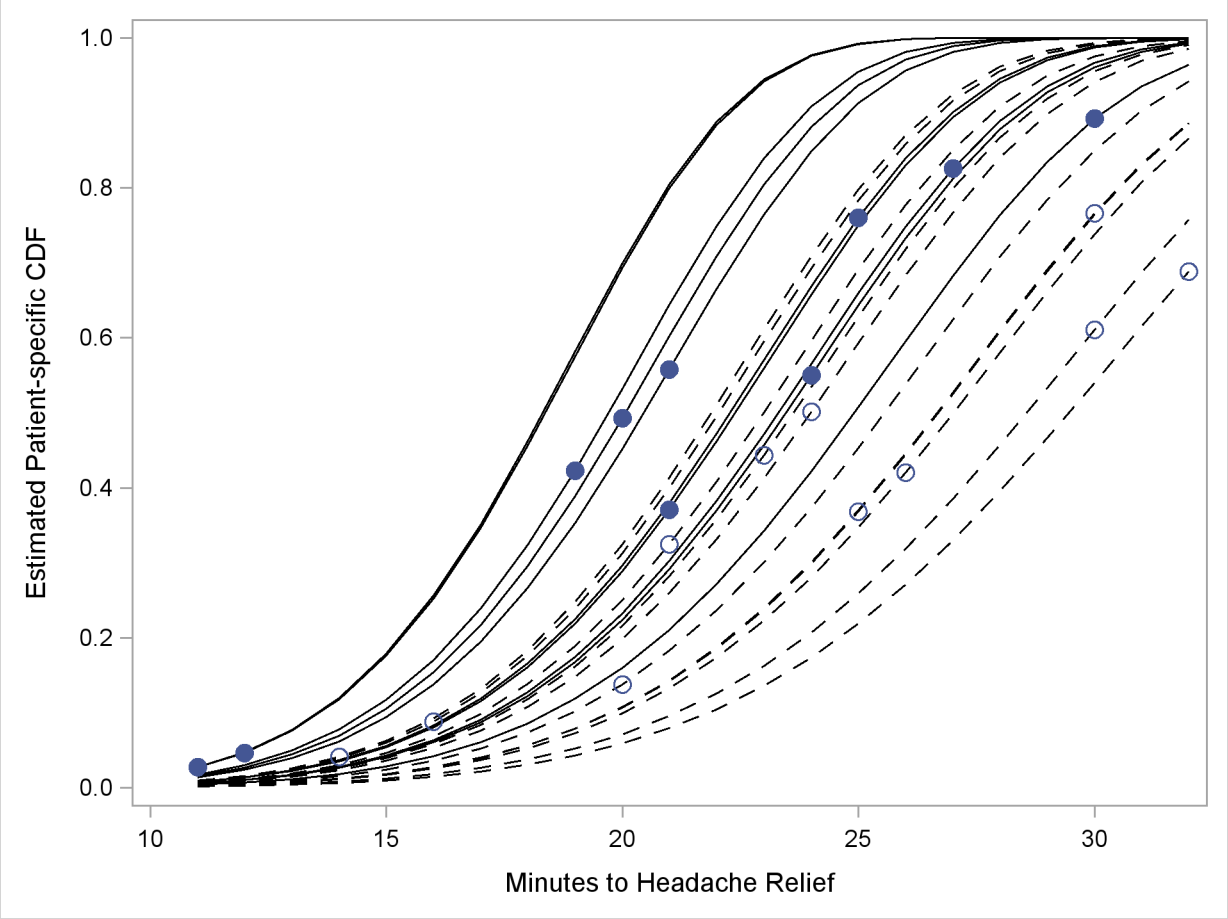
Obs	patient	Effect	Estimate	StdErr Pred
1	1	z	−0.13597	0.23249
2	2	z	−0.13323	0.22793
3	3	z	−0.06294	0.13813
4	4	z	−0.06294	0.13813
5	5	z	−0.06294	0.13813
6	6	z	−0.06294	0.13813
7	7	z	−0.02568	0.11759
8	8	z	−0.04499	0.12618
9	9	z	−0.02568	0.11759
10	10	z	−0.02568	0.11759
11	11	z	−0.04499	0.12618
12	12	z	−0.02568	0.11759
13	13	z	−0.04499	0.12618
14	14	z	−0.02568	0.11759
15	15	z	0.05980	0.11618
16	16	z	0.10458	0.12684
17	17	z	0.17147	0.14550
18	18	z	0.06471	0.13807
19	19	z	0.11157	0.14604
20	20	z	−0.13406	0.22899
21	21	z	−0.12698	0.21667
22	22	z	−0.12698	0.21667
23	23	z	−0.08506	0.15701
24	24	z	−0.08506	0.15701
25	25	z	−0.05797	0.13294
26	26	z	−0.05797	0.13294
27	27	z	−0.05797	0.13294
28	28	z	−0.05797	0.13294
29	29	z	0.06420	0.13956
30	30	z	−0.05797	0.13294
31	31	z	−0.04266	0.12390
32	32	z	−0.04266	0.12390
33	33	z	0.07618	0.14132
34	34	z	0.16292	0.16460
35	35	z	0.13193	0.15528
36	36	z	0.06327	0.12124
37	37	z	0.16292	0.16460
38	38	z	0.02074	0.14160

The predicted values and patient-specific survival distributions can be plotted with the SAS code that follows:

```
proc transpose data=est(keep=estimate) out=trest;
data trest; set trest; rename col1=gamma col2=b0 col3=b1;
data pred; merge eb(keep=estimate)
                 headache(keep=patient group);
    array pp{2} pred1-pred2;
    if _n_ = 1 then merge trest(keep=gamma b0 b1);
    do time=11 to 32;
      linp      = b0 - b1*(group-2) + estimate;
      pp{group} = 1-exp(- (exp(-linp)*time)**gamma);
      symbolid  = patient+1;
      output;
    end;
    keep pred1 pred2 time patient;
run;
data pred;
   merge pred
         cdf(where  = (group=1)
             rename = (pred=pcdf1 minutes=minutes1)
             keep   = pred minutes group)
         cdf(where  = (group=2)
             rename = (pred=pcdf2 minutes=minutes2)
             keep   = pred minutes group);
   drop group;
run;

proc sgplot data=pred noautolegend;
   label minutes1='Minutes to Headache Relief'
         pcdf1   ='Estimated Patient-specific CDF';
   series  x=time     y=pred1  /
           group=patient
           lineattrs=(pattern=solid color=black);
   series  x=time     y=pred2  /
           group=patient
           lineattrs=(pattern=dash color=black);
   scatter x=minutes1 y=pcdf1  /
           markerattrs=(symbol=CircleFilled size=9);
   scatter x=minutes2 y=pcdf2  /
           markerattrs=(symbol=Circle       size=9);
run;
```

The separation of the distribution functions by groups is evident in Output 61.5.12. Most of the distributions of patients in the first group are to the left of the distributions in the second group. The separation is not complete, however. Several patients who are assigned the second pain reliever experience headache relief more quickly than patients assigned to the first group.

Output 61.5.12 Patient-Specific CDFs and Predicted Values. Pain Reliever 1: Solid Lines, Closed Circles; Pain Reliever 2: Dashed Lines, Open Circles.



References

Abramowitz, M. and Stegun, I. A. (1972), *Handbook of Mathematical Functions*, New York: Dover Publications, Inc.

Anderson, D. A. and Aitkin, M. (1985), "Variance Component Models with Binary Response: Interviewer Variability," *Journal of the Royal Statistical Society B,* 47, 203–210.

Beal, S. L. and Sheiner, L. B. (1982), "Estimating Population Kinetics," *CRC Crit. Rev. Biomed. Eng.,* 8, 195–222.

Beal, S. L. and Sheiner, L. B. (1988), "Heteroscedastic Nonlinear Regression," *Technometrics,* 30, 327–338.

Beal, S. L. and Sheiner, L. B., eds. (1992), *NONMEM User's Guide,* University of California, San Francisco, NONMEM Project Group.

Beale, E. M. L. (1972), "A Derivation of Conjugate Gradients," in *Numerical Methods for Nonlinear*

Optimization, ed. F.A. Lootsma, London: Academic Press.

Beitler, P. J. and Landis, J. R. (1985), "A Mixed-Effects Model for Categorical Data," *Biometrics,* 41, 991–1000.

Billingsley, P. (1986), *Probability and Measure,* Second Edition, New York: John Wiley & Sons, Inc.

Booth, J. G. and Hobert, J. P. (1998), "Standard Errors of Prediction in Generalized Linear Mixed Models," *Journal of the American Statistical Association,* 93, 262–272.

Breslow, N. E. and Clayton, D. G. (1993), "Approximate Inference in Generalized Linear Mixed Models," *Journal of the American Statistical Association,* 88, 9–25.

Burnham, K. P. and Anderson, D. R. (1998), *Model Selection and Inference: A Practical Information-Theoretic Approach,* New York: Springer-Verlag.

Cox, C. (1998), "Delta Method," *Encyclopedia of Biostatistics,* Eds. Peter Armitage and Theodore Colton, New York: John Wiley, 1125–1127.

Cox, D. R. and Oakes, D. (1984), *Analysis of Survival Data,* New York: Chapman & Hall.

Cramer, J. S. (1986), *Econometric Applications of Maximum Likelihood Methods*, Cambridge, England: Cambridge University Press.

Crouch, E. A. C. and Spiegelman, D. (1990), "The Evaluation of Integrals of the Form $\int_{-\infty}^{\infty} f(t)\exp(-t^2)dt$: Application to Logistic-Normal Models," *Journal of the American Statistical Association,* 85, 464–469

Davidian, M. and Gallant, R. A. (1993), "The Nonlinear Mixed Effects Model with a Smooth Random Effects Density," *Biometrika,* 80, 475–488.

Davidian, M. and Giltinan, D. M. (1995), *Nonlinear Models for Repeated Measurement Data,* New York: Chapman & Hall.

Dennis, J. E., Gay, D. M., and Welsch, R. E. (1981), "An Adaptive Nonlinear Least-Squares Algorithm," *ACM Transactions on Mathematical Software*, 7, 348–368.

Dennis, J. E. and Mei, H. H. W. (1979), "Two New Unconstrained Optimization Algorithms Which Use Function and Gradient Values," *J. Optim. Theory Appl.*, 28, 453–482.

Dennis, J. E. and Schnabel, R. B. (1983), *Numerical Methods for Unconstrained Optimization and Nonlinear Equations,* Englewood Cliffs, NJ: Prentice-Hall.

Diggle, P. J., Liang, K. Y., and Zeger, S. L. (1994), *Analysis of Longitudinal Data,* Oxford: Clarendon Press.

Draper, D. (1996), "Discussion of the Paper by Lee and Nelder," *Journal of the Royal Statistical Society, Series B*, 58, 662–663.

Draper, N. R. and Smith, H. (1981), *Applied Regression Analysis,* Second Edition, New York: John Wiley & Sons, Inc.

Engel, B. and Keen, A. (1992), "A Simple Approach for the Analysis of Generalized Linear Mixed Models," LWA-92-6, Agricultural Mathematics Group (GLW-DLO), Wageningen, The Netherlands.

Eskow, E. and Schnabel, R. B. (1991), "Algorithm 695: Software for a New Modified Cholesky Factorization," *Transactions on Mathematical Software*, 17(3), 306–312.

Ezzet, F. and Whitehead, J. (1991), "A Random Effects Model for Ordinal Responses from a Crossover Trial," *Statistics in Medicine,* 10, 901–907.

Fletcher, R. (1987), *Practical Methods of Optimization,* Second Edition, Chichester: John Wiley & Sons, Inc.

Galecki, A. T. (1998), "NLMEM: New SAS/IML Macro for Hierarchical Nonlinear Models," *Computer Methods and Programs in Biomedicine,* 55, 207–216.

Gallant, A. R. (1987), *Nonlinear Statistical Models*, New York: John Wiley & Sons, Inc.

Gaver, D. P. and O'Muircheartaigh, I. G. (1987), "Robust Empirical Bayes Analysis of Event Rates," *Technometrics*, 29, 1–15.

Gay, D. M. (1983), "Subroutines for Unconstrained Minimization," *ACM Transactions on Mathematical Software*, 9, 503–524.

Gilmour, A. R., Anderson, R. D., and Rae, A. L. (1985), "The Analysis of Binomial Data by Generalized Linear Mixed Model," *Biometrika,* 72, 593–599.

Goldstein, H. (1991), "Nonlinear Multilevel Models, with an Application to Discrete Response Data," *Biometrika,* 78, 45–51.

Golub, G. H., and Welsch, J. H. (1969), "Calculation of Gaussian Quadrature Rules," *Mathematical Computing,* 23, 221–230.

Harville, D. A. and Mee, R. W. (1984), "A Mixed-Model Procedure for Analyzing Ordered Categorical Data," *Biometrics,* 40, 393–408.

Hedeker, D. and Gibbons, R. D. (1994), "A Random Effects Ordinal Regression Model for Multilevel Analysis," *Biometrics,* 50, 933–944.

Hurvich, C. M. and Tsai, C.-L. (1989), "Regression and Time Series Model Selection in Small Samples," *Biometrika,* 76, 297–307.

Lin, X. and Breslow, N. E. (1996), "Bias Correction in Generalized Linear Mixed Models with Multiple Components of Dispersion," *Journal of the American Statistical Association,* 91, 1007–1016.

Lindstrom, M. J. and Bates, D. M. (1990), "Nonlinear Mixed Effects Models for Repeated Measures Data," *Biometrics,* 46, 673–687.

Littell, R. C., Milliken, G. A., Stroup, W. W., Wolfinger, R. D., and Schabenberger, O. (2006), *SAS System for Mixed Models, Second Edition* Cary, NC: SAS Institute Inc.

Liu, Q. and Pierce, D. A. (1994), "A Note on Gauss-Hermite Quadrature," *Biometrika,* 81, 624–629.

Longford, N. T. (1994), "Logistic Regression with Random Coefficients," *Computational Statistics and Data Analysis,* 17, 1–15.

McCulloch, C. E. (1994), "Maximum Likelihood Variance Components Estimation for Binary Data," *Journal of the American Statistical Association,* 89, 330–335.

McGilchrist, C. E. (1994), "Estimation in Generalized Mixed Models," *Journal of the Royal Statistical Society B,* 56, 61–69.

Moré, J. J. (1978), "The Levenberg-Marquardt Algorithm: Implementation and Theory," in *Lecture Notes in Mathematics 630*, ed. G.A. Watson, Berlin-Heidelberg-New York: Springer Verlag.

Moré, J. J. and Sorensen, D. C. (1983), "Computing a Trust-region Step," *SIAM Journal on Scientific and Statistical Computing*, 4, 553–572.

Ochi, Y. and Prentice, R. L. (1984), "Likelihood Inference in a Correlated Probit Regression Model," *Biometrika,* 71, 531–543.

Pierce, D. A. and Sands, B. R. (1975), *Extra-Bernoulli Variation in Binary Data,* Technical Report 46, Department of Statistics, Oregon State University.

Pinheiro, J. C. and Bates, D. M. (1995), "Approximations to the Log-Likelihood Function in the Nonlinear Mixed-Effects Model," *Journal of Computational and Graphical Statistics,* 4, 12–35.

Pringle, R. M. and Rayner, A. A. (1971), *Generalized Inverse Matrices with Applications to Statistics*, New York: Hafner Publishing Co.

Polak, E. (1971), *Computational Methods in Optimization*, New York, Academic Press.

Powell, J. M. D. (1977), "Restart Procedures for the Conjugate Gradient Method," *Math. Prog.*, 12, 241–254.

Roe, D. J. (1997) "Comparison of Population Pharmacokinetic Modeling Methods Using Simulated Data: Results from the Population Modeling Workgroup," *Statistics in Medicine,* 16, 1241–1262.

Rodriguez, G. and Goldman, N. (1995), "An Assessment of Estimation Procedures for Multilevel Models with Binary Response," *Journal of the Royal Statistical Society, Series A,* 158, 73–89.

Schall, R. (1991). "Estimation in Generalized Linear Models with Random Effects," *Biometrika*, 78, 719–727.

Schittkowski, K. and Stoer, J. (1979), "A Factorization Method for the Solution of Constrained Linear Least Squares Problems Allowing Subsequent Data Changes," *Numerische Mathematik*, 31, 431–463.

Self, S. G. and Liang, K. Y. (1987), "Asymptotic Properties of Maximum Likelihood estimators and Likelihood Ratio Tests under Nonstandard Conditions," *Journal of the American Statistical Association,* 82, 605–610.

Serfling, R. J. (1980), *Approximation Theorems of Mathematical Statistics,* New York, John Wiley & Sons, Inc.

Sheiner L. B. and Beal S. L. (1980), "Evaluation of Methods for Estimating Population Pharmacokinetic Parameters. I. Michaelis-Menten Model: Routine Clinical Pharmacokinetic Data," *Journal of Pharmacokinetics and Biopharmaceutics,* 8, 553–571.

Sheiner, L. B. and Beal, S. L. (1985), "Pharmacokinetic Parameter Estimates from Several Least Squares Procedures: Superiority of Extended Least Squares," *Journal of Pharmacokinetics and Biopharmaceutics,* 13, 185–201.

Smith, S. P. (1995), "Differentiation of the Cholesky Algorithm," *Journal of Computational and Graphical Statistics,* 4, 134–147.

Stiratelli, R., Laird, N. M., and Ware, J. H. (1984), "Random Effects Models for Serial Observations with Binary Response," *Biometrics,* 40, 961–971.

Vonesh, E. F., (1992), "Nonlinear Models for the Analysis of Longitudinal Data," *Statistics in Medicine,* 11, 1929–1954.

Vonesh, E. F., (1996), "A Note on Laplace's Approximation in Nonlinear Mixed Effects Models," *Biometrika,* 83, 447–452.

Vonesh, E. F. and Chinchilli, V. M. (1997), *Linear and Nonlinear Models for the Analysis of Repeated Measurements,* New York: Marcel Dekker.

Weil, C. S. (1970), "Selection of the Valid Number of Sampling Units and Consideration of Their Combination in Toxicological Studies Involving Reproduction, Teratogenesis, or Carcinogenesis," *Food and Cosmetic Toxicology,* 8, 177–182.

White, H. (1982), "Maximum Likelihood Estimation of Misspecified Models," *Econometrica,* 50, 1–25.

Williams, D. A. (1975), "The Analysis of Binary Responses from Toxicological Experiments Involving Reproduction and Teratogenicity," *Biometrics,* 31, 949–952.

Wolfinger R. D. (1993), "Laplace's Approximation for Nonlinear Mixed Models," *Biometrika,* 80, 791–795.

Wolfinger, R.D. (1997), "Comment: Experiences with the SAS Macro NLINMIX," *Statistics in Medicine,* 16, 1258–1259.

Wolfinger, R. D., and Lin, X. (1997), "Two Taylor-Series Approximation Methods for Nonlinear Mixed Models," *Computational Statistics and Data Analysis,* 25, 465–490.

Wolfinger, R. D. and O'Connell, M. (1993), "Generalized Linear Mixed Models: a Pseudo-likelihood Approach," *Journal of Statistical Computation and Simulation,* 48, 233–243.

Yuh, L., Beal, S., Davidian, M., Harrison, F., Hester, A., Kowalski, K., Vonesh, E., Wolfinger, R. (1994), "Population Pharmacokinetic/Pharmacodynamic Methodology and Applications: A Bibliography," *Biometrics,* 50, 566–575.

Chapter 62

The NPAR1WAY Procedure

Contents

Overview: NPAR1WAY Procedure **4444**
Getting Started: NPAR1WAY Procedure **4445**
Syntax: NPAR1WAY Procedure **4456**
PROC NPAR1WAY Statement 4456
BY Statement 4462
CLASS Statement 4462
EXACT Statement 4462
FREQ Statement 4465
OUTPUT Statement 4465
VAR Statement 4466
Details: NPAR1WAY Procedure **4466**
Missing Values 4466
Tied Values 4467
Statistical Computations 4467
Simple Linear Rank Tests for Two-Sample Data 4467
One-Way ANOVA Tests 4469
Scores for Linear Rank and One-Way ANOVA Tests 4470
Hodges-Lehmann Estimation of Location Shift 4472
Tests Based on the Empirical Distribution Function 4474
Exact Tests 4476
Output Data Set 4480
Displayed Output 4486
ODS Table Names 4490
ODS Graphics 4492
Examples: NPAR1WAY Procedure **4493**
Example 62.1: Two-Sample Location Tests and Plots 4493
Example 62.2: EDF Statistics and EDF Plot 4498
Example 62.3: Exact Wilcoxon Two-Sample Test 4499
Example 62.4: Hodges-Lehmann Estimation 4501
Example 62.5: Exact Savage Multisample Test 4502
References **4503**

Overview: NPAR1WAY Procedure

The NPAR1WAY procedure performs nonparametric tests for location and scale differences across a one-way classification. PROC NPAR1WAY also provides a standard analysis of variance on the raw data and tests based on the empirical distribution function.

PROC NPAR1WAY performs tests for location and scale differences based on the following scores of a response variable: Wilcoxon, median, Van der Waerden (normal), Savage, Siegel-Tukey, Ansari-Bradley, Klotz, Mood, and Conover. Additionally, PROC NPAR1WAY provides tests that use the raw input data as scores. When the data are classified into two samples, tests are based on simple linear rank statistics. When the data are classified into more than two samples, tests are based on one-way ANOVA statistics. Both asymptotic and exact *p*-values are available for these tests. PROC NPAR1WAY also provides Hodges-Lehmann estimation, including exact confidence limits for the location shift.

PROC NPAR1WAY computes empirical distribution function (EDF) statistics, which test whether the distribution of a variable is the same across different groups. These include the Kolmogorov-Smirnov test, the Cramer-von Mises test, and, when the data are classified into only two samples, the Kuiper test. Exact *p*-values are available for the two-sample Kolmogorov-Smirnov test.

PROC NPAR1WAY uses the Output Delivery System (ODS), a SAS subsystem that provides capabilities for displaying and controlling the output from SAS procedures. ODS enables you to convert any of the output from PROC NPAR1WAY into a SAS data set. See the section "ODS Table Names" on page 4490 for more information.

PROC NPAR1WAY now uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the statistical graphics available with the NPAR1WAY procedure, see the PLOTS= option in the PROC NPAR1WAY statement and the section "ODS Graphics" on page 4492.

Getting Started: NPAR1WAY Procedure

This example illustrates how you can use PROC NPAR1WAY to perform a one-way nonparametric analysis. The data from Halverson and Sherwood (1930) consist of weight gain measurements for five different levels of gossypol additive in animal feed. Gossypol is a substance contained in cottonseed shells, and these data were collected to study the effect of gossypol on animal nutrition.

The following DATA step statements create the SAS data set Gossypol:

```
data Gossypol;
   input Dose n;
   do i=1 to n;
      input Gain @@;
      output;
      end;
   datalines;
0 16
228 229 218 216 224 208 235 229 233 219 224 220 232 200 208 232
.04 11
186 229 220 208 228 198 222 273 216 198 213
.07 12
179 193 183 180 143 204 114 188 178 134 208 196
.10 17
130 87 135 116 118 165 151 59 126 64 78 94 150 160 122 110 178
.13 11
154 130 130 118 118 104 112 134 98 100 104
;
```

The data set Gossypol contains the variable Dose, which represents the amount of gossypol additive, and the variable Gain, which represents the weight gain.

Researchers are interested in whether there is a difference in weight gain among animals receiving the different dose levels of gossypol. The following statements invoke the NPAR1WAY procedure to perform a nonparametric analysis of this problem:

```
proc npar1way data=Gossypol;
   class Dose;
   var Gain;
run;
```

The variable Dose is the CLASS variable, and the VAR statement specifies the variable Gain is the response variable. The CLASS statement is required, and you must name only one CLASS variable. You can name one or more analysis variables in the VAR statement. If you omit the VAR statement, PROC NPAR1WAY analyzes all numeric variables in the data set except for the CLASS variable, the FREQ variable, and the BY variables.

Since no analysis options are specified in the PROC NPAR1WAY statement, the ANOVA, WILCOXON, MEDIAN, VW, SAVAGE, and EDF options are invoked by default. The tables in the following figures show the results of these analyses.

The tables in Figure 62.1 are produced with the ANOVA option. For each level of the CLASS variable Dose, PROC NPAR1WAY displays the number of observations and the mean of the analysis variable Gain. PROC NPAR1WAY displays a standard analysis of variance on the raw data. This gives the same results as the GLM and ANOVA procedures. The p-value for the F test is <0.0001, which indicates that Dose accounts for a significant portion of the variability of the dependent variable Gain.

Figure 62.1 Analysis of Variance

The NPAR1WAY Procedure

Analysis of Variance for Variable Gain
Classified by Variable Dose

Dose	N	Mean
0	16	222.187500
0.04	11	217.363636
0.07	12	175.000000
0.1	17	120.176471
0.13	11	118.363636

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Among	4	140082.986077	35020.74652	55.8143	<.0001
Within	62	38901.998997	627.45160		

Average scores were used for ties.

The WILCOXON option produces the output in Figure 62.2. PROC NPAR1WAY first provides a summary of the Wilcoxon scores for the analysis variable Gain by class level. For each level of the CLASS variable Dose, PROC NPAR1WAY displays the following information: number of observations, sum of the Wilcoxon scores, expected sum under the null hypothesis of no difference among class levels, standard deviation under the null hypothesis, and mean score.

Next PROC NPAR1WAY displays the one-way ANOVA statistic, which for Wilcoxon scores is known as the Kruskal-Wallis test. The statistic equals 52.6656, with four degrees of freedom, which is the number of class levels minus one. The p-value, or probability of a larger statistic under the null hypothesis, is <0.0001. This leads to rejection of the null hypothesis that there is no difference in location for Gain among the levels of Dose. This p-value is asymptotic, computed from the asymptotic chi-square distribution of the test statistic. For certain data sets it might also be useful to compute the exact p-value—for example, for small data sets or for data sets that are sparse, skewed, or heavily tied. You can use the EXACT statement to request exact p-values for any of the location or scale tests available in PROC NPAR1WAY.

Figure 62.2 Wilcoxon Score Analysis

Wilcoxon Scores (Rank Sums) for Variable Gain
Classified by Variable Dose

Dose	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
0	16	890.50	544.0	67.978966	55.656250
0.04	11	555.00	374.0	59.063588	50.454545
0.07	12	395.50	408.0	61.136622	32.958333
0.1	17	275.50	578.0	69.380741	16.205882
0.13	11	161.50	374.0	59.063588	14.681818

Average scores were used for ties.

Kruskal-Wallis Test

Chi-Square	52.6656
DF	4
Pr > Chi-Square	<.0001

Figure 62.3 through Figure 62.5 display the analyses produced by the MEDIAN, VW, and SAVAGE options. For each score type, PROC NPAR1WAY provides a summary of scores and the one-way ANOVA statistic, as previously described for Wilcoxon scores. Other score types available in PROC NPAR1WAY are Siegel-Tukey, Ansari-Bradley, Klotz, and Mood, which can be used to test for scale differences. Conover scores can be used to test for differences in both location and scale. Additionally, you can specify the SCORES=DATA option, which uses the input data as scores. This option gives you the flexibility to construct any scores for your data with the DATA step and then analyze these scores with PROC NPAR1WAY.

Figure 62.3 Median Score Analysis

Median Scores (Number of Points Above Median) for Variable Gain
Classified by Variable Dose

Dose	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
0	16	16.0	7.880597	1.757902	1.00
0.04	11	11.0	5.417910	1.527355	1.00
0.07	12	6.0	5.910448	1.580963	0.50
0.1	17	0.0	8.373134	1.794152	0.00
0.13	11	0.0	5.417910	1.527355	0.00

Average scores were used for ties.

Median One-Way Analysis

Chi-Square	54.1765
DF	4
Pr > Chi-Square	<.0001

Figure 62.4 Van der Waerden (Normal) Score Analysis

Van der Waerden Scores (Normal) for Variable Gain
Classified by Variable Dose

Dose	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
0	16	16.116474	0.0	3.325957	1.007280
0.04	11	8.340899	0.0	2.889761	0.758264
0.07	12	-0.576674	0.0	2.991186	-0.048056
0.1	17	-14.688921	0.0	3.394540	-0.864054
0.13	11	-9.191777	0.0	2.889761	-0.835616

Average scores were used for ties.

Van der Waerden One-Way Analysis

Chi-Square	47.2972
DF	4
Pr > Chi-Square	<.0001

Figure 62.5 Savage Score Analysis

Savage Scores (Exponential) for Variable Gain
Classified by Variable Dose

Dose	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
0	16	16.074391	0.0	3.385275	1.004649
0.04	11	7.693099	0.0	2.941300	0.699373
0.07	12	-3.584958	0.0	3.044534	-0.298746
0.1	17	-11.979488	0.0	3.455082	-0.704676
0.13	11	-8.203044	0.0	2.941300	-0.745731

Average scores were used for ties.

Savage One-Way Analysis

Chi-Square	39.4908
DF	4
Pr > Chi-Square	<.0001

The tables in Figure 62.6 display the empirical distribution function statistics, comparing the distribution of Gain for the different levels of Dose. These tables are produced by the EDF option, and they include Kolmogorov-Smirnov statistics and Cramer-von Mises statistics.

Figure 62.6 Empirical Distribution Function Analysis

Kolmogorov-Smirnov Test for Variable Gain
Classified by Variable Dose

Dose	N	EDF at Maximum	Deviation from Mean at Maximum
0	16	0.000000	-1.910448
0.04	11	0.000000	-1.584060
0.07	12	0.333333	-0.499796
0.1	17	1.000000	2.153861
0.13	11	1.000000	1.732565
Total	67	0.477612	

Maximum Deviation Occurred at Observation 36
Value of Gain at Maximum = 178.0

Kolmogorov-Smirnov Statistics (Asymptotic)
KS 0.457928 KSa 3.748300

Cramer-von Mises Test for Variable Gain
Classified by Variable Dose

Dose	N	Summed Deviation from Mean
0	16	2.165210
0.04	11	0.918280
0.07	12	0.348227
0.1	17	1.497542
0.13	11	1.335745

Cramer-von Mises Statistics (Asymptotic)
CM 0.093508 CMa 6.265003

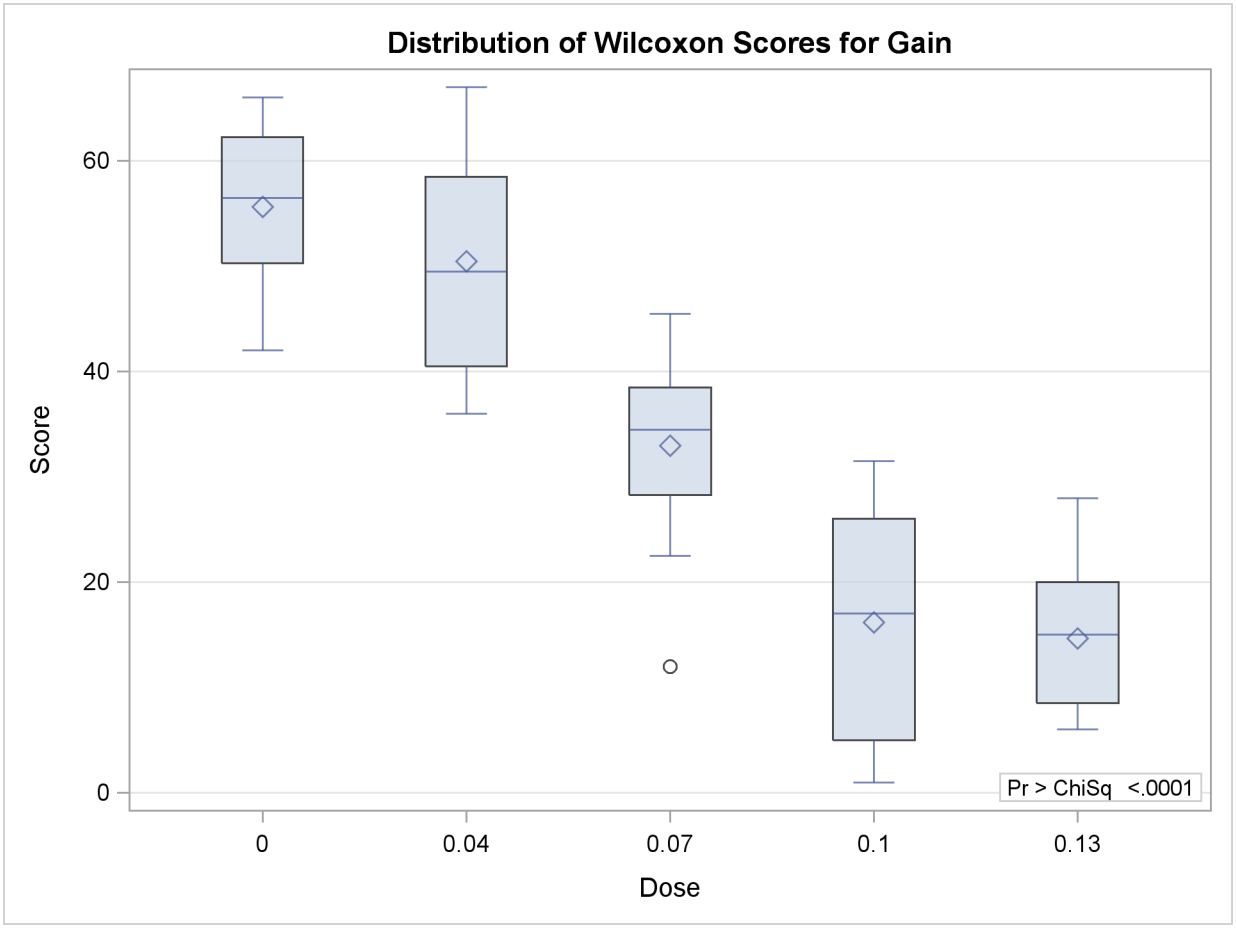
PROC NPAR1WAY now uses ODS Graphics to create graphs as part of its output. The following statements produce a box plot of Wilcoxon scores for Gain classified by Dose. Before requesting graphs you must enable ODS Graphics with the **ODS GRAPHICS ON** statement.

```
ods graphics on;
proc npar1way data=Gossypol plots(only)=wilcoxonboxplot;
   class Dose;
   var Gain;
run;
ods graphics off;
```

Figure 62.7 displays the box plot of Wilcoxon scores. This graph corresponds to the Wilcoxon scores analysis shown in Figure 62.2. To remove the *p*-value from the box plot display, you can specify the NOSTATS plot option in parentheses following the WILCOXONBOXPLOT option.

Box plots are available for all PROC NPAR1WAY score types except median scores, which are displayed with a stacked bar chart. If you do not specify the PLOTS= option but have enabled ODS Graphics, then PROC NPAR1WAY produces all plots associated with the analyses you request.

Figure 62.7 Box Plot of Wilcoxon Scores



In the preceding example, the CLASS variable Dose has five levels, and the analyses examine possible differences among these five levels, or samples. The following statements invoke the NPAR1WAY procedure to perform a nonparametric analysis of the two lowest levels of Dose:

```
proc npar1way data=Gossypol;
   where Dose <= .04;
   class Dose;
   var Gain;
run;
```

The tables in the following figures show the results of this two-sample analysis. The tables in Figure 62.8 are produced by the ANOVA option.

Figure 62.8 Analysis of Variance for Two-Sample Data

The NPAR1WAY Procedure

Analysis of Variance for Variable Gain
Classified by Variable Dose

Dose	N	Mean
0	16	222.187500
0.04	11	217.363636

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Among	1	151.683712	151.683712	0.5587	0.4617
Within	25	6786.982955	271.479318		

Average scores were used for ties.

Figure 62.9 displays the output produced by the WILCOXON option. PROC NPAR1WAY provides a summary of the Wilcoxon scores for the analysis variable Gain for each of the two class levels. Since there are only two levels, PROC NPAR1WAY displays the two-sample test, based on the simple linear rank statistic with Wilcoxon scores. The normal approximation includes a continuity correction. To remove the continuity correction, you can specify the CORRECT=NO option. PROC NPAR1WAY also gives a *t* approximation for the Wilcoxon two-sample test. Like the multisample analysis, PROC NPAR1WAY computes a one-way ANOVA statistic, which for Wilcoxon scores is known as the Kruskal-Wallis test. All these *p*-values show no difference in Gain for the two Dose levels at the 0.05 level of significance.

Figure 62.10 through Figure 62.12 display the two-sample analyses produced by the MEDIAN, VW, and SAVAGE options.

Figure 62.9 Wilcoxon Two-Sample Analysis

Wilcoxon Scores (Rank Sums) for Variable Gain
Classified by Variable Dose

Dose	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
0	16	253.50	224.0	20.221565	15.843750
0.04	11	124.50	154.0	20.221565	11.318182

Average scores were used for ties.

Wilcoxon Two-Sample Test

Statistic	124.5000
Normal Approximation	
Z	-1.4341
One-Sided Pr < Z	0.0758
Two-Sided Pr > |Z|	0.1515
t Approximation	
One-Sided Pr < Z	0.0817
Two-Sided Pr > |Z|	0.1635

Z includes a continuity correction of 0.5.

Kruskal-Wallis Test

Chi-Square	2.1282
DF	1
Pr > Chi-Square	0.1446

Figure 62.10 Median Two-Sample Analysis

Median Scores (Number of Points Above Median) for Variable Gain
Classified by Variable Dose

Dose	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
0	16	9.0	7.703704	1.299995	0.562500
0.04	11	4.0	5.296296	1.299995	0.363636

Average scores were used for ties.

Median Two-Sample Test

Statistic	4.0000
Z	-0.9972
One-Sided Pr < Z	0.1593
Two-Sided Pr > |Z|	0.3187

Median One-Way Analysis

Chi-Square	0.9943
DF	1
Pr > Chi-Square	0.3187

Figure 62.11 Van der Waerden (Normal) Two-Sample Analysis

Van der Waerden Scores (Normal) for Variable Gain
Classified by Variable Dose

Dose	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
0	16	3.346520	0.0	2.320336	0.209157
0.04	11	-3.346520	0.0	2.320336	-0.304229

Average scores were used for ties.

Van der Waerden Two-Sample Test

Statistic	-3.3465
Z	-1.4423
One-Sided Pr < Z	0.0746
Two-Sided Pr > |Z|	0.1492

Van der Waerden One-Way Analysis

Chi-Square	2.0801
DF	1
Pr > Chi-Square	0.1492

Figure 62.12 Savage Two-Sample Analysis

Savage Scores (Exponential) for Variable Gain
Classified by Variable Dose

Dose	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
0	16	1.834554	0.0	2.401839	0.114660
0.04	11	-1.834554	0.0	2.401839	-0.166778

Average scores were used for ties.

Savage Two-Sample Test

Statistic	-1.8346
Z	-0.7638
One-Sided Pr < Z	0.2225
Two-Sided Pr > |Z|	0.4450

Savage One-Way Analysis

Chi-Square	0.5834
DF	1
Pr > Chi-Square	0.4450

The tables in Figure 62.13 display the empirical distribution function statistics, comparing the distribution of Gain for the two levels of Dose. The *p*-value for the Kolmogorov-Smirnov two-sample test is 0.6199, which indicates no rejection of the null hypothesis that the Gain distributions are identical for the two levels of Dose.

Figure 62.13 Two-Sample EDF Tests

```
      Kolmogorov-Smirnov Test for Variable Gain
             Classified by Variable Dose

                        EDF at     Deviation from Mean
Dose          N        Maximum          at Maximum
-------------------------------------------------------
   0         16       0.250000           -0.481481
0.04         11       0.545455            0.580689
Total        27       0.370370

   Maximum Deviation Occurred at Observation 4
         Value of Gain at Maximum = 216.0

 Kolmogorov-Smirnov Two-Sample Test (Asymptotic)
       KS    0.145172     D          0.295455
       KSa   0.754337     Pr > KSa   0.6199

     Cramer-von Mises Test for Variable Gain
           Classified by Variable Dose

                              Summed Deviation
     Dose            N             from Mean
     ----------------------------------------
        0           16              0.098638
     0.04           11              0.143474

     Cramer-von Mises Statistics (Asymptotic)
          CM  0.008967     CMa  0.242112

          Kuiper Test for Variable Gain
           Classified by Variable Dose

                               Deviation
          Dose          N       from Mean
          -------------------------------
             0         16        0.090909
          0.04         11        0.295455

       Kuiper Two-Sample Test (Asymptotic)
  K  0.386364     Ka  0.986440     Pr > Ka  0.8383
```

Syntax: NPAR1WAY Procedure

The following statements are available in PROC NPAR1WAY:

PROC NPAR1WAY < *options* > ;
BY *variables* ;
CLASS *variable* ;
EXACT *statistic-options* < / *computation-options* > ;
FREQ *variable* ;
OUTPUT < **OUT=***SAS-data-set* > < *options* > ;
VAR *variables* ;

Both the PROC NPAR1WAY statement and the CLASS statement are required for the NPAR1WAY procedure.

The rest of this section gives detailed syntax information for the BY, CLASS, EXACT, FREQ, OUTPUT, and VAR statements in alphabetical order after the description of the PROC NPAR1WAY statement. Table 62.1 summarizes the basic function of each PROC NPAR1WAY statement.

Table 62.1 Summary of PROC NPAR1WAY Statements

Statement	Description
BY	provides separate analyses for each BY group
CLASS	identifies the classification variable
EXACT	requests exact tests
FREQ	identifies a frequency variable
OUTPUT	requests an output data set
VAR	identifies analysis variables

PROC NPAR1WAY Statement

PROC NPAR1WAY < *options* > ;

The PROC NPAR1WAY statement invokes the procedure and optionally identifies the input data set or requests particular analyses. By default, the procedure uses the most recently created SAS data set and omits missing values from the analysis. If you do not specify any analysis options, PROC NPAR1WAY performs an analysis of variance (ANOVA option), tests for location differences (WILCOXON, MEDIAN, SAVAGE, and VW options), and performs empirical distribution function tests (EDF option).

Table 62.2 lists the options available with the PROC NPAR1WAY statement. Descriptions follow in alphabetical order.

Table 62.2 PROC NPAR1WAY Statement Options

Task	Options
Specify the input data set	DATA=
Include missing CLASS values	MISSING
Suppress all displayed output	NOPRINT
Request analyses	AB ANOVA CONOVER D EDF HL KLOTZ MEDIAN MOOD SAVAGE SCORES=DATA ST VW | NORMAL WILCOXON
Set confidence level	ALPHA=
Suppress continuity correction	CORRECT=NO
Request plots	PLOTS=

You can specify the following options in the PROC NPAR1WAY statement.

AB
requests an analysis of Ansari-Bradley scores. See the section "Ansari-Bradley Scores" on page 4471 for more information.

ALPHA=α
specifies the level of the confidence limits for location shift, which you request with the HL option. The value of α must be between 0 and 1, and the default is 0.05. A confidence level of α produces $100(1-\alpha)\%$ confidence limits. The default of ALPHA=0.05 produces 95% confidence limits for the Hodges-Lehmann estimate.

ANOVA
requests a standard analysis of variance on the raw data.

CONOVER
requests an analysis of Conover scores. See the section "Conover Scores" on page 4472 for more information.

CORRECT=NO
suppresses the continuity correction for the Wilcoxon two-sample test and the Siegel-Tukey two-sample test. See the section "Continuity Correction" on page 4469 for more information.

D

requests the one-sided Kolmogorov-Smirnov $D+$ and $D-$ statistics and their asymptotic p-values, in addition to the two-sided D statistic produced by the EDF option for two-sample data. The D option invokes the EDF option. The statistics $D+$ and $D-$ are provided by default if you request exact Kolmogorov-Smirnov statistics with the KS option in the EXACT statement for two-sample data. See the section "Tests Based on the Empirical Distribution Function" on page 4474 for details about Kolmogorov-Smirnov statistics.

DATA=*SAS-data-set*

names the SAS data set to be analyzed by PROC NPAR1WAY. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

EDF

requests statistics based on the empirical distribution function. These include the Kolmogorov-Smirnov and Cramer-von Mises tests and, if there are only two classification levels, the Kuiper test. See the section "Tests Based on the Empirical Distribution Function" on page 4474 for more information.

The EDF option produces the Kolmogorov-Smirnov D statistic for two-sample data. You can request the one-sided $D+$ and $D-$ statistics for two-sample data with the D option.

HL

requests Hodges-Lehmann estimation of the location shift for two-sample data. The HL option provides asymptotic confidence limits for the location shift. These are sometimes known as Moses confidence limits. See the section "Hodges-Lehmann Estimation of Location Shift" on page 4472 for details. You can specify the level of the confidence limits by using the ALPHA= option. The default of ALPHA=0.5 produces 95% confidence limits for the location shift.

KLOTZ

requests an analysis of Klotz scores. See the section "Klotz Scores" on page 4472 for more information.

MEDIAN

requests an analysis of median scores. When there are two classification levels, this option produces the two-sample median test. When there are more than two samples, this option produces the multisample median test, which is also known as the Brown-Mood test. See the section "Median Scores" on page 4470 for more information.

MISSING

treats missing values of the CLASS variable as a valid class level.

MOOD

requests an analysis of Mood scores. See the section "Mood Scores" on page 4472 for more information.

NOPRINT

suppresses the display of all output. You can use the NOPRINT option when you only want to create an output data set. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 20, "Using the Output Delivery System."

PLOTS < (*global-plot-options*) > < = *plot-request* < (*plot-option*) > >
PLOTS < (*global-plot-options*) > < = (*plot-request* < (*plot-option*) > < ... *plot-request* < (*plot-option*) >>)>

requests plots for PROC NPAR1WAY to produce by using ODS Graphics. When you specify only one *plot-request*, you can omit the parentheses around the request. For example:

```
plots=all
plots=wilcoxonboxplot
plots=(wilcoxonboxplot edfplot)
plots(only)=(medianplot normalboxplot)
```

You must enable ODS Graphics before requesting plots, as shown in the following statements:

```
ods graphics on;
proc npar1way plots=wilcoxonboxplot;
   variable response;
   class treatment;
run;
ods graphics off;
```

See Figure 62.7, Output 62.1.2, Output 62.1.4, and Output 62.2.2 for examples of the ODS graphical displays available in PROC NPAR1WAY. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

If you do not specify the PLOTS= option but have enabled ODS Graphics, then PROC NPAR1WAY produces all plots associated with the analyses you request.

Global-Plot-Options

Global-plot-options apply to all plots produced by PROC NPAR1WAY, unless they are altered by a specific *plot-option*. The following *global-plot-options* are available with the PLOTS= option:

NOSTATS
suppresses the *p*-values that are displayed on the plots by default.

ONLY
suppresses the default plots and requests only the plots specified as *plot-requests*.

STATS
displays *p*-values on the plots. This is the default.

You cannot specify both STATS and NOSTATS as *global-plot-options*.

Plot-Requests

The following *plot-requests* are available with the PLOTS= option:

ABBOXPLOT | AB
requests a box plot of Ansari-Bradley scores. This plot is associated with the Ansari-Bradley analysis, which you request with the AB option.

ALL
requests all plots associated with the specified analyses. This is the default if you do not specify the PLOTS(ONLY) option.

ANOVABOXPLOT | ANOVA
requests a box plot of the raw data. This plot is associated with the analysis of variance based on the raw data, which you request with the ANOVA option.

CONOVERBOXPLOT | CONOVER
requests a box plot of Conover scores. This plot is associated with the Conover analysis, which you request with the CONOVER option.

DATASCORESBOXPLOT | DATASCORES
requests a box plot of raw data scores. This plot is associated with the analysis that uses input data as scores, which you request with the SCORES=DATA option.

EDFPLOT | EDF
requests an empirical distribution plot. This plot is associated with the analyses based on the empirical distribution function, which you request with the EDF option.

KLOTZBOXPLOT | KLOTZ
requests a box plot of Klotz scores. This plot is associated with the Klotz analysis, which you request with the KLOTZ option.

MEDIANPLOT | MEDIAN
requests a stacked bar chart showing the frequencies above and below the overall median. This plot is associated with the median score analysis, which you request with the MEDIAN option.

MOODBOXPLOT | MOOD
requests a box plot of Mood scores. This plot is associated with the Mood analysis, which you request with the MOOD option.

NONE
suppresses all plots.

SAVAGEBOXPLOT | SAVAGE
requests a box plot of Savage scores. This plot is associated with the Savage analysis, which you request with the SAVAGE option.

STBOXPLOT | ST
requests a box plot of Siegel-Tukey scores. This plot is associated with the Siegel-Tukey analysis, which you request with the ST option.

VWBOXPLOT | VW
NORMALBOXPLOT | NORMAL
requests a box plot of Van der Waerden (normal) scores. This plot is associated with the Van der Waerden analysis, which you request with the VW or NORMAL option.

WILCOXONBOXPLOT | WILCOXON
requests a box plot of Wilcoxon scores. This plot is associated with the Wilcoxon analysis, which you request with the WILCOXON option.

If you request a plot but do not request the corresponding analysis, then PROC NPAR1WAY automatically invokes that analysis. For example, if you specify PLOTS=CONOVERBOXPLOT but do not also specify the CONOVER option in the PROC NPAR1WAY statement, PROC NPAR1WAY produces the Conover scores analysis as well as the requested box plot.

Plot-Options

The following *plot-options* are available for individual plots:

NOSTATS
suppresses the *p*-values that are displayed on the plot by default.

STATS
displays *p*-values on the plot. This is the default.

You cannot specify both STATS and NOSTATS as *plot-options* for the same plot.

If you specify NOSTATS as a *global-plot-option*, specifying STATS as an individual *plot-option* overrides the *global-plot-option* for the individual plot and displays statistics on the plot.

SAVAGE
requests an analysis of Savage scores. See the section "Savage Scores" on page 4471 for more information.

SCORES=DATA
requests an analysis that uses input data as scores. This option gives you the flexibility to construct any scores for your data with the DATA step and then analyze these scores with PROC NPAR1WAY. See the section "Scores for Linear Rank and One-Way ANOVA Tests" on page 4470 for more information. Using the SCORES=DATA option for raw (unscored) two-sample data produces a permutation test known as Pitman's test.

ST
requests an analysis of Siegel-Tukey scores. See the section "Siegel-Tukey Scores" on page 4471 for more information.

VW
NORMAL
requests an analysis of Van der Waerden (normal) scores. See the section "Van der Waerden (Normal) Scores" on page 4471 for more information.

WILCOXON
requests an analysis of Wilcoxon scores. When there are two classification levels, or two samples, this option produces the Wilcoxon rank-sum test. For any number of classification levels, this option produces the Kruskal-Wallis test. See the section "Wilcoxon Scores" on page 4470 for more information.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC NPAR1WAY to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the NPAR1WAY procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variable* **;**

The CLASS statement, which is required, names one and only one classification variable. The variable can be character or numeric. The CLASS variable identifies groups (or samples) in the data, and PROC NPAR1WAY provides analyses to examine differences among these groups. There can be two or more groups in the data.

EXACT Statement

EXACT *statistic-options* < / *computation-options* > **;**

The EXACT statement requests exact tests for the specified statistics. Optionally, PROC NPAR1WAY computes Monte Carlo estimates of the exact *p*-values. The *statistic-options* specify the statistics to provide exact tests for. The *computation-options* specify options for the computation of exact statistics. See the section "Exact Tests" on page 4476 for details

CAUTION: PROC NPAR1WAY computes exact tests with fast and efficient algorithms that are superior to direct enumeration. Exact tests are appropriate when a data set is small, sparse, skewed, or heavily tied. For some large problems, computation of exact tests might require a large amount

of time and memory. Consider using asymptotic tests for such problems. Alternatively, when asymptotic methods might not be sufficient for such large problems, consider using Monte Carlo estimation of exact *p*-values. See the section "Computational Resources" on page 4478 for more information.

Statistic-Options

The *statistic-options* specify the statistics to provide exact tests for.

Exact *p*-values are available for all nonparametric tests of location and scale differences produced by PROC NPAR1WAY. These include tests based on the following scores: Wilcoxon, median, Van der Waerden (normal), Savage, Siegel-Tukey, Ansari-Bradley, Klotz, Mood, and Conover. Additionally, exact *p*-values are available for tests that use the raw input data as scores. The procedure computes exact *p*-values when the data are classified into two levels (two-sample tests) and when the data are classified into more than two levels (multisample tests). Two-sample tests are based on simple linear rank statistics. Multisample tests are based on one-way ANOVA statistics. See the section "Exact Tests" on page 4476 for details.

Exact *p*-values are also available for the two-sample Kolmogorov-Smirnov test. Additionally, exact confidence limits are available for the Hodges-Lehmann estimate of location shift. See the section "Hodges-Lehmann Estimation of Location Shift" on page 4472 for details.

Table 62.3 lists the available *statistic-options* and the exact tests computed. The option names are identical to the corresponding options in the PROC NPAR1WAY statement and the OUTPUT statement.

If you list no *statistic-options* in the EXACT statement, then PROC NPAR1WAY computes all the available exact *p*-values for those tests requested in the PROC NPAR1WAY statement.

Table 62.3 EXACT Statement *Statistic-Options*

Statistic-Option	**Exact Test Computed**
AB	Ansari-Bradley Test
CONOVER	Conover Test
HL	Hodges-Lehmann Confidence Limits
KLOTZ	Klotz Test
KS | EDF	Two-Sample Kolmogorov-Smirnov Test
MEDIAN	Median Test
MOOD	Mood Test
SAVAGE	Savage Test
SCORES=DATA	Test with Input Data as Scores
ST	Siegel-Tukey Test
VW | NORMAL	Van der Waerden (Normal Scores) Test
WILCOXON	Wilcoxon Test for Two-Sample Data
	Kruskal-Wallis Test for Multisample Data

Computation-Options

The *computation-options* specify options for computation of exact statistics. You can specify the following *computation-options* in the EXACT statement:

ALPHA=α

specifies the level of the confidence limits for Monte Carlo p-value estimates. The value of α must be between 0 and 1, and the default is 0.01. A confidence level of α produces $100(1-\alpha)\%$ confidence limits. The default of ALPHA=0.01 produces 99% confidence limits for the Monte Carlo estimates.

The ALPHA= option invokes the MC option.

MAXTIME=*value*

specifies the maximum clock time (in seconds) that PROC NPAR1WAY can use to compute an exact p-value. If the procedure does not complete the computation within the specified time, the computation terminates. The value of MAXTIME= must be a positive number. The MAXTIME= option is valid for Monte Carlo estimation of exact p-values, as well as for direct exact p-value computation. See the section "Computational Resources" on page 4478 for more information.

MC

requests Monte Carlo estimation of exact p-values, instead of direct exact p-value computation. Monte Carlo estimation can be useful for large problems that require a great amount of time and memory for exact computations but for which asymptotic approximations might not be sufficient. See the section "Monte Carlo Estimation" on page 4479 for more information.

The MC option is available for all the EXACT statement *statistic-options* except the HL option, which produces exact Hodges-Lehmann confidence limits. The ALPHA=, N=, and SEED= options also invoke the MC option.

N=*n*

specifies the number of samples for Monte Carlo estimation. The value of n must be a positive integer, and the default is 10,000 samples. Larger values of n produce more precise estimates of exact p-values. Because larger values of n generate more samples, the computation time increases.

The N= option invokes the MC option.

POINT

requests exact point probabilities for the test statistics.

The POINT option is available for all the EXACT statement *statistic-options* except the HL option, which produces exact Hodges-Lehmann confidence limits. The POINT option is not available with the MC option.

SEED=*number*

specifies the initial seed for random number generation for Monte Carlo estimation. The value of the SEED= option must be an integer. If you do not specify the SEED= option or if the SEED= value is negative or zero, PROC NPAR1WAY uses the time of day from the computer's clock to obtain the initial seed.

The SEED= option invokes the MC option.

FREQ Statement

FREQ *variable* **;**

The FREQ statement names a numeric variable that provides a frequency for each observation in the input data set. If you use a FREQ statement, PROC NPAR1WAY assumes that an observation occurs *n* times, where *n* is the value of the FREQ variable for the observation. The sum of the FREQ variable values represents the total number of observations, and the analysis is based on this expanded number of observations.

If the value of the FREQ variable is missing or is less than one, PROC NPAR1WAY does not use that observation in the analysis. If the value of the FREQ variable is not an integer, PROC NPAR1WAY uses only the integer portion as the frequency of the observation.

OUTPUT Statement

OUTPUT < **OUT=***SAS-data-set* > < *options* > **;**

The OUTPUT statement creates a SAS data set that contains statistics computed by PROC NPAR1WAY. You specify which statistics to store in the output data set by using options identical to those used in the PROC NPAR1WAY statement. The output data set contains one observation for each analysis variable named in the VAR statement. For more information about the contents of the output data set, see the section "Output Data Set" on page 4480.

Note that you can use the Output Delivery System (ODS) to create a SAS data set from any piece of PROC NPAR1WAY output. For more information, see the section "ODS Table Names" on page 4490 and Chapter 20, "Using the Output Delivery System."

You can specify the following options in the OUTPUT statement:

OUT=*SAS-data-set*
names the output data set. If you omit the OUT= option, the data set is named DATA*n*, where *n* is the smallest integer that makes the name unique.

options
specifies the statistics you want in the output data set. The options are identical to those used in the PROC NPAR1WAY statement to request analyses. Table 62.4 shows the available options. When you specify one of these options in the OUTPUT statement, the output data set contains all statistics from that analysis. See the section "Output Data Set" on page 4480 for a list of the output data set variables corresponding to each option.

If you do not specify any statistics options in the OUTPUT statement, then the output data set includes statistics from all analyses specified in the PROC NPAR1WAY statement.

Table 62.4 OUTPUT Statement Options

Option	Output Data Set Statistics
AB	Ansari-Bradley Test
ANOVA	Analysis of Variance
CONOVER	Conover Test
EDF	Kolmogorov-Smirnov Test
	Cramer-von Mises Test
	Kuiper Test for Two-Sample Data
HL	Hodges-Lehmann Estimates
KLOTZ	Klotz Test
MEDIAN	Median Test
MOOD	Mood Test
SAVAGE	Savage Test
SCORES=DATA	Test with Input Data as Scores
ST	Siegel-Tukey Test
VW | NORMAL	Van der Waerden (Normal Scores) Test
WILCOXON	Wilcoxon Test for Two-Sample Data
	Kruskal-Wallis Test

VAR Statement

VAR *variables* **;**

The VAR statement names the response or dependent variables to be included in the analysis. These variables must be numeric. If you omit the VAR statement, the procedure includes all numeric variables in the data set except for the CLASS variable, the FREQ variable, and the BY variables.

Details: NPAR1WAY Procedure

Missing Values

If an observation has a missing value for a response variable, PROC NPAR1WAY excludes that observation from the analysis.

By default, PROC NPAR1WAY excludes observations with missing values of the CLASS variable. If you specify the MISSING option, PROC NPAR1WAY treats missing values of the CLASS variable as a valid class level and includes these observations in the analysis.

PROC NPAR1WAY treats missing BY variable values like any other BY variable value. The missing values form a separate BY group. When a value of the FREQ variable is missing, PROC NPAR1WAY excludes the observation from the analysis.

Tied Values

Tied values occur when two or more observations are equal, whether the observations occur in the same sample or in different samples. In theory, nonparametric tests were developed for continuous distributions where the probability of a tie is zero. In practice, however, ties often occur. PROC NPAR1WAY uses the same method to handle ties for all score types. The procedure computes the scores as if there were no ties, averages the scores for tied observations, and assigns this average score to each observation with the same value.

When there are tied values, PROC NPAR1WAY first sorts the observations in ascending order and assigns ranks as if there were no ties. Then the procedure computes the scores based on these ranks by using the formula for the specified score type. The procedure averages the scores for tied observations and assigns this average score to each of the tied observations. Thus, all equal data values have the same score value. PROC NPAR1WAY then computes the test statistic from these scores.

Note that the asymptotic tests might be less accurate when the distribution of the data is heavily tied. For such data, it might be appropriate to use the exact tests provided by PROC NPAR1WAY as described in the section "Exact Tests" on page 4476.

When computing empirical distribution function statistics for data with ties, PROC NPAR1WAY uses the formulas given in the section "Tests Based on the Empirical Distribution Function" on page 4474. No special handling of ties is necessary.

Note that PROC NPAR1WAY bases its computations on the internal numeric values of the analysis variables; the procedure does not format or round these values before analysis. When values differ in their internal representation, even slightly, PROC NPAR1WAY does not treat them as tied values. If this is a concern for your data, then round the analysis variables by an appropriate amount before invoking PROC NPAR1WAY. For information about the ROUND function, see the discussion in *SAS Language Reference: Dictionary*.

Statistical Computations

Simple Linear Rank Tests for Two-Sample Data

Statistics of the form

$$S = \sum_{j=1}^{n} c_j \, a(R_j)$$

are called *simple linear rank statistics*, where

R_j is the rank of observation j

$a(R_j)$ is the score based on the rank of observation j

c_j is an indicator variable denoting the class to which the jth observation belongs

n is the total number of observations

For two-sample data (where the observations are classified into two levels), PROC NPAR1WAY calculates simple linear rank statistics for the scores that you specify. The section "Scores for Linear Rank and One-Way ANOVA Tests" on page 4470 describes the available scores, which you can use to test for differences in location and differences in scale.

To compute the linear rank statistic S, PROC NPAR1WAY sums the scores of the observations in the smaller of the two samples. If both samples have the same number of observations, PROC NPAR1WAY sums those scores for the sample that appears first in the input data set.

For each score that you specify, PROC NPAR1WAY computes an asymptotic test of the null hypothesis of no difference between the two classification levels. Exact tests are also available for these two-sample linear rank statistics. PROC NPAR1WAY computes exact tests for each score type that you specify in the EXACT statement. See the section "Exact Tests" on page 4476 for details.

To compute an asymptotic test for a linear rank sum statistic, PROC NPAR1WAY uses a standardized test statistic z, which has an asymptotic standard normal distribution under the null hypothesis. The standardized test statistic is computed as

$$z = (S - \mathrm{E}_0[S]) \,/\, \sqrt{\mathrm{Var}_0[S]}$$

where $\mathrm{E}_0[S]$ is the expected value of S under the null hypothesis, and $\mathrm{Var}_0[S]$ is the variance under the null hypothesis. As shown in Randles and Wolfe (1979),

$$\mathrm{E}_0[S] = \frac{n_1}{n} \sum_{j=1}^{n} a(R_j)$$

where n_1 is the number of observations in the first (smaller) class level or sample, n_2 is the number of observations in the other class level, and

$$\mathrm{Var}_0[S] = \frac{n_1 n_2}{n(n-1)} \sum_{j=1}^{n} \left(a(R_j) - \bar{a}\right)^2$$

where $\bar{a}$ is the average score,

$$\bar{a} = \frac{1}{n} \sum_{j=1}^{n} a(R_j)$$

Definition of p-Values

PROC NPAR1WAY computes one-sided and two-sided asymptotic p-values for each two-sample linear rank test. When the test statistic z is greater than its null hypothesis expected value of zero, PROC NPAR1WAY computes the right-sided p-value, which is the probability of a larger value of the statistic occurring under the null hypothesis. When the test statistic is less than or equal to zero, PROC NPAR1WAY computes the left-sided p-value, which is the probability of a smaller value of the statistic occurring under the null hypothesis. The one-sided p-value $P_1(z)$ can be expressed as

$$P_1(z) = \begin{cases} \mathrm{Prob}(Z > z) & \text{if } z > 0 \\ \mathrm{Prob}(Z < z) & \text{if } z \leq 0 \end{cases}$$

where Z has a standard normal distribution. The two-sided p-value $P_2(z)$ is computed as

$$P_2(z) = \mathrm{Prob}(|Z| > |z|)$$

Continuity Correction

PROC NPAR1WAY uses a continuity correction for the asymptotic two-sample Wilcoxon and Siegel-Tukey tests by default. You can remove the continuity correction by specifying the CORRECT=NO option. PROC NPAR1WAY incorporates the continuity correction when computing the standardized test statistic z by subtracting 0.5 from the numerator $S - \mathrm{E}_0(S)$ if it is greater than zero. If the numerator is less than zero, PROC NPAR1WAY adds 0.5. Some sources recommend a continuity correction for nonparametric tests that use a continuous distribution to approximate a discrete distribution. (See Sheskin 1997.)

If you specify CORRECT=NO, PROC NPAR1WAY does not use a continuity correction for any test.

One-Way ANOVA Tests

PROC NPAR1WAY computes a one-way ANOVA test for each score type that you specify. Under the null hypothesis of no difference among class levels (or samples), this test statistic has an asymptotic chi-square distribution with $r - 1$ degrees of freedom, where r is the number of class levels. For Wilcoxon scores, this test is known as the Kruskal-Wallis test.

Exact one-way ANOVA tests are also available for multisample data (where the data are classified into more than two levels). For two-sample data, exact simple linear rank tests are available. PROC NPAR1WAY computes exact tests for each score type that you specify in the EXACT statement. See the section "Exact Tests" on page 4476 for details about exact tests.

PROC NPAR1WAY computes the one-way ANOVA test statistic as

$$C = \left(\sum_{i=1}^{r} (T_i - \mathrm{E}_0[T_i])^2 \; / \; n_i \right) \; / \; S^2$$

where T_i is the total of scores for class level i, $\mathrm{E}_0[T_i]$ is the expected total for level i under the null hypothesis of no difference among levels, n_i is the number of observations in level i, and S^2 is the sample variance of the scores. The total of scores for class level i is given by

$$T_i = \sum_{j=1}^{n} c_{ij} \; a(R_j)$$

where $a(R_j)$ is the score for observation j, and c_{ij} indicates whether observation j is in level i. The expected total of scores for class level i under the null hypothesis is equal to

$$\mathrm{E}_0[T_i] = \frac{n_i}{n} \sum_{j=1}^{n} a(R_j)$$

The sample variance of the scores is computed as

$$S^2 = \frac{1}{(n-1)} \sum_{j=1}^{n} \left(a(R_j) - \bar{a} \right)^2$$

where $\bar{a}$ is the average score,

$$\bar{a} = \frac{1}{n} \sum_{j=1}^{n} a(R_j)$$

Scores for Linear Rank and One-Way ANOVA Tests

For each score type that you specify, PROC NPAR1WAY computes a one-way ANOVA statistic and also a linear rank statistic for two-sample data. The following score types are used primarily to test for differences in location: Wilcoxon, median, Van der Waerden (normal), and Savage. The following scores types are used to test for scale differences: Siegel-Tukey, Ansari-Bradley, Klotz, and Mood. Conover scores can be used to test for differences in both location and scale. This section gives formulas for the score types available in PROC NPAR1WAY. For further information about the formulas and the applicability of each score, see Randles and Wolfe (1979), Gibbons and Chakraborti (1992), Conover (1999), and Hollander and Wolfe (1999).

In addition to the score types described in this section, you can specify the SCORES=DATA option to use the input data observations as scores. This enables you to produce a wide variety of tests. You can construct any scores by using the DATA step, and then PROC NPAR1WAY computes the corresponding linear rank and one-way ANOVA tests. You can also analyze the raw data with the SCORES=DATA option; for two-sample data, this permutation test is known as Pitman's test.

Wilcoxon Scores

Wilcoxon scores are the ranks of the observations, which can be written as

$$a(R_j) = R_j$$

where R_j is the rank of observation j, and $a(R_j)$ is the score of observation j.

Using Wilcoxon scores in the linear rank statistic for two-sample data produces the rank sum statistic of the Mann-Whitney-Wilcoxon test. Using Wilcoxon scores in the one-way ANOVA statistic produces the Kruskal-Wallis test. Wilcoxon scores are locally most powerful for location shifts of a logistic distribution.

When computing the asymptotic Wilcoxon two-sample test, PROC NPAR1WAY uses a continuity correction by default, as described in the section "Continuity Correction" on page 4469. If you specify the CORRECT=NO option in the PROC NPAR1WAY statement, the procedure does not use a continuity correction.

Median Scores

Median scores equal 1 for observations greater than the median, and 0 otherwise. In terms of the observation ranks, median scores are defined as

$$a(R_j) = \begin{cases} 1 & \text{if } R_j > (n+1)/2 \\ 0 & \text{if } R_j \leq (n+1)/2 \end{cases}$$

Using median scores in the linear rank statistic for two-sample data produces the two-sample median test. The one-way ANOVA statistic with median scores is equivalent to the Brown-Mood test. Median scores are particularly powerful for distributions that are symmetric and heavy-tailed.

Van der Waerden (Normal) Scores

Van der Waerden scores are the quantiles of a standard normal distribution and are also known as *quantile normal scores*. Van der Waerden scores are computed as

$$a(R_j) = \Phi^{-1}\left(\frac{R_j}{n+1}\right)$$

where Φ is the cumulative distribution function of a standard normal distribution. These scores are powerful for normal distributions.

Savage Scores

Savage scores are expected values of order statistics from the exponential distribution, with 1 subtracted to center the scores around 0. Savage scores are computed as

$$a(R_j) = \sum_{i=1}^{R_j}\left(\frac{1}{n-i+1}\right) - 1$$

Savage scores are powerful for comparing scale differences in exponential distributions or location shifts in extreme value distributions (Hajek 1969, p. 83).

Siegel-Tukey Scores

Siegel-Tukey scores are defined as

$$\begin{array}{llll} a(1)=1, & a(n)=2, & a(n-1)=3, & a(2)=4, \\ a(3)=5, & a(n-2)=6, & a(n-3)=7, & a(4)=8, \ \ldots \end{array}$$

where the score values continue to increase in this pattern toward the middle ranks until all observations have been assigned a score.

When computing the asymptotic Siegel-Tukey two-sample test, PROC NPAR1WAY uses a continuity correction by default, as described in the section "Continuity Correction" on page 4469. If you specify the CORRECT=NO option in the PROC NPAR1WAY statement, the procedure does not use a continuity correction.

Ansari-Bradley Scores

Ansari-Bradley scores are similar to Siegel-Tukey scores, but Ansari-Bradley scoring assigns the same score value to corresponding extreme ranks. (Siegel-Tukey scores are a permutation of the ranks $1, 2, \ldots, n$.) Ansari-Bradley scores are defined as

$$\begin{array}{ll} a(1)=1, & a(n)=1, \\ a(2)=2, & a(n-1)=2, \ \ldots \end{array}$$

Equivalently, Ansari-Bradley scores are equal to

$$a(R_j) = \frac{n+1}{2} - \left|R_j - \frac{n+1}{2}\right|$$

Klotz Scores

Klotz scores are the squares of the Van der Waerden (normal) scores. Klotz scores are computed as

$$a(R_j) = \left(\Phi^{-1}\left(\frac{R_j}{n+1}\right)\right)^2$$

where Φ is the cumulative distribution function of a standard normal distribution.

Mood Scores

Mood scores are computed as the square of the difference between the observation rank and the average rank. Mood scores can be written as

$$a(R_j) = \left(R_j - \frac{n+1}{2}\right)^2$$

Conover Scores

Conover scores are based on the squared ranks of the absolute deviations from the sample means. For observation j the absolute deviation from the mean is computed as

$$U_j = |X_{j(i)} - \bar{X}_i|$$

where $X_{j(i)}$ is the value of observation j, observation j belongs to sample i, and $\bar{X}_i$ is the mean of sample i. The values of U_j are ranked, and the Conover score for observation j is computed as

$$\text{Score}_j = \left(\text{Rank}(U_j)\right)^2$$

Following Conover (1999), when there are ties among the values of U_j, PROC NPAR1WAY assigns the average rank to each of the tied observations. To compute the average rank, PROC NPAR1WAY ranks the U_j as if there were no ties, and then averages the ranks of the tied observations.

The Conover score test is also known as the squared ranks test for variances. See Conover (1999) for more information.

Hodges-Lehmann Estimation of Location Shift

If you specify the HL option, PROC NPAR1WAY computes the Hodges-Lehmann estimate of location shift for two-sample data. PROC NPAR1WAY also provides confidence limits for the location shift. These confidence limits are sometimes called Moses confidence limits. You can set the level of the confidence limits with the ALPHA= option. The default is ALPHA=0.05, which produces 95% confidence limits. Additionally, you can request exact confidence limits for the location shift by specifying the HL option in the EXACT statement.

The Hodges-Lehmann estimator of location shift is associated with the Wilcoxon linear rank statistic. See Hollander and Wolfe (1999) and Hodges and Lehmann (1983) for details.

PROC NPAR1WAY computes the Hodges-Lehmann estimate $\hat{\Delta}$ as the median of all paired differences between observations in the two samples, which can be written as

$$\hat{\Delta} = \text{median}\left((Y_j - X_i) \quad \text{where} \quad j = 1, 2, \ldots, n_1;\ i = 1, 2, \ldots, n_2 \right)$$

The Y_j are observations in sample 1, the X_i are observations in sample 2, and n_1 and n_2 denote the number of observations in sample 1 and sample 2, respectively. PROC NPAR1WAY uses the smaller of the two samples as sample 1. If both samples have the same number of observations, PROC NPAR1WAY uses the sample that appears first in the input data set as sample 1. Sample 1 is the same sample that PROC NPAR1WAY uses to compute the two-sample linear rank statistic.

Let m denote the total number of differences, $n_1 \times n_2$, and let $U^{(k)}$ denote the kth value of $(Y_j - X_i)$ among the ordered differences. When m is an odd number, then the median difference is the value with rank $(m+1)/2$,

$$\hat{\Delta} = U^{(k)} \quad \text{where} \quad k = (m+1)/2$$

When m is an even number, the median difference is the average of the values with ranks $(m/2)$ and $(m/2)+1$,

$$\hat{\Delta} = \left(U^{(k)} + U^{(k+1)} \right) / 2 \quad \text{where} \quad k = m/2$$

Following Hollander and Wolfe (1999), the asymptotic lower and upper confidence limits for the location shift are

$$\left(\Delta_L = U^{(C_\alpha)},\quad \Delta_U = U^{(m+1-C_\alpha)} \right)$$

where C_α is the largest integer less than or equal to ${C_\alpha}^*$, which is computed as

$${C_\alpha}^* = \frac{n_1 n_2}{2} - z_{\alpha/2} \left(\frac{n_1 n_2 (n_1 + n_2 + 1)}{12} \right)^{1/2}$$

where $z_{\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the standard normal distribution.

PROC NPAR1WAY displays the midpoint of the confidence interval (Δ_L, Δ_U), which can also be used as an estimate of location shift. See Lehmann (1963) for details. Additionally, PROC NPAR1WAY provides an estimate of the asymptotic standard error of $\hat{\Delta}$ based on the length of the confidence interval, which is computed as

$$\text{ASE}[\hat{\Delta}] = (\Delta_U - \Delta_L) \ / \ (2\, z_{\alpha/2})$$

Exact Confidence Limits

If you specify the HL option in the EXACT statement, PROC NPAR1WAY computes exact confidence limits for the location shift between the two samples. As for the asymptotic confidence limits, you can set the confidence level with the ALPHA= option. The default is ALPHA=0.05, which produces 95% confidence limits.

PROC NPAR1WAY computes exact confidence limits for the location shift as described in Randles and Wolfe (1979, p. 180). PROC NPAR1WAY first generates the exact conditional distribution of

the Mann-Whitney U statistic, which equals the number of pairwise differences $(Y_i - X_j)$ that are positive, plus half the number of pairwise differences that are zero. The Mann-Whitney statistic is defined as

$$MW = \sum_{i=1}^{n_1} \sum_{j=i}^{n_2} \phi\left(X_i - Y_j\right)$$

where

$$\phi(Y_i, X_j) = \begin{cases} 1 & \text{if } Y_i > X_j \\ 1/2 & \text{if } Y_i = X_j \\ 0 & \text{otherwise} \end{cases}$$

From the exact conditional distribution of the Mann-Whitney statistic MW, PROC NPAR1WAY chooses $C^*_{L,\alpha}$ as the largest value such that Prob($MW >= C^*_{L,\alpha}$) $>= \alpha/2$. Rounding $C^*_{L,\alpha}$ up to the nearest integer $C_{L,\alpha}$, the lower confidence limit equals the difference $(Y_i - X_j)$ that has a rank of $(n_1 n_2 - C_{L,\alpha} + 1)$.

To find the upper confidence limit, PROC NPAR1WAY chooses $C^*_{U,\alpha}$ as the smallest value such that Prob($MW <= C^*_{U,\alpha}$) $>= \alpha/2$. Rounding $C^*_{U,\alpha}$ down to the nearest integer $C_{L,\alpha}$, the upper confidence limit equals the difference $(Y_i - X_j)$ that has a rank of $(n_1 n_2 - C_{U,\alpha})$.

Because this is a discrete problem, the confidence coefficient for these exact confidence limits is not exactly $1 - \alpha$ but is at least $1 - \alpha$. Thus, these confidence limits are conservative.

Tests Based on the Empirical Distribution Function

If you specify the EDF option, PROC NPAR1WAY computes tests based on the empirical distribution function. These include the Kolmogorov-Smirnov and Cramer-von Mises tests, and also the Kuiper test for two-sample data. This section gives formulas for these test statistics. For further information about the formulas and the interpretation of EDF statistics, see Hollander and Wolfe (1999) and Gibbons and Chakraborti (1992). For details about the k-sample analogs of the Kolmogorov-Smirnov and Cramer-von Mises statistics used by NPAR1WAY, see Kiefer (1959).

The *empirical distribution function* (EDF) of a sample $\{x_j\}$, $j = 1, 2, \ldots, n$, is defined as

$$F(x) = \frac{1}{n}(\text{number of } x_j \leq x) = \frac{1}{n} \sum_{j=1}^{n} I(x_j \leq x)$$

where $I(\cdot)$ is an indicator function. PROC NPAR1WAY uses the subsample of values within the ith class level to generate an EDF for the class, F_i. The EDF for the overall sample, pooled over classes, can also be expressed as

$$F(x) = \frac{1}{n} \sum_{i} (n_i F_i(x))$$

where n_i is the number of observations in the ith class level, and n is the total number of observations.

Kolmogorov-Smirnov Test

The Kolmogorov-Smirnov statistic measures the maximum deviation of the EDF within the classes from the pooled EDF. PROC NPAR1WAY computes the Kolmogorov-Smirnov statistic as

$$KS = \max_j \sqrt{\frac{1}{n} \sum_i n_i \left(F_i(x_j) - F(x_j)\right)^2} \quad \text{where} \quad j = 1, 2, \ldots, n$$

The asymptotic Kolmogorov-Smirnov statistic is computed as

$$KS_a = KS \times \sqrt{n}$$

For each class level i and overall, PROC NPAR1WAY displays the value of F_i at the maximum deviation from F and the value $\sqrt{n_i}\,(F_i - F)$ at the maximum deviation from F. PROC NPAR1WAY also gives the observation where the maximum deviation occurs.

If there are only two class levels, PROC NPAR1WAY computes the two-sample Kolmogorov-Smirnov test statistic D as

$$D = \max_j \left|F_1(x_j) - F_2(x_j)\right| \quad \text{where} \quad j = 1, 2, \ldots, n$$

The p-value for this test is the probability that D is greater than the observed value d under the null hypothesis of no difference between class levels or samples. PROC NPAR1WAY computes the asymptotic p-value for D with the approximation

$$\text{Prob}(D > d) = 1 - 2 \sum_{i=1}^{\infty} (-1)^{(i-1)}\, e^{(-2i^2\, z^2)}$$

where

$$z = d \sqrt{n_1\, n_2\, /\, n}$$

The quality of this approximation has been studied by Hodges (1957).

If you specify the D option, or if you request exact Kolmogorov-Smirnov p-values with the KS option in the EXACT statement, PROC NPAR1WAY also computes the one-sided Kolmogorov-Smirnov statistics $D+$ and $D-$ for two-sample data as

$$D+ = \max_j \left(F_1(x_j) - F_2(x_j)\right) \quad \text{where} \quad j = 1, 2, \ldots, n$$

$$D- = \max_j \left(F_2(x_j) - F_1(x_j)\right) \quad \text{where} \quad j = 1, 2, \ldots, n$$

The asymptotic probability that $D+$ is greater than the observed value d^+, under the null hypothesis of no difference between the two class levels, is computed as

$$\text{Prob}(D+ > d^+) = e^{-2z^2} \quad \text{where} \quad z = d^+ \sqrt{n_1\, n_2\, /\, n}$$

Similarly, the asymptotic probability that $D-$ is greater than the observed value d^- is computed as

$$\text{Prob}(D- > d^-) = e^{-2z^2} \quad \text{where} \quad z = d^- \sqrt{n_1\, n_2\, /\, n}$$

To request exact p-values for the Kolmogorov-Smirnov statistics, you can specify the KS option in the EXACT statement. See the section "Exact Tests" on page 4476 for more information.

Cramer-von Mises Test

The Cramer-von Mises statistic is defined as

$$CM = \frac{1}{n^2} \sum_i \left(n_i \sum_{j=1}^{p} t_j \left(F_i(x_j) - F(x_j) \right)^2 \right)$$

where t_j is the number of ties at the jth distinct value and p is the number of distinct values. The asymptotic value is computed as

$$CM_a = CM \times n$$

PROC NPAR1WAY displays the contribution of each class level to the sum CM_a.

Kuiper Test

For data with two class levels, PROC NPAR1WAY computes the Kuiper statistic, its scaled value for the asymptotic distribution, and the asymptotic p-value. The Kuiper statistic is computed as

$$K = \max_j \left(F_1(x_j) - F_2(x_j) \right) - \min_j \left(F_1(x_j) - F_2(x_j) \right) \quad \text{where} \quad j = 1, 2, \ldots, n$$

The asymptotic value is

$$K_a = K \sqrt{n_1 \, n_2 \, / \, n}$$

PROC NPAR1WAY displays the value of $(\max_j \left| F_1(x_j) - F_2(x_j) \right|)$ for each class level.

The p-value for the Kuiper test is the probability of observing a larger value of K_a under the null hypothesis of no difference between the two classes. PROC NPAR1WAY computes this p-value according to Owen (1962, p. 441).

Exact Tests

PROC NPAR1WAY provides exact p-values for tests for location and scale differences based on the following scores: Wilcoxon, median, van der Waerden (normal), Savage, Siegel-Tukey, Ansari-Bradley, Klotz, Mood, and Conover. Additionally, PROC NPAR1WAY provides exact p-values for tests that use the raw data as scores. Exact tests are available for two-sample and multisample data. When the data are classified into two samples, tests are based on simple linear rank statistics. When the data are classified into more than two samples, tests are based on one-way ANOVA statistics.

Exact tests can be useful in situations where the asymptotic assumptions are not met and the asymptotic p-values are not close approximations for the true p-values. Standard asymptotic methods involve the assumption that the test statistic follows a particular distribution when the sample size is sufficiently large. When the sample size is not large, asymptotic results might not be valid, with the asymptotic p-values differing perhaps substantially from the exact p-values. Asymptotic results might also be unreliable when the distribution of the data is sparse, skewed, or heavily tied. See

Agresti (2007) and Bishop, Fienberg, and Holland (1975). Exact computations are based on the statistical theory of exact conditional inference for contingency tables, reviewed by Agresti (1992).

In addition to computation of exact *p*-values, PROC NPAR1WAY provides the option of estimating exact *p*-values by Monte Carlo simulation. This can be useful for problems that are so large that exact computations require a great amount of time and memory, but for which asymptotic approximations might not be sufficient.

The following sections summarize the exact computational algorithms, define the exact *p*-values that PROC NPAR1WAY computes, discuss the computational resource requirements, and describe the Monte Carlo estimation option.

Computational Algorithms

PROC NPAR1WAY computes exact *p*-values by using the network algorithm developed by Mehta and Patel (1983). This algorithm provides a substantial advantage over direct enumeration, which can be very time-consuming and feasible only for small problems. See Agresti (1992) for a review of algorithms for computation of exact *p*-values, and see Mehta, Patel, and Tsiatis (1984) and Mehta, Patel, and Senchaudhuri (1991) for information about the performance of the network algorithm.

PROC NPAR1WAY constructs a contingency table from the input data, with rows formed by the levels of the classification variable and columns formed by the response variable values. The reference set for a given contingency table is the set of all contingency tables with the observed marginal row and column sums. Corresponding to this reference set, the network algorithm forms a directed acyclic network consisting of nodes in a number of stages. A path through the network corresponds to a distinct table in the reference set. The distances between nodes are defined so that the total distance of a path through the network is the corresponding value of the test statistic. At each node, the algorithm computes the shortest and longest path distances for all the paths that pass through that node. For the two-sample linear rank statistics, which can be expressed as a linear combination of cell frequencies multiplied by increasing row and column scores, PROC NPAR1WAY computes shortest and longest path distances by using the algorithm given by Agresti, Mehta, and Patel (1990). For the multisample one-way test statistics, PROC NPAR1WAY computes an upper bound for the longest path and a lower bound for the shortest path by following the approach of Valz and Thompson (1994).

The longest and shortest path distances or bounds for a node are compared to the value of the test statistic to determine whether all paths through the node contribute to the *p*-value, none of the paths through the node contribute to the *p*-value, or neither of these situations occurs. If all paths through the node contribute, the *p*-value is incremented accordingly, and these paths are eliminated from further analysis. If no paths contribute, these paths are eliminated from the analysis. Otherwise, the algorithm continues, still processing this node and the associated paths. The algorithm finishes when all nodes have been accounted for.

In applying the network algorithm, PROC NPAR1WAY uses full numerical precision to represent all statistics, row and column scores, and other quantities involved in the computations. Although it is possible to use rounding to improve the speed and memory requirements of the algorithm, PROC NPAR1WAY does not do this since it can result in reduced accuracy of the *p*-values.

Definition of p-Values

For two-sample linear rank tests, PROC NPAR1WAY computes exact one-sided and two-sided p-values for each test specified in the EXACT statement. For the one-sided test, PROC NPAR1WAY displays the right-sided p-value when the observed value of the test statistic is greater than its expected value. The right-sided p-value is the sum of probabilities for those tables having a test statistic greater than or equal to the observed test statistic. Otherwise, when the test statistic is less than or equal to its expected value, PROC NPAR1WAY displays the left-sided p-value. The left-sided p-value is the sum of probabilities for those tables having a test statistic less than or equal to the one observed. The one-sided p-value P_1 can be expressed as

$$P_1(t) = \begin{cases} \text{Prob}(\text{ Test Statistic} \geq t\ ) & \text{if } \ t > \mathrm{E}_0(T) \\ \text{Prob}(\text{ Test Statistic} \leq t\ ) & \text{if } \ t \leq \mathrm{E}_0(T) \end{cases}$$

where t is the observed value of the test statistic and $\mathrm{E}_0(T)$ is the expected value of the test statistic under the null hypothesis. PROC NPAR1WAY computes the two-sided p-value as the sum of the one-sided p-value and the corresponding area in the opposite tail of the distribution of the statistic, equidistant from the expected value. The two-sided p-value P_2 can be expressed as

$$P_2(t) = \text{Prob}\left(|\text{Test Statistic} - \mathrm{E}_0(T)| \geq |t - \mathrm{E}_0(T)|\right)$$

For multisample data, the tests are based on one-way ANOVA statistics. For a test of this form, large values of the test statistic indicate a departure from the null hypothesis; the test is inherently two-sided. The exact p-value is the sum of probabilities for those tables having a test statistic greater than or equal to the value of the observed test statistic.

If you specify the POINT option in the EXACT statement, PROC NPAR1WAY also displays exact point probabilities for the test statistics. The exact point probability is the exact probability that the test statistic equals the observed value.

Computational Resources

PROC NPAR1WAY uses relatively fast and efficient algorithms for exact computations. These recently developed algorithms, together with improvements in computer power, now make it feasible to perform exact computations for data sets where previously only asymptotic methods could be applied. Nevertheless, there are still large problems that might require a prohibitive amount of time and memory for exact computations, depending on the speed and memory available on your computer. For large problems, consider whether exact methods are really needed or whether asymptotic methods might give results quite close to the exact results while requiring much less computer time and memory. When asymptotic methods might not be sufficient for such large problems, consider using Monte Carlo estimation of exact p-values, as described in the section "Monte Carlo Estimation" on page 4479.

A formula does not exist that can predict in advance how much time and memory are needed to compute an exact p-value for a certain problem. The time and memory required depend on several factors, including which test is being performed, the total sample size, the number of rows and columns, and the specific arrangement of the observations into table cells. Generally, larger problems (in terms of total sample size, number of rows, and number of columns) tend to require more time and memory. Additionally, for a fixed total sample size, time and memory requirements tend

to increase as the number of rows and columns increase, since this corresponds to an increase in the number of tables in the reference set. Also for a fixed sample size, time and memory requirements increase as the marginal row and column totals become more homogeneous. See Agresti, Mehta, and Patel (1990) and Gail and Mantel (1977) for details.

At any time while PROC NPAR1WAY is computing exact p-values, you can terminate the computations by pressing the system interrupt key sequence (see the *SAS Companion* for your system) and choosing to stop computations. After you terminate exact computations, PROC NPAR1WAY completes all other remaining tasks. The procedure produces the requested output and reports missing values for any exact p-values not computed by the time of termination.

You can also use the MAXTIME= option in the EXACT statement to limit the amount of time PROC NPAR1WAY uses for exact computations. You specify a MAXTIME= value that is the maximum amount of time (in seconds) that PROC NPAR1WAY can use to compute an exact p-value. If PROC NPAR1WAY does not finish computing the exact p-value within that time, it terminates the computation and completes all other remaining tasks.

Monte Carlo Estimation

If you specify the MC option in the EXACT statement, PROC NPAR1WAY computes Monte Carlo estimates of the exact p-values instead of directly computing the exact p-values. Monte Carlo estimation can be useful for large problems that require a great amount of time and memory for exact computations but for which asymptotic approximations might not be sufficient. To describe the precision of each Monte Carlo estimate, PROC NPAR1WAY provides the asymptotic standard error and $100(1-\alpha)\%$ confidence limits. The confidence level α is determined by the ALPHA= option in the EXACT statement, which, by default, equals 0.01 and produces 99% confidence limits. The N= option in the EXACT statement specifies the number of samples PROC NPAR1WAY uses for Monte Carlo estimation; the default is 10,000 samples. You can specify a larger value for n to improve the precision of the Monte Carlo estimates. Because larger values of n generate more samples, the computation time increases. Or you can specify a smaller value of n to reduce the computation time.

To compute a Monte Carlo estimate of an exact p-value, PROC NPAR1WAY generates a random sample of tables with the same total sample size, row totals, and column totals as the observed table. PROC NPAR1WAY uses the algorithm of Agresti, Wackerly, and Boyett (1979), which generates tables in proportion to their hypergeometric probabilities conditional on the marginal frequencies. For each sample table, PROC NPAR1WAY computes the value of the test statistic and compares it to the value for the observed table. When estimating a right-sided p-value, PROC NPAR1WAY counts all sample tables for which the test statistic is greater than or equal to the observed test statistic. Then the p-value estimate equals the number of these tables divided by the total number of tables sampled, which can be written as

$$\begin{aligned} \hat{P}_{MC} &= M \;/\; N \\ M &= \text{number of samples with (Test Statistic} \geq t) \\ N &= \text{total number of samples} \\ t &= \text{observed Test Statistic} \end{aligned}$$

PROC NPAR1WAY computes left-sided and two-sided p-value estimates in a similar manner. For left-sided p-values, PROC NPAR1WAY evaluates whether the test statistic for each sampled table is less than or equal to the observed test statistic. For two-sided p-values, PROC NPAR1WAY examines the sample test statistics according to the expression for $P_2(t)$ given in the section "Definition of p-Values" on page 4478.

The variable M is a binomial variable with N trials and success probability p. It follows that the asymptotic standard error of the Monte Carlo estimate is

$$\text{se}[\hat{P}_{MC}] = \sqrt{\hat{P}_{MC}\,(1 - \hat{P}_{MC})\;/\;(N-1)}$$

PROC NPAR1WAY constructs asymptotic confidence limits for the p-values according to

$$\hat{P}_{MC} \;\pm\; \left(z_{\alpha/2} \;\times\; \text{se}[\hat{P}_{MC}]\right)$$

where $z_{\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the standard normal distribution, and the confidence level α is determined by the ALPHA= option in the EXACT statement.

When the Monte Carlo estimate $\hat{P}_{MC}$ equals 0, PROC NPAR1WAY computes confidence limits for the p-value as

$$(\;0,\;\; 1-\alpha^{(1/N)}\;)$$

When the Monte Carlo estimate $\hat{P}_{MC}$ equals 1, PROC NPAR1WAY computes the confidence limits as

$$(\;\alpha^{(1/N)},\;\; 1\;)$$

Output Data Set

The OUTPUT statement creates a SAS data set that contains statistics computed by PROC NPAR1WAY. You specify which statistics to store in the output data set by using options identical to those in the PROC NPAR1WAY statement. When you specify one of these options in the OUTPUT statement, PROC NPAR1WAY includes all available statistics from that analysis in the output data set.

The output data set contains one observation for each analysis variable within a BY group. (You can name the analysis variables in the VAR statement.) The OUTPUT data set includes the following variables:

- BY variables, if you use a BY statement
- _VAR_, which identifies the analysis variable
- variables containing the specified statistics

Table 62.5 lists the variable names and descriptions for all available statistics. Note that some statistics are available only for the two-sample case (where the CLASS variable groups the data into two classes); other statistics are available only for the multisample case.

If you request exact tests by using the EXACT statement, the output data set also includes exact *p*-values for those tests when you specify the corresponding options in the OUTPUT statement. If you do not request exact tests with the EXACT statement, the output data set does not include exact *p*-values.

Monte Carlo estimates of exact *p*-values are not available in this output data set, but you can use the Output Delivery System (ODS) to store Monte Carlo estimates in a SAS data set. You can use the Output Delivery System to create a SAS data set from any piece of PROC NPAR1WAY output. For more information, see the section "ODS Table Names" on page 4490 and Chapter 20, "Using the Output Delivery System."

Table 62.5 Output Data Set Variable Names and Descriptions

Option	Output Variables		Variable Descriptions
AB	_AB_	*	Two-Sample Ansari-Bradley Statistic
	Z_AB	*	Ansari-Bradley Statistic, Standardized
	PL_AB	*	*p*-Value (Left-Sided), Ansari-Bradley Test
	PR_AB	*	*p*-Value (Right-Sided), Ansari-Bradley Test
	P2_AB	*	*p*-Value (Two-Sided), Ansari-Bradley Test
	XPL_AB	*	Exact *p*-Value (Left-Sided), Ansari-Bradley Test
	XPR_AB	*	Exact *p*-Value (Right-Sided), Ansari-Bradley Test
	XPT_AB	*	Exact Point Probability, Ansari-Bradley Test
	XP2_AB	*	Exact *p*-Value (Two-Sided), Ansari-Bradley Test
	CHAB		Ansari-Bradley Chi-Square
	DF_CHAB		Degrees of Freedom, Ansari-Bradley Chi-Square
	P_CHAB		*p*-Value, Ansari-Bradley Chi-Square Test
	XP_CHAB	**	Exact *p*-Value, Ansari-Bradley Chi-Square Test
	XPT_CHAB	**	Exact Point Probability, Ansari-Bradley Chi-Square
ANOVA	_MSA_		ANOVA Effect Mean Square, Among MS
	MSE		ANOVA Error Mean Square, Within MS
	_F_		*F* Statistic for ANOVA
	P_F		*p*-Value, *F* Statistic for ANOVA

Table 62.5 *continued*

Option	Output Variables		Variable Descriptions
CONOVER	_CON_	*	Two-Sample Conover Statistic
	Z_CON	*	Conover Statistic, Standardized
	PL_CON	*	*p*-Value (Left-Sided), Conover Test
	PR_CON	*	*p*-Value (Right-Sided), Conover Test
	P2_CON	*	*p*-Value (Two-Sided), Conover Test
	XPL_CON	*	Exact *p*-Value (Left-Sided), Conover Test
	XPR_CON	*	Exact *p*-Value (Right-Sided), Conover Test
	XPT_CON	*	Exact Point Probability, Conover Test
	XP2_CON	*	Exact *p*-Value (Two-Sided), Conover Test
	CHCON		Conover Chi-Square
	DF_CHCON		Degrees of Freedom, Conover Chi-Square
	P_CHCON		*p*-Value, Conover Chi-Square Test
	XP_CHCON	**	Exact *p*-Value, Conover Chi-Square Test
	XPT_CHCO	**	Exact Point Probability, Conover Chi-Square
EDF	_KS_		Kolmogorov-Smirnov Statistic
	KSA		Kolmogorov-Smirnov Statistic (Asymptotic)
	_Dp_	*	Two-Sample Kolmogorov-Smirnov D+
	P_Dp	*	*p*-Value, D+
	_Dm_	*	Two-Sample Kolmogorov-Smirnov D-
	P_Dm	*	*p*-Value, D-
	_D_	*	Two-Sample Kolmogorov-Smirnov Statistic D
	P_KSA	*	*p*-Value, D
	XP_Dp	*	Exact *p*-Value, D+
	XPT_Dp	*	Exact Point Probability, D+
	XP_Dm	*	Exact *p*-Value, D-
	XPT_Dm	*	Exact Point Probability, D-
	XP_D	*	Exact *p*-Value, D
	XPT_D	*	Exact Point Probability, D
	_CM_		Cramer-von Mises Statistic
	CMA		Cramer-von Mises Statistic (Asymptotic)
	_K_	*	Kuiper Two-Sample Statistic
	_KA_	*	Kuiper Two-Sample Statistic (Asymptotic)
	P_KA	*	*p*-Value, Two-Sample Kuiper Test
HL	_HL_	*	Hodges-Lehmann Estimate, Location Shift
	L_HL	*	Lower Confidence Limit, Hodges-Lehmann
	U_HL	*	Upper Confidence Limit, Hodges-Lehmann
	M_HL	*	Confidence Limit Midpoint, Hodges-Lehmann
	E_HL	*	ASE of Hodges-Lehmann Estimate
	XL_HL	*	Exact Lower Confidence Limit, Hodges-Lehmann
	XU_HL	*	Exact Upper Confidence Limit, Hodges-Lehmann
	XM_HL	*	Exact Confidence Limit Midpoint

Table 62.5 *continued*

Option	Output Variables		Variable Descriptions
KLOTZ	_KLOTZ_	*	Two-Sample Klotz Statistic
	Z_K	*	Klotz Statistic, Standardized
	PL_K	*	*p*-Value (Left-Sided), Klotz Test
	PR_K	*	*p*-Value (Right-Sided), Klotz Test
	P2_K	*	*p*-Value (Two-Sided), Klotz Test
	XPL_K	*	Exact *p*-Value (Left-Sided), Klotz Test
	XPR_K	*	Exact *p*-Value (Right-Sided), Klotz Test
	XPT_K	*	Exact Point Probability, Klotz Test
	XP2_K	*	Exact *p*-Value (Two-Sided), Klotz Test
	CHK		Klotz Chi-Square
	DF_CHK		Degrees of Freedom, Klotz Chi-Square
	P_CHK		*p*-Value, Klotz Chi-Square Test
	XP_CHK	**	Exact *p*-Value, Klotz Chi-Square Test
	XPT_CHK	**	Exact Point Probability, Klotz Chi-Square
MEDIAN	_MED_	*	Two-Sample Median Statistic
	Z_MED	*	Median Statistic, Standardized
	PL_MED	*	*p*-Value (Left-Sided), Median Test
	PR_MED	*	*p*-Value (Right-Sided), Median Test
	P2_MED	*	*p*-Value (Two-Sided), Median Test
	XPL_MED	*	Exact *p*-Value (Left-Sided), Median Test
	XPR_MED	*	Exact *p*-Value (Right-Sided), Median Test
	XPT_MED	*	Exact Point Probability, Median Test
	XP2_MED	*	Exact *p*-Value (Two-Sided), Median Test
	CHMED		Median Chi-Square (Brown-Mood Test)
	DF_CHMED		Degrees of Freedom, Median Chi-Square
	P_CHMED		*p*-Value, Median Chi-Square Test
	XP_CHMED	**	Exact *p*-Value, Median Chi-Square Test
	XPT_CHME	**	Exact Point Probability, Median Chi-Square
MOOD	_MOOD_	*	Two-Sample Mood Statistic
	Z_MOOD	*	Mood Statistic, Standardized
	PL_MOOD	*	*p*-Value (Left-Sided), Mood Test
	PR_MOOD	*	*p*-Value (Right-Sided), Mood Test
	P2_MOOD	*	*p*-Value (Two-Sided), Mood Test
	XPL_MOOD	*	Exact *p*-Value (Left-Sided), Mood Test
	XPR_MOOD	*	Exact *p*-Value (Right-Sided), Mood Test
	XPT_MOOD	*	Exact Point Probability, Mood Test
	XP2_MOOD	*	Exact *p*-Value (Two-Sided), Mood Test
	CHMOOD		Mood Chi-Square
	DF_CHMOO		Degrees of Freedom, Mood Chi-Square
	P_CHMOOD		*p*-Value, Mood Chi-Square Test
	XP_CHMOO	**	Exact *p*-Value, Mood Chi-Square Test
	XPT_CHMO	**	Exact Point Probability, Mood Chi-Square

Table 62.5 *continued*

Option	Output Variables		Variable Descriptions
SAVAGE	_SAV_	*	Two-Sample Savage Statistic
	Z_SAV	*	Savage Statistic, Standardized
	PL_SAV	*	*p*-Value (Left-Sided), Savage Test
	PR_SAV	*	*p*-Value (Right-Sided), Savage Test
	P2_SAV	*	*p*-Value (Two-Sided), Savage Test
	XPL_SAV	*	Exact *p*-Value (Left-Sided), Savage Test
	XPR_SAV	*	Exact *p*-Value (Right-Sided), Savage Test
	XPT_SAV	*	Exact Point Probability, Savage Test
	XP2_SAV	*	Exact *p*-Value (Two-Sided), Savage Test
	CHSAV		Savage Chi-Square
	DF_CHSAV		Degrees of Freedom, Savage Chi-Square
	P_CHSAV		*p*-Value, Savage Chi-Square Test
	XP_CHSAV	**	Exact *p*-Value, Savage Chi-Square Test
	XPT_CHSA	**	Exact Point Probability, Savage Chi-Square
SCORES=DATA	_DATA_	*	Two-Sample Data Scores Statistic
	Z_DATA	*	Data Scores Statistic, Standardized
	PL_DATA	*	*p*-Value (Left-Sided), Data Scores Test
	PR_DATA	*	*p*-Value (Right-Sided), Data Scores Test
	P2_DATA	*	*p*-Value (Two-Sided), Data Scores Test
	XPL_DATA	*	Exact *p*-Value (Left-Sided), Data Scores Test
	XPR_DATA	*	Exact *p*-Value (Right-Sided), Data Scores Test
	XPT_DATA	*	Exact Point Probability, Data Scores Test
	XP2_DATA	*	Exact *p*-Value (Two-Sided), Data Scores Test
	CHDATA		Data Scores Chi-Square
	DF_CHDAT		Degrees of Freedom, Data Scores Chi-Square
	P_CHDATA		*p*-Value, Data Scores Chi-Square Test
	XP_CHDAT	**	Exact *p*-Value, Data Scores Chi-Square Test
	XPT_CHDA	**	Exact Point Probability, Data Scores Chi-Square
ST	_ST_	*	Two-Sample Siegel-Tukey Statistic
	Z_ST	*	Siegel-Tukey Statistic, Standardized
	PL_ST	*	*p*-Value (Left-Sided), Siegel-Tukey Test
	PR_ST	*	*p*-Value (Right-Sided), Siegel-Tukey Test
	P2_ST	*	*p*-Value (Two-Sided), Siegel-Tukey Test
	XPL_ST	*	Exact *p*-Value (Left-Sided), Siegel-Tukey Test
	XPR_ST	*	Exact *p*-Value (Right-Sided), Siegel-Tukey Test
	XPT_ST	*	Exact Point Probability, Siegel-Tukey Test
	XP2_ST	*	Exact *p*-Value (Two-Sided), Siegel-Tukey Test
	CHST		Siegel-Tukey Chi-Square
	DF_CHST		Degrees of Freedom, Siegel-Tukey Chi-Square
	P_CHST		*p*-Value, Siegel-Tukey Chi-Square Test
	XP_CHST	**	Exact *p*-Value, Siegel-Tukey Chi-Square Test
	XPT_CHST	**	Exact Point Probability, Siegel-Tukey Chi-Square

Table 62.5 *continued*

Option	Output Variables		Variable Descriptions
VW | NORMAL	_VW_	*	Two-Sample Van der Waerden Statistic
	Z_VW	*	Van der Waerden Statistic, Standardized
	PL_VW	*	*p*-Value (Left-Sided), Van der Waerden Test
	PR_VW	*	*p*-Value (Right-Sided), Van der Waerden Test
	P2_VW	*	*p*-Value (Two-Sided), Van der Waerden Test
	XPL_VW	*	Exact *p*-Value (Left-Sided), Van der Waerden Test
	XPR_VW	*	Exact *p*-Value (Right-Sided), Van der Waerden Test
	XPT_VW	*	Exact Point Probability, Van der Waerden Test
	XP2_VW	*	Exact *p*-Value (Two-Sided), Van der Waerden Test
	CHVW		Van der Waerden Chi-Square
	DF_CHVW		Degrees of Freedom, Van der Waerden Chi-Square
	P_CHVW		*p*-Value, Van der Waerden Chi-Square Test
	XP_CHVW	**	Exact *p*-Value, Van der Waerden Chi-Square Test
	XPT_CHVW	**	Exact Point Prob, Van der Waerden Chi-Square
WILCOXON	_WIL_	*	Two-Sample Wilcoxon Statistic
	Z_WIL	*	Wilcoxon Statistic, Standardized
	PL_WIL	*	*p*-Value (Left-Sided), Wilcoxon Test
	PR_WIL	*	*p*-Value (Right-Sided), Wilcoxon Test
	P2_WIL	*	*p*-Value (Two-Sided), Wilcoxon Test
	PTL_WIL	*	*p*-Value (Left-Sided), Wilcoxon *t* Approximation
	PTR_WIL	*	*p*-Value (Right-Sided), Wilcoxon *t* Approximation
	PT2_WIL	*	*p*-Value (Two-Sided), Wilcoxon *t* Approximation
	XPL_WIL	*	Exact *p*-Value (Left-Sided), Wilcoxon Test
	XPR_WIL	*	Exact *p*-Value (Right-Sided), Wilcoxon Test
	XPT_WIL	*	Exact Point Probability, Wilcoxon Test
	XP2_WIL	*	Exact *p*-Value (Two-Sided), Wilcoxon Test
	_KW_		Kruskal-Wallis Statistic
	DF_KW		Degrees of Freedom, Kruskal-Wallis Test
	P_KW		*p*-Value, Kruskal-Wallis Test
	XP_KW	**	Exact *p*-Value, Kruskal-Wallis Test
	XPT_KW	**	Exact Point Probability, Kruskal-Wallis Test

* Statistic included only for two-sample cases.

** Statistic included only for multisample cases.

Displayed Output

If you specify the ANOVA option, PROC NPAR1WAY displays a "Class Means" table and an "Analysis of Variance" table for each response variable. The "Class Means" table includes the following information for each CLASS variable value or level:

- N, the number of observations
- the Mean of the response variable

The "Analysis of Variance" table includes the following information for each Source of variation (Among classes, and Within classes):

- DF, the degrees of freedom associated with the source
- the Sum of Squares
- the Mean Square, the sum of squares divided by the degrees of freedom

The "Analysis of Variance" table also includes the following:

- the F Value for testing the hypothesis that the group means are equal. This is computed by dividing the Mean Square (Among) by the Mean Square (Within).
- Pr > F, the significance probability corresponding to the F Value

For each score type that you specify, PROC NPAR1WAY displays a "Class Scores" table. The available score types include Wilcoxon, median, Van der Waerden (normal), Savage, Siegel-Tukey, Ansari-Bradley, Klotz, Mood, Conover, and raw data scores. PROC NPAR1WAY computes the scores for the response variable values, and classifies the scored observations according to the CLASS variable values. The "Class Scores" table includes the following information for each CLASS variable level:

- N, the number of observations
- Sum of Scores
- Expected Under H0, the expected sum of scores under the null hypothesis of no difference among classes
- Std Dev Under H0, the standard deviation under the null hypothesis
- Mean Score

When there are only two levels of the CLASS variable, PROC NPAR1WAY displays the following "Two-Sample Test" results for each analysis of scores:

- Statistic, which is the sum of scores for the class with the smaller sample size
- Z, the standardized test statistic, which has an asymptotic standard normal distribution under the null hypothesis
- One-Sided Pr < Z, or One-Sided Pr > Z, the asymptotic one-sided *p*-value, displayed as Pr < Z or Pr > Z, depending on whether Z is <= 0 or > 0
- Two-Sided Pr > |Z|, the asymptotic two-sided *p*-value

For Wilcoxon scores, PROC NPAR1WAY also displays a *t*-approximation for the two-sample test.

If you request an exact test by specifying the score type in the EXACT statement, PROC NPAR1WAY displays the following exact *p*-values for two-sample data:

- One-Sided Pr <= S, or One-Sided Pr >= S, the one-sided exact *p*-value, displayed as Pr <= S or Pr >= S, depending on whether S <= Mean or S > Mean, where S is the test statistic and Mean is its expected value under the null hypothesis
- Point Pr = S, the point probability, if you specify the POINT option in the EXACT statement
- Two-Sided Pr >= |S - Mean|, the two-sided exact *p*-value

If you request Monte Carlo estimates for the exact test by specifying the MC option in the EXACT statement, PROC NPAR1WAY displays the following information for two-sample data:

- Estimate of One-Sided Pr <= S or One-Sided Pr >= S, the one-sided exact *p*-value, together with its Lower and Upper Confidence Limits
- Estimate of Two-Sided Pr >= |S - Mean|, the two-sided exact *p*-value, together with its Lower and Upper Confidence Limits
- Number of Samples used to compute the Monte Carlo estimates
- Initial Seed used to compute the Monte Carlo estimates

For both two-sample and multisample data, PROC NPAR1WAY displays the following "One-Way Analysis" for each score type:

- Chi-Square, the one-way ANOVA statistic for testing the null hypothesis of no difference among classes
- DF, the degrees of freedom
- Pr > Chi-Square, the asymptotic *p*-value

For multisample data, if you request an exact test by specifying the score type in the EXACT statement, PROC NPAR1WAY also displays the exact p-value as follows:

- Exact Pr >= Chi-Square
- Exact Pr = Chi-Square, the point probability, if you specify the POINT option in the EXACT statement

For multisample data, if you request a Monte Carlo estimate for the exact test by specifying the MC option in the EXACT statement, PROC NPAR1WAY displays the following information:

- Estimate of Pr >= Chi-Square, together with its Lower and Upper Confidence Limits
- Number of Samples used to compute the Monte Carlo estimate
- Initial Seed used to compute the Monte Carlo estimate

If you specify the HL option, PROC NPAR1WAY produces a "Hodges-Lehmann Estimation" table for two-sample data, which includes the following information:

- Location Shift estimate
- Confidence Limits for the Location Shift
- Confidence Interval Midpoint
- Asymptotic Standard Error estimate, based on the confidence interval

If you request exact Hodges-Lehmann confidence limits by specifying the HL option in the EXACT statement, the "Hodges-Lehmann Estimation" table also includes Exact Confidence Limits and the exact Interval Midpoint.

If you specify the EDF option, PROC NPAR1WAY produces tables for the Kolmogorov-Smirnov Test, the Cramer-von Mises Test, and for two-sample data only, the Kuiper Test.

The "Kolmogorov-Smirnov Test" table includes the following information for each CLASS variable level:

- N, the number of observations
- EDF at Maximum, the value of the class EDF (empirical distribution function) at its maximum deviation from the pooled EDF
- Deviation from Mean at Maximum, the value of $\sqrt{n_i}\sqrt{F_i - F}$ at its maximum, where n_i is the class sample size, F_i is the class EDF, and F is the pooled EDF

PROC NPAR1WAY displays the following Kolmogorov-Smirnov statistics:

- KS, the Kolmogorov-Smirnov statistic
- KSa, the asymptotic Kolmogorov-Smirnov statistic, where KSa = $\sqrt{n}$ KS

For two-sample data, PROC NPAR1WAY displays the following Kolmogorov-Smirnov statistics:

- Pr > KSa, the asymptotic p-value for KSa, which equals Pr > D
- D, the two-sample Kolmogorov-Smirnov statistic, which is $\max_j |F_1(x_j) - F_2(x_j)|$

For two-sample data, if you specify the D option, PROC NPAR1WAY also displays the following one-sided Kolmogorov-Smirnov statistics and their asymptotic p-values:

- D+, which is $\max_j (F_1(x_j) - F_2(x_j))$
- Pr > D+
- D-, which is $\max_j (F_2(x_j) - F_1(x_j))$
- Pr > D-

For two-sample data, if you request an exact Kolmogorov-Smirnov test by specifying the KS option in the EXACT statement, PROC NPAR1WAY displays the following exact p-values:

- Exact Pr >= D
- Exact Pr >= D+
- Exact Pr >= D-
- Exact Point Pr = D, Exact Point Pr = D+, and Exact Point Pr = D-, if you specify the POINT option in the EXACT statement

If you request Monte Carlo estimates for the two-sample exact Kolmogorov-Smirnov test, PROC NPAR1WAY displays the following information for two-sample data:

- Estimate of Pr >= D, together with its Lower and Upper Confidence Limits
- Estimate of Pr >= D+, together with its Lower and Upper Confidence Limits
- Estimate of Pr >= D-, together with its Lower and Upper Confidence Limits
- Number of Samples used to compute the Monte Carlo estimates
- Initial Seed used to compute the Monte Carlo estimates

The "Cramer-von Mises Test" table includes the following information for each CLASS variable level:

- N, the number of observations
- Summed Deviation from Mean, which is $(n_i/n)\sum_{j=1}^{p} t_j \left(F_i(x_j) - F(x_j)\right)^2$

PROC NPAR1WAY also displays the following Cramer-von Mises statistics:

- CM, the Cramer-von Mises statistic
- CMa, the asymptotic Cramer-von Mises statistic, where CMa = n CM

For two-sample data, PROC NPAR1WAY displays the "Kuiper Test" table, which includes the following information for each CLASS variable level:

- N, the number of observations
- Deviation from Mean, which is $\max_j |F_1(x_j) - F_2(x_j)|$

PROC NPAR1WAY also displays the following Kuiper two-sample test statistics:

- K, the Kuiper two-sample test statistic
- Ka, the asymptotic Kuiper two-sample test statistic, where Ka = K $\sqrt{n_1 n_2/n}$
- Pr > Ka

ODS Table Names

PROC NPAR1WAY assigns a name to each table it creates. You can use these names to reference tables when using the Output Delivery System (ODS) to select tables and create output data sets. For more information about ODS, see Chapter 20, "Using the Output Delivery System." Table 62.6 lists the table names, along with the corresponding analysis options.

The WILCOXON, MEDIAN, VW, SAVAGE, and EDF options are the default if you do not specify any analysis options in the PROC NPAR1WAY statement.

Table 62.6 ODS Tables Produced by PROC NPAR1WAY

ODS Table Name	Description	Statement	Option
ANOVA	Analysis of variance	PROC	ANOVA
ABAnalysis	Ansari-Bradley one-way analysis	PROC	AB
ABMC	Monte Carlo estimates for the Ansari-Bradley exact test	EXACT	AB / MC
ABScores	Ansari-Bradley scores	PROC	AB
ABTest *	Ansari-Bradley two-sample test	PROC	AB
ClassMeans	Class means	PROC	ANOVA
ConoverAnalysis	Conover one-way analysis	PROC	CONOVER
ConoverMC	Monte Carlo estimates for the Conover exact test	EXACT	CONOVER / MC
ConoverScores	Conover scores	PROC	CONOVER
ConoverTest *	Conover two-sample test	PROC	CONOVER
CVMStats	Cramer-von Mises statistics	PROC	EDF
CVMTest	Cramer-von Mises test	PROC	EDF
DataScores	Data scores	PROC	SCORES=DATA
DataScoresAnalysis	Data scores one-way analysis	PROC	SCORES=DATA
DataScoresMC	Monte Carlo estimates for the data scores exact test	EXACT	SCORES=DATA / MC
DataScoresTest *	Data scores two-sample test	PROC	SCORES=DATA
HodgesLehmann *	Hodges-Lehmann estimation	PROC	HL
KlotzAnalysis	Klotz one-way analysis	PROC	KLOTZ
KlotzMC	Monte Carlo estimates for the Klotz exact test	EXACT	KLOTZ / MC
KlotzScores	Klotz scores	PROC	KLOTZ
KlotzTest *	Klotz two-sample test	PROC	KLOTZ
KolSmir2Stats *	Kolmogorov-Smirnov two-sample statistics	PROC	EDF
KolSmirExactTest *	Kolmogorov-Smirnov exact test	EXACT	KS | EDF
KolSmirStats **	Kolmogorov-Smirnov statistics	PROC	EDF
KolSmirTest	Kolmogorov-Smirnov test	PROC	EDF
KruskalWallisMC **	Monte Carlo estimates for the Kruskal-Wallis exact test	EXACT	WILCOXON / MC
KruskalWallisTest	Kruskal-Wallis test	PROC	WILCOXON
KSMC *	Monte Carlo estimates for the Kolmogorov-Smirnov exact test	EXACT	KS | EDF / MC
KuiperStats *	Kuiper two-sample statistics	PROC	EDF
KuiperTest *	Kuiper test	PROC	EDF
MedianAnalysis	Median one-way analysis	PROC	MEDIAN
MedianMC	Monte Carlo estimates for the median exact test	EXACT	MEDIAN / MC
MedianScores	Median scores	PROC	MEDIAN
MedianTest *	Median two-sample test	PROC	MEDIAN
MoodAnalysis	Mood one-way analysis	PROC	MOOD

Table 62.6 *continued*

ODS Table Name	Description	Statement	Option
MoodMC	Monte Carlo estimates for the Mood exact test	EXACT	MOOD / MC
MoodScores	Mood scores	PROC	MOOD
MoodTest *	Mood two-sample test	PROC	MOOD
SavageAnalysis	Savage one-way analysis	PROC	SAVAGE
SavageMC	Monte Carlo estimates for the Savage exact test	EXACT	SAVAGE / MC
SavageScores	Savage scores	PROC	SAVAGE
SavageTest *	Savage two-sample test	PROC	SAVAGE
STAnalysis	Siegel-Tukey one-way analysis	PROC	ST
STMC	Monte Carlo estimates for the Siegel-Tukey exact test	EXACT	ST / MC
STScores	Siegel-Tukey scores	PROC	ST
STTest *	Siegel-Tukey two-sample test	PROC	ST
VWAnalysis	Van der Waerden one-way analysis	PROC	VW | NORMAL
VWMC	Monte Carlo estimates for the Van der Waerden exact test	EXACT	VW | NORMAL / MC
VWScores	Van der Waerden scores	PROC	VW | NORMAL
VWTest *	Van der Waerden two-sample test	PROC	VW | NORMAL
WilcoxonMC *	Monte Carlo estimates for the Wilcoxon two-sample exact test	EXACT	WILCOXON / MC
WilcoxonScores	Wilcoxon scores	PROC	WILCOXON
WilcoxonTest *	Wilcoxon two-sample test	PROC	WILCOXON

* PROC NPAR1WAY produces this table only for two-sample data.

** PROC NPAR1WAY produces this table only for multisample data.

ODS Graphics

PROC NPAR1WAY assigns a name to each graph it creates with ODS Graphics. You can use these names to reference the graphs. Table 62.7 lists the names of the graphs that PROC NPAR1WAY generates, along with the corresponding analysis options.

To request graphics with PROC NPAR1WAY, you must first enable ODS Graphics by specifying the **ODS GRAPHICS ON** statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. When you have enabled ODS Graphics, you can request specific plots with the PLOTS= option in the PROC NPAR1WAY statement. If you do not specify the PLOTS= option but have enabled ODS Graphics, then PROC NPAR1WAY produces all plots associated with the analyses you request.

Table 62.7 ODS Graphics Produced by PROC NPAR1WAY

ODS Graph Name	Plot Description	Option
ABBoxPlot	Box plot of Ansari-Bradley scores	AB
ANOVABoxPlot	Box plot of raw data	ANOVA
ConoverBoxPlot	Box plot of Conover scores	CONOVER
DataScoresBoxPlot	Box plot of data scores	SCORES=DATA
EDFPlot	Empirical distribution function plot	EDF
KlotzBoxPlot	Box plot of Klotz scores	KLOTZ
MedianPlot	Median plot	MEDIAN
MoodBoxPlot	Box plot of Mood scores	MOOD
SavageBoxPlot	Box plot of Savage scores	SAVAGE
STBoxPlot	Box plot of Siegel-Tukey scores	ST
VWBoxPlot	Box plot of Van der Waerden scores	VW | NORMAL
WilcoxonBoxPlot	Box plot of Wilcoxon scores	WILCOXON

Examples: NPAR1WAY Procedure

Example 62.1: Two-Sample Location Tests and Plots

Fifty-nine female patients with rheumatoid arthritis who participated in a clinical trial were assigned to two groups, active and placebo. The response status (excellent=5, good=4, moderate=3, fair=2, poor=1) of each patient was recorded.

The following SAS statements create the data set Arthritis, which contains the observed status values for all the patients. The variable Treatment denotes the treatment received by a patient, and the variable Response contains the response status of the patient. The variable Freq contains the frequency of the observation, which is the number of patients with the Treatment and Response combination.

```
data Arthritis;
   input Treatment $ Response Freq @@;
   datalines;
Active 5 5 Active 4 11 Active 3 5 Active 2 1 Active 1 5
Placebo 5 2 Placebo 4 4 Placebo 3 7 Placebo 2 7 Placebo 1 12
;
```

The following PROC NPAR1WAY statements test the null hypothesis that there is no difference in the patient response status against the alternative hypothesis that the patient response status differs in the two treatment groups. The WILCOXON option requests the Wilcoxon test for difference in location, and the MEDIAN option requests the median test for difference in location. The variable Treatment is the CLASS variable, and the VAR statement specifies that the variable Response is the analysis variable.

The PLOTS= option requests a box plot of the Wilcoxon scores and a median plot for Response classified by Treatment. The **ODS GRAPHICS ON** statement enables ODS Graphics, which is required before requesting plots.

```
ods graphics on;
proc npar1way data=Arthritis wilcoxon median
      plots=(wilcoxonboxplot medianplot);
   class Treatment;
   var Response;
   freq Freq;
run;
ods graphics off;
```

Output 62.1.1 shows the results of the Wilcoxon analysis. The Wilcoxon two-sample test statistic equals 999.0, which is the sum of the Wilcoxon scores for the smaller sample (Active). This sum is greater than 810.0, which is the expected value under the null hypothesis of no difference between the two samples, Active and Placebo. The one-sided p-value is 0.0016, which indicates that the patient response for the Active treatment is significantly more than for the Placebo group.

Output 62.1.1 Wilcoxon Two-Sample Test

The NPAR1WAY Procedure

Wilcoxon Scores (Rank Sums) for Variable Response
Classified by Variable Treatment

Treatment	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
Active	27	999.0	810.0	63.972744	37.000000
Placebo	32	771.0	960.0	63.972744	24.093750

Average scores were used for ties.

Wilcoxon Two-Sample Test

Statistic	999.0000
Normal Approximation	
Z	2.9466
One-Sided Pr > Z	0.0016
Two-Sided Pr > |Z|	0.0032
t Approximation	
One-Sided Pr > Z	0.0023
Two-Sided Pr > |Z|	0.0046

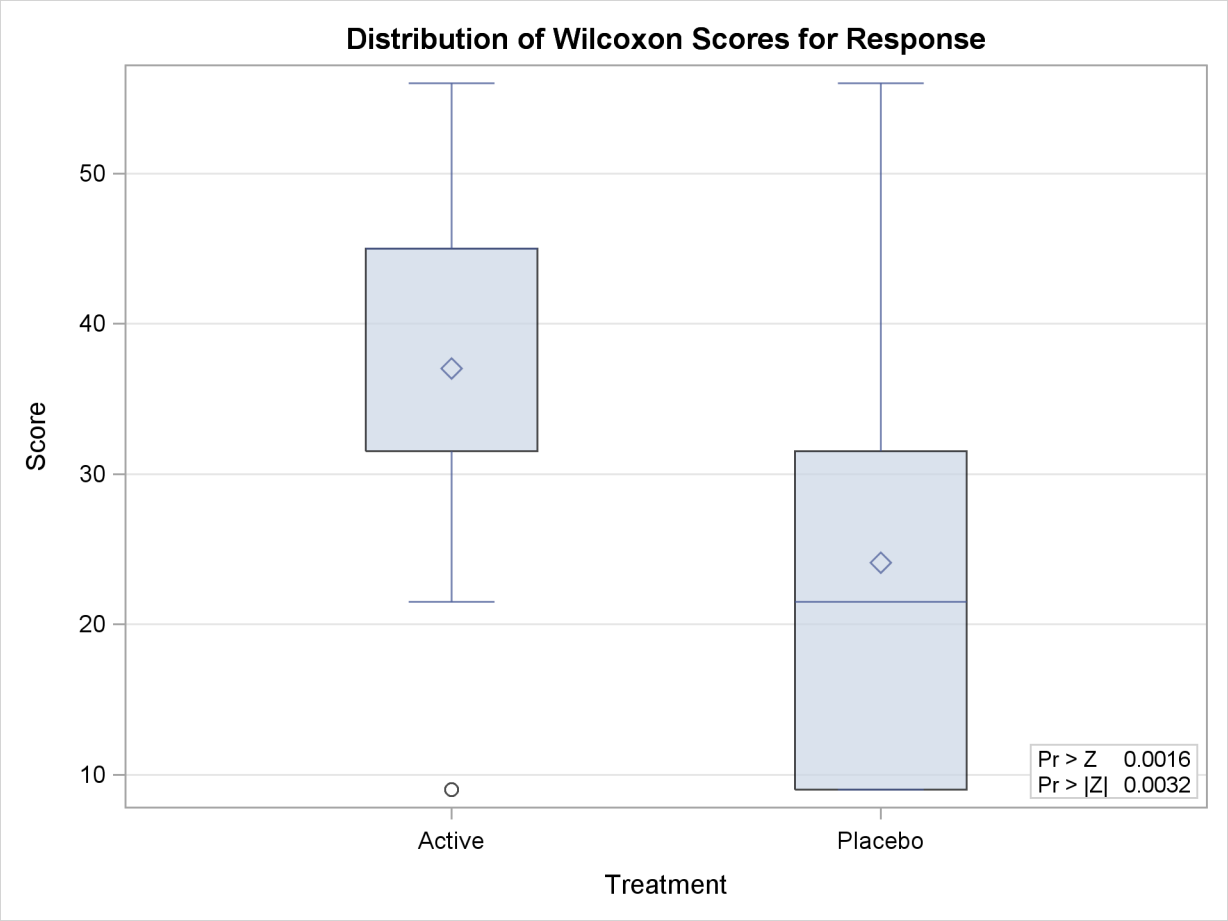
Z includes a continuity correction of 0.5.

Kruskal-Wallis Test

Chi-Square	8.7284
DF	1
Pr > Chi-Square	0.0031

Output 62.1.2 displays the box plot of Wilcoxon scores classified by Treatment, which corresponds to the Wilcoxon analysis in Output 62.1.1. To remove the *p*-values from the box plot display, you can specify the NOSTATS plot option in parentheses following the WILCOXONBOXPLOT option.

Output 62.1.2 Box Plot of Wilcoxon Scores



Output 62.1.3 shows the results of the median two-sample test. The test statistic equals 18.9167, and its standardized Z value is 3.1667. The one-sided p-value Pr > Z equals 0.0005. This supports the alternative hypothesis that the effect of the Active treatment is greater than that of the Placebo.

Output 62.1.4 displays the median plot for the analysis of Response classified by Treatment. The median plot is a stacked bar chart showing the frequencies above and below the overall median. This plot corresponds to the median scores analysis in Output 62.1.3.

Output 62.1.3 Median Two-Sample Test

Median Scores (Number of Points Above Median) for Variable Response
Classified by Variable Treatment

Treatment	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
Active	27	18.916667	13.271186	1.728195	0.700617
Placebo	32	10.083333	15.728814	1.728195	0.315104

Average scores were used for ties.

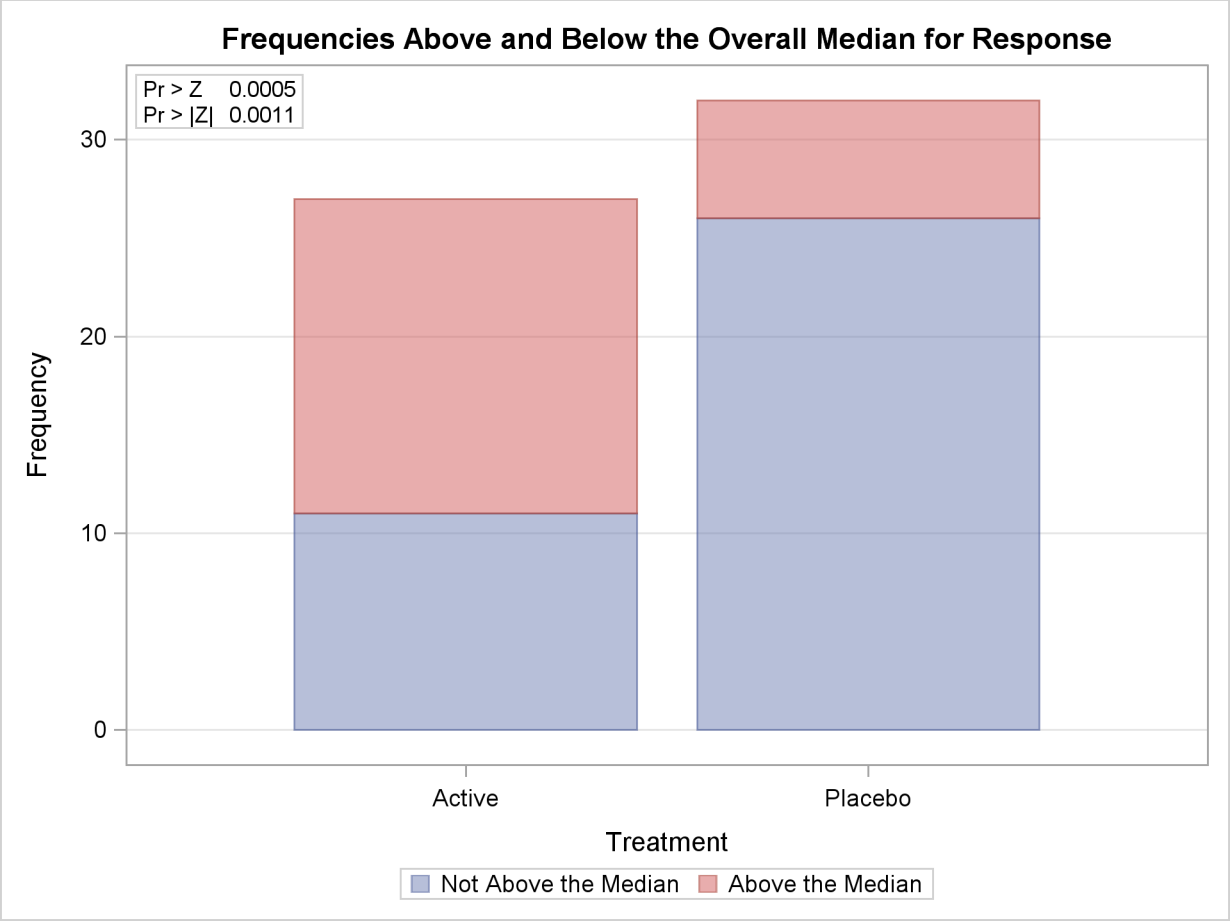
Median Two-Sample Test

Statistic	18.9167
Z	3.2667
One-Sided Pr > Z	0.0005
Two-Sided Pr > |Z|	0.0011

Median One-Way Analysis

Chi-Square	10.6713
DF	1
Pr > Chi-Square	0.0011

Output 62.1.4 Median Plot



Example 62.2: EDF Statistics and EDF Plot

This example uses the SAS data set Arthritis created in Example 62.1. The data set contains the variable Treatment, which denotes the treatment received by a patient, and the variable Response, which contains the response status of the patient. The variable Freq contains the frequency of the observation, which is the number of patients with the Treatment and Response combination.

The following statements request empirical distribution function (EDF) statistics, which test whether the distribution of a variable is the same across different groups. The EDF option requests the EDF analysis. The variable Treatment is the CLASS variable, and the variable Response specified in the VAR statement is the analysis variable. The FREQ statement names Freq as the frequency variable.

The PLOTS= option requests an EDF plot for Response classified by Treatment. The **ODS GRAPHICS ON** statement enables ODS Graphics, which is required before requesting plots.

```
ods graphics on;
proc npar1way edf plots=edfplot data=Arthritis;
   class Treatment;
   var Response;
   freq Freq;
run;
ods graphics off;
```

Output 62.2.1 shows EDF statistics that compare the two levels of Treatment, Active and Placebo. The asymptotic *p*-value for the Kolmogorov-Smirnov test is 0.0164. This supports rejection of the null hypothesis that the distributions are the same for the two samples.

Output 62.2.2 shows the EDF plot for Response classified by Treatment.

Output 62.2.1 Empirical Distribution Function Statistics

The NPAR1WAY Procedure

Kolmogorov-Smirnov Test for Variable Response
Classified by Variable Treatment

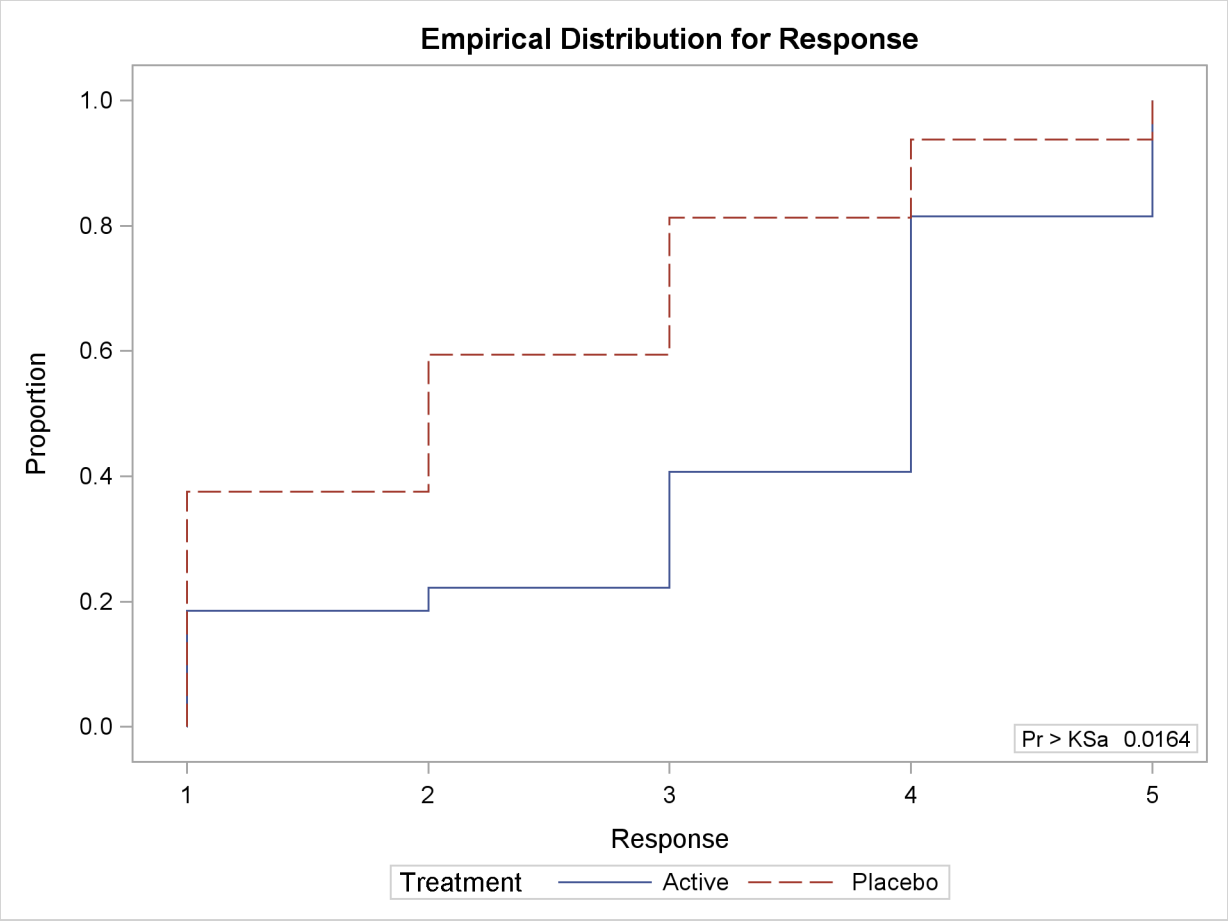
Treatment	N	EDF at Maximum	Deviation from Mean at Maximum
Active	27	0.407407	-1.141653
Placebo	32	0.812500	1.048675
Total	59	0.627119	

Maximum Deviation Occurred at Observation 3
Value of Response at Maximum = 3.0

Kolmogorov-Smirnov Two-Sample Test (Asymptotic)

KS	0.201818	D	0.405093
KSa	1.550191	Pr > KSa	0.0164

Output 62.2.2 Empirical Distribution Function Plot



Example 62.3: Exact Wilcoxon Two-Sample Test

Researchers conducted an experiment to compare the effects of two stimulants. Thirteen randomly selected subjects received the first stimulant, and six randomly selected subjects received the second stimulant. The reaction times (in minutes) were measured while the subjects were under the influence of the stimulants.

The following SAS statements create the data set React, which contains the observed reaction times for each stimulant. The variable Stim represents Stimulant 1 or 2. The variable Time contains the reaction times observed for subjects under the stimulant.

```
data React;
   input Stim Time @@;
   datalines;
1 1.94   1 1.94   1 2.92   1 2.92   1 2.92   1 2.92   1 3.27
1 3.27   1 3.27   1 3.27   1 3.70   1 3.70   1 3.74
2 3.27   2 3.27   2 3.27   2 3.70   2 3.70   2 3.74
;
```

The following statements request a Wilcoxon test of the null hypothesis that there is no difference between the effects of the two stimulants. Stim is the CLASS variable, and Time is the analysis variable. The WILCOXON option requests an analysis of Wilcoxon scores. The CORRECT=NO option removes the continuity correction from the computation of the standardized z test statistic. The WILCOXON option in the EXACT statement requests exact p-values for the Wilcoxon test. Because the sample size is small, the large-sample normal approximation might not be adequate, and it is appropriate to compute the exact test. These statements produce the results shown in Output 62.3.1.

```
proc npar1way wilcoxon correct=no data=React;
   class Stim;
   var Time;
   exact wilcoxon;
run;
```

Output 62.3.1 Wilcoxon Two-Sample Test

The NPAR1WAY Procedure

Wilcoxon Scores (Rank Sums) for Variable Time
Classified by Variable Stim

Stim	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
1	13	110.50	130.0	11.004784	8.500
2	6	79.50	60.0	11.004784	13.250

Average scores were used for ties.

Wilcoxon Two-Sample Test

Statistic (S)	79.5000
Normal Approximation	
Z	1.7720
One-Sided Pr > Z	0.0382
Two-Sided Pr > |Z|	0.0764
t Approximation	
One-Sided Pr > Z	0.0467
Two-Sided Pr > |Z|	0.0933
Exact Test	
One-Sided Pr >= S	0.0527
Two-Sided Pr >= |S - Mean|	0.1054

Kruskal-Wallis Test

Chi-Square	3.1398
DF	1
Pr > Chi-Square	0.0764

Output 62.3.1 displays the results of the Wilcoxon two-sample test. The Wilcoxon statistic equals 79.50. Since this value is greater than 60.0, the expected value under the null hypothesis, PROC NPAR1WAY displays the right-sided p-values. The normal approximation for the Wilcoxon two-sample test yields a one-sided p-value of 0.0421 and a two-sided p-value of 0.0843. For the exact Wilcoxon test, the one-sided p-value is 0.0527, and the two-sided p-value is 0.1054.

Example 62.4: Hodges-Lehmann Estimation

This example uses the SAS data set React created in Example 62.3. The data set contains the variable Stim, which represents Stimulant 1 or 2, and the variable Time, which contains the reaction times observed for subjects under the stimulant.

The following statements request Hodges-Lehmann estimation of the location shift between the two groups. Stim is the CLASS variable, and Time is the analysis variable. The HL option requests Hodges-Lehmann estimation. The ALPHA= option sets the confidence level for the Hodges-Lehmann confidence limits. The HL option in the EXACT statement requests exact confidence limits for the estimate of location shift. The ODS SELECT statement selects which tables to display. Output 62.4.1 shows the Hodges-Lehmann results.

```
proc npar1way hl alpha=.02 data=React;
   class Stim;
   var Time;
   exact hl;
   ods select WilcoxonScores HodgesLehmann;
run;
```

Output 62.4.1 Hodges-Lehmann Estimate of Location Shift

The NPAR1WAY Procedure

Wilcoxon Scores (Rank Sums) for Variable Time
Classified by Variable Stim

Stim	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
1	13	110.50	130.0	11.004784	8.500
2	6	79.50	60.0	11.004784	13.250

Average scores were used for ties.

Hodges-Lehmann Estimation

Location Shift 0.3500

Type	98% Confidence Limits		Interval Midpoint	Asymptotic Standard Error
Asymptotic (Moses)	0.0000	0.8200	0.4100	0.1762
Exact	0.0000	1.3300	0.6650	

The HL option automatically invokes the WILCOXON option, producing a table of Wilcoxon scores (Output 62.4.1). The Hodges-Lehmann estimate of location shift is 0.35, and the asymptotic confidence limits are 0.00 and 0.82. The confidence interval midpoint equals 0.41, which can also be used as an estimate of the location shift. The ASE estimate of 0.1762 is based on the length of the confidence interval. The exact confidence limits are 0.00 and 1.33.

Example 62.5: Exact Savage Multisample Test

A researcher conducting a laboratory experiment randomly assigned 15 mice to receive one of three drugs. The survival time (in days) was then recorded.

The following SAS statements create the data set Mice, which contains the observed survival times for the mice. The variable Treatment denotes the treatment received. The variable Days contains the number of days the mouse survived.

```
data Mice;
   input Treatment $ Days @@;
   datalines;
1 1 1 1 1 3 1 3 1 4
2 3 2 4 2 4 2 4 2 15
3 4 3 4 3 10 3 10 3 26
;
```

The following statements request a Savage test of the null hypothesis that there is no difference in survival time among the three drugs. Treatment is the CLASS variable, and Days is the analysis variable. The SAVAGE option requests an analysis of Savage scores. The SAVAGE option in the EXACT statement requests exact *p*-values for the Savage test. Because the sample size is small, the large-sample normal approximation might not be adequate, and it is appropriate to compute the exact test.

PROC NPAR1WAY tests the null hypothesis that there is no difference in the survival times among the three drugs against the alternative hypothesis of difference among the drugs. The SAVAGE option specifies an analysis based on Savage scores. The variable Treatment is the CLASS variable, and the variable Days is the response variable. The EXACT statement requests the exact Savage test.

```
proc npar1way savage data=Mice;
   class Treatment;
   var Days;
   exact savage;
run;
```

Output 62.5.1 shows the results of the Savage test. The exact *p*-value is 0.0445, which supports a difference in survival times among the drugs at the 0.05 level. The asymptotic *p*-value based on the chi-square approximation is 0.0638.

Output 62.5.1 Savage Multisample Exact Test

The NPAR1WAY Procedure

Savage Scores (Exponential) for Variable Days
Classified by Variable Treatment

Treatment	N	Sum of Scores	Expected Under H0	Std Dev Under H0	Mean Score
1	5	-3.367980	0.0	1.634555	-0.673596
2	5	0.095618	0.0	1.634555	0.019124
3	5	3.272362	0.0	1.634555	0.654472

Average scores were used for ties.

Savage One-Way Analysis

Chi-Square	5.5047
DF	2
Asymptotic Pr > Chi-Square	0.0638
Exact Pr >= Chi-Square	0.0445

References

Agresti, A. (1992), "A Survey of Exact Inference for Contingency Tables," *Statistical Science*, 7 (1), 131–177.

Agresti, A. (2007), *An Introduction to Categorical Data Analysis*, Second Edition, New York: John Wiley & Sons.

Agresti, A. (2002), *Categorical Data Analysis*, Second Edition, New York: John Wiley & Sons.

Agresti, A., Mehta, C. R., and Patel, N. R. (1990), "Exact Inference for Contingency Tables with Ordered Categories," *Journal of American Statistical Association*, 85, 453–458.

Agresti, A., Wackerly, D., and Boyett, J. M. (1979), "Exact Conditional Tests for Cross-Classifications: Approximation of Attained Significance Levels," *Psychometrika*, 44, 75–83.

Bishop, Y., Fienberg, S. E., and Holland, P. W. (1975), *Discrete Multivariate Analysis: Theory and Practice*, Cambridge, MA: MIT Press.

Conover, W. J. (1999), *Practical Nonparametric Statistics*, Third Edition, New York: John Wiley & Sons.

Gail, M. and Mantel, N. (1977), "Counting the Number of $r \times c$ Contingency Tables with Fixed Margins," *Journal of the American Statistical Association*, 72, 859–862.

Gibbons, J. D. and Chakraborti, S. (1992), *Nonparametric Statistical Inference*, Third Edition, New

York: Marcel Dekker.

Hajek, J. (1969), *A Course in Nonparametric Statistics*, San Francisco: Holden-Day.

Halverson, J. O. and Sherwood, F. W. (1930), "Investigations in the Feeding of Cottonseed Meal to Cattle," *North Carolina Agr. Exp. Sta. Tech. Bulletin*, 39, 158pp.

Hodges, J. L., Jr. (1957), "The Significance Probability of the Smirnov Two-Sample Test," *Arkiv for Matematik*, 3, 469–486.

Hodges, J. L., Jr. and Lehmann, E. L. (1983). "Hodges-Lehmann Estimators," in *Encyclopedia of Statistical Sciences*, vol. 3, ed. S. Kotz, N. L. Johnson, and C. B. Read, New York: John Wiley & Sons, 463–465.

Hollander, M. and Wolfe, D. A. (1999), *Nonparametric Statistical Methods*, Second Edition, New York: John Wiley & Sons.

Kiefer, J. (1959), "K-Sample Analogues of the Kolmogorov-Smirnov and Cramer-von Mises Tests," *Annals of Mathematical Statistics*, 30, 420–447.

Lehmann, E. L. (1963). "Nonparametric Confidence Intervals for a Shift Parameter," *Annals of Mathematical Statistics*, 34, 1507–1512.

Mehta, C. R. and Patel, N. R. (1983), "A Network Algorithm for Performing Fisher's Exact Test in $r \times c$ Contingency Tables," *Journal of American Statistical Association*, 78, 427–434.

Mehta, C. R., Patel, N. R., and Senchaudhuri, P. (1991), "Exact Stratified Linear Rank Tests for Binary Data," *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface* (E. M. Keramidas, ed.), 200–207.

Mehta, C. R., Patel, N. R., and Tsiatis, A. A. (1984), "Exact Significance Testing to Establish Treatment Equivalence with Ordered Categorical Data," *Biometrics*, 40, 819–825.

Owen, D. B. (1962), *Handbook of Statistical Tables*, Reading, MA: Addison-Wesley.

Quade, D. (1966), "On Analysis of Variance for the k-Sample Problem," *Annals of Mathematical Statistics*, 37, 1747–1758.

Randles, R. H. and Wolfe, D. A. (1979), *Introduction to the Theory of Nonparametric Statistics*, New York: John Wiley & Sons.

Sheskin, D. J. (1997), *Handbook of Parametric and Nonparametric Statistical Procedures*, Boca Raton, FL: CRC Press.

Valz, P. D. and Thompson, M. E. (1994), "Exact Inference for Kendall's S and Spearman's ρ with Extensions to Fisher's Exact Test in $r \times c$ Contingency Tables," *Journal of Computational and Graphical Statistics*, 3 (4), 459–472.

Chapter 63

The ORTHOREG Procedure

Contents

Overview: ORTHOREG Procedure **4505**
Getting Started: ORTHOREG Procedure **4506**
Longley Data 4506
Syntax: ORTHOREG Procedure **4509**
PROC ORTHOREG Statement 4509
BY Statement 4511
CLASS Statement 4511
MODEL Statement 4512
WEIGHT Statement 4512
Details: ORTHOREG Procedure **4512**
Missing Values 4512
Output Data Set 4512
Displayed Output 4513
ODS Table Names 4514
Examples: ORTHOREG Procedure **4514**
Example 63.1: Precise Analysis of Variance 4514
Example 63.2: Wampler Data 4517
References **4520**

Overview: ORTHOREG Procedure

The ORTHOREG procedure fits general linear models by the method of least squares. Other SAS/STAT software procedures, such as GLM and REG, fit the same types of models, but PROC ORTHOREG can produce more accurate estimates than other regression procedures when your data are ill conditioned. Instead of collecting crossproducts, PROC ORTHOREG uses Gentleman-Givens transformations to update and compute the upper triangular matrix **R** of the QR decomposition of the data matrix, with special care for scaling (Gentleman 1972, 1973). This method has the advantage over other orthogonalization methods (for example, Householder transformations) of not requiring the data matrix to be stored in memory.

The standard SAS regression procedures (REG and GLM) are very accurate for most problems. However, if you have very ill-conditioned data, these procedures can produce estimates that yield

an error sum of squares very close to the minimum but still different from the exact least squares estimates. Normally, this coincides with estimates that have very high standard errors. In other words, the numerical error is much smaller than the statistical standard error.

Note that PROC ORTHOREG fits models by the method of linear least squares, minimizing the sum of the squared residuals for predicting the responses—that is, the distance between the regression line and the observed Ys. The "ORTHO" in the name of the procedure refers to the orthogonalization approach to solving the least squares equations. In particular, PROC ORTHOREG does *not* perform the modeling method known as "orthogonal regression," which minimizes a different criterion (namely, the distance between the regression line and the X/Y points taken together.)

Getting Started: ORTHOREG Procedure

Longley Data

The labor statistics data set of Longley (1967) is noted for being ill conditioned. Both the ORTHOREG and GLM procedures are applied for comparison (only portions of the PROC GLM results are shown). **NOTE:** The results from this example vary from machine to machine, depending on floating-point configuration.

The following statements read the data into the SAS data set Longley:

```
title 'PROC ORTHOREG used with Longley data';
data Longley;
   input Employment Prices GNP Jobless Military PopSize Year;
   datalines;
60323  83.0 234289 2356 1590 107608 1947
61122  88.5 259426 2325 1456 108632 1948
60171  88.2 258054 3682 1616 109773 1949
61187  89.5 284599 3351 1650 110929 1950
63221  96.2 328975 2099 3099 112075 1951
63639  98.1 346999 1932 3594 113270 1952
64989  99.0 365385 1870 3547 115094 1953
63761 100.0 363112 3578 3350 116219 1954
66019 101.2 397469 2904 3048 117388 1955
67857 104.6 419180 2822 2857 118734 1956
68169 108.4 442769 2936 2798 120445 1957
66513 110.8 444546 4681 2637 121950 1958
68655 112.6 482704 3813 2552 123366 1959
69564 114.2 502601 3931 2514 125368 1960
69331 115.7 518173 4806 2572 127852 1961
70551 116.9 554894 4007 2827 130081 1962
;
run;
```

The data set contains one dependent variable, Employment (total derived employment), and six independent variables: Prices (GNP implicit price deflator with year 1954 = 100), GNP (gross national product), Jobless (unemployment), Military (size of armed forces), PopSize (noninstitutional population aged 14 and over), and Year (year).

The following statements use the ORTHOREG procedure to model the Longley data by using a quadratic model in each independent variable, without interaction:

```
proc orthoreg data=Longley;
   model Employment = Prices   Prices*Prices
                      GNP      GNP*GNP
                      Jobless  Jobless*Jobless
                      Military Military*Military
                      PopSize  PopSize*PopSize
                      Year     Year*Year;
run;
```

Figure 63.1 shows the resulting analysis.

Figure 63.1 PROC ORTHOREG Results

```
                    PROC ORTHOREG used with Longley data

                          The ORTHOREG Procedure

                     Dependent Variable: Employment

                                  Sum of
 Source                 DF        Squares     Mean Square    F Value    Pr > F

 Model                  12    184864508.5    15405375.709     320.24    0.0003
 Error                   3   144317.49568    48105.831895
 Corrected Total        15      185008826

                          Root MSE    219.33041717
                          R-Square    0.9992199426

                                                Standard
 Parameter      DF    Parameter Estimate           Error    t Value    Pr > |t|

 Intercept       1      186931078.640216    154201839.66       1.21      0.3122
 Prices          1      1324.50679362506    916.17455832       1.45      0.2440
 Prices**2       1     -6.61923922845539    4.7891445654      -1.38      0.2609
 GNP             1     -0.12768642156232    0.0738897784      -1.73      0.1824
 GNP**2          1    3.1369569286212E-8    8.7167753E-8       0.36      0.7428
 Jobless         1     -4.35507653558708    1.3851792402      -3.14      0.0515
 Jobless**2      1      0.00022132944101    0.0001763541       1.26      0.2983
 Military        1      4.91162014560828     1.826715856       2.69      0.0745
 Military**2     1     -0.00113707146734    0.0003539971      -3.21      0.0489
 PopSize         1      -0.0303997234299    5.9272538242      -0.01      0.9962
 PopSize**2      1    -1.212511414607E-6    0.0000237262      -0.05      0.9625
 Year            1     -194907.139041839    157739.28757      -1.24      0.3045
 Year**2         1      50.8067603538501    40.279878943       1.26      0.2963
```

The estimates in Figure 63.1 compare very well with the best estimates available; for additional information, refer to Longley (1967) and Beaton, Rubin, and Barone (1976).

The following statements request the same analysis from the GLM procedure:

```
proc glm data=Longley;
   model Employment = Prices   Prices*Prices
                      GNP      GNP*GNP
                      Jobless  Jobless*Jobless
                      Military Military*Military
                      PopSize  PopSize*PopSize
                      Year     Year*Year;
   ods select OverallANOVA
              FitStatistics
              ParameterEstimates
              Notes;
run;
```

Figure 63.2 contains the overall ANOVA table and the parameter estimates produced by PROC GLM. Notice that the PROC ORTHOREG fit achieves a somewhat smaller root mean square error (RMSE) and also that the GLM procedure detects spurious singularities.

Figure 63.2 Partial PROC GLM Results

PROC ORTHOREG used with Longley data

The GLM Procedure

Dependent Variable: Employment

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	11	184791061.6	16799187.4	308.58	<.0001
Error	4	217764.4	54441.1		
Corrected Total	15	185008826.0			

R-Square	Coeff Var	Root MSE	Employment Mean
0.998823	0.357221	233.3262	65317.00

Figure 63.2 *continued*

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	−3598851.899 B	1327335.652	−2.71	0.0535
Prices	523.802	688.979	0.76	0.4894
Prices*Prices	−2.326	3.507	−0.66	0.5434
GNP	−0.138	0.078	−1.76	0.1526
GNP*GNP	0.000	0.000	0.24	0.8218
Jobless	−4.599	1.459	−3.15	0.0344
Jobless*Jobless	0.000	0.000	1.14	0.3183
Military	4.994	1.942	2.57	0.0619
Military*Military	−0.001	0.000	−3.15	0.0346
PopSize	−4.246	5.156	−0.82	0.4565
PopSize*PopSize	0.000 B	0.000	0.81	0.4655
Year	0.000 B	.	.	.
Year*Year	1.038	0.419	2.48	0.0683

NOTE: The X'X matrix has been found to be singular, and a generalized inverse was used to solve the normal equations. Terms whose estimates are followed by the letter 'B' are not uniquely estimable.

Syntax: ORTHOREG Procedure

The following statements are available in PROC ORTHOREG:

PROC ORTHOREG < *options* > ;
MODEL *dependent=independents* < / *option* > ;
BY *variables* ;
CLASS *variables* < / *option* > ;
WEIGHT *variable* ;

The BY, CLASS, MODEL, and WEIGHT statements are described after the PROC ORTHOREG statement in alphabetical order.

PROC ORTHOREG Statement

PROC ORTHOREG < *options* > ;

The PROC ORTHOREG statement has the following options:

DATA=*SAS-data-set*
specifies the input SAS data set to use. By default, the procedure uses the most recently created SAS data set. The data set specified cannot be a TYPE=CORR, TYPE=COV, or TYPE=SSCP data set.

NOPRINT

suppresses the normal display of results. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System" for more information.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the order in which you want the levels of the classification variables (specified in the CLASS statement) to be sorted. This ordering determines which parameters in the model correspond to each level in the data. Note that the ORDER= option applies to the levels for all classification variables. The exception is the default ORDER=FORMATTED for numeric variables for which you have supplied no explicit format. In this case, the levels are ordered by their internal value. Note that this represents a change from previous releases for how class levels are ordered. Prior to SAS 8, numeric class levels with no explicit format were ordered by their BEST12. formatted values, and to revert to the previous ordering you can specify this format explicitly for the affected classification variables. The change was implemented because the former default behavior for ORDER=FORMATTED often resulted in levels not being ordered numerically and usually required the user to intervene with an explicit format or ORDER=INTERNAL to get the more natural ordering.

The ORDER= option can take the following values.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

If you omit the ORDER= option, PROC ORTHOREG orders by the external formatted value.

OUTEST=*SAS-data-set*

produces an output data set containing the parameter estimates, the BY variables, and the special variables _TYPE_ (value "**PARMS**"), _NAME_ (blank), and _RMSE_ (root mean squared error).

SINGULAR=*s*

specifies a singularity criterion ($s \geq 0$) for the inversion of the triangular matrix $\mathbf{R}$. By default, SINGULAR=10E−12.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC ORTHOREG to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the ORTHOREG procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* < / *option* > **;**

The CLASS statement names the classification variables to be used in the model. Typical classification variables are Treatment, Sex, Race, Group, and Replication. If you use the CLASS statement, it must appear before the MODEL statement.

By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. Prior to SAS 9, class levels were determined by using no more than the first 16 characters of the formatted values. To revert to this previous behavior you can use the TRUNCATE option in the CLASS statement. In any case, you can use formats to group values into levels. See the discussion of the FORMAT procedure in the *Base SAS Procedures Guide* and the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

You can specify the following option in the CLASS statement after a slash (*/*):

TRUNCATE
specifies that class levels should be determined by using only up to the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases prior to SAS 9.

MODEL Statement

MODEL *dependent=independents* < / *option* > **;**

The MODEL statement names the dependent variable and the independent effects. Only one MODEL statement is allowed. The specification of effects and the parameterization of the linear model are the same as in the GLM procedure; see Chapter 39, "The GLM Procedure" for further details.

The following option can be used in the MODEL statement:

NOINT
omits the intercept term from the model.

WEIGHT Statement

WEIGHT *variable* **;**

A WEIGHT statement names a variable in the input data set whose values are relative weights for a weighted least squares regression. If the weight value is proportional to the reciprocal of the variance for each observation, the weighted estimates are the best linear unbiased estimates (BLUE). For a more complete description of the WEIGHT statement, see the section "WEIGHT Statement" on page 2486 in the GLM procedure.

Details: ORTHOREG Procedure

Missing Values

If there is a missing value for any model variable in an observation, the entire observation is dropped from the analysis.

Output Data Set

The OUTEST= option produces a TYPE=EST output SAS data set containing the BY variables, parameter estimates, and four special variables. For each new value of the BY variables, PROC ORTHOREG outputs an observation to the OUTEST= data set. The variables in the data set are as follows:

- parameter estimates for all variables listed in the MODEL statement
- BY variables
- _TYPE_, which is a character variable with the value PARMS for every observation
- _NAME_, which is a character variable left blank for every observation
- _RMSE_, which is the root mean squared error (the estimate of the standard deviation of the true errors)
- Intercept, which is the estimated intercept. This variable does not exist in the OUTEST= data set if the NOINT option is specified.

Displayed Output

PROC ORTHOREG displays the parameter estimates and associated statistics. These include the following:

- overall model analysis of variance, including the error mean square, which is an estimate of σ^2 (the variance of the true errors), and the overall F test for a model effect
- root mean squared error, which is an estimate of the standard deviation of the true errors. It is calculated as the square root of the mean squared error.
- R-Square, R^2, measures how much variation in the dependent variable can be accounted for by the model. R^2, which can range from 0 to 1, is the ratio of the sum of squares for the model to the corrected total sum of squares. In general, the larger the value of R^2, the better the model's fit.
- estimates for the parameters in the linear model

The table of parameter estimates consists of the following:

- the terms used as regressors, including the intercept, identifying the intercept parameter
- degrees of freedom (DF) for the variable. There is one degree of freedom for each parameter being estimated unless the model is not full rank.
- estimated linear coefficients
- estimates of the standard errors of the parameter estimates
- the critical t values for testing whether the parameters are zero. This is computed as the parameter estimate divided by its standard error.
- the two-sided p-value for the t test, which is the probability that a t statistic would obtain a greater absolute value than that observed given that the true parameter is zero

ODS Table Names

PROC ORTHOREG assigns a name to each table it creates. You can use these names to reference the table when you use the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 63.1 ODS Tables Produced by PROC ORTHOREG

ODS Table Name	Description	Statement
ANOVA	Analysis of variance	default
FitStatistics	Overall statistics for fit	default
Levels	Table of class levels	CLASS statement
ParameterEstimates	Parameter estimates	default

Examples: ORTHOREG Procedure

Example 63.1: Precise Analysis of Variance

The data for the following example are from Powell, Murphy, and Gramlich (1982). In order to calibrate an instrument for measuring atomic weight, 24 replicate measurements of the atomic weight of silver (chemical symbol Ag) are made with the new instrument and with a reference instrument.

NOTE: The results from this example vary from machine to machine depending on floating-point configuration.

The following statements read the measurements for the two instruments into the SAS data set AgWeight:

```
title 'Atomic Weight of Silver by Two Different Instruments';
data AgWeight;
   input Instrument AgWeight @@;
   datalines;
1 107.8681568   1 107.8681465   1 107.8681572   1 107.8681785
1 107.8681446   1 107.8681903   1 107.8681526   1 107.8681494
1 107.8681616   1 107.8681587   1 107.8681519   1 107.8681486
1 107.8681419   1 107.8681569   1 107.8681508   1 107.8681672
1 107.8681385   1 107.8681518   1 107.8681662   1 107.8681424
1 107.8681360   1 107.8681333   1 107.8681610   1 107.8681477
2 107.8681079   2 107.8681344   2 107.8681513   2 107.8681197
2 107.8681604   2 107.8681385   2 107.8681642   2 107.8681365
```

```
2 107.8681151   2 107.8681082   2 107.8681517   2 107.8681448
2 107.8681198   2 107.8681482   2 107.8681334   2 107.8681609
2 107.8681101   2 107.8681512   2 107.8681469   2 107.8681360
2 107.8681254   2 107.8681261   2 107.8681450   2 107.8681368
;
```

Notice that the variation in the atomic weight measurements is several orders of magnitude less than their mean. This is a situation that can be difficult for standard, regression-based analysis-of-variance procedures to handle correctly.

The following statements invoke the ORTHOREG procedure to perform a simple one-way analysis of variance, testing for differences between the two instruments:

```
proc orthoreg data=AgWeight;
   class Instrument;
   model AgWeight = Instrument;
run;
```

Output 63.1.1 shows the resulting analysis.

Output 63.1.1 PROC ORTHOREG Results for Atomic Weight Example

```
              Atomic Weight of Silver by Two Different Instruments

                           The ORTHOREG Procedure

                           Class Level Information
                         Factor        Levels    -Values-

                         Instrument         2    1 2

              Atomic Weight of Silver by Two Different Instruments

                           The ORTHOREG Procedure

                       Dependent Variable: AgWeight

                                    Sum of
 Source                    DF       Squares     Mean Square    F Value    Pr > F

 Model                      1    3.6383419E-9    3.6383419E-9      15.95    0.0002
 Error                     46    1.0495173E-8    2.281559E-10
 Corrected Total           47    1.4133515E-8

                          Root MSE    0.0000151048
                          R-Square    0.2574265445


                                                      Standard
 Parameter            DF   Parameter Estimate          Error   t Value   Pr > |t|

 Intercept             1     107.868136354166   3.0832608E-6   3.499E7     <.0001
 (Instrument='1')      1     0.00001741249999   4.3603893E-6      3.99     0.0002
 (Instrument='2')      0                    0              .         .          .
```

The mean difference between instruments is about 1.74×10^{-5} (the value of the **(Instrument='1')** parameter in the parameter estimates table), whereas the level of background variation in the measurements is about 1.51×10^{-5} (the value of the root mean squared error). At this level of error, the difference is significant, with a p-value of 0.0002.

The National Institute of Standards and Technology (1998) has provided certified ANOVA values for this data set. The following statements use ODS to examine the ANOVA values produced by both the ORTHOREG and GLM procedures more precisely for comparison with the NIST-certified values:

```
ods listing close;
ods output ANOVA         = OrthoregANOVA
           FitStatistics = OrthoregFitStat;

proc orthoreg data=AgWeight;
   class Instrument;
   model AgWeight = Instrument;
run;

ods output OverallANOVA  = GLMANOVA
           FitStatistics = GLMFitStat;
proc glm data=AgWeight;
   class Instrument;
   model AgWeight = Instrument;
run;
ods listing;

data _null_; set OrthoregANOVA  (in=inANOVA)
                 OrthoregFitStat(in=inFitStat);
   if (inANOVA) then do;
      if (Source = 'Model') then put "Model SS: " ss e20.;
      if (Source = 'Error') then put "Error SS: " ss e20.;
   end;
   if (inFitStat) then do;
      if (Statistic = 'Root MSE') then
                           put "Root MSE: " nValue1 e20.;
      if (Statistic = 'R-Square') then
                         put "R-Square: " nValue1 best20.;
   end;
data _null_; set GLMANOVA  (in=inANOVA)
                 GLMFitStat(in=inFitStat);
   if (inANOVA) then do;
      if (Source = 'Model') then put "Model SS: " ss e20.;
      if (Source = 'Error') then put "Error SS: " ss e20.;
   end;
   if (inFitStat) then      put "Root MSE: " RootMSE e20.;
   if (inFitStat) then   put "R-Square: " RSquare best20.;
run;
```

In SAS/STAT software prior to SAS 8, PROC GLM gave much less accurate results than PROC ORTHOREG. Table 63.2 and Table 63.3 compare the ANOVA values certified by NIST with those produced by the two procedures.

Table 63.2 Accuracy Comparison for Sums of Squares

Values	Model SS	Error SS
NIST-certified	3.6383418750000E−09	1.0495172916667E−08
ORTHOREG	3.6383418747907E−09	1.0495172916797E−08
GLM, since SAS 8	3.6383418747907E−09	1.0495172916797E−08
GLM, before SAS 8	0	1.0331496763990E−08

Table 63.3 Accuracy Comparison for Fit Statistics

Values	Root MSE	R-Square
NIST-certified	1.5104831444641E−05	0.25742654453832
ORTHOREG	1.5104831444735E−05	0.25742654452494
GLM, since SAS 8	1.5104831444735E−05	0.25742654452494
GLM, before SAS 8	1.4986585859992E−05	0

While the PROC ORTHOREG values and the PROC GLM values for the current version are quite close to the certified ones, the PROC GLM values for releases prior to SAS 8 are not. In fact, since the model sum of squares is so small, in prior releases the GLM procedure set it (and consequently R^2) to zero.

Example 63.2: Wampler Data

This example applies the ORTHOREG procedure to a collection of data sets noted for being ill conditioned. The OUTEST= data set is used to collect the results for comparison with values certified to be correct by the National Institute of Standards and Technology (1998).

NOTE: The results from this example vary from machine to machine depending on floating-point configuration.

The data are from Wampler (1970). The independent variates for all five data sets are x^i, $i = 1, \ldots 5$, for $x = 0, 1, \ldots, 20$. Two of the five dependent variables are exact linear functions of the independent terms:

$$
\begin{aligned}
y_1 &= 1 + x + x^2 + x^3 + x^4 + x^5 \\
y_2 &= 1 + 0.1x + 0.01x^2 + 0.001x^3 + 0.0001x^4 + 0.00001x^5
\end{aligned}
$$

The other three dependent variables have the same mean value as y_1, but with nonzero errors:

$$
\begin{aligned}
y_3 &= y_1 + \mathbf{e} \\
y_4 &= y_1 + 100\mathbf{e} \\
y_5 &= y_1 + 10000\mathbf{e}
\end{aligned}
$$

where **e** is a vector of values with standard deviation $\sim$2044, chosen to be orthogonal to the mean model for y_1.

The following statements create a SAS data set Wampler containing the Wampler data, run a SAS macro program that uses PROC ORTHOREG to fit a fifth-order polynomial in x to each of the Wampler dependent variables, and collect the results in a data set named ParmEst:

```
data Wampler;
   do x=0 to 20;
      input e @@;
      y1 = 1 +       x    +        x**2 +      x**3
             +       x**4 +        x**5;
      y2 = 1 + .1   *x    + .01   *x**2 + .001*x**3
             + .0001*x**4 + .00001*x**5;
      y3 = y1 +       e;
      y4 = y1 +   100*e;
      y5 = y1 + 10000*e;
      output;
   end;
   datalines;
759 -2048 2048 -2048 2523 -2048 2048 -2048 1838 -2048 2048
-2048 1838 -2048 2048 -2048 2523 -2048 2048 -2048 759
;

%macro WTest;
   data ParmEst; if (0); run;
   %do i = 1 %to 5;
      proc orthoreg data=Wampler outest=ParmEst&i noprint;
         model y&i = x x*x x*x*x x*x*x*x x*x*x*x*x;
      data ParmEst&i; set ParmEst&i; Dep = "y&i";
      data ParmEst; set ParmEst ParmEst&i;
         label Col1='x'     Col2='x**2'  Col3='x**3'
               Col4='x**4'  Col5='x**5';
      run;
   %end;
%mend;
%WTest;
```

Instead of displaying the raw values of the RMSE and parameter estimates, use an additional DATA step as follows to compute the deviations from the values certified to be correct by the National Institute of Standards and Technology (1998):

```
data ParmEst; set ParmEst;
   if      (Dep = 'y1') then
      _RMSE_ = _RMSE_ - 0.00000000000000;
   else if (Dep = 'y2') then
      _RMSE_ = _RMSE_ - 0.00000000000000;
   else if (Dep = 'y3') then
      _RMSE_ = _RMSE_ - 2360.14502379268;
   else if (Dep = 'y4') then
      _RMSE_ = _RMSE_ - 236014.502379268;
   else if (Dep = 'y5') then
      _RMSE_ = _RMSE_ - 23601450.2379268;
   if (Dep ^= 'y2') then do;
      Intercept = Intercept - 1.00000000000000;
      Col1      = Col1      - 1.00000000000000;
      Col2      = Col2      - 1.00000000000000;
      Col3      = Col3      - 1.00000000000000;
      Col4      = Col4      - 1.00000000000000;
      Col5      = Col5      - 1.00000000000000;
   end;
   else do;
      Intercept = Intercept - 1.00000000000000;
      Col1      = Col1      - 0.100000000000000;
      Col2      = Col2      - 0.100000000000000e-1;
      Col3      = Col3      - 0.100000000000000e-2;
      Col4      = Col4      - 0.100000000000000e-3;
      Col5      = Col5      - 0.100000000000000e-4;
   end;
run;

proc print data=ParmEst label noobs;
   title 'Wampler data: Deviations from Certified Values';
   format _RMSE_ Intercept Col1-Col5 e9.;
   var Dep _RMSE_ Intercept Col1-Col5;
run;
```

The results, shown in Output 63.2.1, indicate that the values computed by PROC ORTHOREG are quite close to the NIST-certified values.

Output 63.2.1 Wampler Data: Deviations from Certified Values

Wampler data: Deviations from Certified Values

Dep	_RMSE_	Intercept	x	x**2	x**3	x**4	x**5
y1	0.00E+00	5.46E-12	-9.82E-11	1.55E-11	-5.68E-13	3.55E-14	-6.66E-16
y2	0.00E+00	8.88E-16	-3.19E-15	1.24E-15	-1.88E-16	1.20E-17	-2.57E-19
y3	-2.09E-11	-7.73E-11	1.46E-11	-2.09E-11	2.50E-12	-1.28E-13	2.66E-15
y4	-4.07E-10	-5.38E-10	8.99E-10	-3.29E-10	4.23E-11	-2.27E-12	4.35E-14
y5	-3.35E-08	-4.10E-08	8.07E-08	-2.77E-08	3.54E-09	-1.90E-10	3.64E-12

References

Beaton, A. E., Rubin, D. B., and Barone, J. L. (1976), "The Acceptability of Regression Solutions: Another Look at Computational Accuracy," *Journal of the American Statistical Association*, 71, 158–168.

Gentleman, W. M. (1972), *Basic Procedures for Large, Sparse, or Weighted Least Squares Problems*, Technical Report CSRR-2068, University of Waterloo, Ontario.

Gentleman, W. M. (1973), "Least Squares Computations by Givens Transformations without Square Roots," *J. Inst. Math. Appl.*, 12, 329–336.

Lawson, C. L. and Hanson, R. J. (1974), *Solving Least Squares Problems*, Englewood Cliffs, NJ: Prentice-Hall.

Longley, J. W. (1967), "An Appraisal of Least Squares Programs for the Electronic Computer from the Point of View of the User," *Journal of the American Statistical Association*, 62, 819–841.

National Institute of Standards and Technology (1998), *Statistical Reference Data Sets*, last accessed April 4, 2007.
URL http://www.nist.gov/itl/div898/strd

Powell, L. J., Murphy, T. J., and Gramlich, J. W. (1982), "The Absolute Isotopic Abundance and Atomic Weight of a Reference Sample of Silver," *NBS Journal of Research*, 87, 9–19.

Wampler, R. H. (1970), "A Report of the Accuracy of Some Widely Used Least Squares Computer Programs," *Journal of the American Statistical Association*, 65, 549–563.

Chapter 64

The PHREG Procedure

Contents

Overview: PHREG Procedure **4523**
Getting Started: PHREG Procedure **4526**
Classical Method of Maximum Likelihood 4527
Bayesian Analysis 4531
Syntax: PHREG Procedure **4535**
PROC PHREG Statement 4536
ASSESS Statement 4540
BASELINE Statement 4541
BAYES Statement 4545
BY Statement 4556
CLASS Statement 4557
CONTRAST Statement 4559
FREQ Statement 4561
HAZARDRATIO Statement 4562
ID Statement 4564
MODEL Statement 4564
OUTPUT Statement 4573
Programming Statements 4575
STRATA Statement 4577
TEST Statement 4577
WEIGHT Statement 4578
Details: PHREG Procedure **4579**
Failure Time Distribution 4579
CLASS Variable Parameterization 4580
Clarification of the Time and CLASS Variables Usage 4582
Partial Likelihood Function for the Cox Model 4587
Counting Process Style of Input 4588
Left Truncation of Failure Times 4589
The Multiplicative Hazards Model 4590
Hazard Ratios 4590
Specifics for Classical Analysis 4593
Proportional Rates/Means Models for Recurrent Events 4593
Newton-Raphson Method 4595
Firth's Modification for Maximum Likelihood Estimation 4595

Robust Sandwich Variance Estimate 4597
Testing the Global Null Hypothesis 4597
Confidence Limits for a Hazard Ratio 4598
Testing Linear Hypotheses about Regression Coefficients 4600
Analysis of Multivariate Failure Time Data 4601
Model Fit Statistics 4610
Residuals 4611
Diagnostics Based on Weighted Residuals 4613
Influence of Observations on Overall Fit of the Model 4614
Survivor Function Estimation for the Cox Model 4615
Effect Selection Methods 4617
Assessment of the Proportional Hazards Model 4618
Specifics for Bayesian Analysis 4620
Piecewise Constant Baseline Hazard Model 4621
Priors for Model Parameters 4623
Posterior Distribution 4625
Sampling from the Posterior Distribution 4626
Starting Values of the Markov Chains 4627
Fit Statistics 4628
Posterior Distribution for Quantities of Interest 4628
Computational Resources 4630
Input and Output Data Sets 4630
OUTEST= Output Data Set 4630
INEST= Input Data Set 4631
OUT= Output Data Set in the OUTPUT Statement 4631
OUT= Output Data Set in the BASELINE Statement 4632
OUTPOST= Output Data Set in the BAYES Statement 4632
Displayed Output 4632
Maximum Likelihood Analysis Displayed Output 4632
Bayesian Analysis Displayed Output 4637
ODS Table Names 4640
ODS Graphics 4643
Examples: PHREG Procedure **4644**
Example 64.1: Stepwise Regression 4644
Example 64.2: Best Subset Selection 4652
Example 64.3: Modeling with Categorical Predictors 4654
Example 64.4: Firth's Correction for Monotone Likelihood 4662
Example 64.5: Conditional Logistic Regression for m:n Matching 4664
Example 64.6: Model Using Time-Dependent Explanatory Variables 4668
Example 64.7: Time-Dependent Repeated Measurements of a Covariate . . 4675
Example 64.8: Survivor Function Estimates for Specific Covariate Values . 4683
Example 64.9: Analysis of Residuals 4685
Example 64.10: Analysis of Recurrent Events Data 4687
Example 64.11: Analysis of Clustered Data 4697

Example 64.12: Model Assessment Using Cumulative Sums of Martingale Residuals 4700
Example 64.13: Bayesian Analysis of the Cox Model 4712
Example 64.14: Bayesian Analysis of Piecewise Exponential Model 4723
References **4727**

Overview: PHREG Procedure

The analysis of survival data requires special techniques because the data are almost always incomplete, and familiar parametric assumptions might be unjustifiable. Investigators follow subjects until they reach a prespecified endpoint (for example, death). However, subjects sometimes withdraw from a study, or the study is completed before the endpoint is reached. In these cases, the survival times (also known as failure times) are *censored*; subjects survived to a certain time beyond which their status is unknown. The uncensored survival times are sometimes referred to as *event* times. Methods of survival analysis must account for both censored and uncensored data.

Many types of models have been used for survival data. Two of the more popular types of models are the accelerated failure time model (Kalbfleisch and Prentice 1980) and the Cox proportional hazards model (Cox 1972). Each has its own assumptions about the underlying distribution of the survival times. Two closely related functions often used to describe the distribution of survival times are the survivor function and the hazard function (see the section "Failure Time Distribution" on page 4579 for definitions). The accelerated failure time model assumes a parametric form for the effects of the explanatory variables and usually assumes a parametric form for the underlying survivor function. Cox's proportional hazards model also assumes a parametric form for the effects of the explanatory variables, but it allows an unspecified form for the underlying survivor function.

The PHREG procedure performs regression analysis of survival data based on the Cox proportional hazards model. Cox's semiparametric model is widely used in the analysis of survival data to explain the effect of explanatory variables on hazard rates.

The survival time of each member of a population is assumed to follow its own hazard function, $\lambda_i(t)$, expressed as

$$\lambda_i(t) = \lambda(t; \mathbf{Z}_i) = \lambda_0(t) \exp(\mathbf{Z}_i' \boldsymbol{\beta})$$

where $\lambda_0(t)$ is an arbitrary and unspecified baseline hazard function, $\mathbf{Z}_i$ is the vector of explanatory variables for the ith individual, and $\boldsymbol{\beta}$ is the vector of unknown regression parameters associated with the explanatory variables. The vector $\boldsymbol{\beta}$ is assumed to be the same for all individuals. The survivor function can be expressed as

$$S(t; \mathbf{Z}_i) = [S_0(t)]^{\exp(\mathbf{Z}_i' \boldsymbol{\beta})}$$

where $S_0(t) = \exp(-\int_0^t \lambda_0(u)du)$ is the baseline survivor function. To estimate $\boldsymbol{\beta}$, Cox (1972, 1975) introduced the partial likelihood function, which eliminates the unknown baseline hazard $\lambda_0(t)$ and accounts for censored survival times.

The partial likelihood of Cox also allows time-dependent explanatory variables. An explanatory variable is time-dependent if its value for any given individual can change over time. Time-dependent variables have many useful applications in survival analysis. You can use a time-dependent variable to model the effect of subjects changing treatment groups. Or you can include time-dependent variables such as blood pressure or blood chemistry measures that vary with time during the course of a study. You can also use time-dependent variables to test the validity of the proportional hazards model.

An alternative way to fit models with time-dependent explanatory variables is to use the counting process style of input. The counting process formulation enables PROC PHREG to fit a superset of the Cox model, known as the multiplicative hazards model. This extension also includes recurrent events data and left truncation of failure times. The theory of these models is based on the counting process pioneered by Andersen and Gill (1982), and the model is often referred to as the Andersen-Gill model.

Multivariate failure time data arise when each study subject can potentially experience several events (for instance, multiple infections after surgery) or when there exists some natural or artificial clustering of subjects (for instance, a litter of mice) that induces dependence among the failure times of the same cluster. Data in the former situation are referred to as multiple events data, which include recurrent events data as a special case; data in the latter situation are referred to as clustered data. You can use PROC PHREG to carry out various methods of analyzing these data.

The population under study can consist of a number of subpopulations, each of which has its own baseline hazard function. PROC PHREG performs a stratified analysis to adjust for such subpopulation differences. Under the stratified model, the hazard function for the *j*th individual in the *i*th stratum is expressed as

$$\lambda_{ij}(t) = \lambda_{i0}(t) \exp(\mathbf{Z}_{ij}'\boldsymbol{\beta})$$

where $\lambda_{i0}(t)$ is the baseline hazard function for the *i*th stratum, and $\mathbf{Z}_{ij}$ is the vector of explanatory variables for the individual. The regression coefficients are assumed to be the same for all individuals across all strata.

Ties in the failure times can arise when the time scale is genuinely discrete or when survival times generated from the continuous-time model are grouped into coarser units. The PHREG procedure includes four methods of handling ties. The *discrete* logistic model is available for discrete time-scale data. The other three methods apply to continuous time-scale data. The *exact* method computes the exact conditional probability under the model that the set of observed tied event times occurs before all the censored times with the same value or before larger values. *Breslow* and *Efron* methods provide approximations to the exact method.

Variable selection is a typical exploratory exercise in multiple regression when the investigator is interested in identifying important prognostic factors from a large number of candidate variables. The PHREG procedure provides four selection methods: forward selection, backward elimination, stepwise selection, and best subset selection. The best subset selection method is based on the likelihood score statistic. This method identifies a specified number of best models containing one, two, or three variables and so on, up to the single model containing all of the explanatory variables.

The PHREG procedure also enables you to include an offset variable in the model; to weight the observations in the input data; to test linear hypotheses about the regression parameters; to perform conditional logistic regression analysis for matched case-control studies; to output survivor function

estimates, residuals, and regression diagnostics; and to estimate the survivor function for a new set of covariates.

PROC PHREG can also be used to fit the multinomial logit choice model to discrete choice data. See `http://support.sas.com/resources/papers/tnote/tnote_marketresearch.html` for more information about discrete choice modeling and the multinomial logit model. Look for the "Discrete Choice" report.

The PHREG procedure now uses ODS Graphics to create graphs as part of its output. For example, the ASSESS statement uses a graphical method that uses ODS Graphics to check the adequacy of the model. See Chapter 21, "Statistical Graphics Using ODS," for general information about ODS Graphics.

There have been a number of enhancements to PROC PHREG with this release. The most noticeable additions are the CLASS statement for specifying categorical variables; the CONTRAST statement for estimating and testing linear contrasts; the BAYES statement for performing a Bayesian analysis; and the HAZARDRATIO statement for estimating customized hazard ratios.

The CLASS statement enables you to specify categorical variables (also known as factors or CLASS variables) to be used in the analysis. Model effects, including covariates, main effects (CLASS variables), crossed effects (interactions), and nested effects, can be specified in the same way as in the GLM procedure. The CLASS statement supports less-than-full-rank parameterization as in the GLM procedure as well as various full-rank parameterization methods such as reference coding, effect coding, and orthogonal polynomial coding. For some of the full-rank coding schemes, you can designate a specific value (category or level) of the CLASS variable as the reference level. The CLASS statement also enables you to specify the ordering of the categories of CLASS variables, to reverse the ordering of the categories, and to treat categories with missing values as valid categories.

With the new way of specifying model effects, the CONTRAST statement enables you to test customized hypotheses concerning the regression parameters. Each CONTRAST statement also provides estimation of individual rows of contrasts, which is particularly useful in comparing the hazards between the categories of a CLASS variable.

The BAYES statement invokes a Bayesian analysis of the Cox model or the piecewise constant baseline hazard model (also known as the piecewise exponential model). In essence, the Bayesian paradigm treats parameters as random variables, and inference (measurement of uncertainty) about parameters is based on the posterior distribution of the parameters. A posterior distribution is a weighted likelihood function of the data with a prior distribution that uses the Bayes theorem. Without any past experience or knowledge of what prior distribution to use, you can always start with a noninformative prior. Knowledge of the prior is accumulated over time, and the Bayesian approach can be viewed as a process of learning from experience. A closed form of the posterior distribution is hard to come by, and a Markov chain Monte Carlo method is used to simulate samples from the posterior distribution. See Chapter 7, "Introduction to Bayesian Analysis Procedures," for an introduction to the basic concepts in Bayesian statistics. You can also refer to the section "Bayesian Analysis: Advantages and Disadvantages" on page 149 for a discussion of the advantages and disadvantages of Bayesian analysis. For the Cox model, the partial likelihood is used as the likelihood, which is justified by Sinha, Ibrahim, and Chen (2003). PROC PHREG generates a chain of posterior distribution samples by the Gibbs sampler, using the adaptive rejection sampling algorithm (Gilks and Wild 1992; Gilks, Best, and Tan 1995) to sample each parameter value from its full conditional distribution. Summary statistics (mean, standard deviation, percentiles, HPD

intervals, and equal-tail credible intervals) and convergence diagnostics (autocorrelations; Gelman-Rubin, Geweke, Raftery-Lewis, and Heidelberger-Welch tests; and the effective sample size) are computed for each parameter, as well as the covariance and correlation matrices of the posterior samples. Trace plots, posterior density plots, and autocorrelation function plots are also provided using ODS Graphics.

The HAZARDRATIO statement identifies the variable for which hazard ratios are to be evaluated. For a continuous variable, the hazard ratio compares the hazards for a given change in the variable. For a CLASS variable, a hazard ratio compares the hazards of two levels of the variable. The HAZARDRATIO statement enables you to obtain hazard ratios even in the presence of interactions and nested effects.

Other enhancements include plotting the baseline functions through ODS Graphics, computing profile-likelihood-based confidence limits for hazard ratios, and allowing the bias-reducing penalized likelihood optimization as discussed by Firth (1993) and Heinze and Schemper (2001).

For both the BASELINE and OUTPUT statements, the default method of estimating a survivor function is now based on the Breslow (1972) estimator—that is, METHOD=CH. There are a few other changes in the BASELINE statement with this release. The option NOMEAN in the past releases has become obsolete—that is, requested statistics at the sample average values of the covariates are no longer computed and added to the OUT= data set. However, if the COVARIATES= data set is not specified, the requested statistics are computed and output for the covariate set that consists of the reference levels for the CLASS variables and sample averages for the continuous variable. In addition to the requested statistics, the OUT= data set also contains all variables in the COVARIATES= data set.

The remaining sections of this chapter contain information about how to use PROC PHREG, information about the underlying statistical methodology, and some sample applications of the procedure. The section "Getting Started: PHREG Procedure" on page 4526 introduces PROC PHREG with two examples. The section "Syntax: PHREG Procedure" on page 4535 describes the syntax of the procedure. The section "Details: PHREG Procedure" on page 4579 summarizes the statistical techniques employed in PROC PHREG. The section "Examples: PHREG Procedure" on page 4644 includes eight additional examples of useful applications. Experienced SAS/STAT software users might decide to proceed to the "Syntax" section, while other users might choose to read both the "Getting Started" and "Examples" sections before proceeding to "Syntax" and "Details."

Getting Started: PHREG Procedure

This section uses the two-sample vaginal cancer mortality data from Kalbfleisch and Prentice (1980, p. 2) in two examples to illustrate some of the basic features of PROC PHREG. The first example carries out a classical Cox regression analysis and the second example performs a Bayesian analysis of the Cox model.

Two groups of rats received different pretreatment regimes and then were exposed to a carcinogen. Investigators recorded the survival times of the rats from exposure to mortality from vaginal cancer. Four rats died of other causes, so their survival times are censored. Interest lies in whether the

survival curves differ between the two groups.

The following DATA step creates the data set Rats, which contains the variable Days (the survival time in days), the variable Status (the censoring indicator variable: 0 if censored and 1 if not censored), and the variable Group (the pretreatment group indicator).

```
data Rats;
   label Days  ='Days from Exposure to Death';
   input Days Status Group @@;
   datalines;
143 1 0   164 1 0   188 1 0   188 1 0
190 1 0   192 1 0   206 1 0   209 1 0
213 1 0   216 1 0   220 1 0   227 1 0
230 1 0   234 1 0   246 1 0   265 1 0
304 1 0   216 0 0   244 0 0   142 1 1
156 1 1   163 1 1   198 1 1   205 1 1
232 1 1   232 1 1   233 1 1   233 1 1
233 1 1   233 1 1   239 1 1   240 1 1
261 1 1   280 1 1   280 1 1   296 1 1
296 1 1   323 1 1   204 0 1   344 0 1
;
run;
```

By using ODS Graphics, PROC PHREG allows you to plot the survival curve for Group=0 and the survival curve for Group=1, but first you must save these two covariate values in a SAS data set as in the following DATA step:

```
data Regimes;
  Group=0;
  output;
  Group=1;
  output;
run;
```

Classical Method of Maximum Likelihood

PROC PHREG fits the Cox model by maximizing the partial likelihood and computes the baseline survivor function by using the Breslow (1972) estimate. The following statements produce Figure 64.1 and Figure 64.2:

```
ods graphics on;
proc phreg data=Rats plot(overlay)=survival;
   model Days*Status(0)=Group;
   baseline covariates=regimes out=_null_;
run;
ods graphics off;
```

In the MODEL statement, the response variable, Days, is crossed with the censoring variable, Status, with the value that indicates censoring is enclosed in parentheses. The values of Days are considered

censored if the value of Status is 0; otherwise, they are considered event times. Graphics results are enabled through ODS Graphics with the specification of the **ods graphics on** statement. The survival curves for the two observations in the data set Regime, specified in the COVARIATES= option in the BASELINE statement, are requested through the PLOTS= option with the OVERLAY option for overlaying both survival curves in the same plot.

Figure 64.2 shows a typical printed output of a classical analysis. Since Group takes only two values, the null hypothesis for no difference between the two groups is identical to the null hypothesis that the regression coefficient for Group is 0. All three tests in the "Testing Global Null Hypothesis: BETA=0" table (see the section "Testing the Global Null Hypothesis" on page 4597) suggest that the survival curves for the two pretreatment groups might not be the same. In this model, the hazard ratio (or risk ratio) for Group, defined as the exponentiation of the regression coefficient for Group, is the ratio of the hazard functions between the two groups. The estimate is 0.551, implying that the hazard function for Group=1 is smaller than that for Group=0. In other words, rats in Group=1 lived longer than those in Group=0. This conclusion is also revealed in the plot of the survivor functions in Figure 64.2.

Figure 64.1 Comparison of Two Survival Curves

The PHREG Procedure

Model Information

Data Set	WORK.RATS	
Dependent Variable	Days	Days from Exposure to Death
Censoring Variable	Status	
Censoring Value(s)	0	
Ties Handling	BRESLOW	

Number of Observations Read	40
Number of Observations Used	40

Summary of the Number of Event and Censored Values

Total	Event	Censored	Percent Censored
40	36	4	10.00

Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Without Covariates	With Covariates
-2 LOG L	204.317	201.438
AIC	204.317	203.438
SBC	204.317	205.022

Figure 64.1 *continued*

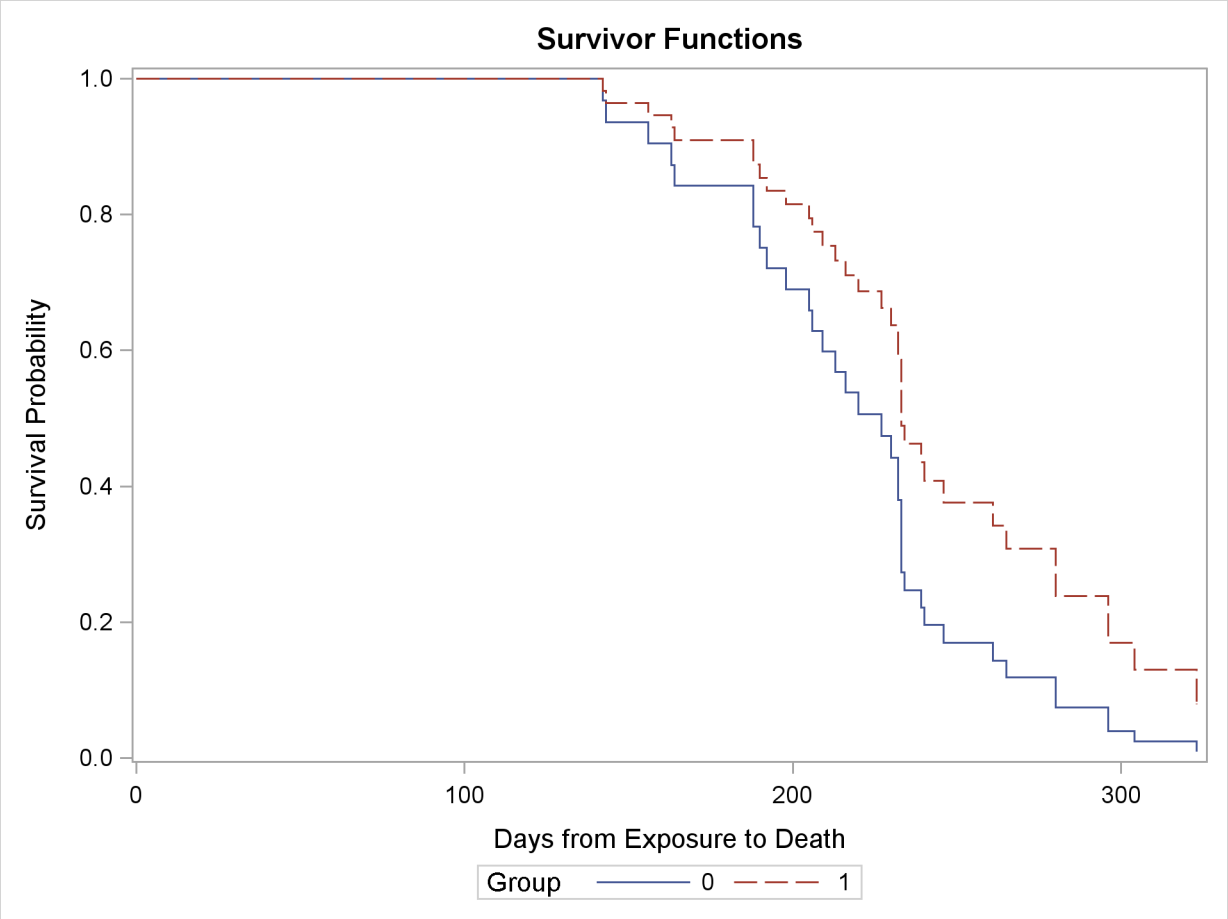
Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	2.8784	1	0.0898
Score	3.0001	1	0.0833
Wald	2.9254	1	0.0872

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
Group	1	-0.59590	0.34840	2.9254	0.0872	0.551

Figure 64.2 Survivorship for the Two Pretreatment Regimes



In this example, the comparison of two survival curves is put in the form of a proportional hazards model. This approach is essentially the same as the log-rank (Mantel-Haenszel) test. In fact, if there are no ties in the survival times, the likelihood score test in the Cox regression analysis is identical

to the log-rank test. The advantage of the Cox regression approach is the ability to adjust for the other variables by including them in the model. For example, the present model could be expanded by including a variable that contains the initial body weights of the rats.

Next, consider a simple test of the validity of the proportional hazards assumption. The proportional hazards model for comparing the two pretreatment groups is given by the following:

$$\lambda(t) = \begin{cases} \lambda_0(t) & \text{if GROUP} = 0 \\ \lambda_0(t)e^{\beta_1} & \text{if GROUP} = 1 \end{cases}$$

The ratio of hazards is e^{β_1}, which does not depend on time. If the hazard ratio changes with time, the proportional hazards model assumption is invalid. Simple forms of departure from the proportional hazards model can be investigated with the following time-dependent explanatory variable $x = x(t)$:

$$x(t) = \begin{cases} 0 & \text{if GROUP} = 0 \\ \log(t) - 5.4 & \text{if GROUP} = 1 \end{cases}$$

Here, $\log(t)$ is used instead of t to avoid numerical instability in the computation. The constant, 5.4, is the average of the logs of the survival times and is included to improve interpretability. The hazard ratio in the two groups then becomes $e^{\beta_1 - 5.4\beta_2} t^{\beta_2}$, where β_2 is the regression parameter for the time-dependent variable x. The term e^{β_1} represents the hazard ratio at the geometric mean of the survival times. A nonzero value of β_2 would imply an increasing ($\beta_2 > 0$) or decreasing ($\beta_2 < 0$) trend in the hazard ratio with time.

The following statements implement this simple test of the proportional hazards assumption. The MODEL statement includes the time-dependent explanatory variable X, which is defined subsequently by the programming statement. At each event time, subjects in the risk set (those alive just before the event time) have their X values changed accordingly.

```
proc phreg data=Rats;
   model Days*Status(0)=Group X;
   X=Group*(log(Days) - 5.4);
run;
```

The analysis of the parameter estimates is displayed in Figure 64.3. The Wald chi-square statistic for testing the null hypothesis that $\beta_2 = 0$ is 0.0158. The statistic is not statistically significant when compared to a chi-square distribution with one degree of freedom ($p = 0.8999$). Thus, you can conclude that there is no evidence of an increasing or decreasing trend over time in the hazard ratio.

Figure 64.3 A Simple Test of Trend in the Hazard Ratio

The PHREG Procedure

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
Group	1	-0.59976	0.34837	2.9639	0.0851	0.549
X	1	-0.22952	1.82489	0.0158	0.8999	0.795

Bayesian Analysis

PROC PHREG uses the partial likelihood of the Cox model as the likelihood and generates a chain of posterior distribution samples by the Gibbs Sampler. Summary statistics, convergence diagnostics, and diagnostic plots are provided for each parameter. The following statements generate Figure 64.4–Figure 64.10:

```
ods graphics on;
proc phreg data=Rats;
   model Days*Status(0)=Group;
   bayes seed=1 outpost=Post;
run;
ods graphics off;
```

The BAYES statement invokes the Bayesian analysis. The SEED= option is specified to maintain reproducibility; the OUTPOST= option saves the posterior distribution samples in a SAS data set for postprocessiong; no other options are specified in the BAYES statement. By default, a uniform prior distribution is assumed on the regression coefficient Group. The uniform prior is a flat prior on the real line with a distribution that reflects ignorance of the location of the parameter, placing equal probability on all possible values the regression coefficient can take. Using the uniform prior in the following example, you would expect the Bayesian estimates to resemble the classical results of maximizing the likelihood. If you can elicit an informative prior on the regression coefficients, you should use the COEFFPRIOR= option to specify it.

You should make sure that the posterior distribution samples have achieved convergence before using them for Bayesian inference. PROC PHREG produces three convergence diagnostics by default. If you enable ODS Graphics before calling PROC PHREG as in the preceding program, diagnostics plots are also displayed.

The results of this analysis are shown in the following figures.

The "Model Information" table in Figure 64.4 summarizes information about the model you fit and the size of the simulation.

Figure 64.4 Model Information

The PHREG Procedure

Bayesian Analysis

Model Information

Data Set	WORK.RATS	
Dependent Variable	Days	Days from Exposure to Death
Censoring Variable	Status	
Censoring Value(s)	0	
Model	Cox	
Ties Handling	BRESLOW	
Burn-In Size	2000	
MC Sample Size	10000	
Thinning	1	

PROC PHREG first fits the Cox model by maximizing the partial likelihood. The only parameter in the model is the regression coefficient of Group. The maximum likelihood estimate (MLE) of the parameter and its 95% confidence interval are shown in Figure 64.5.

Figure 64.5 Classical Parameter Estimates

Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits	
Group	1	−0.5959	0.3484	−1.2788	0.0870

Since no prior is specified for the regression coefficient, the default uniform prior is used. This information is displayed in the "Uniform Prior for Regression Coefficients" table in Figure 64.6.

Figure 64.6 Coefficient Prior

Uniform Prior for Regression Coefficients

Parameter	Prior
Group	Constant

The "Fit Statistics" table in Figure 64.7 lists information about the fitted model. The table displays the DIC (deviance information criterion) and pD (effective number of parameters). See the section "Fit Statistics" on page 4628 for details.

Figure 64.7 Fit Statistics

Fit Statistics

DIC (smaller is better)	203.444
pD (Effective Number of Parameters)	1.003

Summary statistics of the posterior samples are displayed in the "Posterior Summaries" table and "Posterior Intervals" table as shown in Figure 64.8. Note that the mean and standard deviation of the posterior samples are comparable to the MLE and its standard error, respectively, due to the use of the uniform prior.

Figure 64.8 Summary Statistics

The PHREG Procedure

Bayesian Analysis

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
Group	10000	−0.5998	0.3511	−0.8326	−0.5957	−0.3670

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
Group	0.050	−1.3042	0.0721	−1.2984	0.0756

PROC PHREG provides diagnostics to assess the convergence of the generated Markov chain. Figure 64.9 shows three of these diagnostics: the lag1, lag5, lag10, and lag50 autocorrelations; the Geweke diagnostic; and the effective sample size. There is no indication that the Markov chain has not reached convergence. Refer to the section "Statistical Diagnostic Tests" on page 160 for information about interpreting these diagnostics.

Figure 64.9 Convergence Diagnostics

The PHREG Procedure

Bayesian Analysis

Posterior Autocorrelations

Parameter	Lag 1	Lag 5	Lag 10	Lag 50
Group	−0.0079	0.0091	−0.0161	0.0101

Geweke Diagnostics

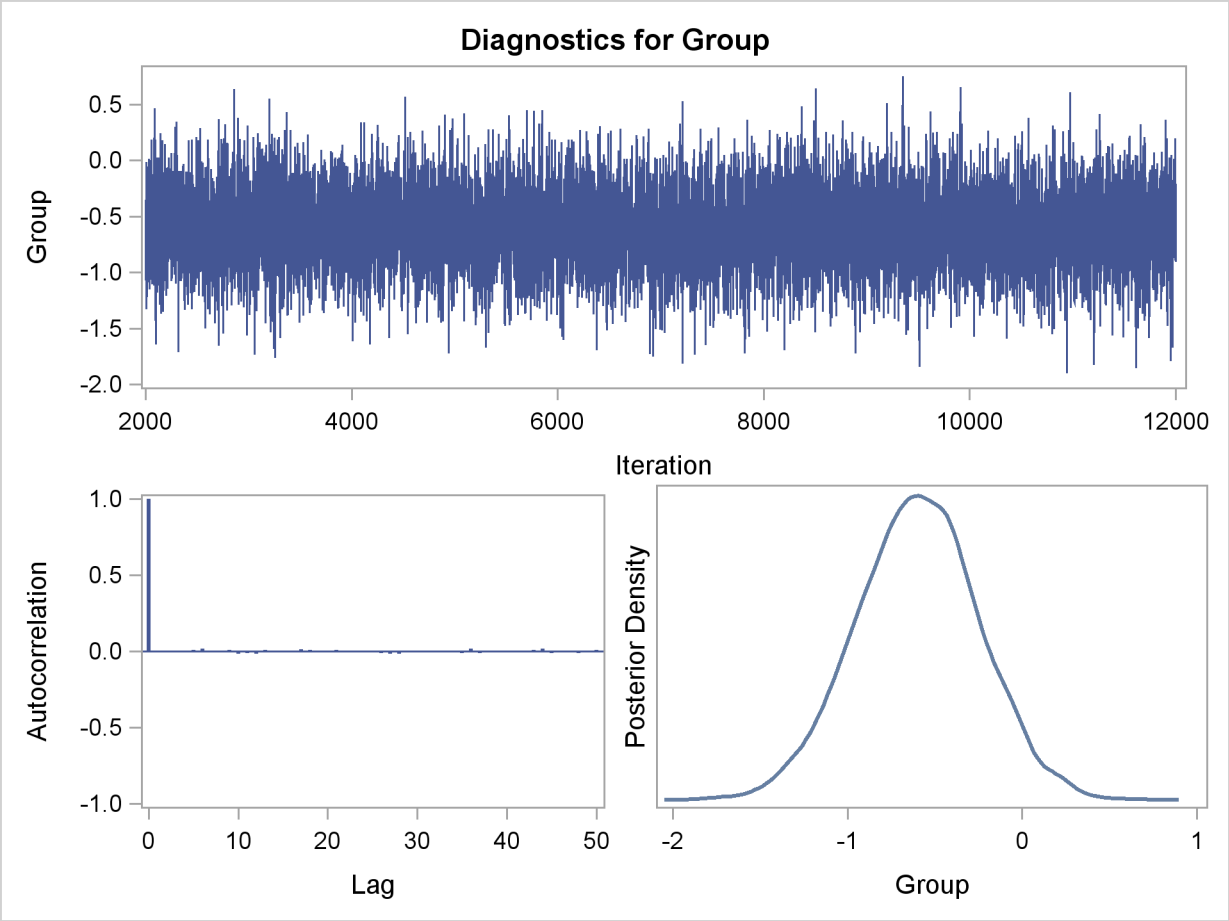
Parameter	z	Pr > |z|
Group	0.0149	0.9881

Effective Sample Sizes

Parameter	ESS	Correlation Time	Efficiency
Group	10000.0	1.0000	1.0000

You can also assess the convergence of the generated Markov chain by examining the trace plot, the autocorrelation function plot, and the posterior density plot. Figure 64.10 displays a panel of these three plots for the parameter Group. This graphical display is automatically produced when ODS Graphics is enabled. Note that the trace of the samples centers on –0.6 with only small fluctuations, the autocorrelations are quite small, and the posterior density appears bell-shaped—all exemplifying the behavior of a converged Markov chain.

Figure 64.10 Diagnostic Plots



The proportional hazards model for comparing the two pretreatment groups is

$$\lambda(t) = \begin{cases} \lambda_0(t) & \text{if Group=0} \\ \lambda_0(t)e^{\beta} & \text{if Group=1} \end{cases}$$

The probability that the hazard of Group=0 is greater than that of Group=1 is

$$\Pr(\lambda_0(t) > \lambda_0(t)e^{\beta}) = \Pr(\beta < 0)$$

This probability can be enumerated from the posterior distribution samples by computing the fraction of samples with a coefficient less than 0. The following DATA step and PROC MEANS perform this calculation:

```
data New;
   set Post;
   Indicator=(Group < 0);
   label Indicator='Group < 0';
   run;
proc means data=New(keep=Indicator) n mean;
   run;
```

Figure 64.11 Prob(Hazard(Group=0) > Hazard(Group=1))

```
                          The MEANS Procedure

                  Analysis Variable : Indicator Group < 0

                             N            Mean
                     ----------------------
                     10000       0.9581000
                     ----------------------
```

The PROC MEANS results are displayed in Output 64.11. There is a 95.8% chance that the hazard rate of Group=0 is greater than that of Group=1. The result is consistent with the fact that the average survival time of Group=0 is less than that of Group=1.

Syntax: PHREG Procedure

The following statements are available in PROC PHREG. Items within < > are optional.

PROC PHREG < *options* > ;
ASSESS *keyword* < */options* > ;
BASELINE < *OUT=SAS-data-set* > < *COVARIATES=SAS-data-set* > < *keyword=name . . . keyword=name* > < */options* > ;
BAYES < *options* > ;
BY *variables* ;
CLASS *variable* < *(options)* > < *. . . variable* < *(options)* > > < */options* > ;
CONTRAST < *'label'* > *effect values* < *,. . . , effect values* > < */options* > ;
FREQ *variable* ;
HAZARDRATIO < *'label'* > *variable* < */options* > ;
ID *variables* ;
MODEL *response* < **censor(list)* > = *variables* < */options* > ;
OUTPUT < *OUT=SAS-data-set* > < *keyword=name . . . keyword=name* > < */options* > ;
programming statements ;
STRATA *variable* < *(list)* > < *. . . variable* < *(list)* > > < */option* > ;
< **label:** >**TEST** *equation1* < *,. . . , equationk* >< */options* > ;
WEIGHT *variable* < */option* > ;

The PROC PHREG and MODEL statements are required. The CLASS statement, if present, must precede the MODEL statement, and the ASSESS or CONTRAST statement, if present, must come after the MODEL statement. The BAYES statement, that invokes a Bayesian analysis, is not compatible with the ASSESS, CONTRAST, ID, OUTPUT, and TEST statements, as well as a number of options in the PROC PHREG and MODEL statements. See the section "Specifics for Bayesian Analysis" on page 4620 for details.

The rest of this section provides detailed syntax information for each statement, beginning with the PROC PHREG statement. The remaining statements are covered in alphabetical order.

PROC PHREG Statement

PROC PHREG < *options* > **;**

You can specify the following options in the PROC PHREG statement.

ALPHA=*number*

specifies the level of significance α for $100(1-\alpha)\%$ confidence intervals. The value *number* must be between 0 and 1; the default value is 0.05, which results in 95% intervals. This value is used as the default confidence level for limits computed by the BASELINE, BAYES, CONTRAST, HAZARDRATIO, and MODEL statements. You can override this default by specifying the ALPHA= option in the separate statements.

COVOUT

adds the estimated covariance matrix of the parameter estimates to the OUTEST= data set. The COVOUT option has no effect unless the OUTEST= option is specified.

COVM

requests that the model-based covariance matrix (which is the inverse of the observed information matrix) be used in the analysis if the COVS option is also specified. The COVM option has no effect if the COVS option is not specified.

COVSANDWICH < (AGGREGATE) >

COVS < (AGGREGATE) >

requests the robust sandwich estimate of Lin and Wei (1989) for the covariance matrix. When this option is specified, this robust sandwich estimate is used in the Wald tests for testing the global null hypothesis, null hypotheses of individual parameters, and the hypotheses in the CONTRAST and TEST statements. In addition, a modified score test is computed in the testing of the global null hypothesis, and the parameter estimates table has an additional StdErrRatio column, which contains the ratios of the robust estimate of the standard error relative to the corresponding model-based estimate. Optionally, you can specify the keyword AGGREGATE enclosed in parentheses after the COVSANDWICH (or COVS) option, which requests a summing up of the score residuals for each distinct ID pattern in the computation of the robust sandwich covariance estimate. This AGGREGATE option has no effects if the ID statement is not specified.

DATA=*SAS-data-set*

names the SAS data set containing the data to be analyzed. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

INEST=*SAS-data-set*

names the SAS data set that contains initial estimates for all the parameters in the model. BY-group processing is allowed in setting up the INEST= data set. See the section "INEST= Input Data Set" on page 4631 for more information.

MULTIPASS

requests that, for each Newton-Raphson iteration, PROC PHREG recompile the risk sets corresponding to the event times for the (start,stop) style of response and recomputes the values of the time-dependent variables defined by the programming statements for each observation in the risk sets. If the MULTIPASS option is not specified, PROC PHREG computes all risk sets and all the variable values and saves them in a utility file. The MULTIPASS option decreases required disk space at the expense of increased execution time; however, for very large data, it might actually save time since it is time-consuming to write and read large utility files. This option has an effect only when the (start,stop) style of response is used or when there are time-dependent explanatory variables.

NAMELEN=*n*

specifies the length of effect names in tables and output data sets to be *n* characters, where *n* is a value between 20 and 200. The default length is 20 characters.

NOPRINT

suppresses all displayed output. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

NOSUMMARY

suppresses the summary display of the event and censored observation frequencies.

OUTEST=*SAS-data-set*

creates an output SAS data set that contains estimates of the regression coefficients. The data set also contains the convergence status and the log likelihood. If you use the COVOUT option, the data set also contains the estimated covariance matrix of the parameter estimators. See the section "OUTEST= Output Data Set" on page 4630 for more information.

PLOTS<(*global-plot-options***)> = ***plot-request*

PLOTS<(*global-plot-options***)> = (***plot-request* < . . . < *plot-request* >**)**

controls the baseline functions plots produced through ODS Graphics. Each observation in the COVARIATES= data set in the BASELINE statement represents a set of covariates for which a curve is produced for each plot request and for each stratum. You can use the ROWID= option in the BASELINE statement to specify a variable in the COVARIATES= data set for identifying the curves produced for the covariate sets. If the ROWID= option is not specified, the curves produced are identified by the covariate values if there is only a single covariate or by the observation numbers of the COVARIATES= data set if the model has two or more covariates. If the COVARIATES= data set is not specified, a reference set of covariates consisting of the reference levels for the CLASS variables and the average values for the continuous variables is used. For plotting more than one curve, you can use the

OVERLAY= option to group the curves in separate plots. When you specify one plot request, you can omit the parentheses around the plot request. Here are some examples:

```
plots=survival
plots=(survival cumhaz)
```

You must enable ODS Graphics before requesting plots, for example, like this:

```
ods graphics on;

proc phreg plots(cl)=survival;
  model Time*Status(0)=X1-X5;
  baseline covariates=One;
run;

ods graphics off;
```

The *global plot options* include the following:

CL<=EQTAIL | HPD>
displays the pointwise interval limits for the specified curves. For the classical analysis, CL displays the confidence limits. For the Bayesian analysis, CL=EQTAIL displays the equal-tail credible limits and CL=HPD displays the HPD limits. Specifying just CL in a Bayesian analysis defaults to CL=HPD.

OVERLAY <=*overlay-option***>**
specifies how the curves for the various strata and covariate sets are overlaid. If the STRATA statement is not specified, specifying OVERLAY without any option will overlay the curves for all the covariate sets. The available *overlay options* are as follows:

BYGROUP
GROUP
overlays, for each stratum, all curves for the covariate sets that have the same GROUP= value in the COVARIATES= data set in the same plot.

INDIVIDUAL
IND
displays, for each stratum, a separate plot for each covariate set.

BYROW
ROW
displays, for each covariate set, a separate plot containing the curves for all the strata.

BYSTRATUM
STRATUM
displays, for each stratum, a separate plot containing the curves for all sets of covariates.

The default is OVERLAY=BYGROUP if the GROUP= option is specified in the BASELINE statement or if the COVARIATES= data set contains the _GROUP_ variable; otherwise the default is OVERLAY=INDIVIDUAL.

TIMERANGE=(< *min* > < *,max* >**)**

TIMERANGE=< *min* > < *,max* >

RANGE=(< *min* > < *,max* >**)**

RANGE=< *min* > < *,max* >

specifies the range of values on the time axis to clip the display. The *min* and *max* values are the lower and upper bounds of the range. By default, *min* is 0 and *max* is the largest event time.

The *plot requests* include the following:

CUMHAZ

plots the estimated cumulative hazard function for each set of covariates in the COVARIATES= data set in the BASELINE statement. If the COVARIATES= data set is not specified, the estimated cumulative hazard function is plotted for the reference set of covariates consisting of reference levels for the CLASS variables and average values for the continuous variables.

MCF

plots the estimated mean cumulative function for each set of covariates in the COVARIATES= data set in the BASELINE statement. If the COVARIATES= data set is not specified, the estimated mean cumulative function is plotted for the reference set of covariates consisting of reference levels for the CLASS variables and average values for the continuous variables.

NONE

suppresses all the plots in the procedure. Specifying this option is equivalent to disabling ODS Graphics for the entire procedure.

SURVIVAL

plots the estimated survivor function for each set of covariates in the COVARIATES= data set in the BASELINE statement. If COVARIATES= data set is not specified, the estimated survivor function is plotted for the reference set of covariates consisting of reference levels for the CLASS variables and average values for the continuous variables.

SIMPLE

displays simple descriptive statistics (mean, standard deviation, minimum, and maximum) for each explanatory variable in the MODEL statement.

ASSESS Statement

ASSESS < **VAR=***(list)* > < **PH** > < */options* > **;**

The ASSESS statement performs the graphical and numerical methods of Lin, Wei, and Ying (1993) for checking the adequacy of the Cox regression model. The methods are derived from cumulative sums of martingale residuals over follow-up times or covariate values. You can assess the functional form of a covariate or you can check the proportional hazards assumption for each covariate in the Cox model. PROC PHREG uses ODS Graphics for the graphical displays. You must specify at least one of the following options to create an analysis.

VAR=(*variable-list***)**
specifies the *list* of explanatory *variables* for which their functional forms are assessed. For each variable on the list, the observed cumulative martingale residuals are plotted against the values of the explanatory variable along with 20 (or *n* if NPATHS=*n* is specified) simulated residual patterns.

PROPORTIONALHAZARDS
PH
requests the checking of the proportional hazards assumption. For each explanatory variable in the model, the observed score process component is plotted against the follow-up time along with 20 (or *n* if NPATHS=*n* is specified) simulated patterns.

The following options can be specified after a slash (/):

NPATHS=*n*
specifies the number of simulated residual patterns to be displayed in a cumulative martingale residual plot or a score process plot. The default is *n*=20.

CRPANEL
requests that a plot with four panels, each containing the observed cumulative martingale residuals and two simulated residual patterns, be created.

RESAMPLE < **=*n*** >
requests that the Kolmogorov-type supremum test be computed on 1,000 simulated patterns or on *n* simulated patterns if *n* is specified.

SEED=*n*
specifies an integer seed for the random number generator used in creating simulated realizations for plots and for the Kolmogorov-type supremum tests. Specifying a seed enables you to reproduce identical graphs and *p*-values for the model assessments from the same PHREG specification. If the SEED= option is not specified, or if you specify a nonpositive seed, a random seed is derived from the time of day.

BASELINE Statement

BASELINE <**OUT=***SAS-data-set*> <**COVARIATES=***SAS-data-set*> <**TIMELIST=***list*> <*keyword=name . . . keyword=name*> </*options*> **;**

The BASELINE statement creates a new SAS data set that contains the baseline function estimates at the event times of each stratum for every set of covariates (**x**) given in the COVARIATES= data set. If the COVARIATES= data set is not specified, a reference set of covariates consisting of the reference levels for the CLASS variables and the average values for the continuous variables is used. No BASELINE data set is created if the model contains a time-dependent variable defined by means of programming statement.

The following options are available in the BASELINE statement.

OUT=*SAS-data-set*

names the output BASELINE data set. If you omit the OUT= option, the data set is created and given a default name by using the DATA*n* convention. See the section "OUT= Output Data Set in the BASELINE Statement" on page 4632 for more information.

COVARIATES=*SAS-data-set*

names the SAS data set that contains the sets of explanatory variable values for which the quantities of interest are estimated. All variables in the COVARIATES= data set are copied to the OUT= data set. Thus, any variable in the COVARIATES= data set can be used to identify the covariate sets in the OUT= data set.

TIMELIST=*list*

specifies a list of time points at which the survival function estimates, cumulative function estimates, or MCF estimates are computed. The following specifications are equivalent:

```
timelist=5,20 to 50 by 10
timelist= 5 20 30 40 50
```

If the TIMELIST= option is not specified, the default is to carry out the prediction at all event times and at time 0. This option can be used only for the Bayesian analysis.

keyword=name

specifies the statistics to be included in the OUT= data set and assigns names to the variables that contain these statistics. Specify a *keyword* for each desired statistic, an equal sign, and the name of the variable for the statistic. Not all *keywords* listed in Table 64.1 (and discussed in the text that follows) are appropriate for both the classical analysis and the Bayesian analysis; and the table summaries the choices for each analysis.

Table 64.1 Summary of the Keyword Choices

Keyword	Classical	Bayesian
Survivor Function		
SURVIVAL	x	x
STDERR	x	x
LOWER	x	x

Table 64.1 *continued*

Options	Classical	Bayesian
UPPER	x	x
LOWERHPD		x
UPPERHPD		x
Cumulative Hazard Function		
CUMHAZ	x	x
STDCUMHAZ	x	x
LOWERCUMHAZ	x	x
UPPERCUMHAZ	x	x
LOWERHPDCUMHAZ		x
UPPERHPDCUMHAZ		x
Cumulative Mean Function		
CMF	x	
STDCMF	x	
LOWERCMF	x	
UPPERCMF	x	
Others		
XBETA	x	x
STDXBETA	x	x
LOGSURV	x	
LOGLOGS	x	

The available *keywords* are as follows.

CMF
MCF

specifies the cumulative mean function estimate for recurrent events data. Specifying CMF=_ALL_ is equivalent to specifying CMF=CMF, STDCMF=StdErrCMF, LOWERCMF=LowerCMF, and UPPERCMF=UpperCMF. Nelson (2002) refers to the mean function estimate as MCF (mean cumulative function).

CUMHAZ

specifies the cumulative hazard function estimate. Specifying CUMHAZ=_ALL_ is equivalent to specifying CUMHAZ=CumHaz, STDCUMHAZ=StdErrCumHaz, LOWERCUMHAZ=LowerCumHaz, and UPPERCUMHAZ=UpperCumHaz. For a Bayesian analysis, CUMHAZ=_ALL_ also includes LOWERHPDCUMHAZ=LowerHPDCumHaz and UpperHPDCUMHAZ=UpperHPDCumHaz.

LOGLOGS

specifies the log of the negative log of SURVIVAL.

LOGSURV

specifies the log of SURVIVAL.

LOWER

L

specifies the lower pointwise confidence limit for the survivor function. For a Bayesian analysis, this is the lower limit of the equal-tail credible interval for the survivor function. The confidence level is determined by the ALPHA= option.

LOWERCMF

LOWERMCF

specifies the lower pointwise confidence limit for the cumulative mean function. The confidence level is determined by the ALPHA= option.

LOWERHPD

specifies the lower limit of the HPD interval for the survivor function. The confidence level is determined by the ALPHA= option.

LOWERHPDCUMHAZ

specifies the lower limit of the HPD interval for the cumulative hazard function. The confidence level is determined by the ALPHA= option.

LOWERCUMHAZ

specifies the lower pointwise confidence limit for the cumulative hazard function. For a Bayesian analysis, this is the lower limit of the equal-tail credible interval for the cumulative hazard function. The confidence level is determined by the ALPHA= option.

STDERR

specifies the standard error of the survivor function estimator. For a Bayesian analysis, this is the standard deviation of the posterior distribution of the survivor function.

STDCMF

STDMCF

specifies the estimated standard error of the cumulative mean function estimator.

STDCUMHAZ

specifies the estimated standard error of the cumulative hazard function estimator. For a Bayesian analysis, this is the standard deviation of the posterior distribution of the cumulative hazard function.

STDXBETA

specifies the estimated standard error of the linear predictor estimator. For a Bayesian analysis, this is the standard deviation of the posterior distribution of the linear predictor.

SURVIVAL

specifies the survivor function ($S(t) = [S_0(t)]^{\exp(\boldsymbol{\beta}'\mathbf{x})}$) estimate. Specifying SURVIVAL=_ALL_ is equivalent to specifying SURVIVAL=Survival, STDERR=StdErrSurvival, LOWER=LowerSurvival, and UPPER=UpperSurvival; and for a Bayesian analyis, SURVIVAL=_ALL_ also specifies LOWERHPD= LowerHPDSurvival and UPPERHPD=UpperHPDSurvival.

UPPER

U

specifies the upper pointwise confidence limit for the survivor function. For a Bayesian analysis, this is the upper limit of the equal-tail credible interval for the survivor function. The confidence level is determined by the ALPHA= option.

UPPERCMF

UPPERMCF

specifies the upper pointwise confidence limit for the cumulative mean function. The confidence level is determined by the ALPHA= option.

UPPERCUMHAZ

specifies the upper pointwise confidence limit for the cumulative hazard function. For a Bayesian analysis, this is the upper limit of the equal-tail credible interval for the cumulative hazard function. The confidence level is determined by the ALPHA= option.

UPPERHPD

specifies the upper limit of the equal-tail credible interval for the survivor function. The confidence level is determined by the ALPHA= option.

UPPERHPDCUMHAZ

specifies the upper limit of the equal-tail credible interval for the cumulative hazard function. The confidence level is determined by the ALPHA= option.

XBETA

specifies the estimate of the linear predictor $\mathbf{x}'\boldsymbol{\beta}$.

The following options can appear in the BASELINE statement after a slash (/). The METHOD= and CLTYPE= options apply only to the estimate of the survivor function in the classical analysis. For the Bayesian analysis, the survivor function is estimated by the Breslow (1972) method.

ALPHA=*value*

specifies the significance level of the confidence interval for the survivor function. The value must be between 0 and 1. The default is the value of the ALPHA= option in the PROC PHREG statement, or 0.05 if that option is not specified.

CLTYPE=*method*

specifies the method used to compute the confidence limits for $S(t,\mathbf{z})$, the survivor function for a subject with a fixed covariate vector $\mathbf{z}$ at event time t. The CLTYPE= option can take the following values:

LOG

specifies that the confidence limits for $\log(S(t,\mathbf{z}))$ be computed using the normal theory approximation. The confidence limits for $S(t,\mathbf{z})$ are obtained by back-transforming the confidence limits for $\log(S(t,\mathbf{z}))$. The default is CLTYPE=LOG.

LOGLOG

specifies that the confidence limits for the $\log(-\log(S(t,\mathbf{z})))$ be computed using normal theory approximation. The confidence limits for $S(t,\mathbf{z})$ are obtained by back-transforming the confidence limits for $\log(-\log(S(t,\mathbf{z})))$.

NORMAL
specifies that the confidence limits for $S(t, \mathbf{z})$ be computed directly using normal theory approximation.

GROUP=*variable*
names a numeric variable in the COVARIATES= data set to group the baseline function curves for the observations into separate plots. This option has no effect if the PLOTS= option in the PROC PHREG statement is not specified. Curves for the covariate sets with the same value of the GROUP= variable are overlaid in the same plot.

METHOD=*method*
specifies the method used to compute the survivor function estimates. The two available methods are as follows:

BRESLOW
CH
EMP
NELSON
specifies that the Breslow (1972) method be used to compute the survivor function—that is, that the survivor function be estimated by exponentiating the negative empirical cumulative hazard function.

PL
specifies that the product-limit estimate of the survivor function be computed.

The default is METHOD=BRESLOW.

ROWID=*variable*
ID=*variable*
ROW=*variable*
names a variable in the COVARIATES= data set for identifying the baseline function curves in the plots. This option has no effect if the PLOTS= option in the PROC PHREG statement is not specified. Values of this variable are used to label the curves for the corresponding rows in the COVARIATES= data set. You can specify ROWID=_OBS_ to use the observation numbers in the COVARIATES= data set for identification.

For recurrent events data, both CMF= and CUMHAZ= statistics are the Nelson estimators, but their standard error are not the same. Confidence limits for the cumulative mean function and cumulative hazard function are based on the log transform.

BAYES Statement

BAYES < *options* > **;**

The BAYES statement requests a Bayesian analysis of the regression model by using Gibbs sampling. The Bayesian posterior samples (also known as the chain) for the regression parameters can be output to a SAS data set. Table 64.2 summarizes the options available in the BAYES statement.

Table 64.2 BAYES Statement Options

Option	Description
Monte Carlo Options	
INITIAL=	specifies initial values of the chain
NBI=	specifies the number of burn-in iterations
NMC=	specifies the number of iterations after burn-in
SEED=	specifies the random number generator seed
THINNING=	controls the thinning of the Markov chain
Model and Prior Options	
COEFFPRIOR=	specifies the prior of the regression coefficients
PIECEWISE=	specifies details of the piecewise exponential model
Summaries and Diagnostics of the Posterior Samples	
DIAGNOSTICS=	displays convergence diagnostics
PLOTS=	displays diagnostic plots
STATISTICS=	displays summary statistics
Posterior Samples	
OUTPOST=	names a SAS data set for the posterior samples

The following list describes these options and their suboptions.

COEFFPRIOR=UNIFORM | NORMAL <(*normal-option*)>

CPRIOR=UNIFORM | NORMAL <(*option*)>

COEFF=UNIFORM | NORMAL <(*option*)>

specifies the prior distribution for the regression coefficients. The default is COEFFPRIOR=UNIFORM. The available prior distributions are as follows:

UNIFORM

specifies a flat prior—that is, the prior that is proportional to a constant ($p(\beta_1, \ldots, \beta_k) \propto 1$ for all $-\infty < \beta_i < \infty$).

NORMAL<(*normal-option*)>

specifies a normal distribution. The *normal-options* include the following:

INPUT=*SAS-data-set*

specifies a SAS data set containing the mean and covariance information of the normal prior. The data set must contain the _TYPE_ variable to identify the observation type, and it must contain a variable to represent each regression coefficient. If the data set also contains the _NAME_ variable, values of this variable are used to identify the covariances for the _TYPE_='COV' observations; otherwise, the _TYPE_='COV' observations are assumed to be in the same order as the explanatory variables in the MODEL statement. PROC PHREG reads the mean vector from the observation with _TYPE_='MEAN' and the covariance matrix from observations with _TYPE_='COV'. For an independent normal prior, the variances can be specified with _TYPE_='VAR'; alternatively, the precisions (inverse of the variances) can be specified with _TYPE_='PRECISION'.

RELVAR <=*c*>
specifies a normal prior $N(\mathbf{0}, c\mathbf{J})$, where $\mathbf{J}$ is a diagonal matrix with diagonal elements equal to the variances of the corresponding ML estimator. By default, c=10^6.

VAR=*c*
specifies the normal prior $N(\mathbf{0}, c\mathbf{I})$, where $\mathbf{I}$ is the identity matrix.

If you do not specify a *normal-option*, the normal prior $N(\mathbf{0}, 10^6\mathbf{I})$, where $\mathbf{I}$ is the identity matrix, is used. See the section "Normal Prior" on page 4625 for details.

DIAGNOSTICS=ALL | NONE | *keyword* | *(keyword-list)*
DIAG=ALL | NONE | *keyword* | *(keyword-list)*
controls the number of diagnostics produced. You can request all the diagnostics in the following list by specifying DIAGNOSTICS=ALL. If you do not want any of these diagnostics, you specify DIAGNOSTICS=NONE. If you want some but not all of the diagnostics, or if you want to change certain settings of these diagnostics, you specify a subset of the following keywords. The default is DIAGNOSTICS=(AUTOCORR GEWEKE ESS).

AUTOCORR <(LAGS= *numeric-list*)>
computes the autocorrelations of lags given by LAGS= list for each parameter. Elements in the list are truncated to integers and repeated values are removed. If the LAGS= option is not specified, autocorrections of lags 1, 5, 10, and 50 are computed for each variable. See the section "Autocorrelations" on page 169 for details.

ESS
computes the effective sample size of Kass et al. (1998), the correlation time, and the efficiency of the chain for each parameter. See the section "Effective Sample Size" on page 169 for details.

MCSE
MCERROR
computes the Monte Carlo standard error for each parameter. The Monte Caro standard error, which measures the simulation accuracy, is the standard error of the posterior mean estimate and is calculated as the posterior standard deviation divided by the square root of the effective sample size. See the section "Standard Error of the Mean Estimate" on page 170 for details.

HEIDELBERGER <*(heidel-options)*>
computes the Heidelberger and Welch tests for each parameter. See the section "Heidelberger and Welch Diagnostics" on page 165 for details. The tests consist of a stationary test and a halfwidth test. The former tests the null hypothesis that the sample values form a stationary process. If the stationarity test is passed, a halfwidth test is then carried out. Optionally, you can specify one or more of the following *heidel-options*:

SALPHA=*value*
specifies the α level ($0 < \alpha < 1$) for the stationarity test. The default is the value of the ALPHA= option in the PROC PHREG statement, or 0.05 if that option is not specified.

HALPHA=*value*

specifies the α level ($0 < \alpha < 1$) for the halfwidth test. The default is the value of the ALPHA= option in the PROC PHREG statement, or 0.05 if that option is not specified.

EPS=*value*

specifies a small positive number ϵ such that if the halfwidth is less than ϵ times the sample mean of the retaining samples, the halfwidth test is passed.

GELMAN < =(*gelman-options***) >**

computes the Gelman and Rubin convergence diagnostics. See the section "Gelman and Rubin Diagnostics" on page 161 for details. You can specify one or more of the following *gelman-options*:

NCHAIN=*number*

N=*number*

specifies the number of parallel chains used to compute the diagnostic and has to be 2 or larger. The default is NCHAIN=3. The NCHAIN= option is ignored when the INITIAL= option is specified in the BAYES statement, and in such a case, the number of parallel chains is determined by the number of valid observations in the INITIAL= data set.

ALPHA=*value*

specifies the significance level for the upper bound. The default is the value of the ALPHA= option in the PROC PHREG statement, or 0.05 if that option is not specified (resulting in a 97.5% bound).

.

GEWEKE < =*geweke-options* **>**

computes the Geweke diagnostics. See the section "Geweke Diagnostics" on page 163 for details. The diagnostic is essentially a two-sample t-test between the first f_1 portion and the last f_2 portion of the chain. The default is f_1=0.1 and f_2=0.5, but you can choose other fractions by using the following *geweke-options*:

FRAC1=*value*

specifies the early f_1 fraction of the Markov chain.

FRAC2=*value*

specifies the latter f_2 fraction of the Markov chain.

RAFTERY < (*raftery-options***) >**

computes the Raftery and Lewis diagnostics. See the section "Raftery and Lewis Diagnostics" on page 166 for details. The diagnostic evaluates the accuracy of the estimated quantile ($\hat{\theta}_Q$ for a given $Q \in (0,1)$) of a chain. $\hat{\theta}_Q$ can achieve any degree of accuracy when the chain is allowed to run for a long time. A stopping criterion is when the estimated probability $\hat{P}_Q = \Pr(\theta \le \hat{\theta}_Q)$ reaches within $\pm R$ of the value Q with probability S; that is, $\Pr(Q - R \le \hat{P}_Q \le Q + R) = S$. The following *raftery-options* enable you to specify Q, R, S, and a precision level ϵ for a stationary test.

QUANTILE=*value*

Q=*value*

specifies the order (a value between 0 and 1) of the quantile of interest. The default is 0.025.

ACCURACY=*value*

R=*value*

specifies a small positive number as the margin of error for measuring the accuracy of estimation of the quantile. The default is 0.005.

PROBABILITY=*value*

P=*value*

specifies the probability of attaining the accuracy of the estimation of the quantile. The default is 0.95.

EPSILON=*value*

EPS=*value*

specifies the tolerance level (a small positive number) for the test. The default is 0.001.

INITIAL=*SAS-data-set*

specifies the SAS data set that contains the initial values of the Markov chains. The INITIAL= data set must contain a variable for each parameter in the model. You can specify multiple rows as the initial values of the parallel chains for the Gelman-Rubin statistics, but posterior summary statistics, diagnostics, and plots are computed only for the first chain.

NBI=*number*

specifies the number of burn-in iterations before the chains are saved. The default is 2000.

NMC=*number*

specifies the number of iterations after the burn-in. The default is 10000.

OUTPOST=*SAS-data-set*

OUT=*SAS-data-set*

names the SAS data set that contains the posterior samples. See the section "OUTPOST= Output Data Set in the BAYES Statement" on page 4632 for more information. Alternatively, you can output the posterior samples into a data set, as shown in the following example in which the data set is named PostSamp.

```
ODS OUTPUT PosteriorSample = PostSamp;
```

PIECEWISE <=*keyword* **<(<NINTERVAL=***number***> <INTERVAL=(***numeric-list***)> <PRIOR=***option***>)>>**

specifies that the piecewise constant baseline hazard model be used in the Bayesian analysis. You can specify one of the following two *keywords*:

HAZARD

models the baseline hazard parameters in the original scale. The hazard parameters are named Lambda1, Lambda2, . . ., and so on.

LOGHAZARD
models the baseline hazard parameters in the log scale. The log-hazard parameters are named Alpha1, Alpha2, . . ., and so on.

Specifying PIECEWISE by itself is the same as specifying PIECEWISE=LOGHAZARD.

You can choose one of the following two options to specify the partition of the time axis into intervals of constant baseline hazards:

NINTERVAL=*number*
N=*number*
specifies the number of intervals with constant baseline hazard rates. PROC PHREG partitions the time axis into the given number of intervals with approximately equal number of events in each interval.

INTERVAL=(*numeric-list***)**
specifies the list of numbers that partition the time axis into disjoint intervals with constant baseline hazard in each interval. For example, INTERVAL=(100, 150, 200, 250, 300) specifies a model with a constant hazard in the intervals [0,100), [100,150), [150,200), [200,250), [250,300), and [300,∞). Each interval must contain at least one event; otherwise, the posterior distribution can be improper, and inferences cannot be derived from an improper posterior distribution.

If neither NINTERVAL= nor INTERVAL= is specified, the default is NINTERVAL=8.

To specify the prior for the baseline hazards $(\lambda_1, \ldots, \lambda_J)$ in the original scale, you specify the following:

PRIOR = IMPROPER | UNIFORM | GAMMA< (*gamma-option***) > | ARGAMMA< (***argamma-option***) >**

The default is PRIOR=IMPROPER. The available prior options include the following:

IMPROPER
specifies the noninformative and improper prior $p(\lambda_1, \ldots, \lambda_J) \propto \prod_i {\lambda_i}^{-1}$ for all $\lambda_i > 0$.

UNIFORM
specifies a uniform prior on the real line; that is, $p(\lambda_i) \propto 1$ for all $\lambda_i > 0$.

GAMMA < (*gamma-option***) >**
specifies an independent gamma prior $G(a, b)$ with density $f(t) = \frac{b(bt)^{a-1}\mathrm{e}^{-bt}}{\Gamma(a)}$, which can be followed by one of the following *gamma-options* enclosed in parentheses. The hyperparameters a and b are the shape and inverse-scale parameters of the gamma distribution, respectively. See the section "Independent Gamma Prior" on page 4624 for details. The default is $G(10^{-4}, 10^{-4})$ for each λ_j, setting the prior mean to 1 with variance 10^4. This prior is proper and reasonably noninformative.

INPUT=*SAS-data-set*
specifies a data set containing the hyperparameters of the independent gamma prior. The data set must contain the _TYPE_ variable to identify

the observation type, and it must contain the variables named Lambda1, Lambda2, ..., and so forth, to represent the hazard parameters. The observation with _TYPE_='SHAPE' identifies the shape parameters, and the observation with _TYPE_='ISCALE' identifies the inverse-scale parameters.

RELSHAPE<=*c*>
specifies independent $G(c\hat{\lambda}_j, c)$ distribution, where $\hat{\lambda}_j$'s are the MLEs of the hazard rates. This prior has mean $\hat{\lambda}_j$ and variance $\frac{\hat{\lambda}_j}{c}$. By default, c=10^{-4}.

SHAPE=*a* and SCALE=*b*
together specify the Gamma(a, b) prior.

SHAPE=*c*

ISCALE=*c*
specifies the $G(c, c)$ prior.

ARGAMMA <(*argamma-option*)>
specifies an autoregressive gamma prior of order 1, which can be followed by one of the following *argamma-options*. See the section "AR1 Prior" on page 4624 for details.

INPUT=*SAS-data-set*
specifies a data set containing the hyperparameters of the correlated gamma prior. The data set must contain the _TYPE_ variable to identify the observation type, and it must contain the variables named Lambda1, Lambda2, ..., and so forth, to represent the hazard parameters. The observation with _TYPE_='SHAPE' identifies the shape parameters, and the observation with _TYPE_='ISCALE' identifies the *relative* inverse-scale parameters; that is, if a_j and b_j are, respectively, the SHAPE and ISCALE values for $\lambda_j, 1 \le j \le J$, then $\lambda_1 \sim G(a_1, b_1)$, and $\lambda_j \sim G(a_j, b_j/\lambda_{j-1})$ for $2 \le j \le J$.

SHAPE=*a* and SCALE=*b*
together specify that $\lambda_1 \sim G(a, b)$ and $\lambda_j \sim G(a, b/\lambda_{j-1})$ for $2 \le j \le J$.

SHAPE=*c*

ISCALE=*c*
specifies that $\lambda_1 \sim G(c, c)$ and $\lambda_j \sim G(c, c/\lambda_{j-1})$ for $2 \le j \le J$.

To specify the prior for $\alpha_1, \ldots, \alpha_J$, the hazard parameters in the log scale, you specifying the following:

PRIOR=UNIFORM | NORMAL<(*normal-option*)>
The default is PRIOR=UNIFORM. The available prior options are as follows:

UNIFORM
specifies the uniform prior on the real line; that is, $\alpha_i \propto 1$ for all $-\infty < \alpha_i < \infty$.

NORMAL<(*normal-option*)>
specifies a normal prior distribution on the log-hazard parameters. The *normal options* include the following. If you do not specify an option, the normal prior $N(\mathbf{0}, 10^6\mathbf{I})$, where $\mathbf{I}$ is the identity matrix, is used.

INPUT=*SAS-data-set*
specifies a SAS data set containing the mean and covariance information of the normal prior. The data set must contain the _TYPE_ variable to identify the observation type, and it must contain variables named Alpha1, Alpha2, ..., and so forth, to represent the log-hazard parameters. If the data set also contains the _NAME_ variable, the value of this variable will be used to identify the covariances for the _TYPE_='COV' observations; otherwise, the _TYPE_='COV' observations are assumed to be in the same order as the explanatory variables in the MODEL statement. PROC PHREG reads the mean vector from the observation with _TYPE_='MEAN' and the covariance matrix from observations with _TYPE_='COV'. See the section "Normal Prior" on page 4625 for details. For an independent normal prior, the variances can be specified with _TYPE_='VAR'; alternatively, the precisions (inverse of the variances) can be specified with _TYPE_='PRECISION'.

If you have a joint normal prior for the log-hazard parameters and the regression coefficients, specify the same data set containing the mean and covariance information of the multivariate normal distribution in both the COEFFPRIOR=NORMAL(INPUT=) and the PIECEWISE=LOGHAZARD(PRIOR=NORMAL(INPUT=)) options. See the section "Joint Multivariate Normal Prior for Log-Hazards and Regression Coefficients" on page 4625 for details.

RELVAR<=*c*>
specifies the normal prior $N(\mathbf{0}, c\mathbf{J})$, where $\mathbf{J}$ is a diagonal matrix with diagonal elements equal to the variances of the corresponding ML estimator. By default, c=10^6.

VAR=*c*
specifies the normal prior $N(\mathbf{0}, c\mathbf{I})$, where $\mathbf{I}$ is the identity matrix.

PLOTS<(*global-plot-options*)>= *plot-request*

PLOTS<(*global-plot-options*)>=(*plot-requests*)
controls the diagnostic plots produced through ODS Graphics. Three types of plots can be requested: trace plots, autocorrelation function plots, and kernel density plots. By default, the plots are displayed in panels unless the global plot option UNPACK is specified. If you specify more than one type of plots, the plots are displayed by parameters unless the global plot option GROUPBY=TYPE is specified. When you specify only one plot request, you can omit the parentheses around the plot request. For example:

```
plots=none
plots(unpack)=trace
plots=(trace autocorr)
```

You must enable ODS Graphics before requesting plots, for example, like this:

```
ods graphics on;

proc phreg;
   model y=x;
   bayes plots=trace;
   run;
end;

ods graphics off;
```

If you have enabled ODS Graphics but do not specify the PLOTS= option in the BAYES statement, then PROC PHREG produces, for each parameter, a panel containing the trace plot, the autocorrelation function plot, and the density plot. This is equivalent to specifying plots=(trace autocorr density).

The global plot options include the following:

FRINGE
creates a fringe plot on the X axis of the density plot.

GROUPBY = PARAMETER | TYPE
specifies how the plots are to be grouped when there is more than one type of plots. The choices are as follows:

TYPE
specifies that the plots be grouped by type.

PARAMETER
specifies that the plots be grouped by parameter.

GROUPBY=PARAMETER is the default.

SMOOTH
displays a fitted penalized B-spline curve each trace plot.

UNPACKPANEL
UNPACK
specifies that all paneled plots be unpacked, meaning that each plot in a panel is displayed separately.

The plot requests include the following:

ALL
specifies all types of plots. PLOTS=ALL is equivalent to specifying PLOTS=(TRACE AUTOCORR DENSITY).

AUTOCORR
displays the autocorrelation function plots for the parameters.

DENSITY
displays the kernel density plots for the parameters.

NONE
suppresses all diagnostic plots.

TRACE
displays the trace plots for the parameters. See the section "Visual Analysis via Trace Plots" on page 156 for details.

Consider a model with four parameters, X1–X4. Displays for various specification are depicted as follows.

1. PLOTS=(TRACE AUTOCORR) displays the trace and autocorrelation plots for each parameter side by side with two parameters per panel:

Display 1	Trace(X1)	Autocorr(X1)
	Trace(X2)	Autocorr(X2)
Display 2	Trace(X3)	Autocorr(X3)
	Trace(X4)	Autocorr(X4)

2. PLOTS(GROUPBY=TYPE)=(TRACE AUTOCORR) displays all the paneled trace plots, followed by panels of autocorrelation plots:

Display 1	Trace(X1)	
	Trace(X2)	
Display 2	Trace(X3)	
	Trace(X4)	
Display 3	Autocorr(X1)	Autocorr(X2)
	Autocorr(X3)	Autocorr(X4)

3. PLOTS(UNPACK)=(TRACE AUTOCORR) displays a separate trace plot and a separate correlation plot, parameter by parameter:

Display 1	Trace(X1)
Display 2	Autocorr(X1)
Display 3	Trace(X2)
Display 4	Autocorr(X2)
Display 5	Trace(X3)
Display 6	Autocorr(X3)
Display 7	Trace(X4)
Display 8	Autocorr(X4)

4. PLOTS(UNPACK GROUPBY=TYPE) = (TRACE AUTOCORR) displays all the separate trace plots followed by the separate autocorrelation plots:

Display 1	Trace(X1)
Display 2	Trace(X2)
Display 3	Trace(X3)
Display 4	Trace(X4)
Display 5	Autocorr(X1)
Display 6	Autocorr(X2)
Display 7	Autocorr(X3)
Display 8	Autocorr(X4)

SEED=*number*

specifies an integer seed ranging from 1 to $2^{31}-1$ for the random number generator in the simulation. Specifying a seed enables you to reproduce identical Markov chains for the same specification. If the SEED= option is not specified, or if you specify a nonpositive seed, a random seed is derived from the time of day.

STATISTICS < **(***global-options***)** > **= ALL** | **NONE** | *keyword* | *(keyword-list)*

STATS < **(***global-statoptions***)** > **= ALL** | **NONE** | *keyword* | *(keyword-list)*

controls the number of posterior statistics produced. Specifying STATISTICS=ALL is equivalent to specifying STATISTICS=(SUMMARY INTERVAL COV CORR). If you do not want any posterior statistics, you specify STATISTICS=NONE. The default is STATISTICS=(SUMMARY INTERVAL). See the section "Summary Statistics" on page 170 for details. The *global-options* include the following:

ALPHA=*numeric-list*

controls the probabilities of the credible intervals. The ALPHA= values must be between 0 and 1. Each ALPHA= value produces a pair of 100(1–ALPHA)% equal-tail and HPD intervals for each parameters. The default is the value of the ALPHA= option in the PROC PHREG statement, or 0.05 if that option is not specified (yielding the 95% credible intervals for each parameter).

PERCENT=*numeric-list*

requests the percentile points of the posterior samples. The PERCENT= values must be between 0 and 100. The default is PERCENT= 25, 50, 75, which yield the 25th, 50th, and 75th percentile points for each parameter.

The list of *keywords* includes the following:

CORR

produces the posterior correlation matrix.

COV

produces the posterior covariance matrix.

SUMMARY

produces the means, standard deviations, and percentile points for the posterior samples. The default is to produce the 25th, 50th, and 75th percentile points, but you can use the global PERCENT= option to request specific percentile points.

INTERVAL

produces equal-tail credible intervals and HPD intervals. The defult is to produce the 95% equal-tail credible intervals and 95% HPD intervals, but you can use the global ALPHA= option to request intervals of any probabilities.

THINNING=*number*

THIN=*number*

controls the thinning of the Markov chain. Only one in every k samples is used when THINNING=k, and if NBI=n_0 and NMC=n, the number of samples kept is

$$\left[\frac{n_0+n}{k}\right]-\left[\frac{n_0}{k}\right]$$

where $[a]$ represents the integer part of the number a. The default is THINNING=1.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC PHREG to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the PHREG procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variable* < *(options)* > < *variable* < *(options)* > . . . > < */ options* > **;**

The CLASS statement names the categorical variables to be used in the analysis. The CLASS statement must precede the MODEL statement. You can specify various *options* for each variable by enclosing them in parentheses after the variable name. You can also specify global *options* for the CLASS statement by placing them after a slash (/). Global *options* are applied to all the variables specified in the CLASS statement. If you specify more than one CLASS statement, the global *options* specified in any one CLASS statement apply to all CLASS statements. However, individual CLASS variable *options* override the global *options*.

CPREFIX= *n*

specifies that, at most, the first *n* characters of a CLASS variable name be used in creating names for the corresponding dummy variables. The default is $32 - \min(32, \max(2, f))$, where f is the formatted length of the CLASS variable.

DESCENDING

DESC

reverses the sorting order of the categorical variable.

LPREFIX= *n*

specifies that, at most, the first *n* characters of a CLASS variable label be used in creating labels for the corresponding dummy variables.

MISSING

allows a missing value (for example,'.' for a numeric variable and blanks for a character variable) as a valid value for the CLASS variable.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the categories of cateogrical variables. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option can be useful when you use the CONTRAST statement. When the default ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values. The following table shows how PROC PHREG interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent. For more information about sorting order, see the chapter on the SORT

procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

PARAM=*keyword*

specifies the parameterization method for the categorical variable or variables. Design matrix columns are created from CLASS variables according to the following coding schemes. The default is PARAM=REF. If PARAM=ORTHPOLY or PARAM=POLY, and the CLASS levels are numeric, then the ORDER= option in the CLASS statement is ignored, and the internal, unformatted values are used. See the section "CLASS Variable Parameterization" on page 4580 for further details.

EFFECT	specifies effect coding.
GLM	specifies less-than-full-rank, reference-cell coding; this option can be used only as a global option.
ORDINAL	specifies the cumulative parameterization for an ordinal CLASS variable.
POLYNOMIAL POLY	specifies polynomial coding.
REFERENCE REF	specifies reference cell coding.
ORTHEFFECT	orthogonalizes PARAM=EFFECT.
ORTHORDINAL	orthogonalizes PARAM=ORDINAL.
ORTHPOLY	orthogonalizes PARAM=POLYNOMIAL.
ORTHREF	orthogonalizes PARAM=REFERENCE.

The EFFECT, POLYNOMIAL, REFERENCE, and ORDINAL coding schemes and their orthogonal parameterizations are full-rank parameterization. The REF= option in the CLASS statement determines the reference level for the EFFECT, REFERENCE, and their orthogonal parameterizations.

Parameter names for a CLASS predictor variable are constructed by concatenating the CLASS variable name with the CLASS levels. However, for the POLYNOMIAL and orthogonal parameterizations, parameter names are formed by concatenating the CLASS variable name and keywords that reflect the parameterization.

REF=*'level'* | *keyword*

specifies the reference level for PARAM=EFFECT or PARAM=REF. For an individual variable, you can specify a specific *level* of the variable in the REF= option. For a global or individual variable REF= *option*, you can use one of the following *keywords*. The default is REF=LAST.

FIRST	designates the first ordered level as reference.
LAST	designates the last ordered level as reference.

TRUNCATE

specifies that class levels should be determined using no more than the first 16 characters of the formatted values of CLASS variables. This is a global option, not an individual CLASS variable option.

CONTRAST Statement

CONTRAST *'label' row-description* <*,... row-description*><*/options*> **;**

where a *row-description* is: *effect values* <*,... effect values*>

The CONTRAST statement provides a mechanism for obtaining customized hypothesis tests. It is similar to the CONTRAST statement in PROC GLM and PROC CATMOD, depending on the coding schemes used with any categorical variables involved.

The CONTRAST statement enables you to specify a matrix, $\mathbf{L}$, for testing the hypothesis $\mathbf{L}\boldsymbol{\beta} = \mathbf{0}$. You must be familiar with the details of the model parameterization that PROC PHREG uses (for more information, see the PARAM= option in the section "CLASS Statement" on page 4557). Optionally, the CONTRAST statement enables you to estimate each row, $\boldsymbol{l}_i'\boldsymbol{\beta}$, of $\mathbf{L}\boldsymbol{\beta}$ and test the hypothesis $\boldsymbol{l}_i'\boldsymbol{\beta} = 0$. Computed statistics are based on the asymptotic chi-square distribution of the Wald statistic.

There is no limit to the number of CONTRAST statements that you can specify, but they must appear after the MODEL statement.

The following parameters are specified in the CONTRAST statement:

label identifies the contrast on the output. A label is required for every contrast specified, and it must be enclosed in quotes.

effect identifies an effect that appears in the MODEL statement. You do not need to include all effects that are included in the MODEL statement.

values are constants that are elements of the $\mathbf{L}$ matrix associated with the effect. To correctly specify your contrast, it is crucial to know the ordering of parameters within each effect and the variable levels associated with any parameter. The "Class Level Information" table shows the ordering of levels within variables. The E option, described later in this section, enables you to verify the proper correspondence of *values* to parameters.

The rows of $\mathbf{L}$ are specified in order and are separated by commas. Multiple degree-of-freedom hypotheses can be tested by specifying multiple *row-descriptions*. For any of the full-rank parameterizations, if an effect is not specified in the CONTRAST statement, all of its coefficients in the $\mathbf{L}$ matrix are set to 0. If too many values are specified for an effect, the extra ones are ignored. If too few values are specified, the remaining ones are set to 0.

When you use effect coding (by specifying PARAM=EFFECT in the CLASS statement), all parameters are directly estimable (involve no other parameters). For example, suppose an effect coded CLASS variable A has four levels. Then there are three parameters $(\alpha_1, \alpha_2, \alpha_3)$ representing the

first three levels, and the fourth parameter is represented by

$$-\alpha_1 - \alpha_2 - \alpha_3$$

To test the first versus the fourth level of A, you would test

$$\alpha_1 = -\alpha_1 - \alpha_2 - \alpha_3$$

or, equivalently,

$$2\alpha_1 + \alpha_2 + \alpha_3 = 0$$

which, in the form $\mathbf{L}\boldsymbol{\beta} = 0$, is

$$\begin{bmatrix} 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = 0$$

Therefore, you would use the following CONTRAST statement:

```
contrast '1 vs. 4' A 2 1 1;
```

To contrast the third level with the average of the first two levels, you would test

$$\frac{\alpha_1 + \alpha_2}{2} = \alpha_3$$

or, equivalently,

$$\alpha_1 + \alpha_2 - 2\alpha_3 = 0$$

Therefore, you would use the following CONTRAST statement:

```
contrast '1&2 vs. 3' A 1 1 -2;
```

Other CONTRAST statements involving classification variables with PARAM=EFFECT are constructed similarly. For example:

```
contrast '1 vs. 2    '  A  1 -1  0;
contrast '1&2 vs. 4  '  A  3  3  2;
contrast '1&2 vs. 3&4'  A  2  2  0;
contrast 'Main Effect'  A  1  0  0,
                        A  0  1  0,
                        A  0  0  1;
```

When you use the less-than-full-rank parameterization (by specifying PARAM=GLM in the CLASS statement), each row is checked for estimability. If PROC PHREG finds a contrast to be nonestimable, it displays missing values in corresponding rows in the results. PROC PHREG handles missing level combinations of categorical variables in the same manner as PROC GLM. Parameters corresponding to missing level combinations are not included in the model. This convention can affect the way in which you specify the **L** matrix in your CONTRAST statement. If the elements of **L** are not specified for an effect that contains a specified effect, then the elements of the specified effect are distributed over the levels of the higher-order effect just as the GLM procedure does for its CONTRAST and ESTIMATE statements. For example, suppose that the model contains effects A and B and their interaction A*B. If you specify a CONTRAST statement involving A alone, the **L** matrix contains nonzero terms for both A and A*B, since A*B contains A.

The Cox model contains no explicit intercept parameter, so it is not valid to specify one in the CONTRAST statement. As a consequence, you can test or estimate only homogeneous linear combinations (those with zero-intercept coefficients, such as contrasts that represent group differences) for the GLM parameterization.

The degrees of freedom are the number of linearly independent constraints implied by the CONTRAST statement—that is, the rank of $\mathbf{L}$.

You can specify the following options after a slash (/).

ALPHA= *p*

specifies the level of significance p for the $100(1-p)\%$ confidence interval for each contrast when the ESTIMATE option is specified. The value p must be between 0 and 1. By default, p is equal to the value of the ALPHA= option in the PROC PHREG statement, or 0.05 if that option is not specified.

E

requests that the $\mathbf{L}$ matrix be displayed.

ESTIMATE=*keyword*

requests that each individual contrast (that is, each row, $l_i'\boldsymbol{\beta}$, of $\mathbf{L}\boldsymbol{\beta}$) or exponentiated contrast ($e^{l_i'\boldsymbol{\beta}}$) be estimated and tested. PROC PHREG displays the point estimate, its standard error, a Wald confidence interval, and a Wald chi-square test for each contrast. The significance level of the confidence interval is controlled by the ALPHA= option. You can estimate the contrast or the exponentiated contrast ($e^{l_i'\boldsymbol{\beta}}$), or both, by specifying one of the following *keywords*:

PARM	specifies that the contrast itself be estimated.
EXP	specifies that the exponentiated contrast be estimated.
BOTH	specifies that both the contrast and the exponentiated contrast be estimated.

SINGULAR=*number*

tunes the estimability check. This option is ignored when the full-rank parameterization is used. If $\boldsymbol{v}$ is a vector, define ABS($\boldsymbol{v}$) to be the largest absolute value of the elements of $\boldsymbol{v}$. For a row vector $\boldsymbol{l}'$ of the contrast matrix $\mathbf{L}$, define c to be equal to ABS($\boldsymbol{l}$) if ABS($\boldsymbol{l}$) is greater than 0; otherwise, c equals 1. If ABS($\boldsymbol{l}' - \boldsymbol{l}'\mathbf{T}$) is greater than $c * number$, then $\boldsymbol{l}$ is declared nonestimable. The $\mathbf{T}$ matrix is the Hermite form matrix $\mathbf{I}_0^-\mathbf{I}_0$, where $\mathbf{I}_0^-$ represents a generalized inverse of the information matrix $\mathbf{I}_0$ of the null model. The value for $number$ must be between 0 and 1; the default value is 1E−4.

FREQ Statement

FREQ *variable* < */option* > **;**

The *variable* in the FREQ statement identifies the variable (in the input data set) containing the frequency of occurrence of each observation. PROC PHREG treats each observation as if it appears n times, where n is the value of the FREQ variable for the observation. If not an integer, the

frequency value is truncated to an integer. If the frequency value is missing, the observation is not used in the estimation of the regression parameters.

The following option can be specified in the FREQ statement after a slash (/):

NOTRUNCATE

NOTRUNC

specifies that frequency values are not truncated to integers.

HAZARDRATIO Statement

HAZARDRATIO < *'label'* > *variable* < / *options* > **;**

The HAZARDRATIO statement enables you to request hazard ratios for any variable in the model at customized settings. For example, if the model contains the interaction of a CLASS variable A and a continuous variable X, the following specification displays a table of hazard ratios comparing the hazards of each pair of levels of A at X=3:

```
hazardratio A / at (X=3) diff=ALL;
```

The HAZARDRATIO statement identifies the variable whose hazard ratios are to be evaluated. If the variable is a continuous variable, the hazard ratio compares the hazards for a given change (by default, a increase of 1 unit) in the variable. For a CLASS variable, a hazard ratio compares the hazards of two levels of the variable. More than one HAZARDRATIO statement can be specified, and an optional label (specified as a quoted string) helps identify the output.

Options for the HAZARDRATIO statement are as follows.

ALPHA=*number*

specifies the alpha level of the interval estimates for the hazard ratios. The value must be between 0 and 1. The default is the value of the ALPHA= option in the PROC PHREG statement, or 0.05 if that option is not specified.

AT (*variable***=ALL | REF |** *list* **< . . .** *variable***=ALL | REF|** *list* **>)**

specifies the variables that interact with the variable of interest and the corresponding values of the interacting variables. If the interacting variable is continuous and a numeric list is specified after the equal sign, hazard ratios are computed for each value in the list. If the interacting variable is a CLASS variable, you can specify, after the equal sign, a list of quoted strings corresponding to various levels of the CLASS variable, or you can specify the keyword ALL or REF. Hazard ratios are computed at each value of the list if the list is specified, or at each level of the interacting variable if ALL is specified, or at the reference level of the interacting variable if REF is specified.

If this option is not specified, PROC PHREG finds all the variables that interact with the variable of interest. If an interacting *variable* is a CLASS variable, *variable*= ALL is the default; if the interacting *variable* is continuous, *variable*=*m* is the default, where *m* is the average of all the sampled values of the continuous *variable*.

Suppose the model contains two interactions: an interaction A*B of CLASS variables A and B, and another interaction A*X of A with a continuous variable X. If 3.5 is the average of the sampled values of X, the following two HAZARDRATIO statements are equivalent:

```
hazardratio A;
hazardratio A / at (B=ALL X=3.5);
```

CL=WALD | PL |BOTH

specifies whether to create the Wald or profile-likelihood confidence limits, or both for the classical analyis. By default, Wald confidence limits are produced. This option is not applicable to a Bayesian analysis.

DIFF=ALL | REF

specifies which differences to consider for the level comparisons of a CLASS variable. The default is DIFF=ALL. This option is ignored in the estimation of hazard ratios for a continuous variable. DIFF=ALL requests all differences, and DIFF=REF requests comparisons between the reference level and all other levels of the CLASS variable.

E

displays the vector $\mathbf{h}$ of linear coefficients such that $\mathbf{h}'\boldsymbol{\beta}$ is the log-hazard ratio, with $\boldsymbol{\beta}$ being the vector of regression coefficients.

PLCONV=*value*

controls the convergence criterion for the profile-likelihood confidence limits. The quantity *value* must be a positive number, with a default value of 1E–4. The PLCONV= option has no effect if profile-likelihood confidence intervals (CL=PL) are not requested.

PLMAXIT=*n*

specifies the maximum number of iterations to achieve the convergence of the profile-likelihood confidence limits. By default, PLMAXITER=25. If convergence is not attained in *n* iterations, the corresponding profile-likelihood confidence limit for the hazard ratio is set to missing. The PLMAXITER= option has no effect if profile-likelihood confidence intervals (CL=PL) are not requested.

PLSINGULAR=*value*

specifies the tolerance for testing the singularity of the Hessian matrix in the computation of the profile-likelihood confidence limits. The test requires that a pivot for sweeping this matrix be at least this number times a norm of the matrix. Values of the PLSINGULAR= option must be numeric. By default, *value* is the machine epsilon times 1E7, which is approximately 1E–9. The PLSINGULAR= option has no effect if profile-likelihood confidence intervals (CL=PL) are not requested.

UNITS=*value*

specifies the units of change in the continuous explanatory variable for which the customized hazard ratio is estimated. The default is UNITS=1. This option is ignored in the computation of the hazard ratios for a CLASS variable.

ID Statement

ID *variables* **;**

The ID statement specifies additional variables for identifying observations in the input data. These variables are placed in the OUT= data set created by the OUTPUT statement. In the computation of the robust sandwich variance estimate, you can aggregate over distinct values of these ID variables.

Only variables in the input data set can be included in the ID statement.

MODEL Statement

MODEL *response* < **censor (list)* > = *effects* < */options* > **;**

MODEL *(t1, t2)* < **censor(list)* > = *effects* < */options* > **;**

The MODEL statement identifies the variables to be used as the failure time variables, the optional censoring variable, and the explanatory effects, including covariates, main effects, interactions, nested effects; see the section "Specification of Effects" on page 2488 of Chapter 39, "The GLM Procedure," for more information. A note of caution: specifying the effect `T*A` in the MODEL statement, where `T` is the time variable and `A` is a CLASS variable, does not make the effect time-dependent. See the section "Clarification of the Time and CLASS Variables Usage" on page 4582 for more information.

Two forms of MODEL syntax can be specified; the first form allows one response variable, while the second form allows two variables for the counting process style of input (see the section "Counting Process Style of Input" on page 4588 for more information).

In the first MODEL statement, preceding the equal sign, is the name of the failure time variable. This can optionally be followed by an asterisk, the name of the censoring variable, and a list of censoring values (separated by blanks or commas if there is more than one) enclosed in parentheses. If the censoring variable takes on one of these values, the corresponding failure time is considered to be censored. The variables following the equal sign are the explanatory variables (sometimes called independent variables or covariates) for the model.

Instead of a single failure time variable, the second MODEL statement identifies a pair of failure time variables. Their names are enclosed in parentheses, and they signify the endpoints of a semi-closed interval $(t1, t2]$ during which the subject is at risk. If the censoring variable takes on one of the censoring values, the time $t2$ is considered to be censored.

The censoring variable and the explanatory variables must be numeric. The failure time variables must contain nonnegative values. Any observation with a negative failure time is excluded from the analysis, as is any observation with a missing value for any of the variables listed in the MODEL statement.

Table 64.3 summarizes the options available in the MODEL statement, which can be specified after a slash (/). Four convergence criteria are allowed for the maximum likelihood optimization:

ABSFCONV=, FCONV=, GCONV=, and XCONV=. If you specify more than one convergence criterion, the optimization is terminated as soon as one of the criteria is satisfied. If none of the criteria is specified, the default is GCONV=1E−8.

Table 64.3 MODEL Statement Options

Option	Description
Model Specification Options	
NOFIT	suppresses model fitting
OFFSET=	specifies offset variable
SELECTION=	specifies effect selection method
Effect Selection Options	
BEST=	controls the number of models displayed for SCORE selection
DETAILS	requests detailed results at each step
HIERARCHY=	specifies whether and how hierarchy is maintained and whether a single effect or multiple effects are allowed to enter or leave the model per step
INCLUDE=	specifies number of effects included in every model
MAXSTEP=	specifies maximum number of steps for STEPWISE selection
SEQUENTIAL	adds or deletes effects in sequential order
SLENTRY=	specifies significance level for entering effects
SLSTAY=	specifies significance level for removing effects
START=	specifies number of variables in first model
STOP=	specifies number of variables in final model
STOPRES	adds or deletes variables by residual chi-square criterion
Maximum Likelihood Optimization Options	
ABSFCONV=	specifies absolute function convergence criterion
FCONV=	specifies relative function convergence criterion
FIRTH	specifies Firth's penalized likelihood method
GCONV=	specifies relative gradient convergence criterion
XCONV=	specifies relative parameter convergence criterion
MAXITER=	specifies maximum number of iterations
RIDGEINIT=	specifies the initial ridging value
RIDGING=	specifies the technique to improve the log likelihood function when its value is worse than that of the previous step
SINGULAR=	specifies tolerance for testing singularity
Confidence Interval Options	
ALPHA=	specifies α for the $100(1-\alpha)\%$ confidence intervals
PLCONV=	specifies profile-likelihood convergence criterion
RISKLIMITS=	computes confidence intervals for hazard ratios
Display Options	
CORRB	displays correlation matrix
COVB	displays covariance matrix
ITPRINT	displays iteration history

Table 64.3 *continued*

Option	Description
NODUMMYPRINT	suppresses "Class Level Information" table
Miscellaneous Options	
ENTRYTIME=	specifies the delayed entry time variable
TIES=	specifies the method of handling ties in failure times

ALPHA=*value*

sets the significance level used for the confidence limits for the hazard ratios. The value must be between 0 and 1. The default is the value of the ALPHA= option in the PROC PHREG statement, or 0.05 if that option is not specified. This option has no effect unless the RISKLIMITS option is specified.

ABSFCONV=*value*

CONVERGELIKE=*value*

specifies the absolute function convergence criterion. Termination requires a small change in the objective function (log partial likelihood function) in subsequent iterations,

$$|l_k - l_{k-1}| < value$$

where l_k is the value of the objective function at iteration k.

BEST=*n*

is used exclusively with the SCORE model selection method. The BEST=*n* option specifies that *n* models with the highest-score chi-square statistics are to be displayed for each model size. If the option is omitted and there are no more than 10 explanatory variables, then all possible models are listed for each model size. If the option is omitted and there are more than 10 explanatory variables, then the number of models selected for each model size is, at most, equal to the number of explanatory variables listed in the MODEL statement.

See Example 64.2 for an illustration of the SCORE selection method and the BEST= option.

CORRB

displays the estimated correlation matrix of the parameter estimates.

COVB

displays the estimated covariance matrix of the parameter estimates.

DETAILS

produces a detailed display at each step of the model-building process. It produces an "Analysis of Variables Not in the Model" table before displaying the variable selected for entry for FORWARD or STEPWISE selection. For each model fitted, it produces the "Analysis of Maximum Likelihood Estimates" table.

See Example 64.1 for a discussion of these tables.

ENTRYTIME=*variable*

ENTRY=*variable*

specifies the name of the variable that represents the left truncation time. This option has no effect when the counting process style of input is specified. See the section "Left Truncation of Failure Times" on page 4589 for more information.

FCONV=*value*

specifies the relative function convergence criterion. Termination requires a small relative change in the objective function (log partial likelihood function) in subsequent iterations,

$$\frac{|l_k - l_{k-1}|}{|l_{k-1}| + 1\mathrm{E}-6} < value$$

where l_k is the value of the objective function at iteration k.

FIRTH

performs Firth's penalized maximum likelihood estimation to reduce bias in the parameter estimates (Heinze and Schemper 2001; Firth 1993). This method is useful when the likelihood is monotone—that is, the likelihood converges to finite value while at least one estimate diverges to infinity.

GCONV=*value*

specifies the relative gradient convergence criterion. Termination requires that the normalized prediction function reduction is small,

$$\frac{\mathbf{g}_k \mathbf{H}_k^{-1} \mathbf{g}_k}{|l_k| + 1\mathrm{E}-6} < value$$

where l_k is the log partial likelihood, $\mathbf{g}_k$ is the gradient vector (first partial derivatives of the log partial likelihood), and $\mathbf{H}_k$ is the negative Hessian matrix (second partial derivatives of the log partial likelihood), all at iteration k.

HIERARCHY=*keyword*

HIER=*keyword*

specifies whether and how the model hierarchy requirement is applied and whether a single effect or multiple effects are allowed to enter or leave the model in one step. You can specify that only CLASS variable effects, or both CLASS and continuous variable effects, be subject to the hierarchy requirement. The HIERARCHY= option is ignored unless you also specify the FORWARD, BACKWARD, or STEPWISE selection method.

Model hierarchy refers to the requirement that, for any term to be in the model, all effects contained in the term must be present in the model. For example, in order for the interaction A*B to enter the model, the main effects A and B must be in the model. Likewise, neither effect A nor B can leave the model while the interaction A*B is in the model.

The keywords you can specify in the HIERARCHY= option are described as follows:

NONE

indicates that the model hierarchy is not maintained. Any single effect can enter or leave the model at any given step of the selection process.

SINGLE
indicates that only one effect can enter or leave the model at one time, subject to the model hierarchy requirement. For example, suppose that you specify the main effects A and B and the interaction of A*B in the model. In the first step of the selection process, either A or B can enter the model. In the second step, the other main effect can enter the model. The interaction effect can enter the model only when both main effects have already been entered. Also, before A or B can be removed from the model, the A*B interaction must first be removed. All effects (CLASS and continuous variables) are subject to the hierarchy requirement.

SINGLECLASS
is the same as HIERARCHY=SINGLE except that only CLASS effects are subject to the hierarchy requirement.

MULTIPLE
indicates that more than one effect can enter or leave the model at one time, subject to the model hierarchy requirement. In a forward selection step, a single main effect can enter the model, or an interaction can enter the model together with all the effects that are contained in the interaction. In a backward elimination step, an interaction itself, or the interaction together with all the effects that the interaction contains, can be removed. All effects (CLASS and continuous variable) are subject to the hierarchy requirement.

MULTIPLECLASS
is the same as HIERARCHY=MULTIPLE except that only CLASS effects are subject to the hierarchy requirement.

The default value is HIERARCHY=SINGLE, which means that model hierarchy is to be maintained for all effects (that is, both CLASS and continuous variable effects) and that only a single effect can enter or leave the model at each step.

INCLUDE=*n*
includes the first *n* effects in the MODEL statement in every model. By default, INCLUDE=0. The INCLUDE= option has no effect when SELECTION=NONE.

ITPRINT
displays the iteration history, including the last evaluation of the gradient vector.

MAXITER=*n*
specifies the maximum number of iterations allowed. The default value for *n* is 25. If convergence is not attained in *n* iterations, the displayed output and all data sets created by PROC PHREG contain results that are based on the last maximum likelihood iteration.

MAXSTEP=*n*
specifies the maximum number of times the explanatory variables can move in and out of the model before the STEPWISE model-building process ends. The default value for *n* is twice the number of explanatory variables in the MODEL statement. The option has no effect for other model selection methods.

NODUMMYPRINT

NODESIGNPRINT

NODP

suppresses the "Class Level Information" table, which shows how the design matrix columns for the CLASS variables are coded.

NOFIT

performs the global score test, which tests the joint significance of all the explanatory variables in the MODEL statement. No parameters are estimated. If the NOFIT option is specified along with other MODEL statement options, NOFIT takes precedence, and all other options are ignored except the TIES= option.

OFFSET=*name*

specifies the name of an offset variable, which is an explanatory variable with a regression coefficient fixed as one. This option can be used to incorporate risk weights for the likelihood function.

PLCONV=*value*

controls the convergence criterion for confidence intervals based on the profile-likelihood function. The quantity *value* must be a positive number, with a default value of 1E−4. The PLCONV= option has no effect if profile-likelihood based confidence intervals are not requested.

RIDGING=*keyword*

specifies the technique to improve the log likelihood when its value is worse than that of the previous step. The available *keywords* are as follows:

ABSOLUTE

specifies that the diagonal elements of the negative (expected) Hessian be inflated by adding the ridge value.

RELATIVE

specifies that the diagonal elements be inflated by the factor equal to 1 plus the ridge value.

NONE

specifies the crude line-search method of taking half a step be used instead of ridging.

The default is RIDGING=RELATIVE.

RIDGEINIT=*value*

specifies the initial ridge value. The maximum ridge value is 2000 times the maximum of 1 and the initial ridge value. The initial ridge value is raised to 1E–4 if it is less than 1E–4. By default, RIDGEINIT=1E–4. This option has no effect for RIDGING=ABSOLUTE.

RISKLIMITS< =*keyword* **>**

RL< =*keyword* **>**

produces confidence intervals for hazard ratios of main effects not involved in interactions or nestings. Computation of these confidence intervals is based on the profile likelihood or

based on individual Wald tests. The confidence coefficient can be specified with the ALPHA= option. You can specify one of the following keywords:

PL
requests profile-likelihood confidence limits.

WALD
requests confidence limits based on the Wald tests.

BOTH
request both profile-likelihood and Wald confidence limits.

Classification main effects that use parameterizations other than REF, EFFECT, or GLM are ignored. If you need to compute hazard ratios for an effect involved in interactions or nestings, or using some other parameterization, then you should specify a HAZARDRATIO statement for that effect.

SELECTION=*method*
specifies the method used to select the model. The *method*s available are as follows:

BACKWARD
B
requests backward elimination.

FORWARD
F
requests forward selection.

NONE
N
fits the complete model specified in the MODEL statement. This is the default value.

SCORE
requests best subset selection. It identifies a specified number of models with the highest-score chi-square statistic for all possible model sizes ranging from one explanatory variable to the total number of explanatory variables listed in the MODEL statement. This option is not allowed if an explanatory effect in the MODEL statement contains a CLASS variable.

STEPWISE
S
requests stepwise selection.

For more information, see the section "Effect Selection Methods" on page 4617.

SEQUENTIAL
forces variables to be added to the model in the order specified in the MODEL statement or to be eliminated from the model in the reverse order of that specified in the MODEL statement.

SINGULAR=*value*

specifies the singularity criterion for determining linear dependencies in the set of explanatory variables. The default value is 10^{-12}.

SLENTRY=*value*

SLE=*value*

specifies the significance level (a value between 0 and 1) for entering an explanatory variable into the model in the FORWARD or STEPWISE method. For all variables not in the model, the one with the smallest *p*-value is entered if the *p*-value is less than or equal to the specified significance level. The default value is 0.05.

SLSTAY=*value*

SLS=*value*

specifies the significance level (a value between 0 and 1) for removing an explanatory variable from the model in the BACKWARD or STEPWISE method. For all variables in the model, the one with the largest *p*-value is removed if the *p*-value exceeds the specified significance level. The default value is 0.05.

START=*n*

begins the FORWARD, BACKWARD, or STEPWISE selection process with the first *n* effects listed in the MODEL statement. The value of *n* ranges from 0 to *s*, where *s* is the total number of effects in the MODEL statement. The default value of *n* is *s* for the BACKWARD method and 0 for the FORWARD and STEPWISE methods. Note that START=*n* specifies only that the first *n* effects appear in the first model, while INCLUDE=*n* requires that the first *n* effects be included in every model. For the SCORE method, START=*n* specifies that the smallest models contain *n* effects, where *n* ranges from 1 to *s*; the default value is 1. The START= option has no effect when SELECTION=NONE.

STOP=*n*

specifies the maximum (FORWARD method) or minimum (BACKWARD method) number of effects to be included in the final model. The effect selection process is stopped when *n* effects are found. The value of *n* ranges from 0 to *s*, where *s* is the total number of effects in the MODEL statement. The default value of *n* is *s* for the FORWARD method and 0 for the BACKWARD method. For the SCORE method, STOP=*n* specifies that the smallest models contain *n* effects, where *n* ranges from 1 to *s*; the default value of *n* is *s*. The STOP= option has no effect when SELECTION=NONE or STEPWISE.

STOPRES

SR

specifies that the addition and deletion of variables be based on the result of the likelihood score test for testing the joint significance of variables not in the model. This score chi-square statistic is referred to as the residual chi-square. In the FORWARD method, the STOPRES option enters the explanatory variables into the model one at a time until the residual chi-square becomes insignificant (that is, until the *p*-value of the residual chi-square exceeds the SLENTRY= value). In the BACKWARD method, the STOPRES option removes variables from the model one at a time until the residual chi-square becomes significant (that is, until the *p*-value of the residual chi-square becomes less than the SLSTAY= value). The STOPRES option has no effect for the STEPWISE method.

TIES=*method*

specifies how to handle ties in the failure time. The available *methods* are as follows:

BRESLOW

uses the approximate likelihood of Breslow (1974). This is the default value.

DISCRETE

replaces the proportional hazards model by the discrete logistic model

$$\frac{\lambda(t;\mathbf{z})}{1-\lambda(t;\mathbf{z})} = \frac{\lambda_0(t)}{1-\lambda_0(t)}\exp(\mathbf{z}'\boldsymbol{\beta})$$

where $\lambda_0(t)$ and $h(t;\mathbf{z})$ are discrete hazard functions.

EFRON

uses the approximate likelihood of Efron (1977).

EXACT

computes the exact conditional probability under the proportional hazards assumption that all tied event times occur before censored times of the same value or before larger values. This is equivalent to summing all terms of the marginal likelihood for $\boldsymbol{\beta}$ that are consistent with the observed data (Kalbfleisch and Prentice 1980; DeLong, Guirguis, and So 1994).

The EXACT method can take a considerable amount of computer resources. If ties are not extensive, the EFRON and BRESLOW methods provide satisfactory approximations to the EXACT method for the continuous time-scale model. In general, Efron's approximation gives results that are much closer to the EXACT method results than Breslow's approximation does. If the time scale is genuinely discrete, you should use the DISCRETE method. The DISCRETE method is also required in the analysis of case-control studies when there is more than one case in a matched set. If there are no ties, all four methods result in the same likelihood and yield identical estimates. The default, TIES=BRESLOW, is the most efficient method when there are no ties.

XCONV=*value*

CONVEREPARM=*value*

specifies the relative parameter convergence criterion. Termination requires a small relative parameter change in subsequent iterations,

$$\max_i |\delta_k^{(i)}| < \mathit{value}$$

where

$$\delta_k^{(i)} = \begin{cases} \theta_k^{(i)} - \theta_{k-1}^{(i)} & |\theta_{k-1}^{(i)}| < .01 \\ \frac{\theta_k^{(i)} - \theta_{k-1}^{(i)}}{\theta_{k-1}^{(i)}} & \text{otherwise} \end{cases}$$

where $\theta_k^{(i)}$ is the estimate of the ith parameter at iteration k.

OUTPUT Statement

OUTPUT < *OUT=SAS-data-set* > < *keyword=name . . . keyword=name* > < */options* > **;**

The OUTPUT statement creates a new SAS data set containing statistics calculated for each observation. These can include the estimated linear predictor ($\mathbf{z}_j'\hat{\boldsymbol{\beta}}$) and its standard error, survival distribution estimates, residuals, and influence statistics. In addition, this data set includes the time variable, the explanatory variables listed in the MODEL statement, the censoring variable (if specified), and the BY, STRATA, FREQ, and ID variables (if specified).

For observations with missing values in the time variable or any explanatory variables, the output statistics are set to missing. However, for observations with missing values only in the censoring variable or the FREQ variable, survival estimates are still computed. Therefore, by adding observations with missing values in the FREQ variable or the censoring variable, you can compute the survivor function estimates for new observations or for settings of explanatory variables not present in the data without affecting the model fit.

No OUTPUT data set is created if the model contains a time-dependent variable defined by means of programming statements.

The following list explains specifications in the OUTPUT statement.

OUT=*SAS-data-set*
names the output data set. If you omit the OUT= option, the OUTPUT data set is created and given a default name by using the DATA*n* convention. See the section "OUT= Output Data Set in the OUTPUT Statement" on page 4631 for more information.

keyword=name
specifies the statistics included in the OUTPUT data set and names the new variables that contain the statistics. Specify a keyword for each desired statistic (see the following list of keywords), an equal sign, and either a variable or a list of variables to contain the statistic. The keywords that accept a list of variables are DFBETA, RESSCH, RESSCO, and WTRESSCH. For these keywords, you can specify as many names in *name* as the number of explanatory variables specified in the MODEL statement. If you specify k names and k is less than the total number of explanatory variables, only the changes for the first k parameter estimates are output. The keywords and the corresponding statistics are as follows:

ATRISK

NUM_LEFT
specifies the number of subjects at risk at the observation time τ_j (or at the right endpoint of the at-risk interval when a counting process MODEL specification is used).

DFBETA
specifies the approximate changes in the parameter estimates ($\hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{(j)}$) when the jth observation is omitted. These variables are a weighted transform of the score residual variables and are useful in assessing local influence and in computing robust variance estimates.

LD
specifies the approximate likelihood displacement when the observation is left out. This diagnostic can be used to assess the impact of each observation on the overall fit of the model.

LMAX
specifies the relative influence of observations on the overall fit of the model. This diagnostic is useful in assessing the sensitivity of the fit of the model to each observation.

LOGLOGS
specifies the log of the negative log of SURVIVAL.

LOGSURV
specifies the log of SURVIVAL.

RESDEV
specifies the deviance residual $\hat{D}_j$. This is a transform of the martingale residual to achieve a more symmetric distribution.

RESMART
specifies the martingale residual $\hat{M}_j$. The residual at the observation time τ_j can be interpreted as the difference over $[0, \tau_j]$ in the observed number of events minus the expected number of events given by the model.

RESSCH
specifies the Schoenfeld residuals. These residuals are useful in assessing the proportional hazards assumption.

RESSCO
specifies the score residuals. These residuals are a decomposition of the first partial derivative of the log likelihood. They can be used to assess the leverage exerted by each subject in the parameter estimation. They are also useful in constructing robust sandwich variance estimators.

STDXBETA
specifies the standard error of the estimated linear predictor, $\sqrt{\mathbf{z}_j' \hat{\mathbf{V}}(\hat{\boldsymbol{\beta}}) \mathbf{z}_j}$.

SURVIVAL
specifies the survivor function estimate $\hat{S}_j = [\hat{S}_0(\tau_j)]^{\exp(\mathbf{z}_j' \hat{\boldsymbol{\beta}})}$, where τ_j is the observation time.

WTRESSCH
specifies the weighted Schoenfeld residuals. These residuals are useful in investigating the nature of nonproportionality if the proportional hazard assumption does not hold.

XBETA
specifies the estimate of the linear predictor, $\mathbf{z}_j' \hat{\boldsymbol{\beta}}$.

The following options can appear in the OUTPUT statement after a slash (/) as follows:

ORDER=*sort_order*

specifies the order of the observations in the OUTPUT data set. Available values for *sort_order* are as follows:

DATA requests that the output observations be sorted the same as the input data set.

SORTED requests that the output observations be sorted by strata and descending order of the time variable within each stratum.

The default is ORDER=DATA.

METHOD=*method*

specifies the method used to compute the survivor function estimates. The two available methods are as follows:

BRESLOW

CH

EMP

specifies that the empirical cumulative hazard function estimate of the survivor function be computed; that is, the survivor function is estimated by exponentiating the negative empirical cumulative hazard function.

PL

specifies that the product-limit estimate of the survivor function be computed.

The default is METHOD=BRESLOW.

Programming Statements

Programming statements are used to create or modify the values of the explanatory variables in the MODEL statement. They are especially useful in fitting models with time-dependent explanatory variables. Programming statements can also be used to create explanatory variables that are not time dependent. For example, you can create indicator variables from a categorical variable and incorporate them into the model. PROC PHREG programming statements cannot be used to create or modify the values of the response variable, the censoring variable, the frequency variable, or the strata variables.

The following DATA step statements are available in PROC PHREG:

```
ABORT
ARRAY
assignment statements
CALL
DO
iterative DO
DO UNTIL
```

```
DO WHILE
END
GOTO
IF-THEN/ELSE
LINK-RETURN
PUT
SELECT
SUM statement
```

By default, the PUT statement in PROC PHREG writes results to the Output window instead of the Log window. If you want the results of the PUT statements to go to the Log window, add the following statement before the PUT statements:

```
FILE LOG;
```

DATA step functions are also available. Use these programming statements the same way you use them in the DATA step. For detailed information, refer to *SAS Language Reference: Dictionary*.

Consider the following example of using programming statements in PROC PHREG. Suppose blood pressure is measured at multiple times during the course of a study investigating the effect of blood pressure on some survival time. By treating the blood pressure as a time-dependent explanatory variable, you are able to use the value of the most recent blood pressure at each specific point of time in the modeling process rather than using the initial blood pressure or the final blood pressure. The values of the following variables are recorded for each patient, if they are available. Otherwise, the variables contain missing values.

Time	survival time
Censor	censoring indicator (with 0 as the censoring value)
BP0	blood pressure on entry to the study
T1	time 1
BP1	blood pressure at T1
T2	time 2
BP2	blood pressure at T2

The following programming statements create a variable BP. At each time T, the value of BP is the blood pressure reading for that time, if available. Otherwise, it is the last blood pressure reading.

```
proc phreg;
   model Time*Censor(0)=BP;
   BP = BP0;
   if Time>=T1 and T1^=. then BP=BP1;
   if Time>=T2 and T2^=. then BP=BP2;
run;
```

For other illustrations of using programming statements, see the section "Classical Method of Maximum Likelihood" on page 4527 and Example 64.6.

STRATA Statement

STRATA *variable* < *(list)* > < *. . . variable* < *(list)* > > < */option* > **;**

The proportional hazards assumption might not be realistic for all data. If so, it might still be reasonable to perform a stratified analysis. The STRATA statement names the variables that determine the stratification. Strata are formed according to the nonmissing values of the STRATA variables unless the MISSING option is specified. In the STRATA statement, *variable* is a variable with values that are used to determine the strata levels, and *list* is an optional list of values for a numeric variable. Multiple variables can appear in the STRATA statement.

The values for *variable* can be formatted or unformatted. If the variable is a character variable, or if the variable is numeric and no list appears, then the strata are defined by the unique values of the variable. If the variable is numeric and is followed by a list, then the levels for that variable correspond to the intervals defined by the list. The corresponding strata are formed by the combination of levels and unique values. The list can include numeric values separated by commas or blanks, *value* to *value* by *value* range specifications, or combinations of these.

For example, the specification

```
strata age (5, 10 to 40 by 10) sex ;
```

indicates that the levels for age are to be less than 5, 5 to 10, 10 to 20, 20 to 30, 30 to 40, and greater than 40. (Note that observations with exactly the cutpoint value fall into the interval preceding the cutpoint.) Thus, with the sex variable, this STRATA statement specifies 12 strata altogether.

The following option can be specified in the STRATA statement after a slash (/):

MISSING
: allows missing values (‘.’ for numeric variables and blanks for character variables) as valid STRATA variable values. Otherwise, observations with missing STRATA variable values are deleted from the analysis.

TEST Statement

< *label:* **>** **TEST** *equation1* < *,. . . , equationk* > < */options* > **;**

The TEST statement tests linear hypotheses about the regression coefficients. PROC PHREG performs a Wald test for the joint hypothesis specified in a single TEST statement. Each equation specifies a linear hypothesis; multiple equations (rows of the joint hypothesis) are separated by commas. The *label*, which must be a valid SAS name, is used to identify the resulting output and should always be included. You can submit multiple TEST statements.

The form of an equation is as follows:

term < ± *term . . .* > <= < ± *term* < ± *term . . .* > > >

where *term* is a variable or a constant or a constant times a variable. The variable is any explanatory variable in the MODEL statement. When no equal sign appears, the expression is set to 0. The following program illustrates possible uses of the TEST statement:

```
proc phreg;
   model time= A1 A2 A3 A4;
   Test1: test A1, A2;
   Test2: test A1=0,A2=0;
   Test3: test A1=A2=A3;
   Test4: test A1=A2,A2=A3;
run;
```

Note that the first and second TEST statements are equivalent, as are the third and fourth TEST statements.

The following options can be specified in the TEST statement after a slash (/):

AVERAGE

enables you to assess the average effect of the variables in the given TEST statement. An overall estimate of the treatment effect is computed as a weighted average of the treatment coefficients as illustrated in the following statement:

```
TREATMENT: test trt1, trt2, trt3, trt4 / average;
```

Let $\beta_1, \beta_2, \beta_3$, and β_4 be corresponding parameters for trt1, trt2, trt3, and trt4, respectively. Let $\hat{\boldsymbol{\beta}} = (\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3, \hat{\beta}_4)'$ be the estimated coefficient vector and let $\hat{\mathbf{V}}(\hat{\boldsymbol{\beta}})$ be the corresponding variance estimate. Assuming $\beta_1 = \beta_2 = \beta_3 = \beta_4$, let $\bar{\beta}$ be the average treatment effect. The effect is estimated by $\mathbf{c}'\hat{\boldsymbol{\beta}}$, where $\mathbf{c} = [\mathbf{1}_4'\hat{\mathbf{V}}^{-1}(\hat{\boldsymbol{\beta}})\mathbf{1}_4]^{-1}\hat{\mathbf{V}}^{-1}(\hat{\boldsymbol{\beta}})\mathbf{1}_4$ and $\mathbf{1}_4 = (1, 1, 1, 1)'$. A test of the null hypothesis $H_0 : \bar{\beta} = 0$ is also included, which is more sensitive than the multivariate test for testing the null hypothesis $H_0 : \beta 1 = \beta 2 = \beta 3 = \beta 4 = 0$.

E

specifies that the linear coefficients and constants be printed. When the AVERAGE option is specified along with the E option, the optimal weights of the average effect are also printed in the same tables as the coefficients.

PRINT

displays intermediate calculations. This includes $\mathbf{L}\hat{\mathbf{V}}(\hat{\boldsymbol{\beta}})\mathbf{L}'$ bordered by $(\mathbf{L}\hat{\boldsymbol{\beta}} - \mathbf{c})$, and $[\mathbf{L}\hat{\mathbf{V}}(\hat{\boldsymbol{\beta}})\mathbf{L}']^{-1}$ bordered by $[\mathbf{L}\hat{\mathbf{V}}(\hat{\boldsymbol{\beta}})\mathbf{L}']^{-1}(\mathbf{L}\hat{\boldsymbol{\beta}} - \mathbf{c})$, where $\mathbf{L}$ is a matrix of linear coefficients and $\mathbf{c}$ is a vector of constants.

See the section "Testing Linear Hypotheses about Regression Coefficients" on page 4600 for details.

WEIGHT Statement

WEIGHT *variable* < */option* > **;**

The *variable* in the WEIGHT statement identifies the variable in the input data set that contains the case weights. When the WEIGHT statement appears, each observation in the input data set is

weighted by the value of the WEIGHT variable. The WEIGHT values can be nonintegral and are not truncated. Observations with negative, zero, or missing values for the WEIGHT variable are not used in the model fitting. When the WEIGHT statement is not specified, each observation is assigned a weight of 1. The WEIGHT statement is available for TIES=BRESLOW and TIES=EFRON only.

The following option can be specified in the WEIGHT statement after a slash (/):

NORMALIZE

NORM

causes the weights specified by the WEIGHT *variable* to be normalized so that they add up the actual sample size. With this option, the estimated covariance matrix of the parameter estimators is invariant to the scale of the WEIGHT variable.

Details: PHREG Procedure

Failure Time Distribution

Let T be a nonnegative random variable representing the failure time of an individual from a homogeneous population. The survival distribution function (also known as the survivor function) of T is written as

$$S(t) = \Pr(T \geq t)$$

A mathematically equivalent way of specifying the distribution of T is through its hazard function. The hazard function $\lambda(t)$ specifies the instantaneous failure rate at t. If T is a continuous random variable, $\lambda(t)$ is expressed as

$$\lambda(t) = \lim_{\Delta t \to 0^+} \frac{\Pr(t \leq T < t + \Delta t \mid T \geq t)}{\Delta t} = \frac{f(t)}{S(t)}$$

where $f(t)$ is the probability density function of T. If T is discrete with masses at $x_1 < x_2 < \ldots$, then $\lambda(t)$ is given by

$$\lambda(t) = \sum_j \lambda_j \delta(t - x_j)$$

where

$$\delta(u) = \begin{cases} 0 & \text{if } u < 0 \\ 1 & \text{otherwise} \end{cases}$$

$$\lambda_j = \Pr(T = x_j \mid T \geq x_j) = \frac{\Pr(T = x_j)}{S(x_j)}$$

for $j = 1, 2, \ldots$

CLASS Variable Parameterization

Consider a model with one CLASS variable A with four levels, 1, 2, 5, and 7. Details of the possible choices for the PARAM= option follow.

EFFECT

Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of −1. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows:

	Design Matrix		
A	**A1**	**A2**	**A5**
1	1	0	0
2	0	1	0
5	0	0	1
7	−1	−1	−1

Parameter estimates of CLASS main effects, using the effect coding scheme, estimate the difference in the effect of each nonreference level compared to the average effect over all four levels.

GLM

As in PROC GLM, four columns are created to indicate group membership. The design matrix columns for A are as follows:

	Design Matrix			
A	**A1**	**A2**	**A5**	**A7**
1	1	0	0	0
2	0	1	0	0
5	0	0	1	0
7	0	0	0	1

Parameter estimates of CLASS main effects, using the GLM coding scheme, estimate the difference in the effects of each level compared to the last level.

ORDINAL

Three columns are created to indicate group membership of the higher levels of the effect. For the first level of the effect (which for A is 1), all three dummy variables have a value of 0. The design matrix columns for A are as follows:

	Design Matrix		
A	**A2**	**A5**	**A7**
1	0	0	0
2	1	0	0
5	1	1	0
7	1	1	1

In the ORDINAL coding scheme, the first level of the effect is a control or baseline level. Parameter estimates of CLASS main effects estimate the differences between effects of successive levels. When the parameters have the same sign, the effect is monotonic across the levels.

POLYNOMIAL

POLY

Three columns are created. The first represents the linear term (x), the second represents the quadratic term (x^2), and the third represents the cubic term (x^3), where x is the level value. If the CLASS levels are not numeric, they are translated into 1, 2, 3, ... according to their sorting order. The design matrix columns for A are as follows:

	Design Matrix		
A	**APOLY1**	**APOLY2**	**APOLY3**
1	1	1	1
2	2	4	8
5	5	25	125
7	7	49	343

REFERENCE

REF

Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of 0. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows:

	Design Matrix		
A	**A1**	**A2**	**A5**
1	1	0	0
2	0	1	0
5	0	0	1
7	0	0	0

Parameter estimates of CLASS main effects, using the reference coding scheme, estimate the difference in the effect of each nonreference level compared to the effect of the reference level.

ORTHEFFECT

The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=EFFECT. The design matrix columns for A are as follows:

	Design Matrix		
A	**AOEFF1**	**AOEFF2**	**AOEFF3**
1	1.41421	−0.81650	−0.57735
2	0.00000	1.63299	−0.57735
5	0.00000	0.00000	1.73205
7	−1.41421	−0.81649	−0.57735

ORTHORDINAL

The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=ORDINAL. The design matrix columns for A are as follows:

	Design Matrix		
A	AOORD1	AOORD2	AOORD3
1	−1.73205	0.00000	0.00000
2	0.57735	−1.63299	0.00000
5	0.57735	0.81650	−1.41421
7	0.57735	0.81650	1.41421

ORTHPOLY

The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=POLY. The design matrix columns for A are as follows:

	Design Matrix		
A	AOPOLY1	AOPOLY2	AOPOLY5
1	−1.153	0.907	−0.921
2	−0.734	−0.540	1.473
5	0.524	−1.370	−0.921
7	1.363	1.004	0.368

ORTHREF

The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=REFERENCE. The design matrix columns for A are as follows:

	Design Matrix		
A	AOREF1	AOREF2	AOREF3
1	1.73205	0.00000	0.00000
2	−0.57735	1.63299	0.00000
5	−0.57735	−0.81650	1.41421
7	−0.57735	−0.81650	−1.41421

Clarification of the Time and CLASS Variables Usage

The following DATA step creates an artificial data set, Foo, to be used in this section. There are four variables in Foo: the variable T contains the failure times; the variable Status is the censoring indicator variable with the value 1 for an uncensored failure time and the value 0 for a censored time; the variable A is a categorical variable with values 1, 2, and 3 representing three different categories; and the variable MirrorT is an exact copy of T.

```
Data Foo;
   input T Status A @@;
   MirrorT = T;
   datalines;
 23       1      1    7        0      1
 23       1      1   10        1      1
```

```
20        0      1    13        0      1
24        1      1    10        1      1
18        1      2     6        1      2
18        0      2     6        1      2
13        0      2    13        1      2
 9        0      2    15        1      2
 8        1      3     6        1      3
12        0      3     4        1      3
11        1      3     8        1      1
 6        1      3     7        1      3
 7        1      3    12        1      3
 9        1      2    15        1      2
 3        1      2    14        0      3
 6        1      1    13        1      2
 ;
```

Time Variable on the Right Side of the MODEL Statement

When the time variable is explicitly used in an explanatory effect in the MODEL statement, the effect is NOT time dependent. In the following specification, T is the time variable, but T does not play the role of the time variable in the explanatory effect T*A:

```
proc phreg data=Foo;
   class A;
   model T*Status(0)=T*A;
   run;
```

The parameter estimates of this model are shown in Figure 64.12.

Figure 64.12 T*A Effect

The PHREG Procedure

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio	Label
T*A	1	-0.16549	0.05042	10.7734	0.0010	.	A 1 * T
T*A	2	-0.11852	0.04181	8.0344	0.0046	.	A 2 * T

To verify that the effect T*A in the MODEL statement is not time dependent, T is replaced by MirrorT, which is an exact copy of T, as in the following statements:

```
proc phreg data=Foo;
   class A;
   model T*Status(0)=A*MirrorT;
   run;
```

The results of fitting this model (Figure 64.13) are identical to those of the previous model (Figure 64.12), except for the parameter names and labels. The effect A*MirrorT is not time de-

pendent, so neither is A*T.

Figure 64.13 T*A Effect

The PHREG Procedure

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq
MirrorT*A 1	1	−0.16549	0.05042	10.7734	0.0010
MirrorT*A 2	1	−0.11852	0.04181	8.0344	0.0046

Analysis of Maximum Likelihood Estimates

Parameter	Hazard Ratio	Label
MirrorT*A 1	.	A 1 * MirrorT
MirrorT*A 2	.	A 2 * MirrorT

CLASS Variables and Programming Statements

In PROC PHREG, the levelization of CLASS variables is determined by the CLASS statement and the input data, and is not affected by user-supplied programming statements. Consider the following statements, which produce the results in Figure 64.14. Variable A is declared as a CLASS variable in the CLASS statement. By default, the reference parameterization is used with A=3 as the reference level. Two regression coefficients are estimated for the two dummy variables of A.

```
proc phreg data=Foo;
   class A;
   model T*Status(0)=A;
   run;
```

Figure 64.14 shows the dummy variables of A and the regression coefficients estimates.

Figure 64.14 Design Variable and Regression Coefficient Estimates

The PHREG Procedure

Class Level Information

Class	Value	Design Variables	
A	1	1	0
	2	0	1
	3	0	0

Figure 64.14 *continued*

Analysis of Maximum Likelihood Estimates

Parameter		DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio	Label
A	1	1	-1.40925	0.64802	4.7293	0.0297	0.244	A 1
A	2	1	-0.65705	0.51764	1.6112	0.2043	0.518	A 2

Now consider the programming statement that attempts to change the value of the CLASS variable A as in the following specification:

```
proc phreg data=Foo;
   class A;
   model T*Status(0)=A;
   if A=3 then A=2;
   run;
```

Results of this analysis are shown in Figure 64.15 and are identical to those in Figure 64.14. The **if A=3 then A=2** programming statement has no effects on the design variables for A, which have already been determined.

Figure 64.15 Design Variable and Regression Coefficient Estimates

The PHREG Procedure

Class Level Information

Class	Value	Design Variables	
A	1	1	0
	2	0	1
	3	0	0

Analysis of Maximum Likelihood Estimates

Parameter		DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio	Label
A	1	1	-1.40925	0.64802	4.7293	0.0297	0.244	A 1
A	2	1	-0.65705	0.51764	1.6112	0.2043	0.518	A 2

Additionally any variable used in a programming statement that has already been declared in the CLASS statement is NOT treated as a collection of the corresponding design variables. Consider the following statements:

```
proc phreg data=Foo;
   class A;
   model T*Status(0)=A X;
   X=T*A;
   run;
```

The CLASS variable A generates two design variables as explanatory variables. The variable X created by the **X=T*A** programming statement is a single time-dependent covariate whose values are evaluated using the exact values of A given in the data.

Figure 64.16 Single Time-Dependent Variable X*A

The PHREG Procedure

Analysis of Maximum Likelihood Estimates

Parameter		DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio	Label
A	1	1	0.15798	1.69338	0.0087	0.9257	1.171	A 1
A	2	1	0.00898	0.87573	0.0001	0.9918	1.009	A 2
X		1	0.09268	0.09535	0.9448	0.3311	1.097	

To generalize the simple test of proportional hazard assumption for the design variables of A (as in the section the "Classical Method of Maximum Likelihood" on page 4527), you specify the following statements, which are not the same as in the preceding program or the specification in the section "Time Variable on the Right Side of the MODEL Statement" on page 4583:

```
proc phreg data=Foo;
   class A;
   model T*Status(0)=A X1 X2;
   X1= T*(A=1);
   X2= T*(A=2);
   run;
```

Results of this test are shown in Figure 64.17.

Figure 64.17 Simple Test of Proportional Hazards Assumption

The PHREG Procedure

Analysis of Maximum Likelihood Estimates

Parameter		DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio	Label
A	1	1	-0.00766	1.69435	0.0000	0.9964	0.992	A 1
A	2	1	-0.88132	1.64298	0.2877	0.5917	0.414	A 2
X1		1	-0.15522	0.20174	0.5920	0.4417	0.856	
X2		1	0.01155	0.18858	0.0037	0.9512	1.012	

Partial Likelihood Function for the Cox Model

Let $\mathbf{Z}_l(t)$ denote the vector explanatory variables for the lth individual at time t. Let $t_1 < t_2 < \ldots < t_k$ denote the k distinct, ordered event times. Let d_i denote the multiplicity of failures at t_i; that is, d_i is the size of the set $\mathcal{D}_i$ of individuals that fail at t_i. Let w_l be the weight associated with the lth individual. Using this notation, the likelihood functions used in PROC PHREG to estimate $\boldsymbol{\beta}$ are described in the following sections.

Continuous Time Scale

Let $\mathcal{R}_i$ denote the risk set just before the ith ordered event time t_i. Let $\mathcal{R}_i^*$ denote the set of individuals whose event or censored times exceed t_i or whose censored times are equal to t_i.

Exact Likelihood

$$L_1(\boldsymbol{\beta}) = \prod_{i=1}^{k} \left\{ \int_0^{\infty} \prod_{j \in \mathcal{D}_i} \left[1 - \exp\left(-\frac{e^{\boldsymbol{\beta}'\mathbf{Z}_j(t_i)}}{\sum_{l \in \mathcal{R}_i^*} e^{\boldsymbol{\beta}'\mathbf{Z}_l(t_i)}} t \right) \right] \exp(-t)\, dt \right\}$$

Breslow Likelihood

$$L_2(\boldsymbol{\beta}) = \prod_{i=1}^{k} \frac{e^{\boldsymbol{\beta}' \sum_{j \in \mathcal{D}_i} \mathbf{Z}_j(t_i)}}{\left[\sum_{l \in \mathcal{R}_i} e^{\boldsymbol{\beta}'\mathbf{Z}_l(t_i)} \right]^{d_i}}$$

Incorporating weights, the Breslow likelihood becomes

$$L_2(\boldsymbol{\beta}) = \prod_{i=1}^{k} \frac{e^{\boldsymbol{\beta}' \sum_{j \in \mathcal{D}_i} w_j \mathbf{Z}_j(t_i)}}{\left[\sum_{l \in \mathcal{R}_i} w_l e^{\boldsymbol{\beta}'\mathbf{Z}_l(t_i)} \right]^{\sum_{j \in \mathcal{D}_i} w_i}}$$

Efron Likelihood

$$L_3(\boldsymbol{\beta}) = \prod_{i=1}^{k} \frac{e^{\boldsymbol{\beta}' \sum_{j \in \mathcal{D}_i} \mathbf{Z}_j(t_i)}}{\prod_{j=1}^{d_i} \left(\sum_{l \in \mathcal{R}_i} e^{\boldsymbol{\beta}'\mathbf{Z}_l(t_i)} - \frac{j-1}{d_i} \sum_{l \in \mathcal{D}_i} e^{\boldsymbol{\beta}'\mathbf{Z}_l(t_i)} \right)}$$

Incorporating weights, the Efron likelihood becomes

$$L_3(\boldsymbol{\beta}) = \prod_{i=1}^{k} \frac{e^{\boldsymbol{\beta}' \sum_{j \in \mathcal{D}_i} w_j \mathbf{Z}_j(t_i)}}{\left[\prod_{j=1}^{d_i} \left(\sum_{l \in \mathcal{R}_i} w_l e^{\boldsymbol{\beta}' \mathbf{Z}_l(t_i)} - \frac{j-1}{d_i} \sum_{l \in \mathcal{D}_i} w_l e^{\boldsymbol{\beta}' \mathbf{Z}_l(t_i)} \right) \right]^{\frac{1}{d_i} \sum_{j \in \mathcal{D}_i} w_j}}$$

Discrete Time Scale

Let $\mathcal{Q}_i$ denote the set of all subsets of d_i individuals from the risk set $\mathcal{R}_i$. For each $\mathbf{q} \in \mathcal{Q}_i$, $\mathbf{q}$ is a d_i-tuple $(q_1, q_2, \ldots, q_{d_i})$ of individuals who might have failed at t_i.

Discrete Logistic Likelihood

$$L_4(\boldsymbol{\beta}) = \prod_{i=1}^{k} \frac{e^{\boldsymbol{\beta}' \sum_{j \in \mathcal{D}_i} \mathbf{Z}_j(t_i)}}{\sum_{\boldsymbol{q} \in \mathcal{Q}_i} e^{\boldsymbol{\beta}' \sum_{l=1}^{d_j} \mathbf{Z}_{q_l}(t_i)}}$$

The computation of $L_4(\boldsymbol{\beta})$ and its derivatives is based on an adaptation of the recurrence algorithm of Gail, Lubin, and Rubinstein (1981) to the logarithmic scale. When there are no ties on the event times (that is, $d_i \equiv 1$), all four likelihood functions $L_1(\boldsymbol{\beta})$, $L_2(\boldsymbol{\beta})$, $L_3(\boldsymbol{\beta})$, and $L_4(\boldsymbol{\beta})$ reduce to the same expression. In a stratified analysis, the partial likelihood is the product of the partial likelihood functions for the individual strata.

Counting Process Style of Input

In the counting process formulation, data for each subject are identified by a triple $\{N, Y, \mathbf{Z}\}$ of counting, at-risk, and covariate processes. Here, $N(t)$ indicates the number of events that the subject experiences over the time interval $(0, t]$; $Y(t)$ indicates whether the subject is at risk at time t (one if at risk and zero otherwise); and $\mathbf{Z}(t)$ is a vector of explanatory variables for the subject at time t. The sample path of N is a step function with jumps of size +1 at the event times, and $N(0) = 0$. Unless $\mathbf{Z}(t)$ changes continuously with time, the data for each subject can be represented by multiple observations, each identifying a semiclosed time interval $(t1, t2]$, the values of the explanatory variables over that interval, and the event status at $t2$. The subject remains at risk during the interval $(t1, t2]$, and an event might occur at $t2$. Values of the explanatory variables for the subject remain unchanged in the interval. This style of data input was originated by Therneau (1994).

For example, a patient has a tumor recurrence at weeks 3, 10, and 15 and is followed up to week 23. The explanatory variables are Trt (treatment), Z1 (initial tumor number), and Z2 (initial tumor size), and, for this patient, the values of Trt, Z1, and Z2 are (1,1,3). The data for this patient are represented by the following four observations:

T1	T2	Event	Trt	Z1	Z2
0	3	1	1	1	3
3	10	1	1	1	3
10	15	1	1	1	3
15	23	0	1	1	3

Here (T1,T2] contains the at-risk intervals. The variable Event is a censoring variable indicating whether a recurrence has occurred at T2; a value of 1 indicates a tumor recurrence, and a value of 0 indicates nonrecurrence. The PHREG procedure fits the multiplicative hazards model, which is specified as follows:

```
proc phreg;
   model (T1,T2) * Event(0) = Trt Z1 Z2;
run;
```

Another useful application of the counting process formulation is delayed entry of subjects into the risk set. For example, in studying the mortality of workers exposed to a carcinogen, the survival time is chosen to be the worker's age at death by malignant neoplasm. Any worker joining the workplace at a later age than a given event failure time is not included in the corresponding risk set. The variables of a worker consist of Entry (age at which the worker entered the workplace), Age (age at death or age censored), Status (an indicator of whether the observation time is censored, with the value 0 identifying a censored time), and X1 and X2 (explanatory variables thought to be related to survival). The specification for such an application is as follows:

```
proc phreg;
   model (Entry, Age) * Status(0) = X1 X2;
run;
```

Alternatively, you can use a time-dependent variable to control the risk set, as illustrated in the following specification:

```
proc phreg;
   model Age * Status(0) = X1 X2;
   if Age < Entry then X1= .;
run;
```

Here, X1 becomes a time-dependent variable. At a given death time t, the value of X1 is reevaluated for each subject with Age $\geq t$; subjects with Entry $> t$ are given a missing value in X1 and are subsequently removed from the risk set. Computationally, this approach is not as efficient as the one that uses the counting process formulation.

Left Truncation of Failure Times

Left truncation arises when individuals come under observation only some known time after the natural time origin of the phenomenon under study. The risk set just prior to an event time does not include individuals whose left truncation times exceed the given event time. Thus, any contribution to the likelihood must be conditional on the truncation limit having been exceeded.

An alternative way to specify left truncation in PROC PHREG is through the counting process style of input. The following specifications are equivalent:

```
proc phreg data=one;
   model t2*dead(0)=x1-x10/entry=t1;
   title 'The ENTRY= option is Specified';
run;

proc phreg data=one;
   model (t1,t2)*dead(0)=x1-x10;
   title 'Counting Process Style of Input';
run;
```

The Multiplicative Hazards Model

Consider a set of n subjects such that the counting process $N_i \equiv \{N_i(t), t \geq 0\}$ for the ith subject represents the number of observed events experienced over time t. The sample paths of the process N_i are step functions with jumps of size $+1$, with $N_i(0) = 0$. Let $\boldsymbol{\beta}$ denote the vector of unknown regression coefficients. The multiplicative hazards function $\Lambda(t, \mathbf{Z}_i(t))$ for N_i is given by

$$Y_i(t)d\Lambda(t, \mathbf{Z}_i(t)) = Y_i(t)\exp(\boldsymbol{\beta}'\mathbf{Z}_i(t))d\Lambda_0(t)$$

where

- $Y_i(t)$ indicates whether the ith subject is at risk at time t (specifically, $Y_i(t) = 1$ if at risk and $Y_i(t) = 0$ otherwise)
- $\mathbf{Z}_i(t)$ is the vector of explanatory variables for the ith subject at time t
- $\Lambda_0(t)$ is an unspecified baseline hazard function

Refer to Fleming and Harrington (1991) and Andersen et al. (1992). The Cox model is a special case of this multiplicative hazards model, where $Y_i(t) = 1$ until the first event or censoring, and $Y_i(t) = 0$ thereafter.

The partial likelihood for n independent triplets $(N_i, Y_i, \mathbf{Z}_i), i = 1, \ldots, n$, has the form

$$\mathcal{L}(\boldsymbol{\beta}) = \prod_{i=1}^{n}\prod_{t\geq 0}\left\{\frac{Y_i(t)\exp(\boldsymbol{\beta}'\mathbf{Z}_i(t))}{\sum_{j=1}^{n}Y_j(t)\exp(\boldsymbol{\beta}'\mathbf{Z}_j(t))}\right\}^{\Delta N_i(t)}$$

where $\Delta N_i(t) = 1$ if $N_i(t) - N_i(t-) = 1$, and $\Delta N_i(t) = 0$ otherwise.

Hazard Ratios

Consider a dichotomous risk factor variable X that takes the value 1 if the risk factor is present and 0 if the risk factor is absent. The log-hazard function is given by

$$\log[\lambda(t|X)] = \log[\lambda_0(t)] + \beta_1 X$$

where $\lambda_0(t)$ is the baseline hazard function.

The hazard ratio ψ is defined as the ratio of the hazard for those with the risk factor ($X = 1$) to the hazard without the risk factor ($X = 0$). The log of the hazard ratio is given by

$$\log(\psi) \equiv \log[\psi(X = 1, X = 0)] = \log[\lambda(t|X = 1)] - \log[\lambda(t|X = 0)] = \beta_1$$

In general, the hazard ratio can be computed by exponentiating the difference of the log-hazard between any two population profiles. This is the approach taken by the HAZARDRATIO statement, so the computations are available regardless of parameterization, interactions, and nestings. However, as shown in the preceding equation for $\log(\psi)$, hazard ratios of main effects can be computed as functions of the parameter estimates, and the remainder of this section is concerned with this methodology.

The parameter, β_1, associated with X represents the change in the log-hazard from $X = 0$ to $X = 1$. So the hazard ratio is obtained by simply exponentiating the value of the parameter associated with the risk factor. The hazard ratio indicates how the hazard change as you change X from 0 to 1. For instance, $\psi = 2$ means that the hazard when $X = 1$ is twice the hazard when $X = 0$.

Suppose the values of the dichotomous risk factor are coded as constants a and b instead of 0 and 1. The hazard when $X = a$ becomes $\lambda(t)\exp(a\beta_1)$, and the hazard when $X = b$ becomes $\lambda(t)\exp(b\beta_1)$. The hazard ratio corresponding to an increase in X from a to b is

$$\psi = \exp[(b-a)\beta_1] = [\exp(\beta_1)]^{b-a} \equiv [\exp(\beta_1)]^c$$

Note that for any a and b such that $c = b - a = 1, \psi = \exp(\beta_1)$. So the hazard ratio can be interpreted as the change in the hazard for any increase of one unit in the corresponding risk factor. However, the change in hazard for some amount other than one unit is often of greater interest. For example, a change of one pound in body weight might be too small to be considered important, while a change of 10 pounds might be more meaningful. The hazard ratio for a change in X from a to b is estimated by raising the hazard ratio estimate for a unit change in X to the power of $c = b-a$ as shown previously.

For a polytomous risk factor, the computation of hazard ratios depends on how the risk factor is parameterized. For illustration, suppose that Cell is a risk factor with four categories: Adeno, Large, Small, and Squamous.

For the effect parameterization scheme (PARAM=EFFECT) with Squamous as the reference group, the design variables for Cell are as follows:

	Design Variables		
Cell	X_1	X_2	X_3
Adeno	1	0	0
Large	0	1	0
Small	0	0	1
Squamous	−1	−1	−1

The log-hazard for Adeno is

$$\begin{aligned}\log[\lambda(t|\text{Adeno})] &= \log[\lambda_0(t)] + \beta_1(X_1 = 1) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0) \\ &= \lambda_0(t) + \beta_1\end{aligned}$$

The log-hazard for Squamous is

$$\begin{aligned}\log[\lambda(t|\text{Squamous})] &= \log[\lambda_0(t)] + \beta_1(X_1 = -1) + \beta_2(X_2 = -1) + \beta_3(X_3 = -1)) \\ &= \log[\lambda_0(t)] - \beta_1 - \beta_2 - \beta_3\end{aligned}$$

Therefore, the log-hazard ratio of Adeno versus Squamous

$$\begin{aligned}\log[\psi(\text{Adeno, Squamous})] &= \log[\lambda(t|\text{Adeno})] - \log[\lambda(t|\text{Squamous})] \\ &= 2\beta_1 + \beta_2 + \beta_3\end{aligned}$$

For the reference cell parameterization scheme (PARAM=REF) with Squamous as the reference cell, the design variables for Cell are as follows:

	Design Variables		
Cell	X_1	X_2	X_3
Adeno	1	0	0
Large	0	1	0
Small	0	0	1
Squamous	0	0	0

The log-hazard ratio of Adeno versus Squamous is given by

$$\begin{aligned}&\log(\psi(\text{Adeno, Squamous})) \\ &= \log[\lambda(t|\text{Adeno})] - \log[\lambda(t|\text{Squamous})] \\ &= (\log[\lambda_0(t)] + \beta_1(X_1 = 1) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0)) - \\ &\quad (\log[\lambda_0(t)] + \beta_1(X_1 = 0) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0)) \\ &= \beta_1\end{aligned}$$

For the GLM parameterization scheme (PARAM=GLM), the design variables are as follows:

	Design Variables			
Cell	X_1	X_2	X_3	X_4
Adeno	1	0	0	0
Large	0	1	0	0
Small	0	0	1	0
Squamous	0	0	0	1

The log-hazard ratio of Adeno versus Squamous is

$$\begin{aligned}
&\log(\psi(\text{Adeno}, \text{Squamous})) \\
&\quad = \log[\lambda(t|\text{Adeno})] - \log[\lambda(t|\text{Squamous})] \\
&\quad = \log[\lambda_0(t)] + \beta_1(X_1 = 1) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0) + \beta_4(X_4 = 0)) - \\
&\qquad (\log(\lambda_0(t)) + \beta_1(X_1 = 0) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0) + \beta_4(X_4 = 1)) \\
&\quad = \beta_1 - \beta_4
\end{aligned}$$

Consider Cell as the only risk factor in the Cox regression in Example 64.3. The computation of hazard ratio of Adeno versus Squamous for various parameterization schemes is tabulated in Table 64.4.

Table 64.4 Hazard Ratio Comparing Adeno to Squamous

	Parameter Estimates				
PARAM=	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	**Hazard Ratio Estimates**
EFFECT	0.5772	–0.2115	0.2454		$\exp(2 \times 0.5772 - 0.2115 + 0.2454) = 3.281$
REF	1.8830	0.3996	0.8565		$\exp(1.8830) = 3.281$
GLM	1.8830	0.3996	0.8565	0.0000	$\exp(1.8830) = 3.281$

The fact that the log-hazard ratio ($\log(\psi)$) is a linear function of the parameters enables the HAZARDRATIO statement to compute the hazard ratio of the main effect even in the presence of interactions and nest effects. The section "Hazard Ratios" on page 4590 details the estimation of the hazard ratios in a classical analysis.

To customize hazard ratios for specific units of change for a continuous risk factor, you can use the UNITS= option in a HAZARDRATIO statement to specify a list of relevant units for each explanatory variable in the model. Estimates of these customized hazard ratios are given in a separate table. Let (L_j, U_j) be a confidence interval for $\log(\psi)$. The corresponding lower and upper confidence limits for the customized hazard ratio $\exp(c\beta_j)$ are $\exp(cL_j)$ and $\exp(cU_j)$, respectively (for $c > 0$), or $\exp(cU_j)$ and $\exp(cL_j)$, respectively (for $c < 0$).

Specifics for Classical Analysis

Proportional Rates/Means Models for Recurrent Events

Let $N(t)$ be the number of events experienced by a subject over the time interval $(0, t]$. Let $dN(t)$ be the increment of the counting process N over $[t, t + dt)$. The rate function is given by

$$d\mu_{\mathbf{Z}}(t) = E[dN(t)|Z(t)] = e^{\boldsymbol{\beta}'\mathbf{Z}(t)} d\mu_0(t)$$

where $\mu_0(.)$ is an unknown continuous function. If the $\mathbf{Z}$ are time independent, the rate model is reduced to the mean model

$$\mu_Z(t) = \mathrm{e}^{\boldsymbol{\beta}' Z}\mu_0(t)$$

The partial likelihood for n independent triplets $(N_i, Y_i, \mathbf{Z}_i), i = 1, \ldots, n$, of counting, at-risk, and covariate processes is the same as that of the multiplicative hazards model. However, a robust sandwich estimate is used for the covariance matrix of the parameter estimator instead of the model-based estimate.

Let T_{ki} be the kth event time of the ith subject. Let C_i be the censoring time of the ith subject. The at-risk indicator and the failure indicator are, respectively,

$$Y_i(t) = I(C_i \geq t) \text{and} \Delta_{ki} = I(T_{ki} \leq C_i)$$

Denote

$$S^{(0)}(\boldsymbol{\beta}, t) = \sum_{i=1}^{n} Y_i(t)\mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_i(t)} \text{and} \bar{\mathbf{Z}}(\boldsymbol{\beta}, t) = \frac{\sum_{i=1}^{n} Y_i(t)\mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_i(t)}\mathbf{Z}_i(t)}{S^{(0)}(\boldsymbol{\beta}, t)}$$

Let $\hat{\boldsymbol{\beta}}$ be the maximum likelihood estimate of $\boldsymbol{\beta}$, and let $\mathcal{I}(\hat{\boldsymbol{\beta}})$ be the observed information matrix. The robust sandwich covariance matrix estimate is given by

$$\mathcal{I}^{-1}(\hat{\boldsymbol{\beta}}) \sum_{i=1}^{n} \left[W_i(\hat{\boldsymbol{\beta}}) W_i'(\hat{\boldsymbol{\beta}}) \right] \mathcal{I}^{-1}(\hat{\boldsymbol{\beta}})$$

where

$$\begin{aligned} W_i(\boldsymbol{\beta}) &= \sum_k \Delta_{ki} \left\{ Z_i(T_{ki}) - \bar{\mathbf{Z}}(\boldsymbol{\beta}, T_{ki}) \right\} - \\ &\quad \sum_{i=1}^{n} \sum_l \frac{\Delta_{lj} Y_i(T_{lj}) \mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_i(T_{lj})}}{S^0(\boldsymbol{\beta}, T_{lj})} \left\{ Z_i(T_{lj}) - \bar{\mathbf{Z}}(\boldsymbol{\beta}, T_{lj}) \right\} \end{aligned}$$

For a given realization of the covariates $\boldsymbol{\xi}$, the Nelson estimator is used to predict the mean function

$$\hat{\mu}_{\boldsymbol{\xi}}(t) = \mathrm{e}^{\hat{\boldsymbol{\beta}}'\boldsymbol{\xi}} \sum_{i=1}^{n} \sum_k \frac{I(T_{ki} \leq t)\Delta_{ki}}{S^{(0)}(\hat{\boldsymbol{\beta}}, T_{ki})}$$

with standard error estimate given by

$$\hat{\sigma}^2(\hat{\mu}_{\boldsymbol{\xi}}(t)) = \sum_{i=1}^{n} \left(\frac{1}{n} \hat{\Psi}_i(t, \boldsymbol{\xi}) \right)^2$$

where

$$\begin{aligned} \frac{1}{n}\hat{\Psi}_i(\boldsymbol{\xi}, t) &= \mathrm{e}^{\hat{\boldsymbol{\beta}}'\boldsymbol{\xi}} \left\{ \sum_k \frac{I(T_{ki} \leq t)\Delta_{ik}}{S^{(0)}(\hat{\boldsymbol{\beta}}, T_{ki})} - \sum_{j=1}^{n} \sum_k \frac{Y_i(T_{kj}) \mathrm{e}^{\hat{\boldsymbol{\beta}}'\mathbf{Z}_i(T_{kj})} I(T_{kj} \leq t)\Delta_{kj}}{[S^{(0)}(\hat{\boldsymbol{\beta}}, T_{kj})]^2} - \right. \\ &\quad \left[\sum_{i=1}^{n} \sum_k \frac{I(T_{ki} \leq t)\Delta_{ik}[\bar{\mathbf{Z}}(\hat{\boldsymbol{\beta}}, T_{ki}) - \boldsymbol{\xi}]}{S^{(0)}(\hat{\boldsymbol{\beta}}, T_{ki})} \right] \\ &\quad \left. \times \mathcal{I}^{-1}(\hat{\boldsymbol{\beta}}) \int_0^{\tau} [\mathbf{Z}_i(u) - \bar{\mathbf{Z}}(\hat{\boldsymbol{\beta}}, u)] d\hat{M}_i(u) \right\} \end{aligned}$$

Since the cumulative mean function is always nonnegative, the log transform is used to compute confidence intervals. The $100(1-\alpha)\%$ pointwise confidence limits for $\mu_{\boldsymbol{\xi}}(t)$ are

$$\hat{\mu}_{\boldsymbol{\xi}}(t) \mathrm{e}^{\pm z_{\alpha/2} \frac{\hat{\sigma}(\hat{\mu}_{\boldsymbol{\xi}}(t))}{\hat{\mu}_{\boldsymbol{\xi}}(t)}}$$

where $z_{\alpha/2}$ is the upper $100\alpha/2$ percentage point of the standard normal distribution.

Newton-Raphson Method

Let $L(\boldsymbol{\beta})$ be one of the likelihood functions described in the previous subsections. Let $l(\boldsymbol{\beta}) = \log L(\boldsymbol{\beta})$. Finding $\boldsymbol{\beta}$ such that $L(\boldsymbol{\beta})$ is maximized is equivalent to finding the solution $\hat{\boldsymbol{\beta}}$ to the likelihood equations

$$\frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = 0$$

With $\hat{\boldsymbol{\beta}}^0 = \mathbf{0}$ as the initial solution, the iterative scheme is expressed as

$$\hat{\boldsymbol{\beta}}^{j+1} = \hat{\boldsymbol{\beta}}^j - \left[\frac{\partial^2 l(\hat{\boldsymbol{\beta}}^j)}{\partial \boldsymbol{\beta}^2}\right]^{-1} \frac{\partial l(\hat{\boldsymbol{\beta}}^j)}{\partial \boldsymbol{\beta}}$$

The term after the minus sign is the Newton-Raphson step. If the likelihood function evaluated at $\hat{\boldsymbol{\beta}}^{j+1}$ is less than that evaluated at $\hat{\boldsymbol{\beta}}^j$, then $\hat{\boldsymbol{\beta}}^{j+1}$ is recomputed using half the step size. The iterative scheme continues until convergence is obtained—that is, until $\hat{\boldsymbol{\beta}}_{j+1}$ is sufficiently close to $\hat{\boldsymbol{\beta}}_j$. Then the maximum likelihood estimate of $\boldsymbol{\beta}$ is $\hat{\boldsymbol{\beta}} = \hat{\boldsymbol{\beta}}_{j+1}$.

The model-based variance estimate of $\hat{\boldsymbol{\beta}}$ is obtained by inverting the information matrix $\mathcal{I}(\hat{\boldsymbol{\beta}})$

$$\hat{\mathbf{V}}_m(\hat{\boldsymbol{\beta}}) = \mathcal{I}^{-1}(\hat{\boldsymbol{\beta}}) = -\left[\frac{\partial^2 l(\hat{\boldsymbol{\beta}})}{\partial \boldsymbol{\beta}^2}\right]^{-1}$$

Firth's Modification for Maximum Likelihood Estimation

In fitting a Cox model, the phenomenon of monotone likelihood is observed if the likelihood converges to a finite value while at least one parameter diverges (Heinze and Schemper 2001).

Let $\mathbf{x}_l(t)$ denote the vector explanatory variables for the lth individual at time t. Let $t_1 < t_2 < \ldots < t_m$ denote the k distinct, ordered event times. Let d_j denote the multiplicity of failures at t_j; that is, d_j is the size of the set $\mathcal{D}_j$ of individuals that fail at t_j. Let $\mathcal{R}_j$ denote the risk set just before t_j. Let $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)'$ be the vector of regression parameters. The Breslow log partial likelihood is given by

$$l(\boldsymbol{\beta}) = \log L(\boldsymbol{\beta}) = \sum_{j=1}^{m} \left\{ \boldsymbol{\beta}' \sum_{l \in \mathcal{D}_j} \mathbf{x}_l(t_j) - d_j \log \sum_{h \in \mathcal{R}_j} \mathrm{e}^{\boldsymbol{\beta}' \mathbf{x}_h(t_j)} \right\}$$

Denote

$$\mathbf{S}_j^{(a)}(\boldsymbol{\beta}) = \sum_{h \in \mathcal{R}j} \mathrm{e}^{\boldsymbol{\beta}' \mathbf{x}_h(t_j)} [\mathbf{x}_h(t_j)]^{\otimes a} \qquad a = 0, 1, 2$$

Then the score function is given by

$$\begin{aligned} \mathbf{U}(\boldsymbol{\beta}) &\equiv (U(\beta_1), \ldots, U(\beta_k))' \\ &= \frac{\partial l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} \\ &= \sum_{j=1}^{m} \left\{ \sum_{l \in \mathcal{D}j} \mathbf{x}_l(t_j) - d_j \frac{\mathbf{S}_j^{(1)}(\boldsymbol{\beta})}{S_j^0(\boldsymbol{\beta})} \right\} \end{aligned}$$

and the Fisher information matrix is given by

$$\begin{aligned} \mathcal{I}(\boldsymbol{\beta}) &= -\frac{\partial^2 l(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^2} \\ &= \sum_{j=1}^{m} d_j \left\{ \frac{\mathbf{S}_j^{(2)}(\boldsymbol{\beta})}{S_j^{(0)}(\boldsymbol{\beta})} - \left[\frac{\mathbf{S}_j^{(1)}(\boldsymbol{\beta})}{\mathbf{S}_j^{(0)}(\boldsymbol{\beta})} \right] \left[\frac{\mathbf{S}_j^{(1)}(\boldsymbol{\beta})}{\mathbf{S}_j^{(0)}(\boldsymbol{\beta})} \right]' \right\} \end{aligned}$$

Heinze (1999); Heinze and Schemper (2001) applied the idea of Firth (1993) by maximizing the penalized partial likelihood

$$l^*(\boldsymbol{\beta}) = l(\boldsymbol{\beta}) + 0.5 \log(|\mathcal{I}(\boldsymbol{\beta})|)$$

The score function $\mathbf{U}(\boldsymbol{\beta})$ is replaced by the modified score function by $\mathbf{U}^*(\boldsymbol{\beta}) \equiv (U^*(\beta_1), \ldots, U^*(\beta_k))'$, where

$$U^*(\beta_r) = U(\beta_r) + 0.5 \mathrm{tr} \left\{ \mathcal{I}^{-1}(\boldsymbol{\beta}) \frac{\partial \mathcal{I}(\boldsymbol{\beta})}{\partial \beta_r} \right\} \qquad r = 1, \ldots, k$$

The Firth estimate is obtained iteratively as

$$\boldsymbol{\beta}^{(s+1)} = \boldsymbol{\beta}^{(s)} + \mathcal{I}^{-1}(\boldsymbol{\beta}^{(s)}) \mathbf{U}^*(\boldsymbol{\beta}^{(s)})$$

The covariance matrix $\hat{\mathbf{V}}$ is computed as $\mathcal{I}^{-1}(\hat{\boldsymbol{\beta}})$, where $\hat{\boldsymbol{\beta}}$ is the maximum penalized partial likelihood estimate.

Explicit formulae for $\frac{\partial \mathcal{I}(\boldsymbol{\beta})}{\partial \beta_r}, r = 1, \ldots, k$

Denote

$$\begin{aligned} \mathbf{x}_h(t) &= (x_{h1}(t), \ldots, x_{hk}(t))' \\ \mathbf{Q}_{jr}^{(a)}(\boldsymbol{\beta}) &= \sum_{h \in \mathcal{R}j} \mathrm{e}^{\boldsymbol{\beta}' \mathbf{x}_h(t_j)} x_{hr}(t_j) [\mathbf{x}_h(t_j)]^{\otimes a} \qquad a = 0, 1, 2; r = 1, \ldots, k \end{aligned}$$

Then

$$\begin{aligned}\frac{\partial \mathcal{I}(\boldsymbol{\beta})}{\partial \beta_r} &= \sum_{j=1}^{m} d_j \left\{ \left[\frac{\mathbf{Q}_{jr}^{(2)}(\boldsymbol{\beta})}{S_j^{(0)}(\boldsymbol{\beta})} - \frac{\mathbf{Q}_{jr}^{(0)}(\boldsymbol{\beta})}{S_j^{(0)}(\boldsymbol{\beta})} \frac{\mathbf{S}_j^{(2)}(\boldsymbol{\beta})}{S_j^{(0)}(\boldsymbol{\beta})} \right] - \right. \\ &\quad \left[\frac{\mathbf{Q}_{jr}^{(1)}(\boldsymbol{\beta})}{S_j^{(0)}(\boldsymbol{\beta})} - \frac{\mathbf{Q}_{jr}^{(0)}(\boldsymbol{\beta})}{S_j^{(0)}(\boldsymbol{\beta})} \frac{\mathbf{S}_j^{(1)}(\boldsymbol{\beta})}{S_j^{(0)}(\boldsymbol{\beta})} \right] \left[\frac{\mathbf{S}_j^{(1)}(\boldsymbol{\beta})}{S_j^{(0)}(\boldsymbol{\beta})} \right]' - \\ &\quad \left. \left[\frac{\mathbf{S}_j^{(1)}(\boldsymbol{\beta})}{S_j^{(0)}(\boldsymbol{\beta})} \right] \left[\frac{\mathbf{Q}_{jr}^{(1)}(\boldsymbol{\beta})}{S_j^{(0)}(\boldsymbol{\beta})} - \frac{\mathbf{Q}_{jr}^{(0)}(\boldsymbol{\beta})}{S_j^{(0)}(\boldsymbol{\beta})} \frac{\mathbf{S}_j^{(1)}(\boldsymbol{\beta})}{S_j^{(0)}(\boldsymbol{\beta})} \right]' \right\} \qquad r = 1, \ldots, k\end{aligned}$$

Robust Sandwich Variance Estimate

For the ith subject, $i = 1, \ldots, n$, let X_i, w_i, and $\mathbf{Z}_i(t)$ be the observed time, weight, and the covariate vector at time t, respectively. Let Δ_i be the event indicator and let $Y_i(t) = I(X_i \geq t)$. Let

$$S^{(r)}(\boldsymbol{\beta}, t) = \sum_{j=1}^{n} w_j Y_j(t) \mathrm{e}^{\boldsymbol{\beta}' \mathbf{Z}_j(t)} \mathbf{Z}_j^{\otimes r}(t), r = 0, 1$$

Let $\bar{\mathbf{Z}}(\boldsymbol{\beta}, t) = \frac{S^{(1)}(\boldsymbol{\beta}, t)}{S^{(0)}(\boldsymbol{\beta}, t)}$. The score residual for the ith subject is

$$\mathbf{L}_i(\boldsymbol{\beta}) = \Delta_i \left\{ \mathbf{Z}_i(X_i) - \bar{\mathbf{Z}}(\beta, X_i) \right\} - \sum_{j=1}^{n} \Delta_j \frac{w_j Y_i(X_j) \mathrm{e}^{\boldsymbol{\beta}' \mathbf{Z}_i(X_j)}}{S^{(0)}(\boldsymbol{\beta}, X_j)} \left\{ \mathbf{Z}_i(X_j) - \bar{\mathbf{Z}}(\boldsymbol{\beta}, X_j) \right\}$$

For TIES=EFRON, the computation of the score residuals is modified to comply with the Efron partial likelihood. See the section "Residuals" on page 4611 for more information.

The robust sandwich variance estimate of $\hat{\boldsymbol{\beta}}$ derived by Binder (1992), who incorporated weights into the analysis, is

$$\hat{\mathbf{V}}_s(\hat{\boldsymbol{\beta}}) = \mathcal{I}^{-1}(\hat{\boldsymbol{\beta}}) \left[\sum_{j=1}^{n} (w_j \mathbf{L}_j(\hat{\boldsymbol{\beta}}))^{\otimes 2} \right] \mathcal{I}^{-1}(\hat{\boldsymbol{\beta}})$$

where $\mathcal{I}(\hat{\boldsymbol{\beta}})$ is the observed information matrix, and $\mathbf{a}^{\otimes 2} = \mathbf{a}\mathbf{a}'$. Note that when $w_i \equiv 1$,

$$\hat{\mathbf{V}}_s(\hat{\boldsymbol{\beta}}) = \mathbf{D}'\mathbf{D}$$

where $\mathbf{D}$ is the matrix of DFBETA residuals. This robust variance estimate was proposed by Lin and Wei (1989) and Reid and Crèpeau (1985).

Testing the Global Null Hypothesis

The following statistics can be used to test the global null hypothesis H_0: $\boldsymbol{\beta}$=$\mathbf{0}$. Under mild assumptions, each statistic has an asymptotic chi-square distribution with p degrees of freedom given the null hypothesis. The value p is the dimension of $\boldsymbol{\beta}$. For clustered data, the likelihood ratio test, the score test, and the Wald test assume independence of observations within a cluster, while the robust Wald test and the robust score test do not need such an assumption.

Likelihood Ratio Test

$$\chi^2_{LR} = 2\left[l(\hat{\boldsymbol{\beta}}) - l(\mathbf{0})\right]$$

Score Test

$$\chi^2_S = \left[\frac{\partial l(\mathbf{0})}{\partial \boldsymbol{\beta}}\right]' \left[-\frac{\partial^2 l(\mathbf{0})}{\partial \boldsymbol{\beta}^2}\right]^{-1} \left[\frac{\partial l(\mathbf{0})}{\partial \boldsymbol{\beta}}\right]$$

Wald's Test

$$\chi^2_W = \hat{\boldsymbol{\beta}}' \left[-\frac{\partial^2 l(\hat{\boldsymbol{\beta}})}{\partial \boldsymbol{\beta}^2}\right] \hat{\boldsymbol{\beta}}$$

Robust Score Test

$$\chi^2_{RS} = \left[\sum_i \mathbf{L}_i^0\right]' \left[\sum_i \mathbf{L}_i^0 \mathbf{L}_i^{0\prime}\right]^{-1} \left[\sum_i \mathbf{L}_i^0\right]$$

where $\mathbf{L}_i^0$ is the score residual of the ith subject at $\boldsymbol{\beta}$=$\mathbf{0}$; that is, $\mathbf{L}_i^0$=$\mathbf{L}_i(\mathbf{0}, \infty)$, where the score process $\mathbf{L}_i(\boldsymbol{\beta}, t)$ is defined in the section "Residuals" on page 4611.

Robust Wald's Test

$$\chi^2_{RW} = \hat{\boldsymbol{\beta}}' [\hat{\mathbf{V}}_s(\hat{\boldsymbol{\beta}})]^{-1} \hat{\boldsymbol{\beta}}$$

where $\hat{\mathbf{V}}_s(\hat{\boldsymbol{\beta}})$ is the sandwich variance estimate (see the section "Robust Sandwich Variance Estimate" on page 4597 for details).

Confidence Limits for a Hazard Ratio

Let $\mathbf{e}_j$ be the jth unit vector—that is, the jth entry of the vector is 1 and all other entries are 0. The hazards ratio for the explanatory variable with regression coefficient $\beta_j = \mathbf{e}_j' \boldsymbol{\beta}$ is defined as $\exp(\beta_j)$. In general, a log-hazard ratio can be written as $\mathbf{h}'\boldsymbol{\beta}$, a linear combination of the regression coefficients, and the hazard ratio $\exp(\mathbf{h}'\boldsymbol{\beta})$ is obtained by replacing $\mathbf{e}_j$ with $\mathbf{h}$.

Point Estimate

The hazards ratio exp$\mathbf{e}_j'\boldsymbol{\beta}$ is estimated by exp$(\mathbf{e}_j'\hat{\beta})$, where $\hat{\boldsymbol{\beta}}$ is the maximum likelihood estimate of the $\boldsymbol{\beta}$.

Wald's Confidence Limits

The $100(1-\alpha)\%$ confidence limits for the hazard ratio are calculated as

$$\exp\left(\mathbf{e}_j'\hat{\beta} \pm z_{\alpha/2}\sqrt{\mathbf{e}_j'\hat{\mathbf{V}}(\hat{\boldsymbol{\beta}})\mathbf{e}_j}\right)$$

where $\hat{\mathbf{V}}(\hat{\boldsymbol{\beta}})$ is estimated covariance matrix, and $z_{\alpha/2}$ is the $100(1-\alpha/2)$ percentile point of the standard normal distribution.

Profile-Likelihood Confidence Limits

The construction of the profile-likelihood-based confidence interval is derived from the asymptotic χ^2 distribution of the generalized likelihood ratio test of Venzon and Moolgavkar (1988). Suppose that the parameter vector is $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)'$ and you want to compute a confidence interval for β_j. The profile-likelihood function for $\beta_j = \gamma$ is defined as

$$l_j^*(\gamma) = \max_{\boldsymbol{\beta}\in\mathcal{B}_j(\gamma)} l(\boldsymbol{\beta})$$

where $\mathcal{B}_j(\gamma)$ is the set of all $\boldsymbol{\beta}$ with the jth element fixed at γ, and $l(\boldsymbol{\beta})$ is the log-likelihood function for $\boldsymbol{\beta}$. If $l_{\max} = l(\hat{\boldsymbol{\beta}})$ is the log likelihood evaluated at the maximum likelihood estimate $\hat{\boldsymbol{\beta}}$, then $2(l_{\max} - l_j^*(\beta_j))$ has a limiting chi-square distribution with one degree of freedom if β_j is the true parameter value. Let $l_0 = l_{\max} - 0.5\chi_1^2(1-\alpha)$, where $\chi_1^2(1-\alpha)$ is the $100(1-\alpha)$ percentile of the chi-square distribution with one degree of freedom. A $100(1-\alpha)\%$ confidence interval for β_j is

$$\{\gamma : l_j^*(\gamma) \geq l_0\}$$

The endpoints of the confidence interval are found by solving numerically for values of β_j that satisfy equality in the preceding relation. To obtain an iterative algorithm for computing the confidence limits, the log-likelihood function in a neighborhood of $\boldsymbol{\beta}$ is approximated by the quadratic function

$$\tilde{l}(\boldsymbol{\beta}+\boldsymbol{\delta}) = l(\boldsymbol{\beta}) + \boldsymbol{\delta}'\mathbf{g} + \frac{1}{2}\boldsymbol{\delta}'\mathbf{V}\boldsymbol{\delta}$$

where $\mathbf{g} = \mathbf{g}(\boldsymbol{\beta})$ is the gradient vector and $\mathbf{V} = \mathbf{V}(\boldsymbol{\beta})$ is the Hessian matrix. The increment $\boldsymbol{\delta}$ for the next iteration is obtained by solving the likelihood equations

$$\frac{d}{d\boldsymbol{\delta}}\{\tilde{l}(\boldsymbol{\beta}+\boldsymbol{\delta}) + \lambda(\mathbf{e}_j'\boldsymbol{\delta} - \gamma)\} = \mathbf{0}$$

where λ is the Lagrange multiplier, $\mathbf{e}_j$ is the jth unit vector, and γ is an unknown constant. The solution is

$$\boldsymbol{\delta} = -\mathbf{V}^{-1}(\mathbf{g} + \lambda \mathbf{e}_j)$$

By substituting this $\boldsymbol{\delta}$ into the equation $\tilde{l}(\boldsymbol{\beta} + \boldsymbol{\delta}) = l_0$, you can estimate λ as

$$\lambda = \pm\left(\frac{2(l_0 - l(\boldsymbol{\beta}) + \frac{1}{2}\mathbf{g}'\mathbf{V}^{-1}\mathbf{g})}{\mathbf{e}_j'\mathbf{V}^{-1}\mathbf{e}_j}\right)^{\frac{1}{2}}$$

The upper confidence limit for β_j is computed by starting at the maximum likelihood estimate of $\boldsymbol{\beta}$ and iterating with positive values of λ until convergence is attained. The process is repeated for the lower confidence limit, using negative values of λ.

Convergence is controlled by value ϵ specified with the PLCONV= option in the MODEL statement (the default value of ϵ is 1E−4). Convergence is declared on the current iteration if the following two conditions are satisfied:

$$|l(\boldsymbol{\beta}) - l_0| \leq \epsilon$$

and

$$(\mathbf{g} + \lambda \mathbf{e}_j)'\mathbf{V}^{-1}(\mathbf{g} + \lambda \mathbf{e}_j) \leq \epsilon$$

The profile-likelihood confidence limits for the hazard ratio $\exp(\mathbf{e}_j'\boldsymbol{\beta})$ are obtained by exponentiating these confidence limits.

Testing Linear Hypotheses about Regression Coefficients

Linear hypotheses for $\boldsymbol{\beta}$ are expressed in matrix form as

$$H_0\colon \mathbf{L}\boldsymbol{\beta} = \mathbf{c}$$

where $\mathbf{L}$ is a matrix of coefficients for the linear hypotheses, and $\mathbf{c}$ is a vector of constants. The Wald chi-square statistic for testing H_0 is computed as

$$\chi^2_W = \left(\mathbf{L}\hat{\boldsymbol{\beta}} - \mathbf{c}\right)'\left[\mathbf{L}\hat{\mathbf{V}}(\hat{\boldsymbol{\beta}})\mathbf{L}'\right]^{-1}\left(\mathbf{L}\hat{\boldsymbol{\beta}} - \mathbf{c}\right)$$

where $\hat{\mathbf{V}}(\hat{\boldsymbol{\beta}})$ is the estimated covariance matrix. Under H_0, χ^2_W has an asymptotic chi-square distribution with r degrees of freedom, where r is the rank of $\mathbf{L}$.

Optimal Weights for the AVERAGE option in the TEST Statement

Let $\boldsymbol{\beta}_0 = (\beta_{i_1}, \ldots, \beta_{i_s})'$, where $\{\beta_{i_1}, \ldots, \beta_{i_s}\}$ is a subset of s regression coefficients. For any vector $\mathbf{e} = (e_1, \ldots, e_s)'$ of length s,

$$\mathbf{e}'\hat{\boldsymbol{\beta}}_0 \sim N(\mathbf{e}'\boldsymbol{\beta}_0, \mathbf{e}'\hat{\mathbf{V}}(\hat{\boldsymbol{\beta}}_0)\mathbf{e})$$

To find $\mathbf{e}$ such that $\mathbf{e}'\hat{\boldsymbol{\beta}}_0$ has the minimum variance, it is necessary to minimize $\mathbf{e}'\hat{\mathbf{V}}(\hat{\boldsymbol{\beta}}_0)\mathbf{e}$ subject to $\sum_{i=1}^{k} e_i = 1$. Let $\mathbf{1}_s$ be a vector of 1's of length s. The expression to be minimized is

$$\mathbf{e}'\hat{\mathbf{V}}(\hat{\boldsymbol{\beta}}_0)\mathbf{e} - \lambda(\mathbf{e}'\mathbf{1}_s - 1)$$

where λ is the Lagrange multipler. Differentiating with respect to $\mathbf{e}$ and λ, respectively, yields

$$\begin{aligned} \hat{\mathbf{V}}(\hat{\boldsymbol{\beta}}_0)\mathbf{e} - \lambda\mathbf{1}_s &= \mathbf{0} \\ \mathbf{e}'\mathbf{1}_s - 1 &= 0 \end{aligned}$$

Solving these equations gives

$$\mathbf{e} = [\mathbf{1}_s'\hat{\mathbf{V}}^{-1}(\hat{\boldsymbol{\beta}}_0)\mathbf{1}_s]^{-1}\hat{\mathbf{V}}^{-1}(\hat{\boldsymbol{\beta}}_0)\mathbf{1}_s$$

This provides a one degree-of-freedom test for testing the null hypothesis $H_0 : \beta_{i_1} = \ldots = \beta_{i_s} = 0$ with normal test statistic

$$Z = \frac{\mathbf{e}'\hat{\boldsymbol{\beta}}_0}{\sqrt{\mathbf{e}'\hat{\mathbf{V}}(\hat{\boldsymbol{\beta}}_0)\mathbf{e}}}$$

This test is more sensitive than the multivariate test specified by the TEST statement

```
Multivariate: test X1, ..., Xs;
```

where X1, ..., X*s* are the variables with regression coefficients $\beta_{i_1}, \ldots, \beta_{i_s}$, respectively.

Analysis of Multivariate Failure Time Data

Multivariate failure time data arise when each study subject can potentially experience several events (for instance, multiple infections after surgery) or when there exists some natural or artificial clustering of subjects (for instance, a litter of mice) that induces dependence among the failure times of the same cluster. Data in the former situation are referred to as multiple events data, and data in the latter situation are referred to as clustered data. The multiple events data can be further classified into ordered and unordered data. For ordered data, there is a natural ordering of the multiple failures within a subject, which includes recurrent events data as a special case. For unordered data, the multiple event times result from several concurrent failure processes.

Multiple events data can be analyzed by the Wei, Lin, and Weissfeld (1989), or WLW, method based on the marginal Cox models. For the special case of recurrent events data, you can fit the intensity model (Andersen and Gill 1982), the proportional rates/means model (Pepe and Cai 1993; Lawless and Nadeau 1995; Lin et al. 2000), or the stratified models for total time and gap time proposed by Prentice, Williams, and Peterson (1981), or PWP. For clustered data, you can carry out the analysis of Lee, Wei, and Amato (1992) based on the marginal Cox model. To use PROC PHREG to perform these analyses correctly and effectively, you have to array your data in a specific way to produce the correct risk sets.

All examples described in this section can be found in the program *phrmult.sas* in the SAS/STAT sample library. Furthermore, the "Examples" section in this chapter contains two examples to illustrate the methods of analyzing recurrent events data and clustered data.

Marginal Cox Models for Multiple Events Data

Suppose there are n subjects and each subject can experience up to K potential events. Let $\mathbf{Z}_{ki}(.)$ be the covariate process associated with the kth event for the ith subject. The marginal Cox models are given by

$$\lambda_k(t;\mathbf{Z}_{ki}) = \lambda_{k0}e^{\boldsymbol{\beta}_k'\mathbf{Z}_{ki}(t)}, k = 1,\ldots,K; i = 1,\ldots,n$$

where $\lambda_{k0}(t)$ is the (event-specific) baseline hazard function for the kth event and $\boldsymbol{\beta}_k$ is the (event-specific) column vector of regression coefficients for the kth event. WLW estimates $\boldsymbol{\beta}_1,\ldots,\boldsymbol{\beta}_K$ by the maximum partial likelihood estimates $\hat{\boldsymbol{\beta}}_1,\ldots,\hat{\boldsymbol{\beta}}_K$, respectively, and uses a robust sandwich covariance matrix estimate for $(\hat{\boldsymbol{\beta}}_1',\ldots,\hat{\boldsymbol{\beta}}_K')'$ to account for the dependence of the multiple failure times.

By using a properly prepared input data set, you can estimate the regression parameters for all the marginal Cox models and compute the robust sandwich covariance estimates in one PROC PHREG invocation. For convenience of discussion, suppose each subject can potentially experience K=3 events and there are two explanatory variables Z1 and Z2. The event-specific parameters to be estimated are $\boldsymbol{\beta}_1 = (\beta_{11},\beta_{21})'$ for the first marginal model, $\boldsymbol{\beta}_2 = (\beta_{12},\beta_{22})'$ for the second marginal model, and $\boldsymbol{\beta}_3 = (\beta_{13},\beta_{23})'$ for the third marginal model. Inference of these parameters is based on the robust sandwich covariance matrix estimate of the parameter estimators. It is necessary that each row of the input data set represent the data for a potential event of a subject. The input data set should contain the following:

- an ID variable for identifying the subject so that all observations of the same subject have the same ID value
- an Enum variable to index the multiple events. For example, Enum=1 for the first event, Enum=2 for the second event, and so on.
- a Time variable to represent the observed time from some time origin for the event. For recurrence events data, it is the time from the study entry to each recurrence.
- a Status variable to indicate whether the Time value is a censored or uncensored time. For example, Status=1 indicates an uncensored time and Status=0 indicates a censored time.
- independent variables (Z1 and Z2)

The WLW analysis can be carried out by specifying the following:

```
proc phreg covs(aggregate);
   model Time*Status(0)=Z11 Z12 Z13 Z21 Z22 Z23;
   strata Enum;
   id ID;
   Z11= Z1 * (Enum=1);
   Z12= Z1 * (Enum=2);
   Z13= Z1 * (Enum=3);
   Z21= Z2 * (Enum=1);
   Z22= Z2 * (Enum=2);
   Z23= Z2 * (Enum=3);
   run;
```

The variable Enum is specified in the STRATA statement so that there is one marginal Cox model for each distinct value of Enum. The variables Z11, Z12, Z13, Z21, Z22, and Z23 in the MODEL statement are event-specific variables derived from the independent variables Z1 and Z2 by the given programming statements. In particular, the variables Z11, Z12, and Z13 are event-specific variables for the explanatory variable Z1; the variables Z21, Z22, and Z23 are event-specific variables for the explanatory variable Z2. For $j = 1, 2$, and $k = 1, 2, 3$, variable Zjk contains the same values as the explanatory variable Zj for the rows that correspond to kth marginal model and the value 0 for all other rows; as such, β_{jk} is the regression coefficient for Zjk. You can avoid using the programming statements in PROC PHREG if you create these event-specific variables in the input data set by using the same programming statements in a DATA step.

The option COVS(AGGREGATE) is specified in the PROC statement to obtain the robust sandwich estimate of the covariance matrix, and the score residuals used in computing the middle part of the sandwich estimate are aggregated over identical ID values. You can also include TEST statements in the PROC PHREG code to test various linear hypotheses of the regression parameters based on the robust sandwich covariance matrix estimate.

Consider the AIDS study data in Wei, Lin, and Weissfeld (1989) from a randomized clinical trial to assess the antiretroviral capacity of ribavirin over time in AIDS patients. Blood samples were collected at weeks 4, 8, and 12 from each patient in three treatment groups (placebo, low dose of ribavirin, and high dose). For each serum sample, the failure time is the number of days before virus positivity was detected. If the sample was contaminated or it took a longer period of time than was achievable in the laboratory, the sample was censored. For example:

- Patient #1 in the placebo group has uncensored times 9, 6, and 7 days (that is, it took 9 days to detect viral positivity in the first blood sample, 6 days for the second blood sample, and 7 days for the third blood sample).
- Patient #14 in the low-dose group of ribavirin has uncensored times of 16 and 17 days for the first and second sample, respectively, and a censored time of 21 days for the third blood sample.
- Patient #28 in the high-dose group has an uncensored time of 21 days for the first sample, no measurement for the second blood sample, and a censored time of 25 days for the third sample.

For a full-rank parameterization, two design variables are sufficient to represent three treatment groups. Based on the reference coding with placebo as the reference, the values of the two dummy explanatory variables Z1 and Z2 representing the treatments are as follows:

Treatment Group	Z1	Z2
Placebo	0	0
Low dose ribavirin	1	0
High dose ribavirin	0	1

The bulk of the task in using PROC PHREG to perform the WLW analysis lies in the preparation of the input data set. As discussed earlier, the input data set should contain the ID, Enum, Time, and Status variables, and event-specific independent variables Z11, Z12, Z13, Z21, Z22, and Z23. Data for the three patients described earlier are arrayed as follows:

ID	Time	Status	Enum	Z1	Z2
1	9	1	1	0	0
1	6	1	2	0	0
1	7	1	3	0	0
14	16	1	1	1	0
14	17	1	2	1	0
14	21	0	3	1	0
28	21	1	1	0	1
28	25	0	3	0	1

The first three rows are data for Patient #1 with event times at 9, 6, and 7 days, one row for each event. The next three rows are data for Patient #14, who has an uncensored time of 16 days for the first serum sample, an uncensored time of 17 days for the second sample, and a censored time of 21 days for the third sample. The last two rows are data for Patient #28 of the high-dose group (Z1=0 and Z2=1). Since the patient did not have a second serum sample, there are only two rows of data.

To perform the WLW analysis, you specify the following statements:

```
proc phreg covs(aggregate);
   model Time*Status(0)=Z11 Z12 Z13 Z21 Z22 Z23;
   strata Enum;
   id ID;
   Z11= Z1 * (Enum=1);
   Z12= Z1 * (Enum=2);
   Z13= Z1 * (Enum=3);
   Z21= Z2 * (Enum=1);
   Z22= Z2 * (Enum=2);
   Z23= Z2 * (Enum=3);
   EqualLowDose: test Z11=Z12, Z12=Z23;
   AverageLow: test Z11,Z12,Z13 / e average;
run;
```

Two linear hypotheses are tested using the TEST statements. The specification

```
   EqualLowDose: test Z11=Z12, Z12=Z13;
```

tests the null hypothesis $\beta_{11} = \beta_{12} = \beta_{13}$ of identical low-dose effects across three marginal models. The specification

```
AverageLow: test Z11,Z12,Z13 / e average;
```

tests the null hypothesis of no low-dose effects (that is, $\beta_{11} = \beta_{12} = \beta_{13} = 0$). The AVERAGE option computes the optimal weights for estimating the average low-dose effect $\beta_1^* = \beta_{11} = \beta_{12} = \beta_{13}$ and performs a 1 DF test for testing the null hypothesis that $\beta_1^* = 0$. The E option displays the coefficients for the linear hypotheses, including the optimal weights.

Marginal Cox Models for Clustered Data

Suppose there are n clusters with K_i members in the ith cluster, $i = 1, \ldots, n$. Let $\mathbf{Z}_{ki}(.)$ be the covariate process associated with the kth member of the ith cluster. The marginal Cox model is

given by

$$\lambda(t;\mathbf{Z}_{ki}) = \lambda_0(t)e^{\boldsymbol{\beta}'\mathbf{Z}_{ki}(t)} k = 1,\ldots,K_i; i = 1,\ldots,n$$

where $\lambda_0(t)$ is an arbitrary baseline hazard function and $\boldsymbol{\beta}$ is the vector of regression coefficients. Lee, Wei, and Amato (1992) estimate $\boldsymbol{\beta}$ by the maximum partial likelihood estimate $\hat{\boldsymbol{\beta}}$ under the independent working assumption, and use a robust sandwich covariance estimate to account for the intracluster dependence.

To use PROC PHREG to analyze the clustered data, each member of a cluster is represented by an observation in the input data set. The input data set to PROC PHREG should contain the following:

- an ID variable to identify the cluster so that members of the same cluster have the same ID value
- a Time variable to represent the observed survival time of a member of a cluster
- a Status variable to indicate whether the Time value is an uncensored or censored time. For example, Status=1 indicates an uncensored time and Status=0 indicates a censored time.
- the explanatory variables thought to be related to the failure time

Consider a tumor study in which one of three female rats of the same litter was randomly subjected to a drug treatment. The failure time is the time from randomization to the detection of tumor. If a rat died before the tumor was detected, the failure time was censored. For example:

- In litter #1, the drug-treated rat has an uncensored time of 101 days, one untreated rat has a censored time of 49 days, and the other untreated rat has a failure time of 104 days.
- In litter #3, the drug-treated rat has a censored time of 104 days, one untreated rat has a censored time of 102 days, and the other untreated rat has a censored time of 104 days.

In this example, a litter is a cluster and the rats of the same litter are members of the cluster. Let Trt be a 0-1 variable representing the treatment a rat received, with value 1 for drug treatment and 0 otherwise. Data for the two litters of rats described earlier contribute six observations to the input data set:

Litter	Time	Status	Trt
1	101	1	1
1	49	0	0
1	104	1	0
3	104	0	1
3	102	0	0
3	104	0	0

The analysis of Lee, Wei, and Amato (1992) can be performed by PROC PHREG as follows:

```
proc phreg covs(aggregate);
   model Time*Status(0)=Treatment;
   id Litter;
run;
```

Intensity and Rate/Mean Models for Recurrent Events Data

Suppose each subject experiences recurrences of the same phenomenon. Let $N(t)$ be the number of events a subject experiences over the interval $[0,t]$ and let $\mathbf{Z}(.)$ be the covariate process of the subject.

The intensity model (Andersen and Gill 1982) is given by

$$\lambda_{\mathbf{Z}}(t)dt = E\{dN(t)|\mathcal{F}_{t-}\} = \lambda_0(t)e^{\boldsymbol{\beta}'\mathbf{Z}(t)}dt$$

where $\mathcal{F}_t$ represents all the information of the processes N and $\mathbf{Z}$ up to time t, $\lambda_0(t)$ is an arbitrary baseline intensity function, and $\boldsymbol{\beta}$ is the vector of regression coefficients. This model consists of two components: (1) all the influence of the prior events on future recurrences, if there is any, is mediated through the time-dependent covariates, and (2) the covariates have multiplicative effects on the instantaneous rate of the counting process. If the covariates are time invariant, the risk of recurrences is unaffected by the past events.

The proportional rates and means models (Pepe and Cai 1993; Lawless and Nadeau 1995; Lin et al. 2000) assume that the covariates have multiplicative effects on the mean and rate functions of the counting process. The rate function is given by

$$d\mu_{\mathbf{Z}}(t) = E\{dN(t)|\mathbf{Z}(t)\} = e^{\boldsymbol{\beta}'\mathbf{Z}(t)}d\mu_0(t)$$

where $\mu_0(t)$ is an unknown continuous function and $\boldsymbol{\beta}$ is the vector of regression parameters. If $\mathbf{Z}$ is time invariant, the mean function is given by

$$\mu_{\mathbf{Z}}(t) = E\{N(t)|\mathbf{Z}\} = e^{\boldsymbol{\beta}'\mathbf{Z}}\mu_0(t)$$

For both the intensity and the proportional rates/means models, estimates of the regression coefficients are obtained by solving the partial likelihood score function. However, the covariance matrix estimate for the intensity model is computed as the inverse of the observed information matrix, while that for the proportional rates/means model is given by a sandwich estimate. For a given pattern of fixed covariates, the Nelson estimate for the cumulative intensity function is the same for the cumulative mean function, but their standard errors are not the same.

To fit the intensity or rate/mean model by using PROC PHREG, the counting process style of input is needed. A subject with K events contributes K+1 observations to the input data set. The kth observation of the subject identifies the time interval from the $(k-1)$th event or time 0 (if $k = 1$) to the kth event, $k = 1, \ldots, K$. The $(K+1)$th observation represents the time interval from the Kth event to time of censorship. The input data set should contain the following variables:

- a TStart variable to represent the $(k-1)$th recurrence time or the value 0 if $k = 1$
- a TStop variable to represent the kth recurrence time or the follow-up time if $k = K+1$
- a Status variable indicating whether the TStop time is a recurrence time or a censored time; for example, Status=1 for a recurrence time and Status=0 for censored time
- explanatory variables thought to be related to the recurrence times

If the rate/mean model is used, the input data should also contain an ID variable for identifying the subjects.

Consider the chronic granulomatous disease (CGD) data listed in Fleming and Harrington (1991). The disease is a rare disorder characterized by recurrent pyrogenic infections. The study is a placebo-controlled randomized clinical trial conducted by the International CGD Cooperative Study to assess the effect of gamma interferon to reduce the rate of infection. For each study patient the times of recurrent infections along with a number of prognostic factors were collected. For example:

- Patient #17404, age 38, in the gamma interferon group had a follow-up time of 293 without any infection.
- Patient #204001, age 12, in the placebo group had an infection at 219 days, a recurrent infection at 373 days, and was followed up to 414 days.

Let Trt be the variable representing the treatment status with value 1 for gamma interferon and value 2 for placebo. Let Age be a covariate representing the age of the CGD patient. Data for the two CGD patients described earlier are given in the following table.

ID	TStart	TStop	Status	Trt	Age
174054	0	293	0	1	38
204001	0	219	1	2	12
204001	219	373	1	2	12
204001	373	414	0	2	12

Since Patient #174054 had no infection through the end of the follow-up period (293 days), there is only one observation representing the period from time 0 to the end of the follow-up. Data for Patient #204001 are broken into three observations, since there are two infections. The first observation represents the period from time 0 to the first infection, the second observation represents the period from the first infection to the second infection, and the third time period represents the period from the second infection to the end of the follow-up.

The following specification fits the intensity model:

```
proc phreg;
   model (TStart,TStop)*Status(0)=Trt Age;
   run;
```

You can predict the cumulative intensity function for a given pattern of fixed covariates by specifying the CUMHAZ= option in the BASELINE statement. Suppose you are interested in two fixed patterns, one for patients of age 30 in the gamma interferon group and the other for patients of age 1 in the placebo group. You first create the SAS data set as follows:

```
data Pattern;
   Trt=1; Age=30;
   output;
   Trt=2; Age=1;
   output;
   run;
```

You then include the following BASELINE statement in the PROC PHREG specification. The CUMHAZ=_all_ option produces the cumulative hazard function estimates, the standard error estimates, and the lower and upper pointwise confidence limits.

```
baseline covariates=Pattern out=out1 cumhaz=_all_;
```

The following specification of PROC PHREG fits the mean model and predicts the cumulative mean function for the two patterns of covariates in the Pattern data set:

```
proc phreg covs(aggregate);
   model (Tstart,Tstop)*Status(0)=Trt Age;
   baseline covariates=Pattern out=out2 cmf=_all_;
   id ID;
```

The COV(AGGREGATE) option, along with the ID statement, computes the robust sandwich covariance matrix estimate. The CMF=_ALL_ option adds the cumulative mean function estimates, the standard error estimates, and the lower and upper pointwise confidence limits to the OUT=Out2 data set.

PWP Models for Recurrent Events Data

Let $N(t)$ be the number of events a subject experiences by time t. Let $\mathbf{Z}(t)$ be the covariate vectors of the subject at time t. For a subject who has K events before censorship takes place, let $t_0 = 0$, let t_k be the kth recurrence time, $k = 1, \ldots, K$, and let t_{K+1} be the censored time. Prentice, Williams, and Peterson (1981) consider two time scales, a total time from the beginning of the study and a gap time from immediately preceding failure. The PWP models are stratified Cox-type models that allow the shape of the hazard function to depend on the number of preceding events and possibly on other characteristics of $\{N(t)$ and $\mathbf{Z}(t)\}$. The total time and gap time models are given, respectively, as follows:

$$\lambda(t|\mathcal{F}_{t-}) = \lambda_{0k}(t)e^{\boldsymbol{\beta}_k'\mathbf{Z}(t)}, \qquad t_{k-1} < t \le t_k$$
$$\lambda(t|\mathcal{F}_{t-}) = \lambda_{0k}(t - t_{k-1})e^{\boldsymbol{\beta}_k'\mathbf{Z}(t)}, \qquad t_{k-1} < t \le t_k$$

where λ_{0k} is an arbitrary baseline intensity functions, and $\boldsymbol{\beta}_k$ is a vector of stratum-specific regression coefficients. Here, a subject moves to the kth stratum immediately after his $(k-1)$th recurrence time and remains there until the kth recurrence occurs or until censorship takes place. For instance, a subject who experiences only one event moves from the first stratum to the second stratum after the event occurs and remains in the second stratum until the end of the follow-up.

You can use PROC PHREG to carry out the analyses of the PWP models, but you have to prepare the input data set to provide the correct risk sets. The input data set for analyzing the total time is the same as the AG model with an additional variable to represent the stratum that the subject is in. A subject with K events contributes $K+1$ observations to the input data set, one for each stratum that the subject moves to. The input data should contain the following variables:

- a TStart variable to represent the $(k-1)$th recurrence time or the value 0 if $k = 1$
- a TStop variable to represent the kth recurrence time or the time of censorship if $k = K + 1$
- a Status variable with value 1 if the Time value is a recurrence time and value 0 if the Time value is a censored time

- an Enum variable representing the index of the stratum that the subject is in. For a subject who has only one event at t_1 and is followed to time t_c, Enum=1 for the first observation (where Time=t_1 and Status=1) and Enum=2 for the second observation (where Time=t_c and Status=0).
- explanatory variables thought to be related to the recurrence times

To analyze gap times, the input data set should also include a GapTime variable that is equal to (TStop − TStart).

Consider the data of two subjects in CGD data described in the previous section:

- Patients #174054, age 38, in the gamma interferon group had a follow-up time of 293 without any infection.
- Patient #204001, age 12, in the placebo group had an infection at 219 days, a recurrent infection at 373 days, and a follow-up time of 414 days.

To illustrate, suppose all subjects have at most two observed events. The data for the two subjects in the input data set are as follows:

ID	TStart	TStop	Gaptime	Status	Enum	Trt	Age
174054	0	293	293	0	1	1	38
204001	0	219	219	1	1	2	12
204001	219	373	154	1	2	2	12
204001	373	414	41	0	3	2	12

Subject #174054 contributes only one observation to the input data, since there is no observed event. Subject #204001 contributes three observations, since there are two observed events.

To fit the total time model of PWP with stratum-specific slopes, either you can create the stratum-specific explanatory variables (Trt1, Trt2, and Trt3 for Trt, and Age1, Age2, and Age3 for Age) in a DATA step, or you can specify them in PROC PHREG by using programming statements as follows:

```
proc phreg;
   model (TStart,TStop)*Status(0)=Trt1 Trt2 Trt3 Age1 Age2 Age3;
   strata Enum;
   Trt1= Trt * (Enum=1);
   Trt2= Trt * (Enum=2);
   Trt3= Trt * (Enum=3);
   Age1= Age * (Enum=1);
   Age2= Age * (Enum=2);
   Age3= Age * (Enum=3);
   run;
```

To fit the total time model of PWP with the common regression coefficients, you specify the following:

```
proc phreg;
   model (TStart,TStop)*Status(0)=Trt Age;
   strata Enum;
   run;
```

To fit the gap time model of PWP with stratum-specific regression coefficients, you specify the following:

```
proc phreg;
   model Gaptime*Status(0)=Trt1 Trt2 Trt3 Age1 Age2 Age3;
   strata Enum;
   Trt1= Trt * (Enum=1);
   Trt2= Trt * (Enum=2);
   Trt3= Trt * (Enum=3);
   Age1= Age * (Enum=1);
   Age2= Age * (Enum=2);
   Age3= Age * (Enum=3);
run;
```

To fit the gap time model of PWP with common regression coefficients, you specify the following:

```
proc phreg;
   model Gaptime*Status(0)=Trt Age;
   strata Enum;
   run;
```

Model Fit Statistics

Suppose the model contains p regression parameters. Let Δ_j and f_j be the event indicator and the frequency, respectively, of the jth observation. The three criteria displayed by the PHREG procedure are calculated as follows:

- -2 Log Likelihood:

  $$-2 \text{ Log L} = -2\log(L_n(\hat{\boldsymbol{\beta}}))$$

  where $L_n(.)$ is a partial likelihood function for the corresponding TIES= option as described in the section "Partial Likelihood Function for the Cox Model" on page 4587, and $\hat{\boldsymbol{\beta}}$ is the maximum likelihood estimate of the regression parameter vector.

- Akaike Information Criterion:

  $$\text{AIC} = -2 \text{ Log L} + 2p$$

- Schwarz Bayesian (Information) Criterion:

  $$\text{SBC} = -2 \text{ Log L} + p\log(\sum_j f_j \Delta_j)$$

The -2 Log Likelihood statistic has a chi-square distribution under the null hypothesis (that all the explanatory effects in the model are zero) and the procedure produces a p-value for this statistic.

The AIC and SBC statistics give two different ways of adjusting the −2 Log Likelihood statistic for the number of terms in the model and the number of observations used. These statistics should be used when comparing different models for the same data (for example, when you use the METHOD=STEPWISE option in the MODEL statement); lower values of the statistic indicate a more desirable model.

Residuals

This section describes the computation of residuals (RESMART=, RESDEV=, RESSCH=, and RESSCO=) in the OUTPUT statement.

First, consider TIES=BRESLOW. Let

$$\begin{aligned} S^{(0)}(\boldsymbol{\beta},t) &= \sum_i Y_i(t)\mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_i(t)} \\ S^{(1)}(\boldsymbol{\beta},t) &= \sum_i Y_i(t)\mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_i(t)}\mathbf{Z}_i(t) \\ \bar{\mathbf{Z}}(\boldsymbol{\beta},t) &= \frac{S^{(1)}(\boldsymbol{\beta},t)}{S^{(0)}(\boldsymbol{\beta},t)} \\ d\Lambda_0(\boldsymbol{\beta},t) &= \sum_i \frac{dN_i(t)}{S^{(0)}(\boldsymbol{\beta},t)} \\ dM_i(\boldsymbol{\beta},t) &= dN_i(t) - Y_i(t)\mathrm{e}^{\boldsymbol{\beta}'\mathbf{Z}_i(t)}d\Lambda_0(\boldsymbol{\beta},t) \end{aligned}$$

The martingale residual at t is defined as

$$\hat{M}_i(t) = \int_0^t dM_i(\hat{\boldsymbol{\beta}},s) = N_i(t) - \int_0^t Y_i(s)\mathrm{e}^{\hat{\boldsymbol{\beta}}'\mathbf{Z}_i(s)}d\Lambda_0(\hat{\boldsymbol{\beta}},s)$$

Here $\hat{M}_i(t)$ estimates the difference over $(0,t]$ between the observed number of events for the ith subject and a conditional expected number of events. The quantity $\hat{M}_i \equiv \hat{M}_i(\infty)$ is referred to as the martingale residual for the ith subject. When the counting process MODEL specification is used, the RESMART= variable contains the component $(\hat{M}_i(t_2) - \hat{M}_i(t_1))$ instead of the martingale residual at t_2. The martingale residual for a subject can be obtained by summing up these component residuals within the subject. For the Cox model with no time-dependent explanatory variables, the martingale residual for the ith subject with observation time t_i and event status Δ_i is

$$\hat{M}_i = \Delta_i - \mathrm{e}^{\hat{\boldsymbol{\beta}}'\mathbf{Z}_i}\int_0^{t_i} d\Lambda_0(\hat{\boldsymbol{\beta}},s)$$

The deviance residuals D_i are a transform of the martingale residuals:

$$D_i = \mathrm{sign}(\hat{M}_i)\sqrt{2\left[-\hat{M}_i - N_i(\infty)\log\left(\frac{N_i(\infty)-\hat{M}_i}{N_i(\infty)}\right)\right]}$$

The square root shrinks large negative martingale residuals, while the logarithmic transformation expands martingale residuals that are close to unity. As such, the deviance residuals are more symmetrically distributed around zero than the martingale residuals. For the Cox model, the deviance residual reduces to the form

$$D_i = \text{sign}(\hat{M}_i)\sqrt{2[-\hat{M}_i - \Delta_i \log(\Delta_i - \hat{M}_i)]}$$

When the counting process MODEL specification is used, values of the RESDEV= variable are set to missing because the deviance residuals can be calculated only on a per-subject basis.

The Schoenfeld (1982) residual vector is calculated on a per-event-time basis. At the jth event time t_{i_j} of the ith subject, the Schoenfeld residual

$$\hat{\mathbf{U}}_i(t_{i_j}) = \mathbf{Z}_i(t_{i_j}) - \bar{\mathbf{Z}}(\hat{\boldsymbol{\beta}}, t_{i_j})$$

is the difference between the ith subject covariate vector at t_{i_j} and the average of the covariate vectors over the risk set at t_{i_j}. Under the proportional hazards assumption, the Schoenfeld residuals have the sample path of a random walk; therefore, they are useful in assessing time trend or lack of proportionality. Harrell (1886) proposed a z-transform of the Pearson correlation between these residuals and the rank order of the failure time as a test statistic for nonproportional hazards. Therneau, Grambsch, and Fleming (1990) considered a Kolmogorov-type test based on the cumulative sum of the residuals.

The score process for the ith subject at time t is

$$\mathbf{L}_i(\boldsymbol{\beta}, t) = \int_0^t [\mathbf{Z}_i(s) - \bar{\mathbf{Z}}(\boldsymbol{\beta}, s)] d M_i(\boldsymbol{\beta}, s)$$

The vector $\hat{\mathbf{L}}_i \equiv \mathbf{L}_i(\hat{\boldsymbol{\beta}}, \infty)$ is the score residual for the ith subject. When the counting process MODEL specification is used, the RESSCO= variables contain the components of $(\mathbf{L}_i(\hat{\boldsymbol{\beta}}, t_2) - \mathbf{L}_i(\hat{\boldsymbol{\beta}}, t_1))$ instead of the score process at t_2. The score residual for a subject can be obtained by summing up these component residuals within the subject.

The score residuals are a decomposition of the first partial derivative of the log likelihood. They are useful in assessing the influence of each subject on individual parameter estimates. They also play an important role in the computation of the robust sandwich variance estimators of Lin and Wei (1989) and Wei, Lin, and Weissfeld (1989).

For TIES=EFRON, the preceding computation is modified to comply with the Efron partial likelihood. Consider an uncensored time t. For a given time t, let $\Delta_i(t)$=1 if the t is an event time of the ith subject and 0 otherwise. Let $d(t) = \sum_i \Delta_i(t)$, which is the number of subjects that have an

event at t. For $1 \leq k \leq d(t)$, let

$$\begin{aligned}
S^{(0)}(\boldsymbol{\beta}, k, t) &= \sum_i Y_i(t)\left\{1 - \frac{k-1}{d(t)}\Delta_i(t)\right\} e^{\boldsymbol{\beta}'\mathbf{Z}_i(t)} \\
S^{(1)}(\boldsymbol{\beta}, k, t) &= \sum_i Y_i(t)\left\{1 - \frac{k-1}{d(t)}\Delta_i(t)\right\} e^{\boldsymbol{\beta}'\mathbf{Z}_i(t)}\mathbf{Z}_i(t) \\
\bar{\mathbf{Z}}(\boldsymbol{\beta}, k, t) &= \frac{S^{(1)}(\boldsymbol{\beta}, k, t)}{S^{(0)}(\boldsymbol{\beta}, k, t)} \\
d\Lambda_0(\boldsymbol{\beta}, k, t) &= \sum_i \frac{dN_i(t)}{S^{(0)}(\boldsymbol{\beta}, k, t)} \\
dM_i(\boldsymbol{\beta}, k, t) &= dN_i(t) - Y_i(t)\left(1 - \Delta_i(t)\frac{k-1}{d(t)}\right) e^{\boldsymbol{\beta}'\mathbf{Z}_i(t)} d\Lambda_0(\boldsymbol{\beta}, k, t)
\end{aligned}$$

The martingale residual at t for the ith subject is defined as

$$\hat{M}_i(t) = \int_0^t \frac{1}{d(s)} \sum_{k=1}^{d(s)} dM_i(\hat{\boldsymbol{\beta}}, k, s) = N_i(t) - \int_0^t \frac{1}{d(s)} \sum_{k=1}^{d(s)} Y_i(s)\left(1 - \Delta_i(s)\frac{k-1}{d(s)}\right) e^{\hat{\boldsymbol{\beta}}'\mathbf{Z}_i(s)} d\Lambda_0(\hat{\boldsymbol{\beta}}, k, s)$$

Deviance residuals are computed by using the same transform on the corresponding martingle residuals as in TIES=BRESLOW.

The Schoenfeld residual vector for the ith subject at event time t_{i_j} is

$$\hat{\mathbf{U}}_i(t_{i_j}) = \mathbf{Z}_i(t_{i_j}) - \frac{1}{d(t_{i_j})} \sum_{k=1}^{d(t_{i_j})} \bar{\mathbf{Z}}(\hat{\boldsymbol{\beta}}, k, t_{i_j})$$

The score process for the ith subject at time t is given by

$$\mathbf{L}_i(\boldsymbol{\beta}, t) = \int_0^t \frac{1}{d(s)} \sum_{k=1}^{d(s)} \left(\mathbf{Z}_i(s) - \bar{\mathbf{Z}}(\boldsymbol{\beta}, k, s)\right) dM_i(\boldsymbol{\beta}, k, s)$$

For TIES=DISCRETE or TIES=EXACT, it is difficult to come up with modifications that are consistent with the correponding partial likelihood. Residuals for these TIES= methods are computed by using the same formulae as in TIES=BRESLOW.

Diagnostics Based on Weighted Residuals

The vector of weighted Schoenfeld residuals, $\mathbf{r}_i$, is computed as

$$\mathbf{r}_i = n_e \mathcal{I}^{-1}(\hat{\boldsymbol{\beta}}) \hat{\mathbf{U}}_i(t_i)$$

where n_e is the total number of events and $\hat{\mathbf{U}}_i(t_i)$ is the vector of Schoenfeld residuals at the event time t_i. The components of $\mathbf{r}_i$ are output to the WTRESSCH= variables.

The weighted Schoenfeld residuals are useful in assessing the proportional hazards assumption. The idea is that most of the common alternatives to the proportional hazards can be cast in terms of a time-varying coefficient model

$$\lambda(t, \mathbf{Z}) = \lambda_0(t) \exp(\beta_1(t)Z_1 + \beta_2(t)Z_2 + \ldots)$$

where $\lambda(t, \mathbf{Z})$ and $\lambda_0(t)$ are hazard rates. Let $\hat{\beta}_j$ and r_{ij} be the jth component of $\hat{\boldsymbol{\beta}}$ and $\mathbf{r}_i$, respectively. Grambsch and Therneau (1994) suggest using a smoothed plot of $(\hat{\beta}_j + r_{ij})$ versus t_i to discover the functional form of the time-varying coefficient $\beta_j(t)$. A zero slope indicates that the coefficient is not varying with time.

The weighted score residuals are used more often than their unscaled counterparts in assessing local influence. Let $\hat{\boldsymbol{\beta}}_{(i)}$ be the estimate of $\boldsymbol{\beta}$ when the ith subject is left out, and let $\delta\hat{\boldsymbol{\beta}}_i = \hat{\boldsymbol{\beta}} - \hat{\boldsymbol{\beta}}_{(i)}$. The jth component of $\delta\hat{\boldsymbol{\beta}}_i$ can be used to assess any untoward effect of the ith subject on $\hat{\beta}_j$. The exact computation of $\delta\hat{\boldsymbol{\beta}}_i$ involves refitting the model each time a subject is omitted. Cain and Lange (1984) derived the following approximation of $\boldsymbol{\Delta}_i$ as weighted score residuals:

$$\delta\hat{\boldsymbol{\beta}}_i = \mathcal{I}^{-1}(\hat{\boldsymbol{\beta}})\hat{\mathbf{L}}_i$$

Here, $\hat{\mathbf{L}}_i$ is the vector of the score residuals for the ith subject. Values of $\delta\hat{\boldsymbol{\beta}}_i$ are output to the DFBETA= variables. Again, when the counting process MODEL specification is used, the DFBETA= variables contain the component $\mathcal{I}^{-1}(\hat{\boldsymbol{\beta}})(\mathbf{L}_i(\hat{\boldsymbol{\beta}}, t_2) - \mathbf{L}_i(\hat{\boldsymbol{\beta}}, t_1))$, where the score process $\mathbf{L}_i(\boldsymbol{\beta}, t)$ is defined in the section "Residuals" on page 4611. The vector $\delta\hat{\boldsymbol{\beta}}_i$ for the ith subject can be obtained by summing these components within the subject.

Note that these DFBETA statistics are a transform of the score residuals. In computing the robust sandwich variance estimators of Lin and Wei (1989) and Wei, Lin, and Weissfeld (1989), it is more convenient to use the DFBETA statistics than the score residuals (see Example 64.10).

Influence of Observations on Overall Fit of the Model

The LD statistic approximates the likelihood displacement, which is the amount by which minus twice the log likelihood ($-2\log L(\hat{\boldsymbol{\beta}})$), under a fitted model, changes when each subject in turn is left out. When the ith subject is omitted, the likelihood displacement is

$$2\log L(\hat{\boldsymbol{\beta}}) - 2\log L(\hat{\boldsymbol{\beta}}_{(i)})$$

where $\hat{\boldsymbol{\beta}}_{(i)}$ is the vector of parameter estimates obtained by fitting the model without the ith subject. Instead of refitting the model without the ith subject, Pettitt and Bin Daud (1989) propose that the likelihood displacement for the ith subject be approximated by

$$LD_i = \hat{\mathbf{L}}_i'\mathcal{I}^{-1}(\hat{\boldsymbol{\beta}})\hat{\mathbf{L}}_i$$

wher $\hat{\mathbf{L}}_i$ is the score residual vector of the ith subject. This approximation is output to the LD= variable.

The LMAX statistic is another global influence statistic. This statistic is based on the symmetric matrix

$$\mathbf{B} = \mathbf{L}\mathcal{I}^{-1}(\hat{\boldsymbol{\beta}})\mathbf{L}'$$

where $\mathbf{L}$ is the matrix with rows that are the score residual vectors $\hat{\mathbf{L}}_i$. The elements of the eigenvector associated with the largest eigenvalue of the matrix $\mathbf{B}$, standardized to unit length, give a measure of the sensitivity of the fit of the model to each observation in the data. The influence of the ith subject on the global fit of the model is proportional to the magnitude of ζ_i, where ζ_i is the ith element of the vector $\boldsymbol{\zeta}$ that satisfies

$$\mathbf{B}\boldsymbol{\zeta} = \lambda_{\max}\boldsymbol{\zeta} \quad \text{and} \quad \boldsymbol{\zeta}'\boldsymbol{\zeta} = 1$$

with $\lambda_{\max}$ being the largest eigenvalue of $\mathbf{B}$. The sign of ζ_i is irrelevant, and its absolute value is output to the LMAX= variable.

When the counting process MODEL specification is used, the LD= and LMAX= variables are set to missing, because these two global influence statistics can be calculated on a per-subject basis only.

Survivor Function Estimation for the Cox Model

Two estimators of the survivor function are available: one is the product-limit estimator (Kalbfleisch and Prentice 1980, pp. 84–86) and the other is the Breslow (1972) estimator based on the empirical cumulative hazard function.

Product-Limit Estimates

Let $\mathcal{C}_i$ denote the set of individuals censored in the half-open interval $[t_i, t_{i+1})$, where $t_0 = 0$ and $t_{k+1} = \infty$. Let γ_l denote the censoring times in $[t_i, t_{i+1})$; l ranges over $\mathcal{C}_i$.

The likelihood function for all individuals is given by

$$\mathcal{L} = \prod_{i=0}^{k} \left\{ \prod_{l \in \mathcal{D}_i} \left([S_0(t_i)]^{\exp(\mathbf{z}_l'\boldsymbol{\beta})} - [S_0(t_i + 0)]^{\exp(\mathbf{z}_l'\boldsymbol{\beta})} \right) \prod_{l \in \mathcal{C}_i} [S_0(\gamma_l + 0)]^{\exp(\mathbf{z}_l'\boldsymbol{\beta})} \right\}$$

where $\mathcal{D}_0$ is empty. The likelihood $\mathcal{L}$ is maximized by taking $S_0(t) = S_0(t_i + 0)$ for $t_i < t \leq t_{i+1}$ and allowing the probability mass to fall only on the observed event times $t_1, \ldots, t_k$. By considering a discrete model with hazard contribution $1 - \alpha_i$ at t_i, you take $S_0(t_i) = S_0(t_{i-1} + 0) = \prod_{j=0}^{i-1} \alpha_j$, where $\alpha_0 = 1$. Substitution into the likelihood function produces

$$\mathcal{L} = \prod_{i=0}^{k} \left\{ \prod_{j \in \mathcal{D}_i} \left(1 - \alpha_i^{\exp(\mathbf{z}_j'\boldsymbol{\beta})} \right) \prod_{l \in \mathcal{R}_i - \mathcal{D}_i} \alpha_i^{\exp(\mathbf{z}_l'\boldsymbol{\beta})} \right\}$$

If you replace $\boldsymbol{\beta}$ with $\hat{\boldsymbol{\beta}}$ estimated from the partial likelihood function and then maximize with respect to $\alpha_1, \ldots, \alpha_k$, the maximum likelihood estimate $\hat{\alpha}_i$ of α_i becomes a solution of

$$\sum_{j \in \mathcal{D}_i} \frac{\exp(\mathbf{z}_j'\hat{\boldsymbol{\beta}})}{1 - \hat{\alpha}_i^{\exp(\mathbf{z}_j'\hat{\boldsymbol{\beta}})}} = \sum_{l \in \mathcal{R}_i} \exp(\mathbf{z}_l'\hat{\boldsymbol{\beta}})$$

When only a single failure occurs at t_i, $\hat{\alpha}_i$ can be found explicitly. Otherwise, an iterative solution is obtained by the Newton method.

The estimated baseline cumulative hazard function is

$$\hat{H}_0(t) = -\log(\hat{S}_0(t))$$

where $\hat{S}_0(t)$ is the estimated baseline survivor function given by

$$\hat{S}_0(t) = \hat{S}_0(t_{i-1} + 0) = \prod_{j=0}^{i-1} \hat{\alpha}_j, t_{i-1} < t \leq t_i$$

For details, refer to Kalbfleisch and Prentice (1980). For a given realization of the explanatory variables $\boldsymbol{\xi}$, the product-limit estimate of the survival function at $\mathbf{Z} = \boldsymbol{\xi}$ is

$$\hat{S}(t, \boldsymbol{\xi}) = [\hat{S}_0(t)]^{\exp(\boldsymbol{\beta}'\boldsymbol{\xi})}$$

Empirical Cumulative Hazards Function Estimates

Let $\boldsymbol{\xi}$ be a given realization of the explanatory variables. The empirical cumulative hazard function estimate at $\mathbf{Z} = \boldsymbol{\xi}$ is

$$\hat{\Lambda}(t, \boldsymbol{\xi}) = \sum_{i=1}^{n} \int_0^t \frac{dN_i(s)}{\sum_{j=1}^{n} Y_j(s) \exp(\hat{\boldsymbol{\beta}}'(\mathbf{z}_j - \boldsymbol{\xi}))}$$

The variance estimator of $\hat{\Lambda}(t, \boldsymbol{\xi})$ is given by the following (Tsiatis 1981):

$$\begin{aligned} & \hat{var}\{n^{\frac{1}{2}}(\hat{\Lambda}(t, \boldsymbol{\xi}) - \Lambda(t, \boldsymbol{\xi}))\} \\ = \; & n\left\{\sum_{i=1}^{n} \int_0^t \frac{dN_i(s)}{[\sum_{j=1}^{n} Y_j(s) \exp(\hat{\boldsymbol{\beta}}'(\mathbf{z}_j - \boldsymbol{\xi}))]^2} + \mathbf{H}'(t, \boldsymbol{\xi}) \hat{\mathbf{V}}(\hat{\boldsymbol{\beta}}) \mathbf{H}(t, \boldsymbol{\xi})\right\} \end{aligned}$$

where $\hat{\mathbf{V}}(\hat{\boldsymbol{\beta}})$ is the estimated covariance matrix of $\hat{\boldsymbol{\beta}}$ and

$$\mathbf{H}(t, \boldsymbol{\xi}) = \sum_{i=1}^{n} \int_0^t \frac{\sum_{l=1}^{n} Y_l(s)(\mathbf{Z}_l - \boldsymbol{\xi}) \exp(\hat{\boldsymbol{\beta}}'(\mathbf{z}_l - \boldsymbol{\xi}))}{[\sum_{j=1}^{n} Y_j(s) \exp(\hat{\boldsymbol{\beta}}'(\mathbf{z}_j - \boldsymbol{\xi}))]^2} dN_i(s)$$

For the marginal model, the variance estimator computation follows Spiekerman and Lin (1998).

The empirical cumulative hazard function (CH) estimate of the survivor function for $\mathbf{Z} = \boldsymbol{\xi}$ is

$$\tilde{S}(t, \boldsymbol{\xi}) = \exp(-\hat{\Lambda}(t, \boldsymbol{\xi}))$$

Confidence Intervals for the Survivor Function

Let $\hat{S}(t, \boldsymbol{\xi})$ and $\tilde{S}(t, \boldsymbol{\xi})$ correspond to the product-limit (PL) and empirical cumulative hazard function (CH) estimates of the survivor function for $\mathbf{Z} = \boldsymbol{\xi}$, respectively. Both the standard error of $\log(\hat{S}(t, \boldsymbol{\xi}))$ and the standard error of $\log(\tilde{S}(t, \boldsymbol{\xi}))$ are approximated by $\tilde{\sigma}_0(t, \boldsymbol{\xi})$, which is the square

root of the variance estimate of $\hat{\Lambda}(t,\boldsymbol{\xi})$; refer to Kalbfleisch and Prentice (1980, p. 116). By the delta method, the standard errors of $\hat{S}(t,\boldsymbol{\xi})$ and $\tilde{S}(t,\boldsymbol{\xi})$ are given by

$$\hat{\sigma}_1(t,\boldsymbol{\xi}) \doteq \hat{S}(t,\boldsymbol{\xi})\tilde{\sigma}_0(t,\boldsymbol{\xi}) \quad \text{and} \quad \tilde{\sigma}_1(t,\boldsymbol{\xi}) \doteq \tilde{S}(t,\boldsymbol{\xi})\tilde{\sigma}_0(t,\boldsymbol{\xi})$$

respectively. The standard errors of $\log[-\log(\hat{S}(t,\boldsymbol{\xi}))]$ and $\log[-\log(\tilde{S}(t,\boldsymbol{\xi}))]$ are given by

$$\hat{\sigma}_2(t,\boldsymbol{\xi}) \doteq \frac{-\tilde{\sigma}_0(t,\boldsymbol{\xi})}{\log(\hat{S}(t,\boldsymbol{\xi}))} \quad \text{and} \quad \tilde{\sigma}_2(t,\boldsymbol{\xi}) \doteq \frac{\tilde{\sigma}_0(t,\boldsymbol{\xi})}{\hat{\Lambda}(t,\boldsymbol{\xi})}$$

respectively.

Let $z_{\alpha/2}$ be the upper $100(1-\frac{\alpha}{2})$ percentile point of the standard normal distribution. A $100(1-\alpha)\%$ confidence interval for the survivor function $S(t,\boldsymbol{\xi})$ is given in the following table.

CLTYPE	Method	Confidence Limits
LOG	PL	$\exp[\log(\hat{S}(t,\boldsymbol{\xi})) \pm z_{\frac{\alpha}{2}}\tilde{\sigma}_0(t,\boldsymbol{\xi})]$
LOG	CH	$\exp[\log(\tilde{S}(t,\boldsymbol{\xi})) \pm z_{\frac{\alpha}{2}}\tilde{\sigma}_0(t,\boldsymbol{\xi})]$
LOGLOG	PL	$\exp\{-\exp[\log(-\log(\hat{S}(t,\boldsymbol{\xi}))) \pm z_{\frac{\alpha}{2}}\hat{\sigma}_2(t,\boldsymbol{\xi})]\}$
LOGLOG	CH	$\exp\{-\exp[\log(-\log(\tilde{S}(t,\boldsymbol{\xi}))) \pm z_{\frac{\alpha}{2}}\tilde{\sigma}_2(t,\boldsymbol{\xi})]\}$
NORMAL	PL	$\hat{S}(t,\boldsymbol{\xi}) \pm z_{\frac{\alpha}{2}}\hat{\sigma}_1(t,\boldsymbol{\xi})$
NORMAL	CH	$\tilde{S}(t,\boldsymbol{\xi}) \pm z_{\frac{\alpha}{2}}\tilde{\sigma}_1(t,\boldsymbol{\xi})$

Effect Selection Methods

Five variable selection methods are available. The simplest method (and the default) is SELECTION=NONE, for which PROC PHREG fits the complete model as specified in the MODEL statement. The other four methods are FORWARD for forward selection, BACKWARD for backward elimination, STEPWISE for stepwise selection, and SCORE for best subsets selection. These methods are specified with the SELECTION= option in the MODEL statement.

When SELECTION=FORWARD, PROC PHREG first estimates parameters for variables forced into the model. These variables are the first n effects in the MODEL statement, where n is the number specified by the START= or INCLUDE= option in the MODEL statement (n is zero by default). Next, the procedure computes the adjusted chi-square statistics for each variable not in the model and examines the largest of these statistics. If it is significant at the SLSENTRY= level, the corresponding variable is added to the model. Once a variable is entered in the model, it is never removed from the model. The process is repeated until none of the remaining variables meet the specified level for entry or until the STOP= value is reached.

When SELECTION=BACKWARD, parameters for the complete model as specified in the MODEL statement are estimated unless the START= option is specified. In that case, only the parameters for the first n effects in the MODEL statement are estimated, where n is the number specified by the START= option. Results of the Wald test for individual parameters are examined. The least significant variable that does not meet the SLSSTAY= level for staying in the model is removed. Once a variable is removed from the model, it remains excluded. The process is repeated until no other variable in the model meets the specified level for removal or until the STOP= value is reached.

The SELECTION=STEPWISE option is similar to the SELECTION=FORWARD option except that variables already in the model do not necessarily remain. Variables are entered into and removed from the model in such a way that each forward selection step can be followed by one or more backward elimination steps. The stepwise selection process terminates if no further variable can be added to the model or if the variable just entered into the model is the only variable removed in the subsequent backward elimination.

For SELECTION=SCORE, PROC PHREG uses the branch-and-bound algorithm of Furnival and Wilson (1974) to find a specified number of models with the highest likelihood score (chi-square) statistic for all possible model sizes, from 1, 2, or 3 variables, and so on, up to the single model containing all of the explanatory variables. The number of models displayed for each model size is controlled by the BEST= option. You can use the START= option to impose a minimum model size, and you can use the STOP= option to impose a maximum model size. For instance, with BEST=3, START=2, and STOP=5, the SCORE selection method displays the best three models (that is, the three models with the highest score chi-squares) containing 2, 3, 4, and 5 variables. One of the limitations of the branch-and-bound algorithm is that it works only when each explanatory effect contains exactly one parameter—the SELECTION=SCORE option is not allowed when an explanatory effect in the MODEL statement contains a CLASS variable.

The SEQUENTIAL and STOPRES options can alter the default criteria for adding variables to or removing variables from the model when they are used with the FORWARD, BACKWARD, or STEPWISE selection method.

Assessment of the Proportional Hazards Model

The proportional hazards model specifies that the hazard function for the failure time T associated with a $p \times 1$ column covariate vector $\mathbf{Z}$ takes the form

$$\lambda(t;\mathbf{Z}) = \lambda_0(t)e^{\boldsymbol{\beta}'\mathbf{Z}}$$

where $\lambda_0(.)$ is an unspecified baseline hazard function and $\boldsymbol{\beta}$ is a $p \times 1$ column vector of regression parameters. Lin, Wei, and Ying (1993) present graphical and numerical methods for model assessment based on the cumulative sums of martingale residuals and their transforms over certain coordinates (such as covariate values or follow-up times). The distributions of these stochastic processes under the assumed model can be approximated by the distributions of certain zero-mean Gaussian processes whose realizations can be generated by simulation. Each observed residual pattern can then be compared, both graphically and numerically, with a number of realizations from the null distribution. Such comparisons enable you to assess objectively whether the observed residual pattern reflects anything beyond random fluctuation. These procedures are useful in determining appropriate functional forms of covariates and assessing the proportional hazards assumption. You use the ASSESS statement to carry out these model-checking procedures.

For a sample of n subjects, let $(X_i, \Delta_i, \mathbf{Z}_i)$ be the data of the ith subject; that is, X_i represents the observed failure time, Δ_i has a value of 1 if X_i is an uncensored time and 0 otherwise, and $\mathbf{Z}_i = (Z_{1i}, \ldots, Z_{pi})'$ is a p-vector of covariates. Let $N_i(t) = \Delta_i I(X_i \le t)$ and $Y_i(t) = I(X_i \ge t)$. Let

$$S^{(0)}(\boldsymbol{\beta}, t) = \sum_{i=1}^{n} Y_i(t)e^{\boldsymbol{\beta}'\mathbf{Z}_i} \quad \text{and} \quad \mathbf{Z}(\boldsymbol{\beta}, t) = \frac{\sum_{i=1}^{n} Y_i(t)e^{\boldsymbol{\beta}'\mathbf{Z}_i}\mathbf{Z}_i}{S^{(0)}(\boldsymbol{\beta}, t)}$$

Let $\hat{\boldsymbol{\beta}}$ be the maximum partial likelihood estimate of $\boldsymbol{\beta}$, and let $\mathcal{I}(\hat{\boldsymbol{\beta}})$ be the observed information matrix.

The martingale residuals are defined as

$$\hat{M}_i(t) = N_i(t) - \int_0^t Y_i(u) \mathrm{e}^{\hat{\boldsymbol{\beta}}' \mathbf{Z}_i} \, d\hat{\Lambda}_0(u), i = 1, \ldots, n$$

where $\hat{\Lambda}_0(t) = \int_0^t \frac{\sum_{i=1}^n dN_i(u)}{S^{(0)}(\hat{\boldsymbol{\beta}}, u)}$.

The empirical score process $\mathbf{U}(\hat{\boldsymbol{\beta}}, t) = (U_1(\hat{\boldsymbol{\beta}}, t), \ldots, U_p(\hat{\boldsymbol{\beta}}, t))'$ is a transform of the martingale residuals:

$$\mathbf{U}(\hat{\boldsymbol{\beta}}, t) = \sum_{i=1}^n \mathbf{Z}_i \hat{M}_i(t)$$

Checking the Functional Form of a Covariate

To check the functional form of the jth covariate, consider the partial-sum process of $\hat{M}_i = \hat{M}_i(\infty)$:

$$W_j(z) = \sum_{i=1}^n I(Z_{ji} \leq z) \hat{M}_i$$

Under that null hypothesis that the model holds, $W_j(z)$ can be approximated by the zero-mean Gaussian process

$$\begin{aligned} \hat{W}_j(z) \quad = \quad & \sum_{l=1}^n \Delta_l \left\{ I(Z_{jl} \leq z) - \frac{\sum_{i=1}^n Y_i(X_l) \mathrm{e}^{\boldsymbol{\beta}' \mathbf{Z}_i} I(Z_{ij} \leq z)}{S^{(0)}(\hat{\boldsymbol{\beta}}, X_l)} \right\} G_l - \\ & \sum_{k=1}^n \int_0^\infty Y_k(s) \mathrm{e}^{\hat{\boldsymbol{\beta}}' \mathbf{Z}_k} I(Z_{jk} \leq z) [\mathbf{Z}_k - \bar{\mathbf{Z}}(\hat{\boldsymbol{\beta}}, s)]' d\hat{\Lambda}_0(s) \\ & \times \mathcal{I}^{-1}(\hat{\boldsymbol{\beta}}) \sum_{l=1}^n \Delta_l [\mathbf{Z}_l - \bar{\mathbf{Z}}(\hat{\boldsymbol{\beta}}, X_l)] G_l \end{aligned}$$

where $(G_1, \ldots, G_n)$ are independent standard normal variables that are independent of $(X_i, \Delta_i, \mathbf{Z}_i), i = 1, \ldots, n$.

You can assess the functional form of the jth covariate by plotting a small number of realizations (the default is 20) of $\hat{W}_j(z)$ on the same graph as the observed $W_j(z)$ and visually comparing them to see how typical the observed pattern of $W_j(z)$ is of the null distribution samples. You can supplement the graphical inspection method with a Kolmogorov-type supremum test. Let s_j be the observed value of $S_j = \sup_z |W_j(z)|$ and let $\hat{S}_j = \sup_z |\hat{W}_j(z)|$. The p-value $\Pr(S_j \geq s_j)$ is approximated by $\Pr(\hat{S}_j \geq s_j)$, which in turn is approximated by generating a large number of realizations (1000 is the default) of $\hat{W}_j(.)$.

Checking the Proportional Hazards Assumption

Consider the standardized empirical score process for the jth component of $\mathbf{Z}$

$$U_j^*(t) = [\mathcal{I}^{-1}(\hat{\boldsymbol{\beta}})_{jj}]^{\frac{1}{2}} U_j(\hat{\boldsymbol{\beta}}, t),$$

Under the null hypothesis that the model holds, $U_j^*(t)$ can be approximated by

$$\begin{aligned}\hat{U}_j^*(t) &= [\mathcal{I}^{-1}(\hat{\boldsymbol{\beta}})_{jj}]^{\frac{1}{2}} \left\{ \sum_{l=1}^{n} I(X_l \le t)\Delta_l [Z_{jl} - \bar{Z}_j(\hat{\boldsymbol{\beta}}, t)] G_l - \right. \\ &\quad \sum_{k=1}^{n} \int_0^t Y_k(s) \mathrm{e}^{\hat{\boldsymbol{\beta}}' \mathbf{Z}_k} Z_{jk} [\mathbf{Z}_k - \bar{\mathbf{Z}}(\hat{\boldsymbol{\beta}}, s)]' d\hat{\Lambda}_0(s) \\ &\quad \left. \times \mathcal{I}^{-1}(\hat{\boldsymbol{\beta}}) \sum_{l=1}^{n} \Delta_l [\mathbf{Z}_l - \bar{\mathbf{Z}}(\hat{\boldsymbol{\beta}}, X_l)] G_l \right\}\end{aligned}$$

where $\bar{Z}_j(\hat{\boldsymbol{\beta}}, t)$ is the jth component of $\bar{\mathbf{Z}}(\hat{\boldsymbol{\beta}}, t)$, and $(G_1, \ldots, G_n)$ are independent standard normal variables that are independent of $(X_i, \Delta_i, \mathbf{Z}_i, (i = 1, \ldots, n)$.

You can assess the proportional hazards assumption for the jth covariate by plotting a few realizations of $\hat{U}_j^*(t)$ on the same graph as the observed $U_j^*(t)$ and visually comparing them to see how typical the observed pattern of $U_j^*(t)$ is of the null distribution samples. Again you can supplement the graphical inspection method with a Kolmogorov-type supremum test. Let s_j^* be the observed value of $S_j^* = \sup_t |U_j^*(t)|$ and let $\hat{S}_j^* = \sup_t |\hat{U}_j^*(t)|$. The p-value $\Pr[S_j^* \ge s_j^*]$ is approximated by $\Pr[\hat{S}_j^* \ge s_j^*]$, which in turn is approximated by generating a large number of realizations (1000 is the default) of $\hat{U}_j^*(.)$.

Specifics for Bayesian Analysis

To request a Bayesian analysis, you specify the new BAYES statement in addition to the PROC PHREG statement and the MODEL statement. You include a CLASS statement if you have effects that involve categorical variables. The FREQ or WEIGHT statement can be included if you have a frequency or weight variable, respectively, in the input data. The STRATA statement can be used to carry out a stratified analysis for the Cox model, but it is not allowed in the piecewise constant baseline hazard model. Programming statements can be used to create time-dependent covariates for the Cox model, but they are not allowed in the piecewise constant baseline hazard model. However, you can use the counting process style of input to accommodate time-dependent covariates that are not continuously changing with time for the piecewise constant baseline hazard model and the Cox model as well. The HAZARDRATIO statement enables you to request a hazard ratio analysis based on the posterior samples. The ASSESS, CONTRAST, ID, OUTPUT, and TEST statements, if specified, are ignored. Also ignored are the COVM and COVS options in the PROC PHREG statement and the following options in the MODEL statement: BEST=, CORRB, COVB, DETAILS, HIERARCHY=, INCLUDE=, MAXSTEP=, NOFIT, PLCONV=, SELECTION=, SEQUENTIAL, SLENTRY=, and SLSTAY=.

Piecewise Constant Baseline Hazard Model

Single Failure Time Variable

Let $\{(t_i, \mathbf{x}_i, \delta_i), i = 1, 2, \ldots, n\}$ be the observed data. Let $a_0 = 0 < a_1 < \ldots < a_{J-1} < a_J = \infty$ be a partition of the time axis.

Hazards in Original Scale

The hazard function for subject i is

$$h(t|\mathbf{x}_i; \boldsymbol{\theta}) = h_0(t) \exp(\boldsymbol{\beta}'\mathbf{x}_i)$$

where

$$h_0(t) = \lambda_j a_{j-1} \leq t < a_j (j = 1, \ldots, J)$$

The baseline cumulative hazard function is

$$H_0(t) = \sum_{j=1}^{J} \lambda_j \Delta_j(t)$$

where

$$\Delta_j(t) = \begin{cases} 0 & t < a_{j-1} \\ t - a_{j-1} & a_{j-1} \leq t < a_j \\ a_j - a_{j-1} & t \geq a_j \end{cases}$$

The log likelihood is given by

$$\begin{aligned} l(\boldsymbol{\lambda}, \boldsymbol{\beta}) &= \sum_{i=1}^{n} \delta_i \left[\sum_{j=1}^{J} I(a_{j-1} \leq t_i < a_j) \log \lambda_j + \boldsymbol{\beta}'\mathbf{x}_i \right] - \sum_{i=1}^{n} \left[\sum_{j=1}^{J} \Delta_j(t_i)\lambda_j \right] \exp(\boldsymbol{\beta}'\mathbf{x}_i) \\ &= \sum_{j=1}^{J} d_j \log \lambda_j + \sum_{i=1}^{n} \delta_i \boldsymbol{\beta}'\mathbf{x}_i - \sum_{j=1}^{J} \lambda_j \left[\sum_{i=1}^{n} \Delta_j(t_i) \exp(\boldsymbol{\beta}'\mathbf{x}_i) \right] \end{aligned}$$

where $d_j = \sum_{i=1}^{n} \delta_i I(a_{j-1} \leq t_i < a_j)$.

Note that for $1 \leq j \leq J$, the full conditional for λ_j is log-concave only when $d_j > 0$, but the full conditionals for the β's are always log-concave.

For a given $\boldsymbol{\beta}$, $\frac{\partial l}{\partial \boldsymbol{\lambda}} = 0$ gives

$$\tilde{\lambda_j}(\boldsymbol{\beta}) = \frac{d_j}{\sum_{i=1}^{n} \Delta_j(t_i) \exp(\boldsymbol{\beta}'\mathbf{x}_i)} (j = 1, \ldots, J)$$

Substituting these values into $l(\boldsymbol{\lambda}, \boldsymbol{\beta})$ gives the profile log likelihood for $\boldsymbol{\beta}$

$$l_p(\boldsymbol{\beta}) = \sum_{i=1}^{n} \delta_i \boldsymbol{\beta}'\mathbf{x}_i - \sum_{j=1}^{J} d_j \log \left[\sum_{l=1}^{n} \Delta_j(t_l) \exp(\boldsymbol{\beta}'\mathbf{x}_l) \right] + c$$

where $c = \sum_j (d_j \log d_j - d_j)$. Since the constant c does not depend on $\boldsymbol{\beta}$, it can be discarded from $l_p(\boldsymbol{\beta})$ in the optimization.

The MLE $\hat{\boldsymbol{\beta}}$ of $\boldsymbol{\beta}$ is obtained by maximizing

$$l_p(\boldsymbol{\beta}) = \sum_{i=1}^{n} \delta_i \boldsymbol{\beta}' \mathbf{x}_i - \sum_{j=1}^{J} d_j \log\left[\sum_{l=1}^{n} \Delta_j(t_l) \exp(\boldsymbol{\beta}' \mathbf{x}_l)\right]$$

with respect to $\boldsymbol{\beta}$, and the MLE $\hat{\boldsymbol{\lambda}}$ of $\boldsymbol{\lambda}$ is given by

$$\hat{\boldsymbol{\lambda}} = \tilde{\boldsymbol{\lambda}}(\hat{\boldsymbol{\beta}})$$

Let

$$\begin{aligned} \mathbf{S}_j^{(r)}(\boldsymbol{\beta}) &= \sum_{l=1}^{n} \Delta_j(t_l) \mathrm{e}^{\boldsymbol{\beta}' \mathbf{x}_l} \mathbf{x}_l^{\otimes r} \; r = 0, 1, 2 (j = 1, \ldots, J) \\ \mathbf{E}_j(\boldsymbol{\beta}) &= \frac{\mathbf{S}_j^{(1)}(\boldsymbol{\beta})}{S_j^{(0)}(\boldsymbol{\beta})} \end{aligned}$$

The partial derivatives of $l_p(\boldsymbol{\beta})$ are

$$\begin{aligned} \frac{\partial l_p(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} &= \sum_{i=1}^{n} \delta_i \mathbf{x}_i - \sum_{j=1}^{J} d_j \mathbf{E}_j(\boldsymbol{\beta}) \\ -\frac{\partial^2 l_p(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}^2} &= \sum_{j=1}^{J} d_j \left\{ \frac{\mathbf{S}_j^{(2)}(\boldsymbol{\beta})}{S_j^{(0)}(\boldsymbol{\beta})} - \left[\mathbf{E}_j(\boldsymbol{\beta})\right]\left[\mathbf{E}_j(\boldsymbol{\beta})\right]' \right\} \end{aligned}$$

The asymptotic covariance matrix for $(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\beta}})$ is obtained as the inverse of the information matrix given by

$$\begin{aligned} -\frac{\partial^2 l(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\beta}})}{\partial \boldsymbol{\lambda}^2} &= \mathcal{D}\left(\frac{d_1}{\hat{\lambda}_1^2}, \ldots, \frac{d_J}{\hat{\lambda}_J^2}\right) \\ -\frac{\partial^2 l(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\beta}})}{\partial \boldsymbol{\beta}^2} &= \sum_{j=1}^{J} \hat{\lambda}_j \mathbf{S}_j^{(2)}(\hat{\boldsymbol{\beta}}) \\ -\frac{\partial^2 l(\hat{\boldsymbol{\lambda}}, \hat{\boldsymbol{\beta}})}{\partial \boldsymbol{\lambda} \partial \boldsymbol{\beta}} &= (\mathbf{S}_1^{(1)}(\hat{\boldsymbol{\beta}}), \ldots, \mathbf{S}_J^{(1)}(\hat{\boldsymbol{\beta}})) \end{aligned}$$

See Example 6.5.1 in Lawless (2003) for details.

Hazards in Log Scale

By letting

$$\alpha_j = \log(\lambda_j) \, j = 1, \ldots, J$$

you can build a prior correlation among the λ_j's by using a correlated prior $\boldsymbol{\alpha} \sim N(\boldsymbol{\alpha}_0, \Sigma_\alpha)$, where $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_J)'$.

The log likelihood is given by

$$l(\boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{j=1}^{J} d_j \alpha_j + \sum_{i=1}^{n} \delta_i \boldsymbol{\beta}' \mathbf{x}_i - \sum_{j=1}^{J} \mathrm{e}^{\alpha_j} S_j^{(0)}(\boldsymbol{\beta})$$

Then the MLE of λ_j is given by

$$\mathrm{e}^{\hat{\alpha}_j} = \hat{\lambda}_j = \frac{d_j}{S_j^0(\hat{\boldsymbol{\beta}})}$$

Note that the full conditionals for α's and β's are always log-concave.

The asymptotic covariance matrix for $(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}})$ is obtained as the inverse of the information matrix formed by

$$\begin{aligned}
-\frac{\partial^2 l(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}})}{\partial \boldsymbol{\alpha}^2} &= \mathcal{D}\left(\mathrm{e}^{\hat{\alpha}_j} S_j^0(\hat{\boldsymbol{\beta}}), \ldots, \mathrm{e}^{\hat{\alpha}_J} S_J^0(\hat{\boldsymbol{\beta}})\right) \\
-\frac{\partial^2 l(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}})}{\partial \boldsymbol{\beta}^2} &= \sum_{j=1}^{J} \mathrm{e}^{\hat{\alpha}_j} \mathbf{S}_j^{(2)}(\hat{\boldsymbol{\beta}}) \\
-\frac{\partial^2 l(\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}})}{\partial \boldsymbol{\alpha} \partial \boldsymbol{\beta}} &= (\mathrm{e}^{\hat{\alpha}_j} \mathbf{S}_1^{(1)}(\hat{\boldsymbol{\beta}}), \ldots, \mathrm{e}^{\hat{\alpha}_j} \mathbf{S}_J^{(1)}(\hat{\boldsymbol{\beta}}))
\end{aligned}$$

Counting Process Style of Input

Let $\{((s_j, t_i], \mathbf{x}_i, \delta_i), i = 1, 2, \ldots, n\}$ be the observed data. Let $a_0 = 0 < a_1 < \ldots < a_k$ be a partition of the time axis, where $a_k > t_i$ for all $i = 1, 2, \ldots, n$.

Replacing $\Delta_j(t_i)$ with

$$\Delta_j((s_i, t_i]) = \begin{cases} 0 & t_i < a_{j-1} \vee s_i > a_j \\ t_i - \max(s_i, a_{j-1}) & a_{j-1} \le t_i < a_j \\ a_j - \max(s_i, a_{j-1}) & t_i \ge a_j \end{cases}$$

the formulation for the single failure time variable applies.

Priors for Model Parameters

For a Cox model, the model parameters are the regression coefficients. For a piecewise exponential model, the model parameters consist of the regression coefficients and the hazards or log-hazards. The priors for the hazards and the priors for the regression coefficients are assumed to be independent, while you can have a joint multivariate normal prior for the log-hazards and the regression coefficients.

Hazard Parameters

Let $\lambda_1, \ldots, \lambda_J$ be the constant baseline hazards.

Improper Prior The joint prior density is given by

$$p(\lambda_1, \ldots, \lambda_J) = \prod_{j=1}^{J} \frac{1}{\lambda_j}. \forall \lambda_j > 0$$

This prior is improper (nonintegrable), but the posterior distribution is proper as long as there is at least one event time in each of the constant hazard intervals.

Uniform Prior The joint prior density is given by

$$p(\lambda_1, \ldots, \lambda_J) \propto 1, \forall \lambda_j > 0$$

This prior is improper (nonintegrable), but the posteriors are proper as long as there is at least one event time in each of the constant hazard intervals.

Gamma Prior The gamma distribution $G(a, b)$ has a pdf

$$f_{a,b}(t) = \frac{b(bt)^{a-1}\mathrm{e}^{-bt}}{\Gamma(a)}, t > 0$$

where a is the shape parameter and b^{-1} is the scale parameter. The mean is $\frac{a}{b}$ and the variance is $\frac{a}{b^2}$.

Independent Gamma Prior Suppose for $j = 1, \ldots, J$, λ_j has an independent $G(a_j, b_j)$ prior. The joint prior density is given by

$$p(\lambda_1, \ldots, \lambda_J) \propto \prod_{j=1}^{J} \left\{ \lambda_j^{a_j - 1} \mathrm{e}^{-b_j \lambda_j} \right\}, \forall \lambda_j > 0$$

AR1 Prior $\lambda_1, \ldots, \lambda_J$ are correlated as follows:

$$\begin{aligned}
\lambda_1 &\sim G(a_1, b_1) \\
\lambda_2 &\sim G\left(a_2, \frac{b_2}{\lambda_1}\right) \\
\ldots &\quad \ldots \\
\lambda_J &\sim G\left(a_J, \frac{b_J}{\lambda_{J-1}}\right)
\end{aligned}$$

The joint prior density is given by

$$p(\lambda_1, \ldots, \lambda_J) \propto \lambda_1^{a_1 - 1} \mathrm{e}^{-b_1 \lambda_1} \prod_{j=2}^{J} \left(\frac{b_j}{\lambda_{j-1}} \right)^{a_j} \lambda_j^{a_j - 1} \mathrm{e}^{-\frac{b_j}{\lambda_{j-1}} \lambda_j}$$

Log-Hazard Parameters

Write $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_J)' \equiv (\log \lambda_1, \ldots, \log \lambda_J)'$.

Uniform Prior The joint prior density is given by

$$p(\alpha_1 \ldots \alpha_J) \propto 1, \forall -\infty < \alpha_i < \infty$$

Note that the uniform prior for the log-hazards is the same as the improper prior for the hazards.

Normal Prior Assume $\boldsymbol{\alpha}$ has a multivariate normal prior with mean vector $\boldsymbol{\alpha}_0$ and covariance matrix $\boldsymbol{\Psi}_0$. The joint prior density is given by

$$p(\boldsymbol{\alpha}) \propto \mathrm{e}^{-\frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\alpha}_0)'\boldsymbol{\Psi}_0^{-1}(\boldsymbol{\alpha}-\boldsymbol{\alpha}_0)}$$

Regression Coefficients

Let $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_k)'$ be the vector of regression coefficients.

Uniform Prior The joint prior density is given by

$$p(\beta_1, \ldots, \beta_k) \propto 1, \forall -\infty < \beta_i < \infty$$

This prior is improper, but the posterior distributions for $\boldsymbol{\beta}$ are proper.

Normal Prior Assume $\boldsymbol{\beta}$ has a multivariate normal prior with mean vector $\boldsymbol{\beta}_0$ and covariance matrix $\boldsymbol{\Sigma}_0$. The joint prior density is given by

$$p(\boldsymbol{\beta}) \propto \mathrm{e}^{-\frac{1}{2}(\boldsymbol{\beta}-\boldsymbol{\beta}_0)'\boldsymbol{\Sigma}_0^{-1}(\boldsymbol{\beta}-\boldsymbol{\beta}_0)}$$

Joint Multivariate Normal Prior for Log-Hazards and Regression Coefficients Assume $(\boldsymbol{\alpha}', \boldsymbol{\beta}')'$ has a multivariate normal prior with mean vector $(\boldsymbol{\alpha}_0', \boldsymbol{\beta}_0')'$ and covariance matrix $\boldsymbol{\Phi}_0$. The joint prior density is given by

$$p(\boldsymbol{\alpha}, \boldsymbol{\beta}) \propto \mathrm{e}^{-\frac{1}{2}[(\boldsymbol{\alpha}-\boldsymbol{\alpha}_0)', (\boldsymbol{\beta}-\boldsymbol{\beta}_0)']\boldsymbol{\Phi}_0^{-1}[(\boldsymbol{\alpha}-\boldsymbol{\alpha}_0)', (\boldsymbol{\beta}-\boldsymbol{\beta}_0)']'}$$

Posterior Distribution

Denote the observed data by D.

Cox Model

$$\pi(\boldsymbol{\beta}|D) \propto L_P(D|\boldsymbol{\beta})p(\boldsymbol{\beta})$$

where $L_P(D|\boldsymbol{\beta})$ is the partial likelihood function with regression coefficients $\boldsymbol{\beta}$ as parameters.

Piecewise Exponential Model

Hazard Parameters

$$\pi(\boldsymbol{\lambda}, \boldsymbol{\beta}|D) \propto L_H(D|\boldsymbol{\lambda}, \boldsymbol{\beta}) p(\boldsymbol{\lambda}) p(\boldsymbol{\beta})$$

where $L_H(D|\boldsymbol{\lambda}, \boldsymbol{\beta})$ is the likelihood function with hazards $\boldsymbol{\lambda}$ and regression coefficients $\boldsymbol{\beta}$ as parameters.

Log-Hazard Parameters

$$\pi(\boldsymbol{\alpha}, \boldsymbol{\beta}|D) \propto \begin{cases} L_{LH}(D|\boldsymbol{\alpha}, \boldsymbol{\beta}) p(\boldsymbol{\alpha}, \boldsymbol{\beta}) & \text{if} \quad (\boldsymbol{\alpha}', \boldsymbol{\beta}')' \sim \text{MVN} \\ L_{LH}(D|\boldsymbol{\alpha}, \boldsymbol{\beta}) p(\boldsymbol{\alpha}) p(\boldsymbol{\beta}) & \text{otherwise} \end{cases}$$

where $L_{LH}(D|\boldsymbol{\alpha}, \boldsymbol{\beta})$ is the likelihood function with log-hazards $\boldsymbol{\alpha}$ and regression coefficients $\boldsymbol{\beta}$ as parameters.

Sampling from the Posterior Distribution

PROC PHREG uses a Gibbs sampler to generate the posterior samples. See the section "Gibbs Sampler" on page 154 for a general discussion.

Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)'$ be the parameter vector. For the Cox model, the θ_i's are the regression coefficients β_i's, and for the piecewise constant baseline hazard model, the θ_i's consist of the baseline hazards λ_i's (or log baseline hazards α_i's) and the regression coefficients β_j's. Let $L(D|\boldsymbol{\theta})$ be the likelihood function, where D is the observed data. Note that for the Cox model, the likelihood contains the infinite-dimensional baseline hazard function and the Gamma process is perhaps the most commonly used prior process (Ibrahim, Chen, and Sinha 2001); however, Sinha, Ibrahim, and Chen (2003) justify using the partial likelihood as the likelihood function for the Bayesian analysis. Let $p(\boldsymbol{\theta})$ be the prior distribution. The full conditional distribution of θ_i is proportional to the joint distribution; that is,

$$\pi(\theta_i|\theta_j, i \neq j, D) \propto L(D|\boldsymbol{\theta}) p(\boldsymbol{\theta})$$

For instance, the one-dimensional conditional distribution of θ_1 given $\theta_j = \theta_j^*, 2 \leq j \leq k$, is computed as

$$\pi(\theta_1|\theta_j = \theta_j^*, 2 \leq j \leq k, D) = L(D|\boldsymbol{\theta} = (\theta_1, \theta_2^*, \ldots, \theta_k^*)') p(\boldsymbol{\theta} = (\theta_1, \theta_2^*, \ldots, \theta_k^*)')$$

Suppose you have a set of arbitrary starting values $\{\theta_1^{(0)}, \ldots, \theta_k^{(0)}\}$. Using the ARMS (adaptive rejection Metropolis sampling) algorithm of Gilks, Best, and Tan (1995), you do the following:

draw $\theta_1^{(1)}$ from $\pi(\theta_1|\theta_2^{(0)}, \ldots, \theta_k^{(0)}, D)$

draw $\theta_2^{(1)}$ from $\pi(\theta_2|\theta_1^{(1)}, \theta_3^{(0)}, \ldots, \theta_k^{(0)}, D)$

. . . . . .

draw $\theta_k^{(1)}$ from $\pi(\theta_k|\theta_1^{(1)}, \ldots, \theta_{k-1}^{(1)}, D)$

This completes one iteration of the Gibbs sampler. After one iteration, you have $\{\theta_1^{(1)}, \ldots, \theta_k^{(1)}\}$. After n iterations, you have $\{\theta_1^{(n)}, \ldots, \theta_k^{(n)}\}$.

You can output these posterior samples into a SAS data set through ODS. The following SAS statement outputs the posterior samples into the SAS data set Post:

```
ods output PosteriorSample=Post;
```

The data set also includes the variable LogPost, representing the log of the posterior log likelihood.

Starting Values of the Markov Chains

When the BAYES statement is specified, PROC PHREG generates one Markov chain containing the approximate posterior samples of the model parameters. Additional chains are produced when the Gelman-Rubin diagnostics are requested. Starting values (or initial values) can be specified in the INITIAL= data set in the BAYES statement. If INITIAL= option is not specified, PROC PHREG picks its own initial values for the chains. If the prior distribution of the parameter ω is proper, the starting values for ω are based on the estimated mean ($\hat{\omega}$) and standard deviation ($\hat{s}(\hat{\omega})$) of the posterior distribution given the MLE. If the prior distribution of ω is improper, the starting values for ω are based on the MLE and its standard error estimate; that is, $\hat{\omega}$ is the MLE of ω and $\hat{s}(\hat{\omega})$ is the standard error of maximum likelihood estimator.

Denote $[x]$ as the integral value of x.

Constant Baseline Hazards λ_i's

For the first chain that the summary statistics and diagnostics are based on, the initial values are

$$\lambda_i^{(0)} = \hat{\lambda}_i$$

For subsequent chains, the starting values are picked in two different ways according to the total number of chains specified. If the total number of chains specified is less than or equal to 10, initial values of the rth chain ($2 \leq r \leq 10$) are given by

$$\lambda_i^{(0)} = \hat{\lambda}_i \mathrm{e}^{\pm\left([\frac{r}{2}]+2\right)\hat{s}(\hat{\lambda}_i)}$$

with the plus sign for odd r and minus sign for even r. If the total number of chains is greater than 10, initial values are picked at random over a wide range of values. Let u_i be a uniform random number between 0 and 1; the initial value for λ_i is given by

$$\lambda_i^{(0)} = \hat{\lambda}_i \mathrm{e}^{16(u_i - 0.5)\hat{s}(\hat{\lambda}_i)}$$

Regression Coefficients and Log-Hazard Parameters θ_i's

The θ_i's are the regression coefficients β_i's, and in the piecewise exponential model, include the log-hazard parameters α_i's. For the first chain that the summary statistics and regression diagnostics are based on, the initial values are

$$\theta_i^{(0)} = \hat{\theta}_i$$

If the number of chains requested is less than or equal to 10, initial values for the rth chain ($2 \leq r \leq 10$) are given by

$$\theta_i^{(0)} = \hat{\theta}_i \pm \left(2 + \left[\frac{r}{2}\right]\right)\hat{s}(\hat{\theta}_i)$$

with the plus sign for odd r and minus sign for even r. When there are more than 10 chains, the initial value for the θ_i is picked at random over the range $(\hat{\theta}_i - 8\hat{s}(\hat{\theta}_i), \hat{\theta}_i + 8\hat{s}(\hat{\theta}_i))$; that is,

$$\theta_i^{(0)} = \hat{\theta}_i + 16(u_i - 0.5)\hat{s}(\hat{\theta}_i)$$

where u_i is a uniform random number between 0 and 1.

Fit Statistics

Denote the observed data by D. Let $\boldsymbol{\theta}$ be the vector of parameters of length k. Let $L(D|\boldsymbol{\theta})$ be the likelihood. The deviance information criterion (DIC) proposed in Spiegelhalter et al. (2002) is a Bayesian model assessment tool. Let $\text{Dev}(\boldsymbol{\theta}) = -2\log L(D|\boldsymbol{\theta})$. Let $\overline{\text{Dev}(\theta)}$ and $\bar{\boldsymbol{\theta}}$ be the corresponding posterior means of $\text{Dev}(\boldsymbol{\theta})$ and $\boldsymbol{\theta}$, respectively. The deviance information criterion is computed as

$$\text{DIC} = 2\overline{\text{Dev}(\boldsymbol{\theta})} - \text{Dev}(\bar{\boldsymbol{\theta}})$$

Also computed is

$$pD = \overline{\text{Dev}(\boldsymbol{\theta})} - \text{Dev}(\bar{\boldsymbol{\theta}})$$

where pD is interpreted as the effective number of parameters.

Note that $\text{Dev}(\boldsymbol{\theta})$ defined here does not have the standardizing term as in the section "Deviance Information Criterion (DIC)" on page 172. Nevertheless, DIC calculated here is still useful for variable selection.

Posterior Distribution for Quantities of Interest

Let $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_k)'$ be the parameter vector. For the Cox model, the θ_i's are the regression coefficients β_i's, and for the piecewise constant baseline hazard model, the θ_i's consist of the baseline hazards λ_i's (or log baseline hazards α_i's) and the regression coefficients β_j's. Let $\mathcal{S} = \{\boldsymbol{\theta}^{(r)}, r = 1, \ldots, N\}$ be the chain representing the posterior distribution for $\boldsymbol{\theta}$.

Consider a quantity of interest τ that can be expressed as a function $f(\boldsymbol{\theta})$ of the parameter vector $\boldsymbol{\theta}$. You can construct the posterior distribution of τ by evaluating the function $f(\boldsymbol{\theta}^{(r)})$ for each $\boldsymbol{\theta}^{(r)}$ in $\mathcal{S}$. The posterior chain for τ is $\{f(\boldsymbol{\theta}^{(r)}), r = 1, \ldots, N\}$. Summary statistics such as mean, standard deviation, percentiles, and credible intervals are used to describe the posterior distribution of τ.

Hazard Ratio

As shown in the section "Hazard Ratios" on page 4590, a log-hazard ratio is a linear combination of the regression coefficients. Let $\mathbf{h}$ be the vector of linear coefficients. The posterior sample for this hazard ratio is the set $\{\exp(\mathrm{h}'\boldsymbol{\beta}^{(r)}), r = 1, \ldots, N\}$.

Survival Distribution

Let x be a covariate vector of interest.

Cox Model Let $\{(t_i, \mathbf{z}_i, \delta_i), i = 1, 2, \ldots, n\}$ be the observed data. Define

$$Y_i(t) = \begin{cases} 1 & t < t_i \\ 0 & \text{otherwise} \end{cases}$$

Consider the rth draw $\boldsymbol{\beta}^{(r)}$ of $\mathcal{S}$. The baseline cumulative hazard function at time t is given by

$$H_0(t|\boldsymbol{\beta}^{(r)}) = \sum_{i:t_i \le t} \frac{\delta_i}{\sum_{l=1}^{n} Y_l(t_i)\exp(\mathbf{z}_l'\boldsymbol{\beta}^{(r)})}$$

For the given covariate vector x, the cumulative hazard function at time t is

$$H(t;\mathbf{x}|\boldsymbol{\beta}^{(r)}) = H_0(t|\boldsymbol{\beta}^{(r)})\exp(\mathbf{x}'\boldsymbol{\beta}^{(r)})$$

and the survival function at time t is

$$S(t;\mathbf{x}|\boldsymbol{\beta}^{(r)}) = \exp[-H^{(r)}(t;\mathbf{x}|\boldsymbol{\beta}^{(r)})]$$

Piecewise Exponential Model Let $0 = a_0 < a_1 < \ldots < a_J < \infty$ be a partition of the time axis. Consider the rth draw $\boldsymbol{\theta}^{(r)}$ in $\mathcal{S}$, where $\boldsymbol{\theta}^{(r)}$ consists of $\boldsymbol{\lambda}^{(r)} = (\lambda_1^{(r)}, \ldots, \lambda_J^{(r)})'$ and $\boldsymbol{\beta}^{(r)}$. The baseline cumulative hazard function at time t is

$$H_0(t|\boldsymbol{\lambda}^{(r)}) = \sum_{j=1}^{J} \lambda_j^{(r)} \Delta_j(t)$$

where

$$\Delta_j(t) = \begin{cases} 0 & t < a_{j-1} \\ t - a_{j-1} & a_{j-1} \le t < a_j \\ a_j - a_{j-1} & t \ge a_j \end{cases}$$

For the given covariate vector $\mathbf{x}$, the cumulative hazard function at time t is

$$H(t;\mathbf{x}|\boldsymbol{\lambda}^{(r)}, \boldsymbol{\beta}^{(r)}) = H_0(t|\boldsymbol{\lambda}^{(r)})\exp(\mathbf{x}'\boldsymbol{\beta}^{(r)})$$

and the survival function at time t is

$$S(t;\mathbf{x}|\boldsymbol{\lambda}^{(r)}, \boldsymbol{\beta}^{(r)}) = \exp[-H(t;\mathbf{x}|\boldsymbol{\lambda}^{(r)}, \boldsymbol{\beta}^{(r)})]$$

Computational Resources

Let n be the number of observations in a BY group. Let p be the number of explanatory variables. The minimum working space (in bytes) needed to process the BY group is

$$\max\{12n, 24p^2 + 160p\}$$

Extra memory is needed for certain TIES= options. Let k be the maximum multiplicity of tied times. The TIES=DISCRETE option requires extra memory (in bytes) of

$$4k(p^2 + 4p)$$

The TIES=EXACT option requires extra memory (in bytes) of

$$24(k^2 + 5k)$$

If sufficient space is available, the input data are also kept in memory. Otherwise, the input data are reread from the utility file for each evaluation of the likelihood function and its derivatives, with the resulting execution time substantially increased.

Input and Output Data Sets

OUTEST= Output Data Set

The OUTEST= data set contains one observation for each BY group containing the maximum likelihood estimates of the regression coefficients. If you also use the COVOUT option in the PROC PHREG statement, there are additional observations containing the rows of the estimated covariance matrix. If you specify SELECTION=FORWARD, BACKWARD, or STEPWISE, only the estimates of the parameters and covariance matrix for the final model are output to the OUTEST= data set.

Variables in the OUTEST= Data Set

The OUTEST= data set contains the following variables:

- any BY variables specified
- _TIES_, a character variable of length 8 with four possible values: BRESLOW, DISCRETE, EFRON, and EXACT. These are the four values of the TIES= option in the MODEL statement.

- _TYPE_, a character variable of length 8 with two possible values: PARMS for parameter estimates or COV for covariance estimates. If both the COVM and COVS options are specified in the PROC LIFETEST statement along with the COVOUT option, _TYPE_='COVM' for the model-based covariance estimates and _TYPE_='COVS' for the robust sandwich covariance estimates.
- _STATUS_, a character variable indicating whether the estimates have converged
- _NAME_, a character variable containing the name of the TIME variable for the row of parameter estimates and the name of each explanatory variable to label the rows of covariance estimates
- one variable for each regression coefficient and one variable for the offset variable if the OFFSET= option is specified. If an explanatory variable is not included in the final model in a variable selection process, the corresponding parameter estimates and covariances are set to missing.
- _LNLIKE_, a numeric variable containing the last computed value of the log likelihood

Parameter Names in the OUTEST= Data Set

For continuous explanatory variables, the names of the parameters are the same as the corresponding variables. For CLASS variables, the parameter names are obtained by concatenating the corresponding CLASS variable name with the CLASS category; see the PARAM= option in the CLASS statement and the section "CLASS Variable Parameterization" on page 4580 for more details. For interaction and nested effects, the parameter names are created by concatenating the names of each component effect.

INEST= Input Data Set

You can specify starting values for the maximum likelihood iterative algorithm in the INEST= data set. The INEST= data set has the same structure as the OUTEST= data set but is not required to have all the variables or observations that appear in the OUTEST= data set.

The INEST= data set must contain variables that represent the regression coefficients of the model. If BY processing is used, the INEST= data set should also include the BY variables, and there must be one observation for each BY group. If the INEST= data set also contains the _TYPE_ variable, only observations with _TYPE_ value 'PARMS' are used as starting values.

OUT= Output Data Set in the OUTPUT Statement

The OUT= data set in the OUTPUT statement contains all the variables in the input data set, along with statistics you request by specifying *keyword*=*name* options. The new variables contain a variety of diagnostics that are calculated for each observation in the input data set.

OUT= Output Data Set in the BASELINE Statement

The OUT= data set in the BASELINE statement contains all the variables in the COVARIATES= data set, along with statistics you request by specifying *keyword=name* options. For unstratified input data, there are 1+n observations in the OUT= data set for each observation in the COVARIATES= data set, where n is the number of distinct event times in the input data. For input data that are stratified into k strata, with n_i distinct events in the ith stratum, $i = 1, \ldots, k$, there are 1+n_i observations for the ith stratum in the OUT= data set for each observation in the COVARIATES= data set.

OUTPOST= Output Data Set in the BAYES Statement

The OUTPOST= data set contains the generated posterior samples. There are 2+n variables, where n is the number of model parameters. The variable Iteration represents the iteration number and the variable LogPost contains the log posterior likelihood values. The other n variables represent the draws of the Markov chain for the model parameters.

Displayed Output

If you use the NOPRINT option in the PROC PHREG statement, the procedure does not display any output. Otherwise, PROC PHREG displays results of the analysis in a collection of tables. The tables are listed separately for the maximum likelihood analysis and for the Bayesian analysis.

Maximum Likelihood Analysis Displayed Output

Model Information

The "Model Information" table displays the two-level name of the input data set, the name and label of the failure time variable, the name and label of the censoring variable and the values indicating censored times, the model (either the Cox model or the piecewise constant baseline hazard model), the name and label of the OFFSET variable, the name and label of the FREQ variable, the name and label of the WEIGHT variable, and the method of handling ties in the failure time for the Cox model. For ODS purposes, the name of the "Model Information" table is "ModelInfo."

Number of Observations

The "Number of Observations" table displays the number of observations read and used in the analysis. For ODS purposes, the name of the "Number of Observations" is "NObs."

Summary of the Number of Event and Censored Values

The "Summary of the Number of Event and Censored Values" table displays, for each stratum, the breakdown of the number of events and censored values. For ODS purposes, the name of the "Summary of the Number of Event and Censored Values" table is "CensoredSummary."

Class Level Information

The "Class Level Information" table is displayed when there are CLASS variables in the model. The table lists the categories of every CLASS variable that is used in the model and the corresponding design variable values. For ODS purposes, the name of the "Class Level Information" table is "ClassLevelInfo."

Descriptive Statistics for Continuous Explanatory Variables

The "Simple Statistics for Continuous Explanatory Variables" table is displayed when you specify the SIMPLE option in the PROC PHREG statement. The table contains, for each stratum, the mean, standard deviation, and minimum and maximum for each continuous explanatory variable in the MODEL statement. For ODS purposes, the name of the "Descriptive Statistics for Continuous Explanatory Variables" table is "SimpleStatistics."

Frequency Distribution of CLASS Variables

The "Frequency Distribution of CLASS Variables" table is displayed if you specify the SIMPLE option in the PROC PHREG statement and there are CLASS variables in the model. The table lists the frequency of the levels of the CLASS variables. For ODS purposes, the name of the "Frequency Distribution of CLASS Variables" table is "ClassLevelFreq."

Maximum Likelihood Iteration History

The "Maximum Likelihood Iteration History" table is displayed if you specify the ITPRINT option in the MODEL statement. The table contains the iteration number, ridge value or step size, log likelihood, and parameter estimates at each iteration. For ODS purposes, the name of the "Maximum Likelihood Iteration History" table is "IterHistory."

Gradient of Last Iteration

The "Gradient of Last Iteration" table is displayed if you specify the ITPRINT option in the MODEL statement. For ODS purposes, the name of the "Gradient of Last Iteration" table is "LastGradient."

Convergence Status

The "Convergence Status" table displays the convergence status of the Newton-Raphson maximization. For ODS purposes, the name of the "Convergence Status" table is "ConvergenceStatus."

Model Fit Statistics

The "Model Fit Statistics" table displays the values of -2 log likelihood for the null model and the fitted model, the AIC, and SBC. For ODS purposes, the name of the "Model Fit Statistics" table is "FitStatistics."

Testing Global Null Hypothesis: BETA=0

The "Testing Global Null Hypothesis: BETA=0" table displays results of the likelihood ratio test, the score test, and the Wald test for testing the hypothesis that all parameters are zero. For ODS purpose, the name of the "Testing Global Null Hypothesis: BETA=0" table is "GlobalTests."

Type 3 Tests

The "Type 3 Tests" table is displayed if the model contains CLASS variables. The table displays type 3 Wald chi-square statistic, the degrees of freedom, and the p-value for each effect in the model. For ODS purposes, the name of "Type 3 Tests" is "Type3."

Analysis of Maximum Likelihood Estimates

The "Analysis of Maximum Likelihood Estimates" table displays the maximum likelihood estimate of the parameter; the estimated standard error, computed as the square root of the corresponding diagonal element of the estimated covariance matrix; the ratio of the robust standard error estimate to the model-based standard error estimate if you specify the COVS option in the PROC statement; the Wald Chi-Square statistic, computed as the square of the parameter estimate divided by its standard error estimate; the degrees of freedom of the Wald chi-square statistic, which has a value of 1 unless the corresponding parameter is redundant or infinite, in which case the value is 0; the p-value of the Wald chi-square statistic with respect to a chi-square distribution with one degree of freedom; the hazard ratio estimate; and the confidence limits for the hazard ratio if you specified the RISKLIMITS option in the MODEL statement. For ODS purposes, the name of the "Analysis of Maximum Likelihood Estimates" table is "ParameterEstimates."

Regression Models Selected by Score Criterion

The "Regression Models Selected by Score Criterion" table is displayed if you specify SELECTION=SCORE in the MODEL statement. The table contains the number of explanatory variables in each model, the score chi-square statistic, and the names of the variables included in the model. For ODS purposes, the name of the "Regression Models Selected by Score Criterion" table is "BestSubsets."

Analysis of Effects Eligible for Entry

The "Analysis of Effects Eligible for Entry" table is displayed if you use the FORWARD or STEPWISE selection method and you specify the DETAILS option in the MODEL statement. The table contains the score chi-square statistic for testing the significance of each variable not in the model (after adjusting for the variables already in the model), and the p-value of the chi-square statistic

with respect to a chi-square distribution with one degree of freedom. This table is produced before a variable is selected for entry in a forward selection step. For ODS purposes, the name of the "Analysis of Effects Eligible for Entry" table is "EffectsToEntry."

Analysis of Effects Eligible for Removal

The "Analysis of Effects Eligible for Removal" table is displayed if you use the BACKWARD or STEPWISE selection method and you specify the DETAILS option in the MODEL statement. The table contains the Wald chi-square statistic for testing the significance of each candidate effect for removal, the degrees of freedom of the Wald chi-square, and the corresponding *p*-value. This table is produced before an effect is selected for removal. For ODS purposes, the name of the "Analysis of Effects Eligible for Removal" table is "EffectsToRemoval."

Summary of Backward Elimination

The "Summary of Backward Elimination" table is displayed if you specify the SELECTION=BACKWARD option in the MODEL statement. The table contains the step number, the effects removed at each step, the corresponding chi-square statistic, the degrees of freedom, and the *p*-value. For ODS purpose, the name of the "Summary of Backward Elimination" table is "ModelBuildingSummary."

Summary of Forward Selection

The "Summary of Forward Selection" table is displayed if you specify the SELECTION=FORWARD option in the MODEL statement. The table contains the step number, the effects entered at each step, the corresponding chi-square statistic, the degrees of freedom, and the *p*-value. For ODS purpose, the name of the "Summary of Forward Selection" table is "ModelBuildingSummary."

Summary of Stepwise Selection

The "Summary of Stepwise Selection" table is displayed if you specify SELECTION=STEPWISE is specified in the MODEL statement. The table contains the step number, the effects entered or removed at each step, the corresponding chi-square statistic, the degrees of freedom, and the corresponding *p*-value. For ODS purpose, the name of the "Summary of Stepwise Selection" table is "ModelBuildingSummary."

Covariance Matrix

The "Covariance Matrix" table is displayed if you specify the COVB option in the MODEL statement. The table contains the estimated covariance matrix for the parameter estimates. For ODS purposes, the name of the "Covariance Matrix" table is "CovB."

Correlation Matrix

The "Correlation Matrix" table is displayed if you specify the COVB option in the MODEL statement. The table contains the estimated correlation matrix for the parameter estimates. For ODS purposes, the name of the "Correlation Matrix" table is "CorrB."

Hazard Ratios for label

The "Hazard Ratios for *label*" table is displayed if you specify the HAZARDRATIO statement. The table displays the estimate and confidence limits for each hazard ratio. For ODS purposes, the name of the "Hazard Ratios for *label*" table is "HazardRatios."

Linear Coefficients for label

The "Linear Coefficients *label*" table is displayed if you specify the E option in the TEST statement with *label* being the TEST statement label. The table contains the coefficients and constants of the linear hypothesis. For ODS purposes, the name of the "Linear Coefficients for *label*" table is "TestCoeff."

L[cov(b)]L' and Lb-c

The "L[cov(b)]L' and Lb-c" table is displayed if you specified the PRINT option in a TEST statement with *label* being the TEST statement label. The table displays the intermediate calculations of the Wald test. For ODS purposes, the name of the "L[cov(b)]L' and Lb-c" table is "TestPrint1."

Ginv(L[cov(b)]L') and Ginv(L[cov(b)]L')(Lb-c)

The "Ginv(L[cov(b)]L') and Ginv(L[cov(b)]L')(Lb-c)" table is displayed if you specified the PRINT option in a TEST statement with *label* being the TEST statement label. The table displays the intermediate calculations of the Wald test. For ODS purposes, the name of the "Ginv(L[cov(b)]L') and Ginv(L[cov(b)]L')(Lb-c)" table is "TestPrint2."

label Test Results

The "*label* Test Results" table is displayed if you specify a TEST statement with *label* being the TEST statement label. The table contains the Wald chi-square statistic, the degrees of freedom, and the p-value. For ODS purposes, the name of "*label* Test Results" table is "TestStmts."

Average Effect for label

The "Average Effect for *label*" table is displayed if the AVERAGE option is specified in a TEST statement with *label* being the TEST statement label. The table contains the weighted average of the parameter estimates for the variables in the TEST statement, the estimated standard error, the z-score, and the p-value. For ODS purposes, the name of the "Average Effect for *label*" is "TestAverage."

Reference Set of Covariates for Plotting

The "Reference Set of Covariates for Plotting" table is displayed if the PLOTS= option is requested without specifying the COVARIATES= data set in the BASELINE statement. The tables contains the values of the covariates for the reference set, where the reference levels are used for the CLASS variables and the sample averages for the continuous variables.

Bayesian Analysis Displayed Output

Model Information

The "Model Information" table displays the two-level name of the input data set, the name and label of the failure time variable, the name and label of the censoring variable and the values indicating censored times, the model (either the Cox model or the piecewise constant baseline hazard model), the name and label of the OFFSET variable, the name and label of the FREQ variable, the name and label of the WEIGHT variable, the method of handling ties in the failure time, the number of burn-in iterations, the number of iterations after the burn-in, and the number of thinning iterations. For ODS purposes, the name of the "Model Information" table is "ModelInfo."

Number of Observations

The "Number of Observations" table displays the number of observations read and used in the analysis. For ODS purposes, the name of the "Number of Observations" is "NObs."

Summary of the Number of Event and Censored Values

The "Summary of the Number of Event and Censored Values" table displays, for each stratum, the breakdown of the number of events and censored values. This table is not produced if the NONSUMMARY option is specified in the PROC PHREG statement. For ODS purposes, the name of the "Summary of the Number of Event and Censored Values" table is "CensoredSummary."

Descriptive Statistics for Continuous Explanatory Variables

The "Simple Statistics for Continuous Explanatory Variables" table is displayed when you specify the SIMPLE option in the PROC PHREG statement. The table contains, for each stratum, the mean, standard deviation, and minimum and maximum for each continuous explanatory variable in the MODEL statement. For ODS purposes, the name of the "Descriptive Statistics for Continuous Explanatory Variables" table is "SimpleStatistics."

Class Level Information

The "Class Level Information" table is displayed if there are CLASS variables in the model. The table lists the categories of every CLASS variable in the model and the corresponding design variable values. For ODS purposes, the name of the "Class Level Information" table is "ClassLevelInfo."

Frequency Distribution of CLASS Variables

The "Frequency Distribution of CLASS Variables" table is displayed if you specify the SIMPLE option in the PROC PHREG statement and there are CLASS variables in the model. The table lists the frequency of the levels of the CLASS variables. For ODS purposes, the name of the "Frequency Distribution of CLASS Variables" table is "ClassLevelFreq."

Regression Parameter Information

The "Regression Parameter Information" table displays the names of the parameters and the corresponding level information of effects containing the CLASS variables. For ODS purposes, the name of the "Regression Parameter Information" table is "ParmInfo."

Constant Baseline Hazard Time Intervals

The "Constant Baseline Hazard Time Intervals" table displays the intervals of constant baseline hazard and the corresponding numbers of failure times and event times. This table is produced only if you specify the PIECEWISE option in the BAYES statement. For ODS purposes, the name of the "Constant Baseline Hazard Time Intervals" table is "Interval."

Maximum Likelihood Estimates

The "Maximum Likelihood Estimates" table displays, for each parameter, the maximum likelihood estimate, the estimated standard error, and the 95% confidence limits. For ODS purposes, the name of the "Maximum Likelihood Estimates" table is "ParameterEstimates."

Hazard Prior

The "Hazard Prior" table is displayed if you specify the PIECEWISE=HAZARD option in the BAYES statement. It describes the prior distribution of the hazard parameters. For ODS purposes, the name of the "Hazard Prior" table is "HazardPrior."

Log-Hazard Prior

The "Log-Hazard Prior" table is displayed if you specify the PIECEWISE=LOGHAZARD option in the BAYES statement. It describes the prior distribution of the log-hazard parameters. For ODS purposes, the name of the "Log-Hazard Prior" table is "HazardPrior."

Coefficient Prior

The "Coefficient Prior" table displays the prior distribution of the regression coefficients. For ODS purposes, the name of the "Coefficient Prior" table is "CoeffPrior."

Initial Values

The "Initial Values" table is displayed if you specify the INITIAL option in the BAYES statement. The table contains the initial values of the parameters for the Gibbs sampling. For ODS purposes, the name of the "Initial Values" table is "InitialValues."

Fit Statistics

The "Fit Statistics" table displays the DIC and pD statistics for each parameter. For ODS purposes, the name of the "Fit Statistics" table is "FitStatistics."

Posterior Summaries

The "Posterior Summaries" table displays the size of the posterior sample, the mean, the standard error, and the percentiles for each model parameter. For ODS purposes, the name of the "Posterior Summaries" table is "PostSummaries."

Posterior Intervals

The "Posterior Intervals" table displays the equal-tail interval and the HPD interval for each model parameter. For ODS purposes, the name of the "Posterior Intervals" table is "PostIntervals."

Posterior Covariance Matrix

The "Posterior Covariance Matrix" table is produced if you include COV in the SUMMARY= option in the BAYES statement. This tables displays the sample covariance of the posterior samples. For ODS purposes, the name of the "Posterior Covariance Matrix" table is "Cov."

Posterior Correlation Matrix

The "Posterior Correlation Matrix" table is displayed if you include CORR in the SUMMARY= option in the BAYES statement. The table contains the sample correlation of the posterior samples. For ODS purposes, the name of the "Posterior Correlation Matrix" table is "Corr."

Posterior Autocorrelations

The "Posterior Autocorrelations" table displays the lag 1, lag 5, lag 10, and lag 50 autocorrelations for each parameter. For ODS purposes, the name of the "Posterior Autocorrelations" table is "AutoCorr."

Gelman-Rubin Diagnostics

The "Gelman-Rubin Diagnostics" table is produced if you include GELMAN in the DIAGNOSTIC= option in the BAYES statement. This table displays the estimate of the potential scale reduction factor and its 97.5% upper confidence limit for each parameter. For ODS purposes, the name of the "Gelman-Rubin Diagnostics" table is "Gelman."

Geweke Diagnostics

The "Geweke Diagnostics" table displays the Geweke statistic and its p-value for each parameter. For ODS purposes, the name of the "Geweke Diagnostics" table is "Geweke."

Raftery-Lewis Diagnostics

The "Raftery-Lewis Diagnostics" tables is produced if you include RAFTERY in the DIAGNOSTIC= option in the BAYES statement. This table displays the Raftery and Lewis diagnostics for each variable. For ODS purposes, the name of the "Raftery-Diagnostics" table is "Raftery."

Heidelberger-Welch Diagnostics

The "Heidelberger-Welch Diagnostics" table is displayed if you include HEIDELBERGER in the DIAGNOSTIC= option in the BAYES statement. This table describes the results of a stationary test and a halfwidth test for each parameter. For ODS purposes, the name of the "Heidelberger-Welch Diagnostics" table is "Heidelberger."

Effective Sample Sizes

The "Effective Sample Sizes" table displays, for each parameter, the effective sample size, the correlation time, and the efficiency. For ODS purposes, the name of the "Effective Sample Sizes" table is "ESS."

Hazard Ratios for label

The "Hazard Ratios for *label*" table is displayed if you specify the HAZARDRATIO statement. The table displays the posterior summary for each hazard ratio. The summary includes the mean, standard error, quartiles, and equal-tailed and HPD intervals. For ODS purposes, the name of the "Hazard Ratios for *label*" table is "HazardRatios."

Reference Set of Covariates for Plotting

The "Reference Set of Covariates for Plotting" table is displayed if the PLOTS= option is requested without specifying the COVARIATES= data set in the BASELINE statement. The table contains the values of the covariates for the reference set, where the reference levels are used for the CLASS variables and the sample averages for the continuous variables.

ODS Table Names

PROC PHREG assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed separately in Table 64.5 for the maximum likelihood analysis and in

Table 64.6 for the Bayesian analysis. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 64.5 ODS Tables for a Maximum Likelihood Analysis Produced by PROC PHREG

ODS Table Name	Description	Statement / Option
BestSubsets	Best subset selection	MODEL / SELECTION=SCORE
CensoredSummary	Summary of event and censored observations	default
ClassLevelFreq	Frequency distribution of CLASS variables	CLASS, PROC / SIMPLE
ClassLevelInfo	CLASS variable levels and design variables	CLASS
ContrastCoeff	L matrix for contrasts	CONTRAST / E
ContrastEstimate	Individual contrast estimates	CONTRAST / ESTIMATE=
ContrastTest	Wald test for contrasts	CONTRAST
ConvergenceStatus	Convergence status	default
CorrB	Estimated correlation matrix of parameter estimators	MODEL / CORRB
CovB	Estimated covariance matrix of parameter estimators	MODEL / COVB
EffectsToEnter	Analysis of effects for entry	MODEL / SELECTION=F|S
EffectsToremove	Analysis of effects for removal	MODEL / SELECTION=B|S
FitStatistics	Model fit statistics	default
FunctionalFormSupTest	Supremum test for functional form	ASSESS / VAR=
GlobalScore	Global chi-square test	MODEL / NOFIT
GlobalTests	Tests of the global null hypothesis	default
HazardRatios	Hazard ratios and confidence limits	HAZARDRATIO
IterHistory	Iteration history	MODEL /ITPRINT
LastGradient	Last evaluation of gradient	MODEL / ITPRINT
ModelBuildingSummary	Summary of model building	MODEL / SELECTION=B|F|S
ModelInfo	Model information	default
NObs	Number of observations	default
ParameterEstimates	Maximum likelihood estimates of model parameters	default
ProportionalHazardsSupTest	Supremum test for proportional hazards assumption	ASSESS / PH
ResidualChiSq	Residual chi-square	MODEL / SELECTION=F|B
ReferenceSet	Reference set of covariates for plotting	PROC / PLOTS=
SimpleStatistics	Summary statistics of input continuous explanatory variables	PROC / SIMPLE
TestAverage	Average effect for test	TEST / AVERAGE
TestCoeff	Coefficients for linear hypotheses	TEST / E
TestPrint1	**L**[cov(**b**)]**L**' and **Lb**-**c**	TEST / PRINT
TestPrint2	Ginv(**L**[cov(**b**)]**L**') and Ginv(**L**[cov(**b**)]**L**')(**Lb**-**c**)	TEST / PRINT

Table 64.5 *continued*

ODS Table Name	Description	Statement / Option
TestStmts	Linear hypotheses testing results	TEST
Type3	Type 3 chi-square tests	CLASS

Table 64.6 ODS Table for a Bayesian Analysis Produced by PROC PHREG

ODS Table Name	Description	Statement / Option
AutoCorr	Autocorrelations of the posterior samples	BAYES
CensoredSummary	Numbers of the event and censored observations	PROC
ClassLevelFreq	Frequency distribution of CLASS variables	CLASS, PROC / SIMPLE
ClassLevelInfo	CLASS variable levels and design variables	CLASS
CoeffPrior	Prior distribution of the regression coefficients	BAYES
Corr	Posterior correlation matrix	BAYES / SUMMARY=CORR
Cov	Posterior covariance Matrix	BAYES / SUMMARY=COV
ESS	Effective sample sizes	BAYES / DIAGNOSTICS=ESS
FitStatistics	Fit statistics	BAYES
Gelman	Gelman-Rubin convergence diagnostics	BAYES / DIAGNOSTICS=GELMAN
Geweke	Geweke convergence diagnostics	BAYES
HazardPrior	Prior distribution of the baseline hazards	BAYES / PIECEWISE
HazardRatios	Posterior summary statistics for hazard ratios	HAZARDRATIO
Heidelberger	Heidelberger-Welch convergence diagnostics	BAYES / DIAGNOSTICS=HEIDELBERGER
InitialValues	Initial values of the Markov chains	BAYES
ModelInfo	Model information	default
NObs	Number of observations	default
MCError	Monte Carlo standard errors	BAYES / DIAGNOSTICS=MCERROR
ParameterEstimates	Maximum likelihood estimates of model parameters	default
ParmInfo	Names of regression coefficients	CLASS,BAYES
Partition	Partition of constant baseline hazard intervals	BAYES / PIECEWISE
PostIntervals	equal-tail and high probability density intervals of the posterior samples	BAYES
PosteriorSample	Posterior samples	BAYES / (for ODS output data set only)

Table 64.6 *continued*

ODS Table Name	Description	Statement / Option
PostSummaries	Summary statistics of the posterior samples	BAYES
Raftery	Raftery-Lewis convergence diagnostics	BAYES / DIAGNOSTICS=RAFTERY
ReferenceSet	Reference set of covariates for plotting	PROC / PLOTS=
SimpleStatistics	Summary statistics of input continuous explanatory variables	PROC / SIMPLE

ODS Graphics

To request graphics with PROC PHREG, you must first enable ODS Graphics by specifying the **ods graphics on** statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC PHREG generates are listed separately in Table 64.7 for the maximum likelihood analysis and in Table 64.8 for the Bayesian analysis.

Table 64.7 ODS Graphics for a Maximum Likelihood Analysis Produced by PROC PHREG

ODS Graph Name	Plot Description	Statement / Option
CumhazPlot	Cumulative hazard function plot	PROC / PLOTS=CUMHAZ
CumulativeResiduals	Cumulative martingale residual plot	ASSESS / VAR=
CumResidPanel	Panel plot of cumulative martingale residuals	ASSESS / VAR=, CRPANEL
MCFPlot	Mean cumulative function plot	PROC / PLOTS=MCF
ScoreProcess	Standardized score process plot	ASSESS / PH
SurvivalPlot	Survivor function plot	PROC / PLOTS=SURVIVAL

Table 64.8 ODS Graphics for a Bayesian Analysis Produced by PROC PHREG

ODS Graph Name	Plot Description	Statement / Option
ADPanel	Autocorrelation function and density panel	BAYES / PLOTS=(AUTOCORR DENSITY)
AutocorrPanel	Autocorrelation function panel	BAYES / PLOTS= AUTOCORR
AutocorrPlot	Autocorrelation function plot	BAYES / PLOTS(UNPACK)=AUTOCORR

Table 64.8 *continued*

ODS Graph Name	Plot Description	Statement / Option
CumhazPlot	Cumulative hazard function plot	PROC / PLOTS=CUMHAZ
DensityPanel	Density panel	BAYES / PLOTS=DENSITY
DensityPlot	Density plot	BAYES / PLOTS(UNPACK)=DENSITY
SurvivalPlot	Survivor function plot	PROC / PLOTS=SURVIVAL
TAPanel	Trace and autocorrelation function panel	BAYES / PLOTS=(TRACE AUTOCORR)
TADPanel	Trace, density, and autocorrelation function panel	BAYES / PLOTS=(TRACE AUTOCORR DENSITY)
TDPanel	Trace and density panel	BAYES / PLOTS=(TRACE DENSITY)
TracePanel	Trace panel	BAYES / PLOTS=TRACE
TracePlot	Trace plot	BAYES / PLOTS(UNPACK)=TRACE

Examples: PHREG Procedure

This section contains 14 examples of PROC PHREG applications. The first 12 examples use the classical method of maximum likelihood, while the last two examples illustrate the Bayesian methodology.

Example 64.1: Stepwise Regression

Krall, Uthoff, and Harley (1975) analyzed data from a study on multiple myeloma in which researchers treated 65 patients with alkylating agents. Of those patients, 48 died during the study and 17 survived. The following DATA step creates the data set Myeloma. The variable Time represents the survival time in months from diagnosis. The variable VStatus consists of two values, 0 and 1, indicating whether the patient was alive or dead, respectively, at the end of the study. If the value of VStatus is 0, the corresponding value of Time is censored. The variables thought to be related to survival are LogBUN (log(BUN) at diagnosis), HGB (hemoglobin at diagnosis), Platelet (platelets at diagnosis: 0=abnormal, 1=normal), Age (age at diagnosis, in years), LogWBC (log(WBC) at diagnosis), Frac (fractures at diagnosis: 0=none, 1=present), LogPBM (log percentage of plasma cells in bone marrow), Protein (proteinuria at diagnosis), and SCalc (serum calcium at diagnosis). Interest lies in identifying important prognostic factors from these nine explanatory variables.

```
data Myeloma;
   input Time VStatus LogBUN HGB Platelet Age LogWBC Frac
         LogPBM Protein SCalc;
   label Time='Survival Time'
         VStatus='0=Alive 1=Dead';
   datalines;
 1.25  1  2.2175   9.4  1  67  3.6628  1  1.9542  12  10
```

```
 1.25  1  1.9395  12.0  1  38  3.9868  1  1.9542  20  18
 2.00  1  1.5185   9.8  1  81  3.8751  1  2.0000   2  15
 2.00  1  1.7482  11.3  0  75  3.8062  1  1.2553   0  12
 2.00  1  1.3010   5.1  0  57  3.7243  1  2.0000   3   9
 3.00  1  1.5441   6.7  1  46  4.4757  0  1.9345  12  10
 5.00  1  2.2355  10.1  1  50  4.9542  1  1.6628   4   9
 5.00  1  1.6812   6.5  1  74  3.7324  0  1.7324   5   9
 6.00  1  1.3617   9.0  1  77  3.5441  0  1.4624   1   8
 6.00  1  2.1139  10.2  0  70  3.5441  1  1.3617   1   8
 6.00  1  1.1139   9.7  1  60  3.5185  1  1.3979   0  10
 6.00  1  1.4150  10.4  1  67  3.9294  1  1.6902   0   8
 7.00  1  1.9777   9.5  1  48  3.3617  1  1.5682   5  10
 7.00  1  1.0414   5.1  0  61  3.7324  1  2.0000   1  10
 7.00  1  1.1761  11.4  1  53  3.7243  1  1.5185   1  13
 9.00  1  1.7243   8.2  1  55  3.7993  1  1.7404   0  12
11.00  1  1.1139  14.0  1  61  3.8808  1  1.2788   0  10
11.00  1  1.2304  12.0  1  43  3.7709  1  1.1761   1   9
11.00  1  1.3010  13.2  1  65  3.7993  1  1.8195   1  10
11.00  1  1.5682   7.5  1  70  3.8865  0  1.6721   0  12
11.00  1  1.0792   9.6  1  51  3.5051  1  1.9031   0   9
13.00  1  0.7782   5.5  0  60  3.5798  1  1.3979   2  10
14.00  1  1.3979  14.6  1  66  3.7243  1  1.2553   2  10
15.00  1  1.6021  10.6  1  70  3.6902  1  1.4314   0  11
16.00  1  1.3424   9.0  1  48  3.9345  1  2.0000   0  10
16.00  1  1.3222   8.8  1  62  3.6990  1  0.6990  17  10
17.00  1  1.2304  10.0  1  53  3.8808  1  1.4472   4   9
17.00  1  1.5911  11.2  1  68  3.4314  0  1.6128   1  10
18.00  1  1.4472   7.5  1  65  3.5682  0  0.9031   7   8
19.00  1  1.0792  14.4  1  51  3.9191  1  2.0000   6  15
19.00  1  1.2553   7.5  0  60  3.7924  1  1.9294   5   9
24.00  1  1.3010  14.6  1  56  4.0899  1  0.4771   0   9
25.00  1  1.0000  12.4  1  67  3.8195  1  1.6435   0  10
26.00  1  1.2304  11.2  1  49  3.6021  1  2.0000  27  11
32.00  1  1.3222  10.6  1  46  3.6990  1  1.6335   1   9
35.00  1  1.1139   7.0  0  48  3.6532  1  1.1761   4  10
37.00  1  1.6021  11.0  1  63  3.9542  0  1.2041   7   9
41.00  1  1.0000  10.2  1  69  3.4771  1  1.4771   6  10
41.00  1  1.1461   5.0  1  70  3.5185  1  1.3424   0   9
51.00  1  1.5682   7.7  0  74  3.4150  1  1.0414   4  13
52.00  1  1.0000  10.1  1  60  3.8573  1  1.6532   4  10
54.00  1  1.2553   9.0  1  49  3.7243  1  1.6990   2  10
58.00  1  1.2041  12.1  1  42  3.6990  1  1.5798  22  10
66.00  1  1.4472   6.6  1  59  3.7853  1  1.8195   0   9
67.00  1  1.3222  12.8  1  52  3.6435  1  1.0414   1  10
88.00  1  1.1761  10.6  1  47  3.5563  0  1.7559  21   9
89.00  1  1.3222  14.0  1  63  3.6532  1  1.6232   1   9
92.00  1  1.4314  11.0  1  58  4.0755  1  1.4150   4  11
 4.00  0  1.9542  10.2  1  59  4.0453  0  0.7782  12  10
 4.00  0  1.9243  10.0  1  49  3.9590  0  1.6232   0  13
 7.00  0  1.1139  12.4  1  48  3.7993  1  1.8573   0  10
 7.00  0  1.5315  10.2  1  81  3.5911  0  1.8808   0  11
 8.00  0  1.0792   9.9  1  57  3.8325  1  1.6532   0   8
12.00  0  1.1461  11.6  1  46  3.6435  0  1.1461   0   7
11.00  0  1.6128  14.0  1  60  3.7324  1  1.8451   3   9
```

```
12.00  0  1.3979   8.8  1  66  3.8388  1  1.3617   0   9
13.00  0  1.6628   4.9  0  71  3.6435  0  1.7924   0   9
16.00  0  1.1461  13.0  1  55  3.8573  0  0.9031   0   9
19.00  0  1.3222  13.0  1  59  3.7709  1  2.0000   1  10
19.00  0  1.3222  10.8  1  69  3.8808  1  1.5185   0  10
28.00  0  1.2304   7.3  1  82  3.7482  1  1.6721   0   9
41.00  0  1.7559  12.8  1  72  3.7243  1  1.4472   1   9
53.00  0  1.1139  12.0  1  66  3.6128  1  2.0000   1  11
57.00  0  1.2553  12.5  1  66  3.9685  0  1.9542   0  11
77.00  0  1.0792  14.0  1  60  3.6812  0  0.9542   0  12
;
```

The stepwise selection process consists of a series of alternating forward selection and backward elimination steps. The former adds variables to the model, while the latter removes variables from the model.

The following statements use PROC PHREG to produce a stepwise regression analyis. Stepwise selection is requested by specifying the SELECTION=STEPWISE option in the MODEL statement. The option SLENTRY=0.25 specifies that a variable has to be significant at the 0.25 level before it can be entered into the model, while the option SLSTAY=0.15 specifies that a variable in the model has to be significant at the 0.15 level for it to remain in the model. The DETAILS option requests detailed results for the variable selection process.

```
proc phreg data=Myeloma;
   model Time*VStatus(0)=LogBUN HGB Platelet Age LogWBC
                         Frac LogPBM Protein SCalc
                         / selection=stepwise slentry=0.25
                           slstay=0.15 details;
run;
```

Results of the stepwise regression analysis are displayed in Output 64.1.1 through Output 64.1.7.

Individual score tests are used to determine which of the nine explanatory variables is first selected into the model. In this case, the score test for each variable is the global score test for the model containing that variable as the only explanatory variable. Output 64.1.1 displays the chi-square statistics and the corresponding p-values. The variable LogBUN has the largest chi-square value (8.5164), and it is significant (p=0.0035) at the SLENTRY=0.25 level. The variable LogBUN is thus entered into the model.

Output 64.1.1 Individual Score Test Results for All Variables

```
                          The PHREG Procedure

                           Model Information

          Data Set                 WORK.MYELOMA
          Dependent Variable       Time             Survival Time
          Censoring Variable       VStatus          0=Alive 1=Dead
          Censoring Value(s)       0
          Ties Handling            BRESLOW
```

Output 64.1.1 *continued*

Summary of the Number of Event and Censored Values

Total	Event	Censored	Percent Censored
65	48	17	26.15

Analysis of Effects Eligible for Entry

Effect	DF	Score Chi-Square	Pr > ChiSq
LogBUN	1	8.5164	0.0035
HGB	1	5.0664	0.0244
Platelet	1	3.1816	0.0745
Age	1	0.0183	0.8924
LogWBC	1	0.5658	0.4519
Frac	1	0.9151	0.3388
LogPBM	1	0.5846	0.4445
Protein	1	0.1466	0.7018
SCalc	1	1.1109	0.2919

Residual Chi-Square Test

Chi-Square	DF	Pr > ChiSq
18.4550	9	0.0302

Output 64.1.2 displays the results of the first model. Since the Wald chi-square statistic is significant ($p = 0.0039$) at the SLSTAY=0.15 level, LogBUN stays in the model.

Output 64.1.2 First Model in the Stepwise Selection Process

Step 1. Effect LogBUN is entered. The model contains the following effects:

LogBUN

Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Without Covariates	With Covariates
-2 LOG L	309.716	301.959
AIC	309.716	303.959
SBC	309.716	305.830

Output 64.1.2 *continued*

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	7.7572	1	0.0053
Score	8.5164	1	0.0035
Wald	8.3392	1	0.0039

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
LogBUN	1	1.74595	0.60460	8.3392	0.0039	5.731

The next step consists of selecting another variable to add to the model. Output 64.1.3 displays the chi-square statistics and p-values of individual score tests (adjusted for LogBUN) for the remaining eight variables. The score chi-square for a given variable is the value of the likelihood score test for testing the significance of the variable in the presence of LogBUN. The variable HGB is selected because it has the highest chi-square value (4.3468), and it is significant ($p = 0.0371$) at the SLENTRY=0.25 level.

Output 64.1.3 Score Tests Adjusted for the Variable LogBUN

Analysis of Effects Eligible for Entry

Effect	DF	Score Chi-Square	Pr > ChiSq
HGB	1	4.3468	0.0371
Platelet	1	2.0183	0.1554
Age	1	0.7159	0.3975
LogWBC	1	0.0704	0.7908
Frac	1	1.0354	0.3089
LogPBM	1	1.0334	0.3094
Protein	1	0.5214	0.4703
SCalc	1	1.4150	0.2342

Residual Chi-Square Test

Chi-Square	DF	Pr > ChiSq
9.3164	8	0.3163

Output 64.1.4 displays the fitted model containing both LogBUN and HGB. Based on the Wald statistics, neither LogBUN nor HGB is removed from the model.

Output 64.1.4 Second Model in the Stepwise Selection Process

Step 2. Effect HGB is entered. The model contains the following effects:

LogBUN HGB

Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Without Covariates	With Covariates
-2 LOG L	309.716	297.767
AIC	309.716	301.767
SBC	309.716	305.509

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	11.9493	2	0.0025
Score	12.7252	2	0.0017
Wald	12.1900	2	0.0023

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
LogBUN	1	1.67440	0.61209	7.4833	0.0062	5.336
HGB	1	-0.11899	0.05751	4.2811	0.0385	0.888

Output 64.1.5 shows Step 3 of the selection process, in which the variable SCalc is added, resulting in the model with LogBUN, HGB, and SCalc as the explanatory variables. Note that SCalc has the smallest Wald chi-square statistic, and it is not significant ($p = 0.1782$) at the SLSTAY=0.15 level.

Output 64.1.5 Third Model in the Stepwise Regression

Step 3. Effect SCalc is entered. The model contains the following effects:

LogBUN HGB SCalc

Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Output 64.1.5 *continued*

Model Fit Statistics

Criterion	Without Covariates	With Covariates
-2 LOG L	309.716	296.078
AIC	309.716	302.078
SBC	309.716	307.692

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	13.6377	3	0.0034
Score	15.3053	3	0.0016
Wald	14.4542	3	0.0023

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
LogBUN	1	1.63593	0.62359	6.8822	0.0087	5.134
HGB	1	-0.12643	0.05868	4.6419	0.0312	0.881
SCalc	1	0.13286	0.09868	1.8127	0.1782	1.142

The variable SCalc is then removed from the model in a step-down phase in Step 4 (Output 64.1.6). The removal of SCalc brings the stepwise selection process to a stop in order to avoid repeatedly entering and removing the same variable.

Output 64.1.6 Final Model in the Stepwise Regression

Step 4. Effect SCalc is removed. The model contains the following effects:

LogBUN HGB

Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Without Covariates	With Covariates
-2 LOG L	309.716	297.767
AIC	309.716	301.767
SBC	309.716	305.509

Output 64.1.6 *continued*

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	11.9493	2	0.0025
Score	12.7252	2	0.0017
Wald	12.1900	2	0.0023

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
LogBUN	1	1.67440	0.61209	7.4833	0.0062	5.336
HGB	1	-0.11899	0.05751	4.2811	0.0385	0.888

NOTE: Model building terminates because the effect to be entered is the effect that was removed in the last step.

The procedure also displays a summary table of the steps in the stepwise selection process, as shown in Output 64.1.7.

Output 64.1.7 Model Selection Summary

Summary of Stepwise Selection

Step	Effect Entered	Effect Removed	DF	Number In	Score Chi-Square	Wald Chi-Square	Pr > ChiSq
1	LogBUN		1	1	8.5164		0.0035
2	HGB		1	2	4.3468		0.0371
3	SCalc		1	3	1.8225		0.1770
4		SCalc	1	2		1.8127	0.1782

The stepwise selection process results in a model with two explanatory variables, LogBUN and HGB.

Example 64.2: Best Subset Selection

An alternative to stepwise selection of variables is best subset selection. This method uses the branch-and-bound algorithm of Furnival and Wilson (1974) to find a specified number of best models containing one, two, or three variables, and so on, up to the single model containing all of the explanatory variables. The criterion used to determine the "best" subset is based on the global score chi-square statistic. For two models A and B, each having the same number of explanatory variables, model A is considered to be better than model B if the global score chi-square statistic for A exceeds that for B.

In the following statements, best subset selection analysis is requested by specifying the SELECTION=SCORE option in the MODEL statement. The BEST=3 option requests the procedure to identify only the three best models for each size. In other words, PROC PHREG will list the three models having the highest score statistics of all the models possible for a given number of covariates.

```
proc phreg data=Myeloma;
   model Time*VStatus(0)=LogBUN HGB Platelet Age LogWBC
                         Frac LogPBM Protein SCalc
                         / selection=score best=3;
run;
```

Output 64.2.1 displays the results of this analysis. The number of explanatory variables in the model is given in the first column, and the names of the variables are listed on the right. The models are listed in descending order of their score chi-square values within each model size. For example, among all models containing two explanatory variables, the model that contains the variables LogBUN and HGB has the largest score value (12.7252), the model that contains the variables LogBUN and Platelet has the second-largest score value (11.1842), and the model that contains the variables LogBUN and SCalc has the third-largest score value (9.9962).

Output 64.2.1 Best Variable Combinations

The PHREG Procedure

Regression Models Selected by Score Criterion

Number of Variables	Score Chi-Square	Variables Included in Model
1	8.5164	LogBUN
1	5.0664	HGB
1	3.1816	Platelet
2	12.7252	LogBUN HGB
2	11.1842	LogBUN Platelet
2	9.9962	LogBUN SCalc
3	15.3053	LogBUN HGB SCalc
3	13.9911	LogBUN HGB Age
3	13.5788	LogBUN HGB Frac
4	16.9873	LogBUN HGB Age SCalc
4	16.0457	LogBUN HGB Frac SCalc
4	15.7619	LogBUN HGB LogPBM SCalc
5	17.6291	LogBUN HGB Age Frac SCalc
5	17.3519	LogBUN HGB Age LogPBM SCalc
5	17.1922	LogBUN HGB Age LogWBC SCalc
6	17.9120	LogBUN HGB Age Frac LogPBM SCalc
6	17.7947	LogBUN HGB Age LogWBC Frac SCalc
6	17.7744	LogBUN HGB Platelet Age Frac SCalc
7	18.1517	LogBUN HGB Platelet Age Frac LogPBM SCalc
7	18.0568	LogBUN HGB Age LogWBC Frac LogPBM SCalc
7	18.0223	LogBUN HGB Platelet Age LogWBC Frac SCalc
8	18.3925	LogBUN HGB Platelet Age LogWBC Frac LogPBM SCalc
8	18.1636	LogBUN HGB Platelet Age Frac LogPBM Protein SCalc
8	18.1309	LogBUN HGB Platelet Age LogWBC Frac Protein SCalc
9	18.4550	LogBUN HGB Platelet Age LogWBC Frac LogPBM Protein SCalc

Example 64.3: Modeling with Categorical Predictors

Consider the data for the Veterans Administration lung cancer trial presented in Appendix 1 of Kalbfleisch and Prentice (1980). In this trial, males with advanced inoperable lung cancer were randomized to a standard therapy and a test chemotherapy. The primary endpoint for the therapy comparison was time to death in days, represented by the variable Time. Negative values of Time are censored values. The data include information about a number of explanatory variables: Therapy (type of therapy: standard or test), Cell (type of tumor cell: adeno, large, small, or squamous), Prior (prior therapy: 0=no, 10=yes), Age (age, in years), Duration (months from diagnosis to randomization), and Kps (Karnofsky performance scale). A censoring indicator variable, Censor, is created from the data, with the value 1 indicating a censored time and the value 0 indicating an event time. The following DATA step saves the data in the data set VALung.

```
proc format;
   value yesno 0='no' 10='yes';
   run;

data VALung;
   drop check m;
   retain Therapy Cell;
   infile cards column=column;
   length Check $ 1;
   label Time='time to death in days'
         Kps='Karnofsky performance scale'
         Duration='months from diagnosis to randomization'
         Age='age in years'
         Prior='prior therapy'
         Cell='cell type'
         Therapy='type of treatment';
   format Prior yesno.;
   M=Column;
   input Check $ @@;
   if M>Column then M=1;
   if Check='s'|Check='t' then do;
      input @M Therapy $ Cell $;
      delete;
      end;
   else do;
      input @M Time Kps Duration Age Prior @@;
      Status=(Time>0);
      Time=abs(Time);
      end;
   datalines;
standard squamous
 72 60  7 69  0   411 70  5 64 10   228 60  3 38  0   126 60  9 63 10
118 70 11 65 10    10 20  5 49  0    82 40 10 69 10   110 80 29 68  0
314 50 18 43  0  -100 70  6 70  0    42 60  4 81  0     8 40 58 63 10
144 30  4 63  0   -25 80  9 52 10    11 70 11 48 10
standard small
 30 60  3 61  0   384 60  9 42  0     4 40  2 35  0    54 80  4 63 10
 13 60  4 56  0  -123 40  3 55  0   -97 60  5 67  0   153 60 14 63 10
```

```
 59 30  2 65  0   117 80  3 46  0    16 30  4 53 10   151 50 12 69  0
 22 60  4 68  0    56 80 12 43 10    21 40  2 55 10    18 20 15 42  0
139 80  2 64  0    20 30  5 65  0    31 75  3 65  0    52 70  2 55  0
287 60 25 66 10    18 30  4 60  0    51 60  1 67  0   122 80 28 53  0
 27 60  8 62  0    54 70  1 67  0     7 50  7 72  0    63 50 11 48  0
392 40  4 68  0    10 40 23 67 10
standard adeno
  8 20 19 61 10    92 70 10 60  0    35 40  6 62  0   117 80  2 38  0
132 80  5 50  0    12 50  4 63 10   162 80  5 64  0     3 30  3 43  0
 95 80  4 34  0
standard large
177 50 16 66 10   162 80  5 62  0   216 50 15 52  0   553 70  2 47  0
278 60 12 63  0    12 40 12 68 10   260 80  5 45  0   200 80 12 41 10
156 70  2 66  0  -182 90  2 62  0   143 90  8 60  0   105 80 11 66  0
103 80  5 38  0   250 70  8 53 10   100 60 13 37 10
test squamous
999 90 12 54 10   112 80  6 60  0   -87 80  3 48  0  -231 50  8 52 10
242 50  1 70  0   991 70  7 50 10   111 70  3 62  0     1 20 21 65 10
587 60  3 58  0   389 90  2 62  0    33 30  6 64  0    25 20 36 63  0
357 70 13 58  0   467 90  2 64  0   201 80 28 52 10     1 50  7 35  0
 30 70 11 63  0    44 60 13 70 10   283 90  2 51  0    15 50 13 40 10
test small
 25 30  2 69  0  -103 70 22 36 10    21 20  4 71  0    13 30  2 62  0
 87 60  2 60  0     2 40 36 44 10    20 30  9 54 10     7 20 11 66  0
 24 60  8 49  0    99 70  3 72  0     8 80  2 68  0    99 85  4 62  0
 61 70  2 71  0    25 70  2 70  0    95 70  1 61  0    80 50 17 71  0
 51 30 87 59 10    29 40  8 67  0
test adeno
 24 40  2 60  0    18 40  5 69 10   -83 99  3 57  0    31 80  3 39  0
 51 60  5 62  0    90 60 22 50 10    52 60  3 43  0    73 60  3 70  0
  8 50  5 66  0    36 70  8 61  0    48 10  4 81  0     7 40  4 58  0
140 70  3 63  0   186 90  3 60  0    84 80  4 62 10    19 50 10 42  0
 45 40  3 69  0    80 40  4 63  0
test large
 52 60  4 45  0   164 70 15 68 10    19 30  4 39 10    53 60 12 66  0
 15 30  5 63  0    43 60 11 49 10   340 80 10 64 10   133 75  1 65  0
111 60  5 64  0   231 70 18 67 10   378 80  4 65  0    49 30  3 37  0
;
```

The following statements use the PHREG procedure to fit the Cox proportional hazards model to these data. The variables Prior, Cell, and Therapy, which are categorical variables, are declared in the CLASS statement. By default, PROC PHREG parameterizes the CLASS variables by using the reference coding with the last category as the reference category. However, you can explicitly specify the reference category of your choice. Here, Prior=no is chosen as the reference category for prior therapy, Cell=large is chosen as the reference category for type of tumor cell, and Therapy=standard is chosen as the reference category for the type of therapy. In the MODEL statement, the term Prior|Therapy is just another way of specifying the main effects Prior, Therapy, and the Prior*Therapy interaction.

```
proc phreg data=VALung;
   class Prior(ref='no') Cell(ref='large') Therapy(ref='standard');
   model Time*Status(0) = Kps Duration Age Cell Prior|Therapy;
   run;
```

Coding of the CLASS variables is displayed in Output 64.3.1. There is one dummy variable for Prior and one for Therapy, since both variables are binary. The dummy variable has a value of 0 for the reference category (Prior=no, Therapy=standard). The variable Cell has four categories and is represented by three dummy variables. Note that the reference category, Cell=large, has a value of 0 for all three dummy variables.

Output 64.3.1 Reference Coding of CLASS Variables

The PHREG Procedure

Class Level Information

Class	Value	Design Variables		
Prior	no	0		
	yes	1		
Cell	adeno	1	0	0
	large	0	0	0
	small	0	1	0
	squamous	0	0	1
Therapy	standard	0		
	test	1		

The test results of individual model effects are shown in Output 64.3.2. There is a strong prognostic effect of Kps on patient's survivivorship ($p < 0.0001$), and the survival times for patients of different Cell types differ significantly ($p = 0.0003$). The Prior*Therapy interaction is marginally significant (p=0.0416)—that is, prior therapy might play a role in whether one treatment is more effective than the other.

Output 64.3.2 Wald Tests of Individual Effects

Type 3 Tests

Effect	DF	Wald Chi-Square	Pr > ChiSq
Kps	1	35.5051	<.0001
Duration	1	0.1159	0.7335
Age	1	1.9772	0.1597
Cell	3	18.5339	0.0003
Prior	1	2.5296	0.1117
Therapy	1	5.2349	0.0221
Prior*Therapy	1	4.1528	0.0416

In the Cox proportional hazards model, the effects of the covariates are to act multiplicatively on the hazard of the survival time, and therefore it is a little easier to interpret the corresponding hazard ratios than the regression parameters. For a parameter that corresponds to a continuous variable, the hazard ratio is the ratio of hazard rates for a increase of one unit of the variable. From Output 64.3.3, the hazard ratio estimate for Kps is 0.968, meaning that an increase of 10 units in

Karnofsky performance scale will shrink the hazard rate by $1-(0.968)^{10}$=28%. For a CLASS variable parameter, the hazard ratio presented in the Output 64.3.3 is the ratio of the hazard rates between the given category and the reference category. The hazard rate of Cell=adeno is 219% that of Cell=large, the hazard rate of Cell=small is 62% that of Cell=large, and the hazard rate of Cell=squamous is only 66% that of Cell=large. Hazard ratios for Prior and Therapy are missing since the model contains the Prior*Therapy interaction. You can use the HAZARDRATIO statement to obtain the hazard ratios for a main effect in the presence of interaction as shown later in this example.

Output 64.3.3 Parameters Estimates with Reference Coding

Analysis of Maximum Likelihood Estimates

Parameter			DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq
Kps			1	-0.03300	0.00554	35.5051	<.0001
Duration			1	0.00323	0.00949	0.1159	0.7335
Age			1	-0.01353	0.00962	1.9772	0.1597
Cell	adeno		1	0.78356	0.30382	6.6512	0.0099
Cell	small		1	0.48230	0.26537	3.3032	0.0691
Cell	squamous		1	-0.40770	0.28363	2.0663	0.1506
Prior	yes		1	0.45914	0.28868	2.5296	0.1117
Therapy	test		1	0.56662	0.24765	5.2349	0.0221
Prior*Therapy	yes	test	1	-0.87579	0.42976	4.1528	0.0416

Analysis of Maximum Likelihood Estimates

Parameter			Hazard Ratio
Kps			0.968
Duration			1.003
Age			0.987
Cell	adeno		2.189
Cell	small		1.620
Cell	squamous		0.665
Prior	yes		.
Therapy	test		.
Prior*Therapy	yes	test	.

Analysis of Maximum Likelihood Estimates

Parameter			Label
Kps			Karnofsky performance scale
Duration			months from diagnosis to randomization
Age			age in years
Cell	adeno		cell type adeno
Cell	small		cell type small
Cell	squamous		cell type squamous
Prior	yes		prior therapy yes
Therapy	test		type of treatment test
Prior*Therapy	yes	test	prior therapy yes * type of treatment test

The following PROC PHREG statements illustrate the use of the backward elimination process to identify the effects that affect the survivorship of the lung cancer patients. The option SELECTION=BACKWARD is specified to carry out the backward elimination. The option SLSTAY=0.1 specifies the significant level for retaining the effects in the model.

```
proc phreg data=VALung;
   class Prior(ref='no') Cell(ref='large') Therapy(ref='standard');
   model Time*Status(0) = Kps Duration Age Cell Prior|Therapy / selection=backward
                                                               slstay=0.1;
   run;
```

Results of the backward elimination process are summarized in Output 64.3.4. The effect Duration was eliminated first and was followed by Age.

Output 64.3.4 Effects Eliminated from the Model

The PHREG Procedure

Summary of Backward Elimination

Step	Effect Removed	DF	Number In	Wald Chi-Square	Pr > ChiSq
1	Duration	1	6	0.1159	0.7335
2	Age	1	5	2.0458	0.1526

Summary of Backward Elimination

Step	Effect Label
1	months from diagnosis to randomization
2	age in years

Output 64.3.5 shows the Type 3 analysis of effects and the maximum likelihood estimates of the regression coefficients of the model. Without controlling for Age and Duration, KPS and Cell remain significant, but the Prior*Therapy interaction is less prominent than before (p=0.0871) though still significant at 0.1 level.

Output 64.3.5 Type 3 Effects and Parameter Estimates for the Selected Model

Type 3 Tests

Effect	DF	Wald Chi-Square	Pr > ChiSq
Kps	1	35.9218	<.0001
Cell	3	17.4134	0.0006
Prior	1	2.3113	0.1284
Therapy	1	3.8030	0.0512
Prior*Therapy	1	2.9269	0.0871

Output 64.3.5 *continued*

Analysis of Maximum Likelihood Estimates

Parameter			DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq
Kps			1	-0.03111	0.00519	35.9218	<.0001
Cell	adeno		1	0.74907	0.30465	6.0457	0.0139
Cell	small		1	0.44265	0.26168	2.8614	0.0907
Cell	squamous		1	-0.41145	0.28309	2.1125	0.1461
Prior	yes		1	0.41755	0.27465	2.3113	0.1284
Therapy	test		1	0.45670	0.23419	3.8030	0.0512
Prior*Therapy	yes	test	1	-0.69443	0.40590	2.9269	0.0871

Analysis of Maximum Likelihood Estimates

Parameter			Hazard Ratio
Kps			0.969
Cell	adeno		2.115
Cell	small		1.557
Cell	squamous		0.663
Prior	yes		.
Therapy	test		.
Prior*Therapy	yes	test	.

Analysis of Maximum Likelihood Estimates

Parameter			Label
Kps			Karnofsky performance scale
Cell	adeno		cell type adeno
Cell	small		cell type small
Cell	squamous		cell type squamous
Prior	yes		prior therapy yes
Therapy	test		type of treatment test
Prior*Therapy	yes	test	prior therapy yes * type of treatment test

Finally, the following statements refit the previous model and computes hazard ratios at settings beyond those displayed in the "Analysis of Maximum Likelihood Estimates" table. You can use either the HAZARDRATIO statement or the CONTRAST statement to obtain hazard ratios. Using the CONTRAST statement to compute hazard ratios for CLASS variables can be a daunting task unless you are familiar with the parameterization schemes (see the section "CLASS Variable Parameterization" on page 4580 for more information), but you have control over which specific hazard ratios you want to compute. HAZARDRATIO statements, on the other hand, are designed specifically to provide hazard ratios. They are easy to use and you can also request both the Wald confidence limits and the profile-likelihood confidence limits; the latter is not available for the CONTRAST statements. Three HAZARDRATIO statements are specified; each has the CL=BOTH option to request both the Wald confidence limits and the profile-likelihood limits. The first HAZARDRATIO statement, labeled 'H1', estimates the hazard ratio for an increase of 10 units in the KPS; the UNITS= option specifies the number of units increase. The second HAZARDRATIO statement, labeled 'H2' computes the hazard ratios for comparing any pairs of tumor Cell types. The third

HAZARDRATIO statement, labeled 'H3', compares the test therapy with the standard therapy. The DIFF=REF option specifies that each nonreference category is compared to the reference category. The purpose of using DIFF=REF here is to ensure that the hazard ratio is comparing the test therapy to the standard therapy instead of the other way around. Three CONTRAST statements, labeled 'C1', 'C2', and 'C3', parallel to the HAZARDRATIO statements 'H1', 'H2', and 'H3', respectively, are specified. The ESTIMATE=EXP option specifies that the linear predictors be estimated in the exponential scale, which are precisely the hazard ratios.

```
proc phreg data=VALung;
   class Prior(ref='no') Cell(ref='large') Therapy(ref='standard');
   model Time*Status(0) = Kps Cell Prior|Therapy;
   hazardratio 'H1' Kps / units=10 cl=both;
   hazardratio 'H2' Cell / cl=both;
   hazardratio 'H3' Therapy / diff=ref cl=both;
   contrast 'C1' Kps 10 / estimate=exp;
   contrast 'C2' cell 1  0  0, /* adeno vs large    */
                 cell 1 -1  0, /* adeno vs small    */
                 cell 1  0 -1, /* adeno vs squamous */
                 cell 0 -1  0, /* large vs small    */
                 cell 0  0 -1, /* large vs Squamous */
                 cell 0  1 -1  /* small vs squamous */
                  / estimate=exp;
   contrast 'C3' Prior 0 Therapy 1  Prior*Therapy 0,
                 Prior 0 Therapy 1  Prior*Therapy 1  / estimate=exp;
   run;
```

Output 64.3.6 displays the results of the three HAZARDRATIO statements in separate tables. Results of the three CONTRAST statements are shown in one table in Output 64.3.7. However, point estimates and the Wald confidence limits for the hazard ratio agree in between the two outputs.

Output 64.3.6 Results from HAZARDRATIO Statements

The PHREG Procedure

H1: Hazard Ratios for Kps

Description	Point Estimate	95% Wald Confidence Limits		95% Profile Likelihood Confidence Limits	
Kps Unit=10	0.733	0.662	0.811	0.662	0.811

H2: Hazard Ratios for Cell

Description	Point Estimate	95% Wald Confidence Limits		95% Profile Likelihood Confidence Limits	
Cell adeno vs large	2.115	1.164	3.843	1.162	3.855
Cell adeno vs small	1.359	0.798	2.312	0.791	2.301
Cell adeno vs squamous	3.192	1.773	5.746	1.770	5.768
Cell large vs small	0.642	0.385	1.073	0.380	1.065
Cell large vs squamous	1.509	0.866	2.628	0.863	2.634
Cell small vs squamous	2.349	1.387	3.980	1.399	4.030

Output 64.3.6 *continued*

H3: Hazard Ratios for Therapy

Description	Point Estimate	95% Wald Confidence Limits	
Therapy test vs standard At Prior=no	1.579	0.998	2.499
Therapy test vs standard At Prior=yes	0.788	0.396	1.568

H3: Hazard Ratios for Therapy

95% Profile Likelihood Confidence Limits	
0.998	2.506
0.390	1.560

Output 64.3.7 Results from CONTRAST Statements

Contrast Rows Estimation and Testing Results

Contrast	Type	Row	Estimate	Standard Error	Alpha	Confidence Limits	
C1	EXP	1	0.7326	0.0380	0.05	0.6618	0.8111
C2	EXP	1	2.1150	0.6443	0.05	1.1641	3.8427
C2	EXP	2	1.3586	0.3686	0.05	0.7982	2.3122
C2	EXP	3	3.1916	0.9575	0.05	1.7727	5.7462
C2	EXP	4	0.6423	0.1681	0.05	0.3846	1.0728
C2	EXP	5	1.5090	0.4272	0.05	0.8664	2.6282
C2	EXP	6	2.3493	0.6318	0.05	1.3868	3.9797
C3	EXP	1	1.5789	0.3698	0.05	0.9977	2.4985
C3	EXP	2	0.7884	0.2766	0.05	0.3964	1.5680

Contrast Rows Estimation and Testing Results

Contrast	Type	Row	Wald Chi-Square	Pr > ChiSq
C1	EXP	1	35.9218	<.0001
C2	EXP	1	6.0457	0.0139
C2	EXP	2	1.2755	0.2587
C2	EXP	3	14.9629	0.0001
C2	EXP	4	2.8614	0.0907
C2	EXP	5	2.1125	0.1461
C2	EXP	6	10.0858	0.0015
C3	EXP	1	3.8030	0.0512
C3	EXP	2	0.4593	0.4980

Example 64.4: Firth's Correction for Monotone Likelihood

In fitting the Cox regression model by maximizing the partial likelihood, the estimate of an explanatory variable X will be infinite if the value of X at each uncensored failure time is the largest of all the values of X in the risk set at that time (Tsiatis 1981; Bryson and Johnson 1981). You can exploit this information to artificially create a data set that has the condition of monotone likelihood for the Cox regression. The following DATA step modifies the Myeloma data in Example 64.1 to create a new explanatory variable, Contrived, which has the value 1 if the observed time is less than or equal to 65 and has the value 0 otherwise. The phenomenon of monotone likelihood will be demonstrated in the new data set Myeloma2.

```
data Myeloma2;
   set Myeloma;
   Contrived= (Time <= 65);
   run;
```

For illustration purposes, consider a Cox model with three explanatory variables, one of which is the variable Contrived. The following statements invoke PROC PHREG to perform the Cox regression. The IPRINT option is specified in the MODEL statement to print the iteration history of the optimization.

```
proc phreg data=Myeloma2;
   model Time*Vstatus(0)=LOGbun HGB Contrived / itprint;
   run;
```

The symptom of monotonity is demonstrated in Output 64.4.1. The log likelihood converges to the value of –136.56 while the coefficient for Contrived diverges. Although the Newton-Raphson optimization process did not fail, it is obvious that convergence is questionable. A close examination of the standard errors in the "Analyis of Maximum Likelihood Estimates" table reveals a very large value for the coefficient of Contrived. This is very typical of a diverged estimate.

Output 64.4.1 Monotone Likelihood Behavior Displayed

The PHREG Procedure

Maximum Likelihood Iteration History

Iter	Ridge	Log Likelihood	LogBUN	HGB	Contrived
0	0	-154.8579914384	0.0000000000	0.000000000	0.000000000
1	0	-140.6934052686	1.9948819671	-0.084318519	1.466331269
2	0	-137.7841629416	1.6794678962	-0.109067888	2.778361123
3	0	-136.9711897754	1.7140611684	-0.111564202	3.938095086
4	0	-136.7078932606	1.7181735043	-0.112273248	5.003053568
5	0	-136.6164264879	1.7187547532	-0.112369756	6.027435769
6	0	-136.5835200895	1.7188294108	-0.112382079	7.036444978
7	0	-136.5715152788	1.7188392687	-0.112383700	8.039763533
8	0	-136.5671126045	1.7188405904	-0.112383917	9.040984886
9	0	-136.5654947987	1.7188407687	-0.112383947	10.041434266
10	0	-136.5648998913	1.7188407928	-0.112383950	11.041599592
11	0	-136.5646810709	1.7188407960	-0.112383951	12.041660414
12	0	-136.5646005760	1.7188407965	-0.112383951	13.041682789
13	0	-136.5645709642	1.7188407965	-0.112383951	14.041691020
14	0	-136.5645600707	1.7188407965	-0.112383951	15.041694049
15	0	-136.5645560632	1.7188407965	-0.112383951	16.041695162
16	0	-136.5645545889	1.7188407965	-0.112383951	17.041695572

Output 64.4.1 *continued*

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
LogBUN	1	1.71884	0.58376	8.6697	0.0032	5.578
HGB	1	-0.11238	0.06090	3.4053	0.0650	0.894
Contrived	1	17.04170	1080	0.0002	0.9874	25183399

Next, the Firth correction was applied as shown in the following statements. Also, the profile-likelihood confidence limits for the hazard ratios are requested by using the RISKLIMITS=PL option.

```
proc phreg data=Myeloma2;
   model Time*Vstatus(0)=LogBUN HGB Contrived /
      firth risklimits=pl itprint;
   run;
```

PROC PHREG uses the penalized likelihood maximum to obtain a finite estimate for the coefficient of Contrived (Output 64.4.2). The much preferred profile-likelihood confidence limits, as shown in (Heinze and Schemper 2001), are also displayed.

Output 64.4.2 Convergence Obtained with the Firth Correction

The PHREG Procedure

Maximum Likelihood Iteration History

Iter	Ridge	Log Likelihood	LogBUN	HGB	Contrived
0	0	-150.7361197494	0.0000000000	0.000000000	0.0000000000
1	0	-136.9933949142	2.0262484120	-0.086519138	1.4338859318
2	0	-134.5796594364	1.6770836974	-0.109172604	2.6221444778
3	0	-134.1572923217	1.7163408994	-0.111166227	3.4458043289
4	0	-134.1229607193	1.7209210332	-0.112007726	3.7923555412
5	0	-134.1228364805	1.7219588214	-0.112178328	3.8174197804
6	0	-134.1228355256	1.7220081673	-0.112187764	3.8151642206

Output 64.4.2 *continued*

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq
LogBUN	1	1.72201	0.58379	8.7008	0.0032
HGB	1	−0.11219	0.06059	3.4279	0.0641
Contrived	1	3.81516	1.55812	5.9955	0.0143

Analysis of Maximum Likelihood Estimates

Parameter	Hazard Ratio	95% Hazard Ratio Profile Likelihood Confidence Limits	
LogBUN	5.596	1.761	17.231
HGB	0.894	0.794	1.007
Contrived	45.384	5.406	6005.404

Example 64.5: Conditional Logistic Regression for m:n Matching

Conditional logistic regression is used to investigate the relationship between an outcome and a set of prognostic factors in matched case-control studies. The outcome is whether the subject is a case or a control. If there is only one case and one control, the matching is 1:1. The *m:n* matching refers to the situation in which there is a varying number of cases and controls in the matched sets. You can perform conditional logistic regression with the PHREG procedure by using the discrete logistic model and forming a stratum for each matched set. In addition, you need to create dummy survival times so that all the cases in a matched set have the same event time value, and the corresponding controls are censored at later times.

Consider the following set of low infant birth-weight data extracted from Appendix 1 of Hosmer and Lemeshow (1989). These data represent 189 women, of whom 59 had low-birth-weight babies and 130 had normal-weight babies. Under investigation are the following risk factors: weight in pounds at the last menstrual period (LWT), presence of hypertension (HT), smoking status during pregnancy (Smoke), and presence of uterine irritability (UI). For HT, Smoke, and UI, a value of 1 indicates a "yes" and a value of 0 indicates a "no." The woman's age (Age) is used as the matching variable. The SAS data set LBW contains a subset of the data corresponding to women between the ages of 16 and 32.

```
data LBW;
   input id Age Low LWT Smoke HT UI @@;
   Time=2-Low;
   datalines;
 25  16   1   130   0  0  0    143  16   0   110   0  0  0
166  16   0   112   0  0  0    167  16   0   135   1  0  0
189  16   0   135   1  0  0    206  16   0   170   0  0  0
216  16   0    95   0  0  0     37  17   1   130   1  0  1
```

```
 45  17   1  110   1  0  0     68  17   1  120   1  0  0
 71  17   1  120   0  0  0     83  17   1  142   0  1  0
 93  17   0  103   0  0  0    113  17   0  122   1  0  0
116  17   0  113   0  0  0    117  17   0  113   0  0  0
147  17   0  119   0  0  0    148  17   0  119   0  0  0
180  17   0  120   1  0  0     49  18   1  148   0  0  0
 50  18   1  110   1  0  0     89  18   0  107   1  0  1
100  18   0  100   1  0  0    101  18   0  100   1  0  0
132  18   0   90   1  0  1    133  18   0   90   1  0  1
168  18   0  229   0  0  0    205  18   0  120   1  0  0
208  18   0  120   0  0  0     23  19   1   91   1  0  1
 33  19   1  102   0  0  0     34  19   1  112   1  0  1
 85  19   0  182   0  0  1     96  19   0   95   0  0  0
 97  19   0  150   0  0  0    124  19   0  138   1  0  0
129  19   0  189   0  0  0    135  19   0  132   0  0  0
142  19   0  115   0  0  0    181  19   0  105   0  0  0
187  19   0  235   1  1  0    192  19   0  147   1  0  0
193  19   0  147   1  0  0    197  19   0  184   1  1  0
224  19   0  120   1  0  0     27  20   1  150   1  0  0
 31  20   1  125   0  0  1     40  20   1  120   1  0  0
 44  20   1   80   1  0  1     47  20   1  109   0  0  0
 51  20   1  121   1  0  1     60  20   1  122   1  0  0
 76  20   1  105   0  0  0     87  20   0  105   1  0  0
104  20   0  120   0  0  1    146  20   0  103   0  0  0
155  20   0  169   0  0  1    160  20   0  141   0  0  1
172  20   0  121   1  0  0    177  20   0  127   0  0  0
201  20   0  120   0  0  0    211  20   0  170   1  0  0
217  20   0  158   0  0  0     20  21   1  165   1  1  0
 28  21   1  200   0  0  1     30  21   1  103   0  0  0
 52  21   1  100   0  0  0     84  21   1  130   1  1  0
 88  21   0  108   1  0  1     91  21   0  124   0  0  0
128  21   0  185   1  0  0    131  21   0  160   0  0  0
144  21   0  110   1  0  1    186  21   0  134   0  0  0
219  21   0  115   0  0  0     42  22   1  130   1  0  1
 67  22   1  130   1  0  0     92  22   0  118   0  0  0
 98  22   0   95   0  1  0    137  22   0   85   1  0  0
138  22   0  120   0  1  0    140  22   0  130   1  0  0
161  22   0  158   0  0  0    162  22   0  112   1  0  0
174  22   0  131   0  0  0    184  22   0  125   0  0  0
204  22   0  169   0  0  0    220  22   0  129   0  0  0
 17  23   1   97   0  0  1     59  23   1  187   1  0  0
 63  23   1  120   0  0  0     69  23   1  110   1  0  0
 82  23   1   94   1  0  0    130  23   0  130   0  0  0
139  23   0  128   0  0  0    149  23   0  119   0  0  0
164  23   0  115   1  0  0    173  23   0  190   0  0  0
179  23   0  123   0  0  0    182  23   0  130   0  0  0
200  23   0  110   0  0  0     18  24   1  128   0  0  0
 19  24   1  132   0  1  0     29  24   1  155   1  0  0
 36  24   1  138   0  0  0     61  24   1  105   1  0  0
118  24   0   90   1  0  0    136  24   0  115   0  0  0
150  24   0  110   0  0  0    156  24   0  115   0  0  0
185  24   0  133   0  0  0    196  24   0  110   0  0  0
199  24   0  110   0  0  0    225  24   0  116   0  0  0
 13  25   1  105   0  1  0     15  25   1   85   0  0  1
```

```
 24  25   1   115   0  0  0     26  25   1    92   1  0  0
 32  25   1    89   0  0  0     46  25   1   105   0  0  0
103  25   0   118   1  0  0    111  25   0   120   0  0  1
120  25   0   155   0  0  0    121  25   0   125   0  0  0
169  25   0   140   0  0  0    188  25   0    95   1  0  1
202  25   0   241   0  1  0    215  25   0   120   0  0  0
221  25   0   130   0  0  0     35  26   1   117   1  0  0
 54  26   1    96   0  0  0     75  26   1   154   0  1  0
 77  26   1   190   1  0  0     95  26   0   113   1  0  0
115  26   0   168   1  0  0    154  26   0   133   1  0  0
218  26   0   160   0  0  0     16  27   1   150   0  0  0
 43  27   1   130   0  0  1    125  27   0   124   1  0  0
  4  28   1   120   1  0  1     79  28   1    95   1  0  0
105  28   0   120   1  0  0    109  28   0   120   0  0  0
112  28   0   167   0  0  0    151  28   0   140   0  0  0
159  28   0   250   1  0  0    212  28   0   134   0  0  0
214  28   0   130   0  0  0     10  29   1   130   0  0  1
 94  29   0   123   1  0  0    114  29   0   150   0  0  0
123  29   0   140   1  0  0    190  29   0   135   0  0  0
191  29   0   154   0  0  0    209  29   0   130   1  0  0
 65  30   1   142   1  0  0     99  30   0   107   0  0  1
141  30   0    95   1  0  0    145  30   0   153   0  0  0
176  30   0   110   0  0  0    195  30   0   137   0  0  0
203  30   0   112   0  0  0     56  31   1   102   1  0  0
107  31   0   100   0  0  1    126  31   0   215   1  0  0
163  31   0   150   1  0  0    222  31   0   120   0  0  0
 22  32   1   105   1  0  0    106  32   0   121   0  0  0
134  32   0   132   0  0  0    170  32   0   134   1  0  0
175  32   0   170   0  0  0    207  32   0   186   0  0  0
;
```

The variable Low is used to determine whether the subject is a case (Low=1, low-birth-weight baby) or a control (Low=0, normal-weight baby). The dummy time variable Time takes the value 1 for cases and 2 for controls.

The following statements produce a conditional logistic regression analysis of the data. The variable Time is the response, and Low is the censoring variable. Note that the data set is created so that all the cases have the same event time and the controls have later censored times. The matching variable Age is used in the STRATA statement so that each unique age value defines a stratum. The variables LWT, Smoke, HT, and UI are specified as explanatory variables. The TIES=DISCRETE option requests the discrete logistic model.

```
proc phreg data=LBW;
   model Time*Low(0)= LWT Smoke HT UI / ties=discrete;
   strata Age;
run;
```

The procedure displays a summary of the number of event and censored observations for each stratum. These are the number of cases and controls for each matched set shown in Output 64.5.1.

Output 64.5.1 Summary of Number of Case and Controls

The PHREG Procedure

Model Information

Data Set	WORK.LBW
Dependent Variable	Time
Censoring Variable	Low
Censoring Value(s)	0
Ties Handling	DISCRETE

Summary of the Number of Event and Censored Values

Stratum	Age	Total	Event	Censored	Percent Censored
1	16	7	1	6	85.71
2	17	12	5	7	58.33
3	18	10	2	8	80.00
4	19	16	3	13	81.25
5	20	18	8	10	55.56
6	21	12	5	7	58.33
7	22	13	2	11	84.62
8	23	13	5	8	61.54
9	24	13	5	8	61.54
10	25	15	6	9	60.00
11	26	8	4	4	50.00
12	27	3	2	1	33.33
13	28	9	2	7	77.78
14	29	7	1	6	85.71
15	30	7	1	6	85.71
16	31	5	1	4	80.00
17	32	6	1	5	83.33
Total		174	54	120	68.97

Results of the conditional logistic regression analysis are shown in Output 64.5.2. Based on the Wald test for individual variables, the variables LWT, Smoke, and HT are statistically significant while UI is marginal.

The hazard ratios, computed by exponentiating the parameter estimates, are useful in interpreting the results of the analysis. If the hazard ratio of a prognostic factor is larger than 1, an increment in the factor increases the hazard rate. If the hazard ratio is less than 1, an increment in the factor decreases the hazard rate. Results indicate that women were more likely to have low-birth-weight babies if they were underweight in the last menstrual cycle, were hypertensive, smoked during pregnancy, or suffered uterine irritability.

Output 64.5.2 Conditional Logistic Regression Analysis for the Low-Birth-Weight Study

Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Output 64.5.2 *continued*

Model Fit Statistics

Criterion	Without Covariates	With Covariates
-2 LOG L	159.069	141.108
AIC	159.069	149.108
SBC	159.069	157.064

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	17.9613	4	0.0013
Score	17.3152	4	0.0017
Wald	15.5577	4	0.0037

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
LWT	1	-0.01498	0.00706	4.5001	0.0339	0.985
Smoke	1	0.80805	0.36797	4.8221	0.0281	2.244
HT	1	1.75143	0.73932	5.6120	0.0178	5.763
UI	1	0.88341	0.48032	3.3827	0.0659	2.419

For matched case-control studies with one case per matched set (1:n matching), the likelihood function for the conditional logistic regression reduces to that of the Cox model for the continuous time scale. For this situation, you can use the default TIES=BRESLOW.

Example 64.6: Model Using Time-Dependent Explanatory Variables

Time-dependent variables can be used to model the effects of subjects transferring from one treatment group to another. One example of the need for such strategies is the Stanford heart transplant program. Patients are accepted if physicians judge them suitable for heart transplant. Then, when a donor becomes available, physicians choose transplant recipients according to various medical criteria. A patient's status can be changed during the study from waiting for a transplant to being a transplant recipient. Transplant status can be defined by the time-dependent covariate function $z = z(t)$ as

$$z(t) = \begin{cases} 0 & \text{if the patient has not received the transplant at time } t \\ 1 & \text{if the patient has received the transplant at time } t \end{cases}$$

The Stanford heart transplant data that appear in Crowley and Hu (1977) consist of 103 patients, 69 of whom received transplants. The data are saved in a SAS data set called Heart in the following DATA step. For each patient in the program, there is a birth date (Bir_Date), a date of acceptance

(Acc_Date), and a date last seen (Ter_Date). The survival time (Time) in days is defined as Time = Ter_Date − Acc_Date. The survival time is said to be uncensored (Status=1) or censored (Status=0), depending on whether Ter_Date is the date of death or the closing date of the study. The age, in years, at acceptance into the program is Acc_Age = (Acc_Date − Bir_Date) / 365. Previous open-heart surgery for each patient is indicated by the variable PrevSurg. For each transplant recipient, there is a date of transplant (Xpl_Date) and three measures (NMismatch, Antigen, Mismatch) of tissue-type mismatching. The waiting period (WaitTime) in days for a transplant recipient is calculated as WaitTime = Xpl_Date − Acc_Date, and the age (in years) at transplant is Xpl_Age = (Xpl_Date − Bir_Date) / 365. For those who do not receive heart transplants, the WaitTime, Xpl_Age, NMismatch, Antigen, and Mismatch variables contain missing values.

The input data contain dates that have a two-digit year representation. The SAS option YEARCUTOFF=1900 is specified to ensure that a two-digit year xx is year 19xx.

```
options yearcutoff=1900;
data Heart;
   input ID
         @5  Bir_Date mmddyy8.
         @14 Acc_Date mmddyy8.
         @23 Xpl_Date mmddyy8.
         @32 Ter_Date mmddyy8.
         @41 Status 1.
         @43 PrevSurg 1.
         @45 NMismatch 1.
         @47 Antigen 1.
         @49 Mismatch 4.
         @54 Reject 1.
         @56 NotTyped $1.;
   label Bir_Date ='Date of birth'
         Acc_Date ='Date of acceptance'
         Xpl_Date ='Date of transplant'
         Ter_Date ='Date last seen'
         Status   =  'Dead=1 Alive=0'
         PrevSurg ='Previous surgery'
         NMismatch= 'No of mismatches'
         Antigen  = 'HLA-A2 antigen'
         Mismatch ='Mismatch score'
         NotTyped = 'y=not tissue-typed';
   Time= Ter_Date - Acc_Date;
   Acc_Age=int( (Acc_Date - Bir_Date)/365 );
   if ( Xpl_Date ne .) then do;
      WaitTime= Xpl_Date - Acc_Date;
      Xpl_Age= int( (Xpl_Date - Bir_Date)/365 );
   end;
   datalines;
 1 01 10 37 11 15 67          01 03 68 1 0
 2 03 02 16 01 02 68          01 07 68 1 0
 3 09 19 13 01 06 68 01 06 68 01 21 68 1 0 2 0 1.11 0
 4 12 23 27 03 28 68 05 02 68 05 05 68 1 0 3 0 1.66 0
 5 07 28 47 05 10 68          05 27 68 1 0
 6 11 08 13 06 13 68          06 15 68 1 0
 7 08 29 17 07 12 68 08 31 68 05 17 70 1 0 4 0 1.32 1
```

```
 8 03 27 23 08 01 68          09 09 68 1 0
 9 06 11 21 08 09 68          11 01 68 1 0
10 02 09 26 08 11 68 08 22 68 10 07 68 1 0 2 0 0.61 1
11 08 22 20 08 15 68 09 09 68 01 14 69 1 0 1 0 0.36 0
12 07 09 15 09 17 68          09 24 68 1 0
13 02 22 14 09 19 68 10 05 68 12 08 68 1 0 3 0 1.89 1
14 09 16 14 09 20 68 10 26 68 07 07 72 1 0 1 0 0.87 1
15 12 04 14 09 27 68          09 27 68 1 1
16 05 16 19 10 26 68 11 22 68 08 29 69 1 0 2 0 1.12 1
17 06 29 48 10 28 68          12 02 68 1 0
18 12 27 11 11 01 68 11 20 68 12 13 68 1 0 3 0 2.05 0
19 10 04 09 11 18 68          12 24 68 1 0
20 10 19 13 01 29 69 02 15 69 02 25 69 1 0 3 1 2.76 1
21 09 29 25 02 01 69 02 08 69 11 29 71 1 0 2 0 1.13 1
22 06 05 26 03 18 69 03 29 69 05 07 69 1 0 3 0 1.38 1
23 12 02 10 04 11 69 04 13 69 04 13 71 1 0 3 0 0.96 1
24 07 07 17 04 25 69 07 16 69 11 29 69 1 0 3 1 1.62 1
25 02 06 36 04 28 69 05 22 69 04 01 74 0 0 2 0 1.06 0
26 10 18 38 05 01 69          03 01 73 0 0
27 07 21 60 05 04 69          01 21 70 1 0
28 05 30 15 06 07 69 08 16 69 08 17 69 1 0 2 0 0.47 0
29 02 06 19 07 14 69          08 17 69 1 0
30 09 20 24 08 19 69 09 03 69 12 18 71 1 0 4 0 1.58 1
31 10 04 14 08 23 69          09 07 69 1 0
32 04 02 05 08 29 69 09 14 69 11 13 69 1 0 4 0 0.69 1
33 01 01 21 11 27 69 01 16 70 04 01 74 0 0 3 0 0.91 0
34 05 24 29 12 12 69 01 03 70 04 01 74 0 0 2 0 0.38 0
35 08 04 26 01 21 70          02 01 70 1 0
36 05 01 21 04 04 70 05 19 70 07 12 70 1 0 2 0 2.09 1
37 10 24 08 04 25 70 05 13 70 06 29 70 1 0 3 1 0.87 1
38 11 14 28 05 05 70 05 09 70 05 09 70 1 0 3 0 0.87 0
39 11 12 19 05 20 70 05 21 70 07 11 70 1 0             y
40 11 30 21 05 25 70 07 04 70 04 01 74 0 1 4 0 0.75 0
41 04 30 25 08 19 70 10 15 70 04 01 74 0 1 2 0 0.98 0
42 03 13 34 08 21 70          08 23 70 1 0
43 06 01 27 10 22 70          10 23 70 1 1
44 05 02 28 11 30 70          01 08 71 1 1
45 10 30 34 01 05 71 01 05 71 02 18 71 1 0 1 0 0.0  0
46 06 01 22 01 10 71 01 11 71 10 01 73 1 1 2 0 0.81 1
47 12 28 23 02 02 71 02 22 71 04 14 71 1 0 3 0 1.38 1
48 01 23 15 02 05 71          02 13 71 1 0
49 06 21 34 02 15 71 03 22 71 04 01 74 0 1 4 0 1.35 0
50 03 28 25 02 15 71 05 08 71 10 21 73 1 1             y
51 06 29 22 03 24 71 04 24 71 01 02 72 1 0 4 1 1.08 1
52 01 24 30 04 25 71          08 04 71 1 0
53 02 27 24 07 02 71 08 11 71 01 05 72 1 0             y
54 09 16 23 07 02 71          07 04 71 1 0
55 02 24 19 08 09 71 08 18 71 10 08 71 1 0 2 0 1.51 1
56 12 05 32 09 03 71 11 08 71 04 01 74 0 0 4 0 0.98 0
57 06 08 30 09 13 71          02 08 72 1 0
58 09 17 23 09 23 71 10 13 71 08 30 72 1 1 2 1 1.82 1
59 05 12 30 09 29 71 12 15 71 04 01 74 0 1 2 0 0.19 0
60 10 29 22 11 18 71 11 20 71 01 24 72 1 0 3 0 0.66 1
61 05 12 19 12 04 71          12 05 71 1 0
```

```
 62 08 01 32 12 09 71          02 15 72 1 0
 63 04 15 39 12 12 71 01 07 72 04 01 74 0 0 3 1 1.93 0
 64 04 09 23 02 01 72 03 04 72 09 06 73 1 1 1 0 0.12 0
 65 11 19 20 03 06 72 03 17 72 05 22 72 1 0 2 0 1.12 1
 66 01 02 19 03 20 72          04 20 72 1 0
 67 09 03 52 03 23 72 05 18 72 01 01 73 1 0 3 0 1.02 0
 68 01 10 27 04 07 72 04 09 72 06 13 72 1 0 3 1 1.68 1
 69 06 05 24 06 01 72 06 10 72 04 01 74 0 0 2 0 1.20 0
 70 06 17 19 06 17 72 06 21 72 07 16 72 1 0 3 1 1.68 1
 71 02 22 25 07 21 72 08 20 72 04 01 74 0 0 3 0 0.97 0
 72 11 22 45 08 14 72 08 17 72 04 01 74 0 0 3 1 1.46 0
 73 05 13 16 09 11 72 10 07 72 12 09 72 1 0 3 1 2.16 1
 74 07 20 43 09 18 72 09 22 72 10 04 72 1 0 1 0 0.61 0
 75 07 25 20 09 29 72          09 30 72 1 0
 76 09 03 20 10 04 72 11 18 72 04 01 74 0 1 3 1 1.70 0
 77 08 27 31 10 06 72          10 26 72 1 0
 78 02 20 24 11 03 72 05 31 73 04 01 74 0 0 3 0 0.81 0
 79 02 18 19 11 30 72 02 04 73 03 05 73 1 0 2 0 1.08 1
 80 06 27 26 12 06 72 12 31 72 04 01 74 0 1 3 0 1.41 0
 81 02 21 20 01 12 73 01 17 73 04 01 74 0 0 4 1 1.94 0
 82 08 19 42 11 01 71          01 01 73 0 0
 83 10 04 19 01 24 73 02 24 73 04 13 73 1 0 4 0 3.05 0
 84 05 13 30 01 30 73 03 07 73 12 29 73 1 0 4 0 0.60 1
 85 02 13 25 02 06 73          02 10 73 1 0
 86 03 30 24 03 01 73 03 08 73 04 01 74 0 0 3 1 1.44 0
 87 12 19 26 03 21 73 05 19 73 07 08 73 1 0 2 0 2.25 1
 88 11 16 18 03 28 73 04 27 73 04 01 74 0 0 3 0 0.68 0
 89 03 19 22 04 05 73 08 21 73 10 28 73 1 0 4 1 1.33 1
 90 03 25 21 04 06 73 09 12 73 10 08 73 1 1 3 1 0.82 0
 91 09 08 25 04 13 73          03 18 74 1 0
 92 05 03 28 04 27 73 03 02 74 04 01 74 0 0 1 0 0.16 0
 93 10 10 25 07 11 73 08 07 73 04 01 74 0 0 2 0 0.33 0
 94 11 11 29 09 14 73 09 17 73 02 25 74 1 1 3 0 1.20 1
 95 06 11 33 09 22 73 09 23 73 10 07 73 1 0            y
 96 02 09 47 10 04 73 10 16 73 04 01 74 0 0 2 0 0.46 0
 97 04 11 50 11 22 73 12 12 73 04 01 74 0 0 3 1 1.78 0
 98 04 28 45 12 14 73 03 19 74 04 01 74 0 0 4 1 0.77 0
 99 02 24 24 12 25 73          01 14 74 1 0
100 01 31 39 02 22 74 03 31 74 04 01 74 0 1 3 0 0.67 0
101 08 25 24 03 02 74          04 01 74 0 0
102 10 30 33 03 22 74          04 01 74 0 0
103 05 20 28 09 13 67          09 18 67 1 0
   ;
```

Crowley and Hu (1977) have presented a number of analyses to assess the effects of various explanatory variables on the survival of patients. This example fits two of the models that they have considered.

The first model consists of two explanatory variables—the transplant status and the age at acceptance. The transplant status (XStatus) is a time-dependent variable defined by the programming statements between the MODEL statement and the RUN statement. The XStatus variable takes the value 1 or 0 at time t (measured from the date of acceptance), depending on whether or not the patient has received a transplant at that time. Note that the value of XStatus changes for subjects in

each risk set (subjects still alive just before each distinct event time); therefore, the variable cannot be created in the DATA step. The variable Acc_Age, which is not time dependent, accounts for the possibility that pretransplant risks vary with age. The following statements fit this model:

```
proc phreg data= Heart;
   model Time*Status(0)= XStatus Acc_Age;
   if (WaitTime = . or Time < WaitTime) then XStatus=0.;
   else  XStatus= 1.0;
run;
```

Results of this analysis are shown in Output 64.6.1. Transplantation appears to be associated with a slight decrease in risk, although the effect is not significant ($p = 0.8261$). The age at acceptance as a pretransplant risk factor adds significantly to the model ($p = 0.0289$). The risk increases significantly with age at acceptance.

Output 64.6.1 Heart Transplant Study Analysis I

The PHREG Procedure

Model Information

Data Set	WORK.HEART	
Dependent Variable	Time	
Censoring Variable	Status	Dead=1 Alive=0
Censoring Value(s)	0	
Ties Handling	BRESLOW	

Number of Observations Read	103
Number of Observations Used	103

Summary of the Number of Event and Censored Values

Total	Event	Censored	Percent Censored
103	75	28	27.18

Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Without Covariates	With Covariates
-2 LOG L	596.651	591.292
AIC	596.651	595.292
SBC	596.651	599.927

Output 64.6.1 *continued*

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	5.3593	2	0.0686
Score	4.8093	2	0.0903
Wald	4.7999	2	0.0907

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
XStatus	1	-0.06720	0.30594	0.0482	0.8261	0.935
Acc_Age	1	0.03158	0.01446	4.7711	0.0289	1.032

The second model consists of three explanatory variables—the transplant status, the transplant age, and the mismatch score. Four transplant recipients who were not typed have no Mismatch values; they are excluded from the analysis by the use of a WHERE clause. The transplant age (XAge) and the mismatch score (XScore) are also time dependent and are defined in a fashion similar to that of XStatus. While the patient is waiting for a transplant, XAge and XScore have a value of 0. After the patient has migrated to the recipient population, XAge takes on the value of Xpl_Age (transplant age for the recipient), and XScore takes on the value of Mismatch (a measure of the degree of dissimilarity between donor and recipient). The following statements fit this model:

```
proc phreg data= Heart;
   model Time*Status(0)= XStatus XAge XScore;
   where NotTyped ^= 'y';
   if (WaitTime = . or Time < WaitTime) then do;
      XStatus=0.;
      XAge=0.;
      XScore= 0.;
   end;
   else do;
      XStatus= 1.0;
      XAge= Xpl_Age;
      XScore= Mismatch;
   end;
run;
```

Results of the analysis are shown in Output 64.6.2. Note that only 99 patients are included in this analysis, instead of 103 patients as in the previous analysis, since four transplant recipients who were not typed are excluded. The variable XAge is statistically significant ($p = 0.0143$), with a hazard ratio exceeding 1. Therefore, patients who had a transplant at younger ages lived longer than those who received a transplant later in their lives. The variable XScore has only minimal effect on the survival ($p = 0.1121$).

Output 64.6.2 Heart Transplant Study Analysis II

The PHREG Procedure

Model Information

Data Set	WORK.HEART	
Dependent Variable	Time	
Censoring Variable	Status	Dead=1 Alive=0
Censoring Value(s)	0	
Ties Handling	BRESLOW	

Number of Observations Read	99
Number of Observations Used	99

Summary of the Number of Event and Censored Values

Total	Event	Censored	Percent Censored
99	71	28	28.28

Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Without Covariates	With Covariates
-2 LOG L	561.680	551.874
AIC	561.680	557.874
SBC	561.680	564.662

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	9.8059	3	0.0203
Score	9.0521	3	0.0286
Wald	9.0554	3	0.0286

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
XStatus	1	-3.19837	1.18746	7.2547	0.0071	0.041
XAge	1	0.05544	0.02263	6.0019	0.0143	1.057
XScore	1	0.44490	0.28001	2.5245	0.1121	1.560

Example 64.7: Time-Dependent Repeated Measurements of a Covariate

Repeated determinations can be made during the course of a study of variables thought to be related to survival. Consider an experiment to study the dosing effect of a tumor-promoting agent. Forty-five rodents initially exposed to a carcinogen were randomly assigned to three dose groups. After the first death of an animal, the rodents were examined every week for the number of papillomas. Investigators were interested in determining the effects of dose on the carcinoma incidence after adjusting for the number of papillomas.

The input data set TUMOR consists of the following 19 variables:

- ID (subject identification)
- Time (survival time of the subject)
- Dead (censoring status where 1=dead and 0=censored)
- Dose (dose of the tumor-promoting agent)
- P1–P15 (number of papillomas at the 15 times that animals died. These 15 death times are weeks 27, 34, 37, 41, 43, 45, 46, 47, 49, 50, 51, 53, 65, 67, and 71. For instance, subject 1 died at week 47; it had no papilloma at week 27, five papillomas at week 34, six at week 37, eight at week 41, and 10 at weeks 43, 45, 46, and 47. For an animal that died before week 71, the number of papillomas is missing for those times beyond its death.)

The following SAS statements create the data set TUMOR:

```
data Tumor;
   infile datalines missover;
   input ID Time Dead Dose P1-P15;
   label ID='Subject ID';
   datalines;
 1 47 1  1.0  0  5  6  8 10 10 10 10
 2 71 1  1.0  0  0  0  0  0  0  0  0  1  1  1  1 1 1 1
 3 81 0  1.0  0  1  1  1  1  1  1  1  1  1  1  1 1 1 1
 4 81 0  1.0  0  0  0  0  0  0  0  0  0  0  0  0 0 0 0
 5 81 0  1.0  0  0  0  0  0  0  0  0  0  0  0  0 0 0 0
 6 65 1  1.0  0  0  0  1  1  1  1  1  1  1  1  1 1
 7 71 0  1.0  0  0  0  0  0  0  0  0  0  0  0  0 0 0 0
 8 69 0  1.0  0  0  0  0  0  0  0  0  0  0  0  0 0 0
 9 67 1  1.0  0  0  1  1  2  2  2  2  3  3  3  3 3 3
10 81 0  1.0  0  0  0  0  0  0  0  0  0  0  0  0 0 0 0
11 37 1  1.0  9  9  9
12 81 0  1.0  0  0  0  0  0  0  0  0  0  0  0  0 0 0 0
13 77 0  1.0  0  0  0  0  1  1  1  1  1  1  1  1 1 1 1
14 81 0  1.0  0  0  0  0  0  0  0  0  0  0  0  0 0 0 0
15 81 0  1.0  0  0  0  0  0  0  0  0  0  0  0  0 0 0 0
16 54 0  2.5  0  1  1  1  2  2  2  2  2  2  2  2
17 53 0  2.5  0  0  0  0  0  0  0  0  0  0  0  0
18 38 0  2.5  5 13 14
```

```
19 54 0  2.5  2  6  6  6  6  6  6  6  6  6  6
20 51 1  2.5 15 15 15 16 16 17 17 17 17 17 17
21 47 1  2.5 13 20 20 20 20 20 20 20
22 27 1  2.5 22
23 41 1  2.5  6 13 13 13
24 49 1  2.5  0  3  3  3  3  3  3  3  3
25 53 0  2.5  0  0  1  1  1  1  1  1  1  1  1
26 50 1  2.5  0  0  2  3  4  6  6  6  6  6
27 37 1  2.5  3 15 15
28 49 1  2.5  2  3  3  3  3  4  4  4  4
29 46 1  2.5  4  6  7  9  9  9  9
30 48 0  2.5 15 26 26 26 26 26 26 26
31 54 0 10.0 12 14 15 15 15 15 15 15 15 15 15 15
32 37 1 10.0 12 16 17
33 53 1 10.0  3  6  6  6  6  6  6  6  6  6  6  6
34 45 1 10.0  4 12 15 20 20 20
35 53 0 10.0  6 10 13 13 13 15 15 15 15 15 15 20
36 49 1 10.0  0  2  2  2  2  2  2  2  2
37 39 0 10.0  7  8  8
38 27 1 10.0 17
39 49 1 10.0  0  6  9 14 14 14 14 14 14
40 43 1 10.0 14 18 20 20 20
41 28 0 10.0  8
42 34 1 10.0 11 18
43 45 1 10.0 10 12 16 16 16 16
44 37 1 10.0  0  1  1
45 43 1 10.0  9 19 19 19 19
;
```

The number of papillomas (NPap) for each animal in the study was measured repeatedly over time. One way of handling time-dependent repeated measurements in the PHREG procedure is to use programming statements to capture the appropriate covariate values of the subjects in each risk set. In this example, NPap is a time-dependent explanatory variable with values that are calculated by means of the programming statements shown in the following SAS statements:

```
proc phreg data=Tumor;
   model Time*Dead(0)=Dose NPap;
   array pp{*} P1-P14;
   array tt{*} t1-t15;
    t1=27;  t2=34;  t3=37;  t4=41;  t5=43;
    t6=45;  t7=46;  t8=47;  t9=49; t10=50;
   t11=51; t12=53; t13=65; t14=67; t15=71;
   if Time <  tt[1]  then NPap=0;
   else if time >= tt[15] then NPap=P15;
   else do i=1 to dim(pp);
      if tt[i] <= Time < tt[i+1] then NPap= pp[i];
   end;
run;
```

At each death time, the NPap value of each subject in the risk set is recalculated to reflect the actual number of papillomas at the given death time. For instance, subject one in the data set Tumor was in the risk sets at weeks 27 and 34; at week 27, the animal had no papilloma, while at week 34, it had five papillomas. Results of the analysis are shown in Output 64.7.1. After the number of papillomas is adjusted for, the dose effect of the tumor-promoting agent is not statistically significant.

Output 64.7.1 Cox Regression Analysis on the Survival of Rodents

The PHREG Procedure

Model Information

Data Set	WORK.TUMOR
Dependent Variable	Time
Censoring Variable	Dead
Censoring Value(s)	0
Ties Handling	BRESLOW

Number of Observations Read	45
Number of Observations Used	45

Summary of the Number of Event and Censored Values

Total	Event	Censored	Percent Censored
45	25	20	44.44

Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Without Covariates	With Covariates
-2 LOG L	166.793	143.269
AIC	166.793	147.269
SBC	166.793	149.707

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	23.5243	2	<.0001
Score	28.0498	2	<.0001
Wald	21.1646	2	<.0001

Output 64.7.1 *continued*

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
Dose	1	0.06885	0.05620	1.5010	0.2205	1.071
NPap	1	0.11714	0.02998	15.2705	<.0001	1.124

Another way to handle time-dependent repeated measurements in the PHREG procedure is to use the counting process style of input. Multiple records are created for each subject, one record for each distinct pattern of the time-dependent measurements. Each record contains a T1 value and a T2 value representing the time interval (T1,T2] during which the values of the explanatory variables remain unchanged. Each record also contains the censoring status at T2.

One advantage of using the counting process formulation is that you can easily obtain various residuals and influence statistics that are not available when programming statements are used to compute the values of the time-dependent variables. On the other hand, creating multiple records for the counting process formulation requires extra effort in data manipulation.

Consider a counting process style of input data set named Tumor1. It contains multiple observations for each subject in the data set Tumor. In addition to variables ID, Time, Dead, and Dose, four new variables are generated:

- T1 (left endpoint of the risk interval)
- T2 (right endpoint of the risk interval)
- NPap (number of papillomas in the time interval (T1,T2])
- Status (censoring status at T2)

For example, five observations are generated for the rodent that died at week 47 and that had no papilloma at week 27, five papillomas at week 34, six at week 37, eight at week 41, and 10 at weeks 43, 45, 46, and 47. The values of T1, T2, NPap, and Status for these five observations are (0,27,0,0), (27,34,5,0), (34,37,6,0), (37,41,8,0), and (41,47,10,1). Note that the variables ID, Time, and Dead are not needed for the estimation of the regression parameters, but they are useful for plotting the residuals.

The following SAS statements create the data set Tumor1:

```
data Tumor1(keep=ID Time Dead Dose T1 T2 NPap Status);
   array pp{*} P1-P14;
   array qq{*} P2-P15;
   array tt{1:15} _temporary_
      (27 34 37 41 43 45 46 47 49 50 51 53 65 67 71);
   set Tumor;
   T1 = 0;
   T2 = 0;
   Status = 0;
   if ( Time = tt[1] ) then do;
      T2 = tt[1];
      NPap = p1;
      Status = Dead;
      output;
   end;
   else do _i_=1 to dim(pp);
      if ( tt[_i_] = Time ) then do;
         T2= Time;
         NPap = pp[_i_] ;
         Status = Dead;
         output;
      end;
      else if (tt[_i_]  < Time ) then do;
         if (pp[_i_]  ^= qq[_i_] ) then do;
            if qq[_i_]  = . then T2= Time;
            else                 T2= tt[_i_] ;
            NPap= pp[_i_] ;
            Status= 0;
            output;
            T1 = T2;
         end;
      end;
   end;
   if ( Time >= tt[15] ) then do;
      T2 = Time;
      NPap = P15;
      Status = Dead;
      output;
   end;
   run;
```

In the following SAS statements, the counting process MODEL specification is used. The DFBETA statistics are output to a SAS data set named Out1. Note that Out1 contains multiple observations for each subject—that is, one observation for each risk interval (T1,T2].

```
proc phreg data=Tumor1;
   model (T1,T2)*Status(0)=Dose NPap;
   output out=Out1 resmart=Mart dfbeta=db1-db2;
   id ID Time Dead;
run;
```

The output from PROC PHREG (not shown) is identical to Output 64.7.1 except for the "Summary of the Number of Event and Censored Values" table. The number of event observations remains unchanged between the two specifications of PROC PHREG, but the number of censored observations differs due to the splitting of each subject's data into multiple observations for the counting process style of input.

Next, the MEANS procedure sums up the component statistics for each subject and outputs the results to a SAS data set named Out2:

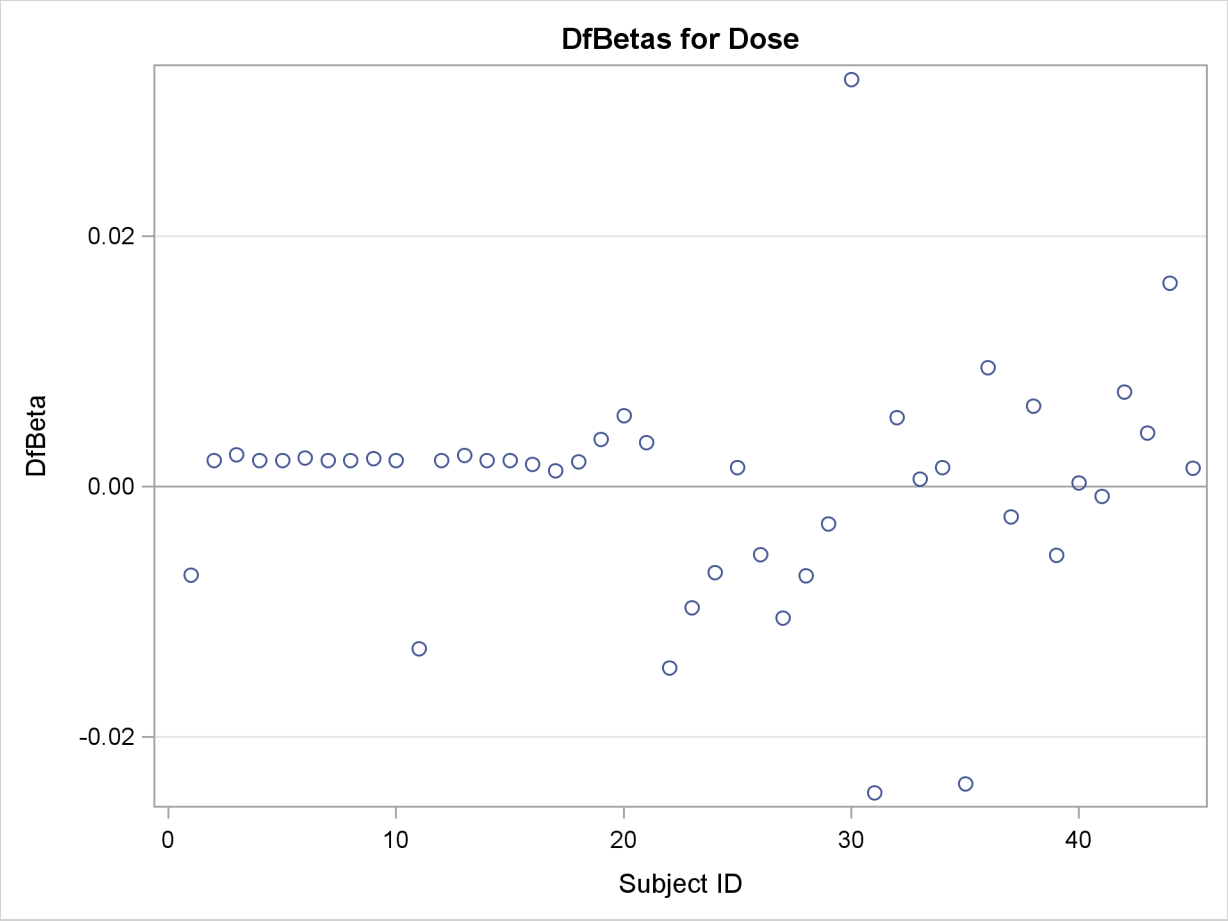
```
proc means data=Out1 noprint;
   by ID Time Dead;
   var Mart db1-db2;
   output out=Out2 sum=Mart db_Dose db_NPap;
run;
```

Finally, DFBETA statistics are plotted against subject ID for easy identification of influential points:

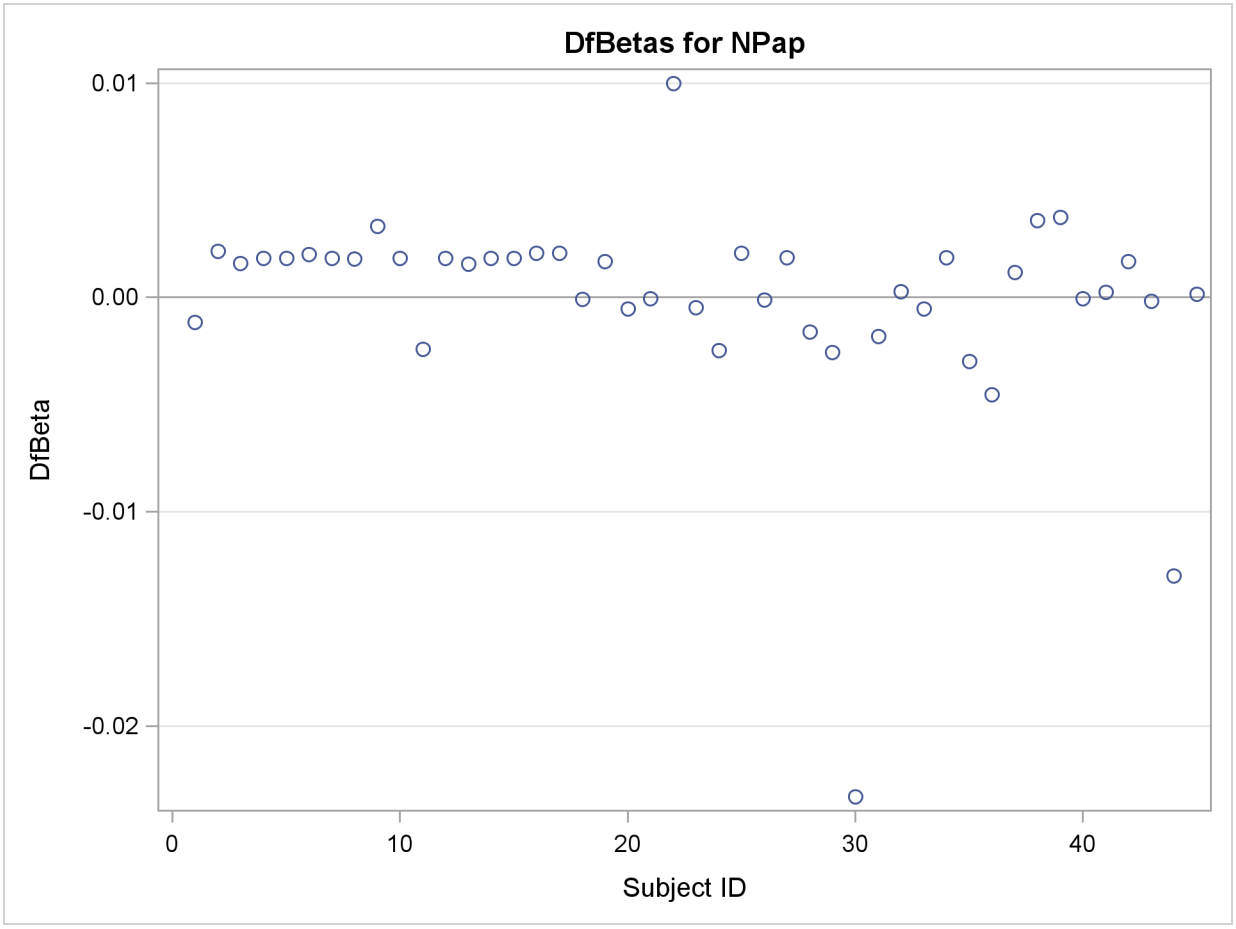
```
title 'DfBetas for Dose';
proc sgplot data=Out2;
   yaxis label="DfBeta" grid;
   refline 0 / axis=y;
   scatter y=db_Dose x=ID;
   run;
title 'DfBetas for NPap';
proc sgplot data=Out2;
   yaxis label="DfBeta" grid;
   refline 0 / axis=y;
   scatter y=db_NPap x=ID;
   run;
```

The plots of the DFBETA statistics are shown in Output 64.7.2 and Output 64.7.3. Subject 30 appears to have a large influence on both the Dose and NPap coefficients. Subjects 31 and 35 have considerable influences on the DOSE coefficient, while subjects 22 and 44 have rather large influences on the NPap coefficient.

Output 64.7.2 Plot of DFBETA Statistic for DOSE versus Subject Number



Output 64.7.3 Plot of DFBETA Statistic for NPAP versus Subject Number



Example 64.8: Survivor Function Estimates for Specific Covariate Values

You might want to use your regression analysis results to generate predicted survival curves for subjects not in the study. The COVARIATES= data set in the BASELINE statement enables you to specify the sets of covariate values for the prediction. By using the PLOTS= option in the PROC PHREG statement, you can display a survival curve for each row of covariates in the COVARIATES= data set. You can elect to output the predicted survival curves in a SAS data set by using just the BASELINE statement. This example illustrates these two tasks by using the Myeloma data in Example 64.1.

In Example 64.1, variables LogBUN and HGB were identified as the most important prognostic factors for the myeloma data. Two sets of covariates for predicting the survivor function are saved in the data set Inrisks in the following DATA step. Also created in this data set is the variable Id, whose values will be used in identifying the covariate sets in the survival plot.

```
data Inrisks;
   length Id $20;
   input LogBUN HGB Id $12-31;
   datalines;
1.00 10.0  logBUN=1.0 HGB=10
1.80 12.0  logBUN=1.8 HGB=12
;
```

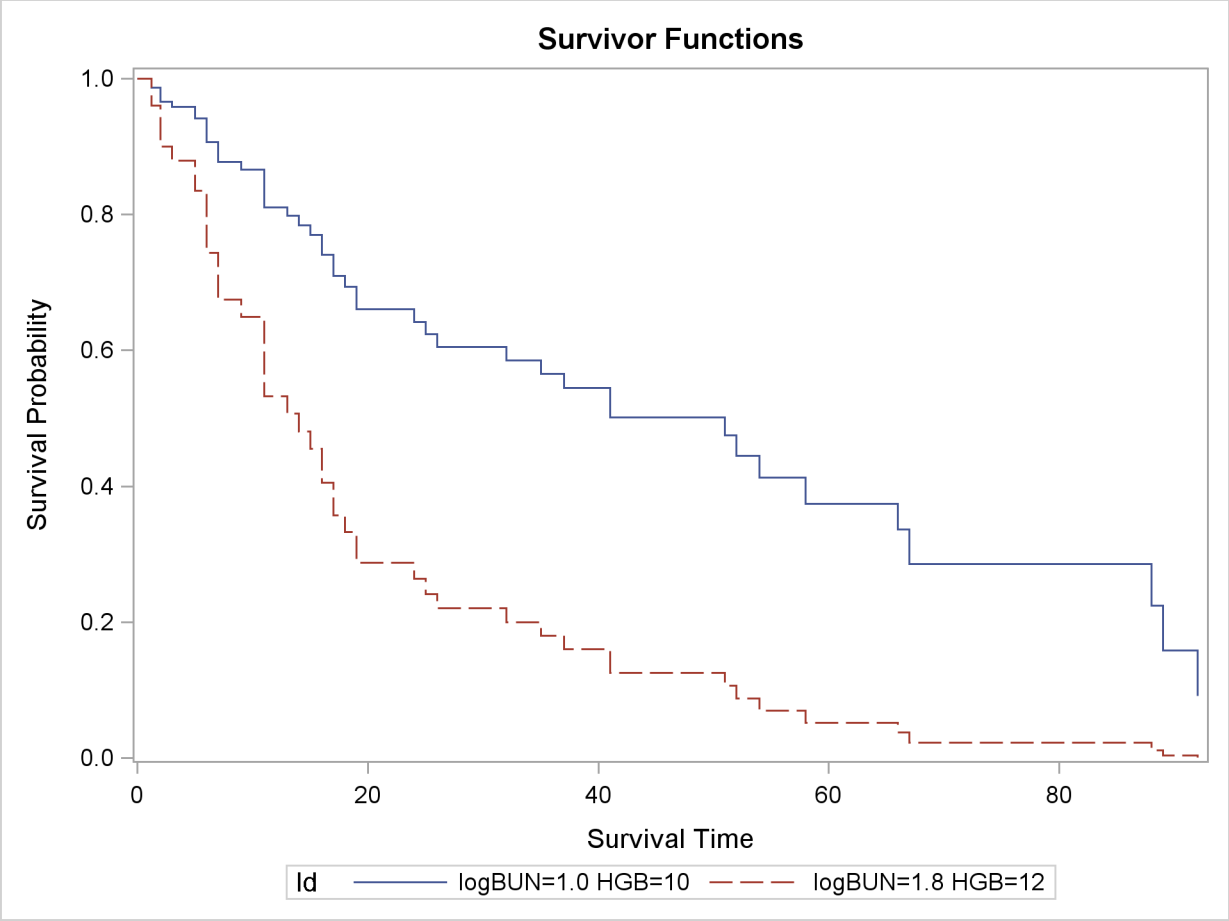
The following statements produce the plot in Output 64.8.1 and create the BASELINE data set Pred1:

```
ods graphics on;
proc phreg data=Myeloma plots(overlay)=survival;
   model Time*VStatus(0)=LogBUN HGB;
   baseline covariates=Inrisks out=Pred1 survival=_all_ / rowid=Id;
run;
ods graphics off;
```

The COVARIATES= option in the BASELINE statement specifies the data set that contains the set of covariates of interest. The PLOTS= option in the PROC PHREG statement creates the survivor plot. The OVERLAY suboption overlays the two curves in the same plot. If the OVERLAY suboption is not specified, each curve is displayed in a separate plot. The ROWID= option in the BASELINE statement specifies that the values of the variable Id in the COVARIATES= data set be used to identify the curves in the plot. The SURVIVAL=_ALL_ option in the BASELINE statement requests that the estimated survivor function, standard error, and lower and upper confidence limits for the survivor function be output into the SAS data set specified in the OUT= option.

The survival Plot (Output 64.8.1) contains two curves, one for each of row of covariates in the data set Inrisks.

Output 64.8.1 Estimated Survivor Function Plot



Finally, PROC PRINT is used to print out the observations in the data set Pred1 for the realization LogBUN=1.00 and HGB=10.0:

```
proc print data=Pred1(where=(logBUN=1 and HGB=10));
run;
```

As shown in Output 64.8.2, there are 32 observations representing the survivor function for the realization LogBUN=1.00 and HGB=10.0. The first observation has survival time 0 and survivor function estimate 1.0. Each of the remaining 31 observations represents a distinct event time in the input data set Myeloma. These observations are presented in ascending order of the event times. Note that all the variables in the COVARIATE=lnRisks data set are included in the OUT=Pred1 data set. Likewise, you can print out the observations that represent the survivor function for the realization LogBUN=1.80 and HGB=12.0.

Output 64.8.2 Survivor Function Estimates for LogBUN=1.0 and HGB=10.0

Obs	Id	Log BUN	HGB	Time	Survival	StdErr Survival	Lower Survival	Upper Survival
1	logBUN=1.0 HGB=10	1	10	0.00	1.00000	.	.	.
2	logBUN=1.0 HGB=10	1	10	1.25	0.98678	0.01043	0.96655	1.00000
3	logBUN=1.0 HGB=10	1	10	2.00	0.96559	0.01907	0.92892	1.00000
4	logBUN=1.0 HGB=10	1	10	3.00	0.95818	0.02180	0.91638	1.00000
5	logBUN=1.0 HGB=10	1	10	5.00	0.94188	0.02747	0.88955	0.99729
6	logBUN=1.0 HGB=10	1	10	6.00	0.90635	0.03796	0.83492	0.98389
7	logBUN=1.0 HGB=10	1	10	7.00	0.87742	0.04535	0.79290	0.97096
8	logBUN=1.0 HGB=10	1	10	9.00	0.86646	0.04801	0.77729	0.96585
9	logBUN=1.0 HGB=10	1	10	11.00	0.81084	0.05976	0.70178	0.93686
10	logBUN=1.0 HGB=10	1	10	13.00	0.79800	0.06238	0.68464	0.93012
11	logBUN=1.0 HGB=10	1	10	14.00	0.78384	0.06515	0.66601	0.92251
12	logBUN=1.0 HGB=10	1	10	15.00	0.76965	0.06779	0.64762	0.91467
13	logBUN=1.0 HGB=10	1	10	16.00	0.74071	0.07269	0.61110	0.89781
14	logBUN=1.0 HGB=10	1	10	17.00	0.71005	0.07760	0.57315	0.87966
15	logBUN=1.0 HGB=10	1	10	18.00	0.69392	0.07998	0.55360	0.86980
16	logBUN=1.0 HGB=10	1	10	19.00	0.66062	0.08442	0.51425	0.84865
17	logBUN=1.0 HGB=10	1	10	24.00	0.64210	0.08691	0.49248	0.83717
18	logBUN=1.0 HGB=10	1	10	25.00	0.62360	0.08921	0.47112	0.82542
19	logBUN=1.0 HGB=10	1	10	26.00	0.60523	0.09136	0.45023	0.81359
20	logBUN=1.0 HGB=10	1	10	32.00	0.58549	0.09371	0.42784	0.80122
21	logBUN=1.0 HGB=10	1	10	35.00	0.56534	0.09593	0.40539	0.78840
22	logBUN=1.0 HGB=10	1	10	37.00	0.54465	0.09816	0.38257	0.77542
23	logBUN=1.0 HGB=10	1	10	41.00	0.50178	0.10166	0.33733	0.74639
24	logBUN=1.0 HGB=10	1	10	51.00	0.47546	0.10368	0.31009	0.72901
25	logBUN=1.0 HGB=10	1	10	52.00	0.44510	0.10522	0.28006	0.70741
26	logBUN=1.0 HGB=10	1	10	54.00	0.41266	0.10689	0.24837	0.68560
27	logBUN=1.0 HGB=10	1	10	58.00	0.37465	0.10891	0.21192	0.66232
28	logBUN=1.0 HGB=10	1	10	66.00	0.33626	0.10980	0.17731	0.63772
29	logBUN=1.0 HGB=10	1	10	67.00	0.28529	0.11029	0.13372	0.60864
30	logBUN=1.0 HGB=10	1	10	88.00	0.22412	0.10928	0.08619	0.58282
31	logBUN=1.0 HGB=10	1	10	89.00	0.15864	0.10317	0.04435	0.56750
32	logBUN=1.0 HGB=10	1	10	92.00	0.09180	0.08545	0.01481	0.56907

Example 64.9: Analysis of Residuals

Residuals are used to investigate the lack of fit of a model to a given subject. You can obtain martingale and deviance residuals for the Cox proportional hazards regression analysis by requesting that they be included in the OUTPUT data set. You can plot these statistics and look for outliers.

Consider the stepwise regression analysis performed in Example 64.1. The final model included variables LogBUN and HGB. You can generate residual statistics for this analysis by refitting the model containing those variables and including an OUTPUT statement as in the following invocation of PROC PHREG. The keywords XBETA, RESMART, and RESDEV identify new variables that contain the linear predictor scores $\mathbf{z}_j'\hat{\boldsymbol{\beta}}$, martingale residuals, and deviance residuals. These variables are xb, mart, and dev, respectively.

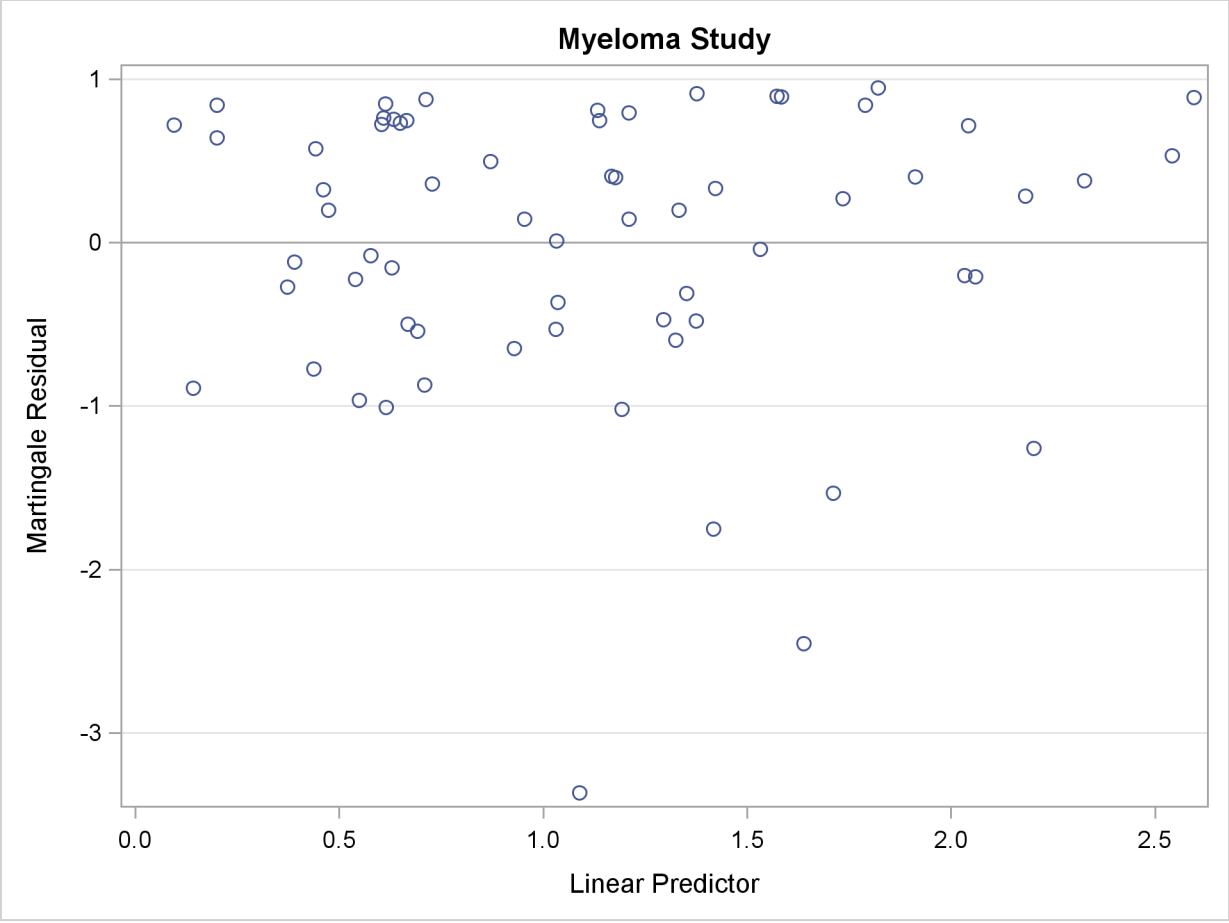
```
proc phreg data=Myeloma noprint;
   model Time*Vstatus(0)=LogBUN HGB;
   output out=Outp xbeta=Xb resmart=Mart resdev=Dev;
run;
```

The following statements plot the residuals against the linear predictor scores:

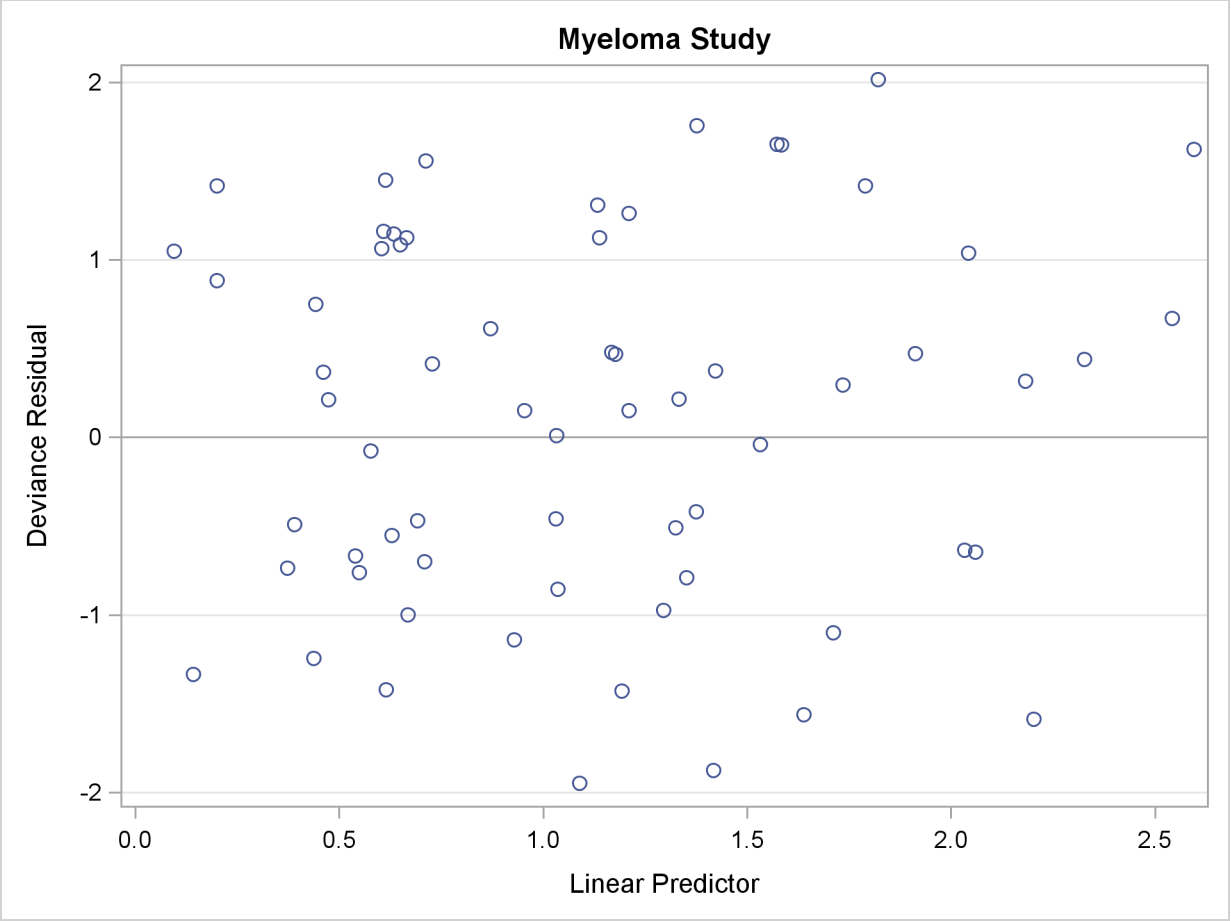
```
title "Myeloma Study";
proc sgplot data=Outp;
   yaxis grid;
   refline 0 / axis=y;
   scatter y=Mart x=Xb;
   run;
proc sgplot data=Outp;
   yaxis grid;
   refline 0 / axis=y;
   scatter y=Dev x=Xb;
   run;
```

The resulting plots are shown in Output 64.9.1 and Output 64.9.2. The martingale residuals are skewed because of the single event setting of the Cox model. The martingale residual plot shows an isolation point (with linear predictor score 1.09 and martingale residual −3.37), but this observation is no longer distinguishable in the deviance residual plot. In conclusion, there is no indication of a lack of fit of the model to individual observations.

Output 64.9.1 Martingale Residual Plot



Output 64.9.2 Deviance Residual Plot



Example 64.10: Analysis of Recurrent Events Data

Recurrent events data consist of times to a number of repeated events for each sample unit—for example, times of recurrent episodes of a disease in patients. Various ways of analyzing recurrent events data are described in the section "Analysis of Multivariate Failure Time Data" on page 4601. The bladder cancer data listed in Wei, Lin, and Weissfeld (1989) are used here to illustrate these methods.

The data consist of 86 patients with superficial bladder tumors, which were removed when the patients entered the study. Of these patients, 48 were randomized into the placebo group, and 38 were randomized into the group receiving thiotepa. Many patients had multiple recurrences of tumors during the study, and new tumors were removed at each visit. The data set contains the first four recurrences of the tumor for each patient, and each recurrence time was measured from the patient's entry time into the study.

The data consist of the following eight variables:

- Trt, treatment group (1=placebo and 2=thiotepa)
- Time, follow-up time
- Number, number of initial tumors
- Size, initial tumor size
- T1, T2, T3, and T4, times of the four potential recurrences of the bladder tumor. A patient with only two recurrences has missing values in T3 and T4.

In the data set Bladder, four observations are created for each patient, one for each of the four potential tumor recurrences. In addition to values of Trt, Number, and Size for the patient, each observation contains the following variables:

- ID, patient's identification (which is the sequence number of the subject)
- Visit, visit number (with value k for the kth potential tumor recurrence)
- TStart, time of the $(k-1)$th recurrence for Visit=k, or the entry time 0 if VISIT=1, or the follow-up time if the $(k-1)$th recurrence does not occur
- TStop, time of the kth recurrence if Visit=k or follow-up time if the kth recurrence does not occur
- Status, event status of TStop (1=recurrence and 0=censored)

For instance, a patient with only one recurrence time at month 6 who was followed until month 10 will have values for Visit, TStart, TStop, and Status of (1,0,6,1), (2,6,10,0), (3,10,10,0), and (4,10,10,0), respectively. The last two observations are redundant for the intensity model and the proportional means model, but they are important for the analysis of the marginal Cox models. If the follow-up time is beyond the time of the fourth tumor recurrence, it is tempting to create a fifth observation with the time of the fourth tumor recurrence as the TStart value, the follow-up time as the TStop value, and a Status value of 0. However, Therneau and Grambsch (2000, Section 8.5) have warned against incorporating such observations into the analysis.

The following SAS statements create the data set Bladder:

```
data Bladder;
   keep ID TStart TStop Status Trt Number Size Visit;
   retain ID TStart 0;
   array tt T1-T4;
   infile datalines missover;
   input Trt Time Number Size T1-T4;
   ID + 1;
   TStart=0;
   do over tt;
      Visit=_i_;
      if tt = . then do;
         TStop=Time;
         Status=0;
      end;
      else do;
         TStop=tt;
         Status=1;
      end;
      output;
      TStart=TStop;
   end;
   if (TStart < Time) then delete;
   datalines;
1       0       1     1
1       1       1     3
1       4       2     1
1       7       1     1
1       10      5     1
1       10      4     1   6
1       14      1     1
1       18      1     1
1       18      1     3   5
1       18      1     1   12  16
1       23      3     3
1       23      1     3   10  15
1       23      1     1   3   16  23
1       23      3     1   3   9   21
1       24      2     3   7   10  16  24
1       25      1     1   3   15  25
1       26      1     2
1       26      8     1   1
1       26      1     4   2   26
1       28      1     2   25
1       29      1     4
1       29      1     2
1       29      4     1
1       30      1     6   28  30
1       30      1     5   2   17  22
1       30      2     1   3   6   8   12
1       31      1     3   12  15  24
1       32      1     2
1       34      2     1
```

```
1       36      2     1
1       36      3     1    29
1       37      1     2
1       40      4     1    9    17   22   24
1       40      5     1    16   19   23   29
1       41      1     2
1       43      1     1    3
1       43      2     6    6
1       44      2     1    3    6    9
1       45      1     1    9    11   20   26
1       48      1     1    18
1       49      1     3
1       51      3     1    35
1       53      1     7    17
1       53      3     1    3    15   46   51
1       59      1     1
1       61      3     2    2    15   24   30
1       64      1     3    5    14   19   27
1       64      2     3    2    8    12   13
2       1       1     3
2       1       1     1
2       5       8     1    5
2       9       1     2
2       10      1     1
2       13      1     1
2       14      2     6    3
2       17      5     3    1    3    5    7
2       18      5     1
2       18      1     3    17
2       19      5     1    2
2       21      1     1    17   19
2       22      1     1
2       25      1     3
2       25      1     5
2       25      1     1
2       26      1     1    6    12   13
2       27      1     1    6
2       29      2     1    2
2       36      8     3    26   35
2       38      1     1
2       39      1     1    22   23   27   32
2       39      6     1    4    16   23   27
2       40      3     1    24   26   29   40
2       41      3     2
2       41      1     1
2       43      1     1    1    27
2       44      1     1
2       44      6     1    2    20   23   27
2       45      1     2
2       46      1     4    2
2       46      1     4
2       49      3     3
2       50      1     1
2       50      4     1    4    24   47
```

```
2        54       3      4
2        54       2      1    38
2        59       1      3
;
run;
```

First, consider fitting the intensity model (Andersen and Gill 1982) and the proportional means model (Lin et al. 2000). The counting process style of input is used in the PROC PHREG specification. For the proportional means model, inference is based on the robust sandwich covariance estimate, which is requested by the COVB(AGGREGATE) option in the PROC PHREG statement. The COVM option is specified for the analysis of the intensity model to use the model-based covariance estimate. Note that some of the observations in the data set Bladder have a degenerated interval of risk. The presence of these observations does not affect the results of the analysis since none of these observations are included in any of the risk sets. However, the procedure will run more efficiently without these observations; consequently, in the following SAS statements, the WHERE clause is used to eliminate these redundant observations:

```
title 'Intensity Model and Proportional Means Model';
proc phreg data=Bladder covm covs(aggregate);
   model (TStart, TStop) * Status(0) = Trt Number Size;
   id id;
   where TStart < TStop;
run;
```

Results of fitting the intensity model and the proportional means model are shown in Output 64.10.1 and Output 64.10.2, respectively. The robust sandwich standard error estimate for Trt is larger than its model-based counterpart, rendering the effect of thiotepa less significant in the proportional means model (p=0.0747) than in the intensity model (p=0.0215).

Output 64.10.1 Analysis of the Intensity Model

Intensity Model and Proportional Means Model

The PHREG Procedure

Analysis of Maximum Likelihood Estimates
with Model-Based Variance Estimate

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
Trt	1	-0.45979	0.19996	5.2873	0.0215	0.631
Number	1	0.17165	0.04733	13.1541	0.0003	1.187
Size	1	-0.04256	0.06903	0.3801	0.5375	0.958

Output 64.10.2 Analysis of the Proportional Means Model

Analysis of Maximum Likelihood Estimates
with Sandwich Variance Estimate

Parameter	DF	Parameter Estimate	Standard Error	StdErr Ratio	Chi-Square	Pr > ChiSq	Hazard Ratio
Trt	1	-0.45979	0.25801	1.290	3.1757	0.0747	0.631
Number	1	0.17165	0.06131	1.296	7.8373	0.0051	1.187
Size	1	-0.04256	0.07555	1.094	0.3174	0.5732	0.958

Next, consider the conditional models of Prentice, Williams, and Peterson (1981). In the PWP models, the risk set for the $(k+1)$th recurrence is restricted to those patients who have experienced the first k recurrences. For example, a patient who experienced only one recurrence is an event observation for the first recurrence; this patient is a censored observation for the second recurrence and should not be included in the risk set for the third or fourth recurrence. The following DATA step eliminates those observations that should not be in the risk sets, forming a new input data set (named Bladder2) for fitting the PWP models. The variable Gaptime, representing the gap times between successive recurrences, is also created.

```
data Bladder2(drop=LastStatus);
    retain LastStatus;
    set Bladder;
    by ID;
    if first.id then LastStatus=1;
    if (Status=0 and LastStatus=0) then delete;
    LastStatus=Status;
    Gaptime=Tstop-Tstart;
run;
```

The following statements fit the PWP total time model. The variables Trt1, Trt2, Trt3, and Trt4 are visit-specific variables for Trt; the variables Number1, Number2, Numvber3, and Number4 are visit-specific variables for Number; and the variables Size1, Size2, Size3, and Size4 are visit-specific variables for Size.

```
title 'PWP Total Time Model with Noncommon Effects';
proc phreg data=Bladder2;
   model (TStart,Tstop) * Status(0) = Trt1-Trt4 Number1-Number4
                                      Size1-Size4;
   Trt1= Trt * (Visit=1);
   Trt2= Trt * (Visit=2);
   Trt3= Trt * (Visit=3);
   Trt4= Trt * (Visit=4);
   Number1= Number * (Visit=1);
   Number2= Number * (Visit=2);
   Number3= Number * (Visit=3);
   Number4= Number * (Visit=4);
   Size1= Size * (Visit=1);
   Size2= Size * (Visit=2);
   Size3= Size * (Visit=3);
   Size4= Size * (Visit=4);
   strata Visit;
run;
```

Results of the analysis of the PWP total time model are shown in Output 64.10.3. There is no significant treatment effect on the total time in any of the four tumor recurrences.

Output 64.10.3 Analysis of the PWP Total Time Model with Noncommon Effects

PWP Total Time Model with Noncommon Effects

The PHREG Procedure

Summary of the Number of Event and Censored Values

Stratum	Visit	Total	Event	Censored	Percent Censored
1	1	85	47	38	44.71
2	2	46	29	17	36.96
3	3	27	22	5	18.52
4	4	20	14	6	30.00
Total		178	112	66	37.08

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
Trt1	1	-0.51757	0.31576	2.6868	0.1012	0.596
Trt2	1	-0.45967	0.40642	1.2792	0.2581	0.631
Trt3	1	0.11700	0.67183	0.0303	0.8617	1.124
Trt4	1	-0.04059	0.79251	0.0026	0.9592	0.960
Number1	1	0.23605	0.07607	9.6287	0.0019	1.266
Number2	1	-0.02044	0.09052	0.0510	0.8213	0.980
Number3	1	0.01219	0.18208	0.0045	0.9466	1.012
Number4	1	0.18915	0.24443	0.5989	0.4390	1.208
Size1	1	0.06790	0.10125	0.4498	0.5024	1.070
Size2	1	-0.15425	0.12300	1.5728	0.2098	0.857
Size3	1	0.14891	0.26299	0.3206	0.5713	1.161
Size4	1	0.0000732	0.34297	0.0000	0.9998	1.000

The following statements fit the PWP gap-time model:

```
title 'PWP Gap-Time Model with Noncommon Effects';
proc phreg data=Bladder2;
   model Gaptime * Status(0) = Trt1-Trt4 Number1-Number4
                               Size1-Size4;
   Trt1= Trt * (Visit=1);
   Trt2= Trt * (Visit=2);
   Trt3= Trt * (Visit=3);
   Trt4= Trt * (Visit=4);
   Number1= Number * (Visit=1);
   Number2= Number * (Visit=2);
   Number3= Number * (Visit=3);
   Number4= Number * (Visit=4);
   Size1= Size * (Visit=1);
   Size2= Size * (Visit=2);
   Size3= Size * (Visit=3);
```

```
   Size4= Size * (Visit=4);
   strata Visit;
run;
```

Results of the analysis of the PWP gap-time model are shown in Output 64.10.4. Note that the regression coefficients for the first tumor recurrence are the same as those of the total time model, since the total time and the gap time are the same for the first recurrence. There is no significant treatment effect on the gap times for any of the four tumor recurrences.

Output 64.10.4 Analysis of the PWP Gap-Time Model with Noncommon Effects

PWP Gap-Time Model with Noncommon Effects

The PHREG Procedure

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
Trt1	1	-0.51757	0.31576	2.6868	0.1012	0.596
Trt2	1	-0.25911	0.40511	0.4091	0.5224	0.772
Trt3	1	0.22105	0.54909	0.1621	0.6873	1.247
Trt4	1	-0.19498	0.64184	0.0923	0.7613	0.823
Number1	1	0.23605	0.07607	9.6287	0.0019	1.266
Number2	1	-0.00571	0.09667	0.0035	0.9529	0.994
Number3	1	0.12935	0.15970	0.6561	0.4180	1.138
Number4	1	0.42079	0.19816	4.5091	0.0337	1.523
Size1	1	0.06790	0.10125	0.4498	0.5024	1.070
Size2	1	-0.11636	0.11924	0.9524	0.3291	0.890
Size3	1	0.24995	0.23113	1.1695	0.2795	1.284
Size4	1	0.03557	0.29043	0.0150	0.9025	1.036

You can fit the PWP total time model with common effects by using the following SAS statements. However, the analysis is not shown here.

```
title2 'PWP Total Time Model with Common Effects';
proc phreg data=Bladder2;
   model (tstart,tstop) * status(0) = Trt Number Size;
   strata Visit;
run;
```

You can fit the PWP gap-time model with common effects by using the following statements. Again, the analysis is not shown here.

```
title2 'PWP Gap Time Model with Common Effects';
proc phreg data=Bladder2;
   model Gaptime * Status(0) = Trt Number Size;
   strata Visit;
run;
```

Recurrent events data are a special case of multiple events data in which the recurrence times are regarded as multivariate failure times and the marginal approach of Wei, Lin, and Weissfeld (1989) can be used. WLW fits a Cox model to each of the component times and makes statistical inference

of the regression parameters based on a robust sandwich covariance matrix estimate. No specific correlation structure is imposed on the multivariate failure times. For the kth marginal model, let $\boldsymbol{\beta}_k$ denote the row vector of regression parameters, let $\hat{\boldsymbol{\beta}}_k$ denote the maximum likelihood estimate of $\boldsymbol{\beta}_k$, let $\hat{\mathbf{A}}_k$ denote the covariance matrix obtained by inverting the observed information matrix, and let $\mathbf{R}_k$ denote the matrix of score residuals. WLW showed that the joint distribution of $(\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_4)'$ can be approximated by a multivariate normal distribution with mean vector $(\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_4)'$ and robust covariance matrix

$$\begin{pmatrix} \mathbf{V}_{11} & \mathbf{V}_{12} & \mathbf{V}_{13} & \mathbf{V}_{14} \\ \mathbf{V}_{21} & \mathbf{V}_{22} & \mathbf{V}_{23} & \mathbf{V}_{24} \\ \mathbf{V}_{31} & \mathbf{V}_{32} & \mathbf{V}_{33} & \mathbf{V}_{34} \\ \mathbf{V}_{41} & \mathbf{V}_{42} & \mathbf{V}_{43} & \mathbf{V}_{44} \end{pmatrix}$$

with the submatrix $\mathbf{V}_{ij}$ given by

$$\mathbf{V}_{ij} = \hat{\mathbf{A}}_i(\mathbf{R}_i'\mathbf{R}_j)\hat{\mathbf{A}}_j$$

In this example, there are four marginal proportional hazards models, one for each potential recurrence time. Instead of fitting one model at a time, you can fit all four marginal models in one analysis by using the STRATA statement and model-specific covariates as in the following statements. Using Visit as the STRATA variable on the input data set Bladder, PROC PHREG simultaneously fits all four marginal models, one for each Visit value. The COVS(AGGREGATE) option is specified to compute the robust sandwich variance estimate by summing up the score residuals for each distinct pattern of ID value. The TEST statement TREATMENT is used to perform the global test of no treatment effect for each tumor recurrence, the AVERAGE option is specified to estimate the parameter for the common treatment effect, and the E option displays the optimal weights for the common treatment effect.

```
title 'Wei-Lin-Weissfeld Model';
proc phreg data=Bladder covs(aggregate);
   model TStop*Status(0)=Trt1-Trt4 Number1-Number4 Size1-Size4;
   Trt1= Trt * (Visit=1);
   Trt2= Trt * (Visit=2);
   Trt3= Trt * (Visit=3);
   Trt4= Trt * (Visit=4);
   Number1= Number * (Visit=1);
   Number2= Number * (Visit=2);
   Number3= Number * (Visit=3);
   Number4= Number * (Visit=4);
   Size1= Size * (Visit=1);
   Size2= Size * (Visit=2);
   Size3= Size * (Visit=3);
   Size4= Size * (Visit=4);
   strata Visit;
   id ID;
   TREATMENT: test trt1,trt2,trt3,trt4/average e;
   run;
```

Out of the 86 patients, 47 patients have only one tumor recurrence, 29 patients have two recurrences, 22 patients have three recurrences, and 14 patients have four recurrences (Output 64.10.5). Parameter estimates for the four marginal models are shown in Output 64.10.6. The 4 DF Wald test (Output 64.10.7) indicates a lack of evidence of a treatment effect in any of the four recurrences (p=0.4105). The optimal weights for estimating the parameter of the common treatment effect are 0.67684, 0.25723, –0.07547, and 0.14140 for Trt1, Trt2, Trt3, and Trt4, respectively, which gives a parameter estimate of –0.5489 with a standard error estimate of 0.2853. A more sensitive test for a treatment effect is the 1 DF test based on this common parameter; however, there is still insufficient evidence for such effect at the 0.05 level (p=0.0543).

Output 64.10.5 Summary of Bladder Tumor Recurrences in 86 Patients

Wei-Lin-Weissfeld Model

The PHREG Procedure

Summary of the Number of Event and Censored Values

Stratum	Visit	Total	Event	Censored	Percent Censored
1	1	86	47	39	45.35
2	2	86	29	57	66.28
3	3	86	22	64	74.42
4	4	86	14	72	83.72
Total		344	112	232	67.44

Output 64.10.6 Analysis of Marginal Cox Models

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	StdErr Ratio	Chi-Square	Pr > ChiSq	Hazard Ratio
Trt1	1	-0.51762	0.30750	0.974	2.8336	0.0923	0.596
Trt2	1	-0.61944	0.36391	0.926	2.8975	0.0887	0.538
Trt3	1	-0.69988	0.41516	0.903	2.8419	0.0918	0.497
Trt4	1	-0.65079	0.48971	0.848	1.7661	0.1839	0.522
Number1	1	0.23599	0.07208	0.947	10.7204	0.0011	1.266
Number2	1	0.13756	0.08690	0.946	2.5059	0.1134	1.147
Number3	1	0.16984	0.10356	0.984	2.6896	0.1010	1.185
Number4	1	0.32880	0.11382	0.909	8.3453	0.0039	1.389
Size1	1	0.06789	0.08529	0.842	0.6336	0.4260	1.070
Size2	1	-0.07612	0.11812	0.881	0.4153	0.5193	0.927
Size3	1	-0.21131	0.17198	0.943	1.5097	0.2192	0.810
Size4	1	-0.20317	0.19106	0.830	1.1308	0.2876	0.816

Output 64.10.7 Tests of Treatment Effects

Wei-Lin-Weissfeld Model

The PHREG Procedure

Linear Coefficients for Test TREATMENT

Parameter	Row 1	Row 2	Row 3	Row 4	Average Effect
Trt1	1	0	0	0	0.67684
Trt2	0	1	0	0	0.25723
Trt3	0	0	1	0	-0.07547
Trt4	0	0	0	1	0.14140
Number1	0	0	0	0	0.00000
Number2	0	0	0	0	0.00000
Number3	0	0	0	0	0.00000
Number4	0	0	0	0	0.00000
Size1	0	0	0	0	0.00000
Size2	0	0	0	0	0.00000
Size3	0	0	0	0	0.00000
Size4	0	0	0	0	0.00000
CONSTANT	0	0	0	0	0.00000

Test TREATMENT Results

Wald Chi-Square	DF	Pr > ChiSq
3.9668	4	0.4105

Average Effect for Test TREATMENT

Estimate	Standard Error	z-Score	Pr > |z|
-0.5489	0.2853	-1.9240	0.0543

Example 64.11: Analysis of Clustered Data

When experimental units are naturally or artificially clustered, failure times of experimental units within a cluster are correlated. Lee, Wei, and Amato (1992) estimate the regression parameters in the Cox model by the maximum partial likelihood estimates under an independent working assumption and use a robust sandwich covariance matrix estimate to account for the intracluster dependence. A subset of data from the Diabetic Retinopathy Study (DRS) is used to illustrate the methodology as in Lin (1994).

The following DATA step creates the data set Blind that represents 197 diabetic patients who have a high risk of experiencing blindness in both eyes as defined by DRS criteria. One eye of each patient

is treated with laser photocoagulation. The hypothesis of interest is whether the laser treatment delays the occurrence of blindness. Since juvenile and adult diabetes have very different courses, it is also desirable to examine how the age of onset of diabetes might affect the time of blindness. Since there are no biological differences between the left eye and the right eye, it is natural to assume a common baseline hazard function for the failure times of the left and right eyes.

Each patient is a cluster that contributes two observations to the input data set, one for each eye. The following variables are in the input data set Blind:

- ID, patient's identification
- Time, failure time
- Status, event indicator (0=censored and 1=uncensored)
- Treatment, treatment received (1=laser photocoagulation and 0=otherwise)
- DiabeticType, type of diabetes (0=juvenile onset with age of onset at 20 or under, and 1= adult onset with age of onset over 20)

```
data Blind;
   input ID Time Status DiabeticType Treatment @@;
   datalines;
   5 46.23 0 1 1    5 46.23 0 1 0   14 42.50 0 0 1   14 31.30 1 0 0
  16 42.27 0 0 1   16 42.27 0 0 0   25 20.60 0 0 1   25 20.60 0 0 0
  29 38.77 0 0 1   29  0.30 1 0 0   46 65.23 0 0 1   46 54.27 1 0 0
  49 63.50 0 0 1   49 10.80 1 0 0   56 23.17 0 0 1   56 23.17 0 0 0
  61  1.47 0 0 1   61  1.47 0 0 0   71 58.07 0 1 1   71 13.83 1 1 0
 100 46.43 1 1 1  100 48.53 0 1 0  112 44.40 0 1 1  112  7.90 1 1 0
 120 39.57 0 1 1  120 39.57 0 1 0  127 30.83 1 1 1  127 38.57 1 1 0
 133 66.27 0 1 1  133 14.10 1 1 0  150 20.17 1 0 1  150  6.90 1 0 0
 167 58.43 0 1 1  167 41.40 1 1 0  176 58.20 0 0 1  176 58.20 0 0 0

... more lines ...

1727 49.97 0 1 1 1727  2.90 1 1 0 1746 45.90 0 0 1 1746  1.43 1 0 0
1749 41.93 0 1 1 1749 41.93 0 1 0
;
run;
```

As a preliminary analysis, PROC FREQ is used to break down the numbers of blindness in the control and treated eyes:

```
proc freq data=Blind;
   table Treatment*Status;
   run;
```

By the end of the study, 54 treated eyes and 101 untreated eyes have developed blindness (Output 64.11.1).

Output 64.11.1 Breakdown of Blindness in the Control and Treated Groups

Wei-Lin-Weissfeld Model

The FREQ Procedure

Table of Treatment by Status

Treatment	Status		
Frequency | Percent | Row Pct | Col Pct	0	1	Total
0	96	101	197
	24.37	25.63	50.00
	48.73	51.27	
	40.17	65.16	
1	143	54	197
	36.29	13.71	50.00
	72.59	27.41	
	59.83	34.84	
Total	239	155	394
	60.66	39.34	100.00

The analysis of Lee, Wei, and Amato (1992) can be carried out by the following PROC PHREG specification. The explanatory variables in this Cox model are Treatment, DiabeticType, and the Treatment × DiabeticType interaction. The COVS(AGGREGATE) is specified to compute the robust sandwich covariance matrix estimate.

```
proc phreg data=Blind covs(aggregate) namelen=22;
   model Time*Status(0)=Treatment DiabeticType Treatment*DiabeticType;
   id ID;
   run;
```

The robust standard error estimates are smaller than the model-based counterparts (Output 64.11.2), since the ratio of the robust standard error estimate relative to the model-based estimate is less than 1 for each variable. Laser photocoagulation appears to be effective (p=0.0217) in delaying the occurrence of blindness. The effect is much more prominent for adult-onset diabetes than for juvenile-onset diabetes.

Output 64.11.2 Inference Based on the Robust Sandwich Covariance

Wei-Lin-Weissfeld Model

The PHREG Procedure

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	StdErr Ratio	Chi-Square	Pr > ChiSq
Treatment	1	-0.42467	0.18497	0.850	5.2713	0.0217
DiabeticType	1	0.34084	0.19558	0.982	3.0371	0.0814
Treatment*DiabeticType	1	-0.84566	0.30353	0.865	7.7622	0.0053

Analysis of Maximum Likelihood Estimates

Parameter	Hazard Ratio	Label
Treatment	.	
DiabeticType	.	
Treatment*DiabeticType	.	Treatment * DiabeticType

Example 64.12: Model Assessment Using Cumulative Sums of Martingale Residuals

The Mayo liver disease example of Lin, Wei, and Ying (1993) is reproduced here to illustrate the checking of the functional form of a covariate and the assessment of the proportional hazards assumption. The data represent 418 patients with primary biliary cirrhosis (PBC), among whom 161 had died as of the date of data listing. A subset of the variables is saved in the SAS data set Liver. The data set contains the following variables:

- Time, follow-up time, in years
- Status, event indicator with value 1 for death time and value 0 for censored time
- Age, age in years from birth to study registration
- Albumin, serum albumin level, in gm/dl
- Bilirubin, serum bilirubin level, in mg/dl
- Edema, edema presence
- Protime, prothrombin time, in seconds

The following statements create the data set Liver:

```
data Liver;
   input Time Status Age Albumin Bilirubin Edema Protime @@;
   label Time="Follow-up Time in Years";
   Time= Time / 365.25;
   datalines;
  400 1 58.7652 2.60 14.5 1.0 12.2 4500 0 56.4463 4.14  1.1 0.0 10.6
 1012 1 70.0726 3.48  1.4 0.5 12.0 1925 1 54.7406 2.54  1.8 0.5 10.3
 1504 0 38.1054 3.53  3.4 0.0 10.9 2503 1 66.2587 3.98  0.8 0.0 11.0
 1832 0 55.5346 4.09  1.0 0.0  9.7 2466 1 53.0568 4.00  0.3 0.0 11.0
 2400 1 42.5079 3.08  3.2 0.0 11.0   51 1 70.5599 2.74 12.6 1.0 11.5
 3762 1 53.7139 4.16  1.4 0.0 12.0  304 1 59.1376 3.52  3.6 0.0 13.6
 3577 0 45.6893 3.85  0.7 0.0 10.6 1217 1 56.2218 2.27  0.8 1.0 11.0

... more lines ...

 1103 0 39.0000 3.83  0.9 0.0 11.2 1055 0 57.0000 3.42  1.6 0.0  9.9
  691 0 58.0000 3.75  0.8 0.0 10.4  976 0 53.0000 3.29  0.7 0.0 10.6
;
run;
```

Consider fitting a Cox model for the survival time of the PCB patients with the covariates Bilirubin, log(Protime), log(Albumin), Age, and Edema. The log transform, which is often applied to blood chemistry measurements, is deliberately not employed for Bilirubin. It is of interest to assess the functional form of the variable Bilirubin in the Cox model. The specifications are as follows:

```
ods graphics on;
proc phreg data=Liver;
   model Time*Status(0)=Bilirubin logProtime logAlbumin Age Edema;
   logProtime=log(Protime);
   logAlbumin=log(Albumin);
   assess var=(Bilirubin) /  resample seed=7548;
   run;
ods graphics off;
```

The ASSESS statement creates a plot of the cumulative martingale residuals against the values of the covariate Bilirubin, which is specified in the VAR= option. The RESAMPLE option computes the p-value of a Kolmogorov-type supremum test based on a sample of 1,000 simulated residual patterns.

Parameter estimates of the model fit are shown in Output 64.12.1. The plot in Output 64.12.2 displays the observed cumulative martingale residual process for Bilirubin together with 20 simulated realizations from the null distribution. This graphical display is requested by specifying the **ods graphics on** statement and the ASSESS statement. It is obvious that the observed process is atypical compared to the simulated realizations. Also, none of the 1,000 simulated realizations has an absolute maximum exceeding that of the observed cumulative martingale residual process. Both the graphical and numerical results indicate that a transform is deemed necessary for Bilirubin in the model.

Output 64.12.1 Cox Model with Bilirubin as a Covariate

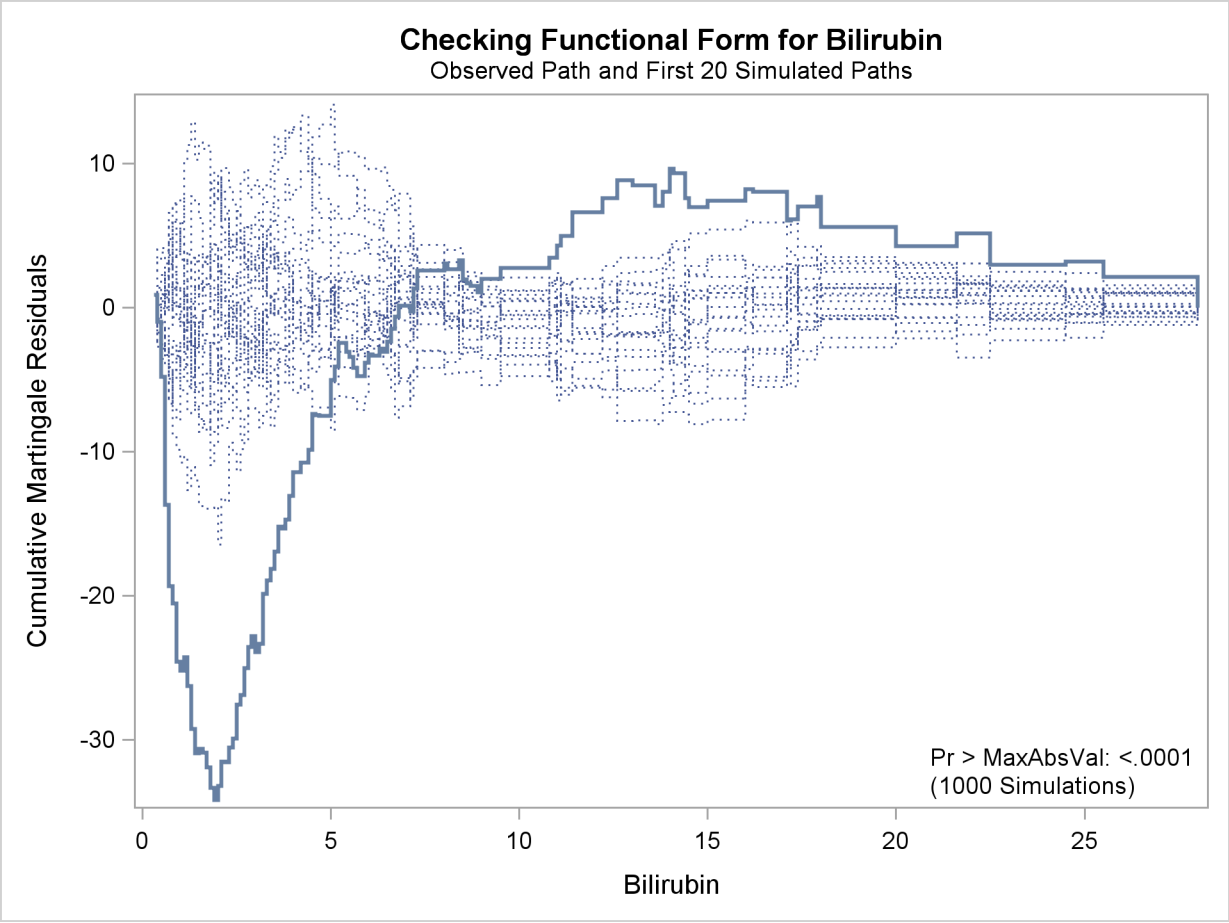
Wei-Lin-Weissfeld Model

The PHREG Procedure

Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
Bilirubin	1	0.11733	0.01298	81.7567	<.0001	1.124
logProtime	1	2.77581	0.71482	15.0794	0.0001	16.052
logAlbumin	1	-3.17195	0.62945	25.3939	<.0001	0.042
Age	1	0.03779	0.00805	22.0288	<.0001	1.039
Edema	1	0.84772	0.28125	9.0850	0.0026	2.334

Output 64.12.2 Cumulative Martingale Residuals vs Bilirubin



The cumulative martingale residual plots in Output 64.12.3 provide guidance in suggesting a more appropriate functional form for a covariate. The four curves were created from simple forms of misspecification by using 1,000 simulated times from a exponential model with 20% censoring. The true and fitted models are shown in Table 64.9. The following statements produce Output 64.12.3.

```
data sim(drop=tmp);
   p = 1 / 91;
   seed = 1;
   do n = 1 to 10000;
      x1 = rantbl( seed, p, p, p, p, p, p, p, p, p, p,
                           p, p, p, p, p, p, p, p, p, p,
                           p, p, p, p, p, p, p, p, p, p,
                           p, p, p, p, p, p, p, p, p, p,
                           p, p, p, p, p, p, p, p, p, p,
                           p, p, p, p, p, p, p, p, p, p,
                           p, p, p, p, p, p, p, p, p, p,
                           p, p, p, p, p, p, p, p, p, p,
                           p, p, p, p, p, p, p, p, p, p );

      x1 = 1 + ( x1 - 1 ) / 10;
      x2= x1 * x1;
      x3= x1 * x2;
      status= rantbl(seed, .8);
      tmp= log(1-ranuni(seed));
      t1= -exp(-log(x1)) * tmp;
      t2= -exp(-.1*(x1+x2)) * tmp;
      t3= -exp(-.01*(x1+x2+x3)) * tmp;
      tt= -exp(-(x1>5)) * tmp;
      output;
   end;
   run;
proc sort data=sim;
   by x1;
   run;
proc phreg data=sim noprint;
   model t1*status(2)=x1;
   output out=out1 resmart=resmart;
   run;
proc phreg data=sim noprint;
   model t2*status(2)=x1;
   output out=out2 resmart=resmart;
   run;

proc phreg data=sim noprint;
   model t3*status(2)=x1 x2;
   output out=out3 resmart=resmart;
   run;

proc phreg data=sim noprint;
   model tt*status(2)=x1;
   output out=out4 resmart=resmart;
   run;
```

```
data out1(keep=x1 cresid1);
   retain cresid1 0;
   set out1;
   by x1;
   cresid1 + resmart;
   if last.x1  then output;
   run;
data out2(keep=x1 cresid2);
   retain cresid2 0;
   set out2;
   by x1;
   cresid2 + resmart;
   if last.x1  then output;
   run;
data out3(keep=x1 cresid3);
   retain cresid3 0;
   set out3;
   by x1;
   cresid3 + resmart;
   if last.x1  then output;
   run;
data out4(keep=x1 cresid4);
   retain cresid4 0;
   set out4;
   by x1;
   cresid4 + resmart;
   if last.x1  then output;
   run;
data all;
   set out1;
   set out2;
   set out3;
   set out4;
   run;

proc template;
   define statgraph MisSpecification;
      BeginGraph;
         entrytitle "Covariate Misspecification";
         layout lattice / columns=2 rows=2 columndatarange=unionall;

            columnaxes;
               columnaxis / display=(ticks tickvalues label) label="x";
               columnaxis / display=(ticks tickvalues label) label="x";
            endcolumnaxes;

            cell;
               cellheader;
                  entry "(a) Data: log(X), Model: X";
               endcellheader;
               layout overlay / xaxisopts=(display=none)
                                yaxisopts=(label="Cumulative Residual");
                  seriesplot y=cresid1 x=x1 / lineattrs=GraphFit;
               endlayout;
```

```
            endcell;

            cell;
               cellheader;
                  entry "(b) Data: X*X, Model: X";
               endcellheader;
               layout overlay / xaxisopts=(display=none)
                                yaxisopts=(label=" ");
                  seriesplot y=cresid2 x=x1 / lineattrs=GraphFit;
               endlayout;
            endcell;

            cell;
                cellheader;
                   entry "(c) Data: X*X*X, Model: X*X";
                endcellheader;
                layout overlay / xaxisopts=(display=none)
                                 yaxisopts=(label="Cumulative Residual");
                   seriesplot y=cresid3 x=x1 / lineattrs=GraphFit;
                endlayout;
            endcell;

            cell;
               cellheader;
                  entry "(d) Data: I(X>5), Model: X";
               endcellheader;
               layout overlay / xaxisopts=(display=none)
                                yaxisopts=(label=" ");
                  seriesplot y=cresid4 x=x1 / lineattrs=GraphFit;
               endlayout;
            endcell;

         endlayout;
      EndGraph;
   end;

proc sgrender data=all template="MisSpecification";
   run;
```

Output 64.12.3 Typical Cumulative Residual Plot Patterns

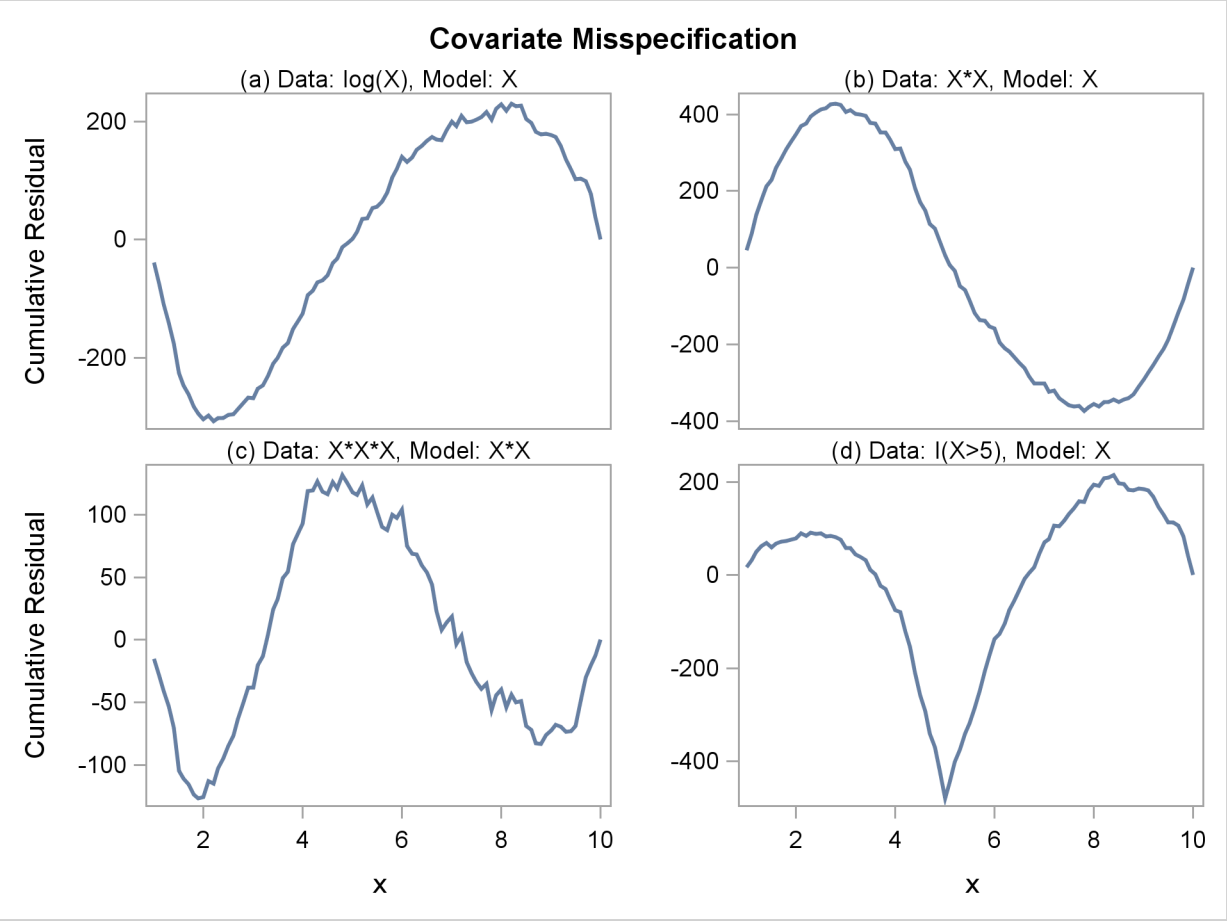


Table 64.9 Model Misspecifications

Plot	Data	Fitted Model
(a)	$\log(X)$	X
(b)	$\{X, X^2\}$	X
(c)	$\{X, X^2, X^3\}$	$\{X, X^2\}$
(d)	$I(X > 5)$	X

The curve of observed cumulative martingale residuals in Output 64.12.2 most resembles the behavior of the curve in plot (a) of Output 64.12.3, indicating that log(Bilirubin) might be a more appropriate term in the model than Bilirubin.

Next, the analysis of the natural history of the PBC is repeated with log(Bilirubin) replacing Bilirubin, and the functional form of log(Bilirubin) is assessed. Also assessed is the proportional hazards assumption for the Cox model. The analysis is carried out by the following statements:

```
ods graphics on;
proc phreg data=Liver;
   model Time*Status(0)=logBilirubin logProtime logAlbumin Age Edema;
   logBilirubin=log(Bilirubin);
   logProtime=log(Protime);
   logAlbumin=log(Albumin);
   assess var=(logBilirubin) ph / crpanel resample seed=19;
   run;
ods html close;
```

The SEED= option specifies a integer seed for generating random numbers. The CRPANEL option in the ASSESS statement requests a panel of four plots. Each plot displays the observed cumulative martingale residual process along with two simulated realizations. The PH option checks the proportional hazards assumption of the model by plotting the observed standardized score process with 20 simulated realizations for each covariate in the model.

Output 64.12.4 displays the parameter estimates of the fitted model. The cumulative martingale residual plots in Output 64.12.5 and Output 64.12.6 show that the observed martingale residual process is more typical of the simulated realizations. The p-value for the Kolmogorov-type supremum test based on 1,000 simulations is 0.052, indicating that the log transform is a much improved functional form for Bilirubin.

Output 64.12.4 Model with log(Bilirubin) as a Covariate

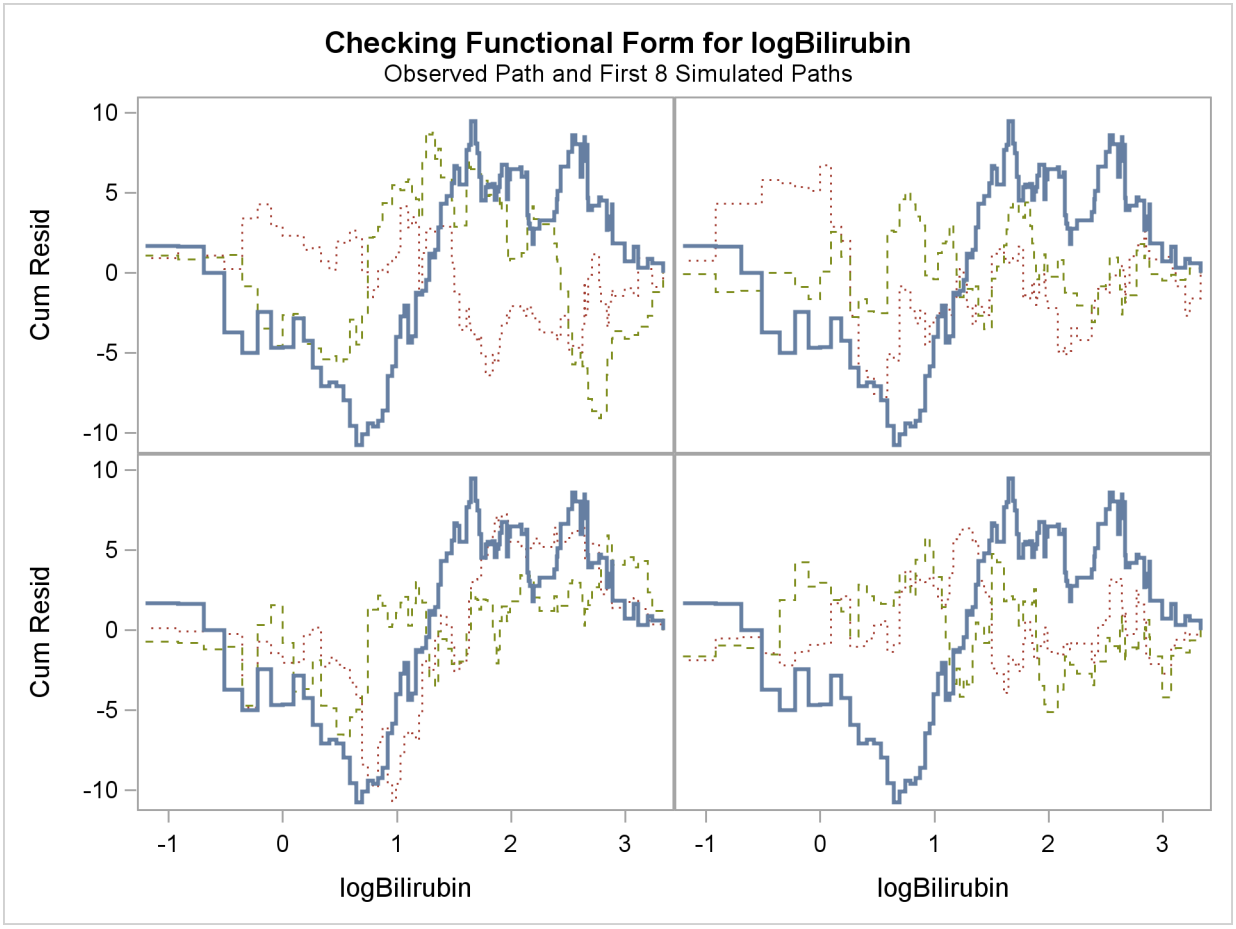
Wei-Lin-Weissfeld Model

The PHREG Procedure

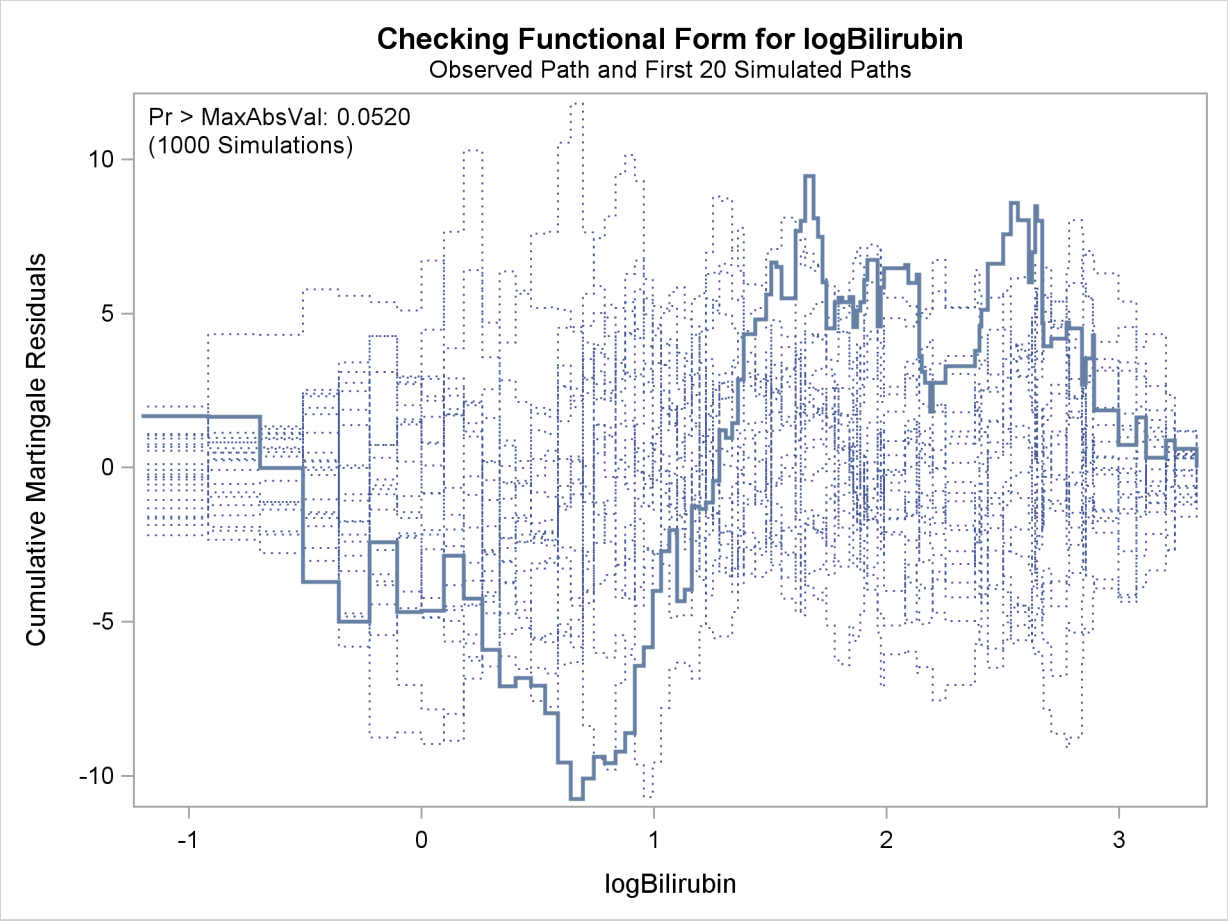
Analysis of Maximum Likelihood Estimates

Parameter	DF	Parameter Estimate	Standard Error	Chi-Square	Pr > ChiSq	Hazard Ratio
logBilirubin	1	0.87072	0.08263	111.0484	<.0001	2.389
logProtime	1	2.37789	0.76674	9.6181	0.0019	10.782
logAlbumin	1	-2.53264	0.64819	15.2664	<.0001	0.079
Age	1	0.03940	0.00765	26.5306	<.0001	1.040
Edema	1	0.85934	0.27114	10.0447	0.0015	2.362

Output 64.12.5 Panel Plot of Cumulative Martingale Residuals versus log(Bilirubin)

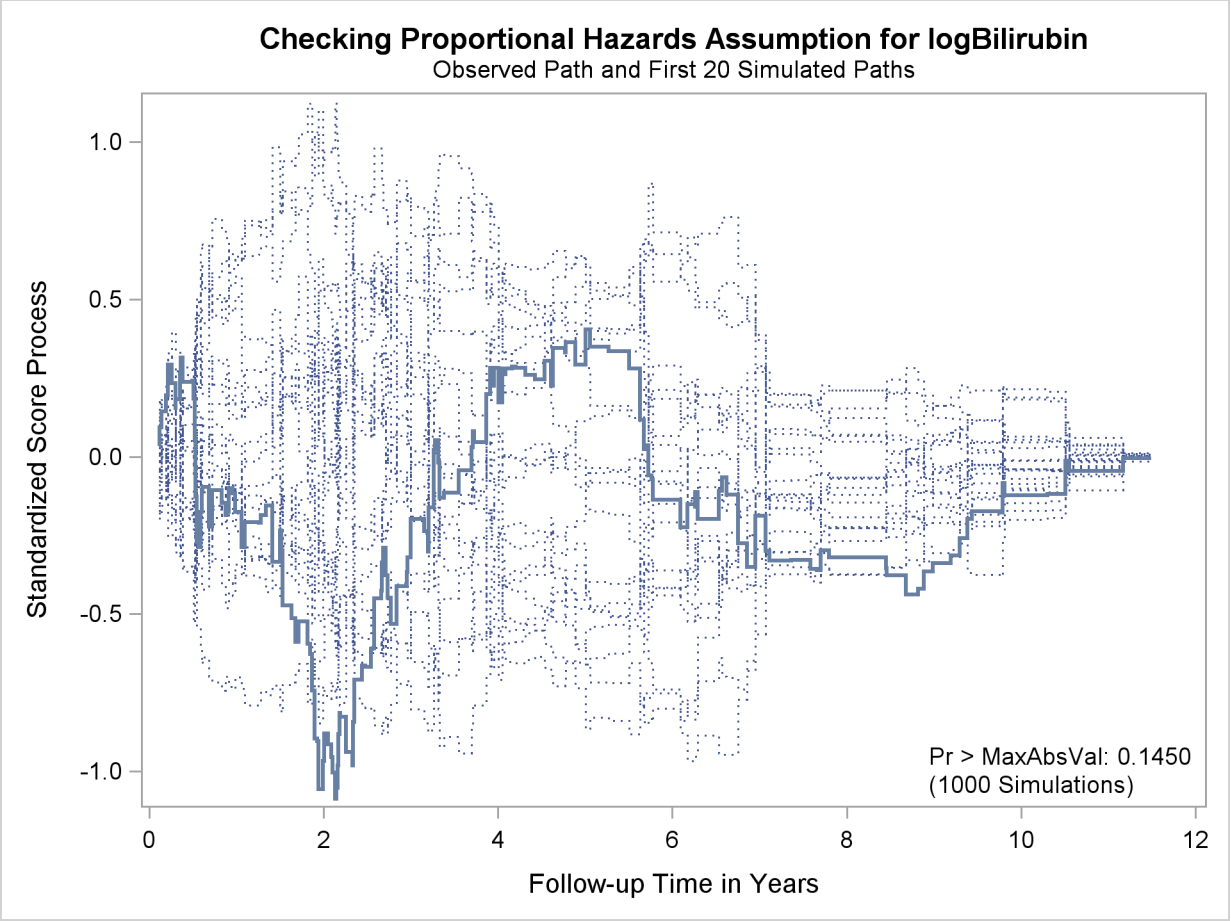


Output 64.12.6 Cumulative Martingale Residuals versus log(Bilirubin)



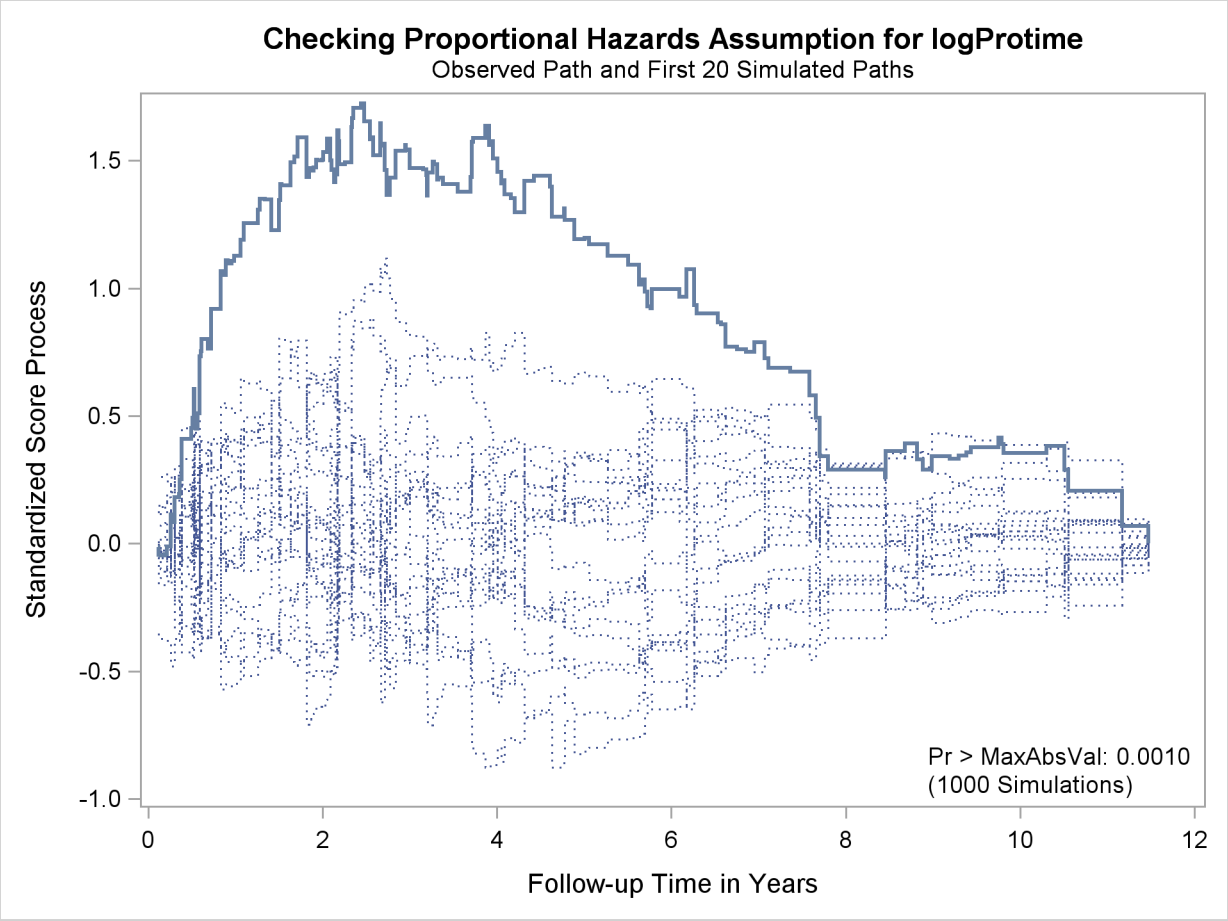
Output 64.12.7 and Output 64.12.8 display the results of proportional hazards assumption assessment for log(Bilirubin) and log(Protime), respectively. The latter plot reveals nonproportional hazards for log(Protime).

Output 64.12.7 Standardized Score Process for log(Bilirubin)



[

Output 64.12.8 Standardized Score Process for log(Protime)



Plots for log(Albumin), Age, and Edema are not shown here. The Kolmogorov-type supremum test results for all the covariates are shown in Output 64.12.9. In addition to log(Protime), the proportional hazards assumption appears to be violated for Edema.

Output 64.12.9 Kolmogorov-Type Supremum Tests for Proportional Hazards Assumption

Supremum Test for Proportionals Hazards Assumption

Variable	Maximum Absolute Value	Replications	Seed	Pr > MaxAbsVal
logBilirubin	1.0880	1000	19	0.1450
logProtime	1.7243	1000	19	0.0010
logAlbumin	0.8443	1000	19	0.4330
Age	0.7387	1000	19	0.4620
Edema	1.4350	1000	19	0.0330

Example 64.13: Bayesian Analysis of the Cox Model

This example illustrates the use of an informative prior. Hazard ratios, which are transformations of the regression parameters, are useful for interpreting survival models. This example also demonstrates the use of the HAZARDRATIO statement to obtain customized hazard ratios.

Consider the VALung data set in Example 64.3. In this example, the Cox model is used for the Bayesian analysis. The parameters are the coefficients of the continuous explanatory variables (Kps, Duration, and Age) and the coefficients of the design variables for the categorical explanatory variables (Prior, Cell, and Therapy). You use the CLASS statement in PROC PHREG to specify the categorical variables and their reference levels. Using the default reference parameterization, the design variables for the categorical variables are Prioryes (for Prior with Prior='no' as reference), Celladeno, Cellsmall, Cellsquamous (for Cell with Cell='large' as reference), and Therapytest (for Therapy='standard' as reference).

Consider the explanatory variable Kps. The Karnofsky performance scale index enables patients to be classified according to their functional impairment. The scale can range from 0 to 100—0 for dead, and 100 for a normal, healthy person with no evidence of disease. Recall that a flat prior was used for the regression coefficient in the example in the section "Bayesian Analysis" on page 4531. A flat prior on the Kps coefficient implies that the coefficient is as likely to be 0.1 as it is to be –100000. A coefficient of –5 means that a decrease of 20 points in the scale increases the hazard by $e^{-20\times-5}$(=2.68×10^{43})-fold, which is a rather unreasonable and unrealistic expectation for the effect of the Karnofsky index, much less than the value of –100000. Suppose you have a more realistic expectation: the effect is somewhat small and is more likely to be negative than positive, and a decrease of 20 points in the Karnofsky index will change the hazard from 0.9-fold (some minor positive effect) to 4-fold (a large negative effect). You can convert this opinion to a more informative prior on the Kps coefficient $\boldsymbol{\beta}_1$. Mathematically,

$$0.9 < e^{-20\boldsymbol{\beta}_1} < 4$$

which is equivalent to

$$-0.0693 < \boldsymbol{\beta}_1 < 0.0053$$

This becomes the plausible range that you believe the Kps coefficient can take. Now you can find a normal distribution that best approximates this belief by placing the majority of the prior distribution mass within this range. Assuming this interval is $\mu \pm 2\sigma$, where μ and σ are the mean and standard deviation of the normal prior, respectively, the hyperparameters μ and σ are computed as follows:

$$\begin{aligned} \mu &= \frac{-0.0693 + 0.0053}{2} = -0.032 \\ \sigma &= \frac{0.0053 - (-0.0693)}{4} = 0.0186 \end{aligned}$$

Note that a normal prior distribution with mean –0.0320 and standard deviation 0.0186 indicates that you believe, before looking at the data, that a decrease of 20 points in the Karnofsky index will probably change the hazard rate by 0.9-fold to 4-fold. This does not rule out the possibility that the Kps coefficient can take a more extreme value such as –5, but the probability of having such extreme values is very small.

Assume the prior distributions are independent for all the parameters. For the coefficient of Kps, you use a normal prior distribution with mean –0.0320 and variance 0.0186^2(=0.00035). For other parameters, you resort to using a normal prior distribution with mean 0 and variance 10^6, which is fairly noninformative. Means and variances of these independent normal distributions are saved in the data set Prior as follows:

```
proc format;
   value yesno 0='no' 10='yes';
   run;

data VALung;
   drop check m;
   retain Therapy Cell;
   infile cards column=column;
   length Check $ 1;
   label Time='time to death in days'
         Kps='Karnofsky performance scale'
         Duration='months from diagnosis to randomization'
         Age='age in years'
         Prior='prior therapy'
         Cell='cell type'
         Therapy='type of treatment';
   format Prior yesno.;
   M=Column;
   input Check $ @@;
   if M>Column then M=1;
   if Check='s'|Check='t' then do;
      input @M Therapy $ Cell $;
      delete;
      end;
   else do;
      input @M Time Kps Duration Age Prior @@;
      Status=(Time>0);
      Time=abs(Time);
      end;
   datalines;
standard squamous
 72 60  7 69  0   411 70  5 64 10   228 60  3 38  0   126 60  9 63 10
118 70 11 65 10    10 20  5 49  0    82 40 10 69 10   110 80 29 68  0
314 50 18 43  0  -100 70  6 70  0    42 60  4 81  0     8 40 58 63 10
144 30  4 63  0   -25 80  9 52 10    11 70 11 48 10
standard small
 30 60  3 61  0   384 60  9 42  0     4 40  2 35  0    54 80  4 63 10
 13 60  4 56  0  -123 40  3 55  0   -97 60  5 67  0   153 60 14 63 10
 59 30  2 65  0   117 80  3 46  0    16 30  4 53 10   151 50 12 69  0
 22 60  4 68  0    56 80 12 43 10    21 40  2 55 10    18 20 15 42  0
139 80  2 64  0    20 30  5 65  0    31 75  3 65  0    52 70  2 55  0
287 60 25 66 10    18 30  4 60  0    51 60  1 67  0   122 80 28 53  0
 27 60  8 62  0    54 70  1 67  0     7 50  7 72  0    63 50 11 48  0
392 40  4 68  0    10 40 23 67 10
```

```
standard adeno
  8 20 19 61 10    92 70 10 60  0    35 40  6 62  0   117 80  2 38  0
132 80  5 50  0    12 50  4 63 10   162 80  5 64  0     3 30  3 43  0
 95 80  4 34  0
standard large
177 50 16 66 10   162 80  5 62  0   216 50 15 52  0   553 70  2 47  0
278 60 12 63  0    12 40 12 68 10   260 80  5 45  0   200 80 12 41 10
156 70  2 66  0  -182 90  2 62  0   143 90  8 60  0   105 80 11 66  0
103 80  5 38  0   250 70  8 53 10   100 60 13 37 10
test squamous
999 90 12 54 10   112 80  6 60  0   -87 80  3 48  0  -231 50  8 52 10
242 50  1 70  0   991 70  7 50 10   111 70  3 62  0     1 20 21 65 10
587 60  3 58  0   389 90  2 62  0    33 30  6 64  0    25 20 36 63  0
357 70 13 58  0   467 90  2 64  0   201 80 28 52 10     1 50  7 35  0
 30 70 11 63  0    44 60 13 70 10   283 90  2 51  0    15 50 13 40 10
test small
 25 30  2 69  0  -103 70 22 36 10    21 20  4 71  0    13 30  2 62  0
 87 60  2 60  0     2 40 36 44 10    20 30  9 54 10     7 20 11 66  0
 24 60  8 49  0    99 70  3 72  0     8 80  2 68  0    99 85  4 62  0
 61 70  2 71  0    25 70  2 70  0    95 70  1 61  0    80 50 17 71  0
 51 30 87 59 10    29 40  8 67  0
test adeno
 24 40  2 60  0    18 40  5 69 10   -83 99  3 57  0    31 80  3 39  0
 51 60  5 62  0    90 60 22 50 10    52 60  3 43  0    73 60  3 70  0
  8 50  5 66  0    36 70  8 61  0    48 10  4 81  0     7 40  4 58  0
140 70  3 63  0   186 90  3 60  0    84 80  4 62 10    19 50 10 42  0
 45 40  3 69  0    80 40  4 63  0
test large
 52 60  4 45  0   164 70 15 68 10    19 30  4 39 10    53 60 12 66  0
 15 30  5 63  0    43 60 11 49 10   340 80 10 64 10   133 75  1 65  0
111 60  5 64  0   231 70 18 67 10   378 80  4 65  0    49 30  3 37  0
;

data Prior;
   input _TYPE_ $ Kps Duration Age Prioryes Celladeno Cellsmall
         Cellsquamous Therapytest;
   datalines;
   Mean -0.0320 0 0 0 0 0 0 0
   Var  0.00035 1e6 1e6 1e6 1e6 1e6 1e6 1e6
   ;
run;
```

In the following BAYES statement, COEFFPRIOR=NORMAL(INPUT=Prior) specifies the normal prior distribution for the regression coefficients with details contained in the data set Prior. Summary statistics of the posterior distribution are produced by default. Autocorrelations and effective sample size are requested as convergence diagnostics as well as the trace plots for visual analysis. For comparisons of hazards, three HAZARDRATIO statements are specified—one for the variable Therapy, one for the variable Age, and one for the variable Cell.

```
ods graphics on;
proc phreg data=VALung;
   class Prior(ref='no') Cell(ref='large') Therapy(ref='standard');
   model Time*Status(0) = Kps Duration Age Prior Cell Therapy;
   bayes seed=1 coeffprior=normal(input=Prior) diagnostic=(autocorr ess)
         plots=trace;
   hazardratio 'Hazard Ratio Statement 1' Therapy;
   hazardratio 'Hazard Ratio Statement 2' Age / unit=10;
   hazardratio 'Hazard Ratio Statement 3' Cell;
run;
ods graphics off;
```

This analysis generates a posterior chain of 10,000 iterations after 2,000 iterations of burn-in, as depicted in Output 64.13.1.

Output 64.13.1 Model Information

The PHREG Procedure

Bayesian Analysis

Model Information

Data Set	WORK.VALUNG	
Dependent Variable	Time	time to death in days
Censoring Variable	Status	
Censoring Value(s)	0	
Model	Cox	
Ties Handling	BRESLOW	
Burn-In Size	2000	
MC Sample Size	10000	
Thinning	1	

Output 64.13.2 displays the names of the parameters and their corresponding effects and categories.

Output 64.13.2 Parameter Names

Regression Parameter Information

Parameter	Effect	Prior	Cell	Therapy
Kps	Kps			
Duration	Duration			
Age	Age			
Prioryes	Prior	yes		
Celladeno	Cell		adeno	
Cellsmall	Cell		small	
Cellsquamous	Cell		squamous	
Therapytest	Therapy			test

PROC PHREG computes the maximum likelihood estimates of regression parameters (Output 64.13.3). These estimates are used as the starting values for the simulation of posterior samples.

Output 64.13.3 Parameter Estimates

Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits	
Kps	1	-0.0326	0.00551	-0.0434	-0.0218
Duration	1	-0.00009	0.00913	-0.0180	0.0178
Age	1	-0.00855	0.00930	-0.0268	0.00969
Prioryes	1	0.0723	0.2321	-0.3826	0.5273
Celladeno	1	0.7887	0.3027	0.1955	1.3819
Cellsmall	1	0.4569	0.2663	-0.0650	0.9787
Cellsquamous	1	-0.3996	0.2827	-0.9536	0.1544
Therapytest	1	0.2899	0.2072	-0.1162	0.6961

Output 64.13.4 displays the independent normal prior for the analysis.

Output 64.13.4 Coefficient Prior

Independent Normal Prior for Regression Coefficients

Parameter	Mean	Precision
Kps	-0.032	2857.143
Duration	0	1E-6
Age	0	1E-6
Prioryes	0	1E-6
Celladeno	0	1E-6
Cellsmall	0	1E-6
Cellsquamous	0	1E-6
Therapytest	0	1E-6

Fit statistics are displayed in Output 64.13.5. These statistics are useful for variable selection.

Output 64.13.5 Fit Statistics

Fit Statistics

DIC (smaller is better)	966.260
pD (Effective Number of Parameters)	7.934

Summary statistics of the posterior samples are shown in Output 64.13.6 and Output 64.13.7. These results are quite comparable to the classical results based on maximizing the likelihood as shown in Output 64.13.3, since the prior distribution for the regression coefficients is relatively flat.

Output 64.13.6 Summary Statistics

The PHREG Procedure

Bayesian Analysis

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
Kps	10000	−0.0326	0.00523	−0.0362	−0.0326	−0.0291
Duration	10000	−0.00159	0.00954	−0.00756	−0.00093	0.00504
Age	10000	−0.00844	0.00928	−0.0147	−0.00839	−0.00220
Prioryes	10000	0.0742	0.2348	−0.0812	0.0737	0.2337
Celladeno	10000	0.7881	0.3065	0.5839	0.7876	0.9933
Cellsmall	10000	0.4639	0.2709	0.2817	0.4581	0.6417
Cellsquamous	10000	−0.4024	0.2862	−0.5927	−0.4025	−0.2106
Therapytest	10000	0.2892	0.2038	0.1528	0.2893	0.4240

Output 64.13.7 Interval Statistics

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
Kps	0.050	−0.0429	−0.0222	−0.0433	−0.0226
Duration	0.050	−0.0220	0.0156	−0.0210	0.0164
Age	0.050	−0.0263	0.00963	−0.0265	0.00941
Prioryes	0.050	−0.3936	0.5308	−0.3832	0.5384
Celladeno	0.050	0.1879	1.3920	0.1764	1.3755
Cellsmall	0.050	−0.0571	1.0167	−0.0888	0.9806
Cellsquamous	0.050	−0.9687	0.1635	−0.9641	0.1667
Therapytest	0.050	−0.1083	0.6930	−0.1284	0.6710

With autocorrelations retreating quickly to 0 (Output 64.13.8) and large effective sample sizes (Output 64.13.9), both diagnostics indicate a reasonably good mixing of the Markov chain. The trace plots in Output 64.13.10 also confirm the convergence of the Markov chain.

Output 64.13.8 Autocorrelation Diagnostics

The PHREG Procedure

Bayesian Analysis

Posterior Autocorrelations

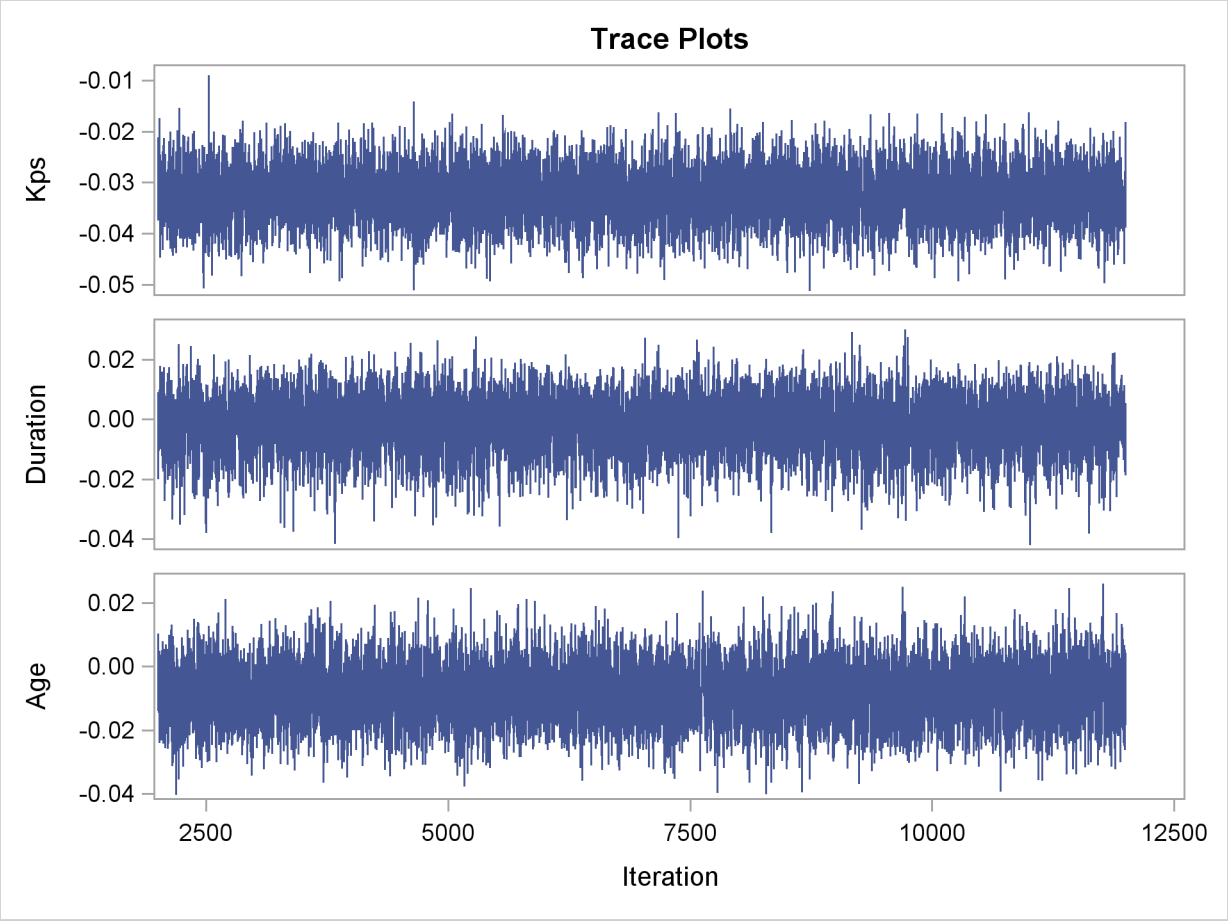
Parameter	Lag 1	Lag 5	Lag 10	Lag 50
Kps	0.1442	−0.0016	0.0096	−0.0013
Duration	0.2672	−0.0054	−0.0004	−0.0011
Age	0.1374	−0.0044	0.0129	0.0084
Prioryes	0.2507	−0.0271	−0.0012	0.0004
Celladeno	0.4160	0.0265	−0.0062	0.0190
Cellsmall	0.5055	0.0277	−0.0011	0.0271
Cellsquamous	0.3586	0.0252	−0.0044	0.0107
Therapytest	0.2063	0.0199	−0.0047	−0.0166

Output 64.13.9 Effective Sample Size Diagnostics

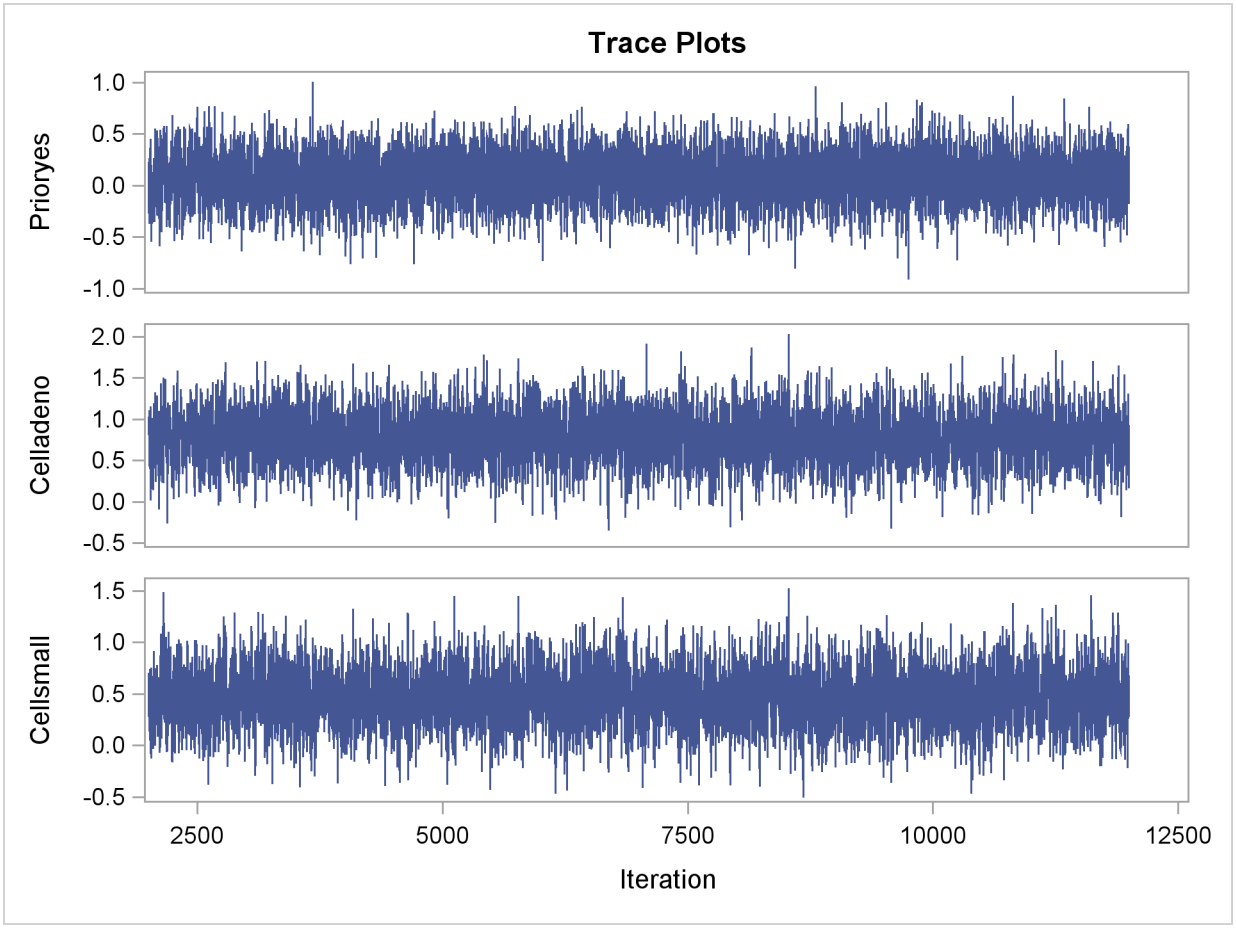
Effective Sample Sizes

Parameter	ESS	Correlation Time	Efficiency
Kps	7046.7	1.4191	0.7047
Duration	5790.0	1.7271	0.5790
Age	7426.1	1.3466	0.7426
Prioryes	6102.2	1.6388	0.6102
Celladeno	3673.4	2.7223	0.3673
Cellsmall	3346.4	2.9883	0.3346
Cellsquamous	4052.8	2.4674	0.4053
Therapytest	6870.8	1.4554	0.6871

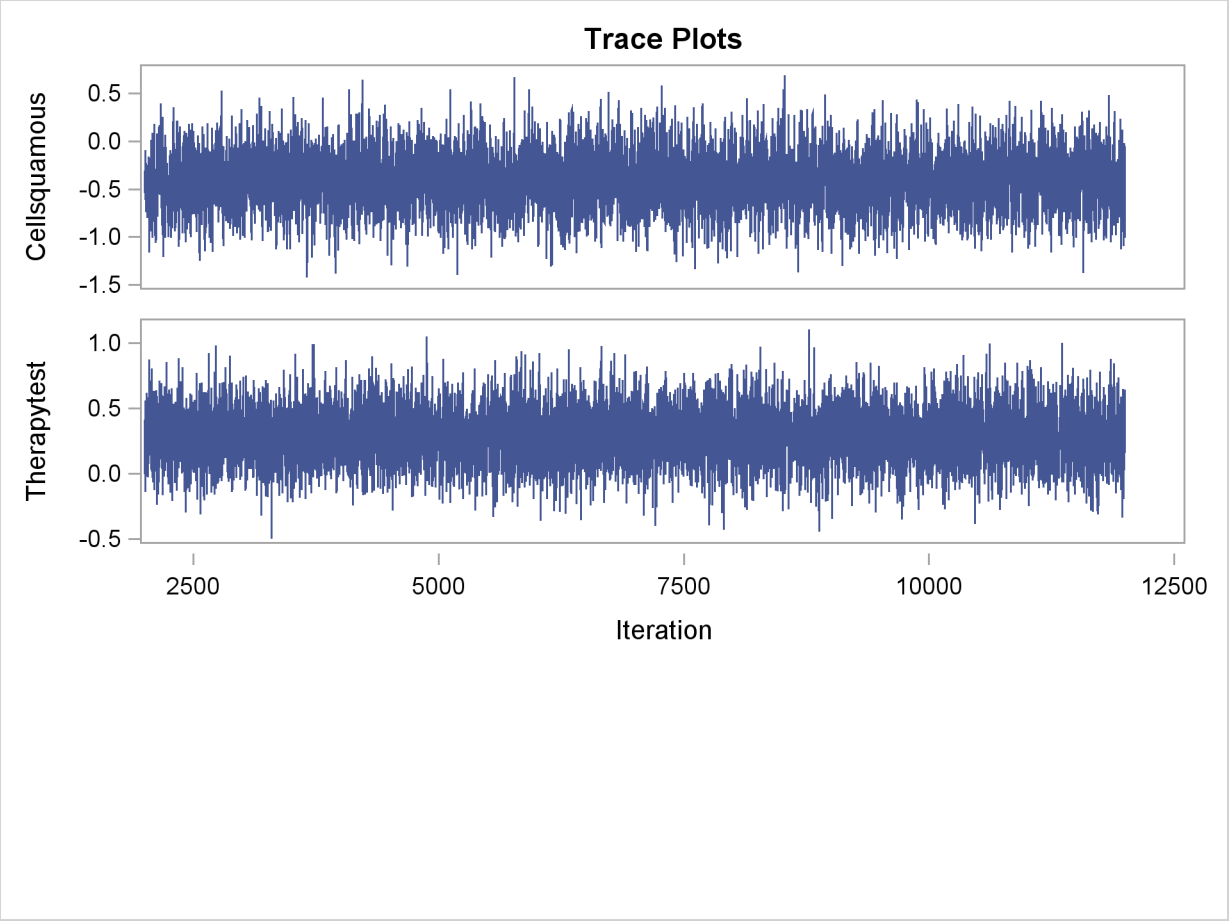
Output 64.13.10 Trace Plots



Output 64.13.10 *continued*



Output 64.13.10 *continued*



The first HAZARDRATIO statement compares the hazards between the standard therapy and the test therapy. Summaries of the posterior distribution of the corresponding hazard ratio are shown in Output 64.13.11. There is a 95% chance that the hazard ratio of standard therapy versus test therapy lies between 0.5 and 1.1.

Output 64.13.11 Hazard Ratio for Treatment

Hazard Ratio Statement 1: Hazard Ratios for Therapy

Description	N	Mean	Standard Deviation
Therapy standard vs test	10000	0.7645	0.1573

Hazard Ratio Statement 1: Hazard Ratios for Therapy

Quantiles			95% Equal-Tail Interval		95% HPD Interval	
25%	50%	75%				
0.6544	0.7488	0.8583	0.5001	1.1143	0.4788	1.0805

The second HAZARDRATIO statement assesses the change of hazards for an increase in Age of 10 years. Summaries of the posterior distribution of the corresponding hazard ratio are shown in Output 64.13.12.

Output 64.13.12 Hazard Ratio for Age

Hazard Ratio Statement 2: Hazard Ratios for Age

Description	N	Mean	Standard Deviation	Quantiles 25%	50%	75%
Age Unit=10	10000	0.9230	0.0859	0.8635	0.9195	0.9782

Hazard Ratio Statement 2: Hazard Ratios for Age

95% Equal-Tail Interval		95% HPD Interval	
0.7685	1.1011	0.7650	1.0960

The third HAZARDRATIO statement compares the changes of hazards between two types of cells. For four types of cells, there are six different pairs of cell comparisons. The results are shown in Output 64.13.13.

Output 64.13.13 Hazard Ratios for Cell

Hazard Ratio Statement 3: Hazard Ratios for Cell

Description	N	Mean	Standard Deviation	Quantiles 25%	50%	75%	95% Equal-Tail Interval		95% HPD Interval	
Cell adeno vs large	10000	2.3048	0.7224	1.7929	2.1982	2.7000	1.2067	4.0227	1.0053	3.7057
Cell adeno vs small	10000	1.4377	0.4078	1.1522	1.3841	1.6704	0.7930	2.3999	0.7309	2.2662
Cell adeno vs squamous	10000	3.4449	1.0745	2.6789	3.2941	4.0397	1.8067	5.9727	1.6303	5.5946
Cell large vs small	10000	0.6521	0.1780	0.5264	0.6325	0.7545	0.3618	1.0588	0.3331	1.0041
Cell large vs squamous	10000	1.5579	0.4548	1.2344	1.4955	1.8089	0.8492	2.6346	0.7542	2.4575
Cell small vs squamous	10000	2.4728	0.7081	1.9620	2.3663	2.8684	1.3789	4.1561	1.2787	3.9263

Example 64.14: Bayesian Analysis of Piecewise Exponential Model

This example illustrates using a piecewise exponential model in a Bayesian analysis. Consider the Rats data set in the section "Getting Started: PHREG Procedure" on page 4526. In the following statements, PROC PHREG is used to carry out a Bayesian analysis for the piecewise exponential model. In the BAYES statement, the option PIECEWISE stipulates a piecewise exponential model, and PIECEWISE=HAZARD requests that the constant hazards be modeled in the original scale. By default, eight intervals of constant hazards are used, and the intervals are chosen such that each has roughly the same number of events.

```
data Rats;
   label Days  ='Days from Exposure to Death';
   input Days Status Group @@;
   datalines;
143 1 0   164 1 0   188 1 0   188 1 0
190 1 0   192 1 0   206 1 0   209 1 0
213 1 0   216 1 0   220 1 0   227 1 0
230 1 0   234 1 0   246 1 0   265 1 0
304 1 0   216 0 0   244 0 0   142 1 1
156 1 1   163 1 1   198 1 1   205 1 1
232 1 1   232 1 1   233 1 1   233 1 1
233 1 1   233 1 1   239 1 1   240 1 1
261 1 1   280 1 1   280 1 1   296 1 1
296 1 1   323 1 1   204 0 1   344 0 1
;
run;

proc phreg data=Rats;
   model Days*Status(0)=Group;
   bayes seed=1 piecewise=hazard;
run;
```

The "Model Information" table in Output 64.14.1 shows that the piecewise exponential model is being used.

Output 64.14.1 Model Information

```
                          The PHREG Procedure

                           Bayesian Analysis

                           Model Information

Data Set                  WORK.RATS
Dependent Variable        Days                      Days from Exposure to Death
Censoring Variable        Status
Censoring Value(s)        0
Model                     Piecewise Exponential
Burn-In Size              2000
MC Sample Size            10000
Thinning                  1
```

By default the time axis is partitioned into eight intervals of constant hazard. Output 64.14.2 details the number of events and observations in each interval. Note that the constant hazard parameters are named Lambda1,..., Lambda8. You can supply your own partition by using the INTERVALS= suboption within the PIECEWISE=HAZARD option.

Output 64.14.2 Interval Partition

Constant Hazard Time Intervals

Interval [Lower,	Upper)	N	Event	Hazard Parameter
0	176	5	5	Lambda1
176	201.5	5	5	Lambda2
201.5	218	7	5	Lambda3
218	232.5	5	5	Lambda4
232.5	233.5	4	4	Lambda5
233.5	253.5	5	4	Lambda6
253.5	288	4	4	Lambda7
288	Infty	5	4	Lambda8

The model parameters consist of the eight hazard parameters Lambda1, ..., Lambda8, and the regression coefficient Group. The maximum likelihood estimates are displayed in Output 64.14.3. Again, these estimates are used as the starting values for simulation of the posterior distribution.

Output 64.14.3 Maximum Likelihood Estimates

Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits	
Lambda1	1	0.000953	0.000443	0.000084	0.00182
Lambda2	1	0.00794	0.00371	0.000672	0.0152
Lambda3	1	0.0156	0.00734	0.00120	0.0300
Lambda4	1	0.0236	0.0115	0.00112	0.0461
Lambda5	1	0.3669	0.1959	-0.0172	0.7509
Lambda6	1	0.0276	0.0148	-0.00143	0.0566
Lambda7	1	0.0262	0.0146	-0.00233	0.0548
Lambda8	1	0.0545	0.0310	-0.00626	0.1152
Group	1	-0.6223	0.3468	-1.3020	0.0573

Without using the PRIOR= suboption within the PIECEWISE=HAZARD option to specify the prior of the hazard parameters, the default is to use the noninformative and improper prior displayed in Output 64.14.4.

Output 64.14.4 Hazard Prior

Improper Prior for Hazards

Parameter	Prior
Lambda1	1 / Lambda1
Lambda2	1 / Lambda2
Lambda3	1 / Lambda3
Lambda4	1 / Lambda4
Lambda5	1 / Lambda5
Lambda6	1 / Lambda6
Lambda7	1 / Lambda7
Lambda8	1 / Lambda8

The noninformative uniform prior is used for the regression coefficient Group (Output 64.14.5), as in the section "Bayesian Analysis" on page 4531.

Output 64.14.5 Coefficient Prior

Uniform Prior for Regression Coefficients

Parameter	Prior
Group	Constant

Summary statistics for all model parameters are shown in Output 64.14.6 and Output 64.14.7.

Output 64.14.6 Summary Statistics

The PHREG Procedure

Bayesian Analysis

Posterior Summaries

Parameter	N	Mean	Standard Deviation	Percentiles 25%	50%	75%
Lambda1	10000	0.000945	0.000444	0.000624	0.000876	0.00118
Lambda2	10000	0.00782	0.00363	0.00519	0.00724	0.00979
Lambda3	10000	0.0155	0.00735	0.0102	0.0144	0.0195
Lambda4	10000	0.0236	0.0116	0.0152	0.0217	0.0297
Lambda5	10000	0.3634	0.1965	0.2186	0.3266	0.4685
Lambda6	10000	0.0278	0.0153	0.0166	0.0249	0.0356
Lambda7	10000	0.0265	0.0151	0.0157	0.0236	0.0338
Lambda8	10000	0.0558	0.0323	0.0322	0.0488	0.0721
Group	10000	−0.6154	0.3570	−0.8569	−0.6186	−0.3788

Output 64.14.7 Interval Statistics

Posterior Intervals

Parameter	Alpha	Equal-Tail Interval		HPD Interval	
Lambda1	0.050	0.000289	0.00199	0.000208	0.00182
Lambda2	0.050	0.00247	0.0165	0.00194	0.0152
Lambda3	0.050	0.00484	0.0331	0.00341	0.0301
Lambda4	0.050	0.00699	0.0515	0.00478	0.0462
Lambda5	0.050	0.0906	0.8325	0.0541	0.7469
Lambda6	0.050	0.00676	0.0654	0.00409	0.0580
Lambda7	0.050	0.00614	0.0648	0.00421	0.0569
Lambda8	0.050	0.0132	0.1368	0.00637	0.1207
Group	0.050	-1.3190	0.0893	-1.3379	0.0652

The default diagnostics—namely, lag1, lag5, lag10, lag50 autocorrelations (Output 64.14.8), the Geweke diagnostics (Output 64.14.9), and the effective sample size diagnostics (Output 64.14.10)—show a good mixing of the Markov chain.

Output 64.14.8 Autocorrelations

The PHREG Procedure

Bayesian Analysis

Posterior Autocorrelations

Parameter	Lag 1	Lag 5	Lag 10	Lag 50
Lambda1	0.0705	0.0015	0.0017	-0.0076
Lambda2	0.0909	0.0206	-0.0013	-0.0039
Lambda3	0.0861	-0.0072	0.0011	0.0002
Lambda4	0.1447	-0.0023	0.0081	0.0082
Lambda5	0.1086	0.0072	-0.0038	-0.0028
Lambda6	0.1281	0.0049	-0.0036	0.0048
Lambda7	0.1925	-0.0011	0.0094	-0.0011
Lambda8	0.2128	0.0322	-0.0042	-0.0045
Group	0.5638	0.0410	-0.0003	-0.0071

Output 64.14.9 Geweke Diagnostics

Geweke Diagnostics

Parameter	z	Pr > |z|
Lambda1	-0.0705	0.9438
Lambda2	-0.4936	0.6216
Lambda3	0.5751	0.5652
Lambda4	1.0514	0.2931
Lambda5	0.8910	0.3729
Lambda6	0.2976	0.7660
Lambda7	1.6543	0.0981
Lambda8	0.6686	0.5038
Group	-1.2621	0.2069

Output 64.14.10 Effective Sample Size

Effective Sample Sizes

Parameter	ESS	Correlation Time	Efficiency
Lambda1	7775.3	1.2861	0.7775
Lambda2	6874.8	1.4546	0.6875
Lambda3	7655.7	1.3062	0.7656
Lambda4	6337.1	1.5780	0.6337
Lambda5	6563.3	1.5236	0.6563
Lambda6	6720.8	1.4879	0.6721
Lambda7	5968.7	1.6754	0.5969
Lambda8	5137.2	1.9466	0.5137
Group	2980.4	3.3553	0.2980

References

Andersen, P. K., Borgan, O., Gill, R. D., and Keiding, N. (1992), *Statistical Models Based on Counting Processes*, New York: Springer-Verlag.

Andersen, P. K. and Gill, R. D. (1982), "Cox's Regression Model Counting Process: A Large Sample Study," *Annals of Statistics*, 10, 1100–1120.

Binder, D. A. (1992), "Fitting Cox's Proportional Hazards Models from Survey Data," *Biometrika*, 79, 139–147.

Breslow, N. E. (1972), "Discussion of Professor Cox's Paper," *J. Royal Stat. Soc. B*, 34, 216–217.

Breslow, N. E. (1974), "Covariance Analysis of Censored Survival Data," *Biometrics*, 30, 89–99.

Bryson, M. C. and Johnson, M. E. (1981), "The Incidence of Monotone Likelihood in the Cox Model," *Technometrics*, 23(4), 381–383.

Cain, K. C. and Lange, N. T. (1984), "Approximate Case Influence for the Proportional Hazards Regression Model with Censored Data," *Biometrics*, 40, 493–499.

Cox, D. R. (1972), "Regression Models and Life Tables," *Journal of the Royal Statistical Society, Series B*, 20, 187–220, with discussion.

Cox, D. R. (1975), "Partial Likelihood," *Biometrika*, 62, 269–276.

Crowley, J. and Hu, M. (1977), "Covariance Analysis of Heart Transplant Survival Data," *Journal of the American Statistical Association*, 72, 27–36.

DeLong, D. M., Guirguis, G. H., and So, Y. C. (1994), "Efficient Computation of Subset Selection Probabilities with Application to Cox Regression," *Biometrika*, 81, 607–611.

Efron, B. (1977), "The Efficiency of Cox's Likelihood Function for Censored Data," *Journal of the American Statistical Association*, 72, 557–565.

Firth, D. (1993), "Bias Reduction of Maximum Likelihood Estimates," *Biometrika*, 80, 27–38.

Fleming, T. R. and Harrington, D. (1991), *Counting Processes and Survival Analysis*, New York: John Wiley & Sons.

Furnival, G. M. and Wilson, R. W. (1974), "Regression by Leaps and Bounds," *Technometrics*, 16, 499–511.

Gail, M. H., Lubin, J. H., and Rubinstein, L. V. (1981), "Likelihood Calculations for Matched Case-Control Studies and Survival Studies with Tied Death Times," *Biometrika*, 68, 703–707.

Gilks, W. R., Best, N. G., and Tan, K. K. C. (1995), "Adaptive Rejection Metropolis Sampling with Gibbs Sampling," *Applied Statistics*, 44, 455–472.

Gilks, W. R. and Wild, P. (1992), "Adaptive Rejection Sampling for Gibbs Sampling," *Applied Statistics*, 41, 337–348.

Grambsch, P. M. and Therneau, T. M. (1994), "Proportional Hazards Tests and Diagnostics Based on Weighted Residuals," *Biometrika*, 81, 515–526.

Heinze, G. (1999), *The Application of Firth's Procedure to Cox and Logistic Regression*, Technical Report 10/1999, update in January 2001, Section of Clinical Biometrics, Department of Medical Computer Sciences, University of Vienna.

Heinze, G. and Schemper, M. (2001), "A Solution to the Problem of Monotone Likelihood in Cox Regression," *Biometrics*, 51, 114–119.

Hosmer, D. W., Jr. and Lemeshow, S. (1989), *Applied Logistic Regression*, New York: John Wiley & Sons.

Ibrahim, J. G., Chen, M. H., and Sinha, D. (2001), *Bayesian Survival Analysis*, New York: Springer-Verlag.

Kalbfleisch, J. D. and Prentice, R. L. (1980), *The Statistical Analysis of Failure Time Data*, New York: John Wiley & Sons.

Kass, R. E., Carlin, B. P., Gelman, A., and Neal, R. (1998), "Markov Chain Monte Carlo in Practice: A Roundtable Discussion," *The American Statistician*, 52, 93–100.

Krall, J. M., Uthoff, V. A., and Harley, J. B. (1975), "A Step-up Procedure for Selecting Variables Associated with Survival," *Biometrics*, 31, 49–57.

Lawless, J. F. (2003), *Statistical Model and Methods for Lifetime Data*, Second Edition, New York: John Wiley & Sons.

Lawless, J. F. and Nadeau, C. (1995), "Some Simple Robust Methods for the Analysis of Recurrent Events," *Technometrics*, 37, 158–168.

Lee, E. W., Wei, L. J., and Amato, D. (1992), "Cox-Type Regression Analysis for Large Numbers of Small Groups of Correlated Failure Time Observations," 237–247, Netherlands: Kluwer Academic.

Lin, D., Wei, L. J., Yang, I., and Ying, Z. (2000), "Semiparametric Regression for the Mean and Rate Functions of Recurrent Events," *Journal of the Royal Statistical Society, Series B*, 62, 711–730.

Lin, D., Wei, L. J., and Ying, Z. (1993), "Checking the Cox Model with Cumulative Sums of Martingale-Based Residuals," *Biometrika*, 80, 557–572.

Lin, D. Y. (1994), "Cox Regression Analysis of Multivariate Failure Time Data: The Marginal Approach," *Statistics in Medicine*, 13, 2233–2247.

Lin, D. Y. and Wei, L. J. (1989), "The Robust Inference for the Proportional Hazards Model," *Journal of the American Statistical Association*, 84, 1074–1078.

Nelson, W. (2002), *Recurrent Events Data Analysis for Product Repairs, Disease Recurrences, and Other Applications*, ASA-SIAM Series on Statistics and Applied Probability.

Pepe, M. S. and Cai, J. (1993), "Some Graphical Displays and Marginal Regression Analyses for Recurrent Failure Times and Time Dependent Covariates," *Journal of the American Statistical Association*, 88, 881–820.

Pettitt, A. N. and Bin Daud, I. (1989), "Case-Weighted Measures of Influence for Proportional Hazards Regression," *Applied Statistics*, 38, 313–329.

Prentice, R. L., Williams, B. J., and Peterson, A. V. (1981), "On the Regression Analysis of Multivariate Failure Time Data," *Biometrika*, 68, 373–379.

Reid, N. and Crèpeau, H. (1985), "Influence Functions for Proportional Hazards Regression," *Biometrika*, 72, 1–9.

Sinha, D., Ibrahim, J. G., and Chen, M. H. (2003), "A Bayesian Justification of Cox's Partial Likelihood," *Biometrika*, 90, 629–641.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van der Linde, A. (2002), "Bayesian Measures of Model Complexity and Fit," *Journal of the Royal Statistical Society, Series B*, 64(4), 583–616, with discussion.

Spiekerman, C. F. and Lin, D. Y. (1998), "Marginal Regression Models for Multivariate Failure Time Data," *Journal of American Statistical Association*, 93, 1164–1175.

Therneau, T. M. (1994), *A Package for Survival Analysis in S*, Technical Report 53, Section of Biostatistics, Mayo Clinic, Rochester, MN.

Therneau, T. M. and Grambsch, P. M. (2000), *Modeling Survival Data: Extending the Cox Model*, New York: Springer-Verlag.

Tsiatis, A. (1981), "A Large Sample Study of the Estimates for the Integrated Hazard Function in Cox's Regression Model for Survival Data," *Annals of Statistics*, 9, 93–108.

Venzon, D. J. and Moolgavkar, S. H. (1988), "A Method for Computing Profile-Likelihood Based Confidence Intervals," *Applied Statistics*, 37, 87–94.

Wei, L. J., Lin, D. Y., and Weissfeld, L. (1989), "Regression Analysis of Multivariate Incomplete Failure Time Data by Modeling Marginal Distribution," *Journal of the American Statistical Association*, 84, 1065–1073.

Chapter 65

The PLAN Procedure

Contents

Overview: PLAN Procedure **4731**
Getting Started: PLAN Procedure **4733**
Three Replications with Four Factors 4733
Randomly Assigning Subjects to Treatments 4734
Syntax: PLAN Procedure **4736**
PROC PLAN Statement 4736
FACTORS Statement 4737
OUTPUT Statement 4739
TREATMENTS Statement 4742
Details: PLAN Procedure **4743**
Using PROC PLAN Interactively 4743
Output Data Sets 4743
Specifying Factor Structures 4745
Randomizing Designs 4748
Displayed Output 4748
ODS Table Names 4749
Examples: PLAN Procedure **4749**
Example 65.1: A Split-Plot Design 4749
Example 65.2: A Hierarchical Design 4750
Example 65.3: An Incomplete Block Design 4752
Example 65.4: A Latin Square Design 4753
Example 65.5: A Generalized Cyclic Incomplete Block Design 4754
Example 65.6: Permutations and Combinations 4755
Example 65.7: Crossover Designs 4759
References **4763**

Overview: PLAN Procedure

The PLAN procedure constructs designs and randomizes plans for factorial experiments, especially nested and crossed experiments and randomized block designs. PROC PLAN can also be used for generating lists of permutations and combinations of numbers. The PLAN procedure can construct the following types of experimental designs:

- full factorial designs, with and without randomization
- certain balanced and partially balanced incomplete block designs
- generalized cyclic incomplete block designs
- Latin square designs

For other kinds of experimental designs, especially fractional factorial, response surface, and orthogonal array designs, see the FACTEX and OPTEX procedures and the ADX Interface in SAS/QC software.

PROC PLAN generates designs by first generating a selection of the levels for the first factor. Then, for the second factor, PROC PLAN generates a selection of its levels for each level of the first factor. In general, for a given factor, the PLAN procedure generates a selection of its levels for all combinations of levels for the factors that precede it.

The selection can be done in five different ways:

- randomized selection, for which the levels are returned in a random order
- ordered selection, for which the levels are returned in a standard order every time a selection is generated
- cyclic selection, for which the levels returned are computed by cyclically permuting the levels of the previous selection
- permuted selection, for which the levels are a permutation of the integers $1, \ldots, n$
- combination selection, for which the m levels are selected as a combination of the integers $1, \ldots, n$ taken m at a time

The randomized selection method can be used to generate randomized plans. Also, by appropriate use of cyclic selection, any of the designs in the very wide class of generalized cyclic block designs (Jarrett and Hall 1978) can be generated.

There is no limit to the depth to which the different factors can be nested, and any number of randomized plans can be generated.

You can also declare a list of factors to be selected simultaneously with the lowest (that is, the most nested) factor. The levels of the factors in this list can be seen as constituting the treatment to be applied to the cells of the design. For this reason, factors in this list are called *treatments*. With this list, you can generate and randomize plans in one run of PROC PLAN.

Getting Started: PLAN Procedure

Three Replications with Four Factors

Suppose you want to determine if the order in which four drugs are given affects the response of a subject. If you have only three subjects to test, you can use the following statements to design the experiment.

```
proc plan seed=27371;
   factors Replicate=3 ordered Drug=4;
run;
```

These statements produce a design with three replicates of the four levels of the factor Drug arranged in random order. The three levels of Replicate are arranged in order, as shown in Figure 65.1.

Figure 65.1 Three Replications and Four Factors

```
                         The PLAN Procedure

              Factor         Select      Levels     Order

              Replicate           3           3    Ordered
              Drug                4           4    Random

                         Replicate      --Drug-

                                 1      3 2 4 1
                                 2      1 2 4 3
                                 3      4 1 2 3
```

You might also want to apply one of four different treatments to each cell of this plan (for example, applying different amounts of each drug). The following additional statements create the output shown in Figure 65.2:

```
   factors Replicate=3 ordered Drug=4;
   treatments Treatment=4;
run;
```

Figure 65.2 Using the TREATMENTS Statement

```
                         The PLAN Procedure

                            Plot Factors

              Factor         Select      Levels     Order

              Replicate           3           3    Ordered
              Drug                4           4    Random
```

Figure 65.2 *continued*

```
                         Treatment Factors

              Factor         Select      Levels     Order

              Treatment           4           4    Random

               Replicate      --Drug-      --Treatment--

                       1      3 1 2 4      2   1   3   4
                       2      4 3 2 1      4   1   2   3
                       3      3 2 4 1      1   4   2   3
```

Randomly Assigning Subjects to Treatments

You can use the PLAN procedure to design a completely randomized design. Suppose you have 12 experimental units, and you want to assign one of two treatments to each unit. Use a DATA step to store the unrandomized design in a SAS data set, and then call PROC PLAN to randomize it by specifying one factor with the default type of RANDOM, having 12 levels. The following statements produce Figure 65.3 and Figure 65.4:

```
title 'Completely Randomized Design';
/* The unrandomized design */

data Unrandomized;
   do Unit=1 to 12;
      if (Unit <= 6) then Treatment=1;
      else                Treatment=2;
      output;
   end;
run;

/* Randomize the design */

proc plan seed=27371;
   factors Unit=12;
   output data=Unrandomized out=Randomized;
run;

proc sort data=Randomized;
   by Unit;
proc print;
run;
```

Figure 65.3 shows that the 12 levels of the unit factor have been randomly reordered and then lists the new ordering.

Figure 65.3 A Completely Randomized Design for Two Treatments

```
                    Completely Randomized Design

                         The PLAN Procedure

            Factor      Select      Levels     Order

            Unit            12          12    Random

               ----------------Unit---------------

                8  5  1  4  6  2 12  7  3  9 10 11
```

After the data set is sorted by the unit variable, the randomized design is displayed (Figure 65.4).

Figure 65.4 A Completely Randomized Design for Two Treatments

```
                    Completely Randomized Design

                      Obs    Unit    Treatment

                        1      1         1
                        2      2         1
                        3      3         2
                        4      4         1
                        5      5         1
                        6      6         1
                        7      7         2
                        8      8         1
                        9      9         2
                       10     10         2
                       11     11         2
                       12     12         2
```

You can also generate the plan by using a TREATMENTS statement instead of a DATA step. The following statements generate the same plan.

```
proc plan seed=27371;
   factors Unit=12;
   treatments Treatment=12 cyclic (1 1 1 1 1 1 2 2 2 2 2 2);
   output out=Randomized;
run;
```

Syntax: PLAN Procedure

The following statements are available in PROC PLAN.

PROC PLAN < *options* > **;**
FACTORS *factor-selections* < / **NOPRINT** > **;**
OUTPUT **OUT=***SAS-data-set* < *factor-value-settings* > **;**
TREATMENTS *factor-selections* **;**

To use PROC PLAN, you need to specify the PROC PLAN statement and at least one FACTORS statement before the first RUN statement. The TREATMENTS statement, OUTPUT statement, and additional FACTORS statements can appear either before the first RUN statement or after it.

The rest of this section gives detailed syntax information for each of the statements, beginning with the PROC PLAN statement. The remaining statements are described in alphabetical order.

You can use PROC PLAN interactively by specifying multiple groups of statements, separated by RUN statements. For details, see the section "Using PROC PLAN Interactively" on page 4743.

PROC PLAN Statement

PROC PLAN < *options* > **;**

The PROC PLAN statement starts the PLAN procedure and, optionally, specifies a random number seed or a default method for selecting levels of factors. By default, the procedure uses a random number seed generated from reading the time of day from the computer's clock and randomly selects levels of factors. These defaults can be modified with the SEED= and ORDERED options, respectively. Unlike many SAS/STAT procedures, the PLAN procedure does not have a DATA= option in the PROC statement; in this procedure, both the input and output data sets are specified in the OUTPUT statement.

You can specify the following options in the PROC PLAN statement:

SEED=*number*
specifies an integer used to start the pseudo-random number generator for selecting factor levels randomly. If you do not specify a seed, or if you specify a value less than or equal to zero, the seed is by default generated from reading the time of day from the computer's clock.

ORDERED
selects the levels of the factor as the integers $1, 2, \ldots, m$, in order. For more detail, see "*Selection-Types*" on page 4737 and "Specifying Factor Structures" on page 4745.

FACTORS Statement

FACTORS *factor-selections* < / **NOPRINT** > **;**

The FACTORS statement specifies the factors of the plan and generates the plan. Taken together, the *factor-selections* specify the plan to be generated; more than one *factor-selection* request can be used in a FACTORS statement. The form of a *factor-selection* is

name **=** *m* < **OF** *n* > < *selection-type* > **;**

where

name is a valid SAS name. This gives the name of a factor in the design.

m is a positive integer that gives the number of values to be selected. If *n* is specified, the value of *m* must be less than or equal to *n*.

n is a positive integer that gives the number of values to be selected from.

selection-type specifies one of five methods for selecting *m* values. Possible values are COMB, CYCLIC, ORDERED, PERM, and RANDOM. The CYCLIC *selection-type* has additional optional specifications that enable you to specify an initial block of numbers to be cyclically permuted and an increment used to permute the numbers. By default, the *selection-type* is RANDOM, unless you use the ORDERED option in the PROC PLAN statement. In this case, the default *selection-type* is ORDERED. For details, see the following section, "*Selection-Types*"; for examples, see the section "Syntax Examples" on page 4738.

The following option can appear in the FACTORS statement after the slash:

NOPRINT
suppresses the display of the plan. This is particularly useful when you require only an output data set. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

Selection-Types

PROC PLAN interprets *selection-type* as follows:

RANDOM selects the *m* levels of the factor randomly without replacement from the integers $1, 2, \ldots, n$. Or, if *n* is not specified, RANDOM selects levels by randomly ordering the integers $1, 2, \ldots, m$.

ORDERED selects the levels of the factor as the integers $1, 2, \ldots, m$, in that order.

PERM selects the *m* levels of the factor as a permutation of the integers $1, \ldots m$ according to an algorithm that cycles through all $m!$ permutations. The permutations are produced in a sorted standard order; see Example 65.6.

COMB selects the *m* levels of the factor as a combination of the integers $1, \ldots, n$ taken *m* at a time, according to an algorithm that cycles through all $n!/(m!(n-m)!)$ combinations. The combinations are produced in a sorted standard order; see Example 65.6.

CYCLIC <**(***initial-block***)** > < *increment* > selects the levels of the factor by cyclically permuting the integers $1, 2, \ldots, n$. Wrapping occurs at m if n is not specified, and at n if n is specified. Additional optional specifications are as follows.

With the *selection-type* CYCLIC, you can optionally specify an *initial-block* and an *increment*. The *initial-block* must be specified within parentheses, and it specifies the block of numbers to permute. The first permutation is the block you specify, the second is the block permuted by 1 (or by the *increment* you specify), and so on. By default, the *initial-block* is the integers $1, 2, \ldots, m$. If you specify an *initial-block*, it must have m values. Values specified in the *initial-block* do not have to be given in increasing order.

The *increment* specifies the increment by which to permute the block of numbers. By default, the *increment* is 1.

Syntax Examples

This section gives some simple syntax examples. For more complex examples and details on how to generate various designs, see "Specifying Factor Structures" on page 4745. The examples in this section assume that you use the default random selection method and do not use the ORDERED option in the PROC PLAN statement.

The following specification generates a random permutation of the numbers 1, 2, 3, 4, and 5.

```
factors A=5;
```

The following specification generates a random permutation of five of the integers from 1 to 8, selected without replacement.

```
factors A=5 of 8;
```

Adding the ORDERED *selection-type* to the two previous specifications generates an ordered list of the integers 1 to 5. The following specification cyclically permutes the integers 1, 2, 3, and 4.

```
factors A=4 cyclic;
```

Since this simple request generates only one permutation of the numbers, the procedure generates an ordered list of the integers 1 to 4. The following specification cyclically permutes the integers 5 to 8.

```
factors A=4 of 8 cyclic (5 6 7 8);
```

In this case, since only one permutation is performed, the procedure generates an ordered list of the integers 5 to 8. The following specification produces an ordered list for A, with values 1 and 2.

```
factors A=2 ordered B=4 of 8 cyclic (5 6 7 8) 2;
```

The associated factor levels for B are 5, 6, 7, 8 for level 1 of A, and 7, 8, 1, 2 for level 2 of A.

Handling More Than One *Factor-Selection*

For cases with more than one *factor-selection* in the same FACTORS statement, PROC PLAN constructs the design as follows:

1. PROC PLAN first generates levels for the first *factor-selection*. These levels are permutations of integers (1, 2, and so on) appropriate for the selection type chosen. If you do not specify a selection type, PROC PLAN uses the default (RANDOM); if you specify the ORDERED option in the PROC PLAN statement, the procedure uses ORDERED as the default selection type.
2. For every integer generated for the first *factor-selection*, levels are generated for the second *factor-selection*. These levels are generated according to the specifications following the second equal sign.
3. This process is repeated until levels for all *factor-selections* have been generated.

The following statements give an example of generating a design with two random factors:

```
proc plan;
   factors One=4 Two=3;
run;
```

The procedure first generates a random permutation of the integers 1 to 4 and then, for each of these, generates a random permutation of the integers 1 to 3. You can think of factor Two as being nested within factor One, where the levels of factor One are to be randomly assigned to 4 units.

As another example, six random permutations of the numbers 1, 2, 3 can be generated by specifying the following statements:

```
proc plan;
   factors a=6 ordered b=3;
run;
```

OUTPUT Statement

OUTPUT OUT=*SAS-data-set* < **DATA=***SAS-data-set* > < *factor-value-settings* > **;**

The OUTPUT statement applies only to the last plan generated. If you use PROC PLAN interactively, the OUTPUT statement for a given plan must be immediately preceded by the FACTORS statement (and the TREATMENTS statement, if appropriate) for the plan.

See "Output Data Sets" on page 4743 for more information about how output data sets are constructed.

You can specify the following options in the OUTPUT statement:

OUT=*SAS-data-set*

DATA=*SAS-data-set*

You can use the OUTPUT statement both to output the last plan generated and to use the last plan generated to randomize another SAS data set.

When you specify only the OUT= option in the OUTPUT statement, PROC PLAN saves the last plan generated to the specified data set. The output data set contains one variable for each factor in the plan and one observation for each cell in the plan. The value of a variable in a given observation is the level of the corresponding factor for that cell. The OUT= option is required.

When you specify both the DATA= and OUT= options in the OUTPUT statement, then PROC PLAN uses the last plan generated to randomize the input data set (DATA=), saving the results to the output data set (OUT=). The output data set has the same form as the input data set but has modified values for the variables that correspond to factors (see the section "Output Data Sets" on page 4743 for details). Values for variables not corresponding to factors are transferred without change.

factor-value-settings

specify the values input or output for the factors in the design. The form for *factor-value-settings* is different when only an OUT= data set is specified and when both OUT= and DATA= data sets are specified.

Both forms are discussed in the following section.

Factor-Value-Settings with Only an OUT= Data Set

When you specify only an OUT= data set, the form for each *factor-value-setting* specification is one of the following:

factor-name < **NVALS=***list-of-n-numbers* > < **ORDERED** | **RANDOM** > **;**

or

factor-name < **CVALS=***list-of-n-strings* > < **ORDERED** | **RANDOM** > **;**

where

factor-name is a factor in the last FACTORS statement preceding the OUTPUT statement.

NVALS= lists n numeric values for the factor. By default, the procedure uses NVALS=(1 2 3 $\cdots$ n).

CVALS= lists n character strings for the factor. Each string can have up to 40 characters, and each string must be enclosed in quotes. **WARNING:** When you use the CVALS= option, the variable created in the output data set has a length equal to the length of the longest string given as a value; shorter strings are padded with trailing blanks. For example, the values output for the first level of a two-level factor with the following two different specifications are not the same.

```
CVALS=('String 1' "String 2")

CVALS=('String 1' "A longer string")
```

The value output with the second specification is 'String 1' followed by seven blanks. In order to match two such values (for example, when merging two plans), you must use the TRIM function in the DATA step (see *SAS Language Reference: Dictionary*).

ORDERED | RANDOM specifies how values (those given with the NVALS= or CVALS= option, or the default values) are associated with the levels of a factor (the integers $1, 2, \ldots, n$). The default association type is ORDERED, for which the first value specified is output for a factor level setting of 1, the second value specified is output for a level of 2, and so on. You can also specify an association type of RANDOM, for which the levels are associated with the values in a random order. Specifying RANDOM is useful for randomizing crossed experiments (see the section "Randomizing Designs" on page 4748).

The following statements give an example of using the OUTPUT statement with only an OUT= data set and with both the NVALS= and CVALS= specifications.

```
proc plan;
   factors a=6 ordered b=3;
   output out=design a nvals=(10 to 60 by 10)
                     b cvals=('HSX' 'SB2' 'DNY');
run;
```

The DESIGN data set contains two variables, a and b. The values of the variable a are 10 when factor a equals 1, 20 when factor a equals 2, and so on. Values of the variable b are 'HSX' when factor b equals 1, 'SB2' when factor b equals 2, and 'DNY' when factor b equals 3.

Factor-Value-Settings with OUT= and DATA= Data Sets

If you specify an input data set with DATA=, then PROC PLAN assumes that each factor in the last plan generated corresponds to a variable in the input set. If the variable name is different from the name of the factor to which it corresponds, the two can be associated in the values specification by

input-variable-name = *factor-name* ;

Then, the NVALS= or CVALS= specification can be used. The values given by NVALS= or CVALS= specify the input values as well as the output values for the corresponding variable.

Since the procedure assumes that the collection of input factor values constitutes a plan position description (see the section "Output Data Sets" on page 4743), the values must correspond to integers less than or equal to m, the number of values selected for the associated factor. If any input values do not correspond, then the collection does not define a plan position, and the corresponding observation is output without changing the values of any of the factor variables.

The following statements demonstrate the use of *factor-value-settings*. The input SAS data set a contains variables Block and Plot, which are renamed Day and Hour, respectively.

```
proc plan;
   factors Day=7 Hour=6;
   output data=a out=b
      Block = Day  cvals=('Mon' 'Tue' 'Wed' 'Thu'
                          'Fri' 'Sat' 'Sun'      )
      Plot  = Hour;
run;
```

For another example of using both a DATA= and OUT= data set, see the section "Randomly Assigning Subjects to Treatments" on page 4734.

TREATMENTS Statement

TREATMENTS *factor-selections* **;**

The TREATMENTS statement specifies the *treatments* of the plan to generate, but it does not generate a plan. If you supply several FACTORS and TREATMENTS statements before the first RUN statement, the procedure uses only the last TREATMENTS specification and applies it to the plans generated by each of the FACTORS statements. The TREATMENTS statement has the same form as the FACTORS statement. The individual *factor-selections* also have the same form as in the FACTORS statement:

name **=** *m* < **OF** *n* > < *selection-type* > **;**

The procedure generates each *treatment* simultaneously with the lowest (that is, the most nested) factor in the last FACTORS statement. The *m* value for each *treatment* must be at least as large as the *m* for the most nested factor.

The following statements give an example of using both a FACTORS and a TREATMENTS statement. First the FACTORS statement sets up the rows and columns of a 3×3 square (factors r and c). Then, the TREATMENTS statement augments the square with two cyclic treatments. The resulting design is a 3×3 Graeco-Latin square, a type of design useful in main-effects factorial experiments.

```
proc plan;
   factors r=3 ordered c=3 ordered;
   treatments a=3 cyclic
              b=3 cyclic 2;
run;
```

The resulting Graeco-Latin square design is shown in Figure 65.5. Notice how the values of r and c are ordered (1, 2, 3) as requested.

Figure 65.5 A 3×3 Graeco-Latin Square

```
                    The PLAN Procedure

          r      --c--      --a--      --b--

          1      1 2 3      1 2 3      1 2 3
          2      1 2 3      2 3 1      3 1 2
          3      1 2 3      3 1 2      2 3 1
```

Details: PLAN Procedure

Using PROC PLAN Interactively

After specifying a design with a FACTORS statement and running PROC PLAN with a RUN statement, you can generate additional plans and output data sets without invoking PROC PLAN again.

In PROC PLAN, all statements can be used interactively. You can execute statements singly or in groups by following the single statement or group of statements with a RUN statement.

If you use PROC PLAN interactively, you can end the procedure with a DATA step, another PROC step, an ENDSAS statement, or a QUIT statement. The syntax of the QUIT statement is

```
quit;
```

When you use PROC PLAN interactively, additional RUN statements do not end the procedure but tell PROC PLAN to execute additional statements.

Output Data Sets

To understand how PROC PLAN creates output data sets, you need to look at how the procedure represents a plan. A plan is a list of values for all the factors, the values being chosen according to the *factor-selection* requests you specify. For example, consider the plan produced by the following statements:

```
proc plan seed=12345;
   factors a=3 b=2;
run;
```

The plan as displayed by PROC PLAN is shown in Figure 65.6.

Figure 65.6 A Simple Plan

```
                    The PLAN Procedure

Factor      Select      Levels     Order

a                3           3    Random
b                2           2    Random

                   a      -b-

                   2      2 1
                   1      1 2
                   3      2 1
```

The first cell of the plan has a=2 and b=2, the second has a=2 and b=1, the third has a=1 and b=1, and so on. If you output the plan to a data set with the OUTPUT statement, by default the output data set contains a numeric variable with that factor's name; the values of this numeric variable are the numbers of the successive levels selected for the factor in the plan. For example, the following statements produce Figure 65.7.

```
proc plan seed=12345;
   factors a=3 b=2;
   output out=out;
proc print data=out;
run;
```

Figure 65.7 Output Data Set from Simple Plan

```
Obs    a    b

 1     2    2
 2     2    1
 3     1    1
 4     1    2
 5     3    2
 6     3    1
```

Alternatively, you can specify the values that are output for a factor with the CVALS= or NVALS= option. Also, you can specify that the internal values be associated with the output values in random order with the RANDOM option. See the section "OUTPUT Statement" on page 4739.

If you also specify an input data set (DATA=), each factor is associated with a variable in the DATA= data set. This occurs either implicitly by the factor and variable having the same name or explicitly as described in the specifications for the OUTPUT statement. In this case, the values of the variables corresponding to the factors are first read and then interpreted as describing the position of a cell in the plan. Then the respective values taken by the factors at that position are assigned to the variables in the OUT= data set. For example, consider the data set defined by the following statements.

```
data in;
   input a b;
   datalines;
1 1
2 1
3 1
;
```

Suppose you specify this data set as an input data set for the OUTPUT statement.

```
proc plan seed=12345;
   factors a=3 b=2;
   output out=out data=in;
proc print data=out;
run;
```

PROC PLAN interprets the first observation as referring to the cell in the first row and column of the plan, since a=1 and b=1; likewise, the second observation is interpreted as the cell in the second row and first column, and the third observation as the cell in the third row and first column. In the output data set, a and b have the values they have in the plan at these positions, as shown in Figure 65.8.

Figure 65.8 Output Form of Input Data Set from Simple Plan

Obs	a	b
1	2	2
2	1	1
3	3	2

When the factors are random, this has the effect of randomizing the input data set in the same manner as the plan produced (see the sections "Randomizing Designs" on page 4748 and "Randomly Assigning Subjects to Treatments" on page 4734).

Specifying Factor Structures

By appropriately combining features of the PLAN procedure, you can construct an extensive set of designs. The basic tools are the *factor-selections*, which are used in the FACTORS and TREATMENTS statements. Table 65.1 summarizes how the procedure interprets various *factor-selections* (assuming that the ORDERED option is not specified in the PROC PLAN statement).

Table 65.1 *Factor-Selection* Interpretation

Form of Request	Interpretation	Example	Results
name=*m*	produce a random permutation of the integers $1, 2, \ldots, m$	`t=15`	lists a random ordering of the numbers $1, 2, \ldots, 15$
name=*m* cyclic	cyclically permute the integers $1, 2, \ldots, m$	`t=5 cyclic`	selects the integers 1 to 5. On the next iteration, selects 2,3,4,5,1; then 3,4,5,1,2; and so on.
name=*m* of *n*	choose a random sample of *m* integers (without replacement) from the set of integers $1, 2, \ldots, n$	`t=5 of 15`	lists a random selection of 5 numbers from 1 to 15. First, the procedure selects 5 numbers and then arranges them in random order.
name=*m* of *n* ordered	has the same effect as *name*=*m* ordered	`t=5 of 15 ordered`	lists the integers 1 to 5 in increasing order (same as t=5 ordered)
name=*m* of *n* cyclic	permute *m* of the *n* integers	`t=5 of 30 cyclic`	selects the integers 1 to 5. On the next iteration, selects 2,3,4,5,6; then 3,4,5,6,7; and so on. The 30th iteration produces 30,1,2,3,4; the 31st iteration produces 1,2,3,4,5; and so on.
name=*m* perm	produce a list of all permutations of *m* integers	`t=5 perm`	lists the integers 1,2,3,4,5 on the first iteration; on the second lists 1,2,3,5,4; on the 119th iteration lists 5,4,3,1,2; and on the last (120th) lists 5,4,3,2,1.
name=*m* of *n* comb	choose combinations of *m* integers from *n* integers	`t=3 of 5 comb`	lists all combinations of 5 choose 3 integers. The first iteration is 1,2,3; the second is 1,2,4; the third is 1,2,5; and so on until the last iteration 3,4,5.

Table 65.1 *continued*

Form of Request	Interpretation	Example	Results
name=*m* of *n* cyclic (*initial-block*)	permute *m* of the *n* integers, starting with the values specified in the *initial-block*	`t=4 of 30 cyclic (2 10 15 18)`	selects the integers 2,10,15,18. On the next iteration, selects 3,11,16,19; then 4,12,17,20; and so on. The thirteenth iteration is 14,22,27,30; the fourteenth iteration is 15,23,28,1; and so on.
name=*m* of *n* cyclic (*initial-block*) *increment*	permute *m* of the *n* integers. Start with the values specified in the *initial-block*, then add the *increment* to each value.	`t=4 of 30 cyclic (2 10 15 18) 2`	selects the integers 2,10,15,18. On the next iteration, selects 4,12,17,20; then 6,14,19,22; and so on. The wrap occurs at the eighth iteration. The eighth iteration is 16,24,29,2; and so on.

In Table 65.1, in order for more than one iteration to appear in the plan, another *name*=*j* factor selection (with $j > 1$) must precede the example factor selection. For example, the following statements produce six of the iterations described in the last entry of Table 65.1.

```
proc plan;
   factors c=6 ordered t=4 of 30 cyclic (2 10 15 18) 2;
run;
```

The following statements create a randomized complete block design and output the design to a data set.

```
proc plan ordered;
   factors blocks=3 cell=5;
   treatments t=5 random;
   output out=rcdb;
run;
```

Randomizing Designs

In many situations, proper randomization is crucial for the validity of any conclusions to be drawn from an experiment. Randomization is used both to neutralize the effect of any unknown systematic biases that might be involved in the design and to provide a basis for the assumptions underlying the analysis.

You can use PROC PLAN to randomize an already existing design: one produced by a previous call to PROC PLAN, perhaps, or a more specialized design taken from a standard reference such as Cochran and Cox (1957). The method is to specify the appropriate block structure in the FACTORS statement and then to specify the data set where the design is stored with the DATA= option in the OUTPUT statement. For an illustration of this method, see the section "Randomly Assigning Subjects to Treatments" on page 4734).

Two sorts of randomization are provided for, corresponding to the RANDOM factor selection and association types in the FACTORS and OUTPUT statements, respectively. Designs in which factors are completely nested (for example, block designs) should be randomized by specifying that the selection type of each factor is RANDOM in the FACTORS statement, which is the default (see Example 65.3). On the other hand, if the factors are crossed (for example, row-and-column designs), they should be randomized by one random reassignment of their values for the whole design. To do this, specify that the association type of each factor is RANDOM in the OUTPUT statement (see Example 65.4).

Displayed Output

The PLAN procedure displays the following output:

- the *m* value for each factor, which is the number of values to be selected
- the *n* value for each factor, which is the number of values to be selected from
- the selection type for each factor, as specified in the FACTORS statement
- the initial block and increment number for cyclic factors
- the *factor-value-selections* making up each plan

In addition, notes are written to the log that give the starting and ending values of the random number seed for each call to PROC PLAN.

ODS Table Names

PROC PLAN assigns a name to each table it creates. You can use these names to reference the table in the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 65.2 ODS Tables Produced by PROC PLAN

ODS Table Name	Description	Statements
FInfo	General factor information	FACTORS and no TREATMENTS
PFInfo	Plot factor information	FACTORS and TREATMENTS
Plan	Computed plan	default
TFInfo	Treatment factor information	FACTORS and TREATMENTS

Examples: PLAN Procedure

Example 65.1: A Split-Plot Design

This plan is appropriate for a split-plot design with main plots forming a randomized complete block design. In this example, there are three blocks, four main plots per block, and two subplots per main plot. First, three random permutations (one for each of the blocks) of the integers 1, 2, 3, and 4 are produced. The four integers correspond to the four levels of the main plot factor a; the permutation determines how the levels of a are assigned to the main plots within a block. For each of these 12 numbers (four numbers per block for three blocks), a random permutation of the integers 1 and 2 is produced. Each two-integer permutation determines the assignment of the two levels of the subplot factor b within a main plot. The following statements produce Output 65.1.1:

```
title 'Split Plot Design';
proc plan seed=37277;
   factors Block=3 ordered a=4 b=2;
run;
```

Output 65.1.1 A Split-Plot Design

Split Plot Design

The PLAN Procedure

Factor	Select	Levels	Order
Block	3	3	Ordered
a	4	4	Random
b	2	2	Random

Block	a	-b-
1	4	2 1
	3	2 1
	1	2 1
	2	2 1
2	4	1 2
	3	1 2
	1	2 1
	2	1 2
3	4	2 1
	2	2 1
	3	2 1
	1	2 1

Example 65.2: A Hierarchical Design

In this example, three plants are nested within four pots, which are nested within three houses. The FACTORS statement requests a random permutation of the numbers 1, 2, and 3 to choose Houses randomly. The second step requests a random permutation of the numbers 1, 2, 3, and 4 for each of those first three numbers to randomly assign Pots to Houses. Finally, the FACTORS statement requests a random permutation of 1, 2, and 3 for each of the 12 integers in the second set of permutations. This last step randomly assigns Plants to Pots. The following statements produce Output 65.2.1:

```
title 'Hierarchical Design';
proc plan seed=17431;
   factors Houses=3 Pots=4 Plants=3 / noprint;
   output out=nested;
run;

proc print data=nested;
run;
```

Output 65.2.1 A Hierarchical Design

Obs	Houses	Pots	Plants
1	1	3	2
2	1	3	3
3	1	3	1
4	1	1	3
5	1	1	1
6	1	1	2
7	1	2	2
8	1	2	3
9	1	2	1
10	1	4	3
11	1	4	2
12	1	4	1
13	2	4	1
14	2	4	3
15	2	4	2
16	2	2	2
17	2	2	1
18	2	2	3
19	2	3	2
20	2	3	3
21	2	3	1
22	2	1	2
23	2	1	3
24	2	1	1
25	3	4	1
26	3	4	3
27	3	4	2
28	3	1	3
29	3	1	2
30	3	1	1
31	3	2	1
32	3	2	2
33	3	2	3
34	3	3	3
35	3	3	2
36	3	3	1

Example 65.3: An Incomplete Block Design

Jarrett and Hall (1978) give an example of a generalized cyclic design with good efficiency characteristics. The design consists of two replicates of 52 treatments in 13 blocks of size 8. The following statements use the PLAN procedure to generate this design in an appropriately randomized form and store it in a SAS data set GCBD. Then the design is sorted and transposed to display in randomized order. The following statements produce Output 65.3.1 and Output 65.3.2:

```
title 'Generalized Cyclic Block Design';
proc plan seed=33373;
   treatments Treatment=8 of 52 cyclic (1 2 3 4 32 43 46 49) 4;
   factors Block=13 Plot=8;
   output out=GCBD;
quit;
proc sort data=GCBD out=GCBD;
   by Block Plot;
proc transpose data= GCBD(rename=(Plot=_NAME_))
                out =tGCBD(drop=_NAME_);
   by Block;
   var Treatment;
proc print data=tGCBD noobs;
run;
```

Output 65.3.1 A Generalized Cyclic Block Design

Generalized Cyclic Block Design

The PLAN Procedure

Plot Factors

Factor	Select	Levels	Order
Block	13	13	Random
Plot	8	8	Random

Treatment Factors

Factor	Select	Levels	Order	Initial Block / Increment
Treatment	8	52	Cyclic	(1 2 3 4 32 43 46 49) / 4

Output 65.3.1 *continued*

Block	------Plot-----	-------Treatment-------
10	7 4 8 1 2 3 5 6	1 2 3 4 32 43 46 49
8	1 2 4 3 8 6 5 7	5 6 7 8 36 47 50 1
9	2 5 4 7 3 1 8 6	9 10 11 12 40 51 2 5
6	4 2 6 8 3 7 1 5	13 14 15 16 44 3 6 9
7	4 7 6 3 1 2 8 5	17 18 19 20 48 7 10 13
4	4 8 1 5 3 6 7 2	21 22 23 24 52 11 14 17
2	6 2 3 8 7 5 1 4	25 26 27 28 4 15 18 21
3	6 2 3 1 7 4 5 8	29 30 31 32 8 19 22 25
1	1 2 7 8 5 6 3 4	33 34 35 36 12 23 26 29
5	5 7 6 8 4 3 1 2	37 38 39 40 16 27 30 33
12	5 8 1 4 7 3 6 2	41 42 43 44 20 31 34 37
13	3 5 1 8 4 2 6 7	45 46 47 48 24 35 38 41
11	4 1 5 2 3 8 6 7	49 50 51 52 28 39 42 45

Output 65.3.2 A Generalized Cyclic Block Design

Generalized Cyclic Block Design

Block	_1	_2	_3	_4	_5	_6	_7	_8
1	33	34	26	29	12	23	35	36
2	18	26	27	21	15	25	4	28
3	32	30	31	19	22	29	8	25
4	23	17	52	21	24	11	14	22
5	30	33	27	16	37	39	38	40
6	6	14	44	13	9	15	3	16
7	48	7	20	17	13	19	18	10
8	5	6	8	7	50	47	1	36
9	51	9	40	11	10	5	12	2
10	4	32	43	2	46	49	1	3
11	50	52	28	49	51	42	45	39
12	43	37	31	44	41	34	20	42
13	47	35	45	24	46	38	41	48

Example 65.4: A Latin Square Design

All of the preceding examples involve designs with completely nested block structures, for which PROC PLAN was especially designed. However, by appropriate coordination of its facilities, a much wider class of designs can be accommodated. A Latin square design is based on experimental units that have a row-and-column block structure. The following example uses the CYCLIC option for a treatment factor tmts to generate a simple 4×4 Latin square. Randomizing a Latin square design involves randomly permuting the row, column, and treatment values independently. In order to do this, use the RANDOM option in the OUTPUT statement of PROC PLAN. The example also uses *factor-value-settings* in the OUTPUT statement. The following statements produce Output 65.4.1:

```
title 'Latin Square Design';
proc plan seed=37430;
   factors Row=4 ordered Col=4 ordered / noprint;
   treatments Tmt=4 cyclic;
   output out=LatinSquare
          Row cvals=('Day 1' 'Day 2' 'Day 3' 'Day 4') random
          Col cvals=('Lab 1' 'Lab 2' 'Lab 3' 'Lab 4') random
          Tmt nvals=(      0     100     250     450) random;
quit;

proc sort data=LatinSquare out=LatinSquare;
   by Row Col;
proc transpose data= LatinSquare(rename=(Col=_NAME_))
                out =tLatinSquare(drop=_NAME_);
   by Row;
   var Tmt;
proc print data=tLatinSquare noobs;
run;
```

Output 65.4.1 A Randomized Latin Square Design

Latin Square Design

Row	Lab_1	Lab_2	Lab_3	Lab_4
Day 1	0	250	100	450
Day 2	250	450	0	100
Day 3	100	0	450	250
Day 4	450	100	250	0

Example 65.5: A Generalized Cyclic Incomplete Block Design

The following statements depict how to create an appropriately randomized generalized cyclic incomplete block design for v treatments (given by the value of t) in b blocks (given by the value of b) of size k (with values of p indexing the cells within a block) with initial block $(e_1\ e_2\ \cdots\ e_k)$ and increment number i.

factors b=b **p=**k **;**
treatments t=k **of** v **cyclic (**$e_1\ e_2\ \cdots\ e_k$ **)** i **;**

For example, the specification

```
proc plan seed=37430;
   factors b=10 p=4;
   treatments t=4 of 30 cyclic (1 3 4 26) 2;
run;
```

generates the generalized cyclic incomplete block design given in Example 1 of Jarrett and Hall (1978), which is given by the rows and columns of the plan associated with the treatment factor t in Output 65.5.1.

Output 65.5.1 A Generalized Cyclic Incomplete Block Design

The PLAN Procedure

Plot Factors

Factor	Select	Levels	Order
b	10	10	Random
p	4	4	Random

Treatment Factors

Factor	Select	Levels	Order	Initial Block / Increment
t	4	30	Cyclic	(1 3 4 26) / 2

b	---p---	-----t-----
2	2 3 1 4	1 3 4 26
1	3 2 4 1	3 5 6 28
3	2 3 4 1	5 7 8 30
10	4 2 3 1	7 9 10 2
9	4 1 2 3	9 11 12 4
4	1 3 2 4	11 13 14 6
5	1 2 4 3	13 15 16 8
8	3 2 4 1	15 17 18 10
7	2 4 1 3	17 19 20 12
6	2 1 4 3	19 21 22 14

Example 65.6: Permutations and Combinations

Occasionally, you might need to generate all possible permutations of n things, or all possible combinations of n things taken m at a time.

For example, suppose you are planning an experiment in cognitive psychology where you want to present four successive stimuli to each subject. You want to observe each permutation of the four stimuli. The following statements use PROC PLAN to create a data set containing all possible permutations of four numbers in random order.

```
title 'All Permutations of 1,2,3,4';
proc plan seed=60359;
   factors    Subject  = 24
              Order    = 4  ordered;
   treatments Stimulus = 4  perm;
   output out=Psych;
run;

proc sort data=Psych out=Psych;
   by Subject Order;
```

```
proc transpose data= Psych(rename=(Order=_NAME_))
               out =tPsych(drop=_NAME_);
   by Subject;
   var Stimulus;
proc print data=tPsych noobs;
run;
```

The variable Subject is set at 24 levels because there are $4! = 24$ total permutations to be listed. If Subject > 24, the list repeats. Output 65.6.1 displays the PROC PLAN output. Note that the variable Subject is listed in random order.

Output 65.6.1 List of Permutations

All Permutations of 1,2,3,4

The PLAN Procedure

Plot Factors

Factor	Select	Levels	Order
Subject	24	24	Random
Order	4	4	Ordered

Treatment Factors

Factor	Select	Levels	Order
Stimulus	4	4	Perm

Subject	-Order-	-Stimulus-
4	1 2 3 4	1 2 3 4
15	1 2 3 4	1 2 4 3
24	1 2 3 4	1 3 2 4
1	1 2 3 4	1 3 4 2
5	1 2 3 4	1 4 2 3
17	1 2 3 4	1 4 3 2
19	1 2 3 4	2 1 3 4
14	1 2 3 4	2 1 4 3
6	1 2 3 4	2 3 1 4
23	1 2 3 4	2 3 4 1
8	1 2 3 4	2 4 1 3
2	1 2 3 4	2 4 3 1
13	1 2 3 4	3 1 2 4
16	1 2 3 4	3 1 4 2
12	1 2 3 4	3 2 1 4
18	1 2 3 4	3 2 4 1
21	1 2 3 4	3 4 1 2
9	1 2 3 4	3 4 2 1
22	1 2 3 4	4 1 2 3
10	1 2 3 4	4 1 3 2
7	1 2 3 4	4 2 1 3
11	1 2 3 4	4 2 3 1
3	1 2 3 4	4 3 1 2
20	1 2 3 4	4 3 2 1

The output data set Psych contains 96 observations of the 3 variables (Subject, Order, and Stimulus). Sorting the output data set by Subject and by Order within Subject results in all possible permutations of Stimulus in random order. PROC TABULATE displays these permutations in Output 65.6.2.

Output 65.6.2 Randomized Permutations

All Permutations of 1,2,3,4

Subject	_1	_2	_3	_4
1	1	3	4	2
2	2	4	3	1
3	4	3	1	2
4	1	2	3	4
5	1	4	2	3
6	2	3	1	4
7	4	2	1	3
8	2	4	1	3
9	3	4	2	1
10	4	1	3	2
11	4	2	3	1
12	3	2	1	4
13	3	1	2	4
14	2	1	4	3
15	1	2	4	3
16	3	1	4	2
17	1	4	3	2
18	3	2	4	1
19	2	1	3	4
20	4	3	2	1
21	3	4	1	2
22	4	1	2	3
23	2	3	4	1
24	1	3	2	4

As another example, suppose you have six alternative treatments, any four of which can occur together in a block (in no particular order). The following statements use PROC PLAN to create a data set containing all possible combinations of six numbers taken four at a time. In this case, you use ODS to create the data set.

```
title 'All Combinations of (6 Choose 4) Integers';
proc plan;
   factors Block=15 ordered
           Treat= 4 of 6 comb;
   ods output Plan=Combinations;
run;

proc print data=Combinations noobs;
run;
```

The variable Block has 15 levels since there are a total of $6!/(4!2!) = 15$ combinations of four integers chosen from six integers. The data set formed by ODS from the displayed plan has one row for each block, with the four values of Treat corresponding to four different variables, as shown in Output 65.6.3 and Output 65.6.4.

Output 65.6.3 List of Combinations

All Combinations of (6 Choose 4) Integers

The PLAN Procedure

Factor	Select	Levels	Order
Block	15	15	Ordered
Treat	4	6	Comb

Block	-Treat-
1	1 2 3 4
2	1 2 3 5
3	1 2 3 6
4	1 2 4 5
5	1 2 4 6
6	1 2 5 6
7	1 3 4 5
8	1 3 4 6
9	1 3 5 6
10	1 4 5 6
11	2 3 4 5
12	2 3 4 6
13	2 3 5 6
14	2 4 5 6
15	3 4 5 6

Output 65.6.4 Combinations Data Set Created by ODS

All Combinations of (6 Choose 4) Integers

Block	Treat1	Treat2	Treat3	Treat4
1	1	2	3	4
2	1	2	3	5
3	1	2	3	6
4	1	2	4	5
5	1	2	4	6
6	1	2	5	6
7	1	3	4	5
8	1	3	4	6
9	1	3	5	6
10	1	4	5	6
11	2	3	4	5
12	2	3	4	6
13	2	3	5	6
14	2	4	5	6
15	3	4	5	6

Example 65.7: Crossover Designs

In *crossover* experiments, the same experimental units or subjects are given multiple treatments in sequence, and the model for the response at any one period includes an effect for the treatment applied in the previous period. A good design for a crossover experiment will therefore be one which balances how often each treatment is preceded by each other treatment. Cox (1992) gives the following example of a balanced crossover experiment for paper production. In this experiment, the subjects are production runs of the mill, with the treatments being six different concentrations of pulp used in sequence. The following statements construct this design in a standard form, as shown in Output 65.7.1.

```
proc plan;
   factors Run=6 ordered Period=6 ordered;
   treatments Treatment=6 cyclic (1 2 6 3 5 4);
run;
```

Output 65.7.1 Crossover Design for Six Treatments

The PLAN Procedure

Plot Factors

Factor	Select	Levels	Order
Run	6	6	Ordered
Period	6	6	Ordered

Treatment Factors

Factor	Select	Levels	Order	Initial Block / Increment
Treatment	6	6	Cyclic	(1 2 6 3 5 4) / 1

Run	---Period--	-Treatment-
1	1 2 3 4 5 6	1 2 6 3 5 4
2	1 2 3 4 5 6	2 3 1 4 6 5
3	1 2 3 4 5 6	3 4 2 5 1 6
4	1 2 3 4 5 6	4 5 3 6 2 1
5	1 2 3 4 5 6	5 6 4 1 3 2
6	1 2 3 4 5 6	6 1 5 2 4 3

The construction method for this example is due to Williams (1949). The initial block for the treatment variable Treatment is defined thus for $n = 6$:

$$(1 \quad 2\ n \quad 3\ n-1 \quad \ldots \quad n/2\ n/2+2 \quad n/2)$$

This general form serves to generate a balanced crossover design for n treatments and n subjects in n periods when n is even. When n is odd, $2n$ subjects are required, with the initial blocks

$$(1 \quad 2\ n \quad 3\ n-1 \quad \ldots \quad n/2+1\ n/2)$$
$$(n/2\ n/2+1 \quad \ldots \quad n-1\ 3 \quad n\ 2 \quad 1)$$

respectively.

In order to randomize Williams' crossover designs, the following statements randomly permute the subjects and treatments:

```
proc plan seed=136149876;
   factors Run=6 ordered Period=6 ordered / noprint;
   treatments Treatment=6 cyclic (1 2 6 3 5 4);
   output out=RandomizedDesign
      Run       random
      Treatment random
      ;

/*
/ Relabel Period to obtain the same design as in Cox (1992).
/-------------------------------------------------------------------*/
data RandomizedDesign; set RandomizedDesign;
   Period = mod(Period+2,6)+1;
run;

proc sort data=RandomizedDesign;
   by Run Period;
proc transpose data=RandomizedDesign out=tDesign(drop=_name_);
   by notsorted Run;
   var Treatment;
data tDesign; set tDesign;
   rename COL1-COL6 = Period_1-Period_6;
proc print data=tDesign noobs;
run;
```

In the preceding, Run and Treatment are randomized by using the RANDOM option in the OUTPUT statement and new labels for Period are obtained in a subsequent DATA step. This Period relabeling is not necessary and might not be valid for Williams' designs in general; it is used in this example only to match results with those of Cox (1992). The SORT and TRANSPOSE steps then prepare the design to be printed in a standard form, shown in Output 65.7.2.

Output 65.7.2 Randomized Crossover Design

Run	Period_1	Period_2	Period_3	Period_4	Period_5	Period_6
1	3	6	2	5	4	1
2	5	3	4	6	1	2
3	1	4	5	2	6	3
4	2	1	6	4	3	5
5	6	5	1	3	2	4
6	4	2	3	1	5	6

The analysis of a crossover experiment requires for each observation a *carryover* variable whose values are the treatment in the preceding period. The following statements add such a variable to the randomized design constructed previously, displaying the values of the carryover variable for each run and period in Output 65.7.3.

```
proc sort data=RandomizedDesign;
   by Run Period;
data RandomizedDesign; set RandomizedDesign;
   by Run period;
   LagTreatment = lag(Treatment);
   if (first.Run) then LagTreatment = .;
run;

proc transpose data=RandomizedDesign out=tDesign(drop=_name_);
   by notsorted Run;
   var LagTreatment;
data tDesign; set tDesign;
   rename COL1-COL6 = Period_1-Period_6;
proc print data=tDesign noobs;
run;
```

Output 65.7.3 Lag Treatment Effect in Crossover Design

Run	Period_1	Period_2	Period_3	Period_4	Period_5	Period_6
1	.	3	6	2	5	4
2	.	5	3	4	6	1
3	.	1	4	5	2	6
4	.	2	1	6	4	3
5	.	6	5	1	3	2
6	.	4	2	3	1	5

Of course, the carryover variable has no effect in the first period, which is why it is coded with a missing value in this case.

The experimental EFFECT statement in PROC GLIMMIX provides a convenient mechanism for incorporating the carryover variable into the analysis. The following statements first add the observed data to the design to create the Mills data set. Then PROC GLIMMIX is invoked and the carryover effect is defined as a multimember effect (having only a single member in this case) with the NOEFFECT option. The NOEFFECT option specifies that the columns for the carryover effect will be all zero when LagTreatment is missing. GLIMMIX is a very general procedure, so ODS is used to trim down the results to show only the parts that are usually of interest in crossover analysis. For more information about the EFFECTS statement in PROC GLIMMIX, see the section "EFFECT Statement (Experimental)" on page 2134. The carryover analysis is shown in Output 65.7.4.

```
data Responses;
   input Response @@;
datalines;
56.7 53.8 54.4 54.4 58.9 54.5
58.5 60.2 61.3 54.4 59.1 59.8
55.7 60.7 56.7 59.9 56.6 59.6
57.3 57.7 55.2 58.1 60.2 60.2
```

```
53.7 57.1 59.2 58.9 58.9 59.6
58.1 55.7 58.9 56.6 59.6 57.5
;
data Mills;
   merge RandomizedDesign Responses;
run;

proc glimmix data=Mills outdesign=d;
   class Run Period Treatment LagTreatment;
   effect CarryOver = mm(LagTreatment / noeffect);
   model Response = Run Period Treatment CarryOver / htype=1;
   lsmeans Treatment CarryOver / obsmargins diff=anom e;
   ods select Tests1 LSMeans Diffs;
run;
```

Output 65.7.4 Carryover Analysis for Crossover Experiment

The GLIMMIX Procedure

Type I Tests of Fixed Effects

Effect	Num DF	Den DF	F Value	Pr > F
Run	5	15	13.76	<.0001
Period	5	15	7.19	0.0013
Treatment	5	15	22.95	<.0001
CarryOver	5	15	7.76	0.0009

Treatment Least Squares Means

Treatment	Margins	Estimate	Standard Error	DF	t Value	Pr > |t|
1	WORK.MILLS	57.1954	0.3220	15	177.65	<.0001
2	WORK.MILLS	57.6204	0.3220	15	178.97	<.0001
3	WORK.MILLS	59.1919	0.3220	15	183.85	<.0001
4	WORK.MILLS	59.2288	0.3220	15	183.97	<.0001
5	WORK.MILLS	57.9829	0.3220	15	180.10	<.0001
6	WORK.MILLS	55.0639	0.3220	15	171.03	<.0001

Differences of Treatment Least Squares Means

Treatment	_Treatment	Margins	Estimate	Standard Error	DF	t Value	Pr > |t|
1	Avg	WORK.MILLS	-0.5185	0.2948	15	-1.76	0.0990
2	Avg	WORK.MILLS	-0.09345	0.2948	15	-0.32	0.7556
3	Avg	WORK.MILLS	1.4780	0.2948	15	5.01	0.0002
4	Avg	WORK.MILLS	1.5149	0.2948	15	5.14	0.0001
5	Avg	WORK.MILLS	0.2690	0.2948	15	0.91	0.3758
6	Avg	WORK.MILLS	-2.6500	0.2948	15	-8.99	<.0001

Output 65.7.4 *continued*

CarryOver Least Squares Means

Carry Over	Margins	Estimate	Standard Error	DF	t Value	Pr > |t|
1	WORK.MILLS	Non-est	.	.	.	.
2	WORK.MILLS	Non-est	.	.	.	.
3	WORK.MILLS	Non-est	.	.	.	.
4	WORK.MILLS	Non-est	.	.	.	.
5	WORK.MILLS	Non-est	.	.	.	.
6	WORK.MILLS	Non-est	.	.	.	.

Differences of CarryOver Least Squares Means

Carry Over	_ Carry Over	Margins	Estimate	Standard Error	DF	t Value	Pr > |t|
1	Avg	WORK.MILLS	0.3726	0.3284	15	1.13	0.2743
2	Avg	WORK.MILLS	-0.2774	0.3284	15	-0.84	0.4116
3	Avg	WORK.MILLS	0.6512	0.3284	15	1.98	0.0660
4	Avg	WORK.MILLS	-1.3274	0.3284	15	-4.04	0.0011
5	Avg	WORK.MILLS	1.3976	0.3284	15	4.26	0.0007
6	Avg	WORK.MILLS	-0.8167	0.3284	15	-2.49	0.0252

The Type I analysis of variance indicates that all effects are significant—in particular, both the direct and the carryover effects of the treatment. In the presence of carryover effects, the LS-means need to be defined with some care. The LS-means for treatments computed using balanced margins for the carryover effect are inestimable, so the OBSMARGINS option is specified in the LSMEANS statement in order to use the observed margins instead. The observed margins take the absence of a carryover effect in the first period into account. Note that the LS-means themselves of the carryover effect are inestimable, but their differences are estimable. The LS-means of the direct effect of the treatment and the ANOM differences for the LS-means of their carryover effect match the "adjusted direct effects" and "adjusted residual effects," respectively, of Cox (1992).

References

Cochran, W. G. and Cox, G. M. (1957), *Experimental Designs*, Second Edition, New York: John Wiley & Sons.

Cox, D. R. (1992), *Planning of Experiments*, Wiley Classics Library Edition, New York: John Wiley & Sons.

Jarrett, R. G. and Hall, W. B. (1978), "Generalized Cyclic Incomplete Block Designs," *Biometrika*, 65, 397–401.

Williams, E. J. (1949), "Experimental Designs Balanced for the Estimation of Residual Effects of Treatments," *Australian Journal of Scientific Research, Series A*, 2, 149–168.

Chapter 66

The PLS Procedure

Contents

Overview: PLS Procedure **4765**
Basic Features 4766
Getting Started: PLS Procedure **4767**
Spectrometric Calibration 4767
Syntax: PLS Procedure **4775**
PROC PLS Statement 4775
BY Statement 4781
CLASS Statement 4782
ID Statement 4782
MODEL Statement 4782
OUTPUT Statement 4783
Details: PLS Procedure **4784**
Regression Methods 4784
Cross Validation 4788
Centering and Scaling 4790
Missing Values 4791
Displayed Output 4791
ODS Table Names 4792
ODS Graphics 4793
Examples: PLS Procedure **4795**
Example 66.1: Examining Model Details 4795
Example 66.2: Examining Outliers 4803
Example 66.3: Choosing a PLS Model by Test Set Validation 4805
References **4811**

Overview: PLS Procedure

The PLS procedure fits models by using any one of a number of linear predictive methods, including *partial least squares* (PLS). Ordinary least squares regression, as implemented in SAS/STAT procedures such as PROC GLM and PROC REG, has the single goal of minimizing sample response prediction error, seeking linear functions of the predictors that explain as much variation in each

response as possible. The techniques implemented in the PLS procedure have the additional goal of accounting for variation in the predictors, under the assumption that directions in the predictor space that are well sampled should provide better prediction for *new* observations when the predictors are highly correlated. All of the techniques implemented in the PLS procedure work by extracting successive linear combinations of the predictors, called *factors* (also called *components*, *latent vectors*, or *latent variables*), which optimally address one or both of these two goals—explaining response variation and explaining predictor variation. In particular, the method of partial least squares balances the two objectives, seeking factors that explain both response and predictor variation.

Note that the name "partial least squares" also applies to a more general statistical method that is *not* implemented in this procedure. The partial least squares method was originally developed in the 1960s by the econometrician Herman Wold (1966) for modeling "paths" of causal relation between any number of "blocks" of variables. However, the PLS procedure fits only *predictive* partial least squares models, with one "block" of predictors and one "block" of responses. If you are interested in fitting more general path models, you should consider using the CALIS procedure.

Basic Features

The techniques implemented by the PLS procedure are as follows:

- principal components regression, which extracts factors to explain as much predictor sample variation as possible
- reduced rank regression, which extracts factors to explain as much response variation as possible. This technique, also known as (maximum) redundancy analysis, differs from multivariate linear regression only when there are multiple responses.
- partial least squares regression, which balances the two objectives of explaining response variation and explaining predictor variation. Two different formulations for partial least squares are available: the original predictive method of Wold (1966) and the SIMPLS method of de Jong (1993).

The number of factors to extract depends on the data. Basing the model on more extracted factors improves the model fit to the observed data, but extracting too many factors can cause *overfitting*—that is, tailoring the model too much to the current data, to the detriment of future predictions. The PLS procedure enables you to choose the number of extracted factors by *cross validation*—that is, fitting the model to part of the data, minimizing the prediction error for the unfitted part, and iterating with different portions of the data in the roles of fitted and unfitted. Various methods of cross validation are available, including one-at-a-time validation and splitting the data into blocks. The PLS procedure also offers test set validation, where the model is fit to the entire primary input data set and the fit is evaluated over a distinct test data set.

You can use the general linear modeling approach of the GLM procedure to specify a model for your design, allowing for general polynomial effects as well as classification or ANOVA effects. You can save the model fit by the PLS procedure in a data set and apply it to new data by using the SCORE procedure.

The PLS procedure now uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the statistical graphics available with the PLS procedure, see the PLOTS options in the PROC PLS statements and the section "ODS Graphics" on page 4793.

Getting Started: PLS Procedure

Spectrometric Calibration

The example in this section illustrates basic features of the PLS procedure. The data are reported in Umetrics (1995); the original source is Lindberg, Persson, and Wold (1983). Suppose that you are researching pollution in the Baltic Sea, and you would like to use the spectra of samples of seawater to determine the amounts of three compounds present in samples from the Baltic Sea: lignin sulfonate (ls: pulp industry pollution), humic acids (ha: natural forest products), and optical whitener from detergent (dt). Spectrometric calibration is a type of problem in which partial least squares can be very effective. The predictors are the spectra emission intensities at different frequencies in sample spectrum, and the responses are the amounts of various chemicals in the sample.

For the purposes of calibrating the model, samples with known compositions are used. The calibration data consist of 16 samples of known concentrations of ls, ha, and dt, with spectra based on 27 frequencies (or, equivalently, wavelengths). The following statements create a SAS data set named Sample for these data.

```
data Sample;
   input obsnam $ v1-v27 ls ha dt @@;
   datalines;
EM1   2766 2610 3306 3630 3600 3438 3213 3051 2907 2844 2796
      2787 2760 2754 2670 2520 2310 2100 1917 1755 1602 1467
      1353 1260 1167 1101 1017         3.0110  0.0000   0.00
EM2   1492 1419 1369 1158  958  887  905  929  920  887  800
       710  617  535  451  368  296  241  190  157  128  106
        89   70   65   56   50         0.0000  0.4005   0.00
EM3   2450 2379 2400 2055 1689 1355 1109  908  750  673  644
       640  630  618  571  512  440  368  305  247  196  156
       120   98   80   61   50         0.0000  0.0000  90.63
EM4   2751 2883 3492 3570 3282 2937 2634 2370 2187 2070 2007
      1974 1950 1890 1824 1680 1527 1350 1206 1080  984  888
       810  732  669  630  582         1.4820  0.1580  40.00
EM5   2652 2691 3225 3285 3033 2784 2520 2340 2235 2148 2094
      2049 2007 1917 1800 1650 1464 1299 1140 1020  909  810
       726  657  594  549  507         1.1160  0.4104  30.45
EM6   3993 4722 6147 6720 6531 5970 5382 4842 4470 4200 4077
      4008 3948 3864 3663 3390 3090 2787 2481 2241 2028 1830
      1680 1533 1440 1314 1227         3.3970  0.3032  50.82
EM7   4032 4350 5430 5763 5490 4974 4452 3990 3690 3474 3357
```

```
      3300 3213 3147 3000 2772 2490 2220 1980 1779 1599 1440
      1320 1200 1119 1032  957         2.4280  0.2981  70.59
EM8   4530 5190 6910 7580 7510 6930 6150 5490 4990 4670 4490
      4370 4300 4210 4000 3770 3420 3060 2760 2490 2230 2060
      1860 1700 1590 1490 1380         4.0240  0.1153  89.39
EM9   4077 4410 5460 5857 5607 5097 4605 4170 3864 3708 3588
      3537 3480 3330 3192 2910 2610 2325 2064 1830 1638 1476
      1350 1236 1122 1044  963         2.2750  0.5040  81.75
EM10  3450 3432 3969 4020 3678 3237 2814 2487 2205 2061 2001
      1965 1947 1890 1776 1635 1452 1278 1128  981  867  753
       663  600  552  507  468         0.9588  0.1450 101.10
EM11  4989 5301 6807 7425 7155 6525 5784 5166 4695 4380 4197
      4131 4077 3972 3777 3531 3168 2835 2517 2244 2004 1809
      1620 1470 1359 1266 1167         3.1900  0.2530 120.00
EM12  5340 5790 7590 8390 8310 7670 6890 6190 5700 5380 5200
      5110 5040 4900 4700 4390 3970 3540 3170 2810 2490 2240
      2060 1870 1700 1590 1470         4.1320  0.5691 117.70
EM13  3162 3477 4365 4650 4470 4107 3717 3432 3228 3093 3009
      2964 2916 2838 2694 2490 2253 2013 1788 1599 1431 1305
      1194 1077  990  927  855         2.1600  0.4360  27.59
EM14  4380 4695 6018 6510 6342 5760 5151 4596 4200 3948 3807
      3720 3672 3567 3438 3171 2880 2571 2280 2046 1857 1680
      1548 1413 1314 1200 1119         3.0940  0.2471  61.71
EM15  4587 4200 5040 5289 4965 4449 3939 3507 3174 2970 2850
      2814 2748 2670 2529 2328 2088 1851 1641 1431 1284 1134
      1020  918  840  756  714         1.6040  0.2856 108.80
EM16  4017 4725 6090 6570 6354 5895 5346 4911 4611 4422 4314
      4287 4224 4110 3915 3600 3240 2913 2598 2325 2088 1917
      1734 1587 1452 1356 1257         3.1620  0.7012  60.00
;
```

Fitting a PLS Model

To isolate a few underlying spectral factors that provide a good predictive model, you can fit a PLS model to the 16 samples by using the following SAS statements:

```
proc pls data=sample;
   model ls ha dt = v1-v27;
run;
```

By default, the PLS procedure extracts at most 15 factors. The procedure lists the amount of variation accounted for by each of these factors, both individual and cumulative; this listing is shown in Figure 66.1.

Figure 66.1 PLS Variation Summary

The PLS Procedure

Percent Variation Accounted for
by Partial Least Squares Factors

Number of Extracted Factors	Model Effects Current	Model Effects Total	Dependent Variables Current	Dependent Variables Total
1	97.4607	97.4607	41.9155	41.9155
2	2.1830	99.6436	24.2435	66.1590
3	0.1781	99.8217	24.5339	90.6929
4	0.1197	99.9414	3.7898	94.4827
5	0.0415	99.9829	1.0045	95.4873
6	0.0106	99.9935	2.2808	97.7681
7	0.0017	99.9952	1.1693	98.9374
8	0.0010	99.9961	0.5041	99.4415
9	0.0014	99.9975	0.1229	99.5645
10	0.0010	99.9985	0.1103	99.6747
11	0.0003	99.9988	0.1523	99.8270
12	0.0003	99.9991	0.1291	99.9561
13	0.0002	99.9994	0.0312	99.9873
14	0.0004	99.9998	0.0065	99.9938
15	0.0002	100.0000	0.0062	100.0000

Note that all of the variation in both the predictors and the responses is accounted for by only 15 factors; this is because there are only 16 sample observations. More important, almost all of the variation is accounted for with even fewer factors—one or two for the predictors and three to eight for the responses.

Selecting the Number of Factors by Cross Validation

A PLS model is not complete until you choose the number of factors. You can choose the number of factors by using cross validation, in which the data set is divided into two or more groups. You fit the model to all groups except one, and then you check the capability of the model to predict responses for the group omitted. Repeating this for each group, you then can measure the overall capability of a given form of the model. The predicted residual sum of squares (PRESS) statistic is based on the residuals generated by this process.

To select the number of extracted factors by cross validation, you specify the CV= option with an argument that says which cross validation method to use. For example, a common method is split-sample validation, in which the different groups are composed of every nth observation beginning with the first, every nth observation beginning with the second, and so on. You can use the CV=SPLIT option to specify split-sample validation with $n = 7$ by default, as in the following SAS statements:

```
proc pls data=sample cv=split;
   model ls ha dt = v1-v27;
run;
```

The resulting output is shown in Figure 66.2 and Figure 66.3.

Figure 66.2 Split-Sample Validated PRESS Statistics for Number of Factors

The PLS Procedure

Split-sample Validation for the Number of Extracted Factors

Number of Extracted Factors	Root Mean PRESS
0	1.107747
1	0.957983
2	0.931314
3	0.520222
4	0.530501
5	0.586786
6	0.475047
7	0.477595
8	0.483138
9	0.485739
10	0.48946
11	0.521445
12	0.525653
13	0.531049
14	0.531049
15	0.531049

Minimum root mean PRESS	0.4750
Minimizing number of factors	6

Figure 66.3 PLS Variation Summary for Split-Sample Validated Model

Percent Variation Accounted for by Partial Least Squares Factors

Number of Extracted Factors	Model Effects		Dependent Variables	
	Current	Total	Current	Total
1	97.4607	97.4607	41.9155	41.9155
2	2.1830	99.6436	24.2435	66.1590
3	0.1781	99.8217	24.5339	90.6929
4	0.1197	99.9414	3.7898	94.4827
5	0.0415	99.9829	1.0045	95.4873
6	0.0106	99.9935	2.2808	97.7681

The absolute minimum PRESS is achieved with six extracted factors. Notice, however, that this is not much smaller than the PRESS for three factors. By using the CVTEST option, you can perform a statistical model comparison suggested by van der Voet (1994) to test whether this difference is significant, as shown in the following SAS statements:

```
proc pls data=sample cv=split cvtest(seed=12345);
   model ls ha dt = v1-v27;
run;
```

The model comparison test is based on a rerandomization of the data. By default, the seed for this randomization is based on the system clock, but it is specified here. The resulting output is shown in Figure 66.4 and Figure 66.5.

Figure 66.4 Testing Split-Sample Validation for Number of Factors

The PLS Procedure

Split-sample Validation for the Number of Extracted Factors

Number of Extracted Factors	Root Mean PRESS	T**2	Prob > T**2
0	1.107747	9.272858	0.0010
1	0.957983	10.62305	<.0001
2	0.931314	8.950878	0.0010
3	0.520222	5.133259	0.1440
4	0.530501	5.168427	0.1340
5	0.586786	6.437266	0.0150
6	0.475047	0	1.0000
7	0.477595	2.809763	0.4750
8	0.483138	7.189526	0.0110
9	0.485739	7.931726	0.0070
10	0.48946	6.612597	0.0150
11	0.521445	6.666235	0.0130
12	0.525653	7.092861	0.0080
13	0.531049	7.538298	0.0030
14	0.531049	7.538298	0.0030
15	0.531049	7.538298	0.0030

Minimum root mean PRESS	0.4750
Minimizing number of factors	6
Smallest number of factors with p > 0.1	3

Figure 66.5 PLS Variation Summary for Tested Split-Sample Validated Model

Percent Variation Accounted for by Partial Least Squares Factors

Number of Extracted Factors	Model Effects Current	Model Effects Total	Dependent Variables Current	Dependent Variables Total
1	97.4607	97.4607	41.9155	41.9155
2	2.1830	99.6436	24.2435	66.1590
3	0.1781	99.8217	24.5339	90.6929

The p-value of 0.1430 in comparing the cross validated residuals from models with 6 and 3 factors indicates that the difference between the two models is insignificant; therefore, the model with fewer factors is preferred. The variation summary shows that over 99% of the predictor variation and over 90% of the response variation are accounted for by the three factors.

Predicting New Observations

Now that you have chosen a three-factor PLS model for predicting pollutant concentrations based on sample spectra, suppose that you have two new samples. The following SAS statements create a data set containing the spectra for the new samples:

```
data newobs;
   input obsnam $ v1-v27 @@;
   datalines;
EM17  3933 4518 5637 6006 5721 5187 4641 4149 3789
      3579 3447 3381 3327 3234 3078 2832 2571 2274
      2040 1818 1629 1470 1350 1245 1134 1050  987
EM25  2904 2997 3255 3150 2922 2778 2700 2646 2571
      2487 2370 2250 2127 2052 1713 1419 1200  984
       795  648  525  426  351  291  240  204  162
;
```

You can apply the PLS model to these samples to estimate pollutant concentration. To do so, append the new samples to the original 16, and specify that the predicted values for all 18 be output to a data set, as shown in the following statements:

```
data all; set sample newobs;
proc pls data=all nfac=3;
   model ls ha dt = v1-v27;
   output out=pred p=p_ls p_ha p_dt;
run;

proc print data=pred;
   where (obsnam in ('EM17','EM25'));
   var obsnam p_ls p_ha p_dt;
run;
```

The new observations are not used in calculating the PLS model, since they have no response values. Their predicted concentrations are shown in Figure 66.6.

Figure 66.6 Predicted Concentrations for New Observations

Obs	obsnam	p_ls	p_ha	p_dt
17	EM17	2.54261	0.31877	81.4174
18	EM25	−0.24716	1.37892	46.3212

Finally, if you enable ODS graphics, PLS also displays by default a plot of the amount of variation accounted for by each factor, as well as a correlations loading plot that summarizes the first two

dimensions of the PLS model. The following statements, which are the same as the previous split-sample validation analysis but with ODS graphics enabled, additionally produce Figure 66.7 and Figure 66.8:

```
ods graphics on;

proc pls data=sample cv=split cvtest(seed=12345);
   model ls ha dt = v1-v27;
run;

ods graphics off;
```

Figure 66.7 Split-Sample Cross Validation Plot

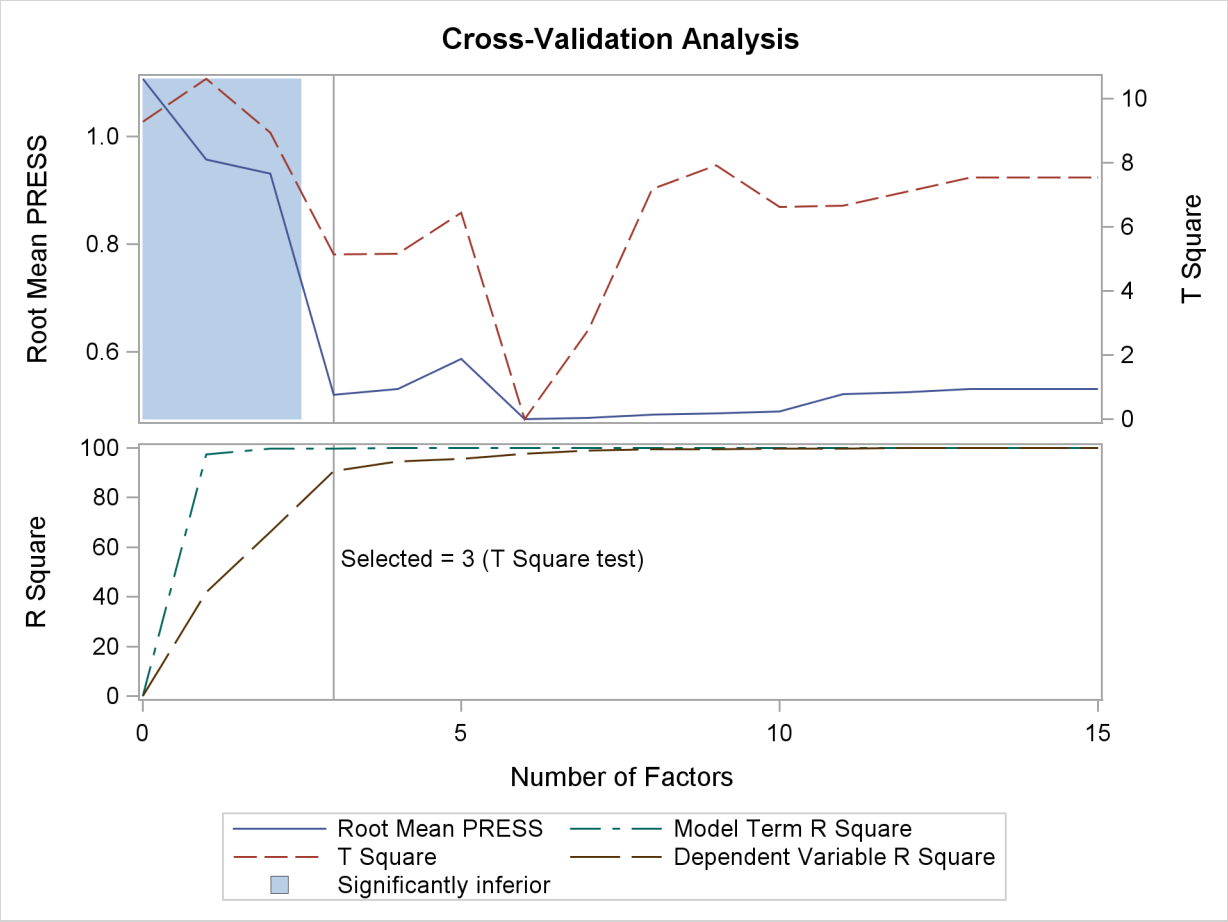
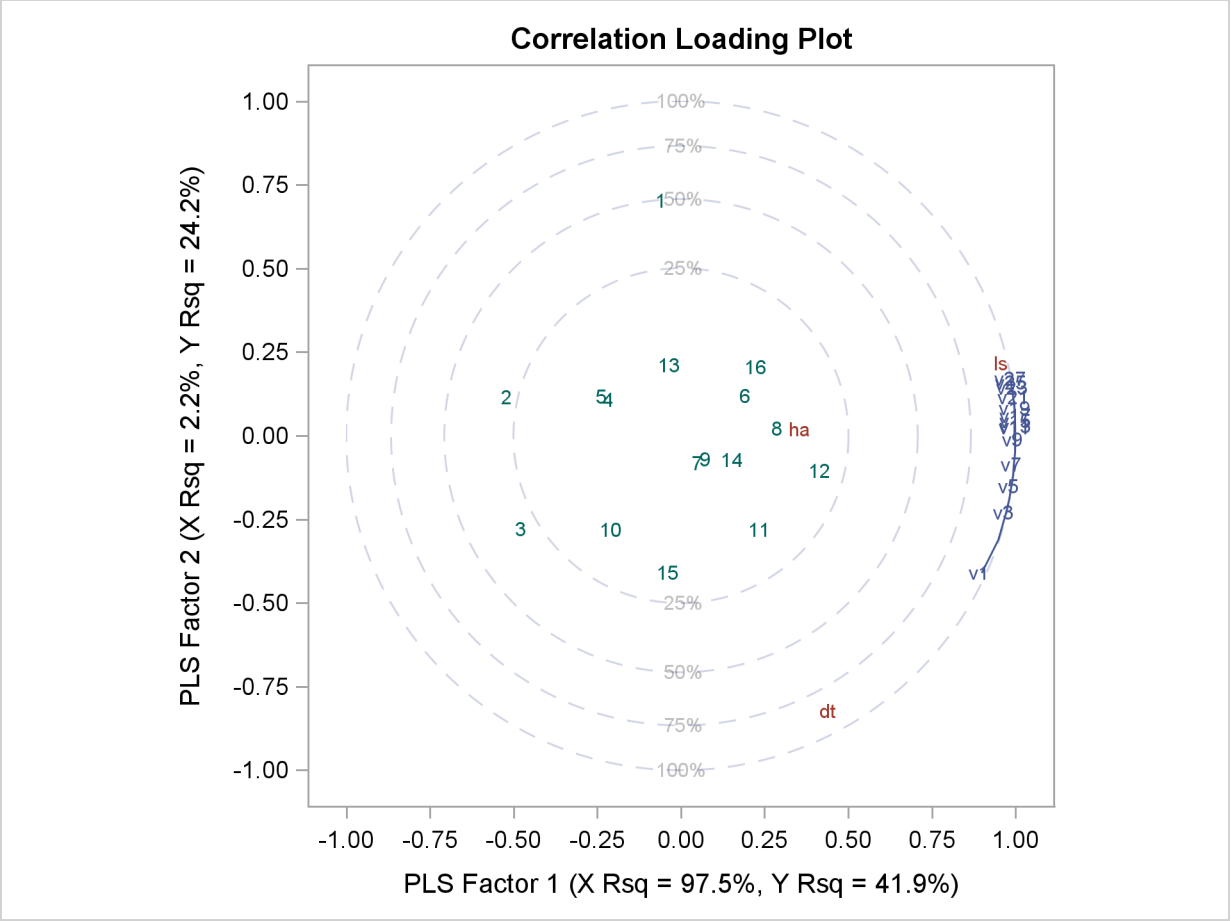


Figure 66.8 Correlation Loadings Plot



The cross validation plot in Figure 66.7 gives a visual representation of the selection of the optimum number of factors discussed previously. The correlation loadings plot is a compact summary of many features of the PLS model. For example, it shows that the first factor is highly positively correlated with all spectral values, indicating that it is approximately an average of them all; the second factor is positively correlated with the lowest frequencies and negatively correlated with the highest, indicating that it is approximately a contrast between the two ends of the spectrum. The observations, represented by their number in the data set on this plot, are generally spaced well apart, indicating that the data give good information about these first two factors. For more details on the interpretation of the correlation loadings plot, see the section "ODS Graphics" on page 4793 and Example 66.1.

Syntax: PLS Procedure

The following statements are available in PROC PLS. Items within the angle brackets are optional.

PROC PLS < *options* > ;
  BY *variables* ;
  CLASS *variables* < / *option* > ;
  ID *variables* ;
  MODEL *dependent-variables* = *effects* < / *options* > ;
  OUTPUT **OUT=***SAS-data-set* < *options* > ;

To analyze a data set, you must use the PROC PLS and MODEL statements. You can use the other statements as needed.

PROC PLS Statement

PROC PLS < *options* > ;

You use the PROC PLS statement to invoke the PLS procedure and, optionally, to indicate the analysis data and method. The following options are available.

CENSCALE
lists the centering and scaling information for each response and predictor.

CV=ONE

CV=SPLIT < (*n*) >

CV=BLOCK < (*n*) >

CV=RANDOM < (*cv-random-opts*) >

CV=TESTSET(*SAS-data-set***)**
specifies the cross validation method to be used. By default, no cross validation is performed. The method CV=ONE requests one-at-a-time cross validation, CV=SPLIT requests that every nth observation be excluded, CV=BLOCK requests that n blocks of consecutive observations be excluded, CV=RANDOM requests that observations be excluded at random, and CV=TESTSET(*SAS-data-set*) specifies a test set of observations to be used for validation (formally, this is called "test set validation" rather than "cross validation"). You can, optionally, specify n for CV=SPLIT and CV=BLOCK; the default is $n = 7$. You can also specify the following optional *cv-random-options* in parentheses after the CV=RANDOM option:

NITER=*n* specifies the number of random subsets to exclude. The default value is 10.

NTEST=*n* specifies the number of observations in each random subset chosen for exclusion. The default value is one-tenth of the total number of observations.

SEED=*n* specifies an integer used to start the pseudo-random number generator for selecting the random test set. If you do not specify a seed, or specify

a value less than or equal to zero, the seed is by default generated from reading the time of day from the computer's clock.

CVTEST <(*cvtest-options*)>
specifies that van der Voet's (1994) randomization-based model comparison test be performed to test models with different numbers of extracted factors against the model that minimizes the predicted residual sum of squares; see the section "Cross Validation" on page 4788 for more information. You can also specify the following *cv-test-options* in parentheses after the CVTEST option:

PVAL=*n* specifies the cutoff probability for declaring an insignificant difference. The default value is 0.10.

STAT=*test-statistic* specifies the test statistic for the model comparison. You can specify either T2, for Hotelling's T^2 statistic, or PRESS, for the predicted residual sum of squares. The default value is T2.

NSAMP=*n* specifies the number of randomizations to perform. The default value is 1000.

SEED=*n* specifies the seed value for randomization generation (the clock time is used by default).

DATA=*SAS-data-set*
names the SAS data set to be used by PROC PLS. The default is the most recently created data set.

DETAILS
lists the details of the fitted model for each successive factor. The details listed are different for different extraction methods; see the section "Displayed Output" on page 4791 for more information.

METHOD=PLS <(*PLS-options*)>
METHOD=SIMPLS
METHOD=PCR
METHOD=RRR
specifies the general factor extraction method to be used. The value PLS requests partial least squares, SIMPLS requests the SIMPLS method of de Jong (1993), PCR requests principal components regression, and RRR requests reduced rank regression. The default is METHOD=PLS. You can also specify the following optional *PLS-options* in parentheses after METHOD=PLS:

ALGORITHM=NIPALS | SVD | EIG | RLGW
names the specific algorithm used to compute extracted PLS factors. NIPALS requests the usual iterative NIPALS algorithm, SVD bases the extraction on the singular value decomposition of $X'Y$, EIG bases the extraction on the eigenvalue decomposition of $Y'XX'Y$, and RLGW is an iterative approach that is efficient when there are many predictors. ALGORITHM=SVD is the most accurate but least efficient approach; the default is ALGORITHM=NIPALS.

MAXITER=*n* specifies the maximum number of iterations for the NIPALS and RLGW algorithms. The default value is 200.

EPSILON=*n* specifies the convergence criterion for the NIPALS and RLGW algorithms. The default value is 10^{-12}.

MISSING=NONE

MISSING=AVG

MISSING=EM <(*EM-options*)>

specifies how observations with missing values are to be handled in computing the fit. The default is MISSING=NONE, for which observations with any missing variables (dependent or independent) are excluded from the analysis. MISSING=AVG specifies that the fit be computed by filling in missing values with the average of the nonmissing values for the corresponding variable. If you specify MISSING=EM, then the procedure first computes the model with MISSING=AVG and then fills in missing values by their predicted values based on that model and computes the model again. For both methods of imputation, the imputed values contribute to the centering and scaling values, and the difference between the imputed values and their final predictions contributes to the percentage of variation explained. You can also specify the following optional *EM-options* in parentheses after MISSING=EM:

MAXITER=*n* specifies the maximum number of iterations for the imputation/fit loop. The default value is 1. If you specify a large value of MAXITER=, then the loop will iterate until it converges (as controlled by the EPSILON= option).

EPSILON=*n* specifies the convergence criterion for the imputation/fit loop. The default value for is 10^{-8}. This option is effective only if you specify a large value for the MAXITER= option.

NFAC=*n*

specifies the number of factors to extract. The default is $\min\{15, p, N\}$, where p is the number of predictors (the number of dependent variables for METHOD=RRR) and N is the number of runs (observations). This is probably more than you need for most applications. Extracting too many factors can lead to an overfit model, one that matches the training data too well, sacrificing predictive ability. Thus, if you use the default NFAC= specification, you should also either use the CV= option to select the appropriate number of factors for the final model or consider the analysis to be preliminary and examine the results to determine the appropriate number of factors for a subsequent analysis.

NOCENTER

suppresses centering of the responses and predictors before fitting. This is useful if the analysis variables are already centered and scaled. See the section "Centering and Scaling" on page 4790 for more information.

NOCVSTDIZE

suppresses re-centering and rescaling of the responses and predictors before each model is fit in the cross validation. See the section "Centering and Scaling" on page 4790 for more information.

NOPRINT
suppresses the normal display of results. This is useful when you want only the output statistics saved in a data set. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System" for more information.

NOSCALE
suppresses scaling of the responses and predictors before fitting. This is useful if the analysis variables are already centered and scaled. See the section "Centering and Scaling" on page 4790 for more information.

PLOTS <(*global-plot-options*)> <= *plot-request* <(*options*)>>

PLOTS <(*global-plot-options*)> <= (*plot-request* <(*options*)> <... *plot-request* <(*options*)>>)>
controls the plots produced through ODS Graphics. When you specify only one plot request, you can omit the parentheses from around the plot request. For example:

```
plots=none
plots=cvplot
plots=(diagnostics cvplot)
plots(unpack)=diagnostics
plots(unpack)=(diagnostics corrload(trace=off))
```

You must enable ODS Graphics before requesting plots—for example, like this:

```
ods graphics on;
proc pls data=pentaTrain;
   model log_RAI = S1-S5 L1-L5 P1-P5;
run;
ods graphics off;
```

For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." If you have enabled ODS Graphics but do not specify the PLOTS= option, then PROC PLS produces by default a plot of the R-square analysis and a correlation loading plot summarizing the first two factors. The global plot options include the following:

FLIP
interchanges the X-axis and Y-axis dimensions for the score, weight, and loading plots.

ONLY
suppresses the default plots. Only plots specifically requested are displayed.

UNPACKPANEL

UNPACK
suppresses paneling. By default, multiple plots can appear in some output panels. Specify UNPACKPANEL to get each plot in a separate panel. You can specify PLOTS(UNPACKPANEL) to unpack only the default plots. You can also specify UNPACKPANEL as a suboption for certain specific plots, as discussed in the following.

The plot requests include the following:

ALL
produces all appropriate plots. You can specify other options with ALL—for example, to request all plots and unpack only the residuals, specify PLOTS=(ALL RESIDUALS(UNPACK)).

CORRLOAD < (TRACE = ON | OFF) >
produces a correlation loading plot (default). The TRACE= option controls how points corresponding to the X-loadings in the correlation loadings plot are depicted. By default, these points are depicted by the name of the corresponding model effect if there are 20 or fewer of them; otherwise, they are depicted by a connected "trace" through the points. You can use this option to change this behavior.

CVPLOT
produces a cross validation and R-square analysis. This plot requires the CV= option to be specified, and is displayed by default in this case.

DIAGNOSTICS < (UNPACK) >
produces a summary panel of the fit for each dependent variable. The summary by default consists of a panel for each dependent variable, with plots depicting the distribution of residuals and predicted values. You can use the UNPACK suboption to specify that the subplots be produced separately.

DMOD
produces the DMODX, DMODY, and DMODXY plots.

DMODX
produces a plot of the distance of each observation to the X model.

DMODXY
produces plots of the distance of each observation to the X and Y models.

DMODY
produces a plot of the distance of each observation to the Y model.

FIT
produces both the fit diagnostics and the ParmProfiles plot.

NONE
suppresses the display of graphics.

PARMPROFILES
produces profiles of the regression coefficients.

SCORES < (UNPACK | FLIP) >
produces the XScores, YScores, XYScores, and DModXY plots. You can use the UNPACK suboption to specify that the subplots for scores be produced separately, and the FLIP option to interchange their default X-axis and Y-axis dimensions.

RESIDUALS < (UNPACK) >
plots the residuals for each dependent variable against each independent variable. Residual plots are by default composed of multiple plots combined into a single panel. You can use the UNPACK suboption to specify that the subplots be produced separately.

VIP
produces profiles of variable importance factors.

WEIGHTS < (UNPACK | FLIP) >
produces all X and Y loading and weight plots, as well as the VIP plot. You can use the UNPACK suboption to specify that the subplots for weights and loadings be produced separately, and the FLIP option to interchange their default X-axis and Y-axis dimensions.

XLOADINGPLOT < (UNPACK | FLIP) >
produces a scatter plot matrix of X-loadings against each other. Loading scatter plot matrices are by default composed of multiple plots combined into a single panel. You can use the UNPACK suboption to specify that the subplots be produced separately, and the FLIP option to interchange the default X-axis and Y-axis dimensions.

XLOADINGPROFILES
produces profiles of the X-loadings.

XSCORES < (UNPACK | FLIP) >
produces a scatter plot matrix of X-scores against each other. Score scatter plot matrices are by default composed of multiple plots combined into a single panel. You can use the UNPACK suboption to specify that the subplots be produced separately, and the FLIP option to interchange the default X-axis and Y-axis dimensions.

XWEIGHTPLOT < (UNPACK | FLIP) >
produces a scatter plot matrix of X-weights against each other. Weight scatter plot matrices are by default composed of multiple plots combined into a single panel. You can use the UNPACK suboption to specify that the subplots be produced separately, and the FLIP option to interchange the default X-axis and Y-axis dimensions.

XWEIGHTPROFILES
produces profiles of the X-weights.

XYSCORES < (UNPACK) >
produces a scatter plot matrix of X-scores against Y-scores. Score scatter plot matrices are by default composed of multiple plots combined into a single panel. You can use the UNPACK suboption to specify that the subplots be produced separately.

YSCORES < (UNPACK | FLIP) >
produces a scatter plot matrix of Y-scores against each other. Score scatter plot matrices are by default composed of multiple plots combined into a single panel. You can use the UNPACK suboption to specify that the subplots be produced separately, and the FLIP option to interchange the default X-axis and Y-axis dimensions.

YWEIGHTPLOT < (UNPACK | FLIP) >
produces a scatter plot matrix of Y-weights against each other. Weight scatter plot matrices are by default composed of multiple plots combined into a single panel. You can use the UNPACK suboption to specify that the subplots be produced separately, and the FLIP option to interchange the default X-axis and Y-axis dimensions.

VARSCALE
specifies that continuous model variables be centered and scaled prior to centering and scaling the model effects in which they are involved. The rescaling specified by the VARSCALE option is sometimes more appropriate if the model involves crossproducts between model variables; however, the VARSCALE option still might not produce the model you expect. See the section "Centering and Scaling" on page 4790 for more information.

VARSS
lists, in addition to the average response and predictor sum of squares accounted for by each successive factor, the amount of variation accounted for in each response and predictor.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC PLS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If you specify more than one BY statement, the procedure uses only the latest BY statement and ignores any previous ones.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the option NOTSORTED or DESCENDING in the BY statement for the PLS procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* < / *option* > **;**

The CLASS statement names the classification variables to be used in the analysis. If the CLASS statement is used, it must appear before the MODEL statement.

Classification variables can be either character or numeric. By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. Prior to SAS 9, class levels were determined by using no more than the first 16 characters of the formatted values. If you want to revert to this previous behavior, you can use the TRUNCATE option in the CLASS statement. In any case, you can use formats to group values into levels. See the discussion of the FORMAT procedure in the *Base SAS Procedures Guide* and the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

Any variable in the model that is not listed in the CLASS statement is assumed to be continuous. Continuous variables must be numeric.

You can specify the following option in the CLASS statement after a slash(/):

TRUNCATE
specifies that class levels should be determined by using only up to the first 16 characters of the formatted values of CLASS variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases prior to SAS 9.

ID Statement

ID *variables* **;**

The ID statement names variables whose values are used to label observations in plots. If you do not specify an ID statement, then each observations is labeled in plots by its corresponding observation number.

MODEL Statement

MODEL *response-variables = predictor-effects* < / *options* > **;**

The MODEL statement names the responses and the predictors, which determine the **Y** and **X** matrices of the model, respectively. Usually you simply list the names of the predictor variables as the model effects, but you can also use the effects notation of PROC GLM to specify polynomial effects and interactions; see the section "Specification of Effects" on page 2488 in Chapter 39, "The GLM Procedure" for further details. The MODEL statement is required. You can specify only one

MODEL statement (in contrast to the REG procedure, for example, which allows several MODEL statements in the same PROC REG run).

You can specify the following options in the MODEL statement after a slash (/).

INTERCEPT
By default, the responses and predictors are centered; thus, no intercept is required in the model. You can specify the INTERCEPT option to override the default.

SOLUTION
lists the coefficients of the final predictive model for the responses. The coefficients for predicting the centered and scaled responses based on the centered and scaled predictors are displayed, as well as the coefficients for predicting the raw responses based on the raw predictors.

OUTPUT Statement

OUTPUT *OUT= SAS-data-set keyword=names < . . . keyword=names >* **;**

You use the OUTPUT statement to specify a data set to receive quantities that can be computed for every input observation, such as extracted factors and predicted values. The following *keywords* are available:

PREDICTED	predicted values for responses
YRESIDUAL	residuals for responses
XRESIDUAL	residuals for predictors
XSCORE	extracted factors (X-scores, latent vectors, latent variables, T)
YSCORE	extracted responses (Y-scores, U)
STDY	standardized (centered and scaled) responses
STDX	standardized (centered and scaled) predictors
H	approximate leverage
PRESS	approximate predicted residuals
TSQUARE	scaled sum of squares of score values
STDXSSE	sum of squares of residuals for standardized predictors
STDYSSE	sum of squares of residuals for standardized responses

Suppose that there are N_x predictors and N_y responses and that the model has N_f selected factors.

- The keywords XRESIDUAL and STDX define an output variable for each predictor, so N_x names are required after each one.
- The keywords PREDICTED, YRESIDUAL, STDY, and PRESS define an output variable for each response, so N_y names are required after each of these keywords.

- The keywords XSCORE and YSCORE specify an output variable for each selected model factor. For these keywords, you provide only one base name, and the variables corresponding to each successive factor are named by appending the factor number to the base name. For example, if $N_f = 3$, then a specification of XSCORE=T would produce the variables T1, T2, and T3.
- Finally, the keywords H, TSQUARE, STDXSSE, and STDYSSE each specify a single output variable, so only one name is required after each of these keywords.

Details: PLS Procedure

Regression Methods

All of the predictive methods implemented in PROC PLS work essentially by finding linear combinations of the predictors (factors) to use to predict the responses linearly. The methods differ only in how the factors are derived, as explained in the following sections.

Partial Least Squares

Partial least squares (PLS) works by extracting one factor at a time. Let $\mathbf{X} = \mathbf{X}_0$ be the centered and scaled matrix of predictors and let $\mathbf{Y} = \mathbf{Y}_0$ be the centered and scaled matrix of response values. The PLS method starts with a linear combination $\mathbf{t} = \mathbf{X}_0\mathbf{w}$ of the predictors, where $\mathbf{t}$ is called a *score* vector and $\mathbf{w}$ is its associated *weight* vector. The PLS method predicts both $\mathbf{X}_0$ and $\mathbf{Y}_0$ by regression on $\mathbf{t}$:

$$\begin{aligned} \hat{\mathbf{X}}_0 &= \mathbf{t}\mathbf{p}', \quad \text{where} \quad \mathbf{p}' = (\mathbf{t}'\mathbf{t})^{-1}\mathbf{t}'\mathbf{X}_0 \\ \hat{\mathbf{Y}}_0 &= \mathbf{t}\mathbf{c}', \quad \text{where} \quad \mathbf{c}' = (\mathbf{t}'\mathbf{t})^{-1}\mathbf{t}'\mathbf{Y}_0 \end{aligned}$$

The vectors $\mathbf{p}$ and $\mathbf{c}$ are called the X- and Y-*loadings*, respectively.

The specific linear combination $\mathbf{t} = \mathbf{X}_0\mathbf{w}$ is the one that has maximum covariance $\mathbf{t}'\mathbf{u}$ with some response linear combination $\mathbf{u} = \mathbf{Y}_0\mathbf{q}$. Another characterization is that the X- and Y-weights $\mathbf{w}$ and $\mathbf{q}$ are proportional to the first left and right singular vectors of the covariance matrix $\mathbf{X}_0'\mathbf{Y}_0$ or, equivalently, the first eigenvectors of $\mathbf{X}_0'\mathbf{Y}_0\mathbf{Y}_0'\mathbf{X}_0$ and $\mathbf{Y}_0'\mathbf{X}_0\mathbf{X}_0'\mathbf{Y}_0$, respectively.

This accounts for how the first PLS factor is extracted. The second factor is extracted in the same way by replacing $\mathbf{X}_0$ and $\mathbf{Y}_0$ with the X- and Y-residuals from the first factor:

$$\begin{aligned} \mathbf{X}_1 &= \mathbf{X}_0 - \hat{\mathbf{X}}_0 \\ \mathbf{Y}_1 &= \mathbf{Y}_0 - \hat{\mathbf{Y}}_0 \end{aligned}$$

These residuals are also called the *deflated* $\mathbf{X}$ and $\mathbf{Y}$ blocks. The process of extracting a score vector and deflating the data matrices is repeated for as many extracted factors as are wanted.

SIMPLS

Note that each extracted PLS factor is defined in terms of different X-variables $\mathbf{X}_i$. This leads to difficulties in comparing different scores, weights, and so forth. The SIMPLS method of de Jong (1993) overcomes these difficulties by computing each score $\mathbf{t}_i = \mathbf{X}\mathbf{r}_i$ in terms of the original (centered and scaled) predictors $\mathbf{X}$. The SIMPLS X-weight vectors r_i are similar to the eigenvectors of $\mathbf{S}\mathbf{S}' = \mathbf{X}'\mathbf{Y}\mathbf{Y}'\mathbf{X}$, but they satisfy a different orthogonality condition. The $\mathbf{r}_1$ vector is just the first eigenvector $\mathbf{e}_1$ (so that the first SIMPLS score is the same as the first PLS score), but whereas the second eigenvector maximizes

$$\mathbf{e}_1' S S' \mathbf{e}_2 \text{ subject to } \mathbf{e}_1'\mathbf{e}_2 = 0$$

the second SIMPLS weight $\mathbf{r}_2$ maximizes

$$\mathbf{r}_1' S S' \mathbf{r}_2 \text{ subject to } \mathbf{r}_1' X' X \mathbf{r}_2 = \mathbf{t}_1'\mathbf{t}_2 = 0$$

The SIMPLS scores are identical to the PLS scores for one response but slightly different for more than one response; see de Jong (1993) for details. The X- and Y-loadings are defined as in PLS, but since the scores are all defined in terms of $\mathbf{X}$, it is easy to compute the overall model coefficients $\mathbf{B}$:

$$\begin{aligned}
\hat{\mathbf{Y}} &= \sum_i \mathbf{t}_i \mathbf{c}_i' \\
&= \sum_i \mathbf{X}\mathbf{r}_i \mathbf{c}_i' \\
&= \mathbf{X}\mathbf{B}, \text{ where } \mathbf{B} = \mathbf{R}\mathbf{C}'
\end{aligned}$$

Principal Components Regression

Like the SIMPLS method, principal components regression (PCR) defines all the scores in terms of the original (centered and scaled) predictors $\mathbf{X}$. However, unlike both the PLS and SIMPLS methods, the PCR method chooses the X-weights/X-scores without regard to the response data. The X-scores are chosen to explain as much variation in $\mathbf{X}$ as possible; equivalently, the X-weights for the PCR method are the eigenvectors of the predictor covariance matrix $\mathbf{X}'\mathbf{X}$. Again, the X- and Y-loadings are defined as in PLS; but, as in SIMPLS, it is easy to compute overall model coefficients for the original (centered and scaled) responses $\mathbf{Y}$ in terms of the original predictors $\mathbf{X}$.

Reduced Rank Regression

As discussed in the preceding sections, partial least squares depends on selecting factors $\mathbf{t} = \mathbf{X}\mathbf{w}$ of the predictors and $\mathbf{u} = \mathbf{Y}\mathbf{q}$ of the responses that have maximum covariance, whereas principal components regression effectively ignores $\mathbf{u}$ and selects $\mathbf{t}$ to have maximum variance, subject to orthogonality constraints. In contrast, reduced rank regression selects $\mathbf{u}$ to account for as much variation in the *predicted* responses as possible, effectively ignoring the predictors for the purposes of factor extraction. In reduced rank regression, the Y-weights $\mathbf{q}_i$ are the eigenvectors of the covariance matrix $\hat{\mathbf{Y}}_{\text{LS}}'\hat{\mathbf{Y}}_{\text{LS}}$ of the responses predicted by ordinary least squares regression; the X-scores are the projections of the Y-scores $\mathbf{Y}\mathbf{q}_i$ onto the X space.

Relationships between Methods

When you develop a predictive model, it is important to consider not only the explanatory power of the model for current responses, but also how well sampled the predictive functions are, since this affects how well the model can extrapolate to future observations. All of the techniques implemented in the PLS procedure work by extracting successive factors, or linear combinations of the predictors, that optimally address one or both of these two goals—explaining response variation and explaining predictor variation. In particular, principal components regression selects factors that explain as much predictor variation as possible, reduced rank regression selects factors that explain as much response variation as possible, and partial least squares balances the two objectives, seeking for factors that explain both response and predictor variation.

To see the relationships between these methods, consider how each one extracts a single factor from the following artificial data set consisting of two predictors and one response:

```
data data;
   input x1 x2 y;
   datalines;
    3.37651  2.30716        0.75615
    0.74193 -0.88845        1.15285
    4.18747  2.17373        1.42392
    0.96097  0.57301        0.27433
   -1.11161 -0.75225       -0.25410
   -1.38029 -1.31343       -0.04728
    1.28153 -0.13751        1.00341
   -1.39242 -2.03615        0.45518
    0.63741  0.06183        0.40699
   -2.52533 -1.23726       -0.91080
    2.44277  3.61077       -0.82590
   ;

proc pls data=data nfac=1 method=rrr;
   model y = x1 x2;
run;

proc pls data=data nfac=1 method=pcr;
   model y = x1 x2;
run;

proc pls data=data nfac=1 method=pls;
   model y = x1 x2;
run;
```

The amount of model and response variation explained by the first factor for each method is shown in Figure 66.9 through Figure 66.11.

Figure 66.9 Variation Explained by First Reduced Rank Regression Factor

The PLS Procedure

Percent Variation Accounted for by Reduced Rank Regression Factors

Number of Extracted Factors	Model Effects Current	Model Effects Total	Dependent Variables Current	Dependent Variables Total
1	15.0661	15.0661	100.0000	100.0000

Figure 66.10 Variation Explained by First Principal Components Regression Factor

The PLS Procedure

Percent Variation Accounted for by Principal Components

Number of Extracted Factors	Model Effects Current	Model Effects Total	Dependent Variables Current	Dependent Variables Total
1	92.9996	92.9996	9.3787	9.3787

Figure 66.11 Variation Explained by First Partial Least Squares Regression Factor

The PLS Procedure

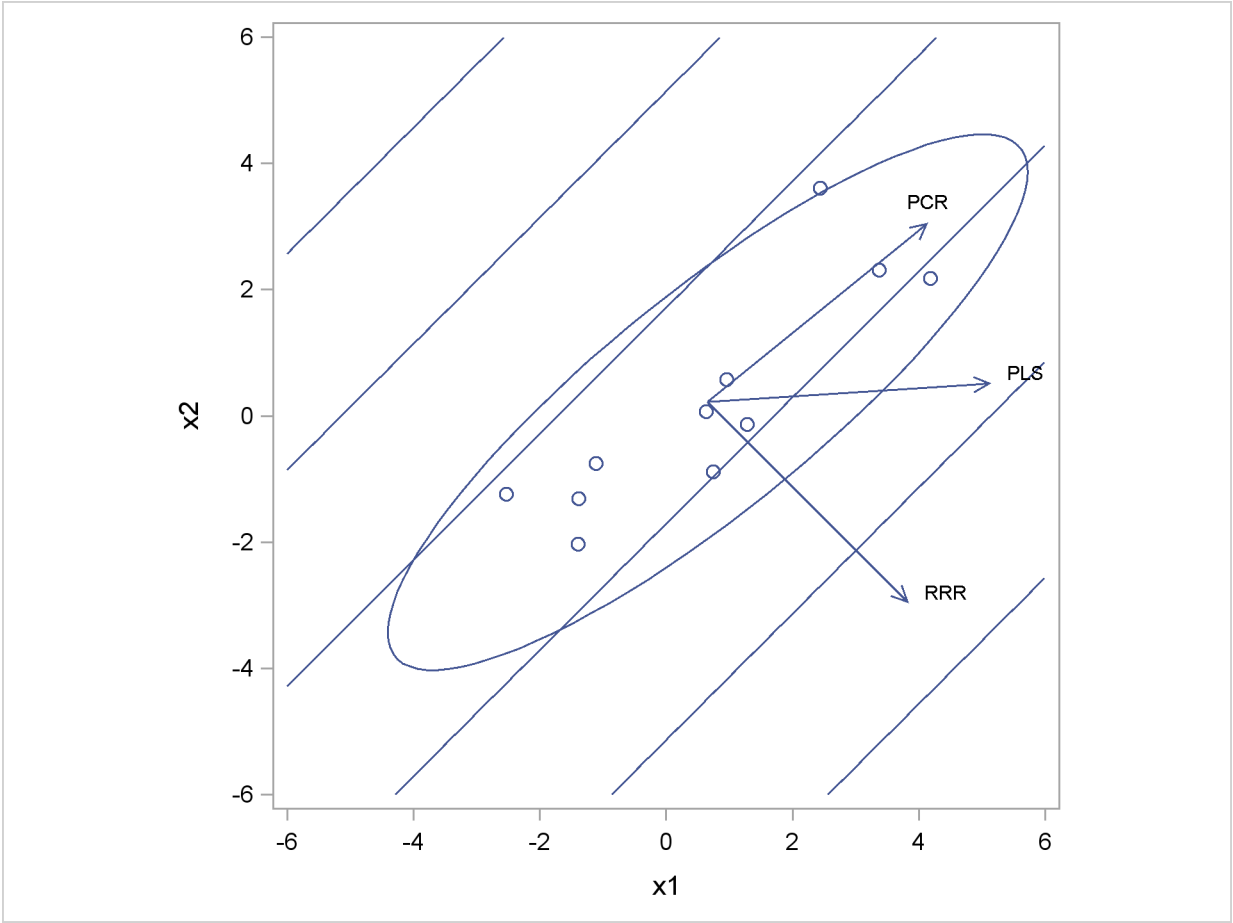
Percent Variation Accounted for by Partial Least Squares Factors

Number of Extracted Factors	Model Effects Current	Model Effects Total	Dependent Variables Current	Dependent Variables Total
1	88.5357	88.5357	26.5304	26.5304

Notice that, while the first reduced rank regression factor explains *all* of the response variation, it accounts for only about 15% of the predictor variation. In contrast, the first principal components regression factor accounts for most of the predictor variation (93%) but only 9% of the response variation. The first partial least squares factor accounts for only slightly less predictor variation than principal components but about three times as much response variation.

Figure 66.12 illustrates how partial least squares balances the goals of explaining response and predictor variation in this case.

Figure 66.12 Depiction of First Factors for Three Different Regression Methods



The ellipse shows the general shape of the 11 observations in the predictor space, with the contours of increasing y overlaid. Also shown are the directions of the first factor for each of the three methods. Notice that, while the predictors vary most in the x1 = x2 direction, the response changes most in the orthogonal x1 = -x2 direction. This explains why the first principal component accounts for little variation in the response and why the first reduced rank regression factor accounts for little variation in the predictors. The direction of the first partial least squares factor represents a compromise between the other two directions.

Cross Validation

None of the regression methods implemented in the PLS procedure fit the observed data any better than ordinary least squares (OLS) regression; in fact, all of the methods approach OLS as more factors are extracted. The crucial point is that, when there are many predictors, OLS can *overfit* the observed data; biased regression methods with fewer extracted factors can provide better predictability of *future* observations. However, as the preceding observations imply, the quality of the observed data fit cannot be used to choose the number of factors to extract; the number of extracted

factors must be chosen on the basis of how well the model fits observations not involved in the modeling procedure itself.

One method of choosing the number of extracted factors is to fit the model to only part of the available data (the *training set*) and to measure how well models with different numbers of extracted factors fit the other part of the data (the *test set*). This is called *test set validation*. However, it is rare that you have enough data to make both parts large enough for pure test set validation to be useful. Alternatively, you can make several different divisions of the observed data into training set and test set. This is called *cross validation*, and there are several different types. In *one-at-a-time* cross validation, the first observation is held out as a single-element test set, with all other observations as the training set; next, the second observation is held out, then the third, and so on. Another method is to hold out successive blocks of observations as test sets—for example, observations 1 through 7, then observations 8 through 14, and so on; this is known as *blocked* validation. A similar method is *split-sample* cross validation, in which successive groups of widely separated observations are held out as the test set—for example, observations {1, 11, 21, ...}, then observations {2, 12, 22, ...}, and so on. Finally, test sets can be selected from the observed data randomly; this is known as *random sample* cross validation.

Which validation you should use depends on your data. Test set validation is preferred when you have enough data to make a division into a sizable training set and test set that represent the predictive population well. You can specify that the number of extracted factors be selected by test set validation by using the CV=TESTSET(*data set*) option, where *data set* is the name of the data set containing the test set. If you do not have enough data for test set validation, you can use one of the cross validation techniques. The most common technique is one-at-a-time validation (which you can specify with the CV=ONE option or just the CV option), unless the observed data are serially correlated, in which case either blocked or split-sample validation might be more appropriate (CV=BLOCK or CV=SPLIT); you can specify the number of test sets in blocked or split-sample validation with a number in parentheses after the CV= option. Note that CV=ONE is the most computationally intensive of the cross validation methods, since it requires a recomputation of the PLS model for every input observation. Also, note that using random subset selection with CV=RANDOM might lead two different researchers to produce different PLS models on the same data (unless the same seed is used).

Whichever validation method you use, the number of factors chosen is usually the one that minimizes the predicted residual sum of squares (PRESS); this is the default choice if you specify any of the CV methods with PROC PLS. However, often models with fewer factors have PRESS statistics that are only marginally larger than the absolute minimum. To address this, van der Voet (1994) has proposed a statistical test for comparing the predicted residuals from different models; when you apply van der Voet's test, the number of factors chosen is the fewest with residuals that are insignificantly larger than the residuals of the model with minimum PRESS.

To see how van der Voet's test works, let $R_{i,jk}$ be the jth predicted residual for response k for the model with i extracted factors; the PRESS statistic is $\sum_{jk} R^2_{i,jk}$. Also, let $i_{\min}$ be the number of factors for which PRESS is minimized. The critical value for van der Voet's test is based on the differences between squared predicted residuals

$$D_{i,jk} = R^2_{i,jk} - R^2_{i_{\min},jk}$$

One alternative for the critical value is $C_i = \sum_{jk} D_{i,jk}$, which is just the difference between the

PRESS statistics for i and $i_{\min}$ factors; alternatively, van der Voet suggests Hotelling's T^2 statistic $C_i = \mathbf{d}'_{i,\cdot} S_i^{-1} \mathbf{d}_{i,\cdot}$, where $\mathbf{d}_{i,\cdot}$ is the sum of the vectors $\mathbf{d}_{i,j} = \{D_{i,j1}, \ldots, D_{i,jN_y}\}'$ and S_i is the sum of squares and crossproducts matrix

$$S_i = \sum_j \mathbf{d}_{i,j} \mathbf{d}'_{i,j}$$

Virtually, the significance level for van der Voet's test is obtained by comparing C_i with the distribution of values that result from randomly exchanging $R^2_{i,jk}$ and $R^2_{i_{\min},jk}$. In practice, a Monte Carlo sample of such values is simulated and the significance level is approximated as the proportion of simulated critical values that are greater than C_i. If you apply van der Voet's test by specifying the CVTEST option, then, by default, the number of extracted factors chosen is the least number with an approximate significance level that is greater than 0.10.

Centering and Scaling

By default, the predictors and the responses are centered and scaled to have mean 0 and standard deviation 1. Centering the predictors and the responses ensures that the criterion for choosing successive factors is based on how much *variation* they explain, in either the predictors or the responses or both. (See the section "Regression Methods" on page 4784 for more details on how different methods explain variation.) Without centering, both the mean variable value and the variation around that mean are involved in selecting factors. Scaling serves to place all predictors and responses on an equal footing relative to their variation in the data. For example, if Time and Temp are two of the predictors, then scaling says that a change of std(Time) in Time is roughly equivalent to a change of std(Temp) in Temp.

Usually, both the predictors and responses should be centered and scaled. However, if their values already represent variation around a nominal or target value, then you can use the NOCENTER option in the PROC PLS statement to suppress centering. Likewise, if the predictors or responses are already all on comparable scales, then you can use the NOSCALE option to suppress scaling.

Note that, if the predictors involve crossproduct terms, then, by default, the variables are *not* standardized before standardizing the crossproduct. That is, if the ith values of two predictors are denoted x_i^1 and x_i^2, then the default standardized ith value of the crossproduct is

$$\frac{x_i^1 x_i^2 - \text{mean}_j(x_j^1 x_j^2)}{\text{std}_j(x_j^1 x_j^2)}$$

If you want the crossproduct to be based instead on standardized variables

$$\frac{x_i^1 - m^1}{s^1} \times \frac{x_i^2 - m^2}{s^2}$$

where $m^k = \text{mean}_j(x_j^k)$ and $s^k = \text{std}_j(x_j^k)$ for $k = 1, 2$, then you should use the VARSCALE option in the PROC PLS statement. Standardizing the variables separately is usually a good idea, but unless the model also contains all crossproducts nested within each term, the resulting model

might not be equivalent to a simple linear model in the same terms. To see this, note that a model involving the crossproduct of two standardized variables

$$\frac{x_i^1 - m^1}{s^1} \times \frac{x_i^2 - m^2}{s^2} = x_i^1 x_i^2 \frac{1}{s^1 s^2} - x_i^1 \frac{m^2}{s^1 s^2} - x_i^2 \frac{m^1}{s^1 s^2} + \frac{m^1 m^2}{s^1 s^2}$$

involves both the crossproduct term and the linear terms for the unstandardized variables.

When cross validation is performed for the number of effects, there is some disagreement among practitioners as to whether each cross validation training set should be retransformed. By default, PROC PLS does so, but you can suppress this behavior by specifying the NOCVSTDIZE option in the PROC PLS statement.

Missing Values

By default, PROC PLS handles missing values very simply. Observations with any missing independent variables (including all classification variables) are excluded from the analysis, and no predictions are computed for such observations. Observations with no missing independent variables but any missing dependent variables are also excluded from the analysis, but predictions are computed.

However, the MISSING= option in the PROC PLS statement provides more sophisticated ways of modeling in the presence of missing values. If you specify MISSING=AVG or MISSING=EM, then all observations in the input data set contribute to both the analysis and the OUTPUT OUT= data set. With MISSING=AVG, the fit is computed by filling in missing values with the average of the nonmissing values for the corresponding variable. With MISSING=EM, the procedure first computes the model with MISSING=AVG, then fills in missing values with their predicted values based on that model and computes the model again. Alternatively, you can specify MISSING=EM(MAXITER=*n*) with a large value of *n* in order to perform this imputation/fit loop until convergence.

Displayed Output

By default, PROC PLS displays just the amount of predictor and response variation accounted for by each factor.

If you perform a cross validation for the number of factors by specifying the CV option in the PROC PLS statement, then the procedure displays a summary of the cross validation for each number of factors, along with information about the optimal number of factors.

If you specify the DETAILS option in the PROC PLS statement, then details of the fitted model are displayed for each successive factor. These details for each number of factors include the following:

- the predictor loadings
- the predictor weights

- the response weights
- the coded regression coefficients (for METHOD=SIMPLS, PCR, or RRR)

If you specify the CENSCALE option in the PROC PLS statement, then centering and scaling information for each response and predictor is displayed.

If you specify the VARSS option in the PROC PLS statement, the procedure displays, in addition to the average response and predictor sum of squares accounted for by each successive factor, the amount of variation accounted for in each response and predictor.

If you specify the SOLUTION option in the MODEL statement, then PROC PLS displays the coefficients of the final predictive model for the responses. The coefficients for predicting the centered and scaled responses based on the centered and scaled predictors are displayed, as well as the coefficients for predicting the raw responses based on the raw predictors.

ODS Table Names

PROC PLS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 66.1. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 66.1 ODS Tables Produced by PROC PLS

ODS Table Name	Description	Statement	Option
CVResults	Results of cross validation	PROC	CV
CenScaleParms	Parameter estimates for centered and scaled data	MODEL	SOLUTION
CodedCoef	Coded coefficients	PROC	DETAILS
MissingIterations	Iterations for missing value imputation	PROC	MISSING=EM
ModelInfo	Model information	PROC	default
NObs	Number of observations	PROC	default
ParameterEstimates	Parameter estimates for raw data	MODEL	SOLUTION
PercentVariation	Variation accounted for by each factor	PROC	default
ResidualSummary	Residual summary from cross validation	PROC	CV
XEffectCenScale	Centering and scaling information for predictor effects	PROC	CENSCALE
XLoadings	Loadings for independents	PROC	DETAILS
XVariableCenScale	Centering and scaling information for predictor variables	PROC	CENSCALE and VARSCALE
XWeights	Weights for independents	PROC	DETAILS
YVariableCenScale	Centering and scaling information for responses	PROC	CENSCALE
YWeights	Weights for dependents	PROC	DETAILS

ODS Graphics

This section describes the use of ODS for creating statistical graphs with the PLS procedure. To request these graphs you must specify the ODS GRAPHICS statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

When the ODS GRAPHICS are in effect, by default the PLS procedure produces a plot of the variation accounted for by each extracted factor, as well as a *correlation loading plot* for the first two extracted factors (if the final model has at least two factors). The plot of the variation accounted for can take several forms:

- If the PLS analysis does not include cross validation, then the plot shows the total R square for both model effects and the dependent variables against the number of factors.
- If you specify the CV= option to select the number of factors in the final model by cross validation, then the plot shows the R-square analysis discussed previously as well as the root mean PRESS from the cross validation analysis, with the selected number of factors identified by a vertical line.

The correlation loading plot for the first two factors summarizes many aspects of the two most significant dimensions of the model. It consists of overlaid scatter plots of the scores of the first two factors, the loadings of the model effects, and the loadings of the dependent variables. The loadings are scaled so that the amount of variation in the variables that is explained by the model is proportional to the distance from the origin; circles indicating various levels of explained variation are also overlaid on the correlation loading plot. Also, the correlation between the model approximations for any two variables is proportional to the length of the projection of the point corresponding to one variable on a line through the origin passing through the point corresponding to the other variable; the sign of the correlation corresponds to which side of the origin the projected point falls on.

The R square and the first two correlation loadings are plotted by default when the ODS GRAPHICS are in effect, but you can produce many other plots for the PROC PLS analysis.

ODS Graph Names

PROC PLS assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 66.2.

To request these graphs you must specify the ODS GRAPHICS statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Table 66.2 ODS Graphics Produced by PROC GLM

ODS Graph Name	Plot Description	Option
CorrLoadPlot	Correlation loading plot (default)	PLOT=CORRLOAD(*option*)
CVPlot	Cross validation and R-square analysis (default, as appropriate)	CV=

Table 66.2 *continued*

ODS Graph Name	Plot Description	Option
DModXPlot	Distance of each observation to the X model	PLOT=DMODX
DModXYPlot	Distance of each observation to the X and Y models	PLOT=DMODXY
DModYPlot	Distance of each observation to the Y model	PLOT=DMODY
DiagnosticsPanel	Panel of summary diagnostics for the fit	PLOT=DIAGNOSTICS
AbsResidualByPredicted	Absolute residual by predicted values	PLOT=DIAGNOSTICS(UNPACK)
ObservedByPredicted	Observed by predicted	PLOT=DIAGNOSTICS(UNPACK)
QQPlot	Residual Q-Q plot	PLOT=DIAGNOSTICS(UNPACK)
ResidualByPredicted	Residual by predicted values	PLOT=DIAGNOSTICS(UNPACK)
ResidualHistogram	Residual histogram	PLOT=DIAGNOSTICS(UNPACK)
RFPlot	RF plot	PLOT=DIAGNOSTICS(UNPACK)
ParmProfiles	Profiles of regression coefficients	PLOT=PARMPROFILES
R2Plot	R-square analysis (default, as appropriate)	
ResidualPlots	Residuals for each dependent variable	PLOT=RESIDUALS
VariableImportancePlot	Profile of variable importance factors	PLOT=VIP
XLoadingPlot	Scatter plot matrix of X-loadings against each other	PLOT=XLOADINGPLOT
XLoadingProfiles	Profiles of the X-loadings	PLOT=XLOADINGPROFILES
XScorePlot	Scatter plot matrix of X-scores against each other	PLOT=XSCORES
XWeightPlot	Scatter plot matrix of X-weights against each other	PLOT=XWEIGHTPLOT
XWeightProfiles	Profiles of the X-weights	PLOT=XWEIGHTPROFILES
XYScorePlot	Scatter plot matrix of X-scores against Y-scores	PLOT=XYSCORES
YScorePlot	Scatter plot matrix of Y-scores against each other	PLOT=YSCORES
YWeightPlot	Scatter plot matrix of Y-weights against each other	PLOT=YWEIGHTPLOT

Examples: PLS Procedure

Example 66.1: Examining Model Details

This example, from Umetrics (1995), demonstrates different ways to examine a PLS model. The data come from the field of drug discovery. New drugs are developed from chemicals that are biologically active. Testing a compound for biological activity is an expensive procedure, so it is useful to be able to predict biological activity from cheaper chemical measurements. In fact, computational chemistry makes it possible to calculate certain chemical measurements without even making the compound. These measurements include size, lipophilicity, and polarity at various sites on the molecule. The following statements create a data set named pentaTrain, which contains these data.

```
data pentaTrain;
   input obsnam $ S1 L1 P1 S2 L2 P2
                 S3 L3 P3 S4 L4 P4
                 S5 L5 P5  log_RAI @@;
   n = _n_;
   datalines;
VESSK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
          1.9607 -1.6324  0.5746  1.9607 -1.6324  0.5746
          2.8369  1.4092 -3.1398                   0.00
VESAK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
          1.9607 -1.6324  0.5746  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                   0.28
VEASK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
          0.0744 -1.7333  0.0902  1.9607 -1.6324  0.5746
          2.8369  1.4092 -3.1398                   0.20
VEAAK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
          0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                   0.51
VKAAK    -2.6931 -2.5271 -1.2871  2.8369  1.4092 -3.1398
          0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                   0.11
VEWAK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
         -4.7548  3.6521  0.8524  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                   2.73
VEAAP    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
          0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
         -1.2201  0.8829  2.2253                   0.18
VEHAK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
          2.4064  1.7438  1.1057  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                   1.53
VAAAK    -2.6931 -2.5271 -1.2871  0.0744 -1.7333  0.0902
          0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                  -0.10
GEAAK     2.2261 -5.3648  0.3049  3.0777  0.3891 -0.0701
          0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                  -0.52
```

```
LEAAK    -4.1921 -1.0285 -0.9801  3.0777  0.3891 -0.0701
          0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                  0.40
FEAAK    -4.9217  1.2977  0.4473  3.0777  0.3891 -0.0701
          0.0744 -1.7333  0.0902  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                  0.30
VEGGK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
          2.2261 -5.3648  0.3049  2.2261 -5.3648  0.3049
          2.8369  1.4092 -3.1398                 -1.00
VEFAK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
         -4.9217  1.2977  0.4473  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                  1.57
VELAK    -2.6931 -2.5271 -1.2871  3.0777  0.3891 -0.0701
         -4.1921 -1.0285 -0.9801  0.0744 -1.7333  0.0902
          2.8369  1.4092 -3.1398                  0.59
;
```

You would like to study the relationship between these measurements and the activity of the compound, represented by the logarithm of the relative Bradykinin activating activity (log_RAI). Notice that these data consist of many predictors relative to the number of observations. Partial least squares is especially appropriate in this situation as a useful tool for finding a few underlying predictive factors that account for most of the variation in the response. Typically, the model is fit for part of the data (the "training" or "work" set), and the quality of the fit is judged by how well it predicts the other part of the data (the "test" or "prediction" set). For this example, the first 15 observations serve as the training set and the rest constitute the test set (refer to Ufkes et al. 1978, 1982).

When you fit a PLS model, you hope to find a few PLS factors that explain most of the variation in both predictors and responses. Factors that explain response variation provide good predictive models for new responses, and factors that explain predictor variation are well represented by the observed values of the predictors. The following statements fit a PLS model with two factors and save predicted values, residuals, and other information for each data point in a data set named outpls.

```
proc pls data=pentaTrain;
   model log_RAI = S1-S5 L1-L5 P1-P5;
run;
```

The PLS procedure displays a table, shown in Output 66.1.1, showing how much predictor and response variation is explained by each PLS factor.

Output 66.1.1 Amount of Training Set Variation Explained

The PLS Procedure

Percent Variation Accounted for
by Partial Least Squares Factors

Number of Extracted Factors	Model Effects Current	Model Effects Total	Dependent Variables Current	Dependent Variables Total
1	16.9014	16.9014	89.6399	89.6399
2	12.7721	29.6735	7.8368	97.4767
3	14.6554	44.3289	0.4636	97.9403
4	11.8421	56.1710	0.2485	98.1889
5	10.5894	66.7605	0.1494	98.3383
6	5.1876	71.9481	0.2617	98.6001
7	6.1873	78.1354	0.2428	98.8428
8	7.2252	85.3606	0.1926	99.0354
9	6.7285	92.0891	0.0725	99.1080
10	7.9076	99.9967	0.0000	99.1080
11	0.0033	100.0000	0.0099	99.1179
12	0.0000	100.0000	0.0000	99.1179
13	0.0000	100.0000	0.0000	99.1179
14	0.0000	100.0000	0.0000	99.1179
15	0.0000	100.0000	0.0000	99.1179

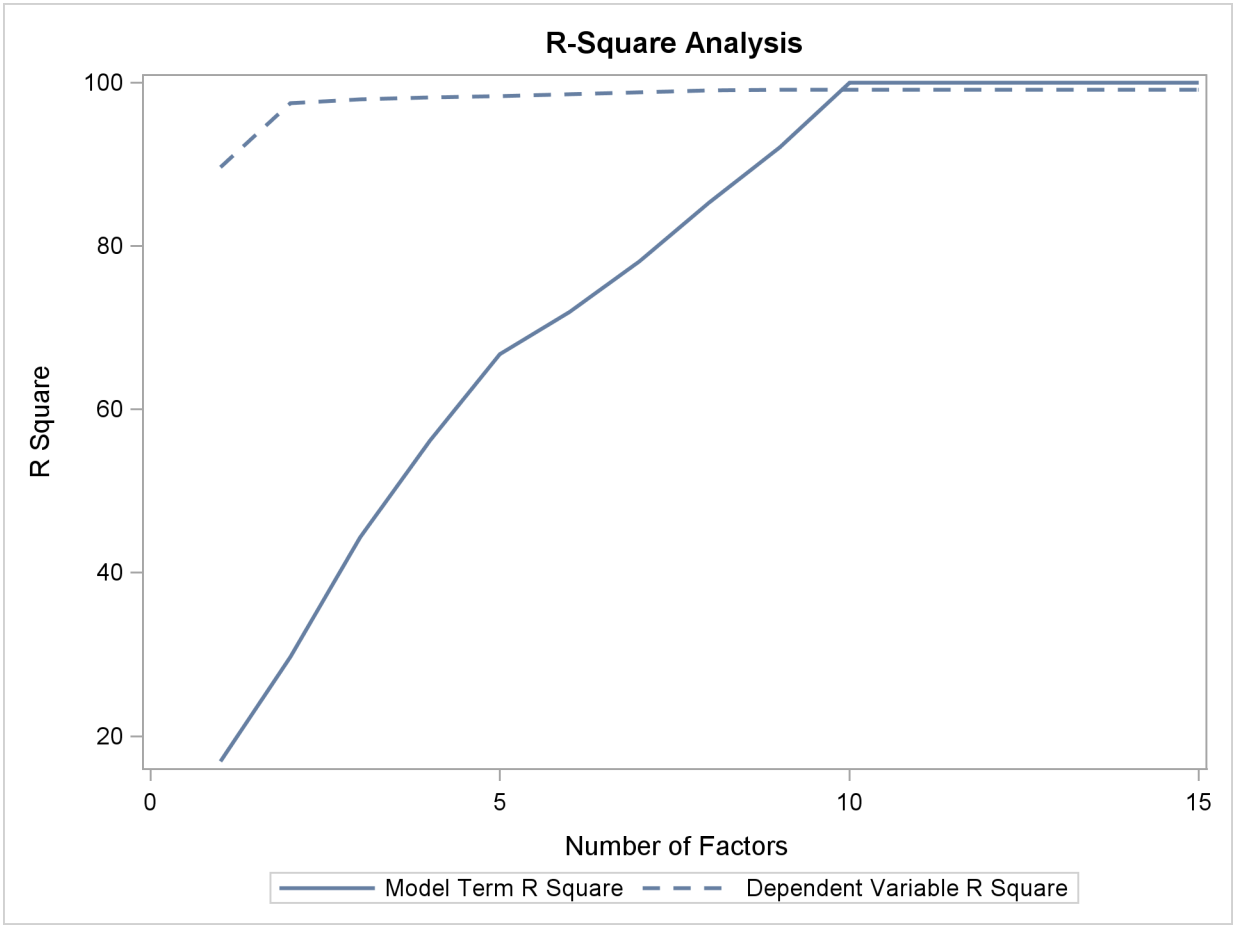
From Output 66.1.1, note that 97% of the response variation is already explained by just two factors, but only 29% of the predictor variation is explained.

The graphics in PROC PLS, available when ODS Graphics is in effect, make it easier to see features of the PLS model.

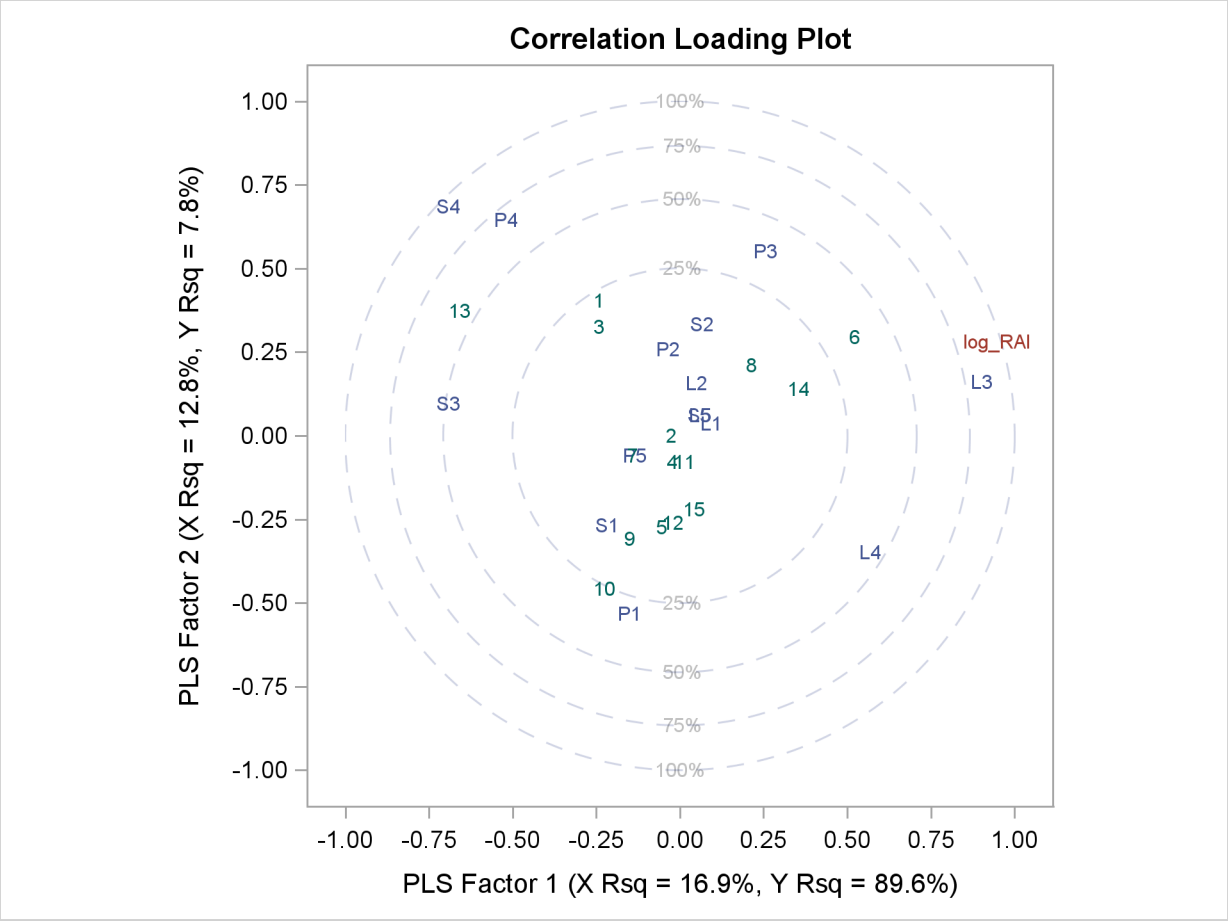
If you enable ODS Graphics, then in addition to the tables discussed previously, PROC PLS displays a graphical depiction of the R-square analysis as well as a correlation loadings plot summarizing the model based on the first two PLS factors. The following statements perform the previous analysis with ODS Graphics enabled, producing Output 66.1.2 and Output 66.1.3.

```
ods graphics on;

proc pls data=pentaTrain;
   model log_RAI = S1-S5 L1-L5 P1-P5;
run;

ods graphics off;
```

Output 66.1.2 Plot of Proportion of Variation Accounted For



Output 66.1.3 Correlation Loadings Plot



The plot in Output 66.1.2 of the proportion of variation explained (or R square) makes it clear that there is a plateau in the response variation after two factors are included in the model. The correlation loading plot in Output 66.1.3 summarizes many features of this two-factor model, including the following:

- The X-scores are plotted as numbers for each observation. You should look for patterns or clearly grouped observations. If you see a curved pattern, for example, you might want to add a quadratic term. Two or more groupings of observations indicate that it might be better to analyze the groups separately, perhaps by including classification effects in the model. This plot appears to show most of the observations close together, with a few being more spread out with larger positive X-scores for factor 2. There are no clear grouping patterns, but observation 13 stands out.
- The loadings show how much variation in each variable is accounted for by the first two factors, jointly by the distance of the corresponding point from the origin and individually by the distance for the projections of this point onto the horizontal and vertical axes. That the dependent variable is well explained by the model is reflected in the fact that the point for log_RAI is near the 100% circle.

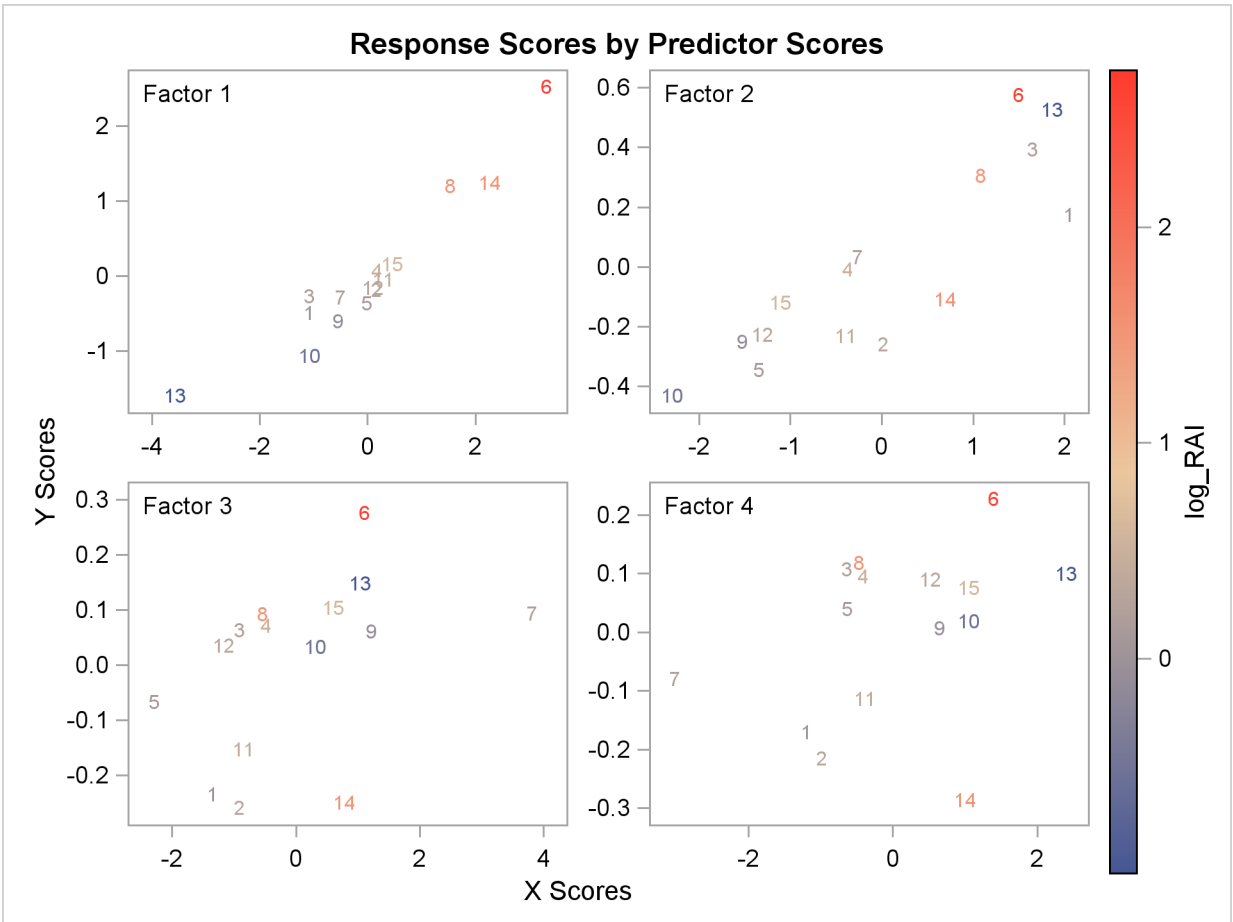
- You can also use the projection interpretation to relate variables to each other. For example, projecting other variables' points onto the line that runs through the log_RAI point and the origin, you can see that the PLS approximation for the predictor L3 is highly positively correlated with log_RAI, S3 is somewhat less correlated but in the negative direction, and several predictors including L1, L5, and S5 have very little correlation with log_RAI.

Other graphics enable you to explore more of the features of the PLS model. For example, you can examine the X-scores versus the Y-scores to explore how partial least squares chooses successive factors. For a good PLS model, the first few factors show a high correlation between the X- and Y-scores. The correlation usually decreases from one factor to the next. When ODS Graphics is in effect, you can plot the X-scores versus the Y-scores by using the PLOT=XYSCORES option, as shown in the following statements.

```
ods graphics on;

proc pls data=pentaTrain nfac=4 plot=XYScores;
   model log_RAI = S1-S5 L1-L5 P1-P5;
run;

ods graphics off;
```

The plot of the X-scores versus the Y-scores for the first four factors is shown in Output 66.1.4.

Output 66.1.4 X-Scores versus Y-Scores



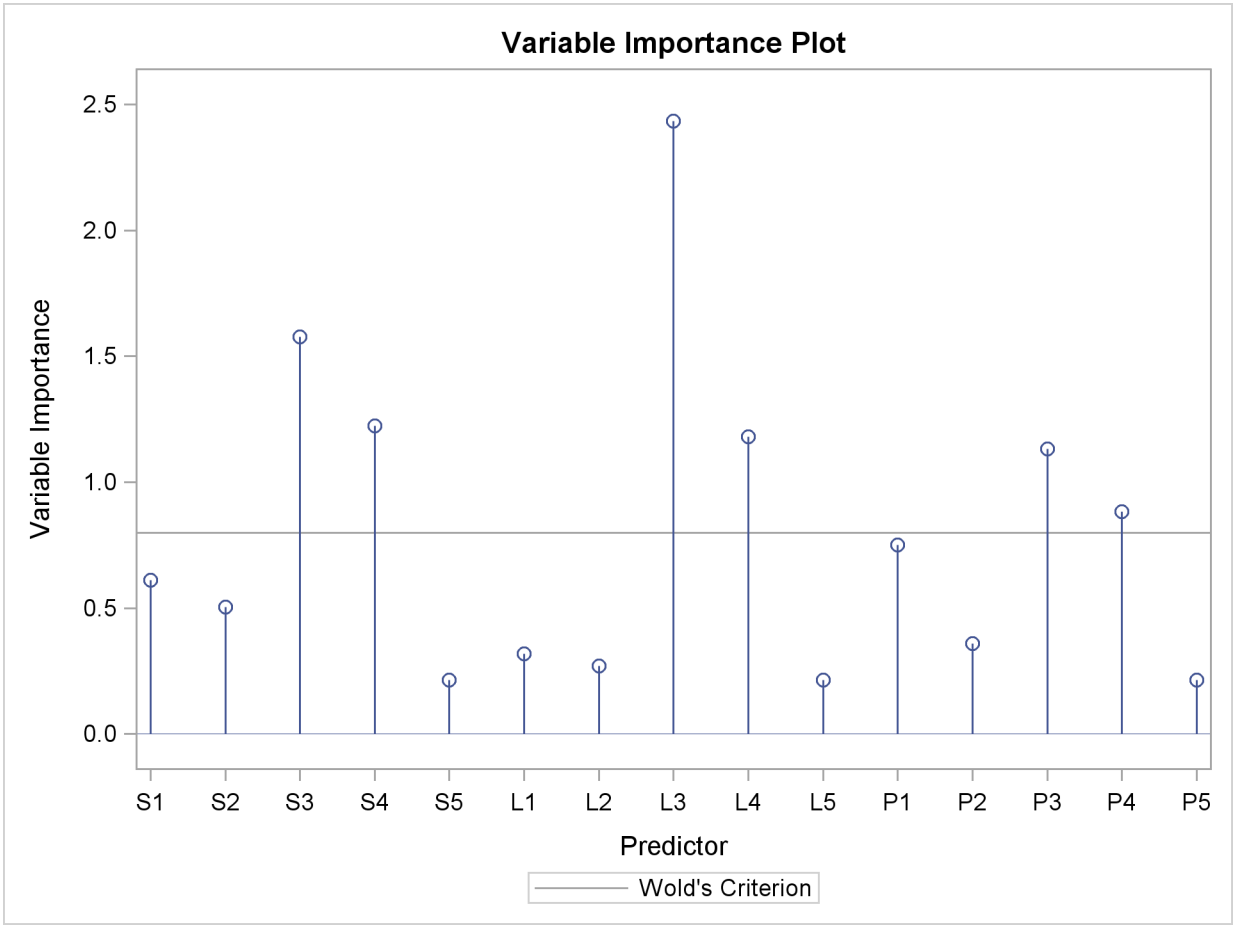
For this example, Output 66.1.4 shows high correlation between X- and Y-scores for the first factor but somewhat lower correlation for the second factor and sharply diminishing correlation after that. This adds strength to the judgment that NFAC=2 is the right number of factors for these data and this model. Note that observation 13 is again extreme in the first two plots. This run might be overly influential for the PLS analysis; thus, you should check to make sure it is reliable.

As explained earlier, you can draw some inferences about the relationship between individual predictors and the dependent variable from the correlation loading plot. However, the regression coefficient profile and the variable importance plot give a more direct indication of which predictors are most useful for predicting the dependent variable. The regression coefficients represent the importance each predictor has in the prediction of just the response. The variable importance plot, on the other hand, represents the contribution of each predictor in fitting the PLS model for both predictors and response. It is based on the *Variable Importance for Projection* (VIP) statistic of Wold (1994), which summarizes the contribution a variable makes to the model. If a predictor has a relatively small coefficient (in absolute value) *and* a small value of VIP, then it is a prime candidate for deletion. Wold in Umetrics (1995) considers a value less than 0.8 to be "small" for the VIP. The following statements fit a two-factor PLS model and display these two additional plots.

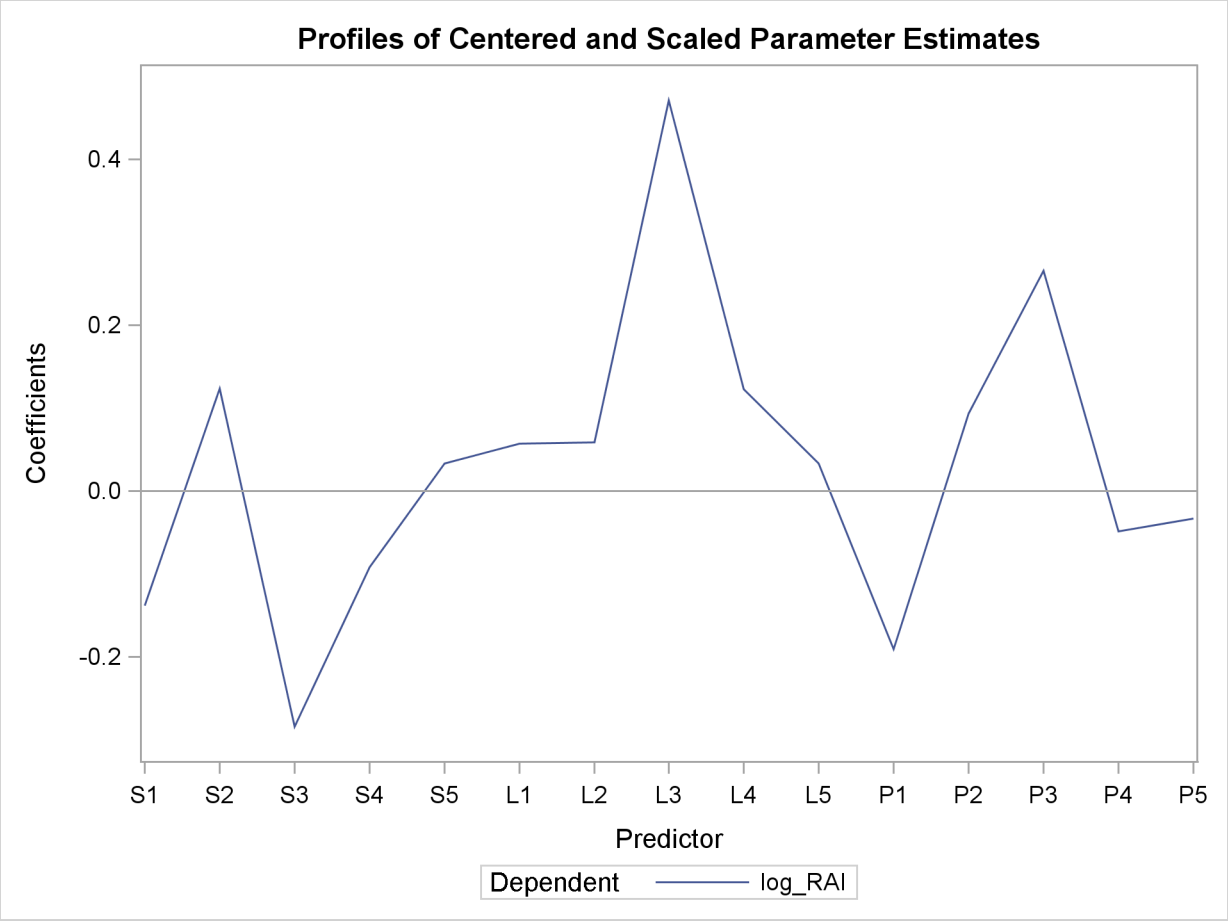
```
ods graphics on;

proc pls data=pentaTrain nfac=2 plot=(ParmProfiles VIP);
   model log_RAI = S1-S5 L1-L5 P1-P5;
run;

ods graphics off;
```

The additional graphics are shown in Output 66.1.5 and Output 66.1.6.

Output 66.1.5 Variable Importance Plots



Output 66.1.6 Regression Parameter Profile



In these two plots, the variables L1, L2, P2, S5, L5, and P5 have small absolute coefficients and small VIP. Looking back at the correlation loadings plot in Output 66.1.2, you can see that these variables tend to be the ones near zero for both PLS factors. You should consider dropping these variables from the model.

Example 66.2: Examining Outliers

This example is a continuation of Example 66.1.

Standard diagnostics for statistical models focus on the response, allowing you to look for patterns that indicate the model is inadequate or for outliers that do not seem to follow the trend of the rest of the data. However, partial least squares effectively models the predictors as well as the responses, so you should consider the pattern of the fit for both. The DModX and DModY statistics give the distance from each point to the PLS model with respect to the predictors and the responses, respectively, and ODS Graphics enables you to plot these values. No point should be dramatically farther from the model than the rest. If there is a group of points that are all farther from the

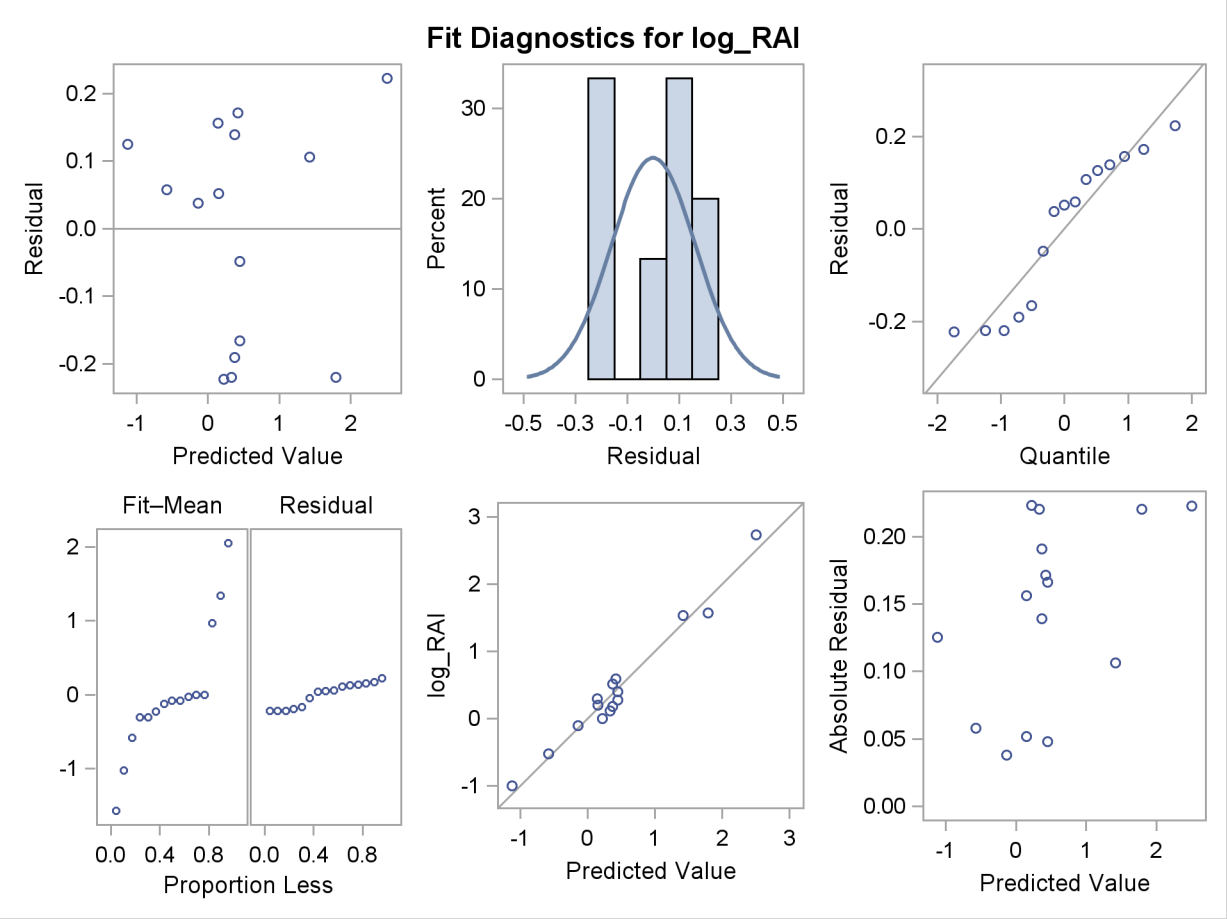
model than the rest, they might have something in common, in which case they should be analyzed separately.

The following statements fit a reduced model to the data discussed in Example 66.1 and plot a panel of standard diagnostics as well as the distances of the observations to the model.

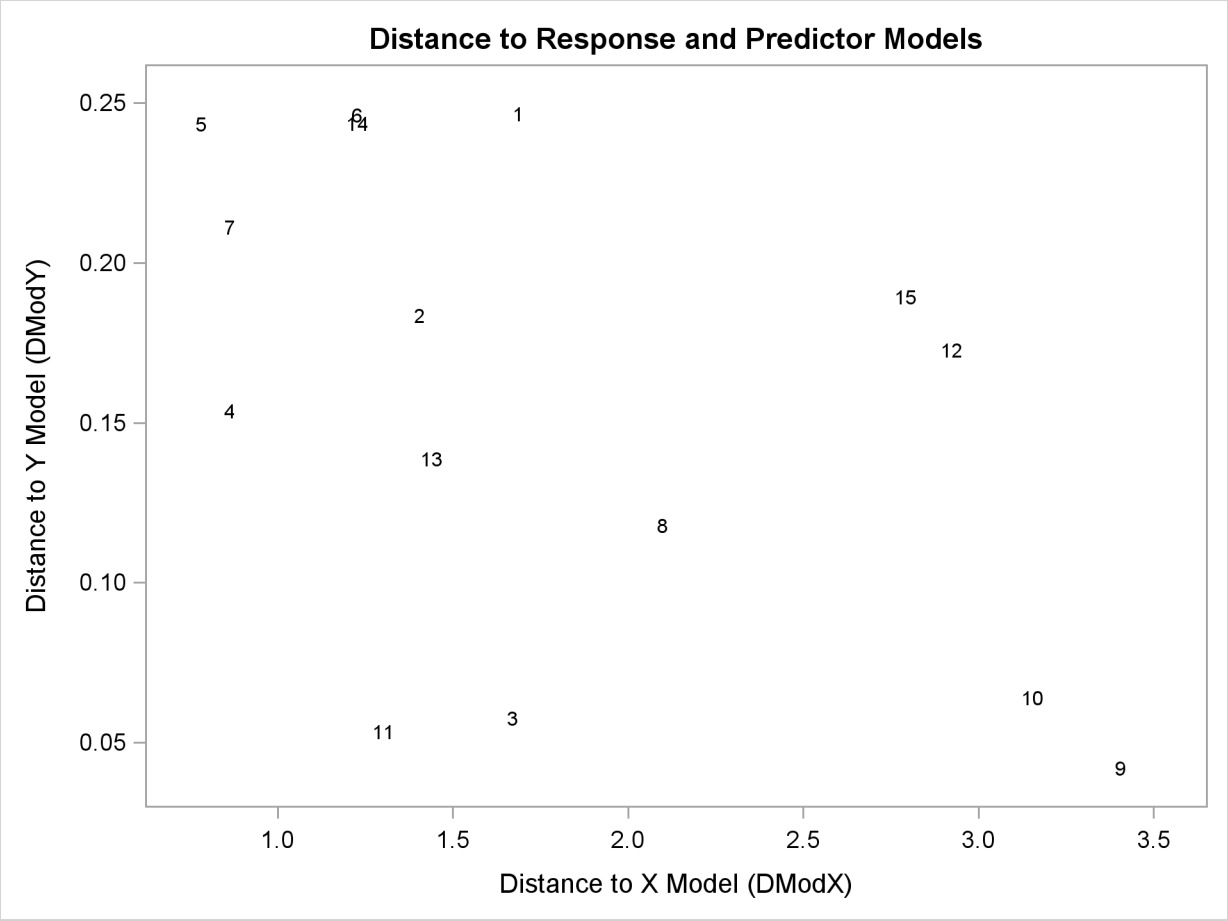
```
ods graphics on;

proc pls data=pentaTrain nfac=2 plot=(diagnostics dmod);
   model log_RAI = S1    P1
                   S2
                   S3 L3 P3
                   S4 L4   ;
run;

ods graphics off;
```

The plots are shown in Output 66.2.1 and Output 66.2.2.

Output 66.2.1 Model Fit Diagnostics



Output 66.2.2 Predictor versus Response Distances to the Model

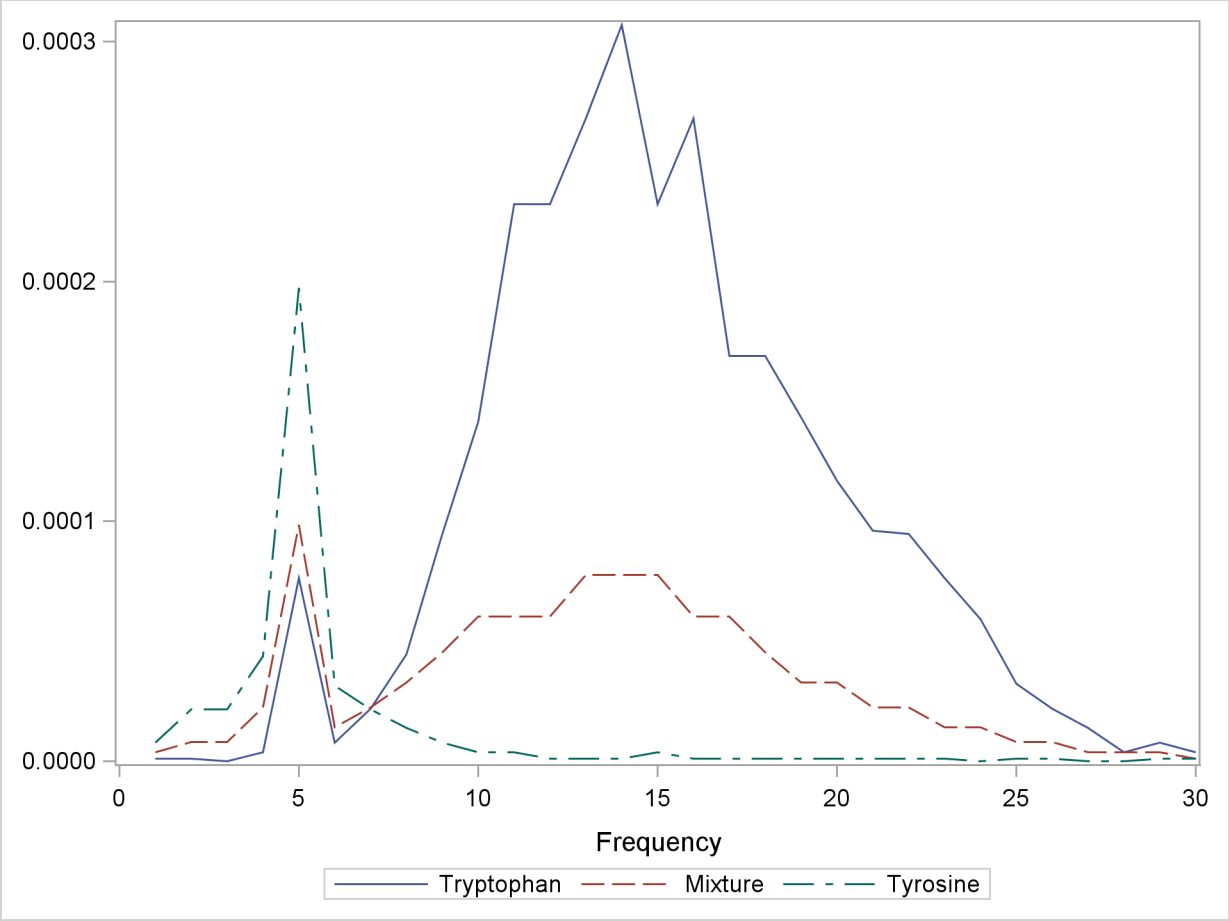


There appear to be no profound outliers in either the predictor space or the response space.

Example 66.3: Choosing a PLS Model by Test Set Validation

This example demonstrates issues in spectrometric calibration. The data (Umetrics 1995) consist of spectrographic readings on 33 samples containing known concentrations of two amino acids, tyrosine and tryptophan. The spectra are measured at 30 frequencies across the overall range of frequencies. For example, Figure 66.3.1 shows the observed spectra for three samples, one with only tryptophan, one with only tyrosine, and one with a mixture of the two, all at a total concentration of 10^{-6}.

Output 66.3.1 Spectra for Three Samples of Tyrosine and Tryptophan



Of the 33 samples, 18 are used as a training set and 15 as a test set. The data originally appear in McAvoy et al. (1989).

These data were created in a lab, with the concentrations fixed in order to provide a wide range of applicability for the model. You want to use a linear function of the logarithms of the spectra to predict the logarithms of tyrosine and tryptophan concentration, as well as the logarithm of the total concentration. Actually, because of the possibility of zeros in both the responses and the predictors, slightly different transformations are used. The following statements create SAS data sets containing the training and test data, named ftrain and ftest, respectively.

```
data ftrain;
   input obsnam $ tot tyr f1-f30 @@;
   try = tot - tyr;
   if (tyr) then tyr_log = log10(tyr); else tyr_log = -8;
   if (try) then try_log = log10(try); else try_log = -8;
   tot_log = log10(tot);
   datalines;
17mix35 0.00003 0
 -6.215 -5.809 -5.114 -3.963 -2.897 -2.269 -1.675 -1.235
 -0.900 -0.659 -0.497 -0.395 -0.335 -0.315 -0.333 -0.377
 -0.453 -0.549 -0.658 -0.797 -0.878 -0.954 -1.060 -1.266
 -1.520 -1.804 -2.044 -2.269 -2.496 -2.714
```

```
19mix35 0.00003 3E-7
 -5.516 -5.294 -4.823 -3.858 -2.827 -2.249 -1.683 -1.218
 -0.907 -0.658 -0.501 -0.400 -0.345 -0.323 -0.342 -0.387
 -0.461 -0.554 -0.665 -0.803 -0.887 -0.960 -1.072 -1.272
 -1.541 -1.814 -2.058 -2.289 -2.496 -2.712
21mix35 0.00003 7.5E-7
 -5.519 -5.294 -4.501 -3.863 -2.827 -2.280 -1.716 -1.262
 -0.939 -0.694 -0.536 -0.444 -0.384 -0.369 -0.377 -0.421
 -0.495 -0.596 -0.706 -0.824 -0.917 -0.988 -1.103 -1.294
 -1.565 -1.841 -2.084 -2.320 -2.521 -2.729

... more lines ...

mix6    0.0001 0.00009
 -1.140 -0.757 -0.497 -0.362 -0.329 -0.412 -0.513 -0.647
 -0.772 -0.877 -0.958 -1.040 -1.104 -1.162 -1.233 -1.317
 -1.425 -1.543 -1.661 -1.804 -1.877 -1.959 -2.034 -2.249
 -2.502 -2.732 -2.964 -3.142 -3.313 -3.576
;
data ftest;
   input obsnam $ tot tyr f1-f30 @@;
   try = tot - tyr;
   if (tyr) then tyr_log = log10(tyr); else tyr_log = -8;
   if (try) then try_log = log10(try); else try_log = -8;
   tot_log = log10(tot);
   datalines;
43trp6  1E-6 0
 -5.915 -5.918 -6.908 -5.428 -4.117 -5.103 -4.660 -4.351
 -4.023 -3.849 -3.634 -3.634 -3.572 -3.513 -3.634 -3.572
 -3.772 -3.772 -3.844 -3.932 -4.017 -4.023 -4.117 -4.227
 -4.492 -4.660 -4.855 -5.428 -5.103 -5.428
59mix6  1E-6 1E-7
 -5.903 -5.903 -5.903 -5.082 -4.213 -5.083 -4.838 -4.639
 -4.474 -4.213 -4.001 -4.098 -4.001 -4.001 -3.907 -4.001
 -4.098 -4.098 -4.206 -4.098 -4.213 -4.213 -4.335 -4.474
 -4.639 -4.838 -4.837 -5.085 -5.410 -5.410
51mix6  1E-6 2.5E-7
 -5.907 -5.907 -5.415 -4.843 -4.213 -4.843 -4.843 -4.483
 -4.343 -4.006 -4.006 -3.912 -3.830 -3.830 -3.755 -3.912
 -4.006 -4.001 -4.213 -4.213 -4.335 -4.483 -4.483 -4.642
 -4.841 -5.088 -5.088 -5.415 -5.415 -5.415

... more lines ...

tyro2   0.0001 0.0001
 -1.081 -0.710 -0.470 -0.337 -0.327 -0.433 -0.602 -0.841
 -1.119 -1.423 -1.750 -2.121 -2.449 -2.818 -3.110 -3.467
 -3.781 -4.029 -4.241 -4.366 -4.501 -4.366 -4.501 -4.501
 -4.668 -4.668 -4.865 -4.865 -5.109 -5.111
;
```

The following statements fit a PLS model with 10 factors.

```
proc pls data=ftrain nfac=10;
   model tot_log tyr_log try_log = f1-f30;
run;
```

The table shown in Output 66.3.2 indicates that only three or four factors are required to explain almost all of the variation in both the predictors and the responses.

Output 66.3.2 Amount of Training Set Variation Explained

The PLS Procedure

Percent Variation Accounted for by Partial Least Squares Factors

Number of Extracted Factors	Model Effects Current	Model Effects Total	Dependent Variables Current	Dependent Variables Total
1	81.1654	81.1654	48.3385	48.3385
2	16.8113	97.9768	32.5465	80.8851
3	1.7639	99.7407	11.4438	92.3289
4	0.1951	99.9357	3.8363	96.1652
5	0.0276	99.9634	1.6880	97.8532
6	0.0132	99.9765	0.7247	98.5779
7	0.0052	99.9817	0.2926	98.8705
8	0.0053	99.9870	0.1252	98.9956
9	0.0049	99.9918	0.1067	99.1023
10	0.0034	99.9952	0.1684	99.2707

In order to choose the optimal number of PLS factors, you can explore how well models based on the training data with different numbers of factors fit the test data. To do so, use the CV=TESTSET option, with an argument pointing to the test data set ftest. The following statements also employ the ODS Graphics features in PROC PLS to display the cross validation results in a plot.

```
ods graphics on;

proc pls data=ftrain nfac=10 cv=testset(ftest)
                             cvtest(stat=press seed=12345);
   model tot_log tyr_log try_log = f1-f30;
run;

ods graphics off;
```

The tabular results of the test set validation are shown in Output 66.3.3, and the graphical results are shown in Output 66.3.4. They indicate that, although five PLS factors give the minimum predicted residual sum of squares, the residuals for four factors are insignificantly different from those for five. Thus, the smaller model is preferred.

Output 66.3.3 Test Set Validation for the Number of PLS Factors

The PLS Procedure

Test Set Validation for the Number of Extracted Factors

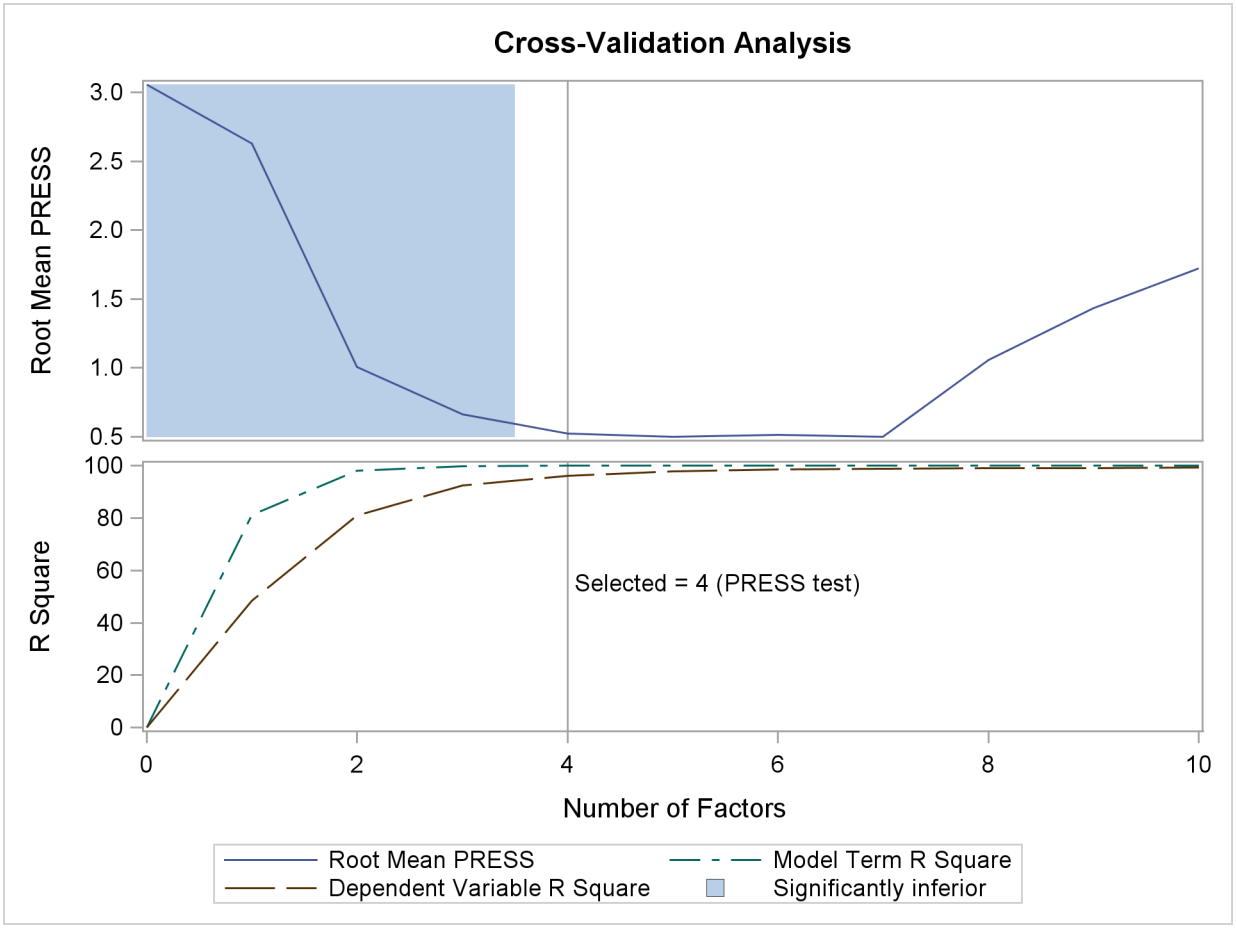
Number of Extracted Factors	Root Mean PRESS	Prob > PRESS
0	3.056797	<.0001
1	2.630561	<.0001
2	1.00706	0.0070
3	0.664603	0.0020
4	0.521578	0.3800
5	0.500034	1.0000
6	0.513561	0.5100
7	0.501431	0.6870
8	1.055791	0.1530
9	1.435085	0.1010
10	1.720389	0.0320

Minimum root mean PRESS	0.5000
Minimizing number of factors	5
Smallest number of factors with p > 0.1	4

Percent Variation Accounted for by Partial Least Squares Factors

Number of Extracted Factors	Model Effects		Dependent Variables	
	Current	Total	Current	Total
1	81.1654	81.1654	48.3385	48.3385
2	16.8113	97.9768	32.5465	80.8851
3	1.7639	99.7407	11.4438	92.3289
4	0.1951	99.9357	3.8363	96.1652

Output 66.3.4 Test Set Validation Plot

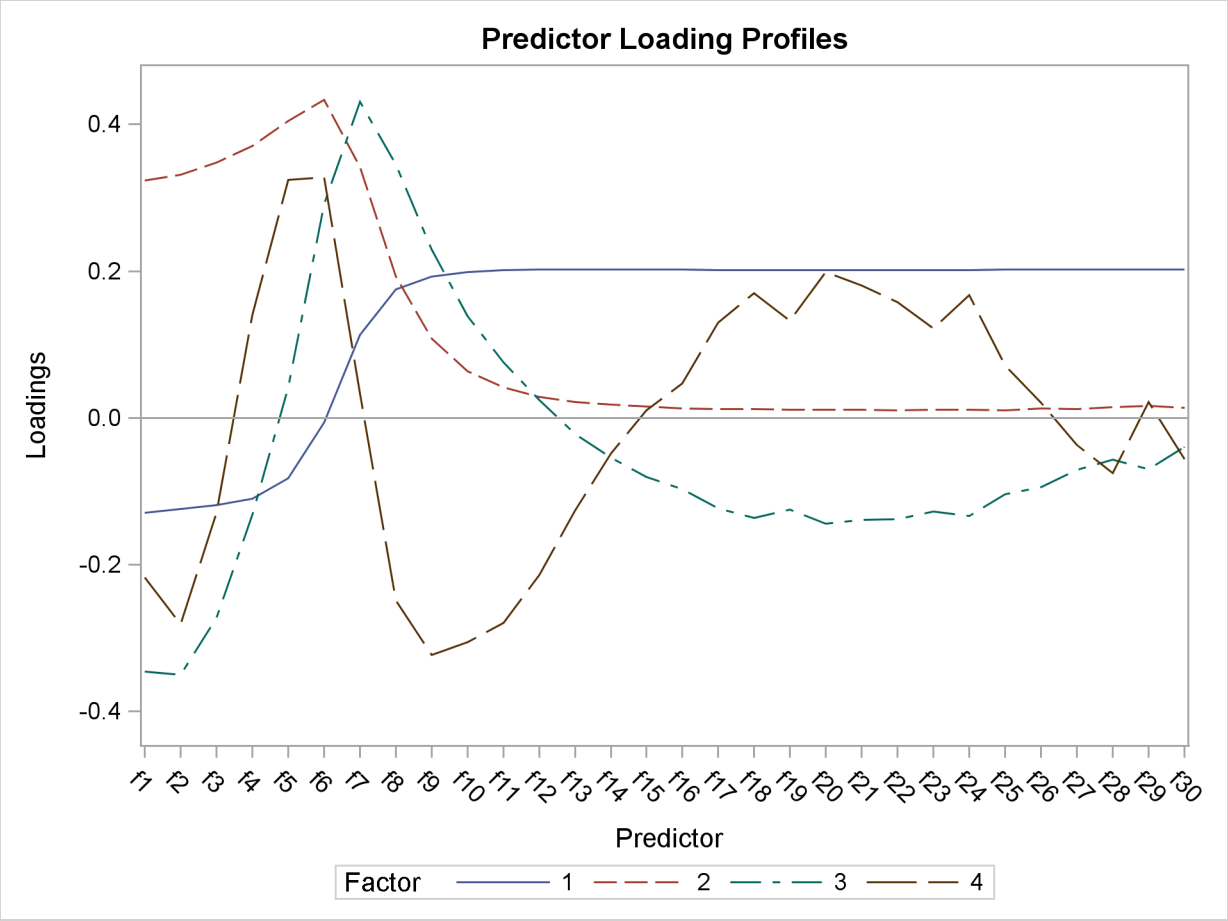


The factor loadings show how the PLS factors are constructed from the centered and scaled predictors. For spectral calibration, it is useful to plot the loadings against the frequency. In many cases, the physical meanings that can be attached to factor loadings help to validate the scientific interpretation of the PLS model. You can use ODS Graphics with PROC PLS to plot the loadings for the four PLS factors against frequency, as shown in the following statements.

```
ods graphics on;

proc pls data=ftrain nfac=4 plot=XLoadingProfiles;
   model tot_log tyr_log try_log = f1-f30;
run;

ods graphics off;
```

The resulting plot is shown in Output 66.3.5.

Output 66.3.5 Predictor Loadings across Frequencies



Notice that all four factors handle frequencies below and above about 7 or 8 differently. For example, the first factor is very nearly a simple contrast between the averages of the two sets of frequencies, and the second factor appears to be approximately a weighted sum of only the frequencies in the first set.

References

de Jong, S. (1993), "SIMPLS: An Alternative Approach to Partial Least Squares Regression," *Chemometrics and Intelligent Laboratory Systems*, 18, 251–263.

de Jong, S. and Kiers, H. (1992), "Principal Covariates Regression," *Chemometrics and Intelligent Laboratory Systems*, 14, 155–164.

Dijkstra, T. (1983), "Some Comments on Maximum Likelihood and Partial Least Squares Methods," *Journal of Econometrics*, 22, 67–90.

Dijkstra, T. (1985), *Latent Variables in Linear Stochastic Models: Reflections on Maximum Likelihood and Partial Least Squares Methods.*, Second Edition, Amsterdam, The Netherlands: Sociometric Research Foundation.

Frank, I. and Friedman, J. (1993), "A Statistical View of Some Chemometrics Regression Tools," *Technometrics*, 35, 109–135.

Geladi, P. and Kowalski, B. (1986), "Partial Least-Squares Regression: A Tutorial," *Analytica Chimica Acta*, 185, 1–17.

Haykin, S. (1994), *Neural Networks: A Comprehensive Foundation*, New York: Macmillan.

Helland, I. (1988), "On the Structure of Partial Least Squares Regression," *Communications in Statistics, Simulation and Computation*, 17, 581–607.

Hoerl, A. and Kennard, R. (1970), "Ridge Regression: Biased Estimation for Non-orthogonal Problems," *Technometrics*, 12, 55–67.

Lindberg, W., Persson, J.-A., and Wold, S. (1983), "Partial Least-Squares Method for Spectrofluorimetric Analysis of Mixtures of Humic Acid and Ligninsulfonate," *Analytical Chemistry*, 55, 643–648.

McAvoy, T. J., Wang, N. S., Naidu, S., Bhat, N., Gunter, J., and Simmons, M. (1989), "Interpreting Biosensor Data via Backpropagation," *International Joint Conference on Neural Networks*, 1, 227–233.

Naes, T. and Martens, H. (1985), "Comparison of Prediction Methods for Multicollinear Data," *Communications in Statistics, Simulation and Computation*, 14, 545–576.

Ränner, S., Lindgren, F., Geladi, P., and Wold, S. (1994), "A PLS Kernel Algorithm for Data Sets with Many Variables and Fewer Objects," *Journal of Chemometrics*, 8, 111–125.

Sarle, W. S. (1994), "Neural Networks and Statistical Models," in *Proceedings of the Nineteenth Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Shao, J. (1993), "Linear Model Selection by Cross-Validation," *Journal of the American Statistical Association*.

Tobias, R. (1995), "An Introduction to Partial Least Squares Regression," in *Proceedings of the Twentieth Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Ufkes, J. G. R., Visser, B. J., Heuver, G., and Van Der Meer, C. (1978), "Structure-Activity Relationships of Bradykinin-Potentiating Peptides," *European Journal of Pharmacology*, 50, 119.

Ufkes, J. G. R., Visser, B. J., Heuver, G., Wynne, H. J., and Van Der Meer, C. (1982), "Further Studies on the Structure-Activity Relationships of Bradykinin-Potentiating Peptides," *European Journal of Pharmacology*, 79, 155.

Umetrics (1995), *Multivariate Analysis (3-day course)*, Winchester, MA.

van den Wollenberg, A. L. (1977), "Redundancy Analysis—An Alternative to Canonical Correlation Analysis," *Psychometrika*, 42, 207–219.

van der Voet, H. (1994), "Comparing the Predictive Accuracy of Models Using a Simple Randomization Test," *Chemometrics and Intelligent Laboratory Systems*, 25, 313–323.

Wold, H. (1966), "Estimation of Principal Components and Related Models by Iterative Least Squares," in P. R. Krishnaiah, ed., *Multivariate Analysis*, New York: Academic Press.

Wold, S. (1994), "PLS for Multivariate Linear Modeling," *QSAR: Chemometric Methods in Molecular Design. Methods and Principles in Medicinal Chemistry*.

Chapter 67

The POWER Procedure

Contents

Overview: POWER Procedure **4816**
Getting Started: POWER Procedure **4818**
Computing Power for a One-Sample *t* Test 4818
Determining Required Sample Size for a Two-Sample *t* Test 4821
Syntax: POWER Procedure **4826**
PROC POWER Statement 4827
LOGISTIC Statement 4827
MULTREG Statement 4835
ONECORR Statement 4840
ONESAMPLEFREQ Statement 4844
ONESAMPLEMEANS Statement 4852
ONEWAYANOVA Statement 4859
PAIREDFREQ Statement 4864
PAIREDMEANS Statement 4871
PLOT Statement 4880
TWOSAMPLEFREQ Statement 4884
TWOSAMPLEMEANS Statement 4891
TWOSAMPLESURVIVAL Statement 4901
TWOSAMPLEWILCOXON Statement 4911
Details: POWER Procedure **4916**
Overview of Power Concepts 4916
Summary of Analyses 4916
Specifying Value Lists in Analysis Statements 4919
Keyword-Lists 4919
Number-Lists 4919
Grouped-Number-Lists 4919
Name-Lists 4921
Grouped-Name-Lists 4921
Sample Size Adjustment Options 4922
Error and Information Output 4922
Displayed Output 4924
ODS Table Names 4925
Computational Resources 4926
Memory 4926

CPU Time 4926
Computational Methods and Formulas 4926
Common Notation 4926
Analyses in the LOGISTIC Statement 4928
Analyses in the MULTREG Statement 4930
Analyses in the ONECORR Statement 4932
Analyses in the ONESAMPLEFREQ Statement 4934
Analyses in the ONESAMPLEMEANS Statement 4953
Analyses in the ONEWAYANOVA Statement 4957
Analyses in the PAIREDFREQ Statement 4958
Analyses in the PAIREDMEANS Statement 4962
Analyses in the TWOSAMPLEFREQ Statement 4966
Analyses in the TWOSAMPLEMEANS Statement 4969
Analyses in the TWOSAMPLESURVIVAL Statement 4975
Analyses in the TWOSAMPLEWILCOXON Statement 4978
Examples: POWER Procedure **4981**
Example 67.1: One-Way ANOVA 4981
Example 67.2: The Sawtooth Power Function in Proportion Analyses 4986
Example 67.3: Simple AB/BA Crossover Designs 4995
Example 67.4: Noninferiority Test with Lognormal Data 4999
Example 67.5: Multiple Regression and Correlation 5002
Example 67.6: Comparing Two Survival Curves 5007
Example 67.7: Confidence Interval Precision 5009
Example 67.8: Customizing Plots 5012
Assigning Analysis Parameters to Axes 5014
Fine-Tuning a Sample Size Axis 5019
Adding Reference Lines 5023
Linking Plot Features to Analysis Parameters 5026
Choosing Key (Legend) Styles 5031
Modifying Symbol Locations 5035
Example 67.9: Binary Logistic Regression with Independent Predictors . . . 5037
Example 67.10: Wilcoxon-Mann-Whitney Test 5039
References **5042**

Overview: POWER Procedure

Power and sample size analysis optimizes the resource usage and design of a study, improving chances of conclusive results with maximum efficiency. The POWER procedure performs prospective power and sample size analyses for a variety of goals, such as the following:

- determining the sample size required to get a significant result with adequate probability (power)
- characterizing the power of a study to detect a meaningful effect
- conducting what-if analyses to assess sensitivity of the power or required sample size to other factors

Here *prospective* indicates that the analysis pertains to planning for a future study. This is in contrast to *retrospective* power analysis for a past study, which is not supported by the procedure.

A variety of statistical analyses are covered:

- t tests, equivalence tests, and confidence intervals for means
- tests, equivalence tests, and confidence intervals for binomial proportions
- multiple regression
- tests of correlation and partial correlation
- one-way analysis of variance
- rank tests for comparing two survival curves
- logistic regression with binary response
- Wilcoxon-Mann-Whitney (rank-sum) test

For more complex linear models, see Chapter 41, "The GLMPOWER Procedure."

Input for PROC POWER includes the components considered in study planning:

- design
- statistical model and test
- significance level (alpha)
- surmised effects and variability
- power
- sample size

You designate one of these components by a missing value in the input, in order to identify it as the result parameter. The procedure calculates this result value over one or more scenarios of input values for all other components. Power and sample size are the most common result values, but for some analyses the result can be something else. For example, you can solve for the sample size of a single group for a two-sample t test.

In addition to tabular results, PROC POWER produces graphs. You can produce the most common types of plots easily with default settings and use a variety of options for more customized graphics.

For example, you can control the choice of axis variables, axis ranges, number of plotted points, mapping of graphical features (such as color, line style, symbol and panel) to analysis parameters, and legend appearance.

The POWER procedure is one of several tools available in SAS/STAT software for power and sample size analysis. PROC GLMPOWER supports more complex linear models. The Power and Sample Size application provides a user interface and implements many of the analyses supported in the procedures. See Chapter 41, "The GLMPOWER Procedure," and Chapter 68, "The Power and Sample Size Application," for details.

The following sections of this chapter describe how to use PROC POWER and discuss the underlying statistical methodology. The section "Getting Started: POWER Procedure" on page 4818 introduces PROC POWER with simple examples of power computation for a one-sample t test and sample size determination for a two-sample t test. The section "Syntax: POWER Procedure" on page 4826 describes the syntax of the procedure. The section "Details: POWER Procedure" on page 4916 summarizes the methods employed by PROC POWER and provides details on several special topics. The section "Examples: POWER Procedure" on page 4981 illustrates the use of the POWER procedure with several applications.

For an overview of methodology and SAS tools for power and sample size analysis, see Chapter 19, "Introduction to Power and Sample Size Analysis." For more discussion and examples, see O'Brien and Castelloe (2007), Castelloe (2000), Castelloe and O'Brien (2001), Muller and Benignus (1992), O'Brien and Muller (1993), and Lenth (2001).

Getting Started: POWER Procedure

Computing Power for a One-Sample *t* Test

Suppose you want to improve the accuracy of a machine used to print logos on sports jerseys. The logo placement has an inherently high variability, but the horizontal alignment of the machine can be adjusted. The operator agrees to pay for a costly adjustment if you can establish a nonzero mean horizontal displacement in either direction with high confidence. You have 150 jerseys at your disposal to measure, and you want to determine your chances of a significant result (power) by using a one-sample t test with a two-sided $\alpha = 0.05$.

You decide that 8 mm is the smallest displacement worth addressing. Hence, you will assume a true mean of 8 in the power computation. Experience indicates that the standard deviation is about 40.

Use the ONESAMPLEMEANS statement in the POWER procedure to compute the power. Indicate power as the result parameter by specifying the POWER= option with a missing value (.). Specify your conjectures for the mean and standard deviation by using the MEAN= and STDDEV= options and for the sample size by using the NTOTAL= option. The statements required to perform this analysis are as follows:

```
proc power;
   onesamplemeans
      mean   = 8
      ntotal = 150
      stddev = 40
      power  = .;
run;
```

Default values for the TEST=, DIST=, ALPHA=, NULLMEAN=, and SIDES= options specify a two-sided t test for a mean of 0, assuming a normal distribution with a significance level of $\alpha = 0.05$.

Figure 67.1 shows the output.

Figure 67.1 Sample Size Analysis for One-Sample *t* Test

The POWER Procedure
One-sample t Test for Mean

Fixed Scenario Elements

Distribution	Normal
Method	Exact
Mean	8
Standard Deviation	40
Total Sample Size	150
Number of Sides	2
Null Mean	0
Alpha	0.05

Computed Power

Power
0.682

The power is about 0.68. In other words, there is about a 2/3 chance that the t test will produce a significant result demonstrating the machine's average off-center displacement. This probability depends on the assumptions for the mean and standard deviation.

Now, suppose you want to account for some of your uncertainty in conjecturing the true mean and standard deviation by evaluating the power for four scenarios, using reasonable low and high values, 5 and 10 for the mean, and 30 and 50 for the standard deviation. Also, you might be able to measure more than 150 jerseys, and you would like to know under what circumstances you could get by with fewer. You want to plot power for sample sizes between 100 and 200 to visualize how sensitive the power is to changes in sample size for these four scenarios of means and standard deviations. The following statements perform this analysis:

```
proc power;
   onesamplemeans
      mean   = 5 10
      ntotal = 150
      stddev =  30 50
      power  = .;
   plot x=n min=100 max=200;
run;
```

The new mean and standard deviation values are specified by using the MEAN= and STDDEV= options in the ONESAMPLEMEANS statement. The PLOT statement with X=N produces a plot with sample size on the X axis. (The result parameter, in this case the power, is always plotted on the other axis.) The MIN= and MAX= options in the PLOT statement determine the sample size range.

Figure 67.2 shows the output, and Figure 67.3 shows the plot.

Figure 67.2 Sample Size Analysis for One-Sample *t* Test with Input Ranges

The POWER Procedure
One-sample t Test for Mean

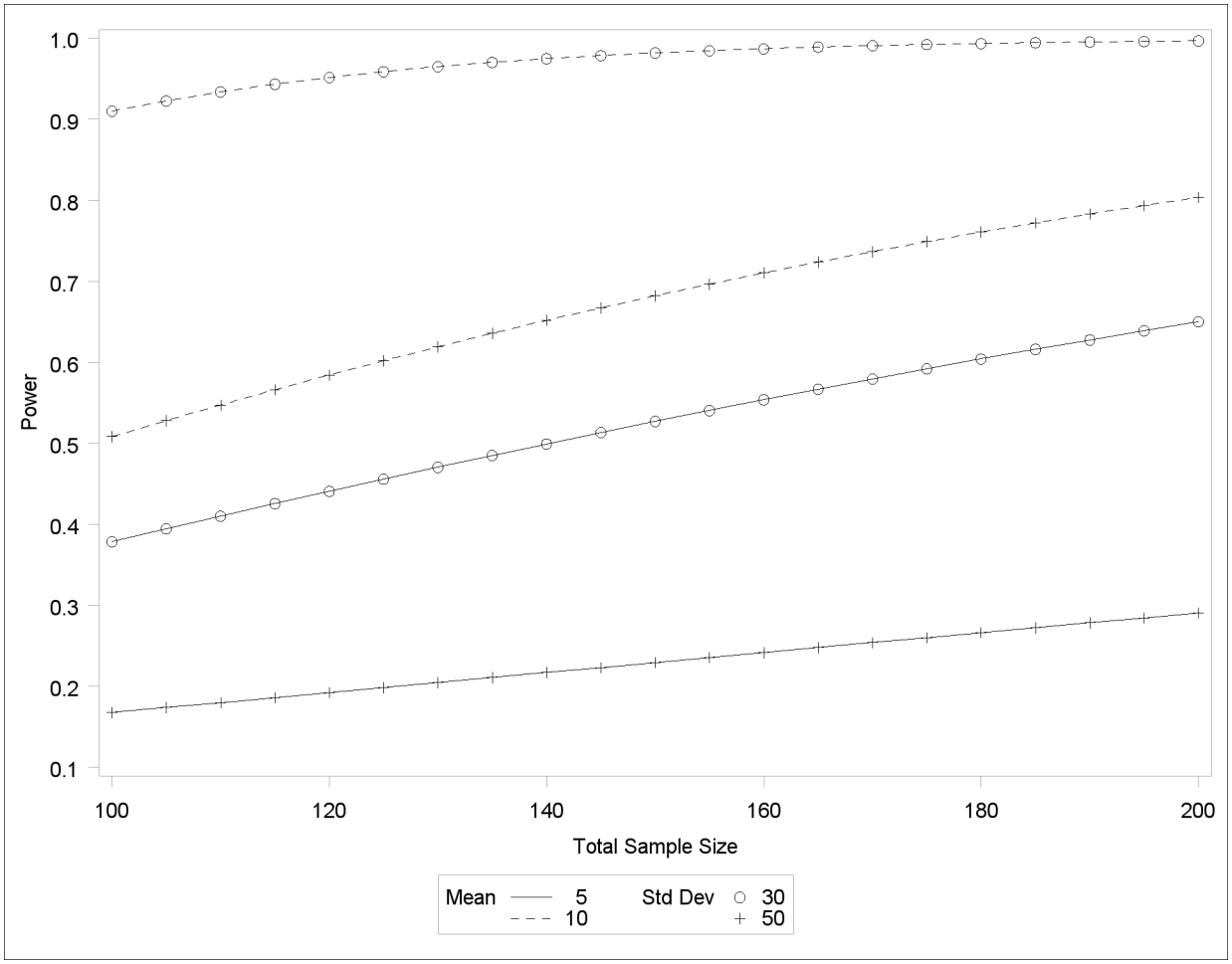
Fixed Scenario Elements

Distribution	Normal
Method	Exact
Total Sample Size	150
Number of Sides	2
Null Mean	0
Alpha	0.05

Computed Power

Index	Mean	Std Dev	Power
1	5	30	0.527
2	5	50	0.229
3	10	30	0.982
4	10	50	0.682

Figure 67.3 Plot of Power versus Sample Size for One-Sample t Test with Input Ranges



The power ranges from about 0.23 to 0.98 for a sample size of 150 depending on the mean and standard deviation. In Figure 67.3, the line style identifies the mean, and the plotting symbol identifies the standard deviation. The locations of plotting symbols indicate computed powers; the curves are linear interpolations of these points. The plot suggests sufficient power for a mean of 10 and standard deviation of 30 (for any of the sample sizes) but insufficient power for the other three scenarios.

Determining Required Sample Size for a Two-Sample *t* Test

In this example you want to compare two physical therapy treatments designed to increase muscle flexibility. You need to determine the number of patients required to achieve a power of at least 0.9 to detect a group mean difference in a two-sample t test. You will use $\alpha = 0.05$ (two-tailed).

The mean flexibility with the standard treatment (as measured on a scale of 1 to 20) is well known to be about 13 and is thought to be between 14 and 15 with the new treatment. You conjecture three alternative scenarios for the means:

1. $\mu_1 = 13, \mu_2 = 14$
2. $\mu_1 = 13, \mu_2 = 14.5$
3. $\mu_1 = 13, \mu_2 = 15$

You conjecture two scenarios for the common group standard deviation:

1. $\sigma = 1.2$
2. $\sigma = 1.7$

You also want to try three weighting schemes:

1. equal group sizes (balanced, or 1:1)
2. twice as many patients with the new treatment (1:2)
3. three times as many patients with the new treatment (1:3)

This makes $3 \times 2 \times 3 = 18$ scenarios in all.

Use the TWOSAMPLEMEANS statement in the POWER procedure to determine the sample sizes required to give 90% power for each of these 18 scenarios. Indicate total sample size as the result parameter by specifying the NTOTAL= option with a missing value (.). Specify your conjectures for the means by using the GROUPMEANS= option. Using the "matched" notation (discussed in the section "Specifying Value Lists in Analysis Statements" on page 4919), enclose the two group means for each scenario in parentheses. Use the STDDEV= option to specify scenarios for the common standard deviation. Specify the weighting schemes by using the GROUPWEIGHTS= option. You could again use the matched notation. But for illustrative purposes, specify the scenarios for each group weight separately by using the "crossed" notation, with scenarios for each group weight separated by a vertical bar (|). The statements that perform the analysis are as follows:

```
proc power;
   twosamplemeans
      groupmeans   = (13 14) (13 14.5) (13 15)
      stddev       = 1.2 1.7
      groupweights = 1 | 1 2 3
      power        = 0.9
      ntotal       = .;
run;
```

Default values for the TEST=, DIST=, NULLDIFF=, ALPHA=, and SIDES= options specify a two-sided t test of group mean difference equal to 0, assuming a normal distribution with a significance level of $\alpha = 0.05$. The results are shown in Figure 67.4.

Figure 67.4 Sample Size Analysis for Two-Sample *t* Test Using Group Means

The POWER Procedure
Two-sample t Test for Mean Difference

Fixed Scenario Elements

Distribution	Normal
Method	Exact
Group 1 Weight	1
Nominal Power	0.9
Number of Sides	2
Null Difference	0
Alpha	0.05

Computed N Total

Index	Mean1	Mean2	Std Dev	Weight2	Actual Power	N Total
1	13	14.0	1.2	1	0.907	64
2	13	14.0	1.2	2	0.908	72
3	13	14.0	1.2	3	0.905	84
4	13	14.0	1.7	1	0.901	124
5	13	14.0	1.7	2	0.905	141
6	13	14.0	1.7	3	0.900	164
7	13	14.5	1.2	1	0.910	30
8	13	14.5	1.2	2	0.906	33
9	13	14.5	1.2	3	0.916	40
10	13	14.5	1.7	1	0.900	56
11	13	14.5	1.7	2	0.901	63
12	13	14.5	1.7	3	0.908	76
13	13	15.0	1.2	1	0.913	18
14	13	15.0	1.2	2	0.927	21
15	13	15.0	1.2	3	0.922	24
16	13	15.0	1.7	1	0.914	34
17	13	15.0	1.7	2	0.921	39
18	13	15.0	1.7	3	0.910	44

The interpretation is that in the best-case scenario (large mean difference of 2, small standard deviation of 1.2, and balanced design), a sample size of $N = 18$ ($n_1 = n_2 = 9$) patients is sufficient to achieve a power of at least 0.9. In the worst-case scenario (small mean difference of 1, large standard deviation of 1.7, and a 1:3 unbalanced design), a sample size of $N = 164$ ($n_1 = 41, n_2 = 123$) patients is necessary. The Nominal Power of 0.9 in the "Fixed Scenario Elements" table represents the input target power, and the Actual Power column in the "Computed N Total" table is the power at the sample size (N Total) adjusted to achieve the specified sample weighting exactly.

Note the following characteristics of the analysis, and ways you can modify them if you want:

- The total sample sizes are rounded up to multiples of the weight sums (2 for the 1:1 design, 3 for the 1:2 design, and 4 for the 1:3 design) to ensure that each group size is an integer. To request raw fractional sample size solutions, use the NFRACTIONAL option.
- Only the group weight that varies (the one for group 2) is displayed as an output column,

while the weight for group 1 appears in the "Fixed Scenario Elements" table. To display the group weights together in output columns, use the matched version of the value list rather than the crossed version.

- If you can specify only differences between group means (instead of their individual values), or if you want to display the mean differences instead of the individual means, use the MEANDIFF= option instead of the GROUPMEANS= option.

The following statements implement all of these modifications:

```
proc power;
   twosamplemeans
      nfractional
      meandiff     = 1 to 2 by 0.5
      stddev       = 1.2 1.7
      groupweights = (1 1) (1 2) (1 3)
      power        = 0.9
      ntotal       = .;
run;
```

Figure 67.5 shows the new results.

Figure 67.5 Sample Size Analysis for Two-Sample *t* Test Using Mean Differences

The POWER Procedure
Two-sample t Test for Mean Difference

Fixed Scenario Elements

Distribution	Normal
Method	Exact
Nominal Power	0.9
Number of Sides	2
Null Difference	0
Alpha	0.05

Figure 67.5 *continued*

Computed Ceiling N Total

Index	Mean Diff	Std Dev	Weight1	Weight2	Fractional N Total	Actual Power	Ceiling N Total
1	1.0	1.2	1	1	62.507429	0.902	63
2	1.0	1.2	1	2	70.065711	0.904	71
3	1.0	1.2	1	3	82.665772	0.901	83
4	1.0	1.7	1	1	123.418482	0.901	124
5	1.0	1.7	1	2	138.598159	0.901	139
6	1.0	1.7	1	3	163.899094	0.900	164
7	1.5	1.2	1	1	28.961958	0.900	29
8	1.5	1.2	1	2	32.308867	0.906	33
9	1.5	1.2	1	3	37.893351	0.901	38
10	1.5	1.7	1	1	55.977156	0.900	56
11	1.5	1.7	1	2	62.717357	0.901	63
12	1.5	1.7	1	3	73.954291	0.900	74
13	2.0	1.2	1	1	17.298518	0.913	18
14	2.0	1.2	1	2	19.163836	0.913	20
15	2.0	1.2	1	3	22.282926	0.910	23
16	2.0	1.7	1	1	32.413512	0.905	33
17	2.0	1.7	1	2	36.195531	0.907	37
18	2.0	1.7	1	3	42.504535	0.903	43

Note that the Nominal Power of 0.9 applies to the raw computed sample size (Fractional N Total), and the Actual Power column applies to the rounded sample size (Ceiling N Total). Some of the adjusted sample sizes in Figure 67.5 are lower than those in Figure 67.4 because underlying group sample sizes are allowed to be fractional (for example, the first Ceiling N Total of 63 corresponding to equal group sizes of 31.5).

Syntax: POWER Procedure

The following statements are available in PROC POWER:

PROC POWER < *options* > ;
LOGISTIC < *options* > ;
MULTREG < *options* > ;
ONECORR < *options* > ;
ONESAMPLEFREQ < *options* > ;
ONESAMPLEMEANS < *options* > ;
ONEWAYANOVA < *options* > ;
PAIREDFREQ < *options* > ;
PAIREDMEANS < *options* > ;
PLOT < *plot-options* > < / *graph-options* > ;
TWOSAMPLEFREQ < *options* > ;
TWOSAMPLEMEANS < *options* > ;
TWOSAMPLESURVIVAL < *options* > ;
TWOSAMPLEWILCOXON < *options* > ;

The statements in the POWER procedure consist of the PROC POWER statement, a set of *analysis statements* (for requesting specific power and sample size analyses), and the PLOT statement (for producing graphs). The PROC POWER statement and at least one of the analysis statements are required. The analysis statements are LOGISTIC, MULTREG, ONECORR, ONESAMPLEFREQ, ONESAMPLEMEANS, ONEWAYANOVA, PAIREDFREQ, PAIREDMEANS, TWOSAMPLEFREQ, TWOSAMPLEMEANS, TWOSAMPLESURVIVAL, and TWOSAMPLEWILCOXON.

You can use multiple analysis statements and multiple PLOT statements. Each analysis statement produces a separate sample size analysis. Each PLOT statement refers to the previous analysis statement and generates a separate graph (or set of graphs).

The name of an analysis statement describes the framework of the statistical analysis for which sample size calculations are desired. You use options in the analysis statements to identify the result parameter to compute, to specify the statistical test and computational options, and to provide one or more scenarios for the values of relevant analysis parameters.

Table 67.1 summarizes the basic functions of each statement in PROC POWER. The syntax of each statement in Table 67.1 is described in the following pages.

Table 67.1 Statements in the POWER Procedure

Statement	Description
PROC POWER	invokes the procedure
LOGISTIC	likelihood ratio chi-square test of a single predictor in logistic regression with binary response
MULTREG	tests of one or more coefficients in multiple linear regression

Table 67.1 *continued*

Statement	Description
ONECORR	Fisher's z test and t test of (partial) correlation
ONESAMPLEFREQ	tests, confidence interval precision, and equivalence tests of a single binomial proportion
ONESAMPLEMEANS	one-sample t test, confidence interval precision, or equivalence test
ONEWAYANOVA	one-way ANOVA including single-degree-of-freedom contrasts
PAIREDFREQ	McNemar's test for paired proportions
PAIREDMEANS	paired t test, confidence interval precision, or equivalence test
PLOT	displays plots for previous sample size analysis
TWOSAMPLEFREQ	chi-square, likelihood ratio, and Fisher's exact tests for two independent proportions
TWOSAMPLEMEANS	two-sample t test, confidence interval precision, or equivalence test
TWOSAMPLESURVIVAL	log-rank, Gehan, and Tarone-Ware tests for comparing two survival curves
TWOSAMPLEWILCOXON	Wilcoxon-Mann-Whitney (rank-sum) test for 2 independent groups

See the section "Summary of Analyses" on page 4916 for a summary of the analyses available and the syntax required for them.

PROC POWER Statement

PROC POWER < *options* > **;**

The PROC POWER statement invokes the POWER procedure. You can specify the following option.

PLOTONLY
specifies that only graphical results from the PLOT statement should be produced.

LOGISTIC Statement

LOGISTIC < *options* > **;**

The LOGISTIC statement performs power and sample size analyses for the likelihood ratio chi-square test of a single predictor in binary logistic regression, possibly in the presence of one or

more covariates. All predictor variables are assumed to be independent of each other. So, this analysis is not applicable to studies with correlated predictors—for example, most observational studies (as opposed to randomized studies).

Summary of Options

Table 67.2 summarizes categories of options available in the LOGISTIC statement.

Table 67.2 Summary of Options in the LOGISTIC Statement

Task	Options
Define analysis	TEST=
Specify analysis information	ALPHA= COVARIATES= TESTPREDICTOR= VARDIST=
Specify effects	RESPONSEPROB= COVODDSRATIOS= COVREGCOEFFS= DEFAULTUNIT= INTERCEPT= TESTODDSRATIO= TESTREGCOEFF= UNITS=
Specify sample size	NTOTAL=
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Specify computational method	DEFAULTNBINS= NBINS=
Control ordering in output	OUTPUTORDER=

Table 67.3 summarizes the valid result parameters in the LOGISTIC statement.

Table 67.3 Summary of Result Parameters in the LOGISTIC Statement

Analyses	Solve For	Syntax
TEST=LRCHI	Power	POWER=.
	Sample size	NTOTAL=.

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to

the usual $0.05 \times 100\% = 5\%$ level of significance. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

COVARIATES=*grouped-name-list*

specifies the distributions of any predictor variables in the model but not being tested, using labels specified with the VARDIST= option. The distributions are assumed to be independent of each other and of the tested predictor. If this option is omitted, then the tested predictor specified by the TESTEDPREDICTOR= option is assumed to be the only predictor in the model. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-name-list*.

COVODDSRATIOS=*grouped-number-list*

specifies the odds ratios for the covariates in the full model (including variables in the TESTPREDICTOR= and COVARIATES= options). The ordering of the values corresponds to the ordering in the COVARIATES= option. If the response variable is coded as $Y = 1$ for success and $Y = 0$ for failure, then the odds ratio for each covariate X is the odds of $Y = 1$ when $X = a$ divided by the odds of $Y = 1$ when $X = b$, where a and b are determined from the DEFAULTUNIT= and UNITS= options. Values must be greater than zero. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

COVREGCOEFFS=*grouped-number-list*

specifies the regression coefficients for the covariates in the full model including the test predictor (as specified by the TESTPREDICTOR= option). The ordering of the values corresponds to the ordering in the COVARIATES= option. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

DEFAULTNBINS=*number*

specifies the default number of categories (or "bins") into which the distribution for each predictor variable is divided in internal calculations. Higher values increase computational time and memory requirements but generally lead to more accurate results. Each test predictor or covariate that is absent from the NBINS= option derives its bin number from the DEFAULTNBINS= option. The default value is DEFAULTNBINS=10.

There are two variable distributions for which the number of bins can be overridden internally:

- For an ordinal distribution, the number of ordinal values is always used as the number of bins.
- For a binomial distribution, if the requested number of bins is larger than $n + 1$, where n is the sample size parameter of the binomial distribution, then exactly $n + 1$ bins are used.

DEFAULTUNIT=*change-spec*

specifies the default change in the predictor variables assumed for odds ratios specified with the COVODDSRATIOS= and TESTODDSRATIO= options. Each test predictor or covariate that is absent from the UNITS= option derives its change value from the DEFAULTUNIT= option. The value must be nonzero. The default value is DEFAULTUNIT=1. This option can be used only if at least one of the COVODDSRATIOS= and TESTODDSRATIO= options is used.

Valid specifications for *change-spec* are as follows:

number defines the odds ratio as the ratio of the response variable odds when $X = a$ to the odds when $X = a - number$ for any constant a.

<+ | ->SD defines the odds ratio as the ratio of the odds when $X = a$ to the odds when $X = a - \sigma$ (or $X = a + \sigma$, if SD is preceded by a minus sign (–)) for any constant a, where σ is the standard deviation of X (as determined from the VARDIST= option).

*multiple**SD defines the odds ratio as the ratio of the odds when $X = a$ to the odds when $X = a - multiple * \sigma$ for any constant a, where σ is the standard deviation of X (as determined from the VARDIST= option).

PERCENTILES(*p1*, *p2*) defines the odds ratio as the ratio of the odds when X is equal to its $p2 * 100$th percentile to the odds when X is equal to its $p1 * 100$th percentile (where the percentiles are determined from the distribution specified in the VARDIST= option). Values for *p1* and *p2* must be strictly between 0 and 1.

INTERCEPT=*number-list*

specifies the intercept in the full model (including variables in the TESTPREDICTOR= and COVARIATES= options). See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NBINS=(*"name"* **=** *number* **<...** *"name"* **=** *number* **>)**

specifies the number of categories (or "bins") into which the distribution for each predictor variable (identified by its *name* from the VARDIST= option) is divided in internal calculations. Higher values increase computational time and memory requirements but generally lead to more accurate results. Each predictor variable that is absent from the NBINS= option derives its bin number from the DEFAULTNBINS= option.

There are two variable distributions for which the NBINS= value can be overridden internally:

- For an ordinal distribution, the number of ordinal values is always used as the number of bins.
- For a binomial distribution, if the requested number of bins is larger than $n + 1$, where *n* is the sample size parameter of the binomial distribution, then exactly $n + 1$ bins are used.

NFRACTIONAL

NFRAC

enables fractional input and output for sample sizes. See the section "Sample Size Adjustment Options" on page 4922 for information about the ramifications of the presence (and absence) of the NFRACTIONAL option.

NTOTAL=*number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). Values must be at least one. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

OUTPUTORDER=INTERNAL

OUTPUTORDER=REVERSE

OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) arranges the parameters in the output according to the following order of their corresponding options:

- DEFAULTNBINS=
- NBINS=
- ALPHA=
- RESPONSEPROB=
- INTERCEPT=
- TESTPREDICTOR=
- TESTODDSRATIO=
- TESTREGCOEFF=
- COVARIATES=
- COVODDSRATIOS=
- COVREGCOEFFS=
- NTOTAL=
- POWER=

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order in which their corresponding options are specified in the LOGISTIC statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order in which their corresponding options are specified in the LOGISTIC statement.

POWER=*number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

RESPONSEPROB=*number-list*

specifies the response probability in the full model when all predictor variables (including variables in the TESTPREDICTOR= and COVARIATES= options) are equal to their means. The log odds of this probability are equal to the intercept in the full model where all predictor are centered at their means. If the response variable is coded as $Y = 1$ for success and $Y = 0$ for failure, then this probability is equal to the mean of Y in the full model when all Xs are equal to their means. Values must be strictly between zero and one. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

TEST=LRCHI

specifies the likelihood ratio chi-square test of a single model parameter in binary logistic regression. This is the default test option.

TESTODDSRATIO=*number-list*

specifies the odds ratio for the predictor variable being tested in the full model (including variables in the TESTPREDICTOR= and COVARIATES= options). If the response variable is coded as $Y = 1$ for success and $Y = 0$ for failure, then the odds ratio for the X being tested is the odds of $Y = 1$ when $X = a$ divided by the odds of $Y = 1$ when $X = b$, where a and b are determined from the DEFAULTUNIT= and UNITS= options. Values must be greater than zero. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

TESTPREDICTOR=*name-list*

specifies the distribution of the predictor variable being tested, using labels specified with the VARDIST= option. This distribution is assumed to be independent of the distributions of the covariates as defined in the COVARIATES= option. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *name-list*.

TESTREGCOEFF=*number-list*

specifies the regression coefficient for the predictor variable being tested in the full model including the covariates specified by the COVARIATES= option. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

UNITS=(*"name"* **=** *change-spec* **<...** *"name"* **=** *change-spec* **>)**

specifies the changes in the predictor variables assumed for odds ratios specified with the COVODDSRATIOS= and TESTODDSRATIO= options. Each predictor variable whose *name* (from the VARDIST= option) is absent from the UNITS option derives its change value from the DEFAULTUNIT= option. This option can be used only if at least one of the COVODDSRATIOS= and TESTODDSRATIO= options is used.

Valid specifications for *change-spec* are as follows:

number defines the odds ratio as the ratio of the response variable odds when $X = a$ to the odds when $X = a - number$ for any constant a.

<+ | ->SD defines the odds ratio as the ratio of the odds when $X = a$ to the odds when $X = a - \sigma$ (or $X = a + \sigma$, if SD is preceded by a minus sign (–)) for any constant a, where σ is the standard deviation of X (as determined from the VARDIST= option).

*multiple**SD defines the odds ratio as the ratio of the odds when $X = a$ to the odds when $X = a - multiple * \sigma$ for any constant a, where σ is the standard deviation of X (as determined from the VARDIST= option).

PERCENTILES(*p1*, *p2*) defines the odds ratio as the ratio of the odds when X is equal to its $p2 * 100$th percentile to the odds when X is equal to its $p1 * 100$th percentile (where the percentiles are determined from the distribution specified in the VARDIST= option). Values for *p1* and *p2* must be strictly between 0 and 1.

Each unit value must be nonzero.

VARDIST("label")=*distribution* **(***parameters***)**

defines a distribution for a predictor variable.

For the VARDIST= option,

label identifies the variable distribution in the output and with the COVARIATES= and TESTPREDICTOR= options.

distribution specifies the distributional form of the variable.

parameters specifies one or more parameters associated with the distribution.

Choices for distributional forms and their parameters are as follows:

ORDINAL ((*values*) : (*probabilities*)) is an ordered categorical distribution. The *values* are any numbers separated by spaces. The *probabilities* are numbers between 0 and 1 (inclusive) separated by spaces. Their sum must be exactly 1. The number of *probabilities* must match the number of *values*.

BETA (*a*, *b* <, *l*, *r* >) is a beta distribution with shape parameters a and b and optional location parameters l and r. The values of a and b must be greater than 0, and l must be less than r. The default values for l and r are 0 and 1, respectively.

BINOMIAL (*p*, *n*) is a binomial distribution with probability of success p and number of independent Bernoulli trials n. The value of p must be greater than 0 and less than 1, and n must be an integer greater than 0.

EXPONENTIAL (λ) is an exponential distribution with scale λ, which must be greater than 0.

GAMMA (*a*, λ) is a gamma distribution with shape a and scale λ. The values of a and λ must be greater than 0.

LAPLACE (θ, λ) is a Laplace distribution with location θ and scale λ. The value of λ must be greater than 0.

LOGISTIC (θ, λ) is a logistic distribution with location θ and scale λ. The value of λ must be greater than 0.

LOGNORMAL (θ, λ) is a lognormal distribution with location θ and scale λ. The value of λ must be greater than 0.

NORMAL (θ, λ) is a normal distribution with mean θ and standard deviation λ. The value of λ must be greater than 0.

POISSON (*m*) is a Poisson distribution with mean m. The value of m must be greater than 0.

UNIFORM (*l*, *r*) is a uniform distribution on the interval $[l, r]$, where $l < r$.

Restrictions on Option Combinations

To specify the intercept in the full model, choose one of the following two parameterizations:

- intercept (using the INTERCEPT= options)
- Prob($Y = 1$) when all predictors are equal to their means (using the RESPONSEPROB= option)

To specify the effect associated with the predictor variable being tested, choose one of the following two parameterizations:

- odds ratio (using the TESTODDSRATIO= options)
- regression coefficient (using the TESTREGCOEFFS= option)

To describe the effects of the covariates in the full model, choose one of the following two parameterizations:

- odds ratios (using the COVODDSRATIOS= options)
- regression coefficients (using the COVREGCOEFFS= options)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the LOGISTIC statement.

Likelihood Ratio Chi-Square Test for One Predictor

You can express effects in terms of response probability and odds ratios, as in the following statements:

```
proc power;
   logistic
      vardist("x1a") = normal(0, 2)
      vardist("x1b") = normal(0, 3)
      vardist("x2") = poisson(7)
      vardist("x3a") = ordinal((-5 0 5) : (.3 .4 .3))
      vardist("x3b") = ordinal((-5 0 5) : (.4 .3 .3))
      testpredictor = "x1a" "x1b"
      covariates = "x2" | "x3a" "x3b"
      responseprob = 0.15
      testoddsratio = 1.75
      covoddsratios = (2.1 1.4)
      ntotal = 100
      power = .;
run;
```

The VARDIST= options define the distributions of the predictor variables. The TESTPREDICTOR= option specifies two scenarios for the test predictor distribution, Normal(10,2) and Normal(10,3). The COVARIATES= option specifies two covariates, the first with a Poisson(7) distribution. The second covariate has an ordinal distribution on the values –5, 0, and 5 with two scenarios for the associated probabilities: (.3, .4, .3) and (.4, .3, .3). The response probability in the full model with all variables equal to zero is specified by the RESPONSEPROB= option as 0.15. The odds ratio for a unit decrease in the tested predictor is specified by the TESTODDSRATIO= option to be 1.75. Corresponding odds ratios for the two covariates in the full model are specified by the COVODDSRATIOS= option to be 2.1 and 1.4. The POWER=. option requests a solution for the power at a sample size of 100 as specified by the NTOTAL= option.

Default values of the TEST= and ALPHA= options specify a likelihood ratio test of the first predictor with a significance level of 0.05. The default of DEFAULTUNIT=1 specifies that all odds

ratios are defined in terms of unit changes in predictors. The default of DEFAULTNBINS=10 specifies that each of the three predictor variables is discretized into a distribution with 10 categories in internal calculations.

You can also express effects in terms of regression coefficients, as in the following statements:

```
proc power;
   logistic
      vardist("x1a") = normal(0, 2)
      vardist("x1b") = normal(0, 3)
      vardist("x2") = poisson(7)
      vardist("x3") = ordinal((-5 0 5) : (.3 .4 .3))
      testpredictor = "x1a" "x1b"
      covariates = "x2" | "x3a" "x3b"
      intercept = -6.928162
      testregcoeff = 0.5596158
      covregcoeffs = (0.7419373 0.3364722)
      ntotal = 100
      power = .;
run;
```

The regression coefficients for the tested predictor (TESTREGCOEFF=0.5596158) and covariates (COVREGCOEFFS=(0.7419373 0.3364722)) are determined by taking the logarithm of the corresponding odds ratios. The intercept in the full model is specified as –6.928162 by the INTERCEPT= option. This number is calculated according to the forumula at the end of "Analyses in the LOGISTIC Statement" on page 4928, which expresses the intercept in terms of the response probability, regression coefficients, and predictor means:

$$\text{Intercept} = \log\left(\frac{0.15}{1-0.15}\right) - (0.5596158(0) + 0.7419373(7) + 0.3364722(0))$$

MULTREG Statement

MULTREG < *options* > **;**

The MULTREG statement performs power and sample size analyses for Type III F tests of sets of predictors in multiple linear regression, assuming either fixed or normally distributed predictors.

Summary of Options

Table 67.4 summarizes categories of options available in the MULTREG statement.

Table 67.4 Summary of Options in the MULTREG Statement

Task	Options
Define analysis	TEST=
Specify analysis information	ALPHA= MODEL= NFULLPREDICTORS= NOINT NREDUCEDPREDICTORS= NTESTPREDICTORS=
Specify effects	PARTIALCORR= RSQUAREDIFF= RSQUAREFULL= RSQUAREREDUCED=
Specify sample size	NTOTAL=
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Control ordering in output	OUTPUTORDER=

Table 67.5 summarizes the valid result parameters in the MULTREG statement.

Table 67.5 Summary of Result Parameters in the MULTREG Statement

Analyses	Solve For	Syntax
TEST=TYPE3	Power	POWER=.
	Sample size	NTOTAL=.

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual $0.05 \times 100\% = 5\%$ level of significance. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

MODEL=*keyword-list*

specifies the assumed distribution of the tested predictors. MODEL=FIXED indicates a fixed predictor distribution. MODEL=RANDOM (the default) indicates a joint multivariate normal distribution for the response and tested predictors. You can use the aliases CONDITIONAL for FIXED and UNCONDITIONAL for RANDOM. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *keyword-list*.

FIXED fixed predictors

RANDOM random (multivariate normal) predictors

NFRACTIONAL

NFRAC

enables fractional input and output for sample sizes. See the section "Sample Size Adjustment Options" on page 4922 for information about the ramifications of the presence (and absence) of the NFRACTIONAL option.

NFULLPREDICTORS=*number-list*

NFULLPRED=*number-list*

specifies the number of predictors in the full model, not counting the intercept. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NOINT

specifies a no-intercept model (for both full and reduced models). By default, the intercept is included in the model. If you want to test the intercept, you can specify the NOINT option and simply consider the intercept to be one of the predictors being tested. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NREDUCEDPREDICTORS=*number-list*

NREDUCEDPRED=*number-list*

NREDPRED=*number-list*

specifies the number of predictors in the reduced model, not counting the intercept. This is the same as the difference between values of the NFULLPREDICTORS= and NTESTPREDICTORS= options. Note that supplying a value of 0 is the same as specifying an F test of a Pearson correlation. This option cannot be used at the same time as the NTESTPREDICTORS= option. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NTESTPREDICTORS=*number-list*

NTESTPRED=*number-list*

specifies the number of predictors being tested. This is the same as the difference between values of the NFULLPREDICTORS= and NREDUCEDPREDICTORS= options. Note that supplying identical values for the NTESTPREDICTORS= and NFULLPREDICTORS= options is the same as specifying an F test of a Pearson correlation. This option cannot be used at the same time as the NREDUCEDPREDICTORS= option. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NTOTAL=*number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). The minimum acceptable value for the sample size depends on the MODEL=, NOINT, NFULLPREDICTORS=, NTESTPREDICTORS=, and NREDUCEDPREDICTORS= options. It ranges from $p+1$ to $p+3$, where p is the value of the NFULLPREDICTORS option. See Table 67.30 for further information about minimum NTOTAL values, and see the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

OUTPUTORDER=INTERNAL

OUTPUTORDER=REVERSE

OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) arranges the parameters in the output according to the following order of their corresponding options:

- MODEL=
- NFULLPREDICTORS=
- NTESTPREDICTORS=
- NREDUCEDPREDICTORS=
- ALPHA=
- PARTIALCORR=
- RSQUAREFULL=
- RSQUAREREDUCED=
- RSQUAREDIFF=
- NTOTAL=
- POWER=

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order in which their corresponding options are specified in the MULTREG statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order in which their corresponding options are specified in the MULTREG statement.

PARTIALCORR=*number-list*

PCORR=*number-list*

specifies the partial correlation between the tested predictors and the response, adjusting for any other predictors in the model. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

POWER=*number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

RSQUAREDIFF=*number-list*

RSQDIFF=*number-list*

specifies the difference in R^2 between the full and reduced models. This is equivalent to the proportion of variation explained by the predictors you are testing. It is also equivalent to the squared semipartial correlation of the tested predictors with the response. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

RSQUAREFULL=*number-list*

RSQFULL=*number-list*

specifies the R^2 of the full model, where R^2 is the proportion of variation explained by the model. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

RSQUAREREDUCED=*number-list*

RSQREDUCED=*number-list*

RSQRED=*number-list*

specifies the R^2 of the reduced model, where R^2 is the proportion of variation explained by the model. If the reduced model is an empty or intercept-only model (in other words, if NREDUCEDPREDICTORS=0 or NTESTPREDICTORS=NFULLPREDICTORS), then RSQUAREREDUCED=0 is assumed. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

TEST=TYPE3

specifies a Type III F test of a set of predictors adjusting for any other predictors in the model. This is the default test option.

Restrictions on Option Combinations

To specify the number of predictors, use any two of these three options:

- the number of predictors in the full model (NFULLPREDICTORS=)
- the number of predictors in the reduced model (NREDUCEDPREDICTORS=)
- the number of predictors being tested (NTESTPREDICTORS=)

To specify the effect, choose one of the following parameterizations:

- partial correlation (by using the PARTIALCORR= option)
- R^2 for the full and reduced models (by using any two of RSQUAREDIFF=, RSQUAREFULL=, and RSQUAREREDUCED=)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the MULTREG statement.

Type III F Test of a Set of Predictors

You can express effects in terms of partial correlation, as in the following statements. Default values of the TEST=, MODEL=, and ALPHA= options specify a Type III F test with a significance level of 0.05, assuming normally distributed predictors.

```
proc power;
   multreg
      model = random
      nfullpredictors = 7
      ntestpredictors = 3
      partialcorr = 0.35
      ntotal = 100
      power = .;
run;
```

You can also express effects in terms of R^2:

```
proc power;
   multreg
      model = fixed
      nfullpredictors = 7
      ntestpredictors = 3
      rsquarefull = 0.9
      rsquarediff = 0.1
      ntotal = .
      power = 0.9;
run;
```

ONECORR Statement

ONECORR < *options* > **;**

The ONECORR statement performs power and sample size analyses for tests of simple and partial Pearson correlation between two variables. Both Fisher's z test and the t test are supported.

Summary of Options

Table 67.6 summarizes categories of options available in the ONECORR statement.

Table 67.6 Summary of Options in the ONECORR Statement

Task	Options
Define analysis	DIST= TEST=
Specify analysis information	ALPHA= MODEL= NPARTIALVARS= NULLCORR= SIDES=
Specify effects	CORR=
Specify sample size	NTOTAL=

Table 67.6 *continued*

Task	Options
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Control ordering in output	OUTPUTORDER=

Table 67.7 summarizes the valid result parameters in the ONECORR statement.

Table 67.7 Summary of Result Parameters in the ONECORR Statement

Analyses	Solve For	Syntax
TEST=PEARSON	Power	POWER=.
	Sample size	NTOTAL=.

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual $0.05 \times 100\% = 5\%$ level of significance. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

CORR=*number-list*

specifies the correlation between two variables, possibly adjusting for other variables as determined by the NPARTIALVARS= option. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

DIST=FISHERZ

DIST=T

specifies the underlying distribution assumed for the test statistic. FISHERZ corresponds to Fisher's z normalizing transformation of the correlation coefficient. T corresponds to the t transformation of the correlation coefficient. Note that DIST=T is equivalent to analyses in the MULTREG statement with NTESTPREDICTORS=1. The default value is FISHERZ.

MODEL=*keyword-list*

specifies the assumed distribution of the first variable when DIST=T. The second variable is assumed to have a normal distribution. MODEL=FIXED indicates a fixed distribution. MODEL=RANDOM (the default) indicates a joint bivariate normal distribution with the second variable. You can use the aliases CONDITIONAL for FIXED and UNCONDITIONAL for RANDOM. This option can be used only for DIST=T. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *keyword-list*.

FIXED fixed variables

RANDOM random (bivariate normal) variables

NFRACTIONAL
NFRAC

enables fractional input and output for sample sizes. See the section "Sample Size Adjustment Options" on page 4922 for information about the ramifications of the presence (and absence) of the NFRACTIONAL option.

NPARTIALVARS=*number-list*
NPVARS=*number-list*

specifies the number of variables adjusted for in the correlation between the two primary variables. The default value is 0, corresponding to a simple correlation. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NTOTAL=*number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). Values for the sample size must be at least $p + 3$ when DIST=T and MODEL=CONDITIONAL, and at least $p + 4$ when either DIST=FISHER or when DIST=T and MODEL=UNCONDITIONAL, where p is the value of the NPARTIALVARS option. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NULLCORR=*number-list*
NULLC=*number-list*

specifies the null value of the correlation. The default value is 0. This option can be used only with the DIST=FISHERZ analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

OUTPUTORDER=INTERNAL
OUTPUTORDER=REVERSE
OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) arranges the parameters in the output according to the following order of their corresponding options:

- MODEL=
- SIDES=
- NULL=
- ALPHA=
- NPARTIALVARS=
- CORR=
- NTOTAL=
- POWER=

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order in which their corresponding options are specified in the ONECORR statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order in which their corresponding options are specified in the ONECORR statement.

POWER=*number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and the direction of the statistical test. Valid keywords are

1	one-sided with alternative hypothesis in same direction as effect
2	two-sided
U	upper one-sided with alternative greater than null value
L	lower one-sided with alternative less than null value

The default value is 2.

TEST=PEARSON

specifies a test of the Pearson correlation coefficient between two variables, possibly adjusting for other variables. This is the default test option.

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the ONECORR statement.

Fisher's z Test for Pearson Correlation

The following statements demonstrate a power computation for Fisher's z test for correlation. Default values of TEST=PEARSON, ALPHA=0.05, SIDES=2, and NPARTIALVARS=0 are assumed.

```
proc power;
   onecorr dist=fisherz
      nullcorr = 0.15
      corr = 0.35
      ntotal = 180
      power = .;
run;
```

t *Test for Pearson Correlation*

The following statements demonstrate a sample size computation for the *t* test for correlation. Default values of TEST=PEARSON, MODEL=RANDOM, ALPHA=0.05, and SIDES=2 are assumed.

```
proc power;
   onecorr dist=t
      npartialvars = 4
      corr = 0.45
      ntotal = .
      power = 0.85;
run;
```

ONESAMPLEFREQ Statement

ONESAMPLEFREQ < *options* > **;**

The ONESAMPLEFREQ statement performs power and sample size analyses for exact and approximate tests (including equivalence, noninferiority, and superiority) and confidence interval precision for a single binomial proportion.

Summary of Options

Table 67.8 summarizes categories of options available in the ONESAMPLEFREQ statement.

Table 67.8 Summary of Options in the ONESAMPLEFREQ Statement

Task	Options
Define analysis	CI= TEST=
Specify analysis information	ALPHA= EQUIVBOUNDS= LOWER= MARGIN= NULLPROPORTION= SIDES= UPPER=
Specify effect	HALFWIDTH= PROPORTION=
Specify variance estimation	VAREST=
Specify sample size	NTOTAL=
Specify power and related probabilities	POWER= PROBWIDTH=
Control sample size rounding	NFRACTIONAL
Choose computational method	METHOD=
Control ordering in output	OUTPUTORDER=

Table 67.9 summarizes the valid result parameters for different analyses in the ONESAMPLEFREQ statement.

Table 67.9 Summary of Result Parameters in the ONESAMPLEFREQ Statement

Analyses	Solve For	Syntax
CI=WILSON	Prob(width)	PROBWIDTH=.
CI=AGRESTICOULL	Prob(width)	PROBWIDTH=.
CI=JEFFREYS	Prob(width)	PROBWIDTH=.
CI=EXACT	Prob(width)	PROBWIDTH=.
CI=WALD	Prob(width)	PROBWIDTH=.
CI=WALD_CORRECT	Prob(width)	PROBWIDTH=.
TEST=ADJZ METHOD=EXACT	Power	POWER=.
TEST=ADJZ METHOD=NORMAL	Power Sample size	POWER=. NTOTAL=.
TEST=EQUIV_ADJZ METHOD=EXACT	Power	POWER=.
TEST=EQUIV_ADJZ METHOD=NORMAL	Power Sample size	POWER=. NTOTAL=.
TEST=EQUIV_EXACT	Power	POWER=.
TEST=EQUIV_Z METHOD=EXACT	Power	POWER=.
TEST=EQUIV_Z METHOD=NORMAL	Power Sample size	POWER=. NTOTAL=.
TEST=EXACT	Power	POWER=.
TEST=Z METHOD=EXACT	Power	POWER=.
TEST=Z METHOD=NORMAL	Power Sample size	POWER=. NTOTAL=.

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual $0.05 \times 100\% = 5\%$ level of significance. If the CI= and SIDES=1 options are used, then the value must be less than 0.5. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

CI

CI=AGRESTICOULL | AC

CI=JEFFREYS

CI=EXACT | CLOPPERPEARSON | CP

CI=WALD

CI=WALD_CORRECT

CI=WILSON | SCORE

specifies an analysis of precision of a confidence interval for the sample binomial proportion.

The value of the CI= option specifies the type of confidence interval. The CI=AGRESTICOULL option is a generalization of the "Adjusted Wald / add 2 successes and 2 failures" interval of Agresti and Coull (1998) and is presented in Brown, Cai, and DasGupta (2001). It corresponds to the TABLES / BINOMIAL (AGRESTICOULL) option in PROC FREQ. The CI=JEFFREYS option specifies the equal-tailed Jeffreys prior Bayesian interval, corresponding to the TABLES / BINOMIAL (JEFFREYS) option in PROC FREQ. The CI=EXACT option specifies the exact Clopper-Pearson confidence interval based on enumeration, corresponding to the TABLES / BINOMIAL (EXACT) option in PROC FREQ. The CI=WALD option specifies the confidence interval based on the Wald test (also commonly called the z test or normal-approximation test), corresponding to the TABLES / BINOMIAL (WALD) option in PROC FREQ. The CI=WALD_CORRECT option specifies the confidence interval based on the Wald test with continuity correction, corresponding to the TABLES / BINOMIAL (CORRECT WALD) option in PROC FREQ. The CI=WILSON option (the default) specifies the confidence interval based on the score statistic, corresponding to the TABLES / BINOMIAL (WILSON) option in PROC FREQ.

Instead of power, the relevant probability for this analysis is the probability of achieving a desired precision. Specifically, it is the probability that the half-width of the confidence interval will be at most the value specified by the HALFWIDTH= option.

EQUIVBOUNDS=*grouped-number-list*

specifies the lower and upper equivalence bounds, representing the same information as the combination of the LOWER= and UPPER= options but grouping them together. The EQUIVBOUNDS= option can be used only with equivalence analyses (TEST=EQUIV_ADJZ | EQUIV_EXACT | EQUIV_Z). Values must be strictly between 0 and 1. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

HALFWIDTH=*number-list*

specifies the desired confidence interval half-width. The half-width for a two-sided interval is the length of the confidence interval divided by two. This option can be used only with the CI= analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

LOWER=*number-list*

specifies the lower equivalence bound for the binomial proportion. The LOWER= option can be used only with equivalence analyses (TEST=EQUIV_ADJZ | EQUIV_EXACT | EQUIV_Z). Values must be strictly between 0 and 1. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

MARGIN=*number-list*

specifies the equivalence or noninferiority or superiority margin, depending on the analysis.

The MARGIN= option can be used with one-sided analyses (SIDES = 1 | U | L), in which case it specifies the margin added to the null proportion value in the hypothesis test, resulting in a noninferiority or superiority test (depending on the agreement between the effect and hypothesis directions and the sign of the margin). A test with a null proportion p_0 and a margin m is the same as a test with null proportion $p_0 + m$ and no margin.

The MARGIN= option can also be used with equivalence analyses (TEST=EQUIV_ADJZ | EQUIV_EXACT | EQUIV_Z) when the NULLPROPORTION= option is used, in which case it specifies the lower and upper equivalence bounds as $p_0 - m$ and $p_0 + m$, where p_0 is the value of the NULLPROPORTION= option and m is the value of the MARGIN= option.

The MARGIN= option cannot be used in conjunction with the SIDES=2 option. (Instead, specify an equivalence analysis by using TEST=EQUIV_ADJZ or TEST=EQUIV_EXACT or TEST=EQUIV_Z). Also, the MARGIN= option cannot be used with the CI= option.

Values must be strictly between –1 and 1. In addition, the sum of NULLPROPORTION and MARGIN must be strictly between 0 and 1 for one-sided analyses, and the derived lower equivalence bound (2 * NULLPROPORTION – MARGIN) must be strictly between 0 and 1 for equivalence analyses.

See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

METHOD=EXACT

METHOD=NORMAL

specifies the computational method. METHOD=EXACT (the default) computes exact results by using the binomial distribution. METHOD=NORMAL computes approximate results by using the normal approximation to the binomial distribution.

NFRACTIONAL

NFRAC

enables fractional input and output for sample sizes. This option is invalid when the METHOD=EXACT option is specified. See the section "Sample Size Adjustment Options" on page 4922 for information about the ramifications of the presence (and absence) of the NFRACTIONAL option.

NTOTAL=*number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NULLPROPORTION=*number-list*

NULLP=*number-list*

specifies the null proportion. A value of 0.5 corresponds to the sign test. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

OUTPUTORDER=INTERNAL

OUTPUTORDER=REVERSE

OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) arranges the parameters in the output according to the following order of their corresponding options:

- SIDES=
- NULLPROPORTION=
- ALPHA=
- PROPORTION=
- NTOTAL=
- POWER=

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order in which their corresponding options are specified in the ONESAMPLEFREQ statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order in which their corresponding options are specified in the ONESAMPLEFREQ statement.

POWER=*number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

PROBWIDTH=*number-list*

specifies the desired probability of obtaining a confidence interval half-width less than or equal to the value specified by the HALFWIDTH= option. A missing value (PROBWIDTH=.) requests a solution for this probability. Values are expressed as probabilities (for example, 0.9) rather than percentages. This option can be used only with the CI= analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

PROPORTION=*number-list*

P=*number-list*

specifies the binomial proportion—that is, the expected proportion of successes in the hypothetical binomial trial. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and the direction of the statistical test. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *keyword-list*. Valid keywords are as follows:

1 one-sided with alternative hypothesis in same direction as effect

2 two-sided

U upper one-sided with alternative greater than null value

L lower one-sided with alternative less than null value

If the effect size is zero, then SIDES=1 is not permitted; instead, specify the direction of the test explicitly in this case with either SIDES=L or SIDES=U. The default value is 2.

TEST

TEST= ADJZ

TEST= EQUIV_ADJZ

TEST= EQUIV_EXACT

TEST= EQUIV_Z

TEST= EXACT

TEST= Z

specifies the statistical analysis. TEST=ADJZ specifies a normal-approximate z test with continuity adjustment. TEST=EQUIV_ADJZ specifies a normal-approximate two-sided equivalence test based on the z statistic with continuity adjustment and a TOST (two one-sided tests) procedure. TEST=EQUIV_EXACT specifies the exact binomial two-sided equivalence test based on a TOST (two one-sided tests) procedure. TEST=EQUIV_Z specifies a normal-approximate two-sided equivalence test based on the z statistic without any continuity adjustment, which is the same as the chi-square statistic, and a TOST (two one-sided tests) procedure. TEST or TEST=EXACT (the default) specifies the exact binomial test. TEST=Z specifies a normal-approximate z test without any continuity adjustment, which is the same as the chi-square test when SIDES=2.

UPPER=*number-list*

specifies the upper equivalence bound for the binomial proportion. The UPPER= option can be used only with equivalence analyses (TEST=EQUIV_ADJZ | EQUIV_EXACT | EQUIV_Z). Values must be strictly between 0 and 1. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

VAREST=*keyword-list*

specifies how the variance is computed in the test statistic for the TEST=Z, TEST=ADJZ, TEST=EQUIV_Z, and TEST=EQUIV_ADJZ analyses. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *keyword-list*. Valid keywords are as follows:

NULL (the default) estimates the variance by using the null proportion(s) (specified by some combination of the NULLPROPORTION=, MARGIN=, LOWER=, and UPPER= options). For TEST=Z and TEST=ADJZ, the null proportion is the value of the NULLPROPORTION= option plus the value of the MARGIN= option (if it is used). For TEST=EQUIV_Z and TEST=EQUIV_ADJZ, there are two null proportions, corresponding to the lower and upper equivalence bounds, one for each test in the TOST (two one-sided tests) procedure.

SAMPLE estimates the variance by using the observed sample proportion.

This option is ignored if the analysis is one other than TEST=Z, TEST=ADJZ, TEST=EQUIV_Z, or TEST=EQUIV_ADJZ.

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the ONESAMPLEFREQ statement.

Exact Test of a Binomial Proportion

The following statements demonstrate a power computation for the exact test of a binomial proportion. Defaults for the SIDES= and ALPHA= options specify a two-sided test with a 0.05 significance level.

```
proc power;
   onesamplefreq test=exact
      nullproportion = 0.2
      proportion = 0.3
      ntotal = 100
      power = .;
run;
```

z Test

The following statements demonstrate a sample size computation for the z test of a binomial proportion. Defaults for the SIDES=, ALPHA=, and VAREST= options specify a two-sided test with a 0.05 significance level that uses the null variance estimate.

```
proc power;
   onesamplefreq test=z method=normal
      nullproportion = 0.8
      proportion = 0.85
      sides = u
      ntotal = .
      power = .9;
run;
```

z Test with Continuity Adjustment

The following statements demonstrate a sample size computation for the z test of a binomial proportion with a continuity adjustment. Defaults for the SIDES=, ALPHA=, and VAREST= options specify a two-sided test with a 0.05 significance level that uses the null variance estimate.

```
proc power;
   onesamplefreq test=adjz method=normal
      nullproportion = 0.15
      proportion = 0.1
      sides = l
      ntotal = .
      power = .9;
run;
```

Exact Equivalence Test for a Binomial Proportion

You can specify equivalence bounds by using the EQUIVBOUNDS= option, as in the following statements:

```
proc power;
   onesamplefreq test=equiv_exact
      proportion = 0.35
      equivbounds = (0.2 0.4)
      ntotal = 50
      power = .;
run;
```

You can also specify the combination of NULLPROPORTION= and MARGIN= options:

```
proc power;
   onesamplefreq test=equiv_exact
      proportion = 0.35
      nullproportion = 0.3
      margin = 0.1
      ntotal = 50
      power = .;
run;
```

Finally, you can specify the combination of LOWER= and UPPER= options:

```
proc power;
   onesamplefreq test=equiv_exact
      proportion = 0.35
      lower = 0.2
      upper = 0.4
      ntotal = 50
      power = .;
run;
```

Note that the three preceding analyses are identical.

Exact Noninferiority Test for a Binomial Proportion

A noninferiority test corresponds to an upper one-sided test with a negative-valued margin, as demonstrated in the following statements:

```
proc power;
   onesamplefreq test=exact
      sides = U
      proportion = 0.15
      nullproportion = 0.1
      margin = -0.02
      ntotal = 130
      power = .;
run;
```

Exact Superiority Test for a Binomial Proportion

A superiority test corresponds to an upper one-sided test with a positive-valued margin, as demonstrated in the following statements:

```
proc power;
   onesamplefreq test=exact
      sides = U
      proportion = 0.15
      nullproportion = 0.1
      margin = 0.02
      ntotal = 130
      power = .;
run;
```

Confidence Interval Precision

The following statements performs a confidence interval precision analysis for the Wilson score-based confidence interval for a binomial proportion. The default value of the ALPHA= option specifies a confidence level of 0.95.

```
proc power;
   onesamplefreq ci=wilson
      halfwidth = 0.1
      proportion = 0.3
      ntotal = 70
      probwidth = .;
run;
```

Restrictions on Option Combinations

To specify the equivalence bounds for TEST=EQUIV_ADJZ, TEST=EQUIV_EXACT, and TEST=EQUIV_Z, use any of these three option sets:

- lower and upper equivalence bounds, using the EQUIVBOUNDS= option
- lower and upper equivalence bounds, using the LOWER= and UPPER= options
- null proportion (NULLPROPORTION=) and margin (MARGIN=)

ONESAMPLEMEANS Statement

ONESAMPLEMEANS < *options* > **;**

The ONESAMPLEMEANS statement performs power and sample size analyses for *t* tests, equivalence tests, and confidence interval precision involving one sample.

Summary of Options

Table 67.10 summarizes categories of options available in the ONESAMPLEMEANS statement.

Table 67.10 Summary of Options in the ONESAMPLEMEANS Statement

Task	Options
Define analysis	CI= DIST= TEST=
Specify analysis information	ALPHA= LOWER= NULLMEAN= SIDES= UPPER=
Specify effects	HALFWIDTH= MEAN=
Specify variability	CV= STDDEV=
Specify sample size	NTOTAL=
Specify power and related probabilities	POWER= PROBTYPE= PROBWIDTH=
Control sample size rounding	NFRACTIONAL
Control ordering in output	OUTPUTORDER=

Table 67.11 summarizes the valid result parameters for different analyses in the ONESAMPLEMEANS statement.

Table 67.11 Summary of Result Parameters in the ONESAMPLEMEANS Statement

Analyses	Solve For	Syntax
TEST=T DIST=NORMAL	Power Sample size Alpha Mean Standard Deviation	POWER=. NTOTAL=. ALPHA=. MEAN=. STDDEV=.
TEST=T DIST=LOGNORMAL	Power Sample size	POWER=. NTOTAL=.
TEST=EQUIV	Power Sample size	POWER=. NTOTAL=.
CI=T	Prob(width) Sample size	PROBWIDTH=. NTOTAL=.

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test or requests a solution for alpha with a missing value (ALPHA=.). The default is 0.05, corresponding to the usual 0.05 × 100% = 5% level of significance. If the CI= and SIDES=1 options are used, then the value must be less than 0.5. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

CI

CI=T

specifies an analysis of precision of the confidence interval for the mean. Instead of power, the relevant probability for this analysis is the probability of achieving a desired precision. Specifically, it is the probability that the half-width of the confidence interval will be at most the value specified by the HALFWIDTH= option. If neither the CI= option nor the TEST= option is used, the default is TEST=T.

CV=*number-list*

specifies the coefficient of variation, defined as the ratio of the standard deviation to the mean. You can use this option only with DIST=LOGNORMAL. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

DIST=LOGNORMAL

DIST=NORMAL

specifies the underlying distribution assumed for the test statistic. NORMAL corresponds the normal distribution, and LOGNORMAL corresponds to the lognormal distribution. The default value is NORMAL.

HALFWIDTH=*number-list*

specifies the desired confidence interval half-width. The half-width is defined as the distance between the point estimate and a finite endpoint. This option can be used only with the CI=T analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

LOWER=*number-list*

specifies the lower equivalence bound for the mean. This option can be used only with the TEST=EQUIV analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

MEAN=*number-list*

specifies the mean, in the original scale, or requests a solution for the mean with a missing value (MEAN=.). The mean is arithmetic if DIST=NORMAL and geometric if DIST=LOGNORMAL. This option can be used only with the TEST=T and TEST=EQUIV analyses. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NFRACTIONAL

NFRAC

enables fractional input and output for sample sizes. See the section "Sample Size Adjustment

Options" on page 4922 for information about the ramifications of the presence (and absence) of the NFRACTIONAL option.

NTOTAL=*number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NULLMEAN=*number-list*

NULLM=*number-list*

specifies the null mean, in the original scale (whether DIST=NORMAL or DIST=LOGNORMAL). The default value is 0 when DIST=NORMAL and 1 when DIST=LOGNORMAL. This option can be used only with the TEST=T analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

OUTPUTORDER=INTERNAL

OUTPUTORDER=REVERSE

OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) arranges the parameters in the output according to the following order of their corresponding options:

- SIDES=
- NULLMEAN=
- LOWER=
- UPPER=
- ALPHA=
- MEAN=
- HALFWIDTH=
- STDDEV=
- CV=
- NTOTAL=
- POWER=
- PROBTYPE=
- PROBWIDTH=

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order in which their corresponding options are specified in the ONESAMPLEMEANS statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order in which their corresponding options are specified in the ONESAMPLEMEANS statement.

POWER=*number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. This option can be used only with the TEST=T and TEST=EQUIV analyses. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

PROBTYPE=*keyword-list*

specifies the type of probability for the PROBWIDTH= option. A value of CONDITIONAL (the default) indicates the conditional probability that the confidence interval half-width is at most the value specified by the HALFWIDTH= option, given that the true mean is captured by the confidence interval. A value of UNCONDITIONAL indicates the unconditional probability that the confidence interval half-width is at most the value specified by the HALFWIDTH= option. You can use the alias GIVENVALIDITY for CONDITIONAL. The PROBTYPE= option can be used only with the CI=T analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *keyword-list*.

CONDITIONAL width probability conditional on interval containing the mean

UNCONDITIONAL unconditional width probability

PROBWIDTH=*number-list*

specifies the desired probability of obtaining a confidence interval half-width less than or equal to the value specified by the HALFWIDTH= option. A missing value (PROBWIDTH=.) requests a solution for this probability. The type of probability is controlled with the PROBTYPE= option. Values are expressed as probabilities (for example, 0.9) rather than percentages. This option can be used only with the CI=T analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and the direction of the statistical test or confidence interval. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *keyword-list*. Valid keywords and their interpretation for the TEST= analyses are as follows:

1 one-sided with alternative hypothesis in same direction as effect

2 two-sided

U upper one-sided with alternative greater than null value

L lower one-sided with alternative less than null value

For confidence intervals, SIDES=U refers to an interval between the lower confidence limit and infinity, and SIDES=L refers to an interval between minus infinity and the upper confidence limit. For both of these cases and SIDES=1, the confidence interval computations are equivalent. The SIDES= option can be used only with the TEST=T and CI=T analyses. The default value is 2.

STDDEV=*number-list*

STD=*number-list*

specifies the standard deviation, or requests a solution for the standard deviation with a missing value (STDDEV=.). You can use this option only with DIST=NORMAL. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

TEST

TEST=EQUIV

TEST=T

specifies the statistical analysis. TEST=EQUIV specifies an equivalence test of the mean by using a two one-sided tests (TOST) analysis (Schuirmann 1987). TEST or TEST=T (the default) specifies a *t* test on the mean. If neither the TEST= option nor the CI= option is used, the default is TEST=T.

UPPER=*number-list*

specifies the upper equivalence bound for the mean, in the original scale (whether DIST=NORMAL or DIST=LOGNORMAL). This option can be used only with the TEST=EQUIV analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

Restrictions on Option Combinations

To define the analysis, choose one of the following parameterizations:

- a statistical test (by using the TEST= option)
- confidence interval precision (by using the CI= option)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the ONESAMPLEMEANS statement.

One-Sample t *Test*

The following statements demonstrate a power computation for the one-sample *t* test. Default values for the DIST=, SIDES=, NULLMEAN=, and ALPHA= options specify a two-sided test for zero mean with a normal distribution and a significance level of 0.05.

```
proc power;
   onesamplemeans test=t
      mean = 7
      stddev = 3
      ntotal = 50
      power = .;
run;
```

One-Sample t Test with Lognormal Data

The following statements demonstrate a sample size computation for the one-sample *t* test for lognormal data. Default values for the SIDES=, NULLMEAN=, and ALPHA= options specify a two-sided test for unit mean with a significance level of 0.05.

```
proc power;
   onesamplemeans test=t dist=lognormal
      mean = 7
      cv = 0.8
      ntotal = .
      power = 0.9;
run;
```

Equivalence Test for Mean of Normal Data

The following statements demonstrate a power computation for the TOST equivalence test for a normal mean. Default values for the DIST= and ALPHA= options specify a normal distribution and a significance level of 0.05.

```
proc power;
   onesamplemeans test=equiv
      lower = 2
      upper = 7
      mean = 4
      stddev = 3
      ntotal = 100
      power = .;
run;
```

Equivalence Test for Mean of Lognormal Data

The following statements demonstrate a sample size computation for the TOST equivalence test for a lognormal mean. The default of ALPHA=0.05 specifies a significance level of 0.05.

```
proc power;
   onesamplemeans test=equiv dist=lognormal
      lower = 1
      upper = 5
      mean = 3
      cv = 0.6
      ntotal = .
      power = 0.85;
run;
```

Confidence Interval for Mean

By default CI=T analyzes the conditional probability of obtaining the desired precision, given that the interval contains the true mean, as in the following statements. The defaults of SIDES=2 and ALPHA=0.05 specify a two-sided interval with a confidence level of 0.95.

```
proc power;
   onesamplemeans ci = t
      halfwidth = 14
      stddev = 8
      ntotal = 50
      probwidth = .;
run;
```

ONEWAYANOVA Statement

ONEWAYANOVA < *options* > **;**

The ONEWAYANOVA statement performs power and sample size analyses for one-degree-of-freedom contrasts and the overall F test in one-way analysis of variance.

Summary of Options

Table 67.12 summarizes categories of options available in the ONEWAYANOVA statement.

Table 67.12 Summary of Options in the ONEWAYANOVA Statement

Task	Options
Define analysis	TEST=
Specify analysis information	ALPHA= CONTRAST= SIDES= NULLCONTRAST=
Specify effects	GROUPMEANS=
Specify variability	STDDEV=
Specify sample size and allocation	GROUPNS= GROUPWEIGHTS= NPERGROUP== NTOTAL=
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Control ordering in output	OUTPUTORDER=

Table 67.13 summarizes the valid result parameters for different analyses in the ONEWAYANOVA statement.

Table 67.13 Summary of Result Parameters in the ONEWAYANOVA Statement

Analyses	Solve For	Syntax
TEST=CONTRAST	Power	POWER=.
	Sample size	NTOTAL=.
		NPERGROUP==.
TEST=OVERALL	Power	POWER=.
	Sample size	NTOTAL=.
		NPERGROUP==.

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual $0.05 \times 100\% = 5\%$ level of significance. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

CONTRAST= (*values*) < (... *values*) >

specifies coefficients for single-degree-of-freedom hypothesis tests. You must provide a coefficient for every mean appearing in the GROUPMEANS= option. Specify multiple contrasts either with additional sets of coefficients or with additional CONTRAST= options. For example, you can specify two different contrasts of five means by using the following:

```
CONTRAST = (1 -1 0 0 0) (1 0 -1 0 0)
```

GROUPMEANS=*grouped-number-list*

GMEANS=*grouped-number-list*

specifies the group means. This option is used to implicitly set the number of groups. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

GROUPNS=*grouped-number-list*

GNS=*grouped-number-list*

specifies the group sample sizes. The number of groups represented must be the same as with the GROUPMEANS= option. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

GROUPWEIGHTS=*grouped-number-list*

GWEIGHTS=*grouped-number-list*

specifies the sample size allocation weights for the groups. This option controls how the total sample size is divided between the groups. Each set of values across all groups represents relative allocation weights. Additionally, if the NFRACTIONAL option is not used, the total sample size is restricted to be equal to a multiple of the sum of the group weights (so that

the resulting design has an integer sample size for each group while adhering exactly to the group allocation weights). The number of groups represented must be the same as with the GROUPMEANS= option. Values must be integers unless the NFRACTIONAL option is used. The default value is 1 for each group, amounting to a balanced design. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

NFRACTIONAL

NFRAC

enables fractional input and output for sample sizes. See the section "Sample Size Adjustment Options" on page 4922 for information about the ramifications of the presence (and absence) of the NFRACTIONAL option.

NPERGROUP=*number-list*

NPERG=*number-list*

specifies the common sample size per group or requests a solution for the common sample size per group with a missing value (NPERGROUP==.). Use of this option implicitly specifies a balanced design. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NTOTAL=*number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NULLCONTRAST=*number-list*

NULLC=*number-list*

specifies the null value of the contrast. The default value is 0. This option can be used only with the TEST=CONTRAST analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

OUTPUTORDER=INTERNAL

OUTPUTORDER=REVERSE

OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) arranges the parameters in the output according to the following order of their corresponding options:

- CONTRAST=
- SIDES=
- NULLCONTRAST=
- ALPHA=
- GROUPMEANS=
- STDDEV=
- GROUPWEIGHTS=
- NTOTAL=

- NPERGROUP==
- GROUPNS=
- POWER=

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order in which their corresponding options are specified in the ONEWAYANOVA statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order in which their corresponding options are specified in the ONEWAYANOVA statement.

POWER=*number-list*
specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

SIDES=*keyword-list*
specifies the number of sides (or tails) and the direction of the statistical test. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *keyword-list*. Valid keywords are as follows:

1	one-sided with alternative hypothesis in same direction as effect
2	two-sided
U	upper one-sided with alternative greater than null value
L	lower one-sided with alternative less than null value

This option can be used only with the TEST=CONTRAST analysis. The default value is 2.

STDDEV=*number-list*
STD=*number-list*
specifies the error standard deviation. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

TEST=CONTRAST
TEST=OVERALL
specifies the statistical analysis. TEST=CONTRAST specifies a one-degree-of-freedom test of a contrast of means. The test is the usual F test for the two-sided case and the usual t test for the one-sided case. TEST=OVERALL specifies the overall F test of equality of all means. The default is TEST=CONTRAST if the CONTRAST= option is used, and TEST=OVERALL otherwise.

Restrictions on Option Combinations

To specify the sample size and allocation, choose one of the following parameterizations:

- sample size per group in a balanced design (by using the NPERGROUP== option)

- total sample size and allocation weights (by using the NTOTAL= and GROUPWEIGHTS= options)
- individual group sample sizes (by using the GROUPNS= option)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the ONEWAYANOVA statement.

One-Degree-of-Freedom Contrast

You can use the NPERGROUP== option in a balanced design, as in the following statements. Default values for the SIDES=, NULLCONTRAST=, and ALPHA= options specify a two-sided test for a contrast value of 0 with a significance level of 0.05.

```
proc power;
   onewayanova test=contrast
      contrast = (1 0 -1)
      groupmeans = 3 | 7 | 8
      stddev = 4
      npergroup = 50
      power = .;
run;
```

You can also specify an unbalanced design with the NTOTAL= and GROUPWEIGHTS= options:

```
proc power;
   onewayanova test=contrast
      contrast = (1 0 -1)
      groupmeans = 3 | 7 | 8
      stddev = 4
      groupweights = (1 2 2)
      ntotal = .
      power = 0.9;
run;
```

Another way to specify the sample sizes is with the GROUPNS= option:

```
proc power;
   onewayanova test=contrast
      contrast = (1 0 -1)
      groupmeans = 3 | 7 | 8
      stddev = 4
      groupns = (20 40 40)
      power = .;
run;
```

Overall F Test

The following statements demonstrate a power computation for the overall F test in a one-way ANOVA. The default of ALPHA=0.05 specifies a significance level of 0.05.

```
proc power;
   onewayanova test=overall
      groupmeans = 3 | 7 | 8
      stddev = 4
      npergroup = 50
      power = .;
run;
```

PAIREDFREQ Statement

PAIREDFREQ < *options* > **;**

The PAIREDFREQ statement performs power and sample size analyses for McNemar's test for paired proportions.

Summary of Options

Table 67.14 summarizes categories of options available in the PAIREDFREQ statement.

Table 67.14 Summary of Options in the PAIREDFREQ Statement

Task	Options
Define analysis	DIST= TEST=
Specify analysis information	ALPHA= NULLDISCPROPRATIO= SIDES=
Specify effects	PAIREDPROPORTIONS= PROPORTIONDIFF= ODDSRATIO= RELATIVERISK= CORR= DISCPROPDIFF= DISCPROPORTIONS= DISCPROPRATIO= REFPROPORTION= TOTALPROPDISC=
Specify sample size	NPAIRS=
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Choose computational method	METHOD=
Control ordering in output	OUTPUTORDER=

Table 67.15 summarizes the valid result parameters in the PAIREDFREQ statement.

Table 67.15 Summary of Result Parameters in the PAIREDFREQ Statement

Analyses	Solve For	Syntax
TEST=MCNEMAR METHOD=CONNOR	Power Sample size	POWER=. NPAIRS=.
TEST=MCNEMAR METHOD=EXACT	Power	POWER=.
TEST=MCNEMAR METHOD=MIETTINEN	Power Sample size	POWER=. NPAIRS=.

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual $0.05 \times 100\% = 5\%$ level of significance. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

CORR=*number-list*

specifies the correlation ϕ between members of a pair. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

DISCPROPORTIONS=*grouped-number-list*

DISCPS=*grouped-number-list*

specifies the two discordant proportions, p_{10} and p_{01}. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

DISCPROPDIFF=*number-list*

DISCPDIFF=*number-list*

specifies the difference $p_{01} - p_{10}$ between discordant proportions. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

DISCPROPRATIO=*number-list*

DISCPRATIO=*number-list*

specifies the ratio p_{01}/p_{10} of discordant proportions. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

DIST=EXACT_COND

DIST=NORMAL

specifies the underlying distribution assumed for the test statistic. EXACT_COND corresponds to the exact conditional test, based on the exact binomial distribution of the two types of discordant pairs given the total number of discordant pairs. NORMAL corresponds to the conditional test based on the normal approximation to the binomial distribution of the two types of discordant pairs given the total number of discordant pairs. The default value is EXACT_COND.

METHOD=CONNOR

METHOD=EXACT

METHOD=MIETTINEN

specifies the computational method. METHOD=EXACT (the default) uses the exact binomial distributions of the total number of discordant pairs and the two types of discordant pairs. METHOD=CONNOR uses an approximation from Connor (1987), and METHOD=MIETTINEN uses an approximation from Miettinen (1968). The CONNOR and MIETTINEN methods are valid only for DIST=NORMAL.

NFRACTIONAL

NFRAC

enables fractional input and output for sample sizes. See the section "Sample Size Adjustment Options" on page 4922 for information about the ramifications of the presence (and absence) of the NFRACTIONAL option. This option cannot be used with METHOD=EXACT.

NPAIRS=*number-list*

specifies the total number of proportion pairs (concordant and discordant) or requests a solution for the number of pairs with a missing value (NPAIRS=.). See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NULLDISCPROPRATIO=*number-list*

NULLDISCPRATIO=*number-list*

NULLRATIO=*number-list*

NULLR=*number-list*

specifies the null value of the ratio of discordant proportions. The default value is 1. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

ODDSRATIO=*number-list*

OR=*number-list*

specifies the odds ratio $[p_{\cdot 1}/(1-p_{\cdot 1})]\,/\,[p_{1\cdot}/(1-p_{1\cdot})]$. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

OUTPUTORDER=INTERNAL

OUTPUTORDER=REVERSE

OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) arranges the parameters in the output according to the following order of their corresponding options:

- SIDES=
- NULLDISCPROPRATIO=
- ALPHA=
- PAIREDPROPORTIONS=
- PROPORTIONDIFF=

- ODDSRATIO=
- RELATIVERISK=
- CORR=
- DISCPROPORTIONS=
- DISCPROPDIFF=
- TOTALPROPDISC=
- REFPROPORTION=
- DISCPROPRATIO=
- NPAIRS=
- POWER=

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order in which their corresponding options are specified in the PAIREDFREQ statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order in which their corresponding options are specified in the PAIREDFREQ statement.

PAIREDPROPORTIONS=*grouped-number-list*

PPROPORTIONS=*grouped-number-list*

PAIREDPS=*grouped-number-list*

PPS=*grouped-number-list*

specifies the two paired proportions, $p_{1\cdot}$ and $p_{\cdot 1}$. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

POWER=*number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

PROPORTIONDIFF=*number-list*

PDIFF=*number-list*

specifies the proportion difference $p_{\cdot 1} - p_{1\cdot}$. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

REFPROPORTION=*number-list*

REFP=*number-list*

specifies either the reference first proportion $p_{1\cdot}$ (when used in conjunction with the PROPORTIONDIFF=, ODDSRATIO=, or RELATIVERISK= option) or the reference discordant proportion p_{10} (when used in conjunction with the DISCPROPDIFF= or DISCPROPRATIO= option). See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

RELATIVERISK=*number-list*

RR=*number-list*

specifies the relative risk $p_{\cdot 1}/p_{1\cdot}$. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and the direction of the statistical test or confidence interval. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *keyword-list*. Valid keywords and their interpretation are as follows:

1	one-sided with alternative hypothesis in same direction as effect
2	two-sided
U	upper one-sided with alternative greater than null value
L	lower one-sided with alternative less than null value

If the effect size is zero, then SIDES=1 is not permitted; instead, specify the direction of the test explicitly in this case with either SIDES=L or SIDES=U. The default value is 2.

TEST=MCNEMAR

specifies the McNemar test of paired proportions. This is the default test option.

TOTALPROPDISC=*number-list*

TOTALPDISC=*number-list*

PDISC=*number-list*

specifies the sum of the two discordant proportions, $p_{10} + p_{01}$. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

Restrictions on Option Combinations

To specify the proportions, choose one of the following parameterizations:

- discordant proportions (using the DISCPROPORTIONS= option)
- difference and sum of discordant proportions (using the DISCPROPDIFF= and TOTALPROPDISC=options)
- difference of discordant proportions and reference discordant proportion (using the DISCPROPDIFF= and REFPROPORTION= options)
- ratio of discordant proportions and reference discordant proportion (using the DISCPROPRATIO= and REFPROPORTION= options)
- ratio and sum of discordant proportions (using the DISCPROPRATIO= and TOTALPROPDISC=options)

- paired proportions and correlation (using the PAIREDPROPORTIONS= and CORR= options)
- proportion difference, reference proportion, and correlation (using the PROPORTIONDIFF=, REFPROPORTION=, and CORR= options)
- odds ratio, reference proportion, and correlation (using the ODDSRATIO=, REFPROPORTION=, and CORR= options)
- relative risk, reference proportion, and correlation (using the RELATIVERISK=, REFPROPORTION=, and CORR= options)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the PAIREDFREQ statement.

McNemar Exact Conditional Test

You can express effects in terms of the individual discordant proportions, as in the following statements. Default values for the TEST=, SIDES=, ALPHA=, and NULLDISCPROPRATIO= options specify a two-sided McNemar test for no effect with a significance level of 0.05.

```
proc power;
   pairedfreq dist=exact_cond
      discproportions = 0.15 | 0.45
      npairs = 80
      power = .;
run;
```

You can also express effects in terms of the difference and sum of discordant proportions:

```
proc power;
   pairedfreq dist=exact_cond
      discpropdiff = 0.3
      totalpropdisc = 0.6
      npairs = 80
      power = .;
run;
```

You can also express effects in terms of the difference of discordant proportions and the reference discordant proportion:

```
proc power;
   pairedfreq dist=exact_cond
      discpropdiff = 0.3
      refproportion = 0.15
      npairs = 80
      power = .;
run;
```

You can also express effects in terms of the ratio of discordant proportions and the denominator of the ratio:

```
proc power;
   pairedfreq dist=exact_cond
      discpropratio = 3
      refproportion = 0.15
      npairs = 80
      power = .;
run;
```

You can also express effects in terms of the ratio and sum of discordant proportions:

```
proc power;
   pairedfreq dist=exact_cond
      discpropratio = 3
      totalpropdisc = 0.6
      npairs = 80
      power = .;
run;
```

You can also express effects in terms of the paired proportions and correlation:

```
proc power;
   pairedfreq dist=exact_cond
      pairedproportions = 0.6 | 0.8
      corr = 0.4
      npairs = 45
      power = .;
run;
```

You can also express effects in terms of the proportion difference, reference proportion, and correlation:

```
proc power;
   pairedfreq dist=exact_cond
      proportiondiff = 0.2
      refproportion = 0.6
      corr = 0.4
      npairs = 45
      power = .;
run;
```

You can also express effects in terms of the odds ratio, reference proportion, and correlation:

```
proc power;
   pairedfreq dist=exact_cond
      oddsratio = 2.66667
      refproportion = 0.6
      corr = 0.4
      npairs = 45
      power = .;
run;
```

You can also express effects in terms of the relative risk, reference proportion, and correlation:

```
proc power;
   pairedfreq dist=exact_cond
      relativerisk = 1.33333
      refproportion = 0.6
      corr = 0.4
      npairs = 45
      power = .;
run;
```

McNemar Normal Approximation Test

The following statements demonstrate a sample size computation for the normal-approximate McNemar test. The default value for the METHOD= option specifies an exact sample size computation. Default values for the TEST=, SIDES=, ALPHA=, and NULLDISCPROPRATIO= options specify a two-sided McNemar test for no effect with a significance level of 0.05.

```
proc power;
   pairedfreq dist=normal method=connor
      discproportions = 0.15 | 0.45
      npairs = .
      power = .9;
run;
```

PAIREDMEANS Statement

PAIREDMEANS < *options* > **;**

The PAIREDMEANS statement performs power and sample size analyses for *t* tests, equivalence tests, and confidence interval precision involving paired samples.

Summary of Options

Table 67.16 summarizes categories of options available in the PAIREDMEANS statement.

Table 67.16 Summary of Options in the PAIREDMEANS Statement

Task	Options
Define analysis	CI= DIST= TEST=
Specify analysis information	ALPHA= LOWER= NULLDIFF= NULLRATIO= SIDES=

Table 67.16 *continued*

Task	Options
	UPPER=
Specify effects	HALFWIDTH= MEANDIFF= MEANRATIO= PAIREDMEANS=
Specify variability	CORR= CV= PAIREDCVS= PAIREDSTDDEVS= STDDEV=
Specify sample size	NPAIRS=
Specify power and related probabilities	POWER= PROBTYPE= PROBWIDTH=
Control sample size rounding	NFRACTIONAL
Control ordering in output	OUTPUTORDER=

Table 67.17 summarizes the valid result parameters for different analyses in the PAIREDMEANS statement.

Table 67.17 Summary of Result Parameters in the PAIREDMEANS Statement

Analyses	Solve For	Syntax
TEST=DIFF	Power	POWER=.
	Sample size	NPAIRS=.
TEST=RATIO	Power	POWER=.
	Sample size	NPAIRS=.
TEST=EQUIV_DIFF	Power	POWER=.
	Sample size	NPAIRS=.
TEST=EQUIV_RATIO	Power	POWER=.
	Sample size	NPAIRS=.
CI=DIFF	Prob(width)	PROBWIDTH=.
	Sample size	NPAIRS=.

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual $0.05 \times 100\% = 5\%$ level of significance. If the CI= and SIDES=1 options are used,

then the value must be less than 0.5. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

CI

CI=DIFF

specifies an analysis of precision of the confidence interval for the mean difference. Instead of power, the relevant probability for this analysis is the probability of achieving a desired precision. Specifically, it is the probability that the half-width of the observed confidence interval will be at most the value specified by the HALFWIDTH= option. If neither the CI= option nor the TEST= option is used, the default is TEST=DIFF.

CORR=*number-list*

specifies the correlation between members of a pair. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

CV=*number-list*

specifies the coefficient of variation assumed to be common to both members of a pair. The coefficient of variation is defined as the ratio of the standard deviation to the mean. You can use this option only with DIST=LOGNORMAL. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

DIST=LOGNORMAL

DIST=NORMAL

specifies the underlying distribution assumed for the test statistic. NORMAL corresponds the normal distribution, and LOGNORMAL corresponds to the lognormal distribution. The default value (also the only acceptable value in each case) is NORMAL for TEST=DIFF, TEST=EQUIV_DIFF, and CI=DIFF; and LOGNORMAL for TEST=RATIO and TEST=EQUIV_RATIO.

HALFWIDTH=*number-list*

specifies the desired confidence interval half-width. The half-width is defined as the distance between the point estimate and a finite endpoint. This option can be used only with the CI=DIFF analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

LOWER=*number-list*

specifies the lower equivalence bound for the mean difference or mean ratio, in the original scale (whether DIST=NORMAL or DIST=LOGNORMAL). This option can be used only with the TEST=EQUIV_DIFF and TEST=EQUIV_RATIO analyses. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

MEANDIFF=*number-list*

specifies the mean difference, defined as the mean of the difference between the second and first members of a pair, $\mu_2 - \mu_1$. This option can be used only with the TEST=DIFF and TEST=EQUIV_DIFF analyses. When TEST=EQUIV_DIFF, the mean difference is interpreted as the treatment mean minus the reference mean. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

MEANRATIO=*number-list*

specifies the geometric mean ratio, defined as γ_2/γ_1. This option can be used only with the TEST=RATIO and TEST=EQUIV_RATIO analyses. When TEST=EQUIV_RATIO, the mean ratio is interpreted as the treatment mean divided by the reference mean. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NFRACTIONAL

NFRAC

enables fractional input and output for sample sizes. See the section "Sample Size Adjustment Options" on page 4922 for information about the ramifications of the presence (and absence) of the NFRACTIONAL option.

NPAIRS=*number-list*

specifies the number of pairs or requests a solution for the number of pairs with a missing value (NPAIRS=.). See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NULLDIFF=*number-list*

NULLD=*number-list*

specifies the null mean difference. The default value is 0. This option can be used only with the TEST=DIFF analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NULLRATIO=*number-list*

NULLR=*number-list*

specifies the null mean ratio. The default value is 1. This option can be used only with the TEST=RATIO analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

OUTPUTORDER=INTERNAL

OUTPUTORDER=REVERSE

OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) arranges the parameters in the output according to the following order of their corresponding options:

- SIDES=
- NULLDIFF=
- NULLRATIO=
- LOWER=
- UPPER=
- ALPHA=
- PAIREDMEANS=
- MEANDIFF=
- MEANRATIO=

- HALFWIDTH=
- STDDEV=
- PAIREDSTDDEVS=
- CV=
- PAIREDCVS=
- CORR=
- NPAIRS=
- POWER=
- PROBTYPE=
- PROBWIDTH=

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order in which their corresponding options are specified in the PAIREDMEANS statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order in which their corresponding options are specified in the PAIREDMEANS statement.

PAIREDCVS=*grouped-number-list*

specifies the coefficient of variation for each member of a pair. Unlike the CV= option, the PAIREDCVS= option supports different values for each member of a pair. Values must be nonnegative (unless both are equal to zero, which is permitted). This option can be used only with DIST=LOGNORMAL. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

PAIREDMEANS=*grouped-number-list*

PMEANS=*grouped-number-list*

specifies the two paired means, in the original scale. The means are arithmetic if DIST=NORMAL and geometric if DIST=LOGNORMAL. This option cannot be used with the CI=DIFF analysis. When TEST=EQUIV_DIFF, the means are interpreted as the reference mean (first) and the treatment mean (second). See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

PAIREDSTDDEVS=*grouped-number-list*

PAIREDSTDS=*grouped-number-list*

PSTDDEVS=*grouped-number-list*

PSTDS=*grouped-number-list*

specifies the standard deviation of each member of a pair. Unlike the STDDEV= option, the PAIREDSTDDEVS= option supports different values for each member of a pair. This option can be used only with DIST=NORMAL. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

POWER=*number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1,

rather than as a percentage. This option cannot be used with the CI=DIFF analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

PROBTYPE=*keyword-list*

specifies the type of probability for the PROBWIDTH= option. A value of CONDITIONAL (the default) indicates the conditional probability that the confidence interval half-width is at most the value specified by the HALFWIDTH= option, given that the true mean difference is captured by the confidence interval. A value of UNCONDITIONAL indicates the unconditional probability that the confidence interval half-width is at most the value specified by the HALFWIDTH= option. you can use the alias GIVENVALIDITY for CONDITIONAL. The PROBTYPE= option can be used only with the CI=DIFF analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *keyword-list*.

CONDITIONAL width probability conditional on interval containing the mean

UNCONDITIONAL unconditional width probability

PROBWIDTH=*number-list*

specifies the desired probability of obtaining a confidence interval half-width less than or equal to the value specified by the HALFWIDTH= option. A missing value (PROBWIDTH=.) requests a solution for this probability. The type of probability is controlled with the PROBTYPE= option. Values are expressed as probabilities (for example, 0.9) rather than percentages. This option can be used only with the CI=DIFF analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and the direction of the statistical test or confidence interval. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *keyword-list*. Valid keywords and their interpretation for the TEST= analyses are as follows:

1 one-sided with alternative hypothesis in same direction as effect

2 two-sided

U upper one-sided with alternative greater than null value

L lower one-sided with alternative less than null value

For confidence intervals, SIDES=U refers to an interval between the lower confidence limit and infinity, and SIDES=L refers to an interval between minus infinity and the upper confidence limit. For both of these cases and SIDES=1, the confidence interval computations are equivalent. The SIDES= option cannot be used with the TEST=EQUIV_DIFF and TEST=EQUIV_RATIO analyses. The default value is 2.

STDDEV=*number-list*

STD=*number-list*

specifies the standard deviation assumed to be common to both members of a pair. This option

can be used only with DIST=NORMAL. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

TEST
TEST=DIFF
TEST=EQUIV_DIFF
TEST=EQUIV_RATIO
TEST=RATIO

specifies the statistical analysis. TEST or TEST=DIFF (the default) specifies a paired t test on the mean difference. TEST=EQUIV_DIFF specifies an additive equivalence test of the mean difference by using a two one-sided tests (TOST) analysis (Schuirmann 1987). TEST=EQUIV_RATIO specifies a multiplicative equivalence test of the mean ratio by using a TOST analysis. TEST=RATIO specifies a paired t test on the geometric mean ratio. If neither the TEST= option nor the CI= option is used, the default is TEST=DIFF.

UPPER=*number-list*

specifies the upper equivalence bound for the mean difference or mean ratio, in the original scale (whether DIST=NORMAL or DIST=LOGNORMAL). This option can be used only with the TEST=EQUIV_DIFF and TEST=EQUIV_RATIO analyses. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

Restrictions on Option Combinations

To define the analysis, choose one of the following parameterizations:

- a statistical test (by using the TEST= option)
- confidence interval precision (by using the CI= option)

To specify the means, choose one of the following parameterizations:

- individual means (by using the PAIREDMEANS= option)
- mean difference (by using the MEANDIFF= option)
- mean ratio (by using the MEANRATIO= option)

To specify the coefficient of variation, choose one of the following parameterizations:

- common coefficient of variation (by using the CV= option)
- individual coefficients of variation (by using the PAIREDCVS= option)

To specify the standard deviation, choose one of the following parameterizations:

- common standard deviation (by using the STDDEV= option)
- individual standard deviations (by using the PAIREDSTDDEVS= option)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the PAIREDMEANS statement.

Paired t *Test*

You can express effects in terms of the mean difference and variability in terms of a correlation and common standard deviation, as in the following statements. Default values for the DIST=, SIDES=, NULLDIFF=, and ALPHA= options specify a two-sided test for no difference with a normal distribution and a significance level of 0.05.

```
proc power;
   pairedmeans test=diff
      meandiff = 7
      corr = 0.4
      stddev = 12
      npairs = 50
      power = .;
run;
```

You can also express effects in terms of individual means and variability in terms of correlation and individual standard deviations:

```
proc power;
   pairedmeans test=diff
      pairedmeans = 8 | 15
      corr = 0.4
      pairedstddevs = (7 12)
      npairs = .
      power = 0.9;
run;
```

Paired t *Test of Mean Ratio with Lognormal Data*

You can express variability in terms of correlation and a common coefficient of variation, as in the following statements. Defaults for the DIST=, SIDES=, NULLRATIO= and ALPHA= options specify a two-sided test of mean ratio = 1 assuming a lognormal distribution and a significance level of 0.05.

```
proc power;
   pairedmeans test=ratio
      meanratio = 7
      corr = 0.3
      cv = 1.2
      npairs = 30
      power = .;
run;
```

You can also express variability in terms of correlation and individual coefficients of variation:

```
proc power;
   pairedmeans test=ratio
      meanratio = 7
      corr = 0.3
      pairedcvs = 0.8 | 0.9
      npairs = 30
      power = .;
run;
```

Additive Equivalence Test for Mean Difference with Normal Data

The following statements demonstrate a sample size computation for a TOST equivalence test for a normal mean difference. Default values for the DIST= and ALPHA= options specify a normal distribution and a significance level of 0.05.

```
proc power;
   pairedmeans test=equiv_diff
      lower = 2
      upper = 5
      meandiff = 4
      corr = 0.2
      stddev = 8
      npairs = .
      power = 0.9;
run;
```

Multiplicative Equivalence Test for Mean Ratio with Lognormal Data

The following statements demonstrate a power computation for a TOST equivalence test for a lognormal mean ratio. Default values for the DIST= and ALPHA= options specify a lognormal distribution and a significance level of 0.05.

```
proc power;
   pairedmeans test=equiv_ratio
      lower = 3
      upper = 7
      meanratio = 5
      corr = 0.2
      cv = 1.1
      npairs = 50
      power = .;
run;
```

Confidence Interval for Mean Difference

By default CI=DIFF analyzes the conditional probability of obtaining the desired precision, given that the interval contains the true mean difference, as in the following statements. The defaults of SIDES=2 and ALPHA=0.05 specify a two-sided interval with a confidence level of 0.95.

```
proc power;
   pairedmeans ci = diff
      halfwidth = 4
      corr = 0.35
      stddev = 8
      npairs = 30
      probwidth = .;
run;
```

PLOT Statement

PLOT < *plot-options* > < / *graph-options* > **;**

The PLOT statement produces a graph or set of graphs for the sample size analysis defined by the previous analysis statement. The *plot-options* define the plot characteristics, and the *graph-options* are SAS/GRAPH-style options.

Options

You can specify the following *plot-options* in the PLOT statement.

INTERPOL=JOIN | NONE
specifies the type of curve to draw through the computed points. The INTERPOL=JOIN option connects computed points by straight lines. The INTERPOL=NONE option leaves computed points unconnected.

KEY=BYCURVE < (*bycurve-options*) >

KEY=BYFEATURE < (*byfeature-options*) >

KEY=ONCURVES
specifies the style of key (or "legend") for the plot. The default is KEY=BYFEATURE, which specifies a key with a column of entries for each plot feature (line style, color, and/or symbol). Each entry shows the mapping between a value of the feature and the value(s) of the analysis parameter(s) linked to that feature. The KEY=BYCURVE option specifies a key with each row identifying a distinct curve in the plot. The KEY=ONCURVES option places a curve-specific label adjacent to each curve.

You can specify the following *byfeature-options* in parentheses after the KEY=BYCURVE option.

NUMBERS=OFF | ON
specifies how the key should identify curves. If NUMBERS=OFF, then the key includes symbol, color, and line style samples to identify the curves. If NUMBERS=ON, then the key includes numbers matching numeric labels placed adjacent to the curves. The default is NUMBERS=ON.

POS=BOTTOM | INSET
specifies the position of the key. The POS=BOTTOM option places the key below the X axis. The POS=INSET option places the key inside the plotting region and attempts to choose the least crowded corner. The default is POS=BOTTOM.

You can specify the following *byfeature-options* in parentheses after KEY=BYFEATURE option.

POS=BOTTOM | INSET
specifies the position of the key. The POS=BOTTOM option places the key below the X axis. The POS=INSET option places the key inside the plotting region and attempts to choose the least crowded corner. The default is POS=BOTTOM.

MARKERS=ANALYSIS | COMPUTED | NICE | NONE
specifies the locations for plotting symbols.

The MARKERS=ANALYSIS option places plotting symbols at locations corresponding to the values of the relevant input parameter from the analysis statement preceding the PLOT statement.

The MARKERS=COMPUTED option (the default) places plotting symbols at the locations of actual computed points from the sample size analysis.

The MARKERS=NICE option places plotting symbols at tick mark locations (corresponding to the argument axis).

The MARKERS=NONE option disables plotting symbols.

MAX=*number* | **DATAMAX**
specifies the maximum of the range of values for the parameter associated with the "argument" axis (the axis that is *not* representing the parameter being solved for). The default is DATAMAX, which specifies the maximum value that occurs for this parameter in the analysis statement that precedes the PLOT statement.

MIN=*number* | **DATAMIN**
specifies the minimum of the range of values for the parameter associated with the "argument" axis (the axis that is *not* representing the parameter being solved for). The default is DATAMIN, which specifies the minimum value that occurs for this parameter in the analysis statement that precedes the PLOT statement.

NPOINTS=*number*

NPTS=*number*
specifies the number of values for the parameter associated with the "argument" axis (the axis that is *not* representing the parameter being solved for). You cannot use the NPOINTS= and STEP= options simultaneously. The default value for typical situations is 20.

STEP=*number*
specifies the increment between values of the parameter associated with the "argument" axis (the axis that is *not* representing the parameter being solved for). You cannot use the STEP= and NPOINTS= options simultaneously. By default, the NPOINTS= option is used instead of the STEP= option.

VARY (*feature* < BY *parameter-list* > < , . . . , *feature* < BY *parameter-list* > >)

specifies how plot features should be linked to varying analysis parameters. Available plot *features* are COLOR, LINESTYLE, PANEL, and SYMBOL. A "panel" refers to a separate plot with a heading identifying the subset of values represented in the plot.

The *parameter-list* is a list of one or more names separated by spaces. Each name must match the name of an analysis option used in the analysis statement preceding the PLOT statement. Also, the name must be the *primary* name for the analysis option—that is, the one listed first in the syntax description.

If you omit the < BY *parameter-list* > portion for a feature, then one or more multivalued parameters from the analysis will be automatically selected for you.

X=EFFECT N POWER

specifies a plot with the requested type of parameter on the X axis and the parameter being solved for on the Y axis. When X=EFFECT, the parameter assigned to the X axis is the one most representative of "effect size." When X=N, the parameter assigned to the X axis is the sample size. When X=POWER, the parameter assigned to the X axis is the one most representative of "power" (either power itself or a similar probability, such as Prob(Width) for confidence interval analyses). You cannot use the X= and Y= options simultaneously. The default is X=POWER, unless the result parameter is power or Prob(Width), in which case the default is X=N.

You can use the X=N option only when a scalar sample size parameter is used as input in the analysis. For example, X=N can be used with total sample size or sample size per group, or with two group sample sizes when one is being solved for.

Table 67.18 summarizes the parameters representing effect size in different analyses.

Table 67.18 Effect Size Parameters for Different Analyses

Analysis Statement and Options	Effect Size Parameters
LOGISTIC	None
MULTREG	Partial correlation or R^2 difference
ONECORR	Correlation
ONESAMPLEFREQ TEST	Proportion
ONESAMPLEFREQ CI	CI half-width
ONESAMPLEMEANS TEST=T, ONESAMPLEMEANS TEST=EQUIV	Mean
ONESAMPLEMEANS CI=T	CI half-width
ONEWAYANOVA	None
PAIREDFREQ	Discordant proportion difference or ratio
PAIREDMEANS TEST=DIFF, PAIREDMEANS TEST=EQUIV_DIFF	Mean difference

Table 67.18 *continued*

Analysis Statement and Options	Effect Size Parameters
PAIREDMEANS TEST=RATIO, PAIREDMEANS TEST=EQUIV_RATIO	Mean ratio
PAIREDMEANS CI=DIFF	CI half-width
TWOSAMPLEFREQ	Proportion difference, odds ratio, or relative risk
TWOSAMPLEMEANS TEST=DIFF, TWOSAMPLEMEANS TEST=DIFF_SATT, TWOSAMPLEMEANS TEST=EQUIV_DIFF	Mean difference
TWOSAMPLEMEANS TEST=RATIO, TWOSAMPLEMEANS TEST=EQUIV_RATIO	Mean ratio
TWOSAMPLEMEANS CI=DIFF	CI half-width
TWOSAMPLESURVIVAL	Hazard ratio if used, else none
TWOSAMPLEWILCOXON	None

XOPTS=(*x-options* **)**
specifies plot characteristics pertaining to the X axis.

You can specify the following *x-options* in parentheses.

CROSSREF=NO | YES
specifies whether the reference lines defined by the REF= *x-option* should be crossed with a reference line on the Y axis that indicates the solution point on the curve.

REF=*number-list*
specifies locations for reference lines extending from the X axis across the entire plotting region. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

Y=EFFECT N POWER
specifies a plot with the requested type of parameter on the Y axis and the parameter being solved for on the X axis. When Y=EFFECT, the parameter assigned to the Y axis is the one most representative of "effect size." When Y=N, the parameter assigned to the Y axis is the sample size. When Y=POWER, the parameter assigned to the Y axis is the one most representative of "power" (either power itself or a similar probability, such as Prob(Width) for confidence interval analyses). You cannot use the Y= and X= options simultaneously. By default, the X= option is used instead of the Y= option.

YOPTS=(*y-options* **)**
specifies plot characteristics pertaining to the Y axis.

You can specify the following *y-options* in parentheses.

CROSSREF=NO | YES
specifies whether the reference lines defined by the REF= *y-option* should be crossed with a reference line on the X axis that indicates the solution point on the curve.

REF=*number-list*
specifies locations for reference lines extending from the Y axis across the entire plotting region. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

You can specify the following *graph-options* in the PLOT statement after a slash (/).

DESCRIPTION='*string* '
specifies a descriptive string of up to 40 characters that appears in the "Description" field of the graphics catalog. The description does not appear on the plots. By default, PROC POWER assigns a description either of the form "*Y* versus *X*" (for a single-panel plot) or of the form "*Y* versus *X* (*S*)," where *Y* is the parameter on the Y axis, *X* is the parameter on the X axis, and *S* is a description of the subset represented on the current panel of a multipanel plot.

NAME='*string* '
specifies a name of up to eight characters for the catalog entry for the plot. The default name is PLOT*n*, where *n* is the number of the plot statement within the current invocation of PROC POWER. If the name duplicates the name of an existing entry, SAS/GRAPH software adds a number to the duplicate name to create a unique entry—for example, PLOT11 and PLOT12 for the second and third panels of a multipanel plot generated in the first PLOT statement in an invocation of PROC POWER.

TWOSAMPLEFREQ Statement

TWOSAMPLEFREQ < *options* > **;**

The TWOSAMPLEFREQ statement performs power and sample size analyses for tests of two independent proportions. Pearson's chi-square, Fisher's exact, and likelihood ratio chi-square tests are supported.

Summary of Options

Table 67.19 summarizes categories of options available in the TWOSAMPLEFREQ statement.

Table 67.19 Summary of Options in the TWOSAMPLEFREQ Statement

Task	Options
Define analysis	TEST=
Specify analysis information	ALPHA= NULLPROPORTIONDIFF=

Table 67.19 *continued*

Task	Options
	NULLODDSRATIO= NULLRELATIVERISK= SIDES=
Specify effects	GROUPPROPORTIONS= ODDSRATIO= PROPORTIONDIFF= REFPROPORTION= RELATIVERISK=
Specify sample size and allocation	GROUPNS= GROUPWEIGHTS= NPERGROUP= NTOTAL=
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Control ordering in output	OUTPUTORDER=

Table 67.20 summarizes the valid result parameters for different analyses in the TWOSAMPLEFREQ statement.

Table 67.20 Summary of Result Parameters in the TWOSAMPLEFREQ Statement

Analyses	Solve For	Syntax
TEST=FISHER	Power	POWER=.
	Sample size	NTOTAL=.
		NPERGROUP=.
TEST=LRCHI	Power	POWER=.
	Sample size	NTOTAL=.
		NPERGROUP=.
TEST=PCHI	Power	POWER=.
	Sample size	NTOTAL=.
		NPERGROUP=.

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual $0.05 \times 100\% = 5\%$ level of significance. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

GROUPPROPORTIONS=*grouped-number-list*

GPROPORTIONS=*grouped-number-list*

GROUPPS=*grouped-number-list*

GPS=*grouped-number-list*

specifies the two independent proportions, p_1 and p_2. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

GROUPNS=*grouped-number-list*

GNS=*grouped-number-list*

specifies the two group sample sizes. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

GROUPWEIGHTS=*grouped-number-list*

GWEIGHTS=*grouped-number-list*

specifies the sample size allocation weights for the two groups. This option controls how the total sample size is divided between the two groups. Each pair of values for the two groups represents relative allocation weights. Additionally, if the NFRACTIONAL option is not used, the total sample size is restricted to be equal to a multiple of the sum of the two group weights (so that the resulting design has an integer sample size for each group while adhering exactly to the group allocation weights). Values must be integers unless the NFRACTIONAL option is used. The default value is (1 1), a balanced design with a weight of 1 for each group. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

NFRACTIONAL

NFRAC

enables fractional input and output for sample sizes. See the section "Sample Size Adjustment Options" on page 4922 for information about the ramifications of the presence (and absence) of the NFRACTIONAL option.

NPERGROUP=*number-list*

NPERG=*number-list*

specifies the common sample size per group or requests a solution for the common sample size per group with a missing value (NPERGROUP=.). Use of this option implicitly specifies a balanced design. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NTOTAL=*number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NULLODDSRATIO=*number-list*

NULLOR=*number-list*

specifies the null odds ratio. The default value is 1. This option can be used only with the ODDSRATIO= option in the TEST=PCHI analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NULLPROPORTIONDIFF=*number-list*

NULLPDIFF=*number-list*

specifies the null proportion difference. The default value is 0. This option can be used only with the GROUPPROPORTIONS= or PROPORTIONDIFF= option in the TEST=PCHI analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NULLRELATIVERISK=*number-list*

NULLRR=*number-list*

specifies the null relative risk. The default value is 1. This option can be used only with the RELATIVERISK= option in the TEST=PCHI analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

ODDSRATIO=*number-list*

OR=*number-list*

specifies the odds ratio $[p_2/(1-p_2)]\,/\,[p_1/(1-p_1)]$. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

OUTPUTORDER=INTERNAL

OUTPUTORDER=REVERSE

OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) arranges the parameters in the output according to the following order of their corresponding options:

- SIDES=
- NULLPROPORTIONDIFF=
- NULLODDSRATIO=
- NULLRELATIVERISK=
- ALPHA=
- GROUPPROPORTIONS=
- REFPROPORTION=
- PROPORTIONDIFF=
- ODDSRATIO=
- RELATIVERISK=
- GROUPWEIGHTS=
- NTOTAL=
- NPERGROUP=
- GROUPNS=
- POWER=

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order in which their corresponding options are specified in the TWOSAMPLEFREQ statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order in which their corresponding options are specified in the TWOSAMPLEFREQ statement.

POWER=*number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

PROPORTIONDIFF=*number-list*

PDIFF=*number-list*

specifies the proportion difference $p_2 - p_1$. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

REFPROPORTION=*number-list*

REFP=*number-list*

specifies the reference proportion p_1. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

RELATIVERISK=*number-list*

RR=*number-list*

specifies the relative risk p_2/p_1. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and the direction of the statistical test or confidence interval. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *keyword-list*. Valid keywords and their interpretation are as follows:

1	one-sided with alternative hypothesis in same direction as effect
2	two-sided
U	upper one-sided with alternative greater than null value
L	lower one-sided with alternative less than null value

If the effect size is zero, then SIDES=1 is not permitted; instead, specify the direction of the test explicitly in this case with either SIDES=L or SIDES=U. The default value is 2.

TEST=FISHER

TEST=LRCHI

TEST=PCHI

specifies the statistical analysis. TEST=FISHER specifies Fisher's exact test. TEST=LRCHI specifies the likelihood ratio chi-square test. TEST=PCHI (the default) specifies Pearson's chi-square test.

Restrictions on Option Combinations

To specify the proportions, choose one of the following parameterizations:

- individual proportions (by using the GROUPPROPORTIONS= option)
- difference between proportions and reference proportion (by using the PROPORTIONDIFF= and REFPROPORTION= options)
- odds ratio and reference proportion (by using the ODDSRATIO= and REFPROPORTION= options)
- relative risk and reference proportion (by using the RELATIVERISK= and REFPROPORTION= options)

To specify the sample size and allocation, choose one of the following parameterizations:

- sample size per group in a balanced design (by using the NPERGROUP= option)
- total sample size and allocation weights (by using the NTOTAL= and GROUPWEIGHTS= options)
- individual group sample sizes (by using the GROUPNS= option)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the TWOSAMPLEFREQ statement.

Pearson Chi-Square Test for Two Proportions

You can use the NPERGROUP= option in a balanced design and express effects in terms of the individual proportions, as in the following statements. Default values for the SIDES= and ALPHA= options specify a two-sided test with a significance level of 0.05.

```
proc power;
   twosamplefreq test=pchi
      groupproportions = (.15 .25)
      nullproportiondiff = .03
      npergroup = 50
      power = .;
run;
```

You can also specify an unbalanced design by using the NTOTAL= and GROUPWEIGHTS= options and express effects in terms of the odds ratio. The default value of the NULLODDSRATIO= option specifies a test of no effect.

```
proc power;
   twosamplefreq test=pchi
      oddsratio = 2.5
      refproportion = 0.3
      groupweights = (1 2)
      ntotal = .
      power = 0.8;
run;
```

You can also specify sample sizes with the GROUPNS= option and express effects in terms of relative risks. The default value of the NULLRELATIVERISK= option specifies a test of no effect.

```
proc power;
   twosamplefreq test=pchi
      relativerisk = 1.5
      refproportion = 0.2
      groupns = 40 | 60
      power = .;
run;
```

You can also express effects in terms of the proportion difference. The default value of the NULLPROPORTIONDIFF= option specifies a test of no effect, and the default value of the GROUPWEIGHTS= option specifies a balanced design.

```
proc power;
   twosamplefreq test=pchi
      proportiondiff = 0.15
      refproportion = 0.4
      ntotal = 100
      power = .;
run;
```

Fisher's Exact Conditional Test for Two Proportions

The following statements demonstrate a power computation for Fisher's exact conditional test for two proportions. Default values for the SIDES= and ALPHA= options specify a two-sided test with a significance level of 0.05.

```
proc power;
   twosamplefreq test=fisher
      grouppproportions = (.35 .15)
      npergroup = 50
      power = .;
run;
```

Likelihood Ratio Chi-Square Test for Two Proportions

The following statements demonstrate a sample size computation for the likelihood ratio chi-square test for two proportions. Default values for the SIDES= and ALPHA= options specify a two-sided test with a significance level of 0.05.

```
proc power;
   twosamplefreq test=lrchi
      oddsratio = 2
      refproportion = 0.4
      npergroup = .
      power = 0.9;
run;
```

TWOSAMPLEMEANS Statement

TWOSAMPLEMEANS < *options* > **;**

The TWOSAMPLEMEANS statement performs power and sample size analyses for pooled and unpooled *t* tests, equivalence tests, and confidence interval precision involving two independent samples.

Summary of Options

Table 67.21 summarizes categories of options available in the TWOSAMPLEMEANS statement.

Table 67.21 Summary of Options in the TWOSAMPLEMEANS Statement

Task	Options
Define analysis	CI= DIST= TEST=
Specify analysis information	ALPHA= LOWER= NULLDIFF= NULLRATIO= SIDES= UPPER=
Specify effects	HALFWIDTH= GROUPMEANS= MEANDIFF= MEANRATIO=
Specify variability	CV= GROUPSTDDEVS== STDDEV=
Specify sample size and allocation	GROUPNS= GROUPWEIGHTS= NPERGROUP= NTOTAL=
Specify power and related probabilities	POWER= PROBTYPE= PROBWIDTH=
Control sample size rounding	NFRACTIONAL
Control ordering in output	OUTPUTORDER=

Table 67.22 summarizes the valid result parameters for different analyses in the TWOSAMPLEMEANS statement.

Table 67.22 Summary of Result Parameters in the TWOSAMPLEMEANS Statement

Analyses	Solve For	Syntax
TEST=DIFF	Power	POWER=.
	Sample size	NTOTAL=.
		NPERGROUP=.
	Group sample size	GROUPNS= *n1* | .
		GROUPNS=. | *n2*
		GROUPNS= (*n1* .)
		GROUPNS= (. *n2*)
	Group weight	GROUPWEIGHTS= *w1* | .
		GROUPWEIGHTS=. | *w2*
		GROUPWEIGHTS= (*w1* .)
		GROUPWEIGHTS= (. *w2*)
	Alpha	ALPHA=.
	Group mean	GROUPMEANS= *mean1* | .
		GROUPMEANS=. | *mean2*
		GROUPMEANS= (*mean1* .)
		GROUPMEANS= (. *mean2*)
	Mean difference	MEANDIFF=.
	Standard deviation	STDDEV=.
TEST=DIFF_SATT	Power	POWER=.
	Sample size	NTOTAL=.
		NPERGROUP=.
TEST=RATIO	Power	POWER=.
	Sample size	NTOTAL=.
		NPERGROUP=.
TEST=EQUIV_DIFF	Power	POWER=.
	Sample size	NTOTAL=.
		NPERGROUP=.
TEST=EQUIV_RATIO	Power	POWER=.
	Sample size	NTOTAL=.
		NPERGROUP=.
CI=DIFF	Prob(width)	PROBWIDTH=.
	Sample size	NTOTAL=.
		NPERGROUP=.

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test or requests a solution for alpha with a missing value (ALPHA=.). The default is 0.05, corresponding to the usual 0.05 × 100% = 5% level of significance. If the CI= and SIDES=1 options are used, then the value must be less than 0.5. See the section "Specifying Value Lists in Analysis Statements" on page 4919

for information about specifying the *number-list*.

CI

CI=DIFF

specifies an analysis of precision of the confidence interval for the mean difference, assuming equal variances. Instead of power, the relevant probability for this analysis is the probability that the interval half-width is at most the value specified by the HALFWIDTH= option. If neither the TEST= option nor the CI= option is used, the default is TEST=DIFF.

CV=*number-list*

specifies the coefficient of variation assumed to be common to both groups. The coefficient of variation is defined as the ratio of the standard deviation to the mean. You can use this option only with DIST=LOGNORMAL. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

DIST=LOGNORMAL

DIST=NORMAL

specifies the underlying distribution assumed for the test statistic. NORMAL corresponds the normal distribution, and LOGNORMAL corresponds to the lognormal distribution. The default value (also the only acceptable value in each case) is NORMAL for TEST=DIFF, TEST=DIFF_SATT, TEST=EQUIV_DIFF, and CI=DIFF; and LOGNORMAL for TEST=RATIO and TEST=EQUIV_RATIO.

GROUPMEANS=*grouped-number-list*

GMEANS=*grouped-number-list*

specifies the two group means or requests a solution for one group mean given the other. Means are in the original scale. They are arithmetic if DIST=NORMAL and geometric if DIST=LOGNORMAL. This option cannot be used with the CI=DIFF analysis. When TEST=EQUIV_DIFF, the means are interpreted as the reference mean (first) and the treatment mean (second). See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

GROUPNS=*grouped-number-list*

GNS=*grouped-number-list*

specifies the two group sample sizes or requests a solution for one group sample size given the other. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

GROUPSTDDEVS=*grouped-number-list*

GSTDDEVS=*grouped-number-list*

GROUPSTDS=*grouped-number-list*

GSTDS=*grouped-number-list*

specifies the standard deviation of each group. Unlike the STDDEV= option, the GROUPSTDDEVS== option supports different values for each group. It is valid only for the Satterthwaite *t* test (TEST=DIFF_SATT DIST=NORMAL). See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

GROUPWEIGHTS=*grouped-number-list*

GWEIGHTS=*grouped-number-list*

specifies the sample size allocation weights for the two groups, or requests a solution for one group weight given the other. This option controls how the total sample size is divided between the two groups. Each pair of values for the two groups represents relative allocation weights. Additionally, if the NFRACTIONAL option is not used, the total sample size is restricted to be equal to a multiple of the sum of the two group weights (so that the resulting design has an integer sample size for each group while adhering exactly to the group allocation weights). Values must be integers unless the NFRACTIONAL option is used. The default value is (1 1), a balanced design with a weight of 1 for each group. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

HALFWIDTH=*number-list*

specifies the desired confidence interval half-width. The half-width is defined as the distance between the point estimate and a finite endpoint. This option can be used only with the CI=DIFF analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

LOWER=*number-list*

specifies the lower equivalence bound for the mean difference or mean ratio, in the original scale (whether DIST=NORMAL or DIST=LOGNORMAL). Values must be greater than 0 when DIST=LOGNORMAL. This option can be used only with the TEST=EQUIV_DIFF and TEST=EQUIV_RATIO analyses. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

MEANDIFF=*number-list*

specifies the mean difference, defined as $\mu_2 - \mu_1$, or requests a solution for the mean difference with a missing value (MEANDIFF=.). This option can be used only with the TEST=DIFF, TEST=DIFF_SATT, and TEST=EQUIV_DIFF analyses. When TEST=EQUIV_DIFF, the mean difference is interpreted as the treatment mean minus the reference mean. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

MEANRATIO=*number-list*

specifies the geometric mean ratio, defined as γ_2/γ_1. This option can be used only with the TEST=RATIO and TEST=EQUIV_RATIO analyses. When TEST=EQUIV_RATIO, the mean ratio is interpreted as the treatment mean divided by the reference mean. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NFRACTIONAL

NFRAC

enables fractional input and output for sample sizes. See the section "Sample Size Adjustment Options" on page 4922 for information about the ramifications of the presence (and absence) of the NFRACTIONAL option.

NPERGROUP=*number-list*

NPERG=*number-list*

specifies the common sample size per group or requests a solution for the common sample size per group with a missing value (NPERGROUP=.). Use of this option implicitly specifies a balanced design. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NTOTAL=*number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NULLDIFF=*number-list*

NULLD=*number-list*

specifies the null mean difference. The default value is 0. This option can be used only with the TEST=DIFF and TEST=DIFF_SATT analyses. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NULLRATIO=*number-list*

NULLR=*number-list*

specifies the null mean ratio. The default value is 1. This option can be used only with the TEST=RATIO analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

OUTPUTORDER=INTERNAL

OUTPUTORDER=REVERSE

OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) arranges the parameters in the output according to the following order of their corresponding options:

- SIDES=
- NULLDIFF=
- NULLRATIO=
- LOWER=
- UPPER=
- ALPHA=
- GROUPMEANS=
- MEANDIFF=
- MEANRATIO=
- HALFWIDTH=
- STDDEV=
- GROUPSTDDEVS==
- CV=

- GROUPWEIGHTS=
- NTOTAL=
- NPERGROUP=
- GROUPNS=
- POWER=
- PROBTYPE=
- PROBWIDTH=

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order in which their corresponding options are specified in the TWOSAMPLEMEANS statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order in which their corresponding options are specified in the TWOSAMPLEMEANS statement.

POWER=*number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. This option cannot be used with the CI=DIFF analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

PROBTYPE=*keyword-list*

specifies the type of probability for the PROBWIDTH= option. A value of CONDITIONAL (the default) indicates the conditional probability that the confidence interval half-width is at most the value specified by the HALFWIDTH= option, given that the true mean difference is captured by the confidence interval. A value of UNCONDITIONAL indicates the unconditional probability that the confidence interval half-width is at most the value specified by the HALFWIDTH= option. you can use the alias GIVENVALIDITY for CONDITIONAL. The PROBTYPE= option can be used only with the CI=DIFF analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *keyword-list*.

CONDITIONAL width probability conditional on interval containing the mean

UNCONDITIONAL unconditional width probability

PROBWIDTH=*number-list*

specifies the desired probability of obtaining a confidence interval half-width less than or equal to the value specified by the HALFWIDTH= option. A missing value (PROBWIDTH=.) requests a solution for this probability. The type of probability is controlled with the PROBTYPE= option. Values are expressed as probabilities (for example, 0.9) rather than percentages. This option can be used only with the CI=DIFF analysis. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and the direction of the statistical test or confidence interval. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for

information about specifying the *keyword-list*. Valid keywords and their interpretation for the TEST= analyses are as follows:

1 one-sided with alternative hypothesis in same direction as effect

2 two-sided

U upper one-sided with alternative greater than null value

L lower one-sided with alternative less than null value

For confidence intervals, SIDES=U refers to an interval between the lower confidence limit and infinity, and SIDES=L refers to an interval between minus infinity and the upper confidence limit. For both of these cases and SIDES=1, the confidence interval computations are equivalent. The SIDES= option cannot be used with the TEST=EQUIV_DIFF and TEST=EQUIV_RATIO analyses. The default value is 2.

STDDEV=*number-list*

STD=*number-list*

specifies the standard deviation assumed to be common to both groups, or requests a solution for the common standard deviation with a missing value (STDDEV=.). This option can be used only with DIST=NORMAL. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

TEST

TEST=DIFF

TEST=DIFF_SATT

TEST=EQUIV_DIFF

TEST=EQUIV_RATIO

TEST=RATIO

specifies the statistical analysis. TEST or TEST=DIFF (the default) specifies a pooled t test on the mean difference, assuming equal variances. TEST=DIFF_SATT specifies a Satterthwaite unpooled t test on the mean difference, assuming unequal variances. TEST=EQUIV_DIFF specifies an additive equivalence test of the mean difference by using a two one-sided tests (TOST) analysis (Schuirman 1987). TEST=EQUIV_RATIO specifies a multiplicative equivalence test of the mean ratio by using a TOST analysis. TEST=RATIO specifies a pooled t test on the mean ratio, assuming equal coefficients of variation. If neither the TEST= option nor the CI= option is used, the default is TEST=DIFF.

UPPER=*number-list*

specifies the upper equivalence bound for the mean difference or mean ratio, in the original scale (whether DIST=NORMAL or DIST=LOGNORMAL). This option can be used only with the TEST=EQUIV_DIFF and TEST=EQUIV_RATIO analyses. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

Restrictions on Option Combinations

To define the analysis, choose one of the following parameterizations:

- a statistical test (by using the TEST= option)
- confidence interval precision (by using the CI= option)

To specify the means, choose one of the following parameterizations:

- individual group means (by using the GROUPMEANS= option)
- mean difference (by using the MEANDIFF= option)
- mean ratio (by using the MEANRATIO= option)

To specify standard deviations in the Satterthwaite *t* test (TEST=DIFF_SATT), choose one of the following parameterizations:

- common standard deviation (by using the STDDEV= option)
- individual group standard deviations (by using the GROUPSTDDEVS== option)

To specify the sample sizes and allocation, choose one of the following parameterizations:

- sample size per group in a balanced design (by using the NPERGROUP= option)
- total sample size and allocation weights (by using the NTOTAL= and GROUPWEIGHTS= options)
- individual group sample sizes (by using the GROUPNS= option)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the TWOSAMPLEMEANS statement.

Two-Sample t *Test Assuming Equal Variances*

You can use the NPERGROUP= option in a balanced design and express effects in terms of the mean difference, as in the following statements. Default values for the DIST=, SIDES=, NULLDIFF=, and ALPHA= options specify a two-sided test for no difference with a normal distribution and a significance level of 0.05.

```
proc power;
   twosamplemeans test=diff
      meandiff = 7
      stddev = 12
      npergroup = 50
      power = .;
run;
```

You can also specify an unbalanced design by using the NTOTAL= and GROUPWEIGHTS= options and express effects in terms of individual group means:

```
proc power;
   twosamplemeans test=diff
      groupmeans = 8 | 15
      stddev = 4
      groupweights = (2 3)
      ntotal = .
      power = 0.9;
run;
```

Another way to specify the sample sizes is with the GROUPNS= option:

```
proc power;
   twosamplemeans test=diff
      groupmeans = 8 | 15
      stddev = 4
      groupns = (25 40)
      power = .;
run;
```

Two-Sample Satterthwaite t Test Assuming Unequal Variances

The following statements demonstrate a power computation for the two-sample Satterthwaite *t* test allowing unequal variances. Default values for the DIST=, SIDES=, NULLDIFF=, and ALPHA= options specify a two-sided test for no difference with a normal distribution and a significance level of 0.05.

```
proc power;
   twosamplemeans test=diff_satt
      meandiff = 3
      groupstddevs = 5 | 8
      groupweights = (1 2)
      ntotal = 60
      power = .;
run;
```

Two-Sample Pooled t Test of Mean Ratio with Lognormal Data

The following statements demonstrate a power computation for the pooled *t* test of a lognormal mean ratio. Default values for the DIST=, SIDES=, NULLRATIO=, and ALPHA= options specify a two-sided test of mean ratio = 1 assuming a lognormal distribution and a significance level of 0.05.

```
proc power;
   twosamplemeans test=ratio
      meanratio = 7
      cv = 0.8
      groupns = 50 | 70
      power = .;
run;
```

Additive Equivalence Test for Mean Difference with Normal Data

The following statements demonstrate a sample size computation for the TOST equivalence test for a normal mean difference. A default value of GROUPWEIGHTS=(1 1) specifies a balanced design. Default values for the DIST= and ALPHA= options specify a significance level of 0.05 and an assumption of normally distributed data.

```
proc power;
   twosamplemeans test=equiv_diff
      lower = 2
      upper = 5
      meandiff = 4
      stddev = 8
      ntotal = .
      power = 0.9;
run;
```

Multiplicative Equivalence Test for Mean Ratio with Lognormal Data

The following statements demonstrate a power computation for the TOST equivalence test for a lognormal mean ratio. Default values for the DIST= and ALPHA= options specify a significance level of 0.05 and an assumption of lognormally distributed data.

```
proc power;
   twosamplemeans test=equiv_ratio
      lower = 3
      upper = 7
      meanratio = 5
      cv = 0.75
      npergroup = 50
      power = .;
run;
```

Confidence Interval for Mean Difference

By default CI=DIFF analyzes the conditional probability of obtaining the desired precision, given that the interval contains the true mean difference, as in the following statements. The defaults of SIDES=2 and ALPHA=0.05 specify a two-sided interval with a confidence level of 0.95.

```
proc power;
   twosamplemeans ci = diff
      halfwidth = 4
      stddev = 8
      groupns = (30 35)
      probwidth = .;
run;
```

TWOSAMPLESURVIVAL Statement

TWOSAMPLESURVIVAL < *options* > **;**

The TWOSAMPLESURVIVAL statement performs power and sample size analyses for comparing two survival curves. The log-rank, Gehan, and Tarone-Ware rank tests are supported.

Summary of Options

Table 67.23 summarizes categories of options available in the TWOSAMPLESURVIVAL statement.

Table 67.23 Summary of Options in the TWOSAMPLESURVIVAL Statement

Task	Options
Define analysis	TEST=
Specify analysis information	ALPHA= ACCRUALTIME= FOLLOWUPTIME= TOTALTIME= SIDES=
Specify effects	CURVE= GROUPMEDSURVTIMES= GROUPSURVEXPHAZARDS= GROUPSURVIVAL= HAZARDRATIO= REFSURVEXPHAZARD= REFSURVIVAL=
Specify loss information	GROUPLOSS= GROUPLOSSEXPHAZARDS= GROUPMEDLOSSTIMES=
Specify sample size and allocation	GROUPNS= GROUPWEIGHTS= NPERGROUP= NTOTAL=
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Specify computational method	NSUBINTERVAL=
Control ordering in output	OUTPUTORDER=

Table 67.24 summarizes the valid result parameters for different analyses in the TWOSAMPLESURVIVAL statement.

Table 67.24 Summary of Result Parameters in the TWOSAMPLESURVIVAL Statement

Analyses	Solve For	Syntax
TEST=GEHAN	Power	POWER=.
	Sample size	NTOTAL=.
		NPERGROUP=.
TEST=LOGRANK	Power	POWER=.
	Sample size	NTOTAL=.
		NPERGROUP=.
TEST=TARONEWARE	Power	POWER=.
	Sample size	NTOTAL=.
		NPERGROUP=.

Dictionary of Options

ACCRUALTIME=*number-list* | **MAX**

ACCTIME=*number-list* | **MAX**

ACCRUALT=*number-list* | **MAX**

ACCT=*number-list* | **MAX**

specifies the accrual time. Accrual is assumed to occur uniformly from time 0 to the time specified by the ACCRUALTIME= option. If the GROUPSURVIVAL= or REFSURVIVAL= option is used, then the value of the total time (the sum of accrual and follow-up times) must be less than or equal to the largest time in *each* multipoint (piecewise linear) survival curve. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

ACCRUALTIME=MAX can be used when each scenario in the analysis contains at least one piecewise linear survival curve (in the GROUPSURVIVAL= or REFSURVIVAL= option). It causes the accrual time to be automatically set, separately for each scenario, to the maximum possible time supported by the piecewise linear survival curve(s) in that scenario. It is not compatible with the FOLLOWUPTIME=MAX option or the TOTALTIME= option.

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual $0.05 \times 100\% = 5\%$ level of significance. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

CURVE("label")=points

defines a survival curve.

For the CURVE= option,

label identifies the curve in the output and with the GROUPLOSS=, GROUPSURVIVAL=, and REFSURVIVAL= options.

points specifies one or more (time, survival) pairs on the curve, where the survival value denotes the probability of surviving until at least the specified time.

A single-point curve is interpreted as exponential, and a multipoint curve is interpreted as piecewise linear. Points can be expressed in either of two forms:

- a series of time:survival pairs separated by spaces. For example:

```
1:0.9 2:0.7 3:0.6
```

- a DOLIST of times enclosed in parentheses, followed by a colon (:), followed by a DOLIST of survival values enclosed in parentheses. For example:

```
(1 to 3 by 1):(0.9 0.7 0.6)
```

  The DOLIST format is the same as in the DATA step.

Points can also be expressed as combinations of the two forms. For example:

```
1:0.9 2:0.8 (3 to 6 by 1):(0.7 0.65 0.6 0.55)
```

The points have the following restrictions:

- The time values must be nonnegative and strictly increasing.
- The survival values must be strictly decreasing.
- The survival value at a time of 0 must be equal to 1.
- If there is only one point, then the time must be greater than 0, and the survival value cannot be 0 or 1.

FOLLOWUPTIME=*number-list* | **MAX**

FUTIME=*number-list* | **MAX**

FOLLOWUPT=*number-list* | **MAX**

FUT=*number-list* | **MAX**

specifies the follow-up time, the amount of time in the study past the accrual time. If the GROUPSURVIVAL= or REFSURVIVAL= option is used, then the value of the total time (the sum of accrual and follow-up times) must be less than or equal to the largest time in *each* multipoint (piecewise linear) survival curve. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

FOLLOWUPTIME=MAX can be used when each scenario in the analysis contains at least one piecewise linear survival curve (in the GROUPSURVIVAL= or REFSURVIVAL= option). It causes the follow-up time to be automatically set, separately for each scenario, to the maximum possible time supported by the piecewise linear survival curve(s) in that scenario. It is not compatible with the ACCRUALTIME=MAX option or the TOTALTIME= option.

GROUPLOSS=*grouped-name-list*

GLOSS=*grouped-name-list*

specifies the exponential loss survival curve for each group, by using labels specified with the CURVE= option. Loss is assumed to follow an exponential curve, indicating the expected rate of censoring (in other words, loss to follow-up) over time. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-name-list*.

GROUPLOSSEXPHAZARDS=*grouped-number-list*

GLOSSEXPHAZARDS=*grouped-number-list*

GROUPLOSSEXPHS=*grouped-number-list*

GLOSSEXPHS=*grouped-number-list*

specifies the exponential hazards of the loss in each group. Loss is assumed to follow an exponential curve, indicating the expected rate of censoring (in other words, loss to follow-up) over time. If none of the GROUPLOSSEXPHAZARDS=, GROUPLOSS=, and GROUPMEDLOSSTIMES= options are used, the default of GROUPLOSSEXPHAZARDS=(0 0) indicates no loss to follow-up. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

GROUPMEDLOSSTIMES=*grouped-number-list*

GMEDLOSSTIMES=*grouped-number-list*

GROUPMEDLOSSTS=*grouped-number-list*

GMEDLOSSTS=*grouped-number-list*

specifies the median times of the loss in each group. Loss is assumed to follow an exponential curve, indicating the expected rate of censoring (in other words, loss to follow-up) over time. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

GROUPMEDSURVTIMES=*grouped-number-list*

GMEDSURVTIMES=*grouped-number-list*

GROUPMEDSURVTS=*grouped-number-list*

GMEDSURVTS=*grouped-number-list*

specifies the median survival times in each group. When the GROUPMEDSURVTIMES= option is used, the survival curve in each group is assumed to be exponential. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

GROUPNS=*grouped-number-list*

GNS=*grouped-number-list*

specifies the two group sample sizes. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

GROUPSURVEXPHAZARDS=*grouped-number-list*

GSURVEXPHAZARDS=*grouped-number-list*

GROUPSURVEXPHS=*grouped-number-list*

GEXPHS=*grouped-number-list*

specifies exponential hazard rates of the survival curve for each group. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

GROUPSURVIVAL=*grouped-name-list*

GSURVIVAL=*grouped-name-list*

GROUPSURV=*grouped-name-list*

GSURV=*grouped-name-list*

specifies the survival curve for each group, by using labels specified with the CURVE= option. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-name-list*.

GROUPWEIGHTS=*grouped-number-list*

GWEIGHTS=*grouped-number-list*

specifies the sample size allocation weights for the two groups. This option controls how the total sample size is divided between the two groups. Each pair of values for the two groups represents relative allocation weights. Additionally, if the NFRACTIONAL option is not used, the total sample size is restricted to be equal to a multiple of the sum of the two group weights (so that the resulting design has an integer sample size for each group while adhering exactly to the group allocation weights). Values must be integers unless the NFRACTIONAL option is used. The default value is (1 1), a balanced design with a weight of 1 for each group. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

HAZARDRATIO=*number-list*

HR=*number-list*

specifies the hazard ratio of the second group's survival curve to the first group's survival curve. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NFRACTIONAL

NFRAC

enables fractional input and output for sample sizes. See the section "Sample Size Adjustment Options" on page 4922 for information about the ramifications of the presence (and absence) of the NFRACTIONAL option.

NPERGROUP=*number-list*

NPERG=*number-list*

specifies the common sample size per group or requests a solution for the common sample size per group with a missing value (NPERGROUP=.). Use of this option implicitly specifies a balanced design. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NSUBINTERVAL=*number-list*

NSUBINTERVALS=*number-list*

NSUB=*number-list*

NSUBS=*number-list*

specifies the number of subintervals per unit time to use in internal calculations. Higher values increase computational time and memory requirements but generally lead to more accurate results. The default value is 12. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NTOTAL=*number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

OUTPUTORDER=INTERNAL

OUTPUTORDER=REVERSE

OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) arranges the parameters in the output according to the following order of their corresponding options:

- SIDES=
- ACCRUALTIME=
- FOLLOWUPTIME=
- TOTALTIME=
- NSUBINTERVAL=
- ALPHA=
- REFSURVIVAL=
- GROUPSURVIVAL=
- REFSURVEXPHAZARD=
- HAZARDRATIO=
- GROUPSURVEXPHAZARDS=
- GROUPMEDSURVTIMES=
- GROUPLOSSEXPHAZARDS=
- GROUPLOSS=
- GROUPMEDLOSSTIMES=
- GROUPWEIGHTS=
- NTOTAL=
- NPERGROUP=
- GROUPNS=
- POWER=

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order in which their corresponding options are specified in the TWOSAMPLESURVIVAL statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order in which their corresponding options are specified in the TWOSAMPLESURVIVAL statement.

POWER=*number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

REFSURVEXPHAZARD=*number-list*

REFSURVEXPH=*number-list*

specifies the exponential hazard rate of the survival curve for the first (reference) group. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

REFSURVIVAL=*name-list*

REFSURV=*name-list*

specifies the survival curve for the first (reference) group, by using labels specified with the CURVE= option. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *name-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and the direction of the statistical test or confidence interval. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *keyword-list*. Valid keywords and their interpretation are as follows:

1 one-sided with alternative hypothesis in same direction as effect

2 two-sided

U upper one-sided with the alternative hypothesis favoring better survival in the second group

L lower one-sided with the alternative hypothesis favoring better survival in the first (reference) group

The default value is 2.

TEST=GEHAN

TEST=LOGRANK

TEST=TARONEWARE

specifies the statistical analysis. TEST=GEHAN specifies the Gehan rank test. TEST=LOGRANK (the default) specifies the log-rank test. TEST=TARONEWARE specifies the Tarone-Ware rank test.

TOTALTIME=*number-list* | **MAX**

TOTALT=*number-list* | **MAX**

specifies the total time, which is equal to the sum of accrual and follow-up times. If the GROUPSURVIVAL= or REFSURVIVAL= option is used, then the value of the total time must be less than or equal to the largest time in *each* multipoint (piecewise linear) survival curve. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

TOTALTIME=MAX can be used when each scenario in the analysis contains at least one piecewise linear survival curve (in the GROUPSURVIVAL= or REFSURVIVAL= option). It causes the total time to be automatically set, separately for each scenario, to the maximum possible time supported by the piecewise linear survival curve(s) in that scenario. It is not compatible with the ACCRUALTIME=MAX option or the FOLLOWUPTIME=MAX option.

Restrictions on Option Combinations

To specify the survival curves, choose one of the following parameterizations:

- arbitrary piecewise linear or exponential curves (by using the CURVE= and GROUPSURVIVAL= options)
- curves with proportional hazards (by using the CURVE=, REFSURVIVAL=, and HAZARDRATIO= options)
- exponential curves, by using one of the following parameterizations:
  - median survival times (by using the GROUPMEDSURVTIMES= option)
  - the hazard ratio and the hazard of the reference curve (by using the HAZARDRATIO= and REFSURVEXPHAZARD= options)
  - the individual hazards (by using the GROUPSURVEXPHAZARDS= option)

To specify the study time, use any two of the following three options:

- accrual time (by using the ACCRUALTIME= option)
- follow-up time (by using the FOLLOWUPTIME= option)
- total time, the sum of accrual and follow-up times (by using the TOTALTIME= option)

To specify the sample size and allocation, choose one of the following parameterizations:

- sample size per group in a balanced design (by using the NPERGROUP= option)
- total sample size and allocation weights (by using the NTOTAL= and GROUPWEIGHTS= options)
- individual group sample sizes (by using the GROUPNS= option)

To specify the exponential loss curves, choose one of the following parameterizations:

- a point on the loss curve of each group (by using the CURVE= and GROUPLOSS= options)
- median loss times (by using the GROUPMEDLOSSTIMES= option)
- the individual loss hazards (by using the GROUPLOSSEXPHAZARDS= option)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the TWOSAMPLESURVIVAL statement.

Log-Rank Test for Two Survival Curves

You can use the NPERGROUP= option in a balanced design and specify piecewise linear or exponential survival curves by using the CURVE= and GROUPSURVIVAL= options, as in the following statements. Default values for the SIDES=, ALPHA=, NSUBINTERVAL=, and GROUPLOSSEXPHAZARDS= options specify a two-sided test with a significance level of 0.05, an assumption of no loss to follow-up, and the use of 12 subintervals per unit time in computations.

```
proc power;
   twosamplesurvival test=logrank
      curve("Control")   = (1 2 3):(0.8 0.7 0.6)
      curve("Treatment") = (5):(.6)
      groupsurvival = "Control" | "Treatment"
      accrualtime = 2
      followuptime = 1
      npergroup = 50
      power = .;
run;
```

In the preceding example, the "Control" curve is piecewise linear (since it has more than one point), and the "Treatment" curve is exponential (since it has only one point).

You can also specify an unbalanced design by using the NTOTAL= and GROUPWEIGHTS= options and specify piecewise linear or exponential survival curves with proportional hazards by using the CURVE=, REFSURVIVAL=, and HAZARDRATIO= options:

```
proc power;
   twosamplesurvival test=logrank
      curve("Control")   = (1 2 3):(0.8 0.7 0.6)
      refsurvival = "Control"
      hazardratio = 1.5
      accrualtime = 2
      followuptime = 1
      groupweights = (1 2)
      ntotal = .
      power = 0.8;
run;
```

You can also specify sample sizes with the GROUPNS= option and specify exponential survival curves in terms of median survival times:

```
proc power;
   twosamplesurvival test=logrank
      groupmedsurvtimes = (16 22)
      accrualtime = 6
      totaltime = 18
      groupns = 40 | 60
      power = .;
run;
```

You can also specify exponential survival curves in terms of the hazard ratio and reference hazard. The default value of the GROUPWEIGHTS= option specifies a balanced design.

```
proc power;
   twosamplesurvival test=logrank
      hazardratio = 1.2
      refsurvexphazard = 0.7
      accrualtime = 2
      totaltime = 4
      ntotal = 100
      power = .;
run;
```

You can also specify exponential survival curves in terms of the individual hazards, as in the following statements:

```
proc power;
   twosamplesurvival test=logrank
      groupsurvexphazards = 0.7 | 0.84
      accrualtime = 2
      totaltime = 4
      ntotal = .
      power = 0.9;
run;
```

Gehan Rank Test for Two Survival Curves

In addition to the logrank test, you can also specify the Gehan tank test, as in the following statements. Default values for the SIDES=, ALPHA=, NSUBINTERVAL=, and GROUPLOSSEXPHAZARDS= options specify a two-sided test with a significance level of 0.05, an assumption of no loss to follow-up, and the use of 12 subintervals per unit time in computations.

```
proc power;
   twosamplesurvival test=gehan
      groupmedsurvtimes = 5 | 7
      accrualtime = 3
      totaltime = 6
      npergroup = .
      power = 0.8;
run;
```

Tarone-Ware Rank Test for Two Survival Curves

You can also specify the Tarone-Ware tank test, as in the following statements. Default values for the SIDES=, ALPHA=, NSUBINTERVAL=, and GROUPLOSSEXPHAZARDS= options specify a two-sided test with a significance level of 0.05, an assumption of no loss to follow-up, and the use of 12 subintervals per unit time in computations.

```
proc power;
   twosamplesurvival test=taroneware
      groupmedsurvtimes = 5 | 7
      accrualtime = 3
      totaltime = 6
      npergroup = 100
      power = .;
run;
```

TWOSAMPLEWILCOXON Statement

TWOSAMPLEWILCOXON < *options* > **;**

The TWOSAMPLEWILCOXON statement performs power and sample size analyses for the Wilcoxon-Mann-Whitney test (also called the Wilcoxon rank-sum test, Mann-Whitney-Wilcoxon test, or Mann-Whitney U test) for two independent groups.

Note that the O'Brien-Castelloe approach to computing power for the Wilcoxon test is approximate, based on asymptotic behavior as the total sample size gets large. The quality of the power approximation degrades for small sample sizes; conversely, the quality of the sample size approximation degrades if the two distributions are far apart, so that only a small sample is needed to detect a significant difference. But this degradation is rarely a problem in practical situations, in which experiments are usually performed for relatively close distributions.

Summary of Options

Table 67.25 summarizes categories of options available in the TWOSAMPLEWILCOXON statement.

Table 67.25 Summary of Options in the TWOSAMPLEWILCOXON Statement

Task	Options
Define analysis	TEST=
Specify analysis information	ALPHA= SIDES=
Specify distributions	VARDIST= VARIABLES=
Specify sample size and allocation	GROUPNS= GROUPWEIGHTS= NPERGROUP= NTOTAL=
Specify power	POWER=
Control sample size rounding	NFRACTIONAL
Specify computational options	NBINS=
Control ordering in output	OUTPUTORDER=

Table 67.26 summarizes the valid result parameters in the TWOSAMPLEWILCOXON statement.

Table 67.26 Summary of Result Parameters in the TWOSAMPLEWILCOXON Statement

Analyses	Solve For	Syntax
TEST=WMW	Power	POWER=.
	Sample size	NTOTAL=. NPERGROUP=.

Dictionary of Options

ALPHA=*number-list*

specifies the level of significance of the statistical test. The default is 0.05, corresponding to the usual $0.05 \times 100\% = 5\%$ level of significance. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

GROUPNS=*grouped-number-list*

GNS=*grouped-number-list*

specifies the two group sample sizes. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

GROUPWEIGHTS=*grouped-number-list*

GWEIGHTS=*grouped-number-list*

specifies the sample size allocation weights for the two groups. This option controls how the total sample size is divided between the two groups. Each pair of values for the two groups represents relative allocation weights. Additionally, if the NFRACTIONAL option is not used, the total sample size is restricted to be equal to a multiple of the sum of the two group weights (so that the resulting design has an integer sample size for each group while adhering exactly to the group allocation weights). Values must be integers unless the NFRACTIONAL option is used. The default value is (1 1), a balanced design with a weight of 1 for each group. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-number-list*.

NBINS=*number-list*

specifies the number of categories (or "bins") each variable's distribution is divided into (unless it is ordinal, in which case the categories remain intact) in internal calculations. Higher values increase computational time and memory requirements but generally lead to more accurate results. However, if the value is too high, then numerical instability can occur. The default value is 1000. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NFRACTIONAL

NFRAC

enables fractional input and output for sample sizes. See the section "Sample Size Adjustment Options" on page 4922 for information about the ramifications of the presence (and absence) of the NFRACTIONAL option.

NPERGROUP=*number-list*

NPERG=*number-list*

specifies the common sample size per group or requests a solution for the common sample size per group with a missing value (NPERGROUP=.). Use of this option implicitly specifies a balanced design. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

NTOTAL=*number-list*

specifies the sample size or requests a solution for the sample size with a missing value (NTOTAL=.). See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

OUTPUTORDER=INTERNAL

OUTPUTORDER=REVERSE

OUTPUTORDER=SYNTAX

controls how the input and default analysis parameters are ordered in the output. OUTPUTORDER=INTERNAL (the default) arranges the parameters in the output according to the following order of their corresponding options:

- SIDES
- NBINS=
- ALPHA=
- VARIABLES=
- GROUPWEIGHTS=
- NTOTAL=
- NPERGROUP=
- GROUPNS=
- POWER=

The OUTPUTORDER=SYNTAX option arranges the parameters in the output in the same order in which their corresponding options are specified in the TWOSAMPLEWILCOXON statement. The OUTPUTORDER=REVERSE option arranges the parameters in the output in the reverse of the order in which their corresponding options are specified in the TWOSAMPLEWILCOXON statement.

POWER=*number-list*

specifies the desired power of the test or requests a solution for the power with a missing value (POWER=.). The power is expressed as a probability, a number between 0 and 1, rather than as a percentage. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *number-list*.

SIDES=*keyword-list*

specifies the number of sides (or tails) and the direction of the statistical test. Valid keywords are as follows:

1	one-sided with alternative hypothesis in same direction as effect
2	two-sided
U	upper one-sided with alternative greater than null value
L	lower one-sided with alternative less than null value

The default value is 2.

TEST=WMW

specifies the Wilcoxon-Mann-Whitney test for two independent groups This is the default test option.

VARDIST("label")=*distribution* **(***parameters***)**
defines a distribution for a variable.

For the VARDIST= option,

label identifies the variable distribution in the output and with the VARIABLES= option.

distribution specifies the distributional form of the variable.

parameters specifies one or more parameters associated with the distribution.

Choices for distributional forms and their parameters are as follows:

ORDINAL ((*values*) : (*probabilities*)) is an ordered categorical distribution. The *values* are any numbers separated by spaces. The *probabilities* are numbers between 0 and 1 (inclusive) separated by spaces. Their sum must be exactly 1. The number of *probabilities* must match the number of *values*.

BETA (*a*, *b* <, *l*, *r* >) is a beta distribution with shape parameters a and b and optional location parameters l and r. The values of a and b must be greater than 0, and l must be less than r. The default values for l and r are 0 and 1, respectively.

BINOMIAL (*p*, *n*) is a binomial distribution with probability of success p and number of independent Bernoulli trials n. The value of p must be greater than 0 and less than 1, and n must be an integer greater than 0.

EXPONENTIAL (λ) is an exponential distribution with scale λ, which must be greater than 0.

GAMMA (*a*, λ) is a gamma distribution with shape a and scale λ. The values of a and λ must be greater than 0.

LAPLACE (θ, λ) is a Laplace distribution with location θ and scale λ. The value of λ must be greater than 0.

LOGISTIC (θ, λ) is a logistic distribution with location θ and scale λ. The value of λ must be greater than 0.

LOGNORMAL (θ, λ) is a lognormal distribution with location θ and scale λ. The value of λ must be greater than 0.

NORMAL (θ, λ) is a normal distribution with mean θ and standard deviation λ. The value of λ must be greater than 0.

POISSON (*m*) is a Poisson distribution with mean m. The value of m must be greater than 0.

UNIFORM (*l*, *r*) is a uniform distribution on the interval $[l, r]$, where $l < r$.

VARIABLES=*grouped-name-list*

VARS=*grouped-name-list*
specifies the distributions of two or more variables, using labels specified with the VARDIST= option. See the section "Specifying Value Lists in Analysis Statements" on page 4919 for information about specifying the *grouped-name-list*.

Restrictions on Option Combinations

To specify the sample size and allocation, choose one of the following parameterizations:

- sample size per group in a balanced design (using the NPERGROUP= option)
- total sample size and allocation weights (using the NTOTAL= and GROUPWEIGHTS= options)
- individual group sample sizes (using the GROUPNS= option)

Option Groups for Common Analyses

This section summarizes the syntax for the common analyses supported in the TWOSAMPLEWILCOXON statement.

Wilcoxon-Mann-Whitney Test for Comparing Two Distributions

The following statements performs a power analysis for Wilcoxon-Mann-Whitney tests comparing an ordinal variable with each other type of distribution. Default values for the ALPHA=, NBINS=, SIDES=, and TEST= options specify a two-sided test with a significance level of 0.05 and the use of 1000 categories per distribution when discretization is needed.

```
proc power;
   twosamplewilcoxon
      vardist("myordinal") = ordinal ((0 1 2) : (.2 .3 .5))
      vardist("mybeta1") = beta (1, 2)
      vardist("mybeta2") = beta (1, 2, 0, 2)
      vardist("mybinomial") = binomial (.3, 3)
      vardist("myexponential") = exponential (2)
      vardist("mygamma") = gamma (1.5, 2)
      vardist("mylaplace") = laplace (1, 2)
      vardist("mylogistic") = logistic (1, 2)
      vardist("mylognormal") = lognormal (1, 2)
      vardist("mynormal") = normal (3, 2)
      vardist("mypoisson") = poisson (2)
      vardist("myuniform") = uniform (0, 2)
      variables = "myordinal" | "mybeta1" "mybeta2" "mybinomial"
                                "myexponential" "mygamma" "mylaplace"
                                "mylogistic" "mylognormal" "mynormal"
                                "mypoisson" "myuniform"
      ntotal = 40
      power = .;
run;
```

Details: POWER Procedure

Overview of Power Concepts

In statistical hypothesis testing, you typically express the belief that some effect exists in a population by specifying an alternative hypothesis H_1. You state a null hypothesis H_0 as the assertion that the effect does *not* exist and attempt to gather evidence to reject H_0 in favor of H_1. Evidence is gathered in the form of sample data, and a statistical test is used to assess H_0. If H_0 is rejected but there really is *no* effect, this is called a *Type I error*. The probability of a Type I error is usually designated "alpha" or α, and statistical tests are designed to ensure that α is suitably small (for example, less than 0.05).

If there really is an effect in the population but H_0 is *not* rejected in the statistical test, then a *Type II error* has been made. The probability of a Type II error is usually designated "beta" or β. The probability $1 - \beta$ of avoiding a Type II error—that is, correctly rejecting H_0 and achieving statistical significance—is called the *power*. (**NOTE:** Another more general definition of power is the probability of rejecting H_0 for any given set of circumstances, even those corresponding to H_0 being true. The POWER procedure uses this more general definition.)

An important goal in study planning is to ensure an acceptably high level of power. Sample size plays a prominent role in power computations because the focus is often on determining a sufficient sample size to achieve a certain power, or assessing the power for a range of different sample sizes.

Some of the analyses in the POWER procedure focus on *precision* rather than power. An analysis of confidence interval precision is analogous to a traditional power analysis, with "CI Half-Width" taking the place of effect size and "Prob(Width)" taking the place of power. The *CI Half-Width* is the margin of error associated with the confidence interval, the distance between the point estimate and an endpoint. The *Prob(Width)* is the probability of obtaining a confidence interval with *at most* a target half-width.

Summary of Analyses

Table 67.27 gives a summary of the analyses supported in the POWER procedure. The name of the analysis statement reflects the type of data and design. The TEST=, CI=, and DIST= options specify the focus of the statistical hypothesis (in other words, the criterion on which the research question is based) and the test statistic to be used in data analysis.

Table 67.27 Summary of Analyses

Analysis	Statement	Options
Logistic regression: likelihood ratio chi-square test	LOGISTIC	
Multiple linear regression: Type III F test	MULTREG	
Correlation: Fisher's z test	ONECORR	DIST=FISHERZ
Correlation: t test	ONECORR	DIST=T
Binomial proportion: exact test	ONESAMPLEFREQ	TEST=EXACT
Binomial proportion: z test	ONESAMPLEFREQ	TEST=Z
Binomial proportion: z test with continuity adjustment	ONESAMPLEFREQ	TEST=ADJZ
Binomial proportion: exact equivalence test	ONESAMPLEFREQ	TEST=EQUIV_EXACT
Binomial proportion: z equivalence test	ONESAMPLEFREQ	TEST=EQUIV_Z
Binomial proportion: z test with continuity adjustment	ONESAMPLEFREQ	TEST=EQUIV_ADJZ
Binomial proportion: confidence interval	ONESAMPLEFREQ	CI=AGRESTICOULL
		CI=JEFFREYS
		CI=EXACT
		CI=WALD
		CI=WALD_CORRECT
		CI=WILSON
One-sample t test	ONESAMPLEMEANS	TEST=T
One-sample t test with lognormal data	ONESAMPLEMEANS	TEST=T DIST=LOGNORMAL
One-sample equivalence test for mean of normal data	ONESAMPLEMEANS	TEST=EQUIV
One-sample equivalence test for mean of lognormal data	ONESAMPLEMEANS	TEST=EQUIV DIST=LOGNORMAL
Confidence interval for a mean	ONESAMPLEMEANS	CI=T
One-way ANOVA: one-degree-of-freedom contrast	ONEWAYANOVA	TEST=CONTRAST
One-way ANOVA: overall F test	ONEWAYANOVA	TEST=OVERALL
McNemar exact conditional test	PAIREDFREQ	
McNemar normal approximation test	PAIREDFREQ	DIST=NORMAL

Table 67.27 *continued*

Analysis	Statement	Options
Paired *t* test	PAIREDMEANS	TEST=DIFF
Paired *t* test of mean ratio with lognormal data	PAIREDMEANS	TEST=RATIO
Paired additive equivalence of mean difference with normal data	PAIREDMEANS	TEST=EQUIV_DIFF
Paired multiplicative equivalence of mean ratio with lognormal data	PAIREDMEANS	TEST=EQUIV_RATIO
Confidence interval for mean of paired differences	PAIREDMEANS	CI=DIFF
Pearson chi-square test for two independent proportions	TWOSAMPLEFREQ	TEST=PCHI
Fisher's exact test for two independent proportions	TWOSAMPLEFREQ	TEST=FISHER
Likelihood ratio chi-square test for two independent proportions	TWOSAMPLEFREQ	TEST=LRCHI
Two-sample *t* test assuming equal variances	TWOSAMPLEMEANS	TEST=DIFF
Two-sample Satterthwaite *t* test assuming unequal variances	TWOSAMPLEMEANS	TEST=DIFF_SATT
Two-sample pooled *t* test of mean ratio with lognormal data	TWOSAMPLEMEANS	TEST=RATIO
Two-sample additive equivalence of mean difference with normal data	TWOSAMPLEMEANS	TEST=EQUIV_DIFF
Two-sample multiplicative equivalence of mean ratio with lognormal data	TWOSAMPLEMEANS	TEST=EQUIV_RATIO
Two-sample confidence interval for mean difference	TWOSAMPLEMEANS	CI=DIFF
Log-rank test for comparing two survival curves	TWOSAMPLESURVIVAL	TEST=LOGRANK
Gehan rank test for comparing two survival curves	TWOSAMPLESURVIVAL	TEST=GEHAN
Tarone-Ware rank test for comparing two survival curves	TWOSAMPLESURVIVAL	TEST=TARONEWARE
Wilcoxon-Mann-Whitney (rank-sum) test	TWOSAMPLEWILCOXON	

Specifying Value Lists in Analysis Statements

To specify one or more scenarios for an analysis parameter (or set of parameters), you provide a list of values for the statement option that corresponds to the parameter(s). To identify the parameter you want to solve for, you place missing values in the appropriate list.

There are five basic types of such lists: *keyword-lists*, *number-lists*, *grouped-number-lists*, *name-lists*, and *grouped-name-lists*. Some parameters, such as the direction of a test, have values represented by one or more keywords in a *keyword-list*. Scenarios for scalar-valued parameters, such as power, are represented by a *number-list*. Scenarios for groups of scalar-valued parameters, such as group sample sizes in a multigroup design, are represented by a *grouped-number-list*. Scenarios for named parameters, such as reference survival curves, are represented by a *name-list*. Scenarios for groups of named parameters, such as group survival curves, are represented by a *grouped-name-list*.

The following subsections explain these five basic types of lists.

Keyword-Lists

A *keyword-list* is a list of one or more keywords separated by spaces. For example, you can specify both two-sided and upper-tailed versions of a one-sample *t* test:

```
SIDES = 2 U
```

Number-Lists

A *number-list* can be one of two things: a series of one or more numbers expressed in the form of one or more DOLISTs, or a missing value indicator (.).

The DOLIST format is the same as in the DATA step language. For example, for the one-sample *t* test you can specify four scenarios (30, 50, 70, and 100) for a total sample size in any of the following ways.

```
NTOTAL = 30 50 70 100
NTOTAL = 30 to 70 by 20 100
```

A missing value identifies a parameter as the result parameter; it is valid only with options representing parameters you can solve for in a given analysis. For example, you can request a solution for NTOTAL:

```
NTOTAL = .
```

Grouped-Number-Lists

A *grouped-number-list* specifies multiple scenarios for numeric values in two or more groups, possibly including missing value indicators to solve for a specific group. The list can assume one of two general forms, a “crossed” version and a “matched” version.

Crossed Grouped-Number-Lists

The crossed version of a grouped number list consists of a series of *number-lists* (see the section "Number-Lists" on page 4919), one representing each group, with groups separated by a vertical bar (|). The values for each group represent multiple scenarios for that group, and the scenarios for each individual group are crossed to produce the set of all scenarios for the analysis option. For example, you can specify the following six scenarios for the sizes (n_1, n_2) of two groups

$$(20, 30)(20, 40)(20, 50)$$
$$(25, 30)(25, 40)(25, 50)$$

as follows:

```
GROUPNS = 20 25 | 30 40 50
```

If the analysis can solve for a value in one group given the other groups, then one of the *number-lists* in a *crossed grouped-number-list* can be a missing value indicator (.). For example, in a two-sample t test you can posit three scenarios for the group 2 sample size while solving for the group 1 sample size:

```
GROUPNS = . | 30 40 50
```

Some analyses can involve more than two groups. For example, you can specify $2 \times 3 \times 1 = 6$ scenarios for the means of three groups in a one-way ANOVA as follows:

```
GROUPMEANS = 10 12 | 10 to 20 by 5 | 24
```

Matched Grouped-Number-Lists

The matched version of a grouped number list consists of a series of numeric lists, each enclosed in parentheses. Each list consists of a value for each group and represents a single scenario for the analysis option. Multiple scenarios for the analysis option are represented by multiple lists. For example, you can express the crossed grouped-number-list

```
GROUPNS = 20 25 | 30 40 50
```

alternatively in a matched format:

```
GROUPNS = (20 30) (20 40) (20 50) (25 30) (25 40) (25 50)
```

The matched version is particularly useful when you want to include only a subset of all combinations of individual group values. For example, you might want to pair 20 only with 50, and 25 only with 30 and 40:

```
GROUPNS = (20 50) (25 30) (25 40)
```

If the analysis can solve for a value in one group given the other groups, then you can replace the value for that group with a missing value indicator (.). If used, the missing value indicator must occur in the same group in every scenario. For example, you can solve for the group 1 sample size (as in the section "Crossed Grouped-Number-Lists" on page 4920) by using a matched format:

```
GROUPNS = (. 30) (. 40) (. 50)
```

Some analyses can involve more than two groups. For example, you can specify two scenarios for the means of three groups in a one-way ANOVA:

```
GROUPMEANS = (15 24 32) (12 25 36)
```

Name-Lists

A *name-list* is a list of one or more names in single or double quotes, separated by spaces. For example, you can specify two scenarios for the reference survival curve in a log-rank test:

```
REFSURVIVAL = "Curve A" "Curve B"
```

Grouped-Name-Lists

A *grouped-name-list* specifies multiple scenarios for names in two or more groups. The list can assume one of two general forms, a "crossed" version and a "matched" version.

Crossed Grouped-Name-Lists

The crossed version of a grouped name list consists of a series of *name-lists* (see the section "Name-Lists" on page 4921), one representing each group, with groups separated by a vertical bar (|). The values for each group represent multiple scenarios for that group, and the scenarios for each individual group are crossed to produce the set of all scenarios for the analysis option. For example, you can specify the following six scenarios for the survival curves (c_1, c_2) of two groups

("Curve A", "Curve C")("Curve A", "Curve D")("Curve A", "Curve E")
("Curve B", "Curve C")("Curve B", "Curve D")("Curve B", "Curve E")

as follows:

```
GROUPSURVIVAL = "Curve A" "Curve B" | "Curve C" "Curve D"
                                            "Curve E"
```

Matched Grouped-Name-Lists

The matched version of a grouped name list consists of a series of name lists, each enclosed in parentheses. Each list consists of a name for each group and represents a single scenario for the analysis option. Multiple scenarios for the analysis option are represented by multiple lists. For example, you can express the crossed grouped-name-list

```
GROUPSURVIVAL = "Curve A" "Curve B" | "Curve C" "Curve D"
                                      "Curve E"
```

alternatively in a matched format:

```
GROUPSURVIVAL = ("Curve A" "Curve C")
                ("Curve A" "Curve D")
                ("Curve A" "Curve E")
                ("Curve B" "Curve C")
                ("Curve B" "Curve D")
                ("Curve B" "Curve E")
```

The matched version is particularly useful when you want to include only a subset of all combinations of individual group values. For example, you might want to pair “Curve A” only with “Curve C”, and “Curve B” only with “Curve D” and “Curve E”:

```
GROUPSURVIVAL = ("Curve A" "Curve C")
                ("Curve B" "Curve D")
                ("Curve B" "Curve E")
```

Sample Size Adjustment Options

By default, PROC POWER rounds sample sizes conservatively (down in the input, up in the output) so that all total sizes (and individual group sample sizes, if a multigroup design) are integers. This is generally considered conservative because it selects the closest realistic design providing *at most* the power of the (possibly fractional) input or mathematically optimized design. In addition, in a multigroup design, all group sizes are adjusted to be multiples of the corresponding group weights. For example, if GROUPWEIGHTS = (2 6), then all group 1 sample sizes become multiples of 2, and all group 2 sample sizes become multiples of 6 (and all total sample sizes become multiples of 8).

With the NFRACTIONAL option, sample size input is not rounded, and sample size output (whether total or groupwise) are reported in two versions, a raw “fractional” version and a “ceiling” version rounded up to the nearest integer.

Whenever an input sample size is adjusted, both the original (“nominal”) and adjusted (“actual”) sample sizes are reported. Whenever computed output sample sizes are adjusted, both the original input (“nominal”) power and the achieved (“actual”) power at the adjusted sample size are reported.

Error and Information Output

The Error column in the main output table provides reasons for missing results and flags numerical results that are bounds rather than exact answers. For example, consider the sample size analysis implemented by the following statements:

```
proc power;
   twosamplefreq test=pchi
     method=normal
     oddsratio= 1.0001
     refproportion=.4
     nulloddsratio=1
     power=.9
     ntotal=.;
run;
```

Figure 67.6 Error Column

The POWER Procedure
Pearson Chi-square Test for Two Proportions

Fixed Scenario Elements

Distribution	Asymptotic normal
Method	Normal approximation
Null Odds Ratio	1
Reference (Group 1) Proportion	0.4
Odds Ratio	1.0001
Nominal Power	0.9
Number of Sides	2
Alpha	0.05
Group 1 Weight	1
Group 2 Weight	1

Computed N Total

Actual Power	N Total	Error
0.206	2.15E+09	Solution is a lower bound

The output in Figure 67.6 reveals that the sample size to achieve a power of 0.9 could not be computed, but that the sample size 2.15E+09 achieves a power of 0.206.

The Info column provides further details about Error column entries, warnings about any boundary conditions detected, and notes about any adjustments to input. Note that the Info column is hidden by default in the main output. You can view it by using the ODS OUTPUT statement to save the output as a data set and the PRINT procedure. For example, the following SAS statements print both the Error and Info columns for a power computation in a two-sample t test:

```
proc power;
   twosamplemeans
      meandiff= 0 7
      stddev=2
      ntotal=2 5
      power=.;
   ods output output=Power;
run;

proc print noobs data=Power;
   var MeanDiff NominalNTotal NTotal Power Error Info;
run;
```

The output is shown in Figure 67.7.

Figure 67.7 Error and Info Columns

Mean Diff	Nominal NTotal	NTotal	Power	Error	Info
0	2	2	.	Invalid input	N too small / No effect
0	5	4	0.050		Input N adjusted / No effect
7	2	2	.	Invalid input	N too small
7	5	4	0.477		Input N adjusted

The mean difference of 0 specified with the MEANDIFF= option leads to a "No effect" message to appear in the Info column. The sample size of 2 specified with the NTOTAL= option leads to an "Invalid input" message in the Error column and an "NTotal too small" message in the Info column. The sample size of 5 leads to an "Input N adjusted" message in the Info column because it is rounded down to 4 to produce integer group sizes of 2 per group.

Displayed Output

If you use the PLOTONLY option in the PROC POWER statement, the procedure displays only graphical output. Otherwise, the displayed output of the POWER procedure includes the following:

- the "Fixed Scenario Elements" table, which shows all applicable single-valued analysis parameters, in the following order: distribution, method, parameters input explicitly, and parameters supplied with defaults
- an output table showing the following when applicable (in order): the index of the scenario, all multivalued input, ancillary results, the primary computed result, and error descriptions
- plots (if requested)

For each input parameter, the order of the input values is preserved in the output.

Ancillary results include the following:

- Actual Power, the achieved power, if it differs from the input (Nominal) power value
- Actual Prob(Width), the achieved precision probability, if it differs from the input (Nominal) probability value
- Actual Alpha, the achieved significance level, if it differs from the input (Nominal) alpha value
- fractional sample size, if the NFRACTIONAL option is used in the analysis statement

If sample size is the result parameter and the NFRACTIONAL option is used in the analysis statement, then both "Fractional" and "Ceiling" sample size results are displayed. Fractional sample sizes correspond to the "Nominal" values of power or precision probability. Ceiling sample sizes are simply the fractional sample sizes rounded up to the nearest integer; they correspond to "Actual" values of power or precision probability.

ODS Table Names

PROC POWER assigns a name to each table that it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 67.28. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 67.28 ODS Tables Produced by PROC POWER

ODS Table Name	Description	Statement
FixedElements	factoid with single-valued analysis parameters	default*
Output	all input and computed analysis parameters, error messages, and information messages for each scenario	default
PlotContent	data contained in plots, including analysis parameters and indices identifying plot features. (**NOTE:** This table is saved as a data set and not displayed in PROC POWER output.)	PLOT

*Depends on input.

The ODS pathnames are created as follows:

- Power.<analysis statement name>< n >.FixedElements
- Power.<analysis statement name>< n >.Output
- Power.<analysis statement name>< n >.PlotContent
- Power.<analysis statement name>< n >.Plot< m >

where

- The Plot< m > objects are the graphs.
- The < n > indexing the analysis statement name is used only if there is more than one instance.
- The < n > indexing the plots increases with every panel in every plot statement, resetting to 1 only at new analysis statements.

Computational Resources

Memory

In the TWOSAMPLESURVIVAL statement, the amount of required memory is roughly proportional to the product of the number of subintervals (specified by the NSUBINTERVAL= option) and the total time of the study (specified by the ACCRUALTIME=, FOLLOWUPTIME=, and TOTALTIME= options). If you run out of memory, then you can try either specifying a smaller number of subintervals, changing the time scale to a use a longer time unit (for example, years instead of months), or both.

CPU Time

In the Satterthwaite t test analysis (TWOSAMPLEMEANS TEST=DIFF_SATT), the required CPU time grows as the mean difference decreases relative to the standard deviations. In the PAIREDFREQ statement, the required CPU time for the exact power computation (METHOD=EXACT) grows with the sample size.

Computational Methods and Formulas

This section describes the approaches used in PROC POWER to compute power for each analysis. The first subsection defines some common notation. The following subsections describe the various power analyses, including discussions of the data, statistical test, and power formula for each analysis. Unless otherwise indicated, computed values for parameters besides power (for example, sample size) are obtained by solving power formulas for the desired parameters.

Common Notation

Table 67.29 displays notation for some of the more common parameters across analyses. The Associated Syntax column shows examples of relevant analysis statement options, where applicable.

Table 67.29 Common Notation

Symbol	Description	Associated Syntax
α	significance level	ALPHA=
N	total sample size	NTOTAL=, NPAIRS=
n_i	sample size in ith group	NPERGROUP=, GROUPNS=
w_i	allocation weight for ith group (standardized to sum to 1)	GROUPWEIGHTS=
μ	(arithmetic) mean	MEAN=

Table 67.29 *continued*

Symbol	Description	Associated Syntax
μ_i	(arithmetic) mean in ith group	GROUPMEANS=, PAIREDMEANS=
μ_{diff}	(arithmetic) mean difference, $\mu_2 - \mu_1$ or $\mu_T - \mu_R$	MEANDIFF=
μ_0	null mean or mean difference (arithmetic)	NULL=, NULLDIFF=
γ	geometric mean	MEAN=
γ_i	geometric mean in ith group	GROUPMEANS=, PAIREDMEANS=
γ_0	null mean or mean ratio (geometric)	NULL=, NULLRATIO=
σ	standard deviation (or common standard deviation per group)	STDDEV=
σ_i	standard deviation in ith group	GROUPSTDDEVS=, PAIREDSTDDEVS=
σ_{diff}	standard deviation of differences	
CV	coefficient of variation, defined as the ratio of the standard deviation to the (arithmetic) mean	CV=, PAIREDCVS=
ρ	correlation	CORR=
μ_T, μ_R	treatment and reference (arithmetic) means for equivalence test	GROUPMEANS=, PAIREDMEANS=
γ_T, γ_R	treatment and reference geometric means for equivalence test	GROUPMEANS=, PAIREDMEANS=
θ_L	lower equivalence bound	LOWER=
θ_U	upper equivalence bound	UPPER=
$t(\nu, \delta)$	t distribution with d.f. ν and noncentrality δ	
$F(\nu_1, \nu_2, \lambda)$	F distribution with numerator d.f. ν_1, denominator d.f. ν_2, and noncentrality λ	
$t_{p;\nu}$	pth percentile of t distribution with d.f. ν	
$F_{p;\nu_1,\nu_2}$	pth percentile of F distribution with numerator d.f. ν_1 and denominator d.f. ν_2	
Bin(N, p)	binomial distribution with sample size N and proportion p	

A "lower one-sided" test is associated with SIDES=L (or SIDES=1 with the effect smaller than the null value), and an "upper one-sided" test is associated with SIDES=U (or SIDES=1 with the effect larger than the null value).

Owen (1965) defines a function, known as Owen's Q, that is convenient for representing terms in

power formulas for confidence intervals and equivalence tests:

$$Q_\nu(t,\delta;a,b) = \frac{\sqrt{2\pi}}{\Gamma(\frac{\nu}{2})2^{\frac{\nu-2}{2}}}\int_a^b \Phi\left(\frac{tx}{\sqrt{\nu}} - \delta\right) x^{\nu-1}\phi(x)\mathrm{d}x$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the density and cumulative distribution function of the standard normal distribution, respectively.

Analyses in the LOGISTIC Statement

Likelihood Ratio Chi-Square Test for One Predictor (TEST=LRCHI)

The power computing formula is based on Shieh and O'Brien (1998), Shieh (2000), and Self, Mauritsen, and Ohara (1992).

Define the following notation for a logistic regression analysis:

N = # subjects (NTOTAL)
K = # predictors (not counting intercept)
$\mathbf{x} = (x_1,\ldots,x_K)'$ = random variables for predictor vector
$\boldsymbol{\mu} = (\mu_1,\ldots,\mu_K)' = \mathrm{E}\mathbf{x}$ = mean predictor vector
$\mathbf{x}_i = (x_{i1},\ldots,x_{iK})'$ = predictor vector for subject i $(i \in 1,\ldots,N)$
Y = random variable for response (0 or 1)
Y_i = response for subject i $(i \in 1,\ldots,N)$
$p_i = \mathrm{Prob}(Y_i = 1|\mathbf{x}_i)$ $(i \in 1,\ldots,N)$
$\phi = \mathrm{Prob}(Y_i = 1|\mathbf{x}_i = \boldsymbol{\mu})$ (RESPONSEPROB)
U_j = unit change for jth predictor (UNITS)
$\mathrm{OR}_j = \mathrm{Odds}(Y_i = 1|x_{ij} = c)/\mathrm{Odds}(Y_i = 1|x_{ij} = c - U_j)$ (c arbitrary, $i \in 1,\ldots,N$, $j \in 1,\ldots,K$) (TESTODDSRATIO if $j = 1$, COVODDSRATIOS if $j > 1$)
Ψ_0 = intercept in full model (INTERCEPT)
$\boldsymbol{\Psi} = (\Psi_1,\ldots,\Psi_K)'$ = regression coefficients in full model (Ψ_1 = TESTREGCOEFF, others = COVREGCOEFFS)
c_j = # distinct possible values of x_{ij} $(j \in 1,\ldots,K)$ (for any i) (NBINS)
$x^\star_{gj}$ = gth possible value of x_{ij} $(g \in 1,\ldots,c_j)$ $(j \in 1,\ldots,K)$ (for any i) (VARDIST)
$\pi_{gj} = \mathrm{Prob}\left(x_{ij} = x^\star_{gj}\right)$ $(g \in 1,\ldots,c_j)$ $(j \in 1,\ldots,K)$ (for any i) (VARDIST)
$C = \prod_{j=1}^{K} c_j$ = # possible values of $\mathbf{x}_i$ (for any i)
$\mathbf{x}^\star_m$ = mth possible value of $\mathbf{x}_i$ $(m \in 1,\ldots,C)$
$\pi_m = \mathrm{Prob}\left(\mathbf{x}_i = \mathbf{x}^\star_m\right)$ $(m \in 1,\ldots,C)$

The logistic regression model is

$$\log\left(\frac{p_i}{1-p_i}\right) = \Psi_0 + \boldsymbol{\Psi}'\mathbf{x}_i$$

The hypothesis test of the first predictor variable is

$$H_0{:}\Psi_1 = 0$$
$$H_1{:}\Psi_1 \neq 0$$

Assuming independence among all predictor variables, π_m is defined as follows:

$$\pi_m = \prod_{j=1}^{K} \pi_{h(m,j)j} \quad (m \in 1, \ldots, C)$$

where $h(m, j)$ is calculated according to the following algorithm:

$z = m;$
do $j = K$ to 1;
  $h(m, j) = \text{mod}(z - 1, c_j) + 1;$
  $z = \text{floor}((z - 1)/c_j) + 1;$
end;

This algorithm causes the elements of the transposed vector $\{h(m, 1), \ldots, h(m, K)\}$ to vary fastest to slowest from right to left as m increases, as shown in the following table of $h(m, j)$ values:

			j		
$h(m, j)$	1	2	$\cdots$	$K-1$	K
1	1	1	$\cdots$	1	1
1	1	1	$\cdots$	1	2
$\vdots$			$\vdots$		
$\vdots$	1	1	$\cdots$	1	c_K
$\vdots$	1	1	$\cdots$	2	1
$\vdots$	1	1	$\cdots$	2	2
$\vdots$			$\vdots$		
m $\vdots$	1	1	$\cdots$	2	c_K
$\vdots$			$\vdots$		
$\vdots$	c_1	c_2	$\cdots$	c_{K-1}	1
$\vdots$	c_1	c_2	$\cdots$	c_{K-1}	2
$\vdots$			$\vdots$		
C	c_1	c_2	$\cdots$	c_{K-1}	c_K

The $\mathbf{x}_m^\star$ values are determined in a completely analogous manner.

The discretization is handled as follows (unless the distribution is ordinal, or binomial with sample size parameter at least as large as requested number of bins): for x_j, generate c_j quantiles at evenly spaced probability values such that each such quantile is at the midpoint of a bin with probability $\frac{1}{c_j}$. In other words,

$$x^{\star}_{gj} = \left(\frac{g-0.5}{c_j}\right) \text{th quantile of relevant distribution,}$$
$$(g \in 1,\ldots,c_j)(j \in 1,\ldots,K)$$
$$\pi_{gj} = \frac{1}{c_j} \quad \text{(same for all } g\text{)}$$

The primary noncentrality for the power computation is

$$\Delta^{\star} = 2\sum_{m=1}^{C} \pi_m \left[b'(\theta_m)\left(\theta_m - \theta^{\star}_m\right) - \left(b(\theta_m) - b(\theta^{\star}_m)\right)\right]$$

where

$$b'(\theta) = \frac{\exp(\theta)}{1+\exp(\theta)}$$
$$b(\theta) = \log\left(1+\exp(\theta)\right)$$
$$\theta_m = \Psi_0 + \boldsymbol{\Psi}'\mathbf{x}^{\star}_m$$
$$\theta^{\star}_m = \Psi^{\star}_0 + \boldsymbol{\Psi}^{\star\prime}\mathbf{x}^{\star}_m$$

where

$$\Psi^{\star}_0 = \Psi_0 + \Psi_1\mu_1 = \text{ intercept in reduced model, absorbing the tested predictor}$$
$$\boldsymbol{\Psi}^{\star} = (0,\Psi_2,\ldots,\Psi_K)' = \text{ coefficients in reduced model}$$

The power is

$$\text{power} = P\left(\chi^2(1,\Delta^{\star}N) \geq \chi^2_{1-\alpha}(1)\right)$$

Alternative input parameterizations are handled by the following transformations:

$$\Psi_0 = \log\left(\frac{\phi}{1-\phi}\right) - \boldsymbol{\Psi}'\boldsymbol{\mu}$$
$$\Psi_j = \frac{\log(\mathrm{OR}_j)}{U_j} \quad (j \in 1,\ldots,K)$$

Analyses in the MULTREG Statement

Type III F Test in Multiple Regression (TEST=TYPE3)

Maxwell (2000) discusses a number of different ways to represent effect sizes (and to compute exact power based on them) in multiple regression. PROC POWER supports two of these, multiple partial correlation and R^2 in full and reduced models.

Let p denote the total number of predictors in the full model (excluding the intercept), and let Y denote the response variable. You are testing that the coefficients of $p_1 \geq 1$ predictors in a set X_1 are 0, controlling for all of the other predictors X_{-1}, which consists of $p - p_1 \geq 0$ variables.

The hypotheses can be expressed in two different ways. The first is in terms of $\rho_{YX_1|X_{-1}}$, the multiple partial correlation between the predictors in X_1 and the response Y adjusting for the predictors in X_{-1}:

$$H_0: \rho^2_{YX_1|X_{-1}} = 0$$
$$H_1: \rho^2_{YX_1|X_{-1}} > 0$$

The second is in terms of the multiple correlations in full ($\rho_{Y|(X_1,X_{-1})}$) and reduced ($\rho_{Y|X_{-1}}$) nested models:

$$H_0: \rho^2_{Y|(X_1,X_{-1})} - \rho^2_{Y|X_{-1}} = 0$$
$$H_1: \rho^2_{Y|(X_1,X_{-1})} - \rho^2_{Y|X_{-1}} > 0$$

Note that the squared values of $\rho_{Y|(X_1,X_{-1})}$ and $\rho_{Y|X_{-1}}$ are the population R^2 values for full and reduced models.

The test statistic can be written in terms of the sample multiple partial correlation $R_{YX_1|X_{-1}}$,

$$F = \begin{cases} (N-1-p)\dfrac{R^2_{YX_1|X_{-1}}}{1-R^2_{YX_1|X_{-1}}}, & \text{intercept} \\ (N-p)\dfrac{R^2_{YX_1|X_{-1}}}{1-R^2_{YX_1|X_{-1}}}, & \text{no intercept} \end{cases}$$

or the sample multiple correlations in full ($R_{Y|(X_1,X_{-1})}$) and reduced ($R_{Y|X_{-1}}$) models,

$$F = \begin{cases} (N-1-p)\dfrac{R^2_{Y|(X_1,X_{-1})}-R^2_{Y|X_{-1}}}{1-R^2_{Y|(X_1,X_{-1})}}, & \text{intercept} \\ (N-p)\dfrac{R^2_{Y|(X_1,X_{-1})}-R^2_{Y|X_{-1}}}{1-R^2_{Y|(X_1,X_{-1})}}, & \text{no intercept} \end{cases}$$

The test is the usual Type III F test in multiple regression:

$$\text{Reject} \quad H_0 \quad \text{if} \begin{cases} F \geq F_{1-\alpha}(p_1, N-1-p), & \text{intercept} \\ F \geq F_{1-\alpha}(p_1, N-p), & \text{no intercept} \end{cases}$$

Although the test is invariant to whether the predictors are assumed to be random or fixed, the power is affected by this assumption. If the response and predictors are assumed to have a joint multivariate normal distribution, then the exact power is given by the following formula:

$$\begin{aligned} \text{power} &= \begin{cases} P\left[\left(\frac{N-1-p}{p_1}\right)\left(\frac{R^2_{Y|(X_1,X_{-1})}}{1-R^2_{Y|(X_1,X_{-1})}}\right) \geq F_{1-\alpha}(p_1, N-1-p)\right], & \text{intercept} \\ P\left[\left(\frac{N-p}{p_1}\right)\left(\frac{R^2_{Y|(X_1,X_{-1})}}{1-R^2_{Y|(X_1,X_{-1})}}\right) \geq F_{1-\alpha}(p_1, N-p)\right], & \text{no intercept} \end{cases} \\ &= \begin{cases} P\left[R^2_{Y|(X_1,X_{-1})} \geq \frac{F_{1-\alpha}(p_1,N-1-p)}{F_{1-\alpha}(p_1,N-1-p)+\frac{N-1-p}{p_1}}\right], & \text{intercept} \\ P\left[R^2_{Y|(X_1,X_{-1})} \geq \frac{F_{1-\alpha}(p_1,N-p)}{F_{1-\alpha}(p_1,N-p)+\frac{N-p}{p_1}}\right], & \text{no intercept} \end{cases} \end{aligned}$$

The distribution of $R^2_{Y|(X_1,X_{-1})}$ (for any $\rho^2_{Y|(X_1,X_{-1})}$) is given in Chapter 32 of Johnson, Kotz, and Balakrishnan (1995). Sample size tables are presented in Gatsonis and Sampson (1989).

If the predictors are assumed to have fixed values, then the exact power is given by the noncentral F distribution. The noncentrality parameter is

$$\lambda = N \frac{\rho^2_{YX_1|X_{-1}}}{1 - \rho^2_{YX_1|X_{-1}}}$$

or equivalently,

$$\lambda = N \frac{\rho^2_{Y|(X_1,X_{-1})} - \rho^2_{Y|X_{-1}}}{1 - \rho^2_{Y|(X_1,X_{-1})}}$$

The power is

$$\text{power} = \begin{cases} P\left(F(p_1, N-1-p, \lambda) \geq F_{1-\alpha}(p_1, N-1-p)\right), & \text{intercept} \\ P\left(F(p_1, N-p, \lambda) \geq F_{1-\alpha}(p_1, N-p)\right), & \text{no intercept} \end{cases}$$

The minimum acceptable input value of N depends on several factors, as shown in Table 67.30.

Table 67.30 Minimum Acceptable Sample Size Values in the MULTREG Statement

Predictor Type	Intercept in Model?	$p_1 = 1$?	Minimum N
Random	Yes	Yes	$p + 3$
Random	Yes	No	$p + 2$
Random	No	Yes	$p + 2$
Random	No	No	$p + 1$
Fixed	Yes	Yes or No	$p + 2$
Fixed	No	Yes or No	$p + 1$

Analyses in the ONECORR Statement

Fisher's z Test for Pearson Correlation (TEST=PEARSON DIST=FISHERZ)

Fisher's z transformation (Fisher 1921) of the sample correlation $R_{Y|(X_1,X_{-1})}$ is defined as

$$z = \frac{1}{2} \log\left(\frac{1 + R_{Y|(X_1,X_{-1})}}{1 - R_{Y|(X_1,X_{-1})}}\right)$$

Fisher's z test assumes the approximate normal distribution $N(\mu, \sigma^2)$ for z, where

$$\mu = \frac{1}{2} \log\left(\frac{1 + \rho_{Y|(X_1,X_{-1})}}{1 - \rho_{Y|(X_1,X_{-1})}}\right) + \frac{\rho_{Y|(X_1,X_{-1})}}{2(N - 1 - p^\star)}$$

and

$$\sigma^2 = \frac{1}{N - 3 - p^\star}$$

where $p^\star$ is the number of variables partialed out (Anderson 1984, pp. 132–133) and $\rho_{Y|(X_1,X_{-1})}$ is the partial correlation between Y and X_1 adjusting for the set of zero or more variables X_{-1}.

The test statistic

$$z^\star = (N-3-p^\star)^{\frac{1}{2}}\left[z - \frac{1}{2}\log\left(\frac{1+\rho_0}{1-\rho_0}\right) - \frac{\rho_0}{2(N-1-p^\star)}\right]$$

is assumed to have a normal distribution $N(\delta, \nu)$, where ρ_0 is the null partial correlation and δ and ν are derived from Section 16.33 of Stuart and Ord (1994):

$$\delta = (N-3-p^\star)^{\frac{1}{2}}\left[\frac{1}{2}\log\left(\frac{1+\rho_{Y|(X_1,X_{-1})}}{1-\rho_{Y|(X_1,X_{-1})}}\right) + \frac{\rho_{Y|(X_1,X_{-1})}}{2(N-1-p^\star)}\left(1 + \frac{5+\rho^2_{Y|(X_1,X_{-1})}}{4(N-1-p^\star)} + \frac{11+2\rho^2_{Y|(X_1,X_{-1})}+3\rho^4_{Y|(X_1,X_{-1})}}{8(N-1-p^\star)^2}\right) - \frac{1}{2}\log\left(\frac{1+\rho_0}{1-\rho_0}\right) - \frac{\rho_0}{2(N-1-p^\star)}\right]$$

$$\nu = \frac{N-3-p^\star}{N-1-p^\star}\left[1 + \frac{4-\rho^2_{Y|(X_1,X_{-1})}}{2(N-1-p^\star)} + \frac{22-6\rho^2_{Y|(X_1,X_{-1})}-3\rho^4_{Y|(X_1,X_{-1})}}{6(N-1-p^\star)^2}\right]$$

The approximate power is computed as

$$\text{power} = \begin{cases} \Phi\left(\frac{\delta - z_{1-\alpha}}{\nu^{\frac{1}{2}}}\right), & \text{upper one-sided} \\ \Phi\left(\frac{-\delta - z_{1-\alpha}}{\nu^{\frac{1}{2}}}\right), & \text{lower one-sided} \\ \Phi\left(\frac{\delta - z_{1-\frac{\alpha}{2}}}{\nu^{\frac{1}{2}}}\right) + \Phi\left(\frac{-\delta - z_{1-\frac{\alpha}{2}}}{\nu^{\frac{1}{2}}}\right), & \text{two-sided} \end{cases}$$

Because the test is biased, the achieved significance level might differ from the nominal significance level. The actual alpha is computed in the same way as the power except with the correlation $\rho_{Y|(X_1,X_{-1})}$ replaced by the null correlation ρ_0.

t *Test for Pearson Correlation (TEST=PEARSON DIST=T)*

The two-sided case is identical to multiple regression with an intercept and $p_1 = 1$, which is discussed in the section "Analyses in the MULTREG Statement" on page 4930.

Let $p^\star$ denote the number of variables partialed out. For the one-sided cases, the test statistic is

$$t = (N-2-p^\star)^{\frac{1}{2}}\frac{R_{YX_1|X_{-1}}}{\left(1-R^2_{YX_1|X_{-1}}\right)^{\frac{1}{2}}}$$

which is assumed to have a null distribution of $t(N-2-p^\star)$.

If the X and Y variables are assumed to have a joint multivariate normal distribution, then the exact power is given by the following formula:

$$\text{power} = \begin{cases} P\left[\dfrac{(N-2-p^\star)^{\frac{1}{2}} R_{YX_1|X_{-1}}}{\left(1-R^2_{YX_1|X_{-1}}\right)^{\frac{1}{2}}} \geq t_{1-\alpha}(N-2-p^\star)\right], & \text{upper one-sided} \\ P\left[\dfrac{(N-2-p^\star)^{\frac{1}{2}} R_{YX_1|X_{-1}}}{\left(1-R^2_{YX_1|X_{-1}}\right)^{\frac{1}{2}}} \leq t_{\alpha}(N-2-p^\star)\right], & \text{lower one-sided} \end{cases}$$

$$= \begin{cases} P\left[R_{Y|(X_1,X_{-1})} \geq \dfrac{t_{1-\alpha}(N-2-p^\star)}{\left(t^2_{1-\alpha}(N-2-p^\star)+N-2-p^\star\right)^{\frac{1}{2}}}\right], & \text{upper one-sided} \\ P\left[R_{Y|(X_1,X_{-1})} \leq \dfrac{t_{\alpha}(N-2-p^\star)}{\left(t^2_{\alpha}(N-2-p^\star)+N-2-p^\star\right)^{\frac{1}{2}}}\right], & \text{lower one-sided} \end{cases}$$

The distribution of $R_{Y|(X_1,X_{-1})}$ (given the underlying true correlation $\rho_{Y|(X_1,X_{-1})}$) is given in Chapter 32 of Johnson, Kotz, and Balakrishnan (1995).

If the X variables are assumed to have fixed values, then the exact power is given by the noncentral t distribution $t(N-2-p^\star, \delta)$, where the noncentrality is

$$\delta = N^{\frac{1}{2}} \frac{\rho_{YX_1|X_{-1}}}{\left(1-\rho^2_{YX_1|X_{-1}}\right)^{\frac{1}{2}}}$$

The power is

$$\text{power} = \begin{cases} P\left(t(N-2-p^\star, \delta) \geq t_{1-\alpha}(N-2-p^\star)\right), & \text{upper one-sided} \\ P\left(t(N-2-p^\star, \delta) \leq t_{\alpha}(N-2-p^\star)\right), & \text{lower one-sided} \end{cases}$$

Analyses in the ONESAMPLEFREQ Statement

Exact Test of a Binomial Proportion (TEST=EXACT)

Let X be distributed as Bin(N, p). The hypotheses for the test of the proportion p are as follows:

$$H_0: p = p_0$$

$$H_1: \begin{cases} p \neq p_0, & \text{two-sided} \\ p > p_0, & \text{upper one-sided} \\ p < p_0, & \text{lower one-sided} \end{cases}$$

The exact test assumes binomially distributed data and requires $N \geq 1$ and $0 < p_0 < 1$. The test statistic is

$$X = \text{number of successes} \sim \text{Bin}(N, p)$$

The significance probability α is split symmetrically for two-sided tests, in the sense that each tail is filled with as much as possible up to $\alpha/2$.

Exact power computations are based on the binomial distribution and computing formulas such as the following from Johnson, Kotz, and Kemp (1992, equation 3.20):

$$P(X \geq C|N,p) = P\left(F_{\nu_1,\nu_2} \leq \frac{\nu_2 p}{\nu_1(1-p)}\right) \quad \text{where } \nu_1 = 2C \text{ and } \nu_2 = 2(N-C+1)$$

Let C_L and C_U denote lower and upper critical values, respectively. Let α_a denote the achieved (actual) significance level, which for two-sided tests is the sum of the favorable major tail (α_M) and the opposite minor tail (α_m).

For the upper one-sided case,

$$\begin{aligned} C_U &= \min\{C : P(X \geq C|p_0) \leq \alpha\} \\ \text{Reject } H_0 \text{ if } & X \geq C_U \\ \alpha_a &= P(X \geq C_U|p_0) \\ \text{power} &= P(X \geq C_U|p) \end{aligned}$$

For the lower one-sided case,

$$\begin{aligned} C_L &= \max\{C : P(X \leq C|p_0) \leq \alpha\} \\ \text{Reject } H_0 \text{ if } & X \leq C_L \\ \alpha_a &= P(X \leq C_L|p_0) \\ \text{power} &= P(X \leq C_L|p) \end{aligned}$$

For the two-sided case,

$$\begin{aligned} C_L &= \max\{C : P(X \leq C|p_0) \leq \frac{\alpha}{2}\} \\ C_U &= \min\{C : P(X \geq C|p_0) \leq \frac{\alpha}{2}\} \\ \text{Reject } H_0 \text{ if } & X \leq C_L \text{ or } X \geq C_U \\ \alpha_a &= P(X \leq C_L \text{ or } X \geq C_U|p_0) \\ \text{power} &= P(X \leq C_L \text{ or } X \geq C_U|p) \end{aligned}$$

z Test for Binomial Proportion Using Null Variance (TEST=Z VAREST=NULL)

For the normal approximation test, the test statistic is

$$Z(X) = \frac{X - Np_0}{[Np_0(1-p_0)]^{\frac{1}{2}}}$$

For the METHOD=EXACT option, the computations are the same as described in the section "Exact Test of a Binomial Proportion (TEST=EXACT)" on page 4934 except for the definitions of the critical values.

For the upper one-sided case,

$$C_U = \min\{C : Z(C) \geq z_{1-\alpha}\}$$

For the lower one-sided case,

$$C_L = \max\{C : Z(C) \leq z_\alpha\}$$

For the two-sided case,

$$C_L = \max\{C : Z(C) \leq z_{\frac{\alpha}{2}}\}$$
$$C_U = \min\{C : Z(C) \geq z_{1-\frac{\alpha}{2}}\}$$

For the METHOD=NORMAL option, the test statistic $Z(X)$ is assumed to have the normal distribution

$$\mathrm{N}\left(\frac{N^{\frac{1}{2}}(p-p_0)}{[p_0(1-p_0)]^{\frac{1}{2}}}, \frac{p(1-p)}{p_0(1-p_0)}\right)$$

The approximate power is computed as

$$\text{power} = \begin{cases} \Phi\left(\dfrac{z_\alpha + \sqrt{N}\frac{p-p_0}{\sqrt{p_0(1-p_0)}}}{\sqrt{\frac{p(1-p)}{p_0(1-p_0)}}}\right), & \text{upper one-sided} \\ \Phi\left(\dfrac{z_\alpha - \sqrt{N}\frac{p-p_0}{\sqrt{p_0(1-p_0)}}}{\sqrt{\frac{p(1-p)}{p_0(1-p_0)}}}\right), & \text{lower one-sided} \\ \Phi\left(\dfrac{z_{\frac{\alpha}{2}} + \sqrt{N}\frac{p-p_0}{\sqrt{p_0(1-p_0)}}}{\sqrt{\frac{p(1-p)}{p_0(1-p_0)}}}\right) + \Phi\left(\dfrac{z_{\frac{\alpha}{2}} - \sqrt{N}\frac{p-p_0}{\sqrt{p_0(1-p_0)}}}{\sqrt{\frac{p(1-p)}{p_0(1-p_0)}}}\right), & \text{two-sided} \end{cases}$$

The approximate sample size is computed in closed form for the one-sided cases by inverting the power equation,

$$N = \left(\frac{z_{\text{power}}\sqrt{p(1-p)} + z_{1-\alpha}\sqrt{p_0(1-p_0)}}{p-p_0}\right)^2$$

and by numerical inversion for the two-sided case.

z Test for Binomial Proportion Using Sample Variance (TEST=Z VAREST=SAMPLE)

For the normal approximation test using the sample variance, the test statistic is

$$Z_s(X) = \frac{X - Np_0}{[N\hat{p}(1-\hat{p})]^{\frac{1}{2}}}$$

where $\hat{p} = X/N$.

For the METHOD=EXACT option, the computations are the same as described in the section "Exact Test of a Binomial Proportion (TEST=EXACT)" on page 4934 except for the definitions of the critical values.

For the upper one-sided case,

$$C_U = \min\{C : Z_s(C) \geq z_{1-\alpha}\}$$

For the lower one-sided case,

$$C_L = \max\{C : Z_s(C) \leq z_{\alpha}\}$$

For the two-sided case,

$$C_L = \max\{C : Z_s(C) \leq z_{\frac{\alpha}{2}}\}$$
$$C_U = \min\{C : Z_s(C) \geq z_{1-\frac{\alpha}{2}}\}$$

For the METHOD=NORMAL option, the test statistic $Z_s(X)$ is assumed to have the normal distribution

$$\mathrm{N}\left(\frac{N^{\frac{1}{2}}(p-p_0)}{[p(1-p)]^{\frac{1}{2}}}, 1\right)$$

(see Chow, Shao, and Wang (2003), p. 82).

The approximate power is computed as

$$\text{power} = \begin{cases} \Phi\left(z_\alpha + \sqrt{N}\frac{p-p_0}{\sqrt{p(1-p)}}\right), & \text{upper one-sided} \\ \Phi\left(z_\alpha - \sqrt{N}\frac{p-p_0}{\sqrt{p(1-p)}}\right), & \text{lower one-sided} \\ \Phi\left(z_{\frac{\alpha}{2}} + \sqrt{N}\frac{p-p_0}{\sqrt{p(1-p)}}\right) + \Phi\left(z_{\frac{\alpha}{2}} - \sqrt{N}\frac{p-p_0}{\sqrt{p(1-p)}}\right), & \text{two-sided} \end{cases}$$

The approximate sample size is computed in closed form for the one-sided cases by inverting the power equation,

$$N = p(1-p)\left(\frac{z_{\text{power}} + z_{1-\alpha}}{p-p_0}\right)^2$$

and by numerical inversion for the two-sided case.

z Test for Binomial Proportion with Continuity Adjustment Using Null Variance (TEST=ADJZ VAREST=NULL)

For the normal approximation test with continuity adjustment, the test statistic is (Pagano and Gauvreau 1993 p. 295):

$$Z_c(X) = \frac{X - Np_0 + 0.5(1_{\{X<Np_0\}}) - 0.5(1_{\{X>Np_0\}})}{[Np_0(1-p_0)]^{\frac{1}{2}}}$$

For the METHOD=EXACT option, the computations are the same as described in the section "Exact Test of a Binomial Proportion (TEST=EXACT)" on page 4934 except for the definitions of the critical values.

For the upper one-sided case,

$$C_U = \min\{C : Z_c(C) \geq z_{1-\alpha}\}$$

For the lower one-sided case,

$$C_L = \max\{C : Z_c(C) \leq z_{\alpha}\}$$

For the two-sided case,

$$C_L = \max\{C : Z_c(C) \leq z_{\frac{\alpha}{2}}\}$$
$$C_U = \min\{C : Z_c(C) \geq z_{1-\frac{\alpha}{2}}\}$$

For the METHOD=NORMAL option, the test statistic $Z_c(X)$ is assumed to have the normal distribution $N(\mu, \sigma^2)$, where μ and σ^2 are derived as follows.

For convenience of notation, define

$$k = \frac{1}{2\sqrt{Np_0(1-p_0)}}$$

Then

$$E\left[Z_c(X)\right] = 2kNp - 2kNp_0 + kP(X < Np_0) - kP(X > Np_0)$$

and

$$\begin{aligned}\mathrm{Var}\left[Z_c(X)\right] = {} & 4k^2Np(1-p) + k^2\left[1 - P(X = Np_0)\right] - k^2\left[P(X < Np_0) - P(X > Np_0)\right]^2 \\ & + 4k^2\left[E\left(X1_{\{X<Np_0\}}\right) - E\left(X1_{\{X>Np_0\}}\right)\right] - 4k^2Np\left[P(X < Np_0) - P(X > Np_0)\right]\end{aligned}$$

The probabilities $P(X = Np_0)$, $P(X < Np_0)$, and $P(X > Np_0)$ and the truncated expectations $E\left(X1_{\{X<Np_0\}}\right)$ and $E\left(X1_{\{X>Np_0\}}\right)$ are approximated by assuming the normal-approximate distribution of X, $N(Np, Np(1-p))$. Letting $\phi(\cdot)$ and $\Phi(\cdot)$ denote the standard normal PDF and CDF, respectively, and defining d as

$$d = \frac{Np_0 - Np}{[Np(1-p)]^{\frac{1}{2}}}$$

the terms are computed as follows:

$$\begin{aligned}P(X = Np_0) &= 0 \\ P(X < Np_0) &= \Phi(d) \\ P(X > Np_0) &= 1 - \Phi(d) \\ E\left(X1_{\{X<Np_0\}}\right) &= Np\Phi(d) - [Np(1-p)]^{\frac{1}{2}}\phi(d) \\ E\left(X1_{\{X>Np_0\}}\right) &= Np\left[1 - \Phi(d)\right] + [Np(1-p)]^{\frac{1}{2}}\phi(d)\end{aligned}$$

The mean and variance of $Z_c(X)$ are thus approximated by

$$\mu = k\left[2Np - 2Np_0 + 2\Phi(d) - 1\right]$$

and

$$\sigma^2 = 4k^2\left[Np(1-p) + \Phi(d)\left(1-\Phi(d)\right) - 2\left(Np(1-p)\right)^{\frac{1}{2}}\phi(d)\right]$$

The approximate power is computed as

$$\text{power} = \begin{cases} \Phi\left(\frac{z_\alpha+\mu}{\sigma}\right), & \text{upper one-sided} \\ \Phi\left(\frac{z_\alpha-\mu}{\sigma}\right), & \text{lower one-sided} \\ \Phi\left(\frac{z_{\frac{\alpha}{2}}+\mu}{\sigma}\right) + \Phi\left(\frac{z_{\frac{\alpha}{2}}-\mu}{\sigma}\right), & \text{two-sided} \end{cases}$$

The approximate sample size is computed by numerical inversion.

z Test for Binomial Proportion with Continuity Adjustment Using Sample Variance (TEST=ADJZ VAREST=SAMPLE)

For the normal approximation test with continuity adjustment using the sample variance, the test statistic is

$$Z_{cs}(X) = \frac{X - Np_0 + 0.5(1_{\{X<Np_0\}}) - 0.5(1_{\{X>Np_0\}})}{\left[N\hat{p}(1-\hat{p})\right]^{\frac{1}{2}}}$$

where $\hat{p} = X/N$.

For the METHOD=EXACT option, the computations are the same as described in the section "Exact Test of a Binomial Proportion (TEST=EXACT)" on page 4934 except for the definitions of the critical values.

For the upper one-sided case,

$$C_U = \min\{C : Z_{cs}(C) \geq z_{1-\alpha}\}$$

For the lower one-sided case,

$$C_L = \max\{C : Z_{cs}(C) \leq z_\alpha\}$$

For the two-sided case,

$$C_L = \max\{C : Z_{cs}(C) \leq z_{\frac{\alpha}{2}}\}$$
$$C_U = \min\{C : Z_{cs}(C) \geq z_{1-\frac{\alpha}{2}}\}$$

For the METHOD=NORMAL option, the test statistic $Z_{cs}(X)$ is assumed to have the normal distribution $N(\mu, \sigma^2)$, where μ and σ^2 are derived as follows.

For convenience of notation, define

$$k = \frac{1}{2\sqrt{Np(1-p)}}$$

Then

$$E\left[Z_{cs}(X)\right] \approx 2kNp - 2kNp_0 + kP(X < Np_0) - kP(X > Np_0)$$

and

$$\begin{aligned}\mathrm{Var}\left[Z_{cs}(X)\right] \approx\ & 4k^2Np(1-p) + k^2\left[1 - P(X = Np_0)\right] - k^2\left[P(X < Np_0) - P(X > Np_0)\right]^2 \\ & + 4k^2\left[E\left(X1_{\{X<Np_0\}}\right) - E\left(X1_{\{X>Np_0\}}\right)\right] - 4k^2Np\left[P(X < Np_0) - P(X > Np_0)\right]\end{aligned}$$

The probabilities $P(X = Np_0)$, $P(X < Np_0)$, and $P(X > Np_0)$ and the truncated expectations $E\left(X1_{\{X<Np_0\}}\right)$ and $E\left(X1_{\{X>Np_0\}}\right)$ are approximated by assuming the normal-approximate distribution of X, $N(Np, Np(1-p))$. Letting $\phi(\cdot)$ and $\Phi(\cdot)$ denote the standard normal PDF and CDF, respectively, and defining d as

$$d = \frac{Np_0 - Np}{[Np(1-p)]^{\frac{1}{2}}}$$

the terms are computed as follows:

$$\begin{aligned}P(X = Np_0) &= 0 \\ P(X < Np_0) &= \Phi(d) \\ P(X > Np_0) &= 1 - \Phi(d) \\ E\left(X1_{\{X<Np_0\}}\right) &= Np\Phi(d) - [Np(1-p)]^{\frac{1}{2}}\phi(d) \\ E\left(X1_{\{X>Np_0\}}\right) &= Np\left[1 - \Phi(d)\right] + [Np(1-p)]^{\frac{1}{2}}\phi(d)\end{aligned}$$

The mean and variance of $Z_{cs}(X)$ are thus approximated by

$$\mu = k\left[2Np - 2Np_0 + 2\Phi(d) - 1\right]$$

and

$$\sigma^2 = 4k^2\left[Np(1-p) + \Phi(d)\left(1 - \Phi(d)\right) - 2\left(Np(1-p)\right)^{\frac{1}{2}}\phi(d)\right]$$

The approximate power is computed as

$$\text{power} = \begin{cases} \Phi\left(\frac{z_\alpha + \mu}{\sigma}\right), & \text{upper one-sided} \\ \Phi\left(\frac{z_\alpha - \mu}{\sigma}\right), & \text{lower one-sided} \\ \Phi\left(\frac{z_{\frac{\alpha}{2}} + \mu}{\sigma}\right) + \Phi\left(\frac{z_{\frac{\alpha}{2}} - \mu}{\sigma}\right), & \text{two-sided} \end{cases}$$

The approximate sample size is computed by numerical inversion.

Exact Equivalence Test of a Binomial Proportion (TEST=EQUIV_EXACT)

The hypotheses for the equivalence test are

$$H_0: p < \theta_L \quad \text{or} \quad p > \theta_U$$
$$H_1: \theta_L \leq p \leq \theta_U$$

where θ_L and θ_U are the lower and upper equivalence bounds, respectively.

The analysis is the two one-sided tests (TOST) procedure as described in Chow, Shao, and Wang (2003) on p. 84, but using exact critical values as on p. 116 instead of normal-based critical values.

Two different hypothesis tests are carried out:

$$H_{a0}: p < \theta_L$$
$$H_{a1}: p \geq \theta_L$$

and

$$H_{b0}: p > \theta_U$$
$$H_{b1}: p \leq \theta_U$$

If H_{a0} is rejected in favor of H_{a1} *and* H_{b0} is rejected in favor of H_{b1}, then H_0 is rejected in favor of H_1.

The test statistic for each of the two tests (H_{a0} versus H_{a1} and H_{b0} versus H_{b1}) is

$$X = \text{number of successes} \sim \text{Bin}(N, p)$$

Let C_U denote the critical value of the exact upper one-sided test of H_{a0} versus H_{a1}, and let C_L denote the critical value of the exact lower one-sided test of H_{b0} versus H_{b1}. These critical values are computed in the section "Exact Test of a Binomial Proportion (TEST=EXACT)" on page 4934. Both of these tests are rejected if and only if $C_U \leq X \leq C_L$. Thus, the exact power of the equivalence test is

$$\begin{aligned} \text{power} &= P\left(C_U \leq X \leq C_L\right) \\ &= P\left(X \geq C_U\right) - P\left(X \geq C_L + 1\right) \end{aligned}$$

The probabilities are computed using Johnson and Kotz (1970, equation 3.20).

z Equivalence Test for Binomial Proportion Using Null Variance (TEST=EQUIV_Z VAREST=NULL)

The hypotheses for the equivalence test are

$$H_0: p < \theta_L \quad \text{or} \quad p > \theta_U$$
$$H_1: \theta_L \leq p \leq \theta_U$$

where θ_L and θ_U are the lower and upper equivalence bounds, respectively.

The analysis is the two one-sided tests (TOST) procedure as described in Chow, Shao, and Wang (2003) on p. 84, but using the null variance instead of the sample variance.

Two different hypothesis tests are carried out:

$$H_{a0}: p < \theta_L$$
$$H_{a1}: p \geq \theta_L$$

and

$$H_{b0}: p > \theta_U$$
$$H_{b1}: p \leq \theta_U$$

If H_{a0} is rejected in favor of H_{a1} *and* H_{b0} is rejected in favor of H_{b1}, then H_0 is rejected in favor of H_1.

The test statistic for the test of H_{a0} versus H_{a1} is

$$Z_L(X) = \frac{X - N\theta_L}{[N\theta_L(1-\theta_L)]^{\frac{1}{2}}}$$

The test statistic for the test of H_{b0} versus H_{b1} is

$$Z_U(X) = \frac{X - N\theta_U}{[N\theta_U(1-\theta_U)]^{\frac{1}{2}}}$$

For the METHOD=EXACT option, let C_U denote the critical value of the exact upper one-sided test of H_{a0} versus H_{a1} using $Z_L(X)$. This critical value is computed in the section "z Test for Binomial Proportion Using Null Variance (TEST=Z VAREST=NULL)" on page 4935. Similarly, let C_L denote the critical value of the exact lower one-sided test of H_{b0} versus H_{b1} using $Z_U(X)$. Both of these tests are rejected if and only if $C_U \leq X \leq C_L$. Thus, the exact power of the equivalence test is

$$\begin{aligned}\text{power} &= P\left(C_U \leq X \leq C_L\right) \\ &= P\left(X \geq C_U\right) - P\left(X \geq C_L + 1\right)\end{aligned}$$

The probabilities are computed using Johnson and Kotz (1970, equation 3.20).

For the METHOD=NORMAL option, the test statistic $Z_L(X)$ is assumed to have the normal distribution

$$\mathrm{N}\left(\frac{N^{\frac{1}{2}}(p-\theta_L)}{[\theta_L(1-\theta_L)]^{\frac{1}{2}}}, \frac{p(1-p)}{\theta_L(1-\theta_L)}\right)$$

and the test statistic $Z_U(X)$ is assumed to have the normal distribution

$$\mathrm{N}\left(\frac{N^{\frac{1}{2}}(p-\theta_U)}{[\theta_U(1-\theta_U)]^{\frac{1}{2}}}, \frac{p(1-p)}{\theta_U(1-\theta_U)}\right)$$

(see Chow, Shao, and Wang (2003), p. 84).

The approximate power is computed as

$$\text{power} = \Phi\left(\frac{z_\alpha - \sqrt{N}\frac{p-\theta_U}{\sqrt{\theta_U(1-\theta_U)}}}{\sqrt{\frac{p(1-p)}{\theta_U(1-\theta_U)}}}\right) + \Phi\left(\frac{z_\alpha + \sqrt{N}\frac{p-\theta_L}{\sqrt{\theta_L(1-\theta_L)}}}{\sqrt{\frac{p(1-p)}{\theta_L(1-\theta_L)}}}\right) - 1$$

The approximate sample size is computed by numerically inverting the power formula, using the sample size estimate N_0 of Chow, Shao, and Wang (2003, p. 85) as an initial guess:

$$N_0 = p(1-p)\left(\frac{z_{1-\alpha} + z_{(1+\text{power})/2}}{0.5(\theta_U - \theta_L) - \mid p - 0.5(\theta_L + \theta_U)\mid}\right)^2$$

z Equivalence Test for Binomial Proportion Using Sample Variance (TEST=EQUIV_Z VAREST=SAMPLE)

The hypotheses for the equivalence test are

$$H_0: p < \theta_L \quad \text{or} \quad p > \theta_U$$
$$H_1: \theta_L \le p \le \theta_U$$

where θ_L and θ_U are the lower and upper equivalence bounds, respectively.

The analysis is the two one-sided tests (TOST) procedure as described in Chow, Shao, and Wang (2003) on p. 84.

Two different hypothesis tests are carried out:

$$H_{a0}: p < \theta_L$$
$$H_{a1}: p \ge \theta_L$$

and

$$H_{b0}: p > \theta_U$$
$$H_{b1}: p \le \theta_U$$

If H_{a0} is rejected in favor of H_{a1} *and* H_{b0} is rejected in favor of H_{b1}, then H_0 is rejected in favor of H_1.

The test statistic for the test of H_{a0} versus H_{a1} is

$$Z_{sL}(X) = \frac{X - N\theta_L}{[N\hat{p}(1-\hat{p})]^{\frac{1}{2}}}$$

where $\hat{p} = X/N$.

The test statistic for the test of H_{b0} versus H_{b1} is

$$Z_{sU}(X) = \frac{X - N\theta_U}{[N\hat{p}(1-\hat{p})]^{\frac{1}{2}}}$$

For the METHOD=EXACT option, let C_U denote the critical value of the exact upper one-sided test of H_{a0} versus H_{a1} using $Z_{sL}(X)$. This critical value is computed in the section "z Test for Binomial Proportion Using Sample Variance (TEST=Z VAREST=SAMPLE)" on page 4936. Similarly, let C_L denote the critical value of the exact lower one-sided test of H_{b0} versus H_{b1} using $Z_{sU}(X)$. Both of these tests are rejected if and only if $C_U \leq X \leq C_L$. Thus, the exact power of the equivalence test is

$$\begin{aligned}\text{power} &= P\left(C_U \leq X \leq C_L\right) \\ &= P\left(X \geq C_U\right) - P\left(X \geq C_L + 1\right)\end{aligned}$$

The probabilities are computed using Johnson and Kotz (1970, equation 3.20).

For the METHOD=NORMAL option, the test statistic $Z_{sL}(X)$ is assumed to have the normal distribution

$$\mathrm{N}\left(\frac{N^{\frac{1}{2}}(p-\theta_L)}{[p(1-p)]^{\frac{1}{2}}}, 1\right)$$

and the test statistic $Z_{sU}(X)$ is assumed to have the normal distribution

$$\mathrm{N}\left(\frac{N^{\frac{1}{2}}(p-\theta_U)}{[p(1-p)]^{\frac{1}{2}}}, 1\right)$$

(see Chow, Shao, and Wang (2003), p. 84).

The approximate power is computed as

$$\text{power} = \Phi\left(z_\alpha - \sqrt{N}\frac{p-\theta_U}{\sqrt{p(1-p)}}\right) + \Phi\left(z_\alpha + \sqrt{N}\frac{p-\theta_L}{\sqrt{p(1-p)}}\right) - 1$$

The approximate sample size is computed by numerically inverting the power formula, using the sample size estimate N_0 of Chow, Shao, and Wang (2003, p. 85) as an initial guess:

$$N_0 = p(1-p)\left(\frac{z_{1-\alpha} + z_{(1+\text{power})/2}}{0.5(\theta_U - \theta_L) - \mid p - 0.5(\theta_L + \theta_U)\mid}\right)^2$$

z Equivalence Test for Binomial Proportion with Continuity Adjustment Using Null Variance (TEST=EQUIV_ADJZ VAREST=NULL)

The hypotheses for the equivalence test are

$$\begin{aligned}H_0&: p < \theta_L \quad \text{or} \quad p > \theta_U \\ H_1&: \theta_L \leq p \leq \theta_U\end{aligned}$$

where θ_L and θ_U are the lower and upper equivalence bounds, respectively.

The analysis is the two one-sided tests (TOST) procedure as described in Chow, Shao, and Wang (2003) on p. 84, but using the null variance instead of the sample variance.

Two different hypothesis tests are carried out:

$$H_{a0}: p < \theta_L$$
$$H_{a1}: p \geq \theta_L$$

and

$$H_{b0}: p > \theta_U$$
$$H_{b1}: p \leq \theta_U$$

If H_{a0} is rejected in favor of H_{a1} *and* H_{b0} is rejected in favor of H_{b1}, then H_0 is rejected in favor of H_1.

The test statistic for the test of H_{a0} versus H_{a1} is

$$Z_{cL}(X) = \frac{X - N\theta_L + 0.5(1_{\{X<N\theta_L\}}) - 0.5(1_{\{X>N\theta_L\}})}{\left[N\hat{\theta_L}(1-\hat{\theta_L})\right]^{\frac{1}{2}}}$$

where $\hat{p} = X/N$.

The test statistic for the test of H_{b0} versus H_{b1} is

$$Z_{cU}(X) = \frac{X - N\theta_U + 0.5(1_{\{X<N\theta_U\}}) - 0.5(1_{\{X>N\theta_U\}})}{\left[N\hat{\theta_U}(1-\hat{\theta_U})\right]^{\frac{1}{2}}}$$

For the METHOD=EXACT option, let C_U denote the critical value of the exact upper one-sided test of H_{a0} versus H_{a1} using $Z_{cL}(X)$. This critical value is computed in the section "z Test for Binomial Proportion with Continuity Adjustment Using Null Variance (TEST=ADJZ VAREST=NULL)" on page 4937. Similarly, let C_L denote the critical value of the exact lower one-sided test of H_{b0} versus H_{b1} using $Z_{cU}(X)$. Both of these tests are rejected if and only if $C_U \leq X \leq C_L$. Thus, the exact power of the equivalence test is

$$\begin{aligned}\text{power} &= P\left(C_U \leq X \leq C_L\right) \\ &= P\left(X \geq C_U\right) - P\left(X \geq C_L + 1\right)\end{aligned}$$

The probabilities are computed using Johnson and Kotz (1970, equation 3.20).

For the METHOD=NORMAL option, the test statistic $Z_{cL}(X)$ is assumed to have the normal distribution $N(\mu_L, \sigma_L^2)$, and $Z_{cU}(X)$ is assumed to have the normal distribution $N(\mu_U, \sigma_U^2)$, where μ_L, μ_U, σ_L^2, and σ_U^2 are derived as follows.

For convenience of notation, define

$$k_L = \frac{1}{2\sqrt{N\theta_L(1-\theta_L)}}$$
$$k_U = \frac{1}{2\sqrt{N\theta_U(1-\theta_U)}}$$

Then

$$E\left[Z_{cL}(X)\right] \approx 2k_L Np - 2k_L N\theta_L + k_L P(X < N\theta_L) - k_L P(X > N\theta_L)$$
$$E\left[Z_{cU}(X)\right] \approx 2k_U Np - 2k_U N\theta_U + k_U P(X < N\theta_U) - k_U P(X > N\theta_U)$$

and

$$\begin{aligned}\mathrm{Var}\left[Z_{cL}(X)\right] &\approx 4k_L^2 Np(1-p) + k_L^2\left[1 - P(X = N\theta_L)\right] - k_L^2\left[P(X < N\theta_L) - P(X > N\theta_L)\right]^2 \\ &\quad + 4k_L^2\left[E\left(X1_{\{X<N\theta_L\}}\right) - E\left(X1_{\{X>N\theta_L\}}\right)\right] - 4k_L^2 Np\left[P(X < N\theta_L) - P(X > N\theta_L)\right] \\ \mathrm{Var}\left[Z_{cU}(X)\right] &\approx 4k_U^2 Np(1-p) + k_U^2\left[1 - P(X = N\theta_U)\right] - k_U^2\left[P(X < N\theta_U) - P(X > N\theta_U)\right]^2 \\ &\quad + 4k_U^2\left[E\left(X1_{\{X<N\theta_U\}}\right) - E\left(X1_{\{X>N\theta_U\}}\right)\right] - 4k_U^2 Np\left[P(X < N\theta_U) - P(X > N\theta_U)\right]\end{aligned}$$

The probabilities $P(X = N\theta_L)$, $P(X < N\theta_L)$, $P(X > N\theta_L)$, $P(X = N\theta_U)$, $P(X < N\theta_U)$, and $P(X > N\theta_U)$ and the truncated expectations $E\left(X1_{\{X<N\theta_L\}}\right)$, $E\left(X1_{\{X>N\theta_L\}}\right)$, $E\left(X1_{\{X<N\theta_L\}}\right)$, and $E\left(X1_{\{X>N\theta_L\}}\right)$ are approximated by assuming the normal-approximate distribution of X, $N(Np, Np(1-p))$. Letting $\phi(\cdot)$ and $\Phi(\cdot)$ denote the standard normal PDF and CDF, respectively, and defining d_L and d_U as

$$d_L = \frac{N\theta_L - Np}{[Np(1-p)]^{\frac{1}{2}}}$$
$$d_U = \frac{N\theta_U - Np}{[Np(1-p)]^{\frac{1}{2}}}$$

the terms are computed as follows:

$$\begin{aligned}P(X = N\theta_L) &= 0 \\ P(X = N\theta_U) &= 0 \\ P(X < N\theta_L) &= \Phi(d_L) \\ P(X < N\theta_U) &= \Phi(d_U) \\ P(X > N\theta_L) &= 1 - \Phi(d_L) \\ P(X > N\theta_U) &= 1 - \Phi(d_U) \\ E\left(X1_{\{X<N\theta_L\}}\right) &= Np\Phi(d_L) - [Np(1-p)]^{\frac{1}{2}}\phi(d_L) \\ E\left(X1_{\{X<N\theta_U\}}\right) &= Np\Phi(d_U) - [Np(1-p)]^{\frac{1}{2}}\phi(d_U) \\ E\left(X1_{\{X>N\theta_L\}}\right) &= Np\left[1 - \Phi(d_L)\right] + [Np(1-p)]^{\frac{1}{2}}\phi(d_L) \\ E\left(X1_{\{X>N\theta_U\}}\right) &= Np\left[1 - \Phi(d_U)\right] + [Np(1-p)]^{\frac{1}{2}}\phi(d_U)\end{aligned}$$

The mean and variance of $Z_{cL}(X)$ and $Z_{cU}(X)$ are thus approximated by

$$\mu_L = k_L\left[2Np - 2N\theta_L + 2\Phi(d_L) - 1\right]$$
$$\mu_U = k_U\left[2Np - 2N\theta_U + 2\Phi(d_U) - 1\right]$$

and

$$\sigma_L^2 = 4k_L^2\left[Np(1-p) + \Phi(d_L)\left(1-\Phi(d_L)\right) - 2\left(Np(1-p)\right)^{\frac{1}{2}}\phi(d_L)\right]$$
$$\sigma_U^2 = 4k_U^2\left[Np(1-p) + \Phi(d_U)\left(1-\Phi(d_U)\right) - 2\left(Np(1-p)\right)^{\frac{1}{2}}\phi(d_U)\right]$$

The approximate power is computed as

$$\text{power} = \Phi\left(\frac{z_\alpha - \mu_U}{\sigma_U}\right) + \Phi\left(\frac{z_\alpha + \mu_L}{\sigma_L}\right) - 1$$

The approximate sample size is computed by numerically inverting the power formula.

z Equivalence Test for Binomial Proportion with Continuity Adjustment Using Sample Variance (TEST=EQUIV_ADJZ VAREST=SAMPLE)

The hypotheses for the equivalence test are

$$H_0{:}p < \theta_L \quad \text{or} \quad p > \theta_U$$
$$H_1{:}\theta_L \le p \le \theta_U$$

where θ_L and θ_U are the lower and upper equivalence bounds, respectively.

The analysis is the two one-sided tests (TOST) procedure as described in Chow, Shao, and Wang (2003) on p. 84.

Two different hypothesis tests are carried out:

$$H_{a0}{:}p < \theta_L$$
$$H_{a1}{:}p \ge \theta_L$$

and

$$H_{b0}{:}p > \theta_U$$
$$H_{b1}{:}p \le \theta_U$$

If H_{a0} is rejected in favor of H_{a1} *and* H_{b0} is rejected in favor of H_{b1}, then H_0 is rejected in favor of H_1.

The test statistic for the test of H_{a0} versus H_{a1} is

$$Z_{csL}(X) = \frac{X - N\theta_L + 0.5(1_{\{X<N\theta_L\}}) - 0.5(1_{\{X>N\theta_L\}})}{[N\hat{p}(1-\hat{p})]^{\frac{1}{2}}}$$

where $\hat{p} = X/N$.

The test statistic for the test of H_{b0} versus H_{b1} is

$$Z_{csU}(X) = \frac{X - N\theta_U + 0.5(1_{\{X<N\theta_U\}}) - 0.5(1_{\{X>N\theta_U\}})}{[N\hat{p}(1-\hat{p})]^{\frac{1}{2}}}$$

For the METHOD=EXACT option, let C_U denote the critical value of the exact upper one-sided test of H_{a0} versus H_{a1} using $Z_{csL}(X)$. This critical value is computed in the section "z Test for Binomial Proportion with Continuity Adjustment Using Sample Variance (TEST=ADJZ VAREST=SAMPLE)" on page 4939. Similarly, let C_L denote the critical value of the exact lower one-sided test of H_{b0} versus H_{b1} using $Z_{csU}(X)$. Both of these tests are rejected if and only if $C_U \leq X \leq C_L$. Thus, the exact power of the equivalence test is

$$\begin{aligned}\text{power} &= P\left(C_U \leq X \leq C_L\right) \\ &= P\left(X \geq C_U\right) - P\left(X \geq C_L + 1\right)\end{aligned}$$

The probabilities are computed using Johnson and Kotz (1970, equation 3.20).

For the METHOD=NORMAL option, the test statistic $Z_{csL}(X)$ is assumed to have the normal distribution $N(\mu_L, \sigma_L^2)$, and $Z_{csU}(X)$ is assumed to have the normal distribution $N(\mu_U, \sigma_U^2)$, where μ_L, μ_U, σ_L^2 and σ_U^2 are derived as follows.

For convenience of notation, define

$$k = \frac{1}{2\sqrt{Np(1-p)}}$$

Then

$$\begin{aligned}E\left[Z_{csL}(X)\right] &\approx 2kNp - 2kN\theta_L + kP(X < N\theta_L) - kP(X > N\theta_L) \\ E\left[Z_{csU}(X)\right] &\approx 2kNp - 2kN\theta_U + kP(X < N\theta_U) - kP(X > N\theta_U)\end{aligned}$$

and

$$\begin{aligned}\text{Var}\left[Z_{csL}(X)\right] &\approx 4k^2Np(1-p) + k^2\left[1 - P(X = N\theta_L)\right] - k^2\left[P(X < N\theta_L) - P(X > N\theta_L)\right]^2 \\ &\quad + 4k^2\left[E\left(X1_{\{X<N\theta_L\}}\right) - E\left(X1_{\{X>N\theta_L\}}\right)\right] - 4k^2Np\left[P(X < N\theta_L) - P(X > N\theta_L)\right] \\ \text{Var}\left[Z_{csU}(X)\right] &\approx 4k^2Np(1-p) + k^2\left[1 - P(X = N\theta_U)\right] - k^2\left[P(X < N\theta_U) - P(X > N\theta_U)\right]^2 \\ &\quad + 4k^2\left[E\left(X1_{\{X<N\theta_U\}}\right) - E\left(X1_{\{X>N\theta_U\}}\right)\right] - 4k^2Np\left[P(X < N\theta_U) - P(X > N\theta_U)\right]\end{aligned}$$

The probabilities $P(X = N\theta_L)$, $P(X < N\theta_L)$, $P(X > N\theta_L)$, $P(X = N\theta_U)$, $P(X < N\theta_U)$, and $P(X > N\theta_U)$ and the truncated expectations $E\left(X1_{\{X<N\theta_L\}}\right)$, $E\left(X1_{\{X>N\theta_L\}}\right)$, $E\left(X1_{\{X<N\theta_L\}}\right)$, and $E\left(X1_{\{X>N\theta_L\}}\right)$ are approximated by assuming the normal-approximate distribution of X, $N(Np, Np(1-p))$. Letting $\phi(\cdot)$ and $\Phi(\cdot)$ denote the standard normal PDF and CDF, respectively, and defining d_L and d_U as

$$\begin{aligned}d_L &= \frac{N\theta_L - Np}{[Np(1-p)]^{\frac{1}{2}}} \\ d_U &= \frac{N\theta_U - Np}{[Np(1-p)]^{\frac{1}{2}}}\end{aligned}$$

the terms are computed as follows:

$$
\begin{aligned}
P(X = N\theta_L) &= 0 \\
P(X = N\theta_U) &= 0 \\
P(X < N\theta_L) &= \Phi(d_L) \\
P(X < N\theta_U) &= \Phi(d_U) \\
P(X > N\theta_L) &= 1 - \Phi(d_L) \\
P(X > N\theta_U) &= 1 - \Phi(d_U) \\
E\left(X1_{\{X<N\theta_L\}}\right) &= Np\Phi(d_L) - [Np(1-p)]^{\frac{1}{2}}\,\phi(d_L) \\
E\left(X1_{\{X<N\theta_U\}}\right) &= Np\Phi(d_U) - [Np(1-p)]^{\frac{1}{2}}\,\phi(d_U) \\
E\left(X1_{\{X>N\theta_L\}}\right) &= Np\left[1-\Phi(d_L)\right] + [Np(1-p)]^{\frac{1}{2}}\,\phi(d_L) \\
E\left(X1_{\{X>N\theta_U\}}\right) &= Np\left[1-\Phi(d_U)\right] + [Np(1-p)]^{\frac{1}{2}}\,\phi(d_U)
\end{aligned}
$$

The mean and variance of $Z_{csL}(X)$ and $Z_{csU}(X)$ are thus approximated by

$$
\begin{aligned}
\mu_L &= k\left[2Np - 2N\theta_L + 2\Phi(d_L) - 1\right] \\
\mu_U &= k\left[2Np - 2N\theta_U + 2\Phi(d_U) - 1\right]
\end{aligned}
$$

and

$$
\begin{aligned}
\sigma_L^2 &= 4k^2\left[Np(1-p) + \Phi(d_L)\left(1-\Phi(d_L)\right) - 2\left(Np(1-p)\right)^{\frac{1}{2}}\phi(d_L)\right] \\
\sigma_U^2 &= 4k^2\left[Np(1-p) + \Phi(d_U)\left(1-\Phi(d_U)\right) - 2\left(Np(1-p)\right)^{\frac{1}{2}}\phi(d_U)\right]
\end{aligned}
$$

The approximate power is computed as

$$
\text{power} = \Phi\left(\frac{z_\alpha - \mu_U}{\sigma_U}\right) + \Phi\left(\frac{z_\alpha + \mu_L}{\sigma_L}\right) - 1
$$

The approximate sample size is computed by numerically inverting the power formula.

Wilson Score Confidence Interval for Binomial Proportion (CI=WILSON)

The two-sided $100(1-\alpha)\%$ confidence interval for p is

$$
\frac{X + \frac{z_{1-\alpha/2}^2}{2}}{N + z_{1-\alpha/2}^2} \quad \pm \quad \frac{z_{1-\alpha/2}N^{\frac{1}{2}}}{N + z_{1-\alpha/2}^2}\left(\hat{p}(1-\hat{p}) + \frac{z_{1-\alpha/2}^2}{4N}\right)^{\frac{1}{2}}
$$

So the half-width for the two-sided $100(1-\alpha)\%$ confidence interval is

$$\text{half-width} = \frac{z_{1-\alpha/2} N^{\frac{1}{2}}}{N + z^2_{1-\alpha/2}} \left(\hat{p}(1-\hat{p}) + \frac{z^2_{1-\alpha/2}}{4N} \right)^{\frac{1}{2}}$$

Prob(Width) is calculated exactly by adding up the probabilities of observing each $X \in \{1, \ldots, N\}$ that produces a confidence interval whose half-width is at most a target value h:

$$\text{Prob(Width)} = \sum_{i=0}^{N} P(X = i) 1_{\text{half-width} < h}$$

For references and more details about this and all other confidence intervals associated with the CI= option, see "Binomial Proportion" on page 1745 of Chapter 35, "The FREQ Procedure."

Agresti-Coull "Add k Successes and Failures" Confidence Interval for Binomial Proportion (CI=AGRESTICOULL)

The two-sided $100(1-\alpha)\%$ confidence interval for p is

$$\frac{X + \frac{z^2_{1-\alpha/2}}{2}}{N + z^2_{1-\alpha/2}} \quad \pm \quad z_{1-\alpha/2} \left(\frac{\frac{X+\frac{z^2_{1-\alpha/2}}{2}}{N+z^2_{1-\alpha/2}} \left(1 - \frac{X+\frac{z^2_{1-\alpha/2}}{2}}{N+z^2_{1-\alpha/2}} \right)}{N + z^2_{1-\alpha/2}} \right)^{\frac{1}{2}}$$

So the half-width for the two-sided $100(1-\alpha)\%$ confidence interval is

$$\text{half-width} = z_{1-\alpha/2} \left(\frac{\frac{X+\frac{z^2_{1-\alpha/2}}{2}}{N+z^2_{1-\alpha/2}} \left(1 - \frac{X+\frac{z^2_{1-\alpha/2}}{2}}{N+z^2_{1-\alpha/2}} \right)}{N + z^2_{1-\alpha/2}} \right)^{\frac{1}{2}}$$

Prob(Width) is calculated exactly by adding up the probabilities of observing each $X \in \{1, \ldots, N\}$ that produces a confidence interval whose half-width is at most a target value h:

$$\text{Prob(Width)} = \sum_{i=0}^{N} P(X = i) 1_{\text{half-width} < h}$$

Jeffreys Confidence Interval for Binomial Proportion (CI=JEFFREYS)

The two-sided $100(1-\alpha)\%$ confidence interval for p is

$$[L_J(X), U_J(X)]$$

where

$$L_J(X) = \begin{cases} 0, & X = 0 \\ \text{Beta}_{\alpha/2;X+1/2,N-X+1/2}, & X > 0 \end{cases}$$

and

$$U_J(X) = \begin{cases} \text{Beta}_{1-\alpha/2;X+1/2,N-X+1/2}, & X < N \\ 1, & X = N \end{cases}$$

The half-width of this two-sided $100(1-\alpha)\%$ confidence interval is defined as half the width of the full interval:

$$\text{half-width} = \frac{1}{2}\left(U_J(X) - L_J(X)\right)$$

Prob(Width) is calculated exactly by adding up the probabilities of observing each $X \in \{1, \ldots, N\}$ that produces a confidence interval whose half-width is at most a target value h:

$$\text{Prob(Width)} = \sum_{i=0}^{N} P(X = i) 1_{\text{half-width} < h}$$

Exact Clopper-Pearson Confidence Interval for Binomial Proportion (CI=EXACT)

The two-sided $100(1-\alpha)\%$ confidence interval for p is

$$[L_E(X), U_E(X)]$$

where

$$L_E(X) = \begin{cases} 0, & X = 0 \\ \text{Beta}_{\alpha/2;X,N-X+1}, & X > 0 \end{cases}$$

and

$$U_E(X) = \begin{cases} \text{Beta}_{1-\alpha/2;X+1,N-X}, & X < N \\ 1, & X = N \end{cases}$$

The half-width of this two-sided $100(1-\alpha)\%$ confidence interval is defined as half the width of the full interval:

$$\text{half-width} = \frac{1}{2}\left(U_E(X) - L_E(X)\right)$$

Prob(Width) is calculated exactly by adding up the probabilities of observing each $X \in \{1, \ldots, N\}$ that produces a confidence interval whose half-width is at most a target value h:

$$\text{Prob(Width)} = \sum_{i=0}^{N} P(X = i) 1_{\text{half-width}<h}$$

Wald Confidence Interval for Binomial Proportion (CI=WALD)

The two-sided $100(1-\alpha)\%$ confidence interval for p is

$$\hat{p} \quad \pm \quad z_{1-\alpha/2}\left(\frac{\hat{p}(1-\hat{p})}{N}\right)^{\frac{1}{2}}$$

So the half-width for the two-sided $100(1-\alpha)\%$ confidence interval is

$$\text{half-width} = z_{1-\alpha/2}\left(\frac{\hat{p}(1-\hat{p})}{N}\right)^{\frac{1}{2}}$$

Prob(Width) is calculated exactly by adding up the probabilities of observing each $X \in \{1, \ldots, N\}$ that produces a confidence interval whose half-width is at most a target value h:

$$\text{Prob(Width)} = \sum_{i=0}^{N} P(X = i) 1_{\text{half-width}<h}$$

Continuity-Corrected Wald Confidence Interval for Binomial Proportion (CI=WALD_CORRECT)

The two-sided $100(1-\alpha)\%$ confidence interval for p is

$$\hat{p} \quad \pm \quad \left[z_{1-\alpha/2}\left(\frac{\hat{p}(1-\hat{p})}{N}\right)^{\frac{1}{2}} + \frac{1}{2N}\right]$$

So the half-width for the two-sided $100(1-\alpha)\%$ confidence interval is

$$\text{half-width} = z_{1-\alpha/2}\left(\frac{\hat{p}(1-\hat{p})}{N}\right)^{\frac{1}{2}} + \frac{1}{2N}$$

Prob(Width) is calculated exactly by adding up the probabilities of observing each $X \in \{1, \ldots, N\}$ that produces a confidence interval whose half-width is at most a target value h:

$$\text{Prob(Width)} = \sum_{i=0}^{N} P(X = i) 1_{\text{half-width}<h}$$

Analyses in the ONESAMPLEMEANS Statement

One-Sample t *Test (TEST=T)*

The hypotheses for the one-sample t test are

$$H_0{:}\mu = \mu_0$$

$$H_1{:} \begin{cases} \mu \neq \mu_0, & \text{two-sided} \\ \mu > \mu_0, & \text{upper one-sided} \\ \mu < \mu_0, & \text{lower one-sided} \end{cases}$$

The test assumes normally distributed data and requires $N \geq 2$. The test statistics are

$$t = N^{\frac{1}{2}}\left(\frac{\bar{x} - \mu_0}{s}\right) \quad \sim \quad t(N-1, \delta)$$

$$t^2 \sim F(1, N-1, \delta^2)$$

where $\bar{x}$ is the sample mean, s is the sample standard deviation, and

$$\delta = N^{\frac{1}{2}}\left(\frac{\mu - \mu_0}{\sigma}\right)$$

The test is

$$\text{Reject} \quad H_0 \quad \text{if} \begin{cases} t^2 \geq F_{1-\alpha}(1, N-1), & \text{two-sided} \\ t \geq t_{1-\alpha}(N-1), & \text{upper one-sided} \\ t \leq t_{\alpha}(N-1), & \text{lower one-sided} \end{cases}$$

Exact power computations for t tests are discussed in O'Brien and Muller (1993, Section 8.2), although not specifically for the one-sample case. The power is based on the noncentral t and F distributions:

$$\text{power} = \begin{cases} P\left(F(1, N-1, \delta^2) \geq F_{1-\alpha}(1, N-1)\right), & \text{two-sided} \\ P\left(t(N-1, \delta) \geq t_{1-\alpha}(N-1)\right), & \text{upper one-sided} \\ P\left(t(N-1, \delta) \leq t_{\alpha}(N-1)\right), & \text{lower one-sided} \end{cases}$$

Solutions for N, α, and δ are obtained by numerically inverting the power equation. Closed-form solutions for other parameters, in terms of δ, are as follows:

$$\mu = \delta\sigma N^{-\frac{1}{2}} + \mu_0$$

$$\sigma = \begin{cases} \delta^{-1} N^{\frac{1}{2}}(\mu - \mu_0), & |\delta| > 0 \\ \text{undefined}, & \text{otherwise} \end{cases}$$

One-Sample t Test with Lognormal Data (TEST=T DIST=LOGNORMAL)

The lognormal case is handled by reexpressing the analysis equivalently as a normality-based test on the log-transformed data, by using properties of the lognormal distribution as discussed in Johnson, Kotz, and Balakrishnan (1994, Chapter 14). The approaches in the section "One-Sample t Test (TEST=T)" on page 4953 then apply.

In contrast to the usual t test on normal data, the hypotheses with lognormal data are defined in terms of geometric means rather than arithmetic means. This is because the transformation of a null arithmetic mean of lognormal data to the normal scale depends on the unknown coefficient of variation, resulting in an ill-defined hypothesis on the log-transformed data. Geometric means transform cleanly and are more natural for lognormal data.

The hypotheses for the one-sample t test with lognormal data are

$$H_0: \frac{\gamma}{\gamma_0} = 1$$

$$H_1: \begin{cases} \frac{\gamma}{\gamma_0} \neq 1, & \text{two-sided} \\ \frac{\gamma}{\gamma_0} > 1, & \text{upper one-sided} \\ \frac{\gamma}{\gamma_0} < 1, & \text{lower one-sided} \end{cases}$$

Let $\mu^\star$ and $\sigma^\star$ be the (arithmetic) mean and standard deviation of the normal distribution of the log-transformed data. The hypotheses can be rewritten as follows:

$$H_0: \mu^\star = \log(\gamma_0)$$

$$H_1: \begin{cases} \mu^\star \neq \log(\gamma_0), & \text{two-sided} \\ \mu^\star > \log(\gamma_0), & \text{upper one-sided} \\ \mu^\star < \log(\gamma_0), & \text{lower one-sided} \end{cases}$$

where $\mu^\star = \log(\gamma)$.

The test assumes lognormally distributed data and requires $N \geq 2$.

The power is

$$\text{power} = \begin{cases} P\left(F(1, N-1, \delta^2) \geq F_{1-\alpha}(1, N-1)\right), & \text{two-sided} \\ P\left(t(N-1, \delta) \geq t_{1-\alpha}(N-1)\right), & \text{upper one-sided} \\ P\left(t(N-1, \delta) \leq t_{\alpha}(N-1)\right), & \text{lower one-sided} \end{cases}$$

where

$$\delta = N^{\frac{1}{2}}\left(\frac{\mu^\star - \log(\gamma_0)}{\sigma^\star}\right)$$
$$\sigma^\star = \left[\log(\mathrm{CV}^2 + 1)\right]^{\frac{1}{2}}$$

Equivalence Test for Mean of Normal Data (TEST=EQUIV DIST=NORMAL)

The hypotheses for the equivalence test are

$$H_0{:}\mu < \theta_L \quad \text{or} \quad \mu > \theta_U$$
$$H_1{:}\theta_L \le \mu \le \theta_U$$

The analysis is the two one-sided tests (TOST) procedure of Schuirmann (1987). The test assumes normally distributed data and requires $N \ge 2$. Phillips (1990) derives an expression for the exact power assuming a two-sample balanced design; the results are easily adapted to a one-sample design:

$$\text{power} = Q_{N-1}\left((-t_{1-\alpha}(N-1)), \frac{\mu-\theta_U}{\sigma N^{-\frac{1}{2}}}; 0, \frac{(N-1)^{\frac{1}{2}}(\theta_U-\theta_L)}{2\sigma N^{-\frac{1}{2}}(t_{1-\alpha}(N-1))}\right) -$$
$$Q_{N-1}\left((t_{1-\alpha}(N-1)), \frac{\mu-\theta_L}{\sigma N^{-\frac{1}{2}}}; 0, \frac{(N-1)^{\frac{1}{2}}(\theta_U-\theta_L)}{2\sigma N^{-\frac{1}{2}}(t_{1-\alpha}(N-1))}\right)$$

where $Q.(\cdot,\cdot;\cdot,\cdot)$ is Owen's Q function, defined in the section "Common Notation" on page 4926.

Equivalence Test for Mean of Lognormal Data (TEST=EQUIV DIST=LOGNORMAL)

The lognormal case is handled by reexpressing the analysis equivalently as a normality-based test on the log-transformed data, by using properties of the lognormal distribution as discussed in Johnson, Kotz, and Balakrishnan (1994, Chapter 14). The approaches in the section "Equivalence Test for Mean of Normal Data (TEST=EQUIV DIST=NORMAL)" on page 4955 then apply.

In contrast to the additive equivalence test on normal data, the hypotheses with lognormal data are defined in terms of geometric means rather than arithmetic means. This is because the transformation of an arithmetic mean of lognormal data to the normal scale depends on the unknown coefficient of variation, resulting in an ill-defined hypothesis on the log-transformed data. Geometric means transform cleanly and are more natural for lognormal data.

The hypotheses for the equivalence test are

$$H_0{:}\gamma \le \theta_L \quad \text{or} \quad \gamma \ge \theta_U$$
$$H_1{:}\theta_L < \gamma < \theta_U$$

$$\text{where} \quad 0 < \theta_L < \theta_U$$

The analysis is the two one-sided tests (TOST) procedure of Schuirmann (1987) on the log-transformed data. The test assumes lognormally distributed data and requires $N \ge 2$. Diletti,

Hauschke, and Steinijans (1991) derive an expression for the exact power assuming a crossover design; the results are easily adapted to a one-sample design:

$$\text{power} = Q_{N-1}\left((-t_{1-\alpha}(N-1)), \frac{\log(\gamma)-\log(\theta_U)}{\sigma^\star N^{-\frac{1}{2}}}; 0, \frac{(N-1)^{\frac{1}{2}}(\log(\theta_U)-\log(\theta_L))}{2\sigma^\star N^{-\frac{1}{2}}(t_{1-\alpha}(N-1))}\right) - Q_{N-1}\left((t_{1-\alpha}(N-1)), \frac{\log(\gamma)-\log(\theta_L)}{\sigma^\star N^{-\frac{1}{2}}}; 0, \frac{(N-1)^{\frac{1}{2}}(\log(\theta_U)-\log(\theta_L))}{2\sigma^\star N^{-\frac{1}{2}}(t_{1-\alpha}(N-1))}\right)$$

where

$$\sigma^\star = \left[\log(\text{CV}^2+1)\right]^{\frac{1}{2}}$$

is the standard deviation of the log-transformed data, and $Q_\cdot(\cdot,\cdot;\cdot,\cdot)$ is Owen's Q function, defined in the section "Common Notation" on page 4926.

Confidence Interval for Mean (CI=T)

This analysis of precision applies to the standard t-based confidence interval:

$$\begin{cases} \left[\bar{x} - t_{1-\frac{\alpha}{2}}(N-1)\frac{s}{\sqrt{N}}, \quad \bar{x} + t_{1-\frac{\alpha}{2}}(N-1)\frac{s}{\sqrt{N}}\right], & \text{two-sided} \\ \left[\bar{x} - t_{1-\alpha}(N-1)\frac{s}{\sqrt{N}}, \quad \infty\right), & \text{upper one-sided} \\ \left(-\infty, \quad \bar{x} + t_{1-\alpha}(N-1)\frac{s}{\sqrt{N}}\right], & \text{lower one-sided} \end{cases}$$

where $\bar{x}$ is the sample mean and s is the sample standard deviation. The "half-width" is defined as the distance from the point estimate $\bar{x}$ to a finite endpoint,

$$\text{half-width} = \begin{cases} t_{1-\frac{\alpha}{2}}(N-1)\frac{s}{\sqrt{N}}, & \text{two-sided} \\ t_{1-\alpha}(N-1)\frac{s}{\sqrt{N}}, & \text{one-sided} \end{cases}$$

A "valid" conference interval captures the true mean. The exact probability of obtaining at most the target confidence interval half-width h, unconditional or conditional on validity, is given by Beal (1989):

$$\Pr(\text{half-width} \le h) = \begin{cases} P\left(\chi^2(N-1) \le \frac{h^2 N(N-1)}{\sigma^2(t^2_{1-\frac{\alpha}{2}}(N-1))}\right), & \text{two-sided} \\ P\left(\chi^2(N-1) \le \frac{h^2 N(N-1)}{\sigma^2(t^2_{1-\alpha}(N-1))}\right), & \text{one-sided} \end{cases}$$

$$\Pr(\text{half-width} \le h \mid \text{validity}) = \begin{cases} \left(\frac{1}{1-\alpha}\right) 2\left[Q_{N-1}\left((t_{1-\frac{\alpha}{2}}(N-1)), 0; 0, b_1\right) - Q_{N-1}(0,0;0,b_1)\right], & \text{two-sided} \\ \left(\frac{1}{1-\alpha}\right) Q_{N-1}\left((t_{1-\alpha}(N-1)), 0; 0, b_1\right), & \text{one-sided} \end{cases}$$

where

$$b_1 = \frac{h(N-1)^{\frac{1}{2}}}{\sigma(t_{1-\frac{\alpha}{c}}(N-1))N^{-\frac{1}{2}}}$$

$$c = \text{number of sides}$$

and $Q.(\cdot,\cdot;\cdot,\cdot)$ is Owen's Q function, defined in the section "Common Notation" on page 4926.

A "quality" confidence interval is both sufficiently narrow (half-width $\leq h$) and valid:

$$\begin{aligned}\Pr(\text{quality}) &= \Pr(\text{half-width} \leq h \text{ and validity}) \\ &= \Pr(\text{half-width} \leq h \mid \text{validity})(1-\alpha)\end{aligned}$$

Analyses in the ONEWAYANOVA Statement

One-Degree-of-Freedom Contrast (TEST=CONTRAST)

The hypotheses are

$$H_0: c_1\mu_1 + \cdots + c_G\mu_G = c_0$$

$$H_1: \begin{cases} c_1\mu_1 + \cdots + c_G\mu_G \neq c_0, & \text{two-sided} \\ c_1\mu_1 + \cdots + c_G\mu_G > c_0, & \text{upper one-sided} \\ c_1\mu_1 + \cdots + c_G\mu_G < c_0, & \text{lower one-sided} \end{cases}$$

where G is the number of groups, $\{c_1, \ldots, c_G\}$ are the contrast coefficients, and c_0 is the null contrast value.

The test is the usual F test for a contrast in one-way ANOVA. It assumes normal data with common group variances and requires $N \geq G + 1$ and $n_i \geq 1$.

O'Brien and Muller (1993, Section 8.2.3.2) give the exact power as

$$\text{power} = \begin{cases} P\left(F(1, N-G, \delta^2) \geq F_{1-\alpha}(1, N-G)\right), & \text{two-sided} \\ P\left(t(N-G, \delta) \geq t_{1-\alpha}(N-G)\right), & \text{upper one-sided} \\ P\left(t(N-G, \delta) \leq t_{\alpha}(N-G)\right), & \text{lower one-sided} \end{cases}$$

where

$$\delta = N^{\frac{1}{2}} \left(\frac{\sum_{i=1}^{G} c_i \mu_i - c_0}{\sigma \left(\sum_{i=1}^{G} \frac{c_i^2}{w_i} \right)^{\frac{1}{2}}} \right)$$

Overall F Test (TEST=OVERALL)

The hypotheses are

$$H_0: \mu_1 = \mu_2 = \cdots = \mu_G$$
$$H_1: \mu_i \neq \mu_j \text{ for some } i, j$$

where G is the number of groups.

The test is the usual overall F test for equality of means in one-way ANOVA. It assumes normal data with common group variances and requires $N \geq G + 1$ and $n_i \geq 1$.

O'Brien and Muller (1993, Section 8.2.3.1) give the exact power as

$$\text{power} = P\left(F(G-1, N-G, \lambda) \geq F_{1-\alpha}(G-1, N-G)\right)$$

where the noncentrality is

$$\lambda = N\left(\frac{\sum_{i=1}^{G} w_i(\mu_i - \bar{\mu})^2}{\sigma^2}\right)$$

and

$$\bar{\mu} = \sum_{i=1}^{G} w_i \mu_i$$

Analyses in the PAIREDFREQ Statement

Overview of Conditional McNemar tests

Notation:

		Case		
		Failure	Success	
Control	Failure	n_{00}	n_{01}	$n_{0\cdot}$
	Success	n_{10}	n_{11}	$n_{1\cdot}$
		$n_{\cdot 0}$	$n_{\cdot 1}$	N

$$\begin{aligned}
n_{00} &= \#\{\text{control=failure, case=failure }\} \\
n_{01} &= \#\{\text{control=failure, case=success }\} \\
n_{10} &= \#\{\text{control=success, case=failure }\} \\
n_{11} &= \#\{\text{control=success, case=success }\} \\
N &= n_{00} + n_{01} + n_{10} + n_{11} \\
n_D &= n_{01} + n_{10} \equiv \ \#\text{ discordant pairs} \\
\hat{\pi}_{ij} &= \frac{n_{ij}}{N} \\
\pi_{ij} &= \text{theoretical population value of } \hat{\pi}_{ij} \\
\pi_{1\cdot} &= \pi_{10} + \pi_{11} \\
\pi_{\cdot 1} &= \pi_{01} + \pi_{11} \\
\phi &= \text{Corr(control observation, case observation)} \quad \text{(within a pair)} \\
\text{DPR} &= \text{“discordant proportion ratio”} = \frac{\pi_{01}}{\pi_{10}} \\
\text{DPR}_0 &= \text{null DPR}
\end{aligned}$$

Power formulas are given here in terms of the discordant proportions π_{10} and π_{01}. If the input is specified in terms of $\{\pi_{1\cdot}, \pi_{\cdot 1}, \phi\}$, then it can be converted into values for $\{\pi_{10}, \pi_{01}\}$ as follows:

$$\begin{aligned}
\pi_{01} &= \pi_{\cdot 1}(1 - \pi_{1\cdot}) - \phi((1 - \pi_{1\cdot})\pi_{1\cdot}(1 - \pi_{\cdot 1})\pi_{\cdot 1})^{\frac{1}{2}} \\
\pi_{10} &= \pi_{01} + \pi_{1\cdot} - \pi_{\cdot 1}
\end{aligned}$$

All McNemar tests covered in PROC POWER are *conditional*, meaning that n_D is assumed fixed at its observed value.

For the usual $\text{DPR}_0 = 1$, the hypotheses are

$$H_0{:}\pi_{\cdot 1} = \pi_{1\cdot}$$

$$H_1{:}\begin{cases} \pi_{\cdot 1} \neq \pi_{1\cdot}, & \text{two-sided} \\ \pi_{\cdot 1} > \pi_{1\cdot}, & \text{upper one-sided} \\ \pi_{\cdot 1} < \pi_{1\cdot}, & \text{lower one-sided} \end{cases}$$

The test statistic for both tests covered in PROC POWER (DIST=EXACT_COND and DIST=NORMAL) is the McNemar statistic Q_M, which has the following form when $\text{DPR}_0 = 1$:

$$Q_{M_0} = \frac{(n_{01} - n_{10})^2}{n_{01} + n_{10}}$$

For the conditional McNemar tests, this is equivalent to the square of the $Z(X)$ statistic for the test of a single proportion (normal approximation to binomial), where the proportion is $\frac{\pi_{01}}{\pi_{01}+\pi_{10}}$, the null is 0.5, and "$N$" is n_D (see, for example, Schork and Williams 1980):

$$\begin{aligned} Z(X) &= \frac{n_{01} - n_D(0.5)}{[n_D 0.5(1-0.5)]^{\frac{1}{2}}} \quad \dot{\sim} \text{N}\left(\frac{n_D^{\frac{1}{2}}(\frac{\pi_{01}}{\pi_{01}+\pi_{10}} - 0.5)}{[0.5(1-0.5)]^{\frac{1}{2}}}, \frac{\frac{\pi_{01}}{\pi_{01}+\pi_{10}}\left(1 - \frac{\pi_{01}}{\pi_{01}+\pi_{10}}\right)}{0.5(1-0.5)} \right) \\ &= \frac{n_{01} - (n_{01} + n_{10})(0.5)}{[(n_{01} + n_{10})0.5(1-0.5)]^{\frac{1}{2}}} \\ &= \frac{n_{01} - n_{10}}{[n_{01} + n_{10}]^{\frac{1}{2}}} \\ &= \sqrt{Q_{M_0}} \end{aligned}$$

This can be generalized to a custom null for $\frac{\pi_{01}}{\pi_{01}+\pi_{10}}$, which is equivalent to specifying a custom null DPR:

$$\left[\frac{\pi_{01}}{\pi_{01} + \pi_{10}}\right]_0 \equiv \left[\frac{1}{1 + \frac{1}{\frac{\pi_{01}}{\pi_{10}}}}\right]_0 \equiv \frac{1}{1 + \frac{1}{\text{DPR}_0}}$$

So, a conditional McNemar test (asymptotic or exact) with a custom null is equivalent to the test of a single proportion $p_1 \equiv \frac{\pi_{01}}{\pi_{01}+\pi_{10}}$ with a null value $p_0 \equiv \frac{1}{1+\frac{1}{\text{DPR}_0}}$, with a sample size of n_D:

$$H_0{:}p_1 = p_0$$

$$H_1{:}\begin{cases} p_1 \neq p_0, & \text{two-sided} \\ p_1 > p_0, & \text{one-sided U} \\ p_1 < p_0, & \text{one-sided L} \end{cases}$$

which is equivalent to

$$H_0\text{:DPR} = \text{DPR}_0$$

$$H_1\text{:} \begin{cases} \text{DPR} \neq \text{DPR}_0, & \text{two-sided} \\ \text{DPR} > \text{DPR}_0, & \text{one-sided U} \\ \text{DPR} < \text{DPR}_0, & \text{one-sided L} \end{cases}$$

The general form of the test statistic is thus

$$Q_M = \frac{(n_{01} - n_D p_0)^2}{n_D p_0 (1 - p_0)}$$

The two most common conditional McNemar tests assume either the exact conditional distribution of Q_M (covered by the DIST=EXACT_COND analysis) or a standard normal distribution for Q_M (covered by the DIST=NORMAL analysis).

McNemar Exact Conditional Test (TEST=MCNEMAR DIST=EXACT_COND)

For DIST=EXACT_COND, the power is calculated assuming that the test is conducted by using the exact conditional distribution of Q_M (conditional on n_D). The power is calculated by first computing the conditional power for each possible n_D. The unconditional power is computed as a weighted average over all possible outcomes of n_D:

$$\text{power} = \sum_{n_D=0}^{N} P(n_D) P(\text{Reject } p_1 = p_0 | n_D)$$

where $n_D \sim \text{Bin}(\pi_{01} + \pi_{10}, N)$, and $P(\text{Reject } p_1 = p_0 | n_D)$ is calculated by using the exact method in the section "Exact Test of a Binomial Proportion (TEST=EXACT)" on page 4934.

The achieved significance level, reported as "Actual Alpha" in the analysis, is computed in the same way except by using the actual alpha of the one-sample test in place of its power:

$$\text{actual alpha} = \sum_{n_D=0}^{N} P(n_D) \alpha^{\star}(p_1, p_0 | n_D)$$

where $\alpha^{\star}(p_1, p_0 | n_D)$ is the actual alpha calculated by using the exact method in the section "Exact Test of a Binomial Proportion (TEST=EXACT)" on page 4934 with proportion p_1, null p_0, and sample size n_D.

McNemar Normal Approximation Test (TEST=MCNEMAR DIST=NORMAL)

For DIST=NORMAL, power is calculated assuming the test is conducted by using the normal-approximate distribution of Q_M (conditional on n_D).

For the METHOD=EXACT option, the power is calculated in the same way as described in the section "McNemar Exact Conditional Test (TEST=MCNEMAR DIST=EXACT_COND)" on page 4960, except that $P(\text{Reject } p_1 = p_0 | n_D)$ is calculated by using the exact method in the section

"z Test for Binomial Proportion Using Null Variance (TEST=Z VAREST=NULL)" on page 4935. The achieved significance level is calculated in the same way as described at the end of the section "McNemar Exact Conditional Test (TEST=MCNEMAR DIST=EXACT_COND)" on page 4960.

For the METHOD=MIETTINEN option, approximate sample size for the one-sided cases is computed according to equation (5.6) in Miettinen (1968):

$$N = \frac{\left\{ z_{1-\alpha}(p_{10}+p_{01}) + z_{power}\left[(p_{10}+p_{01})^2 - \frac{1}{4}(p_{01}-p_{10})^2(3+p_{10}+p_{01})\right]^{\frac{1}{2}} \right\}^2}{(p_{10}+p_{01})(p_{01}-p_{10})^2}$$

Approximate power for the one-sided cases is computed by solving the sample size equation for power, and approximate power for the two-sided case follows easily by summing the one-sided powers each at $\alpha/2$:

$$\text{power} = \begin{cases} \Phi\left(\frac{(p_{01}-p_{10})[N(p_{10}+p_{01})]^{\frac{1}{2}} - z_{1-\alpha}(p_{10}+p_{01})}{\left[(p_{10}+p_{01})^2 - \frac{1}{4}(p_{01}-p_{10})^2(3+p_{10}+p_{01})\right]^{\frac{1}{2}}}\right), & \text{upper one-sided} \\ \Phi\left(\frac{-(p_{01}-p_{10})[N(p_{10}+p_{01})]^{\frac{1}{2}} - z_{1-\alpha}(p_{10}+p_{01})}{\left[(p_{10}+p_{01})^2 - \frac{1}{4}(p_{01}-p_{10})^2(3+p_{10}+p_{01})\right]^{\frac{1}{2}}}\right), & \text{lower one-sided} \\ \Phi\left(\frac{(p_{01}-p_{10})[N(p_{10}+p_{01})]^{\frac{1}{2}} - z_{1-\frac{\alpha}{2}}(p_{10}+p_{01})}{\left[(p_{10}+p_{01})^2 - \frac{1}{4}(p_{01}-p_{10})^2(3+p_{10}+p_{01})\right]^{\frac{1}{2}}}\right) + & \\ \quad \Phi\left(\frac{-(p_{01}-p_{10})[N(p_{10}+p_{01})]^{\frac{1}{2}} - z_{1-\frac{\alpha}{2}}(p_{10}+p_{01})}{\left[(p_{10}+p_{01})^2 - \frac{1}{4}(p_{01}-p_{10})^2(3+p_{10}+p_{01})\right]^{\frac{1}{2}}}\right), & \text{two-sided} \end{cases}$$

The two-sided solution for N is obtained by numerically inverting the power equation.

In general, compared to METHOD=CONNOR, the METHOD=MIETTINEN approximation tends to be slightly more accurate but can be slightly anticonservative in the sense of underestimating sample size and overestimating power (Lachin 1992, p. 1250).

For the METHOD=CONNOR option, approximate sample size for the one-sided cases is computed according to equation (3) in Connor (1987):

$$N = \frac{\left\{ z_{1-\alpha}(p_{10}+p_{01})^{\frac{1}{2}} + z_{power}\left[p_{10}+p_{01}-(p_{01}-p_{10})^2\right]^{\frac{1}{2}} \right\}^2}{(p_{01}-p_{10})^2}$$

Approximate power for the one-sided cases is computed by solving the sample size equation for power, and approximate power for the two-sided case follows easily by summing the one-sided powers each at $\alpha/2$:

$$\text{power} = \begin{cases} \Phi\left(\frac{(p_{01}-p_{10})N^{\frac{1}{2}} - z_{1-\alpha}(p_{10}+p_{01})^{\frac{1}{2}}}{\left[p_{10}+p_{01}-(p_{01}-p_{10})^2\right]^{\frac{1}{2}}}\right), & \text{upper one-sided} \\ \Phi\left(\frac{-(p_{01}-p_{10})N^{\frac{1}{2}} - z_{1-\alpha}(p_{10}+p_{01})^{\frac{1}{2}}}{\left[p_{10}+p_{01}-(p_{01}-p_{10})^2\right]^{\frac{1}{2}}}\right), & \text{lower one-sided} \\ \Phi\left(\frac{(p_{01}-p_{10})N^{\frac{1}{2}} - z_{1-\frac{\alpha}{2}}(p_{10}+p_{01})^{\frac{1}{2}}}{\left[p_{10}+p_{01}-(p_{01}-p_{10})^2\right]^{\frac{1}{2}}}\right) + & \\ \quad \Phi\left(\frac{-(p_{01}-p_{10})N^{\frac{1}{2}} - z_{1-\frac{\alpha}{2}}(p_{10}+p_{01})^{\frac{1}{2}}}{\left[p_{10}+p_{01}-(p_{01}-p_{10})^2\right]^{\frac{1}{2}}}\right), & \text{two-sided} \end{cases}$$

The two-sided solution for N is obtained by numerically inverting the power equation.

In general, compared to METHOD=MIETTINEN, the METHOD=CONNOR approximation tends to be slightly less accurate but slightly conservative in the sense of overestimating sample size and underestimating power (Lachin 1992, p. 1250).

Analyses in the PAIREDMEANS Statement

Paired t *Test (TEST=DIFF)*

The hypotheses for the paired t test are

$$H_0: \mu_{\text{diff}} = \mu_0$$

$$H_1: \begin{cases} \mu_{\text{diff}} \neq \mu_0, & \text{two-sided} \\ \mu_{\text{diff}} > \mu_0, & \text{upper one-sided} \\ \mu_{\text{diff}} < \mu_0, & \text{lower one-sided} \end{cases}$$

The test assumes normally distributed data and requires $N \geq 2$. The test statistics are

$$t = N^{\frac{1}{2}} \left(\frac{\bar{d} - \mu_0}{s_d} \right) \quad \sim t(N-1, \delta)$$

$$t^2 \sim F(1, N-1, \delta^2)$$

where $\bar{d}$ and s_d are the sample mean and standard deviation of the differences and

$$\delta = N^{\frac{1}{2}} \left(\frac{\mu_{\text{diff}} - \mu_0}{\sigma_{\text{diff}}} \right)$$

and

$$\sigma_{\text{diff}} = \left(\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2 \right)^{\frac{1}{2}}$$

The test is

$$\text{Reject} \quad H_0 \quad \text{if} \begin{cases} t^2 \geq F_{1-\alpha}(1, N-1), & \text{two-sided} \\ t \geq t_{1-\alpha}(N-1), & \text{upper one-sided} \\ t \leq t_{\alpha}(N-1), & \text{lower one-sided} \end{cases}$$

Exact power computations for t tests are given in O'Brien and Muller (1993, Section 8.2.2):

$$\text{power} = \begin{cases} P\left(F(1, N-1, \delta^2) \geq F_{1-\alpha}(1, N-1) \right), & \text{two-sided} \\ P\left(t(N-1, \delta) \geq t_{1-\alpha}(N-1) \right), & \text{upper one-sided} \\ P\left(t(N-1, \delta) \leq t_{\alpha}(N-1) \right), & \text{lower one-sided} \end{cases}$$

Paired t Test for Mean Ratio with Lognormal Data (TEST=RATIO)

The lognormal case is handled by reexpressing the analysis equivalently as a normality-based test on the log-transformed data, by using properties of the lognormal distribution as discussed in Johnson, Kotz, and Balakrishnan (1994, Chapter 14). The approaches in the section "Paired t Test (TEST=DIFF)" on page 4962 then apply.

In contrast to the usual t test on normal data, the hypotheses with lognormal data are defined in terms of geometric means rather than arithmetic means.

The hypotheses for the paired t test with lognormal pairs $\{Y_1, Y_2\}$ are

$$H_0: \frac{\gamma_2}{\gamma_1} = \gamma_0$$

$$H_1: \begin{cases} \frac{\gamma_2}{\gamma_1} \neq \gamma_0, & \text{two-sided} \\ \frac{\gamma_2}{\gamma_1} > \gamma_0, & \text{upper one-sided} \\ \frac{\gamma_2}{\gamma_1} < \gamma_0, & \text{lower one-sided} \end{cases}$$

Let $\mu_1^\star$, $\mu_2^\star$, $\sigma_1^\star$, $\sigma_2^\star$, and $\rho^\star$ be the (arithmetic) means, standard deviations, and correlation of the bivariate normal distribution of the log-transformed data $\{\log Y_1, \log Y_2\}$. The hypotheses can be rewritten as follows:

$$H_0: \mu_2^\star - \mu_1^\star = \log(\gamma_0)$$

$$H_1: \begin{cases} \mu_2^\star - \mu_1^\star \neq \log(\gamma_0), & \text{two-sided} \\ \mu_2^\star - \mu_1^\star > \log(\gamma_0), & \text{upper one-sided} \\ \mu_2^\star - \mu_1^\star < \log(\gamma_0), & \text{lower one-sided} \end{cases}$$

where

$$\mu_1^\star = \log \gamma_1$$
$$\mu_2^\star = \log \gamma_2$$
$$\sigma_1^\star = \left[\log(\mathrm{CV}_1^2 + 1)\right]^{\frac{1}{2}}$$
$$\sigma_2^\star = \left[\log(\mathrm{CV}_2^2 + 1)\right]^{\frac{1}{2}}$$
$$\rho^\star = \frac{\log\{\rho \mathrm{CV}_1 \mathrm{CV}_2 + 1\}}{\sigma_1^\star \sigma_2^\star}$$

and CV_1, CV_2, and ρ are the coefficients of variation and the correlation of the original untransformed pairs $\{Y_1, Y_2\}$. The conversion from ρ to $\rho^\star$ is shown in Jones and Miller (1966).

The test assumes lognormally distributed data and requires $N \geq 2$. The power is

$$\text{power} = \begin{cases} P\left(F(1, N-1, \delta^2) \geq F_{1-\alpha}(1, N-1)\right), & \text{two-sided} \\ P\left(t(N-1, \delta) \geq t_{1-\alpha}(N-1)\right), & \text{upper one-sided} \\ P\left(t(N-1, \delta) \leq t_{\alpha}(N-1)\right), & \text{lower one-sided} \end{cases}$$

where

$$\delta = N^{\frac{1}{2}} \left(\frac{\mu_1^\star - \mu_2^\star - \log(\gamma_0)}{\sigma^\star} \right)$$

and

$$\sigma^\star = \left(\sigma_1^{\star 2} + \sigma_2^{\star 2} - 2\rho^\star \sigma_1^\star \sigma_2^\star \right)^{\frac{1}{2}}$$

Additive Equivalence Test for Mean Difference with Normal Data (TEST=EQUIV_DIFF)

The hypotheses for the equivalence test are

$$H_0: \mu_{\text{diff}} < \theta_L \quad \text{or} \quad \mu_{\text{diff}} > \theta_U$$
$$H_1: \theta_L \leq \mu_{\text{diff}} \leq \theta_U$$

The analysis is the two one-sided tests (TOST) procedure of Schuirmann (1987). The test assumes normally distributed data and requires $N \geq 2$. Phillips (1990) derives an expression for the exact power assuming a two-sample balanced design; the results are easily adapted to a paired design:

$$\text{power} = Q_{N-1}\left((-t_{1-\alpha}(N-1)), \frac{\mu_{\text{diff}} - \theta_U}{\sigma_{\text{diff}} N^{-\frac{1}{2}}}; 0, \frac{(N-1)^{\frac{1}{2}}(\theta_U - \theta_L)}{2\sigma_{\text{diff}} N^{-\frac{1}{2}} (t_{1-\alpha}(N-1))} \right) -$$
$$Q_{N-1}\left((t_{1-\alpha}(N-1)), \frac{\mu_{\text{diff}} - \theta_L}{\sigma_{\text{diff}} N^{-\frac{1}{2}}}; 0, \frac{(N-1)^{\frac{1}{2}}(\theta_U - \theta_L)}{2\sigma_{\text{diff}} N^{-\frac{1}{2}} (t_{1-\alpha}(N-1))} \right)$$

where

$$\sigma_{\text{diff}} = \left(\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2 \right)^{\frac{1}{2}}$$

and $Q_{\cdot}(\cdot, \cdot; \cdot, \cdot)$ is Owen's Q function, defined in the section "Common Notation" on page 4926.

Multiplicative Equivalence Test for Mean Ratio with Lognormal Data (TEST=EQUIV_RATIO)

The lognormal case is handled by reexpressing the analysis equivalently as a normality-based test on the log-transformed data, by using properties of the lognormal distribution as discussed in Johnson, Kotz, and Balakrishnan (1994, Chapter 14). The approaches in the section "Additive Equivalence Test for Mean Difference with Normal Data (TEST=EQUIV_DIFF)" on page 4964 then apply.

In contrast to the additive equivalence test on normal data, the hypotheses with lognormal data are defined in terms of geometric means rather than arithmetic means.

The hypotheses for the equivalence test are

$$H_0: \frac{\gamma_T}{\gamma_R} \leq \theta_L \quad \text{or} \quad \frac{\gamma_T}{\gamma_R} \geq \theta_U$$
$$H_1: \theta_L < \frac{\gamma_T}{\gamma_R} < \theta_U$$

where $0 < \theta_L < \theta_U$

The analysis is the two one-sided tests (TOST) procedure of Schuirmann (1987) on the log-transformed data. The test assumes lognormally distributed data and requires $N \geq 2$. Diletti, Hauschke, and Steinijans (1991) derive an expression for the exact power assuming a crossover design; the results are easily adapted to a paired design:

$$\text{power} = Q_{N-1}\left((-t_{1-\alpha}(N-1)), \frac{\log\left(\frac{\gamma_T}{\gamma_R}\right) - \log(\theta_U)}{\sigma^\star N^{-\frac{1}{2}}}; 0, \frac{(N-1)^{\frac{1}{2}}(\log(\theta_U) - \log(\theta_L))}{2\sigma^\star N^{-\frac{1}{2}}(t_{1-\alpha}(N-1))}\right) -$$
$$Q_{N-1}\left((t_{1-\alpha}(N-1)), \frac{\log\left(\frac{\gamma_T}{\gamma_R}\right) - \log(\theta_L)}{\sigma^\star N^{-\frac{1}{2}}}; 0, \frac{(N-1)^{\frac{1}{2}}(\log(\theta_U) - \log(\theta_L))}{2\sigma^\star N^{-\frac{1}{2}}(t_{1-\alpha}(N-1))}\right)$$

where $\sigma^\star$ is the standard deviation of the differences between the log-transformed pairs (in other words, the standard deviation of $\log(Y_T) - \log(Y_R)$, where Y_T and Y_R are observations from the treatment and reference, respectively), computed as

$$\sigma^\star = \left(\sigma_R^{\star 2} + \sigma_T^{\star 2} - 2\rho^\star\sigma_R^\star\sigma_T^\star\right)^{\frac{1}{2}}$$
$$\sigma_R^\star = \left[\log(\text{CV}_R^2 + 1)\right]^{\frac{1}{2}}$$
$$\sigma_T^\star = \left[\log(\text{CV}_T^2 + 1)\right]^{\frac{1}{2}}$$
$$\rho^\star = \frac{\log\{\rho\text{CV}_R\text{CV}_T + 1\}}{\sigma_R^\star\sigma_T^\star}$$

where CV_R, CV_T, and ρ are the coefficients of variation and the correlation of the original untransformed pairs $\{Y_T, Y_R\}$, and $Q.(\cdot,\cdot;\cdot,\cdot)$ is Owen's Q function. The conversion from ρ to $\rho^\star$ is shown in Jones and Miller (1966), and Owen's Q function is defined in the section "Common Notation" on page 4926.

Confidence Interval for Mean Difference (CI=DIFF)

This analysis of precision applies to the standard t-based confidence interval:

$$\begin{cases} \left[\bar{d} - t_{1-\frac{\alpha}{2}}(N-1)\frac{s_d}{\sqrt{N}}, \quad \bar{d} + t_{1-\frac{\alpha}{2}}(N-1)\frac{s_d}{\sqrt{N}}\right], & \text{two-sided} \\ \left[\bar{d} - t_{1-\alpha}(N-1)\frac{s_d}{\sqrt{N}}, \quad \infty\right), & \text{upper one-sided} \\ \left(-\infty, \quad \bar{d} + t_{1-\alpha}(N-1)\frac{s_d}{\sqrt{N}}\right], & \text{lower one-sided} \end{cases}$$

where $\bar{d}$ and s_d are the sample mean and standard deviation of the differences. The "half-width" is defined as the distance from the point estimate $\bar{d}$ to a finite endpoint,

$$\text{half-width} = \begin{cases} t_{1-\frac{\alpha}{2}}(N-1)\frac{s_d}{\sqrt{N}}, & \text{two-sided} \\ t_{1-\alpha}(N-1)\frac{s_d}{\sqrt{N}}, & \text{one-sided} \end{cases}$$

A "valid" conference interval captures the true mean difference. The exact probability of obtaining at most the target confidence interval half-width h, unconditional or conditional on validity, is given

by Beal (1989):

$$\Pr(\text{half-width} \le h) = \begin{cases} P\left(\chi^2(N-1) \le \frac{h^2 N(N-1)}{\sigma^2_{\text{diff}}(t^2_{1-\frac{\alpha}{2}}(N-1))}\right), & \text{two-sided} \\ P\left(\chi^2(N-1) \le \frac{h^2 N(N-1)}{\sigma^2_{\text{diff}}(t^2_{1-\alpha}(N-1))}\right), & \text{one-sided} \end{cases}$$

$$\Pr(\text{half-width} \le h \mid \text{validity}) = \begin{cases} \left(\frac{1}{1-\alpha}\right) 2\left[Q_{N-1}\left((t_{1-\frac{\alpha}{2}}(N-1)), 0; \right.\right. \\ \quad \left.\left. 0, b_1\right) - Q_{N-1}(0,0;0,b_1)\right], & \text{two-sided} \\ \left(\frac{1}{1-\alpha}\right) Q_{N-1}\left((t_{1-\alpha}(N-1)), 0; 0, b_1\right), & \text{one-sided} \end{cases}$$

where

$$\sigma_{\text{diff}} = \left(\sigma_1^2 + \sigma_2^2 - 2\rho\sigma_1\sigma_2\right)^{\frac{1}{2}}$$

$$b_1 = \frac{h(N-1)^{\frac{1}{2}}}{\sigma_{\text{diff}}(t_{1-\frac{\alpha}{c}}(N-1))N^{-\frac{1}{2}}}$$

$$c = \text{number of sides}$$

and $Q_{\cdot}(\cdot,\cdot;\cdot,\cdot)$ is Owen's Q function, defined in the section "Common Notation" on page 4926.

A "quality" confidence interval is both sufficiently narrow (half-width $\le h$) and valid:

$$\begin{aligned} \Pr(\text{quality}) &= \Pr(\text{half-width} \le h \text{ and validity}) \\ &= \Pr(\text{half-width} \le h \mid \text{validity})(1-\alpha) \end{aligned}$$

Analyses in the TWOSAMPLEFREQ Statement

Overview of the 2×2 *Table*

Notation:

		Outcome		
		Failure	Success	
Group	1	$n_1 - x_1$	x_1	n_1
	2	$n_2 - x_2$	x_2	n_2
		$N - m$	m	N

$$x_1 = \#\text{ successes in group 1}$$

$$x_2 = \#\text{ successes in group 2}$$

$$m = x_1 + x_2 = \text{ total \# successes}$$

$$\hat{p}_1 = \frac{x_1}{n_1}$$

$$\hat{p}_2 = \frac{x_2}{n_2}$$

$$\hat{p} = \frac{m}{N} = w_1\hat{p}_1 + w_2\hat{p}_2$$

The hypotheses are

$$H_0: p_2 - p_1 = p_0$$

$$H_1: \begin{cases} p_2 - p_1 \neq p_0, & \text{two-sided} \\ p_2 - p_1 > p_0, & \text{upper one-sided} \\ p_2 - p_1 < p_0, & \text{lower one-sided} \end{cases}$$

where p_0 is constrained to be 0 for all but the unconditional Pearson chi-square test.

Internal calculations are performed in terms of p_1, p_2, and p_0. An input set consisting of OR, p_1, and OR_0 is transformed as follows:

$$p_2 = \frac{(OR)p_1}{1 - p_1 + (OR)p_1}$$
$$p_{10} = p_1$$
$$p_{20} = \frac{OR_0 p_{10}}{1 - p_{10} + (OR_0)p_{10}}$$
$$p_0 = p_{20} - p_{10}$$

An input set consisting of RR, p_1, and RR_0 is transformed as follows:

$$p_2 = (RR)p_1$$
$$p_{10} = p_1$$
$$p_{20} = (RR_0)p_{10}$$
$$p_0 = p_{20} - p_{10}$$

Note that the transformation of either OR_0 or RR_0 to p_0 is not unique. The chosen parameterization fixes the null value p_{10} at the input value of p_1.

Pearson Chi-Square Test for Two Proportions (TEST=PCHI)

The usual Pearson chi-square test is unconditional. The test statistic

$$z_P = \frac{\hat{p}_2 - \hat{p}_1 - p_0}{\left[\hat{p}(1-\hat{p})\left(\frac{1}{n_1} + \frac{1}{n_2}\right)\right]^{\frac{1}{2}}} = [N w_1 w_2]^{\frac{1}{2}} \frac{\hat{p}_2 - \hat{p}_1 - p_0}{\hat{p}(1-\hat{p})}$$

is assumed to have a null distribution of $N(0, 1)$.

Sample size for the one-sided cases is given by equation (4) in Fleiss, Tytun, and Ury (1980). One-sided power is computed as suggested by Diegert and Diegert (1981) by inverting the sample size formula. Power for the two-sided case is computed by adding the lower-sided and upper-sided powers each with $\alpha/2$, and sample size for the two-sided case is obtained by numerically inverting

the power formula. A custom null value p_0 for the proportion difference $p_2 - p_1$ is also supported.

$$\text{power} = \begin{cases} \Phi\left(\frac{(p_2-p_1-p_0)(Nw_1w_2)^{\frac{1}{2}} - z_{1-\alpha}[(w_1p_1+w_2p_2)(1-w_1p_1-w_2p_2)]^{\frac{1}{2}}}{[w_2p_1(1-p_1)+w_1p_2(1-p_2)]^{\frac{1}{2}}}\right), & \text{upper one-sided} \\ \Phi\left(\frac{-(p_2-p_1-p_0)(Nw_1w_2)^{\frac{1}{2}} - z_{1-\alpha}[(w_1p_1+w_2p_2)(1-w_1p_1-w_2p_2)]^{\frac{1}{2}}}{[w_2p_1(1-p_1)+w_1p_2(1-p_2)]^{\frac{1}{2}}}\right), & \text{lower one-sided} \\ \Phi\left(\frac{(p_2-p_1-p_0)(Nw_1w_2)^{\frac{1}{2}} - z_{1-\frac{\alpha}{2}}[(w_1p_1+w_2p_2)(1-w_1p_1-w_2p_2)]^{\frac{1}{2}}}{[w_2p_1(1-p_1)+w_1p_2(1-p_2)]^{\frac{1}{2}}}\right) + & \\ \quad \Phi\left(\frac{-(p_2-p_1-p_0)(Nw_1w_2)^{\frac{1}{2}} - z_{1-\frac{\alpha}{2}}[(w_1p_1+w_2p_2)(1-w_1p_1-w_2p_2)]^{\frac{1}{2}}}{[w_2p_1(1-p_1)+w_1p_2(1-p_2)]^{\frac{1}{2}}}\right), & \text{two-sided} \end{cases}$$

For the one-sided cases, a closed-form inversion of the power equation yield an approximate total sample size

$$N = \frac{\left[z_{1-\alpha}\{(w_1p_1 + w_2p_2)(1 - w_1p_1 - w_2p_2)\}^{\frac{1}{2}} + z_{\text{power}}\{w_2p_1(1-p_1) + w_1p_2(1-p_2)\}^{\frac{1}{2}}\right]^2}{w_1w_2(p_2 - p_1 - p_0)^2}$$

For the two-sided case, the solution for N is obtained by numerically inverting the power equation.

Likelihood Ratio Chi-Square Test for Two Proportions (TEST=LRCHI)

The usual likelihood ratio chi-square test is unconditional. The test statistic

$$z_{LR} = (-1_{\{p_2<p_1\}})\sqrt{2N\sum_{i=1}^{2}\left[w_i\hat{p}_i\log\left(\frac{\hat{p}_i}{\hat{p}}\right) + w_i(1-\hat{p}_i)\log\left(\frac{1-\hat{p}_i}{1-\hat{p}}\right)\right]}$$

is assumed to have a null distribution of $N(0,1)$ and an alternative distribution of $N(\delta,1)$, where

$$\delta = N^{\frac{1}{2}}(-1_{\{p_2<p_1\}})\sqrt{2\sum_{i=1}^{2}\left[w_ip_i\log\left(\frac{p_i}{w_1p_1+w_2p_2}\right) + w_i(1-p_i)\log\left(\frac{1-p_i}{1-(w_1p_1+w_2p_2)}\right)\right]}$$

The approximate power is

$$\text{power} = \begin{cases} \Phi(\delta - z_{1-\alpha}), & \text{upper one-sided} \\ \Phi(-\delta - z_{1-\alpha}), & \text{lower one-sided} \\ \Phi\left(\delta - z_{1-\frac{\alpha}{2}}\right) + \Phi\left(-\delta - z_{1-\frac{\alpha}{2}}\right), & \text{two-sided} \end{cases}$$

For the one-sided cases, a closed-form inversion of the power equation yield an approximate total sample size

$$N = \left(\frac{z_{\text{power}} + z_{1-\alpha}}{\delta}\right)^2$$

For the two-sided case, the solution for N is obtained by numerically inverting the power equation.

Fisher's Exact Conditional Test for Two Proportions (Test=FISHER)

Fisher's exact test is conditional on the observed total number of successes m. Power and sample size computations for the METHOD=WALTERS option are based on a test with similar power properties, the continuity-adjusted arcsine test. The test statistic

$$z_A = (4Nw_1w_2)^{\frac{1}{2}}\left[\arcsin\left(\left[\hat{p}_2 + \frac{1}{2Nw_2}(1_{\{\hat{p}_2<\hat{p}_1\}} - 1_{\{\hat{p}_2>\hat{p}_1\}})\right]^{\frac{1}{2}}\right) - \arcsin\left(\left[\hat{p}_1 + \frac{1}{2Nw_1}(1_{\{\hat{p}_1<\hat{p}_2\}} - 1_{\{\hat{p}_1>\hat{p}_2\}})\right]^{\frac{1}{2}}\right)\right]$$

is assumed to have a null distribution of $N(0,1)$ and an alternative distribution of $N(\delta,1)$, where

$$\delta = (4Nw_1w_2)^{\frac{1}{2}}\left[\arcsin\left(\left[p_2 + \frac{1}{2Nw_2}(1_{\{p_2<p_1\}} - 1_{\{p_2>p_1\}})\right]^{\frac{1}{2}}\right) - \arcsin\left(\left[p_1 + \frac{1}{2Nw_1}(1_{\{p_1<p_2\}} - 1_{\{p_1>p_2\}})\right]^{\frac{1}{2}}\right)\right]$$

The approximate power for the one-sided balanced case is given by Walters (1979) and is easily extended to the unbalanced and two-sided cases:

$$\text{power} = \begin{cases} \Phi(\delta - z_{1-\alpha}), & \text{upper one-sided} \\ \Phi(-\delta - z_{1-\alpha}), & \text{lower one-sided} \\ \Phi\left(\delta - z_{1-\frac{\alpha}{2}}\right) + \Phi\left(-\delta - z_{1-\frac{\alpha}{2}}\right), & \text{two-sided} \end{cases}$$

Analyses in the TWOSAMPLEMEANS Statement

Two-Sample t Test Assuming Equal Variances (TEST=DIFF)

The hypotheses for the two-sample t test are

$$H_0: \mu_{\text{diff}} = \mu_0$$

$$H_1: \begin{cases} \mu_{\text{diff}} \neq \mu_0, & \text{two-sided} \\ \mu_{\text{diff}} > \mu_0, & \text{upper one-sided} \\ \mu_{\text{diff}} < \mu_0, & \text{lower one-sided} \end{cases}$$

The test assumes normally distributed data and common standard deviation per group, and it requires $N \geq 3$, $n_1 \geq 1$, and $n_2 \geq 1$. The test statistics are

$$t = N^{\frac{1}{2}}(w_1w_2)^{\frac{1}{2}}\left(\frac{\bar{x}_2 - \bar{x}_1 - \mu_0}{s_p}\right) \sim t(N-2,\delta)$$

$$t^2 \sim F(1, N-2, \delta^2)$$

where $\bar{x}_1$ and $\bar{x}_2$ are the sample means and s_p is the pooled standard deviation, and

$$\delta = N^{\frac{1}{2}}(w_1w_2)^{\frac{1}{2}}\left(\frac{\mu_{\text{diff}} - \mu_0}{\sigma}\right)$$

The test is

$$\text{Reject} \quad H_0 \quad \text{if} \begin{cases} t^2 \geq F_{1-\alpha}(1, N-2), & \text{two-sided} \\ t \geq t_{1-\alpha}(N-2), & \text{upper one-sided} \\ t \leq t_{\alpha}(N-2), & \text{lower one-sided} \end{cases}$$

Exact power computations for t tests are given in O'Brien and Muller (1993, Section 8.2.1):

$$\text{power} = \begin{cases} P\left(F(1, N-2, \delta^2) \geq F_{1-\alpha}(1, N-2)\right), & \text{two-sided} \\ P\left(t(N-2, \delta) \geq t_{1-\alpha}(N-2)\right), & \text{upper one-sided} \\ P\left(t(N-2, \delta) \leq t_{\alpha}(N-2)\right), & \text{lower one-sided} \end{cases}$$

Solutions for N, n_1, n_2, α, and δ are obtained by numerically inverting the power equation. Closed-form solutions for other parameters, in terms of δ, are as follows:

$$\mu_{\text{diff}} = \delta\sigma(Nw_1w_2)^{-\frac{1}{2}} + \mu_0$$

$$\mu_1 = \delta\sigma(Nw_1w_2)^{-\frac{1}{2}} + \mu_0 - \mu_2$$

$$\mu_2 = \delta\sigma(Nw_1w_2)^{-\frac{1}{2}} + \mu_0 - \mu_1$$

$$\sigma = \begin{cases} \delta^{-1}(Nw_1w_2)^{\frac{1}{2}}(\mu_{\text{diff}} - \mu_0), & |\delta| > 0 \\ \text{undefined}, & \text{otherwise} \end{cases}$$

$$w_1 = \begin{cases} \frac{1}{2} \pm \frac{1}{2}\left[1 - \frac{4\delta^2\sigma^2}{N(\mu_{\text{diff}}-\mu_0)^2}\right]^{\frac{1}{2}}, & 0 < |\delta| \leq \frac{1}{2}N^{\frac{1}{2}}\frac{|\mu_{\text{diff}}-\mu_0|}{\sigma} \\ \text{undefined}, & \text{otherwise} \end{cases}$$

$$w_2 = \begin{cases} \frac{1}{2} \pm \frac{1}{2}\left[1 - \frac{4\delta^2\sigma^2}{N(\mu_{\text{diff}}-\mu_0)^2}\right]^{\frac{1}{2}}, & 0 < |\delta| \leq \frac{1}{2}N^{\frac{1}{2}}\frac{|\mu_{\text{diff}}-\mu_0|}{\sigma} \\ \text{undefined}, & \text{otherwise} \end{cases}$$

Finally, here is a derivation of the solution for w_1:

Solve the δ equation for w_1 (which requires the quadratic formula). Then determine the range of δ given w_1:

$$\min_{w_1}(\delta) = \begin{cases} 0, & \text{when} \quad w_1 = 0 \quad \text{or} \quad 1, \quad \text{if} \quad (\mu_{\text{diff}} - \mu_0) \geq 0 \\ \frac{1}{2}N^{\frac{1}{2}}\frac{(\mu_{\text{diff}}-\mu_0)}{\sigma}, & \text{when} \quad w_1 = \frac{1}{2}, \quad \text{if} \quad (\mu_{\text{diff}} - \mu_0) < 0 \end{cases}$$

$$\max_{w_1}(\delta) = \begin{cases} 0, & \text{when} \quad w_1 = 0 \quad \text{or} \quad 1, \quad \text{if} \quad (\mu_{\text{diff}} - \mu_0) < 0 \\ \frac{1}{2}N^{\frac{1}{2}}\frac{(\mu_{\text{diff}}-\mu_0)}{\sigma}, & \text{when} \quad w_1 = \frac{1}{2}, \quad \text{if} \quad (\mu_{\text{diff}} - \mu_0) \geq 0 \end{cases}$$

This implies

$$|\delta| \leq \frac{1}{2}N^{\frac{1}{2}}\frac{|\mu_{\text{diff}} - \mu_0|}{\sigma}$$

Two-Sample Satterthwaite t Test Assuming Unequal Variances (TEST=DIFF_SATT)

The hypotheses for the two-sample Satterthwaite t test are

$$H_0: \mu_{\text{diff}} = \mu_0$$

$$H_1: \begin{cases} \mu_{\text{diff}} \neq \mu_0, & \text{two-sided} \\ \mu_{\text{diff}} > \mu_0, & \text{upper one-sided} \\ \mu_{\text{diff}} < \mu_0, & \text{lower one-sided} \end{cases}$$

The test assumes normally distributed data and requires $N \geq 3$, $n_1 \geq 1$, and $n_2 \geq 1$. The test statistics are

$$t = \frac{\bar{x}_2 - \bar{x}_1 - \mu_0}{\left[\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right]^{\frac{1}{2}}} = N^{\frac{1}{2}} \frac{\bar{x}_2 - \bar{x}_1 - \mu_0}{\left[\frac{s_1^2}{w_1} + \frac{s_2^2}{w_2}\right]^{\frac{1}{2}}}$$

$$F = t^2$$

where $\bar{x}_1$ and $\bar{x}_2$ are the sample means and s_1 and s_2 are the sample standard deviations.

As DiSantostefano and Muller (1995, p. 585) state, the test is based on assuming that under H_0, F is distributed as $F(1, \nu)$, where ν is given by Satterthwaite's approximation (Satterthwaite 1946),

$$\nu = \frac{\left[\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}\right]^2}{\frac{\left[\frac{\sigma_1^2}{n_1}\right]^2}{n_1 - 1} + \frac{\left[\frac{\sigma_2^2}{n_2}\right]^2}{n_2 - 1}} = \frac{\left[\frac{\sigma_1^2}{w_1} + \frac{\sigma_2^2}{w_2}\right]^2}{\frac{\left[\frac{\sigma_1^2}{w_1}\right]^2}{Nw_1 - 1} + \frac{\left[\frac{\sigma_2^2}{w_2}\right]^2}{Nw_2 - 1}}$$

Since ν is unknown, in practice it must be replaced by an estimate

$$\hat{\nu} = \frac{\left[\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}\right]^2}{\frac{\left[\frac{s_1^2}{n_1}\right]^2}{n_1 - 1} + \frac{\left[\frac{s_2^2}{n_2}\right]^2}{n_2 - 1}} = \frac{\left[\frac{s_1^2}{w_1} + \frac{s_2^2}{w_2}\right]^2}{\frac{\left[\frac{s_1^2}{w_1}\right]^2}{Nw_1 - 1} + \frac{\left[\frac{s_2^2}{w_2}\right]^2}{Nw_2 - 1}}$$

So the test is

$$\text{Reject} \quad H_0 \quad \text{if} \begin{cases} F \geq F_{1-\alpha}(1, \hat{\nu}), & \text{two-sided} \\ t \geq t_{1-\alpha}(\hat{\nu}), & \text{upper one-sided} \\ t \leq t_{\alpha}(\hat{\nu}), & \text{lower one-sided} \end{cases}$$

Exact solutions for power for the two-sided and upper one-sided cases are given in Moser, Stevens, and Watts (1989). The lower one-sided case follows easily by using symmetry. The equations are

as follows:

$$\text{power} = \begin{cases} \int_0^\infty P\left(F(1, N-2, \lambda) > h(u)F_{1-\alpha}(1, v(u))|u\right) f(u)\mathrm{d}u, & \text{two-sided} \\ \int_0^\infty P\left(t(N-2, \lambda^{\frac{1}{2}}) > [h(u)]^{\frac{1}{2}} t_{1-\alpha}(v(u))|u\right) f(u)\mathrm{d}u, & \text{upper one-sided} \\ \int_0^\infty P\left(t(N-2, \lambda^{\frac{1}{2}}) < [h(u)]^{\frac{1}{2}} t_{\alpha}(v(u))|u\right) f(u)\mathrm{d}u, & \text{lower one-sided} \end{cases}$$

where

$$h(u) = \frac{\left(\frac{1}{n_1} + \frac{u}{n_2}\right)(n_1 + n_2 - 2)}{\left[(n_1 - 1) + (n_2 - 1)\frac{u\sigma_1^2}{\sigma_2^2}\right]\left(\frac{1}{n_1} + \frac{\sigma_2^2}{\sigma_1^2 n_2}\right)}$$

$$v(u) = \frac{\left(\frac{1}{n_1} + \frac{u}{n_2}\right)^2}{\frac{1}{n_1^2(n_1-1)} + \frac{u^2}{n_2^2(n_2-1)}}$$

$$\lambda = \frac{(\mu_{\text{diff}} - \mu_0)^2}{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}$$

$$f(u) = \frac{\Gamma\left(\frac{n_1+n_2-2}{2}\right)}{\Gamma\left(\frac{n_1-1}{2}\right)\Gamma\left(\frac{n_2-1}{2}\right)} \left[\frac{\sigma_1^2(n_2-1)}{\sigma_2^2(n_1-1)}\right]^{\frac{n_2-1}{2}} u^{\frac{n_2-3}{2}} \left[1 + \left(\frac{n_2-1}{n_1-1}\right)\frac{u\sigma_1^2}{\sigma_2^2}\right]^{-\left(\frac{n_1+n_2-2}{2}\right)}$$

The density $f(u)$ is obtained from the fact that

$$\frac{u\sigma_1^2}{\sigma_2^2} \sim F(n_2 - 1, n_1 - 1)$$

Two-Sample Pooled t Test of Mean Ratio with Lognormal Data (TEST=RATIO)

The lognormal case is handled by reexpressing the analysis equivalently as a normality-based test on the log-transformed data, by using properties of the lognormal distribution as discussed in Johnson, Kotz, and Balakrishnan (1994, Chapter 14). The approaches in the section "Two-Sample *t* Test Assuming Equal Variances (TEST=DIFF)" on page 4969 then apply.

In contrast to the usual t test on normal data, the hypotheses with lognormal data are defined in terms of geometric means rather than arithmetic means. The test assumes equal coefficients of variation in the two groups.

The hypotheses for the two-sample t test with lognormal data are

$$H_0: \frac{\gamma_2}{\gamma_1} = \gamma_0$$

$$H_1: \begin{cases} \frac{\gamma_2}{\gamma_1} \neq \gamma_0, & \text{two-sided} \\ \frac{\gamma_2}{\gamma_1} > \gamma_0, & \text{upper one-sided} \\ \frac{\gamma_2}{\gamma_1} < \gamma_0, & \text{lower one-sided} \end{cases}$$

Let $\mu_1^\star$, $\mu_2^\star$, and $\sigma^\star$ be the (arithmetic) means and common standard deviation of the corresponding normal distributions of the log-transformed data. The hypotheses can be rewritten as follows:

$$H_0{:}\mu_2^\star - \mu_1^\star = \log(\gamma_0)$$
$$H_1{:}\begin{cases} \mu_2^\star - \mu_1^\star \neq \log(\gamma_0), & \text{two-sided} \\ \mu_2^\star - \mu_1^\star > \log(\gamma_0), & \text{upper one-sided} \\ \mu_2^\star - \mu_1^\star < \log(\gamma_0), & \text{lower one-sided} \end{cases}$$

where

$$\mu_1^\star = \log \gamma_1$$
$$\mu_2^\star = \log \gamma_2$$

The test assumes lognormally distributed data and requires $N \geq 3$, $n_1 \geq 1$, and $n_2 \geq 1$.

The power is

$$\text{power} = \begin{cases} P\left(F(1, N-2, \delta^2) \geq F_{1-\alpha}(1, N-2)\right), & \text{two-sided} \\ P\left(t(N-2, \delta) \geq t_{1-\alpha}(N-2)\right), & \text{upper one-sided} \\ P\left(t(N-2, \delta) \leq t_{\alpha}(N-2)\right), & \text{lower one-sided} \end{cases}$$

where

$$\delta = N^{\frac{1}{2}}(w_1 w_2)^{\frac{1}{2}}\left(\frac{\mu_2^\star - \mu_1^\star - \log(\gamma_0)}{\sigma^\star}\right)$$
$$\sigma^\star = \left[\log(\text{CV}^2 + 1)\right]^{\frac{1}{2}}$$

Additive Equivalence Test for Mean Difference with Normal Data (TEST=EQUIV_DIFF)

The hypotheses for the equivalence test are

$$H_0{:}\mu_{\text{diff}} < \theta_L \quad \text{or} \quad \mu_{\text{diff}} > \theta_U$$
$$H_1{:}\theta_L \leq \mu_{\text{diff}} \leq \theta_U$$

The analysis is the two one-sided tests (TOST) procedure of Schuirmann (1987). The test assumes normally distributed data and requires $N \geq 3$, $n_1 \geq 1$, and $n_2 \geq 1$. Phillips (1990) derives an expression for the exact power assuming a balanced design; the results are easily adapted to an unbalanced design:

$$\text{power} = Q_{N-2}\left((-t_{1-\alpha}(N-2)), \frac{\mu_{\text{diff}} - \theta_U}{\sigma N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}}; 0, \frac{(N-2)^{\frac{1}{2}}(\theta_U - \theta_L)}{2\sigma N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}(t_{1-\alpha}(N-2))}\right) -$$
$$Q_{N-2}\left((t_{1-\alpha}(N-2)), \frac{\mu_{\text{diff}} - \theta_L}{\sigma N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}}; 0, \frac{(N-2)^{\frac{1}{2}}(\theta_U - \theta_L)}{2\sigma N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}(t_{1-\alpha}(N-2))}\right)$$

where $Q_\cdot(\cdot, \cdot; \cdot, \cdot)$ is Owen's Q function, defined in the section "Common Notation" on page 4926.

Multiplicative Equivalence Test for Mean Ratio with Lognormal Data (TEST=EQUIV_RATIO)

The lognormal case is handled by reexpressing the analysis equivalently as a normality-based test on the log-transformed data, by using properties of the lognormal distribution as discussed in Johnson, Kotz, and Balakrishnan (1994, Chapter 14). The approaches in the section "Additive Equivalence Test for Mean Difference with Normal Data (TEST=EQUIV_DIFF)" on page 4973 then apply.

In contrast to the additive equivalence test on normal data, the hypotheses with lognormal data are defined in terms of geometric means rather than arithmetic means.

The hypotheses for the equivalence test are

$$H_0: \frac{\gamma_T}{\gamma_R} \leq \theta_L \quad \text{or} \quad \frac{\gamma_T}{\gamma_R} \geq \theta_U$$
$$H_1: \theta_L < \frac{\gamma_T}{\gamma_R} < \theta_U$$
$$\text{where} \quad 0 < \theta_L < \theta_U$$

The analysis is the two one-sided tests (TOST) procedure of Schuirmann (1987) on the log-transformed data. The test assumes lognormally distributed data and requires $N \geq 3$, $n_1 \geq 1$, and $n_2 \geq 1$. Diletti, Hauschke, and Steinijans (1991) derive an expression for the exact power assuming a crossover design; the results are easily adapted to an unbalanced two-sample design:

$$\text{power} = Q_{N-2}\left((-t_{1-\alpha}(N-2)), \frac{\log\left(\frac{\gamma_T}{\gamma_R}\right) - \log(\theta_U)}{\sigma^\star N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}}; 0, \frac{(N-2)^{\frac{1}{2}}(\log(\theta_U) - \log(\theta_L))}{2\sigma^\star N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}(t_{1-\alpha}(N-2))}\right) -$$
$$Q_{N-2}\left((t_{1-\alpha}(N-2)), \frac{\log\left(\frac{\gamma_T}{\gamma_R}\right) - \log(\theta_L)}{\sigma^\star N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}}; 0, \frac{(N-2)^{\frac{1}{2}}(\log(\theta_U) - \log(\theta_L))}{2\sigma^\star N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}(t_{1-\alpha}(N-2))}\right)$$

where

$$\sigma^\star = \left[\log(\text{CV}^2 + 1)\right]^{\frac{1}{2}}$$

is the (assumed common) standard deviation of the normal distribution of the log-transformed data, and $Q_\cdot(\cdot, \cdot; \cdot, \cdot)$ is Owen's Q function, defined in the section "Common Notation" on page 4926.

Confidence Interval for Mean Difference (CI=DIFF)

This analysis of precision applies to the standard t-based confidence interval:

$$\begin{aligned}
&\left[(\bar{x}_2 - \bar{x}_1) - t_{1-\frac{\alpha}{2}}(N-2)\frac{s_p}{\sqrt{N w_1 w_2}},\right. \\
&\quad \left.(\bar{x}_2 - \bar{x}_1) + t_{1-\frac{\alpha}{2}}(N-2)\frac{s_p}{\sqrt{N w_1 w_2}}\right], && \text{two-sided} \\
&\left[(\bar{x}_2 - \bar{x}_1) - t_{1-\alpha}(N-2)\frac{s_p}{\sqrt{N w_1 w_2}}, \quad \infty\right), && \text{upper one-sided} \\
&\left(-\infty, \quad (\bar{x}_2 - \bar{x}_1) + t_{1-\alpha}(N-2)\frac{s_p}{\sqrt{N w_1 w_2}}\right], && \text{lower one-sided}
\end{aligned}$$

where $\bar{x}_1$ and $\bar{x}_2$ are the sample means and s_p is the pooled standard deviation. The "half-width" is defined as the distance from the point estimate $\bar{x}_2 - \bar{x}_1$ to a finite endpoint,

$$\text{half-width} = \begin{cases} t_{1-\frac{\alpha}{2}}(N-2)\frac{s_p}{\sqrt{N w_1 w_2}}, & \text{two-sided} \\ t_{1-\alpha}(N-2)\frac{s_p}{\sqrt{N w_1 w_2}}, & \text{one-sided} \end{cases}$$

A "valid" conference interval captures the true mean. The exact probability of obtaining at most the target confidence interval half-width h, unconditional or conditional on validity, is given by Beal (1989):

$$\Pr(\text{half-width} \le h) = \begin{cases} P\left(\chi^2(N-2) \le \frac{h^2 N(N-2)(w_1 w_2)}{\sigma^2 (t^2_{1-\frac{\alpha}{2}}(N-2))}\right), & \text{two-sided} \\ P\left(\chi^2(N-2) \le \frac{h^2 N(N-2)(w_1 w_2)}{\sigma^2 (t^2_{1-\alpha}(N-2))}\right), & \text{one-sided} \end{cases}$$

$$\Pr(\text{half-width} \le h \mid \text{validity}) = \begin{cases} \left(\frac{1}{1-\alpha}\right) 2\left[Q_{N-2}\left((t_{1-\frac{\alpha}{2}}(N-2)), 0; \right.\right. \\ \quad \left.\left. 0, b_2\right) - Q_{N-2}(0,0;0,b_2)\right], & \text{two-sided} \\ \left(\frac{1}{1-\alpha}\right) Q_{N-2}\left((t_{1-\alpha}(N-2)), 0; 0, b_2\right), & \text{one-sided} \end{cases}$$

where

$$b_2 = \frac{h(N-2)^{\frac{1}{2}}}{\sigma(t_{1-\frac{\alpha}{c}}(N-2))N^{-\frac{1}{2}}(w_1 w_2)^{-\frac{1}{2}}}$$

$$c = \text{number of sides}$$

and $Q_\cdot(\cdot,\cdot;\cdot,\cdot)$ is Owen's Q function, defined in the section "Common Notation" on page 4926.

A "quality" confidence interval is both sufficiently narrow (half-width $\le h$) and valid:

$$\begin{aligned} \Pr(\text{quality}) &= \Pr(\text{half-width} \le h \text{ and validity}) \\ &= \Pr(\text{half-width} \le h \mid \text{validity})(1-\alpha) \end{aligned}$$

Analyses in the TWOSAMPLESURVIVAL Statement

Rank Tests for Two Survival Curves (TEST=LOGRANK, TEST=GEHAN, TEST=TARONEWARE)

The method is from Lakatos (1988) and Cantor (1997, pp. 83–92).

Define the following notation:

$$
\begin{aligned}
X_j(i) &= i\text{th input time point on survival curve for group } j \\
S_j(i) &= \text{input survivor function value corresponding to } X_j(i) \\
h_j(t) &= \text{hazard rate for group } j \text{ at time } t \\
\Psi_j(t) &= \text{loss hazard rate for group } j \text{ at time } t \\
\lambda_j &= \text{exponential hazard rate for group } j \\
R &= \text{hazard ratio of group 2 to group 1} \equiv \text{(assumed constant) value of } \frac{h_2(t)}{h_1(t)} \\
m_j &= \text{median survival time for group } j \\
b &= \text{number of subintervals per time unit} \\
T &= \text{accrual time} \\
\tau &= \text{postaccrual follow-up time} \\
L_j &= \text{exponential loss rate for group } j \\
XL_j &= \text{input time point on loss curve for group } j \\
SL_j &= \text{input survivor function value corresponding to } XL_j \\
mL_j &= \text{median survival time for group } j \\
r_i &= \text{rank for } i\text{th time point}
\end{aligned}
$$

Each survival curve can be specified in one of several ways.

- For exponential curves:
  - a single point $(X_j(1), S_j(1))$ on the curve
  - median survival time
  - hazard rate
  - hazard ratio (for curve 2, with respect to curve 1)
- For piecewise linear curves with proportional hazards:
  - a set of points $\{(X_1(1), S_1(1)), (X_1(2), S_1(2)), \ldots\}$ (for curve 1)
  - hazard ratio (for curve 2, with respect to curve 1)
- For arbitrary piecewise linear curves:
  - a set of points $\{(X_j(1), S_j(1)), (X_j(2), S_j(2)), \ldots\}$

A total of M evenly spaced time points $\{t_0 = 0, t_1, t_2, \ldots, t_M = T + \tau\}$ are used in calculations, where

$$M = \text{floor}\left((T + \tau)b\right)$$

The hazard function is calculated for each survival curve at each time point. For an exponential curve, the (constant) hazard is given by one of the following, depending on the input parameteriza-

tion:

$$h_j(t) = \begin{cases} \lambda_j \\ \lambda_1 R \\ \frac{-\log(\frac{1}{2})}{m_j} \\ \frac{-\log(S_j(1))}{X_j(1)} \\ \frac{-\log(S_1(1))}{X_1(1)} R \end{cases}$$

For a piecewise linear curve, define the following additional notation:

t_i^- = largest input time X such that $X \le t_i$

t_i^+ = smallest input time X such that $X > t_i$

The hazard is computed by using linear interpolation as follows:

$$h_j(t_i) = \frac{S_j(t_i^-) - S_j(t_i^+)}{\left[S_j(t_i^+) - S_j(t_i^-)\right]\left[t_i - t_i^-\right] + S_j(t_i^-)\left[t_i^+ - t_i^-\right]}$$

With proportional hazards, the hazard rate of group 2's curve in terms of the hazard rate of group 1's curve is

$$h_2(t) = h_1(t)R$$

Hazard function values $\{\Psi_j(t_i)\}$ for the loss curves are computed in an analogous way from $\{L_j, XL_j, SL_j, mL_j\}$.

The expected number at risk $N_j(i)$ at time i in group j is calculated for each group and time points 0 through $M-1$, as follows:

$$N_j(0) = Nw_j$$

$$N_j(i+1) = N_j(i)\left[1 - h_j(t_i)\left(\frac{1}{b}\right) - \Psi_j(t_i)\left(\frac{1}{b}\right) - \left(\frac{1}{b(T+\tau-t_i)}\right)1_{\{t_i>\tau\}}\right]$$

Define θ_i as the ratio of hazards and ϕ_i as the ratio of expected numbers at risk for time t_i:

$$\theta_i = \frac{h_2(t_i)}{h_1(t_i)}$$

$$\phi_i = \frac{N_2(i)}{N_1(i)}$$

The expected number of deaths in each subinterval is calculated as follows:

$$D_i = [h_1(t_i)N_1(i) + h_2(t_i)N_2(i)]\left(\frac{1}{b}\right)$$

The rank values are calculated as follows according to which test statistic is used:

$$r_i = \begin{cases} 1, & \text{log-rank} \\ N_1(i) + N_2(i), & \text{Gehan} \\ \sqrt{N_1(i) + N_2(i)}, & \text{Tarone-Ware} \end{cases}$$

The distribution of the test statistic is approximated by $N(E,1)$ where

$$E = \frac{\sum_{i=0}^{M-1} D_i r_i \left[\frac{\phi_i \theta_i}{1+\phi_i \theta_i} - \frac{\phi_i}{1+\phi_i}\right]}{\sqrt{\sum_{i=0}^{M-1} D_i r_i^2 \frac{\phi_i}{(1+\phi_i)^2}}}$$

Note that $N^{\frac{1}{2}}$ can be factored out of the mean E, and so it can be expressed equivalently as

$$E = N^{\frac{1}{2}} E^\star = N^{\frac{1}{2}} \left[\frac{\sum_{i=0}^{M-1} D_i^\star r_i^\star \left[\frac{\phi_i \theta_i}{1+\phi_i \theta_i} - \frac{\phi_i}{1+\phi_i}\right]}{\sqrt{\sum_{i=0}^{M-1} D_i^\star r_i^{\star 2} \frac{\phi_i}{(1+\phi_i)^2}}}\right]$$

where $E^\star$ is free of N and

$$D_i^\star = \left[h_1(t_i) N_1^\star(i) + h_2(t_i) N_2^\star(i)\right]\left(\frac{1}{b}\right)$$

$$r_i^\star = \begin{cases} 1, & \text{log-rank} \\ N_1^\star(i) + N_2^\star(i), & \text{Gehan} \\ \sqrt{N_1^\star(i) + N_2^\star(i)}, & \text{Tarone-Ware} \end{cases}$$

$$N_j^\star(0) = w_j$$

$$N_j^\star(i+1) = N_j^\star(i)\left[1 - h_j(t_i)\left(\frac{1}{b}\right) - \Psi_j(t_i)\left(\frac{1}{b}\right) - \left(\frac{1}{b(T+\tau-t_i)}\right) 1_{\{t_i > \tau\}}\right]$$

The approximate power is

$$\text{power} = \begin{cases} \Phi\left(-N^{\frac{1}{2}} E^\star - z_{1-\alpha}\right), & \text{upper one-sided} \\ \Phi\left(N^{\frac{1}{2}} E^\star - z_{1-\alpha}\right), & \text{lower one-sided} \\ \Phi\left(-N^{\frac{1}{2}} E^\star - z_{1-\frac{\alpha}{2}}\right) + \Phi\left(N^{\frac{1}{2}} E^\star - z_{1-\frac{\alpha}{2}}\right), & \text{two-sided} \end{cases}$$

Note that the upper and lower one-sided cases are expressed differently than in other analyses. This is because $E^\star > 0$ corresponds to a higher survival curve in group 1 and thus, by the convention used in PROC power for two-group analyses, the lower side.

For the one-sided cases, a closed-form inversion of the power equation yield an approximate total sample size

$$N = \left(\frac{z_{\text{power}} + z_{1-\alpha}}{E^\star}\right)^2$$

For the two-sided case, the solution for N is obtained by numerically inverting the power equation.

Analyses in the TWOSAMPLEWILCOXON Statement

Wilcoxon-Mann-Whitney Test for Comparing Two Distributions (TEST=WMW)

The power approximation in this section is applicable to the Wilcoxon-Mann-Whitney (WMW) test as invoked with the WILCOXON option in the PROC NPAR1WAY statement of the NPAR1WAY

procedure. The approximation is based on O'Brien and Castelloe (2006) and an estimator called $\widehat{WMW_{\text{odds}}}$. See O'Brien and Castelloe (2006) for a definition of $\widehat{WMW_{\text{odds}}}$, which need not be derived in detail here for purposes of explaining the power formula.

Let Y_1 and Y_2 be independent observations from any two distributions that you want to compare using the WMW test. For purposes of deriving the asymptotic distribution of $\widehat{WMW_{\text{odds}}}$ (and consequently the power computation as well), these distributions must be formulated as ordered categorical ("ordinal") distributions.

If a distribution is continuous, it can be discretized using a large number of categories with negligible loss of accuracy. Each nonordinal distribution is divided into b categories, where b is the value of the NBINS parameter, with breakpoints evenly spaced on the probability scale. That is, each bin contains an equal probability $1/b$ for that distribution. Then the breakpoints across both distributions are pooled to form a collection of C bins (heretofore called "categories"), and the probabilities of bin membership for each distribution are recalculated. The motivation for this method of binning is to avoid degenerate representations of the distributions—that is, small handfuls of large probabilities among mostly empty bins—as can be caused by something like an evenly spaced grid across raw values rather than probabilities.

After the discretization process just mentioned, there are now two ordinal distributions, each with a set of probabilities across a common set of C ordered categories. For simplicity of notation, assume (without loss of generality) the response values to be $1, \ldots, C$. Represent the conditional probabilities as

$$\tilde{p}_{ij} = \text{Prob}\left(Y_i = j \mid \text{group} = i\right), i \in \{1, 2\} \quad \text{and} \quad j \in \{1, \ldots, C\}$$

and the group allocation weights as

$$w_i = \frac{n_i}{N} = \text{Prob}\left(\text{group} = i\right), \quad i \in \{1, 2\}$$

The joint probabilities can then be calculated simply as

$$p_{ij} = \text{Prob}\left(\text{group} = i, Y_i = j\right) = w_i \tilde{p}_{ij}, i \in \{1, 2\} \quad \text{and} \quad j \in \{1, \ldots, C\}$$

The next step in the power computation is to compute the probabilities that a randomly chosen pair of observations from the two groups is concordant, discordant, or tied. It is useful to define these probabilities as functions of the terms Rs_{ij} and Rd_{ij}, defined as follows, where Y is a random observation drawn from the joint distribution across groups and categories:

$$\begin{aligned} Rs_{ij} &= \text{Prob}\left(Y \text{ is concordant with cell}(i, j)\right) + \frac{1}{2}\text{Prob}\left(Y \text{ is tied with cell}(i, j)\right) \\ &= \text{Prob}\left((\text{group} < i \text{ and } Y < j) \text{ or } (\text{group} > i \text{ and } Y > j)\right) + \\ &\quad \frac{1}{2}\text{Prob}\left(\text{group} \neq i \text{ and } Y = j\right) \\ &= \sum_{g=1}^{2} \sum_{c=1}^{C} w_g \tilde{p}_{gc} \left[\text{I}_{(g-i)(c-j)>0} + \frac{1}{2}\text{I}_{g \neq i, c=j} \right] \end{aligned}$$

and

$$\begin{aligned} Rd_{ij} &= \text{Prob}\,(Y \text{ is discordant with cell}(i,j)) + \frac{1}{2}\text{Prob}\,(Y \text{ is tied with cell}(i,j)) \\ &= \text{Prob}\,((\text{group} < i \text{ and } Y > j) \text{ or } (\text{group} > i \text{ and } Y < j)) + \\ &\quad \frac{1}{2}\text{Prob}\,(\text{group} \neq i \text{ and } Y = j) \\ &= \sum_{g=1}^{2}\sum_{c=1}^{C} w_g \tilde{p}_{gc} \left[\mathrm{I}_{(g-i)(c-j)<0} + \frac{1}{2}\mathrm{I}_{g\neq i, c=j} \right] \end{aligned}$$

For an independent random draw Y_1, Y_2 from the two distributions,

$$\begin{aligned} P_c &= \text{Prob}\,(Y_1, Y_2 \text{ concordant}) + \frac{1}{2}\text{Prob}\,(Y_1, Y_2 \text{ tied}) \\ &= \sum_{i=1}^{2}\sum_{j=1}^{C} w_i \tilde{p}_{ij} Rs_{ij} \end{aligned}$$

and

$$\begin{aligned} P_d &= \text{Prob}\,(Y_1, Y_2 \text{ discordant}) + \frac{1}{2}\text{Prob}\,(Y_1, Y_2 \text{ tied}) \\ &= \sum_{i=1}^{2}\sum_{j=1}^{C} w_i \tilde{p}_{ij} Rd_{ij} \end{aligned}$$

Then

$$WMW_{\text{odds}} = \frac{P_c}{P_d}$$

Proceeding to compute the theoretical standard error associated with WMW_{odds} (that is, the population analogue to the sample standard error),

$$\text{SE}(WMW_{\text{odds}}) = \frac{2}{P_d}\left[\sum_{i=1}^{2}\sum_{j=1}^{C} w_i \tilde{p}_{ij} \left(WMW_{\text{odds}} Rd_{ij} - Rs_{ij} \right)^2 / N \right]^{\frac{1}{2}}$$

Converting to the natural log scale and using the delta method,

$$\text{SE}(\log(WMW_{\text{odds}})) = \frac{\text{SE}(WMW_{\text{odds}})}{WMW_{\text{odds}}}$$

The next step is to produce a "smoothed" version of the $2 \times C$ cell probabilities that conforms to the null hypothesis of the Wilcoxon-Mann-Whitney test (in other words, independence in the $2 \times C$ contingency table of probabilities). Let $\text{SE}_{H_0}(\log(WMW_{\text{odds}}))$ denote the theoretical standard error of $\log(WMW_{\text{odds}})$ assuming H_0.

Finally we have all of the terms needed to compute the power, using the noncentral chi-square and normal distributions:

$$\text{power} = \begin{cases} P\left(Z \geq \frac{\text{SE}_{H_0}(\log(WMW_{\text{odds}}))}{\text{SE}(\log(WMW_{\text{odds}}))} z_{1-\alpha} - \delta^\star N^{\frac{1}{2}}\right), & \text{upper one-sided} \\ P\left(Z \leq \frac{\text{SE}_{H_0}(\log(WMW_{\text{odds}}))}{\text{SE}(\log(WMW_{\text{odds}}))} z_{\alpha} - \delta^\star N^{\frac{1}{2}}\right), & \text{lower one-sided} \\ P\left(\chi^2(1, (\delta^\star)^2 N) \geq \left[\frac{\text{SE}_{H_0}(\log(WMW_{\text{odds}}))}{\text{SE}(\log(WMW_{\text{odds}}))}\right]^2 \chi^2_{1-\alpha}(1)\right), & \text{two-sided} \end{cases}$$

where

$$\delta^\star = \frac{\log(WMW_{\text{odds}})}{N^{\frac{1}{2}}\text{SE}(\log(WMW_{\text{odds}}))}$$

is the primary noncentrality—that is, the "effect size" that quantifies how much the two conjectured distributions differ. Z is a standard normal random variable, $\chi^2(df, nc)$ is a noncentral χ^2 random variable with degrees of freedom df and noncentrality nc, and N is the total sample size.

Examples: POWER Procedure

Example 67.1: One-Way ANOVA

This example deals with the same situation as in Example 41.1 of Chapter 41, "The GLMPOWER Procedure."

Hocking (1985, p. 109) describes a study of the effectiveness of electrolytes in reducing lactic acid buildup for long-distance runners. You are planning a similar study in which you will allocate five different fluids to runners on a 10-mile course and measure lactic acid buildup immediately after the run. The fluids consist of water and two commercial electrolyte drinks, EZDure and LactoZap, each prepared at two concentrations, low (EZD1 and LZ1) and high (EZD2 and LZ2).

You conjecture that the standard deviation of lactic acid measurements given any particular fluid is about 3.75, and that the expected lactic acid values will correspond roughly to those in Table 67.31. You are least familiar with the LZ1 drink and hence decide to consider a range of reasonable values for that mean.

Table 67.31 Mean Lactic Acid Buildup by Fluid

Water	EZD1	EZD2	LZ1	LZ2
35.6	33.7	30.2	29 or 28	25.9

You are interested in four different comparisons, shown in Table 67.32 with appropriate contrast coefficients.

Table 67.32 Planned Comparisons

Comparison	Contrast Coefficients				
	Water	EZD1	EZD2	LZ1	LZ2
Water versus electrolytes	4	–1	–1	–1	–1
EZD versus LZ	0	1	1	–1	–1
EZD1 versus EZD2	0	1	–1	0	0
LZ1 versus LZ2	0	0	0	1	–1

For each of these contrasts you want to determine the sample size required to achieve a power of 0.9 for detecting an effect with magnitude in accord with Table 67.31. You are not yet attempting to choose a single sample size for the study, but rather checking the range of sample sizes needed for individual contrasts. You plan to test each contrast at $\alpha = 0.025$. In the interests of reducing costs, you will provide twice as many runners with water as with any of the electrolytes; in other words, you will use a sample size weighting scheme of 2:1:1:1:1. Use the ONEWAYANOVA statement in the POWER procedure to compute the sample sizes.

The statements required to perform this analysis are as follows:

```
proc power;
   onewayanova
      groupmeans = 35.6 | 33.7 | 30.2 | 29 28 | 25.9
      stddev = 3.75
      groupweights = (2 1 1 1 1)
      alpha = 0.025
      ntotal = .
      power = 0.9
      contrast = (4 -1 -1 -1 -1) (0  1  1 -1 -1)
                 (0  1 -1  0  0) (0  0  0  1 -1);
run;
```

The NTOTAL= option with a missing value (.) indicates total sample size as the result parameter. The GROUPMEANS= option with values from Table 67.31 specifies your conjectures for the means. With only one mean varying (the LZ1 mean), the "crossed" notation is simpler, showing scenarios for each group mean, separated by vertical bars (|). See the section "Specifying Value Lists in Analysis Statements" on page 4919 for more details on crossed and matched notations for grouped values. The contrasts in Table 67.32 are specified with the CONTRAST= option, by using the "matched" notation with each contrast enclosed in parentheses. The STDDEV=, ALPHA=, and POWER= options specify the error standard deviation, significance level, and power. The GROUPWEIGHTS= option specifies the weighting schemes. Default values for the NULLCONTRAST= and SIDES= options specify a two-sided t test of the contrast equal to 0. See Output 67.1.1 for the results.

Output 67.1.1 Sample Sizes for One-Way ANOVA Contrasts

The POWER Procedure
Single DF Contrast in One-Way ANOVA

Fixed Scenario Elements

Method	Exact
Alpha	0.025
Standard Deviation	3.75
Group Weights	2 1 1 1 1
Nominal Power	0.9
Number of Sides	2
Null Contrast Value	0

Computed N Total

Index	Contrast					Means					Actual Power	N Total
1	4	-1	-1	-1	-1	35.6	33.7	30.2	29	25.9	0.947	30
2	4	-1	-1	-1	-1	35.6	33.7	30.2	28	25.9	0.901	24
3	0	1	1	-1	-1	35.6	33.7	30.2	29	25.9	0.929	60
4	0	1	1	-1	-1	35.6	33.7	30.2	28	25.9	0.922	48
5	0	1	-1	0	0	35.6	33.7	30.2	29	25.9	0.901	174
6	0	1	-1	0	0	35.6	33.7	30.2	28	25.9	0.901	174
7	0	0	0	1	-1	35.6	33.7	30.2	29	25.9	0.902	222
8	0	0	0	1	-1	35.6	33.7	30.2	28	25.9	0.902	480

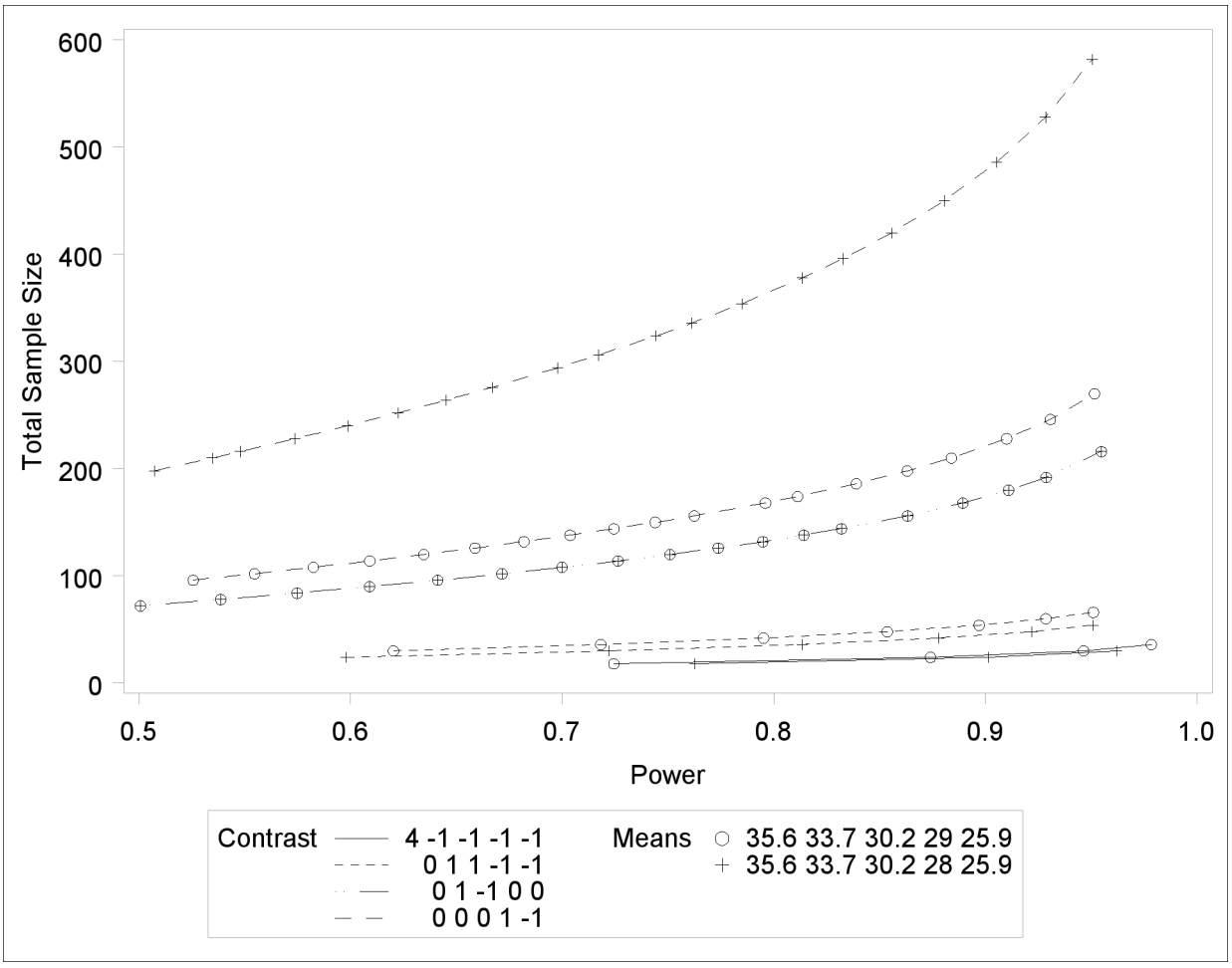
The sample sizes in Output 67.1.1 range from 24 for the comparison of water versus electrolytes to 480 for the comparison of LZ1 versus LZ2, both assuming the smaller LZ1 mean. The sample size for the latter comparison is relatively large because the small mean difference of $28 - 25.9 = 2.1$ is hard to detect.

The Nominal Power of 0.9 in the "Fixed Scenario Elements" table in Output 67.1.1 represents the input target power, and the Actual Power column in the "Computed N Total" table is the power at the sample size (N Total) adjusted to achieve the specified sample weighting. Note that all of the sample sizes are rounded up to multiples of 6 to preserve integer group sizes (since the group weights add up to 6). You can use the NFRACTIONAL option in the ONEWAYANOVA statement to compute raw fractional sample sizes.

Suppose you want to plot the required sample size for the range of power values from 0.5 to 0.95. First, define the analysis by specifying the same statements as before, but add the PLOTONLY option to the PROC POWER statement to disable the nongraphical results. Next, specify the PLOT statement with X=POWER to request a plot with power on the X axis. (The result parameter, here sample size, is always plotted on the other axis.) Use the MIN= and MAX= options in the PLOT statement to specify the power range. The following statements produce the plot shown in Output 67.1.2.

```
proc power plotonly;
   onewayanova
      groupmeans = 35.6 | 33.7 | 30.2 | 29 28 | 25.9
      stddev = 3.75
      groupweights = (2 1 1 1 1)
      alpha = 0.025
      ntotal = .
      power = 0.9
      contrast = (4 -1 -1 -1 -1) (0  1  1 -1 -1)
                 (0  1 -1  0  0) (0  0  0  1 -1);
   plot x=power min=.5 max=.95;
run;
```

Output 67.1.2 Plot of Sample Size versus Power for One-Way ANOVA Contrasts



In Output 67.1.2, the line style identifies the contrast, and the plotting symbol identifies the group means scenario. The plot shows that the required sample size is highest for the (0 0 0 1 –1) contrast, corresponding to the test of LZ1 versus LZ2 that was previously found to require the most resources, in either cell means scenario.

Note that some of the plotted points in Output 67.1.2 are unevenly spaced. This is because the plotted points are the *rounded* sample size results at their corresponding *actual* power levels. The

range specified with the MIN= and MAX= values in the PLOT statement corresponds to *nominal* power levels. In some cases, actual power is substantially higher than nominal power. To obtain plots with evenly spaced points (but with *fractional* sample sizes at the computed points), you can use the NFRACTIONAL option in the analysis statement preceding the PLOT statement.

Finally, suppose you want to plot the power for the range of sample sizes you will likely consider for the study (the range of 24 to 480 that achieves 0.9 power for different comparisons). In the ONEWAYANOVA statement, identify power as the result (POWER=.), and specify NTOTAL=24. The following statements produce the plot:

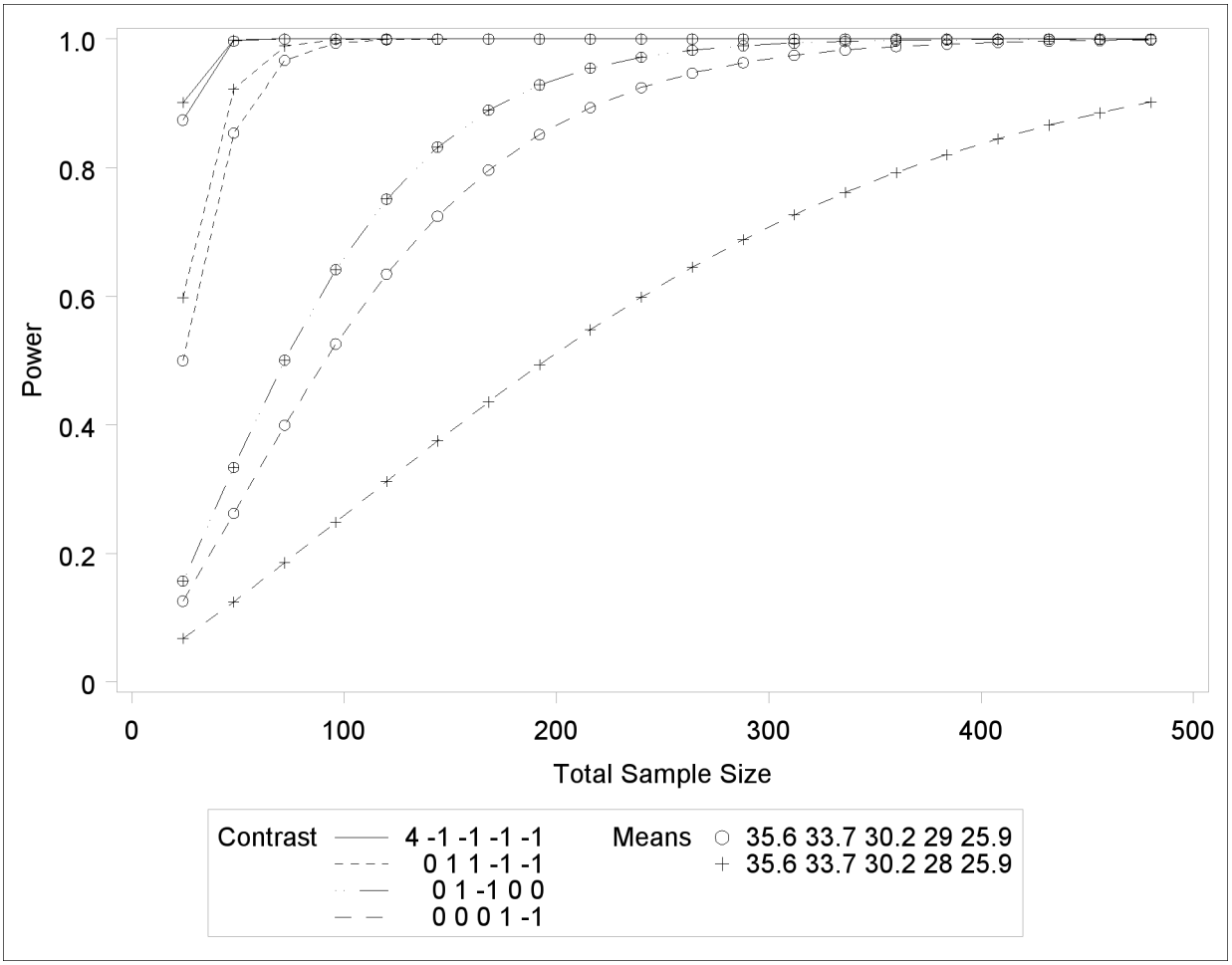
```
proc power plotonly;
   onewayanova
      groupmeans = 35.6 | 33.7 | 30.2 | 29 28 | 25.9
      stddev = 3.75
      groupweights = (2 1 1 1 1)
      alpha = 0.025
      ntotal = 24
      power = .
      contrast = (4 -1 -1 -1 -1) (0  1  1 -1 -1)
                 (0  1 -1  0  0) (0  0  0  1 -1);
   plot x=n min=24 max=480;
run;
```

The X=N option in the PLOT statement requests a plot with sample size on the X axis.

Note that the value specified with the NTOTAL=24 option is not used. It is overridden in the plot by the MIN= and MAX= options in the PLOT statement, and the PLOTONLY option in the PROC POWER statement disables nongraphical results. But the NTOTAL= option (along with a value) is still needed in the ONEWAYANOVA statement as a placeholder, to identify the desired parameterization for sample size.

Output 67.1.3 shows the resulting plot.

Output 67.1.3 Plot of Power versus Sample Size for One-Way ANOVA Contrasts



Although Output 67.1.2 and Output 67.1.3 surface essentially the same computations for practical power ranges, they each provide a different quick visual assessment. Output 67.1.2 reveals the range of required sample sizes for powers of interest, and Output 67.1.3 reveals the range of achieved powers for sample sizes of interest.

Example 67.2: The Sawtooth Power Function in Proportion Analyses

For many common statistical analyses, the power curve is monotonically increasing: the more samples you take, the more power you achieve. However, in statistical analyses of discrete data, such as tests of proportions, the power curve is often nonmonotonic. A small increase in sample size can result in a *decrease* in power, a decrease that is sometimes substantial. The explanation is that the actual significance level (in other words, the achieved Type I error rate) for discrete tests strays below the target level and varies with sample size. The power loss from a decrease in the Type I error rate can outweigh the power gain from an increase in sample size. The example discussed here demonstrates this "sawtooth" phenomenon. For additional discussion on the topic, see Chernick and Liu (2002).

Suppose you have a new scheduling system for an airline, and you want to determine how many flights you must observe to have at least an 80% chance of establishing an improvement in the proportion of late arrivals on a specific travel route. You will use a one-sided exact binomial proportion test with a null proportion of 30%, the frequency of late arrivals under the previous scheduling system, and a nominal significance level of $\alpha = 0.05$. Well-supported predictions estimate the new late arrival rate to be about 20%, and you will base your sample size determination on this assumption.

The POWER procedure does not currently compute exact sample size directly for the exact binomial test. But you can get an initial estimate by computing the approximate sample size required for a z test. Use the ONESAMPLEFREQ statement in the POWER procedure with TEST=Z and METHOD=NORMAL to compute the approximate sample size to achieve a power of 0.8 by using the z test. The following statements perform the analysis:

```
proc power;
   onesamplefreq test=z method=normal
      sides           = 1
      alpha           = 0.05
      nullproportion  = 0.3
      proportion      = 0.2
      ntotal          = .
      power           = 0.8;
run;
```

The NTOTAL= option with a missing value (.) indicates sample size as the result parameter. The SIDES=1 option specifies a one-sided test. The ALPHA=, NULLPROPORTION=, and POWER= options specify the significance level of 0.05, null value of 0.3, and target power of 0.8, respectively. The PROPORTION= option specifies your conjecture of 0.3 for the true proportion.

Output 67.2.1 Approximate Sample Size for z Test of a Proportion

The POWER Procedure
Z Test for Binomial Proportion

Fixed Scenario Elements

Method	Normal approximation
Number of Sides	1
Null Proportion	0.3
Alpha	0.05
Binomial Proportion	0.2
Nominal Power	0.8
Variance Estimate	Null Variance

Computed N Total

Actual Power	N Total
0.800	119

The results, shown in Output 67.2.1, indicate that you need to observe about N=119 flights to have an 80% chance of rejecting the hypothesis of a late arrival proportion of 30% or higher, if the true proportion is 20%, by using the z test. A similar analysis (Output 67.2.2) reveals an approxi-

mate sample size of N=129 for the z test with continuity correction, which is performed by using TEST=ADJZ:

```
proc power;
   onesamplefreq test=adjz method=normal
      sides           = 1
      alpha           = 0.05
      nullproportion  = 0.3
      proportion      = 0.2
      ntotal          = .
      power           = 0.8;
run;
```

Output 67.2.2 Approximate Sample Size for z Test with Continuity Correction

The POWER Procedure
Z Test for Binomial Proportion with Continuity Adjustment

Fixed Scenario Elements

Method	Normal approximation
Number of Sides	1
Null Proportion	0.3
Alpha	0.05
Binomial Proportion	0.2
Nominal Power	0.8
Variance Estimate	Null Variance

Computed N Total

Actual Power	N Total
0.801	129

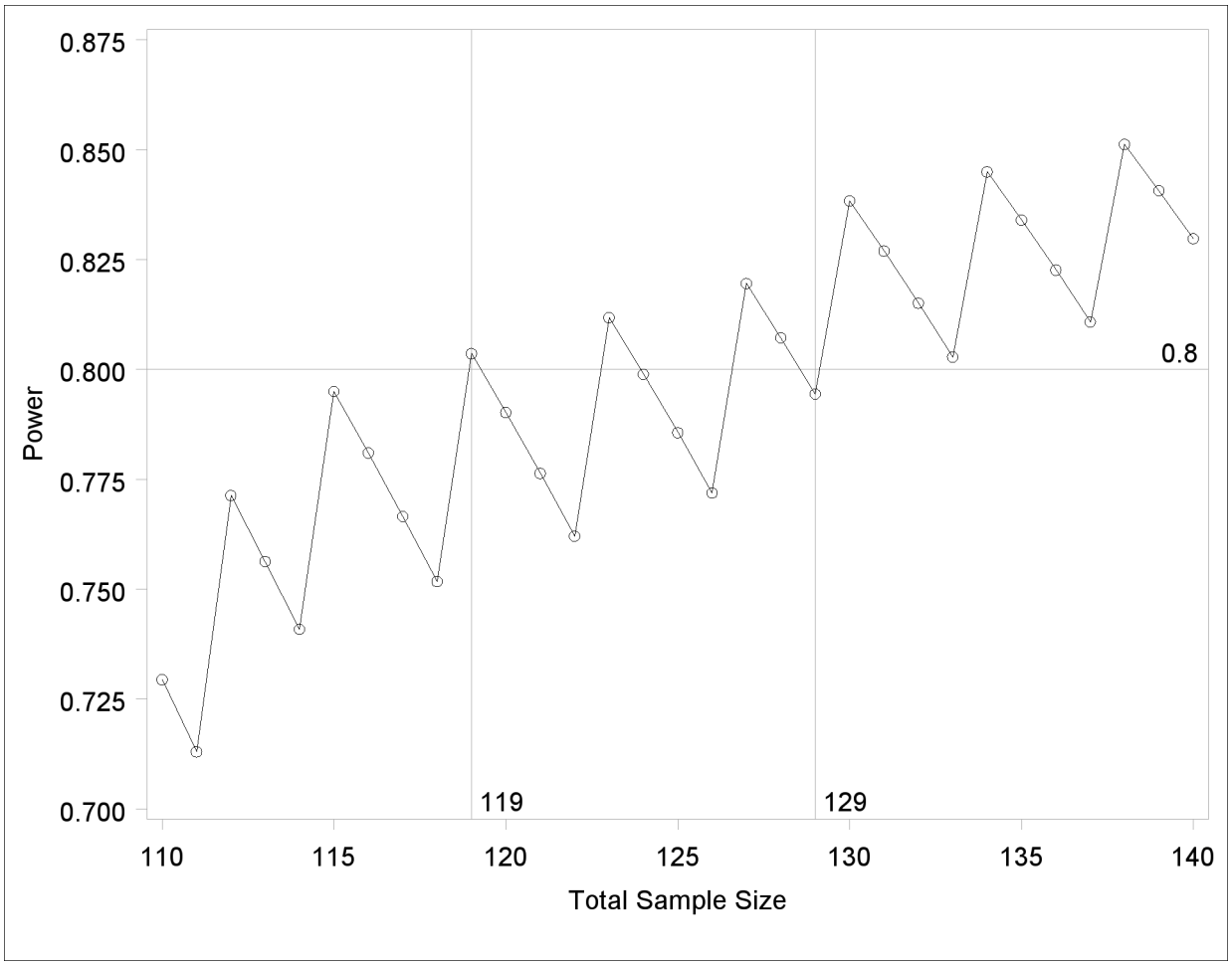
Based on the approximate sample size results, you decide to explore the power of the exact binomial test for sample sizes between 110 and 140. The following statements produce the plot:

```
proc power plotonly;
   onesamplefreq test=exact
      sides           = 1
      alpha           = 0.05
      nullproportion  = 0.3
      proportion      = 0.2
      ntotal          = 119
      power           = .;
   plot x=n min=110 max=140 step=1
      yopts=(ref=.8) xopts=(ref=119 129);
run;
```

The TEST=EXACT option in the ONESAMPLEFREQ statement specifies the exact binomial test, and the missing value (.) for the POWER= option indicates power as the result parameter. The PLOTONLY option in the PROC POWER statement disables nongraphical output. The PLOT

statement with X=N requests a plot with sample size on the X axis. The MIN= and MAX= options in the PLOT statement specify the sample size range. The YOPTS=(REF=) and XOPTS=(REF=) options add reference lines to highlight the approximate sample size results. The STEP=1 option produces a point at each integer sample size. The sample size value specified with the NTOTAL= option in the ONESAMPLEFREQ statement is overridden by the MIN= and MAX= options in the PLOT statement. Output 67.2.3 shows the resulting plot.

Output 67.2.3 Plot of Power versus Sample Size for Exact Binomial Test



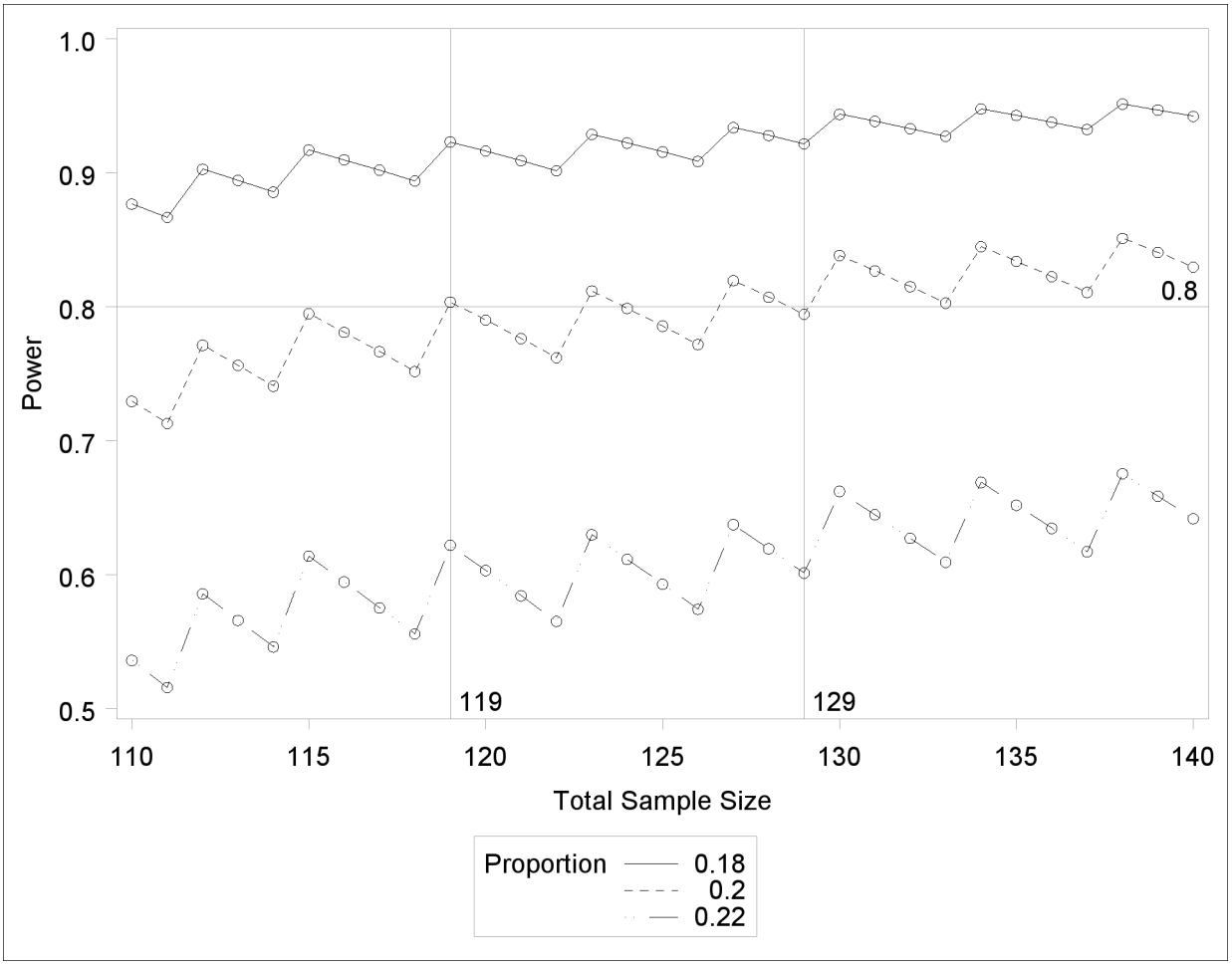
Note the sawtooth pattern in Output 67.2.3. Although the power surpasses the target level of 0.8 at N=119, it decreases to 0.79 with N=120 and further to 0.76 with N=122 before rising again to 0.81 with N=123. Not until N=130 does the power stay above the 0.8 target. Thus, a more conservative sample size recommendation of 130 might be appropriate, depending on the precise goals of the sample size determination.

In addition to considering alternative sample sizes, you might also want to assess the sensitivity of the power to inaccuracies in assumptions about the true proportion. The following statements produce a plot including true proportion values of 0.18 and 0.22. They are identical to the previous statements except for the additional true proportion values specified with the PROPORTION= option in the ONESAMPLEFREQ statement.

```
proc power plotonly;
   onesamplefreq test=exact
      sides           = 1
      alpha           = 0.05
      nullproportion = 0.3
      proportion      = 0.18 0.2 0.22
      ntotal          = 119
      power           = .;
   plot x=n min=110 max=140 step=1
      yopts=(ref=.8) xopts=(ref=119 129);
run;
```

Output 67.2.4 shows the resulting plot.

Output 67.2.4 Plot for Assessing Sensitivity to True Proportion Value



The plot reveals a dramatic sensitivity to the true proportion value. For N=119, the power is about 0.92 if the true proportion is 0.18, and as low as 0.62 if the proportion is 0.22. Note also that the power jumps occur at the same sample sizes in all three curves; the curves are only shifted and stretched vertically. This is because spikes and valleys in power curves are invariant to the true proportion value; they are due to changes in the critical value of the test.

A closer look at some ancillary output from the analysis sheds light on this property of the sawtooth pattern. You can add an ODS OUTPUT statement to save the plot content corresponding to Output 67.2.3 to a data set:

```
proc power plotonly;
   ods output plotcontent=PlotData;
   onesamplefreq test=exact
      sides          = 1
      alpha          = 0.05
      nullproportion = 0.3
      proportion     = 0.2
      ntotal         = 119
      power          = .;
   plot x=n min=110 max=140 step=1
      yopts=(ref=.8) xopts=(ref=119 129);
run;
```

The PlotData data set contains parameter values for each point in the plot. The parameters include underlying characteristics of the putative test. The following statements print the critical value and actual significance level along with sample size and power:

```
proc print data=PlotData;
   var NTotal LowerCritVal Alpha Power;
run;
```

Output 67.2.5 shows the plot data.

Output 67.2.5 Numerical Content of Plot

Obs	NTotal	Lower CritVal	Alpha	Power
1	110	24	0.0356	0.729
2	111	24	0.0313	0.713
3	112	25	0.0446	0.771
4	113	25	0.0395	0.756
5	114	25	0.0349	0.741
6	115	26	0.0490	0.795
7	116	26	0.0435	0.781
8	117	26	0.0386	0.767
9	118	26	0.0341	0.752
10	119	27	0.0478	0.804
11	120	27	0.0425	0.790
12	121	27	0.0377	0.776
13	122	27	0.0334	0.762
14	123	28	0.0465	0.812
15	124	28	0.0414	0.799
16	125	28	0.0368	0.786
17	126	28	0.0327	0.772
18	127	29	0.0453	0.820
19	128	29	0.0404	0.807
20	129	29	0.0359	0.794
21	130	30	0.0493	0.838
22	131	30	0.0441	0.827
23	132	30	0.0394	0.815
24	133	30	0.0351	0.803
25	134	31	0.0480	0.845
26	135	31	0.0429	0.834
27	136	31	0.0384	0.823
28	137	31	0.0342	0.811
29	138	32	0.0466	0.851
30	139	32	0.0418	0.841
31	140	32	0.0374	0.830

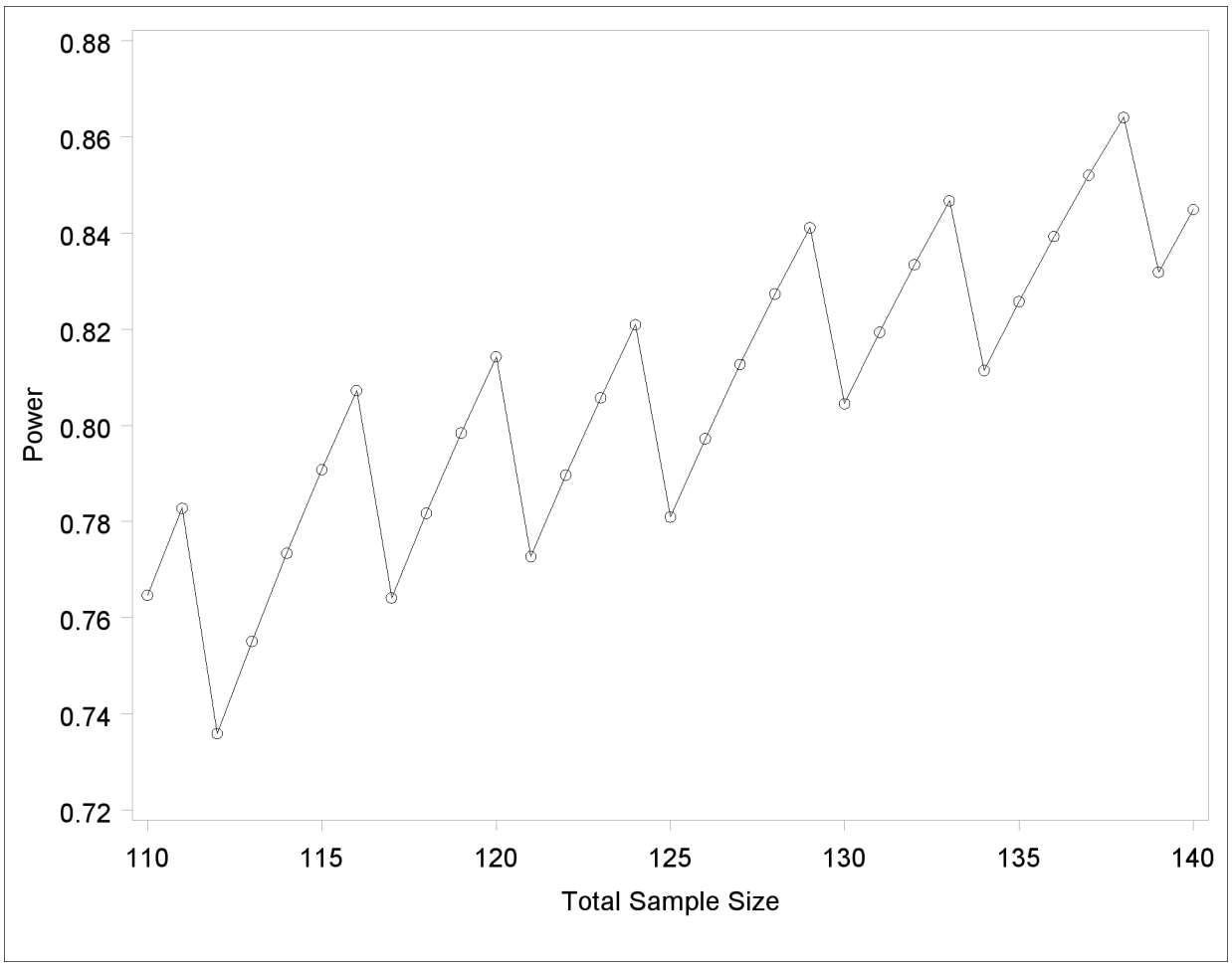
Note that whenever the critical value changes, the actual α jumps up to a value close to the nominal α=0.05, and the power also jumps up. Then while the critical value stays constant, the actual α and power slowly decrease. The critical value is independent of the true proportion value. So you can achieve a locally maximal power by choosing a sample size corresponding to a spike on the sawtooth curve, and this choice is locally optimal *regardless* of the unknown value of the true proportion. Locally optimal sample sizes in this case include 115, 119, 123, 127, 130, and 134.

As a point of interest, the power does not always jump sharply and decrease gradually. The shape of the sawtooth depends on the direction of the test and the location of the null proportion relative to 0.5. For example, if the direction of the hypothesis in this example is reversed (by switching true and null proportion values) so that the rejection region is in the upper tail, then the power curve exhibits sharp decreases and gradual increases. The following statements are similar to those producing the plot in Output 67.2.3 but with values of the PROPORTION= and NULLPROPORTION= options switched:

```
proc power plotonly;
   onesamplefreq test=exact
      sides           = 1
      alpha           = 0.05
      nullproportion = 0.2
      proportion      = 0.3
      ntotal          = 119
      power           = .;
   plot x=n min=110 max=140 step=1;
run;
```

The resulting plot is shown in Output 67.2.6.

Output 67.2.6 Plot of Power versus Sample Size for Another One-sided Test

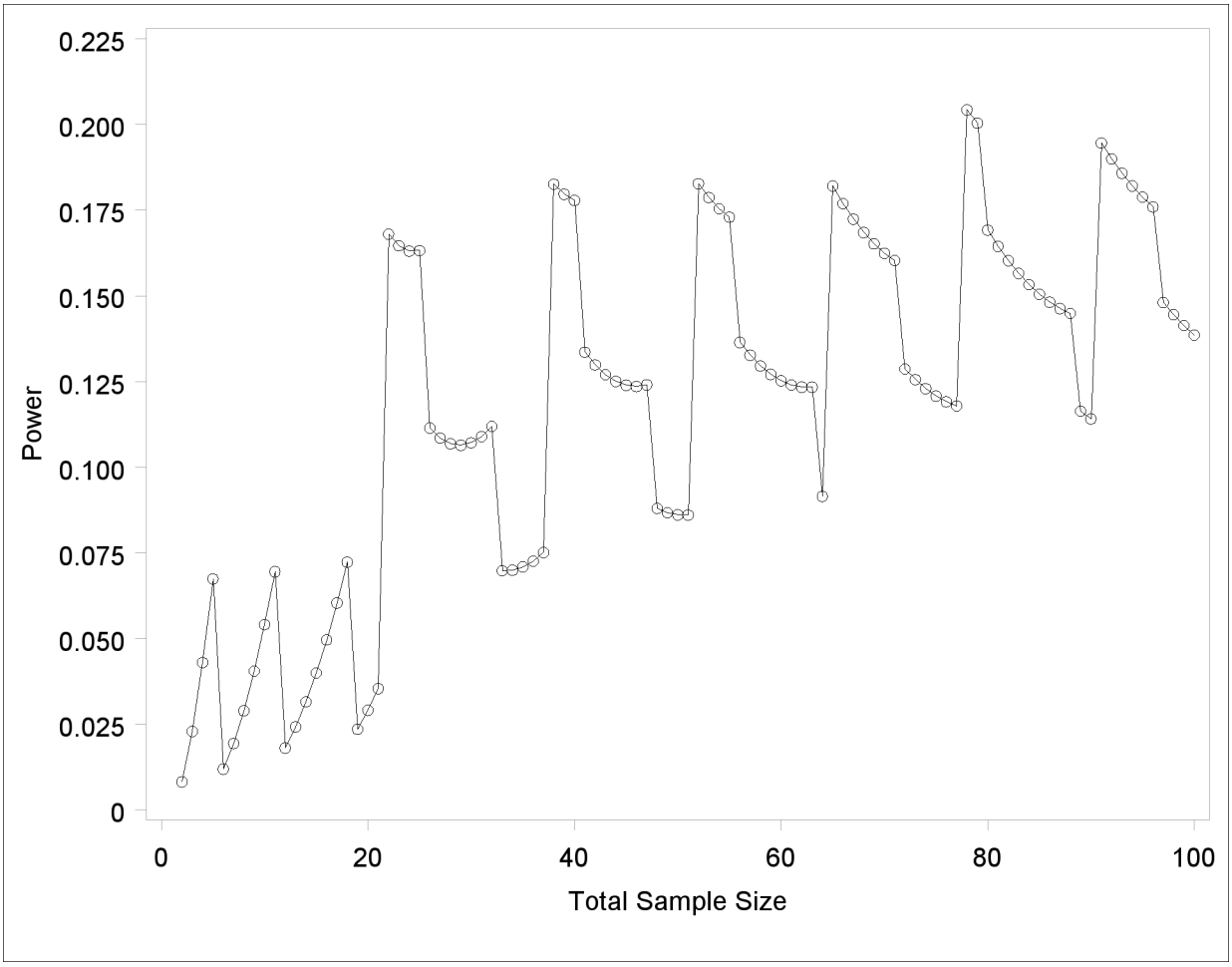


Finally, two-sided tests can lead to even more irregular power curve shapes, since changes in lower and upper critical values affect the power in different ways. The following statements produce a plot of power versus sample size for the scenario of a two-sided test with high alpha and a true proportion close to the null value:

```
proc power plotonly;
   onesamplefreq test=exact
      sides           = 2
      alpha           = 0.2
      nullproportion = 0.1
      proportion      = 0.09
      ntotal          = 10
      power           = .;
   plot x=n min=2 max=100 step=1;
run;
```

Output 67.2.7 shows the resulting plot.

Output 67.2.7 Plot of Power versus Sample Size for a Two-Sided Test



Due to the irregular shapes of power curves for proportion tests, the question "Which sample size should I use?" is often insufficient. A sample size solution produced directly in PROC POWER reveals the smallest possible sample size to achieve your target power. But as the examples in this section demonstrate, it is helpful to consult graphs for answers to questions such as the following:

- Which sample size will guarantee that all higher sample sizes also achieve my target power?

- Given a candidate sample size, can I increase it slightly to achieve locally maximal power, or perhaps even decrease it and get higher power?

Example 67.3: Simple AB/BA Crossover Designs

Crossover trials are experiments in which each subject is given a sequence of different treatments. They are especially common in clinical trials for medical studies. The reduction in variability from taking multiple measurements on a subject allows for more precise treatment comparisons. The simplest such design is the AB/BA crossover, in which each subject receives each of two treatments in a randomized order.

Under certain simplifying assumptions, you can test the treatment difference in an AB/BA crossover trial by using either a paired or two-sample t test (or equivalence test, depending on the hypothesis). This example will demonstrate when and how you can use the PAIREDMEANS statement in PROC POWER to perform power analyses for AB/BA crossover designs.

Senn (1993, Chapter 3) discusses a study comparing the effects of two bronchodilator medications in treatment of asthma, by using an AB/BA crossover design. Suppose you want to plan a similar study comparing two new medications, "Xilodol" and "Brantium." Half of the patients would be assigned to sequence AB, getting a dose of Xilodol in the first treatment period, a wash-out period of one week, and then a dose of Brantium in the second treatment period. The other half would be assigned to sequence BA, following the same schedule but with the drugs reversed. In each treatment period you would administer the drugs in the morning and then measure peak expiratory flow (PEF) at the end of the day, with higher PEF representing better lung function.

You conjecture that the mean and standard deviation of PEF are about $\mu_A = 310$ and $\sigma_A = 40$ for Xilodol and $\mu_B = 330$ and $\sigma_B = 55$ for Brantium, and that each pair of measurements on the same subject will have a correlation of about 0.3. You want to compute the power of both one-sided and two-sided tests of mean difference, with a significance level of $\alpha = 0.01$, for a sample size of 100 patients and also plot the power for a range of 50 to 200 patients. Note that the allocation ratio of patients to the two sequences is irrelevant in this analysis.

The choice of statistical test depends on which assumptions are reasonable. One possibility is a t test. A paired or two-sample t test is valid when there is no carryover effect and no interactions between patients, treatments, and periods. See Senn (1993, Chapter 3) for more details. The choice between a paired or a two-sample test depends on what you assume about the period effect. If you assume no period effect, then a paired t test is the appropriate analysis for the design, with the first member of each pair being the Xilodol measurement (regardless of which sequence the patient belongs to). Otherwise the two-sample t test approach is called for, since this analysis adjusts for the period effect by using an extra degree of freedom.

Suppose you assume no period effect. Then you can use the PAIREDMEANS statement in PROC POWER with the TEST=DIFF option to perform a sample size analysis for the paired t test. Indicate power as the result parameter by specifying the POWER= option with a missing value (.). Specify the conjectured means and standard deviations for each drug by using the PAIREDMEANS= and PAIREDSTDDEVS= options and the correlation by using the CORR= option. Specify both one- and two-sided tests by using the SIDES= option, the significance level by using the ALPHA= op-

tion, and the sample size (in terms of number of pairs) by using the NPAIRS= option. Generate a plot of power versus sample size by specifying the PLOT statement with X=N to request a plot with sample size on the X axis. (The result parameter, here power, is always plotted on the other axis.) Use the MIN= and MAX= options in the PLOT statement to specify the sample size range (as numbers of pairs).

The following statements perform the sample size analysis:

```
proc power;
   pairedmeans test=diff
      pairedmeans   = (330 310)
      pairedstddevs = (40 55)
      corr          = 0.3
      sides         = 1 2
      alpha         = 0.01
      npairs        = 100
      power         = .;
   plot x=n min=50 max=200;
run;
```

Default values for the NULLDIFF= and DIST= options specify a null mean difference of 0 and the assumption of normally distributed data. The output is shown in Output 67.3.1 and Output 67.3.2.

Output 67.3.1 Power for Paired *t* Analysis of Crossover Design

The POWER Procedure
Paired t Test for Mean Difference

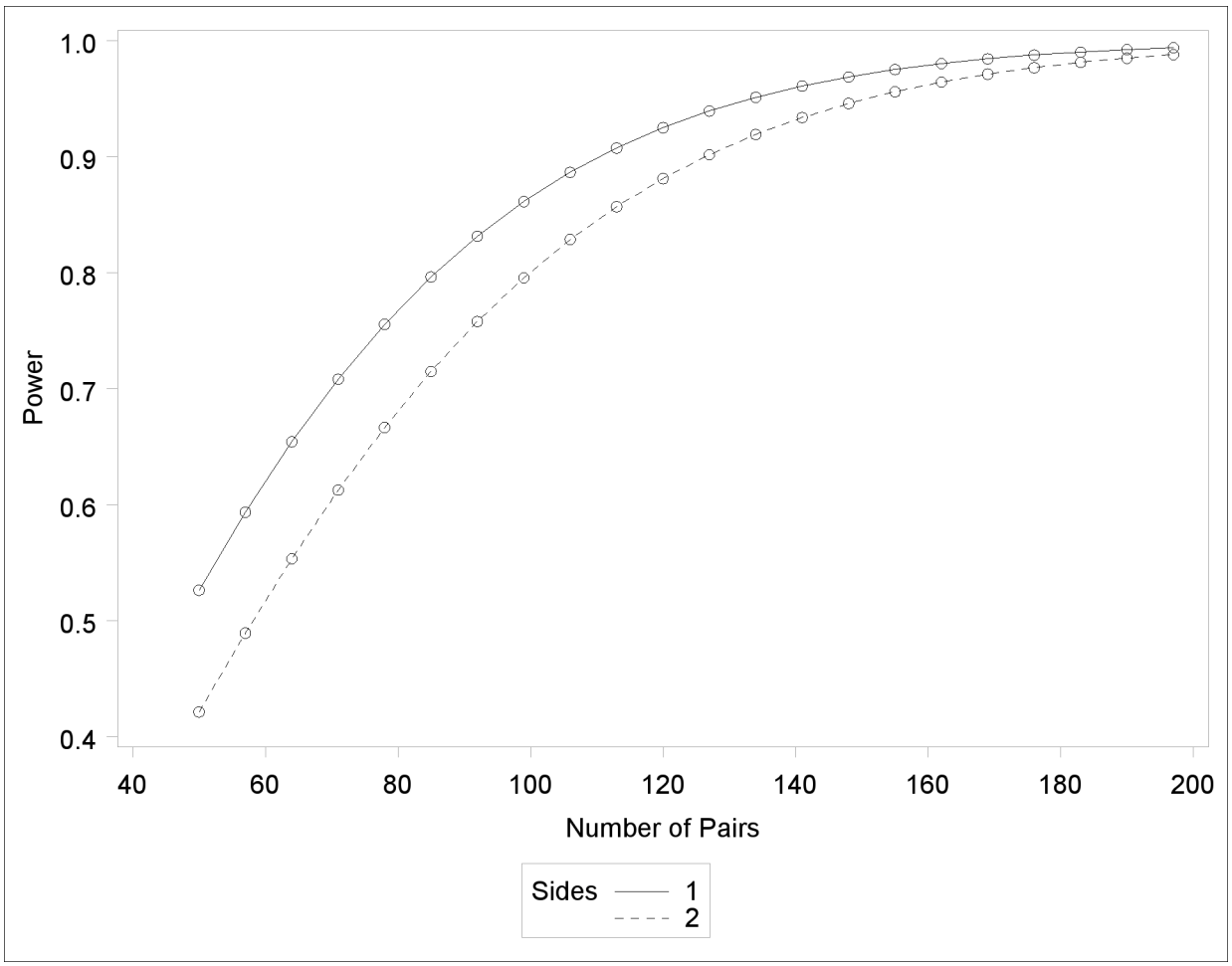
Fixed Scenario Elements

Distribution	Normal
Method	Exact
Alpha	0.01
Mean 1	330
Mean 2	310
Standard Deviation 1	40
Standard Deviation 2	55
Correlation	0.3
Number of Pairs	100
Null Difference	0

Computed Power

Index	Sides	Power
1	1	0.865
2	2	0.801

Output 67.3.2 Plot of Power versus Sample Size for Paired *t* Analysis of Crossover Design



The "Computed Power" table in Output 67.3.1 shows that the power with 100 patients is about 0.8 for the two-sided test and 0.87 for the one-sided test with the alternative of larger Brantium mean. In Output 67.3.2, the line style identifies the number of sides of the test. The plotting symbols identify locations of actual computed powers; the curves are linear interpolations of these points. The plot demonstrates how much higher the power is in the one-sided test than in the two-sided test for the range of sample sizes.

Suppose now that instead of detecting a difference between Xilodol and Brantium, you want to establish that they are similar—in particular, that the absolute mean PEF difference is at most 35. You might consider this goal if, for example, one of the drugs has fewer side effects and if a difference of no more than 35 is considered clinically small. Instead of a standard *t* test, you would conduct an *equivalence test* of the treatment mean difference for the two drugs. You would test the hypothesis that the true difference is less than –35 or more than 35 against the alternative that the mean difference is between –35 and 35, by using an additive model and a two one-sided tests ("TOST") analysis.

Assuming no period effect, you can use the PAIREDMEANS statement with the TEST=EQUIV_DIFF option to perform a sample size analysis for the paired equivalence test. Indicate power as the result parameter by specifying the POWER= option with a missing value (.).

Use the LOWER= and UPPER= options to specify the equivalence bounds of –35 and 35. Use the PAIREDMEANS=, PAIREDSTDDEVS=, CORR=, and ALPHA= options in the same way as in the *t* test at the beginning of this example to specify the remaining parameters.

The following statements perform the sample size analysis:

```
proc power;
   pairedmeans test=equiv_add
      lower          = -35
      upper          = 35
      pairedmeans    = (330 310)
      pairedstddevs = (40 55)
      corr           = 0.3
      alpha          = 0.01
      npairs         = 100
      power          = .;
run;
```

The default option DIST=NORMAL specifies an assumption of normally distributed data. The output is shown in Output 67.3.3.

Output 67.3.3 Power for Paired Equivalence Test for Crossover Design

The POWER Procedure
Equivalence Test for Paired Mean Difference

Fixed Scenario Elements

Distribution	Normal
Method	Exact
Lower Equivalence Bound	-35
Upper Equivalence Bound	35
Alpha	0.01
Reference Mean	330
Treatment Mean	310
Standard Deviation 1	40
Standard Deviation 2	55
Correlation	0.3
Number of Pairs	100

Computed Power

Power
0.598

The power for the paired equivalence test with 100 patients is about 0.6.

Example 67.4: Noninferiority Test with Lognormal Data

The typical goal in noninferiority testing is to conclude that a new treatment or process or product is not appreciably worse than some standard. This is accomplished by convincingly rejecting a one-sided null hypothesis that the new treatment is appreciably worse than the standard. When designing such studies, investigators must define precisely what constitutes "appreciably worse."

You can use the POWER procedure for sample size analyses for a variety of noninferiority tests, by specifying custom, one-sided null hypotheses for common tests. This example illustrates the strategy (often called Blackwelder's scheme; Blackwelder 1982) by comparing the means of two independent lognormal samples. The logic applies to one-sample, two-sample, and paired-sample problems involving normally distributed measures and proportions.

Suppose you are designing a study hoping to show that a new (less expensive) manufacturing process does not produce appreciably more pollution than the current process. Quantifying "appreciably worse" as 10%, you seek to show that the mean pollutant level from the new process is less than 110% of that from the current process. In standard hypothesis testing notation, you seek to reject

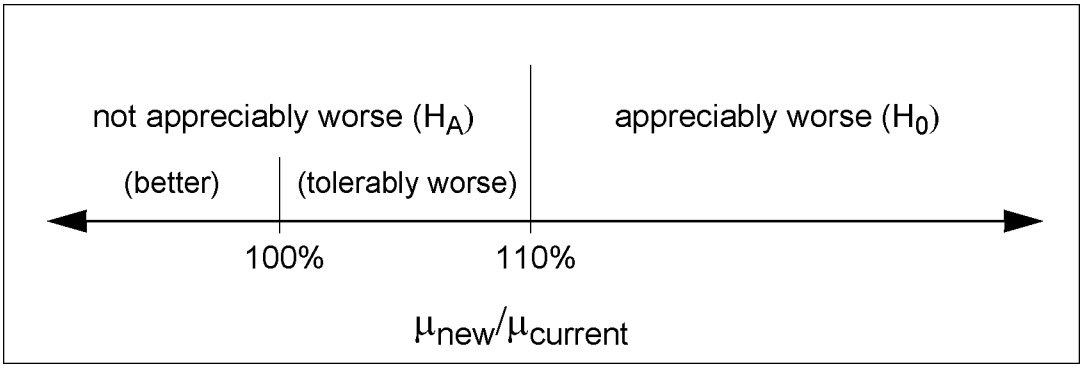
$$H_0: \frac{\mu_{\text{new}}}{\mu_{\text{current}}} \geq 1.10$$

in favor of

$$H_A: \frac{\mu_{\text{new}}}{\mu_{\text{current}}} < 1.10$$

This is described graphically in Figure 67.8. Mean ratios below 100% are better levels for the new process; a ratio of 100% indicates absolute equivalence; ratios of 100–110% are "tolerably" worse; and ratios exceeding 110% are appreciably worse.

Figure 67.8 Hypotheses for the Pollutant Study



An appropriate test for this situation is the common two-group t test on log-transformed data. The hypotheses become

$$H_0: \log(\mu_{\text{new}}) - \log(\mu_{\text{current}}) \geq \log(1.10)$$
$$H_A: \log(\mu_{\text{new}}) - \log(\mu_{\text{current}}) < \log(1.10)$$

Measurements of the pollutant level will be taken by using laboratory models of the two processes and will be treated as independent lognormal observations with a coefficient of variation (σ/μ)

between 0.5 and 0.6 for both processes. You will end up with 300 measurements for the current process and 180 for the new one. It is important to avoid a Type I error here, so you set the Type I error rate to 0.01. Your theoretical work suggests that the new process will actually reduce the pollutant by about 10% (to 90% of current), but you need to compute and graph the power of the study if the new levels are actually between 70% and 120% of current levels.

Implement the sample size analysis by using the TWOSAMPLEMEANS statement in PROC POWER with the TEST=RATIO option. Indicate power as the result parameter by specifying the POWER= option with a missing value (.). Specify a series of scenarios for the mean ratio between 0.7 and 1.2 by using the MEANRATIO= option. Use the NULLRATIO= option to specify the null mean ratio of 1.10. Specify SIDES=L to indicate a one-sided test with the alternative hypothesis stating that the mean ratio is *lower* than the null value. Specify the significance level, scenarios for the coefficient of variation, and the group sample sizes by using the ALPHA=, CV=, and GROUPNS= options. Generate a plot of power versus mean ratio by specifying the PLOT statement with the X=EFFECT option to request a plot with mean ratio on the X axis. (The result parameter, here power, is always plotted on the other axis.) Use the STEP= option in the PLOT statement to specify an interval of 0.05 between computed points in the plot.

The following statements perform the desired analysis:

```
proc power;
   twosamplemeans test=ratio
      meanratio = 0.7 to 1.2 by 0.1
      nullratio = 1.10
      sides     = L
      alpha     = 0.01
      cv        = 0.5 0.6
      groupns   = (300 180)
      power     = .;
   plot x=effect step=0.05;
run;
```

Note the use of SIDES=L, which forces computations for cases that need a rejection region that is opposite to the one providing the most one-tailed power; in this case, it is the lower tail. Such cases will show power that is less than the prescribed Type I error rate. The default option DIST=LOGNORMAL specifies the assumption of lognormally distributed data. The default MIN= and MAX= options in the plot statement specify an X axis range identical to the effect size range in the TWOSAMPLEMEANS statement (mean ratios between 0.7 and 1.2).

Output 67.4.1 and Output 67.4.2 show the results.

Output 67.4.1 Power for Noninferiority Test of Ratio

The POWER Procedure
Two-sample t Test for Mean Ratio

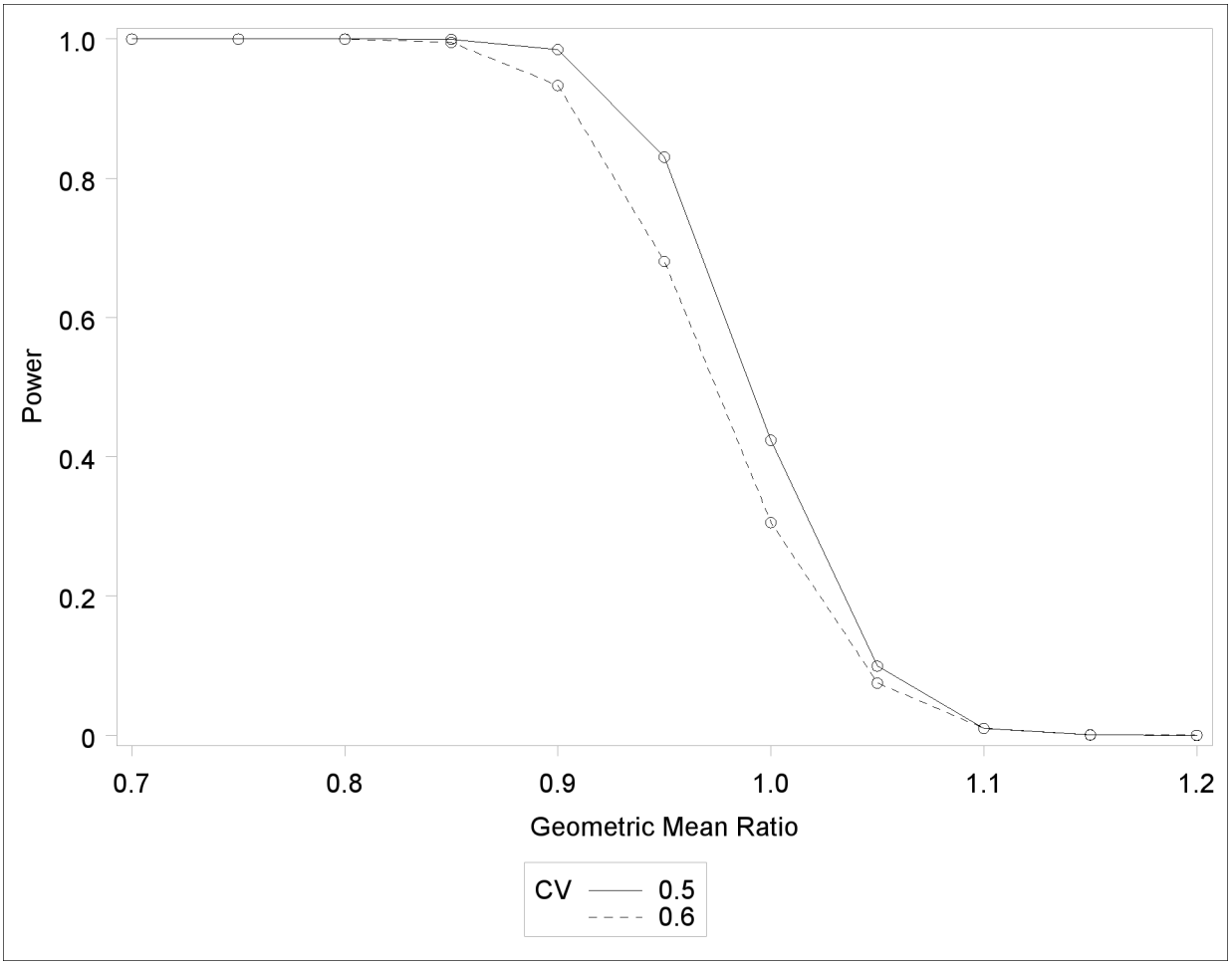
Fixed Scenario Elements

Distribution	Lognormal
Method	Exact
Number of Sides	L
Null Geometric Mean Ratio	1.1
Alpha	0.01
Group 1 Sample Size	300
Group 2 Sample Size	180

Computed Power

Index	Geo Mean Ratio	CV	Power
1	0.7	0.5	>.999
2	0.7	0.6	>.999
3	0.8	0.5	>.999
4	0.8	0.6	>.999
5	0.9	0.5	0.985
6	0.9	0.6	0.933
7	1.0	0.5	0.424
8	1.0	0.6	0.306
9	1.1	0.5	0.010
10	1.1	0.6	0.010
11	1.2	0.5	<.001
12	1.2	0.6	<.001

Output 67.4.2 Plot of Power versus Mean Ratio for Noninferiority Test



The "Computed Power" table in Output 67.4.1 shows that power exceeds 0.90 if the true mean ratio is 90% or less, as surmised. But power is unacceptably low (0.31–0.42) if the processes happen to be truly equivalent. Note that the power is identical to the alpha level (0.01) if the true mean ratio is 1.10 and below 0.01 if the true mean ratio is appreciably worse (>110%). In Output 67.4.2, the line style identifies the coefficient of variation. The plotting symbols identify locations of actual computed powers; the curves are linear interpolations of these points.

Example 67.5: Multiple Regression and Correlation

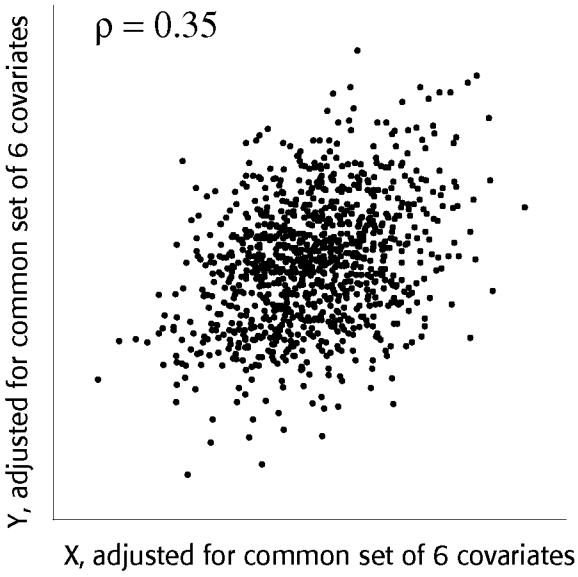
You are working with a team of preventive cardiologists investigating whether elevated serum homocysteine levels are linked to atherosclerosis (plaque buildup) in coronary arteries. The planned analysis is an ordinary least squares regression to assess the relationship between total homocysteine level (tHcy) and a plaque burden index (PBI), adjusting for six other variables: age, gender, plasma levels of folate, vitamin B_6, vitamin B_{12}, and a serum cholesterol index. You will regress PBI on tHcy and the six other predictors (plus the intercept) and use a Type III F test to assess whether

tHcy is a significant predictor after adjusting for the others. You wonder whether 100 subjects will provide adequate statistical power.

This is a correlational study at a single time. Subjects will be screened so that about half will have had a heart problem. All eight variables will be measured during one visit. Most clinicians are familiar with simple correlations between two variables, so you decide to pose the statistical problem in terms of estimating and testing the partial correlation between X_1 = tHcy and Y = PBI, controlling for the six other predictor variables ($R_{YX_1|X_{-1}}$). This greatly simplifies matters, especially the elicitation of the conjectured effect.

You use partial regression plots like that shown in Figure 67.9 to teach the team that the partial correlation between PBI and tHcy is the correlation of two sets of residuals obtained from ordinary regression models, one from regressing PBI on the six covariates and the other from regressing tHcy on the same covariates. Thus each subject has "expected" tHcy and PBI values based on the six covariates. The cardiologists believe that subjects whose tHcy is relatively higher than expected will also have a PBI that is relatively higher than expected. The partial correlation quantifies that adjusted association just as a standard simple correlation does with the unadjusted linear association between two variables.

Figure 67.9 Partial Regression Plot



Based on previously published studies of various coronary risk factors and after viewing a set of scatterplots showing various correlations, the team surmises that the true partial correlation is likely to be at least 0.35.

You want to compute the statistical power for a sample size of $N = 100$ by using $\alpha = 0.05$. You also want to plot power for sample sizes between 50 and 150. Use the MULTREG statement to compute the power and the PLOT statement to produce the graph. Since the predictors are observed rather than fixed in advanced, and a joint multivariate normal assumption seems tenable, use MODEL=RANDOM. The following statements perform the power analysis:

```
proc power;
   multreg
      model = random
      nfullpredictors = 7
      ntestpredictors = 1
      partialcorr = 0.35
      ntotal = 100
      power = .;
   plot x=n min=50 max=150;
run;
```

The POWER=. option identifies power as the parameter to compute. The NFULLPREDICTORS= option specifies seven total predictors (not including the intercept), and the NTESTPREDICTORS= option indicates that one of those predictors is being tested. The PARTIALCORR= and NTOTAL= options specify the partial correlation and sample size, respectively. The default value for the ALPHA= option sets the significance level to 0.05. The X=N option in the plot statement requests a plot of sample size on the X axis, and the MIN= and MAX= options specify the sample size range.

Output 67.5.1 shows the output, and Output 67.5.2 shows the plot.

Output 67.5.1 Power Analysis for Multiple Regression

The POWER Procedure
Type III F Test in Multiple Regression

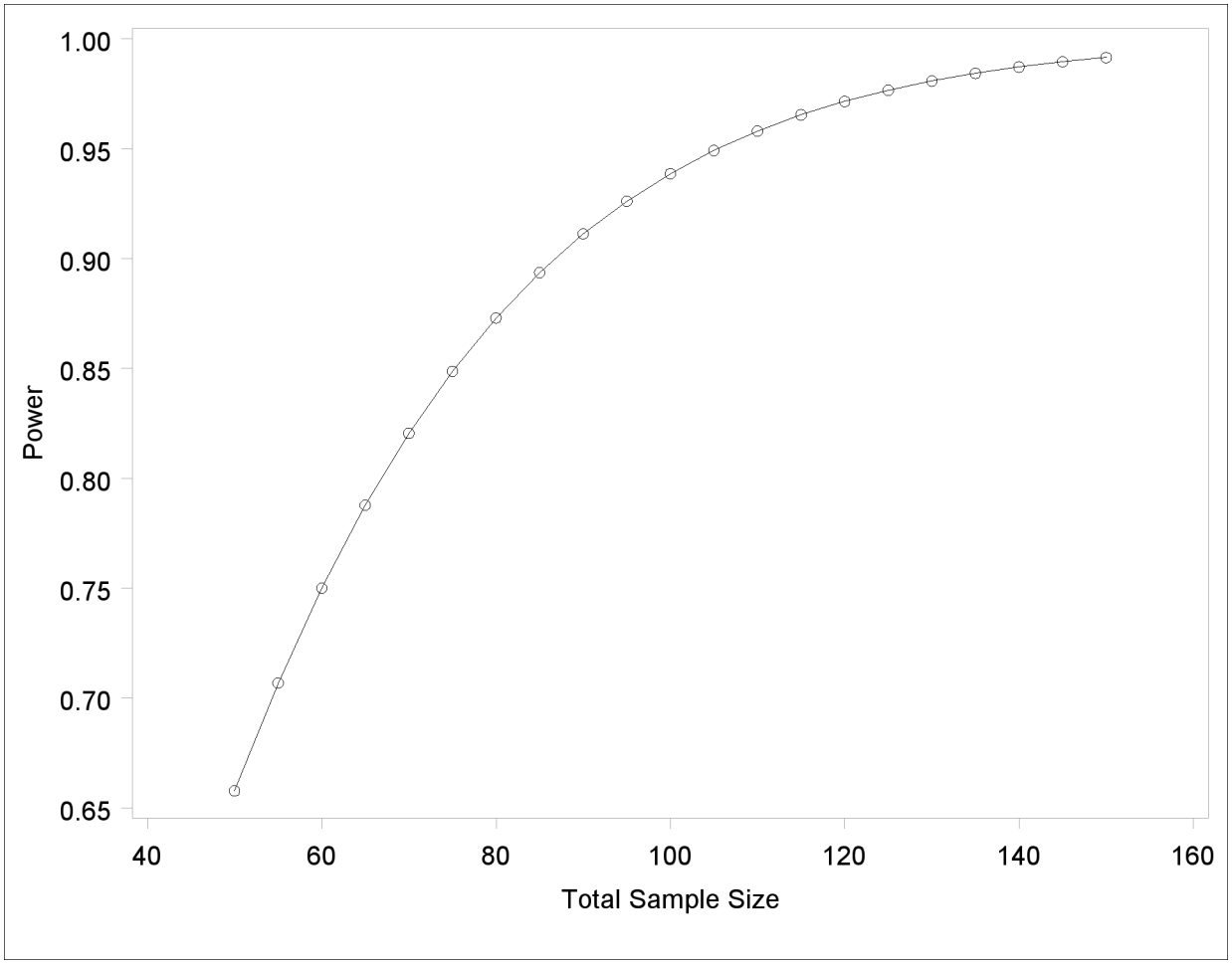
Fixed Scenario Elements

Method	Exact
Model	Random X
Number of Predictors in Full Model	7
Number of Test Predictors	1
Partial Correlation	0.35
Total Sample Size	100
Alpha	0.05

Computed Power

Power
0.939

Output 67.5.2 Plot of Power versus Sample Size for Multiple Regression



For the sample size $N = 100$, the study is almost balanced with respect to Type I and Type II error rates, with $\alpha = 0.05$ and $\beta = 1 - 0.937 = 0.063$. The study thus seems well designed at this sample size.

Now suppose that in a follow-up meeting with the cardiologists, you discover that their specific intent is to demonstrate that the (partial) correlation between PBI and tHcy is greater than 0.2. You suggest changing the planned data analysis to a one-sided Fisher's z test with a null correlation of 0.2. The following statements perform a power analysis for this test:

```
proc power;
   onecorr dist=fisherz
      npvars = 6
      corr = 0.35
      nullcorr = 0.2
      sides = 1
      ntotal = 100
      power = .;
run;
```

The DIST=FISHERZ option in the ONECORR statement specifies Fisher's z test. The NPARTIALVARS= option specifies that six additional variables are adjusted for in the partial correlation. The CORR= option specifies the conjectured correlation of 0.35, and the NULLCORR= option indicates the null value of 0.2. The SIDES= option specifies a one-sided test.

Output 67.5.3 shows the output.

Output 67.5.3 Power Analysis for Fisher's z Test

The POWER Procedure
Fisher's z Test for Pearson Correlation

Fixed Scenario Elements

Distribution	Fisher's z transformation of r
Method	Normal approximation
Number of Sides	1
Null Correlation	0.2
Number of Variables Partialled Out	6
Correlation	0.35
Total Sample Size	100
Nominal Alpha	0.05

Computed Power

Actual Alpha	Power
0.05	0.466

The power for Fisher's z test is less than 50%, the decrease being mostly due to the smaller effect size (relative to the null value). When asked for a recommendation for a new sample size goal, you compute the required sample size to achieve a power of 0.95 (to balance Type I and Type II errors) and 0.85 (a threshold deemed to be minimally acceptable to the team). The following statements perform the sample size determination:

```
proc power;
   onecorr dist=fisherz
      npvars = 6
      corr = 0.35
      nullcorr = 0.2
      sides = 1
      ntotal = .
      power = 0.85 0.95;
run;
```

The NTOTAL=. option identifies sample size as the parameter to compute, and the POWER= option specifies the target powers.

Output 67.5.4 Sample Size Determination for Fisher's *z* Test

The POWER Procedure
Fisher's z Test for Pearson Correlation

Fixed Scenario Elements

Distribution	Fisher's z transformation of r
Method	Normal approximation
Number of Sides	1
Null Correlation	0.2
Number of Variables Partialled Out	6
Correlation	0.35
Nominal Alpha	0.05

Computed N Total

Index	Nominal Power	Actual Alpha	Actual Power	N Total
1	0.85	0.05	0.850	280
2	0.95	0.05	0.950	417

The results in Output 67.5.4 reveal a required sample size of 417 to achieve a power of 0.95 and a required sample size of 280 to achieve a power of 0.85.

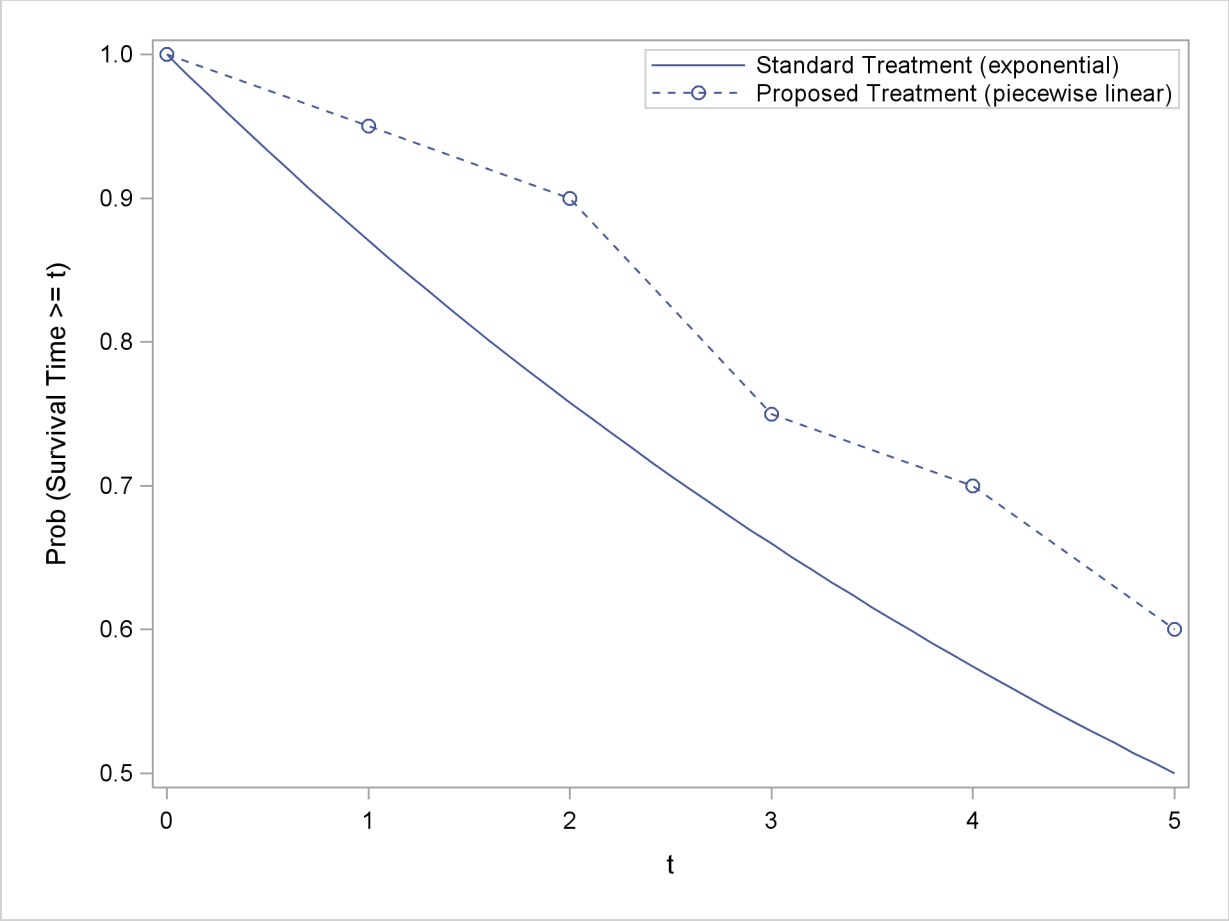
Example 67.6: Comparing Two Survival Curves

You are consulting for a clinical research group planning a trial to compare survival rates for proposed and standard cancer treatments. The planned data analysis is a log-rank test to nonparametrically compare the overall survival curves for the two treatments. Your goal is to determine an appropriate sample size to achieve a power of 0.8 for a two-sided test with $\alpha = 0.05$ by using a balanced design.

The survival curve for patients on the standard treatment is well known to be approximately exponential with a median survival time of five years. The research group conjectures that the new proposed treatment will yield a (nonexponential) survival curve similar to the dashed line in Figure 67.6.1.

Patients will be accrued uniformly over two years and then followed for an additional three years past the accrual period. Some loss to follow-up is expected, with roughly exponential rates that would result in about 50% loss with the standard treatment within 10 years. The loss to follow-up with the proposed treatment is more difficult to predict, but 50% loss would be expected to occur sometime between years 5 and 20.

Output 67.6.1 Survival Curves



Use the TWOSAMPLESURVIVAL statement with the TEST=LOGRANK option to compute the required sample size for the log-rank test. The following statements perform the analysis:

```
proc power;
   twosamplesurvival test=logrank
      curve("Standard") = 5 : 0.5
      curve("Proposed") = (1 to 5 by 1):(0.95 0.9 0.75 0.7 0.6)
      groupsurvival = "Standard" | "Proposed"
      accrualtime = 2
      followuptime = 3
      groupmedlosstimes = 10 | 20 5
      power = 0.8
      npergroup = .;
run;
```

The CURVE= option defines the two survival curves. The "Standard" curve has only one point, specifying an exponential form with a survival probability of 0.5 at year 5. The "Proposed" curve is a piecewise linear curve defined by the five points shown in Figure 67.6.1. The GROUPSURVIVAL= option assigns the survival curves to the two groups, and the ACCRUALTIME= and FOLLOWUPTIME= options specify the accrual and follow-up times. The GROUPMEDLOSSTIMES= option specifies the years at which 50% loss is expected to occur.

The POWER= option specifies the target power, and the NPERGROUP=. option identifies sample size per group as the parameter to compute. Default values for the SIDES= and ALPHA= options specify a two-sided test with $\alpha = 0.05$.

Output 67.6.2 shows the results.

Output 67.6.2 Sample Size Determination for Log-Rank Test

The POWER Procedure
Log-Rank Test for Two Survival Curves

Fixed Scenario Elements

Method	Lakatos normal approximation
Accrual Time	2
Follow-up Time	3
Group 1 Survival Curve	Standard
Form of Survival Curve 1	Exponential
Group 2 Survival Curve	Proposed
Form of Survival Curve 2	Piecewise Linear
Group 1 Median Loss Time	10
Nominal Power	0.8
Number of Sides	2
Number of Time Sub-Intervals	12
Alpha	0.05

Computed N Per Group

Index	Median Loss Time 2	Actual Power	N Per Group
1	20	0.800	228
2	5	0.801	234

The required sample size per group to achieve a power of 0.8 is 228 if the median loss time is 20 years for the proposed treatment. Only six more patients are required in each group if the median loss time is as short as five years.

Example 67.7: Confidence Interval Precision

An investment firm has hired you to help plan a study to estimate the success of a new investment strategy called "IntuiVest." The study involves complex simulations of market conditions over time, and it tracks the balance of a hypothetical brokerage account starting with $50,000. Each simulation is very expensive in terms of computing time. You are asked to determine an appropriate number of simulations to estimate the average change in the account balance at the end of three years. The goal is to have a 95% chance of obtaining a 90% confidence interval whose half-width is at most $1,000. That is, the firm wants to have a 95% chance of being able to correctly claim at the end of the study

that "Our research shows with 90% confidence that IntuiVest yields a profit of $X +/– $1,000 at the end of three years on an initial investment of $50,000 (under simulated market conditions)."

The probability of achieving the desired precision (that is, a small interval width) can be calculated either unconditionally or conditionally given that the true mean is captured by the interval. You decide to use the conditional form, considering two of its advantages:

- The conditional probability is usually lower than the unconditional probability for the same sample size, meaning that the conditional form is generally conservative.
- The overall probability of achieving the desired precision *and* capturing the true mean is easily computed as the product of the half-width probability and the confidence level. In this case, the overall probability is $0.95 \times 0.9 = 0.855$.

Based on some initial simulations, you expect a standard deviation between $25,000 and $45,000 for the ending account balance. You will consider both of these values in the sample size analysis.

As mentioned in the section "Overview of Power Concepts" on page 4916, an analysis of confidence interval precision is analogous to a traditional power analysis, with "CI Half-Width" taking the place of effect size and "Prob(Width)" taking the place of power. In this example, the target CI Half-Width is 1000, and the desired Prob(Width) is 0.95.

In addition to computing sample sizes for a half-width of $1,000, you are asked to plot the required number of simulations for a range of half-widths between $500 and $2,000. Use the ONESAMPLEMEANS statement with the CI=T option to implement the sample size determination. The following statements perform the analysis:

```
proc power;
   onesamplemeans ci=t
      alpha = 0.1
      halfwidth = 1000
      stddev = 25000 45000
      probwidth = 0.95
      ntotal = .;
   plot x=effect min=500 max=2000;
run;
```

The NTOTAL=. option identifies sample size as the parameter to compute. The ALPHA=0.1 option specifies a confidence level of $1 - \alpha = 0.9$. The HALFWIDTH= option specifies the target half-width, and the STDDEV= option specifies the conjectured standard deviation values. The PROBWIDTH= option specifies the desired probability of achieving the target precision. The default value PROBTYPE=CONDITIONAL specifies that this probability is conditional on the true mean being captured by the interval. The default of SIDES=2 indicates a two-sided interval.

Output 67.7.1 shows the output, and Output 67.7.2 shows the plot.

Output 67.7.1 Sample Size Determination for Confidence Interval Precision

The POWER Procedure
Confidence Interval for Mean

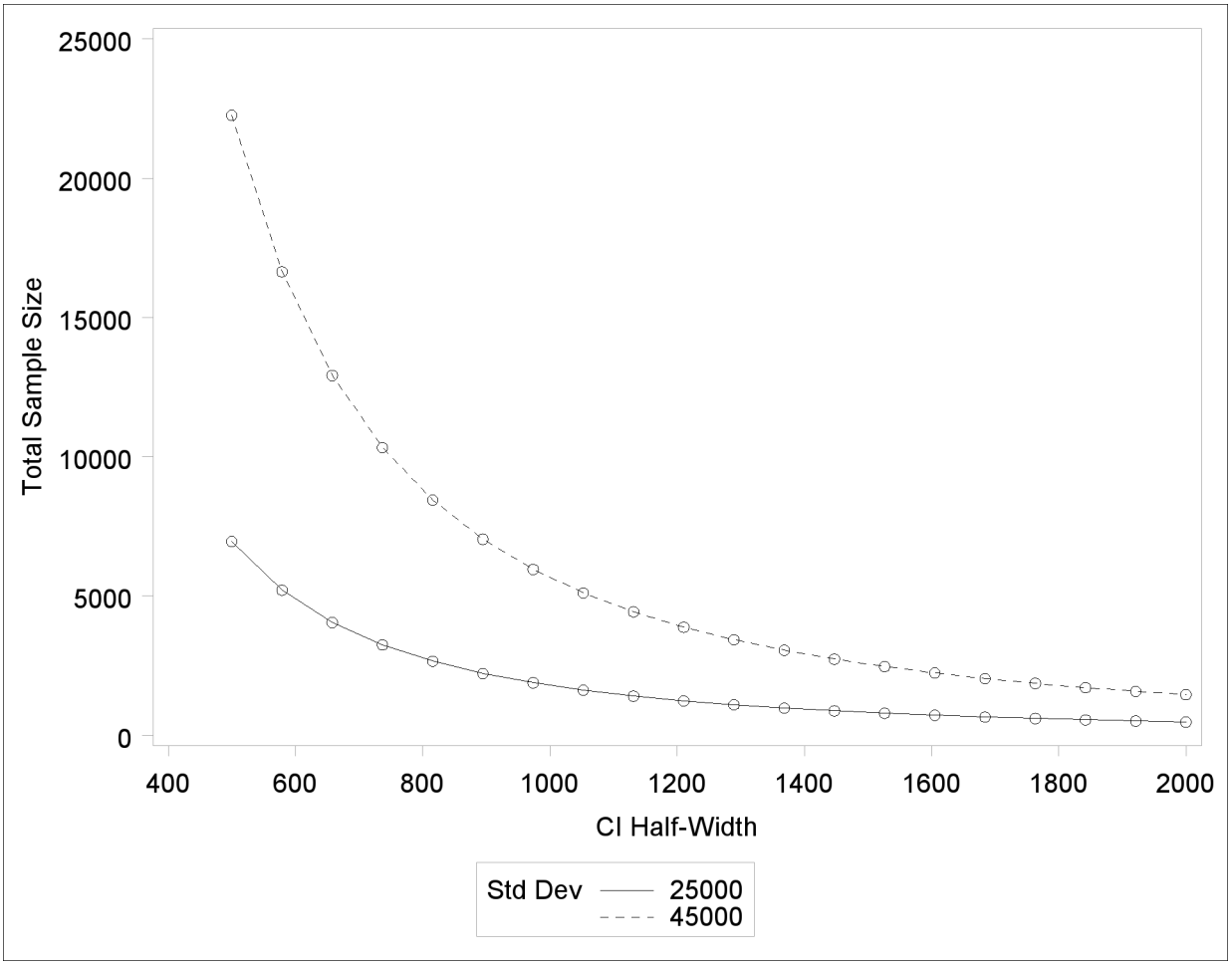
Fixed Scenario Elements

Distribution	Normal
Method	Exact
Alpha	0.1
CI Half-Width	1000
Nominal Prob(Width)	0.95
Number of Sides	2
Prob Type	Conditional

Computed N Total

Index	Std Dev	Actual Prob (Width)	N Total
1	25000	0.951	1788
2	45000	0.950	5652

Output 67.7.2 Plot of Sample Size versus Confidence Interval Half-Width



The number of simulations required in order to have a 95% chance of obtaining a half-width of at most 1000 is between 1788 and 5652, depending on the standard deviation. The plot reveals that more than 20,000 simulations would be required for a half-width of 500, assuming the higher standard deviation.

Example 67.8: Customizing Plots

This example demonstrates various ways you can modify and enhance plots:

- assigning analysis parameters to axes
- fine-tuning a sample size axis
- adding reference lines
- linking plot features to analysis parameters

- choosing key (legend) styles
- modifying symbol locations

The example plots are all based on a sample size analysis for a two-sample t test of group mean difference. You start by computing the sample size required to achieve a power of 0.9 by using a two-sided test with $\alpha = 0.05$, assuming the first mean is 12, the second mean is either 15 or 18, and the standard deviation is either 7 or 9.

Use the TWOSAMPLEMEANS statement with the TEST=DIFF option to compute the required sample sizes. Indicate total sample size as the result parameter by supplying a missing value (.) with the NTOTAL= option. Use the GROUPMEANS=, STDDEV=, and POWER= options to specify values of the other parameters. The following statements perform the sample size computations:

```
proc power;
   twosamplemeans test=diff
      groupmeans   = 12 | 15 18
      stddev       = 7 9
      power        = 0.9
      ntotal       = .;
run;
```

Default values for the NULLDIFF=, SIDES=, GROUPWEIGHTS=, and DIST= options specify a null mean difference of 0, two-sided test, balanced design, and assumption of normally distributed data, respectively.

Output 67.8.1 shows that the required sample size ranges from 60 to 382, depending on the unknown standard deviation and second mean.

Output 67.8.1 Computed Sample Sizes

The POWER Procedure
Two-sample t Test for Mean Difference

Fixed Scenario Elements

Distribution	Normal
Method	Exact
Group 1 Mean	12
Nominal Power	0.9
Number of Sides	2
Null Difference	0
Alpha	0.05
Group 1 Weight	1
Group 2 Weight	1

Computed N Total

Index	Mean2	Std Dev	Actual Power	N Total
1	15	7	0.902	232
2	15	9	0.901	382
3	18	7	0.904	60
4	18	9	0.904	98

Assigning Analysis Parameters to Axes

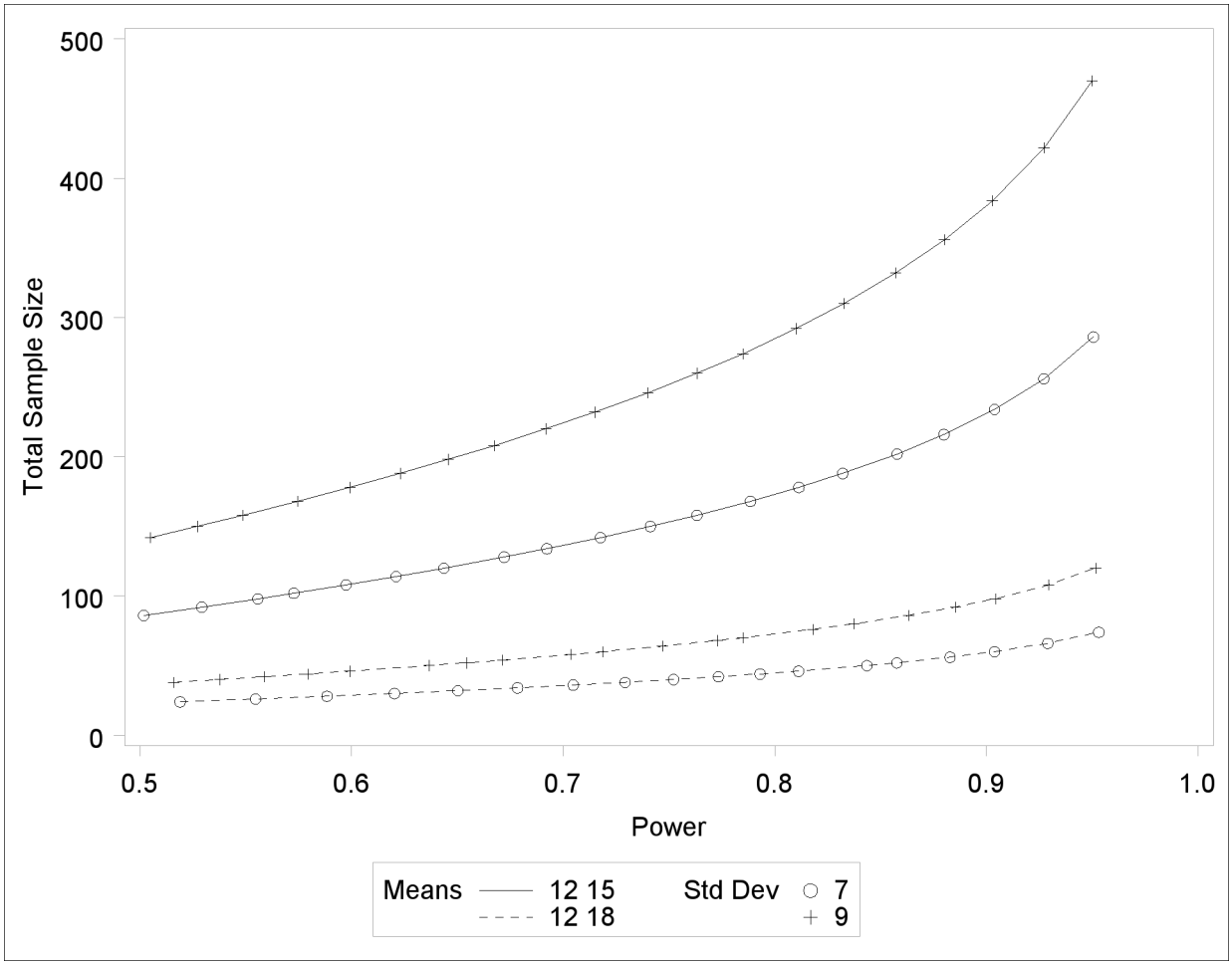
Use the PLOT statement to produce plots for all power and sample size analyses in PROC POWER. For the sample size analysis described at the beginning of this example, suppose you want to plot the required sample size on the Y axis against a range of powers between 0.5 and 0.95 on the X axis. The X= and Y= options specify which parameter to plot against the result and which axis to assign to this parameter. You can use either the X= or the Y= option, but not both. Use the X=POWER option in the PLOT statement to request a plot with power on the X axis. The result parameter, here total sample size, is always plotted on the other axis. Use the MIN= and MAX= options to specify the range of the axis indicated with either the X= or the Y= option. Here, specify MIN=0.5 and MAX=0.95 to specify the power range. The following statements produce the plot:

```
proc power plotonly;
   twosamplemeans test=diff
      groupmeans   = 12 | 15 18
      stddev       = 7 9
      power        = 0.9
      ntotal       = .;
   plot x=power min=0.5 max=0.95;
run;
```

Note that the value (0.9) of the POWER= option in the TWOSAMPLEMEANS statement is only a placeholder when the PLOTONLY option is used and both the MIN= and MAX= options are used, because the values of the MIN= and MAX= options override the value of 0.9. But the POWER= option itself is still required in the TWOSAMPLEMEANS statement, to provide a complete specification of the sample size analysis.

The resulting plot is shown in Output 67.8.2.

Output 67.8.2 Plot of Sample Size versus Power

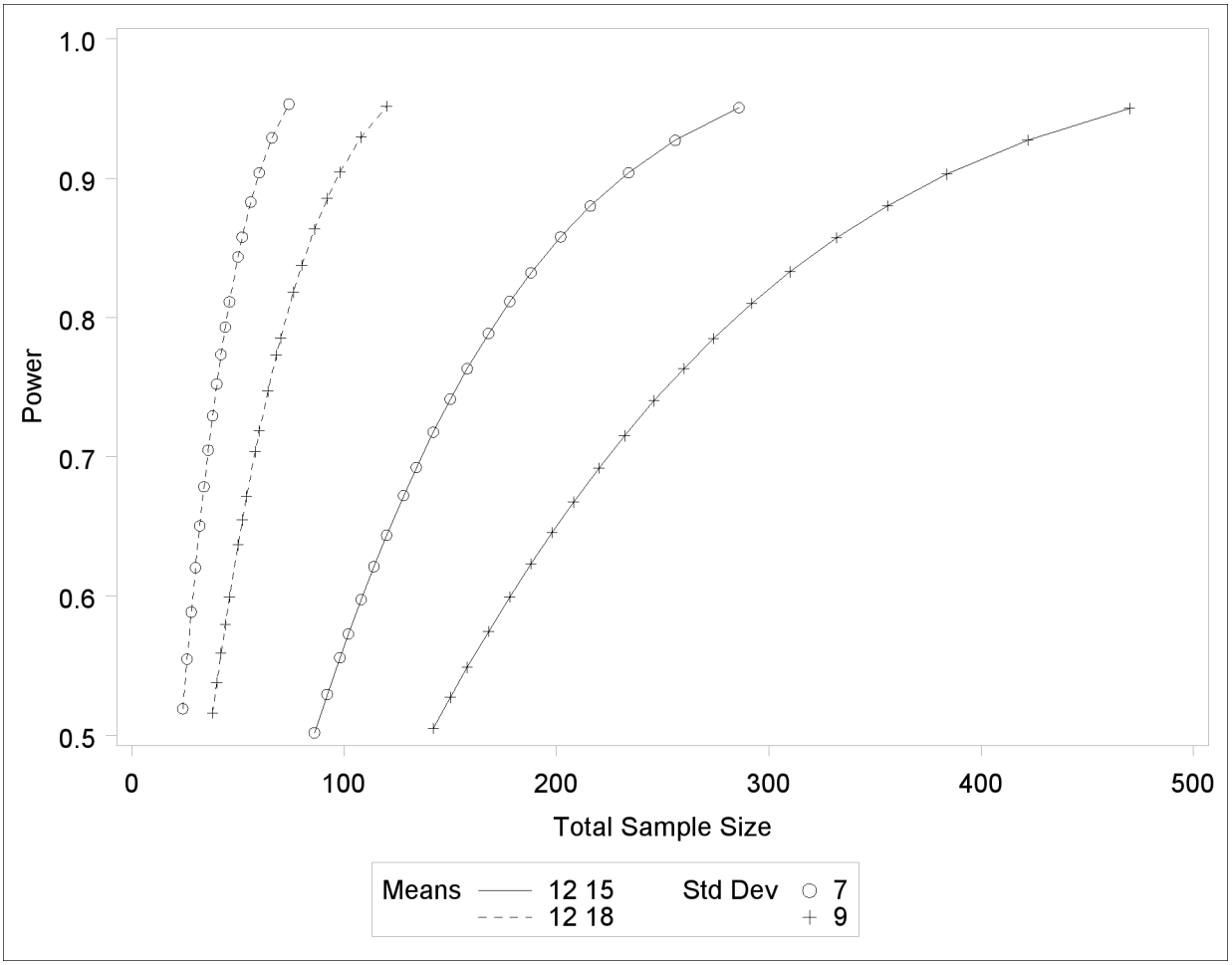


The line style identifies the group means scenario, and the plotting symbol identifies the standard deviation scenario. The locations of plotting symbols indicate computed sample sizes; the curves are linear interpolations of these points. By default, each curve consists of approximately 20 computed points (sometimes slightly more or less, depending on the analysis).

If you would rather plot power on the Y axis versus sample size on the X axis, you have two general strategies to choose from. One strategy is to use the Y= option instead of the X= option in the PLOT statement:

```
plot y=power min=0.5 max=0.95;
```

Output 67.8.3 Plot of Power versus Sample Size using First Strategy



Note that the resulting plot (Output 67.8.3) is essentially a mirror image of Output 67.8.2. The axis ranges are set such that each curve in Output 67.8.3 contains similar values of Y instead of X. Each plotted point represents the computed value of the X axis at the input value of the Y axis.

A second strategy for plotting power versus sample size (when originally solving for sample size) is to invert the analysis and base the plot on computed power for a given range of sample sizes. This strategy works well for monotonic power curves (as is the case for the *t* test and most other continuous analyses). It is advantageous in the sense of preserving the traditional role of the Y axis as the computed parameter. A common way to implement this strategy is as follows:

- Determine the range of sample sizes sufficient to cover at the desired power range for all curves (where each "curve" represents a scenario for standard deviation and second group mean).
- Use this range for the X axis of a plot.

To determine the required sample sizes for target powers of 0.5 and 0.95, change the values in the POWER= option as follows to reflect this range:

```
proc power;
   twosamplemeans test=diff
      groupmeans   = 12 | 15 18
      stddev       = 7 9
      power        = 0.5 0.95
      ntotal       = .;
run;
```

Output 67.8.4 reveals that a sample size range of 24 to 470 is approximately sufficient to cover the desired power range of 0.5 to 0.95 for all curves ("approximately" because the actual power at the rounded sample size of 24 is slightly higher than the nominal power of 0.5).

Output 67.8.4 Computed Sample Sizes

The POWER Procedure
Two-sample t Test for Mean Difference

Fixed Scenario Elements

Distribution	Normal
Method	Exact
Group 1 Mean	12
Number of Sides	2
Null Difference	0
Alpha	0.05
Group 1 Weight	1
Group 2 Weight	1

Computed N Total

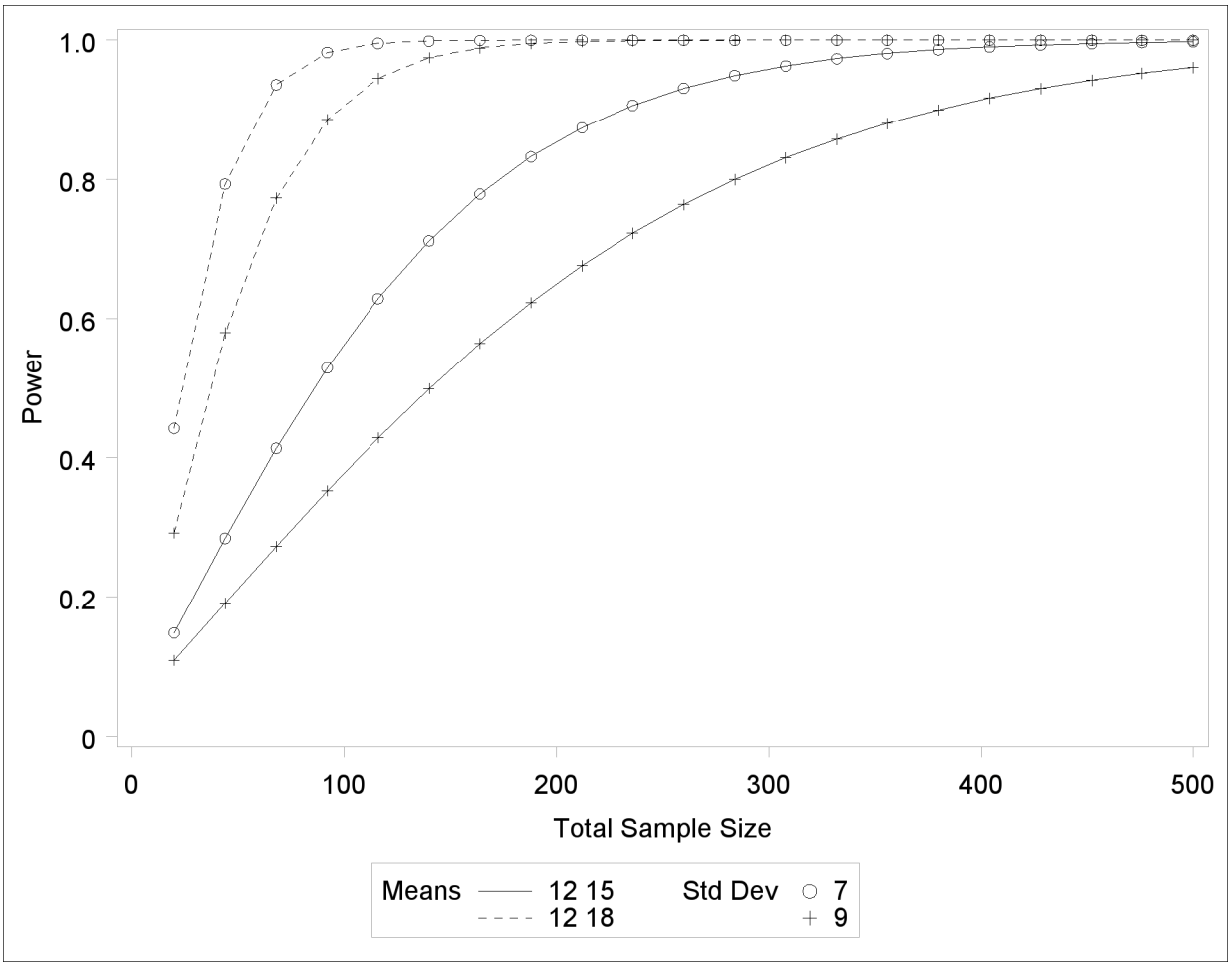
Index	Mean2	Std Dev	Nominal Power	Actual Power	N Total
1	15	7	0.50	0.502	86
2	15	7	0.95	0.951	286
3	15	9	0.50	0.505	142
4	15	9	0.95	0.950	470
5	18	7	0.50	0.519	24
6	18	7	0.95	0.953	74
7	18	9	0.50	0.516	38
8	18	9	0.95	0.952	120

To plot power on the Y axis for sample sizes between 20 and 500, use the X=N option in the PLOT statement with MIN=20 and MAX=500:

```
proc power plotonly;
   twosamplemeans test=diff
      groupmeans   = 12 | 15 18
      stddev       = 7 9
      power        = .
      ntotal       = 200;
   plot x=n min=20 max=500;
run;
```

Each curve in the resulting plot in Output 67.8.5 covers at least a power range of 0.5 to 0.95.

Output 67.8.5 Plot of Power versus Sample Size Using Second Strategy

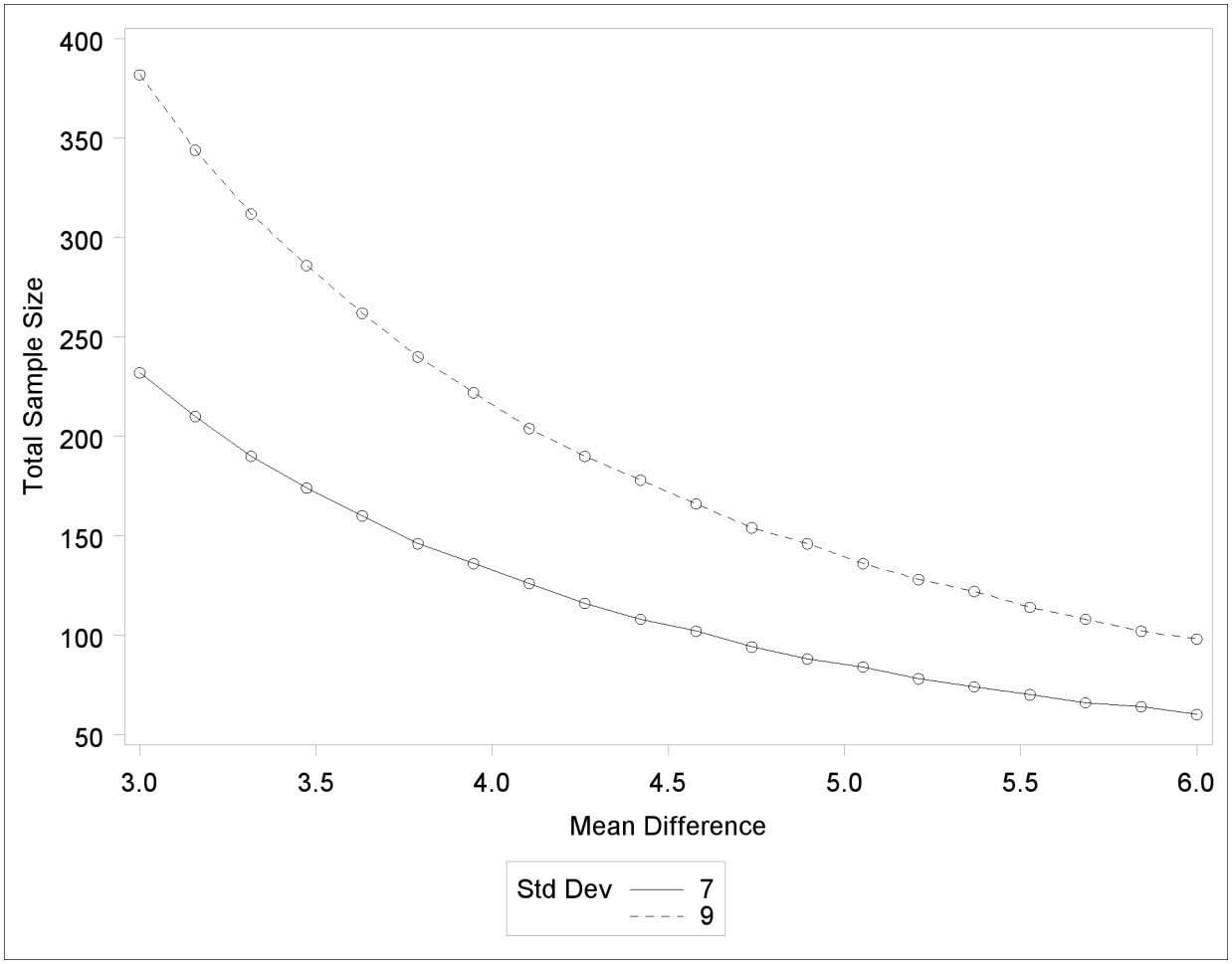


Finally, suppose you want to produce a plot of sample size versus effect size for a power of 0.9. In this case, the "effect size" is defined to be the mean difference. You need to reparameterize the analysis by using the MEANDIFF= option instead of the GROUPMEANS= option to produce a plot, since each plot axis must be represented by a scalar parameter. Use the X=EFFECT option in the PLOT statement to assign the mean difference to the X axis. The following statements produce a plot of required sample size to detect mean differences between 3 and 6:

```
proc power plotonly;
   twosamplemeans test=diff
      meandiff     = 3 6
      stddev       = 7 9
      power        = 0.9
      ntotal       = .;
   plot x=effect min=3 max=6;
run;
```

The resulting plot Output 67.8.6 shows how the required sample size decreases with increasing mean difference.

Output 67.8.6 Plot of Sample Size versus Mean Difference



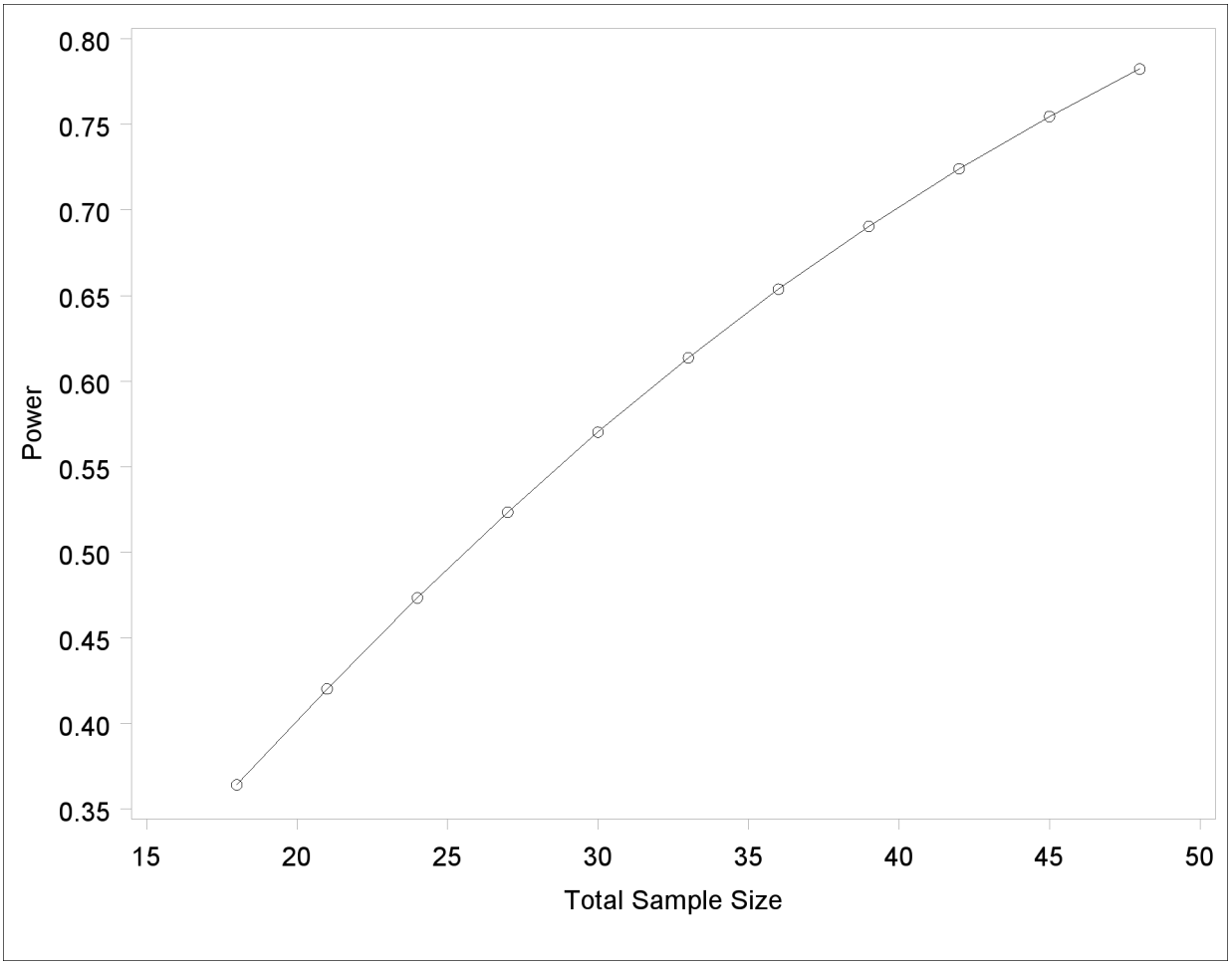
Fine-Tuning a Sample Size Axis

Consider the following plot request for a sample size analysis similar to the one in Output 67.8.1 but with only a single scenario, and with unbalanced sample size allocation of 2:1:

```
proc power plotonly;
   ods output plotcontent=PlotData;
   twosamplemeans test=diff
      groupmeans   = 12 | 18
      stddev       = 7
      groupweights = 2 | 1
      power        = .
      ntotal       = 20;
   plot x=n min=20 max=50 npoints=20;
run;
```

The MIN=, MAX=, and NPOINTS= options in the PLOT statement request a plot with 20 points between 20 and 50. But the resulting plot (Output 67.8.7) appears to have only 11 points, and they range from 18 to 48.

Output 67.8.7 Plot with Overlapping Points



The reason that this plot has fewer points than usual is due to the rounding of sample sizes. If you do not use the NFRACTIONAL option in the analysis statement (here, the TWOSAMPLEMEANS statement), then the set of sample size points determined by the MIN=, MAX=, NPOINTS=, and STEP= options in the PLOT statement can be rounded to satisfy the allocation weights. In this case, they are rounded down to the nearest multiples of 3 (the sum of the weights), and many of the points overlap. To see the overlap, you can print the NominalNTotal (unadjusted) and NTotal (rounded) variables in the PlotContent ODS object (here saved to a data set called PlotData):

```
proc print data=PlotData;
   var NominalNTotal NTotal;
run;
```

The output is shown in Output 67.8.8.

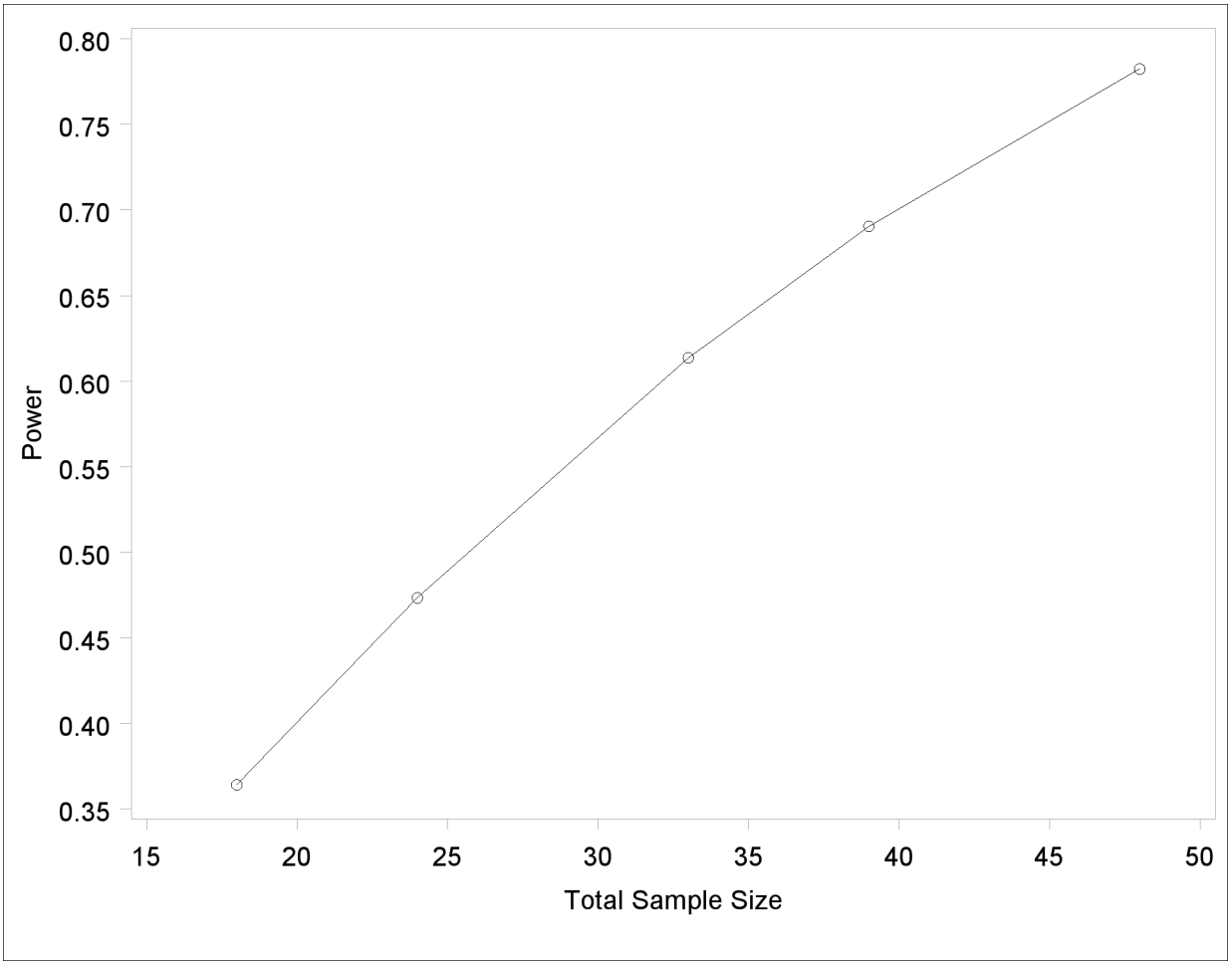
Output 67.8.8 Sample Sizes

Obs	Nominal NTotal	NTotal
1	18.0	18
2	19.6	18
3	21.2	21
4	22.7	21
5	24.3	24
6	25.9	24
7	27.5	27
8	29.1	27
9	30.6	30
10	32.2	30
11	33.8	33
12	35.4	33
13	36.9	36
14	38.5	36
15	40.1	39
16	41.7	39
17	43.3	42
18	44.8	42
19	46.4	45
20	48.0	48

Besides overlapping of sample size points, another peculiarity that might occur without the NFRACTIONAL option is unequal spacing—for example, in the plot in Output 67.8.9, created with the following statements:

```
proc power plotonly;
   twosamplemeans test=diff
      groupmeans   = 12 | 18
      stddev       = 7
      groupweights = 2 | 1
      power        = .
      ntotal       = 20;
   plot x=n min=20 max=50 npoints=5;
run;
```

Output 67.8.9 Plot with Unequally Spaced Points

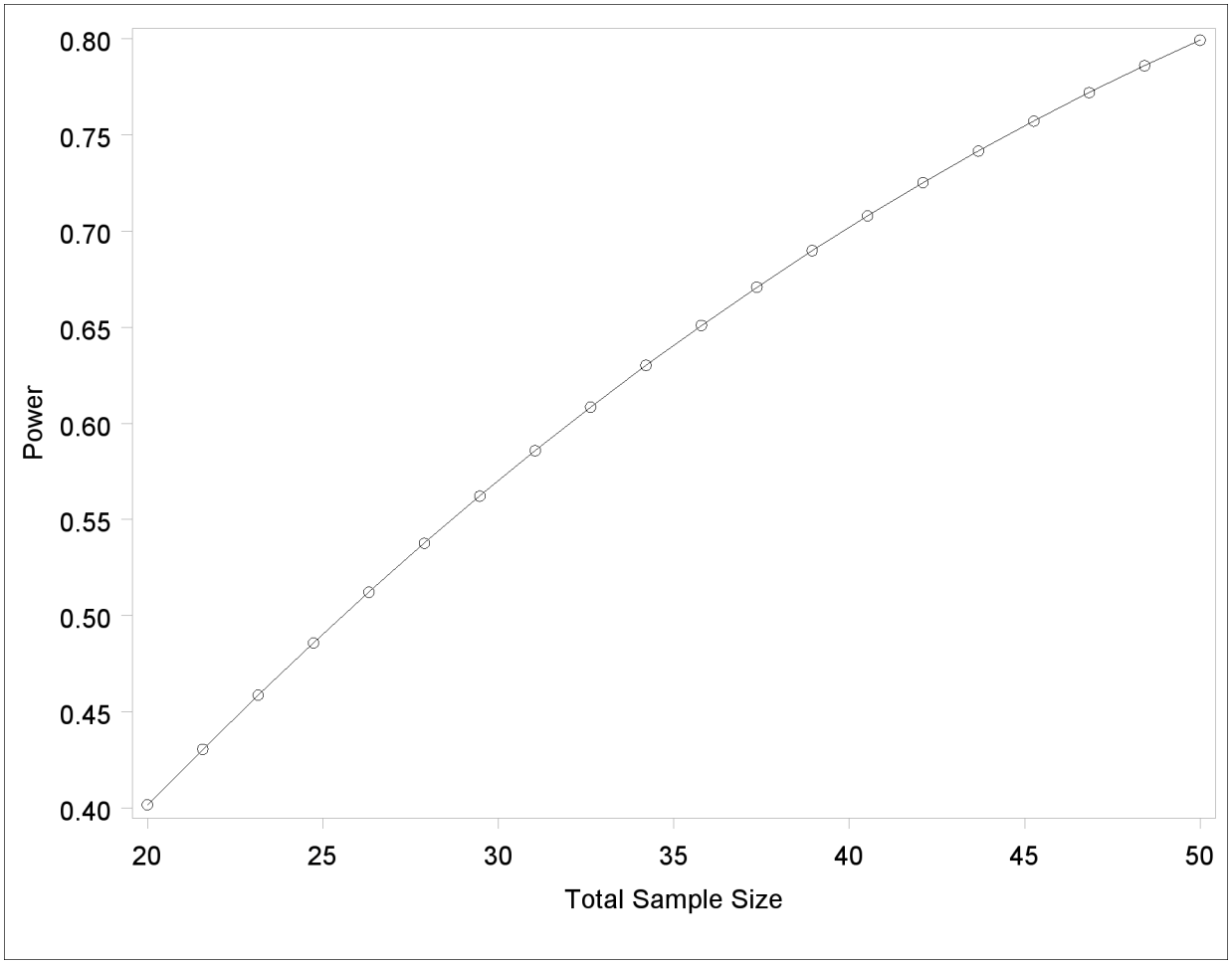


If you want to guarantee evenly spaced, nonoverlapping sample size points in your plots, you can either (1) use the NFRACTIONAL option in the analysis statement preceding the PLOT statement or (2) use the STEP= option and provide values for the MIN=, MAX=, and STEP= options in the PLOT statement that are multiples of the sum of the allocation weights. Note that this sum is simply 1 for one-sample and paired designs and 2 for balanced two-sample designs. So any integer step value works well for one-sample and paired designs, and any even step value works well for balanced two-sample designs. Both of these strategies will avoid rounding adjustments.

The following statements implement the first strategy to create the plot in Output 67.8.10, by using the NFRACTIONAL option in the TWOSAMPLEMEANS statement:

```
proc power plotonly;
   twosamplemeans test=diff
      nfractional
      groupmeans   = 12 | 18
      stddev       = 7
      groupweights = 2 | 1
      power        = .
      ntotal       = 20;
   plot x=n min=20 max=50 npoints=20;
run;
```

Output 67.8.10 Plot with Fractional Sample Sizes



To implement the second strategy, use multiples of 3 for the STEP=, MIN=, and MAX= options in the PLOT statement (because the sum of the allocation weights is $2 + 1 = 3$). The following statements use STEP=3, MIN=18, and MAX=48 to create a plot that looks identical to the plot in Output 67.8.7 but suffers no overlapping of points:

```
proc power plotonly;
   twosamplemeans test=diff
      groupmeans   = 12 | 18
      stddev       = 7
      groupweights = 2 | 1
      power        = .
      ntotal       = 20;
   plot x=n min=18 max=48 step=3;
run;
```

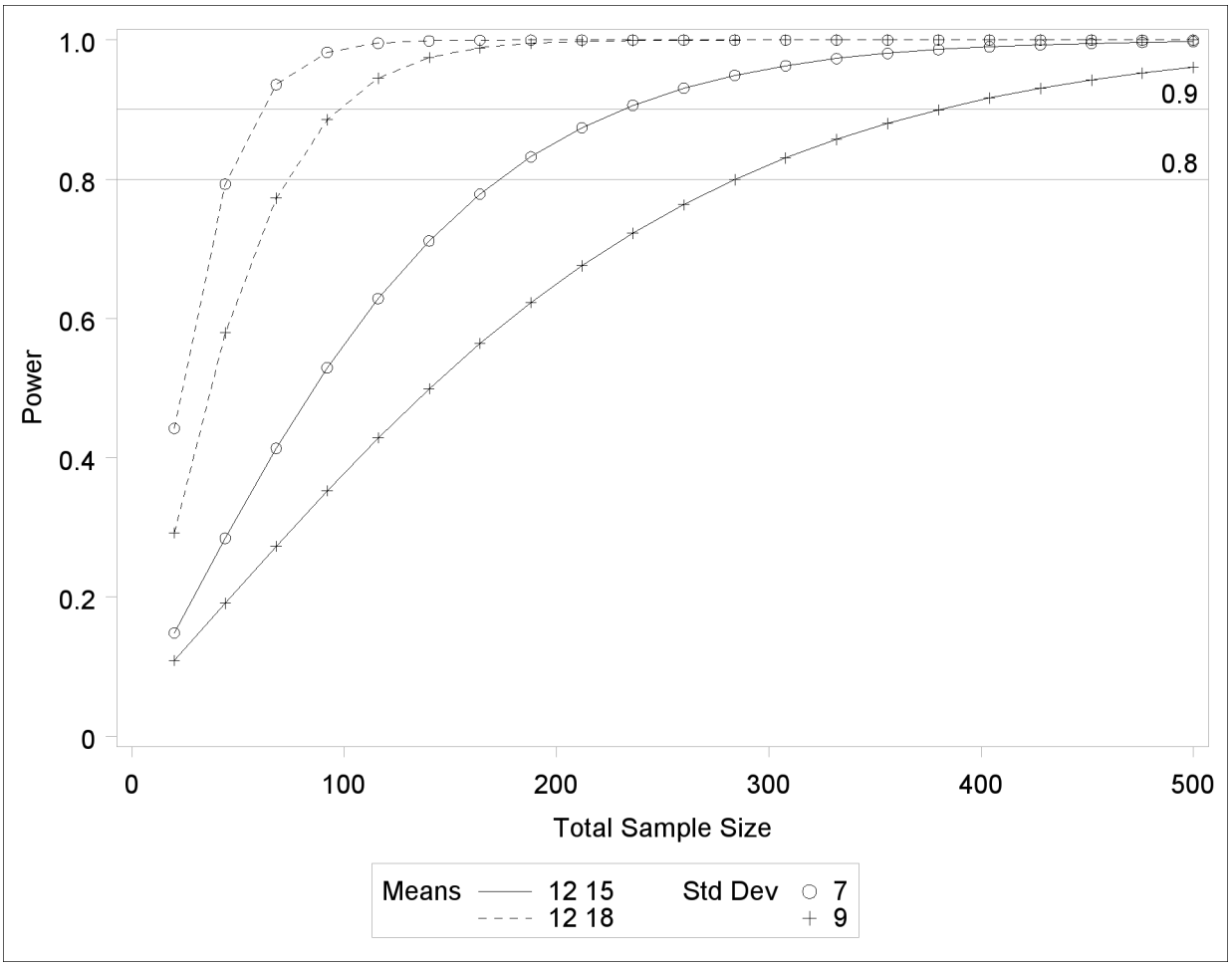
Adding Reference Lines

Suppose you want to add reference lines to highlight power=0.8 and power=0.9 on the plot in Output 67.8.5. You can add simple reference lines by using the YOPTS= option and REF= subop-

tion in the PLOT statement to produce Output 67.8.11, with the following statements:

```
proc power plotonly;
   twosamplemeans test=diff
      groupmeans   = 12 | 15 18
      stddev       = 7 9
      power        = .
      ntotal       = 100;
   plot x=n min=20 max=500
      yopts=(ref=0.8 0.9);
run;
```

Output 67.8.11 Plot with Simple Reference Lines on Y Axis

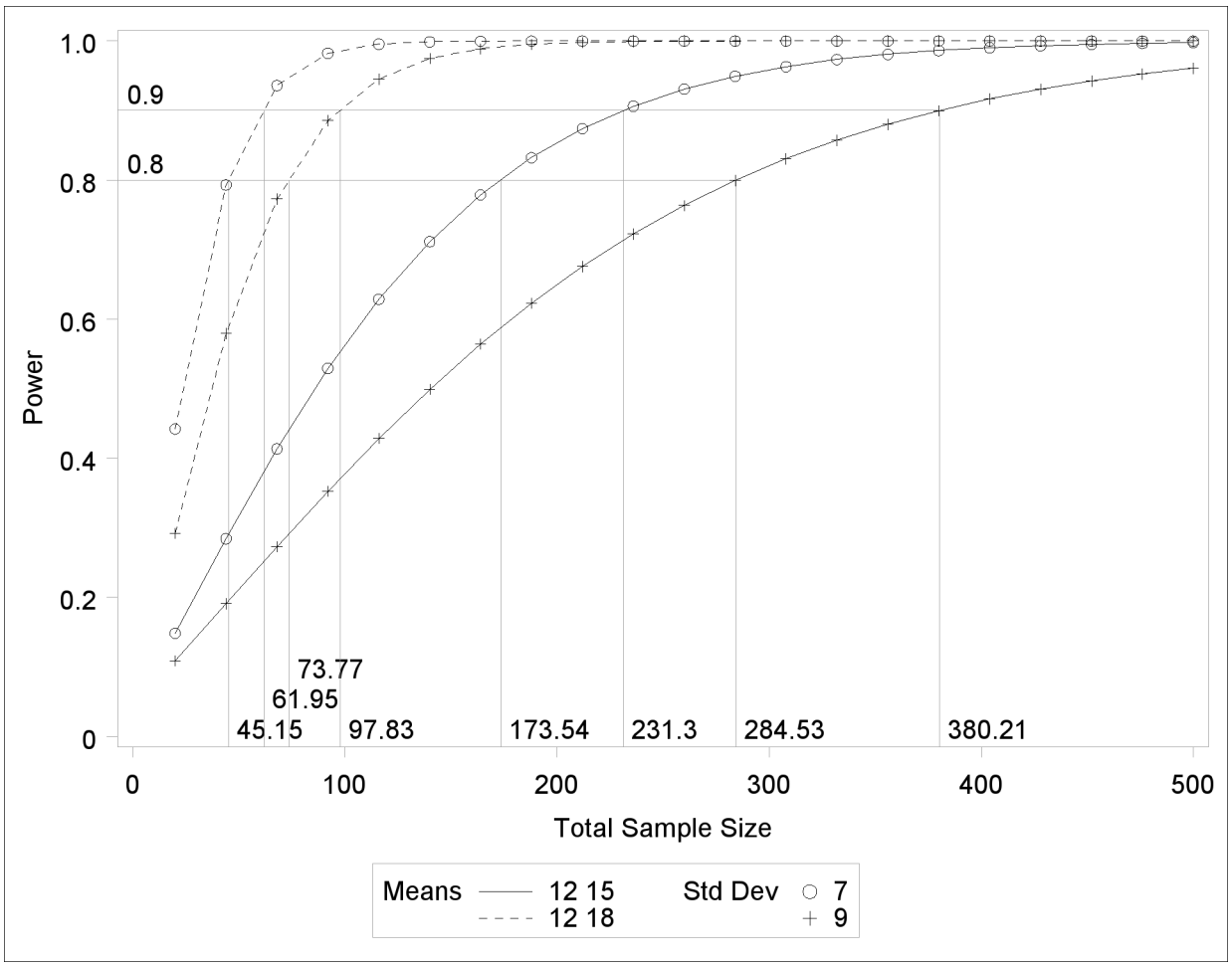


Or you can specify CROSSREF=YES to add reference lines that intersect each curve and cross over to the other axis:

```
plot x=n min=20 max=500
   yopts=(ref=0.8 0.9 crossref=yes);
```

The resulting plot is shown in Output 67.8.12.

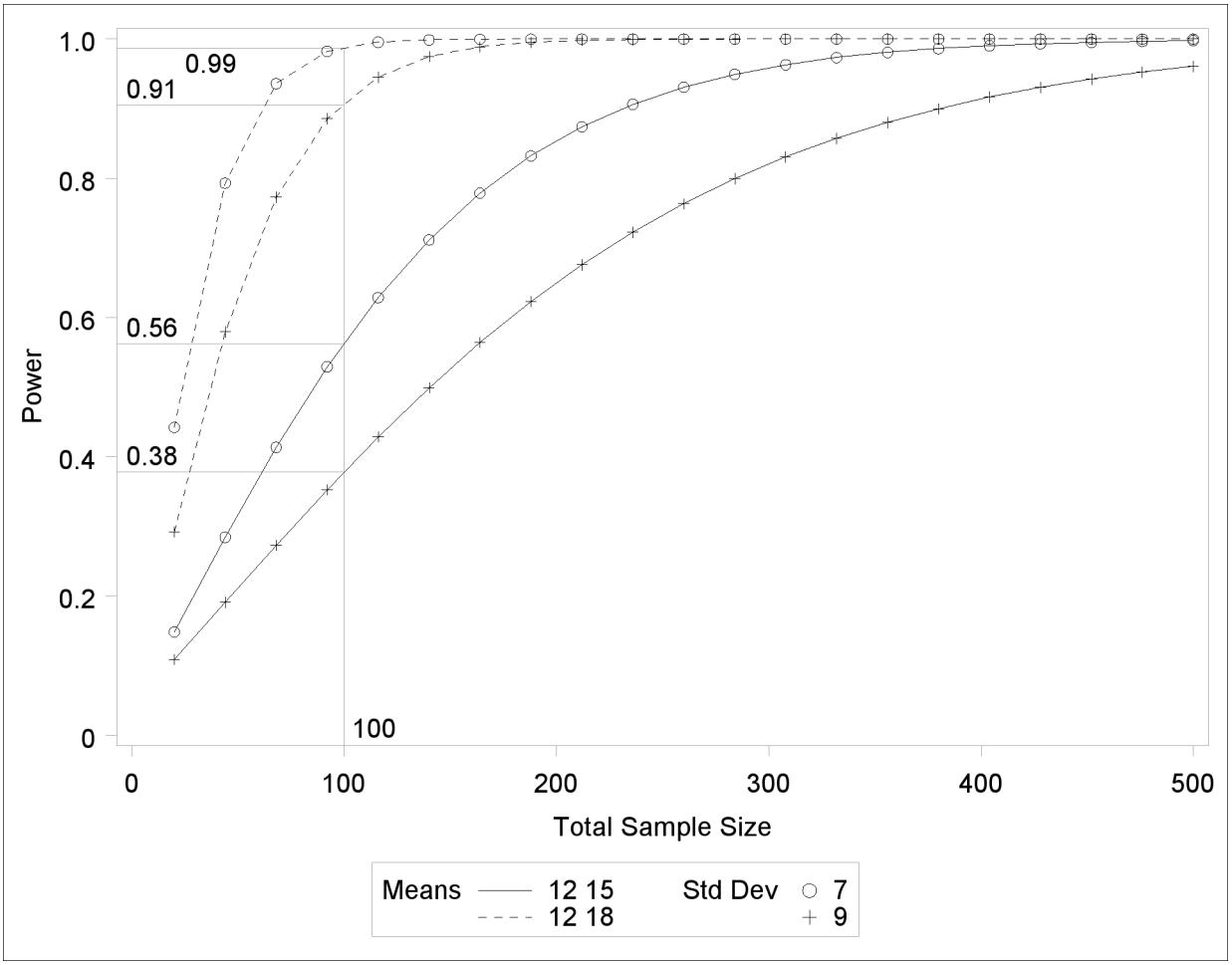
Output 67.8.12 Plot with CROSSREF=YES Style Reference Lines from Y Axis



You can also add reference lines for the X axis by using the XOPTS= option instead of the YOPTS= option. For example, the following PLOT statement produces Output 67.8.13, which has crossing reference lines highlighting the sample size of 100:

```
plot x=n min=20 max=500
   xopts=(ref=100 crossref=yes);
```

Output 67.8.13 Plot with CROSSREF=YES Style Reference Lines from X Axis



Linking Plot Features to Analysis Parameters

You can use the VARY option in the PLOT statement to specify which of the following features you want to associate with analysis parameters.

- line style
- plotting symbol
- color
- panel

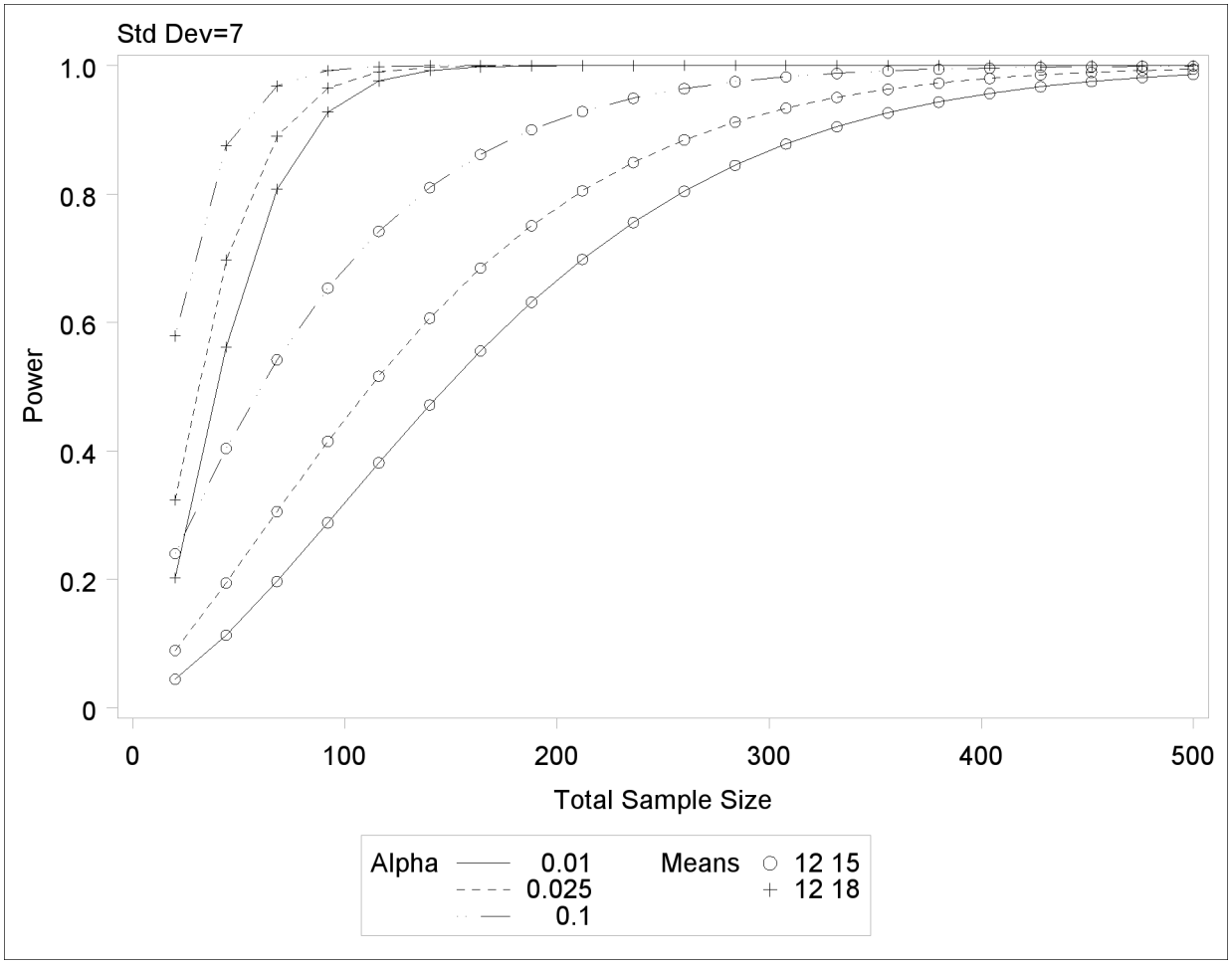
You can specify mappings between each of these features and one or more analysis parameters, or you can simply choose a subset of these features to use (and rely on default settings to associate these features with multiple-valued analysis parameters).

Suppose you supplement the sample size analysis in Output 67.8.5 to include three values of alpha, by using the following statements:

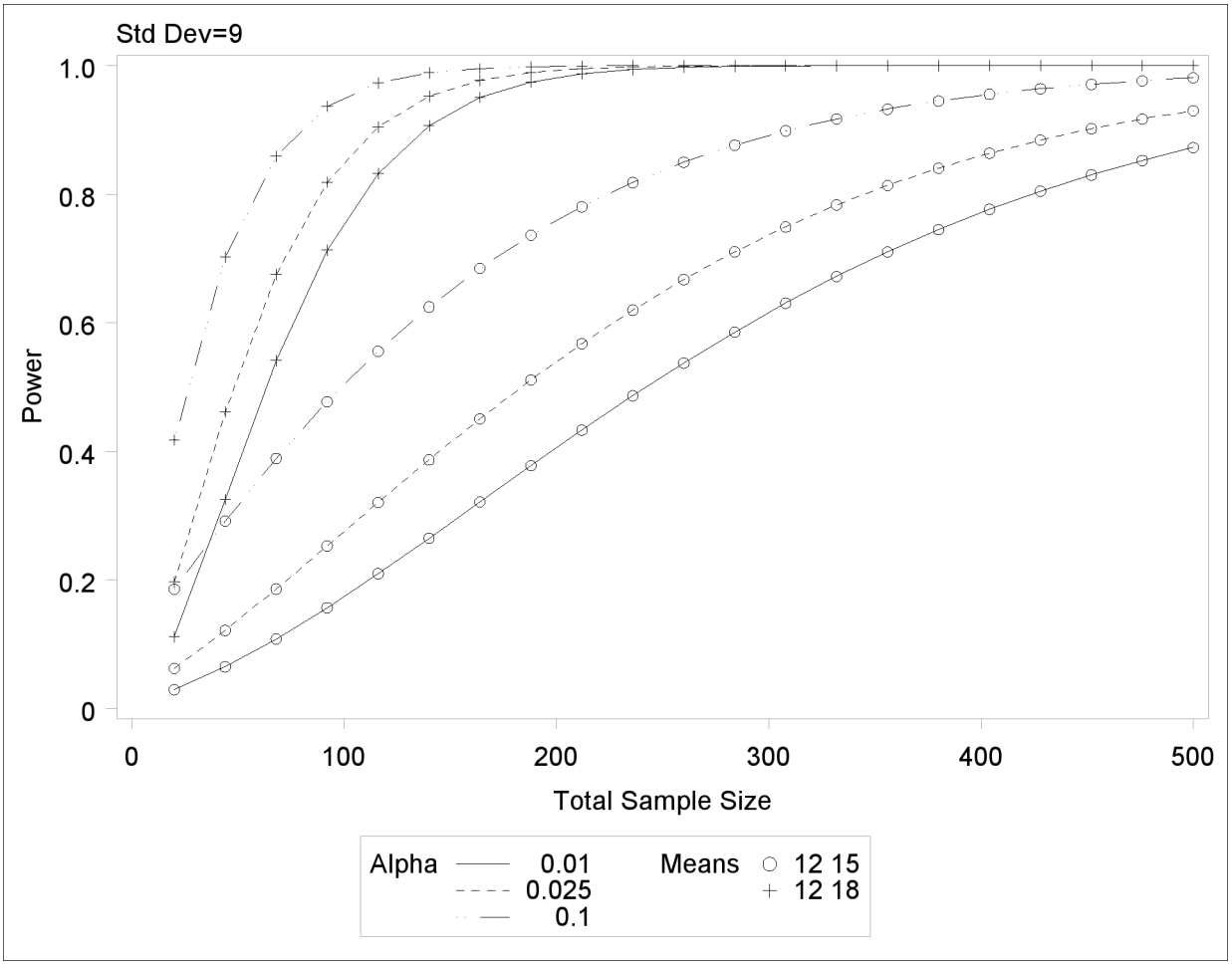
```
proc power plotonly;
   twosamplemeans test=diff
      groupmeans    = 12 | 15 18
      stddev        = 7 9
      alpha         = 0.01 0.025 0.1
      power         = .
      ntotal        = 100;
   plot x=n min=20 max=500;
run;
```

The defaults for the VARY option in the PLOT statement specify line style varying by the ALPHA= parameter, plotting symbol varying by the GROUPMEANS= parameter, panel varying by the STDDEV= parameter, and color remaining constant. The resulting plot, consisting of two panels, is shown in Output 67.8.14.

Output 67.8.14 Plot with Default VARY Settings



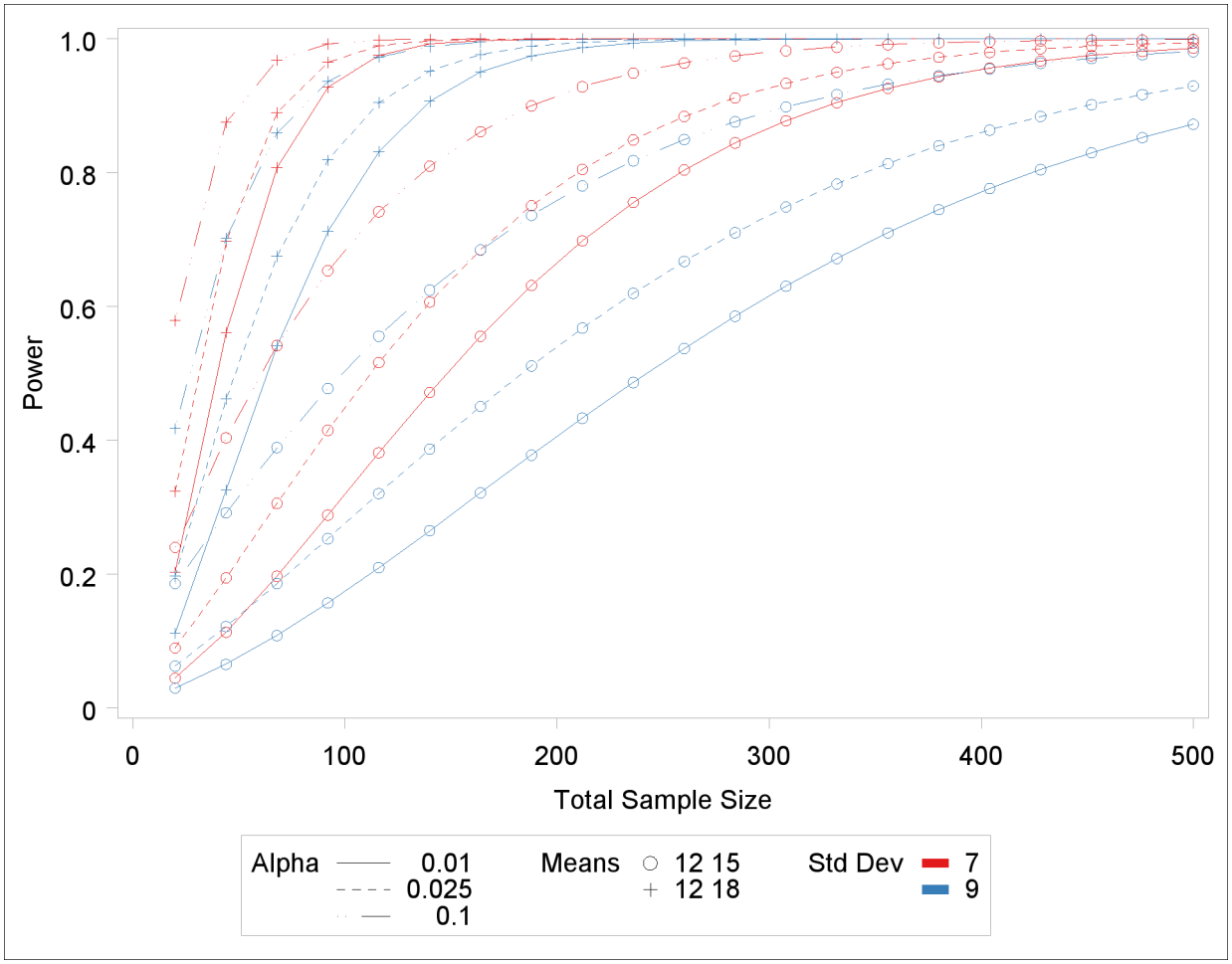
Output 67.8.14 *continued*



Suppose you want to produce a plot with only one panel that varies color in addition to line style and plotting symbol. Include the LINESTYLE, SYMBOL, and COLOR keywords in the VARY option in the PLOT statement, as follows, to produce the plot in Output 67.8.15:

```
plot x=n min=20 max=500
   vary (linestyle, symbol, color);
```

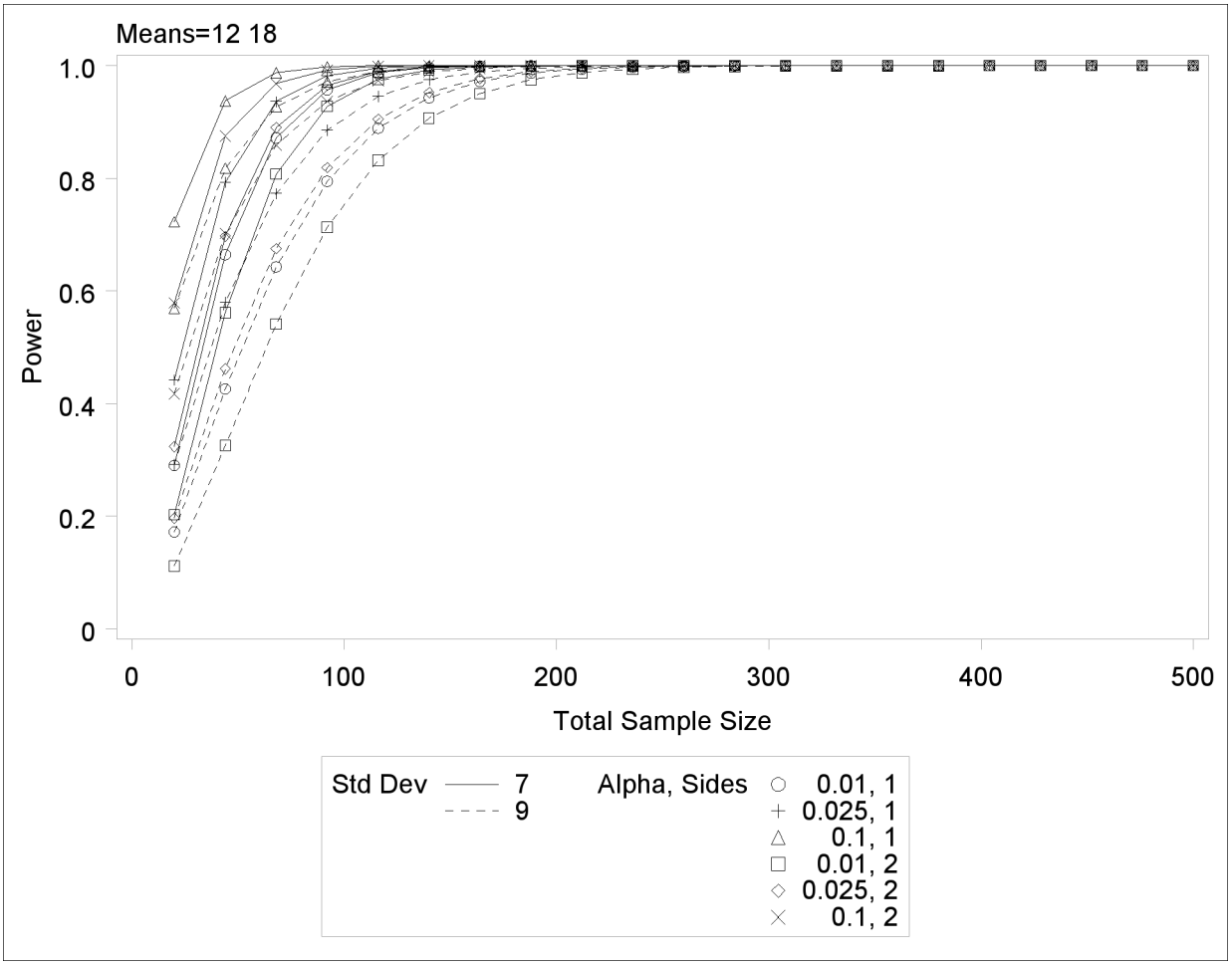
Output 67.8.15 Plot with Varying Color Instead of Panel



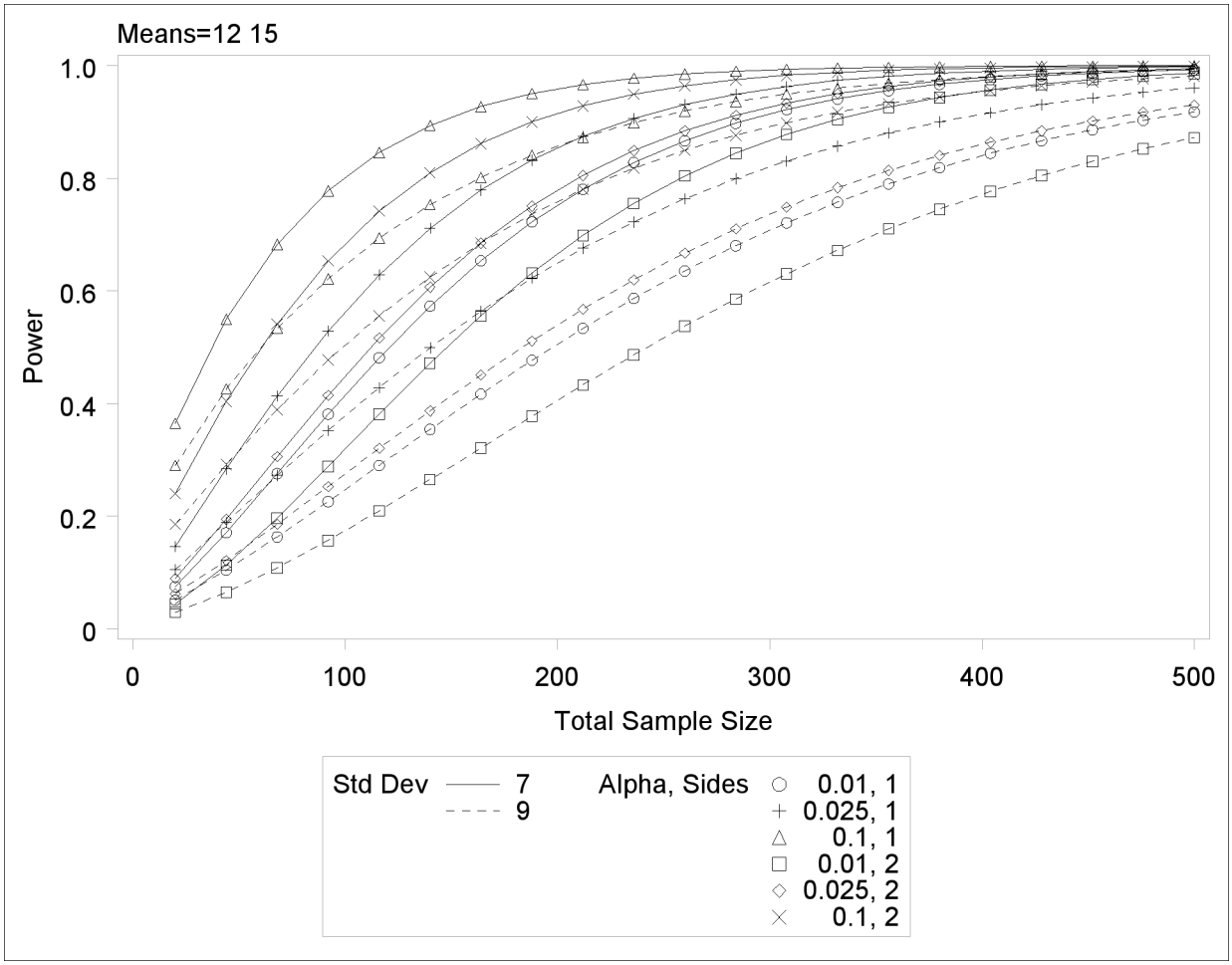
Finally, suppose you want to specify which features are used *and* which analysis parameters they are linked to. The following PLOT statement produces a two-panel plot (shown in Output 67.8.16) in which line style varies by standard deviation, plotting symbol varies by both alpha and sides, and panel varies by means:

```
plot x=n min=20 max=500
   vary (linestyle by stddev,
         symbol by alpha sides,
         panel by groupmeans);
```

Output 67.8.16 Plot with Features Explicitly Linked to Parameters



Output 67.8.16 *continued*



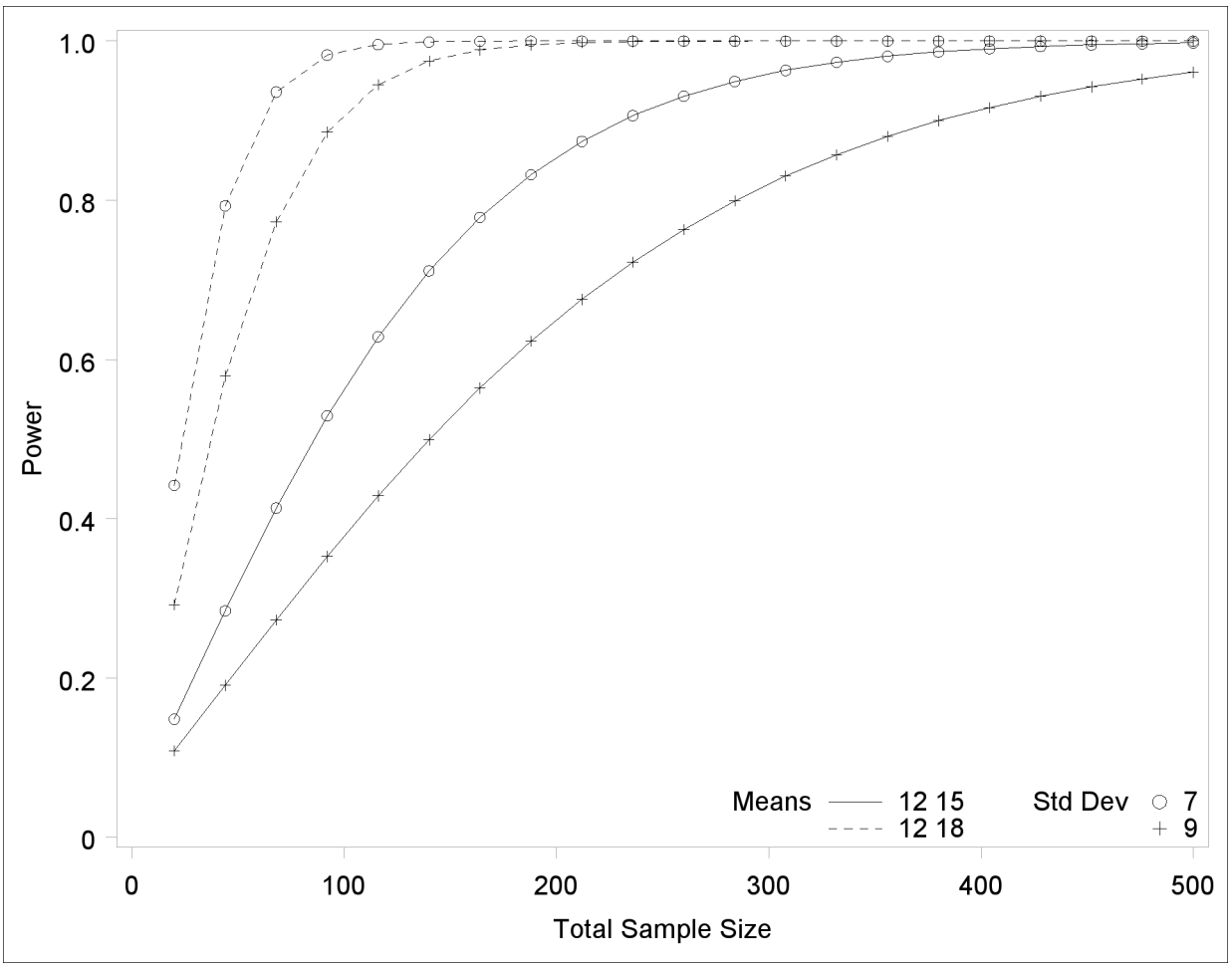
Choosing Key (Legend) Styles

The default style for the key (or "legend") is one that displays the association between levels of features and levels of analysis parameters, located below the X axis. For example, Output 67.8.5 demonstrates this style of key.

You can reproduce Output 67.8.5 with the same key but a different location, inside the plotting region, by using the POS=INSET option within the KEY=BYFEATURE option in the PLOT statement. The following statements product the plot in Output 67.8.17:

```
proc power plotonly;
   twosamplemeans test=diff
      groupmeans   = 12 | 15 18
      stddev       = 7 9
      power        = .
      ntotal       = 200;
   plot x=n min=20 max=500
      key = byfeature(pos=inset);
run;
```

Output 67.8.17 Plot with a By-Feature Key inside the Plotting Region

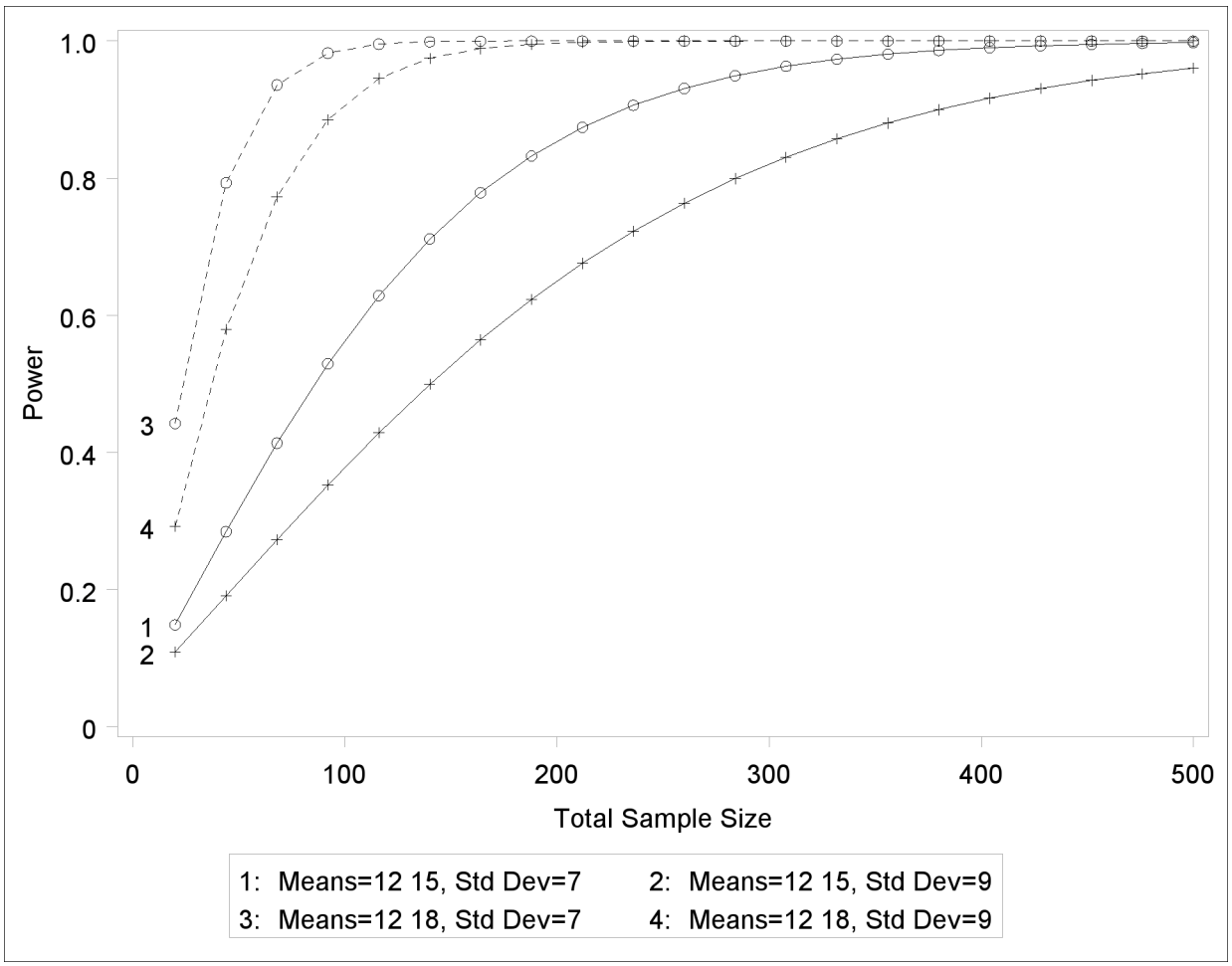


Alternatively, you can specify a key that identifies each individual curve separately by number by using the KEY=BYCURVE option in the PLOT statement:

```
plot x=n min=20 max=500
   key = bycurve;
```

The resulting plot is shown in Output 67.8.18.

Output 67.8.18 Plot with a Numbered By-Curve Key

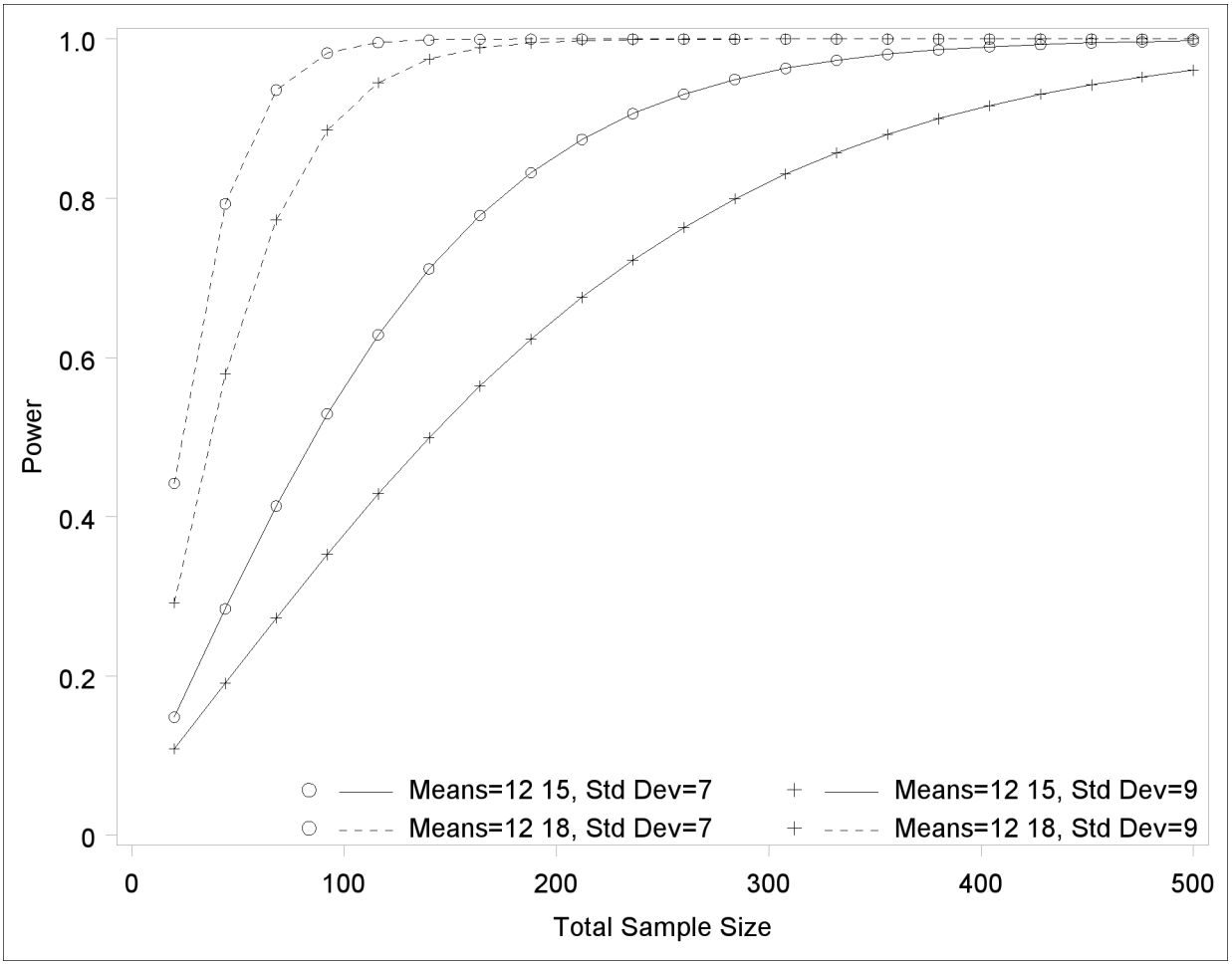


Use the NUMBERS=OFF option within the KEY=BYCURVE option to specify a nonnumbered key that identifies curves with samples of line styles, symbols, and colors:

```
plot x=n min=20 max=500
   key = bycurve(numbers=off pos=inset);
```

The POS=INSET suboption places the key within the plotting region. The resulting plot is shown in Output 67.8.19.

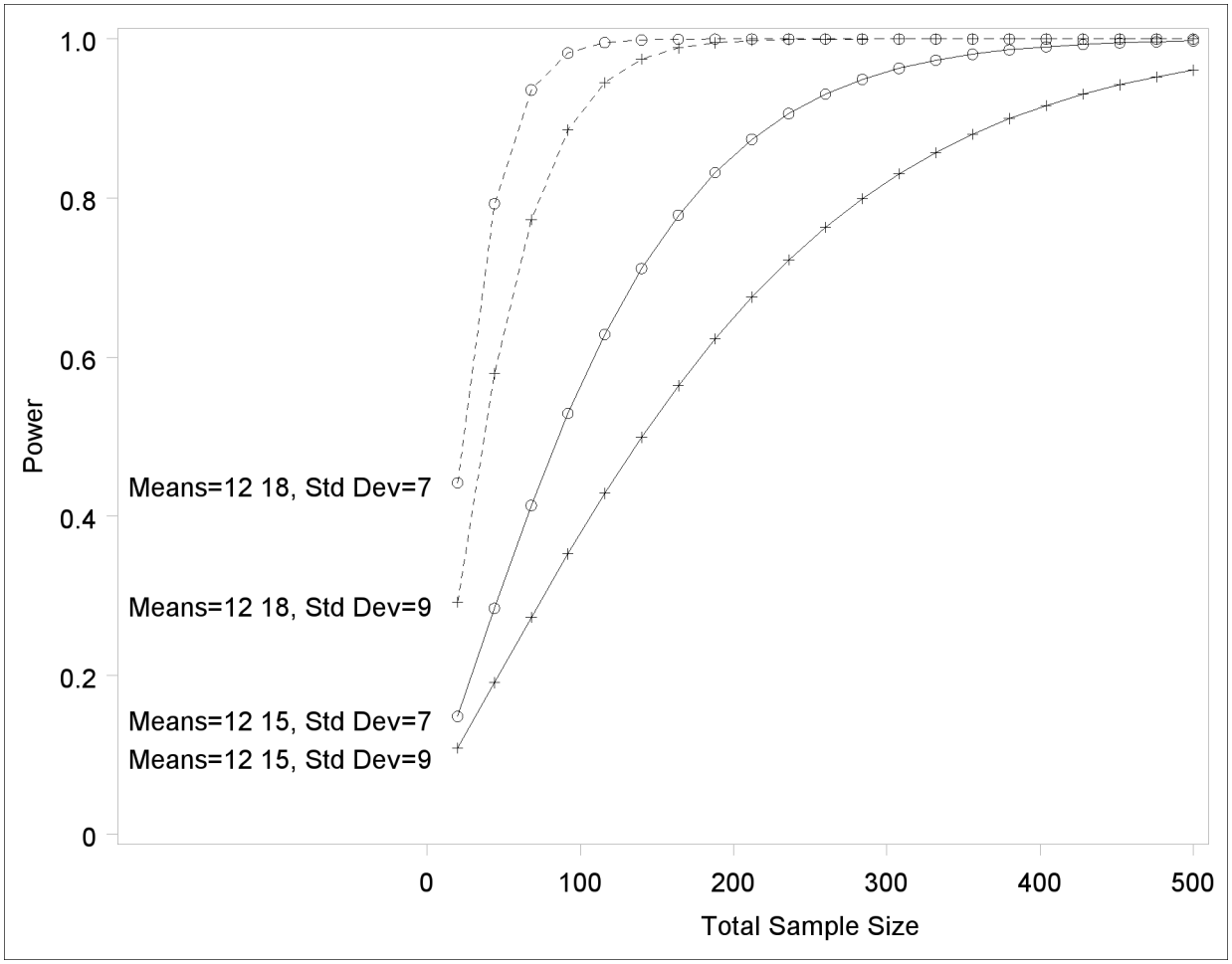
Output 67.8.19 Plot with a Nonnumbered By-Curve Key



Finally, you can attach labels directly to curves with the KEY=ONCURVES option. The following PLOT statement produces Output 67.8.20:

```
plot x=n min=20 max=500
   key = oncurves;
```

Output 67.8.20 Plot with Directly Labeled Curves



Modifying Symbol Locations

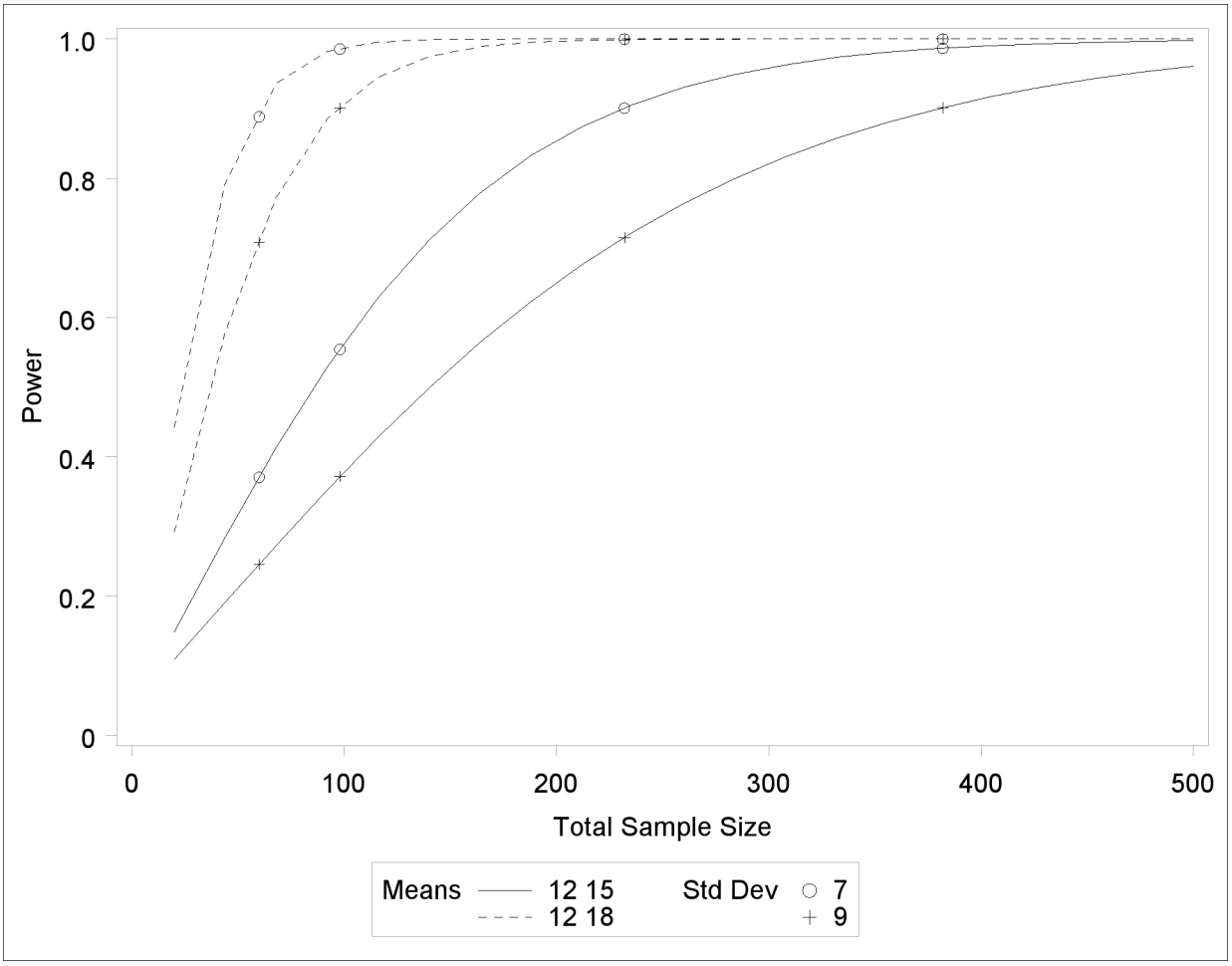
The default locations for plotting symbols are the points computed directly from the power and sample size algorithms. For example, Output 67.8.5 shows plotting symbols corresponding to computed points. The curves connecting these points are interpolated (as indicated by the INTERPOL= option in the PLOT statement).

You can modify the locations of plotting symbols by using the MARKERS= option in the PLOT statement. The MARKERS=ANALYSIS option places plotting symbols at locations corresponding to the input specified in the analysis statement preceding the PLOT statement. You might prefer this as an alternative to using reference lines to highlight specific points. For example, you can reproduce Output 67.8.5, but with the plotting symbols located at the sample sizes shown in Output 67.8.1, by using the following statements:

```
proc power plotonly;
   twosamplemeans test=diff
      groupmeans   = 12 | 15 18
      stddev       = 7 9
      power        = .
      ntotal       = 232 382 60 98;
   plot x=n min=20 max=500
      markers=analysis;
run;
```

The analysis statement here is the TWOSAMPLEMEANS statement. The MARKERS=ANALYSIS option in the PLOT statement causes the plotting symbols to occur at sample sizes specified by the NTOTAL= option in the TWOSAMPLEMEANS statement: 232, 382, 60, and 98. The resulting plot is shown in Output 67.8.21.

Output 67.8.21 Plot with MARKERS=ANALYSIS

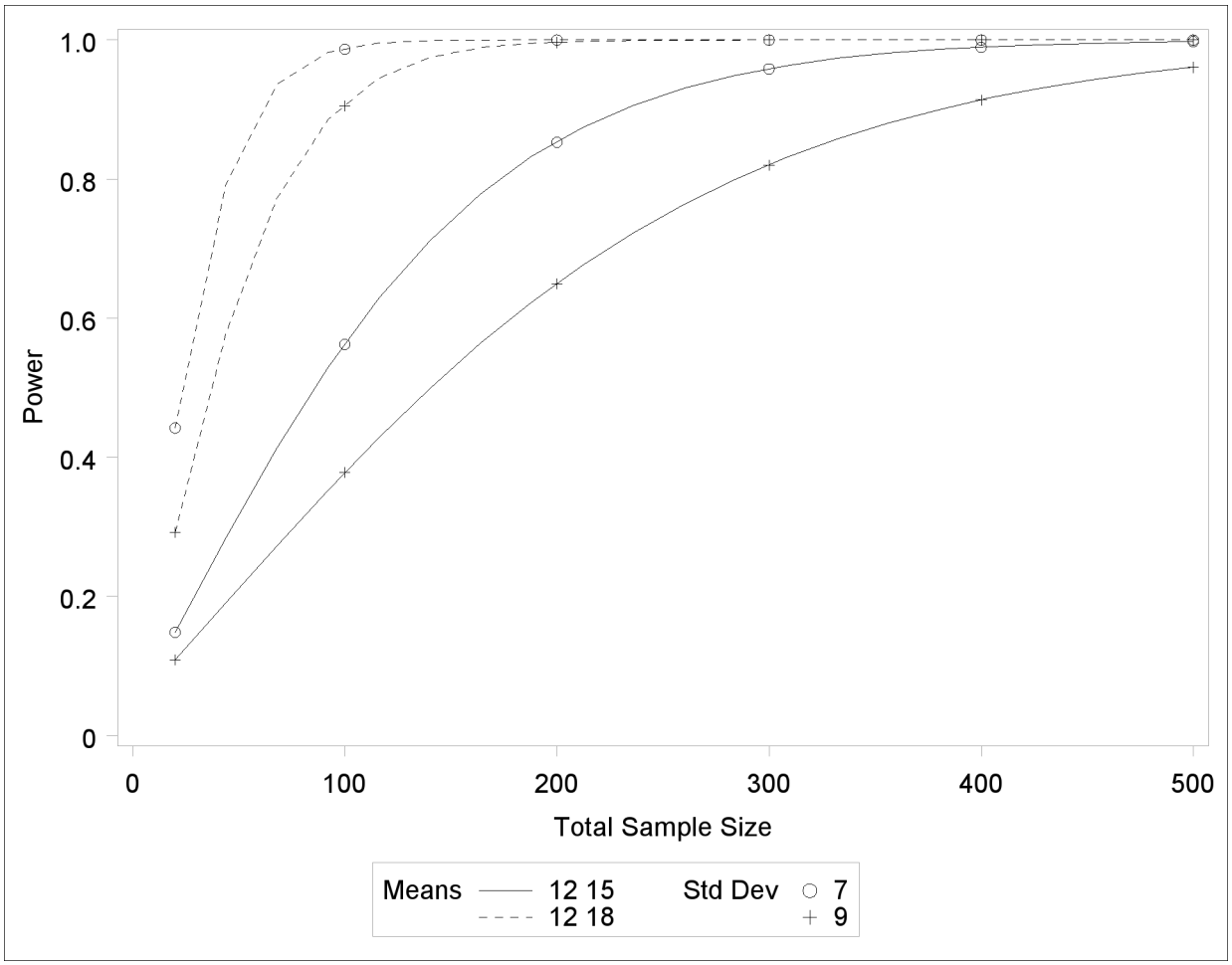


You can also use the MARKERS=NICE option to align symbols with the tick marks on one of the axes (the X axis when the X= option is used, or the Y axis when the Y= option is used):

```
plot x=n min=20 max=500
   markers=nice;
```

The plot created by this PLOT statement is shown in Output 67.8.22.

Output 67.8.22 Plot with MARKERS=NICE



Note that the plotting symbols are aligned with the tick marks on the X axis because the X= option is specified.

Example 67.9: Binary Logistic Regression with Independent Predictors

Suppose you are planning an industrial experiment similar to the analysis in "Getting Started: LOGISTIC Procedure" on page 3260 of Chapter 51, "The LOGISTIC Procedure," but for a different type of ingot. The primary test of interest is the likelihood ratio chi-square test of the effect of heating time on the readiness of the ingots for rolling. Ingots will be randomized independently into one of four different heating times (5, 10, 15, and 20 minutes) with allocation ratios 2:3:3:2 and three different soaking times (2, 4, and 6 minutes) with allocation ratios 2:2:1. The mass of each ingot will be measured as a covariate.

You want to know how many ingots you must sample to have a 90% chance of detecting an odds

ratio as small as 1.2 for a five-minute heating time increase. The odds ratio is defined here as the odds of the ingot not being ready given a heating time of h minutes divided by the odds given a heating time of $h - 5$ minutes, for any time h. You will use a significance level of $\alpha = 0.1$ to balance Type I and Type II errors since you consider their importance to be roughly equal.

The distributions of heating time and soaking time are determined by the design, but you must conjecture the distribution of ingot mass. Suppose you expect its distribution to be approximately normal with mean 4 kg and standard deviation between 1 kg and 2 kg.

You are powering the study for an odds ratio of 1.2 for the heating time, but you must also conjecture odds ratios for soaking time and mass. You suspect that the odds ratio for a unit increase in soaking time is about 1.4, and the odds ratio for a unit increase in mass is between 1 and 1.3.

Finally, you must provide a guess for the average probability of an ingot not being ready for rolling, averaged across all possible design profiles. Existing data suggest that this probability lies between 0.15 and 0.25.

You decide to evaluate sample size at the two extremes of each parameter for which you conjectured a range. Use the following statements to perform the sample size determination:

```
proc power;
   logistic
      vardist("Heat") = ordinal((5 10 15 20) : (0.2 0.3 0.3 0.2))
      vardist("Soak") = ordinal((2 4 6) : (0.4 0.4 0.2))
      vardist("Mass1") = normal(4, 1)
      vardist("Mass2") = normal(4, 2)
      testpredictor = "Heat"
      covariates = "Soak" | "Mass1" "Mass2"
      responseprob = 0.15 0.25
      testoddsratio = 1.2
      units= ("Heat" = 5)
      covoddsratios = 1.4 | 1 1.3
      alpha = 0.1
      power = 0.9
      ntotal = .;
run;
```

The VARDIST= option is used to define the distributions of the predictor variables. The distributions of heating and soaking times are defined by the experimental design, with ordinal probabilities derived from the allocation ratios. The two conjectured standard deviations for the ingot mass are represented in the Mass1 and Mass2 distributions. The TESTPREDICTOR= option identifies the predictor being tested, and the COVARIATES= option specifies the scenarios for the remaining predictors in the model (soaking time and mass). The RESPONSEPROB= option specifies the overall response probability, and the TESTODDSRATIO= and UNITS= options indicate the odds ratio and increment for heating time. The COVODDSRATIOS= option specifies the scenarios for the odds ratios of soaking time and mass. The default DEFAULTUNIT=1 option specifies a unit change for both of these odds ratios. The ALPHA= option sets the significance level, and the POWER= option defines the target power. Finally, the NTOTAL= option with a missing value (.) identifies the parameter to solve for.

Output 67.9.1 shows the results.

Output 67.9.1 Sample Sizes for Test of Heating Time in Logistic Regression

The POWER Procedure
Likelihood Ratio Chi-Square Test for One Predictor

Fixed Scenario Elements

Method	Shieh-O'Brien approximation
Alpha	0.1
Test Predictor	Heat
Odds Ratio for Test Predictor	1.2
Unit for Test Pred Odds Ratio	5
Nominal Power	0.9

Computed N Total

Index	Response Prob	-Covariates-		--Cov ORs-		-Cov Units-		Total N Bins	Actual Power	N Total
1	0.15	Soak	Mass1	1.4	1.0	1	1	120	0.900	1878
2	0.15	Soak	Mass1	1.4	1.3	1	1	120	0.900	1872
3	0.15	Soak	Mass2	1.4	1.0	1	1	120	0.900	1878
4	0.15	Soak	Mass2	1.4	1.3	1	1	120	0.900	1857
5	0.25	Soak	Mass1	1.4	1.0	1	1	120	0.900	1342
6	0.25	Soak	Mass1	1.4	1.3	1	1	120	0.900	1348
7	0.25	Soak	Mass2	1.4	1.0	1	1	120	0.900	1342
8	0.25	Soak	Mass2	1.4	1.3	1	1	120	0.900	1369

The required sample size ranges from 1342 to 1878, depending on the unknown true values of the overall response probability, mass standard deviation, and soaking time odds ratio. The overall response probability clearly has the largest influence among these parameters, with a sample size increase of almost 40% going from 0.25 to 0.15.

Example 67.10: Wilcoxon-Mann-Whitney Test

Consider a hypothetical clinical trial to treat interstitial cystitis (IC), a painful, chronic inflammatory condition of the bladder with no known cause that most commonly affects women. Two treatments will be compared: lidocaine alone ("lidocaine") versus lidocaine plus a fictitious experimental drug called Mironel ("Mir+lido"). The design is balanced, randomized, double-blind, and female-only. The primary outcome is a measure of overall improvement at week 4 of the study, measured on a seven-point Likert scale as shown in Table 67.33.

Table 67.33 Self-Report Improvement Scale

"Compared to when I started this study, my condition is:"	
much worse	–3
worse	–2
slightly worse	–1
the same	0
slightly better	+1
better	+2
much better	+3

The planned data analysis is a one-sided Wilcoxon-Mann-Whitney test with $\alpha = 0.05$ where the alternative hypothesis represents greater improvement for "Mir+lido."

You are asked to graphically assess the power of the planned trial for sample sizes between 100 and 250, assuming that the conditional outcome probabilities given treatment are equal to the values in Table 67.34.

Table 67.34 Conjectured Conditional Probabilities

	Response						
Treatment	–3	–2	–1	0	+1	+2	+3
lidocaine	0.01	0.04	0.20	0.50	0.20	0.04	0.01
Mir+lido	0.01	0.03	0.15	0.35	0.30	0.10	0.06

Use the following statements to compute the power at sample sizes of 100 and 250 and generate a power curve:

```
proc power;
   twosamplewilcoxon
      vardist("lidocaine") = ordinal ((-3 -2 -1 0 1 2 3) :
                                      (.01 .04 .20 .50 .20 .04 .01))
      vardist("Mir+lido") = ordinal ((-3 -2 -1 0 1 2 3) :
                                     (.01 .03 .15 .35 .30 .10 .06))
      variables = "lidocaine" | "Mir+lido"
      sides = u
      ntotal = 100 250
      power = .;
   plot step=10;
run;
```

The VARDIST= option is used to define the distribution for each treatment, and the VARIABLES= option specifies the distributions to compare. The SIDES=U option corresponds to the alternative hypothesis that the second distribution ("Mir+lido") is more favorable. The NTOTAL= option specifies the total sample sizes of interest, and the POWER= option with a missing value (.) identifies the parameter to solve for. The default GROUPWEIGHTS= and ALPHA= options specify a balanced design and significance level $\alpha = 0.05$.

The STEP=10 option in the PLOT statement requests a point for each sample size increment of 10. The default values for the X=, MIN=, and MAX= plot options specify a sample size range of 100 to 250 (the same as in the analysis) for the X axis.

The tabular and graphical results are shown in Output 67.10.1 and Output 67.10.2, respectively.

Output 67.10.1 Power Values for Wilcoxon-Mann-Whitney Test

The POWER Procedure
Wilcoxon-Mann-Whitney Test

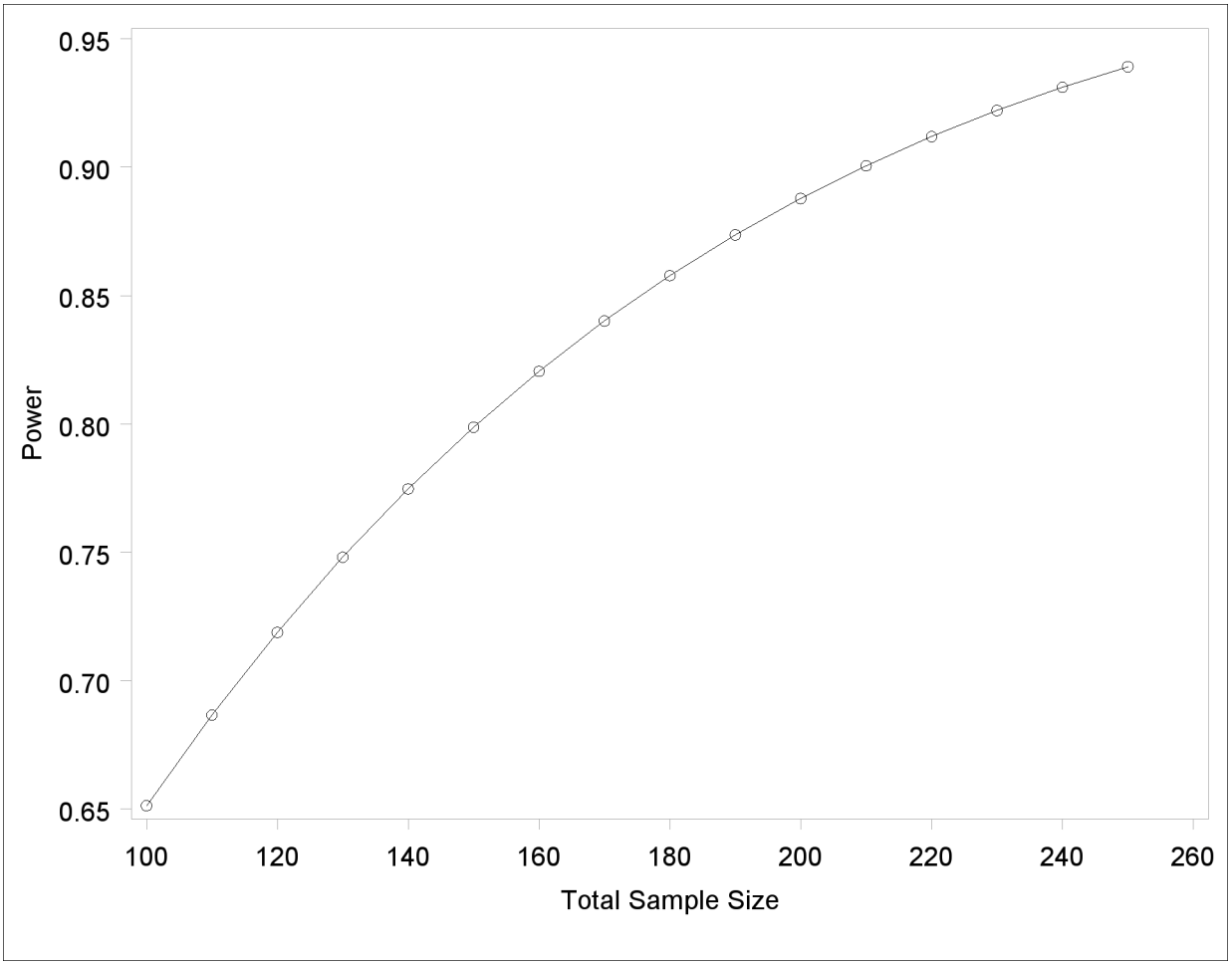
Fixed Scenario Elements

Method	O'Brien-Castelloe approximation
Number of Sides	U
Group 1 Variable	lidocaine
Group 2 Variable	Mir+lido
Pooled Number of Bins	7
Alpha	0.05
Group 1 Weight	1
Group 2 Weight	1
NBins Per Group	1000

Computed Power

Index	N Total	Power
1	100	0.651
2	250	0.939

Output 67.10.2 Plot of Power versus Sample Size for Wilcoxon Power Analysis



The achieved power ranges from 0.651 to 0.939, increasing with sample size.

References

Agresti, A. (1980), "Generalized Odds Ratios for Ordinal Data," *Biometrics*, 36, 59–67.

Agresti, A. and Coull, B.A. (1998), "Approximate Is Better Than "Exact" for Interval Estimation of Binomial Proportions," *The American Statistician*, 52, 119–126.

Anderson, T. W. (1984), *An Introduction to Multivariate Statistical Analysis*, Second Edition, New York: John Wiley & Sons.

Beal, S. L. (1989), "Sample Size Determination for Confidence Intervals on the Population Means and on the Difference between Two Population Means," *Biometrics*, 45, 969–977.

Blackwelder, W. C. (1982), "'Proving the Null Hypothesis' in Clinical Trials," *Controlled Clinical Trials*, 3, 345–353.

Brown, L. D., Cai, T. T. and DasGupta, A. (2001), "Interval Estimation for a Binomial Proportion," *Statistical Science*, 16, 101–133.

Cantor, A. B. (1997), *Extending SAS Survival Analysis Techniques for Medical Research*, Cary, NC: SAS Institute Inc.

Castelloe, J. M. (2000), "Sample Size Computations and Power Analysis with the SAS System," *Proceedings of the Twenty-fifth Annual SAS Users Group International Conference,* Paper 265-25, Cary, NC: SAS Institute Inc.

Castelloe, J. M. and O'Brien, R. G. (2001), "Power and Sample Size Determination for Linear Models," *Proceedings of the Twenty-sixth Annual SAS Users Group International Conference,* Paper 240-26, Cary, NC: SAS Institute Inc.

Chernick, M. R. and Liu, C. Y. (2002), "The Saw-Toothed Behavior of Power Versus Sample Size and Software Solutions: Single Binomial Proportion Using Exact Methods," *The American Statistician*, 56, 149–155.

Chow, S.-C., Shao, J. and Wang, H. (2003), *Sample Size Calculations in Clinical Research*, Boca Raton, FL: CRC Press, Taylor & Francis Group.

Connor, R. J. (1987), "Sample Size for Testing Differences in Proportions for the Paired-Sample Design," *Biometrics*, 43, 207–211.

Diegert, C. and Diegert, K. V. (1981), "Note on Inversion of Casagrande-Pike-Smith Approximate Sample-Size Formula for Fisher-Irwin Test on 2×2 Tables," *Biometrics*, 37, 595.

Diletti, D., Hauschke, D., and Steinijans, V. W. (1991), "Sample Size Determination for Bioequivalence Assessment by Means of Confidence Intervals," *International Journal of Clinical Pharmacology, Therapy and Toxicology*, 29, 1–8.

DiSantostefano, R. L. and Muller, K. E. (1995), "A Comparison of Power Approximations for Satterthwaite's Test," *Communications in Statistics—Simulation and Computation*, 24 (3), 583–593.

Fisher, R. A. (1921), "On the 'Probable Error' of a Coefficient of Correlation Deduced from a Small Sample," *Metron*, 1, 3–32.

Fleiss, J. L., Tytun, A. and Ury, H. K. (1980), "A Simple Approximation for Calculating Sample Sizes for Comparing Independent Proportions," *Biometrics*, 36, 343–346.

Gatsonis, C. and Sampson, A. R. (1989), "Multiple Correlation: Exact Power and Sample Size Calculations," *Psychological Bulletin*, 106, 516–524.

Hocking, R. R. (1985), *The Analysis of Linear Models*, Monterey, CA: Brooks/Cole.

Johnson, N. L. and Kotz, S. (1970), *Distributions in Statistics: Continuous Univariate Distributions — 1*, New York: John Wiley & Sons.

Johnson, N. L., Kotz, S., and Balakrishnan, N. (1994), *Continuous Univariate Distributions, Volume 1*, Second Edition, New York: John Wiley & Sons.

Johnson, N. L., Kotz, S., and Balakrishnan, N. (1995), *Continuous Univariate Distributions, Volume 2*, Second Edition, New York: John Wiley & Sons.

Johnson, N. L., Kotz, S., and Kemp, A. W. (1992), *Univariate Discrete Distributions*, Second Edition, New York: John Wiley & Sons.

Jones R. M. and Miller, K. S. (1966), "On the Multivariate Lognormal Distribution," *Journal of Industrial Mathematics*, 16, 63–76.

Kolassa, J. E. (1995), "A Comparison of Size and Power Calculations for the Wilcoxon Statistic for Ordered Categorical Data," *Statistics in Medicine*, 14, 1577–1581.

Lachin, J. M. (1992), "Power and Sample Size Evaluation for the McNemar Test with Application to Matched Case-Control Studies," *Statistics in Medicine*, 11, 1239–1251.

Lakatos, E. (1988), "Sample Sizes Based on the Log-Rank Statistic in Complex Clinical Trials," *Biometrics*, 44, 229–241.

Lenth, R. V. (2001), "Some Practical Guidelines for Effective Sample Size Determination," *The American Statistician*, 55, 187–193.

Maxwell, S. E. (2000), "Sample Size and Multiple Regression Analysis," *Psychological Methods*, 5, 434–458.

Miettinen, O. S. (1968), "The Matched Pairs Design in the Case of All-or-None Responses," *Biometrics*, 339–352.

Moser, B. K., Stevens, G. R., and Watts, C. L. (1989), "The Two-Sample T Test versus Satterthwaite's Approximate F Test," *Communications in Statistics A — Theory and Methods*, 18, 3963–3975.

Muller, K. E. and Benignus, V. A. (1992), "Increasing Scientific Power with Statistical Power," *Neurotoxicology and Teratology*, 14, 211–219.

O'Brien, R. G. and Castelloe, J. M. (2006), "Exploiting the Link between the Wilcoxon-Mann-Whitney Test and a Simple Odds Statistic," *Proceedings of the Thirty-first Annual SAS Users Group International Conference,* Paper 209-31, Cary, NC: SAS Institute Inc.

O'Brien, R. G. and Castelloe, J. (2007), "Sample-Size Analysis for Traditional Hypothesis Testing: Concepts and Issues," in *Pharmaceutical Statistics Using SAS: A Practical Guide*, ed. A. Dmitrienko, C. Chuang-Stein, and R. D'Agostino, Cary, NC: SAS Institute Inc., Chapter 10, 237–271.

O'Brien, R. G. and Muller, K. E. (1993), "Unified Power Analysis for *t*-Tests through Multivariate Hypotheses," in *Applied Analysis of Variance in Behavioral Science*, ed. L. K. Edwards, New York: Marcel Dekker, 297–344.

Owen, D. B. (1965), "A Special Case of a Bivariate Non-central *t*-Distribution," *Biometrika*, 52,

437–446.

Pagano, M. and Gauvreau, K. (1993), *Principles of Biostatistics*, Belmont, CA: Wadsworth.

Phillips, K. F. (1990), "Power of the Two One-Sided Tests Procedure in Bioequivalence," *Journal of Pharmacokinetics and Biopharmaceutics*, 18, 137–144.

Satterthwaite, F. W. (1946), "An Approximate Distribution of Estimates of Variance Components," *Biometrics Bulletin*, 2, 110–114.

Schork, M. and Williams, G. (1980), "Number of Observations Required for the Comparison of Two Correlated Proportions," *Communications in Statistics—Simulation and Computation 9*, 349–357.

Schuirmann, D. J. (1987), "A Comparison of the Two One-Sided Tests Procedure and the Power Approach for Assessing the Equivalence of Average Bioavailability," *Journal of Pharmacokinetics and Biopharmaceutics*, 15, 657–680.

Self, S. G., Mauritsen, R. H. and Ohara, J. (1992), "Power Calculations for Likelihood Ratio Tests in Generalized Linear Models," *Biometrics*, 48, 31–39.

Senn, S. (1993), *Cross-over Trials in Clinical Research*, New York: John Wiley & Sons.

Shieh, G. (2000), "A Comparison of Two Approaches for Power and Sample Size Calculations in Logistic Regression Models," *Communications in Statistics—Simulation*, 29, 763–791.

Shieh, G. and O'Brien, R. G. (1998), "A Simpler Method to Compute Power for Likelihood Ratio Tests in Generalized Linear Models," paper presented at the Annual Joint Statistical Meetings of the American Statistical Association, Dallas, TX.

Stuart, A. and Ord, J. K. (1994), *Kendall's Advanced Theory of Statistics, Volume 1: Distribution Theory*, Sixth Edition, Baltimore: Edward Arnold Publishers.

Walters, D. E. (1979). "In Defence of the Arc Sine Approximation," *The Statistician*, 28, 219–232.

Wellek, S. (2003), *Testing Statistical Hypotheses of Equivalence*, Boca Raton, FL: Chapman & Hall, CRC Press.

Chapter 68

The Power and Sample Size Application

Contents

Overview: PSS Application **5048**
SAS Power and Sample Size 5048
Getting Started: PSS Application **5050**
Overview 5050
The Basic Steps 5050
A Simple Example 5051
How to Use: PSS Application **5068**
Overview 5068
SAS Connections 5068
Setting Preferences 5072
Creating and Editing PSS Projects 5076
Importing and Exporting Projects 5083
Details: PSS Application **5085**
Software Requirements 5085
Installation 5085
Configuration 5086
Example: Two-Sample *t* Test **5086**
Overview 5086
Test of Two Independent Means for Equal Variances 5086
Test of Two Independent Means for Unequal Variances 5098
Test of Mean Ratios 5101
Additional Topics 5108
Example: Analysis of Variance **5111**
Overview 5111
The Example 5111
Additional Topics 5120
Example: Two-Sample Survival Rank Tests **5127**
Overview 5127
The Example 5127
Additional Topics 5141

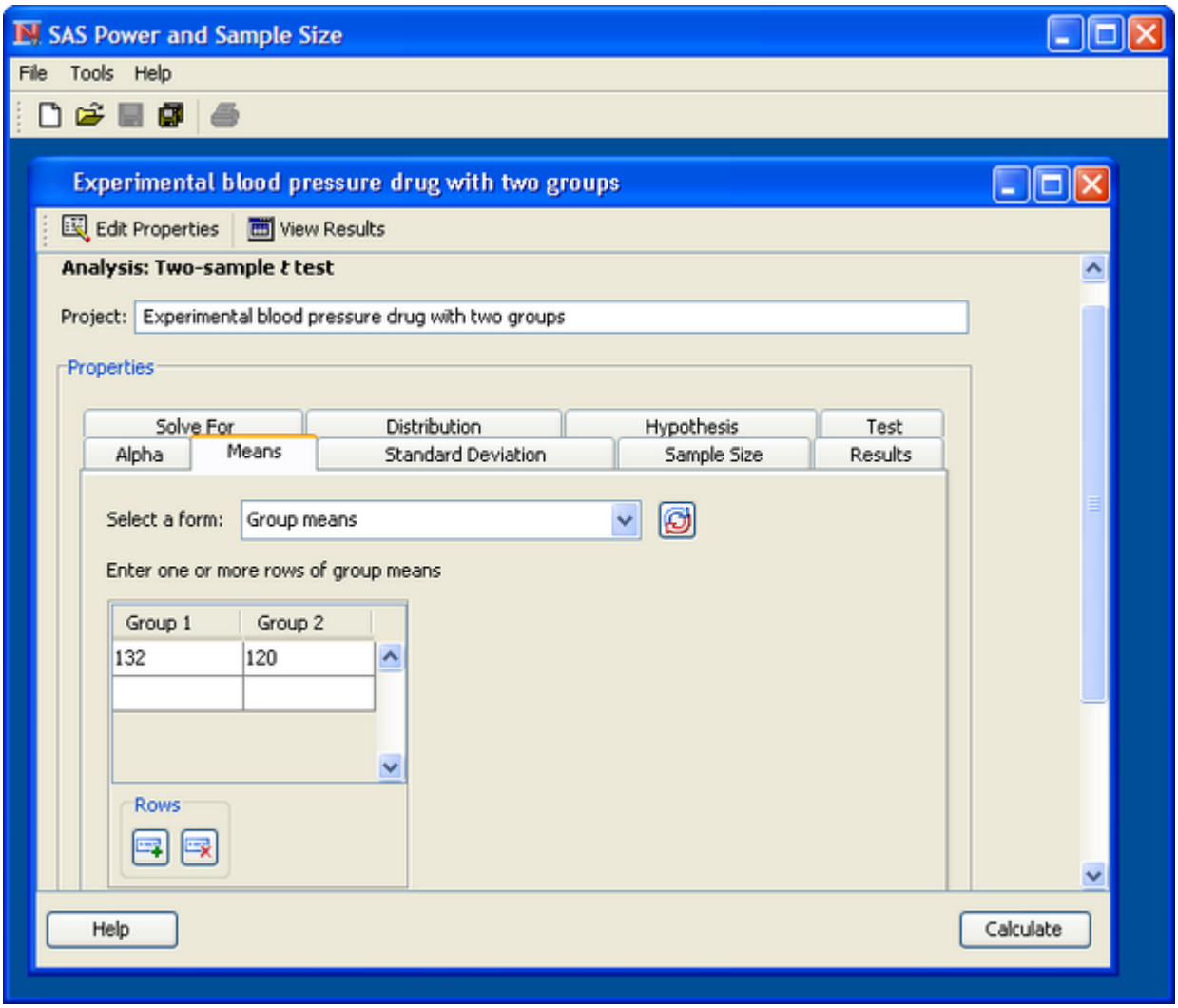
Overview: PSS Application

SAS Power and Sample Size

The SAS Power and Sample Size application (PSS) is a desktop application that provides easy access to power analysis and sample size determination techniques. The application is intended for students and researchers as well as experienced SAS users and statisticians.

Figure 68.1 shows the graphical user interface. PSS relies on the SAS/STAT procedures POWER and GLMPOWER for its computations.

Figure 68.1 PSS Application



This section describes the statistical tasks that are available with the application as well as its principal features.

Analyses

PSS provides power and sample size computations for a variety of statistical analyses. Included are *t* tests for means; equivalence tests and confidence intervals for means and proportions; exact binomial, chi-square, Fisher's exact, and McNemar tests for proportions; correlation and regression (multiple and logistic); one-way analysis of variance; linear models; tests of distribution; and rank tests for comparing survival curves.

Table 68.1 lists the analyses that are available.

Table 68.1 Available Analyses

Category	Analysis
Means	One-sample *t* test
	Paired *t* test
	Two-sample *t* test
Confidence intervals	One proportion
	One-sample means
	Paired means
	Two-sample means
Equivalence tests	One proportion
	One-sample means
	Paired means
	Two-sample means
Proportions	One proportion
	Two correlated proportions
	Two independent proportions
Correlation and regression	Pearson correlation coefficient
	Logistic regression with a binary response
	Multiple regression
Analysis of variance and linear models	One-way ANOVA
	General linear univariate models
Survival analysis	Two-sample survival rank tests
Distribution tests	Wilcoxon Mann-Whitney test for two distributions

Features

PSS provides multiple input parameter options, stores the results in a project format, displays power curves, and produces narratives for the results. Narratives are descriptions of the input parameters and include a statement about the computed power or sample size. The SAS log and SAS code are also available.

All analyses offer computation of power or sample size. Some analyses offer computation of sample size per group as well as total sample size.

Where appropriate, several alternate ways of entering values for certain parameters are offered. For example, in the two-sample *t* test analysis, means can be entered for individual groups or as a difference. The null mean difference can be specified as a default of zero or can be explicitly entered.

Information about existing analyses is stored in a project format. You can access each project to review the results or to edit your input parameters and produce another analysis.

Getting Started: PSS Application

Overview

This section is intended to get you off to a quick start with PSS. More detailed information about using the application is found in "How to Use: PSS Application" on page 5068 and in the example sections.

To start the application on a PC using the Windows operating system, select **Start►Programs►SAS►SAS Power and Sample Size 3.1** (or the latest release).

When you first use the application for a release, you are asked some configuration questions. For more information see the section "Configuration" on page 5086.

As an inital step, you also must define a SAS connection. If you have Foundation SAS software installed on the PC that you are using for PSS, this step can be done for you automatically. To define a connection or to determine whether one has already been defined, see the section "SAS Connections" on page 5068.

The Basic Steps

Here are the basic steps that you follow to use PSS.

1. Start a new project by selecting **File►New** on the menu bar or clicking the **New** icon on the toolbar.
2. In the New window, select the desired analysis type and click **OK**.

   A project window for the analysis type appears with the Edit Properties page displayed. (The tabs on the Edit Properties page and their content vary according to the analysis type.)
3. Click each tab to enter the relevant data for the analysis. (For more information about the types of data to enter, see the example sections.)

4. After you have entered all the data, click the **Calculate** button.
5. After PSS calculates the results, the project window displays the View Results page with the Summary Table tab displayed by default.
6. To view other results or to review the SAS code or the SAS log, click any of the tabs on the left side of the View Results page.
7. To print any results page, select **File►Print** on the menu bar.

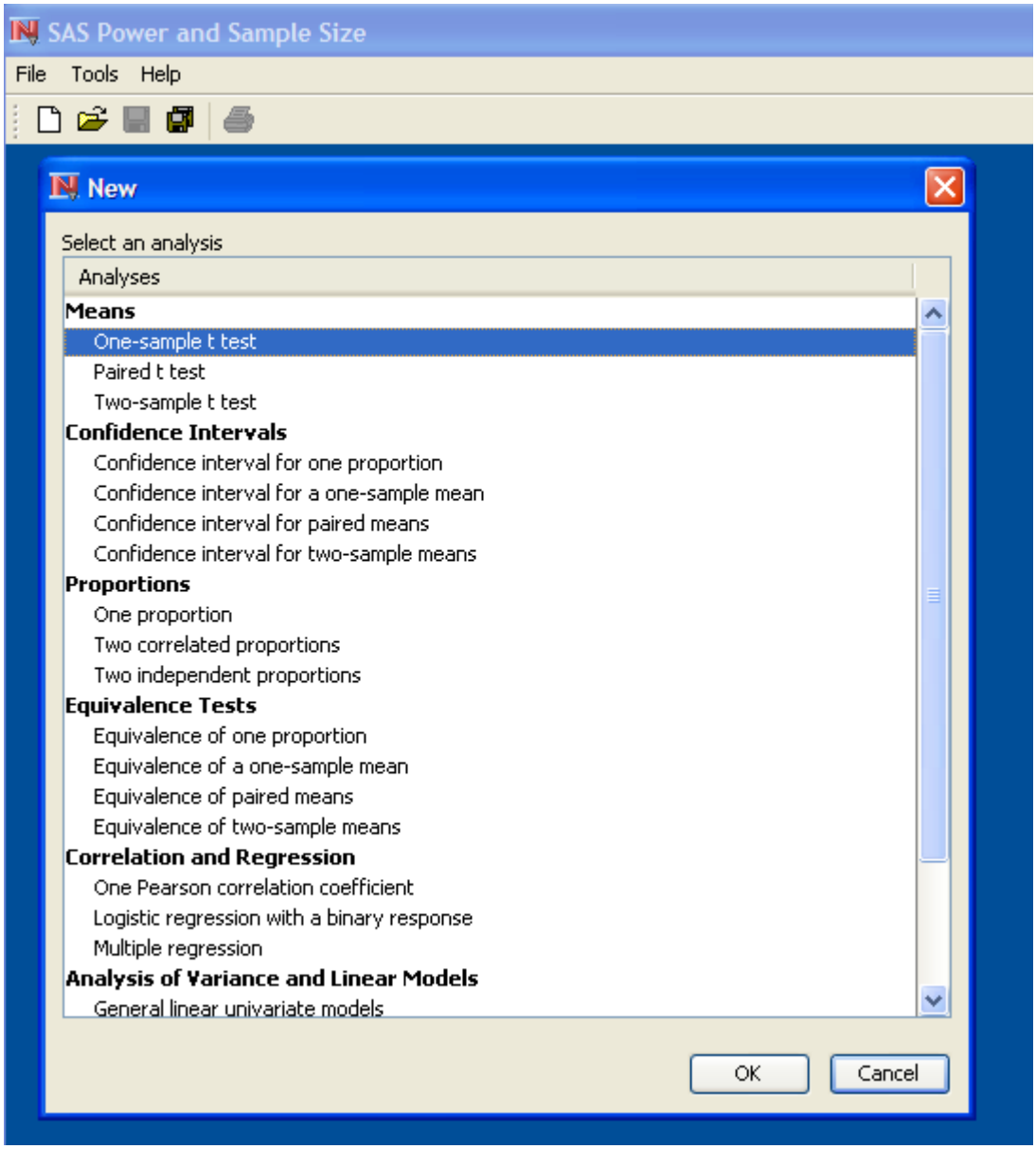
The remainder of this section takes you through a simple example.

A Simple Example

Suppose you want to determine the power for a new marketing study. You want to compare car sales in the southeastern region to the national average of 1.0 car per salesperson per day. You believe that the actual average for the region is 1.6 cars per salesperson per day. You want to test if the mean for a single group is larger than a specific value, so the one-sample *t* test is the appropriate analysis. The conjectured mean is 1.6 and the null mean is 1.0. You intend to use a significance level of 0.05 for the one-sided test. You want to calculate power for two standard deviations, 0.5 and 0.75, and two sample sizes, 10 and 20 dealerships.

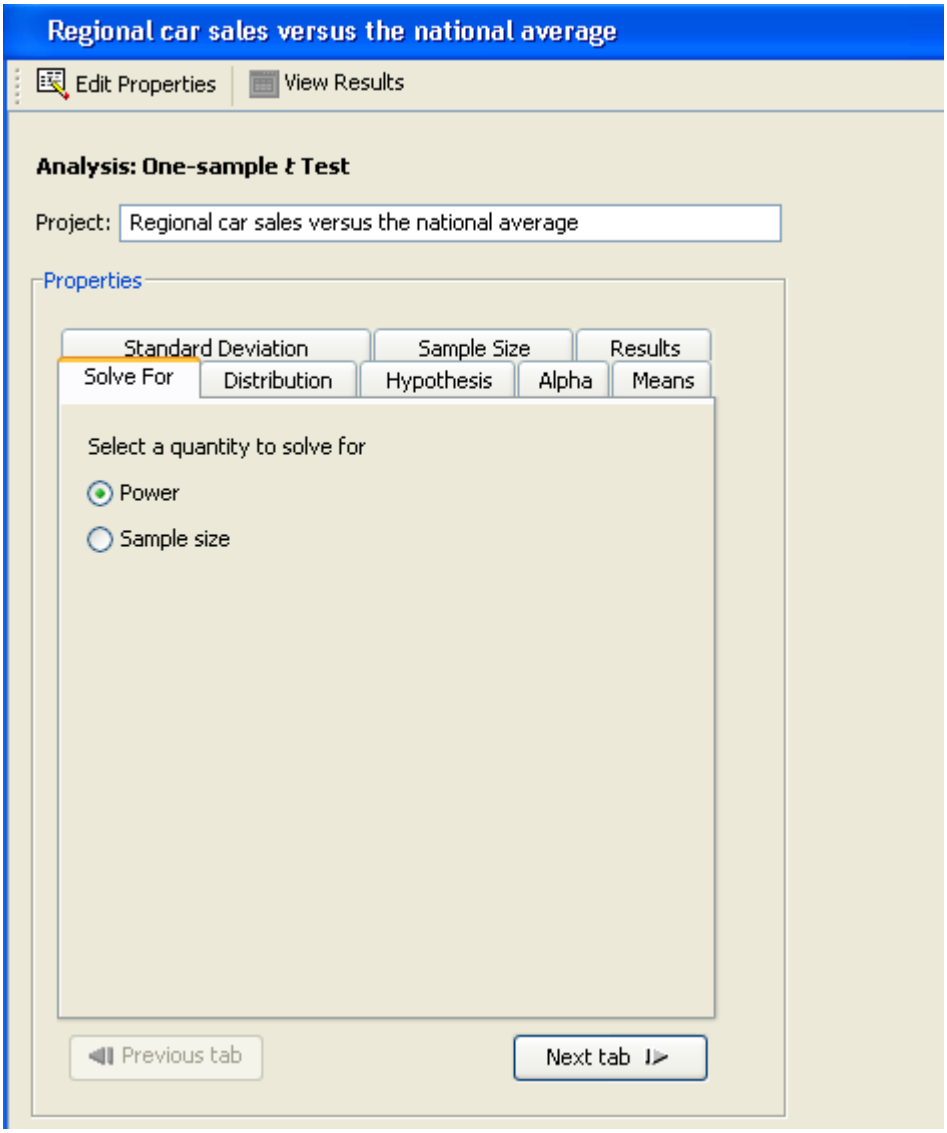
First, open a new project by selecting **File►New** on the menu bar or clicking the **New** icon on the toolbar. The New window appears. Then, select the appropriate analysis.

Figure 68.2 New Window



For this example, the selected analysis is the **One-sample t test** in the **Means** section, as shown at the top of Figure 68.2. Select the analysis from the list and click **OK**. The **One-sample t test** project window appears with the Edit Properties page displayed, as shown in Figure 68.3.

Figure 68.3 Edit Properties Page



Enter a descriptive label of the project in the **Project:** field. For the example, change the description to `Regional car sales versus the national average`. The description is used to identify the project when you reopen it from the Open window.

Select **File►Save** to save the description change. Note in Figure 68.3 that the title bar of the window contains your project description after you have saved the change.

Properties of the project are displayed on several tabs. You can change from tab to tab by clicking a tab or by clicking the **Next tab** or **Previous tab** buttons. To display help about the properties for a tab, click the **Help** button at the bottom of the Edit Properties page.

Entering Parameter Values

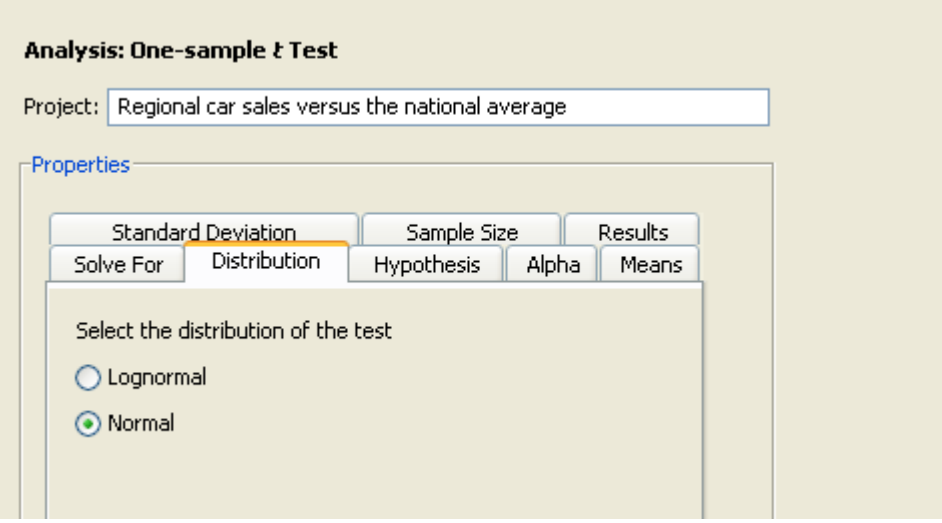
First, click the **Solve For** tab and choose to calculate power or sample size. For this example, select the **Power** option, as shown in Figure 68.3.

Next, you must provide values for two analysis options and four parameters. These parameters are set in separate tabs on the Edit Properties page and are labeled **Distribution**, **Hypothesis**, **Alpha**, **Mean**, **Standard Deviation**, and **Sample Size**.

Distribution

Click the **Distribution** tab to select a **Normal** or **Lognormal** distribution. For the example, you are using means rather than mean ratios, so select **Normal**, as shown in Figure 68.4.

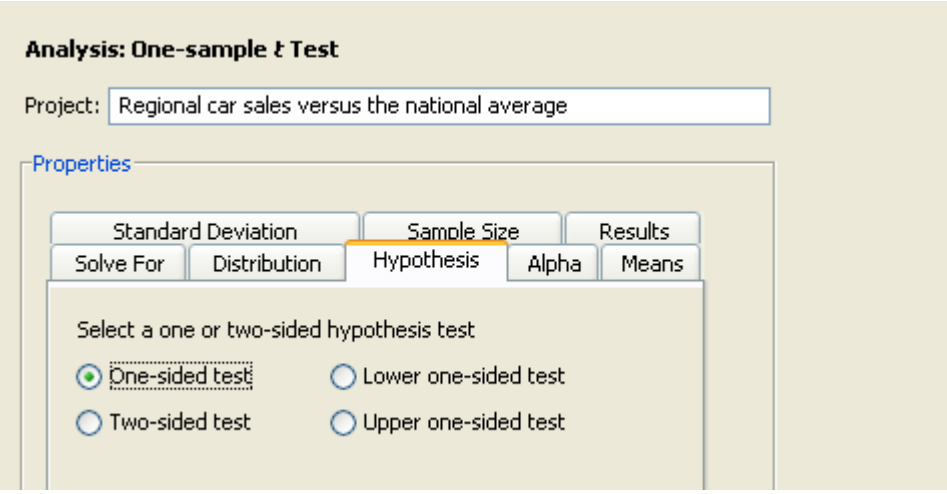
Figure 68.4 Distribution Tab



Hypothesis

Click the **Hypothesis** tab to select a one- or two-sided test. Because you are interested only in whether the southeastern region produces higher daily car sales than the national average, select **One-sided test**, as shown in Figure 68.5.

Figure 68.5 Hypothesis Tab

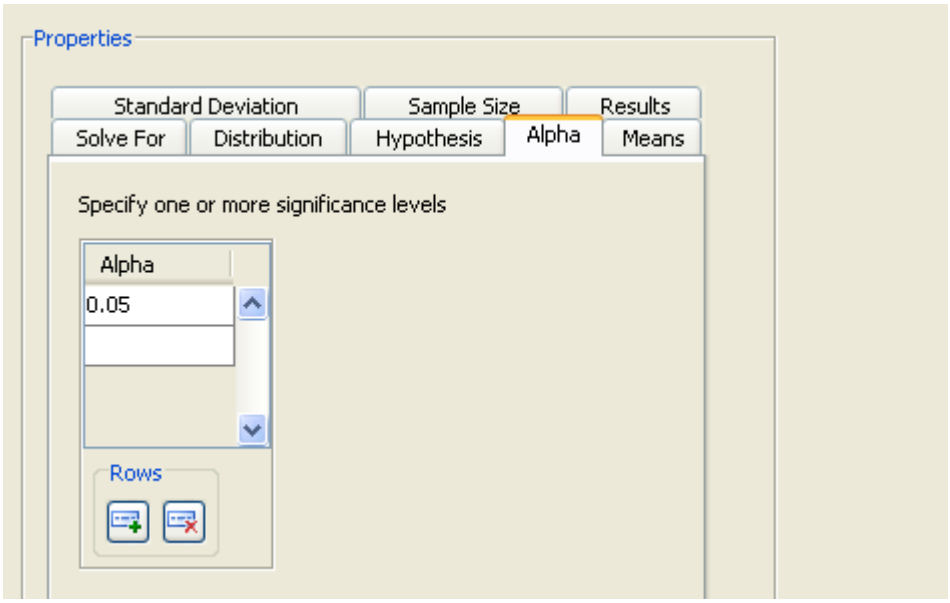


There are three one-sided test options: **One-sided test**, **Upper one-sided test**, and **Lower one-sided test**. The **Upper one-sided test** option would also be appropriate for this example.

Alpha

Click the **Alpha** tab to specify one or more significance levels. Enter `0.05`, as shown in Figure 68.6.

Figure 68.6 Alpha Tab

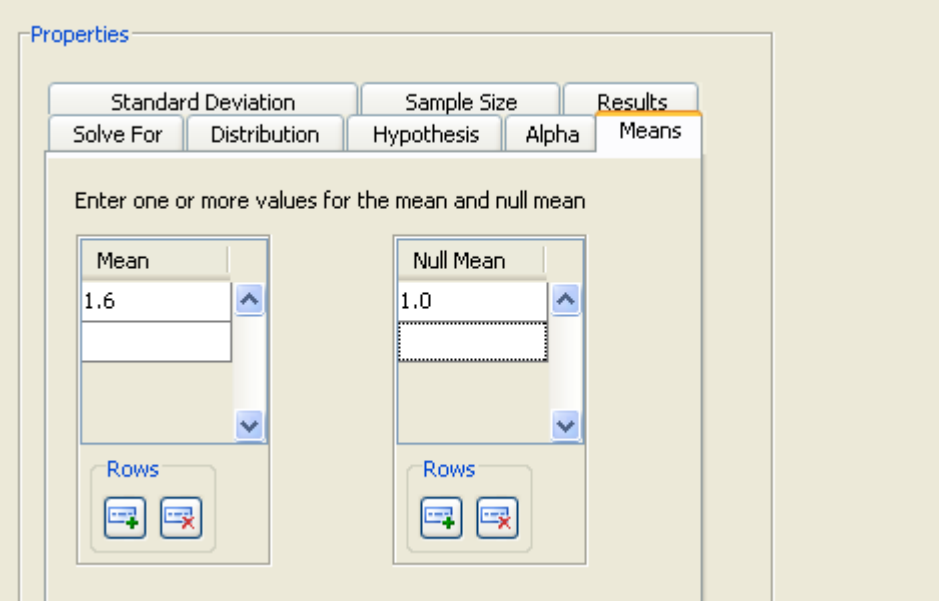


This value will be the default unless the default has been changed in the Preferences window. To set preferences, select **Tools►Preferences** on the menu bar. For more information about setting preferences, see the section "Setting Preferences" on page 5072.

Mean

Click the **Means** tab to enter one or more means and null means. For the example, enter `1.6` in the Mean table and `1.0` in the Null Mean table. Figure 68.7 shows the entered values.

Figure 68.7 Means Tab

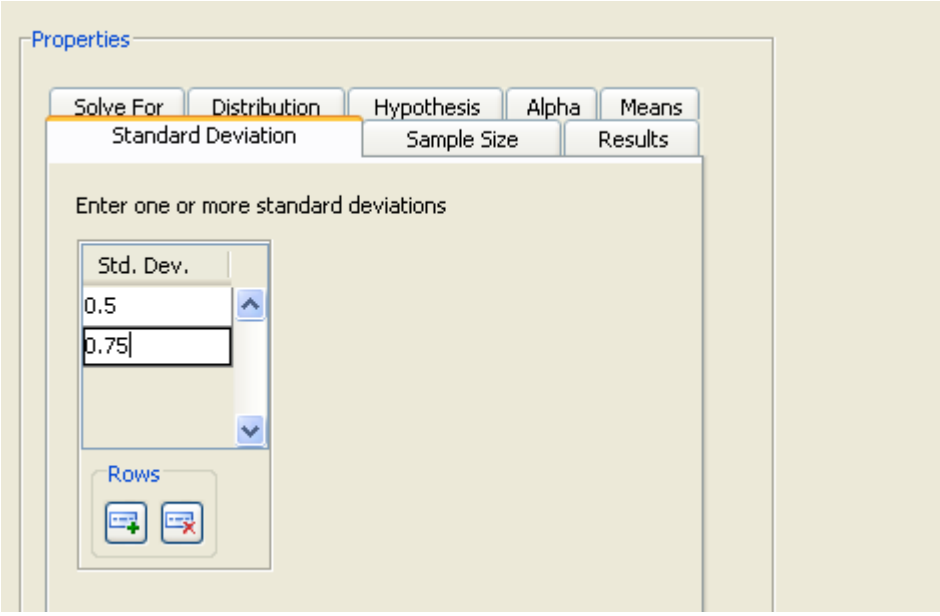


Note that additional input rows are available if you want to enter additional sets of parameters. You can also append and delete rows using the and buttons beneath the table. In addition, by selecting a row and right-clicking, you can choose to insert and delete rows in the body of the table from a pop-up menu.

Standard Deviation

Click the **Standard Deviation** tab to enter standard deviations. You are interested in two standard deviations, `0.5` and `0.75`. Enter them in the table, as shown in Figure 68.8.

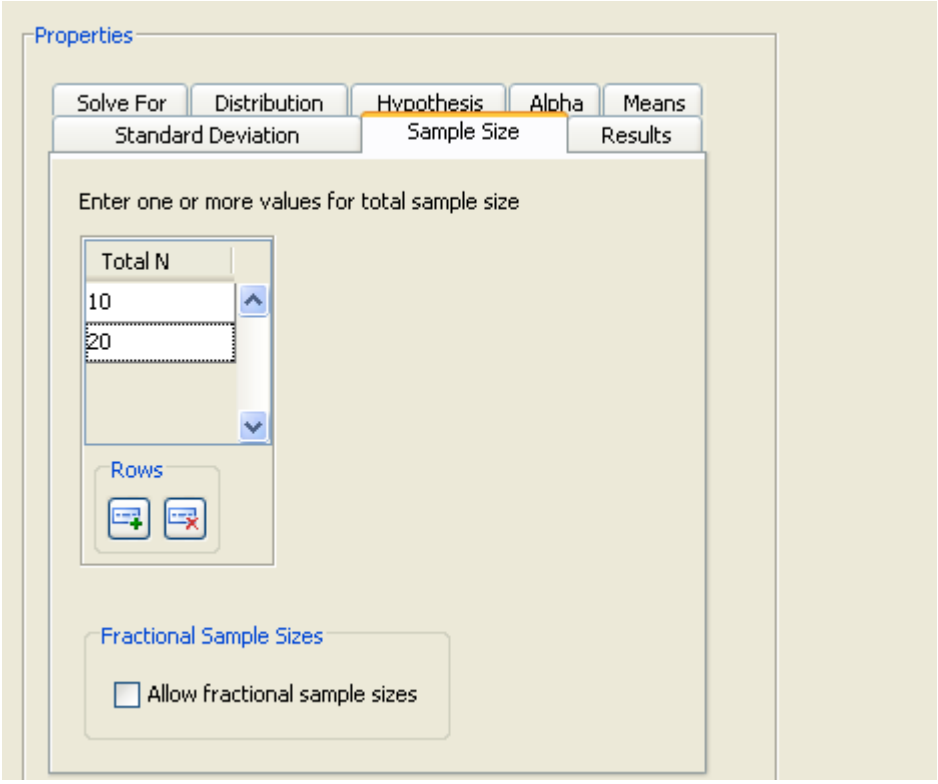
Figure 68.8 Standard Deviation Tab



Sample Size

You want to be able to sample between `10` and `20` dealerships. Click the **Sample Size** tab and enter these two values, as shown in Figure 68.9.

Figure 68.9 Sample Size Tab



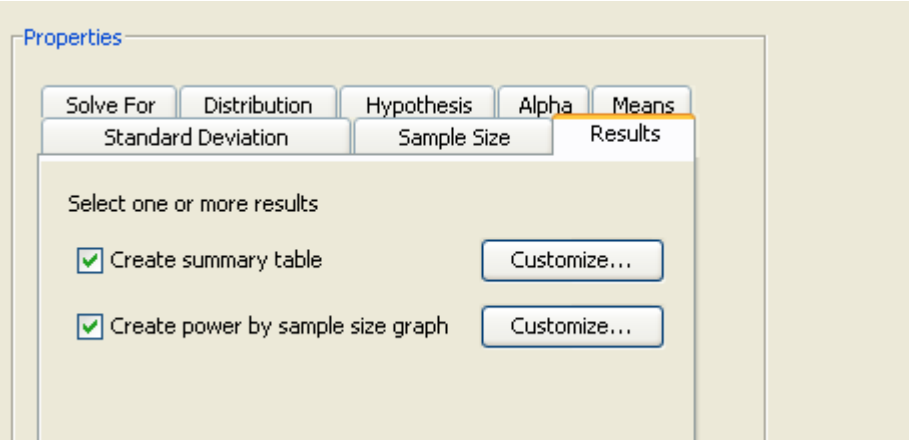
Scenarios

The input values are combined into one or more scenarios. In this case, each of the two standard deviations is combined with each of the two sample sizes for a total of four scenarios. Then power is computed for each scenario. In this example, only a single value or setting is present for the mean, null mean, and alpha level, so they are common to all scenarios.

Results Options

Click the **Results** tab to select results options including a Summary Table and a Power by Sample Size graph.

Figure 68.10 Results Tab

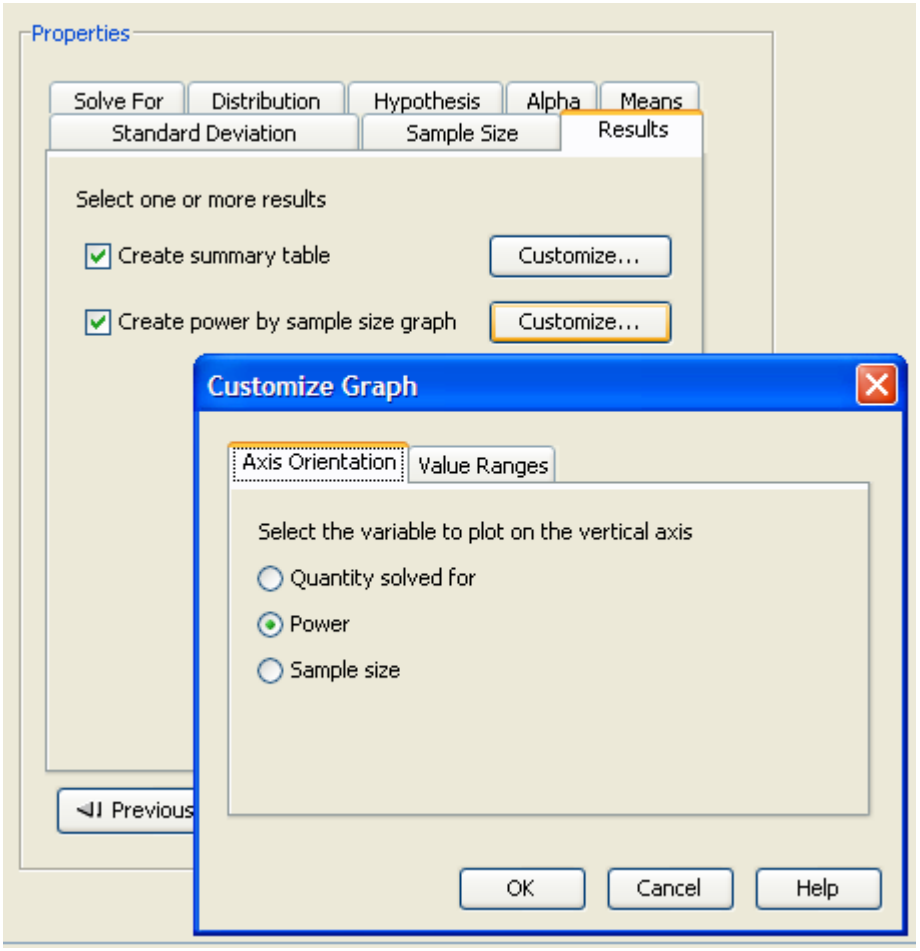


For this example, select both results check boxes: **Create summary table** and **Create power by sample size graph**, as shown in Figure 68.10. These selections can also be set as preferences; see the section "Setting Preferences" on page 5072.

Customizing the Power by Sample Size Graph

Click the **Customize** button beside the **Create power by sample size graph** check box to customize the graph. The Customize Graph window contains two tabs: **Axis Orientation** and **Value Ranges**, as shown in Figure 68.11.

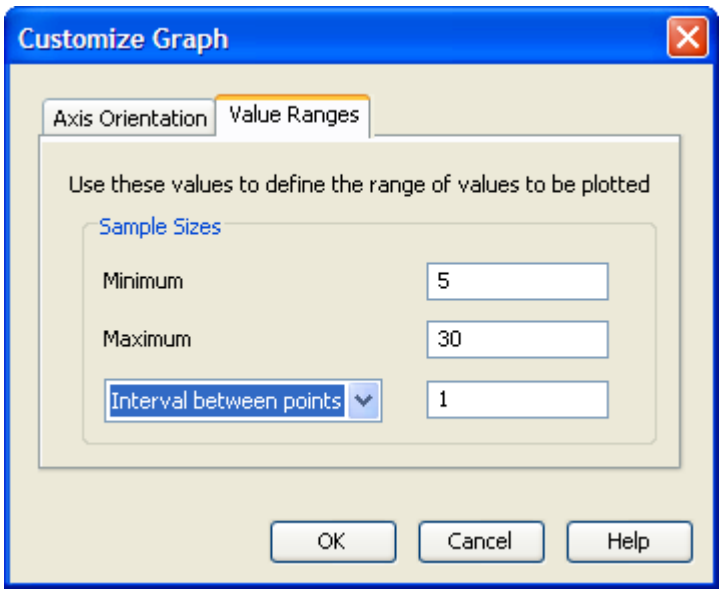
Figure 68.11 Customize Graph Window with Axis Orientation Tab



Click the **Axis Orientation** tab to select which quantity you would like to plot on the vertical axis. You can choose to display the quantity solved for (either power or sample size) on the vertical axis or you can choose to display power or sample size on the vertical axis with the other quantity appearing on the horizontal axis. The default is **Quantity solved for** (or power) on the vertical axis, which is appropriate for this graph.

The summary table is created using the two sample sizes specified in the Sample Size table, 10 and 20. If you want to create a graph that contains more than these two sample sizes, you can do so by customizing the value ranges for the graph. Click the **Value Ranges** tab to set the axis range for sample sizes, as shown in Figure 68.12.

Figure 68.12 Customize Graph Window with Value Ranges Tab



Enter `5` for the minimum and `30` for the maximum. Also, select **Interval between points** in the drop-down list and enter a value of `1`. These values set the sample size axis to range from 5 to 30 in increments of 1. The completed Value Ranges section of the window is shown in Figure 68.12.

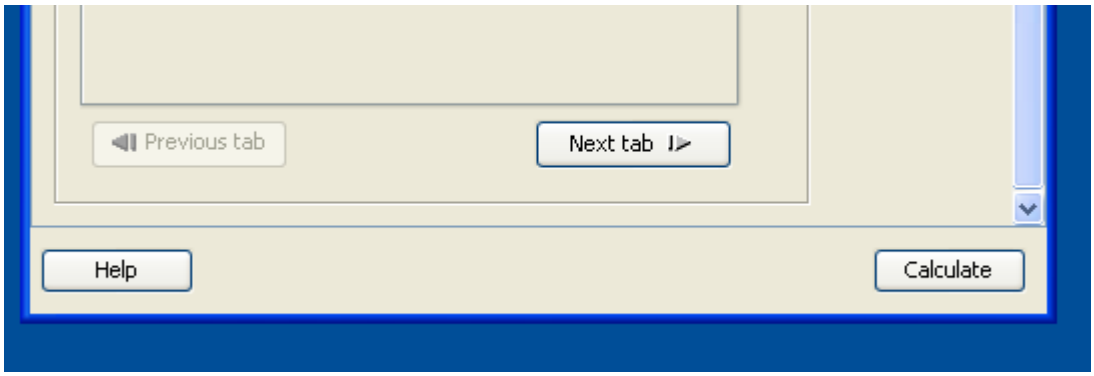
When you solve for power, you can set a range for sample size values, but not for the powers; and vice versa when you solve for sample size. That is, you cannot set the range of axis values for the quantity that you are solving for.

Click **OK** to save the values that you have entered and return to the Edit Properties page.

Performing the Analysis

You have now specified all of the necessary input values. Click **Calculate** to perform the analysis, as shown in Figure 68.13.

Figure 68.13 Calculate Button on the Edit Properties Page



Alternatively, you could choose to save the information that you have entered by selecting **File►Save** from the menu bar or clicking the **Save** toolbar icon, and perform the analysis at another time. No error checking is done when you save the project.

You can close the project by selecting **File►Close** on the menu bar or clicking the window close **X** in the upper right corner of the project window. You can reopen a project by selecting **File►Open** on the menu bar or clicking the **Open** toolbar icon.

For this example, click **Calculate**.

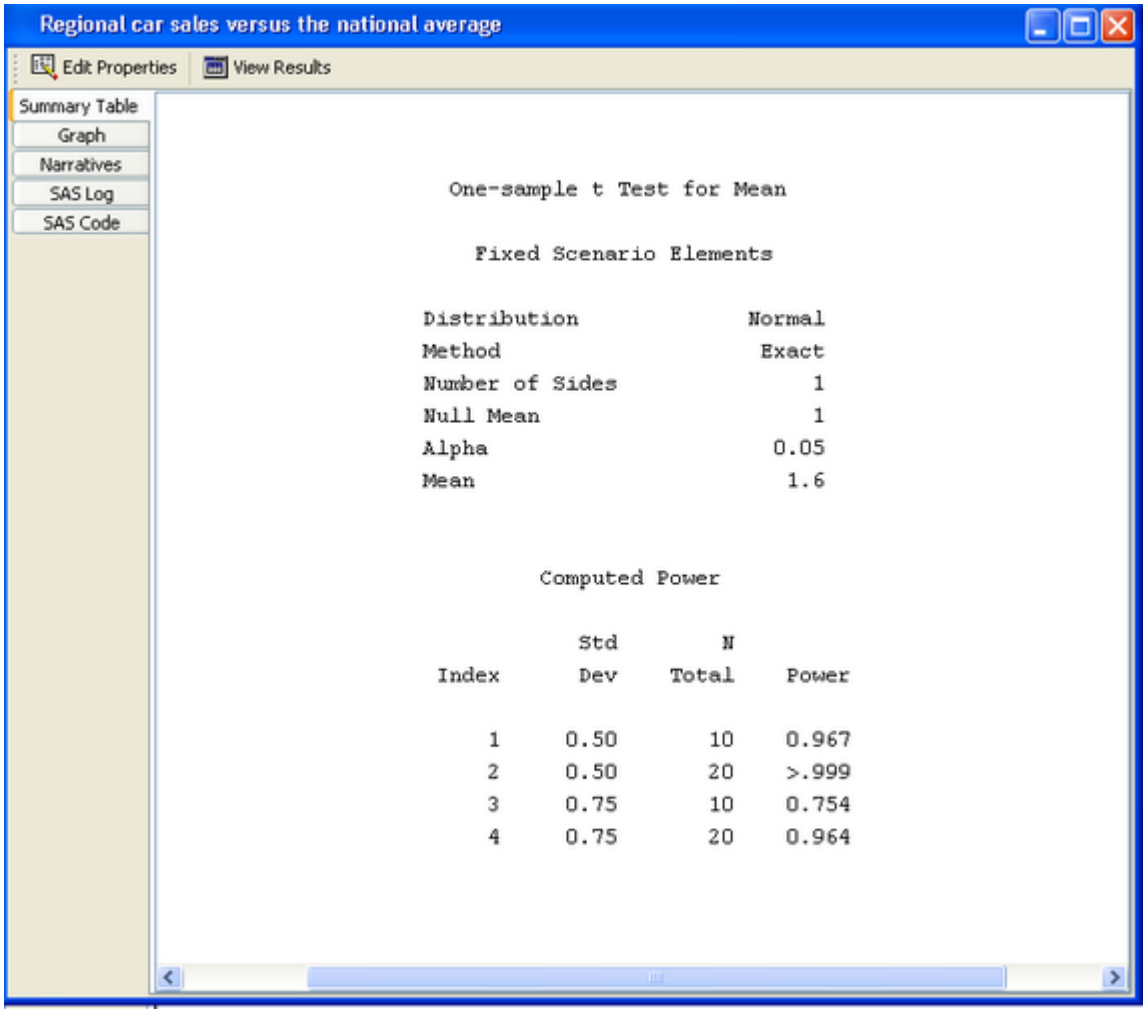
Viewing the Results

Results appear on the View Results page and are viewable in separate tabs. The tabs include **Summary Table**, **Graph**, **Narratives**, **SAS Log**, and **SAS Code** (located on the left side of the View Results page). The **Summary Table** and **Graph** tabs appear if you selected those options on the **Results** tab of the Edit Properties page. The other tabs always appear.

Summary Table

Click the **Summary Table** tab to view the summary table.

Figure 68.14 Summary Table Tab with Fixed Scenario Elements and Computed Power Tables



The Summary table consists of two subtables, as shown in Figure 68.14. The `Fixed Scenario Elements` table includes the parameters or options that have a single value for the analysis. The `Computed Power` table contains the input parameters that have been given more than one value, and it shows the computed quantity, power.

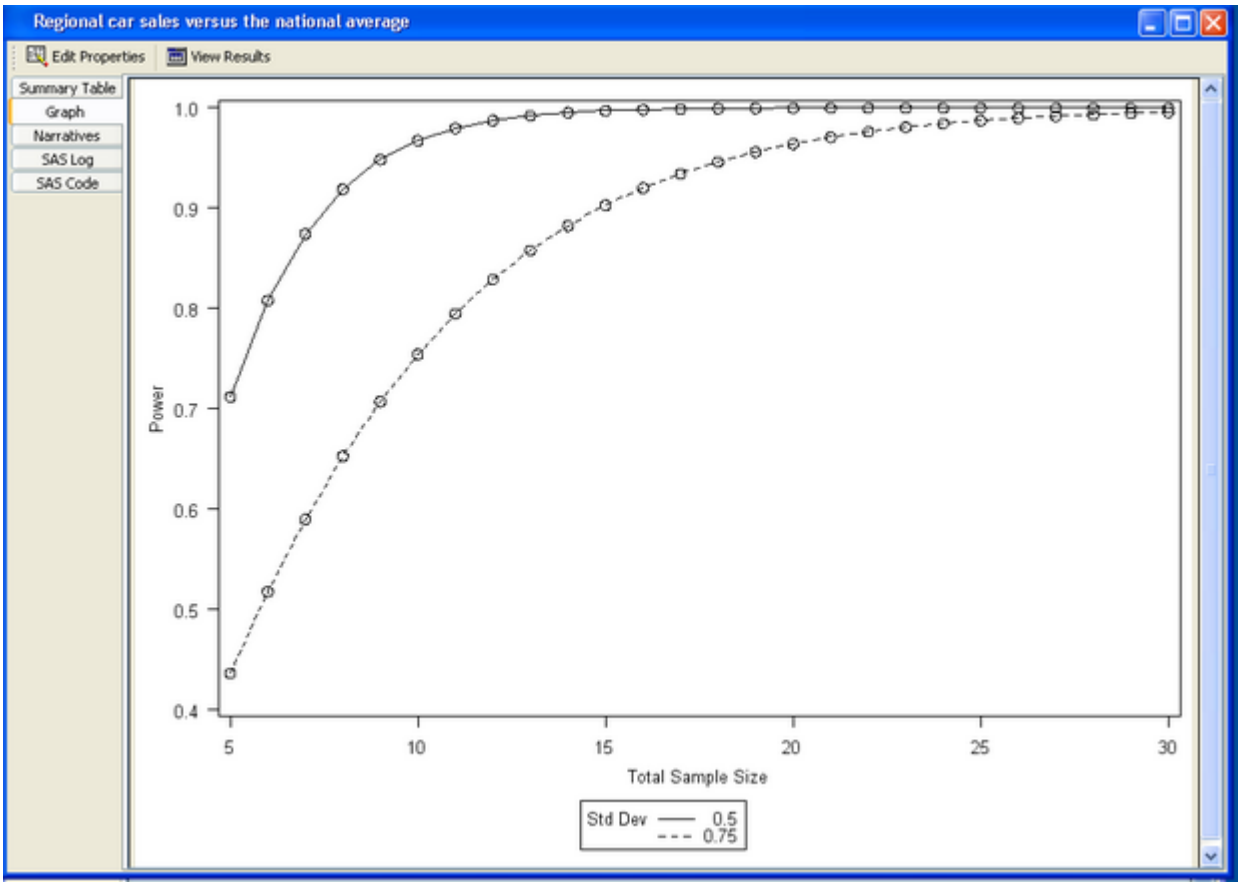
Thus, the `Computed Power` table contains four rows for the four combinations of standard deviation and sample size. From the table you can see that all four powers are high. The smallest value of power, 0.754, is associated with the largest standard deviation and the smallest sample size. In other words, the probability of rejecting the null hypothesis is greater than 75% in all four scenarios.

Power by Sample Size Graph

Click the **Graph** tab to view the power by sample size graph.

The power by sample size graph in Figure 68.15 contains one curve for each standard deviation. For a standard deviation of 0.5 (the upper curve), increasing sample size above 10 does not lead to much increase in power. If you are satisfied with a power of 0.75 or greater, 10 samples would be adequate for standard deviations between 0.5 and 0.75.

Figure 68.15 Graph Tab with Power by Sample Size Graph



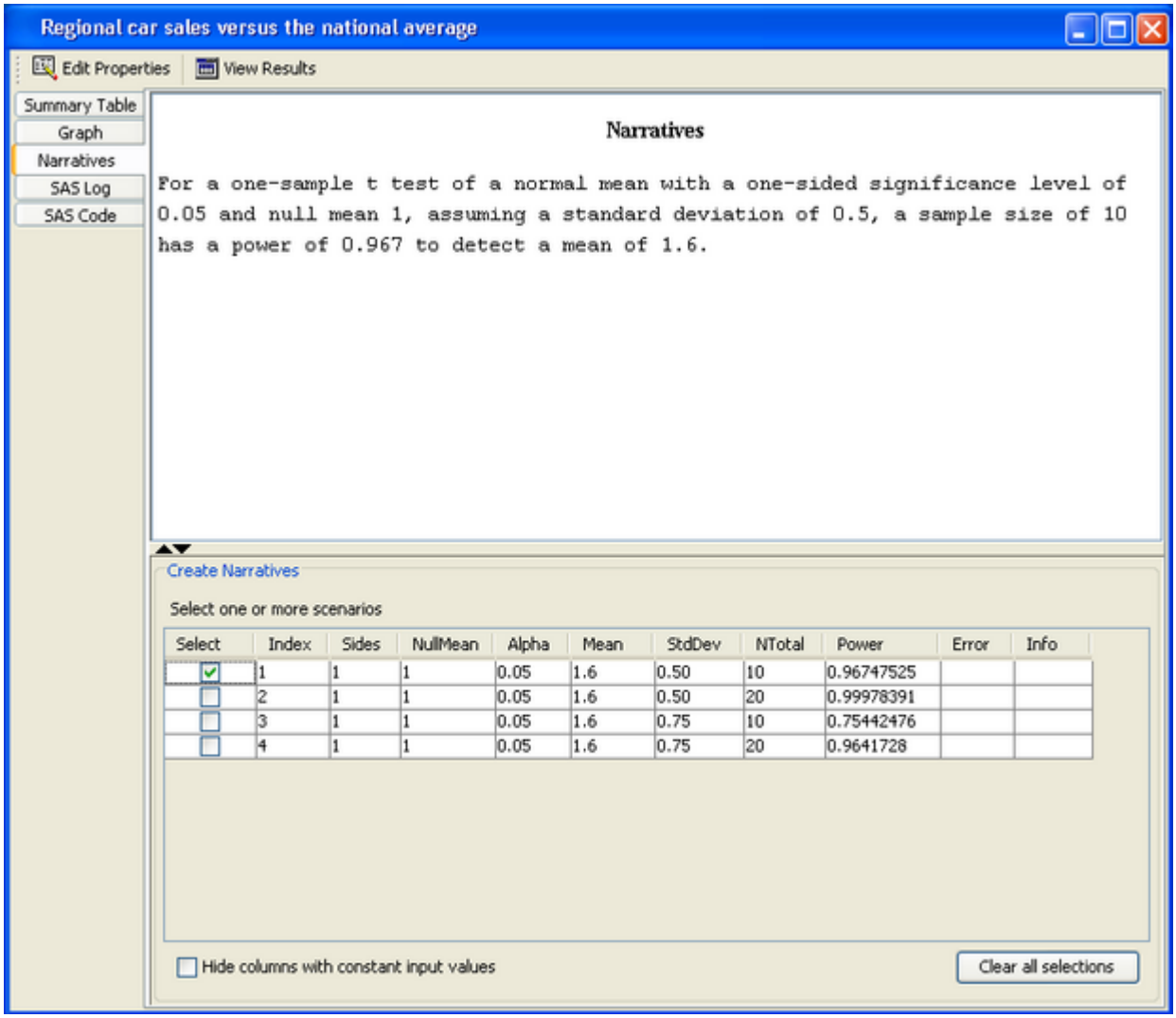
Narratives

Click the **Narratives** tab to display a facility for creating narratives.

Narratives are descriptions of the values that compose each scenario and include a statement about the computed power or sample size.

To create narratives, choose one or more scenarios in the table at the bottom of the tab. A narrative for each selected scenario is displayed in the top portion of the tab. See Figure 68.16.

Figure 68.16 Narrative Tab



For the example, select the first row in the table. The following narrative is displayed for the scenario with a standard deviation of 0.5 and a sample size of 10:

```
For a one-sample t test of a normal mean with a one-sided significance
level of 0.05 and null mean 1, assuming a standard deviation of
0.5, a sample size of 10 has a power of 0.967 to detect a mean of
1.6.
```

You can select several rows in the table. As you select each one, a corresponding narrative is created and displayed in the top portion of the table. Selecting a second scenario (the third row) produces the following output, where the narrative for the first row is followed by the narrative for the third row:

```
For a one-sample t test of a normal mean with a one-sided significance
level of 0.05 and null mean 1, assuming a standard deviation of
0.5, a sample size of 10 has a power of 0.967 to detect a mean of
1.6.

For a one-sample t test of a normal mean with a one-sided significance
level of 0.05 and null mean 1, assuming a standard deviation of
0.75, a sample size of 10 has a power of 0.754 to detect a mean of
1.6.
```

Other Results

Other results include the SAS log and the SAS code.

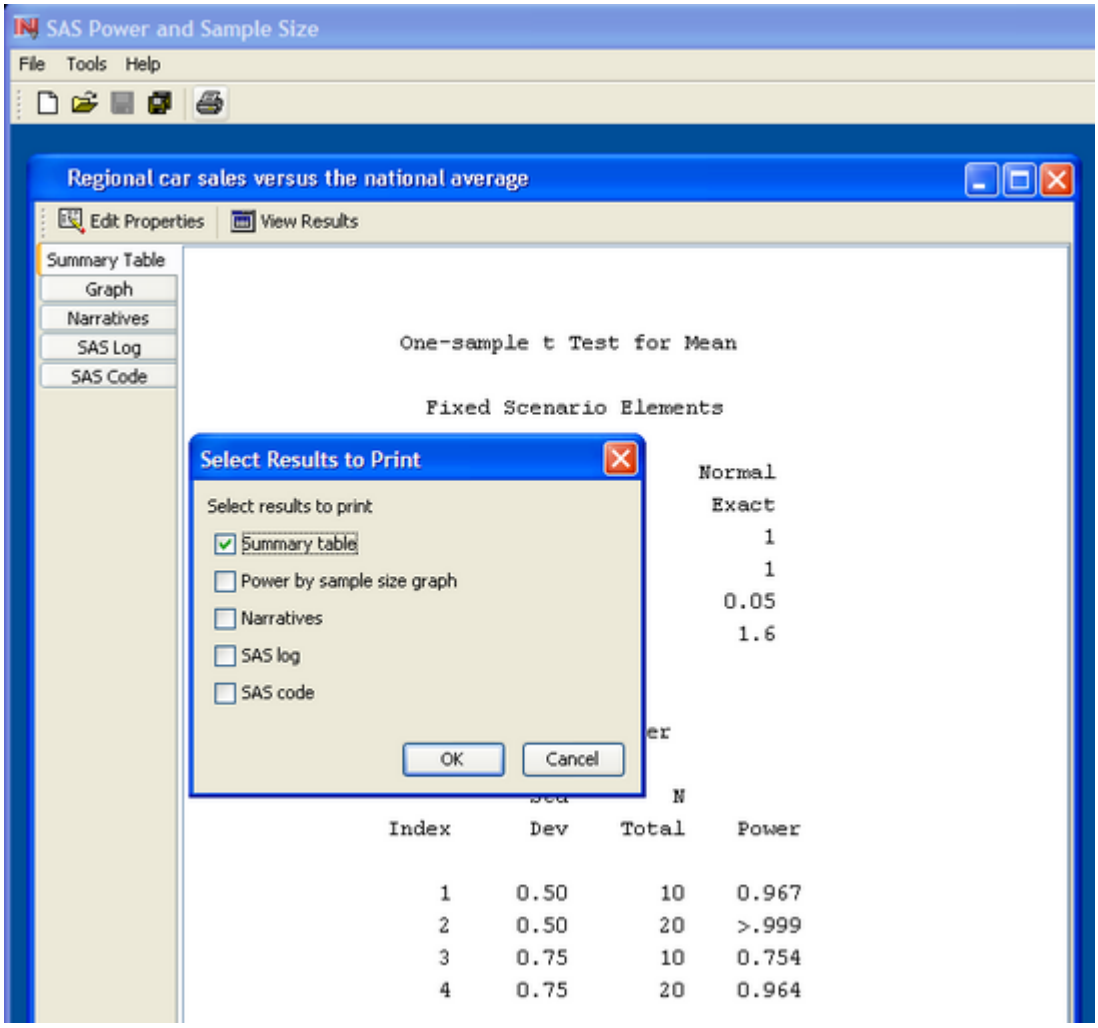
The SAS log that was produced when the **Calculate** button was last clicked appears on the **SAS Log** tab.

The SAS statements that produced the results appear on the **SAS Code** tab.

Printing Results

To print one or more results, select **File►Print** from the menu bar or click the **Print** toolbar icon. A window is displayed that lists all available results, as shown in Figure 68.17. Select the results that you want to print and click **OK**.

Figure 68.17 Print Selection Window



Changing Properties

If you want to change some values of the properties and rerun the analysis, change to the Edit Properties page and continue. The icons for selecting the Edit Properties and View Results pages are in the command bar just below the project window title.

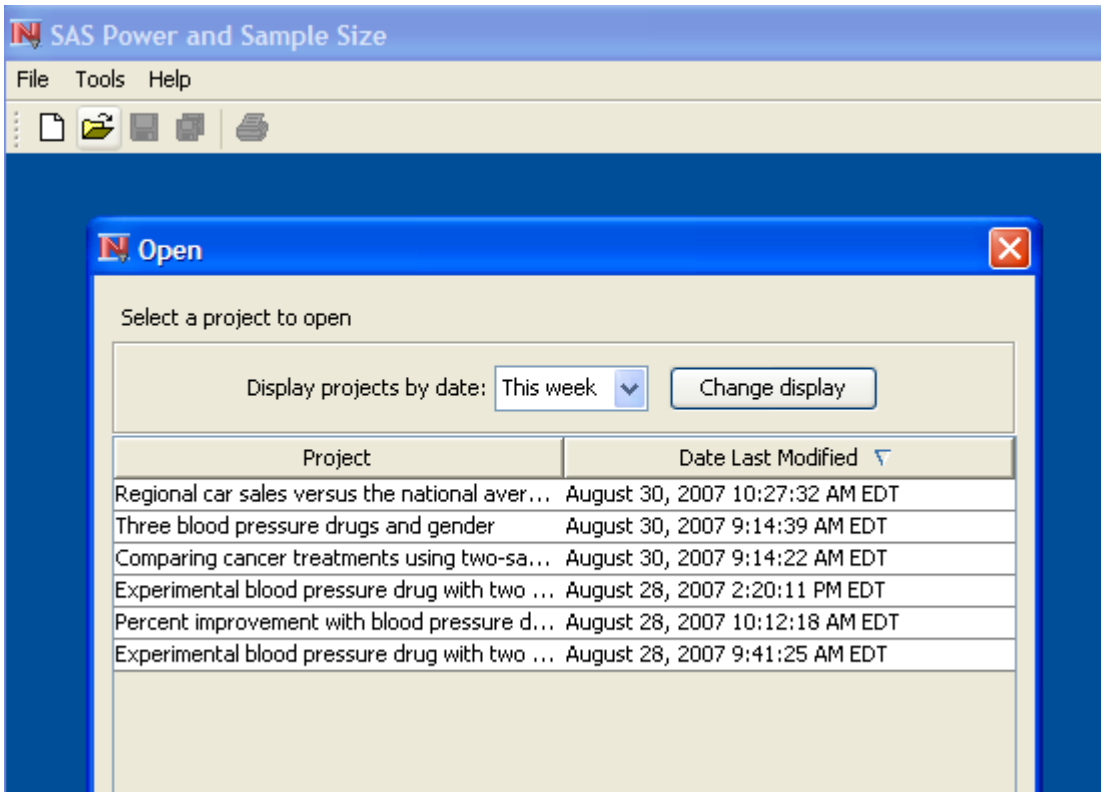
Closing the Project

When you are finished working with a project, close it by clicking the **X** in the upper right corner of the project window or selecting **File►Close** on the menu bar. If you have not saved the project, you will be asked if you want to save it before closing.

Opening a Project

You can reopen existing projects using the Open window. Select **File►Open** on the menu bar or click the **Open** toolbar icon.

Figure 68.18 Open Window Containing the Analysis Created in the Example



As shown in Figure 68.18, the analysis that you just completed is listed in the table. The label that you assigned to it, `Regional car sales versus the national average`, appears in the **Project** column of the table. The table also contains the date that the analysis was last modified. If you do not see the project that you are looking for, change the value of the **Display projects by date** box to `All` by selecting `All` from the drop-down list, and click the **Change display** button.

You can sort the projects in the table by clicking the header of the desired column. The sort direction is indicated by arrows displayed in the column header.

Select the project that you want to open and click **OK**. You can also double-click the project entry to open it.

Changing Values and Rerunning the Analysis

After viewing the graph, you might want to re-create the graph with a different range for sample sizes. On the **Results** tab of the Edit Properties page, click the **Customize** button for the power by sample size graph. The **Customize Graph** window is displayed.

On the **Value Ranges** tab of the window, change the Maximum value in the Sample Size table from `30` to `20`. Click **OK**.

Rerun the analysis by clicking **Calculate**. The View Results page is displayed again and the graph now has the new maximum value for the sample size axis.

How to Use: PSS Application

Overview

The PSS application is an application that resides on your desktop. It requires a connection to SAS software either on your desktop machine or a remote machine. You can set default values for several parameters and options as preferences. More detail on creating and editing projects is provided. Projects can be imported and exported.

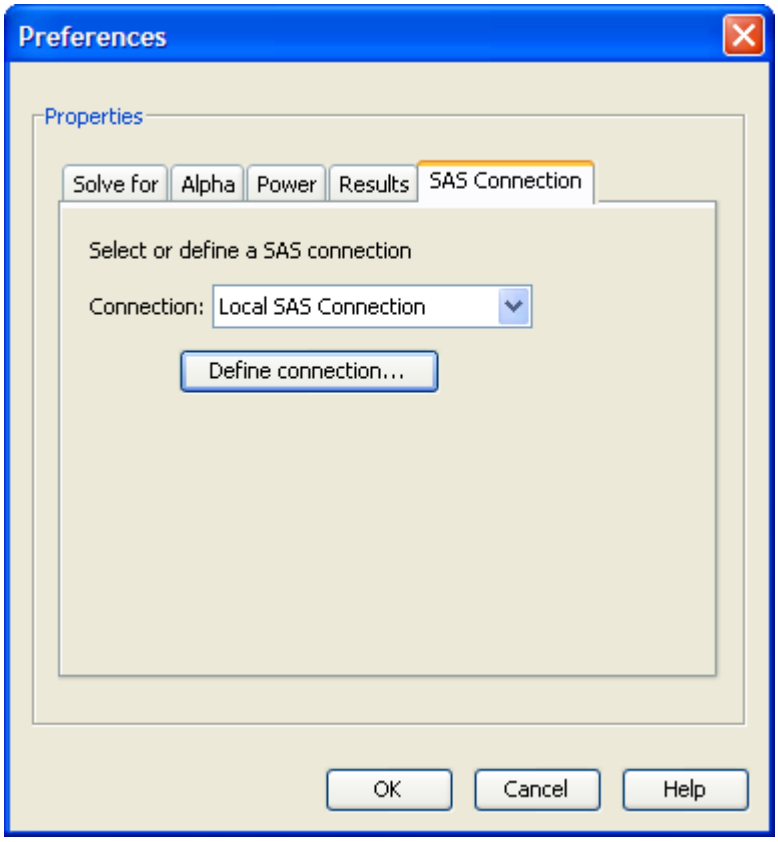
SAS Connections

Connections to SAS servers are defined in the Preferences window. To access the Preferences window, select **Tools►Preferences** on the menu bar.

Click the **SAS Connection** tab to select or define a connection to a SAS server. A connection to a SAS server is required in order to calculate results. The server can be on your local (desktop) machine or on a remote machine.

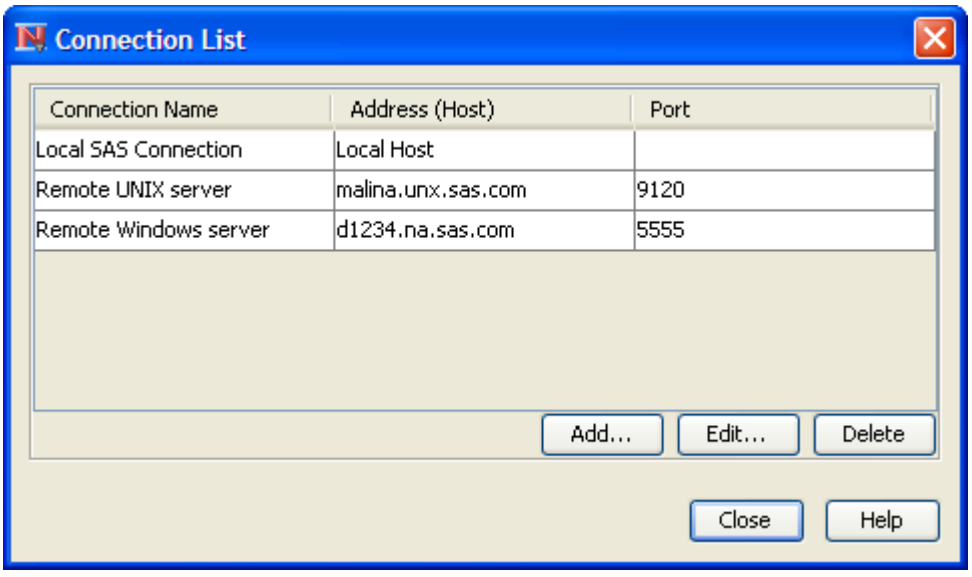
You can define several SAS connections and choose the one you want to use. To select a previously defined connection, choose it from the **Connection** list on the **SAS Connection** tab; see Figure 68.19.

Figure 68.19 SAS Connection Tab



To define a SAS connection, click the **Define connection** button. The Connection List window appears, as shown in Figure 68.20. To create a new connection, click **Add**. To edit an existing connection, select it in the Connection List and click **Edit**.

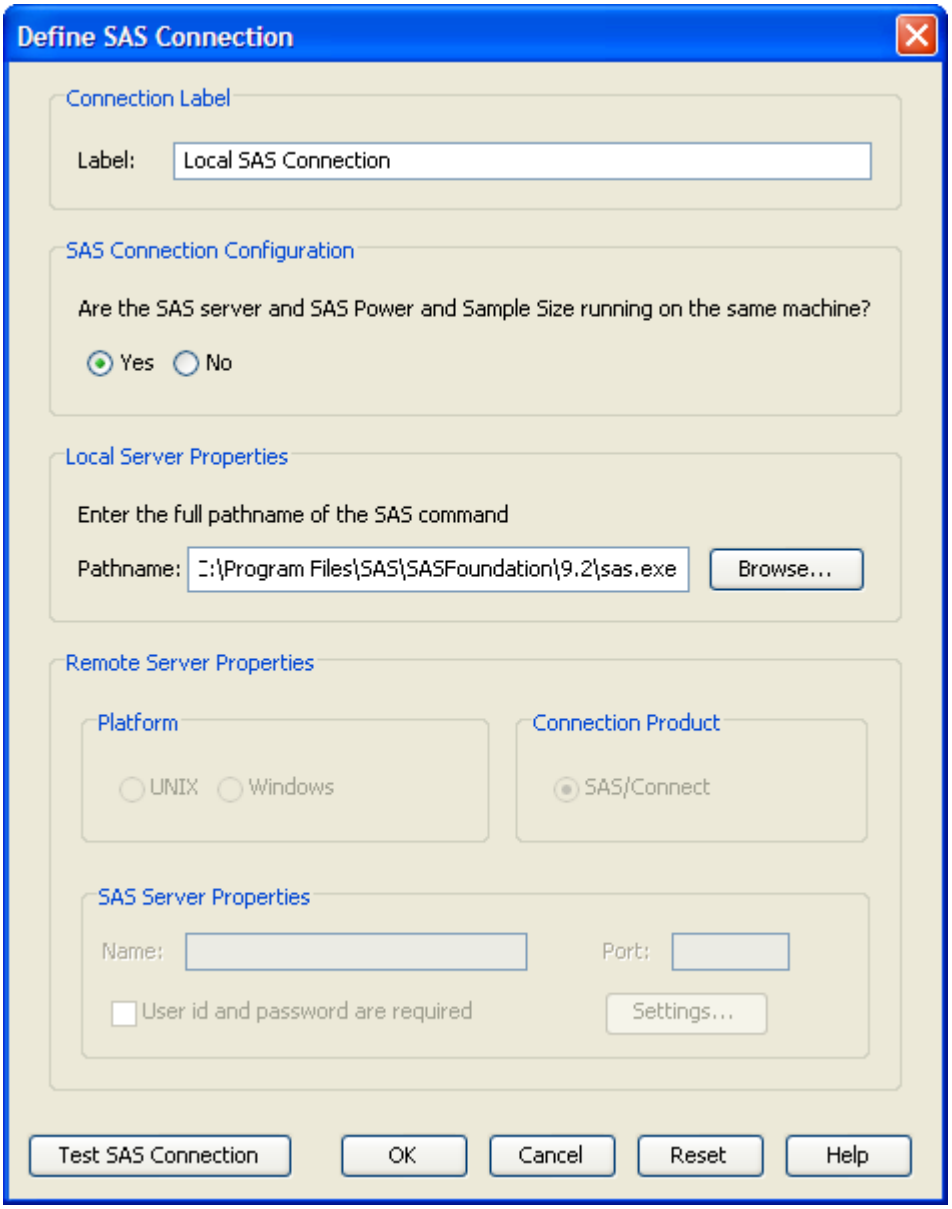
Figure 68.20 Connection List



Defining a SAS Connection

After you click the **Add** or **Edit** button, the Define SAS Connection window appears, as shown in Figure 68.21. If you clicked **Edit**, the previously defined information is available for editing.

Figure 68.21 Define Connection Window



Enter a descriptive label for the connection. The label is used to distinguish among the connections in the connections list.

Then, select **Yes** or **No** to specify whether the SAS connection is to the local machine (that is, the one on which PSS is running) or to a remote machine, respectively.

Defining a Local Connection

To define a connection to the local machine, enter the full path name of (or browse for) the SAS executable file (*sas.exe* on Microsoft Windows).

Test the SAS connection by clicking the **Test SAS Connection** button.

Defining a Remote Connection

To define a connection to a remote machine, select either the **UNIX** or **Windows** option to indicate that the remote SAS server is on a machine running the UNIX or Microsoft Windows operating systems, respectively. Then, specify the machine name and port number that the SAS/Connect spawner is using on the remote machine. Contact the SAS server administrator for this information.

If the remote machine is running Microsoft Windows, select the **User id and password are required** if authentication is required to access the SAS server (that is, if the SAS **`-security`** option is used). By default, authentication is required for SAS servers running on UNIX operating systems.

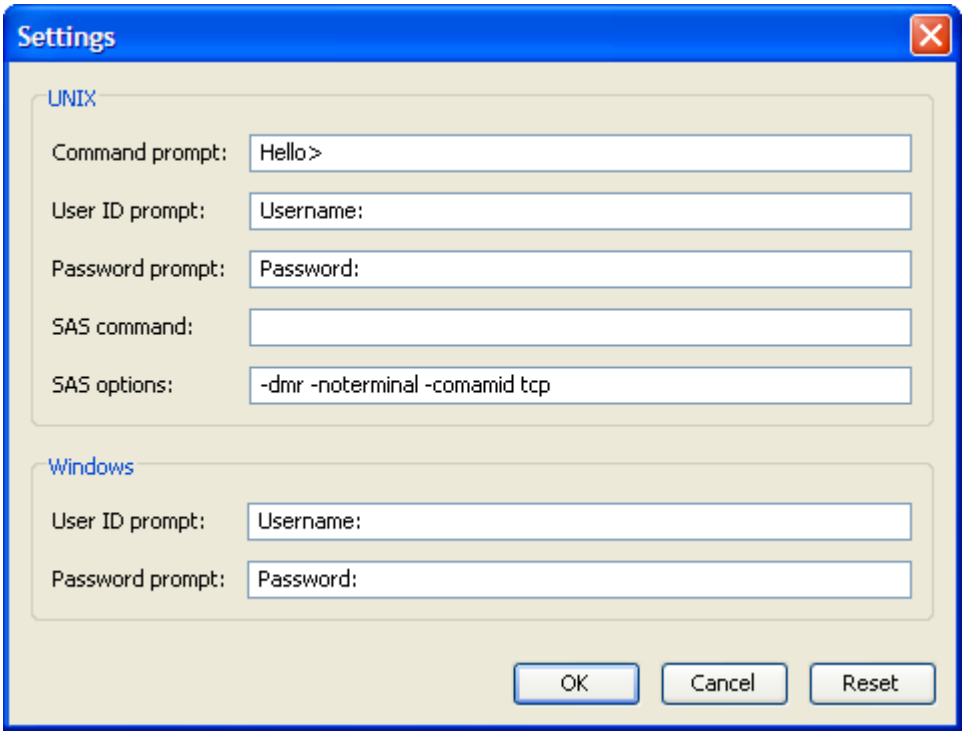
Test the SAS connection by clicking the **Test SAS Connection** button.

Additional Settings

Click the **Settings** button on the Define SAS Connection window to access some additional settings for a remote connection to a SAS server. For the most part these settings are prompts that PSS expects to receive from the SAS/CONNECT spawner on the remote machine, as shown in Figure 68.22.

If the remote SAS server is on a UNIX machine, you must specify the full pathname of the SAS command. Contact the SAS server administrator for this information.

Figure 68.22 Connection Settings Window

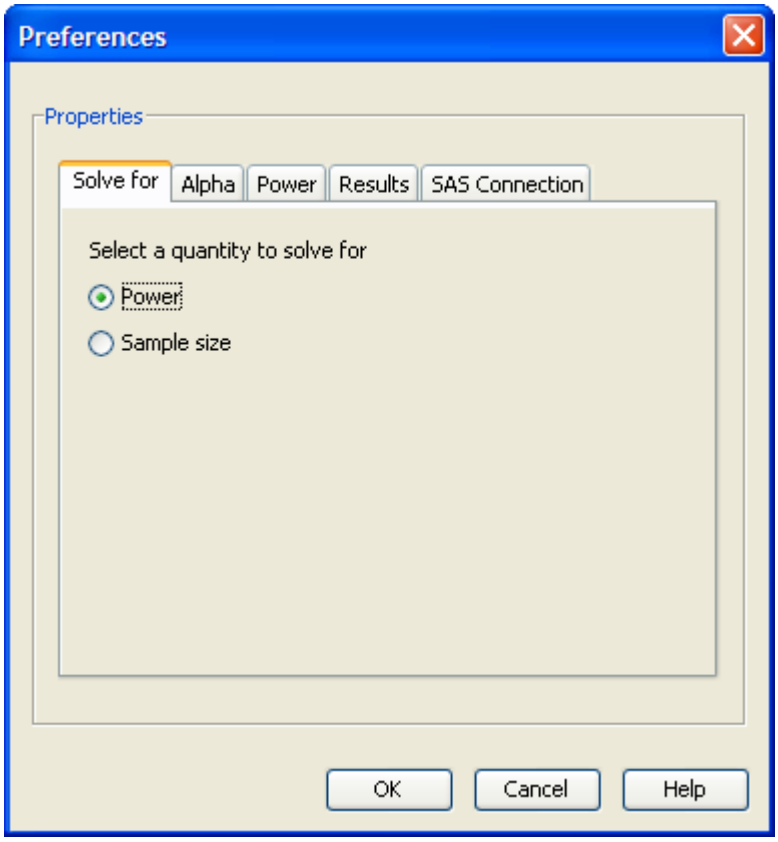


Setting Preferences

In the Preferences window you can set default values for options that are used by all analyses.

To access the Preferences window, select **Tools►Preferences** on the menu bar. Figure 68.23 shows the Preferences window.

Figure 68.23 Preferences Window



Preference values are used as the defaults for each newly opened project (that is, those that are opened from the New window). For a specific project, each of these default values can be overridden on the Edit Properties page.

Changes in preferences do not change the state of an existing analysis (that is, one that is accessed from the Open window).

Selecting the Quantity to Solve For

Click the **Solve For** tab to select **Power** or **Sample Size** as the default value to be solved for; see Figure 68.23. For confidence interval analyses, selecting **Power** is equivalent to selecting **Prob(Width)**.

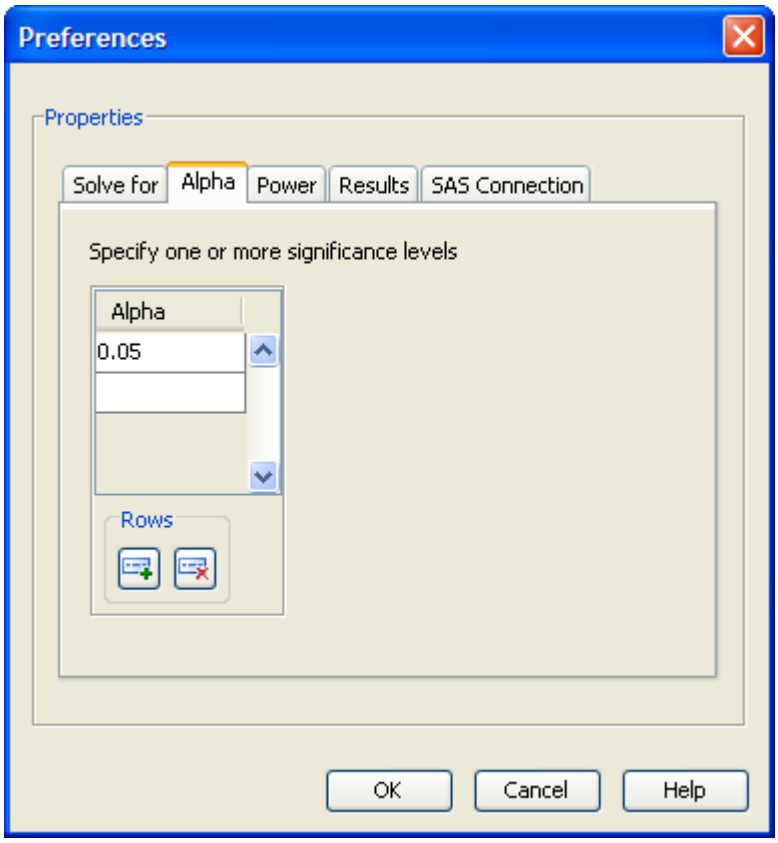
For analyses that offer both **Sample size per group** and **Total sample size**, the **Sample size** option on this page corresponds to total sample size.

Setting Alphas

Click the **Alpha** tab to enter one or more values for alpha. Alpha is the significance level (false positive probability). For confidence interval analyses, alpha values are transformed into confidence levels by $(1 - alpha)$. For example, an alpha of 0.05 would represent a confidence level of 0.95.

To set default values of alpha, enter one or more values in the **Alpha** data entry table. See Figure 68.24. It is not necessary to have any default values for alpha. Add more rows to the table as needed using the button at the bottom of the table.

Figure 68.24 Alpha Preference Tab

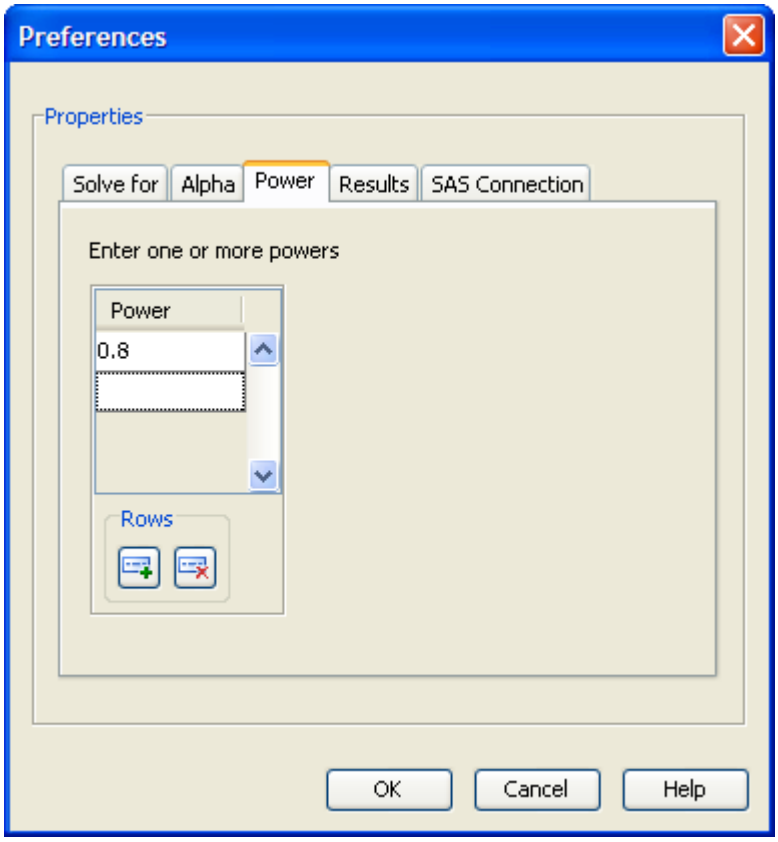


Setting Powers

Click the **Power** tab to enter one or more values for power. It is not necessary to have any default values for power. For confidence interval analyses, power values are treated as prob(width) values.

To set default values of power, enter one or more values in the **Power** data entry table; see Figure 68.25.

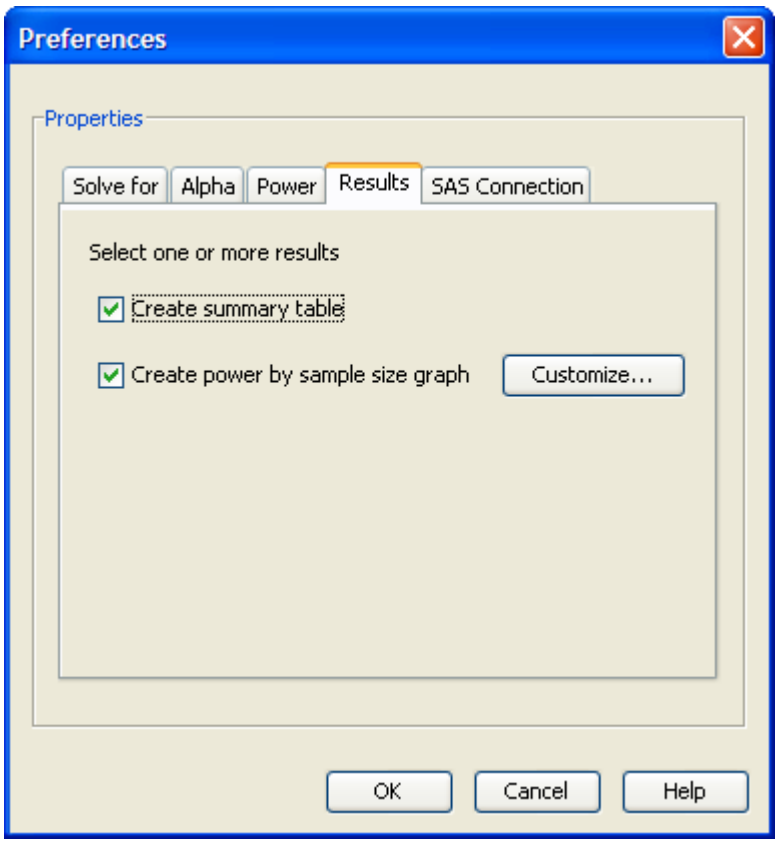
Figure 68.25 Power Preference Tab



Setting Results Options

Click the **Results** tab to make default selections for the summary table and the power by sample size graph options; see Figure 68.26.

Figure 68.26 Results Options Preferences Tab



The summary table consists of the input parameter values and the calculated quantity (power or sample size). Select the **Create summary table** check box to create the table by default.

To request that an analysis create a power by sample size graph by default, select the **Create power by sample size graph** check box.

Creating and Editing PSS Projects

A PSS project is an instance of an analysis. The first decision in using PSS is to choose the appropriate test or design. Select the **File►New** on the menu bar or click the **New** icon on the toolbar. The New window appears with a list of the available analyses. Select the type of analysis that you want from the list and click **OK**.

When the project is first opened, the Edit Properties page is displayed. It is described in the section "Editing Properties" on page 5077.

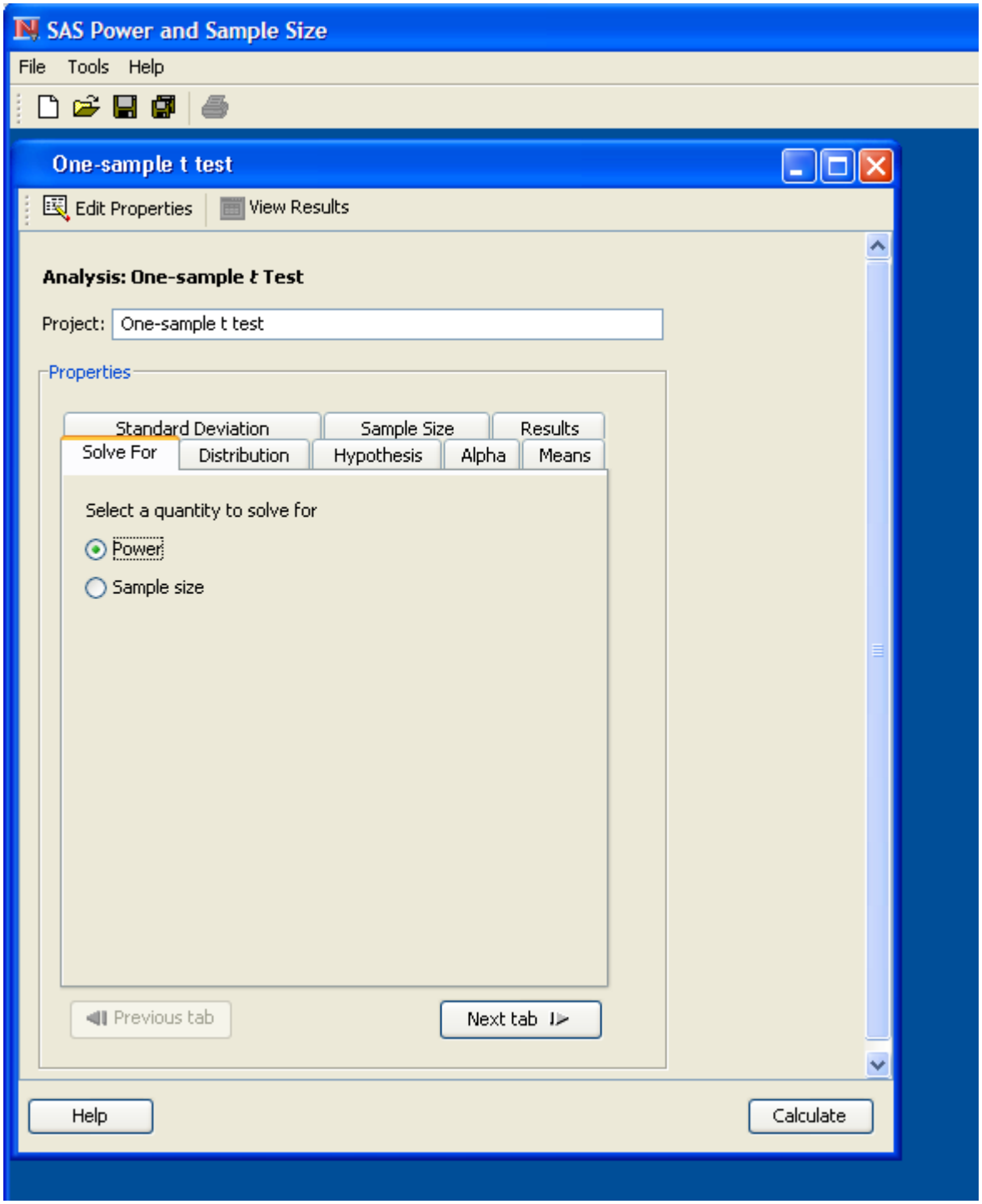
After the properties have been specified and the analysis is performed, the View Results page is displayed. See the section "Viewing the Results" on page 5081.

A project that has been saved and closed can be reopened from the Open window. Select **File►Open** on the menu bar or click the **Open** icon on the toolbar.

Editing Properties

The Edit Properties page consists of several analysis options and input parameters that are relevant to the particular analysis. These options and parameters are organized on several tabs, as shown in Figure 68.27.

Figure 68.27 Edit Properties Page

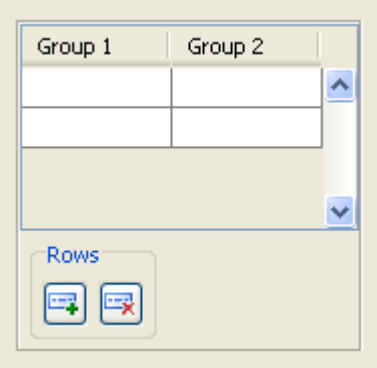


The Edit Properties page contains various controls by which you can enter values or select choices. In addition to the usual data entry controls such as text fields and check boxes, several specialized controls are present: data entry tables and the Alternate Forms control. More detailed descriptions follow.

Using Data Tables

Data entry tables are composed of data entry fields for one or more rows and columns. Figure 68.28 shows a two-row, two-column table.

Figure 68.28 Two-Column Data Entry Table with Controls



Type an appropriate value in each field. It is not necessary to type data in all rows or to delete empty rows. However, if a table has more than one column, the cells of a row must be completely filled or completely blank. Rows with values in some but not all cells are not allowed.

To append more rows, click the button beneath the table. To delete the last row of the table, click the button.

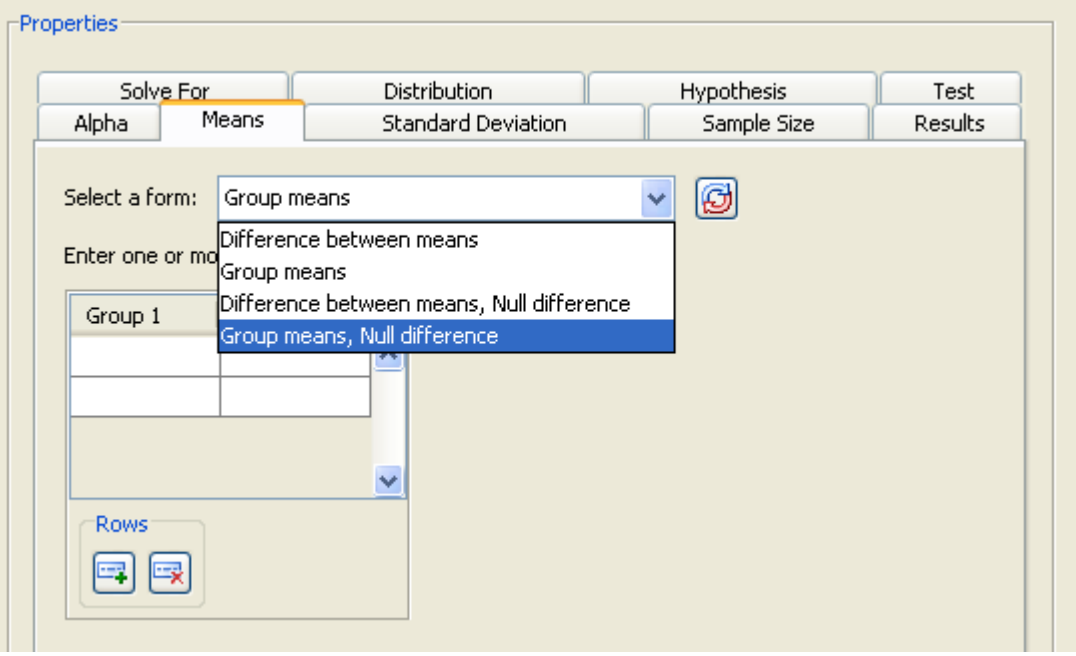
Also, you can display a pop-up menu to perform additional actions such as inserting and deleting rows. First, select the row to insert before or delete, then right-click to display the pop-menu and select the desired action.

Using Alternate Forms

For some input parameters, there are several ways in which data may be entered. For example, in the two-sample *t* test analysis, group means can be entered as either individual means or a difference between means.

The alternate forms are displayed in a drop-down list with an adjacent button, as shown in Figure 68.29. The button enables you to cycle through the alternatives, displaying each one in turn. To see what forms are available, you can open the drop-down list and select the one you want or you can click the button until the form that you want is displayed.

Figure 68.29 Select a form Drop-Down List and Button



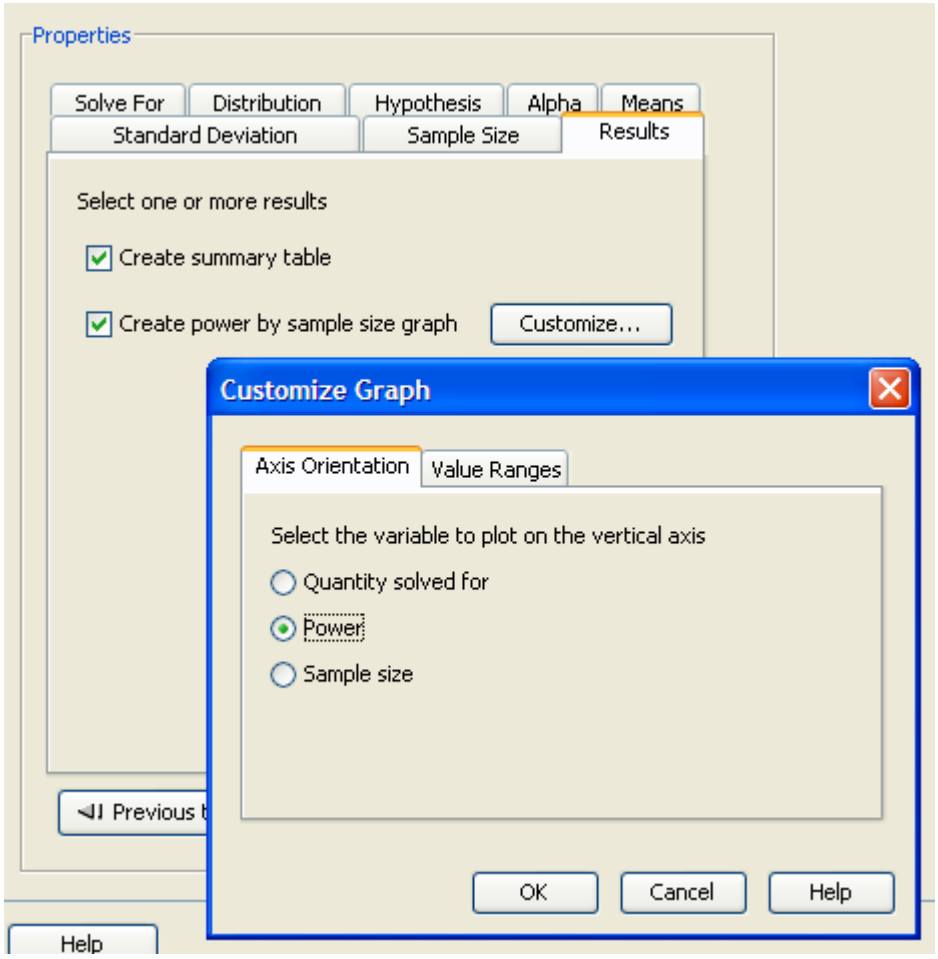
The alternate form last used for an analysis is saved and displayed as the default when a new instance of the analysis is opened.

Customizing Graphs

The Edit Properties page for all analyses contains a **Results** tab. You can choose to create a graph, and you can optionally choose to customize the graph by clicking the **Customize** button that is beside the **Create power and sample size graph** choice.

As shown in Figure 68.30, the Customize Graph window consists of an **Axis Orientation** tab and a **Value Ranges** tab. Use the **Axis Orientation** options to specify which axes you want used for power and for sample size. Use the **Value Ranges** settings to specify the axis range for the non-target quantity (that is, the power axis if you are solving for sample size or the sample size axis if you are solving for power).

Figure 68.30 Customize Graph Window



When specifying a value range, you can specify a minimum value and a maximum value. Also, you can select either the `Number of points` or the `Interval between points` choice for the axis and specify a value. All of these values are optional; specify only the ones you want.

Scenarios

A scenario is one instance of a complete set of values for an analysis. For example, if two alpha values and two total sample size values are specified with all other input parameters taking only a single value, there would be four scenarios—the four combinations of two alphas and two sample sizes.

Performing the Analysis

To perform the analysis, click **Calculate** at the lower right of the Edit Properties page. The input parameters are checked for validity, and the analysis is performed. The View Results page is then displayed.

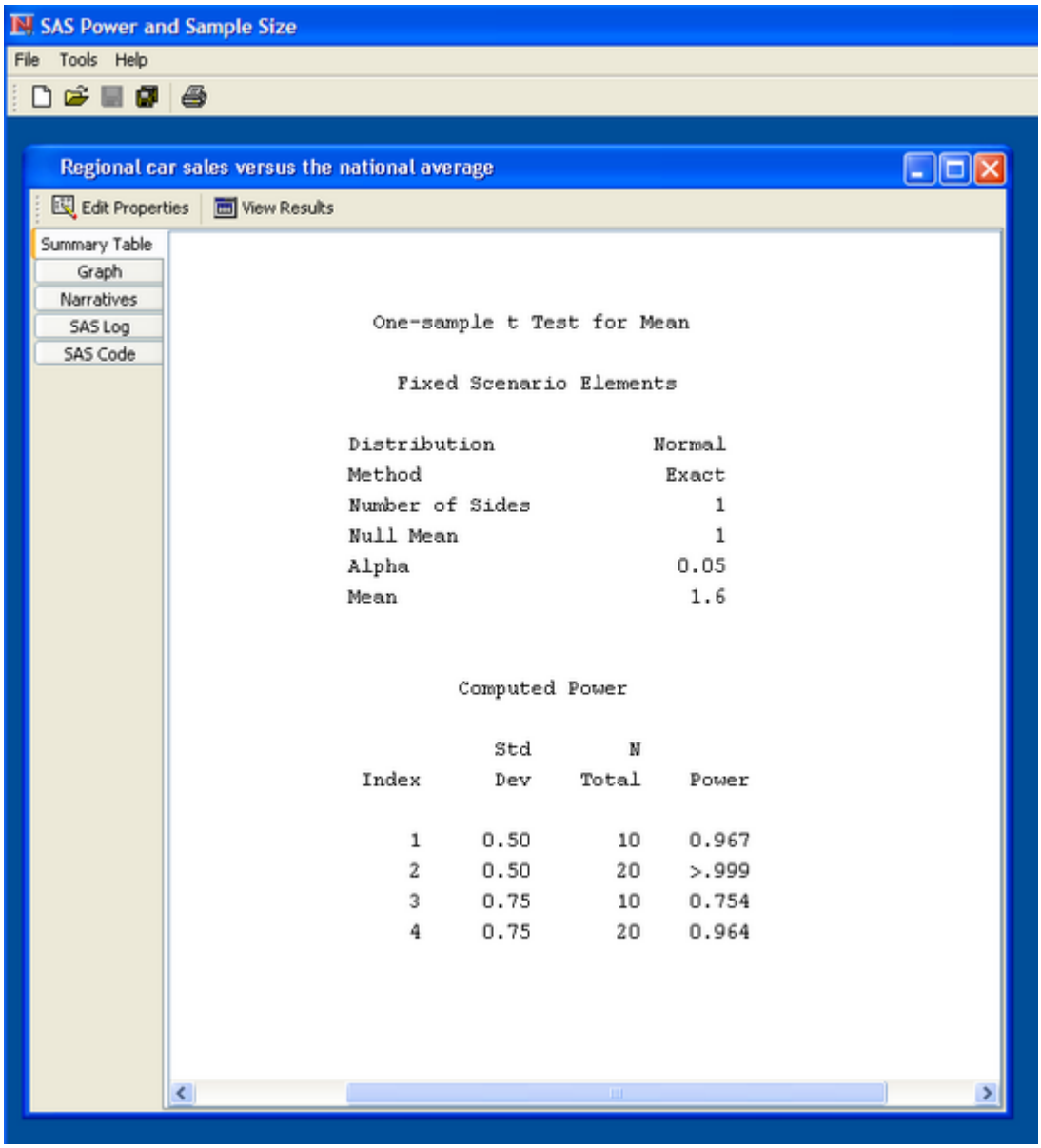
Viewing the Results

The results appear in separate tabs on the View Results Page. These tabs include **Summary Table**, **Graph**, **Narratives**, **SAS Log**, and **SAS Code**.

Viewing the Summary Table

Click the **Summary Table** tab to view the summary table. It consists of two subtables, as shown in Figure 68.31. The `Fixed Scenario Elements` table includes the options and parameter values that are constant for the analysis. The `Computed Power` table includes the calculated power or sample size values and the values for input parameters that have multiple values specified for the analysis.

Figure 68.31 View Results Page with Summary Table

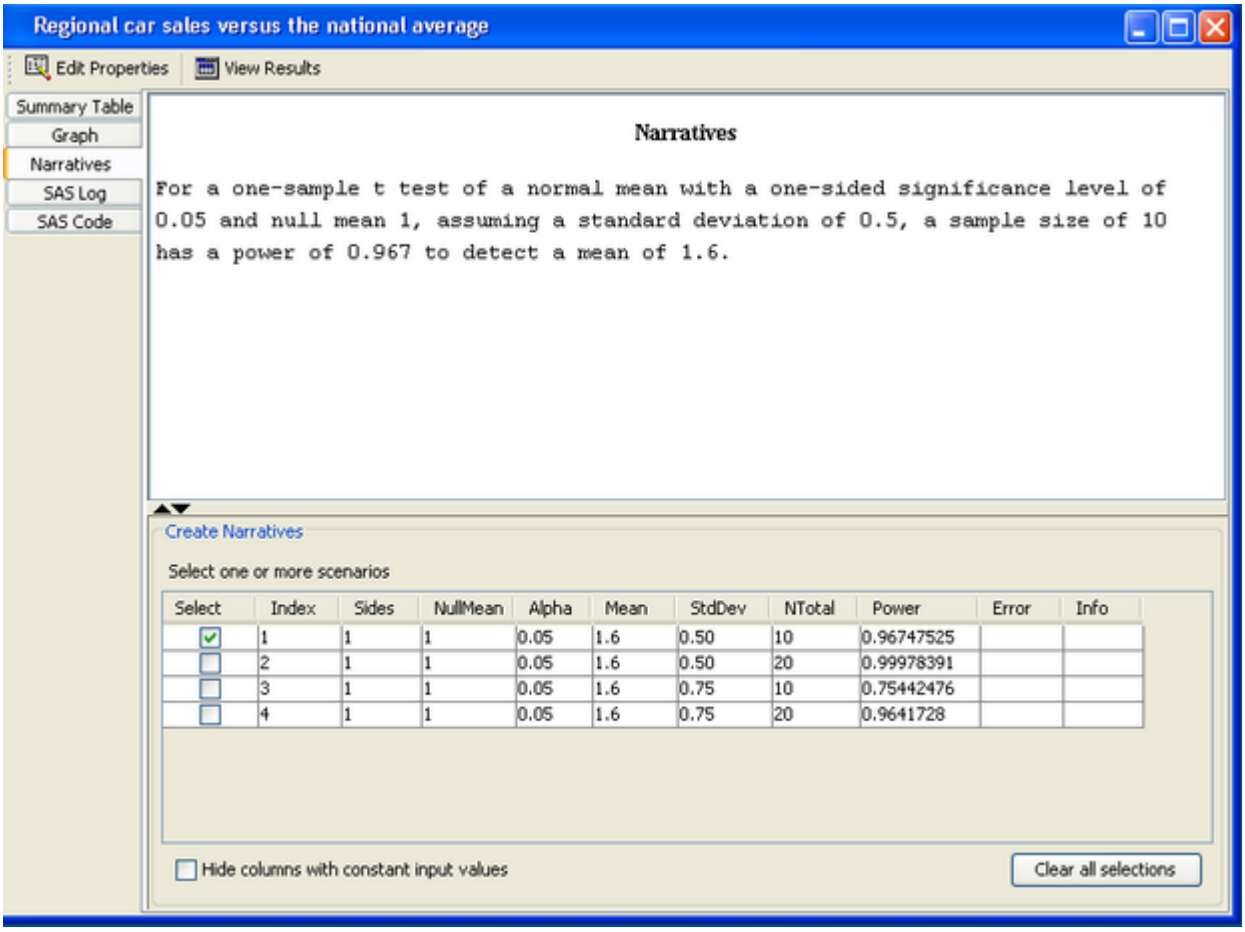


Creating Narratives

Click the **Narratives** tab to display a facility to create narratives. Narratives are descriptions of the input parameter values and calculated quantities in sentence or paragraph form. Each narrative corresponds to one calculated quantity value.

The **Narratives** tab is divided into a narrative selector panel and a narrative display panel. To create a narrative, select the row in the narrative selector panel that corresponds to it. You can select as many rows as you want. See Figure 68.32.

Figure 68.32 Narrative Selector and Display



The narrative selector table often contains columns whose values do not vary. For example, in Figure 68.32, the Sides, NullMean, Alpha, and Mean columns contain values that do not vary. You can hide these columns by selecting the **Hide columns with constant input values** check box.

Viewing the SAS Log and Code

Click the **SAS Code** tab to view the SAS statements that are used to generate the analysis results. Click the **SAS Log** tab to view the SAS log that corresponds to the analysis.

The SAS code differs slightly from the statements in the SAS log. Statements that are used to place the results in the location maintained by the application are not included. This is done to prevent you from overwriting the results stored by the application if you run the SAS code outside of the application.

Printing Results

To print one or more results, click the **Print** icon on the toolbar or select **File►Print** on the menu bar. The Select Results to Print window is displayed. You can choose to print one or more of the results by selecting the corresponding options here.

Saving the Project

To save a project, click the **Save** toolbar icon or select **File►Save** from the menu bar. Projects can be saved even if some of the information is invalid. Error checking is performed when the **Calculate** button is clicked.

Closing the Project

To close a project, click the **X** in the upper right corner of the project window or select **File►Close** from the menu bar.

Importing and Exporting Projects

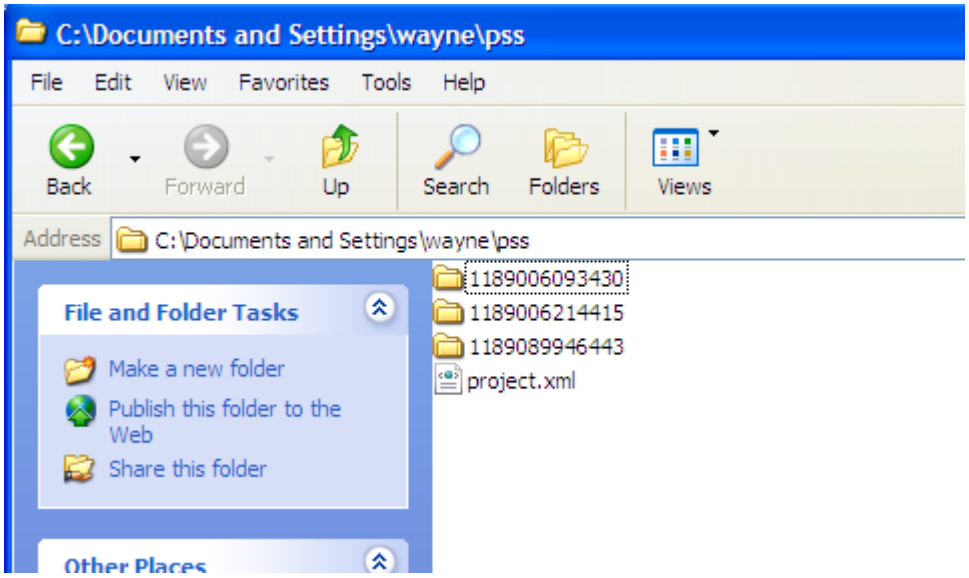
PSS projects can be imported from the same machine or a different machine. Also, the active project (the project that is open and on top of any other open projects) can be exported.

Importing Projects

A PSS project that was created on another machine or by another user can be imported and used. Also, importing projects is the recommended way of moving existing PSS projects that were created with PSS release 2.0 (a Web application) to PSS release 3.1 (a desktop application).

PSS files are stored in a folder entitled *pss*. The *pss* folder contains a *project.xml* file and individual folders for each project. See Figure 68.33.

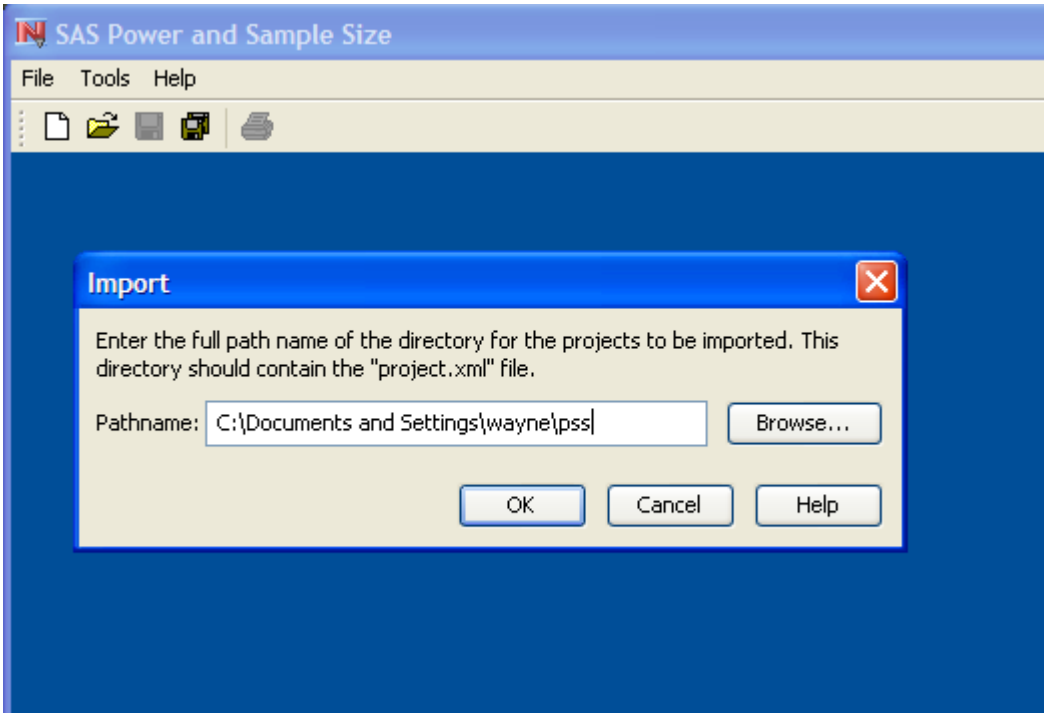
Figure 68.33 PSS Directory Structure



If PSS files are on another machine, they must first be copied to a temporary location on the desktop machine that is running PSS. The entire *pss* folder should be copied.

To import projects, select **File►Import** from the menu bar. Then, specify the full pathname of the *pss* folder.

Figure 68.34 Import Projects Window



To import PSS 2.0 files, you need to find the *pss* folder. The easiest way to do this is to search for the *project.xml* file. If you find several files with this name, you need to decide which one or more to import.

Exporting the Active Project

If you want to send a PSS project to someone, you can export the active project. The active project is the one that is open and that has focus (is displayed on top of any other open projects). Select **File►Export active project** and specify a temporary directory to hold the exported project.

The recipient must import the project using PSS.

Details: PSS Application

Software Requirements

PSS is available in SAS 9.2 for the following platforms: Microsoft Windows XP and Vista.

Two configurations are available for SAS connections: local and remote. With the local configuration, PSS and SAS 9.2 must reside on the same machine. With the remote configuration, PSS and SAS 9.2 can reside on different machines. SAS connections are defined and selected on the **SAS Connection** tab on the Preferences window. More information about SAS connections is found in the section "SAS Connections" on page 5068.

For both configurations, Base SAS and SAS/STAT software must be installed and SAS/GRAPH software is recommended.

For the remote configuration, SAS/CONNECT and SAS/IntrNet software must also be installed. For more information about configuring the remote SAS server, click **Help►Contents** on the menu bar and then click **Configuring a Remote SAS Server** under **Special Topics** in the table of contents.

Installation

SAS Power and Sample Size is installed separately from the SAS/STAT product. Contact your SAS site representative to have the application installed.

SAS Power and Sample Size is installed using the SAS Software Deployment Wizard. It is listed as an available product with, but separate from, Foundation SAS which contains the SAS/STAT and SAS/GRAPH products that are required for using the application.

Configuration

When you first run SAS Power and Sample Size 3.1 (PSS), you are asked to provide configuration information.

First, you are asked for the name of a directory to contain the your power and sample size projects. A folder named *pss* is created in the specified directory, and projects are stored in the *pss* folder. This directory cannot be the same as the one used by PSS 2.0. If it is, PSS requires that another folder name be provided.

Then, if the appropriate release of the SAS System is available on the desktop machine, you are asked whether a connection should be automatically created to it. If you respond `No`, then PSS informs you that a connection to the SAS server is necessary and asks if you want to select one now or later. A connection to a SAS server is not necessary to use the application until the **Calculate** button on the Edit Properties page of a project is clicked. More information about connections is available in the section "Setting Preferences" on page 5072.

Then, PSS displays a wizard to help you import existing PSS projects from either a previous release (PSS 2.0) or the current one (PSS 3.1). More information is available in the section "Importing Projects" on page 5083.

Example: Two-Sample *t* Test

Overview

The one-sample *t* test compares the mean of a sample to a given value. The two-sample *t* test compares the means of two samples. The paired *t* test compares the mean of the differences in the observations to a given number. PSS provides power and sample size computations for all of these types of *t* tests. For more information about power and sample size analysis for *t* tests, see Chapter 67, "The POWER Procedure."

The two-sample *t* test tests for differences or ratios between means for two groups. The groups are assumed to be independent. This example describes three examples using the two-sample *t* test: for equal variances, for unequal variances, and for mean ratios.

Test of Two Independent Means for Equal Variances

Suppose you are interested in testing whether an experimental drug produces a lower systolic blood pressure than a placebo does. Will 25 subjects per treatment group yield a satisfactory power for this test? From previous work, you expect that the blood pressure is 132 for the control group and

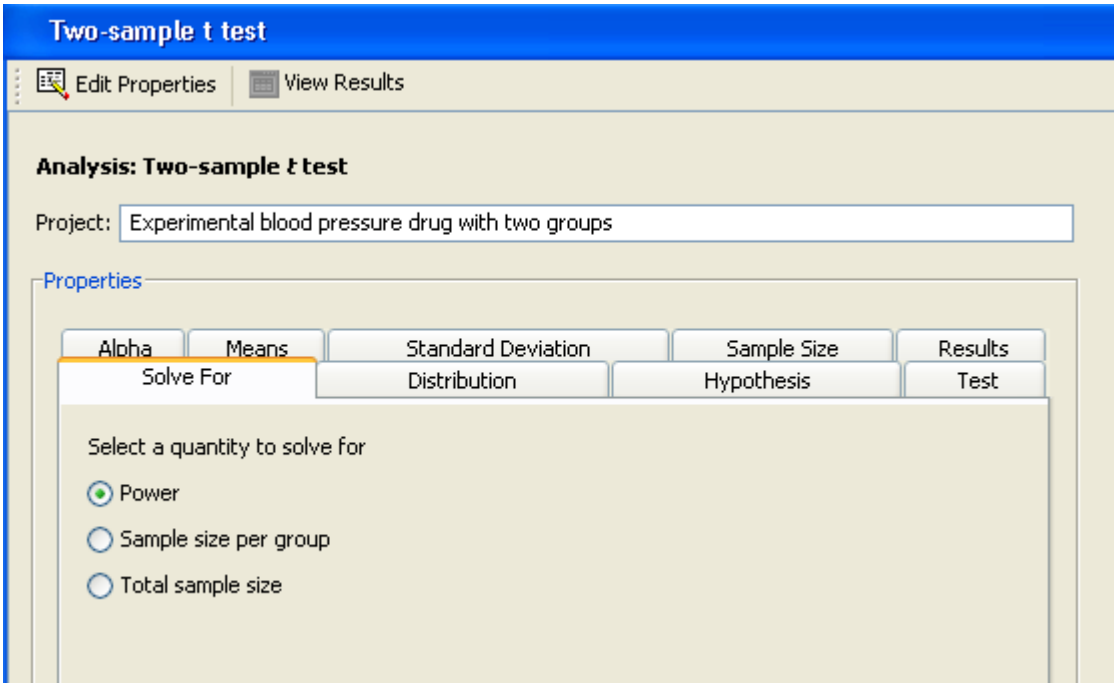
120 for the drug treatment group and that the standard deviation is 15 for both groups. You want to use a one-sided test with a significance level of 0.05. Because there are two independent groups and you are assuming that blood pressure is normally distributed, the two-sample *t* test is an appropriate analysis.

Start by creating a new project. Select **File►New**. In the New window, select **Two-sample *t* test** from the list. The Two-Sample *t* test project window appears, with the Edit Properties page displayed.

Editing Properties

On this page enter a name to describe the project and enter project properties. Click each tab on the Edit Properties page to enter the desired properties. You can also change tabs by clicking the **Next tab** or **Previous tab** buttons. See Figure 68.3.

Figure 68.35 Two-Sample *t* Test



Project Description

The description is used to identify this particular project in the Open and Delete windows. Type a description for your project in the **Project:** text box.

For this example, change the description to `Experimental blood pressure drug with two groups`, as shown in Figure 68.35.

Solve For

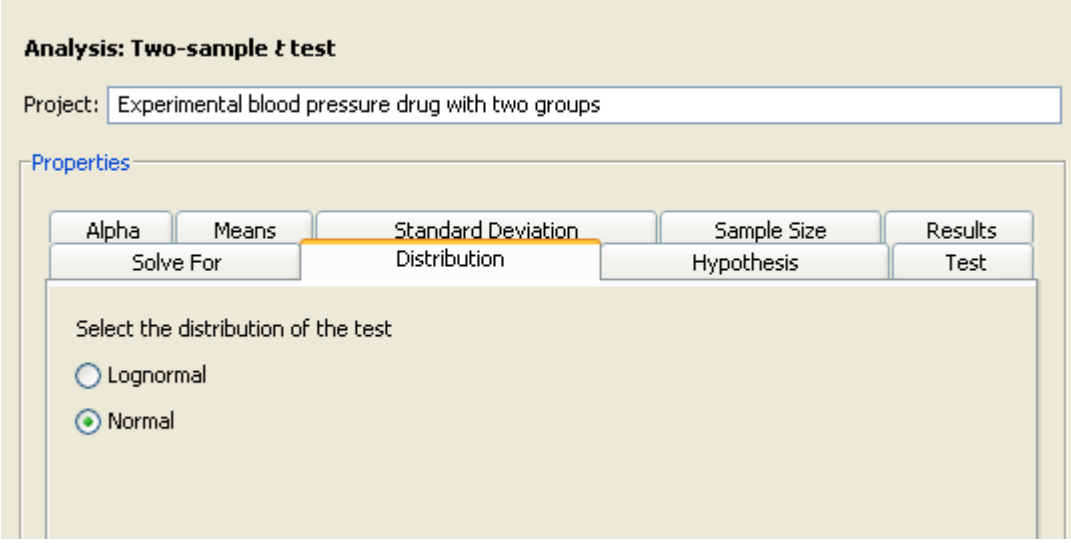
For the two-sample *t* test analysis, you can choose to solve for power, sample size per group, or total sample size. Specify the desired quantity type on the **Solve For** tab.

Click the **Solve For** tab and select the **Power** option as shown in Figure 68.35. For information about solving for sample size, see the section "Solving for Sample Size" on page 5108.

Distribution

Click the **Distribution** tab to select a distribution option that specifies the underlying distribution for the test statistic, as shown in Figure 68.36.

Figure 68.36 Distribution Tab

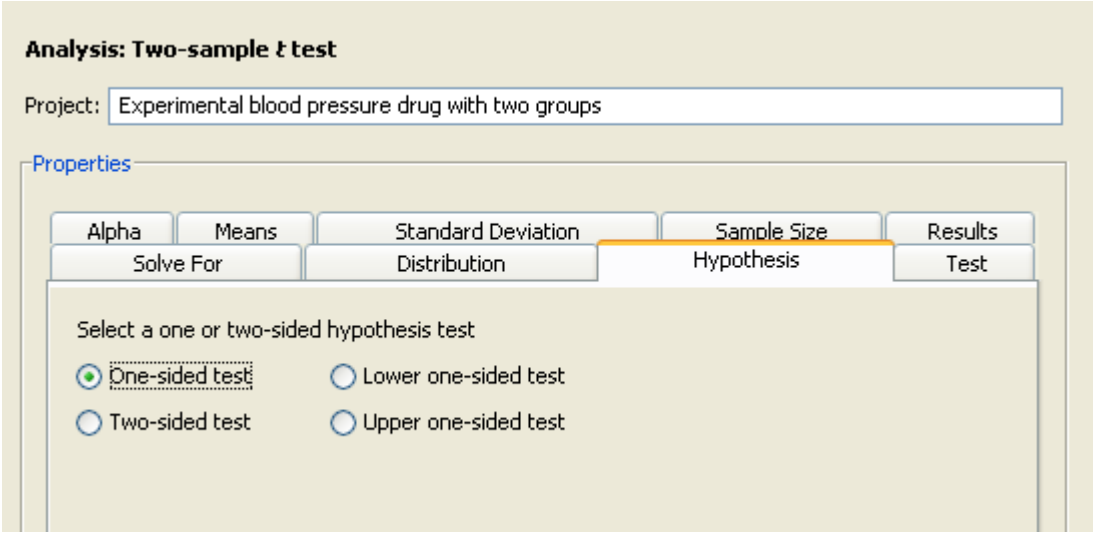


For this example, you are interested in means rather than mean ratios, so select the **Normal** option.

Hypothesis

Click the **Hypothesis** tab to select the type of test; see Figure 68.37.

Figure 68.37 Hypothesis Tab



You can choose either a one- or two-sided test. If you do not know the direction of the effect (that is, whether it is positive or negative), the two-sided test is appropriate. If you know the effect's direction, the one-sided test is appropriate. For the one-sided test, the alternative hypothesis is assumed to be in the same direction as the effect. If you specify a one-sided test and the effect is in the unexpected direction, the results of the analysis are invalid.

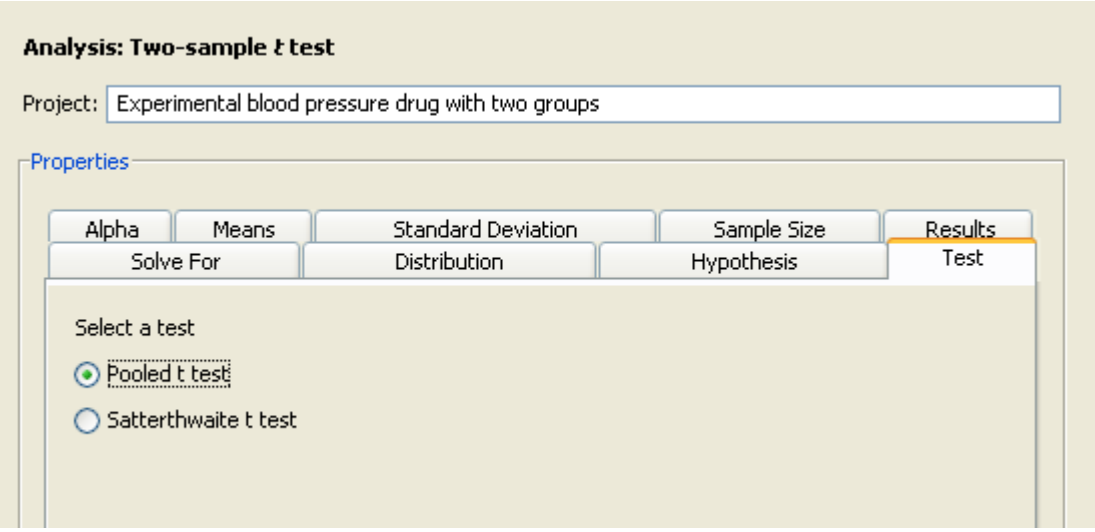
The **One-sided test** option assumes that you know the correct direction of the test. Select the **Lower one-sided test** and **Upper one-sided test** options to explicitly indicate the direction of the one-sided test.

Because you are interested only in whether the experimental drug lowers blood pressure, select the **One-sided test** option on the **Hypothesis** tab.

Test

Click the **Test** tab to select either the pooled t test or the Satterthwaite t test.

Figure 68.38 Test Tab

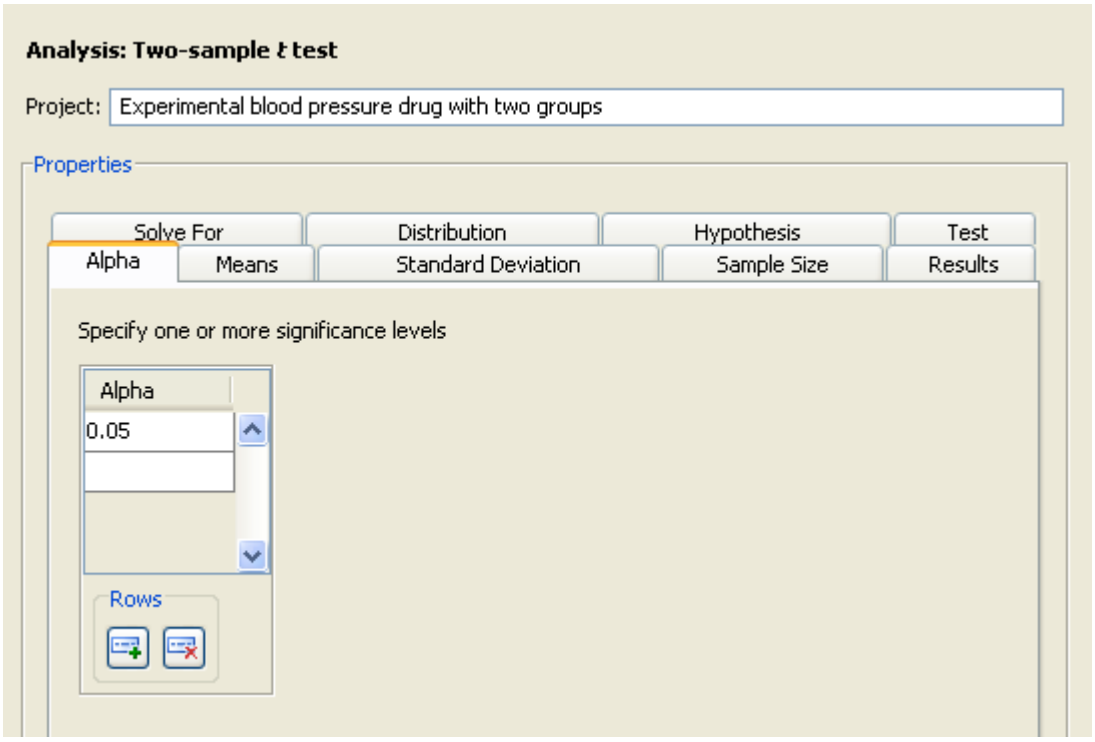


With the independent variances that the example uses, select **Pooled *t* test** option. The Satterthwaite *t* test is used with unequal variances; it is available only with the normal distribution.

Alpha

Click the **Alpha** tab to specify one or more significance levels, as shown in Figure 68.39.

Figure 68.39 Alpha Tab



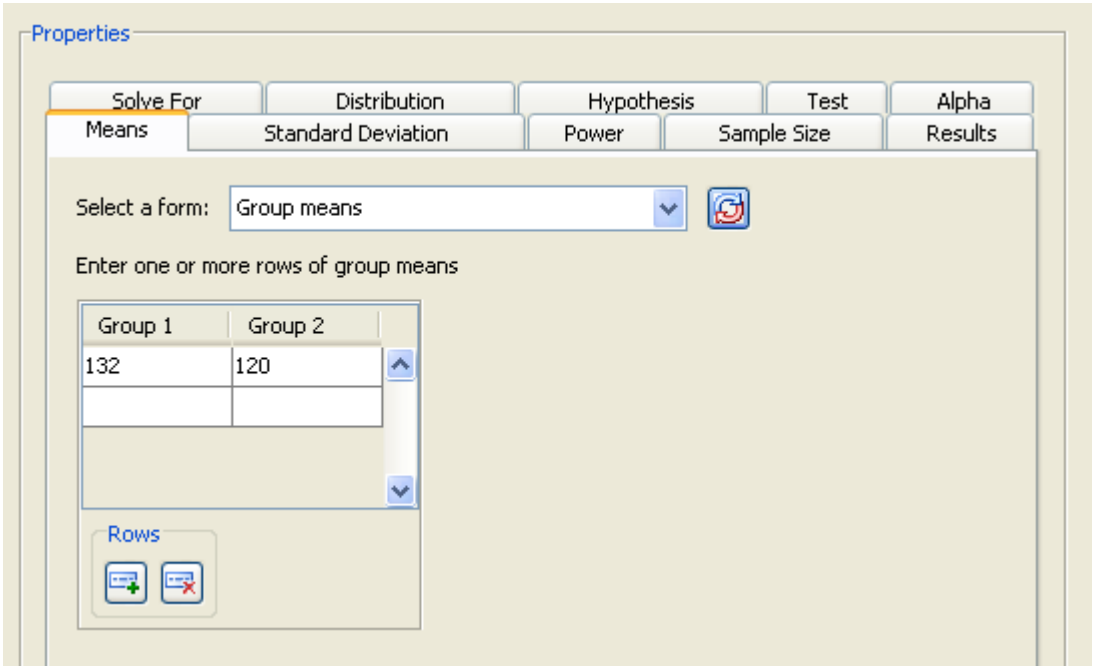
Alpha is the significance level (that is, the probability of falsely rejecting the null hypothesis). If you frequently use the same values for alpha, set them as defaults in the Preferences window. See the section "Setting Preferences" on page 5072 for more information about setting preferences.

Type the desired significance level of 0.05 in the first cell of the Alpha table (if it is not already the default value).

Means

Click the **Means** tab to select one of four possible ways to enter the means and the null mean difference, as shown in Figure 68.40.

Figure 68.40 Means Tab



Select one of the following forms from the **Select A Form** list. The four available forms are:

Difference between means
: Enter the difference between the group means. The null mean difference is assumed to be 0.

Group means
: Enter the means for each group. The null mean difference is assumed to be 0. The difference is formed by subtracting the mean for group 1 from the mean for group 2.

Difference between means, Null difference
: Enter the difference between the group means and a null mean difference.

Group means, Null difference
: Enter the means for each group and a null mean difference. The difference is formed by subtracting the mean for group 1 from the mean for group 2.

For this analysis, you can enter the means for the two groups either individually or as a difference. If your null mean difference is not zero, enter that value in the Null Mean table. (The Null Mean table is displayed only for the **Group means, Null Difference** and **Difference between means, Null difference** forms.)

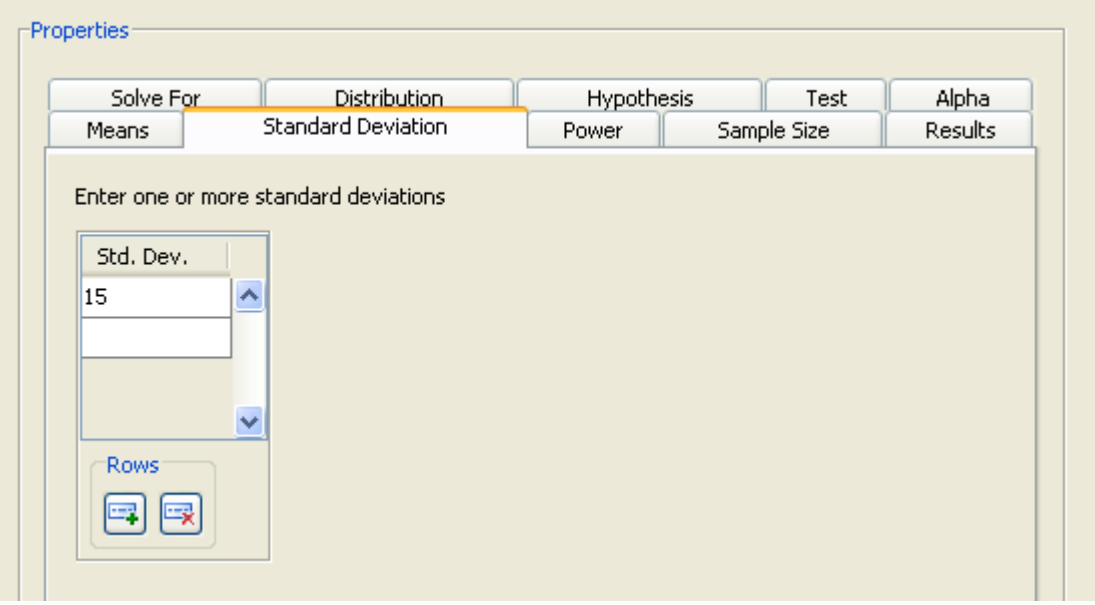
For this example, a null mean difference of 0 is reasonable, so select the **Group means** form from the list, as shown in Figure 68.40. Enter the control mean of `132` in the first row of the first column and the experimental mean of `120` in the first row of the second column.

Standard Deviation

Click the **Standard Deviation** tab to enter the standard deviation for the two groups. It is assumed to be equal for both groups.

For the example, enter a single value of `15`, as shown in Figure 68.41.

Figure 68.41 Standard Deviation Tab



Sample Size

Click the **Sample Size** tab to select one of three possible ways to enter the sample sizes, as shown in Figure 68.40.

Select one of the following forms from the **Select A Form** list:

Sample size per group
: Enter the sample size for one of the two groups. The group sizes are assumed to be equal.

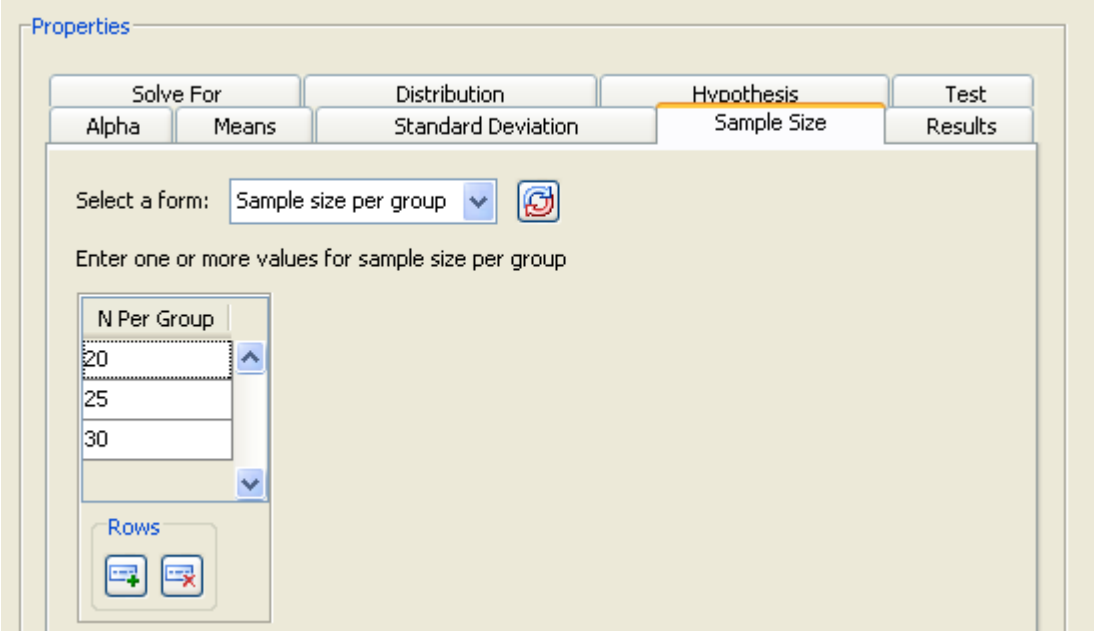
Group sample sizes
: Enter the sample size for each of the two groups. The group sizes can be equal or unequal.

Total N, Group weights

Enter the total sample size for the two groups and the relative sample sizes for each group. For more information about using relative sample sizes, see the section "Using Unequal Group Sizes" on page 5109.

Examine the alternatives by clicking the **Select a form** down arrow. For this example, select the **Sample size per group** form. You want to examine a curve of powers in the power by sample size graph, so enter the values `20`, `25`, and `30` in the Sample Size table, as shown in Figure 68.42. If you need to add more rows to the table, add them by clicking the button beneath the table.

Figure 68.42 Sample Size Tab



Summary of Properties

Table 68.2 contains the values of the input parameters for the example.

Table 68.2 Summary of Input Properties

Parameter	Value
Solve for	Power
Distribution	Normal
Hypothesis	One-sided test
Test	Pooled t test
Alpha	0.05
Means form	Group means
Means	132, 120
Standard deviation	15
Sample size form	Sample size per group
Sample size	20, 25, 30

Results

Click the **Results** tab to request desired results. Summary table and power by sample size graph options are available.

For the example, select the **Create summary table** and **Create power by sample size graph** check boxes.

Click **Calculate** to perform the analysis. If there are no errors in the input values, the View Results page appears. If there are errors in the input parameter values, you are prompted to correct them.

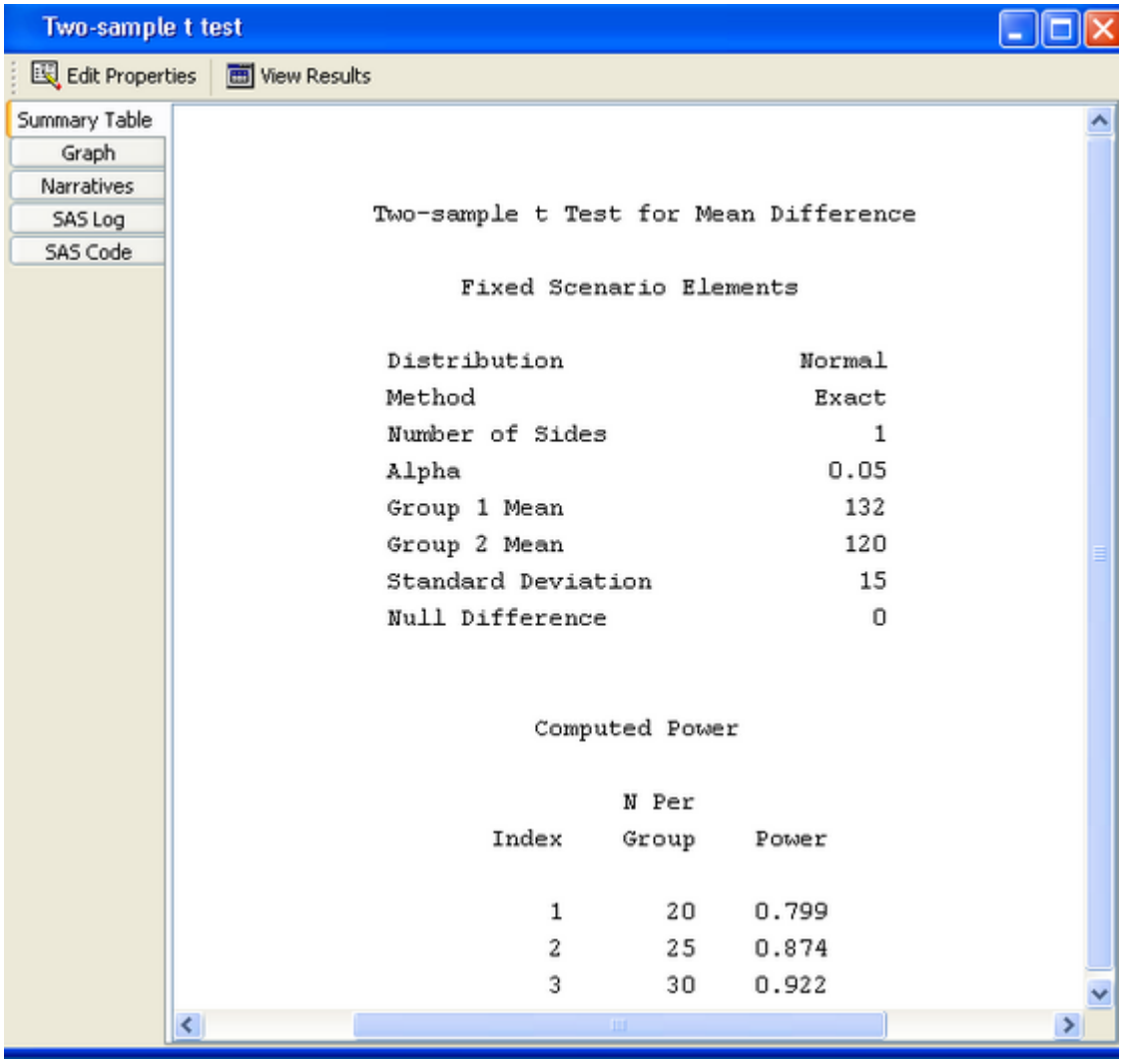
Viewing Results

The results are listed on separate tabs on the View Results page. Click the tab of each result that you want to view.

Summary Table

Click the **Summary Table** tab to view a table that includes the values of the input parameters and the computed quantity (in this example, power). See Figure 68.43.

Figure 68.43 Results Page with Summary Table

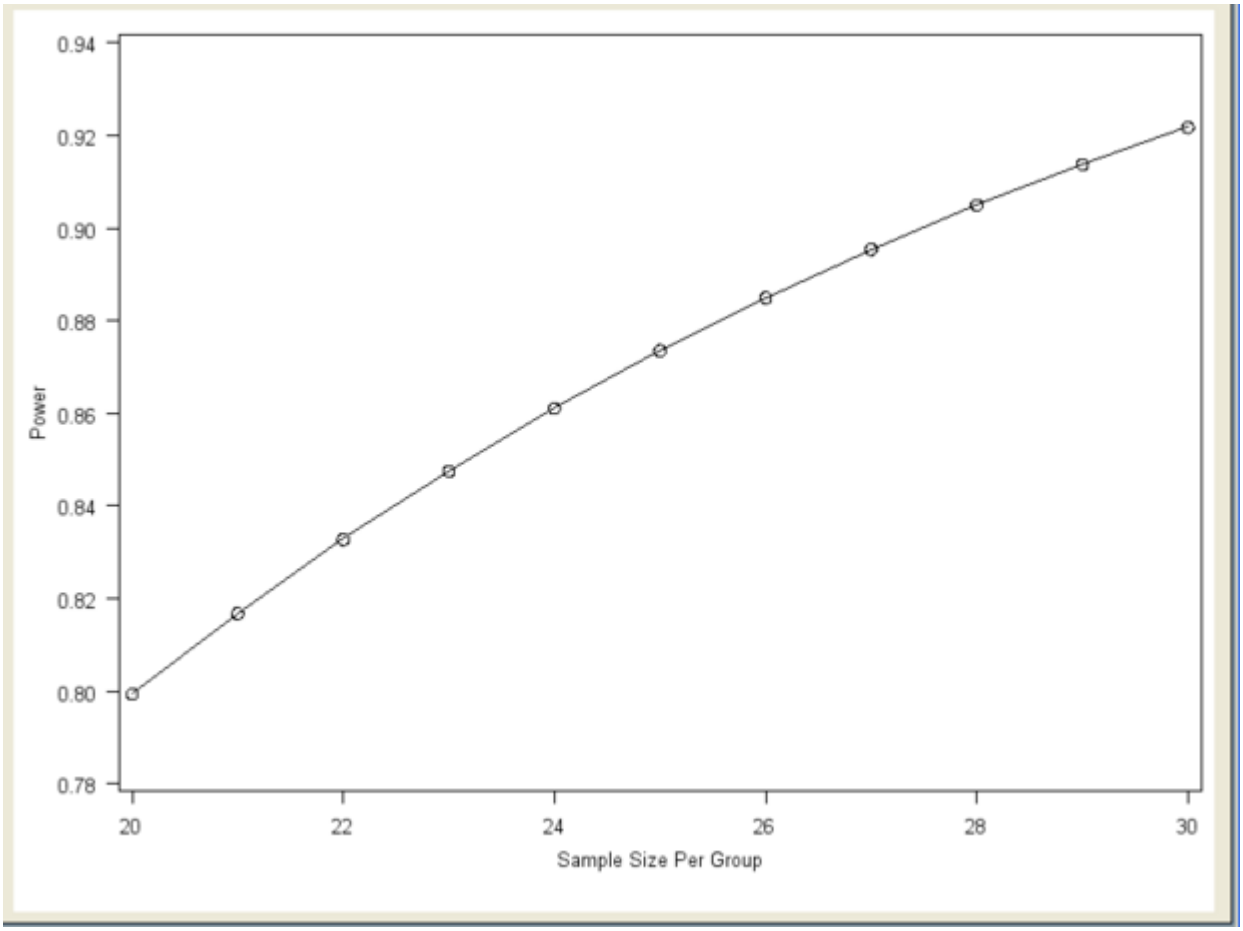


The table consists of two subtables: the `Fixed Scenario Elements` table that contains the input parameters that have only one value for the analysis, and the `Computed Power` table that contains the input parameters that have more than one value for the analysis and the corresponding power. Only the N per group parameter appears in the `Computed Power` table; all of the other input parameters have a single value. The computed power for a sample size per group of 25 is 0.874. Thus, you have a probability of 0.87 that the study will find the expected result if the assumptions and conjectured values are correct.

Power by Sample Size Graph

Click the **Graph** tab to view a power by sample size graph that displays power on the vertical axis and sample size per group on the horizontal axis. See Figure 68.44.

Figure 68.44 Power by Sample Size Graph

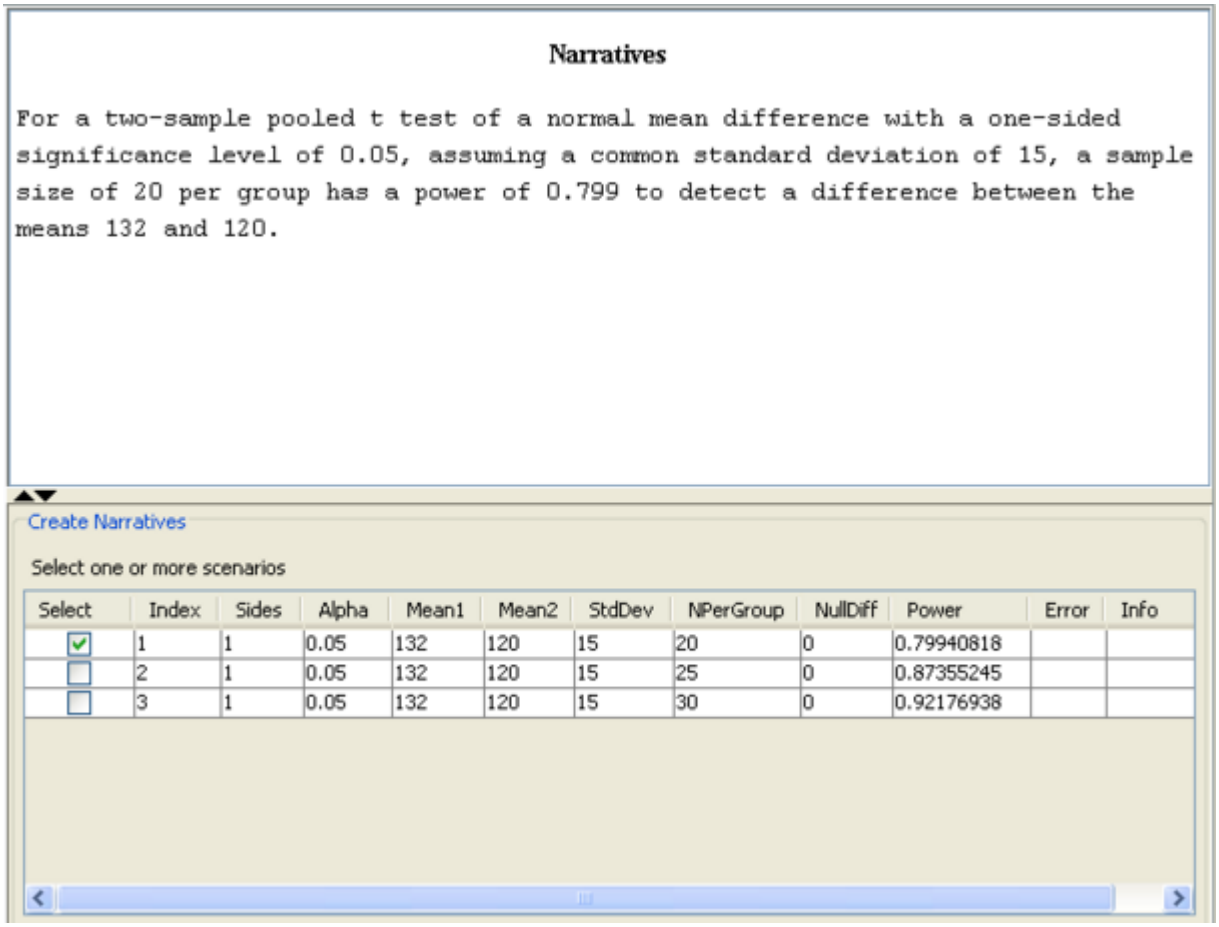


The range of values for the horizontal axis is 20 to 30, which were the smallest and largest values, respectively, that you entered on the **Sample Size** tab. You can customize the graph by specifying the values for the sample size axis (see the section "Customizing Graphs" on page 5079).

Narratives

Click the **Narratives** tab to create and display a sentence- or paragraph-length text summary of the input parameter values and the computed quantity for combinations of the input parameter values; see Figure 68.45.

Figure 68.45 Narrative Selector and Display



To create a narrative, selected the desired scenario (row) in the narrative selector table at the bottom of the **Narratives** tab.

In this example, select the narrative for the sample size per group of 20, which yields a power of 0.799. The following text summary is displayed:

> For a two-sample pooled *t* test of a normal mean difference with a one-sided significance level of 0.05, assuming a common standard deviation of 15, a sample size of 20 per group has a power of 0.799 to detect a difference between the means 132 and 120.

To create other narratives, select the desired rows in the narrative selector table. If you also select the second row for the sample size of 25, another text summary is displayed below the first one:

> For a two-sample pooled *t* test of a normal mean difference with a one-sided significance level of 0.05, assuming a common standard deviation of 15, a sample size of 20 per group has a power of 0.799 to detect a difference between the means 132 and 120.

```
For a two-sample pooled t test of a normal mean difference with a
one-sided significance level of 0.05, assuming a common standard
deviation of 15, a sample size of 25 per group has a power of
0.874 to detect a difference between the means 132 and 120.
```

To change some values of the analysis and rerun it, select the Edit Properties page, change the desired properties, and click the **Calculate** button again.

Test of Two Independent Means for Unequal Variances

In the preceding example, you assumed that the population standard deviations were equal. If you believe that the population standard deviations are not equal, use the same two-sample *t* test analysis as with the preceding example, but change the test option and enter group standard deviations.

You can use the previous example to demonstrate this test. If the project is not already open, open it by selecting **File►Open** on the menu bar, and then selecting the project that you have been using.

Make a copy of the project by selecting **File►Save As**. Enter a different project description, `Experimental blood pressure drug with two groups for unequal variances`. Click **OK**.

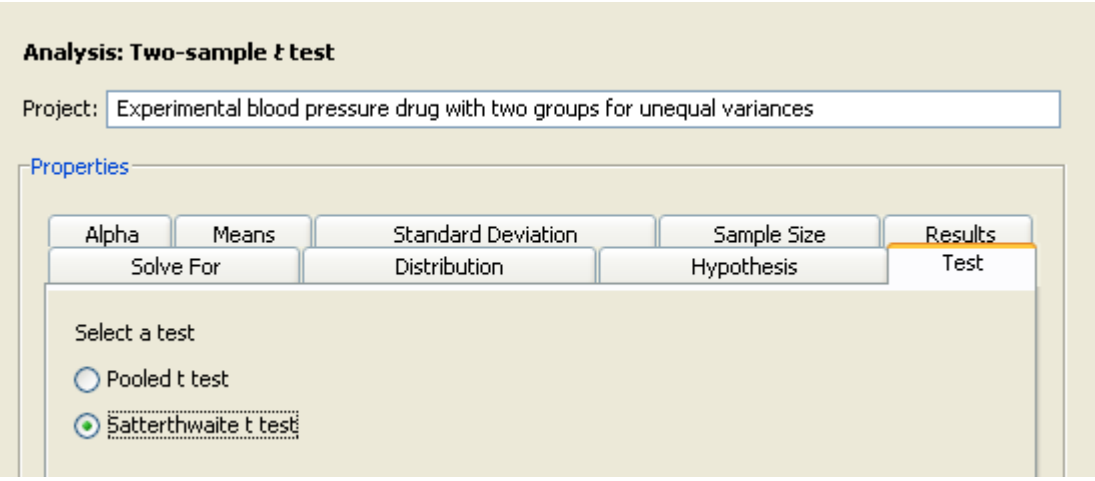
The copy of the project is opened, and the current project is closed.

Editing Properties

Test

On the **Test** tab of the copied project, change the test to **Satterthwaite t test**, as shown in Figure 68.46.

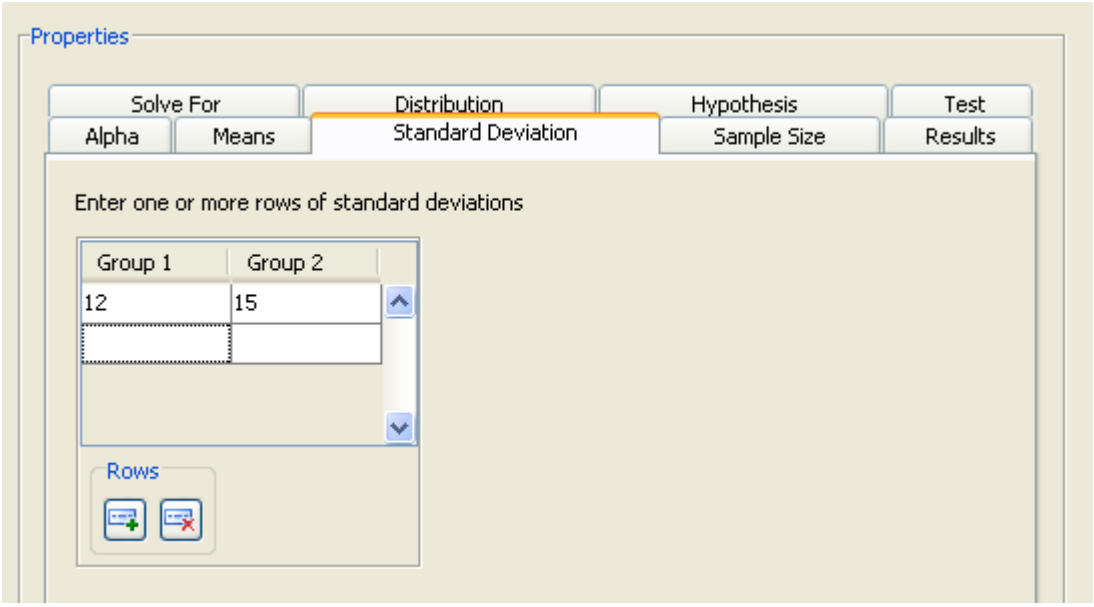
Figure 68.46 Satterthwaite *t* Test Option



Specifying Group Standard Deviations

Click the **Standard Deviation** tab and enter the group standard deviations of 12 and 15 on a single row, as shown in Figure 68.47.

Figure 68.47 Group Standard Deviations



Summary of Input Parameters

Table 68.3 contains the values of the input parameters for the example.

Table 68.3 Summary of Input Parameters

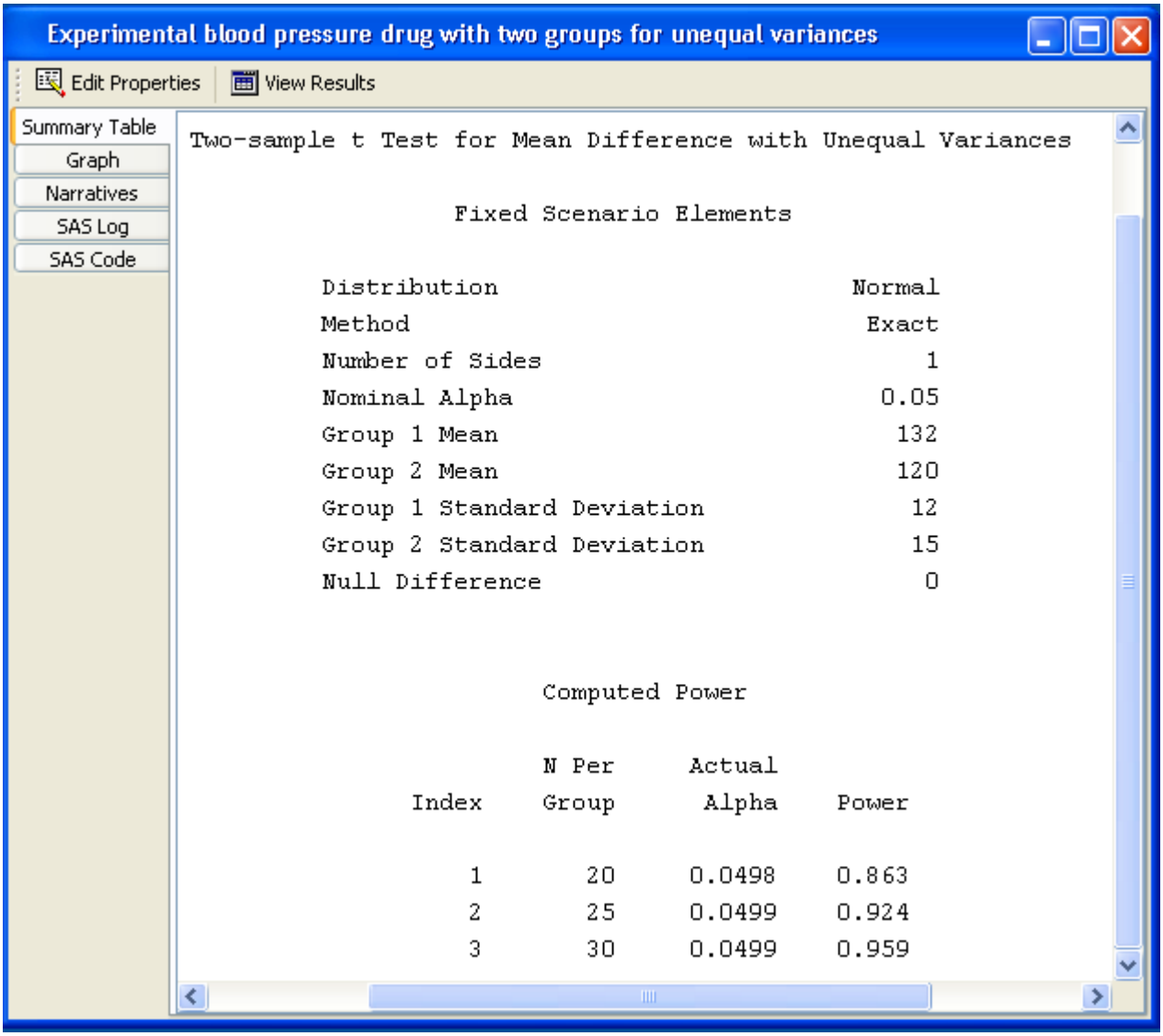
Parameter	Value
Distribution	Normal
Hypothesis	One-sided test
Test	Satterthwaite t test
Alpha	0.05
Means form	Group means
Means	132, 120
Standard deviation	12, 15
Sample size form	Sample size per group
Sample size	20, 25, 30

Click **Calculate** to run the analysis.

Viewing Results

The power for a sample size per group of 25 is 0.924, as shown in Figure 68.48. Notice that the actual alpha is 0.0499. This is because the Satterthwaite *t* test is (slightly) biased.

Figure 68.48 Satterthwaite Test Results



If you modified the previous example, when you select the **Narratives** tab, the following message is displayed:

```
Previously selected narratives have been cleared because one or more
input parameter values have changed.
```

In the previous analysis, you created narratives for two scenarios. Because this analysis uses group standard deviations, those selected narratives were cleared. The message would also have appeared if you had changed the number of scenarios.

Use the narrative selector table to create other narratives.

Test of Mean Ratios

Instead of comparing means for a control and drug treatment group, you might want to investigate whether the blood pressure of the treatment group is lowered by a given percentage of the control group, say 10 percent. That is, you expect the ratio of the treatment group to the control group to be 90% or less.

PSS provides a two-sample *t* test of a mean ratio when the data are lognormally distributed.

For mean ratios, the coefficient of variation (CV) is used instead of standard deviation. In this example, you can expect the CV to be between 0.5 and 0.6. You also want to compare an equally weighted sampling of groups with an overweighted sampling in which the control group contains twice as many subjects as the treatment group: 50 and 25, respectively.

Make a copy of the project by selecting **File►Save As**. Enter a different project description, `Percent improvement with blood pressure drug`.

The copy of the project is opened.

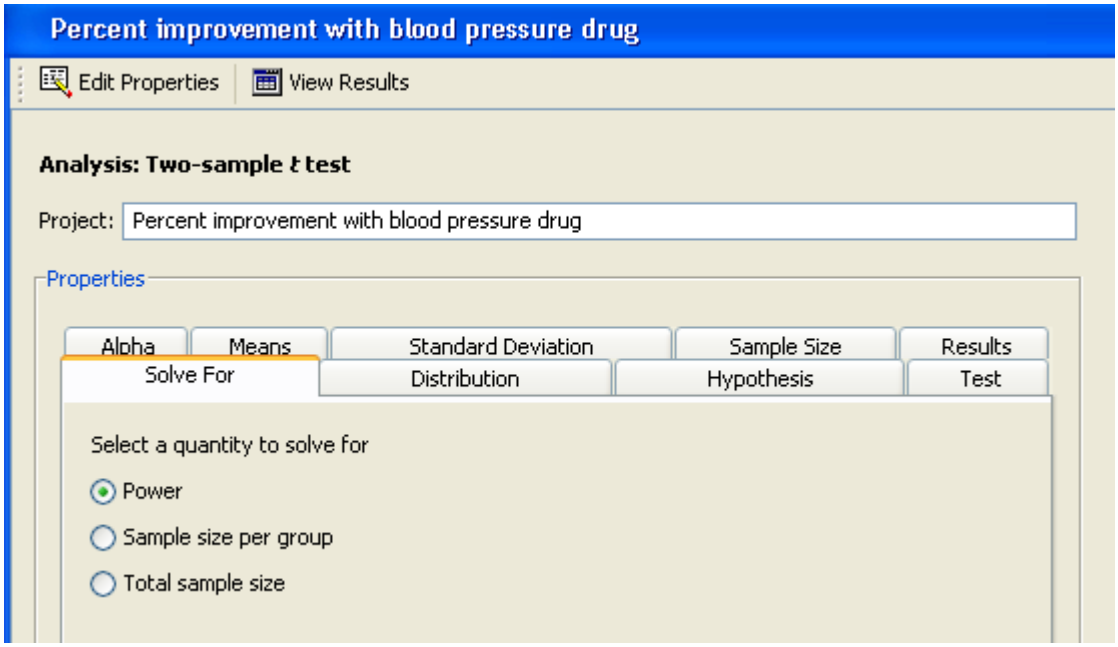
Editing Properties

Several of the input parameters for the test of mean ratios differ from the ones described in the section "Test of Two Independent Means for Equal Variances" on page 5086. Mean ratios and coefficients of variation are used instead of mean differences and standard deviations. These two parameters are discussed in detail in this section. For the input parameters and options that have been discussed previously in this example, only the values for this example are given.

Solve For Tab

Click the **Solve For** tab to select the **Power** option as the quantity to be solved for, as shown in Figure 68.49.

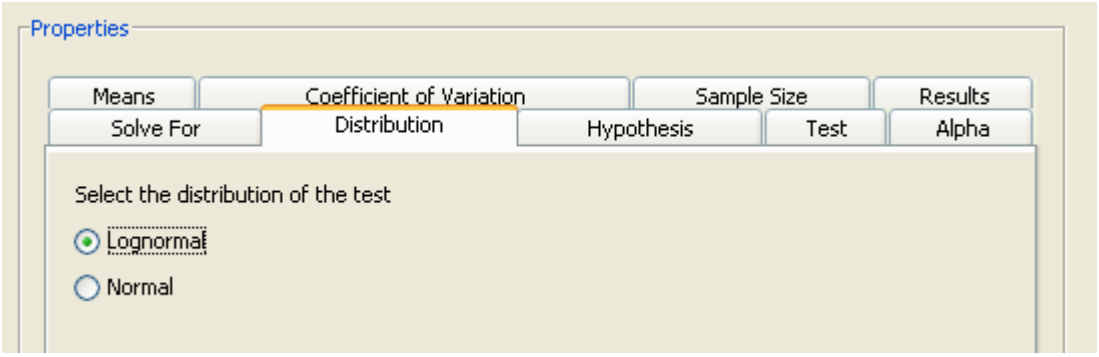
Figure 68.49 Project Description, Solve for Tab



Distribution

You are interested in mean ratios rather than means, so select the **Lognormal** option on the **Distribution** tab, as shown in Figure 68.50.

Figure 68.50 Distribution Tab with Lognormal Option



Hypothesis and Alpha

Click the **Hypothesis** tab and select the **One-sided test** option.

Click the **Alpha** tab and type 0.05 as the significance level in the first cell of the table, if it is not already there.

Means

Click the **Means** tab to select the input form for entering mean ratios. There are four alternate forms for entering means or mean ratios:

Mean ratio
: Enter the ratio of the two group means—that is, the treatment mean divided by the reference mean. The null ratio is assumed to be 1.

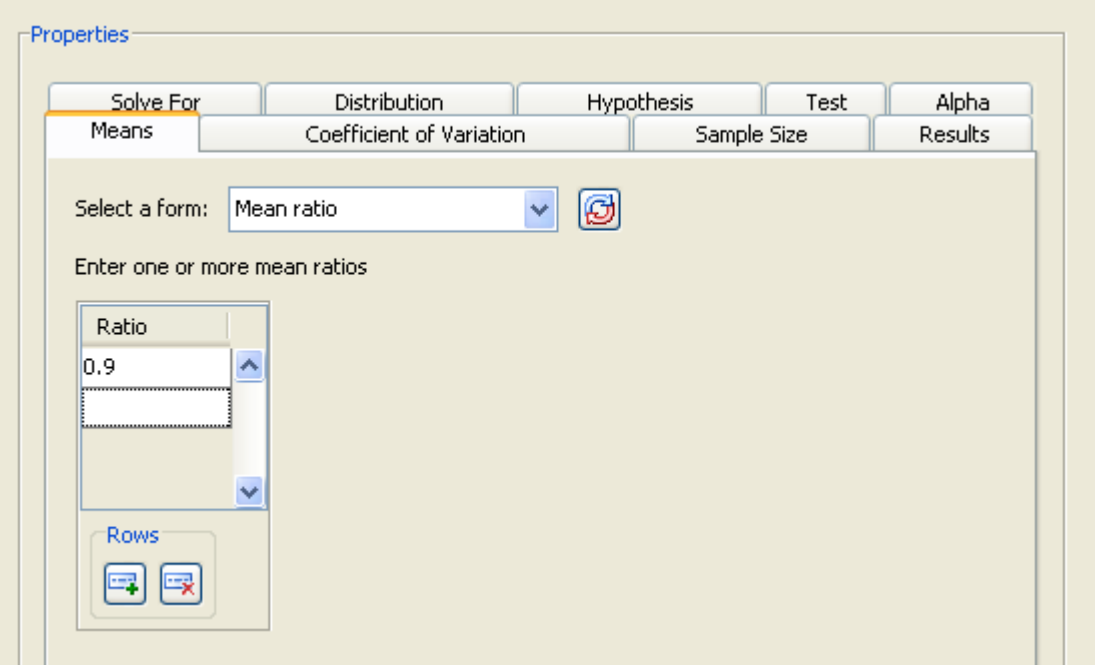
Group means
: Enter the means for each group. The ratio of the means is formed by dividing the mean for group 2 by the mean for group 1. The null ratio is assumed to be 1.

Mean ratio, Null ratio
: Enter the ratio of the two group means—that is, the treatment mean divided by the reference mean. Enter the null ratio.

Group means, Null ratio
: Enter the means for each group. The ratio of the means is formed by dividing the mean for group 2 by the mean for group 1. Enter the null ratio.

As shown in Figure 68.51, select the **Mean ratio** form which uses a default null ratio of 1. Enter a single mean ratio value of 0.9.

Figure 68.51 Means Tab with Mean Ration Form and Values

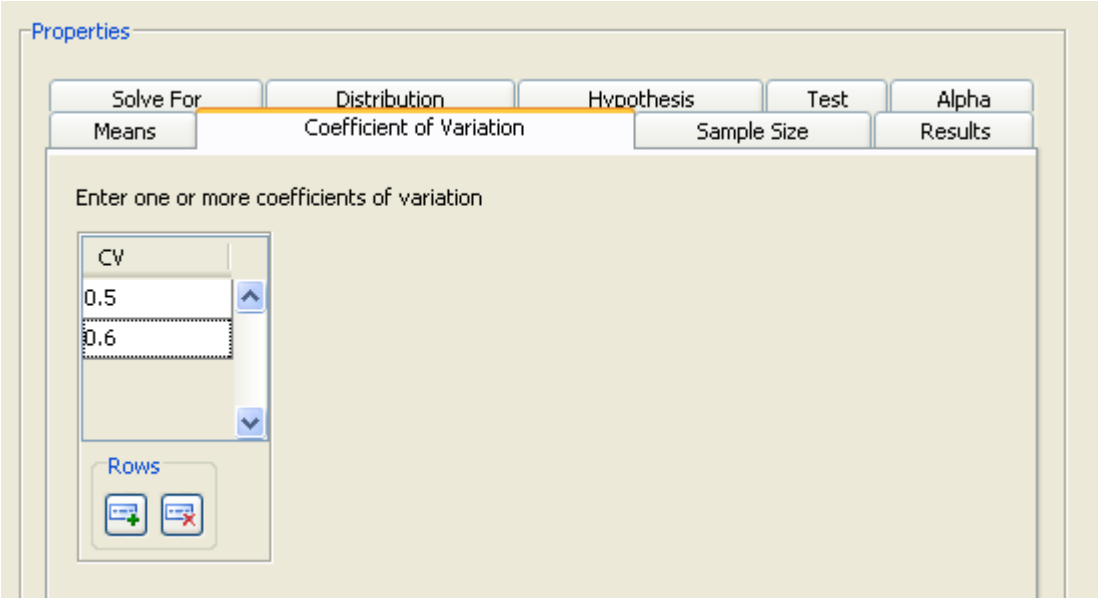


Coefficient of Variation

On the **Coefficient of Variation** tab, enter the coefficients of variation. They are assumed to be equal for the two groups.

For this example, enter 0.5 and 0.6, as shown in Figure 68.52.

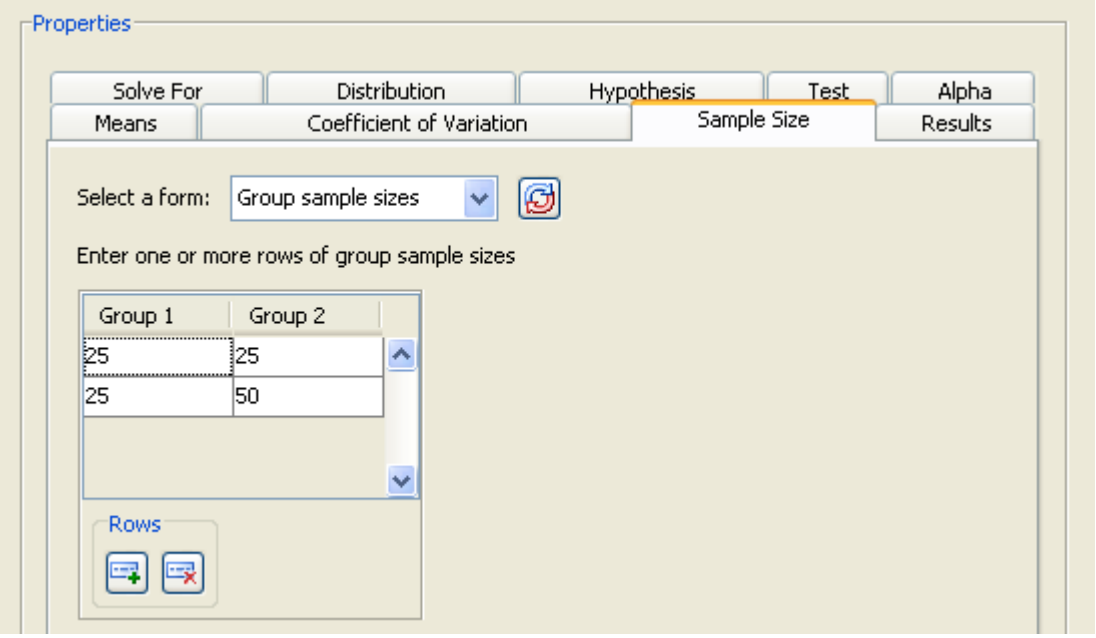
Figure 68.52 Coefficient of Variation Tab



Sample Size

On the **Sample Size** tab, select the **Group sample sizes** form and enter two sets of values: 25 and 25 in the first row and 25 and 50 in the second row, as shown in Figure 68.53.

Figure 68.53 Sample Sizes



Summary of Input Parameters

Table 68.4 contains the values of the input parameters for the example.

Table 68.4 Summary of Input Parameters

Parameter	Value
Hypothesis	One-sided test
Distribution	Lognormal
Alpha	0.05
Means form	Mean ratio
Mean ratio	0.9
Coefficients of variation	0.5, 0.6
Sample size form	Group sample sizes
Sample Size	(25, 25), (25, 50)

Results

On the **Results** tab, select the **Create summary table** and **Create power by sample size graph** check boxes.

Click **Calculate** to perform the analysis.

In this case, the following message is displayed:

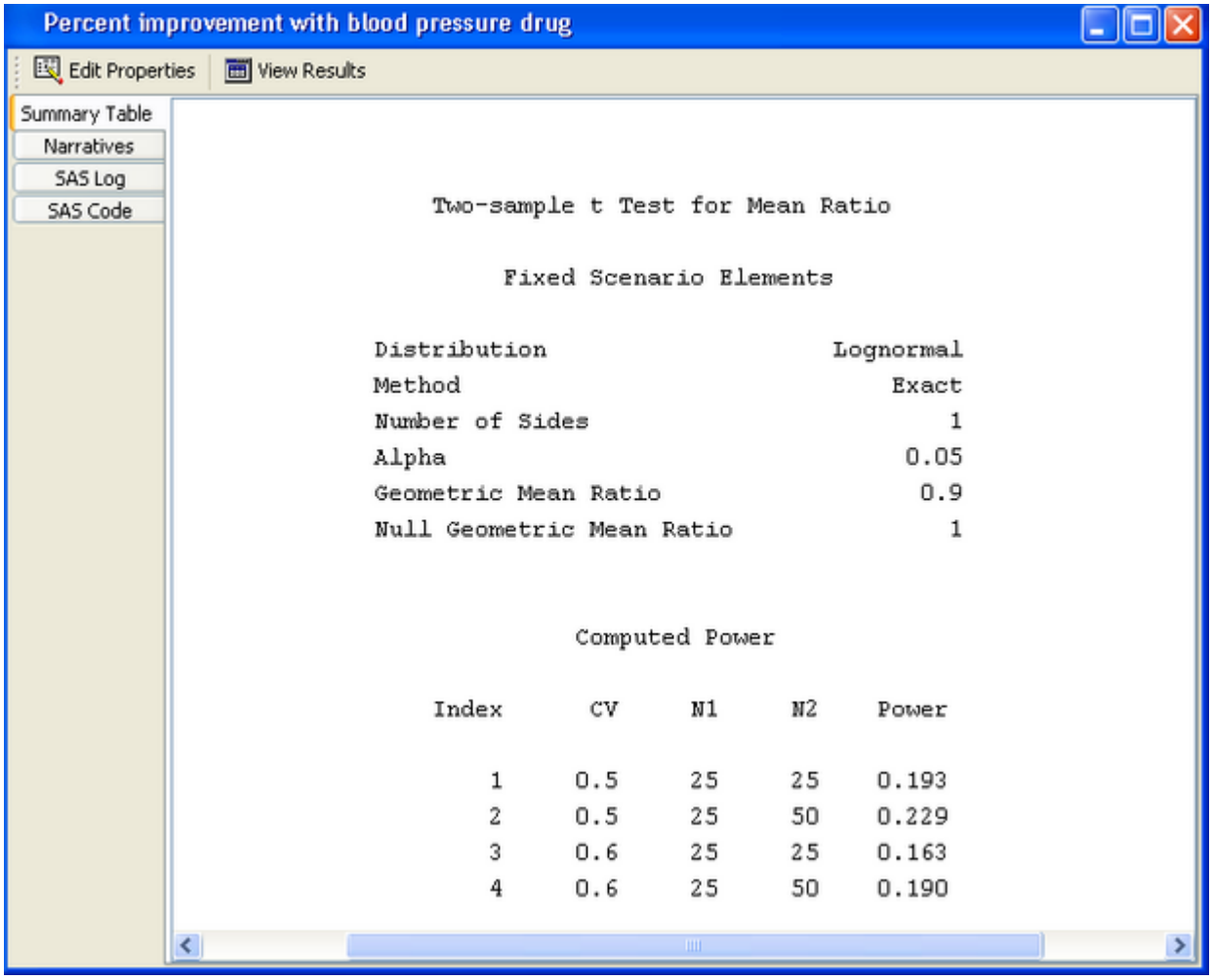
```
The power by sample size graph is not available when specifying sample
sizes for two groups.
```

If you want a power by sample size graph, you can choose to plot total sample size instead by using the `Total N, Group weights` sample size form on the **Sample Size** tab. For more information about using this input form, see the section "Using Unequal Group Sizes" on page 5109.

Viewing Results

The first thing that you notice from the summary table in Figure 68.54 is that the calculated powers are quite low—they range from 0.163 to 0.229. You have less than a 25% probability of detecting the difference that you are looking for. Clearly, this set of parameter values leads to insufficient power. To increase power, you might choose a larger sample size or a larger alpha.

Figure 68.54 Summary Table



You can also see that oversampling the control group improves power slightly, 0.229 versus 0.193 for the coefficient of variation of 0.5. However, this is a marginal increase that is probably not worth the added expense.

For the example, use larger sample sizes with equal cell sizes. Return to the Edit Properties page by clicking the **Edit Properties** icon near the top of the window.

Then, on the **Sample size** tab, change to the **Sample size per group** form. Specify sample sizes of 50, 100, 150, and 200, as shown in Figure 68.55.

Figure 68.55 Modified Sample Size Values

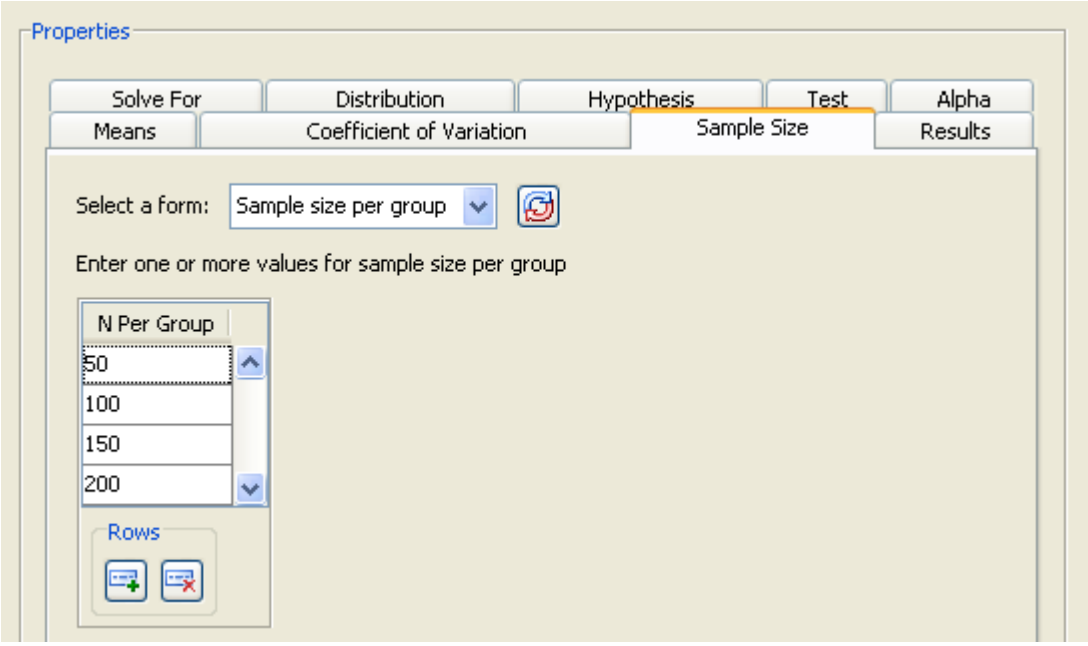


Table 68.5 contains the modified values of the input parameters for the example.

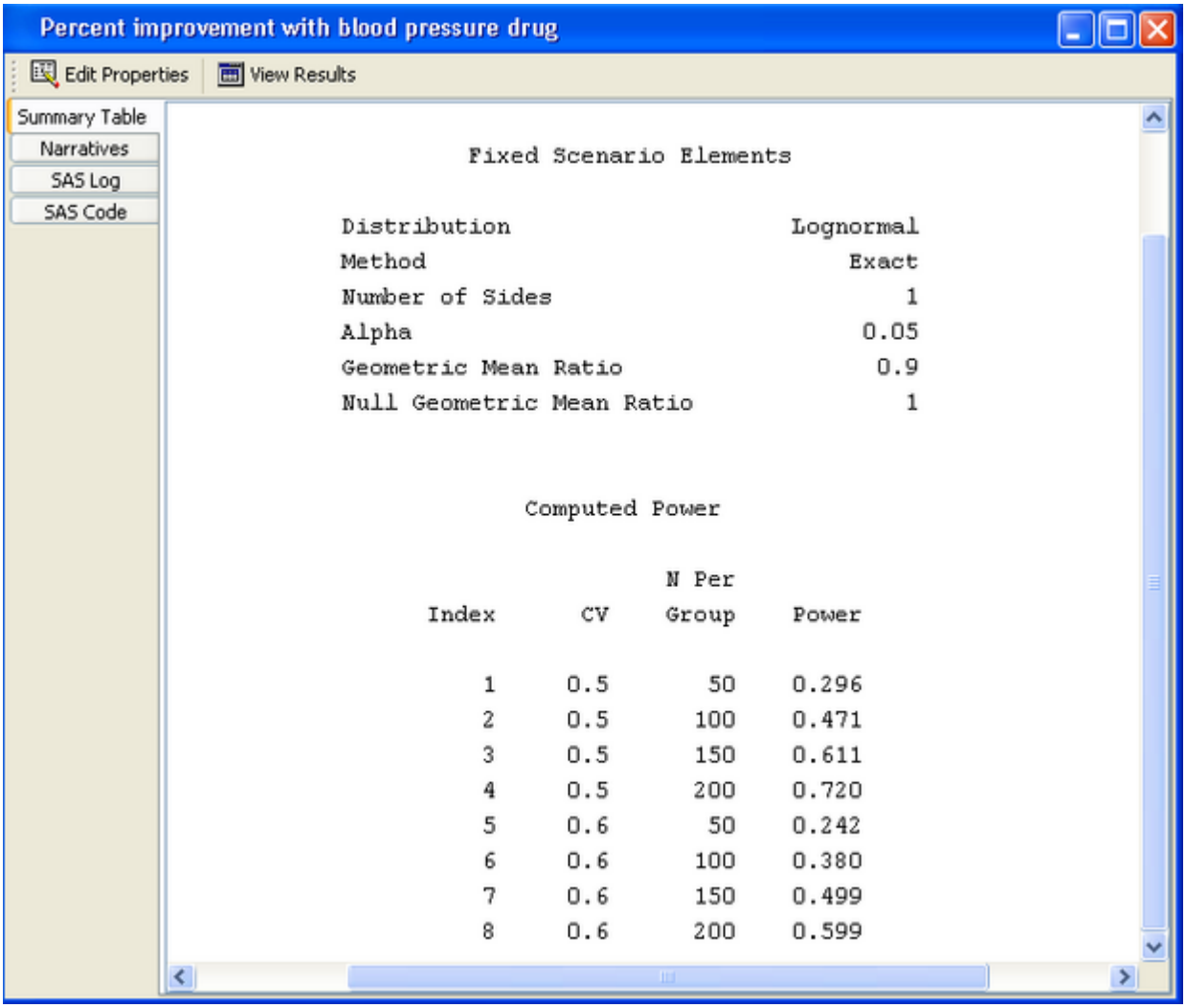
Table 68.5 Modified Summary of Input Parameters

Parameter	Value
Sample size form	Sample size per group
Sample size	50, 100, 150, 200

Rerun the analysis by clicking **Calculate**.

Figure 68.56 displays the summary table. The largest sample size of 200 (per group) yields a power of 0.72 for a coefficient of variation of 0.5, and 0.599 for one of 0.6. With a total of 400 subjects, you still have a 30% to 40% probability of not detecting the effect even if it exists.

Figure 68.56 Summary Table for Modified Sample Sizes

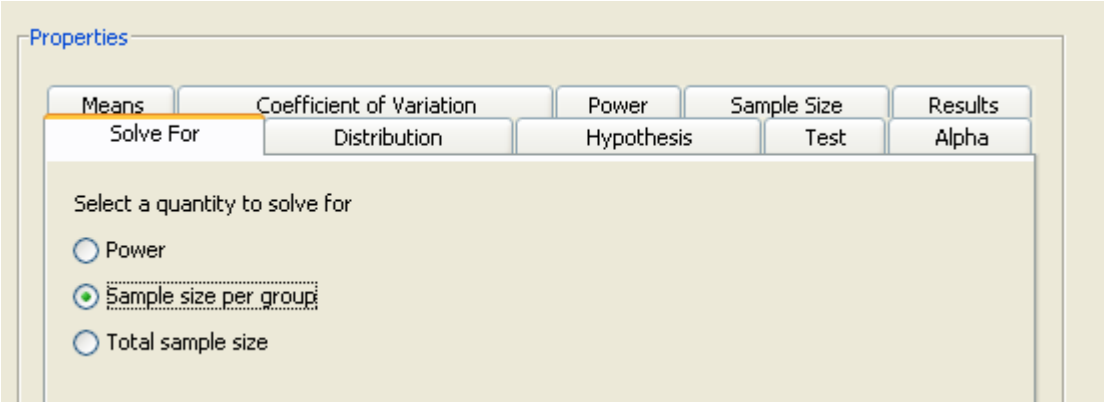


Additional Topics

Solving for Sample Size

Several types of analysis enable you to solve for either total sample size or sample size per group. The sample size per group choice assumes equal group sizes. When solving for total sample size, the group sizes can be equal or unequal. Select the desired quantity on the **Solve For** tab. An example of these options is shown in Figure 68.57.

Figure 68.57 Solve For Tab with Sample Size Selected



For either of the two sample size options, you must specify one or more values for power on the **Power** tab. If you frequently use the same values for power, set them as the default in the Preferences window, which is accessed by **Tools►Preferences**. Changing preferences affects only projects that you create after the change; existing projects are not affected.

If you select total sample size, you must specify whether the group sizes are equal or unequal. Select the appropriate option on the **Sample Size** tab. For unequal group sizes, you must specify the relative sample sizes for the two groups. For information about providing relative sample sizes, see the section "Using Unequal Group Sizes" on page 5109.

Using Unequal Group Sizes

When solving for either power or total sample size, you might have unequal group sizes. If so, you must provide relative sample sizes for the groups. Weights must be greater than 0 but do not have to sum to 1.

Select the **Total N, Group weights** form on the **Sample Size** tab. Enter total sample sizes of 30 and 60 in the **Total N** table. Select the **Unequal group sizes** option and click **Enter Relative Sample Sizes**, as seen in Figure 68.58.

Figure 68.58 Sample Size Tab with Group Weights Form

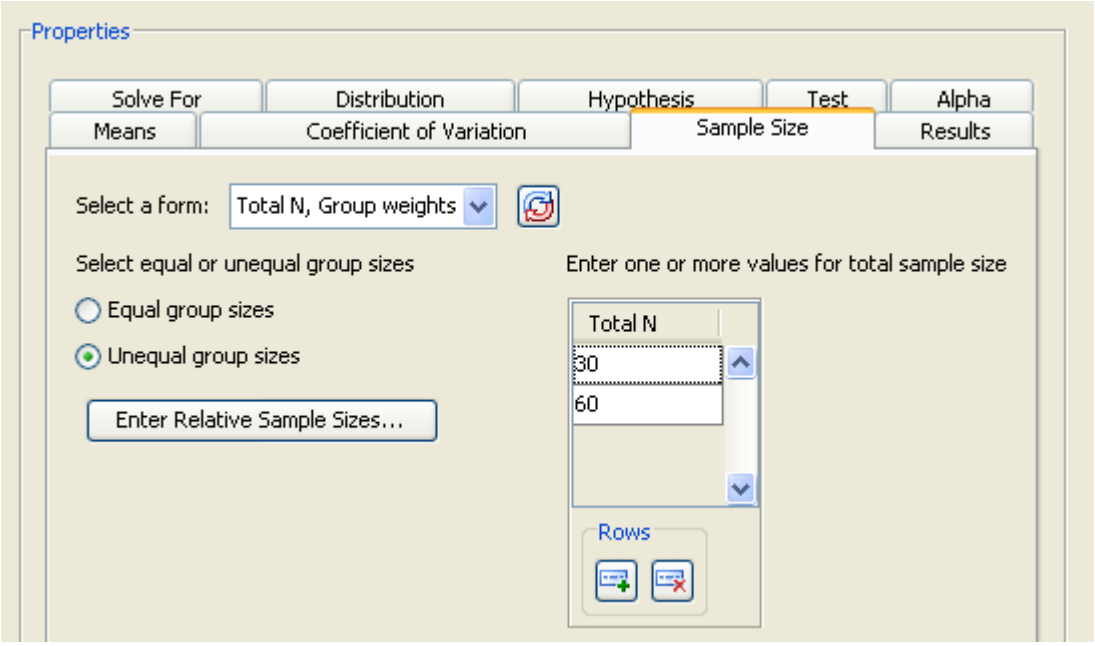
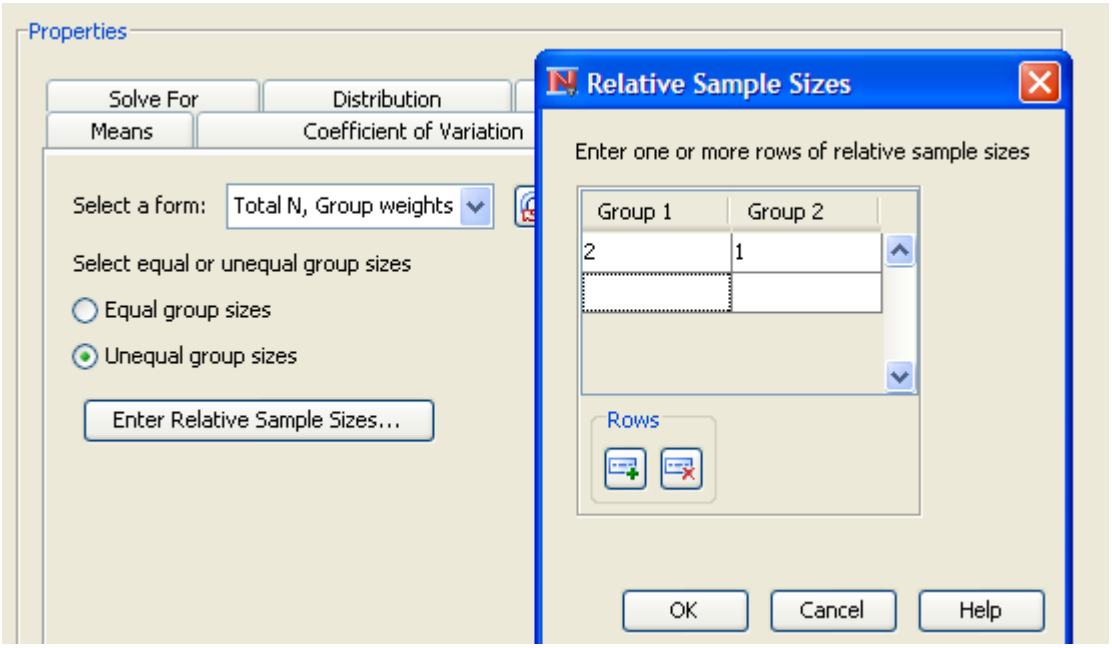


Figure 68.59 displays the window in which you can enter relative sample sizes. As an example, enter 2 for the first group and 1 for the second. In this case, you are sampling the drug treatment group twice as often as the control group.

Figure 68.59 Relative Sample Sizes Window



The weights control how the total sample size is divided between the two groups. In the example, the sample size for groups 1 and 2 is 20 and 10, respectively, for a total sample size of 30.

Click **OK** to save the values and return to the Edit Properties page.

Example: Analysis of Variance

Overview

PSS offers power and sample size calculations for analysis of variance in two tasks: one-way ANOVA and general linear univariate models. Optional contrasts are available in both tasks.

In the one-way ANOVA task, you can solve for sample size per group as well as total sample size. The contrast facility for the one-way ANOVA task enables you to select orthogonal polynomials as well as to specify contrast coefficients. For more information about power and sample size analysis for one-way ANOVA, see Chapter 67, "The POWER Procedure."

In the general linear univariate models task, you specify linear models for a single dependent variable. Type III tests and contrasts of fixed effects are included, and the model can include covariates. For more information about power and sample size analysis for linear univariate models, see Chapter 41, "The GLMPOWER Procedure."

The Example

Suppose you are interested in testing how two experimental drugs affect systolic blood pressure relative to a standard drug. You want to include both men and women in the study. You have a two-factor design: a drug factor with three levels and a gender factor with two levels. You choose a main-effects-only model because you do not expect a drug by gender interaction. You want to calculate the sample size that will produce a power of 0.9 using a significance level of 0.05. You believe that the error standard deviation is between 5 and 7 mm pressure. This is a two-way analysis of variance, so the general linear univariate models task is the appropriate one.

Editing Properties

Start by opening the New window (**File►New**). In the **Analysis of Variance and Linear models** section of the New window, select **General linear univariate models**. The **General univariate linear models** project appears, with the Edit Properties page displayed.

Project Description

For the example, change the project description to `Three blood pressure drugs and gender`.

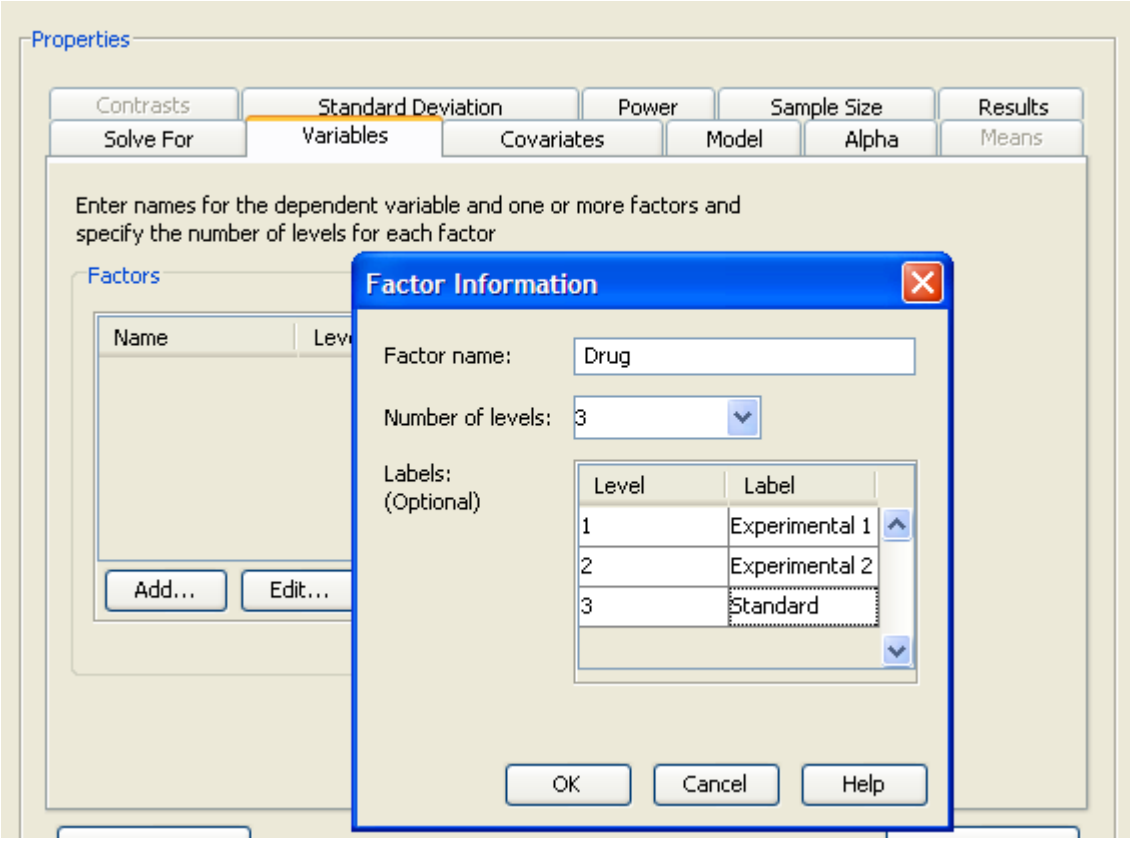
Solve For

Click the **Solve For** tab and select the **Sample size** option.

Variables

Click the **Variables** tab to enter the names of the factors in the design. Click the **Add** button. The Factor Definition window appears, as shown in Figure 68.60.

Figure 68.60 Factor Definition Window



Enter the name for the first factor, `Drug`, and enter the number of factor levels in the **Number of levels:** list box. There are three levels for this factor. Optionally, you can provide a label for each factor level. This label is used to identify factor levels on other tabs of the Edit Properties page. For this example enter the labels `Experimental 1`, `Experimental 2`, and `Standard` for the three levels of the `Drug` factor. Click **OK** when you are finished.

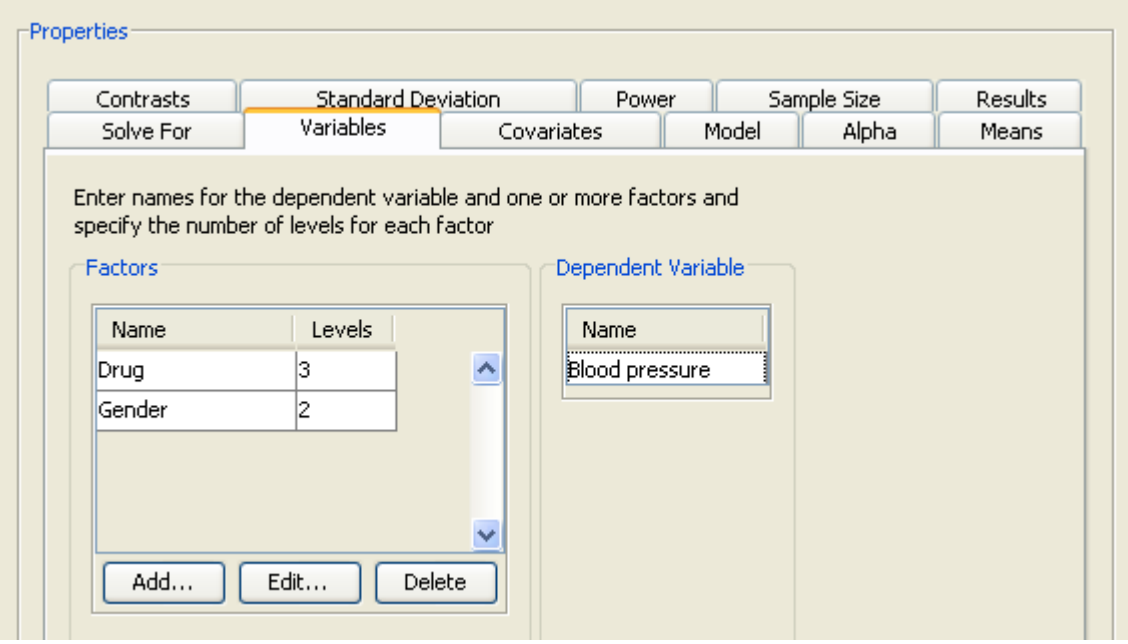
Click the **Add** button again and repeat the process for the second factor, `Gender` with two levels and labels `Female` and `Male`.

Factors can contain blanks and other special characters. Do not use an asterisk (*) because a factor name with an asterisk might be confused with an interaction effect. Factor names can be any length, but they must be distinct from one another in the first 32 characters.

On the **Variables** tab, you can also specify the name of the dependent variable; in this example, `Blood pressure` is used.

The completed **Variables** tab is shown in Figure 68.61.

Figure 68.61 Variables Tab with Factors and Number of Levels



Model

Click the **Model** tab, then choose from three model options:

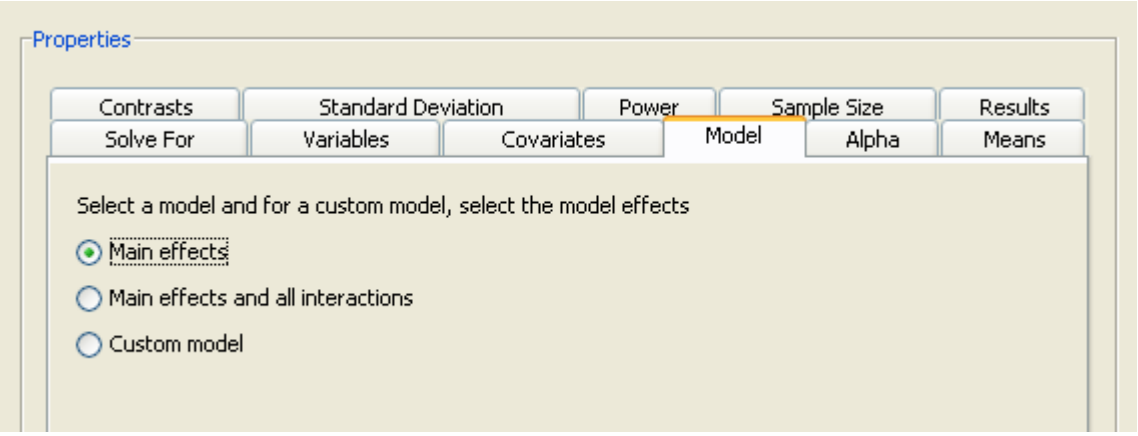
Main effects
: Only the main effects are included in the model.

Main effects and all interactions
: The main effects and all possible interactions are included in the model.

Custom model
: Selected effects are included in the model. The effects are selected in a model builder that is displayed when this model is selected. For more information about specifying a custom model, see the section "Specifying a Custom Model" on page 5124.

For this example, choose the default **Main effects** model, as shown in Figure 68.62.

Figure 68.62 Model Tab with Main Effects Selected



Alpha

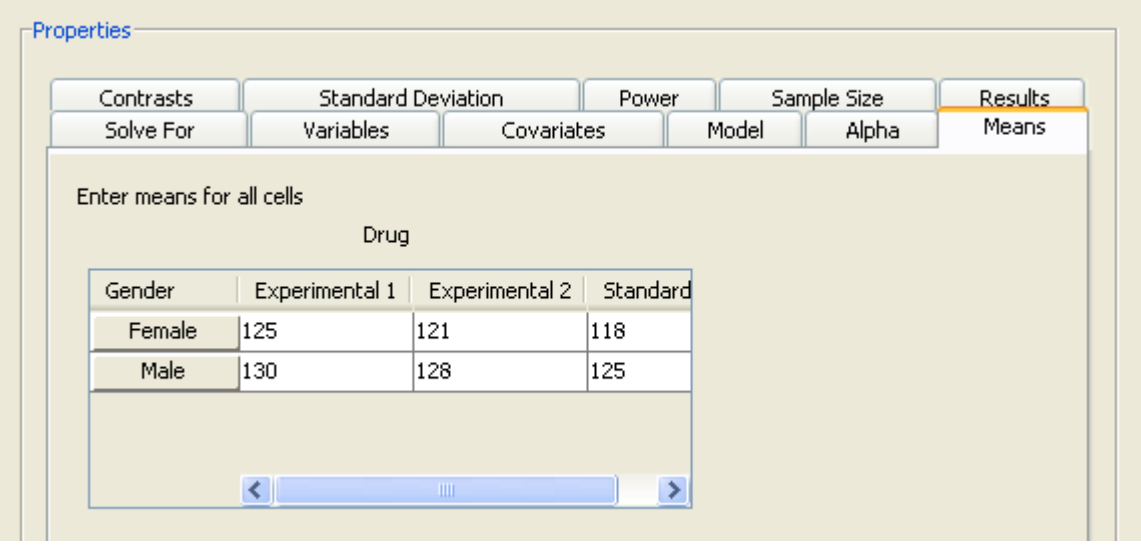
Click the **Alpha** tab to specify one or more significance levels. For the example, specify a single significance level of 0.05.

Alpha is the significance level (that is, the probability of falsely rejecting the null hypothesis). If you frequently use the same values for alpha, set them as the defaults in the Preferences window (**Tools►Preferences**).

Means

Click the **Means** tab to enter projected cell means for each cell of the design. The completed means for the example are shown in Figure 68.63.

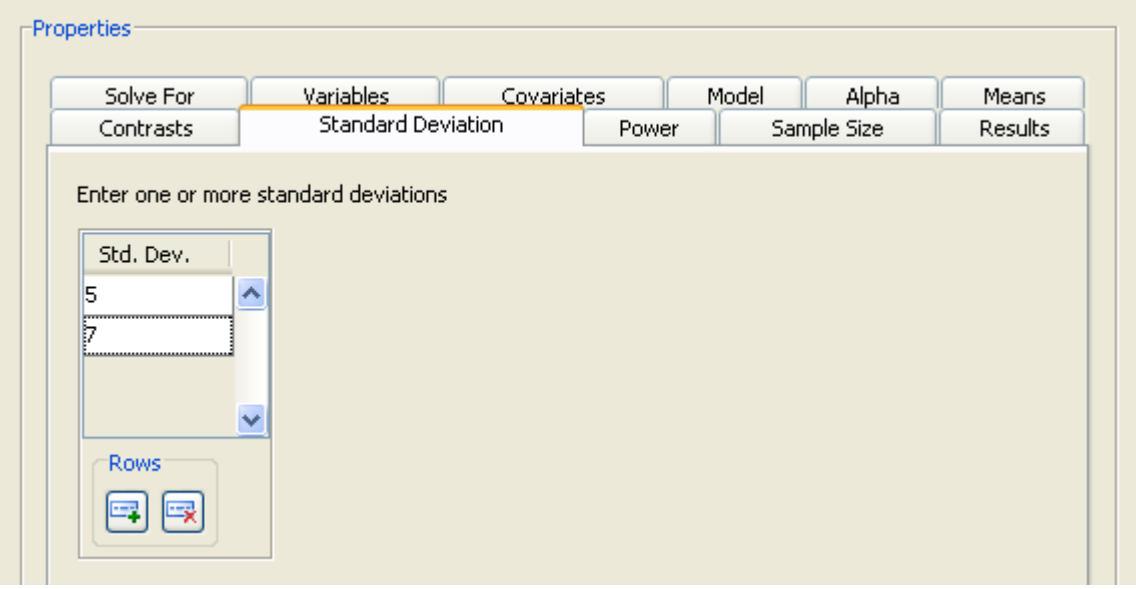
Figure 68.63 Means Tab with Cell Means



Standard Deviation

Click the **Standard Deviation** tab to specify one or more conjectured error standard deviations. The standard deviation is the same as the root mean squared error. For this example, enter two standard deviations, 5 and 7, as shown in Figure 68.64.

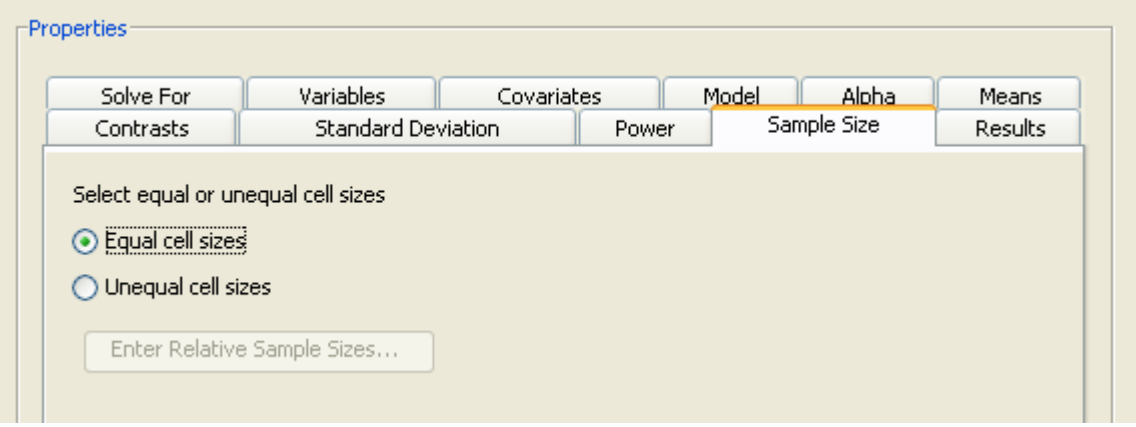
Figure 68.64 Standard Deviations Tab



Relative Sample Size

Click the **Sample Size** tab to select whether cell sample sizes are equal or unequal.

Figure 68.65 Sample Size Tab with Equal Cell Sample Sizes



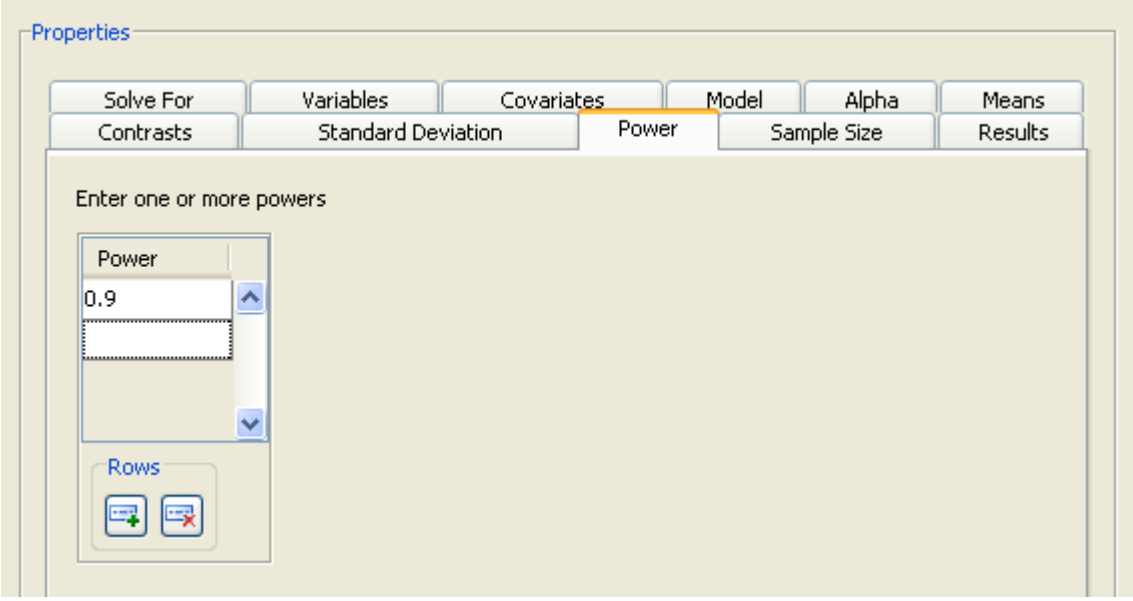
For the example, select the **Equal cell sizes** option, as shown in Figure 68.65.

When solving for sample size, it is necessary to specify whether the cell sample sizes are equal or unequal. If cell sizes are unequal, relative sample size weights must also be specified. For more information about providing sample size weights, see the section "Using Unequal Cell Sizes" on page 5121.

Power

Click the **Power** tab to specify one or more powers. For this example, enter a single power of 0.9, as shown in Figure 68.66.

Figure 68.66 Power Tab



Summary of Input Parameters

Table 68.6 contains the values of the input parameters for the example.

Table 68.6 Summary of Input Parameters

Parameter	Value
Model	Main effects
Alpha	0.05
Means	See Table 68.7
Standard deviation	5, 7
Relative sample sizes	Equal cell sizes
Power	0.9

Table 68.7 Cell Means

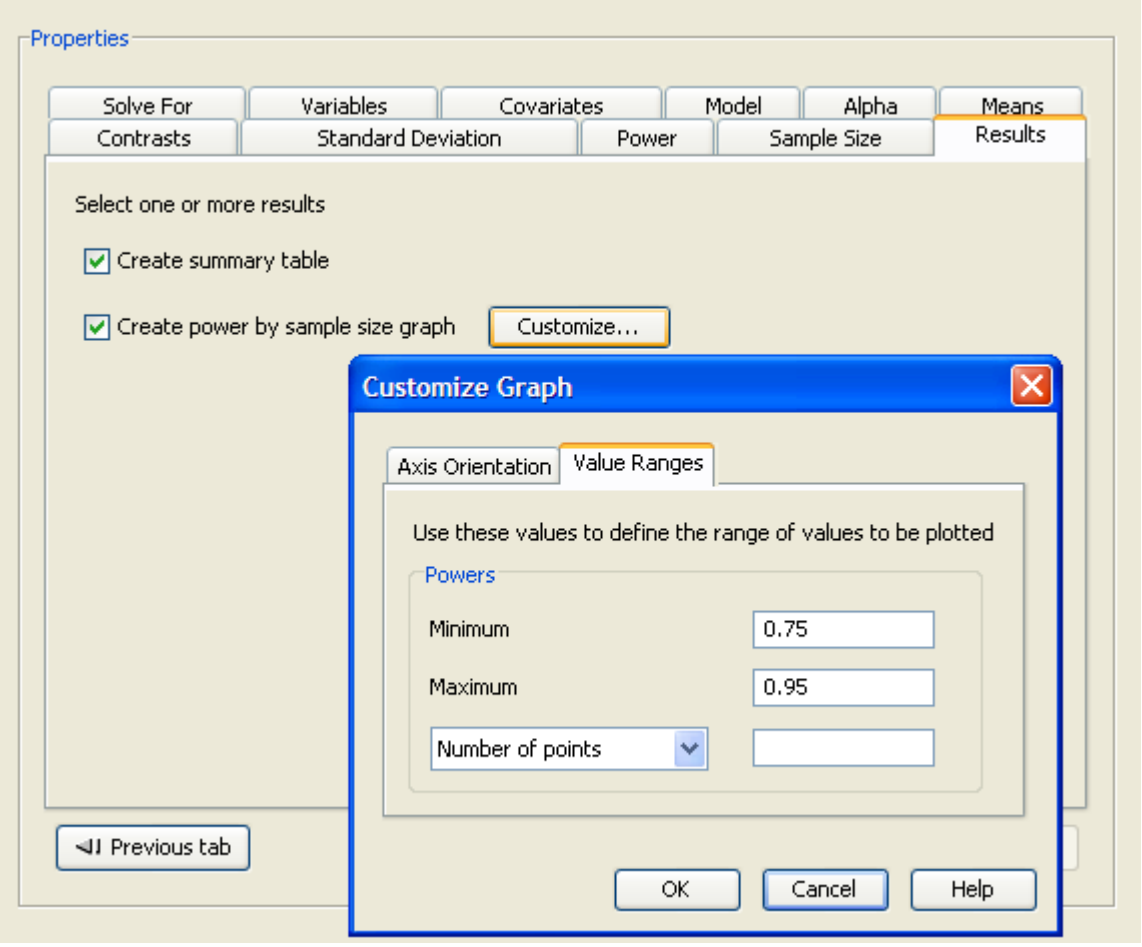
Gender	Drug Experimental 1	Experimental 2	Standard
Female	125	121	118
Male	130	128	125

Results Options

Click the **Results** tab to select desired results. For the example, select both the **Create summary table** and **Create power by sample size graph** check boxes.

The graph consists of four points, one for each of the four scenarios that were created by combining the two factor main effects with the two standard deviations. This graph is not very informative, so specify a range of powers for the horizontal power axis. To change the power axis of the graph, click the **Customize** button beside the **Create power by sample size graph** check box to open the Customize Graph window.

Figure 68.67 Value Ranges on Customize Graph Window



Click the **Value Ranges** tab and enter a minimum power of 0.75 and a maximum power of 0.95, as shown in Figure 68.67. Click **OK** to close the window.

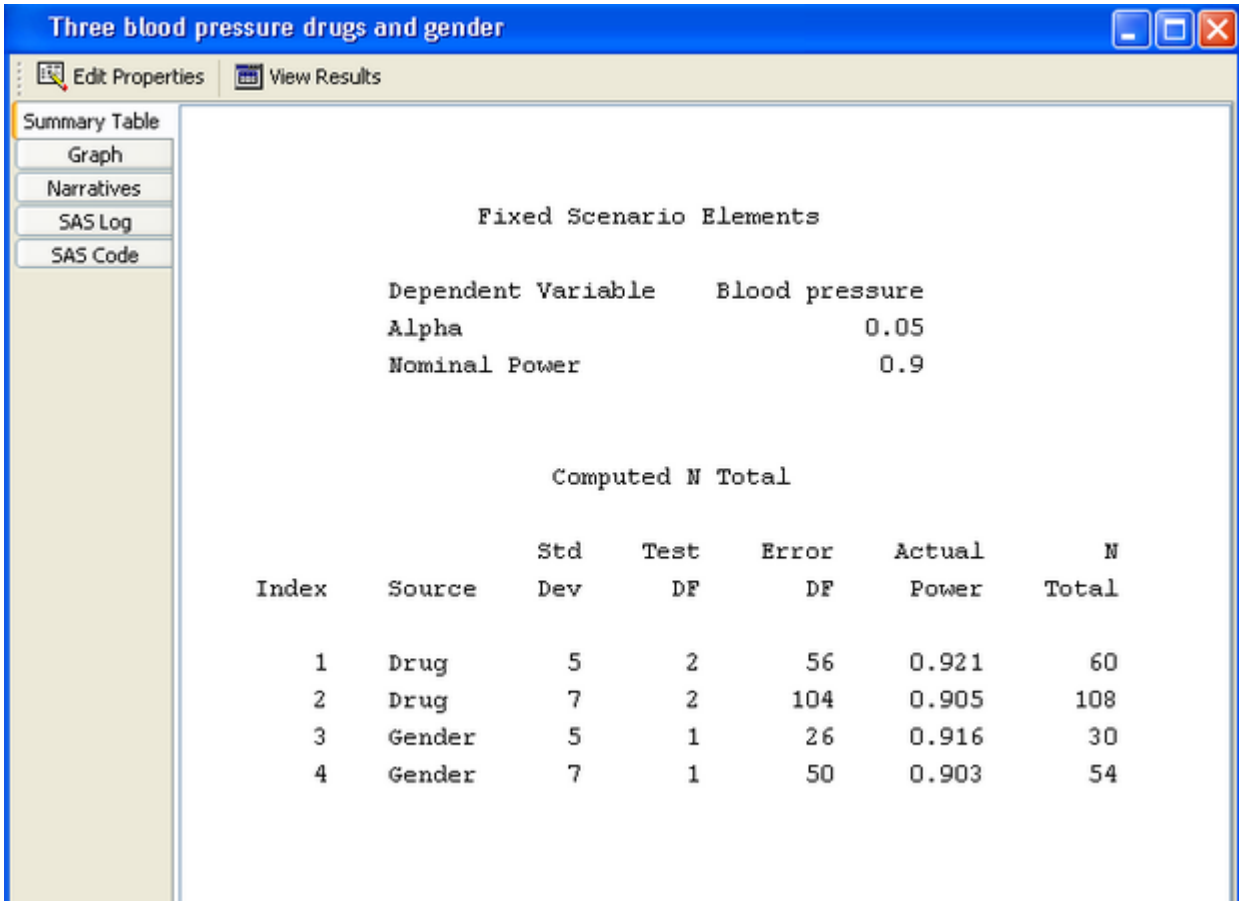
Now, click **Calculate** to perform the analysis.

Viewing Results

The results are displayed in separate tabs on the View Results page.

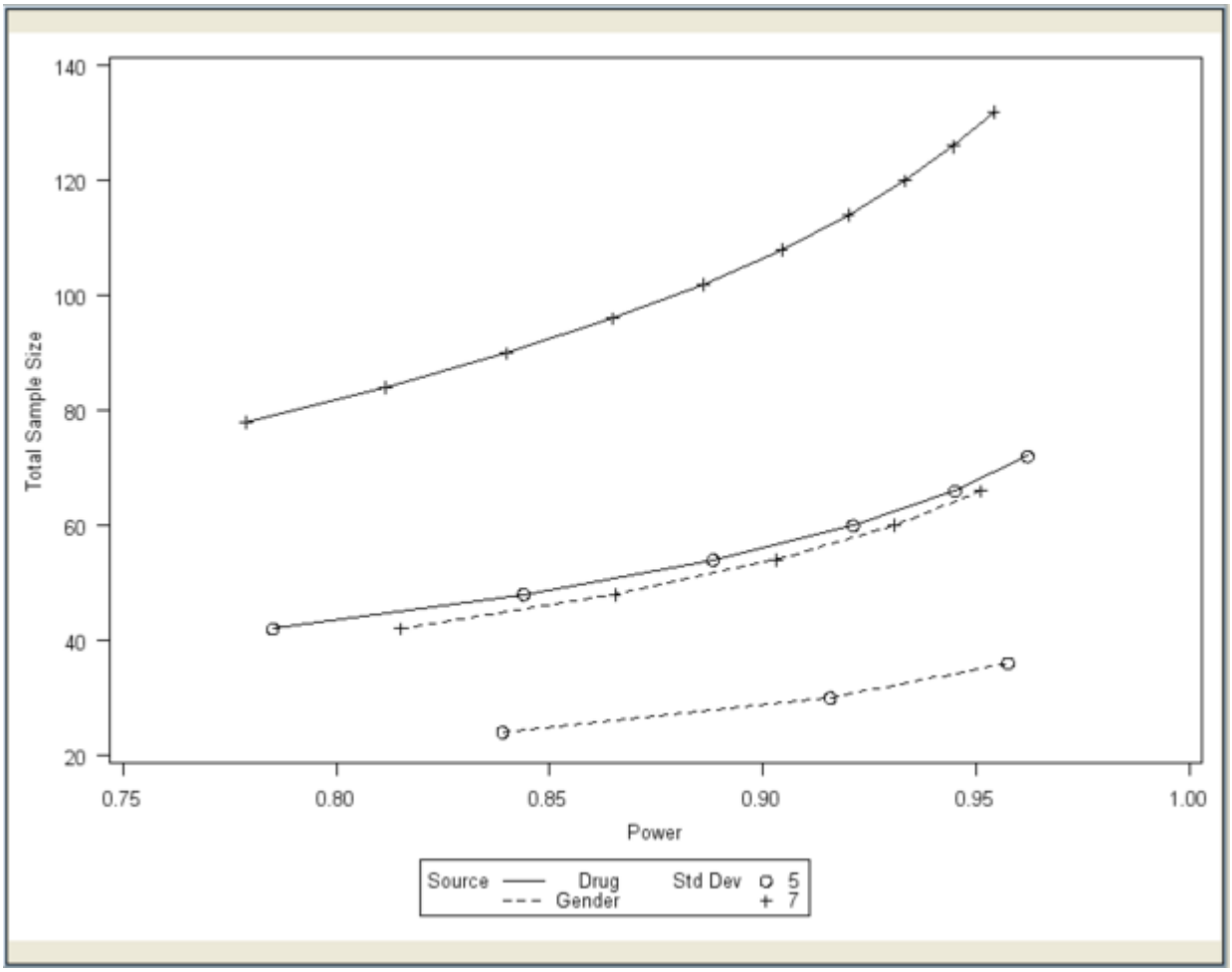
Click the **Summary Table** tab to view the summary table. In the `Computed N Total` table, sample sizes are listed for each combination of factor and standard deviation (Figure 68.68). You need a total sample size between 60 and 108 to yield a power of 0.9 for the `Drug` effect if the standard deviation is between 5 and 7. You need a sample size of half that for the `Gender` effect.

Figure 68.68 Summary Table



Click the **Graph** tab to view the power by sample size graph, as shown in Figure 68.69. One approximately linear curve is displayed for each standard deviation and factor combination.

Figure 68.69 Power by Sample Size Graph



Click the **Narratives** tab to create narratives of one or more scenarios. Select the first scenario, the Drug effect with the standard deviation of 5, in the narrative selector table. Note that the cell means are not included in the following narrative description:

```
For the usual F test of the Drug effect in the general linear
univariate model with fixed class effects [Blood pressure = Drug
Gender] using a significance level of 0.05, assuming the specified
cell means and an error standard deviation of 5, a total sample
size of 60 assuming a balanced design is required to obtain a
power of at least 0.9.  The actual power is 0.921.
```

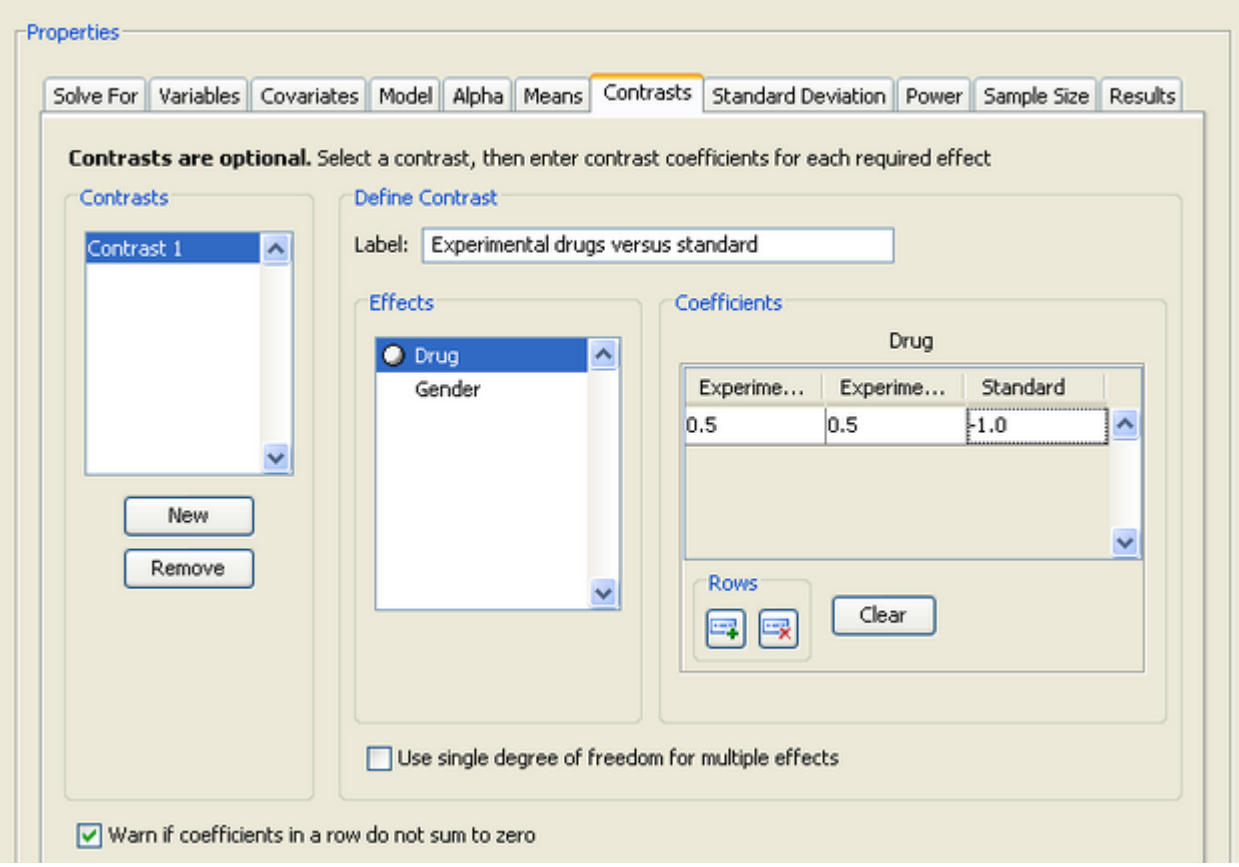
For more information about using the narrative facility, see the section "Creating Narratives" on page 5082.

Additional Topics

Adding Contrasts

Click the **Contrasts** tab to define one or more contrasts. Contrasts are optional. PSS allows contrasts to be added when using either a main effects model or a main effects and interactions model. At least two factors must have been specified in order to be able to enter contrasts. The contrast tab appears in Figure 68.70.

Figure 68.70 Contrast Tab with Coefficients



To create a contrast, click the **New** button. Then, select the newly created contrast (`Contrast 1`) from the list.

Specify a label for the contrast in the **Label** field. The label should be different from all of the factor names and all interactions in the model, as well as other contrast labels.

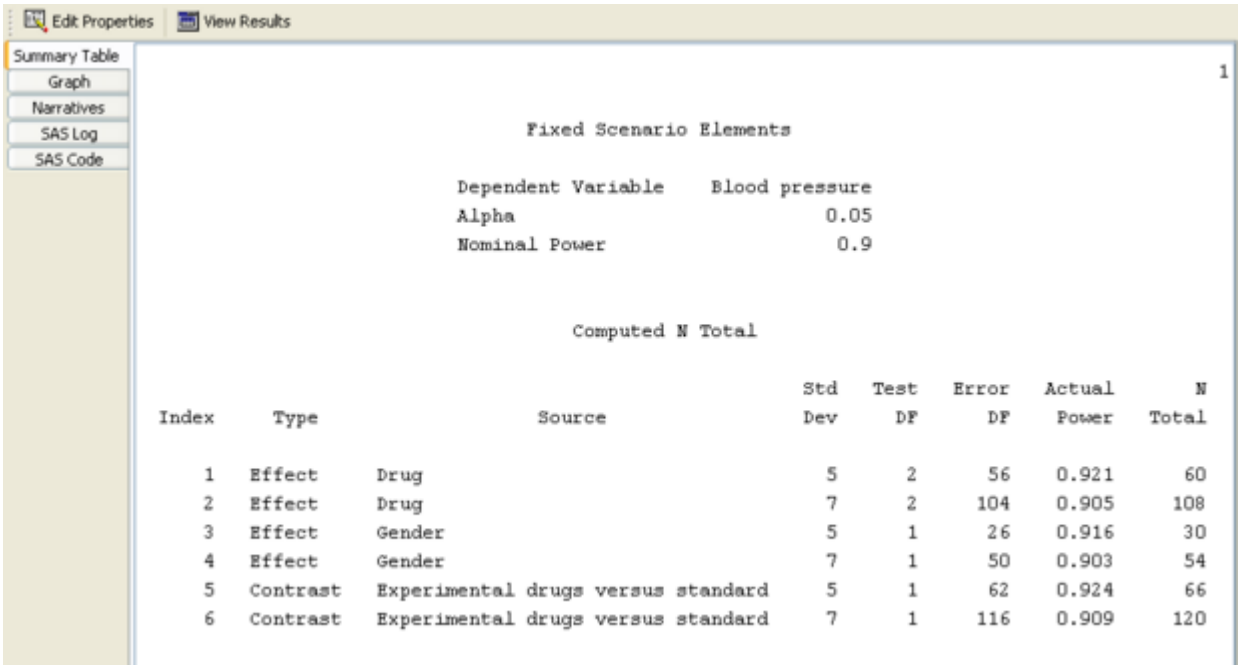
Then, for each term you want to include in the contrast, select the term in the **Effects** list and enter at least two coefficients per term. It is not necessary to enter zeros; blanks are considered to be zeros.

To clear all of the contrast coefficients for a term, click the **Clear** button. To remove a previously defined contrast, select it from the **Contrasts** list and click the **Remove** button.

In this example, you are interested in comparing the two experimental drugs to the standard drug. As shown in Figure 68.70, the contrast coefficients are 0.5, 0.5, and −1 for the three levels of the Drug effect.

Figure 68.71 shows the two scenarios for the contrast at the bottom of the `Computed N Total` table. The two scenarios also appear in the graph but the graph is not shown here.

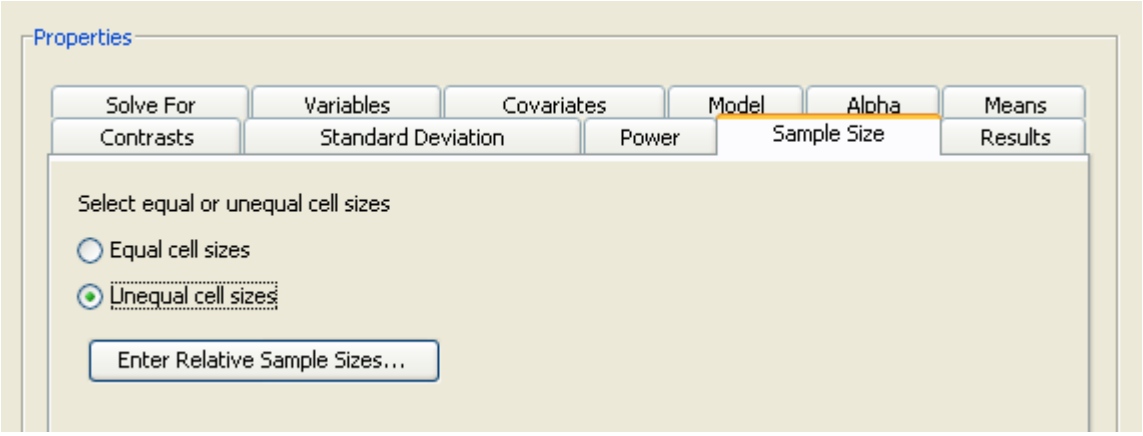
Figure 68.71 Computed N Total Table for the Contrast



Using Unequal Cell Sizes

Click the **Sample Size** tab to select the equal or unequal cell sizes option.

Figure 68.72 Sample Size Tab



For the example, select the **Unequal cell sizes** option, as seen in Figure 68.72, and then click the **Enter Relative Sample Sizes** button.

Figure 68.73 shows the window in which you can enter relative sample sizes. As an example, enter the sample size weights from Table 68.8.

Table 68.8 Sample Size Weights

	Drug		
Gender	**Experimental 1**	**Experimental 2**	**Standard**
Males	1	1	2
Females	1	1	2

If you have unequal cell sizes, you must enter relative sample size weights for the cells. Weights do not have to sum to 1 across the cells. Some weights can be zero, but enough weights must be greater than zero so that the effects and contrasts are estimable.

In this case, you want the sample size of the standard group to be twice that of each of the two experimental groups. Click **OK** to save the values and return to the Edit Properties page.

Figure 68.73 Relative Sample Sizes Window

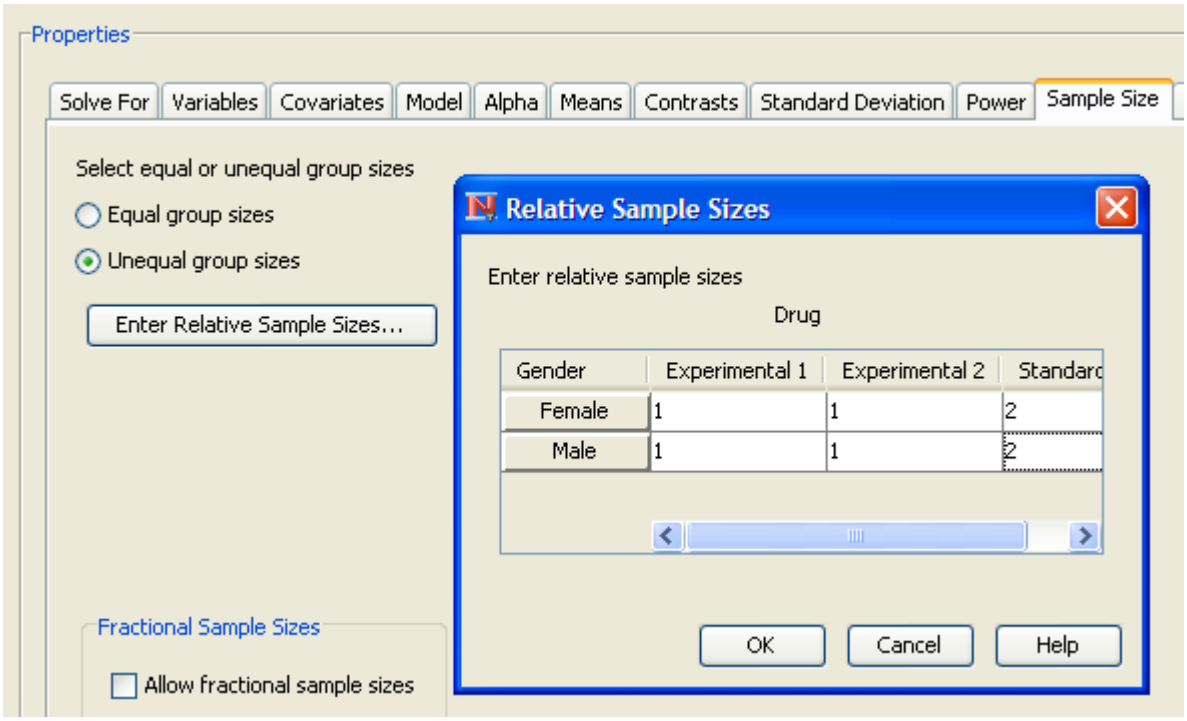


Figure 68.74 shows the summary table for the Drug by Gender example.

Figure 68.74 Summary Table for Unbalanced Design Example

1

Fixed Scenario Elements

Dependent Variable	Blood pressure
Weight Variable	_Weight_
Alpha	0.05
Nominal Power	0.9

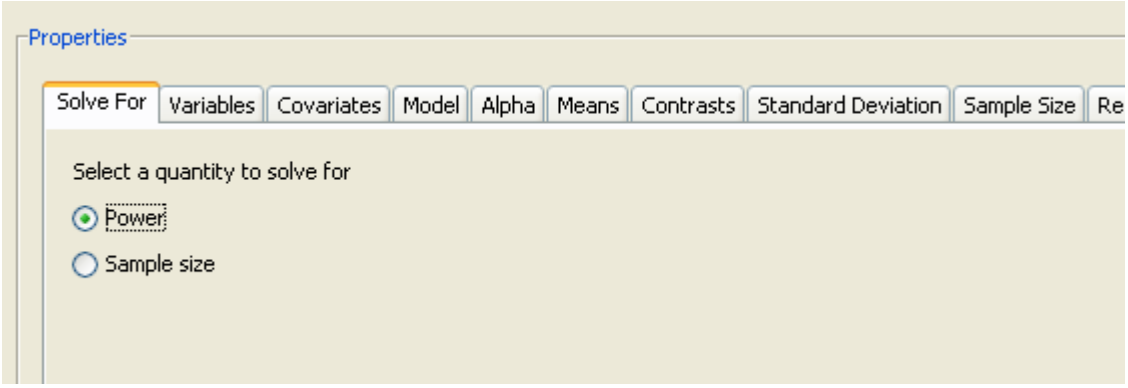
Computed N Total

Index	Type	Source	Std Dev	Test DF	Error DF	Actual Power	N Total
1	Effect	Drug	5	2	52	0.910	56
2	Effect	Drug	7	2	100	0.902	104
3	Effect	Gender	5	1	28	0.944	32
4	Effect	Gender	7	1	52	0.926	56
5	Contrast	Experimental drugs versus standard	5	1	52	0.911	56
6	Contrast	Experimental drugs versus standard	7	1	100	0.901	104

Solving for Power

In addition to solving for sample size, you can also solve for power. Figure 68.75 shows the two options. Click the **Solve For** tab to select the **Power** option.

Figure 68.75 Solve For Tab with Power Option Selected



When solving for power, you must provide sample size information. For the general linear univariate model analysis, you provide this information by using one of two alternate forms. To choose the desired alternate form, select the desired form from the **Select a form** list box on the **Sample Size** tab. The alternate forms are:

Sample size per cell
: Enter the sample size for a cell. Cell sizes are assumed to be equal. Sample size is reported in the summary table as total sample size.

Total N, Cell weights
: Enter the total sample size and specify whether cell sizes are to be equal or unequal. Select the **Equal cell sizes** or **Unequal cell sizes** option. For unequal cell sizes, you also enter cell weights. Click the **Enter Relative Sample Sizes** button to display a window that is used to enter the data. For more information about using unequal cell sizes, see the section "Using Unequal Cell Sizes" on page 5121.

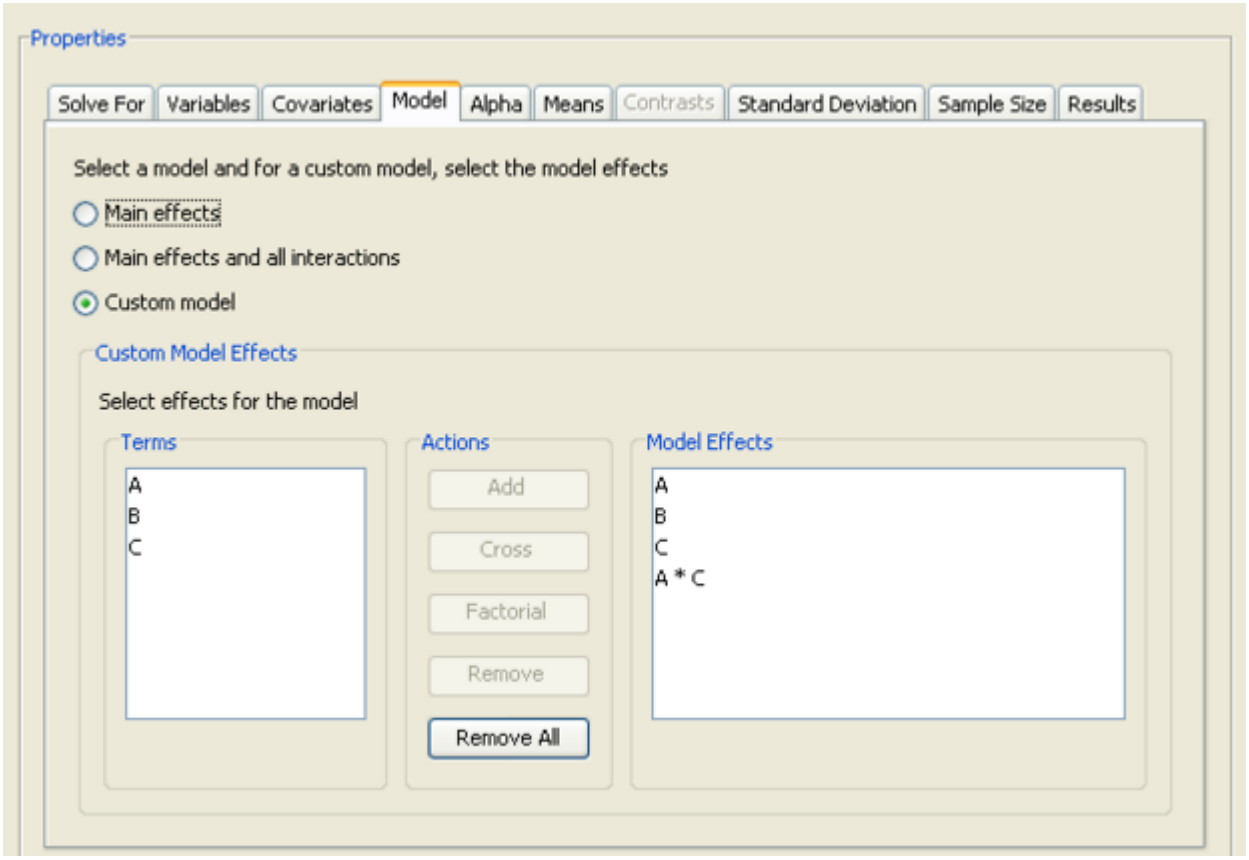
Specifying a Custom Model

Click the **Model** tab to select from three types of models: a **Main effects** model, a **Main effects and all interactions** model, and a **Custom model**.

To specify a custom model, select the **Custom model** option; then a model building facility is displayed.

The facility displays a list of the factors on the left. Construct the desired model using the **Add**, **Cross**, and **Factorial** buttons. The example shown in Figure 68.76 has the three main effects and one of the four possible interactions.

Figure 68.76 Model Tab with Custom Model Builder Displayed



Add the three main effects (`A`, `B`, `C`) by selecting them in the **Terms** list and clicking the **Add** button. Add the `A*B` interaction by selecting the `A` and `B` factors in the **Terms** list and clicking the **Cross** button.

To create the complete factorial design of several factors, select the factors in the **Terms** list, then click the **Factorial** button. All possible main effects and interactions are added to the **Model Effects** list.

To remove effects, select them in the **Model Effects** list and click the **Remove** button. Clicking the **Remove All** button removes all effects in the model.

Including Covariates

Click the **Covariates** tab to enter covariate information.

Figure 68.77 Covariates Tab with Proportional Reduction in Variance Form

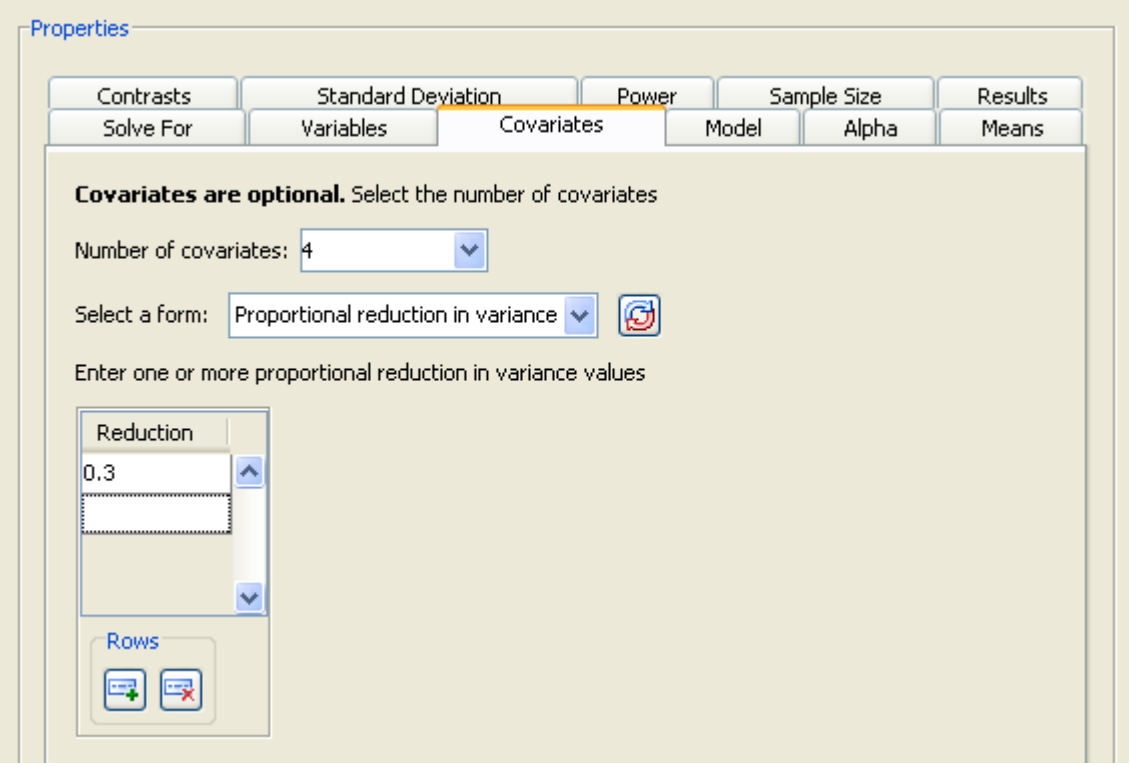


Figure 68.77 illustrates four covariates and a proportional reduction in variation of 0.3. The results for the analysis are not shown.

Covariates are optional. If you have covariates, include the total number of degrees of freedom for all covariates. To do this, add the number of continuous covariates and the sum of the degrees of freedom of the classification covariates, and enter this total in the **Number of Covariates** list box. For example, with two continuous covariates and a single classification covariate factor with three levels, the total would be $2 + (3 - 1) = 4$.

Also, you must enter the correlation between the dependent variable and the set of covariates. Two alternate forms are available: **Multiple correlation** and **Proportional reduction in variance**. Select the desired form and enter one or more values.

The multiple correlation is between the set of covariates and the dependent variable. Proportional reduction in variation is how much the variance of the dependent variable is reduced by the inclusion of the covariates, expressed as a proportion between 0 and 1.

Example: Two-Sample Survival Rank Tests

Overview

Survival analysis often involves the comparison of survival curves. PSS provides sample size and power calculations for two-sample survival rank analyses. Several rank tests are available: Gehan, log-rank, and Tarone-Ware. There are also several ways to specify the survival functions. For more information about power and sample size analysis for survival rank tests, see Chapter 67, "The POWER Procedure."

The Example

Suppose you want to compare survival rates for an existing cancer treatment and a new treatment. You intend to use a log-rank test to compare the overall survival curves for the two treatments. You want to determine a sample size to achieve a power of 0.8 for a two-sided test using a balanced design, with a significance level of 0.05.

The survival curve of patients for the existing treatment is known to be approximately exponential with a median survival time of five years. You think that the proposed treatment will yield a survival curve described by the times and probabilities listed in Table 68.9. Patients are to be accrued uniformly over two years and followed for three years.

Table 68.9 Survival Probabilities for Proposed Treatment

Time	Probability
1	0.95
2	0.90
3	0.75
4	0.70
5	0.60

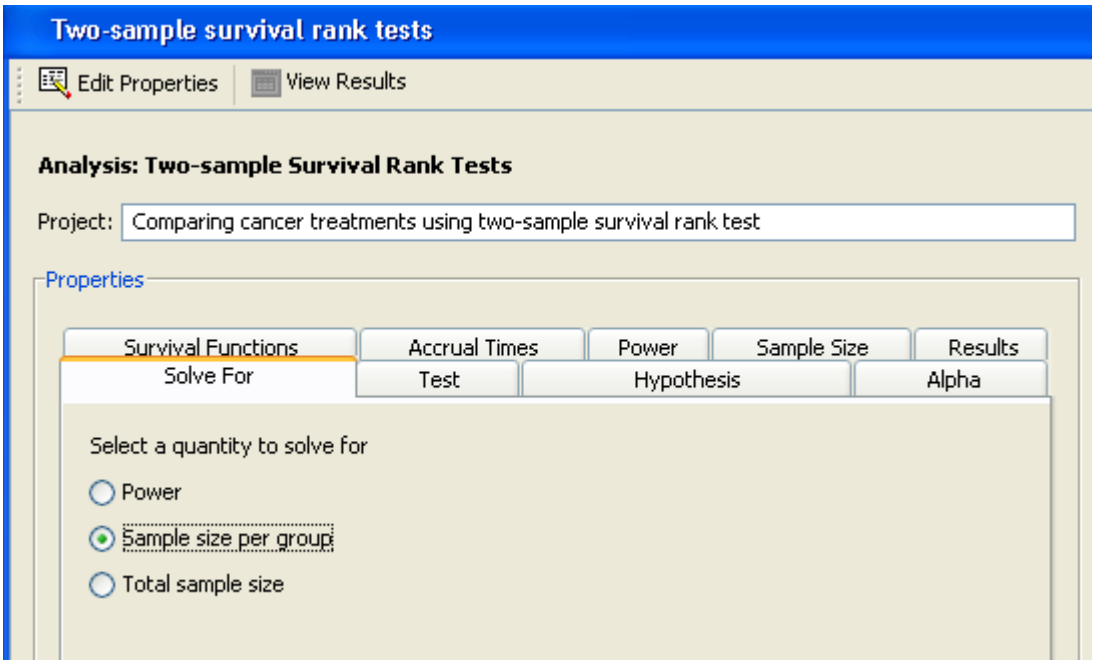
To create a new survival analysis project, select **File►New**, Then, under the **Survival Analysis** section, select **Two-sample survival rank tests** and click **OK**. The **Two-sample survival rank tests** project appears with the Edit Properties page displayed.

Editing Properties

Project Description

For the example, change the project description to `Comparing cancer treatments using two-sample survival rank test`.

Figure 68.78 Project Description and Solve For Tab



Solve For

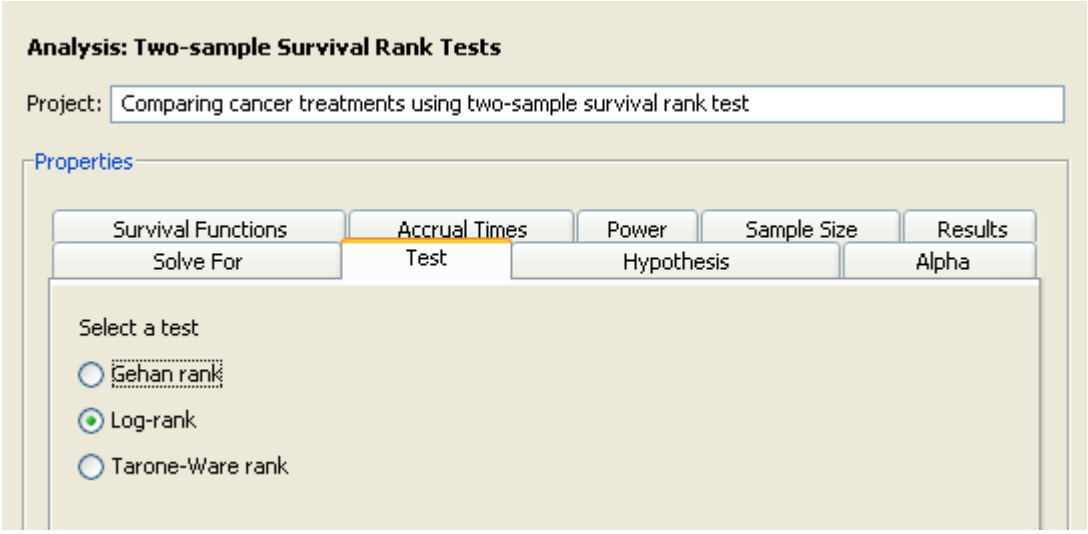
Click the **Solve For** tab to select the quantity to solve for. For this example, select the **Sample size per group** option, as shown in Figure 68.78. For information about calculating total sample size, see the section "Solving for Sample Size" on page 5108.

In this analysis you can solve for power, sample size per group, or total sample size.

Test

Click the **Test** tab to select a rank test. For this example, select the **Log-rank** option, as shown in Figure 68.79.

Figure 68.79 Test Tab

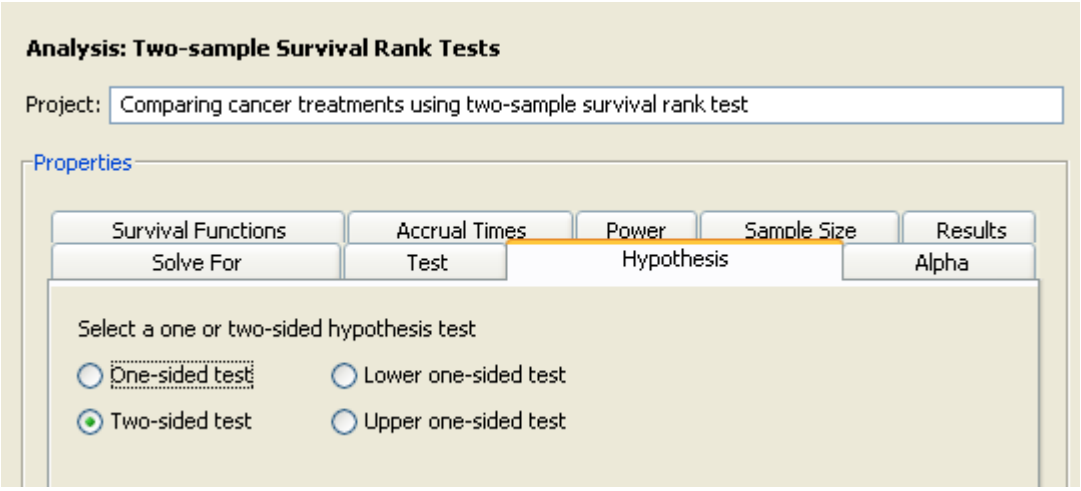


Several rank tests are available: Gehan, log-rank, and Tarone-Ware. The Gehan test is most sensitive to survival differences near the beginning of the study period, the log-rank test is uniformly sensitive throughout the study period, and the Tarone-Ware test is somewhere in between.

Hypothesis

Click the **Hypothesis** tab to select a one- or two-sided test. For the example, select the **Two-sided test** option, as shown in Figure 68.80.

Figure 68.80 Hypothesis Tab

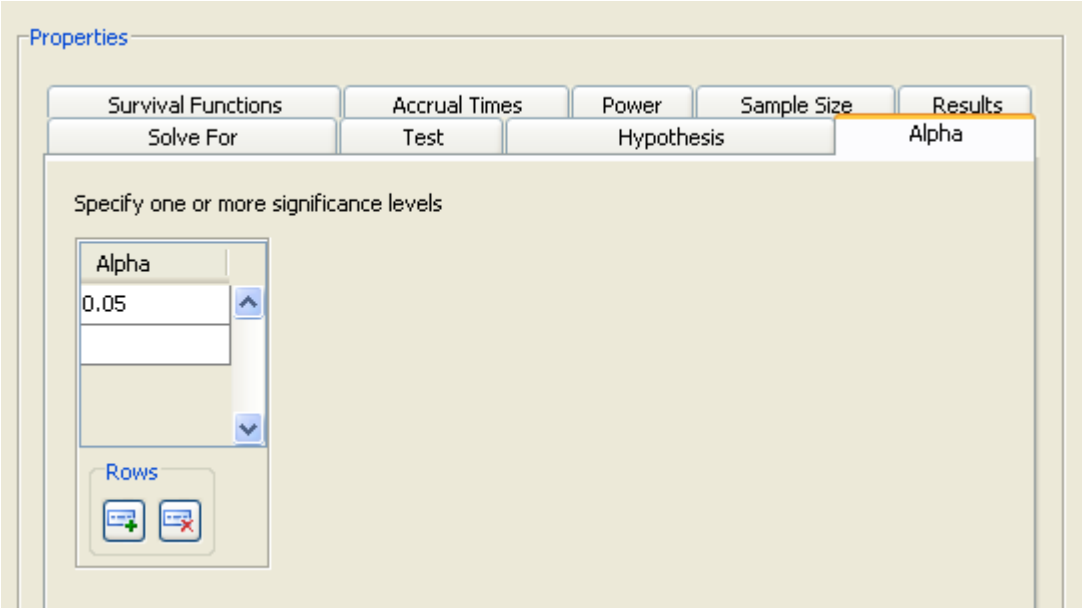


You can choose either a one- or two-sided test. For the one-sided test, the alternative hypothesis is assumed to be in the same direction as the effect. If you do not know the direction of the effect (that is, whether it is positive or negative), the two-sided test is appropriate. If you know the effect's direction, the one-sided test is appropriate. If you specify a one-sided test and the effect is in the unexpected direction, the results of the analysis are invalid.

Alpha

Click the **Alpha** tab to enter one or more values for the significance level. For the example, enter the desired significance level of 0.05 in the first cell of the Alpha table, as shown in Figure 68.81, if it is not already the default value.

Figure 68.81 Alpha Tab

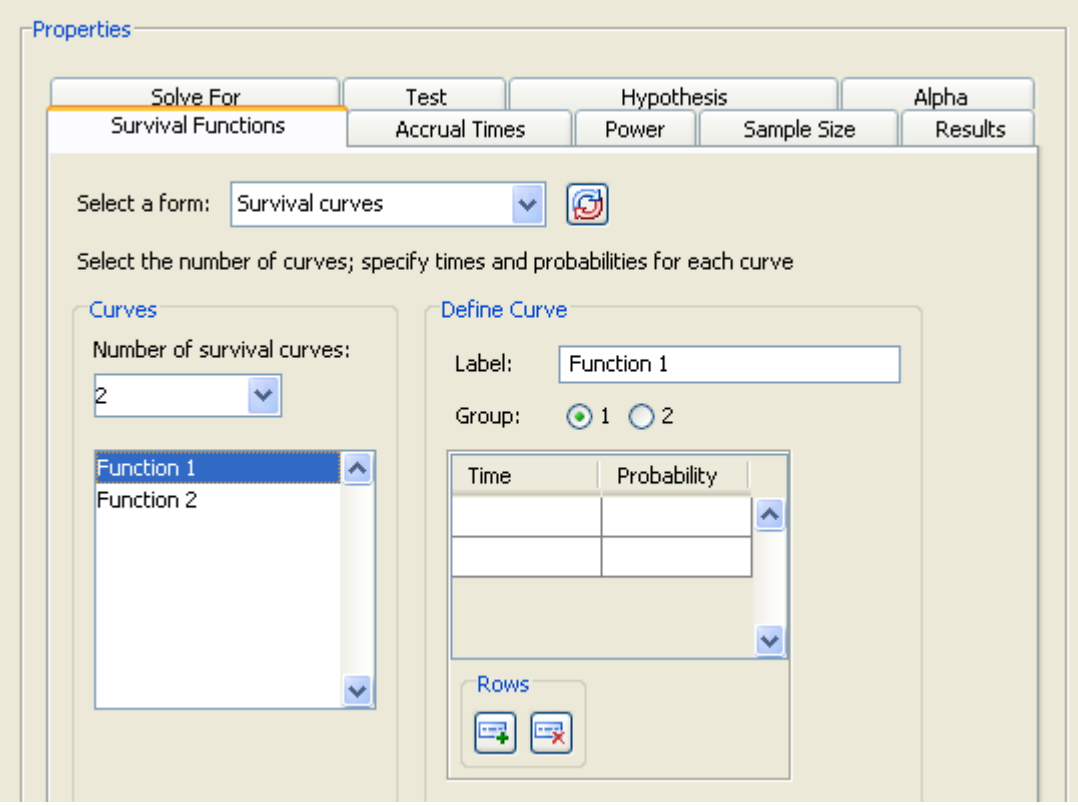


The significance level is the probability of falsely rejecting the null hypothesis. If you frequently use the same values for alpha, set them as the defaults in the Preferences window.

Survival Functions

Click the **Survival Functions** tab to select the input form for the survival functions.

Figure 68.82 Survival Functions Tab with Number of Curves



Examine the input alternatives available in the **Select a form** list. There are four alternate forms for entering survival functions. The first three apply only to exponential curves; the fourth applies to both piecewise linear and exponential curves.

Group median survival times
: Enter median survival times for the two groups.

Group hazards
: Enter hazards for the two groups.

Hazards, Hazard ratios
: Enter hazards for the reference group and hazard ratios.

Survival curves
: Enter survival probabilities and their associated times for each of several curves. Select or enter the number of curves from the drop-down list; at least two curves are required. Then, for each curve, select it in the left-hand list, select the Group 1 or Group 2 option, and then define the survival curve by entering pairs of times and probabilities. Enter a time and probability pair only if the probability is less than that of the previous pair.

For information about using the other forms, see the section "Using the Other Survival Curve Forms" on page 5141.

For each survival curve, select the curve in the left-hand list. Then, enter a descriptive label and select which group it is for. The labels should be unique. Finally, enter pairs of survival times and probabilities.

When you enter probabilities, enter a time and probability pair only when the probability for a survival curve changes. For example, if the probability for curve 1 at time 1 and 2 is 0.9 and at time 3 is 0.8, enter 0.9 for time 1 and 0.8 for time 3.

To specify an exponential survival curve, enter a single time and probability pair. In the example, the exponential curve for the existing treatment is defined by a probability of 0.5 at time 5.

The units of time for the survival curves must correspond to the units for the accrual, follow-up, and total times, which are described in the section "Accrual Times" on page 5134.

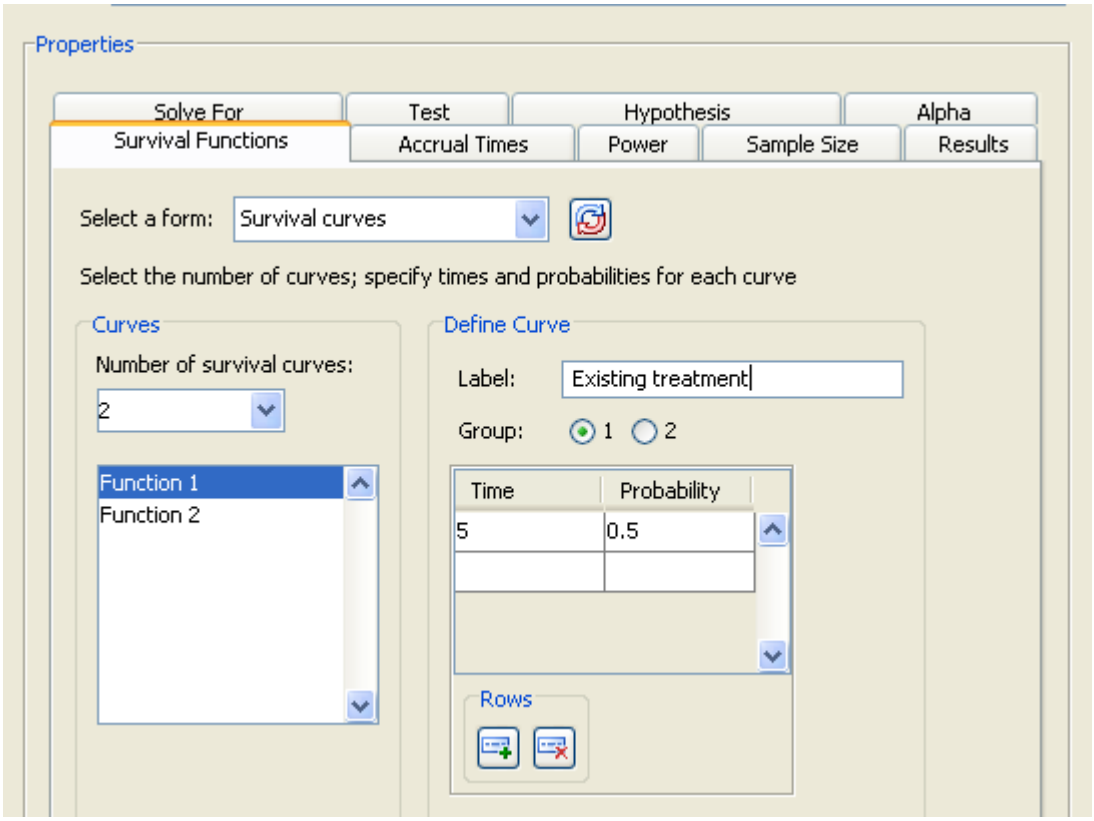
You can also compare several survival curves. For example, if you have two scenarios, A and B, for group 1's curve and two scenarios, C and D, for group 2's curve, then specify probabilities for the four curves and assign A and B to group 1 and C and D to group 2.

For the example, select the **Survival curves** form, as shown in Figure 68.82. Enter the value, 2, in the **Number of survival curves** list box.

For the example enter the following values:

- For the first survival curve, enter a label of `Existing treatment` and select the **Group 1** option. For the first curve, enter a time of 5 and a probability of 0.5. Figure 68.83 shows the resulting values.

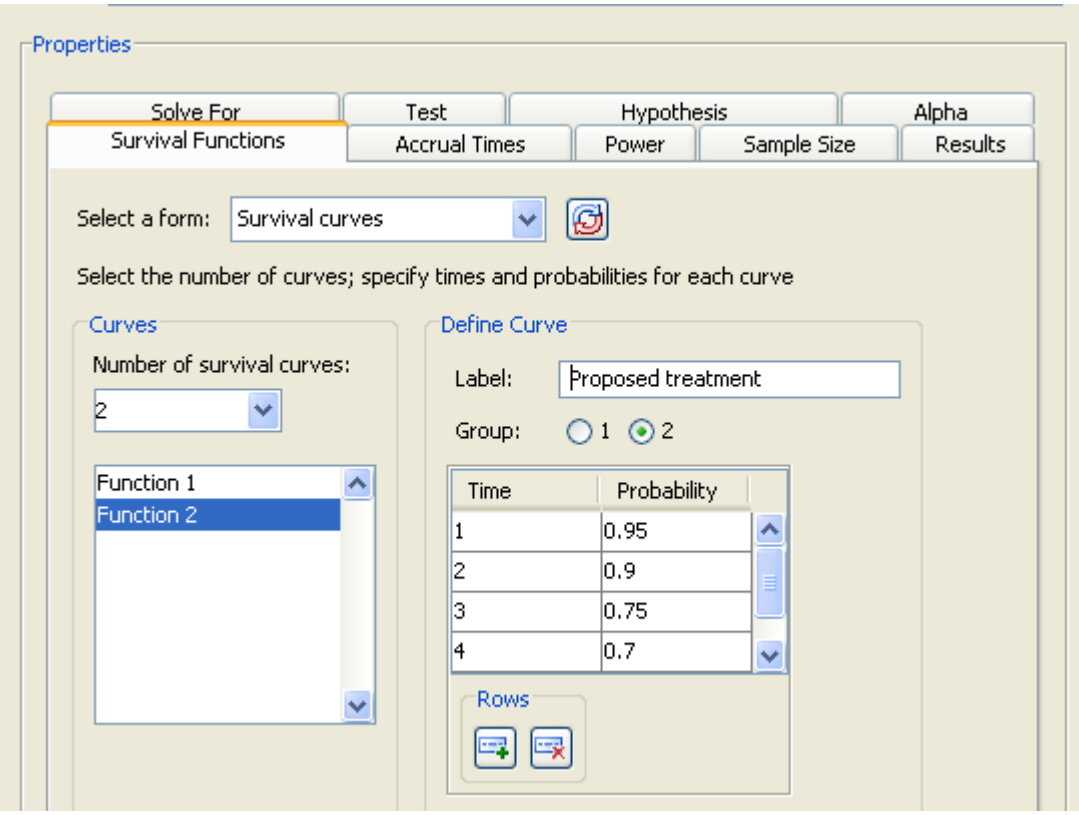
Figure 68.83 Survival Times and Probabilities for Curve 1



- For the second curve select `Function 2` in the selection list on the left side of the tab. Enter a label of `Proposed treatment` and select the **Group 2** option. Then, enter time values of 1 through 5 and the corresponding probabilities of 0.95, 0.9, 0.75, 0.7, and 0.6. To add rows to the table, click the button beneath the table.

  Figure 68.84 shows these values; the last row of the time and probability table is not displayed.

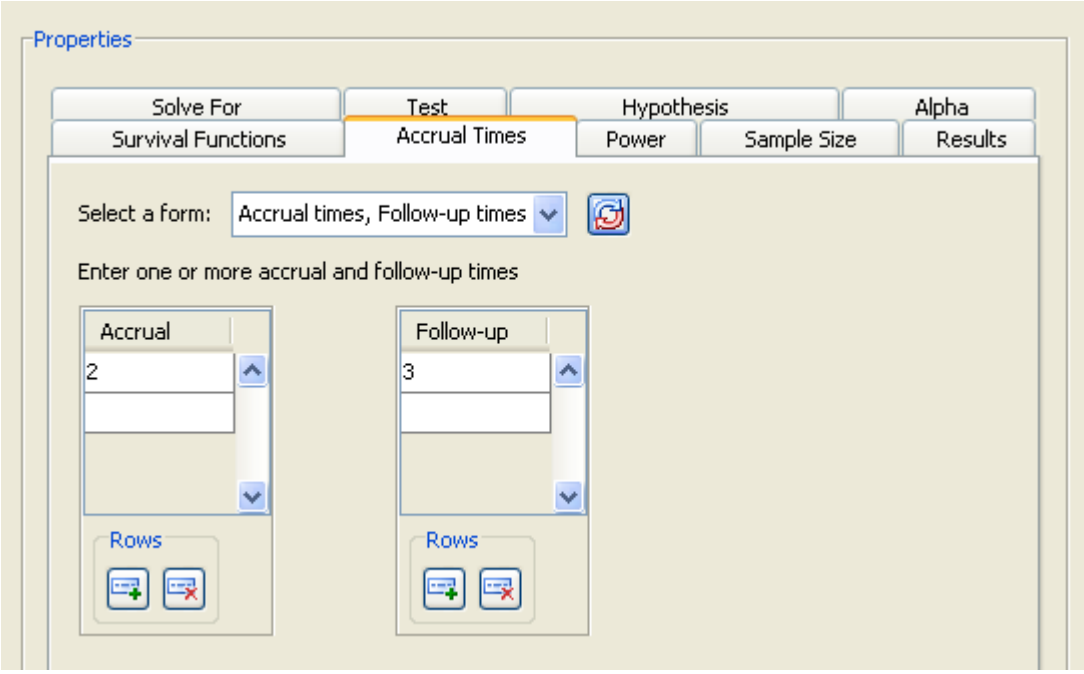
Figure 68.84 Survival Times and Probabilities for Curve 2



Accrual Times

Click the **Accrual times** tab to select an input form for accrual times and to enter the times.

Figure 68.85 Accrual Times Tab



Examine the alternatives available in the **Select a form** list.

Accrual time is the period during which subjects are brought into the study. Follow-up time is the period during which subjects are observed after all subjects have been included in the study. Total time is the sum of accrual and follow-up time. The units of time for the accrual, follow-up, and total times must correspond to the units you used specified for the survival curves.

When you enter survival curves, the sum of the accrual and follow-up times must be less than the largest time for each survival curve. This does not apply to survival curves represented by a single time, which represent exponential curves.

On the **Accrual Times** tab, there are three alternate forms for entering accrual and follow-up times:

Accrual times, Follow-up times
: Enter accrual and follow-up times.

Accrual times, Total times
: Enter accrual and total times.

Follow-up times, Total times
: Enter follow-up and total times.

For the example, select the **Accrual times, Follow-up times** form. Then enter a single value of `2` in the Accrual table and a value of `3` in the Follow-up table, as shown in Figure 68.85.

Power

Click the **Power** tab to enter one or more power values. For the example, enter a single value of `0.8`.

When you calculate sample size, it is necessary to specify one or more powers.

Summary of Input Parameters

Table 68.10 contains the values of the input parameters for the example.

Table 68.10 Summary of Input Parameters

Parameter	Value
Solve for	Sample size per group
Test	Log-rank
Hypothesis	Two-sided test
Alpha	0.05
Survival function form	Survival curves
Survival curves	See Table 68.11 and Table 68.12
Accrual and follow-up times form	Accrual time, Follow-up times
Accrual times	2
Follow-up times	3
Power	0.8

Table 68.11 and Table 68.12 contain times and probabilities for the two survival curves, respectively.

Table 68.11 Survival Times and Probabilities for Existing Treatment (Survival Curve 1)

Time	Probability
5	0.5

Table 68.12 Survival Times and Probabilities for Proposed Treatment (Survival Curve 2)

Time	Probability
1	0.95
2	0.90
3	0.75
4	0.70
5	0.60

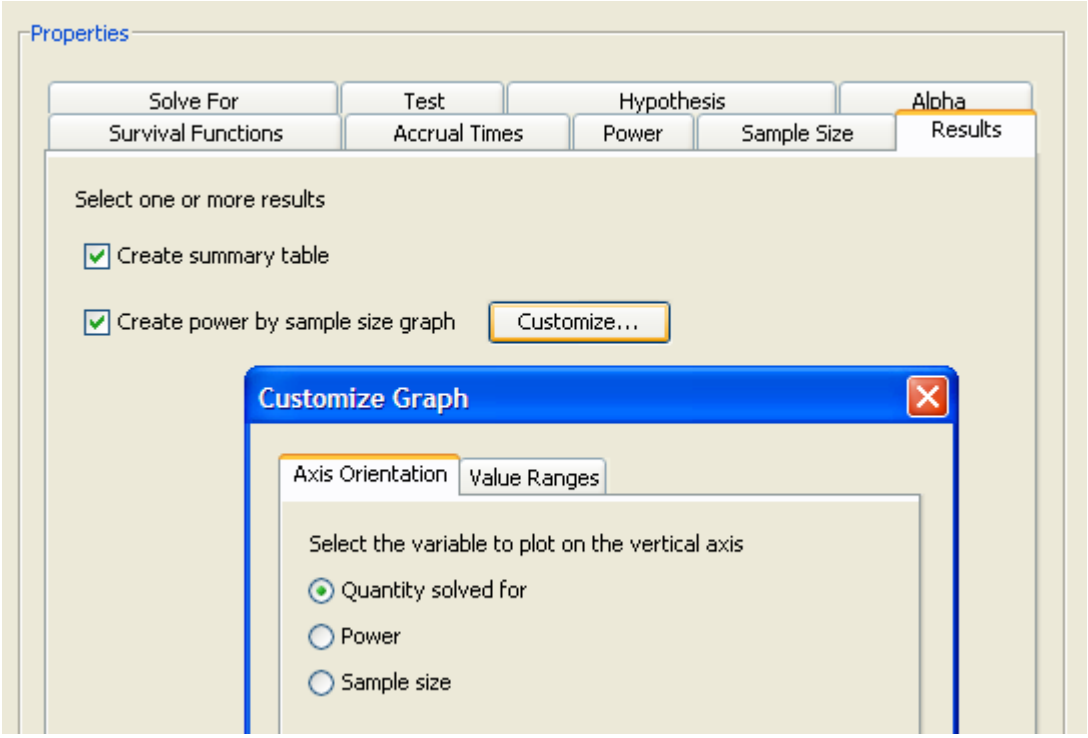
Result Options

Click the **Results** tab to specify the desired result options. For the example, request both results by selecting both the **Create summary table** and **Create power by sample size graph** check boxes.

Specifying only one power (as in this example) produces a graph with a single point. You might be interested in a plot of sample sizes for a range of powers—say, between 0.75 and 0.85. You can customize the graph by specifying the values for the power axis. Also, you might want to change the appearance of the graph to have sample size (per group) on the vertical axis and power on the horizontal axis.

Click the **Customize** button beside the **Create power by sample size graph** check box to customize the graph. The Customize Graph window is displayed, as shown in Figure 68.86.

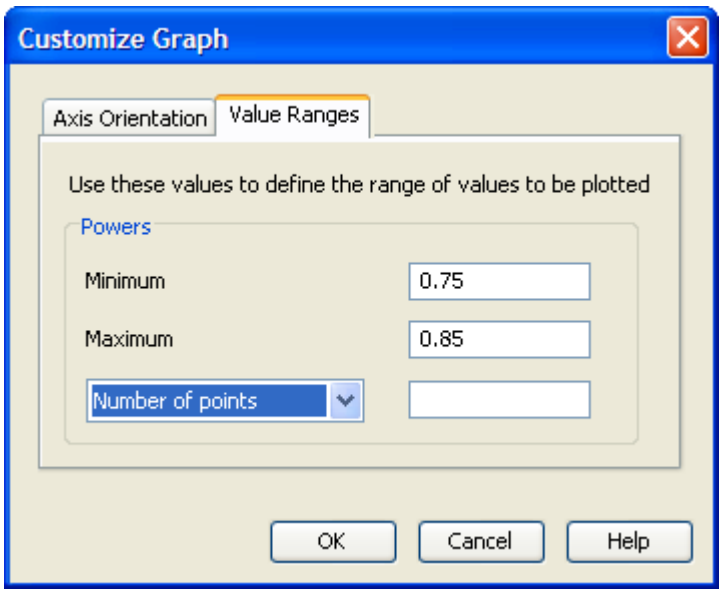
Figure 68.86 Customize Graph Window with Axis Orientation Tab



Click the **Axis Orientation** tab to select which variable to plot on the vertical axis. For the example, select the **Quantity solved for** option, as shown in Figure 68.86. This option plots sample size on the vertical axis and power on the horizontal axis. You could also have chosen the **Sample size** option.

Click the **Value Ranges** tab to enter minimum and maximum values for a plot axis. For the example, enter a minimum of `0.75` and a maximum of `0.85` in the Powers text boxes. This sets the range of values on the axis for powers. The completed Value Ranges tab of the window is displayed in Figure 68.87. You can set the axis values only for the quantity that is not being solved for.

Figure 68.87 Customize Graph Window with Value Ranges Tab



Click **OK** to save the values that you have entered and return to the Edit Properties page.

Then, click **Calculate** to perform the analysis. If there are no errors in the input parameter values, the View Results page appears. If there are errors in the input parameter values, you are prompted to correct them.

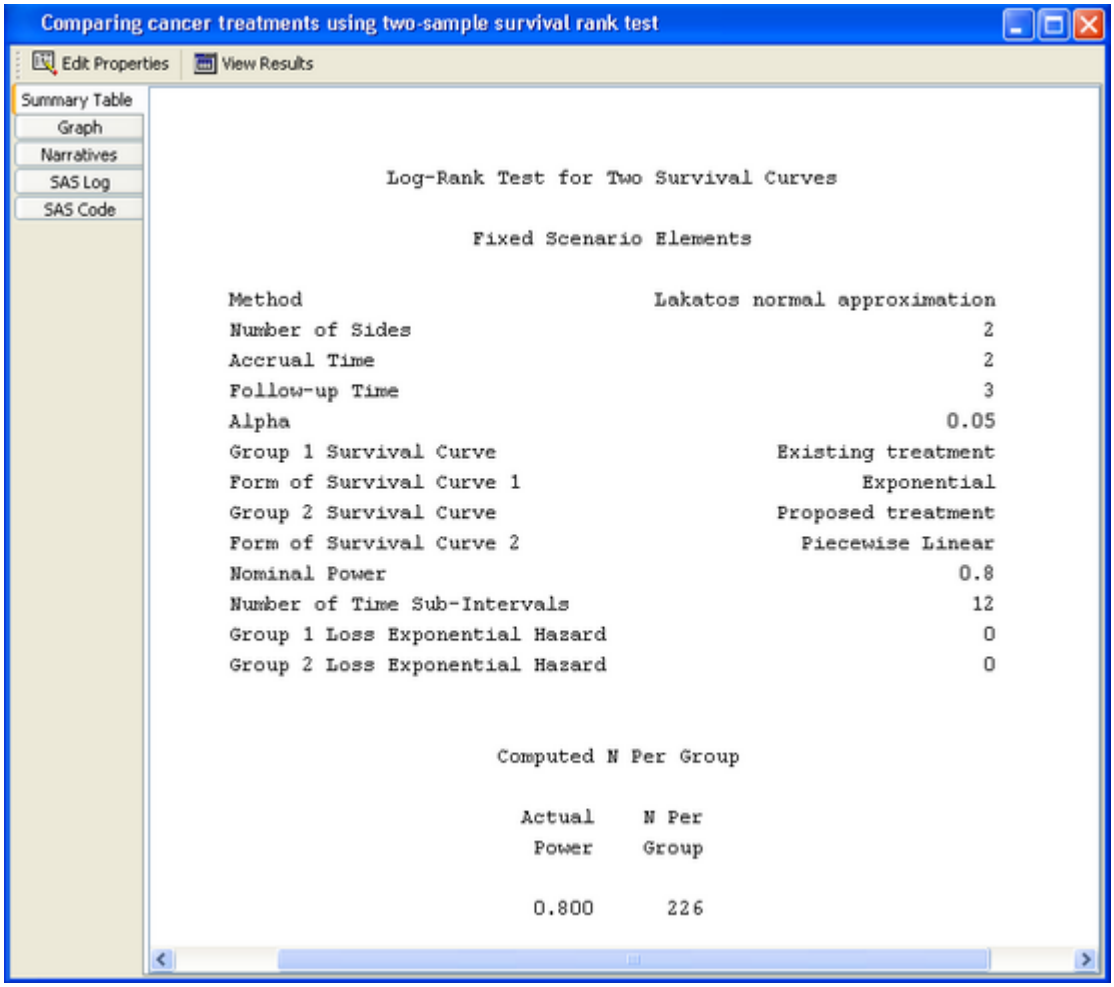
Viewing Results

The results appear in separate tabs on the View Results page of the project. Select the tab of each result that you want to view.

Summary Table

Click the **Summary Table** tab to view the summary table. It is composed of two subtables. As shown in Figure 68.88, the `Fixed Scenario Elements` and `Computed N Per Group` tables include the values of the input parameters and the computed quantity (in this case, sample size per group, `N per group`). The sample size per group for the single requested scenario is 226.

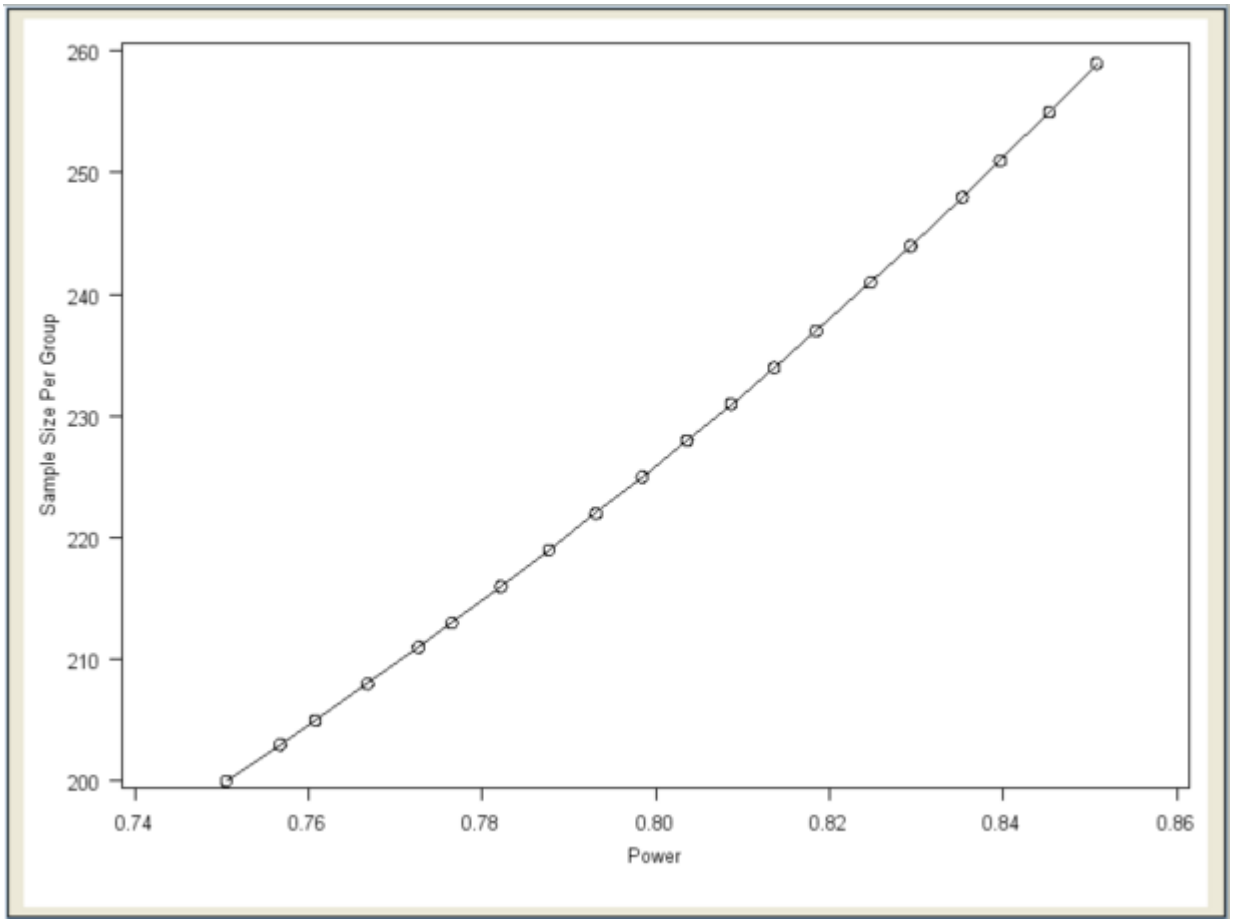
Figure 68.88 Summary Table



Power by Sample Size Graph

Click the **Graph** tab to view the power by sample size graph.

Figure 68.89 Power by Sample Size Graph



As you can see in Figure 68.89, the graph is curved slightly upward with larger powers associated with larger sample sizes. Sample size is plotted on the vertical axis as requested in the Customize Graph window.

Narratives

Click the **Narratives** tab to create one or more narratives. To generate a narrative, select the single scenario in the narrative selector table at the bottom of the tab. The narrative for this task does not include the survival times and probabilities for the survival curves:

```
For a log-rank test comparing two survival curves with a two-sided
significance level of 0.05, assuming uniform accrual with an
accrual time of 2 and a follow-up time of 3, a sample size of 226
per group is required to obtain a power of at least 0.8 for the
exponential curve, "Existing treatment," and the piecewise linear
curve, "Proposed treatment."  The actual power is 0.800.
```

For information about selecting additional narratives when multiple scenarios are present, see the section "Creating Narratives" on page 5082.

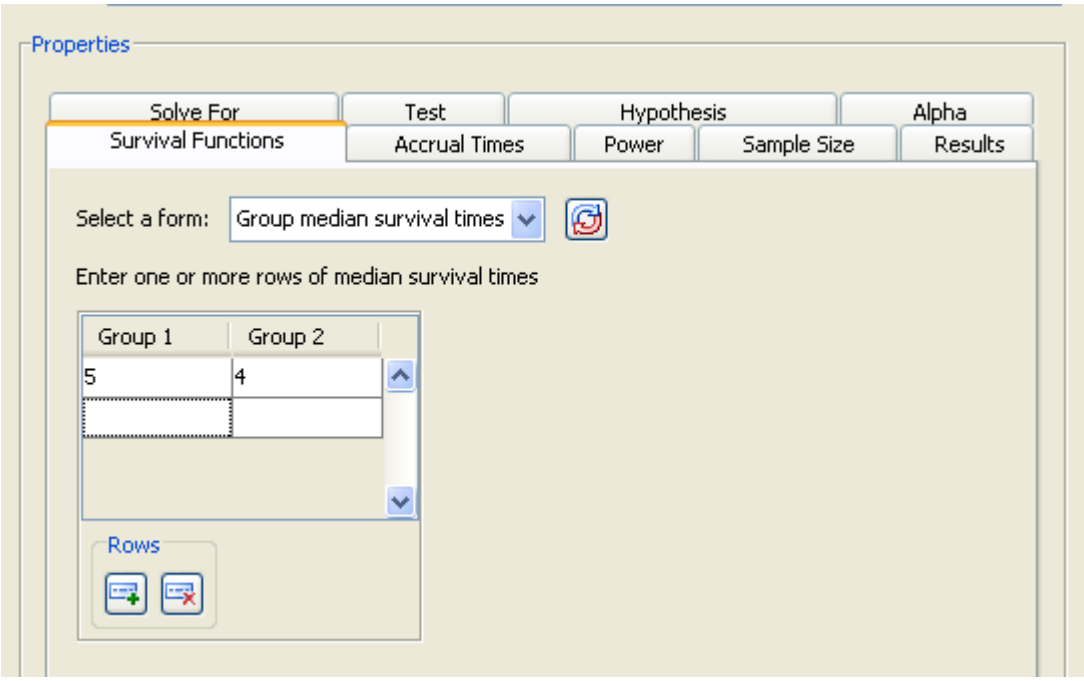
Additional Topics

Using the Other Survival Curve Forms

Survival functions can be specified as median survival times, hazards, or a combination of hazards for one group and hazard ratios. These all assume exponential curves.

Suppose you are interested in comparing the proposed and existing treatments using their median survival times. The survival times are five years and four years for the two groups, respectively.

Figure 68.90 Median Survival Times and List of Alternate Forms



Click the **Survival Functions** tab and examine the list of alternate forms available in the **Select a form:** list. For this example, select the **Group median survival times** option.

For the example, enter `5` and `4` in the first row of the table. The completed table is shown in Figure 68.90.

You can enter one or more sets of two median survival times. The results of the analysis are not shown.

Chapter 69

The PRINCOMP Procedure

Contents

Overview: PRINCOMP Procedure **5143**
Getting Started: PRINCOMP Procedure **5145**
Syntax: PRINCOMP Procedure **5150**
PROC PRINCOMP Statement 5151
BY Statement 5156
FREQ Statement 5157
ID Statement 5157
PARTIAL Statement 5157
VAR Statement 5158
WEIGHT Statement 5158
Details: PRINCOMP Procedure **5158**
Missing Values 5158
Output Data Sets 5158
Computational Resources 5161
Displayed Output 5162
ODS Table Names 5162
ODS Graphics 5163
Examples: PRINCOMP Procedure **5164**
Example 69.1: Temperatures 5164
Example 69.2: Basketball Data 5168
Example 69.3: Job Ratings 5175
References **5192**

Overview: PRINCOMP Procedure

The PRINCOMP procedure performs principal component analysis. As input you can use raw data, a correlation matrix, a covariance matrix, or a sum-of-squares-and-crossproducts (SSCP) matrix. You can create output data sets containing eigenvalues, eigenvectors, and standardized or unstandardized principal component scores.

Principal component analysis is a multivariate technique for examining relationships among several quantitative variables. The choice between using factor analysis and using principal component

analysis depends in part on your research objectives. You should use the PRINCOMP procedure if you are interested in summarizing data and detecting linear relationships. You can use principal components to reduce the number of variables in regression, clustering, and so on. See Chapter 9, "Introduction to Multivariate Procedures," for a detailed comparison of the PRINCOMP and FACTOR procedures.

You can use ODS Graphics to display the scree plot, component pattern plot, component pattern profile plot, matrix plot of component scores, and component score plots. These plots are especially valuable tools in exploratory data analysis.

Principal component analysis was originated by Pearson (1901) and later developed by Hotelling (1933). The application of principal components is discussed by Rao (1964), Cooley and Lohnes (1971), and Gnanadesikan (1977). Excellent statistical treatments of principal components are found in Kshirsagar (1972), Morrison (1976), and Mardia, Kent, and Bibby (1979).

Given a data set with p numeric variables, you can compute p principal components. Each principal component is a linear combination of the original variables, with coefficients equal to the eigenvectors of the correlation or covariance matrix. The eigenvectors are customarily taken with unit length. The principal components are sorted by descending order of the eigenvalues, which are equal to the variances of the components.

Principal components have a variety of useful properties (Rao 1964; Kshirsagar 1972):

- The eigenvectors are orthogonal, so the principal components represent jointly perpendicular directions through the space of the original variables.
- The principal component scores are jointly uncorrelated. Note that this property is quite distinct from the previous one.
- The first principal component has the largest variance of any unit-length linear combination of the observed variables. The jth principal component has the largest variance of any unit-length linear combination orthogonal to the first $j-1$ principal components. The last principal component has the smallest variance of any linear combination of the original variables.
- The scores on the first j principal components have the highest possible generalized variance of any set of unit-length linear combinations of the original variables.
- The first j principal components provide a least squares solution to the model

  $$\mathbf{Y} = \mathbf{XB} + \mathbf{E}$$

  where $\mathbf{Y}$ is an $n \times p$ matrix of the centered observed variables; $\mathbf{X}$ is the $n \times j$ matrix of scores on the first j principal components; $\mathbf{B}$ is the $j \times p$ matrix of eigenvectors; $\mathbf{E}$ is an $n \times p$ matrix of residuals; and you want to minimize trace($\mathbf{E}'\mathbf{E}$), the sum of all the squared elements in $\mathbf{E}$. In other words, the first j principal components are the best linear predictors of the original variables among all possible sets of j variables, although any nonsingular linear transformation of the first j principal components would provide equally good prediction. The same result is obtained if you want to minimize the determinant or the Euclidean (Schur, Frobenious) norm of $\mathbf{E}'\mathbf{E}$ rather than the trace.
- In geometric terms, the j-dimensional linear subspace spanned by the first j principal components provides the best possible fit to the data points as measured by the sum of squared

perpendicular distances from each data point to the subspace. This is in contrast to the geometric interpretation of least squares regression, which minimizes the sum of squared vertical distances. For example, suppose you have two variables. Then, the first principal component minimizes the sum of squared perpendicular distances from the points to the first principal axis. This is in contrast to least squares, which would minimize the sum of squared vertical distances from the points to the fitted line.

Principal component analysis can also be used for exploring polynomial relationships and for multivariate outlier detection (Gnanadesikan 1977), and it is related to factor analysis, correspondence analysis, allometry, and biased regression techniques (Mardia, Kent, and Bibby 1979).

Getting Started: PRINCOMP Procedure

The following data provide crime rates per 100,000 people in seven categories for each of the 50 states in 1977. Since there are seven numeric variables, it is impossible to plot all the variables simultaneously. Principal components can be used to summarize the data in two or three dimensions, and they help to visualize the data. The following statements produce Figure 69.1 through Figure 69.5.

```
data Crime;
   input State $1-15 Murder Rape Robbery Assault
         Burglary Larceny Auto_Theft;
   datalines;
Alabama         14.2 25.2  96.8 278.3 1135.5 1881.9 280.7
Alaska          10.8 51.6  96.8 284.0 1331.7 3369.8 753.3
Arizona          9.5 34.2 138.2 312.3 2346.1 4467.4 439.5
Arkansas         8.8 27.6  83.2 203.4  972.6 1862.1 183.4
California      11.5 49.4 287.0 358.0 2139.4 3499.8 663.5
Colorado         6.3 42.0 170.7 292.9 1935.2 3903.2 477.1
Connecticut      4.2 16.8 129.5 131.8 1346.0 2620.7 593.2
Delaware         6.0 24.9 157.0 194.2 1682.6 3678.4 467.0
Florida         10.2 39.6 187.9 449.1 1859.9 3840.5 351.4
Georgia         11.7 31.1 140.5 256.5 1351.1 2170.2 297.9
Hawaii           7.2 25.5 128.0  64.1 1911.5 3920.4 489.4
Idaho            5.5 19.4  39.6 172.5 1050.8 2599.6 237.6
Illinois         9.9 21.8 211.3 209.0 1085.0 2828.5 528.6
Indiana          7.4 26.5 123.2 153.5 1086.2 2498.7 377.4
Iowa             2.3 10.6  41.2  89.8  812.5 2685.1 219.9
Kansas           6.6 22.0 100.7 180.5 1270.4 2739.3 244.3
Kentucky        10.1 19.1  81.1 123.3  872.2 1662.1 245.4
Louisiana       15.5 30.9 142.9 335.5 1165.5 2469.9 337.7
Maine            2.4 13.5  38.7 170.0 1253.1 2350.7 246.9
Maryland         8.0 34.8 292.1 358.9 1400.0 3177.7 428.5
Massachusetts    3.1 20.8 169.1 231.6 1532.2 2311.3 1140.1
Michigan         9.3 38.9 261.9 274.6 1522.7 3159.0 545.5
Minnesota        2.7 19.5  85.9  85.8 1134.7 2559.3 343.1
Mississippi     14.3 19.6  65.7 189.1  915.6 1239.9 144.4
```

```
Missouri        9.6 28.3 189.0 233.5 1318.3 2424.2 378.4
Montana         5.4 16.7  39.2 156.8  804.9 2773.2 309.2
Nebraska        3.9 18.1  64.7 112.7  760.0 2316.1 249.1
Nevada         15.8 49.1 323.1 355.0 2453.1 4212.6 559.2
New Hampshire   3.2 10.7  23.2  76.0 1041.7 2343.9 293.4
New Jersey      5.6 21.0 180.4 185.1 1435.8 2774.5 511.5
New Mexico      8.8 39.1 109.6 343.4 1418.7 3008.6 259.5
New York       10.7 29.4 472.6 319.1 1728.0 2782.0 745.8
North Carolina 10.6 17.0  61.3 318.3 1154.1 2037.8 192.1
North Dakota    0.9  9.0  13.3  43.8  446.1 1843.0 144.7
Ohio            7.8 27.3 190.5 181.1 1216.0 2696.8 400.4
Oklahoma        8.6 29.2  73.8 205.0 1288.2 2228.1 326.8
Oregon          4.9 39.9 124.1 286.9 1636.4 3506.1 388.9
Pennsylvania    5.6 19.0 130.3 128.0  877.5 1624.1 333.2
Rhode Island    3.6 10.5  86.5 201.0 1489.5 2844.1 791.4
South Carolina 11.9 33.0 105.9 485.3 1613.6 2342.4 245.1
South Dakota    2.0 13.5  17.9 155.7  570.5 1704.4 147.5
Tennessee      10.1 29.7 145.8 203.9 1259.7 1776.5 314.0
Texas          13.3 33.8 152.4 208.2 1603.1 2988.7 397.6
Utah            3.5 20.3  68.8 147.3 1171.6 3004.6 334.5
Vermont         1.4 15.9  30.8 101.2 1348.2 2201.0 265.2
Virginia        9.0 23.3  92.1 165.7  986.2 2521.2 226.7
Washington      4.3 39.6 106.2 224.8 1605.6 3386.9 360.3
West Virginia   6.0 13.2  42.2  90.9  597.4 1341.7 163.3
Wisconsin       2.8 12.9  52.2  63.7  846.9 2614.2 220.7
Wyoming         5.4 21.9  39.7 173.9  811.6 2772.2 282.0
;

ods graphics on;
title 'Crime Rates per 100,000 Population by State';
proc princomp out=Crime_Components plots= score(ellipse ncomp=3);
   id State;
run;
ods graphics off;
```

Figure 69.1 displays the PROC PRINCOMP output, beginning with simple statistics followed by the correlation matrix. The PROC PRINCOMP statement requests by default principal components computed from the correlation matrix, so the total variance is equal to the number of variables, 7.

Figure 69.1 Number of Observations and Simple Statistics from the PRINCOMP Procedure

```
                Crime Rates per 100,000 Population by State

                        The PRINCOMP Procedure

                      Observations          50
                      Variables              7
```

Figure 69.1 *continued*

Simple Statistics

	Murder	Rape	Robbery	Assault
Mean	7.444000000	25.73400000	124.0920000	211.3000000
StD	3.866768941	10.75962995	88.3485672	100.2530492

Simple Statistics

	Burglary	Larceny	Auto_Theft
Mean	1291.904000	2671.288000	377.5260000
StD	432.455711	725.908707	193.3944175

Correlation Matrix

	Murder	Rape	Robbery	Assault	Burglary	Larceny	Auto_ Theft
Murder	1.0000	0.6012	0.4837	0.6486	0.3858	0.1019	0.0688
Rape	0.6012	1.0000	0.5919	0.7403	0.7121	0.6140	0.3489
Robbery	0.4837	0.5919	1.0000	0.5571	0.6372	0.4467	0.5907
Assault	0.6486	0.7403	0.5571	1.0000	0.6229	0.4044	0.2758
Burglary	0.3858	0.7121	0.6372	0.6229	1.0000	0.7921	0.5580
Larceny	0.1019	0.6140	0.4467	0.4044	0.7921	1.0000	0.4442
Auto_Theft	0.0688	0.3489	0.5907	0.2758	0.5580	0.4442	1.0000

Figure 69.2 displays the eigenvalues. The first principal component explains about 58.8% of the total variance, the second principal component explains about 17.7%, and the third principal component explains about 10.4%. Note that the eigenvalues sum to the total variance.

The eigenvalues indicate that two or three components provide a good summary of the data, two components accounting for 76% of the total variance and three components explaining 87%. Subsequent components contribute less than 5% each.

Figure 69.2 Results of Principal Component Analysis: PROC PRINCOMP

Eigenvalues of the Correlation Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	4.11495951	2.87623768	0.5879	0.5879
2	1.23872183	0.51290521	0.1770	0.7648
3	0.72581663	0.40938458	0.1037	0.8685
4	0.31643205	0.05845759	0.0452	0.9137
5	0.25797446	0.03593499	0.0369	0.9506
6	0.22203947	0.09798342	0.0317	0.9823
7	0.12405606		0.0177	1.0000

Figure 69.3 displays the eigenvectors. From the eigenvectors matrix, you can represent the first principal component Prin1 as a linear combination of the original variables:

$$\begin{aligned}\text{Prin1} = {} & 0.300279 \times (\text{Murder}) \\ & + 0.431759 \times (\text{Rape}) \\ & + 0.396875 \times (\text{Robbery}) \\ & \cdot \\ & \cdot \\ & \cdot \\ & + 0.295177 \times (\text{Auto\_Theft})\end{aligned}$$

Similarly, the second principal component Prin2 is

$$\begin{aligned}\text{Prin2} = {} & -0.629174 \times (\text{Murder}) \\ & - 0.169435 \times (\text{Rape}) \\ & + 0.042247 \times (\text{Robbery}) \\ & \cdot \\ & \cdot \\ & \cdot \\ & - 0.502421 \times (\text{Auto\_Theft})\end{aligned}$$

where the variables are standardized.

Figure 69.3 Results of Principal Component Analysis: PROC PRINCOMP

Eigenvectors

	Prin1	Prin2	Prin3	Prin4	Prin5	Prin6	Prin7
Murder	0.300279	-.629174	0.178245	-.232114	0.538123	0.259117	0.267593
Rape	0.431759	-.169435	-.244198	0.062216	0.188471	-.773271	-.296485
Robbery	0.396875	0.042247	0.495861	-.557989	-.519977	-.114385	-.003903
Assault	0.396652	-.343528	-.069510	0.629804	-.506651	0.172363	0.191745
Burglary	0.440157	0.203341	-.209895	-.057555	0.101033	0.535987	-.648117
Larceny	0.357360	0.402319	-.539231	-.234890	0.030099	0.039406	0.601690
Auto_Theft	0.295177	0.502421	0.568384	0.419238	0.369753	-.057298	0.147046

The first component is a measure of the overall crime rate since the first eigenvector shows approximately equal loadings on all variables. The second eigenvector has high positive loadings on variables Auto_Theft and Larceny and high negative loadings on variables Murder and Assault. There is also a small positive loading on Burglary and a small negative loading on Rape. This component seems to measure the preponderance of property crime over violent crime. The interpretation of the third component is not obvious.

The ODS GRAPHICS statement enables the PRINCOMP procedure to produce statistical graphs by using ODS Graphics. See Chapter 21, "Statistical Graphics Using ODS," for more information.

PLOTS=SCORE(ELLIPSE NCOMP=3) in the PROC PRINCOMP statement requests the pairwise component score plots for the first three components with a 95% prediction ellipse overlaid on each of the scatter plot. Figure 69.4 shows the plot of the first two components. It is possible to identify regional trends on the plot of the first two components. Nevada and California are at the extreme right, with high overall crime rates but an average ratio of property crime to violent crime. North and South Dakota are at the extreme left, with low overall crime rates. Southeastern states tend to be at the bottom of the plot, with a higher-than-average ratio of violent crime to property crime. New England states tend to be in the upper part of the plot, with a higherr-than-average ratio of property crime to violent crime. Assuming the first two components are from a bivariate normal distribution, the ellipse identifies Nevada as a possible outlier.

Figure 69.4 Plot of the First Two Component Scores

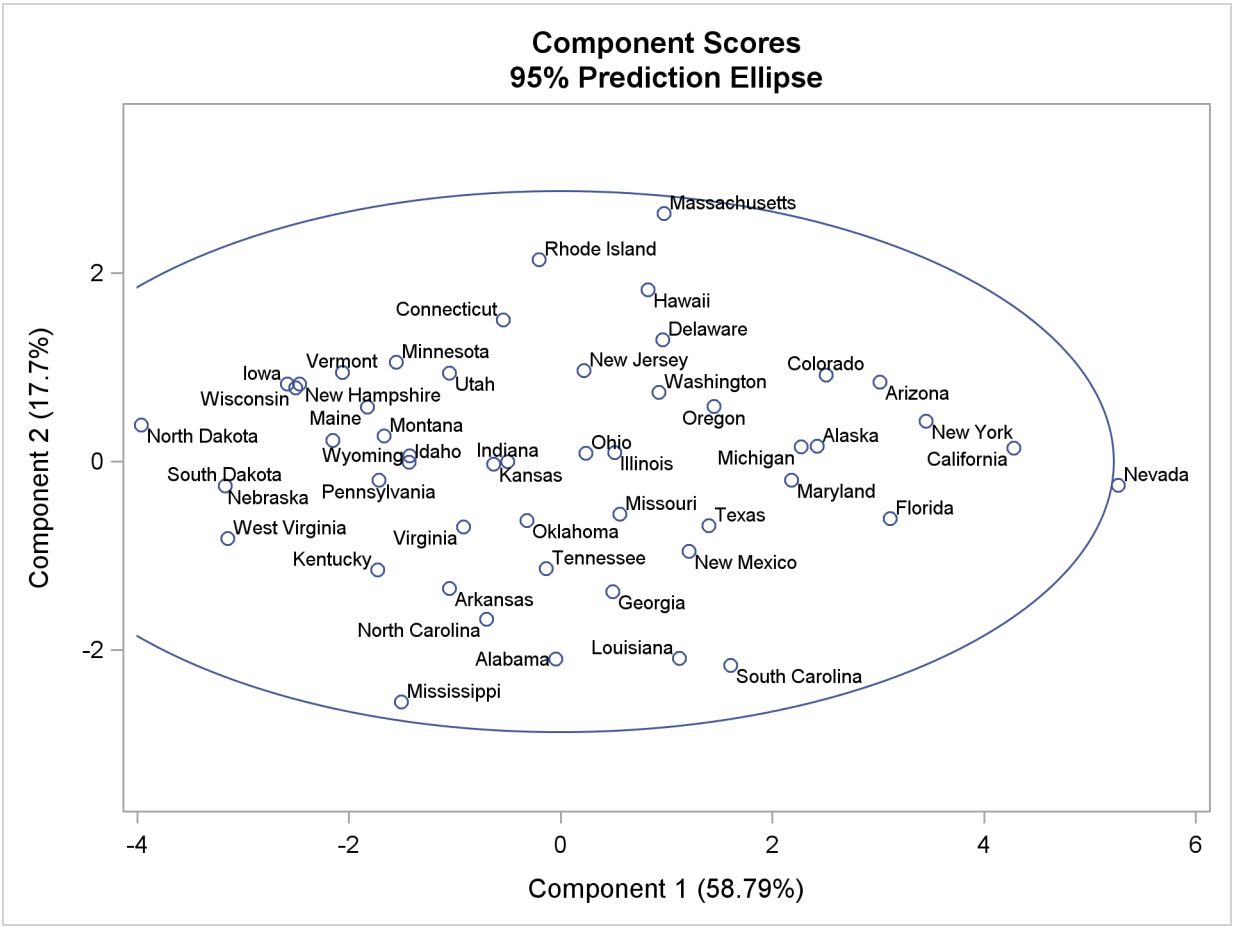
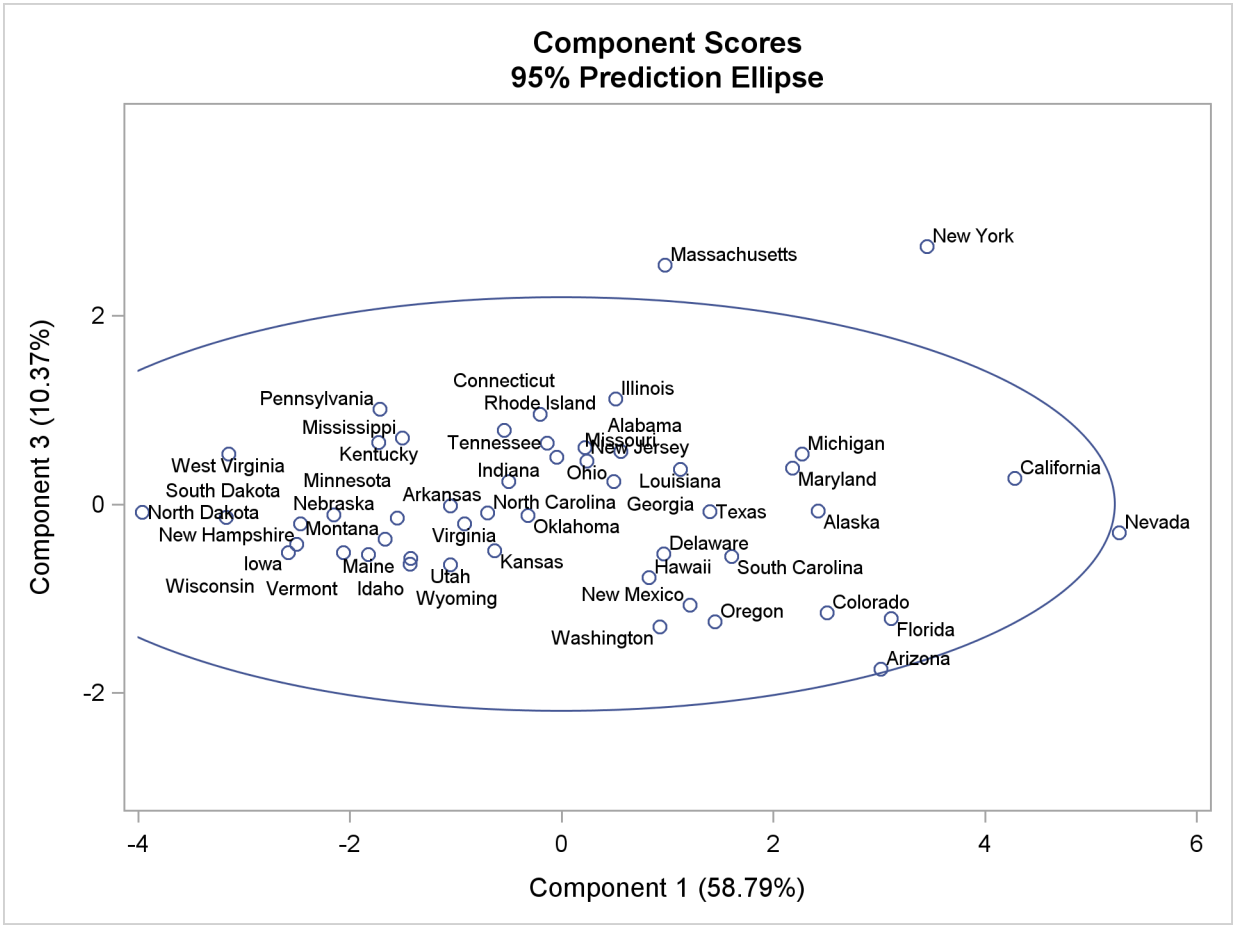


Figure 69.5 shows the plot of the first and third components. Assuming the first and the third components are from a bivariate normal distribution, the ellipse identifies Nevada, Massachusetts, and New York as possible outliers.

Figure 69.5 Plot of the First and Third Component Scores



The most striking feature of the plot of the first and third principal components is that Massachusetts and New York are outliers on the third component.

Syntax: PRINCOMP Procedure

The following statements are available in PROC PRINCOMP:

PROC PRINCOMP < *options* > ;
BY *variables* ;
FREQ *variable* ;
ID *variables* ;
PARTIAL *variables* ;
VAR *variables* ;
WEIGHT *variable* ;

Usually only the VAR statement is used in addition to the PROC PRINCOMP statement. The rest of this section provides detailed syntax information for each of the preceding statements, beginning

with the PROC PRINCOMP statement. The remaining statements are described in alphabetical order.

PROC PRINCOMP Statement

PROC PRINCOMP < *options* > **;**

The PROC PRINCOMP statement starts the PRINCOMP procedure and optionally identifies input and output data sets, specifies the analyses performed, and controls displayed output. Table 69.1 summarizes the options.

Table 69.1 Summary of PROC PRINCOMP Statement Options

Option	Description
Specify data sets	
DATA=	specifies input data set name
OUT=	specifies output data set name
OUTSTAT=	specifies output data set name containing various statistics
Specify details of analysis	
COV	computes the principal components from the covariance matrix
N=	specifies the number of principal components to be computed
NOINT	omits the intercept from the model
PREFIX=	specifies a prefix for naming the principal components
RPREFIX=	specifies a prefix for naming the residual variables
SINGULAR=	specifies the singularity criterion
STD	standardizes the principal component scores
VARDEF=	specifies the divisor used in calculating variances and standard deviations
Suppress the display of output	
NOPRINT	suppresses the display of all output
Specify ODS Graphics details	
PLOTS=	specifies options that control the details of the plots

The following list provides details about these options.

COVARIANCE

COV

computes the principal components from the covariance matrix. If you omit the COV option, the correlation matrix is analyzed. Use of the COV option causes variables with large variances to be more strongly associated with components with large eigenvalues and causes variables with small variances to be more strongly associated with components with small eigenvalues. You should not specify the COV option unless the units in which the variables are measured are comparable or the variables are standardized in some way.

DATA=*SAS-data-set*

specifies the SAS data set to be analyzed. The data set can be an ordinary SAS data set or a TYPE=ACE, TYPE=CORR, TYPE=COV, TYPE=FACTOR, TYPE=SSCP, TYPE=UCORR, or TYPE=UCOV data set (see Appendix A, "Special SAS Data Sets"). Also, the PRINCOMP procedure can read the _TYPE_='COVB' matrix from a TYPE=EST data set. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

N=*number*

specifies the number of principal components to be computed. The default is the number of variables. The value of the N= option must be an integer greater than or equal to zero.

NOINT

omits the intercept from the model. In other words, the NOINT option requests that the covariance or correlation matrix not be corrected for the mean. When you use the PRINCOMP procedure with the NOINT option, the covariance matrix and, hence, the standard deviations are not corrected for the mean. If you are interested in the standard deviations corrected for the mean, you can get them by using a procedure such as the MEANS procedure.

If you use a TYPE=SSCP data set as input to the PRINCOMP procedure and list the variable Intercept in the VAR statement, the procedure acts as if you had also specified the NOINT option. If you use NOINT and also create an OUTSTAT= data set, the data set is TYPE=UCORR or TYPE=UCOV rather than TYPE=CORR or TYPE=COV.

NOPRINT

suppresses the display of all output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 20, "Using the Output Delivery System."

OUT=*SAS-data-set*

creates an output SAS data set that contains all the original data as well as the principal component scores.

If you want to create a permanent SAS data set, you must specify a two-level name. For details about OUT= data sets, see the section "Output Data Sets" on page 5158. See *SAS Language Reference: Concepts* for more information about permanent SAS data sets.

OUTSTAT=*SAS-data-set*

creates an output SAS data set that contains means, standard deviations, number of observations, correlations or covariances, eigenvalues, and eigenvectors. If you specify the COV option, the data set is TYPE=COV or TYPE=UCOV, depending on the NOINT option, and it contains covariances; otherwise, the data set is TYPE=CORR or TYPE=UCORR, depending on the NOINT option, and it contains correlations. If you specify the PARTIAL statement, the OUTSTAT= data set contains R squares as well.

If you want to create a permanent SAS data set, you must specify a two-level name. For details about OUTSTAT= data sets, see the section "Output Data Sets" on page 5158. See *SAS Language Reference: Concepts* for more information about permanent SAS data sets.

PLOTS < **(***global-plot-options***)** > < **=** *plot-request* < *(options)* > >

PLOTS < **(***global-plot-options***)** > < **= (***plot-request* < *(options)* > < **...** *plot-request* < *(options)* > **>) >**

controls the plots produced through ODS Graphics. When you specify only one plot request, you can omit the parentheses around the plot request. Here are some examples:

```
plots=none
plots=(scatter pattern)
plots(unpack)=scree
plots(ncomp=3 flip)=(pattern(circles=0.5 1.0) score)
```

You must enable ODS Graphics before requesting plots—for example, like this:

```
ods graphics on;
proc princomp plots=all;
   var x1--x10;
run;
ods graphics off;
```

For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." If you have enabled ODS Graphics but do not specify the PLOTS= option, PROC PRINCOMP produces the scree plot by default.

The global plot options include the following:

FLIP

flips or interchanges the X-axis and Y-axis dimension for the component score plots and the component pattern plots. For example, if there are three components, the default plots (y * x) are Component 2 * Component 1, Component 3 * Component 1, and Component 3 * Component 2. When you specify PLOTS(FLIP), the plots are Component 1 * Component 2, Component 1 * Component 3, and Component 2 * Component 3.

NCOMP=*n*

specifies the number of components $n(\geq 2)$ to be plotted for the component pattern plots and the component score plots. If the NCOMP= option is again specified in an individual plot, such as PLOTS=SCORE(NCOMP= m), the value m will determine the number of components to be plotted in the component score plots. Be aware that the number of plots ($\frac{n\times(n-1)}{2}$) produced grows quadratically when n increases. The default is 5 or the total number of components $m(\geq 2)$, whichever is smaller. If $n > m$, NCOMP= m will be used.

ONLY

suppresses the default plots. Only plots specifically requested are displayed.

UNPACKPANEL

UNPACK

suppresses paneling in the scree plot. By default, multiple plots can appear in an output panel. Specify UNPACKPANEL to get each plot in a separate panel. You can specify PLOTS(UNPACKPANEL) to unpack the default plots. You can also specify UNPACKPANEL as a suboption with SCREE (such as PLOTS=SCREE(UNPACKPANEL)).

The plot requests include the following:

ALL
produces all appropriate plots. You can specify other options with ALL; for example, to request all plots and unpack only the scree plot, specify PLOTS=(ALL SCREE(UNPACKPANEL)).

EIGEN | EIGENVALUE | SCREE < (UNPACKPANEL) >
produces the scree plot of eigenvalues and proportion variance explained. By default, both plots are output in a panel. Specify PLOTS= SCREE(UNPACKPANEL) to get each plot in a separate panel.

MATRIX
produces the matrix plot of principal component scores.

NONE
suppresses the display of all graphics output.

PATTERN < (*pattern-options*) >
produces the pairwise component pattern plots. Each variable is plotted as an observation whose coordinates are correlations between the variable and the two corresponding components on the plot. Use the NCOMP= option (for instance, PLOTS=PATTERN(NCOMP=3)) described in the following to control the number of plots to be displayed.

The available *pattern-options* are as follows:

CIRCLES < = *number list* >
plots the variance percentage circles. Each number in the list must be greater than 0. If the number is greater than or equal to 1, it is interpreted as a percentage and divided by 100; CIRCLES=0.05 and CIRCLES=5 are equivalent. For each number (c) specified, a ($c \times 100\%$) variance circle is created.

By default, there is no circle for the scatter pattern plot (PLOTS=PATTERN) and a unit circle with a 100% variance circle is plotted for the vector pattern plot (PLOTS=PATTERN (VECTOR)). You can display multiple circles by specifying PLOTS=PATTERN(CIRCLES=). For example, specifying PLOTS=PATTERN(CIRCLES= .3 .6 1.0) will display the 30%, 60%, and 100% variance circles in the pattern plots.

FLIP
flips or interchanges the X-axis and Y-axis dimensions for the component pattern plots. Specify PLOTS=PATTERN(FLIP) to flip the X-axis and Y-axis dimensions.

NCOMP=*n*
specifies the number of components $n(\geq 2)$ to be plotted. The default is 5 or the total number of components $m(\geq 2)$, whichever is smaller. If $n > m$, NCOMP= m will be used. Be aware that the number of plots ($\frac{n \times (n-1)}{2}$) produced grows quadratically when n increases.

VECTOR

plots pattern in a vector form.

PATTERNPROFILE | PROFILE

produces the pattern profile plot. There is a profile for each component. The Y-axis value represents the correlation between the variable (corresponding to the X-axis value) and the profiled principal component.

SCORE < (*score-options*) >

produces the pairwise component score plots. Use the NCOMP= option (for instance, PLOTS=SCORE(NCOMP=3)) described in the following to control the number of plots to be displayed.

The available *score-options* are as follows:

ALPHA=*number list*

specifies a list of numbers for the prediction ellipses to be displayed in the score plots. Each value (α) in the list must be greater than 0. If α is greater than or equal to 1, it is interpreted as a percentage and divided by 100; ALPHA=0.05 and ALPHA=5 are equivalent.

ELLIPSE

requests prediction ellipses for the principal component scores of a new observation to be created in the principal component score plots. See the section "Confidence and Prediction Ellipses" in "The CORR Procedure" (*Base SAS Procedures Guide: Statistical Procedures*), for details about the computation of a prediction ellipse.

FLIP

flips or interchanges the X-axis and Y-axis dimensions for the component score plots. Specify PLOTS=SCORE(FLIP) to flip the X-axis and Y-axis dimensions.

NCOMP=*n*

specifies the number of components $n(\geq 2)$ to be plotted. The default is 5 or the total number of components $m(\geq 2)$, whichever is smaller. If $n > m$, NCOMP= m will be used. Be aware that the number of plots ($\frac{n\times(n-1)}{2}$) produced grows quadratically when n increases.

PREFIX=*name*

specifies a prefix for naming the principal components. By default, the names are Prin1, Prin2, ..., Prin$n$. If you specify PREFIX=ABC, the components are named ABC1, ABC2, ABC3, and so on. The number of characters in the prefix plus the number of digits required to designate the variables should not exceed the current name length defined by the VALIDVARNAME= system option.

PARPREFIX=*name*

PPREFIX=*name*

specifies a prefix for naming the residual variables in the OUT= data set and the OUTSTAT= data set. By default, the prefix is R_. The number of characters in the prefix plus the maximum length of the variable names should not exceed the current name length defined by the VALIDVARNAME= system option.

SINGULAR=*p*

SING=*p*

specifies the singularity criterion, where $0 < p < 1$. If a variable in a PARTIAL statement has an R square as large as $1-p$ when predicted from the variables listed before it in the statement, the variable is assigned a standardized coefficient of 0. By default, SINGULAR=1E−8.

STANDARD

STD

standardizes the principal component scores in the OUT= data set to unit variance. If you omit the STANDARD option, the scores have variance equal to the corresponding eigenvalue. Note that STANDARD has no effect on the eigenvalues themselves.

VARDEF=DF | N | WDF | WEIGHT | WGT

specifies the divisor used in calculating variances and standard deviations. By default, VARDEF=DF. The following table displays the values and associated divisors.

Value	Divisor		Formula
DF	error degrees of freedom	$n - i$	(before partialing)
		$n - p - i$	(after partialing)
N	number of observations	n	
WEIGHT | WGT	sum of weights	$\sum_{j=1}^{n} w_j$	
WDF	sum of weights minus one	$\left(\sum_{j=1}^{n} w_j\right) - i$	(before partialing)
		$\left(\sum_{j=1}^{n} w_j\right) - p - i$	(after partialing)

In the formulas for VARDEF=DF and VARDEF=WDF, p is the number of degrees of freedom of the variables in the PARTIAL statement, and i is 0 if the NOINT option is specified and 1 otherwise.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC PRINCOMP to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC PRINCOMP. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.

- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide.*

FREQ Statement

FREQ *variable* **;**

The FREQ statement specifies a variable that provides frequencies for each observation in the DATA= data set. Specifically, if n is the value of the FREQ variable for a given observation, then that observation is used n times.

The analysis produced using a FREQ statement reflects the expanded number of observations. The total number of observations is considered equal to the sum of the FREQ variable. You could produce the same analysis (without the FREQ statement) by first creating a new data set that contains the expanded number of observations. For example, if the value of the FREQ variable is 5 for the first observation, the first 5 observations in the new data set would be identical. Each observation in the old data set would be replicated n_j times in the new data set, where n_j is the value of the FREQ variable for that observation.

If the value of the FREQ variable is missing or is less than one, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

ID Statement

ID *variables* **;**

The ID statement labels observations with values from the first ID variable in the principal component score plot. If one or more ID variables are specified, their values are displayed in tooltips of the component score plot and the matrix plot of component scores.

PARTIAL Statement

PARTIAL *variables* **;**

If you want to analyze a partial correlation or covariance matrix, specify the names of the numeric variables to be partialed out in the PARTIAL statement. The PRINCOMP procedure computes the principal components of the residuals from the prediction of the VAR variables by the PARTIAL variables. If you request an OUT= or OUTSTAT= data set, the residual variables are named by prefixing the characters R_ by default or the string specified in the RPREFIX= option to the VAR variables.

VAR Statement

VAR *variables* **;**

The VAR statement lists the numeric variables to be analyzed. If you omit the VAR statement, all numeric variables not specified in other statements are analyzed. If, however, the DATA= data set is TYPE=SSCP, the default set of variables used as VAR variables does not include Intercept so that the correlation or covariance matrix is constructed correctly. If you want to analyze Intercept as a separate variable, you should specify it in the VAR statement.

WEIGHT Statement

WEIGHT *variable* **;**

If you want to use relative weights for each observation in the input data set, place the weights in a variable in the data set and specify the name in a WEIGHT statement. This is often done when the variance associated with each observation is different and the values of the weight variable are proportional to the reciprocals of the variances.

The observation is used in the analysis only if the value of the WEIGHT statement variable is nonmissing and is greater than zero.

Details: PRINCOMP Procedure

Missing Values

Observations with missing values for any variable in the VAR, PARTIAL, FREQ, or WEIGHT statement are omitted from the analysis and are given missing values for principal component scores in the OUT= data set. If a correlation, covariance, or SSCP matrix is read, it can contain missing values as long as every pair of variables has at least one nonmissing entry.

Output Data Sets

OUT= Data Set

The OUT= data set contains all the variables in the original data set plus new variables containing the principal component scores. The N= option determines the number of new variables. The

names of the new variables are formed by concatenating the value given by the PREFIX= option (or Prin if PREFIX= is omitted) and the numbers 1, 2, 3, and so on. The new variables have mean 0 and variance equal to the corresponding eigenvalue, unless you specify the STANDARD option to standardize the scores to unit variance. Also, if you specify the COV option, the procedure computes the principal component scores from the corrected or the uncorrected (if the NOINT option is specified) variables rather than the standardized variables.

If you use a PARTIAL statement, the OUT= data set also contains the residuals from predicting the VAR variables from the PARTIAL variables.

An OUT= data set cannot be created if the DATA= data set is TYPE=ACE, TYPE=CORR, TYPE=COV, TYPE=EST, TYPE=FACTOR, TYPE=SSCP, TYPE=UCORR, or TYPE=UCOV.

OUTSTAT= Data Set

The OUTSTAT= data set is similar to the TYPE=CORR data set produced by the CORR procedure. The following table relates the TYPE= value for the OUTSTAT= data set to the options specified in the PROC PRINCOMP statement.

Options	**TYPE=**
(default)	CORR
COV	COV
NOINT	UCORR
COV NOINT	UCOV

Note that the default (neither the COV nor NOINT option) produces a TYPE=CORR data set.

The new data set contains the following variables:

- the BY variables, if any
- two new variables, _TYPE_ and _NAME_, both character variables
- the variables analyzed (that is, those in the VAR statement); or, if there is no VAR statement, all numeric variables not listed in any other statement; or, if there is a PARTIAL statement, the residual variables as described under the OUT= data set

Each observation in the new data set contains some type of statistic as indicated by the _TYPE_ variable. The values of the _TYPE_ variable are as follows:

MEAN — mean of each variable. If you specify the PARTIAL statement, this observation is omitted.

STD — standard deviations. If you specify the COV option, this observation is omitted, so the SCORE procedure does not standardize the variables before computing scores. If you use the PARTIAL statement, the standard deviation of a variable is computed as its root mean squared error as predicted from the PARTIAL variables.

USTD — uncorrected standard deviations. When you specify the NOINT option in the PROC PRINCOMP statement, the OUTSTAT= data set contains standard deviations not corrected for the mean. However, if you also specify the COV option in the PROC PRINCOMP statement, this observation is omitted.

N — number of observations on which the analysis is based. This value is the same for each variable. If you specify the PARTIAL statement and the value of the VARDEF= option is DF or unspecified, then the number of observations is decremented by the degrees of freedom for the PARTIAL variables.

SUMWGT — the sum of the weights of the observations. This value is the same for each variable. If you specify the PARTIAL statement and VARDEF=WDF, then the sum of the weights is decremented by the degrees of freedom for the PARTIAL variables. This observation is output only if the value is different from that in the observation with _TYPE_='N'.

CORR — correlations between each variable and the variable specified by the _NAME_ variable. The number of observations with _TYPE_='CORR' is equal to the number of variables being analyzed. If you specify the COV option, no _TYPE_='CORR' observations are produced. If you use the PARTIAL statement, the partial correlations, not the raw correlations, are output.

UCORR — uncorrected correlation matrix. When you specify the NOINT option without the COV option in the PROC PRINCOMP statement, the OUTSTAT= data set contains a matrix of correlations not corrected for the means. However, if you also specify the COV option in the PROC PRINCOMP statement, this observation is omitted.

COV — covariances between each variable and the variable specified by the _NAME_ variable. _TYPE_='COV' observations are produced only if you specify the COV option. If you use the PARTIAL statement, the partial covariances, not the raw covariances, are output.

UCOV — uncorrected covariance matrix. When you specify the NOINT and COV options in the PROC PRINCOMP statement, the OUTSTAT= data set contains a matrix of covariances not corrected for the means.

EIGENVAL — eigenvalues. If the N= option requested fewer than the maximum number of principal components, only the specified number of eigenvalues are produced, with missing values filling out the observation.

SCORE — eigenvectors. The _NAME_ variable contains the name of the corresponding principal component as constructed from the PREFIX= option. The number of observations with _TYPE_='SCORE' equals the number of principal components computed. The eigenvectors have unit length unless you specify the STD option, in which case the unit-length eigenvectors are divided by the square roots of the eigenvalues to produce scores with unit standard deviations.

To obtain the principal component scores, if the COV option is not specified, these coefficients should be multiplied by the standardized data. With the COV option, these coefficients should be multiplied by the centered data. Means obtained from the observation with _TYPE_='MEAN' and standard deviations obtained from the observation with _TYPE_='STD' should be used for centering and standardizing the data.

USCORE	scoring coefficients to be applied without subtracting the mean from the raw variables. _TYPE_='USCORE' observations are produced when you specify the NOINT option in the PROC PRINCOMP statement. To obtain the principal component scores, these coefficients should be multiplied by the data that are standardized by the uncorrected standard deviations obtained from the observation with _TYPE_='USTD'.
RSQUARED	R squares for each VAR variable as predicted by the PARTIAL variables
B	regression coefficients for each VAR variable as predicted by the PARTIAL variables. This observation is produced only if you specify the COV option.
STB	standardized regression coefficients for each VAR variable as predicted by the PARTIAL variables. If you specify the COV option, this observation is omitted.

The data set can be used with the SCORE procedure to compute principal component scores, or it can be used as input to the FACTOR procedure specifying METHOD=SCORE to rotate the components. If you use the PARTIAL statement, the scoring coefficients should be applied to the residuals, not the original variables.

Computational Resources

Let

$$
\begin{aligned}
n &= \text{number of observations} \\
v &= \text{number of VAR variables} \\
p &= \text{number of PARTIAL variables} \\
c &= \text{number of components}
\end{aligned}
$$

- The minimum allocated memory required (in bytes) is

$$232v + 120p + 48c + \max(8cv, 8vp + 4(v + p)(v + p + 1))$$

- The time required to compute the correlation matrix is roughly proportional to

$$n(v + p)^2 + \frac{p}{2}(v + p)(v + p + 1)$$

- The time required to compute eigenvalues is roughly proportional to v^3.
- The time required to compute eigenvectors is roughly proportional to cv^2.

Displayed Output

The PRINCOMP procedure displays the following items if the DATA= data set is not TYPE=CORR, TYPE=COV, TYPE=SSCP, TYPE=UCORR, or TYPE=UCOV:

- simple statistics, including the mean and std (standard deviation) for each variable. If you specify the NOINT option, the uncorrected standard deviation (ustd) is displayed.
- the correlation or, if you specify the COV option, the covariance matrix

The PRINCOMP procedure displays the following items if you use the PARTIAL statement:

- regression statistics, giving the R square and RMSE (root mean squared error) for each VAR variable as predicted by the PARTIAL variables (not shown)
- standardized regression coefficients or, if you specify the COV option, regression coefficients for predicting the VAR variables from the PARTIAL variables (not shown)
- the partial correlation matrix or, if you specify the COV option, the partial covariance matrix (not shown)

The PRINCOMP procedure displays the following item if you specify the COV option:

- the total variance

The PRINCOMP procedure displays the following items unless you specify the NOPRINT option:

- eigenvalues of the correlation or covariance matrix, as well as the difference between successive eigenvalues, the proportion of variance explained by each eigenvalue, and the cumulative proportion of variance explained
- the eigenvectors

ODS Table Names

PROC PRINCOMP assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 69.2. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

All of the tables are created with the specification of the PROC PRINCOMP statement; a few tables need an additional PARTIAL statement.

Table 69.2 ODS Tables Produced by PROC PRINCOMP

ODS Table Name	Description	Statement / Option
Corr	Correlation matrix	default
Cov	Covariance matrix	COV
Eigenvalues	Eigenvalues	default
Eigenvectors	Eigenvectors	default
NObsNVar	Number of observations, variables, and partial variables	default
ParCorr	Partial correlation matrix	PARTIAL statement
ParCov	Uncorrected partial covariance matrix	PARTIAL statement and COV
RegCoef	Regression coefficients	PARTIAL statement and COV
RSquareRMSE	Regression statistics: R squares and RMSEs	PARTIAL statement
SimpleStatistics	Simple statistics	default
StdRegCoef	Standardized regression coefficients	PARTIAL statement
TotalVariance	Total variance	COV

ODS Graphics

To request graphics with PROC PRINCOMP, you must first enable ODS Graphics by specifying the **ods graphics on** statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. Some graphs are produced by default; other graphs are produced by using statements and options. You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC PRINCOMP generates are listed in Table 69.3, along with the required statements and options.

Table 69.3 ODS Graphics Produced by PROC PRINCOMP

ODS Graph Name	Plot Description	Statement and Option
PaintedScorePlot	Score plot of component 3 versus component 2, painted by component 1	PLOTS=SCORE when number of variables ≥ 3
PatternPlot	Component pattern plot	PLOTS=PATTERN
PatternProfilePlot	Component pattern profile plot	PLOTS=PATTERNPROFILE
ScoreMatrixPlot	Matrix plot of component scores	PLOTS=MATRIX
ScorePlot	Component score plot	PLOTS=SCORE
ScreePlot	Scree and variance plots	default and PLOTS=SCREE
VariancePlot	Variance proportion explained plot	PLOTS=SCREE(UNPACKPANEL)

Examples: PRINCOMP Procedure

Example 69.1: Temperatures

This example analyzes mean daily temperatures in selected cities in January and July. Both the raw data and the principal components are plotted to illustrate how principal components are orthogonal rotations of the original variables.

The following statements create the Temperature data set.

```
data Temperature;
   length Cityid $ 2;
   title 'Mean Temperature in January and July for Selected Cities ';
   input City $1-15 January July;
   Cityid = substr(City,1,2);
   datalines;
Mobile          51.2 81.6
Phoenix         51.2 91.2
Little Rock     39.5 81.4
Sacramento      45.1 75.2
Denver          29.9 73.0
Hartford        24.8 72.7
Wilmington      32.0 75.8
Washington DC   35.6 78.7
Jacksonville    54.6 81.0
Miami           67.2 82.3
Atlanta         42.4 78.0
Boise           29.0 74.5
Chicago         22.9 71.9
Peoria          23.8 75.1
Indianapolis    27.9 75.0
Des Moines      19.4 75.1
Wichita         31.3 80.7
Louisville      33.3 76.9
New Orleans     52.9 81.9
Portland, ME    21.5 68.0
Baltimore       33.4 76.6
Boston          29.2 73.3
Detroit         25.5 73.3
Sault Ste Marie 14.2 63.8
Duluth           8.5 65.6
Minneapolis     12.2 71.9
Jackson         47.1 81.7
Kansas City     27.8 78.8
St Louis        31.3 78.6
Great Falls     20.5 69.3
Omaha           22.6 77.2
Reno            31.9 69.3
Concord         20.6 69.7
```

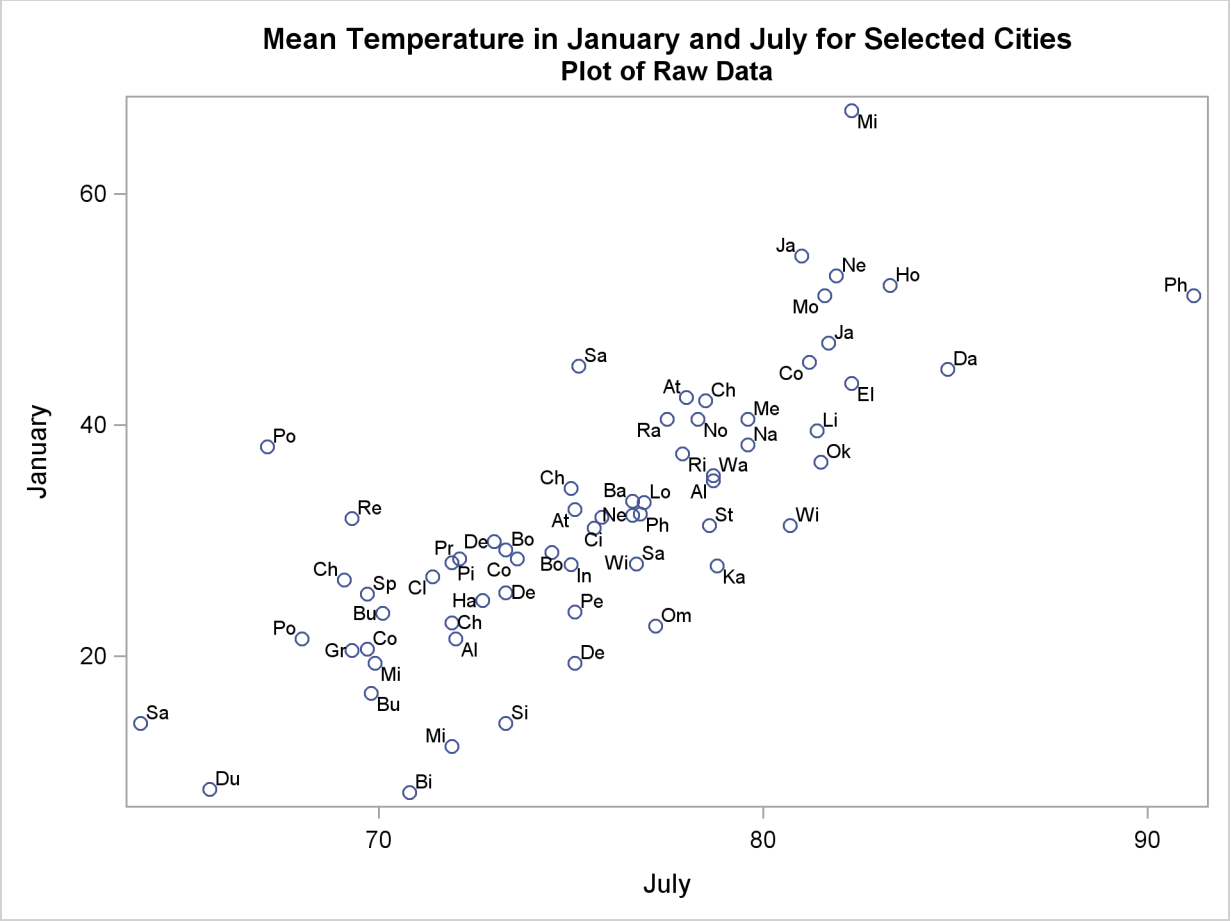
```
Atlantic City   32.7 75.1
Albuquerque     35.2 78.7
Albany          21.5 72.0
Buffalo         23.7 70.1
New York        32.2 76.6
Charlotte       42.1 78.5
Raleigh         40.5 77.5
Bismarck         8.2 70.8
Cincinnati      31.1 75.6
Cleveland       26.9 71.4
Columbus        28.4 73.6
Oklahoma City   36.8 81.5
Portland, OR    38.1 67.1
Philadelphia    32.3 76.8
Pittsburgh      28.1 71.9
Providence      28.4 72.1
Columbia        45.4 81.2
Sioux Falls     14.2 73.3
Memphis         40.5 79.6
Nashville       38.3 79.6
Dallas          44.8 84.8
El Paso         43.6 82.3
Houston         52.1 83.3
Salt Lake City  28.0 76.7
Burlington      16.8 69.8
Norfolk         40.5 78.3
Richmond        37.5 77.9
Spokane         25.4 69.7
Charleston, WV  34.5 75.0
Milwaukee       19.4 69.9
Cheyenne        26.6 69.1
;
```

The following statements plot the temperature data set. The Cityid variable instead of City is used as a data label in the scatter plot for possible label clashing.

```
title 'Mean Temperature in January and July for Selected Cities';
title2 'Plot of Raw Data';
proc sgplot data=Temperature;
   scatter x=July y=January / datalabel=Cityid;
run;
```

The results are displayed in Output 69.1.1, which shows a scatter diagram of the 64 pairs of data points with July temperatures plotted against January temperatures.

Output 69.1.1 Plot of Raw Data



The following statement requests a principal component analysis on the Temperature data set:

```
ods graphics on;
title 'Mean Temperature in January and July for Selected Cities';
proc princomp data=Temperature cov plots=score(ellipse);
   var July January;
   id Cityid;
run;
ods graphics off;
```

Output 69.1.2 displays the PROC PRINCOMP output. The standard deviation of January (11.712) is higher than the standard deviation of July (5.128). The COV option in the PROC PRINCOMP statement requests the principal components to be computed from the covariance matrix. The total variance is 163.474. The first principal component explains about 94% of the total variance, and the second principal component explains only about 6%. The eigenvalues sum to the total variance.

Note that January receives a higher loading on Prin1 because it has a higher standard deviation than July, and the PRINCOMP procedure calculates the scores by using the centered variables rather than the standardized variables.

Output 69.1.2 Results of Principal Component Analysis

Mean Temperature in January and July for Selected Cities

The PRINCOMP Procedure

Observations	64
Variables	2

Simple Statistics

	July	January
Mean	75.60781250	32.09531250
StD	5.12761910	11.71243309

Covariance Matrix

	July	January
July	26.2924777	46.8282912
January	46.8282912	137.1810888

Total Variance 163.47356647

Eigenvalues of the Covariance Matrix

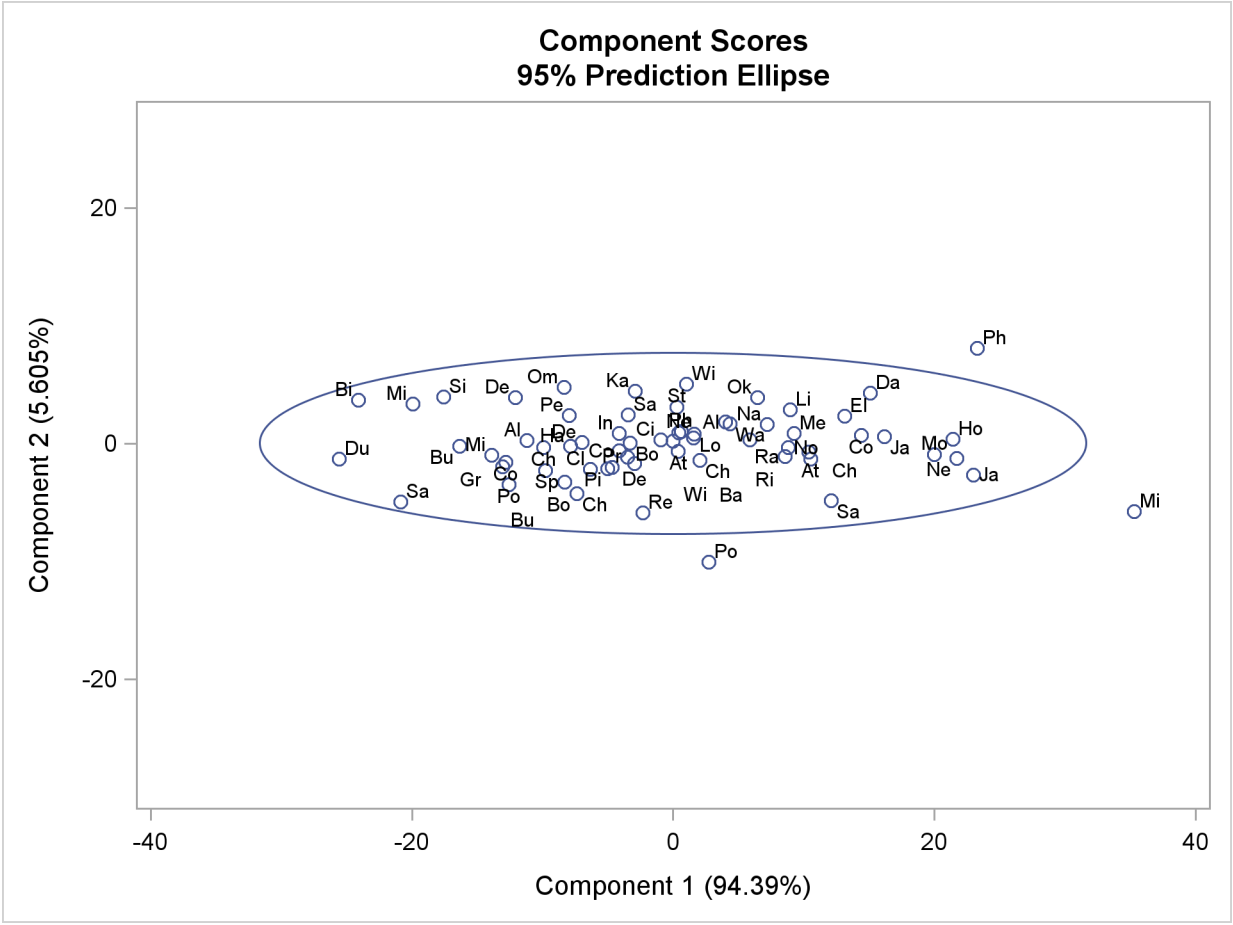
	Eigenvalue	Difference	Proportion	Cumulative
1	154.310607	145.147647	0.9439	0.9439
2	9.162960		0.0561	1.0000

Eigenvectors

	Prin1	Prin2
July	0.343532	0.939141
January	0.939141	-.343532

PLOTS=SCORE in the PROC PRINCOMP statement requests a plot of the second principal component against the first principal component as shown in Output 69.1.3. It is clear from this plot that the principal components are orthogonal rotations of the original variables and that the first principal component has a larger variance than the second principal component. In fact, the first component has a larger variance than either of the original variables July and January. The ellipse indicates that Miami, Phoenix, and Portland are possible outliers.

Output 69.1.3 Plot of Component 2 by Component 1



Example 69.2: Basketball Data

The data in this example are rankings of 35 college basketball teams. The rankings were made before the start of the 1985–86 season by 10 news services.

The purpose of the principal component analysis is to compute a single variable that best summarizes all 10 of the preseason rankings.

Note that the various news services rank different numbers of teams, varying from 20 through 30 (there is a missing rank in one of the variables, WashPost). And, of course, not all services rank the same teams, so there are missing values in these data. Each of the 35 teams is ranked by at least one news service.

The PRINCOMP procedure omits observations with missing values. To obtain principal component scores for all of the teams, it is necessary to replace the missing values. Since it is the best teams that are ranked, it is not appropriate to replace missing values with the mean of the nonmissing values. Instead, an ad hoc method is used that replaces missing values with the mean of the unassigned ranks. For example, if 20 teams are ranked by a news service, then ranks 21 through 35 are unassigned. The mean of ranks 21 through 35 is 28, so missing values for that variable are replaced

by the value 28. To prevent the method of missing-value replacement from having an undue effect on the analysis, each observation is weighted according to the number of nonmissing values it has. See Example 70.2 in Chapter 70, "The PRINQUAL Procedure," for an alternative analysis of these data.

Since the first principal component accounts for 78% of the variance, there is substantial agreement among the rankings. The eigenvector shows that all the news services are about equally weighted; this is also suggested by the nearly horizontal line of the pattern profile plot in Output 69.2.3. So a simple average would work almost as well as the first principal component. The following statements produce Output 69.2.1.

```
/*-------------------------------------------------------------*/
/*                                                             */
/* Pre-season 1985 College Basketball Rankings                 */
/* (rankings of 35 teams by 10 news services)                  */
/*                                                             */
/* Note: (a) news services rank varying numbers of teams;      */
/*       (b) not all teams are ranked by all news services;    */
/*       (c) each team is ranked by at least one service;      */
/*       (d) rank 20 is missing for UPI.                       */
/*                                                             */
/*-------------------------------------------------------------*/

data HoopsRanks;
   input School $13. CSN DurSun DurHer WashPost USAToday
         Sport InSports UPI AP SI;
   label CSN      = 'Community Sports News (Chapel Hill, NC)'
         DurSun   = 'Durham Sun'
         DurHer   = 'Durham Morning Herald'
         WashPost = 'Washington Post'
         USAToday = 'USA Today'
         Sport    = 'Sport Magazine'
         InSports = 'Inside Sports'
         UPI      = 'United Press International'
         AP       = 'Associated Press'
         SI       = 'Sports Illustrated'
         ;
   format CSN--SI 5.1;
   datalines;
Louisville     1  8  1  9  8  9  6 10  9  9
Georgia Tech   2  2  4  3  1  1  1  2  1  1
Kansas         3  4  5  1  5 11  8  4  5  7
Michigan       4  5  9  4  2  5  3  1  3  2
Duke           5  6  7  5  4 10  4  5  6  5
UNC            6  1  2  2  3  4  2  3  2  3
Syracuse       7 10  6 11  6  6  5  6  4 10
Notre Dame     8 14 15 13 11 20 18 13 12  .
Kentucky       9 15 16 14 14 19 11 12 11 13
LSU           10  9 13  . 13 15 16  9 14  8
DePaul        11  . 21 15 20  . 19  .  . 19
Georgetown    12  7  8  6  9  2  9  8  8  4
Navy          13 20 23 10 18 13 15  . 20  .
Illinois      14  3  3  7  7  3 10  7  7  6
```

```
Iowa          15 16  .  . 23  .  . 14  . 20
Arkansas      16  .  .  . 25  .  .  .  . 16
Memphis State 17  . 11  . 16  8 20  . 15 12
Washington    18  .  .  .  .  .  . 17  .  .
UAB           19 13 10  . 12 17  . 16 16 15
UNLV          20 18 18 19 22  . 14 18 18  .
NC State      21 17 14 16 15  . 12 15 17 18
Maryland      22  .  .  . 19  .  .  . 19 14
Pittsburgh    23  .  .  .  .  .  .  .  .  .
Oklahoma      24 19 17 17 17 12 17  . 13 17
Indiana       25 12 20 18 21  .  .  .  .  .
Virginia      26  . 22  .  . 18  .  .  .  .
Old Dominion  27  .  .  .  .  .  .  .  .  .
Auburn        28 11 12  8 10  7  7 11 10 11
St. Johns     29  .  .  .  . 14  .  .  .  .
UCLA          30  .  .  .  .  .  . 19  .  .
St. Joseph's   .  . 19  .  .  .  .  .  .  .
Tennessee      .  . 24  .  . 16  .  .  .  .
Montana        .  .  . 20  .  .  .  .  .  .
Houston        .  .  .  . 24  .  .  .  .  .
Virginia Tech  .  .  .  .  .  . 13  .  .  .
;

/* PROC MEANS is used to output a data set containing the      */
/* maximum value of each of the newspaper and magazine         */
/* rankings.  The output data set, maxrank, is then used       */
/* to set the missing values to the next highest rank plus     */
/* thirty-six, divided by two (that is, the mean of the        */
/* missing ranks).  This ad hoc method of replacing missing    */
/* values is based more on intuition than on rigorous          */
/* statistical theory.  Observations are weighted by the       */
/* number of nonmissing values.                                */
/*                                                             */

title 'Pre-Season 1985 College Basketball Rankings';
proc means data=HoopsRanks;
   output out=MaxRank
          max=CSNMax DurSunMax DurHerMax
              WashPostMax USATodayMax SportMax
              InSportsMax UPIMax APMax SIMax;
run;
```

Output 69.2.1 Summary Statistics for Basketball Rankings Using PROC MEANS

Pre-Season 1985 College Basketball Rankings

The MEANS Procedure

Variable	Label	N	Mean
CSN	Community Sports News (Chapel Hill, NC)	30	15.5000000
DurSun	Durham Sun	20	10.5000000
DurHer	Durham Morning Herald	24	12.5000000
WashPost	Washington Post	19	10.4210526
USAToday	USA Today	25	13.0000000
Sport	Sport Magazine	20	10.5000000
InSports	Inside Sports	20	10.5000000
UPI	United Press International	19	10.0000000
AP	Associated Press	20	10.5000000
SI	Sports Illustrated	20	10.5000000

Variable	Label	Std Dev	Minimum
CSN	Community Sports News (Chapel Hill, NC)	8.8034084	1.0000000
DurSun	Durham Sun	5.9160798	1.0000000
DurHer	Durham Morning Herald	7.0710678	1.0000000
WashPost	Washington Post	6.0673607	1.0000000
USAToday	USA Today	7.3598007	1.0000000
Sport	Sport Magazine	5.9160798	1.0000000
InSports	Inside Sports	5.9160798	1.0000000
UPI	United Press International	5.6273143	1.0000000
AP	Associated Press	5.9160798	1.0000000
SI	Sports Illustrated	5.9160798	1.0000000

Variable	Label	Maximum
CSN	Community Sports News (Chapel Hill, NC)	30.0000000
DurSun	Durham Sun	20.0000000
DurHer	Durham Morning Herald	24.0000000
WashPost	Washington Post	20.0000000
USAToday	USA Today	25.0000000
Sport	Sport Magazine	20.0000000
InSports	Inside Sports	20.0000000
UPI	United Press International	19.0000000
AP	Associated Press	20.0000000
SI	Sports Illustrated	20.0000000

The following statements produce Output 69.2.2 and Output 69.2.3:

```
   data Basketball;
      set HoopsRanks;
      if _n_=1 then set MaxRank;
      array Services{10} CSN--SI;
      array MaxRanks{10} CSNMax--SIMax;
      keep School CSN--SI Weight;
      Weight=0;
      do i=1 to 10;
         if Services{i}=. then Services{i}=(MaxRanks{i}+36)/2;
         else Weight=Weight+1;
      end;
   run;

ods graphics on;
proc princomp data=Basketball n=1 out=PCBasketball standard
              plots=patternprofile;
   var CSN--SI;
   weight Weight;
run;
ods graphics off;
```

Output 69.2.2 Principal Components Analysis of Basketball Rankings Using PROC PRINCOMP

Pre-Season 1985 College Basketball Rankings

The PRINCOMP Procedure

Observations	35
Variables	10

Simple Statistics

	CSN	DurSun	DurHer	WashPost	USAToday
Mean	13.33640553	13.06451613	12.88018433	13.83410138	12.55760369
StD	22.08036285	21.66394183	21.38091837	23.47841791	20.48207965

Simple Statistics

	Sport	InSports	UPI	AP	SI
Mean	13.83870968	13.24423963	13.59216590	12.83410138	13.52534562
StD	23.37756267	22.20231526	23.25602811	21.40782406	22.93219584

Output 69.2.2 *continued*

Correlation Matrix

		CSN	DurSun	DurHer
CSN	Community Sports News (Chapel Hill, NC)	1.0000	0.6505	0.6415
DurSun	Durham Sun	0.6505	1.0000	0.8341
DurHer	Durham Morning Herald	0.6415	0.8341	1.0000
WashPost	Washington Post	0.6121	0.7667	0.7035
USAToday	USA Today	0.7456	0.8860	0.8877
Sport	Sport Magazine	0.4806	0.6940	0.7788
InSports	Inside Sports	0.6558	0.7702	0.7900
UPI	United Press International	0.7007	0.9015	0.7676
AP	Associated Press	0.6779	0.8437	0.8788
SI	Sports Illustrated	0.6135	0.7518	0.7761

Correlation Matrix

	Wash Post	USAToday	Sport	In Sports	UPI	AP	SI
CSN	0.6121	0.7456	0.4806	0.6558	0.7007	0.6779	0.6135
DurSun	0.7667	0.8860	0.6940	0.7702	0.9015	0.8437	0.7518
DurHer	0.7035	0.8877	0.7788	0.7900	0.7676	0.8788	0.7761
WashPost	1.0000	0.7984	0.6598	0.8717	0.6953	0.7809	0.5952
USAToday	0.7984	1.0000	0.7716	0.8475	0.8539	0.9479	0.8426
Sport	0.6598	0.7716	1.0000	0.7176	0.6220	0.8217	0.7701
InSports	0.8717	0.8475	0.7176	1.0000	0.7920	0.8830	0.7332
UPI	0.6953	0.8539	0.6220	0.7920	1.0000	0.8436	0.7738
AP	0.7809	0.9479	0.8217	0.8830	0.8436	1.0000	0.8212
SI	0.5952	0.8426	0.7701	0.7332	0.7738	0.8212	1.0000

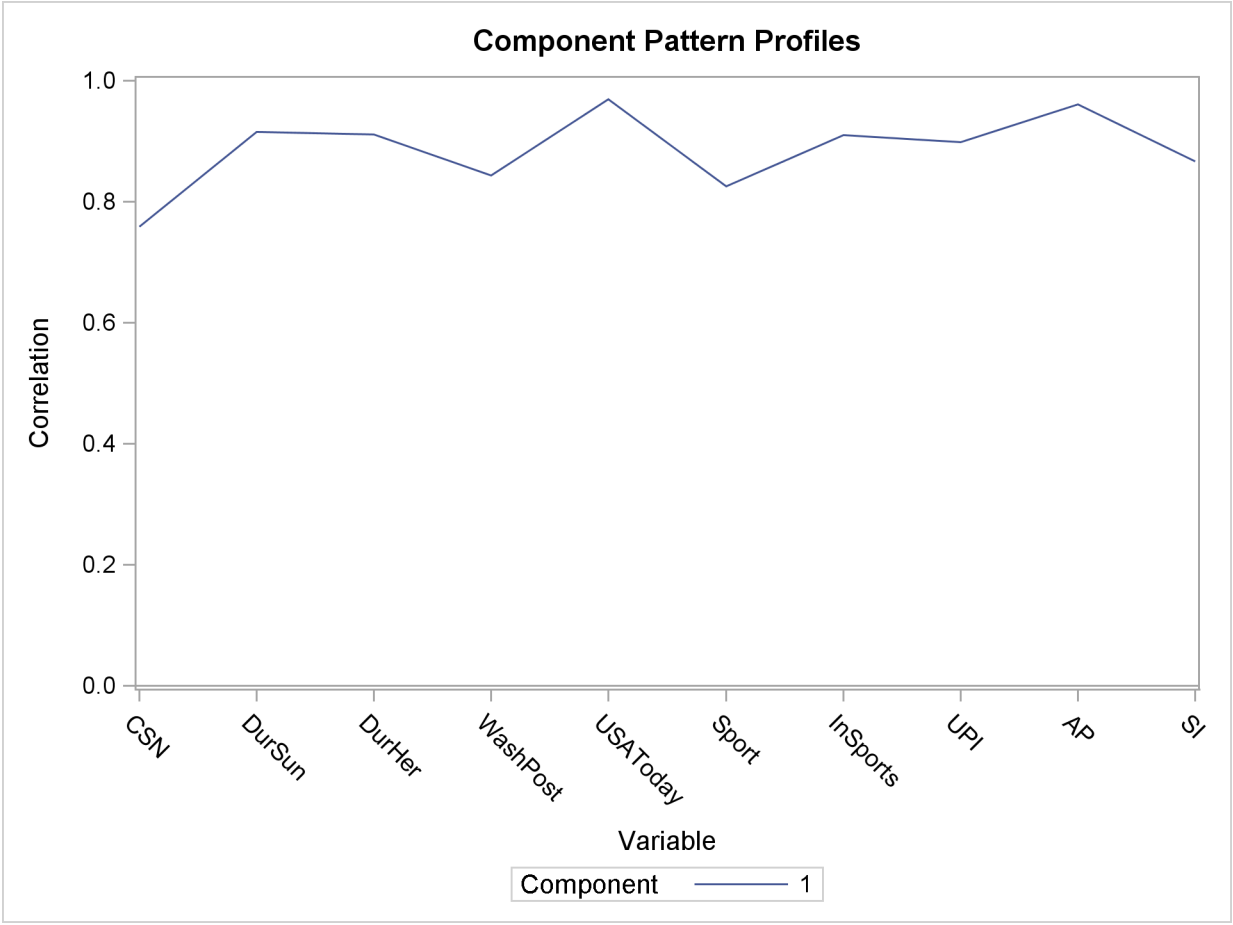
Eigenvalues of the Correlation Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	7.88601647		0.7886	0.7886

Eigenvectors

		Prin1
CSN	Community Sports News (Chapel Hill, NC)	0.270205
DurSun	Durham Sun	0.326048
DurHer	Durham Morning Herald	0.324392
WashPost	Washington Post	0.300449
USAToday	USA Today	0.345200
Sport	Sport Magazine	0.293881
InSports	Inside Sports	0.324088
UPI	United Press International	0.319902
AP	Associated Press	0.342151
SI	Sports Illustrated	0.308570

Output 69.2.3 Pattern Profile Plot



The following statements produce Output 69.2.4:

```
proc sort data=PCBasketball;
   by Prin1;
run;

proc print;
   var School Prin1;
   title 'Pre-Season 1985 College Basketball Rankings';
   title2 'College Teams as Ordered by PROC PRINCOMP';
run;
```

Output 69.2.4 Basketball Rankings Using PROC PRINCOMP

Pre-Season 1985 College Basketball Rankings
College Teams as Ordered by PROC PRINCOMP

OBS	School	Prin1
1	Georgia Tech	-0.58068
2	UNC	-0.53317
3	Michigan	-0.47874
4	Kansas	-0.40285
5	Duke	-0.38464
6	Illinois	-0.33586
7	Syracuse	-0.31578
8	Louisville	-0.31489
9	Georgetown	-0.29735
10	Auburn	-0.09785
11	Kentucky	0.00843
12	LSU	0.00872
13	Notre Dame	0.09407
14	NC State	0.19404
15	UAB	0.19771
16	Oklahoma	0.23864
17	Memphis State	0.25319
18	Navy	0.28921
19	UNLV	0.35103
20	DePaul	0.43770
21	Iowa	0.50213
22	Indiana	0.51713
23	Maryland	0.55910
24	Arkansas	0.62977
25	Virginia	0.67586
26	Washington	0.67756
27	Tennessee	0.70822
28	St. Johns	0.71425
29	Virginia Tech	0.71638
30	St. Joseph's	0.73492
31	UCLA	0.73965
32	Pittsburgh	0.75078
33	Houston	0.75534
34	Montana	0.75790
35	Old Dominion	0.76821

Example 69.3: Job Ratings

This example uses the PRINCOMP procedure to analyze job performance. Police officers were rated by their supervisors in 14 categories as part of standard police departmental administrative procedure.

The following statements create the Jobratings data set:

```
options validvarname=any;
data Jobratings;
   input ('Communication Skills'n
          'Problem Solving'n
          'Learning Ability'n
          'Judgment Under Pressure'n
          'Observational Skills'n
          'Willingness to Confront Problems'n
          'Interest in People'n
          'Interpersonal Sensitivity'n
          'Desire for Self-Improvement'n
          'Appearance'n
          'Dependability'n
          'Physical Ability'n
          'Integrity'n
          'Overall Rating'n) (1.);
   datalines;
26838853879867
74758876857667
56757863775875
67869777988997
99997798878888
89897899888799
89999889899798
87794798468886
35652335143113
89888879576867
76557899446397
97889998898989
76766677598888

... more lines ...

99899899899899
76656399567486
;
```

The data set Jobratings contains 14 variables. Each variable contains the job ratings, using a scale measurement from 1 to 10 (1=fail to comply, 10=exceptional). The last variable Overall Rating contains a score as an overall index on how each officer performs.

The following statements request a principal component analysis on the Jobratings data set, output the scores to the Scores data set (OUT= Scores), and produce default plots. Note that variable Overall Rating is excluded from the analysis.

```
ods graphics on;
proc princomp data=Jobratings(drop='Overall Rating'n);
run;
```

Figure 69.3.1 and Figure 69.3.2 display the PROC PRINCOMP output, beginning with simple statistics followed by the correlation matrix. By default, the PROC PRINCOMP statement requests

principal components computed from the correlation matrix, so the total variance is equal to the number of variables, 13. In this example, it would also be reasonable to use the COV option, which would cause variables with a high variance (such as Dependability) to have more influence on the results than variables with a low variance (such as Learning Ability). If you used the COV option, scores would be computed from centered rather than standardized variables.

Output 69.3.1 Simple Statistics and Correlation Matrix from the PRINCOMP Procedure

Pre-Season 1985 College Basketball Rankings
College Teams as Ordered by PROC PRINCOMP

The PRINCOMP Procedure

Observations 103
Variables 13

Simple Statistics

	Communication Skills	Problem Solving	Learning Ability	Judgment Under Pressure	Observational Skills
Mean	6.650485437	6.631067961	6.990291262	6.737864078	6.932038835
StD	1.764068036	1.590352602	1.339411238	1.731830976	1.761584269

Simple Statistics

	Willingness to Confront Problems	Interest in People	Interpersonal Sensitivity	Desire for Self-Improvement	Appearance
Mean	7.291262136	6.708737864	6.621359223	6.572815534	7.000000000
StD	1.525155524	1.892353385	1.760773587	1.729796212	1.798692335

Simple Statistics

	Dependability	Physical Ability	Integrity
Mean	6.825242718	7.203883495	7.213592233
StD	1.917040123	1.555251845	1.845240223

Output 69.3.1 *continued*

Correlation Matrix

	Communication Skills	Problem Solving	Learning Ability	Judgment Under Pressure
Communication Skills	1.0000	0.6280	0.5546	0.5538
Problem Solving	0.6280	1.0000	0.5690	0.6195
Learning Ability	0.5546	0.5690	1.0000	0.4892
Judgment Under Pressure	0.5538	0.6195	0.4892	1.0000
Observational Skills	0.5381	0.4284	0.6230	0.3733
Willingness to Confront Problems	0.5265	0.5015	0.5245	0.4004
Interest in People	0.4391	0.3972	0.2735	0.6226
Interpersonal Sensitivity	0.5030	0.4398	0.1855	0.6134
Desire for Self-Improvement	0.5642	0.4090	0.5737	0.4826
Appearance	0.4913	0.3873	0.3988	0.2266
Dependability	0.5471	0.4546	0.5110	0.5471
Physical Ability	0.2192	0.3201	0.2269	0.3476
Integrity	0.5081	0.3846	0.3142	0.5883

Correlation Matrix

	Observational Skills	Willingness to Confront Problems	Interest in People
Communication Skills	0.5381	0.5265	0.4391
Problem Solving	0.4284	0.5015	0.3972
Learning Ability	0.6230	0.5245	0.2735
Judgment Under Pressure	0.3733	0.4004	0.6226
Observational Skills	1.0000	0.7300	0.2616
Willingness to Confront Problems	0.7300	1.0000	0.2233
Interest in People	0.2616	0.2233	1.0000
Interpersonal Sensitivity	0.1655	0.1291	0.8051
Desire for Self-Improvement	0.5985	0.5307	0.4857
Appearance	0.4177	0.4825	0.2679
Dependability	0.5626	0.4870	0.6074
Physical Ability	0.4274	0.4872	0.3768
Integrity	0.3906	0.3260	0.7452

Correlation Matrix

	Interpersonal Sensitivity	Desire for Self-Improvement	Appearance
Communication Skills	0.5030	0.5642	0.4913
Problem Solving	0.4398	0.4090	0.3873
Learning Ability	0.1855	0.5737	0.3988
Judgment Under Pressure	0.6134	0.4826	0.2266

Output 69.3.1 *continued*

Correlation Matrix

	Interpersonal Sensitivity	Desire for Self-Improvement	Appearance
Observational Skills	0.1655	0.5985	0.4177
Willingness to Confront Problems	0.1291	0.5307	0.4825
Interest in People	0.8051	0.4857	0.2679
Interpersonal Sensitivity	1.0000	0.3713	0.2600
Desire for Self-Improvement	0.3713	1.0000	0.4474
Appearance	0.2600	0.4474	1.0000
Dependability	0.5408	0.5981	0.5089
Physical Ability	0.2182	0.3752	0.3820
Integrity	0.6920	0.5664	0.4135

Correlation Matrix

	Dependability	Physical Ability	Integrity
Communication Skills	0.5471	0.2192	0.5081
Problem Solving	0.4546	0.3201	0.3846
Learning Ability	0.5110	0.2269	0.3142
Judgment Under Pressure	0.5471	0.3476	0.5883
Observational Skills	0.5626	0.4274	0.3906
Willingness to Confront Problems	0.4870	0.4872	0.3260
Interest in People	0.6074	0.3768	0.7452
Interpersonal Sensitivity	0.5408	0.2182	0.6920
Desire for Self-Improvement	0.5981	0.3752	0.5664
Appearance	0.5089	0.3820	0.4135
Dependability	1.0000	0.4461	0.6536
Physical Ability	0.4461	1.0000	0.3810
Integrity	0.6536	0.3810	1.0000

Figure 69.3.2 displays the eigenvalues. The first principal component explains about 50% of the total variance, the second principal component explains about 13.6%, and the third principal component explains about 7.7%. Note that the eigenvalues sum to the total variance. The eigenvalues indicate that three to five components provide a good summary of the data, with three components accounting for about 71.7% of the total variance and five components explaining about 82.7%. Subsequent components contribute less than 5% each.

Output 69.3.2 Eigenvalues and Eigenvectors from the PRINCOMP Procedure

Eigenvalues of the Correlation Matrix

	Eigenvalue	Difference	Proportion	Cumulative
1	6.54740242	4.77468744	0.5036	0.5036
2	1.77271499	0.76747933	0.1364	0.6400
3	1.00523565	0.26209665	0.0773	0.7173
4	0.74313901	0.06479499	0.0572	0.7745
5	0.67834402	0.22696368	0.0522	0.8267
6	0.45138034	0.06922167	0.0347	0.8614
7	0.38215866	0.08432613	0.0294	0.8908
8	0.29783254	0.02340663	0.0229	0.9137
9	0.27442591	0.01208809	0.0211	0.9348
10	0.26233782	0.01778332	0.0202	0.9550
11	0.24455450	0.04677622	0.0188	0.9738
12	0.19777828	0.05508241	0.0152	0.9890
13	0.14269586		0.0110	1.0000

Output 69.3.2 *continued*

Eigenvectors

	Prin1	Prin2	Prin3	Prin4
Communication Skills	0.303548	0.052039	-.329181	-.227039
Problem Solving	0.278034	0.057046	-.400112	0.300476
Learning Ability	0.266521	0.288152	-.354591	-.020735
Judgment Under Pressure	0.294376	-.199458	-.255164	0.397306
Observational Skills	0.276641	0.366979	0.065959	0.035711
Willingness to Confront Problems	0.267580	0.392989	0.098723	0.184409
Interest in People	0.278060	-.432916	0.118113	0.046047
Interpersonal Sensitivity	0.253814	-.495662	-.064547	-.060000
Desire for Self-Improvement	0.299833	0.099077	0.061097	-.211279
Appearance	0.237358	0.190065	0.248353	-.544587
Dependability	0.319480	-.049742	0.169476	-.156070
Physical Ability	0.213868	0.097499	0.614959	0.514519
Integrity	0.298246	-.301812	0.190222	-.169062

Eigenvectors

	Prin5	Prin6	Prin7	Prin8
Communication Skills	0.181087	-.416563	0.143543	0.333846
Problem Solving	0.453604	0.096750	0.048904	0.199259
Learning Ability	-.219329	0.578388	-.114808	0.064088
Judgment Under Pressure	-.030188	0.102087	0.068204	-.591822
Observational Skills	-.325257	-.301254	-.297894	0.163484
Willingness to Confront Problems	0.038278	-.458585	-.044796	-.365684
Interest in People	-.111279	0.030870	-.011105	0.154829
Interpersonal Sensitivity	0.107807	-.170305	-.088194	0.192725
Desire for Self-Improvement	-.427477	0.105369	0.689011	0.087453
Appearance	0.568044	0.221643	0.049267	-.257497
Dependability	-.130575	0.202301	-.594850	0.081242
Physical Ability	0.203995	0.173168	0.169247	0.302536
Integrity	-.130757	-.100039	0.029456	-.317545

Eigenvectors

	Prin9	Prin10	Prin11	Prin12
Communication Skills	-.430955	0.375983	0.028370	-.252778
Problem Solving	0.256098	-.372914	-.434417	0.069863
Learning Ability	0.224706	0.287031	0.210540	-.284355
Judgment Under Pressure	-.358618	0.178270	0.118318	0.306490
Observational Skills	0.258377	0.223793	-.079692	0.565290
Willingness to Confront Problems	0.129976	-.330710	0.275249	-.386151
Interest in People	0.321200	-.081470	0.393841	-.210915
Interpersonal Sensitivity	0.137468	-.074821	0.285447	0.276824
Desire for Self-Improvement	-.121474	-.363854	-.052085	0.151436
Appearance	0.087395	0.061890	0.168369	0.236655

Output 69.3.2 *continued*

Eigenvectors

	Prin9	Prin10	Prin11	Prin12
Dependability	-.495598	-.377561	-.164909	-.090904
Physical Ability	-.149625	0.258321	-.006202	-.055828
Integrity	0.271060	0.297010	-.612497	-.276273

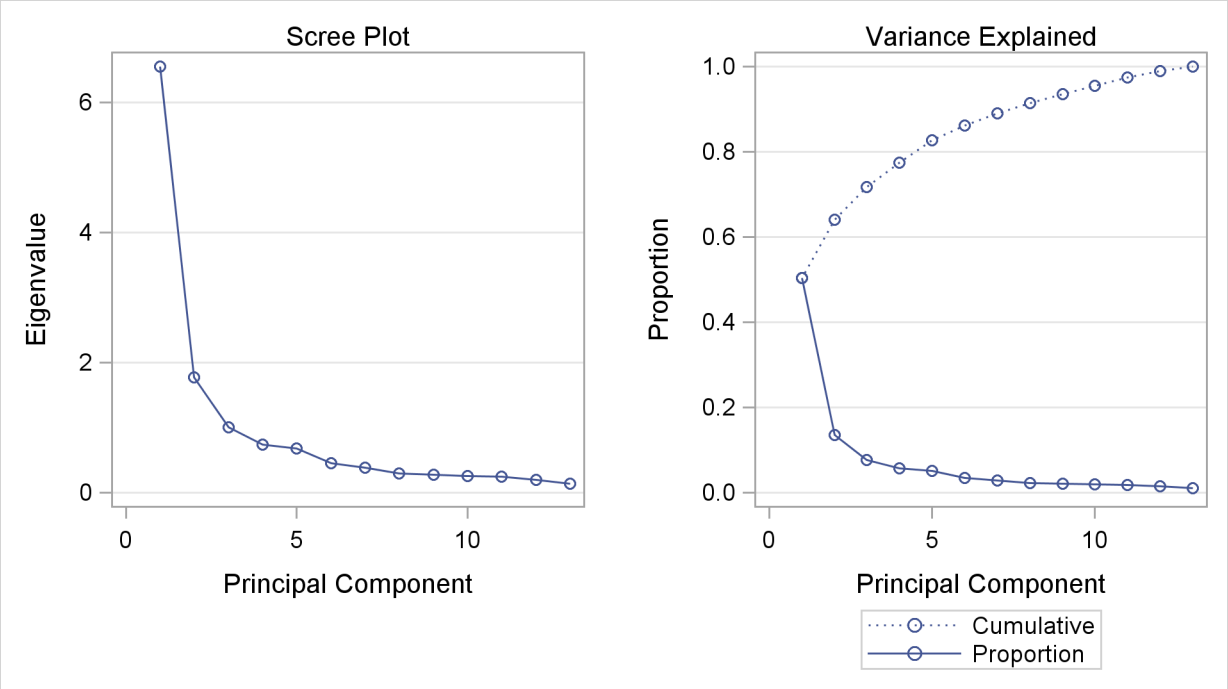
Eigenvectors

	Prin13
Communication Skills	-.122809
Problem Solving	-.116642
Learning Ability	0.248555
Judgment Under Pressure	-.126636
Observational Skills	-.168555
Willingness to Confront Problems	0.177688
Interest in People	-.610215
Interpersonal Sensitivity	0.643410
Desire for Self-Improvement	0.053834
Appearance	-.113705
Dependability	-.018094
Physical Ability	0.133430
Integrity	0.114965

When the **ods graphics on** statement is specified, PROC PRINCOMP produces the scree plot as shown in Figure 69.3.3 by default, which helps to visualize and choose the number of components. You can obtain more plots by specifying the PLOTS= option in the PROC PRINCOMP statement.

The "Scree Plot" on the left shows that the eigenvalue of the first component is approximately 6.5 and the eigenvalue of the second component is largely decreased to under 2.0. The "Variance Explained" plot on the right shows that you can explain a near 80% of total variance with the first four principal components.

Output 69.3.3 Scree Plot from the PRINCOMP Procedure



The first component reflects overall performance since the first eigenvector shows approximately equal loadings on all variables. The second eigenvector has high positive loadings on the variables Observational Skills and Willingness to Confront Problems but even higher negative loadings on the variables Interest in People and Interpersonal Sensitivity. This component seems to reflect the ability to take action, but it also reflects a lack of interpersonal skills. The third eigenvector has a very high positive loading on the variable Physical Ability and high negative loadings on the variables Problem Solving and Learning Ability. This component seems to reflect physical strength, but also shows poor learning and problem-solving skills.

In short, the three components represent the following:

First Component:	overall performance
Second Component:	smart, tough, and introverted
Third Component:	superior strength and average intellect

PROC PRINCOMP also produces other plots besides the scree plot, which are helpful while interpreting the results. The following statements request plots from the PRINCOMP procedure:

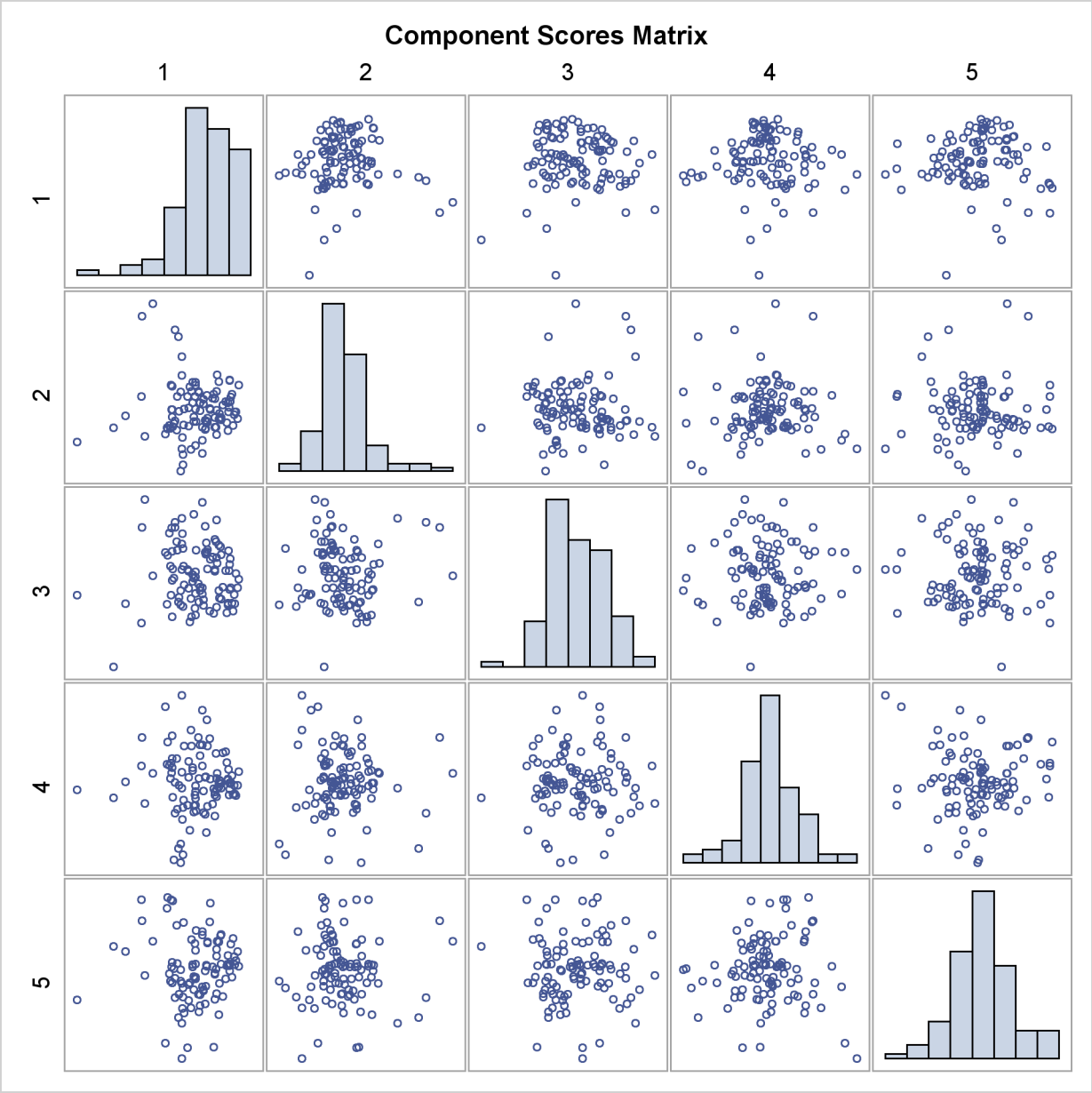
```
proc princomp data=Jobratings(drop='Overall Rating'n)
              plots(ncomp=3)=all n=5;
run;
ods graphics off;
```

PLOTS=ALL(NCOMP=3) in the PROC PRINCOMP statement requests all plots to be produced but limits the number of components to be plotted in the component pattern plots and the component score plots to three. The N=5 option sets the number of principal components to be computed to

five. Besides a scree plot similar to the one shown before, the rest of plots are displayed in the following context.

Output 69.3.4 shows a matrix plot of component scores between the first five principal components. The histogram of each component is displayed in the diagonal element of the matrix. The histograms indicate that the first principal component is skewed to the left and the second principal component is slightly skewed to the right.

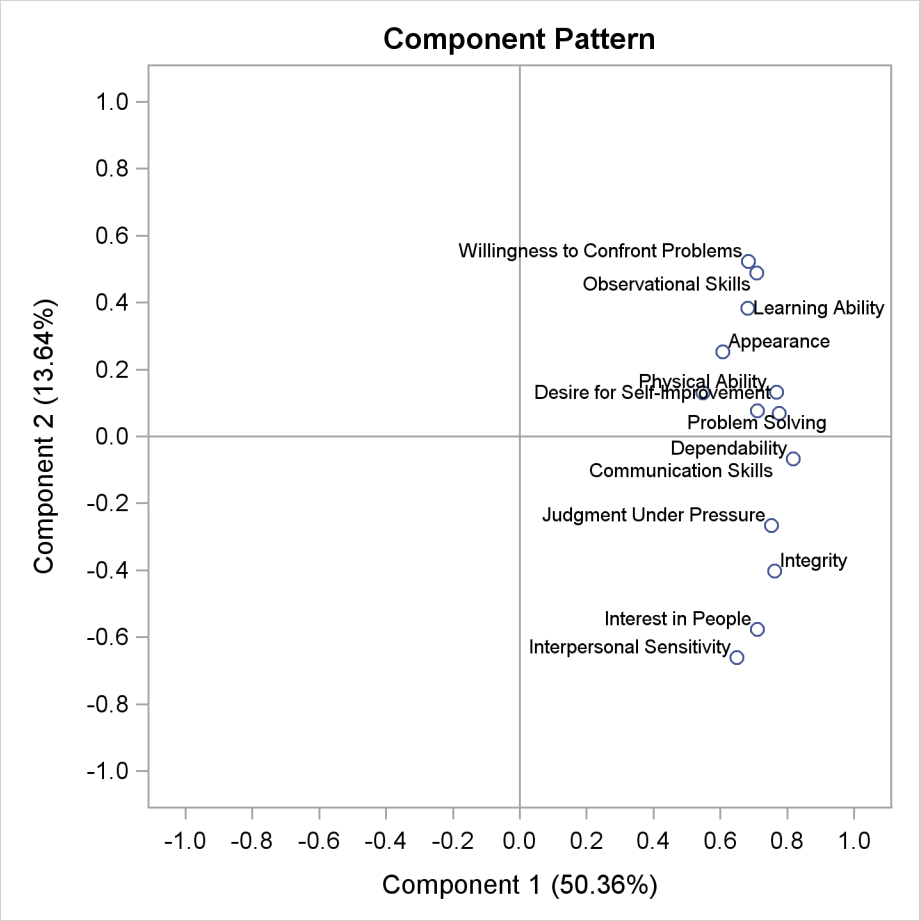
Output 69.3.4 Matrix Plot of Component Scores



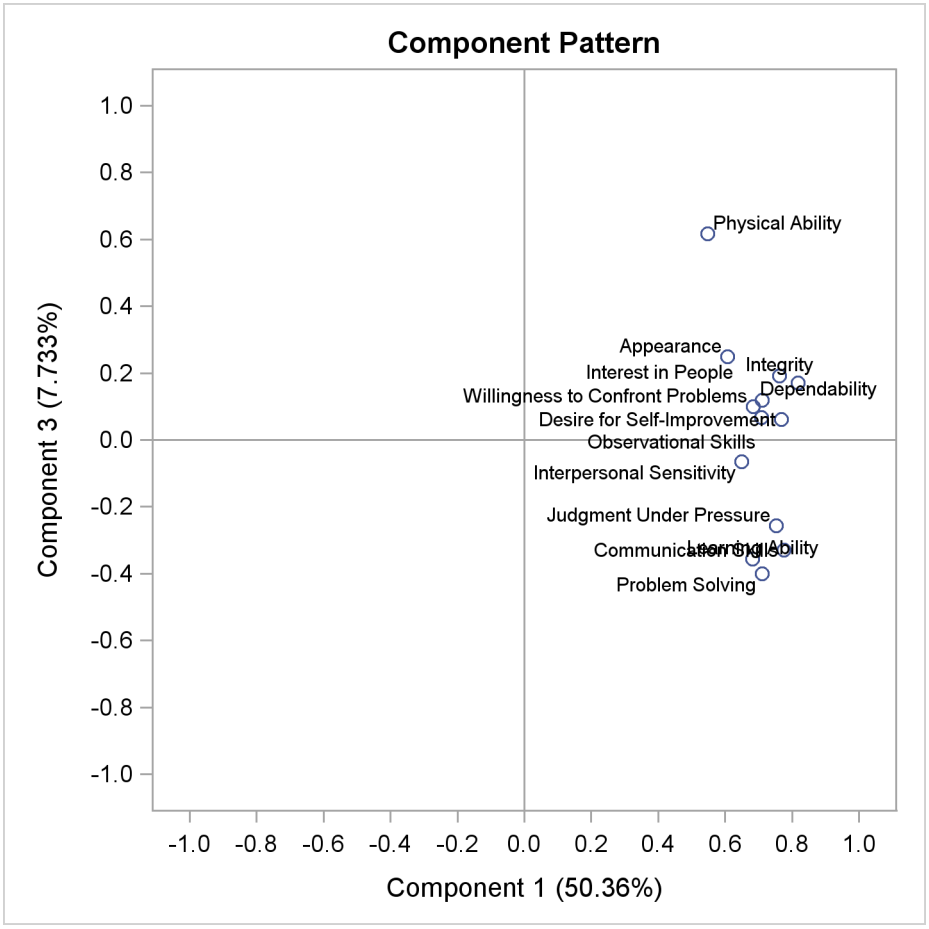
The pairwise component pattern plots are shown in Output 69.3.5 to Output 69.3.7. The pattern plots show the following:

- All variables positively and evenly correlate with the first principal component (Output 69.3.5 and Output 69.3.6).
- The variables Observational Skills and Willingness to Confront Problems correlate highly with the second component, and the variables Interest in People and Interpersonal Sensitivity correlate highly but negatively with the second component (Output 69.3.5).
- The variable Physical Ability correlates highly with the third component, and the variables Problem Solving and Learning Ability correlate highly but negatively with the third component (Output 69.3.6).
- The variable Observational Skills, Willingness to Confront Problems, Interest in People, and Interpersonal Sensitivity correlate highly (either positively or negatively) with the second component, but all have very low correlations with the third component; the variables Physical Ability and Problem Solving correlate highly (either positively or negatively) with the third component, but both have very low correlations with the second component (Output 69.3.7).

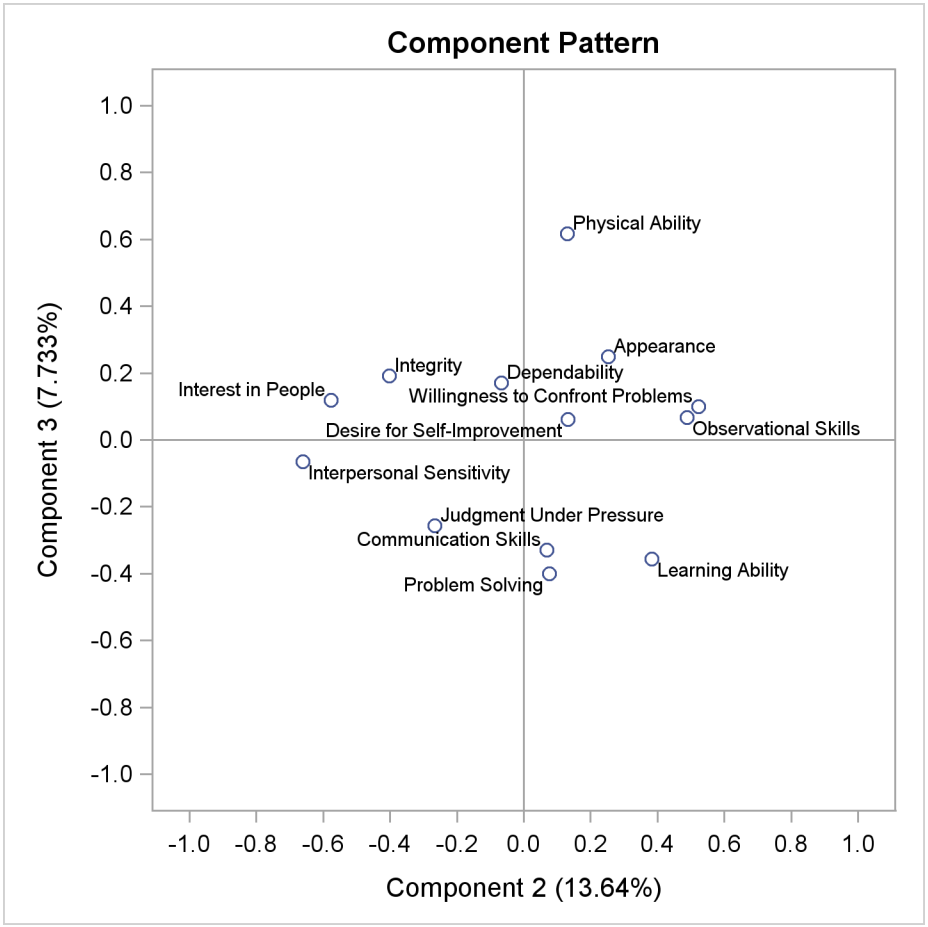
Output 69.3.5 Pattern Plot of Component 2 by Component 1



Output 69.3.6 Pattern Plot of Component 3 by Component 1

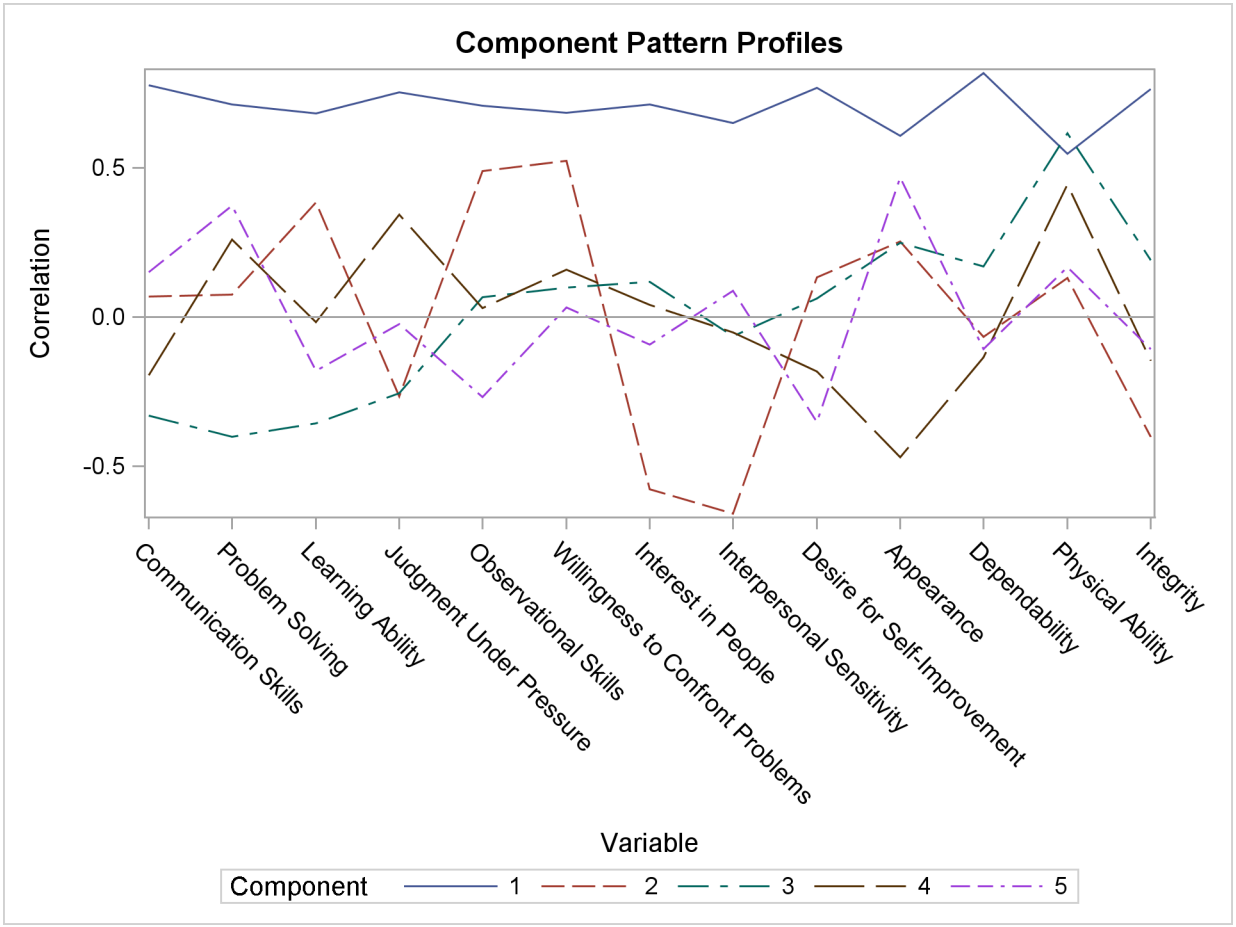


Output 69.3.7 Pattern Plot of Component 3 by Component 2



Output 69.3.8 shows a component pattern profile. As it was shown in the pattern plots, the nearly horizontal profile from the first component indicates that the first component is mostly correlated evenly across all variables.

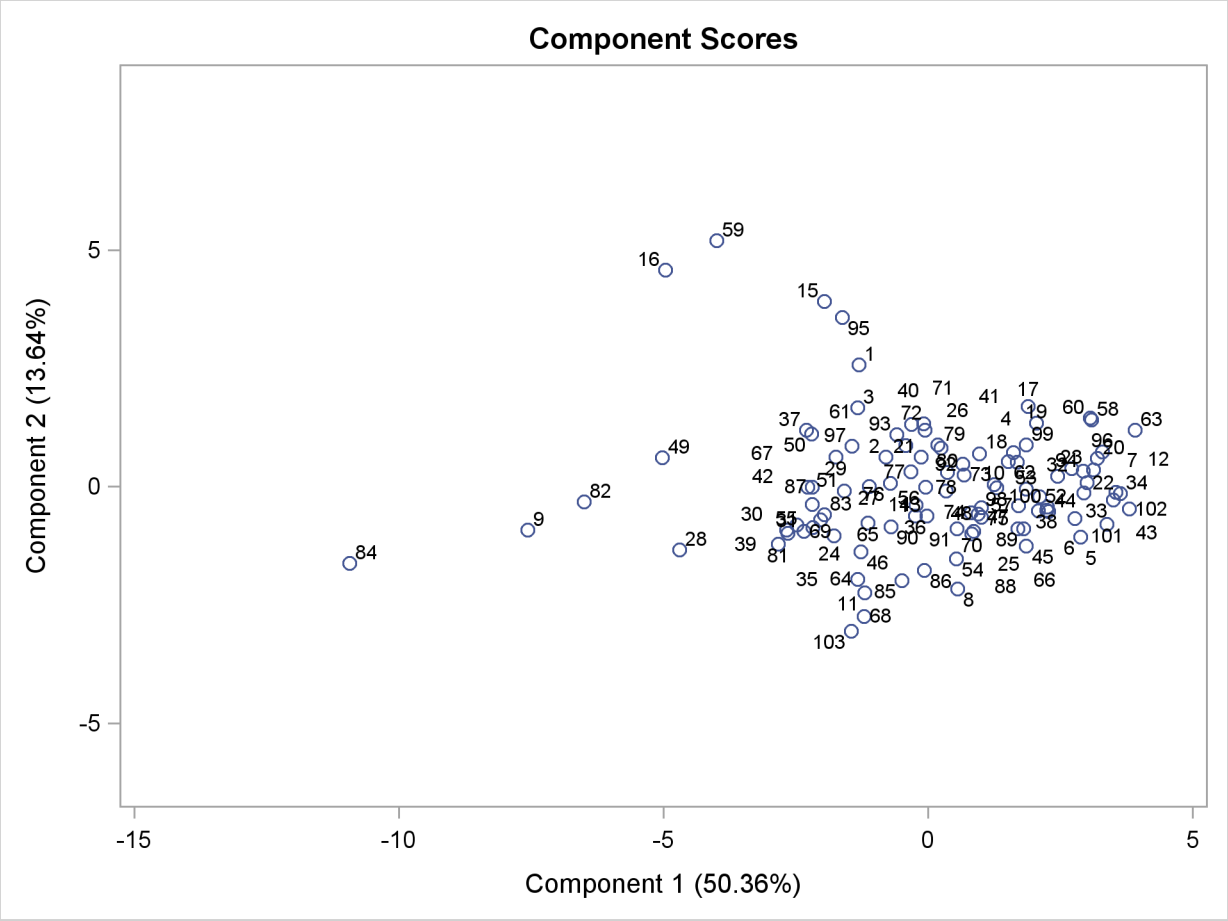
Output 69.3.8 Component Pattern Profile Plot from the PRINCOMP Procedure



Output 69.3.9 through Output 69.3.11 display the pairwise component score plots. Observation numbers are used as the plotting symbol.

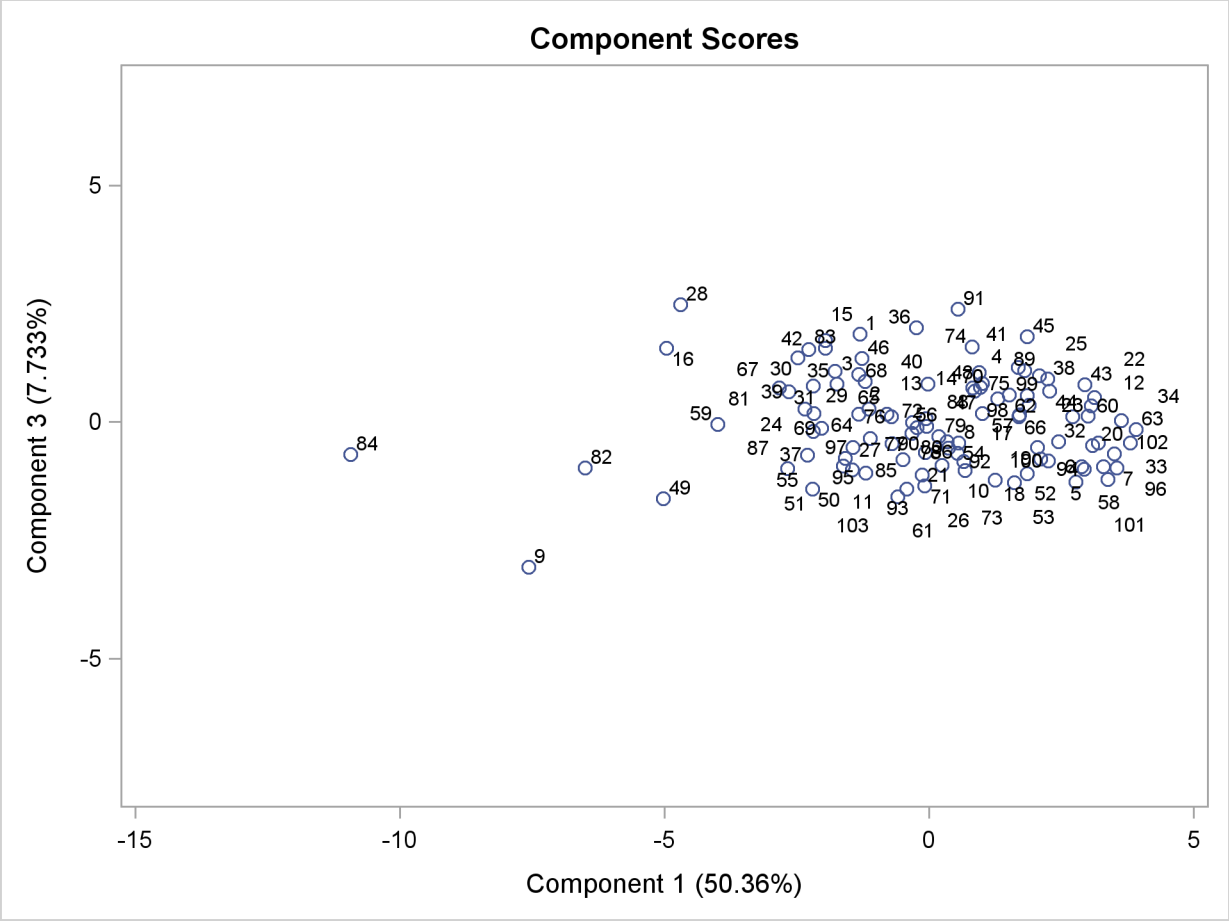
Output 69.3.9 shows a scatter plot of the first and third components. Observations 82, 9, and 84 seem like outliers on the first component; Observations 16 and 59 can be potential outliers on the second component.

Output 69.3.9 Component 2 versus Component 1



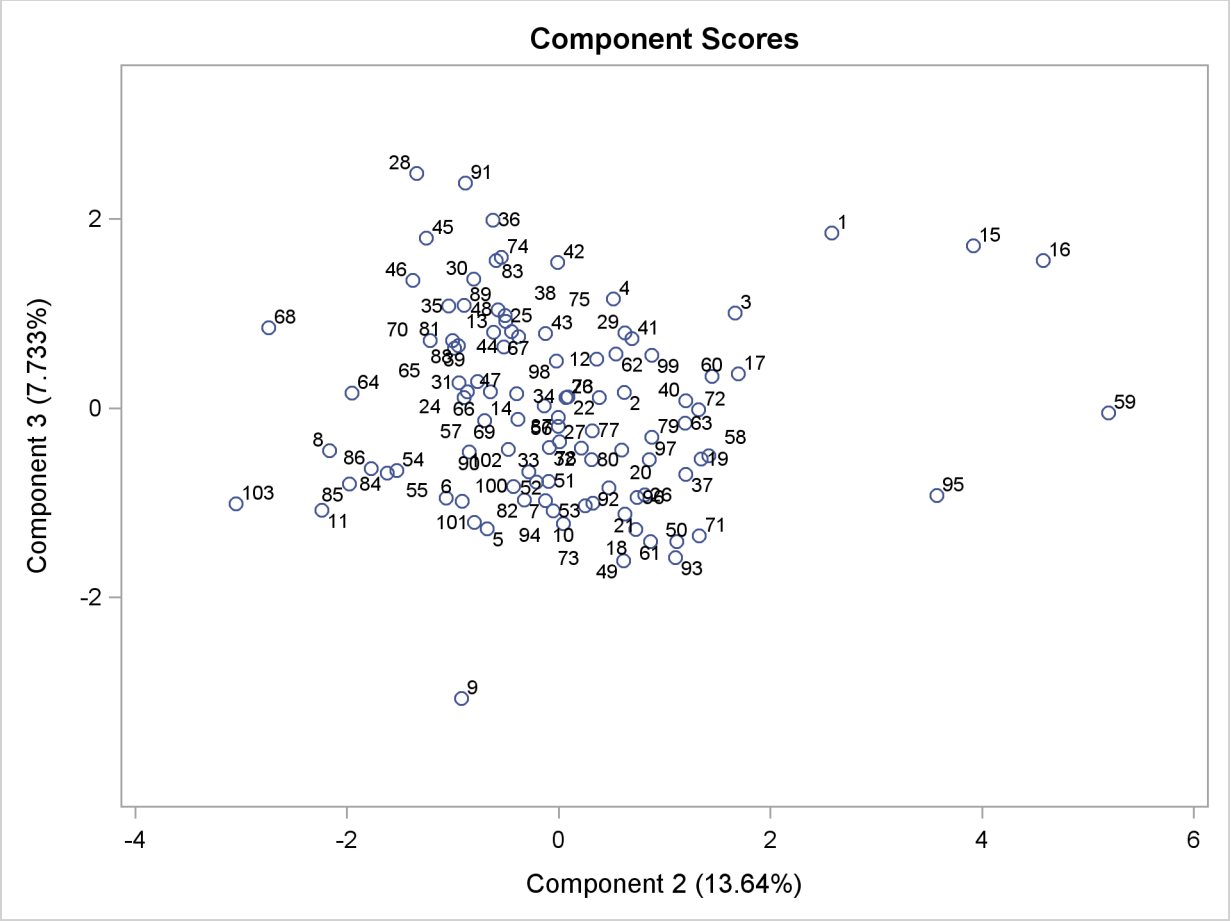
Output 69.3.10 shows a scatter plot of the first and third components. Observations 82, 9, and 84 seem like outliers on the first component.

Output 69.3.10 Component 3 versus Component 1



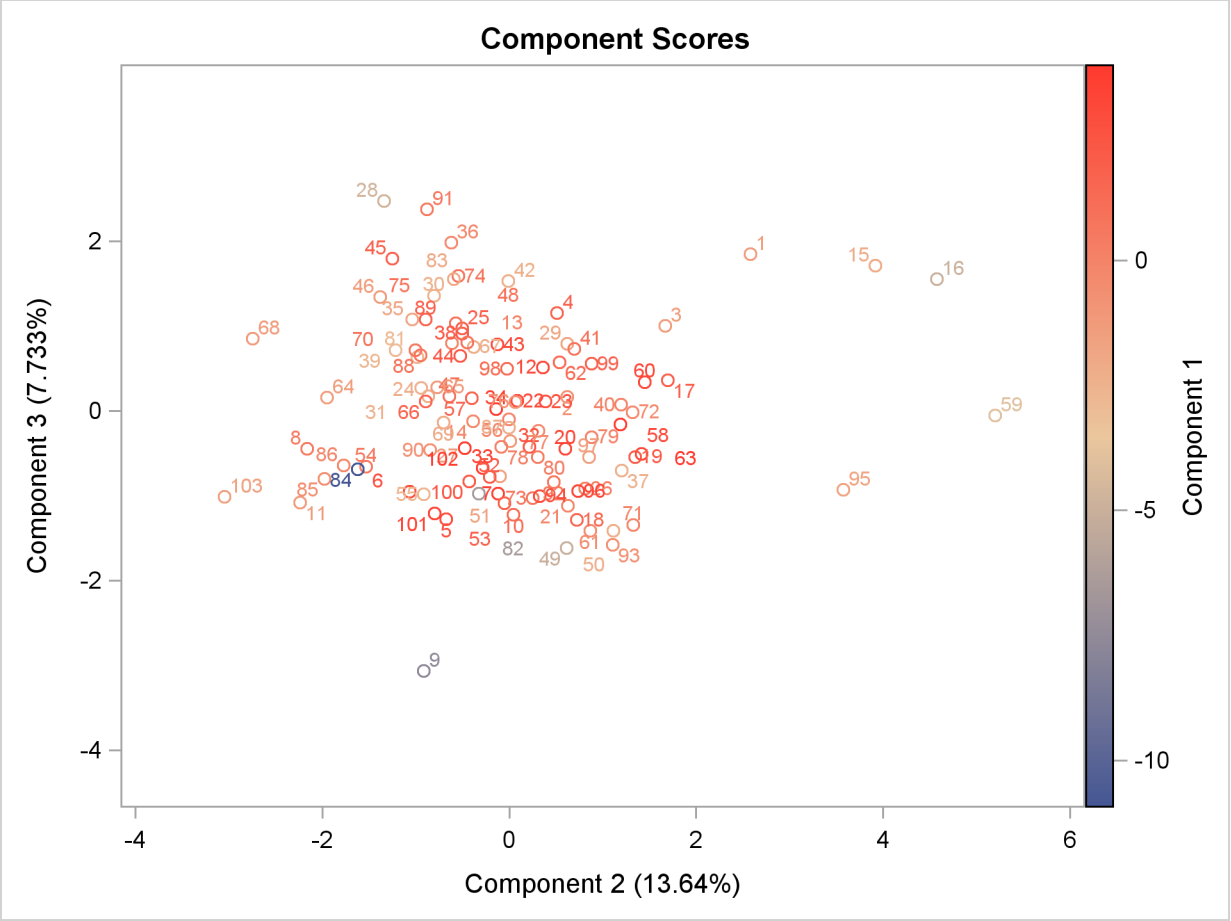
Output 69.3.11 shows a scatter plot of the second and third components. Observations 95, 15, 16, and 59 can be potential outliers on the second component.

Output 69.3.11 Component 3 versus Component 2



Output 69.3.12 shows a scatter plot of the second and third components, displaying density with color. Color interpolation is based on the first component, such as in the statistical style, going from blue (minimum density) to tan (median density) and to red (maximum density).

Output 69.3.12 Component 3 versus Component 2, Painted by Component 1



References

Cooley, W. W. and Lohnes, P. R. (1971), *Multivariate Data Analysis*, New York: John Wiley & Sons.

Gnanadesikan, R. (1977), *Methods for Statistical Data Analysis of Multivariate Observations*, New York: John Wiley & Sons.

Hotelling, H. (1933), "Analysis of a Complex of Statistical Variables into Principal Components," *Journal of Educational Psychology*, 24, 417–441, 498–520.

Kshirsagar, A. M. (1972), *Multivariate Analysis*, New York: Marcel Dekker.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979), *Multivariate Analysis*, London: Academic Press.

Morrison, D. F. (1976), *Multivariate Statistical Methods*, Second Edition, New York: McGraw-Hill.

Pearson, K. (1901), "On Lines and Planes of Closest Fit to Systems of Points in Space," *Philosophical Magazine*, 6(2), 559–572.

Rao, C. R. (1964), "The Use and Interpretation of Principal Component Analysis in Applied Research," *Sankhya A*, 26, 329–358.

Chapter 70

The PRINQUAL Procedure

Contents

Overview: PRINQUAL Procedure **5196**
Getting Started: PRINQUAL Procedure **5198**
Syntax: PRINQUAL Procedure **5202**
PROC PRINQUAL Statement 5203
BY Statement 5211
FREQ Statement 5211
ID Statement 5212
TRANSFORM Statement 5212
WEIGHT Statement 5220
Details: PRINQUAL Procedure **5220**
The Three Methods of Variable Transformation 5220
Understanding How PROC PRINQUAL Works 5222
Splines 5226
Missing Values 5227
Controlling the Number of Iterations 5227
Performing a Principal Component Analysis of Transformed Data 5228
Using the MAC Method 5229
Output Data Set 5229
Avoiding Constant Transformations 5232
Constant Variables 5233
Character OPSCORE Variables 5233
REITERATE Option Usage 5233
Passive Observations 5234
Computational Resources 5235
Displayed Output 5236
ODS Table Names 5237
ODS Graphics 5237
Examples: PRINQUAL Procedure **5238**
Example 70.1: Multidimensional Preference Analysis of Automobile Data . 5238
Example 70.2: Principal Components of Basketball Rankings 5245
References **5254**

Overview: PRINQUAL Procedure

The PRINQUAL procedure performs principal component analysis (PCA) of qualitative, quantitative, or mixed data. PROC PRINQUAL is based on the work of Kruskal and Shepard (1974); Young, Takane, and de Leeuw (1978); Young (1981); and Winsberg and Ramsay (1983). PROC PRINQUAL finds linear and nonlinear transformations of variables, using the method of alternating least squares, that optimize properties of the transformed variables' correlation or covariance matrix. Nonoptimal transformations such as logarithm and rank are also available. You can use ODS Graphics to display the results. You can use PROC PRINQUAL to do the following:

- fit metric and nonmetric principal component analyses
- perform metric and nonmetric multidimensional preference (MDPREF) analyses (Carroll 1972)
- transform data prior to their use in other analyses
- reduce the number of variables for subsequent use in regression analyses, cluster analyses, and other analyses
- detect nonlinear relationships

PROC PRINQUAL provides three methods, each of which seeks to optimize a different property of the transformed variables' covariance or correlation matrix. These methods are as follows:

- maximum total variance, or MTV
- minimum generalized variance, or MGV
- maximum average correlation, or MAC

The MTV method is based on a PCA model, and it is the most commonly used method. All three methods attempt to find transformations that decrease the rank of the covariance matrix computed from the transformed variables. Transforming the variables to maximize the total variance accounted for by a few linear combinations locates the observations in a space with a dimensionality that approximates the stated number of linear combinations as much as possible, given the transformation constraints. Transforming the variables to minimize their generalized variance or maximize the average correlations also reduces the dimensionality, but without a stated target for the final dimensionality. See the section "The Three Methods of Variable Transformation" on page 5220 for more information about all three methods.

The data can contain variables measured on nominal, ordinal, interval, and ratio scales of measurement (Siegel 1956). Any mix is allowed with all methods. PROC PRINQUAL can do the following:

- transform nominal variables by optimally scoring the categories (Fisher 1938)
- transform ordinal variables monotonically by scoring the ordered categories so that order is weakly preserved (adjacent categories can be merged) and the covariance matrix is optimized. You can undo ties optimally or leave them tied (Kruskal 1964). You can also transform ordinal variables to ranks.
- transform interval and ratio scale of measurement variables linearly, or transform them nonlinearly with spline transformations (de Boor 1978; van Rijckevorsel 1982) or monotone spline transformations (Winsberg and Ramsay 1983). In addition, nonoptimal transformations for logarithm, rank, exponential, power, logit, and inverse trigonometric sine are available.
- estimate missing data without constraint, with category constraints (missing values within the same group get the same value), and with order constraints (missing value estimates in adjacent groups can be tied to preserve a specified ordering). See Gifi (1990) and Young (1981).

The transformed qualitative (nominal and ordinal) variables can be thought of as being quantified by the analysis, with the quantification done in the context set by the algorithm. The data are quantified so that the proportion of variance accounted for by a stated number of principal components is locally maximized, the generalized variance of the variables is locally minimized, or the average of the correlations is locally maximized.

The PROC PRINQUAL iterations produce a set of transformed variables. Each variable's new scoring satisfies a set of constraints based on the original scoring of the variable and the specified transformation type. First, all variables are required to satisfy standardization constraints; that is, all variables have a fixed mean and variance. The other constraints include linear constraints, weak order constraints, category constraints, and smoothness constraints. The new set of scores is selected from the sets of possible scorings that do not violate the constraints so that the method criterion is locally optimized.

Getting Started: PRINQUAL Procedure

PROC PRINQUAL can be used to fit a principal component model with nonlinear transformations of the variables and graphically display the results. This example finds monotonic transformations of ratings of automobiles.

```
title 'Ratings for Automobiles Manufactured in 1980';

data cars;
   input Origin $ 1-8 Make $ 10-19 Model $ 21-36
         (MPG Reliability Acceleration Braking Handling Ride
          Visibility Comfort Quiet Cargo) (1.);
   datalines;
GMC      Buick      Century         3334444544
GMC      Buick      Electra         2434453555
GMC      Buick      Lesabre         2354353545

... more lines ...

GMC      Pontiac    Sunbird         3134533234
;

ods graphics on;

proc prinqual data=cars plots=all maxiter=100;
   transform monotone(mpg -- cargo);
   id model;
run;

ods graphics off;
```

The PROC PRINQUAL statement names the input data set Cars. The ODS GRAPHICS statement, along with the PLOTS=ALL option, requests all graphical displays. The MDPREF option requests the PCA plot with the scores (automobiles) represented as points and the structure (variables) represented as vectors. By default, the vector lengths are increased by a factor of 2.5 to produce a better graphical display. If instead you were to specify MDPREF=1, you would get the actual vectors, and they would all be short and would end near the origin where there are a lot of points. It is often the case that increasing the vector lengths by a factor of 2 or 3 makes a better graphical display, so by default the vector lengths are increased by a factor of 2.5. Up to 100 iterations are requested with the MAXITER= option. All of the numeric variable are specified with a MONOTONE transformation, so their original values, 1 to 5, are optimally rescored to maximize fit to a two-component model while preserving the original order. The Model variable provides the labels for the row points in the plot.

The iteration history table is shown in Figure 70.1. The monotonic transformations allow the PCA to account for 5% more variance in two principal components than the ordinary PCA model applied to the untransformed data.

Figure 70.1 Automobile Ratings Iteration History

Ratings for Automobiles Manufactured in 1980

The PRINQUAL Procedure

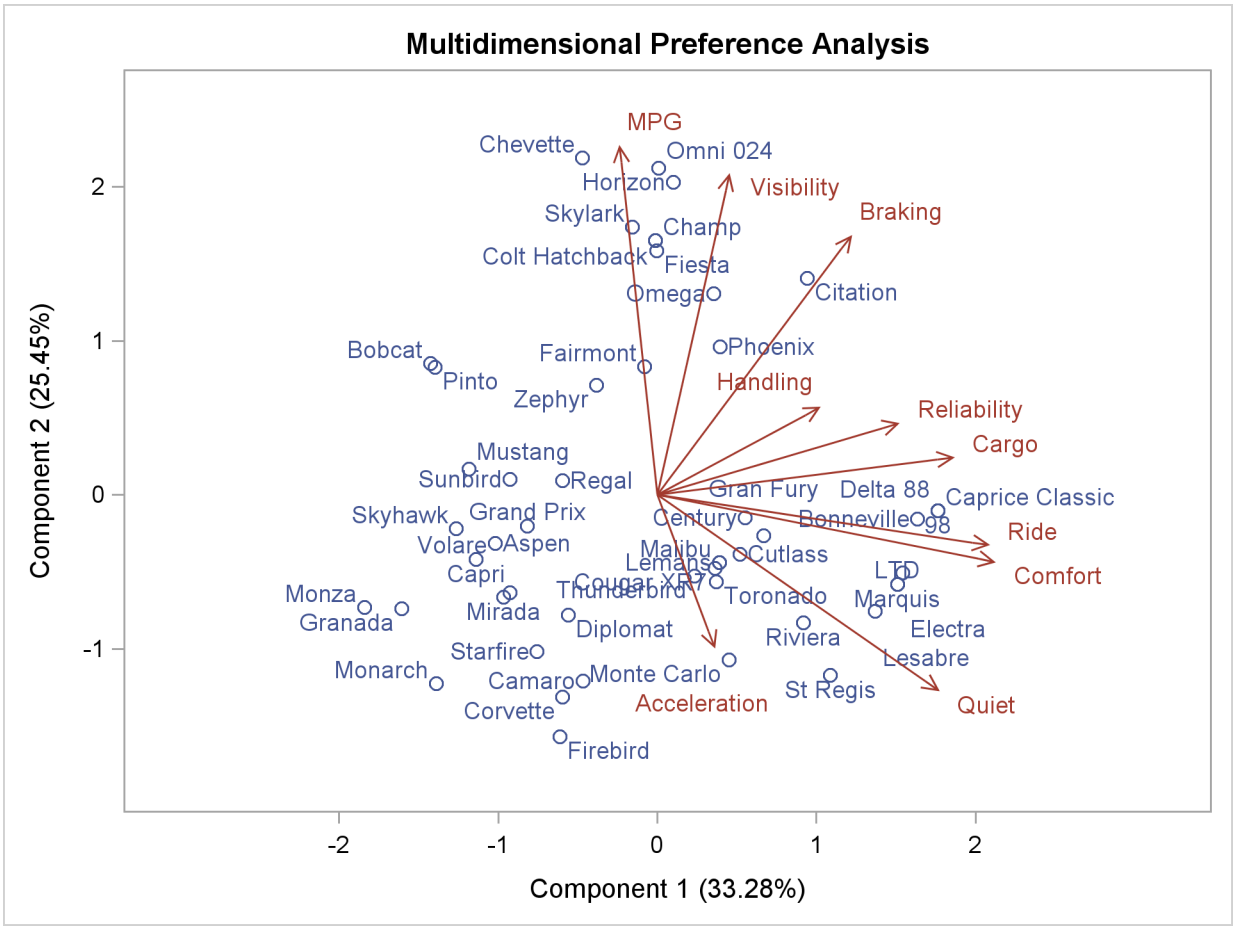
PRINQUAL MTV Algorithm Iteration History

Iteration Number	Average Change	Maximum Change	Proportion of Variance	Criterion Change	Note
1	0.18087	1.24219	0.53742		
2	0.06916	0.77503	0.57244	0.03502	
3	0.04653	0.38237	0.57978	0.00734	
4	0.03387	0.18682	0.58300	0.00321	
5	0.02661	0.13506	0.58484	0.00185	
6	0.01730	0.09213	0.58600	0.00115	
7	0.00969	0.07107	0.58660	0.00061	
8	0.00705	0.04798	0.58685	0.00025	
9	0.00544	0.03482	0.58699	0.00014	
10	0.00442	0.02641	0.58708	0.00009	
11	0.00363	0.02062	0.58714	0.00006	
12	0.00298	0.01643	0.58717	0.00004	
13	0.00245	0.01325	0.58720	0.00002	
14	0.00201	0.01077	0.58721	0.00002	
15	0.00165	0.00880	0.58723	0.00001	
16	0.00136	0.00721	0.58723	0.00001	
17	0.00112	0.00591	0.58724	0.00001	
18	0.00092	0.00485	0.58724	0.00000	
19	0.00075	0.00399	0.58724	0.00000	
20	0.00062	0.00328	0.58725	0.00000	
21	0.00051	0.00269	0.58725	0.00000	
22	0.00042	0.00221	0.58725	0.00000	
23	0.00035	0.00182	0.58725	0.00000	
24	0.00028	0.00149	0.58725	0.00000	
25	0.00023	0.00123	0.58725	0.00000	
26	0.00019	0.00101	0.58725	0.00000	
27	0.00016	0.00083	0.58725	0.00000	
28	0.00013	0.00068	0.58725	0.00000	
29	0.00011	0.00056	0.58725	0.00000	
30	0.00009	0.00046	0.58725	0.00000	
31	0.00007	0.00038	0.58725	0.00000	
32	0.00006	0.00031	0.58725	0.00000	
33	0.00005	0.00025	0.58725	0.00000	
34	0.00004	0.00021	0.58725	0.00000	
35	0.00003	0.00017	0.58725	0.00000	
36	0.00003	0.00014	0.58725	0.00000	
37	0.00002	0.00012	0.58725	0.00000	
38	0.00002	0.00010	0.58725	0.00000	
39	0.00001	0.00008	0.58725	0.00000	
40	0.00001	0.00006	0.58725	0.00000	
41	0.00001	0.00005	0.58725	0.00000	
42	0.00001	0.00004	0.58725	0.00000	Converged

Algorithm converged.

The PCA biplot in Figure 70.2 shows the transformed automobile ratings projected into the two-dimensional plane of the analysis. The automobiles on the left tend to be smaller than the autos on the right, and the autos at the top tend to be cheaper than the autos at the bottom. The vectors can help you interpret the plot of the scores. Longer vectors show the variables that better fit the two-dimensional model. A larger component of them is in the plane of the plot. In contrast, shorter vectors show the variables that do not fit the two-dimensional model as well. They tend to be located less in the plot and more away from the plot; hence their projection into the plot is shorter. To envision this, lay a pencil on your desk directly under a light, and slowly rotate it up to form a 90-degree angle with your desk. As you do so, the shadow or projection of the pencil onto your desk will get progressively shorter. The results show, for example, that the Chevette would be expected to do well on gas mileage but not well on quiet and acceleration. In contrast, the Corvette and the Firebird have the opposite pattern.

Figure 70.2 Automobile Ratings PCA Biplot



There are many patterns shown in the transformations in Figure 70.3. The transformation of Braking, for example, is not very different from the original scoring. The optimal scoring for other variables, such as Acceleration and Handling, is binary. Automobiles are differentiated by high versus everything else or low versus everything else.

Figure 70.3 Automobile Ratings Transformations

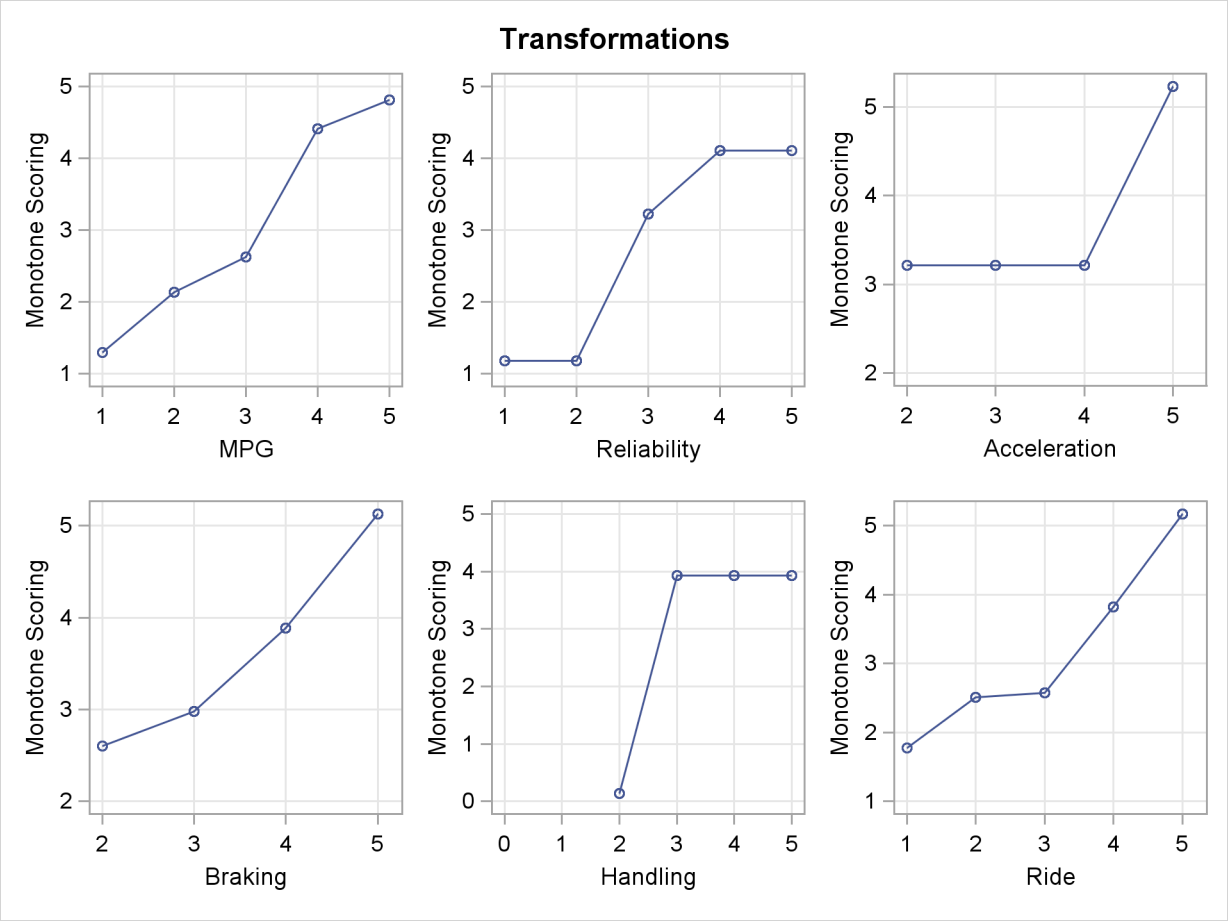
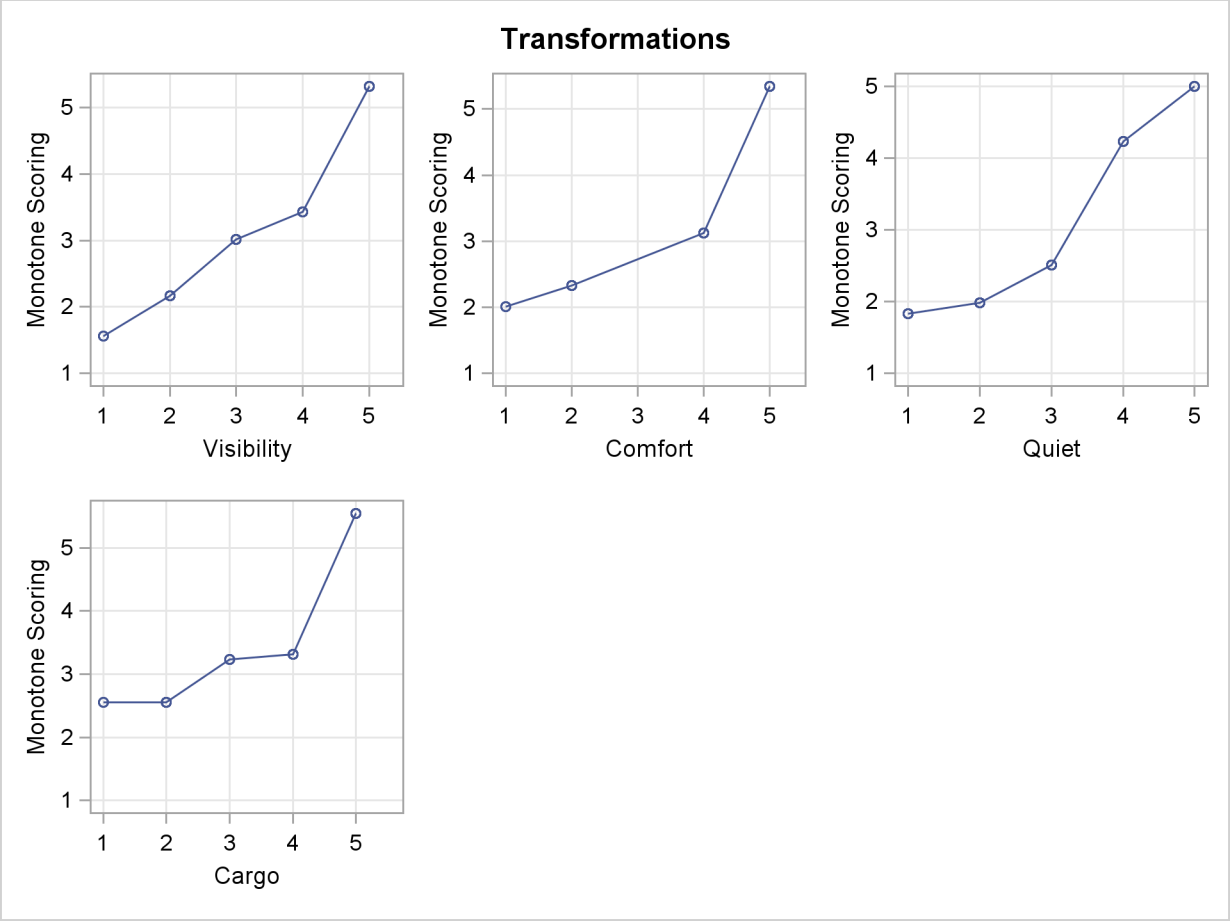


Figure 70.3 *continued*



Syntax: PRINQUAL Procedure

The following statements are available in PROC PRINQUAL.

PROC PRINQUAL < *options* > ;
  TRANSFORM *transform(variables* < / *t-options* >*)*
    < *transform(variables* < / *t-options* >*) . . .* > ;
  ID *variables* ;
  FREQ *variable* ;
  WEIGHT *variable* ;
  BY *variables* ;

To use PROC PRINQUAL, you need the PROC PRINQUAL and TRANSFORM statements. You can abbreviate all *options* and *t-options* to their first three letters. This is a special feature of PROC PRINQUAL and is not generally true of other SAS/STAT procedures.

The rest of this section provides detailed syntax information about each of the preceding statements, beginning with the PROC PRINQUAL statement. The remaining statements are described in alphabetical order.

PROC PRINQUAL Statement

PROC PRINQUAL < *options* > **;**

The PROC PRINQUAL statement invokes the PRINQUAL procedure. Optionally, this statement identifies an input data set, creates an output data set, specifies the algorithm and other computational details, and controls displayed output. The options listed in Table 70.1 are available in the PROC PRINQUAL statement.

Table 70.1 Summary of PROC PRINQUAL Statement Options

Option	Description
Input Data Set Options	
DATA=	specifies input SAS data set
Output Data Set Details	
APPROXIMATIONS	outputs approximations to transformed variables
APREFIX=	specifies prefix for approximation variables
CORRELATIONS	outputs correlations and component structure matrix
MDPREF=	specifies a multidimensional preference analysis
OUT=	specifies output data set
PREFIX=	specifies prefix for principal component scores
REPLACE	replaces raw data with transformed data
SCORES	outputs principal component scores
STANDARD	standardizes principal component scores
TPREFIX=	specifies prefix for transformed variables
TSTANDARD=	specifies transformation standardization
Method and Iterations	
CCONVERGE=	specifies minimum criterion change
CHANGE=	specifies number of first iteration to be displayed
CONVERGE=	specifies minimum data change
COVARIANCE	analyzes covariances
DUMMY	initializes using dummy variables
INITITER=	specifies number of MAC initialization iterations
MAXITER=	specifies maximum number of iterations
METHOD=	specifies iterative algorithm
NOCHECK	suppresses numerical error checking
N	specifies number of principal components
REFRESH=	specifies number of MGV models before refreshing
REITERATE	restarts iterations
SINGULAR=	specifies singularity criterion
TYPE=	specifies input observation type
Missing Data Handling	
MONOTONE=	includes monotone special missing values
NOMISS	excludes observations with missing values
UNTIE=	unties special missing values

Table 70.1 *continued*

Option	Description
Control Displayed Output	
NOPRINT	suppresses displayed output
PLOTS=	specifies ODS Graphics details

The following list describes these options in alphabetical order.

APREFIX=*name*

APR=*name*

specifies a prefix for naming the approximation variables. By default, APREFIX=A. Specifying the APREFIX= option also implies the APPROXIMATIONS option.

APPROXIMATIONS

APPROX

APP

includes principal component approximations to the transformed variables (Eckart and Young 1936) in the output data set. Variable names are constructed from the value of the APREFIX= option and the input variable names. If you specify the APREFIX= option, then approximations are automatically included. If you specify the APPROXIMATIONS option and not the APREFIX= option, then the APPROXIMATIONS option uses the default, APREFIX=A, to construct the variable names.

CCONVERGE=*n*

CCO=*n*

specifies the minimum change in the criterion being optimized that is required to continue iterating. By default, CCONVERGE=0.0. The CCONVERGE= option is ignored for METHOD=MAC. For the MGV method, specify CCONVERGE=–2 to ensure data convergence.

CHANGE=*n*

CHA=*n*

specifies the number of the first iteration to be displayed in the iteration history table. The default is CHANGE=1. When you specify a larger value for n, the first $n - 1$ iterations are not displayed, thus speeding up the analysis. The CHANGE= option is most useful with the MGV method, which is much slower than the other methods.

CONVERGE=*n*

CON=*n*

specifies the minimum average absolute change in standardized variable scores that is required to continue iterating. By default, CONVERGE=0.00001. Average change is computed over only those variables that can be transformed by the iterations—that is, all LINEAR, OPSCORE, MONOTONE, UNTIE, SPLINE, MSPLINE, and SSPLINE variables and nonoptimal transformation variables with missing values. For more information, see the section "Optimal Transformations" on page 5215.

COVARIANCE
COV

computes the principal components from the covariance matrix. The variables are always centered to mean zero. If you do not specify the COVARIANCE option, the variables are also standardized to variance one, which means the analysis is based on the correlation matrix.

CORRELATIONS
COR

includes correlations and the component structure matrix in the output data set. By default, this information is not included.

DATA=*SAS-data-set*

specifies the SAS data set to be analyzed. The data set must be an ordinary SAS data set; it cannot be a TYPE=CORR or TYPE=COV data set. If you omit the DATA= option, PROC PRINQUAL uses the most recently created SAS data set.

DUMMY
DUM

expands variables specified for OPSCORE optimal transformations to dummy variables for the initialization (Tenenhaus and Vachette 1977). By default, the initial values of OPSCORE variables are the actual data values. The dummy variable nominal initialization requires considerable time and memory, so it might not be possible to use the DUMMY option with large data sets. No separate report of the initialization is produced. Initialization results are incorporated into the first iteration displayed in the iteration history table. For details, see the section "Optimal Transformations" on page 5215.

INITITER=*n*
INI=*n*

specifies the number of MAC iterations required to initialize the data before starting MTV or MGV iterations. By default, INITITER=0. The INITITER= option is ignored if METHOD=MAC.

MAXITER=*n*
MAX=*n*

specifies the maximum number of iterations. By default, MAXITER=30.

MDPREF<=*n***>**
MDP<=*n***>**

specifies a multidimensional preference analysis by implying the STANDARD, SCORES, and CORRELATIONS options. This option also suppresses warnings when there are more variables than observations.

When ODS Graphics is enabled, an MDPREF plot is produced with points for each row and vectors for each column. Often, the vectors are short, and a better graphical display is produced when the vectors are stretched. The absolute lengths of each vector can optionally be changed by specifying MDPREF=n. Then the vector coordinates are all multiplied by n. Usually, n will be a value such as 2, 2.5, or 3. The default is 2.5. Specify MDPREF=1 to see the vectors without any stretching. The relative lengths of the different vectors is important and interpretable, and these are preserved by the stretching.

METHOD=MAC | MGV | MTV

MET=MAC | MGV | MTV

specifies the optimization method. By default, METHOD=MTV. Values of the METHOD= option are MTV, for maximum total variance; MGV, for minimum generalized variance; and MAC, for maximum average correlation. You can use the MAC method when all variables are positively correlated or when no monotonicity constraints are placed on any transformations. See the section "The Three Methods of Variable Transformation" on page 5220 for more information.

MONOTONE=*two-letters*

MON=*two-letters*

specifies the first and last special missing value in the list of those special missing values to be estimated using within-variable order and category constraints. By default, there are no order constraints on missing value estimates. The *two-letters* value must consist of two letters in alphabetical order. For example, MONOTONE=DF means that the estimate of .D must be less than or equal to the estimate of .E, which must be less than or equal to the estimate of .F; no order constraints are placed on estimates of ._, .A through .C, and .G through .Z. For details, see the sections "Missing Values" on page 5227 and "Optimal Scaling" on page 7259 in Chapter 90, "The TRANSREG Procedure."

N=*n*

specifies the number of principal components to be computed. By default, N=2.

NOCHECK

NOC

turns off computationally intensive numerical error checking for the MGV method. If you do not specify the NOCHECK option, the procedure computes R square from the squared length of the predicted values vector and compares this value to the R square computed from the error sum of squares that is a byproduct of the sweep algorithm (Goodnight 1978). If the two values of R square differ by more than the square root of the value of the SINGULAR= option, a warning is displayed, the value of the REFRESH= option is halved, and the model is refit after refreshing. Specifying the NOCHECK option slightly speeds up the algorithm. Note that other less computationally intensive error checking is always performed.

NOMISS

NOM

excludes all observations with missing values from the analysis, but does not exclude them from the OUT= data set. If you omit the NOMISS option, PROC PRINQUAL simultaneously computes the optimal transformations of the nonmissing values and estimates the missing values that minimize squared error.

Casewise deletion of observations with missing values occurs when you specify the NOMISS option, when there are missing values in IDENTITY variables, when there are weights less than or equal to 0, or when there are frequencies less than 1. Excluded observations are output with a blank value for the _TYPE_ variable, and they have a weight of 0. They do not contribute to the analysis but are scored and transformed as *supplementary* or passive observations. See the sections "Passive Observations" on page 5234 and "Missing Values" on page 5227 for more information about excluded observations and missing data.

NOPRINT

NOP

suppresses the display of all output. This option disables the Output Delivery System (ODS), including ODS Graphics, for the duration of the procedure. For more information, see Chapter 20, "Using the Output Delivery System."

OUT=*SAS-data-set*

specifies an output SAS data set that contains results of the analysis. If you omit the OUT= option, PROC PRINQUAL still creates an output data set and names it by using the DATA*n* convention. If you want to create a permanent SAS data set, you must specify a two-level name. (See the discussion in *SAS Language Reference: Concepts*.) You can use the REPLACE, APPROXIMATIONS, SCORES, and CORRELATIONS options to control what information is included in the output data set. For details, see the section "Output Data Set" on page 5229.

PLOTS <(*global-plot-options***)> <=** *plot-request* **<(***options***)>>**

PLOTS <(*global-plot-options***)> <= (***plot-request* **<(***options***)> <...** *plot-request* **<(***options***)>>)>**

controls the plots produced through ODS Graphics. When you specify only one plot request, you can omit the parentheses from around the plot request. Here are some examples:

```
plots=none
plots=transformation
plots(unpack)=transformation
```

You must enable ODS Graphics before requesting plots, for example, like this:

```
ods graphics on;

proc prinqual plots=all;
   transformation spline(x1-x10);
run;

ods graphics off;
```

For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." If you have enabled ODS Graphics but do not specify the PLOTS= option, then PROC PRINQUAL produces an MDPREF plot when the MDPREF option is specified.

The global plot options include the following:

FLIP

FLI

flips or interchanges the X-axis and Y-axis dimensions for MDPREF plots. The FLIP option can be specified either as a global plot option (for example, PLOTS(FLIP)) or with the MDPREF option (for example, PLOTS=MDPREF(FLIP)).

INTERPOLATE

INT

uses observations that are excluded from the analysis for interpolation in the fit and transformation plots. By default, observations with zero weight are excluded from all plots. These include observations with a zero, negative, or missing weight or frequency and observations excluded due to missing and invalid values. You can specify PLOTS(INTERPOLATE)=(*plot-requests*) to include some of these observations in the plots. You can use this option, for example, with sparse data sets to show smoother functions over the range of the data (see the section "The PLOTS(INTERPOLATE) Option" on page 7302 in Chapter 90, "The TRANSREG Procedure").

ONLY

ONL

suppresses the default plots. Only plots specifically requested are displayed.

UNPACKPANEL

UNPACK

UNP

suppresses paneling. By default, multiple plots can appear in some output panels. Specify UNPACKPANEL to get each plot in a separate panel.

The plot requests include the following:

ALL

produces all appropriate plots.

TRANSFORMATION

TRA

TRANSFORMATION(UNPACK)

TRA(UNP)

plots the variable transformations. By default, multiple plots can appear in some output panels. Specify UNPACKPANEL to display each plot in a separate panel.

MDPREF

MDP

plots multidimensional preference analysis results. The MDPREF plot can also be requested by specifying the MDPREF option in the PROC statement outside the PLOTS= option.

NONE

suppresses all plots.

PREFIX=*name*

PRE=*name*

specifies a prefix for naming the principal components. By default, PREFIX=Prin. As a result, the principal component default names are Prin1, Prin2,..., Prin$n$.

REFRESH=*n*

REF=*n*

specifies the number of variables to scale in the MGV method before computing a new inverse. By default, REFRESH=5. PROC PRINQUAL uses the REFRESH= option in the sweep algorithm of the MGV method. Large values for the REFRESH= option make the method run faster but with more numerical error. Small values make the method run more slowly but with more numerical accuracy.

REITERATE

REI

enables PROC PRINQUAL to use previous transformations as starting points. The REITERATE option affects only variables that are iteratively transformed (specified as LINEAR, SPLINE, MSPLINE, SSPLINE, UNTIE, OPSCORE, and MONOTONE). For iterative transformations, the REITERATE option requests a search in the input data set for a variable that consists of the value of the TPREFIX= option followed by the original variable name. If such a variable is found, it is used to provide the initial values for the first iteration. The final transformation is a member of the transformation family defined by the original variable, not the transformation family defined by the initialization variable. See the section "REITERATE Option Usage" on page 5233.

REPLACE

REP

replaces the original data with the transformed data in the output data set. The names of the transformed variables in the output data set correspond to the names of the original variables in the input data set. If you do not specify the REPLACE option, both original variables and transformed variables (with names constructed from the TPREFIX= option and the original variable names) are included in the output data set.

SCORES

SCO

includes principal component scores in the output data set. By default, scores are not included.

SINGULAR=*n*

SIN=*n*

specifies the largest value within rounding error of zero. By default, SINGULAR=1E–8. PROC PRINQUAL uses the value of the SINGULAR= option for checking $(1 - R^2)$ when constructing full-rank matrices of predictor variables, checking denominators before dividing, and so on.

STANDARD

STD

standardizes the principal component scores in the output data set to mean zero and variance one instead of the default mean zero and variance equal to the corresponding eigenvalue. See the SCORES option.

TPREFIX=*name*

TPR=*name*

specifies a prefix for naming the transformed variables. By default, TPREFIX=T. The TPREFIX= option is ignored if you specify the REPLACE option.

TSTANDARD=CENTER | NOMISS | ORIGINAL | Z

TST=CEN | NOM | ORI | Z

specifies the standardization of the transformed variables in the OUT= data set. By default, TSTANDARD=ORIGINAL. When you specify the TSTANDARD= option in the PROC statement, it the default standardization for all variables. When you specify TSTANDARD= as a *t-option*, it overrides the default standardization just for selected variables.

CENTER centers the output variables to mean zero, but the variances are the same as the variances of the input variables.

NOMISS sets the means and variances of the transformed variables in the OUT= data set, computed over all output values that correspond to nonmissing values in the input data set, to the means and variances computed from the nonmissing observations of the original variables. The TSTANDARD=NOMISS specification is useful with missing data. When a variable is linearly transformed, the final variable contains the original nonmissing values and the missing value estimates. In other words, the nonmissing values are unchanged. If your data have no missing values, TSTANDARD=NOMISS and TSTANDARD=ORIGINAL produce the same results.

ORIGINAL sets the means and variances of the transformed variables to the means and variances of the original variables. This is the default.

Z standardizes the variables to mean zero, variance one.

For nonoptimal variable transformations, the means and variances of the original variables are actually the means and variances of the nonlinearly transformed variables, unless you specify the ORIGINAL nonoptimal *t-option* in the TRANSFORM statement. For example, if a variable X with no missing values is specified as LOG, then, by default, the final transformation of X is simply LOG(X), not LOG(X) standardized to the mean of X and variance of X.

TYPE='*text* '|*name*

TYP='*text* '|*name*

specifies the valid value for the _TYPE_ variable in the input data set. If PROC PRINQUAL finds an input _TYPE_ variable, it uses only observations with a _TYPE_ value that matches the TYPE= value. This enables a PROC PRINQUAL OUT= data set containing correlations to be used as input to PROC PRINQUAL without requiring a WHERE statement to exclude the correlations. If a _TYPE_ variable is not in the data set, all observations are used. The default is TYPE='SCORE', so if you do not specify the TYPE= option, only observations with _TYPE_ = 'SCORE' are used.

PROC PRINQUAL displays a note when it reads observations with blank values of _TYPE_, but it does not automatically exclude those observations. Data sets created by the TRANSREG and PRINQUAL procedures have blank _TYPE_ values for those observations that

were excluded from the analysis due to nonpositive weights, nonpositive frequencies, or missing data. When these observations are read again, they are excluded for the same reason that they were excluded from their original analysis, not because their _TYPE_ value is blank.

UNTIE=*two-letters*
UNT=*two-letters*
specifies the first and last special missing values in the list of those special missing values that are to be estimated with within-variable order constraints but no category constraints. The *two-letters* value must consist of two letters in alphabetical order. By default, there are category constraints but no order constraints on special missing value estimates. For details, see the section "Missing Values" on page 5227. Also, see the section "Optimal Scaling" on page 7259 in Chapter 90, "The TRANSREG Procedure."

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC PRINQUAL to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC PRINQUAL. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

FREQ Statement

FREQ *variable* **;**

If one variable in the input data set represents the frequency of occurrence for other values in the observation, list the variable's name in a FREQ statement. PROC PRINQUAL then treats the data set as if each observation appeared n times, where n is the value of the FREQ variable for the observation. Noninteger values of the FREQ variable are truncated to the largest integer less than the FREQ value. The observation is used in the analysis only if the value of the FREQ statement variable is greater than or equal to 1.

ID Statement

ID *variables* **;**

The ID statement includes additional character or numeric variables in the output data set. The variables must be contained in the input data set.

TRANSFORM Statement

TRANSFORM *transform(variables < / t-options >) < transform(variables < / t-options >) . . . >* **;**

The TRANSFORM statement lists the variables to be analyzed (*variables*) and specifies the transformation (*transform*) to apply to each variable listed. You must specify a transformation for each variable list in the TRANSFORM statement. The variables are variables in the data set. The *t-options* are transformation options that provide details for the transformation; these depend on the *transform* chosen. The *t-options* are listed after a slash in the parentheses that enclose the variables.

For example, the following statements find a quadratic polynomial transformation of all variables in the data set:

```
proc princqual;
   transform spline(_all_ / degree=2);
run;
```

Or, if N1 through N10 are nominal variables and M1 through M10 are ordinal variables, you can use the following statements:

```
proc princqual;
   transform opscore(N1-N10) monotone(M1-M10);
run;
```

The following sections describe the transformations available (specified with *transform*) and the options available for some of the transformations (specified with *t-options*).

Families of Transformations

There are three types of transformation families: nonoptimal, optimal, and other. The families are described as follows:

Nonoptimal transformations	preprocess the specified variables, replacing each one with a single new nonoptimal, nonlinear transformation.
Optimal transformations	replace the specified variables with new, iteratively derived optimal transformation variables that fit the specified model better than the original variable (except in contrived cases where the transformation fits the model exactly as well as the original variable).

Other transformations are the IDENTITY and SSPLINE transformations. These do not fit into either of the preceding categories.

Table 70.2 summarizes the transformations in each family.

Table 70.2 Transformation Families

Transformation	Description
Nonoptimal Transformations	
ARSIN	inverse trigonometric sine
EXP	exponential
LOG	logarithm
LOGIT	logit
POWER	raises variables to specified power
RANK	transforms to ranks
Optimal Transformations	
LINEAR	linear
MONOTONE	monotonic, ties preserved
MSPLINE	monotonic B-spline
OPSCORE	optimal scoring
SPLINE	B-spline
UNTIE	monotonic, ties not preserved
Other Transformations	
IDENTITY	identity, no transformation
SSPLINE	iterative smoothing spline

The *transform* is followed by a variable (or list of variables) enclosed in parentheses. Optionally, depending on the *transform*, the parentheses can also contain *t-options*, which follow the variables and a slash. For example, the following statement computes the LOG transformation of X and Y:

```
transform log(X Y);
```

A more complex example follows:

```
transform spline(Y / nknots=2) log(X1 X2 X3);
```

The preceding statement uses the SPLINE transformation of the variable Y and the LOG transformation of the variables X1, X2, and X3. In addition, it uses the NKNOTS= option with the SPLINE transformation and specifies two knots.

The rest of this section provides syntax details for members of the three families of transformations. The *t-options* are discussed in the section "Transformation Options (t-options)" on page 5216.

Nonoptimal Transformations

Nonoptimal transformations are computed before the iterative algorithm begins. Nonoptimal transformations create a single new transformed variable that replaces the original variable. The new variable is not transformed by the subsequent iterative algorithms (except for a possible linear transformation and missing value estimation).

The following list provides syntax and details for nonoptimal variable transformations.

ARSIN

ARS

finds an inverse trigonometric sine transformation. Variables specified in the ARSIN *transform* must be numeric and in the interval ($-1.0 \leq x \leq 1.0$), and they are typically continuous.

EXP

exponentiates variables (x is transformed to a^x). To specify the value of a, use the PARAMETER= *t-option*. By default, a is the mathematical constant $e = 2.718\ldots$. Variables specified with the EXP *transform* must be numeric, and they are typically continuous.

LOG

transforms variables to logarithms (x is transformed to $\log_a(x)$). To specify the base of the logarithm, use the PARAMETER= *t-option*. The default is a natural logarithm with base $e = 2.718\ldots$. Variables specified with the LOG *transform* must be numeric and positive, and they are typically continuous.

LOGIT

finds a logit transformation on the variables. The logit of x is $\log(x/(1-x))$. Unlike other transformations, LOGIT does not have a three-letter abbreviation. Variables specified with the LOGIT *transform* must be numeric and in the interval ($0.0 < x < 1.0$), and they are typically continuous.

POWER

POW

raises variables to a specified power (x is transformed to x^a). You must specify the power parameter a by specifying the PARAMETER= *t-option* following the variables.

```
power(variable / parameter=number)
```

You can use POWER for squaring variables (PARAMETER=2), reciprocal transformations (PARAMETER=–1), square roots (PARAMETER=0.5), and so on. Variables specified with the POWER *transform* must be numeric, and they are typically continuous.

RANK

RAN

transforms variables to ranks. Ranks are averaged within ties. The smallest input value is assigned the smallest rank. Variables specified with the RANK *transform* must be numeric.

Optimal Transformations

Optimal transformations are iteratively derived. Missing values for these types of variables can be optimally estimated (see the section "Missing Values" on page 5227). See the sections "OPSCORE, MONOTONE, UNTIE, and LINEAR Transformations" on page 7260 and "SPLINE and MSPLINE Transformations" on page 7261 in Chapter 90, "The TRANSREG Procedure," for more information about the optimal transformations.

The following list provides syntax and details for optimal transformations.

LINEAR

LIN

finds an optimal linear transformation of each variable. For variables with no missing values, the transformed variable is the same as the original variable. For variables with missing values, the transformed nonmissing values have a different scale and origin than the original values. Variables specified with the LINEAR *transform* must be numeric.

MONOTONE

MON

finds a monotonic transformation of each variable, with the restriction that ties are preserved. The Kruskal (1964) secondary least-squares monotonic transformation is used. This transformation weakly preserves order and category membership (ties). Variables specified with the MONOTONE *transform* must be numeric, and they are typically discrete.

MSPLINE

MSP

finds a monotonically increasing B-spline transformation with monotonic coefficients (de Boor 1978; de Leeuw 1986) of each variable. You can specify the DEGREE=, KNOTS=, NKNOTS=, and EVENLY= *t-options* with MSPLINE. By default, PROC PRINQUAL uses a quadratic spline. Variables specified with the MSPLINE *transform* must be numeric, and they are typically continuous.

OPSCORE

OPS

finds an optimal scoring of each variable. The OPSCORE transformation assigns scores to each class (level) of the variable. The Fisher (1938) optimal scoring method is used. Variables specified with the OPSCORE *transform* can be either character or numeric; numeric variables should be discrete.

SPLINE

SPL

finds a B-spline transformation (de Boor 1978) of each variable. By default, PROC PRINQUAL uses a cubic polynomial transformation. You can specify the DEGREE=, KNOTS=, NKNOTS=, and EVENLY *t-options* with SPLINE. Variables specified with the SPLINE *transform* must be numeric, and they are typically continuous.

UNTIE
UNT

finds a monotonic transformation of each variable without the restriction that ties are preserved. PROC PRINQUAL uses the Kruskal (1964) primary least-squares monotonic transformation method. This transformation weakly preserves order but not category membership (it might untie some previously tied values). Variables specified with the UNTIE *transform* must be numeric, and they are typically discrete.

Other Transformations

IDENTITY
IDE

specifies variables that are not changed by the iterations. The IDENTITY transformation is used for variables when no transformation and no missing data estimation are desired. However, the REFLECT, ADDITIVE, TSTANDARD=Z, and TSTANDARD=CENTER options can linearly transform all variables, including IDENTITY variables, after the iterations. Observations with missing values in IDENTITY variables are excluded from the analysis, and no optimal scores are computed for missing values in IDENTITY variables. Variables specified with the IDENTITY *transform* must be numeric.

SSPLINE
SSP

finds an iterative smoothing spline transformation of each variable. The SSPLINE transformation does not generally minimize squared error. You can specify the smoothing parameter with either the SM= *t-option* or the PARAMETER= *t-option*. The default smoothing parameter is SM=0. Variables specified with the SSPLINE *transform* must be numeric, and they are typically continuous.

Transformation Options (t-options)

If you use a nonoptimal, optimal, or other transformation, you can use *t-options*, which specify additional details of the transformation. The *t-options* are specified within the parentheses that enclose variables and are listed after a slash. For example:

```
proc prinqual;
   transform spline(X Y / nknots=3);
run;
```

The preceding statements find an optimal variable transformation (SPLINE) of the variables X and Y and use a *t-option* to specify the number of knots (NKNOTS=). The following is a more complex example:

```
proc prinqual;
   transform spline(Y / nknots=3) spline(X1 X2 / nknots=6);
run;
```

These statements use the SPLINE transformation for all three variables and use *t-options* as well; the NKNOTS= option specifies the number of knots for the spline.

The following sections discuss the *t-options* available for nonoptimal, optimal, and other transformations.

Table 70.3 summarizes the *t-options*.

Table 70.3 Transformation Options

Option	Description
Nonoptimal Transformation	
ORIGINAL	uses original mean and variance
Parameter Specification	
PARAMETER=	specifies miscellaneous parameters
SM	specifies smoothing parameter
Spline	
DEGREE=	specifies the degree of the spline
EVENLY	spaces the knots evenly
KNOTS=	specifies the interior knots or break points
NKNOTS=	creates *n* knots
Other t-options	
NAME=	renames variables
REFLECT	reflects the variable around the mean
TSTANDARD=	specifies transformation standardization

Nonoptimal Transformation t-options

ORIGINAL

ORI

matches the variable's final mean and variance to the mean and variance of the original variable. By default, the mean and variance are based on the transformed values. The ORIGINAL *t-option* is available for all of the nonoptimal transformations.

Parameter t-options

PARAMETER=*number*

PAR=*number*

specifies the transformation parameter. The PARAMETER= *t-option* is available for the EXP, LOG, POWER, SMOOTH, and SSPLINE transformations. For EXP, the parameter is the value to be exponentiated; for LOG, the parameter is the base value; and for POWER, the parameter is the power. For SMOOTH and SSPLINE, the parameter is the raw smoothing parameter. (See the SM= option for an alternative way to specify the smoothing parameter.) The default for the PARAMETER= *t-option* for the LOG and EXP transformations is $e = 2.718\ldots$. The default parameter for SSPLINE is computed from SM=0. For the POWER transformation, you must specify the PARAMETER= *t-option*; there is no default.

SM=*n*

specifies a smoothing parameter in the range 0 to 100, just like PROC GPLOT uses. For example, SM=50 in PROC PRINQUAL is equivalent to I=SM50 on the SYMBOL statement with PROC GPLOT. You can specify the SM= *t-option* only with the SSPLINE transformation. The smoothness of the function increases as the value of the smoothing parameter increases. By default, SM=0.

Spline t-options

The following *t-options* are available with the SPLINE and MSPLINE optimal transformations.

DEGREE=*n*

DEG=*n*

specifies the degree of the B-spline transformation. The degree must be a nonnegative integer. The defaults are DEGREE=3 for SPLINE variables and DEGREE=2 for MSPLINE variables.

The polynomial degree should be a small integer, usually 0, 1, 2, or 3. Larger values are rarely useful. If you have any doubt as to what degree to specify, use the default.

EVENLY<=*n***>**

EVE<=*n***>**

is used with the NKNOTS= *t-option* to space the knots evenly. The differences between adjacent knots are constant. If you specify NKNOTS=k, k knots are created at

$$\text{minimum} + i\,((\text{maximum} - \text{minimum})/(k+1))$$

for $i = 1, \ldots, k$. For example, if you specify

```
spline(X / knots=2 evenly)
```

and the variable X has a minimum of 4 and a maximum of 10, then the two interior knots are 6 and 8. Without the EVENLY *t-option*, the NKNOTS= *t-option* places knots at percentiles, so the knots are not evenly spaced.

By default for the SPLINE and MSPLINE transformations, the smaller exterior knots are all the same and just a little smaller than the minimum. Similarly, by default, the larger exterior knots are all the same and just a little larger than the maximum. However, if you specify EVENLY=*n*, then the n exterior knots are evenly spaced as well. The number of exterior knots must be greater than or equal to the degree. You can specify values larger than the degree when you want to interpolate slightly beyond the range or your data. The exterior knots must be less than the minimum or greater than the maximum, and hence the knots across all sets are not precisely equally spaced. For example, with data ranging from 0 to 10, and with EVENLY=3 and NKNOTS=4, the first exterior knots are –4.000000000001, –2.000000000001, and –0.000000000001, the interior knots are 2, 4, 6, and 8, and the second exterior knots are 10.000000000001, 12.000000000001, and 14.000000000001.

KNOTS=*number-list* | *n* **TO** *m* **BY** *p*

KNO=*number-list* | *n* **TO** *m* **BY** *p*

specifies the interior knots or break points. By default, there are no knots. The first time you specify a value in the knot list, it indicates a discontinuity in the nth (from DEGREE=n) derivative of the transformation function at the value of the knot. The second mention of a value indicates a discontinuity in the $(n-1)$th derivative of the transformation function at the value of the knot. Knots can be repeated any number of times to decrease the smoothness at the break points, but the values in the knot list can never decrease.

You cannot use the KNOTS= *t-option* with the NKNOTS= *t-option*. You should keep the number of knots small. (See the section "Specifying the Number of Knots" on page 7262 in Chapter 90, "The TRANSREG Procedure.")

NKNOTS=*n*

NKN=*n*

creates n knots, the first at the $100/(n+1)$ percentile, the second at the $200/(n+1)$ percentile, and so on. Knots are always placed at data values; there is no interpolation. For example, if NKNOTS=3, knots are placed at the 25th percentile, the median, and the 75th percentile. By default, NKNOTS=0. The NKNOTS= *t-option* must be ≥ 0.

You cannot use the NKNOTS= *t-option* with the KNOTS= *t-option*. You should keep the number of knots small. (See the section "Specifying the Number of Knots" on page 7262 in Chapter 90, "The TRANSREG Procedure.")

Other t-options

The following *t-options* are available for all transformations.

NAME=*(variable-list)*

NAM=*(variable-list)*

renames variables as they are used in the TRANSFORM statement. This option allows a variable to be used more than once. For example, if the variable X is a character variable, then the following step stores both the original character variable X and a numeric variable XC that contains category numbers in the output data set.

```
proc princual data=A n=1 out=B;
   transform linear(Y) opscore(X / name=(XC));
   id X;
run;
```

REFLECT

REF

reflects the transformation

$$y = -(y - \bar{y}) + \bar{y}$$

after the iterations are completed and before the final standardization and results calculations.

TSTANDARD=CENTER | NOMISS | ORIGINAL | Z

TST=CEN | NOM | ORI | Z

specifies the standardization of the transformed variables in the OUT= data set. By default, TSTANDARD=ORIGINAL. When the TSTANDARD= option is specified in the PROC PRINQUAL statement, it specifies the default standardization for all variables. When you specify TSTANDARD= as a *t-option*, it overrides the default standardization only for selected variables.

WEIGHT Statement

WEIGHT *variable* **;**

When you use a WEIGHT statement, a weighted residual sum of squares is minimized. The WEIGHT statement has no effect on degrees of freedom or number of observations, but the weights affect most other calculations. The observation is used in the analysis only if the value of the WEIGHT statement variable is greater than 0.

Details: PRINQUAL Procedure

The Three Methods of Variable Transformation

The three methods of variable transformation provided by PROC PRINQUAL are discussed in the following sections.

The Maximum Total Variance (MTV) Method

The MTV method (Young, Takane, and de Leeuw 1978) is based on the principal component model, and it attempts to maximize the sum of the first r eigenvalues of the covariance matrix. This method transforms variables to be (in a least-squares sense) as similar to linear combinations of r principal component score variables as possible, where r can be much smaller than the number of variables. This maximizes the total variance of the first r components (the trace of the covariance matrix of the first r principal components). See *SAS Technical Report R-108*.

On each iteration, the MTV algorithm alternates classical principal component analysis (Hotelling 1933) with optimal scaling (Young 1981). When all variables are ordinal preference ratings, this corresponds to MDPREF analysis (Carroll 1972). You can request the dummy variable initialization method suggested by Tenenhaus and Vachette (1977), who independently propose the same iterative algorithm for nominal and interval scale-of-measurement variables.

The Minimum Generalized Variance (MGV) Method

The MGV method (Sarle 1984) uses an iterated multiple regression algorithm in an attempt to minimize the determinant of the covariance matrix of the transformed variables. This method transforms each variable to be (in a least-squares sense) as similar to linear combinations of the remaining variables as possible. This locally minimizes the generalized variance of the transformed variables, the determinant of the covariance matrix, the volume of the parallelepiped defined by the transformed variables, and the sphericity (the extent to which a quadratic form in the optimized covariance matrix defines a sphere). See *SAS Technical Report R-108*.

On each iteration for each variable, the MGV algorithm alternates multiple regression with optimal scaling. The multiple regression involves predicting the selected variable from all other variables. You can request a dummy variable initialization by using a modification of the Tenenhaus and Vachette (1977) method that is appropriate with a regression algorithm. This method can be viewed as a way of investigating the nature of the linear and nonlinear dependencies in, and the rank of, a data matrix containing variables that can be nonlinearly transformed. This method tries to create a less-than-full-rank data matrix. The matrix contains the transformation of each variable that is most similar to what the other transformed variables predict.

The Maximum Average Correlation (MAC) Method

The MAC method (de Leeuw 1985) uses an iterated constrained multiple regression algorithm in an attempt to maximize the average of the elements of the correlation matrix. This method transforms each variable to be (in a least-squares sense) as similar to the average of the remaining variables as possible.

On each iteration for each variable, the MAC algorithm alternates computing an equally weighted average of the other variables with optimal scaling. The MAC method is similar to the MGV method in that each variable is scaled to be as similar to a linear combination of the other variables as possible, given the constraints on the transformation. However, optimal weights are not computed. You can use the MAC method when all variables are positively correlated or when no monotonicity constraints are placed on any transformations. Do not use this method with negatively correlated variables when some optimal transformations are constrained to be increasing because the signs of the correlations are not taken into account. The MAC method is useful as an initialization method for the MTV and MGV methods.

Understanding How PROC PRINQUAL Works

In the following example, PROC PRINQUAL uses the MTV method to linearize a curved scatter plot. Let

$$\begin{aligned} X_1 &= -1 \text{ to } 1 \text{ by } 0.02 \\ X_2 &= X_1^3 + \epsilon \\ X_3 &= X_2^5 + \epsilon \end{aligned}$$

where ϵ is normal error.

These three variables define a curved swarm of points in three-dimensional space. First, the SGSCATTER procedure is used to display two-dimensional views of these data. Next, PROC PRINQUAL is used to straighten the scatter plot, making it more one-dimensional by finding a smooth transformation of each variable. The N=1 option in the PROC PRINQUAL statement requests one principal component. The TRANSFORM statement requests a cubic spline transformation with nine knots. *Splines* are curves, which are usually required to be continuous and smooth. See the section "Splines" on page 5226 for more information about splines. See Smith (1979) for an excellent introduction to splines.

PROC PRINQUAL transforms each variable to be as much as possible like the first principal component (or more generally, to be close to the space defined by the first N= principal components). One component accounts for 92 percent of the variance of the untransformed data and over 99 percent of the variance of the transformed data (see Figure 70.5). Note that the results did not converge in the default 50 iterations, so more iterations were requested using the MAXITER= option. The transformations are requested by specifying PLOTS=TRANSFORMATION and are displayed in Figure 70.6.

PROC PRINQUAL creates an output data set that contains both the original and transformed variables. The original variables are named X1, X2, and X3, and the transformed variables are named TX1, TX2, and TX3. The transformed variables are displayed using the SGSCATTER procedure in Figure 70.7.

The following statements produce Figure 70.4 through Figure 70.7:

```
ods graphics on;

* Generate Three-Dimensional Data;
data X;
   do X1 = -1 to 1 by 0.02;
      X2 = X1 ** 3 + 0.05 * normal(7);
      X3 = X1 ** 5 + 0.05 * normal(7);
      output;
   end;
run;
```

```
proc sgscatter data=x;
   plot x1*x2 x1*x3 x3*x2;
   run;

* Try to Straighten the Scatter Plot;
proc prinqual data=X n=1 maxiter=2000 plots=transformation out=results;
   title 'Linearize the Scatter Plot';
   transform spline(X1-X3 / nknots=9);
run;


* Plot the Linearized Scatter Plot;
proc sgscatter data=results;
   plot tx1*tx2 tx1*tx3 tx3*tx2;
   run;
```

The three-dimensional data in Figure 70.4 and Figure 70.7 are displayed in three two-dimensional plots, arrayed as if they were three faces of a cube that was flattened as you might flatten a box.

Figure 70.4 Three-Dimensional Scatter Plot

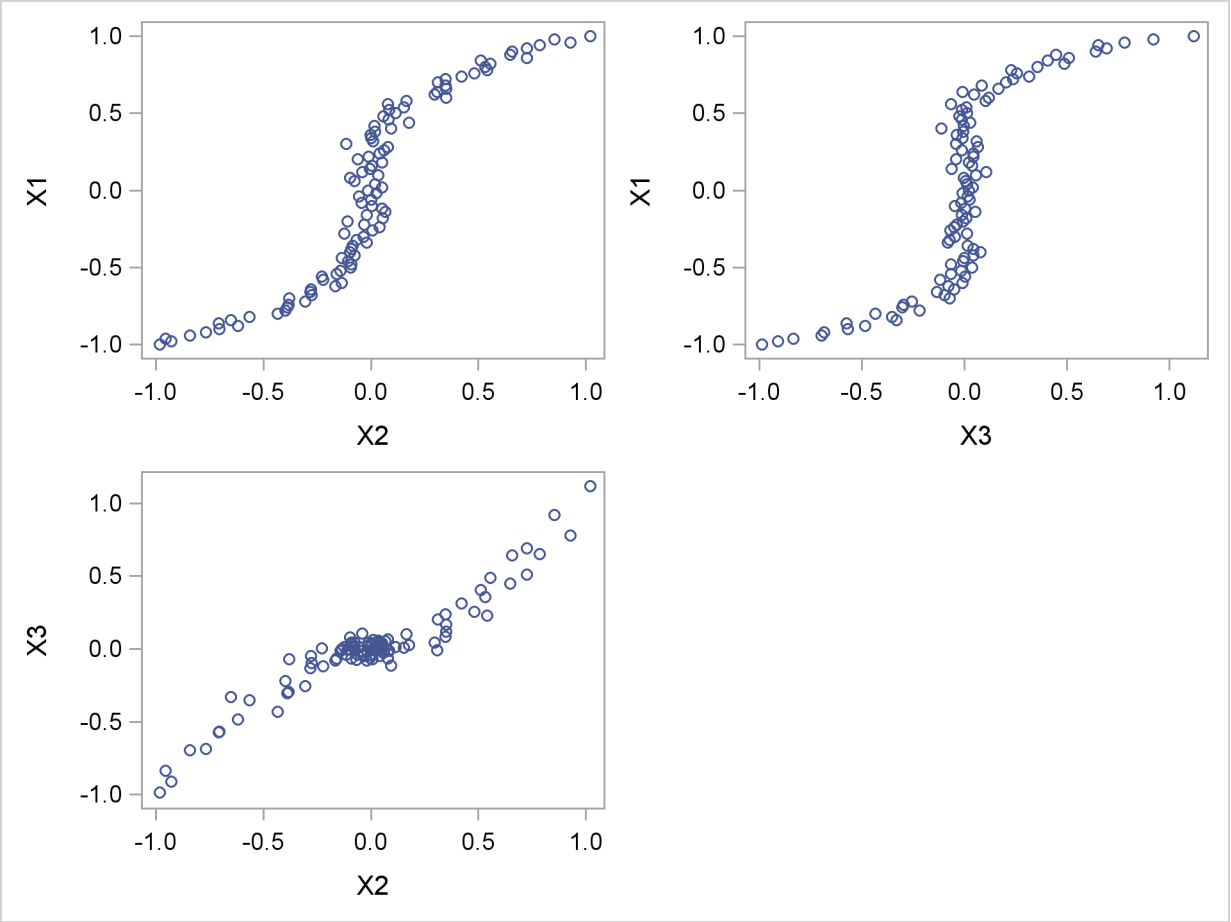


Figure 70.5 PRINQUAL Iteration History

Linearize the Scatter Plot

The PRINQUAL Procedure

PRINQUAL MTV Algorithm Iteration History

Iteration Number	Average Change	Maximum Change	Proportion of Variance	Criterion Change	Note
1	0.15125	0.93453	0.92376		
2	0.04589	0.14682	0.98030	0.05653	
3	0.03154	0.10125	0.98626	0.00596	
4	0.02258	0.06890	0.98890	0.00265	
5	0.01682	0.04777	0.99028	0.00137	
6	0.01297	0.03782	0.99106	0.00078	
7	0.01032	0.03029	0.99154	0.00048	
8	0.00851	0.02514	0.99186	0.00032	
9	0.00722	0.02124	0.99209	0.00023	
10	0.00625	0.01871	0.99226	0.00017	
.					
.					
.					
1670	0.00001	0.00005	0.99371	0.00000	
1671	0.00001	0.00005	0.99371	0.00000	
1672	0.00001	0.00005	0.99371	0.00000	Converged

Algorithm converged.

Figure 70.6 Transformations

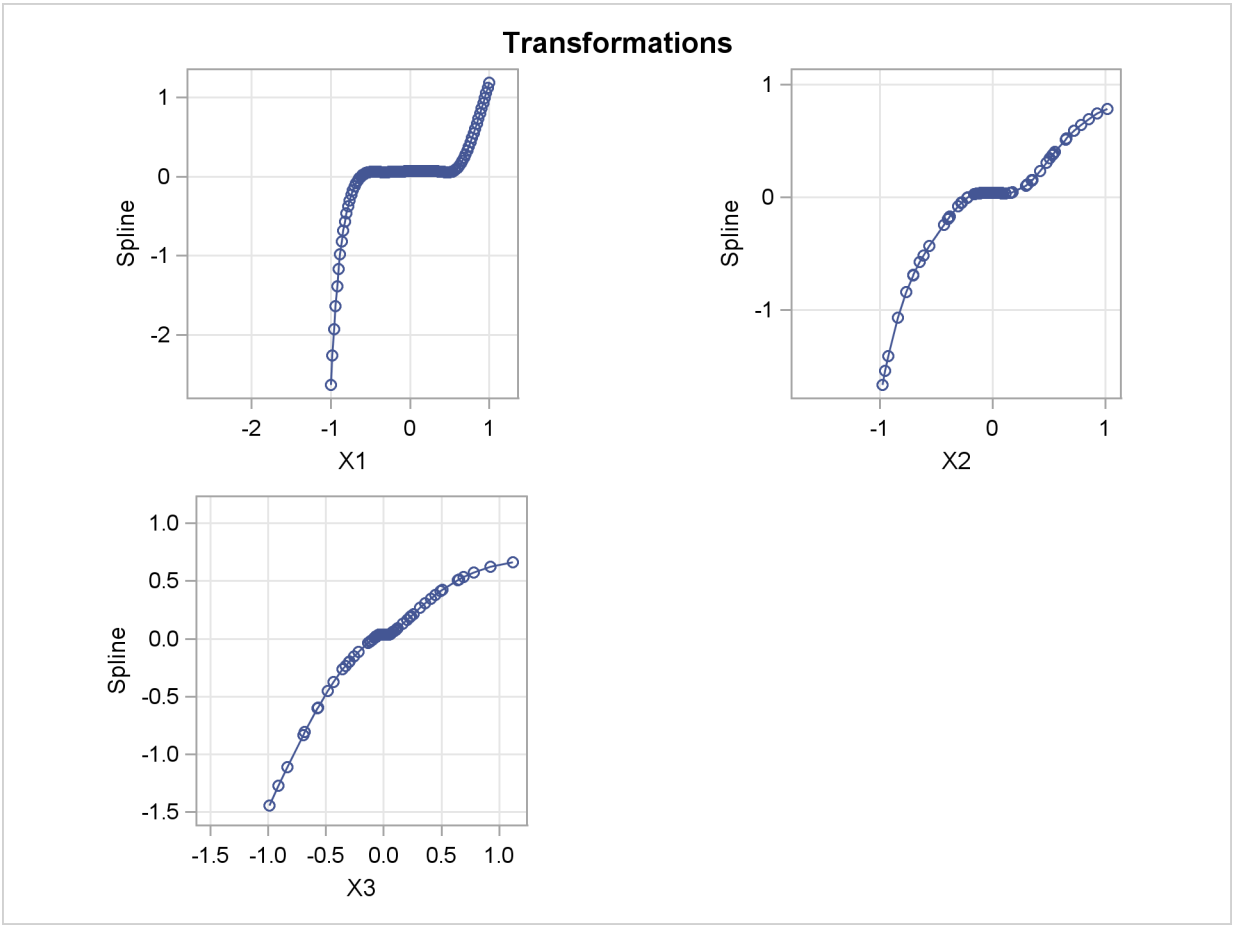
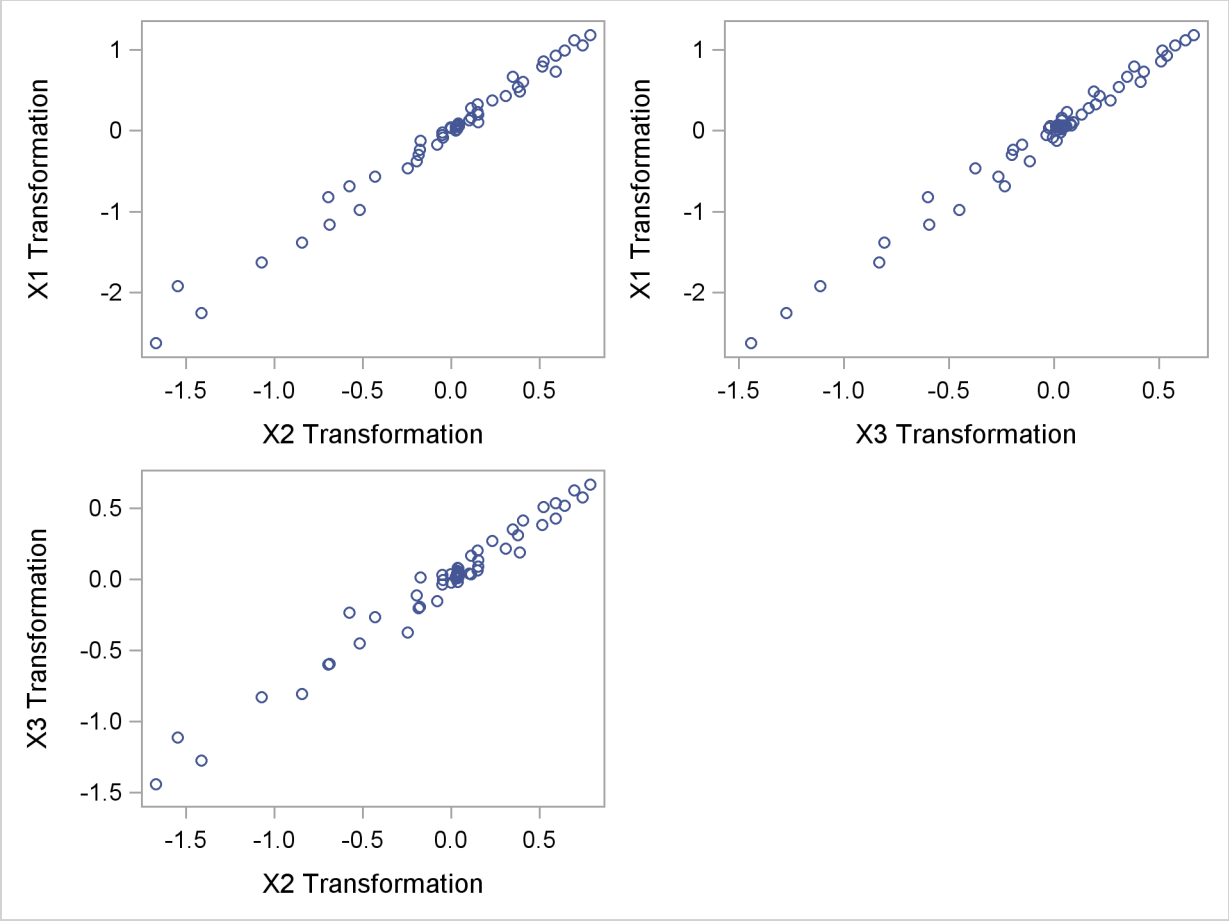


Figure 70.7 Linearized Scatter Plot



Splines

Splines are curves, and they are usually required to be continuous and smooth. Splines are usually defined as piecewise polynomials of degree n with function values and first $n - 1$ derivatives that agree at the points where they join. The abscissa values of the join points are called *knots*. The term "spline" is also used for polynomials (splines with no knots) and piecewise polynomials with more than one discontinuous derivative. Splines with no knots are generally smoother than splines with knots, which are generally smoother than splines with multiple discontinuous derivatives. Splines with few knots are generally smoother than splines with many knots; however, increasing the number of knots usually increases the fit of the spline function to the data. Knots give the curve freedom to bend to follow the data more closely. See Smith (1979) for an excellent introduction to splines. There are many examples and detailed discussions of splines in Chapter 90, "The TRANSREG Procedure." See the sections "Linear and Nonlinear Regression Functions" on page 7209, "Smoothing Splines" on page 7222, "SPLINE and MSPLINE Transformations" on page 7261, "Specifying the Number of Knots" on page 7262, "SPLINE, BSPLINE, and PSPLINE Comparisons" on page 7264, "Linear and Nonlinear Regression Functions" on page 7209, "Simultaneously Fitting Two Regression Functions" on page 7213, and examples Example 90.18 and Example 90.1.

Missing Values

PROC PRINQUAL can estimate missing values, subject to optional constraints, so that the covariance matrix is optimized. The procedure provides several approaches for handling missing data. When you specify the NOMISS option in the PROC PRINQUAL statement, observations with missing values are excluded from the analysis. Otherwise, missing data are estimated, using variable means as initial estimates. Missing values for OPSCORE character variables are treated the same as any other category during the initialization. See the section "Missing Values" on page 7249 in Chapter 90, "The TRANSREG Procedure," for more information about missing data estimation.

Controlling the Number of Iterations

Several options in the PROC PRINQUAL statement control the number of iterations performed. Iteration terminates when any one of the following conditions is satisfied:

- The number of iterations equals the value of the MAXITER= option.
- The average absolute change in variable scores from one iteration to the next is less than the value of the CONVERGE= option.
- The criterion change is less than the value of the CCONVERGE= option.

With the MTV method, the change in the proportion of variance criterion can become negative when the data have converged so that it is numerically impossible, within machine precision, to increase the criterion. Because the MTV algorithm is convergent, a negative criterion change is the result of very small amounts of rounding error. The MGV method displays the average squared multiple correlation (which is not the criterion being optimized), so the criterion change can become negative well before convergence. The MAC method criterion (average correlation) is never computed, so the CCONVERGE= option is ignored for METHOD=MAC. You can specify a negative value for either convergence option if you want to define convergence only in terms of the other convergence option.

With the MGV method, iterations minimize the generalized variance (determinant), but the generalized variance is not reported for two reasons. First, in most data sets, the generalized variance is almost always near zero (or will be after one or two iterations), which is its minimum. This does not mean that iteration is complete; it simply means that at least one multiple correlation is at or near one. The algorithm continues minimizing the determinant in $(m-1), (m-2)$ dimensions, and so on. Because the generalized variance is almost always near zero, it does not provide a good indication of how the iterations are progressing. The mean R square provides a better indication of convergence. The second reason for not reporting the generalized variance is that almost no additional time is required to compute R square values for each step. This is because the error sum of squares is a byproduct of the algorithm at each step. Computing the determinant at the end of each iteration adds more computations to an already computationally intensive algorithm.

You can increase the number of iterations to ensure convergence by increasing the value of the MAXITER= option and decreasing the value of the CONVERGE= option. Because the average absolute change in standardized variable scores seldom decreases below 1E–11, you typically do not specify a value for the CONVERGE= option less than 1E–8 or 1E–10. Most of the data changes occur during the first few iterations, but the data can still change after 50 or even 100 iterations. You can try different combinations of values for the CONVERGE= and MAXITER= options to ensure convergence without extreme overiteration. If the data do not converge with the default specifications, specify the REITERATE option, or try CONVERGE=1E–8 and MAXITER=50, or CONVERGE=1E–10 and MAXITER=200.

Performing a Principal Component Analysis of Transformed Data

PROC PRINQUAL produces an iteration history table that displays (for each iteration) the iteration number, the maximum and average absolute change in standardized variable scores computed over the iteratively transformed variables, the criterion being optimized, and the criterion change. In order to examine the results of the analysis in more detail, you can analyze the information in the output data set by using other SAS procedures.

Specifically, use the PRINCOMP procedure to perform a principal components analysis on the transformed data. PROC PRINCOMP accepts the raw data from PROC PRINQUAL but issues a warning, because the PROC PRINQUAL output data set has _NAME_ and _TYPE_ variables but is not a TYPE=CORR data set. You can ignore this warning.

If the output data set contains both scores and correlations, you must subset it for analysis with PROC PRINCOMP. Otherwise, the correlation observations are treated as ordinary observations and the PROC PRINCOMP results are incorrect. For example, consider the following statements:

```
proc prinqual data=a out=b correlations replace;
   transform spline(var1-var50 / nknots=3);
run;

proc princomp data=b;
   where _TYPE_='SCORE';
run;
```

Also note that the proportion of variance accounted for, as reported by PROC PRINCOMP, can exceed the proportion of variance accounted for in the last PROC PRINQUAL iteration. This is because PROC PRINQUAL reports the variance accounted for by the components analysis that generated the current scaling of the data, not a components analysis of the current scaling of the data.

Using the MAC Method

You can use the MAC algorithm alone by specifying METHOD=MAC, or you can use it as an initialization algorithm for METHOD=MTV and METHOD=MGV analyses by specifying the iteration option INITITER=. If any variables are negatively correlated, do not use the MAC algorithm with monotonic transformations (MONOTONE, UNTIE, and MSPLINE) because the signs of the correlations among the variables are not used when computing variable approximations. If an approximation is negatively correlated with the original variable, monotone constraints would make the optimally scaled variable a constant, which is not allowed (see the section "Avoiding Constant Transformations" on page 5232). When used with other transformations, the MAC algorithm can reverse the scoring of the variables. So, for example, if variable X is designated LOG(X) with METHOD=MAC and TSTANDARD=ORIGINAL, the final transformation (for example, TX) might not be LOG(X). If TX is not LOG(X), it has the same mean as LOG(X) and the same variance as LOG(X), and it is perfectly negatively correlated with LOG(X). PROC PRINQUAL displays a note for every variable that is reversed in this manner.

You can use the METHOD=MAC algorithm to reverse the scorings of some rating variables before a factor analysis. The correlations among bipolar ratings such as 'like - dislike', 'hot - cold', and 'fragile - monumental' are typically both positive and negative. If some items are reversed to say 'dislike - like', 'cold - hot', and 'monumental - fragile', some of the negative signs can be eliminated, and the factor pattern matrix would be cleaner. You can use PROC PRINQUAL with METHOD=MAC and LINEAR transformations to reverse some items, maximizing the average of the intercorrelations.

Output Data Set

PROC PRINQUAL produces an output data set by default. By specifying the OUT=, APPROXIMATIONS, SCORES, REPLACE, and CORRELATIONS options in the PROC PRINQUAL statement, you can name this data set and control its contents.

By default, the procedure creates an output data set that contains variables with _TYPE_='SCORE'. These observations contain original variables, transformed variables, components, or data approximations. If you specify the CORRELATIONS option in the PROC PRINQUAL statement, the data set also contains observations with _TYPE_='CORR'; these observations contain correlations or component structure information.

Structure and Content

The output data set can have 16 different forms, depending on the specified combinations of the REPLACE, SCORES, APPROXIMATIONS, and CORRELATIONS options. You can specify any combination of these options. To illustrate, assume that the data matrix consists of N observations and m variables, and n components are computed. Then define the following:

D the $N \times m$ matrix of original data with variable names that correspond to the names of the variables in the input data set. However, when you use the OPSCORE transformation on character variables, those variables are replaced by numeric variables that contain category numbers.

T the $N \times m$ matrix of transformed data with variable names constructed from the value of the TPREFIX= option (if you do not specify the REPLACE option) and the names of the variables in the input data set

S the $N \times n$ matrix of component scores with variable names constructed from the value of the PREFIX= option and integers

A the $N \times m$ matrix of data approximations with variable names constructed from the value of the APREFIX= option and the names of the variables in the input data set

$\mathbf{R_{TD}}$ the $m \times m$ matrix of correlations between the transformed variables and the original variables with variable names that correspond to the names of the variables in the input data set. When missing values exist, casewise deletion is used to compute the correlations.

$\mathbf{R_{TT}}$ the $m \times m$ matrix of correlations among the transformed variables with the variable names constructed from the value of the TPREFIX= option (if you do not specify the REPLACE option) and the names of the variables in the input data set

$\mathbf{R_{TS}}$ the $m \times n$ matrix of correlations between the transformed variables and the principal component scores (component structure matrix) with variable names constructed from the value of the PREFIX= option and integers

$\mathbf{R_{TA}}$ the $m \times m$ matrix of correlations between the transformed variables and the variable approximations with variable names constructed from the value of the APREFIX= option and the names of the variables in the input data set

To create a data set WORK.A that contains all information, specify the following options in the PROC PRINQUAL statement:

```
proc prinqual scores approximations correlations out=a;
```

Also use a TRANSFORM statement appropriate for your data. Then the WORK.A data set contains the following:

D	T	S	A
$\mathbf{R_{TD}}$	$\mathbf{R_{TT}}$	$\mathbf{R_{TS}}$	$\mathbf{R_{TA}}$

To eliminate the bottom partitions that contain the correlations and component structure, do not specify the CORRELATIONS option. For example, use the following PROC PRINQUAL statement with an appropriate TRANSFORM statement:

```
proc prinqual scores approximations out=a;
```

Then the WORK.A data set contains the following:

D	T	S	A

Suppose you use the following PROC PRINQUAL statement (with an appropriate TRANSFORM statement):

```
proc prinqual out=a;
```

This creates a data set WORK.A of the following form:

```
D T
```

To output transformed data and component scores only, specify the following options in the PROC PRINQUAL statement:

```
proc prinqual replace scores out=a;
```

Then the WORK.A data set contains the following:

```
T S
```

TYPE and _NAME_ Variables

In addition to the preceding information, the output data set contains two character variables, the variable _TYPE_ (length 8) and the variable _NAME_ (length 32).

The _TYPE_ variable has the value 'SCORE' if the observation contains variables, transformed variables, components, or data approximations; the _TYPE_ variable has the value 'CORR' if the observation contains correlations or component structure.

By default, the _NAME_ variable has values 'ROW1', 'ROW2', and so on, for the observations with _TYPE_='SCORE'. If you use an ID statement, the variable _NAME_ contains the formatted ID variable for SCORES observations. The values of the variable _NAME_ for observations with _TYPE_='CORR' are the names of the transformed variables.

Certain procedures, such as PROC PRINCOMP, which can use the PROC PRINQUAL output data set, issue a warning that the PROC PRINQUAL data set contains _NAME_ and _TYPE_ variables but is not a TYPE=CORR data set. You can ignore this warning.

Variable Names

The TPREFIX=, APREFIX=, and PREFIX= options specify prefixes for the transformed and approximation variable names and for principal component score variables, respectively. PROC PRINQUAL constructs transformed and approximation variable names from a prefix and the first characters of the original variable name. The number of characters in the prefix plus the number of characters in the original variable name (including the final digits, if any) required to uniquely designate the new variables should not exceed 32. For example, if the APREFIX= parameter that you specify is one character, PROC PRINQUAL adds the first 31 characters of the original variable name; if your prefix is four characters, only the first 28 characters of the original variable name are added.

Effect of the TSTANDARD= and COVARIANCE Options

The values in the output data set are affected by the TSTANDARD= and COVARIANCE options. If you specify TSTANDARD=NOMISS, the NOMISS standardization is performed on the transformed data after the iterations have been completed, but before the output data set is created. The new means and variances are used in creating the output data set. Then, if you do not specify the COVARIANCE option, the data are transformed to mean zero and variance one. The principal component scores and data approximations are computed from the resulting matrix. The data are then linearly transformed to have the mean and variance specified by the TSTANDARD= option. The data approximations are transformed so that the means within each pair of a transformed variable and its approximation are the same. The ratio of the variance of a variable approximation to the variance of the corresponding transformed variable equals the proportion of the variance of the variable that is accounted for by the components model.

If you specify the COVARIANCE option and do not specify TSTANDARD=Z, you can input the transformed data to PROC PRINCOMP, again specifying the COVARIANCE option, to perform a components analysis of the results of PROC PRINQUAL. Similarly, if you do not specify the COVARIANCE option with PROC PRINQUAL and you input the transformed data to PROC PRINCOMP without the COVARIANCE option, you receive the same report. However, some combinations of PROC PRINQUAL options, such as COVARIANCE and TSTANDARD=Z, while valid, produce approximations and scores that cannot be reproduced by PROC PRINCOMP.

The component scores in the output data set are computed from the correlations among the transformed variables, or from the covariances if you specified the COVARIANCE option. The component scores are computed after the TSTANDARD=NOMISS transformation, if specified. The means of the component scores in the output data set are always zero. The variances equal the corresponding eigenvalues, unless you specify the STANDARD option; then the variances are set to one.

Avoiding Constant Transformations

There are times when the optimal scaling produces a constant transformed variable. This can happen with the MONOTONE, UNTIE, and MSPLINE transformations when the target is negatively correlated with the original input variable. It can happen with all transformations when the target is uncorrelated with the original input variable. When this happens, the procedure modifies the target to avoid a constant transformation. This strategy avoids certain nonoptimal solutions.

If the transformation is monotonic and a constant transformed variable results, the procedure multiplies the target by –1 and tries the optimal scaling again. If the transformation is not monotonic or if the multiplication by –1 did not help, the procedure tries using a random target. If the transformation is still constant, the previous nonconstant transformation is retained. When a constant transformation is avoided by any strategy, this message is displayed: “A constant transformation was avoided for *name*.”

Constant Variables

Constant and almost constant variables are zeroed and ignored.

Character OPSCORE Variables

Character OPSCORE variables are replaced by a numeric variable containing category numbers before the iterations, and the character values are discarded. Only the first eight characters are considered in determining category membership. If you want the original character variable in the output data set, give it a different name in the OPSCORE specification (OPSCORE(x / name=(x2)) and name the original variable in the ID statement (ID x;).

REITERATE Option Usage

You can use the REITERATE option to perform additional iterations when PROC PRINQUAL stops before the data have adequately converged. For example, suppose you execute the following code:

```
proc prinqual data=A cor out=B;
   transform mspline(X1-X5);
run;
```

If the transformations do not converge in the default 30 iterations, you can perform more iterations without repeating the first 30 iterations, as follows:

```
proc prinqual data=B reiterate cor out=B;
   transform mspline(X1-X5);
run;
```

Note that a WHERE statement is not necessary to exclude the correlation observations. They are automatically excluded because their _TYPE_ variable value is not 'SCORE'.

You can also use the REITERATE option to specify starting values other than the original values for the transformations. Providing alternate starting points might avoid local optima. Here are two examples.

```
proc prinqual data=A out=B;
   transform rank(X1-X5);
run;

proc prinqual data=B reiterate out=C;
   /* Use ranks as the starting point. */
   transform monotone(X1-X5);
run;
```

```
data B;
   set A;
   array TXS[5] TX1-TX5;
   do j = 1 to 5;
      TXS[j] = normal(0);
   end;
run;

proc prinqual data=B reiterate out=C;
   /* Use a random starting point. */
   transform monotone(X1-X5);
run;
```

Note that divergence with the REITERATE option, particularly in the second iteration, is not an error since the initial transformation is not required to be a valid member of the transformation family. When you specify the REITERATE option, the iteration does not terminate when the criterion change is negative during the first 10 iterations.

Passive Observations

Observations can be excluded from the analysis for several reasons, including zero weight, zero frequency, missing values in variables designated as IDENTITY, or missing values with the NOMISS option specified. These observations are passive in that they do not contribute to determining transformations, R square, total variance, and so on. However, some information can be computed for them, such as approximations, principal component scores, and transformed values. Passive observations in the output data set have a blank value for the variable _TYPE_.

Missing value estimates for passive observations might converge slowly with METHOD=MTV. In the following example, the missing value estimates should be 2, 5, and 8. Since the nonpassive observations do not change, the procedure converges in one iteration but the missing value estimates do not converge. The extra iterations produced by specifying CONVERGE=–1 and CCONVERGE=–1, as shown in the second PROC PRINQUAL step that follows, generate the expected results.

```
data A;
   input X Y;
   datalines;
1 1
2 .
3 3
4 4
5 .
6 6
7 7
8 .
9 9
;

proc prinqual nomiss data=A nomiss n=1 out=B method=mtv;
   transform lin(X Y);
run;
```

```
proc print;
run;

proc prinqual nomiss data=A nomiss n=1 out=B method=mtv
   converge=-1 cconverge=-1;
   transform lin(X Y);
run;

proc print;
run;
```

Computational Resources

This section provides information about the computational resources required to run PROC PRINQUAL.

Let

N = number of observations
m = number of variables
n = number of principal components
k = maximum spline degree
p = maximum number of knots

- For the MTV algorithm, more than

$$56m + 8Nm + 8\,(6N + (p + k + 2)(p + k + 11))$$

  bytes of array space are required.

- For the MGV and MAC algorithms, more than $56m$ plus the maximum of the data matrix size and the optimal scaling work space bytes of array space are required. The data matrix size is $8Nm$ bytes. The optimal scaling work space requires less than $8\,(6N + (p + k + 2)(p + k + 11))$ bytes.

- For the MTV and MGV algorithms, more than $56m + 4m(m + 1)$ bytes of array space are required.

- PROC PRINQUAL tries to store the original and transformed data in memory. If there is not enough memory, a utility data set is used, potentially resulting in a large increase in execution time. The amount of memory for the preceding data formulas is an underestimate of the amount of memory needed to handle most problems. These formulas give an absolute minimum amount of memory required. If a utility data set is used, and if memory could be used with perfect efficiency, then roughly the amount of memory stated previously would be needed. In reality, most problems require at least two or three times the minimum.

- PROC PRINQUAL sorts the data once. The sort time is roughly proportional to $mN^{3/2}$.
- For the MTV algorithm, the time required to compute the variable approximations is roughly proportional to $2Nm^2 + 5m^3 + nm^2$.
- For the MGV algorithm, one regression analysis per iteration is required to compute model parameter estimates. The time required to accumulate the crossproduct matrix is roughly proportional to Nm^2. The time required to compute the regression coefficients is roughly proportional to m^3. For each variable for each iteration, the swept crossproduct matrix is updated with time roughly proportional to m(N+m). The swept crossproduct matrix is updated for each variable with time roughly proportional to m^2, until computations are refreshed, requiring all sweeps to be performed again.
- The only computationally intensive part of the MAC algorithm is the optimal scaling, since variable approximations are simple averages.
- Each optimal scaling is a multiple regression problem, although some transformations are handled with faster special-case algorithms. The number of regressors for the optimal scaling problems depends on the original values of the variable and the type of transformation. For each monotone spline transformation, an unknown number of multiple regressions is required to find a set of coefficients that satisfies the constraints. The B-spline basis is generated twice for each SPLINE and MSPLINE transformation for each iteration. The time required to generate the B-spline basis is roughly proportional to Nk^2.

Displayed Output

The main output from PROC PRINQUAL is the output data set. However, the procedure does produce displayed output in the form of an iteration history table that includes the following:

- iteration number
- the criterion being optimized
- criterion change
- maximum and average absolute change in standardized variable scores computed over variables that can be iteratively transformed
- notes
- final convergence status

ODS Table Names

PROC PRINQUAL assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 70.4 along with the PROC statement options needed to produce the table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 70.4 ODS Tables Produced by PROC PRINQUAL

ODS Table Name	Description	Option
ConvergenceStatus	Convergence Status	default
Footnotes	Iteration History Footnotes	default
MAC	MAC Iteration History	METHOD=MAC
MGV	MGV Iteration History	METHOD=MGV
MTV	MTV Iteration History	METHOD=MTV

ODS Graphics

To request graphics with PROC PRINQUAL, you must first enable ODS Graphics by specifying the ODS GRAPHICS ON statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. The plots are produced only when you specify the options shown in the table. You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC PRINQUAL generates are listed in Table 70.5, along with the required statements and options.

Table 70.5 ODS Graphics Produced by PROC PRINQUAL

ODS Graph Name	Plot Description	Option
MDPrefPlot	Multidimensional preference analysis	MDPREF
TransformationPlot	Variable transformation	PLOTS=TRANSFORMATION

Examples: PRINQUAL Procedure

Example 70.1: Multidimensional Preference Analysis of Automobile Data

This example uses PROC PRINQUAL to perform a nonmetric multidimensional preference (MDPREF) analysis (Carroll 1972). MDPREF analysis is a principal component analysis of a data matrix with columns that correspond to people and rows that correspond to objects. The data are ratings or rankings of each person's preference for each object. The data are the transpose of the usual multivariate data matrix. (In other words, the columns are people; in the more typical matrix the rows represent people.) The final result of an MDPREF analysis is a biplot (Gabriel 1981) of the resulting preference space. A biplot displays the judges and objects in a single plot by projecting them onto the plane in the transformed variable space that accounts for the most variance.

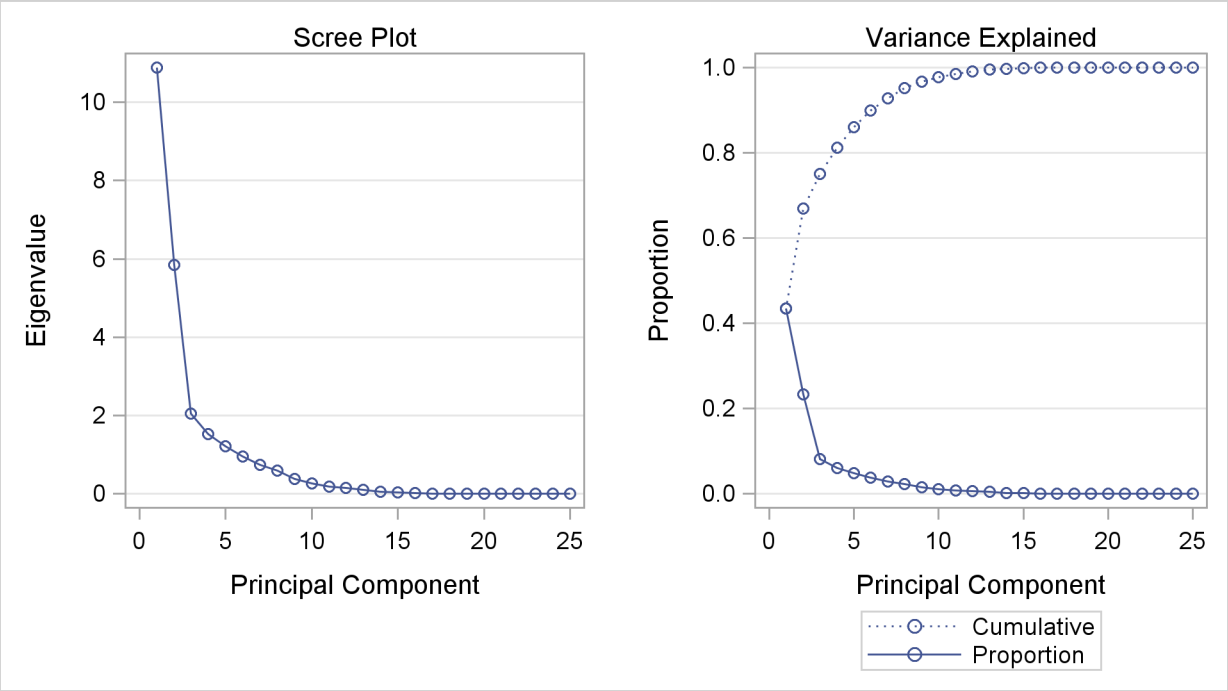
In 1980, 25 judges gave their preferences for each of 17 new automobiles. The ratings were made on a 0 to 9 scale, with 0 meaning very weak preference and 9 meaning very strong preference for the automobile. The following statements create a SAS data set with the manufacturer and model of each automobile along with the ratings:

```
title 'Preference Ratings for Automobiles Manufactured in 1980';

options validvarname=any;

data CarPref;
   input Make $ 1-10 Model $ 12-22 @25 ('1'n-'25'n) (1.);
   datalines;
Cadillac   Eldorado     800799049124050897109380 9
Chevrolet  Chevette     005120042345104300351569 8
Chevrolet  Citation     405330581416164354474779 5
Chevrolet  Malibu       602740072312134554566865 8
Ford       Fairmont     202400671502144353064865 5
Ford       Mustang      500719770502110185065755 6
Ford       Pinto        002100030303020150051407 8
Honda      Accord       595689760969995299897507 8
Honda      Civic        483670950748885256776507 5
Lincoln    Continental  700899059223040996209190 9
Plymouth   Gran Fury    700600043410110733345870 8
Plymouth   Horizon      300500563546130244467565 5
Plymouth   Volare       400500361402160275447655 5
Pontiac    Firebird     010789561320120695826590 7
Volkswagen Dasher       485869650887779537789500 0
Volkswagen Rabbit       485850970969579548788500 0
Volvo      DL           998999890999998798991900 0
;
```

The following statements run PROC PRINCOMP and create a scree plot. The results of this step are shown in Output 70.1.1.

```
ods graphics on;

* Principal Component Analysis of the Original Data;
proc princomp data=CarPref;
   ods select EigenvaluePlot;
   var '1'n-'25'n;
run;
```

Output 70.1.1 Eigenvalue Plot



The scree or eigenvalue plot in Output 70.1.1 shows that two principal components should be retained. There is a clear separation between the first two components and the remaining components. There are eight eigenvalues that are precisely zero because there are eight fewer observations than variables in the data matrix. One additional eigenvalue is zero, for a total of nine zero eigenvalues, since the correlation matrix is based on centered data. The following statements create the data set and perform a principal component analysis of the original data.

PROC PRINQUAL fits the nonmetric MDPREF model. PROC PRINQUAL monotonically transforms the raw judgments to maximize the proportion of variance accounted for by the first two principal components. The MONOTONE option is specified in the TRANSFORM statement to request a nonmetric MDPREF analysis; alternatively, you can instead specify the IDENTITY option for a metric analysis. Several options are used in the PROC PRINQUAL statement. The option DATA=CarPref specifies the input data set, OUT=Results creates an output data set, and N=2 and the default METHOD=MTV transform the data to better fit a two-component model. The REPLACE option replaces the original data with the monotonically transformed data in the OUT= data set. The MDPREF option standardizes the component scores to variance one so that the geometry of the biplot is correct, and it creates two variables in the OUT= data set named Prin1 and Prin2. These

variables contain the standardized principal component scores and structure matrix, which are used to make the biplot. If the variables in data matrix $\mathbf{X}$ are standardized to mean zero and variance one, and n is the number of rows in $\mathbf{X}$, then $\mathbf{X} = \mathbf{V}\mathbf{\Lambda}^{1/2}\mathbf{W}'$ is the principal component model, where $\mathbf{X}'\mathbf{X}/(n-1) = \mathbf{W}\mathbf{\Lambda}\mathbf{W}'$. The $\mathbf{W}$ and $\mathbf{\Lambda}$ contain the eigenvectors and eigenvalues of the correlation matrix of $\mathbf{X}$. The first two columns of $\mathbf{V}$, the standardized component scores, and $\mathbf{W}\mathbf{\Lambda}^{1/2}$, which is the structure matrix, are output. The advantage of creating a biplot based on principal components is that coordinates do not depend on the sample size. The following statements transform the data and produce Output 70.1.2.

```
* Transform the Data to Better Fit a Two Component Model;
proc prinqual data=CarPref out=Results n=2 replace mdpref;
   title2 'Multidimensional Preference (MDPREF) Analysis';
   title3 'Optimal Monotonic Transformation of Preference Data';
   id model;
   transform monotone('1'n-'25'n);
run;
```

Output 70.1.2 PRINQUAL Iteration History

Preference Ratings for Automobiles Manufactured in 1980
Multidimensional Preference (MDPREF) Analysis
Optimal Monotonic Transformation of Preference Data

The PRINQUAL Procedure

PRINQUAL MTV Algorithm Iteration History

Iteration Number	Average Change	Maximum Change	Proportion of Variance	Criterion Change	Note
1	0.24994	1.28017	0.66946		
2	0.07223	0.36958	0.80194	0.13249	
3	0.04522	0.29026	0.81598	0.01404	
4	0.03096	0.25213	0.82178	0.00580	
5	0.02182	0.23045	0.82493	0.00315	
6	0.01602	0.19017	0.82680	0.00187	
7	0.01219	0.14748	0.82793	0.00113	
8	0.00953	0.11031	0.82861	0.00068	
9	0.00737	0.06461	0.82904	0.00043	
10	0.00556	0.04469	0.82930	0.00026	
11	0.00445	0.04087	0.82944	0.00014	
12	0.00381	0.03706	0.82955	0.00011	
13	0.00319	0.03348	0.82965	0.00009	
14	0.00255	0.02999	0.82971	0.00006	
15	0.00213	0.02824	0.82976	0.00005	
16	0.00183	0.02646	0.82980	0.00004	
17	0.00159	0.02472	0.82983	0.00003	
18	0.00139	0.02305	0.82985	0.00003	
19	0.00123	0.02145	0.82988	0.00002	
20	0.00109	0.01993	0.82989	0.00002	
21	0.00096	0.01850	0.82991	0.00001	
22	0.00086	0.01715	0.82992	0.00001	
23	0.00076	0.01588	0.82993	0.00001	
24	0.00067	0.01440	0.82994	0.00001	
25	0.00059	0.00871	0.82994	0.00001	
26	0.00050	0.00720	0.82995	0.00000	
27	0.00043	0.00642	0.82995	0.00000	
28	0.00037	0.00573	0.82995	0.00000	
29	0.00031	0.00510	0.82995	0.00000	
30	0.00027	0.00454	0.82995	0.00000	Not Converged

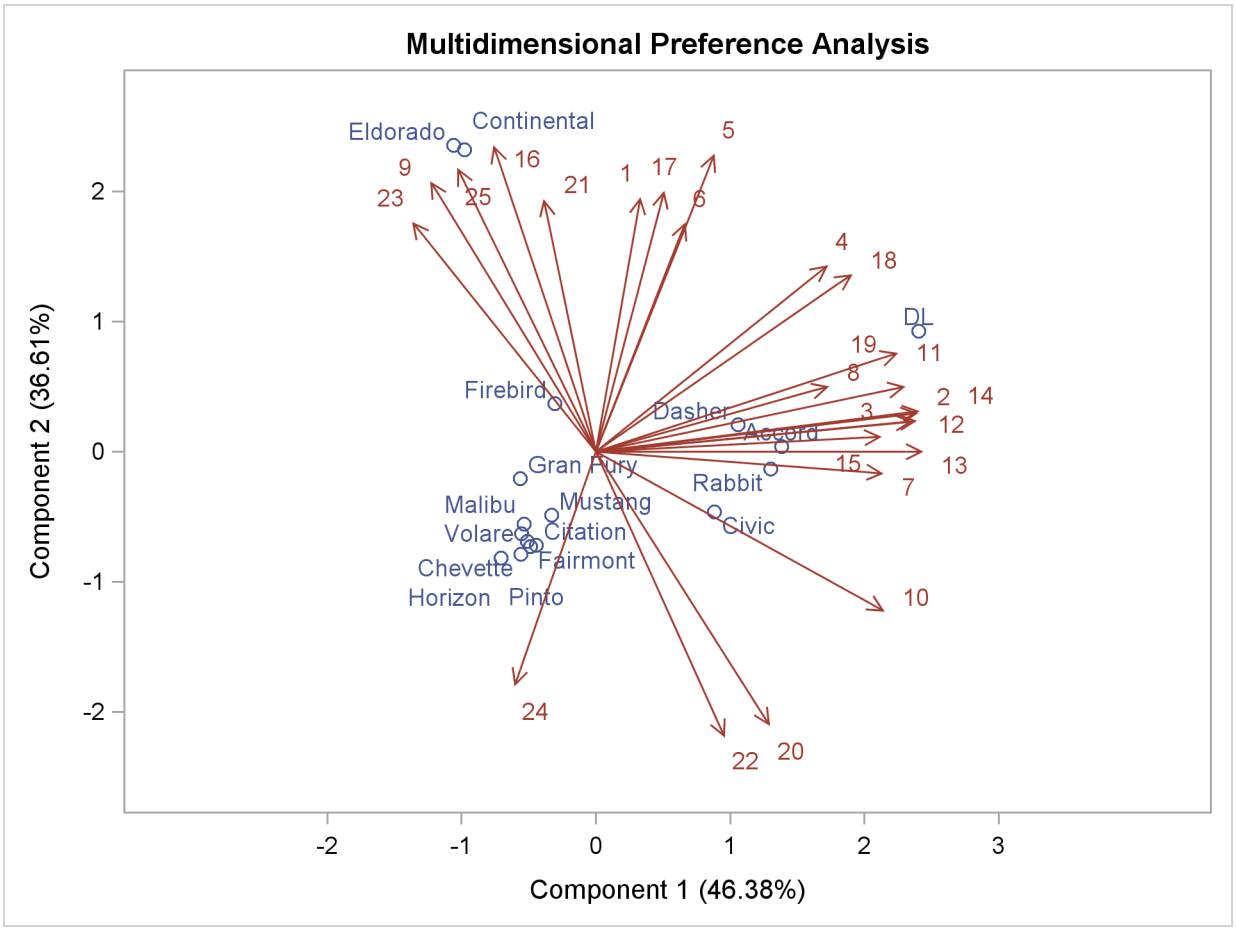
WARNING: Failed to converge, however criterion change is less than 0.0001.

The iteration history displayed by PROC PRINQUAL indicates that the proportion of variance is increased from an initial 0.66946 to 0.82995. The proportion of variance accounted for by PROC PRINQUAL on the first iteration equals the cumulative proportion of variance shown by PROC PRINCOMP for the first two principal components. PROC PRINQUAL's initial iteration performs a standard principal component analysis of the raw data. The columns labeled Average Change, Maximum Change, and Criterion Change contain values that always decrease, indicating that PROC PRINQUAL is improving the transformations at a monotonically decreasing rate over the iterations. This does not always happen, and when it does not, it suggests that the analysis might be converging

to a degenerate solution. See Example 70.2 for a discussion of a degenerate solution. The algorithm does not converge in 30 iterations. However, the criterion change is small, indicating that more iterations are unlikely to have much effect on the results.

The biplot, shown in Output 70.1.3, is automatically displayed by PROC PRINQUAL when ODS Graphics is enabled and the MDPREF option is specified.

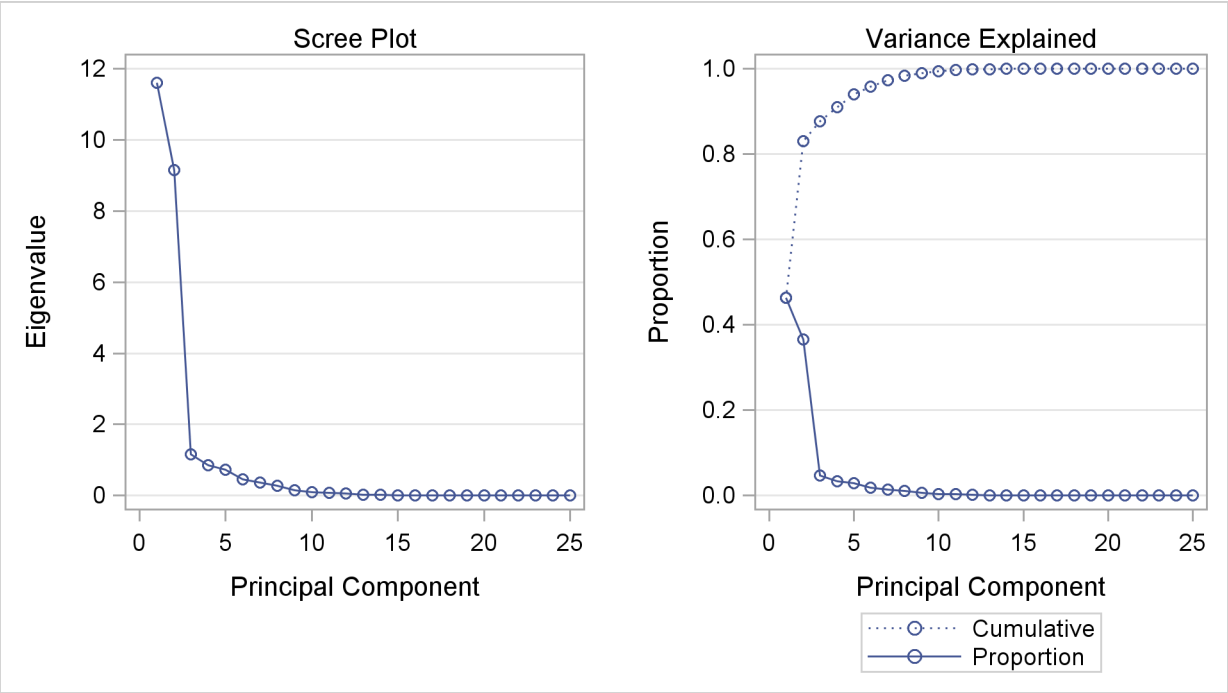
Output 70.1.3 Biplot Made with PRINQUAL



The second PROC PRINCOMP analysis is performed on the transformed data. The WHERE statement is used to retain only the monotonically transformed judgments. The scree plot shows that the first two eigenvalues are now much larger than the remaining smaller eigenvalues. The second eigenvalue has increased markedly at the expense of the next several eigenvalues. Two principal components seem to be necessary and sufficient to adequately describe these judges' preferences for these automobiles. The cumulative proportion of variance displayed by PROC PRINCOMP for the first two principal components is 0.83. The following statements perform the analysis and produce Output 70.1.4:

```
* Final Principal Component Analysis;
proc princomp data=Results;
   ods select EigenvaluePlot;
   var '1'n-'25'n;
   where _TYPE_='SCORE';
run;

ods graphics off;
```

Output 70.1.4 Transformed Data Eigenvalue Plot



The remainder of the example discusses the MDPREF biplot. A biplot is a plot that displays the relation between the row points and the columns of a data matrix. The rows of $\mathbf{V}$, the standardized component scores, and $\mathbf{W}\mathbf{\Lambda}^{1/2}$, the structure matrix, contain enough information to reproduce $\mathbf{X}$. The (i, j) element of $\mathbf{X}$ is the product of row i of $\mathbf{V}$ and row j of $\mathbf{W}\mathbf{\Lambda}^{1/2}$. If all but the first two columns of $\mathbf{V}$ and $\mathbf{W}\mathbf{\Lambda}^{1/2}$ are discarded, the (i, j) element of $\mathbf{X}$ is approximated by the product of row i of $\mathbf{V}$ and row j of $\mathbf{W}\mathbf{\Lambda}^{1/2}$.

Since the MDPREF analysis is based on a principal component model, the dimensions of the MDPREF biplot are the first two principal components. The first principal component is the longest dimension through the MDPREF biplot. The first principal component is overall preference, which is the most salient dimension in the preference judgments. One end points in the direction that is on the average preferred most by the judges, and the other end points in the least preferred direction. The second principal component is orthogonal to the first principal component, and it is the orthogonal direction that is the second most salient. The interpretation of the second dimension varies from example to example.

With an MDPREF biplot, it is geometrically appropriate to represent each automobile (object) by a point and each judge by a vector. The automobile points have coordinates that are the scores of the automobile on the first two principal components. The judge vectors emanate from the origin of the space and go through a point whose coordinates are the coefficients of the judge (variable) on the first two principal components.

The absolute length of a vector is arbitrary. However, the relative lengths of the vectors indicate fit, with the squared lengths being proportional to the communalities that you can get in PROC FACTOR output. The direction of the vector indicates the direction that is most preferred by the individual judge, with preference increasing as the vector moves from the origin. Let $\mathbf{v}'$ be row i of $\mathbf{V}$, $\mathbf{u}'$ be row j of $\mathbf{U} = \mathbf{W}\mathbf{\Lambda}^{1/2}$, $\|\mathbf{v}\|$ be the length of $\mathbf{v}$, $\|\mathbf{u}\|$ be the length of $\mathbf{u}$, and θ be the angle between $\mathbf{v}$ and $\mathbf{u}$. The predicted degree of preference that an individual judge has for an automobile is $\mathbf{u}'\mathbf{v} = \|\mathbf{u}\|\ \|\mathbf{v}\| \cos\theta$. Each automobile point can be orthogonally projected onto the vector. The projection of automobile i on vector j is $\mathbf{u}((\mathbf{u}'\mathbf{v})/(\mathbf{u}'\mathbf{u}))$, and the length of this projection is $\|\mathbf{v}\| \cos\theta$. The automobile that projects farthest along a vector in the direction it points is that judge's most preferred automobile, since the length of this projection, $\|\mathbf{v}\| \cos\theta$, differs from the predicted preference, $\|\mathbf{u}\|\ \|\mathbf{v}\| \cos\theta$, only by $\|\mathbf{u}\|$, which is constant for each judge.

To interpret the biplot, look for directions through the plot that show a continuous change in some attribute of the automobiles, or look for regions in the plot that contain clusters of automobile points and determine what attributes the automobiles have in common. Points that are tightly clustered in a region of the plot represent automobiles that have the same preference patterns across the judges. Vectors that point in roughly the same direction represent judges who have similar preference patterns.

In the biplot, American automobiles are located at the left of the space, while European and Japanese automobiles are located at the right. At the top of the space are expensive American automobiles (Cadillac Eldorado, Lincoln Continental), while at the bottom are inexpensive ones (Ford Pinto, Chevrolet Chevette). The first principal component differentiates American from imported automobiles, and the second arranges automobiles by price and other associated characteristics.

The two expensive American automobiles form a cluster, the sporty automobile (Pontiac Firebird) is by itself, the Volvo DL is by itself, and the remaining imported autos form a cluster, as do the remaining American autos. It seems there are 5 prototypical automobiles in this set of 17, in terms of preference patterns among the 25 judges.

Most of the judges prefer the imported automobiles, especially the Volvo. There is also a fairly large minority that prefer the expensive autos, whether or not they are American (those with vectors that point toward one o'clock), or simply prefer expensive American automobiles (vectors that point toward eleven o'clock). There are two judges who prefer anything except expensive American autos (five o'clock vectors), and one who prefers inexpensive American autos (seven o'clock vector).

Several vectors point toward the upper-right corner of the plot, toward a region with no automobiles. This is the region between the European and Japanese autos at the right and the luxury autos at the top. This suggests that there is a market for luxury Japanese and European automobiles.

Example 70.2: Principal Components of Basketball Rankings

The data in this example are 1985–1986 preseason rankings of 35 U.S. college basketball teams by 10 different news services. The services do not all rank the same teams or the same number of teams, so there are missing values in these data. Each of the 35 teams in the data set is ranked by at least one news service. One way of summarizing these data is with a principal component analysis, since the rankings should all be related to a single underlying variable, the first principal component.

You can use PROC PRINQUAL to estimate the missing ranks and compute scores for all observations. You can formulate a PROC PRINQUAL analysis that assumes that the observed ranks are ordinal variables and replaces the ranks with new numbers that are monotonic with the ranks and better fit the one principal component model. The missing rank estimates need to be constrained since a news service would have positioned the unranked teams below the teams it ranked. PROC PRINQUAL should impose order constraints within the nonmissing values and between the missing and nonmissing values, but not within the missing values. PROC PRINQUAL has sophisticated missing data handling facilities; however, these facilities cannot directly handle this problem. The solution requires reformulating the problem.

By performing some preliminary data manipulations, specifying the N=1 option in the PROC PRINQUAL statement, and specifying the UNTIE transformation in the TRANSFORM statement, you can make the missing value estimates conform to the requirements. The PROC MEANS step finds the largest rank for each variable. The next DATA step replaces missing values with a value that is one larger than the largest observed rank. The PROC PRINQUAL N=1 option specifies that the variables should be transformed to make them as one-dimensional as possible. The UNTIE transformation in the TRANSFORM statement monotonically transforms the ranks, untying any ties in an optimal way. Because the only ties are for the values that replace the missing values, and because these values are larger than the observed values, the rescoring of the data satisfies the preceding requirements.

The following statements create the data set and perform the transformations discussed previously. These statements produce Output 70.2.1 and Output 70.2.2.

```
*  Preseason 1985 College Basketball Rankings
*  (rankings of 35 teams by 10 news services)
*
*  Note:(a) Various news services rank varying numbers of teams.
*       (b) Not all 35 teams are ranked by all news services.
*       (c) Each team is ranked by at least one service.
*       (d) Rank 20 is missing for UPI.;

title1 '1985 Preseason College Basketball Rankings';

data bballm;
   input School $13. CSN DurhamSun DurhamHerald WashingtonPost
         USA_Today SportMagazine InsideSports UPI AP SportsIllustrated;
   label CSN               = 'Community Sports News (Chapel Hill, NC)'
         DurhamSun         = 'Durham Sun'
         DurhamHerald      = 'Durham Morning Herald'
         WashingtonPost    = 'Washington Post'
         USA_Today         = 'USA Today'
         SportMagazine     = 'Sport Magazine'
         InsideSports      = 'Inside Sports'
         UPI               = 'United Press International'
         AP                = 'Associated Press'
         SportsIllustrated = 'Sports Illustrated'
         ;
   format CSN--SportsIllustrated 5.1;
   datalines;
Louisville     1  8  1  9  8  9  6 10  9  9
Georgia Tech   2  2  4  3  1  1  1  2  1  1
Kansas         3  4  5  1  5 11  8  4  5  7
Michigan       4  5  9  4  2  5  3  1  3  2
Duke           5  6  7  5  4 10  4  5  6  5
UNC            6  1  2  2  3  4  2  3  2  3
Syracuse       7 10  6 11  6  6  5  6  4 10
Notre Dame     8 14 15 13 11 20 18 13 12  .
Kentucky       9 15 16 14 14 19 11 12 11 13
LSU           10  9 13  . 13 15 16  9 14  8
DePaul        11  . 21 15 20  . 19  .  . 19
Georgetown    12  7  8  6  9  2  9  8  8  4
Navy          13 20 23 10 18 13 15  . 20  .
Illinois      14  3  3  7  7  3 10  7  7  6
Iowa          15 16  .  . 23  .  . 14  . 20
Arkansas      16  .  .  . 25  .  .  .  . 16
Memphis State 17  . 11  . 16  8 20  . 15 12
Washington    18  .  .  .  .  .  . 17  .  .
UAB           19 13 10  . 12 17  . 16 16 15
UNLV          20 18 18 19 22  . 14 18 18  .
NC State      21 17 14 16 15  . 12 15 17 18
Maryland      22  .  .  . 19  .  .  . 19 14
Pittsburgh    23  .  .  .  .  .  .  .  .  .
Oklahoma      24 19 17 17 17 12 17  . 13 17
Indiana       25 12 20 18 21  .  .  .  .  .
Virginia      26  . 22  .  . 18  .  .  .  .
Old Dominion  27  .  .  .  .  .  .  .  .  .
Auburn        28 11 12  8 10  7  7 11 10 11
```

```
St. Johns     29  .  .  .  . 14  .  .  .  .
UCLA          30  .  .  .  .  .  . 19  .  .
St. Joseph's   .  . 19  .  .  .  .  .  .  .
Tennessee      .  . 24  .  . 16  .  .  .  .
Montana        .  .  . 20  .  .  .  .  .  .
Houston        .  .  .  . 24  .  .  .  .  .
Virginia Tech  .  .  .  .  .  . 13  .  .  .
;
* Find maximum rank for each news service and replace
* each missing value with the next highest rank.;

proc means data=bballm noprint;
   output out=maxrank
      max=mcsn mdurs mdurh mwas musa mspom mins mupi map mspoi;
run;

data bball;
   set bballm;
   if _n_=1 then set maxrank;
   array services[10] CSN--SportsIllustrated;
   array maxranks[10] mcsn--mspoi;
   keep School CSN--SportsIllustrated;
   do i=1 to 10;
      if services[i]=. then services[i]=maxranks[i]+1;
      end;
run;

* Assume that the ranks are ordinal and that unranked teams
* would have been ranked lower than ranked teams.  Monotonically
* transform all ranked teams while estimating the unranked teams.
* Enforce the constraint that the missing ranks are estimated to
* be less than the observed ranks.  Order the unranked teams
* optimally within this constraint.  Do this so as to maximize
* the variance accounted for by one linear combination.  This
* makes the data as nearly rank one as possible, given the
* constraints.
*
* NOTE: The UNTIE transformation should be used with caution.
* It frequently produces degenerate results.;

ods graphics on;

proc prinqual data=bball out=tbball scores n=1 tstandard=z
   plots=transformations;
   title2 'Optimal Monotonic Transformation of Ranked Teams';
   title3 'with Constrained Estimation of Unranked Teams';
   transform untie(CSN -- SportsIllustrated);
   id School;
run;
```

Output 70.2.1 PRINQUAL Iteration History

1985 Preseason College Basketball Rankings
Optimal Monotonic Transformation of Ranked Teams
with Constrained Estimation of Unranked Teams

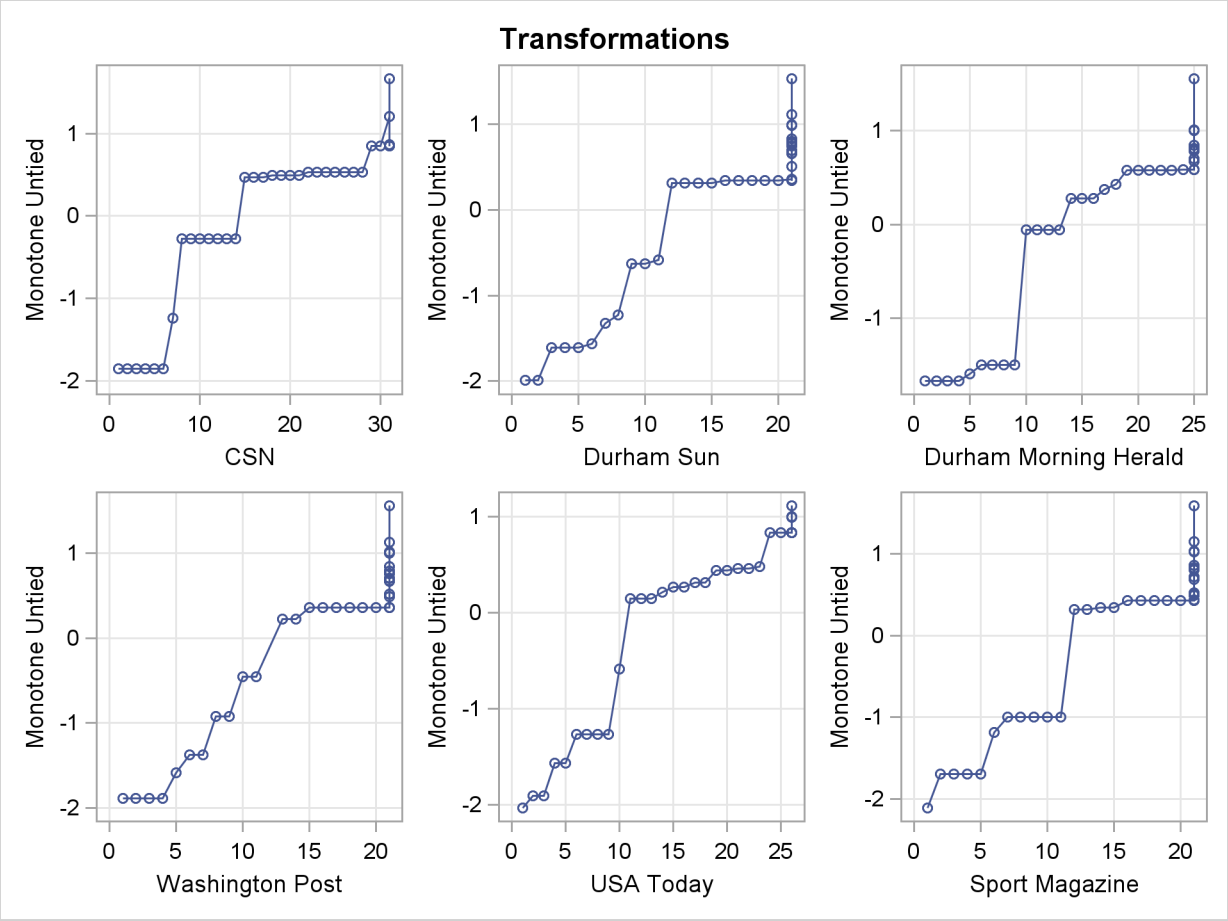
The PRINQUAL Procedure

PRINQUAL MTV Algorithm Iteration History

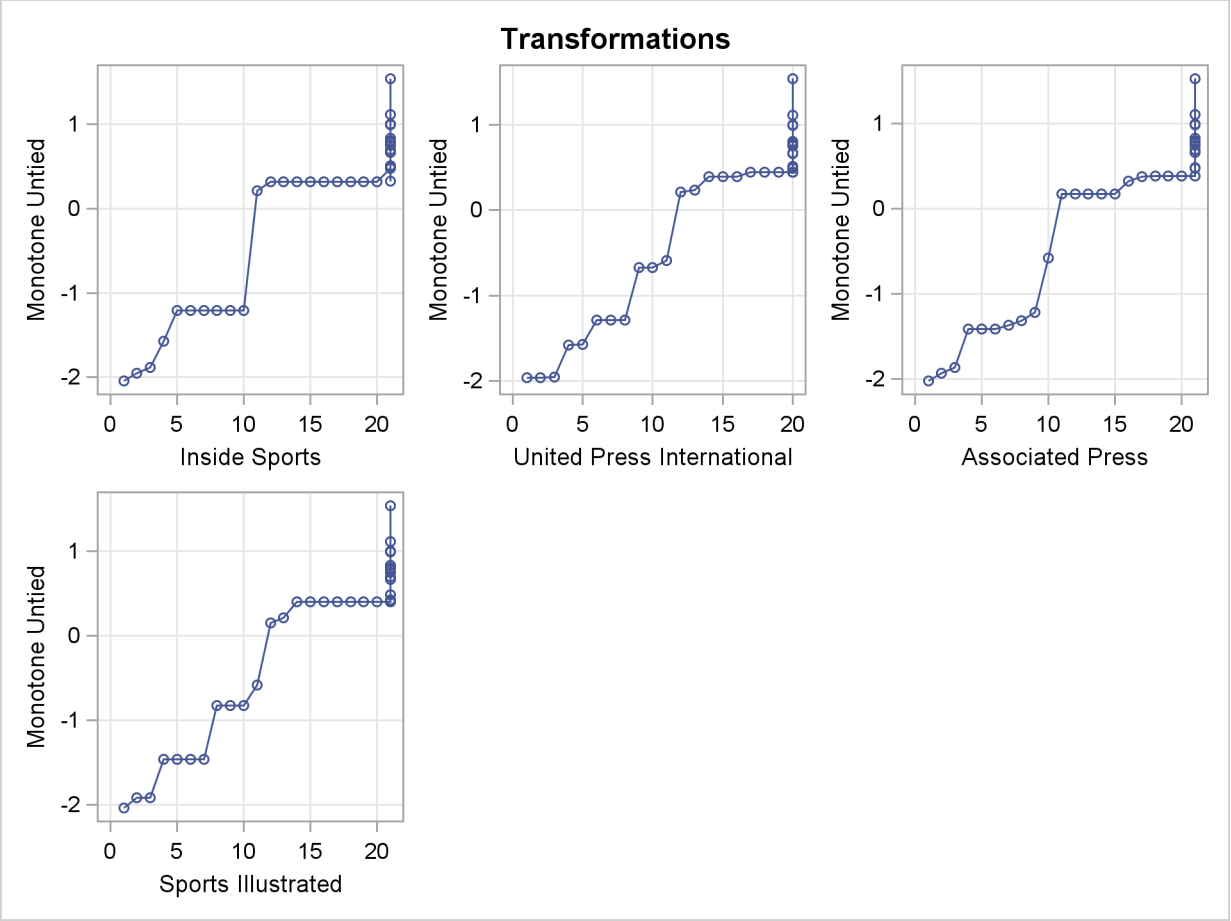
Iteration Number	Average Change	Maximum Change	Proportion of Variance	Criterion Change	Note
1	0.18563	0.76531	0.85850		
2	0.03225	0.14627	0.94362	0.08512	
3	0.02126	0.10530	0.94669	0.00307	
4	0.01467	0.07526	0.94801	0.00132	
5	0.01067	0.05282	0.94865	0.00064	
6	0.00800	0.03669	0.94899	0.00034	
7	0.00617	0.02862	0.94919	0.00020	
8	0.00486	0.02636	0.94932	0.00013	
9	0.00395	0.02453	0.94941	0.00009	
10	0.00327	0.02300	0.94947	0.00006	
11	0.00275	0.02166	0.94952	0.00005	
12	0.00236	0.02041	0.94956	0.00004	
13	0.00205	0.01927	0.94959	0.00003	
14	0.00181	0.01818	0.94962	0.00003	
15	0.00162	0.01719	0.94964	0.00002	
16	0.00147	0.01629	0.94966	0.00002	
17	0.00136	0.01546	0.94968	0.00002	
18	0.00128	0.01469	0.94970	0.00002	
19	0.00121	0.01398	0.94971	0.00001	
20	0.00115	0.01332	0.94973	0.00001	
21	0.00111	0.01271	0.94974	0.00001	
22	0.00105	0.01213	0.94975	0.00001	
23	0.00099	0.01155	0.94976	0.00001	
24	0.00095	0.01095	0.94977	0.00001	
25	0.00091	0.01038	0.94978	0.00001	
26	0.00088	0.00986	0.94978	0.00001	
27	0.00084	0.00936	0.94979	0.00001	
28	0.00081	0.00889	0.94980	0.00001	
29	0.00077	0.00846	0.94980	0.00000	
30	0.00073	0.00805	0.94980	0.00000	Not Converged

WARNING: Failed to converge, however criterion change is less than 0.0001.

Output 70.2.2 Transformations



Output 70.2.2 *continued*



An alternative approach is to use the pairwise deletion option of the CORR procedure to compute the correlation matrix and then use PROC PRINCOMP or PROC FACTOR to perform the principal component analysis. This approach has several disadvantages. The correlation matrix might not be positive semidefinite (psd), an assumption required for principal component analysis. PROC PRINQUAL always produces a psd correlation matrix. Even with pairwise deletion, PROC CORR removes the six observations that have only a single nonmissing value from this data set. Finally, it is still not possible to calculate scores on the principal components for those teams that have missing values.

You can compute the composite ranking by using PROC PRINCOMP and some preliminary data manipulations, similar to those discussed previously.

Chapter 69, "The PRINCOMP Procedure," contains an example where the average of the unused ranks in each poll is substituted for the missing values, and each observation is weighted by the number of nonmissing values. This method has much to recommend it. It is much faster and simpler than using PROC PRINQUAL. It is also much less prone to degeneracies and capitalization on chance. However, PROC PRINCOMP does not allow the nonmissing ranks to be monotonically transformed and the missing values untied to optimize fit.

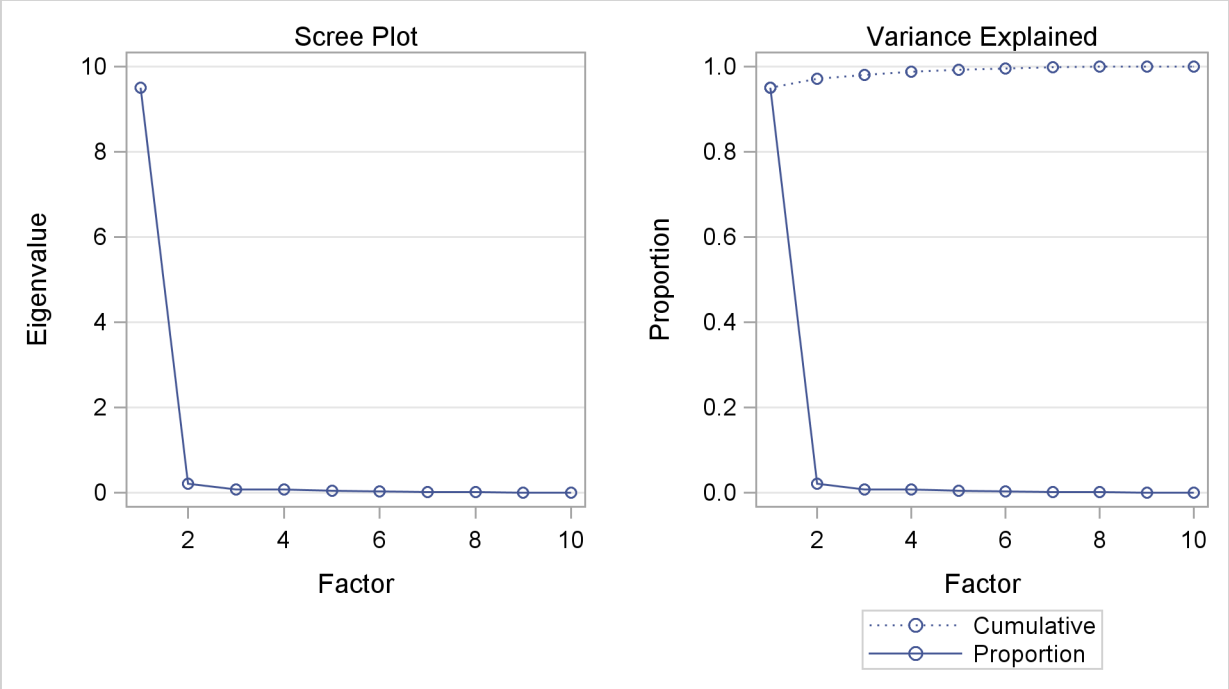
PROC PRINQUAL monotonically transforms the observed ranks and estimates the missing ranks (within the constraints given previously) to account for almost 95 percent of the variance of the transformed data by just one dimension. PROC FACTOR is then used to report details of the principal component analysis of the transformed data. As shown by the Factor Pattern values in Output 70.2.3, nine of the ten news services have a correlation of 0.95 or larger with the scores on the first principal component after the data are optimally transformed. The scores are sorted and the composite ranking is displayed following the PROC FACTOR output. More confidence can be placed in the stability of the scores for teams that are ranked by the majority of the news services than in scores for teams that are seldom ranked.

The monotonic transformations are plotted for each of the ten news services. See Output 70.2.2. These plots show the values of the raw ranks (with the missing ranks replaced by the maximum rank plus one) versus the rescored (transformed) ranks. The transformations are the step functions that maximize the fit of the data to the principal component model. Smoother transformations could be found by using MSPLINE transformations, but MSPLINE transformations would not correctly handle the missing data problem.

The following statements perform the final analysis and produce Output 70.2.3:

```
* Perform the Final Principal Component Analysis;
proc factor nfactors=1 plots=scree;
   title4 'Principal Component Analysis';
   ods select factorpattern screeplot;
   var TCSN -- TSportsIllustrated;
run;

proc sort;
   by Prin1;
run;

* Display Scores on the First Principal Component;
proc print;
   title4 'Teams Ordered by Scores on First Principal Component';
   var School Prin1;
run;

ods graphics off;
```

Output 70.2.3 Principal Components of College Basketball Rankings



Output 70.2.3 *continued*

Factor Pattern

		Factor1
TCSN	CSN Transformation	0.91136
TDurhamSun	DurhamSun Transformation	0.98887
TDurhamHerald	DurhamHerald Transformation	0.97402
TWashingtonPost	WashingtonPost Transformation	0.97408
TUSA_Today	USA_Today Transformation	0.98867
TSportMagazine	SportMagazine Transformation	0.95331
TInsideSports	InsideSports Transformation	0.98521
TUPI	UPI Transformation	0.98534
TAP	AP Transformation	0.99590
TSportsIllustrated	SportsIllustrated Transformation	0.98615

Output 70.2.3 *continued*

1985 Preseason College Basketball Rankings
Optimal Monotonic Transformation of Ranked Teams
with Constrained Estimation of Unranked Teams
Teams Ordered by Scores on First Principal Component

Obs	School	Prin1
1	Georgia Tech	-6.20315
2	UNC	-5.93314
3	Michigan	-5.71034
4	Kansas	-4.78699
5	Duke	-4.75896
6	Illinois	-4.19220
7	Georgetown	-4.02861
8	Louisville	-3.73087
9	Syracuse	-3.47497
10	Auburn	-1.78429
11	LSU	-0.35928
12	Memphis State	0.46737
13	Kentucky	0.63661
14	Notre Dame	0.71919
15	Navy	0.76187
16	UAB	0.98316
17	DePaul	1.09891
18	Oklahoma	1.12012
19	NC State	1.15144
20	UNLV	1.28766
21	Iowa	1.45260
22	Indiana	1.48123
23	Maryland	1.54935
24	Virginia	2.01385
25	Arkansas	2.02718
26	Washington	2.10878
27	Tennessee	2.27770
28	Virginia Tech	2.36103
29	St. Johns	2.37387
30	Montana	2.43502
31	UCLA	2.52481
32	Pittsburgh	3.00907
33	Old Dominion	3.03324
34	St. Joseph's	3.39259
35	Houston	4.69614

The ordinary PROC PRINQUAL missing data handling facilities do not work for these data because they do not constrain the missing data estimates properly. If you code the missing ranks as missing and specify linear transformations, then you can compute least-squares estimates of the missing values without transforming the observed values. The first principal component then accounts for 92 percent of the variance after 20 iterations. However, Virginia Tech is ranked number 11 by its score even though it appeared in only one poll (Inside Sports ranked it number 13, anchoring it firmly in the middle). Specifying monotone transformations is also inappropriate since they too allow unranked teams to move in between ranked teams.

With these data, the combination of monotone transformations and the freedom to score the missing ranks without constraint leads to degenerate transformations. PROC PRINQUAL tries to merge the 35 points into two points, producing a perfect fit in one dimension. There is evidence for this after 20 iterations when the Average Change, Maximum Change, and Criterion Change values are all increasing, instead of the more stable decreasing change rate seen in the analysis shown. The change rates all stop increasing after 41 iterations, and it is clear by 70 or 80 iterations that one component will account for 100 percent of the transformed variables variance after sufficient iteration. While this might seem desirable (after all, it is a perfect fit), you should, in fact, be on guard when this happens. Whenever convergence is slow, the rates of change increase, or the final data perfectly fit the model, the solution is probably degenerating because of too few constraints on the scorings.

PROC PRINQUAL can account for 100 percent of the variance by scoring Montana and UCLA with one positive value on all variables and scoring all the other teams with one negative value on all variables. This inappropriate analysis suggests that all ranked teams are equally good except for two teams that are less good. Both of these two teams are ranked by only one news service, and their only nonmissing rank is last in the poll. This accounts for the degeneracy.

References

Carroll, J. D. (1972), "Individual Differences and Multidimensional Scaling," in R. N. Shepard, A. K. Romney, and S. B. Nerlove, eds., *Multidimensional Scaling: Theory and Applications in the Behavioral Sciences (Volume 1)*, New York: Seminar Press.

de Boor, C. (1978), *A Practical Guide to Splines*, New York: Springer Verlag.

de Leeuw, J. (1985), personal communication, Leiden.

de Leeuw, J. (1986), *Regression with Optimal Scaling of the Dependent Variable*, Leiden: Department of Data Theory, University of Leiden.

Eckart, C. and Young, G. (1936), "The Approximation of One Matrix by Another of Lower Rank," *Psychometrika*, 1, 211–218.

Fisher, R. A. (1938), *Statistical Methods for Research Workers*, Tenth Edition, Edinburgh: Oliver & Boyd.

Gabriel, K. R. (1981), "Biplot Display of Multivariate Matrices for Inspection of Data and Diagnosis," in V. Barnett, ed., *Interpreting Multivariate Data*, London: John Wiley & Sons.

Gifi, A. (1990), *Nonlinear Multivariate Analysis*, New York: John Wiley & Sons.

Goodnight, J. H. (1978), *The SWEEP Operator: Its Importance in Statistical Computing*, Technical Report R-106, SAS Institute Inc, Cary, NC.

Hotelling, H. (1933), "Analysis of a Complex of Statistical Variables into Principal Components," *Journal of Educational Psychology*, 24, 417–441, 498–520.

Kruskal, J. B. (1964), "Nonmetric Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis," *Psychometrika*, 29, 1–27.

Kruskal, J. B. and Shepard, R. N. (1974), "A Nonmetric Variety of Linear Factor Analysis," *Psychometrika*, 38, 123–157.

Sarle, W. S. (1984), personal communication, Cary, NC.

Siegel, S. (1956), *Nonparametric Statistics*, New York: McGraw-Hill.

Smith, P. L. (1979), "Splines as a Useful and Convenient Statistical Tool," *The American Statistician*, 33, 57–62.

Tenenhaus, M. and Vachette, J. L. (1977), "PRINQUAL: Un Programme d'Analyse en Composantes Principales d'un Ensemble de Variables Nominales ou Numeriques," *Les Cahiers de Recherche*, 68, CESA, Jout-en-Josas, France.

van Rijckevorsel, J. (1982), "Canonical Analysis with B-Splines," in H. Caussinus, P. Ettinger, and R. Tomassone, eds., *COMPUSTAT 1982, Part I*, Vienna: Physica Verlag.

Winsberg, S. and Ramsay, J. O. (1983), "Monotone Spline Transformations for Dimension Reduction," *Psychometrika*, 48, 575–595.

Young, F. W. (1981), "Quantitative Analysis of Qualitative Data," *Psychometrika*, 46, 357–388.

Young, F. W., Takane, Y., and de Leeuw, J. (1978), "The Principal Components of Mixed Measurement Level Multivariate Data: An Alternating Least Squares Method with Optimal Scaling Features," *Psychometrika*, 43, 279–281.

Chapter 71

The PROBIT Procedure

Contents

Overview: PROBIT Procedure **5258**
Getting Started: PROBIT Procedure **5259**
Estimating the Natural Response Threshold Parameter 5259
Syntax: PROBIT Procedure **5263**
PROC PROBIT Statement 5264
BY Statement 5268
CDFPLOT Statement 5268
CLASS Statement 5277
INSET Statement 5277
IPPPLOT Statement 5279
LPREDPLOT Statement 5287
MODEL Statement 5295
OUTPUT Statement 5299
PREDPPLOT Statement 5300
WEIGHT Statement 5309
Details: PROBIT Procedure **5309**
Missing Values 5309
Response Level Ordering 5309
Computational Method 5310
Distributions 5311
INEST= *SAS-data-set* 5312
Model Specification 5313
Lack-of-Fit Tests 5313
Rescaling the Covariance Matrix 5314
Tolerance Distribution 5315
Inverse Confidence Limits 5315
OUTEST= *SAS-data-set* 5316
XDATA= *SAS-data-set* 5317
Traditional High-Resolution Graphics 5318
Displayed Output 5323
ODS Table Names 5324
ODS Graphics 5325
Examples: PROBIT Procedure **5329**
Example 71.1: Dosage Levels 5329

Example 71.2: Multilevel Response 5337
Example 71.3: Logistic Regression 5343
Example 71.4: An Epidemiology Study 5345
References **5356**

Overview: PROBIT Procedure

The PROBIT procedure calculates maximum likelihood estimates of regression parameters and the natural (or threshold) response rate for quantal response data from biological assays or other discrete event data. This includes probit, logit, ordinal logistic, and extreme value (or gompit) regression models.

Probit analysis developed from the need to analyze qualitative (dichotomous or polytomous) dependent variables within the regression framework. Many response variables are binary by nature (yes/no), while others are measured ordinally rather than continuously (degree of severity). Collett (1991) and Agresti (1990), for example, have shown ordinary least squares (OLS) regression to be inadequate when the dependent variable is discrete. Probit or logit analyses are more appropriate in this case.

The PROBIT procedure computes maximum likelihood estimates of the parameters β and C of the probit equation by using a modified Newton-Raphson algorithm. When the response Y is binary, with values 0 and 1, the probit equation is

$$p = \Pr(Y = 0) = C + (1 - C)F(\mathbf{x}'\beta)$$

where

β is a vector of parameter estimates

F is a cumulative distribution function (normal, logistic, or extreme value)

$\mathbf{x}$ is a vector of explanatory variables

p is the probability of a response

C is the natural (threshold) response rate

Notice that PROC PROBIT, by default, models the probability of the *lower* response levels. The choice of the distribution function F (normal for the probit model, logistic for the logit model, and extreme value or Gompertz for the gompit model) determines the type of analysis. For most problems, there is relatively little difference between the normal and logistic specifications of the model. Both distributions are symmetric about the value zero. The extreme value (or Gompertz) distribution, however, is not symmetric, approaching 0 on the left more slowly than it approaches 1 on the right. You can use the extreme value distribution where such asymmetry is appropriate.

For ordinal response models, the response, Y, of an individual or an experimental unit can be restricted to one of a (usually small) number, $k+1 (k \geq 1)$, of ordinal values, denoted for convenience by $1, \ldots, k, k+1$. For example, the severity of coronary disease can be classified into three response

categories as 1=no disease, 2=angina pectoris, and 3=myocardial infraction. The PROBIT procedure fits a common slopes cumulative model, which is a parallel-lines regression model based on the cumulative probabilities of the response categories rather than on their individual probabilities. The cumulative model has the form

$$\Pr(Y \leq 1 \mid \mathbf{x}) = F(\mathbf{x}'\beta)$$

$$\Pr(Y \leq i \mid \mathbf{x}) = F(\alpha_i + \mathbf{x}'\beta), \quad 2 \leq i \leq k$$

where $\alpha_2, \ldots, \alpha_k$ are $k - 1$ intercept parameters. By default, the covariate vector $\mathbf{x}$ contains an overall intercept term.

You can set or estimate the natural (threshold) response rate C. Estimation of C can begin either from an initial value that you specify or from the rate observed in a control group. By default, the natural response rate is fixed at zero.

An observation in the data set analyzed by the PROBIT procedure might contain the response and explanatory values for one subject. Alternatively, it might provide the number of observed events from a number of subjects at a particular setting of the explanatory variables. In this case, PROC PROBIT models the probability of an event.

The PROBIT procedure now uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

Getting Started: PROBIT Procedure

The following example illustrates how you can use the PROBIT procedure to compute the threshold response rate and regression parameter estimates for quantal response data.

Estimating the Natural Response Threshold Parameter

Suppose you want to test the effect of a drug at 12 dosage levels. You randomly divide 180 subjects into 12 groups of 15—one group for each dosage level. You then conduct the experiment and, for each subject, record the presence or absence of a positive response to the drug. You summarize the data by counting the number of subjects responding positively in each dose group. Your data set is as follows:

```
data study;
   input Dose Respond @@;
   Number = 15;
   datalines;
0     3   1.1   4   1.3   4   2.0   3   2.2   5   2.8   4
3.7   5   3.9   9   4.4   8   4.8  11   5.9  12   6.8  13
;
run;
```

The variable dose represents the amount of drug administered. The first group, receiving a dose level of 0, is the control group. The variable number represents the number of subjects in each group. All groups are equal in size; hence, number has the value 15 for all observations. The variable respond represents the number of subjects responding to the associated drug dosage.

You can model the probability of positive response as a function of dosage by using the following statements:

```
ods graphics on;

proc probit data=study log10 optc plots=(predpplot ippplot);
   model respond/number=dose;
   output out=new p=p_hat;
run;

ods graphics off;
```

The DATA= option specifies that PROC PROBIT analyze the SAS data set study. The LOG10 option replaces the first continuous independent variable (dose) with its common logarithm. The OPTC option estimates the natural response rate. When you use the LOG10 option with the OPTC option, any observations with a dose value less than or equal to zero are used in the estimation as a control group.

The PLOTS= option in the PROC PROBIT statement, together with the ODS GRAPHICS statement, requests two plots for the estimated probability values and dosage levels. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the PROBIT procedure, see the section "ODS Graphics" on page 5325.

The MODEL statement specifies a proportional response by using the variables respond and number in *events/trials* syntax. The variable dose is the stimulus or explanatory variable.

The OUTPUT statement creates a new data set, new, that contains all the variables in the original data set, and a new variable, p_hat, that represents the predicted probabilities.

The results from this analysis are displayed in the following figures.

Figure 71.1 displays background information about the model fit. Included are the name of the input data set, the response variables used, and the number of observations, events, and trials. The last line in Figure 71.1 shows the final value of the log-likelihood function.

Figure 71.2 displays the table of parameter estimates for the model. The parameter C, which is the natural response threshold or the proportion of individuals responding at zero dose, is estimated to be 0.2409. Since both the intercept and the slope coefficient have significant p-values (0.0020, 0.0010), you can write the model for

$$\Pr(\text{response}) = C + (1 - C)F(\mathbf{x}'\beta)$$

as

$$\Pr(\text{response}) = 0.2409 + 0.7591(\Phi(-4.1439 + 6.2308 \times \log_{10}(\text{dose})))$$

where Φ is the normal cumulative distribution function.

Finally, PROC PROBIT specifies the resulting tolerance distribution by providing the mean MU and scale parameter SIGMA as well as the covariance matrix of the distribution parameters in Figure 71.3.

Figure 71.1 Model Fitting Information for the PROBIT Procedure

The Probit Procedure

Model Information

Data Set	WORK.STUDY
Events Variable	Respond
Trials Variable	Number
Number of Observations	12
Number of Events	81
Number of Trials	180
Number of Events In Control Group	3
Number of Trials In Control Group	15
Name of Distribution	Normal
Log Likelihood	-104.3945783

Figure 71.2 Model Parameter Estimates for the PROBIT Procedure

Analysis of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-4.1438	1.3415	-6.7731	-1.5146	9.54	0.0020
Log10(Dose)	1	6.2308	1.8996	2.5076	9.9539	10.76	0.0010
_C_	1	0.2409	0.0523	0.1385	0.3433		

Figure 71.3 Tolerance Distribution Estimates for the PROBIT Procedure

Estimated Covariance Matrix for Tolerance Parameters

	MU	SIGMA	_C_
MU	0.001158	-0.000493	0.000954
SIGMA	-0.000493	0.002394	-0.000999
_C_	0.000954	-0.000999	0.002731

The PLOT=PREDPPLOT option creates the plot in Figure 71.4, showing the relationship between dosage level, observed response proportions, and estimated probability values. The dashed lines represent pointwise confidence bands for the fitted probabilities, and a reference line is plotted at the estimated threshold value of 0.24.

The PLOT=IPPPLOT option creates the plot in Figure 71.5, showing the inverse relationship between dosage level and observed response proportions/estimated probability values. The dashed lines represent pointwise fiducial limits for the predicted values of the dose variable, and a reference line is also plotted at the estimated threshold value of 0.24.

The two plot options can be put together with the PLOTS= option, as shown in the PROC PROBIT statement.

Figure 71.4 Plot of Observed and Fitted Probabilities versus Dose Level

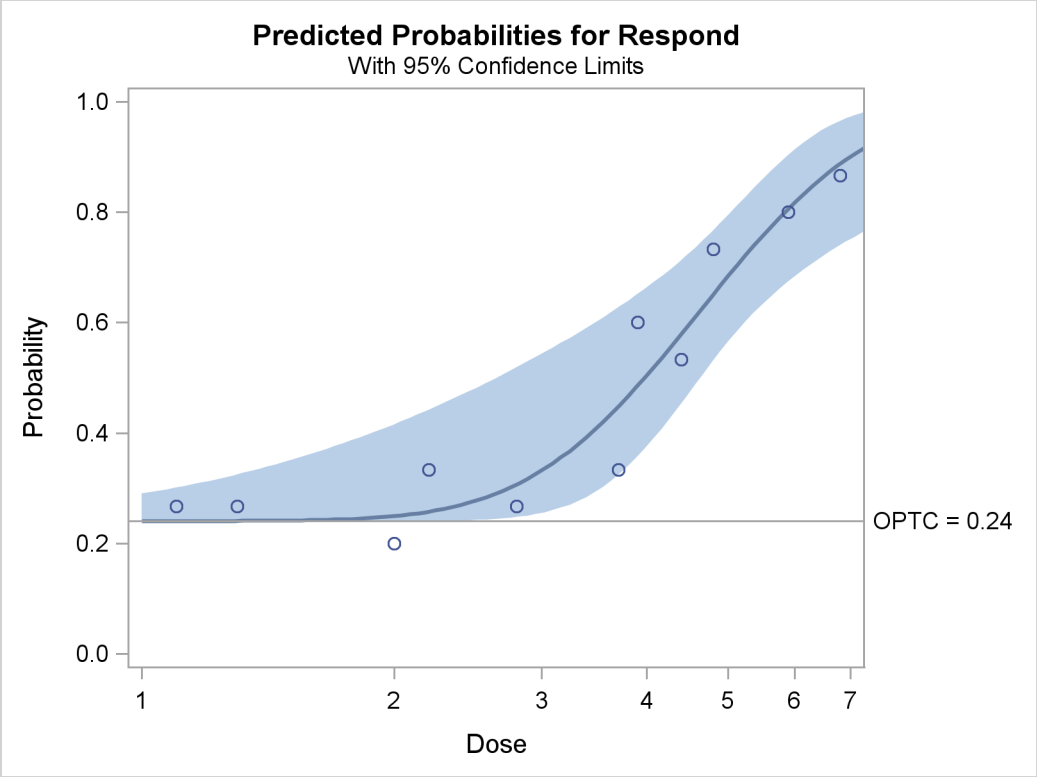
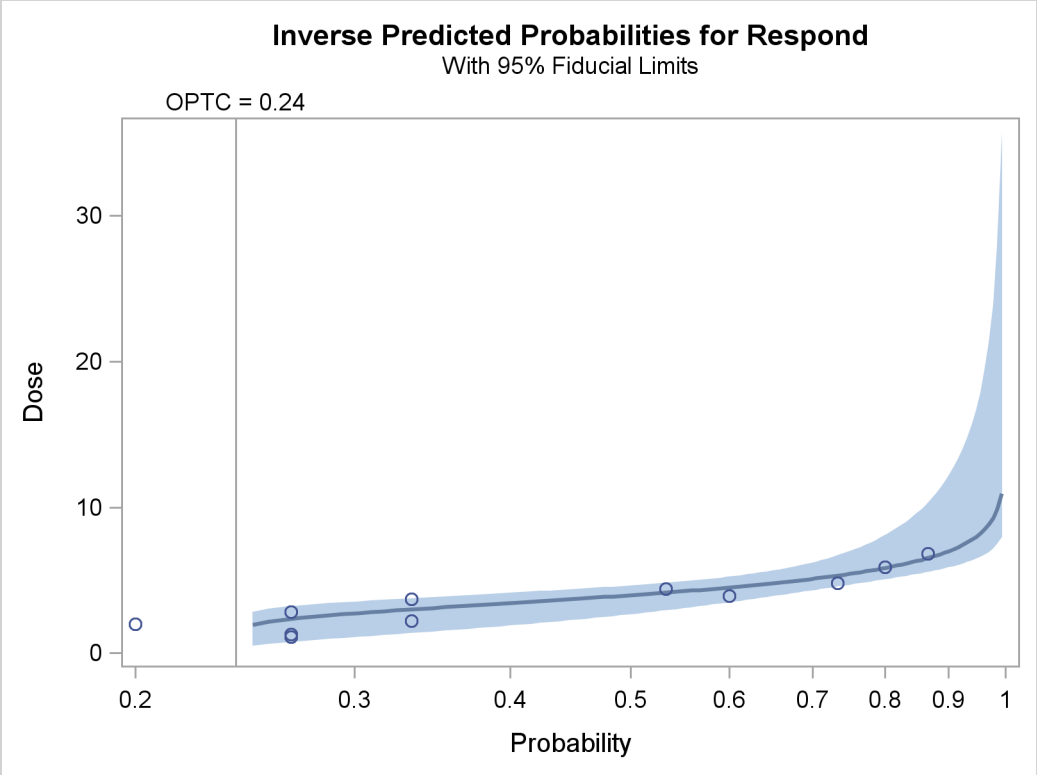


Figure 71.5 Inverse Predicted Probability Plot with Fiducial Limits



Syntax: PROBIT Procedure

The following statements are available in PROC PROBIT:

PROC PROBIT < *options* > ;
MODEL *response=independents* < / *options* > ;

BY *variables* ;
CLASS *variables* ;
OUTPUT < *OUT=SAS-data-set* > < *options* > ;
WEIGHT *variable* ;

CDFPLOT < *VAR = variable* > < *options* > ;
INSET < *keyword-list* > < / *options* > ;
IPPPLOT < *VAR = variable* > < *options* > ;
LPREDPLOT < *VAR = variable* > < *options* > ;
PREDPPLOT < *VAR = variable* > < *options* > ;

A MODEL statement is required. Only a single MODEL statement can be used with one invocation of the PROBIT procedure. If multiple MODEL statements are present, only the last one is used. Main effects and higher-order terms can be specified in the MODEL statement, as in the GLM procedure. If a CLASS statement is used, it must precede the MODEL statement.

The CDFPLOT, INSET, IPPPLOT, LPREDPLOT, and PREDPPLOT statements are used to produce graphical output. You can use any appropriate combination of the graphical statements after the MODEL statement.

PROC PROBIT Statement

PROC PROBIT < *options* > **;**

The PROC PROBIT statement starts the procedure. You can specify the following options in the PROC PROBIT statement.

COVOUT
writes the parameter estimate covariance matrix to the OUTEST= data set.

C=*rate*
OPTC
controls how the natural response is handled. Specify the OPTC option to request that the natural response rate C be estimated. Specify the C=*rate* option to set the natural response rate or to provide the initial estimate of the natural response rate. The natural response rate value must be a number between 0 and 1.

- If you specify neither the OPTC nor the C= option, a natural response rate of zero is assumed.
- If you specify both the OPTC and the C= option, the C= option should be a reasonable initial estimate of the natural response rate. For example, you could use the ratio of the number of responses to the number of subjects in a control group.
- If you specify the C= option but not the OPTC option, the natural response rate is set to the specified value and not estimated.
- If you specify the OPTC option but not the C= option, PROC PROBIT's action depends on the response variable, as follows:
  - If you specify either the LN or LOG10 option and some subjects have the first independent variable (dose) values less than or equal to zero, these subjects are treated as a control group. The initial estimate of C is then the ratio of the number of responses to the number of subjects in this group.
  - If you do not specify the LN or LOG10 option or if there is no control group, then one of the following occurs:
    - If all responses are greater than zero, the initial estimate of the natural response rate is the minimal response rate (the ratio of the number of responses to the number of subjects in a dose group) across all dose levels.
    - If one or more of the responses is zero (making the response rate zero in that dose group), the initial estimate of the natural rate is the reciprocal of twice the largest number of subjects in any dose group in the experiment.

DATA=*SAS-data-set*
specifies the SAS data set to be used by PROC PROBIT. By default, the procedure uses the most recently created SAS data set.

GOUT=*graphics-catalog*

specifies a graphics catalog in which to save graphics output.

HPROB=*p*

specifies a minimum probability level for the Pearson chi-square to indicate a good fit. The default value is 0.10. The LACKFIT option must also be specified for this option to have any effect. For Pearson goodness-of-fit chi-square values with probability greater than the HPROB= value, the fiducial limits, if requested with the INVERSECL option, are computed by using a critical value of 1.96. For chi-square values with probability less than the value of the HPROB= option, the critical value is a 0.95 two-sided quantile value taken from the t distribution with degrees of freedom equal to $(k-1)\times m-q$, where k is the number of levels for the response variable, m is the number of different sets of independent variable values, and q is the number of parameters fit in the model. Note that the HPROB= option can also appear in the MODEL statement.

INEST=*SAS-data-set*

specifies an input SAS data set that contains initial estimates for all the parameters in the model. See the section "INEST= *SAS-data-set*" on page 5312 for a detailed description of the contents of the INEST= data set.

INVERSECL

computes confidence limits for the values of the first continuous independent variable (such as dose) that yield selected response rates. If the algorithm fails to converge (this can happen when C is nonzero), missing values are reported for the confidence limits. See the section "Inverse Confidence Limits" on page 5315 for details. Note that the INVERSECL option can also appear in the MODEL statement.

LACKFIT

performs two goodness-of-fit tests (a Pearson chi-square test and a log-likelihood ratio chi-square test) for the fitted model.

To compute the test statistics, proper grouping of the observations into subpopulations is needed. You can use the AGGREGATE or AGGREGATE= option for this end. See the entry for the AGGREGATE and AGGREGATE= options under the MODEL statement. If neither AGGREGATE nor AGGREGATE= is specified, PROC PROBIT assumes each observation is from a separate subpopulation and computes the goodness-of-fit test statistics only for the *events/trials* syntax.

NOTE: This test is not appropriate if the data are very sparse, with only a few values at each set of the independent variable values.

If the Pearson chi-square test statistic is significant, then the covariance estimates and standard error estimates are adjusted. See the section "Lack-of-Fit Tests" on page 5313 for a description of the tests. Note that the LACKFIT option can also appear in the MODEL statement.

LOG

LN

analyzes the data by replacing the first continuous independent variable with its natural logarithm. This variable is usually the level of some treatment such as dosage. In addition to the

usual output given by the INVERSECL option, the estimated dose values and 95% fiducial limits for dose are also displayed. If you specify the OPTC option, any observations with a dose value less than or equal to zero are used in the estimation as a control group. If you do not specify the OPTC option with the LOG or LN option, then any observations with the first continuous independent variable values less than or equal to zero are ignored.

LOG10

specifies an analysis like that of the LN or LOG option, except that the common logarithm (log to the base 10) of the dose value is used rather than the natural logarithm.

NAMELEN=*n*

specifies the length of effect names in tables and output data sets to be *n* characters, where *n* is a value between 20 and 200. The default length is 20 characters.

NOPRINT

suppresses the display of all output including graphics. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 20, "Using the Output Delivery System."

OPTC

controls how the natural response is handled. See the description of the C= option on page 5264 for details.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of the classification variables specified in the CLASS statement, including the levels of the response variable. Response level ordering is important since PROC PROBIT always models the probability of response levels at the beginning of the ordering. See the section "Response Level Ordering" on page 5309 for further details. This ordering also determines which parameters in the model correspond to each level in the data. The following table shows how PROC PROBIT interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	formatted value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For the values FORMATTED and INTERNAL, the sort order is machine dependent. For more information about sorting order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide*.

OUTEST=*SAS-data-set*

specifies a SAS data set to contain the parameter estimates and, if the COVOUT option is specified, their estimated covariances. If you omit this option, the output data set is not created. The contents of the data set are described in the section "OUTEST= *SAS-data-set*" on page 5316.

PLOT | PLOTS <=*plot-request*>

PLOT | PLOTS <=(*plot-request* <...*plot-request*>)>

specifies options that control details of the plots created by ODS Graphics. These plots are related to a dose variable, which is identified as the first single continuous independent variable in the MODEL statement. If there are interaction terms with this variable in the model, the PROBIT procedure will not produce any plot.

You can specify more than one plot request within the parentheses after PLOTS=. For a single plot request, you can omit the parentheses. The following plot requests are available.

ALL

creates all appropriate plots.

CDFPLOT<(LEVEL=(*character-list*))>

requests the plot of predicted cumulative distribution function (CDF) of the multinomial response variable as a function of a single continuous independent variable (dose variable). This single continuous independent variable must be the first single continuous independent variable listed in the MODEL statement. You can request this plot only with a multinomial model.

The LEVEL= suboption specify the levels of the multinomial response variable for which the CDF curves are requested. There are $k-1$ curves for a k-level multinomial response variable (for the highest level, it is the constant line 1). You can specify any of them to be plotted by the LEVEL= suboption.

IPPPLOT

requests the inverse plot of the predicted probability against the first single continuous variable (dose variable) in the MODEL statement for the binomial model. You can request this plot only with a binomial model. The confidence limits for the predicted values of the dose variable are the computed fiducial limits, not the inverse of the confidence limits of the predicted probabilities. Refer to the section "Inverse Confidence Limits" on page 5315 for more details.

LPREDPLOT<(LEVEL=(*character-list*))>

requests the plot of the linear predictor $\mathbf{x}'\mathbf{b}$ against the first single continuous variable (dose variable) in the MODEL statement for either the binomial model or the multinomial model. The confidence limits for the predicted values are available only for the binomial model.

For the multinomial model, you can use the LEVEL= suboption to specify the levels for which the linear predictor lines are plotted.

NONE

suppresses all plots.

PREDPPLOT<(LEVEL=(*character-list*))>

requests the plot of the predicted probability against the first single continuous variable (dose variable) in the MODEL statement for both the binomial model and the multinomial model. Confidence limits are available only for the binomial model.

For the multinomial model, you can use the LEVEL= suboption to specify the levels for which the linear predictor lines are plotted.

XDATA=*SAS-data-set*
specifies an input SAS data set that contains values for all the independent variables in the MODEL statement and variables in the CLASS statement. If there are covariates specified in a MODEL statement, you specify fixed values for the effects in the MODEL statement by the XDATA= data set when predicted values and/or fiducial limits for a single continuous variable (dose variable) are required. These specified values for the effects in the MODEL statement are also used for generating plots. See the section "XDATA= *SAS-data-set*" on page 5317 for a detailed description of the contents of the XDATA= data set.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC PROBIT to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order on each of the BY variables, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the PROBIT procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CDFPLOT Statement

CDFPLOT < *var = variable* > < *options* > **;**

The CDFPLOT statement plots the predicted cumulative distribution function (CDF) of the multinomial response variable as a function of a single continuous independent variable (dose variable). You can use this statement only after a multinomial model statement.

VAR=*variable*

specifies a single continuous variable (dose variable) in the independent variable list of the MODEL statement. If a VAR= variable is not specified, the first single continuous variable in the independent variable list of the MODEL statement is used. If such a variable does not exist in the independent variable list of the MODEL statement, an error is reported.

The predicted cumulative distribution function is defined as

$$\hat{F}_j(\mathbf{x}) = C + (1 - C)F(\hat{a}_j + \mathbf{x}'\hat{\mathbf{b}})$$

where $j = 1, \ldots, k$ are the indexes of the k levels of the multinomial response variable, F is the CDF of the distribution used to model the cumulative probabilities, $\hat{\mathbf{b}}$ is the vector of estimated parameters, $\mathbf{x}$ is the covariate vector, $\hat{a}_j$ are estimated ordinal intercepts with $\hat{a}_1 = 0$, and C is the threshold parameter, either known or estimated from the model. Let x_1 be the covariate corresponding to the dose variable and $\mathbf{x}_{-1}$ be the vector of the rest of the covariates. Let the corresponding estimated parameters be $\hat{b}_1$ and $\hat{\mathbf{b}}_{-1}$. Then

$$\hat{F}_j(\mathbf{x}) = C + (1 - C)F(\hat{a}_j + x_1\hat{b}_1 + \mathbf{x}'_{-1}\hat{\mathbf{b}}_{-1})$$

To plot $\hat{F}_j$ as a function of x_1, $\mathbf{x}_{-1}$ must be specified. You can use the XDATA= option to provide the values of $\mathbf{x}_{-1}$ (see the XDATA= option in the PROC PROBIT statement for details), or use the default values that follow the rules:

- If the effect contains a continuous variable (or variables), the overall mean of this effect is used.
- If the effect is a single classification variable, the highest level of the variable is used.

options

specify the levels of the multinomial response variable for which the CDF curves are requested, and add features to the plot. There are $k - 1$ curves for a k-level multinomial response variable (for the highest level, it is the constant line 1). You can specify any of them to be plotted by the LEVEL= option in the CDFPLOT statement. See the LEVEL= option for how to specify the levels.

An attached box on the right side of the plot is used to label these curves with the names of their levels. You can specify the color of this box by using the CLABBOX= option.

You can use options in the CDFPLOT statement to do the following:

- superimpose specification limits
- specify the levels for which the CDF curves are requested
- specify graphical enhancements (such as color or text height)

Summary of Options

Table 71.1 through Table 71.7 list all *options* by function. The "Dictionary of Options" on page 5272 describes each option in detail.

CDF Options

Table 71.1 Options for CDFPLOT

LEVEL=(*character-list*)	specifies the names of the levels for which the CDF curves are requested
NOTHRESH	suppresses the threshold line
THRESHLABPOS=*value*	specifies the position for the label of the threshold line

General Options

Table 71.2 Color Options

CAXIS=*color*	specifies color for axis
CFIT=*color*	specifies color for fitted curves
CFRAME=*color*	specifies color for frame
CGRID=*color*	specifies color for grid lines
CHREF=*color*	specifies color for HREF= lines
CLABBOX=*color*	specifies color for label box
CTEXT=*color*	specifies color for text
CVREF=*color*	specifies color for VREF= lines

Table 71.3 Options to Enhance Plots Produced on Graphics Devices

ANNOTATE=*SAS-data-set*	specifies an Annotate data set
INBORDER	requests a border around plot
LFIT=*linetype*	specifies line style for fitted curves
LGRID=*linetype*	specifies line style for grid lines
NOFRAME	suppresses the frame around plotting areas
NOGRID	suppresses grid lines
NOFIT	suppresses CDF curves
NOHLABEL	suppresses horizontal labels
NOHTICK	suppresses horizontal ticks
NOVTICK	suppresses vertical ticks
TURNVLABELS	vertically strings out characters in vertical labels
WFIT=*n*	specifies thickness for fitted curves
WGRID=*n*	specifies thickness for grids
WREFL=*n*	specifies thickness for reference lines

Table 71.4 Axis Options

HAXIS=*value1 to value2 <by value3>*	specifies tick mark values for horizontal axis
HOFFSET=*value*	specifies offset for horizontal axis
HLOWER=*value*	specifies lower limit on horizontal axis scale
HUPPER=*value*	specifies upper limit on horizontal axis scale
NHTICK=*n*	specifies number of ticks for horizontal axis
NVTICK=*n*	specifies number of ticks for vertical axis
VAXIS=*value1 to value2 <by value3>*	specifies tick mark values for vertical axis
VAXISLABEL=*'label'*	specifies label for vertical axis
VOFFSET=*value*	specifies offset for vertical axis
VLOWER=*value*	specifies lower limit on vertical axis scale
VUPPER=*value*	specifies upper limit on vertical axis scale
WAXIS=*n*	specifies thickness for axis

Table 71.5 Graphics Catalog Options

DESCRIPTION=*'string'*	specifies description for graphics catalog member
NAME=*'string'*	specifies name for plot in graphics catalog

Table 71.6 Options for Text Enhancement

FONT=*font*	specifies software font for text
HEIGHT=*value*	specifies height of text outside framed areas
INFONT=*font*	specifies software font for text inside framed areas
INHEIGHT=*value*	specifies height of text inside framed areas

Table 71.7 Options for Reference Lines

HREF< (INTERSECT)> =value-list	requests horizontal reference line
HREFLABELS= ('label1',...,'labeln')	specifies labels for HREF= lines
HREFLABPOS=*n*	specifies vertical position of labels for HREF= lines
LHREF=*linetype*	specifies line style for HREF= lines
LVREF=*linetype*	specifies line style for VREF= lines
VREF<(INTERSECT)> =value-list	requests vertical reference line
VREFLABELS= ('label1',...,'labeln')	specifies labels for VREF= lines
VREFLABPOS=*n*	specifies horizontal position of labels for VREF= lines

Dictionary of Options

The following entries provide detailed descriptions of the *options* in the CDFPLOT statement.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an Annotate data set, as described in *SAS/GRAPH Software: Reference*, that enables you to add features to the CDF plot. The ANNOTATE= data set you specify in the CDFPLOT statement is used for all plots created by the statement.

CAXIS=*color*

CAXES=*color*

specifies the color used for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement. The default is the first color in the device color list.

CFIT=*color*

specifies the color for the fitted CDF curves. The default is the first color in the device color list.

CFRAME=*color*

CFR=*color*

specifies the color for the area enclosed by the axes and frame. This area is not shaded by default.

CGRID=*color*

specifies the color for grid lines. The default is the first color in the device color list.

CLABBOX=*color*

specifies the color for the area enclosed by the label box for CDF curves. This area is not shaded by default.

CHREF=*color*

CH=*color*

specifies the color for lines requested by the HREF= option. The default is the first color in the device color list.

CTEXT=*color*

specifies the color for tick mark values and axis labels. The default is the color specified for the CTEXT= option in the most recent GOPTIONS statement.

CVREF=*color*

CV=*color*

specifies the color for lines requested by the VREF= option. The default is the first color in the device color list.

DESCRIPTION="*string*"

DES="*string*"

specifies a description, up to 40 characters, that appears in the PROC GREPLAY master menu. The default is the variable name.

FONT=*font*

specifies a software font for reference line and axis labels. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the most recent GOPTIONS statement. Hardware characters are used by default.

HAXIS=*value1 to value2< by value3>*

specifies tick mark values for the horizontal axis; *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. If *value3* is omitted, a value of 1 is used.

Examples of HAXIS= lists follow:

```
haxis = 0 to 10
haxis = 2 to 10 by 2
haxis = 0 to 200 by 10
```

HEIGHT=*value*

specifies the height of text used outside framed areas. The default value is 3.846 (in percentage).

HLOWER=*value*

specifies the lower limit on the horizontal axis scale. The HLOWER= option specifies *value* as the lower horizontal axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the HAXIS= option is used.

HOFFSET=*value*

specifies offset for horizontal axis. The default value is 1.

HUPPER=*value*

specifies *value* as the upper horizontal axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the HAXIS= option is used.

HREF < (INTERSECT) > =*value-list*

requests reference lines perpendicular to the horizontal axis. If (INTERSECT) is specified, a second reference line perpendicular to the vertical axis is drawn that intersects the fit line at the same point as the horizontal axis reference line. If a horizontal axis reference line label is specified, the intersecting vertical axis reference line is labeled with the vertical axis value. See also the CHREF=, HREFLABELS=, and LHREF= options.

HREFLABELS=*'label1',...,'labeln'*

HREFLABEL=*'label1',...,'labeln'*

HREFLAB=*'label1',...,'labeln'*

specifies labels for the lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

HREFLABPOS=*n*

specifies the vertical position of labels for HREF= lines. The following table shows valid values for *n* and the corresponding label placements.

n	**label placement**
1	top
2	staggered from top
3	bottom
4	staggered from bottom
5	alternating from top
6	alternating from bottom

INBORDER

requests a border around CDF plots.

LEVEL=(*character-list***)**

ORDINAL=(*character-list***)**

specifies the names of the levels for which CDF curves are requested. Names should be quoted and separated by space. If there is no correct name provided, no CDF curve is plotted.

LFIT=*linetype*

specifies a line style for fitted curves. By default, fitted curves are drawn by connecting solid lines (*linetype = 1*).

LGRID=*linetype*

specifies a line style for all grid lines. *linetype* is between 1 and 46. The default is 35.

LHREF=*linetype*

LH=*linetype*

specifies the line type for lines requested by the HREF= option. The default is 2, which produces a dashed line.

LVREF=*linetype*

LV=*linetype*

specifies the line type for lines requested by the VREF= option. The default is 2, which produces a dashed line.

NAME=*'string'*

specifies a name for the plot, up to eight characters, that appears in the PROC GREPLAY master menu. The default is 'PROBIT'.

NOFIT

suppresses the fitted CDF curves.

NOFRAME

suppresses the frame around plotting areas.

NOGRID

suppresses grid lines.

NOHLABEL

suppresses horizontal labels.

NOHTICK

suppresses horizontal tick marks.

NOTHRESH

suppresses the threshold line.

NOVLABEL

suppresses vertical labels.

NOVTICK

suppresses vertical tick marks.

THRESHLABPOS=*n*

specifies the horizontal position of labels for the threshold line. The following table shows valid values for *n* and the corresponding label placements.

n	**label placement**
1	left
2	right

VAXIS=*value1 to value2< by value3 >*

specifies tick mark values for the vertical axis; *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. This method of specification of tick marks is not valid for logarithmic axes. If *value3* is omitted, a value of 1 is used.

Examples of VAXIS= lists follow:

```
vaxis = 0 to 10
vaxis = 0 to 2 by .1
```

VAXISLABEL=*'string'*
specifies a label for the vertical axis.

VLOWER=*value*
specifies the lower limit on the vertical axis scale. The VLOWER= option specifies *value* as the lower vertical axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

VREF=*value-list*
requests reference lines perpendicular to the vertical axis. If (INTERSECT) is specified, a second reference line perpendicular to the horizontal axis is drawn that intersects the fit line at the same point as the vertical axis reference line. If a vertical axis reference line label is specified, the intersecting horizontal axis reference line is labeled with the horizontal axis value. See also the CVREF=, LVREF=, and VREFLABELS= options.

VREFLABELS=*'label1',...,'labeln'*

VREFLABEL=*'label1',...,'labeln'*

VREFLAB=*'label1',...,'labeln'*
specifies labels for the lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

VREFLABPOS=*n*
specifies the horizontal position of labels for VREF= lines. The following table shows valid values for *n* and the corresponding label placements.

n	**label placement**
1	left
2	right

VUPPER=*value*
specifies the upper limit on the vertical axis scale. The VUPPER= option specifies *value* as the upper vertical axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

WAXIS=*n*
specifies line thickness for axes and frame. The default value is 1.

WFIT=*n*
specifies line thickness for fitted curves. The default value is 1.

WGRID=*n*
specifies line thickness for grids. The default value is 1.

WREFL=*n*
specifies line thickness for reference lines. The default value is 1.

CLASS Statement

CLASS *variables* **;**

The CLASS statement names the classification variables to be used in the analysis. Classification variables can be either character or numeric. If a single response variable is specified in the MODEL statement, it must also be specified in a CLASS statement.

Class levels are determined from the formatted values of the CLASS variables. Thus, you can use formats to group values into levels. See the discussion of the FORMAT procedure in *SAS Language Reference: Dictionary*.

If the CLASS statement is used, it must appear before any of the MODEL statements.

INSET Statement

INSET < *keyword-list* > < / *options* > **;**

The box or table of summary information produced on plots made with the CDFPLOT, IPPPLOT, LPREDPLOT, or PREDPPLOT statement is called an *inset*. You can use the INSET statement to customize both the information that is printed in the inset box and the appearance of the inset box. To supply the information that is displayed in the inset box, you specify *keywords* corresponding to the information you want shown. For example, the following statements produce a predicted probability plot with the number of trials, the number of events, the name of the distribution, and the estimated optimum natural threshold in the inset.

```
proc probit data=epidemic;
   model r/n = dose;
   predpplot ;
   inset nobs ntrials nevents dist optc;
run;
```

By default, inset entries are identified with appropriate labels. However, you can provide a customized label by specifying the *keyword* for that entry followed by the equal sign (=) and the label in quotes. For example, the following INSET statement produces an inset containing the number of observations and the name of the distribution, labeled “Sample Size” and “Distribution” in the inset.

```
inset nobs='Sample Size' dist='Distribution';
```

If you specify a keyword that does not apply to the plot you are creating, then the keyword is ignored.

The *options* control the appearance of the box.

If you specify more than one INSET statement, only the first one is used.

Keywords Used in the INSET Statement

Table 71.8 and Table 71.9 list keywords available in the INSET statement to display summary statistics, distribution parameters, and distribution fitting information.

Table 71.8 Summary Statistics

NOBS	number of observations
NTRIALS	number of trials
NEVENTS	number of events
C	user-input threshold
OPTC	estimated natural threshold
NRESPLEV	number of levels of the response variable

Table 71.9 General Information

CONFIDENCE	confidence coefficient for all confidence intervals or for the Weibayes fit
DIST	name of the distribution

Options Used in the INSET Statement

Table 71.10 and Table 71.11 list the options available in the INSET statement.

Table 71.10 Color and Pattern Options

CFILL=*color*	specifies color for filling box
CFILLH=*color*	specifies color for filling box header
CFRAME=*color*	specifies color for frame
CHEADER=*color*	specifies color for text in header
CTEXT=*color*	specifies color for text

Table 71.11 General Appearance Options

FONT=*font*	specifies software font for text
HEIGHT=*value*	specifies height of text
HEADER=*'quoted string'*	specifies text for header or box title
NOFRAME	omits frame around box
POS= *value* <DATA | PERCENT>	determines the position of the inset. The *value* can be a compass point (N, NE, E, SE, S, SW, W, NW) or a pair of coordinates (x, y) enclosed in parentheses. The coordinates can be specified in axis percentage units or axis data units.
REFPOINT= *name*	specifies the reference point for an inset that is positioned by a pair of coordinates with the POS= option. You use the REFPOINT= option in conjunction with the POS= coordinates. The REFPOINT= option specifies which corner of the inset frame you have specified with coordinates (x, y), and it can take the value of BR (bottom right), BL (bottom left), TR (top right), or TL (top left). The default is REFPOINT=BL. If the inset position is specified as a compass point, then the REFPOINT= option is ignored.

IPPPLOT Statement

IPPPLOT < *var = variable* > < *options* > **;**

The IPPPLOT statement plots the inverse of the predicted probability against a single continuous variable (dose variable) in the MODEL statement for the binomial model. You can only use this statement after a binomial model statement. The confidence limits for the predicted values of the dose variable are the computed fiducial limits, not the inverse of the confidence limits of the predicted probabilities. Refer to the section "Inverse Confidence Limits" on page 5315 for more details.

VAR= *variable*

specifies a single continuous variable (dose variable) in the independent variable list of the MODEL statement. If a VAR= variable is not specified, the first single continuous variable in the independent variable list of the MODEL statement is used. If such a variable does not exist in the independent variable list of the MODEL statement, an error is reported.

For the binomial model, the response variable is a probability. An estimate of the dose level $\hat{x}_1$ needed for a response of p is given by

$$\hat{x}_1 = (F^{-1}(p) - \mathbf{x}'_{-1}\hat{\mathbf{b}}_{-1})/\hat{b}_1$$

where F is the cumulative distribution function used to model the probability, $\mathbf{x}_{-1}$ is the vector of the rest of the covariates, $\hat{\mathbf{b}}_{-1}$ is the vector of the estimated parameters corresponding to $\mathbf{x}_{-1}$, and $\hat{b}_1$ is the estimated parameter for the dose variable of interest.

To plot $\hat{x}_1$ as a function of p, $\mathbf{x}_{-1}$ must be specified. You can use the XDATA= option to provide the values of $\mathbf{x}_{-1}$ (see the XDATA= option in the PROC PROBIT statement for details), or use the default values that follow the rules:

- If the effect contains a continuous variable (or variables), the overall mean of this effect is used.
- If the effect is a single classification variable, the highest level of the variable is used.

options

add features to the plot.

You can use options in the IPPPLOT statement to do the following:

- superimpose specification limits
- suppress or add the observed data points on the plot
- suppress or add the fiducial limits on the plot
- specify graphical enhancements (such as color or text height)

Summary of Options

Table 71.12 through Table 71.18 list all *options* by function. The "Dictionary of Options" on page 5282 describes each option in detail.

IPP Options

Table 71.12 Plot Layout Options for IPPPLOT

NOCONF	suppresses fiducial limits
NODATA	suppresses observed data points on the plot
NOTHRESH	suppresses the threshold line
THRESHLABPOS=*value*	specifies the position for the label of the threshold line

General Options

Table 71.13 Color Options

CAXIS=*color*	specifies color for axis
CFIT=*color*	specifies color for fitted curves
CFRAME=*color*	specifies color for frame
CGRID=*color*	specifies color for grid lines
CHREF=*color*	specifies color for HREF= lines
CTEXT=*color*	specifies color for text
CVREF=*color*	specifies color for VREF= lines

Table 71.14 Options to Enhance Plots Produced on Graphics Devices

ANNOTATE=*SAS-data-set*	specifies an Annotate data set
INBORDER	requests a border around plot
LFIT=*linetype*	specifies line style for fitted curves and confidence limits
LGRID=*linetype*	specifies line style for grid lines
NOFRAME	suppresses the frame around plotting areas
NOGRID	suppresses grid lines
NOFIT	suppresses fitted curves
NOHLABEL	suppresses horizontal labels
NOHTICK	suppresses horizontal ticks
NOVTICK	suppresses vertical ticks
TURNVLABELS	vertically strings out characters in vertical labels
WFIT=*n*	specifies thickness for fitted curves
WGRID=*n*	specifies thickness for grids
WREFL=*n*	specifies thickness for reference lines

Table 71.15 Axis Options

HAXIS=*value1 to value2 <by value3>*	specifies tick mark values for horizontal axis
HOFFSET=*value*	specifies offset for horizontal axis
HLOWER=*value*	specifies lower limit on horizontal axis scale
HUPPER=*value*	specifies upper limit on horizontal axis scale
NHTICK=*n*	specifies number of ticks for horizontal axis
NVTICK=*n*	specifies number of ticks for vertical axis
VAXIS=*value1 to value2 <by value3>*	specifies tick mark values for vertical axis
VAXISLABEL=*'label'*	specifies label for vertical axis
VOFFSET=*value*	specifies offset for vertical axis
VLOWER=*value*	specifies lower limit on vertical axis scale
VUPPER=*value*	specifies upper limit on vertical axis scale
WAXIS=*n*	specifies thickness for axis

Table 71.16 Options for Reference Lines

HREF<(INTERSECT)> =*value-list*	requests horizontal reference line
HREFLABELS= ('label1',...,'labeln')	specifies labels for HREF= lines
HREFLABPOS=*n*	specifies vertical position of labels for HREF= lines
LHREF=*linetype*	specifies line style for HREF= lines
LVREF=*linetype*	specifies line style for VREF= lines
VREF<(INTERSECT)> =*value-list*	requests vertical reference line
VREFLABELS= ('label1',...,'labeln')	specifies labels for VREF= lines
VREFLABPOS=*n*	specifies horizontal position of labels for VREF= lines

Table 71.17 Graphics Catalog Options

DESCRIPTION='*string*'	specifies description for graphics catalog member
NAME='*string*'	specifies name for plot in graphics catalog

Table 71.18 Options for Text Enhancement

FONT=*font*	specifies software font for text
HEIGHT=*value*	specifies height of text used outside framed areas
INFONT=*font*	specifies software font for text inside framed areas
INHEIGHT=*value*	specifies height of text inside framed areas

Dictionary of Options

The following entries provide detailed descriptions of the *options* in the IPPPLOT statement.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an Annotate data set, as described in *SAS/GRAPH Software: Reference*, that enables you to add features to the ipp plot. The ANNOTATE= data set you specify in the IPPPLOT statement is used for all plots created by the statement.

CAXIS=*color*

CAXES=*color*

specifies the color used for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement. The default is the first color in the device color list.

CFIT=*color*

specifies the color for the fitted ipp curves. The default is the first color in the device color list.

CFRAME=*color*

CFR=*color*

specifies the color for the area enclosed by the axes and frame. This area is not shaded by default.

CGRID=*color*

specifies the color for grid lines. The default is the first color in the device color list.

CHREF=*color*

CH=*color*

specifies the color for lines requested by the HREF= option. The default is the first color in the device color list.

CTEXT=*color*

specifies the color for tick mark values and axis labels. The default is the color specified for the CTEXT= option in the most recent GOPTIONS statement.

CVREF=*color*

CV=*color*

specifies the color for lines requested by the VREF= option. The default is the first color in the device color list.

DESCRIPTION=*'string'*

DES=*'string'*

specifies a description, up to 40 characters, that appears in the PROC GREPLAY master menu. The default is the variable name.

FONT=*font*

specifies a software font for reference line and axis labels. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the most recent GOPTIONS statement. Hardware characters are used by default.

HAXIS=*value1 to value2< by value3>*

specifies tick mark values for the horizontal axis; *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. If *value3* is omitted, a value of 1 is used.

Examples of HAXIS= lists follow:

```
haxis = 0 to 10
haxis = 2 to 10 by 2
haxis = 0 to 200 by 10
```

HEIGHT=*value*

specifies the height of text used outside framed areas. The default value is 3.846 (in percentage).

HLOWER=*value*

specifies the lower limit on the horizontal axis scale. The HLOWER= option specifies *value* as the lower horizontal axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the HAXIS= option is used.

HOFFSET=*value*

specifies offset for horizontal axis. The default value is 1.

HUPPER=*value*

specifies *value* as the upper horizontal axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the HAXIS= option is used.

HREF < (INTERSECT) > =*value-list*

requests reference lines perpendicular to the horizontal axis. If (INTERSECT) is specified, a second reference line perpendicular to the vertical axis is drawn that intersects the fit line at the same point as the horizontal axis reference line. If a horizontal axis reference line label is specified, the intersecting vertical axis reference line is labeled with the vertical axis value. See also the CHREF=, HREFLABELS=, and LHREF= options.

HREFLABELS=*'label1'***,...,***'labeln'*

HREFLABEL=*'label1'***,...,***'labeln'*

HREFLAB=*'label1'***,...,***'labeln'*

specifies labels for the lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

HREFLABPOS=*n*

specifies the vertical position of labels for HREF= lines. The following table shows valid values for *n* and the corresponding label placements.

n	label placement
1	top
2	staggered from top
3	bottom
4	staggered from bottom
5	alternating from top
6	alternating from bottom

INBORDER

requests a border around ipp plots.

LFIT=*linetype*

specifies a line style for fitted curves and confidence limits. By default, fitted curves are drawn by connecting solid lines (*linetype = 1*) and confidence limits are drawn by connecting dashed lines (*linetype = 3*).

LGRID=*linetype*

specifies a line style for all grid lines. The value for *linetype* must be between 1 and 46. The default is 35.

LHREF=*linetype*

LH=*linetype*

specifies the line type for lines requested by the HREF= option. The default is 2, which produces a dashed line.

LVREF=*linetype*

LV=*linetype*

specifies the line type for lines requested by the VREF= option. The default is 2, which produces a dashed line.

NAME=*'string'*

specifies a name for the plot, up to eight characters, that appears in the PROC GREPLAY master menu. The default is 'PROBIT'.

NOCONF

suppresses fiducial limits from the plot.

NODATA

suppresses observed data points from the plot.

NOFIT

suppresses the fitted ipp curves.

NOFRAME

suppresses the frame around plotting areas.

NOGRID

suppresses grid lines.

NOHLABEL

suppresses horizontal labels.

NOHTICK

suppresses horizontal tick marks.

NOTHRESH

suppresses the threshold line.

NOVLABEL

suppresses vertical labels.

NOVTICK

suppresses vertical tick marks.

THRESHLABPOS=*n*

specifies the vertical position of labels for the threshold line. The following table shows valid values for *n* and the corresponding label placements.

n	label placement
1	top
2	bottom

VAXIS=*value1 to value2< by value3 >*

specifies tick mark values for the vertical axis; *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. This method of specification of tick marks is not valid for logarithmic axes. If *value3* is omitted, a value of 1 is used.

Examples of VAXIS= lists follow:

```
vaxis = 0 to 10
vaxis = 0 to 2 by .1
```

VAXISLABEL=*'string'*

specifies a label for the vertical axis.

VLOWER=*value*

specifies the lower limit on the vertical axis scale. The VLOWER= option specifies *value* as the lower vertical axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

VREF=*value-list*

requests reference lines perpendicular to the vertical axis. If (INTERSECT) is specified, a second reference line perpendicular to the horizontal axis is drawn that intersects the fit line at the same point as the vertical axis reference line. If a vertical axis reference line label is specified, the intersecting horizontal axis reference line is labeled with the horizontal axis value. See also the CVREF=, LVREF=, and VREFLABELS= options.

VREFLABELS=*'label1',...,'labeln'*

VREFLABEL=*'label1',...,'labeln'*

VREFLAB=*'label1',...,'labeln'*

specifies labels for the lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

VREFLABPOS=*n*

specifies the horizontal position of labels for VREF= lines. The following table shows valid values for *n* and the corresponding label placements.

n	label placement
1	left
2	right

VUPPER=*value*

specifies the upper limit on the vertical axis scale. The VUPPER= option specifies *value* as the upper vertical axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

WAXIS=*n*

specifies line thickness for axes and frame. The default value is 1.

WFIT=*n*
specifies line thickness for fitted curves. The default value is 1.

WGRID=*n*
specifies line thickness for grids. The default value is 1.

WREFL=*n*
specifies line thickness for reference lines. The default value is 1.

LPREDPLOT Statement

LPREDPLOT < *var = variable* > < *options* > **;**

The LPREDPLOT statement plots the linear predictor $\mathbf{x}'\mathbf{b}$ against a single continuous variable (dose variable) in the MODEL statement for either the binomial model or the multinomial model. The confidence limits for the predicted values are available only for the binomial model.

VAR= *variable*
specifies a single continuous variable (dose variable) in the independent variable list of the MODEL statement for which the linear predictor plot is plotted. If a VAR= variable is not specified, the first single continuous variable in the independent variable list of the MODEL statement is used. If such a variable does not exist in the independent variable list of the MODEL statement, an error is reported.

Let x_1 be the covariate of the dose variable, $\mathbf{x}_{-1}$ be the vector of the rest of the covariates, $\hat{\mathbf{b}}_{-1}$ be the vector of estimated parameters corresponding to $\mathbf{x}_{-1}$, and $\hat{b}_1$ be the estimated parameter for the dose variable of interest.

To plot $\hat{\mathbf{x}}'\mathbf{b}$ as a function of x_1, $\mathbf{x}_{-1}$ must be specified. You can use the XDATA= option to provide the values of $\mathbf{x}_{-1}$ (see the XDATA= option in the PROC PROBIT statement for details), or use the default values that follow these rules:

- If the effect contains a continuous variable (or variables), the overall mean of this effect is used.
- If the effect is a single classification variable, the highest level of the variable is used.

options
add features to the plot.

For the multinomial model, you can use the LEVEL= option to specify the levels for which the linear predictor lines are plotted. The lines are labeled by the names of their levels in the middle.

You can use options in the LPREDPLOT statement to do the following:

- superimpose specification limits
- suppress or add the observed data points on the plot for the binomial model
- suppress or add the confidence limits for the binomial model

- specify the levels for which the linear predictor lines are requested for the multinomial model
- specify graphical enhancements (such as color or text height)

Summary of Options

Table 71.19 through Table 71.25 list all *options* by function. The "Dictionary of Options" on page 5290 describes each option in detail.

LPRED Options

Table 71.19 Plot Layout Options for LPREDPLOT

LEVEL=(*character-list*)	specifies the names of the levels for which the linear predictor lines are requested (only for the multinomial model)
NOCONF	suppresses fiducial limits (only for the binomial model)
NODATA	suppresses observed data points on the plot (only for the binomial model)
NOTHRESH	suppresses the threshold line
THRESHLABPOS=*value*	specifies the position for the label of the threshold line

General Options

Table 71.20 Color Options

CAXIS=*color*	specifies color for axis
CFIT=*color*	specifies color for fitted curves
CFRAME=*color*	specifies color for frame
CGRID=*color*	specifies color for grid lines
CHREF=*color*	specifies color for HREF= lines
CTEXT=*color*	specifies color for text
CVREF=*color*	specifies color for VREF= lines

Table 71.21 Options to Enhance Plots Produced on Graphics Devices

ANNOTATE=*SAS-data-set*	specifies an Annotate data set
INBORDER	requests a border around plot
LFIT=*linetype*	specifies line style for fitted curves and confidence limits
LGRID=*linetype*	specifies line style for grid lines
NOFRAME	suppresses the frame around plotting areas
NOGRID	suppresses grid lines
NOFIT	suppresses fitted curves
NOHLABEL	suppresses horizontal labels
NOHTICK	suppresses horizontal ticks
NOVTICK	suppresses vertical ticks
TURNVLABELS	vertically strings out characters in vertical labels
WFIT=*n*	specifies thickness for fitted curves
WGRID=*n*	specifies thickness for grids
WREFL=*n*	specifies thickness for reference lines

Table 71.22 Axis Options

HAXIS=*value1 to value2 <by value3>*	specifies tick mark values for horizontal axis
HOFFSET=*value*	specifies offset for horizontal axis
HLOWER=*value*	specifies lower limit on horizontal axis scale
HUPPER=*value*	specifies upper limit on horizontal axis scale
NHTICK=*n*	specifies number of ticks for horizontal axis
NVTICK=*n*	specifies number of ticks for vertical axis
VAXIS=*value1 to value2 <by value3>*	specifies tick mark values for vertical axis
VAXISLABEL=*'label'*	specifies label for vertical axis
VOFFSET=*value*	specifies offset for vertical axis
VLOWER=*value*	specifies lower limit on vertical axis scale
VUPPER=*value*	specifies upper limit on vertical axis scale
WAXIS=*n*	specifies thickness for axis

Table 71.23 Graphics Catalog Options

DESCRIPTION=*'string'*	specifies description for graphics catalog member
NAME=*'string'*	specifies name for plot in graphics catalog

Table 71.24 Options for Text Enhancement

FONT=*font*	specifies software font for text
HEIGHT=*value*	specifies height of text used outside framed areas
INFONT=*font*	specifies software font for text inside framed areas
INHEIGHT=*value*	specifies height of text inside framed areas

Table 71.25 Options for Reference Lines

HREF<(INTERSECT)> =*value-list*	requests horizontal reference line
HREFLABELS= ('label1',...,'labeln')	specifies labels for HREF= lines
HREFLABPOS=*n*	specifies vertical position of labels for HREF= lines
LHREF=*linetype*	specifies line style for HREF= lines
LVREF=*linetype*	specifies line style for VREF= lines
VREF<(INTERSECT)> =*value-list*	requests vertical reference line
VREFLABELS= ('label1',...,'labeln')	specifies labels for VREF= lines
VREFLABPOS=*n*	specifies horizontal position of labels for VREF= lines

Dictionary of Options

The following entries provide detailed descriptions of the *options* in the LPREDPLOT statement.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an Annotate data set, as described in *SAS/GRAPH Software: Reference*, that enables you to add features to the lpred plot. The ANNOTATE= data set you specify in the LPREDPLOT statement is used for all plots created by the statement.

CAXIS=*color*

CAXES=*color*

specifies the color used for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement. The default is the first color in the device color list.

CFIT=*color*

specifies the color for the fitted lpred lines. The default is the first color in the device color list.

CFRAME=*color*

CFR=*color*

specifies the color for the area enclosed by the axes and frame. This area is not shaded by default.

CGRID=*color*

specifies the color for grid lines. The default is the first color in the device color list.

CHREF=*color*

CH=*color*

specifies the color for lines requested by the HREF= option. The default is the first color in the device color list.

CTEXT=*color*

specifies the color for tick mark values and axis labels. The default is the color specified for the CTEXT= option in the most recent GOPTIONS statement.

CVREF=*color*

CV=*color*

specifies the color for lines requested by the VREF= option. The default is the first color in the device color list.

DESCRIPTION=*'string'*

DES=*'string'*

specifies a description, up to 40 characters, that appears in the PROC GREPLAY master menu. The default is the variable name.

FONT=*font*

specifies a software font for reference line and axis labels. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the most recent GOPTIONS statement. Hardware characters are used by default.

HAXIS=*value1 to value2< by value3 >*

specifies tick mark values for the horizontal axis; *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. If *value3* is omitted, a value of 1 is used.

Examples of HAXIS= lists follow:

```
haxis = 0 to 10
haxis = 2 to 10 by 2
haxis = 0 to 200 by 10
```

HEIGHT=*value*

specifies the height of text used outside framed areas. The default value is 3.846 (in percentage).

HLOWER=*value*

specifies the lower limit on the horizontal axis scale. The HLOWER= option specifies *value* as the lower horizontal axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the HAXIS= option is used.

HOFFSET=*value*

specifies offset for horizontal axis. The default value is 1.

HUPPER=*value*

specifies *value* as the upper horizontal axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the HAXIS= option is used.

HREF <(INTERSECT)> =*value-list*

requests reference lines perpendicular to the horizontal axis. If (INTERSECT) is specified, a second reference line perpendicular to the vertical axis is drawn that intersects the fit line at the same point as the horizontal axis reference line. If a horizontal axis reference line label is specified, the intersecting vertical axis reference line is labeled with the vertical axis value. See also the CHREF=, HREFLABELS=, and LHREF= options.

HREFLABELS=*'label1'***,...,***'labeln'*

HREFLABEL=*'label1'***,...,***'labeln'*

HREFLAB=*'label1'***,...,***'labeln'*

specifies labels for the lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

HREFLABPOS=*n*

specifies the vertical position of labels for HREF= lines. The following table shows valid values for *n* and the corresponding label placements.

n	label placement
1	top
2	staggered from top
3	bottom
4	staggered from bottom
5	alternating from top
6	alternating from bottom

INBORDER

requests a border around lpred plots.

LEVEL=(*character-list***)**

ORDINAL=(*character-list***)**

specifies the names of the levels for which linear predictor lines are requested. Names should be quoted and separated by space. If there is no correct name provided, no lpred line is plotted.

LFIT=*linetype*

specifies a line style for fitted curves and confidence limits. By default, fitted curves are drawn by connecting solid lines (*linetype = 1*) and confidence limits are drawn by connecting dashed lines (*linetype = 3*).

LGRID=*linetype*

specifies a line style for all grid lines. The value for *linetype* is between 1 and 46. The default is 35.

LHREF=*linetype*

LH=*linetype*

specifies the line type for lines requested by the HREF= option. The default is 2, which produces a dashed line.

LVREF=*linetype*

LV=*linetype*

specifies the line type for lines requested by the VREF= option. The default is 2, which produces a dashed line.

NAME=*'string'*

specifies a name for the plot, up to eight characters, that appears in the PROC GREPLAY master menu. The default is 'PROBIT'.

NOCONF

suppresses confidence limits from the plot. This works only for the binomial model. Confidence limits are not plotted for the multinomial model.

NODATA

suppresses observed data points from the plot. This works only for the binomial model. Data points are not plotted for the multinomial model.

NOFIT

suppresses the fitted lpred lines.

NOFRAME

suppresses the frame around plotting areas.

NOGRID

suppresses grid lines.

NOHLABEL

suppresses horizontal labels.

NOHTICK

suppresses horizontal tick marks.

NOTHRESH

suppresses the threshold line.

NOVLABEL

suppresses vertical labels.

NOVTICK

suppresses vertical tick marks.

THRESHLABPOS=*n*

specifies the horizontal position of labels for the threshold line. The following table shows valid values for *n* and the corresponding label placements.

n	label placement
1	left
2	right

VAXIS=*value1 to value2< by value3 >*

specifies tick mark values for the vertical axis; *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. This method of specification of tick marks is not valid for logarithmic axes. If *value3* is omitted, a value of 1 is used.

Examples of VAXIS= lists follow:

```
vaxis = 0 to 10
vaxis = 0 to 2 by .1
```

VAXISLABEL=*'string'*

specifies a label for the vertical axis.

VLOWER=*value*

specifies the lower limit on the vertical axis scale. The VLOWER= option specifies *value* as the lower vertical axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

VREF=*value-list*

requests reference lines perpendicular to the vertical axis. If (INTERSECT) is specified, a second reference line perpendicular to the horizontal axis is drawn that intersects the fit line at the same point as the vertical axis reference line. If a vertical axis reference line label is specified, the intersecting horizontal axis reference line is labeled with the horizontal axis value. See also the CVREF=, LVREF=, and VREFLABELS= options.

VREFLABELS=*'label1',...,'labeln'*

VREFLABEL=*'label1',...,'labeln'*

VREFLAB=*'label1',...,'labeln'*

specifies labels for the lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

VREFLABPOS=*n*

specifies the horizontal position of labels for VREF= lines. The following table shows valid values for *n* and the corresponding label placements.

n	label placement
1	left
2	right

VUPPER=*number*
specifies the upper limit on the vertical axis scale. The VUPPER= option specifies *number* as the upper vertical axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

WAXIS=*n*
specifies line thickness for axes and frame. The default value is 1.

WFIT=*n*
specifies line thickness for fitted lines. The default value is 1.

WGRID=*n*
specifies line thickness for grids. The default value is 1.

WREFL=*n*
specifies line thickness for reference lines. The default value is 1.

MODEL Statement

< label: > MODEL *response=effects* < / *options* > **;**

< label: > MODEL *events/trials=effects* < / *options* > **;**

The MODEL statement names the variables used as the response and the independent variables. Additionally, you can specify the distribution used to model the response, as well as other options. Only a single MODEL statement can be used with one invocation of the PROBIT procedure. If multiple MODEL statements are present, only the last is used. Main effects and interaction terms can be specified in the MODEL statement, as in the GLM procedure.

The optional *label*, which must be a valid SAS name, is used to label output from the matching MODEL statement.

The *response* can be a single variable with a value that is used to indicate the level of the observed response. For example, the response might be a variable called Symptoms that takes on the values 'None,' 'Mild,' or 'Severe.' Note that, for dichotomous response variables, the probability of the lower sorted value is modeled by default (see the section "Details: PROBIT Procedure" on page 5309). Because the model fit by the PROBIT procedure requires ordered response levels, you might need to use either the ORDER=DATA option in the PROC PROBIT statement or a numeric coding of the response to get the desired ordering of levels.

Alternatively, the response can be specified as a pair of variable names separated by a slash (/). The value of the first variable, *events*, is the number of positive responses (or events). The value of the second variable, *trials*, is the number of trials. Both variables must be numeric and nonnegative, and the ratio of the first variable value to the second variable value must be between 0 and 1, inclusive. For example, the variables might be hits, a variable containing the number of hits for a baseball player, and AtBats, a variable containing the number of times at bat. A model for hitting proportion (batting average) as a function of age could be specified as

```
model hits/AtBats=age;
```

The *effects* following the equal sign are the covariates in the model. Higher-order effects, such as interactions and nested terms, are allowed in the list, as in the GLM procedure. Variable names and combinations of variable names representing higher-order terms are allowed to appear in this list. Classification variables can be used as effects, and indicator variables are generated for the class levels. If you do not specify any covariates following the equal sign, an intercept-only model is fit.

The following options are available in the MODEL statement.

AGGREGATE

AGGREGATE=*variable-list*

specifies the subpopulations on which the Pearson chi-square test statistic and the log-likelihood ratio chi-square test statistic (deviance) are calculated if the LACKFIT option is specified. See the section "Rescaling the Covariance Matrix" on page 5314 for details of Pearson's chi-square and deviance calculations.

Observations with common values in the given list of variables are regarded as coming from the same subpopulation. Variables in the list can be any variables in the input data set. Specifying the AGGREGATE option is equivalent to specifying the AGGREGATE= option with a variable list that includes all independent variables in the MODEL statement. The PROBIT procedure sorts the input data set according to the variables specified in this list. Information for the sorted data set is reported in the "Response-Covariate Profile" table.

The deviance and Pearson goodness-of-fit statistics are calculated if the LACKFIT option is specified in the MODEL statement. The calculated results are reported in the "Goodness-of-Fit" table. If the Pearson chi-square test is significant with the test level specified by the HPROB= option, the fiducial limits, if required with the INVERSECL option in the MODEL statement, are modified (see the section "Inverse Confidence Limits" on page 5315 for details). Also, the covariance matrix is rescaled by the dispersion parameter when the SCALE= option is specified.

ALPHA=*value*

sets the significance level for the confidence intervals for regression parameters, fiducial limits for the predicted values, and confidence intervals for the predicted probabilities. The value must be between 0 and 1. The default value is ALPHA=0.05.

CONVERGE=*value*

specifies the convergence criterion. Convergence is declared when the maximum change in the parameter estimates between Newton-Raphson steps is less than the value specified. The change is a relative change if the parameter is greater than 0.01 in absolute value; otherwise, it is an absolute change.

By default, CONVERGE=1.0E−8.

CORRB

displays the estimated correlation matrix of the parameter estimates.

COVB

displays the estimated covariance matrix of the parameter estimates.

DISTRIBUTION=*distribution-type*

DIST=*distribution-type*

D=*distribution-type*

specifies the cumulative distribution function used to model the response probabilities. The distributions are described in the section "Details: PROBIT Procedure" on page 5309. Valid values for *distribution-type* are as follows:

NORMAL the normal distribution for the probit model

LOGISTIC the logistic distribution for the logit model

EXTREMEVALUE | EXTREME | GOMPERTZ the extreme value, or Gompertz distribution for the gompit model

By default, DISTRIBUTION=NORMAL.

HPROB=*p*

specifies a minimum probability level for the Pearson chi-square to indicate a good fit. The default value is 0.10. The LACKFIT option must also be specified for this option to have any effect. For Pearson goodness-of-fit chi-square values with probability greater than the HPROB= value, the fiducial limits, if requested with the INVERSECL option, are computed by using a critical value of 1.96. For chi-square values with probability less than the value of the HPROB= option, the critical value is a 0.95 two-sided quantile value taken from the t distribution with degrees of freedom equal to $(k-1)\times m-q$, where k is the number of levels for the response variable, m is the number of different sets of independent variable values, and q is the number of parameters fit in the model. If you specify the HPROB= option in both the PROC PROBIT and MODEL statements, the MODEL statement option takes precedence.

INITIAL=*values*

sets initial values for the parameters in the model other than the intercept. The values must be given in the order in which the variables are listed in the MODEL statement. If some of the independent variables listed in the MODEL statement are classification variables, then there must be as many values given for that variable as there are classification levels minus 1. The INITIAL option can be specified as follows.

Type of List	Specification
list separated by blanks	`initial=3 4 5`
list separated by commas	`initial=3,4,5`

By default, all parameters have initial estimates of zero.

NOTE: The INITIAL= option is overwritten by the INEST= option in the PROC PROBIT statement.

INTERCEPT=*value*

initializes the intercept parameter to *value*. By default, INTERCEPT=0.

INVERSECL

computes confidence limits for the values of the first continuous independent variable (such as dose) that yield selected response rates. If the algorithm fails to converge (this can happen

when C is nonzero), missing values are reported for the confidence limits. See the section "Inverse Confidence Limits" on page 5315 for details.

ITPRINT

displays the iteration history, the final evaluation of the gradient, and the second derivative matrix (Hessian).

LACKFIT

performs two goodness-of-fit tests (a Pearson chi-square test and a log-likelihood ratio chi-square test) for the fitted model.

To compute the test statistics, proper grouping of the observations into subpopulations is needed. You can use the AGGREGATE or AGGREGATE= option for this purpose. See the entry for the AGGREGATE and AGGREGATE= options under the MODEL statement. If neither AGGREGATE nor AGGREGATE= is specified, PROC PROBIT assumes each observation is from a separate subpopulation and computes the goodness-of-fit test statistics only for the *events/trials* syntax.

NOTE: This test is not appropriate if the data are very sparse, with only a few values at each set of the independent variable values.

If the Pearson chi-square test statistic is significant, then the covariance estimates and standard error estimates are adjusted. See the section "Lack-of-Fit Tests" on page 5313 for a description of the tests. Note that the LACKFIT option can also appear in the PROC PROBIT statement. See the section "PROC PROBIT Statement" on page 5264 for details.

MAXITER=*value*

MAXIT=*value*

specifies the maximum number of iterations to be performed in estimating the parameters. By default, MAXITER=50.

NOINT

fits a model with no intercept parameter. If the INTERCEPT= option is also specified, the intercept is fixed at the specified value; otherwise, it is set to zero. This is most useful when the response is binary. When the response has k levels, then $k-1$ intercept parameters are fit. The NOINT option sets the intercept parameter corresponding to the lowest response level equal to zero. A Lagrange multiplier, or score, test for the restricted model is computed when the NOINT option is specified.

SCALE=*scale*

enables you to specify the method for estimating the dispersion parameter. To correct for overdispersion or underdispersion, the covariance matrix is multiplied by the estimate of the dispersion parameter. Valid values for *scale* are as follows:

D | DEVIANCE	specifies that the dispersion parameter be estimated by the deviance divided by its degrees of freedom.
P | PEARSON	specifies that the dispersion parameter be estimated by the Pearson chi-square statistic divided by its degrees of freedom. This is set as the default method for estimating the dispersion parameter.

You can use the AGGREGATE= option to define the subpopulations for calculating the Pearson chi-square statistic and the deviance.

The "Goodness-of-Fit " table includes the Pearson chi-square statistic, the deviance, their degrees of freedom, the ratio of each statistic divided by its degrees of freedom, and the corresponding p-value.

SINGULAR=*value*
specifies the singularity criterion for determining linear dependencies in the set of independent variables. The sum of squares and crossproducts matrix of the independent variables is formed and swept. If the relative size of a pivot becomes less than the value specified, then the variable corresponding to the pivot is considered to be linearly dependent on the previous set of variables considered. By default, SINGULAR=1E−12.

OUTPUT Statement

OUTPUT < *OUT=SAS-data-set* > *keyword=name* < . . . *keyword=name* > **;**

The OUTPUT statement creates a new SAS data set containing all variables in the input data set and, optionally, the fitted probabilities, the estimate of $\mathbf{x}'\beta$, and the estimate of its standard error. Estimates of the probabilities, $\mathbf{x}'\beta$, and the standard errors are computed for observations with missing response values as long as the values of all the explanatory variables are nonmissing. This enables you to compute these statistics for additional settings of the explanatory variables that are of interest but for which responses are not observed.

You can specify multiple OUTPUT statements. Each OUTPUT statement creates a new data set and applies only to the preceding MODEL statement. If you want to create a permanent SAS data set, you must specify a two-level name (refer to *SAS Language Reference: Concepts* for more information about permanent SAS data sets).

Details on the specifications in the OUTPUT statement are as follows:

keyword=name specifies the statistics to include in the output data set and assigns names to the new variables that contain the statistics. Specify a keyword for each desired statistic (see the following list of keywords), an equal sign, and the variable to contain the statistic.

The keywords allowed and the statistics they represent are as follows:

PROB | P cumulative probability estimates

$$p = C + (1 - C)F(a_j + \mathbf{x}'\beta)$$

STD standard error estimates of $a_j + \mathbf{x}'\mathbf{b}$

XBETA estimates of $a_j + \mathbf{x}'\beta$

OUT=*SAS-data-set* names the output data set. By default, the new data set is named by using the DATA*n* convention.

When the *single variable response* syntax is used, the _LEVEL_ variable is added to the output data set, and there are $k-1$ output observations for each input observation, where k is the number of response levels. There is no observation output corresponding to the highest response level. For each of the $k-1$ observations, the PROB variable contains the fitted probability of obtaining a response level up to the level indicated by the _LEVEL_ variable, the XBETA variable contains $a_j + \mathbf{x}'\mathbf{b}$, where j references the levels ($a_1 = 0$), and the STD variable contains the standard error estimate of the XBETA variable. See the section "Details: PROBIT Procedure" on page 5309 for the formulas for the parameterizations.

PREDPPLOT Statement

PREDPPLOT < *var = variable* > < *options* > **;**

The PREDPPLOT statement plots the predicted probability against a single continuous variable (dose variable) in the MODEL statement for both the binomial model and the multinomial model. Confidence limits are available only for the binomial model. An attached box on the right side of the plot is used to label predicted probability curves with the names of their levels for the multinomial model. You can specify the color of this box by using the CLABBOX= option.

VAR=*variable*

specifies a single continuous variable (dose variable) in the independent variable list of the MODEL statement. If a VAR= variable is not specified, the first single continuous variable in the independent variable list of the MODEL statement is used. If such a variable does not exist in the independent variable list of the MODEL statement, an error is reported.

The predicted probability is

$$\hat{p} = C + (1-C)F(\mathbf{x}'\hat{\mathbf{b}})$$

for the binomial model and

$$\begin{aligned}
\hat{p}_1 &= C + (1-C)F(\mathbf{x}'\hat{\mathbf{b}}) \\
\hat{p}_j &= (1-C)(F(\hat{a}_j + \mathbf{x}'\hat{\mathbf{b}}) - F(\hat{a}_{j-1} + \mathbf{x}'\hat{\mathbf{b}})) \quad j = 2, \ldots, k-1 \\
\hat{p}_k &= (1-C)(1 - F(\hat{a}_{k-1} + \mathbf{x}'\hat{\mathbf{b}}))
\end{aligned}$$

for the multinomial model with k response levels, where F is the cumulative distribution function used to model the probability, $\mathbf{x}'$ is the vector of the covariates, $\hat{a}_j$ are the estimated ordinal intercepts with $\hat{a}_1 = 0$, C is the threshold parameter, either known or estimated from the model, and $\hat{\mathbf{b}}'$ is the vector of estimated parameters.

To plot $\hat{p}$ (or $\hat{p}_j$) as a function of a continuous variable x_1, the remaining covariates $\mathbf{x}_{-1}$ must be specified. You can use the XDATA= option to provide the values of $\mathbf{x}_{-1}$ (see the XDATA= option in the PROC PROBIT statement for details), or use the default values that follow these rules:

- If the effect contains a continuous variable (or variables), the overall mean of this effect is used.
- If the effect is a single classification variable, the highest level of the variable is used.

options

enable you to plot the observed data and add features to the plot.

You can use options in the PREDPPLOT statement to do the following:

- superimpose specification limits
- suppress or add observed data points for the binomial model
- suppress or add confidence limits for the binomial model
- specify the levels for which predicted probability curves are requested for the multinomial model
- specify graphical enhancements (such as color or text height)

Summary of Options

Table 71.26 through Table 71.32 list all *options* by function. The "Dictionary of Options" on page 5304 describes each option in detail.

PREDPPLOT Options

Table 71.26 Plot Layout Options for PREDPPLOT

LEVEL=(*character-list*)	specifies the names of the levels for which the predicted probability curves are requested (only for the multinomial model)
NOCONF	suppresses confidence limits
NODATA	suppresses observed data points on the plot
NOTHRESH	suppresses the threshold line
THRESHLABPOS=*value*	specifies the position for the label of the threshold line

General Options

Table 71.27 Color Options

CAXIS=*color*	specifies color for the axes
CFIT=*color*	specifies color for fitted curves
CFRAME=*color*	specifies color for frame
CGRID=*color*	specifies color for grid lines
CHREF=*color*	specifies color for HREF= lines
CLABBOX=*color*	specifies color for label box
CTEXT=*color*	specifies color for text
CVREF=*color*	specifies color for VREF= lines

Table 71.28 Options to Enhance Plots Produced on Graphics Devices

ANNOTATE=*SAS-data-set*	specifies an Annotate data set
INBORDER	requests a border around plot
LFIT=*linetype*	specifies line style for fitted curves and confidence limits
LGRID=*linetype*	specifies line style for grid lines
NOFRAME	suppresses the frame around plotting areas
NOGRID	suppresses grid lines
NOFIT	suppresses fitted curves
NOHLABEL	suppresses horizontal labels
NOHTICK	suppresses horizontal ticks
NOVTICK	suppresses vertical ticks
TURNVLABELS	vertically strings out characters in vertical labels
WFIT=*n*	specifies thickness for fitted curves
WGRID=*n*	specifies thickness for grids
WREFL=*n*	specifies thickness for reference lines

Table 71.29 Axis Options

HAXIS=*value1 to value2 <by value3>*	specifies tick mark values for horizontal axis
HOFFSET=*value*	specifies offset for horizontal axis
HLOWER=*value*	specifies lower limit on horizontal axis scale
HUPPER=*value*	specifies upper limit on horizontal axis scale
NHTICK=*n*	specifies number of ticks for horizontal axis
NVTICK=*n*	specifies number of ticks for vertical axis
VAXIS=*value1 to value2 <by value3>*	specifies tick mark values for vertical axis
VAXISLABEL=*'label'*	specifies label for vertical axis
VOFFSET=*value*	specifies offset for vertical axis
VLOWER=*value*	specifies lower limit on vertical axis scale
VUPPER=*value*	specifies upper limit on vertical axis scale
WAXIS=*n*	specifies thickness for axis

Table 71.30 Graphics Catalog Options

DESCRIPTION=*'string'*	specifies description for graphics catalog member
NAME=*'string'*	specifies name for plot in graphics catalog

Table 71.31 Options for Text Enhancement

FONT=*font*	specifies software font for text
HEIGHT=*value*	specifies height of text used outside framed areas
INFONT=*font*	specifies software font for text inside framed areas
INHEIGHT=*value*	specifies height of text inside framed areas

Table 71.32 Options for Reference Lines

HREF<(INTERSECT)> =*value-list*	requests horizontal reference line
HREFLABELS= ('label1',...,'labeln')	specifies labels for HREF= lines
HREFLABPOS=*n*	specifies vertical position of labels for HREF= lines
LHREF=*linetype*	specifies line style for HREF= lines
LVREF=*linetype*	specifies line style for VREF= lines
VREF<(INTERSECT)> =*value-list*	requests vertical reference line
VREFLABELS= ('label1',...,'labeln')	specifies labels for VREF= lines
VREFLABPOS=*n*	specifies horizontal position of labels for VREF= lines

Dictionary of Options

The following entries provide detailed descriptions of the *options* in the PREDPPLOT statement.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an Annotate data set, as described in *SAS/GRAPH Software: Reference*, that enables you to add features to the predicted probability plot. The ANNOTATE= data set you specify in the PREDPPLOT statement is used for all plots created by the statement.

CAXIS=*color*

CAXES=*color*

specifies the color used for the axes and tick marks. This option overrides any COLOR= specifications in an AXIS statement. The default is the first color in the device color list.

CFIT=*color*

specifies the color for the fitted predicted probability curves. The default is the first color in the device color list.

CFRAME=*color*

CFR=*color*

specifies the color for the area enclosed by the axes and frame. This area is not shaded by default.

CGRID=*color*

specifies the color for grid lines. The default is the first color in the device color list.

CHREF=*color*

CH=*color*

specifies the color for lines requested by the HREF= option. The default is the first color in the device color list.

CTEXT=*color*

specifies the color for tick mark values and axis labels. The default is the color specified for the CTEXT= option in the most recent GOPTIONS statement.

CVREF=*color*

CV=*color*

specifies the color for lines requested by the VREF= option. The default is the first color in the device color list.

DESCRIPTION=*'string'*

DES=*'string'*

specifies a description, up to 40 characters, that appears in the PROC GREPLAY master menu. The default is the variable name.

FONT=*font*

specifies a software font for reference line and axis labels. You can also specify fonts for axis labels in an AXIS statement. The FONT= font takes precedence over the FTEXT= font specified in the most recent GOPTIONS statement. Hardware characters are used by default.

HAXIS=*value1 to value2< by value3 >*

specifies tick mark values for the horizontal axis; *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. If *value3* is omitted, a value of 1 is used.

Examples of HAXIS= lists follow:

```
haxis = 0 to 10
haxis = 2 to 10 by 2
haxis = 0 to 200 by 10
```

HEIGHT=*value*

specifies the height of text used outside framed areas.

HLOWER=*value*

specifies the lower limit on the horizontal axis scale. The HLOWER= option specifies *value* as the lower horizontal axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the HAXIS= option is used.

HOFFSET=*value*

specifies the offset for the horizontal axis. The default value is 1.

HUPPER=*value*

specifies *value* as the upper horizontal axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the HAXIS= option is used.

HREF < (INTERSECT) > =*value-list*

requests reference lines perpendicular to the horizontal axis. If (INTERSECT) is specified, a second reference line perpendicular to the vertical axis is drawn that intersects the fit line at

the same point as the horizontal axis reference line. If a horizontal axis reference line label is specified, the intersecting vertical axis reference line is labeled with the vertical axis value. See also the CHREF=, HREFLABELS=, and LHREF= options.

HREFLABELS=*'label1',...,'labeln'*

HREFLABEL=*'label1',...,'labeln'*

HREFLAB=*'label1',...,'labeln'*

specifies labels for the lines requested by the HREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

HREFLABPOS=*n*

specifies the vertical position of labels for HREF= lines. The following table shows valid values for *n* and the corresponding label placements.

n	label placement
1	top
2	staggered from top
3	bottom
4	staggered from bottom
5	alternating from top
6	alternating from bottom

INBORDER

requests a border around predicted probability plots.

LEVEL=(*character-list***)**

ORDINAL= (*character-list***)**

specifies the names of the levels for which predicted probability curves are requested. Names should be quoted and separated by space. If there is no correct name provided, no fitted probability curve is plotted.

LFIT=*linetype*

specifies a line style for fitted curves and confidence limits. By default, fitted curves are drawn by connecting solid lines (*linetype = 1*) and confidence limits are drawn by connecting dashed lines (*linetype = 3*).

LGRID=*linetype*

specifies a line style for all grid lines. The value for *linetype* is between 1 and 46. The default is 35.

LHREF=*linetype*

LH=*linetype*

specifies the line type for lines requested by the HREF= option. The default is 2, which produces a dashed line.

LVREF=*linetype*

LV=*linetype*

specifies the line type for lines requested by the VREF= option. The default is 2, which produces a dashed line.

NAME=*'string'*

specifies a name for the plot, up to eight characters, that appears in the PROC GREPLAY master menu. The default is 'PROBIT'.

NOCONF

suppresses confidence limits from the plot. This works only for the binomial model. Confidence limits are not plotted for the multinomial model.

NODATA

suppresses observed data points from the plot. This works only for the binomial model. The data points are not plotted for the multinomial model.

NOFIT

suppresses the fitted predicted probability curves.

NOFRAME

suppresses the frame around plotting areas.

NOGRID

suppresses grid lines.

NOHLABEL

suppresses horizontal labels.

NOHTICK

suppresses horizontal tick marks.

NOTHRESH

suppresses the threshold line.

NOVLABEL

suppresses vertical labels.

NOVTICK

suppresses vertical tick marks.

THRESHLABPOS=*n*

specifies the horizontal position of labels for the threshold line. The following table shows valid values for *n* and the corresponding label placements.

n	label placement
1	left
2	right

VAXIS=*value1 to value2< by value3 >*

specifies tick mark values for the vertical axis; *value1*, *value2*, and *value3* must be numeric, and *value1* must be less than *value2*. The lower tick mark is *value1*. Tick marks are drawn at increments of *value3*. The last tick mark is the greatest value that does not exceed *value2*. This method of specification of tick marks is not valid for logarithmic axes. If *value3* is omitted, a value of 1 is used.

Examples of VAXIS= lists follow:

```
vaxis = 0 to 10
vaxis = 0 to 2 by .1
```

VAXISLABEL=*'string'*
specifies a label for the vertical axis.

VLOWER=*value*
specifies the lower limit on the vertical axis scale. The VLOWER= option specifies *value* as the lower vertical axis tick mark. The tick mark interval and the upper axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

VREF=*value-list*
requests reference lines perpendicular to the vertical axis. If (INTERSECT) is specified, a second reference line perpendicular to the horizontal axis is drawn that intersects the fit line at the same point as the vertical axis reference line. If a vertical axis reference line label is specified, the intersecting horizontal axis reference line is labeled with the horizontal axis value. See also the CVREF=, LVREF=, and VREFLABELS= options.

VREFLABELS=*'label1',...,'labeln'*

VREFLABEL=*'label1',...,'labeln'*

VREFLAB=*'label1',...,'labeln'*
specifies labels for the lines requested by the VREF= option. The number of labels must equal the number of lines. Enclose each label in quotes. Labels can be up to 16 characters.

VREFLABPOS=*n*
specifies the horizontal position of labels for VREF= lines. The following table shows valid values for *n* and the corresponding label placements.

n	label placement
1	left
2	right

VUPPER=*value*
specifies the upper limit on the vertical axis scale. The VUPPER= option specifies *value* as the upper vertical axis tick mark. The tick mark interval and the lower axis limit are determined automatically. This option has no effect if the VAXIS= option is used.

WAXIS=*n*
specifies line thickness for axes and frame. The default value is 1.

WFIT=*n*
specifies line thickness for fitted curves. The default value is 1.

WGRID=*n*
specifies line thickness for grids. The default value is 1.

WREFL=*n*
specifies line thickness for reference lines. The default value is 1.

WEIGHT Statement

WEIGHT *variable* **;**

A WEIGHT statement can be used with PROC PROBIT to weight each observation by the value of the variable specified. The contribution of each observation to the likelihood function is multiplied by the value of the weight variable. Observations with zero, negative, or missing weights are not used in model estimation.

Details: PROBIT Procedure

Missing Values

PROC PROBIT does not use any observations having missing values for any of the independent variables, the response variables, or the weight variable. If only the response variables are missing, statistics requested in the OUTPUT statement are computed.

Response Level Ordering

For binary response data, PROC PROBIT fits the following model by default:

$$\Phi^{-1}\left(\frac{p-C}{1-C}\right) = \mathbf{x}'\beta$$

where p is the probability of the response level identified as the first level in the "Weighted Frequency Counts for the Ordered Response Categories" table in the output and Φ is the normal cumulative distribution function. By default, the covariate vector $\mathbf{x}$ contains an intercept term. This is sometimes called Abbot's formula.

Because of the symmetry of the normal (and logistic) distribution, the effect of reversing the order of the two response values is to change the signs of β in the preceding equation.

By default, response levels appear in ascending, sorted order (that is, the lowest level appears first, and then the next lowest, and so on). There are a number of ways that you can control the sort order of the response categories and, therefore, which level is assigned the first ordered level. One of the most common sets of response levels is {0,1}, with 1 representing the event with the probability that is to be modeled.

Consider the example where Y takes the values 1 and 0 for event and nonevent, respectively, and EXPOSURE is the explanatory variable. By default, PROC PROBIT assigns the first ordered level

to response level 0, causing the probability of the nonevent to be modeled. There are several ways to change this.

Besides recoding the variable Y, you can do the following:

- assign a format to Y such that the first formatted value (when the formatted values are put in sorted order) corresponds to the event. For the following example, Y=0 could be assigned formatted value 'nonevent' and Y=1 could be assigned formatted value 'event.' Since ORDER=FORMATTED by default, Y=1 becomes the first ordered level. See Example 71.3 for an illustration of this method.

```
proc format;
   value disease 1='event' 0='nonevent';
run;
proc probit;
   model y=exposure;
   format y disease.;
run;
```

- arrange the input data set so that Y=1 appears first and use the ORDER=DATA option in the PROC PROBIT statement. Since ORDER=DATA sorts levels in order of their appearance in the data set, Y=1 becomes the first ordered level. Note that this option causes classification variables to be sorted by their order of appearance in the data set, also.

Computational Method

The log-likelihood function is maximized by means of a ridge-stabilized Newton-Raphson algorithm. Initial regression parameter estimates are set to zero. The INITIAL= and INTERCEPT= options in the MODEL statement can be used to give nonzero initial estimates.

The log-likelihood function, L, is computed as

$$L = \sum_i w_i \ln(p_i)$$

where the sum is over the observations in the data set, w_i is the weight for the ith observation, and p_i is the modeled probability of the observed response. In the case of the events/trials syntax in the MODEL statement, each observation contributes two terms corresponding to the probability of the event and the probability of its complement:

$$L = \sum_i w_i [r_i \ln(p_i) + (n_i - r_i) \ln(1 - p_i)]$$

where r_i is the number of events and n_i is the number of trials for observation i. This log-likelihood function differs from the log-likelihood function for a binomial or multinomial distribution by additive terms consisting of the log of binomial or multinomial coefficients. These terms are parameter-independent and do not affect the model estimation or the standard errors and tests.

The estimated covariance matrix, $\mathbf{V}$, of the parameter estimates is computed as the negative inverse of the information matrix of second derivatives of L with respect to the parameters evaluated at

the final parameter estimates. Thus, the estimated covariance matrix is derived from the observed information matrix rather than the expected information matrix (these are generally not the same). The standard error estimates for the parameter estimates are taken as the square roots of the corresponding diagonal elements of **V**.

If convergence of the maximum likelihood estimates is attained, a Type III chi-square test statistic is computed for each effect, testing whether there is any contribution from any of the levels of the effect. This statistic is computed as a quadratic form in the appropriate parameter estimates by using the corresponding submatrix of the asymptotic covariance matrix estimate. Refer to Chapter 39, "The GLM Procedure," and Chapter 15, "The Four Types of Estimable Functions," for more information about Type III estimable functions.

The asymptotic covariance matrix is computed as the inverse of the observed information matrix. Note that if the NOINT option is specified and classification variables are used, the first classification variable contains a contribution from an intercept term. The results are displayed in an ODS table named Type3Analysis.

Chi-square tests for individual parameters are Wald tests based on the observed information matrix and the parameter estimates. If an effect has a single degree of freedom in the parameter estimates table, the chi-square test for this parameter is equivalent to the Type III test for this effect.

Prior to SAS 8.2, a multiple-degrees-of-freedom statistic was computed for each effect to test for contribution from any level of the effect. In general, the Type III test statistic in a main-effect-only model (no interaction terms) will be equal to the previously computed effect statistic, unless there are collinearities among the effects. If there are collinearities, the Type III statistic will adjust for them, and the value of the Type III statistic and the number of degrees of freedom might not be equal to those of the previous effect statistic.

The theory behind these tests assumes large samples. If the samples are not large, it might be better to base the tests on log-likelihood ratios. These changes in log likelihood can be obtained by fitting the model twice, once with all the parameters of interest and once leaving out the parameters to be tested. Refer to Cox and Oakes (1984) for a discussion of the merits of some possible test methods.

If some of the independent variables are perfectly correlated with the response pattern, then the theoretical parameter estimates can be infinite. Although fitted probabilities of 0 and 1 are not especially pathological, infinite parameter estimates are required to yield these probabilities. Due to the finite precision of computer arithmetic, the actual parameter estimates are not infinite. Indeed, since the tails of the distributions allowed in the PROBIT procedure become small rapidly, an argument to the cumulative distribution function of around 20 becomes effectively infinite. In the case of such parameter estimates, the standard error estimates and the corresponding chi-square tests are not trustworthy.

Distributions

The distributions, $F(x)$, allowed in the PROBIT procedure are specified with the DISTRIBUTION= option in the MODEL statement. The cumulative distribution functions for the available distributions are

$$\int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right) dz \text{ (normal)}$$

$$\frac{1}{1+e^{-x}} \text{(logistic)}$$

$$1 - e^{-e^{x}} \text{(extreme value or Gompertz)}$$

The variances of these three distributions are not all equal to 1, and their means are not all equal to zero. Their means and variances are shown in the following table, where γ is the Euler constant.

Distribution	Mean	Variance
Normal	0	1
Logistic	0	$\pi^2/3$
Extreme value or Gompertz	$-\gamma$	$\pi^2/6$

When comparing parameter estimates by using different distributions, you need to take into account the different scalings and, for the extreme value (or Gompertz) distribution, a possible shift in location. For example, if the fitted probabilities are in the neighborhood of 0.1 to 0.9, then the parameter estimates from the logistic model should be about $\pi/\sqrt{3}$ larger than the estimates from the probit model.

INEST= *SAS-data-set*

The INEST= data set names a SAS data set that specifies initial estimates for all the parameters in the model.

The INEST= data set must contain the intercept variables (named Intercept for binary response model and Intercept, Intercept2, Intercept3, and so forth, for multinomial response models) and all independent variables in the MODEL statement.

If BY processing is used, the INEST= data set should also include the BY variables, and there must be at least one observation for each BY group. If there is more than one observation in a BY group, the first one read is used for that BY group.

If the INEST= data set also contains the _TYPE_ variable, only observations with the _TYPE_ value "PARMS" are used as starting values. Combining the INEST= data set and the option MAXIT= in the MODEL statement, partial scoring can be done, such as predicting on a validation data set by using the model built from a training data set.

You can specify starting values for the iterative algorithm in the INEST= data set. This data set overwrites the INITIAL= option in the MODEL statement, which is a little difficult to use for models with multilevel interaction effects. The INEST= data set has the same structure as the "OUTEST= *SAS-data-set*" on page 5316, but it is not required to have all the variables or observations that appear in the OUTEST= data set. One simple use of the INEST= option is passing the previous OUTEST= data set directly to the next model as an INEST= data set, assuming that the two models have the same parameterization.

Model Specification

For a two-level response, the probability that the lesser response occurs is modeled by the probit equation as

$$p = C + (1 - C)F(\mathbf{x}'\mathbf{b})$$

The probability of the other (complementary) event is $1 - p$.

For a multilevel response with outcomes labeled l_i for $i = 1, 2, \ldots, k$, the probability, p_j, of observing level l_j is as follows:

$$\begin{aligned}
p_1 &= C + (1 - C)F(\mathbf{x}'\mathbf{b}) \\
p_2 &= (1 - C)\left(F(a_2 + \mathbf{x}'\mathbf{b}) - F(\mathbf{x}'\mathbf{b})\right) \\
&\vdots \\
p_j &= (1 - C)\left(F(a_j + \mathbf{x}'\mathbf{b}) - F(a_{j-1} + \mathbf{x}'\mathbf{b})\right) \\
&\vdots \\
p_k &= (1 - C)(1 - F(a_{k-1} + \mathbf{x}'\mathbf{b}))
\end{aligned}$$

Thus, for a k-level response, there are $k - 2$ additional parameters, $a_2, a_3, \ldots, a_{k-1}$, estimated. These parameters are denoted by Intercept$j$, $j = 2, 3, \ldots, k - 1$, in the output.

An intercept parameter is always added to the set of independent variables as the first term in the model unless the NOINT option is specified in the MODEL statement. If a classification variable taking on k levels is used as one of the independent variables, a set of k indicator variables is generated to model the effect of this variable. Because of the presence of the intercept term, there are at most $k - 1$ degrees of freedom for this effect in the model.

Lack-of-Fit Tests

Two goodness-of-fit tests can be requested from the PROBIT procedure: a Pearson chi-square test and a log-likelihood ratio chi-square test.

To compute the test statistics, you can use the AGGREGATE or AGGREGATE= option grouping the observations into subpopulations. If neither AGGREGATE nor AGGREGATE= is specified, PROC PROBIT assumes that each observation is from a separate subpopulation and computes the goodness-of-fit test statistics only for the *events/trials* syntax.

If the Pearson goodness-of-fit chi-square test is requested and the p-value for the test is too small, variances and covariances are adjusted by a heterogeneity factor (the goodness-of-fit chi-square

divided by its degrees of freedom) and a critical value from the t distribution is used to compute the fiducial limits. The Pearson chi-square test statistic is computed as

$$\chi^2_P = \sum_{i=1}^{m}\sum_{j=1}^{k}\frac{(r_{ij} - n_i\hat{p}_{ij})^2}{n_i\hat{p}_{ij}}$$

where the sum on i is over grouping, the sum on j is over levels of response, r_{ij} is the frequency of response level j for the ith grouping, n_i is the total frequency for the ith grouping, and $\hat{p}_{ij}$ is the fitted probability for the jth level at the ith grouping.

The likelihood ratio chi-square test statistic is computed as

$$\chi^2_D = 2\sum_{i=1}^{m}\sum_{j=1}^{k} r_{ij}\ln\left(\frac{r_{ij}}{n_i\hat{p}_{ij}}\right)$$

This quantity is sometimes called the deviance. If the modeled probabilities fit the data, these statistics should be approximately distributed as chi-square with degrees of freedom equal to $(k-1)\times m - q$, where k is the number of levels of the multinomial or binomial response, m is the number of sets of independent variable values (covariate patterns), and q is the number of parameters fit in the model.

In order for the Pearson statistic and the deviance to be distributed as chi-square, there must be sufficient replication within the groupings. When this is not true, the data are sparse, and the p-values for these statistics are not valid and should be ignored. Similarly, these statistics, divided by their degrees of freedom, cannot serve as indicators of overdispersion. A large difference between the Pearson statistic and the deviance provides some evidence that the data are too sparse to use either statistic.

Rescaling the Covariance Matrix

One way of correcting overdispersion is to multiply the covariance matrix by a dispersion parameter. You can supply the value of the dispersion parameter directly, or you can estimate the dispersion parameter based on either the Pearson chi-square statistic or the deviance for the fitted model.

The Pearson chi-square statistic χ^2_P and the deviance χ^2_D are defined in the section "Lack-of-Fit Tests" on page 5313. If the SCALE= option is specified in the MODEL statement, the dispersion parameter is estimated by

$$\widehat{\sigma}^2 = \begin{cases} \chi^2_P/(m(k-1)-q) & \text{SCALE=PEARSON} \\ \chi^2_D/(m(k-1)-q) & \text{SCALE=DEVIANCE} \\ (constant)^2 & \text{SCALE=}constant \end{cases}$$

In order for the Pearson statistic and the deviance to be distributed as chi-square, there must be sufficient replication within the subpopulations. When this is not true, the data are sparse, and the p-values for these statistics are not valid and should be ignored. Similarly, these statistics, divided by their degrees of freedom, cannot serve as indicators of overdispersion. A large difference between

the Pearson statistic and the deviance provides some evidence that the data are too sparse to use either statistic.

You can use the AGGREGATE (or AGGREGATE=) option to define the subpopulation profiles. If you do not specify this option, each observation is regarded as coming from a separate subpopulation. For *events/trials* syntax, each observation represents n Bernoulli trials, where n is the value of the *trials* variable; for *single-trial* syntax, each observation represents a single trial. Without the AGGREGATE (or AGGREGATE=) option, the Pearson chi-square statistic and the deviance are calculated only for *events/trials* syntax.

Note that the parameter estimates are not changed by this method. However, their standard errors are adjusted for overdispersion, affecting their significance tests.

Tolerance Distribution

For a single independent variable, such as a dosage level, the models for the probabilities can be justified on the basis of a population with mean μ and scale parameter σ of tolerances for the subjects. Then, given a dose x, the probability, P, of observing a response in a particular subject is the probability that the subject's tolerance is less than the dose or

$$P = F\left(\frac{x-\mu}{\sigma}\right)$$

Thus, in this case, the intercept parameter, b_0, and the regression parameter, b_1, are related to μ and σ by

$$b_0 = -\frac{\mu}{\sigma},\ b_1 = \frac{1}{\sigma}$$

NOTE: The parameter σ is not equal to the standard deviation of the population of tolerances for the logistic and extreme value distributions.

Inverse Confidence Limits

In bioassay problems, estimates of the values of the independent variables that yield a desired response are often needed. For instance, the value yielding a 50% response rate (called the ED50 or LD50) is often used. The INVERSECL option requests that confidence limits be computed for the value of the independent variable that yields a specified response. These limits are computed only for the first continuous variable effect in the model. The other variables are set either at their mean values if they are continuous or at the reference (last) level if they are discrete variables. For a discussion of inverse confidence limits, see Hubert, Bohidar, and Peace (1988).

For the PROBIT procedure, the response variable is a probability. An estimate of the first continuous variable value needed to achieve a response of p is given by

$$\hat{x}_1 = \frac{1}{b_1}\left(F^{-1}(p) - \mathbf{x}^{*\prime}\mathbf{b}^*\right)$$

where F is the cumulative distribution function used to model the probability, $\mathbf{x}^*$ is the vector of independent variables excluding the first one, which can be specified by the XDATA= option described in the section "XDATA= *SAS-data-set*" on page 5317, $\mathbf{b}^*$ is the vector of parameter estimates excluding the first one, and b_1 is the estimated regression coefficient for the independent variable of interest. This estimate assumes that there is no natural response rate ($C = 0$). When C is nonzero, the quantiles and confidence limits for the independent variable correspond to the adjusted probability $C+(1-C)p$, rather than to p. As a result, an estimate of the value yielding response rate p is associated with the $(p-C)/(1-C)$ quantile. For example, if $C = 0.1$ then an estimate of the LD50 is found corresponding to the 0.44 quantile. This value can be thought of as yielding 50% of the variable's effect, but a 44% response rate. For both binary and ordinal models, the INVERSECL option provides estimates of the value of x_1, which yields Pr(first response level) $= p$, for various values of p.

This estimator is given as a ratio of random variables, such as $r = a/b$. Confidence limits for this ratio can be computed by using Fieller's theorem. A brief description of this theorem follows. See Finney (1971) for a more complete description of Fieller's theorem.

If the random variables a and b are thought to be distributed as jointly normal, then for any fixed value r the following probability statement holds if z is an $\alpha/2$ quantile from the standard normal distribution and $\mathbf{V}$ is the variance-covariance matrix of a and b:

$$\Pr\left((a-rb)^2 > z^2(V_{aa} - 2rV_{ab} + r^2V_{bb})\right) = \alpha$$

Usually the inequality can be solved for r to yield a confidence interval. The PROBIT procedure uses a value of 1.96 for z, corresponding to an α value of 0.05, unless the goodness-of-fit p-value is less than the specified value of the HPROB= option. When this happens, the covariance matrix is scaled by the heterogeneity factor, and a t distribution quantile is used for z.

It is possible for the roots of the equation for r to be imaginary or for the confidence interval to be all points outside of an interval. In these cases, the limits are set to missing by the PROBIT procedure.

Although the normal and logistic distribution give comparable fitted values of p if the empirically observed proportions are not too extreme, they can give appreciably different values when extrapolated into the tails. Correspondingly, the estimates of the confidence limits and dose values can be different for the two distributions even when they agree quite well in the body of the data. Extrapolation outside of the range of the actual data is often sensitive to model assumptions, and caution is advised if extrapolation is necessary.

It should be pointed out that, when there is a natural threshold C, the quantiles and confidence limits for the independent variable correspond to the adjusted probability $C + (1-C)p$.

OUTEST= *SAS-data-set*

The OUTEST= data set contains parameter estimates and the log likelihood for the model. You can specify a label in the MODEL statement to distinguish between the estimates for different models used by the PROBIT procedure. If you specify the COVOUT option, the OUTEST= data set also contains the estimated covariance matrix of the parameter estimates.

The OUTEST= data set contains each variable used as a dependent or independent variable in any MODEL statement. One observation consists of parameter values for the model with the dependent variable having the value −1. If you specify the COVOUT option, there are additional observations containing the rows of the estimated covariance matrix. For these observations, the dependent variable contains the parameter estimate for the corresponding row variable. The following variables are also added to the data set:

MODEL	a character variable containing the label of the MODEL statement, if present, or blank otherwise
NAME	a character variable containing the name of the dependent variable for the parameter estimates observations or the name of the row for the covariance matrix estimates
TYPE	a character variable containing the type of the observation, either PARMS for parameter estimates or COV for covariance estimates
DIST	a character variable containing the name of the distribution modeled
LNLIKE	a numeric variable containing the last computed value of the log likelihood
_C_	a numeric variable containing the estimated threshold parameter
INTERCEPT	a numeric variable containing the intercept parameter estimates and covariances

Any BY variables specified are also added to the OUTEST= data set.

XDATA= *SAS-data-set*

The XDATA= data set is used for specifying values for the effects in the MODEL statement when predicted values and/or fiducial limits for a single continuous variable (dose variable) are required. It is also used for plots specified by the CDFPLOT, IPPPLOT, LPREDPLOT, and PREDPPLOT statement.

The XDATA= data names a SAS data set that contains user input values for all the independent variables in the MODEL statement and the variables in the CLASS statement. The XDATA= data set has the same structure as the DATA= data set but is not required to have all the variables or observations that appear in the DATA= data set.

The XDATA= data set must contain all the independent variables in the MODEL statement and variables in the CLASS statement. Even though variables in the CLASS statement are not used in the MODEL statement, valid values are required for these variables in the XDATA= data set. Missing values are not allowed. For independent variables in the MODEL statement, although the dose variable's value is not used in the computing of predicted values and/or fiducial limits for the dose variable, missing values are not allowed in the XDATA= data set for any of the independent variables. Missing values are allowed for the dependent variables and other variables if they are included in the XDATA= data set and not listed in the CLASS statement.

If BY processing is used, the XDATA= data set should also include the BY variables, and there must be at least one valid observation for each BY group. If there is more than one valid observation in one BY group, the last one read is used for that BY group.

If there is no XDATA= data set in the PROC PROBIT statement, by default, the PROBIT procedure will use overall mean for effects containing continuous variable (or variables) and the highest level of a single classification variable as reference level. The rules are summarized as follows:

- If the effect contains a continuous variable (or variables), the overall mean of this effect is used.
- If the effect is a single classification variable, the highest level of the variable is used.

Traditional High-Resolution Graphics

This section provides examples of using syntax available with the traditional high-resolution plots. There are four plot statements that you can use to request traditional high-resolution plots: CDFPLOT, IPPPLOT, LPREDPLOT, and PREDPPLOT. Some of these statements apply only to either the binomial model or the multinomial model. Table 71.33 shows the availability of these statements for different models.

Table 71.33 Plot Statement Availability

Statement	Binomial	Multinomial
CDFPLOT	NO	YES
IPPPLOT	YES	NO
LPREDPLOT	YES	YES
PREDPPLOT	YES	YES

An alternative is to use ODS Graphics to obtain these plots. See the section "ODS Graphics" on page 5325 for details.

The following example uses the data set study in the section "Estimating the Natural Response Threshold Parameter" on page 5259 to illustrate how to create high-resolution plots for the binomial model:

```
proc probit data=study log10 optc;
   model respond/number=dose;
   predpplot var=dose cfit=blue; inset;
   lpredplot var=dose cfit=blue; inset;
   ippplot   var=dose cfit=blue; inset/pos=se;
run;
```

All plot statements must follow the MODEL statement. The VAR= option specifies a continuous independent variable (dose variable) against which the predicted probability or the linear predictor is plotted. The INSET statement requests the inset box with summary information. See the section "INSET Statement" on page 5277 for more details.

The PREDPPLOT statement creates the plot in Figure 71.6. This plot, similar to that in Figure 71.4, shows the relationship between dosage level, observed response proportions, and estimated probability values. The dashed lines represent pointwise confidence bands for the fitted probabilities,

which can be suppressed with the NOCONF option. See the section "PREDPPLOT Statement" on page 5300 for more details.

The IPPPLOT statement creates the plot in Figure 71.8, similar to that in Figure 71.5. See the section "IPPPLOT Statement" on page 5279 for details about this plot. The LPREDPLOT statement creates the linear predictor plot in Figure 71.7, which is described in the section "LPREDPLOT Statement" on page 5287.

Figure 71.6 Predicted Probability Plot

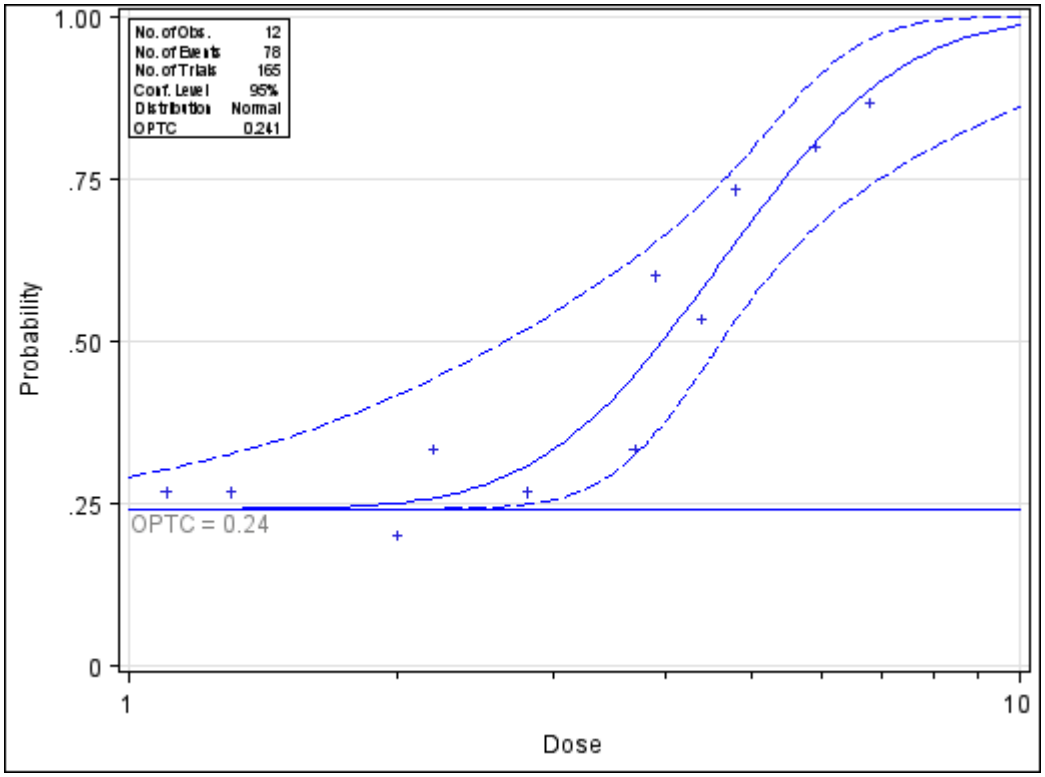


Figure 71.7 Linear Predictor Plot

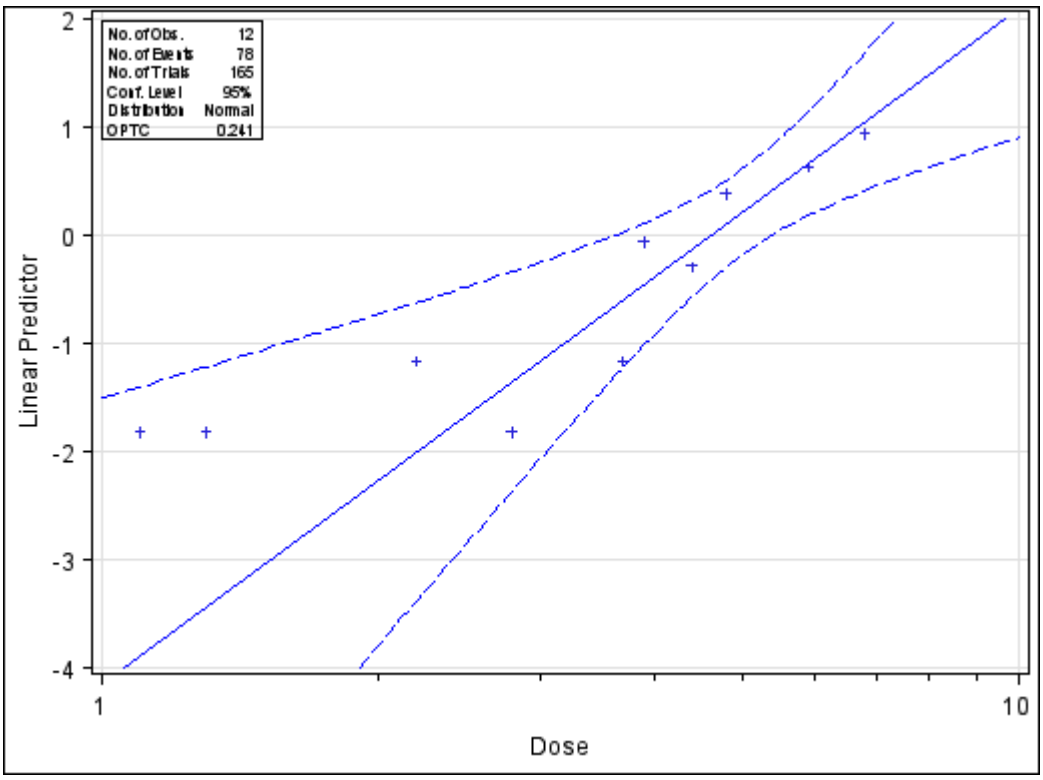
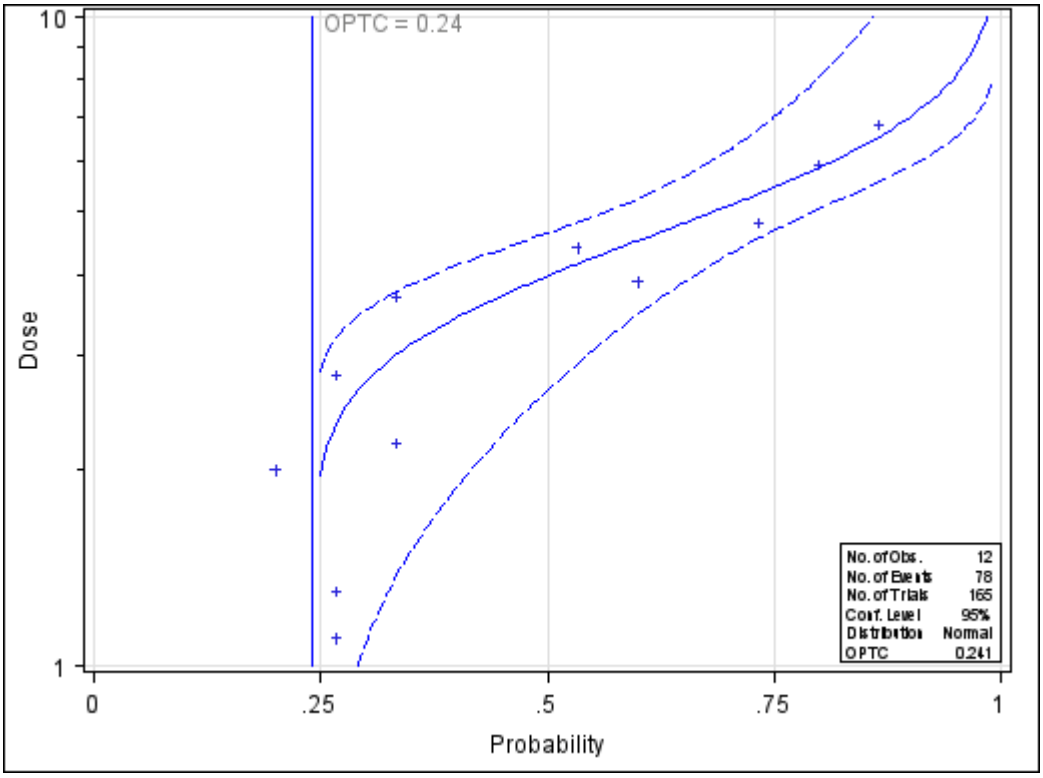


Figure 71.8 Inverse Predicted Probability Plot



The following example uses the data set multi from Example 71.2 to illustrate how to create high-resolution plots for the multinomial model:

```
proc probit data=multi order=data;
   class prep symptoms;
   model symptoms=prep ldose;
   cdfplot var=ldose  level=("None" "Mild" "Severe")
             cfit=blue cframe=ligr noconf;
   lpredplot var=ldose  level=("None" "Mild" "Severe")
             cfit=blue cframe=ligr;
   predpplot var=ldose  level=("None" "Mild" "Severe")
             cfit=blue cframe=ligr;
   weight n;
run;
```

The CDFPLOT statement creates the plot in Figure 71.9. This plot, similar to that in Output 71.2.4, shows the relationship between the cumulative response probabilities and the dose levels. The plots in Figure 71.10 and Figure 71.11 are similar to those with the binomial model.

Figure 71.9 Cumulative Probability Plot

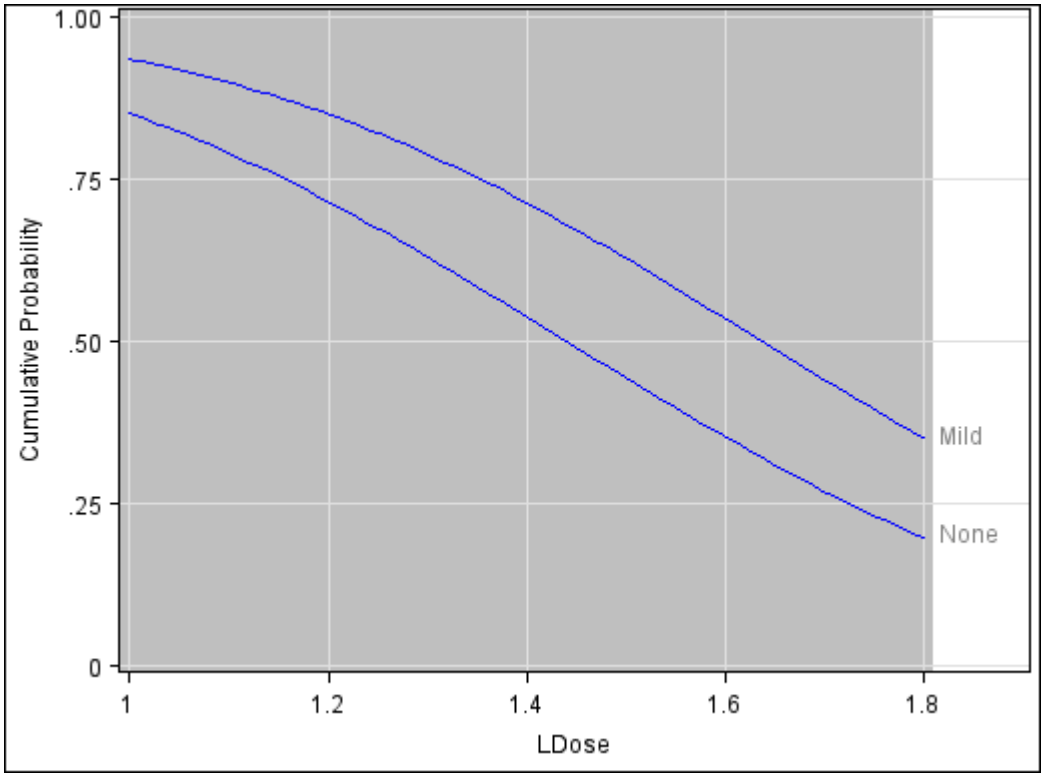


Figure 71.10 Linear Predictor Plot

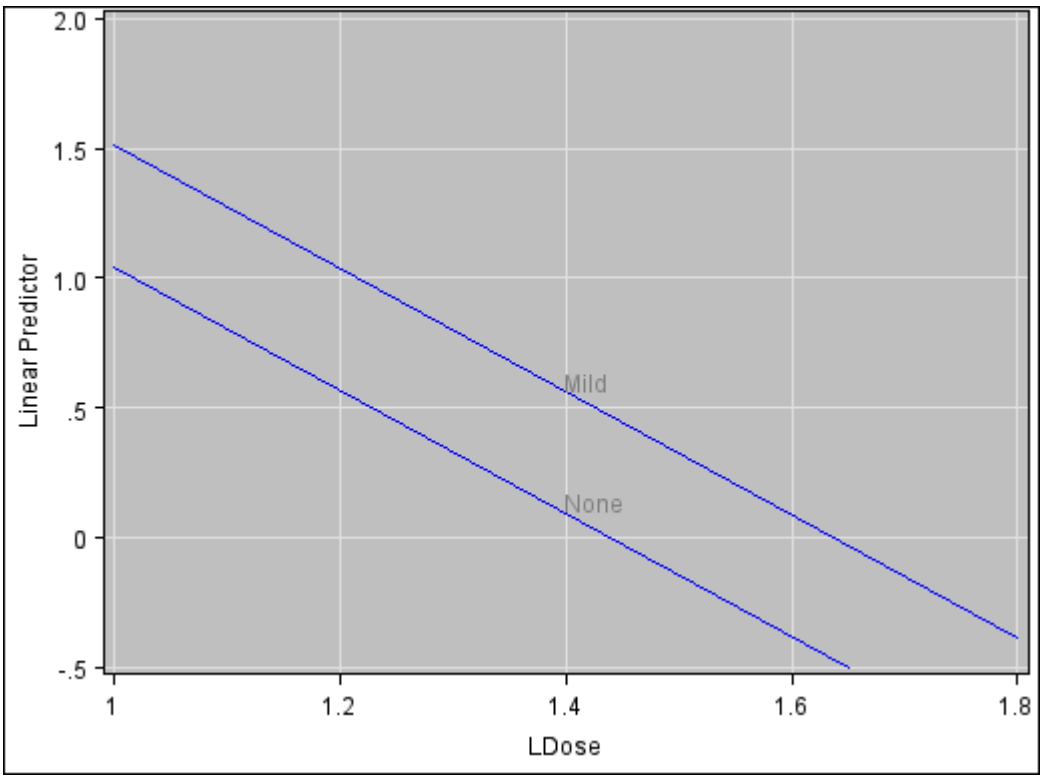
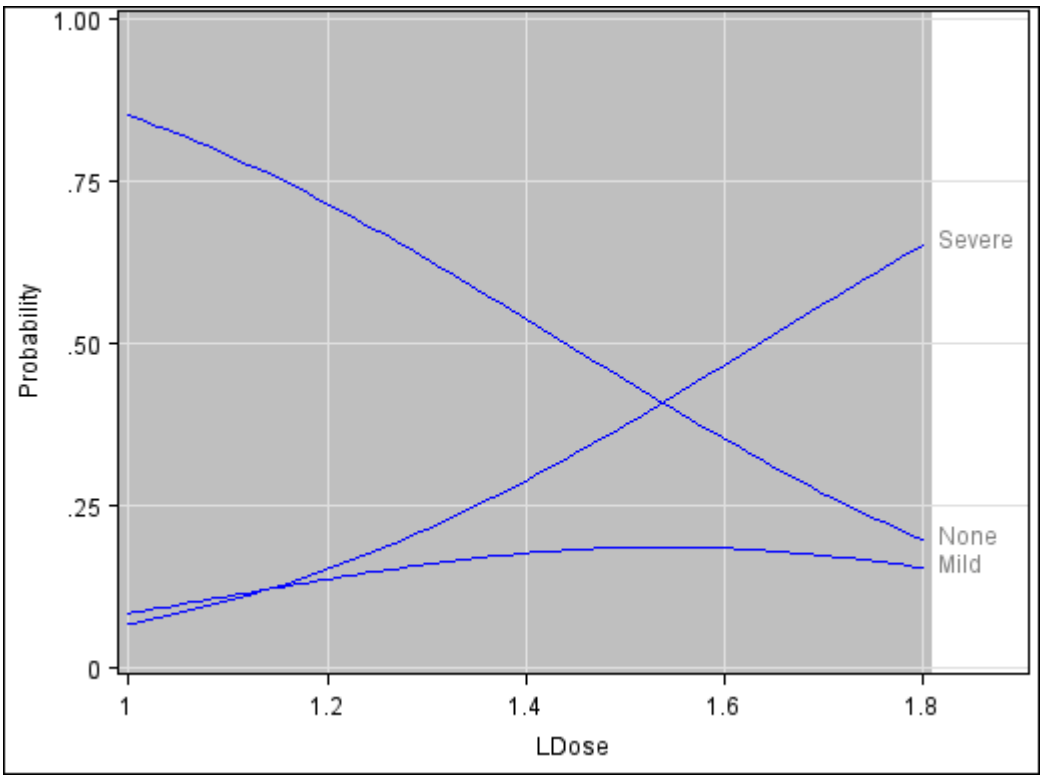


Figure 71.11 Predicted Probability Plot



Displayed Output

If you request the iteration history (ITPRINT), PROC PROBIT displays the following:

- the current value of the log likelihood
- the ridging parameter for the modified Newton-Raphson optimization process
- the current estimate of the parameters
- the current estimate of the parameter C for a natural (threshold) model
- the values of the gradient and the Hessian on the last iteration

If you include classification variables, PROC PROBIT displays the following:

- the numbers of levels for each classification variable
- the (ordered) values of the levels
- the number of observations used

After the model is fit, PROC PROBIT displays the following:

- the name of the input data set
- the name of the dependent variables
- the number of observations used
- the number of events and the number of trials
- the final value of the log-likelihood function
- the parameter estimates
- the standard error estimates of the parameter estimates
- approximate chi-square test statistics for the test

If you specify the COVB or CORRB options, PROC PROBIT displays the following:

- the estimated covariance matrix for the parameter estimates
- the estimated correlation matrix for the parameter estimates

If you specify the LACKFIT option, PROC PROBIT displays the following:

- a count of the number of levels of the response and the number of distinct sets of independent variables
- a goodness-of-fit test based on the Pearson chi-square
- a goodness-of-fit test based on the likelihood-ratio chi-square

If you specify only one independent variable, the normal distribution is used to model the probabilities, and the response is binary, then PROC PROBIT displays the following:

- the mean MU of the stimulus tolerance
- the scale parameter SIGMA of the stimulus tolerance
- the covariance matrix for MU, SIGMA, and the natural response parameter *C*

If you specify the INVERSECL options, PROC PROBIT also displays the following:

- the estimated dose along with the 95% fiducial limits for probability levels 0.01 to 0.10, 0.15 to 0.85 by 0.05, and 0.90 to 0.99

ODS Table Names

PROC PROBIT assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 71.34 ODS Tables Produced by PROC PROBIT

ODS Table Name	Description	Statement	Option
ClassLevels	Classification variable levels	CLASS	default
ConvergenceStatus	Convergence status	MODEL	default
CorrB	Parameter estimate correlation matrix	MODEL	CORRB
CovB	Parameter estimate covariance matrix	MODEL	COVB
CovTolerance	Covariance matrix for location and scale	MODEL	default
GoodnessOfFit	Goodness-of-fit tests	MODEL	LACKFIT
Heterogeneity	Heterogeneity correction	MODEL	LACKFIT
IterHistory	Iteration history	MODEL	ITPRINT
LagrangeStatistics	Lagrange statistics	MODEL	NOINT
LastGrad	Last evaluation of the gradient	MODEL	ITPRINT
LastHess	Last evaluation of the Hessian	MODEL	ITPRINT
LogProbitAnalysis	Probit analysis for log dose	MODEL	INVERSECL
ModelInfo	Model information	MODEL	default

Table 71.34 (continued)

ODS Table Name	Description	Statement	Option
MuSigma	Location and scale	MODEL	default
NObs	Observations summary	PROC	default
ParameterEstimates	Parameter estimates	MODEL	default
ParmInfo	Parameter indices	MODEL	default
ProbitAnalysis	Probit analysis for linear dose	MODEL	INVERSECL
ResponseLevels	Response-covariate profile	MODEL	LACKFIT
ResponseProfiles	Counts for ordinal data	MODEL	default
Type3Analysis	Type 3 tests	MODEL	default

ODS Graphics

The PROBIT procedure has two different systems to produce graphics. You can use the CDFPLOT, IPPPLOT, LPREDPLOT, and PREDPPLOT statements with a specified graph output destination to produce GRSEG type graphs. In SAS 9.2, you can also use ODS Graphics to produce these graphs.

These ODS graphs are controlled by the PLOTS= option in the PROC statement. You can specify more than one graph request with the PLOTS= option. Table 71.35 summarizes these requests.

In addition to the PLOTS= option, you must specify the ODS GRAPHICS statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

The names of the graphs that PROC PROBIT generates are listed in Table 71.36, along with the required statements and options. The following subsections provide information about these graphs.

Table 71.35 Options for Plots

Option	Plot
ALL	All appropriate plots
CDFPLOT	Estimated cumulative probability
IPPPLOT	Inverse predicted probability
LPREDPLOT	Linear predictor
NONE	No plot
PREDPPLOT	Predicted probability

CDF Plot

For a multinomial model, the predicted cumulative distribution function is defined as

$$\hat{F}_j(\mathbf{x}) = C + (1 - C)F(\hat{a}_j + \mathbf{x}'\hat{\mathbf{b}})$$

where $j = 1, \ldots, k$ are the indexes of the k levels of the multinomial response variable, F is the CDF of the distribution used to model the cumulative probabilities, $\hat{\mathbf{b}}$ is the vector of estimated

parameters, $\mathbf{x}$ is the covariate vector, $\hat{a}_j$ are estimated ordinal intercepts with $\hat{a}_1 = 0$, and C is the threshold parameter, either known or estimated from the model. Let x_1 be the covariate corresponding to the dose variable and $\mathbf{x}_{-1}$ be the vector of the rest of the covariates. Let the corresponding estimated parameters be $\hat{b}_1$ and $\hat{\mathbf{b}}_{-1}$. Then

$$\hat{F}_j(\mathbf{x}) = C + (1 - C)F(\hat{a}_j + x_1\hat{b}_1 + \mathbf{x}'_{-1}\hat{\mathbf{b}}_{-1})$$

To plot $\hat{F}_j$ as a function of x_1, $\mathbf{x}_{-1}$ must be specified. You can use the XDATA= option to provide the values of $\mathbf{x}_{-1}$ (see the XDATA= option in the PROC PROBIT statement for details), or use the default values that follow these rules:

- If the effect contains a continuous variable (or variables), the overall mean of this effect is used.
- If the effect is a single classification variable, the highest level of the variable is used.

The LEVEL= suboption specify the levels of the multinomial response variable for which the CDF curves are requested. There are $k - 1$ curves for a k-level multinomial response variable (for the highest level, it is the constant line 1). You can specify any of them to be plotted by the LEVEL= suboption. See the plot in Output 71.2.6 for an example.

Inverse Predicted Probability Plot

For the binomial model, the response variable is a probability. An estimate of the dose level $\hat{x}_1$ needed for a response of p is given by

$$\hat{x}_1 = (F^{-1}(p) - \mathbf{x}'_{-1}\hat{\mathbf{b}}_{-1})/\hat{b}_1$$

where F is the cumulative distribution function used to model the probability, $\mathbf{x}_{-1}$ is the vector of the rest of the covariates, $\hat{\mathbf{b}}_{-1}$ is the vector of the estimated parameters corresponding to $\mathbf{x}_{-1}$, and $\hat{b}_1$ is the estimated parameter for the dose variable of interest.

To plot $\hat{x}_1$ as a function of p, $\mathbf{x}_{-1}$ must be specified. You can use the XDATA= option to provide the values of $\mathbf{x}_{-1}$ (see the XDATA= option in the PROC PROBIT statement for details), or use the default values that follow these rules:

- If the effect contains a continuous variable (or variables), the overall mean of this effect is used.
- If the effect is a single classification variable, the highest level of the variable is used.

Output 71.4.12 in Example 71.4 shows an inverse predicted probability plot.

Linear Predictor Plot

For both binomial models and multinomial models, the linear predictor $\mathbf{x}'\mathbf{b}$ can be plotted against the first single continuous variable (dose variable) in the MODEL statement.

Let x_1 be the covariate of the dose variable, $\mathbf{x}_{-1}$ be the vector of the rest of the covariates, $\hat{\mathbf{b}}_{-1}$ be the vector of estimated parameters corresponding to $\mathbf{x}_{-1}$, and $\hat{b}_1$ be the estimated parameter for the dose variable of interest.

To plot $\hat{\mathbf{x}}'\mathbf{b}$ as a function of x_1, $\mathbf{x}_{-1}$ must be specified. You can use the XDATA= option to provide the values of $\mathbf{x}_{-1}$ (see the XDATA= option in the PROC PROBIT statement for details), or use the default values that follow these rules:

- If the effect contains a continuous variable (or variables), the overall mean of this effect is used.
- If the effect is a single classification variable, the highest level of the variable is used.

For the multinomial model, you can use the LEVEL= suboption to specify the levels for which the linear predictor lines are plotted.

The confidence limits for the predicted values are only available for the binomial model. Output 71.4.13 in Example 71.4 shows a linear predictor plot for a binomial model.

Predicted Probability Plot

The predicted probability is

$$\hat{p} = C + (1 - C)F(\mathbf{x}'\hat{\mathbf{b}})$$

for the binomial model and

$$\begin{aligned}
\hat{p}_1 &= C + (1 - C)F(\mathbf{x}'\hat{\mathbf{b}}) \\
\hat{p}_j &= (1 - C)(F(\hat{a}_j + \mathbf{x}'\hat{\mathbf{b}}) - F(\hat{a}_{j-1} + \mathbf{x}'\hat{\mathbf{b}})), \quad j = 2, \ldots, k-1 \\
\hat{p}_k &= (1 - C)(1 - F(\hat{a}_{k-1} + \mathbf{x}'\hat{\mathbf{b}}))
\end{aligned}$$

for the multinomial model with k response levels, where F is the cumulative distribution function used to model the probability, $\mathbf{x}'$ is the vector of the covariates, $\hat{a}_j$ are the estimated ordinal intercepts with $\hat{a}_1 = 0$, C is the threshold parameter, either known or estimated from the model, and $\hat{\mathbf{b}}'$ is the vector of estimated parameters.

To plot $\hat{p}$ (or $\hat{p}_j$) as a function of a continuous variable x_1, the remaining covariates $\mathbf{x}_{-1}$ must be specified. You can use the XDATA= option to provide the values of $\mathbf{x}_{-1}$ (see the XDATA= option in the PROC PROBIT statement for details), or use the default values that follow these rules:

- If the effect contains a continuous variable (or variables), the overall mean of this effect is used.
- If the effect is a single classification variable, the highest level of the variable is used.

For the multinomial model, you can use the LEVEL= suboption to specify the levels for which the linear predictor lines are plotted.

Confidence limits are plotted only for the binomial model. Output 71.1.7 in Example 71.1 shows a predicted probability plot for a binomial model; and Output 71.2.3 in Example 71.2 shows a predicted probability plot for a multinomial model.

ODS Graph Names

PROC PROBIT assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 71.36.

To request these graphs you must specify the ODS GRAPHICS statement in addition to the PLOTS= option described in Table 71.35. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Table 71.36 ODS Graphics Produced by PROC PROBIT

ODS Graph Name	Plot Description	Statement	PLOTS= Option
CDFPlot	Estimated cumulative probability	PROC	CDFPLOT
IPPPlot	Inverse predicted probability	PROC	IPPPLOT
LPredPlot	Linear predictor	PROC	LPREDPLOT
PredPPlot	Predicted probability	PROC	PREDPPLOT

Examples: PROBIT Procedure

Example 71.1: Dosage Levels

In this example, Dose is a variable representing the level of a stimulus, N represents the number of subjects tested at each level of the stimulus, and Response is the number of subjects responding to that level of the stimulus. Both probit and logit response models are fit to the data. The LOG10 option in the PROC PROBIT statement requests that the log base 10 of Dose is used as the independent variable. Specifically, for a given level of Dose, the probability p of a positive response is modeled as

$$p = \Pr(\text{Response}) = F\left(b_0 + b_1 \times \log_{10}(\text{Dose})\right)$$

The probabilities are estimated first by using the normal distribution function (the default) and then by using the logistic distribution function. Note that, in this model specification, the natural rate is assumed to be zero.

The LACKFIT option specifies lack-of-fit tests and the INVERSECL option specifies inverse confidence limits.

In the DATA step that reads the data, a number of observations are generated that have a missing value for the response. Although the PROBIT procedure does not use the observations with the missing values to fit the model, it does give predicted values for all nonmissing sets of independent variables. These data points fill in the plot of fitted and observed values in the logistic model displayed in Output 71.1.7. The plot, requested with the PLOT=PREDPPLOT option, displays the estimated logistic cumulative distribution function and the observed response rates.

The following statements produce Output 71.1.1:

```
data a;
   infile cards eof=eof;
   input Dose N Response @@;
   Observed= Response/N;
   output;
   return;
eof: do Dose=0.5 to 7.5 by 0.25;
        output;
     end;
   datalines;
1 10 1  2 12 2  3 10 4  4 10 5
5 12 8  6 10 8  7 10 10
;
proc probit log10;
   model Response/N=Dose / lackfit inversecl itprint;
   output out=B p=Prob std=std xbeta=xbeta;
run;
```

Output 71.1.1 Probit Analysis with Normal Distribution

The Probit Procedure

Iteration History for Parameter Estimates

Iter	Ridge	Loglikelihood	Intercept	Log10(Dose)
0	0	-51.292891	0	0
1	0	-37.881166	-1.355817008	2.635206083
2	0	-37.286169	-1.764939171	3.3408954936
3	0	-37.280389	-1.812147863	3.4172391614
4	0	-37.280388	-1.812704962	3.418117919
5	0	-37.280388	-1.812704962	3.418117919

Model Information

Data Set	WORK.A
Events Variable	Response
Trials Variable	N
Number of Observations	7
Number of Events	38
Number of Trials	74
Name of Distribution	Normal
Log Likelihood	-37.28038802

Last Evaluation of the Negative of the Gradient

Intercept	Log10(Dose)
3.4349069E-7	-2.09809E-8

Last Evaluation of the Negative of the Hessian

	Intercept	Log10(Dose)
Intercept	36.005280383	20.152675982
Log10(Dose)	20.152675982	13.078826305

Goodness-of-Fit Tests

Statistic	Value	DF	Value/DF	Pr > ChiSq
Pearson Chi-Square	3.6497	5	0.7299	0.6009
L.R. Chi-Square	4.6381	5	0.9276	0.4616

Response-Covariate Profile

Response Levels	2
Number of Covariate Values	7

Output 71.1.1 *continued*

Analysis of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	−1.8127	0.4493	−2.6934	−0.9320	16.27	<.0001
Log10(Dose)	1	3.4181	0.7455	1.9569	4.8794	21.02	<.0001

Probit Model in Terms of Tolerance Distribution

MU	SIGMA
0.53032254	0.29255866

Estimated Covariance Matrix for Tolerance Parameters

	MU	SIGMA
MU	0.002418	−0.000409
SIGMA	−0.000409	0.004072

The p-values in the goodness-of-fit table of 0.6009 for the Pearson chi-square and 0.4616 for the likelihood ratio chi-square indicate an adequate fit for the model fit with the normal distribution.

Tolerance distribution parameter estimates for the normal distribution indicate a mean tolerance for the population of 0.5303.

Output 71.1.2 displays probit analysis with the logarithm of dose levels. The LD50 (ED50 for log dose) is 0.5303, the dose corresponding to a probability of 0.5. This is the same as the mean tolerance for the normal distribution.

Output 71.1.2 Probit Analysis with Normal Distribution

The Probit Procedure

Probit Analysis on Log10(Dose)

Probability	Log10(Dose)	95% Fiducial Limits	
0.01	−0.15027	−0.69518	0.07710
0.02	−0.07052	−0.55766	0.13475
0.03	−0.01992	−0.47064	0.17156
0.04	0.01814	−0.40534	0.19941
0.05	0.04911	−0.35233	0.22218
0.06	0.07546	−0.30731	0.24165
0.07	0.09857	−0.26793	0.25881
0.08	0.11926	−0.23273	0.27425
0.09	0.13807	−0.20080	0.28837
0.10	0.15539	−0.17147	0.30142
0.15	0.22710	−0.05086	0.35631
0.20	0.28410	0.04369	0.40124
0.25	0.33299	0.12343	0.44116
0.30	0.37690	0.19348	0.47857
0.35	0.41759	0.25658	0.51504
0.40	0.45620	0.31429	0.55182
0.45	0.49356	0.36754	0.58999
0.50	0.53032	0.41693	0.63057
0.55	0.56709	0.46296	0.67451
0.60	0.60444	0.50618	0.72271
0.65	0.64305	0.54734	0.77603
0.70	0.68374	0.58745	0.83550
0.75	0.72765	0.62776	0.90265
0.80	0.77655	0.66999	0.98008
0.85	0.83354	0.71675	1.07279
0.90	0.90525	0.77313	1.19191
0.91	0.92257	0.78646	1.22098
0.92	0.94139	0.80083	1.25265
0.93	0.96208	0.81653	1.28759
0.94	0.98519	0.83394	1.32672
0.95	1.01154	0.85367	1.37149
0.96	1.04250	0.87669	1.42424
0.97	1.08056	0.90480	1.48928
0.98	1.13116	0.94189	1.57602
0.99	1.21092	0.99987	1.71321

Output 71.1.3 displays probit analysis with dose levels. The ED50 for dose is 3.39 with a 95% confidence interval of (2.61, 4.27).

Output 71.1.3 Probit Analysis with Normal Distribution

The Probit Procedure

Probit Analysis on Dose

Probability	Dose	95% Fiducial Limits	
0.01	0.70750	0.20175	1.19427
0.02	0.85012	0.27691	1.36380
0.03	0.95517	0.33834	1.48444
0.04	1.04266	0.39324	1.58274
0.05	1.11971	0.44429	1.66793
0.06	1.18976	0.49282	1.74443
0.07	1.25478	0.53960	1.81473
0.08	1.31600	0.58515	1.88042
0.09	1.37427	0.62980	1.94252
0.10	1.43019	0.67380	2.00181
0.15	1.68696	0.88950	2.27147
0.20	1.92353	1.10584	2.51906
0.25	2.15276	1.32870	2.76161
0.30	2.38180	1.56128	3.01000
0.35	2.61573	1.80543	3.27374
0.40	2.85893	2.06200	3.56306
0.45	3.11573	2.33098	3.89038
0.50	3.39096	2.61175	4.27138
0.55	3.69051	2.90374	4.72619
0.60	4.02199	3.20759	5.28090
0.65	4.39594	3.52651	5.97077
0.70	4.82770	3.86765	6.84706
0.75	5.34134	4.24385	7.99189
0.80	5.97787	4.67724	9.55169
0.85	6.81617	5.20900	11.82480
0.90	8.03992	5.93105	15.55653
0.91	8.36704	6.11584	16.63320
0.92	8.73752	6.32165	17.89163
0.93	9.16385	6.55431	19.39034
0.94	9.66463	6.82245	21.21881
0.95	10.26925	7.13949	23.52275
0.96	11.02811	7.52816	26.56066
0.97	12.03830	8.03149	30.85201
0.98	13.52585	8.74763	37.67206
0.99	16.25233	9.99709	51.66627

The following statements request probit analysis of dosage levels with the logistic distribution:

```
ods graphics on;

proc probit log10 plot=predpplot;
   model Response/N=Dose / d=logistic inversecl;
   output out=B p=Prob std=std xbeta=xbeta;
run;

ods graphics off;
```

The regression parameter estimates in Output 71.1.4 for the logistic model of −3.22 and 5.97 are approximately $\pi/\sqrt{3}$ times as large as those for the normal model.

Output 71.1.4 Probit Analysis with Logistic Distribution

The Probit Procedure

Model Information

Data Set	WORK.B
Events Variable	Response
Trials Variable	N
Number of Observations	7
Number of Events	38
Number of Trials	74
Name of Distribution	Logistic
Log Likelihood	-37.11065336

Analysis of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-3.2246	0.8861	-4.9613	-1.4880	13.24	0.0003
Log10(Dose)	1	5.9702	1.4492	3.1299	8.8105	16.97	<.0001

Output 71.1.5 and Output 71.1.6 show that both the ED50 and the LD50 are similar to those for the normal model.

Output 71.1.5 Probit Analysis with Logistic Distribution

The Probit Procedure

Probit Analysis on Log10(Dose)

Probability	Log10(Dose)	95% Fiducial Limits	
0.01	−0.22955	−0.97441	0.04234
0.02	−0.11175	−0.75158	0.12404
0.03	−0.04212	−0.62018	0.17265
0.04	0.00780	−0.52618	0.20771
0.05	0.04693	−0.45265	0.23533
0.06	0.07925	−0.39205	0.25826
0.07	0.10686	−0.34037	0.27796
0.08	0.13103	−0.29521	0.29530
0.09	0.15259	−0.25502	0.31085
0.10	0.17209	−0.21875	0.32498
0.15	0.24958	−0.07552	0.38207
0.20	0.30792	0.03092	0.42645
0.25	0.35611	0.11742	0.46451
0.30	0.39820	0.19143	0.49932
0.35	0.43644	0.25684	0.53275
0.40	0.47221	0.31588	0.56619
0.45	0.50651	0.36986	0.60089
0.50	0.54013	0.41957	0.63807
0.55	0.57374	0.46559	0.67894
0.60	0.60804	0.50846	0.72474
0.65	0.64381	0.54896	0.77673
0.70	0.68205	0.58815	0.83637
0.75	0.72414	0.62752	0.90582
0.80	0.77233	0.66915	0.98876
0.85	0.83067	0.71631	1.09242
0.90	0.90816	0.77562	1.23343
0.91	0.92766	0.79014	1.26931
0.92	0.94922	0.80607	1.30912
0.93	0.97339	0.82378	1.35391
0.94	1.00100	0.84384	1.40523
0.95	1.03332	0.86713	1.46546
0.96	1.07245	0.89511	1.53864
0.97	1.12237	0.93053	1.63228
0.98	1.19200	0.97952	1.76329
0.99	1.30980	1.06166	1.98569

Output 71.1.6 Probit Analysis with Logistic Distribution

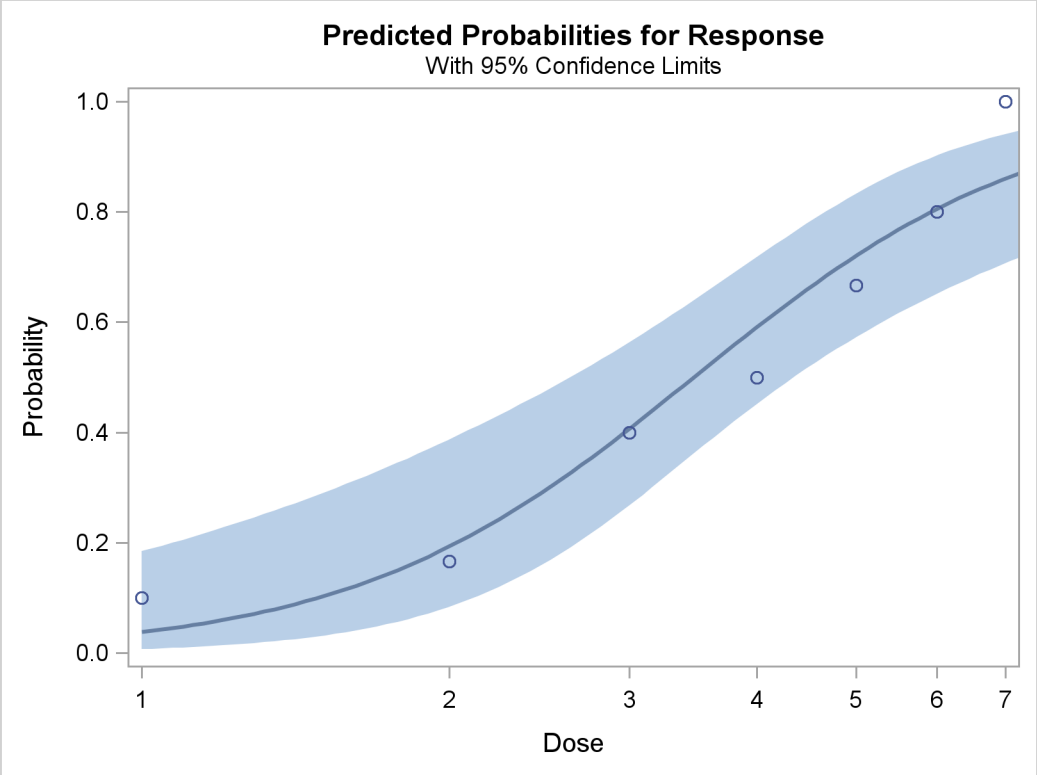
The Probit Procedure

Probit Analysis on Dose

Probability	Dose	95% Fiducial Limits	
0.01	0.58945	0.10607	1.10241
0.02	0.77312	0.17718	1.33058
0.03	0.90757	0.23978	1.48817
0.04	1.01813	0.29773	1.61327
0.05	1.11413	0.35266	1.71922
0.06	1.20018	0.40546	1.81244
0.07	1.27896	0.45670	1.89654
0.08	1.35218	0.50675	1.97379
0.09	1.42100	0.55588	2.04572
0.10	1.48625	0.60430	2.11339
0.15	1.77656	0.84038	2.41030
0.20	2.03199	1.07379	2.66961
0.25	2.27043	1.31046	2.91416
0.30	2.50152	1.55393	3.15736
0.35	2.73172	1.80652	3.40996
0.40	2.96627	2.06957	3.68292
0.45	3.21006	2.34345	3.98927
0.50	3.46837	2.62768	4.34578
0.55	3.74746	2.92138	4.77466
0.60	4.05546	3.22451	5.30573
0.65	4.40366	3.53961	5.98041
0.70	4.80891	3.87391	6.86079
0.75	5.29836	4.24155	8.05044
0.80	5.92009	4.66820	9.74455
0.85	6.77126	5.20365	12.37149
0.90	8.09391	5.96508	17.11715
0.91	8.46559	6.16800	18.59129
0.92	8.89644	6.39837	20.37592
0.93	9.40575	6.66469	22.58957
0.94	10.02317	6.97977	25.42292
0.95	10.79732	7.36428	29.20549
0.96	11.81534	7.85438	34.56521
0.97	13.25466	8.52173	42.88232
0.98	15.55972	9.53941	57.98207
0.99	20.40815	11.52549	96.75820

The PLOT=PREDPPLOT option together with the ODS GRAPHICS statement creates the plot of observed and fitted probabilities in Output 71.1.7. The dashed line represent pointwise confidence bands for the probabilities.

Output 71.1.7 Plot of Observed and Fitted Probabilities



Example 71.2: Multilevel Response

In this example, two preparations, a standard preparation and a test preparation, are each given at several dose levels to groups of insects. The symptoms are recorded for each insect within each group, and two multilevel probit models are fit. Because the natural sort order of the three levels is not the same as the response order, the ORDER=DATA option is specified in the PROC PROBIT statement to get the desired order.

The following statements fit two models:

```
data multi;
   input Prep $ Dose Symptoms $ N;
   LDose=log10(Dose);
   if Prep='test' then PrepDose=LDose;
   else PrepDose=0;
   datalines;
stand     10      None       33
stand     10      Mild        7
stand     10      Severe     10
stand     20      None       17
stand     20      Mild       13
stand     20      Severe     17
stand     30      None       14
stand     30      Mild        3
stand     30      Severe     28
```

```
   stand      40       None         9
   stand      40       Mild         8
   stand      40       Severe      32
   test       10       None        44
   test       10       Mild         6
   test       10       Severe       0
   test       20       None        32
   test       20       Mild        10
   test       20       Severe      12
   test       30       None        23
   test       30       Mild         7
   test       30       Severe      21
   test       40       None        16
   test       40       Mild         6
   test       40       Severe      19
   ;
proc probit order=data data=multi;
   class Prep Symptoms;
   nonpara: model Symptoms=Prep LDose PrepDose / lackfit;
   weight N;
run;
proc probit order=data data=multi ;
   class Prep Symptoms;
   parallel: model Symptoms=Prep LDose / lackfit;
   weight N;
run;
```

Results of these two models are shown in Output 71.2.1 and Output 71.2.2. The first model allows for nonparallelism between the dose response curves for the two preparations by inclusion of an interaction between Prep and LDose. The interaction term is labeled PrepDose in the "Analysis of Parameter Estimates" table. The results of this first model indicate that the parameter for the interaction term is not significant, having a Wald chi-square of 0.73. Also, since the first model is a generalization of the second, a likelihood ratio test statistic for this same parameter can be obtained by multiplying the difference in log likelihoods between the two models by 2. The value obtained, $2 \times (-345.94 - (-346.31))$, is 0.73. This is in close agreement with the Wald chi-square from the first model. The lack-of-fit test statistics for the two models do not indicate a problem with either fit.

Output 71.2.1 Multilevel Response: Nonparallel Analysis

The Probit Procedure

Model Information

Data Set	WORK.MULTI
Dependent Variable	Symptoms
Weight Variable	N
Number of Observations	23
Name of Distribution	Normal
Log Likelihood	-345.9401767

Output 71.2.1 *continued*

Class Level Information

Name	Levels	Values
Prep	2	stand test
Symptoms	3	None Mild Severe

Analysis of Maximum Likelihood Parameter Estimates

Parameter		DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept		1	3.8080	0.6252	2.5827	5.0333	37.10	<.0001
Intercept2		1	0.4684	0.0559	0.3589	0.5780	70.19	<.0001
Prep	stand	1	-1.2573	0.8190	-2.8624	0.3479	2.36	0.1247
Prep	test	0	0.0000	.	.	.	.	.
LDose		1	-2.1512	0.3909	-2.9173	-1.3851	30.29	<.0001
PrepDose		1	-0.5072	0.5945	-1.6724	0.6580	0.73	0.3935

Output 71.2.2 Multilevel Response: Parallel Analysis

The Probit Procedure

Model Information

Data Set	WORK.MULTI
Dependent Variable	Symptoms
Weight Variable	N
Number of Observations	23
Name of Distribution	Normal
Log Likelihood	-346.306141

Class Level Information

Name	Levels	Values
Prep	2	stand test
Symptoms	3	None Mild Severe

Analysis of Maximum Likelihood Parameter Estimates

Parameter		DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept		1	3.4148	0.4126	2.6061	4.2235	68.50	<.0001
Intercept2		1	0.4678	0.0558	0.3584	0.5772	70.19	<.0001
Prep	stand	1	-0.5675	0.1259	-0.8142	-0.3208	20.33	<.0001
Prep	test	0	0.0000	.	.	.	.	.
LDose		1	-2.3721	0.2949	-2.9502	-1.7940	64.68	<.0001

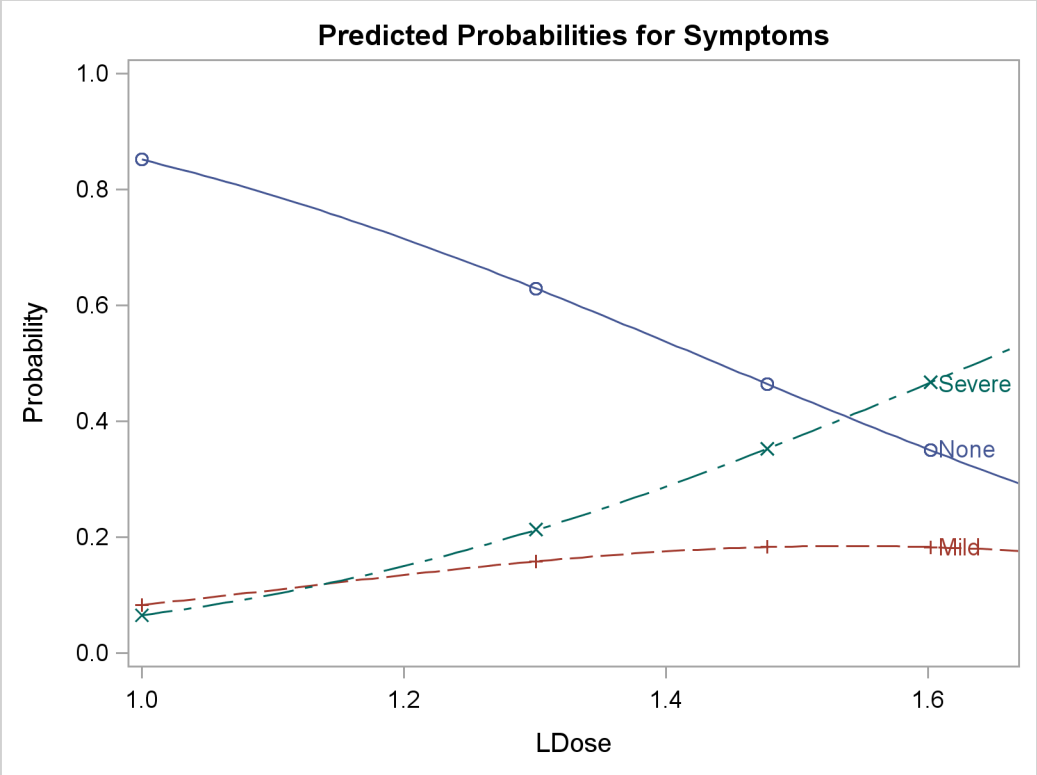
The negative coefficient associated with LDose indicates that the probability of having no symptoms (Symptoms='None') or no or mild symptoms (Symptoms='None' or Symptoms='Mild') decreases as LDose increases; that is, the probability of a severe symptom increases with LDose. This association is apparent for both treatment groups.

The negative coefficient associated with the standard treatment group (Prep = stand) indicates that the standard treatment is associated with more severe symptoms across all Ldose values.

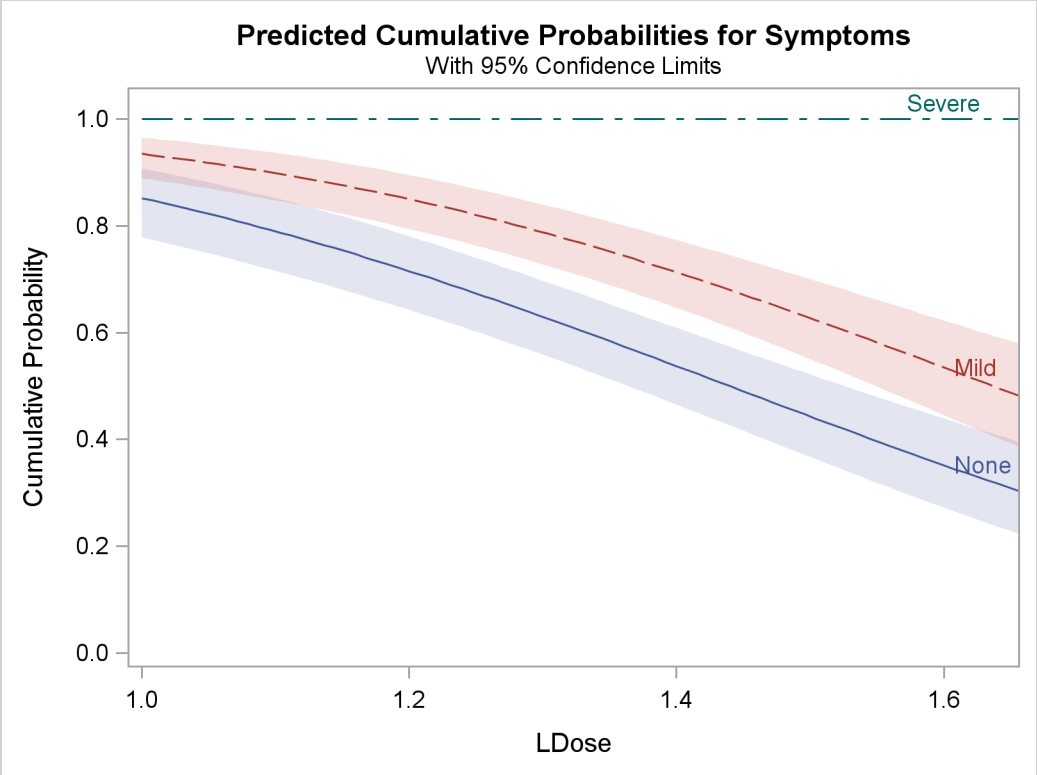
The following statements use the PLOTS= option to create the plot shown in Output 71.2.3 and Output 71.2.4. Output 71.2.3 is the plot of the probabilities of the response taking on individual levels as a function of LDose. Since there are two covariates, LDose and Prep, the value of the classification variable Prep is fixed at the highest level, test. Instead of individual response level probabilities, the CDFPLOT option creates the plot of the cumulative response probabilities with confidence limits shown in Output 71.2.4.

```
ods graphics on;

proc probit data=multi order=data
            plots=(predpplot(level=("None" "Mild" "Severe"))
                   cdfplot(level=("None" "Mild" "Severe")));
   class Prep Symptoms;
   parallel: model Symptoms=Prep LDose / lackfit;
   weight N;
run;

ods graphics off;
```

Output 71.2.3 Plot of Predicted Probabilities for the Test Preparation Group



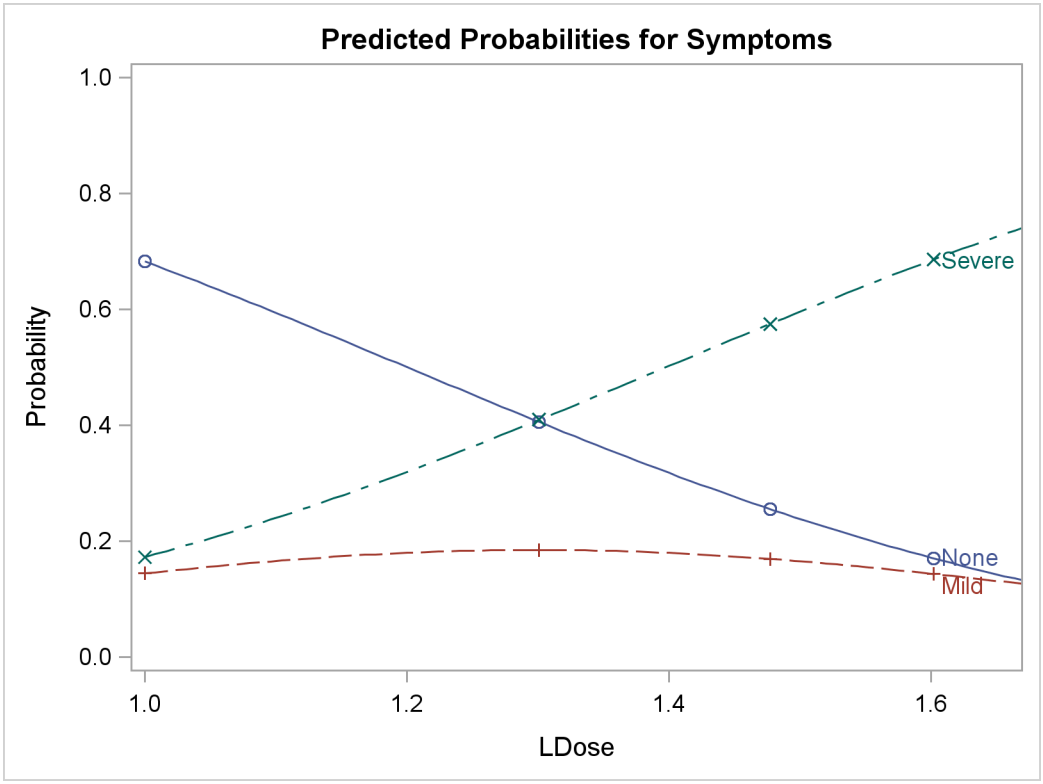
Output 71.2.4 Plot of Predicted Cumulative Probabilities for the Test Preparation Group



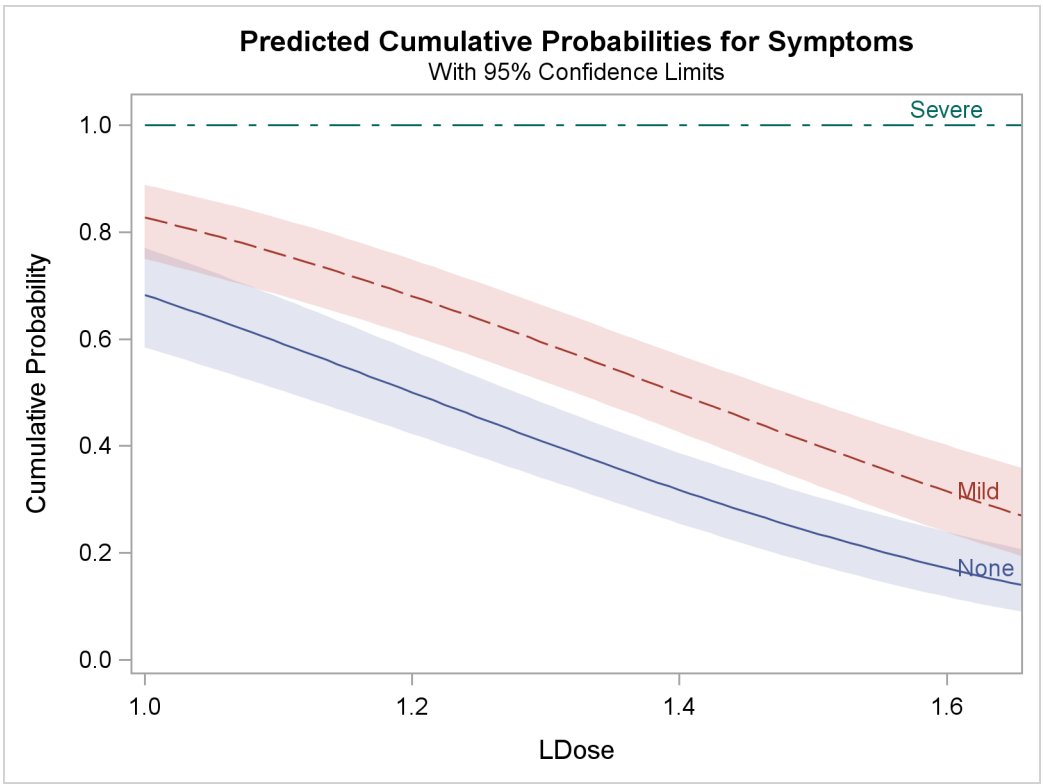
The following statements use the XDATA= data set to create plots of predicted probabilities and cumulative probabilities with Prep set to the stand level. The resulting plots are shown in Output 71.2.5 and Output 71.2.6.

```
data xrow;
   input Prep $ Dose Symptoms $ N;
   LDose=log10(Dose);
   datalines;
stand     40      Severe     32
run;

ods graphics on;

proc probit data=multi order=data xdata=xrow
            plots=(predpplot(level=("None" "Mild" "Severe"))
                   cdfplot(level=("None" "Mild" "Severe")));
   class Prep Symptoms;
   parallel: model Symptoms=Prep LDose / lackfit;
   weight N;
run;

ods graphics off;
```

Output 71.2.5 Plot of Predicted Probabilities for the Standard Preparation Group



Output 71.2.6 Plot of Predicted Cumulative Probabilities for the Standard Preparation Group



Example 71.3: Logistic Regression

In this example, a series of people are asked whether or not they would subscribe to a new newspaper. For each person, the variables sex (Female, Male), age, and subs (1=yes,0=no) are recorded. The PROBIT procedure is used to fit a logistic regression model to the probability of a positive response (subscribing) as a function of the variables sex and age. Specifically, the probability of subscribing is modeled as

$$p = \Pr(\text{subs} = 1) = F\left(b_0 + b_1 \times \text{sex} + b_2 \times \text{age}\right)$$

where F is the cumulative logistic distribution function.

By default, the PROBIT procedure models the probability of the lower response level for binary data. One way to model $\Pr(\text{subs} = 1)$ is to format the response variable so that the formatted value corresponding to subs=1 is the lower level. The following statements format the values of subs as 1 = 'accept' and 0 = 'reject', so that PROBIT models Pr(accept) = $\Pr(\text{subs} = 1)$. They produce Output 71.3.1.

```
   data news;
      input sex $ age subs @@;
      datalines;
   Female     35    0   Male       44    0
   Male       45    1   Female     47    1
   Female     51    0   Female     47    0
   Male       54    1   Male       47    1
   Female     35    0   Female     34    0
   Female     48    0   Female     56    1
   Male       46    1   Female     59    1
   Female     46    1   Male       59    1
   Male       38    1   Female     39    0
   Male       49    1   Male       42    1
   Male       50    1   Female     45    0
   Female     47    0   Female     30    1
   Female     39    0   Female     51    0
   Female     45    0   Female     43    1
   Male       39    1   Male       31    0
   Female     39    0   Male       34    0
   Female     52    1   Female     46    0
   Male       58    1   Female     50    1
   Female     32    0   Female     52    1
   Female     35    0   Female     51    0
   ;

   proc format;
      value subscrib 1 = 'accept' 0 = 'reject';
   run;

proc probit data=news;
   class subs sex;
   model subs=sex age / d=logistic itprint;
   format subs subscrib.;
run;
```

Output 71.3.1 Logistic Regression of Subscription Status

The Probit Procedure

Iteration History for Parameter Estimates

Iter	Ridge	Loglikelihood	Intercept	sexFemale	age
0	0	-27.725887	0	0	0
1	0	-20.142659	-3.634567629	-1.648455751	0.1051634384
2	0	-19.52245	-5.254865196	-2.234724956	0.1506493473
3	0	-19.490439	-5.728485385	-2.409827238	0.1639621828
4	0	-19.490303	-5.76187293	-2.422349862	0.1649007124
5	0	-19.490303	-5.7620267	-2.422407743	0.1649050312
6	0	-19.490303	-5.7620267	-2.422407743	0.1649050312

Model Information

Data Set	WORK.NEWS
Dependent Variable	subs
Number of Observations	40
Name of Distribution	Logistic
Log Likelihood	-19.49030281

Class Level Information

Name	Levels	Values
subs	2	accept reject
sex	2	Female Male

Last Evaluation of the Negative of the Gradient

Intercept	sexFemale	age
-5.95557E-12	8.768324E-10	-1.6367E-8

Last Evaluation of the Negative of the Hessian

	Intercept	sexFemale	age
Intercept	6.4597397447	4.6042218284	292.04051848
sexFemale	4.6042218284	4.6042218284	216.20829515
age	292.04051848	216.20829515	13487.329973

Analysis of Maximum Likelihood Parameter Estimates

Parameter		DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept		1	-5.7620	2.7635	-11.1783	-0.3458	4.35	0.0371
sex	Female	1	-2.4224	0.9559	-4.2959	-0.5489	6.42	0.0113
sex	Male	0	0.0000	.	.	.	.	.
age		1	0.1649	0.0652	0.0371	0.2927	6.40	0.0114

Output 71.3.1 shows that there appears to be an effect due to both the variables sex and age. The positive coefficient for age indicates that older people are more likely to subscribe than younger people. The negative coefficient for sex indicates that females are less likely to subscribe than males.

Example 71.4: An Epidemiology Study

The data in this example, which are from an epidemiology study, consist of five variables: the number, r, of individuals surviving after an epidemic, out of n treated, for combinations of medicine dosage (dose), treatment (treat = A, B), and sex (sex = 0(Female), 1(Male)).

To see whether the two treatments have different effects on male and female individual survival rates, the interaction term between the two variables treat and sex is included in the model.

The following invocation of PROC PROBIT fits the binary probit model to the grouped data:

```
   data epidemic;
      input treat$ dose n r sex @@;
      label dose = Dose;
      datalines;
   A  2.17 142 142  0   A   .57 132  47  1
   A  1.68 128 105  1   A  1.08 126 100  0
   A  1.79 125 118  0   B  1.66 117 115  1
   B  1.49 127 114  0   B  1.17  51  44  1
   B  2.00 127 126  0   B   .80 129 100  1
   ;

   data xval;
      input treat $ dose sex ;
      datalines;
   B  2.  1
   ;

ods graphics on;

proc probit optc lackfit covout data=epidemic
            outest = out1 xdata = xval
            Plots=(predpplot ippplot lpredplot)           ;
   class treat sex;
   model r/n = dose treat sex sex*treat/corrb covb inversecl;
   output out = out2 p =p;
run;

ods graphics off;
```

The results of this analysis are shown in the outputs that follow.

Beginning with SAS 8.2, the PROBIT procedure does not support multiple MODEL statements. Only the last one is used if there is more than one MODEL statement in an invocation of the PROBIT procedure.

Output 71.4.1 displays the table of level information for *all* classification variables in the CLASS statement.

Output 71.4.1 Class Level Information

The Probit Procedure

Class Level Information

Name	Levels	Values
treat	2	A B
sex	2	0 1

Output 71.4.2 displays the table of parameter information for the effects in the MODEL statement. The name of a parameter is generated by combining the variable names and level names in the effect. The maximum length of a parameter name is 32. The names of the effects are specified in the MODEL statement. The length of names of effects can be specified by the NAMELEN= option in the PROC PROBIT statement, with the default length 20.

Output 71.4.2 Parameter Information

Parameter Information

Parameter	Effect	treat	sex
Intercept	Intercept		
dose	dose		
treatA	treat	A	
treatB	treat	B	
sex0	sex		0
sex1	sex		1
treatAsex0	treat*sex	A	0
treatAsex1	treat*sex	A	1
treatBsex0	treat*sex	B	0
treatBsex1	treat*sex	B	1

Output 71.4.3 displays background information about the model fit. Included are the name of the input data set, the response variables used, and the number of observations, events, and trials. The table also includes the status of the convergence of the model fitting algorithm and the final value of the log-likelihood function.

Output 71.4.3 Model Information

The Probit Procedure

Model Information

Data Set	WORK.EPIDEMIC
Events Variable	r
Trials Variable	n
Number of Observations	10
Number of Events	1011
Number of Trials	1204
Name of Distribution	Normal
Log Likelihood	-387.2467391

Output 71.4.4 displays the table of goodness-of-fit tests requested with the LACKFIT option in the PROC PROBIT statement. Two goodness-of-fit statistics, the Pearson chi-square statistic and the likelihood ratio chi-square statistic, are computed. The grouping method for computing these statistics can be specified by the AGGREGATE= option. The details can be found in the AGGREGATE= option, and an example can be found in the second part of this example. By default, the PROBIT procedure uses the covariates in the MODEL statement to do grouping. Observations with the same values of the covariates in the MODEL statement are grouped into cells and the two statistics are computed according to these cells. The total number of cells and the number of levels for the response variable are reported next in the "Response-Covariate Profile."

In this example, neither the Pearson chi-square nor the log-likelihood ratio chi-square tests are significant at the 0.1 level, which is the default test level used by the PROBIT procedure. That means that the model, which includes the interaction of treat and sex, is suitable for this epidemiology data set. (Further investigation shows that models without the interaction of treat and sex are not acceptable by either test.)

Output 71.4.4 Goodness-of-Fit Tests and Response-Covariate Profile

Goodness-of-Fit Tests

Statistic	Value	DF	Value/DF	Pr > ChiSq
Pearson Chi-Square	4.9317	4	1.2329	0.2944
L.R. Chi-Square	5.7079	4	1.4270	0.2220

Response-Covariate Profile

Response Levels	2
Number of Covariate Values	10

Output 71.4.5 displays the Type III test results for all effects specified in the MODEL statement, which include the degrees of freedom for the effect, the Wald Chi-Square test statistic, and the p-value.

Output 71.4.5 Type III Tests

Type III Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
dose	1	42.1691	<.0001
treat	1	16.1421	<.0001
sex	1	1.7710	0.1833
treat*sex	1	13.9343	0.0002

Output 71.4.6 displays the table of parameter estimates for the model. The PROBIT procedure displays information for all the parameters of an effect. Degenerate parameters are indicated by 0 degree of freedom. Confidence intervals are computed for all parameters with nonzero degrees of freedom, including the natural threshold C if the OPTC option is specified in the PROC PROBIT statement. The confidence level can be specified by the ALPHA= option in the MODEL statement. The default confidence level is 95%.

Output 71.4.6 Analysis of Parameter Estimates

Analysis of Maximum Likelihood Parameter Estimates

Parameter		DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept		1	−0.8871	0.3632	−1.5991	−0.1752	5.96	0.0146
dose		1	1.6774	0.2583	1.1711	2.1837	42.17	<.0001
treat	A	1	−1.2537	0.2616	−1.7664	−0.7410	22.97	<.0001
treat	B	0	0.0000	.	.	.	.	.
sex	0	1	−0.4633	0.2289	−0.9119	−0.0147	4.10	0.0429
sex	1	0	0.0000	.	.	.	.	.
treat*sex	A 0	1	1.2899	0.3456	0.6126	1.9672	13.93	0.0002
treat*sex	A 1	0	0.0000	.	.	.	.	.
treat*sex	B 0	0	0.0000	.	.	.	.	.
treat*sex	B 1	0	0.0000	.	.	.	.	.
_C_		1	0.2735	0.0946	0.0881	0.4589		

From Table 71.4.6, you can see the following results:

- The variable dose has a significant positive effect on the survival rate.
- Individuals under treatment A have a lower survival rate.
- Male individuals have a higher survival rate.
- Female individuals under treatment A have a higher survival rate.

Output 71.4.7 and Output 71.4.8 display tables of estimated covariance matrix and estimated correlation matrix for estimated parameters with a nonzero degree of freedom, respectively. They are computed by the inverse of the Hessian matrix of the estimated parameters.

Output 71.4.7 Estimated Covariance Matrix

Estimated Covariance Matrix

	Intercept	dose	treatA	sex0	treatAsex0
Intercept	0.131944	−0.087353	0.053551	0.030285	−0.067056
dose	−0.087353	0.066723	−0.047506	−0.034081	0.058620
treatA	0.053551	−0.047506	0.068425	0.036063	−0.075323
sex0	0.030285	−0.034081	0.036063	0.052383	−0.063599
treatAsex0	−0.067056	0.058620	−0.075323	−0.063599	0.119408
_C_	−0.028073	0.018196	−0.017084	−0.008088	0.019134

Estimated Covariance Matrix

	_C_
Intercept	−0.028073
dose	0.018196
treatA	−0.017084
sex0	−0.008088
treatAsex0	0.019134
_C_	0.008948

Output 71.4.8 Estimated Correlation Matrix

Estimated Correlation Matrix

	Intercept	dose	treatA	sex0	treatAsex0
Intercept	1.000000	−0.930998	0.563595	0.364284	−0.534227
dose	−0.930998	1.000000	−0.703083	−0.576477	0.656744
treatA	0.563595	−0.703083	1.000000	0.602359	−0.833299
sex0	0.364284	−0.576477	0.602359	1.000000	−0.804154
treatAsex0	−0.534227	0.656744	−0.833299	−0.804154	1.000000
_C_	−0.817027	0.744699	−0.690420	−0.373565	0.585364

Estimated Correlation Matrix

	_C_
Intercept	−0.817027
dose	0.744699
treatA	−0.690420
sex0	−0.373565
treatAsex0	0.585364
_C_	1.000000

Output 71.4.9 displays the computed values and fiducial limits for the first single continuous variable dose in the MODEL statement, given the probability levels, without the effect of the natural threshold, and when the option INSERSECL in the MODEL statement is specified. If there is no single continuous variable in the MODEL specification but the INVERSECL option is specified, an error is reported.

Output 71.4.9 Probit Analysis on Dose

The Probit Procedure

Probit Analysis on dose

Probability	dose	95% Fiducial Limits	
0.01	−0.85801	−1.81301	−0.33743
0.02	−0.69549	−1.58167	−0.21116
0.03	−0.59238	−1.43501	−0.13093
0.04	−0.51482	−1.32476	−0.07050
0.05	−0.45172	−1.23513	−0.02130
0.06	−0.39802	−1.15888	0.02063
0.07	−0.35093	−1.09206	0.05742
0.08	−0.30877	−1.03226	0.09039
0.09	−0.27043	−0.97790	0.12040
0.10	−0.23513	−0.92788	0.14805
0.15	−0.08900	−0.72107	0.26278
0.20	0.02714	−0.55706	0.35434
0.25	0.12678	−0.41669	0.43322
0.30	0.21625	−0.29095	0.50437
0.35	0.29917	−0.17477	0.57064
0.40	0.37785	−0.06487	0.63387
0.45	0.45397	0.04104	0.69546
0.50	0.52888	0.14481	0.75654
0.55	0.60380	0.24800	0.81819
0.60	0.67992	0.35213	0.88157
0.65	0.75860	0.45879	0.94803
0.70	0.84151	0.56985	1.01942
0.75	0.93099	0.68770	1.09847
0.80	1.03063	0.81571	1.18970
0.85	1.14677	0.95926	1.30171
0.90	1.29290	1.12867	1.45386
0.91	1.32819	1.16747	1.49273
0.92	1.36654	1.20867	1.53590
0.93	1.40870	1.25284	1.58450
0.94	1.45579	1.30084	1.64012
0.95	1.50949	1.35397	1.70515
0.96	1.57258	1.41443	1.78353
0.97	1.65015	1.48626	1.88238
0.98	1.75326	1.57833	2.01720
0.99	1.91577	1.71776	2.23537

If the XDATA= option is used to input a data set for the independent variables in the MODEL statement, the PROBIT procedure uses these values for the independent variables other than the single continuous variable. Missing values are not permitted in the XDATA= data set for the independent variables, although the value for the single continuous variable is not used in the computing of the fiducial limits. A suitable valid value should be given. In the data set xval created by the SAS statements on page 5345, dose = 2.

See the section "XDATA= *SAS-data-set*" on page 5317 for the default values for those effects other than the single continuous variable, for which the fiducial limits are computed.

In this example, there are two classification variables, treat and sex. Fiducial limits for the dose

variable are computed for the highest level of the classification variables, treat = B and sex = 1, which is the default specification. Since these are the default values, you would get the same values and fiducial limits if you did not specify the XDATA= option in this example. The confidence level for the fiducial limits can be specified by the ALPHA= option in the MODEL statement. The default level is 95%.

If a LOG10 or LOG option is used in the PROC PROBIT statement, the values and the fiducial limits are computed for both the single continuous variable and its logarithm.

Output 71.4.10 displays the OUTEST= data set. All parameters for an effect are included.

Output 71.4.10 Outest Data Set for Epidemiology Study

Obs	_MODEL_	_NAME_	_TYPE_	_DIST_	_STATUS_	_LNLIKE_	r	Intercept
1		r	PARMS	Normal	0 Converged	-387.247	-1.00000	-0.88714
2		Intercept	COV	Normal	0 Converged	-387.247	-0.88714	0.13194
3		dose	COV	Normal	0 Converged	-387.247	1.67739	-0.08735
4		treatA	COV	Normal	0 Converged	-387.247	-1.25367	0.05355
5		treatB	COV	Normal	0 Converged	-387.247	0.00000	0.00000
6		sex0	COV	Normal	0 Converged	-387.247	-0.46329	0.03029
7		sex1	COV	Normal	0 Converged	-387.247	0.00000	0.00000
8		treatAsex0	COV	Normal	0 Converged	-387.247	1.28991	-0.06706
9		treatAsex1	COV	Normal	0 Converged	-387.247	0.00000	0.00000
10		treatBsex0	COV	Normal	0 Converged	-387.247	0.00000	0.00000
11		treatBsex1	COV	Normal	0 Converged	-387.247	0.00000	0.00000
12		_C_	COV	Normal	0 Converged	-387.247	0.27347	-0.02807

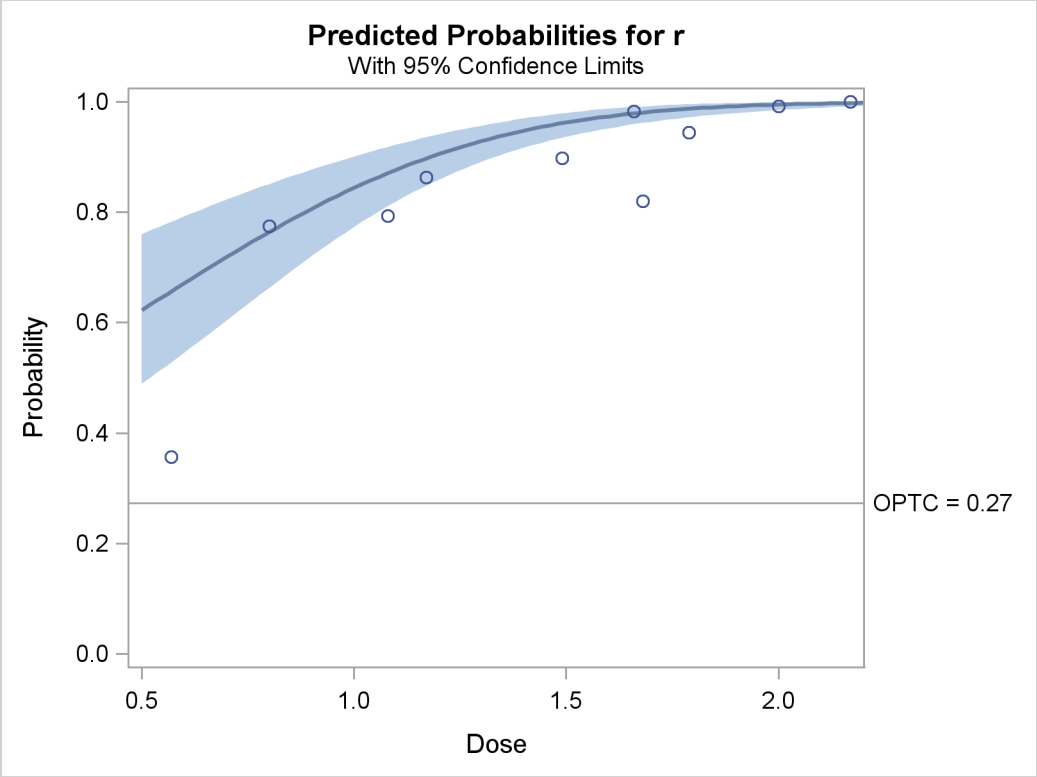
Obs	dose	treatA	treat B	sex0	sex1	treat Asex0	treat Asex1	treat Bsex0	treat Bsex1	_C_
1	1.67739	-1.25367	0	-0.46329	0	1.28991	0	0	0	0.27347
2	-0.08735	0.05355	0	0.03029	0	-0.06706	0	0	0	-0.02807
3	0.06672	-0.04751	0	-0.03408	0	0.05862	0	0	0	0.01820
4	-0.04751	0.06843	0	0.03606	0	-0.07532	0	0	0	-0.01708
5	0.00000	0.00000	0	0.00000	0	0.00000	0	0	0	0.00000
6	-0.03408	0.03606	0	0.05238	0	-0.06360	0	0	0	-0.00809
7	0.00000	0.00000	0	0.00000	0	0.00000	0	0	0	0.00000
8	0.05862	-0.07532	0	-0.06360	0	0.11941	0	0	0	0.01913
9	0.00000	0.00000	0	0.00000	0	0.00000	0	0	0	0.00000
10	0.00000	0.00000	0	0.00000	0	0.00000	0	0	0	0.00000
11	0.00000	0.00000	0	0.00000	0	0.00000	0	0	0	0.00000
12	0.01820	-0.01708	0	-0.00809	0	0.01913	0	0	0	0.00895

The plots in the following three outputs, Output 71.4.11, Output 71.4.12, and Output 71.4.13, are generated by the PLOTS= option. The first plot, specified with the PREDPPLOT option, is the plot of the predicted probability against the first single continuous variable dose in the MODEL statement. You can specify values of other independent variables in the MODEL statement by using an XDATA= data set or by using the default values.

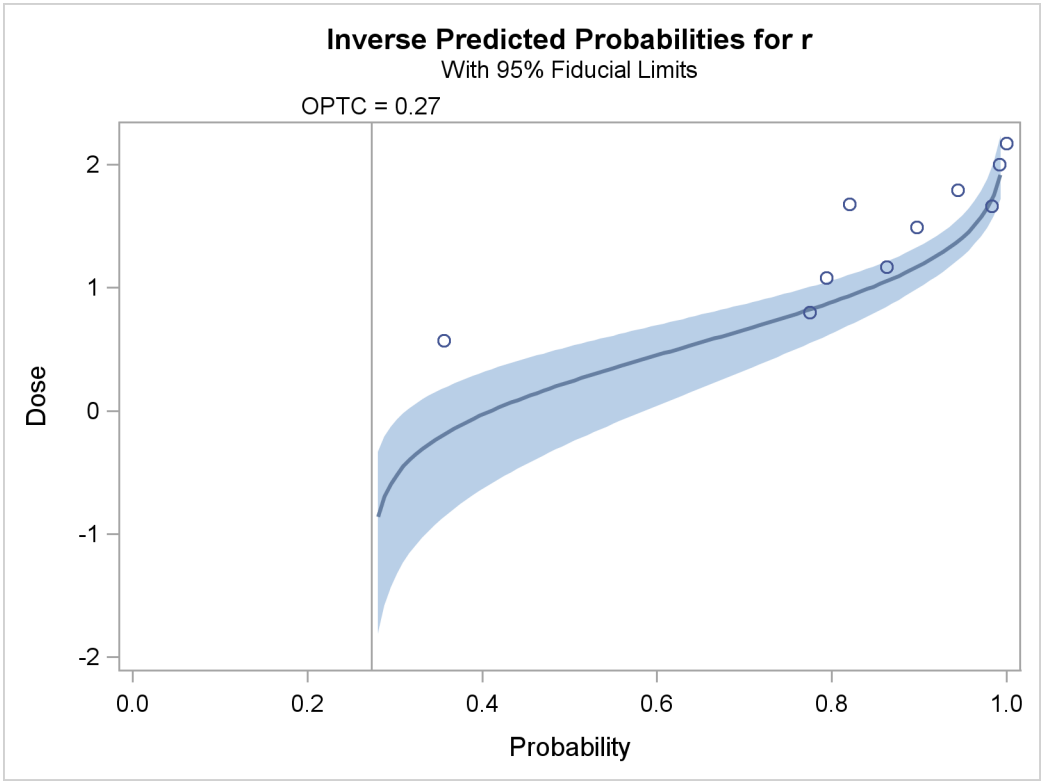
The second plot, specified with the IPPPLOT option, is the inverse of the predicted probability plot with the fiducial limits. It should be pointed out that the fiducial limits are *not* just the inverse of the confidence limits in the predicted probability plot; see the section "Inverse Confidence Limits"

on page 5315 for the computation of these limits. The third plot, specified with the LPREDPLOT option, is the plot of the linear predictor $\mathbf{x}'_i$ against the first single continuous variable with the Wald confidence intervals.

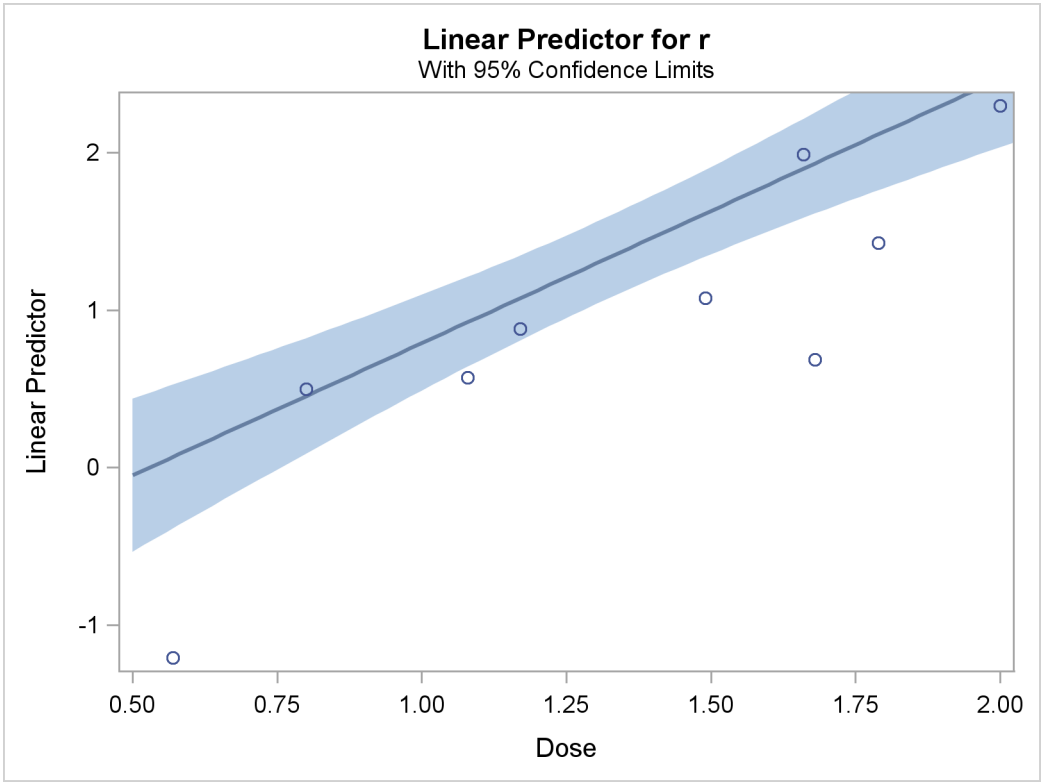
Output 71.4.11 Predicted Probability Plot



Output 71.4.12 Inverse Predicted Probability Plot



Output 71.4.13 Linear Predictor Plot



When you combine the INEST= data set and the MAXIT= option in the MODEL statement, the PROBIT procedure can do prediction, if the parameterizations for the models used for the training data and the validation data are exactly the same. The following SAS statements show an example:

```
   data validate;
      input treat $ dose  sex  n r group;
      datalines;
   B  2.0  0  44 43  1   B  2.0  1  54 52  2
   B  1.5  1  36 32  3   B  1.5  0  45 40  4
   A  2.0  0  66 64  5   A  2.0  1  89 89  6
   A  1.5  1  45 39  7   A  1.5  0  66 60  8
   B  2.0  0  44 44  1   B  2.0  1  54 54  2
   B  1.5  1  36 30  3   B  1.5  0  45 41  4
   A  2.0  0  66 65  5   A  2.0  1  89 88  6
   A  1.5  1  45 38  7   A  1.5  0  66 59  8
   ;

proc probit optc data=validate inest=out1;
   class treat sex;
   model r/n = dose treat sex sex*treat  / maxit = 0 ;
   output out = out3 p =p;
run ;

proc probit optc lackfit data=validate inest=out1;
   class treat sex;
   model r/n = dose treat sex sex*treat / aggregate = group ;
   output out = out4 p =p;
run ;
```

After the first invocation of PROC PROBIT, you have the estimated parameters and their covariance matrix in the data set OUTEST = Out1, and the fitted probabilities for the training data set epidemic in the data set OUTPUT = Out2. See Output 71.4.10 for the data set Out1 and Output 71.4.14 for the data set Out2.

The validation data are collected in data set validate. The second invocation of PROC PROBIT simply passes the estimated parameters from the training data set epidemic to the validation data set validate for prediction. The predicted probabilities are stored in the data set OUTPUT = Out3 (see Output 71.4.15). The third invocation of PROC PROBIT passes the estimated parameters as initial values for a new fit of the validation data set with the same model. Predicted probabilities are stored in the data set OUTPUT = Out4 (see Output 71.4.16). Goodness-of-fit tests are computed based on the cells grouped by the AGGREGATE= group variable. Results are shown in Output 71.4.17.

Output 71.4.14 Out2

Obs	treat	dose	n	r	sex	p
1	A	2.17	142	142	0	0.99272
2	A	0.57	132	47	1	0.35925
3	A	1.68	128	105	1	0.81899
4	A	1.08	126	100	0	0.77517
5	A	1.79	125	118	0	0.96682
6	B	1.66	117	115	1	0.97901
7	B	1.49	127	114	0	0.90896
8	B	1.17	51	44	1	0.89749
9	B	2.00	127	126	0	0.98364
10	B	0.80	129	100	1	0.76414

Output 71.4.15 Out3

Obs	treat	dose	sex	n	r	group	p
1	B	2.0	0	44	43	1	0.98364
2	B	2.0	1	54	52	2	0.99506
3	B	1.5	1	36	32	3	0.96247
4	B	1.5	0	45	40	4	0.91145
5	A	2.0	0	66	64	5	0.98500
6	A	2.0	1	89	89	6	0.91835
7	A	1.5	1	45	39	7	0.74300
8	A	1.5	0	66	60	8	0.91666
9	B	2.0	0	44	44	1	0.98364
10	B	2.0	1	54	54	2	0.99506
11	B	1.5	1	36	30	3	0.96247
12	B	1.5	0	45	41	4	0.91145
13	A	2.0	0	66	65	5	0.98500
14	A	2.0	1	89	88	6	0.91835
15	A	1.5	1	45	38	7	0.74300
16	A	1.5	0	66	59	8	0.91666

Output 71.4.16 Out4

Obs	treat	dose	sex	n	r	group	p
1	B	2.0	0	44	43	1	0.98954
2	B	2.0	1	54	52	2	0.98262
3	B	1.5	1	36	32	3	0.86187
4	B	1.5	0	45	40	4	0.90095
5	A	2.0	0	66	64	5	0.98768
6	A	2.0	1	89	89	6	0.98614
7	A	1.5	1	45	39	7	0.88075
8	A	1.5	0	66	60	8	0.88964
9	B	2.0	0	44	44	1	0.98954
10	B	2.0	1	54	54	2	0.98262
11	B	1.5	1	36	30	3	0.86187
12	B	1.5	0	45	41	4	0.90095
13	A	2.0	0	66	65	5	0.98768
14	A	2.0	1	89	88	6	0.98614
15	A	1.5	1	45	38	7	0.88075
16	A	1.5	0	66	59	8	0.88964

Output 71.4.17 Goodness-of-Fit Table

The Probit Procedure

Goodness-of-Fit Tests

Statistic	Value	DF	Value/DF	Pr > ChiSq
Pearson Chi-Square	2.8101	2	1.4050	0.2454
L.R. Chi-Square	2.8080	2	1.4040	0.2456

References

Agresti, A. (1990), *Categorical Data Analysis*, New York: John Wiley & Sons.

Collett, D (1991), *Modelling Binary Data*, London: Chapman & Hall.

Cox, D. R. (1970), *Analysis of Binary Data*, London: Chapman & Hall.

Cox, D. R. and Oakes, D. (1984), *Analysis of Survival Data*, London: Chapman & Hall.

Finney, D. J. (1971), *Probit Analysis*, Third Edition, London: Cambridge University Press.

Hubert, J. J., Bohidar, N. R., and Peace, K. E. (1988), "Assessment of Pharmacological Activity," in *Biopharmaceutical Statistics for Drug Development*, ed. K. E. Peace, New York: Marcel Dekker.

Chapter 72

The QUANTREG Procedure

Contents

Overview: QUANTREG Procedure **5358**
Features 5360
Quantile Regression 5361
Getting Started: QUANTREG Procedure **5362**
Analysis of Fish-Habitat Relationships 5363
Growth Charts for Body Mass Index 5368
Syntax: QUANTREG Procedure **5372**
PROC QUANTREG Statement 5372
BY Statement 5377
CLASS Statement 5377
EFFECT Statement (Experimental) 5378
ID Statement 5378
MODEL Statement 5378
OUTPUT Statement 5381
PERFORMANCE Statement 5382
TEST Statement 5383
WEIGHT Statement 5383
Details: QUANTREG Procedure **5384**
Quantile Regression as an Optimization Problem 5384
Optimization Algorithms 5385
Confidence Interval 5392
Covariance-Correlation 5396
Linear Test 5396
Leverage Point and Outlier Detection 5397
INEST= Data Set 5398
OUTEST= Data Set 5399
Computational Resources 5399
ODS Table Names 5400
ODS Graphics 5401
Examples: QUANTREG Procedure **5406**
Example 72.1: Comparison of Algorithms 5406
Example 72.2: Quantile Regression for Econometric Growth Data 5411
Example 72.3: Quantile Regression Analysis of Birth-Weight Data 5419
Example 72.4: Nonparametric Quantile Regression for Ozone Levels 5424

Example 72.5: Quantile Polynomial Regression for Salary Data 5426
References 5430

Overview: QUANTREG Procedure

The QUANTREG procedure models the effects of covariates on the conditional quantiles of a response variable by means of quantile regression.

Ordinary least squares (OLS) regression models the relationship between one or more covariates X and the *conditional mean* of the response variable Y given $X = x$. Quantile regression, which was introduced by Koenker and Bassett (1978), extends the regression model to *conditional quantiles* of the response variable, such as the median or the 90th percentile. Quantile regression is particularly useful when the rate of change in the conditional quantile, expressed by the regression coefficients, depends on the quantile.

Figure 72.1 Trout Density in Streams

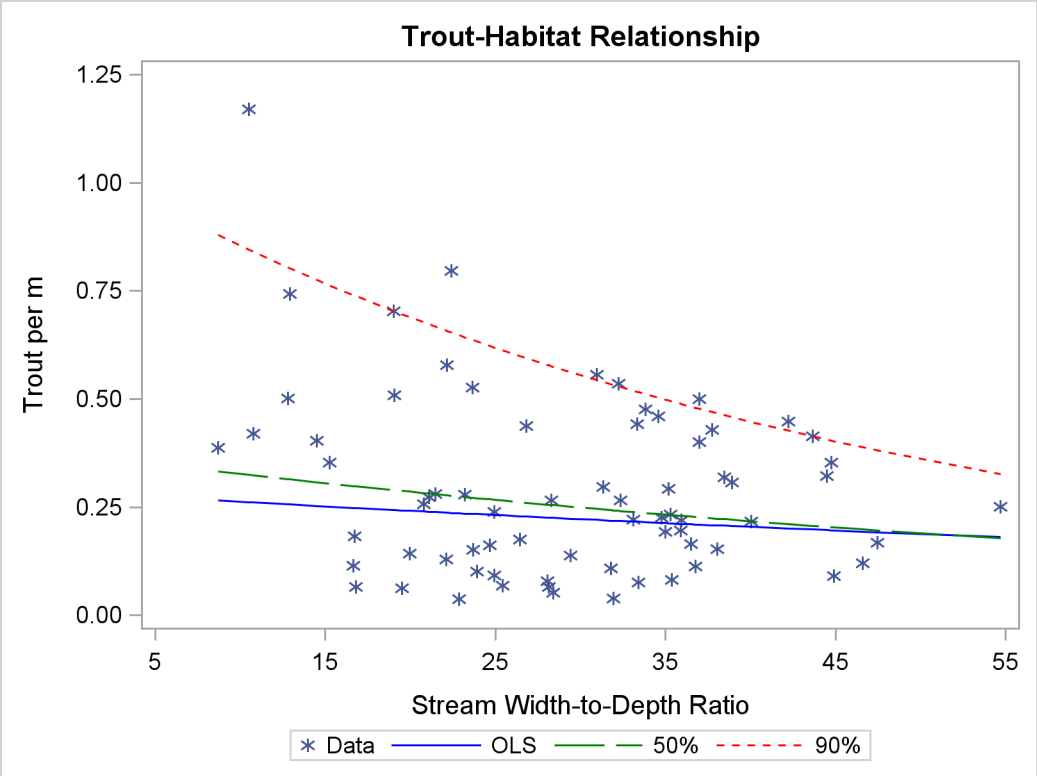


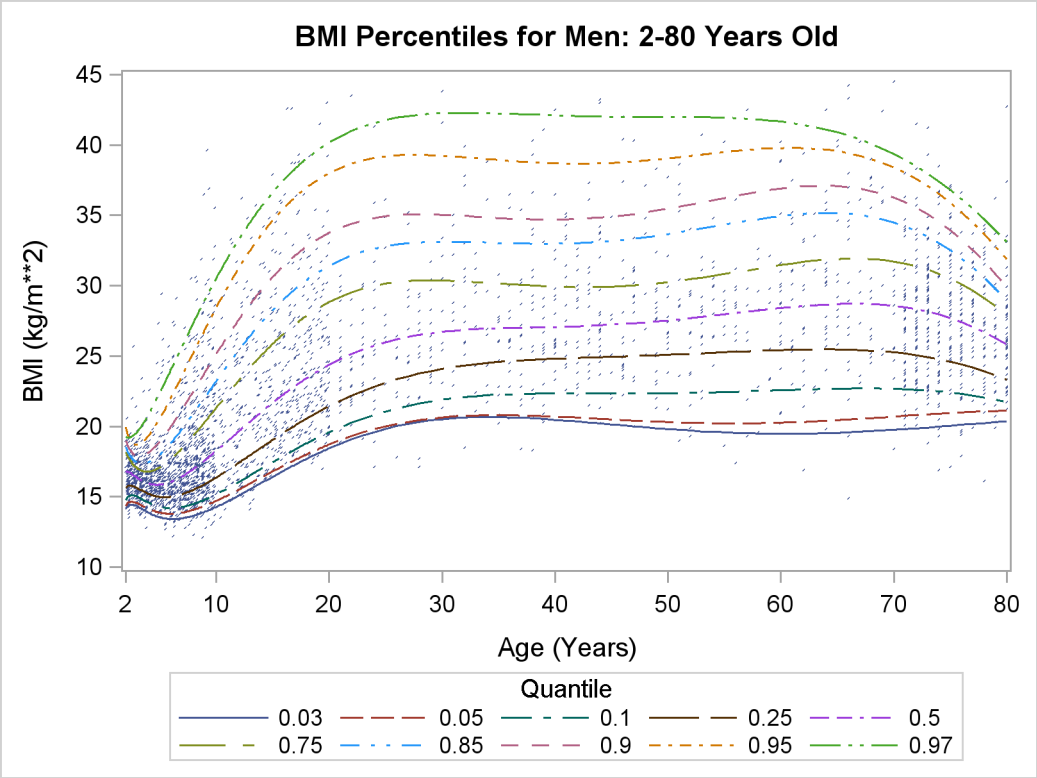
Figure 72.1 illustrates an ecological study in which it is revealing to model upper conditional quantiles. The points represent measurements of trout density and stream width-to-depth ratio taken at 13 streams over seven years.

As analyzed by Dunham, Cade, and Terrell (2002), in addition to the ratio, trout density depends on

a number of unmeasured limiting factors related to the integrity of stream habitat. The interaction of these factors results in unequal variances for the conditional distributions of density given the ratio. When the ratio is the "active" limiting effect, changes in the upper conditional percentiles of density provide a better estimate of this effect than changes in the conditional mean.

The two dashed curves represent the conditional 90th and 50th percentiles of density as determined with the QUANTREG procedure. The analysis was done by using a simple linear regression model for the logarithm of density. (The curves in Figure 72.1 were obtained by transforming the fitted lines back to the original scale. For more details, see the section "Analysis of Fish-Habitat Relationships" on page 5363.) The slope parameter for the 90th percentile has an estimated value of -0.0215 and is significant with a p-value less than 0.01. On the other hand, the slope parameter for the 50th percentile is not significantly different from zero. Similarly, the slope parameter for the mean, obtained with OLS regression, is not significantly different from zero.

Figure 72.2 Quantiles for Body Mass Index



Quantile regression is especially useful with data that are heterogeneous in the sense that the tails and the central location of the conditional distributions vary differently with the covariates. An even more pronounced example of heterogeneity is shown in Figure 72.2, which plots the body mass index of 8,250 men versus their age.

Here, both upper (overweight) and lower (underweight) conditional quantiles are important because they provide the basis for developing growth charts and establishing health standards. The curves in Figure 72.2 were determined with the QUANTREG procedure by using polynomial quantile regression; details are provided in the section "Growth Charts for Body Mass Index" on page 5368. Clearly, the rate of change with age (as expressed by the regression coefficients), particularly for ages less than 20, is different for each conditional quantile.

Heterogeneous data occur in many fields, including biomedicine, econometrics, survival analysis, and ecology. Quantile regression, which includes median regression as a special case, provides a complete picture of the covariate effect when a set of percentiles is modeled, and so it offers the capability to capture important features of the data that might be missed by models that average over the conditional distribution.

Because it makes no distributional assumption about the error term in the model, quantile regression offers considerable model robustness. The assumption of normality, which is often made with OLS regression in order to compute conditional quantiles as offsets from the mean, forces a common set of regression coefficients for all the quantiles. Obviously, quantiles with common slopes would be inappropriate in the preceding examples.

Quantile regression is also flexible in the sense that it does not involve a link function that relates the variance and the mean of the response variable. Generalized linear models, which you can fit with the GENMOD procedure, require both a link function and a distributional assumption such as the normal or Poisson distribution. The goal of generalized linear models is inference about the regression parameters in the linear predictor for the mean of the population. In contrast, the goal of quantile regression is inference on regression coefficients for the conditional quantiles of a response variable that is usually assumed to be continuous.

Quantile regression also offers a degree of data robustness. Unlike OLS regression, it is robust to extreme points in the response direction (outliers). However, it is not robust to extreme points in the covariate space (leverage points). When both types of robustness are of concern, you should consider using the ROBUSTREG procedure (Chapter 74, "The ROBUSTREG Procedure.")

Also, unlike OLS regression, quantile regression is equivariant to monotone transformations of the response variable. For instance, as illustrated in the trout example, the logarithm of the 90th conditional percentile of trout density is the 90th conditional percentile of the logarithm of density.

Note that quantile regression cannot be carried out simply by segmenting the unconditional distribution of the response variable and then obtaining least squares fits for the subsets. This approach leads to disastrous results when, for example, the data include outliers. In contrast, quantile regression uses *all* of the data for fitting quantiles, even the extreme quantiles.

Features

The main features of the QUANTREG procedure are as follows:

- offers simplex, interior point, and smoothing algorithms for estimation
- provides sparsity, rank, and resampling methods for confidence intervals
- provides asymptotic and bootstrap methods for covariance and correlation matrices of the estimated parameters
- provides the Wald and likelihood ratio tests for the regression parameter estimates
- provides outlier and leverage-point diagnostics

- enables parallel computing when multiple processors are available
- provides row-wise or column-wise output data sets with multiple quantiles
- provides regression quantile spline fits
- produces fit plots, diagnostic plots, and quantile process plots by using ODS Graphics

The next section provides notation and a formal definition for quantile regression.

Quantile Regression

Quantile regression generalizes the concept of a univariate quantile to a conditional quantile given one or more covariates. Recall that a student's score on a test is at the τth quantile if his or her score is better than that of $100\tau\%$ of the students who took the test. The score is also said to be at the 100τth percentile.

For a random variable Y with probability distribution function

$$F(y) = \text{Prob}\,(Y \leq y)$$

the τth quantile of Y is defined as the inverse function

$$Q(\tau) = \inf\{y : F(y) \geq \tau\}$$

where $0 < \tau < 1$. In particular, the median is $Q(1/2)$.

For a random sample $\{y_1, \ldots, y_n\}$ of Y, it is well known that the sample median minimizes the sum of absolute deviations

$$\text{median} = \arg\min_{\xi \in \mathbf{R}} \sum_{i=1}^{n} |y_i - \xi|$$

Likewise, the general τth sample quantile $\xi(\tau)$, which is the analog of $Q(\tau)$, is formulated as the minimizer

$$\xi(\tau) = \arg\min_{\xi \in \mathbf{R}} \sum_{i=1}^{n} \rho_\tau(y_i - \xi)$$

where $\rho_\tau(z) = z(\tau - I(z < 0))$, $0 < \tau < 1$, and where $I(\cdot)$ denotes the indicator function. The loss function ρ_τ assigns a weight of τ to positive residuals $y_i - \xi$ and a weight of $1 - \tau$ to negative residuals.

Using this loss function, the linear conditional quantile function extends the τth sample quantile $\xi(\tau)$ to the regression setting in the same way that the linear conditional mean function extends the sample mean. Recall that OLS regression estimates the linear conditional mean function $E(Y|X = x) = x'\beta$ by solving for

$$\hat{\beta} = \arg\min_{\beta \in \mathbf{R}^p} \sum_{i=1}^{n} (y_i - x_i'\beta)^2$$

The estimated parameter $\hat{\beta}$ minimizes the sum of squared residuals in the same way that the sample mean $\hat{\mu}$ minimizes the sum of squares:

$$\hat{\mu} = \arg\min_{\mu \in \mathbf{R}} \sum_{i=1}^{n} (y_i - \mu)^2$$

Likewise, quantile regression estimates the linear conditional quantile function, $Q(\tau|X = x) = x'\beta(\tau)$, by solving

$$\hat{\beta}(\tau) = \arg\min_{\beta \in \mathbf{R}^p} \sum_{i=1}^{n} \rho_\tau(y_i - x_i'\beta)$$

for any quantile $\tau \in (0, 1)$. The quantity $\hat{\beta}(\tau)$ is called the τth *regression quantile*. The case $\tau = 0.5$, which minimizes the sum of absolute residuals, corresponds to median regression, which is also known as L_1 regression.

The set of regression quantiles

$$\{\beta(\tau) : \tau \in (0, 1)\}$$

is referred to as the *quantile process*.

The QUANTREG procedure computes the quantile function $Q(\tau|X = x)$ and conducts statistical inference on the estimated parameters $\hat{\beta}(\tau)$.

Getting Started: QUANTREG Procedure

The following examples demonstrate how you can use the QUANTREG procedure to fit linear models for selected quantiles or for the entire quantile process. The first example explains the use of the procedure in the fish-habitat example, and the second example explains the use of the procedure to construct growth charts for body mass index.

Analysis of Fish-Habitat Relationships

Quantile regression is used extensively in ecological studies (Cade and Noon 2003). Recently, Dunham, Cade, and Terrell (2002) applied quantile regression to analyze fish-habitat relationships for Lahontan cutthroat trout in 13 streams of the eastern Lahontan basin, which covers most of northern Nevada and parts of southern Oregon. The density of trout (number of trout per meter) was measured by sampling stream sites from 1993 to 1999. The width-to-depth ratio of the stream site was determined as a measure of stream habitat.

The goal of this study was to explore the relationship between the conditional quantiles of trout density and the width-to-depth ratio. The scatter plot of the data in Figure 72.1 indicates a nonlinear relationship, and so it is reasonable to fit regression models for the conditional quantiles of the log of density. Since regression quantiles are equivariant under any monotonic (linear or nonlinear) transformation (Koenker and Hallock 2001), the exponential transformation converts the conditional quantiles to the original density scale.

The data set trout, which follows, includes the average numbers of Lahontan cutthroat trout per meter of stream (Density), the logarithm of Density (LnDensity), and the width-to-depth ratios (WDRatio) for 71 samples.

```
data trout;
   input Density WDRatio LnDensity @@;
   datalines;
0.38732     8.6819    -0.94850    1.16956    10.5102     0.15662
0.42025    10.7636    -0.86690    0.50059    12.7884    -0.69197
0.74235    12.9266    -0.29793    0.40385    14.4884    -0.90672
0.35245    15.2476    -1.04284    0.11499    16.6495    -2.16289
0.18290    16.7188    -1.69881    0.06619    16.7859    -2.71523
0.70330    19.0141    -0.35197    0.50845    19.0548    -0.67639
0.06279    19.4959    -2.76796    0.14190    19.9446    -1.95265
0.25725    20.7852    -1.35772    0.27240    21.0870    -1.30048
0.27983    21.4564    -1.27357    0.12860    22.0917    -2.05105
0.57867    22.1627    -0.54702    0.79667    22.4070    -0.22731
0.03730    22.8553    -3.28880    0.27897    23.2003    -1.27666
0.52587    23.6662    -0.64270    0.15075    23.6937    -1.89215
0.10071    23.9129    -2.29548    0.16128    24.6643    -1.82461
0.09254    24.9451    -2.38011    0.23937    24.9492    -1.42974
0.06914    25.4138    -2.67158    0.17586    26.4412    -1.73805
0.43725    26.8025    -0.82725    0.07812    28.0558    -2.54945
0.06576    28.1194    -2.72174    0.26539    28.3045    -1.32654
0.05159    28.3949    -2.96440    0.13779    29.4083    -1.98202
0.55589    30.9569    -0.58719    0.29714    31.3376    -1.21354
0.10857    31.7868    -2.22035    0.03897    31.9464    -3.24485
0.53572    32.2492    -0.62414    0.26580    32.3725    -1.32500
0.22114    33.1017    -1.50896    0.44212    33.3530    -0.81618
0.07646    33.4036    -2.57099    0.47616    33.8079    -0.74200
0.45934    34.5639    -0.77796    0.22627    34.7844    -1.48603
0.19356    35.0004    -1.64215    0.29216    35.1803    -1.23045
0.23243    35.2959    -1.45917    0.08155    35.3704    -2.50654
0.19528    35.9115    -1.63332    0.22023    35.9382    -1.51308
```

```
0.16411    36.4884    -1.80723    0.11296    36.7694    -2.18072
0.49981    36.9893    -0.69353    0.40012    37.0055    -0.91599
0.42912    37.7344    -0.84601    0.15294    38.0394    -1.87770
0.31935    38.4524    -1.14147    0.30667    38.9076    -1.18198
0.21722    40.0388    -1.52685    0.44777    42.2364    -0.80347
0.41371    43.6465    -0.88259    0.32136    44.4753    -1.13520
0.35369    44.7545    -1.03935    0.09101    44.9001    -2.39684
0.11982    46.6135    -2.12175    0.16831    47.4509    -1.78197
0.25125    54.6916    -1.38129
;
```

The following statements use the QUANTREG procedure to fit a simple linear model for the 90th percentile of LnDensity:

```
proc quantreg data=trout alpha=0.01 ci=resampling;
   model LnDensity = WDRatio / quantile=0.9
                               CovB CorrB
                               seed=12345;
   test WDRatio / wald lr;
run;
```

The MODEL statement specifies a simple linear regression model with LnDensity as the response variable Y and WDRatio as the covariate X. The option QUANTILE=0.9 requests that the regression quantile function $Q(0.9|X = x) = x'\beta(0.9)$ is to be estimated by solving

$$\hat{\beta}(0.9) = \arg\min_{\beta \in \mathbf{R}^2} \sum_{i=1}^{n} \rho_{0.9}(y_i - x_i'\beta)$$

By default, the regression coefficients $\hat{\beta}(0.9)$ are estimated with the simplex algorithm, which is explained in the section "Simplex Algorithm" on page 5385. The option ALPHA=0.01 requests 99% confidence limits for the regression parameters, and the option CI=RESAMPLING specifies that the intervals are to be computed with the MCMB resampling method of He and Hu (2002). By specifying the CI=RESAMPLING option, the QUANTREG procedure also computes standard errors, t values, and p-values of regression parameters with the MCMB resampling method. The SEED= option specifies a seed for the resampling method. The options COVB and CORRB request covariance and correlation matrices for the estimated regression coefficients, and the TEST statement requests tests for the hypothesis that the slope parameter (the coefficient of WDRatio) is zero.

Figure 72.3 displays model information and summary statistics for the variables in the model. The summary statistics include the median and the standardized median absolute deviation (MAD), which are robust measures of univariate location and scale, respectively. See Huber (1981, p. 108) for more details about the standardized MAD.

Figure 72.3 Model Fitting Information and Summary Statistics

The QUANTREG Procedure

Model Information

Data Set	WORK.TROUT
Dependent Variable	LnDensity
Number of Independent Variables	1
Number of Observations	71
Optimization Algorithm	Simplex
Method for Confidence Limits	Resampling

Summary Statistics

Variable	Q1	Median	Q3	Mean	Standard Deviation	MAD
WDRatio	22.0917	29.4083	35.9382	29.1752	9.9859	10.4970
LnDensity	-2.0511	-1.3813	-0.8669	-1.4973	0.7682	0.8214

Figure 72.4 displays the parameter estimates, standard errors, 99% confidence limits, t values, and p-values computed by the resampling method.

Figure 72.4 Parameter Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	99% Confidence Limits		t Value	Pr > |t|
Intercept	1	0.0576	0.2727	-0.6648	0.7801	0.21	0.8333
WDRatio	1	-0.0215	0.0073	-0.0408	-0.0022	-2.96	0.0042

The 90th percentile of trout density can be predicted from the width-to-depth ratio as follows:

$$y_{90} = \exp(0.0576 - 0.0215x)$$

This is the upper dashed curve plotted in Figure 72.1. The lower dashed curve for the median can be obtained by changing the option QUANTILE=0.9 to QUANTILE=0.5.

The covariance and correlation matrices for the estimated parameters are shown in Figure 72.5. The resampling method used for the confidence intervals is used to compute these matrices.

Figure 72.5 Covariance and Correlation

Estimated Covariance Matrix

	Intercept	WDRatio
Intercept	0.074384	-.001934
WDRatio	-.001934	0.000053

Figure 72.5 *continued*

Estimated Correlation Matrix

	Intercept	WDRatio
Intercept	1.00000	-0.97357
WDRatio	-0.97357	1.00000

The tests requested with the TEST statement are shown in Figure 72.6. Both the Wald test and the likelihood ratio test indicate that the coefficient of width-to-depth ratio is significantly different from zero.

Figure 72.6 Tests of Significance

Tests

Test	Test Statistic	DF	Chi-Square	Pr > ChiSq
Wald	8.7467	1	8.75	0.0031
Likelihood Ratio	9.0529	1	9.05	0.0026

In many quantile regression problems it is useful to examine how the estimated regression parameters for each covariate change as a function of the quantile τ in the interval $(0, 1)$. The following statements use the QUANTREG procedure to request the estimated quantile processes $\hat{\beta}(\tau)$ for the slope and intercept parameters:

```
ods graphics on;

proc quantreg data=trout alpha=0.1 ci=resampling;
   model LnDensity = WDRatio / quantile=process seed=12345
                               plot=quantplot;
run;

ods graphics off;
```

The QUANTILE=PROCESS option requests an estimate of the quantile process for each regression parameter. The options ALPHA=0.1 and CI=RESAMPLING specify that 90% confidence bands for the quantile processes are to be computed with the resampling method.

Figure 72.7 displays a portion of the objective function table for the entire quantile process. The objective function is evaluated at 77 values of τ in the interval $(0, 1)$. The table also provides predicted values of the conditional quantile function $Q(\tau)$ at the mean for WDRatio, which can be used to estimate the conditional density function.

Figure 72.7 Objective Function

Objective Function for Quantile Process

Label	Quantile	Objective Function	Predicted at Mean
t0	0.005634	0.7044	−3.2582
t1	0.020260	2.5331	−3.0331
t2	0.031348	3.7421	−2.9376
t3	0.046131	5.2538	−2.7013
.	.	.	.
.	.	.	.
.	.	.	.
t73	0.945705	4.1433	−0.4361
t74	0.966377	2.5858	−0.4287
t75	0.976060	1.8512	−0.4082
t76	0.994366	0.4356	−0.4082

Figure 72.8 displays a portion of the table of the quantile processes for the estimated parameters and confidence limits.

Figure 72.8 Objective Function

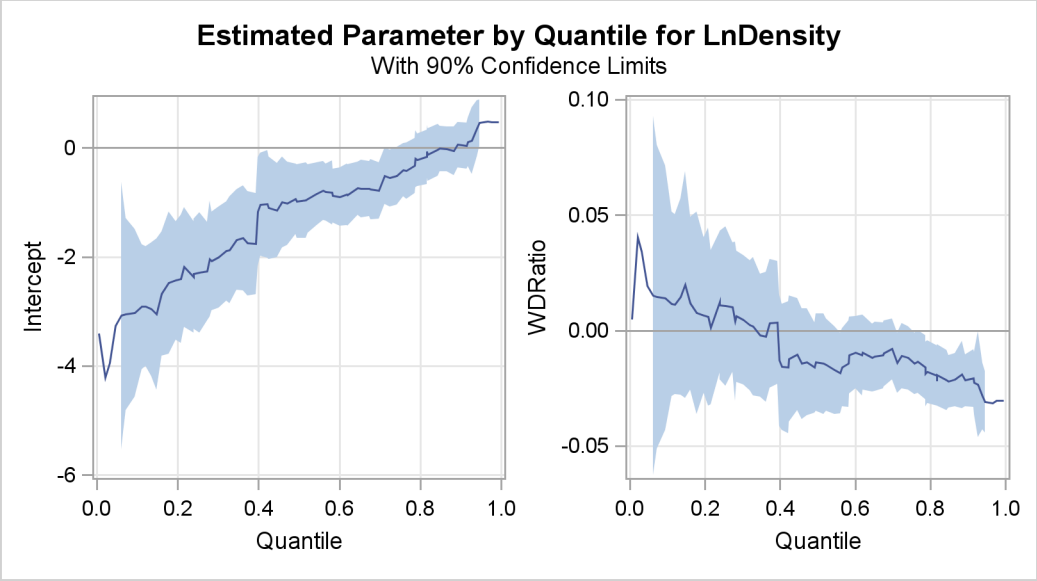
Parameter Estimates for Quantile Process

Label	Quantile	Intercept	WDRatio
.	.	.	.
.	.	.	.
.	.	.	.
t57	0.765705	−0.42205	−0.01335
lower90	0.765705	−0.94184	−0.02696
upper90	0.765705	0.09773	0.00027
t58	0.786206	−0.32688	−0.01592
lower90	0.786206	−0.83355	−0.02984
upper90	0.786206	0.17978	−0.00200
.	.	.	.
.	.	.	.
.	.	.	.

The PLOT=QUANTPLOT option in the MODEL statement, together with the ODS GRAPHICS statement, requests a plot of the estimated quantile processes. The left side of Figure 72.9 displays the process for the intercept, and the right side displays the process for the coefficient of WDRatio.

The process plot for WDRatio shows that the slope parameter changes from positive to negative as the quantile increases, and it changes sign with a sharp drop at the 40th percentile. The 90% confidence bands show that the relationship between LnDensity and WDRatio (expressed by the slope) is not significant below the 78th percentile. This situation can also be seen in Figure 72.8, which shows that 0 falls between the lower and upper confidence limits of the slope parameter for quantiles below 0.78. Since the confidence intervals for the extreme quantiles are not stable due to insufficient data, the confidence band is not displayed outside the interval (0.05, 0.95).

Figure 72.9 Quantile Processes for Intercept and Slope



Growth Charts for Body Mass Index

Body mass index (BMI) is defined as the ratio of weight (kg) to squared height (m^2) and is a widely used measure for categorizing individuals as overweight or underweight. The percentiles of BMI for specified ages are of particular interest. As age increases, these percentiles provide growth patterns of BMI not only for the majority of the population, but also for underweight or overweight extremes of the population. In addition, the percentiles of BMI for a specified age provide a reference for individuals at that age with respect to the population.

Smooth quantile curves have been widely used for reference charts in medical diagnosis to identify unusual subjects, whose measurements lie in the tails of the reference distribution. This example explains how to use the QUANTREG procedure to create growth charts for BMI.

A SAS data set named bmimen was created by merging and cleaning the 1999–2000 and 2001–2002 survey results for men published by the National Center for Health Statistics. This data set contains the variables WEIGHT (kg), HEIGHT (m), BMI (kg/m^2), AGE (year), and SEQN (respondent sequence number) for 8,250 men. More details can be found in Chen (2005).

The data set used in this example is a subset of the original data set of Chen (2005). It contains the two variables BMI and AGE with 3264 observations.

```
data bmimen0;
   input bmi age @@;
datalines;
18.6  2.0 17.1  2.0 19.0  2.0 16.8  2.0 19.0  2.1  15.5   2.1
16.7  2.1 16.1  2.1 18.0  2.1 17.8  2.1 18.3  2.1  16.9   2.1
15.9  2.1 20.6  2.1 16.7  2.1 15.4  2.1 15.9  2.1  17.7   2.1
15.7  2.1 16.8  2.1 15.6  2.1 18.1  2.1 15.7  2.1  17.2   2.1
14.5  2.2 17.2  2.2 16.3  2.2 15.4  2.2 16.0  2.2  15.8   2.2

... more lines ...

29.0 80.0 24.1 80.0 26.6 80.0 24.2 80.0 22.7 80.0  28.4  80.0
26.3 80.0 25.6 80.0 24.8 80.0 28.6 80.0 25.7 80.0  25.8  80.0
22.5 80.0 25.1 80.0 27.0 80.0 27.9 80.0 28.5 80.0  21.7  80.0
33.5 80.0 26.1 80.0 28.4 80.0 22.7 80.0 28.0 80.0  42.7  80.0
;
```

The logarithm of BMI is used as the response (although this does not improve the quantile regression fit, it helps with statistical inference.) A preliminary median regression is fitted with a parametric model, which involves six powers of AGE.

```
data bmimen;
   set bmimen0;
   sqrtage = sqrt(age);
   inveage = 1/age;
   logbmi  = log(bmi);
run;
```

The following statements invoke the QUANTREG procedure:

```
proc quantreg data=bmimen algorithm=interior ci=resampling;
   model logbmi = inveage sqrtage age sqrtage*age
                  age*age age*age*age
                  / diagnostics cutoff=4.5 quantile=.5;
   id age bmi;
   test_age_cubic: test age*age*age / wald lr;
run;
```

The MODEL statement provides the model, and the option QUANTILE=0.5 requests median regression, which computes $\hat{\beta}(\frac{1}{2})$ by using the interior point algorithm as requested with the ALGORITHM= option. See the section "Interior Point Algorithm" on page 5386 for details about this algorithm.

Figure 72.10 displays the estimated parameters, standard errors, 95% confidence intervals, t values, and p-values that are computed by the resampling method as requested by the CI= option. All of the parameters are considered significant since the p-values are smaller than 0.001.

Figure 72.10 Parameter Estimates with Median Regression: Men

The QUANTREG Procedure

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		t Value	Pr > |t|
Intercept	1	7.8909	0.7755	6.3704	9.4115	10.17	<.0001
inveage	1	−1.8354	0.4187	−2.6563	−1.0144	−4.38	<.0001
sqrtage	1	−5.1247	0.6765	−6.4511	−3.7983	−7.58	<.0001
age	1	1.9759	0.2410	1.5033	2.4484	8.20	<.0001
sqrtage*age	1	−0.3347	0.0405	−0.4141	−0.2554	−8.27	<.0001
age*age	1	0.0227	0.0028	0.0172	0.0282	8.12	<.0001
age*age*age	1	−0.0000	0.0000	−0.0001	−0.0000	−7.69	<.0001

The TEST statement requests Wald and likelihood ratio tests for the significance of the cubic term in AGE. The test results, shown in Figure 72.11, indicate that this term is significant. Higher-order terms are not significant.

Figure 72.11 Test of Significance for Cubic Term

Tests

test_age_cubic

Test	Test Statistic	DF	Chi-Square	Pr > ChiSq
Wald	59.0752	1	59.08	<.0001
Likelihood Ratio	56.9473	1	56.95	<.0001

Median regression and, more generally, quantile regression are robust to extremes of the response variable. The DIAGNOSTICS option in the MODEL statement requests a diagnostic table of outliers, shown in Figure 72.12, which uses a cutoff value specified with the CUTOFF= option. The variables specified in the ID statement are included in the table.

With CUTOFF=4.5, 14 men are identified as outliers. All of these men have large positive standardized residuals, which indicates that they are overweight for their age. The cutoff value 4.5 is ad hoc; it corresponds to a probability less than 0.5E−5 if normality is assumed, but the standardized residuals for median regression usually do not meet this assumption.

In order to construct the chart shown in Figure 72.2, the same model used for median regression is used for other quantiles. Note that the QUANTREG procedure can compute fitted values for multiple quantiles.

Figure 72.12 Diagnostics with Median Regression

Diagnostics

Obs	age	bmi	Standardized Residual	Outlier
1337	8.900000	36.500000	5.3575	*
1376	9.200000	39.600000	5.8723	*
1428	9.400000	36.900000	5.3036	*
1505	9.900000	35.500000	4.8862	*
1764	14.900000	46.800000	5.6403	*
1838	16.200000	50.400000	5.9138	*
1845	16.300000	42.600000	4.6683	*
1870	16.700000	42.600000	4.5930	*
1957	18.100000	49.900000	5.5053	*
2002	18.700000	52.700000	5.8106	*
2016	18.900000	48.400000	5.1603	*
2264	32.000000	55.600000	5.3085	*
2291	35.000000	60.900000	5.9406	*
2732	66.000000	14.900000	-4.7849	*

The following statements request fitted values for 10 quantiles ranging from 0.03 to 0.97:

```
proc quantreg data=bmimen ci=none algorithm=interior(tolerance=1e-6);
model logbmi = inveage sqrtage age sqrtage*age
                 age*age age*age*age
                 / quantile=0.03,0.05,0.1,0.25,0.5,0.75,
                            0.85,0.90,0.95,0.97;
output out=outp pred=p/columnwise;
run;

data outbmi;
   set outp;
   pbmi = exp(p);
run;

proc sgplot data=outbmi;
   title 'BMI Percentiles for Men: 2-80 Years Old';
   yaxis label='BMI (kg/m**2)' min=10 max=45
         values=(10 15 20 25 30 35 40 45);
   xaxis label='Age (Years)' min=2 max=80
         values=(2 10 20 30 40 50 60 70 80);
   scatter x=age y=bmi /markerattrs=(size=1);
   series  x=age y=pbmi/group=QUANTILE;
run;
```

The fitted values are stored in the OUTPUT data set outp. The COLUMNWISE option arranges these fitted values for all quantiles in the single variable p by groups of the quantiles. After the exponential transformation, the fitted BMI values together with the original BMI values are plotted against AGE to create the display shown in Figure 72.2.

The fitted quantile curves reveal important information. During the quick growth period (ages 2 to 20), the dispersion of BMI increases dramatically; it becomes stable during middle age, and then it

contracts after age 60. This pattern suggests that effective population weight control should start in childhood.

Compared to the 97th percentile in reference growth charts published by CDC in 2000 (Kuczmarski et al. 2002), the 97th percentile for 10-year-old boys in Figure 72.2 is 6.4 BMI units higher (an increase of 27%). This can be interpreted as a warning of overweight or obesity. See Chen (2005) for a detailed analysis.

Syntax: QUANTREG Procedure

PROC QUANTREG < *options* > ;
  BY *variables* ;
  CLASS *variables* ;
  EFFECT *name* = *effect-type* (*variables* < / *options* >) ;
  ID *variables* ;
  MODEL *response* = *independents* < / *options* > ;
  OUTPUT < *OUT= SAS-data-set* > < *options* > ;
  PERFORMANCE < *options* > ;
  TEST *effects* < / *options* > ;
  WEIGHT *variable* ;

The PROC QUANTREG statement invokes the procedure. The CLASS statement specifies which explanatory variables are treated as categorical. The ID statement names variables to identify observations in the outlier diagnostics tables. The MODEL statement is required and specifies the variables used in the regression. Main effects and interaction terms can be specified in the MODEL statement, as in the GLM procedure (Chapter 39, "The GLM Procedure.") The OUTPUT statement creates an output data set containing predicted values, residuals, and estimated standard errors. The PERFORMANCE statement tunes the performance of PROC QUANTREG by using single or multiple processors available in the hardware. The TEST statement requests linear tests for the model parameters. The WEIGHT statement identifies a variable in the input data set whose values are used to weight the observations. In one invocation of PROC QUANTREG, multiple OUTPUT and TEST statements are allowed.

PROC QUANTREG Statement

PROC QUANTREG < *options* > ;

The PROC QUANTREG statement invokes the procedure. You can specify the following options in the PROC QUANTREG statement.

ALGORITHM=*algorithm* **< (** *suboptions* **) >**
  specifies an algorthm to estimate the regression parameters. Three algorithms are available:

simplex (SIMPLEX), interior point (INTERIOR), and smoothing (SMOOTH). The default algorithm is simplex. The following table summarizes these methods.

Table 72.1 Options for Estimation Algorithms

ALGORITHM= Value	Algorithm	Suboptions
SIMPLEX	simplex	
INTERIOR	interior point	KAPPA= MAXIT= TOLERANCE=
SMOOTH	smoothing	RRATIO=

With ALGORITHM=INTERIOR you can specify the following *suboptions*:

- KAPPA=*value* specifies the step length parameter for the interior point algorithm. This parameter should be between 0 and 1. The larger the parameter, the faster the algorithm. However, numeric instability can occur as the parameter approaches 1. By default, KAPPA=0.99995. See the section "Interior Point Algorithm" on page 5386 for details.
- MAXIT=*n* sets the maximum number of iterations for the interior point algorithm. By default, *n*=1000.
- TOLERANCE=*value* specifies the tolerance for the convergence criterion of the interior point algorithm. The default *value* is 1E−8. The QUANTREG procedure uses the duality gap as the convergence criterion. See the section "Interior Point Algorithm" on page 5386 for details.

With the interior point algorithm, you can use the PERFORMANCE statement to enable parallel computing when multiple processors are available in the hardware.

With ALGORITHM=SMOOTH you can specify the following *suboption*:

- RRATIO=*value* specifies the reduction ratio for the smoothing algorithm. This ratio is used for reducing the threshold of the smoothing algorithm. The *value* should be between 0 and 1. In theory, the smaller the reduction ration, the faster the smoothing algorithm. However, the optimal ratio is quite data dependent in practice. See the section "Smoothing Algorithm" on page 5390 for details.

ALPHA=*value*

sets the confidence level for the confidence intervals for regression parameters. The *value* must be between 0 and 1. The default is ALPHA=0.05, corresponding to a 0.95 confidence interval.

CI=NONE | RANK | SPARSITY<(BF | HS)></IID> | RESAMPLING<(NREP=*n*)>

specifies a method to compute confidence intervals for regression parameters. When you specify CI=SPARSITY or CI=RESAMPLING, the QUANTREG procedure also computes standard errors, t values, and p-values for regression parameters.

The following table summarizes these methods.

Table 72.2 Options for Confidence Intervals

Value of CI=	Method	Additional Options
NONE	no confidence intervals computed	
RANK	by inverting rank-score tests	
SPARSITY	by estimating sparsity function	HS BF IID
RESAMPLING	by resampling	NREP

By default, the QUANTREG procedure computes confidence intervals by using the rank method for data sets with fewer than 5,000 observations and fewer than 20 variables; otherwise, the default is the resampling method.

By default, confidence intervals are not computed for the quantile process, which is estimated when you specify the QUANTILE=PROCESS option in the MODEL statement. Confidence intervals for the quantile process are computed with the sparsity or resampling methods when you specify CI=SPARSITY or CI=RESAMPLING, respectively. The rank method for confidence intervals is not available with quantile processes because it is computationally prohibitive.

With the SPARSITY option, there are two suboptions for estimating the sparsity function. If you specify the IID suboption, the sparsity function is estimated by assuming that the errors in the linear model are i.i.d. By default, the sparsity function is estimated by assuming that the conditional quantile function is locally linear. See the section "Sparsity" on page 5393 for details. With both methods two bandwidth selection methods are available. You can specify the Bofinger method with the BF suboption or the Hall-Sheather method with the HS suboption. By default, the Hall-Sheather method is used.

With the RESAMPLING option, you can specify the number of repeats with the NREP=*n* suboption. By default, NREP=200. The value of *n* must be greater than 50.

DATA=*SAS-data-set*

specifies the input SAS data set used by the QUANTREG procedure. By default, the most recently created SAS data set is used.

INEST=*SAS-data-set*

specifies an input SAS data set that contains initial estimates for all the parameters in the model. The interior point algorithm and the smoothing algorithm use these estimates as a start. See the section "INEST= Data Set" on page 5398 for a detailed description of the contents of the INEST= data set.

NAMELEN=*n*

specifies the length of effect names in tables and output data sets to be *n* characters, where *n* is a value between 20 and 200. The default length is 20 characters.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of the classification variables (specified in the CLASS statement). This order determines which parameters in the model correspond to each level in

the data. The following table illustrates how the QUANTREG procedure interprets values of the ORDER= option.

Table 72.3 Options for Order

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	formatted value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent. For more information about sorting order, refer to the chapter titled "The SORT Procedure" in the *Base SAS Procedures Guide*.

OUTEST=*SAS-data-set*

specifies an output SAS data set containing the parameter estimates for all quantiles. See the section "OUTEST= Data Set" on page 5399 for a detailed description of the contents of the OUTEST= data set.

PLOT | PLOTS<(*global-plot-options***)> <=***plot-request***>**

PLOT | PLOTS<(*global-plot-options***)> <=(***plot-request* **< . . .** *plot-request* **>)>**

specifies options that control details of the plots. These plots fall into two categories, diagnostic plots and fit plots. If you do not specify the PLOTS= option, PROC QUANTREG produces the quantile fit plot by default when a single continuous variable is specified in the model. You can use the PLOTS= option in the PROC statement to request various diagnostic plots. In addition to these two categories of plots, you can use the PLOT= option in the MODEL statement to request the quantile process plot for any effects specified in the model.

To request any plots you must specify the ODS GRAPHICS statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

The *global-plot-options* apply to all plots generated by the QUANTREG procedure. The following global plot option is available:

ONLY

suppresses the default quantile fit plot. Only plots specifically requested are displayed.

You can specify more than one plot request within the parentheses after PLOTS=. For a single plot request, you can omit the parentheses. The following plot requests are available.

ALL

creates all appropriate plots.

DDPLOT<(LABEL=ALL | LEVERAGE | NONE | OUTLIER)>

creates a plot of robust distance against Mahalanobis distance. See the section "Leverage Point and Outlier Detection" on page 5397 for details about robust distance.

The LABEL= option specifies how the points on this plot are to be labeled, as summarized by the following table.

Table 72.4 Options for Label

Value of LABEL=	Label Method
ALL	label all points
LEVERAGE	label leverage points
NONE	no labels
OUTLIERS	label outliers

By default, the QUANTREG procedure labels both outliers and leverage points.

If you specify ID variables in the ID statement, the values of the first ID variable are used as labels; otherwise, observation numbers are used as labels.

FITPLOT<(NOLIMITS | NODATA)>
creates a plot of fitted conditional quantiles against the single continuous variable specified in the model. You can request this plot when only a single continuous variable is specified in the model in the form of one or more polynomials or spline effects. Multiple lines or curves are drawn on this plot if you specify several quantiles with the QUANTILE= option in the MODEL statement. For a single quantile, confidence limits are added to the plot by default. The NOLIMITS option suppresses these limits. The NODATA option suppresses the observed data, which are superimposed on the plot by default.

HISTOGRAM
creates a histogram for the standardized residuals based on the quantile regression estimates. The histogram is superimposed with a normal density curve and a kernel density curve.

NONE
suppresses all plots.

QQPLOT
creates the normal quantile-quantile plot for the standardized residuals based on the quantile regression estimates.

RDPLOT<(LABEL=ALL | LEVERAGE | NONE | OUTLIER)>
creates the plot of standardized residual against robust distance. See the section "Leverage Point and Outlier Detection" on page 5397 for details about robust distance. The LABEL= option specifies a label method for points on this plot. These label methods are described in Table 72.4.

By default, the QUANTREG procedure labels both outliers and leverage points.

If you specify ID variables in the ID statement, the values of the first ID variable are used as labels; otherwise, observation numbers are used as labels.

PP

requests preprocessing to speed up the interior point algorithm or the smoothing algorithm. The preprocessing uses a subsampling algorithm to reduce the original problem to a smaller one iteratively. It assumes that the data set is evenly distributed. Preprocessing should be used only for very large data sets, such as data sets with more than 100,000 observations. See Portnoy and Koenker (1997) for details.

BY Statement

BY *variables* **;**

You can specify a BY statement with the QUANTREG procedure to obtain separate analysis on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the QUANTREG procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

Variables that are classification variables rather than quantitative numeric variables must be listed in the CLASS statement. For each explanatory variable listed in the CLASS statement, indicator variables are generated for the levels assumed by the CLASS variable. The parameterization of the CLASS variable is the same as in the GLM procedure (Chapter 39, "The GLM Procedure.") If the CLASS statement is used, it must appear before the MODEL statement.

EFFECT Statement (Experimental)

EFFECT *name = spline(variables < / options >)* **;**

The experimental EFFECT statement names a constructed effect that you can use to specify terms in the MODEL statement. Each constructed effect corresponds to a collection of columns that are referred to using the name you supply. You can specify multiple EFFECT statements, and all EFFECT statements must precede the MODEL statement.

After the keyword EFFECT, the name of the effect is specified. This name must appear in only one EFFECT statement and must not be the name of a variable in the input data set. After an equal sign, the *effect-type* is specified followed by the list of variables used in defining the effect within parentheses. In SAS 9.2, the QUANTREG procedure supports only spline effects. You can also specify options pertaining to the effect definition after a slash following the variable list. The SPLIT option is always turned on for the QUANTREG procedure. For more details about the syntax with spline effects, see the section "Constructed Effects and the EFFECT Statement (Experimental)" on page 377 of Chapter 18, "Shared Concepts and Topics."

The QUANTREG procedure supports the regression spline effect. A spline effect expands variables into spline bases whose form depends on the options that you specify. Design matrix columns are generated separately for each of the numeric variables specified, and these columns are collectively referred to by the name that you specify. By default the spline basis generated for each variable is a cubic B-spline basis with three equally spaced knots positioned between the minimum and maximum values of that variable. You can find more details about regression splines and spline bases in the section "Constructed Effects and the EFFECT Statement (Experimental)" on page 377 of Chapter 18, "Shared Concepts and Topics."

ID Statement

ID *variables* **;**

When the diagnostics table is requested with the DIAGNOSTICS option in the MODEL statement, the variables listed in the ID statement are displayed in addition to the observation number. These values are useful for identifying observations. If the ID statement is omitted, only the observation number is displayed.

MODEL Statement

< label: > MODEL *response = < effects > < / options >* **;**

Main effects and interaction terms can be specified in the MODEL statement, as in the GLM procedure (Chapter 39, "The GLM Procedure.") Classification variables in the MODEL statement must be specified in the CLASS statement.

The optional *label*, which must be a valid SAS name, is used to label output from the matching MODEL statement.

Options

You can specify the following options for the model fit.

CORRB
produces the estimated correlation matrix of the parameter estimates. When the resampling method is used to compute the confidence intervals, the QUANTREG procedure computes the bootstrap correlation. When the sparsity method is used to compute the confidence intervals, the procedure computes the asymptotic correlation based on an estimator of the sparsity function. The rank method for confidence intervals does not provide a correlation estimate.

COVB
produces the estimated covariance matrix of the parameter estimates. When the resampling method is used to compute the confidence intervals, the QUANTREG procedure computes the bootstrap covariance. When the sparsity method is used to compute the confidence intervals, the procedure computes the asymptotic covariance based on an estimator of the sparsity function. The rank method for confidence intervals does not provide a covariance estimate.

CUTOFF=*value*
specifies the multiplier of the cutoff value for outlier detection. The default *value* is 3.

DIAGNOSTICS<(ALL)>
requests the outlier diagnostics. By default, only observations identified as outliers or leverage points are displayed. To request that all observations be displayed, specify the ALL option.

ITPRINT
displays the iteration history of the interior point algorithm or the smoothing algorithm.

LEVERAGE<(CUTOFF=*value* | **CUTOFFALPHA=***value* | **H=***n***)>**
requests an analysis of leverage points for the continuous covariates. The results are added to the diagnostics table, which you can request with the DIAGNOSTICS option in the MODEL statement. You can specify the cutoff value for leverage-point detection with the CUTOFF= option. The default cutoff value is $\sqrt{\chi^2_{p;1-\alpha}}$, where α can be specified with the CUTOFFALPHA= option. By default, $\alpha = 0.025$. You can use the H= option to specify the number of points to be minimized for the MCD algorithm used for the leverage-point analysis. By default, H=$[(3n + p + 1)/4]$, where n is the number of observations and p is the number of independent variables. The LEVERAGE option is ignored if the model includes classification variables as covariates.

NODIAG
suppresses the computation for outlier diagnostics. If you specify the NODIAG option, the diagnostics summary table will not be provided.

NOINT
specifies no intercept regression.

NOSUMMARY

suppresses the computation for summary statistics. If you specify the NOSUMMARY option, the summary statistics table will not be provided.

PLOT=*plot-option*

PLOTS=(*plot-options***)**

You can use the PLOTS= option in the MODEL statement together with the ODS GRAPHICS statement to request the quantile process plot in addition to all plots available with the PLOT= option in the PROC statement. The plot options in the PROC statement overwrite the plot options in the MODEL statement if you specify the same options in both statements.

You can specify the following plot option only in the MODEL statement:

QUANTPLOT<(EFFECTS) </ <NOLIMITS> <UNPACK> <OLS> > >

plots the regression quantile process. The estimated coefficient of each specified covariate effect is plotted as a function of the quantile. If you do not specify a covariate effect, quantile processes are plotted for all covariate effects in the MODEL statement. You can use the NOLIMITS option to suppress confidence bands for the quantile processes. By default, confidence bands are plotted, and process plots are displayed in panels, each of which can hold up to four plots. You can use the UNPACK option to create individual process plots. For an individual process plot, you can superimpose the ordinary least squares estimate and its confidence limits by specifying the OLS option. The confidence level of the ordinary least squares estimate is specified with the ALPHA= option in the PROC statement.

Again, to request these plots you must specify the ODS GRAPHICS statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

QUANTILE=*number-list* | **PROCESS**

specifies the quantiles for the quantile regression. You can specify any number of quantiles in $(0, 1)$. You can also compute the entire quantile process by specifying the PROCESS option. Only the simplex algorithm is available for computing the quantile process. Statistical inference is available only when a single quantile is specified.

If you do not specify the QUANTILE= option, the QUANTREG procedure fits a median regression, which corresponds to QUANTILE=0.5.

SCALE=*number*

specifies the scale value used to compute the standardized residuals. By default, the scale is computed as the corrected median of absolute residuals. See the section "Leverage Point and Outlier Detection" on page 5397 for details.

SEED=*number*

specifies the seed for the random number generator used to compute the MCMB confidence intervals. This seed is also used to randomly select the subgroups for preprocessing when you specify the PP option in the PROC statement. If you do not specify a seed, or if you specify a value less than or equal to zero, the seed is generated from reading the time of day from the computer clock.

By default or if you specify zero, the QUANTREG procedure generates a seed between one and one billion.

SINGULAR=*value*
sets the tolerance for testing singularity of the information matrix and the crossproducts matrix for the initial least squares estimates. Roughly, the test requires that a pivot be at least this value times the original diagonal value. By default, SINGULAR=1E−12.

OUTPUT Statement

OUTPUT < *OUT=SAS-data-set* > *keyword=name* < *. . . keyword=name* > < / *COLUMNWISE* > **;**

The OUTPUT statement creates a SAS data set containing statistics calculated after fitting models for all specified quantiles with the QUANTILE= option in the MODEL statement. At least one specification of the form *keyword=name* is required.

All variables in the original data set are included in the new data set, along with the variables created as options to the OUTPUT statement. These new variables contain fitted values and estimated quantiles. If you want to create a permanent SAS data set, you must specify a two-level name (refer to *SAS Language Reference: Concepts* for more information about permanent SAS data sets).

If you specify multiple quantiles in the MODEL statement, the COLUMNWISE option arranges the created OUTPUT data set in column-wise form. This arrangement repeats the input data for each quantile. By default, the OUTPUT data set is created in row-wise form. For each appropriate keyword specified in the OUTPUT statement, one variable for each specified quantile is generated. These variables appear in the sorted order of the specified quantiles.

The following specifications can appear in the OUTPUT statement:

OUT=*SAS-data-set* specifies the new data set. By default, the procedure uses the DATA*n* convention to name the new data set.

keyword=name specifies the statistics to include in the output data set and gives names to the new variables. Specify a keyword for each desired statistic (see the following list of keywords), an equal sign, and the variable to contain the statistic. The diagnostic statistics LEVERAGE, MAHADIST, OUTLIER, and ROBDIST can be requested when only a single quantile is specified in the MODEL statement.

The keywords allowed and the statistics they represent are as follows:

LEVERAGE specifies a variable to indicate leverage points. To include this variable in the OUTPUT data set, you must specify the LEVERAGE option in the MODEL statement. See the section "Leverage Point and Outlier Detection" on page 5397 for how to define LEVERAGE.

MAHADIST | MD specifies a variable to contain the Mahalanobis distance. To include this variable in the OUTPUT data set, you must specify the LEVERAGE option in the MODEL statement.

OUTLIER specifies a variable to indicate outliers. See the section "Leverage Point and Outlier Detection" on page 5397 for how to define OUTLIER.

PREDICTED | P specifies a variable to contain the estimated response.

QUANTILE | Q specifies a variable to contain the quantile for which the quantile regression is fitted. If you specify the COLUMNWISE option, this variable is created by default. If multiple quantiles are specified in the MODEL statement and the COLUMNWISE option is not specified, this variable is not created.

RESIDUAL | RES specifies a variable to contain the residuals (unstandardized)

$$y_i - \mathbf{x}_i^T \hat{\beta}$$

ROBDIST | RD specifies a variable to contain the robust MCD distance. To include this variable in the OUTPUT data set, you must specify the LEVERAGE option in the MODEL statement.

SPLINE | SP specifies a variable to contain the estimated spline effect, which includes all spline effects in the model and their interactions.

SRESIDUAL | SR specifies a variable to contain the standardized residuals

$$\frac{y_i - \mathbf{x}_i^T \hat{\beta}}{\hat{\sigma}}$$

See the section "Leverage Point and Outlier Detection" on page 5397 for how to compute σ.

STDP specifies a variable to contain the estimates of the standard errors of the estimated response.

PERFORMANCE Statement

The PERFORMANCE statement is used to change default options that affect the performance of PROC QUANTREG and to request tables that show the performance options in effect and timing details.

PERFORMANCE < *options* > **;**

The following options are available:

CPUCOUNT= *1-1024*

CPUCOUNT= *ACTUAL*

specifies the number of processors to use in the computation of the interior point algorithm. CPUCOUNT=ACTUAL sets CPUCOUNT to be the number of physical processors available. Note that this can be less than the physical number of CPUs if the SAS process has been restricted by system administration tools. Setting CPUCOUNT= to a number greater than the

actual number of available CPUs might result in reduced performance. This option overrides the SAS system option CPUCOUNT=. If CPUCOUNT=1, then NOTHREADS is in effect, and PROC QUANTREG uses singly threaded code.

DETAILS
requests the "PerfSettings" table that shows the performance settings in effect and the "Timing" table that provides a broad timing breakdown of the PROC QUANTREG step.

THREADS
enables multithreaded computation for the interior point algorithm. This option overrides the SAS system option THREADS | NOTHREADS. If you do not specify the ALGORITHM=INTERIOR option, then PROC QUANTREG ignores this option and uses singly threaded code.

NOTHREADS
disables multithreaded computation for the interior point algorithm. This option overrides the SAS system option THREADS | NOTHREADS.

TEST Statement

< label: > TEST *effects* < */WALD | LR* > **;**

The TEST statement provides a means of obtaining a test for the canonical linear hypothesis concerning the parameters of the tested effects:

$$\beta_j = 0, \quad j = i_1, \ldots, i_q$$

where q is the total number of parameters of the tested effects.

You can request Wald tests with the WALD option and likelihood ratio tests with the LR option. By default, only Wald tests are provided. See the section "Details: QUANTREG Procedure" on page 5384 for more information about these tests.

The optional *label*, which must be a valid SAS name, is used to label output from the corresponding TEST statement.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement specifies a weight variable in the input data set.

To request weighted quantile regression, place the weights in a variable and specify the name in the WEIGHT statement. The values of the WEIGHT variable can be nonintegral and are not truncated. Observations with nonpositive or missing values for the weight variable do not contribute to the fit of the model. See the section "Details: QUANTREG Procedure" on page 5384 for more information about weighted quantile regression.

Details: QUANTREG Procedure

Quantile Regression as an Optimization Problem

The model for linear quantile regression is

$$y = A'\beta + \epsilon$$

where $y = (y_1, \ldots, y_n)'$ is the $(n \times 1)$ vector of responses, $A' = (x_1, \ldots, x_n)'$ is the $(n \times p)$ regressor matrix, $\beta = (\beta_1, \ldots, \beta_p)'$ is the $(p \times 1)$ vector of unknown parameters, and $\epsilon = (\epsilon_1, \ldots, \epsilon_n)'$ is the $(n \times 1)$ vector of unknown errors.

L_1 regression, also known as median regression, is a natural extension of the sample median when the response is conditioned on the covariates. In L_1 regression, the least absolute residuals estimate $\hat{\beta}_{LAR}$, referred to as the L_1-*norm estimate*, is obtained as the solution of the minimization problem

$$\min_{\beta \in \mathbf{R}^p} \sum_{i=1}^{n} |y_i - x_i'\beta|$$

More generally, for quantile regression Koenker and Bassett (1978) defined the τth *regression quantile*, $0 < \tau < 1$, as any solution to the minimization problem

$$\min_{\beta \in \mathbf{R}^p} [\sum_{i \in \{i: y_i \geq x_i'\beta\}} \tau |y_i - x_i'\beta| + \sum_{i \in \{i: y_i < x_i'\beta\}} (1-\tau)|y_i - x_i'\beta|]$$

The solution is denoted as $\hat{\beta}(\tau)$, and the L_1-norm estimate corresponds to $\hat{\beta}(1/2)$. The τth regression quantile is an extension of the τth sample quantile $\hat{\xi}(\tau)$, which can be formulated as the solution of

$$\min_{\xi \in \mathbf{R}} \left[\sum_{i \in \{i: y_i \geq \xi\}} \tau |y_i - \xi| + \sum_{i \in \{i: y_i < \xi\}} (1-\tau)|y_i - \xi| \right]$$

If you specify weights $w_i, i = 1, \ldots, n$, with the WEIGHT statement, weighted quantile regression is carried out by solving

$$\min_{\beta_w \in \mathbf{R}^p} [\sum_{i \in \{i: y_i \geq x_i'\beta_w\}} w_i \tau |y_i - x_i'\beta_w| + \sum_{i \in \{i: y_i < x_i'\beta_w\}} w_i (1-\tau)|y_i - x_i'\beta_w|]$$

Weighted regression quantiles β_w can be used for L-estimation; refer to Koenker and Zhao (1994).

Optimization Algorithms

The optimization problem for median regression has been formulated and solved as a linear programming (LP) problem since the 1950s. Variations of the simplex algorithm, especially the method of Barrodale and Roberts (1973), have been widely used to solve this problem. The simplex algorithm is computationally demanding in large statistical applications, and in theory the number of iterations can increase exponentially with the sample size. This algorithm is often useful with data containing no more than tens of thousands of observations.

Several alternatives have been developed to handle L_1 regression for larger data sets. The interior point approach of Karmarkar (1984) solves a sequence of quadratic problems in which the relevant interior of the constraint set is approximated by an ellipsoid. The worst-case performance of the interior point algorithm has been proved to be better than that of the simplex algorithm. More important, experience has shown that the interior point algorithm is advantageous for larger problems.

Like L_1 regression, general quantile regression fits nicely into the standard primal-dual formulations of linear programming.

In addition to the interior point method, various heuristic approaches are available for computing L_1-type solutions. Among these, the finite smoothing algorithm of Madsen and Nielsen (1993) is the most useful. It approximates the L_1-type objective function with a smoothing function, so that the Newton-Raphson algorithm can be used iteratively to obtain a solution after a finite number of iterations. The smoothing algorithm extends naturally to general quantile regression.

The QUANTREG procedure implements the simplex, interior point, and smoothing algorithms. The remainder of this section describes these algorithms in more detail.

Simplex Algorithm

Let $\mu = [y - A'\beta]_+$, $\nu = [A'\beta - y]_+$, $\phi = [\beta]_+$, and $\varphi = [-\beta]_+$, where $[z]_+$ is the nonnegative part of z.

Let $D_{LAR}(\beta) = \sum_{i=1}^{n} |y_i - x_i'\beta|$. For the L_1 problem, the simplex approach solves $\min_\beta D_{LAR}(\beta)$ by reformulating it as the constrained minimization problem

$$\min_{\beta}\{e'\mu + e'\nu | y = A'\beta + \mu - \nu, \{\mu, \nu\} \in \mathbf{R}^n_+\}$$

where e denotes an $(n \times 1)$ vector of ones.

Let $B = [A' \; -A' \; I \; -I]$, $\theta = (\phi' \; \varphi' \; \mu' \; \nu')'$, and $d = (\mathbf{0}' \; \mathbf{0}' \; e' \; e')'$, where $\mathbf{0}' = (0\,0 \; \ldots \; 0)_p$. The reformulation presents a standard LP problem:

$$\begin{array}{ll} (P) & \min_{\theta} d'\theta \\ \text{subject to} & B\theta = y \\ & \theta \geq 0 \end{array}$$

This problem has the dual formulation

$$\begin{aligned} (D) \qquad & \max_z y'z \\ \text{subject to} \quad & B'z \leq d \end{aligned}$$

which can be simplified as

$$\max_z y'z; \;\; \text{subject to } Az = 0, z \in [-1, 1]^n$$

By setting $\eta = \frac{1}{2}z + \frac{1}{2}e, b = \frac{1}{2}Ae$, the problem becomes

$$\max_\eta y'\eta; \;\; \text{subject to } A\eta = b, \eta \in [0, 1]^n$$

For quantile regression, the minimization problem is $\min_\beta \sum \rho_\tau(y_i - x_i'\beta)$, and a similar set of steps leads to the dual formulation

$$\max_z y'z; \;\; \text{subject to } Az = (1-\tau)Ae, z \in [0, 1]^n$$

The QUANTREG procedure solves this LP problem by using the simplex algorithm of Barrodale and Roberts (1973). This algorithm solves the primary LP problem (P) by two stages, which exploit the special structure of the coefficient matrix B. The first stage picks the columns in A' or $-A'$ as pivotal columns. The second stage interchanges the columns in I or $-I$ as basis or nonbasis columns, respectively. The algorithm obtains an optimal solution by executing these two stages interactively. Moreover, because of the special structure of B, only the main data matrix A is stored in the current memory.

Although this special version of the simplex algorithm was introduced for median regression, it extends naturally to quantile regression for any given quantile and even to the entire quantile process (Koenker and d'Orey 1994). It greatly reduces the computing time required by the general simplex algorithm, and it is suitable for data sets with fewer than 5,000 observations and 50 variables.

Interior Point Algorithm

There are many variations of interior point algorithms. The QUANTREG procedure uses the primal-dual predictor-corrector algorithm implemented by Lustig, Marsden, and Shanno (1992). The text by Roos, Terlaky, and Vial (1997) provides more information about this particular algorithm. The following brief introduction of this algorithm uses the notation in the first reference.

To be consistent with the conventional LP setting, let $c = -y$, $b = (1-\tau)Ae$, and let u be the general upper bound. The linear program to be solved is

$$\begin{aligned} & min\{c'z\} \\ \text{subject to} \quad & Az = b \\ & 0 \leq z \leq u \end{aligned}$$

To simplify the computation, this is treated as the *primal* problem. The problem has n variables. The index i denotes a variable number, and k denotes an iteration number. If k is used as a subscript or superscript, it denotes "of iteration k."

Let v be the primal slack so that $z + v = u$. Associate dual variables w with these constraints. The interior point algorithm solves the system of equations to satisfy the Karush-Kuhn-Tucker (KKT) conditions for optimality:

$$Az = b$$
$$z + v = u$$
$$A't + s - w = c$$
$$ZSe = 0$$
$$VWe = 0$$
$$z, s, v, w \geq 0$$

where $W = diag(w)$ (that is, $W_{i,j} = w_i$ if $i = j$, $W_{i,j} = 0$ otherwise)

$V = diag(v), Z = diag(z), S = diag(s)$

These are the conditions for feasibility, with the addition of *complementarity* conditions $ZSe = 0$ and $VWe = 0$. $c'z = b't - u'w$ must occur at the optimum. Complementarity forces the optimal objectives of the primal and dual to be equal, $c'z_{opt} = b't_{opt} - u'w_{opt}$, as

$$0 = v'_{opt}w_{opt} = (u - z_{opt})'w_{opt} = u'w_{opt} - z'_{opt}w_{opt}$$

$$0 = z'_{opt}s_{opt} = s'_{opt}z_{opt} = (c - A't_{opt} + w_{opt})'z_{opt} =$$
$$c'z_{opt} - t'_{opt}(Az_{opt}) + w'_{opt}z_{opt} = c'z_{opt} - b't_{opt} + u'w_{opt}$$

Therefore

$$0 = c'z_{opt} - b't_{opt} + u'w_{opt}$$

The *duality gap*, $c'z - b't + u'w$, is used to measure the convergence of the algorithm. You can specify a tolerance for this convergence criterion with the TOLERANCE= option in the PROC statement.

Before the optimum is reached, it is possible for a solution (z, t, s, v, w) to violate the KKT conditions in one of several ways:

- Primal bound constraints can be broken, $\delta_b = u - z - v \neq 0$.
- Primal constraints can be broken, $\delta_c = b - Az \neq 0$.
- Dual constraints can be broken, $\delta_d = c - A't - s + w \neq 0$.
- Complementarity conditions are unsatisfied, $z's \neq 0$ and $v'w \neq 0$.

The interior point algorithm works by using Newton's method to find a direction $(\Delta z^k, \Delta t^k, \Delta s^k, \Delta v^k, \Delta w^k)$ to move from the current solution $(z^k, t^k, s^k, v^k, w^k)$ toward a better solution:

$$(z^{k+1}, t^{k+1}, s^{k+1}, v^{k+1}, w^{k+1}) = (z^k, t^k, s^k, v^k, w^k) + \kappa(\Delta z^k, \Delta t^k, \Delta s^k, \Delta v^k, \Delta w^k)$$

κ is the *step length* and is assigned a value as large as possible, but not so large that a z_i^{k+1} or s_i^{k+1} is "too close" to zero. You can control the step length with the KAPPA= option in the PROC statement.

The QUANTREG procedure implements a predictor-corrector variant of the primal-dual interior point algorithm. First, Newton's method is used to find a direction $(\Delta z_{aff}^k, \Delta t_{aff}^k, \Delta s_{aff}^k, \Delta v_{aff}^k, \Delta w_{aff}^k)$ in which to move. This is known as the *affine* step.

In iteration k, the *affine* step system that must be solved is

$$
\begin{aligned}
&\Delta z_{aff} + \Delta v_{aff} = \delta_b \\
&A\Delta z_{aff} = \delta_c \\
&A'\Delta t_{aff} + \Delta s_{aff} - \Delta w_{aff} = \delta_d \\
&S\Delta z_{aff} + Z\Delta s_{aff} = -ZSe \\
&V\Delta w_{aff} + W\Delta z_{aff} = -VWe
\end{aligned}
$$

Therefore, the computations involved in solving the affine step are

$$
\begin{aligned}
&\Theta = SZ^{-1} + WV^{-1} \\
&\rho = \Theta^{-1}(\delta_d + (S - W)e - V^{-1}W\delta_b) \\
&\Delta t_{aff} = (A\Theta^{-1}A')^{-1}(\delta_c + A\rho) \\
&\Delta z_{aff} = \Theta^{-1}A'\Delta t_{aff} - \rho \\
&\Delta v_{aff} = \delta_b - \Delta z_{aff} \\
&\Delta w_{aff} = -We - V^{-1}W\Delta z_{aff} \\
&\Delta s_{aff} = -Se - Z^{-1}S\Delta z_{aff}
\end{aligned}
$$

$$
\begin{aligned}
&(z_{aff}, t_{aff}, s_{aff}, v_{aff}, w_{aff}) = (z, t, s, v, w) + \\
&\kappa(\Delta z_{aff}, \Delta t_{aff}, \Delta s_{aff}, \Delta v_{aff}, \Delta w_{aff})
\end{aligned}
$$

κ is the *step length* as before.

The success of the affine step is gauged by calculating the complementarity of $z's$ and $v'w$ at $(z_{aff}^k, t_{aff}^k, s_{aff}^k, v_{aff}^k, w_{aff}^k)$ and comparing it with the complementarity at the starting point $(z^k, t^k, s^k, v^k, w^k)$. If the affine step was successful in reducing the complementarity by a substantial amount, the need for centering is not great, and a value close to zero is assigned to σ in a second linear system (see following), which is used to determine a centering vector. If, however, the affine step was unsuccessful, then centering is deemed beneficial, and a value close to 1.0 is assigned to σ. In other words, the value of σ is adaptively altered depending on progress made toward the optimum.

The following linear system is solved to determine a centering vector $(\Delta z_c, \Delta t_c, \Delta s_c, \Delta v_c, \Delta w_c)$ from $(z_{aff}, t_{aff}, s_{aff}, v_{aff}, w_{aff})$:

$$
\begin{aligned}
&\Delta z_c + \Delta v_c = 0 \\
&A\Delta z_c = 0
\end{aligned}
$$

$$A'\Delta t_c + \Delta s_c - \Delta w_c = 0$$
$$S\Delta z_c + Z\Delta s_c = -Z_{aff}S_{aff}e + \sigma\mu e$$
$$V\Delta w_c + W\Delta v_c = -V_{aff}W_{aff}e + \sigma\mu e$$

where

$\zeta_{start} = z's + v'w$, complementarity at the start of the iteration
$\zeta_{aff} = z'_{aff}s_{aff} + v'_{aff}w_{aff}$, the affine complementarity
$\mu = \zeta_{aff}/2n$, the average complementarity
$\sigma = (\zeta_{aff}/\zeta_{start})^3$

Therefore, the computations involved in solving the centering step are

$$\rho = \Theta^{-1}(\sigma\mu(Z^{-1} - V^{-1})e - Z^{-1}Z_{aff}S_{aff}e + V^{-1}V_{aff}W_{aff}e)$$
$$\Delta t_c = (A\Theta^{-1}A')^{-1}A\rho$$
$$\Delta z_c = \Theta^{-1}A'\Delta t_c - \rho$$
$$\Delta v_c = -\Delta z_c$$
$$\Delta w_c = \sigma\mu V^{-1}e - V^{-1}V_{aff}W_{aff}e - V^{-1}W_{aff}\Delta v_c$$
$$\Delta s_c = \sigma\mu Z^{-1}e - Z^{-1}Z_{aff}S_{aff}e - Z^{-1}S_{aff}\Delta z_c$$

Then

$$(\Delta z, \Delta t, \Delta s, \Delta v, \Delta w) =$$
$$(\Delta z_{aff}, \Delta t_{aff}, \Delta s_{aff}, \Delta v_{aff}, \Delta w_{aff}) +$$
$$(\Delta z_c, \Delta t_c, \Delta s_c, \Delta v_c, \Delta w_c)$$

$$(z^{k+1}, t^{k+1}, s^{k+1}, v^{k+1}, w^{k+1}) =$$
$$(z^k, t^k, s^k, v^k, w^k) +$$
$$\kappa(\Delta z, \Delta t, \Delta s, \Delta v, \Delta w)$$

where, as before, κ is the *step length* assigned a value as large as possible, but not so large that a z_i^{k+1}, s_i^{k+1}, v_i^{k+1}, or w_i^{k+1} is "too close" to zero.

Although the predictor-corrector variant entails solving two linear systems instead of one, fewer iterations are usually required to reach the optimum. The additional overhead of the second linear system is small because the matrix $(A\Theta^{-1}A')$ has already been factorized in order to solve the first linear system.

You can specify the starting point with the INEST= option in the PROC statement. By default, the starting point is set to be the least squares estimate.

Smoothing Algorithm

To minimize the sum of the absolute residuals $D_{LAR}(\beta)$, the smoothing algorithm approximates the nondifferentiable function D_{LAR} by the following smooth function, which is referred to as the Huber function:

$$D_\gamma(\beta) = \sum_{i=1}^{n} H_\gamma(r_i(\beta))$$

where

$$H_\gamma(t) = \begin{cases} t^2/(2\gamma) & \text{if } |t| \le \gamma \\ |t| - \gamma/2 & \text{if } |t| > \gamma \end{cases}$$

Here $r_i(\beta) = y_i - x_i'\beta$, and the *threshold* γ is a positive real number. The function D_γ is continuously differentiable and a minimizer β_γ of D_γ is close to a minimizer $\hat{\beta}_{LAR}$ of $D_{LAR}(\beta)$ when γ is close to zero.

The advantage of the smoothing algorithm as described in Madsen and Nielsen (1993) is that the L_1 solution $\hat{\beta}_{LAR}$ can be detected when $\gamma > 0$ is small. In other words, it is not necessary to let γ converge to zero in order to find a minimizer of $D_{LAR}(\beta)$. The algorithm terminates before going through the entire sequence of values of γ that are generated by the algorithm. Convergence is indicated by no change of the status of residuals $r_i(\beta)$ as γ goes through this sequence.

The smoothing algorithm extends naturally from L_1 regression to general quantile regression; refer to Chen (2007). The function

$$D_{\rho_\tau}(\beta) = \sum_{i=1}^{n} \rho_\tau(y_i - x_i'\beta)$$

can be approximated by the smooth function

$$D_{\gamma,\tau}(\beta) = \sum_{i=1}^{n} H_{\gamma,\tau}(r_i(\beta))$$

where

$$H_{\gamma,\tau}(t) = \begin{cases} t(\tau-1) - \frac{1}{2}(\tau-1)^2\gamma & \text{if } t \le (\tau-1)\gamma \\ \frac{t^2}{2\gamma} & \text{if } (\tau-1)\gamma \le t \le \tau\gamma \\ t\tau - \frac{1}{2}\tau^2\gamma & \text{if } t \ge \tau\gamma \end{cases}$$

The function $H_{\gamma,\tau}$ is determined by whether $r_i(\beta) \le (\tau-1)\gamma$, $r_i(\beta) \ge \tau\gamma$, or $(\tau-1)\gamma \le r_i(\beta) \le \tau\gamma$. These inequalities divide $\mathbf{R}^p$ into subregions separated by the parallel hyperplanes $r_i(\beta) = (\tau-1)\gamma$ and $r_i(\beta) = \tau\gamma$. The set of all such hyperplanes is denoted by $B_{\gamma,\tau}$:

$$B_{\gamma,\tau} = \{\beta \in \mathbf{R}^p | \exists i : r_i(\beta) = (\tau-1)\gamma \text{ or } r_i(\beta) = \tau\gamma\}$$

Define the sign vector $s_\gamma(\beta) = (s_1(\beta), \ldots, s_n(\beta))'$ as

$$s_i = s_i(\beta) = \begin{cases} -1 & \text{if } r_i(\beta) \le (\tau-1)\gamma \\ 0 & \text{if } (\tau-1)\gamma \le r_i(\beta) \le \tau\gamma \\ 1 & \text{if } r_i(\beta) \ge \tau\gamma \end{cases}$$

and introduce

$$w_i = w_i(\beta) = 1 - s_i^2(\beta)$$

Therefore,

$$H_{\gamma,\tau}(r_i(\beta)) = \frac{1}{2\gamma} w_i r_i^2(\beta)$$
$$+ s_i[\frac{1}{2} r_i(\beta) + \frac{1}{4}(1-2\tau)\gamma + s_i(r_i(\beta)(\tau - \frac{1}{2}) - \frac{1}{4}(1 - 2\tau + 2\tau^2)\gamma)]$$

yielding

$$D_{\gamma,\tau}(\beta) = \frac{1}{2\gamma} r' W_{\gamma,\tau} r + v'(s) r + c(s)$$

where $W_{\gamma,\tau}$ is the diagonal $n \times n$ matrix with diagonal elements $w_i(\beta)$, $v'(s) = (s_1((2\tau-1)s_1+1)/2, \ldots, s_n((2\tau - 1)s_n + 1)/2)$, $c(s) = \sum[\frac{1}{4}(1 - 2\tau)\gamma s_i - \frac{1}{4} s_i^2(1 - 2\tau + 2\tau^2)\gamma]$, and $r(\beta) = (r_1(\beta), \ldots, r_n(\beta))'$.

The gradient of $D_{\gamma,\tau}$ is given by

$$D^{(1)}_{\gamma,\tau}(\beta) = -A[\frac{1}{\gamma} W_{\gamma,\tau}(\beta) r(\beta) + v(s)]$$

and for $\beta \in \mathbf{R}^p \backslash B_{\gamma,\tau}$ the Hessian exists and is given by

$$D^{(2)}_{\gamma,\tau}(\beta) = \frac{1}{\gamma} A W_{\gamma,\tau}(\beta) A'$$

The gradient is a continuous function in $\mathbf{R}^p$, whereas the Hessian is piecewise constant.

Following Madsen and Nielsen (1993), the vector s is referred to as a *γ-feasible sign vector* if there exists $\beta \in \mathbf{R}^p \backslash B_{\gamma,\tau}$ with $s_\gamma(\beta) = s$. If s is *γ-feasible*, then Q_s is defined as the quadratic function $Q_s(\alpha)$ that is derived from $D_{\gamma,\tau}(\beta)$ by substituting s for s_γ. Thus, for any β with $s_\gamma = s$,

$$Q_s(\alpha) = \frac{1}{2}(\alpha - \beta)' D^{(2)}_{\gamma,\tau}(\beta)(\alpha - \beta) + D^{(1)}_{\gamma,\tau}(\beta)(\alpha - \beta) + D_{\gamma,\tau}(\beta)$$

In the domain $C_s = \{\alpha | s_\gamma(\alpha) = s\}$

$$D_{\gamma,\tau}(\alpha) = Q_s(\alpha)$$

For each $\gamma > 0$ and $\theta \in \mathbf{R}^p$, there can be one or several corresponding quadratics Q_s. If $\theta \notin B_{\gamma,\tau}$ then Q_s is characterized by θ and γ, but for $\theta \in B_{\gamma,\tau}$ the quadratic is not unique. Therefore, a *reference*

$$(\gamma, \theta, s)$$

determines the quadratic.

Again following Madsen and Nielsen (1993), let

(γ, θ, s) be a *feasible reference* if s is a γ-*feasible sign vector* with $\theta \in C_s$, and

(γ, θ, s) be a *solution reference* if it is feasible and θ minimizes $D_{\gamma,\tau}$.

The smoothing algorithm for minimizing D_{ρ_τ} is based on minimizing $D_{\gamma,\tau}$ for a set of decreasing γ. For each new value of γ, information from the previous solution is used. Finally, when γ is small enough, a solution can be found by the modified Newton-Raphson algorithm as stated by Madsen and Nielsen (1993):

find an initial solution reference $(\gamma, \beta_\gamma, s)$

repeat

decrease γ

find a solution reference $(\gamma, \beta_\gamma, s)$

until $\gamma = 0$

β_0 is the solution.

By default, the initial solution reference is found by letting β_γ be the least squares solution. Alternatively, you can specify the initial solution reference with the INEST= option in the PROC statement. Then γ and s are chosen according to these initial values.

There are several approaches for determining a decreasing sequence of values of γ. The QUANTREG procedure uses a strategy by Madsen and Nielsen (1993). The computation involved is not significant comparing with the Newton-Raphson step. You can control the ratio of consecutive decreasing values of γ with the RRATIO= suboption of the ALGORITHM= option in the PROC statement. By default,

$$\text{RRATIO} = \begin{cases} 0.1 & \text{if } n \geq 10000 \text{ and } p \leq 20 \\ 0.9 & \text{if } \frac{p}{n} \geq 0.1 \text{ or } \{n \leq 5000 \text{ and } p \geq 300\} \\ 0.5 & \text{otherwise} \end{cases}$$

For the L_1 and quantile regression, it turns out that the smoothing algorithm is very efficient and competitive, especially for a *fat* data set—namely, when $\frac{p}{n} > 0.05$ and AA' is dense. Refer to Chen (2007) for a complete smoothing algorithm and details.

Confidence Interval

The QUANTREG procedure provides three methods to compute confidence intervals for the regression quantile parameter $\beta(\tau)$: sparsity, rank, and resampling. The sparsity method is the most direct and the fastest, but it involves estimation of the sparsity function, which is not robust for data that are not independently and identically distributed. To deal with this problem, the QUANTREG procedure computes a Huber sandwich estimate by using a local estimate of the sparsity function. The rank method, which computes confidence intervals by inverting the rank score test, does not suffer

from this problem, but it uses the simplex algorithm and is computationally expensive with large data sets. The resampling method, which uses the bootstrap, can overcome all of these problems, but it is unstable for small data sets.

Based on these properties, the QUANTREG uses a combination of the resampling and rank methods as the default. For data sets with more than either 5,000 observations or 20 variables, the QUANTREG procedure uses the MCMB resampling method; otherwise it uses the rank method. You can request a particular method by using the CI= option in the PROC statement.

Sparsity

Consider the linear model

$$y_i = x_i'\beta + \epsilon_i$$

and assume that $\{\epsilon_i\}$, $i = 1, \ldots, n$, are i.i.d. with a distribution F and a density $f = F'$, where $f(F^{-1}(\tau)) > 0$ in a neighborhood of τ. Under some mild conditions

$$\sqrt{n}(\hat{\beta}(\tau) - \beta(\tau)) \rightarrow N(0, \omega^2(\tau, F)\Omega^{-1})$$

where $\omega^2(\tau, F) = \tau(1-\tau)/f^2(F^{-1}(\tau))$ and $\Omega = \lim_{n\rightarrow\infty} n^{-1}\sum x_i x_i'$. Refer to Koenker and Bassett (1982).

This asymptotic distribution for the regression quantile $\hat{\beta}(\tau)$ can be used to construct confidence intervals. However, the reciprocal of the density function

$$s(\tau) = [f(F^{-1}(\tau))]^{-1}$$

which is called the *sparsity function*, must first be estimated.

Since

$$s(t) = \frac{d}{dt}F^{-1}(t)$$

$s(t)$ can be estimated by the difference quotient of the empirical quantile function—that is,

$$\hat{s}_n(t) = [\hat{F}_n^{-1}(t + h_n) - \hat{F}_n^{-1}(t - h_n)]/2h_n$$

where $\hat{F}_n$ is an estimate of F^{-1} and h_n is a bandwidth that tends to zero as $n \rightarrow \infty$.

The QUANTREG procedure provides two bandwidth methods. The Bofinger bandwidth

$$h_n = n^{-1/5}\left(\frac{4.5s^2(t)}{(s^{(2)}(t))^2}\right)^{1/5}$$

is an optimizer of mean squared error for standard density estimation, and the Hall-Sheather bandwidth

$$h_n = n^{-1/3}z_\alpha^{2/3}\left(\frac{1.5s(t)}{s^{(2)}(t)}\right)^{1/3}$$

is based on Edgeworth expansions for studentized quantiles, where $s^{(2)}(t)$ is the second derivative of $s(t)$ and z_α satisfies $\Phi(z_\alpha) = 1 - \alpha/2$ for the construction of $1 - \alpha$ confidence intervals. The quantity

$$\frac{s(t)}{s^{(2)}(t)} = \frac{f^2}{2(f^{(1)}/f)^2 + [(f^{(1)}/f)^2 - f^{(2)}/f]}$$

is not sensitive to f and can be estimated by assuming f is Gaussian.

$\hat{F}^{-1}$ can be estimated by the empirical quantile function of the residuals from the quantile regression fit,

$$\hat{F}^{-1}(t) = r_{(i)}, \quad \text{for } t \in [(i-1)/n, i/n),$$

or the empirical quantile function of regression proposed by Bassett and Koenker (1982),

$$\hat{F}^{-1}(t) = \overline{x}'\hat{\beta}(t)$$

The QUANTREG procedure interpolates the first empirical quantile function and gets the piecewise linear version

$$\hat{F}^{-1}(t) = \begin{cases} r_{(1)} & \text{if } t \in [0, 1/2n) \\ \lambda r_{(i+1)} + (1-\lambda) r_{(i)} & \text{if } t \in [(2j-1)/2n, (2i+1)/2n) \\ r_{(n)} & \text{if } t \in [(2n-1), 1] \end{cases}$$

$\hat{F}^{-1}$ is set to a constant if $t \pm h_n$ falls outside $[0, 1]$.

This estimator of the sparsity function is sensitive to the i.i.d. assumption. Alternately, Koenker and Machado (1999) considered the non-i.i.d. case. By assuming local linearity of the conditional quantile function $Q(\tau|x)$ in x, they proposed a local estimator of the density function by using the difference quotient. A Huber sandwich estimate of the covariance and standard error is computed and used to construct the confidence intervals. One difficulty with this method is the selection of the bandwidth when using the difference quotient. With a small sample size, either the Bofinger or the Hall-Sheather bandwidth tends to be too large to assure local linearity of the conditional quantile function. The QUANTREG procedure uses a heuristic bandwidth selection in these cases.

By default, the QUANTREG procedure computes non-i.i.d. confidence intervals. You can request i.i.d. confidence intervals with the IID option in the PROC statement.

Inversion of Rank Tests

The classical theory of rank tests can be extended to test the hypothesis H_0: $\beta_2 = \eta$ in the linear regression model $y = X_1\beta_1 + X_2\beta_2 + \epsilon$. Here $(X_1, X_2) = A'$. See Gutenbrunner and Jureckova (1992) for more details. By inverting this test, confidence intervals can be computed for the regression quantiles corresponding to β_2.

The rank score function $\hat{a}_n(t) = (\hat{a}_{n1}(t), \ldots, \hat{a}_{nn}(t))$ can be obtained by solving the dual problem

$$\max_a \{(y - X_2\eta)'a | X_1'a = (1-t)X_1'e,\ a \in [0,1]^n\}$$

For a fixed quantile τ, integrating $\hat{a}_{ni}(t)$ with respect to the τ-quantile score function

$$\varphi_\tau(t) = \tau - I(t < \tau)$$

yields the τ-quantile scores

$$\hat{b}_{ni} = -\int_0^1 \varphi_\tau(t) d\hat{a}_{ni}(t) = \hat{a}_{ni}(\tau) - (1-\tau)$$

Under the null hypothesis H_0: $\beta_2 = \eta$

$$S_n(\eta) = n^{-1/2} X_2' \hat{b}_n(\eta) \to N(0, \tau(1-\tau)\Omega_n)$$

for large n, where $\Omega_n = n^{-1} X_2'(I - X_1(X_1'X_1)^{-1}X_1')X_2$.

Let

$$T_n(\eta) = \frac{1}{\sqrt{\tau(1-\tau)}} S_n(\eta)\Omega_n^{-1/2}$$

then $T_n(\hat{\beta}_2(\tau)) = 0$ from the constraint $A\hat{a} = (1-\tau)Ae$ in the full model. In order to obtain confidence intervals for β_2, a critical value can be specified for T_n. The dual vector $\hat{a}_n(\eta)$ is a piecewise constant in η, and η can be altered without compromising the optimality of $\hat{a}_n(\eta)$ as long as the sign of the residuals in the primal quantile regression problem do not change. When η gets to such a boundary the solution does change, but can be restored by taking one simplex pivot. The process can continue in this way until $T_n(\eta)$ exceeds the specified critical value. Since $T_n(\eta)$ is piecewise constant, interpolation can be used to obtain the desired level of confidence interval; see Koenker and d'Orey (1994).

Resampling

The bootstrap can be implemented to compute confidence intervals for regression quantile estimates. As in other regression applications, both the residual bootstrap and the xy-pair bootstrap can be used. The former assumes i.i.d. random errors and resamples from the residuals, while the later resamples xy pairs and accommodates some forms of heteroscedasticity. Koenker (1994) considered a more interesting resampling mechanism, resampling directly from the full regression quantile process, which he called the Heqf bootstrap.

In contrast with these bootstrap methods, Parzen, Wei, and Ying (1994) observed that

$$S(\beta) = n^{-1/2} \sum_{i=1}^{n} x_i(\tau - I(y_i \le x_i'\beta))$$

which is the estimating equation for the τth regression quantile, is a pivotal quantity for the τth quantile regression parameter β_τ. In other words, the distribution of $S(\beta)$ can be generated exactly by a random vector U, which is a weighted sum of independent, re-centered Bernoulli variables. They further showed that for large n, the distribution of $\hat{\beta}(\tau) - \beta_\tau$ can be approximated by the conditional distribution of $\hat{\beta}_U - \hat{\beta}_n(\tau)$, where $\hat{\beta}_U$ solves an augmented quantile regression problem with $n+1$ observations with $x_{n+1} = -n^{-1/2}u/\tau$ and y_{n+1} sufficiently large for a given realization

of u. By exploiting the asymptotically pivot role of the quantile regression "gradient condition," this approach also achieves some robustness to certain heteroscedasticity.

Although the bootstrap method by Parzen, Wei, and Ying (1994) is much simpler, it is too time-consuming for relatively large data sets, especially for high-dimensional data sets. The QUANTREG procedure implements a new, general resampling method developed by He and Hu (2002), which is referred to as the Markov chain marginal bootstrap (MCMB). For quantile regression, the MCMB method has the advantage that it solves p one-dimensional equations instead of p-dimensional equations, as do the previous bootstrap methods. This greatly improves the feasibility of the resampling method in computing confidence intervals for regression quantiles. Since resampling methods achieve stability only for relatively large data sets, they are not recommended for small data sets ($n < 5000$ and $p < 20$).

Covariance-Correlation

You can specify the COVB and CORRB options in the MODEL statement to request covariance and correlation matrices for the estimated parameters. The QUANTREG procedure computes these matrices only for a single quantile at a time.

The QUANTREG procedure provides two methods for computing the covariance and correlation matrices of the estimated parameters: an asymptotic method and a bootstrap method. Bootstrap covariance and correlation matrices are computed when resampling confidence intervals are computed. Asymptotic covariance and correlation matrices are computed when asymptotic confidence intervals are computed. The rank method for confidence intervals does not provide a covariance-correlation estimate.

Asymptotic Covariance-Correlation

This method corresponds to the sparsity method for the confidence intervals. For the sparsity function in the computation of the asymptotic covariance and correlation, the QUANTREG procedure provides both i.i.d. and non-i.i.d. estimates. By default, the QUANTREG procedure computes non-i.i.d. estimates.

Bootstrap Covariance-Correlation

This method corresponds to the resampling method for the confidence intervals. The Markov chain marginal bootstrap (MCMB) method is used.

Linear Test

Two tests are available in the QUANTREG procedure for the linear null hypothesis $H_0 : \beta_2 = 0$. Here β_2 denotes a subset of the parameters, where the parameter vector $\beta(\tau)$ is partitioned as

$\beta'(\tau) = (\beta_1'(\tau), \beta_2'(\tau))$, and the covariance matrix Ω for the parameter estimates is partitioned correspondingly as Ω_{ij} with $i = 1, 2$; $j = 1, 2$; and $\Omega^{22} = (\Omega_{22} - \Omega_{21}\Omega_{11}^{-1}\Omega_{12})^{-1}$.

The Wald test statistic, which is based on the estimated coefficients for the unrestricted model, is given by

$$T_W(\tau) = \hat{\beta}_2'(\tau)\hat{\Sigma}(\tau)^{-1}\hat{\beta}_2(\tau)$$

where $\hat{\Sigma}(\tau)$ is an estimator of the covariance of $\hat{\beta}_2(\tau)$. The QUANTREG procedure provides two estimators for the covariance, as described in the previous section. The estimator based on the asymptotic covariance is

$$\hat{\Sigma}(\tau) = \frac{1}{n}\hat{\omega}(\tau)^2\Omega^{22}$$

where $\hat{\omega}(\tau) = \sqrt{\tau(1-\tau)}\hat{s}(\tau)$ and $\hat{s}(\tau)$ is the estimated sparsity function. The estimator based on the bootstrap covariance is the empirical covariance of the MCMB samples.

The likelihood ratio test is based on the difference between the objective function values in the restricted and unrestricted models. Let $D_0(\tau) = \sum \rho_\tau(y_i - x_i\hat{\beta}(\tau))$ and $D_1(\tau) = \sum \rho_\tau(y_i - x_{1i}\hat{\beta}_1(\tau))$, and set

$$T_{LR}(\tau) = 2(\tau(1-\tau)\hat{s}(\tau))^{-1}(D_1(\tau) - D_0(\tau))$$

where $\hat{s}(\tau)$ is the estimated sparsity function.

Koenker and Machado (1999) prove that these two tests are asymptotically equivalent and that the distributions of the test statistics converge to χ_q^2 under the null hypothesis, where q is the dimension of β_2.

Leverage Point and Outlier Detection

The QUANTREG procedure uses robust multivariate location and scale estimates for leverage-point detection.

Mahalanobis distance is defined as

$$MD(x_i) = [(x_i - \bar{x})'\bar{C}(A)^{-1}(x_i - \bar{x})]^{1/2}$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^n x_i$ and $\bar{C}(A) = \frac{1}{n-1}\sum_{i=1}^n (x_i - \bar{x})'(x_i - \bar{x})$ are the empirical multivariate location and scale. Here, $x_i = (x_{i1}, \ldots, x_{i(p-1)})'$ does not include the intercept variable. The relationship between the Mahalanobis distance $MD(x_i)$ and the matrix $H = (h_{ij}) = A'(AA')^{-1}A$ is

$$h_{ii} = \frac{1}{n-1}MD_i^2 + \frac{1}{n}$$

Robust distance is defined as

$$RD(x_i) = [(x_i - T(A))' C(A)^{-1} (x_i - T(A))]^{1/2}$$

where $T(A)$ and $C(A)$ are robust multivariate location and scale estimates computed with the minimum covariance determinant (MCD) method of Rousseeuw and Van Driessen (1999).

These distances are used to detect leverage points. You can use the DIAGNOSTICS and LEVERAGE options in the MODEL statement to request leverage-point and outlier diagnostics. Two new variables, LEVERAGE and OUTLIER, are created and saved in an output data set specified in the OUTPUT statement.

Let $C(p) = \sqrt{\chi^2_{p;1-\alpha}}$ be the cutoff value. The variable LEVERAGE is defined as

$$\text{LEVERAGE} = \begin{cases} 0 & \text{if } RD(x_i) \le C(p) \\ 1 & \text{otherwise} \end{cases}$$

You can specify a cutoff value with the LEVERAGE option in the MODEL statement.

Residuals $r_i, i = 1, \ldots, n$, based on quantile regression estimates are used to detect vertical outliers. The variable OUTLIER is defined as

$$\text{OUTLIER} = \begin{cases} 0 & \text{if } |r_i| \le k\sigma \\ 1 & \text{otherwise} \end{cases}$$

You can specify the multiplier k of the cutoff value with the CUTOFF= option in the MODEL statement. You can specify the scale σ with the SCALE= option in the MODEL statement. By default, $k = 3$ and the scale σ is computed as the corrected median of the absolute residuals $\sigma = \text{median}\{|r_i|/\beta_0, i = 1, \ldots, n\}$, where $\beta_0 = \Phi^{-1}(0.75)$ is an adjustment constant for consistency with the normal distribution.

An ODS table called DIAGNOSTICS contains these two variables.

INEST= Data Set

The INEST= data set specifies initial estimates for all the parameters in the model. The INEST= data set must contain the intercept variable (named Intercept) and all independent variables in the MODEL statement.

If BY processing is used, the INEST= data set should also include the BY variables, and there must be at least one observation for each BY group. If there is more than one observation in one BY group, the first one read is used for that BY group.

If the INEST= data set also contains the _TYPE_ variable, only observations with the _TYPE_ value 'PARMS' are used as starting values.

You can specify starting values for the interior point algorithm or the smoothing algorithm in the INEST= data set. The INEST= data set has the same structure as the OUTEST= data set but is not required to have all the variables or observations that appear in the OUTEST= data set. One simple

use of the INEST= option is passing the previous OUTEST= data set directly to the next model as an INEST= data set, assuming that the two models have the same parameterization. If you specify more than one quantile in the MODEL statement, the same initial values are used for all quantiles.

OUTEST= Data Set

The OUTEST= data set contains parameter estimates for the specified model with all quantiles. A set of observations is created for each quantile specified. You can also specify a label in the MODEL statement to distinguish between the estimates for different models used by the QUANTREG procedure.

Note that, if the QUANTREG procedure does not produce valid solutions, the parameter estimates are set to missing in the OUTEST data set.

If created, this data set contains all variables specified in the MODEL statement and the BY statement. Each observation consists of parameter values for a specified quantile with the dependent variable having the value -1.

The following variables are also added to the data set:

MODEL	a character variable of length 8 containing the label of the MODEL statement, if present. Otherwise, the variable's value is blank.
ALGORITHM	a character variable of length 8 containing the name of the algorithm used for computing the parameter estimates, either SIMPLEX, INTERIOR, or SMOOTH
TYPE	a character variable of length 8 containing the type of the observation. it is fixed as PARMS to indicate that the observation includes parameter estimates.
STATUS	a character variable of length 12 containing the status of model fitting, either NORMAL, NOUNIQUE, or NOVALID
INTERCEPT	a numeric variable containing the intercept parameter estimates
QUANTILE	a numeric variable containing the specified quantiles

Any BY variables specified are also added to the OUTEST= data set.

Computational Resources

The various algorithms need different amounts of memory for working space. Let p be the number of parameters estimated and n be the number of observations used in the model estimation.

For the simplex algorithm, the minimum working space (in bytes) needed is

$$2np + 6n + 10p$$

for the interior point algorithm,

$$np + p^2 + 13n + 4p$$

and for the smoothing algorithm,

$$np + p^2 + 6n + 4p$$

For the last two algorithms, if you want to use preprocessing, an extra amount

$$np + 6n + 2p$$

is needed.

If sufficient space is available, the input data set is kept in memory; otherwise, the input data set is reread as necessary, and the execution time of the procedure increases substantially.

ODS Table Names

The QUANTREG procedure assigns a name to each table it creates. You can specify these names when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table.

Table 72.5 ODS Tables Produced in PROC QUANTREG

ODS Table Name	Description	Statement	Option
ClassLevels	Classification variable levels	CLASS	default
CorrB	Parameter estimate correlation matrix	MODEL	CORRB
CovB	Parameter estimate covariance matrix	MODEL	COVB
Diagnostics	Outlier diagnostics	MODEL	DIAGNOSTICS
DiagSummary	Summary of the outlier diagnostics	MODEL	DIAGNOSTICS
IPIterHistory	Iteration history (Interior Point)	MODEL	ITPRINT
ModelInfo	Model information	MODEL	default
NObs	Number of observations	PROC	default
ObjFunction	Objective function	MODEL	default
ParameterEstimates	Parameter estimates	MODEL	default
ParmInfo	Parameter indices	MODEL	default
PerfSettings	Performance settings	PERFORMANCE	DETAILS
ProcessEst	Quantile process estimates	MODEL	QUANTILE=
ProcessObj	Objective function for quantile process	MODEL	QUANTILE=
SMIterHistory	Iteration history (Smoothing)	MODEL	ITPRINT
SummaryStatistics	Summary statistics for model variables	MODEL	default
Tests	Results for tests	TEST	default
ScalableTiming	Timing details	PERFORMANCE	DETAILS

ODS Graphics

The QUANTREG procedure uses ODS Graphics to produce graphical displays for model fitting and model diagnostics. To request these plots you must specify the ODS GRAPHICS statement in addition to specific plot options in the PROC statement or the MODEL statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

For a single quantile, two plots are particularly useful in revealing outliers and leverage points. The first is a scatter plot of the standardized residuals for the specified quantile against the robust distances. The second is a scatter plot of the robust distances against the classical Mahalanobis distances. You can request these two plots by using the PLOT=RDPLOT and PLOT=DDPLOT options.

You can also request a normal quantile-quantile plot and a histogram of the standardized residuals for the specified quantile with the PLOT=QQPLOT and PLOT=HISTOGRAM options, respectively.

You can request a plot of fitted conditional quantiles by the single continuous variable specified in the model with the PLOT=FITPLOT option.

All these plots can be requested by specifying corresponding plot options in either the PROC statement or the MODEL statement. If you specify same plot options in both statements, options in the PROC statement override options in the MODEL statement.

You can specify the PLOT=QUANTPLOT option in only the MODEL statement to request a quantile process plot with confidence bands.

The plot options in the PROC statement and the MODEL statement are summarized in Table 72.6. See the PLOT= option in the PROC statement and the PLOT= option in the MODEL statement for details.

Table 72.6 Options for Plots

Keyword	Plot
ALL	All appropriate plots
DDPLOT	Robust distance vs. Mahalanobis distance
FITPLOT	Conditional quantile fit vs. independent variable
HISTOGRAM	Histogram of standardized robust residuals
NONE	No plot
QUANTPLOT	Scatter plot of regression quantile
QQPLOT	Q-Q plot of standardized robust residuals
RDPLOT	Standardized robust residual vs. robust distance

The names of the graphs that PROC QUANTREG generates are listed in Table 72.7, along with the required statements and options. The following subsections provide information about these graphs.

Fit Plot

When the model has a single independent continuous variable (with or without the intercept), the QUANTREG procedure automatically creates a plot of fitted conditional quantiles against this independent variable for one or more quantiles specified in the MODEL statement.

The following example reuses the trout data set in the section "Analysis of Fish-Habitat Relationships" on page 5363 to show the fit plot for one or several quantiles.

```
ods graphics on;

proc quantreg data=trout ci=resampling;
   model LnDensity = WDRatio / quantile=0.9 ;
run;

proc quantreg data=trout ci=resampling;
   model LnDensity = WDRatio / quantile=0.5 0.75 0.9 ;
run;

ods graphics off;
```

For a single quantile, the confidence limits for the fitted conditional quantiles are also plotted if you specify the CI=RESAMPLING or CI=SPARSITY option. (See Figure 72.13.) For multiple quantiles, confidence limits are not plotted. (See Figure 72.14.)

The QUANTREG procedure also provides fit plots for quantile regression splines and polynomials if they are based on a single continuous variable. Refer to Example 72.4 and Example 72.5 for some examples.

Figure 72.13 Fit Plot with Confidence Limits

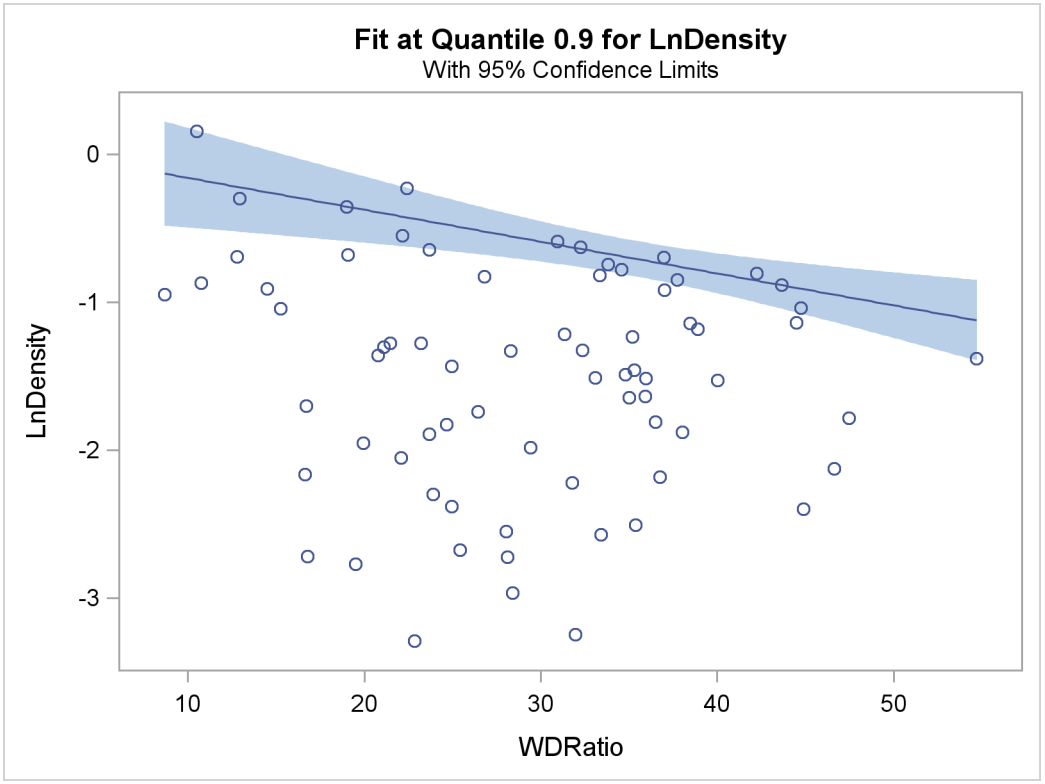
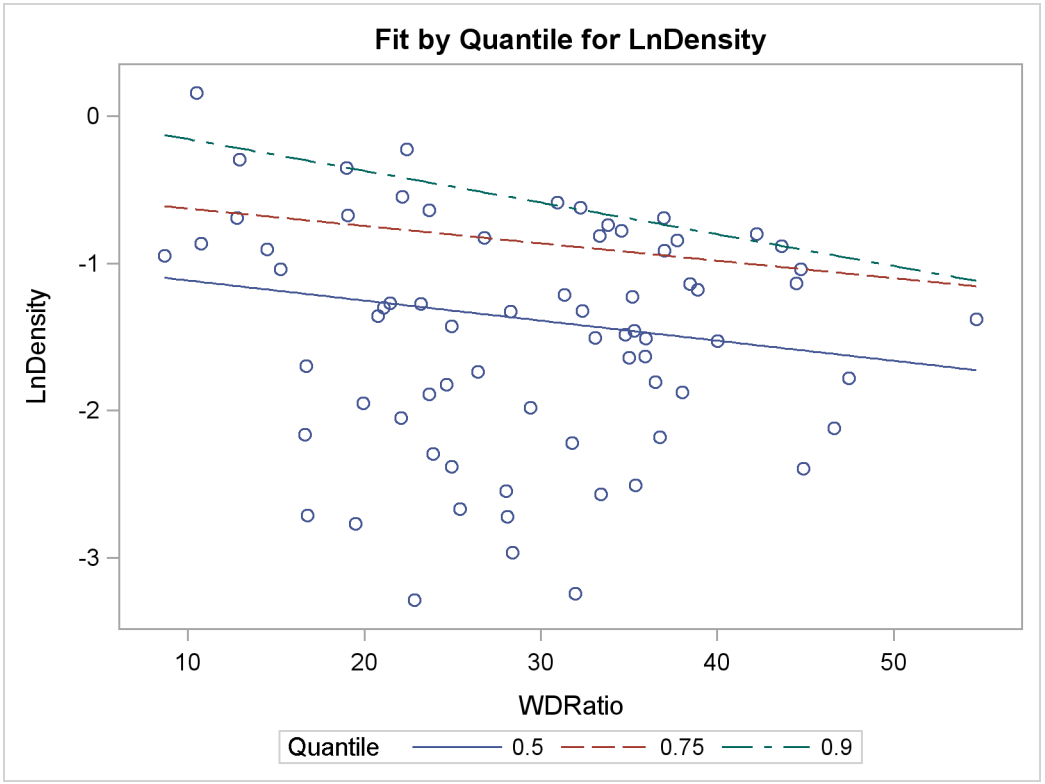


Figure 72.14 Fit Plot for Multiple Quantiles



Quantile Process Plot

A quantile process plot is a scatter plot of an estimated regression parameter against quantile. You can request this plot with the PLOT=QUANTPLOT option in the MODEL statement when multiple regression quantiles or the entire quantile process is computed. Quantile process plots are often used to check model variations at different quantiles, which is usually called model heterogeneity.

By default, panels are used to hold multiple process plots (up to four in each panel). You can use the UNPACK option to request individual process plots. Figure 72.9 in the section "Analysis of Fish-Habitat Relationships" on page 5363 shows a panel with two quantile process plots. Output 72.2.9 in Example 72.2 shows a single quantile process plot. Example 72.3 demonstrates more quantile process plots and their usage.

The diagnostics plots discussed in the following sections are available when you fit the regression model with a single quantile, especially the median.

Distance-Distance Plot

The distance-distance plot (DDPLOT) is mainly used for leverage-point diagnostics. It is a scatter plot of the robust distances against the classical Mahalanobis distances for the continuous independent variables. See the section "Leverage Point and Outlier Detection" on page 5397 for details about the robust distance. If there is a classification variable specified in the model, this plot is not created.

You can use the PLOT=DDPLOT option to request this plot. The following statements use the growth data set in Example 72.2 to create a single plot, shown in Output 72.2.4 in Example 72.2:

```
ods graphics on;

proc quantreg data=growth ci=resampling plot=ddplot;
    model GDP = lgdp2 mse2 fse2 fhe2 mhe2 lexp2
                lintr2 gedy2 Iy2 gcony2 lblakp2 pol2 ttrad2
                / quantile=.5 diagnostics leverage(cutoff=8);
    id Country;
run;

ods graphics off;
```

The reference lines represent the cutoff values. The diagonal line is also drawn to show the distribution of the distances. By default, all outliers and leverage points are labeled with observation numbers. To change the default, you can use the LABEL= option as described in Table 72.4.

Residual-Distance Plot

The residual-distance plot (RDPLOT) is used for both outlier and leverage-point diagnostics. It is a scatter plot of the standardized residuals against the robust distances. See the section "Leverage Point and Outlier Detection" on page 5397 for details about the robust distance. If a classification variable is specified in the model, this plot is not created.

You can use the PLOT=RDPLOT option to request this plot. The following statements use the growth data set in Example 72.2 to create a single plot, shown in Output 72.2.3 in Example 72.2:

```
ods graphics on;

proc quantreg data=growth ci=resampling plot=rdplot;
    model GDP = lgdp2 mse2 fse2 fhe2 mhe2 lexp2
                lintr2 gedy2 Iy2 gcony2 lblakp2 pol2 ttrad2
                / quantile=.5 diagnostics leverage(cutoff=8);
    id Country;
run;

ods graphics off;
```

The reference lines represent the cutoff values. By default, all outliers and leverage points are labeled with observation numbers. To change the default, you can use the LABEL= option as described in Table 72.4.

If you specify ID variables in the ID statement, instead of observation numbers, the values of the first ID variable are used as labels.

Histogram and Q-Q Plot

PROC QUANTREG produces a histogram and a Q-Q plot for the standardized residuals. The histogram is superimposed with a normal density curve and a kernel density curve. Using the growth data set in Example 72.2, the following statements create the plot shown in Output 72.2.5 in Example 72.2:

```
ods graphics on;

proc quantreg data=growth ci=resampling plot=histogram;
    model GDP = lgdp2 mse2 fse2 fhe2 mhe2 lexp2
                lintr2 gedy2 Iy2 gcony2 lblakp2 pol2 ttrad2
                / quantile=.5 diagnostics leverage(cutoff=8);
    id Country;
run;

ods graphics off;
```

ODS Graph Names

The QUANTREG procedure assigns a name to each graph it creates. You can use these names to reference the graphs when using ODS. The names are listed in Table 72.7.

To request these graphs you must specify the ODS GRAPHICS statement in addition to the plot options in the PROC statement or the MODEL statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Table 72.7 ODS Graphics Produced by PROC QUANTREG

ODS Graph Name	Plot Description	Statement	Option
DDPlot	Robust distance vs. Mahalanobis distance	PROC MODEL	DDPLOT
FitPlot	Quantile fit vs. independent variable	PROC MODEL	FITPLOT
Histogram	Histogram of standardized robust residuals	PROC MODEL	HISTOGRAM
QQPlot	Q-Q plot of standardized robust residuals	PROC MODEL	QQPLOT
QuantPanel	Panel of quantile plots with confidence limits	MODEL	QUANTPLOT
QuantPlot	Scatter plot for regression quantiles with confidence limits	MODEL	QUANTPLOT UNPACK
RDPlot	Standardized robust residual vs. robust distance	PROC MODEL	RDPLOT

Examples: QUANTREG Procedure

Example 72.1: Comparison of Algorithms

This example illustrates and compares the three algorithms for regression estimation available in the QUANTREG procedure. The simplex algorithm is the default because of its stability. Although this algorithm is slower than the interior point and smoothing algorithms for large data sets, the difference is not as significant for data sets with fewer than 5,000 observations and 50 variables. The simplex algorithm can also compute the entire quantile process, which is shown in Example 72.2.

The following statements generate 1,000 random observations. The first 950 observations are from a linear model, and the last 50 observations are significantly biased in the y-direction. In other words, 5% of the observations are contaminated with outliers.

```
data a (drop=i);
   do i=1 to 1000;
      x1=rannor(1234);
      x2=rannor(1234);
      e=rannor(1234);
      if i > 950 then y=100 + 10*e;
      else y=10 + 5*x1 + 3*x2 + 0.5 * e;
      output;
   end;
run;
```

The following statements invoke the QUANTREG procedure to fit a median regression model with the default simplex algorithm. They produce the results in Output 72.1.1 through Output 72.1.3.

```
proc quantreg data=a;
   model y = x1 x2;
run;
```

Output 72.1.1 displays model information and summary statistics for variables in the model. It indicates that the simplex algorithm is used to compute the optimal solution and the rank method is used to compute confidence intervals of the parameters.

By default, the QUANTREG procedure fits a median regression model. This is indicated by the quantile value 0.5 in Output 72.1.2, which also displays the objective function value and the predicted value of the response at the means of the covariates.

Output 72.1.3 displays parameter estimates and confidence limits. These estimates are reasonable, which indicates that median regression is robust to the 50 outliers.

Output 72.1.1 Model Fit Information and Summary Statistics with Simplex Algorithm

The QUANTREG Procedure

Model Information

Data Set	WORK.A
Dependent Variable	y
Number of Independent Variables	2
Number of Observations	1000
Optimization Algorithm	Simplex
Method for Confidence Limits	Inv_Rank

Summary Statistics

Variable	Q1	Median	Q3	Mean	Standard Deviation	MAD
x1	-0.6546	0.0230	0.7099	0.0222	0.9933	1.0085
x2	-0.7891	-0.0747	0.6839	-0.0401	1.0394	1.0857
y	6.1045	10.6936	14.9569	14.4864	20.4087	6.5696

Output 72.1.2 Quantile and Objective Function with Simplex Algorithm

Quantile and Objective Function

Quantile	0.5
Objective Function	2441.1927
Predicted Value at Mean	10.0259

Output 72.1.3 Parameter Estimates with Simplex Algorithm

Parameter Estimates

Parameter	DF	Estimate	95% Confidence Limits	
Intercept	1	10.0364	9.9959	10.0756
x1	1	5.0106	4.9602	5.0388
x2	1	3.0294	2.9944	3.0630

The following statements refit the model by using the interior point algorithm:

```
proc quantreg algorithm=interior(tolerance=1e-6)
              ci=none data=a;
   model y = x1 x2 / itprint nosummary;
run;
```

The TOLERANCE= option specifies the stopping criterion for convergence of the interior point algorithm, which is controlled by the duality gap. Although the default criterion is 1E−8, the value 1E−6 is often sufficient. The ITPRINT option requests the iteration history for the algorithm. The option CI=NONE suppresses the computation of confidence limits, and the option NOSUMMARY suppresses the table of summary statistics.

Output 72.1.4 displays model fit information.

Output 72.1.4 Model Fit Information with Interior Point Algorithm

The QUANTREG Procedure

Model Information

Data Set	WORK.A
Dependent Variable	y
Number of Independent Variables	2
Number of Observations	1000
Optimization Algorithm	Interior

Output 72.1.5 displays the iteration history of the interior point algorithm. Note that the duality gap is less than 1E−6 in the final iteration. The table also provides the number of iterations, the number of corrections, the primal step length, the dual step length, and the objective function value at each iteration.

Output 72.1.5 Iteration History for the Interior Point Algorithm

Iteration History of Interior Point Algorithm

Duality Gap	Iter	Correction	Primal Step	Dual Step	Objective Function
2623	1	1	0.3113	0.4910	3303.4688
3215	2	2	0.0427	1.0000	2461.3774
1127	3	3	0.9882	0.3653	2451.1337
760.88658	4	4	0.3381	1.0000	2442.8104
77.10290	5	5	1.0000	0.8916	2441.2627
8.43666	6	6	0.9370	0.8381	2441.2085
1.82868	7	7	0.8375	0.7674	2441.1985
0.40584	8	8	0.6980	0.8636	2441.1948
0.09550	9	9	0.9438	0.5955	2441.1930
0.00665	10	10	0.9818	0.9304	2441.1927
0.0002248	11	11	0.9179	0.9994	2441.1927
5.44651E-8	12	12	1.0000	1.0000	2441.1927

Output 72.1.6 displays the parameter estimates obtained with the interior point algorithm, which are identical to those obtained with the simplex algorithm.

Output 72.1.6 Parameter Estimates with Interior Point Algorithm

Parameter Estimates

Parameter	DF	Estimate
Intercept	1	10.0364
x1	1	5.0106
x2	1	3.0294

The following statements refit the model by using the smoothing algorithm. They produce the results in Output 72.1.7 through Output 72.1.9.

```
proc quantreg algorithm=smooth(rratio=.5) ci=none data=a;
   model y = x1 x2 / itprint nosummary;
run;
```

The RRATIO= option controls the reduction speed of the threshold. Output 72.1.7 displays the model fit information.

Output 72.1.7 Model Fit Information with Smoothing Algorithm

The QUANTREG Procedure

Model Information

Data Set	WORK.A
Dependent Variable	y
Number of Independent Variables	2
Number of Observations	1000
Optimization Algorithm	Smooth

Output 72.1.8 displays the iteration history of the smoothing algorithm. The threshold controls the convergence. Note that the thresholds decrease by a factor of at least 0.5, the value specified with the RRATIO= option. The table also provides the number of iterations, the number of factorizations, the number of full updates, the number of partial updates, and the objective function value in each iteration. For details concerning the smoothing algorithm, refer to Chen (2007).

Output 72.1.8 Iteration History for the Smoothing Algorithm

Iteration History of Smoothing Algorithm

Threshold	Iter	Refac	Full Update	Partial Update	Objective Function
227.24557	1	1	1000	0	4267.0988
116.94090	15	4	1480	2420	3631.9653
1.44064	17	4	1480	2583	2441.4719
0.72032	20	5	1980	2598	2441.3315
0.36016	22	6	2248	2607	2441.2369
0.18008	24	7	2376	2608	2441.2056
0.09004	26	8	2446	2613	2441.1997
0.04502	28	9	2481	2617	2441.1971
0.02251	30	10	2497	2618	2441.1956
0.01126	32	11	2505	2620	2441.1946
0.00563	34	12	2510	2621	2441.1933
0.00281	35	13	2514	2621	2441.1930
0.0000846	36	14	2517	2621	2441.1927
1E-12	37	14	2517	2621	2441.1927

Output 72.1.9 displays the parameter estimates obtained with the smoothing algorithm, which are identical to those obtained with the simplex and interior point algorithms. All three algorithms should have the same parameter estimates unless the problem does not have a unique solution.

Output 72.1.9 Parameter Estimates with Smoothing Algorithm

Parameter Estimates		
Parameter	**DF**	**Estimate**
Intercept	1	10.0364
x1	1	5.0106
x2	1	3.0294

The interior point algorithm and the smoothing algorithm offer better performance than the simplex algorithm for large data sets. Refer to Chen (2004) for more details on choosing an appropriate algorithm on the basis of data set size. All three algorithms should have the same parameter estimates, unless the optimization problem has multiple solutions.

Example 72.2: Quantile Regression for Econometric Growth Data

This example uses a SAS data set named growth, which contains economic growth rates for countries during two time periods, 1965–1975 and 1975–1985. The data come from a study by Barro and Lee (1994) and have also been analyzed by Koenker and Machado (1999).

There are 161 observations and 15 variables in the data set. The variables, which are listed in the following table, include the national growth rates (GDP) for the two periods, 13 covariates, and a name variable (Country) for identifying the countries in one of the two periods.

Variable	Description
Country	Country's Name and Period
GDP	Annual Change Per Capita GDP
lgdp2	Initial Per Capita GDP
mse2	Male Secondary Education
fse2	Female Secondary Education
fhe2	Female Higher Education
mhe2	Male Higher Education
lexp2	Life Expectancy
lintr2	Human Capital
gedy2	Education/GDP
Iy2	Investment/GDP
gcony2	Public Consumption/GDP
lblakp2	Black Market Premium
pol2	Political Instability
ttrad2	Growth Rate Terms Trade

The goal is to study the effect of the covariates on GDP. The following statements request median regression for a preliminary exploration. They produce the results in Output 72.2.1 through Output 72.2.6.

```
data growth;
   length Country$ 22;
   input Country GDP lgdp2 mse2 fse2 fhe2 mhe2 lexp2 lintr2 gedy2
         Iy2 gcony2 lblakp2 pol2 ttrad2 @@;
datalines;
Algeria75          .0415 7.330 .1320 .0670 .0050 .0220 3.880 .1138 .0382
                   .1898 .0601 .3823 .0833 .1001
Algeria85          .0244 7.745 .2760 .0740 .0070 .0370 3.978 -.107 .0437
                   .3057 .0850 .9386 .0000 .0657
Argentina75        .0187 8.220 .7850 .6200 .0740 .1660 4.181 .4060 .0221
                   .1505 .0596 .1924 .3575 -.011
Argentina85        -.014 8.407 .9360 .9020 .1320 .2030 4.211 .1914 .0243
                   .1467 .0314 .3085 .7010 -.052


... more lines ...

Zambia75           .0120 6.989 .3760 .1190 .0130 .0420 3.757 .4388 .0339
                   .3688 .2513 .3945 .0000 -.032
Zambia85           -.046 7.109 .4200 .2740 .0110 .0270 3.854 .8812 .0477
                   .1632 .2637 .6467 .0000 -.033
Zimbabwe75         .0320 6.860 .1450 .0170 .0080 .0450 3.833 .7156 .0337
                   .2276 .0246 .1997 .0000 -.040
Zimbabwe85         -.011 7.180 .2200 .0650 .0060 .0400 3.944 .9296 .0520
                   .1559 .0518 .7862 .7161 -.024
;
ods graphics on;

proc quantreg data=growth ci=resampling
                plots=(rdplot ddplot reshistogram);
    model GDP = lgdp2 mse2 fse2 fhe2 mhe2 lexp2
                lintr2 gedy2 Iy2 gcony2 lblakp2 pol2 ttrad2
                / quantile=.5 diagnostics leverage(cutoff=8);
    id Country;
    test_lgdp2: test lgdp2 / lr wald;
run;

ods graphics off;
```

The QUANTREG procedure employs the default simplex algorithm to estimate the parameters. The MCMB resampling method is used to compute confidence limits.

Output 72.2.1 displays model information and summary statistics for the variables in the model. Six summary statistics are computed, including the median and the median absolute deviation (MAD), which are robust measures of univariate location and scale, respectively. For the variable lintr2 (Human Capital), both the mean and standard deviation are much larger than the corresponding robust measures, median and MAD. This indicates that this variable might have outliers.

Output 72.2.2 displays parameter estimates and 95% confidence limits computed with the rank method.

Output 72.2.1 Model Information and Summary Statistics

The QUANTREG Procedure

Model Information

Data Set	WORK.GROWTH
Dependent Variable	GDP
Number of Independent Variables	13
Number of Observations	161
Optimization Algorithm	Simplex
Method for Confidence Limits	Resampling

Summary Statistics

Variable	Q1	Median	Q3	Mean	Standard Deviation	MAD
lgdp2	6.9890	7.7450	8.6080	7.7905	0.9543	1.1579
mse2	0.3160	0.7230	1.2675	0.9666	0.8574	0.6835
fse2	0.1270	0.4230	0.9835	0.7117	0.8331	0.5011
fhe2	0.0110	0.0350	0.0890	0.0792	0.1216	0.0400
mhe2	0.0400	0.1060	0.2060	0.1584	0.1752	0.1127
lexp2	3.8670	4.0640	4.2430	4.0440	0.2028	0.2728
lintr2	0.00160	0.5604	1.8805	1.4625	2.5491	1.0058
gedy2	0.0248	0.0343	0.0466	0.0360	0.0141	0.0151
Iy2	0.1396	0.1955	0.2671	0.2010	0.0877	0.0981
gcony2	0.0480	0.0767	0.1276	0.0914	0.0617	0.0566
lblakp2	0	0.0696	0.2407	0.1916	0.3070	0.1032
pol2	0	0.0500	0.2429	0.1683	0.2409	0.0741
ttrad2	-0.0240	-0.0100	0.00730	-0.00570	0.0375	0.0239
GDP	0.00290	0.0196	0.0351	0.0191	0.0248	0.0237

Output 72.2.2 Parameter Estimates

Parameter Estimates

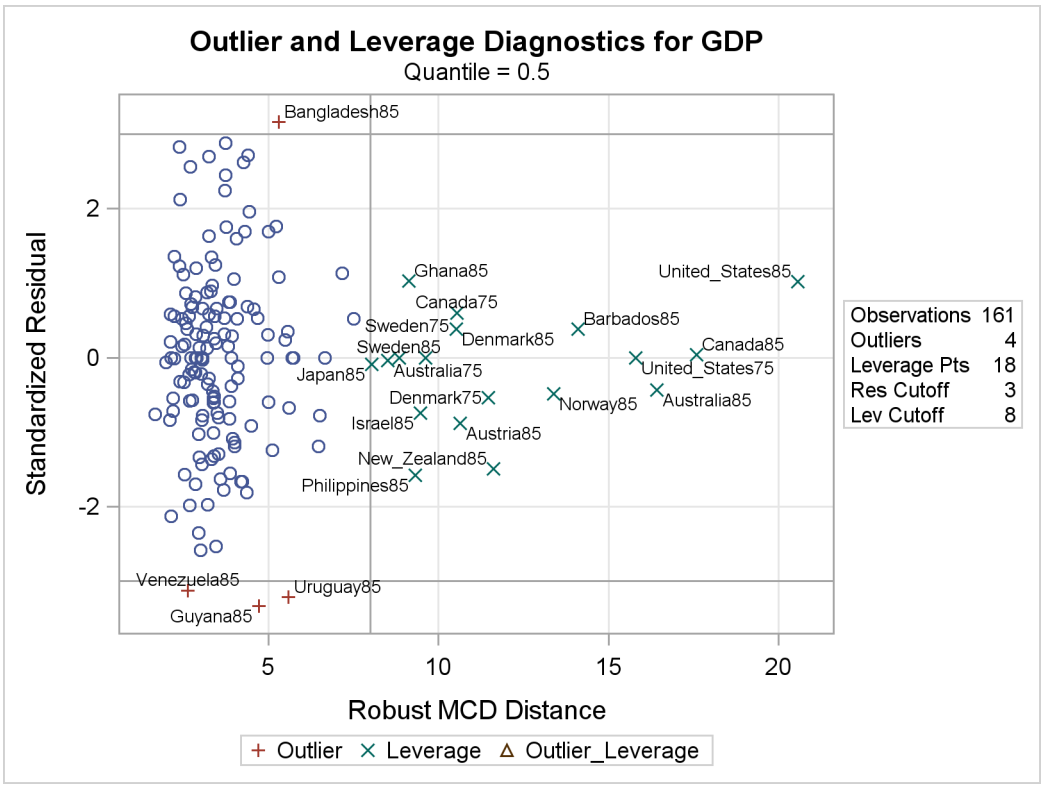
Parameter	DF	Estimate	Standard Error	95% Confidence Limits		t Value	Pr > |t|
Intercept	1	-0.0488	0.0679	-0.1829	0.0853	-0.72	0.4731
lgdp2	1	-0.0269	0.0044	-0.0356	-0.0182	-6.13	<.0001
mse2	1	0.0110	0.0079	-0.0046	0.0266	1.40	0.1640
fse2	1	-0.0011	0.0080	-0.0169	0.0146	-0.14	0.8860
fhe2	1	0.0148	0.0345	-0.0534	0.0831	0.43	0.6681
mhe2	1	0.0043	0.0300	-0.0551	0.0636	0.14	0.8869
lexp2	1	0.0683	0.0225	0.0238	0.1128	3.04	0.0028
lintr2	1	-0.0022	0.0015	-0.0051	0.0007	-1.51	0.1336
gedy2	1	-0.0508	0.1662	-0.3793	0.2776	-0.31	0.7601
Iy2	1	0.0723	0.0243	0.0244	0.1203	2.98	0.0034
gcony2	1	-0.0935	0.0397	-0.1720	-0.0151	-2.36	0.0198
lblakp2	1	-0.0269	0.0083	-0.0433	-0.0106	-3.26	0.0014
pol2	1	-0.0301	0.0098	-0.0494	-0.0108	-3.08	0.0025
ttrad2	1	0.1613	0.0796	0.0039	0.3186	2.03	0.0447

Diagnostics for the median regression fit are displayed in Output 72.2.3 and Output 72.2.4, which are requested with the PLOTS= option. Output 72.2.3 plots the standardized residuals from median regression against the robust MCD distance. This display is used to diagnose both vertical outliers and horizontal leverage points. Output 72.2.4 plots the robust MCD distance against the Mahalanobis distance. This display is used to diagnose leverage points.

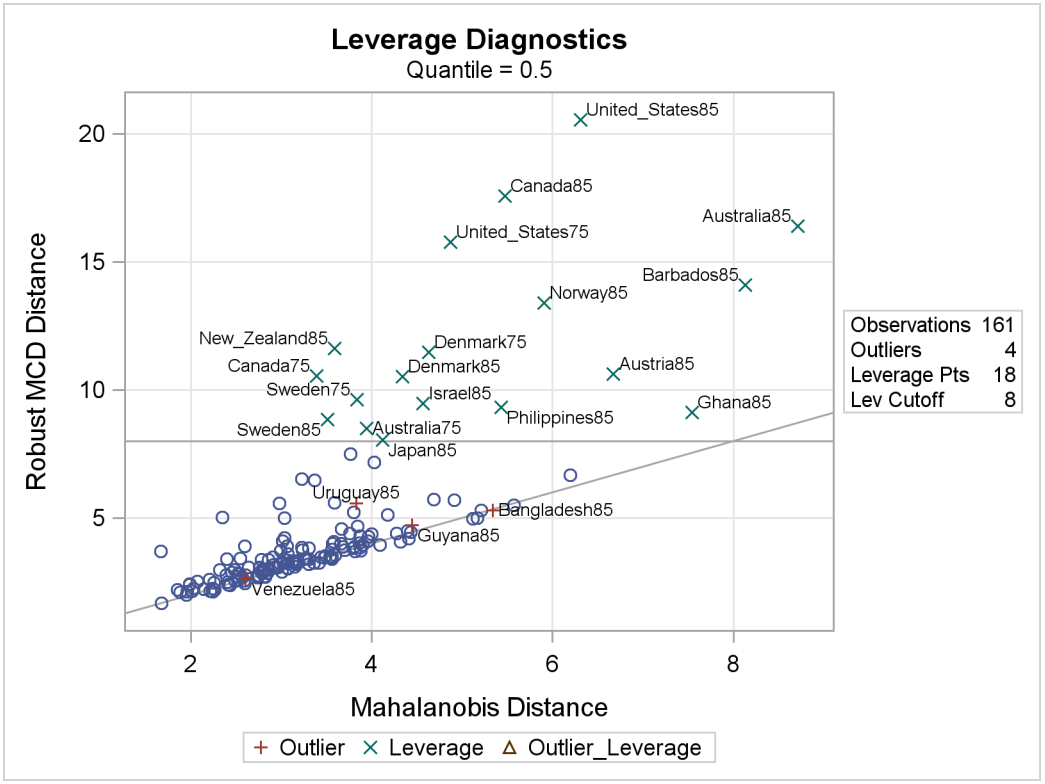
The cutoff value 8 specified with the LEVERAGE option is close to the maximum of the Mahalanobis distance. Eighteen points are diagnosed as high leverage points, and almost all are countries with high human capital, which is the major contributor to the high leverage as observed from the summary statistics. Four points are diagnosed as outliers by using the default cutoff value of 3. However, these are not extreme outliers.

A histogram of the standardized residuals and two fitted density curves are displayed in Output 72.2.5. This shows that median regression fits the data well.

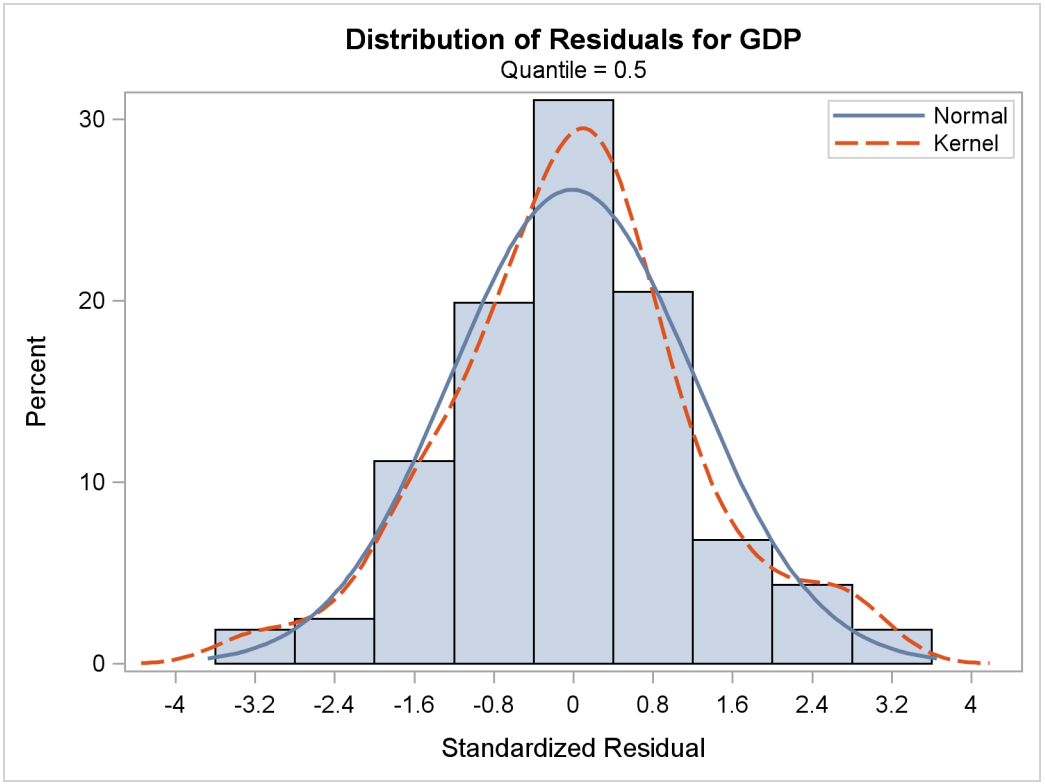
Output 72.2.3 Residual-Robust Distance Plot



Output 72.2.4 Robust Distance-Mahalanobis Distance Plot



Output 72.2.5 Histogram for Residuals



Tests of significance for the initial per-capita GDP (LGDP2) are shown in Output 72.2.6.

Output 72.2.6 Tests for Regression Coefficient

Tests

test_lgdp2

Test	Test Statistic	DF	Chi-Square	Pr > ChiSq
Wald	37.5857	1	37.59	<.0001
Likelihood Ratio	36.3047	1	36.30	<.0001

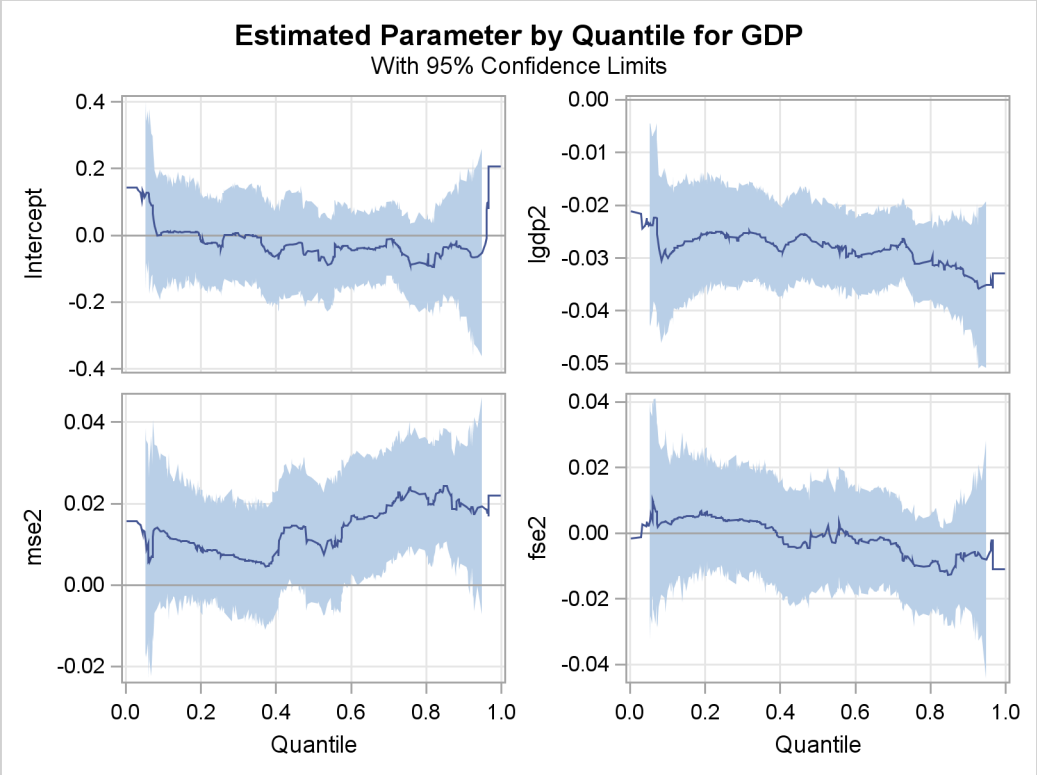
The QUANTREG procedure computes entire quantile processes for covariates when you specify QUANTILE=PROCESS in the MODEL statement, as follows:

```
ods graphics on;

proc quantreg data=growth ci=resampling;
    model GDP = lgdp2 mse2 fse2 fhe2 mhe2 lexp2 lintr2
                gedy2 Iy2 gcony2 lblakp2 pol2 ttrad2
                / quantile=process plot=quantplot;
run;

ods graphics off;
```

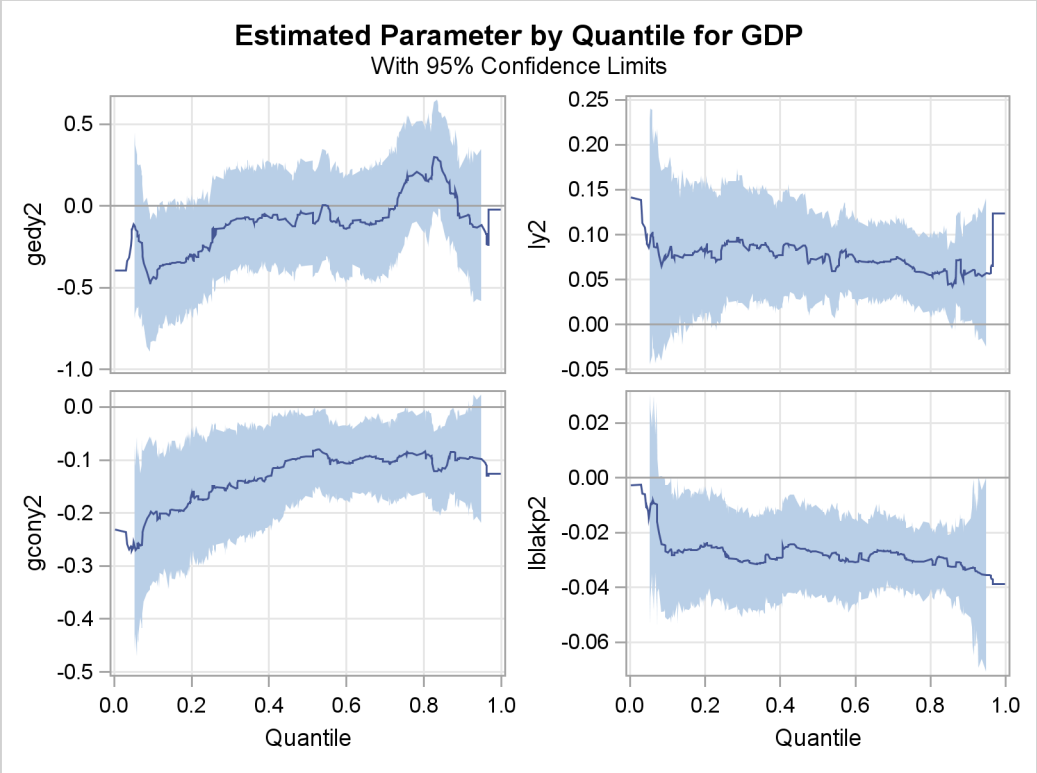
Confidence limits for quantile processes can be computed with the sparsity or resampling methods, but not the rank method, because the computation would be prohibitively expensive.

A total of 14 quantile process plots are produced. Output 72.2.7 and Output 72.2.8 display two panels of eight selected process plots. The 95% confidence bands are shaded.

Output 72.2.7 Quantile Processes with 95% Confidence Bands



Output 72.2.8 Quantile Processes with 95% Confidence Bands

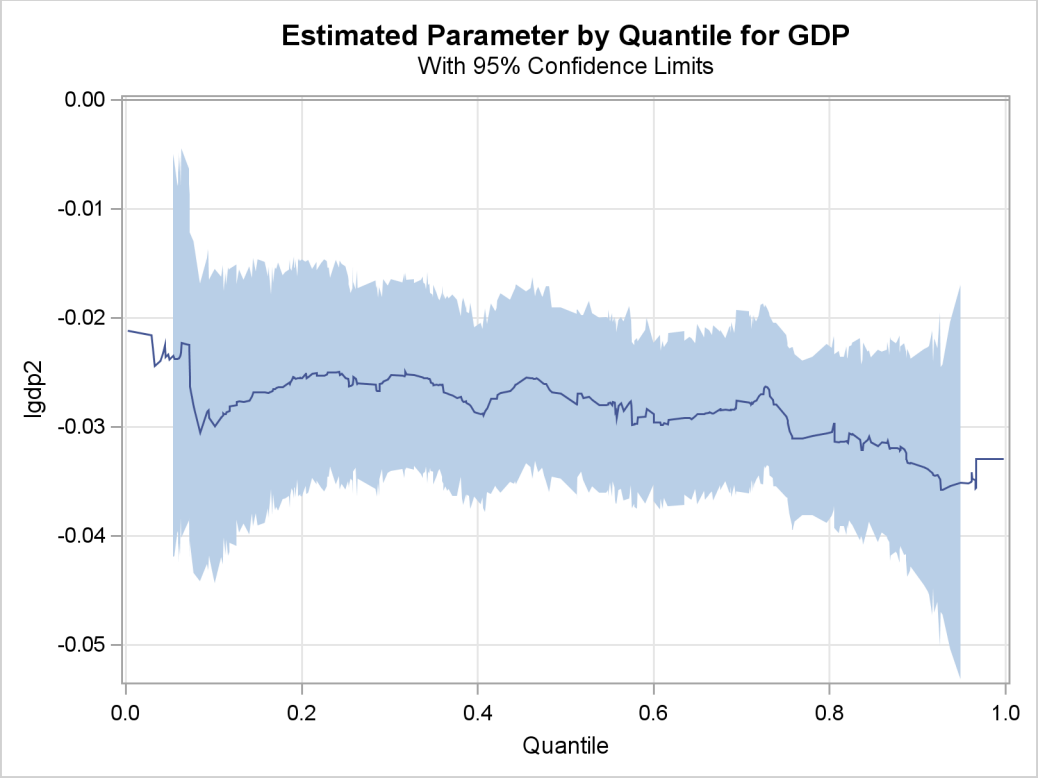


As pointed out by Koenker and Machado (1999), previous studies of the Barro growth data have focused on the effect of the initial per-capita GDP on the growth of this variable (annual change per-capita GDP). A single process plot for this effect can be requested with the following statements:

```
ods graphics on;

proc quantreg data=growth ci=resampling;
    model GDP = lgdp2 mse2 fse2 fhe2 mhe2 lexp2 lintr2
                gedy2 Iy2 gcony2 lblakp2 pol2 ttrad2
                / quantile=process plot=quantplot(lgdp2);
run;

ods graphics off;
```

The plot is shown in Output 72.2.9.

Output 72.2.9 Quantile Process Plot for LGDP2



The confidence bands here are computed with the MCMB resampling method, unlike in Koenker and Machado (1999), where the rank method was used to compute confidence limits for a few selected points. Output 72.2.9 suggests that the effect of the initial level of GDP is relatively constant over the entire distribution, with a slightly stronger effect in the upper tail.

The effects of other covariates are quite varied. An interesting covariate is public consumption/GDP (gcony2) (first plot in second panel), which has a constant effect over the upper half of the distribution and a larger effect in the lower tail. For an analysis of the effects of the other covariates, refer to Koenker and Machado (1999).

Example 72.3: Quantile Regression Analysis of Birth-Weight Data

This example is patterned after a quantile regression analysis of covariates associated with birth weight that was carried out by Koenker and Hallock (2001). Their study used a subset of the June 1997 Detailed Natality Data published by the National Center for Health Statistics and demonstrated that conditional quantile functions provide more complete information about the covariate effects than ordinary least squares regression.

As in Koenker and Hallock (2001) and Abreveya (2001), this example uses data for live, singleton births to mothers in the United States who were recorded as black or white, and who were between the ages of 18 and 45. For convenience, this example uses 50,000 observations, which were randomly selected from the qualified observations. Observations with missing data for any of the variables were deleted.

The following table describes the variables in the data.

Variable	Description
weight	Infant's Birth Weight
black	Indicator of Black Mother
married	Indicator of Married Mother
boy	Indicator of Boy
visit	Prenatal Visit: 0 - no visit, 1 - visit in second trimester, 2 - visit in last trimester, 3 - visit in first trimester
ed	Mother's Education Level: 0 - high school, 1 - some college, 2 - college, 3 - less than high school
smoke	Indicator of Smoking Mother
cigsper	Number of Cigarettes Smoked per Day
mom_age	Mother's Age
m_wtgain	Mother's Weight Gain during Pregnancy

There are four levels of education of the mother. By default, the QUANTREG procedure treats the highest level (3 - less than high school) as a reference level. The regression coefficients of other levels measure the effect relative to this level. Likewise, there are four levels of prenatal medical care of the mother, and a first visit in the first trimester serves as the reference level. These two variables are treated as classification variables in the model.

The following statements fit a regression model for 19 quantiles of birth weight, which are evenly spaced in the interval $(0, 1)$. The model includes linear and quadratic effects for the age of the mother and for weight gain during pregnancy.

```
ods graphics on;

proc quantreg ci=sparsity/iid algorithm=interior(tolerance=1.e-4)
              data=sashelp.bweight;
   class visit ed;
   model weight = black married boy visit ed smoke
                  cigsper mom_age mom_age*mom_age
                  m_wtgain m_wtgain*m_wtgain /
                  quantile= 0.05 to 0.95 by 0.05
                  plot=quantplot;
run;

ods graphics off;
```

Output 72.3.1 Model Information and Summary Statistics

The QUANTREG Procedure

Model Information

Data Set	SASHELP.BWEIGHT
Dependent Variable	weight
Number of Independent Variables	9
Number of Continuous Independent Variables	7
Number of Class Independent Variables	2
Number of Observations	50000
Optimization Algorithm	Interior
Method for Confidence Limits	Sparsity

Summary Statistics

Variable	Q1	Median	Q3	Mean	Standard Deviation	MAD
black	0	0	0	0.1628	0.3692	0
married	0	1.0000	1.0000	0.7126	0.4525	0
boy	0	1.0000	1.0000	0.5158	0.4998	0
smoke	0	0	0	0.1307	0.3370	0
cigsper	0	0	0	1.4766	4.6541	0
mom_age	-4.0000	0	5.0000	0.4161	5.7285	5.9304
mom_age*mom_age	4.0000	16.0000	49.0000	32.9877	39.2861	22.2390
m_wtgain	-8.0000	0	9.0000	0.7092	12.8761	11.8608
m_wtgain*m_wtgain	16.0000	64.0000	196.0	166.3	298.8	88.9561
weight	3062.0	3402.0	3720.0	3370.8	566.4	504.1

Output 72.3.1 displays the model information and summary statistics for the variables in the model.

Among the 11 independent variables, black, married, boy, and smoke are binary variables. For these variables, the mean represents the proportion in the category. The two continuous variables, mom_age and m_wtgain, are centered at their medians, which are 27 and 30, respectively.

The quantile plots for the intercept and the other 15 factors with nonzero degree of freedom are shown in the following four panels. In each plot, the regression coefficient at a given quantile

indicates the effect on birth weight of a unit change in that factor, assuming that the other factors are fixed. The bands represent 95% confidence intervals.

Although the data set used here is a subset of the Natality data set, the results are quite similar to those of Koenker and Hallock (2001) for the full data set.

In Output 72.3.2, the first plot is for the intercept. As explained by Koenker and Hallock (2001), the intercept "may be interpreted as the estimated conditional quantile function of the birth-weight distribution of a girl born to an unmarried, white mother with less than a high school education, who is 27 years old and had a weight gain of 30 pounds, didn't smoke, and had her first prenatal visit in the first trimester of the pregnancy."

The second plot shows that infants born to black mothers weigh less than infants born to white mothers, especially in the lower tail of the birth-weight distribution. The third plot shows that marital status has a large positive effect on birth weight, especially in the lower tail. The fourth plot shows that boys weigh more than girls for any chosen quantile; this difference is smaller in the lower quantiles of the distribution.

In Output 72.3.3, the first three plots deal with prenatal care. Compared with babies born to mothers who had a prenatal visit in the first trimester, babies born to mothers who received no prenatal care weigh less, especially in the lower quantiles of the birth-weight distributions. As noted by Koenker and Hallock (2001), "babies born to mothers who delayed prenatal visits until the second or third trimester have substantially *higher* birthweights in the lower tail than mothers who had a prenatal visit in the first trimester. This might be interpreted as the self-selection effect of mothers confident about favorable outcomes."

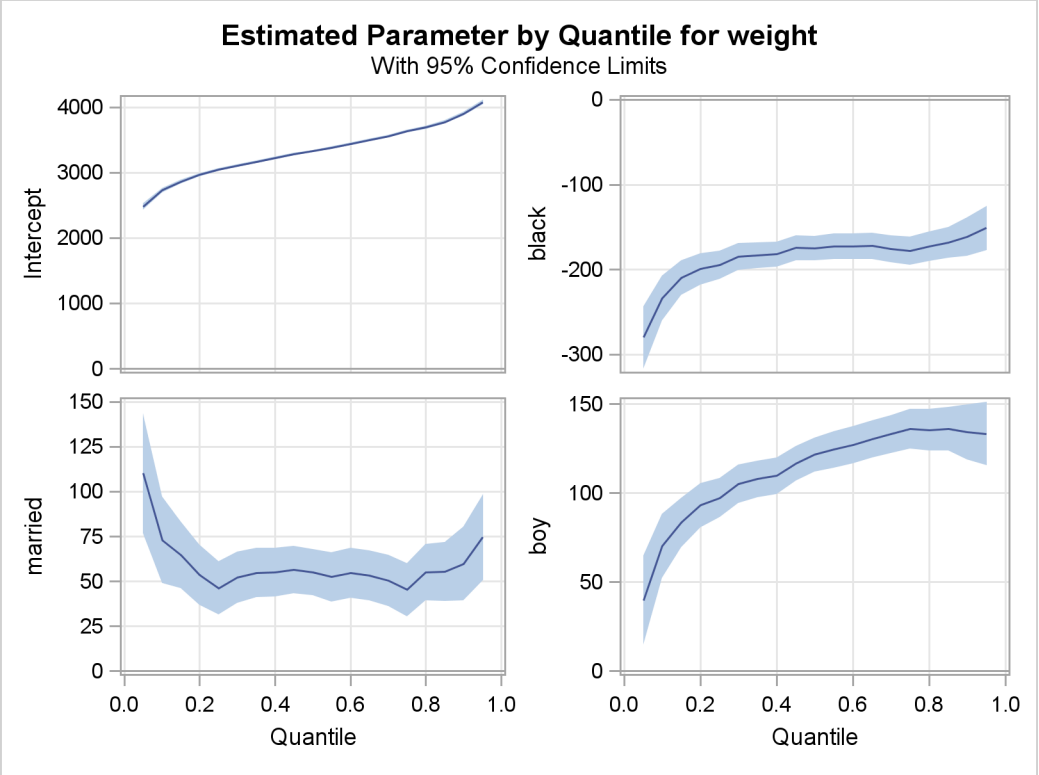
The fourth plot in Output 72.3.3 and the first two plots in Output 72.3.4 are for variables related to education. Education beyond high school is associated with a positive effect on birth weight. The effect of high school education is uniformly around 15 grams across the entire birth-weight distribution (this is a pure location shift effect), while the effect of some college and college education is more positive in the lower quantiles than the upper quantiles.

The remaining two plots in Output 72.3.4 show that smoking is associated with a large negative effect on birth weight.

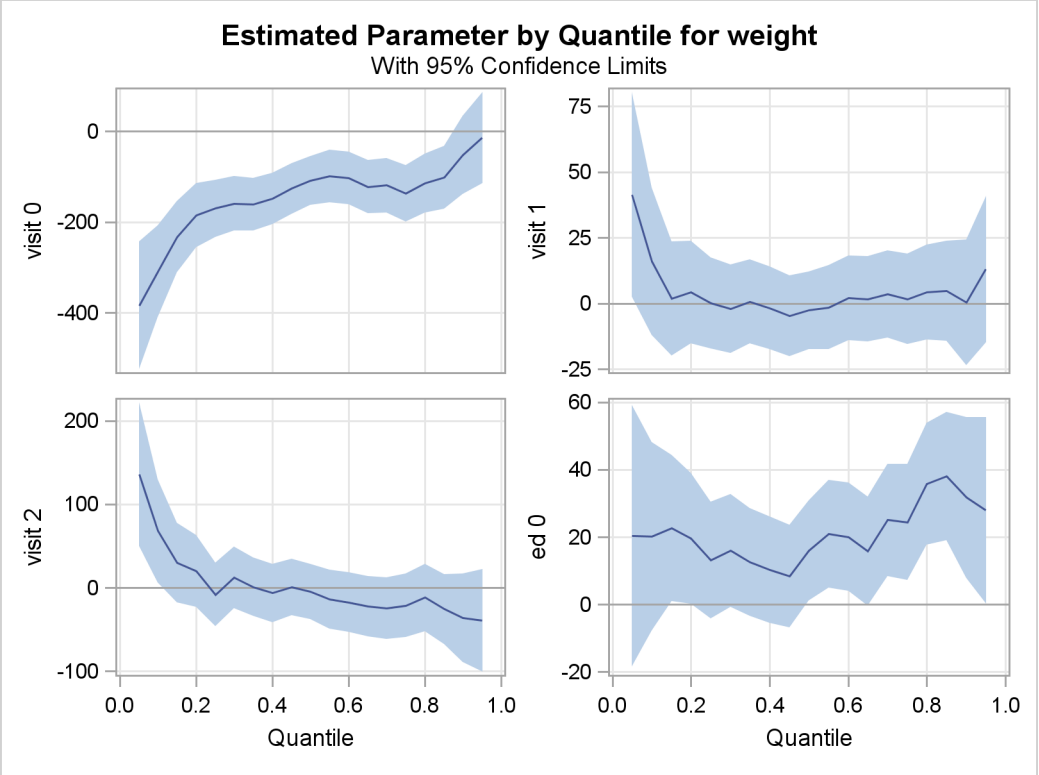
The linear and quadratic effects for the two continuous variables are shown in Output 72.3.5. Both of these variables are centered at their median. At the lower quantiles, the quadratic effect of the mother's age is more concave. The optimal age at the first quantile is about 33, and the optimal age at the third quantile is about 38. The effect of the mother's weight gain is clearly positive, as indicated by the narrow confidence bands for both linear and quadratic coefficients.

Refer to Koenker and Hallock (2001) for more details about the covariate effects discovered with quantile regression.

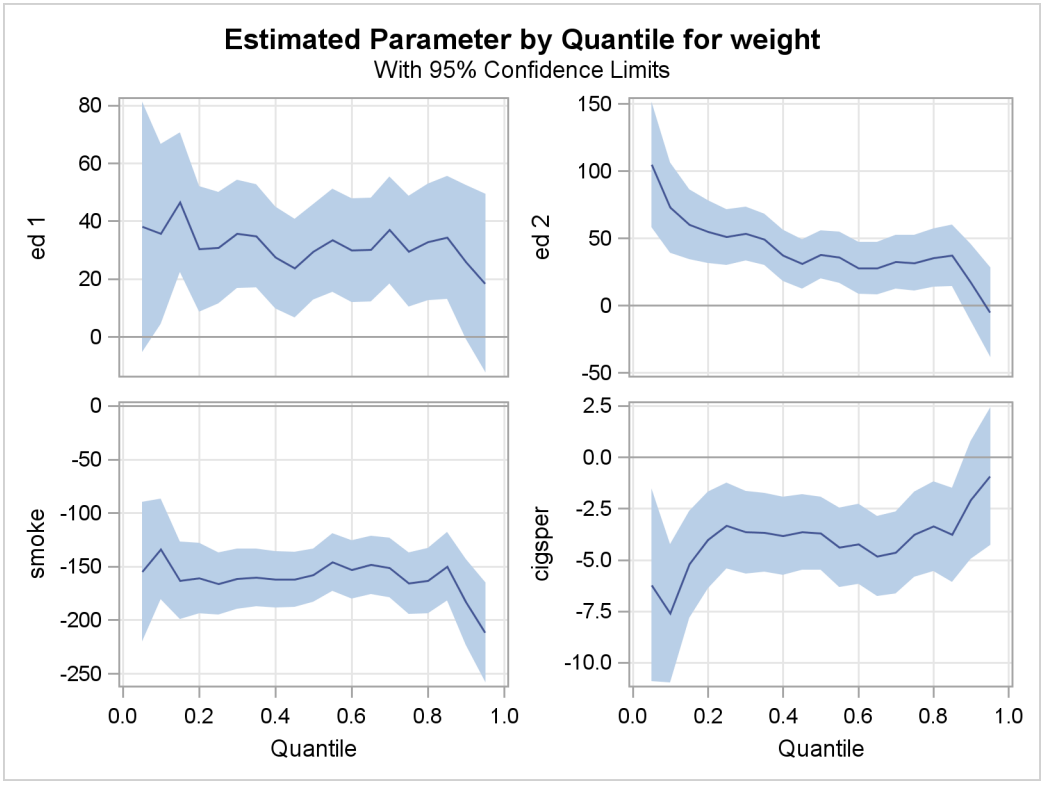
Output 72.3.2 Quantile Processes with 95% Confidence Bands



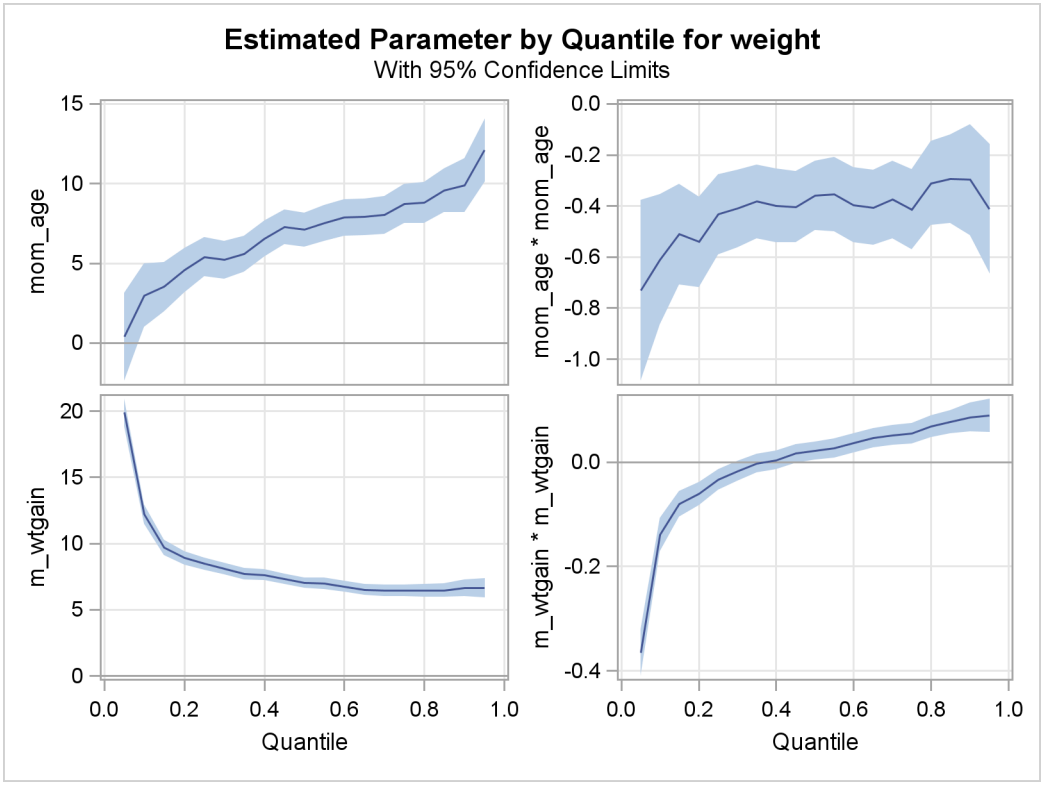
Output 72.3.3 Quantile Processes with 95% Confidence Bands



Output 72.3.4 Quantile Processes with 95% Confidence Bands



Output 72.3.5 Quantile Processes with 95% Confidence Bands



Example 72.4: Nonparametric Quantile Regression for Ozone Levels

Tracing seasonal trends in the level of tropospheric ozone is essential for predicting high-level periods, observing long-term trends, and discovering potential changes in pollution. Traditional methods for modeling seasonal effects are based on the conditional mean of ozone concentration; however, the upper conditional quantiles are more critical from a public health perspective. In this example, the QUANTREG procedure fits conditional quantile curves for seasonal effects by using nonparametric quantile regression with cubic B-splines.

The data used here are from Chock, Winkler, and Chen (2000), who studied the association between daily mortality and ambient air pollutant concentrations in Pittsburgh, Pennsylvania. The data set ozone contains the following two variables: ozone (daily-maximum one-hour ozone concentration (ppm)) and days (index of 1,095 days (3 years)).

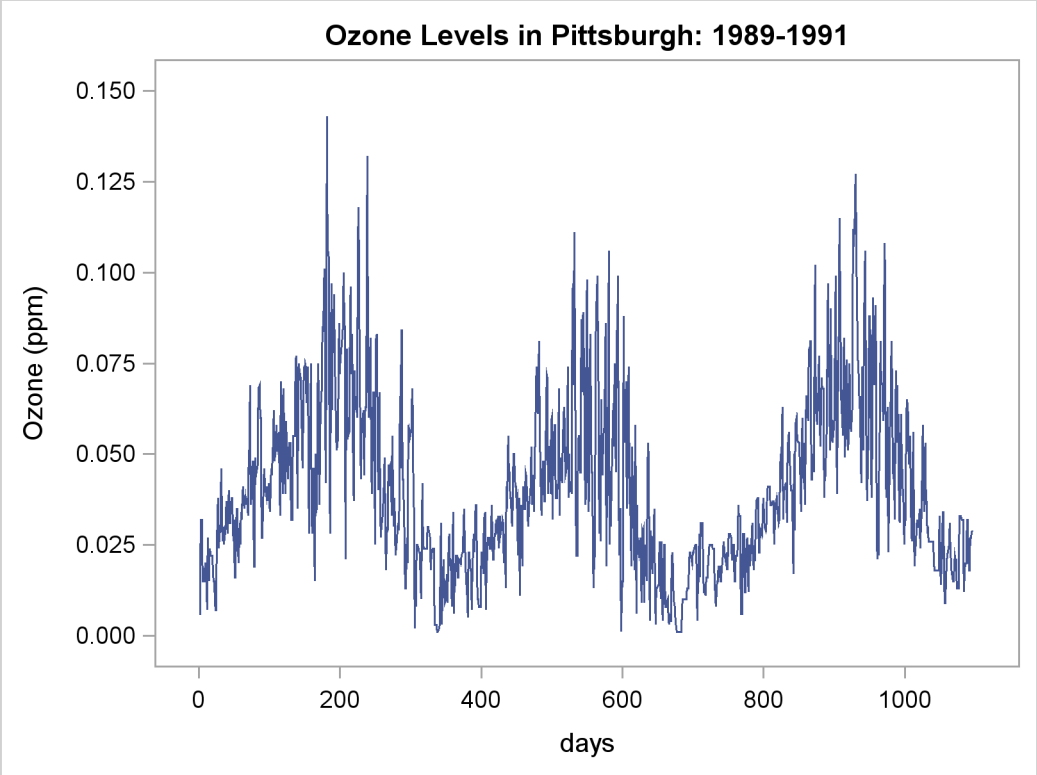
```
data ozone;
  days = _n_;
  input ozone @@;
datalines;
0.0060 0.0060 0.0320 0.0320 0.0320 0.0150 0.0150 0.0150 0.0200 0.0200
0.0160 0.0070 0.0270 0.0160 0.0150 0.0240 0.0220 0.0220 0.0220 0.0185
0.0150 0.0150 0.0110 0.0070 0.0070 0.0240 0.0380 0.0240 0.0265 0.0290

... more lines ...

0.0220 0.0210 0.0210 0.0130 0.0130 0.0130 0.0330 0.0330 0.0330 0.0325
0.0320 0.0320 0.0320 0.0120 0.0200 0.0200 0.0200 0.0320 0.0320 0.0250
0.0180 0.0180 0.0270 0.0270 0.0290
;
```

Output 72.4.1, which displays the time series plot of ozone concentration for the three years, shows a clear seasonal pattern.

In this example, cubic B-splines are used to fit the seasonal effect. These splines are generated with 11 knots, which split the 3 years into 12 seasons. The following statements create the spline basis and fit multiple quantile regression spline curves:

```
ods graphics on;

proc quantreg data=ozone algorithm=smooth plot=fitplot(nodata);
   effect sp = spline( days / knotmethod = list
          (90 182 272 365 455 547 637 730 820 912 1002) );
   model ozone = sp / quantile = 0.5 0.75 0.90 0.95;
run;

ods graphics off;
```

Output 72.4.1 Time Series of Ozone Levels in Pittsburgh, Pennsylvania

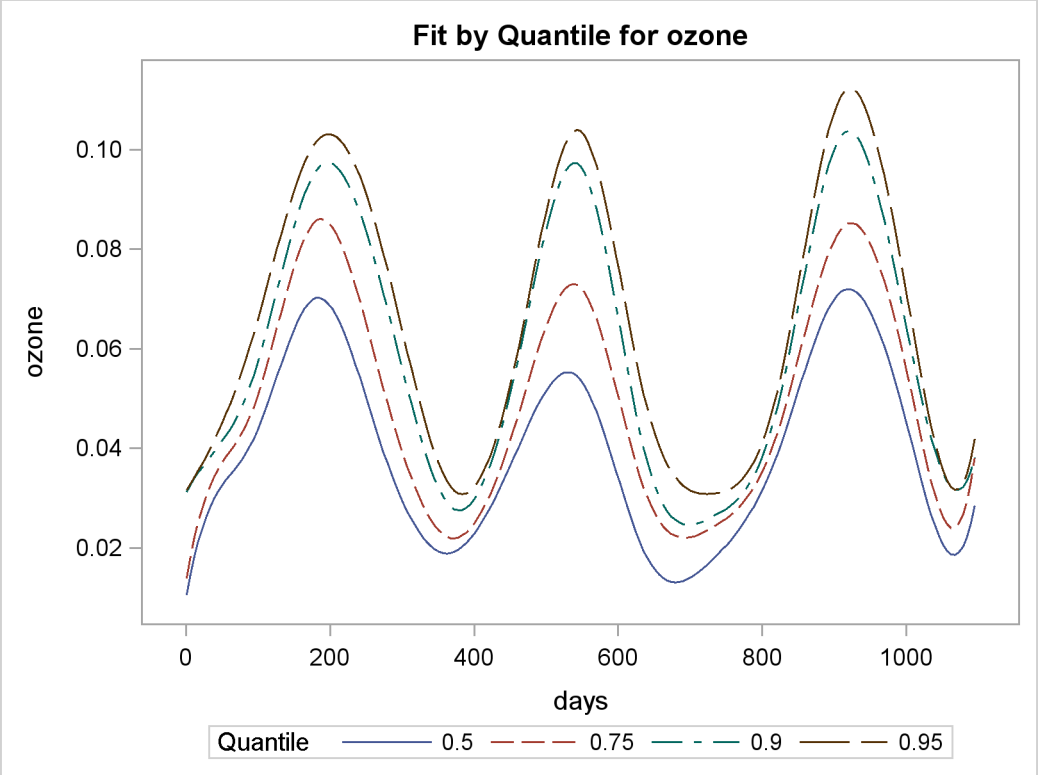


The EFFECT statement creates spline bases for the variable days. The KNOTMETHOD=LIST option provides all internal knots for these bases. Cubic spline bases are generated by default. These bases are treated as components of the spline effect *sp*, which is used in the MODEL statement. Spline fits for four quantiles are requested with the QUANTILE= option.

When you enable ODS Graphics, the QUANTREG procedure automatically generates a fit plot, which includes all fitted curves.

Output 72.4.2 displays these curves obtained with the QUANTREG procedure. The curves show that peak ozone levels occur in the summer. For the three years (1989–1991), the median curve (labeled 50%) does not cross the 0.08 ppm line, which is the 1997 EPA 8-hour standard. The median curve and the 75% curve show a drop for the ozone concentration levels in 1990. However, with the 90% and 95% curves, peak ozone levels tend to increase. This indicates that there might have been more days with low ozone concentration in 1990, but the top 10% and 5% tend to have higher ozone concentration levels.

Output 72.4.2 Quantiles of Ozone Levels in Pittsburgh, Pennsylvania



The quantile curves also show that high ozone concentration in 1989 had a longer duration than in 1990 and 1991. This is indicated by the wider spread of the quantile curves in 1989.

Example 72.5: Quantile Polynomial Regression for Salary Data

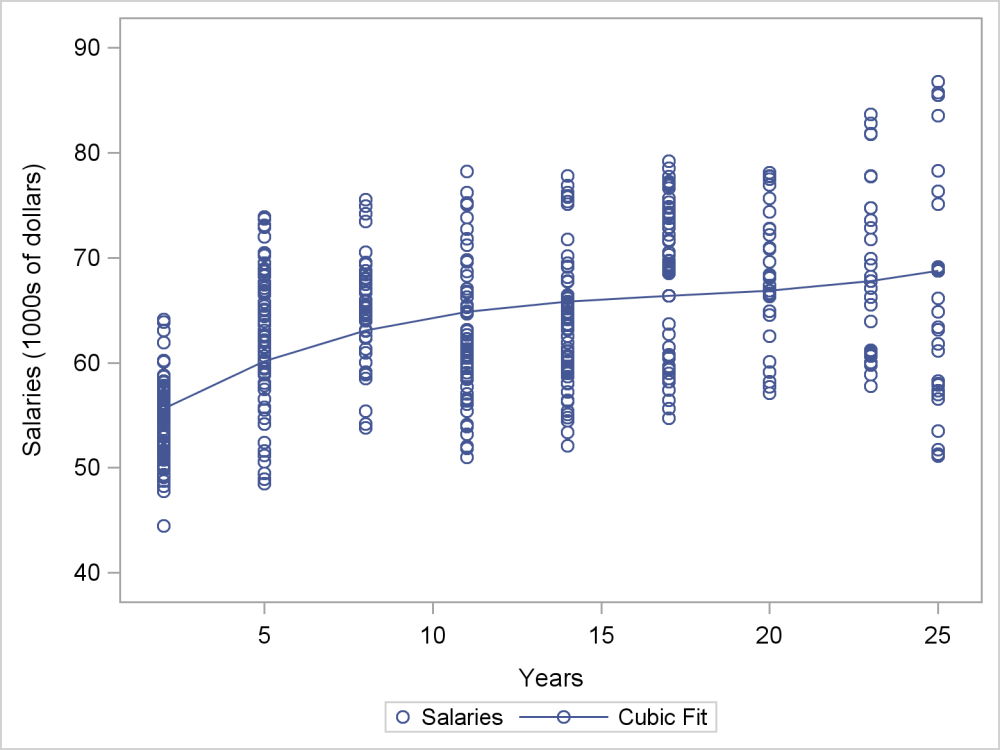
This example uses the data set from a university union survey of salaries of professors in 1991. The survey covered departments in U.S. colleges and universities that list programs in statistics. The goal here is to examine the relationship between faculty salaries and years of service.

The data include salaries and years of service for 459 professors. The scatter plot in Output 72.5.1 shows that the relationship is not linear, and a quadratic or cubic regression curve is appropriate. Output 72.5.1 shows a cubic curve.

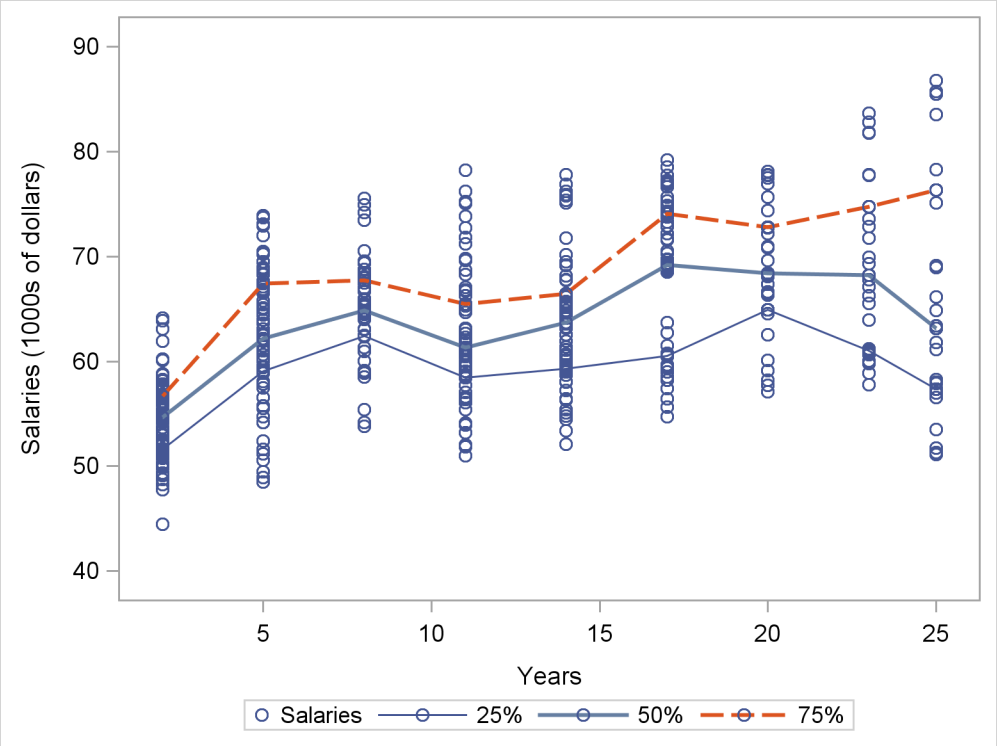
The curve in Output 72.5.1 does not adequately describe the conditional salary distributions and how they change with length of service. Output 72.5.2 shows the 25th, 50th, and 75th percentiles for each number of years, which gives a better picture of the conditional distributions.

```
data salary;
  label salaries='Salaries (1000s of dollars)'
        years   ='Years';
  input salaries years @@;
datalines;
54.94   2  58.24   2  58.11   2  52.23   2  52.98    2  57.62    2
44.48   2  57.22   2  54.24   2  54.79   2  56.42    2  61.90    2
63.90   2  64.10   2  47.77   2  54.86   2  49.31    2  53.37    2

... more lines ...

85.72  25  64.87  25  51.76  25  51.11  25  51.31   25  78.28   25
57.91  25  86.78  25  58.27  25  56.56  25  76.33   25  61.83   25
69.13  25  63.15  25  66.13  25
;
```

Output 72.5.1 Salary with Years as Professor: Cubit Fit



Output 72.5.2 Salary with Years as Professor: Sample Quantiles



These descriptive percentiles do not clearly show trends with length of service. The following statements use the QUANTREG procedure to obtain a smooth version by using polynomial quantile regression. The results are shown in Output 72.5.3 and Output 72.5.4.

```
ods graphics on;

proc quantreg data=salary ci=none;
   model salaries = years years*years years*years*years
                    /quantile=0.25 0.5 0.75;
run;

ods graphics off;
```

Output 72.5.3 shows the regression coefficients for the three quantiles.

Output 72.5.3 Regression Coefficients

The QUANTREG Procedure

Parameter Estimates

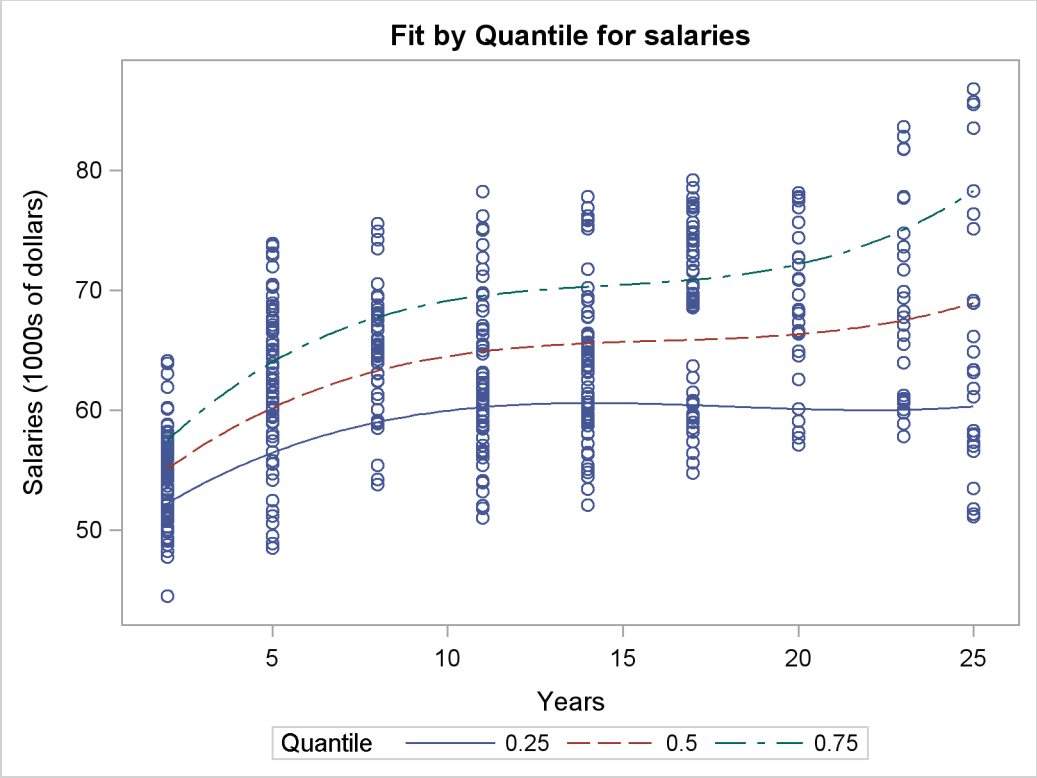
Parameter	DF	Estimate
Intercept	1	48.2509
years	1	2.2234
years*years	1	-0.1292
years*years*years	1	0.0024

Output 72.5.3 *continued*

```
                    Parameter Estimates

                Parameter         DF Estimate

                Intercept          1  50.2512
                years              1   2.7173
                years*years        1  -0.1632
                years*years*years  1   0.0034

                    Parameter Estimates

                Parameter         DF Estimate

                Intercept          1  51.0298
                years              1   3.6513
                years*years        1  -0.2390
                years*years*years  1   0.0055
```

Output 72.5.4 displays the three cubic percentile curves.

Output 72.5.4 Salary with Years as Professor: Regression Quantiles



The three curves show that salary dispersion increases gradually with length of service. After 15 years, a salary over $70,000 is relatively high, while a salary less than $60,000 is relatively low. Note that percentile curves of this type are useful in medical science as reference curves; see Yu, Lu, and Stabder (2003).

References

Abreveya, J. (2001), "The Effects of Demographics and Maternal Behavior on the Distribution of Birth Outcomes," *Journal of Economics*, 26, 247–257.

Barro, R. and Lee, J. W. (1994), "Data Set for a Panel of 138 Countries," discussion paper, National Bureau of Econometric Research. <http://www.nber.org/pub/barro.lee>.

Barrodale, I. and Roberts, F. D. K. (1973), "An Improved Algorithm for Discrete l_1 Linear Approximation," *SIAM Journal of Numerical Analysis,* 10, 839–848.

Bassett, G. W. and Koenker, R. (1982), "An Empirical Quantile Function for Linear Models with iid Errors," *Journal of the American Statistical Association*, 77, 401–415.

Cade, B. S. and Noon B. R. (2003), "A Gentle Introduction to Quantile Regression for Ecologists," *Frontiers in Ecology and the Environment*, 1(8), 412–420.

Chen, C. (2004), "An Adaptive Algorithm for Quantile Regression," *Theory and Applications of Recent Robust Methods*, ed. M. Hubert, G. Pison, A. Struyf, and S. Van Aelst, Series: Statistics for Industry and Technology, Birkhauser, Basel, 39–48.

Chen, C. (2005), "Growth Charts of Body Mass Index (BMI) with Quantile Regression," *Proceedings of 2005 International Conference on Algorithmic Mathematics and Computer Science*, June 20–23, 2005, Las Vegas, Nevada.

Chen, C. (2007), "A Finite Smoothing Algorithm for Quantile Regression," *Journal of Computational and Graphical Statistics*, 16, 136–164.

Chock, D. P., Winkler, S. L., and Chen, C. (2000), "A Study of the Association between Daily Mortality and Ambient Air Pollutant Concentrations in Pittsburgh, Pennsylvania," *Journal of the Air and Waste Management Association*, 50, 1481–1500.

Dunham, J. B., Cade, B. S., and Terrell J. W. (2002), "Influences of Spatial and Temporal Variation on Fish-Habitat Relationships Defined by Regression Quantiles," *Transactions of the American Fisheries Society*, 131, 86–98.

Gutenbrunner, C. and Jureckova, J. (1992), "Regression Rank Scores and Regression Quantiles." *Annals of Statistics*, 20, 305–330.

Hastie, T., Tibshirani, R., and Friedman, J. (2001), *The Elements of Statistical Learning,* New York: Springer-Verlag.

He, X. and Hu, F. (2002), "Markov Chain Marginal Bootstrap," *Journal of the American Statistical Association*, 97, 783–795.

Huber, P. J. (1981), *Robust Statistics.* New York: John Wiley & Sons.

Karmarkar, N. (1984), "A New Polynomial-Time Algorithm for Linear Programming," *Combinatorica*, 4, 373–395.

Koenker, R. (1994), "Confidence Intervals for Regression Quantiles," *Asymptotic Statistics*, eds. P. Mandl and M. Huskova, 349–359, New York: Springer-Verlag.

Koenker, R. and Bassett, G. W. (1978), "Regression Quantiles," *Econometrica*, 46, 33–50.

Koenker, R. and Bassett, G. W. (1982), "Robust Tests for Heteroscedasticity Based on Regression Quantiles," *Econometrica*, 50, 43–61.

Koenker, R. and d'Orey, V. (1994), "Remark AS R92: A Remark on Algorithm AS 229: Computing Dual Regression Quantiles and Regression Rank Scores," *Applied Statistics*, 43, 410–414.

Koenker, R. and Hallock, K. (2001), "Quantile Regression: An Introduction," *Journal of Economic Perspectives*, 15, 143–156.

Koenker, R. and Machado, A. F. (1999), "Goodness of Fit and Related Inference Processes for Quantile Regression," *Journal of the American Statistical Association*, 94, 1296–1310.

Koenker, R. and Zhao, Q. (1994), "L-Estimation for Linear Heteroscedastic Models," *Journal of Nonparametric Statistics*, 3, 223–235.

Kuczmarski, R. J., Ogden, C. L., Guo, S. S., et al. (2002), "2000 CDC Growth Charts for the United States: Methods and Development," *Vital Health Stat.*, 11, 246, 1–190.

Lustig, I. J., Marsden, R. E., and Shanno, D. F. (1992), "On Implementing Mehrotra's Predictor-Corrector Interior-Point Method for Linear Programming," *SIAM Journal on Optimization*, 2, 435–449.

Madsen, K. and Nielsen, H. B. (1993), "A Finite Smoothing Algorithm for Linear L_1 Estimation," *SIAM Journal on Optimization*, 3, 223–235.

Parzen, M. I., Wei, L. J., and Ying, Z. (1994), "A Resampling Method Based on Pivotal Estimating Functions," *Biometrika*, 81, 341–350.

Portnoy, S. and Koenker, R. (1997), "The Gaussian Hare and the Laplacian Tortoise: Computation of Squared-Error vs. Absolute-Error Estimators," *Statistical Science*, 12, 279–300.

Roos, C., Terlaky, T., and Vial, J.-Ph. (1997), "Theory and Algorithms for Linear Optimization," Chichester, England: John Wiley & Sons.

Rousseeuw, P. J. and Van Driessen, K. (1999), "A Fast Algorithm for the Minimum Covariance Determinant Estimator," *Technometrics*, 41, 212–223.

Yu, K., Lu, Z., and Stabder, J. (2003), "Quantile Regression: Application and Current Research Areas," *The Statistician*, 52, 331–350.

Chapter 73

The REG Procedure

Contents

Overview: REG Procedure **5434**
Getting Started: REG Procedure **5436**
Simple Linear Regression 5436
Polynomial Regression 5440
Using PROC REG Interactively 5450
Syntax: REG Procedure **5451**
PROC REG Statement 5453
ADD Statement 5467
BY Statement 5467
DELETE Statement 5468
FREQ Statement 5468
ID Statement 5468
MODEL Statement 5469
MTEST Statement 5480
OUTPUT Statement 5482
PAINT Statement 5484
PLOT Statement 5487
PRINT Statement 5499
REFIT Statement 5500
RESTRICT Statement 5500
REWEIGHT Statement 5502
TEST Statement 5506
VAR Statement 5507
WEIGHT Statement 5507
Details: REG Procedure **5507**
Missing Values 5507
Input Data Sets 5508
Output Data Sets 5512
Interactive Analysis 5519
Model-Selection Methods 5523
Criteria Used in Model-Selection Methods 5526
Limitations in Model-Selection Methods 5527
Parameter Estimates and Associated Statistics 5527
Predicted and Residual Values 5531

Line Printer Scatter Plot Features 5534
Traditional Graphics 5547
Models of Less Than Full Rank 5553
Collinearity Diagnostics 5555
Model Fit and Diagnostic Statistics 5557
Influence Statistics 5559
Reweighting Observations in an Analysis 5569
Testing for Heteroscedasticity 5575
Testing for Lack of Fit 5576
Multivariate Tests 5577
Autocorrelation in Time Series Data 5581
Computations for Ridge Regression and IPC Analysis 5583
Construction of Q-Q and P-P Plots 5583
Computational Methods 5584
Computer Resources in Regression Analysis 5584
Displayed Output 5584
ODS Table Names 5587
ODS Graphics 5589
Examples: REG Procedure **5592**
Example 73.1: Modeling Salaries of Major League Baseball Players 5592
Example 73.2: Aerobic Fitness Prediction 5608
Example 73.3: Predicting Weight by Height and Age 5627
Example 73.4: Regression with Quantitative and Qualitative Variables . . . 5633
Example 73.5: Ridge Regression for Acetylene Data 5638
Example 73.6: Chemical Reaction Response 5642
References **5644**

Overview: REG Procedure

The REG procedure is one of many regression procedures in the SAS System. It is a general-purpose procedure for regression, while other SAS regression procedures provide more specialized applications.

Other SAS/STAT procedures that perform at least one type of regression analysis are the CATMOD, GENMOD, GLM, LOGISTIC, MIXED, NLIN, ORTHOREG, PROBIT, RSREG, and TRANSREG procedures. SAS/ETS procedures are specialized for applications in time series or simultaneous systems. These other SAS/STAT regression procedures are summarized in Chapter 4, "Introduction to Regression Procedures," which also contains an overview of regression techniques and defines many of the statistics computed by PROC REG and other regression procedures.

PROC REG provides the following capabilities:

- multiple MODEL statements
- nine model-selection methods
- interactive changes both in the model and the data used to fit the model
- linear equality restrictions on parameters
- tests of linear hypotheses and multivariate hypotheses
- collinearity diagnostics
- predicted values, residuals, studentized residuals, confidence limits, and influence statistics
- correlation or crossproduct input
- requested statistics available for output through output data sets
- ODS Graphics is now available. For more information, see the section "ODS Graphics" on page 5589. These plots are available in addition to the line printer and the traditional graphics currently available in PROC REG.

Nine model-selection methods are available in PROC REG. In the simplest method, PROC REG fits the complete model that you specify. The other eight methods involve various ways of including or excluding variables from the model. You specify these methods with the SELECTION= option in the MODEL statement.

The methods are identified in the following list and are explained in detail in the section "Model-Selection Methods" on page 5523.

NONE	no model selection. This is the default. The complete model specified in the MODEL statement is fit to the data.
FORWARD	forward selection. This method starts with no variables in the model and adds variables.
BACKWARD	backward elimination. This method starts with all variables in the model and deletes variables.
STEPWISE	stepwise regression. This is similar to the FORWARD method except that variables already in the model do not necessarily stay there.
MAXR	forward selection to fit the best one-variable model, the best two-variable model, and so on. Variables are switched so that R^2 is maximized.
MINR	similar to the MAXR method, except that variables are switched so that the increase in R^2 from adding a variable to the model is minimized.
RSQUARE	finds a specified number of models with the highest R^2 in a range of model sizes.
ADJRSQ	finds a specified number of models with the highest adjusted R^2 in a range of model sizes.
CP	finds a specified number of models with the lowest C_p in a range of model sizes.

Getting Started: REG Procedure

Simple Linear Regression

Suppose that a response variable Y can be predicted by a linear function of a regressor variable X. You can estimate β_0, the intercept, and β_1, the slope, in

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$

for the observations $i = 1, 2, \ldots, n$. Fitting this model with the REG procedure requires only the following MODEL statement, where y is the outcome variable and x is the regressor variable.

```
proc reg;
   model y=x;
run;
```

For example, you might use regression analysis to find out how well you can predict a child's weight if you know that child's height. The following data are from a study of nineteen children. Height and weight are measured for each child.

```
title 'Simple Linear Regression';
data Class;
   input Name $ Height Weight Age @@;
   datalines;
Alfred  69.0 112.5 14  Alice   56.5  84.0 13  Barbara 65.3  98.0 13
Carol   62.8 102.5 14  Henry   63.5 102.5 14  James   57.3  83.0 12
Jane    59.8  84.5 12  Janet   62.5 112.5 15  Jeffrey 62.5  84.0 13
John    59.0  99.5 12  Joyce   51.3  50.5 11  Judy    64.3  90.0 14
Louise  56.3  77.0 12  Mary    66.5 112.0 15  Philip  72.0 150.0 16
Robert  64.8 128.0 12  Ronald  67.0 133.0 15  Thomas  57.5  85.0 11
William 66.5 112.0 15
;
```

The equation of interest is

$$\text{Weight} = \beta_0 + \beta_1 \text{Height} + \epsilon$$

The variable Weight is the response or dependent variable in this equation, and β_0 and β_1 are the unknown parameters to be estimated. The variable Height is the regressor or independent variable, and ϵ is the unknown error. The following commands invoke the REG procedure and fit this model to the data.

```
ods graphics on;

proc reg;
   model Weight = Height;
run;

ods graphics off;
```

Figure 73.1 includes some information concerning model fit.

The F statistic for the overall model is highly significant (F=57.076, p<0.0001), indicating that the model explains a significant portion of the variation in the data.

The degrees of freedom can be used in checking accuracy of the data and model. The model degrees of freedom are one less than the number of parameters to be estimated. This model estimates two parameters, β_0 and β_1; thus, the degrees of freedom should be $2 - 1 = 1$. The corrected total degrees of freedom are always one less than the total number of observations in the data set, in this case $19 - 1 = 18$.

Several simple statistics follow the ANOVA table. The Root MSE is an estimate of the standard deviation of the error term. The coefficient of variation, or Coeff Var, is a unitless expression of the variation in the data. The R-square and Adj R-square are two statistics used in assessing the fit of the model; values close to 1 indicate a better fit. The R-square of 0.77 indicates that Height accounts for 77% of the variation in Weight.

Figure 73.1 ANOVA Table

Simple Linear Regression

The REG Procedure
Model: MODEL1
Dependent Variable: Weight

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	7193.24912	7193.24912	57.08	<.0001
Error	17	2142.48772	126.02869		
Corrected Total	18	9335.73684			

Root MSE	11.22625	R-Square	0.7705
Dependent Mean	100.02632	Adj R-Sq	0.7570
Coeff Var	11.22330		

The "Parameter Estimates" table in Figure 73.2 contains the estimates of β_0 and β_1. The table also contains the t statistics and the corresponding p-values for testing whether each parameter is significantly different from zero. The p-values ($t = -4.43$, $p = 0.0004$ and $t = 7.55$, $p < 0.0001$) indicate that the intercept and Height parameter estimates, respectively, are highly significant.

From the parameter estimates, the fitted model is

$$\text{Weight} = -143.0 + 3.9 \times \text{Height}$$

Figure 73.2 Parameter Estimates

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-143.02692	32.27459	-4.43	0.0004
Height	1	3.89903	0.51609	7.55	<.0001

If you enable ODS Graphics with an ODS GRAPHICS statement, then PROC REG produces a variety of useful plots. Figure 73.3 shows a plot of the residuals versus the regressor and Figure 73.4 shows a panel of diagnostic plots.

Figure 73.3 Residuals vs. Regressor

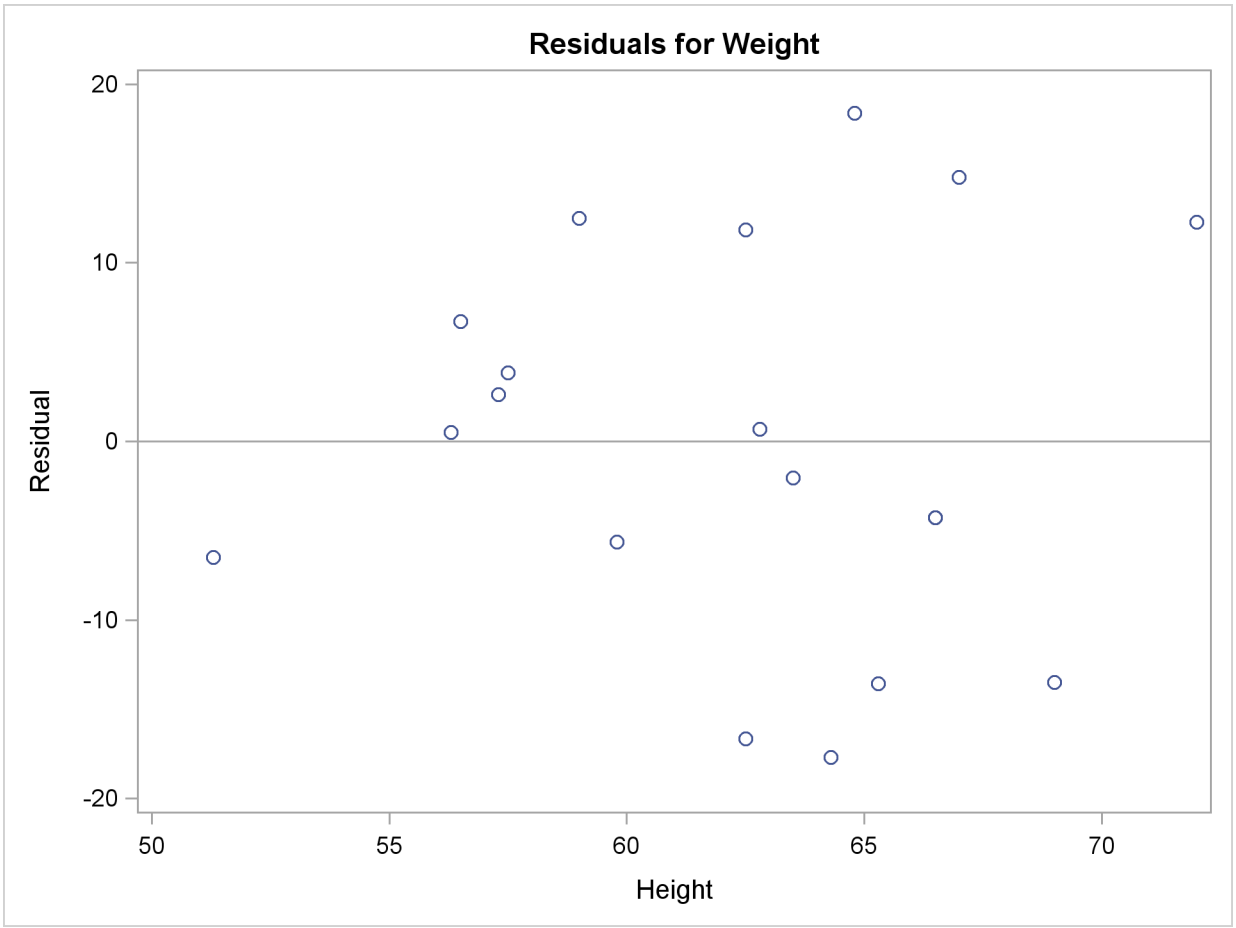
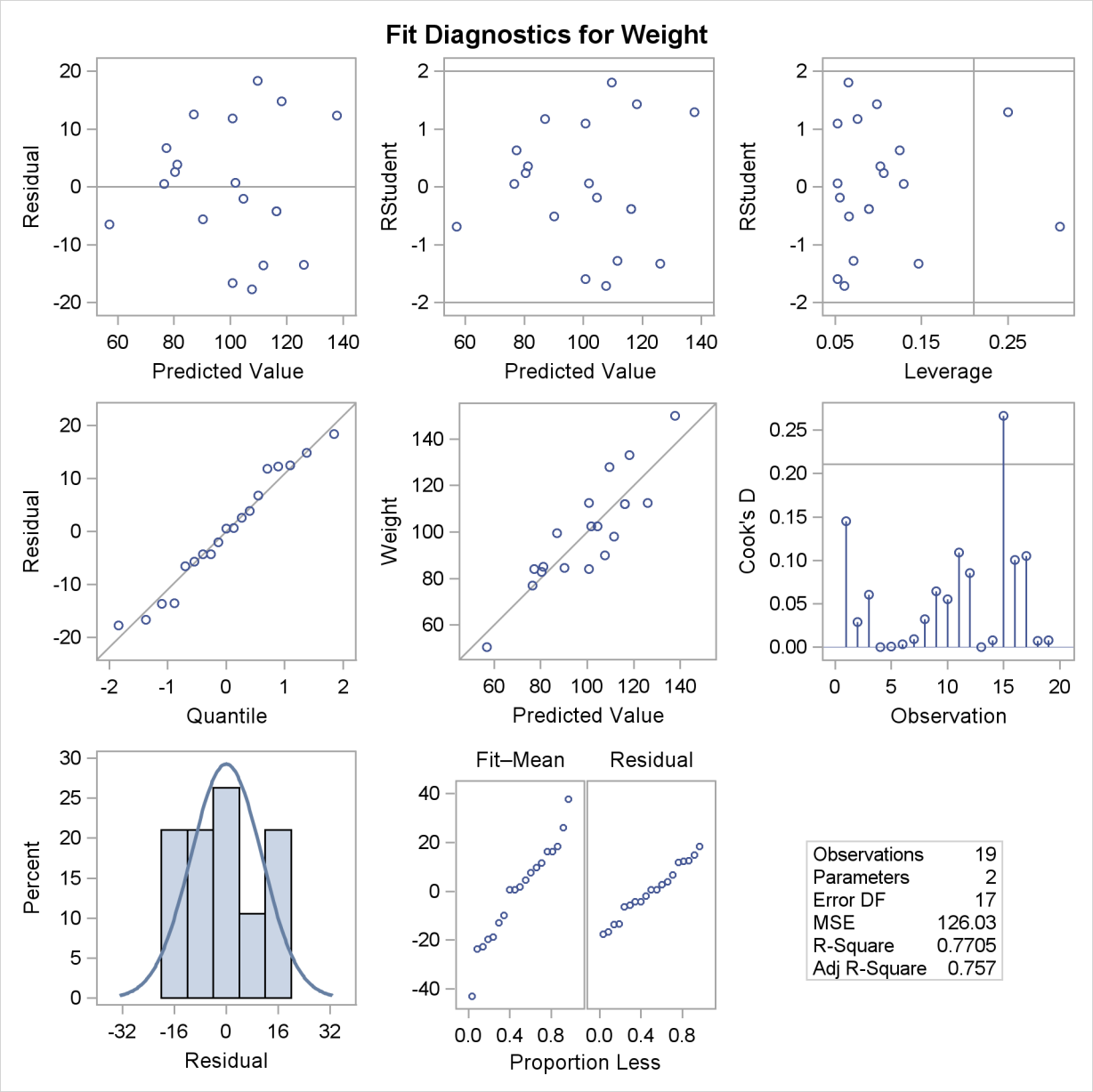


Figure 73.4 Fit Diagnostics



A trend in the residuals would indicate nonconstant variance in the data. The plot of residuals by predicted values in the upper-left corner of the diagnostics panel in Figure 73.4 might indicate a slight trend in the residuals; they appear to increase slightly as the predicted values increase. A fan-shaped trend might indicate the need for a variance-stabilizing transformation. A curved trend (such as a semicircle) might indicate the need for a quadratic term in the model. Since these residuals have no apparent trend, the analysis is considered to be acceptable.

Polynomial Regression

Consider a response variable Y that can be predicted by a polynomial function of a regressor variable X. You can estimate β_0, the intercept; β_1, the slope due to X; and β_2, the slope due to X^2, in

$$Y_i = \beta_0 + \beta_1 X_i + \beta_2 X_i^2 + \epsilon_i$$

for the observations $i = 1, 2, \ldots, n$.

Consider the following example on population growth trends. The population of the United States from 1790 to 2000 is fit to linear and quadratic functions of time. Note that the quadratic term, YearSq, is created in the DATA step; this is done since polynomial effects such as Year*Year cannot be specified in the MODEL statement in PROC REG. The data are as follows:

```
data USPopulation;
   input Population @@;
   retain Year 1780;
   Year       = Year+10;
   YearSq     = Year*Year;
   Population = Population/1000;
datalines;
3929 5308 7239 9638 12866 17069 23191 31443 39818 50155
62947 75994 91972 105710 122775 131669 151325 179323 203211
226542 248710 281422
;

ods graphics on;

proc reg data=USPopulation plots=ResidualByPredicted;
   var YearSq;
   model Population=Year / r clm cli;
run;
```

The DATA option ensures that the procedure uses the intended data set. Any variable that you might add to the model but that is not included in the first MODEL statement must appear in the VAR statement.

The "Analysis of Variance" and "Parameter Estimates" tables are displayed in Figure 73.5.

Figure 73.5 ANOVA Table and Parameter Estimates

The REG Procedure
Model: MODEL1
Dependent Variable: Population

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	146869	146869	228.92	<.0001
Error	20	12832	641.58160		
Corrected Total	21	159700			

Root MSE	25.32946	R-Square	0.9197
Dependent Mean	94.64800	Adj R-Sq	0.9156
Coeff Var	26.76175		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-2345.85498	161.39279	-14.54	<.0001
Year	1	1.28786	0.08512	15.13	<.0001

The Model F statistic is significant (F=228.92, p<0.0001), indicating that the model accounts for a significant portion of variation in the data. The R-square indicates that the model accounts for 92% of the variation in population growth. The fitted equation for this model is

$$\text{Population} = -2345.85 + 1.29 \times \text{Year}$$

In the MODEL statement, three options are specified: R requests a residual analysis to be performed, CLI requests 95% confidence limits for an individual value, and CLM requests these limits for the expected value of the dependent variable. You can request specific $100(1-\alpha)\%$ limits with the ALPHA= option in the PROC REG or MODEL statement.

Figure 73.6 shows the "Output Statistics" table. The residual, its standard error, and the studentized residuals are displayed for each observation. The studentized residual is the residual divided by its standard error. The magnitude of each studentized residual is shown in a print plot. Studentized residuals follow a t distribution and can be used to identify outlying or extreme observations. Asterisks (*) extending beyond the dashed lines indicate that the residual is more than three standard errors from zero. Many observations having absolute studentized residuals greater than two might indicate an inadequate model. Cook's D is a measure of the change in the predicted values upon deletion of that observation from the data set; hence, it measures the influence of the observation on the estimated regression coefficients.

Figure 73.6 Output Statistics

The REG Procedure
Model: MODEL1
Dependent Variable: Population

Output Statistics

Obs	Dependent Variable	Predicted Value	Std Error Mean Predict	95% CL Mean		95% CL Predict		Residual
1	3.9290	-40.5778	10.4424	-62.3602	-18.7953	-97.7280	16.5725	44.5068
2	5.3080	-27.6991	9.7238	-47.9826	-7.4156	-84.2950	28.8968	33.0071
3	7.2390	-14.8205	9.0283	-33.6533	4.0123	-70.9128	41.2719	22.0595
4	9.6380	-1.9418	8.3617	-19.3841	15.5004	-57.5827	53.6991	11.5798
5	12.8660	10.9368	7.7314	-5.1906	27.0643	-44.3060	66.1797	1.9292
6	17.0690	23.8155	7.1470	8.9070	38.7239	-31.0839	78.7148	-6.7465
7	23.1910	36.6941	6.6208	22.8834	50.5048	-17.9174	91.3056	-13.5031
8	31.4430	49.5727	6.1675	36.7075	62.4380	-4.8073	103.9528	-18.1297
9	39.8180	62.4514	5.8044	50.3436	74.5592	8.2455	116.6573	-22.6334
10	50.1550	75.3300	5.5491	63.7547	86.9053	21.2406	129.4195	-25.1750
11	62.9470	88.2087	5.4170	76.9090	99.5084	34.1776	142.2398	-25.2617
12	75.9940	101.0873	5.4170	89.7876	112.3870	47.0562	155.1184	-25.0933
13	91.9720	113.9660	5.5491	102.3907	125.5413	59.8765	168.0554	-21.9940
14	105.7100	126.8446	5.8044	114.7368	138.9524	72.6387	181.0505	-21.1346
15	122.7750	139.7233	6.1675	126.8580	152.5885	85.3432	194.1033	-16.9483
16	131.6690	152.6019	6.6208	138.7912	166.4126	97.9904	207.2134	-20.9329
17	151.3250	165.4805	7.1470	150.5721	180.3890	110.5812	220.3799	-14.1555
18	179.3230	178.3592	7.7314	162.2317	194.4866	123.1163	233.6020	0.9638
19	203.2110	191.2378	8.3617	173.7956	208.6801	135.5969	246.8787	11.9732
20	226.5420	204.1165	9.0283	185.2837	222.9493	148.0241	260.2088	22.4255
21	248.7100	216.9951	9.7238	196.7116	237.2786	160.3992	273.5910	31.7149
22	281.4220	229.8738	10.4424	208.0913	251.6562	172.7235	287.0240	51.5482

Output Statistics

Obs	Std Error Residual	Student Residual	-2-1 0 1 2	Cook's D
1	23.077	1.929	|***|	0.381
2	23.389	1.411	|**|	0.172
3	23.666	0.932	|*|	0.063
4	23.909	0.484	|	0.014
5	24.121	0.0800	|	0.000
6	24.300	-0.278	|	0.003
7	24.449	-0.552	*|	0.011
8	24.567	-0.738	*|	0.017
9	24.655	-0.918	*|	0.023
10	24.714	-1.019	**|	0.026
11	24.743	-1.021	**|	0.025
12	24.743	-1.014	**|	0.025
13	24.714	-0.890	*|	0.020
14	24.655	-0.857	*|	0.020
15	24.567	-0.690	*|	0.015
16	24.449	-0.856	*|	0.027
17	24.300	-0.583	*|	0.015
18	24.121	0.0400	|	0.000
19	23.909	0.501	|*	0.015
20	23.666	0.948	|*	0.065
21	23.389	1.356	|**	0.159
22	23.077	2.234	|****	0.511

Figure 73.7 shows the residual statistics table. A fairly close agreement between the PRESS statistic (see Table 73.8) and the Sum of Squared Residuals indicates that the MSE is a reasonable measure of the predictive accuracy of the fitted model (Neter, Wasserman, and Kutner 1990).

Figure 73.7 Residual Statistics

Sum of Residuals	0
Sum of Squared Residuals	12832
Predicted Residual SS (PRESS)	16662

Graphical representations are very helpful in interporting the information in the "Output Statistics" table. When you enable ODS Graphics, the REG procedure produces a default set of diagnostic plots that are appropriate for the requested analysis.

Figure 73.8 displays a panel of diagnostics plots. These diagnostics indicate an inadequate model:

- The plots of residual and studentized residual versus predicted value show a clear quadratic pattern.
- The plot of studentized residual versus leverage seems to indicate that there are two outlying data points. However, the plot of Cook's D distance versus observation number reveals that these two points are just the data points for the endpoint years 1790 and 2000. These points show up as apparent outliers because the departure of the linear model from the underlying quadratic behavior in the data shows up most strongly at these endpoints.
- The normal quantile plot of the residuals and the residual histogram are not consistent with the assumption of Gaussian errors. This occurs as the residuals themselves still contain the quadratic behavior that is not captured by the linear model.
- The plot of the dependent variable versus the predicted value exhibits a quadratic form around the 45-degree line that represents a perfect fit.
- The "Residual-Fit" (or RF) plot consisting of side-by-side quantile plots of the centered fit and the residuals shows that the spread in the residuals is no greater than the spread in the centered fit. For inappropriate models, the spread of the residuals in such a plot is often greater than the spread of the centered fit. In this case, the RF plot shows that the linear model does indeed capture the increasing trend in the data, and hence accounts for much of the variation in the response.

Figure 73.8 Diagnostics Panel

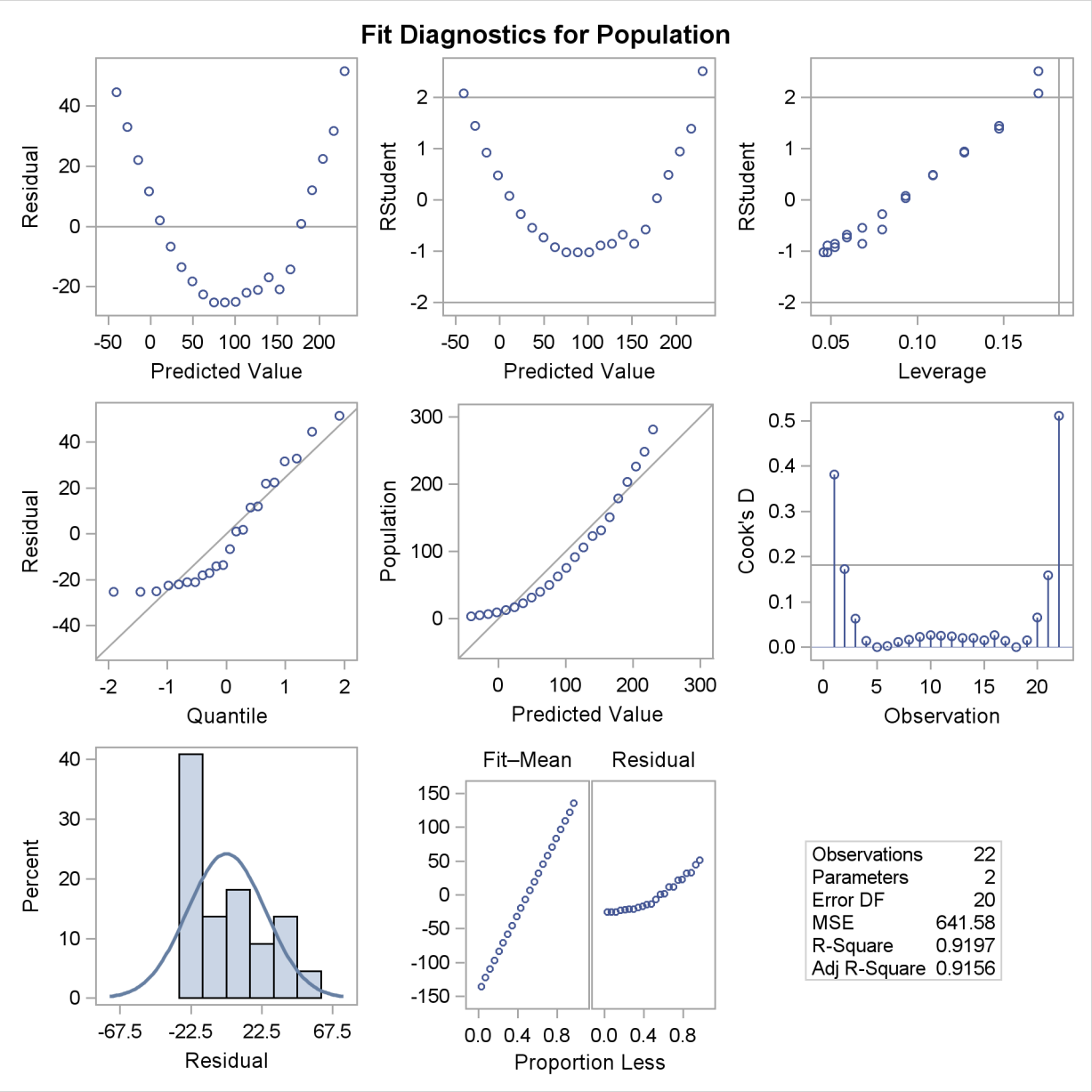


Figure 73.9 shows a plot of residuals versus Year. Again you can see the quadratic pattern that strongly indicates that a quadratic term should be added to the model.

Figure 73.9 Residual Plot

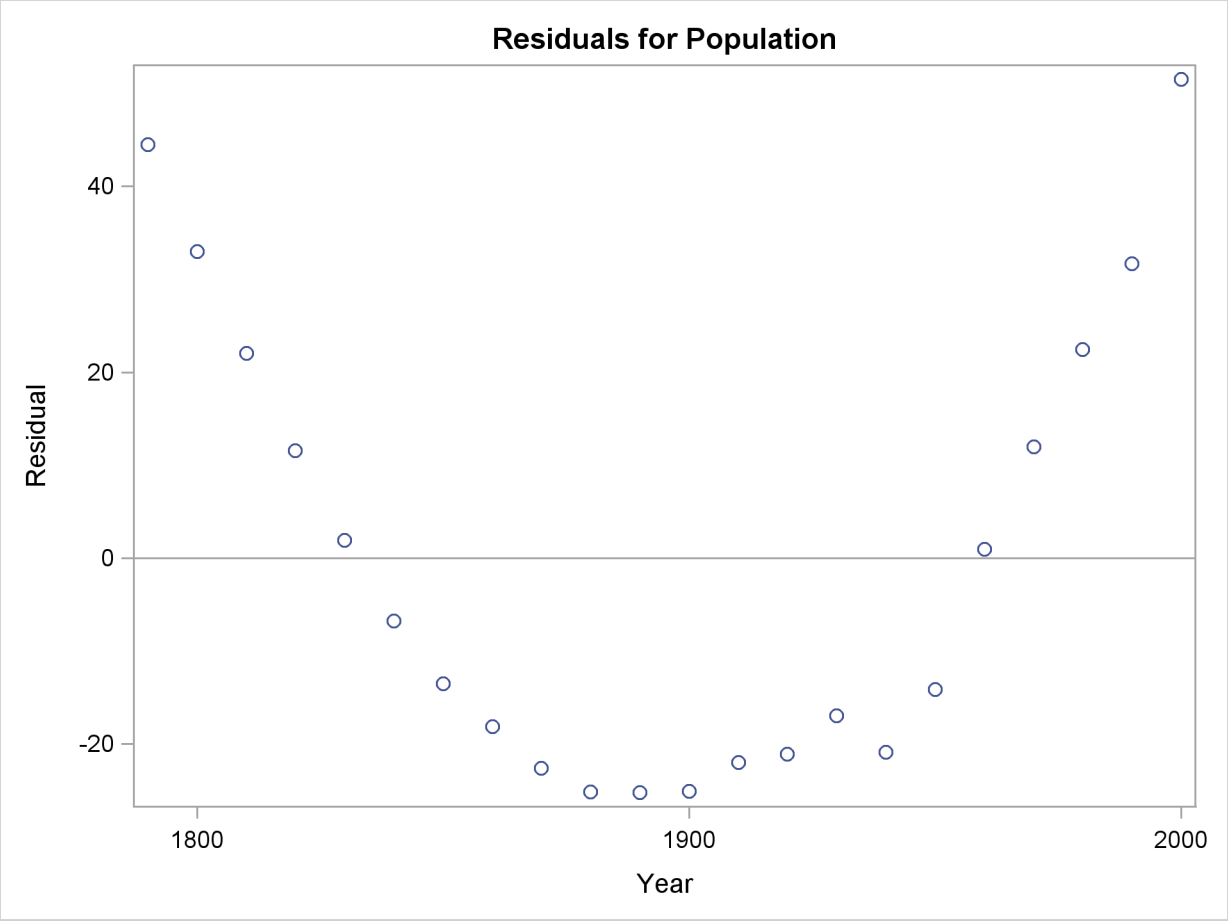
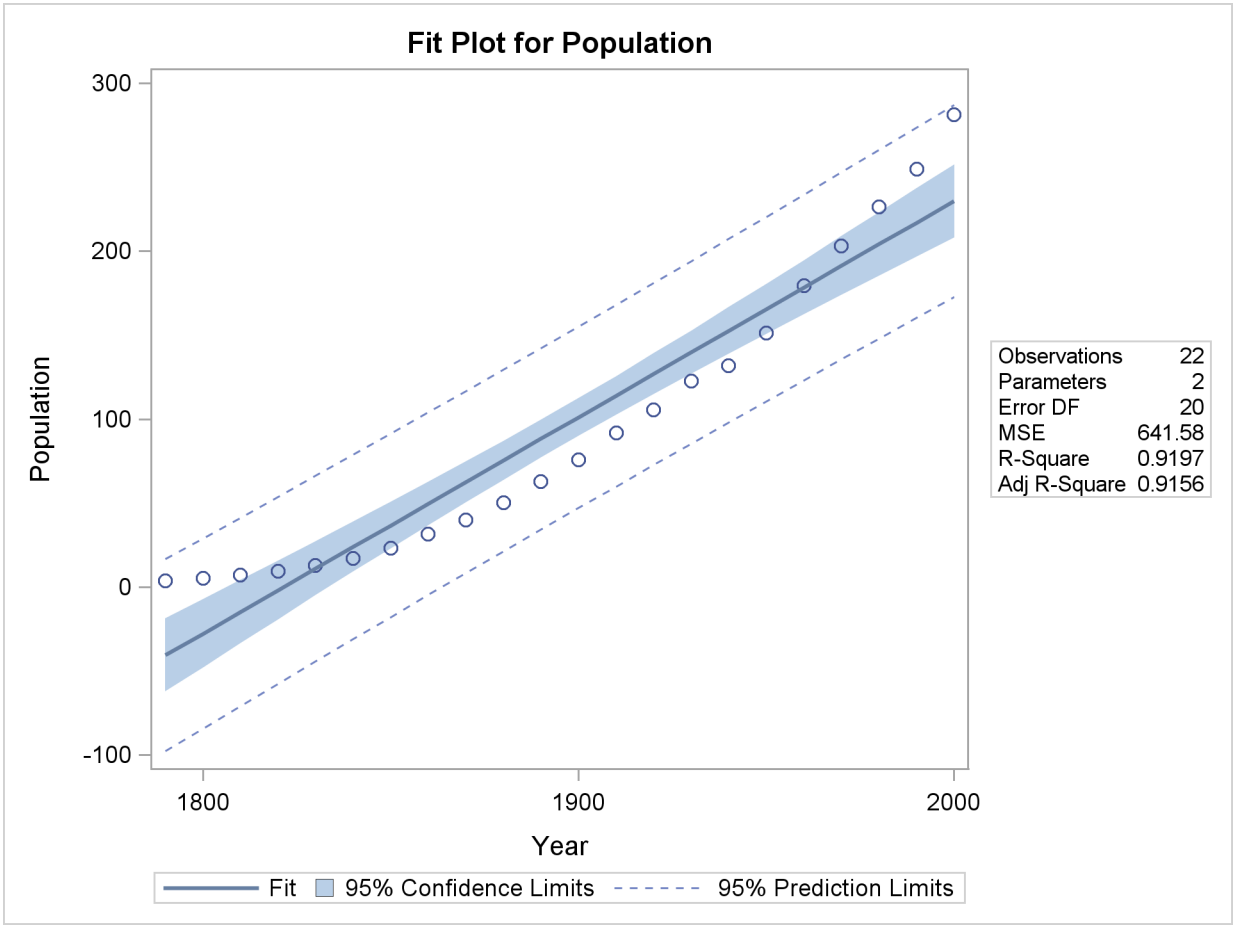


Figure 73.10 shows the "FitPlot" consisting of a scatter plot of the data overlaid with the regression line, and 95% confidence and prediction limits. Note that this plot also indicates that the model fails to capture the quadratic nature of the data. This plot is produced for models containing a single regressor. You can use the ALPHA= option in the model statement to change the significance level of the confidence band and prediction limits.

Figure 73.10 Fit Plot



These default plots provide strong evidence that the Yearsq needs to be added to the model. You can use the interactive feature of PROC REG to do this by specifying the following statements:

```
add YearSq;
print;
run;
```

The ADD statement requests that YearSq be added to the model, and the PRINT command causes the model to be refit and displays the ANOVA and parameter estimates for the new model. The print statement also produces updated ODS graphical displays.

Figure 73.11 displays the ANOVA table and parameter estimates for the new model.

Figure 73.11 ANOVA Table and Parameter Estimates

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	159529	79765	8864.19	<.0001
Error	19	170.97193	8.99852		
Corrected Total	21	159700			

Root MSE	2.99975	R-Square	0.9989
Dependent Mean	94.64800	Adj R-Sq	0.9988
Coeff Var	3.16938		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	21631	639.50181	33.82	<.0001
Year	1	-24.04581	0.67547	-35.60	<.0001
YearSq	1	0.00668	0.00017820	37.51	<.0001

The overall F statistic is still significant (F=8864.19, p<0.0001). The R-square has increased from 0.9197 to 0.9989, indicating that the model now accounts for 99.9% of the variation in Population. All effects are significant with p<0.0001 for each effect in the model.

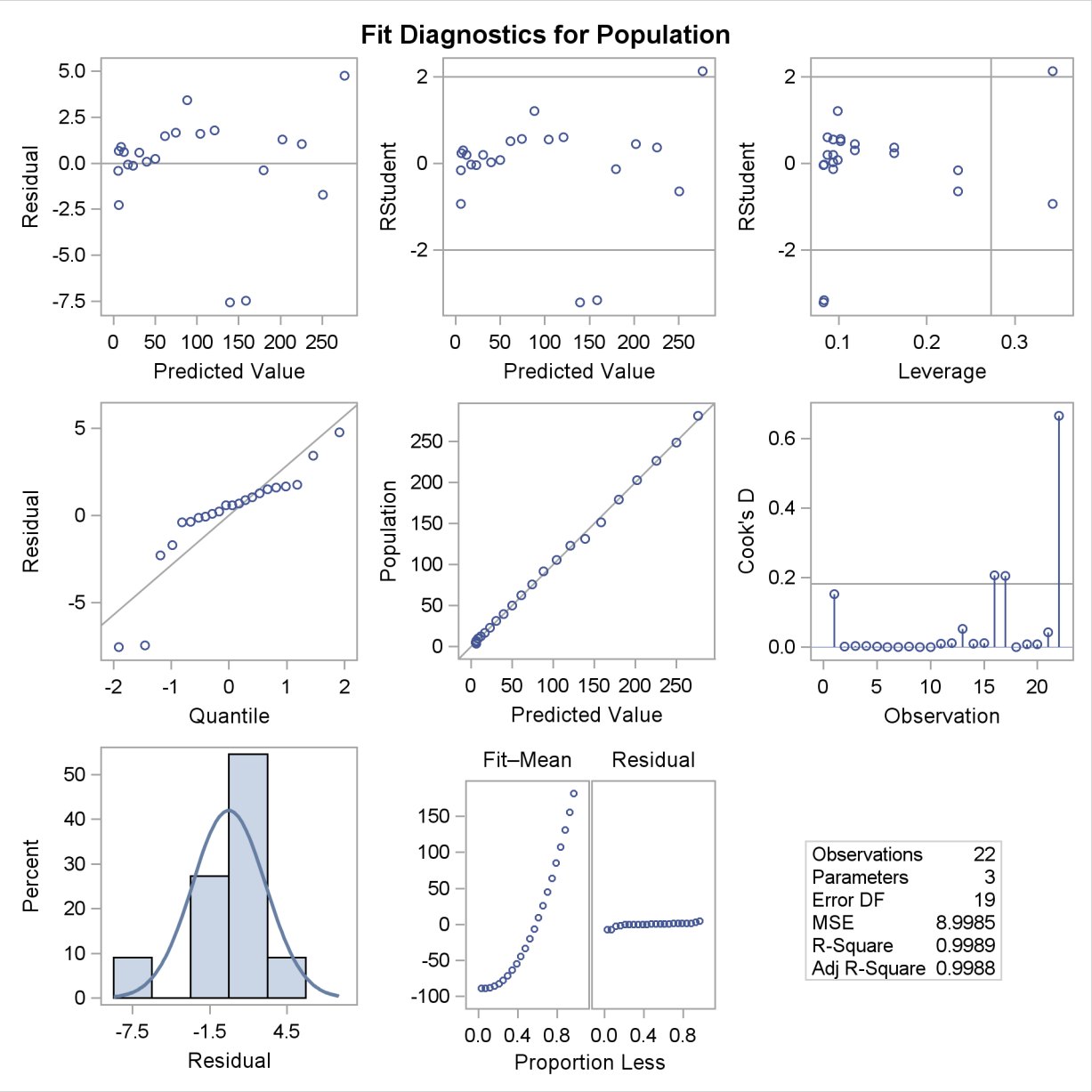
The fitted equation is now

$$\text{Population} = 21631 - 24.046 \times \text{Year} + 0.0067 \times \text{Yearsq}$$

Figure 73.12 show the panel of diagnostics for this quadratic polynomial model. These diagnostics indicate that this model is considerably more successful than the corresponding linear model:

- The plots of residuals and studentized residuals versus predicted values exhibit no obvious patterns.
- The points on the plot of the dependent variable versus the predicted values lie along a 45-degree line, indicating that the model successfully predicts the behavior of the dependent variable.
- The plot of studentized residual versus leverage shows that the years 1790 and 2000 are leverage points with 2000 showing up as an outlier. This is confirmed by the plot of Cook's D distance versus observation number. This suggests that while the quadratic model fits the current data well, the model might not be quite so successful over a wider range of data. You might want to investigate whether the population trend over the last couple of decades is growing slightly faster than quadratically.

Figure 73.12 Diagnostics Panel

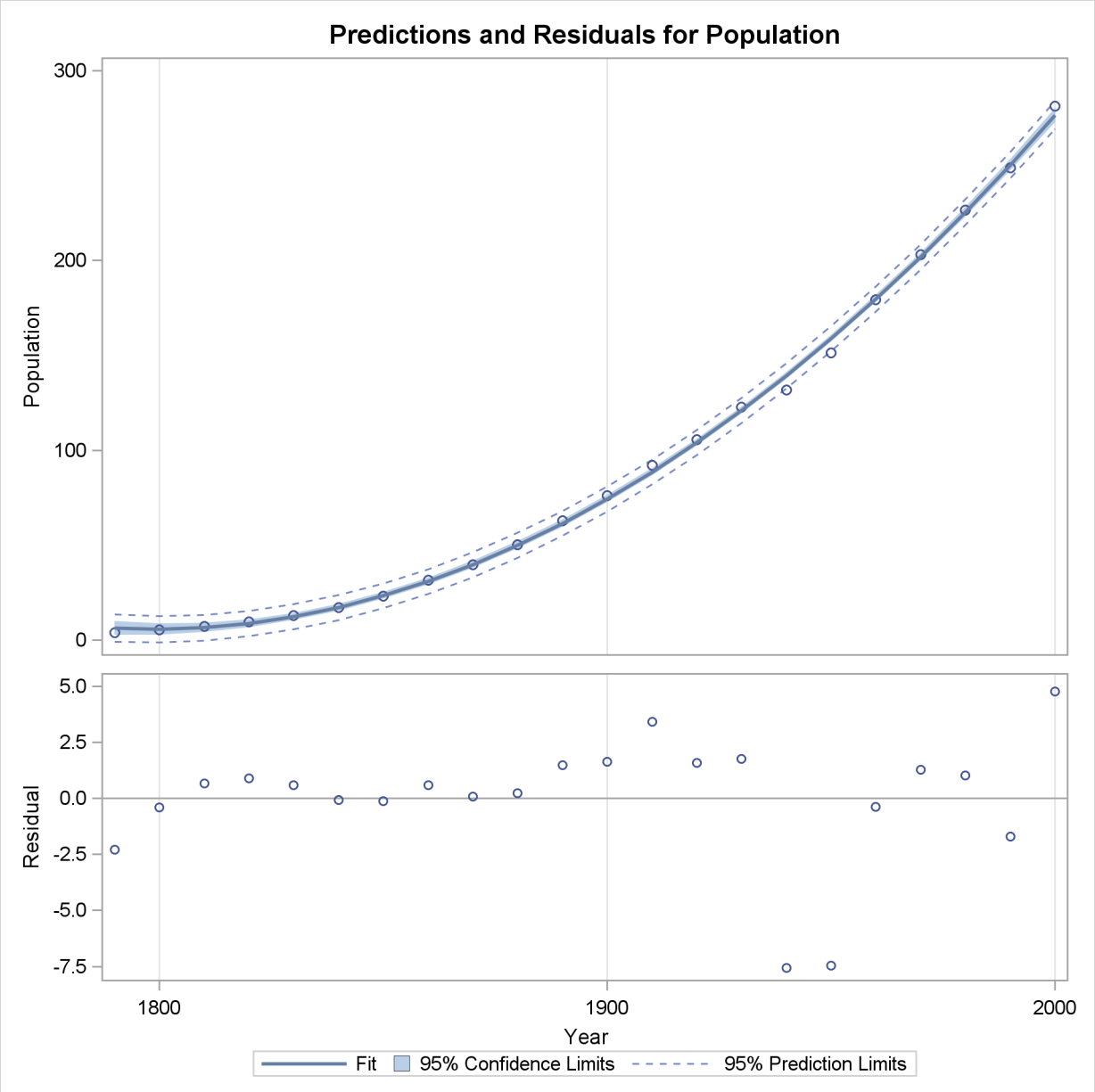


When a model contains more than one regressor, PROC REG does not produce a fit plot. However, when all the regressors in the model are functions of a single variable, it is appropriate to plot predictions and residuals as a function of that variable. You request such plots by using the PLOTS=PREDICTIONS option in the PROC REG statement, as the following code illustrates:

```
proc reg data=USPopulation plots=predictions(X=Year);
   model Population=Year Yearsq;
quit;

ods graphics off;
```

Figure 73.13 shows the data, predictions, and residuals by Year. These plots confirm that the quadratic polynomial model successfully model the growth in U.S. population between the years 1780 and 2000.

Figure 73.13 Predictions and Residuals by Year



To complete an analysis of these data, you might want to examine influence statistics and, since the data are essentially time series data, examine the Durbin-Watson statistic.

Using PROC REG Interactively

The REG procedure can be used interactively. After you specify a model with a MODEL statement and run PROC REG with a RUN statement, a variety of statements can be executed without reinvoking PROC REG.

The section "Interactive Analysis" on page 5519 describes which statements can be used interactively. These interactive statements can be executed singly or in groups by following the single statement or group of statements with a RUN statement. Note that the MODEL statement can be repeated. This is an important difference from the GLM procedure, which supports only one MODEL statement.

If you use PROC REG interactively, you can end the REG procedure with a DATA step, another PROC step, an ENDSAS statement, or a QUIT statement. The syntax of the QUIT statement is

```
quit;
```

When you are using PROC REG interactively, additional RUN statements do not end PROC REG but tell the procedure to execute additional statements.

When a BY statement is used with PROC REG, interactive processing is not possible; that is, once the first RUN statement is encountered, processing proceeds for each BY group in the data set, and no further statements are accepted by the procedure.

When you use PROC REG interactively, you can fit a model, perform diagnostics, and then refit the model and perform diagnostics on the refitted model. Most of the interactive statements implicitly refit the model; for example, if you use the ADD statement to add a variable to the model, the regression equation is automatically recomputed. The two exceptions to this automatic recomputing are the PAINT and REWEIGHT statements. These two statements do not cause the model to be refitted. To refit the model, you can follow these statements either with a REFIT statement, which causes the model to be explicitly recomputed, or with another interactive statement that causes the model to be implicitly recomputed.

Syntax: REG Procedure

The following statements are available in PROC REG:

PROC REG < *options* > ;
< **label:** >**MODEL** *dependents*=< *regressors* > < / *options* > ;
BY *variables* ;
FREQ *variable* ;
ID *variables* ;
VAR *variables* ;
WEIGHT *variable* ;
ADD *variables* ;
DELETE *variables* ;
< **label:** >**MTEST** < *equation, . . . ,equation* > < / *options* > ;
OUTPUT < **OUT=***SAS-data-set* >< *keyword=names* > < . . . *keyword=names* > ;
PAINT < *condition* | **ALLOBS** > < / *options* > | < **STATUS** | **UNDO** > ;
RESTRICT *equation, . . . ,equation* ;
REWEIGHT < *condition* | **ALLOBS** > < / *options* > | < **STATUS** | **UNDO** > ;
PLOT < *yvariable*xvariable* > < =*symbol* > < . . . *yvariable*xvariable* > < =*symbol* > < / *options* > ;
PRINT < *options* > < **ANOVA** > < **MODELDATA** > ;
REFIT ;
RESTRICT *equation, . . . ,equation* ;
REWEIGHT < *condition* | **ALLOBS** > < / *options* > | < **STATUS** | **UNDO** > ;
< **label:** >**TEST** *equation,*< *,. . . ,equation* > < / *option* > ;

Although there are numerous statements and options available in PROC REG, many analyses use only a few of them. Often you can find the features you need by looking at an example or by scanning this section.

In the preceding list, brackets denote optional specifications, and vertical bars denote a choice of one of the specifications separated by the vertical bars. In all cases, *label* is optional.

The PROC REG statement is required. To fit a model to the data, you must specify the MODEL statement. If you want to use only the options available in the PROC REG statement, you do not need a MODEL statement, but you must use a VAR statement. (See the example in the section "OUTSSCP= Data Sets" on page 5518.) Several MODEL statements can be used. In addition, several MTEST, OUTPUT, PAINT, PLOT, PRINT, RESTRICT, and TEST statements can follow each MODEL statement.

The ADD, DELETE, and REWEIGHT statements are used interactively to change the regression model and the data used in fitting the model. The ADD, DELETE, MTEST, OUTPUT, PLOT, PRINT, RESTRICT, and TEST statements implicitly refit the model; changes made to the model are reflected in the results from these statements. The REFIT statement is used to refit the model explicitly and is most helpful when it follows PAINT and REWEIGHT statements, which do not refit the model.

The BY, FREQ, ID, VAR, and WEIGHT statements are optionally specified once for the entire PROC step, and they must appear before the first RUN statement.

When a TYPE=CORR, TYPE=COV, or TYPE=SSCP data set is used as an input data set to PROC REG, statements and options that require the original data are not available. Specifically, the OUTPUT, PAINT, PLOT, and REWEIGHT statements and the MODEL and PRINT statement options P, R, CLM, CLI, DW, DWPROB, INFLUENCE, PARTIAL, and PARTIALDATA are disabled.

You can specify the following statements with the REG procedure in addition to the PROC REG statement:

ADD	adds independent variables to the regression model.
BY	specifies variables to define subgroups for the analysis.
DELETE	deletes independent variables from the regression model.
FREQ	specifies a frequency variable.
ID	names a variable to identify observations in the tables.
MODEL	specifies the dependent and independent variables in the regression model, requests a model selection method, displays predicted values, and provides details on the estimates (according to which options are selected).
MTEST	performs multivariate tests across multiple dependent variables.
OUTPUT	creates an output data set and names the variables to contain predicted values, residuals, and other diagnostic statistics.
PAINT	paints points in scatter plots.
PLOT	generates scatter plots.
PRINT	displays information about the model and can reset options.
REFIT	refits the model.
RESTRICT	places linear equality restrictions on the parameter estimates.
REWEIGHT	excludes specific observations from analysis or changes the weights of observations used.
TEST	performs an F test on linear functions of the parameters.
VAR	lists variables for which crossproducts are to be computed, variables that can be interactively added to the model, or variables to be used in scatter plots.
WEIGHT	declares a variable to weight observations.

PROC REG Statement

PROC REG < *options* > **;**

The PROC REG statement is required. If you want to fit a model to the data, you must also use a MODEL statement. If you want to use only the PROC REG options, you do not need a MODEL statement, but you must use a VAR statement. If you do not use a MODEL statement, then the COVOUT and OUTEST= options are not available.

Table 73.1 lists the options you can use with the PROC REG statement. Note that any option specified in the PROC REG statement applies to all MODEL statements.

Table 73.1 PROC REG Statement Options

Option	Description
Data Set Options	
DATA=	names a data set to use for the regression
OUTEST=	outputs a data set that contains parameter estimates and other model fit summary statistics
OUTSSCP=	outputs a data set that contains sums of squares and crossproducts
COVOUT	outputs the covariance matrix for parameter estimates to the OUTEST= data set
EDF	outputs the number of regressors, the error degrees of freedom, and the model R^2 to the OUTEST= data set
OUTSEB	outputs standard errors of the parameter estimates to the OUTEST= data set
OUTSTB	outputs standardized parameter estimates to the OUTEST= data set. Use only with the RIDGE= or PCOMIT= option.
OUTVIF	outputs the variance inflation factors to the OUTEST= data set. Use only with the RIDGE= or PCOMIT= option.
PCOMIT=	performs incomplete principal component analysis and outputs estimates to the OUTEST= data set
PRESS	outputs the PRESS statistic to the OUTEST= data set
RIDGE=	performs ridge regression analysis and outputs estimates to the OUTEST= data set
RSQUARE	same effect as the EDF option
TABLEOUT	outputs standard errors, confidence limits, and associated test statistics of the parameter estimates to the OUTEST= data set
ODS Graphics Options	
PLOTS=	produces ODS graphical displays
Traditional Graphics Options	
ANNOTATE=	specifies an annotation data set
GOUT=	specifies the graphics catalog in which graphics output is saved
Display Options	
CORR	displays correlation matrix for variables listed in MODEL and VAR statements

Table 73.1 *continued*

Option	Description
SIMPLE	displays simple statistics for each variable listed in MODEL and VAR statements
USCCP	displays uncorrected sums of squares and crossproducts matrix
ALL	displays all statistics (CORR, SIMPLE, and USSCP)
NOPRINT	suppresses output
LINEPRINTER	creates plots requested as line printer plot
Other Options	
ALPHA=	sets significance value for confidence and prediction intervals and tests
SINGULAR=	sets criterion for checking for singularity

Following are explanations of the options that you can specify in the PROC REG statement (in alphabetical order).

Note that any option specified in the PROC REG statement applies to all MODEL statements.

ALL

requests the display of many tables. Using the ALL option in the PROC REG statement is equivalent to specifying ALL in every MODEL statement. The ALL option also implies the CORR, SIMPLE, and USSCP options.

ALPHA=*number*

sets the significance level used for the construction of confidence intervals. The value must be between 0 and 1; the default value of 0.05 results in 95% intervals. This option affects the PROC REG option TABLEOUT; the MODEL options CLB, CLI, and CLM; the OUTPUT statement keywords LCL, LCLM, UCL, and UCLM; the PLOT statement keywords LCL., LCLM., UCL., and UCLM.; and the PLOT statement options CONF and PRED.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an input data set containing annotate variables, as described in *SAS/GRAPH Software: Reference*. You can use this data set to add features to the traditional graphics that you request with the PLOT statement. Features provided in this data set are applied to all plots produced in the current run of PROC REG. To add features to individual plots, use the ANNOTATE= option in the PLOT statement. This option cannot be used if the LINEPRINTER option is specified.

CORR

displays the correlation matrix for all variables listed in the MODEL or VAR statement.

COVOUT

outputs the covariance matrices for the parameter estimates to the OUTEST= data set. This option is valid only if the OUTEST= option is also specified. See the section "OUTEST= Data Set" on page 5512.

DATA=*SAS-data-set*

names the SAS data set to be used by PROC REG. The data set can be an ordinary SAS data set or a TYPE=CORR, TYPE=COV, or TYPE=SSCP data set. If one of these special TYPE= data sets is used, the OUTPUT, PAINT, PLOT, and REWEIGHT statements, ODS Graphics, and some options in the MODEL and PRINT statements are not available. See Appendix A, "Special SAS Data Sets," for more information about TYPE= data sets. If the DATA= option is not specified, PROC REG uses the most recently created SAS data set.

EDF

outputs the number of regressors in the model excluding and including the intercept, the error degrees of freedom, and the model R^2 to the OUTEST= data set.

GOUT=*graphics-catalog*

specifies the graphics catalog in which traditional graphics output is saved. The default *graphics-catalog* is WORK.GSEG. The GOUT= option cannot be used if the LINEPRINTER option is specified.

LINEPRINTER | LP

creates plots requested as line printer plots. If you do not specify this option, requested plots are created on a high-resolution graphics device. This option is required if plots are requested and you do not have SAS/GRAPH software.

NOPRINT

suppresses the normal display of results. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

OUTEST=*SAS-data-set*

requests that parameter estimates and optional model fit summary statistics be output to this data set. See the section "OUTEST= Data Set" on page 5512 for details. If you want to create a permanent SAS data set, you must specify a two-level name (refer to the section "SAS Files" in *SAS Language Reference: Concepts* for more information about permanent SAS data sets).

OUTSEB

outputs the standard errors of the parameter estimates to the OUTEST= data set. The value SEB for the variable _TYPE_ identifies the standard errors. If the RIDGE= or PCOMIT= option is specified, additional observations are included and identified by the values RIDGESEB and IPCSEB, respectively, for the variable _TYPE_. The standard errors for ridge regression estimates and IPC estimates are limited in their usefulness because these estimates are biased. This option is available for all model selection methods except RSQUARE, ADJRSQ, and CP.

OUTSSCP=*SAS-data-set*

requests that the sums of squares and crossproducts matrix be output to this TYPE=SSCP data set. See the section "OUTSSCP= Data Sets" on page 5518 for details. If you want to create a permanent SAS data set, you must specify a two-level name (refer to the section "SAS Files" in *SAS Language Reference: Concepts* for more information about permanent SAS data sets).

OUTSTB
outputs the standardized parameter estimates as well as the usual estimates to the OUTEST= data set when the RIDGE= or PCOMIT= option is specified. The values RIDGESTB and IPCSTB for the variable _TYPE_ identify ridge regression estimates and IPC estimates, respectively.

OUTVIF
outputs the variance inflation factors (VIF) to the OUTEST= data set when the RIDGE= or PCOMIT= option is specified. The factors are the diagonal elements of the inverse of the correlation matrix of regressors as adjusted by ridge regression or IPC analysis. These observations are identified in the output data set by the values RIDGEVIF and IPCVIF for the variable _TYPE_.

PCOMIT=*list*
requests an incomplete principal component (IPC) analysis for each value *m* in the list. The procedure computes parameter estimates by using all but the last *m* principal components. Each value of *m* produces a set of IPC estimates, which are output to the OUTEST= data set. The values of *m* are saved by the variable _PCOMIT_, and the value of the variable _TYPE_ is set to IPC to identify the estimates. Only nonnegative integers can be specified with the PCOMIT= option.

If you specify the PCOMIT= option, RESTRICT statements are ignored.

PLOTS <(*global-plot-options***)> <= *plot-request* <(***options***)>>**
PLOTS <(*global-plot-options***)> <= (***plot-request* <(***options***)>** **<... *plot-request* <(***options***)>>)>**
controls the plots produced through ODS Graphics. When you specify only one plot request, you can omit the parentheses around the plot request. Here are some examples:

```
plots         = none
plots         = diagnostics(unpack)
plots         = (all fit(stats)=none)
plots(label)  = (rstudentbyleverage cooksd)
plots(only)   = (diagnostics(stats=all) fit(nocli stats=(aic sbc)
```

You must enable ODS Graphics before requesting plots, as shown in the following example. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

```
ods graphics on;

proc reg;
   model y = x1-x10;
run;

proc reg plots=diagnostics(stats=(default aic sbc));
   model y = x1-x10;
run;

ods graphics off;
```

If you have enabled ODS Graphics but do not specify the PLOTS= option, then PROC REG produces a default set of plots. Table 73.2 lists the default set of plots produced.

Table 73.2 Default ODS Graphics Produced

Plot	Conditional On
DiagnosticsPanel	Unconditional
ResidualPlot	Unconditional
FitPlot	Model with one regressor (excluding intercept)
PartialPlot	PARTIAL option specified in MODEL statement
RidgePanel	RIDGE= option specified in PROC REG or MODEL statement

For models with multiple dependent variables, separate plots are produced for each dependent variable. For jobs with more than one MODEL statement, plots are produced for each model statement.

The *global-options* apply to all plots generated by the REG procedure, unless it is altered by a *specific-plot-option*. The following global plot options are available:

LABEL
specifies that the LABEL option be applied to each plot that supports a LABEL option. See the descriptions of the specific plots for details.

MAXPOINTS=NONE | *number*
specifies that plots with elements that require processing more than *number* points be suppressed. The default is MAXPOINTS=5000. This cutoff is ignored if you specify MAXPOINTS=NONE.

MODELLABEL
requests that the model label be displayed in the upper-left corner of all plots. This option is useful when you use more than one MODEL statement.

ONLY
suppress the default plots. Only plots specifically requested are displayed.

STATS=ALL | **DEFAULT** | **NONE** | (*plot-statistics*)
requests statistics that are included on the fit plot and diagnostics panel. Table 73.3 lists the statistics that you can request. STATS=ALL requests all these statistics; STATS=NONE suppresses them.

Table 73.3 Statistics Available on Plots

Keyword	Default	Description
ADJRSQ	x	adjusted R-square
AIC		Akaike's information criterion
BIC		Sawa's Bayesian information criterion
CP		Mallows' C_p statistic

Table 73.3 *continued*

Keyword	Default	Description
COEFFVAR		coefficient of variation
DEPMEAN		mean of dependent
DEFAULT		all default statistics
EDF	x	error degrees of freedom
GMSEP		estimated MSE of prediction, assuming multivariate normality
JP		final prediction error
MSE	x	mean squared error
NOBS	x	number of observations used
NPARM	x	number of parameters in the model (including the intercept)
PC		Amemiya's prediction criterion
RSQUARE	x	R-square
SBC		SBC statistic
SP		SP statistic
SSE		error sum of squares

You request statistics in addition to the default set by including the keyword DEFAULT in the *plot-statistics* list.

UNPACK
suppresses paneling.

USEALL
specifies that predicted values at data points with missing dependent variable(s) be included on appropriate plots. By default, only points used in constructing the SSCP matrix appear on plots.

The following specific plots are available:

ADJRSQ < (*adjrsq-options*) >
displays the adjusted R-square values for the models examined when you request variable selection with the SELECTION= option in the MODEL statement.

The following *adjrsq-options* are available for models where you request the RSQUARE, ADJRSQ, or CP selection method:

LABEL
requests that the model number corresponding to the one displayed in the "Subset Selection Summary" table be used to label the model with the largest adjusted R-square statistic at each value of the number of parameters.

LABELVARS
requests that the list (excluding the intercept) of the regressors in the relevant model be used to label the model with the largest adjusted R-square statistic at each value of the number of parameters.

AIC <(*aic-options*)>
displays Akaike's information criterion (AIC) for the models examined when you request variable selection with the SELECTION= option in the MODEL statement.

The following *aic-options* are available for models where you request the RSQUARE, ADJRSQ, or CP selection method:

LABEL
requests that the model number corresponding to the one displayed in the "Subset Selection Summary" table be used to label the model with the smallest AIC statistic at each value of the number of parameters.

LABELVARS
requests that the list (excluding the intercept) of the regressors in the relevant model be used to label the model with the smallest AIC statistic at each value of the number of parameters.

ALL
produces all appropriate plots.

BIC <(*bic-options*)>
displays Sawa's Bayesian information criterion (BIC) for the models examined when you request variable selection with the SELECTION= option in the MODEL statement.

The following *bic-options* are available for models where you request the RSQUARE, ADJRSQ, or CP selection method:

LABEL
requests that the model number corresponding to the one displayed in the "Subset Selection Summary" table be used to label the model with the smallest BIC statistic at each value of the number of parameters.

LABELVARS
requests that the list (excluding the intercept) of the regressors in the relevant model be used to label the model with the smallest BIC statistic at each value of the number of parameters.

COOKSD <(LABEL)>
plots Cook's D statistic by observation number. Observations whose Cook's D statistic lies above the horizontal reference line at value $4/n$, where n is the number of observations used, are deemed to be influential (Rawlings 1998). If you specify the LABEL option, then points deemed as influential are labeled. If you do not specify an ID variable, the observation number within the current BY group is used as the label. If you specify one or more ID variables in one or more ID statements, then the first ID variable you specify is used for the labeling.

CP <(*cp-options*)>
displays Mallow's C_p statistic for the models examined when you request variable selection with the SELECTION= option in the MODEL statement. For models where you request the RSQUARE, ADJRSQ, or CP selection, reference lines corresponding

to the equations $C_p = p$ and $C_p = 2p - p_{full}$, where p_{full} is the number of parameters in the full model (excluding the intercept) and p is the number of parameters in the subset model (including the intercept), are displayed on the plot of C_p versus p. For the purpose of parameter estimation, Hocking (1976) suggests selecting a model where $C_p \leq 2p - p_{full}$. For the purpose of prediction, Hocking suggests the criterion $C_p \leq p$. Mallows (1973) suggests that all subset models with C_p small and near p be considered for further study.

The following *cp-options* are available for models where you request the RSQUARE, ADJRSQ, or CP selection method:

LABEL

requests that the model number corresponding to the one displayed in the "Subset Selection Summary" table be used to label the model with the smallest C_p statistic at each value of the number of parameters.

LABELVARS

requests that the list (excluding the intercept) of the regressors in the relevant model be used to label the model with the smallest C_p statistic at each value of the number of parameters.

CRITERIA | CRITERIONPANEL < (*criteria-options*) >

produces a panel of fit criteria for the models examined when you request variable selection with the SELECTION= option in the MODEL statement. The fit criteria displayed are R-square, adjusted R-square, Mallow's C_p, Akaike's information criterion (AIC), Sawa's Bayesian information criterion (BIC), and Schwarz's Bayesian information criterion (SBC). For SELECTION=RSQUARE, SELECTION=ADJRSQ, or SELECTION=CP, scatter plots of these statistics versus the number of parameters (including the intercept) are displayed. For other selection methods, line plots of these statistics as function of the selection step number are displayed.

The following *criteria-options* are available:

LABEL

requests that the model number corresponding to the one displayed in the "Subset Selection Summary" table be used to label the best model at each value of the number of parameters. This option applies only to the RSQUARE, ADJRSQ, and CP selection methods.

LABELVARS

requests that the list (excluding the intercept) of the regressors in the relevant model be used to label the best model at each value of the number of parameters. Since these labels are typically long, LABELVARS is supported only when the panel is unpacked. This option applies only to the RSQUARE, ADJRSQ, and CP selection methods.

UNPACK

suppresses paneling. Separate plots are produced for each of the six fit statistics. For models where you request the RSQUARE, ADJRSQ, or CP selection, two reference lines corresponding to the equations $C_p = p$ and $C_p = 2p - p_{full}$,

where p_{full} is the number of parameters in the full model (excluding the intercept) and p is the number of parameters in the subset model (including the intercept), are displayed on the plot of C_p versus p. For the purpose of parameter estimation, Hocking (1976) suggests selecting a model where $C_p \leq 2p - p_{full}$. For the purpose of prediction, Hocking suggests the criterion $C_p \leq p$. Mallows (1973) suggests that all subset models with C_p small and near p be considered for further study.

DFBETAS < (*DFBETAS-options*) >

produces panels of DFBETAS by observation number for the regressors in the model. Note that each panel contains at most six plots, and multiple panels are used in the case where there are more than six regressors (including the intercept) in the model. Observations whose DFBETAS' statistics for a regressor are greater in magnitude than $2/\sqrt{n}$, where n is the number of observations used, are deemed to be influential for that regressor (Rawlings 1998).

The following *DFBETAS-options* are available:

COMMONAXES

specifies that the same DFBETAS axis be used in all panels when multiple panels are needed. By default, the DFBETAS axis is chosen independently for each panel. If you also specify the *UNPACK* option, then the same DFBETAS axis is used for each regressor.

LABEL

specifies that observations whose magnitude are greater than $2/\sqrt{n}$ be labeled. If you do not specify an ID variable, the observation number within the current BY group is used as the label. If you specify one or more ID variables on one or more ID statements, then the first ID variable you specify is used for the labeling.

UNPACK

suppresses paneling. The DFBETAS statistics for each regressor are displayed on separate plots.

DFFITS < (LABEL) >

plots the DFFITS statistic by observation number. Observations whose DFFITS' statistic is greater in magnitude than $2\sqrt{p/n}$, where n is the number of observations used and p is the number of regressors, are deemed to be influential (Rawlings 1998). If you specify the LABEL option, then these influential observations are labeled. If you do not specify an ID variable, the observation number within the current BY group is used as the label. If you specify one or more ID variables in one or more ID statements, then the first ID variable you specify is used for the labeling.

DIAGNOSTICS < (*diagnostics-options*) >
produces a summary panel of fit diagnostics:

- residuals versus the predicted values
- studentized residuals versus the predicted values
- studentized residuals versus the leverage
- normal quantile plot of the residuals
- dependent variable values versus the predicted values
- Cook's D versus observation number
- histogram of the residuals
- "Residual-Fit" (or RF) plot consisting of side-by-side quantile plots of the centered fit and the residuals
- box plot of the residuals if you specify the STATS=NONE suboption

You can specify the following *diagnostics-options*:

STATS=*stats-options*
determines which model fit statistics are included in the panel. See the global STATS= suboption for details. The PLOTS= suboption of the DIAGNOSTICSPANEL option overrides the global PLOTS= suboption.

UNPACK
produces the eight plots in the panel as individual plots. Note that you can also request individual plots in the panel by name without having to unpack the panel.

FITPLOT | FIT < (*fit-options*) >
produces a scatter plot of the data overlaid with the regression line, confidence band, and prediction band for models that depend on at most one regressor excluding the intercept.

You can specify the following *fit-options*:

NOCLI
suppresses the prediction limits.

NOCLM
suppresses the confidence limits.

NOLIMITS
suppresses the confidence and prediction limits.

STATS=*stats-options*
determines which model fit statistics are included in the panel. See the global STATS= suboption for details. The PLOTS= suboption of the FITPLOT option overrides the global PLOTS= suboption.

OBSERVEDBYPREDICTED < (LABEL) >
plots dependent variable values by the predicted values. If you specify the LABEL option, then points deemed as outliers or influential (see the RSTUDENTBYLEVERAGE option for details) are labeled.

NONE
suppresses all plots.

PARTIAL < (UNPACK) >
produces panels of partial regression plots for each regressor with at most six regressors per panel. If you specify the UNPACK option, then all partial plot panels are unpacked.

PREDICTIONS (X=*numeric-variable* **<** *prediction-options* **>)**
produces a panel of two plots whose horizontal axis is the variable you specify in the required X= suboption. The upper plot in the panel is a scatter plot of the residuals. The lower plot shows the data overlaid with the regression line, confidence band, and prediction band. This plot is appropriate for models where all regressors are known to be functions of the single variable that you specify in the X= suboption.

You can specify the following *prediction-options*:

NOCLI
suppresses the prediction limits.

NOCLM
suppresses the confidence limits

NOLIMITS
suppresses the confidence and prediction limits

SMOOTH
requests a nonparametric smooth of the residuals as a function of the variable you specify in the X= suboption. This nonparametric fit is a loess fit that uses local linear polynomials, linear interpolation, and a smoothing parameter selected that yields a local minimum of the corrected Akaike information criterion (AICC). See Chapter 50, "The LOESS Procedure," for details. The SMOOTH option is not supported when a FREQ statement is used.

UNPACK
suppresses paneling.

QQPLOT | QQ
produces a normal quantile plot of the residuals.

RESIDUALBOXPLOT | BOXPLOT < (LABEL) >
produces a box plot consisting of the residuals. If you specify label option, points deemed far-outliers are labeled. If you do not specify an ID variable, the observation number within the current BY group is used as the label. If you specify one or more ID variables in one or more ID statements, then the first ID variable you specify is used for the labeling.

RESIDUALBYPREDICTED < (LABEL) >
plots residuals by predicted values. If you specify the LABEL option, then points deemed as outliers or influential (see the RSTUDENTBYLEVERAGE option for details) are labeled.

RESIDUALS < *residual-options***) >**

produces panels of the residuals versus the regressors in the model. Note that each panel contains at most six plots, and multiple panels are used in the case where there are more than six regressors (including the intercept) in the model.

The following *residual-options* are available:

SMOOTH

requests a nonparametric smooth of the residuals for each regressor. Each nonparametric fit is a loess fit that uses local linear polynomials, linear interpolation, and a smoothing parameter selected that yields a local minimum of the corrected Akaike information criterion (AICC). See Chapter 50, "The LOESS Procedure," for details. The SMOOTH option is not supported when a FREQ statement is used.

UNPACK

suppresses paneling.

RESIDUALHISTOGRAM

produces a histogram of the residuals.

RFPLOT | RF

produces a "Residual-Fit" (or RF) plot consisting of side-by-side quantile plots of the centered fit and the residuals. This plot "shows how much variation in the data is explained by the fit and how much remains in the residuals" (Cleveland 1993).

RIDGE | RIDGEPANEL | RIDGEPLOT <(*ridge-options***)>**

creates panels of VIF values and standardized ridge estimates by ridge values for each coefficient. The VIF values for each coefficient are connected by lines and are displayed in the upper plot in each panel. The points corresponding to the standardized estimates of each coefficient are connected by lines and are displayed in the lower plot in each panel. By default, at most 10 coefficients are represented in a panel and multiple panels are produced for models with more than 10 regressors. For ridge estimates to be computed and plotted, the OUTEST= option must be specified in the PROC REG statement, and the RIDGE= list must be specified in either the PROC REG or the MODEL statement. (See Example 73.5.)

The following *ridge-options* are available:

COMMONAXES

specifies that the same VIF axis and the same standardized estimate axis are used in all panels when multiple panels are needed. By default, these axes are chosen independently for the regressors shown in each panel.

RIDGEAXIS=LINEAR | LOG

specifies the axis type used to display the ridge parameters. The default is RIDGEAXIS=LINEAR. Note that the point with the ridge parameter equal to zero is not displayed if you specify RIDGEAXIS=LOG.

UNPACK
suppresses paneling. The traces of the VIF statistics and standardized estimates are shown in separate plots.

VARSPERPLOT=ALL

VARSPERPLOT=*number*
specifies the maximum number of regressors displayed in each panel or in each plot if you additionally specify the *UNPACK* option. If you specify VARSPERPLOT=ALL, then the VIF values and ridge traces for all regressors are displayed in a single panel.

VIFAXIS=LINEAR | LOG
specifies the axis type used to display the VIF statistics. The default is VIFAXIS=LINEAR.

RSQUARE < (*rsquare-options***) >**
displays the R-square values for the models examined when you request variable selection with the SELECTION= option in the MODEL statement.

The following *rsquare-options* are available for models where you request the RSQUARE, ADJRSQ, or CP selection method:

LABEL
requests that the model number corresponding to the one displayed in the "Subset Selection Summary" table be used to label the model with the largest R-square statistic at each value of the number of parameters.

LABELVARS
requests that the list (excluding the intercept) of the regressors in the relevant model be used to label the model with the largest R-square statistic at each value of the number of parameters.

RSTUDENTBYLEVERAGE < (LABEL) >
plots studentized residuals by leverage. Observations whose studentized residuals lie outside the band between the reference lines RSTUDENT $= \pm 2$ are deemed outliers. Observations whose leverage values are greater than the vertical reference LEVERAGE $= 2p/n$, where p is the number of parameters excluding the intercept and n is the number of observations used, are deemed influential (Rawlings 1998). If you specify the LABEL option, then points deemed as outliers or influential are labeled. If you do not specify an ID variable, the observation number within the current BY group is used as the label. If you specify one or more ID variables in one or more ID statements, then the first ID variable you specify is used for the labeling.

RSTUDENTBYPREDICTED < (LABEL) >
plots studentized residuals by predicted values. If you specify the LABEL option, then points deemed as outliers or influential (see the RSTUDENTBYLEVERAGE option for details) are labeled.

SBC <(*sbc-options*)>
displays Schwarz's Bayesian information criterion (SBC) for the models examined when you request variable selection with the SELECTION= option in the MODEL statement.

The following *sbc-options* are available for models where you request the RSQUARE, ADJRSQ, or CP selection method:

LABEL
requests that the model number corresponding to the one displayed in the "Subset Selection Summary" table be used to label the model with the smallest SBC statistic at each value of the number of parameters.

LABELVARS
requests that the list (excluding the intercept) of the regressors in the relevant model be used to label the model with the smallest SBC statistic at each value of the number of parameters.

PRESS
outputs the PRESS statistic to the OUTEST= data set. The values of this statistic are saved in the variable _PRESS_. This option is available for all model selection methods except RSQUARE, ADJRSQ, and CP.

RIDGE=*list*
requests a ridge regression analysis and specifies the values of the ridge constant k (see the section "Computations for Ridge Regression and IPC Analysis" on page 5583). Each value of k produces a set of ridge regression estimates that are placed in the OUTEST= data set. The values of k are saved by the variable _RIDGE_, and the value of the variable _TYPE_ is set to RIDGE to identify the estimates.

Only nonnegative numbers can be specified with the RIDGE= option. Example 73.5 illustrates this option.

If ODS Graphics is in effect (see the section "ODS Graphics" on page 5589), then ridge regression plots are automatically produced. These plots consist of panels containing ridge traces for the regressors, with at most eight ridge traces per panel.

If you specify the RIDGE= option, RESTRICT statements are ignored.

RSQUARE
has the same effect as the EDF option.

SIMPLE
displays the sum, mean, variance, standard deviation, and uncorrected sum of squares for each variable used in PROC REG.

SINGULAR=*n*
tunes the mechanism used to check for singularities. The default value is machine dependent but is approximately 1E−7 on most machines. This option is rarely needed.

Singularity checking is described in the section "Computational Methods" on page 5584.

TABLEOUT

outputs the standard errors and $100(1-\alpha)\%$ confidence limits for the parameter estimates, the t statistics for testing if the estimates are zero, and the associated p-values to the OUTEST= data set. The _TYPE_ variable values STDERR, LnB, UnB, T, and PVALUE, where $n = 100(1-\alpha)$, identify these rows in the OUTEST= data set. The α level can be set with the ALPHA= option in the PROC REG or MODEL statement. The OUTEST= option must be specified in the PROC REG statement for this option to take effect.

USSCP

displays the uncorrected sums-of-squares and crossproducts matrix for all variables used in the procedure.

ADD Statement

ADD *variables* **;**

The ADD statement adds independent variables to the regression model. Only variables used in the VAR statement or used in MODEL statements before the first RUN statement can be added to the model. You can use the ADD statement interactively to add variables to the model or to include a variable that was previously deleted with a DELETE statement. Each use of the ADD statement modifies the MODEL label.

See the section "Interactive Analysis" on page 5519 for an example.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC REG to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in the order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives.

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the REG procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

When a BY statement is used with PROC REG, interactive processing is not possible; that is, once the first RUN statement is encountered, processing proceeds for each BY group in the data set, and no further statements are accepted by the procedure. A BY statement that appears after the first RUN statement is ignored.

For more information about the BY statement, see *SAS Language Reference: Contents*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

DELETE Statement

DELETE *variables* **;**

The DELETE statement deletes independent variables from the regression model. The DELETE statement performs the opposite function of the ADD statement and is used in a similar manner. Each use of the DELETE statement modifies the MODEL label.

For an example of how the ADD statement is used (and how the DELETE statement can be used), see the section "Interactive Analysis" on page 5519.

FREQ Statement

FREQ *variable* **;**

When a FREQ statement appears, each observation in the input data set is assumed to represent n observations, where n is the value of the FREQ variable. The analysis produced when you use a FREQ statement is the same as an analysis produced by using a data set that contains n observations in place of each observation in the input data set. When the procedure determines degrees of freedom for significance tests, the total number of observations is considered to be equal to the sum of the values of the FREQ variable.

If the value of the FREQ variable is missing or is less than 1, the observation is not used in the analysis. If the value is not an integer, only the integer portion is used.

The FREQ statement must appear before the first RUN statement, or it is ignored.

ID Statement

ID *variables* **;**

When one of the MODEL statement options CLI, CLM, P, R, and INFLUENCE is requested, the variables listed in the ID statement are displayed beside each observation. These variables can be used to identify each observation. If the ID statement is omitted, the observation number is used to identify the observations.

Although there are no restrictions on the length of ID variables, PROC REG might truncate ID values to 16 characters for display purposes.

MODEL Statement

< *label:* > MODEL *dependents=< regressors > < / options >* **;**

After the keyword MODEL, the dependent (response) variables are specified, followed by an equal sign and the regressor variables. Variables specified in the MODEL statement must be numeric variables in the data set being analyzed. For example, if you want to specify a quadratic term for variable *X1* in the model, you cannot use X1*X1 in the MODEL statement but must create a new variable (for example, X1SQUARE=X1*X1) in a DATA step and use this new variable in the MODEL statement. The label in the MODEL statement is optional.

Table 73.4 lists the options available in the MODEL statement. Equations for the statistics available are given in the section "Model Fit and Diagnostic Statistics" on page 5557.

Table 73.4 MODEL Statement Options

Option	Description
Model Selection and Details of Selection	
SELECTION=	specifies model selection method
BEST=	specifies maximum number of subset models displayed or output to the OUTEST= data set
DETAILS	produces summary statistics at each step
DETAILS=	specifies the display details for FORWARD, BACKWARD, and STEPWISE methods
GROUPNAMES=	provides names for groups of variables
INCLUDE=	includes first n variables in the model
MAXSTEP=	specifies maximum number of steps that might be performed
NOINT	fits a model without the intercept term
PCOMIT=	performs incomplete principal component analysis and outputs estimates to the OUTEST= data set
RIDGE=	performs ridge regression analysis and outputs estimates to the OUTEST= data set
SLE=	sets criterion for entry into model
SLS=	sets criterion for staying in model
START=	specifies number of variables in model to begin the comparing and switching process
STOP=	stops selection criterion
Statistics	
ADJRSQ	computes adjusted R^2
AIC	computes Akaike's information criterion
B	computes parameter estimates for each model
BIC	computes Sawa's Bayesian information criterion
CP	computes Mallows' C_p statistic

Table 73.4 *continued*

Option	Description
GMSEP	computes estimated MSE of prediction assuming multivariate normality
JP	computes J_p, the final prediction error
MSE	computes MSE for each model
PC	computes Amemiya's prediction criterion
RMSE	displays root MSE for each model
SBC	computes the SBC statistic
SP	computes S_p statistic for each model
SSE	computes error sum of squares for each model
Data Set Options	
EDF	outputs the number of regressors, the error degrees of freedom, and the model R^2 to the OUTEST= data set
OUTSEB	outputs standard errors of the parameter estimates to the OUTEST= data set
OUTSTB	outputs standardized parameter estimates to the OUTEST= data set. Use only with the RIDGE= or PCOMIT= option.
OUTVIF	outputs the variance inflation factors to the OUTEST= data set. Use only with the RIDGE= or PCOMIT= option.
PRESS	outputs the PRESS statistic to the OUTEST= data set
RSQUARE	has same effect as the EDF option
Regression Calculations	
I	displays inverse of sums of squares and crossproducts
XPX	displays sums-of-squares and crossproducts matrix
Details on Estimates	
ACOV	displays heteroscedasticity- consistent covariance matrix of estimates and heteroscedasticity-consistent standard errors
ACOVMETHOD=	specifies method for computing the asymptotic heteroscedasticity-consistent covariance matrix
COLLIN	produces collinearity analysis
COLLINOINT	produces collinearity analysis with intercept adjusted out
CORRB	displays correlation matrix of estimates
COVB	displays covariance matrix of estimates
HCC	displays heteroscedasticity-consistent standard errors
HCCMETHOD=	specifies method for computing the asymptotic heteroscedasticity-consistent covariance matrix
LACKFIT	performs lack-of-fit test
PARTIALR2	displays squared semipartial correlation coefficients computed using Type I sums of squares
PCORR1	displays squared partial correlation coefficients computed using Type I sums of squares
PCORR2	displays squared partial correlation coefficients computed using Type II sums of squares

Table 73.4 *continued*

Option	Description
SCORR1	displays squared semipartial correlation coefficients computed using Type I sums of squares
SCORR2	displays squared semipartial correlation coefficients computed using Type II sums of squares
SEQB	displays a sequence of parameter estimates during selection process
SPEC	tests that first and second moments of model are correctly specified
SS1	displays the sequential sums of squares
SS2	displays the partial sums of squares
STB	displays standardized parameter estimates
TOL	displays tolerance values for parameter estimates
WHITE	displays heteroscedasticity-consistent standard errors
VIF	computes variance-inflation factors
Predicted and Residual Values	
CLB	computes $100(1-\alpha)\%$ confidence limits for the parameter estimates
CLI	computes $100(1-\alpha)\%$ confidence limits for an individual predicted value
CLM	computes $100(1-\alpha)\%$ confidence limits for the expected value of the dependent variable
DW	computes a Durbin-Watson statistic
DWPROB	computes a Durbin-Watson statistic and p-value
INFLUENCE	computes influence statistics
P	computes predicted values
PARTIAL	displays partial regression plots for each regressor
PARTIALDATA	displays partial regression data
R	produces analysis of residuals
Display Options and Other Options	
ALL	requests the following options: ACOV, CLB, CLI, CLM, CORRB, COVB, HCC, I, P, PCORR1, PCORR2, R, SCORR1, SCORR2, SEQB, SPEC, SS1, SS2, STB, TOL, VIF, XPX
ALPHA=	sets significance value for confidence and prediction intervals and tests
NOPRINT	suppresses display of results
SIGMA=	specifies the true standard deviation of error term for computing CP and BIC
SINGULAR=	sets criterion for checking for singularity

You can specify the following options in the MODEL statement after a slash (/).

ACOV

displays the estimated asymptotic covariance matrix of the estimates under the hypothesis of heteroscedasticity and heteroscedasticity-consistent standard errors of parameter estimates. See the HCCMETHOD= option and the HCC option and the section "Testing for Heteroscedasticity" on page 5575 for more information.

ACOVMETHOD=0,1,2, or 3

See the HCCMETHOD= option.

ADJRSQ

computes R^2 adjusted for degrees of freedom for each model selected (Darlington 1968; Judge et al. 1980).

AIC

outputs Akaike's information criterion for each model selected (Akaike 1969; Judge et al. 1980) to the OUTEST= data set. If SELECTION=ADJRSQ, SELECTION=RSQUARE, or SELECTION=CP is specified, then the AIC statistic is also added to the SubsetSelSummary table.

ALL

requests all these options: ACOV, CLB, CLI, CLM, CORRB, COVB, HCC, I, P, PCORR1, PCORR2, R, SCORR1, SCORR2, SEQB, SPEC, SS1, SS2, STB, TOL, VIF, and XPX.

ALPHA=*number*

sets the significance level used for the construction of confidence intervals for the current MODEL statement. The value must be between 0 and 1; the default value of 0.05 results in 95% intervals. This option affects the MODEL options CLB, CLI, and CLM; the OUTPUT statement keywords LCL, LCLM, UCL, and UCLM; the PLOT statement keywords LCL., LCLM., UCL., and UCLM.; and the PLOT statement options CONF and PRED. If you specify this option in the MODEL statement, it takes precedence over the ALPHA= option in the PROC REG statement.

B

is used with the RSQUARE, ADJRSQ, and CP model-selection methods to compute estimated regression coefficients for each model selected.

BEST=*n*

is used with the RSQUARE, ADJRSQ, and CP model-selection methods. If SELECTION=CP or SELECTION=ADJRSQ is specified, the BEST= option specifies the maximum number of subset models to be displayed or output to the OUTEST= data set. For SELECTION=RSQUARE, the BEST= option requests the maximum number of subset models for each size.

If the BEST= option is used without the B option (displaying estimated regression coefficients), the variables in each MODEL are listed in order of inclusion instead of the order in which they appear in the MODEL statement.

If the BEST= option is omitted and the number of regressors is less than 11, all possible subsets are evaluated. If the BEST= option is omitted and the number of regressors is greater

than 10, the number of subsets selected is, at most, equal to the number of regressors. A small value of the BEST= option greatly reduces the CPU time required for large problems.

BIC

outputs Sawa's Bayesian information criterion for each model selected (Sawa 1978; Judge et al. 1980) to the OUTEST= data set. If SELECTION=ADJRSQ, SELECTION=RSQUARE, or SELECTION=CP is specified, then the BIC statistic is also added to the SubsetSelSummary table.

CLB

requests the $100(1-\alpha)\%$ upper and lower confidence limits for the parameter estimates. By default, the 95% limits are computed; the ALPHA= option in the PROC REG or MODEL statement can be used to change the α level. If any of the MODEL statement options ACOV, HCC, or WHITE are in effect, then the CLB option also produces heteroscedasticity-consistent $100(1-\alpha)\%$ upper and lower confidence limits for the parameter estimates.

CLI

requests the $100(1-\alpha)\%$ upper and lower confidence limits for an individual predicted value. By default, the 95% limits are computed; the ALPHA= option in the PROC REG or MODEL statement can be used to change the α level. The confidence limits reflect variation in the error, as well as variation in the parameter estimates. See the section "Predicted and Residual Values" on page 5531 and Chapter 4, "Introduction to Regression Procedures," for more information.

CLM

displays the $100(1-\alpha)\%$ upper and lower confidence limits for the expected value of the dependent variable (mean) for each observation. By default, the 95% limits are computed; the ALPHA= in the PROC REG or MODEL statement can be used to change the α level. This is not a prediction interval (see the CLI option) because it takes into account only the variation in the parameter estimates, not the variation in the error term. See the section "Predicted and Residual Values" on page 5531 and Chapter 4, "Introduction to Regression Procedures," for more information.

COLLIN

requests a detailed analysis of collinearity among the regressors. This includes eigenvalues, condition indices, and decomposition of the variances of the estimates with respect to each eigenvalue. See the section "Collinearity Diagnostics" on page 5555.

COLLINOINT

requests the same analysis as the COLLIN option with the intercept variable adjusted out rather than included in the diagnostics. See the section "Collinearity Diagnostics" on page 5555.

CORRB

displays the correlation matrix of the estimates. This is the $(\mathbf{X}'\mathbf{X})^{-1}$ matrix scaled to unit diagonals.

COVB

displays the estimated covariance matrix of the estimates. This matrix is $(\mathbf{X}'\mathbf{X})^{-1}s^2$, where s^2 is the estimated mean squared error.

CP

outputs Mallows' C_p statistic for each model selected (Mallows 1973; Hocking 1976) to the OUTEST= data set. See the section "Criteria Used in Model-Selection Methods" on page 5526 for a discussion of the use of C_p. If SELECTION=ADJRSQ, SELECTION=RSQUARE, or SELECTION=CP is specified, then the C_p statistic is also added to the SubsetSelSummary table.

DETAILS

DETAILS=*name*

specifies the level of detail produced when the BACKWARD, FORWARD, or STEPWISE method is used, where *name* can be ALL, STEPS, or SUMMARY. The DETAILS or DETAILS=ALL option produces entry and removal statistics for each variable in the model building process, ANOVA and parameter estimates at each step, and a selection summary table. The option DETAILS=STEPS provides the step information and summary table. The option DETAILS=SUMMARY produces only the summary table. The default if the DETAILS option is omitted is DETAILS=STEPS.

DW

calculates a Durbin-Watson statistic to test whether or not the errors have first-order autocorrelation. (This test is appropriate only for time series data.) Note that your data should be sorted by the date/time ID variable before you use this option. The sample autocorrelation of the residuals is also produced. See the section "Autocorrelation in Time Series Data" on page 5581.

DWPROB

calculates a Durbin-Watson statistic and a p-value to test whether or not the errors have first-order autocorrelation. Note that it is not necessary to specify the DW option if the DWPROB option is specified. (This test is appropriate only for time series data.) Note that your data should be sorted by the date/time ID variable before you use this option. The sample autocorrelation of the residuals is also produced. See the section "Autocorrelation in Time Series Data" on page 5581.

EDF

outputs the number of regressors in the model excluding and including the intercept, the error degrees of freedom, and the model R^2 to the OUTEST= data set.

GMSEP

outputs the estimated mean square error of prediction assuming that both independent and dependent variables are multivariate normal (Stein 1960; Darlington 1968) to the OUTEST= data set. (Note that Hocking's formula (1976, eq. 4.20) contains a misprint: "$n - 1$" should read "$n - 2$.") If SELECTION=ADJRSQ, SELECTION=RSQUARE, or SELECTION=CP is specified, then the GMSEP statistic is also added to the SubsetSelSummary table.

GROUPNAMES=*'name1' 'name2' ...*

provides names for variable groups. This option is available only in the BACKWARD, FORWARD, and STEPWISE methods. The group name can be up to 32 characters. Subsets of independent variables listed in the MODEL statement can be designated as variable groups. This is done by enclosing the appropriate variables in braces. Variables in the same group are entered into or removed from the regression model at the same time. However, if the

tolerance of any variable (see the TOL option on page 5480) in a group is less than the setting of the SINGULAR= option, then the variable is not entered into the model with the rest of its group. If the GROUPNAMES= option is not used, then the names GROUP1, GROUP2, ..., GROUP*n* are assigned to groups encountered in the MODEL statement. Variables not enclosed by braces are used as groups of a single variable.

For example:

```
model y={x1 x2} x3 / selection=stepwise
  groupnames='x1 x2' 'x3';
```

Another example:

```
model y={ht wgt age} bodyfat / selection=forward
  groupnames='htwgtage' 'bodyfat';
```

HCC

requests heteroscedasticity-consistent standard errors of the parameter estimates. You can use the HCCMETHOD= option to specify the method used to compute the heteroscedasticity-consistent covariance matrix.

HCCMETHOD=0,1,2, or 3

specifies the method used to obtain a heteroscedasticity-consistent covariance matrix for use with the ACOV, HCC, or WHITE option in the MODEL statement and for heteroscedasticity-consistent tests with the TEST statement. The default is HCCMETHOD=0. See the section "Testing for Heteroscedasticity" on page 5575 for details.

I

displays the $(\mathbf{X}'\mathbf{X})^{-1}$ matrix. The inverse of the crossproducts matrix is bordered by the parameter estimates and SSE matrices.

INCLUDE=*n*

forces the first n independent variables listed in the MODEL statement to be included in all models. The selection methods are performed on the other variables in the MODEL statement. The INCLUDE= option is not available with SELECTION=NONE.

INFLUENCE

requests a detailed analysis of the influence of each observation on the estimates and the predicted values. See the section "Influence Statistics" on page 5559 for details.

JP

outputs J_p, the estimated mean square error of prediction for each model selected assuming that the values of the regressors are fixed and that the model is correct to the OUTEST= data set. The J_p statistic is also called the final prediction error (FPE) by Akaike (Nicholson 1948; Lord 1950; Mallows 1967; Darlington 1968; Rothman 1968; Akaike 1969; Hocking 1976; Judge et al. 1980). If SELECTION=ADJRSQ, SELECTION=RSQUARE, or SELECTION=CP is specified, then the J_p statistic is also added to the SubsetSelSummary table.

LACKFIT

performs a lack-of-fit test. See the section "Testing for Lack of Fit" on page 5576 for more information. Refer to Draper and Smith (1981) for a discussion of lack-of-fit tests.

MSE

computes the mean square error for each model selected (Darlington 1968).

MAXSTEP=*n*

specifies the maximum number of steps that are done when SELECTION=FORWARD, SELECTION=BACKWARD, or SELECTION=STEPWISE is used. The default value is the number of independent variables in the model for the FORWARD and BACKWARD methods and three times this number for the stepwise method.

NOINT

suppresses the intercept term that is otherwise included in the model.

NOPRINT

suppresses the normal display of regression results. Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

OUTSEB

outputs the standard errors of the parameter estimates to the OUTEST= data set. The value SEB for the variable _TYPE_ identifies the standard errors. If the RIDGE= or PCOMIT= option is specified, additional observations are included and identified by the values RIDGESEB and IPCSEB, respectively, for the variable _TYPE_. The standard errors for ridge regression estimates and incomplete principal components (IPC) estimates are limited in their usefulness because these estimates are biased. This option is available for all model-selection methods except RSQUARE, ADJRSQ, and CP.

OUTSTB

outputs the standardized parameter estimates as well as the usual estimates to the OUTEST= data set when the RIDGE= or PCOMIT= option is specified. The values RIDGESTB and IPCSTB for the variable _TYPE_ identify ridge regression estimates and IPC estimates, respectively.

OUTVIF

outputs the variance inflation factors (VIF) to the OUTEST= data set when the RIDGE= or PCOMIT= option is specified. The factors are the diagonal elements of the inverse of the correlation matrix of regressors as adjusted by ridge regression or IPC analysis. These observations are identified in the output data set by the values RIDGEVIF and IPCVIF for the variable _TYPE_.

P

calculates predicted values from the input data and the estimated model. The display includes the observation number, the ID variable (if one is specified), the actual and predicted values, and the residual. If the CLI, CLM, or R option is specified, the P option is unnecessary. See the section "Predicted and Residual Values" on page 5531 for more information.

PARTIAL

requests partial regression leverage plots for each regressor. You can use the PARTIALDATA option to obtain a tabular display of the partial regression leverage data. If ODS Graphics is in effect (see the section "ODS Graphics" on page 5589), then these partial plots are produced

in panels with up to six plots per panel. See the section "Influence Statistics" on page 5559 for more information.

PARTIALDATA

requests partial regression leverage data for each regressor. You can request partial regression leverage plots of these data with the PARTIAL option. See the section "Influence Statistics" on page 5559 for more information.

PARTIALR2 < (< TESTS > < SEQTESTS >) >

See the SCORR1 option.

PC

outputs Amemiya's prediction criterion for each model selected (Amemiya 1976; Judge et al. 1980) to the OUTEST= data set. If SELECTION=ADJRSQ, SELECTION=RSQUARE, or SELECTION=CP is specified, then the PC statistic is also added to the SubsetSelSummary table.

PCOMIT=*list*

requests an IPC analysis for each value m in the list. The procedure computes parameter estimates by using all but the last m principal components. Each value of m produces a set of IPC estimates, which is output to the OUTEST= data set. The values of m are saved by the variable _PCOMIT_, and the value of the variable _TYPE_ is set to IPC to identify the estimates. Only nonnegative integers can be specified with the PCOMIT= option.

If you specify the PCOMIT= option, RESTRICT statements are ignored. The PCOMIT= option is ignored if you use the SELECTION= option in the MODEL statement.

PCORR1

displays the squared partial correlation coefficients computed using Type I sum of squares (SS). This is calculated as SS/(SS+SSE), where SSE is the error sum of squares.

PCORR2

displays the squared partial correlation coefficients computed using Type II sums of squares. These are calculated the same way as with the PCORR1 option, except that Type II SS are used instead of Type I SS.

PRESS

outputs the PRESS statistic to the OUTEST= data set. The values of this statistic are saved in the variable _PRESS_. This option is available for all model-selection methods except RSQUARE, ADJRSQ, and CP.

R

requests an analysis of the residuals. The results include everything requested by the P option plus the standard errors of the mean predicted and residual values, the studentized residual, and Cook's D statistic to measure the influence of each observation on the parameter estimates. See the section "Predicted and Residual Values" on page 5531 for more information.

RIDGE=*list*

requests a ridge regression analysis and specifies the values of the ridge constant k (see the section "Computations for Ridge Regression and IPC Analysis" on page 5583). Each value

of k produces a set of ridge regression estimates that are placed in the OUTEST= data set. The values of k are saved by the variable _RIDGE_, and the value of the variable _TYPE_ is set to RIDGE to identify the estimates.

Only nonnegative numbers can be specified with the RIDGE= option. Example 73.5 illustrates this option.

If you specify the RIDGE= option, RESTRICT statements are ignored. The RIDGE= option is ignored if you use the SELECTION= option in the MODEL statement.

RMSE

displays the root mean square error for each model selected.

RSQUARE

has the same effect as the EDF option.

SBC

outputs the SBC statistic for each model selected (Schwarz 1978; Judge et al. 1980) to the OUTEST= data set. If SELECTION=ADJRSQ, SELECTION=RSQUARE, or SELECTION=CP is specified, then the SBC statistic is also added to the SubsetSelSummary table.

SCORR1 < (< TESTS > < SEQTESTS >) >

displays the squared semipartial correlation coefficients computed using Type I sums of squares. This is calculated as SS/SST, where SST is the corrected total SS. If the NOINT option is used, the uncorrected total SS is used in the denominator. The optional arguments TESTS and SEQTESTS request are sequentially added to a model. The F-test values are computed as the Type I sum of squares for the variable in question divided by a mean square error. If you specify the TESTS option, the denominator MSE is the residual mean square for the full model specified in the MODEL statement. If you specify the SEQTESTS option, the denominator MSE is the residual mean square for the model containing all the independent variables that have been added to the model up to and including the variable in question. The TESTS and SEQTESTS options are not supported if you specify model selection methods or the RIDGE or PCOMIT options. Note that the PARTIALR2 option is a synonym for the SCORR1 option.

SCORR2 < (TESTS) >

displays the squared semipartial correlation coefficients computed using Type II sums of squares. These are calculated the same way as with the SCORR1 option, except that Type II SS are used instead of Type I SS. The optional TEST argument requests F tests and p-values as variables are sequentially added to a model. The F-test values are computed as the Type II sum of squares for the variable in question divided by the residual mean square for the full model specified in the MODEL statement. The TESTS option is not supported if you specify model selection methods or the RIDGE or PCOMIT options.

SELECTION=*name*

specifies the method used to select the model, where *name* can be FORWARD (or F), BACKWARD (or B), STEPWISE, MAXR, MINR, RSQUARE, ADJRSQ, CP, or NONE (use the full model). The default method is NONE. See the section "Model-Selection Methods" on page 5523 for a description of each method.

SEQB

produces a sequence of parameter estimates as each variable is entered into the model. This is displayed as a matrix where each row is a set of parameter estimates.

SIGMA=*n*

specifies the true standard deviation of the error term to be used in computing the CP and BIC statistics. If the SIGMA= option is not specified, an estimate from the full model is used. This option is available in the RSQUARE, ADJRSQ, and CP model-selection methods only.

SINGULAR=*n*

tunes the mechanism used to check for singularities. If you specify this option in the MODEL statement, it takes precedence over the SINGULAR= option in the PROC REG statement. The default value is machine dependent but is approximately 1E−7 on most machines. This option is rarely needed. Singularity checking is described in the section "Computational Methods" on page 5584.

SLENTRY=*value*

SLE=*value*

specifies the significance level for entry into the model used in the FORWARD and STEPWISE methods. The defaults are 0.50 for FORWARD and 0.15 for STEPWISE.

SLSTAY=*value*

SLS=*value*

specifies the significance level for staying in the model for the BACKWARD and STEPWISE methods. The defaults are 0.10 for BACKWARD and 0.15 for STEPWISE.

SP

outputs the S_p statistic for each model selected (Hocking 1976) to the OUTEST= data set. If SELECTION=ADJRSQ, SELECTION=RSQUARE, or SELECTION=CP is specified, then the SP statistic is also added to the SubsetSelSummary table.

SPEC

performs a test that the first and second moments of the model are correctly specified. See the section "Testing for Heteroscedasticity" on page 5575 for more information.

SS1

displays the sequential sums of squares (Type I SS) along with the parameter estimates for each term in the model. See Chapter 15, "The Four Types of Estimable Functions," for more information about the different types of sums of squares.

SS2

displays the partial sums of squares (Type II SS) along with the parameter estimates for each term in the model. See the SS1 option also.

SSE

computes the error sum of squares for each model selected.

START=*s*
is used to begin the comparing-and-switching process in the MAXR, MINR, and STEPWISE methods for a model containing the first s independent variables in the MODEL statement, where s is the START value. For these methods, the default is START=0.

For the RSQUARE, ADJRSQ, and CP methods, START=s specifies the smallest number of regressors to be reported in a subset model. For these methods, the default is START=1.

The START= option cannot be used with model-selection methods other than the six described here.

STB
produces standardized regression coefficients. A standardized regression coefficient is computed by dividing a parameter estimate by the ratio of the sample standard deviation of the dependent variable to the sample standard deviation of the regressor.

STOP=*s*
causes PROC REG to stop when it has found the "best" s-variable model, where s is the STOP value. For the RSQUARE, ADJRSQ, and CP methods, STOP=s specifies the largest number of regressors to be reported in a subset model. For the MAXR and MINR methods, STOP=s specifies the largest number of regressors to be included in the model.

The default setting for the STOP= option is the number of variables in the MODEL statement. This option can be used only with the MAXR, MINR, RSQUARE, ADJRSQ, and CP methods.

TOL
produces tolerance values for the estimates. Tolerance for a variable is defined as $1 - R^2$, where R^2 is obtained from the regression of the variable on all other regressors in the model. See the section "Collinearity Diagnostics" on page 5555 for more details.

VIF
produces variance inflation factors with the parameter estimates. Variance inflation is the reciprocal of tolerance. See the section "Collinearity Diagnostics" on page 5555 for more detail.

WHITE
See the HCC option.

XPX
displays the $\mathbf{X}'\mathbf{X}$ crossproducts matrix for the model. The crossproducts matrix is bordered by the $\mathbf{X}'\mathbf{Y}$ and $\mathbf{Y}'\mathbf{Y}$ matrices.

MTEST Statement

< *label:* **>** **MTEST** < *equation* <, . . . , *equation* > > < / *options* > **;**

where each *equation* is a linear function composed of coefficients and variable names. The *label* is optional.

The MTEST statement is used to test hypotheses in multivariate regression models where there are several dependent variables fit to the same regressors. If no equations or options are specified, the MTEST statement tests the hypothesis that all estimated parameters except the intercept are zero.

The hypotheses that can be tested with the MTEST statement are of the form

$$(\mathbf{L}\boldsymbol{\beta} - \mathbf{cj})\mathbf{M} = 0$$

where $\mathbf{L}$ is a linear function on the regressor side, $\boldsymbol{\beta}$ is a matrix of parameters, $\mathbf{c}$ is a column vector of constants, $\mathbf{j}$ is a row vector of ones, and $\mathbf{M}$ is a linear function on the dependent side. The special case where the constants are zero is

$$\mathbf{L}\boldsymbol{\beta}\mathbf{M} = \mathbf{0}$$

See the section "Multivariate Tests" on page 5577 for more details.

Each linear function extends across either the regressor variables or the dependent variables. If the equation is across the dependent variables, then the constant term, if specified, must be zero. The equations for the regressor variables form the $\mathbf{L}$ matrix and $\mathbf{c}$ vector in the preceding formula; the equations for dependent variables form the $\mathbf{M}$ matrix. If no equations for the dependent variables are given, PROC REG uses an identity matrix for $\mathbf{M}$, testing the same hypothesis across all dependent variables. If no equations for the regressor variables are given, PROC REG forms a linear function corresponding to a test that all the nonintercept parameters are zero.

As an example, consider the following statements:

```
model y1 y2 y3=x1 x2 x3;
mtest x1,x2;
mtest y1-y2, y2 -y3, x1;
mtest y1-y2;
```

The first MTEST statement tests the hypothesis that the $X1$ and $X2$ parameters are zero for $Y1$, $Y2$, and $Y3$. In addition, the second MTEST statement tests the hypothesis that the $X1$ parameter is the same for all three dependent variables. For the same model, the third MTEST statement tests the hypothesis that all parameters except the intercept are the same for dependent variables $Y1$ and $Y2$.

You can specify the following options in the MTEST statement:

CANPRINT

displays the canonical correlations for the hypothesis combinations and the dependent variable combinations. If you specify

```
mtest / canprint;
```

the canonical correlations between the regressors and the dependent variables are displayed.

DETAILS

displays the $\mathbf{M}$ matrix and various intermediate calculations.

MSTAT=FAPPROX

MSTAT=EXACT

specifies the method of evaluating the multivariate test statistics. The default is MSTAT=FAPPROX, which specifies that the multivariate tests are evaluated by using the usual approximations based on the F distribution, as discussed in the "Multivariate Tests" section in Chapter 4, "Introduction to Regression Procedures." Alternatively, you can specify MSTAT=EXACT to compute exact p-values for three of the four tests (Wilks' lambda, the Hotelling-Lawley trace, and Roy's greatest root) and an improved F approximation for the fourth (Pillai's trace). While MSTAT=EXACT provides better control of the significance probability for the tests, especially for Roy's greatest root, computations for the exact p-values can be appreciably more demanding, and are in fact infeasible for large problems (many dependent variables). Thus, although MSTAT=EXACT is more accurate for most data, it is not the default method.

PRINT

displays the **H** and **E** matrices.

OUTPUT Statement

OUTPUT < **OUT=***SAS-data-set*>< *keyword=names*> < . . . *keyword=names*> **;**

The OUTPUT statement creates a new SAS data set that saves diagnostic measures calculated after fitting the model. The OUTPUT statement refers to the most recent MODEL statement. At least one *keyword=names* specification is required.

All the variables in the original data set are included in the new data set, along with variables created in the OUTPUT statement. These new variables contain the values of a variety of statistics and diagnostic measures that are calculated for each observation in the data set. If you want to create a permanent SAS data set, you must specify a two-level name (for example, *libref.data-set-name*).

For more information about permanent SAS data sets, refer to the section "SAS Files" in *SAS Language Reference: Concepts*.

The OUTPUT statement cannot be used when a TYPE=CORR, TYPE=COV, or TYPE=SSCP data set is used as the input data set for PROC REG. See the section "Input Data Sets" on page 5508 for more details.

The statistics created in the OUTPUT statement are described in this section. More details are given in the section "Predicted and Residual Values" on page 5531 and the section "Influence Statistics" on page 5559. Also see Chapter 4, "Introduction to Regression Procedures," for definitions of the statistics available from the REG procedure.

You can specify the following options in the OUTPUT statement:

OUT=*SAS data set*

gives the name of the new data set. By default, the procedure uses the DATA*n* convention to name the new data set.

keyword=names

specifies the statistics to include in the output data set and names the new variables that contain the statistics. Specify a keyword for each desired statistic (see the following list of keywords), an equal sign, and the variable or variables to contain the statistic.

In the output data set, the first variable listed after a keyword in the OUTPUT statement contains that statistic for the first dependent variable listed in the MODEL statement; the second variable contains the statistic for the second dependent variable in the MODEL statement, and so on. The list of variables following the equal sign can be shorter than the list of dependent variables in the MODEL statement. In this case, the procedure creates the new names in order of the dependent variables in the MODEL statement.

For example, the following SAS statements create an output data set named b:

```
proc reg data=a;
   model y z=x1 x2;
   output out=b
      p=yhat zhat
      r=yresid zresid;
run;
```

In addition to the variables in the input data set, b contains the following variables:

- yhat, with values that are predicted values of the dependent variable y
- zhat, with values that are predicted values of the dependent variable z
- yresid, with values that are the residual values of y
- zresid, with values that are the residual values of z

You can specify the following keywords in the OUTPUT statement. See the section "Model Fit and Diagnostic Statistics" on page 5557 for computational formulas.

Table 73.5 Keywords for OUTPUT Statement

Keyword	Description
COOKD=*names*	Cook's D influence statistic
COVRATIO=*names*	standard influence of observation on covariance of betas, as discussed in the section "Influence Statistics" on page 5559
DFFITS=*names*	standard influence of observation on predicted value
H=*names*	leverage, $x_i(\mathbf{X}'\mathbf{X})^{-1}x_i'$
LCL=*names*	lower bound of a $100(1-\alpha)\%$ confidence interval for an individual prediction. This includes the variance of the error, as well as the variance of the parameter estimates.
LCLM=*names*	lower bound of a $100(1-\alpha)\%$ confidence interval for the expected value (mean) of the dependent variable
PREDICTED | P=*names*	predicted values
PRESS=*names*	ith residual divided by $(1-h)$, where h is the leverage, and where the model has been refit without the ith observation
RESIDUAL | R=*names*	residuals, calculated as ACTUAL minus PREDICTED

Table 73.5 *continued*

Keyword	Description
RSTUDENT=*names*	a studentized residual with the current observation deleted
STDI=*names*	standard error of the individual predicted value
STDP=*names*	standard error of the mean predicted value
STDR=*names*	standard error of the residual
STUDENT=*names*	studentized residuals, which are the residuals divided by their standard errors
UCL=*names*	upper bound of a $100(1-\alpha)\%$ confidence interval for an individual prediction
UCLM=*names*	upper bound of a $100(1-\alpha)\%$ confidence interval for the expected value (mean) of the dependent variable

PAINT Statement

PAINT < *condition* | **ALLOBS** > < / *options* > **;**

PAINT < *STATUS* | *UNDO* > **;**

The PAINT statement selects observations to be *painted* or highlighted in a scatter plot on line printer output; the PAINT statement is ignored if the LINEPRINTER option is not specified in the PROC REG statement.

All observations that satisfy *condition* are painted using some specific symbol. The PAINT statement does not generate a scatter plot and must be followed by a PLOT statement, which does generate a scatter plot. Several PAINT statements can be used before a PLOT statement, and all prior PAINT statement requests are applied to all later PLOT statements.

The PAINT statement lists the observation numbers of the observations selected, the total number of observations selected, and the plotting symbol used to paint the points.

On a plot, paint symbols take precedence over all other symbols. If any position contains more than one painted point, the paint symbol for the observation plotted last is used.

The PAINT statement cannot be used when a TYPE=CORR, TYPE=COV, or TYPE=SSCP data set is used as the input data set for PROC REG. Also, the PAINT statement cannot be used for models with more than one dependent variable. Note that the syntax for the PAINT statement is the same as the syntax for the REWEIGHT statement.

For detailed examples of painting scatter plots, see the section "Painting Scatter Plots" on page 5543.

Specifying Condition

Condition is used to select observations to be painted. The syntax of *condition* is

variable compare value

or

variable compare value *logical* *variable compare value*

where

variable is one of the following:

- a variable name in the input data set
- OBS**.**, which is the observation number
- *keyword***.**, where *keyword* is a keyword for a statistic requested in the OUTPUT statement

compare is an operator that compares *variable* to *value*. *Compare* can be any one of the following: <, <=, >, >=, =, ^ =. The operators LT, LE, GT, GE, EQ, and NE, respectively, can be used instead of the preceding symbols. Refer to the "Expressions" section in *SAS Language Reference: Concepts* for more information about comparison operators.

value gives an unformatted value of *variable*. Observations are selected to be painted if they satisfy the condition created by *variable compare value*. *Value* can be a number or a character string. If *value* is a character string, it must be eight characters or less and must be enclosed in quotes. In addition, *value* is case-sensitive. In other words, the statements

```
paint name='henry';
```

and

```
paint name='Henry';
```

are not the same.

logical is one of two logical operators. Either AND or OR can be used. To specify AND, use AND or the symbol &. To specify OR, use OR or the symbol |.

Here are some examples of the *variable compare value* form:

```
paint name='Henry';
paint residual.>=20;
paint obs.=99;
```

Here are some examples of the *variable compare value* *logical* *variable compare value* form:

```
paint name='Henry'|name='Mary';
paint residual.>=20 or residual.<=10;
paint obs.>=11 and residual.<=20;
```

Using ALLOBS

Instead of specifying *condition*, the ALLOBS option can be used to select all observations. This is most useful when you want to unpaint all observations. For example,

```
paint allobs / reset;
```

resets the symbols for all observations.

Options in the PAINT Statement

The following options can be used when either a condition is specified, the ALLOBS option is specified, or nothing is specified before the slash. If only an option is listed, the option applies to the observations selected in the previous PAINT statement, *not* to the observations selected by reapplying the condition from the previous PAINT statement. For example, in the statements

```
paint r.>0 / symbol='a';
reweight r.>0;
refit;
paint / symbol='b';
```

the second PAINT statement paints only those observations selected in the first PAINT statement. No additional observations are painted even if, after refitting the model, there are new observations that meet the condition in the first PAINT statement.

NOTE: Options are not available when either the UNDO or STATUS option is used.

You can specify the following options after a slash (/).

NOLIST
suppresses the display of the selected observation numbers. If the NOLIST option is not specified, a list of observations selected is written to the log. The list includes the observation numbers and painting symbol used to paint the points. The total number of observations selected to be painted is also shown.

RESET
changes the painting symbol to the current default symbol, effectively unpainting the observations selected. If you set the default symbol by using the SYMBOL= option in the PLOT statement, the RESET option in the PAINT statement changes the painting symbol to the symbol you specified. Otherwise, the default symbol of '1' is used.

SYMBOL='character'
specifies a painting symbol. If the SYMBOL= option is omitted, the painting symbol is either the one used in the most recent PAINT statement or, if there are no previous PAINT statements, the symbol '@'. For example,

```
paint / symbol='#';
```

changes the painting symbol for the observations selected by the most recent PAINT statement to '#'. As another example,

```
paint temp lt 22 / symbol='c';
```

changes the painting symbol to 'c' for all observations with TEMP<22. In general, the numbers 1, 2, . . . , 9 and the asterisk are not recommended as painting symbols. These symbols are used as default symbols in the PLOT statement, where they represent the number of replicates at a point. If SYMBOL=" is used, no painting is done in the current plot. If SYMBOL=' ' is used, observations are painted with a blank and are no longer seen on the plot.

STATUS and UNDO

Instead of specifying *condition* or the ALLOBS option, you can use the STATUS or UNDO option as follows:

STATUS
lists (in the log) the observation number and plotting symbol of all currently painted observations.

UNDO
undoes changes made by the most recent PAINT statement. Observations might be, but are not necessarily, unpainted. For example:

```
paint obs. <=10 / symbol='a';
\Codecomment{...other interactive statements}
paint obs.=1 / symbol='b';
\Codecomment{...other interactive statements}
paint undo;
```

The last PAINT statement changes the plotting symbol used for observation 1 back to 'a'. If the statement

```
paint / reset;
```

is used instead, observation 1 is unpainted.

PLOT Statement

PLOT < *yvariable***xvariable* > < =*symbol* > < . . . *yvariable***xvariable* > < =*symbol* > < / *options* > ;

The PLOT statement in PROC REG displays scatter plots with *yvariable* on the vertical axis and *xvariable* on the horizontal axis. Line printer plots are generated if the LINEPRINTER option is specified in the PROC REG statement; otherwise, the traditional graphics are created. Points in line printer plots can be marked with *symbols*, while global graphics statements such as GOPTIONS and SYMBOL are used to enhance the traditional graphics. Note that the plots you request by using the PLOT statement are independent of the ODS graphical displays (see the section "ODS Graphics" on page 5589) that are now available in PROC REG.

As with most other interactive statements, the PLOT statement implicitly refits the model. For example, if a PLOT statement is preceded by a REWEIGHT statement, the model is recomputed, and the plot reflects the new model.

If there are multiple MODEL statements preceding a PLOT statement, then the PLOT statement refers to the latest MODEL statement.

The PLOT statement cannot be used when a TYPE=CORR, TYPE=COV, or TYPE=SSCP data set is used as input to PROC REG.

You can specify several PLOT statements for each MODEL statement, and you can specify more than one plot in each PLOT statement.

For detailed examples of using the PLOT statement and its options, see the section "Producing Scatter Plots" on page 5535.

Specifying Yvariables, Xvariables, and Symbol

More than one *yvariable*∗*xvariable* pair can be specified to request multiple plots. The *yvariables* and *xvariables* can be as follows:

- any variables specified in the VAR or MODEL statement before the first RUN statement
- *keyword*., where *keyword* is a regression diagnostic statistic available in the OUTPUT statement (see Table 73.6). For example,

```
plot predicted.*residual.;
```

  generates one plot of the predicted values by the residuals for each dependent variable in the MODEL statement. These statistics can also be plotted against any of the variables in the VAR or MODEL statements.

- the keyword OBS. (the observation number), which can be plotted against any of the preceding variables
- the keyword NPP. or NQQ., which can be used with any of the preceding variables to construct normal P-P or Q-Q plots, respectively (see the section "Construction of Q-Q and P-P Plots" on page 5583 and "Traditional Normal Quantile and Normal Probability Plots" on page 5551 for more information)
- keywords for model fit summary statistics available in the OUTEST= data set with _TYPE_= PARMS (see Table 73.6). A SELECTION= method (other than NONE) must be requested in the MODEL statement for these variables to be plotted. If one member of a *yvariable*∗*xvariable* pair is from the OUTEST= data set, the other member must also be from the OUTEST= data set.

The OUTPUT statement and the OUTEST= option are not required when their keywords are specified in the PLOT statement.

The *yvariable* and *xvariable* specifications can be replaced by a set of variables and statistics enclosed in parentheses. When this occurs, all possible combinations of *yvariable* and *xvariable* are generated. For example, the following two statements are equivalent:

```
plot (y1 y2)*(x1 x2);
plot y1*x1 y1*x2 y2*x1 y2*x2;
```

The statement

```
plot;
```

is equivalent to respecifying the most recent PLOT statement without any options. However, the line printer options COLLECT, HPLOTS=, SYMBOL=, and VPLOTS=, described in the section "Line Printer Plots" on page 5497, apply across PLOT statements and remain in effect if they have been previously specified.

Options used for the traditional graphics are described in the following section; see "Line Printer Plots" on page 5497 for more information.

Traditional Graphics

The display of traditional graphics is described in the following paragraphs, the options are summarized in Table 73.6 and described in the section "Dictionary of PLOT Statement Options" on page 5493, and the section "Traditional Graphics" on page 5547 contains several examples of the graphics output.

Several line printer statements and options are not supported for the traditional graphics. In particular the PAINT statement is disabled, as are the PLOT statement options CLEAR, COLLECT, HPLOTS=, NOCOLLECT, SYMBOL=, and VPLOTS=. To display more than one plot per page or to collect plots from multiple PLOT statements, use the PROC GREPLAY statement (refer to *SAS/GRAPH Software: Reference*). Also note that traditional graphics options are not recognized for line printer plots.

The fitted model equation and a label are displayed in the top margin of the plot; this display can be suppressed with the NOMODEL option. If the label is requested but cannot fit on one line, it is not displayed. The equation and label are displayed on one line when possible; if more lines are required, the label is displayed in the first line with the model equation in successive lines. If displaying the entire equation causes the plot to be unacceptably small, the equation is truncated. Table 73.7 lists options to control the display of the equation. The section "Traditional Graphics for Simple Linear Regression" on page 5547 illustrates the display of the model equation.

Four statistics are displayed by default in the right margin: the number of observations, R^2, the adjusted R^2, and the root mean square error. (See Figure 73.43.) The display of these statistics can be suppressed with the NOSTAT option. You can specify other options to request the display of various statistics in the right margin; see Table 73.7.

A default reference line at zero is displayed if residuals are plotted. If the dependent variable is plotted against the independent variable in a simple linear regression model, the fitted regression line is displayed by default. (See Figure 73.42.) Default reference lines can be suppressed with the NOLINE option; the lines are not displayed if the OVERLAY option is specified.

Specialized plots are requested with special options. For each coefficient, the RIDGEPLOT option plots the ridge estimates against the ridge values k; see the description of the RIDGEPLOT option in the section "Dictionary of PLOT Statement Options" on page 5493 for more details. The CONF option plots $100(1-\alpha)\%$ confidence intervals for the mean while the PRED option plots $100(1-\alpha)\%$ prediction intervals; see the description of these options in the section "Dictionary of PLOT Statement Options" on page 5493 for more details.

If a SELECTION= method is requested, the fitted model equation and the statistics displayed in the margin correspond to the selected model. For the ADJRSQ and CP methods, the selected model is treated as a submodel of the full model. If a CP.*NP. plot is requested, the CHOCKING= and CMALLOWS= options display model selection reference lines; see the descriptions of these options in the section "Dictionary of PLOT Statement Options" on page 5493 and "Traditional Graphics for Variable Selection" on page 5550 for more details.

PLOT Statement variable Keywords

The following table lists the keywords available as PLOT statement *xvariables* and *yvariables*. All keywords have a trailing dot; for example, "*COOKD.*" requests Cook's D statistic. Neither the OUTPUT statement nor the OUTEST= option needs to be specified.

Table 73.6 Keywords for PLOT Statement *xvariables*

Keyword	Description
Diagnostic Statistics	
COOKD.	Cook's D influence statistics
COVRATIO.	standard influence of observation on covariance of betas
DFFITS.	standard influence of observation on predicted value
H.	leverage
LCL.	lower bound of $100(1-\alpha)\%$ confidence interval for individual prediction
LCLM.	lower bound of $100(1-\alpha)\%$ confidence interval for the mean of the dependent variable
PREDICTED. | PRED. | P.	predicted values
PRESS.	residuals from refitting the model with current observation deleted
RESIDUAL. | R.	residuals
RSTUDENT.	studentized residuals with the current observation deleted
STDI.	standard error of the individual predicted value
STDP.	standard error of the mean predicted value
STDR.	standard error of the residual
STUDENT.	residuals divided by their standard errors
UCL.	upper bound of $100(1-\alpha)\%$ confidence interval for individual prediction
UCLM.	upper bound of $100(1-\alpha)\%$ confidence interval for the mean of the dependent variables
Other Keywords Used with Diagnostic Statistics	
NPP.	normal probability-probability plot
NQQ.	normal quantile-quantile plot

Table 73.6 *continued*

Keyword	Description
OBS.	observation number (cannot plot against OUTEST= statistics)
Model Fit Summary Statistics	
ADJRSQ.	adjusted R-square
AIC.	Akaike's information criterion
BIC.	Sawa's Bayesian information criterion
CP.	Mallows' C_p statistic
EDF.	error degrees of freedom
GMSEP.	estimated MSE of prediction, assuming multivariate normality
IN.	number of regressors in the model not including the intercept
JP.	final prediction error
MSE.	mean squared error
NP.	number of parameters in the model (including the intercept)
PC.	Amemiya's prediction criterion
RMSE.	root MSE
RSQ.	R-square
SBC.	SBC statistic
SP.	SP statistic
SSE.	error sum of squares

Summary of PLOT Statement Graphics Options

The following table lists the PLOT statement *options* by function. These *options* are available unless the LINEPRINTER option is specified in the PROC REG statement. For complete descriptions, see the section "Dictionary of PLOT Statement Options" on page 5493.

Table 73.7 Traditional Graphics Options

Option	Description
General Graphics Options	
ANNOTATE=*SAS-data-set*	specifies the annotate data set
CHOCKING=*color*	requests a reference line for C_p model selection criteria
CMALLOWS=*color*	requests a reference line for the C_p model selection criterion
CONF	requests plots of $100(1-\alpha)\%$ confidence intervals for the mean
DESCRIPTION='*string*'	specifies a description for graphics catalog member
NAME='*string*'	names the plot in the graphics catalog
OVERLAY	overlays plots from the same model
PRED	requests plots of $100(1-\alpha)\%$ prediction intervals for individual responses
RIDGEPLOT	requests the ridge trace for ridge regression
Axis and Legend Options	
LEGEND=*LEGENDn*	specifies LEGEND statement to be used
HAXIS=*values*	specifies tick mark values for horizontal axis
VAXIS=*values*	specifies tick mark values for vertical axis

Table 73.7 *continued*

Option	Description
Reference Line Options	
HREF=*values*	specifies reference lines perpendicular to horizontal axis
LHREF=*linetype*	specifies line style for HREF= lines
LLINE=*linetype*	specifies line style for lines displayed by default
LVREF=*linetype*	specifies line style for VREF= lines
NOLINE	suppresses display of any default reference line
VREF=*values*	specifies reference lines perpendicular to vertical axis
Color Options	
CAXIS=*color*	specifies color for axis line and tick marks
CFRAME=*color*	specifies color for frame
CHREF=*color*	specifies color for HREF= lines
CLINE=*color*	specifies color for lines displayed by default
CTEXT=*color*	specifies color for text
CVREF=*color*	specifies color for VREF= lines
Options for Displaying the Fitted Model Equation	
MODELFONT=*font*	specifies font of model equation and model label
MODELHT=*value*	specifies text height of model equation and model label
MODELLAB='*label*'	specifies model label
NOMODEL	suppresses display of the fitted model and the label
Options for Displaying Statistics in the Plot Margin	
AIC	displays Akaike's information criterion
BIC	displays Sawa's Bayesian information criterion
CP	displays Mallows' C_p statistic
EDF	displays the error degrees of freedom
GMSEP	displays the estimated MSE of prediction assuming multivariate normality
IN	displays the number of regressors in the model not including the intercept
JP	displays the J_p statistic
MSE	displays the mean squared error
NOSTAT	suppresses display of the default statistics: the number of observations, R-square, adjusted R-square, and root mean square error
NP	displays the number of parameters in the model including the intercept, if any
PC	displays the PC statistic
SBC	displays the SBC statistic
SP	displays the S_p statistic
SSE	displays the error sum of squares
STATFONT=*font*	specifies font of text displayed in the margin
STATHT=*value*	specifies height of text displayed in the margin

Dictionary of PLOT Statement Options

The following entries describe the PLOT statement *options* in detail. Note that these *options* are available unless you specify the LINEPRINTER option in the PROC REG statement.

AIC

displays Akaike's information criterion in the plot margin.

ANNOTATE=*SAS-data-set*

ANNO=*SAS-data-set*

specifies an input data set that contains appropriate variables for annotation. This applies only to displays created with the current PLOT statement. Refer to *SAS/GRAPH Software: Reference* for more information.

BIC

displays Sawa's Bayesian information criterion in the plot margin.

CAXIS=*color*

CAXES=*color*

CA=*color*

specifies the color for the axes, frame, and tick marks.

CFRAME=*color*

CFR=*color*

specifies the color for filling the area enclosed by the axes and the frame.

CHOCKING=*color*

requests reference lines corresponding to the equations $C_p = p$ and $C_p = 2p - p_{full}$, where p_{full} is the number of parameters in the full model (excluding the intercept) and p is the number of parameters in the subset model (including the intercept). The *color* must be specified; the $C_p = p$ line is solid and the $C_p = 2p - p_{full}$ line is dashed. Only PLOT statements of the form PLOT CP.*NP. produce these lines.

For the purpose of parameter estimation, Hocking (1976) suggests selecting a model where $C_p \leq 2p - p_{full}$. For the purpose of prediction, Hocking suggests the criterion $C_p \leq p$. You can request the single reference line $C_p = p$ with the CMALLOWS= option. If, for example, you specify both CHOCKING=RED and CMALLOWS=BLUE, then the $C_p = 2p - p_{full}$ line is red and the $C_p = p$ line is blue (see Figure 73.45).

CHREF=*color*

CH=*color*

specifies the color for lines requested with the HREF= option.

CLINE=*color*

CL=*color*

specifies the color for lines displayed by default. See the NOLINE option for details.

CMALLOWS=*color*

requests a $C_p = p$ reference line, where p is the number of parameters (including the intercept) in the subset model. The *color* must be specified; the line is solid. Only PLOT statements of the form PLOT CP.*NP. produce this line.

Mallows (1973) suggests that all subset models with C_p small and near p be considered for further study. See the CHOCKING= option for related model-selection criteria.

CONF

is a keyword used as a shorthand option to request plots that include $(100 - \alpha)\%$ confidence intervals for the mean response (see Figure 73.44). The ALPHA= option in the PROC REG or MODEL statement selects the significance level α, which is 0.05 by default. The CONF option is valid for simple regression models only, and is ignored for plots where confidence intervals are inappropriate. The CONF option replaces the CONF95 option; however, the CONF95 option is still supported when the ALPHA= option is not specified. The OVERLAY option is ignored when the CONF option is specified.

CP

displays Mallows' C_p statistic in the plot margin.

CTEXT=*color*

CT=*color*

specifies the color for text including tick mark labels, axis labels, the fitted model label and equation, the statistics displayed in the margin, and legends.

CVREF=*color*

CV=*color*

specifies the color for lines requested with the VREF= option.

DESCRIPTION='*string* **'**

DESC='*string* **'**

specifies a descriptive string, up to 40 characters, that appears in the description field of the PROC GREPLAY master menu.

EDF

displays the error degrees of freedom in the plot margin.

GMSEP

displays the estimated mean square error of prediction in the plot margin. Note that the estimate is calculated under the assumption that both independent and dependent variables have a multivariate normal distribution.

HAXIS=*values*

HA=*values*

specifies tick mark values for the horizontal axis.

HREF=*values*

specifies where reference lines perpendicular to the horizontal axis are to appear.

IN

displays the number of regressors in the model (not including the intercept) in the plot margin.

JP

displays the J_p statistic in the plot margin.

LEGEND=LEGEND*n*

specifies the LEGEND*n* statement to be used. The LEGEND*n* statement is a global graphics statement; refer to *SAS/GRAPH Software: Reference* for more information.

LHREF=*linetype*

LH=*linetype*

specifies the line style for lines requested with the HREF= option. The default *linetype* is 2. Note that LHREF=1 requests a solid line. Refer to *SAS/GRAPH Software: Reference* for a table of available line types.

LLINE=*linetype*

LL=*linetype*

specifies the line style for reference lines displayed by default; see the NOLINE option for details. The default *linetype* is 2. Note that LLINE=1 requests a solid line.

LVREF=*linetype*

LV=*linetype*

specifies the line style for lines requested with the VREF= option. The default *linetype* is 2. Note that LVREF=1 requests a solid line.

MODELFONT=*font*

specifies the font used for displaying the fitted model label and the fitted model equation. Refer to *SAS/GRAPH Software: Reference* for tables of software fonts.

MODELHT=*height*

specifies the text height for the fitted model label and the fitted model equation.

MODELLAB='*label* **'**

specifies the label to be displayed with the fitted model equation. By default, no label is displayed. If the label does not fit on one line, it is not displayed. See the section "Traditional Graphics" on page 5489 for more information.

MSE

displays the mean squared error in the plot margin.

NAME='*string* **'**

specifies a descriptive string, up to eight characters, that appears in the name field of the PROC GREPLAY master menu. The default *string* is REG.

NOLINE

suppresses the display of default reference lines. A default reference line at zero is displayed if residuals are plotted. If the dependent variable is plotted against the independent variable in a simple regression model, then the fitted regression line is displayed by default. Default reference lines are not displayed if the OVERLAY option is specified.

NOMODEL
suppresses the display of the fitted model equation.

NOSTAT
suppresses the display of statistics in the plot margin. By default, the number of observations, R-square, adjusted R-square, and root MSE are displayed.

NP
displays the number of regressors in the model including the intercept, if any, in the plot margin.

OVERLAY
overlays all plots specified in the PLOT statement from the same model on one set of axes. The variables for the first plot label the axes. The procedure automatically scales the axes to fit all of the variables unless the HAXIS= or VAXIS= option is used. Default reference lines are not displayed. A default legend is produced; the LEGEND= option can be used to customize the legend.

PC
displays the PC statistic in the plot margin.

PRED
is a keyword used as a shorthand option to request plots that include $(100-\alpha)\%$ prediction intervals for individual responses (see Figure 73.44). The ALPHA= option in the PROC REG or MODEL statement selects the significance level α, which is 0.05 by default. The PRED option is valid for simple regression models only, and is ignored for plots where prediction intervals are inappropriate. The PRED option replaces the PRED95 option; however, the PRED95 option is still supported when the ALPHA= option is not specified. The OVERLAY option is ignored when the PRED option is specified.

RIDGEPLOT
creates overlaid plots of ridge estimates against ridge values for each coefficient. The points corresponding to the estimates of each coefficient in the plot are connected by lines. For ridge estimates to be computed and plotted, the OUTEST= option must be specified in the PROC REG statement, and the RIDGE=list must be specified in either the PROC REG or MODEL statement.

SBC
displays the SBC statistic in the plot margin.

SP
displays the S_p statistic in the plot margin.

SSE
displays the error sum of squares in the plot margin.

STATFONT=*font*
specifies the font used for displaying the statistics that appear in the plot margin. Refer to *SAS/GRAPH Software: Reference* for tables of software fonts.

STATHT=*height*
specifies the text height of the statistics that appear in the plot margin.

USEALL
specifies that predicted values at data points with missing dependent variable(s) be included on appropriate plots. By default, only points used in constructing the SSCP matrix appear on plots.

VAXIS=*values*

VA=*values*
specifies tick mark values for the vertical axis.

VREF=*values*
specifies where reference lines perpendicular to the vertical axis are to appear.

Line Printer Plots

Line printer plots are requested with the LINEPRINTER option in the PROC REG statement. Points in line printer plots can be marked with *symbols*, which can be specified as a single character enclosed in quotes or the name of any variable in the input data set.

If a character variable is used for the symbol, the first (leftmost) nonblank character in the formatted value of the variable is used as the plotting symbol. If a character in quotes is specified, that character becomes the plotting symbol. If a character is used as the plotting symbol, and if there are different plotting symbols needed at the same point, the symbol '?' is used at that point.

If an unformatted numeric variable is used for the symbol, the symbols '1', '2', ..., '9' are used for variable values 1, 2, ..., 9. For noninteger values, only the integer portion is used as the plotting symbol. For values of 10 or greater, the symbol '*' is used. For negative values, a '?' is used. If a numeric variable is used, and if there is more than one plotting symbol needed at the same point, the sum of the variable values is used at that point. If the sum exceeds 9, the symbol '*' is used.

If a symbol is not specified, the number of replicates at the point is displayed. The symbol '*' is used if there are 10 or more replicates.

If the LINEPRINTER option is used, you can specify the following options in the PLOT statement after a slash (/):

CLEAR
clears any collected scatter plots before plotting begins but does not turn off the COLLECT option. Use this option when you want to begin a new collection with the plots in the current PLOT statement. For more information about collecting plots, see the COLLECT and NOCOLLECT options in this section.

COLLECT
specifies that plots begin to be collected from one PLOT statement to the next and that subsequent plots show an overlay of all collected plots. This option enables you to overlay plots before and after changes to the model or to the data used to fit the model. Plots collected before changes are unaffected by the changes and can be overlaid on later plots. You can

request more than one plot with this option, and you do not need to request the same number of plots in subsequent PLOT statements. If you specify an unequal number of plots, plots in corresponding positions are overlaid. For example, the statements

```
plot residual.*predicted. y*x / collect;
run;
```

produce two plots. If these statements are then followed by

```
plot residual.*x;
run;
```

two plots are again produced. The first plot shows residual against X values overlaid on residual against predicted values. The second plot is the same as that produced by the first PLOT statement.

Axes are scaled for the first plot or plots collected. The axes are not rescaled as more plots are collected.

Once specified, the COLLECT option remains in effect until the NOCOLLECT option is specified.

HPLOTS=*number*

sets the number of scatter plots that can be displayed across the page. The procedure begins with one plot per page. The value of the HPLOTS= option remains in effect until you change it in a later PLOT statement. See the VPLOTS= option for an example.

NOCOLLECT

specifies that the collection of scatter plots ends after adding the plots in the current PLOT statement. PROC REG starts with the NOCOLLECT option in effect. After you specify the NOCOLLECT option, any following PLOT statement produces a new plot that contains only the plots requested by that PLOT statement.

For more information, see the COLLECT option.

OVERLAY

enables requested scatter plots to be superimposed. The axes are scaled so that points on all plots are shown. If the HPLOTS= or VPLOTS= option is set to more than one, the overlaid plot occupies the first position on the page. The OVERLAY option is similar to the COLLECT option in that both options produce superimposed plots. However, OVERLAY superimposes only the plots in the associated PLOT statement; COLLECT superimposes plots across PLOT statements. The OVERLAY option can be used when the COLLECT option is in effect.

SYMBOL='*character***'**

changes the default plotting symbol used for all scatter plots produced in the current and in subsequent PLOT statements. Both SYMBOL=" and SYMBOL=' ' are allowed.

If the SYMBOL= option has not been specified, the default symbol is '1' for positions with one observation, '2' for positions with two observations, and so on. For positions with more than 9 observations, '*' is used. The SYMBOL= option (or a plotting symbol) is needed to avoid any confusion caused by this default convention. Specifying a particular symbol is especially important when either the OVERLAY or COLLECT option is being used.

If you specify the SYMBOL= option and use a number for *character*, that number is used for all points in the plot. For example, the statement

```
plot y*x / symbol='1';
```

produces a plot with the symbol '1' used for all points.

If you specify a plotting symbol and the SYMBOL= option, the plotting symbol overrides the SYMBOL= option. For example, in the statements

```
plot y*x y*v='.' / symbol='*';
```

the symbol used for the plot of Y against X is '*', and a '.' is used for the plot of Y against V.

If a paint symbol is defined with a PAINT statement, the paint symbol takes precedence over both the SYMBOL= option and the default plotting symbol for the PLOT statement.

VPLOTS=*number*

sets the number of scatter plots that can be displayed down the page. The procedure begins with one plot per page. The value of the VPLOTS= option remains in effect until you change it in a later PLOT statement.

For example, to specify a total of six plots per page, with two rows of three plots, use the HPLOTS= and VPLOTS= options as follows:

```
plot y1*x1 y1*x2 y1*x3 y2*x1 y2*x2 y2*x3 /
     hplots=3 vplots=2;
run;
```

PRINT Statement

PRINT < *options* > < *ANOVA* > < *MODELDATA* > **;**

The PRINT statement enables you to interactively display the results of MODEL statement options, produce an ANOVA table, display the data for variables used in the current model, or redisplay the options specified in a MODEL or a previous PRINT statement. In addition, like most other interactive statements in PROC REG, the PRINT statement implicitly refits the model; thus, effects of REWEIGHT statements are seen in the resulting tables. If ODS Graphics is in effect (see the section "ODS Graphics" on page 5589), the PRINT statement also requests the use of the ODS graphical displays associated with the current model.

The following specifications can appear in the PRINT statement:

options

interactively displays the results of MODEL statement options, where *options* is one or more of the following: ACOV, ALL, CLI, CLM, COLLIN, COLLINOINT, CORRB, COVB, DW, I, INFLUENCE, P, PARTIAL, PCORR1, PCORR2, R, SCORR1, SCORR2, SEQB, SPEC, SS1, SS2, STB, TOL, VIF, or XPX. See the section "MODEL Statement" on page 5469 for a description of these options.

ANOVA

produces the ANOVA table associated with the current model. This is either the model specified in the last MODEL statement or the model that incorporates changes made by ADD, DELETE, or REWEIGHT statements after the last MODEL statement.

MODELDATA
displays the data for variables used in the current model.

Use the statement

```
print;
```

to reprint options in the most recently specified PRINT or MODEL statement.

Options that require original data values, such as R or INFLUENCE, cannot be used when a TYPE=CORR, TYPE=COV, or TYPE=SSCP data set is used as the input data set to PROC REG. See the section "Input Data Sets" on page 5508 for more detail.

REFIT Statement

REFIT ;

The REFIT statement causes the current model and corresponding statistics to be recomputed immediately. No output is generated by this statement. The REFIT statement is needed after one or more REWEIGHT statements to cause them to take effect before subsequent PAINT or REWEIGHT statements. This is sometimes necessary when you are using statistical conditions in REWEIGHT statements. For example, consider the following statements:

```
paint student.>2;
plot student.*p.;
reweight student.>2;
refit;
paint student.>2;
plot student.*p.;
```

The second PAINT statement paints any additional observations that meet the condition after deleting observations and refitting the model. The REFIT statement is used because the REWEIGHT statement does not cause the model to be recomputed. In this particular example, the same effect could be achieved by replacing the REFIT statement with a PLOT statement.

Most interactive statements can be used to implicitly refit the model; any plots or statistics produced by these statements reflect changes made to the model and changes made to the data used to compute the model. The two exceptions are the PAINT and REWEIGHT statements, which do not cause the model to be recomputed.

RESTRICT Statement

RESTRICT *equation* < , . . . , *equation* > **;**

A RESTRICT statement is used to place restrictions on the parameter estimates in the MODEL preceding it. More than one RESTRICT statement can follow each MODEL statement. Each RE-

STRICT statement replaces any previous RESTRICT statement. To lift all restrictions on a model, submit a new MODEL statement. If there are several restrictions, separate them with commas. The statement

```
restrict equation1=equation2=equation3;
```

is equivalent to imposing the two restrictions

```
restrict equation1=equation2;
restrict equation2=equation3;
```

Each restriction is written as a linear equation and can be written as

equation

or

equation = equation

The form of each *equation* is

$$c_1 \times variable_1 \pm c_2 \times variable_2 \pm \cdots \pm c_n \times variable_n$$

where the c_j's are constants and the $variable_j$'s are any regressor variables.

When no equal sign appears, the linear combination is set equal to zero. Each variable name mentioned must be a variable in the MODEL statement to which the RESTRICT statement refers. The keyword INTERCEPT can also be used as a variable name, and it refers to the intercept parameter in the regression model.

Note that the parameters associated with the variables are restricted, not the variables themselves. Restrictions should be consistent and not redundant.

Examples of valid RESTRICT statements include the following:

```
restrict x1;
restrict a+b=1;
restrict a=b=c;
restrict a=b, b=c;
restrict 2*f=g+h, intercept+f=0;
restrict f=g=h=intercept;
```

The third and fourth statements in this list produce identical restrictions. You cannot specify

```
restrict f-g=0,
         f-intercept=0,
         g-intercept=1;
```

because the three restrictions are not consistent. If these restrictions are included in a RESTRICT statement, one of the restrict parameters is set to zero and has zero degrees of freedom, indicating that PROC REG is unable to apply a restriction.

The restrictions usually operate even if the model is not of full rank. Check to ensure that DF= -1 for each restriction. In addition, the model DF should decrease by 1 for each restriction.

The parameter estimates are those that minimize the quadratic criterion (SSE) subject to the restrictions. If a restriction cannot be applied, its parameter value and degrees of freedom are listed as zero.

The method used for restricting the parameter estimates is to introduce a Lagrangian parameter for each restriction (Pringle and Rayner 1971). The estimates of these parameters are displayed with test statistics. Note that the t statistic reported for the Lagrangian parameters does not follow a Student's t distribution, but its square follows a beta distribution (LaMotte 1994). The p-value for these parameters is computed using the beta distribution.

The Lagrangian parameter γ measures the sensitivity of the SSE to the restriction constant. If the restriction constant is changed by a small amount ϵ, the SSE is changed by $2\gamma\epsilon$. The t ratio tests the significance of the restrictions. If γ is zero, the restricted estimates are the same as the unrestricted estimates, and a change in the restriction constant in either direction increases the SSE.

RESTRICT statements are ignored if the PCOMIT= or RIDGE= option is specified in the PROC REG statement.

REWEIGHT Statement

REWEIGHT < *condition* | *ALLOBS* > < / *options* > **;**

REWEIGHT < *STATUS* | *UNDO* > **;**

The REWEIGHT statement interactively changes the weights of observations that are used in computing the regression equation. The REWEIGHT statement can change observation weights, or set them to zero, which causes selected observations to be excluded from the analysis. When a REWEIGHT statement sets observation weights to zero, the observations are not deleted from the data set. More than one REWEIGHT statement can be used. The requests from all REWEIGHT statements are applied to the subsequent statements. Each use of the REWEIGHT statement modifies the MODEL label.

The model and corresponding statistics are not recomputed after a REWEIGHT statement. For example, consider the following statements:

```
reweight r.>0;
reweight r.>0;
```

The second REWEIGHT statement does not exclude any additional observations since the model is not recomputed after the first REWEIGHT statement. Either use a REFIT statement to explicitly refit the model, or implicitly refit the model by following the REWEIGHT statement with any other interactive statement except a PAINT statement or another REWEIGHT statement.

The REWEIGHT statement cannot be used if a TYPE=CORR, TYPE=COV, or TYPE=SSCP data set is used as an input data set to PROC REG. Note that the syntax used in the REWEIGHT statement is the same as the syntax in the PAINT statement.

The syntax of the REWEIGHT statement is described in the following sections.

For detailed examples of using this statement, see the section "Reweighting Observations in an Analysis" on page 5569.

Specifying Condition

Condition is used to find observations to be reweighted. The syntax of *condition* is

variable compare value

or

variable compare value *logical* *variable compare value*

where

variable is one of the following:

- a variable name in the input data set
- OBS**.**, which is the observation number
- *keyword***.**, where *keyword* is a keyword for a statistic requested in the OUTPUT statement. The keyword specification is applied to all dependent variables in the model.

compare is an operator that compares *variable* to *value*. *Compare* can be any one of the following: <, <=, >, >=, =, ^ =. The operators LT, LE, GT, GE, EQ, and NE, respectively, can be used instead of the preceding symbols. Refer to the "Expressions" chapter in *SAS Language Reference: Concepts* for more information about comparison operators.

value gives an unformatted value of *variable*. Observations are selected to be reweighted if they satisfy the condition created by *variable compare value*. *Value* can be a number or a character string. If *value* is a character string, it must be eight characters or less and must be enclosed in quotes. In addition, *value* is case-sensitive. In other words, the following two statements are not the same:

```
reweight name='steve';

reweight name='Steve';
```

logical is one of two logical operators. Either AND or OR can be used. To specify AND, use AND or the symbol &. To specify OR, use OR or the symbol |.

Here are some examples of the *variable compare value* form:

```
reweight obs. le 10;
reweight temp=55;
reweight type='new';
```

Here are some example of the *variable compare value* *logical* *variable compare value* form:

```
reweight obs.<=10 and residual.<2;
reweight student.<-2 or student.>2;
reweight name='Mary' | name='Susan';
```

Using ALLOBS

Instead of specifying *condition*, you can use the ALLOBS option to select all observations. This is most useful when you want to restore the original weights of all observations. For example,

```
reweight allobs / reset;
```

resets weights for all observations and uses all observations in the subsequent analysis. Note that

```
reweight allobs;
```

specifies that all observations be excluded from analysis. Consequently, using ALLOBS is useful only if you also use one of the options discussed in the following section.

Options in the REWEIGHT Statement

The following options can be used when either a condition, ALLOBS, or nothing is specified before the slash. If only an option is listed, the option applies to the observations selected in the previous REWEIGHT statement, not to the observations selected by reapplying the condition from the previous REWEIGHT statement. For example, consider the following statements:

```
reweight r.>0 / weight=0.1;
refit;
reweight;
```

The second REWEIGHT statement excludes from the analysis only those observations selected in the first REWEIGHT statement. No additional observations are excluded even if there are new observations that meet the condition in the first REWEIGHT statement.

NOTE: Options are not available when either the UNDO or STATUS option is used.

NOLIST

suppresses the display of the selected observation numbers. If you omit the NOLIST option, a list of observations selected is written to the log.

RESET

resets the observation weights to their original values as defined by the WEIGHT statement or to WEIGHT=1 if no WEIGHT statement is specified. For example,

```
reweight / reset;
```

resets observation weights to the original weights in the data set. If previous REWEIGHT statements have been submitted, this REWEIGHT statement applies only to the observations selected by the previous REWEIGHT statement. Note that, although the RESET option does reset observation weights to their original values, it does not cause the model and corresponding statistics to be recomputed.

WEIGHT=*value*

changes observation weights to the specified nonnegative real number. If you omit the WEIGHT= option, the observation weights are set to zero, and observations are excluded from the analysis. For example:

```
reweight name='Alan';
\Codecomment{...other interactive statements}
reweight / weight=0.5;
```

The first REWEIGHT statement changes weights to zero for all observations with name='Alan', effectively deleting these observations. The subsequent analysis does not include these observations. The second REWEIGHT statement applies only to those observations selected by the previous REWEIGHT statement, and it changes the weights to 0.5 for all the observations with NAME='Alan'. Thus, the next analysis includes all original observations; however, those observations with NAME='Alan' have their weights set to 0.5.

STATUS and UNDO

If you omit *condition* and the ALLOBS options, you can specify one of the following options.

STATUS

writes to the log the observation's number and the weight of all reweighted observations. If an observation's weight has been set to zero, it is reported as deleted. However, the observation is not deleted from the data set, only from the analysis.

UNDO

undoes the changes made by the most recent REWEIGHT statement. Weights might be, but are not necessarily, reset. For example, consider the following statements:

```
reweight student.>2 / weight=0.1;
reweight;
reweight undo;
```

The first REWEIGHT statement sets the weights of observations that satisfy the condition to 0.1. The second REWEIGHT statement sets the weights of the same observations to zero. The third REWEIGHT statement undoes the second, changing the weights back to 0.1.

TEST Statement

< label: > TEST *equation,< ,. . . ,equation> < / option>* **;**

The TEST statement tests hypotheses about the parameters estimated in the preceding MODEL statement. It has the same syntax as the RESTRICT statement except that it supports an option. Each equation specifies a linear hypothesis to be tested. The rows of the hypothesis are separated by commas.

Variable names must correspond to regressors, and each variable name represents the coefficient of the corresponding variable in the model. An optional label is useful to identify each test with a name. The keyword INTERCEPT can be used instead of a variable name to refer to the model's intercept.

The REG procedure performs an F test for the joint hypotheses specified in a single TEST statement. More than one TEST statement can accompany a MODEL statement. The numerator is the usual quadratic form of the estimates; the denominator is the mean squared error. If hypotheses can be represented by

$$\mathbf{L}\boldsymbol{\beta} = \mathbf{c}$$

then the numerator of the F test is

$$\mathbf{Q} = (\mathbf{Lb} - \mathbf{c})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{Lb} - \mathbf{c})$$

divided by degrees of freedom, where $\mathbf{b}$ is the estimate of $\boldsymbol{\beta}$. For example:

```
model y=a1 a2 b1 b2;
aplus: test a1+a2=1;
b1:    test b1=0, b2=0;
b2:    test b1, b2;
```

The last two statements are equivalent; since no constant is specified, zero is assumed.

Note that, when the ACOV, HCC, or WHITE option is specified in the MODEL statement, tests are recomputed using the heteroscedasticity-consistent covariance matrix specified with the HCCMETHOD= option in the MODEL statement (see the section "Testing for Heteroscedasticity" on page 5575).

One option can be specified in the TEST statement after a slash (/):

PRINT
displays intermediate calculations. This includes $\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}'$ bordered by $\mathbf{Lb} - \mathbf{c}$, and $(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}$ bordered by $(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{Lb} - \mathbf{c})$.

VAR Statement

VAR *variables* **;**

The VAR statement is used to include numeric variables in the crossproducts matrix that are not specified in the first MODEL statement.

Variables not listed in MODEL statements before the first RUN statement must be listed in the VAR statement if you want the ability to add them interactively to the model with an ADD statement, to include them in a new MODEL statement, or to plot them in a scatter plot with the PLOT statement.

In addition, if you want to use options in the PROC REG statement and do not want to fit a model to the data (with a MODEL statement), you must use a VAR statement.

WEIGHT Statement

WEIGHT *variable* **;**

A WEIGHT statement names a variable in the input data set with values that are relative weights for a weighted least squares fit. If the weight value is proportional to the reciprocal of the variance for each observation, then the weighted estimates are the best linear unbiased estimates (BLUE).

Values of the weight variable must be nonnegative. If an observation's weight is zero, the observation is deleted from the analysis. If a weight is negative or missing, it is set to zero, and the observation is excluded from the analysis. A more complete description of the WEIGHT statement can be found in Chapter 39, "The GLM Procedure."

Observation weights can be changed interactively with the REWEIGHT statement.

Details: REG Procedure

Missing Values

PROC REG constructs only one crossproducts matrix for the variables in all regressions. If any variable needed for any regression is missing, the observation is excluded from all estimates. If you include variables with missing values in the VAR statement, the corresponding observations are excluded from all analyses, even if you never include the variables in a model. PROC REG assumes that you might want to include these variables after the first RUN statement and deletes observations with missing values.

Input Data Sets

PROC REG does not compute new regressors. For example, if you want a quadratic term in your model, you should create a new variable when you prepare the input data. For example, the statement

```
model y=x1 x1*x1;
```

is not valid. Note that this MODEL statement is valid in the GLM procedure.

The input data set for most applications of PROC REG contains standard rectangular data, but special TYPE=CORR, TYPE=COV, and TYPE=SSCP data sets can also be used. TYPE=CORR and TYPE=COV data sets created by the CORR procedure contain means and standard deviations. In addition, TYPE=CORR data sets contain correlations and TYPE=COV data sets contain covariances. TYPE=SSCP data sets created in previous runs of PROC REG that used the OUTSSCP= option contain the sums of squares and crossproducts of the variables.

See Appendix A, "Special SAS Data Sets," and the "SAS Files" section in *SAS Language Reference: Concepts* for more information about special SAS data sets.

These summary files save CPU time. It takes nk^2 operations (where n=number of observations and k=number of variables) to calculate crossproducts; the regressions are of the order k^3. When n is in the thousands and k is less than 10, you can save 99% of the CPU time by reusing the SSCP matrix rather than recomputing it.

When you want to use a special SAS data set as input, PROC REG must determine the TYPE for the data set. PROC CORR and PROC REG automatically set the type for their output data sets. However, if you create the data set by some other means (such as a DATA step), you must specify its type with the TYPE= data set option. If the TYPE for the data set is not specified when the data set is created, you can specify TYPE= as a data set option in the DATA= option in the PROC REG statement. For example:

```
proc reg data=a(type=corr);
```

When a TYPE=CORR, TYPE=COV, or TYPE=SSCP data set is used with PROC REG, statements and options that require the original data values have no effect. The OUTPUT, PAINT, PLOT, and REWEIGHT statements and the MODEL and PRINT statement options P, R, CLM, CLI, DW, INFLUENCE, and PARTIAL are disabled since the original observations needed to calculate predicted and residual values are not present.

Example Using TYPE=CORR Data Set

The following statements use PROC CORR to produce an input data set for PROC REG. The fitness data for this analysis can be found in Example 73.2.

```
proc corr data=fitness outp=r noprint;
   var Oxygen RunTime Age Weight RunPulse MaxPulse RestPulse;
proc print data=r;
proc reg data=r;
```

```
    model Oxygen=RunTime Age Weight;
run;
```

Since the OUTP= data set from PROC CORR is automatically set to TYPE=CORR, the TYPE= data set option is not required in this example. The data set containing the correlation matrix is displayed by the PRINT procedure as shown in Figure 73.14. Figure 73.15 shows results from the regression that uses the TYPE=CORR data as an input data set.

Figure 73.14 TYPE=CORR Data Set Created by PROC CORR

Obs	_TYPE_	_NAME_	Oxygen	RunTime	Age	Weight	RunPulse	MaxPulse	RestPulse
1	MEAN		47.3758	10.5861	47.6774	77.4445	169.645	173.774	53.4516
2	STD		5.3272	1.3874	5.2114	8.3286	10.252	9.164	7.6194
3	N		31.0000	31.0000	31.0000	31.0000	31.000	31.000	31.0000
4	CORR	Oxygen	1.0000	−0.8622	−0.3046	−0.1628	−0.398	−0.237	−0.3994
5	CORR	RunTime	−0.8622	1.0000	0.1887	0.1435	0.314	0.226	0.4504
6	CORR	Age	−0.3046	0.1887	1.0000	−0.2335	−0.338	−0.433	−0.1641
7	CORR	Weight	−0.1628	0.1435	−0.2335	1.0000	0.182	0.249	0.0440
8	CORR	RunPulse	−0.3980	0.3136	−0.3379	0.1815	1.000	0.930	0.3525
9	CORR	MaxPulse	−0.2367	0.2261	−0.4329	0.2494	0.930	1.000	0.3051
10	CORR	RestPulse	−0.3994	0.4504	−0.1641	0.0440	0.352	0.305	1.0000

Figure 73.15 Regression on TYPE=CORR Data Set

The REG Procedure
Model: MODEL1
Dependent Variable: Oxygen

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	656.27095	218.75698	30.27	<.0001
Error	27	195.11060	7.22632		
Corrected Total	30	851.38154			

Root MSE	2.68818	R-Square	0.7708
Dependent Mean	47.37581	Adj R-Sq	0.7454
Coeff Var	5.67416		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	93.12615	7.55916	12.32	<.0001
RunTime	1	−3.14039	0.36738	−8.55	<.0001
Age	1	−0.17388	0.09955	−1.75	0.0921
Weight	1	−0.05444	0.06181	−0.88	0.3862

The following example uses the saved crossproducts matrix:

```
proc reg data=fitness outsscp=sscp noprint;
   model Oxygen=RunTime Age Weight RunPulse MaxPulse RestPulse;
proc print data=sscp;
proc reg data=sscp;
   model Oxygen=RunTime Age Weight;
run;
```

First, all variables are used to fit the data and create the SSCP data set. Figure 73.16 shows the PROC PRINT display of the SSCP data set. The SSCP data set is then used as the input data set for PROC REG, and a reduced model is fit to the data.

Figure 73.16 TYPE=SSCP Data Set Created by PROC REG

Obs	_TYPE_	_NAME_	Intercept	RunTime	Age	Weight
1	SSCP	Intercept	31.00	328.17	1478.00	2400.78
2	SSCP	RunTime	328.17	3531.80	15687.24	25464.71
3	SSCP	Age	1478.00	15687.24	71282.00	114158.90
4	SSCP	Weight	2400.78	25464.71	114158.90	188008.20
5	SSCP	RunPulse	5259.00	55806.29	250194.00	407745.67
6	SSCP	MaxPulse	5387.00	57113.72	256218.00	417764.62
7	SSCP	RestPulse	1657.00	17684.05	78806.00	128409.28
8	SSCP	Oxygen	1468.65	15356.14	69767.75	113522.26
9	N		31.00	31.00	31.00	31.00

Obs	RunPulse	MaxPulse	RestPulse	Oxygen
1	5259.00	5387.00	1657.00	1468.65
2	55806.29	57113.72	17684.05	15356.14
3	250194.00	256218.00	78806.00	69767.75
4	407745.67	417764.62	128409.28	113522.26
5	895317.00	916499.00	281928.00	248497.31
6	916499.00	938641.00	288583.00	254866.75
7	281928.00	288583.00	90311.00	78015.41
8	248497.31	254866.75	78015.41	70429.86
9	31.00	31.00	31.00	31.00

Figure 73.17 also shows the PROC REG results for the reduced model. (For the PROC REG results for the full model, see Figure 73.29.)

Figure 73.17 Regression on TYPE=SSCP Data Set

The REG Procedure
Model: MODEL1
Dependent Variable: Oxygen

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	656.27095	218.75698	30.27	<.0001
Error	27	195.11060	7.22632		
Corrected Total	30	851.38154			

Root MSE	2.68818	R-Square	0.7708
Dependent Mean	47.37581	Adj R-Sq	0.7454
Coeff Var	5.67416		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	93.12615	7.55916	12.32	<.0001
RunTime	1	−3.14039	0.36738	−8.55	<.0001
Age	1	−0.17388	0.09955	−1.75	0.0921
Weight	1	−0.05444	0.06181	−0.88	0.3862

In the preceding example, the TYPE= data set option is not required since PROC REG sets the OUTSSCP= data set to TYPE=SSCP.

Output Data Sets

OUTEST= Data Set

The OUTEST= specification produces a TYPE=EST output SAS data set containing estimates and optional statistics from the regression models. For each BY group on each dependent variable occurring in each MODEL statement, PROC REG outputs an observation to the OUTEST= data set. The variables output to the data set are as follows:

- the BY variables, if any
- _MODEL_, a character variable containing the label of the corresponding MODEL statement, or MODEL*n* if no label is specified, where *n* is 1 for the first MODEL statement, 2 for the second model statement, and so on
- _TYPE_, a character variable with the value 'PARMS' for every observation

- _DEPVAR_, the name of the dependent variable
- _RMSE_, the root mean squared error or the estimate of the standard deviation of the error term
- Intercept, the estimated intercept, unless the NOINT option is specified
- all the variables listed in any MODEL or VAR statement. Values of these variables are the estimated regression coefficients for the model. A variable that does not appear in the model corresponding to a given observation has a missing value in that observation. The dependent variable in each model is given a value of -1.

If you specify the COVOUT option, the covariance matrix of the estimates is output after the estimates; the _TYPE_ variable is set to the value 'COV' and the names of the rows are identified by the character variable, _NAME_.

If you specify the TABLEOUT option, the following statistics listed by _TYPE_ are added after the estimates:

- STDERR, the standard error of the estimate
- T, the t statistic for testing if the estimate is zero
- PVALUE, the associated p-value
- LnB, the $100(1-\alpha)$ lower confidence limit for the estimate, where n is the nearest integer to $100(1-\alpha)$ and α defaults to 0.05 or is set by using the ALPHA= option in the PROC REG or MODEL statement
- UnB, the $100(1-\alpha)$ upper confidence limit for the estimate

Specifying the option ADJRSQ, AIC, BIC, CP, EDF, GMSEP, JP, MSE, PC, RSQUARE, SBC, SP, or SSE in the PROC REG or MODEL statement automatically outputs these statistics and the model R^2 for each model selected, regardless of the model selection method. Additional variables, in order of occurrence, are as follows:

- _IN_, the number of regressors in the model not including the intercept
- _P_, the number of parameters in the model including the intercept, if any
- _EDF_, the error degrees of freedom
- _SSE_, the error sum of squares, if the SSE option is specified
- _MSE_, the mean squared error, if the MSE option is specified
- _RSQ_, the R^2 statistic
- _ADJRSQ_, the adjusted R^2, if the ADJRSQ option is specified
- _CP_, the C_p statistic, if the CP option is specified

- _SP_, the S_p statistic, if the SP option is specified
- _JP_, the J_p statistic, if the JP option is specified
- _PC_, the PC statistic, if the PC option is specified
- _GMSEP_, the GMSEP statistic, if the GMSEP option is specified
- _AIC_, the AIC statistic, if the AIC option is specified
- _BIC_, the BIC statistic, if the BIC option is specified
- _SBC_, the SBC statistic, if the SBC option is specified

The following statements produce and display the OUTEST= data set. This example uses the population data given in the section "Polynomial Regression" on page 5440. Figure 73.18 through Figure 73.20 show the regression equations and the resulting OUTEST= data set.

```
proc reg data=USPopulation outest=est;
   m1: model Population=Year;
   m2: model Population=Year YearSq;
proc print data=est;
run;
```

Figure 73.18 Regression Output for Model M1

The REG Procedure
Model: m1
Dependent Variable: Population

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	146869	146869	228.92	<.0001
Error	20	12832	641.58160		
Corrected Total	21	159700			

Root MSE	25.32946	R-Square	0.9197
Dependent Mean	94.64800	Adj R-Sq	0.9156
Coeff Var	26.76175		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-2345.85498	161.39279	-14.54	<.0001
Year	1	1.28786	0.08512	15.13	<.0001

Figure 73.19 Regression Output for Model M2

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	159529	79765	8864.19	<.0001
Error	19	170.97193	8.99852		
Corrected Total	21	159700			

Root MSE	2.99975	R-Square	0.9989
Dependent Mean	94.64800	Adj R-Sq	0.9988
Coeff Var	3.16938		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	21631	639.50181	33.82	<.0001
Year	1	-24.04581	0.67547	-35.60	<.0001
YearSq	1	0.00668	0.00017820	37.51	<.0001

Figure 73.20 OUTEST= Data Set

Obs	_MODEL_	_TYPE_	_DEPVAR_	_RMSE_	Intercept	Year	Population	YearSq
1	m1	PARMS	Population	25.3295	-2345.85	1.2879	-1	.
2	m2	PARMS	Population	2.9998	21630.89	-24.0458	-1	.006684346

The following modification of the previous example uses the TABLEOUT and ALPHA= options to obtain additional information in the OUTEST= data set:

```
proc reg data=USPopulation outest=est tableout alpha=0.1;
   m1: model Population=Year/noprint;
   m2: model Population=Year YearSq/noprint;
proc print data=est;
run;
```

Notice that the TABLEOUT option causes standard errors, t statistics, p-values, and confidence limits for the estimates to be added to the OUTEST= data set. Also note that the ALPHA= option is used to set the confidence level at 90%. The OUTEST= data set is shown in Figure 73.21.

Figure 73.21 The OUTEST= Data Set When TABLEOUT Is Specified

Obs	_MODEL_	_TYPE_	_DEPVAR_	_RMSE_	Intercept	Year	Population	YearSq
1	m1	PARMS	Population	25.3295	-2345.85	1.2879	-1	.
2	m1	STDERR	Population	25.3295	161.39	0.0851	.	.
3	m1	T	Population	25.3295	-14.54	15.1300	.	.
4	m1	PVALUE	Population	25.3295	0.00	0.0000	.	.
5	m1	L90B	Population	25.3295	-2624.21	1.1411	.	.
6	m1	U90B	Population	25.3295	-2067.50	1.4347	.	.
7	m2	PARMS	Population	2.9998	21630.89	-24.0458	-1	0.0067
8	m2	STDERR	Population	2.9998	639.50	0.6755	.	0.0002
9	m2	T	Population	2.9998	33.82	-35.5988	.	37.5096
10	m2	PVALUE	Population	2.9998	0.00	0.0000	.	0.0000
11	m2	L90B	Population	2.9998	20525.11	-25.2138	.	0.0064
12	m2	U90B	Population	2.9998	22736.68	-22.8778	.	0.0070

A slightly different OUTEST= data set is created when you use the RSQUARE selection method. The following statements request only the "best" model for each subset size but ask for a variety of model selection statistics, as well as the estimated regression coefficients. An OUTEST= data set is created and displayed. See Figure 73.22 and Figure 73.23 for the results.

```
proc reg data=fitness outest=est;
   model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse
         / selection=rsquare mse jp gmsep cp aic bic sbc b best=1;
proc print data=est;
run;
```

Figure 73.22 PROC REG Output for Physical Fitness Data: Best Models

The REG Procedure
Model: MODEL1
Dependent Variable: Oxygen

R-Square Selection Method

Number in Model	R-Square	C(p)	AIC	BIC	Estimated MSE of Prediction	J(p)	MSE
1	0.7434	13.6988	64.5341	65.4673	8.0546	8.0199	7.53384
2	0.7642	12.3894	63.9050	64.8212	7.9478	7.8621	7.16842
3	0.8111	6.9596	59.0373	61.3127	6.8583	6.7253	5.95669
4	0.8368	4.8800	56.4995	60.3996	6.3984	6.2053	5.34346
5	0.8480	5.1063	56.2986	61.5667	6.4565	6.1782	5.17634
6	0.8487	7.0000	58.1616	64.0748	6.9870	6.5804	5.36825

Number in Model	R-Square	SBC	Parameter Estimates: Intercept	Age	Weight	RunTime
1	0.7434	67.40210	82.42177	.	.	-3.31056
2	0.7642	68.20695	88.46229	-0.15037	.	-3.20395
3	0.8111	64.77326	111.71806	-0.25640	.	-2.82538
4	0.8368	63.66941	98.14789	-0.19773	.	-2.76758
5	0.8480	64.90250	102.20428	-0.21962	-0.07230	-2.68252
6	0.8487	68.19952	102.93448	-0.22697	-0.07418	-2.62865

Number in Model	R-Square	Parameter Estimates: RunPulse	RestPulse	MaxPulse
1	0.7434	.	.	.
2	0.7642	.	.	.
3	0.8111	-0.13091	.	.
4	0.8368	-0.34811	.	0.27051
5	0.8480	-0.37340	.	0.30491
6	0.8487	-0.36963	-0.02153	0.30322

Figure 73.23 PROC PRINT Output for Physical Fitness Data: OUTEST= Data Set

Obs	_MODEL_	_TYPE_	_DEPVAR_	_RMSE_	Intercept	Age	Weight
1	MODEL1	PARMS	Oxygen	2.74478	82.422	.	.
2	MODEL1	PARMS	Oxygen	2.67739	88.462	-0.15037	.
3	MODEL1	PARMS	Oxygen	2.44063	111.718	-0.25640	.
4	MODEL1	PARMS	Oxygen	2.31159	98.148	-0.19773	.
5	MODEL1	PARMS	Oxygen	2.27516	102.204	-0.21962	-0.072302
6	MODEL1	PARMS	Oxygen	2.31695	102.934	-0.22697	-0.074177

Obs	RunTime	RunPulse	RestPulse	Max Pulse	Oxygen	_IN_	_P_	_EDF_	_MSE_
1	-3.31056	.	.	.	-1	1	2	29	7.53384
2	-3.20395	.	.	.	-1	2	3	28	7.16842
3	-2.82538	-0.13091	.	.	-1	3	4	27	5.95669
4	-2.76758	-0.34811	.	0.27051	-1	4	5	26	5.34346
5	-2.68252	-0.37340	.	0.30491	-1	5	6	25	5.17634
6	-2.62865	-0.36963	-0.021534	0.30322	-1	6	7	24	5.36825

Obs	_RSQ_	_CP_	_JP_	_GMSEP_	_AIC_	_BIC_	_SBC_
1	0.74338	13.6988	8.01990	8.05462	64.5341	65.4673	67.4021
2	0.76425	12.3894	7.86214	7.94778	63.9050	64.8212	68.2069
3	0.81109	6.9596	6.72530	6.85833	59.0373	61.3127	64.7733
4	0.83682	4.8800	6.20531	6.39837	56.4995	60.3996	63.6694
5	0.84800	5.1063	6.17821	6.45651	56.2986	61.5667	64.9025
6	0.84867	7.0000	6.58043	6.98700	58.1616	64.0748	68.1995

OUTSSCP= Data Sets

The OUTSSCP= option produces a TYPE=SSCP output SAS data set containing sums of squares and crossproducts. A special row (observation) and column (variable) of the matrix called Intercept contain the number of observations and sums. Observations are identified by the character variable _NAME_. The data set contains all variables used in MODEL statements. You can specify additional variables that you want included in the crossproducts matrix with a VAR statement.

The SSCP data set is used when a large number of observations are explored in many different runs. The SSCP data set can be saved and used for subsequent runs, which are much less expensive since PROC REG never reads the original data again. If you run PROC REG once to create only a SSCP data set, you should list all the variables that you might need in a VAR statement or include all the variables that you might need in a MODEL statement.

The following statements use the fitness data from Example 73.2 to produce an output data set with the OUTSSCP= option. The resulting output is shown in Figure 73.24.

```
proc reg data=fitness outsscp=sscp;
   var Oxygen RunTime Age Weight RestPulse RunPulse MaxPulse;
proc print data=sscp;
run;
```

Since a model is not fit to the data and since the only request is to create the SSCP data set, a MODEL statement is not required in this example. However, since the MODEL statement is not used, the VAR statement is required.

Figure 73.24 SSCP Data Set Created with OUTSSCP= Option: REG Procedure

Obs	_TYPE_	_NAME_	Intercept	Oxygen	RunTime	Age
1	SSCP	Intercept	31.00	1468.65	328.17	1478.00
2	SSCP	Oxygen	1468.65	70429.86	15356.14	69767.75
3	SSCP	RunTime	328.17	15356.14	3531.80	15687.24
4	SSCP	Age	1478.00	69767.75	15687.24	71282.00
5	SSCP	Weight	2400.78	113522.26	25464.71	114158.90
6	SSCP	RestPulse	1657.00	78015.41	17684.05	78806.00
7	SSCP	RunPulse	5259.00	248497.31	55806.29	250194.00
8	SSCP	MaxPulse	5387.00	254866.75	57113.72	256218.00
9	N		31.00	31.00	31.00	31.00

Obs	Weight	RestPulse	RunPulse	MaxPulse
1	2400.78	1657.00	5259.00	5387.00
2	113522.26	78015.41	248497.31	254866.75
3	25464.71	17684.05	55806.29	57113.72
4	114158.90	78806.00	250194.00	256218.00
5	188008.20	128409.28	407745.67	417764.62
6	128409.28	90311.00	281928.00	288583.00
7	407745.67	281928.00	895317.00	916499.00
8	417764.62	288583.00	916499.00	938641.00
9	31.00	31.00	31.00	31.00

Interactive Analysis

PROC REG enables you to change interactively both the model and the data used to compute the model, and to produce and highlight scatter plots. See the section "Using PROC REG Interactively" on page 5450 for an overview of interactive analysis that uses PROC REG. The following statements can be used interactively (without reinvoking PROC REG): ADD, DELETE, MODEL, MTEST, OUTPUT, PAINT, PLOT, PRINT, REFIT, RESTRICT, REWEIGHT, and TEST. All interactive features are disabled if there is a BY statement.

The ADD, DELETE, and REWEIGHT statements can be used to modify the current MODEL. Every use of an ADD, DELETE, or REWEIGHT statement causes the model label to be modified by attaching an additional number to it. This number is the cumulative total of the number of ADD, DELETE, or REWEIGHT statements following the current MODEL statement.

A more detailed explanation of changing the data used to compute the model is given in the section "Reweighting Observations in an Analysis" on page 5569. Extra features for line printer scatter plots are discussed in the section "Line Printer Scatter Plot Features" on page 5534.

The following statements illustrate the usefulness of the interactive features. First, the full regression model is fit to the Class data (see the section "Getting Started: REG Procedure" on page 5436), and Figure 73.25 is produced.

```
ods graphics on;

proc reg data=Class plots(modelLabel only)=ResidualByPredicted;
   model Weight=Age Height;
run;
```

Figure 73.25 Interactive Analysis: Full Model

The REG Procedure
Model: MODEL1
Dependent Variable: Weight

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	7215.63710	3607.81855	27.23	<.0001
Error	16	2120.09974	132.50623		
Corrected Total	18	9335.73684			

Root MSE	11.51114	R-Square	0.7729
Dependent Mean	100.02632	Adj R-Sq	0.7445
Coeff Var	11.50811		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-141.22376	33.38309	-4.23	0.0006
Age	1	1.27839	3.11010	0.41	0.6865
Height	1	3.59703	0.90546	3.97	0.0011

Next, the regression model is reduced by the following statements, and Figure 73.26 is produced.

```
delete age;
print;
run;
```

Figure 73.26 Interactive Analysis: Reduced Model

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	7193.24912	7193.24912	57.08	<.0001
Error	17	2142.48772	126.02869		
Corrected Total	18	9335.73684			

Root MSE	11.22625	R-Square	0.7705
Dependent Mean	100.02632	Adj R-Sq	0.7570
Coeff Var	11.22330		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	-143.02692	32.27459	-4.43	0.0004
Height	1	3.89903	0.51609	7.55	<.0001

Note that the MODEL label has been changed from MODEL1 to MODEL1.1, since the original MODEL has been changed by the delete statement.

When ODS Graphics is enabled, updated plots are produced whenever a PRINT statement is used. The option

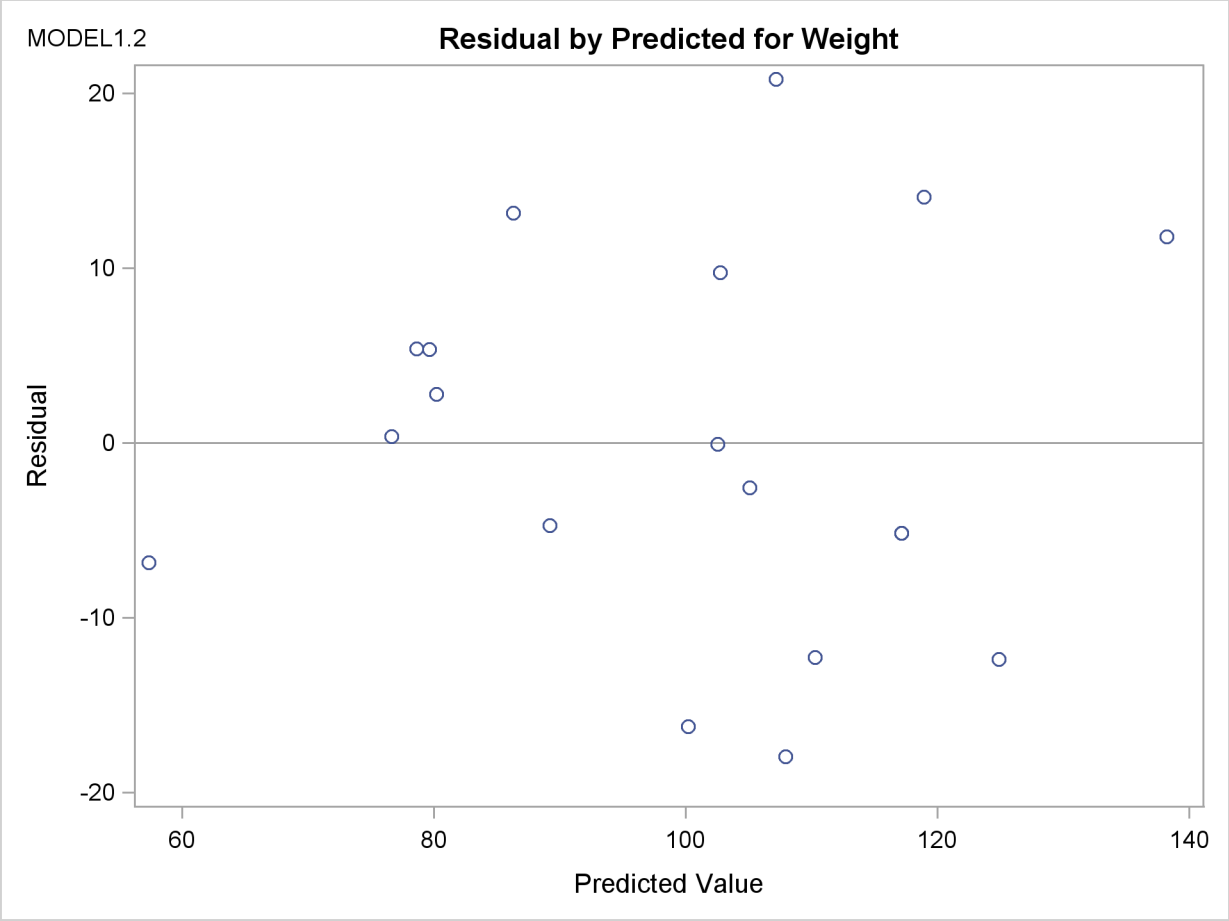
```
plots(modelLabel only)=ResidualByPredicted
```

in the PROC REG statement specifies that the only plot produced is a scatter plot of residuals by predicted values. The MODELLABEL option specifies that the current model label is added to the plot.

The following statements generate a scatter plot of the residuals against the predicted values from the full model. Figure 73.27 is produced, and the scatter plot shows a possible outlier.

```
add age;
print;
run;
```

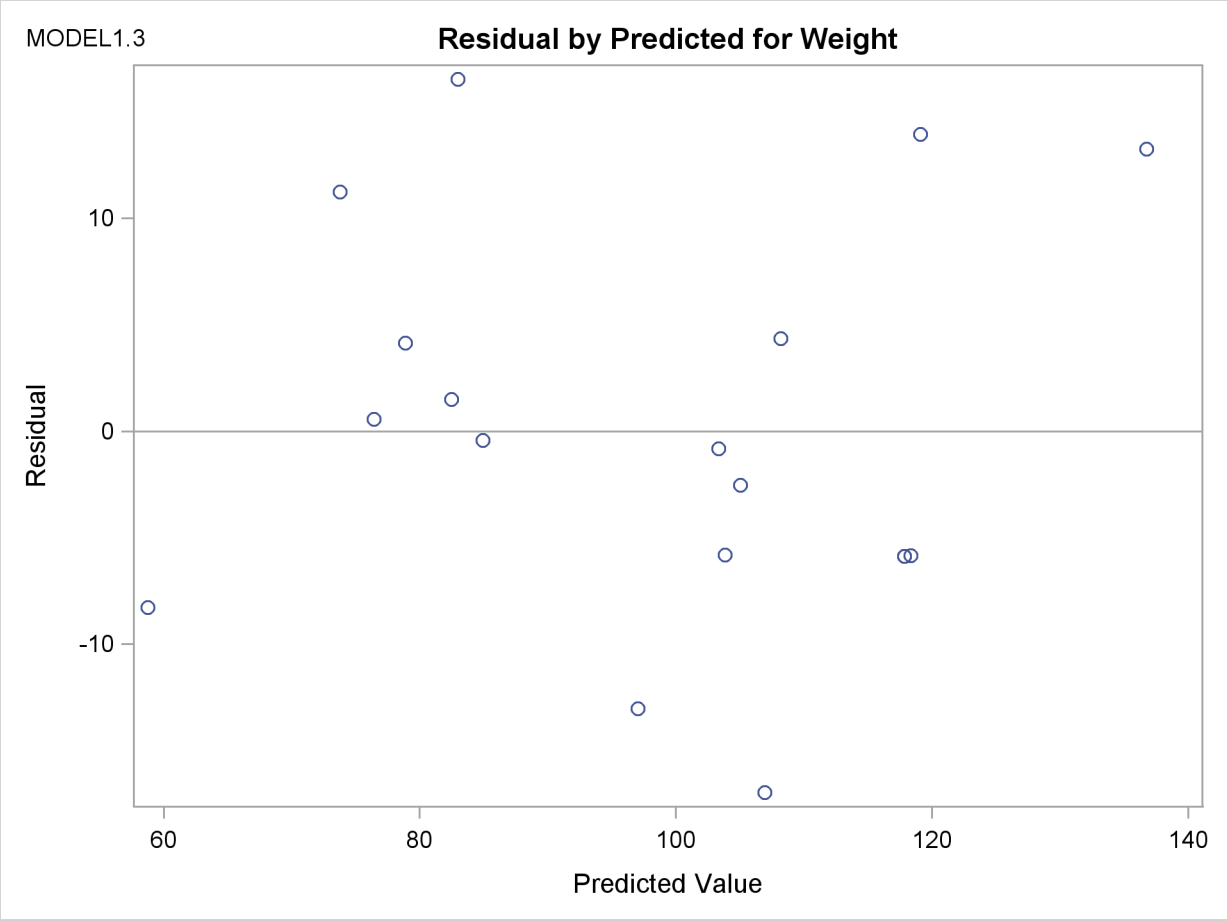
Figure 73.27 Interactive Analysis: Scatter Plot



The following statements delete the observation with the largest residual, refit the regression model, and produce a scatter plot of residuals against predicted values for the refitted model. Figure 73.28 shows the new scatter plot.

```
reweight r.>20;
print;
run;

ods graphics off;
```

Figure 73.28 Interactive Analysis: Scatter Plot



Model-Selection Methods

The nine methods of model selection implemented in PROC REG are specified with the SELECTION= option in the MODEL statement. Each method is discussed in this section.

Full Model Fitted (NONE)

This method is the default and provides no model selection capability. The complete model specified in the MODEL statement is used to fit the model. For many regression analyses, this might be the only method you need.

Forward Selection (FORWARD)

The forward-selection technique begins with no variables in the model. For each of the independent variables, the FORWARD method calculates F statistics that reflect the variable's contribution to the model if it is included. The p-values for these F statistics are compared to the SLENTRY=

value that is specified in the MODEL statement (or to 0.50 if the SLENTRY= option is omitted). If no F statistic has a significance level greater than the SLENTRY= value, the FORWARD selection stops. Otherwise, the FORWARD method adds the variable that has the largest F statistic to the model. The FORWARD method then calculates F statistics again for the variables still remaining outside the model, and the evaluation process is repeated. Thus, variables are added one by one to the model until no remaining variable produces a significant F statistic. Once a variable is in the model, it stays.

Backward Elimination (BACKWARD)

The backward elimination technique begins by calculating F statistics for a model, including all of the independent variables. Then the variables are deleted from the model one by one until all the variables remaining in the model produce F statistics significant at the SLSTAY= level specified in the MODEL statement (or at the 0.10 level if the SLSTAY= option is omitted). At each step, the variable showing the smallest contribution to the model is deleted.

Stepwise (STEPWISE)

The stepwise method is a modification of the forward-selection technique and differs in that variables already in the model do not necessarily stay there. As in the forward-selection method, variables are added one by one to the model, and the F statistic for a variable to be added must be significant at the SLENTRY= level. After a variable is added, however, the stepwise method looks at all the variables already included in the model and deletes any variable that does not produce an F statistic significant at the SLSTAY= level. Only after this check is made and the necessary deletions are accomplished can another variable be added to the model. The stepwise process ends when none of the variables outside the model has an F statistic significant at the SLENTRY= level and every variable in the model is significant at the SLSTAY= level, or when the variable to be added to the model is the one just deleted from it.

Maximum R^2 Improvement (MAXR)

The maximum R^2 improvement technique does not settle on a single model. Instead, it tries to find the "best" one-variable model, the "best" two-variable model, and so forth, although it is not guaranteed to find the model with the largest R^2 for each size.

The MAXR method begins by finding the one-variable model producing the highest R^2. Then another variable, the one that yields the greatest increase in R^2, is added. Once the two-variable model is obtained, each of the variables in the model is compared to each variable not in the model. For each comparison, the MAXR method determines if removing one variable and replacing it with the other variable increases R^2. After comparing all possible switches, the MAXR method makes the switch that produces the largest increase in R^2. Comparisons begin again, and the process continues until the MAXR method finds that no switch could increase R^2. Thus, the two-variable model achieved is considered the "best" two-variable model the technique can find. Another variable is then added to the model, and the comparing-and-switching process is repeated to find the "best" three-variable model, and so forth.

The difference between the STEPWISE method and the MAXR method is that all switches are evaluated before any switch is made in the MAXR method. In the STEPWISE method, the "worst" variable might be removed without considering what adding the "best" remaining variable might accomplish. The MAXR method might require much more computer time than the STEPWISE method.

Minimum R^2 (MINR) Improvement

The MINR method closely resembles the MAXR method, but the switch chosen is the one that produces the smallest increase in R^2. For a given number of variables in the model, the MAXR and MINR methods usually produce the same "best" model, but the MINR method considers more models of each size.

R^2 Selection (RSQUARE)

The RSQUARE method finds subsets of independent variables that best predict a dependent variable by linear regression in the given sample. You can specify the largest and smallest number of independent variables to appear in a subset and the number of subsets of each size to be selected. The RSQUARE method can efficiently perform all possible subset regressions and display the models in decreasing order of R^2 magnitude within each subset size. Other statistics are available for comparing subsets of different sizes. These statistics, as well as estimated regression coefficients, can be displayed or output to a SAS data set.

The subset models selected by the RSQUARE method are optimal in terms of R^2 for the given sample, but they are not necessarily optimal for the population from which the sample is drawn or for any other sample for which you might want to make predictions. If a subset model is selected on the basis of a large R^2 value or any other criterion commonly used for model selection, then all regression statistics computed for that model under the assumption that the model is given a priori, including all statistics computed by PROC REG, are biased.

While the RSQUARE method is a useful tool for exploratory model building, no statistical method can be relied on to identify the "true" model. Effective model building requires substantive theory to suggest relevant predictors and plausible functional forms for the model.

The RSQUARE method differs from the other selection methods in that RSQUARE always identifies the model with the largest R^2 for each number of variables considered. The other selection methods are not guaranteed to find the model with the largest R^2. The RSQUARE method requires much more computer time than the other selection methods, so a different selection method such as the STEPWISE method is a good choice when there are many independent variables to consider.

Adjusted R^2 Selection (ADJRSQ)

This method is similar to the RSQUARE method, except that the adjusted R^2 statistic is used as the criterion for selecting models, and the method finds the models with the highest adjusted R^2 within the range of sizes.

Mallows' C_p Selection (CP)

This method is similar to the ADJRSQ method, except that Mallows' C_p statistic is used as the criterion for model selection. Models are listed in ascending order of C_p.

Additional Information about Model-Selection Methods

If the RSQUARE or STEPWISE procedure (as documented in *SAS User's Guide: Statistics, Version 5 Edition*) is requested, PROC REG with the appropriate model-selection method is actually used.

Reviews of model-selection methods by Hocking (1976) and Judge et al. (1980) describe these and other variable-selection methods.

Criteria Used in Model-Selection Methods

When many significance tests are performed, each at a level of, for example, 5%, the overall probability of rejecting at least one true null hypothesis is much larger than 5%. If you want to guard against including any variables that do not contribute to the predictive power of the model in the population, you should specify a very small SLE= significance level for the FORWARD and STEPWISE methods and a very small SLS= significance level for the BACKWARD and STEPWISE methods.

In most applications, many of the variables considered have some predictive power, however small. If you want to choose the model that provides the best prediction computed using the sample estimates, you need only to guard against estimating more parameters than can be reliably estimated with the given sample size, so you should use a moderate significance level, perhaps in the range of 10% to 25%.

In addition to R^2, the C_p statistic is displayed for each model generated in the model-selection methods. The C_p statistic is proposed by Mallows (1973) as a criterion for selecting a model. It is a measure of total squared error defined as

$$C_p = \frac{SSE_p}{s^2} - (N - 2p)$$

where s^2 is the MSE for the full model, and SSE_p is the sum-of-squares error for a model with p parameters including the intercept, if any. If C_p is plotted against p, Mallows recommends the model where C_p first approaches p. When the right model is chosen, the parameter estimates are unbiased, and this is reflected in C_p near p. For further discussion, refer to Daniel and Wood (1980).

The adjusted R^2 statistic is an alternative to R^2 that is adjusted for the number of parameters in the model. The adjusted R^2 statistic is calculated as

$$\text{ADJRSQ} = 1 - \frac{(n-i)(1-R^2)}{n-p}$$

where n is the number of observations used in fitting the model, and i is an indicator variable that is 1 if the model includes an intercept, and 0 otherwise.

Limitations in Model-Selection Methods

The use of model-selection methods can be time-consuming in some cases because there is no built-in limit on the number of independent variables, and the calculations for a large number of independent variables can be lengthy. The recommended limit on the number of independent variables for the MINR method is $20 + i$, where i is the value of the INCLUDE= option.

For the RSQUARE, ADJRSQ, or CP method, with a large value of the BEST= option, adding one more variable to the list from which regressors are selected might significantly increase the CPU time. Also, the time required for the analysis is highly dependent on the data and on the values of the BEST=, START=, and STOP= options.

Parameter Estimates and Associated Statistics

The following example uses the fitness data from Example 73.2. Figure 73.30 shows the parameter estimates and the tables from the SS1, SS2, STB, CLB, COVB, and CORRB options:

```
proc reg data=fitness;
   model Oxygen=RunTime Age Weight RunPulse MaxPulse RestPulse
         / ss1 ss2 stb clb covb corrb;
run;
```

The procedure first displays an analysis of variance table (Figure 73.29). The F statistic for the overall model is significant, indicating that the model explains a significant portion of the variation in the data.

Figure 73.29 ANOVA Table

The REG Procedure
Model: MODEL1
Dependent Variable: Oxygen

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	722.54361	120.42393	22.43	<.0001
Error	24	128.83794	5.36825		
Corrected Total	30	851.38154			

Root MSE	2.31695	R-Square	0.8487
Dependent Mean	47.37581	Adj R-Sq	0.8108
Coeff Var	4.89057		

The procedure next displays parameter estimates and some associated statistics (Figure 73.30). First, the estimates are shown, followed by their standard errors. The next two columns of the table contain the t statistics and the corresponding probabilities for testing the null hypothesis that the parameter is not significantly different from zero. These probabilities are usually referred to as p-values. For example, the Intercept term in the model is estimated to be 102.9 and is significantly different from zero. The next two columns of the table are the result of requesting the SS1 and SS2 options, and they show sequential and partial sums of squares (SS) associated with each variable. The standardized estimates (produced by the STB option) are the parameter estimates that result when all variables are standardized to a mean of 0 and a variance of 1. These estimates are computed by multiplying the original estimates by the standard deviation of the regressor (independent) variable and then dividing by the standard deviation of the dependent variable. The CLB option adds the upper and lower 95% confidence limits for the parameter estimates; the α level can be changed by specifying the ALPHA= option in the PROC REG or MODEL statement.

Figure 73.30 SS1, SS2, STB, CLB, COVB, and CORRB Options: Parameter Estimates

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Type I SS
Intercept	1	102.93448	12.40326	8.30	<.0001	69578
RunTime	1	-2.62865	0.38456	-6.84	<.0001	632.90010
Age	1	-0.22697	0.09984	-2.27	0.0322	17.76563
Weight	1	-0.07418	0.05459	-1.36	0.1869	5.60522
RunPulse	1	-0.36963	0.11985	-3.08	0.0051	38.87574
MaxPulse	1	0.30322	0.13650	2.22	0.0360	26.82640
RestPulse	1	-0.02153	0.06605	-0.33	0.7473	0.57051

Parameter Estimates

Variable	DF	Type II SS	Standardized Estimate	95% Confidence Limits	
Intercept	1	369.72831	0	77.33541	128.53355
RunTime	1	250.82210	-0.68460	-3.42235	-1.83496
Age	1	27.74577	-0.22204	-0.43303	-0.02092
Weight	1	9.91059	-0.11597	-0.18685	0.03850
RunPulse	1	51.05806	-0.71133	-0.61699	-0.12226
MaxPulse	1	26.49142	0.52161	0.02150	0.58493
RestPulse	1	0.57051	-0.03080	-0.15786	0.11480

The final two tables are produced as a result of requesting the COVB and CORRB options (Figure 73.31). These tables show the estimated covariance matrix of the parameter estimates, and the estimated correlation matrix of the estimates.

Figure 73.31 SS1, SS2, STB, CLB, COVB, and CORRB Options: Covariances and Correlations

Covariance of Estimates

Variable	Intercept	RunTime	Age	Weight
Intercept	153.84081152	0.7678373769	-0.902049478	-0.178237818
RunTime	0.7678373769	0.1478880839	-0.014191688	-0.004417672
Age	-0.902049478	-0.014191688	0.009967521	0.0010219105
Weight	-0.178237818	-0.004417672	0.0010219105	0.0029804131
RunPulse	0.280796516	-0.009047784	-0.001203914	0.0009644683
MaxPulse	-0.832761667	0.0046249498	0.0035823843	-0.001372241
RestPulse	-0.147954715	-0.010915224	0.0014897532	0.0003799295

Covariance of Estimates

Variable	RunPulse	MaxPulse	RestPulse
Intercept	0.280796516	-0.832761667	-0.147954715
RunTime	-0.009047784	0.0046249498	-0.010915224
Age	-0.001203914	0.0035823843	0.0014897532
Weight	0.0009644683	-0.001372241	0.0003799295
RunPulse	0.0143647273	-0.014952457	-0.000764507
MaxPulse	-0.014952457	0.0186309364	0.0003425724
RestPulse	-0.000764507	0.0003425724	0.0043631674

Correlation of Estimates

Variable	Intercept	RunTime	Age	Weight
Intercept	1.0000	0.1610	-0.7285	-0.2632
RunTime	0.1610	1.0000	-0.3696	-0.2104
Age	-0.7285	-0.3696	1.0000	0.1875
Weight	-0.2632	-0.2104	0.1875	1.0000
RunPulse	0.1889	-0.1963	-0.1006	0.1474
MaxPulse	-0.4919	0.0881	0.2629	-0.1842
RestPulse	-0.1806	-0.4297	0.2259	0.1054

Correlation of Estimates

Variable	RunPulse	MaxPulse	RestPulse
Intercept	0.1889	-0.4919	-0.1806
RunTime	-0.1963	0.0881	-0.4297
Age	-0.1006	0.2629	0.2259
Weight	0.1474	-0.1842	0.1054
RunPulse	1.0000	-0.9140	-0.0966
MaxPulse	-0.9140	1.0000	0.0380
RestPulse	-0.0966	0.0380	1.0000

For further discussion of the parameters and statistics, see the section "Displayed Output" on page 5584, and Chapter 4, "Introduction to Regression Procedures."

Predicted and Residual Values

The display of the predicted values and residuals is controlled by the P, R, CLM, and CLI options in the MODEL statement. The P option causes PROC REG to display the observation number, the ID value (if an ID statement is used), the actual value, the predicted value, and the residual. The R, CLI, and CLM options also produce the items under the P option. Thus, P is unnecessary if you use one of the other options.

The R option requests more detail, especially about the residuals. The standard errors of the mean predicted value and the residual are displayed. The studentized residual, which is the residual divided by its standard error, is both displayed and plotted. A measure of influence, Cook's D, is displayed. Cook's D measures the change to the estimates that results from deleting each observation (Cook 1977, 1979). This statistic is very similar to DFFITS.

The CLM option requests that PROC REG display the $100(1-\alpha)\%$ lower and upper confidence limits for the mean predicted values. This accounts for the variation due to estimating the parameters only. If you want a $100(1-\alpha)\%$ confidence interval for observed values, then you can use the CLI option, which adds in the variability of the error term. The α level can be specified with the ALPHA= option in the PROC REG or MODEL statement.

You can use these statistics in PLOT and PAINT statements. This is useful in performing a variety of regression diagnostics. For definitions of the statistics produced by these options, see Chapter 4, "Introduction to Regression Procedures."

The following statements use the U.S. population data found in the section "Polynomial Regression" on page 5440. The results are shown in Figure 73.32 and Figure 73.33.

```
data USPop2;
   input Year @@;
   YearSq=Year*Year;
   datalines;
2010 2020 2030
;
data USPop2;
   set USPopulation USPop2;

proc reg data=USPop2;
   id Year;
   model Population=Year YearSq / r cli clm;
run;
```

Figure 73.32 Regression Using the R, CLI, and CLM Options

The REG Procedure
Model: MODEL1
Dependent Variable: Population

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	159529	79765	8864.19	<.0001
Error	19	170.97193	8.99852		
Corrected Total	21	159700			

Root MSE	2.99975	R-Square	0.9989
Dependent Mean	94.64800	Adj R-Sq	0.9988
Coeff Var	3.16938		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	21631	639.50181	33.82	<.0001
Year	1	-24.04581	0.67547	-35.60	<.0001
YearSq	1	0.00668	0.00017820	37.51	<.0001

Figure 73.33 Regression Using the R, CLI, and CLM Options

The REG Procedure
Model: MODEL1
Dependent Variable: Population

Output Statistics

Obs	Year	Dependent Variable	Predicted Value	Std Error Mean Predict	95% CL Mean		95% CL Predict	
1	1790	3.9290	6.2127	1.7565	2.5362	9.8892	−1.0631	13.4884
2	1800	5.3080	5.7226	1.4560	2.6751	8.7701	−1.2565	12.7017
3	1810	7.2390	6.5694	1.2118	4.0331	9.1057	−0.2021	13.3409
4	1820	9.6380	8.7531	1.0305	6.5963	10.9100	2.1144	15.3918
5	1830	12.8660	12.2737	0.9163	10.3558	14.1916	5.7087	18.8386
6	1840	17.0690	17.1311	0.8650	15.3207	18.9415	10.5968	23.6655
7	1850	23.1910	23.3254	0.8613	21.5227	25.1281	16.7932	29.8576
8	1860	31.4430	30.8566	0.8846	29.0051	32.7080	24.3107	37.4024
9	1870	39.8180	39.7246	0.9163	37.8067	41.6425	33.1597	46.2896
10	1880	50.1550	49.9295	0.9436	47.9545	51.9046	43.3476	56.5114
11	1890	62.9470	61.4713	0.9590	59.4641	63.4785	54.8797	68.0629
12	1900	75.9940	74.3499	0.9590	72.3427	76.3571	67.7583	80.9415
13	1910	91.9720	88.5655	0.9436	86.5904	90.5405	81.9836	95.1473
14	1920	105.7100	104.1178	0.9163	102.2000	106.0357	97.5529	110.6828
15	1930	122.7750	121.0071	0.8846	119.1556	122.8585	114.4612	127.5529
16	1940	131.6690	139.2332	0.8613	137.4305	141.0359	132.7010	145.7654
17	1950	151.3250	158.7962	0.8650	156.9858	160.6066	152.2618	165.3306
18	1960	179.3230	179.6961	0.9163	177.7782	181.6139	173.1311	186.2610
19	1970	203.2110	201.9328	1.0305	199.7759	204.0896	195.2941	208.5715
20	1980	226.5420	225.5064	1.2118	222.9701	228.0427	218.7349	232.2779
21	1990	248.7100	250.4168	1.4560	247.3693	253.4644	243.4378	257.3959
22	2000	281.4220	276.6642	1.7565	272.9877	280.3407	269.3884	283.9400
23	2010	.	304.2484	2.1073	299.8377	308.6591	296.5754	311.9214
24	2020	.	333.1695	2.5040	327.9285	338.4104	324.9910	341.3479
25	2030	.	363.4274	2.9435	357.2665	369.5883	354.6310	372.2238

Output Statistics

Obs	Year	Residual	Std Error Residual	Student Residual	−2−1 0 1 2	Cook's D
1	1790	−2.2837	2.432	−0.939	| *| |	0.153
2	1800	−0.4146	2.623	−0.158	| | |	0.003
3	1810	0.6696	2.744	0.244	| | |	0.004
4	1820	0.8849	2.817	0.314	| | |	0.004
5	1830	0.5923	2.856	0.207	| | |	0.001
6	1840	−0.0621	2.872	−0.0216	| | |	0.000
7	1850	−0.1344	2.873	−0.0468	| | |	0.000
8	1860	0.5864	2.866	0.205	| | |	0.001
9	1870	0.0934	2.856	0.0327	| | |	0.000
10	1880	0.2255	2.847	0.0792	| | |	0.000

Figure 73.33 *continued*

The REG Procedure
Model: MODEL1
Dependent Variable: Population

Output Statistics

Obs	Year	Residual	Std Error Residual	Student Residual	-2-1 0 1 2	Cook's D
11	1890	1.4757	2.842	0.519	| |* |	0.010
12	1900	1.6441	2.842	0.578	| |* |	0.013
13	1910	3.4065	2.847	1.196	| |** |	0.052
14	1920	1.5922	2.856	0.557	| |* |	0.011
15	1930	1.7679	2.866	0.617	| |* |	0.012
16	1940	-7.5642	2.873	-2.632	| *****| |	0.208
17	1950	-7.4712	2.872	-2.601	| *****| |	0.205
18	1960	-0.3731	2.856	-0.131	| | |	0.001
19	1970	1.2782	2.817	0.454	| | |	0.009
20	1980	1.0356	2.744	0.377	| | |	0.009
21	1990	-1.7068	2.623	-0.651	| *| |	0.044
22	2000	4.7578	2.432	1.957	| |*** |	0.666
23	2010	.	.	.		.
24	2020	.	.	.		.
25	2030	.	.	.		.

After producing the usual analysis of variance and parameter estimates tables (Figure 73.32), the procedure displays the results of requesting the options for predicted and residual values (Figure 73.33). For each observation, the requested information is shown. Note that the ID variable is used to identify each observation. Also note that, for observations with missing dependent variables, the predicted value, standard error of the predicted value, and confidence intervals for the predicted value are still available.

The columnar print plot of studentized residuals and Cook's D statistics are displayed as a result of requesting the R option. In the plot of studentized residuals, the large number of observations with absolute values greater than two indicates an inadequate model. You can use ODS Graphics to obtain high-resolution plots of studentized residuals by predicted values or leverage; see Example 73.1 for a similar example.

Line Printer Scatter Plot Features

This section discusses the special options available with line printer scatter plots. Detailed examples of traditional graphics and options are given in the section "Traditional Graphics" on page 5547.

Producing Scatter Plots

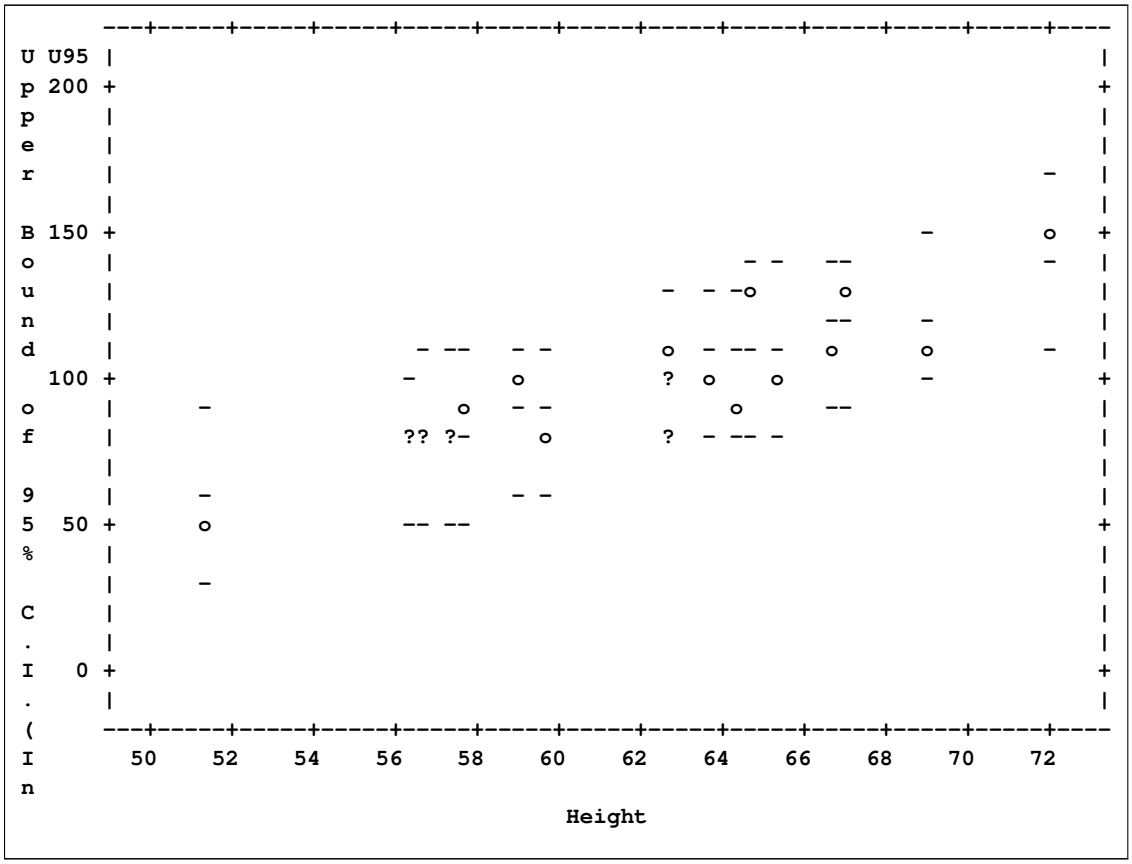
The interactive PLOT statement available in PROC REG enables you to look at scatter plots of data and diagnostic statistics. These plots can help you to evaluate the model and detect outliers in your data. Several options enable you to place multiple plots on a single page, superimpose plots, and collect plots to be overlaid by later plots. The PAINT statement can be used to highlight points on a plot. See the section "Painting Scatter Plots" on page 5543 for more information about painting.

The Class data set introduced in the section "Simple Linear Regression" on page 5436 is used in the following examples.

You can superimpose several plots with the OVERLAY option. With the following statements, a plot of Weight against Height is overlaid with plots of the predicted values and the 95% prediction intervals. The model on which the statistics are based is the full model including Height and Age. These statements produce the plot in Figure 73.34:

```
proc reg data=Class lineprinter;
   model Weight=Height Age / noprint;
   plot (ucl. lcl. p.)*Height='-' Weight*Height
        / overlay symbol='o';
run;
```

Figure 73.34 Scatter Plot Showing Data, Predicted Values, and Confidence Limits



In this plot, the data values are marked with the symbol 'o' and the predicted values and prediction interval limits are labeled with the symbol '-'. The plot is scaled to accommodate the points from all plots. This is an important difference from the COLLECT option, which does not rescale plots after the first plot or plots are collected. You could separate the overlaid plots by using the following statements:

```
plot;
run;
```

This places each of the four plots on a separate page, while the statements

```
plot / overlay;
run;
```

repeat the previous overlaid plot. In general, the statement

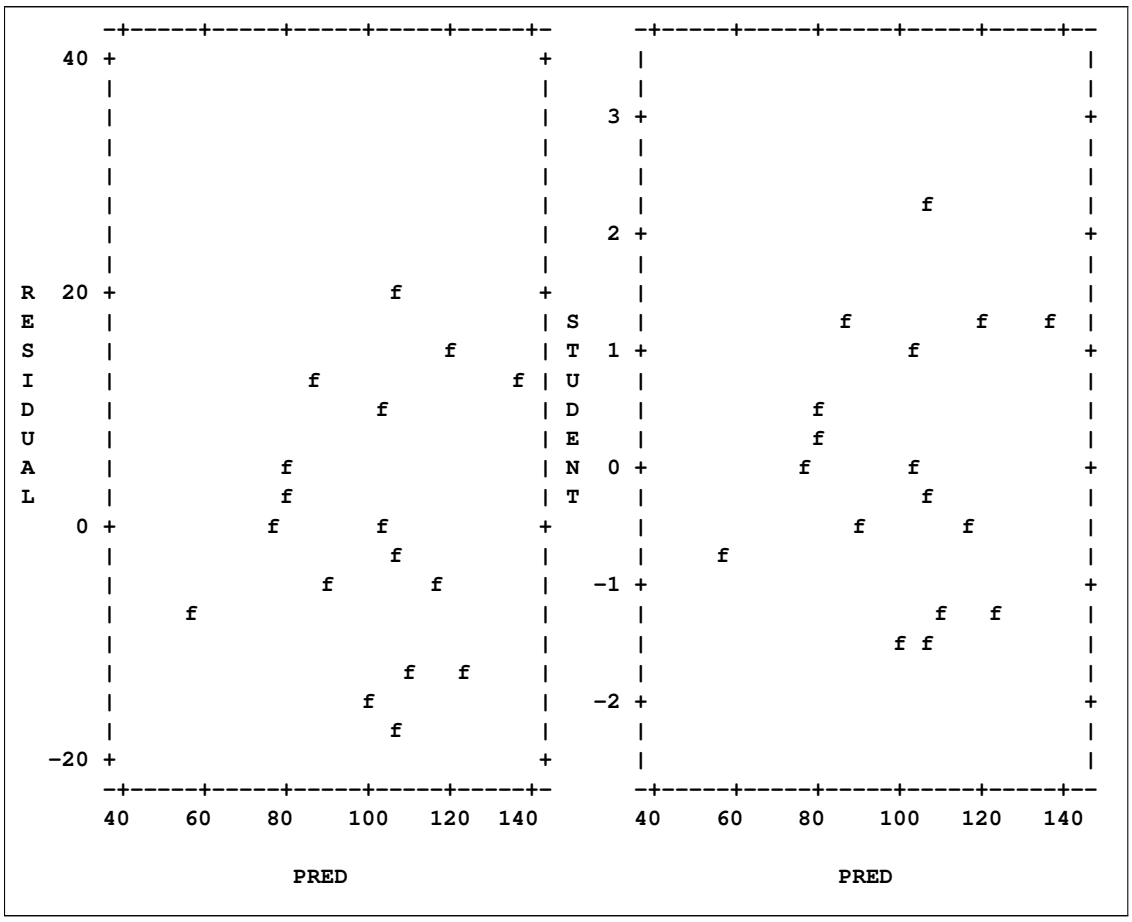
```
plot;
```

is equivalent to respecifying the most recent PLOT statement without any options. However, the COLLECT, HPLOTS=, SYMBOL=, and VPLOTS= options apply across PLOT statements and remain in effect.

The next example shows how you can overlay plots of statistics before and after a change in the model. For the full model involving Height and Age, the ordinary residuals and the studentized residuals are plotted against the predicted values. The COLLECT option causes these plots to be collected or retained for redisplay later. The option HPLOTS=2 enables the two plots to appear side by side on one page. The symbol 'f' is used on these plots to identify them as resulting from the full model. These statements produce Figure 73.35:

```
plot r.*p. student.*p. / collect hplots=2 symbol='f';
run;
```

Figure 73.35 Collecting Residual Plots for the Full Model

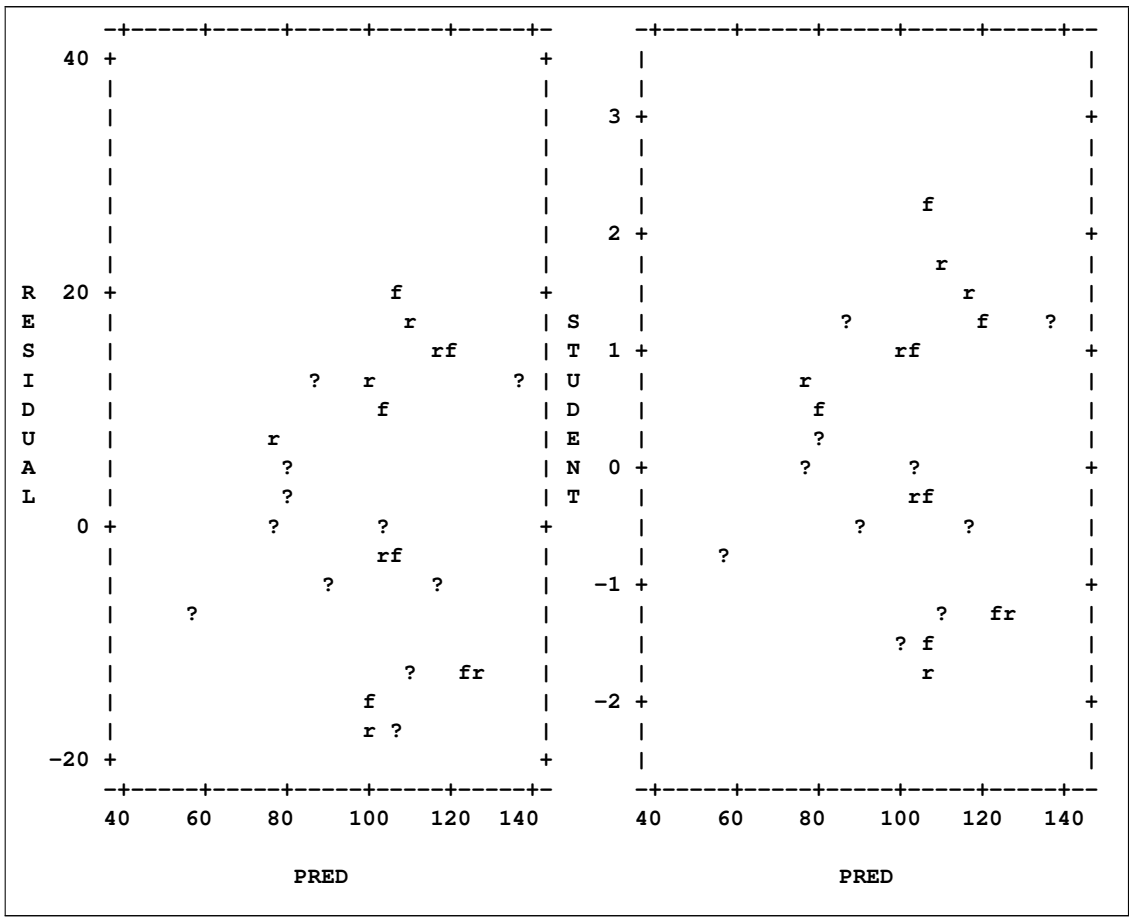


Note that these plots are not overlaid. The COLLECT option does not overlay the plots in one PLOT statement but retains them so that they can be overlaid by later plots. When the COLLECT option appears in a PLOT statement, the plots in that statement become the first plots in the collection.

Next, the model is reduced by deleting the Age variable. The PLOT statement requests the same plots as before but labels the points with the symbol 'r' denoting the reduced model. The following statements produce Figure 73.36:

```
delete Age;
plot r.*p. student.*p. / symbol='r';
run;
```

Figure 73.36 Overlaid Residual Plots for Full and Reduced Models

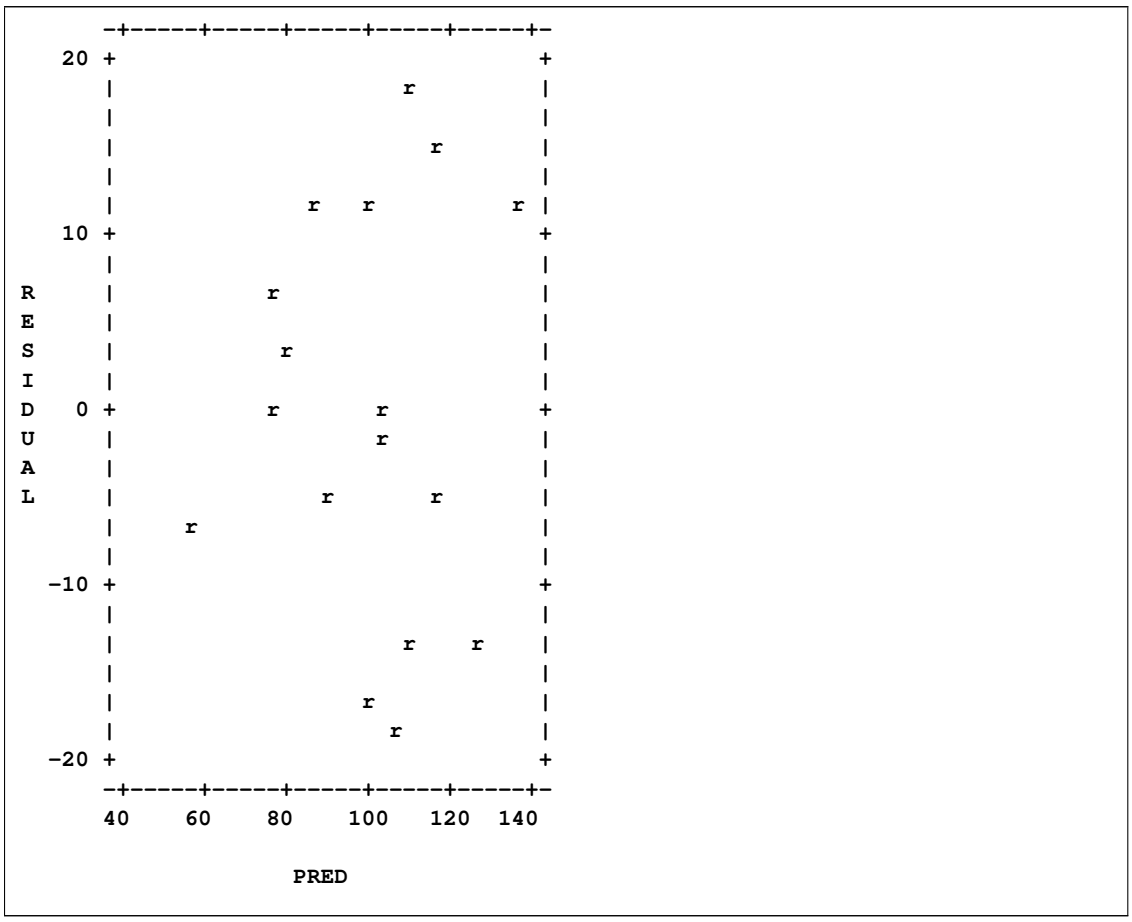


Notice that the COLLECT option causes the corresponding plots to be overlaid. Also notice that the DELETE statement causes the model label to be changed from MODEL1 to MODEL1.1. The points labeled 'f' are from the full model, and the points labeled 'r' are from the reduced model. Positions labeled '?' contain at least one point from each model. In this example, the OVERLAY option cannot be used because all of the plots to be overlaid cannot be specified in one PLOT statement. With the COLLECT option, any changes to the model or the data used to fit the model do not affect plots collected before the changes. Collected plots are always reproduced exactly as they first appear. (Similarly, a PAINT statement does not affect plots collected before the PAINT statement is issued.)

The previous example overlays the residual plots for two different models. You might prefer to see them side by side on the same page. This can also be done with the COLLECT option by using a blank plot. Continuing from the last example, the COLLECT, HPLOTS=2, and SYMBOL='r' options are still in effect. In the following PLOT statement, the CLEAR option deletes the collected plots and enables the specified plot to begin a new collection. The plot created is the residual plot for the reduced model. These statements produce Figure 73.37:

```
plot r.*p. / clear;
run;
```

Figure 73.37 Residual Plot for Reduced Model Only



The next statements add the variable AGE to the model and place the residual plot for the full model next to the plot for the reduced model. Notice that a blank plot is created in the first plot request by placing nothing between the quotes. Since the COLLECT option is in effect, this plot is superimposed on the residual plot for the reduced model. The residual plot for the full model is created by the second request. The result is the desired side-by-side plots. The NOCOLLECT option turns off the collection process after the specified plots are added and displayed. Any PLOT statements that follow show only the newly specified plots. These statements produce Figure 73.38:

```
add Age;
plot r.*p.='' r.*p.='f' / nocollect;
run;
```

Figure 73.38 Side-by-Side Residual Plots for the Full and Reduced Models

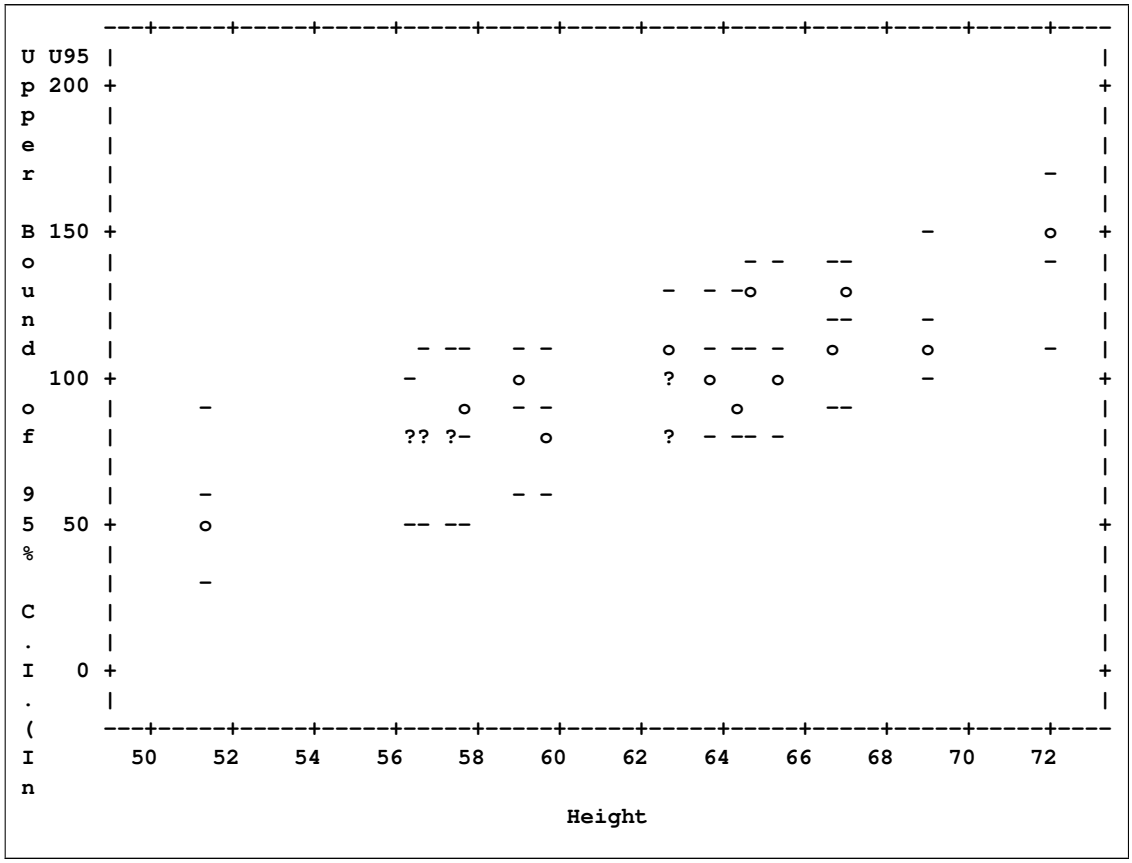


Figure 73.38 *continued*

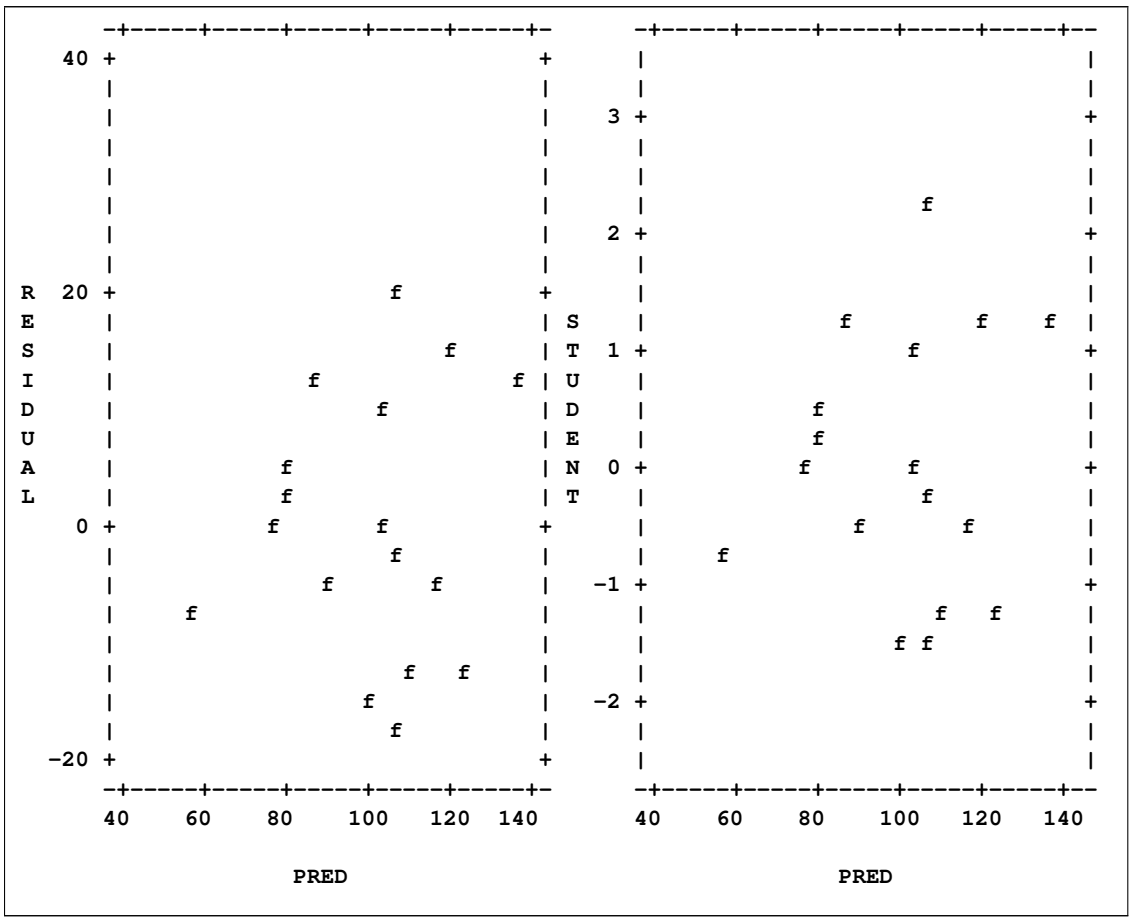


Figure 73.38 *continued*

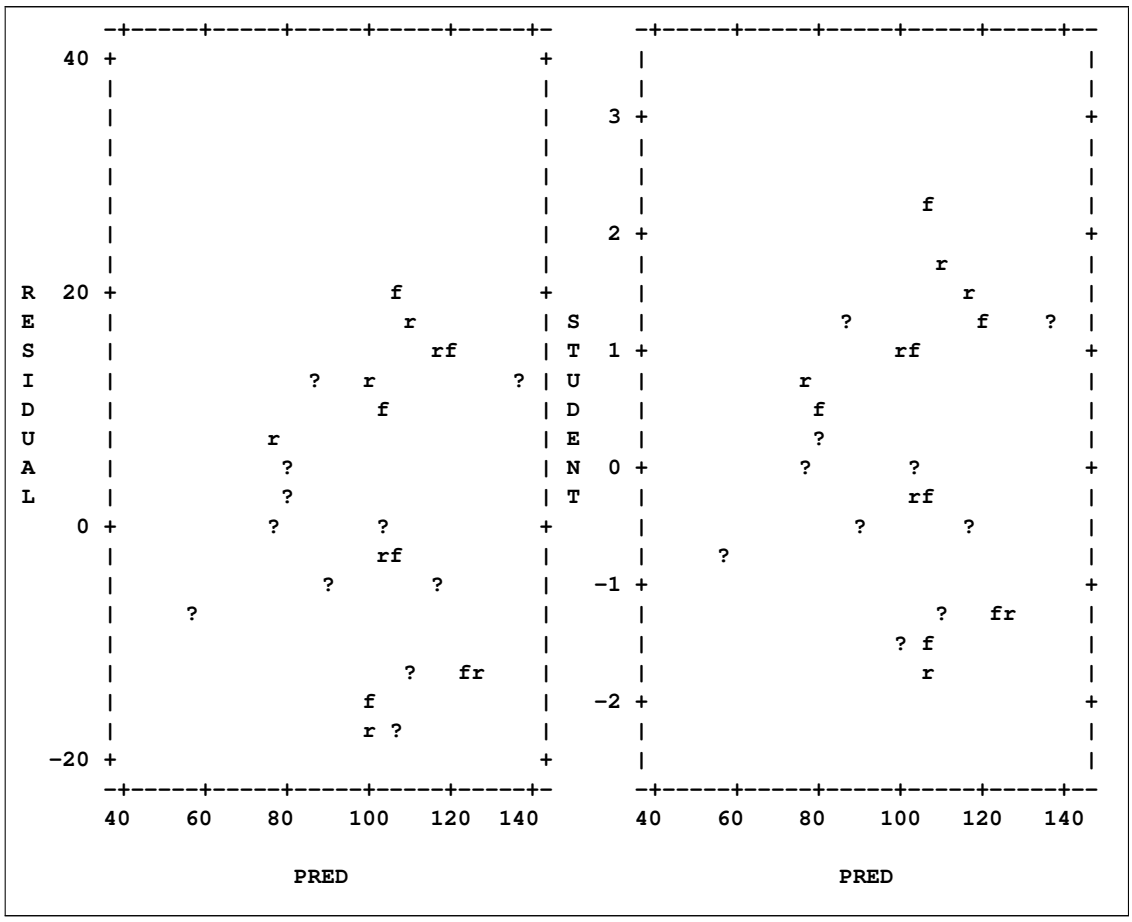
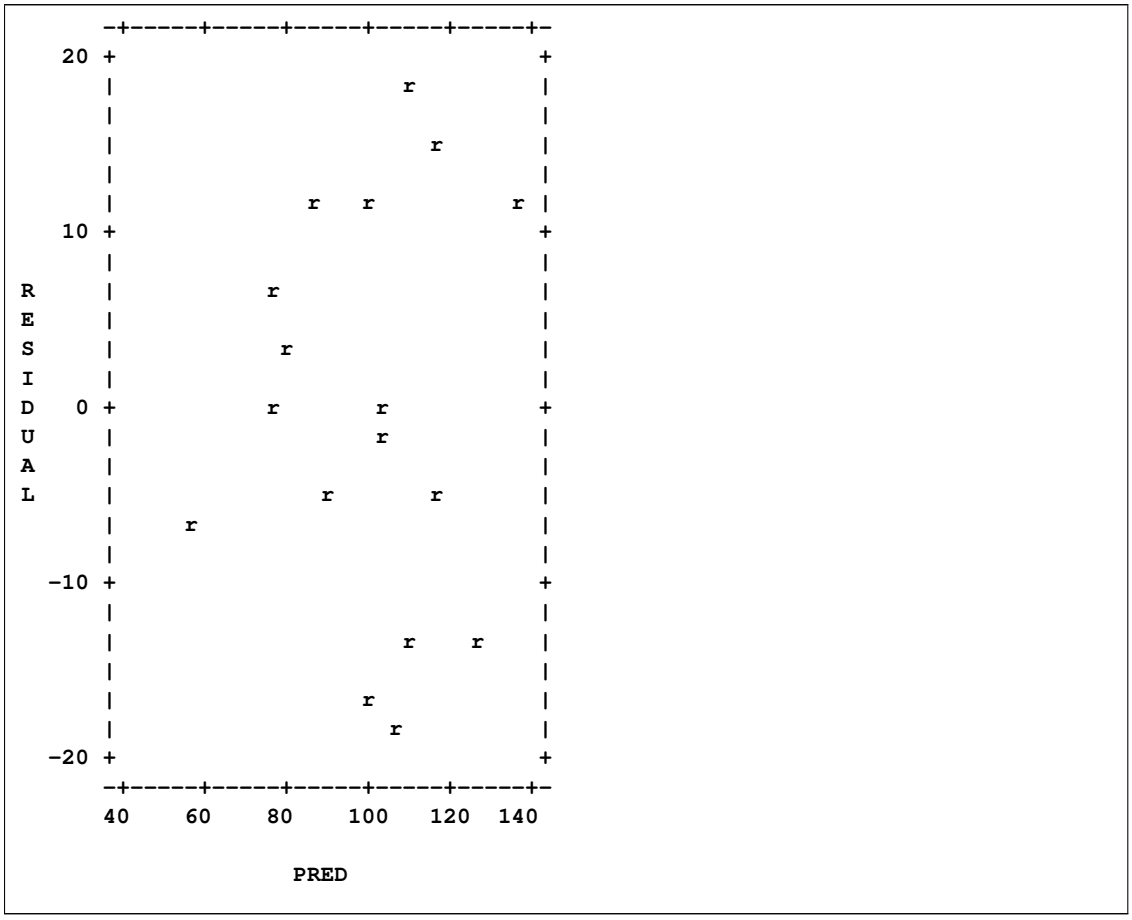


Figure 73.38 *continued*



Frequently, when the COLLECT option is in effect, you want the current and following PLOT statements to show only the specified plots. To do this, use both the CLEAR and NOCOLLECT options in the current PLOT statement.

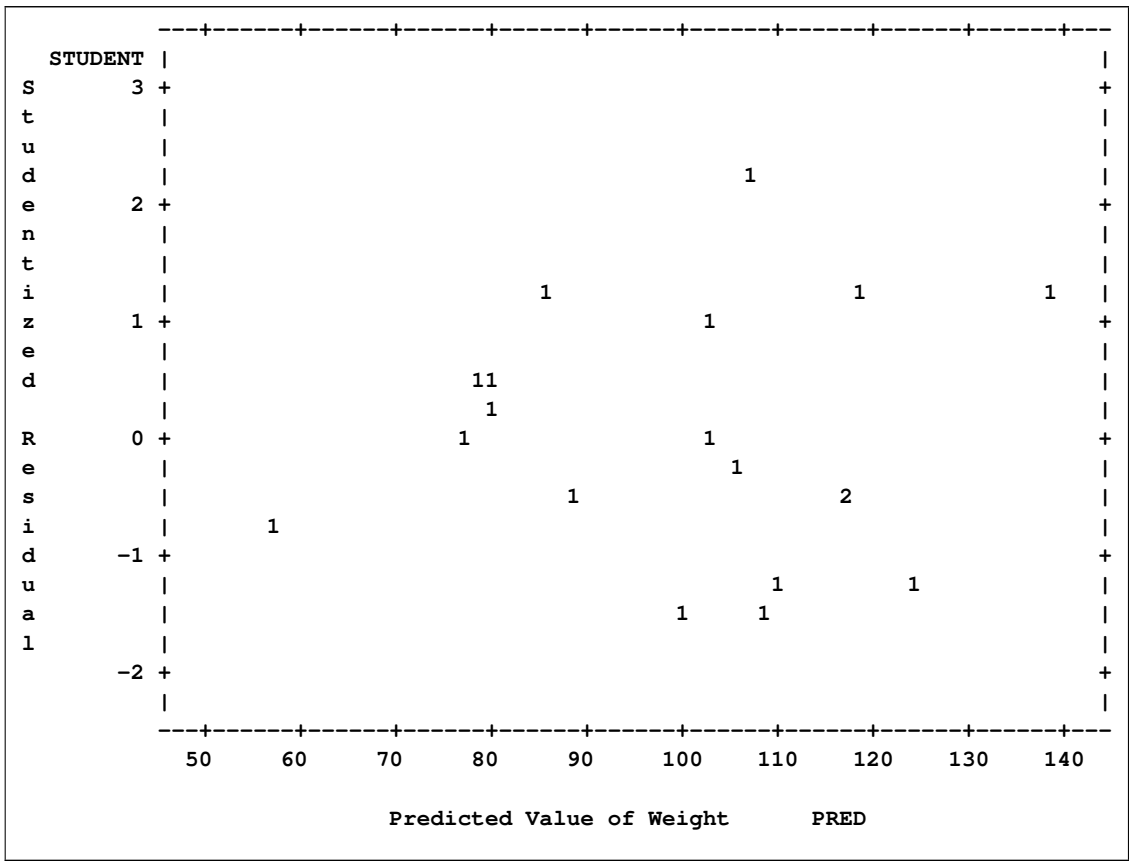
Painting Scatter Plots

Painting scatter plots is a useful interactive tool that enables you to mark points of interest in scatter plots. Painting can be used to identify extreme points in scatter plots or to reveal the relationship between two scatter plots. The Class data (from the section "Simple Linear Regression" on page 5436) is used to illustrate some of these applications.

The following statements produce the scatter plot of the studentized residuals against the predicted values in Figure 73.39.

```
proc reg data=Class lineprinter;
   model Weight=Age Height / noprint;
   plot student.*p.;
run;
```

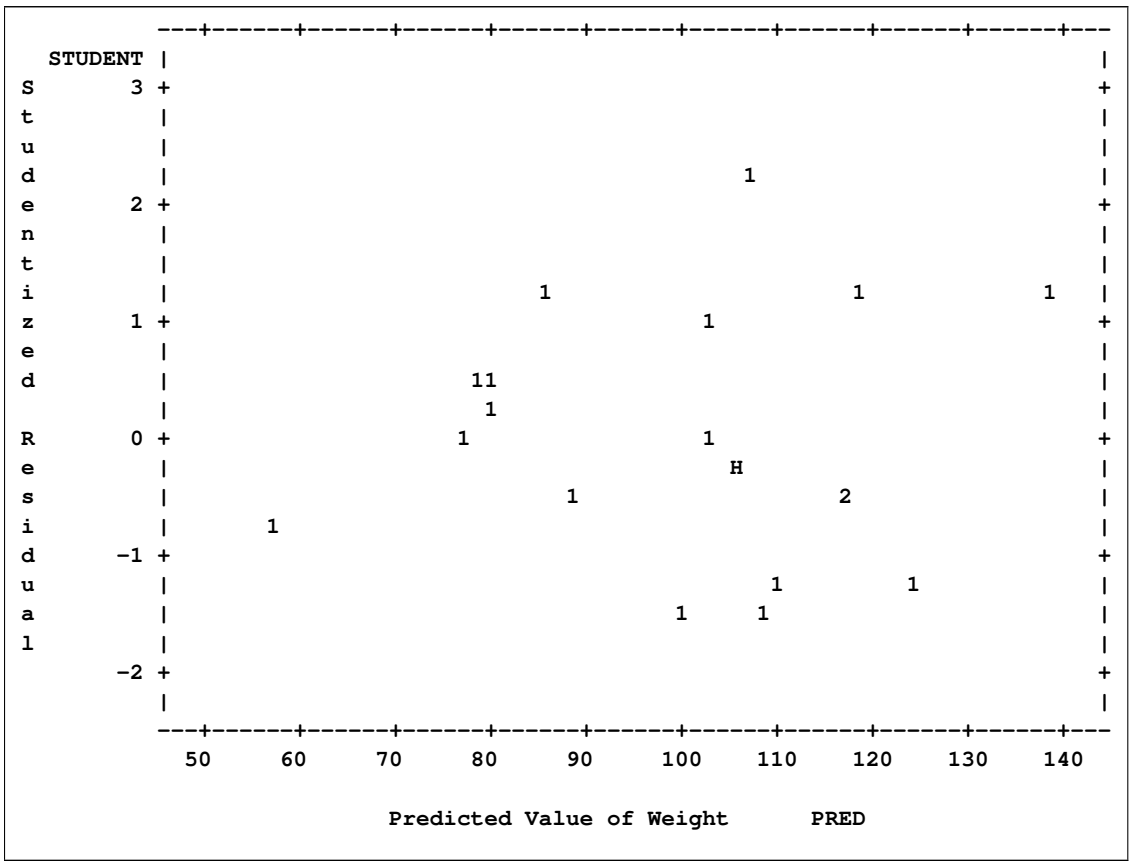
Figure 73.39 Plotting Studentized Residuals against Predicted Values



Then, the following statements identify the observation 'Henry' in the scatter plot and produce the plot in Figure 73.40:

```
paint Name='Henry' / symbol = 'H';
plot;
run;
```

Figure 73.40 Painting One Observation

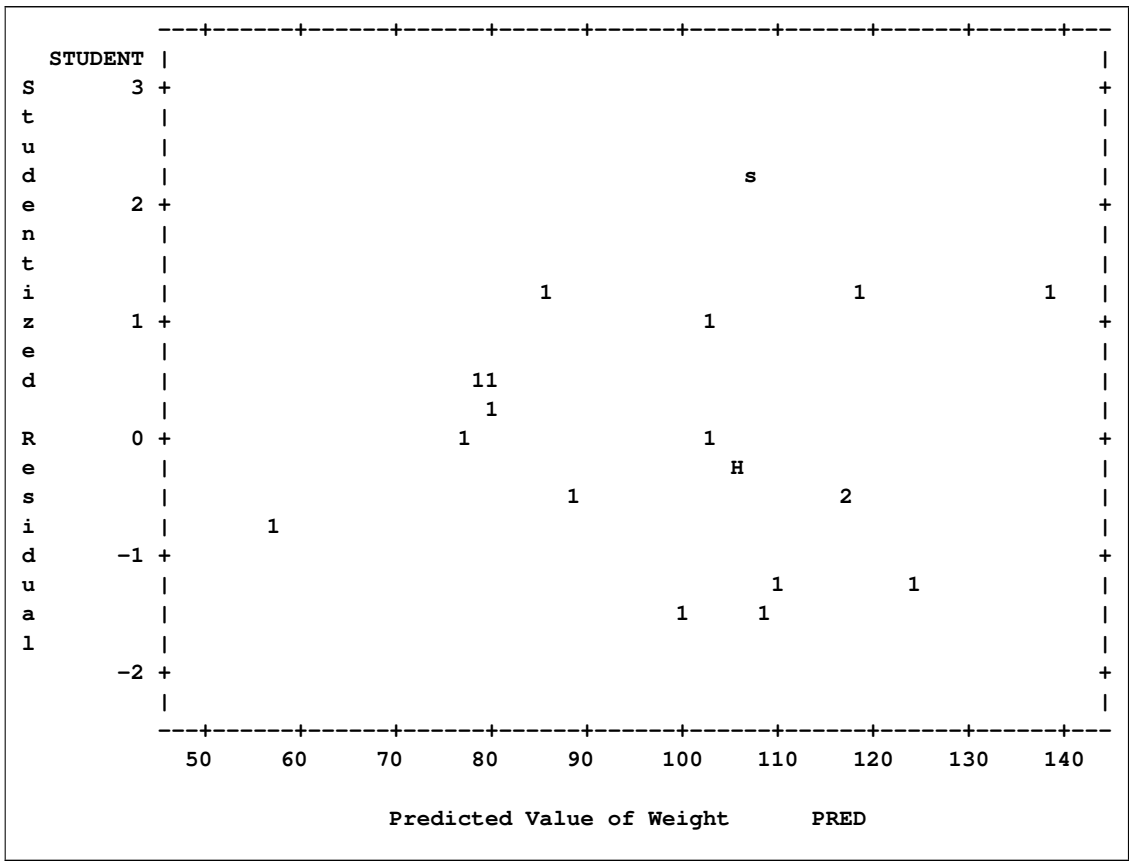


Next, the following statements identify observations with large absolute residuals:

```
paint student.>=2 or student.<=-2 / symbol='s';
plot;
run;
```

The log shows the observation numbers found with these conditions and gives the painting symbol and the number of observations found. Note that the previous PAINT statement is also used in the PLOT statement. Figure 73.41 shows the scatter plot produced by the preceding statements.

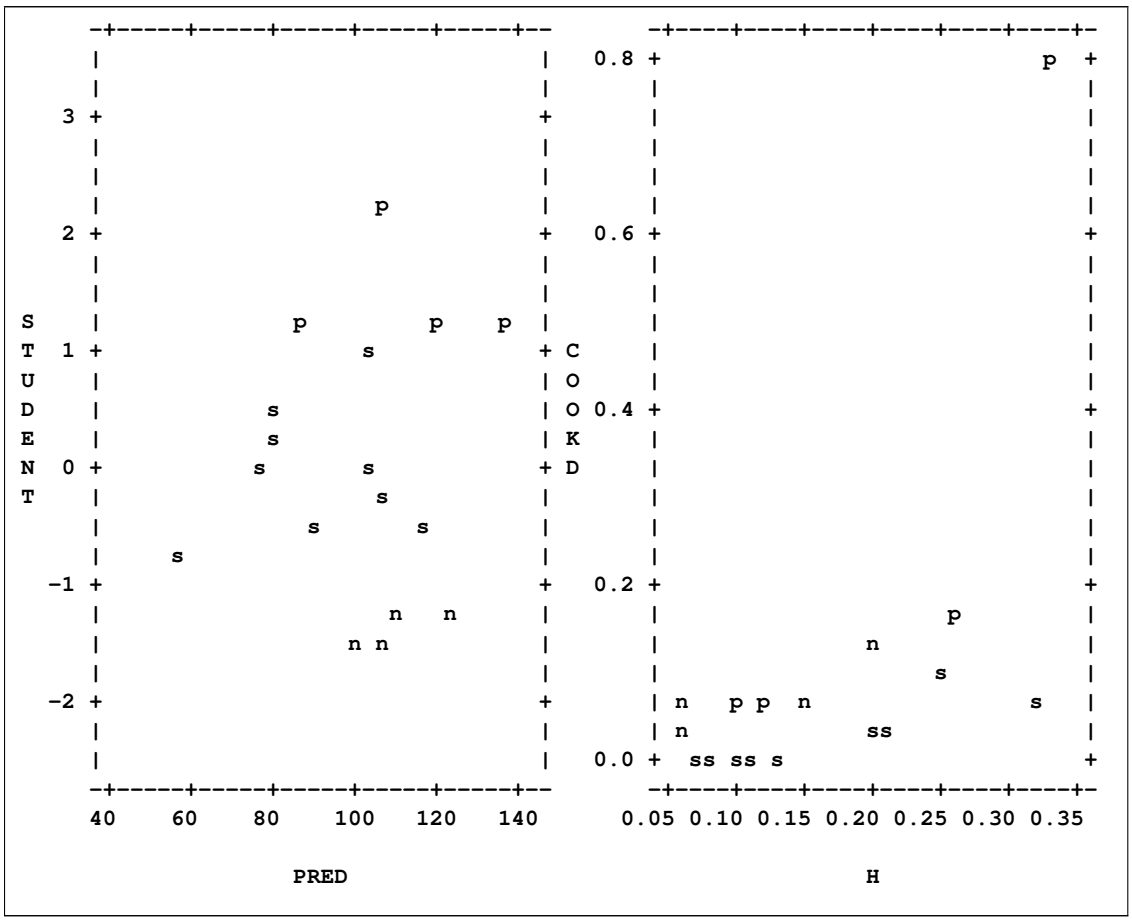
Figure 73.41 Painting Several Observations



The following statements relate two different scatter plots. These statements produce the plot in Figure 73.42.

```
paint student.>=1 / symbol='p';
paint student.<1 and student.>-1 / symbol='s';
paint student.<=-1 / symbol='n';
plot student. * p. cookd. * h. / hplots=2;
run;
```

Figure 73.42 Painting Observations on More Than One Plot



Traditional Graphics

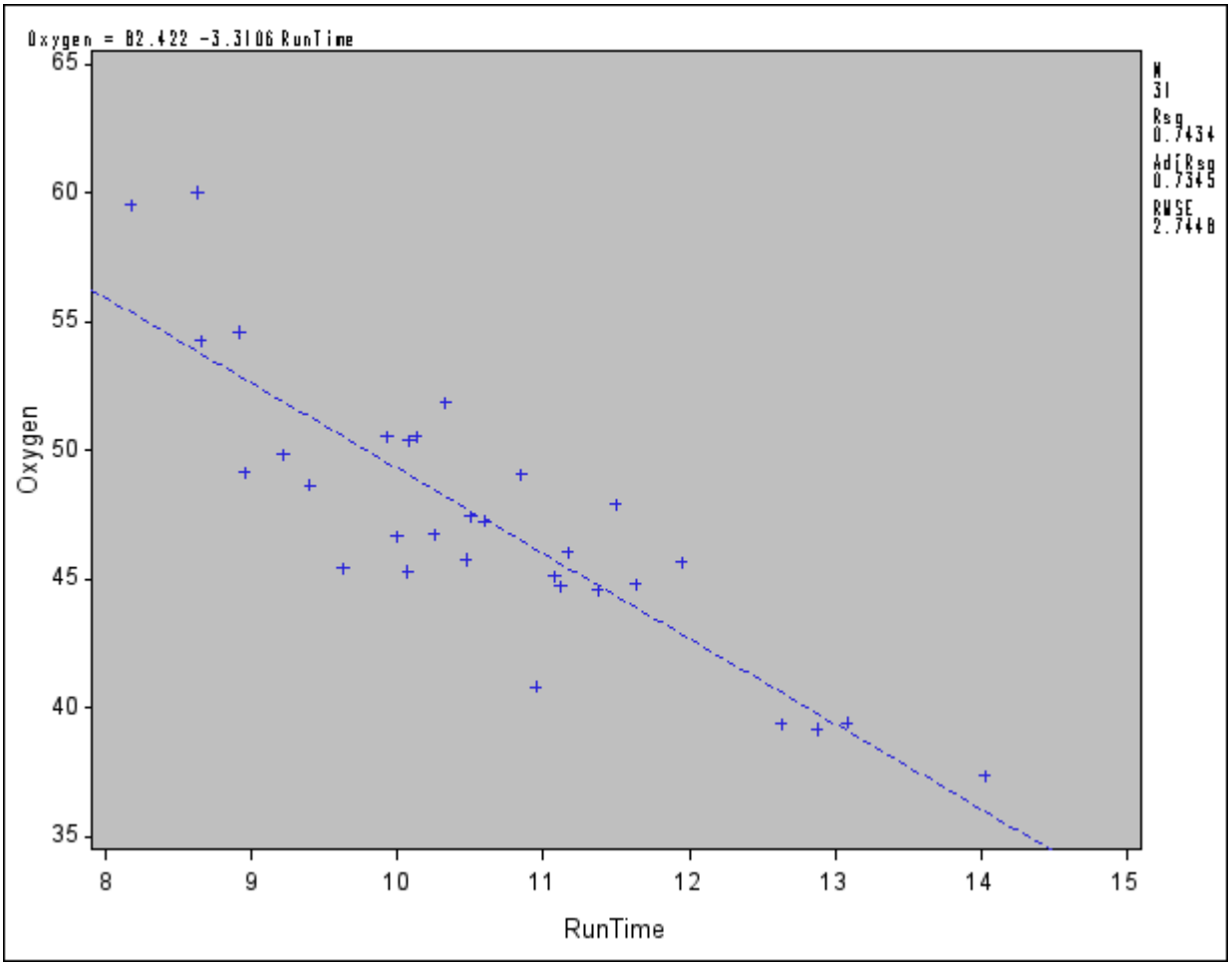
This section provides examples of using options available with the traditional graphics that you request with the PLOT statement. An alternative is to use ODS Graphics to obtain plots relevant to the analysis. See Example 73.1, Example 73.2, and Example 73.5 for examples of obtaining graphical displays with ODS Graphics. Examples in this section use the Fitness data set that is described in Example 73.2.

Traditional Graphics for Simple Linear Regression

The following statements introduce the basic PLOT statement graphics syntax. A simple linear regression of Oxygen on RunTime is performed, and a plot of Oxygen∗RunTime is requested. The fitted model, the regression line, and the four default statistics are also displayed in Figure 73.43.

```
proc reg data=fitness;
   model Oxygen=RunTime;
   plot Oxygen*RunTime / cframe=ligr;
run;
```

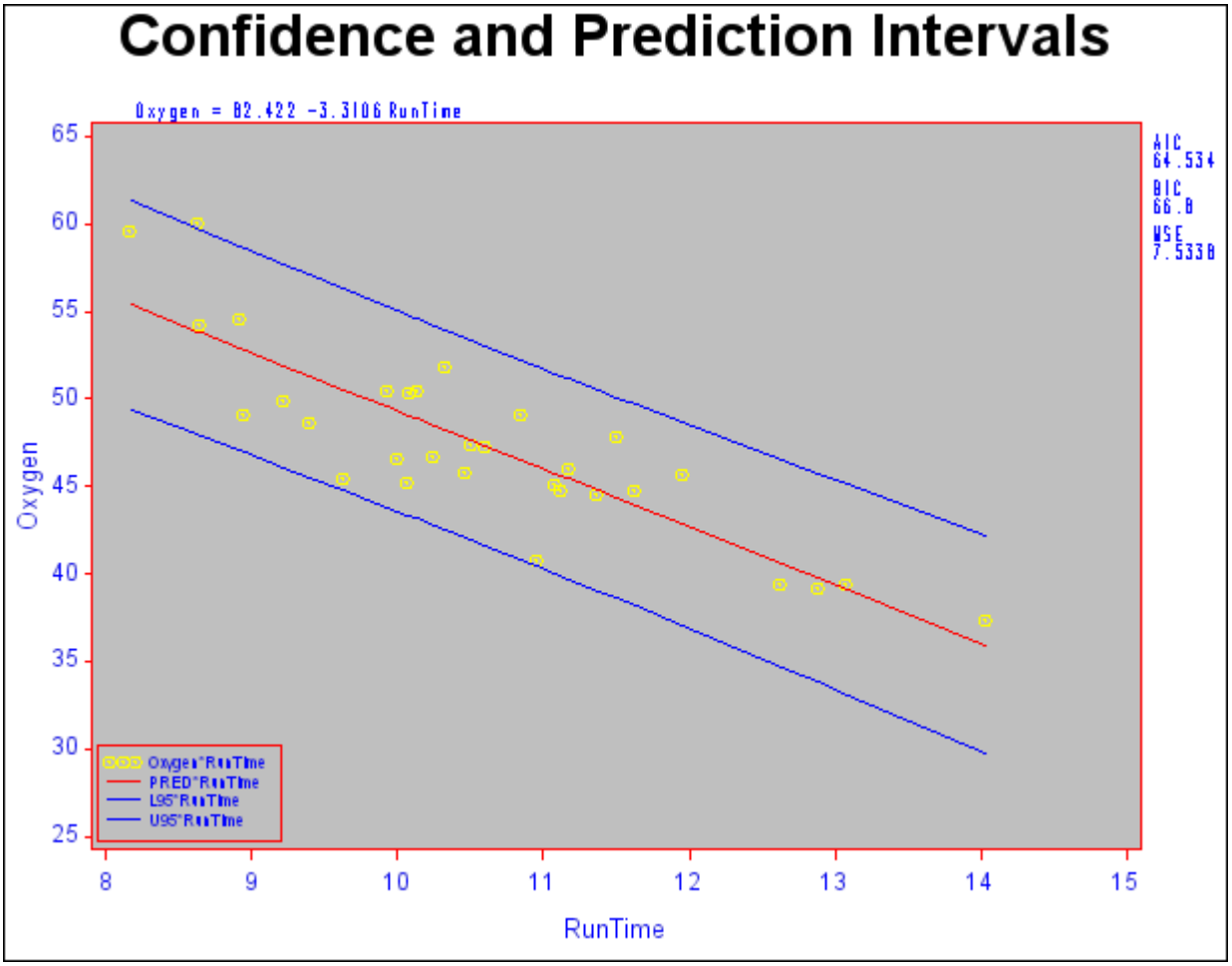
Figure 73.43 Simple Linear Regression



You can use shorthand commands to plot the dependent variable, the predicted value, and the 95% confidence or prediction intervals against a regressor. The following statements use the CONF and PRED options to create a plot with confidence and prediction intervals. Results are displayed in Figure 73.44. Note that the statistics displayed by default in the margin are suppressed while three other statistics are exhibited. Furthermore, global graphics LEGEND and SYMBOL statements and PLOT statement options are used to control the appearance of the plot. For more information about the global graphics statements, see *SAS/GRAPH Software: Reference*.

```
legend1 position=(bottom left inside)
        across=1 cborder=red offset=(0,0)
        shape=symbol(3,1) label=none
        value=(height=.8);
title 'Confidence and Prediction Intervals';
symbol1 c=yellow v=- h=1;
symbol2 c=red;
symbol3 c=blue;
symbol4 c=blue;

proc reg data=fitness;
   model Oxygen=RunTime / noprint;
   plot Oxygen*RunTime / pred nostat mse aic bic
        caxis=red ctext=blue cframe=ligr
        legend=legend1 modellab='          ';
run;
```

Figure 73.44 Simple Linear Regression



Traditional Graphics for Variable Selection

When you use the SELECTION= option in the MODEL statement, you can produce plots showing model fit summary statistics for the models examined. The following statements produce the plot shown in Figure 73.45 of the C_p statistic for model selection plotted against the number of parameters in the model; the CHOCKING= and CMALLOWS= options draw useful reference lines.

```
goptions ctitle=black   htitle=3.5pct ftitle=swiss
         ctext =magenta htext =3.0pct ftext =swiss
         cback =ligr    border;
symbol1 v=circle c=red h=1 w=2;
title 'Cp Plot with Reference Lines';
symbol1 c=green;
proc reg data=fitness;
   model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse
         / selection=rsquare noprint;
   plot cp.*np.
        / chocking=red  cmallows=blue
          vaxis=0 to 15 by 5;
run;
```

In the GOPTIONS statement,

BORDER frames the entire display.

CBACK= specifies the background color.

CTEXT= selects the default color for the border and all text, including titles, footnotes, and notes.

CTITLE= specifies the title, footnote, note, and border color.

HTEXT= specifies the height for all text in the display.

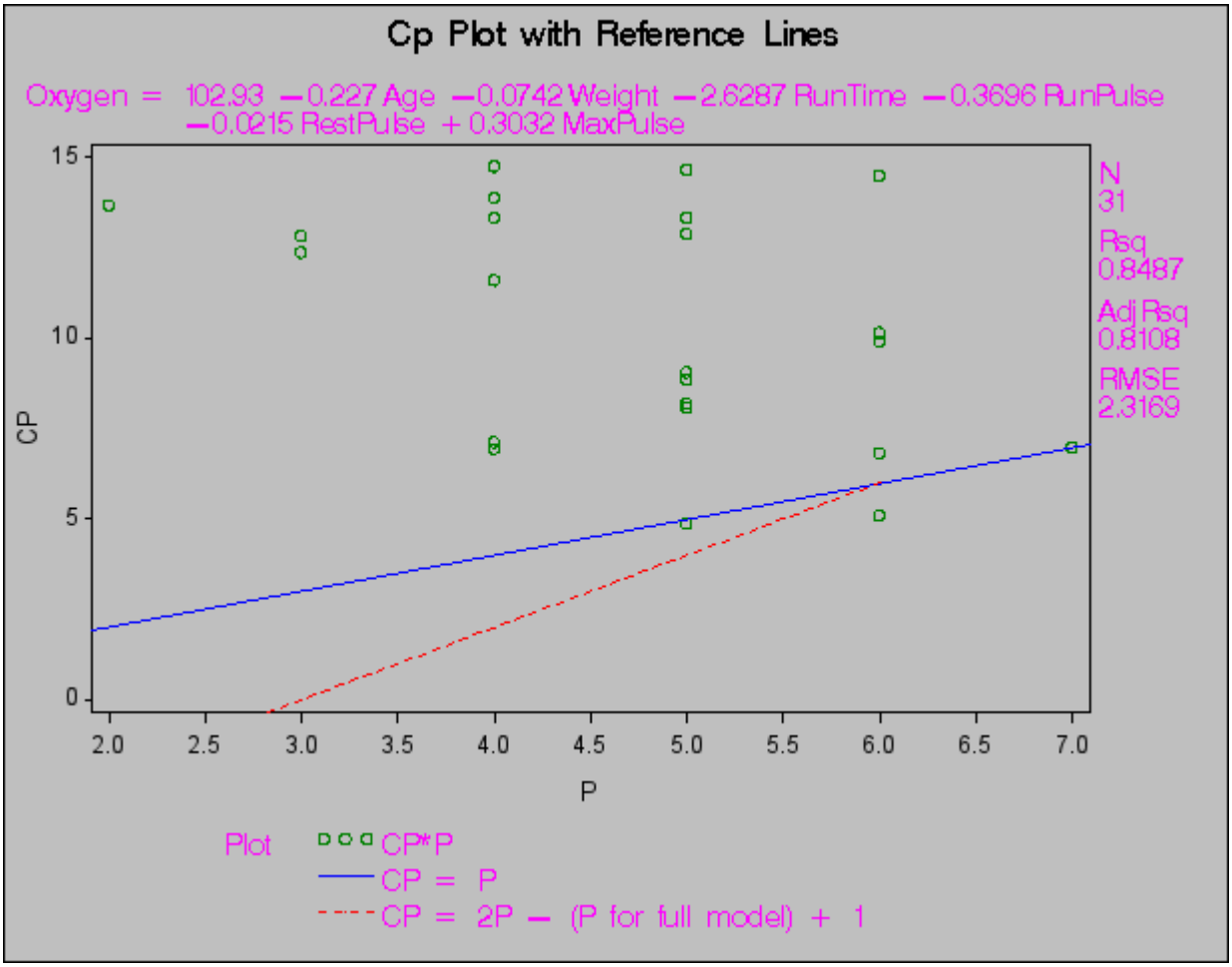
HTITLE= specifies the height for the first title line.

FTEXT= selects the default font for all text, including titles, footnotes, notes, the model label and equation, the statistics, the axis labels, the tick values, and the legend.

FTITLE= specifies the first title font.

For more information about the GOPTIONS statement and other global graphics statements, see *SAS/GRAPH Software: Reference*.

Figure 73.45 C_p Plot with Reference Lines



Traditional Normal Quantile and Normal Probability Plots

The following statements create probability-probability plots and quantile-quantile plots of the residuals (Figure 73.46 and Figure 73.47, respectively). An annotation data set is created to produce the (0,0)−(1,1) reference line for the P-P plot. Note that the NOSTAT option for the P-P plot suppresses the statistics that would be displayed in the margin.

```
data annote1;
   length function color $8;
   retain ysys xsys '2' color 'black';
   function='move';
      x=0;
      y=0;
      output;
   function='draw';
      x=1;
      y=1;
      output;
run;
```

```
symbol1 c=blue;
proc reg data=fitness;
   title 'PP Plot';
   model Oxygen=RunTime / noprint;
   plot npp.*r.
        / annotate=annote1 nostat cframe=ligr
          modellab="'Best' Two-Parameter Model:";
run;
   title 'QQ Plot';
   plot r.*nqq.
        / noline mse cframe=ligr
          modellab="'Best' Two-Parameter Model:";
run;
```

Figure 73.46 Normal Probability-Probability Plot for the Residuals

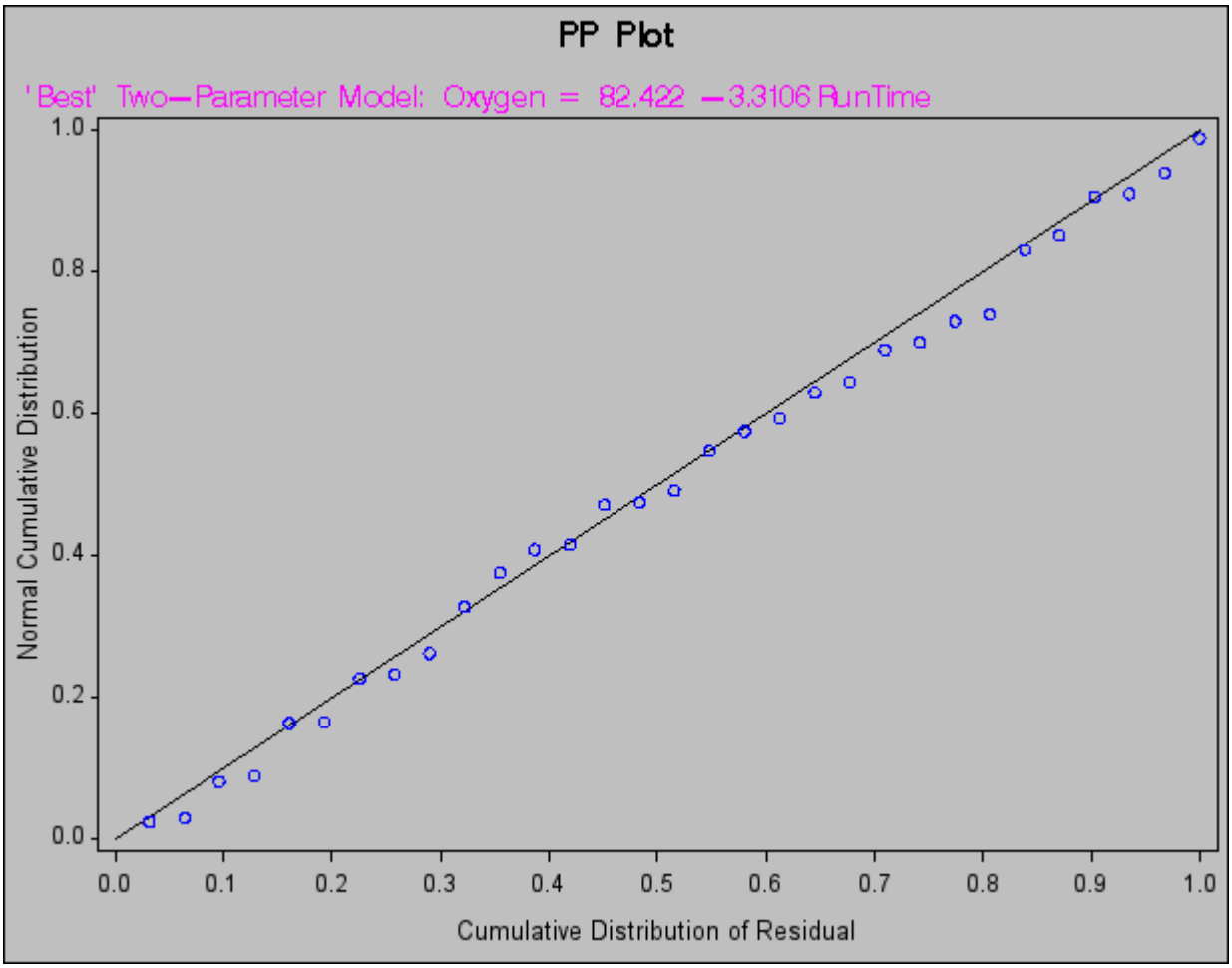
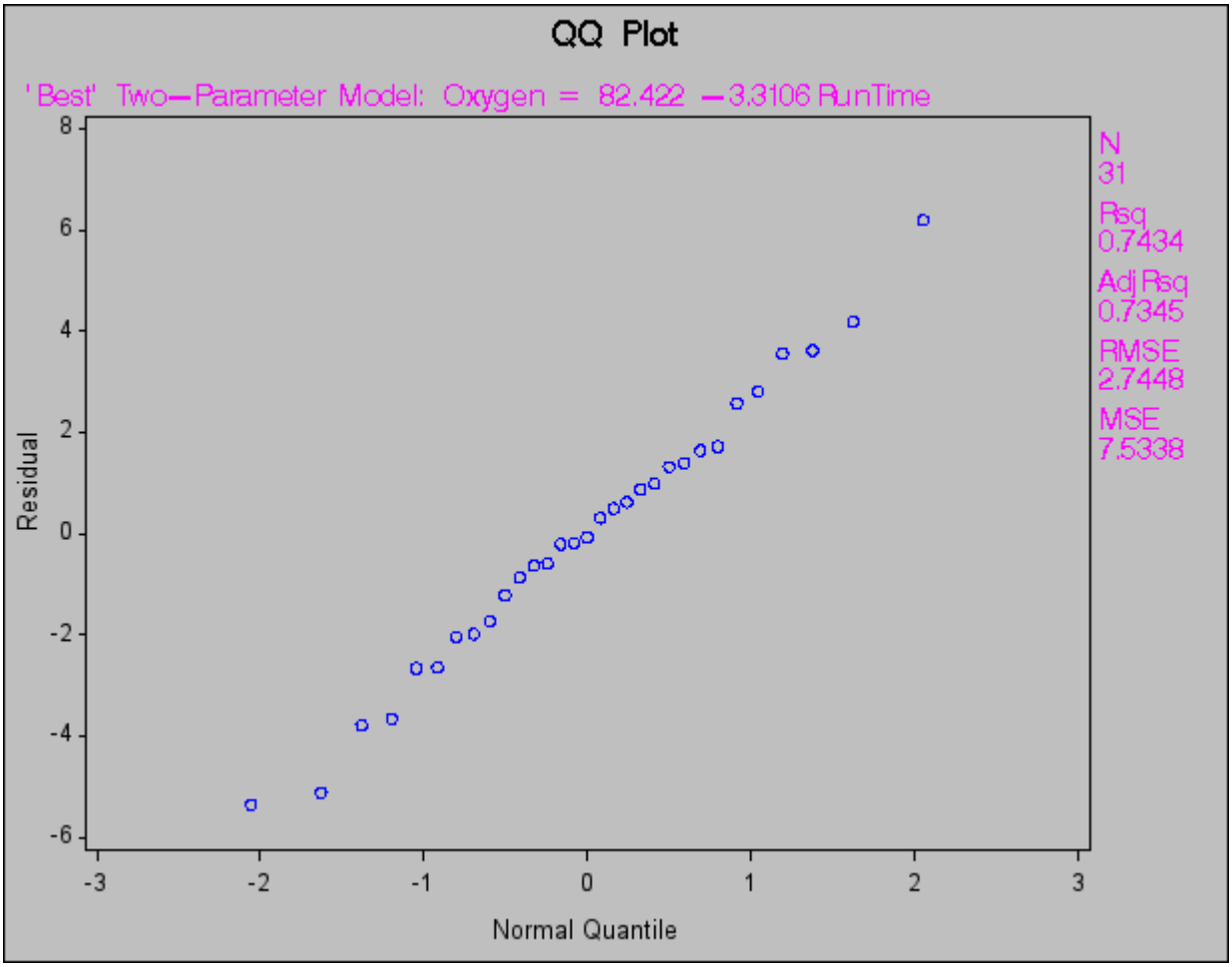


Figure 73.47 Normal Quantile-Quantile Plot for the Residuals



Models of Less Than Full Rank

If the model is not full rank, there are an infinite number of least squares solutions for the estimates. PROC REG chooses a nonzero solution for all variables that are linearly independent of previous variables and a zero solution for other variables. This solution corresponds to using a generalized inverse in the normal equations, and the expected values of the estimates are the Hermite normal form of $\mathbf{X}$ multiplied by the true parameters:

$$E(\mathbf{b}) = (\mathbf{X}'\mathbf{X})^{-}(\mathbf{X}'\mathbf{X})\boldsymbol{\beta}$$

Degrees of freedom for the zeroed estimates are reported as zero. The hypotheses that are not testable have t tests reported as missing. The message that the model is not full rank includes a display of the relations that exist in the matrix.

The following statements use the fitness data from Example 73.2. The variable Dif=RunPulse−RestPulse is created. When this variable is included in the model along with RunPulse and RestPulse, there is a linear dependency (or exact collinearity) between the independent variables. Figure 73.48 shows how this problem is diagnosed.

```
data fit2;
   set fitness; Dif=RunPulse-RestPulse;
proc reg data=fit2;
   model Oxygen=RunTime Age Weight RunPulse MaxPulse RestPulse Dif;
run;
```

Figure 73.48 Model That Is Not Full Rank: REG Procedure

The REG Procedure
Model: MODEL1
Dependent Variable: Oxygen

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	722.54361	120.42393	22.43	<.0001
Error	24	128.83794	5.36825		
Corrected Total	30	851.38154			

Root MSE	2.31695	R-Square	0.8487
Dependent Mean	47.37581	Adj R-Sq	0.8108
Coeff Var	4.89057		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	102.93448	12.40326	8.30	<.0001
RunTime	1	−2.62865	0.38456	−6.84	<.0001
Age	1	−0.22697	0.09984	−2.27	0.0322
Weight	1	−0.07418	0.05459	−1.36	0.1869
RunPulse	B	−0.36963	0.11985	−3.08	0.0051
MaxPulse	1	0.30322	0.13650	2.22	0.0360
RestPulse	B	−0.02153	0.06605	−0.33	0.7473
Dif	0	0	.	.	.

PROC REG produces a message informing you that the model is less than full rank. Parameters with DF=0 are not estimated, and parameters with DF=B are biased. In addition, the form of the linear dependency among the regressors is displayed.

Collinearity Diagnostics

When a regressor is nearly a linear combination of other regressors in the model, the affected estimates are unstable and have high standard errors. This problem is called *collinearity* or *multicollinearity*. It is a good idea to find out which variables are nearly collinear with which other variables. The approach in PROC REG follows that of Belsley, Kuh, and Welsch (1980). PROC REG provides several methods for detecting collinearity with the COLLIN, COLLINOINT, TOL, and VIF options.

The COLLIN option in the MODEL statement requests that a collinearity analysis be performed. First, $\mathbf{X}'\mathbf{X}$ is scaled to have 1s on the diagonal. If you specify the COLLINOINT option, the intercept variable is adjusted out first. Then the eigenvalues and eigenvectors are extracted. The analysis in PROC REG is reported with eigenvalues of $\mathbf{X}'\mathbf{X}$ rather than singular values of $\mathbf{X}$. The eigenvalues of $\mathbf{X}'\mathbf{X}$ are the squares of the singular values of $\mathbf{X}$.

The condition indices are the square roots of the ratio of the largest eigenvalue to each individual eigenvalue. The largest condition index is the condition number of the scaled $\mathbf{X}$ matrix. Belsey, Kuh, and Welsch (1980) suggest that, when this number is around 10, weak dependencies might be starting to affect the regression estimates. When this number is larger than 100, the estimates might have a fair amount of numerical error (although the statistical standard error almost always is much greater than the numerical error).

For each variable, PROC REG produces the proportion of the variance of the estimate accounted for by each principal component. A collinearity problem occurs when a component associated with a high condition index contributes strongly (variance proportion greater than about 0.5) to the variance of two or more variables.

The VIF option in the MODEL statement provides the variance inflation factors (VIF). These factors measure the inflation in the variances of the parameter estimates due to collinearities that exist among the regressor (independent) variables. There are no formal criteria for deciding if a VIF is large enough to affect the predicted values.

The TOL option requests the tolerance values for the parameter estimates. The tolerance is defined as $1/VIF$.

For a complete discussion of the preceding methods, refer to Belsley, Kuh, and Welsch (1980). For a more detailed explanation of using the methods with PROC REG, refer to Freund and Littell (1986).

This example uses the COLLIN option on the fitness data found in Example 73.2. The following statements produce Figure 73.49.

```
proc reg data=fitness;
   model Oxygen=RunTime Age Weight RunPulse MaxPulse RestPulse
         / tol vif collin;
run;
```

Figure 73.49 Regression Using the TOL, VIF, and COLLIN Options

The REG Procedure
Model: MODEL1
Dependent Variable: Oxygen

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	722.54361	120.42393	22.43	<.0001
Error	24	128.83794	5.36825		
Corrected Total	30	851.38154			

Root MSE	2.31695	R-Square	0.8487
Dependent Mean	47.37581	Adj R-Sq	0.8108
Coeff Var	4.89057		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|	Tolerance
Intercept	1	102.93448	12.40326	8.30	<.0001	.
RunTime	1	-2.62865	0.38456	-6.84	<.0001	0.62859
Age	1	-0.22697	0.09984	-2.27	0.0322	0.66101
Weight	1	-0.07418	0.05459	-1.36	0.1869	0.86555
RunPulse	1	-0.36963	0.11985	-3.08	0.0051	0.11852
MaxPulse	1	0.30322	0.13650	2.22	0.0360	0.11437
RestPulse	1	-0.02153	0.06605	-0.33	0.7473	0.70642

Parameter Estimates

Variable	DF	Variance Inflation
Intercept	1	0
RunTime	1	1.59087
Age	1	1.51284
Weight	1	1.15533
RunPulse	1	8.43727
MaxPulse	1	8.74385
RestPulse	1	1.41559

Figure 73.49 *continued*

Collinearity Diagnostics

Number	Eigenvalue	Condition Index	Proportion of Variation: Intercept	RunTime	Age
1	6.94991	1.00000	0.00002326	0.00021086	0.00015451
2	0.01868	19.29087	0.00218	0.02522	0.14632
3	0.01503	21.50072	0.00061541	0.12858	0.15013
4	0.00911	27.62115	0.00638	0.60897	0.03186
5	0.00607	33.82918	0.00133	0.12501	0.11284
6	0.00102	82.63757	0.79966	0.09746	0.49660
7	0.00017947	196.78560	0.18981	0.01455	0.06210

Collinearity Diagnostics

Number	Proportion of Variation: Weight	RunPulse	MaxPulse	RestPulse
1	0.00019651	0.00000862	0.00000634	0.00027850
2	0.01042	0.00000244	0.00000743	0.39064
3	0.23571	0.00119	0.00125	0.02809
4	0.18313	0.00149	0.00123	0.19030
5	0.44442	0.01506	0.00833	0.36475
6	0.10330	0.06948	0.00561	0.02026
7	0.02283	0.91277	0.98357	0.00568

Model Fit and Diagnostic Statistics

This section gathers the formulas for the statistics available in the MODEL, PLOT, and OUTPUT statements. The model to be fit is $\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}$, and the parameter estimate is denoted by $\mathbf{b} = (\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{Y}$. The subscript i denotes values for the ith observation, the parenthetical subscript (i) means that the statistic is computed by using all observations except the ith observation, and the subscript jj indicates the jth diagonal matrix entry. The ALPHA= option in the PROC REG or MODEL statement is used to set the α value for the t statistics.

Table 73.8 contains the summary statistics for assessing the fit of the model.

Table 73.8 Formulas and Definitions for Model Fit Summary Statistics

MODEL Option or Statistic	Definition or Formula
n	the number of observations
p	the number of parameters including the intercept
i	1 if there is an intercept, 0 otherwise
$\hat{\sigma}^2$	the estimate of pure error variance from the SIGMA= option or from fitting the full model
SST_0	the uncorrected total sum of squares for the dependent variable

Table 73.8 *continued*

MODEL Option or Statistic	Definition or Formula
SST_1	the total sum of squares corrected for the mean for the dependent variable
SSE	the error sum of squares
MSE	$\frac{SSE}{n-p}$
R^2	$1-\frac{SSE}{SST_i}$
ADJRSQ	$1-\frac{(n-i)(1-R^2)}{n-p}$
AIC	$n\ln\left(\frac{SSE}{n}\right)+2p$
BIC	$n\ln\left(\frac{SSE}{n}\right)+2(p+2)q-2q^2$ where $q=\frac{n\hat{\sigma}^2}{SSE}$
CP (C_p)	$\frac{SSE}{\hat{\sigma}^2}+2p-n$
GMSEP	$\frac{MSE(n+1)(n-2)}{n(n-p-1)}=\frac{1}{n}S_p(n+1)(n-2)$
JP (J_p)	$\frac{n+p}{n}MSE$
PC	$\frac{n+p}{n-p}(1-R^2)=J_p\left(\frac{n}{SST_i}\right)$
PRESS	the sum of squares of $predr_i$ (see Table 73.9)
RMSE	$\sqrt{MSE}$
SBC	$n\ln\left(\frac{SSE}{n}\right)+p\ln(n)$
SP (S_p)	$\frac{MSE}{n-p-1}$

Table 73.9 contains the diagnostic statistics and their formulas; these formulas and further information can be found in Chapter 4, "Introduction to Regression Procedures," and in the section "Influence Statistics" on page 5559. Each statistic is computed for each observation.

Table 73.9 Formulas and Definitions for Diagnostic Statistics

MODEL Option or Statistic	Formula
PRED ($\widehat{\mathbf{Y}}_i$)	$\mathbf{X}_i\mathbf{b}$
RES (r_i)	$\mathbf{Y}_i-\widehat{\mathbf{Y}}_i$
H (h_i)	$\mathbf{x}_i(\mathbf{X}'\mathbf{X})^{-}\mathbf{x}_i'$
STDP	$\sqrt{h_i\widehat{\sigma}^2}$

Table 73.9 *continued*

MODEL Option or Statistic	Formula
STDI	$\sqrt{(1+h_i)\widehat{\sigma}^2}$
STDR	$\sqrt{(1-h_i)\widehat{\sigma}^2}$
LCL	$\widehat{Y}_i - t_{\frac{\alpha}{2}}\text{STDI}$
LCLM	$\widehat{Y}_i - t_{\frac{\alpha}{2}}\text{STDP}$
UCL	$\widehat{Y}_i + t_{\frac{\alpha}{2}}\text{STDI}$
UCLM	$\widehat{Y}_i + t_{\frac{\alpha}{2}}\text{STDP}$
STUDENT	$\frac{r_i}{\text{STDR}_i}$
RSTUDENT	$\frac{r_i}{\hat{\sigma}_{(i)}\sqrt{1-h_i}}$
COOKD	$\frac{1}{p}\text{STUDENT}^2\frac{\text{STDP}^2}{\text{STDR}^2}$
COVRATIO	$\frac{\det(\hat{\sigma}^2_{(i)}(\mathbf{x}'_{(i)}\mathbf{x}_{(i)})^{-1}}{\det(\hat{\sigma}^2(\mathbf{X}'\mathbf{X})^{-1})}$
DFFITS	$\frac{(\widehat{\mathbf{Y}}_i - \widehat{\mathbf{Y}}_{(i)})}{(\hat{\sigma}_{(i)}\sqrt{h_i})}$
DFBETAS$_j$	$\frac{\mathbf{b}_j - \mathbf{b}_{(i)j}}{\hat{\sigma}_{(i)}\sqrt{(\mathbf{X}'\mathbf{X})_{jj}}}$
PRESS(predr$_i$)	$\frac{r_i}{1-h_i}$

Influence Statistics

This section discusses the INFLUENCE option, which produces several influence statistics, and the PARTIAL option, which produces partial regression leverage plots.

The INFLUENCE Option

The INFLUENCE option (in the MODEL statement) requests the statistics proposed by Belsley, Kuh, and Welsch (1980) to measure the influence of each observation on the estimates. Influential observations are those that, according to various criteria, appear to have a large influence on the parameter estimates.

Let $\mathbf{b}(i)$ be the parameter estimates after deleting the ith observation; let $s(i)^2$ be the variance estimate after deleting the ith observation; let $\mathbf{X}(i)$ be the $\mathbf{X}$ matrix without the ith observation; let $\hat{y}(i)$ be the ith value predicted without using the ith observation; let $r_i = y_i - \hat{y}_i$ be the ith

residual; and let h_i be the ith diagonal of the projection matrix for the predictor space, also called the *hat matrix*:

$$h_i = \mathbf{x}_i(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_i'$$

Belsley, Kuh, and Welsch (1980) propose a cutoff of $2p/n$, where n is the number of observations used to fit the model and p is the number of parameters in the model. Observations with h_i values above this cutoff should be investigated.

For each observation, PROC REG first displays the residual, the studentized residual (RSTUDENT), and the h_i. The studentized residual RSTUDENT differs slightly from STUDENT since the error variance is estimated by $s^2_{(i)}$ without the ith observation, not by s^2. For example,

$$\text{RSTUDENT} = \frac{r_i}{s_{(i)}\sqrt{(1-h_i)}}$$

Observations with RSTUDENT larger than 2 in absolute value might need some attention.

The COVRATIO statistic measures the change in the determinant of the covariance matrix of the estimates by deleting the ith observation:

$$\text{COVRATIO} = \frac{\det\left(s^2(i)(\mathbf{X}'_{(i)}\mathbf{X}_{(i)})^{-1}\right)}{\det\left(s^2(\mathbf{X}'\mathbf{X})^{-1}\right)}$$

Belsley, Kuh, and Welsch (1980) suggest that observations with

$$|\text{COVRATIO} - 1| \geq \frac{3p}{n}$$

where p is the number of parameters in the model and n is the number of observations used to fit the model, are worth investigation.

The DFFITS statistic is a scaled measure of the change in the predicted value for the ith observation and is calculated by deleting the ith observation. A large value indicates that the observation is very influential in its neighborhood of the $\mathbf{X}$ space.

$$\text{DFFITS} = \frac{\hat{y}_i - \hat{y}_{(i)}}{s_{(i)}\sqrt{h_{(i)}}}$$

Large values of DFFITS indicate influential observations. A general cutoff to consider is 2; a size-adjusted cutoff recommended by Belsley, Kuh, and Welsch (1980) is $2\sqrt{p/n}$, where n and p are as defined previously.

The DFFITS statistic is very similar to Cook's D, defined in the section "Predicted and Residual Values" on page 5531.

The DFBETAS statistics are the scaled measures of the change in each parameter estimate and are calculated by deleting the ith observation:

$$\text{DFBETAS}_j = \frac{b_j - b_{(i)j}}{s_{(i)}\sqrt{(\mathbf{X}'\mathbf{X})_{jj}}}$$

where $(\mathbf{X}'\mathbf{X})_{jj}$ is the (j, j)th element of $(\mathbf{X}'\mathbf{X})^{-1}$.

In general, large values of DFBETAS indicate observations that are influential in estimating a given parameter. Belsley, Kuh, and Welsch (1980) recommend 2 as a general cutoff value to indicate influential observations and $2/\sqrt{n}$ as a size-adjusted cutoff.

The following statements use the population example in the section "Polynomial Regression" on page 5440. See Figure 73.32 for the fitted regression equation. The INFLUENCE option produces the tables shown in Figure 73.50 and Figure 73.51.

```
proc reg data=USPopulation;
   model Population=Year YearSq / influence;
run;
```

Figure 73.50 Regression Using the INFLUENCE Option

The REG Procedure
Model: MODEL1
Dependent Variable: Population

Output Statistics

Obs	Residual	RStudent	Hat Diag H	Cov Ratio	DFFITS
1	-2.2837	-0.9361	0.3429	1.5519	-0.6762
2	-0.4146	-0.1540	0.2356	1.5325	-0.0855
3	0.6696	0.2379	0.1632	1.3923	0.1050
4	0.8849	0.3065	0.1180	1.3128	0.1121
5	0.5923	0.2021	0.0933	1.2883	0.0648
6	-0.0621	-0.0210	0.0831	1.2827	-0.0063
7	-0.1344	-0.0455	0.0824	1.2813	-0.0136
8	0.5864	0.1994	0.0870	1.2796	0.0615
9	0.0934	0.0318	0.0933	1.2969	0.0102
10	0.2255	0.0771	0.0990	1.3040	0.0255
11	1.4757	0.5090	0.1022	1.2550	0.1717
12	1.6441	0.5680	0.1022	1.2420	0.1916
13	3.4065	1.2109	0.0990	1.0320	0.4013
14	1.5922	0.5470	0.0933	1.2345	0.1755
15	1.7679	0.6064	0.0870	1.2123	0.1871
16	-7.5642	-3.2147	0.0824	0.3286	-0.9636
17	-7.4712	-3.1550	0.0831	0.3425	-0.9501
18	-0.3731	-0.1272	0.0933	1.2936	-0.0408
19	1.2782	0.4440	0.1180	1.2906	0.1624
20	1.0356	0.3687	0.1632	1.3741	0.1628
21	-1.7068	-0.6406	0.2356	1.4380	-0.3557
22	4.7578	2.1312	0.3429	0.9113	1.5395

Output Statistics

	DFBETAS		
Obs	Intercept	Year	YearSq
1	-0.4924	0.4862	-0.4802
2	-0.0540	0.0531	-0.0523
3	0.0517	-0.0505	0.0494
4	0.0335	-0.0322	0.0310
5	0.0040	-0.0032	0.0025
6	0.0012	-0.0012	0.0013
7	0.0054	-0.0055	0.0056
8	-0.0339	0.0343	-0.0347
9	-0.0067	0.0067	-0.0068
10	-0.0182	0.0183	-0.0183
11	-0.1272	0.1275	-0.1276
12	-0.1426	0.1426	-0.1424
13	-0.2895	0.2889	-0.2880
14	-0.1173	0.1167	-0.1160
15	-0.1076	0.1067	-0.1056
16	0.4130	-0.4063	0.3987
17	0.2131	-0.2048	0.1957
18	-0.0007	0.0012	-0.0016
19	0.0415	-0.0432	0.0449
20	0.0732	-0.0749	0.0766
21	-0.2107	0.2141	-0.2176
22	1.0656	-1.0793	1.0933

Figure 73.51 Residual Statistics

Sum of Residuals	-4.7569E-11
Sum of Squared Residuals	170.97193
Predicted Residual SS (PRESS)	237.71229

In Figure 73.50, observations 16, 17, and 19 exceed the cutoff value of 2 for RSTUDENT. None of the observations exceeds the general cutoff of 2 for DFFITS or the DFBETAS, but observations 16, 17, and 19 exceed at least one of the size-adjusted cutoffs for these statistics. Observations 1 and 19 exceed the cutoff for the hat diagonals, and observations 1, 2, 16, 17, and 18 exceed the cutoffs for COVRATIO. Taken together, these statistics indicate that you should look first at observations 16, 17, and 19 and then perhaps investigate the other observations that exceeded a cutoff.

When you enable ODS Graphics, you can request influence diagnostic plots by using the PLOTS= option in the PROC REG statement as shown in the following statements:

```
ods graphics on;

proc reg data=USPopulation
      plots(label)=(CooksD RStudentByLeverage DFFITS DFBETAS);
   id Year;
   model Population=Year YearSq;
run;

ods graphics off;
```

The LABEL suboption specified in the PLOTS(LABEL)= option requests that observations that exceed the relevant cutoffs for the statistics being plotted are labeled. Since Year has been named in an ID statement, the value of Year is used for the labels. The requested plots are shown in Figure 73.52.

Figure 73.52 Influence Diagnostics

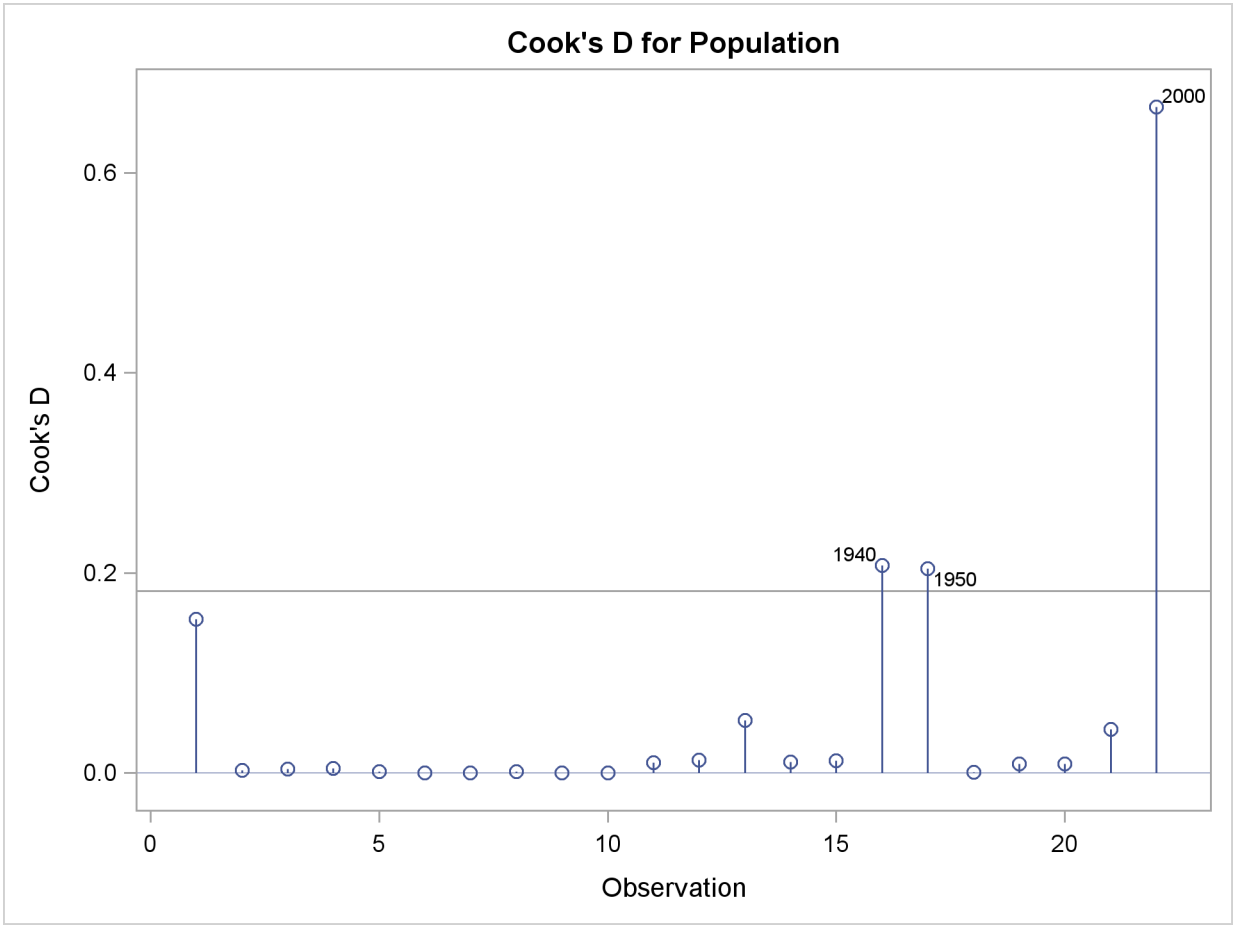


Figure 73.52 *continued*

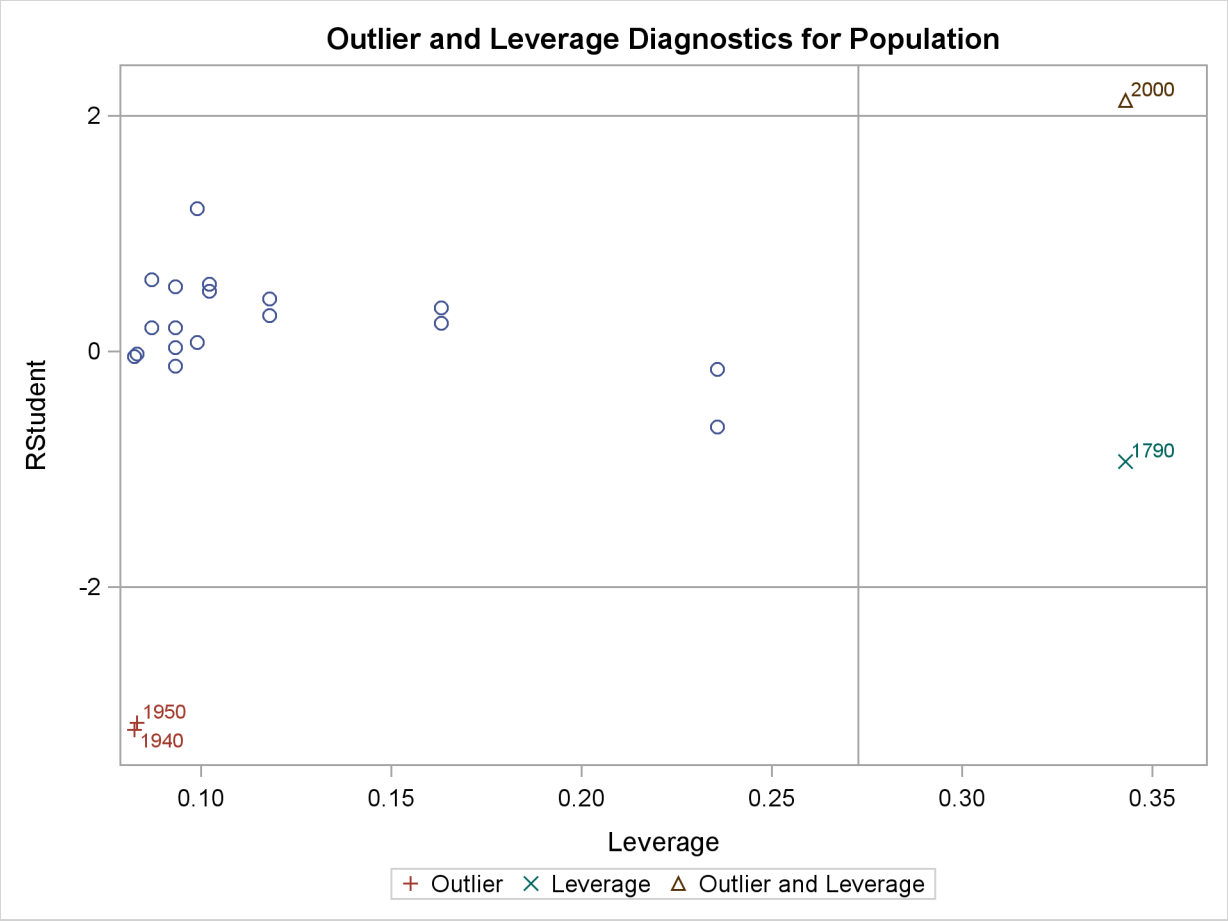


Figure 73.52 *continued*

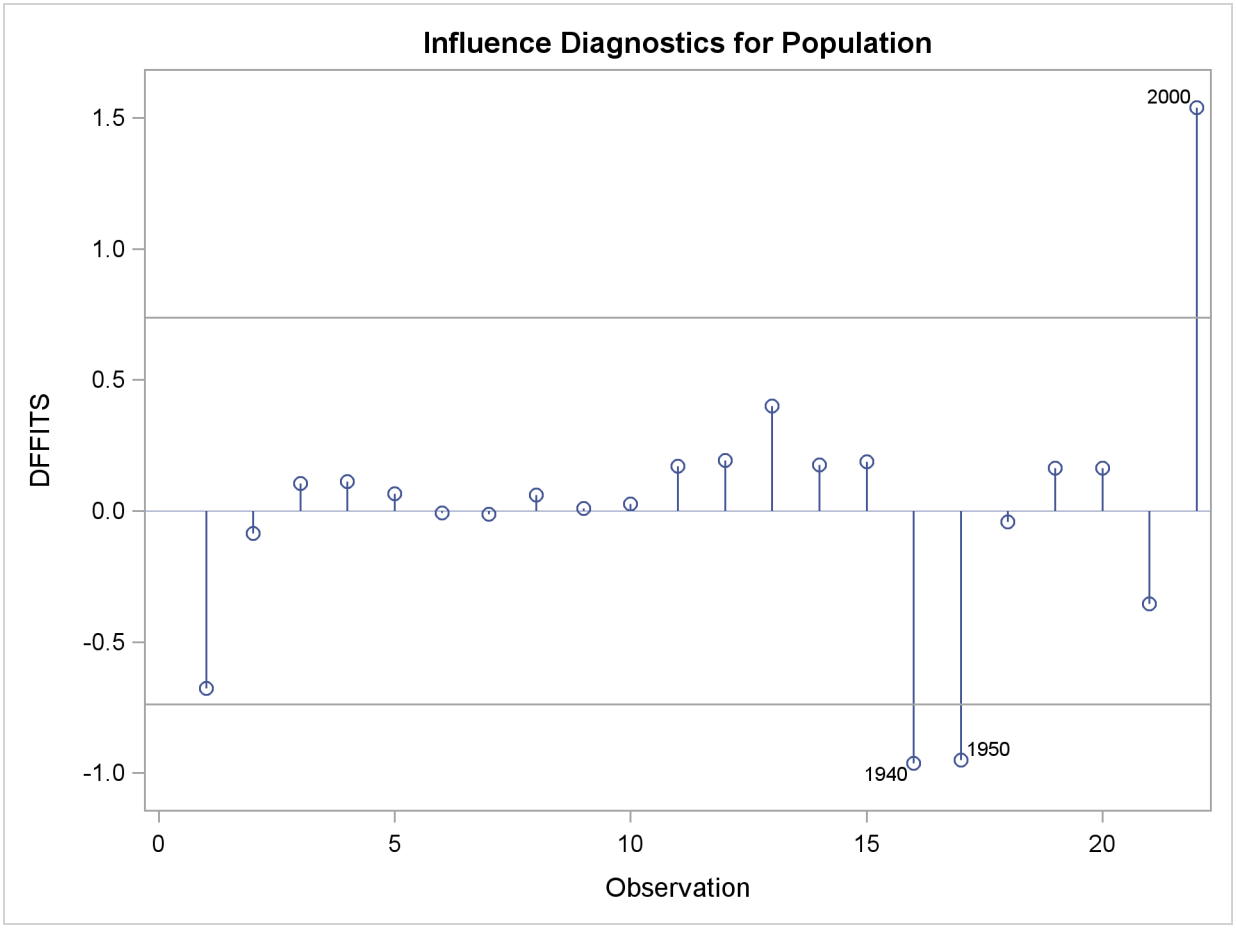
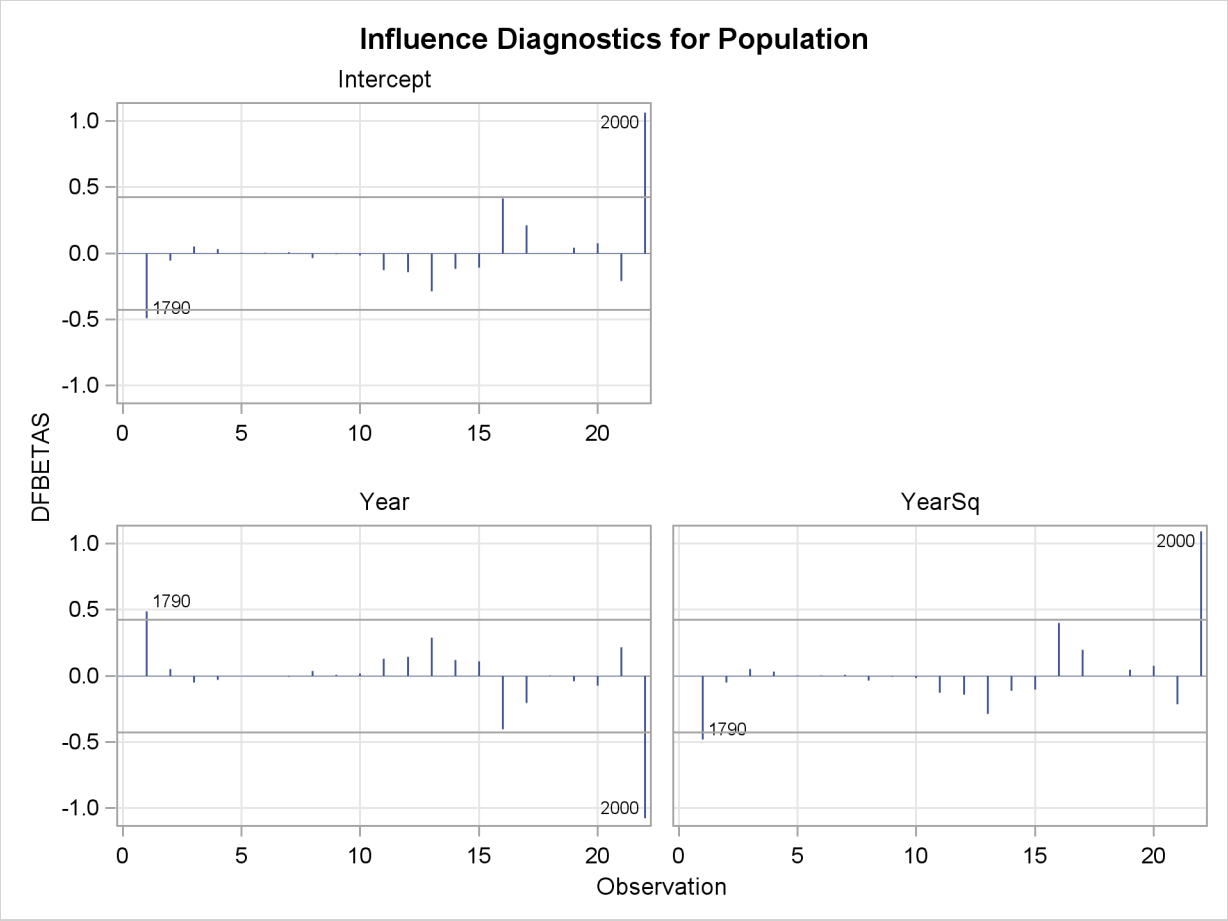


Figure 73.52 *continued*



The PARTIAL and PARTIALDATA Options

The PARTIAL option in the MODEL statement produces partial regression leverage plots. If ODS Graphics is not in effect, this option requires the use of the LINEPRINTER option in the PROC REG statement. One plot is created for each regressor in the current full model. For example, plots are produced for regressors included by using ADD statements; plots are not produced for interim models in the various model-selection methods but only for the full model. If you use a model-selection method and the final model contains only a subset of the original regressors, the PARTIAL option still produces plots for all regressors in the full model. If ODS Graphics is in effect, these plots are produced as high-resolution graphics, in panels with a maximum of six partial regression leverage plots per panel. Multiple panels are displayed for models with more than six regressors.

For a given regressor, the partial regression leverage plot is the plot of the dependent variable and the regressor after they have been made orthogonal to the other regressors in the model. These can be obtained by plotting the residuals for the dependent variable against the residuals for the selected regressor, where the residuals for the dependent variable are calculated with the selected regressor omitted, and the residuals for the selected regressor are calculated from a model where the selected regressor is regressed on the remaining regressors. A line fit to the points has a slope equal to the parameter estimate in the full model.

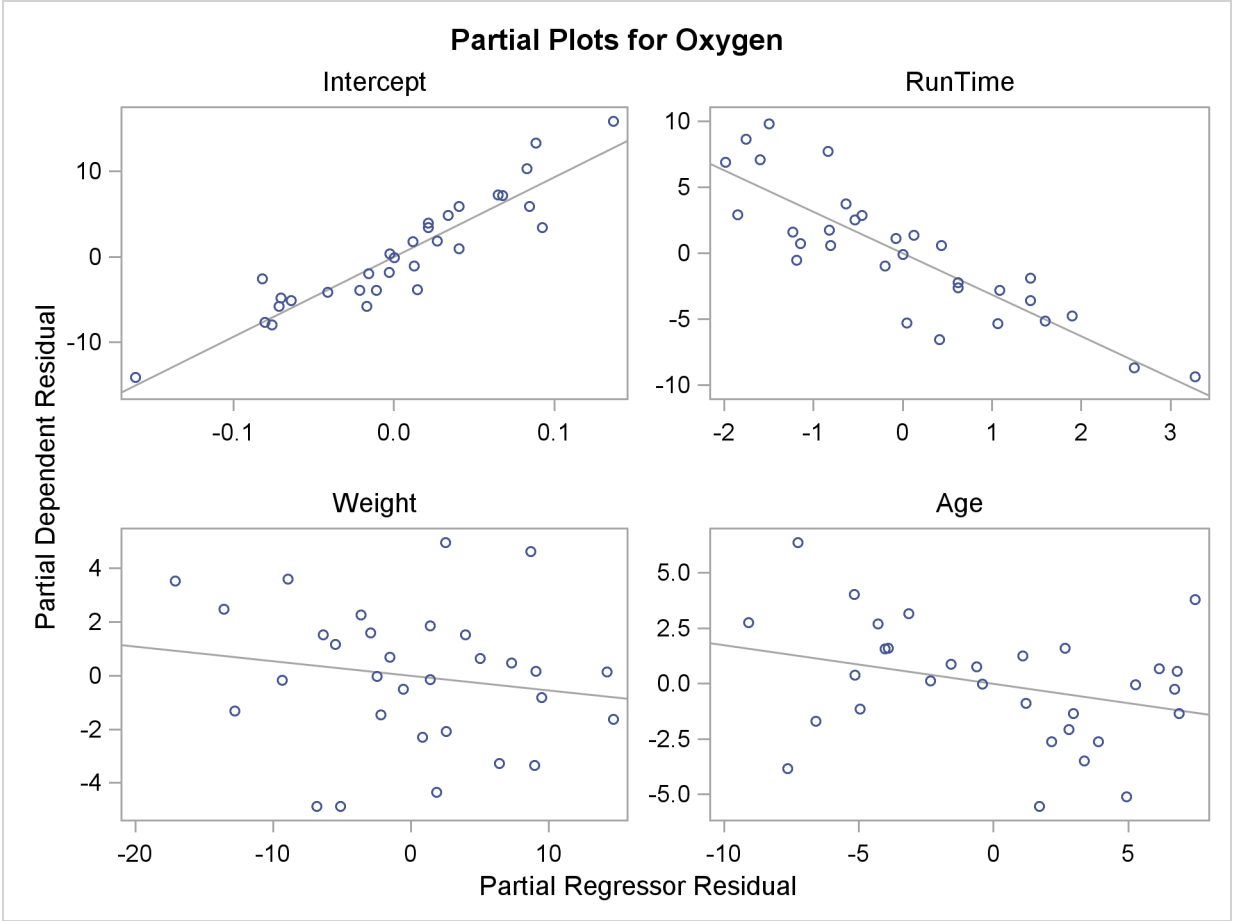
When ODS Graphics is not in effect, points in the plot are marked by the number of replicates appearing at one position. The symbol '*' is used if there are 10 or more replicates. If an ID statement is specified, the leftmost nonblank character in the value of the ID variable is used as the plotting symbol.

The PARTIALDATA option in the MODEL statement produces a table that contains the partial regression data that are displayed in the partial regression leverage plots. You can request partial regression data even if you do not requests plots with the PARTIAL option.

The following statements use the fitness data in Example 73.2 with the PARTIAL option and the ODS GRAPHICS statement to produce the partial regression leverage plots. The plots are shown in Figure 73.53.

```
ods graphics on;

proc reg data=fitness;
   model Oxygen=RunTime Weight Age / partial;
run;

ods graphics off;
```

Figure 73.53 Partial Regression Leverage Plots



Reweighting Observations in an Analysis

Reweighting observations is an interactive feature of PROC REG that enables you to change the weights of observations used in computing the regression equation. Observations can also be deleted from the analysis (not from the data set) by changing their weights to zero. In the following statements, the Class data (in the section "Getting Started: REG Procedure" on page 5436) are used to illustrate some of the features of the REWEIGHT statement. First, the full model is fit, and the residuals are displayed in Figure 73.54.

```
proc reg data=Class;
   model Weight=Age Height / p;
   id Name;
run;
```

Figure 73.54 Full Model for Class Data, Residuals Shown

The REG Procedure
Model: MODEL1
Dependent Variable: Weight

Output Statistics

Obs	Name	Dependent Variable	Predicted Value	Residual
1	Alfred	112.5000	124.8686	-12.3686
2	Alice	84.0000	78.6273	5.3727
3	Barbara	98.0000	110.2812	-12.2812
4	Carol	102.5000	102.5670	-0.0670
5	Henry	102.5000	105.0849	-2.5849
6	James	83.0000	80.2266	2.7734
7	Jane	84.5000	89.2191	-4.7191
8	Janet	112.5000	102.7663	9.7337
9	Jeffrey	84.0000	100.2095	-16.2095
10	John	99.5000	86.3415	13.1585
11	Joyce	50.5000	57.3660	-6.8660
12	Judy	90.0000	107.9625	-17.9625
13	Louise	77.0000	76.6295	0.3705
14	Mary	112.0000	117.1544	-5.1544
15	Philip	150.0000	138.2164	11.7836
16	Robert	128.0000	107.2043	20.7957
17	Ronald	133.0000	118.9529	14.0471
18	Thomas	85.0000	79.6676	5.3324
19	William	112.0000	117.1544	-5.1544

Sum of Residuals	0
Sum of Squared Residuals	2120.09974
Predicted Residual SS (PRESS)	3272.72186

Upon examining the data and residuals, you realize that observation 17 (Ronald) was mistakenly included in the analysis. Also, you would like to examine the effect of reweighting to 0.5 those observations with residuals that have absolute values greater than or equal to 17. The following statements show how you request this reweighting:

```
reweight obs.=17;
reweight r. le -17 or r. ge 17 / weight=0.5;
print p;
run;
```

At this point, a message appears (in the log) that tells you which observations have been reweighted and what the new weights are. Figure 73.55 is produced.

Figure 73.55 Model with Reweighted Observations

The REG Procedure
Model: MODEL1.2
Dependent Variable: Weight

Output Statistics

Obs	Name	Weight Variable	Dependent Variable	Predicted Value	Residual
1	Alfred	1.0000	112.5000	121.6250	-9.1250
2	Alice	1.0000	84.0000	79.9296	4.0704
3	Barbara	1.0000	98.0000	107.5484	-9.5484
4	Carol	1.0000	102.5000	102.1663	0.3337
5	Henry	1.0000	102.5000	104.3632	-1.8632
6	James	1.0000	83.0000	79.9762	3.0238
7	Jane	1.0000	84.5000	87.8225	-3.3225
8	Janet	1.0000	112.5000	103.6889	8.8111
9	Jeffrey	1.0000	84.0000	98.7606	-14.7606
10	John	1.0000	99.5000	85.3117	14.1883
11	Joyce	1.0000	50.5000	58.6811	-8.1811
12	Judy	0.5000	90.0000	106.8740	-16.8740
13	Louise	1.0000	77.0000	76.8377	0.1623
14	Mary	1.0000	112.0000	116.2429	-4.2429
15	Philip	1.0000	150.0000	135.9688	14.0312
16	Robert	0.5000	128.0000	103.5150	24.4850
17	Ronald	0	133.0000	117.8121	15.1879
18	Thomas	1.0000	85.0000	78.1398	6.8602
19	William	1.0000	112.0000	116.2429	-4.2429

Sum of Residuals	0
Sum of Squared Residuals	1500.61194
Predicted Residual SS (PRESS)	2287.57621

The first REWEIGHT statement excludes observation 17, and the second REWEIGHT statement reweights observations 12 and 16 to 0.5. An important feature to note from this example is that the model is not refit until after the PRINT statement. REWEIGHT statements do not cause the model to be refit. This is so that multiple REWEIGHT statements can be applied to a subsequent model.

In this example, since the intent is to reweight observations with large residuals, the observation that was mistakenly included in the analysis should be deleted; then the model should be fit for those remaining observations, and the observations with large residuals should be reweighted. To accomplish this, use the REFIT statement. Note that the model label has been changed from MODEL1 to MODEL1.2 since two REWEIGHT statements have been used. The following statements produce Figure 73.56:

```
reweight allobs / weight=1.0;
reweight obs.=17;
refit;
reweight r. le -17 or r. ge 17 / weight=.5;
print;
run;
```

Figure 73.56 Observations Excluded from Analysis, Model Refitted, and Observations Reweighted

The REG Procedure
Model: MODEL1.5
Dependent Variable: Weight

Output Statistics

Obs	Name	Weight Variable	Dependent Variable	Predicted Value	Residual
1	Alfred	1.0000	112.5000	120.9716	-8.4716
2	Alice	1.0000	84.0000	79.5342	4.4658
3	Barbara	1.0000	98.0000	107.0746	-9.0746
4	Carol	1.0000	102.5000	101.5681	0.9319
5	Henry	1.0000	102.5000	103.7588	-1.2588
6	James	1.0000	83.0000	79.7204	3.2796
7	Jane	1.0000	84.5000	87.5443	-3.0443
8	Janet	1.0000	112.5000	102.9467	9.5533
9	Jeffrey	1.0000	84.0000	98.3117	-14.3117
10	John	1.0000	99.5000	85.0407	14.4593
11	Joyce	1.0000	50.5000	58.6253	-8.1253
12	Judy	1.0000	90.0000	106.2625	-16.2625
13	Louise	1.0000	77.0000	76.5908	0.4092
14	Mary	1.0000	112.0000	115.4651	-3.4651
15	Philip	1.0000	150.0000	134.9953	15.0047
16	Robert	0.5000	128.0000	103.1923	24.8077
17	Ronald	0	133.0000	117.0299	15.9701
18	Thomas	1.0000	85.0000	78.0288	6.9712
19	William	1.0000	112.0000	115.4651	-3.4651

Sum of Residuals	0
Sum of Squared Residuals	1637.81879
Predicted Residual SS (PRESS)	2473.87984

Notice that this results in a slightly different model than the previous set of statements: only observation 16 is reweighted to 0.5. Also note that the model label is now MODEL1.5 since five REWEIGHT statements have been used for this model.

Another important feature of the REWEIGHT statement is the ability to nullify the effect of a previous or all REWEIGHT statements. First, assume that you have several REWEIGHT statements in effect and you want to restore the original weights of all the observations. The following REWEIGHT statement accomplishes this and produces Figure 73.57:

```
reweight allobs / reset;
print;
run;
```

Figure 73.57 Restoring Weights of All Observations

The REG Procedure
Model: MODEL1.6
Dependent Variable: Weight

Output Statistics

Obs	Name	Dependent Variable	Predicted Value	Residual
1	Alfred	112.5000	124.8686	-12.3686
2	Alice	84.0000	78.6273	5.3727
3	Barbara	98.0000	110.2812	-12.2812
4	Carol	102.5000	102.5670	-0.0670
5	Henry	102.5000	105.0849	-2.5849
6	James	83.0000	80.2266	2.7734
7	Jane	84.5000	89.2191	-4.7191
8	Janet	112.5000	102.7663	9.7337
9	Jeffrey	84.0000	100.2095	-16.2095
10	John	99.5000	86.3415	13.1585
11	Joyce	50.5000	57.3660	-6.8660
12	Judy	90.0000	107.9625	-17.9625
13	Louise	77.0000	76.6295	0.3705
14	Mary	112.0000	117.1544	-5.1544
15	Philip	150.0000	138.2164	11.7836
16	Robert	128.0000	107.2043	20.7957
17	Ronald	133.0000	118.9529	14.0471
18	Thomas	85.0000	79.6676	5.3324
19	William	112.0000	117.1544	-5.1544

Sum of Residuals	0
Sum of Squared Residuals	2120.09974
Predicted Residual SS (PRESS)	3272.72186

The resulting model is identical to the original model specified at the beginning of this section. Notice that the model label is now MODEL1.6. Note that the Weight column does not appear, since all observations have been reweighted to have weight=1.

Now suppose you want only to undo the changes made by the most recent REWEIGHT statement. Use REWEIGHT UNDO for this. The following statements produce Figure 73.58:

```
reweight r. le -12 or r. ge 12 / weight=.75;
reweight r. le -17 or r. ge 17 / weight=.5;
reweight undo;
print;
run;
```

Figure 73.58 Example of UNDO in REWEIGHT Statement

The REG Procedure
Model: MODEL1.9
Dependent Variable: Weight

Output Statistics

Obs	Name	Weight Variable	Dependent Variable	Predicted Value	Residual
1	Alfred	0.7500	112.5000	125.1152	-12.6152
2	Alice	1.0000	84.0000	78.7691	5.2309
3	Barbara	0.7500	98.0000	110.3236	-12.3236
4	Carol	1.0000	102.5000	102.8836	-0.3836
5	Henry	1.0000	102.5000	105.3936	-2.8936
6	James	1.0000	83.0000	80.1133	2.8867
7	Jane	1.0000	84.5000	89.0776	-4.5776
8	Janet	1.0000	112.5000	103.3322	9.1678
9	Jeffrey	0.7500	84.0000	100.2835	-16.2835
10	John	0.7500	99.5000	86.2090	13.2910
11	Joyce	1.0000	50.5000	57.0745	-6.5745
12	Judy	0.7500	90.0000	108.2622	-18.2622
13	Louise	1.0000	77.0000	76.5275	0.4725
14	Mary	1.0000	112.0000	117.6752	-5.6752
15	Philip	1.0000	150.0000	138.9211	11.0789
16	Robert	0.7500	128.0000	107.0063	20.9937
17	Ronald	0.7500	133.0000	119.4681	13.5319
18	Thomas	1.0000	85.0000	79.3061	5.6939
19	William	1.0000	112.0000	117.6752	-5.6752

Sum of Residuals	0
Sum of Squared Residuals	1694.87114
Predicted Residual SS (PRESS)	2547.22751

The resulting model reflects changes made only by the first REWEIGHT statement since the third REWEIGHT statement negates the effect of the second REWEIGHT statement. Observations 1, 3, 9, 10, 12, 16, and 17 have their weights changed to 0.75. Note that the label MODEL1.9 reflects the use of nine REWEIGHT statements for the current model.

Now suppose you want to reset the observations selected by the most recent REWEIGHT statement to their original weights. Use the REWEIGHT statement with the RESET option to do this. The following statements produce Figure 73.59:

```
reweight r. le -12 or r. ge 12 / weight=.75;
reweight r. le -17 or r. ge 17 / weight=.5;
reweight / reset;
print;
run;
```

Figure 73.59 REWEIGHT Statement with RESET option

The REG Procedure
Model: MODEL1.12
Dependent Variable: Weight

Output Statistics

Obs	Name	Weight Variable	Dependent Variable	Predicted Value	Residual
1	Alfred	0.7500	112.5000	126.0076	-13.5076
2	Alice	1.0000	84.0000	77.8727	6.1273
3	Barbara	0.7500	98.0000	111.2805	-13.2805
4	Carol	1.0000	102.5000	102.4703	0.0297
5	Henry	1.0000	102.5000	105.1278	-2.6278
6	James	1.0000	83.0000	80.2290	2.7710
7	Jane	1.0000	84.5000	89.7199	-5.2199
8	Janet	1.0000	112.5000	102.0122	10.4878
9	Jeffrey	0.7500	84.0000	100.6507	-16.6507
10	John	0.7500	99.5000	86.6828	12.8172
11	Joyce	1.0000	50.5000	56.7703	-6.2703
12	Judy	1.0000	90.0000	108.1649	-18.1649
13	Louise	1.0000	77.0000	76.4327	0.5673
14	Mary	1.0000	112.0000	117.1975	-5.1975
15	Philip	1.0000	150.0000	138.7581	11.2419
16	Robert	1.0000	128.0000	108.7016	19.2984
17	Ronald	0.7500	133.0000	119.0957	13.9043
18	Thomas	1.0000	85.0000	80.3076	4.6924
19	William	1.0000	112.0000	117.1975	-5.1975

Sum of Residuals	0
Sum of Squared Residuals	1879.08980
Predicted Residual SS (PRESS)	2959.57279

Note that observations that meet the condition of the second REWEIGHT statement (residuals with an absolute value greater than or equal to 17) now have weights reset to their original value of 1. Observations 1, 3, 9, 10, and 17 have weights of 0.75, but observations 12 and 16 (which meet the condition of the second REWEIGHT statement) have their weights reset to 1.

Notice how the last three examples show three ways to change weights back to a previous value. In the first example, ALLOBS and the RESET option are used to change weights for all observations back to their original values. In the second example, the UNDO option is used to negate the effect of a previous REWEIGHT statement, thus changing weights for observations selected in the previous REWEIGHT statement to the weights specified in still another REWEIGHT statement. In the third example, the RESET option is used to change weights for observations selected in a pre-

vious REWEIGHT statement back to their original values. Finally, note that the label MODEL1.12 indicates that 12 REWEIGHT statements have been applied to the original model.

Testing for Heteroscedasticity

The regression model is specified as $y_i = \mathbf{x}_i\boldsymbol{\beta} + \epsilon_i$, where the ϵ_i's are identically and independently distributed: $E(\boldsymbol{\epsilon}) = 0$ and $E(\boldsymbol{\epsilon}'\boldsymbol{\epsilon}) = \sigma^2\mathbf{I}$. If the ϵ_i's are not independent or their variances are not constant, the parameter estimates are unbiased, but the estimate of the covariance matrix is inconsistent.

In the case of heteroscedasticity, if the regression data are from a simple random sample, then White (1980), showed that matrix

$$\mathrm{HC}_0 = (\mathbf{X}'\mathbf{X})^{-1}(\mathbf{X}'\mathrm{diag}(e_i^2)\mathbf{X})(\mathbf{X}'\mathbf{X})^{-1}$$

where

$$e_i = y_i - \mathbf{x}_i\mathbf{b}$$

is an asymptotically consistent estimate of the covariance matrix. MacKinnon and White (1985) introduced three alternative heteroscedasticity-consistent covariance matrix estimators that are all asymptotically equivalent to the estimator HC_0 but that typically have better small sample behavior. These estimators labeled HC_1, HC_2, and HC_3 are defined as follows:

$$\mathrm{HC}_1 = \frac{n}{n-p}\mathrm{HC}_0$$

where n is the number of observations and p is the number of regressors including the intercept.

$$\mathrm{HC}_2 = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathrm{diag}\left(\frac{e_i^2}{1-h_{ii}}\right)\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$$

where

$$h_{ii} = \mathbf{x}_i(\mathbf{X}'\mathbf{X})^{-1}\mathbf{x}_i'$$

is the leverage of the ith observation.

$$\mathrm{HC}_3 = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathrm{diag}\left(\frac{e_i^2}{(1-h_{ii})^2}\right)\mathbf{X}(\mathbf{X}'\mathbf{X})^{-1}$$

Long and Ervin (2000) studied the performance of these estimators and recommend using the HC_3 estimator if the sample size is less than 250.

You can use the HCCMETHOD=0,1,2, or 3 in the MODEL statement to select a heteroscedasticity-consistent covariance matrix estimator, with HC_0 being the default. The ACOV option in the MODEL statement displays the heteroscedasticity-consistent covariance matrix estimator in effect and adds heteroscedasticity-consistent standard errors, also known as White standard errors, to the parameter estimates table. If you specify the HCC or WHITE option in the MODEL statement, but do not also specify the ACOV option, then the heteroscedasticity-consistent standard errors are added to the parameter estimates table but the heteroscedasticity- consistent covariance matrix is not displayed.

The SPEC option performs a model specification test. The null hypothesis for this test maintains that the errors are homoscedastic and independent of the regressors and that several technical assumptions about the model specification are valid. For details, see theorem 2 and assumptions 1–7 of White (1980). When the model is correctly specified and the errors are independent of the regressors, the rejection of this null hypothesis is evidence of heteroscedasticity. In implementing this test, an estimator of the average covariance matrix (White 1980, p. 822) is constructed and inverted. The nonsingularity of this matrix is one of the assumptions in the null hypothesis about the model specification. When PROC REG determines this matrix to be numerically singular, a generalized inverse is used and a note to this effect is written to the log. In such cases, care should be taken in interpreting the results of this test.

When you specify the SPEC, ACOV, HCC, or WHITE option in the MODEL statement, tests listed in the TEST statement are performed with both the usual covariance matrix and the heteroscedasticity-consistent covariance matrix requested with the HCCMETHOD= option. Tests performed with the consistent covariance matrix are asymptotic. For more information, refer to White (1980).

Both the ACOV and SPEC options can be specified in a MODEL or PRINT statement.

Testing for Lack of Fit

The test for lack of fit compares the variation around the model with "pure" variation within replicated observations. This measures the adequacy of the specified model. In particular, if there are n_i replicated observations $Y_{i1}, \ldots, Y_{in_i}$ of the response all at the same values $\mathbf{x}_i$ of the regressors, then you can predict the true response at $\mathbf{x}_i$ either by using the predicted value $\hat{Y}_i$ based on the model or by using the mean $\bar{Y}_i$ of the replicated values. The test for lack of fit decomposes the residual error into a component due to the variation of the replications around their mean value (the "pure" error) and a component due to the variation of the mean values around the model prediction (the "bias" error):

$$\sum_i \sum_{j=1}^{n_i} \left(Y_{ij} - \hat{Y}_i\right)^2 = \sum_i \sum_{j=1}^{n_i} \left(Y_{ij} - \bar{Y}_i\right)^2 + \sum_i n_i \left(\bar{Y}_i - \hat{Y}_i\right)^2$$

If the model is adequate, then both components estimate the nominal level of error; however, if the bias component of error is much larger than the pure error, then this constitutes evidence that there is significant lack of fit.

If some observations in your design are replicated, you can test for lack of fit by specifying the LACKFIT option in the MODEL statement (see Example 73.6). Note that, since all other tests use total error rather than pure error, you might want to hand-calculate the tests with respect to pure error if the lack of fit is significant. On the other hand, significant lack of fit indicates that the specified model is inadequate, so if this is a problem you can also try to refine the model.

Multivariate Tests

The MTEST statement described in the section "MTEST Statement" on page 5480 can test hypotheses involving several dependent variables in the form

$$(\mathbf{L}\boldsymbol{\beta} - \mathbf{cj})\mathbf{M} = 0$$

where $\mathbf{L}$ is a linear function on the regressor side, $\boldsymbol{\beta}$ is a matrix of parameters, $\mathbf{c}$ is a column vector of constants, $\mathbf{j}$ is a row vector of ones, and $\mathbf{M}$ is a linear function on the dependent side. The special case where the constants are zero is

$$\mathbf{L}\boldsymbol{\beta}\mathbf{M} = 0$$

To test this hypothesis, PROC REG constructs two matrices called $\mathbf{H}$ and $\mathbf{E}$ that correspond to the numerator and denominator of a univariate F test:

$$\begin{aligned} \mathbf{H} &= \mathbf{M}'(\mathbf{LB} - \mathbf{cj})'(\mathbf{L}(\mathbf{X}'\mathbf{X})^{-}\mathbf{L}')^{-1}(\mathbf{LB} - \mathbf{cj})\mathbf{M} \\ \mathbf{E} &= \mathbf{M}'(\mathbf{Y}'\mathbf{Y} - \mathbf{B}'(\mathbf{X}'\mathbf{X})\mathbf{B})\mathbf{M} \end{aligned}$$

These matrices are displayed for each MTEST statement if the PRINT option is specified.

Four test statistics based on the eigenvalues of $\mathbf{E}^{-1}\mathbf{H}$ or $(\mathbf{E}+\mathbf{H})^{-1}\mathbf{H}$ are formed. These are Wilks' lambda, Pillai's trace, the Hotelling-Lawley trace, and Roy's greatest root. These test statistics are discussed in Chapter 4, "Introduction to Regression Procedures."

The following code creates MANOVA data from Morrison (1976):

```
* Manova Data from Morrison (1976, 190);
data a;
input sex $ drug $ @;
do rep=1 to 4;
   input y1 y2 @;
   sexcode=(sex='m')-(sex='f');
   drug1=(drug='a')-(drug='c');
   drug2=(drug='b')-(drug='c');
   sexdrug1=sexcode*drug1;
   sexdrug2=sexcode*drug2;
   output;
end;
datalines;
m a  5  6  5  4  9  9  7  6
m b  7  6  7  7  9 12  6  8
m c 21 15 14 11 17 12 12 10
f a  7 10  6  6  9  7  8 10
f b 10 13  8  7  7  6  6  9
f c 16 12 14  9 14  8 10  5
;
```

The following statements perform a multivariate analysis of variance and produce Figure 73.60 through Figure 73.63:

```
proc reg;
   model y1 y2=sexcode drug1 drug2 sexdrug1 sexdrug2;
   y1y2drug: mtest y1=y2, drug1,drug2;
   drugshow: mtest drug1, drug2 / print canprint;
run;
```

Figure 73.60 Multivariate Analysis of Variance: REG Procedure

The REG Procedure
Model: MODEL1
Dependent Variable: y1

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	316.00000	63.20000	12.04	<.0001
Error	18	94.50000	5.25000		
Corrected Total	23	410.50000			

Root MSE	2.29129	R-Square	0.7698
Dependent Mean	9.75000	Adj R-Sq	0.7058
Coeff Var	23.50039		

Figure 73.60 *continued*

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	9.75000	0.46771	20.85	<.0001
sexcode	1	0.16667	0.46771	0.36	0.7257
drug1	1	−2.75000	0.66144	−4.16	0.0006
drug2	1	−2.25000	0.66144	−3.40	0.0032
sexdrug1	1	−0.66667	0.66144	−1.01	0.3269
sexdrug2	1	−0.41667	0.66144	−0.63	0.5366

Figure 73.61 Multivariate Analysis of Variance: REG Procedure

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	69.33333	13.86667	2.19	0.1008
Error	18	114.00000	6.33333		
Corrected Total	23	183.33333			

Root MSE	2.51661	R-Square	0.3782
Dependent Mean	8.66667	Adj R-Sq	0.2055
Coeff Var	29.03782		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	8.66667	0.51370	16.87	<.0001
sexcode	1	0.16667	0.51370	0.32	0.7493
drug1	1	−1.41667	0.72648	−1.95	0.0669
drug2	1	−0.16667	0.72648	−0.23	0.8211
sexdrug1	1	−1.16667	0.72648	−1.61	0.1257
sexdrug2	1	−0.41667	0.72648	−0.57	0.5734

Figure 73.62 Multivariate Analysis of Variance: First Test

The REG Procedure
Model: MODEL1
Multivariate Test: y1y2drug

Multivariate Statistics and Exact F Statistics

S=1 M=0 N=8

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.28053917	23.08	2	18	<.0001
Pillai's Trace	0.71946083	23.08	2	18	<.0001
Hotelling-Lawley Trace	2.56456456	23.08	2	18	<.0001
Roy's Greatest Root	2.56456456	23.08	2	18	<.0001

The four multivariate test statistics are all highly significant, giving strong evidence that the coefficients of drug1 and drug2 are not the same across dependent variables y1 and y2.

Figure 73.63 Multivariate Analysis of Variance: Second Test

The REG Procedure
Model: MODEL1
Multivariate Test: drugshow

Error Matrix (E)

94.5	76.5
76.5	114

Hypothesis Matrix (H)

301	97.5
97.5	36.333333333

Figure 73.63 *continued*

	Canonical Correlation	Adjusted Canonical Correlation	Approximate Standard Error	Squared Canonical Correlation
1	0.905903	0.899927	0.040101	0.820661
2	0.244371	.	0.210254	0.059717

Eigenvalues of Inv(E)*H
= CanRsq/(1-CanRsq)

	Eigenvalue	Difference	Proportion	Cumulative
1	4.5760	4.5125	0.9863	0.9863
2	0.0635		0.0137	1.0000

Test of H0: The canonical correlations in the current row and all that follow are zero

	Likelihood Ratio	Approximate F Value	Num DF	Den DF	Pr > F
1	0.16862952	12.20	4	34	<.0001
2	0.94028273	1.14	1	18	0.2991

Multivariate Statistics and F Approximations

S=2 M=-0.5 N=7.5

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.16862952	12.20	4	34	<.0001
Pillai's Trace	0.88037810	7.08	4	36	0.0003
Hotelling-Lawley Trace	4.63953666	19.40	4	19.407	<.0001
Roy's Greatest Root	4.57602675	41.18	2	18	<.0001

NOTE: F Statistic for Roy's Greatest Root is an upper bound.
NOTE: F Statistic for Wilks' Lambda is exact.

The four multivariate test statistics are all highly significant, giving strong evidence that the coefficients of drug1 and drug2 are not zero for both dependent variables.

Autocorrelation in Time Series Data

When regression is performed on time series data, the errors might not be independent. Often errors are autocorrelated; that is, each error is correlated with the error immediately before it. Autocorrelation is also a symptom of systematic lack of fit. The DW option provides the Durbin-Watson d statistic to test that the autocorrelation is zero:

$$d = \frac{\sum_{i=2}^{n}(e_i - e_{i-1})^2}{\sum_{i=1}^{n} e_i^2}$$

The value of d is close to 2 if the errors are uncorrelated. The distribution of d is reported by Durbin and Watson (1951). Tables of the distribution are found in most econometrics textbooks, such as Johnston (1972) and Pindyck and Rubinfeld (1981).

The sample autocorrelation estimate is displayed after the Durbin-Watson statistic. The sample is computed as

$$r = \frac{\sum_{i=2}^{n} e_i e_{i-1}}{\sum_{i=1}^{n} e_i^2}$$

This autocorrelation of the residuals might not be a very good estimate of the autocorrelation of the true errors, especially if there are few observations and the independent variables have certain patterns. If there are missing observations in the regression, these measures are computed as though the missing observations did not exist.

Positive autocorrelation of the errors generally tends to make the estimate of the error variance too small, so confidence intervals are too narrow and true null hypotheses are rejected with a higher probability than the stated significance level. Negative autocorrelation of the errors generally tends to make the estimate of the error variance too large, so confidence intervals are too wide and the power of significance tests is reduced. With either positive or negative autocorrelation, least squares parameter estimates are usually not as efficient as generalized least squares parameter estimates. For more details, refer to Judge et al. (1985, Chapter 8) and the *SAS/ETS User's Guide*.

The following SAS statements request the DWPROB option for the U.S. population data (see Figure 73.64). If you use the DW option instead of the DWPROB option, then p-values are not produced.

```
proc reg data=USPopulation;
   model Population=Year YearSq / dwProb;
run;
```

Figure 73.64 Regression Using DW Option

```
                    The REG Procedure
                      Model: MODEL1
                Dependent Variable: Population

Durbin-Watson D                 1.191
Pr < DW                        0.0050
Pr > DW                        0.9950
Number of Observations             22
1st Order Autocorrelation       0.323
```

Computations for Ridge Regression and IPC Analysis

In ridge regression analysis, the crossproduct matrix for the independent variables is centered (the NOINT option is ignored if it is specified) and scaled to one on the diagonal elements. The ridge constant k (specified with the RIDGE= option) is then added to each diagonal element of the crossproduct matrix. The ridge regression estimates are the least squares estimates obtained by using the new crossproduct matrix.

Let $\mathbf{X}$ be an $n \times p$ matrix of the independent variables after centering the data, and let $\mathbf{Y}$ be an $n \times 1$ vector corresponding to the dependent variable. Let $\mathbf{D}$ be a $p \times p$ diagonal matrix with diagonal elements as in $\mathbf{X}'\mathbf{X}$. The ridge regression estimate corresponding to the ridge constant k can be computed as

$$\mathbf{D}^{-\frac{1}{2}}(\mathbf{Z}'\mathbf{Z} + k\mathbf{I}_p)^{-1}\mathbf{Z}'\mathbf{Y}$$

where $\mathbf{Z} = \mathbf{X}\mathbf{D}^{-\frac{1}{2}}$ and $\mathbf{I}_p$ is a $p \times p$ identity matrix.

For IPC analysis, the smallest m eigenvalues of $\mathbf{Z}'\mathbf{Z}$ (where m is specified with the PCOMIT= option) are omitted to form the estimates.

For information about ridge regression and IPC standardized parameter estimates, parameter estimate standard errors, and variance inflation factors, refer to Rawlings (1988), Neter, Wasserman, and Kutner (1990), and Marquardt and Snee (1975). Unlike Rawlings (1988), the REG procedure uses the mean squared errors of the submodels instead of the full model MSE to compute the standard errors of the parameter estimates.

Construction of Q-Q and P-P Plots

If a normal probability-probability or quantile-quantile plot for the variable x is requested, the n nonmissing values of x are first ordered from smallest to largest:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$$

If a Q-Q plot is requested (with a PLOT statement of the form PLOT *yvariable*∗NQQ.), the ith-ordered value $x_{(i)}$ is represented by a point with y-coordinate $x_{(i)}$ and x-coordinate $\Phi^{-1}\left(\frac{i-0.375}{n+0.25}\right)$, where $\Phi(\cdot)$ is the standard normal distribution.

If a P-P plot is requested (with a PLOT statement of the form PLOT *yvariable*∗NPP.), the ith-ordered value $x_{(i)}$ is represented by a point with y-coordinate $\frac{i}{n}$ and x-coordinate $\Phi\left(\frac{x_{(i)}-\mu}{\sigma}\right)$, where μ is the mean of the nonmissing x-values and σ is the standard deviation. If an x-value has multiplicity k (that is, $x_{(i)} = \cdots = x_{(i+k-1)}$), then only the point $\left(\Phi\left(\frac{x_{(i)}-\mu}{\sigma}\right), \frac{i+k-1}{n}\right)$ is displayed.

Computational Methods

The REG procedure first composes a crossproducts matrix. The matrix can be calculated from input data, reformed from an input correlation matrix, or read in from an SSCP data set. For each model, the procedure selects the appropriate crossproducts from the main matrix. The normal equations formed from the crossproducts are solved by using a sweep algorithm (Goodnight 1979). The method is accurate for data that are reasonably scaled and not too collinear.

The mechanism that PROC REG uses to check for singularity involves the diagonal (pivot) elements of $\mathbf{X}'\mathbf{X}$ as it is being swept. If a pivot is less than SINGULAR*CSS, then a singularity is declared and the pivot is not swept (where CSS is the corrected sum of squares for the regressor and SINGULAR is machine dependent but is approximately 1E−7 on most machines or reset in the PROC REG statement).

The sweep algorithm is also used in many places in the model-selection methods. The RSQUARE method uses the leaps-and-bounds algorithm by Furnival and Wilson (1974).

Computer Resources in Regression Analysis

The REG procedure is efficient for ordinary regression; however, requests for optional features can greatly increase the amount of time required.

The major computational expense in the regression analysis is the collection of the crossproducts matrix. For p variables and n observations, the time required is proportional to np^2. For each model run, PROC REG needs time roughly proportional to k^3, where k is the number of regressors in the model. Include an additional nk^2 for the R, CLM, or CLI option and another nk^2 for the INFLUENCE option.

Most of the memory that PROC REG needs to solve large problems is used for crossproducts matrices. PROC REG requires $4p^2$ bytes for the main crossproducts matrix plus $4k^2$ bytes for the largest model. If several output data sets are requested, memory is also needed for buffers.

See the section "Input Data Sets" on page 5508 for information about how to use TYPE=SSCP data sets to reduce computing time.

Displayed Output

Many of the more specialized tables are described in detail in previous sections. Most of the formulas for the statistics are in Chapter 4, "Introduction to Regression Procedures," while other formulas can be found in the section "Model Fit and Diagnostic Statistics" on page 5557 and the section "Influence Statistics" on page 5559.

The analysis-of-variance table includes the following:

- the Source of the variation, Model for the fitted regression, Error for the residual error, and C Total for the total variation after correcting for the mean. The Uncorrected Total Variation is produced when the NOINT option is used.
- the degrees of freedom (DF) associated with the source
- the Sum of Squares for the term
- the Mean Square, the sum of squares divided by the degrees of freedom
- the F Value for testing the hypothesis that all parameters are zero except for the intercept. This is formed by dividing the mean square for Model by the mean square for Error.
- the Prob>F, the probability of getting a greater F statistic than that observed if the hypothesis is true. This is the significance probability.

Other statistics displayed include the following:

- Root MSE is an estimate of the standard deviation of the error term. It is calculated as the square root of the mean square error.
- Dep Mean is the sample mean of the dependent variable.
- C.V. is the coefficient of variation, computed as 100 times Root MSE divided by Dep Mean. This expresses the variation in unitless values.
- R-square is a measure between 0 and 1 that indicates the portion of the (corrected) total variation that is attributed to the fit rather than left to residual error. It is calculated as SS(Model) divided by SS(Total). It is also called the *coefficient of determination*. It is the square of the multiple correlation—in other words, the square of the correlation between the dependent variable and the predicted values.
- Adj R-square, the adjusted R^2, is a version of R^2 that has been adjusted for degrees of freedom. It is calculated as

$$\bar{R}^2 = 1 - \frac{(n-i)(1-R^2)}{n-p}$$

  where i is equal to 1 if there is an intercept and 0 otherwise, n is the number of observations used to fit the model, and p is the number of parameters in the model.

The parameter estimates and associated statistics are then displayed, and they include the following:

- the Variable used as the regressor, including the name Intercept to represent the estimate of the intercept parameter
- the degrees of freedom (DF) for the variable. There is one degree of freedom unless the model is not full rank.

- the Parameter Estimate
- the Standard Error, the estimate of the standard deviation of the parameter estimate
- T for H0: Parameter=0, the t test that the parameter is zero. This is computed as the Parameter Estimate divided by the Standard Error.
- the Prob > |T|, the probability that a t statistic would obtain a greater absolute value than that observed given that the true parameter is zero. This is the two-tailed significance probability.

If model-selection methods other than NONE, RSQUARE, ADJRSQ, and CP are used, the analysis-of-variance table and the parameter estimates with associated statistics are produced at each step. Also displayed are the following:

- C(p), which is Mallows' C_p statistic
- bounds on the condition number of the correlation matrix for the variables in the model (Berk 1977)

After statistics for the final model are produced, the following is displayed when the method chosen is FORWARD, BACKWARD, or STEPWISE:

- a Summary table listing Step number, Variable Entered or Removed, Partial and Model R-square, and C(p) and F statistics

The RSQUARE method displays its results beginning with the model containing the fewest independent variables and producing the largest R^2. Results for other models with the same number of variables are then shown in order of decreasing R^2, and so on, for models with larger numbers of variables. The ADJRSQ and CP methods group models of all sizes together and display results beginning with the model having the optimal value of adjusted $R^2$and C_p, respectively.

For each model considered, the RSQUARE, ADJRSQ, and CP methods display the following:

- Number in Model or IN, the number of independent variables used in each model
- R-square or RSQ, the squared multiple correlation coefficient

If the B option is specified, the RSQUARE, ADJRSQ, and CP methods produce the following:

- Parameter Estimates, the estimated regression coefficients

If the B option is not specified, the RSQUARE, ADJRSQ, and CP methods display the following:

- Variables in Model, the names of the independent variables included in the model

ODS Table Names

PROC REG assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 73.10 ODS Tables Produced by PROC REG

ODS Table Name	Description	Statement	Option
ACovEst	Consistent covariance of estimates matrix	MODEL	ALL, ACOV
ACovTestANOVA	Test ANOVA using ACOV estimates	TEST	ACOV (MODEL statement)
ANOVA	Model ANOVA table	MODEL	default
CanCorr	Canonical correlations for hypothesis combinations	MTEST	CANPRINT
CollinDiag	Collinearity Diagnostics table	MODEL	COLLIN
CollinDiagNoInt	Collinearity Diagnostics for no intercept model	MODEL	COLLINOINT
ConditionBounds	Bounds on condition number	MODEL	(SELECTION=BACKWARD | FORWARD | STEPWISE | MAXR | MINR) and DETAILS
Corr	Correlation matrix for analysis variables	PROC	ALL, CORR
CorrB	Correlation of estimates	MODEL	CORRB
CovB	Covariance of estimates	MODEL	COVB
CrossProducts	Bordered model **X'X** matrix	MODEL	ALL, XPX
DWStatistic	Durbin-Watson statistic	MODEL	ALL, DW
DependenceEquations	Linear dependence equations	MODEL	default if needed
Eigenvalues	MTest eigenvalues	MTEST	CANPRINT
Eigenvectors	MTest eigenvectors	MTEST	CANPRINT
EntryStatistics	Entry statistics for selection methods	MODEL	(SELECTION=BACKWARD | FORWARD | STEPWISE | MAXR | MINR) and DETAILS
ErrorPlusHypothesis	MTest error plus hypothesis matrix **H+E**	MTEST	PRINT
ErrorSSCP	MTest error matrix **E**	MTEST	PRINT
FitStatistics	Model fit statistics	MODEL	default
HypothesisSSCP	MTest hypothesis matrix	MTEST	PRINT
InvMTestCov	Inv(**L** Ginv(**X'X**) **L'**) and Inv(**Lb-c**)	MTEST	DETAILS

Table 73.10 *continued*

ODS Table Name	Description	Statement	Option
InvTestCov	Inv(**L** Ginv(**X**'**X**) **L**') and Inv(**Lb-c**)	TEST	PRINT
InvXPX	Bordered **X**'**X** inverse matrix	MODEL	I
MTestCov	**L** Ginv(**X**'**X**) **L**' and **Lb-c**	MTEST	DETAILS
MTransform	MTest matrix **M**, across dependents	MTEST	DETAILS
MultStat	Multivariate test statistics	MTEST	default
NObs	Number of observations		default
OutputStatistics	Output statistics table	MODEL	ALL, CLI, CLM, INFLUENCE, P, R
PartialData	Partial regression leverage data	MODEL	PARTIALDATA
ParameterEstimates	Model parameter estimates	MODEL	default
RemovalStatistics	Removal statistics for selection methods	MODEL	(SELECTION=BACKWARD | STEPWISE | MAXR | MINR) and DETAILS
ResidualStatistics	Residual statistics and PRESS statistic	MODEL	ALL, CLI, CLM, INFLUENCE, P, R
SelParmEst	Parameter estimates for selection methods	MODEL	SELECTION=BACKWARD | FORWARD | STEPWISE | MAXR | MINR
SelectionSummary	Selection summary for FORWARD, BACKWARD, and STEPWISE methods	MODEL	SELECTION=BACKWARD | FORWARD | STEPWISE
SeqParmEst	Sequential parameter estimates	MODEL	SEQB
SimpleStatistics	Simple statistics for analysis variables	PROC	ALL, SIMPLE
SpecTest	White's heteroscedasticity test	MODEL	ALL, SPEC
SubsetSelSummary	Selection summary for R-square, Adj-RSq, and Cp methods	MODEL	SELECTION=RSQUARE | ADJRSQ | CP
TestANOVA	Test ANOVA table	TEST	default
TestCov	**L** Ginv(**X**'**X**) **L**' and **Lb-c**	TEST	PRINT
USSCP	Uncorrected SSCP matrix for analysis variables	PROC	ALL, USSCP

ODS Graphics

This section describes the use of ODS for creating statistical graphs with the REG procedure. To request these graphs you must specify the ODS GRAPHICS statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS." The following sections describe the ODS graphical displays produced by PROC REG.

Diagnostics Panel

The "Diagnostics Panel" provides a display that you can use to get an overall assessment of your model. See Figure 73.8 for an example.

The panel contains the following plots:

- residuals versus the predicted values
- externally studentized residuals (RSTUDENT) versus the predicted values
- externally studentized residuals versus the leverage
- normal quantile-quantile plot (Q-Q plot) of the residuals
- dependent variable values versus the predicted values
- Cook's D versus observation number
- histogram of the residuals
- "Residual-Fit" (or RF) plot consisting of side-by-side quantile plots of the centered fit and the residuals
- box plot of the residuals if you specify the STATS=NONE suboption

Patterns in the plots of residuals or studentized residuals versus the predicted values, or spread of the residuals being greater than the spread of the centered fit in the RF plot, are indications of an inadequate model. Patterns in the spread about the 45-degree reference line in the plot of the dependent variable values versus the predicted values are also indications of an inadequate model.

The Q-Q plot, residual histogram, and box plot of the residuals are useful for diagnosing violations of the normality and homoscedasticity assumptions. If the data in a Q-Q plot come from a normal distribution, the points will cluster tightly around the reference line. A normal density is overlaid on the residual histogram to help in detecting departures form normality.

Following Rawlings (1998), reference lines are shown on the relevant plots to identify observations deemed outliers or influential. Observations whose externally studentized residual magnitudes exceed 2 are deemed outliers. Observations whose leverage value exceeds $2p/n$ or whose Cook's D value exceeds $4/n$ are deemed influential (p is the number of regressors including the intercept, and n is the number of observations used in the analysis). If you specify the LABEL suboption of

the PLOTS=DIAGNOSTICS option, then the points deemed outliers or influential are labeled on the appropriate plots.

Fit statistics are shown in the lower right of the plot and can be customized or suppressed by using the STATS= suboption of the PLOTS=DIAGNOSTICS option.

Residuals by Regressor Plots

Panels of plots of the residuals versus each of the regressors in the model are produced by default. Patterns in these plots are indications of an inadequate model. To help in detecting patterns, you can use the SMOOTH= suboption of the PLOTS=RESIDUALS option to add loess fits to these residual plots. See Figure 73.1.6 for an example.

Fit and Prediction Plots

A fit plot consisting of a scatter plot of the data overlaid with the regression line, as well as confidence and prediction limits, is produced for models depending on a single regressor. Fit statistics are shown to the right of the plot and can be customized or suppressed by using the STATS= suboption of the PLOTS=FIT option.

When a model contains more than one regressor, a fit plot is not appropriate. However, if all the regressors in the model are transformations of a single variable in the input data set, then you can request a scatter plot of the dependent variable overlaid with a fit line and confidence and prediction limits versus this variable. You can also plot residuals versus this variable. You request these plots, shown in a panel, with the PLOTS=PREDICTION option. See Figure 73.13 for an example.

Influence Plots

In addition to the "Cook's D Plot" and the "RStudent By Leverage Plot," you can request plots of the DFBETAS and DFFITS statistics versus observation number by using the PLOTS=DFBETAS and PLOTS=DFFITS options. You can also obtain partial regression leverage plots by using the PLOTS=PARTIAL option. See the section "Influence Statistics" on page 5559 for examples of these plots and details about their interpretation.

Ridge and VIF Plots

When you use ridge regression, you can request plots of the variance inflation factor (VIF) values and standardized ridge estimates by ridge values for each coefficient with the PLOTS=RIDGE option. See Example 73.5 for examples.

Variable Selection Plots

When you request variable selection by using the SELECTION= option in the MODEL statement, you can request plots of fit criteria for the models examined by using the PLOTS=CRITERIA

option. The fit criteria are displayed versus the step number for the FORWARD, BACKWARD, and STEPWISE selection methods and the step at which the optimal value of each criterion is obtained is indicated using a "Star" marker. For the all-subset-based selection methods (SELECTION=RSQUARE|ADJRSQ|CP), the fit criteria are displayed versus the number of observations in the model.

The criteria are shown in a panel, but you can use the UNPACK suboption of the PLOTS=CRITERIA option to obtain separate plots for each criterion. You can also use the LABEL suboption of the PLOTS=CRITERIA option to request that optimal models be labeled on the plots. Example 73.2 provides several examples.

ODS Graph Names

PROC REG assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 73.11.

To request these graphs you must specify the ODS GRAPHICS statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Table 73.11 ODS Graphical Displays Produced by PROC REG

ODS Graph Name	Plot Description	PLOTS Option
AdjrsqPlot	Adjusted R-square statistic for models examined doing variable selection	ADJRSQ
AICPlot	AIC statistic for models examined doing variable selection	AIC
BICPlot	BIC statistic for models examined doing variable selection	BIC
CooksDPlot	Cook's D statistic versus observation number	COOKSD
CPPlot	C_p statistic for models examined doing variable selection	CP
DFFITSPlot	DFFITS statistics versus observation number	DFFITS
DFBETASPanel	Panel of DFBETAS statistics versus observation number	DFBETAS
DFBETASPlot	DFBETAS statistics versus observation number	DFBETAS(UNPACK)
DiagnosticsPanel	Panel of fit diagnostics	DIAGNOSTICS
FitPlot	Regression line, confidence limits, and prediction limits overlaid on scatter plot of data	FIT
ObservedByPredicted	Dependent variable versus predicted values	OBSERVEDBYPREDICTED
PartialPlot	Partial regression plot	PARTIAL
PredictionPanel	Panel of residuals and fit versus specified variable	PREDICTIONS

Table 73.11 *continued*

ODS Graph Name	Plot Description	PLOTS Option
PredictionPlot	Regression line, confidence limits, and prediction limits versus specified variable	PREDICTIONS(UNPACK)
PredictionResidualPlot	Residuals versus specified variable	PREDICTIONS(UNPACK)
QQPlot	Normal quantile plot of residuals	QQ
ResidualBoxPlot	Box plot of residuals	BOXPLOT
ResidualByPredicted	Residuals versus predicted values	RESIDUALBYPREDICTED
ResidualHistogram	Histogram of fit residuals	RESIDUALHISTOGRAM
ResidualPlot	Plot of residuals versus regressor	RESIDUALS
RFPlot	Side-by-side plots of quantiles of centered fit and residuals	RF
RidgePanel	Plot of VIF and ridge traces	RIDGE
RidgePlot	Plot of ridge traces	RIDGE(UNPACK)
RSquarePlot	R-square statistic for models examined doing variable selection	RSQUARE
RStudentByLeverage	Studentized residuals versus leverage	RSTUDENTBYLEVERAGE
RStudentByPredicted	Studentized residuals versus predicted values	RSTUDENTBYPREDITED
SBCPlot	SBC statistic for models examined doing variable selection	SBC
SelectionCriterionPanel	Panel of fit statistics for models examined doing variable selection	CRITERIA
VIFPlot	Plot of VIF traces	RIDGE(UNPACK)

Examples: REG Procedure

Example 73.1: Modeling Salaries of Major League Baseball Players

This example features the use of ODS Graphics in the process of building models by using the REG procedure and highlights the use of fit and influence diagnostics.

The following data set contains salary and performance information for Major League Baseball players who played at least one game in both the 1986 and 1987 seasons, excluding pitchers. The salaries (*Sports Illustrated,* April 20, 1987) are for the 1987 season and the performance measures are from 1986 (Collier Books, *The 1987 Baseball Encyclopedia Update*).

```
 data baseball;
   length name $ 18;
   length team $ 12;
   input name $ 1-18 no_atbat no_hits no_home no_runs no_rbi no_bb yr_major
         cr_atbat cr_hits cr_home cr_runs cr_rbi cr_bb league $
         division $ team $ position $ no_outs no_assts no_error salary;
   logSalary = log10(salary);
   label name="Player's Name"
      no_hits="Hits in 1986"
      no_runs="Runs in 1986"
      no_rbi="RBIs in 1986"
      no_bb="Walks in 1986"
      yr_major="Years in MLB"
      cr_hits="Career Hits"
      salary="1987 Salary in $ Thousands"
      logSalary = "log10(Salary)";
   datalines;
Allanson, Andy       293    66     1    30    29    14
                       1   293    66     1    30    29    14
                    American East Cleveland C 446 33 20 .
Ashby, Alan          315    81     7    24    38    39
                      14  3449   835    69   321   414   375
                    National West Houston C 632 43 10 475
Davis, Alan          479   130    18    66    72    76
                       3  1624   457    63   224   266   263
                    American West Seattle 1B 880 82 14 480
Dawson, Andre        496   141    20    65    78    37
                      11  5628  1575   225   828   838   354
                    National East Montreal RF 200 11 3 500
Galarraga, Andres    321    87    10    39    42    30
                       2   396   101    12    48    46    33
                    National East Montreal 1B 805 40 4 91.5
Griffin, Alfredo     594   169     4    74    51    35
                      11  4408  1133    19   501   336   194

... more lines ...

Wilson, Willie       631   170     9    77    44    31
                      11  4908  1457    30   775   357   249
                    American West KansasCity CF 408 4 3 1000
;
```

Suppose you want to investigate whether you can model the players' salaries for the 1987 season based on batting statistics for the previous season and lifetime batting performance. Since the variation in salaries is much greater for higher salaries, it is appropriate to apply a log transformation for this analysis. The following statements begin the analysis:

```
ods graphics on;

proc reg data=baseball;
   id name team league;
   model logSalary = no_hits no_runs no_rbi no_bb yr_major cr_hits;
run;
```

Output 73.1.1 shows the default output produced by PROC REG. The number of observations table shows that 59 observations are excluded because they have missing values for at least one of the variables used in the analysis. The analysis of variance and parameter estimates tables provide details about the fitted model.

Output 73.1.1 Default Output from PROC REG

The REG Procedure
Model: MODEL1
Dependent Variable: logSalary log10(Salary)

Number of Observations Read	322
Number of Observations Used	263
Number of Observations with Missing Values	59

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	22.92208	3.82035	60.56	<.0001
Error	256	16.14954	0.06308		
Corrected Total	262	39.07162			

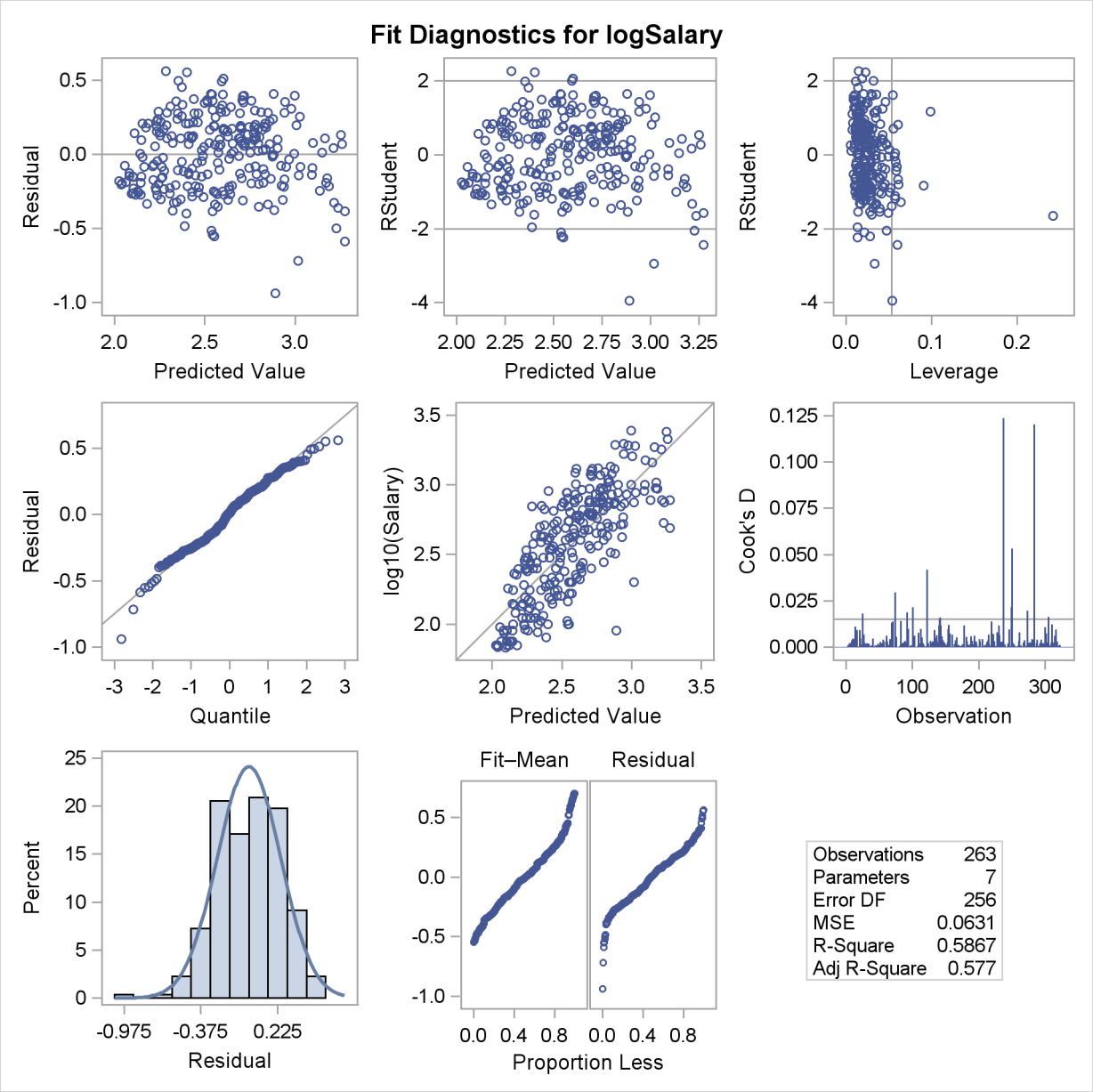
Root MSE	0.25117	R-Square	0.5867
Dependent Mean	2.57416	Adj R-Sq	0.5770
Coeff Var	9.75719		

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	1.80065	0.05912	30.46	<.0001
no_hits	Hits in 1986	1	0.00288	0.00091244	3.15	0.0018
no_runs	Runs in 1986	1	0.00008638	0.00173	0.05	0.9602
no_rbi	RBIs in 1986	1	0.00054382	0.00102	0.53	0.5947
no_bb	Walks in 1986	1	0.00292	0.00104	2.81	0.0054
yr_major	Years in MLB	1	0.03087	0.00836	3.69	0.0003
cr_hits	Career Hits	1	0.00010384	0.00006328	1.64	0.1020

Before you accept a regression model, it is important to examine influence and fit diagnostics to see whether the model might be unduly influenced by a few observations and whether the data support the assumptions that underlie the linear regression. To facilitate such investigations, you can obtain diagnostic plots by enabling ODS Graphics.

Output 73.1.2 Fit Diagnostics

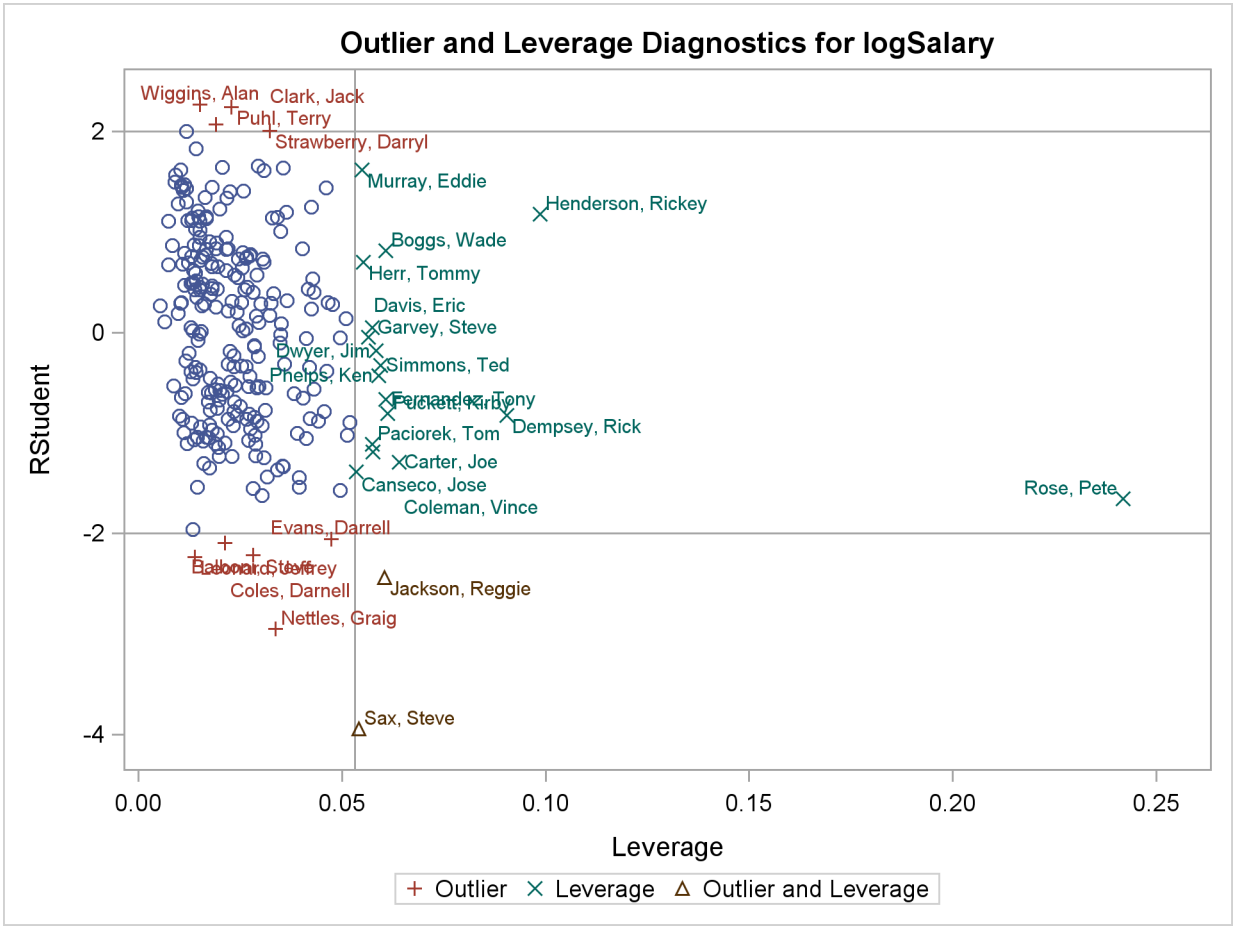


Output 73.1.2 shows a panel of diagnostic plots. The plot of externally studentized residuals (RStudent) by leverage values reveals that there is one observation with very high leverage that might be overly influencing the fit produced. The plot of Cook's D by observation also indicates two highly influential observations. To investigate further, you can use the PLOTS= option in the PROC REG statement as follows to produce labeled versions of these plots:

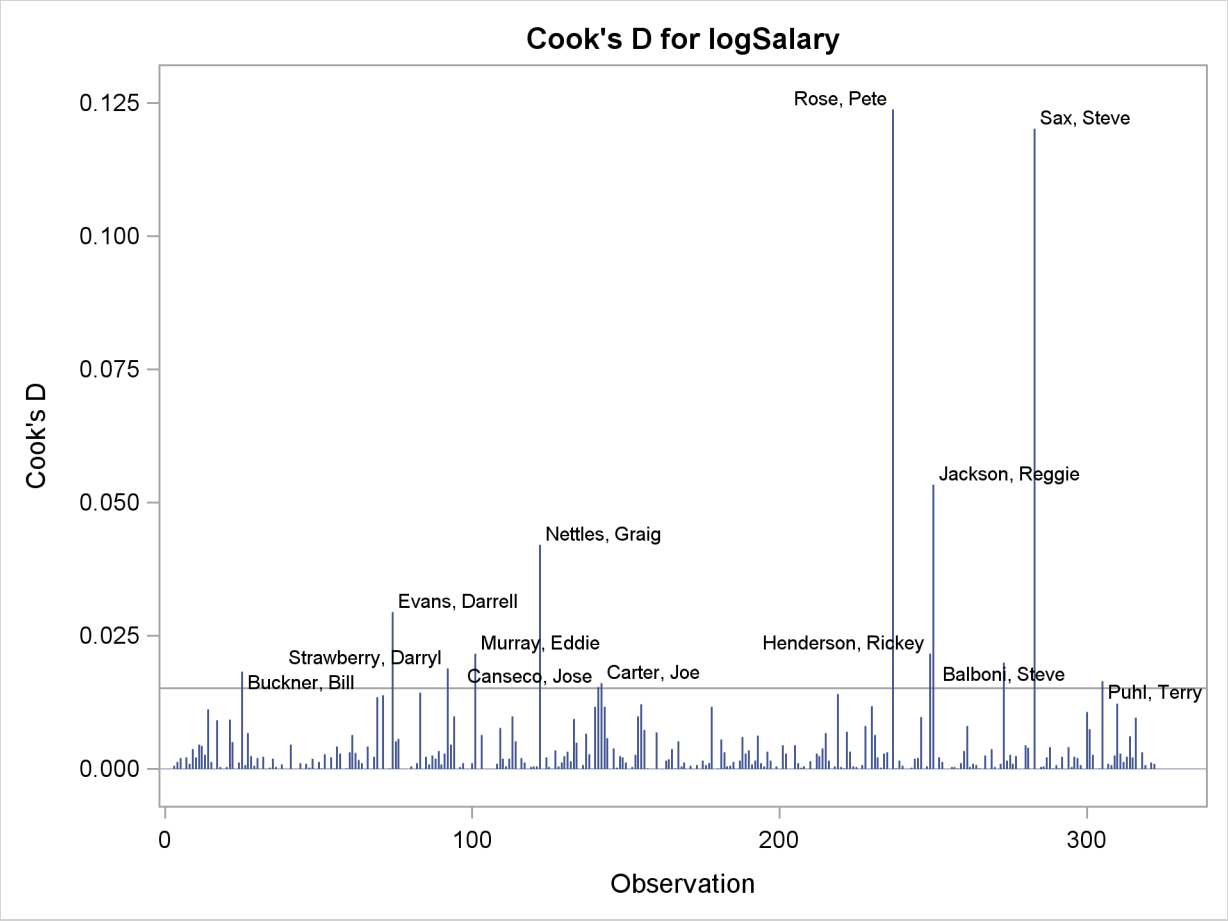
```
proc reg data=baseball
         plots(only label)=(RStudentByLeverage CooksD);
   id name team league;
   model logSalary = no_hits no_runs no_rbi no_bb yr_major cr_hits;
run;
```

Output 73.1.3 and Output 73.1.4 reveal that Pete Rose is the highly influential observation. You might obtain a better fit to the remaining data if you omit his statistics when building the model.

Output 73.1.3 Outlier and Leverage Diagnostics



Output 73.1.4 Cook's D

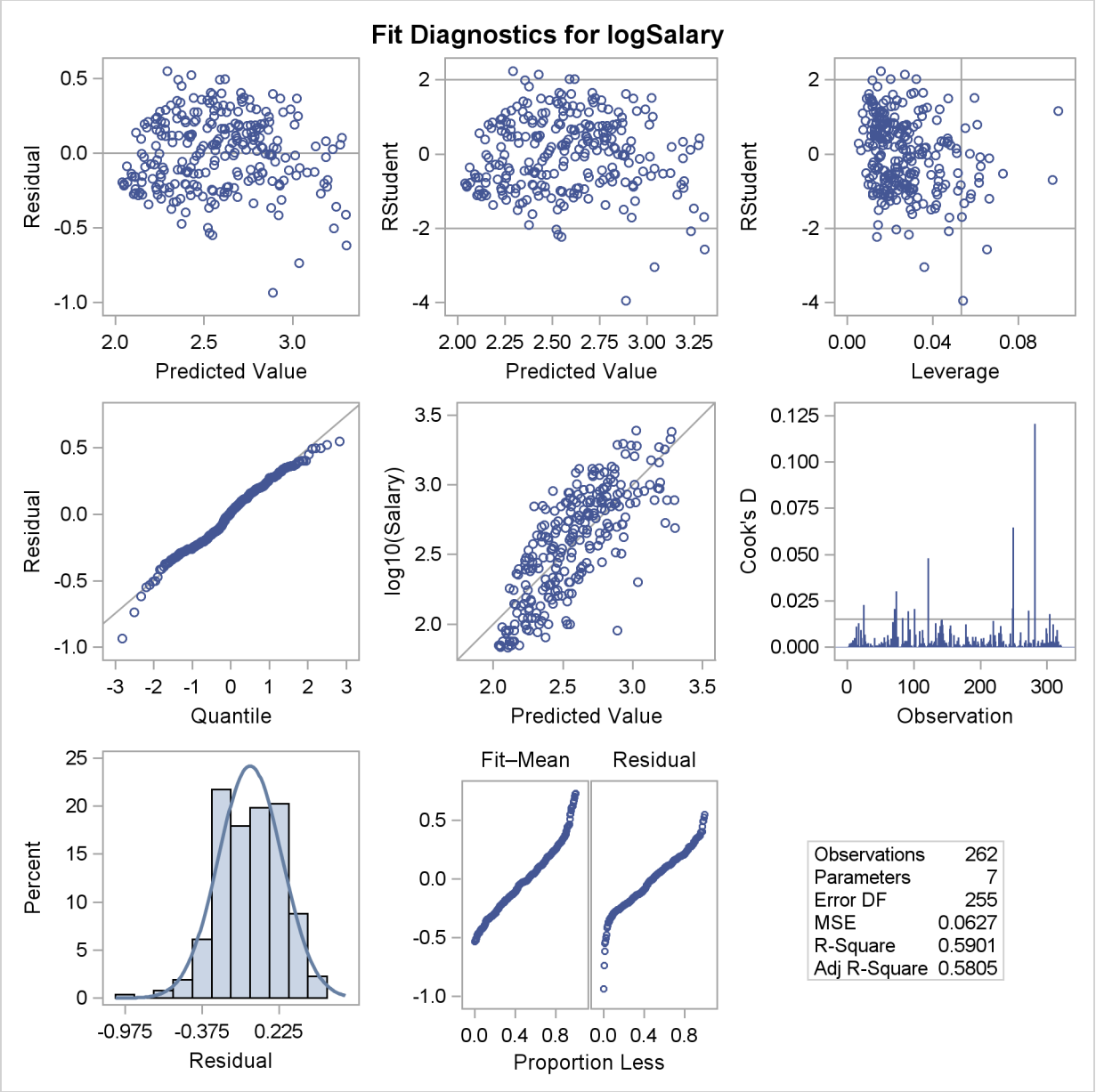


The following statements use a WHERE statement to omit Pete Rose's statistics when building the model. An alternative way to do this within PROC REG is to use a REWEIGHT statement. See "Reweighting Observations in an Analysis" on page 5569 for details about reweighting.

```
proc reg data=baseball
         plots=(RStudentByLeverage(label) residuals(smooth));
   where name^="Rose, Pete";
   id name team league;
   model logSalary = no_hits no_runs no_rbi no_bb yr_major cr_hits;
run;
```

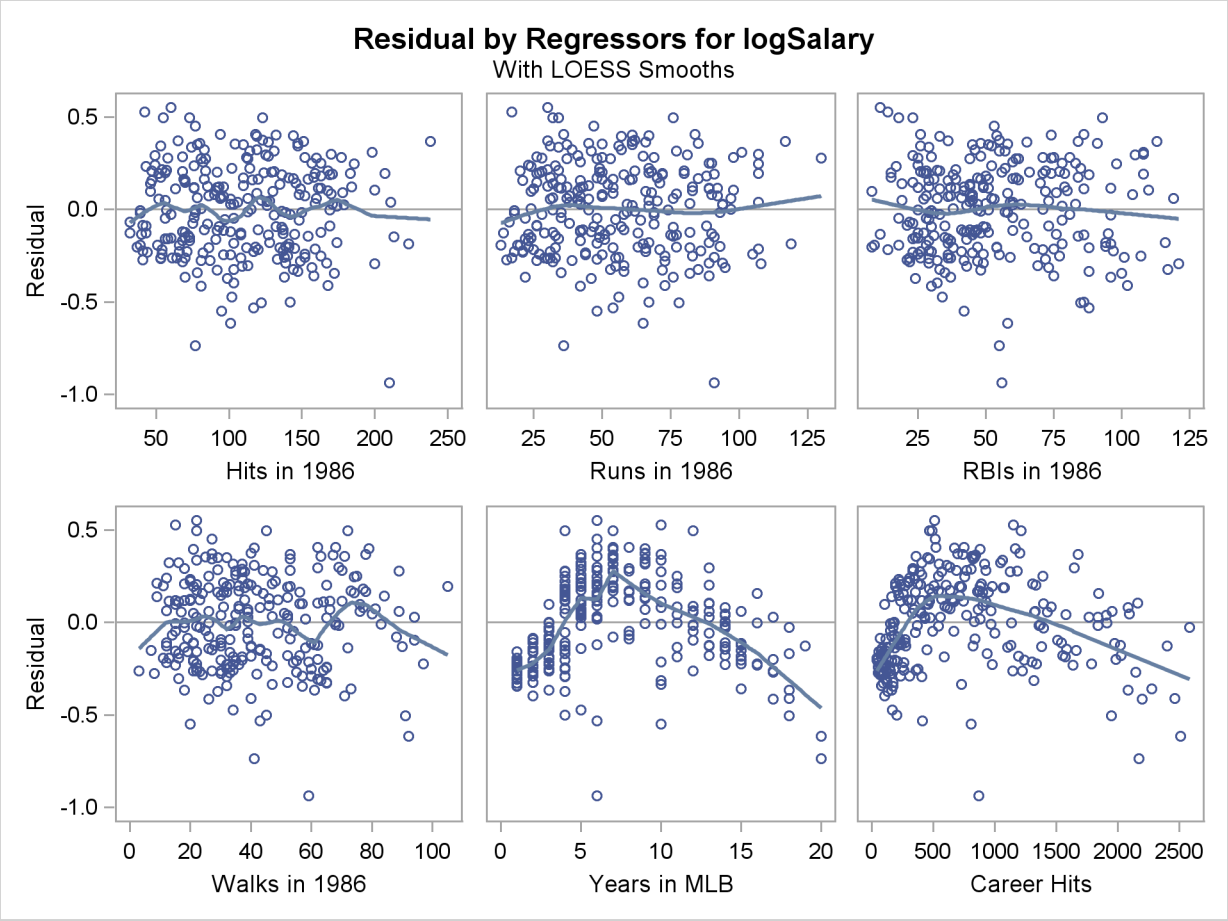
Output 73.1.5 shows the new fit diagnostics panel. You can see that there are still several influential and outlying observations. One possible reason for observing outliers is that the linear model specified is not appropriate to capture the variation in this data. You can often see evidence of an inappropriate model by observing patterns in plots of residuals.

Output 73.1.5 Fit Diagnostics



Output 73.1.6 shows plots of the residuals by the regressors in the model. When you specify the RESIDUALS(SMOOTH) suboption of the PLOTS option in the PROC REG statement, a loess fit is overlaid on each of these plots. You can see the same clear pattern in the residual plots for yr_major and cr_hits. Players near the start of their careers and players near the end of their careers get paid less than the model predicts.

Output 73.1.6 Residuals by Regressors



You can address this lack of fit by using polynomials of degree 2 for these two variables as shown in the following statements:

```
data baseball;
   set baseball(where=(name^="Rose, Pete")) ;
   yr_major2 = yr_major*yr_major;
   cr_hits2  = cr_hits*cr_hits;
run;

proc reg data=baseball
      plots=(diagnostics(stats=none) RStudentByLeverage(label)
             CooksD(label) Residuals(smooth)
             DFFITS(label) DFBETAS ObservedByPredicted(label));
   id name team league;
   model logSalary = no_hits no_runs no_rbi no_bb yr_major cr_hits
                     yr_major2 cr_hits2;
run;
ods graphics off;
```

Output 73.1.7 shows the analysis of variance and parameter estimates for this model. Note that the R-square value of 0.787 for this model is considerably larger than the R-square value of 0.587 for the initial model shown in Output 73.1.1.

Output 73.1.7 Output from PROC REG

The REG Procedure
Model: MODEL1
Dependent Variable: logSalary log10(Salary)

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	8	30.69367	3.83671	117.13	<.0001
Error	253	8.28706	0.03276		
Corrected Total	261	38.98073			

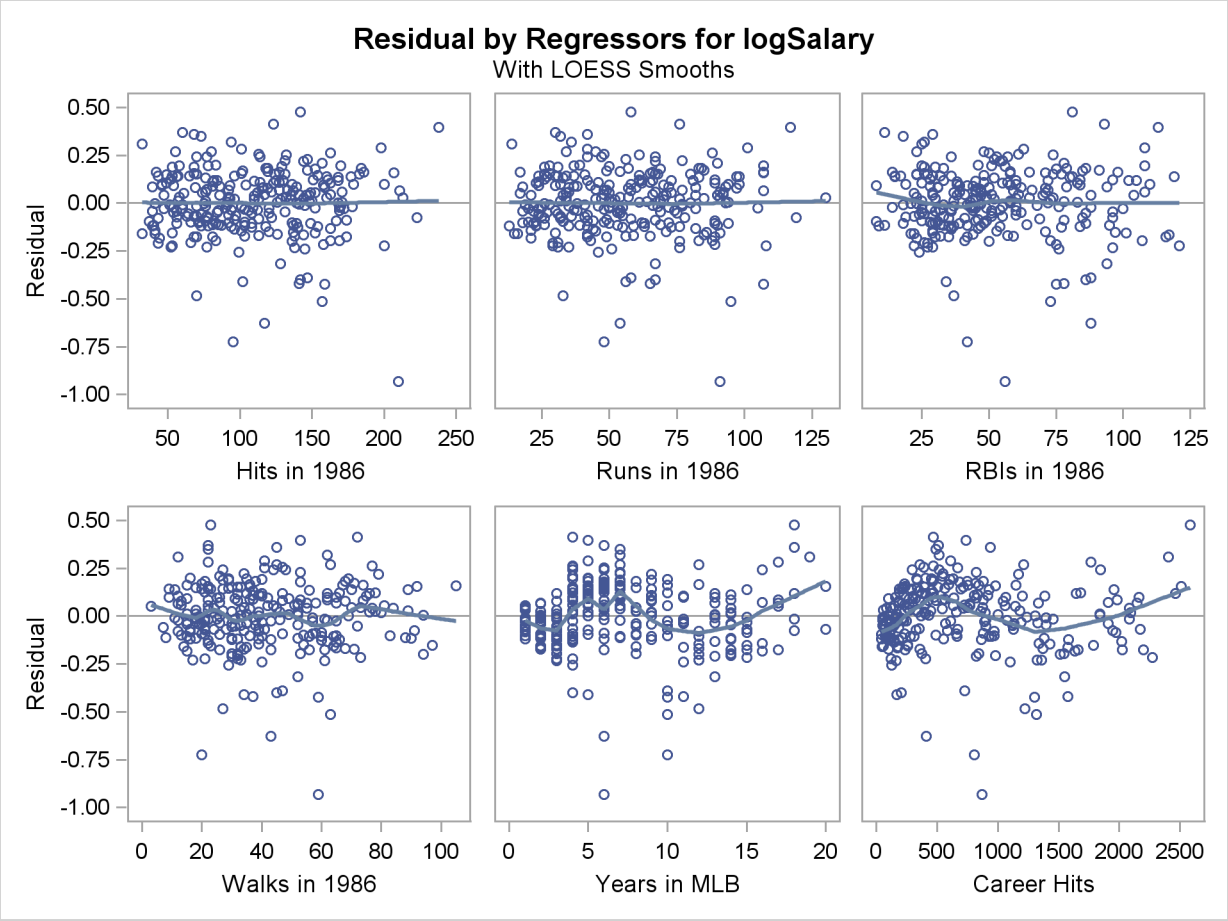
Root MSE	0.18098	R-Square	0.7874
Dependent Mean	2.57301	Adj R-Sq	0.7807
Coeff Var	7.03393		

Parameter Estimates

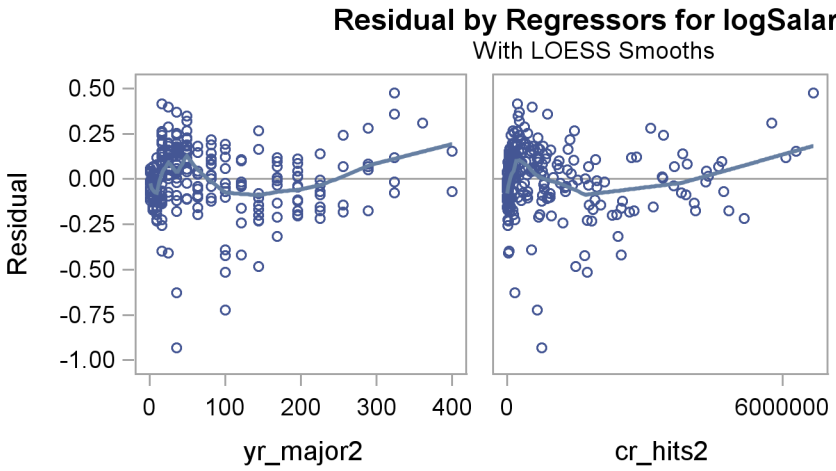
Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	1.64564	0.05030	32.72	<.0001
no_hits	Hits in 1986	1	-0.00005539	0.00069200	-0.08	0.9363
no_runs	Runs in 1986	1	0.00093586	0.00125	0.75	0.4549
no_rbi	RBIs in 1986	1	0.00187	0.00074649	2.51	0.0127
no_bb	Walks in 1986	1	0.00218	0.00075057	2.90	0.0040
yr_major	Years in MLB	1	0.10383	0.01495	6.94	<.0001
cr_hits	Career Hits	1	0.00073955	0.00011970	6.18	<.0001
yr_major2		1	-0.00625	0.00071687	-8.73	<.0001
cr_hits2		1	-1.44072E-7	4.348471E-8	-3.31	0.0011

The plots of residuals by regressors in Output 73.1.8 and Output 73.1.9 show that the strong pattern in the plots for cr_majors and cr_hits has been reduced, although there is still some indication of a pattern remaining in these residuals. This suggests that a quadratic function might be insufficient to capture dependence of salary on these regressors.

Output 73.1.8 Residuals by Regressors

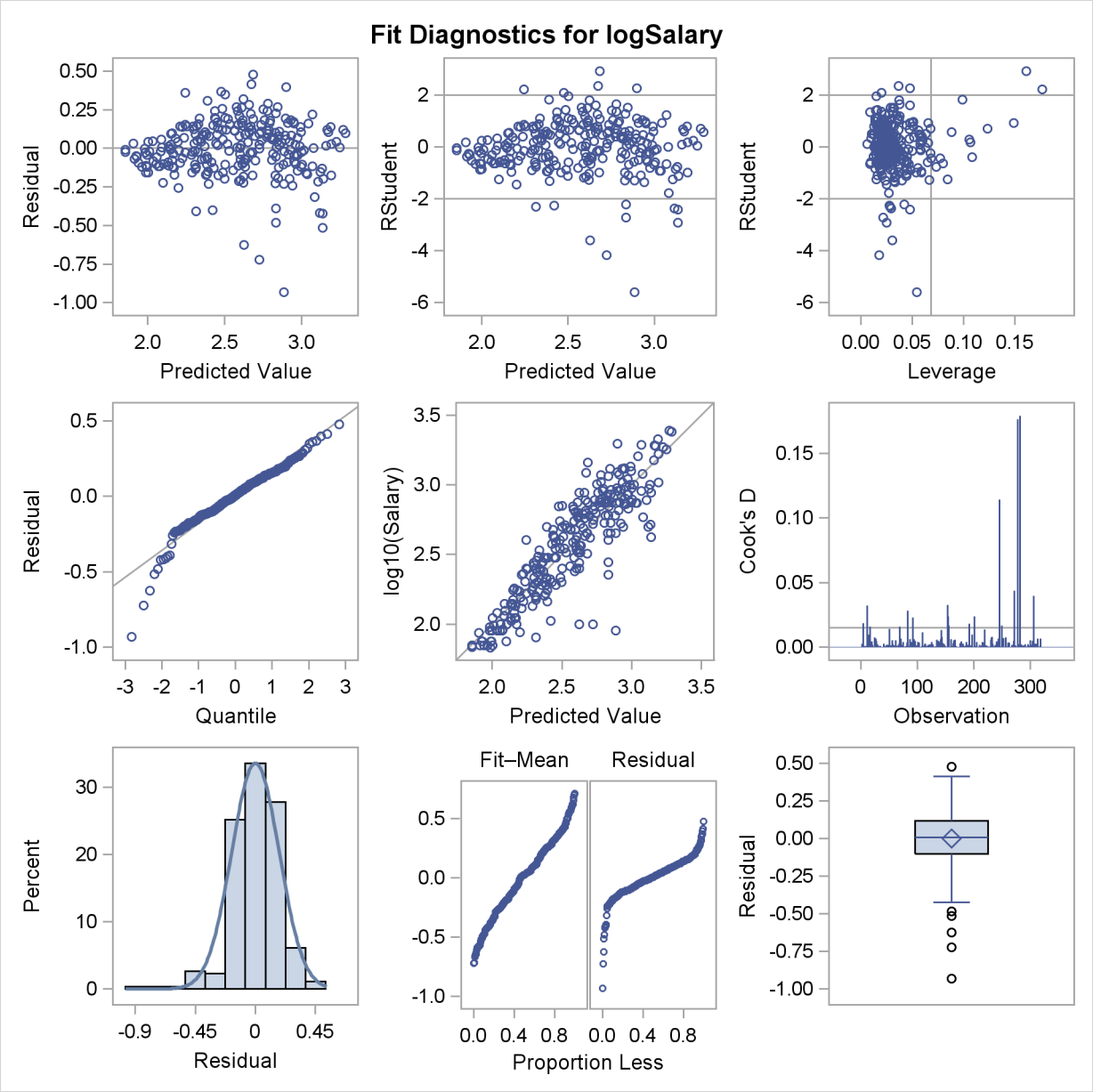


Output 73.1.9 Residuals by Regressors

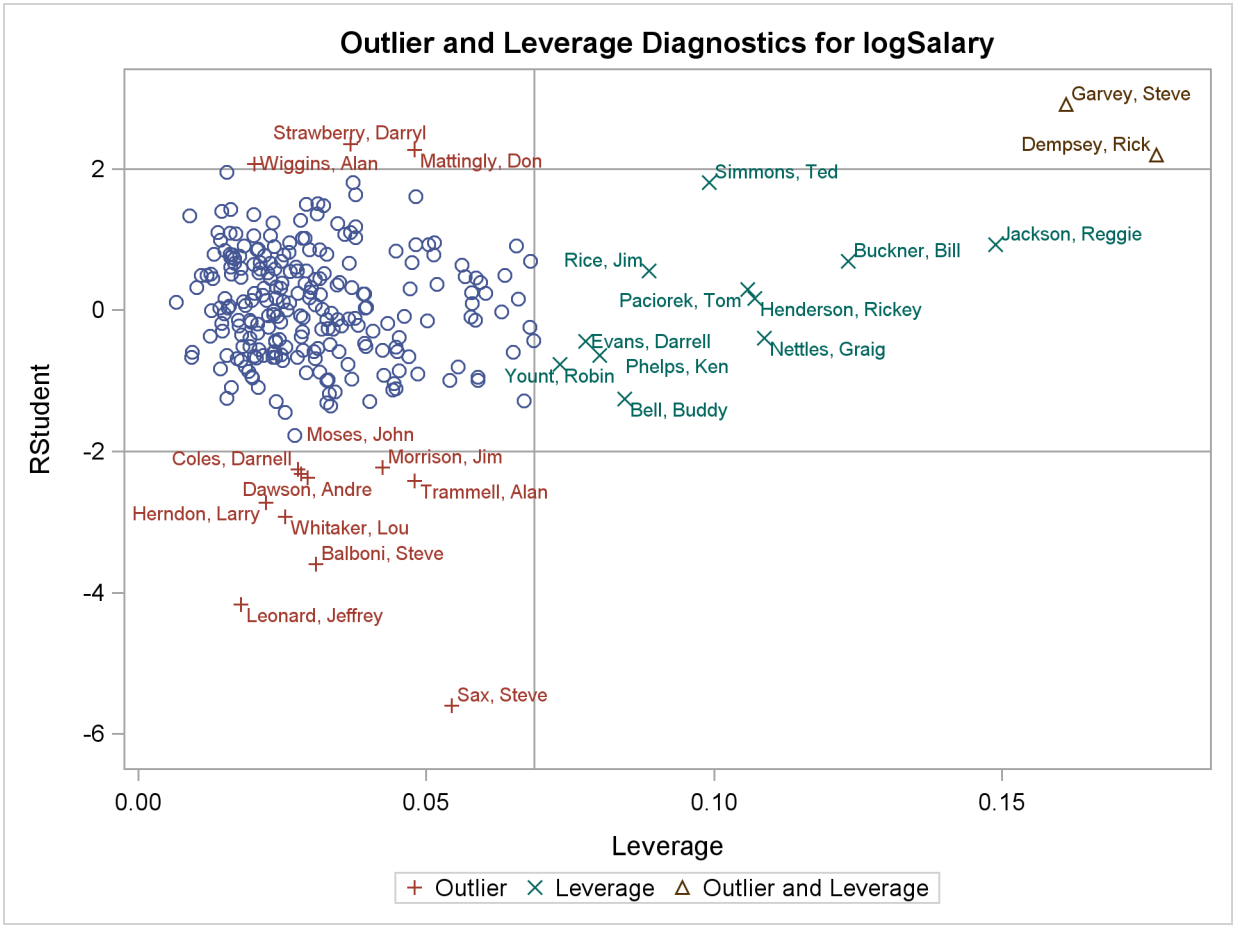


Output 73.1.10 show the diagnostics plots; three of the plots, with points of interest labeled, are shown individually in Output 73.1.11, Output 73.1.12, and Output 73.1.13. The STATS=NONE suboption specified in the PLOTS=DIAGNOSTICS option replaces the inset of statistics with a box plot of the residuals in the fit diagnostics panel. The observed by predicted value plot reveals a reasonably successful model for explaining the variation in salary for most of the players. However, the model tends to overpredict the salaries of several players near the lower end of the salary range. This bias can also be seen in the distribution of the residuals that you can see in the histogram, Q-Q plot, and box plot in Output 73.1.10.

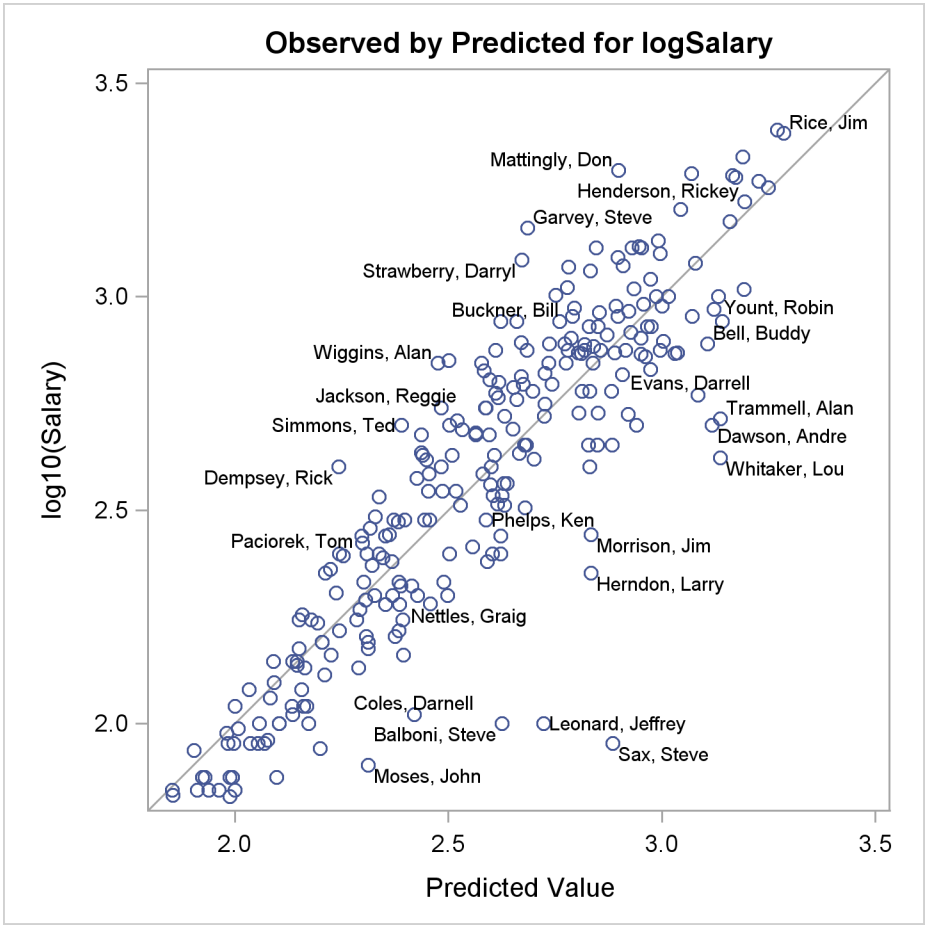
Output 73.1.10 Fit Diagnostics



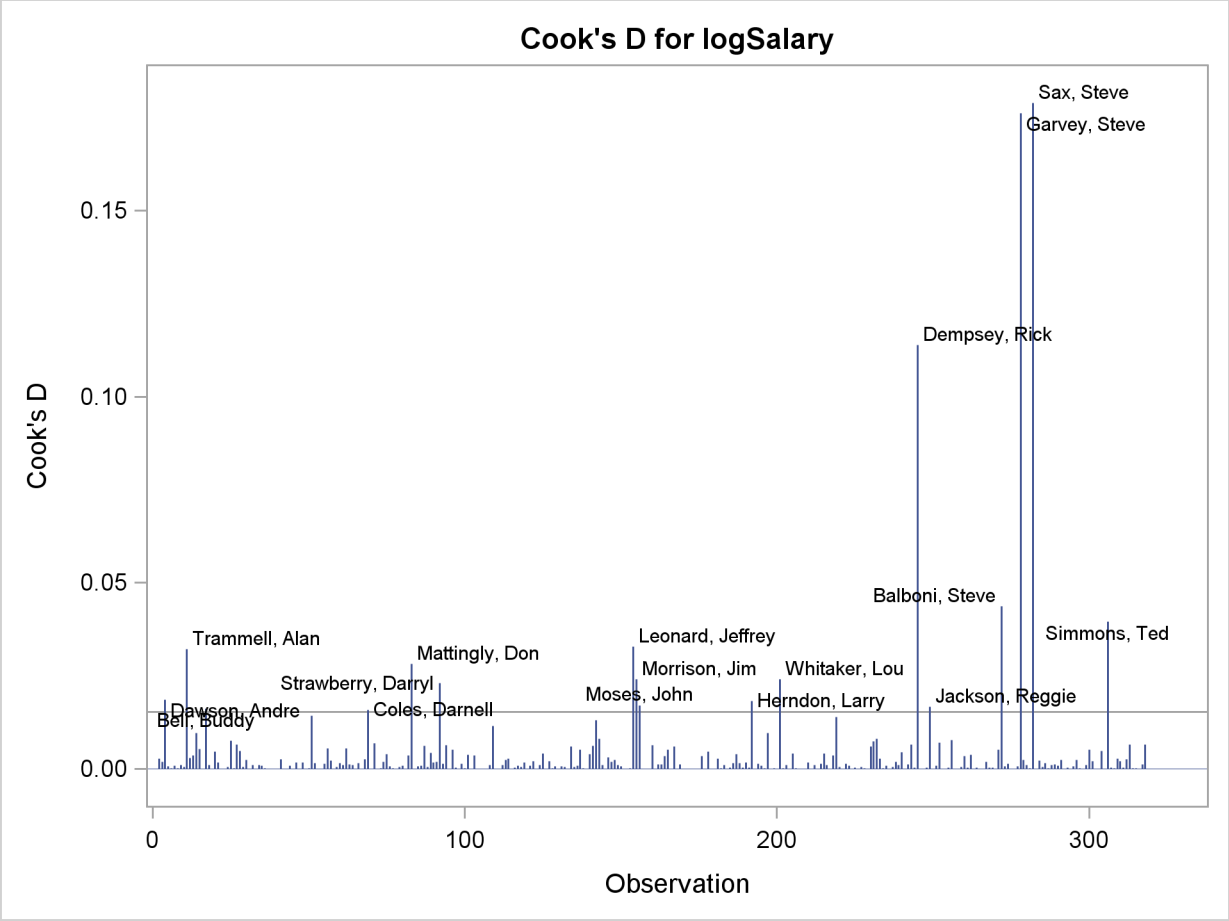
Output 73.1.11 Outlier and Leverage Diagnostics



Output 73.1.12 Observed by Predicted Values

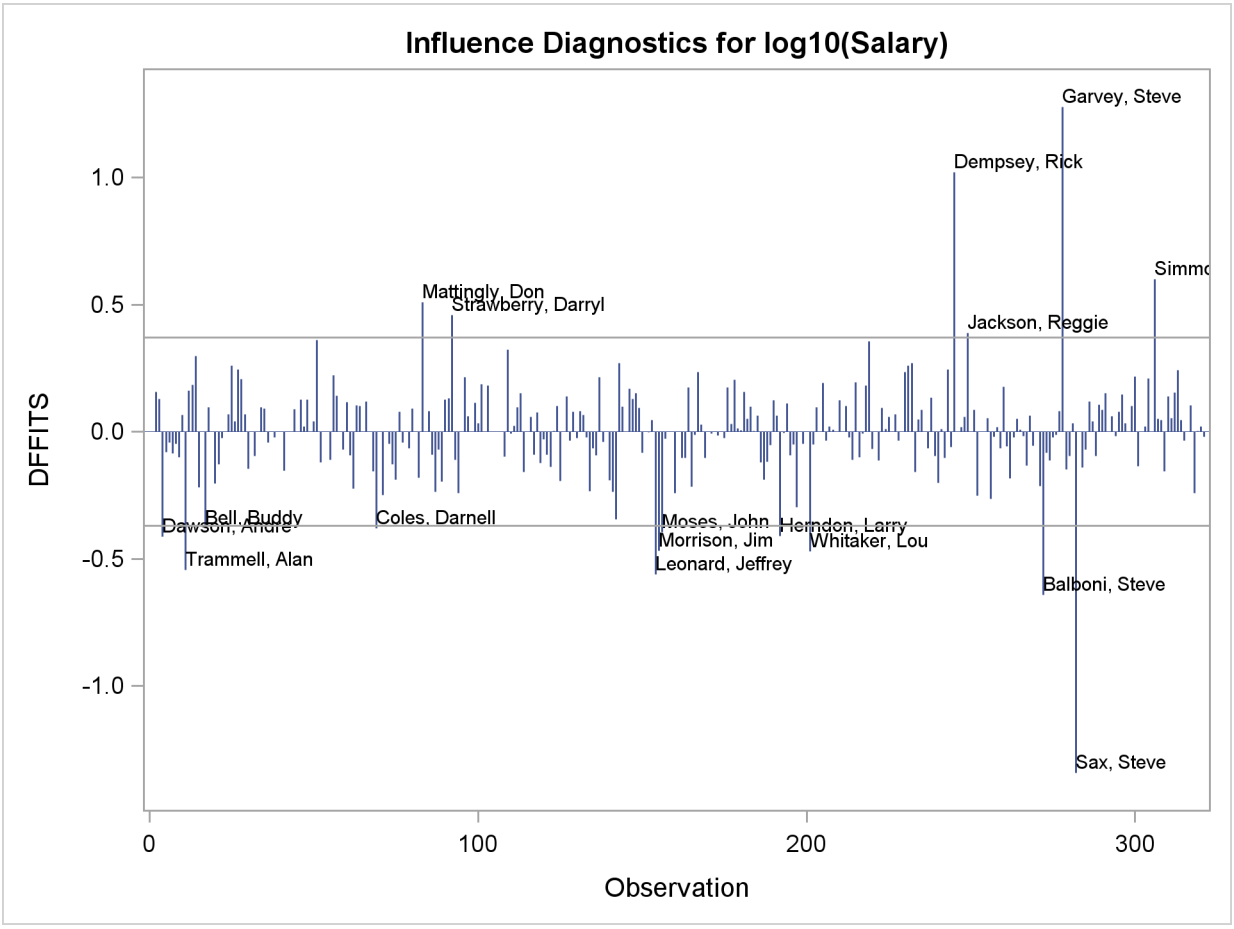


Output 73.1.13 Cook's D

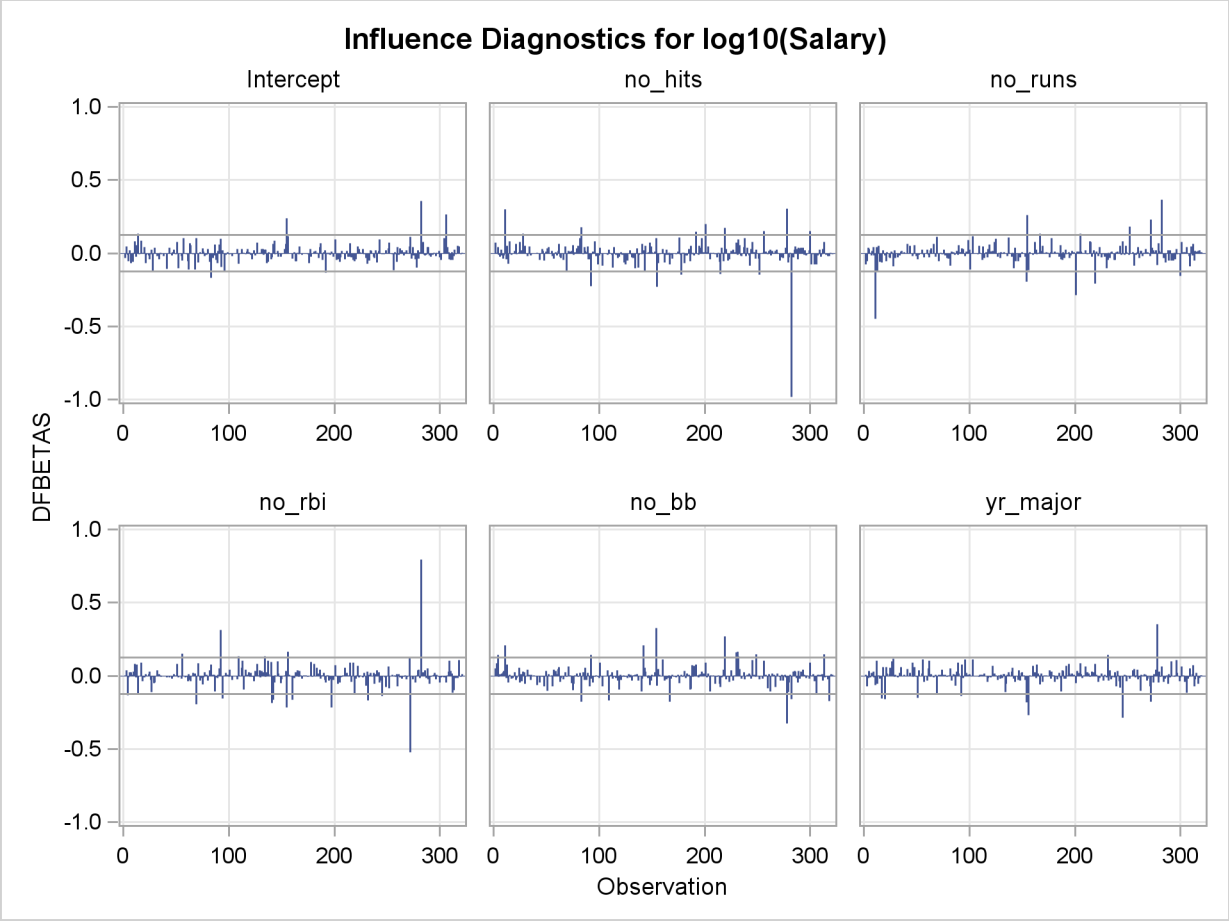


The RStudent by leverage plot in Output 73.1.11 and the Cook's *D* plot in Output 73.1.13 show that there are still a number of influential observations. By specifying the DFFITS and DFBETAS suboptions of the PLOTS= option, you obtain additional influence diagnostics plots shown in Output 73.1.14 and Output 73.1.15. See "Influence Statistics" on page 5559 for details about the interpretation DFFITS and DFBETAS statistics.

Output 73.1.14 DFFITS



Output 73.1.15 DFBETAS



You can continue this analysis by investigating how the influential observations identified in the various influence plots affect the fit. You can also use PROC ROBUSTREG to obtain a fit that is resistant to the presence of high leverage points and outliers.

Example 73.2: Aerobic Fitness Prediction

Aerobic fitness (measured by the ability to consume oxygen) is fit to some simple exercise tests. The goal is to develop an equation to predict fitness based on the exercise tests rather than on expensive and cumbersome oxygen consumption measurements. Three model-selection methods are used: forward selection, backward selection, and MAXR selection. Here are the data:

```
*------------------Data on Physical Fitness------------------*
| These measurements were made on men involved in a physical  |
| fitness course at N.C.State Univ. The variables are Age     |
| (years), Weight (kg), Oxygen intake rate (ml per kg body    |
| weight per minute), time to run 1.5 miles (minutes), heart  |
| rate while resting, heart rate while running (same time     |
| Oxygen rate measured), and maximum heart rate recorded while |
| running.                                                    |
| ***Certain values of MaxPulse were changed for this analysis.|
*-------------------------------------------------------------*;
data fitness;
   input Age Weight Oxygen RunTime RestPulse RunPulse MaxPulse @@;
datalines;
44 89.47 44.609 11.37 62 178 182   40 75.07 45.313 10.07 62 185 185
44 85.84 54.297  8.65 45 156 168   42 68.15 59.571  8.17 40 166 172
38 89.02 49.874  9.22 55 178 180   47 77.45 44.811 11.63 58 176 176
40 75.98 45.681 11.95 70 176 180   43 81.19 49.091 10.85 64 162 170
44 81.42 39.442 13.08 63 174 176   38 81.87 60.055  8.63 48 170 186
44 73.03 50.541 10.13 45 168 168   45 87.66 37.388 14.03 56 186 192
45 66.45 44.754 11.12 51 176 176   47 79.15 47.273 10.60 47 162 164
54 83.12 51.855 10.33 50 166 170   49 81.42 49.156  8.95 44 180 185
51 69.63 40.836 10.95 57 168 172   51 77.91 46.672 10.00 48 162 168
48 91.63 46.774 10.25 48 162 164   49 73.37 50.388 10.08 67 168 168
57 73.37 39.407 12.63 58 174 176   54 79.38 46.080 11.17 62 156 165
52 76.32 45.441  9.63 48 164 166   50 70.87 54.625  8.92 48 146 155
51 67.25 45.118 11.08 48 172 172   54 91.63 39.203 12.88 44 168 172
51 73.71 45.790 10.47 59 186 188   57 59.08 50.545  9.93 49 148 155
49 76.32 48.673  9.40 56 186 188   48 61.24 47.920 11.50 52 170 176
52 82.78 47.467 10.50 53 170 172
;
```

The following statements demonstrate the FORWARD, BACKWARD, and MAXR model selection methods:

```
proc reg data=fitness;
   model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse
         / selection=forward;
   model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse
         / selection=backward;
   model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse
         / selection=maxr;
run;
```

Output 73.2.1 shows the sequence of models produced by the FORWARD model-selection method.

Output 73.2.1 Forward Selection Method: PROC REG

The REG Procedure
Model: MODEL1
Dependent Variable: Oxygen

Forward Selection: Step 1

Variable RunTime Entered: R-Square = 0.7434 and C(p) = 13.6988

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	1	632.90010	632.90010	84.01	<.0001
Error	29	218.48144	7.53384		
Corrected Total	30	851.38154			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	82.42177	3.85530	3443.36654	457.05	<.0001
RunTime	-3.31056	0.36119	632.90010	84.01	<.0001

Bounds on condition number: 1, 1

Forward Selection: Step 2

Variable Age Entered: R-Square = 0.7642 and C(p) = 12.3894

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	650.66573	325.33287	45.38	<.0001
Error	28	200.71581	7.16842		
Corrected Total	30	851.38154			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	88.46229	5.37264	1943.41071	271.11	<.0001
Age	-0.15037	0.09551	17.76563	2.48	0.1267
RunTime	-3.20395	0.35877	571.67751	79.75	<.0001

Output 73.2.1 *continued*

Bounds on condition number: 1.0369, 4.1478

Forward Selection: Step 3

Variable RunPulse Entered: R-Square = 0.8111 and C(p) = 6.9596

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	690.55086	230.18362	38.64	<.0001
Error	27	160.83069	5.95669		
Corrected Total	30	851.38154			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	111.71806	10.23509	709.69014	119.14	<.0001
Age	-0.25640	0.09623	42.28867	7.10	0.0129
RunTime	-2.82538	0.35828	370.43529	62.19	<.0001
RunPulse	-0.13091	0.05059	39.88512	6.70	0.0154

Bounds on condition number: 1.3548, 11.597

Forward Selection: Step 4

Variable MaxPulse Entered: R-Square = 0.8368 and C(p) = 4.8800

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	712.45153	178.11288	33.33	<.0001
Error	26	138.93002	5.34346		
Corrected Total	30	851.38154			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	98.14789	11.78569	370.57373	69.35	<.0001
Age	-0.19773	0.09564	22.84231	4.27	0.0488
RunTime	-2.76758	0.34054	352.93570	66.05	<.0001
RunPulse	-0.34811	0.11750	46.90089	8.78	0.0064
MaxPulse	0.27051	0.13362	21.90067	4.10	0.0533

Output 73.2.1 *continued*

Bounds on condition number: 8.4182, 76.851

Forward Selection: Step 5

Variable Weight Entered: R-Square = 0.8480 and C(p) = 5.1063

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	721.97309	144.39462	27.90	<.0001
Error	25	129.40845	5.17634		
Corrected Total	30	851.38154			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	102.20428	11.97929	376.78935	72.79	<.0001
Age	−0.21962	0.09550	27.37429	5.29	0.0301
Weight	−0.07230	0.05331	9.52157	1.84	0.1871
RunTime	−2.68252	0.34099	320.35968	61.89	<.0001
RunPulse	−0.37340	0.11714	52.59624	10.16	0.0038
MaxPulse	0.30491	0.13394	26.82640	5.18	0.0316

Bounds on condition number: 8.7312, 104.83

The final variable available to add to the model, RestPulse, is not added since it does not meet the 50% (the default value of the SLE option is 0.5 for FORWARD selection) significance-level criterion for entry into the model.

The BACKWARD model-selection method begins with the full model. Output 73.2.2 shows the steps of the BACKWARD method. RestPulse is the first variable deleted, followed by Weight. No other variables are deleted from the model since the variables remaining (Age, RunTime, RunPulse, and MaxPulse) are all significant at the 10% (the default value of the SLS option is 0.1 for the BACKWARD elimination method) significance level.

Output 73.2.2 Backward Selection Method: PROC REG

Backward Elimination: Step 0

All Variables Entered: R-Square = 0.8487 and C(p) = 7.0000

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	722.54361	120.42393	22.43	<.0001
Error	24	128.83794	5.36825		
Corrected Total	30	851.38154			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	102.93448	12.40326	369.72831	68.87	<.0001
Age	-0.22697	0.09984	27.74577	5.17	0.0322
Weight	-0.07418	0.05459	9.91059	1.85	0.1869
RunTime	-2.62865	0.38456	250.82210	46.72	<.0001
RunPulse	-0.36963	0.11985	51.05806	9.51	0.0051
RestPulse	-0.02153	0.06605	0.57051	0.11	0.7473
MaxPulse	0.30322	0.13650	26.49142	4.93	0.0360

Bounds on condition number: 8.7438, 137.13

Backward Elimination: Step 1

Variable RestPulse Removed: R-Square = 0.8480 and C(p) = 5.1063

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	721.97309	144.39462	27.90	<.0001
Error	25	129.40845	5.17634		
Corrected Total	30	851.38154			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	102.20428	11.97929	376.78935	72.79	<.0001
Age	-0.21962	0.09550	27.37429	5.29	0.0301
Weight	-0.07230	0.05331	9.52157	1.84	0.1871
RunTime	-2.68252	0.34099	320.35968	61.89	<.0001
RunPulse	-0.37340	0.11714	52.59624	10.16	0.0038
MaxPulse	0.30491	0.13394	26.82640	5.18	0.0316

Output 73.2.2 *continued*

```
                Bounds on condition number: 8.7312, 104.83
------------------------------------------------------------------------------

                        Backward Elimination: Step 2

         Variable Weight Removed: R-Square = 0.8368 and C(p) = 4.8800

                            Analysis of Variance

                                    Sum of           Mean
Source                    DF       Squares         Square    F Value    Pr > F

Model                      4     712.45153      178.11288      33.33    <.0001
Error                     26     138.93002        5.34346
Corrected Total           30     851.38154

                  Parameter     Standard
      Variable     Estimate        Error   Type II SS  F Value  Pr > F

      Intercept    98.14789     11.78569    370.57373    69.35  <.0001
      Age          -0.19773      0.09564     22.84231     4.27  0.0488
      RunTime      -2.76758      0.34054    352.93570    66.05  <.0001
      RunPulse     -0.34811      0.11750     46.90089     8.78  0.0064
      MaxPulse      0.27051      0.13362     21.90067     4.10  0.0533

                Bounds on condition number: 8.4182, 76.851
```

The MAXR method tries to find the "best" one-variable model, the "best" two-variable model, and so on. Output 73.2.3 shows that the one-variable model contains RunTime; the two-variable model contains RunTime and Age; the three-variable model contains RunTime, Age, and RunPulse; the four-variable model contains Age, RunTime, RunPulse, and MaxPulse; the five-variable model contains Age, Weight, RunTime, RunPulse, and MaxPulse; and finally, the six-variable model contains all the variables in the MODEL statement.

Output 73.2.3 Maximum R-Square Improvement Selection Method: PROC REG

```
                    Maximum R-Square Improvement: Step 1

          Variable RunTime Entered: R-Square = 0.7434 and C(p) = 13.6988

                            Analysis of Variance

                                    Sum of           Mean
Source                    DF       Squares         Square    F Value    Pr > F

Model                      1     632.90010      632.90010      84.01    <.0001
Error                     29     218.48144        7.53384
Corrected Total           30     851.38154
```

Output 73.2.3 *continued*

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	82.42177	3.85530	3443.36654	457.05	<.0001
RunTime	-3.31056	0.36119	632.90010	84.01	<.0001

Bounds on condition number: 1, 1

The above model is the best 1-variable model found.

Maximum R-Square Improvement: Step 2

Variable Age Entered: R-Square = 0.7642 and C(p) = 12.3894

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	650.66573	325.33287	45.38	<.0001
Error	28	200.71581	7.16842		
Corrected Total	30	851.38154			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	88.46229	5.37264	1943.41071	271.11	<.0001
Age	-0.15037	0.09551	17.76563	2.48	0.1267
RunTime	-3.20395	0.35877	571.67751	79.75	<.0001

Bounds on condition number: 1.0369, 4.1478

The above model is the best 2-variable model found.

Maximum R-Square Improvement: Step 3

Variable RunPulse Entered: R-Square = 0.8111 and C(p) = 6.9596

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	690.55086	230.18362	38.64	<.0001
Error	27	160.83069	5.95669		
Corrected Total	30	851.38154			

Output 73.2.3 *continued*

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	111.71806	10.23509	709.69014	119.14	<.0001
Age	−0.25640	0.09623	42.28867	7.10	0.0129
RunTime	−2.82538	0.35828	370.43529	62.19	<.0001
RunPulse	−0.13091	0.05059	39.88512	6.70	0.0154

Bounds on condition number: 1.3548, 11.597

The above model is the best 3-variable model found.

Maximum R-Square Improvement: Step 4

Variable MaxPulse Entered: R-Square = 0.8368 and C(p) = 4.8800

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	4	712.45153	178.11288	33.33	<.0001
Error	26	138.93002	5.34346		
Corrected Total	30	851.38154			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	98.14789	11.78569	370.57373	69.35	<.0001
Age	−0.19773	0.09564	22.84231	4.27	0.0488
RunTime	−2.76758	0.34054	352.93570	66.05	<.0001
RunPulse	−0.34811	0.11750	46.90089	8.78	0.0064
MaxPulse	0.27051	0.13362	21.90067	4.10	0.0533

Output 73.2.3 *continued*

Bounds on condition number: 8.4182, 76.851

The above model is the best 4-variable model found.

Maximum R-Square Improvement: Step 5

Variable Weight Entered: R-Square = 0.8480 and C(p) = 5.1063

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	721.97309	144.39462	27.90	<.0001
Error	25	129.40845	5.17634		
Corrected Total	30	851.38154			

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	102.20428	11.97929	376.78935	72.79	<.0001
Age	-0.21962	0.09550	27.37429	5.29	0.0301
Weight	-0.07230	0.05331	9.52157	1.84	0.1871
RunTime	-2.68252	0.34099	320.35968	61.89	<.0001
RunPulse	-0.37340	0.11714	52.59624	10.16	0.0038
MaxPulse	0.30491	0.13394	26.82640	5.18	0.0316

Bounds on condition number: 8.7312, 104.83

The above model is the best 5-variable model found.

Maximum R-Square Improvement: Step 6

Variable RestPulse Entered: R-Square = 0.8487 and C(p) = 7.0000

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	722.54361	120.42393	22.43	<.0001
Error	24	128.83794	5.36825		
Corrected Total	30	851.38154			

Output 73.2.3 *continued*

Variable	Parameter Estimate	Standard Error	Type II SS	F Value	Pr > F
Intercept	102.93448	12.40326	369.72831	68.87	<.0001
Age	-0.22697	0.09984	27.74577	5.17	0.0322
Weight	-0.07418	0.05459	9.91059	1.85	0.1869
RunTime	-2.62865	0.38456	250.82210	46.72	<.0001
RunPulse	-0.36963	0.11985	51.05806	9.51	0.0051
RestPulse	-0.02153	0.06605	0.57051	0.11	0.7473
MaxPulse	0.30322	0.13650	26.49142	4.93	0.0360

Bounds on condition number: 8.7438, 137.13

Note that for all three of these methods, RestPulse contributes least to the model. In the case of forward selection, it is not added to the model. In the case of backward selection, it is the first variable to be removed from the model. In the case of MAXR selection, RestPulse is included only for the full model.

For the STEPWISE, BACKWARD, and FORWARD selection methods, you can control the amount of detail displayed by using the DETAILS option, and you can use ODS Graphics to produce plots that show how selection criteria progress as the selection proceeds. For example, the following statements display only the selection summary table for the FORWARD selection method (Output 73.2.4) and produce the plots shown in Output 73.2.5 and Output 73.2.6.

```
ods graphics on;

proc reg data=fitness plots=(criteria sbc);
   model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse
         / selection=forward details=summary;
run;
```

Output 73.2.4 Forward Selection Summary

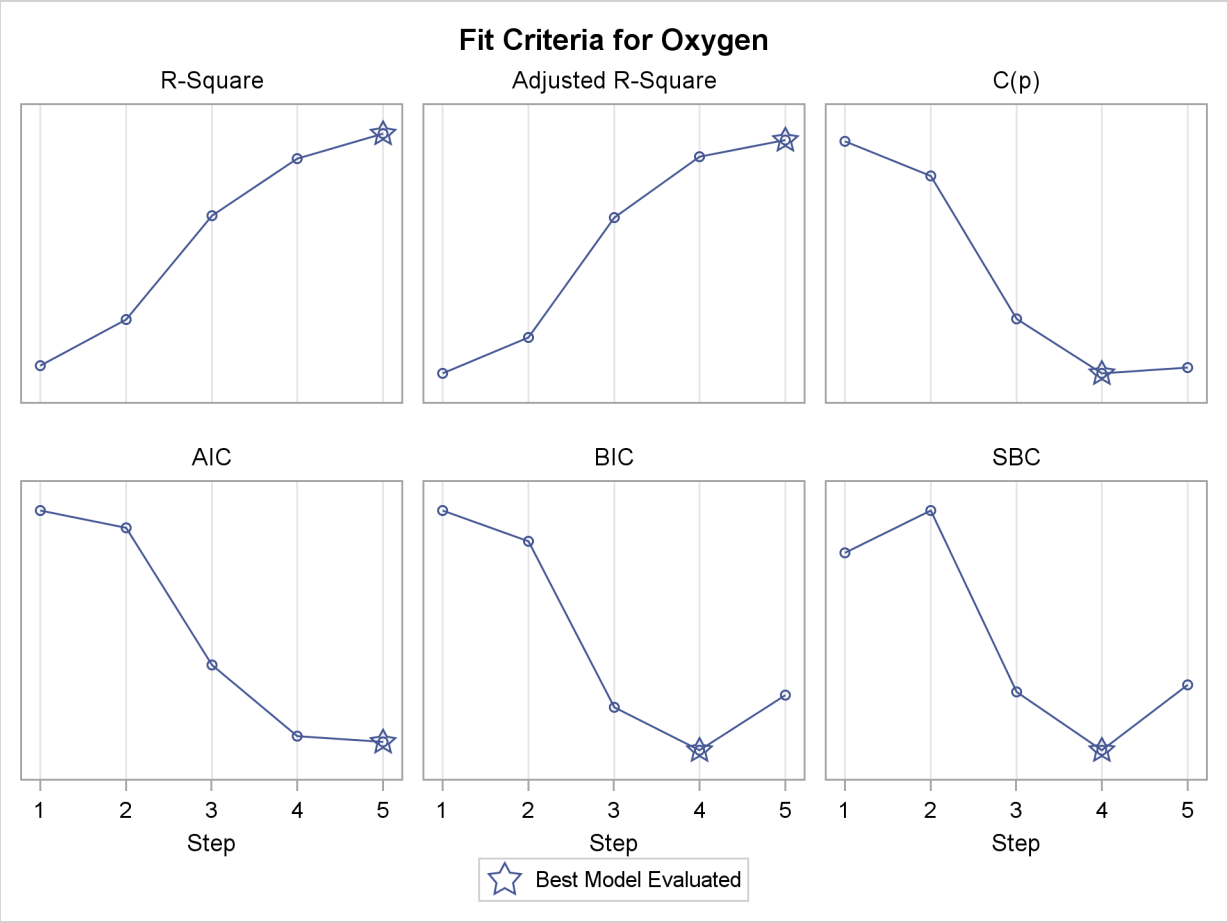
The REG Procedure
Model: MODEL1
Dependent Variable: Oxygen

Summary of Forward Selection

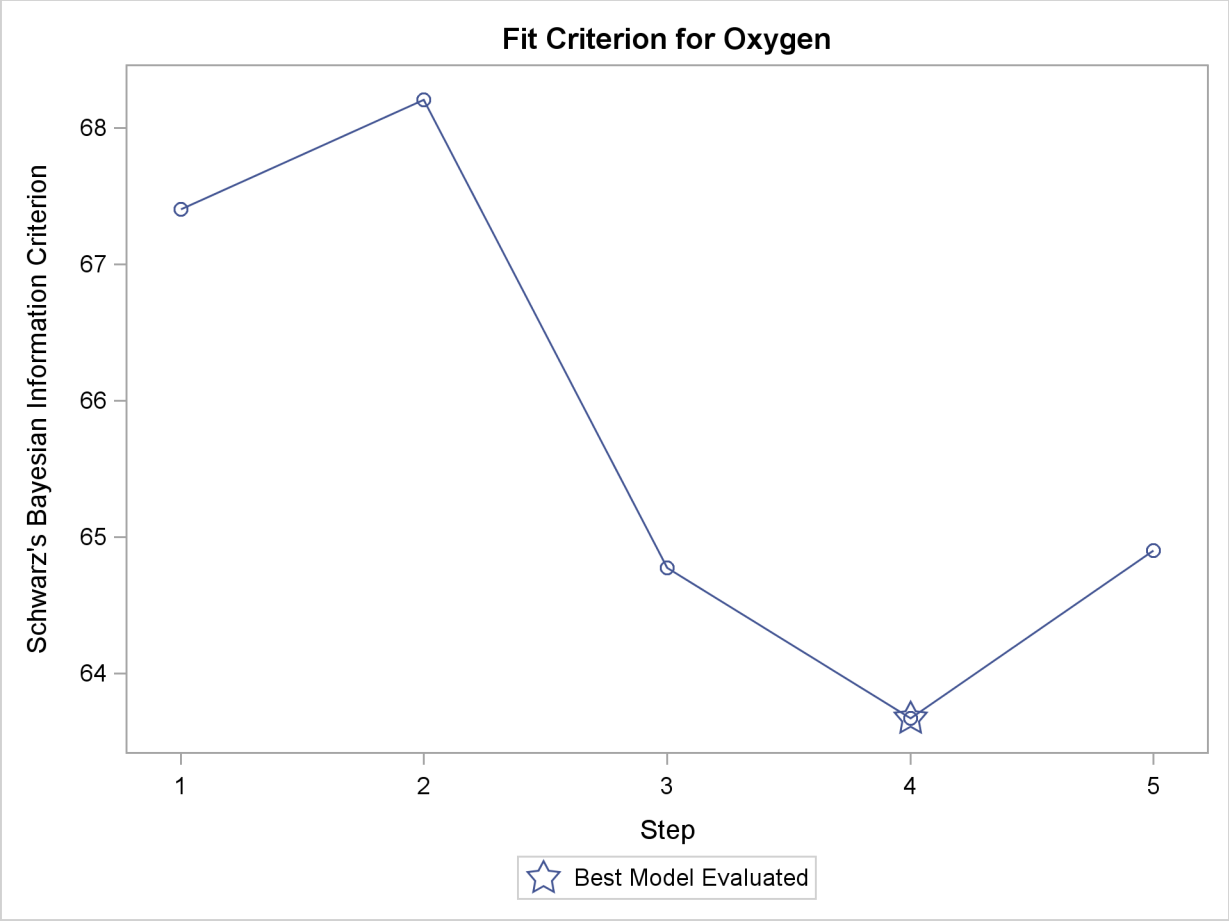
Step	Variable Entered	Number Vars In	Partial R-Square	Model R-Square	C(p)	F Value	Pr > F
1	RunTime	1	0.7434	0.7434	13.6988	84.01	<.0001
2	Age	2	0.0209	0.7642	12.3894	2.48	0.1267
3	RunPulse	3	0.0468	0.8111	6.9596	6.70	0.0154
4	MaxPulse	4	0.0257	0.8368	4.8800	4.10	0.0533
5	Weight	5	0.0112	0.8480	5.1063	1.84	0.1871

Output 73.2.5 show how six fit criteria progress as the forward selection proceeds. The step at which each criterion achieves its best value is indicated. For example, the BIC criterion achieves its minimum value for the model at step 4. Note that this does not mean that the model at step 4 achieves the smallest BIC criterion among all possible models that use a subset of the regressors; the model at step 4 yields the smallest BIC statistic among the models at each step of the forward selection. Output 73.2.6 show the progression of the SBC statistic in its own plot. If you want to see six of the selection criteria in individual plots, you can specify the UNPACK suboption of the PLOTS=CRITERIA option in the PROC REG statement.

Output 73.2.5 Fit Criteria



Output 73.2.6 SBC Criterion



Next, the RSQUARE model-selection method is used to request R^2 and C_p statistics for all possible combinations of the six independent variables. The following statements produce Output 73.2.7:

```
proc reg data=fitness plots=(criteria(label) cp);
   model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse
         / selection=rsquare cp;
   title 'Physical fitness data: all models';
run;
```

Output 73.2.7 All Models by the RSQUARE Method: PROC REG

Physical fitness data: all models

The REG Procedure
Model: MODEL1
Dependent Variable: Oxygen

R-Square Selection Method

Model Index	Number in Model	R-Square	C(p)	Variables in Model
1	1	0.7434	13.6988	RunTime
2	1	0.1595	106.3021	RestPulse
3	1	0.1584	106.4769	RunPulse
4	1	0.0928	116.8818	Age
5	1	0.0560	122.7072	MaxPulse
6	1	0.0265	127.3948	Weight
7	2	0.7642	12.3894	Age RunTime
8	2	0.7614	12.8372	RunTime RunPulse
9	2	0.7452	15.4069	RunTime MaxPulse
10	2	0.7449	15.4523	Weight RunTime
11	2	0.7435	15.6746	RunTime RestPulse
12	2	0.3760	73.9645	Age RunPulse
13	2	0.3003	85.9742	Age RestPulse
14	2	0.2894	87.6951	RunPulse MaxPulse
15	2	0.2600	92.3638	Age MaxPulse
16	2	0.2350	96.3209	RunPulse RestPulse
17	2	0.1806	104.9523	Weight RestPulse
18	2	0.1740	105.9939	RestPulse MaxPulse
19	2	0.1669	107.1332	Weight RunPulse
20	2	0.1506	109.7057	Age Weight
21	2	0.0675	122.8881	Weight MaxPulse
22	3	0.8111	6.9596	Age RunTime RunPulse
23	3	0.8100	7.1350	RunTime RunPulse MaxPulse
24	3	0.7817	11.6167	Age RunTime MaxPulse
25	3	0.7708	13.3453	Age Weight RunTime
26	3	0.7673	13.8974	Age RunTime RestPulse
27	3	0.7619	14.7619	RunTime RunPulse RestPulse
28	3	0.7618	14.7729	Weight RunTime RunPulse
29	3	0.7462	17.2588	Weight RunTime MaxPulse
30	3	0.7452	17.4060	RunTime RestPulse MaxPulse
31	3	0.7451	17.4243	Weight RunTime RestPulse
32	3	0.4666	61.5873	Age RunPulse RestPulse
33	3	0.4223	68.6250	Age RunPulse MaxPulse
34	3	0.4091	70.7102	Age Weight RunPulse
35	3	0.3900	73.7424	Age RestPulse MaxPulse
36	3	0.3568	79.0013	Age Weight RestPulse
37	3	0.3538	79.4891	RunPulse RestPulse MaxPulse

Output 73.2.7 *continued*

Physical fitness data: all models

The REG Procedure
Model: MODEL1
Dependent Variable: Oxygen

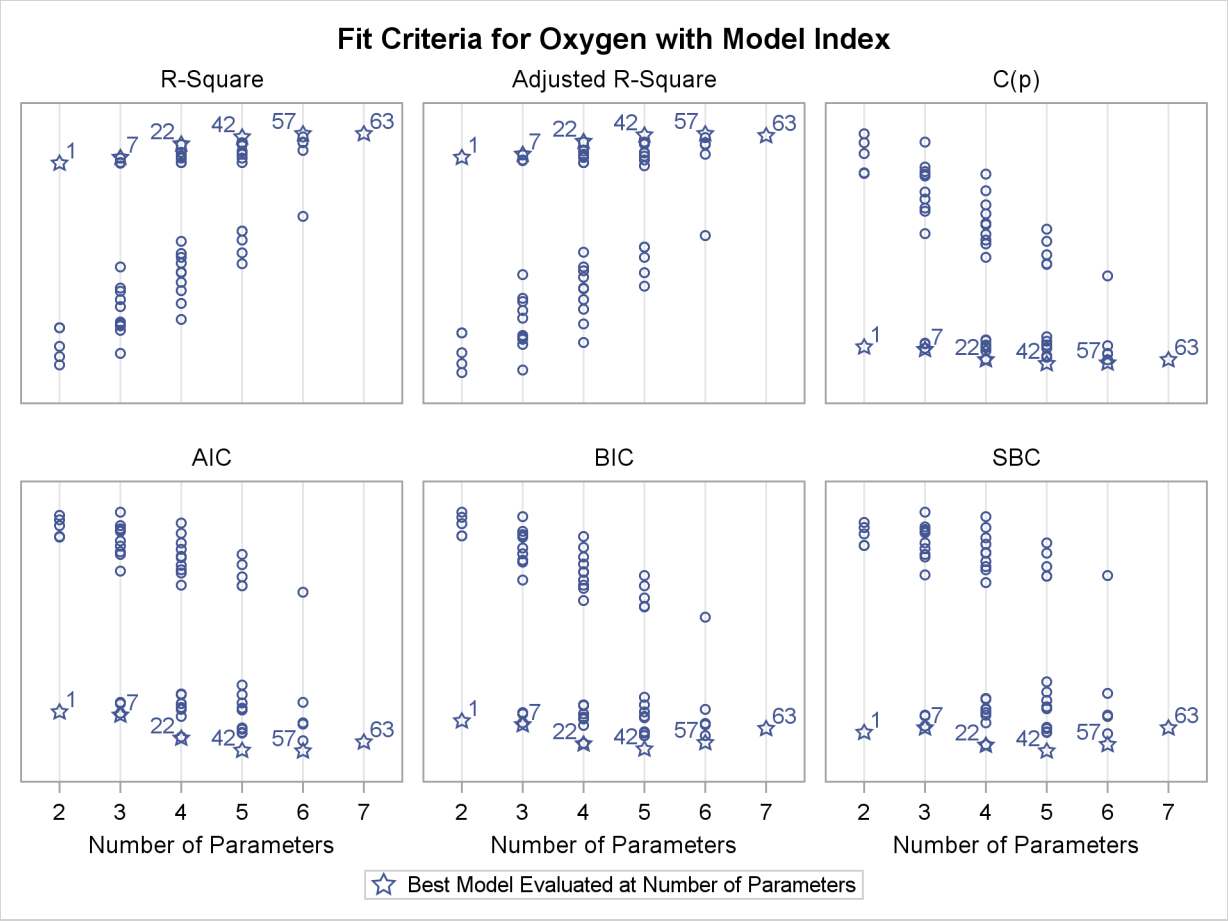
R-Square Selection Method

Model Index	Number in Model	R-Square	C(p)	Variables in Model
38	3	0.3208	84.7216	Weight RunPulse MaxPulse
39	3	0.2902	89.5693	Age Weight MaxPulse
40	3	0.2447	96.7952	Weight RunPulse RestPulse
41	3	0.1882	105.7430	Weight RestPulse MaxPulse
42	4	0.8368	4.8800	Age RunTime RunPulse MaxPulse
43	4	0.8165	8.1035	Age Weight RunTime RunPulse
44	4	0.8158	8.2056	Weight RunTime RunPulse MaxPulse
45	4	0.8117	8.8683	Age RunTime RunPulse RestPulse
46	4	0.8104	9.0697	RunTime RunPulse RestPulse MaxPulse
47	4	0.7862	12.9039	Age Weight RunTime MaxPulse
48	4	0.7834	13.3468	Age RunTime RestPulse MaxPulse
49	4	0.7750	14.6788	Age Weight RunTime RestPulse
50	4	0.7623	16.7058	Weight RunTime RunPulse RestPulse
51	4	0.7462	19.2550	Weight RunTime RestPulse MaxPulse
52	4	0.5034	57.7590	Age Weight RunPulse RestPulse
53	4	0.5025	57.9092	Age RunPulse RestPulse MaxPulse
54	4	0.4717	62.7830	Age Weight RunPulse MaxPulse
55	4	0.4256	70.0963	Age Weight RestPulse MaxPulse
56	4	0.3858	76.4100	Weight RunPulse RestPulse MaxPulse
57	5	0.8480	5.1063	Age Weight RunTime RunPulse MaxPulse
58	5	0.8370	6.8461	Age RunTime RunPulse RestPulse MaxPulse
59	5	0.8176	9.9348	Age Weight RunTime RunPulse RestPulse
60	5	0.8161	10.1685	Weight RunTime RunPulse RestPulse MaxPulse
61	5	0.7887	14.5111	Age Weight RunTime RestPulse MaxPulse
62	5	0.5541	51.7233	Age Weight RunPulse RestPulse MaxPulse
63	6	0.8487	7.0000	Age Weight RunTime RunPulse RestPulse MaxPulse

The models in Output 73.2.7 are arranged first by the number of variables in the model and then by the magnitude of R^2 for the model.

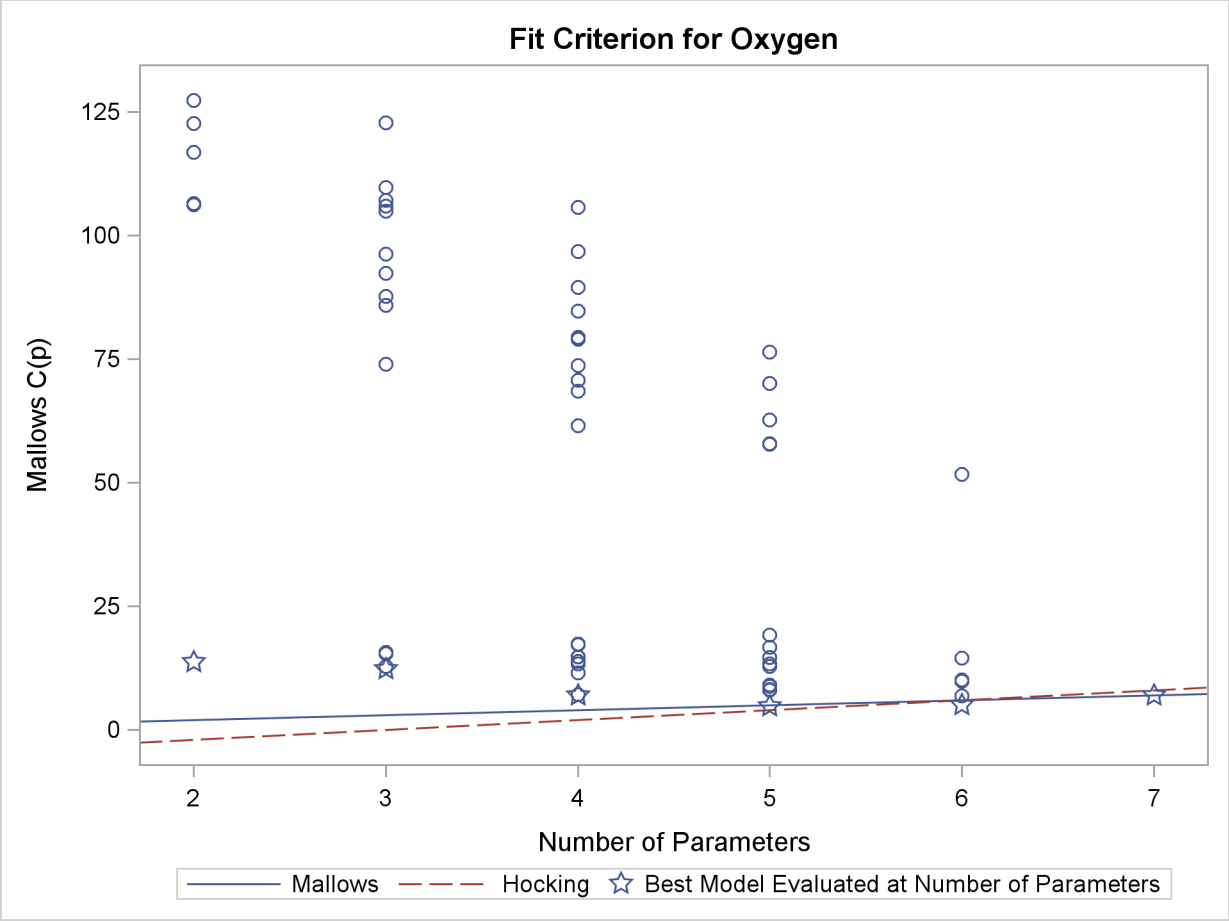
Output 73.2.8 shows the panel of fit criteria for the RSQUARE selection method. The best models (based on the R-square statistic) for each subset size are indicated on the plots. The LABEL suboption specifies that these models are labeled by the model number that appears in the summary table shown in Output 73.2.7.

Output 73.2.8 Fit Criteria



Output 73.2.9 shows the plot of the C_p criterion by number of regressors in the model. Useful reference lines suggested by Mallows (1973) and Hocking (1976) are included on the plot. However, because all possible subset models are included on this plot, the better models are all compressed near the bottom of the plot.

Output 73.2.9 C_p Criterion



The following statements use the BEST=20 option in the model statement and SELECTION=CP to restrict attention to the models that yield the 20 smallest values of the C_p statistic:

```
proc reg data=fitness plots(only)=cp(label);
   model Oxygen=Age Weight RunTime RunPulse RestPulse MaxPulse
         / selection=cp best=20;
run;

ods graphics off;
```

Output 73.2.10 shows the summary table listing the regressors in the 20 models that yield the smallest C_p values, and Output 73.2.11 presents the results graphically. Reference lines $C_p = 2p - p_{full}$ and $C_p = p$ are shown on this plot. See the PLOTS=CP option on page 5459 for interpretations of these lines. For the Fitness data, these lines indicate that a six-variable model is a reasonable choice for doing parameter estimation, while a five-variable model might be suitable for doing prediction.

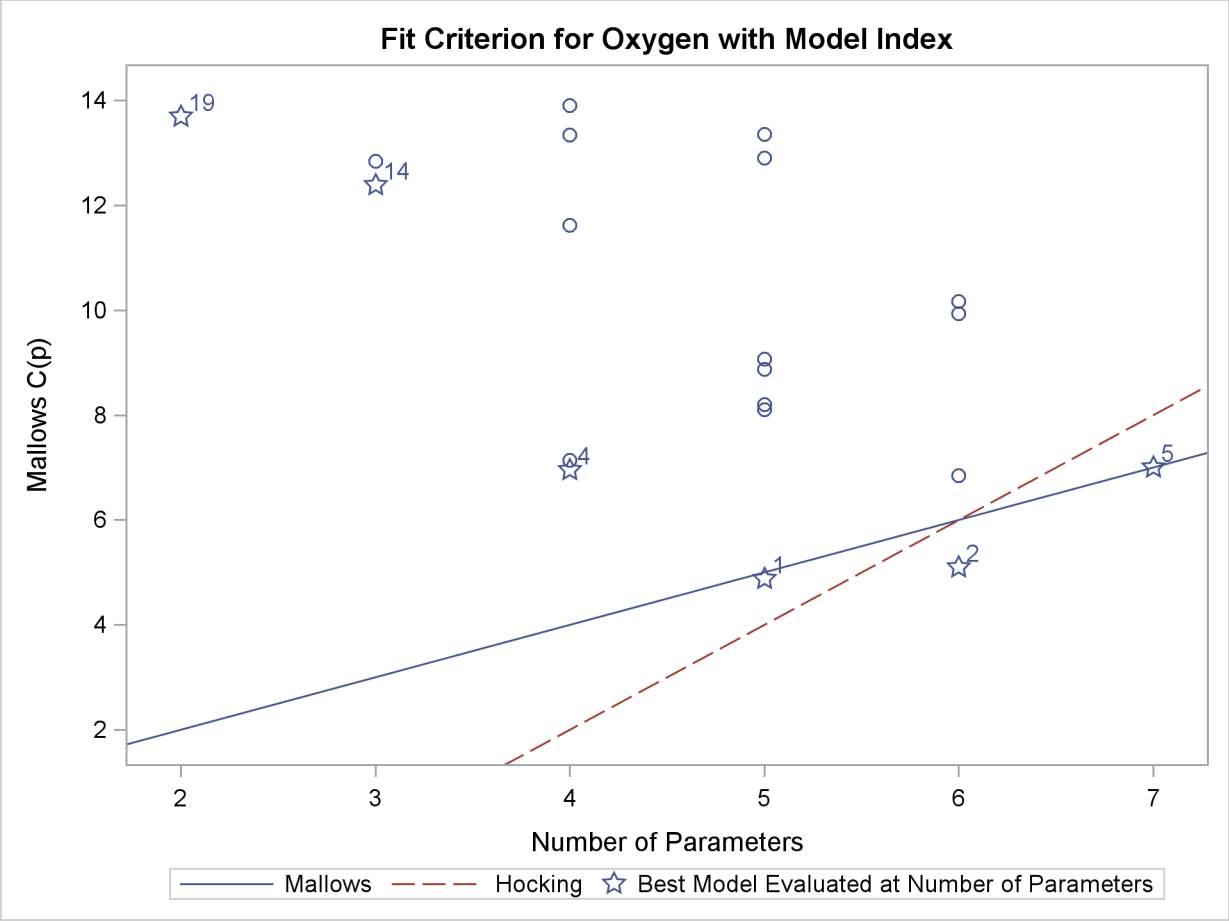
Output 73.2.10 C_p Selection Summary: PROC REG

The REG Procedure
Model: MODEL1
Dependent Variable: Oxygen

C(p) Selection Method

Model Index	Number in Model	C(p)	R-Square	Variables in Model
1	4	4.8800	0.8368	Age RunTime RunPulse MaxPulse
2	5	5.1063	0.8480	Age Weight RunTime RunPulse MaxPulse
3	5	6.8461	0.8370	Age RunTime RunPulse RestPulse MaxPulse
4	3	6.9596	0.8111	Age RunTime RunPulse
5	6	7.0000	0.8487	Age Weight RunTime RunPulse RestPulse MaxPulse
6	3	7.1350	0.8100	RunTime RunPulse MaxPulse
7	4	8.1035	0.8165	Age Weight RunTime RunPulse
8	4	8.2056	0.8158	Weight RunTime RunPulse MaxPulse
9	4	8.8683	0.8117	Age RunTime RunPulse RestPulse
10	4	9.0697	0.8104	RunTime RunPulse RestPulse MaxPulse
11	5	9.9348	0.8176	Age Weight RunTime RunPulse RestPulse
12	5	10.1685	0.8161	Weight RunTime RunPulse RestPulse MaxPulse
13	3	11.6167	0.7817	Age RunTime MaxPulse
14	2	12.3894	0.7642	Age RunTime
15	2	12.8372	0.7614	RunTime RunPulse
16	4	12.9039	0.7862	Age Weight RunTime MaxPulse
17	3	13.3453	0.7708	Age Weight RunTime
18	4	13.3468	0.7834	Age RunTime RestPulse MaxPulse
19	1	13.6988	0.7434	RunTime
20	3	13.8974	0.7673	Age RunTime RestPulse

Output 73.2.11 C_p Criterion



Before making a final decision about which model to use, you would want to perform collinearity diagnostics. Note that, since many different models have been fit and the choice of a final model is based on R^2, the statistics are biased and the p-values for the parameter estimates are not valid.

Example 73.3: Predicting Weight by Height and Age

In this example, the weights of schoolchildren are modeled as a function of their heights and ages. The example shows the use of a BY statement with PROC REG, multiple MODEL statements, and the OUTEST= and OUTSSCP= options, which create data sets. Here are the data:

```
*------------Data on Age, Weight, and Height of Children-------*
| Age (months), height (inches), and weight (pounds) were      |
| recorded for a group of school children.                     |
| From Lewis and Taylor (1967).                                |
*--------------------------------------------------------------*;

data htwt;
   input sex $ age :3.1 height weight @@;
   datalines;
f 143 56.3  85.0 f 155 62.3 105.0 f 153 63.3 108.0 f 161 59.0  92.0
f 191 62.5 112.5 f 171 62.5 112.0 f 185 59.0 104.0 f 142 56.5  69.0
f 160 62.0  94.5 f 140 53.8  68.5 f 139 61.5 104.0 f 178 61.5 103.5
f 157 64.5 123.5 f 149 58.3  93.0 f 143 51.3  50.5 f 145 58.8  89.0

... more lines ...

m 164 66.5 112.0 m 189 65.0 114.0 m 164 61.5 140.0 m 167 62.0 107.5
m 151 59.3  87.0
;
```

Modeling is performed separately for boys and girls. Since the BY statement is used, interactive processing is not possible in this example; no statements can appear after the first RUN statement.

The following statements produce Output 73.3.1 through Output 73.3.4:

```
proc reg outest=est1 outsscp=sscp1 rsquare;
   by sex;
   eq1: model  weight=height;
   eq2: model  weight=height age;

proc print data=sscp1;
   title2 'SSCP type data set';

proc print data=est1;
   title2 'EST type data set';
run;
```

Output 73.3.1 Height and Weight Data: Submodel for Female Children

```
------------------------------------ sex=f -------------------------------------

                            The REG Procedure
                                Model: eq1
                        Dependent Variable: weight

                            Analysis of Variance

                                    Sum of           Mean
Source                    DF       Squares         Square    F Value    Pr > F

Model                      1         21507          21507     141.09    <.0001
Error                    109         16615      152.42739
Corrected Total          110         38121


              Root MSE            12.34615    R-Square     0.5642
              Dependent Mean      98.87838    Adj R-Sq     0.5602
              Coeff Var           12.48620


                            Parameter Estimates

                         Parameter       Standard
     Variable     DF       Estimate          Error    t Value    Pr > |t|

     Intercept     1     -153.12891       21.24814      -7.21      <.0001
     height        1        4.16361        0.35052      11.88      <.0001
```

Output 73.3.2 Height and Weight Data: Full Model for Female Children

```
------------------------------------ sex=f -------------------------------------

                            The REG Procedure
                                Model: eq2
                        Dependent Variable: weight

                            Analysis of Variance

                                    Sum of           Mean
Source                    DF       Squares         Square    F Value    Pr > F

Model                      2         22432          11216      77.21    <.0001
Error                    108         15689      145.26700
Corrected Total          110         38121


              Root MSE            12.05268    R-Square     0.5884
              Dependent Mean      98.87838    Adj R-Sq     0.5808
              Coeff Var           12.18939


                            Parameter Estimates

                         Parameter       Standard
     Variable     DF       Estimate          Error    t Value    Pr > |t|

     Intercept     1     -150.59698       20.76730      -7.25      <.0001
     height        1        3.60378        0.40777       8.84      <.0001
     age           1        1.90703        0.75543       2.52      0.0130
```

Output 73.3.3 Height and Weight Data: Submodel for Male Children

```
---------------------------------- sex=m -----------------------------------

                          The REG Procedure
                              Model: eq1
                       Dependent Variable: weight

                          Analysis of Variance

                                    Sum of           Mean
Source                      DF     Squares         Square    F Value    Pr > F

Model                        1       31126          31126     206.24    <.0001
Error                      124       18714      150.92222
Corrected Total            125       49840


              Root MSE            12.28504    R-Square     0.6245
              Dependent Mean     103.44841    Adj R-Sq     0.6215
              Coeff Var           11.87552


                          Parameter Estimates

                       Parameter       Standard
     Variable     DF       Estimate          Error    t Value    Pr > |t|

     Intercept     1     -125.69807       15.99362      -7.86      <.0001
     height        1        3.68977        0.25693      14.36      <.0001
```

Output 73.3.4 Height and Weight Data: Full Model for Male Children

```
---------------------------------- sex=m -----------------------------------

                          The REG Procedure
                              Model: eq2
                       Dependent Variable: weight

                          Analysis of Variance

                                    Sum of           Mean
Source                      DF     Squares         Square    F Value    Pr > F

Model                        2       32975          16487     120.24    <.0001
Error                      123       16866      137.11922
Corrected Total            125       49840


              Root MSE            11.70979    R-Square     0.6616
              Dependent Mean     103.44841    Adj R-Sq     0.6561
              Coeff Var           11.31945


                          Parameter Estimates

                       Parameter       Standard
     Variable     DF     Estimate          Error    t Value    Pr > |t|

     Intercept     1    -113.71346       15.59021      -7.29      <.0001
     height        1       2.68075        0.36809       7.28      <.0001
     age           1       3.08167        0.83927       3.67      0.0004
```

For both female and male children, the overall F statistics for both models are significant, indicating that the model explains a significant portion of the variation in the data. For females, the full model is

$$\text{weight} = -150.57 + 3.60 \times \text{height} + 1.91 \times \text{age}$$

and for males, the full model is

$$\text{weight} = -113.71 + 2.68 \times \text{height} + 3.08 \times \text{age}$$

The OUTSSCP= data set is shown in Output 73.3.5. Note how the BY groups are separated. Observations with _TYPE_='N' contain the number of observations in the associated BY group. Observations with _TYPE_='SSCP' contain the rows of the uncorrected sums of squares and crossproducts matrix. The observations with _NAME_='Intercept' contain crossproducts for the intercept.

Output 73.3.5 SSCP Matrix

Obs	sex	_TYPE_	_NAME_	Intercept	height	weight	age
1	f	SSCP	Intercept	111.0	6718.40	10975.50	1824.90
2	f	SSCP	height	6718.4	407879.32	669469.85	110818.32
3	f	SSCP	weight	10975.5	669469.85	1123360.75	182444.95
4	f	SSCP	age	1824.9	110818.32	182444.95	30363.81
5	f	N		111.0	111.00	111.00	111.00
6	m	SSCP	Intercept	126.0	7825.00	13034.50	2072.10
7	m	SSCP	height	7825.0	488243.60	817919.60	129432.57
8	m	SSCP	weight	13034.5	817919.60	1398238.75	217717.45
9	m	SSCP	age	2072.1	129432.57	217717.45	34515.95
10	m	N		126.0	126.00	126.00	126.00

The OUTEST= data set is displayed in Output 73.3.6; again, the BY groups are separated. The _MODEL_ column contains the labels for models from the MODEL statements. If no labels are specified, the defaults MODEL1 and MODEL2 would appear as values for _MODEL_. Note that _TYPE_='PARMS' for all observations, indicating that all observations contain parameter estimates. The _DEPVAR_ column displays the dependent variable, and the _RMSE_ column gives the root mean square error for the associated model. The Intercept column gives the estimate for the intercept for the associated model, and variables with the same name as variables in the original data set (height, age) give parameter estimates for those variables. The dependent variable, weight, is shown with a value of −1. The _IN_ column contains the number of regressors in the model not including the intercept; _P_ contains the number of parameters in the model; _EDF_ contains the error degrees of freedom; and _RSQ_ contains the R^2 statistic. Finally, note that the _IN_, _P_, _EDF_, and _RSQ_ columns appear in the OUTEST= data set since the RSQUARE option is specified in the PROC REG statement.

Output 73.3.6 OUTEST Data Set

Obs	sex	_MODEL_	_TYPE_	_DEPVAR_	_RMSE_	Intercept	height	weight	age	_IN_	_P_	_EDF_	_RSQ_
1	f	eq1	PARMS	weight	12.3461	-153.129	4.16361	-1	.	1	2	109	0.56416
2	f	eq2	PARMS	weight	12.0527	-150.597	3.60378	-1	1.90703	2	3	108	0.58845
3	m	eq1	PARMS	weight	12.2850	-125.698	3.68977	-1	.	1	2	124	0.62451
4	m	eq2	PARMS	weight	11.7098	-113.713	2.68075	-1	3.08167	2	3	123	0.66161

Example 73.4: Regression with Quantitative and Qualitative Variables

At times it is desirable to have independent variables in the model that are qualitative rather than quantitative. This is easily handled in a regression framework. Regression uses qualitative variables to distinguish between populations. There are two main advantages of fitting both populations in one model. You gain the ability to test for different slopes or intercepts in the populations, and more degrees of freedom are available for the analysis.

Regression with qualitative variables is different from analysis of variance and analysis of covariance. Analysis of variance uses qualitative independent variables only. Analysis of covariance uses quantitative variables in addition to the qualitative variables in order to account for correlation in the data and reduce MSE; however, the quantitative variables are not of primary interest and merely improve the precision of the analysis.

Consider the case where Y_i is the dependent variable, $\mathbf{X1}_i$ is a quantitative variable, $\mathbf{X2}_i$ is a qualitative variable taking on values 0 or 1, and $\mathbf{X1}_i\mathbf{X2}_i$ is the interaction. The variable $\mathbf{X2}_i$ is called a dummy, binary, or indicator variable. With values 0 or 1, it distinguishes between two populations. The model is of the form

$$\mathbf{Y}_i = \beta_0 + \beta_1\mathbf{X1}_i + \beta_2\mathbf{X2}_i + \beta_3\mathbf{X1}_i\mathbf{X2}_i + \boldsymbol{\epsilon}_i$$

for the observations $i = 1, 2, \ldots, n$. The parameters to be estimated are β_0, β_1, β_2, and β_3. The number of dummy variables used is one less than the number of qualitative levels. This yields a nonsingular $\mathbf{X}'\mathbf{X}$ matrix. See Chapter 10 of Neter, Wasserman, and Kutner (1990) for more details.

An example from Neter, Wasserman, and Kutner (1990) follows. An economist is investigating the relationship between the size of an insurance firm and the speed at which it implements new insurance innovations. He believes that the type of firm might affect this relationship and suspects that there might be some interaction between the size and type of firm. The dummy variable in the model enables the two firms to have different intercepts. The interaction term enables the firms to have different slopes as well.

In this study, $\mathbf{Y}_i$ is the number of months from the time the first firm implemented the innovation to the time it was implemented by the ith firm. The variable $\mathbf{X1}_i$ is the size of the firm, measured in total assets of the firm. The variable $\mathbf{X2}_i$ denotes the firm type; it is 0 if the firm is a mutual fund company and 1 if the firm is a stock company. The dummy variable enables each firm type to have a different intercept and slope.

The previous model can be broken down into a model for each firm type by plugging in the values for $\mathbf{X2}_i$. If $\mathbf{X2}_i = 0$, the model is

$$\mathbf{Y}_i = \beta_0 + \beta_1\mathbf{X1}_i + \boldsymbol{\epsilon}_i$$

This is the model for a mutual company. If $\mathbf{X2}_i = 1$, the model for a stock firm is

$$\mathbf{Y}_i = (\beta_0 + \beta_2) + (\beta_1 + \beta_3)\mathbf{X1}_i + \boldsymbol{\epsilon}_i$$

This model has intercept $\beta_0 + \beta_2$ and slope $\beta_1 + \beta_3$.

The data[1] follow. Note that the interaction term is created in the DATA step since polynomial effects such as size*type are not allowed in the MODEL statement in the REG procedure.

```
title 'Regression With Quantitative and Qualitative Variables';
data insurance;
   input time size type @@;
   sizetype=size*type;
   datalines;
17 151 0   26  92 0   21 175 0   30  31 0   22 104 0
 0 277 0   12 210 0   19 120 0    4 290 0   16 238 0
28 164 1   15 272 1   11 295 1   38  68 1   31  85 1
21 224 1   20 166 1   13 305 1   30 124 1   14 246 1
;
run;
```

The following statements begin the analysis:

```
proc reg data=insurance;
   model time = size type sizetype;
run;
```

The ANOVA table is displayed in Output 73.4.1.

Output 73.4.1 ANOVA Table and Parameter Estimates

Regression With Quantitative and Qualitative Variables

The REG Procedure
Model: MODEL1
Dependent Variable: time

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	1504.41904	501.47301	45.49	<.0001
Error	16	176.38096	11.02381		
Corrected Total	19	1680.80000			

Root MSE	3.32021	R-Square	0.8951
Dependent Mean	19.40000	Adj R-Sq	0.8754
Coeff Var	17.11450		

[1] From Neter, J., et al., *Applied Linear Statistical Models*, Third Edition, Copyright (c) 1990, Richard D. Irwin. Reprinted with permission of The McGraw-Hill Companies.

Output 73.4.1 *continued*

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	33.83837	2.44065	13.86	<.0001
size	1	−0.10153	0.01305	−7.78	<.0001
type	1	8.13125	3.65405	2.23	0.0408
sizetype	1	−0.00041714	0.01833	−0.02	0.9821

The overall F statistic is significant (F=45.490, p<0.0001). The interaction term is not significant (t=−0.023, p=0.9821). Hence, this term should be removed and the model refitted, as shown in the following statements:

```
   delete sizetype;
   print;
run;
```

The DELETE statement removes the interaction term (sizetype) from the model. The new ANOVA and parameter estimates tables are shown in Output 73.4.2.

Output 73.4.2 ANOVA Table and Parameter Estimates

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	1504.41333	752.20667	72.50	<.0001
Error	17	176.38667	10.37569		
Corrected Total	19	1680.80000			

Root MSE	3.22113	R-Square	0.8951
Dependent Mean	19.40000	Adj R-Sq	0.8827
Coeff Var	16.60377		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	33.87407	1.81386	18.68	<.0001
size	1	−0.10174	0.00889	−11.44	<.0001
type	1	8.05547	1.45911	5.52	<.0001

The overall F statistic is still significant (F=72.497, p<0.0001). The intercept and the coefficients associated with size and type are significantly different from zero (t=18.675, p<0.0001; t=−11.443, p<0.0001; t=5.521, p<0.0001, respectively). Notice that the R^2 did not change with the omission of the interaction term.

The fitted model is

$$\text{time} = 33.87 - 0.102 \times \text{size} + 8.055 \times \text{type}$$

The fitted model for a mutual fund company ($X2_i = 0$) is

$$\text{time} = 33.87 - 0.102 \times \text{size}$$

and the fitted model for a stock company ($X2_i = 1$) is

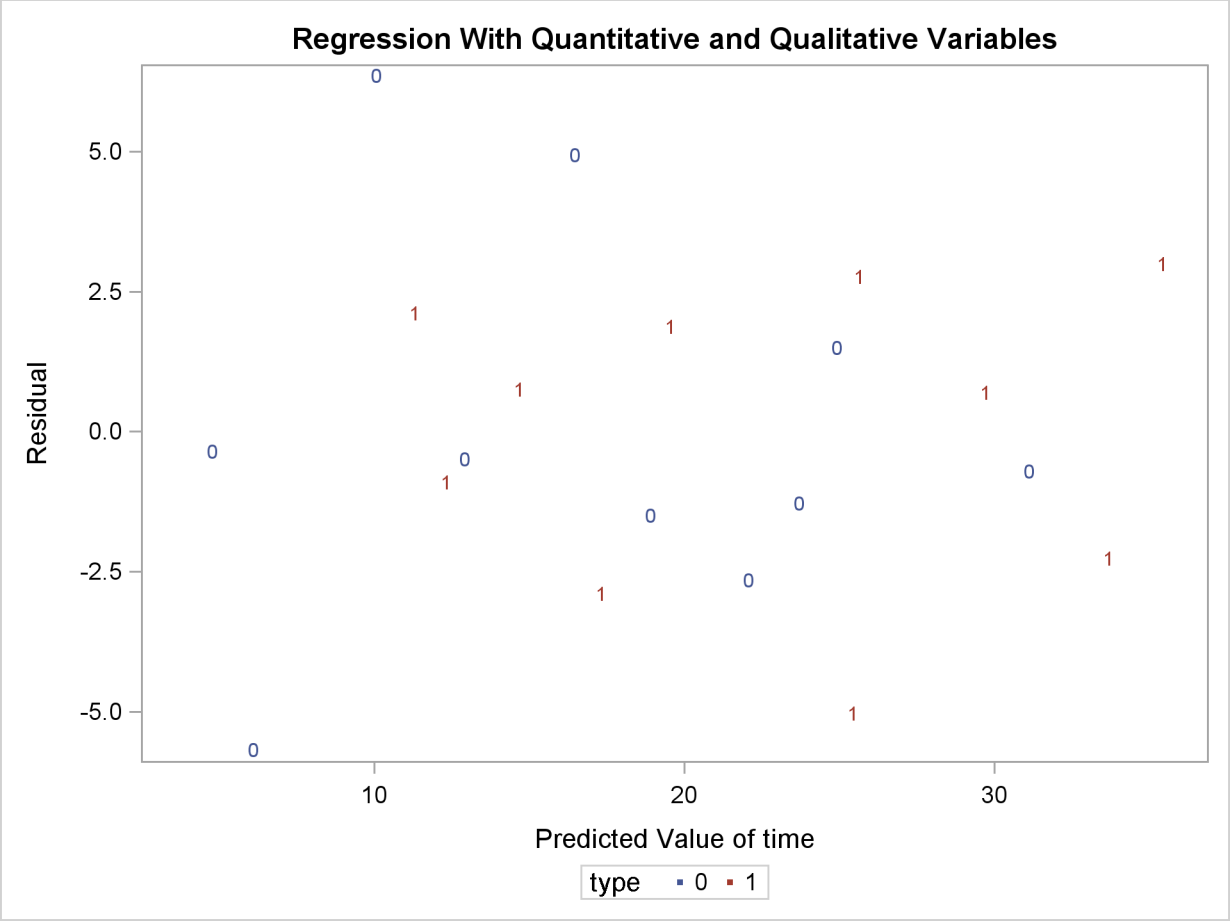
$$\text{time} = (33.87 + 8.055) - 0.102 \times \text{size}$$

So the two models have different intercepts but the same slope.

The following statements first use an OUTPUT statement to save the residuals and predicted values from the new model in the OUT= data set. Next PROC SGPLOT is used to produce Output 73.4.3, which plots residuals versus predicted values. The firm type is used as the plot symbol; this can be useful in determining if the firm types have different residual patterns.

```
   output out=out r=r p=p;
run;

proc sgplot data=out;
  scatter x=p y=r / markerchar=type group=type;
run;
```

Output 73.4.3 Plot of Residual vs. Predicted Values

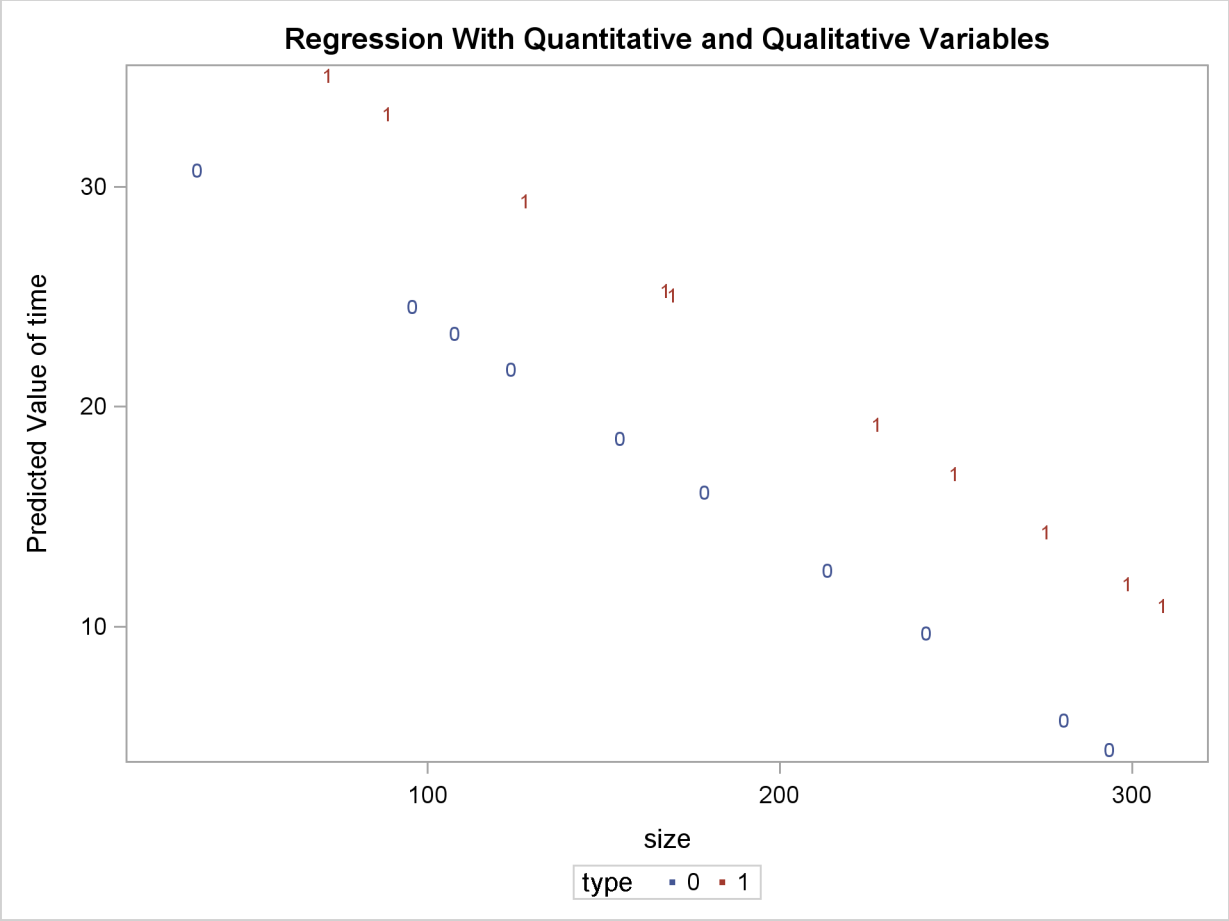


The residuals show no major trend. Neither firm type by itself shows a trend either. This indicates that the model is satisfactory.

The following statements produce the plot of the predicted values versus size that appears in Output 73.4.4, where the firm type is again used as the plotting symbol:

```
proc sgplot data=out;
  scatter x=size y=p / markerchar=type group=type;
run;
```

Output 73.4.4 Plot of Predicted vs. Size



The different intercepts are very evident in this plot.

Example 73.5: Ridge Regression for Acetylene Data

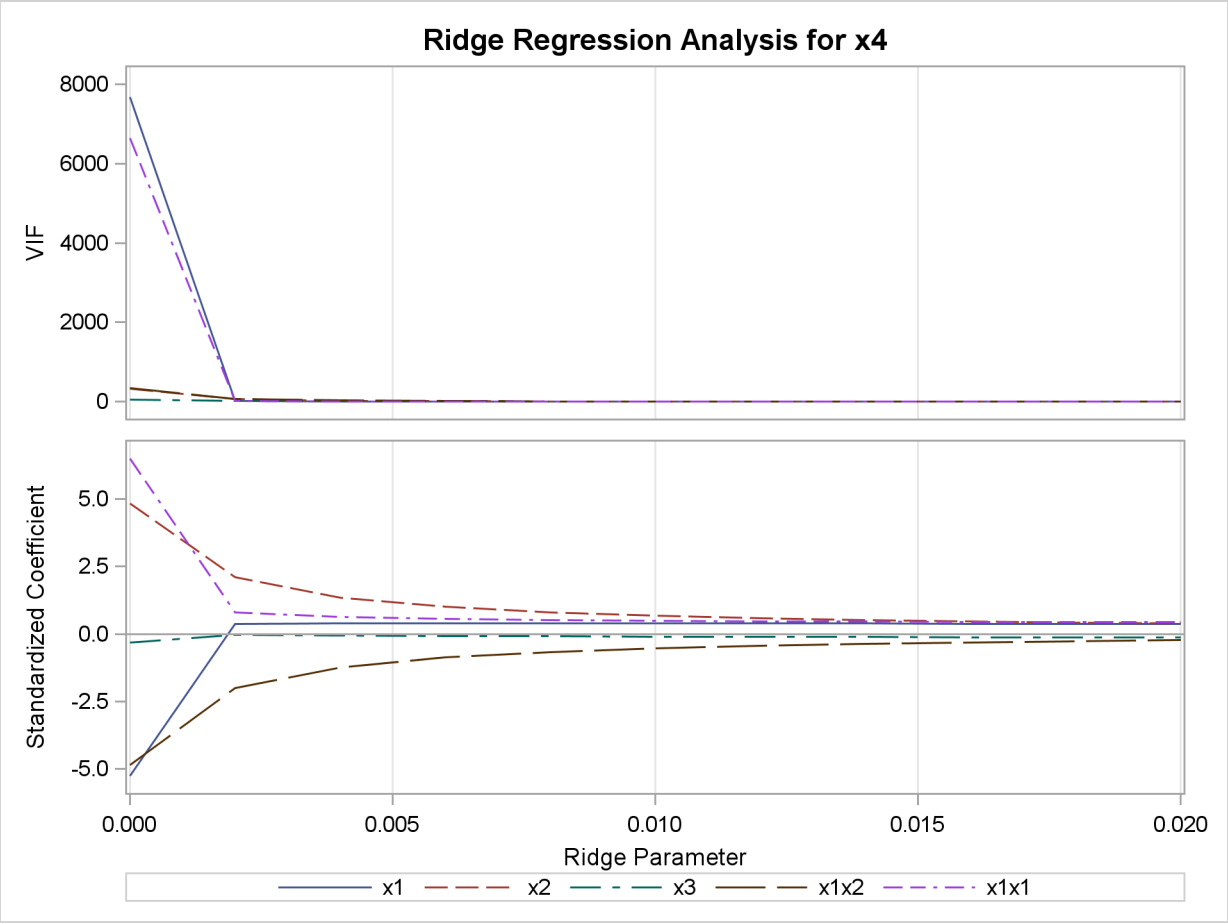
This example uses the acetylene data in Marquardt and Snee (1975) to illustrate the RIDGEPLOT and OUTVIF options. Here are the data:

```
data acetyl;
   input x1-x4 @@;
   x1x2 = x1 * x2;
   x1x1 = x1 * x1;
   label x1   = 'reactor temperature(celsius)'
         x2   = 'h2 to n-heptone ratio'
         x3   = 'contact time(sec)'
         x4   = 'conversion percentage'
         x1x2= 'temperature-ratio interaction'
         x1x1= 'squared temperature';
   datalines;
```

```
1300  7.5 .012 49   1300  9   .012  50.2 1300 11 .0115 50.5
1300 13.5 .013 48.5 1300 17   .0135 47.5 1300 23 .012  44.5
1200  5.3 .04  28   1200  7.5 .038  31.5 1200 11 .032  34.5
1200 13.5 .026 35   1200 17   .034  38   1200 23 .041  38.5
1100  5.3 .084 15   1100  7.5 .098  17   1100 11 .092  20.5
1100 17   .086 29.5
;

ods graphics on;

proc reg data=acetyl outvif
         outest=b ridge=0 to 0.02 by .002;
   model x4=x1 x2 x3 x1x2 x1x1;
run;
proc print data=b;
run;
```

When you enable ODS Graphics and you request ridge regression by using the RIDGE= option in the PROC REG statement, PROC REG produces a panel showing variance inflation factors (VIF) in the upper plot in the panel and ridge traces in the lower plot. This panel is shown in Output 73.5.1.

Output 73.5.1 Ridge Regression and VIF Traces



The OUTVIF option outputs the variance inflation factors to the OUTEST= data set that is shown in Output 73.5.2.

Output 73.5.2 OUTEST Data Set Showing VIF Values

Obs	_MODEL_	_TYPE_	_DEPVAR_	_RIDGE_	_PCOMIT_	_RMSE_	Intercept
1	MODEL1	PARMS	x4	.	.	1.15596	390.538
2	MODEL1	RIDGEVIF	x4	0.000	.	.	.
3	MODEL1	RIDGE	x4	0.000	.	1.15596	390.538
4	MODEL1	RIDGEVIF	x4	0.002	.	.	.
5	MODEL1	RIDGE	x4	0.002	.	2.69721	-103.388
6	MODEL1	RIDGEVIF	x4	0.004	.	.	.
7	MODEL1	RIDGE	x4	0.004	.	3.22340	-93.797
8	MODEL1	RIDGEVIF	x4	0.006	.	.	.
9	MODEL1	RIDGE	x4	0.006	.	3.47752	-87.687
10	MODEL1	RIDGEVIF	x4	0.008	.	.	.
11	MODEL1	RIDGE	x4	0.008	.	3.62677	-83.593
12	MODEL1	RIDGEVIF	x4	0.010	.	.	.
13	MODEL1	RIDGE	x4	0.010	.	3.72505	-80.603
14	MODEL1	RIDGEVIF	x4	0.012	.	.	.
15	MODEL1	RIDGE	x4	0.012	.	3.79477	-78.276
16	MODEL1	RIDGEVIF	x4	0.014	.	.	.
17	MODEL1	RIDGE	x4	0.014	.	3.84693	-76.381
18	MODEL1	RIDGEVIF	x4	0.016	.	.	.
19	MODEL1	RIDGE	x4	0.016	.	3.88750	-74.785
20	MODEL1	RIDGEVIF	x4	0.018	.	.	.
21	MODEL1	RIDGE	x4	0.018	.	3.92004	-73.407
22	MODEL1	RIDGEVIF	x4	0.020	.	.	.
23	MODEL1	RIDGE	x4	0.020	.	3.94679	-72.193

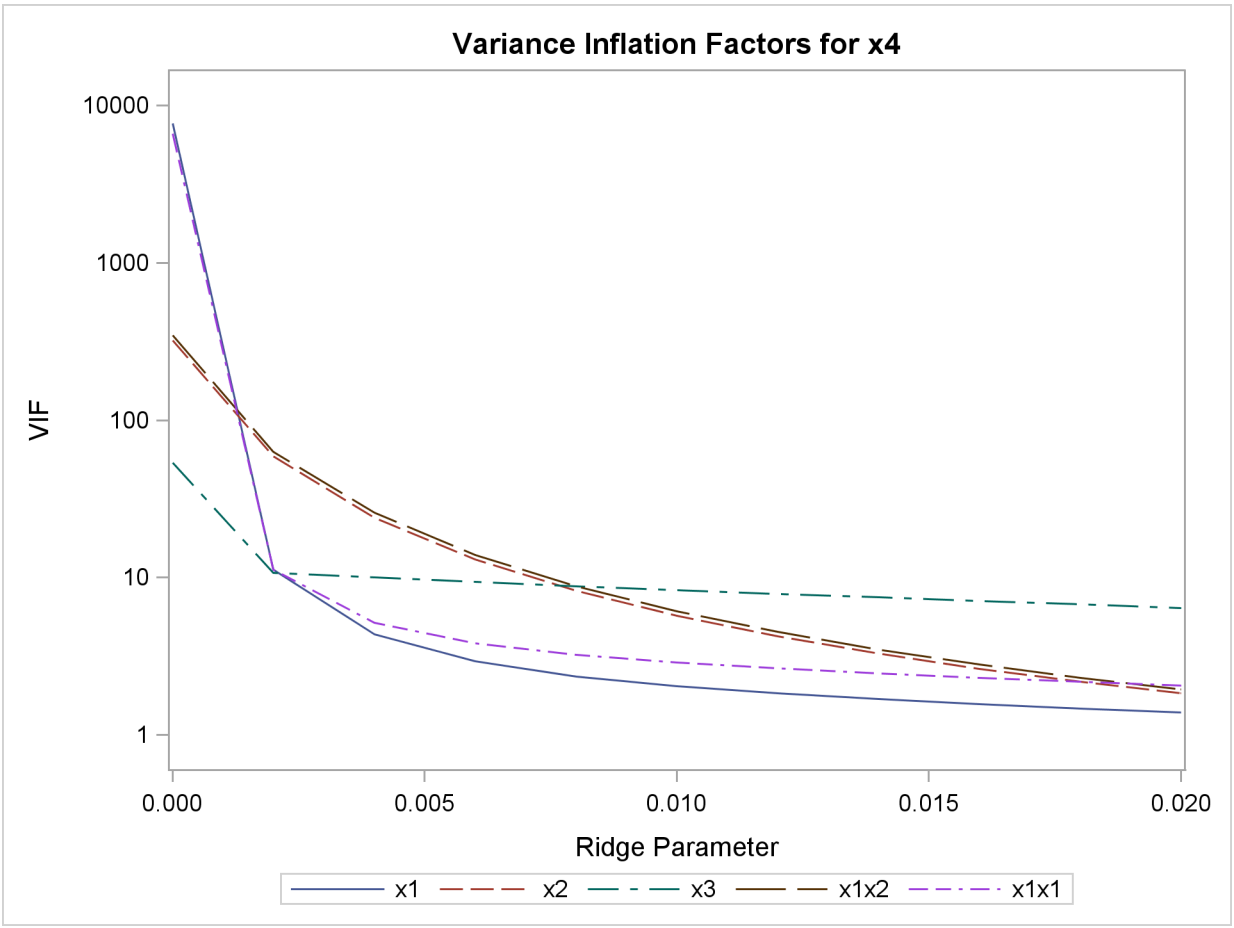
Obs	x1	x2	x3	x1x2	x1x1	x4
1	-0.78	10.174	-121.626	-0.008	0.00	-1
2	7682.37	320.022	53.525	344.545	6643.32	-1
3	-0.78	10.174	-121.626	-0.008	0.00	-1
4	11.18	58.731	10.744	63.208	11.22	-1
5	0.05	4.404	-9.065	-0.003	0.00	-1
6	4.36	23.939	9.996	25.744	5.15	-1
7	0.06	2.839	-21.338	-0.002	0.00	-1
8	2.93	13.011	9.383	13.976	3.81	-1
9	0.06	2.110	-28.447	-0.001	0.00	-1
10	2.36	8.224	8.838	8.821	3.23	-1
11	0.06	1.689	-33.377	-0.001	0.00	-1
12	2.04	5.709	8.343	6.112	2.89	-1
13	0.06	1.414	-37.177	-0.001	0.00	-1
14	1.84	4.226	7.891	4.514	2.65	-1
15	0.06	1.221	-40.297	-0.001	0.00	-1
16	1.69	3.279	7.476	3.493	2.46	-1
17	0.06	1.078	-42.965	-0.001	0.00	-1
18	1.57	2.637	7.094	2.801	2.31	-1
19	0.06	0.968	-45.309	-0.001	0.00	-1
20	1.47	2.182	6.741	2.310	2.18	-1
21	0.06	0.880	-47.407	-0.000	0.00	-1
22	1.39	1.847	6.415	1.949	2.06	-1
23	0.06	0.809	-49.310	-0.000	0.00	-1

If you want to obtain separate plots containing the ridge traces and VIF traces, you can specify the UNPACK suboption in the PLOTS=RIDGE option. You can also request that one or both of the VIF axis and ridge parameter axis be displayed on a logarithmic scale. You can see in Output 73.5.1 that the VIF traces for several of the parameters are nearly indistinguishable when displayed on a linear scale. The following code illustrates how you obtain separate VIF and ridge traces with the VIF values displayed on a logarithmic scale. Note that you can obtain plots of VIF values even though you do not specify the OUTVIF option in the PROC REG statement.

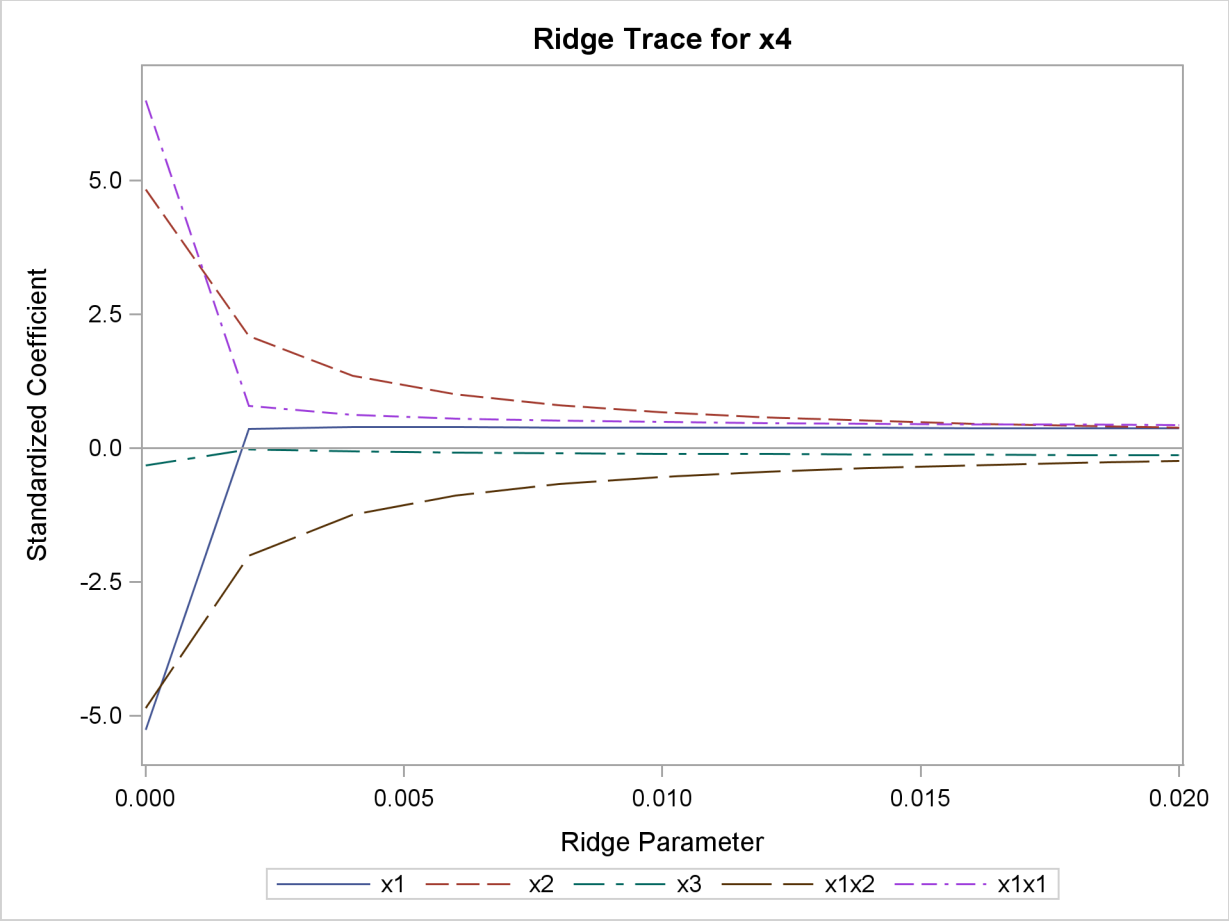
```
proc reg data=acetyl plots(only)=ridge(unpack VIFaxis=log)
         outest=b ridge=0 to 0.02 by .002;
   model x4=x1 x2 x3 x1x2 x1x1;
run;

ods graphics off;
```

The requested plots are shown in Output 73.5.3 and Output 73.5.4.

Output 73.5.3 VIF Traces



Output 73.5.4 Ridge Traces



Example 73.6: Chemical Reaction Response

This example shows how you can use lack-of-fit tests with the REG procedure. See the section "Testing for Lack of Fit" on page 5576 for details about lack-of-fit tests.

In a study of the percentage of raw material that responds in a reaction, researchers identified the following five factors:

- the feed rate of the chemicals (FeedRate), ranging from 10 to 15 liters per minute
- the percentage of the catalyst (Catalyst), ranging from 1% to 2%
- the agitation rate of the reactor (AgitRate), ranging from 100 to 120 revolutions per minute
- the temperature (Temperature), ranging from 140 to 180 degrees Celsius
- the concentration (Concentration), ranging from 3% to 6%

The following data set contains the results of an experiment designed to estimate main effects for all factors:

```
data reaction;
  input FeedRate Catalyst AgitRate Temperature
        Concentration ReactionPercentage;
datalines;
10.0   1.0  100   140  6.0   37.5
10.0   1.0  120   180  3.0   28.5
10.0   2.0  100   180  3.0   40.4
10.0   2.0  120   140  6.0   48.2
15.0   1.0  100   180  6.0   50.7
15.0   1.0  120   140  3.0   28.9
15.0   2.0  100   140  3.0   43.5
15.0   2.0  120   180  6.0   64.5
12.5   1.5  110   160  4.5   39.0
12.5   1.5  110   160  4.5   40.3
12.5   1.5  110   160  4.5   38.7
12.5   1.5  110   160  4.5   39.7
;
```

The first eight runs of this experiment enable orthogonal estimation of the main effects for all factors. The last four comprise four replicates of the centerpoint.

The following statements fit a linear model. Because this experiment includes replications, you can test for lack of fit by using the LACKFIT option in the MODEL statement.

```
proc reg data=reaction;
   model  ReactionPercentage=FeedRate Catalyst AgitRate
                             Temperature Concentration / lackfit;
run;
```

Output 73.6.1 shows that the lack of fit for the linear model is significant, indicating that a more complex model is required. Models that include interactions should be investigated. In this case, this will require additional experimentation to obtain appropriate data for estimating the effects.

Output 73.6.1 Analysis of Variance

The REG Procedure
Model: MODEL1
Dependent Variable: ReactionPercentage

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	990.27000	198.05400	33.29	0.0003
Error	6	35.69917	5.94986		
Lack of Fit	3	34.15167	11.38389	22.07	0.0151
Pure Error	3	1.54750	0.51583		
Corrected Total	11	1025.96917			

Root MSE	2.43923	R-Square	0.9652
Dependent Mean	41.65833	Adj R-Sq	0.9362
Coeff Var	5.85533		

References

Akaike, H. (1969), "Fitting Autoregressive Models for Prediction," *Annals of the Institute of Statistical Mathematics*, 21, 243–247.

Allen, D. M. (1971), "Mean Square Error of Prediction as a Criterion for Selecting Variables," *Technometrics*, 13, 469–475.

Allen, D. M. and Cady, F. B. (1982), *Analyzing Experimental Data by Regression*, Belmont, CA: Lifetime Learning Publications.

Amemiya, T. (1976), "Selection of Regressors," Technical Report No. 225, Stanford, CA: Stanford University.

Belsley, D. A., Kuh, E., and Welsch, R. E. (1980), *Regression Diagnostics*, New York: John Wiley & Sons.

Berk, K. N. (1977), "Tolerance and Condition in Regression Computations," *Journal of the American Statistical Association*, 72, 863–866.

Bock, R. D. (1975), *Multivariate Statistical Methods in Behavioral Research*, New York: McGraw-Hill.

Box, G. E. P. (1966), "The Use and Abuse of Regression," *Technometrics*, 8, 625–629.

Cleveland, W. S. (1993), *Visualizing Data*, Summit, NJ: Hobart Press.

Collier Books (1987), *The 1987 Baseball Encyclopedia Update,* New York: Macmillan.

Cook, R. D. (1977), "Detection of Influential Observations in Linear Regression," *Technometrics*, 19, 15–18.

Cook, R. D. (1979), "Influential Observations in Linear Regression," *Journal of the American Statistical Association*, 74, 169–174.

Daniel, C. and Wood, F. (1980), *Fitting Equations to Data*, Revised Edition, New York: John Wiley & Sons.

Darlington, R. B. (1968), "Multiple Regression in Psychological Research and Practice," *Psychological Bulletin*, 69, 161–182.

Draper, N. and Smith, H. (1981), *Applied Regression Analysis*, Second Edition, New York: John Wiley & Sons.

Durbin, J. and Watson, G. S. (1951), "Testing for Serial Correlation in Least Squares Regression," *Biometrika*, 37, 409–428.

Freund, R. J. and Littell, R. C. (1986), *SAS System for Regression, 1986 Edition*, Cary, NC: SAS Institute Inc.

Furnival, G. M. and Wilson, R. W. (1974), "Regression by Leaps and Bounds," *Technometrics*, 16, 499–511.

Goodnight, J. H. (1979), "A Tutorial on the SWEEP Operator," *The American Statistician*, 33, 149–158. (Also available as *The Sweep Operator: Its Importance in Statistical Computing*, SAS Technical Report R-106.)

Hocking, R. R. (1976), "The Analysis and Selection of Variables in Linear Regression," *Biometrics*, 32, 1–50.

Johnston, J. (1972), *Econometric Methods*, New York: McGraw-Hill.

Judge, G. G., Griffiths, W. E., Hill, R. C., and Lee, T. (1980), *The Theory and Practice of Econometrics*, New York: John Wiley & Sons.

Judge, G. G., Griffiths, W. E., Hill, R. C., Lutkepohl, H., and Lee, T. C. (1985), *The Theory and Practice of Econometrics,* Second Edition, New York: John Wiley & Sons.

Kennedy, W. J. and Gentle, J. E. (1980), *Statistical Computing*, New York: Marcel Dekker.

LaMotte, L. R. (1994), "A Note on the Role of Independence in t Statistics Constructed from Linear Statistics in Regression Models," *The American Statistician*, 48, 238–240.

Lewis, T. and Taylor, L. R. (1967), *Introduction to Experimental Ecology*, New York: Academic Press.

Long, J. S. and Ervin, L. H.(2000), "Correcting for Heteroscedasticity with Heteroscedasticity Consistent Standard Errors in the Linear Regression Model: Small Sample Considerations," *The American Statistician*, 54, 217–224.

Lord, F. M. (1950), "Efficiency of Prediction When a Progression Equation from One Sample Is Used in a New Sample," Research Bulletin No. 50-40, Princeton, NJ: Educational Testing Service.

MacKinnon, J. G. and White, H. (1985), "Some Heteroskedasticity Consistent Covariance matrix Estimators with Improved Finite Sample Properties," *Journal of Econometrics*, 29, 53–57.

Mallows, C. L. (1967), "Choosing a Subset Regression," unpublished report, Bell Telephone Laboratories.

Mallows, C. L. (1973), "Some Comments on C_p," *Technometrics*, 15, 661–675.

Mardia, K. V., Kent, J. T., and Bibby, J. M. (1979), *Multivariate Analysis*, London: Academic Press.

Marquardt, D. W. and Snee, R. D. (1975), "Ridge Regression in Practice," *American Statistician*, 29 (1), 3–20.

Morrison, D. F. (1976), *Multivariate Statistical Methods*, Second Edition, New York: McGraw-Hill.

Mosteller, F. and Tukey, J. W. (1977), *Data Analysis and Regression*, Reading, MA: Addison-Wesley.

Neter, J., Wasserman, W., and Kutner, M. H. (1990), *Applied Linear Statistical Models*, Third

Edition, Homewood, IL: Irwin.

Nicholson, G. E., Jr. (1948), "The Application of a Regression Equation to a New Sample," unpublished Ph.D. dissertation, University of North Carolina at Chapel Hill.

Pillai, K. C. S. (1960), *Statistical Table for Tests of Multivariate Hypotheses*, Manila: The Statistical Center, University of the Philippines.

Pindyck, R. S. and Rubinfeld, D. L. (1981), *Econometric Models and Econometric Forecasts*, Second Edition, New York: McGraw-Hill.

Pringle, R. M. and Rayner, A. A. (1971), *Generalized Inverse Matrices with Applications to Statistics*, New York: Hafner Publishing.

Rao, C. R. (1973), *Linear Statistical Inference and Its Applications*, Second Edition, New York: John Wiley & Sons.

Rawlings, J. O. (1988), *Applied Regression Analysis: A Research Tool*, Belmont, CA: Wadsworth.

Rothman, D. (1968), letter to the editor, *Technometrics*, 10, 432.

Sall, J. P. (1981), *SAS Regression Applications*, Revised Edition, SAS Technical Report A-102, Cary, NC: SAS Institute Inc.

Sawa, T. (1978), "Information Criteria for Discriminating among Alternative Regression Models," *Econometrica*, 46, 1273–1282.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Sports Illustrated, April 20, 1987.

Stein, C. (1960), "Multiple Regression," in *Contributions to Probability and Statistics*, eds. I. Olkin et al., Stanford, CA: Stanford University Press.

Timm, N. H. (1975), *Multivariate Analysis with Applications in Education and Psychology*, Monterey, CA: Brooks-Cole.

Weisberg, S. (1980), *Applied Linear Regression*, New York: John Wiley & Sons.

White, H. (1980), "A Heteroskedasticity-Consistent Covariance Matrix Estimator and a Direct Test for Heteroskedasticity," *Econometrics*, 48, 817–838.

Chapter 74

The ROBUSTREG Procedure

Contents

Overview: ROBUSTREG Procedure **5648**
Features 5648
Getting Started: ROBUSTREG Procedure **5649**
M Estimation 5649
LTS Estimation 5656
Syntax: ROBUSTREG Procedure **5660**
PROC ROBUSTREG Statement 5660
BY Statement 5667
CLASS Statement 5668
ID Statement 5668
MODEL Statement 5668
OUTPUT Statement 5670
PERFORMANCE Statement 5671
TEST Statement 5671
WEIGHT Statement 5672
Details: ROBUSTREG Procedure **5672**
M Estimation 5672
High Breakdown Value Estimation 5679
MM Estimation 5684
Robust Distance 5688
Leverage Point and Outlier Detection 5689
INEST= Data Set 5689
OUTEST= Data Set 5690
Computational Resources 5690
ODS Table Names 5691
ODS Graphics 5692
Examples: ROBUSTREG Procedure **5697**
Example 74.1: Comparison of Robust Estimates 5697
Example 74.2: Robust ANOVA 5704
Example 74.3: Growth Study of De Long and Summers 5708
References **5716**

Overview: ROBUSTREG Procedure

The main purpose of robust regression is to detect outliers and provide resistant (stable) results in the presence of outliers. In order to achieve this stability, robust regression limits the influence of outliers. Historically, three classes of problems have been addressed with robust regression techniques:

- problems with outliers in the y-direction (response direction)
- problems with multivariate outliers in the x-space (i.e., outliers in the covariate space, which are also referred to as leverage points)
- problems with outliers in both the y-direction and the x-space

Many methods have been developed in response to these problems. However, in statistical applications of outlier detection and robust regression, the methods most commonly used today are Huber M estimation, high breakdown value estimation, and combinations of these two methods. The ROBUSTREG procedure in SAS 9.2 provides four such methods: M estimation, LTS estimation, S estimation, and MM estimation.

1. M estimation was introduced by Huber (1973), and it is the simplest approach both computationally and theoretically. Although it is not robust with respect to leverage points, it is still used extensively in analyzing data for which it can be assumed that the contamination is mainly in the response direction.
2. Least trimmed squares (LTS) estimation is a high breakdown value method introduced by Rousseeuw (1984). The breakdown value is a measure of the proportion of contamination that an estimation method can withstand and still maintain its robustness. The performance of this method was improved by the FAST-LTS algorithm of Rousseeuw and Van Driessen (2000).
3. S estimation is a high breakdown value method introduced by Rousseeuw and Yohai (1984). With the same breakdown value, it has a higher statistical efficiency than LTS estimation.
4. MM estimation, introduced by Yohai (1987), combines high breakdown value estimation and M estimation. It has both the high breakdown property and a higher statistical efficiency than S estimation.

Features

The main features of the ROBUSTREG procedure are as follows:

- offers four estimation methods: M, LTS, S, and MM

- provides 10 weight functions for M estimation
- provides robust R^2 and deviance for all estimates
- provides asymptotic covariance and confidence intervals for regression parameter with the M, S, and MM methods
- provides robust Wald and F tests for regression parameters with the M and MM methods
- provides outlier and leverage-point diagnostics
- supports parallel computing for S and LTS estimates
- produces fit plots and diagnostic plots by using ODS Graphics

Getting Started: ROBUSTREG Procedure

The following examples demonstrate how you can use the ROBUSTREG procedure to fit a linear regression model and obtain outlier and leverage-point diagnostics.

M Estimation

This example shows how you can use the ROBUSTREG procedure to do M estimation, which is a commonly used method for outlier detection and robust regression when contamination is mainly in the response direction.

```
data stack;
   input  x1 x2 x3 y exp$ @@;
datalines;
80  27   89  42   e1   80  27   88  37   e2
75  25   90  37   e3   62  24   87  28   e4
62  22   87  18   e5   62  23   87  18   e6
62  24   93  19   e7   62  24   93  20   e8
58  23   87  15   e9   58  18   80  14  e10
58  18   89  14  e11   58  17   88  13  e12
58  18   82  11  e13   58  19   93  12  e14
50  18   89   8  e15   50  18   86   7  e16
50  19   72   8  e17   50  19   79   8  e18
50  20   80   9  e19   56  20   82  15  e20
70  20   91  15  e21
;
```

The data set stack is the well-known stackloss data set presented by Brownlee (1965). The data describe the operation of a plant for the oxidation of ammonia to nitric acid and consist of 21 four-dimensional observations. The explanatory variables for the response stackloss (y) are the rate of operation (x1), the cooling water inlet temperature (x2), and the acid concentration (x3).

The following ROBUSTREG statements analyze the data:

```
proc robustreg data=stack;
   model y = x1 x2 x3 / diagnostics leverage;
   id    exp;
   test  x3;
run;
```

By default, the procedure does M estimation with the bisquare weight function, and it uses the median method for estimating the scale parameter. The MODEL statement specifies the covariate effects. The DIAGNOSTICS option requests a table for outlier diagnostics, and the LEVERAGE option adds leverage-point diagnostic results to this table for continuous covariate effects. The ID statement specifies that the variable exp is used to identify each observation (experiment) in this table. If the ID statement is omitted, the observation number is used to identify the observations. The TEST statement requests a test of significance for the covariate effects specified. The results of this analysis are displayed in the following figures.

Figure 74.1 Model Fitting Information and Summary Statistics

The ROBUSTREG Procedure

Model Information

Data Set	WORK.STACK
Dependent Variable	y
Number of Independent Variables	3
Number of Observations	21
Method	M Estimation

Summary Statistics

Variable	Q1	Median	Q3	Mean	Standard Deviation	MAD
x1	53.0000	58.0000	62.0000	60.4286	9.1683	5.9304
x2	18.0000	20.0000	24.0000	21.0952	3.1608	2.9652
x3	82.0000	87.0000	89.5000	86.2857	5.3586	4.4478
y	10.0000	15.0000	19.5000	17.5238	10.1716	5.9304

Figure 74.1 displays the model fitting information and summary statistics for the response variable and the continuous covariates. The columns labeled Q1, Median, and Q3 provide the lower quantile, median, and upper quantile, respectively. The column labeled MAD provides a robust estimate of the univariate scale, which is computed as the standardized median absolute deviation (MAD). See Huber (1981, p. 108) for more details about the standardized MAD. The columns labeled Mean and Standard Deviation provide the usual mean and standard deviation. A large difference between the standard deviation and the MAD for a variable indicates some extreme values for this variable. In the stackloss data, the stackloss (response y) has the biggest difference between the standard deviation and the MAD.

Figure 74.2 Model Parameter Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-42.2854	9.5045	-60.9138	-23.6569	19.79	<.0001
x1	1	0.9276	0.1077	0.7164	1.1387	74.11	<.0001
x2	1	0.6507	0.2940	0.0744	1.2270	4.90	0.0269
x3	1	-0.1123	0.1249	-0.3571	0.1324	0.81	0.3683
Scale	1	2.2819					

Figure 74.2 displays the table of robust parameter estimates, standard errors, and confidence limits. The row labeled Scale provides a point estimate of the scale parameter in the linear regression model, which is obtained by the median method. See the section "M Estimation" on page 5672 for more information about scale estimation methods. For the stackloss data, M estimation yields the fitted linear model:

$$\hat{y} = -42.2845 + 0.9276\text{x1} + 0.6507\text{x2} - 0.1123\text{x3}$$

Figure 74.3 Diagnostics

Diagnostics

Obs	exp	Mahalanobis Distance	Robust MCD Distance	Leverage	Standardized Robust Residual	Outlier
1	e1	2.2536	5.5284	*	1.0995	
2	e2	2.3247	5.6374	*	-1.1409	
3	e3	1.5937	4.1972	*	1.5604	
4	e4	1.2719	1.5887		3.0381	*
21	e21	2.1768	3.6573	*	-4.5733	*

Figure 74.3 displays outlier and leverage-point diagnostics. Standardized robust residuals are computed based on the estimated parameters. Both the Mahalanobis distance and the robust MCD distance are displayed. Outliers and leverage points, identified with asterisks, are defined by the standardized robust residuals and robust MCD distances that exceed the corresponding cutoff values displayed in the diagnostics summary. Observations 4 and 21 are outliers because their standardized robust residuals exceed the cutoff value in absolute value. The procedure detects 4 observations with high leverage. Leverage points (points with high leverage) with smaller standardized robust residuals than the cutoff value in absolute value are called good leverage points; others are called bad leverage points. Observation 21 is a bad leverage point.

Two particularly useful plots for revealing outliers and leverage points are a scatter plot of the standardized robust residuals against the robust distances (RDPLOT) and a scatter plot of the robust distances against the classical Mahalanobis distances (DDPLOT).

For the stackloss data, the following statements produce the RDPLOT in Figure 74.4 and the DDPLOT in Figure 74.5. The histogram and the normal quantile-quantile plots for the standardized robust residuals are also created with the HISTOGRAM and QQPLOT options in the PROC ROBUSTREG statement. See Figure 74.6 and Figure 74.7.

```
ods graphics on;

proc robustreg data=stack
        plots=(rdplot ddplot reshistogram resqqplot);
   model y = x1 x2 x3;
run;

ods graphics off;
```

Figure 74.4 RDPLOT for Stackloss Data

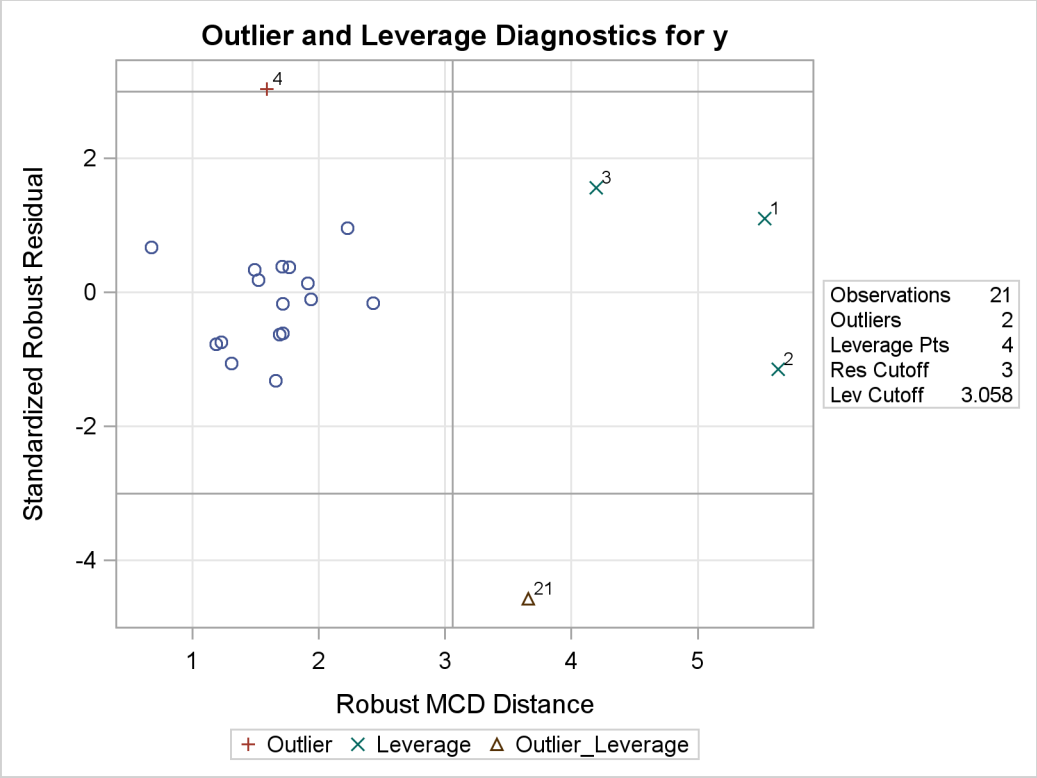


Figure 74.5 DDPLOT for Stackloss Data

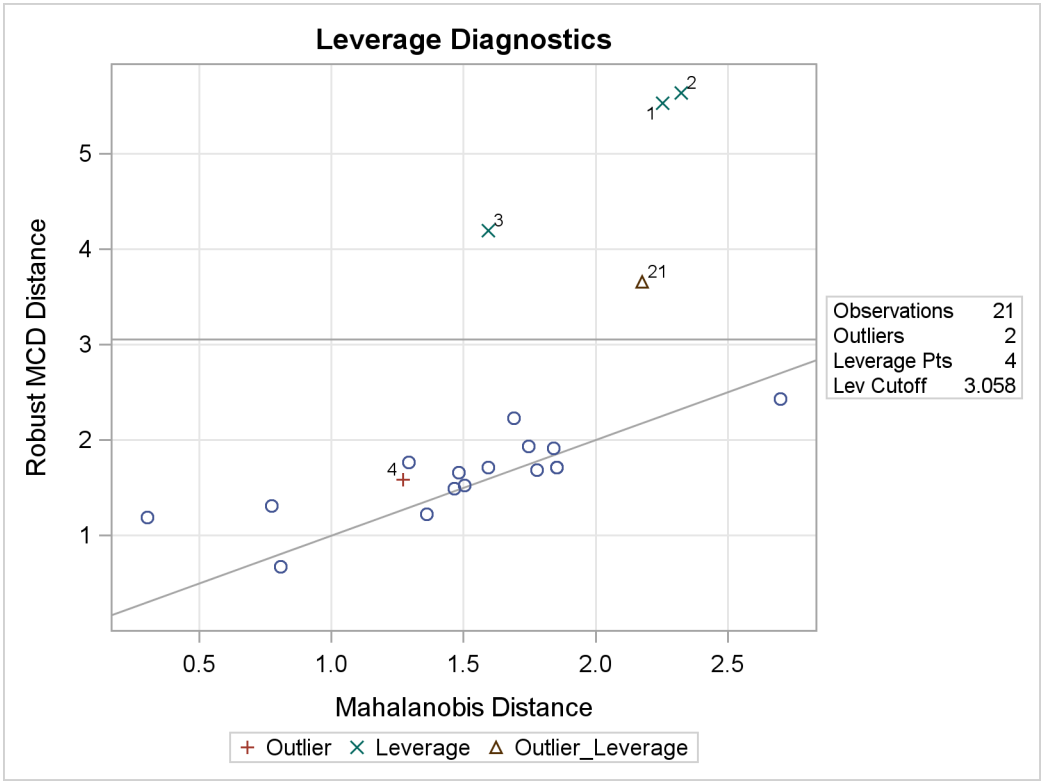


Figure 74.6 Histogram

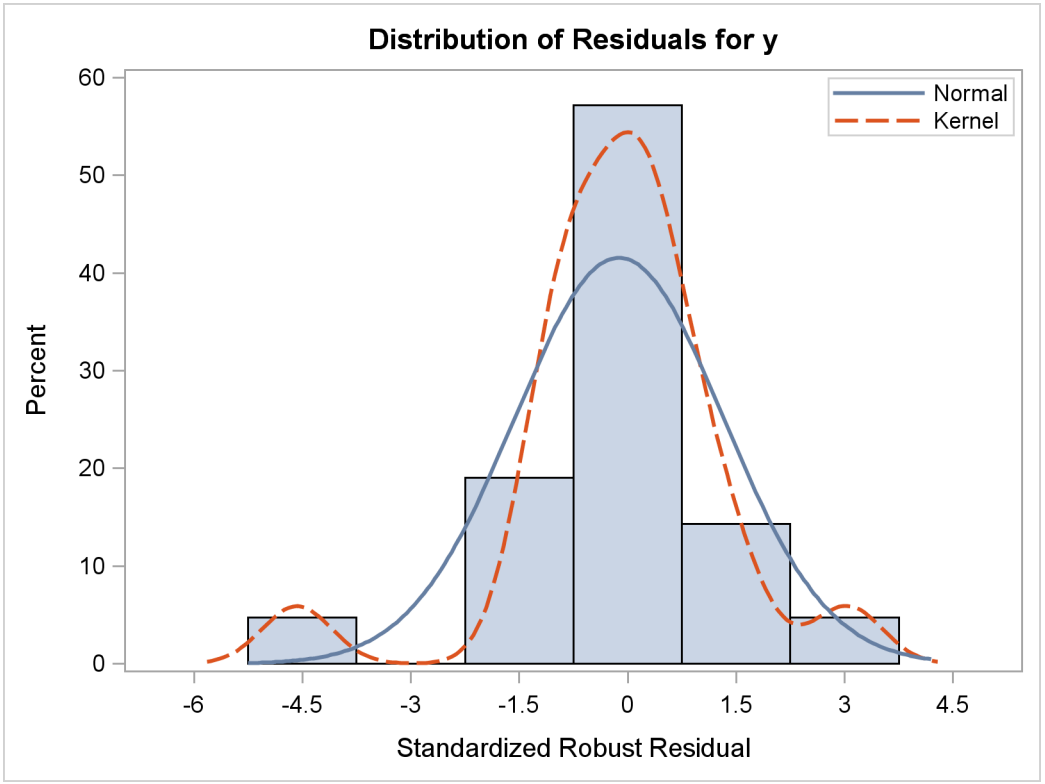
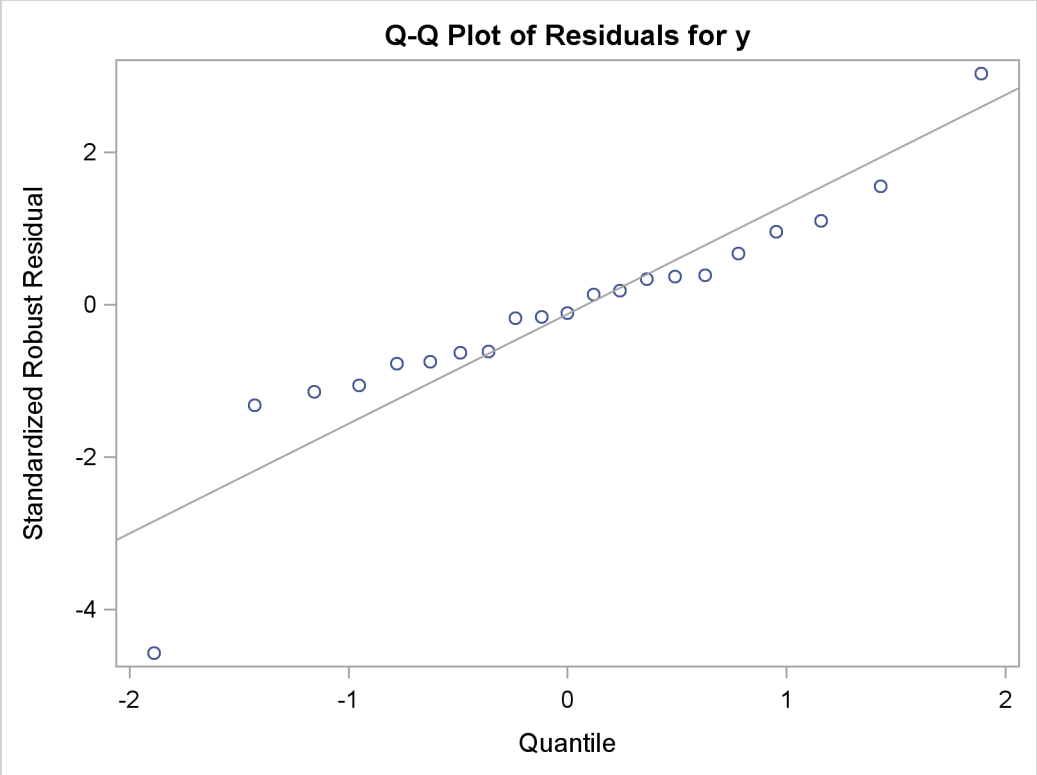


Figure 74.7 Q-Q Plot



These plots are helpful in identifying outliers as well as good and bad high leverage points.

These graphical displays are requested by specifying the ODS GRAPHICS statement and the PLOTS= option in the PROC statement. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the ROBUSTREG procedure, see the section "ODS Graphics" on page 5692.

Figure 74.8 Goodness-of-Fit Statistics

Goodness-of-Fit

Statistic	Value
R-Square	0.6659
AICR	29.5231
BICR	36.3361
Deviance	125.7905

Figure 74.8 displays robust versions of goodness-of-fit statistics for the model. You can use the robust information criteria, AICR and BICR, for model selection and comparison. For both AICR and BICR, the lower the value, the more desirable the model.

Figure 74.9 Test of Significance

Robust Linear Tests

Test

Test	Test Statistic	Lambda	DF	Chi-Square	Pr > ChiSq
Rho	0.9378	0.7977	1	1.18	0.2782
Rn2	0.8092		1	0.81	0.3683

Figure 74.9 displays the test results requested by the TEST statement. The ROBUSTREG procedure conducts two robust linear tests, the ρ test and the R_n^2 test. See the section "Linear Tests" on page 5678 for information about how the procedure computes test statistics and the correction factor lambda. Due to the large p-values for both tests, you can conclude that the effect x3 is not significant at the 5% level.

For the bisquare weight function, the default tuning constant, $c = 4.685$, is chosen to yield a 95% asymptotic efficiency of the M estimates with the Gaussian distribution. See the section "M Estimation" on page 5672 for details. The smaller the constant c, the lower the asymptotic efficiency but the sharper the M estimate as an outlier detector. For the stackloss data set, you could consider using a sharper outlier detector.

In the following invocation of the ROBUSTREG procedure, a smaller constant, $c = 3.5$, is used. This tuning constant corresponds to an efficiency close to 85%. See Chen and Yin (2002) for relations between the tuning constant and asymptotic efficiency of M estimates.

```
proc robustreg method=m(wf=bisquare(c=3.5)) data=stack;
   model y = x1 x2 x3 / diagnostics leverage;
   id    exp;
   test  x3;
run;
```

Figure 74.10 Model Parameter Estimates

The ROBUSTREG Procedure

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-37.1076	5.4731	-47.8346	-26.3805	45.97	<.0001
x1	1	0.8191	0.0620	0.6975	0.9407	174.28	<.0001
x2	1	0.5173	0.1693	0.1855	0.8492	9.33	0.0022
x3	1	-0.0728	0.0719	-0.2138	0.0681	1.03	0.3111
Scale	1	1.4265					

Figure 74.10 displays the table of robust parameter estimates, standard errors, and confidence limits with the constant $c = 3.5$.

The refitted linear model is

$$\hat{y} = -37.1076 + 0.8191x1 + 0.5173x2 - 0.0728x3$$

Figure 74.11 Diagnostics

Diagnostics

Obs	exp	Mahalanobis Distance	Robust MCD Distance	Leverage	Standardized Robust Residual	Outlier
1	e1	2.2536	5.5284	*	4.2719	*
2	e2	2.3247	5.6374	*	0.7158	
3	e3	1.5937	4.1972	*	4.4142	*
4	e4	1.2719	1.5887		5.7792	*
21	e21	2.1768	3.6573	*	-6.2727	*

Figure 74.11 displays outlier and leverage-point diagnostics with the constant $c = 3.5$. Besides observations 4 and 21, observations 1 and 3 are also detected as outliers.

LTS Estimation

If the data are contaminated in the x-space, M estimation does not do well. The following example shows how you can use LTS estimation to deal with this situation.

```
data hbk;
   input index$ x1 x2 x3 y @@;
datalines;
1   10.1  19.6  28.3   9.7     39  2.1  0.0  1.2  -0.7
2    9.5  20.5  28.9  10.1     40  0.5  2.0  1.2  -0.5
3   10.7  20.2  31.0  10.3     41  3.4  1.6  2.9  -0.1
4    9.9  21.5  31.7   9.5     42  0.3  1.0  2.7  -0.7
5   10.3  21.1  31.1  10.0     43  0.1  3.3  0.9   0.6
6   10.8  20.4  29.2  10.0     44  1.8  0.5  3.2  -0.7
7   10.5  20.9  29.1  10.8     45  1.9  0.1  0.6  -0.5
8    9.9  19.6  28.8  10.3     46  1.8  0.5  3.0  -0.4
9    9.7  20.7  31.0   9.6     47  3.0  0.1  0.8  -0.9
10   9.3  19.7  30.3   9.9     48  3.1  1.6  3.0   0.1
11  11.0  24.0  35.0  -0.2     49  3.1  2.5  1.9   0.9
12  12.0  23.0  37.0  -0.4     50  2.1  2.8  2.9  -0.4
13  12.0  26.0  34.0   0.7     51  2.3  1.5  0.4   0.7
14  11.0  34.0  34.0   0.1     52  3.3  0.6  1.2  -0.5
15   3.4   2.9   2.1  -0.4     53  0.3  0.4  3.3   0.7
16   3.1   2.2   0.3   0.6     54  1.1  3.0  0.3   0.7
17   0.0   1.6   0.2  -0.2     55  0.5  2.4  0.9   0.0
18   2.3   1.6   2.0   0.0     56  1.8  3.2  0.9   0.1
19   0.8   2.9   1.6   0.1     57  1.8  0.7  0.7   0.7
```

```
20   3.1   3.4   2.2   0.4     58  2.4  3.4  1.5  -0.1
21   2.6   2.2   1.9   0.9     59  1.6  2.1  3.0  -0.3
22   0.4   3.2   1.9   0.3     60  0.3  1.5  3.3  -0.9
23   2.0   2.3   0.8  -0.8     61  0.4  3.4  3.0  -0.3
24   1.3   2.3   0.5   0.7     62  0.9  0.1  0.3   0.6
25   1.0   0.0   0.4  -0.3     63  1.1  2.7  0.2  -0.3
26   0.9   3.3   2.5  -0.8     64  2.8  3.0  2.9  -0.5
27   3.3   2.5   2.9  -0.7     65  2.0  0.7  2.7   0.6
28   1.8   0.8   2.0   0.3     66  0.2  1.8  0.8  -0.9
29   1.2   0.9   0.8   0.3     67  1.6  2.0  1.2  -0.7
30   1.2   0.7   3.4  -0.3     68  0.1  0.0  1.1   0.6
31   3.1   1.4   1.0   0.0     69  2.0  0.6  0.3   0.2
32   0.5   2.4   0.3  -0.4     70  1.0  2.2  2.9   0.7
33   1.5   3.1   1.5  -0.6     71  2.2  2.5  2.3   0.2
34   0.4   0.0   0.7  -0.7     72  0.6  2.0  1.5  -0.2
35   3.1   2.4   3.0   0.3     73  0.3  1.7  2.2   0.4
36   1.1   2.2   2.7  -1.0     74  0.0  2.2  1.6  -0.9
37   0.1   3.0   2.6  -0.6     75  0.3  0.4  2.6   0.2
38   1.5   1.2   0.2   0.9
;
```

The data set hbk is an artificial data set generated by Hawkins, Bradu, and Kass (1984). Both ordinary least squares (OLS) estimation and M estimation (not shown here) suggest that observations 11 to 14 are outliers. However, these four observations were generated from the underlying model, whereas observations 1 to 10 were contaminated. The reason that OLS estimation and M estimation do not pick up the contaminated observations is that they cannot distinguish good leverage points (observations 11 to 14) from bad leverage points (observations 1 to 10). In such cases, the LTS method identifies the true outliers.

The following statements invoke the ROBUSTREG procedure with the LTS estimation method:

```
proc robustreg data=hbk fwls method=lts;
   model y = x1 x2 x3 / diagnostics leverage;
   id index;
run;
```

Figure 74.12 Model Fitting Information and Summary Statistics

The ROBUSTREG Procedure

Model Information

Data Set	WORK.HBK
Dependent Variable	y
Number of Independent Variables	3
Number of Observations	75
Method	LTS Estimation

Figure 74.12 *continued*

Summary Statistics

Variable	Q1	Median	Q3	Mean	Standard Deviation	MAD
x1	0.8000	1.8000	3.1000	3.2067	3.6526	1.9274
x2	1.0000	2.2000	3.3000	5.5973	8.2391	1.6309
x3	0.9000	2.1000	3.0000	7.2307	11.7403	1.7791
y	−0.5000	0.1000	0.7000	1.2787	3.4928	0.8896

Figure 74.12 displays the model fitting information and summary statistics for the response variable and independent covariates.

Figure 74.13 LTS Profile

LTS Profile

Total Number of Observations	75
Number of Squares Minimized	57
Number of Coefficients	4
Highest Possible Breakdown Value	0.2533

Figure 74.13 displays information about the LTS fit, which includes the breakdown value of the LTS estimate. The breakdown value is a measure of the proportion of contamination that an estimation method can withstand and still maintain its robustness. In this example the LTS estimate minimizes the sum of 57 smallest squares of residuals. It can still estimate the true underlying model if the remaining 18 observations are contaminated. This corresponds to the breakdown value around 0.25, which is set as the default.

Figure 74.14 LTS Parameter Estimates

LTS Parameter Estimates

Parameter	DF	Estimate
Intercept	1	−0.3431
x1	1	0.0901
x2	1	0.0703
x3	1	−0.0731
Scale (sLTS)	0	0.7451
Scale (Wscale)	0	0.5749

Figure 74.14 displays parameter estimates for covariates and scale. Two robust estimates of the scale parameter are displayed. See the section "Final Weighted Scale Estimator" on page 5681 for how these estimates are computed. The weighted scale estimator (Wscale) is a more efficient estimator of the scale parameter.

Figure 74.15 Diagnostics

Diagnostics

Obs	index	Mahalanobis Distance	Robust MCD Distance	Leverage	Standardized Robust Residual	Outlier
1	1	1.9168	29.4424	*	17.0868	*
3	2	1.8558	30.2054	*	17.8428	*
5	3	2.3137	31.8909	*	18.3063	*
7	4	2.2297	32.8621	*	16.9702	*
9	5	2.1001	32.2778	*	17.7498	*
11	6	2.1462	30.5892	*	17.5155	*
13	7	2.0105	30.6807	*	18.8801	*
15	8	1.9193	29.7994	*	18.2253	*
17	9	2.2212	31.9537	*	17.1843	*
19	10	2.3335	30.9429	*	17.8021	*
21	11	2.4465	36.6384	*	0.0406	
23	12	3.1083	37.9552	*	−0.0874	
25	13	2.6624	36.9175	*	1.0776	
27	14	6.3816	41.0914	*	−0.7875	

Figure 74.15 displays outlier and leverage-point diagnostics. The ID variable index is used to identify the observations. If you do not specify this ID variable, the observation number is used to identify the observations. However, the observation number depends on how the data are read. The first 10 observations are identified as outliers, and observations 11 to 14 are identified as good leverage points.

Figure 74.16 Final Weighted LS Estimates

Parameter Estimates for Final Weighted Least Squares Fit

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	−0.1805	0.1044	−0.3852	0.0242	2.99	0.0840
x1	1	0.0814	0.0667	−0.0493	0.2120	1.49	0.2222
x2	1	0.0399	0.0405	−0.0394	0.1192	0.97	0.3242
x3	1	−0.0517	0.0354	−0.1210	0.0177	2.13	0.1441
Scale	0	0.5572					

Figure 74.16 displays the final weighted least squares estimates. These estimates are least squares estimates computed after deleting the detected outliers.

Syntax: ROBUSTREG Procedure

The following statements are available in PROC ROBUSTREG:

PROC ROBUSTREG < *options* > **;**
BY *variables* **;**
CLASS *variables* **;**
ID *variables* **;**
MODEL *response*=< *effects* > < / *options* > **;**
OUTPUT < *OUT=SAS-data-set* > < *options* > **;**
PERFORMANCE < *options* > **;**
TEST *effects* **;**
WEIGHT *variable* **;**

The PROC ROBUSTREG statement invokes the procedure. The METHOD= option in the PROC ROBUSTREG statement selects one of the four estimation methods, M, LTS, S, and MM. By default, Huber M estimation is used. The MODEL statement is required and specifies the variables used in the regression. Main effects and interaction terms can be specified in the MODEL statement, as in the GLM procedure (Chapter 39, "The GLM Procedure.") The CLASS statement specifies which explanatory variables are treated as categorical. These variables are allowed in the MODEL statement only for M estimation and not for other estimation methods. The ID statement names variables to identify observations in the outlier diagnostics tables. The WEIGHT statement identifies a variable in the input data set whose values are used to weight the observations. The OUTPUT statement creates an output data set containing final weights, predicted values, and residuals. The TEST statement requests robust linear tests for the model parameters. The PERFORMANCE statement tunes the performance of the procedure by using single or multiple processors available on the hardware. In one invocation of PROC ROBUSTREG, multiple OUTPUT and TEST statements are allowed.

PROC ROBUSTREG Statement

PROC ROBUSTREG < *options* > **;**

The PROC ROBUSTREG statement invokes the procedure. You can specify the following options in the PROC ROBUSTREG statement.

COVOUT
saves the estimated covariance matrix in the OUTEST= data set. This option is not supported for LTS estimation.

DATA=*SAS-data-set*
specifies the input SAS data set used by PROC ROBUSTREG. By default, the most recently created SAS data set is used.

FWLS

requests that final weighted least squares estimates be computed. These estimates are equivalent to the least squares estimates after the detected outliers are deleted.

INEST=*SAS-data-set*

specifies an input SAS data set that contains initial estimates for all the parameters in the model. See the section "INEST= Data Set" on page 5689 for a detailed description of the contents of the INEST= data set.

ITPRINT

displays the iteration history for the iteratively reweighted least squares algorithm used by M and MM estimation. You can also use this option in the MODEL statement.

NAMELEN=*n*

specifies the length of effect names in tables and output data sets to be *n* characters, where *n* is a value between 20 and 200. The default length is 20 characters.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of the classification variables (specified in the CLASS statement). This ordering determines which parameters in the model correspond to each level in the data. The following table explains how PROC ROBUSTREG interprets values of the ORDER= option.

Table 74.1 Options for Order

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	formatted value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent. For more information about sorting order, refer to the chapter titled "The SORT Procedure" in the *Base SAS Procedures Guide*.

OUTEST=*SAS-data-set*

specifies an output SAS data set containing the parameter estimates, and, if the COVOUT option is specified, the estimated covariance matrix. See the section "OUTEST= Data Set" on page 5690 for a detailed description of the contents of the OUTEST= data set.

PLOT | PLOTS <(*global-plot-options***)> <=***plot-request***>**

PLOT | PLOTS<(*global-plot-options***)> <=(***plot-request* **<...***plot-request***>)>**

specifies options that control details of the plots. If you have enabled ODS GRAPHICS but do not specify the PLOTS= option, then PROC ROBUSTREG produces the robust fit plot by default when the model includes a single continuous independent variable.

The *global-plot-options* apply to all plots generated by the ROBUSTREG procedure. The following global plot option is available:

ONLY
suppresses the default robust fit plot. Only plots specifically requested are displayed.

You can specify more than one plot request within the parentheses after PLOTS=. For a single plot request, you can omit the parentheses. The following plot requests are available.

ALL
creates all appropriate plots.

DDPLOT<(LABEL=ALL | LEVERAGE | NONE | OUTLIER)>
creates a plot of robust distance against Mahalanobis distance. See the section "Leverage Point and Outlier Detection" on page 5689 for details about robust distance. The LABEL= option specifies how the points on this plot are to be labeled, as summarized by the following table.

Table 74.2 Options for Label

Value of LABEL=	Label Method
ALL	label all points
LEVERAGE	label leverage points
NONE	no labels
OUTLIERS	label outliers

By default, the ROBUSTREG procedure labels both outliers and leverage points.

If you specify ID variables in the ID statement, the values of the first ID variable are used as labels; otherwise, observation numbers are used as labels.

FITPLOT<(NOLIMITS)>
creates a plot of robust fit against the single independent continuous variable specified in the model. You can request this plot when only a single independent continuous variable is specified in the model. Confidence limits are added on the plot by default. The NOLIMITS option suppresses these limits.

HISTOGRAM
creates a histogram for the standardized robust residuals. The histogram is superimposed with a normal density curve and a kernel density curve.

NONE
suppresses all plots.

QQPLOT
creates the normal quantile-quantile plot for the standardized robust residuals.

RDPLOT<(LABEL=ALL | LEVERAGE | NONE | OUTLIER)>
creates the plot of standardized robust residual against robust distance. See the section "Leverage Point and Outlier Detection" on page 5689 for details about robust distance. The LABEL= option specifies a label method for points on this plot. These label methods are described in Table 74.2.

If you specify ID variables in the ID statement, the values of the first ID variable are used as labels; otherwise, observation numbers are used as labels.

SEED=*number*

specifies the seed for the random number generator used to randomly select the subgroups and subsets for LTS and S estimation. By default or if you specify zero, the ROBUSTREG procedure generates a random seed.

METHOD=*method type* **<(** *options* **)>**

specifies the estimation method and *options* specify some additional options for the estimation method. PROC ROBUSTREG provides four estimation methods: M estimation, LTS estimation, S estimation, and MM estimation. The default method is M estimation.

NOTE: Since the LTS and S methods use subsampling algorithms, these methods are not suitable in an analysis with categorical independent variables specified in the CLASS statement. These methods are not suitable in an analysis with continuous independent variables that have only a few unequal values or a few unequal values within one BY group. This also applies to the initial LTS and S estimates in the MM method. In summary, if the model includes categorical independent variables or continuous independent variables with a few unequal values, the M method is recommended.

Options with METHOD=M

With METHOD=M, you can specify the following additional *options*:

ASYMPCOV=*H1* | *H2* | *H3*

specifies the type of asymptotic covariance computed for the M estimate. The three types are described in the section "Asymptotic Covariance and Confidence Intervals" on page 5677. By default, ASYMPCOV= H1.

CONVERGENCE=*criterion***<(EPS=***value***)>**

specifies a convergence criterion for the M estimate. The three criteria listed in the following table are available.

Table 74.3 Options to Specify Convergence Criteria

Type	Option
coefficient	CONVERGENCE=COEF
residual	CONVERGENCE=RESID
weight	CONVERGENCE=WEIGHT

By default, CONVERGENCE = COEF. You can specify the precision of the convergence criterion with the EPS= option. By default, EPS=1.E−8.

MAXITER=*n*

sets the maximum number of iterations during the parameter estimation. By default, MAXITER=1000.

SCALE=*scale type | value*

specifies the scale parameter or a method for estimating the scale parameter. These methods and options are summarized in the following table.

Table 74.4 Options to Specify Scale

Scale	Option	Default d
Fixed constant	SCALE=*value*	
Huber estimate	SCALE=HUBER<(D=d)>	2.5
Median estimate	SCALE=MED	
Tukey estimate	SCALE=TUKEY<(D=d)>	2.5

By default, SCALE = MED.

WF | WEIGHTFUNCTION=*function type*

specifies the weight function used for the M estimate. The ROBUSTREG procedure provides 10 weight functions, which are listed in the following table. You can specify the parameters in these functions with the A=, B=, and C= options. These functions are described in the section "M Estimation" on page 5672. The default weight function is bisquare.

Table 74.5 Options to Specify Weight Functions

Weight Function	Option	Default a, b, c
andrews	WF=ANDREWS<(C=c)>	1.339
bisquare	WF=BISQUARE<(C=c)>	4.685
cauchy	WF=CAUCHY<(C=c)>	2.385
fair	WF=FAIR<(C=c)>	1.4
hampel	WF=HAMPEL<(<A=a> <B=b> <C=c>)>	2, 4, 8
huber	WF=HUBER<(C=c)>	1.345
logistic	WF=LOGISTIC<(C=c)>	1.205
median	WF=MEDIAN<(C=c)>	0.01
talworth	WF=TALWORTH<(C=c)>	2.795
welsch	WF=WELSCH<(C=c)>	2.985

Options with METHOD=LTS

With METHOD=LTS, you can specify the following additional *options*:

CSTEP=*n*

specifies the number of C-steps for the LTS estimate. See the section "LTS Estimate" on page 5679 for how the default value is determined.

IADJUST=ALL | NONE

requests (IADJUST=ALL) or suppresses (IADJUST=NONE) the intercept adjustment for all estimates in the LTS algorithm. By default, the intercept adjustment is used for data sets with less than 10000 observations. See the section "Algorithm" on page 5680 for details.

H=*n*

specifies the quantile for the LTS estimate. See the section "LTS Estimate" on page 5679 for how the default value is determined.

NBEST=*n*

specifies the number of best solutions kept for each subgroup during the computation of the LTS estimate. The default number is 10, which is the maximum number allowed.

NREP=*n*

specifies the number of repeats of least squares fit in subgroups during the computation of the LTS estimate. See the section "LTS Estimate" on page 5679 for how the default number is determined.

SUBANALYSIS

requests a display of the subgrouping information and parameter estimates within subgroups. This option generates the following ODS tables.

Table 74.6 ODS Tables Available with SUBANALYSIS Option

ODS Table Name	Description
BestEstimates	Best final estimates for LTS
BestSubEstimates	Best estimates for each subgroup
CStep	C-step information for LTS
Groups	Grouping information for LTS

SUBGROUPSIZE=*n*

specifies the data set size of the subgroups in the computation of the LTS estimate. The default number is 300.

Options with METHOD=S

With METHOD=S, you can specify the following additional *options*:

ASYMPCOV=*H1* | *H2* | *H3* | *H4*

specifies the type of asymptotic covariance computed for the S estimate. The four types are described in the section "Asymptotic Covariance and Confidence Intervals" on page 5684. By default, ASYMPCOV= H4.

CHIF= TUKEY | YOHAI

specifies the χ function for the S estimate. PROC ROBUSTREG provides two χ functions, Tukey's bisquare function and Yohai's optimal function, which you can request with CHIF=TUKEY and CHIF=YOHAI, respectively. The default is Tukey's bisquare function.

EFF=*value*

specifies the efficiency (as a fraction) for the S estimate. The parameter k_0 in the χ function is determined by this efficiency. The default efficiency is determined such that

the consistent S estimate has the breakdown value of 25%. This option is overwritten by the K0= option if both of them are used.

K0=*value*
specifies the $k0$ parameter in the χ function of the S estimate. For CHIF=TUKEY, the default is 1.548. For CHIF=YOHAI, the default is 0.66. These default values correspond to a 50% breakdown value of the consistent S estimate.

MAXITER=*n*
sets the maximum number of iterations for computing the scale parameter of the S estimate. By default, MAXITER=1000.

NREP=*n*
specifies the number of repeats of subsampling in the computation of the S estimate. See the section "Algorithm" on page 5682 for how the default number of repeats is determined.

NOREFINE
suppresses the refinement for the S estimate. See the section "Algorithm" on page 5682 for details.

SUBSETSIZE=*n*
specifies the size of the subset for the S estimate. See the section "Algorithm" on page 5682 for how its default value is determined.

TOLERANCE=*value*
specifies the tolerance for the S estimate of the scale. The default value is 0.001.

Options with METHOD=MM

With METHOD=MM, you can specify the following additional *options*:

ASYMPCOV=*H1* | *H2* | *H3* | *H4*
specifies the type of asymptotic covariance computed for the MM estimate. The four types are described in the section "Details: ROBUSTREG Procedure" on page 5672. By default, ASYMPCOV= H4.

BIASTEST< (ALPHA=*number***) >**
requests the bias test for the final MM estimate. See the section "Bias Test" on page 5686 for details about this test.

CHIF= TUKEY | YOHAI
selects the χ function for the MM estimate. PROC ROBUSTREG provides two χ functions: Tukey's bisquare function and Yohai's optimal function, which you can request with CHIF=TUKEY and CHIF=YOHAI, respectively. The default is Tukey's bisquare function. This χ function is also used by the initial S estimate if you specify the INITEST=S option.

CONVERGENCE=*criterion***<(EPS=***number***)>**

specifies a convergence criterion for the MM estimate. The three criteria listed in the following table are available.

Table 74.7 Options to Specify Convergence Criteria

Type	Option
coefficient	CONVERGENCE=COEF
residual	CONVERGENCE=RESID
weight	CONVERGENCE=WEIGHT

By default, CONVERGENCE = COEF. You can specify the precision of the convergence criterion with the EPS= option. By default, EPS=1.E−8.

EFF=*value*

specifies the efficiency (as a fraction) for the MM estimate. The parameter k_1 in the χ function is determined by this efficiency. The default efficiency is set to 0.85, which corresponds to $k_1 = 3.440$ for CHIF=TUKEY or $k_1 = 0.868$ for CHIF=YOHAI.

INITH=*n*

specifies the integer h for the initial LTS estimate used by the MM estimator. See the section "Algorithm" on page 5686 for how to specify h and how the default is determined.

INITEST=LTS | S

specifies the initial estimator for the MM estimator. By default, the LTS estimator with its default settings is used as the initial estimator for the MM estimator.

K0=*number*

specifies the parameter k_0 in the χ function for the MM estimate. For CHIF=TUKEY, the default is $k_0 = 2.9366$. For CHIF=YOHAI, the default is $k_0 = 0.7405$. These default values correspond to the 25% breakdown value of the MM estimator.

MAXITER=*n*

sets the maximum number of iterations during the parameter estimation. By default, MAXITER=1000.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC ROBUSTREG to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the ROBUSTREG procedure. The NOTSORTED option does not mean that the data are unsorted, but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

Explanatory variables that are classification variables rather than quantitative numeric variables must be listed in the CLASS statement. For each explanatory variable listed in the CLASS statement, indicator variables are generated for the levels assumed by the CLASS variable. If the CLASS statement is used, it must appear before the MODEL statement.

ID Statement

ID *variables* **;**

When the diagnostics table is requested with the DIAGNOSTICS option in the MODEL statement, the variables listed in the ID statement are displayed in addition to the observation number. These variables can be used to identify each observation. If the ID statement is omitted, the observation number is used to identify the observations.

MODEL Statement

<label:> **MODEL** *response = < effects > < / options >* **;**

Main effects and interaction terms can be specified in the MODEL statement, as in the GLM procedure (Chapter 39, "The GLM Procedure.") Classification variables are not allowed in the MODEL statement when you specify MM estimation or LTS estimation by using the METHOD= option in the PROC statement.

The optional *label*, which must be a valid SAS name, is used to label the model in the OUTEST data set.

You can specify the following options for the model fit.

ALPHA=*value*
specifies the significance level for the confidence intervals for regression parameters. The value must be between 0 and 1. By default, ALPHA=0.05.

CORRB
produces the estimated correlation matrix of the parameter estimates.

COVB
produces the estimated covariance matrix of the parameter estimates.

CUTOFF=*value*
specifies the multiplier of the cutoff value for outlier detection. By default, CUTOFF=3.

DIAGNOSTICS<(ALL)>
requests the outlier diagnostics. By default, only observations identified as outliers or leverage points are displayed. To request that all observations be displayed, specify the ALL option.

ITPRINT
displays the iteration history for the iteratively reweighted least squares algorithm used by M and MM estimation. You can also use this option in the PROC statement.

LEVERAGE<(<CUTOFF=*value* | **CUTOFFALPHA=***value***> <QUANTILE=***n***>)>**

requests an analysis of leverage points for the continuous covariates. The results are added to the diagnostics table, which you can request with the DIAGNOSTICS option in the MODEL statement. You can specify the cutoff value for leverage-point detection with the CUTOFF= option. The default cutoff value is $\sqrt{\chi^2_{p;1-\alpha}}$, where α can be specified with the CUTOFFALPHA= option. By default, $\alpha = 0.025$. You can use the QUANTILE= option to specify the quantile to be minimized for the MCD algorithm used for the leverage-point analysis. By default, QUANTILE=$[(3n + p + 1)/4]$, where n is the number of observations and p is the number of independent variables. The LEVERAGE option is ignored if the model includes classification variables as covariates.

Since the MCD algorithm uses subsampling, it is not suitable to apply the leverage-point analysis to continuous variables that have only a few nonzero values or a few nonzero values within one BY group.

NOGOODFIT
suppresses the computation of goodness-of-fit statistics.

NOINT
specifies no-intercept regression.

SINGULAR=*value*
specifies the tolerance for testing singularity of the information matrix and the crossproducts matrix for the initial least squares estimates. Roughly, the test requires that a pivot be at least this value times the original diagonal value. By default, SINGULAR=1.E−12.

OUTPUT Statement

OUTPUT < *OUT=SAS-data-set* > *keyword=name* < . . . *keyword=name* > **;**

The OUTPUT statement creates an output SAS data set containing statistics calculated after fitting the model. At least one specification of the form *keyword=name* is required.

All variables in the original data set are included in the new data set, along with the variables created with *keyword* options in the OUTPUT statement. These new variables contain fitted values and estimated quantiles. If you want to create a permanent SAS data set, you must specify a two-level name (refer to *SAS Language Reference: Concepts* for more information about permanent SAS data sets).

The following specifications can appear in the OUTPUT statement:

OUT=*SAS-data-set* specifies the new data set. By default, the procedure uses the DATA*n* convention to name the new data set.

keyword=name specifies the statistics to include in the output data set and gives names to the new variables. Specify a keyword for each desired statistic (see the following list), an equal sign, and the variable to contain the statistic.

The keywords allowed and the statistics they represent are as follows:

LEVERAGE specifies a variable to indicate leverage points. To include this variable in the OUTPUT data set, you must specify the LEVERAGE option in the PROC statement. See the section "Leverage Point and Outlier Detection" on page 5689 for how to define LEVERAGE.

OUTLIER specifies a variable to indicate outliers. See the section "Leverage Point and Outlier Detection" on page 5689 for how to define OUTLIER.

PREDICTED | P specifies a variable to contain the estimated response.

RESIDUAL | R specifies a variable to contain the unstandardized residuals

$$y_i - \mathbf{x}_i^T \hat{\theta}$$

SRESIDUAL | SR specifies a variable to contain the standardized residuals

$$\frac{y_i - \mathbf{x}_i^T \hat{\theta}}{\hat{\sigma}}$$

STDP specifies a variable to contain the estimates of the standard errors of the estimated response.

WEIGHT specifies a variable to contain the computed final weights.

PERFORMANCE Statement

The PERFORMANCE statement is used to change default options that affect the performance of PROC ROBUSTREG and to request tables that show the performance options in effect and timing details. See Chen (2002) for some empirical results.

PERFORMANCE < *options* > **;**

The following options are available:

CPUCOUNT=*1-1024*

CPUCOUNT=*ACTUAL*

specifies the number of processors to use for forming crossproduct matrices. CPUCOUNT=ACTUAL sets CPUCOUNT to be the number of physical processors available. Note that this can be less than the physical number of CPUs if the SAS process has been restricted by system administration tools. Setting CPUCOUNT= to a number greater than the actual number of available CPUs might result in reduced performance. This option overrides the SAS system option CPUCOUNT=. If CPUCOUNT=1, then NOTHREADS is in effect, and PROC ROBUSTREG uses singly threaded code.

DETAILS

requests the "PerfSettings" table that shows the performance settings in effect and the "Timing" table that provides a broad timing breakdown of the PROC ROBUSTREG step.

THREADS

enables multithreaded computation. This option overrides the SAS system option THREADS | NOTHREADS.

NOTHREADS

disables multithreaded computation. This option overrides the SAS system option THREADS | NOTHREADS.

TEST Statement

<*label:*> **TEST** *effects* **;**

With M estimation and MM estimation, the TEST statement provides a means of obtaining a test for the canonical linear hypothesis concerning the parameters of the tested effects:

$$\theta_j = 0, \quad j = i_1, \ldots, i_q$$

where q is the total number of parameters of the tested effects.

PROC ROBUSTREG provides two kinds of robust tests: the ρ test and the R_n^2 test. They are described in the section "Details: ROBUSTREG Procedure" on page 5672. No test is available for LTS and S estimation.

The optional *label*, which must be a valid SAS name, is used to label output from the corresponding TEST statement.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement specifies a weight variable in the input data set.

If you want to use fixed weights for each observation in the input data set, place the weights in a variable in the data set and specify the name in a WEIGHT statement. The values of the WEIGHT variable can be nonintegral and are not truncated. Observations with nonpositive or missing values for the weight variable do not contribute to the fit of the model.

Details: ROBUSTREG Procedure

This section describes the statistical and computational aspects of the ROBUSTREG procedure. The following notation is used throughout this section.

Let $X = (x_{ij})$ denote an $n \times p$ matrix, $y = (y_1, \ldots, y_n)^T$ denote a given n-vector of responses, and $\theta = (\theta_1, \ldots, \theta_p)^T$ denote an unknown p-vector of parameters or coefficients whose components are to be estimated. The matrix X is called the design matrix. Consider the usual linear model

$$y = X\theta + e$$

where $e = (e_1, \ldots, e_n)^T$ is an n-vector of unknown errors. It is assumed that (for a given X) the components e_i of e are independent and identically distributed according to a distribution $L(\cdot/\sigma)$, where σ is a scale parameter (usually unknown). Often $L(\cdot) \approx \Phi(\cdot)$, the standard normal distribution function. The vector of residuals for a given value of $\hat{\theta}$ is denoted by $r = (r_1, \ldots, r_n)^T$ and the ith row of the matrix X is denoted by x_i^T.

M Estimation

M estimation in the context of regression was first introduced by Huber (1973) as a result of making the least squares approach robust. Although M estimators are not robust with respect to leverage points, they are popular in applications where leverage points are not an issue.

Instead of minimizing a sum of squares of the residuals, a Huber-type M estimator $\hat{\theta}_M$ of θ minimizes a sum of less rapidly increasing functions of the residuals:

$$Q(\theta) = \sum_{i=1}^{n} \rho(\frac{r_i}{\sigma})$$

where $r = y - X\theta$. For the ordinary least squares estimation, ρ is the square function, $\rho(z) = z^2$.

If σ is known, then by taking derivatives with respect to θ, $\hat{\theta}_{\boldsymbol{M}}$ is also a solution of the system of p equations:

$$\sum_{i=1}^{n} \psi(\frac{r_i}{\sigma}) x_{ij} = 0, \ j = 1, \ldots, p$$

where $\psi = \frac{\partial \rho}{\partial z}$. If ρ is convex, $\hat{\theta}_{\boldsymbol{M}}$ is the unique solution.

The ROBUSTREG procedure solves this system by using iteratively reweighted least squares (IRLS). The weight function $w(x)$ is defined as

$$w(z) = \frac{\psi(z)}{z}$$

The ROBUSTREG procedure provides 10 kinds of weight functions through the WEIGHTFUNCTION= option in the MODEL statement. Each weight function corresponds to a ρ function. See the section "Weight Functions" on page 5674 for a complete discussion. You can specify the scale parameter σ with the SCALE= option in the PROC statement.

If σ is unknown, both θ and σ are estimated by minimizing the function

$$Q(\theta, \sigma) = \sum_{i=1}^{n} [\rho(\frac{r_i}{\sigma}) + a]\sigma, \ a > 0$$

The algorithm proceeds by alternately improving $\hat{\theta}$ in a location step and $\hat{\sigma}$ in a scale step.

For the scale step, three methods are available to estimate σ, which you can select with the SCALE= option.

1. (SCALE=HUBER<(D=d)>) Compute $\hat{\sigma}$ by the iteration

$$(\hat{\sigma}^{(m+1)})^2 = \frac{1}{nh} \sum_{i=1}^{n} \chi_d(\frac{r_i}{\hat{\sigma}^{(m)}})(\hat{\sigma}^{(m)})^2$$

where

$$\chi_d(x) = \begin{cases} x^2/2 & \text{if } |x| < d \\ d^2/2 & \text{otherwise} \end{cases}$$

is the Huber function and $h = \frac{n-p}{n}(d^2 + (1-d^2)\Phi(d) - 0.5 - d\sqrt{2\pi}e^{-\frac{1}{2}d^2})$ is the Huber constant (refer to Huber 1981, p. 179). You can specify d with the D= option. By default, $d = 2.5$.

2. (SCALE=TUKEY<(D=d)>) Compute $\hat{\sigma}$ by solving the supplementary equation

$$\frac{1}{n-p}\sum_{i=1}^{n}\chi_d(\frac{r_i}{\sigma}) = \beta$$

where

$$\chi_d(x) = \begin{cases} \frac{3x^2}{d^2} - \frac{3x^4}{d^4} + \frac{x^6}{d^6} & \text{if } |x| < d \\ 1 & \text{otherwise} \end{cases}$$

Here $\psi = \frac{1}{6}\chi_1'$ is Tukey's bisquare function, and $\beta = \int \chi_d(s)d\Phi(s)$ is the constant such that the solution $\hat{\sigma}$ is asymptotically consistent when $L(\cdot/\sigma) = \Phi(\cdot)$ (refer to Hampel et al. 1986, p. 149). You can specify d with the D= option. By default, $d = 2.5$.

3. (SCALE=MED) Compute $\hat{\sigma}$ by the iteration

$$\hat{\sigma}^{(m+1)} = \text{median}\{|y_i - x_i^T\hat{\theta}^{(m)}|/\beta_0, i = 1, \ldots, n\}$$

where $\beta_0 = \Phi^{-1}(.75)$ is the constant such that the solution $\hat{\sigma}$ is asymptotically consistent when $L(\cdot/\sigma) = \Phi(\cdot)$ (refer to Hampel et al. 1986, p. 312).

Note that SCALE = MED is the default.

Algorithm

The basic algorithm for computing M estimates for regression is iteratively reweighted least squares (IRLS). As the name suggests, a weighted least squares fit is carried out inside an iteration loop. For each iteration, a set of weights for the observations is used in the least squares fit. The weights are constructed by applying a weight function to the current residuals. Initial weights are based on residuals from an initial fit. The ROBUSTREG procedure uses the unweighted least squares fit as a default initial fit. The iteration terminates when a convergence criterion is satisfied. The maximum number of iterations is set to 1000. You can specify the weight function and the convergence criteria.

Weight Functions

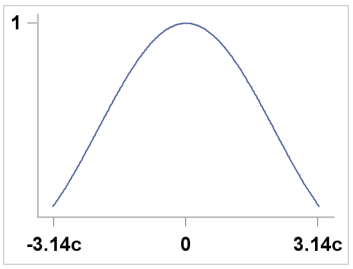
You can specify the weight function for M estimation with the WEIGHTFUNCTION= option. The ROBUSTREG procedure provides 10 weight functions. By default, the procedure uses the bisquare

weight function. In most cases, M estimates are more sensitive to the parameters of these weight functions than to the type of the weight function. The median weight function is not stable and is seldom recommended in data analysis; it is included in the procedure for completeness. You can specify the parameters for these weight functions. Except for the hampel and median weight functions, default values for these parameters are defined such that the corresponding M estimates have 95% asymptotic efficiency in the location model with the Gaussian distribution (see Holland and Welsch 1977).

The following list shows the weight functions available. See Table 74.5 for the default values of the constants in these weight functions.

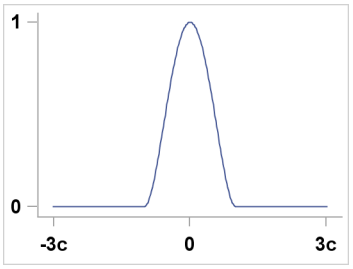
andrews

$$W(x,c) = \begin{cases} \frac{\sin(\frac{x}{c})}{\frac{x}{c}} & \text{if } |x| \leq \pi c \\ 0 & \text{otherwise} \end{cases}$$



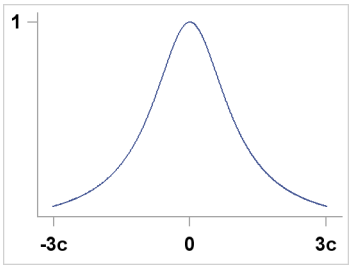
bisquare

$$W(x,c) = \begin{cases} (1-(\frac{x}{c})^2)^2 & \text{if } |x| < c \\ 0 & \text{otherwise} \end{cases}$$



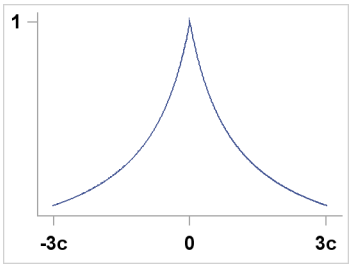
cauchy

$$W(x,c) = \frac{1}{1+(\frac{|x|}{c})^2}$$

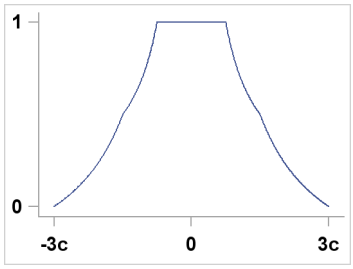


fair

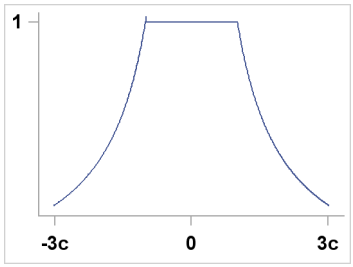
$$W(x,c) = \frac{1}{(1+\frac{|x|}{c})}$$



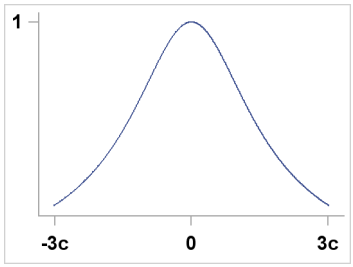
hampel $$W(x,a,b,c) = \begin{cases} 1 & |x| < a \\ \frac{a}{|x|} & a < |x| \le b \\ \frac{a}{|x|}\frac{c-|x|}{c-b} & b < |x| \le c \\ 0 & \text{otherwise} \end{cases}$$



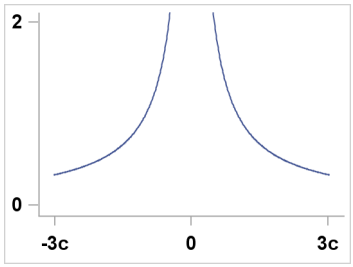
huber $$W(x,c) = \begin{cases} 1 & \text{if } |x| < c \\ \frac{c}{|x|} & \text{otherwise} \end{cases}$$



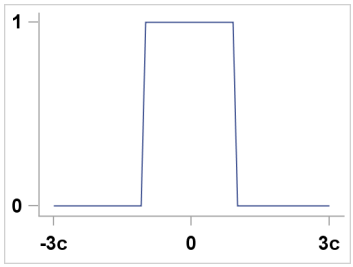
logistic $$W(x,c) = \frac{\tanh(\frac{x}{c})}{\frac{x}{c}}$$



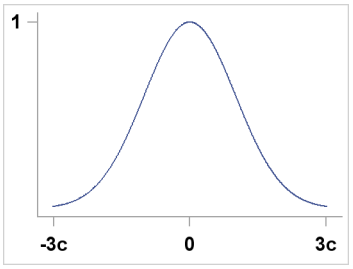
median $$W(x,c) = \begin{cases} \frac{1}{c} & \text{if } x = 0 \\ \frac{1}{|x|} & \text{otherwise} \end{cases}$$



talworth $$W(x,c) = \begin{cases} 1 & \text{if } |x| < c \\ 0 & \text{otherwise} \end{cases}$$



welsch $W(x,c) = \exp(-\frac{1}{2}(\frac{x}{c})^2)$



Convergence Criteria

The following convergence criteria are available in PROC ROBUSTREG:

- relative change in the coefficients (CONVERGENCE= COEF)
- relative change in the scaled residuals (CONVERGENCE= RESID)
- relative change in weights (CONVERGENCE= WEIGHT)

You can specify the criteria with the CONVERGENCE= option in the PROC statement. The default is CONVERGENCE= COEF.

You can specify the precision of the convergence criterion with the EPS= suboption. The default is EPS=1.E−8.

In addition to these convergence criteria, a convergence criterion based on scale-independent measure of the gradient is always checked. See Coleman et al. (1980) for more details. A warning is issued if this criterion is not satisfied.

Asymptotic Covariance and Confidence Intervals

The following three estimators of the asymptotic covariance of the robust estimator are available in PROC ROBUSTREG:

H1: $K^2 \dfrac{[1/(n-p)]\sum(\psi(r_i))^2}{[(1/n)\sum(\psi'(r_i))]^2}(X^TX)^{-1}$

H2: $K \dfrac{[1/(n-p)]\sum(\psi(r_i))^2}{[(1/n)\sum(\psi'(r_i))]}W^{-1}$

H3: $K^{-1}\dfrac{1}{(n-p)}\sum(\psi(r_i))^2 W^{-1}(X^TX)W^{-1}$

where $K = 1 + \frac{p}{n}\frac{\mathrm{Var}(\psi')}{(E\psi')^2}$ is a correction factor and $W_{jk} = \sum \psi'(r_i)x_{ij}x_{ik}$. Refer to Huber (1981, p. 173) for more details.

You can specify the asymptotic covariance estimate with the option ASYMPCOV= option. The ROBUSTREG procedure uses H1 as the default because of its simplicity and stability. Confidence intervals are computed from the diagonal elements of the estimated asymptotic covariance matrix.

R Square and Deviance

The robust version of R^2 is defined as

$$R^2 = \frac{\sum \rho(\frac{y_i-\hat{\mu}}{\hat{s}}) - \sum \rho(\frac{y_i-x_i^T\hat{\theta}}{\hat{s}})}{\sum \rho(\frac{y_i-\hat{\mu}}{\hat{s}})}$$

and the robust deviance is defined as the optimal value of the objective function on the σ^2 scale:

$$D = 2(\hat{s})^2 \sum \rho(\frac{y_i - x_i^T\hat{\theta}}{\hat{s}})$$

where $\rho' = \psi$, $\hat{\theta}$ is the M estimator of θ, $\hat{\mu}$ is the M estimator of location, and $\hat{s}$ is the M estimator of the scale parameter in the full model.

Linear Tests

Two tests are available in PROC ROBUSTREG for the canonical linear hypothesis

$$H_0: \quad \theta_j = 0, \quad j = i_1, \ldots, i_q$$

where q is the total number of parameters of the tested effects. The first test is a robust version of the F test, which is referred to as the ρ test. Denote the M estimators in the full and reduced model as $\hat{\theta}(0) \in \Omega_0$ and $\hat{\theta}(1) \in \Omega_1$, respectively. Let

$$\begin{aligned} Q_0 &= Q(\hat{\theta}(0)) = \min\{Q(\theta)|\theta \in \Omega_0\} \\ Q_1 &= Q(\hat{\theta}(1)) = \min\{Q(\theta)|\theta \in \Omega_1\} \end{aligned}$$

with

$$Q = \sum_{i=1}^{n} \rho(\frac{r_i}{\sigma})$$

The robust F test is based on the test statistic

$$S_n^2 = \frac{2}{q}[Q_1 - Q_0]$$

Asymptotically $S_n^2 \sim \lambda\chi_q^2$ under H_0, where the standardization factor is $\lambda = \int \psi^2(s)d\Phi(s) / \int \psi'(s)\ d\Phi(s)$ and Φ is the cumulative distribution function of the standard normal distribution. Large values of S_n^2 are significant. This test is a special case of the general τ test of Hampel et al. (1986, Section 7.2).

The second test is a robust version of the Wald test, which is referred to as R_n^2 test. The test uses a test statistic

$$R_n^2 = n(\hat{\theta}_{i_1}, \ldots, \hat{\theta}_{i_q}) H_{22}^{-1} (\hat{\theta}_{i_1}, \ldots, \hat{\theta}_{i_q})^T$$

where $\frac{1}{n}H_{22}$ is the $q \times q$ block (corresponding to $\theta_{i_1}, \ldots, \theta_{i_q}$) of the asymptotic covariance matrix of the M estimate $\hat{\theta}_M$ of θ in a p-parameter linear model.

Under H_0, the statistic R_n^2 has an asymptotic χ^2 distribution with q degrees of freedom. Large values of R_n^2 are significant. Refer to Hampel et al. (1986, Chapter 7) for more details.

Model Selection

When M estimation is used, two criteria are available in PROC ROBUSTREG for model selection. The first criterion is a counterpart of the Akaike (1974) AIC criterion for robust regression, and it is defined as

$$AICR = 2\sum_{i=1}^{n}\rho(r_{i:p}) + \alpha p$$

where $r_{i:p} = (y_i - x_i^T\hat{\theta})/\hat{\sigma}$, $\hat{\sigma}$ is a robust estimate of σ and $\hat{\theta}$ is the M estimator with p-dimensional design matrix.

As with AIC, α is the weight of the penalty for dimensions. The ROBUSTREG procedure uses $\alpha = 2E\psi^2/E\psi'$ (Ronchetti 1985) and estimates it by using the final robust residuals.

The second criterion is a robust version of the Schwarz information criteria (BIC), and it is defined as

$$BICR = 2\sum_{i=1}^{n}\rho(r_{i:p}) + p\log(n)$$

High Breakdown Value Estimation

The *breakdown value* of an estimator is defined as the smallest fraction of contamination that can cause the estimator to take on values arbitrarily far from its value on the uncontamined data. The breakdown value of an estimator can be used as a measure of the robustness of the estimator. Rousseeuw and Leroy (1987) and others introduced the following high breakdown value estimators for linear regression.

LTS Estimate

The least trimmed squares (LTS) estimate proposed by Rousseeuw (1984) is defined as the p-vector

$$\hat{\theta}_{LTS} = \arg\min_{\theta} Q_{LTS}(\theta)$$

where

$$Q_{LTS}(\theta) = \sum_{i=1}^{h} r_{(i)}^2$$

$r_{(1)}^2 \le r_{(2)}^2 \le \ldots \le r_{(n)}^2$ are the ordered squared residuals $r_i^2 = (y_i - x_i^T\theta)^2$, $i = 1, \ldots, n$, and h is defined in the range $\frac{n}{2} + 1 \le h \le \frac{3n+p+1}{4}$.

You can specify the parameter h with the H= option in the PROC statement. By default, $h = [\frac{3n+p+1}{4}]$. The breakdown value is $\frac{n-h}{n}$ for the LTS estimate.

The ROBUSTREG procedure computes LTS estimates by using the FAST-LTS algorithm of Rousseeuw and Van Driessen (2000). The estimates are often used to detect outliers in the data, which are then downweighted in the resulting weighted LS regression.

Algorithm

Least trimmed squares (LTS) regression is based on the subset of h observations (out of a total of n observations) whose least squares fit possesses the smallest sum of squared residuals. The coverage h can be set between $\frac{n}{2}$ and n. The LTS method was proposed by Rousseeuw (1984, p. 876) as a highly robust regression estimator with breakdown value $\frac{n-h}{n}$. The ROBUSTREG procedure uses the FAST-LTS algorithm given by Rousseeuw and Van Driessen (2000). The intercept adjustment technique is also used in this implementation. However, because this adjustment is expensive to compute, it is optional. You can use the IADJUST option in the PROC statement to request or suppress the intercept adjustment. By default, PROC ROBUSTREG does intercept adjustment for data sets with fewer than 10000 observations. The steps of the algorithm are described briefly as follows. Refer to Rousseeuw and Van Driessen (2000) for details.

1. The default h is $[\frac{3n+p+1}{4}]$, where p is the number of independent variables. You can specify any integer h with $[\frac{n}{2}] + 1 \leq h \leq [\frac{3n+p+1}{4}]$ with the H= option in the MODEL statement. The breakdown value for LTS, $\frac{n-h}{n}$, is reported. The default h is a good compromise between breakdown value and statistical efficiency.

2. If $p = 1$ (single regressor), the procedure uses the exact algorithm of Rousseeuw and Leroy (1987, p. 172).

3. If $p \geq 2$, the procedure uses the following algorithm. If $n < 2ssubs$, where *ssubs* is the size of the subgroups (you can specify *ssubs* by using the SUBGROUPSIZE= option in the PROC statement; by default, $ssubs = 300$), draw a random p-subset and compute the regression coefficients by using these p points (if the regression is degenerate, draw another p-subset). Compute the absolute residuals for all observations in the data set, and select the first h points with smallest absolute residuals. From this selected h-subset, carry out *nsteps* C-steps (Concentration step; see Rousseeuw and Van Driessen (2000) for details. You can specify *nsteps* with the CSTEP= option in the PROC statement; by default, $nsteps = 2$). Redraw p-subsets and repeat the preceding computing procedure *nrep* times, and then find the *nbsol* (at most) solutions with the lowest sums of h squared residuals. *nrep* can be specified with the NREP= option in the PROC statement. By default, NREP=min$\{500, \binom{n}{p}\}$. For small n and p, all $\binom{n}{p}$ subsets are used and the NREP= option is ignored (Rousseeuw and Hubert 1996). *nbsol* can be specified with the NBEST= option in the PROC statement. By default, NBEST=10. For each of these *nbsol* best solutions, take C-steps until convergence and find the best final solution.

4. If $n \geq 5ssubs$, construct 5 disjoint random subgroups with size *ssubs*. If $2ssubs < n < 5ssubs$, the data are split into at most four subgroups with *ssubs* or more observations in each subgroup, so that each observation belongs to a subgroup and the subgroups have roughly the same size. Let *nsubs* denote the number of subgroups. Inside each subgroup, repeat the procedure in step 3 $[\frac{nrep}{nsubs}]$ times and keep the *nbsol* best solutions. Pool the subgroups, yielding the merged set of size n_{merged}. In the merged set, for each of the $nsubs \times nbsol$ best solutions, carry out *nsteps* C-steps by using n_{merged} and $h_{merged} = [n_{merged}\frac{h}{n}]$ and keep the *nbsol* best solutions. In the full data set, for each of these *nbsol* best solutions, take C-steps by using n and h until convergence and find the best final solution.

R^2

The robust version of R^2 for the LTS estimate is defined as

$$R^2_{LTS} = 1 - \frac{s^2_{LTS}(X, y)}{s^2_{LTS}(\mathbf{1}, y)}$$

for models with the intercept term and as

$$R^2_{LTS} = 1 - \frac{s^2_{LTS}(X, y)}{s^2_{LTS}(\mathbf{0}, y)}$$

for models without the intercept term, where

$$s_{LTS}(X, y) = d_{h,n}\sqrt{\frac{1}{h}\sum_{i=1}^{h} r^2_{(i)}}$$

s_{LTS} is a preliminary estimate of the parameter σ in the distribution function $L(\cdot/\sigma)$.

Here $d_{h,n}$ is chosen to make s_{LTS} consistent, assuming a Gaussian model. Specifically,

$$\begin{aligned} d_{h,n} &= 1/\sqrt{1 - \frac{2n}{hc_{h,n}}\phi(1/c_{h,n})} \\ c_{h,n} &= 1/\Phi^{-1}(\frac{h+n}{2n}) \end{aligned}$$

with Φ and ϕ being the distribution function and the density function of the standard normal distribution, respectively.

Final Weighted Scale Estimator

The ROBUSTREG procedure displays two scale estimators, s_{LTS} and Wscale. The estimator Wscale is a more efficient scale estimator based on the preliminary estimate s_{LTS}, and it is defined as

$$\text{Wscale} = \sqrt{\frac{\sum_i w_i r_i^2}{\sum_i w_i - p}}$$

where

$$w_i = \begin{cases} 0 & \text{if } |r_i|/s_{LTS} > k \\ 1 & \text{otherwise} \end{cases}$$

You can specify k with the CUTOFF= option in the MODEL statement. By default, $k = 3$.

S Estimate

The S estimate proposed by Rousseeuw and Yohai (1984) is defined as the p-vector

$$\hat{\theta}_S = \arg\min_{\theta} S(\theta)$$

where the dispersion $S(\theta)$ is the solution of

$$\frac{1}{n-p}\sum_{i=1}^{n}\chi\left(\frac{y_i - x_i^T\theta}{S}\right) = \beta$$

Here β is set to $\int \chi(s)d\Phi(s)$ such that $\hat{\theta}_S$ and $S(\hat{\theta}_S)$ are asymptotically consistent estimates of θ and σ for the Gaussian regression model. The breakdown value of the S estimate is

$$\frac{\beta}{\max_s \chi(s)}$$

The ROBUSTREG procedure provides two choices for χ: Tukey's bisquare function and Yohai's optimal function.

Tukey's bisquare function, which you can specify with the option CHIF=TUKEY, is

$$\chi_{k_0}(s) = \begin{cases} 3(\frac{s}{k_0})^2 - 3(\frac{s}{k_0})^4 + (\frac{s}{k_0})^6, & \text{if } |s| \le k_0 \\ 1 & \text{otherwise} \end{cases}$$

The constant k_0 controls the breakdown value and efficiency of the S estimate. If you specify the efficiency by using the EFF= option, you can determine the corresponding k_0. The default k_0 is 2.9366 such that the breakdown value of the S estimate is 0.25 with a corresponding asymptotic efficiency for the Gaussian model of 75.9%.

The Yohai function, which you can specify with the option CHIF=YOHAI, is

$$\chi_{k_0}(s) = \begin{cases} \frac{s^2}{2} & \text{if } |s| \le 2k_0 \\ k_0^2[b_0 + b_1(\frac{s}{k_0})^2 + b_2(\frac{s}{k_0})^4 & \\ \quad +b_3(\frac{s}{k_0})^6 + b_4(\frac{s}{k_0})^8] & \text{if } 2k_0 < |s| \le 3k_0 \\ 3.25k_0^2 & \text{if } |s| > 3k_0 \end{cases}$$

where $b_0 = 1.792$, $b_1 = -0.972$, $b_2 = 0.432$, $b_3 = -0.052$, and $b_4 = 0.002$. If you specify the efficiency by using the EFF= option, you can determine the corresponding k_0. By default, k_0 is set to 0.7405 such that the breakdown value of the S estimate is 0.25 with a corresponding asymptotic efficiency for the Gaussian model of 72.7%.

Algorithm

The ROBUSTREG procedure implements the algorithm by Marazzi (1993) for the S estimate, which is a refined version of the algorithm proposed by Ruppert (1992). The refined algorithm is briefly described as follows.

Initialize iter $= 1$.

1. Draw a random q-subset of the total n observations and compute the regression coefficients by using these q observations (if the regression is degenerate, draw another q-subset), where $q \geq p$ can be specified with the SUBSIZE= option. By default, $q = p$.

2. Compute the residuals: $r_i = y_i - \sum_{j=1}^{p} x_{ij}\theta_j$ for $i = 1, \ldots, n$. If iter $= 1$, set $s^* = 2\text{median}\{|r_i|, i = 1, \ldots, n\}$; if $s^* = 0$, set $s^* = \min\{|r_i|, i = 1, \ldots, n\}$;
while $\sum_{i=1}^{n} \chi(r_i/s^*) > (n-p)\beta$, set $s^* = 1.5s^*$; go to step 3.
If iter > 1 and $\sum_{i=1}^{n} \chi(r_i/s^*) <= (n-p)\beta$, go to step 3; otherwise, go to step 5.

3. Solve for s the equation

$$\frac{1}{n-p}\sum_{i=1}^{n}\chi(r_i/s) = \beta$$

using an iterative algorithm.

4. If iter > 1 and $s > s^*$, go to step 5. Otherwise, set $s^* = s$ and $\theta^* = \theta$. If $s^* < TOLS$, return s^* and θ^*; otherwise, go to step 5.

5. If iter $< NREP$, set iter $=$ iter $+ 1$ and return to step 1; otherwise, return s^* and θ^*.

The ROBUSTREG procedure does the following refinement step by default. You can request that this refinement not be done by using the NOREFINE option in the PROC statement.

6. Let $\psi = \chi'$. Using the values s^* and θ^* from the previous steps, compute M estimates θ_M and σ_M of θ and σ with the setup for M estimation in the section "M Estimation" on page 5672. If $\sigma_M > s^*$, give a warning and return s^* and θ^*; otherwise, return σ_M and θ_M.

You can specify $TOLS$ with the TOLERANCE= option; by default, TOLERANCE=0.001. Alternately, you can specify $NREP$ with the NREP= option. You can also use the options NREP=NREP0 or NREP=NREP1 to determine $NREP$ according to the following table. NREP=NREP0 is set as the default.

Table 74.9 Default NREP

P	NREP0	NREP1
1	150	500
2	300	1000
3	400	1500
4	500	2000
5	600	2500
6	700	3000
7	850	3000
8	1250	3000
9	1500	3000
>9	1500	3000

R^2 and Deviance

The robust version of R^2 for the S estimate is defined as

$$R_S^2 = 1 - \frac{(n-p)S_p^2}{(n-1)S_\mu^2}$$

for the model with the intercept term and

$$R_S^2 = 1 - \frac{(n-p)S_p^2}{nS_0^2}$$

for the model without the intercept term, where S_p is the S estimate of the scale in the full model, S_μ is the S estimate of the scale in the regression model with only the intercept term, and S_0 is the S estimate of the scale without any regressor. The deviance D is defined as the optimal value of the objective function on the σ^2 scale:

$$D = S_p^2$$

Asymptotic Covariance and Confidence Intervals

Since the S estimate satisfies the first-order necessary conditions as the M estimate, it has the same asymptotic covariance as that of the M estimate. All three estimators of the asymptotic covariance for the M estimate in the section "Asymptotic Covariance and Confidence Intervals" on page 5677 can be used for the S estimate. Besides, the weighted covariance estimator H4 described in the section "Asymptotic Covariance and Confidence Intervals" on page 5687 is also available and is set as the default. Confidence intervals for estimated parameters are computed from the diagonal elements of the estimated asymptotic covariance matrix.

MM Estimation

MM estimation is a combination of high breakdown value estimation and efficient estimation, which was introduced by Yohai (1987). It has the following three steps.

1. Compute an initial (consistent) high breakdown value estimate $\hat{\theta}'$. The ROBUSTREG procedure provides two kinds of estimates as the initial estimate: the LTS estimate and the S estimate. By default, the LTS estimate is used because of its speed and high breakdown value. The breakdown value of the final MM estimate is decided by the breakdown value of the initial LTS estimate and the constant k_0 in the χ function. To use the S estimate as the initial estimate, you specify the INITEST=S option in the PROC statement. In this case, the breakdown value of the final MM estimate is decided only by the constant k_0. Instead of computing the LTS estimate or the S estimate as the initial estimate, you can also specify the initial estimate explicitly by using the INEST= option in the PROC statement. See the section "INEST= Data Set" on page 5689 for details.

2. Find $\hat{\sigma}'$ such that

$$\frac{1}{n-p}\sum_{i=1}^{n}\chi(\frac{y_i - x_i^T\hat{\theta}'}{\hat{\sigma}'}) = \beta$$

where $\beta = \int \chi(s)d\Phi(s)$.

The ROBUSTREG procedure provides two choices for χ: Tukey's bisquare function and Yohai's optimal function.

Tukey's bisquare function, which you can specify with the option CHIF=TUKEY, is

$$\chi_{k_0}(s) = \begin{cases} 3(\frac{s}{k_0})^2 - 3(\frac{s}{k_0})^4 + (\frac{s}{k_0})^6, & \text{if } |s| \le k_0 \\ 1 & \text{otherwise} \end{cases}$$

where k_0 can be specified with the K0= option. The default k_0 is 2.9366 such that the asymptotically consistent scale estimate $\hat{\sigma}'$ has the breakdown value of 25%.

Yohai's optimal function, which you can specify with the option CHIF=YOHAI, is

$$\chi_{k_0}(s) = \begin{cases} \frac{s^2}{2} & \text{if } |s| \le 2k_0 \\ k_0^2[b_0 + b_1(\frac{s}{k_0})^2 + b_2(\frac{s}{k_0})^4 & \\ \quad +b_3(\frac{s}{k_0})^6 + b_4(\frac{s}{k_0})^8] & \text{if } 2k_0 < |s| \le 3k_0 \\ 3.25k_0^2 & \text{if } |s| > 3k_0 \end{cases}$$

where $b_0 = 1.792$, $b_1 = -0.972$, $b_2 = 0.432$, $b_3 = -0.052$, and $b_4 = 0.002$. You can specify k_0 with the K0= option. The default k_0 is 0.7405 such that the asymptotically consistent scale estimate $\hat{\sigma}'$ has the breakdown value of 25%.

3. Find a local minimum $\hat{\theta}_{MM}$ of

$$Q_{MM} = \sum_{i=1}^{n}\rho(\frac{y_i - x_i^T\theta}{\hat{\sigma}'})$$

such that $Q_{MM}(\hat{\theta}_{MM}) \le Q_{MM}(\hat{\theta}')$. The algorithm for M estimation is used here.

The ROBUSTREG procedure provides two choices for ρ: Tukey's bisquare function and Yohai's optimal function.

Tukey's bisquare function, which you can specify with the option CHIF=TUKEY, is

$$\rho(s) = \chi_{k_1}(s) = \begin{cases} 3(\frac{s}{k_1})^2 - 3(\frac{s}{k_1})^4 + (\frac{s}{k_1})^6, & \text{if } |s| \le k_1 \\ 1 & \text{otherwise} \end{cases}$$

where k_1 can be specified with the K1= option. The default k_1 is 3.440 such that the MM estimate has 85% asymptotic efficiency with the Gaussian distribution.

Yohai's optimal function, which you can specify with the option CHIF=YOHAI, is

$$\rho(s) = \chi_{k_1}(s) = \begin{cases} \frac{s^2}{2} & \text{if } |s| \le 2k_1 \\ k_1^2[b_0 + b_1(\frac{s}{k_1})^2 + b_2(\frac{s}{k_1})^4 & \\ \quad +b_3(\frac{s}{k_1})^6 + b_4(\frac{s}{k_1})^8] & \text{if } 2k_1 < |s| \le 3k_1 \\ 3.25k_1^2 & \text{if } |s| > 3k_1 \end{cases}$$

where k_1 can be specified with the K1= option. The default k_1 is 0.868 such that the MM estimate has 85% asymptotic efficiency with the Gaussian distribution.

Algorithm

The initial LTS estimate is computed using the algorithm described in the section "LTS Estimate" on page 5679. You can control the quantile of the LTS estimate with the option INITH=h, where h is an integer between $[\frac{n}{2}] + 1$ and $[\frac{3n+p+1}{4}]$. By default, $h = [\frac{3n+p+1}{4}]$, which corresponds to a breakdown value of around 25%.

The initial S estimate is computed using the algorithm described in the section "S Estimate" on page 5682. You can control the breakdown value and efficiency of this initial S estimate by the constant k_0, which can be specified with the K0 option.

The scale parameter σ is solved by an iterative algorithm

$$(\sigma^{(m+1)})^2 = \frac{1}{(n-p)\beta} \sum_{i=1}^{n} \chi_{k_0}(\frac{r_i}{\sigma^{(m)}})(\sigma^{(m)})^2$$

where $\beta = \int \chi_{k_0}(s) d\Phi(s)$.

Once the scale parameter is computed, the iteratively reweighted least squares (IRLS) algorithm with fixed scale parameter is used to compute the final MM estimate.

Convergence Criteria

In the iterative algorithm for the scale parameter, the relative change of the scale parameter controls the convergence.

In the iteratively reweighted least squares algorithm, the same convergence criteria for the M estimate used before are used here.

Bias Test

Although the final MM estimate inherits the high breakdown value property, its bias due to the distortion of the outliers can be high. Yohai, Stahel, and Zamar (1991) introduced a bias test. The ROBUSTREG procedure implements this test when you specify the BIASTEST option in the PROC statement. This test is based on the initial scale estimate $\hat{\sigma}'$ and the final scale estimate $\hat{\sigma}'_1$, which is the solution of

$$\frac{1}{n-p} \sum_{i=1}^{n} \chi(\frac{y_i - x_i^T \hat{\theta}_{MM}}{\hat{\sigma}'_1}) = \beta$$

Let $\psi_{k_0}(z) = \frac{\partial \chi_{k_0}(z)}{\partial z}$ and $\psi_{k_1}(z) = \frac{\partial \chi_{k_1}(z)}{\partial z}$. Compute

$$\begin{aligned} \tilde{r}_i &= (y_i - x_i^T \hat{\theta}')/\hat{\sigma}' \quad \text{for } i = 1, \ldots, n \\ v_0 &= \frac{(1/n) \sum \psi'_{k_0}(\tilde{r}_i)}{(\hat{\sigma}'_1/n) \sum \psi_{k_0}(\tilde{r}_i)\tilde{r}_i} \end{aligned}$$

$$
\begin{aligned}
p_i^{(0)} &= \frac{\psi_{k_0}(\tilde{r}_i)}{(1/n)\sum \psi'_{k_0}(\tilde{r}_i)} \quad \text{for } i = 1, \ldots, n \\
p_i^{(1)} &= \frac{\psi_{k_1}(\tilde{r}_i)}{(1/n)\sum \psi'_{k_1}(\tilde{r}_i)} \quad \text{for } i = 1, \ldots, n \\
d^2 &= \frac{1}{n}\sum (p_i^{(1)} - p_i^{(0)})^2
\end{aligned}
$$

Let

$$T = \frac{2n(\hat{\sigma}'_1 - \hat{\sigma}')}{v_0 d^2 (\hat{\sigma}')^2}$$

Standard asymptotic theory shows that T approximately follows a χ^2 distribution with p degrees of freedom. If T exceeds the α quantile χ^2_α of the χ^2 distribution with p degrees of freedom, then the ROBUSTREG procedure gives a warning and recommends that you use other methods. Otherwise the final MM estimate and the initial scale estimate are reported. You can specify α with the ALPHA= option following the BIASTEST option. By default, ALPHA=0.99.

Asymptotic Covariance and Confidence Intervals

Since the MM estimate is computed as a M estimate with a fixed scale in the last step, the asymptotic covariance for the M estimate can be used here for the asymptotic covariance of the MM estimate. Besides the three estimators H1, H2, and H3 as described in the section "Asymptotic Covariance and Confidence Intervals" on page 5677, a weighted covariance estimator H4 is available:

$$\text{H4: } K^2 \frac{[1/(n-p)]\sum(\psi(r_i))^2}{[(1/n)\sum(\psi'(r_i))]^2} W^{-1}$$

where $K = 1 + \frac{p}{n}\frac{\text{Var}(\psi')}{(E\psi')^2}$ is the correction factor and $W_{jk} = \frac{1}{\bar{w}}\sum w_i x_{ij} x_{ik}$, $\bar{w} = \frac{1}{n}\sum w_i$.

You can specify these estimators with the option ASYMPCOV= [H1 | H2 | H3 | H4]. The ROBUSTREG procedure uses H4 as the default. Confidence intervals for estimated parameters are computed from the diagonal elements of the estimated asymptotic covariance matrix.

R Square and Deviance

The robust version of R^2 for the MM estimate is defined as

$$R^2 = \frac{\sum \rho(\frac{y_i - \hat{\mu}}{\hat{s}}) - \sum \rho(\frac{y_i - x_i^T\hat{\theta}}{\hat{s}})}{\sum \rho(\frac{y_i - \hat{\mu}}{\hat{s}})}$$

and the robust deviance is defined as the optimal value of the objective function on the σ^2 scale:

$$D = 2(\hat{s})^2 \sum \rho(\frac{y_i - x_i^T\hat{\theta}}{\hat{s}})$$

where $\rho' = \psi$, $\hat{\theta}$ is the MM estimator of θ, $\hat{\mu}$ is the MM estimator of location, and $\hat{s}$ is the MM estimator of the scale parameter in the full model.

Linear Tests

For MM estimation, the same ρ test and R_n^2 test used for M estimation can be used. See the section "Linear Tests" on page 5678 for details.

Model Selection

For MM estimation, the same two model selection methods used for M estimation can be used. See the section "Model Selection" on page 5679 for details.

Robust Distance

The ROBUSTREG procedure uses the robust multivariate location and scale estimates for leverage-point detection. The procedure computes a robust version of the Mahalanobis distance by using the minimum covariance determinant (MCD) method of Rousseeuw (1984).

Algorithm

PROC ROBUSTREG implements the algorithm given by Rousseeuw and Van Driessen (1999) for MCD, which is similar to the algorithm for LTS.

Robust Distance

The Mahalanobis distance is defined as

$$MD(x_i) = [(x_i - \bar{x})^T \bar{C}(X)^{-1}(x_i - \bar{x})]^{1/2}$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$ and $\bar{C}(X) = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^T(x_i - \bar{x})$ are the empirical multivariate location and scale. Here $x_i = (x_{i1}, \ldots, x_{i(p-1)})^T$ does not include the intercept. The relation between the Mahalanobis distance $MD(x_i)$ and the hat matrix $H = (h_{ij}) = X(X^TX)^{-1}X^T$ is

$$h_{ii} = \frac{1}{n-1}MD_i^2 + \frac{1}{n}$$

The robust distance is defined as

$$RD(x_i) = [(x_i - T(X))^T C(X)^{-1}(x_i - T(X))]^{1/2}$$

where $T(X)$ and $C(X)$ are the robust multivariate location and scale obtained by MCD.

These distances are used to detect leverage points.

Leverage Point and Outlier Detection

Let $C(p) = \sqrt{\chi^2_{p;1-\alpha}}$ be the cutoff value. The variable LEVERAGE is defined as

$$\text{LEVERAGE} = \begin{cases} 0 & \text{if } RD(x_i) \leq C(p) \\ 1 & \text{otherwise} \end{cases}$$

You can specify a cutoff value with the LEVERAGE option in the MODEL statement.

Residuals $r_i, i = 1, \ldots, n$, based on robust regression estimates are used to detect vertical outliers. The variable OUTLIER is defined as

$$\text{OUTLIER} = \begin{cases} 0 & \text{if } |r_i| \leq k\hat{\sigma} \\ 1 & \text{otherwise} \end{cases}$$

You can specify the multiplier k of the cutoff value by using the CUTOFF= option in the MODEL statement. By default, $k = 3$. $\hat{\sigma}$ is estimated scale in the model.

An ODS table called DIAGNOSTICS contains these two variables.

INEST= Data Set

When you use the M or MM estimation, you can use the INEST= data set to specify initial estimates for all the parameters in the model. The INEST= option is ignored if you specify LTS or S estimation by using the METHOD=LTS or METHOD=S option or if you specify the INITEST= option after the METHOD=MM option in the PROC statement. The INEST= data set must contain the intercept variable (named Intercept) and all independent variables in the MODEL statement.

If BY processing is used, the INEST= data set should also include the BY variables, and there must be at least one observation for each BY group. If there is more than one observation in a BY group, the first one read is used for that BY group.

If the INEST= data set also contains the _TYPE_ variable, only observations with _TYPE_ value "PARMS" are used as starting values.

You can specify starting values for the iteratively reweighted least squares algorithm in the INEST= data set. The INEST= data set has the same structure as the OUTEST= data set but is not required to have all the variables or observations that appear in the OUTEST= data set. One simple use of the INEST= option is passing the previous OUTEST= data set directly to the next model as an INEST= data set, assuming that the two models have the same parameterization.

OUTEST= Data Set

The OUTEST= data set contains parameter estimates for the model. You can specify a label in the MODEL statement to distinguish between the estimates for different models used by the ROBUSTREG procedure. If the COVOUT option is specified, the OUTEST= data set also contains the estimated covariance matrix of the parameter estimates. Note that, if the ROBUSTREG procedure does not converge, the parameter estimates are set to missing in the OUTEST data set.

The OUTEST= data set contains all variables specified in the MODEL statement and the BY statement. One observation consists of parameter values for the model with the dependent variable having the value -1. If the COVOUT option is specified, there are additional observations containing the rows of the estimated covariance matrix. For these observations, the dependent variable contains the parameter estimate for the corresponding row variable. The following variables are also added to the data set:

MODEL	is a character variable containing the label of the MODEL statement, if present. Otherwise, the variable's value is blank.
NAME	is a character variable containing the name of the dependent variable for the parameter estimates or the name of the row for the covariance matrix estimates.
TYPE	is a character variable containing the type of the observation, either PARMS for parameter estimates or COV for covariance estimates.
METHOD	is a character variable containing the type of estimation method, either M estimation, LTS estimation, S estimation, or MM estimation.
STATUS	is a character variable containing the status of model fitting, either Converged, Warning, or Failed.
INTERCEPT	is a numeric variable containing the intercept parameter estimates and covariances.
SCALE	is a numeric variable containing the scale parameter estimates.

Any BY variables specified are also added to the OUTEST= data set.

Computational Resources

The algorithms for the various estimation methods need a different amount of memory for working space. Let p be the number of parameters estimated and n be the number of observations used in the model estimation.

For M estimation, the minimum working space (in bytes) needed is

$$3n + 2p^2 + 30p$$

If sufficient space is available, the input data set is also kept in memory; otherwise, the input data set is reread for computing the iteratively reweighted least squares estimates and the execution

time of the procedure increases substantially. For each reweighted least squares, $O(np^2 + p^3)$ multiplications and additions are required for computing the crossproduct matrix and its inverse. The $O(v)$ notation means that, for large values of the argument, v, $O(v)$ is approximately a constant times v.

Since the iteratively reweighted least squares algorithm converges very quickly (normally within fewer than 20 iterations), the computation of M estimates is fast.

LTS estimation is more expensive in computation. The minimum working space (in bytes) needed is

$$np + 12n + 4p^2 + 60p$$

The memory is mainly used to store the current data used by LTS for modeling. The LTS algorithm uses subsampling and spends much of its computing time on resampling and computing estimates for subsamples. Since it resamples if singularity is detected, it might take more time if the data set has serious singularities.

The MCD algorithm for leverage-point diagnostics is similar to the LTS algorithm.

ODS Table Names

The ROBUSTREG procedure assigns a name to each table it creates. You can specify these names when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the following table.

Table 74.10 ODS Tables Produced by PROC ROBUSTREG

ODS Table Name	Description	Statement	Option
BestEstimates	Best final estimates for LTS	PROC	SUBANALYSIS
BestSubEstimates	Best estimates for each subgroup	PROC	SUBANALYSIS
BiasTest	Bias test for MM estimation	PROC	BIASTEST
ClassLevels	Classification variable levels	CLASS	default
CorrB	Parameter estimate correlation matrix	MODEL	CORRB
CovB	Parameter estimate covariance matrix	MODEL	COVB
CStep	C-step for LTS fitting	PROC	SUBANALYSIS
Diagnostics	Outlier diagnostics	MODEL	DIAGNOSTICS
DiagSummary	Summary of the outlier diagnostics	MODEL	default
GoodFit	R2, deviance, AIC, and BIC	MODEL	METHOD
InitLTSProfile	Profile for initial LTS estimate	PROC	METHOD
InitSProfile	Profile for initial S estimate	PROC	METHOD
IterHistory	Iteration history	PROC	ITPRINT
LTSEstimates	LTS parameter estimates	PROC	METHOD
LTSLocationScale	Location and scale for LTS	PROC	METHOD
LTSProfile	Profile for LTS estimator	PROC	METHOD
LTSRsquare	R2 for LTS estimate	PROC	METHOD
MMProfile	Profile for MM estimator	PROC	METHOD
ModelInfo	Model information	MODEL	default

Table 74.10 (continued)

ODS Table Name	Description	Statement	Option
NObs	Observations summary	PROC	default
ParameterEstimates	Parameter estimates	MODEL	default
ParameterEstimatesF	Final weighted LS estimates	PROC	FWLS
ParameterEstimatesR	Reduced parameter estimates	TEST	default
ParmInfo	Parameter indices	MODEL	default
SProfile	Profile for S estimator	PROC	METHOD
Groups	Groups for LTS fitting	PROC	SUBANALYSIS
SummaryStatistics	Summary statistics for model variables	MODEL	default
Tests	Results for tests	TEST	default

ODS Graphics

Graphical displays are important in robust regression and outlier detection. This section provides information about the basic ODS statistical graphics produced by the ROBUSTREG procedure.

If the model includes a single continuous independent variable, a plot of robust fit against this variable (FITPLOT) is provided by default. For diagnostics, two plots are particularly useful in revealing outliers and leverage points. The first is a scatter plot of the standardized robust residuals against the robust distances (RDPLOT). The second is a scatter plot of the robust distances against the classical Mahalanobis distances (DDPLOT). In addition to these two plots, a histogram and a quantile-quantile plot of the standardized robust residuals are also helpful.

These plots are controlled by the PLOTS= option in the PROC statement. You can specify more than one plot request with the PLOTS= option. Table 74.11 summarizes these requests.

In addition to the PLOTS= option, you must specify the ODS GRAPHICS statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

The names of the graphs that PROC ROBUSTREG generates are listed in Table 74.12, along with the required statements and options. The following subsections provide information about these graphs.

Table 74.11 Options for Plots

Option	Plot
ALL	All appropriate plots
DDPLOT	Robust distance vs. Mahalanobis distance
FITPLOT	Robust fit vs. independent variable
HISTOGRAM	Histogram of standardized robust residuals
NONE	No plot
QQPLOT	Q-Q plot of standardized robust residuals
RDPLOT	Standardized robust residual vs. robust distance

Fit Plot

When the model has a single independent continuous variable (with or without the intercept), the ROBUSTREG procedure automatically creates a plot of robust fit against this independent variable.

The following simple example is used to show the fit plot. The data, from Rousseeuw and Leroy (1987, Table 3), include the logarithm of surface temperature and the logarithm of light intensity for 47 stars in the direction of the constellation Cygnus.

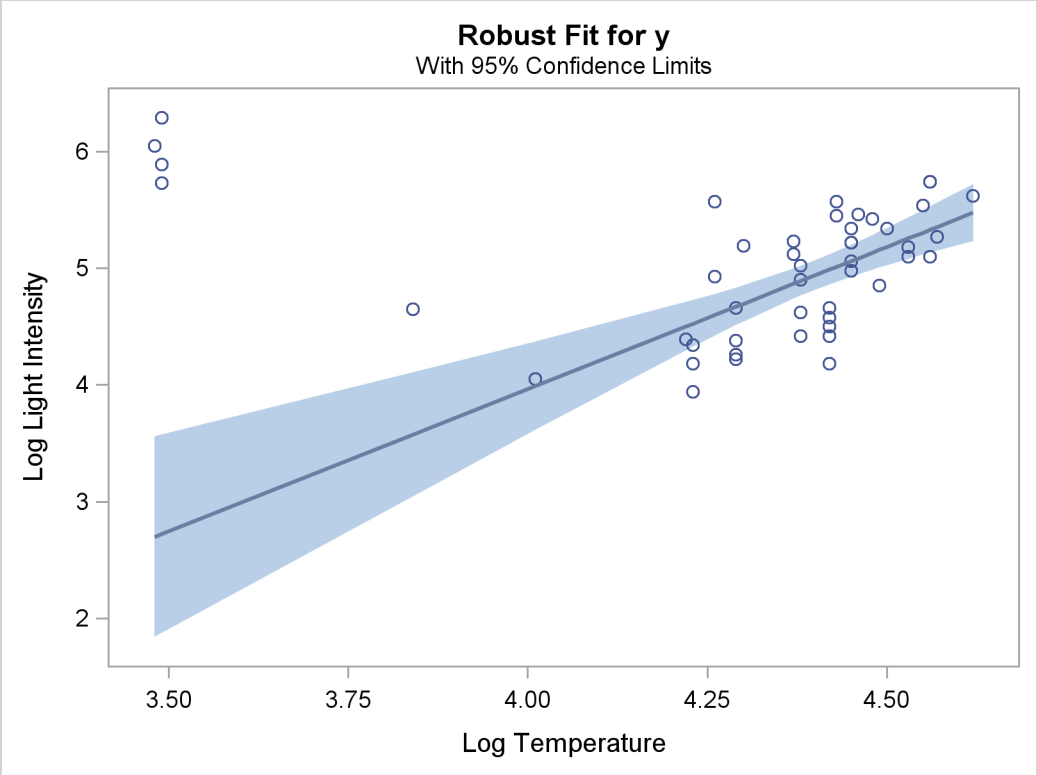
```
data star;
   input index x y @@;
   label x = 'Log Temperature'
         y = 'Log Light Intensity';
datalines;
1   4.37  5.23     25  4.38  5.02
2   4.56  5.74     26  4.42  4.66
3   4.26  4.93     27  4.29  4.66
4   4.56  5.74     28  4.38  4.90
5   4.30  5.19     29  4.22  4.39
6   4.46  5.46     30  3.48  6.05
7   3.84  4.65     31  4.38  4.42
8   4.57  5.27     32  4.56  5.10
9   4.26  5.57     33  4.45  5.22
10  4.37  5.12     34  3.49  6.29
11  3.49  5.73     35  4.23  4.34
12  4.43  5.45     36  4.62  5.62
13  4.48  5.42     37  4.53  5.10
14  4.01  4.05     38  4.45  5.22
15  4.29  4.26     39  4.53  5.18
16  4.42  4.58     40  4.43  5.57
17  4.23  3.94     41  4.38  4.62
18  4.42  4.18     42  4.45  5.06
19  4.23  4.18     43  4.50  5.34
20  3.49  5.89     44  4.45  5.34
21  4.29  4.38     45  4.55  5.54
22  4.29  4.22     46  4.45  4.98
23  4.42  4.42     47  4.42  4.50
24  4.49  4.85
;
```

The fit plot in Figure 74.17, created with the following statements, plots the robust fit of the logarithm of light intensity with the MM method against the logarithm of the surface temperature.

```
ods graphics on;

proc robustreg data=star method=mm ;
    model y = x;
run;

ods graphics off;
```

Confidence limits are added on the plot by default. The NOLIMITS option suppresses these limits.

Figure 74.17 Robust Fit




```
ods graphics on;

proc robustreg data=star method=mm plot=fitplot(nolimits);
    model y = x;
run;

ods graphics off;
```

Distance-Distance Plot

The distance-distance plot (DDPLOT) is mainly used for leverage-point diagnostics. It is a scatter plot of the robust distances against the classical Mahalanobis distances for the continuous independent variables. See the section "Leverage Point and Outlier Detection" on page 5689 for details about the robust distance. If there is a classification variable specified in the model, this plot is not created.

You can use the PLOT=DDPLOT option to request this plot. The following statements use the stack data set in the section "M Estimation" on page 5649 to create the single plot shown in Figure 74.5.

```
ods graphics on;

proc robustreg data=stack plot=ddplot;
   model y = x1 x2 x3;
run;

ods graphics off;
```

The reference lines represent the cutoff values. The diagonal line is also drawn to show the distribution of the distances. By default, all outliers and leverage points are labeled with observation numbers. To change the default, you can use the LABEL= option as described in Table 74.2.

If you specify ID variables in the ID statement, instead of observation numbers, the values of the first ID variable are used as labels.

Residual-Distance Plot

The residual-distance plot (RDPLOT) is used for both outlier and leverage-point diagnostics. It is a scatter plot of the standardized robust residuals against the robust distances. See the section "Leverage Point and Outlier Detection" on page 5689 for details about the robust distance. If there is a classification variable specified in the model, this plot is not created.

You can use the PLOT=RDPLOT option to request this plot. The following statements use the stack data set in the section "M Estimation" on page 5649 to create a single plot shown in Figure 74.4.

```
ods graphics on;

proc robustreg data=stack plot=rdplot;
   model y = x1 x2 x3;
run;

ods graphics off;
```

The reference lines represent the cutoff values. By default, all outliers and leverage points are labeled with observation numbers. To change the default, you can use the LABEL= option as described in Table 74.2.

If you specify ID variables in the ID statement instead of observation numbers, the values of the first ID variable are used as labels.

Histogram and Q-Q Plot

PROC ROBUSTREG produces a histogram and a Q-Q plot for the standardized robust residuals. The histogram is superimposed with a normal density curve and a kernel density curve. Using the stack data set in the section "M Estimation" on page 5649, the following statements create the plots in Figure 74.6 and Figure 74.7.

```
ods graphics on;

proc robustreg data=stack plots=(histogram qqplot);
   model y = x1 x2 x3;
run;

ods graphics off;
```

ODS Graph Names

PROC ROBUSTREG assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 74.12.

To request these graphs you must specify the ODS GRAPHICS statement in addition to the PLOTS= option described in Table 74.11. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Table 74.12 ODS Graphics Produced by PROC ROBUSTREG

ODS Graph Name	Plot Description	Statement	PLOTS= Option
DDPlot	Robust distance vs. Mahalanobis distance	PROC	DDPLOT
FitPlot	Robust fit vs. independent variable	PROC	FITPLOT
Histogram	Histogram of standardized robust residuals	PROC	HISTOGRAM
QQPlot	Q-Q plot of standardized robust residuals	PROC	QQPLOT
RDPlot	Standardized robust residual vs. robust distance	PROC	RDPLOT

Examples: ROBUSTREG Procedure

Example 74.1: Comparison of Robust Estimates

This example contrasts several of the robust methods available in the ROBUSTREG procedure.

The following statements generate 1000 random observations. The first 900 observations are from a linear model, and the last 100 observations are significantly biased in the y-direction. In other words, 10% of the observations are contaminated with outliers.

```
data a (drop=i);
   do i=1 to 1000;
      x1=rannor(1234);
      x2=rannor(1234);
      e=rannor(1234);
      if i > 900 then y=100 + e;
      else y=10 + 5*x1 + 3*x2 + .5 * e;
      output;
   end;
run;
```

The following statements invoke PROC REG and PROC ROBUSTREG with the data set a.

```
proc reg data=a;
   model y = x1 x2;
run;
proc robustreg data=a method=m ;
   model y = x1 x2;
run;
proc robustreg data=a method=mm;
   model y = x1 x2;
run;
proc robustreg data=a method=s ;
   model y = x1 x2;
run;
proc robustreg data=a method=lts ;
   model y = x1 x2;
run;
```

The tables of parameter estimates generated with the ROBUSTREG procedure by using M estimation, MM estimation, S estimation, and LTS estimation are shown in Output 74.1.2, Output 74.1.3, Output 74.1.4, and Output 74.1.5, respectively. For comparison, the ordinary least squares (OLS) estimates produced by the REG procedure (Chapter 73, "The REG Procedure") are shown in Output 74.1.1. The four robust methods, M, MM, S, and LTS, correctly estimate the regression coefficients for the underlying model (10, 5, and 3), but the OLS estimate does not.

Output 74.1.1 OLS Estimates for Data with 10% Contamination

The REG Procedure
Model: MODEL1
Dependent Variable: y

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	19.06712	0.86322	22.09	<.0001
x1	1	3.55485	0.86892	4.09	<.0001
x2	1	2.12341	0.83039	2.56	0.0107

Output 74.1.2 M Estimates for Data with 10% Contamination

The ROBUSTREG Procedure

Model Information

Data Set	WORK.A
Dependent Variable	y
Number of Independent Variables	2
Number of Observations	1000
Method	M Estimation

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	10.0024	0.0174	9.9683	10.0364	331908	<.0001
x1	1	5.0077	0.0175	4.9735	5.0420	82106.9	<.0001
x2	1	3.0161	0.0167	2.9834	3.0488	32612.5	<.0001
Scale	1	0.5780					

Output 74.1.3 MM Estimates for Data with 10% Contamination

The ROBUSTREG Procedure

Model Information

Data Set	WORK.A
Dependent Variable	y
Number of Independent Variables	2
Number of Observations	1000
Method	MM Estimation

Output 74.1.3 *continued*

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	10.0035	0.0176	9.9690	10.0379	323947	<.0001
x1	1	5.0085	0.0178	4.9737	5.0433	79600.6	<.0001
x2	1	3.0181	0.0168	2.9851	3.0511	32165.0	<.0001
Scale	0	0.6733					

Output 74.1.4 S Estimates for Data with 10% Contamination

The ROBUSTREG Procedure

Model Information

Data Set	WORK.A
Dependent Variable	y
Number of Independent Variables	2
Number of Observations	1000
Method	S Estimation

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	10.0055	0.0180	9.9703	10.0408	309917	<.0001
x1	1	5.0096	0.0182	4.9740	5.0452	76045.2	<.0001
x2	1	3.0210	0.0172	2.9873	3.0547	30841.3	<.0001
Scale	0	0.6721					

Output 74.1.5 LTS Estimates for Data with 10% Contamination

The ROBUSTREG Procedure

Model Information

Data Set	WORK.A
Dependent Variable	y
Number of Independent Variables	2
Number of Observations	1000
Method	LTS Estimation

Output 74.1.5 *continued*

LTS Parameter Estimates

Parameter	DF	Estimate
Intercept	1	10.0083
x1	1	5.0316
x2	1	3.0396
Scale (sLTS)	0	0.5880
Scale (Wscale)	0	0.5113

The next statements demonstrate that if the percentage of contamination is increased to 40%, the M method and the MM method with default options fail to estimate the underlying model. Output 74.1.6 and Output 74.1.7 display these estimates. However, by tuning the constant c for the M method and the constants INITH and K0 for the MM method, you can increase the breakdown values of the estimates and capture the right model. Output 74.1.8 and Output 74.1.9 display these estimates. Similarly, you can tune the constant EFF for the S method and the constant H for the LTS method and correctly estimate the underlying model with these methods. Results are not presented.

```
data b (drop=i);
   do i=1 to 1000;
      x1=rannor(1234);
      x2=rannor(1234);
      e=rannor(1234);
      if i > 600 then y=100 + e;
      else y=10 + 5*x1 +  3*x2 + .5 * e;
      output;
   end;
run;

proc robustreg data=b method=m ;
   model y = x1 x2;
run;

proc robustreg data=b method=mm;
   model y = x1 x2;
run;

proc robustreg data=b method=m(wf=bisquare(c=2));
   model y = x1 x2;
run;

proc robustreg data=b method=mm(inith=502 k0=1.8);
   model y = x1 x2;
run;
```

Output 74.1.6 M Estimates (Default Setting) for Data with 40% Contamination

The ROBUSTREG Procedure

Model Information

Data Set	WORK.B
Dependent Variable	y
Number of Independent Variables	2
Number of Observations	1000
Method	M Estimation

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	44.8991	1.5609	41.8399	47.9584	827.46	<.0001
x1	1	2.4309	1.5712	-0.6485	5.5104	2.39	0.1218
x2	1	1.3742	1.5015	-1.5687	4.3171	0.84	0.3601
Scale	1	56.6342					

Output 74.1.7 MM Estimates (Default Setting) for Data with 40% Contamination

The ROBUSTREG Procedure

Model Information

Data Set	WORK.B
Dependent Variable	y
Number of Independent Variables	2
Number of Observations	1000
Method	MM Estimation

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	43.0607	1.7978	39.5370	46.5844	573.67	<.0001
x1	1	2.7369	1.8140	-0.8185	6.2924	2.28	0.1314
x2	1	1.5211	1.7265	-1.8628	4.9049	0.78	0.3783
Scale	0	52.8496					

Output 74.1.8 M Estimates (Tuned) for Data with 40% Contamination

The ROBUSTREG Procedure

Model Information

Data Set	WORK.B
Dependent Variable	y
Number of Independent Variables	2
Number of Observations	1000
Method	M Estimation

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	10.0137	0.0219	9.9708	10.0565	209688	<.0001
x1	1	4.9905	0.0220	4.9473	5.0336	51399.1	<.0001
x2	1	3.0399	0.0210	2.9987	3.0811	20882.4	<.0001
Scale	1	1.0531					

Output 74.1.9 MM Estimates (Tuned) for Data with 40% Contamination

The ROBUSTREG Procedure

Model Information

Data Set	WORK.B
Dependent Variable	y
Number of Independent Variables	2
Number of Observations	1000
Method	MM Estimation

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	10.0103	0.0213	9.9686	10.0520	221639	<.0001
x1	1	4.9890	0.0218	4.9463	5.0316	52535.9	<.0001
x2	1	3.0363	0.0201	2.9970	3.0756	22895.5	<.0001
Scale	0	1.8992					

When there are bad leverage points, the M method fails to estimate the underlying model no matter what constant c you use. In this case, other methods (LTS, S, and MM) in PROC ROBUSTREG, which are robust to bad leverage points, will correctly estimate the underlying model.

The following statements generate 1000 observations with 1% bad high leverage points.

```
data c (drop=i);
   do i=1 to 1000;
      x1=rannor(1234);
      x2=rannor(1234);
      e=rannor(1234);
      if i > 600 then y=100 + e;
      else y=10 + 5*x1 + 3*x2 + .5 * e;
      if i < 11 then x1=200 * rannor(1234);
      if i < 11 then x2=200 * rannor(1234);
      if i < 11 then y= 100*e;
      output;
   end;
run;
proc robustreg data=c method=mm(inith=502 k0=1.8);
   model y = x1 x2;
run;
proc robustreg data=c method=s(k0=1.8);
   model y = x1 x2;
run;
proc robustreg data=c method=lts(h=502);
   model y = x1 x2;
run;
```

Output 74.1.10 displays the MM estimates with initial LTS estimates, Output 74.1.11 displays the S estimates, and Output 74.1.12 displays the LTS estimates.

Output 74.1.10 MM Estimates for Data with 1% Leverage Points

The ROBUSTREG Procedure

Model Information

Data Set	WORK.C
Dependent Variable	y
Number of Independent Variables	2
Number of Observations	1000
Method	MM Estimation

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	9.9820	0.0215	9.9398	10.0241	215369	<.0001
x1	1	5.0303	0.0206	4.9898	5.0707	59469.1	<.0001
x2	1	3.0222	0.0221	2.9789	3.0655	18744.9	<.0001
Scale	0	2.2134					

Output 74.1.11 S Estimates for Data with 1% Leverage Points

The ROBUSTREG Procedure

Model Information

Data Set	WORK.C
Dependent Variable	y
Number of Independent Variables	2
Number of Observations	1000
Method	S Estimation

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	9.9808	0.0216	9.9383	10.0232	212532	<.0001
x1	1	5.0303	0.0208	4.9896	5.0710	58656.3	<.0001
x2	1	3.0217	0.0222	2.9782	3.0652	18555.7	<.0001
Scale	0	2.2094					

Output 74.1.12 LTS Estimates for Data with 1% Leverage Points

The ROBUSTREG Procedure

Model Information

Data Set	WORK.C
Dependent Variable	y
Number of Independent Variables	2
Number of Observations	1000
Method	LTS Estimation

LTS Parameter Estimates

Parameter	DF	Estimate
Intercept	1	9.9742
x1	1	5.0010
x2	1	3.0219
Scale (sLTS)	0	0.9952
Scale (Wscale)	0	0.5216

Example 74.2: Robust ANOVA

The classical analysis of variance (ANOVA) technique based on least squares assumes that the underlying experimental errors are normally distributed. However, data often contain outliers due to recording or other errors. In other cases, extreme responses occur when control variables in the experiments are set to extremes. It is important to distinguish these extreme points and determine

whether they are outliers or important extreme cases. You can use the ROBUSTREG procedure for robust analysis of variance based on M estimation. Typically, there are no high leverage points in a well-designed experiment, so M estimation is appropriate.

The following example shows how to use the ROBUSTREG procedure for robust ANOVA.

An experiment was carried out to study the effects of two successive treatments (T1, T2) on the recovery time of mice with certain diseases. Sixteen mice were randomly assigned into four groups for the four different combinations of the treatments. The recovery times (time) were recorded (in hours) as shown in the following data set recover.

```
data recover;
   input  T1 $ T2 $ time @@;
datalines;
0 0 20.2  0 0 23.9  0 0 21.9  0 0 42.4
1 0 27.2  1 0 34.0  1 0 27.4  1 0 28.5
0 1 25.9  0 1 34.5  0 1 25.1  0 1 34.2
1 1 35.0  1 1 33.9  1 1 38.3  1 1 39.9
;
```

The following statements invoke the GLM procedure (Chapter 39, "The GLM Procedure") for a standard ANOVA.

```
proc glm data=recover;
    class T1 T2;
    model time = T1 T2 T1*T2;
run;
```

Output 74.2.1 Overall ANOVA

The GLM Procedure

Dependent Variable: time

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	209.9118750	69.9706250	1.86	0.1905
Error	12	451.9225000	37.6602083		
Corrected Total	15	661.8343750			

R-Square	Coeff Var	Root MSE	time Mean
0.317167	19.94488	6.136791	30.76875

Output 74.2.2 Model ANOVA

Source	DF	Type I SS	Mean Square	F Value	Pr > F
T1	1	81.4506250	81.4506250	2.16	0.1671
T2	1	106.6056250	106.6056250	2.83	0.1183
T1*T2	1	21.8556250	21.8556250	0.58	0.4609

Output 74.2.1 indicates that the overall model effect is not significant at the 10% level, and Output 74.2.2 indicates that neither treatment is significant at the 10% level.

The following statements invoke the ROBUSTREG procedure with the same model.

```
proc robustreg data=recover;
   class T1 T2;
   model time = T1 T2 T1*T2 / diagnostics;
   T1_T2: test T1*T2;
   output out=robout r=resid sr=stdres;
run;
```

Output 74.2.3 shows some basic information about the model and the response variable time.

Output 74.2.3 Model Fitting Information and Summary Statistics

The ROBUSTREG Procedure

Model Information

Data Set	WORK.RECOVER
Dependent Variable	time
Number of Independent Variables	2
Number of Continuous Independent Variables	0
Number of Class Independent Variables	2
Number of Observations	16
Method	M Estimation

Summary Statistics

Variable	Q1	Median	Q3	Mean	Standard Deviation	MAD
time	25.5000	31.2000	34.7500	30.7688	6.6425	6.8941

The "Parameter Estimates" table in Output 74.2.4 indicates that the main effects of both treatments are significant at the 5% level.

Output 74.2.4 Model Parameter Estimates

Parameter Estimates

Parameter			DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept			1	36.7655	2.0489	32.7497	40.7814	321.98	<.0001
T1	0		1	-6.8307	2.8976	-12.5100	-1.1514	5.56	0.0184
T1	1		0	0.0000	.	.	.	.	.
T2	0		1	-7.6755	2.8976	-13.3548	-1.9962	7.02	0.0081
T2	1		0	0.0000	.	.	.	.	.
T1*T2	0	0	1	-0.2619	4.0979	-8.2936	7.7698	0.00	0.9490
T1*T2	0	1	0	0.0000	.	.	.	.	.
T1*T2	1	0	0	0.0000	.	.	.	.	.
T1*T2	1	1	0	0.0000	.	.	.	.	.
Scale			1	3.5346					

The reason for the difference between the traditional ANOVA and the robust ANOVA is explained by Output 74.2.5, which shows that the fourth observation is an outlier. Further investigation shows that the original value of 24.4 for the fourth observation was recorded incorrectly.

Output 74.2.6 displays the robust test results. The interaction between the two treatments is not significant. Output 74.2.7 displays the robust residuals and standardized robust residuals.

Output 74.2.5 Diagnostics

Diagnostics

Obs	Standardized Robust Residual	Outlier
4	5.7722	*

Output 74.2.6 Test of Significance

Robust Linear Tests

T1_T2

Test	Test Statistic	Lambda	DF	Chi-Square	Pr > ChiSq
Rho	0.0041	0.7977	1	0.01	0.9431
Rn2	0.0041		1	0.00	0.9490

Output 74.2.7 ROBUSTREG Output

Obs	T1	T2	time	resid	stdres
1	0	0	20.2	-1.7974	-0.50851
2	0	0	23.9	1.9026	0.53827
3	0	0	21.9	-0.0974	-0.02756
4	0	0	42.4	20.4026	5.77222
5	1	0	27.2	-1.8900	-0.53472
6	1	0	34.0	4.9100	1.38911
7	1	0	27.4	-1.6900	-0.47813
8	1	0	28.5	-0.5900	-0.16693
9	0	1	25.9	-4.0348	-1.14152
10	0	1	34.5	4.5652	1.29156
11	0	1	25.1	-4.8348	-1.36785
12	0	1	34.2	4.2652	1.20668
13	1	1	35.0	-1.7655	-0.49950
14	1	1	33.9	-2.8655	-0.81070
15	1	1	38.3	1.5345	0.43413
16	1	1	39.9	3.1345	0.88679

Example 74.3: Growth Study of De Long and Summers

Robust regression and outlier detection techniques have considerable applications to econometrics. The following example from Zaman, Rousseeuw, and Orhan (2001) shows how these techniques substantially improve the ordinary least squares (OLS) results for the growth study of De Long and Summers.

De Long and Summers (1991) studied the national growth of 61 countries from 1960 to 1985 by using OLS with the following data set growth.

```
data growth;
   input country$ GDP LFG EQP NEQ GAP @@;
datalines;
Argentin  0.0089 0.0118 0.0214 0.2286 0.6079
Austria   0.0332 0.0014 0.0991 0.1349 0.5809
Belgium   0.0256 0.0061 0.0684 0.1653 0.4109
Bolivia   0.0124 0.0209 0.0167 0.1133 0.8634
Botswana  0.0676 0.0239 0.1310 0.1490 0.9474
Brazil    0.0437 0.0306 0.0646 0.1588 0.8498
Cameroon  0.0458 0.0169 0.0415 0.0885 0.9333
Canada    0.0169 0.0261 0.0771 0.1529 0.1783
Chile     0.0021 0.0216 0.0154 0.2846 0.5402
Colombia  0.0239 0.0266 0.0229 0.1553 0.7695
CostaRic  0.0121 0.0354 0.0433 0.1067 0.7043
Denmark   0.0187 0.0115 0.0688 0.1834 0.4079
Dominica  0.0199 0.0280 0.0321 0.1379 0.8293
Ecuador   0.0283 0.0274 0.0303 0.2097 0.8205
ElSalvad  0.0046 0.0316 0.0223 0.0577 0.8414
Ethiopia  0.0094 0.0206 0.0212 0.0288 0.9805
Finland   0.0301 0.0083 0.1206 0.2494 0.5589
France    0.0292 0.0089 0.0879 0.1767 0.4708
Germany   0.0259 0.0047 0.0890 0.1885 0.4585
Greece    0.0446 0.0044 0.0655 0.2245 0.7924
Guatemal  0.0149 0.0242 0.0384 0.0516 0.7885
Honduras  0.0148 0.0303 0.0446 0.0954 0.8850
HongKong  0.0484 0.0359 0.0767 0.1233 0.7471
India     0.0115 0.0170 0.0278 0.1448 0.9356
Indonesi  0.0345 0.0213 0.0221 0.1179 0.9243
Ireland   0.0288 0.0081 0.0814 0.1879 0.6457
Israel    0.0452 0.0305 0.1112 0.1788 0.6816
Italy     0.0362 0.0038 0.0683 0.1790 0.5441
IvoryCoa  0.0278 0.0274 0.0243 0.0957 0.9207
Jamaica   0.0055 0.0201 0.0609 0.1455 0.8229
Japan     0.0535 0.0117 0.1223 0.2464 0.7484
Kenya     0.0146 0.0346 0.0462 0.1268 0.9415
Korea     0.0479 0.0282 0.0557 0.1842 0.8807
Luxembou  0.0236 0.0064 0.0711 0.1944 0.2863
Madagasc -0.0102 0.0203 0.0219 0.0481 0.9217
Malawi    0.0153 0.0226 0.0361 0.0935 0.9628
Malaysia  0.0332 0.0316 0.0446 0.1878 0.7853
Mali      0.0044 0.0184 0.0433 0.0267 0.9478
Mexico    0.0198 0.0349 0.0273 0.1687 0.5921
```

```
Morocco    0.0243 0.0281 0.0260 0.0540 0.8405
Netherla   0.0231 0.0146 0.0778 0.1781 0.3605
Nigeria   -0.0047 0.0283 0.0358 0.0842 0.8579
Norway     0.0260 0.0150 0.0701 0.2199 0.3755
Pakistan   0.0295 0.0258 0.0263 0.0880 0.9180
Panama     0.0295 0.0279 0.0388 0.2212 0.8015
Paraguay   0.0261 0.0299 0.0189 0.1011 0.8458
Peru       0.0107 0.0271 0.0267 0.0933 0.7406
Philippi   0.0179 0.0253 0.0445 0.0974 0.8747
Portugal   0.0318 0.0118 0.0729 0.1571 0.8033
Senegal   -0.0011 0.0274 0.0193 0.0807 0.8884
Spain      0.0373 0.0069 0.0397 0.1305 0.6613
SriLanka   0.0137 0.0207 0.0138 0.1352 0.8555
Tanzania   0.0184 0.0276 0.0860 0.0940 0.9762
Thailand   0.0341 0.0278 0.0395 0.1412 0.9174
Tunisia    0.0279 0.0256 0.0428 0.0972 0.7838
U.K.       0.0189 0.0048 0.0694 0.1132 0.4307
U.S.       0.0133 0.0189 0.0762 0.1356 0.0000
Uruguay    0.0041 0.0052 0.0155 0.1154 0.5782
Venezuel   0.0120 0.0378 0.0340 0.0760 0.4974
Zambia    -0.0110 0.0275 0.0702 0.2012 0.8695
Zimbabwe   0.0110 0.0309 0.0843 0.1257 0.8875
;
```

The regression equation they used is

$$GDP = \beta_0 + \beta_1 LFG + \beta_2 GAP + \beta_3 EQP + \beta_4 NEQ + \epsilon$$

where the response variable is the growth in gross domestic product per worker (GDP) and the regressors are labor force growth (LFG), relative GDP gap (GAP), equipment investment (EQP), and nonequipment investment (NEQ).

The following statements invoke the REG procedure (Chapter 73, "The REG Procedure") for the OLS analysis:

```
proc reg data=growth;
   model GDP  = LFG GAP EQP NEQ ;
run;
```

Output 74.3.1 OLS Estimates

The REG Procedure
Model: MODEL1
Dependent Variable: GDP

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	−0.01430	0.01028	−1.39	0.1697
LFG	1	−0.02981	0.19838	−0.15	0.8811
GAP	1	0.02026	0.00917	2.21	0.0313
EQP	1	0.26538	0.06529	4.06	0.0002
NEQ	1	0.06236	0.03482	1.79	0.0787

The OLS analysis shown in Output 74.3.1 indicates that GAP and EQP have a significant influence on GDP at the 5% level.

The following statements invoke the ROBUSTREG procedure with the default M estimation.

```
ods graphics on;

proc robustreg data=growth plots=all;
   model GDP  = LFG GAP EQP NEQ / diagnostics leverage;
   id country;
run;

ods graphics off;
```

Output 74.3.2 displays model information and summary statistics for variables in the model.

Output 74.3.2 Model Fitting Information and Summary Statistics

The ROBUSTREG Procedure

Model Information

Data Set	WORK.GROWTH
Dependent Variable	GDP
Number of Independent Variables	4
Number of Observations	61
Method	M Estimation

Summary Statistics

Variable	Q1	Median	Q3	Mean	Standard Deviation	MAD
LFG	0.0118	0.0239	0.0281	0.0211	0.00979	0.00949
GAP	0.5796	0.8015	0.8863	0.7258	0.2181	0.1778
EQP	0.0265	0.0433	0.0720	0.0523	0.0296	0.0325
NEQ	0.0956	0.1356	0.1812	0.1399	0.0570	0.0624
GDP	0.0121	0.0231	0.0310	0.0224	0.0155	0.0150

Output 74.3.3 displays the M estimates. Besides GAP and EQP, the robust analysis also indicates that NEQ is significant. This new finding is explained by Output 74.3.4, which shows that Zambia, the 60th country in the data, is an outlier. Output 74.3.4 also identifies leverage points based on the robust MCD distances; however, there are no serious high-leverage points in this data set.

Output 74.3.3 M Estimates

Parameter Estimates

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-0.0247	0.0097	-0.0437	-0.0058	6.53	0.0106
LFG	1	0.1040	0.1867	-0.2619	0.4699	0.31	0.5775
GAP	1	0.0250	0.0086	0.0080	0.0419	8.36	0.0038
EQP	1	0.2968	0.0614	0.1764	0.4172	23.33	<.0001
NEQ	1	0.0885	0.0328	0.0242	0.1527	7.29	0.0069
Scale	1	0.0099					

Output 74.3.4 Diagnostics

Diagnostics

Obs	country	Mahalanobis Distance	Robust MCD Distance	Leverage	Standardized Robust Residual	Outlier
1	Argentin	2.6083	4.0639	*	-0.9424	
5	Botswana	3.4351	6.7391	*	1.4200	
8	Canada	3.1876	4.6843	*	-0.1972	
9	Chile	3.6752	5.0599	*	-1.8784	
17	Finland	2.6024	3.8186	*	-1.7971	
23	HongKong	2.1225	3.8238	*	1.7161	
27	Israel	2.6461	5.0336	*	0.0909	
31	Japan	2.9179	4.7140	*	0.0216	
53	Tanzania	2.2600	4.3193	*	-1.8082	
57	U.S.	3.8701	5.4874	*	0.1448	
58	Uruguay	2.5953	3.9671	*	-0.0978	
59	Venezuel	2.9239	4.1663	*	0.3573	
60	Zambia	1.8562	2.7135		-4.9798	*
61	Zimbabwe	1.9634	3.9128	*	-2.5959	

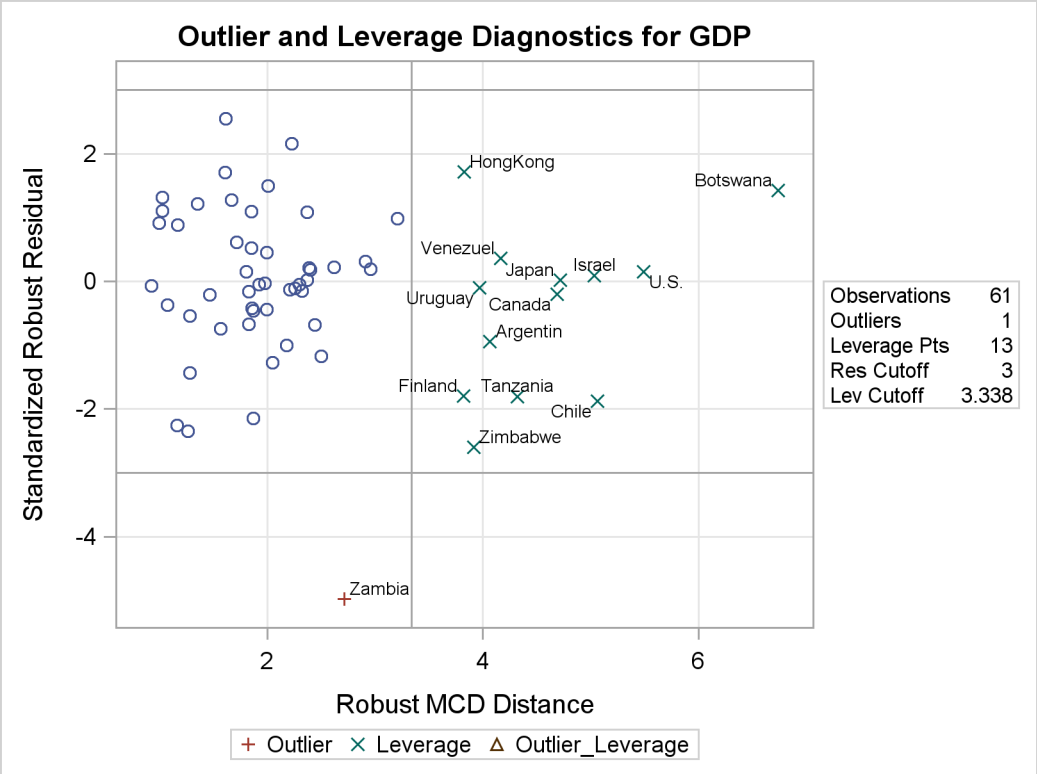
Figure 74.3.5 displays robust versions of goodness-of-fit statistics for the model.

Output 74.3.5 Goodness-of-Fit Statistics

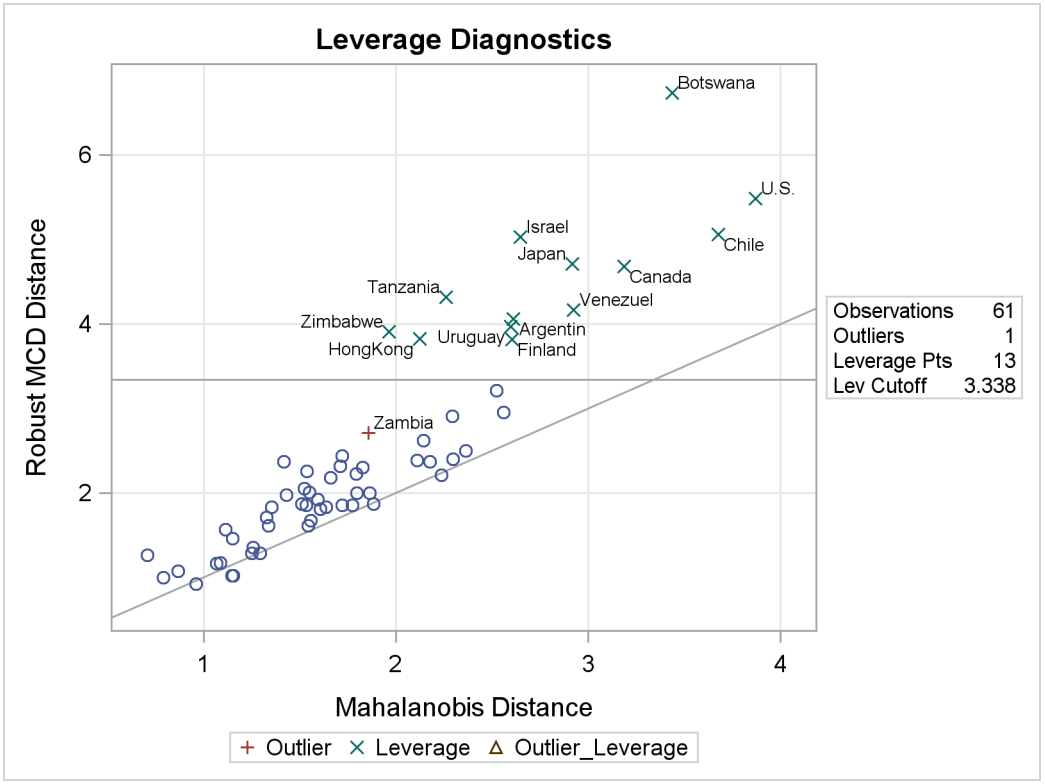
Goodness-of-Fit	
Statistic	**Value**
R-Square	0.3178
AICR	80.2134
BICR	91.5095
Deviance	0.0070

The PLOTS=ALL option generates four diagnostic plots. Figure 74.3.6 and Figure 74.3.7 are for outlier and leverage-point diagnostics. Figure 74.3.8 and Figure 74.3.9 are a histogram and a Q-Q plot of the standardized robust residuals, respectively.

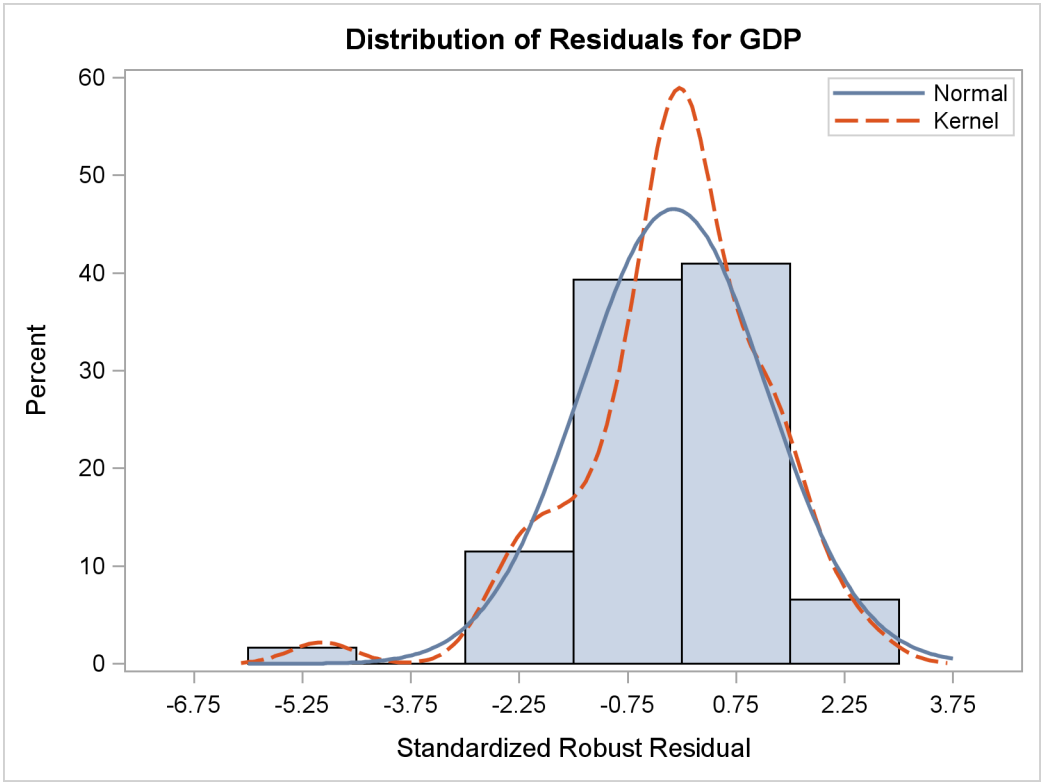
Output 74.3.6 RDPLOT for Stackloss Data



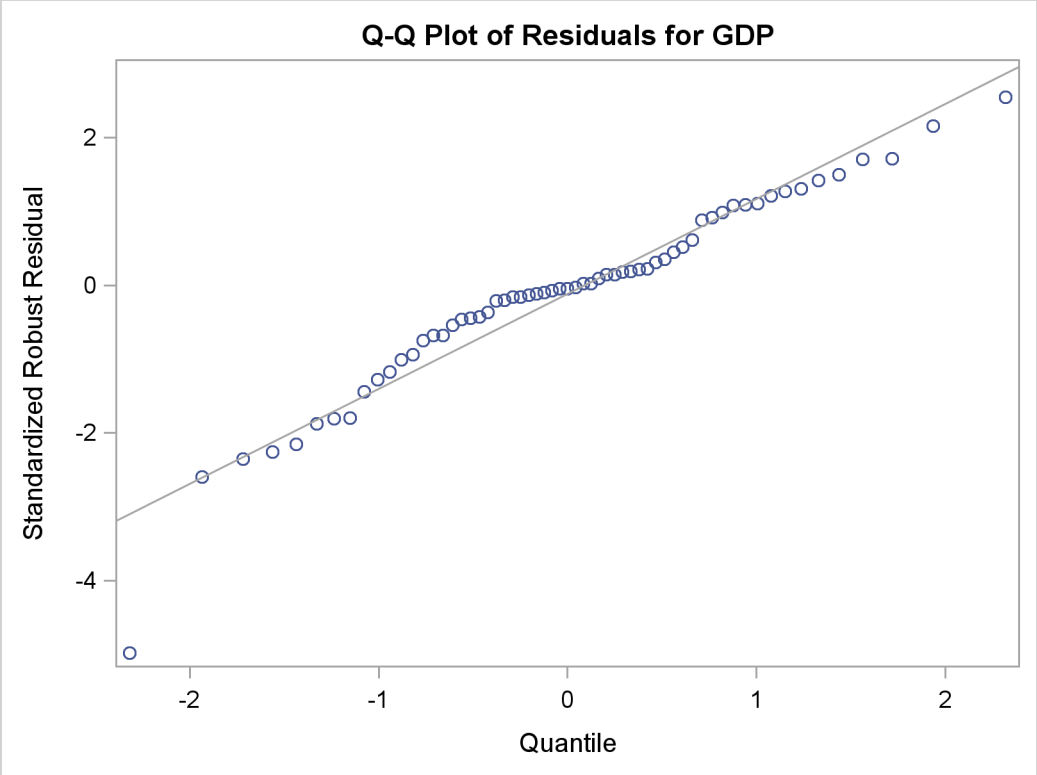
Output 74.3.7 DDPLOT for Stackloss Data



Output 74.3.8 Histogram



Output 74.3.9 Q-Q Plot



The following statements invoke the ROBUSTREG procedure with LTS estimation, which was used by Zaman, Rousseeuw, and Orhan (2001). The results are consistent with those of M estimation.

```
proc robustreg method=lts(h=33) fwls data=growth;
   model GDP  = LFG GAP EQP NEQ / diagnostics leverage ;
   id country;
run;
```

Output 74.3.10 LTS Estimates

The ROBUSTREG Procedure

LTS Parameter Estimates

Parameter	DF	Estimate
Intercept	1	-0.0249
LFG	1	0.1123
GAP	1	0.0214
EQP	1	0.2669
NEQ	1	0.1110
Scale (sLTS)	0	0.0076
Scale (Wscale)	0	0.0109

Output 74.3.10 displays the LTS estimates.

Output 74.3.11 Diagnostics and LTS R Square

Diagnostics

Obs	country	Mahalanobis Distance	Robust MCD Distance	Leverage	Standardized Robust Residual	Outlier
1	Argentin	2.6083	4.0639	*	-1.0715	
5	Botswana	3.4351	6.7391	*	1.6574	
8	Canada	3.1876	4.6843	*	-0.2324	
9	Chile	3.6752	5.0599	*	-2.0896	
17	Finland	2.6024	3.8186	*	-1.6367	
23	HongKong	2.1225	3.8238	*	1.7570	
27	Israel	2.6461	5.0336	*	0.2334	
31	Japan	2.9179	4.7140	*	0.0971	
53	Tanzania	2.2600	4.3193	*	-1.2978	
57	U.S.	3.8701	5.4874	*	0.0605	
58	Uruguay	2.5953	3.9671	*	-0.0857	
59	Venezuel	2.9239	4.1663	*	0.4113	
60	Zambia	1.8562	2.7135		-4.4984	*
61	Zimbabwe	1.9634	3.9128	*	-2.1201	

R-Square for LTS Estimation

R-Square	0.7418

Output 74.3.11 displays outlier and leverage-point diagnostics based on the LTS estimates.

Output 74.3.12 Final Weighted LS Estimates

Parameter Estimates for Final Weighted Least Squares Fit

Parameter	DF	Estimate	Standard Error	95% Confidence Limits		Chi-Square	Pr > ChiSq
Intercept	1	-0.0222	0.0093	-0.0405	-0.0039	5.65	0.0175
LFG	1	0.0446	0.1771	-0.3026	0.3917	0.06	0.8013
GAP	1	0.0245	0.0082	0.0084	0.0406	8.89	0.0029
EQP	1	0.2824	0.0581	0.1685	0.3964	23.60	<.0001
NEQ	1	0.0849	0.0314	0.0233	0.1465	7.30	0.0069
Scale	0	0.0116					

Output 74.3.12 displays the final weighted least squares estimates, which are identical to those reported in Zaman, Rousseeuw, and Orhan (2001).

References

Akaike, H. (1974), "A New Look at the Statistical Identification Model," *IEEE Trans. Automat Control*, 19, 716–723.

Brownlee, K.A. (1965), *Statistical Theory and Methodology in Science and Engineering*, 2nd ed., New York: John Wiley & Sons, Inc.

Chen, C. (2002), "Robust Regression and Outlier Detection with the ROBUSTREG Procedure," *Proceedings of the Twenty-seventh Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Chen, C. and Yin, G. (2002), "Computing the Efficiency and Tuning Constants for M-Estimation," *Proceedings of the Joint Statistical Meetings*, 478–482.

Coleman, D. Holland, P., Kaden, N., Klema, V., and Peters, S.C. (1980), "A System of Subroutines for Iteratively Reweighted Least-Squares Computations," *ACM Transactions on Mathematical Software*, 6, 327–336.

De Long, J.B. and Summers, L.H. (1991), "Equipment Investment and Economic Growth," *Quarterly Journal of Economics*, 106, 445–501.

Hampel, F. R., Ronchetti, E.M., Rousseeuw, P.J. and Stahel, W.A. (1986), *Robust Statistics: The Approach Based on Influence Functions*, New York: John Wiley & Sons, Inc.

Hawkins, D.M., Bradu, D. and Kass, G.V. (1984), "Location of Several Outliers in Multiple Regression Data Using Elemental Sets," *Technometrics*, 26, 197–208.

Holland, P. and Welsch, R. (1977), "Robust Regression Using Interactively Reweighted Least-Squares," *Commun. Statist. Theor. Meth.*, 6, 813–827.

Huber, P.J. (1973), "Robust Regression: Asymptotics, Conjectures and Monte Carlo," *Ann. Stat.*, 1, 799–821.

Huber, P.J. (1981), *Robust Statistics*, New York: John Wiley & Sons, Inc.

Marazzi, A. (1993), *Algorithm, Routines, and S Functions for Robust Statistics*, Pacific Grove, CA: Wadsworth & Brooks/Cole.

Ronchetti, E. (1985), "Robust Model Selection in Regression," *Statistics and Probability Letters*, 3, 21–23.

Rousseeuw, P.J. (1984), "Least Median of Squares Regression," *Journal of the American Statistical Association*, 79, 871–880.

Rousseeuw, P.J. and Hubert, M. (1996), "Recent Development in PROGRESS," *Computational Statistics and Data Analysis*, 21, 67–85.

Rousseeuw, P.J. and Leroy, A.M. (1987), *Robust Regression and Outlier Detection*, New York: John Wiley & Sons, Inc.

Rousseeuw, P.J. and Van Driessen, K. (1999), "A Fast Algorithm for the Minimum Covariance Determinant Estimator," *Technometrics*, 41, 212–223.

Rousseeuw, P.J. and Van Driessen, K. (2000), "An Algorithm for Positive-Breakdown Regression Based on Concentration Steps," *Data Analysis: Scientific Modeling and Practical Application*, ed. W. Gaul, O. Opitz, and M. Schader, New York: Springer-Verlag, 335–346.

Rousseeuw, P.J. and Yohai, V. (1984), "Robust Regression by Means of S Estimators," in *Robust and Nonlinear Time Series Analysis*, ed. J. Franke, W. Härdle, and R.D. Martin, Lecture Notes in Statistics, 26, New York: Springer-Verlag, 256–274.

Ruppert, D. (1992), "Computing S Estimators for Regression and Multivariate Location/Dispersion," *Journal of Computational and Graphical Statistics*, 1, 253–270.

Yohai V.J. (1987), "High Breakdown Point and High Efficiency Robust Estimates for Regression," *Annals of Statistics*, 15, 642–656.

Yohai V.J., Stahel, W.A. and Zamar, R.H. (1991), "A Procedure for Robust Estimation and Inference in Linear Regression," in Stahel, W.A. and Weisberg, S.W., eds., *Directions in Robust Statistics and Diagnostics, Part II*, New York: Springer-Verlag.

Yohai, V.J. and Zamar, R.H. (1997), "Optimal Locally Robust M Estimate of Regression," *Journal of Statist. Planning and Inference*, 64, 309–323.

Zaman, A., Rousseeuw, P.J., Orhan, M. (2001), "Econometric Applications of High-Breakdown Robust Regression Techniques," *Econometrics Letters*, 71, 1–8.

Chapter 75

The RSREG Procedure

Contents

Overview: RSREG Procedure **5720**
Comparison to Other SAS Software 5720
Terminology 5721
Getting Started: RSREG Procedure **5721**
A Response Surface with a Simple Optimum 5721
Syntax: RSREG Procedure **5726**
PROC RSREG Statement 5726
BY Statement 5730
ID Statement 5730
MODEL Statement 5731
RIDGE Statement 5733
WEIGHT Statement 5735
Details: RSREG Procedure **5735**
Introduction to Response Surface Experiments 5735
Coding the Factor Variables 5738
Missing Values 5738
Plotting the Surface 5738
Searching for Multiple Response Conditions 5739
Handling Covariates 5741
Computational Method 5742
Output Data Sets 5743
Displayed Output 5744
ODS Table Names 5746
ODS Graphics 5747
Examples: RSREG Procedure **5748**
Example 75.1: A Saddle Surface Response Using Ridge Analysis 5748
Example 75.2: Response Surface Analysis with Covariates 5752
References **5757**

Overview: RSREG Procedure

The RSREG procedure uses the method of least squares to fit quadratic response surface regression models. Response surface models are a kind of general linear model in which attention focuses on characteristics of the fit response function and in particular, where optimum estimated response values occur.

In addition to fitting a quadratic function, you can use the RSREG procedure to do the following:

- test for lack of fit
- test for the significance of individual factors
- analyze the canonical structure of the estimated response surface
- compute the ridge of optimum response
- predict new values of the response

The RSREG procedure uses ODS Graphics to display the response surfaces, residuals, fit diagnostics, and ridges of optimum response. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS."

Comparison to Other SAS Software

Other SAS/STAT procedures can be used to fit the response surface, but the RSREG procedure is more specialized. PROC RSREG uses a much more compact model syntax than other procedures; for example, the following statements model a three-factor response surface in the REG, GLM, and RSREG procedures:

```
proc reg;
   model y=x1 x1*x1
            x2 x1*x2 x2*x2
            x3 x1*x3 x2*x3 x3*x3;
run;

proc glm;
   model y=x1|x2|x3@2;
run;

proc rsreg;
   model y=x1 x2 x3;
run;
```

Additionally, PROC RSREG includes specialized methodology for analyzing the fitted response surface, such as canonical analysis and optimum response ridges.

Note that the ADX Interface in SAS/QC software provides an *interactive* environment for constructing and analyzing many different kinds of experiments, including response surface experiments. The ADX Interface is the preferred interactive SAS System tool for analyzing experiments,

since it includes facilities for checking underlying assumptions and graphically optimizing the response surface; see *Getting Started with the SAS ADX Interface* for more information. The RSREG procedure is appropriate for analyzing experiments in a batch environment.

Terminology

Variables are referred to according to the following conventions:

factor variables	independent variables used to construct the quadratic response surface. To estimate the necessary parameters, each variable must have at least three distinct values in the data. Independent variables must be numeric.
response variables	the dependent variables to which the quadratic response surfaces are fit. Dependent variables must be numeric.
covariates	additional independent variables for use in the regression but not in the formation of the quadratic response surface. Covariates must be numeric.
WEIGHT variable	a variable for weighting the observations in the regression. The WEIGHT variable must be numeric.
ID variables	variables not previously described that are transferred to an output data set containing statistics for each observation in the input data set. This data set is created by using the OUT= option in the PROC RSREG statement. ID variables can be either character or numeric.
BY variables	variables for grouping observations. Separate analyses are obtained for each BY group. BY variables can be either character or numeric.

Getting Started: RSREG Procedure

A Response Surface with a Simple Optimum

This example uses the three-factor quadratic model discussed in John (1971). Settings of the temperature, gas–liquid ratio, and packing height are controlled factors in the production of a certain chemical; Schneider and Stockett (1963) performed an experiment in order to determine the values of these three factors that minimize the unpleasant odor of the chemical. The following statements input the SAS data set smell; the variable Odor is the response, while the variables T, R, and H are the independent factors.

```
title 'Response Surface with a Simple Optimum';
data smell;
   input Odor T R H @@;
   label
      T = "Temperature"
      R = "Gas-Liquid Ratio"
      H = "Packing Height";
   datalines;
 66 40 .3 4     39 120 .3 4     43 40 .7 4     49 120 .7  4
 58 40 .5 2     17 120 .5 2     -5 40 .5 6    -40 120 .5  6
 65 80 .3 2      7  80 .7 2     43 80 .3 6    -22  80 .7  6
-31 80 .5 4    -35  80 .5 4    -26 80 .5 4
;
```

The following statements invoke PROC RSREG on the data set smell. Figure 75.1 through Figure 75.3 display the results of the analysis, including a lack-of-fit test requested with the LACKFIT option.

```
proc rsreg data=smell;
   model Odor = T R H / lackfit;
run;
```

Figure 75.1 displays the coding coefficients for the transformation of the independent variables to lie between −1 and 1, simple statistics for the response variable, hypothesis tests for linear, quadratic, and crossproduct terms, and the lack-of-fit test. The hypothesis tests can be used to gain a rough idea of importance of the effects; here the crossproduct terms are not significant. However, the lack of fit for the model is significant, so more complicated modeling or further experimentation with additional variables should be performed before firm conclusions are made concerning the underlying process.

Figure 75.1 Summary Statistics and Analysis of Variance

```
                Response Surface with a Simple Optimum

                        The RSREG Procedure

          Coding Coefficients for the Independent Variables

              Factor    Subtracted off      Divided by

              T              80.000000       40.000000
              R               0.500000        0.200000
              H               4.000000        2.000000

                 Response Surface for Variable Odor

              Response Mean                  15.200000
              Root MSE                       22.478508
              R-Square                        0.8820
              Coefficient of Variation      147.8849
```

Figure 75.1 *continued*

Regression	DF	Type I Sum of Squares	R-Square	F Value	Pr > F
Linear	3	7143.250000	0.3337	4.71	0.0641
Quadratic	3	11445	0.5346	7.55	0.0264
Crossproduct	3	293.500000	0.0137	0.19	0.8965
Total Model	9	18882	0.8820	4.15	0.0657

Residual	DF	Sum of Squares	Mean Square	F Value	Pr > F
Lack of Fit	3	2485.750000	828.583333	40.75	0.0240
Pure Error	2	40.666667	20.333333		
Total Error	5	2526.416667	505.283333		

Parameter estimates and the factor ANOVA are shown in Figure 75.2. Looking at the parameter estimates, you can see that the crossproduct terms are not significantly different from zero, as noted previously. The Estimate column contains estimates based on the raw data, and the Parameter Estimate from Coded Data column contains estimates based on the coded data. The factor ANOVA table displays tests for all four parameters corresponding to each factor—the parameters corresponding to the linear effect, the quadratic effect, and the effects of the crossproducts with each of the other two factors. The only factor with a significant overall effect is R, indicating that the level of noise left unexplained by the model is still too high to estimate the effects of T and H accurately. This might be due to the lack of fit.

Figure 75.2 Parameter Estimates and Hypothesis Tests

Parameter	DF	Estimate	Standard Error	t Value	Pr > |t|	Parameter Estimate from Coded Data
Intercept	1	568.958333	134.609816	4.23	0.0083	-30.666667
T	1	-4.102083	1.489024	-2.75	0.0401	-12.125000
R	1	-1345.833333	335.220685	-4.01	0.0102	-17.000000
H	1	-22.166667	29.780489	-0.74	0.4902	-21.375000
T*T	1	0.020052	0.007311	2.74	0.0407	32.083333
R*T	1	1.031250	1.404907	0.73	0.4959	8.250000
R*R	1	1195.833333	292.454665	4.09	0.0095	47.833333
H*T	1	0.018750	0.140491	0.13	0.8990	1.500000
H*R	1	-4.375000	28.098135	-0.16	0.8824	-1.750000
H*H	1	1.520833	2.924547	0.52	0.6252	6.083333

Factor	DF	Sum of Squares	Mean Square	F Value	Pr > F	Label
T	4	5258.016026	1314.504006	2.60	0.1613	Temperature
R	4	11045	2761.150641	5.46	0.0454	Gas-Liquid Ratio
H	4	3813.016026	953.254006	1.89	0.2510	Packing Height

Figure 75.3 displays the canonical analysis and eigenvectors. The canonical analysis indicates that the directions of principal orientation for the predicted response surface are along the axes associated with the three factors, confirming the small interaction effect in the regression ANOVA (Figure 75.1). The largest eigenvalue (48.8588) corresponds to the eigenvector $\{0.238091,\ 0.971116,\ -0.015690\}$, the largest component of which (0.971116) is associated with R; similarly, the second-largest eigenvalue (31.1035) is associated with T. The third eigenvalue (6.0377), associated with H, is quite a bit smaller than the other two, indicating that the response surface is relatively insensitive to changes in this factor. The coded form of the canonical analysis indicates that the estimated response surface is at a minimum when T and R are both near the middle of their respective ranges (that is, the coded critical values for T and R are both near 0) and H is relatively high; in uncoded terms, the model predicts that the unpleasant odor is minimized when T = 84.876502, R = 0.539915, and H = 7.541050.

Figure 75.3 Canonical Analysis and Eigenvectors

	Critical Value		
Factor	Coded	Uncoded	Label
T	0.121913	84.876502	Temperature
R	0.199575	0.539915	Gas-Liquid Ratio
H	1.770525	7.541050	Packing Height

Predicted value at stationary point: -52.024631

	Eigenvectors		
Eigenvalues	T	R	H
48.858807	0.238091	0.971116	-0.015690
31.103461	0.970696	-0.237384	0.037399
6.037732	-0.032594	0.024135	0.999177

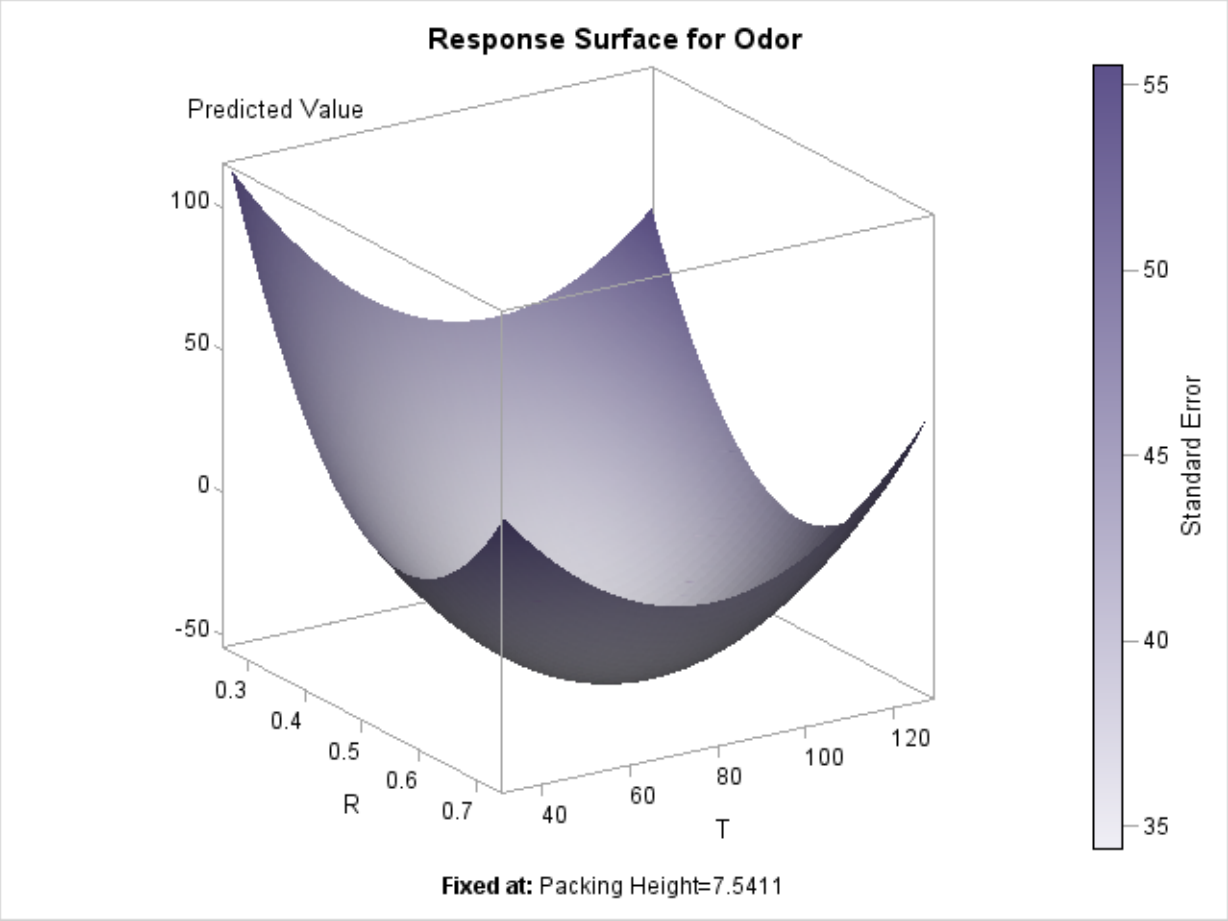
Stationary point is a minimum.

To plot the response surface with respect to two of the factor variables, fix H, the least significant factor variable, at its estimated optimum value. The following statements use ODS Graphics to display the surface:

```
ods graphics on;
proc rsreg data=smell
           plots(unpack)=surface(3d at(H=7.541050));
   model Odor = T R H;
   ods select 'T * R = Pred';
run;
ods graphics off;
```

Note that the ODS SELECT statement is specified to select the plot of interest.

Figure 75.4 The Response Surface at the Optimum H



Alternatively, the following statements produce an output data set containing the surface information, which you can then use for plotting surfaces or searching for optima. The first DATA step fixes H, the least significant factor variable, at its estimated optimum value (7.541), and generates a grid of points for T and R. To ensure that the grid data do not affect parameter estimates, the response variable (Odor) is set to missing. (See the section "Missing Values" on page 5738.) The second DATA step concatenates these grid points to the original data. Then PROC RSREG computes predictions for the combined data. The last DATA step subsets the predicted values over just the grid points, which excludes the predictions at the original data.

```
data grid;
   do;
      Odor =  .  ;
      H    = 7.541;
      do T = 20 to 140 by 5;
         do R = .1 to .9 by .05;
            output;
         end;
      end;
   end;
```

```
data grid;
   set smell grid;
run;

proc rsreg data=grid out=predict noprint;
   model Odor = T R H / predict;
run;

data grid;
   set predict;
   if H = 7.541;
run;
```

Syntax: RSREG Procedure

The following statements are available in PROC RSREG.

PROC RSREG < *options* > ;
MODEL *responses= independents* < / *options* > ;
RIDGE < *options* > ;
WEIGHT *variable* ;
ID *variables* ;
BY *variables* ;

The PROC RSREG and MODEL statements are required.

The BY, ID, MODEL, RIDGE, and WEIGHT statements are described after the PROC RSREG statement, and they can appear in any order.

PROC RSREG Statement

PROC RSREG < *options* > ;

The PROC RSREG statement invokes the procedure. You can specify the following options in the PROC RSREG statement.

DATA=*SAS-data-set*
specifies the input SAS data set that contains the data to be analyzed. By default, PROC RSREG uses the most recently created SAS data set.

NOPRINT
suppresses the normal display of results when only the output data set is required.

For more information, see the description of the NOPRINT option in the MODEL and RIDGE statements.

Note that this option temporarily disables the Output Delivery System (ODS); see Chapter 20, "Using the Output Delivery System," for more information.

OUT=*SAS-data-set*

creates an output SAS data set that contains statistics for each observation in the input data set. In particular, this data set contains the BY variables, the ID variables, the WEIGHT variable, the variables in the MODEL statement, and the output options requested in the MODEL statement. You must specify output statistic options in the MODEL statement; otherwise, the output data set is created but contains no observations. To create a permanent SAS data set, you must specify a two-level name (see *SAS Language Reference: Concepts* for more information about permanent SAS data sets). For more details, see the section "OUT=SAS-data-set" on page 5743.

PLOTS <(*global-plot-option***) >=***plot-request*< *(options)*>

PLOTS <(*global-plot-option***) >=(***plot-request*< *(options)*>< . . . *plot-request*< *(options)*> >)**

controls the plots produced through ODS Graphics. When you specify only one *plot-request*, you can omit the parentheses from around the *plot-request*. For example:

```
plots = all
plots = (diagnostics ridge surface(unpack))
plots(unpack) = surface(overlaypairs)
```

In order to produce plots, you must enable ODS Graphics and specify a *plot-request*, as shown in the following statements:

```
ods graphics on;
proc rsreg plots=all;
   model y=x;
run;
ods graphics off;
```

See Figure 75.4, Output 75.1.5, Output 75.1.6, Output 75.2.3, and Output 75.2.4 for examples of the ODS graphical displays. For general information about ODS graphics, see Chapter 21, "Statistical Graphics Using ODS."

The following *global-plot-option* is available.

UNPACKPANELS | UNPACK

suppresses paneling. By default, multiple plots can appear in some output *panels*. Specify the UNPACK option to display each plot separately.

The following *plot-requests* are available.

ALL

produces all appropriate plots. You can specify other options with ALL; for example, to display all plots and unpack the SURFACE contours you can specify `plots=(all surface(unpack))`.

DIAGNOSTICS <(LABEL | UNPACK)>

displays a panel of summary fit diagnostic plots. The plots produced and their usage are discussed in Table 75.1.

Table 75.1 Diagnostic Plots

Diagnostic Plot	Usage
Cook's *D* statistic versus observation number	Evaluate influence of an observation on the entire parameter estimate vector
Dependent variable values versus predicted values	Evaluate adequacy of fit and detect influential observations
Externally studentized residuals (RStudent) versus leverage	Detect outliers and influential (high-leverage) observations
Externally studentized residuals versus predicted values	Evaluate adequacy of fit and detect outliers
Histogram of residuals	Confirm normality of error terms
Normal quantile plot of residuals	Confirm normality and homogeneity of error terms, and detect outliers
Residuals versus predicted values	Evaluate adequacy of fit and detect outliers
Residual-fit (RF) spread plot	side-by-side quantile plots of the centered fit and the residuals show "how much variation in the data is explained by the fit and how much remains in the residuals" (Cleveland 1993)

Observations satisfying RStudent > 2 or RStudent < –2 are called *outliers*, and observations with leverage > $2p/n$ are called *influential*, where n is the number of observations used in fitting the model and p is the number of parameters used in the model (Rawlings, Pantula, and Dickey 1998). Specifying the LABEL option labels the influential and outlying observations—the label is the first ID variable if the ID statement is specified; otherwise, it is the observation number. Note in the Cook's *D* plot that only observations with *D* exceeding $4/n$ are labeled; these are also called influential observations. The UNPACK option displays each diagnostic plot separately. See Output 75.2.3 for an example of the diagnostics panel.

FIT < (GRIDSIZE=*number*) >

plots the predicted values against a single predictor when you have only one factor or only one covariate in the model. The GRIDSIZE= option specifies the number of points at which the fitted values are computed; by default, GRIDSIZE=200.

NONE

suppresses all plots.

RESIDUALS < (UNPACK | SMOOTH) >

displays plots of residuals against each factor and covariate. The UNPACK option displays each residual plot separately. The SMOOTH option overlays a loess smooth on each residual plot; see Chapter 50, "The LOESS Procedure," for more information. See Output 75.1.5 for an example of this plot.

RIDGE <(UNPACK)>
displays the maximum and/or minimum ridge plots. This option is available only when a MAXIMUM or MINIMUM option is specified in the RIDGE statement. The UNPACK option displays the estimated response and factor level ridge plots separately. See Output 75.1.5 for an example of this plot.

SURFACE <(*surface-options*)>
displays the response surface for each response variable and each pair of factors with all other factors and covariates fixed at their means. By default a panel of contour plots is produced; see Output 75.1.6 for an example of this plot. The following *surface-options* can be specified:

3D displays three-dimensional surface plots instead of contour plots. See Figure 75.4 for an example of this plot.

AT <*keyword*><(*variable=value-list* | *keyword* <...*variable=value-list* | *keyword*>)>
specifies fixed values for factors and covariates. You can specify one or more numbers in the *value-list* or one of the following *keywords*:

MIN	sets the variable to its minimum value.
MEAN	sets the variable to its mean value.
MIDRANGE	sets the variable to the middle value: $\frac{\max + \min}{2}$.
MAX	sets the variable to its maximum value.

Specifying a keyword immediately after AT sets the default value of all variables; for example, **`AT MIN`** sets all variables not displayed on an axis to their minimum values. By default, continuous variables are set to their means (**`AT MEAN`**) when they are not used on an axis. For example, if your model contains variables X1, X2, and X3, then specifying **`AT(X1=7 9)`** produces a contour plot of X2 versus X3 fixing X1 $= 7$ and then another contour plot with X1 $= 9$, along with contour plots of X1 versus X2 fixing X3 at its mean, and X1 versus X3 fixing X2 at its mean.

EXTEND=*value* extends the surface *value*-times the range of each factor in each direction, which enables you to see more of the fitted surface. For example, if factor A has range $[0, 10]$, then specifying **`EXTEND=0.1`** will compute and display the surface for A in $[-1, 11]$. You can specify $value \geq 0$; by default, $value = 0.1$.

FILL=PRED | SE | NONE produces a filled contour plot for either the predicted values or the standard errors. FILL=SE is the default. If the 3D option is also specified, then the contour plot is projected onto the surface.

GRIDSIZE=*n* creates an $n \times n$ grid of points at which the estimated values for the surface and standard errors are computed, for $n \geq 1$. By default, $n = 50$.

LINE<=PRED | SE | NONE> produces a contour line plot for either the predicted values or the standard errors. LINE=PRED is the default. If the 3D option is also specified, then specifying LINE displays a grid on the surface, and the other LINE= specifications are ignored.

NODESIGN suppresses the display of the design points on the contour surface plots and the overlaid contour-line plots.

OVERLAYPAIRS produces overlaid contour line plots for all pairs of response variables in addition to the contour surface plots. See Figure 75.6 for an example of this plot.

ROTATE=*angle* rotates the 3D surface plots *angle* degrees, –180 < *angle* < 180. By default, *angle* = 57.

TILT=*angle* tilts the 3D surface plots *angle* degrees, –180 < *angle* < 180. By default, *angle* = 20.

UNPACKPANELS | UNPACK suppresses paneling, and displays each surface plot separately.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC RSREG to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the RSREG procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

ID Statement

ID *variables* **;**

The ID statement names variables that are to be transferred to the data set created by the OUT= option in the PROC RSREG statement.

MODEL Statement

MODEL *responses=independents* < / *options* > **;**

In the MODEL statement, you specify the response (dependent) variables followed by an equal sign and then the independent variables, some of which can be covariates.

Table 75.2 summarizes the *options* available in the MODEL statement. The *statistic options* specify which statistics are output to the OUT= data set. If none of the *statistic options* are selected, the data set is created but contains no observations. The *statistic option* keywords become values of the special variable _TYPE_ in the output data set.

Table 75.2 MODEL Statement Options

Task	Options
Analyze original data	NOCODE
Fit model to first BY group only	BYOUT
Declare covariates	COVAR=
Request additional statistics	PRESS
Request additional tests	LACKFIT
Suppress displayed output	NOANOVA
	NOOPTIMAL
	NOPRINT
Task	**Statistic Options**
Output statistics	ACTUAL
	PREDICT
	RESIDUAL
	L95
	U95
	L95M
	U95M
	D

The following list describes these options in alphabetical order.

ACTUAL
specifies that the observed response values from the input data set be written to the output data set.

BYOUT
uses only the first BY group to estimate the model. Subsequent BY groups have scoring statistics computed in the output data set only. The BYOUT option is used only when a BY statement is specified.

COVAR=*n*

declares that the first *n* variables on the right side of the model are simple linear regressors (covariates) and not factors in the quadratic response surface. By default, PROC RSREG forms quadratic and crossproduct effects for all regressor variables in the MODEL statement.

See the section "Handling Covariates" on page 5741 for more details and Example 75.2 for an example that uses covariates.

D

specifies that Cook's *D* influence statistic be written to the output data set.

See Chapter 4, "Introduction to Regression Procedures," for details and formulas.

LACKFIT

performs a lack-of-fit test.

See Draper and Smith (1981) for a discussion of lack-of-fit tests.

L95

specifies that the lower bound of a 95% confidence interval for an individual predicted value be written to the output data set. The variance used in calculating this bound is a function of both the mean square error and the variance of the parameter estimates.

See Chapter 4, "Introduction to Regression Procedures," for details and formulas.

L95M

specifies that the lower bound of a 95% confidence interval for the expected value of the dependent variable be written to the output data set. The variance used in calculating this bound is a function of the variance of the parameter estimates.

See Chapter 4, "Introduction to Regression Procedures," for details and formulas.

NOANOVA

NOAOV

suppresses the display of the analysis of variance and parameter estimates from the model fit.

NOCODE

performs the canonical and ridge analyses with the parameter estimates derived from fitting the response to the original values of the factor variables, rather than their coded values (see the section "Coding the Factor Variables" on page 5738 for more details). Use this option if the data are already stored in a coded form.

NOOPTIMAL

NOOPT

suppresses the display of the canonical analysis for the quadratic response surface.

NOPRINT

suppresses the display of both the analysis of variance and the canonical analysis.

PREDICT
specifies that the values predicted by the model be written to the output data set.

PRESS
computes and displays the predicted residual sum of squares (PRESS) statistic for each dependent variable in the model. The PRESS statistic is added to the summary information at the beginning of the analysis of variance, so if the NOANOVA or NOPRINT option is specified, then the PRESS option has no effect.

See Chapter 4, "Introduction to Regression Procedures," for details and formulas.

RESIDUAL
specifies that the residuals, calculated as ACTUAL − PREDICTED, be written to the output data set.

U95
specifies that the upper bound of a 95% confidence interval for an individual predicted value be written to the output data set. The variance used in calculating this bound is a function of both the mean square error and the variance of the parameter estimates.

See Chapter 4, "Introduction to Regression Procedures," for details and formulas.

U95M
specifies that the upper bound of a 95% confidence interval for the expected value of the dependent variable be written to the output data set. The variance used in calculating this bound is a function of the variance of the parameter estimates.

See Chapter 4, "Introduction to Regression Procedures," for details and formulas.

RIDGE Statement

RIDGE < *options* > **;**

A RIDGE statement computes the ridge of optimum response. The ridge starts at a given point $\mathbf{x}_0$, and the point on the ridge at radius r from $\mathbf{x}_0$ is the collection of factor settings that optimizes the predicted response at this radius. You can think of the ridge as climbing or falling as fast as possible on the surface of predicted response. Thus, the ridge analysis can be used as a tool to help interpret an existing response surface or to indicate the direction in which further experimentation should be performed.

The default starting point, $\mathbf{x}_0$, has each coordinate equal to the point midway between the highest and lowest values of the factor in the design. The default radii at which the ridge is computed are 0, 0.1, . . ., 0.9, 1. If the ridge analysis is based on the response surface fit to coded values for the factor variables (see the section "Coding the Factor Variables" on page 5738 for details), then this results in a ridge that starts at the point with a coded zero value for each coordinate and extends toward, but not beyond, the edge of the range of experimentation. Alternatively, both the center point of the ridge and the radii at which it is to be computed can be specified.

You can specify the following options in the RIDGE statement:

CENTER=*uncoded-factor-values*
gives the coordinates of the point $\mathbf{x}_0$ from which to begin the ridge. The coordinates should be given in the original (uncoded) factor variable values and should be separated by commas. There must be as many coordinates specified as there are factors in the model, and the order of the coordinates must be the same as that used in the MODEL statement. This starting point should be well inside the range of experimentation. The default sets each coordinate equal to the value midway between the highest and lowest values for the associated factor.

MAXIMUM
MAX
computes the ridge of maximum response. Both the MIN and MAX options can be specified; at least one must be specified.

MINIMUM
MIN
computes the ridge of minimum response. Both the MIN and MAX options can be specified; at least one must be specified.

NOPRINT
suppresses the display of the ridge analysis when only an output data set is required.

OUTR=*SAS-data-set*
creates an output SAS data set containing the computed optimum ridge.

For details, see the section "OUTR=SAS-data-set" on page 5743.

RADIUS=*coded-radii*
gives the distances from the ridge starting point at which to compute the optima. The values in the list represent distances between coded points. The list can take any of the following forms or can be composed of mixtures of them:

$m_1, m_2, \ldots, m_n$	specifies several values.
m TO n	specifies a sequence where m equals the starting value, n equals the ending value, and the increment equals 1.
m TO n BY i	specifies a sequence where m equals the starting value, n equals the ending value, and i equals the increment.

Mixtures of the preceding forms should be separated by commas. The default list runs from 0 to 1 by increments of 0.1. The following are examples of valid lists.

```
radius=0 to 5 by .5;
radius=0, .2, .25, .3, .5 to 1.0 by .1;
```

WEIGHT Statement

WEIGHT *variable* **;**

When a WEIGHT statement is specified, a weighted residual sum of squares

$$\sum_i w_i (y_i - \hat{y}_i)^2$$

is minimized, where w_i is the value of the variable specified in the WEIGHT statement, y_i is the observed value of the response variable, and $\hat{y}_i$ is the predicted value of the response variable.

The observation is used in the analysis only if the value of the WEIGHT statement variable is greater than zero. The WEIGHT statement has no effect on degrees of freedom or number of observations. If the weights for the observations are proportional to the reciprocals of the error variances, then the weighted least squares estimates are best linear unbiased estimators (BLUE).

Details: RSREG Procedure

Introduction to Response Surface Experiments

Many industrial experiments are conducted to discover which values of given factor variables optimize a response. If each factor is measured at three or more values, a quadratic response surface can be estimated by least squares regression. The predicted optimal value can be found from the estimated surface if the surface is shaped like a simple hill or valley. If the estimated surface is more complicated, or if the predicted optimum is far from the region of experimentation, then the shape of the surface can be analyzed to indicate the directions in which new experiments should be performed.

Suppose that a response variable y is measured at combinations of values of two factor variables, x_1 and x_2. The quadratic response surface model for this variable is written as

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_1^2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 + \epsilon$$

The steps in the analysis for such data are as follows:

1. model fitting and analysis of variance, including lack-of-fit testing, to estimate parameters
2. canonical analysis to investigate the shape of the predicted response surface
3. ridge analysis to search for the region of optimum response

Model Fitting and Analysis of Variance

The first task in analyzing the response surface is to estimate the parameters of the model by least squares regression and to obtain information about the fit in the form of an analysis of variance. The estimated surface is typically curved: a *hill* with the peak occurring at the unique estimated point of maximum response, a *valley,* or a *saddle surface* with no unique minimum or maximum. Use the results of this phase of the analysis to answer the following questions:

- What is the contribution of each type of effect—linear, quadratic, and crossproduct—to the statistical fit? The ANOVA table with sources labeled "Regression" addresses this question.
- What part of the residual error is due to lack of fit? Does the quadratic response model adequately represent the true response surface? If you specify the LACKFIT option in the MODEL statement, then the ANOVA table with sources labeled "Residual" addresses this question. See the section "Lack-of-Fit Test" on page 5736 for details.
- What is the contribution of each factor variable to the statistical fit? Can the response be predicted accurately if the variable is removed? The ANOVA table with sources labeled "Factor" addresses this question.
- What are the predicted responses for a grid of factor values? (See the section "Plotting the Surface" on page 5738 and the section "Searching for Multiple Response Conditions" on page 5739.)

Lack-of-Fit Test

The lack-of-fit test compares the variation around the model with *pure* variation within replicated observations. This measures the adequacy of the quadratic response surface model. In particular, if there are n_i replicated observations $Y_{i1}, \ldots, Y_{in_i}$ of the response all at the same values $\mathbf{x}_i$ of the factors, then you can predict the true response at $\mathbf{x}_i$ either by using the predicted value $\hat{Y}_i$ based on the model or by using the mean $\bar{Y}_i$ of the replicated values. The lack-of-fit test decomposes the residual error into a component due to the variation of the replications around their mean value (the *pure* error) and a component due to the variation of the mean values around the model prediction (the *bias* error):

$$\sum_i \sum_{j=1}^{n_i} \left(Y_{ij} - \hat{Y}_i\right)^2 = \sum_i \sum_{j=1}^{n_i} \left(Y_{ij} - \bar{Y}_i\right)^2 + \sum_i n_i \left(\bar{Y}_i - \hat{Y}_i\right)^2$$

If the model is adequate, then both components estimate the nominal level of error; however, if the bias component of error is much larger than the pure error, then this constitutes evidence that there is significant lack of fit.

If some observations in your design are replicated, you can test for lack of fit by specifying the LACKFIT option in the MODEL statement. Note that, since all other tests use total error rather than pure error, you might want to hand-calculate the tests with respect to pure error if the lack of fit is significant. On the other hand, significant lack of fit indicates that the quadratic model is inadequate, so if this is a problem you can also try to refine the model, possibly by using PROC GLM for general

polynomial modeling; see Chapter 39, "The GLM Procedure," for more information. Example 75.1 illustrates the use of the LACKFIT option.

Canonical Analysis

The second task in analyzing the response surface is to examine the overall shape of the curve and determine whether the estimated stationary point is a maximum, a minimum, or a saddle point. The canonical analysis can be used to answer the following questions:

- Is the surface shaped like a hill, a valley, or a saddle, or is it flat?
- If there is a unique optimum combination of factor values, where is it?
- To which factor or factors are the predicted responses most sensitive?

The eigenvalues and eigenvectors in the matrix of second-order parameters characterize the shape of the response surface. The eigenvectors point in the directions of principal orientation for the surface, and the signs and magnitudes of the associated eigenvalues give the shape of the surface in these directions. Positive eigenvalues indicate directions of upward curvature, and negative eigenvalues indicate directions of downward curvature. The larger an eigenvalue is in absolute value, the more pronounced is the curvature of the response surface in the associated direction. Often, all the coefficients of an eigenvector except for one are relatively small, indicating that the vector points roughly along the axis associated with the factor corresponding to the single large coefficient. In this case, the canonical analysis can be used to determine the relative sensitivity of the predicted response surface to variations in that factor. (See the section "Getting Started: RSREG Procedure" on page 5721 for an example.)

Ridge Analysis

If the estimated surface is found to have a simple optimum well within the range of experimentation, the analysis performed by the preceding two steps might be sufficient. In more complicated situations, further search for the region of optimum response is required. The method of ridge analysis computes the estimated ridge of optimum response for increasing radii from the center of the original design. The ridge analysis answers the following question:

- If there is not a unique optimum of the response surface within the range of experimentation, in which direction should further searching be done in order to locate the optimum?

You can use the RIDGE statement to compute the ridge of maximum or minimum response.

Coding the Factor Variables

For the results of the canonical and ridge analyses to be interpretable, the values of different factor variables should be comparable. This is because the canonical and ridge analyses of the response surface are not invariant with respect to differences in scale and location of the factor variables. The analysis of variance is not affected by these changes. Although the actual predicted surface does not change, its parameterization does. The usual solution to this problem is to code each factor variable so that its minimum in the experiment is -1 and its maximum is 1 and to carry through the analysis with the coded values instead of the original ones. This practice has the added benefit of making 1 a reasonable boundary radius for the ridge analysis since 1 represents approximately the edge of the experimental region. By default, PROC RSREG computes the linear transformation to perform this coding as the data are initially read in, and the canonical and ridge analyses are performed on the model fit to the coded data. The actual form of the coding operation for each value of a variable is

$$coded\ value = (original\ value - M)/S$$

where M is the average of the highest and lowest values for the variable in the design and S is half their difference.

Missing Values

If an observation has missing data for any of the variables used by the procedure, then that observation is not used in the estimation process. If one or more response variables are missing, but no factor or covariate variables are missing, then predicted values and confidence limits are computed for the output data set, but the residual and Cook's D statistic are missing.

Plotting the Surface

Specifying the PLOTS=SURFACE option in the PROC RSREG statement displays contour plots for all pairs of factors in the model (see Example 75.1), while specifying the PLOTS=SURFACE(3D) option displays a three-dimensional surface as shown in Figure 75.4.

You can also generate predicted values for a grid of points with the PREDICT option (see the section "Getting Started: RSREG Procedure" on page 5721 for an example) and then use these values to create a contour plot or a three-dimensional plot of the response surface over a two-dimensional grid. Any two factor variables can be chosen to form the grid for the plot. Several plots can be generated by using different pairs of factor variables.

Searching for Multiple Response Conditions

Suppose you have the following data with two factors and three responses, and you want to find the factor setting that produces responses in a certain region:

```
data a;
   input x1 x2 y1 y2 y3;
   datalines;
-1      -1         1.8 1.940  3.6398
-1       1         2.6 1.843  4.9123
 1      -1         5.4 1.063  6.0128
 1       1         0.7 1.639  2.3629
 0       0         8.5 0.134  9.0910
 0       0         3.0 0.545  3.7349
 0       0         9.8 0.453 10.4412
 0       0         4.1 1.117  5.0042
 0       0         4.8 1.690  6.6245
 0       0         5.9 1.165  6.9420
 0       0         7.3 1.013  8.7442
 0       0         9.3 1.179 10.2762
 1.4142  0         3.9 0.945  5.0245
-1.4142  0         1.7 0.333  2.4041
 0       1.4142    3.0 1.869  5.2695
 0      -1.4142    5.7 0.099  5.4346
;
```

You want to find the values of x1 and x2 that maximize y1 subject to y2<2 and y3<y2+y1. The exact answer is not easy to obtain analytically, but you can obtain a practically feasible solution by checking conditions across a grid of values in the range of interest. First, append a grid of factor values to the observed data, with missing values for the responses:

```
data b;
   set a end=eof;
   output;
   if eof then do;
      y1=.;
      y2=.;
      y3=.;
      do x1=-2 to 2 by .1;
         do x2=-2 to 2 by .1;
            output;
         end;
      end;
   end;
run;
```

Next, use PROC RSREG to fit a response surface model to the data and to compute predicted values for both the observed data and the grid, putting the predicted values in a data set c:

```
proc rsreg data=b out=c;
   model y1 y2 y3=x1 x2 / predict;
run;
```

Finally, find the subset of predicted values that satisfy the constraints, sort by the unconstrained variable, and display the top five predictions:

```
data d;
   set c;
   if y2<2;
   if y3<y2+y1;

proc sort data=d;
   by descending y1;
run;

data d; set d;
   if (_n_ <= 5);
proc print;
run;
```

The results are displayed in Figure 75.5. They indicate that optimal values of the factors are around 0.3 for x1 and around –0.5 for x2.

Figure 75.5 Top Five Predictions

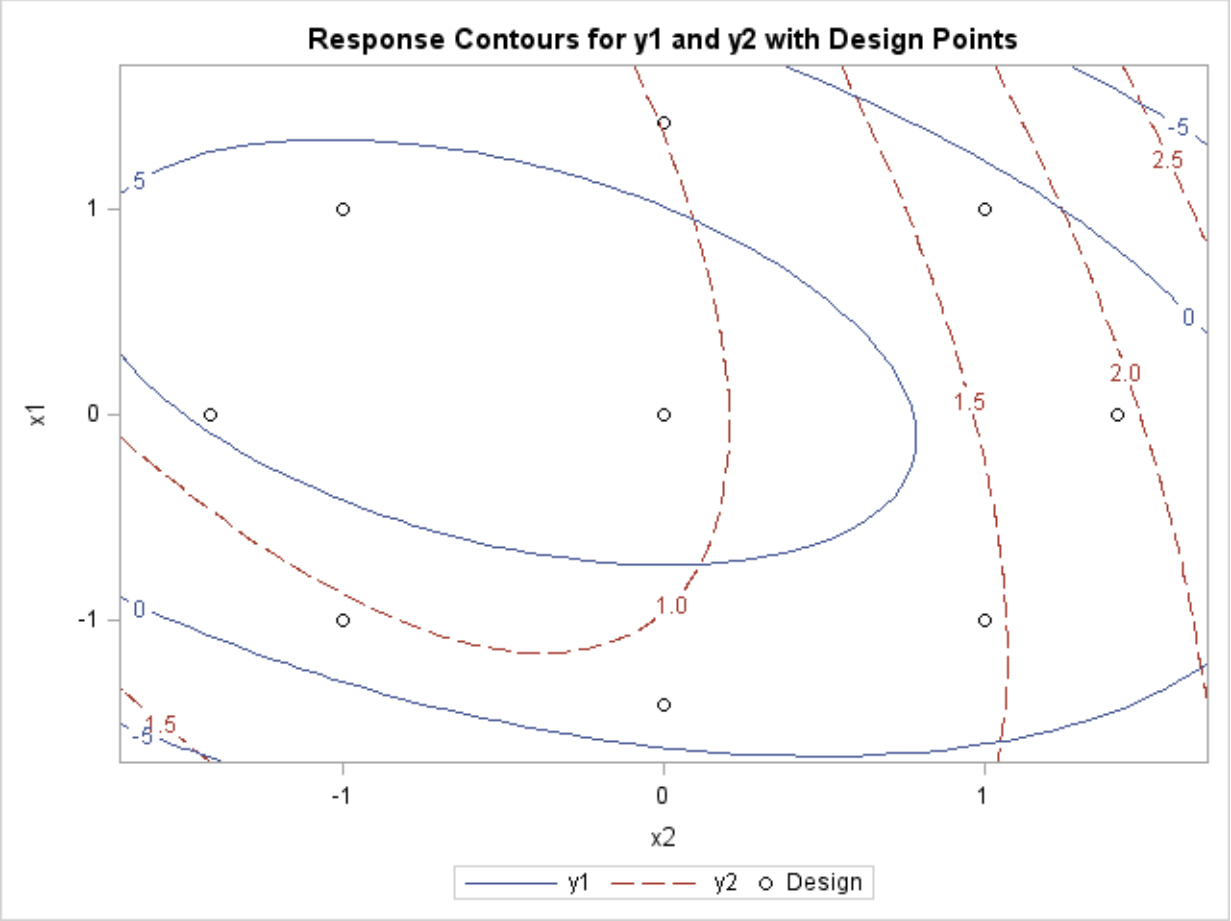
Obs	x1	x2	_TYPE_	y1	y2	y3
1	0.3	−0.5	PREDICT	6.92570	0.75784	7.60471
2	0.3	−0.6	PREDICT	6.91424	0.74174	7.54194
3	0.3	−0.4	PREDICT	6.91003	0.77870	7.64341
4	0.4	−0.6	PREDICT	6.90769	0.73357	7.51836
5	0.4	−0.5	PREDICT	6.90540	0.75135	7.56883

If you are also interested in simultaneously optimizing y1 and y2, you can specify the following statements to make a visual comparison of the two response surfaces by overlaying their contour plots:

```
ods graphics on;
proc rsreg data=a plots=surface(overlaypairs);
   model y1 y2=x1 x2;
run;
ods graphics off;
```

Figure 75.6 shows that you have to make some compromises in any attempt to maximize both y1 and y2; however, you might be able to maximize y1 while minimizing y2.

Figure 75.6 Overlaid Line Contours of Predicted Responses



Handling Covariates

Covariate regressors are added to a response surface model because they are believed to account for a sizable yet relatively uninteresting portion of the variation in the data. What the experimenter is really interested in is the response corrected for the effect of the covariates. A common example is the block effect in a block design. In the canonical and ridge analyses of a response surface, which estimate responses at hypothetical levels of the factor variables, the actual value of the predicted response is computed by using the average values of the covariates. The estimated response values do optimize the estimated surface of the response corrected for covariates, but true prediction of the response requires actual values for the covariates. You can use the COVAR= option in the MODEL statement to include covariates in the response surface model. Example 75.2 illustrates the use of this option.

Computational Method

Canonical Analysis

For each response variable, the model can be written in the form

$$y_i = \mathbf{x}_i' \mathbf{A} \mathbf{x}_i + \mathbf{b}' \mathbf{x}_i + \mathbf{c}' \mathbf{z}_i + \epsilon_i$$

where

y_i is the ith observation of the response variable.

$\mathbf{x}_i$ $= (x_{i1}, x_{i2}, \ldots, x_{ik})'$ are the k factor variables for the ith observation.

$\mathbf{z}_i$ $= (z_{i1}, z_{i2}, \ldots, z_{iL})'$ are the L covariates, including the intercept term.

$\mathbf{A}$ is the $k \times k$ symmetrized matrix of quadratic parameters, with diagonal elements equal to the coefficients of the pure quadratic terms in the model and off-diagonal elements equal to half the coefficient of the corresponding crossproduct.

$\mathbf{b}$ is the $k \times 1$ vector of linear parameters.

$\mathbf{c}$ is the $L \times 1$ vector of covariate parameters, one of which is the intercept.

ϵ_i is the error associated with the ith observation. Tests performed by PROC RSREG assume that errors are independently and normally distributed with mean zero and variance σ^2.

The parameters in $\mathbf{A}$, $\mathbf{b}$, and $\mathbf{c}$ are estimated by least squares. To optimize $\mathbf{y}$ with respect to $\mathbf{x}$, take partial derivatives, set them to zero, and solve:

$$\frac{\partial y}{\partial \mathbf{x}} = 2\mathbf{x}'\mathbf{A} + \mathbf{b}' = \mathbf{0} \quad \Longrightarrow \quad \mathbf{x} = -\frac{1}{2}\mathbf{A}^{-1}\mathbf{b}$$

You can determine if the solution is a maximum or minimum by looking at the eigenvalues of $\mathbf{A}$:

If the eigenvalues...	then the solution is...
are all negative	a maximum
are all positive	a minimum
have mixed signs	a saddle point
contain zeros	in a flat area

Ridge Analysis

If the largest eigenvalue is positive, its eigenvector gives the direction of steepest ascent from the stationary point; if the largest eigenvalue is negative, its eigenvector gives the direction of steepest descent. The eigenvectors corresponding to small or zero eigenvalues point in directions of relative flatness.

The point on the optimum response ridge at a given radius R from the ridge origin is found by optimizing

$$(\mathbf{x}_0 + \mathbf{d})'\mathbf{A}(\mathbf{x}_0 + \mathbf{d}) + \mathbf{b}'(\mathbf{x}_0 + \mathbf{d})$$

over $\mathbf{d}$ satisfying $\mathbf{d}'\mathbf{d} = R^2$, where $\mathbf{x}_0$ is the $k \times 1$ vector containing the ridge origin and $\mathbf{A}$ and $\mathbf{b}$ are as previously discussed. By the method of Lagrange multipliers, the optimal $\mathbf{d}$ has the form

$$\mathbf{d} = -(\mathbf{A} - \mu\mathbf{I})^{-1}(\mathbf{A}\mathbf{x}_0 + 0.5\mathbf{b})$$

where $\mathbf{I}$ is the $k \times k$ identity matrix and μ is chosen so that $\mathbf{d}'\mathbf{d} = R^2$. There can be several values of μ that satisfy this constraint; the correct one depends on which sort of response ridge is of interest. If you are searching for the ridge of maximum response, then the appropriate μ is the unique one that satisfies the constraint and is greater than all the eigenvalues of $\mathbf{A}$. Similarly, the appropriate μ for the ridge of minimum response satisfies the constraint and is less than all the eigenvalues of $\mathbf{A}$. (See Myers and Montgomery (1995) for details.)

Output Data Sets

OUT=SAS-data-set

An output data set containing statistics requested with options in the MODEL statement for each observation in the input data set is created whenever the OUT= option is specified in the PROC RSREG statement. The data set contains the following variables:

- the BY variables
- the ID variables
- the WEIGHT variable
- the independent variables in the MODEL statement
- the variable _TYPE_, which identifies the observation type in the output data set. _TYPE_ is a character variable with a length of eight, and it takes on the values 'ACTUAL', 'PREDICT', 'RESIDUAL', 'U95M', 'L95M', 'U95', 'L95', and 'D', corresponding to the options specified.
- the response variables containing special output values identified by the _TYPE_ variable

All confidence limits use the two-tailed Student's t value.

OUTR=SAS-data-set

An output data set containing the optimum response ridge is created when the OUTR= option is specified in the RIDGE statement. The data set contains the following variables:

- the current values of the BY variables
- a character variable _DEPVAR_ containing the name of the dependent variable

- a character variable _TYPE_ identifying the type of ridge being computed, MINIMUM or MAXIMUM. If both MAXIMUM and MINIMUM are specified, the data set contains observations for the minimum ridge followed by observations for the maximum ridge.
- a numeric variable _RADIUS_ giving the distance from the ridge starting point
- the values of the model factors at the estimated optimum point at distance _RADIUS_ from the ridge starting point
- a numeric variable _PRED_, which is the estimated expected value of the dependent variable at the optimum
- a numeric variable _STDERR_, which is the standard error of the estimated expected value

Displayed Output

All estimates and hypothesis tests assume that the model is correctly specified and the errors are distributed according to classical statistical assumptions.

The output displayed by PROC RSREG includes the following.

Estimation and Analysis of Variance

- The actual form of the coding operation for each value of a variable is

$$coded\ value = \frac{1}{S}(original\ value - M)$$

  where M is the average of the highest and lowest values for the variable in the design and S is half their difference. The Subtracted off column contains the M values for this formula for each factor variable, and S is found in the Divided by column.

- The summary table for the response variable contains the following information.
  - "Response Mean" is the mean of the response variable in the sample. When a WEIGHT statement is specified, the mean $\bar{y}$ is calculated by

$$\bar{y} = \frac{\sum_i w_i y_i}{\sum_i w_i}$$

  - "Root MSE" estimates the standard deviation of the response variable and is calculated as the square root of the "Total Error" mean square.
  - The "R-Square" value is R^2, or the coefficient of determination. R^2 measures the proportion of the variation in the response that is attributed to the model rather than to random error.
  - The "Coefficient of Variation" is 100 times the ratio of the "Root MSE" to the "Response Mean."

- A table analyzing the significance of the terms of the regression is displayed. Terms are brought into the regression in four steps: (1) the "Intercept" and any covariates in the model, (2) "Linear" terms like X1 and X2, (3) pure "Quadratic" terms like X1*X1 or X2*X2, and (4) "Crossproduct" terms like X1*X2. The table displays the following information:
  - the degrees of freedom in the DF column, which should be the same as the number of corresponding parameters unless one or more of the parameters are not estimable
  - Type I Sum of Squares, also called the sequential sums of squares, which measures the reduction in the error sum of squares as sets of terms (Linear, Quadratic, and so forth) are added to the model
  - R-Square, which measures the portion of total R^2 contributed as each set of terms (Linear, Quadratic, and so forth) is added to the model
  - F Value, which tests the null hypothesis that all parameters in the term are zero by using the Total Error mean square as the denominator. This is a test of a Type I hypothesis, containing the usual F test numerator, conditional on the effects of subsequent variables not being in the model.
  - Pr > F, which is the significance value or probability of obtaining at least as great an F ratio given that the null hypothesis is true.
- The Sum of Squares column partitions the "Total Error" into "Lack of Fit" and "Pure Error." When "Lack of Fit" is significant, there is variation around the model other than random error (such as cubic effects of the factor variables).
  - The "Total Error" Mean Square estimates σ^2, the variance.
  - F Value tests the null hypothesis that the variation is adequately described by random error.
- A table containing the parameter estimates from the model is displayed.
  - The Estimate column contains the parameter estimates based on the *uncoded* values of the factor variables. If an effect is a linear combination of previous effects, the parameter for the effect is not estimable. When this happens, the degrees of freedom are zero, the parameter estimate is set to zero, and estimates and tests on other parameters are conditional on this parameter being zero.
  - The Standard Error column contains the estimated standard deviations of the parameter estimates based on *uncoded* data.
  - The t Value column contains t values of a test of the null hypothesis that the true parameter is zero when the *uncoded* values of the factor variables are used.
  - The Pr > |T| column gives the significance value or probability of a greater absolute t ratio given that the true parameter is zero.
  - The Parameter Estimate from Coded Data column contains the parameter estimates based on the *coded* values of the factor variables. These are the estimates used in the subsequent canonical and ridge analyses.
- The sum of squares are partitioned by the factors in the model, and an analysis table is displayed. The test on a factor is a joint test on all the parameters involving that factor. For example, the test for the factor X1 tests the null hypothesis that the true parameters for X1, X1*X1, and X1*X2 are all zero.

Canonical Analysis

- The Critical Value columns contain the values of the factor variables that correspond to the stationary point of the fitted response surface. The critical values can be at a minimum, maximum, or saddle point.
- The eigenvalues and eigenvectors are from the matrix of quadratic parameter estimates based on the coded data. They characterize the shape of the response surface.

Ridge Analysis

- The Coded Radius column contains the distance from the coded version of the associated point to the coded version of the origin of the ridge. The origin is given by the point at radius zero.
- The Estimated Response column contains the estimated value of the response variable at the associated point. The standard error of this estimate is also given. This quantity is useful for assessing the relative credibility of the prediction at a given radius. Typically, this standard error increases rapidly as the ridge moves up to and beyond the design perimeter, reflecting the inherent difficulty of making predictions beyond the range of experimentation.
- The Uncoded Factor Values columns contain the values of the uncoded factor variables that give the optimum response at this radius from the ridge origin.

ODS Table Names

PROC RSREG assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 75.3. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 75.3 ODS Tables Produced by PROC RSREG

ODS Table Name	Description	Statement
Coding	Coding coefficients for the independent variables	default
ErrorANOVA	Error analysis of variance	default
FactorANOVA	Factor analysis of variance	default
FitStatistics	Overall statistics for fit	default
ModelANOVA	Model analysis of variance	default
ParameterEstimates	Estimated linear parameters	default
Ridge	Ridge analysis for optimum response	RIDGE
Spectral	Spectral analysis	default
StationaryPoint	Stationary point of response surface	default

ODS Graphics

PROC RSREG assigns a name to each graph it creates using ODS. You can use these names to reference the graphs when using ODS. The names are listed in Table 75.4.

To request these graphs you must specify the ODS GRAPHICS statement in addition to the PLOTS= option and any other options indicated in Table 75.4. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Table 75.4 ODS Graphics Produced by PROC RSREG

ODS Graph Name	Plot Description	PLOTS= Option
FitPlot	Fit plot for 1 predictor	FIT
DiagnosticsPanel	Panel of fit diagnostics	DIAGNOSTICS
CooksDPlot	Cook's *D* plot	DIAGNOSTICS(UNPACK)
ObservedByPredicted	Observed by predicted	DIAGNOSTICS(UNPACK)
QQPlot	Residual Q-Q plot	DIAGNOSTICS(UNPACK)
ResidualByPredicted	Residual by predicted values	DIAGNOSTICS(UNPACK)
ResidualHistogram	Residual histogram	DIAGNOSTICS(UNPACK)
RFPlot	RF plot	DIAGNOSTICS(UNPACK)
RStudentByPredicted	Studentized residuals by predicted	DIAGNOSTICS(UNPACK)
RStudentByLeverage	RStudent by hat diagonals	DIAGNOSTICS(UNPACK)
ResidualPlots	Panel of residuals by predictors	RESIDUALS
	Residuals by predictors	RESIDUALS(UNPACK)
RidgePlots	Panel of ridge plot and factors	RIDGE (with RIDGE MAX or MIN)
	Ridge plot	RIDGE(UNPACK) (with RIDGE MAX or MIN)
	Ridge factors	RIDGE(UNPACK) (with RIDGE MAX or MIN)
Contour	Panel of contour plots	SURFACE
	Contour plots	SURFACE(UNPACK)
Surface	Panel of 3D surface plots	SURFACE(3D)
	3D surface plots	SURFACE(3D UNPACK)
ContourOverlay	Panel of overlaid line-contour plots	SURFACE(OVERLAYPAIRS)
	Overlaid line-contour plots	SURFACE(OVERLAYPAIRS UNPACK)

Examples: RSREG Procedure

Example 75.1: A Saddle Surface Response Using Ridge Analysis

Myers (1976) analyzes an experiment reported by Frankel (1961) aimed at maximizing the yield of mercaptobenzothiazole (MBT) by varying processing time and temperature. Myers (1976) uses a two-factor model in which the estimated surface does not have a unique optimum. A ridge analysis is used to determine the region in which the optimum lies. The objective is to find the settings of time and temperature in the processing of a chemical that maximize the yield. The following statements produce Output 75.1.1 through Output 75.1.5:

```
data d;
   input Time Temp MBT;
   label Time = "Reaction Time (Hours)"
         Temp = "Temperature (Degrees Centigrade)"
         MBT  = "Percent Yield Mercaptobenzothiazole";
   datalines;
 4.0   250   83.8
20.0   250   81.7
12.0   250   82.4
12.0   250   82.9
12.0   220   84.7
12.0   280   57.9
12.0   250   81.2
 6.3   229   81.3
 6.3   271   83.1
17.7   229   85.3
17.7   271   72.7
 4.0   250   82.0
;
ods graphics on;
proc rsreg data=d plots=(ridge surface);
   model MBT=Time Temp / lackfit;
   ridge max;
run;
ods graphics off;
```

Output 75.1.1 displays the coding coefficients for the transformation of the independent variables to lie between −1 and 1 and some simple statistics for the response variable.

Output 75.1.1 Coding and Response Variable Information

The RSREG Procedure

Coding Coefficients for the Independent Variables

Factor	Subtracted off	Divided by
Time	12.000000	8.000000
Temp	250.000000	30.000000

Response Surface for Variable MBT: Percent Yield Mercaptobenzothiazole

Response Mean	79.916667
Root MSE	4.615964
R-Square	0.8003
Coefficient of Variation	5.7760

Output 75.1.2 shows that the lack of fit for the model is highly significant. Since the quadratic model does not fit the data very well, firm statements about the underlying process should not be based only on the current analysis. Note from the analysis of variance for the model that the test for the time factor is not significant. If further experimentation is undertaken, it might be best to fix Time at a moderate to high value and to concentrate on the effect of temperature. In the actual experiment discussed here, extra runs were made that confirmed the results of the following analysis.

Output 75.1.2 Analyses of Variance

Regression	DF	Type I Sum of Squares	R-Square	F Value	Pr > F
Linear	2	313.585803	0.4899	7.36	0.0243
Quadratic	2	146.768144	0.2293	3.44	0.1009
Crossproduct	1	51.840000	0.0810	2.43	0.1698
Total Model	5	512.193947	0.8003	4.81	0.0410

Residual	DF	Sum of Squares	Mean Square	F Value	Pr > F
Lack of Fit	3	124.696053	41.565351	39.63	0.0065
Pure Error	3	3.146667	1.048889		
Total Error	6	127.842720	21.307120		

Parameter	DF	Estimate	Standard Error	t Value	Pr > |t|	Parameter Estimate from Coded Data
Intercept	1	-545.867976	277.145373	-1.97	0.0964	82.173110
Time	1	6.872863	5.004928	1.37	0.2188	-1.014287
Temp	1	4.989743	2.165839	2.30	0.0608	-8.676768
Time*Time	1	0.021631	0.056784	0.38	0.7164	1.384394
Temp*Time	1	-0.030075	0.019281	-1.56	0.1698	-7.218045
Temp*Temp	1	-0.009836	0.004304	-2.29	0.0623	-8.852519

Output 75.1.2 *continued*

Factor	DF	Sum of Squares	Mean Square	F Value	Pr > F
Time	3	61.290957	20.430319	0.96	0.4704
Temp	3	461.250925	153.750308	7.22	0.0205

Factor	Label
Time	Reaction Time (Hours)
Temp	Temperature (Degrees Centigrade)

The canonical analysis (Output 75.1.3) indicates that the predicted response surface is shaped like a saddle. The eigenvalue of 2.5 shows that the valley orientation of the saddle is less curved than the hill orientation, with an eigenvalue of −9.99. The coefficients of the associated eigenvectors show that the valley is more aligned with Time and the hill with Temp. Because the canonical analysis resulted in a saddle point, the estimated surface does not have a unique optimum.

Output 75.1.3 Canonical Analysis

Factor	Critical Value Coded	Uncoded	Label
Time	−0.441758	8.465935	Reaction Time (Hours)
Temp	−0.309976	240.700718	Temperature (Degrees Centigrade)

Predicted value at stationary point: 83.741940

Eigenvalues	Eigenvectors Time	Temp
2.528816	0.953223	−0.302267
−9.996940	0.302267	0.953223

Stationary point is a saddle point.

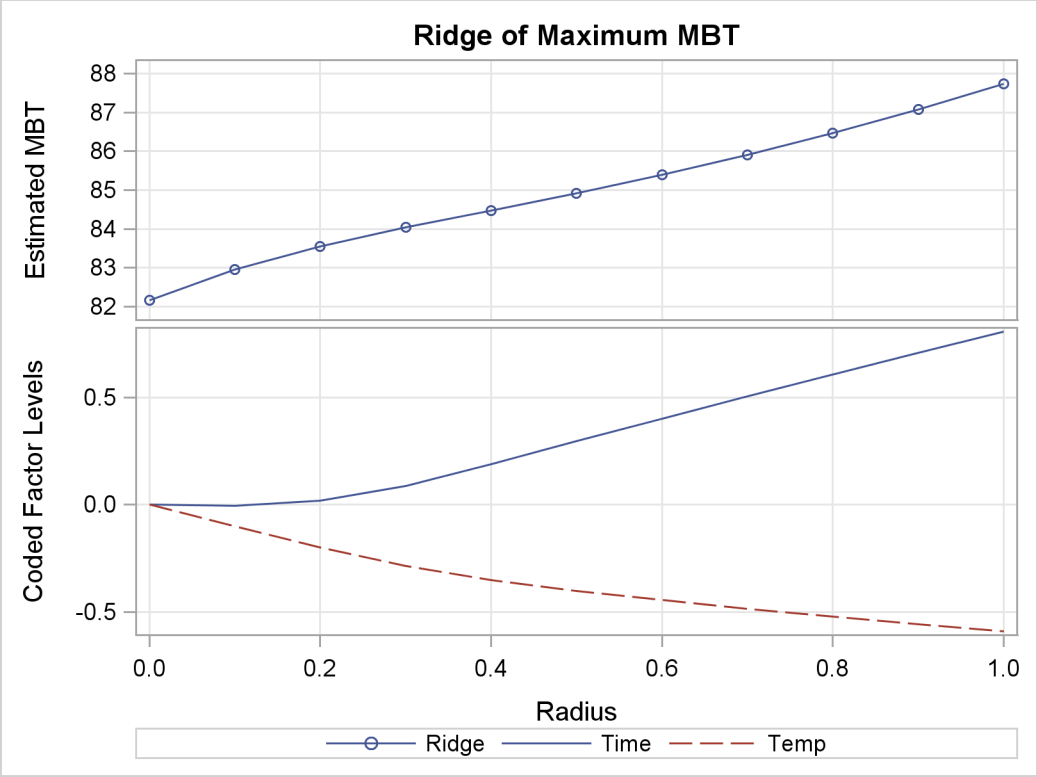
However, the ridge analysis in Output 75.1.4 and the ridge plot in Output 75.1.5 indicate that maximum yields result from relatively high reaction times and low temperatures. A contour plot of the predicted response surface, shown in Output 75.1.6, confirms this conclusion.

Output 75.1.4 Ridge Analysis

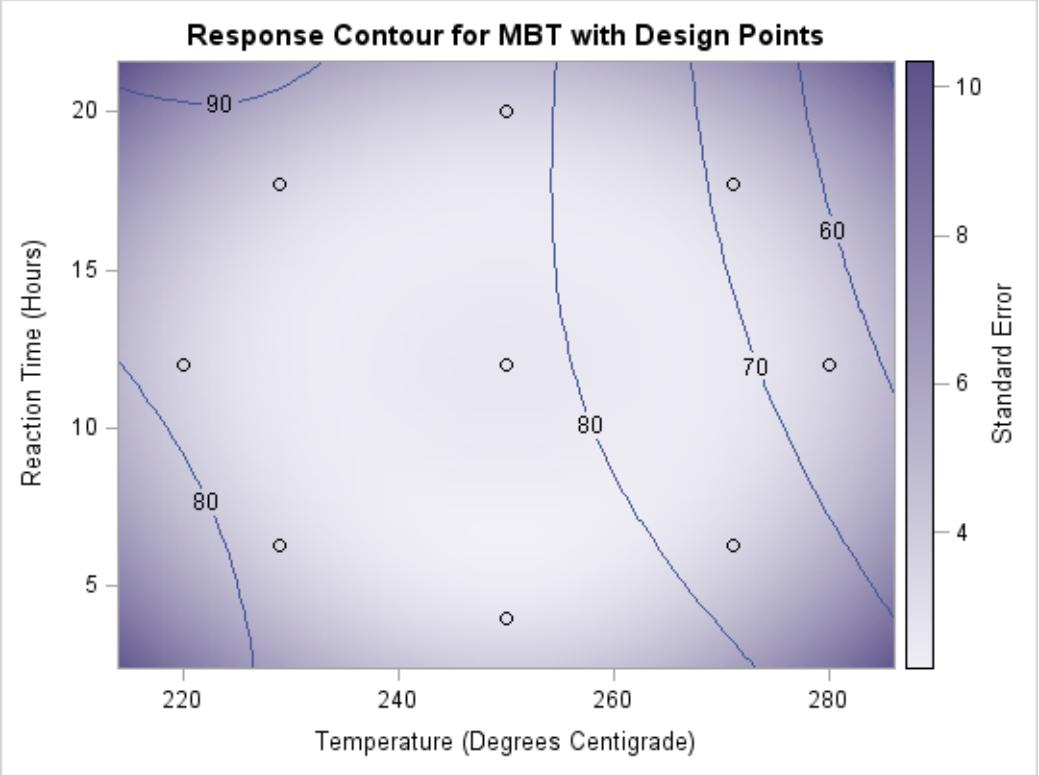
Estimated Ridge of Maximum Response for Variable MBT: Percent Yield Mercaptobenzothiazole

Coded Radius	Estimated Response	Standard Error	Uncoded Factor Values Time	Temp
0.0	82.173110	2.665023	12.000000	250.000000
0.1	82.952909	2.648671	11.964493	247.002956
0.2	83.558260	2.602270	12.142790	244.023941
0.3	84.037098	2.533296	12.704153	241.396084
0.4	84.470454	2.457836	13.517555	239.435227
0.5	84.914099	2.404616	14.370977	237.919138
0.6	85.390012	2.410981	15.212247	236.624811
0.7	85.906767	2.516619	16.037822	235.449230
0.8	86.468277	2.752355	16.850813	234.344204
0.9	87.076587	3.130961	17.654321	233.284652
1.0	87.732874	3.648568	18.450682	232.256238

Output 75.1.5 Ridge Plot of Predicted Response Surface



Output 75.1.6 Contour Plot of Predicted Response Surface



Example 75.2: Response Surface Analysis with Covariates

One way of viewing covariates is as extra sources of variation in the dependent variable that can mask the variation due to primary factors. This example demonstrates the use of the COVAR= option in PROC RSREG to fit a response surface model to the dependent variables corrected for the covariates.

You have a chemical process with a yield that you hypothesize to be dependent on three factors: reaction time, reaction temperature, and reaction pressure. You perform an experiment to measure this dependence. You are willing to include up to 20 runs in your experiment, but you can perform no more than 8 runs on the same day, so the design for the experiment is composed of three blocks. Additionally, you know that the grade of raw material for the reaction has a significant impact on the yield. You have no control over this, but you keep track of it. The following statements create a SAS data set containing the results of the experiment:

```
data Experiment;
   input Day Grade Time Temp Pressure Yield;
   datalines;
1 67      -1     -1     -1         32.98
1 68      -1      1      1         47.04
1 70       1     -1      1         67.11
1 66       1      1     -1         26.94
1 74       0      0      0        103.22
1 68       0      0      0         42.94
2 75      -1     -1      1        122.93
2 69      -1      1     -1         62.97
2 70       1     -1     -1         72.96
2 71       1      1      1         94.93
2 72       0      0      0         93.11
2 74       0      0      0        112.97
3 69       1.633  0      0         78.88
3 67      -1.633  0      0         52.53
3 68       0      1.633  0         68.96
3 71       0     -1.633  0         92.56
3 70       0      0      1.633     88.99
3 72       0      0     -1.633    102.50
3 70       0      0      0         82.84
3 72       0      0      0        103.12
;
```

Your first analysis neglects to take the covariates into account. The following statements use PROC RSREG to fit a response surface to the observed yield, but note that Day and Grade are omitted:

```
proc rsreg data=Experiment;
   model Yield = Time Temp Pressure;
run;
```

The ANOVA results shown in Output 75.2.1 indicate that *no* process variable effects are significantly larger than the background noise.

Output 75.2.1 Analysis of Variance Ignoring Covariates

The RSREG Procedure

Regression	DF	Type I Sum of Squares	R-Square	F Value	Pr > F
Linear	3	1880.842426	0.1353	0.67	0.5915
Quadratic	3	2370.438681	0.1706	0.84	0.5023
Crossproduct	3	241.873250	0.0174	0.09	0.9663
Total Model	9	4493.154356	0.3233	0.53	0.8226

Residual	DF	Sum of Squares	Mean Square
Total Error	10	9405.129724	940.512972

However, when the yields are adjusted for covariate effects of day and grade of raw material, very strong process variable effects are revealed. The following statements produce the ANOVA results in Output 75.2.2. Note that in order to include the effects of the classification factor Day as covariates, you need to create dummy variables indicating each day separately.

```
data Experiment;
   set Experiment;
   d1 = (Day = 1);
   d2 = (Day = 2);
   d3 = (Day = 3);

ods graphics on;
proc rsreg data=Experiment plots=all;
   model Yield = d1-d3 Grade Time Temp Pressure / covar=4;
run;
ods graphics off;
```

The results show very strong effects due to both the covariates and the process variables.

Output 75.2.2 Analysis of Variance Including Covariates

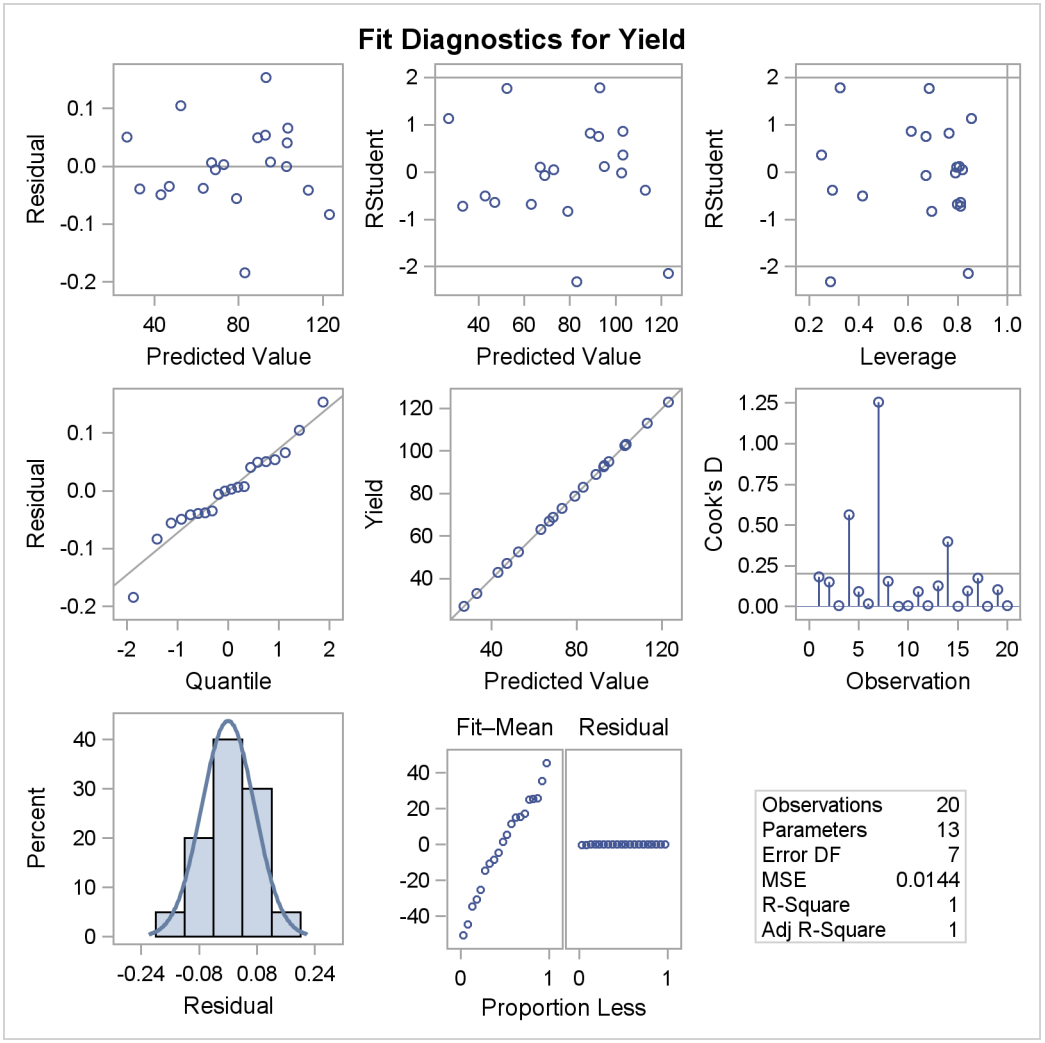
The RSREG Procedure

Regression	DF	Type I Sum of Squares	R-Square	F Value	Pr > F
Covariates	3	13695	0.9854	316957	<.0001
Linear	3	156.524497	0.0113	3622.53	<.0001
Quadratic	3	22.989775	0.0017	532.06	<.0001
Crossproduct	3	23.403614	0.0017	541.64	<.0001
Total Model	12	13898	1.0000	80413.2	<.0001

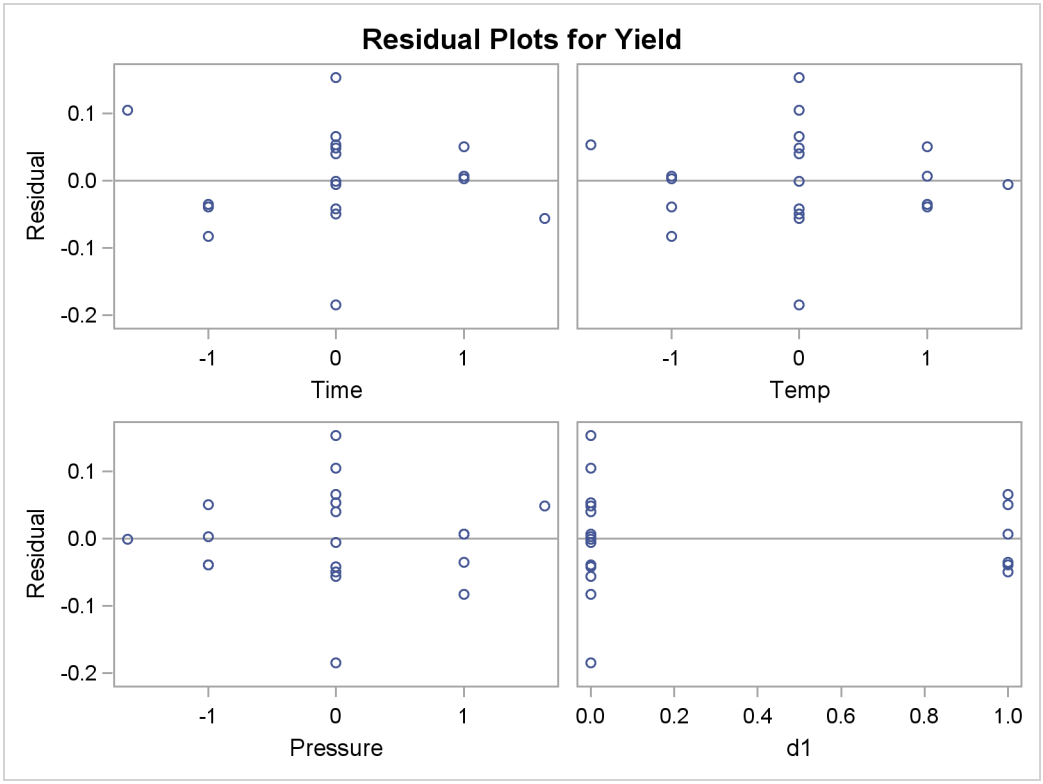
Residual	DF	Sum of Squares	Mean Square
Total Error	7	0.100820	0.014403

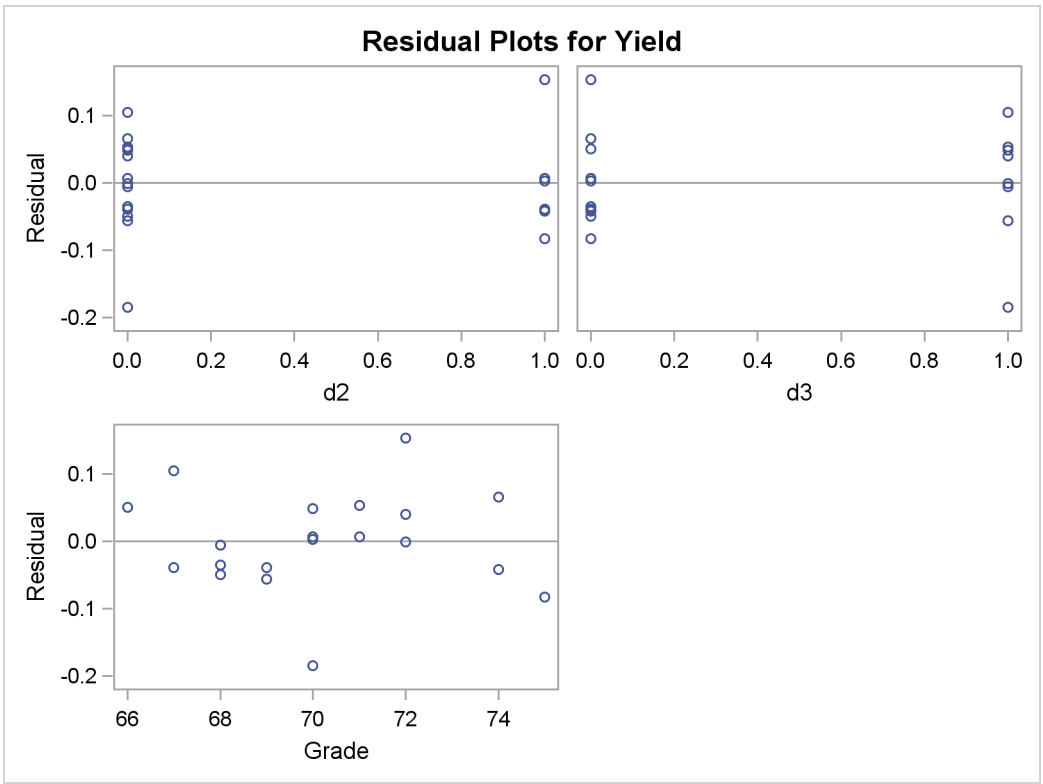
The number of observations in the data set might be too small for the diagnostic plots in Output 75.2.3 to dependably identify problems; however, some outliers are indicated. The residual plots in Output 75.2.4 do not display any obvious structure.

Output 75.2.3 Fit Diagnostics



Output 75.2.4 Residual Plots





References

Box, G. E. P. (1954), "The Exploration and Exploitation of Response Surfaces: Some General Considerations," *Biometrics*, 10, 16.

Box, G. E. P. and Draper, N. R. (1982), "Measures of Lack of Fit for Response Surface Designs and Predictor Variable Transformations," *Technometrics*, 24, 1–8.

Box, G. E. P. and Draper, N. R. (1987), *Empirical Model Building and Response Surfaces*, New York: John Wiley & Sons.

Box, G. E. P. and Hunter, J. S. (1957), "Multifactor Experimental Designs for Exploring Response Surfaces," *Annals of Mathematical Statistics*, 28, 195–242.

Box, G. E. P., Hunter, W. G., and Hunter, J. S. (1978), *Statistics for Experimenters*, New York: John Wiley & Sons.

Box, G. E. P. and Wilson, K. J. (1951), "On the Experimental Attainment of Optimum Conditions," *Journal of the Royal Statistical Society*.

Cleveland, W. S. (1993), *Visualizing Data*, Summit, NJ: Hobart Press.

Cochran, W. G. and Cox, G. M. (1957), *Experimental Designs*, Second Edition, New York: John Wiley & Sons.

Draper, N. R. (1963), "Ridge Analysis of Response Surfaces," *Technometrics*, 5, 469–479.

Draper, N. R. and John, J. A. (1988), "Response Surface Designs for Quantitative and Qualitative Variables," *Technometrics*, 30, 423–428.

Draper, N. R. and Smith, H. (1981), *Applied Regression Analysis*, Second Edition, New York: John Wiley & Sons.

Frankel, S. A. (1961), "Statistical Design of Experiments for Process Development of MBT," *Rubber Age*, 89, 453.

John, P. W. M. (1971), *Statistical Design and Analysis of Experiments*, New York: Macmillan.

Mead, R. and Pike, D. J. (1975), "A Review of Response Surface Methodology from a Biometric Point of View," *Biometrics*, 31, 803.

Meyer, D. C. (1963), "Response Surface Methodology in Education and Psychology," *Journal of Experimental Education*, 31, 329.

Myers, R. H. (1976), *Response Surface Methodology*, Blacksburg: Virginia Polytechnic Institute and State University.

Myers, R. H. and Montgomery, D. C. (1995), *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*, New York: John Wiley & Sons.

Rawlings, J. O., Pantula, S. G., and Dickey, D. A. (1998), *Applied Regression Analysis: A Research Tool*, Springer Texts in Statistics, Second Edition, New York: Springer-Verlag.

Schneider, A. M. and Stockett, A. L. (1963), "An Experiment to Select Optimum Operating Conditions on the Basis of Arbitrary Preference Ratings," *Chemical Engineering Progress Symposium Series*.

Chapter 76

The SCORE Procedure

Contents

Overview: SCORE Procedure **5759**
Raw Data Set 5760
Scoring Coefficients Data Set 5760
Standardization of Raw Data 5761
Getting Started: SCORE Procedure **5762**
Syntax: SCORE Procedure **5766**
PROC SCORE Statement 5766
BY Statement 5767
ID Statement 5768
VAR Statement 5768
Details: SCORE Procedure **5769**
Missing Values 5769
Regression Parameter Estimates from PROC REG 5769
Output Data Set 5769
Computational Resources 5770
Examples: SCORE Procedure **5770**
Example 76.1: Factor Scoring Coefficients 5770
Example 76.2: Regression Parameter Estimates 5775
Example 76.3: Custom Scoring Coefficients 5780
References **5781**

Overview: SCORE Procedure

The SCORE procedure multiplies values from two SAS data sets, one containing coefficients (for example, factor-scoring coefficients or regression coefficients) and the other containing raw data to be scored using the coefficients from the first data set. The result of this multiplication is a SAS data set containing linear combinations of the coefficients and the raw data values.

Many statistical procedures output coefficients that PROC SCORE can apply to raw data to produce scores. The new score variable is formed as a linear combination of raw data and scoring coefficients. For each observation in the raw data set, PROC SCORE multiplies the value of a variable in the raw data set by the matching scoring coefficient from the data set of scoring coefficients. This

multiplication process is repeated for each variable in the VAR statement. The resulting products are then summed to produce the value of the new score variable. This entire process is repeated for each observation in the raw data set. In other words, PROC SCORE cross multiplies part of one data set with another.

Raw Data Set

The raw data set can contain the original data used to calculate the scoring coefficients, or it can contain an entirely different data set. The raw data set must contain all the variables needed to produce scores. In addition, the scoring coefficients and the variables in the raw data set that are used in scoring must have the same names. See the section "Getting Started: SCORE Procedure" on page 5762 for more information.

Scoring Coefficients Data Set

The data set containing scoring coefficients must contain two special variables: the _TYPE_ variable and the _NAME_ or _MODEL_ variable.

- The _TYPE_ variable identifies the observations that contain scoring coefficients.
- The _NAME_ or _MODEL_ variable provides a SAS name for the new score variable.

PROC SCORE first looks for a _NAME_ variable in the SCORE= input data set. If there is such a variable, the variable's value is what SCORE uses to name the new score variable. If the SCORE= data set does not have a _NAME_ variable, then PROC SCORE looks for a _MODEL_ variable.

For example, PROC FACTOR produces an output data set that contains factor-scoring coefficients. In this output data set, the scoring coefficients are identified by _TYPE_='SCORE'. For _TYPE_='SCORE', the _NAME_ variable has values of 'Factor1', 'Factor2', and so forth. PROC SCORE gives the new score variables the names Factor1, Factor2, and so forth.

As another example, the REG procedure produces an output data set that contains parameter estimates. In this output data set, the parameter estimates are identified by _TYPE_='PARMS'. The _MODEL_ variable contains the label used in the MODEL statement in PROC REG, or it uses MODEL*n* if no label is specified. This label is the name PROC SCORE gives to the new score variable.

Standardization of Raw Data

PROC SCORE automatically standardizes or centers the DATA= variables for you, based on information from the original variables and analysis from the SCORE= data set.

If the SCORE= scoring coefficients data set contains observations with _TYPE_='MEAN' and _TYPE_='STD', then PROC SCORE standardizes the raw data before scoring. For example, this type of SCORE= data set can come from PROC PRINCOMP without the COV option.

If the SCORE= scoring coefficients data set contains observations with _TYPE_='MEAN' but _TYPE_='STD' is absent, then PROC SCORE centers the raw data (the means are subtracted) before scoring. For example, this type of SCORE= data set can come from PROC PRINCOMP with the COV option.

If the SCORE= scoring coefficients data set does not contain observations with _TYPE_='MEAN' and _TYPE_='STD', or if you use the NOSTD option, then PROC SCORE does not center or standardize the raw data.

If the SCORE= scoring coefficients are obtained from observations with _TYPE_='USCORE', then PROC SCORE "standardizes" the raw data by using the uncorrected standard deviations identified by _TYPE_='USTD', and the means are not subtracted from the raw data. For example, this type of SCORE= data set can come from PROC PRINCOMP with the NOINT option. For more information about _TYPE_='USCORE' scoring coefficients in TYPE=UCORR or TYPE=UCOV output data sets, see Appendix A, "Special SAS Data Sets."

You can use PROC SCORE to score the data that were also used to generate the scoring coefficients, although more typically, scoring results are directly obtained from the OUT= data set in a procedure that computes scoring coefficients. When scoring new data, it is important to realize that PROC SCORE assumes that the new data have approximately the same scales as the original data. For example, if you specify the COV option with PROC PRINCOMP for the original analysis, the scoring coefficients in the PROC PRINCOMP OUTSTAT= data set are not appropriate for standardized data. With the COV option, PROC PRINCOMP will not output _TYPE_='STD' observations to the OUTSTAT= data set, and PROC SCORE will only subtract the means of the original (not new) variables from the new variables before multiplying. Without the COV option in PROC PRINCOMP, both the original variable means and standard deviations will be in the OUTSTAT= data set, and PROC SCORE will subtract the original variable means from the new variables and divide them by the original variable standard deviations before multiplying.

In general, procedures that output scoring coefficients in their OUTSTAT= data sets provide the necessary information for PROC SCORE to determine the appropriate standardization. However, if you use PROC SCORE with a scoring coefficients data set that you constructed without _TYPE_='MEAN' and _TYPE_='STD' observations, you might have to do the relevant centering or standardization of the new data first. If you do this, you must use the means and standard deviations of the original variables—that is, the variables that were used to generate the coefficients—not the means and standard deviations of the variables to be scored.

See the section "Getting Started: SCORE Procedure" on page 5762 for further illustration.

Getting Started: SCORE Procedure

The SCORE procedure multiplies the values from two SAS data sets and creates a new data set to contain the results of the multiplication. The variables in the new data set are linear combinations of the variables in the two input data sets. Typically, one of these data sets contains raw data that you want to score, and the other data set contains scoring coefficients.

The following example demonstrates how to use the SCORE procedure to multiply values from two SAS data sets, one containing factor-scoring coefficients and the other containing raw data to be scored using the scoring coefficients.

Suppose you are interested in the performance of three different types of schools: private schools, state-run urban schools, and state-run rural schools. You want to compare the schools' performances as measured by student grades on standard tests in English, mathematics, and biology. You administer these tests and record the scores for each of the three types of schools.

The following DATA step creates the SAS data set Schools. The data are provided by Chaseling (1996).

```
data Schools;
   input Type $ English Math Biology @@;
   datalines;
p  52  55  45  p  42  49  40  p  63  64  54
p  47  50  51  p  64  69  47  p  63  67  54
p  59  63  42  p  56  61  41  p  41  44  72

... more lines ...

r  50  47  49  r  55  48  46  r  38  36  51
;
```

The data set Schools contains the character variable Type, which represents the type of school. Valid values are p (private schools), r (state-run rural schools), and u (state-run urban schools).

The three numeric variables in the data set are English, Math, and Biology, which represent the student scores for English, mathematics, and biology, respectively. The double trailing at sign (@@) in the INPUT statement specifies that observations are input from each line until all values are read.

The following statements invoke the FACTOR procedure to compute the data set of factor scoring coefficients. The statements perform a principal components factor analysis that uses all three numeric variables in the SAS data set Schools. The OUTSTAT= option requests that PROC FACTOR output the factor scores to the data set Scores. The NOPRINT option suppresses display of the output.

```
proc factor data=Schools score outstat=Scores noprint;
   var english math biology;
run;

proc score data=schools score=Scores out=New;
   var english math biology;
   id type;
run;
```

The SCORE procedure is then invoked using Schools as the raw data set to be scored and Scores as the scoring data set. The OUT= option creates the SAS data set New to contain the linear combinations.

The VAR statement specifies that the variables English, Math, and Biology are used in computing scores. The ID statement copies the variable Type from the Schools data set to the output data set New.

The following statements print the SAS output data set Scores, the first two observations from the original data set Schools, and the first two observations of the resulting data set New.

```
title 'OUTSTAT= Data Set from PROC FACTOR';
proc print data=Scores;
run;

title 'First Two Observations of the DATA= Data Set from PROC SCORE';
proc print data=Schools(obs=2);
run;

title 'First Two Observations of the OUT= Data Set from PROC SCORE';
proc print data=New(obs=2);
run;
```

Figure 76.1 displays the output data set Scores produced by the FACTOR procedure. The last observation (number 11) contains the scoring coefficients (_TYPE_='SCORE'). Only one factor has been retained.

Figure 76.1 Listing of the Data Set Created by PROC FACTOR

OUTSTAT= Data Set from PROC FACTOR

Obs	_TYPE_	_NAME_	English	Math	Biology
1	MEAN		55.525	52.325	50.350
2	STD		12.949	12.356	12.239
3	N		120.000	120.000	120.000
4	CORR	English	1.000	0.833	0.672
5	CORR	Math	0.833	1.000	0.594
6	CORR	Biology	0.672	0.594	1.000
7	COMMUNAL		0.881	0.827	0.696
8	PRIORS		1.000	1.000	1.000
9	EIGENVAL		2.405	0.437	0.159
10	PATTERN	Factor1	0.939	0.910	0.834
11	SCORE	Factor1	0.390	0.378	0.347

Figure 76.2 lists the first two observations of the original SAS data set (Schools).

Figure 76.2 First Two Observations of the Schools Data Set

First Two Observations of the DATA= Data Set from PROC SCORE

Obs	Type	English	Math	Biology
1	p	52	55	45
2	p	42	49	40

Figure 76.3 lists the first two observations of the output data set New created by PROC SCORE.

Figure 76.3 Listing of the New Data Set

First Two Observations of the OUT= Data Set from PROC SCORE

Obs	Type	Factor1
1	p	-0.17604
2	p	-0.80294

The score variable Factor1 in the New data set is named according to the value of the _NAME_ variable in the Scores data set. The values of the variable Factor1 are computed as follows: the DATA= data set variables are standardized using the same means and standard deviations that PROC FACTOR used when extracting the factors because the Scores data set contains observations with _TYPE_='MEAN' and _TYPE_='STD'.

Note that in order to correctly use standardized scoring coefficients created by other procedures such as PROC FACTOR in this example, the data to be scored must be standardized in the same way that the data were standardized when the scoring coefficients were computed. Otherwise, the resulting scores might be incorrect. PROC SCORE does this automatically if the SCORE= data set is the original OUTSTAT= data set output from the procedure creating the scoring coefficients.

These standardized variables are then multiplied by their respective standardized scoring coefficients from the data set Scores. These products are summed over all three variables, and the sum is the value of the new variable Factor1. The first two values of the scored variable Factor1 are obtained as follows:

$$\left(\frac{(52-55.525)}{12.949}\times 0.390\right)+\left(\frac{(55-52.325)}{12.356}\times 0.378\right)+\left(\frac{(45-50.350)}{12.239}\times 0.347\right)=-0.17604$$

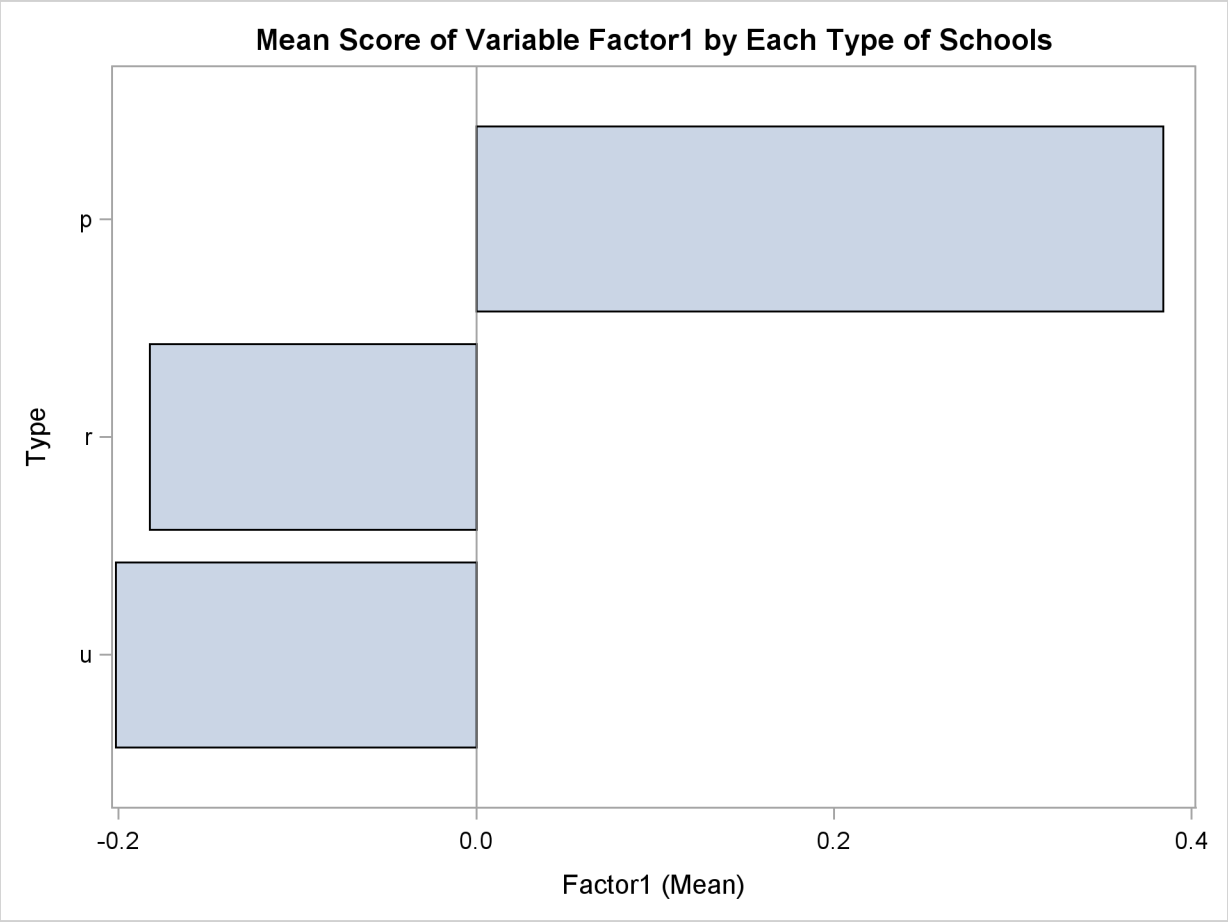
$$\left(\frac{(42-55.525)}{12.949}\times 0.390\right)+\left(\frac{(49-52.325)}{12.356}\times 0.378\right)+\left(\frac{(40-50.350)}{12.239}\times 0.347\right)=-0.80294$$

The following statements request that the SGPLOT procedure produce a horizontal bar chart of the variable Type. The length of each bar represents the mean of the variable Factor1.

```
title 'Mean Score of Variable Factor1 by Each Type of Schools';
proc sgplot data=New;
   hbar type / stat = mean response=Factor1;
run;
```

Figure 76.4 displays the mean score of the variable Factor1 for each of the three school types. For private schools (Type=p), the average value of the variable Factor1 is 0.384, while for state-run schools the average values are much lower. The state-run urban schools (Type=u) have the lowest mean value of –0.202, and the state-run rural schools (Type=r) have a mean value of –0.183.

Figure 76.4 Bar Chart of School Type



Syntax: SCORE Procedure

The following statements are available in the SCORE procedure:

PROC SCORE DATA=SAS-data-set< *options* > **;**
BY *variables* **;**
ID *variables* **;**
VAR *variables* **;**

The only required statement is the PROC SCORE statement. The BY, ID, and VAR statements are described following the PROC SCORE statement.

PROC SCORE Statement

PROC SCORE DATA= *SAS-data-set* < *options* > **;**

You can specify the following options in the PROC SCORE statement.

DATA=*SAS-data-set*
names the input SAS data set containing the raw data to score. This option is required.

NOSTD
suppresses centering and scaling of the raw data. Ordinarily, if PROC SCORE finds _TYPE_='MEAN', _TYPE_= 'USCORE', _TYPE_='USTD', or _TYPE_='STD' observations in the SCORE= data set, the procedure uses these to standardize the raw data before scoring.

OUT=*SAS-data-set*
specifies the name of the SAS data set created by PROC SCORE. If you want to create a permanent SAS data set, you must specify a two-level name. See *SAS Language Reference: Concepts* for more information about permanent SAS data sets. If the OUT= option is omitted, PROC SCORE still creates an output data set and automatically names it according to the DATA*n* convention, as if you omitted a data set name in a DATA statement.

PREDICT
specifies that PROC SCORE should treat coefficients of −1 in the SCORE= data set as 0. In regression applications, the dependent variable is coded with a coefficient of −1. Applied directly to regression results, PROC SCORE produces negative residuals (see the description of the RESIDUAL option, which follows); the PREDICT option produces predicted values instead.

RESIDUAL
reverses the sign of each score. Applied directly to regression results, PROC SCORE produces negative residuals (PREDICT−ACTUAL); the RESIDUAL option produces positive residuals (ACTUAL−PREDICT) instead.

SCORE=*SAS-data-set*

names the data set containing the scoring coefficients. If you omit the SCORE= option, the most recently created SAS data set is used. This data set must have two special variables: _TYPE_ and either _NAME_ or _MODEL_.

TYPE=*name* **or** *'string'*

specifies the observations in the SCORE= data set that contain scoring coefficients. The TYPE= procedure option is unrelated to the data set option that has the same name. PROC SCORE examines the values of the special variable _TYPE_ in the SCORE= data set. When the value of _TYPE_ matches TYPE=*name*, the observation in the SCORE= data set is used to score the raw data in the DATA= data set.

If you omit the TYPE= option, scoring coefficients are read from observations with either _TYPE_='SCORE' or _TYPE_='USCORE'. Because the default for PROC SCORE is TYPE=SCORE, you need not specify the TYPE= option for factor scoring or for computing scores from OUTSTAT= data sets from the CANCORR, CANDISC, PRINCOMP, or VARCLUS procedure. When you use regression coefficients from PROC REG, specify TYPE=PARMS.

The maximum length of the argument specified in the TYPE= option depends on the length defined by the VALIDVARNAME= SAS system option. For additional information, see *SAS Language Reference: Dictionary*.

Note that the TYPE= option setting is not case sensitive. For example, the two option settings TYPE='MyScore' and TYPE='myscore' are equivalent.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC SCORE to obtain separate scoring for observations in groups defined by the BY variables. You can also specify a BY statement to apply separate groups of scoring coefficients to the entire DATA= data set.

If your SCORE= input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC SCORE. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide.*

If the DATA= data set does not contain any of the BY variables, the entire DATA= data set is scored by each BY group of scoring coefficients in the SCORE= data set.

If the DATA= data set contains some but not all of the BY variables, or if some BY variables do not have the same type or length in the DATA= data set as in the SCORE= data set, then PROC SCORE prints an error message and stops.

If all the BY variables appear in the DATA= data set with the same type and length as in the SCORE= data set, then each BY group in the DATA= data set is scored using scoring coefficients from the corresponding BY group in the SCORE= data set. The BY groups in the DATA= data set must be in the same order as in the SCORE= data set. All BY groups in the DATA= data set must also appear in the SCORE= data set. If you do not specify the NOTSORTED option, some BY groups can appear in the SCORE= data set but not in the DATA= data set; such BY groups are not used in computing scores.

ID Statement

ID *variables* **;**

The ID statement identifies variables from the DATA= data set to be included in the OUT= data set. If there is no ID statement, all variables from the DATA= data set are included in the OUT= data set. The ID variables can be character or numeric.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the variables to be used in computing scores. These variables must be in both the DATA= and SCORE= input data sets and must be numeric. If you do not specify a VAR statement, the procedure uses all numeric variables in the SCORE= data set. You should almost always specify a VAR statement with PROC SCORE because you would rarely use all the numeric variables in your data set to compute scores.

Details: SCORE Procedure

Missing Values

If one of the scoring variables in the DATA= data set has a missing value for an observation, all the scores have missing values for that observation. The exception to this criterion is that if the PREDICT option is specified, the variable with a coefficient of −1 can tolerate a missing value and still produce a prediction score. Also, a variable with a coefficient of 0 can tolerate a missing value.

If a scoring coefficient in the SCORE= data set has a missing value for an observation, the coefficient is not used in creating the new score variable for the observation. In other words, missing values of scoring coefficients are treated as zeros. This treatment affects only the observation in which the missing value occurs.

Regression Parameter Estimates from PROC REG

If the SCORE= data set is an OUTEST= data set produced by PROC REG and if you specify TYPE=PARMS, the interpretation of the new score variables depends on the PROC SCORE options chosen and the variables listed in the VAR statement. If the VAR statement contains only the independent variables used in a model in PROC REG, the new score variables give the predicted values. If the VAR statement contains the dependent variables and the independent variables used in a model in PROC REG, the interpretation of the new score variables depends on the PROC SCORE options chosen. If you omit both the PREDICT and the RESIDUAL options, the new score variables give negative residuals (PREDICT−ACTUAL). If you specify the RESIDUAL option, the new score variables give positive residuals (ACTUAL−PREDICT). If you specify the PREDICT option, the new score variables give predicted values.

Unless you specify the NOINT option for PROC REG, the OUTEST= data set contains the variable Intercept. The SCORE procedure uses the intercept value in computing the scores.

Output Data Set

PROC SCORE produces an output data set but displays no output. The output OUT= data set contains the following variables:

- the ID variables, if any
- all variables from the DATA= data set, if no ID variables are specified
- the BY variables, if any

- the new score variables, named from the _NAME_ or _MODEL_ values in the SCORE= data set

Computational Resources

Let

v = number of variables used in computing scores
s = number of new score variables
b = maximum number of new score variables in a BY group
n = original input value

Memory

The array storage required is approximately $8(4v + (3 + v)b + s)$ bytes. When you do not use BY processing, the array storage required is approximately $8(4v + (4 + v)s)$ bytes.

Time

The time required to construct the scoring matrix is roughly proportional to vs, and the time needed to compute the scores is roughly proportional to nvs.

Examples: SCORE Procedure

The following three examples use a subset of the Fitness data set. The complete data set is given in Chapter 73, "The REG Procedure."

Example 76.1: Factor Scoring Coefficients

This example shows how to use PROC SCORE with factor scoring coefficients. First, the FACTOR procedure produces an output data set containing scoring coefficients in observations identified by _TYPE_='SCORE'. These data, together with the original data set Fitness, are supplied to PROC SCORE, resulting in a data set containing scores Factor1 and Factor2. The following statements produce Output 76.1.1 through Output 76.1.3:

```
   /* This data set contains only the first 12 observations   */
   /* from the full data set used in the chapter on PROC REG. */
data Fitness;
   input Age Weight Oxygen RunTime RestPulse RunPulse @@;
   datalines;
44 89.47  44.609 11.37 62 178     40 75.07  45.313 10.07 62 185
44 85.84  54.297  8.65 45 156     42 68.15  59.571  8.17 40 166
38 89.02  49.874  9.22 55 178     47 77.45  44.811 11.63 58 176
40 75.98  45.681 11.95 70 176     43 81.19  49.091 10.85 64 162
44 81.42  39.442 13.08 63 174     38 81.87  60.055  8.63 48 170
44 73.03  50.541 10.13 45 168     45 87.66  37.388 14.03 56 186
;
proc factor data=Fitness outstat=FactOut
            method=prin rotate=varimax score;
   var Age Weight RunTime RunPulse RestPulse;
title 'FACTOR SCORING EXAMPLE';
run;

proc print data=FactOut;
title2 'Data Set from PROC FACTOR';
run;

proc score data=Fitness score=FactOut out=FScore;
   var Age Weight RunTime RunPulse RestPulse;
run;

proc print data=FScore;
title2 'Data Set from PROC SCORE';
run;
```

Output 76.1.1 shows the PROC FACTOR output. The scoring coefficients for the two factors are shown at the end of the PROC FACTOR output.

Output 76.1.1 Creating an OUTSTAT= Data Set with PROC FACTOR

```
                        FACTOR SCORING EXAMPLE

                          The FACTOR Procedure
                  Initial Factor Method: Principal Components

                      Prior Communality Estimates: ONE


          Eigenvalues of the Correlation Matrix: Total = 5  Average = 1

                Eigenvalue    Difference    Proportion    Cumulative

             1  2.30930638    1.11710686        0.4619        0.4619
             2  1.19219952    0.30997249        0.2384        0.7003
             3  0.88222702    0.37965990        0.1764        0.8767
             4  0.50256713    0.38886717        0.1005        0.9773
             5  0.11369996                      0.0227        1.0000
```

Output 76.1.1 *continued*

2 factors will be retained by the MINEIGEN criterion.

Factor Pattern

	Factor1	Factor2
Age	0.29795	0.93675
Weight	0.43282	−0.17750
RunTime	0.91983	0.28782
RunPulse	0.72671	−0.38191
RestPulse	0.81179	−0.23344

Variance Explained by Each Factor

Factor1	Factor2
2.3093064	1.1921995

Final Communality Estimates: Total = 3.501506

Age	Weight	RunTime	RunPulse	RestPulse
0.96628351	0.21883401	0.92893333	0.67396207	0.71349297

FACTOR SCORING EXAMPLE

The FACTOR Procedure
Rotation Method: Varimax

Orthogonal Transformation Matrix

	1	2
1	0.92536	0.37908
2	−0.37908	0.92536

Rotated Factor Pattern

	Factor1	Factor2
Age	−0.07939	0.97979
Weight	0.46780	−0.00018
RunTime	0.74207	0.61503
RunPulse	0.81725	−0.07792
RestPulse	0.83969	0.09172

Variance Explained by Each Factor

Factor1	Factor2
2.1487753	1.3527306

Output 76.1.1 *continued*

Final Communality Estimates: Total = 3.501506

Age	Weight	RunTime	RunPulse	RestPulse
0.96628351	0.21883401	0.92893333	0.67396207	0.71349297

FACTOR SCORING EXAMPLE

The FACTOR Procedure
Rotation Method: Varimax

Scoring Coefficients Estimated by Regression

Squared Multiple Correlations of the Variables with Each Factor

Factor1	Factor2
1.0000000	1.0000000

Standardized Scoring Coefficients

	Factor1	Factor2
Age	-0.17846	0.77600
Weight	0.22987	-0.06672
RunTime	0.27707	0.37440
RunPulse	0.41263	-0.17714
RestPulse	0.39952	-0.04793

Output 76.1.2 lists the OUTSTAT= data set from PROC FACTOR. Note that observations 18 and 19 have _TYPE_='SCORE'. Observations 1 and 2 have _TYPE_='MEAN' and _TYPE_='STD', respectively. These four observations are used by PROC SCORE.

Output 76.1.2 OUTSTAT= Data Set from PROC FACTOR Reproduced with PROC PRINT

FACTOR SCORING EXAMPLE
Data Set from PROC FACTOR

Obs	_TYPE_	_NAME_	Age	Weight	RunTime	RunPulse	Rest Pulse
1	MEAN		42.4167	80.5125	10.6483	172.917	55.6667
2	STD		2.8431	6.7660	1.8444	8.918	9.2769
3	N		12.0000	12.0000	12.0000	12.000	12.0000
4	CORR	Age	1.0000	0.0128	0.5005	-0.095	-0.0080
5	CORR	Weight	0.0128	1.0000	0.2637	0.173	0.2396
6	CORR	RunTime	0.5005	0.2637	1.0000	0.556	0.6620
7	CORR	RunPulse	-0.0953	0.1731	0.5555	1.000	0.4853
8	CORR	RestPulse	-0.0080	0.2396	0.6620	0.485	1.0000
9	COMMUNAL		0.9663	0.2188	0.9289	0.674	0.7135
10	PRIORS		1.0000	1.0000	1.0000	1.000	1.0000
11	EIGENVAL		2.3093	1.1922	0.8822	0.503	0.1137
12	UNROTATE	Factor1	0.2980	0.4328	0.9198	0.727	0.8118
13	UNROTATE	Factor2	0.9368	-0.1775	0.2878	-0.382	-0.2334
14	TRANSFOR	Factor1	0.9254	-0.3791	.	.	.
15	TRANSFOR	Factor2	0.3791	0.9254	.	.	.
16	PATTERN	Factor1	-0.0794	0.4678	0.7421	0.817	0.8397
17	PATTERN	Factor2	0.9798	-0.0002	0.6150	-0.078	0.0917
18	SCORE	Factor1	-0.1785	0.2299	0.2771	0.413	0.3995
19	SCORE	Factor2	0.7760	-0.0667	0.3744	-0.177	-0.0479

Since the PROC SCORE statement does not contain the NOSTD option, the data in the Fitness data set are standardized before scoring. For each variable specified in the VAR statement, the mean and standard deviation are obtained from the FactOut data set. For each observation in the Fitness data set, the variables are then standardized. For example, for observation 1 in the Fitness data set, the variable Age is standardized to $0.5569 = [(44 - 42.4167)/2.8431]$.

After the data in the Fitness data set are standardized, the standardized values of the variables in the VAR statement are multiplied by the matching coefficients in the FactOut data set, and the resulting products are summed. This sum is output as a value of the new score variable.

Output 76.1.3 displays the FScore data set produced by PROC SCORE. This data set contains the variables Age, Weight, Oxygen, RunTime, RestPulse, and RunPulse from the Fitness data set. It also contains Factor1 and Factor2, the two new score variables.

Output 76.1.3 OUT= Data Set from PROC SCORE Reproduced with PROC PRINT

FACTOR SCORING EXAMPLE
Data Set from PROC SCORE

Obs	Age	Weight	Oxygen	Run Time	Rest Pulse	Run Pulse	Factor1	Factor2
1	44	89.47	44.609	11.37	62	178	0.82129	0.35663
2	40	75.07	45.313	10.07	62	185	0.71173	-0.99605
3	44	85.84	54.297	8.65	45	156	-1.46064	0.36508
4	42	68.15	59.571	8.17	40	166	-1.76087	-0.27657
5	38	89.02	49.874	9.22	55	178	0.55819	-1.67684
6	47	77.45	44.811	11.63	58	176	-0.00113	1.40715
7	40	75.98	45.681	11.95	70	176	0.95318	-0.48598
8	43	81.19	49.091	10.85	64	162	-0.12951	0.36724
9	44	81.42	39.442	13.08	63	174	0.66267	0.85740
10	38	81.87	60.055	8.63	48	170	-0.44496	-1.53103
11	44	73.03	50.541	10.13	45	168	-1.11832	0.55349
12	45	87.66	37.388	14.03	56	186	1.20836	1.05948

Example 76.2: Regression Parameter Estimates

In this example, PROC REG computes regression parameter estimates for the Fitness data. (See Example 76.1 to for more information about how to create the Fitness data set.) The parameter estimates are output to a data set and used as scoring coefficients. For the first part of this example, PROC SCORE is used to score the Fitness data, which are the same data used in the regression.

In the second part of this example, PROC SCORE is used to score a new data set, Fitness2. For PROC SCORE, the TYPE= specification is PARMS, and the names of the score variables are found in the variable _MODEL_, which gets its values from the model label. The following code produces Output 76.2.1 through Output 76.2.3:

```
proc reg data=Fitness outest=RegOut;

OxyHat: model Oxygen=Age Weight RunTime RunPulse RestPulse;
   title 'REGRESSION SCORING EXAMPLE';
run;

proc print data=RegOut;
   title2 'OUTEST= Data Set from PROC REG';
run;

proc score data=Fitness score=RegOut out=RScoreP type=parms;
   var Age Weight RunTime RunPulse RestPulse;
run;

proc print data=RScoreP;
   title2 'Predicted Scores for Regression';
run;
```

```
proc score data=Fitness score=RegOut out=RScoreR type=parms;
   var Oxygen Age Weight RunTime RunPulse RestPulse;
run;

proc print data=RScoreR;
   title2 'Negative Residual Scores for Regression';
run;
```

Output 76.2.1 shows the PROC REG output. The column labeled "Parameter Estimates" lists the parameter estimates. These estimates are output to the RegOut data set.

Output 76.2.1 Creating an OUTEST= Data Set with PROC REG

REGRESSION SCORING EXAMPLE

The REG Procedure
Model: OxyHat
Dependent Variable: Oxygen

Number of Observations Read	12
Number of Observations Used	12

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	509.62201	101.92440	15.80	0.0021
Error	6	38.70060	6.45010		
Corrected Total	11	548.32261			

Root MSE	2.53970	R-Square	0.9294
Dependent Mean	48.38942	Adj R-Sq	0.8706
Coeff Var	5.24847		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	151.91550	31.04738	4.89	0.0027
Age	1	-0.63045	0.42503	-1.48	0.1885
Weight	1	-0.10586	0.11869	-0.89	0.4068
RunTime	1	-1.75698	0.93844	-1.87	0.1103
RunPulse	1	-0.22891	0.12169	-1.88	0.1090
RestPulse	1	-0.17910	0.13005	-1.38	0.2176

Output 76.2.2 lists the RegOut data set. Note that _TYPE_='PARMS' and _MODEL_='OXYHAT', which are from the label in the MODEL statement in PROC REG.

Output 76.2.2 OUTEST= Data Set from PROC REG Reproduced with PROC PRINT

REGRESSION SCORING EXAMPLE
OUTEST= Data Set from PROC REG

Obs	_MODEL_	_TYPE_	_DEPVAR_	_RMSE_	Intercept	Age
1	OxyHat	PARMS	Oxygen	2.53970	151.916	-0.63045

Obs	Weight	RunTime	RunPulse	Rest Pulse	Oxygen
1	-0.10586	-1.75698	-0.22891	-0.17910	-1

Output 76.2.3 lists the data sets created by PROC SCORE. Since the SCORE= data set does not contain observations with _TYPE_='MEAN' or _TYPE_='STD', the data in the Fitness data set are not standardized before scoring. The SCORE= data set contains the variable Intercept, so this intercept value is used in computing the score. To produce the RScoreP data set, the VAR statement in PROC SCORE includes only the independent variables from the model in PROC REG. As a result, the OxyHat variable contains predicted values. To produce the RScoreR data set, the VAR statement in PROC SCORE includes both the dependent variables and the independent variables from the model in PROC REG. As a result, the OxyHat variable contains negative residuals (PREDICT−ACTUAL) as shown in Output 76.2.4. If the RESIDUAL option is specified, the variable OxyHat contains positive residuals (ACTUAL−PREDICT). If the PREDICT option is specified, the OxyHat variable contains predicted values.

Output 76.2.3 Predicted Scores from the OUT= Data Set Created by PROC SCORE

REGRESSION SCORING EXAMPLE
Predicted Scores for Regression

Obs	Age	Weight	Oxygen	Run Time	Rest Pulse	Run Pulse	OxyHat
1	44	89.47	44.609	11.37	62	178	42.8771
2	40	75.07	45.313	10.07	62	185	47.6050
3	44	85.84	54.297	8.65	45	156	56.1211
4	42	68.15	59.571	8.17	40	166	58.7044
5	38	89.02	49.874	9.22	55	178	51.7386
6	47	77.45	44.811	11.63	58	176	42.9756
7	40	75.98	45.681	11.95	70	176	44.8329
8	43	81.19	49.091	10.85	64	162	48.6020
9	44	81.42	39.442	13.08	63	174	41.4613
10	38	81.87	60.055	8.63	48	170	56.6171
11	44	73.03	50.541	10.13	45	168	52.1299
12	45	87.66	37.388	14.03	56	186	37.0080

Output 76.2.4 Residual Scores from the OUT= Data Set Created by PROC SCORE

REGRESSION SCORING EXAMPLE
Negative Residual Scores for Regression

Obs	Age	Weight	Oxygen	Run Time	Rest Pulse	Run Pulse	OxyHat
1	44	89.47	44.609	11.37	62	178	-1.73195
2	40	75.07	45.313	10.07	62	185	2.29197
3	44	85.84	54.297	8.65	45	156	1.82407
4	42	68.15	59.571	8.17	40	166	-0.86657
5	38	89.02	49.874	9.22	55	178	1.86460
6	47	77.45	44.811	11.63	58	176	-1.83542
7	40	75.98	45.681	11.95	70	176	-0.84811
8	43	81.19	49.091	10.85	64	162	-0.48897
9	44	81.42	39.442	13.08	63	174	2.01935
10	38	81.87	60.055	8.63	48	170	-3.43787
11	44	73.03	50.541	10.13	45	168	1.58892
12	45	87.66	37.388	14.03	56	186	-0.38002

The second part of this example uses the parameter estimates to score a new data set. The following statements produce Output 76.2.5 and Output 76.2.6:

```
   /* The FITNESS2 data set contains observations 13-16 from */
   /* the FITNESS data set used in EXAMPLE 2 in the PROC REG */
   /* chapter.                                               */
data Fitness2;
   input Age Weight Oxygen RunTime RestPulse RunPulse;
   datalines;
45  66.45  44.754  11.12  51  176
47  79.15  47.273  10.60  47  162
54  83.12  51.855  10.33  50  166
49  81.42  49.156   8.95  44  180
;
proc print data=Fitness2;
   title 'REGRESSION SCORING EXAMPLE';
   title2 'New Raw Data Set to be Scored';
run;


proc score data=Fitness2 score=RegOut out=NewPred type=parms
           nostd predict;
   var Oxygen Age Weight RunTime RunPulse RestPulse;
run;


proc print data=NewPred;
   title2 'Predicted Scores for Regression';
   title3 'for Additional Data from FITNESS2';
run;
```

Output 76.2.5 lists the Fitness2 data set.

Output 76.2.5 Listing of the Fitness2 Data Set

REGRESSION SCORING EXAMPLE
New Raw Data Set to be Scored

Obs	Age	Weight	Oxygen	Run Time	Rest Pulse	Run Pulse
1	45	66.45	44.754	11.12	51	176
2	47	79.15	47.273	10.60	47	162
3	54	83.12	51.855	10.33	50	166
4	49	81.42	49.156	8.95	44	180

PROC SCORE scores the Fitness2 data set by using the parameter estimates in the RegOut data set. These parameter estimates result from fitting a regression equation to the Fitness data set. The NOSTD option is specified, so the raw data are not standardized before scoring. (However, the NOSTD option is not necessary here. The SCORE= data set does not contain observations with _TYPE_='MEAN' or _TYPE_='STD', so standardization is not performed.) The VAR statement contains the dependent variables and the independent variables used in PROC REG. In addition, the PREDICT option is specified. This combination gives predicted values for the new score variable. The name of the new score variable is OxyHat, from the value of the _MODEL_ variable in the SCORE= data set. Output 76.2.6 shows the data set produced by PROC SCORE.

Output 76.2.6 Predicted Scores from the OUT= Data Set Created by PROC SCORE and Reproduced Using PROC PRINT

REGRESSION SCORING EXAMPLE
Predicted Scores for Regression
for Additional Data from FITNESS2

Obs	Age	Weight	Oxygen	Run Time	Rest Pulse	Run Pulse	OxyHat
1	45	66.45	44.754	11.12	51	176	47.5507
2	47	79.15	47.273	10.60	47	162	49.7802
3	54	83.12	51.855	10.33	50	166	43.9682
4	49	81.42	49.156	8.95	44	180	47.5949

Example 76.3: Custom Scoring Coefficients

This example uses a specially created custom scoring data set and produces Output 76.3.1 and Output 76.3.2. The first scoring coefficient creates a variable that is Age−Weight; the second scoring coefficient evaluates the variable RunPulse−RstPulse; and the third scoring coefficient totals all six variables. Since the scoring coefficients data set (data set A) does not contain any observations with _TYPE_='MEAN' or _TYPE_='STD', the data in the Fitness data set (see Example 76.1) are not standardized before scoring.

The following statements produce Output 76.3.1 and Output 76.3.2:

```
data A;
   input _type_ $ _name_ $
         Age Weight RunTime RunPulse RestPulse;
   datalines;
SCORE  AGE_WGT  1 -1  0  0  0
SCORE  RUN_RST  0  0  0  1 -1
SCORE  TOTAL    1  1  1  1  1
;

proc print data=A;
   title 'CONSTRUCTED SCORING EXAMPLE';
   title2 'Scoring Coefficients';
run;


proc score data=Fitness score=A out=B;
   var Age Weight RunTime RunPulse RestPulse;
run;


proc print data=B;
   title2 'Scored Data';
run;
```

Output 76.3.1 Custom Scoring Data Set and Scored Fitness Data: PROC PRINT

CONSTRUCTED SCORING EXAMPLE
Scoring Coefficients

Obs	_type_	_name_	Age	Weight	Run Time	Run Pulse	Rest Pulse
1	SCORE	AGE_WGT	1	−1	0	0	0
2	SCORE	RUN_RST	0	0	0	1	−1
3	SCORE	TOTAL	1	1	1	1	1

Output 76.3.2 Custom Scored Fitness Data: PROC PRINT

CONSTRUCTED SCORING EXAMPLE
Scored Data

Obs	Age	Weight	Oxygen	Run Time	Rest Pulse	Run Pulse	AGE_WGT	RUN_RST	TOTAL
1	44	89.47	44.609	11.37	62	178	-45.47	116	384.84
2	40	75.07	45.313	10.07	62	185	-35.07	123	372.14
3	44	85.84	54.297	8.65	45	156	-41.84	111	339.49
4	42	68.15	59.571	8.17	40	166	-26.15	126	324.32
5	38	89.02	49.874	9.22	55	178	-51.02	123	369.24
6	47	77.45	44.811	11.63	58	176	-30.45	118	370.08
7	40	75.98	45.681	11.95	70	176	-35.98	106	373.93
8	43	81.19	49.091	10.85	64	162	-38.19	98	361.04
9	44	81.42	39.442	13.08	63	174	-37.42	111	375.50
10	38	81.87	60.055	8.63	48	170	-43.87	122	346.50
11	44	73.03	50.541	10.13	45	168	-29.03	123	340.16
12	45	87.66	37.388	14.03	56	186	-42.66	130	388.69

References

Chaseling, J. (1996), "Standard Test Results of Students at Three Types of Schools," sample data, Faculty of Environmental Sciences, Griffith University, Queensland, Australia.

Chapter 77

The SEQDESIGN Procedure

Contents

Overview: SEQDESIGN Procedure **5784**
Boundaries for Group Sequential Designs 5788
Group Sequential Methods 5790
Getting Started: SEQDESIGN Procedure **5791**
Syntax: SEQDESIGN Procedure **5799**
PROC SEQDESIGN Statement 5799
DESIGN Statement 5803
SAMPLESIZE Statement 5809
Details: SEQDESIGN Procedure **5817**
Fixed-Sample Clinical Trials 5817
One-Sided Fixed-Sample Tests in Clinical Trials 5821
Two-Sided Fixed-Sample Tests in Clinical Trials 5824
Group Sequential Methods 5827
Statistical Assumptions for Group Sequential Designs 5829
Boundary Scales 5832
Boundary Variables 5834
Type I and Type II Errors 5836
Unified Family Methods 5839
Haybittle-Peto Method 5844
Whitehead Methods 5845
Error Spending Methods 5848
Boundary Adjustments for Overlapping Lower and Upper β Boundaries . . 5850
Specified and Derived Parameters 5851
Applicable Boundary Keys 5853
Sample Size Computation 5854
Applicable One-Sample Tests and Sample Size Computation 5858
Applicable Two-Sample Tests and Sample Size Computation 5860
Applicable Regression Parameter Tests and Sample Size Computation 5868
Aspects of Group Sequential Designs 5872
Summary of Methods in Group Sequential Designs 5874
Table Output 5876
ODS Table Names 5880
Graphics Output 5880
ODS Graphics 5882

Examples: SEQDESIGN Procedure **5883**
Example 77.1: Creating Fixed-Sample Designs 5883
Example 77.2: Creating a One-Sided O'Brien-Fleming Design 5890
Example 77.3: Creating Two-Sided Pocock and O'Brien-Fleming Designs . 5896
Example 77.4: Generating Graphics Display for Sequential Designs 5906
Example 77.5: Creating Designs Using Haybittle-Peto Methods 5911
Example 77.6: Creating Designs with Various Stopping Criteria 5919
Example 77.7: Creating Whitehead's Triangular Designs 5930
Example 77.8: Creating a One-Sided Error Spending Design 5940
Example 77.9: Creating Designs with Various Number of Stages 5947
Example 77.10: Creating Two-Sided Error Spending Designs with and without Overlapping Lower and Upper β Boundaries 5955
Example 77.11: Creating a Two-Sided Asymmetric Error Spending Design with Early Stopping to Reject H_0 5961
Example 77.12: Creating a Two-Sided Asymmetric Error Spending Design with Early Stopping to Reject or Accept H_0 5970
Acknowledgments **5979**
References **5979**

Overview: SEQDESIGN Procedure

The purpose of the SEQDESIGN procedure is to design interim analyses for clinical trials. Clinical trials are experiments on human subjects to demonstrate the efficacy and safety of new drugs or treatments. A simple example is a trial to test the effectiveness of a new drug in humans by comparing the outcomes in a group of patients who receive the new drug with the outcomes in a comparable group of patients who receive a placebo.

A clinical trial is conducted according to a plan called a *protocol*. A protocol details the objectives of the trial, the data collection process, and the analysis. The protocol specifies the null hypothesis and an alternative hypothesis, a test statistic, the probability α of a Type I error, the probability β of a Type II error, the sample size needed to attain a specified power of $1 - \beta$ at an alternative reference, and critical values that are associated with the test statistic.

In a fixed-sample trial, data about all individuals are first collected and then examined at the end of the study. Most major trials have committees that periodically monitor safety and efficacy data during the trial and recommend that a trial be stopped for safety concerns such as an unacceptable toxicity level. In certain situations, the committee might recommend that a trial be stopped for efficacy. In contrast to a fixed-sample trial, a group sequential trial provides for interim analyses before the completion of the trial while maintaining the specified overall Type I and Type II error probabilities.

A group sequential trial is most useful in situations where it is important to monitor the trial to prevent unnecessary exposure of patients to an unsafe new drug, or alternatively to a placebo treatment if the new drug shows significant improvement. In most cases, if a group sequential trial stops early

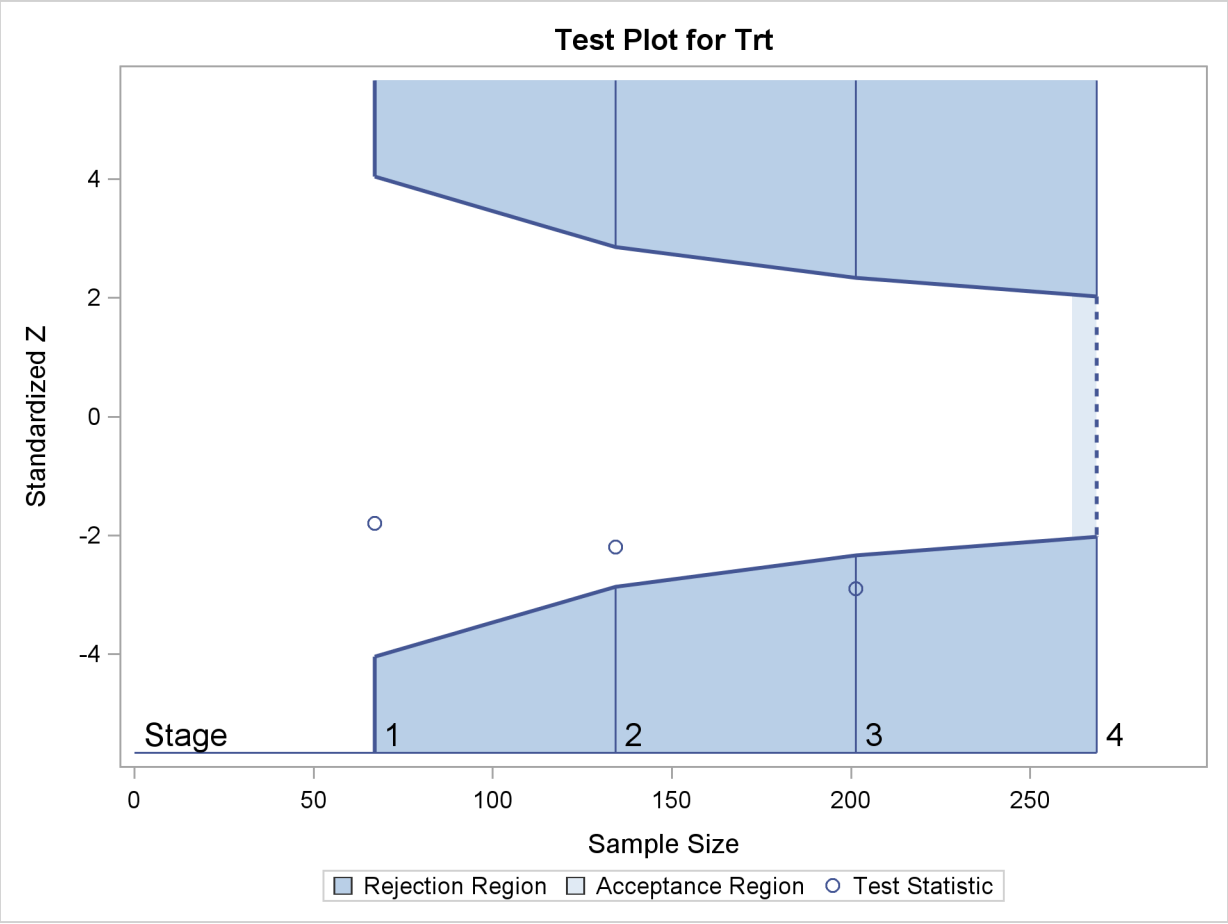
for safety concerns, fewer patients are exposed to the new treatment than in the fixed-sample trial. If a trial stops early for efficacy reasons, the new treatment is available sooner than it would be in a fixed-sample trial. Early stopping can also save time and resources.

A group sequential design provides detailed specifications for a group sequential trial. In addition to the usual specification for a fixed-sample design, it provides the total number of stages (the number of interim stages plus a final stage) and a stopping criterion to reject, to accept, or to either reject or accept the null hypothesis at each interim stage. It also provides critical values and the sample size at each stage for the trial.

At each interim stage, the data collected at the current stage in addition to the data collected at previous stages are analyzed, and statistics such as a maximum likelihood test statistic and its associated standard error are computed. The test statistic is then compared with critical values that are generated from the sequential design, and the trial is stopped or continued. If a trial continues to the final stage, the null hypothesis is either rejected or accepted. The critical values for each stage are chosen in such a way that the overall α and β are maintained at the specified levels.

Figure 77.1 shows a two-sided symmetric group sequential trial that stops early to reject the null hypothesis that the parameter Trt is zero.

Figure 77.1 Sequential Plot for Two-Sided Test



The trial has four stages, which are indicated by the vertical lines labeled 1, 2, 3, and 4. With early stopping to reject the null hypothesis, the lower rejection boundary is constructed by connecting the lower critical values for the stages. Similarly, the upper rejection boundary is constructed by connecting the upper critical values for the stages. The horizontal axis indicates the sample size for the group sequential trial, and the vertical axis indicates the values of the test statistic on the standardized Z scale.

At each interim stage, if the test statistic falls into a rejection region (the darker shaded areas in Figure 77.1), the trial stops and the null hypothesis is rejected. Otherwise, the trial continues to the next stage. At the final stage (stage 4), the null hypothesis is rejected if Z falls into a rejection region. Otherwise, the null hypothesis is not rejected. In Figure 77.1, the test statistic does not fall into the rejection regions for stages 1 and 2, and so the trial continues to stage 3. At stage 3, the test statistic falls into the rejection region, and the null hypothesis is rejected.

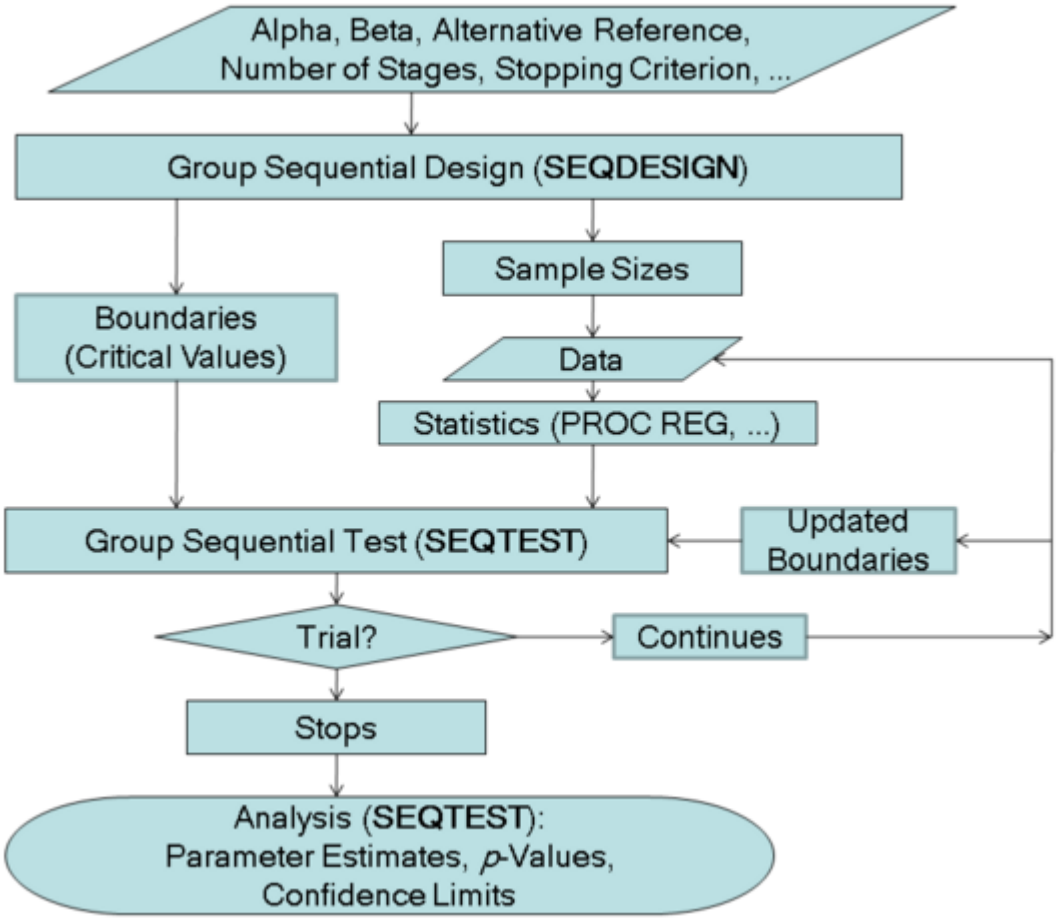
A group sequential trial usually involves six steps:

1. You specify the statistical details of the design, including the null and alternative hypotheses, a test statistic for the hypothesis test, the Type I and II error probabilities, a stopping criterion, the total number of stages, and the relative information level at each stage.
2. You compute the boundary values for the trial based on the specifications in Step 1. You also compute the sample size required at each stage for the specified hypothesis test.
3. At each stage, you collect additional data with the required sample sizes. The data available at each stage include the data collected at previous stages in addition to the data collected at the current stage.
4. At each stage, you analyze the available data with a procedure such as the REG procedure, and you compute the test statistic.
5. At each stage, you compare the test statistic with the corresponding boundary values. You stop the trial to reject or accept the hypothesis, or you continue the trial to the next stage. If you continue the trial to the final stage, you either accept or reject the hypothesis.
6. After the trial stops, you compute parameter estimates, confidence limits for the parameter, and a p-value for the hypothesis test.

You use the SEQDESIGN procedure at Step 2 to compute the initial boundary values and required sample sizes for the trial. You use the companion SEQTEST procedure at Step 5 to compare the test statistic with its boundary values. At stage 1, the boundary values are derived by using the boundary information tables created by the SEQDESIGN procedure. These boundary information tables are structured for input to the SEQTEST procedure. At each subsequent stage, the boundary values are derived by using the test information tables are created by the SEQTEST procedure at the previous stage. These test information tables are also structured for input to the SEQTEST procedure. You also use the SEQTEST procedure at Step 6 to compute parameter estimates, confidence limits, and p-values after the trial stops.

The flowchart in Figure 77.2 summarizes the steps in a typical group sequential trial and the relevant SAS procedures.

Figure 77.2 Group Sequential Trial



Features of the SEQDESIGN Procedure

The SEQDESIGN procedure assumes that the standardized Z test statistics for the null hypothesis $H_0 : \theta = 0$ at the stages have the joint canonical distribution with the information levels at the stages for the parameter θ. This implies that these test statistics are normally distributed. If the test statistic is not normally distributed, then it is assumed that the test statistic is computed from a large sample such that the statistic has an approximately normal distribution. See the section "Statistical Assumptions for Group Sequential Designs" on page 5829 for a detailed description of the joint canonical distribution.

You can use the SEQDESIGN procedure to compute required sample sizes for commonly used hypothesis tests. Note that for a fixed-sample design, you should use the POWER and GLMPOWER procedures to compute sample sizes.

The applicable tests include tests for binomial proportions and the log-rank test for two survival distributions. See the section "Applicable One-Sample Tests and Sample Size Computation" on page 5858, the section "Applicable Two-Sample Tests and Sample Size Computation" on page 5860, and the section "Applicable Regression Parameter Tests and Sample Size Computation" on page 5868 for examples of applicable tests in group sequential trials.

At each stage, the data are analyzed with a statistical procedure such as the REG procedure, and a test statistic and its associated information level are computed. The information level is the amount of information available about the unknown parameter. For a maximum likelihood statistic, the information level is the inverse of its variance.

At each stage, you use the SEQTEST procedure to derive the boundary values that correspond to the information level associated with the test statistic. You then use the SEQTEST procedure to compare the test statistic with these boundary values. When a trial is stopped at an interim stage or at the final stage, the SEQTEST procedure also derives parameter estimates, confidence limits for the parameter, and a p-value for hypothesis testing.

Output from the SEQDESIGN Procedure

In addition to computing the boundary values for a group sequential design, the SEQDESIGN procedure computes the following quantities:

- maximum sample size (as a percentage of the corresponding fixed-sample size) if the trial does not stop at an interim stage
- average sample numbers (as percentages of the corresponding fixed-sample sizes for nonsurvival data or fixed-sample numbers of events for survival data) under various hypothetical references, including the null and alternative references
- stopping probabilities at each stage under various hypothetical references to indicate how likely it is that the trial will stop at that stage
- sample sizes required at each stage for the specified hypothesis test
- numbers of events required at each stage for the specified hypothesis test with survival data

You can create more than one design with multiple DESIGN statements in the SEQDESIGN procedure and then choose the design with the most desirable features. The next two subsections introduce some basic aspects of group sequential designs that are useful for getting started with the SEQDESIGN procedure.

Boundaries for Group Sequential Designs

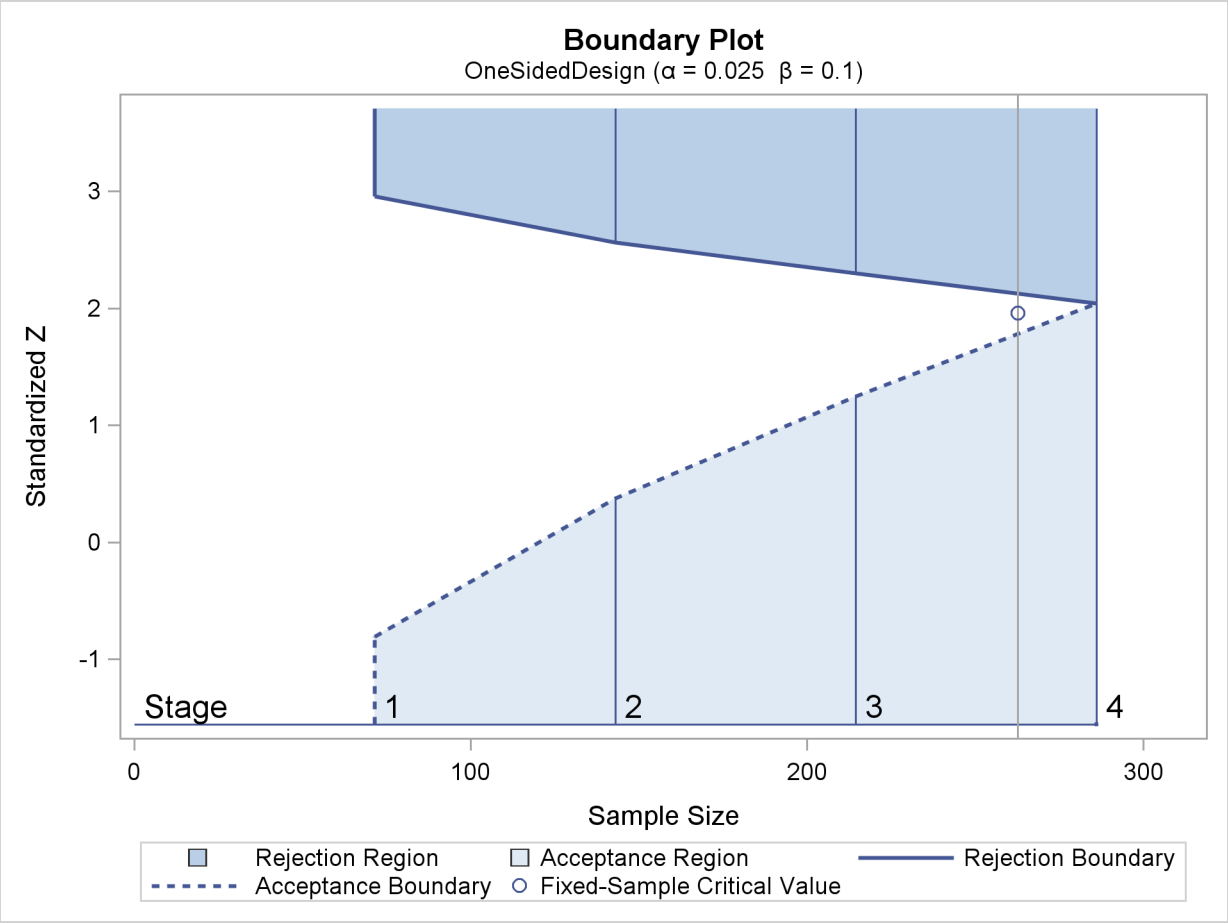
A one-sided test is a test of a hypothesis with either a lower alternative ($H_1 : \theta < 0$) or an upper alternative ($H_1 : \theta > 0$), and a two-sided test is a test with a two-sided alternative ($H_1 : \theta \neq 0$). The number of critical values for a test depends on whether the alternative is one-sided or two-sided, and it also depends on whether the trial is conducted with a fixed-sample design or a group sequential design.

For a fixed-sample trial, a one-sided test has one critical value and a two-sided test has two critical values. These critical values are computed with the specified Type I error probability α. In contrast, at each interim stage of a group sequential trial, a one-sided test has up to two critical values and a two-sided group sequential test has up to four critical values. Thus, there are two or four possible

boundaries for a group sequential design, and each boundary is a set of critical values, one from each stage.

Figure 77.3 illustrates the boundaries for a one-sided test with an upper alternative that allows for early stopping to either reject or accept the null hypothesis.

Figure 77.3 Boundary Plot for One-Sided Design



With an upper alternative, as in this example, the design has the following two boundaries: an upper α (rejection) boundary for the rejection region that consists of upper rejection critical values and an upper β (acceptance) boundary for the acceptance region that consists of upper acceptance critical values. The stages are indicated by vertical lines with accompanying stage numbers. In Figure 77.3, the horizontal axis indicates the cumulative sample size for the group sequential trial. The vertical axis indicates the critical values at each stage on the standardized Z scale. Other scales can be used for the vertical axis, including the MLE scale, score statistic scale, and p-value scale.

At each interim stage, if the test statistic is in the rejection region (darker shaded area in Figure 77.3), the trial stops and the null hypothesis is rejected. If the test statistic is in the acceptance region (lightly shaded area in Figure 77.3), the trial stops and the hypothesis is accepted. Otherwise, the trial continues to the next stage. If the trial proceeds to the final stage (stage 4), the upper α and upper β critical values are identical, and the trial stops to either reject or accept the null hypothesis.

Group Sequential Methods

For a group sequential design, there are two possible boundaries for a one-sided test and four possible boundaries for a two-sided test. Each boundary consists of one boundary value (critical value) for each stage. The SEQDESIGN procedure provides various methods for computing the boundary values.

The boundary value for a fixed-sample test cannot be applied to each stage of sequential design because, as shown by Armitage, McPherson, and Rowe (1969), repeated significance tests at a fixed level on accumulating data increase the probability of a Type I error. For example, with a fixed-sample two-sided test, the critical values ± 1.96 for a standardized Z statistic produce a Type I error probability level $\alpha = 0.05$. But for a two-sided group sequential test with two equally spaced stages, if the same critical values ± 1.96 are used to reject the null hypothesis at these two stages, the Type I error probability level is $\alpha = 0.083$, larger than the fixed-sample α.

Numerous methods are available for deriving the critical values for each boundary in a sequential design. Pocock (1977) applies repeated significance tests to group sequential trials with equal-size groups and derives a constant critical value on the standardized normal Z scale across all stages that maintains the specified Type I error probability level. O'Brien and Fleming (1979) propose a sequential procedure that has boundary values (in absolute value) decrease over the stages on the standardized normal Z scale.

The SEQDESIGN procedure provides the following three types of methods:

- fixed boundary shape methods, which derive boundaries with specified boundary shapes
- Whitehead methods, which adjust boundaries derived for continuous monitoring so that they apply to discrete monitoring
- error spending methods

Each type of methods uses a distinct approach to derive the boundary values for a group sequential trial. Whitehead methods require much less computation with resulting Type I error probability and power that are close but differ slightly from the specified values due to the approximations used in deriving the tests (Jennison and Turnbull 2000, p. 106). Fixed boundary shape methods derive boundary values by estimating a fixed number of parameters and require more computation. Error spending methods derive boundary values at each stage sequentially and require much more computation than other types of methods for group sequential trials with a large number of stages.

Within each type of methods, you can choose methods that creating boundary values range from conservative stopping boundary values at early stages to liberal stopping boundary values at very early stages.

You can use the SEQDESIGN procedure to specify methods from the same type for each design. A different method can be specified for each boundary separately, but all methods in a design must be of the same type.

Fixed Boundary Shape Methods

The fixed boundary shape methods include the unified family methods and the Haybittle-Peto method. The unified family methods (Kittelson and Emerson 1999) derive boundaries from specified boundary shapes. These methods include Pocock's method (Pocock 1977) and the O'Brien-Fleming method (O'Brien and Fleming 1979) as special cases.

The Haybittle-Peto method (Haybittle 1971; Peto et al. 1976) uses a value of 3 for the critical values in interim stages, so the critical value at the final stage is close to the critical value for the fixed-sample design. In the SEQDESIGN procedure, the Haybittle-Peto method has been generalized to allow for different boundary values at interim stages.

Whitehead Methods

Whitehead and Stratton (1983) and Whitehead (1997, 2001) develop triangular and straight-line boundaries by adapting tests constructed for continuous monitoring to discrete monitoring of group sequential tests. With continuous monitoring, the values for each boundary fall in a straight line when plotted on the score statistic scale. The discrete boundary is derived by subtracting the expected overshoot from the continuous boundary to obtain the desired Type I and Type II error probabilities. For a design with early stopping to either reject or accept the null hypothesis, the boundaries form a triangle when plotted on the score statistic scale. See the section "Score Statistic" on page 5819 for a detailed description of the score statistic.

Error Spending Methods

For every sequential design, the α and β errors at each stage can be computed from the boundary values. On the other hand, you can derive the boundary values from specified α and β errors for each stage. The error spending function approach (Lan and DeMets 1983) uses an error spending function to specify the errors at each stage for each boundary and then derives the boundary values.

Getting Started: SEQDESIGN Procedure

This section illustrates a clinical study design that uses a two-sided O'Brien-Fleming design (O'Brien and Fleming 1979) to stop the trial early for ethical concerns about possible harm or for unexpectedly strong efficacy of the new drug.

Suppose that a pharmaceutical company is conducting a clinical trial to test the efficacy of a new cholesterol-lowering drug. The primary focus is low-density lipoprotein (LDL), the so-called bad cholesterol, which is a risk factor for coronary heart disease. LDL is measured in mg/dL, milligrams per deciliter of blood.

The trial consists of two groups of equally allocated patients with elevated LDL levels: an experimental group given the new drug and a placebo control group. Suppose the changes in LDL level

after the treatment for individuals in the experimental and control groups are normally distributed with means μ_e and μ_c, respectively, and have a common variance σ^2. Then the null hypothesis of no effect for the new drug is $H_0 : \theta = 0$, where $\theta = \mu_e - \mu_c$.

For a fixed-sample design with a total sample size N, the MLE for θ is computed as $\hat{\theta} = \hat{\mu}_e - \hat{\mu}_c$, where $\hat{\mu}_e$ and $\hat{\mu}_c$ are the sample means of the decreases in LDL level in the experimental and control groups, respectively.

Following the derivation in the section "Test for the Difference between Two Normal Means" on page 5860, the statistic $\hat{\theta}$ has a normal distribution

$$\hat{\theta} \sim N\left(\theta, \frac{4\sigma^2}{N}\right)$$

Thus, under the null hypothesis $H_0 : \theta = 0$, the standardized statistic

$$Z = \frac{\hat{\theta}}{\sqrt{\frac{4\sigma^2}{N}}} \sim N(0, 1)$$

The Z statistic can be used to test the null hypothesis H_0. If the variance σ^2 is unknown, the sample variance can be used to compute the test statistic if it is assumed that the sample variance is computed from a large sample such that the Z statistic has an approximately standard normal distribution.

With a Type I error probability $\alpha = 0.05$, the critical values for the Z statistic are given by $\Phi^{-1}(\alpha/2) = -1.96$ and $\Phi^{-1}(1-\alpha/2) = 1.96$, where Φ is the cumulative standard normal distribution function. At the end of study, if $Z \geq 1.96$, the null hypothesis is rejected for harmful drug effect, and if $Z \leq -1.96$, the null hypothesis is rejected for efficacy of the new drug. Otherwise, the null hypothesis is not rejected and the drug effect is not significant.

Also suppose that for the trial, the alternative reference $\theta = -10$ is the clinically meaningful difference that the trial should detect with a high probability (power). Further suppose that a good estimate of the standard deviation for the changes in LDL level is $\hat{\sigma} = 20$. The following statements invoke the SEQDESIGN procedure and request a four-stage O'Brien-Fleming design for standardized normal test statistics:

```
ods graphics on;
 proc seqdesign altref=-10
                plots=boundary(hscale=samplesize)
                ;
   TwoSidedOBrienFleming: design nstages=4
                          method=obf
                          ;
   samplesize model=twosamplemean(stddev=20);
ods output Boundary=Bnd_LDL;
run;
ods graphics off;
```

The ALTREF= option specifies the alternative reference, and the actual maximum information is derived in the SEQDESIGN procedure. With the specified ODS GRAPHICS ON statement, the PLOTS=BOUNDARY option displays a boundary plot with the rejection and acceptance regions.

In the DESIGN statement, the label `TwoSidedOBrienFleming` identifies the design in the output tables. The default ALT=TWOSIDED option specifies a two-sided alternative hypothesis. The default STOP=REJECT option specifies early stopping in the interim stages only for the purpose of rejecting the null hypothesis. That is, at each interim stage, the trial will either be stopped to reject the null hypothesis or continue to the next stage.

The NSTAGES=4 option in the DESIGN statement specifies the total number of stages in the group sequential trial, including three interim stages and a final stage. In the SEQDESIGN procedure, the null hypothesis for the design is $H_0 : \theta = 0$. The default ALPHA=0.05 option specifies a Type I error probability $\alpha = 0.05$, and the default BETA=0.10 option specifies a Type II error probability $\beta = 0.10$, which corresponds to a power of $1-\beta = 0.90$ at the alternative reference $H_1 : \theta = -10$.

For a two-sided design with early stopping to reject the null hypothesis, there are two boundaries for the design: an upper α boundary that consists of upper rejection critical values and a lower α boundary that consists of lower rejection critical values. Each boundary is a set of critical values, one from each stage. With the METHOD=OBF option in the DESIGN statement, the O'Brien-Fleming method is used for the two boundaries for the design; see Figure 77.7.

A property of the boundaries constructed with the O'Brien-Fleming design is that the null hypothesis is more difficult to reject in the early stages than in the later stages. That is, the trial is rejected in the early stages only with overwhelming evidence, because in these stages there might not be a sufficient number of responses for a reliable estimate of the treatment effect.

The SAMPLESIZE statement with the MODEL=TWOSAMPLEMEAN option uses the derived maximum information to compute required sample sizes for a two-sample test for mean difference. The ODS OUTPUT statement with the BOUNDARY=BND_LDL option creates an output data set named BND_LDL which contains the resulting boundary information.

In a clinical trial, the amount of information about an unknown parameter available from the data can be measured by the Fisher information. For a maximum likelihood statistic, the information level is the inverse of its variance. See the section "Maximum Likelihood Estimator" on page 5818 for a detailed description of Fisher information. At each stage of the trial, data are collected and analyzed with a statistical procedure, and a test statistic and its corresponding information level are computed.

In this example, you can use the REG procedure to compute the maximum likelihood estimate $\hat{\theta}$ for the drug effect and the corresponding standard error for $\hat{\theta}$. At stage 1, you can use the SEQTEST procedure to compare the test statistic with adjusted boundaries derived from the boundary information stored in the BOUND_LDL data set. At each subsequent stage, you can use the SEQTEST procedure to compare the test statistic with adjusted boundaries derived from the boundary information stored in the test information table created by the SEQTEST procedure at the previous stage. The test information tables are structured for input to the SEQTEST procedure.

At each interim stage, the trial will either be stopped to reject the null hypothesis or continue to the next stage. At the final stage, the null hypothesis is either rejected or accepted.

By default, the SEQDESIGN procedure derives boundary values with equally spaced information levels for all stages—that is, the same information increment between successive stages. The "Design Information," "Method Information," and "Boundary Information" tables are displayed by default, as shown in Figure 77.4, Figure 77.5, and Figure 77.6, respectively.

The "Design Information" table in Figure 77.4 displays design specifications and four derived statistics: the actual maximum information, the maximum information, the average sample number under the null hypothesis (Null Ref ASN), and the average sample number under the alternative hypothesis (Alt Ref ASN). Except for the actual maximum information, each statistic is expressed as a percentage of the identical statistic for the corresponding fixed-sample information. The average sample number is the expected sample size (for nonsurvival data) or expected number of events (for survival data). Note that for a symmetric two-sided design, the ALTREF=−10 option implies a lower alternative reference of −10 and an upper alternative reference of 10.

Figure 77.4 O'Brien-Fleming Design Information

The SEQDESIGN Procedure
Design: TwoSidedOBrienFleming

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Reject Null
Method	O'Brien-Fleming
Boundary Key	Both
Alternative Reference	-10
Number of Stages	4
Alpha	0.05
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	102.2163
Max Information	0.107403
Null Ref ASN (Percent of Fixed Sample)	101.5728
Alt Ref ASN (Percent of Fixed Sample)	76.7397

The maximum information is the information level at the final stage of the group sequential trial. The Max Information (Percent Fixed-Sample) is the maximum information for the sequential design expressed as a percentage of the information for the corresponding fixed-sample design. In Figure 77.4, the Max Information (Percent Fixed-Sample) is 102.22%, which means that the information needed for the group sequential trial is 2.22% more than that of the corresponding fixed-sample design if the trial does not stop at any interim stage.

The Null Ref ASN (Percent Fixed-Sample) is the average sample number (expected sample size) required under the null hypothesis for the group sequential design expressed as a percentage of the sample size for the corresponding fixed-sample design. In Figure 77.4, the Null Ref ASN is 101.57%, which means that the expected sample size for the group sequential trial is 1.57% greater than the corresponding fixed-sample size.

Similarly, the Alt Ref ASN (Percent Fixed-Sample) is the average sample number (expected sample size) required under the alternative hypothesis for the group sequential design expressed as a percentage of the sample size for the corresponding fixed-sample design. In Figure 77.4, the Alt Ref ASN is 76.74%, which means that the expected sample size for the group sequential trial is 76.74% of the corresponding fixed-sample size. That is, if the alternative hypothesis is true, then on average, only 76.74% of the fixed-sample size is needed for the group sequential trial.

In this example, the O'Brien-Fleming design requires only a slight increase in sample size if the trial proceeds to the final stage. On the other hand, if the alternative hypothesis is correct, this design provides a substantial saving in sample size on average.

The "Method Information" table in Figure 77.5 displays the computed Type I and Type II error probabilities α and β, and the derived drift parameter for the design. For a two-sided test with early stopping to reject the null hypothesis, both lower and upper α boundaries are created. With the specified ALTREF= option, the alternative references are also included.

With the zero null reference, the drift parameter is the standardized alternative reference at the final stage $\theta_1\sqrt{I_X}$, where θ_1 is the alternative reference and I_X is the maximum information. See the section "Specified and Derived Parameters" on page 5851 for a detailed description of the drift parameter. The drift parameters for the design are derived in the SEQDESIGN procedure even if the alternative reference is not specified or derived in the procedure.

Figure 77.5 Method Information

Method Information

Boundary	Method	Alpha	Beta	Unified Family Rho	Unified Family Tau	Unified Family C
Upper Alpha	O'Brien-Fleming	0.02500	0.10000	0.5	0	2.02429
Lower Alpha	O'Brien-Fleming	0.02500	0.10000	0.5	0	2.02429

Method Information

Boundary	Alternative Reference	Drift
Upper Alpha	10	3.277238
Lower Alpha	-10	-3.27724

The O'Brien-Fleming method belongs to the unified family of designs, which is parameterized by two parameters, ρ and τ, as implemented in the SEQDESIGN procedure. See Table 77.3 for parameter values of commonly used methods in the unified family. The "Method Information" table in Figure 77.5 displays the values of $\rho = 0.5$ and $\tau = 0$, which are the parameters for the O'Brien-Fleming method. The table also displays the derived parameter $C_\alpha = 2.0243$, which is used in the construction of symmetric lower and upper α boundaries; see the section "Unified Family Methods" on page 5839.

The "Boundary Information" table in Figure 77.6 displays the information level, including the proportion, actual level, and corresponding sample size (N) at each stage. The table also displays the lower and upper alternative references, and the lower and upper boundary values at each stage. By default, equally spaced information levels for all stages are used to derive boundary values.

Figure 77.6 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

	Information Level			Alternative Reference	
Stage	Proportion	Actual	N	Lower	Upper
1	0.2500	0.026851	42.96116	−1.63862	1.63862
2	0.5000	0.053701	85.92233	−2.31736	2.31736
3	0.7500	0.080552	128.8835	−2.83817	2.83817
4	1.0000	0.107403	171.8447	−3.27724	3.27724

Boundary Information (Standardized Z Scale)
Null Reference = 0

	Boundary Values	
	Lower	Upper
Stage	Alpha	Alpha
1	−4.04859	4.04859
2	−2.86278	2.86278
3	−2.33745	2.33745
4	−2.02429	2.02429

The information proportion is the proportion of maximum information available at each stage and N is the corresponding sample size. The default BOUNDARYSCALE=STDZ option specifies that the standardized Z scale be used to display the boundary values in the boundary information table and the boundary plot. The alternative reference on the standardized Z scale at stage k is given by $\theta_1\sqrt{I_k}$, where θ_1 is the alternative reference and I_k is the information available at stage k, $k = 1, 2, 3, 4$. These standardized alternative references for the design are derived in the SEQDESIGN procedure even if the alternative reference is not specified or derived in the procedure.

In this example, a standardized Z statistic is computed by standardizing the parameter estimate of the effect in LDL level. A lower Z test statistic indicates a beneficial effect. Consequently, at each interim stage, if the standardized Z test statistic is less than or equal to the corresponding lower α boundary value, the hypothesis $H_0 : \theta = 0$ is rejected for efficacy. If the test statistic is greater than or equal to the corresponding upper α boundary value, the hypothesis H_0 is rejected for harmful effect. Otherwise, the process continues to the next stage. At the final stage (stage 4), the hypothesis H_0 is rejected for efficacy if the Z statistic is less than or equal to the corresponding lower α boundary value -2.0243, and the hypothesis H_0 is rejected for harmful effect if the Z statistic is greater than or equal to the corresponding upper α boundary value 2.0243. Otherwise, the hypothesis of no significant difference is accepted.

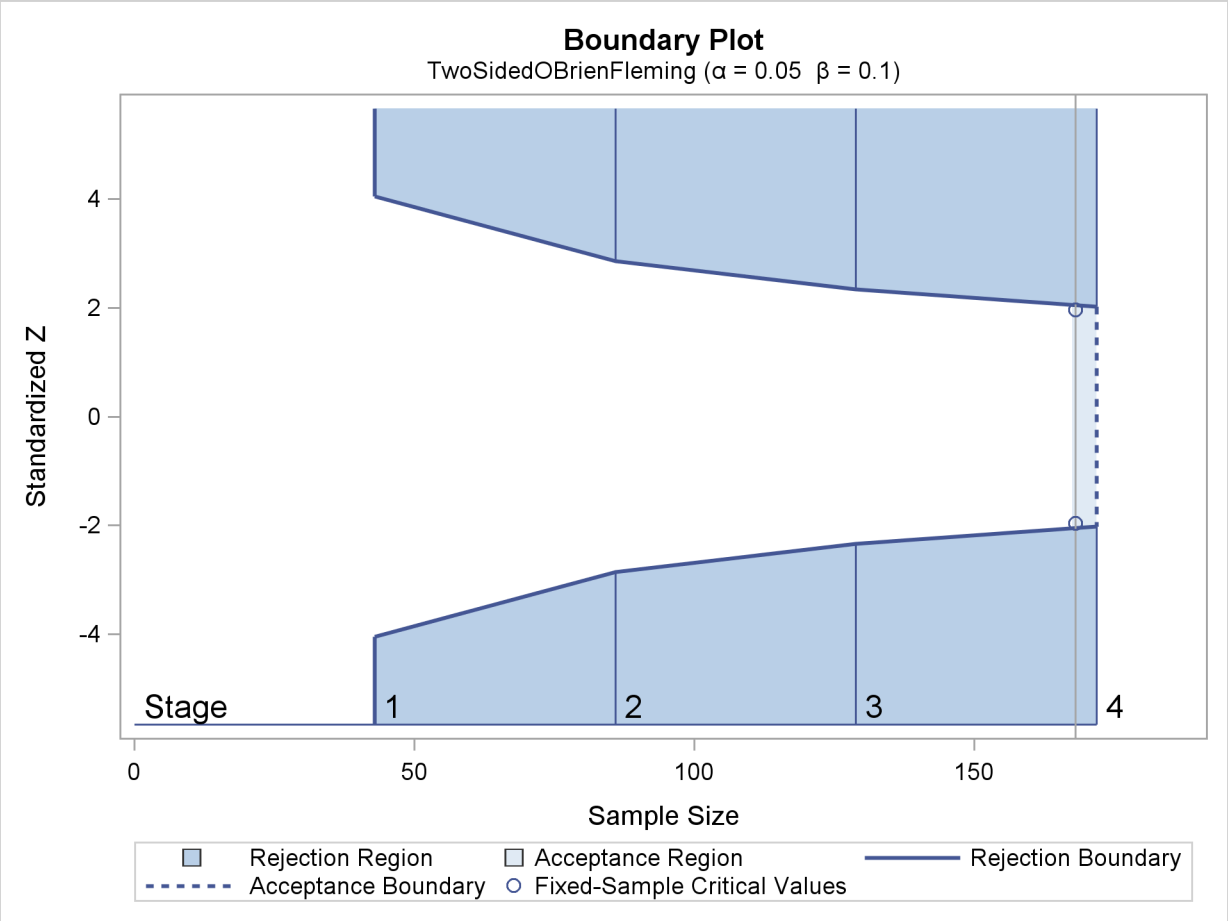
Note that in a typical trial, the actual information levels do not match the information levels specified in the design. The SEQTEST procedure modifies the boundary values stored in the BOUND_LDL data set to adjust for these new information levels.

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Figure 77.7. This plot displays the boundary values in the "Boundary Information" table in Figure 77.6. The stages are indicated by vertical lines

with accompanying stage numbers. The horizontal axis indicates the sample sizes for the stages. Note that comparing with a fixed-sample design, only a small increase in sample size is needed for the O'Brien-Fleming design, as shown in Figure 77.7.

If a test statistic at an interim stage is in the rejection region (shaded area), the trial stops and the null hypothesis is rejected. If the statistic is not in any rejection region, the trial continues to the next stage.

Figure 77.7 Boundary Plot



The boundary plot also displays critical values for the corresponding fixed-sample design. The symbol "∘" identifies the fixed-sample critical values of -1.96 and 1.96, and the accompanying vertical line indicates the required sample size for the fixed-sample design at the horizontal axis. Note that the boundary values ± 2.0243 at the final stage are close to the fixed-sample critical values ± 1.96.

When you specify the SAMPLESIZE statement, the maximum information (either explicitly specified or derived in the SEQDESIGN procedure) is used to compute the required sample sizes for the study. The MODEL=TWOSAMPLEMEAN(STDDEV=20) option specifies the test for the difference between two normal means. See the section "Test for the Difference between Two Normal Means" on page 5860 for a detailed derivation of these required sample sizes.

The "Sample Size Summary" table in Figure 77.8 displays the parameters for the sample size computation and the resulting maximum and expected sample sizes.

Figure 77.8 Sample Size Summary

Sample Size Summary

Test	Two-Sample Means
Mean Difference	-10
Standard Deviation	20
Max Sample Size	171.8447
Expected Sample Size (Null Ref)	170.7627
Expected Sample Size (Alt Ref)	129.0137

The "Sample Sizes (N)" table in Figure 77.9 displays the required sample sizes at each stage for the trial, in both fractional and integer numbers. The derived fractional sample sizes are displayed under the heading "Fractional N." These sample sizes are rounded up to integers under the heading "Ceiling N." With the default WEIGHT=1 option in the SAMPLESIZE statement, the sample sizes for the two groups are equal for the two-sample test.

Figure 77.9 Derived Sample Sizes

Sample Sizes (N)
Two-Sample Z Test for Mean Difference

	Fractional N			
Stage	N	N(Grp 1)	N(Grp 2)	Information
1	42.96	21.48	21.48	0.0269
2	85.92	42.96	42.96	0.0537
3	128.88	64.44	64.44	0.0806
4	171.84	85.92	85.92	0.1074

Sample Sizes (N)
Two-Sample Z Test for Mean Difference

	Ceiling N			
Stage	N	N(Grp 1)	N(Grp 2)	Information
1	44	22	22	0.0275
2	86	43	43	0.0538
3	130	65	65	0.0812
4	172	86	86	0.1075

In practice, integer sample sizes are used in the trial, and the resulting information levels increase slightly. Thus, 22, 43, 65, and 86 individuals are needed in each of the two groups for the four stages, respectively.

Syntax: SEQDESIGN Procedure

The following statements are available in PROC SEQDESIGN:

PROC SEQDESIGN < *options* > **;**
  < *label:* > **DESIGN** *options* **;**
  SAMPLESIZE < *MODEL= option* > **;**

The PROC SEQDESIGN statement and the DESIGN statement are required for the SEQDESIGN procedure. Each DESIGN statement requests a new group sequential design, and multiple DESIGN statements can be used to create more than one design for comparison of features. The label, which must be a valid SAS name, is used to identify the design in the output tables and graphics. The SAMPLESIZE statement computes the required sample sizes for the design specified in each DESIGN statement. With a selected design, the SAMPLESIZE statement computes the required sample sizes for the trial.

PROC SEQDESIGN Statement

PROC SEQDESIGN < *options* > **;**

Table 77.1 summarizes the options in the PROC SEQDESIGN statement.

Table 77.1 Summary of PROC SEQDESIGN Options

Option	Description
Design Parameters	
ALTREF=	specifies alternative reference
BOUNDARYSCALE=	specifies statistic scale for the boundary
MAXINFO=	specifies maximum information level
Table Output	
ERRSPEND	displays cumulative error spending at each stage
PSS	displays powers and expected sample sizes
STOPPROB	displays expected cumulative stopping probabilities
Graphics Output	
PLOTS=ASN	displays expected sample numbers plot
PLOTS=BOUNDARY	displays detailed boundary plot
PLOTS=COMBINEDBOUNDARY	displays combined boundary plot
PLOTS=ERRSPEND	displays error spending plot
PLOTS=POWER	displays powers plot

By default, the SEQDESIGN procedure displays tables of design information, method information, and boundary information for each specified design. If the ODS GRAPHICS ON statement is specified, it also displays a detailed boundary plot.

In addition, you can use output options to display output tables such as expected cumulative stopping probability at each stage under various hypothetical references. If the ODS GRAPHICS ON statement is specified, you can also use output options to display plots such as powers and expected sample sizes under various hypothetical references.

The following options can be used in the PROC SEQDESIGN statement to derive boundary values for all sequential designs in the procedure. They are listed in alphabetical order.

ALTREF=θ_1 < (< LOWER=θ_{1l} > < UPPER=θ_{1u} >) >

specifies the alternative reference—that is, the hypothetical reference under the alternative hypothesis at which the power is computed. The LOWER= and UPPER= options are applicable only for a two-sided design with different lower and upper alternative references.

For a one-sided design, $\theta_{1l} = -|\theta_1|$ is the lower alternative reference and $\theta_{1u} = |\theta_1|$ is the upper alternative reference. For a two-sided design, the specified θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively. If the LOWER= option is not specified, $\theta_{1l} = -|\theta_1|$, and if the UPPER= option is not specified, $\theta_{1u} = |\theta_1|$.

The specification of the ALTREF= option depends on the hypothesis used in the clinical trial. For example, suppose the null hypothesis $H_0 : \theta = 0$ with an alternative hypothesis $H_1 : \theta = \theta_1$ is used to compare two binomial populations, $p_a = p_b$. Then θ_1 is the proportion difference under H_1 if $\theta = p_a - p_b$, and θ_1 is the log odds ratio under H_1 if $\theta = \log\left(\frac{p_a(1-p_b)}{p_b(1-p_a)}\right)$.

If the ALTREF= option is not specified, the alternative reference θ_1 can also be specified or derived in the SAMPLESIZE statement. If θ_1 is specified or derived in the SAMPLESIZE statement, $\theta_{1l} = -|\theta_1|$ and $\theta_{1u} = |\theta_1|$ are the lower and upper alternative references, respectively.

Note that if the SAMPLESIZE statement is specified with a two-sided design, the sample sizes derived by using the lower and upper alternatives might be different. If θ_1 is specified or derived in the SAMPLESIZE statement, it is used to compute the sample sizes. Otherwise, the θ_1 specified in the ALTREF= option is used.

BOUNDARYSCALE=MLE | SCORE | STDZ | PVALUE

BSCALE=MLE | SCORE | STDZ | PVALUE

specifies the scale for the statistic that is displayed in the boundary table and boundary plots. The keywords MLE, SCORE, STDZ, and PVALUE correspond to the boundary with the maximum likelihood estimate scale, the score statistic scale, the standardized normal Z scale, and the p-value scale, respectively. The default is BOUNDARYSCALE=STDZ.

With the BOUNDARYSCALE=MLE or BOUNDARYSCALE=SCORE option, the maximum information must be either explicitly specified with the MAXINFO= option or derived in the SEQDESIGN procedure to provide the necessary information level at each stage to compute the boundary values. See the section "Boundary Scales" on page 5832 for a detailed description of the statistic scale for the boundary values.

Note that for a two-sided design, the p-value scale displays the one-sided fixed-sample p-value under the null hypothesis with a lower alternative hypothesis.

MAXINFO=*number*

specifies the maximum information level for the design. If the MAXINFO=option is specified and the alternative reference is either specified explicitly with the ALTREF= option or derived from the SAMPLESIZE statement, then the Type I and Type II error probability levels cannot be met simultaneously. In this case, the ALPHA= option in the DESIGN statement is applicable only with the default BOUNDARYKEY=ALPHA option in the DESIGN statement, and the Type II error probability β is derived. The BETA= option in the DESIGN statement is applicable only with the BOUNDARYKEY=BETA option in the DESIGN statement, and the Type I error probability α is derived.

Table Output Options

The following options can be used in the PROC SEQDESIGN statement to display addition table output. They are listed in alphabetical order.

ERRSPEND

displays the error spending at each stage for each boundary in the design.

PSS < (CREF= *numbers* **) >**

displays powers and expected sample sizes under various hypothetical references, where the numbers $c_i \geq 0$.

For a one-sided design, the power and expected sample sizes under hypotheses $\theta = c_i\,\theta_1$ are displayed, where θ_1 is the alternative reference and c_i are the values specified in the CREF= option.

For a two-sided design, the power and expected sample sizes under hypotheses $\theta = c_i\theta_{1l}$ and $\theta = c_i\theta_{1u}$ are displayed, where θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively. The default is CREF= 0 0.5 1.0 1.5.

Note that for a symmetric two-sided design, only the power and expected sample sizes under hypotheses $\theta = c_i\,\theta_{1u}$ are derived. See the section "Type I and Type II Errors" on page 5836 for a detailed description of the power computation. See the section "Powers and Expected Sample Sizes" on page 5878 for a detailed description of the expected sample size computation.

STOPPROB < (CREF= *numbers* **) >**

displays expected cumulative stopping probabilities under various hypothetical references, where the numbers $c_i \geq 0$.

For a one-sided design, expected cumulative stopping probabilities at each stage under hypotheses $\theta = c_i\,\theta_1$ are displayed, where θ_1 is the alternative reference and c_i are the values specified in the CREF= option.

For a two-sided design, expected cumulative stopping probabilities at each stage under hypotheses $\theta = c_i\theta_{1l}$ and $\theta = c_i\theta_{1u}$ are displayed, where θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively. Note that for a symmetric two-sided design, only expected cumulative stopping probabilities under hypotheses $\theta = c_i\,\theta_{1u}$ are derived. The default is CREF= 0 0.5 1.0 1.5.

Graphics Output Options

This section describes the options for using ODS Graphics with the SEQDESIGN procedure to create plots. To request these graphs, you must specify the ODS GRAPHICS ON statement in addition to the following options in the PROC SEQDESIGN statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

The following options can be used in the PROC SEQDESIGN statement to display graphs with ODS Graphics. They are listed in alphabetical order.

PLOTS < (ONLY) > < = *plot-request* >

PLOTS < (ONLY) > < = (*plot-request* < . . . *plot-request* >) >

specifies options that control the details of the plots. The default is PLOTS=BOUNDARY. The global plot option ONLY suppresses the default plots and displays only plots specifically requested.

The plot request options are as follows.

ALL

produces all appropriate plots.

ASN < (CREF= *numbers*) >

displays a plot of the average sample numbers (expected sample sizes for nonsurvival data or expected numbers of events for survival data) under various hypothetical references, where the numbers $c_i \geq 0$. These average sample numbers are displayed as percentages of the average sample numbers for the corresponding fixed-sample design.

For a one-sided design, expected sample numbers under hypotheses $\theta = c_i\ \theta_1$ are displayed, where θ_1 is the alternative reference and c_i are the values specified in the CREF= option.

For a two-sided design, expected sample numbers under hypotheses $\theta = c_i \theta_{1l}$ and $\theta = c_i \theta_{1u}$ are displayed, where θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively. Note that for a symmetric two-sided design, only the average sample numbers under hypotheses $\theta = c_i\ \theta_{1u}$ are derived. The default is CREF= 0 to 1.5 by 0.01.

BOUNDARY < (HSCALE=INFO | SAMPLESIZE) >

displays a plot of the resulting sequential boundaries with the acceptance and rejection regions for each design. Either the information level (HSCALE=INFO) or the sample size (HSCALE=SAMPLESIZE) is displayed on the horizontal axis. If the maximum information is not available for the design, the information in percentage of its corresponding fixed-sample design are used in the plot. The stage number for each stage is displayed inside the plot. The default is HSCALE=INFO.

If the HSCALE=SAMPLESIZE option is specified. the SAMPLESIZE statement must also be specified. The options MODEL=INPUTNEVENTS, MODEL=TWOSAMPLESURVIVAL, and MODEL=PHREG in the SAMPLESIZE statement indicate survival data. For a sample that does not contain survival data, the sample size at each stage is displayed on the horizontal axis. For survival data, the number of events is displayed on the horizontal axis at each stage. The critical values for the corresponding fixed-sample design are also displayed in the plot.

COMBINEDBOUNDARY < (HSCALE=INFO | SAMPLESIZE | STAGE) >

displays a plot of the resulting sequential boundaries for all designs simultaneously. You can

display the information level (HSCALE=INFO), the sample size (HSCALE=SAMPLESIZE), or the stage number (HSCALE=STAGE) on the horizontal axis. The default is HSCALE=INFO. With HSCALE=INFO, if the maximum information is not available for the design, then the information in percentage of its corresponding fixed-sample design is used in the plot.

If the HSCALE=SAMPLESIZE option is specified, the SAMPLESIZE statement must also be specified. The options MODEL=INPUTNEVENTS, MODEL=TWOSAMPLESURVIVAL, and MODEL=PHREG in the SAMPLESIZE statement indicate survival data. For a sample that does not contain survival data, the sample size at each stage is displayed on the horizontal axis. For survival data, the number of events is displayed on the horizontal axis at each stage.

ERRSPEND < (HSCALE=INFO | STAGE) >

displays a plot of the error spending for all sequential boundaries in the designs simultaneously. You can display the information level (HSCALE=INFO) or the stage number (HSCALE=STAGE) on the horizontal axis. With HSCALE=INFO, the information fractions are used in the plot. The default is HSCALE=STAGE.

NONE

suppresses all plots.

POWER < (CREF= *numbers* **) >**

displays a plot of the power curves under various hypothetical references, where the numbers $c_i \geq 0$.

For a one-sided design, powers under hypotheses $\theta = c_i\,\theta_1$ are displayed, where θ_1 is the alternative reference and c_i are the values specified in the CREF= option.

For a two-sided design, powers under hypotheses $\theta = c_i\theta_{1l}$ and $\theta = c_i\theta_{1u}$ are displayed, where θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively. Note that for a symmetric two-sided design, only powers under hypotheses $\theta = c_i\,\theta_{1u}$ are derived. The default is CREF= 0 to 1.5 by 0.01.

DESIGN Statement

< label: > **DESIGN** *< options >* **;**

The DESIGN statement requests a new group sequential design. You can use multiple DESIGN statements, and each DESIGN statement corresponds to a separate group sequential design.

Table 77.2 lists the options available in the DESIGN statement.

Table 77.2 Design Statement Options

Option	Description
Design Parameters	
ALPHA=	specifies Type I error probability level α
ALT=	specifies type of alternative hypothesis
BETA=	specifies Type II error probability level β

Table 77.2 *continued*

Option	Description
BETAOVERLAP=	checks for overlapping of the lower and upper β boundaries in a two-sided design with error spending methods
BOUNDARYKEY=	specifies type of error probability to maintain
INFO=	specifies information levels
NSTAGES=	specifies number of stages
STOP=	specifies condition for early stopping
Boundary Methods	
METHOD=	specifies methods for boundary values

The required NSTAGES= option specifies the number of stages. The METHOD= option is required if the number of stages specified in the NSTAGES= option is greater than one. The following options can be used in the DESIGN statement. They are listed in alphabetical order.

ALPHA=α < (< LOWER=α_l > < UPPER=α_u >) >

specifies the Type I error probability α. The default is $\alpha = 0.05$. The LOWER= and UPPER= options are applicable only for the two-sided design. The LOWER= option specifies the lower Type I error probability α_l, and the upper Type I error probability is computed as $\alpha_u = \alpha - \alpha_l$. The UPPER= option specifies the upper Type I error probability α_u, and the lower Type I error probability is computed as $\alpha_l = \alpha - \alpha_u$. If both LOWER= and UPPER= options are not specified, $\alpha_l = \alpha_u = \alpha/2$.

If both the MAXINFO= and ALTREF= options are specified, then the Type I and Type II error probability levels cannot be met simultaneously. In this case, the ALPHA= option is applicable only with the default BOUNDARYKEY=ALPHA option, and the Type II error probability β is derived.

ALT=LOWER | UPPER | TWOSIDED

specifies the type of alternative hypothesis in the design. For a test of $H_0 : \theta = 0$, the keywords LOWER, UPPER, and TWOSIDED correspond to the alternatives of $\theta < 0$, $\theta > 0$, and $\theta \neq 0$, respectively. The default is ALT=TWOSIDED.

BETA=β < (< LOWER=β_l > < UPPER=β_u >) >

specifies the Type II error probability level β. The default is $\beta = 0.10$. The LOWER= and UPPER= options are applicable only for the two-sided design. The LOWER= option specifies the lower Type II error probability level β_l, and the UPPER= option specifies the upper Type II error probability level β_u. If the LOWER= or UPPER= option is not specified, β is used.

If both the MAXINFO= and ALTREF= options are specified, then the Type I and Type II error probability levels cannot be met simultaneously. In this case, the BETA= option is applicable only with the BOUNDARYKEY=BETA option, and the Type I error probability α is derived.

BETAOVERLAP=ADJUST | NOADJUST

OVERLAP=ADJUST | NOADJUST

specifies whether to check for overlapping of the lower and upper β boundaries for the two corresponding one-sided tests. This option applies to two-sided designs with STOP=ACCEPT or STOP=BOTH that are constructed with error spending methods, and this

type of overlapping might result from a small β spending at an interim stage. When you specify BETAOVERLAP=ADJUST, the procedure checks for this type of overlapping. If such overlapping is found, the β boundaries for the two-sided design at that stage are set to missing, and the β spending values at subsequent stages are adjusted, as described in the section "Boundary Adjustments for Overlapping Lower and Upper β Boundaries" on page 5850".

You can specify BETAOVERLAP=NOADJUST to request that no adjustment be made. The default is BETAOVERLAP=ADJUST.

BOUNDARYKEY=ALPHA | BETA | BOTH | NONE

specifies types of errors to be maintained in the resulting boundary. The default is BOUNDARYKEY=ALPHA if both ALTREF= and MAXINFO= options are specified. Otherwise, the default is BOUNDARYKEY=NONE for Whitehead methods with the STOP=BOTH option, and it is BOUNDARYKEY=BOTH for others.

See the section "Applicable Boundary Keys" on page 5853 for a detailed description of applicable boundary keys.

INFO=EQUAL

INFO=CUM(*numbers* **)**

specifies relative information levels for all stages in the design. The INFO=EQUAL option specifies equally spaced information levels, and the INFO=CUM option specifies cumulative relative information levels. The default is INFO=EQUAL.

If the number of information levels specified in the INFO=CUM option is less than the number of stages specified in the NSTAGES= option, the last available information increment is used as the information increment for each subsequent stage.

METHOD=WHITEHEAD < (TAU=τ < (<LOWER=τ_l > <UPPER=τ_u >) >) >

METHOD=*method*

METHOD(*boundary***) =** *method*

specifies the methods for the boundaries in the design, where $0 \leq \tau < 0.5$.

For a one-sided design, an α boundary is created with the STOP=REJECT or STOP=BOTH option, and a β boundary is created with the STOP=ACCEPT or STOP=BOTH option. For a two-sided design, lower and upper α boundaries are created with the STOP=REJECT or STOP=BOTH option, and lower and upper β boundaries are created with the STOP=ACCEPT or STOP=BOTH option.

There are three types of methods available in the SEQDESIGN procedure. The unified family methods and Haybittle-Peto methods derive boundary values with fixed boundary shape; the Whitehead methods derive boundary values by adjusting the boundary values generated from continuous monitoring; and the error spending methods derive the boundary values from the specified errors used at each stage. You can specify different methods for the same design, but all methods must be from the same group.

For a design with early stopping to reject or accept the null hypothesis, the METHOD=WHITEHEAD option uses Whitehead's triangular design and double-triangular design for a one-sided design and two-sided design, respectively (Whitehead and Stratton 1983; Whitehead 1997, 2001). For a design with early stopping only to reject the null hypothesis or only to accept the null hypothesis, you can specify the slope of the boundary line in the score statistic scale with the TAU= τ option. The default is TAU=0.25. See the section "Whitehead Methods" on page 5845 for a detailed description of the Whitehead methods.

The following options specify available error spending methods for the boundary. Each of these methods can be specified with the METHOD= option for all boundaries, or with the METHOD(*boundary*) = option for an individual boundary. See the section "Error Spending Methods" on page 5848 for a detailed description of these error spending methods.

ERRFUNCGAMMA < (GAMMA= γ **) >**
specifies a gamma cumulative error spending function for the boundary (Hwang, Shih, and DeCani 1990). The GAMMA= option specifies the gamma parameter γ in the function, where $\gamma \leq 3$. The boundaries created with $\gamma = 1$ are similar to the boundaries from the Pocock method, and the boundaries created with $\gamma = -4$ or $\gamma = -5$ are similar to the boundaries from the O'Brien-Fleming method. The default is GAMMA=−2, which is the average of $\gamma = 1$ and $\gamma = -5$.

ERRFUNCOBF
specifies the O'Brien-Fleming-type cumulative error spending function for the boundary (Lan and DeMets 1983).

ERRFUNCPOC
specifies the Pocock-type cumulative error spending function for the boundary (Lan and DeMets 1983).

ERRFUNCPOW < (RHO=ρ **) >**
specifies a power cumulative error spending function for the boundary (Jennison and Turnbull 2000, p. 148). The RHO= option specifies the power parameter ρ in the function, where $\rho \geq 0.25$. The boundaries created with $\rho = 1$ are similar to the boundaries from the Pocock method, and the boundaries created with $\rho = 3$ are similar to the boundaries from the O'Brien-Fleming method. The default is RHO=2, which is the average of $\rho = 1$ and $\rho = 3$.

ERRSPEND (*numbers* **)**
specifies the relative cumulative error spending at each stage.

With a fixed boundary shape, you can use the following available Haybittle-Peto methods and unified family methods to derive the boundary. You can specify each of these methods in the METHOD= option for all boundaries, or in the METHOD(*boundary*) = option for an individual boundary. See the section "Haybittle-Peto Method" on page 5844 for a detailed description of the Haybittle-Peto methods, and see the section "Unified Family Methods" on page 5839 for a detailed description of unified family methods.

HP | HAYBITTLE | PETO < (Z= *numbers* **| PVALUE=** *numbers* **) >**
specifies the Haybittle-Peto method (Haybittle 1971; Peto et al. 1976). The values specified are used to create the boundary values. The boundary value at the final stage

can be derived in the procedure to maintain the Type I and Type II error probability levels. The default is Z=3.

OBF | OBRIENFLEMING
specifies the O'Brien-Fleming method (O'Brien and Fleming 1979). The O'Brien-Fleming method is equivalent to a power family method with RHO=0.5.

POC | POCOCK
specifies the Pocock method (Pocock 1977). The Pocock method is equivalent to a power family method with RHO=0.

POW | POWER < (RHO=ρ) >
specifies a power family method (Wang and Tsiatis 1987; Emerson and Fleming 1989; Pampallona and Tsiatis 1994). The RHO= option specifies the power parameter ρ in the power family method, where $\rho \geq -0.25$. The power family method with $\rho = 0$ corresponds to the Pocock method, and the power family method with $\rho = 0.5$ corresponds to the O'Brien-Fleming method. The default is RHO=0.25, a value halfway between the Pocock and O'Brien-Fleming methods. A power family method is equivalent to a unified family method with RHO=ρ and TAU=0.

TRI | TRIANGULAR < (TAU=τ) >
specifies a unified family triangular method (Kittelson and Emerson 1999), where $0 \leq \tau \leq 1$. The default is TAU=1.0. The triangular method is identical to the unified family method with RHO=0.5 and TAU=τ. Note that this unified family triangular method is different from Whitehead's triangular method.

UNI | UNIFIED < (< TAU=τ > < RHO=ρ >) >
specifies a unified family method (Kittelson and Emerson 1999). The TAU= and RHO= options specify the τ and ρ parameters in a unified family method, respectively, where $\rho \geq 0$ and $0 \leq \tau \leq 2\rho$. The defaults are TAU=0 and RHO=0.25. See the section "Unified Family Methods" on page 5839 for a detailed description of the unified family methods.

The O'Brien-Fleming, Pocock, power family, and triangular methods are all special cases of the unified family methods. Table 77.3 summarizes the corresponding parameters in the unified family for these methods.

Table 77.3 Parameters in the Unified Family for Various Methods

Method	Option	Unified Family	
		Rho	Tau
Pocock	POC	0	0
O'Brien-Fleming	OBF	0.5	0
Power family	POW (RHO=ρ)	ρ	0
Triangular	TRI (TAU=τ)	0.5	τ

Note that the power parameter $\rho = 1/2-\Delta = \rho^*-1/2$, where Δ is the power parameter used in Jennison and Turnbull (2000) and Wang and Tsiatis (1987) and ρ^* is the power parameter used in Kittelson and Emerson (1999).

If a method with specified parameters is used for all boundaries in the design, you can use the METHOD= option to specify the method. Otherwise, you can use the following METHOD(*boundary*)= options to specify different methods from the same group for the boundaries.

METHOD(ALPHA)=*method*

METHOD(REJECT)=*method*

specifies the method for the α boundary of a one-sided design or the lower and upper α boundaries for a two-sided design.

METHOD(LOWERALPHA)=*method*

METHOD(LOWERREJECT)=*method*

specifies the method for the lower α boundary of a two-sided design.

METHOD(UPPERALPHA)=*method*

METHOD(UPPERREJECT)=*method*

specifies the method for the upper α boundary of a two-sided design.

METHOD(BETA)=*method*

METHOD(ACCEPT)=*method*

specifies the method for the β boundary of a one-sided design or the lower and upper β boundaries for a two-sided design.

METHOD(LOWERBETA)=*method*

METHOD(LOWERACCEPT)=*method*

specifies the method for the lower β boundary of a two-sided design.

METHOD(UPPERBETA)=*method*

METHOD(UPPERACCEPT)=*method*

specifies the method for the upper β boundary of a two-sided design.

NSTAGES=*number*

specifies the number of stages for the design. This option is required in the DESIGN statement, and the maximum allowed number of stages is 25.

STOP=ACCEPT | REJECT | BOTH

specifies the condition of early stopping for the design. The keywords ACCEPT, REJECT, and BOTH correspond to early stopping only to accept, only to reject, and either to accept or reject the null hypothesis H_0, respectively. The default is STOP=REJECT.

SAMPLESIZE Statement

SAMPLESIZE < *MODEL= option* > **;**

If each observation in the data set provides one unit of information in a hypothesis testing such as a one-sample test for the mean, the SAMPLESIZE statement computes the required sample sizes for the sequential design specified in each DESIGN statement. However, for a survival analysis, an individual in the survival time data might provide only partial information because of censoring. For this hypothesis, the SAMPLESIZE statement computes the required numbers of events. With additional accrual information in a survival analysis, the sample sizes can also be computed.

Only one SAMPLESIZE statement can be specified. For each specified group sequential design, the SAMPLESIZE statement computes the required sample sizes or numbers of events. The SAMPLESIZE statement is not required if the SEQDESIGN procedure is used only to compare features among different designs. Table 77.4 lists the options available in the SAMPLESIZE statement.

Table 77.4 SAMPLESIZE Statement Options

Option	Description
Fixed-Sample Models	
INPUTNOBS	specifies sample size for fixed-sample design
INPUTNEVENTS	specifies number of events for fixed-sample design
One-Sample Models	
ONESAMPLEMEAN	specifies one-sample Z test for mean
ONESAMPLEFREQ	specifies one-sample test for binomial proportion
Two-Sample Models	
TWOSAMPLEMEAN	specifies two-sample Z test for mean difference
TWOSAMPLEFREQ	specifies two-sample test for binomial proportions
TWOSAMPLESURVIVAL	specifies log-rank test for two survival distributions
Regression Models	
REG	specifies test for a regression parameter
LOGISTIC	specifies test for a logistic regression parameter
PHREG	specifies test for a proportional hazards regression parameter

The MODEL= option specifies the input sample size or number of events from a fixed-sample study, or it specifies a statistical model to compute the required sample size. The MODEL=INPUTNOBS option specifies the input sample size from a fixed-sample study of nonsurvival data, and the MODEL=INPUTNEVENTS option specifies the number of events from a fixed-sample study of survival data. The remaining MODEL= options specify the statistical models used to compute the required sample size. The default is MODEL=TWOSAMPLEMEAN, the two-sample Z test for the mean difference.

With the MODEL=INPUTNOBS or MODEL=INPUTNEVENTS option, the required sample size or number of events for the group sequential trial is computed by multiplying the input sample size or number of events by the ratio between the design information level and its corresponding fixed-sample information level. This ratio can be obtained by dividing the Max Information (Percent

Fixed-Sample) in the "Design Information" table by 100. See the section "Design Information" on page 5877 for a description of the "Design Information" table.

Fixed-Sample Models

The following two options compute the required sample size or number of events for a group sequential trial by using the sample size or number of events for the fixed-sample design.

MODEL=INPUTNOBS < (*options*) >
specifies the sample size information for a fixed-sample design. The available options are as follows:

- N= n
- SAMPLE= ONE | TWO
- WEIGHT= w_a < w_b >
- MATCHNOBS= YES | NO

The required N=n option specifies the sample size n for the fixed-sample design. The SAMPLE=ONE option specifies a one-sample test, and the SAMPLE=TWO option specifies a two-sample test. The default is SAMPLE=ONE.

With a two-sample test, the WEIGHT= option specifies the sample size allocation weights for the two groups. If w_b is not specified, $w_b = 1$ is used. The default is WEIGHT=1, equal allocation for the two groups. The derived fractional sample sizes are rounded up to integers, and the MATCHNOBS=YES option requests these integer sample sizes to match the sample size allocation.

See the section "Input Sample Size for Fixed-Sample Design" on page 5856 for a detailed description of the input sample size for the fixed-sample design in sample size computation.

MODEL=INPUTNEVENTS < (*options*) >
specifies the number of events D for a fixed-sample survival test. The available options are as follows:

- D= d
- SAMPLE= ONE | TWO

The required D=d option specifies the fixed-sample number of events d. The SAMPLE=ONE option specifies a one-sample test, and the SAMPLE=TWO option specifies a two-sample test. The default is SAMPLE=ONE.

In order to derive the sample size, addition options are needed. The available options for the sample size computation are as follows:

- HAZARD= h_a < h_b >
- MEDSURVTIME= t_a < t_b >
- WEIGHT= w_a < w_b >
- ACCRATE= r_a
- ACCTIME= T_a

- FOLTIME= T_f
- TOTALTIME= T

The hazard rates are needed for the sample size computation. For a one-sample test, the HAZARD= h_a option specifies the hazard rate h_a explicitly, and the MEDSURVTIME=t_a option specifies the hazard rate implicitly through the median survival time t_a. Similarly, for a two-sample test, the HAZARD= h_a h_b option specifies the hazard rates h_a and h_b for groups A and B explicitly, and the MEDSURVTIME=t_a t_b option specifies hazard rates for groups A and B implicitly through the median survival times t_a and t_b. Also, for a two-sample test, $h_b = h_a$ if h_b is not specified and $t_b = t_a$ if t_b is not specified.

With a two-sample test, the WEIGHT= option specifies the sample size allocation weights for the two groups. If w_b is not specified, $w_b = 1$ is used. The default is WEIGHT=1.

Assuming that the hazard rates are constant and the individual accrual is uniform in the accrual time T_a with a constant accrual rate r_a, the sample size and study time can be derived.

The ACCRATE= option specifies the constant accrual rate r_a, and the ACCTIME= and FOLTIME= options specify the accrual time T_a and follow-up time T_f, respectively. The TOTALTIME= option specifies the total study time, $T = T_a + T_f$.

If the ACCRATE= option is specified, then one of the ACCTIME=, FOLTIME, and TOTALTIME options is required for the sample size computation. Otherwise, two of the ACCTIME=, FOLTIME, and TOTALTIME options are required to compute the accrual rate and sample size.

See the section "Input Number of Events for Fixed-Sample Design" on page 5856 for a detailed description of the input number of events for the fixed-sample design in sample size computation.

One-Sample Models

The following two options compute the required sample size for a one-sample group sequential test.

MODEL=ONESAMPLEMEAN < (*options*) >

specifies the one-sample Z test for mean. The available options are as follows:

- MEAN= μ_1
- STDDEV= σ

The MEAN= option specifies the alternative reference μ_1 and is required if the alternative reference is not specified or derived in the procedure. If the MEAN=option is not specified, the specified or derived alternative reference is used.

The STDDEV= option specifies the standard deviation σ. The default is STDDEV=1. See the section "Test for a Normal Mean" on page 5858 for a detailed description of the one-sample Z test for mean.

Note that the one-sample Z test for mean also includes the paired difference in two-treatment comparison (Jennison and Turnbull 2000, pp. 51–52), where μ_1 is the mean of differences within pairs under the alternative hypothesis and σ is the standard deviation for the mean of differences within pairs.

MODEL=ONESAMPLEFREQ < (*options*) >

specifies the one-sample test for binomial proportion with the null hypothesis $H_0 : \theta = 0$ and the alternative hypothesis $H_1 : \theta = \theta_1$, where $\theta = p - p_0$ and $\theta_1 = p_1 - p_0$. The available options are as follows:

- NULLPROP= p_0
- PROP= p_1
- REF= NULLPROP | PROP

The NULLPROP= and PROP= options specify the proportions under the null and alternative hypotheses, respectively. The default for the null reference is NULLPROP=0.5. The PROP= option is required if the alternative reference is not specified or derived in the procedure. If the PROP= option is not specified, the specified or derived alternative reference θ_1 is used to compute the alternative reference $p_1 = p_0 + \theta_1$.

The REF= option specifies the hypothesis under which the proportion is used in the sample size computation. The REF=NULLPROP option uses the null hypothesis, and the REF=PROP option uses the alternative hypothesis to compute the sample size. The default is REF=PROP. See the section "Test for a Binomial Proportion" on page 5859 for a detailed description of the one-sample tests for proportion.

Two-Sample Models

The following three options compute the required sample size or number of events for a two-sample group sequential trial.

MODEL=TWOSAMPLEMEAN < (*options*) >

specifies the two-sample Z test for mean difference. The available options are as follows:

- MEANDIFF= θ_1
- STDDEV= σ_a $< \sigma_b >$
- WEIGHT= w_a $< w_b >$
- MATCHNOBS= YES | NO

The MEANDIFF= option specifies the alternative reference θ_1 and is required if the alternative reference is not specified or derived in the procedure. If the MEANDIFF= option is not specified, the specified or derived alternative reference is used.

The STDDEV= option specifies the standard deviations σ_a and σ_b. If σ_b is not specified, $\sigma_b = \sigma_a$. The default is STDDEV=1.

The WEIGHT= option specifies the sample size allocation weights for the two groups. If w_b is not specified, $w_b = 1$ is used. The default is WEIGHT=1, equal sample size for the two groups. The derived fractional sample sizes are rounded up to integers, and the MATCHNOBS=YES option requests these integer sample sizes to match the sample size allocation. The default is MATCHNOBS=NO.

See the section "Test for the Difference between Two Normal Means" on page 5860 for a detailed description of the two-sample Z test for mean difference.

MODEL=TWOSAMPLEFREQ < (*options*) >

specifies the two-sample test for binomial proportions. The available options are as follows:

- NULLPROP= $p_{0a} < p_{0b} >$
- PROP= p_{1a}
- TEST= PROP | LOGOR | LOGRR
- REF= NULLPROP | PROP | AVGNULLPROP | AVGPROP
- WEIGHT= $w_a < w_b >$
- MATCHNOBS= YES | NO

The NULLPROP= option specifies proportions $p_a = p_{0a}$ and $p_b = p_{0b}$ in groups A and B, respectively, under the null hypothesis. If p_{0b} is not specified, $p_{0b} = p_{0a}$. The default is NULLPROP=0.5.

The PROP= option specifies proportion $p_a = p_{1a}$ in group A under the alternative hypothesis. The proportion p_{1b} in group B under the alternative hypothesis is given by $p_{1b} = p_{0b}$. The PROP= option is required if the alternative reference is not specified or derived in the procedure. If the PROP= option is not specified, the specified or derived alternative reference is used to compute p_{1a}, the proportion in group A under the alternative hypothesis.

The TEST= option specified the null hypothesis $H_0 : \theta = 0$ in the test. The TEST=PROP option uses the difference in proportions $\theta = (p_a - p_b) - (p_{0a} - p_{0b})$, the TEST=LOGOR option uses the log odds-ratio test $\theta = \delta - \delta_0$, where

$$\delta = \log\left(\frac{p_a(1-p_b)}{p_b(1-p_a)}\right) \qquad \delta_0 = \log\left(\frac{p_{0a}(1-p_{0b})}{p_{0b}(1-p_{0a})}\right)$$

and the TEST=LOGRR option uses the log relative risk test with $\theta = \delta - \delta_0$, where

$$\delta = \log\left(\frac{p_a}{p_b}\right) \qquad \delta_0 = \log\left(\frac{p_{0a}}{p_{0b}}\right)$$

The default is TEST=LOGOR.

The REF= option specifies the hypothesis under which the proportions are used in the sample size computation. The REF=NULLPROP option uses the null proportions p_{0a} and p_{0b}, the REF=PROP option uses the alternative proportions p_{1a} and p_{1b}, the REF=AVGNULLPROP option uses the average null proportion, and the REF=AVGPROP option uses the average alternative proportion. The default is REF=PROP.

The WEIGHT= option specifies the sample size allocation weights for the two groups. If w_b is not specified, $w_b = 1$ is used. The default is WEIGHT=1, equal sample size for the two groups. The derived fractional sample sizes are also rounded up to integers, and the MATCHNOBS=YES option requests that these integer sample sizes match the sample size allocation.

See the section "Test for the Difference between Two Binomial Proportions" on page 5862, the section "Test for Two Binomial Proportions with a Log Odds Ratio Statistic" on page 5864, and the section "Test for Two Binomial Proportions with a Log Relative Risk Statistic" on page 5865 for a detailed description of the two-sample tests for proportions.

MODEL=TWOSAMPLESURVIVAL < (*options*) >

MODEL=TWOSAMPLESURV < (*options*) >

specifies the log-rank test for two survival distributions with the null hypothesis $H_0 : \theta = \delta - \delta_0 = 0$, where the parameter $\delta = -\log(h_a/h_b)$, δ_0 is the value of δ under the null hypothesis and the values h_a and h_b are the hazard rates for groups A and B, respectively.

The available options for the number of events are as follows:

- NULLHAZARD= $h_{0a} < h_{0b} >$
- NULLMEDSURVTIME= $t_{0a} < t_{0b} >$
- HAZARD= h_{1a}
- MEDSURVTIME= t_{1a}
- HAZARDRATIO= λ_1

The NULLHAZARD= option specifies hazard rates $h_a = h_{0a}$ and $h_b = h_{0b}$ for groups A and B, respectively, under the null hypothesis. If h_{0b} is not specified, $h_{0b} = h_{0a}$. The NULLMEDSURVTIME= option specifies the median survival times $t_a = t_{0a}$ and $t_b = t_{0b}$ under the null hypothesis. If t_{0b} is not specified, $t_{0b} = t_{0a}$. If both NULLHAZARD= and NULLMEDSURVTIME= option are not specified, NULLHAZARD=0.06931, which corresponds to NULLMEDSURVTIME=10, is used.

The hazard rate for group B under the alternative hypothesis $h_{1b} = h_{0b}$, as the hazard rate under the null hypothesis. The HAZARD=, MEDSURVTIME=, and HAZARDRATIO= options specify the group A hazard rate h_{1a}, the group A median survival time t_{1a}, and the hazard ratio $\lambda_1 = h_{1a}/h_{1b}$, respectively, under the alternative hypothesis. The HAZARD=, MEDSURVTIME=, or HAZARDRATIO= option is required if the alternative reference is not specified or derived in the procedure. If these three options are not specified, the specified or derived alternative reference θ_1 is used to compute h_{1a} from the equation:

$$\theta_1 = -\log(\frac{h_{1a}}{h_{1b}}) - (-\log(\frac{h_{0a}}{h_{0b}})) = -\log(\frac{h_{1a}}{h_{0a}})$$

In order to derive the sample size, additional options are needed. The available options for the sample size computation are as follows:

- REF= NULLHAZARD | HAZARD
- WEIGHT= $w_a < w_b >$
- ACCRATE= r_a
- ACCTIME= T_a
- FOLTIME= T_f
- TOTALTIME= T

The REF= option specifies the hypothesis under which the hazard is used in the sample size computation. The REF=NULLHAZARD option uses the null hypothesis, and the REF=HAZARD option uses the alternative hypothesis. The default is REF=HAZARD.

The WEIGHT= option specifies the sample size allocation weights for the two groups. If w_b is not specified, $w_b = 1$ is used. The default is WEIGHT=1, equal sample size for the two groups.

With the available maximum information, the number of events can be derived for the specified hypothesis. Assuming that the hazard rates are constant and the individual accrual is uniform in the accrual time T_a with a constant accrual rate r_a, the sample size and study time can be derived.

The ACCRATE= option specifies the constant accrual rate r_a, and the ACCTIME= and FOLTIME= options specify the accrual time T_a and follow-up time T_f, respectively. The TOTALTIME= option specifies the total study time, $T = T_a + T_f$.

If the ACCRATE= option is specified, then one of the ACCTIME=, FOLTIME=, and TOTALTIME= options is required for the sample size computation. Otherwise, two of the ACCTIME=, FOLTIME=, and TOTALTIME= options are required to compute the accrual rate and sample size.

See the section "Test for Two Survival Distributions with a Log-Rank Test" on page 5866 for a detailed description of the two-sample log-rank test for survival data.

Regression Models

The following three options compute the required sample size or number of events for group sequential tests on a regression parameter.

MODEL=REG < (*options*) >

specifies the Z test for a normal regression parameter. The available options are as follows:

- BETA= β_1
- VARIANCE= σ_y^2
- XVARIANCE= σ_x^2
- XRSQUARE= r_x^2

The BETA= option specifies the alternative reference β_1 and is required if the alternative reference is not specified or derived in the procedure. If the BETA= option is not specified, $\beta_1 = \theta_1$, the specified or derived alternative reference.

The VARIANCE= and XVARIANCE= options specify the variances for the response variable Y and covariate X, respectively. The defaults are VARIANCE=1 and XVARIANCE=1. For a model with more than one covariate, the XRSQUARE= option can be used to derive the variance of X after adjusting for other covariates. The default is XRSQUARE=0.

See the section "Test for a Parameter in the Regression Model" on page 5869 for a detailed description of the Z test for the regression parameter.

MODEL=LOGISTIC < (*options*) >

specifies the Z test for a logistic regression parameter. The available options are as follows:

- BETA= β_1
- PROP= p
- XVARIANCE= σ_x^2
- XRSQUARE= r_x^2

The BETA= option specifies the alternative reference β_1 and is required if the alternative reference is not specified or derived in the procedure. If the BETA= option is not specified, $\beta_1 = \theta_1$, the specified or derived alternative reference.

The PROP= option specifies the proportion of the binary response variable Y. The default is PROP=0.5. The XVARIANCE= option specifies the variance of the covariate X. The default is XVARIANCE=1. For a model with more than one covariate, the XRSQUARE= option can be used to derive the variance of X after adjusting for other covariates. The default is XRSQUARE=0.

See the section "Test for a Parameter in the Logistic Regression Model" on page 5870 for a detailed description of the Z test for the logistic regression parameter.

MODEL=PHREG < (*options*) >

specifies the Z test for a proportional hazards regression parameter. The available options for the number of events are as follows:

- BETA= β_1
- XVARIANCE= σ_x^2
- XRSQUARE= r_x^2

The BETA= option specifies the alternative reference β_1 and is required if the alternative reference is not specified or derived in the procedure. If the BETA= option is not specified, $\beta_1 = \theta_1$, the specified or derived alternative reference.

The XVARIANCE= option specifies the variance of the covariate X. The default is XVARIANCE=1. For a model with more than one covariate, the XRSQUARE= option can be used to derive the variance of X after adjusting for other covariates. The default is XRSQUARE=0.

In order to derive the sample size, additional options are needed. The available options for the sample size computation are as follows:

- HAZARD= h_a
- MEDSURVTIME= t_a
- ACCRATE= r_a
- ACCTIME= T_a
- FOLTIME= T_f
- TOTALTIME= T

The hazard rate is required for the sample size computation. The HAZARD= h_a option specifies the hazard rate h_a explicitly, and the MEDSURVTIME=t_a option specifies the hazard rate implicitly through the median survival time t_a.

Assuming that the hazard rates are constant and the individual accrual is uniform in the accrual time T_a with a constant accrual rate r_a, the sample size and study time can be derived.

The ACCRATE= option specifies the constant accrual rate r_a, the ACCTIME= option specifies the accrual time T_a, and the FOLTIME= option specifies the follow-up time T_f. The TOTALTIME= option specifies the total study time, $T = T_a + T_f$.

If the ACCRATE= option is specified, then one of the ACCTIME=, FOLTIME=, and TOTALTIME= options is required for the sample size computation. Otherwise, two of the ACCTIME=, FOLTIME=, and TOTALTIME= options are required to compute the accrual rate and sample size.

See the section "Test for a Parameter in the Proportional Hazards Regression Model" on page 5871 for a detailed description of the Z test for the proportional hazards regression parameter.

Details: SEQDESIGN Procedure

Fixed-Sample Clinical Trials

A clinical trial is a research study in consenting human beings to answer specific health questions. One type of trial is a treatment trial, which tests the effectiveness of an experimental treatment. An example is a planned experiment designed to assess the efficacy of a treatment in humans by comparing the outcomes in a group of patients who receive the test treatment with the outcomes in a comparable group of patients who receive a placebo control treatment, where patients in both groups are enrolled, treated, and followed over the same time period.

A clinical trial is conducted according to a plan called a protocol. The protocol provides detailed description of the study. For a fixed-sample trial, the study protocol contains detailed information such as the null hypothesis, the one-sided or two-sided test, and the Type I and II error probability levels. It also includes the test statistic and its associated critical values in the hypothesis testing.

Generally, the efficacy of a new treatment is demonstrated by testing a hypothesis $H_0 : \theta = 0$ in a clinical trial, where θ is the parameter of interest. For example, to test whether a population mean μ is greater than a specified value μ_0, $\theta = \mu - \mu_0$ can be used with an alternative $\theta > 0$.

A one-sided test is a test of the hypothesis with either an upper (greater) or a lower (lesser) alternative, and a two-sided test is a test of the hypothesis with a two-sided alternative. The drug industry often prefers to use a one-sided test to demonstrate clinical superiority based on the argument that a study should not be run if the test drug would be worse (Chow, Shao, and Wang 2003, p. 28). But in practice, two-sided tests are commonly performed in drug development (Senn 1997, p. 161). For a fixed Type I error probability α, the sample sizes required by one-sided and two-sided tests are

different. Refer to Senn (1997, pp. 161–167) for a detailed description of issues involving one-sided and two-sided tests.

For independent and identically distributed observations $y_1, y_2, \ldots, y_n$ of a random variable, the likelihood function for θ is

$$L(\theta) = \prod_{j=1}^{n} L_i(\theta)$$

where θ is the population parameter and $L_i(\theta)$ is the probability or probability density of y_i. Using the likelihood function, two statistics can be derived that are useful for inference: the maximum likelihood estimator and the score statistic.

Maximum Likelihood Estimator

The maximum likelihood estimate (MLE) of θ is the value $\hat{\theta}$ that maximizes the likelihood function for θ. Under mild regularity conditions, $\hat{\theta}$ is an asymptotically unbiased estimate of θ with variance $1/E_\theta(I(\theta))$, where $I(\theta)$ is the Fisher information

$$I(\theta) = -\frac{\partial^2 \log(L(\theta))}{\partial \theta^2}$$

and $E_\theta(I(\theta))$ is the expected Fisher information (Diggle et al. 2002, p. 340)

$$E_\theta(I(\theta)) = -E_\theta\left(\frac{\partial^2 \log(L(\theta))}{\partial \theta^2}\right)$$

The score function for θ is defined as

$$S(\theta) = \frac{\partial \log(L(\theta))}{\partial \theta}$$

and usually, the MLE can be derived by solving the likelihood equation $S(\theta) = 0$. Asymptotically, the MLE is normally distributed (Lindgren 1976, p. 272):

$$\hat{\theta} \sim N\left(\theta, \frac{1}{E_\theta(I(\theta))}\right)$$

If the Fisher information $I(\theta)$ does not depend on θ, then $I(\theta)$ is known. Otherwise, either the expected information evaluated at the MLE $\hat{\theta}$ ($E_{\theta=\hat{\theta}}(I(\theta))$) or the observed information $I(\hat{\theta})$ can be used for the Fisher information (Cox and Hinkley, 1974, p. 302; Efron and Hinkley, 1978, p. 458), where the observed Fisher information

$$I(\hat{\theta}) = -\left(\frac{\partial^2 \log(L(\theta))}{\partial \theta^2} \mid \theta = \hat{\theta}\right)$$

If the Fisher information $I(\theta)$ does depend on θ, the observed Fisher information is recommended for the variance of the maximum likelihood estimator (Efron and Hinkley, 1978, p. 457).

Thus, asymptotically, for large n,

$$\hat{\theta} \sim N\left(\theta, \frac{1}{I}\right)$$

where I is the information, either the expected Fisher information $E_{\theta=0}(I(\theta))$ or the observed Fisher information $I\hat{\theta})$.

So to test $H_0 : \theta = 0$ versus $H_1 : \theta \neq 0$, you can use the standardized Z test statistic

$$Z = \frac{\hat{\theta}}{\sqrt{\text{Var}(\hat{\theta})}} = \hat{\theta}\sqrt{I} \sim N\,(0,\ 1)$$

and the two-sided p-value is given by

$$\text{Prob}(|Z| > |z_0|) = 1 - 2\Phi(|z_0|)$$

where Φ is the cumulative standard normal distribution function and z_0 is the observed Z statistic.

If the BOUNDARYSCALE=SCORE is specified in the SEQDESIGN procedure, the boundary values for the test statistic are displayed in the score statistic scale. With the standardized Z statistic, the score statistic $S = Z\sqrt{I} = \hat{\theta}I$ and

$$S \sim N\,(0,\ I)$$

Score Statistic

The score statistic is based on the score function for θ,

$$S(\theta) = \frac{\partial \log(L(\theta))}{\partial\theta}$$

Under the null hypothesis $H_0 : \theta = 0$, the score statistic $S(0)$ is the first derivative of the log likelihood evaluated at the null reference 0:

$$S(0) = \frac{\partial \log(L(\theta))}{\partial\theta} \,|\, \theta = 0$$

Under regularity conditions, $S(0)$ is asymptotically normally distributed with mean zero and variance $E_{\theta=0}(I(\theta))$, the expected Fisher information evaluated at the null hypothesis $\theta = 0$ (Kalbfleisch and Prentice, 1980, p. 45), where $I(\theta)$ is the Fisher information

$$I(\theta) = -E\left(\frac{\partial^2 \log(L(\theta))}{\partial\theta^2}\right)$$

That is, for large n,

$$S(0) \sim N\,(0,\ E_{\theta=0}(I(\theta)))$$

Asymptotically, the variance of the score statistic $S(0)$, $E_{\theta=0}(I(\theta))$, can also be replaced by the expected Fisher information evaluated at the MLE $\theta = \hat{\theta}$ ($E_{\theta=\hat{\theta}}(I(\theta))$), the observed Fisher information evaluated at the null hypothesis $\theta = 0$ ($I(0)$), or the observed Fisher information evaluated at the MLE $\theta = \hat{\theta}$ ($I(\hat{\theta})$) (Kalbfleisch and Prentice, 1980, p. 46), where

$$I(0) = -\left(\frac{\partial^2 \log(L(\theta))}{\partial \theta^2} \mid \theta = 0\right)$$

$$I(\hat{\theta}) = -\left(\frac{\partial^2 \log(L(\theta))}{\partial \theta^2} \mid \theta = \hat{\theta}\right)$$

Thus, asymptotically, for large n,

$$S(0) \sim N\,(0,\ I)$$

where I is the information, either an expected Fisher information ($E_{\theta=0}(I(\theta))$ or $E_{\theta=\hat{\theta}}(I(\theta))$) or a observed Fisher information ($I(0)$ or $I(\hat{\theta})$).

So to test $H_0 : \theta = 0$ versus $H_1 : \theta \neq 0$, you can use the standardized Z test statistic

$$Z = \frac{S(0)}{\sqrt{I}}$$

If the BOUNDARYSCALE=MLE is specified in the SEQDESIGN procedure, the boundary values for the test statistic are displayed in the MLE scale. With the standardized Z statistic, the MLE statistic $\hat{\theta} = Z/\sqrt{I} = U(0)/I$ and

$$\hat{\theta} \sim N\left(0,\ \frac{1}{I}\right)$$

One-Sample Test for Mean

The following one-sample test for mean is used to demonstrate fixed-sample clinical trials in the section "One-Sided Fixed-Sample Tests in Clinical Trials" on page 5821 and the section "Two-Sided Fixed-Sample Tests in Clinical Trials" on page 5824.

Suppose $y_1, y_2, \ldots, y_n$ are n observations of a response variable Y from a normal distribution

$$y_i \sim N\left(\theta,\ \sigma^2\right)$$

where θ is the unknown mean and σ^2 is the known variance.

Then the log likelihood function for θ is

$$\log(L(\theta)) = \sum_{j=1}^{n} -\frac{1}{2}\frac{(y_j - \theta)^2}{\sigma^2} + c$$

where c is a constant. The first derivative is

$$\frac{\partial \log(L(\theta))}{\partial \theta} = \frac{1}{\sigma^2}\sum_{j=1}^{n}(y_j - \theta) = \frac{n}{\sigma^2}(\overline{y} - \theta)$$

where $\overline{y}$ is the sample mean.

Setting the first derivative to zero, the MLE of θ is $\hat{\theta} = \overline{y}$, the sample mean. The variance for $\hat{\theta}$ can be derived from the Fisher information

$$I(\theta) = -\frac{\partial^2 \log(L(\theta))}{\partial \theta^2} = \frac{n}{\sigma^2}$$

Since the Fisher information $I_0 = I(\theta)$ does not depend on θ in this case, $1/I_0$ is used as the variance for $\hat{\theta}$. Thus the sample mean $\overline{y}$ has a normal distribution with mean θ and variance σ^2/n:

$$\hat{\theta} = \overline{y} \sim N\left(\theta, \frac{1}{I_0}\right) = N\left(\theta, \frac{\sigma^2}{n}\right)$$

Under the null hypothesis $H_0 : \theta = 0$, the score statistic

$$S(0) = \frac{\partial \log(L(\theta))}{\partial \theta} | \theta = 0 = \frac{n}{\sigma^2}\overline{y}$$

has a mean zero and variance

$$I(\theta) = -\frac{\partial^2 \log(L(\theta))}{\partial \theta^2} = \frac{n}{\sigma^2}$$

With the MLE $\hat{\theta}$, the corresponding standardized statistic is computed as $Z = \hat{\theta}\sqrt{I_0} = \overline{y}/(\sigma/\sqrt{n})$, which has a normal distribution with variance 1:

$$Z \sim N\left(\theta\sqrt{I_0}, 1\right) = N\left(\frac{\theta}{\sigma/\sqrt{n}}, 1\right)$$

Also, the corresponding score statistic is computed as $S = \hat{\theta} I_0 = n\overline{y}/\sigma^2$ and

$$S \sim N\left(\theta I_0, I_0\right) = N\left(\frac{n\theta}{\sigma^2}, \frac{n}{\sigma^2}\right)$$

which is identical to $S(0)$ computed under the null hypothesis $H_0 : \theta = 0$.

Note that if the variable Y does not have a normal distribution, then it is assumed that the sample size n is large such that the sample mean has an approximately normal distribution.

One-Sided Fixed-Sample Tests in Clinical Trials

A one-sided test has either an upper (greater) or a lower (lesser) alternative. This section describes one-sided tests with upper alternatives only. Corresponding results for one-sided tests with lower alternatives can be derived similarly.

For a one-sided test of $H_0 : \delta \leq \delta_0$ with an upper alternative $H_1 : \delta > \delta_0$, an equivalent null hypothesis is $H_0 : \theta \leq 0$ with an upper alternative $H_1 : \theta > 0$, where $\theta = \delta - \delta_0$. A fixed-sample test rejects H_0 if the standardized test statistic $Z_0 = \hat{\theta}\sqrt{I_0} \geq C_\alpha$, where $\hat{\theta}$ is the sample estimate of θ and $C_\alpha = \Phi^{-1}(1-\alpha)$ is the critical value.

The p-value of the test is given by $1 - \Phi(Z_0)$, and the hypothesis H_0 is rejected if the p-value is less than α. An upper $(1-\alpha)$ confidence interval has the lower limit

$$\theta_l = \hat{\theta} - \frac{\Phi^{-1}(1-\alpha)}{\sqrt{I_0}} = \frac{Z_0 - \Phi^{-1}(1-\alpha)}{\sqrt{I_0}}$$

The hypothesis H_0 is rejected if the confidence interval for the parameter θ does not contain zero—that is, if the lower limit θ_l is greater than 0.

With an alternative reference $\theta = \theta_1$, $\theta_1 > 0$, a Type II error probability is defined as

$$\beta = P_{\theta=\theta_1}(Z_0 < C_\alpha)$$

which is equivalent to

$$\beta = P_{\theta=\theta_1}\left(Z_0 - \theta_1\sqrt{I_0} \; < \; C_\alpha - \theta_1\sqrt{I_0}\right) = \Phi\left(C_\alpha - \theta_1\sqrt{I_0}\right)$$

Thus, $\Phi^{-1}(\beta) = C_\alpha - \theta_1\sqrt{I_0}$. Then, with $C_\alpha = \Phi^{-1}(1-\alpha)$,

$$\theta_1\sqrt{I_0} = \Phi^{-1}(1-\alpha) + \Phi^{-1}(1-\beta)$$

The drift parameter $\theta_1\sqrt{I_0}$ can be computed for specified α and β and the maximum information is given by

$$I_0 = \left(\frac{\Phi^{-1}(1-\alpha) + \Phi^{-1}(1-\beta)}{\theta_1}\right)^2$$

If the maximum information is available, then the required sample size can be derived. For example, in a one-sample test for the mean with a specific standard deviation σ, the sample size n required for the test is

$$n = \sigma^2 I_0 = \sigma^2\left(\frac{\Phi^{-1}(1-\alpha) + \Phi^{-1}(1-\beta)}{\theta_1}\right)^2$$

On the other hand, if the alternative reference θ_1, standard deviation σ, and sample size n are all specified, then α can be derived for a given β and, similarly, β can be derived for a given α.

With an alternative reference $\theta = \theta_1$, $\theta_1 > 0$, the power $1 - \beta$ is the probability of correctly rejecting the null hypothesis H_0 at θ_1:

$$1 - \beta = 1 - P_{\theta=\theta_1}(Z_0 < C_\alpha) = \Phi\left(\theta_1\sqrt{I_0} - C_\alpha\right)$$

Superiority Trials

A superiority trial that tests the response to a new drug is clinically superior to a comparative placebo control or active control therapy. If a positive value indicates a beneficial effect, a test for superiority has

$$H_0 : \theta \leq 0 \qquad H_1 : \theta > 0$$

where H_0 is the hypothesis of nonsuperiority and H_1 is the alternative hypothesis of superiority.

The superiority test rejects the hypothesis H_0 and declares superiority if the standardized statistic $Z_0 = \hat{\theta}\sqrt{I_0} \geq C_\alpha$, where the critical value $C_\alpha = \Phi^{-1}(1-\alpha)$.

For example, if θ is the response difference between the treatment and placebo control groups, then a superiority trial can be

$$H_0 : \theta \leq 0 \qquad H_1 : \theta = 6$$

with a Type I error probability level $\alpha = 0.025$ and a power $1 - \beta = 0.90$ at $\theta_1 = 6$.

Noninferiority Trials

A noninferiority trial does not compare the response to a new treatment with the response to a placebo. Instead, it demonstrates the effectiveness of a new treatment compared with that of a nonexisting placebo by showing that the response of a new treatment is not clinically inferior to the response of a standard therapy with an established effect. That is, this type of trial attempts to demonstrate that the new treatment effect is not worse than the standard therapy effect by an acceptable margin. These trials are often performed when there is an existing effective therapy for a serious disease, and therefore a placebo control group cannot be ethically included.

It can be difficult to specify an appropriate noninferiority margin. One practice is to choose with reference to the effect of the active control in historical placebo-controlled trials (Snapinn 2000, p. 20). With this practice, there is some basis to imply that the new treatment is better than the placebo for a positive noninferiority trial.

If a positive value indicates a beneficial effect, a test for noninferiority has a null hypothesis $\delta \leq -\delta_0$ and an alternative hypothesis $\delta = \delta_1 > -\delta_0$, where $\delta_0 > 0$ is the specified noninferiority margin.

An equivalent test has

$$H_0 : \theta \leq 0 \qquad H_1 : \theta = \theta_1 > 0$$

where the parameter $\theta = \delta + \delta_0$, H_0 is the null hypothesis of inferiority, and H_1 is the alternative hypothesis of noninferiority,

The noninferiority test rejects the hypothesis H_0 and declares noninferiority if the standardized statistic $Z_0 = \hat{\theta}\sqrt{I_0} = (\hat{\delta} + \delta_0)\sqrt{I_0} \geq C_\alpha$, where the critical value $C_\alpha = \Phi^{-1}(1-\alpha)$.

For example, if δ is the response difference between the treatment and active control groups and $\delta_0 = 2$ is the noninferiority margin, then a noninferiority trial with a power $1-\beta = 0.90$ at $\delta_1 = 1$ might be

$$H_0 : \theta \leq 0 \qquad H_1 : \theta = 3$$

where $\theta = \delta + \delta_0 = \delta + 2$.

Two-Sided Fixed-Sample Tests in Clinical Trials

A two-sided test is a test of a hypothesis with a two-sided alternative. Two-sided tests include simple symmetric tests and more complicated asymmetric tests that might have distinct lower and upper alternative references.

Symmetric Two-Sided Tests for Equality

For a symmetric two-sided test with the null hypothesis $\delta = \delta_0$ against the alternative $\delta \neq \delta_0$, an equivalent null hypothesis is $H_0 : \theta = 0$ with a two-sided alternative $H_1 : \theta \neq 0$, where $\theta = \delta - \delta_0$. A fixed-sample test rejects H_0 if $|\hat{\theta}\sqrt{I_0}| \geq C_{\alpha/2}$, where $\hat{\theta}$ is a sample estimate of θ and $C_{\alpha/2} = \Phi^{-1}(1-\alpha/2)$ is the critical value.

A common two-sided test is the test for the response difference between a treatment group and a control group. The null and alternative hypotheses are $H_0 : \theta = 0$ and $H_1 : \theta \neq 0$, respectively, where θ is the response difference between the two groups. If a greater value indicates a beneficial effect, then there are three possible results:

- The test rejects the hypothesis H_0 of equality and indicates that the treatment is significantly better if the standardized statistic $Z_0 = \hat{\theta}\sqrt{I_0} \geq C_{\alpha/2}$.
- The test rejects the hypothesis H_0 and indicates the treatment is significantly worse if the standardized statistic $Z_0 = \hat{\theta}\sqrt{I_0} \leq -C_{\alpha/2}$.
- The test indicates no significant difference between the two responses if $-C_{\alpha/2} < \hat{\theta}\sqrt{I_0} < C_{\alpha/2}$.

The p-value of the test is $2(1-\Phi(Z_0))$ if $Z_0 > 0$ and $2\Phi(Z_0)$ if $Z_0 \leq 0$. The hypothesis H_0 is rejected if the p-value of the test is less than α—that is, if $1-\Phi(Z_0) < \alpha/2$ or $\Phi(Z_0) < \alpha/2$. A symmetric $(1-\alpha)$ confidence interval for θ has lower and upper limits

$$\left(\hat{\theta} - \frac{C_{\alpha/2}}{\sqrt{I_0}}\,,\ \hat{\theta} + \frac{C_{\alpha/2}}{\sqrt{I_0}}\right)$$

which is

$$\left(\frac{1}{\sqrt{I_0}}\left(Z_0 - C_{\alpha/2}\right)\,,\ \frac{1}{\sqrt{I_0}}\left(Z_0 + C_{\alpha/2}\right)\right)$$

The hypothesis H_0 is rejected if the confidence interval for the parameter θ does not contain zero. That is, the lower limit is greater than zero or the upper limit is less than zero.

With an alternative reference $\theta = \theta_1 > 0$, a Type II error probability is defined as

$$\beta = P_{\theta=\theta_1}(-C_{\alpha/2} \ < \ Z_0 \ < \ C_{\alpha/2})$$

which is

$$\beta = P_{\theta=\theta_1}\left((-C_{\alpha/2} - \theta_1\sqrt{I_0}) \ < \ (Z_0 - \theta_1\sqrt{I_0}) \ < \ (C_{\alpha/2} - \theta_1\sqrt{I_0})\right)$$

Thus

$$\beta = \Phi\left(C_{\alpha/2} - \theta_1\sqrt{I_0}\right) - \Phi\left(-C_{\alpha/2} - \theta_1\sqrt{I_0}\right)$$

The resulting power $1 - \beta$ is the probability of correctly rejecting the null hypothesis, which includes the probability for the lower alternative and the probability for the upper alternative. The SEQDESIGN procedure uses only the probability of correctly rejecting the null hypothesis for the correct alternative in the power computation.

Thus, under the upper alternative hypothesis, the power in the SEQDESIGN procedure is computed as the probability of rejecting the null hypothesis for the upper alternative, $1. - \Phi\left(C_{\alpha/2} - \theta_1\sqrt{I_0}\right) = \Phi\left(\theta_1\sqrt{I_0} - C_{\alpha/2}\right)$, and a very small probability of rejecting the null hypothesis for the lower alternative, $\Phi\left(-C_{\alpha/2} - \theta_1\sqrt{I_0}\right)$, is ignored. This power computation is more rational than the power based on the probability of correctly rejecting the null hypothesis (Whitehead 1997, p. 75).

That is,

$$\beta = P_{\theta=\theta_1}\left((Z_0 - \theta_1\sqrt{I_0}) \ < \ (C_{\alpha/2} - \theta_1\sqrt{I_0})\right) = \Phi\left(C_{\alpha/2} - \theta_1\sqrt{I_0}\right)$$

Then with $\Phi^{-1}(\beta) = C_{\alpha/2} - \theta_1\sqrt{I_0}$,

$$\theta_1\sqrt{I_0} = C_{\alpha/2} - \Phi^{-1}(\beta) = \Phi^{-1}(1-\alpha/2) + \Phi^{-1}(1-\beta)$$

The drift parameter $\theta_1\sqrt{I_0}$ can be derived for specified α and β, and the maximum information is given by

$$I_0 = \left(\frac{\Phi^{-1}(1-\alpha/2) + \Phi^{-1}(1-\beta)}{\theta_1}\right)^2$$

If the maximum information is available, then the required sample size can be derived. For example, in a one-sample test for mean, if the standard deviation σ is known, the sample size n required for the test is

$$n = \sigma^2 I_0 = \sigma^2\left(\frac{\Phi^{-1}(1-\alpha/2) + \Phi^{-1}(1-\beta)}{\theta_1}\right)^2$$

On the other hand, if the alternative reference θ_1, standard deviation σ, and sample size n are all known, then α can be derived with a given β and, similarly, β can be derived with a given α.

Generalized Two-Sided Tests for Equality

For a generalized two-sided test with the null hypothesis $\delta = \delta_0$ against the alternative $\delta \neq \delta_0$, an equivalent null hypothesis is $H_0 : \theta \leq 0$ with a two-sided alternative $H_1 : \theta \neq 0$, where $\theta = \delta - \delta_0$. A fixed-sample test rejects H_0 if the standardized statistic $Z_0 = \hat{\theta}\sqrt{I_0} < -C_{\alpha_l}$ or $Z_0 = \hat{\theta}\sqrt{I_0} > C_{\alpha_u}$, where the critical values $C_{\alpha_l} = \Phi^{-1}(1-\alpha_l)$ and $C_{\alpha_u} = \Phi^{-1}(1-\alpha_u)$.

With the lower alternative reference $\theta_{1l} < 0$, a lower Type II error probability is defined as

$$\beta_l = P_{\theta=\theta_{1l}}\left(-C_{\alpha_l} \leq Z_{0l}\sqrt{I_0}\right) = P_{\theta=\theta_{1l}}\left(-C_{\alpha_l} - \theta_{1l}\sqrt{I_0} \leq Z_{0l}\sqrt{I_0} - \theta_{1l}\sqrt{I_0}\right)$$

This implies

$$\beta_l = 1 - \Phi(-C_{\alpha_l} - \theta_{1l}\sqrt{I_0})$$

and the power is the probability of correctly rejecting the null hypothesis for the lower alternative,

$$1 - \beta_l = \Phi(-C_{\alpha_l} - \theta_{1l}\sqrt{I_0})$$

The lower drift parameter is derived as

$$\theta_{1l}\sqrt{I_0} = -\left(\Phi^{-1}(1-\alpha_l) + \Phi^{-1}(1-\beta_l)\right)$$

Then, with specified α_l and β_l, if the maximum information is known, the lower alternative reference θ_{1l} can be derived. If the maximum information is unknown, then with the specified lower alternative reference θ_{1l}, the maximum information required is

$$I_{0l} = \left(\frac{\Phi^{-1}(1-\alpha_l) + \Phi^{-1}(1-\beta_l)}{-\theta_{1l}}\right)^2$$

Similarly, the upper drift parameter is derived as

$$\theta_{1u}\sqrt{I_0} = \Phi^{-1}(1-\alpha_u) + \Phi^{-1}(1-\beta_u)$$

For a given α_u, β_u, and the upper alternative reference θ_{1u}, the maximum information required is

$$I_{0u} = \left(\frac{\Phi^{-1}(1-\alpha_u) + \Phi^{-1}(1-\beta_u)}{\theta_{1u}}\right)^2$$

Thus, the maximum information required for the design is given by

$$I_0 = \max\left(I_{0l},\ I_{0u}\right)$$

Note that with the maximum information level I_0, if $I_{0l} < I_0$, then the derived power from the lower alternative is larger than the specified $1-\beta_l$. Similarly, if $I_{0u} < I_0$, then the derived power from the upper alternative is larger than the specified $1-\beta_u$.

If maximum information is available, the required sample size can be derived. For example, in a one-sample test for mean, if the standard deviation σ is known, the sample size n required for the test is $n = \sigma^2 I_0$.

On the other hand, if the alternative references, Type I error probabilities α_l and α_u, standard deviation σ, and sample size n are all specified, then the Type II error probabilities β_l and β_u and the corresponding powers can be derived.

Group Sequential Methods

A group sequential design provides interim analyses before the formal completion of a trial. The monitoring process provides possible early stopping for either positive or negative results and thus reduces the time to complete the trial. With a specified number of stages, the design creates critical values such that at each interim analysis, a hypothesis can be rejected, accepted, or continued to the next time point. At the final stage, a hypothesis is either rejected or accepted. Usually, the critical values are derived such that the specified overall Type I and Type II error probability levels are maintained in the design.

For example, to test a null hypothesis H_0 with an upper alternative in a fixed-sample design, a critical value c_α is created. The null hypothesis H_0 is rejected if the test statistic is greater than or equal to the critical value c_α. Otherwise, H_0 is accepted. But, for a group sequential design with early stopping to reject or accept the null hypothesis H_0, there are two critical values created at each interim analysis: an α critical value $c_{\alpha k}$ to reject the null hypothesis and a β critical value $c_{\beta k}$ to accept the null hypothesis. The null hypothesis H_0 is rejected if the test statistic is greater than or equal to the α critical value $c_{\alpha k}$, and H_0 is accepted if the test statistic is less than the β critical value $c_{\beta k}$. If the test statistic is between these two critical values, the process continues to the next stage. At the final stage, the two critical values are equal, and the hypothesis is either rejected or accepted.

Armitage, McPherson, and Rowe (1969) showed that repeated significance tests at a fixed level on accumulating data increase the probability of obtaining a significant result under the null hypothesis. For example, with a significance level 0.05 in a two-sided fixed-sample test, the critical value is 1.96. If this value is used in a five-stage group sequential trial with early stopping to reject the null hypothesis, then the probability of rejecting the null hypothesis at or before the fifth stage is 0.14169, much larger than the nominal value 0.05 (Armitage, McPherson, and Rowe 1969, p. 239).

Pocock (1977) applied these repeated significance tests to group sequential trials with equally spaced information levels and derives a constant critical value on the standardized normal Z scale across all stages that maintains the Type I error probability level. For example, with a significance level 0.05 in a two-sided test, the derived critical value at each stage is 2.413 on the standardized normal Z scale, larger than the fixed-sample critical value 1.96. The corresponding nominal p-value is 0.0158, which is smaller than the fixed-sample p-value 0.025 (Pocock 1977, p. 193).

O'Brien and Fleming (1979) proposed a sequential procedure that has boundary values decrease over the stages on the standardized normal Z scale to make the early stop less likely. The procedure has conservative stopping boundary values at very early stages, and boundary values at the final stage are close to the fixed-sample design. For example, with a significance level 0.05 in a two-sided test, the derived critical values at these five stages on the standardized normal Z scale are 4.562, 3.226, 2.634, 2.281, and 2.040.

Wang and Tsiatis (1987), Emerson and Fleming (1989) and Pampallona and Tsiatis (1994) generalized the Pocock and O'Brien-Fleming methods to the power family, where a power parameter is used to allow a continuous set of designs between the Pocock and O'Brien-Fleming methods.

Kittelson and Emerson (1999) extended the methods in the power family even further to the unified family, which also includes the exact triangular method. The shape and location of each of the four boundaries can be independently specified in the unified family methods.

Whitehead and Stratton (1983) and Whitehead (1997, 2001) developed triangular methods by adapting tests for continuous monitoring to discrete monitoring. With early stopping to reject or accept the null hypothesis in a one-sided test, the derived continuation region has a triangular shape for the score-scaled boundaries. Only elementary calculations are needed to derive the boundary values for Whitehead's triangular methods.

For a sequential design, you can derive the α and β error probabilities at each stage from the boundaries. On the other hand, you can derive the boundaries from specified α and β error probabilities at each stage. The error spending function approach (Lan and DeMets 1983) uses the error spending function to specify the error probabilities at each stage and then uses these probabilities to derive the boundaries. You can specify α and β explicitly or implicitly with an error spending function for the cumulative probabilities.

Refer to Jennison and Turnbull (2000, pp. 5–11) for a more detailed history of group sequential methods.

The following three types of methods are available in the SEQDESIGN procedure to derive boundaries in a sequential design:

- fixed boundary shape methods, which derive boundaries with specified boundary shapes. These include the unified family method and Haybittle-Peto method.
- Whitehead methods, which adjust the boundaries from continuous monitoring for discrete monitoring
- error spending methods

You can use the SEQDESIGN procedure to specify methods from the same group for each design. A different method can be specified for each boundary separately, but all methods in a design must be from the same group.

Fixed Boundary Shape Methods

The fixed boundary shape methods include the unified family method (Kittelson and Emerson 1999) and the Haybittle-Peto method (Haybittle 1971; Peto et al. 1976). The unified family methods derive the boundary values with the specified boundary shape. The unified family methods include the Pocock method (Pocock 1977), the O'Brien-Fleming method (O'Brien and Fleming 1979), the power family method (Wang and Tsiatis 1987; Emerson and Fleming 1989; Pampallona and Tsiatis 1994), and the triangular method (Kittelson and Emerson 1999). See the section "Unified Family Methods" on page 5839 for a detailed description of the methods that use the unified family approach.

The Haybittle-Peto method uses a value of 3 for the critical values in interim stages, so that the critical value at the final stage is close to the original design without interim monitoring. In the SEQDESIGN procedure, the Haybittle-Peto method has been generalized to allow for different boundary values at different stages. See the section "Haybittle-Peto Method" on page 5844 for a detailed description of the Haybittle-Peto method.

Whitehead Methods

The Whitehead methods (Whitehead and Stratton 1983; Whitehead 1997, 2001) derive the boundary values by adapting the continuous monitoring tests to the discrete monitoring of group sequential tests. The Type I error probability and power corresponding to the resulting boundaries are extremely close but differ slightly from the specified values because of the approximations used in deriving the tests (Jennison and Turnbull 2000, p. 106). The SEQDESIGN procedure provides the BOUNDARYKEY= option to adjust the boundary value at the final stage for the exact Type I or Type II error probability level. See the section "Whitehead Methods" on page 5845 for a detailed description of Whitehead's methods.

Error Spending Methods

An error spending method (Lan and DeMets 1983) uses the error spending function to specify the error spending at each stage and then uses these error probabilities to derive the boundary values. You can specify these errors explicitly or with an error spending function for these cumulative errors. See the section "Error Spending Methods" on page 5848 for a detailed description of the error spending methods.

Error spending methods derive boundary values at each stage sequentially and require much more computation than other types of methods for group sequential trials with a large number of stages, especially for a two-sided asymmetric design with early stopping to accept H_0, or to reject or accept H_0.

The sample size requirement for some applicable tests can also be computed in the procedure. After the actual data from a clinical trial are collected, you can then use the boundary information created in the SEQDESIGN procedure to perform a group sequential test in the SEQTEST procedure.

Statistical Assumptions for Group Sequential Designs

The SEQDESIGN procedure assumes that with a total number of stages K, the sequence of the standardized test statistics $\{Z_1, Z_2, \ldots, Z_K\}$ has the canonical joint distribution with information levels $\{I_1, I_2, \ldots, I_K\}$ for the parameter θ (Jennison and Turnbull 2000, p. 49):

- $(Z_1, Z_2, \ldots, Z_K)$ is multivariate normal
- $Z_k \sim N\left(\theta\sqrt{I_k}, 1\right), \; k = 1, 2, \ldots, K$
- $\mathrm{Cov}(Z_{k_1}, Z_{k_2}) = \sqrt{(I_{k_1}/I_{k_2})}, \quad 1 \le k_1 \le k_2 \le K$

In terms of the maximum likelihood estimator, $\hat{\theta}_k = Z_k/\sqrt{I_k}, k = 1, 2, \ldots, K$, the canonical joint distribution can be expressed as follows:

- $(\hat{\theta}_1, \hat{\theta}_2, \ldots, \hat{\theta}_K)$ is multivariate normal
- $\hat{\theta}_k \sim N\left(\theta,\ 1/I_k\right),\ k = 1, 2, \ldots, K$
- $\mathrm{Cov}(\hat{\theta}_{k_1}, \hat{\theta}_{k_2}) = 1/I_{k_2},\quad 1 \le k_1 \le k_2 \le K$

Furthermore, in terms of the score statistics $S_k = Z_k\sqrt{I_k}, k = 1, 2, \ldots, K$, the canonical joint distribution can be expressed as follows:

- $(S_1, S_2, \ldots, S_K)$ is multivariate normal
- $S_k \sim N\left(\theta\, I_k,\ I_k\right),\ k = 1, 2, \ldots, K$
- $\mathrm{Cov}(S_{k_1}, S_{k_2}) = \mathrm{Var}(S_{k_1}) = I_{k_1},\quad 1 \le k_1 \le k_2 \le K$

That is, the increments S_1, $S_2 - S_1$, …, and $S_K - S_{(K-1)}$ are independently distributed.

If the test statistic is computed from the data that are not from a normal distribution, such as a binomial distribution, then it is assumed that the test statistic is computed from a large sample such that the statistic has an approximately normal distribution.

If the increments S_1, $S_2 - S_1$, …, and $S_K - S_{(K-1)}$ are not independently distributed, then it is inappropriate to use group sequential methods in the SEQDESIGN procedure. One such example is the Gehan statistic, which is a weighted log-rank statistic for censored data. Refer to Jennison and Turnbull (2000 pp. 232–233, 276–277) and Proschan, Lan, and Wittes (2006 pp. 150–151) for a description of statistics with nonindependent increments.

If a trial stops at an early interim stage with only a small number of responses observed, it can lead to a distrust of the statistical findings, which rely on the assumption that the sample is large (Whitehead 1997, p. 167). A group sequential design can be specified such that at the first interim analysis, there are a sufficient number of responses to ensure that the analysis to be conducted is both reliable and persuasive (Whitehead 1997, p. 167).

Alternatively, a method such as the O'Brien-Fleming method can be used to derive conservative stopping boundary values at very early stages to make the early stop less likely. That is, the trial is stopped in early stages only with overwhelming evidence.

A simple example of the group sequential tests is the test for a normal mean, $\mu = \mu_0$. Suppose $y_1, y_2, \ldots, y_n$ are n observations of a response variable Y in a data set from a normal distribution with an unknown mean μ and a known variance σ^2. Then the maximum likelihood estimate of μ is the sample mean

$$\overline{y} = \frac{1}{n}\sum_{j=1}^{n} y_j$$

The sample mean has a normal distribution with mean μ and variance σ^2/n:

$$\overline{y} \sim N\left(\mu,\ \frac{\sigma^2}{n}\right)$$

An equivalent hypothesis for $\mu = \mu_0$ is $H_0 : \theta = 0$, where $\theta = \mu - \mu_0$. The MLE statistic for θ,

$$\hat{\theta} = \overline{y} - \mu_0 \sim N\left(\theta,\ {I_0}^{-1}\right)$$

where the information $I_0 = n/\sigma^2$.

For a group sequential test with K stages, there are $N_1, N_2, \ldots, N_K$ observations available at these stages. At stage k, the sample mean is computed as

$$\overline{y}_k = \frac{1}{N_k} \sum_{j=1}^{N_k} y_{kj}$$

where y_{kj} is the value of the jth observation available at the kth stage and N_k is the cumulative sample size at stage k, which includes the N_{k-1} observations collected at previous stages and the $N_k - N_{k-1}$ observations collected at the current stage.

The maximum likelihood estimate

$$\hat{\theta}_k = \overline{y}_k - \mu_0 \sim N\left(\theta,\ {I_k}^{-1}\right)$$

where the information

$$I_k = \frac{1}{\mathrm{Var}(\overline{y}_k)} = \frac{N_k}{\sigma^2}$$

is the inverse of the variance.

Thus, the standardized statistic

$$Z_k = \hat{\theta}_k \sqrt{I_k} = (\overline{y}_k - \mu_0)\sqrt{I_k} \sim N\left(\theta \sqrt{I_k},\ 1\right)$$

The covariance of Z_{k_1} and Z_{k_2}, $1 \le k_1 \le k_2 \le K$ can be expressed as

$$\mathrm{Cov}(Z_{k_1}, Z_{k_2}) = \frac{1}{\sqrt{(I_{k_1} I_{k_2})}} \mathrm{Cov}(S_{k_1}, S_{k_2})$$

where $S_{k_1} = Z_{k_1}\sqrt{I_{k_1}}$ and $S_{k_2} = Z_{k_2}\sqrt{I_{k_2}}$.

Since $S_{k_2} - S_{k_1}$ is independent of S_{k_1}, $\mathrm{Cov}(S_{k_1}, S_{k_2}) = \mathrm{Var}(S_{k_1}) = I_{k_1}$ and

$$\mathrm{Cov}(Z_{k_1}, Z_{k_2}) = \frac{1}{\sqrt{(I_{k_1} I_{k_2})}} I_{k_1} = \sqrt{I_{k_1}/I_{k_2}}$$

Thus the statistics $\{Z_1, Z_2, \ldots, Z_K\}$ has the canonical joint distribution with information levels $\{I_1, I_2, \ldots, I_K\}$ for the parameter μ. See the section "Applicable One-Sample Tests and Sample Size Computation" on page 5858, the section "Applicable Two-Sample Tests and Sample Size Computation" on page 5860, and the section "Applicable Regression Parameter Tests and Sample Size Computation" on page 5868 for more examples of applicable tests in group sequential trials.

Boundary Scales

The boundaries computed by the SEQDESIGN procedure are applied to test statistics computed during the analysis, and so generally, the scale you select for the boundaries is determined by the scale of the statistics that you will be using.

The following scales are available in the SEQDESIGN procedure:

- MLE, maximum likelihood estimate
- standardized Z
- score statistic S
- p-value

These scales are all equivalent for a given set of boundary values—that is, there exists a unique transformation between any two of these scales. If you know the boundary values in terms of statistics from one scale, you can uniquely derive the boundary values of statistics for other scales. You can specify the scale with the BOUNDARYSCALE= option, and the default is the standardized Z scale.

You can also select the boundary scale to better examine the features of an individual group sequential design or to compare features among multiple designs. For example, with the standardized Z scale, the boundary values for the Pocock design are identical across all stages, and the O'Brien-Fleming design has boundary values (in absolute value) that decrease over the stages.

The remaining section demonstrates the transformations from one scale to the other scales. If the maximum likelihood estimate $\hat{\theta}$ is computed by the analysis, then

$$\hat{\theta} \sim N\left(\theta,\ \frac{1}{I}\right)$$

where I is the Fisher information if it does not depend on θ. Otherwise, I is either the expected Fisher information evaluated at $\hat{\theta}$ or the observed Fisher information. See the section "Maximum Likelihood Estimator" on page 5818 for a detailed description of these statistics.

With the MLE statistic $\hat{\theta}$, the corresponding standardized Z statistic is computed as

$$Z = \hat{\theta}\sqrt{I} \sim N\left(\theta\sqrt{I},\ 1\right)$$

and the corresponding score statistic is computed as

$$S = \hat{\theta}\, I \sim N\left(\theta\, I,\ I\right)$$

Similarly, if a score statistic S is computed by the analysis, then with

$$S \sim N\left(\theta\, I,\ I\right)$$

where I is the information, either an expected Fisher information ($E_{\theta=0}(I(\theta))$ or $E_{\theta=\hat{\theta}}(I(\theta))$) or an observed Fisher information ($I(0)$ or $I(\hat{\theta})$).

The corresponding standardized Z statistic is computed as

$$Z = \frac{S}{\sqrt{I}} \sim N\left(\theta\sqrt{I},\ 1\right)$$

and the corresponding MLE-scaled statistic is computed as

$$\hat{\theta} = \frac{S}{I} \sim N\left(\theta,\ \frac{1}{I}\right)$$

With a standardized normal Z statistic, the corresponding fixed-sample nominal p-value depends on the type of alternative hypothesis. With an upper alternative, the nominal p-value is defined as the one-sided p-value under the null hypothesis $H_0 : \theta = 0$ with an upper alternative:

$$p_k = 1 - \Phi(Z)$$

With a lower alternative or a two-sided alternative, the nominal p-value is defined as the one-sided p-value under the null hypothesis $H_0 : \theta = 0$ with a lower alternative:

$$p_k = \Phi(Z)$$

which is an increasing function of the standardized Z statistic (Emerson, Kittelson, and Gillen 2005, p. 12).

The BOUNDARYSCALE= MLE, STDZ, SCORE, and PVALUE options display the boundary values in the MLE, standardize Z, score, and p-value scales, respectively. For example, suppose $y_{k1}, y_{k2}, \ldots, y_{kn_k}$ are n_k observations of a response variable Y in a data set from a normal distribution with an unknown mean μ and a known variance σ^2. Then

$$y_{kj} \sim N\left(\mu,\ \sigma^2\right)$$

for $k = 1, 2, \ldots, K$, where K is the number of groups and n_k is the number of observations at group k.

If N_k is the cumulative number of observations for the first k groups, then the sample mean from these N_k observations

$$\overline{y}_k = \frac{1}{N_k} \sum_{j=1}^{N_k} y_{kj}$$

has a normal distribution with mean θ and variance σ^2/N_k:

$$\overline{y}_k \sim N\left(\theta,\ \frac{\sigma^2}{N_k}\right)$$

To test the null hypothesis $\mu = \mu_0$, $H_0 : \theta = 0$, where $\theta = \mu - \mu_0$ can be used. The MLE of θ is $\hat{\theta}_k = \overline{y}_k - \mu_0$ and

$$\hat{\theta}_k \sim N\left(\theta,\ \frac{1}{I_k}\right)$$

where the information is the inverse of the variance of $\overline{y}_k$,

$$I_k = \frac{N_k}{\sigma^2}$$

The corresponding standardized Z statistic is

$$Z_k = \hat{\theta}_k \, I_k^{\frac{1}{2}} \sim N\left(\theta \, I_k^{\frac{1}{2}}, \, 1\right)$$

The score statistic in the SEQDESIGN procedure is then given by

$$S_k = \hat{\theta}_k \, I_k = Z_k \, I_k^{\frac{1}{2}} \sim N\left(\theta \, I_k, \, I_k\right)$$

For a null hypothesis $H_0 : \theta = 0$ with an upper alternative, the nominal p-value of the standardized Z statistic is $p_k = 1 - \Phi(Z_k)$. For a null hypothesis $H_0 : \theta = 0$ with a lower alternative or a two-sided alternative, the nominal p-value of the standardized Z statistic is $p_k = \Phi(Z_k)$.

Boundary Variables

The boundaries created in group sequential trials depend on the type of the alternative hypothesis and the early stopping criterion. Table 77.5 shows the boundaries created with various design specifications.

Table 77.5 Boundary Variables

Specifications		Boundary Variables			
Alternative		**Lower**		**Upper**	
Hypothesis	**Early Stopping**	**Alpha**	**Beta**	**Beta**	**Alpha**
Lower	Accept H_0		X		
	Reject H_0	X			
	Accept/Reject H_0	X	X		
Upper	Accept H_0			X	
	Reject H_0				X
	Accept/Reject H_0			X	X
Two-sided	Accept H_0		X	X	
	Reject H_0	X			X
	Accept/Reject H_0	X	X	X	X

Up to four boundaries can be generated in a group sequential design:

- the upper α boundary, to reject the null hypothesis for the upper alternative
- the upper β boundary, to accept the null hypothesis with an upper alternative

- the lower β boundary, to accept the null hypothesis with a lower alternative
- the lower α boundary, to reject the null hypothesis for the lower alternative

For a two-sided design, the null hypothesis is accepted only if both the hypothesis is accepted with an upper alternative and the hypothesis is accepted with a lower alternative.

For a one-sided design with a lower alternative, only the lower boundaries are created. Similarly, for a one-sided design with an upper alternative, only the upper boundaries are created. For example, Figure 77.10 shows the boundary plot for a one-sided test with an upper alternative.

Figure 77.10 Boundary Plot for One-Sided Test

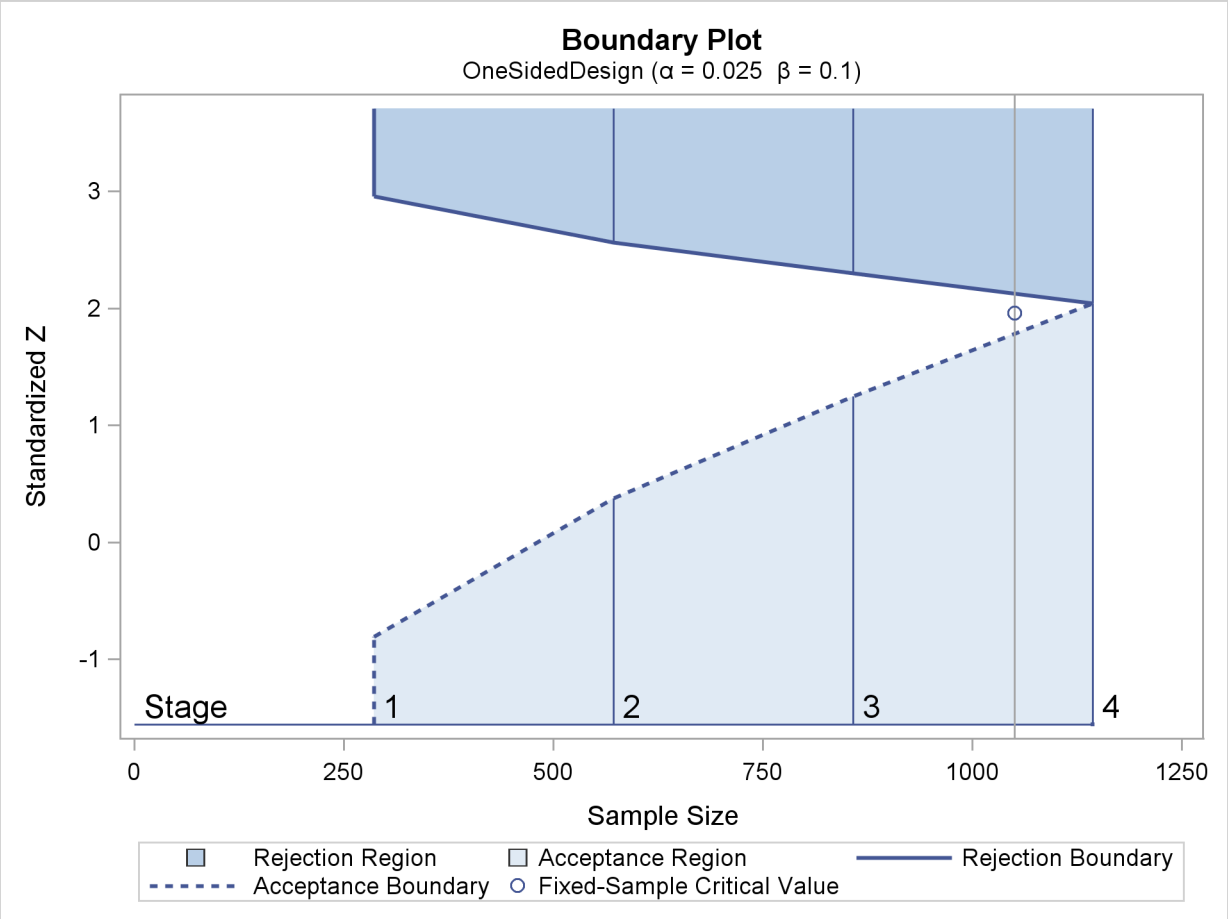


Figure 77.10 corresponds to a one-sided sequential design with early stopping to reject or accept the null hypothesis. For a sequential test with early stopping only to reject the null hypothesis, there are no acceptance boundary values at interim stages. The acceptance boundary value and its associated acceptance region are displayed only at the final stage. Similarly, for a sequential test with early stopping only to accept the null hypothesis, there are no rejection boundary values at interim stages. The rejection boundary value and its associated rejection region are displayed only at the final stage.

For a two-sided design, both the lower and upper boundaries are created. For a design with early stopping to reject the null hypothesis, α boundaries are created. Similarly, for a design with early

stopping to accept the null hypothesis, β boundaries are created. For a design with early stopping to accept or reject the null hypothesis, both the α and β boundaries are created.

For example, Figure 77.11 shows the boundary plot for a two-sided test.

Figure 77.11 Boundary Plot for Two-Sided Test

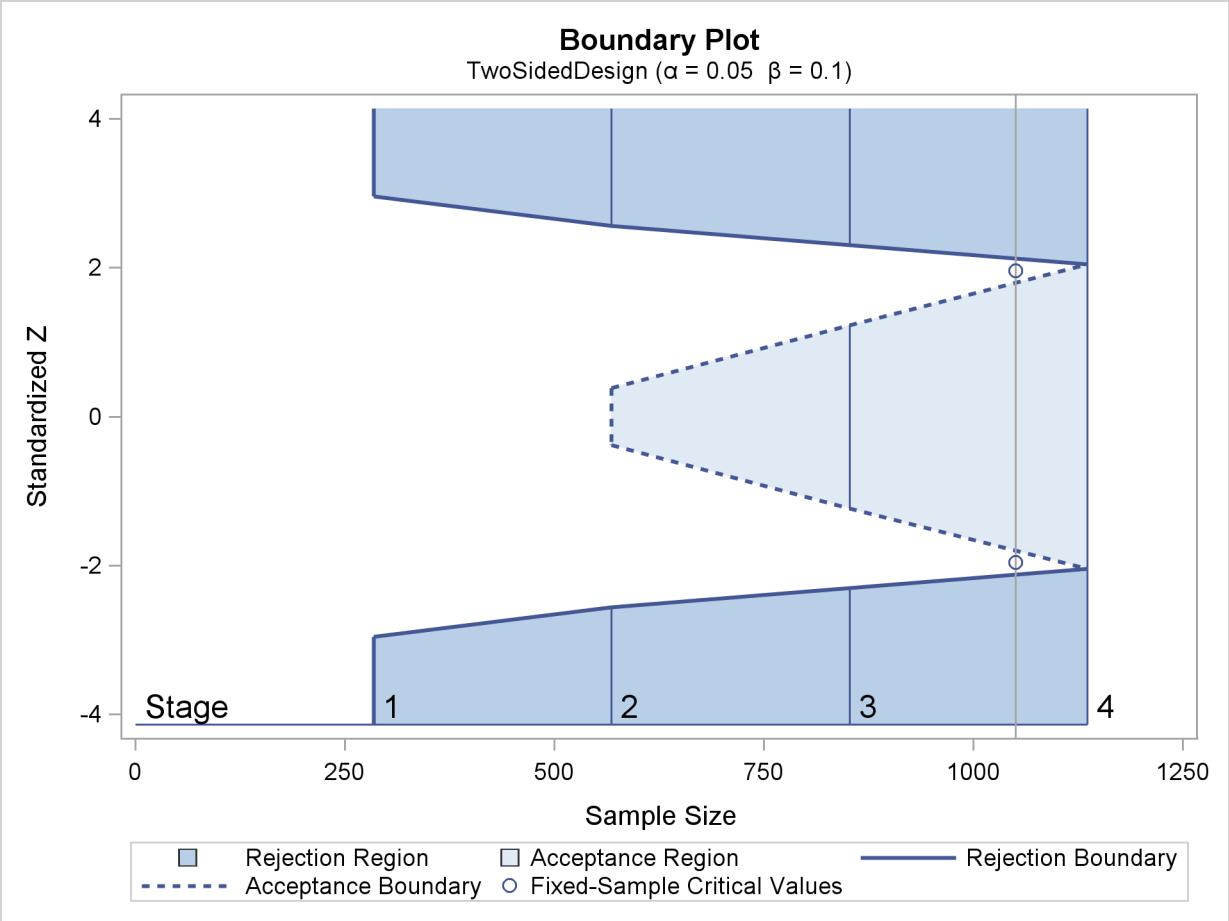


Figure 77.11 corresponds to a two-sided sequential design with early stopping to reject or accept the null hypothesis. For a sequential test with early stopping only to reject the null hypothesis, there are no acceptance boundary values at interim stages. The acceptance boundary value and its associated acceptance region are displayed only at the final stage. Similarly, for a sequential test with early stopping only to accept the null hypothesis, there are no rejection boundary values at interim stages. The rejection boundary value and its associated rejection region are displayed only at the final stage.

Type I and Type II Errors

The Type I error is the error of rejecting the null hypothesis when the null hypothesis is correct, and the Type II error is the error of not rejecting the null hypothesis when the null hypothesis is incorrect. The level of significance α is the probability of making a Type I error. The Type II error depends

on the hypothetical reference of the alternative hypothesis, and the Type II error probability β is defined as the probability of not rejecting the null hypothesis when a specific alternative reference is true. The power $1 - \beta$ is then defined as the probability of rejecting the null hypothesis at the alternative reference.

In a sequential design, if the maximum information and alternative reference are not both specified, the critical values are created such that both the specified Type I and the specified Type II error probability levels are maintained in the design. Otherwise, the critical values are created such that either the specified Type I error probability or the specified Type II error probability is maintained.

One-Sided Tests

For a K-stage group sequential design with an upper alternative hypothesis $H_1 : \theta = \theta_1$ and early stopping to reject or accept the null hypothesis $H_0 : \theta = 0$, the boundaries contain the upper α critical values a_k and upper β critical values $b_k, k = 1, 2, \ldots, K$. At each interim stage, $b_k < a_k$, the null hypothesis H_0 is rejected if the observed statistic $z_k \geq a_k$, H_0 is accepted if $z_k < b_k$, or the process is continued to the next stage if $b_k \leq z_k < a_k$. At the final stage $b_K = a_K$, the hypothesis is either rejected or accepted.

The overall Type I error probability α is given by

$$\alpha = \sum_{k=1}^{K} \alpha_k$$

where α_k is the α spending at stage k. That is, at stage 1,

$$\alpha_1 = P_{\theta=0}(z_1 \geq a_1)$$

At a subsequent stage k,

$$\alpha_k = P_{\theta=0}(b_j \leq z_j < a_j, j = 1, 2, \ldots, k-1, z_k \geq a_k)$$

Similarly, the Type II error probability

$$\beta = \sum_{k=1}^{K} \beta_k$$

where β_k is the β spending at stage k. That is, at stage 1,

$$\beta_1 = P_{\theta=\theta_1}(z_1 < b_1)$$

At a subsequent stage k,

$$\beta_k = P_{\theta=\theta_1}(b_j \leq z_j < a_j, j = 1, 2, \ldots, k-1, z_k < b_k)$$

With an upper alternative hypothesis $H_1 : \theta = \theta_1 > 0$, the power $1 - \beta$ is the probability of rejecting the null hypothesis for the upper alternative.

$$1 - \beta = 1 - \sum_{k=1}^{K} \beta_k = \sum_{k=1}^{K} P_{\theta=\theta_1}(b_j \le z_j < a_j, j = 1, 2, \ldots, k-1, z_k \ge a_k)$$

For a design with early stopping to reject H_0 only, the interim upper β critical values are set to $-\infty$, $b_k = -\infty, k = 1, 2, \ldots, K-1$, and $\beta = \beta_K$. For a design with early stopping to accept H_0 only, the interim upper α critical values are set to ∞, $a_k = \infty, k = 1, 2, \ldots, K-1$, and $\alpha = \alpha_K$.

Similarly, the Type I and Type II error probabilities for a K-stage design with a lower alternative hypothesis $H_0 : \theta = -\theta_1$ can also be derived.

Two-Sided Tests

For a K-stage group sequential design with two-sided alternative hypotheses $H_{1u} : \theta = \theta_{1u}$ and $H_{1l} : \theta = \theta_{1l}$, and early stopping to reject or accept the null hypothesis $H_0 : \theta = 0$, the boundaries contain the upper α critical values a_k, upper β critical values b_k, lower β critical values $\_b_k$, and lower α critical values $\_a_k$, $k = 1, 2, \ldots, K$. At each interim stage, $\_a_k < \_b_k \le b_k < a_k$, the null hypothesis H_0 is rejected if the observed statistic $z_k \ge a_k$ or $z_k \le \_a_k$, H_0 is accepted if $\_b_k < z_k < b_k$, or the process is continued to the next stage if $b_k \le z_k < a_k$ or $\_a_k < z_k \le \_b_k$. At the final stage $b_K = a_K$ and $\_b_K = \_a_K$, the hypothesis is either rejected or accepted.

The overall upper Type I error probability α_u is given by

$$\alpha_u = \sum_{k=1}^{K} \alpha_{uk}$$

where α_{uk} is the α spending at stage k for the upper alternative. That is, at stage 1,

$$\alpha_{u1} = P_{\theta=0}(z_1 \ge a_1)$$

At a subsequent stage k,

$$\alpha_{uk} = P_{\theta=0}(\_a_j < z_j \le \_b_j \text{ or } b_j \le z_j < a_j, j = 1, 2, \ldots, k-1, z_k \ge a_k)$$

Similarly, the overall lower Type I error probability α_l can also be derived, and the overall Type I error probability $\alpha = \alpha_l + \alpha_u$.

The overall upper Type II error probability β_u is given by

$$\beta_u = \sum_{k=1}^{K} \beta_{uk}$$

where β_{uk} is the upper β spending at stage k. That is, at stage 1,

$$\beta_{u1} = P_{\theta=\theta_{1u}}(z_1 < \_a_1 \text{ or } \_b_1 < z_1 < b_1)$$

At a subsequent stage k,

$$\beta_{uk} = P_{\theta=\theta_{1u}}(\_a_j < z_j \le \_b_j \text{ or } b_j \le z_j < a_j,\ j = 1, 2, \ldots, k-1,\ z_k < \_a_k \text{ or } \_b_k < z_k < b_k)$$

With an upper alternative hypothesis $H_1 : \theta = \theta_{1u} > 0$, the power $1 - \beta_u$ is the probability of rejecting the null hypothesis for the upper alternative:

$$1 - \beta_u = 1 - \sum_{k=1}^{K} \beta_{uk}$$

which is

$$P_{\theta=\theta_{1u}}(\_a_j < z_j \le \_b_j \text{ or } b_j \le z_j < a_j,\ j = 1, 2, \ldots, k-1,\ z_k \ge a_k)$$

The overall lower Type II error probability β_l and power $1 - \beta_l$ can be similarly derived.

For a design with early stopping only to reject H_0, both the interim lower and upper β critical values are set to missing, $k = 1, 2, \ldots, K-1$, and $\beta_{lK} = \beta_l$, $\beta_{uK} = \beta_u$. For a design with early stopping only to accept H_0, the interim upper α critical values are set to ∞, $a_{uk} = \infty$, and the interim lower α critical values are set to $-\infty$, $a_{lk} = -\infty$, $k = 1, 2, \ldots, K-1$, and $\alpha_{uK} = \alpha_u$, $\alpha_{lK} = \alpha_l$.

Unified Family Methods

Unified family methods (Kittelson and Emerson 1999) derive boundary values with a specified boundary shape. For example, Pocock's method (Pocock 1977) derives equal boundary values for all stages in the standardized Z scale. In addition to Pocock's method, the unified family methods include the O'Brien-Fleming, power family, and unified family triangular methods.

The boundary values at each stage depend on the information fractions

$$\Pi_k = \frac{I_k}{I_X}$$

where I_k is the information available at stage k and I_X is the maximum information, the information available at the end of the trial if the trial does not stop early.

Boundary Values in Standardized *Z* Scale

With the unified family method, the boundary values for the upper α boundary $Z_{\alpha u}$, upper β boundary $Z_{\beta u}$, lower β boundary $Z_{\beta l}$, and lower α boundary $Z_{\alpha l}$, using the standardized normal scale, are given by the following:

- $Z_{\alpha u}(\Pi_k) = f_{\alpha u}(\Pi_k)\, C_{\alpha u}$
- $Z_{\beta u}(\Pi_k) = \theta_{1u}\, I_k^{\frac{1}{2}} - f_{\beta u}(\Pi_k)\, C_{\beta u}$

- $Z_{\beta l}(\Pi_k) = \theta_{1l}\, I_k^{\frac{1}{2}} + f_{\beta l}(\Pi_k)\, C_{\beta l}$
- $Z_{\alpha l}(\Pi_k) = -f_{\alpha l}(\Pi_k)\, C_{\alpha l}$

where $\theta_{1l}(< 0)$ and $\theta_{1u}(> 0)$ are the lower and upper alternative references, $f_{\alpha l}(\Pi_k)$, $f_{\beta l}(\Pi_k)$, $f_{\beta u}(\Pi_k)$, and $f_{\alpha u}(\Pi_k)$ are the specified shape functions, and $C_{\alpha l}$, $C_{\beta l}$, $C_{\beta u}$, and $C_{\alpha u}$ are the critical values derived to achieve the specified α and β levels.

If a derived lower β boundary value $Z_{\beta l}(\Pi_k)$ is greater than its corresponding upper β boundary value $Z_{\beta u}(\Pi_k)$, then both values are set to missing.

Note that the drift parameters $d_l = \theta_{1l}\sqrt{I_X}$ and $d_u = \theta_{1u}\sqrt{I_X}$ are derived in the SEQDESIGN procedure. The boundary values in standardized Z scale can be derived without specifying the maximum information and alternative reference.

Shape Parameters

The shape function in the SEQDESIGN procedure is given by

$$f(\Pi_k) = f(\Pi_k; \tau, \rho) = \tau\, \Pi_k^{\frac{1}{2}} + \Pi_k^{-\rho} = \Pi_k^{\frac{1}{2}}\, (\tau + \Pi_k^{-(\rho+\frac{1}{2})})$$

where the parameters $\rho \geq 0$ and $0 \leq \tau \leq 2\rho$. can be specified for each boundary separately.

The parameters τ and ρ determine the shape of the boundaries. Special cases of the unified family methods also include power family methods and triangular methods. Table 77.6 summarizes the corresponding parameter values in the unified family for these methods.

Table 77.6 Parameters in the Unified Family for Various Methods

Method	Option	Unified Family	
		Rho	Tau
Pocock	POC	0	0
O'Brien-Fleming	OBF	0.5	0
Power family	POW (RHO=ρ)	ρ	0
Triangular	TRI (TAU=τ)	0.5	τ

Note that the power parameter $\rho = 1/2 - \Delta = \rho^* - 1/2$, where Δ is the power parameter used in Jennison and Turnbull (2000) and Wang and Tsiatis (1987) and ρ^* is the power parameter used in Kittelson and Emerson (1999).

Also note that instead of the three parameters used in the unified family methods by Kittelson and Emerson (1999), only two parameters are used in the SEQDESIGN procedure. The other parameter is fixed at zero.

Boundary Values in MLE Scale

If the maximum information is available, the boundary values derived from a unified family method can also be displayed in the MLE scale:

- $\theta_{\alpha u}(\Pi_k) = I_k^{-\frac{1}{2}} f_{\alpha u}(\Pi_k) C_{\alpha u}$
- $\theta_{\beta u}(\Pi_k) = \theta_{1u} - I_k^{-\frac{1}{2}} f_{\beta u}(\Pi_k) C_{\beta u}$
- $\theta_{\beta l}(\Pi_k) = \theta_{1l} + I_k^{-\frac{1}{2}} f_{\beta l}(\Pi_k) C_{\beta l}$
- $\theta_{\alpha l}(\Pi_k) = -I_k^{-\frac{1}{2}} f_{\alpha l}(\Pi_k) C_{\alpha l}$

These MLE scale boundary values are computed by multiplying $I_k^{-\frac{1}{2}}$ by the standardized Z scale boundary values at stage k.

Boundary Values in Score Scale

If the maximum information is available, the boundary values derived from a unified family method can also be displayed in the score scale:

- $S_{\alpha u}(\Pi_k) = I_k^{\frac{1}{2}} f_{\alpha u}(\Pi_k) C_{\alpha u}$
- $S_{\beta u}(\Pi_k) = \theta_{1u} I_k - I_k^{\frac{1}{2}} f_{\beta u}(\Pi_k) C_{\beta u}$
- $S_{\beta l}(\Pi_k) = \theta_{1l} I_k + I_k^{\frac{1}{2}} f_{\beta l}(\Pi_k) C_{\beta l}$
- $S_{\alpha l}(\Pi_k) = -I_k^{\frac{1}{2}} f_{\alpha l}(\Pi_k) C_{\alpha l}$

These MLE scale boundary values are computed by multiplying $I_k^{\frac{1}{2}}$ by the standardized Z scale boundary values at stage k.

Boundary Values in *p*-Value Scale

For a design with a lower alternative or a two-sided alternative, the p-value scale boundary values are the cumulative normal distribution function values of the standardized Z boundary values:

- $P_{\alpha u}(\Pi_k) = \Phi(Z_{\alpha u}(\Pi_k))$
- $P_{\beta u}(\Pi_k) = \Phi(Z_{\beta u}(\Pi_k))$
- $P_{\beta l}(\Pi_k) = \Phi(Z_{\beta l}(\Pi_k))$
- $P_{\alpha l}(\Pi_k) = \Phi(Z_{\alpha l}(\Pi_k))$

These nominal p-values are the one-sided fixed-sample p-values under the null hypothesis with a lower alternative.

For a one-sided design with an upper alternative, the p-value scale boundary values are the one-sided fixed-sample p-values under the null hypothesis with an upper alternative:

- $P_{\alpha u}(\Pi_k) = 1 - \Phi(Z_{\alpha u}(\Pi_k))$
- $P_{\beta u}(\Pi_k) = 1 - \Phi(Z_{\beta u}(\Pi_k))$

Pocock's Method

The shape function for Pocock's method (Pocock 1977) is given by

$$f(\Pi_k) = 1$$

The resulting boundary values for a two-sided design with an early stopping to reject the null hypothesis $H_0 : \theta = 0$ are as follows:

- $Z_{\alpha u}(\Pi_k) = C_{\alpha u}$
- $Z_{\alpha l}(\Pi_k) = -C_{\alpha l}$

That is, the rejection boundary values are constant over all stages of different information levels in the standardized Z scale.

Note that compared with other designs, Pocock's design tends to stop the trials early with a larger p-value. For a new treatment, Pocock's design to stop a trial early with a large p-value might not be persuasive enough to make a new treatment widely accepted (Pocock and White 1999). A Pocock design is illustrated in Example 77.3.

O'Brien-Fleming Method

The shape function for the O'Brien-Fleming method (O'Brien and Fleming 1979) is given by

$$f(\Pi_k) = \Pi_k^{-\frac{1}{2}}$$

The resulting boundary values for a two-sided design with early stopping to reject the null hypothesis $H_0 : \theta = 0$ are as follows:

- $Z_{\alpha u}(\Pi_k) = \Pi_k^{-\frac{1}{2}} C_{\alpha u}$
- $Z_{\alpha l}(\Pi_k) = -\Pi_k^{-\frac{1}{2}} C_{\alpha l}$

That is, the rejection boundaries are inversely proportional to the square root of the information levels in the standardized Z scale.

In the score scale, these boundaries can be displayed as follows:

- $S_{\alpha u}(\Pi_k) = C_{\alpha u}\, I_X^{\frac{1}{2}}$
- $S_{\alpha l}(\Pi_k) = -C_{\alpha l}\, I_X^{\frac{1}{2}}$

which are constants over all stages in the score scale. An O'Brien-Fleming design is illustrated in Example 77.2.

Power Family Method

The shape function for a power family method (Wang and Tsiatis 1987; Emerson and Fleming 1989; Pampallona and Tsiatis 1994) is given by

$$f(\Pi_k) = \Pi_k^{-\rho}$$

The resulting boundary values for a two-sided design with early stopping to reject the null hypothesis $H_0 : \theta = 0$ are as follows:

- $Z_{\alpha u}(\Pi_k) = \Pi_k^{-\rho}\, C_{\alpha u}$
- $Z_{\alpha l}(\Pi_k) = -\Pi_k^{-\rho}\, C_{\alpha l}$

The rejection boundaries depend on the power parameter ρ. The power family includes the Pocock and O'Brien-Fleming methods, and the power parameter is used to allow continuous movement between these two methods. A power family design is illustrated in Example 77.5.

Triangular Method

The shape function for a triangular method (Kittelson and Emerson 1999) in the unified family is given by

$$f(\Pi_k) = \Pi_k^{-\frac{1}{2}} + \tau\, \Pi_k^{\frac{1}{2}}$$

The resulting boundary values for a two-sided design with early stopping to reject the null hypothesis $H_0 : \theta = 0$ are as follows:

- $Z_{\alpha u}(\Pi_k) = (\Pi_k^{-\frac{1}{2}} + \tau\, \Pi_k^{\frac{1}{2}})\, C_{\alpha u} = C_{\alpha u}\, \Pi_k^{-\frac{1}{2}}\, (1 + \tau\, \Pi_k)$
- $Z_{\alpha l}(\Pi_k) = -(\Pi_k^{-\frac{1}{2}} + \tau\, \Pi_k^{\frac{1}{2}})\, C_{\alpha l} = -C_{\alpha l}\, \Pi_k^{-\frac{1}{2}}\, (1 + \tau\, \Pi_k)$

In the score scale, these boundaries are as follows:

- $S_{\alpha u}(\Pi_k) = C_{\alpha u}\, I_X^{\frac{1}{2}}\,(1 + \tau\,\Pi_k) = C_{\alpha u}\, I_X^{\frac{1}{2}} + C_{\alpha u}\tau I_X^{-\frac{1}{2}}\, I_k$
- $S_{\alpha l}(\Pi_k) = -C_{\alpha l}\, I_X^{\frac{1}{2}}\,(1 + \tau\,\Pi_k) = -C_{\alpha l}\, I_X^{\frac{1}{2}} - C_{\alpha l}\tau I_X^{-\frac{1}{2}}\, I_k$

Thus, in the score scale, the boundary function is a linear function of the information I_k. With these straight-line boundaries, a triangular method for a one-sided trial with early stopping to reject or accept the null hypothesis produces a triangular continuation region. Similarly, for a two-sided design, the continuation region is a union of two separate triangular regions. A triangular method is illustrated in Example 77.6.

Haybittle-Peto Method

The Haybittle-Peto method (Haybittle 1971; Peto et al. 1976) uses a value of 3 for the critical values in interim stages, so that the critical value at the final stage is close to the original design without interim monitoring.

In the SEQDESIGN procedure, the Haybittle-Peto method has been generalized to allow for different boundary values at different stages. That is, with the standardized normal scale, the boundary values are given by the following:

- $Z_{\alpha u}(\Pi_k) = z_{\alpha uk}$
- $Z_{\beta u}(\Pi_k) = \theta_{1u}\, I_k^{\frac{1}{2}} - z_{\beta uk}$
- $Z_{\beta l}(\Pi_k) = \theta_{1l}\, I_k^{\frac{1}{2}} + z_{\beta lk}$
- $Z_{\alpha l}(\Pi_k) = -z_{\alpha lk}$

where θ_{1l} and θ_{1u} are the lower and upper alternative references and the boundary values $z_{\alpha uk}$, $z_{\beta uk}$, $z_{\beta lk}$, and $z_{\alpha lk}$ are specified either explicitly with the HP(Z= *numbers*) option or implicitly with the HP(PVALUE= *numbers*) option. The HP(PVALUE= *numbers*) option specifies the nominal p-values p_k for the corresponding boundary values z_k:

$$z_k = \Phi^{-1}(1 - p_k)$$

The Haybittle-Peto method is illustrated in Example 77.5.

Whitehead Methods

The Whitehead methods (Whitehead and Stratton 1983; Whitehead 1997, 2001) derive boundary values by adjusting the boundary values generated from continuous monitoring. With continuous monitoring, the boundary values are on a straight line in the score scale for each boundary. For a group sequential design, the boundary values at an interim stage k depend on the information fractions

$$\Pi_k = \frac{I_k}{I_X}$$

where I_k is the information available at stage k and I_X is the maximum information, the information available at the end of the trial if the trial does not stop early.

One-Sided Symmetric Designs

A one-sided symmetric design is a one-sided design with identical Type I and Type II error probabilities. For a one-sided symmetric design with an upper alternative, $\alpha_u = \beta_u$, the boundary values in the score scale from continuous monitoring are as follows:

- $S_{\alpha u}(\Pi_k) = C_u\theta_u^{-1} + \tau_u\theta_u I_k$
- $S_{\beta u}(\Pi_k) = \theta_u\, I_k - (C_u\theta_u^{-1} - \tau_u\theta_u I_k)$

where θ_u is the upper alternative reference, τ_u is a specified constant for the slope, $0 \le \tau_u < \frac{1}{2}$, and C_u is a constant, fixed for STOP=BOTH and derived for STOP=ACCEPT and STOP=REJECT.

The upper β boundary value can also be expressed as

- $S_{\beta u}(\Pi_k) = -C_u\theta_u^{-1} + (1-\tau_u)\theta_u I_k$

Thus, these straight-line boundaries form a triangle in the score statistic scale.

To adjust for the nature of discrete monitoring, the group sequential boundary values are given by the following:

- $S_{\alpha u}(\Pi_k) = C_u\theta_u^{-1} + \tau_u\theta_u I_k - g_k$
- $S_{\beta u}(\Pi_k) = -C_u\theta_u^{-1} + (1-\tau_u)\theta_u I_k + g_k$

where $g_1 = 0.583\sqrt{I_1}$ and $g_k = 0.583\sqrt{I_k - I_{(k-1)}}$, $k > 1$ are the adjustments.

Note that with the adjustment g_k, the resulting boundaries form a Christmas tree shape within the original triangle and are referred to as the Christmas tree boundaries (Whitehead 1997, p. 73).

One-Sided Asymmetric Designs

For a one-sided asymmetric design with an upper alternative, $\alpha_u \neq \beta_u$, the boundary values computed using the score scale, are given by the following:

- $S_{\alpha u}(\Pi_k) = C_u \tilde{\theta}_u^{-1} + \tau_u \tilde{\theta}_u I_k - g_k$
- $S_{\beta u}(\Pi_k) = -C_u \tilde{\theta}_u^{-1} + (1-\tau_u) \tilde{\theta}_u I_k + g_k$

where $\tilde{\theta_u}$ is the modified alternative reference

$$\tilde{\theta_u} = \frac{2\Phi^{-1}(1-\alpha_u)}{\Phi^{-1}(1-\alpha_u) + \Phi^{-1}(1-\beta_u)}\,\theta_u$$

The modified alternative reference $\tilde{\theta}_u = \theta_u$ if $\alpha_u = \beta_u$.

For a design with early stopping to reject or accept the null hypothesis, $S_{\alpha u}(1) = S_{\beta u}(1)$, the boundary values at the final stage are equal. The modified drift parameter $\tilde{d_u}$ is given by

$$\tilde{d_u} = \tilde{\theta}_u \sqrt{I_X} = \frac{1}{1-2\tau_u}\left(\sqrt{{h_K}^2 + 2C_u(1-2\tau_u)} - h_K\right)$$

where $h_K = g_K\, {I_X}^{-\frac{1}{2}} = 0.583\sqrt{1-\Pi_{(K-1)}}$.

A one-sided Whitehead design with early stopping to reject or accept the null hypothesis is illustrated in Example 77.7.

Two-Sided Designs

The boundary values for a two-sided design are generated by combining boundary values from two one-sided designs. With the STOP=BOTH option, this produces a double triangular design (Whitehead 1997, p. 98).

The boundary values for a two-sided design, using the score scale, are then given by the following:

- $S_{\alpha u}(\Pi_k) = C_u \tilde{\theta}_u^{-1} + \tau_u \tilde{\theta}_u I_k - g_k$
- $S_{\beta u}(\Pi_k) = -C_u \tilde{\theta}_u^{-1} + (1-\tau_u) \tilde{\theta}_u I_k + g_k$
- $S_{\beta l}(\Pi_k) = -C_l \tilde{\theta}_l^{-1} + (1-\tau_l) \tilde{\theta}_l I_k - g_k$
- $S_{\alpha l}(\Pi_k) = C_l \tilde{\theta}_l^{-1} + \tau_l \tilde{\theta}_l I_k + g_k$

where the modified alternative references are

$$\tilde{\theta_u} = \frac{2\Phi^{-1}(1-\alpha_u)}{\Phi^{-1}(1-\alpha_u) + \Phi^{-1}(1-\beta_u)}\,\theta_u$$

$$\tilde{\theta}_l = \frac{2\Phi^{-1}(1-\alpha_l)}{\Phi^{-1}(1-\alpha_l)+\Phi^{-1}(1-\beta_l)}\,\theta_l$$

The modified alternative reference $\tilde{\theta}_u = \theta_u$ if $\alpha_u = \beta_u$ and $\tilde{\theta}_l = \theta_l$ if $\alpha_l = \beta_l$.

For a design with early stopping to reject or accept the null hypothesis, the two upper boundary values at the final stage are identical and the two lower boundary values at the final stage are identical. That is, $S_{\alpha l}(1) = S_{\beta l}(1)$ and $S_{\alpha u}(1) = S_{\beta u}(1)$. These modified drift parameters are then given by

$$\tilde{d}_l = \tilde{\theta}_l\sqrt{I_X} = \frac{1}{1-2\tau_l}\left(\sqrt{{h_K}^2 + 2C_l(1-2\tau_l)} - h_K\right)$$

$$\tilde{d}_u = \tilde{\theta}_u\sqrt{I_X} = \frac{1}{1-2\tau_u}\left(\sqrt{{h_K}^2 + 2C_u(1-2\tau_u)} - h_K\right)$$

where $h_K = g_K I_X^{-\frac{1}{2}} = 0.583\sqrt{1-\Pi_{(K-1)}}$.

For a design with early stopping to reject the null hypothesis, or a design with early stopping to accept the null hypothesis, you can specify the slope parameters τ_u and τ_l in the TAU= option, and then the intercept parameters C_u and C_l, and the resulting boundary values are derived. If both the maximum information and alternative references are specified, the procedure derives C_u and C_l by maintaining either the overall α levels (BOUNDARYKEY=ALPHA) or the overall β levels (BOUNDARYKEY=BETA). If the maximum information and alternative reference are not both specified, the procedure derives the boundary values C_u and C_l by maintaining both the overall α and overall β levels.

For a design with early stopping to reject or accept the null hypothesis (STOP=BOTH), Whitehead's triangular test uses $\tau_u = \tau_l = 0.25$ and compute $C_u = -2\log(2\alpha_u)$ and $C_l = -2\log(2\alpha_l)$ for the boundary values. If the maximum information and alternative reference are both specified, the BOUNDARYKEY=ALPHA option uses the specified α values to compute the β values and boundary values. The final-stage boundary values are modified to maintain the overall α levels if they exist. Similarly, the BOUNDARYKEY=BETA option uses the specified β values to compute the α values and boundary values. The final-stage boundary values are modified to maintain the overall β levels if they exist.

If the maximum information and alternative reference are not both specified, the specified α and β values are used to derive boundary values. The BOUNDARYKEY=NONE option uses these boundary values without adjustment. The BOUNDARYKEY=ALPHA option modifies the final-stage boundary values to maintain the overall α levels if they exist. Similarly, the BOUNDARYKEY=BETA option modifies the final-stage boundary values to maintain the overall β levels if they exist.

Two-sided Whitehead designs with early stopping to reject the null hypothesis are illustrated in Example 77.9.

Applicable Boundary Keys

Table 77.7 lists applicable boundary keys for a design that uses Whitehead methods.

Table 77.7 Applicable Boundary Keys for Whitehead Methods

	Specified Parameters		Boundary Keys			
Early Stopping	**(Alt Ref – Max Info)**	**Tau**	**Alpha**	**Beta**	**None**	**Both**
Reject H_0	X	X	X	X		
Accept H_0	X	X	X	X		
Reject/Accept H_0	X	0.25	X	X		
Reject H_0		X				X
Accept H_0		X				X
Reject/Accept H_0		0.25	X	X	X	

Note that the symbol "X" under "(Alt Ref – Max Info)" indicates that both alternative reference and maximum information are specified.

For a design with early stopping to reject the null hypothesis, or a design with early stopping to accept the null hypothesis, you can specify the slope parameter τ_u in the TAU= option, and then the intercept parameter C_u and the resulting boundary values are derived. If both the maximum information and alternative reference are specified, the procedure derives C_u by maintaining either the overall α levels (BOUNDARYKEY=ALPHA) or the overall β levels (BOUNDARYKEY=BETA). If the maximum information and alternative reference are not both specified, the procedure derives the boundary values and C_u by maintaining both the overall α and overall β levels.

For a design with early stopping to reject or accept the null hypothesis (STOP=BOTH), Whitehead's triangular test uses $\tau_u = 0.25$ and solves $C_u = 2\log(\frac{1}{2\alpha_u})$ for the boundary values. If the maximum information and alternative reference are both specified, the BOUNDARYKEY=ALPHA option uses the specified α value to compute the β value and boundary values. The final-stage boundary value is modified to maintain the overall α level if it exists. Similarly, the BOUNDARYKEY=BETA option uses the specified β value to compute the α value and boundary values. The final-stage boundary value is modified to maintain the overall β level if it exists.

If the maximum information and alternative reference are not both specified, the specified α and β values are used to derive boundary values. The BOUNDARYKEY=NONE option uses these boundary values without adjustment. The BOUNDARYKEY=ALPHA option modifies the final-stage boundary value to maintain the overall α level if it exists. Similarly, the BOUNDARYKEY=BETA option modifies the final-stage boundary value to maintain the overall β level if it exists.

Error Spending Methods

For each sequential design, the α and β errors spent at each stage can be computed from the boundary values. For example, for a K-stage design with an upper alternative hypothesis $H_1 : \theta = \theta_1$ and early stopping to reject the null hypothesis $H_0 : \theta = 0$, the boundary values in a standardized Z

scale are the upper α critical values $a_k, k = 1, 2, \ldots, K$. At each interim stage, the null hypothesis H_0 is rejected if the observed standardized Z statistic $z_k \geq a_k$. Otherwise, the process continues to the next stage. At the final stage, the hypothesis is rejected if $z_K \geq a_K$. Otherwise, the null hypothesis is accepted.

The boundary values a_k are derived such that the overall Type I error probability

$$\alpha = \sum_{k=1}^{K} \alpha_k$$

where α_k is the α spending at stage k. That is, at stage 1,

$$\alpha_1 = P_{\theta=0}\,(\,z_1 \geq a_1\,)$$

At a subsequent stage k,

$$\alpha_k = P_{\theta=0}\,(\,z_j < a_j, j = 1, 2, \ldots, k-1,\ z_k \geq a_k\,)$$

Since each design can be uniquely identified by the α and β errors spent at each stage, a design can then be derived by specifying the α and β errors to be used at each stage. The error spending method (Lan and DeMets 1983) distributes the error to be used at each stage and then derives the boundary values. Numerous forms of the error spending function are available. Kim and DeMets (1987) examine the functions $f(t) = t$, $f(t) = t^{\frac{3}{2}}$, and $f(t) = t^2$, where t is the information fraction. Jennison and Turnbull (2000, p. 148) generalize these functions to the power functions $f(t; \rho) = t^\rho, \rho > 0$.

The ERRFUNCPOC option uses the cumulative error spending function (Lan and DeMets 1983)

$$E(t) = \begin{cases} 1 & \text{if } t \geq 1 \\ \log(\,1 + (e-1)t) & \text{if } 0 < t < 1 \\ 0 & \text{otherwise} \end{cases}$$

With a specified error of α or β, the cumulative error spending at stage k is $\alpha\, E(\Pi_k)$ or $\beta\, E(\Pi_k)$, where $\Pi_k = I_k / I_X$ is the information fraction at stage k. The method produces boundaries similar to those produced with Pocock's method.

The ERRFUNCOBF option uses the cumulative error spending function (Lan and DeMets 1983)

$$E(t; a) = \begin{cases} 1 & \text{if } t \geq 1 \\ \frac{1}{a}\, 2\,(1 - \Phi(\frac{z_{(1-a/2)}}{\sqrt{t}})) & \text{if } 0 < t < 1 \\ 0 & \text{otherwise} \end{cases}$$

where a is either α for the α spending function or β for the β spending function. That is, with a specified error of α or β, the cumulative error spending at stage k is $\alpha\, E(\Pi_k; \alpha)$ or $\beta\, E(\Pi_k; \beta)$. The method produces boundaries similar to those produced with the O'Brien-Fleming method.

The ERRFUNCGAMMA option uses the gamma cumulative error spending function (Hwang, Shih, and DeCani 1990)

$$E(t;\gamma) = \begin{cases} 1 & \text{if } t \geq 1 \\ \frac{1-e^{-\gamma t}}{1-e^{-\gamma}} & \text{if } 0 < t < 1, \gamma \neq 0 \\ t & \text{if } 0 < t < 1, \gamma = 0 \\ 0 & \text{otherwise} \end{cases}$$

where γ is the parameter γ specified in the GAMMA= option. That is, with a specified error of α or β, the cumulative error spending at stage k is $\alpha\, E(\Pi_k;\gamma)$ or $\beta\, E(\Pi_k;\gamma)$.

The ERRFUNCPOW option uses the cumulative error spending function (Jennison and Turnbull 2000, p. 148)

$$E(t;\rho) = \begin{cases} 1 & \text{if } t \geq 1 \\ t^{\rho} & \text{if } 0 < t < 1 \\ 0 & \text{otherwise} \end{cases}$$

where ρ is the power parameter specified in the RHO= option. That is, with a specified error of α or β, the cumulative error spending at stage k is $\alpha\, E(\Pi_k;\rho)$ or $\beta\, E(\Pi_k;\rho)$.

Error spending methods derive boundary values at each stage sequentially and require much more computation than other types of methods for group sequential trials with a large number of stages, especially for a two-sided asymmetric design with early stopping to accept H_0, or to reject or accept H_0.

Note that for a two-sided design with the STOP=BOTH or STOP=ACCEPT option, at each interim stage, the SEQDESIGN procedure first produces the lower and upper β boundary values based on the one-sided β spending. If the lower β boundary value is greater than or equal to its corresponding upper β boundary value, there is no early stopping to accept the null hypothesis at this stage, and the corresponding β spending is distributed proportionally to the remaining stages.

For the error spending functions not available in the SEQDESIGN procedure, you can first compute the corresponding error spending at each stage explicitly, then use the SEQDESIGN procedure with the ERRSPEND= option to specify these errors directly.

For example, if the information levels are equally spaced in a five-stage design, the option ERRFUNCPOW (RHO=2) produces relative cumulative errors of $(1/5)^2$, $(2/5)^2$, $(3/5)^2$, $(4/5)^2$, and 1. This is equivalent to using the option ERRSPEND (1 4 9 16 25).

A one-sided error spending design is illustrated in Example 77.8 and a two-sided asymmetric error spending design is illustrated in Example 77.11.

Boundary Adjustments for Overlapping Lower and Upper β Boundaries

For the fixed boundary shape methods and Whitehead methods, the boundary values for all stages are derived simultaneously for each boundary. For a two-sided design with STOP=ACCEPT or

STOP=BOTH, simultaneous derivation might result in overlapping of the lower and upper β boundaries. That is, at an interim stage k, the lower β boundary value might be greater than its corresponding upper β boundary value. In this case, these two β boundary values are set to missing and the design does not stop at stage k to accept the null hypothesis (Jennison and Turnbull 2000, p. 113).

For the error spending methods, the boundary values are derived sequentially for the stages. For a two-sided design with STOP=ACCEPT or STOP=BOTH, a small β spending at an interim stage might result in overlapping of the lower and upper β boundaries for the two corresponding one-sided tests. Specifically, this form of overlapping occurs at an interim stage k if the upper β boundary value derived from the one-sided test for the upper alternative is less than the lower β boundary value derived from the one-sided test for the lower alternative (Kittelson and Emerson 1999, pp. 881–882; Rudser and Emerson 2007, p. 6). You can use the BETAOVERLAP= option to specify how this type of overlapping is to be handled.

If BETAOVERLAP=ADJUST (which is the default) is specified, the procedure derives the boundary values for the two-sided design and then checks for overlapping of the two one-sided β boundaries at interim stages. If overlapping occurs at a particular stage, the β boundary values for the two-sided design are set to missing (so the trial does not stop to accept the null hypothesis at this stage), and the β spending values at subsequent stages are adjusted proportionally as follows.

If the β boundary values are set to missing at stage k in a K-stage trial, the adjusted β spending value at stage k, e'_k, is updated for these missing β boundary values, and then the β spending values at subsequent stages are adjusted proportionally by

$$e'_j = e'_k + \frac{e_j - e_k}{e_K - e_k}\,(e_K - e'_k)$$

for $j = k+1, \ldots, K$, where e_j and e'_j are the cumulative β spending values st stage j before and after the adjustment, respectively.

After all these adjusted β spending values are computed, the boundary values are then further modified for these adjusted β spending values.

If you specify BETAOVERLAP=NOADJUST, no adjustment is made when overlapping of one-sided β boundaries occurs. The BETAOVERLAP= option is illustrated in Example 77.10.

Specified and Derived Parameters

In the SEQDESIGN procedure, the type of alternative hypothesis (ALT= option) and the condition for early stopping (STOP= option) must be specified for each sequential design. The drift parameters are derived for each design specified. Other parameters, such as Type I error probability α, Type II error probability β, the alternative reference θ_1, and maximum information are either specified or derived in the SEQDESIGN procedure.

Table 77.8 summarizes the available combinations for the specified and derived parameters in the SEQDESIGN procedure.

Table 77.8 Specified and Derived Parameters in the SEQDESIGN Procedure

Specified Parameters				Derived Parameters				
Alt Ref	**Max Info**	**Alpha**	**Beta**	**Alt Ref**	**Max Info**	**Alpha**	**Beta**	**Drift**
Z	X	X					X	X
Z	X		X			X		X
Z	X					X	X	X
Z		X	X		X			X
	X	X	X	X				X
		X	X					X

The symbol "X" indicates that the parameter is either specified or derived in the design and the symbol "Z" indicates that the alternative reference is either specified explicitly with the ALTREF= option or derived from the SAMPLESIZE statement. The drift parameter is always derived in the SEQDESIGN procedure.

For example, if the ALTREF= option is specified without the MAXINFO= option being specified, then the maximum information is derived in the SEQDESIGN procedure with the specified α and β, as illustrated in Example 77.5.

The drift parameter is the standardized reference difference at the final stage. For a design, the drift parameter is

$$d_l = \theta_{1l}\sqrt{I_X}$$

if it has a lower alternative, and

$$d_u = \theta_{1u}\sqrt{I_X}$$

if it has an upper alternative, where I_X is the maximum information and θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively.

If the alternative reference and the maximum information are not both specified, then the specified α and β are used to derive the drift parameter. Then if either the alternative reference or the maximum information is specified, the other is derived from the drift parameter.

If both the alternative reference and the maximum information are specified, then either the α error or the β error is derived in the procedure. However, for a Haybittle-Peto method with the BOUNDARYKEY=NONE option, both α and β errors are derived from the completely specified boundary values.

For a nonsymmetric two-sided design with different lower and upper specifications (such as different lower and upper α errors, β errors, or alternative references in absolute values), the derived lower and upper boundaries are not symmetric. If the alternative references θ_{1l} and θ_{1u} are not both specified, then the SEQDESIGN procedure assumes symmetric alternative references, $\theta_1 = \theta_{1u} = -\theta_{1l}$, for the computation of the boundary values.

Applicable Boundary Keys

In the SEQDESIGN procedure, the BOUNDARYKEY= option in the DESIGN statement specifies the types of errors to be maintained for the design. Table 77.9 lists applicable boundary keys for designs that use unified family and Haybittle-Peto methods, designs that use error spending methods, and designs that use the Haybittle-Peto method only.

Table 77.9 Applicable Boundary Keys for Designs without Whitehead Methods

Method	Specified Parameters (Alt Ref – Max Info)	Boundary Keys Alpha	Beta	None	Both
Unified	X	X	X		
Unified/Haybittle-Peto	X	X	X		
Error spending	X	X	X		
Haybittle-Peto	X	X	X	X	
Unified/Haybittle-Peto					X
Error spending					X
Haybittle-Peto					X

Note that the symbol "X" under "(Alt Ref – Max Info)" indicates that both the alternative reference and maximum information are specified, and the method "Unified/Haybittle-Peto" indicates that both the unified method and the Haybittle-Peto method are used in the same design.

If the ALTREF= and MAXINFO= options are both specified, then Type I and Type II error probability levels cannot be met simultaneously if both error probabilities are specified. The BOUNDARYKEY=ALPHA option maintains the Type I error probability level α and derives Type II error probability β. The BOUNDARYKEY=BETA option maintains the Type II error probability level β and derives Type I error probability α.

If the Haybittle-Peto method is used for all boundaries, the BOUNDARYKEY=NONE option uses the specified α boundary value (STOP=REJECT or STOP=BOTH) and the specified β boundary value (STOP=ACCEPT) at the final stage for the design.

If the ALTREF= and MAXINFO= options are not both specified, the BOUNDARYKEY=BOTH option derives boundary values that maintain both Type I and Type II error probability levels.

Table 77.10 lists applicable boundary keys for a design that uses Whitehead methods.

Table 77.10 Applicable Boundary Keys for Whitehead Methods

Early Stopping	Specified Parameters (Alt Ref – Max Info)	Tau	Boundary Keys Alpha	Beta	None	Both
Reject H_0	X	X	X	X		
Accept H_0	X	X	X	X		
Reject/Accept H_0	X	0.25	X	X		
Reject H_0		X				X
Accept H_0		X				X
Reject/Accept H_0		0.25	X	X	X	

Note that the symbol "X" under "(Alt Ref – Max Info)" indicates that both alternative reference and maximum information are specified.

If the ALTREF= and MAXINFO= options are both specified, then Type I and Type II error probability levels cannot be achieved simultaneously if both are specified. the BOUNDARYKEY=ALPHA option maintains the Type I error probability level α and derives Type II error probability β. The BOUNDARYKEY=BETA option maintains the Type II error probability level β and derives the Type I error probability α.

If the ALTREF= and MAXINFO= options are not both specified, then for a design with the STOP=REJECT or STOP=ACCEPT option, the BOUNDARYKEY=BOTH option derives boundary values that maintain both Type I and Type II error probability levels. If the STOP=BOTH option is specified, Whitehead's triangular method produces boundaries with approximate Type I and Type II error probabilities. The BOUNDARYKEY=NONE option specifies no adjustment to these boundaries. The BOUNDARYKEY=ALPHA and BOUNDARYKEY=BETA options maintain the Type I error probability level α and Type II error probability level β, respectively, by adjusting boundary values at the final stage.

Sample Size Computation

The SEQDESIGN procedure assumes that the data are from a multivariate normal distribution and the sequence of the standardized test statistics $\{Z_1, Z_2, \ldots, Z_K\}$ has the following canonical joint distribution:

- $(Z_1, Z_2, \ldots, Z_K)$ is multivariate normal
- $Z_k \sim N\left(\theta\sqrt{I_k}, 1\right)$
- $\mathrm{Cov}(Z_{k_1}, Z_{k_2}) = \sqrt{I_{k_1}/I_{k_2}} \quad , \quad 1 \le k_1 \le k_2 \le K$

where K is the total number of stages and I_k is the information available at stage k.

If the test statistic is computed from the data that are not from a normal distribution, such as a binomial distribution, then it is assumed that the test statistic is computed from a large sample such that the statistic has an approximately normal distribution.

In a typical clinical trial, the sample size required depends on the Type I error probability level α, alternative reference θ_1, power $1-\beta$, and variance of the response variable. Given a one-sided null hypothesis $H_0 : \theta = 0$ with an upper alternative hypothesis $H_1 : \theta = \theta_1$, the information required for a fixed-sample test is given by

$$I_0 = \frac{(\Phi^{-1}(1-\alpha) + \Phi^{-1}(1-\beta))^2}{\theta_1^2}$$

The parameter θ and the subsequent alternative reference θ_1 depend on the test specified in the clinical trial. For example, suppose you are comparing two binomial populations $p_a = p_b$; then $\theta = p_a - p_b$ is the difference between two proportions if the proportion difference statistic is used,

and $\theta = \log\left(\frac{p_a(1-p_b)}{p_b(1-p_a)}\right)$, the log odds ratio for the two proportions if the log odds ratio statistic is used.

If the maximum likelihood estimate $\hat{\theta}$ from the likelihood function can be derived, then the asymptotic variance for $\hat{\theta}$ is $\text{Var}(\hat{\theta}) = 1/I$, where I is Fisher information for θ. The resulting statistic $\hat{\theta}$ corresponds to the MLE statistic scale as specified in the BOUNDARYSCALE=MLE option in the PROC SEQDESIGN statement, $\hat{\theta}\sqrt{I}$ corresponds to the standardized Z scale (BOUNDARYSCALE=STDZ), and $\hat{\theta}\, I$ corresponds to the score statistic scale (BOUNDARYSCALE=SCORE).

Alternatively, if the score statistic S is derived in a statistical procedure, it can be used as the test statistic and its asymptotic variance is given by Fisher information, I. In this case, $S/\sqrt{I}$ corresponds to the standardized Z scale and S/I corresponds to the MLE statistic scale.

For a group sequential trial, the maximum information I_X is derived in the SEQDESIGN procedure with the specified α, β, and θ_1. With the maximum information

$$I_X = \frac{1}{\text{Var}(\hat{\theta})}$$

the sample size required for a specified test statistic in the trial can be evaluated or estimated from the known or estimated variance of the response variable. Note that different designs might produce different maximum information levels for the same hypothesis, and this in turn might require a different number of observations for the trial.

If each observation in the data set provides one unit of information in a hypothesis testing, such as a one-sample test for the mean, the required sample size for the sequential design can be derived from the maximum information. However, for a survival analysis, an individual in the survival time data might provide only partial information because of censoring. In this case, the required number of events can be derived from the maximum information. With addition accrual information, the sample size can also be computed.

The SEQDESIGN procedure provides sample size computation for some one-sample and two-sample tests in the SAMPLESIZE statement. It also provides sample size computation for tests of a parameter in regression models such as normal regression, logistic regression, and proportional hazards regression. In addition, the procedure can also compute the required sample size or number of events from the corresponding number in the fixed-sample design.

Table 77.11 lists the options available in the SAMPLESIZE statement.

Table 77.11 SAMPLESIZE Statement Options

Option	Description
Fixed-Sample Models	
INPUTNOBS	specifies sample size for fixed-sample design
INPUTNEVENTS	specifies number of events for fixed-sample design
One-Sample Models	
ONESAMPLEMEAN	specifies one-sample Z test for mean
ONESAMPLEFREQ	specifies one-sample test for binomial proportion

Table 77.11 *continued*

Option	Description
Two-Sample Models	
TWOSAMPLEMEAN	specifies two-sample Z test for mean difference
TWOSAMPLEFREQ	specifies two-sample test for binomial proportions
TWOSAMPLESURVIVAL	specifies log-rank test for two survival distributions
Regression Models	
REG	specifies test for a regression parameter
LOGISTIC	specifies test for a logistic regression parameter
PHREG	specifies test for a proportional hazards regression parameter

The MODEL=INPUTNOBS and MODEL=INPUTNEVENTS options are described next, and the remaining options are described in the next three sections.

Input Sample Size for Fixed-Sample Design

The MODEL=INPUTNOBS option derives the sample size required for a group sequential trial from the sample size n_0 for the corresponding fixed-sample design. With the N= n_0 option specifying the sample size n_0 for a fixed-sample design, the sample size required for a group sequential trial is then computed as

$$N_X = \frac{I_X}{I_0}\, n_0$$

where I_X is the maximum information for the group sequential design and I_0 is the information for the corresponding fixed-sample design. The information ratio between I_X and I_0 is derived in the SEQDESIGN procedure.

The SAMPLE=ONE option specifies a one-sample test, and the SAMPLE=TWO option specifies a two-sample test. For a two-sample test, the WEIGHT= option specifies the sample size allocation weights for the two groups.

Input Number of Events for Fixed-Sample Design

The MODEL=INPUTNOBS option derives the number of events required for a group sequential trial from the number of events d_0 for the corresponding fixed-sample design. With the D= d_0 option specifies the number of events d_0 for a fixed-sample survival analysis, the number of events required for a group sequential trial is then computed as

$$d_X = \frac{I_X}{I_0}\, d_0$$

where I_X is the maximum information for the group sequential design and I_0 is the information for the corresponding fixed-sample design. The information ratio between I_X and I_0 is derived in the SEQDESIGN procedure.

The SAMPLE=ONE option specifies a one-sample test, and the SAMPLE=TWO option specifies a two-sample test. For a two-sample test, the WEIGHT= option specifies the sample size allocation weights for the two groups.

With the computed number of events d_X for a group sequential survival design, the required total sample size and sample size at each stage can be derived with specifications of hazard rates, accrual rate, and accrual time.

For a study group, if the hazard rate h is constant, corresponding to an exponential survival distribution, and the individual accrual is uniform in the accrual time T_a with a constant accrual rate r_a, Kim and Tsiatis (1990, pp. 83–84) show that the expected number of events by time t is given by

$$D_h(t) = \begin{cases} r_a \left(t - \frac{1-e^{-ht}}{h} \right) & \text{if } t \le T_a \\ r_a \left(T_a - \frac{e^{-ht}}{h}(e^{hT_a} - 1) \right) & \text{if } t > T_a \end{cases}$$

For a one-sample design, such as a proportional hazards regression, the expected number of events by time t is $E(t) = D_h(t)$, where h is the hazard rate for the group. For a two-sample design, such as a log-rank test for two survival distributions, the expected number of events by time t is

$$E(t) = \frac{R}{R+1} D_{h_a}(t) + \frac{1}{R+1} D_{h_b}(t)$$

where h_a and h_b are hazard rates in groups A and B, respectively, and R is the ratio of the sample size allocation weights w_a/w_b.

If the accrual rate r_a is specified without the accrual time T_a, follow-up time T_f, and total study time $T = T_a + T_f$, the SEQDESIGN procedure computes the minimum and maximum accrual times from the following equation, as described in Kim and Tsiatis (1990, p. 85):

$$\frac{d_X}{r_a} \le T_a \le E^{-1}(d_X)$$

If the accrual rate r_a is specified with one of the three time parameters—the accrual time, follow-up time, and total study time—then the other two time parameters are computed in the SEQDESIGN procedure. Similarly, if the accrual rate r_a is not specified, but two of the three time parameters are specified, then the accrual rate is derived in the SEQDESIGN procedure.

With the accrual rate r_a and the accrual time T_a, the total sample size is

$$N_X = r_a \; T_a$$

At each stage k, the number of events is given by

$$d_k = \frac{I_k}{I_X} \; d_X$$

The corresponding time T_k can be derived from the equation for the expected number of events, $E(t) = d_k$, and the resulting sample size is computed as

$$N_k = r_a \; T_k$$

The following three sections describe examples of test statistics with their resulting information levels, which can then be used to derive the required sample size. The maximum likelihood estimators are used for all tests except to compare two survival distributions with a log-rank test, where a score statistic is used.

Applicable One-Sample Tests and Sample Size Computation

The SEQDESIGN procedure provides sample size computation for two one-sample tests: normal mean and binomial proportion. The required sample size depends on the variance of the response variable—that is, the sample proportion for a binomial proportion test.

In a typical clinical trial, a hypothesis is designed to reject, not accept, the null hypothesis to show the evidence for the alternative hypothesis. Thus, in most cases, the proportion under the alternative hypothesis is used to derive the required sample size. For a test of the binomial proportion, the REF=NULLPROP and REF=PROP options use proportions under the null and alternative hypotheses, respectively.

Test for a Normal Mean

The MODEL=ONESAMPLEMEAN option in the SAMPLESIZE statement derives the sample size required to test a normal mean by using the sample mean statistic for the null hypothesis $\mu = \mu_0$. At stage k, the sample mean is computed as

$$\overline{y}_k = \frac{1}{N_k} \sum_{j=1}^{N_k} y_{kj}$$

where y_{kj} is the value of the jth observation available in the kth stage and N_k is the cumulative sample size at stage k.

An equivalent hypothesis is $H_0 : \theta = 0$, where $\theta = \mu - \mu_0$.

The MLE statistic for θ,

$$\hat{\theta}_k = \overline{y}_k - \mu_0 \sim N\left(\theta,\ {I_k}^{-1}\right)$$

where the information

$$I_k = \frac{1}{\mathrm{Var}(\hat{\theta})} = \frac{1}{\mathrm{Var}(\overline{y}_k)} = \frac{N_k}{\sigma^2}$$

is the inverse of the variance.

That is, the standardized statistic

$$Z_k = \hat{\theta}_k \sqrt{I_k} = (\overline{y}_k - \mu_0)\sqrt{I_k} \sim N\left(\theta\sqrt{I_k},\ 1\right)$$

Thus, to test the hypothesis $H_0 : \theta = 0$ against a two-sided alternative $H_1 : \theta = \theta_1$, H_0 is rejected at stage k if the statistic Z_k is less than or equal to the lower α boundary value or if Z_k is greater than or equal to the upper α boundary value at stage k.

If the variance σ^2 is unknown, the sample variance can be used if it is assumed that the sample variance is computed from a large sample such that the test statistic has an approximately normal distribution.

The maximum information is needed to derive the required sample size. If the maximum information is not specified or derived with the ALTREF= option in the procedure, the MEAN=θ_1 option in the SAMPLESIZE statement is used to specify the alternative reference and thus to derive the maximum information.

In the SEQDESIGN procedure, the computed total sample size

$$N_K = \sigma^2 \; I_X$$

where I_X is the maximum information and σ is the specified standard deviation. With an available maximum information, you can specify the MODEL=ONESAMPLEMEAN(STDDEV= σ) option in the SAMPLESIZE statement to compute the required total sample size and individual sample size at each stage. A procedure such as PROC MEANS can be used to derive a one-sample Z test for a normal mean.

Test for a Binomial Proportion

The MODEL=ONESAMPLEFREQ option in the SAMPLESIZE statement derives the sample size required to test a binomial proportion by using the null hypothesis $p = p_0$, where p is the proportion of a binomial population. At stage k, the MLE for p is computed as

$$\hat{p}_k = \frac{1}{N_k} \sum_{j=1}^{N_k} y_{kj}$$

where y_{kj} is the value of the jth observation available in the kth stage and N_k is the cumulative sample size at stage k.

An equivalent hypothesis is $H_0 : \theta = 0$, where $\theta = p - p_0$. If p_0 is not close to 0 or 1, then for a large sample, $\hat{\theta}_k = \hat{p}_k - p_0$ has an approximately normal distribution

$$\hat{\theta}_k \sim N\left(\theta, \; I_k^{-1}\right)$$

where the information $I_k = (p\,(1-p) \,/\, N_k)^{-1}$ is the inverse of the variance Var($\hat{\theta}$).

Then the standardized statistic

$$Z_k = \hat{\theta}_k \sqrt{I_k} \sim N\left(\theta \sqrt{I_k}, \; 1\right)$$

In practice, the estimated sample proportion $\hat{p}$ at stage k can be used to derive the information I_k and test statistic Z_k. Thus, to test the hypothesis H_0 against an upper alternative $H_1 : \theta = \theta_1 > 0$, H_0 is rejected at stage k if the statistic Z_k is greater than or equal to the upper α boundary at stage k.

The maximum information I_X is needed to derive the required sample size. If the maximum information is not specified or derived with the ALTREF= option in the procedure, the PROP= option in the SAMPLESIZE statement is used to specify the alternative reference and to derive the maximum information for the sample size calculation.

It is assumed that the sample size is sufficiently large such that the test statistic has an approximately normal distribution. With the hypotheses $H_0 : p = p_0$ and $H_1 : p = p_1$, the SEQDESIGN procedure derives the total sample size

$$N_X = p^* (1 - p^*) I_X$$

where $p^* = p_0$ if REF=NULLPROP is specified. Otherwise, $p^* = p_1$.

If the PROP= option in the SAMPLESIZE statement is not specified, then the alternative reference θ_1 derived in the SEQDESIGN procedure is used to compute $p_1 = p_0 + \theta_1$.

The ALTREF= option in the PROC statement can be used to specify θ_1. Otherwise, the PROP= option in the SAMPLESIZE statement must be specified.

For example, with $H_0 : p = 0.5$, $H_1 : p = 0.6$ and the default REF=PROP,

$$N_K = p^*(1 - p^*) I_X = (0.6 \times 0.4) I_X = 0.24 I_X$$

You can specify the MODEL=ONESAMPLEFREQ option in the SAMPLESIZE statement to compute the required total sample size and individual sample size at each stage. A procedure such as PROC GENMOD with the default DIST=NORMAL option in the MODEL statement can be used to derive the Z test for a binomial proportion.

Applicable Two-Sample Tests and Sample Size Computation

The SEQDESIGN procedure provides sample size computation for two-sample tests: the test for the difference between two normal means, tests for binomial proportions, and the log-rank test for two survival distributions. These tests for binomial proportions include the test for the difference between two binomial proportions, the log odds ratio test for binomial proportions, and the log relative risk test for binomial proportions,

For a test of difference between two sample means, the required sample size depends on the assumed sample variances. Similarly, for a test of two-sample proportions, the required sample size depends on the assumed sample proportions. For a log-rank test of two survival distributions, the required sample size depends on the assumed sample hazard rates, accrual rate, and accrual time.

If the REF=NULLPROP or REF=NULLHAZARD option is specified, the proportions or hazard rates under the null hypothesis are used to derive the required sample size or number of events. Otherwise, the default REF=PROP option in the TWOSAMPLEFREQ option or the REF=HAZARD option in the TWOSAMPLESURVIVAL option uses proportions or hazard rates under the alternative hypothesis to derive the required sample size or number of events.

Test for the Difference between Two Normal Means

The MODEL=TWOSAMPLEMEAN option in the SAMPLESIZE statement derives the sample size required to test the difference between the means of two normal populations μ_a and μ_b by using the null hypothesis $H_0 : \theta = 0$, where $\theta = \mu_a - \mu_b$.

At stage k, the MLE for θ is computed as

$$\hat{\theta}_k = \overline{y}_{ak} - \overline{y}_{bk} = \frac{1}{N_{ak}} \sum_{j=1}^{N_{ak}} y_{akj} - \frac{1}{N_{bk}} \sum_{j=1}^{N_{bk}} y_{bkj}$$

where y_{akj} and y_{bkj} are the values of the jth observation available in the kth stage groups A and B, respectively, and N_{ak} and N_{bk} are the cumulative sample sizes at stage k for these two groups.

The statistic $\hat{\theta}_k$ has a normal distribution

$$\hat{\theta}_k \sim N\left(\theta,\ {I_k}^{-1}\right)$$

where the information I_k is the inverse of the variance $\text{Var}(\hat{\theta}_k) = \sigma_a^2/N_{ak} + \sigma_b^2/N_{bk}$.

Then the standardized statistic

$$Z_k = \hat{\theta}_k \sqrt{I_k} \sim N\left(\theta \sqrt{I_k},\ 1\right)$$

Thus, to test the hypothesis $H_0 : \theta = 0$ against an upper alternative $H_1 : \theta = \theta_1, \theta_1 > 0$, H_0 is rejected at stage k if the statistic $Z_k \geq a_k$, the upper α boundary for the standardized Z statistic at stage k.

If the variances σ_a^2 and σ_b^2 are unknown, the sample variances can be used to derive the information I_k if it is assumed that each sample variance is computed from a large sample such that the test statistic has an approximately normal distribution.

The maximum information is needed to derive the required sample size. If the maximum information is not specified or derived in the procedure, the alternative reference θ_1^* specified in the MEANDIFF option is used to derive the maximum information.

Note that in order to derive the sample sizes N_{ak} and N_{bk} uniquely from the information, $N_{ak} = R\, N_{bk}$ is assumed for $k = 1, 2, \ldots, K$, where $R = w_a/w_b$ is the constant allocation ratio computed from the WEIGHT=w_a w_b option in the SAMPLESIZE statement.

In PROC SEQDESIGN, the computed total sample sizes for the two groups are

$$N_{aK} = (\sigma_a^2 + R\,\sigma_b^2)\, I_X = R\,(\frac{\sigma_a^2}{R} + \sigma_b^2)\, I_X$$

$$N_{bK} = (\frac{\sigma_a^2}{R} + \sigma_b^2)\, I_X$$

where I_X is the maximum information derived in the SEQDESIGN procedure, R is the constant allocation ratio, and σ_a and σ_b are the specified standard deviations.

For $R = 1$, the two sample sizes are equal, then

$$N_{aK} = N_{bK} = \frac{N_K}{2} = (\sigma_a^2 + \sigma_b^2)\, I_X$$

If the variances from the two groups are equal, $\sigma_a^2 = \sigma_b^2 = \sigma^2$, then the total sample sizes for the two groups are

$$N_{aK} = (1 + R)\,\sigma^2\, I_X$$

$$N_{bK} = (1 + \frac{1}{R})\,\sigma^2\, I_X$$

and the total sample size is

$$N_X = N_{aK} + N_{bK} = \frac{(R+1)^2}{R}\,\sigma^2\, I_X$$

Furthermore, for $R = 1$, the two sample sizes are equal, then

$$N_{aK} = N_{bK} = \frac{N_X}{2} = 2\,\sigma^2\, I_X$$

With an available maximum information, you can specify the MODEL=TWOSAMPLEMEAN(WEIGHT= R STDDEV= σ_a σ_b) option in the SAMPLESIZE statement to compute the required total sample size and individual sample size at each stage. A procedure such as PROC GLM can be used to derive the two-sample Z test for the mean difference.

Test for the Difference between Two Binomial Proportions

The MODEL=TWOSAMPLEFREQ(TEST=PROP) option in the SAMPLESIZE statement derives the sample size required to test the difference between two binomial populations with $H_0 : \theta = 0$, where $\theta = p_a - p_b$. At stage k, the MLE for θ is

$$\hat{\theta}_k = \hat{p}_{ak} - \hat{p}_{bk} = \frac{1}{N_{ak}} \sum_{j=1}^{N_{ak}} y_{akj} - \frac{1}{N_{bk}} \sum_{j=1}^{N_{bk}} y_{bkj}$$

where y_{akj} and y_{bkj} are the values of the jth observation available in the kth stage for groups A and B, respectively, and N_{ak} and N_{bk} are the cumulative sample sizes at stage k for these two groups.

For sufficiently large sample sizes N_{ak} and N_{bk}, the statistic $\hat{\theta}_k$ has an approximate normal distribution

$$\hat{\theta}_k \sim N\left(\theta,\, {I_k}^{-1}\right)$$

where the information is the inverse of the variance

$$\mathrm{Var}(\hat{\theta}_k) = \frac{p_a\,(1-p_a)}{N_{ak}} + \frac{p_b\,(1-p_b)}{N_{bk}}$$

Thus, the standardized statistic

$$Z_k = \hat{\theta}_k \sqrt{I_k} \sim N\left(\theta\sqrt{I_k},\, 1\right)$$

In practice, $p_a = \hat{p}_a$ and $p_b = \hat{p}_b$, the estimated sample proportions for groups A and B, respectively, at stage k, can be used to derive the information I_k and the test statistic Z_k. Thus, to test the hypothesis H_0 against an upper alternative $H_1 : \theta > 0$, H_0 is rejected at stage k if the statistic $Z_k \geq a_k$, the upper α boundary for the standardized Z statistic at stage k.

The maximum information I_X is needed to derive the required sample size. If the maximum information is not specified or derived with the ALTREF= option in the procedure, the PROP= option in the SAMPLESIZE statement is used to provide proportions under the alternative hypothesis for the alternative reference and then to derive the maximum information.

The proportions in the two groups are needed to derive the sample size. Also, in order to derive the sample sizes N_{ak} and N_{bk} uniquely from the information, $N_{ak} = R\, N_{bk}$ is assumed for $k = 1, 2, \ldots, K$, where $R = w_a/w_b$ is the constant allocation ratio computed from the WEIGHT=w_a w_b option in the SAMPLESIZE statement. Then

$$I_X = \left(\frac{p_a\,(1-p_a)}{N_{aK}} + \frac{p_b\,(1-p_b)}{N_{bK}} \right)^{-1} = \frac{N_{aK}}{p_a(1-p_a) + R\,p_b(1-p_b)}$$

In PROC SEQDESIGN, the total sample sizes in the two groups are computed as

$$N_{aK} = \left(p_a^*\,(1-p_a^*) + R\,p_b^*\,(1-p_b^*) \right)\, I_X$$

$$N_{bK} = \frac{1}{R}\, N_{aK}$$

where $R = w_a/w_b$ is the constant allocation ratio, and p_a^* and p_b^* are proportions specified with the REF= option:

- REF=NULLPROP uses proportions under H_0: $p_a^* = p_{0a}$, $p_b^* = p_{0b}$
- REF=AVGNULLPROP uses the average proportion under H_0: $p_a^* = p_b^* = (Rp_{0a} + p_{0b})/(R+1)$
- REF=PROP uses proportions under H_1: $p_a^* = p_{1a}$, $p_b^* = p_{1b}$
- REF=AVGPROP uses the average proportion under $H_1 : p_a^* = p_b^* = (Rp_{1a} + p_{1b})/(R+1)$

The total sample size is given by

$$N_X = N_{aK} + N_{bK} = (R+1)\left(\frac{1}{R}\, p_a^*\,(1-p_a^*) + p_b^*\,(1-p_b^*) \right)\, I_X$$

For $R = 1$, the two sample sizes are equal,

$$N_{aK} = N_{bK} = \frac{N_X}{2} = \left(p_a^*\,(1-p_a^*) + p_b^*\,(1-p_b^*) \right)\, I_X$$

You can specify the MODEL=TWOSAMPLEFREQ(TEST=PROP WEIGHT=R) option in the SAMPLESIZE statement to compute the required total sample size and individual sample size at each stage. A procedure such as PROC GENMOD with the default DIST=NORMAL option in the MODEL statement can be used to derive the two-sample Z test for proportion difference.

Test for Two Binomial Proportions with a Log Odds Ratio Statistic

The MODEL=TWOSAMPLEFREQ(TEST=LOGOR) option in the SAMPLESIZE statement derives the sample size required to test two binomial proportions by using a log odds ratio statistic. The odds ratio is the ratio of the odds in one group to the odds in the other group, and the log odds ratio is the logarithm of the odds ratio

$$\theta = \log\left(\frac{p_a/(1-p_a)}{p_b/(1-p_b)}\right) = \log\left(\frac{p_a(1-p_b)}{p_b(1-p_a)}\right)$$

The hypothesis of no difference between two proportions, $p_a = p_b$, can be tested through the null hypothesis $H_0 : \theta = 0$, where θ is the log odds ratio. For example, with $H_0 : p_a = p_b = 0.6$ and $H_1 : p_a = 0.8$, $p_b = 0.6$, it corresponds to the equivalent hypothesis $H_0 : \theta = 0$ and $H_1 : \theta = \log\left(\frac{0.8(1-0.6)}{0.6(1-0.8)}\right) = \log(8/3) = 0.98083$.

The maximum likelihood estimate of θ is given by

$$\hat{\theta} = \log\left(\frac{\hat{p}_a(1-\hat{p}_b)}{\hat{p}_b(1-\hat{p}_a)}\right)$$

with an asymptotic variance

$$\mathrm{Var}(\hat{\theta}) = I^{-1} = \frac{1}{N_a\, p_a(1-p_a)} + \frac{1}{N_b\, p_b(1-p_b)}$$

where I is the information (Diggle et al. 2002, pp. 341–342). That is, the standardized statistic

$$Z_k = \hat{\theta}_k \sqrt{I_k} \sim N\left(\theta\sqrt{I_k},\, 1\right)$$

In practice, $p_a = \hat{p}_a$ and $p_b = \hat{p}_b$, the estimated sample proportions for groups A and B, respectively, at stage k, can be used to derive the information I_k and the test statistic $Z_k = \hat{\theta}_k\sqrt{I_k}$ if the two sample sizes N_a and N_b are sufficiently large such that the test statistic has an approximately normal distribution.

The maximum information I_X is needed to derive the required sample size. If the maximum information is not specified or derived with the ALTREF= option in the procedure, the PROP= option in the SAMPLESIZE statement is used to provide proportions under the alternative hypothesis for the alternative reference and then to derive the maximum information.

In order to derive the sample sizes N_{ak} and N_{bk} uniquely from the information, $N_{ak} = R\, N_{bk}$ is assumed for $k = 1, 2, \ldots, K$, where $R = w_a/w_b$ is the constant allocation ratio computed from the WEIGHT=w_a w_b option in the SAMPLESIZE statement. Then with

$$I_X = N_{bK}\left(\frac{1}{R\, p_a(1-p_a)} + \frac{1}{p_b(1-p_b)}\right)^{-1}$$

the sample size can be computed.

In PROC SEQDESIGN, the total sample sizes in the two groups are computed as

$$N_{bK} = I_X \left(\frac{1}{R\, p_a^*(1-p_a^*)} + \frac{1}{p_b^*(1-p_b^*)} \right)$$

$$N_{aK} = R\, N_{bK}$$

where $R = w_a/w_b$ is the constant allocation ratio, and p_a^* and p_b^* are proportions specified with the REF= option:

- REF=NULLPROP uses proportions under H_0: $p_a^* = p_{0a}$, $p_b^* = p_{0b}$
- REF=AVGNULLPROP uses the average proportion under H_0: $p_a^* = p_b^* = (Rp_{0a} + p_{0b})/(R+1)$
- REF=PROP uses proportions under H_1: $p_a^* = p_{1a}$, $p_b^* = p_{1b}$
- REF=AVGPROP uses the average proportion under H_1 : $p_a^* = p_b^* = (Rp_{1a} + p_{1b})/(R+1)$

You can specify the MODEL=TWOSAMPLEFREQ(TEST=LOGOR WEIGHT=R) option in the SAMPLESIZE statement to compute the required total sample size and individual sample size at each stage. A procedure such as PROC LOGISTIC can be used to derive the log odds ratio statistic.

Test for Two Binomial Proportions with a Log Relative Risk Statistic

The MODEL=TWOSAMPLEFREQ(TEST=LOGRR) option in the SAMPLESIZE statement derives the sample size required to test two binomial proportions by using a log relative risk statistic. The relative risk is the ratio of the proportion in one group to the proportion in the other group. The log relative risk statistic is the logarithm of the relative risk

$$\theta = \log\left(\frac{p_a}{p_b}\right)$$

The hypothesis of no difference between two proportions, $p_a = p_b$, can be tested through the null hypothesis $H_0 : \theta = 0$. For example, with $H_0 : p_a = p_b = 0.6$ and $H_1 : p_a = 0.8$, $p_b = 0.6$, it corresponds to the equivalent hypothesis $H_0 : \theta = 0$ and $H_1 : \theta = \log\left(\frac{0.8}{0.6}\right) = \log(4/3) = 0.28768$.

The maximum likelihood estimate of θ is given by

$$\hat{\theta} = \log\left(\frac{\hat{p}_a}{\hat{p}_b}\right)$$

with an asymptotic variance

$$I^{-1} = \frac{1-p_a}{N_a\, p_a} + \frac{1-p_b}{N_b\, p_b}$$

where I is the information (Chow and Liu 1998, p. 329).

In practice, $p_a = \hat{p}_a$ and $p_b = \hat{p}_b$, the estimated sample proportions for groups A and B, respectively, at stage k, are used to derive the information I_k and the test statistic $Z_k = \hat{\theta}_k \sqrt{I_k}$.

The maximum information I_X and proportions p_a and p_b are needed to derive the required sample size. If the maximum information is not specified or derived with the ALTREF= option in the procedure, the PROP= option in the SAMPLESIZE statement is used to provide proportions under the alternative hypothesis for the alternative reference and then to derive the maximum information.

Note that in order to derive the sample sizes N_{ak} and N_{bk} uniquely from the information, $N_{ak} = R\, N_{bk}$ is assumed for $k = 1, 2, \ldots, K$, where $R = w_a/w_b$ is the constant allocation ratio computed from the WEIGHT=w_a w_b option in the SAMPLESIZE statement. Then the sample size can be computed from

$$I_X = N_{bK} \left(\frac{1 - p_a}{R\, p_a} + \frac{1 - p_b}{p_b} \right)^{-1}$$

In PROC SEQDESIGN, the computed sample sizes in the two groups are

$$N_{bK} = I_X \left(\frac{1 - p_a^*}{R\, p_a^*} + \frac{1 - p_b^*}{p_b^*} \right)$$

$$N_{aK} = R\, N_{bK}$$

where $R = w_a/w_b$ is the constant allocation ratio, and p_a^* and p_b^* are proportions specified with the REF= option:

- REF=NULLPROP uses proportions under H_0: $p_a^* = p_{0a}$, $p_b^* = p_{0b}$
- REF=AVGNULLPROP uses the average proportion under H_0: $p_a^* = p_b^* = (Rp_{0a} + p_{0b})/(R+1)$
- REF=PROP uses proportions under H_1: $p_a^* = p_{1a}$, $p_b^* = p_{1b}$
- REF=AVGPROP uses the average proportion under H_1 : $p_a^* = p_b^* = (Rp_{1a} + p_{1b})/(R+1)$

You can specify the MODEL=TWOSAMPLEFREQ(TEST=LOGRR WEIGHT=R) option in the SAMPLESIZE statement to compute the required total sample size and individual sample size at each stage. A procedure such as PROC LOGISTIC can be used to derive the log relative risk statistic.

Test for Two Survival Distributions with a Log-Rank Test

The MODEL=TWOSAMPLESURV option in the SAMPLESIZE statement derives the number of events required for a log-rank test of two survival distributions. The analysis of survival data involves the survival times for both censored and uncensored data. A noncensored survival time is the time from treatment to an event such as remission or relapse for an individual. A censored survival time is the time from treatment to the time of analysis for an individual surviving at that time, and the status is unknown beyond that time.

Let T be the random variable of the survival time. Then the survival function

$$S(t) = \Pr(T > t)$$

is the probability that an individual from the population has a survival time that exceeds t. And the hazard function is given by

$$h(t) = \frac{f(t)}{S(t)}$$

where $f(t)$ is the density function of T.

The hazard functions can be used to test the equality of two survival distributions $S_a(t) = S_b(t)$ with the null hypothesis $H_0 : h_a(t) = h_b(t), t > 0$, where $S_a(t)$ and $S_b(t)$ are survival functions for groups A and B, respectively, and $h_a(t)$ and $h_b(t)$ are the corresponding hazard functions.

If the two hazards are proportional, $h_a(t) = \lambda\, h_b(t)$, where λ is a constant, then an equivalent null hypothesis is

$$H_0 : \lambda = \frac{h_a(t)}{h_b(t)} = 1$$

Alternatively, another equivalent null hypothesis is given by

$$H_0 : \theta = -\log(\lambda) = 0$$

Suppose that the hazard rate h is a constant. Then with a specified median survival time T_m, the hazard rate can be derived from the equation

$$e^{-h\,T_m} = \frac{1}{2}$$

Denote the distinct event times at stage k as $\tau_{kj}, j = 1, 2, \ldots, t_k$, where t_k is the total number of distinct event times. Then the score statistic is the log-rank statistic (Jennison and Turnbull 2000, pp. 259–261; Whitehead 1997, pp. 36–39)

$$S_k = \sum_{j=1}^{t_k} (d_{akj} - e_{akj})$$

where d_{akj} is the number of events from group A and e_{akj} is the number of expected events from A. The number of expected events from A is computed as

$$e_{akj} = d_{kj} \frac{r_{akj}}{r_{kj}}$$

where d_{kj} is the number of events from both groups, r_{akj} is the number of individuals from the treatment group who survived up to time τ_{kj}, and r_{kj} is the number of individuals from both groups who survived up to time τ_{kj}.

If the number of events d_{kj} is small relative to r_{kj}, the number of individuals survived up to time τ_{kj}, then with a sufficiently large sample size, S_k has an approximately normal distribution

$$S_k \sim N(\theta\, I_k,\ I_k)$$

where the variance of S_k is the estimated information

$$I_k = \sum_{j=1}^{t_k} \frac{r_{akj}\, r_{bkj}\, d_{kj}}{r_{kj}^2}$$

In order to derive the number of events from the information I_k, $N_{ak} = R\, N_{bk}$ is assumed for $k = 1, 2, \ldots, K$, where $R = w_a/w_b$ is the constant allocation ratio computed from the WEIGHT=w_a w_b option in the SAMPLESIZE statement.

The maximum information I_X is needed to derive the required sample size. If the maximum information is specified or derived with the ALTREF= option in the procedure, the HAZARD=, MEDSURVTIME=, and HAZARDRATIO= options are not applicable. Otherwise, the HAZARD=, MEDSURVTIME=, or HAZARDRATIO= option is used to compute the alternative reference and then to derive the maximum information for the sample size calculation.

With $N_{aK} = R\, N_{bK}$, if the number of events is few relative to the number of individuals who survived, then $r_{aKj} \approx R\, r_{bKj}$, and

$$I_X \approx \sum_{j=1}^{t_K} \frac{R}{(R+1)^2} d_{Kj} = \frac{R}{(R+1)^2}\, D_X$$

where D_X is the total number of events.

Thus, the required total number of events

$$D_X = \frac{(R+1)^2}{R}\, I_X$$

For a study group, if the hazard rate is constant, corresponding to an exponential survival distribution, and the individual accrual is uniform in the accrual time T_a with a constant accrual rate r_a, then the required total sample size and sample size at each stage can be derived. See the section "Input Number of Events for Fixed-Sample Design" on page 5856 for a detailed description of the sample size computation that uses hazard rates, accrual rate, and accrual time.

You can specify the MODEL=TWOSAMPLESURVIVAL option in the SAMPLESIZE statement to compute the required total number of events and individual number of events at each stage. With the specifications of hazard rates, accrual rate, and accrual time, the required total sample size and individual sample size at each stage can also be derived. If the REF=NULLHAZARD option is specified, the hazard rates under the null hypothesis, h_{0a} and h_{0b}, are used in the sample size computation. Otherwise, the hazard rates under the alternative hypothesis, h_{1a} and h_{1b}, are used. A procedure such as PROC LIFETEST can be used to derive the log-rank statistic.

Applicable Regression Parameter Tests and Sample Size Computation

The SEQDESIGN procedure provides sample size computation for tests of a regression parameter in three regression models: normal regression, logistic regression, and proportional hazards regression.

To test a parameter β_1 in a regression model, the variance of the parameter estimate $\hat{\beta}_1$ is needed for the sample size computation. In a simple regression model with one covariate X1, the variance of $\hat{\beta}_1$ is inversely related to the variance of X1, σ_x^2. That is,

$$\text{Var}(\hat{\beta}_1) \propto \frac{1}{N\,\sigma_x^2}$$

for the normal regression and logistic regression models, where N is the sample size, and

$$\text{Var}(\hat{\beta}_1) \propto \frac{1}{D\,\sigma_x^2}$$

for the proportional hazards regression model, where D is the number of events.

For a regression model with more than one covariate, the variance of $\hat{\beta}_1$ for the normal regression and logistic regression models is inversely related to the variance of X1 after adjusting for other covariates. That is,

$$\text{Var}(\hat{\beta}_1) \propto \frac{1}{N\,(1-r_x^2)\,\sigma_x^2}$$

where $\hat{\beta}_1$ is the estimate of the parameter β_1 in the model and r_x^2 is the R square from the regression of X1 on other covariates—that is, the proportion of the variance σ_x^2 explained by these covariates.

Similarly, for a proportional hazards regression model,

$$\text{Var}(\hat{\beta}_1) \propto \frac{1}{D\,(1-r_x^2)\,\sigma_x^2}$$

Thus, with the derived maximum information, the required sample size or number of events can also be computed for the testing of a parameter in a regression model with covariates.

Test for a Parameter in the Regression Model

The MODEL=REG option in the SAMPLESIZE statement derives the sample size required for a Z test of a normal regression. For a normal linear regression model, the response variable is normally distributed with the mean equal to a linear function of the explanatory variables and the constant variance σ^2.

The normal linear model is

$$\mathbf{y} \sim N\left(\mathbf{X}\boldsymbol{\beta},\, \sigma_y^2\,\mathbf{I}_{(N)}\right)$$

where $\mathbf{Y}_{(N\times 1)}$ is the vector of the N observed responses, $\mathbf{X}_{(N\times p)}$ is the design matrix for these N observations, $\boldsymbol{\beta}_{(p\times 1)}$ is the parameter vector, and $\mathbf{I}_{(N)}$ is the $(N \times N)$ identity matrix.

The least squares estimate is

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}'\mathbf{X}\right)^{-1}\mathbf{X}'\mathbf{Y}$$

and is normally distributed with mean $\boldsymbol{\beta}$ and variance

$$\text{Var}(\hat{\boldsymbol{\beta}}) = \sigma_y^2(\mathbf{X}'\mathbf{X})^{-1}$$

For a model with only one covariate X1,

$$\hat{\beta}_1 \sim N\left(\beta_1,\ \text{Var}(\hat{\beta}_1)\right)$$

where the variance

$$\text{Var}(\hat{\beta}_1) = I_{\beta_1}^{-1} = \sigma_y^2\ \frac{1}{N\,\sigma_x^2}$$

Thus, with the derived maximum information $I_X = I_{\beta_1}$, the required sample size is given by

$$N = I_X\ \frac{\sigma_y^2}{\sigma_x^2}$$

For a normal linear model with more than one covariate, the variance of a single parameter β_1 is

$$\text{Var}(\hat{\beta}_1) = \sigma_y^2\ (\mathbf{X}'\mathbf{X})^{-1}_{(11)} = \sigma_y^2\ \frac{1}{N\,\sigma_x^2\,(1-r_x^2)}$$

where $(\mathbf{X}'\mathbf{X})^{-1}_{(11)}$ is the diagonal element of the $(\mathbf{X}'\mathbf{X})^{-1}$ matrix corresponding to the parameter β_1, σ_x^2 is the variance of the variable X1, and r_x^2 is the proportion of variance of X1 explained by other covariates. The value $\sigma_x^2\,(1-r_x^2)$ represents the variance of X1 after adjusting for all other covariates.

Thus, with the derived maximum information I_X, the required sample size is

$$N = I_X\ \frac{\sigma_y^2}{(1-r_x^2)\,\sigma_x^2}$$

In the SEQDESIGN procedure, you can specify the MODEL=REG(VARIANCE=σ_y^2 XVARIANCE=σ_x^2 XRSQUARE=r_x^2) option in the SAMPLESIZE statement to compute the required total sample size and individual sample size at each stage. A SAS procedure such as PROC REG can be used to compute the parameter estimate and its standard error at each stage.

Test for a Parameter in the Logistic Regression Model

The MODEL=LOGISTIC option in the SAMPLESIZE statement derives the sample size required for a Z test of a logistic regression parameter. The linear logistic model has the form

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \mathbf{x}\boldsymbol{\beta}$$

where p is the response probability to be modeled and $\boldsymbol{\beta}$ is a vector of parameters.

Following the derivation in the section "Test for a Parameter in the Regression Model" on page 5869, the required sample size for testing a parameter in $\boldsymbol{\beta}$ is given by

$$N = I_X\ \frac{\sigma_y^2}{(1-r_x^2)\,\sigma_x^2}$$

With the variance of the logit response, $\sigma_y^2 = 1/(p(1-p))$,

$$N = I_X \frac{1}{p(1-p)} \frac{1}{(1-r_x^2)\,\sigma_x^2}$$

where σ_x^2 is the variance of X and r_x^2 is the proportion of variance explained by other covariates.

In the SEQDESIGN procedure, you can specify the MODEL=LOGISTIC(PROP=p XVARIANCE=σ_x^2 XRSQUARE=r_x^2) option in the SAMPLESIZE statement to compute the required total sample size and individual sample size at each stage.

A SAS procedure such as PROC LOGISTIC can be used to compute the parameter estimate and its standard error at each stage.

Test for a Parameter in the Proportional Hazards Regression Model

The MODEL=PHREG option in the SAMPLESIZE statement derives the number of events required for a Z test of a proportional hazards regression parameter. For analyses of survival data, Cox's semiparametric model is often used to examine the effect of explanatory variables on hazard rates. The survival time of each observation in the population is assumed to follow its own hazard function, $h_i(t)$, expressed as

$$h_i(t) = h(t; \mathbf{X}_i) = h_0(t)\exp(\mathbf{X}_i'\boldsymbol{\beta})$$

where $h_0(t)$ is an arbitrary and unspecified baseline hazard function, $\mathbf{x}_i$ is the vector of explanatory variables for the ith individual, and $\boldsymbol{\beta}$ is the vector of regression parameters associated with the explanatory variables.

Hsieh and Lavori (2000, p. 553) show that the required number of events for testing a parameter in $\boldsymbol{\beta}$, β_1, associated with the variable X1 is given by

$$D_X = I_X \frac{1}{(1-r_x^2)\,\sigma_x^2}$$

where σ_x^2 is the variance of X1 and r_x^2 is the proportion of variance of X1 explained by other covariates.

In the SEQDESIGN procedure, you can specify the MODEL=PHREG(XVARIANCE=σ_x^2 XRSQUARE=r_x^2) option in the SAMPLESIZE statement to compute the required number of events and individual number of events at each stage.

A SAS procedure such as PROC PHREG can be used to compute the parameter estimate and its standard error at each stage.

Note that for a two-sample test, X1 is an indicator variable and is the only covariate in the model. Thus, if the two sample sizes are equal, then the variance $\sigma_x^2 = 1/4$ and the required number of events for testing the parameter β_1 is given by

$$D_X = I_X \frac{1}{\sigma_x^2} = 4\,I_X$$

See the section "Input Number of Events for Fixed-Sample Design" on page 5856 for a detailed description of the sample size computation that uses hazard rates, accrual rate, and accrual time.

Aspects of Group Sequential Designs

This section summarizes various aspects of group sequential designs that are encountered in applications of the SEQDESIGN procedure. Features are illustrated through two-sided designs with $\alpha = 0.05$ and $\beta = 0.10$. The null hypothesis $H_0 : \theta = 0$ and an alternative reference $\theta_1 = \pm 0.25$ are used for the designs with early stopping only to reject the null hypothesis.

Canonical Joint Distribution

The SEQDESIGN procedure assumes that with a total number of stages K, the sequence of the standardized test statistics $\{Z_1, Z_2, \ldots, Z_K\}$ has the canonical joint distribution with information levels $\{I_1, I_2, \ldots, I_K\}$ for the parameter θ (Jennison and Turnbull 2000, p. 49):

- $(Z_1, Z_2, \ldots, Z_K)$ is multivariate normal
- $Z_k \sim N\left(\theta\sqrt{I_k},\ 1\right),\ k = 1, 2, \ldots, K$
- $\mathrm{Cov}(Z_{k_1}, Z_{k_2}) = \sqrt{(I_{k_1}/I_{k_2})},\quad 1 \le k_1 \le k_2 \le K$

Normality Assumption

The SEQDESIGN procedure derives the boundary values by assuming that the sequence of the standardized test statistics $\{Z_1, Z_2, \ldots, Z_K\}$ has the canonical joint distribution with information levels $\{I_1, I_2, \ldots, I_K\}$ for the parameter θ. If the test statistic Z_k does not have a normal distribution, it is assumed that the test statistic is computed from a large sample, so that the resulting statistic has an approximately normal distribution.

Number of Stages

For group sequential trials with fixed significance level α, power $1 - \beta$, and alternative reference, if the number of stages is increased, the required maximum information is also increased, but the average sample number under the alternative hypothesis is likely to decrease.

For example, with O'Brien-Fleming designs specified in the section, the maximum information increases from 168.12 for a fixed-sample design to 169.32 for a two-stage design, 172.57 for a five-stage design, and then 174.42 for a ten-stage design. In the mean time, the average sample number (as a percentage of fixed-sample) under the alternative hypothesis decreases from 100 for a fixed-sample design to 85.11 for a two-stage design, 75.03 for a five-stage design, and then 71.80 for a ten-stage design. The reduction in average sample number decreases as the number of stages increases. Thus there seems to be little to gain from choosing a design with more than five stages (Pocock 1982, p. 155).

Alternative Reference

The alternative reference θ_1 is the hypothetical reference under the alternative hypothesis at which the power is computed. It is a treatment value that the investigators would hope to detect with high probability (Jennison and Turnbull 2000, p. 21).

For a group sequential design with specified parameters such as α and β errors, the drift parameter $\theta_1\sqrt{I_X}$ is always derived in the SEQDESIGN procedure. Thus, with a smaller alternative reference θ_1, a larger maximum information level I_X is needed. That is, in order to detect a smaller difference with the same high power, a larger sample size is required.

Maximum Information

In a clinical trial, the amount of information about an unknown parameter available from the data can be measured by the Fisher information, the variance of the score statistic. The maximum information is the information level needed at the final stage of the group sequential trial if the trial does not stop at an interim stage. For a group sequential design, the maximum information can be derived with the specified alternative reference.

The maximum information is proportional to the sample size or number of events required for the design. Thus, it can also be used to compare different designs. Generally, a design with a larger probability to stop the trial early tends to have a larger maximum information. For example, with four-stage designs specified in the section, the Pocock method has a maximum information of 198.91 and the O'Brien-Fleming method has a maximum information of 171.85, indicating a much larger information level required for the Pocock method.

Drift Parameter

The drift parameter $\theta_1\sqrt{I_X}$ is derived for each design in the SEQDESIGN procedure, where θ_1 is the alternative reference. It is proportional to the square root of maximum information required for the design and can be used to compare maximum information for different designs with the same alternative reference. For example, with the four-stage designs specified in the section, the Pocock method has a drift parameter 3.526 and the O'Brien-Fleming method has a drift parameter 3.277, indicating that a larger maximum information level is required for the Pocock method than for the O'Brien-Fleming method.

Average Sample Number

The average sample number is the expected sample size (for nonsurvival data) or expected number of events (for survival data) of the design under a specific hypothetical reference. The percent average sample numbers with respect to the corresponding fixed-sample design are displayed in the SEQDESIGN procedure.

The design that requires a larger maximum information level tends to have a smaller average sample number under the alternative hypothesis. For example, with the four-stage designs specified in the section, the Pocock design has a maximum information of 198.91 and an average sample number

(in percentage of fixed-sample design) of 69.748 under the alternative hypothesis, and the O'Brien-Fleming design has a maximum information of 171.85 and an average sample number of 76.740.

Sample Size

The maximum information for the sequential design expressed as a percentage of its corresponding fixed-sample information is derived in the SEQDESIGN procedure. The sample size or number of events needed for a group sequential trial is computed by multiplying the sample size or number of events for the corresponding fixed-sample design by the derived percentage.

If the sample size or number of events for the fixed-sample design is available, you can use the MODEL=INPUTNOBS or MODEL=INPUTNEVENTS option in the SAMPLESIZE statement to derive the sample size or number of events needed at each stage. Otherwise, with the specified or derived maximum information, you can use the MODEL= option in the SAMPLESIZE statement to specify a hypothesis test and then to derive the sample size or number of events needed at each stage. See the section "Sample Size Computation" on page 5854 for the sample size computation for commonly used tests.

Summary of Methods in Group Sequential Designs

There are three different types of methods available in the SEQDESIGN procedure: fixed boundary shape methods for specified boundary shape, Whitehead methods for boundaries from continuous monitoring, and error spending methods for specified error spending at each stage.

The fixed boundary shape methods include unified family methods and Haybittle-Peto methods. The unified family methods include Pocock, O'Brien-Fleming, power family, and triangular methods.

Pocock Method

Pocock derives the constant boundary on the standardized Z scale to demonstrate the sequential design while maintaining the overall α and β levels (Pocock 1977). The resulting boundary tends to stop the trials early with a larger p-value. This boundary is commonly called a Pocock boundary, but Pocock himself does not advocate these boundary values for stopping a trial early to reject the null hypothesis, because large p-values might not be persuasive enough (Pocock and White 1999). Also, the nominal p-value at the final stage is much smaller than the overall p-value of the design. That is, the trial might stop at the final stage with a small nominal p-value, but the test is not rejected, which might not be easy to justify.

O'Brien-Fleming Method

O'Brien-Fleming boundary values are inversely proportional to the square root of information levels on the standardized Z scale (O'Brien and Fleming 1979). The O'Brien-Fleming boundary is

conservative in the early stages and tends to stop the trials early only with a small p-value. But the nominal value at the final stage is close to the overall p-value of the design.

Power Family Method

The power family method (Wang and Tsiatis 1987; Emerson and Fleming 1989; Pampallona and Tsiatis 1994) generalizes the Pocock and O'Brien-Fleming methods with a power parameter to allow continuous movement between the Pocock and O'Brien-Fleming methods. The power parameter is $\rho = 0$ for the Pocock method and $\rho = 0.5$ for the O'Brien-Fleming method.

Triangular Method

The unified family triangular method (Kittelson and Emerson 1999) contains straight-line boundaries on the score scale. For a one-sided trial with early stopping either to reject and to accept the null hypothesis, the method produces a triangular continuation region. The boundary shape is specified with the slope parameter τ.

Unified Family Method

The unified family method (Kittelson and Emerson 1999) extends power family methods to incorporate the triangular method, which contains straight-line boundaries on the score scale.

Haybittle-Peto Method

The Haybittle-Peto method (Haybittle 1971; Peto et al. 1976) uses a Z value of 3 for the critical values in interim stages and derives the critical value at the final stage. With this method, the final-stage critical value is close to the original design without interim monitoring. The SEQDESIGN procedure extends this method further to allow for different Z or nominal p-values for the boundaries.

Whitehead Method

Whitehead methods (Whitehead and Stratton 1983; Whitehead 1997, 2001) derive the boundary values by adapting the continuous monitoring tests to the discrete monitoring of group sequential tests. With early stopping to reject or accept the null hypothesis in a one-sided test, the derived continuation region has a triangular shape on the score-scaled boundaries. Only elementary calculations are needed to derive the boundary values in Whitehead's triangular methods. The resulting Type I error probability and power are extremely close but differ slightly from the specified values due to the approximations used in deriving the tests (Jennison and Turnbull 2000, p. 106). The SEQDESIGN procedure provides the BOUNDARYKEY= option to adjust the boundary value at the final stage for the exact Type I or Type II error probability levels.

Error Spending Method

The error spending method uses the specified α and β errors to be used at each stage of the design to derive the boundary values.

Error Spending Function Method

The error spending function method uses the error spending function to compute the α and β errors to be used at each stage of the design and then to derive the boundary values for these errors. The following four error spending functions are available in the SEQDESIGN procedure:

- The Pocock-type error spending function (Lan and DeMets 1983) produces boundaries similar to those produced with Pocock's method.
- The O'Brien-Fleming-type error spending function (Lan and DeMets 1983) produces boundaries similar to those produced with the O'Brien-Fleming method.
- The gamma error spending function (Hwang, Shih, and DeCani 1990) specifies a gamma cumulative error spending function indexed by the gamma parameter γ. The boundaries created with $\gamma = 1$ are similar to the boundaries from the Pocock method, and the boundaries created with $\gamma = -4$ or $\gamma = -5$ are similar to the boundaries from the O'Brien-Fleming method.
- The power error spending function (Jennison and Turnbull 2000, p. 148) specifies a power cumulative error spending function indexed by the power parameter ρ. The boundaries created with $\rho = 1$ are similar to the boundaries from the Pocock method, and the boundaries created with $\rho = 3$ are similar to the boundaries from the O'Brien-Fleming method.

Table Output

For each design, the SEQDESIGN procedure displays the "Design Information," "Method Information," and "Boundary Information" tables by default.

Boundary Information

The "Boundary Information" table displays the following information at each stage:

- proportion of information
- actual information level, if the maximum information is either specified or derived
- alternative references with the specified statistic scale. If a p-value scale is specified, the standardized Z scale is used.
- boundary values with the specified statistic scale to reject or accept the null hypothesis

Note that implicitly, the boundary information table also contains variables for the boundary scale, stopping criterion, and type of alternative hypothesis. That is, if an ODS statement is used to save the table, the data set also contains the variables _Scale_ for the boundary scale, _Stop_ for the stopping criterion, and _ALT_ for the type of alternative hypothesis.

Design Information

The "Design Information" table displays the design specifications and derived statistics. The derived Max Information (Percent Fixed-Sample) is the maximum information for the sequential design in percentage of the corresponding fixed-sample information.

The Null Ref ASN (Percent Fixed-Sample) is the average sample number (expected sample size for nonsurvival data or expected number of events for survival data) required under the null hypothesis for the group sequential design in percentage of the corresponding fixed-sample design. Similarly, the Alt Ref ASN (Percent Fixed-Sample) is the average sample number required under the alternative reference for the group sequential design in percentage of the corresponding fixed-sample design.

If both the maximum information (MAXINFO= option) and the alternative reference θ_1 (ALTREF= option) are specified, then either the ALPHA= option is used to derive the Type II error probability β (BOUNDARYKEY=ALPHA) or the BETA= option is used to derive the Type I error probability α (BOUNDARYKEY=BETA).

Error Spending Information

The "Error Spending Information" table displays the following information at each stage:

- proportion of information
- actual information level, if the maximum information is either specified or derived
- cumulative error spending for each boundary

Method Information

The "Method Information" table displays detailed method information for the design. For each boundary, it displays the following:

- the group sequential method used
- the α or β errors
- the specified parameter ρ, if an error spending function is used
- the specified parameters ρ and τ with the derived critical value C, if a unified family method is used

- the alternative reference θ_1, if either the ALTREF= or the MAXINFO= option is specified
- the derived drift parameter, $\theta_1\sqrt{I_X}$, where I_X is the maximum information and θ_1 is the alternative reference

Note that the alternative references are displayed with the MLE scale in the "Method Information" table. In contrast, the alternative references in the "Boundary Information" table are displayed with the specified statistic scale (if the p-value scale is not specified) or the standardized Z scale (if the p-value scale is specified).

Powers and Expected Sample Sizes

The "Powers and Expected Sample Sizes" table displays the following information under each of the specified hypothetical references $\theta = c_i\theta_1$, where θ_1 is the alternative reference and c_i are values specified in the CREF= option.

- coefficient c_i for the hypothetical references. The value $c_i = 0$ corresponds to the null hypothesis and $c_i = 1$ corresponds to the alternative hypothesis
- power
- expected sample size, as percentage of fixed-sample size

For a one-sided design, the power and expected sample sizes under the hypothetical references $\theta = c_i\theta_1$ are displayed.

For a two-sided symmetric design, the power and expected sample sizes under each of the hypothetical references $\theta = c_i\theta_{1u}$ are displayed, where θ_{1u} is the upper alternative reference.

For a two-sided asymmetric design, the power and expected sample sizes under each of the hypothetical references $\theta = c_i\theta_{1l}$ and $\theta = c_i\theta_{1u}$ are displayed, where θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively.

For a two-sided design, the power is the probability of correctly rejecting the null hypothesis for the correct alternative. Thus, under the null hypothesis, the displayed power corresponds to a one-sided Type I error probability level—that is, the lower α level or the upper α level.

The expected sample size as a percentage of the corresponding fixed-sample design is

$$100 \times \frac{\sum_{k=1}^{K} p_k\, I_k}{I_0}$$

where p_k is the stopping probability at stage k, $\sum_{k=1}^{K} p_k\, I_k$ is the expected information level, and I_0 is the information level for the fixed-sample design.

Sample Size Summary

When you use the SAMPLESIZE statement with the SEQDESIGN procedure, the "Sample Size Summary" table displays parameters for the sample size computation. It also displays the expected sample sizes or numbers of events for the model under both the null and alternative hypotheses.

The expected sample size is the average sample size

$$\frac{\sum_{k=1}^{K} p_k I_k}{I_0} N_0$$

where p_k is the stopping probability at stage k, $\sum_{k=1}^{K} p_k I_k$ is the expected information level, I_0 is the information level for the fixed-sample design, and N_0 is the sample size for the fixed-sample design.

The expected number of events is the average number of events

$$\frac{\sum_{k=1}^{K} p_k I_k}{I_0} D_0$$

where D_0 is the fixed-sample number of events for the model.

Sample Size Information

The "Sample Sizes (N)" table displays the required sample sizes and information levels at each stage, in both fractional and integer numbers. The derived fractional sample sizes are under the heading "Fractional N." These sample sizes are rounded up to integers under the heading "Ceiling N." The matched integer sample sizes are also displayed for two-sample tests.

The "Required Number of Events (D)" table displays the required number of events required and information level at each stage.

The "Number of Events (D) and Sample Sizes (N)" table displays the number of events and sample size required at each stage with the study time. The derived times under the heading "Fractional Time" are not integers. These times are rounded up to integers under the heading "Ceiling Time."

Stopping Probabilities

The "Expected Cumulative Stopping Probabilities" table displays the following information under each of the specified hypothetical references $\theta = c_i \theta_1$, where c_i are values specified in the CREF= option, and θ_1 is the alternative reference:

- coefficient c_i for the hypothetical references. The value $c_i = 0$ corresponds to the null hypothesis, and $c_i = 1$ corresponds to the alternative hypothesis
- expected stopping stage
- source of the stopping probability: reject H_0 (with STOP=REJECT or STOP=BOTH), accept H_0 (with STOP=ACCEPT or STOP=BOTH), or either reject or accept H_0 (with STOP=BOTH)
- expected cumulative stopping probabilities at each stage

For a one-sided design, the expected cumulative stopping probabilities under the hypothetical references $\theta = c_i \theta_1$ are displayed.

For a two-sided design, the expected cumulative stopping probabilities under each of the hypothetical references $\theta = c_i\theta_{1l}$ and $\theta = c_i\theta_{1u}$ are displayed, where θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively.

Note that for a symmetric two-sided design, only the expected cumulative stopping probabilities under the hypothetical references $\theta = c_i\theta_{1u}$ are derived.

The expected stopping stage is given by $k_0 + d$, where the integer k_0 and the fraction d $(0 \leq d < 1)$ are derived from the expected information level equation

$$\sum_{k=1}^{K} p_k\, I_k = I_{k_0} + d\,(I_{(k_0+1)} - I_{k_0})$$

where p_k is the stopping probability at stage k.

For equally spaced information levels, the expected stopping stage is reduced to the weighted average

$$\sum_{k=1}^{K} p_k\, k$$

ODS Table Names

PROC SEQDESIGN assigns a name to each table it creates. You must use these names to reference tables when using the Output Delivery System (ODS). These names are listed in Table 77.12. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 77.12 ODS Tables Produced by PROC SEQDESIGN

ODS Table Name	Description	Statement	Option
Boundary	Boundary values		
Design	Design information		
ErrSpend	Error spending		ERRSPEND
Method	Method information		
PowerSampleSize	Power and expected sample size		PSS
SampleSize	Derived sample sizes	SAMPLESIZE	
SampleSizeSummary	Sample size summary	SAMPLESIZE	
StopProb	Stopping probabilities		STOPPROB

Graphics Output

This section describes the use of ODS for creating graphics with the SEQDESIGN procedure. To request these graphs, you must specify the ODS GRAPHICS ON statement in addition to the associated graphics options in the PROC SEQDESIGN statement. Except for the PLOTS=BOUNDARY

option, where a detailed boundary plot is generated for each design separately, each option produces a plot for all designs together. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Sequential ASN Plot

The PLOTS=ASN option displays the average sample numbers (expected sample sizes for nonsurvival data or expected numbers of events for survival data) under various hypothetical references. The average sample numbers are connected for each design, and these connected curves for all designs are displayed in the "Sequential ASN Plot" graph.

For a one-sided design, average sample numbers under the hypothetical references $\theta = c_i\theta_1$ are displayed, where c_i are the values specified in the CREF= option and θ_1 is the alternative reference. The horizontal axis displays the c_i values of these hypothetical references.

For a two-sided design, average sample numbers under each of the hypothetical references $\theta = c_i\theta_{1l}$ and $\theta = c_i\theta_{1u}$ are displayed, where θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively. The horizontal axis displays $-c_i$ values for lower hypothetical references $\theta = c_i\theta_{1l}$ and c_i values for upper hypothetical references $\theta = c_i\theta_{1u}$.

Note that for a symmetric two-sided design, only average sample numbers under the hypothetical references $\theta = c_i\theta_{1u}$ are derived.

Sequential Boundary Plot

The PLOTS=BOUNDARY option displays boundary values and the acceptance and rejection regions at each stage for each design separately in the "Detailed Boundary Information" graph. The BOUNDARYSCALE= option is used to specify the scale of the boundaries on the vertical axis. The keywords MLE, SCORE, STDZ, and PVALUE in the BOUNDARYSCALE= option correspond to the boundary with the MLE scale, score statistic scale, standardized normal Z scale, and p-value scale, respectively.

The stage numbers are displayed on the horizontal axis. In addition, the HSCALE= option in the PLOTS=BOUNDARY option can be used to specify the scale on the horizontal axis. The keywords INFO and SAMPLESIZE in the HSCALE= option correspond to the information levels and sample sizes, respectively.

Combined Sequential Boundary Plot

The PLOTS=COMBINEDBOUNDARY option displays boundary values. The boundary values are connected for each boundary in each design, and these connected curves for all designs are displayed in the "Sequential Boundary Information" graph. The BOUNDARYSCALE= option is used to specify the scale of the boundaries on the vertical axis. The keywords MLE, SCORE, STDZ, and PVALUE in the BOUNDARYSCALE= option correspond to the boundary with the MLE scale, score statistic scale, standardized normal Z scale, and p-value scale, respectively.

The HSCALE= option in the PLOTS=COMBINEDBOUNDARY option can be used to specify the scale on the horizontal axis. The keywords INFO, SAMPLESIZE, and STAGE in the HSCALE= option correspond to the information levels, sample sizes, and stage numbers, respectively.

Sequential Error Spending Plot

The PLOTS=ERRSPEND option displays the cumulative error spending at each stage on each boundary in the "Sequential Error Spending Plot" graph. A legend table uses the design labels to identify the curves for the corresponding design in the plot. Another legend table uses symbols to identify boundaries in the plot.

Sequential Power Plot

The PLOTS=POWER option displays the powers under various hypothetical references. The powers are connected for each design, and these connected curves for all designs are displayed in the "Sequential Power Plot" graph.

For a one-sided design, powers under hypothetical references $\theta = c_i\theta_1$ are displayed, where c_i are the values specified in the CREF= option and θ_1 is the alternative reference. The horizontal axis displays the c_i values of these hypothetical references.

For a two-sided design, powers under hypothetical references $\theta = c_i\theta_{1l}$ and $\theta = c_i\theta_{1u}$ are displayed, where θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively. The horizontal axis displays $-c_i$ values for lower hypothetical references $\theta = c_i\theta_{1l}$ and c_i values for upper hypothetical references $\theta = c_i\theta_{1u}$.

Note that for a symmetric two-sided design, only powers under hypothetical references $\theta = c_i\theta_{1u}$ are derived.

ODS Graphics

PROC SEQDESIGN assigns a name to each graph it creates. You can use these names to reference the graphs when using ODS. The names are listed in Table 77.13.

To request these graphs, you must specify the ODS GRAPHICS ON statement in addition to the options indicated in Table 77.13. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Table 77.13 ODS Graphics Produced by PROC SEQDESIGN

ODS Graph Name	Plot Description	Option
ASNPlot	Average sample numbers	PLOTS=ASN
BoundaryPlot	Detailed boundary values	PLOTS=BOUNDARY
CombinedBoundaryPlot	Boundary values	PLOTS=COMBINEDBOUNDARY
ErrSpendPlot	Error spending	PLOTS=ERRSPEND
PowerPlot	Power curves	PLOTS=POWER

Examples: SEQDESIGN Procedure

The following examples demonstrate the usage of group sequential methods. Example 77.1 uses the NSTAGES=1 option to derive boundaries of critical values for a fixed-sample design. The remaining examples use different methods to create boundaries for various group sequential designs.

Example 77.1: Creating Fixed-Sample Designs

This example demonstrates a one-sided fixed-sample design and a two-sided fixed-sample design. The following statements request a fixed-sample design with an upper alternative:

```
ods graphics on;
proc seqdesign pss
               ;
   OneSidedFixedSample: design nstages=1
                               alt=upper
                               alpha=0.025 beta=0.10
                               ;
   samplesize model=onesamplemean(mean=0.25);
run;
ods graphics off;
```

In the DESIGN statement, the label `OneSidedFixedSample` identifies the design in the output tables. The NSTAGES=1 option specifies that the design has only one stage; this corresponds to a fixed-sample design. In the SEQDESIGN procedure, the null hypothesis for the design is $H_0 : \theta = 0$ and the ALT=UPPER option specifies an upper alternative hypothesis $H_1 : \theta = \theta_1 > 0$. The MEAN=0.25 option in the SAMPLESIZE statement specifies the upper alternative reference $\theta_1 = 0.25$.

The options ALPHA=0.025 and BETA=0.10 specify the Type I error probability level $\alpha = 0.025$ and the Type II error probability level $\beta = 0.10$. That is, the design has a power $1 - \beta = 0.90$ at $\theta_1 = 0.25$.

The "Design Information" table in Output 77.1.1 displays design specifications and the derived statistics such as power. As expected, the derived statistics such as maximum information and average sample number (in percentage of its corresponding fixed-sample information) are 100 for the fixed-sample design (NSTAGES=1). Also, for a fixed-sample design, the STOP= and METHOD= options in the DESIGN statement are not applicable.

Output 77.1.1 One-Sided Fixed-Sample Design Information

The SEQDESIGN Procedure
Design: OneSidedFixedSample

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Upper
Alternative Reference	0.25
Number of Stages	1
Alpha	0.025
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	100
Max Information	168.1188
Null Ref ASN (Percent of Fixed Sample)	100
Alt Ref ASN (Percent of Fixed Sample)	100

The "Method Information" table in Output 77.1.2 displays the α and β error levels. It also displays the derived drift parameter, which is the standardized reference improvement, $\theta_1\sqrt{I_0}$, where θ_1 is the alternative reference and I_0 is the maximum information for the design. If either θ_1 or I_0 is specified, the other statistic is derived in the SEQDESIGN procedure. For a fixed-sample design,

$$\theta_1\sqrt{I_0} = \Phi^{-1}(1-\alpha) + \Phi^{-1}(1-\beta) = \Phi^{-1}(0.975) + \Phi^{-1}(0.90) = 3.2415$$

Output 77.1.2 Method Information

Method Information

Boundary	Alpha	Beta	Alternative Reference	Drift
Upper Alpha	0.02500	0.10000	0.25	3.241516

The "Boundary Information" table in Output 77.1.3 displays information level, alternative reference, and boundary value at each stage. The information proportion indicates the proportion of maximum information available at the stage. With only one stage for a fixed-sample design, the proportion is 1. With the SAMPLESIZE statement, the required sample size N is also displayed under the heading "Information Level."

Output 77.1.3 Boundary Information

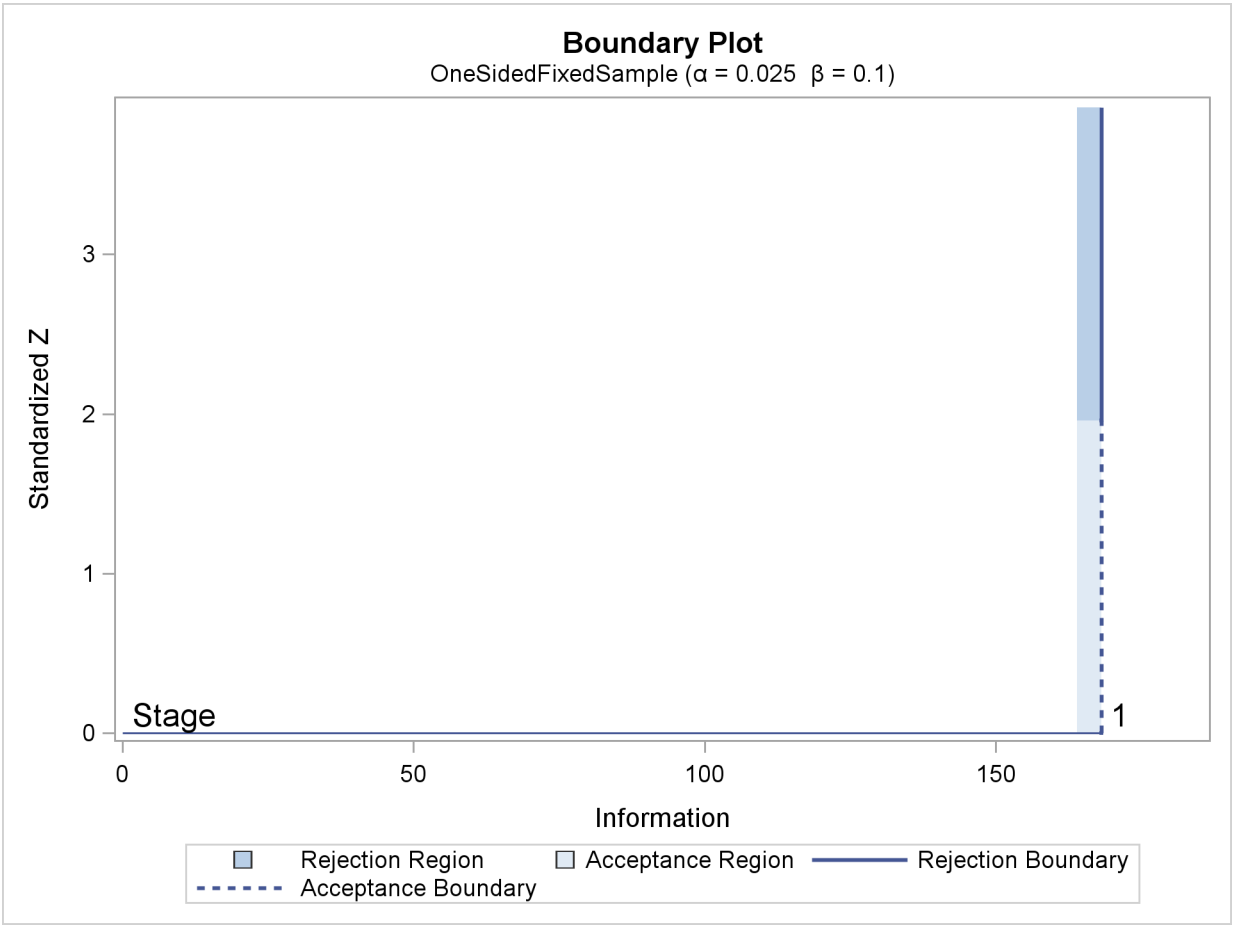
Boundary Information (Standardized Z Scale)
Null Reference = 0

	--------Information Level-------			-Alternative- --Reference--	-Boundary Values- ------Upper------
Stage	Proportion	Actual	N	Upper	Alpha
1	1.0000	168.1188	168.1188	3.24152	1.95996

With the default BOUNDARYSCALE=STDZ option, output alternative references and boundaries are displayed with the standardized normal Z scale. The alternative reference on the standardized Z scale at stage 1 is given by $\theta_1\sqrt{I_1}$, where I_1 is the information level at stage 1. With a boundary value 1.96, the hypothesis of $\theta = 0$ is rejected if the standardized normal statistic $Z \geq 1.96$.

If the ODS GRAPHICS ON statement is specified, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 77.1.4. The boundary values in the "Boundary Information" table in Figure 77.1.3 are displayed in the plot.

Output 77.1.4 Boundary Plot



The "Sample Size Summary" table in Output 77.1.5 displays parameters for the sample size computation of the test for a normal mean.

Output 77.1.5 Sample Size Summary

Sample Size Summary

Test	One-Sample Mean
Mean	0.25
Standard Deviation	1
Max Sample Size	168.1188
Expected Sample Size (Null Ref)	168.1188
Expected Sample Size (Alt Ref)	168.1188

The "Sample Sizes (N)" table in Output 77.1.6 displays the derived sample sizes, in both fractional and integer numbers. With the resulting integer sample sizes, the corresponding information level is slightly larger than the level from the design. This can increase the power slightly if the integer sample size is used in the trial.

Output 77.1.6 Derived Sample Sizes

Sample Sizes (N)
One-Sample Z Test for Mean

	Fractional N		Ceiling N	
Stage	N	Information	N	Information
1	168.12	168.1	169	169.0

The following statements request a two-sided fixed-sample design with a specified alternative reference:

```
ods graphics on;
proc seqdesign altref=1.2
               pss
               ;
   TwoSidedFixedSample: design nstages=1
                               alt=twosided
                               alpha=0.05 beta=0.10
                               ;
   samplesize model=twosamplemean(stddev=2 weight=2);
run;
ods graphics off;
```

In the SEQDESIGN procedure, the null hypothesis for the design is $H_0 : \theta = 0$. The ALT=TWOSIDED option specifies a two-sided alternative hypothesis $H_1 : \theta = \theta_1 \neq 0$. The ALTREF=1.2 option in the PROC SEQDESIGN statement specifies the alternative reference $\theta_1 = \pm 1.2$.

The ALPHA=0.05 option specifies the two-sided Type I error probability level $\alpha = 0.05$. That is, the lower and upper Type I error probabilities $\alpha_l = \alpha_u = 0.025$. The BETA=0.10 option specifies the Type II error probability level $\beta = 0.10$, and the design has a power $1 - \beta = 0.90$ at the alternative reference $\theta_1 = \pm 1.2$.

The "Design Information" table in Output 77.1.7 displays design specifications and the derived power. With a specified alternative reference, the maximum information is derived.

Output 77.1.7 Two-Sided Fixed-Sample Design Information

The SEQDESIGN Procedure
Design: TwoSidedFixedSample

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Alternative Reference	1.2
Number of Stages	1
Alpha	0.05
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	100
Max Information	7.296822
Null Ref ASN (Percent of Fixed Sample)	100
Alt Ref ASN (Percent of Fixed Sample)	100

The "Method Information" table in Output 77.1.8 displays the α and β errors, alternative references, and drift parameter. For a fixed-sample design, the derived drift parameter

$$\theta_1 \sqrt{I_0} = \Phi^{-1}(1 - \frac{\alpha}{2}) + \Phi^{-1}(1 - \beta) = \Phi^{-1}(0.975) + \Phi^{-1}(0.90) = 3.2415$$

Output 77.1.8 Method Information

Method Information

Boundary	Alpha	Beta	Alternative Reference	Drift
Upper Alpha	0.02500	0.10000	1.2	3.241516
Lower Alpha	0.02500	0.10000	-1.2	-3.24152

With a specified alternative reference $\theta_1 = 1.2$, the maximum information

$$I_0 = \left(\frac{3.2415}{1.2}\right)^2 = 7.2968$$

The default "Boundary Information" table in Output 77.1.9 displays information level, alternative reference, and boundary values. With the default BOUNDARYSCALE=STDZ option, alternative reference and boundary values are displayed with the standardized normal Z scale. Thus, the standardized alternative references $\pm\theta_1\sqrt{I_0}$ are displayed.

Output 77.1.9 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

	Information Level			Alternative Reference	
Stage	Proportion	Actual	N	Lower	Upper
1	1.0000	7.296822	131.3428	-3.24152	3.24152

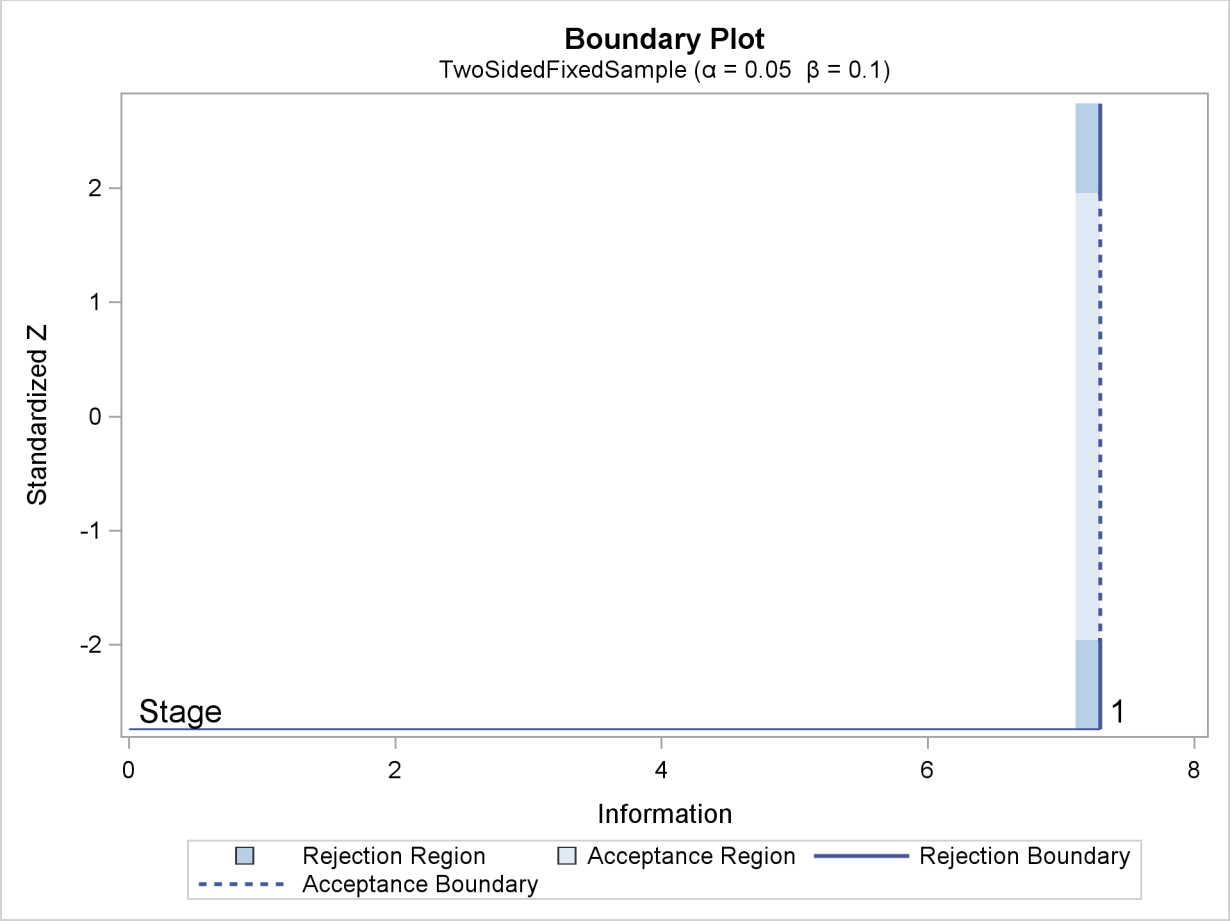
Boundary Information (Standardized Z Scale)
Null Reference = 0

	Boundary Values	
	Lower	Upper
Stage	Alpha	Alpha
1	-1.95996	1.95996

With boundary values of -1.96 and 1.96, the hypothesis of $\theta = 0$ is rejected if the standardized normal statistic $Z \geq 1.96$ or $Z \leq -1.96$.

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 77.1.10 . The boundary values in the "Boundary Information" table in Figure 77.1.9 are displayed in the plot.

Output 77.1.10 Boundary Plot



The "Sample Size Summary" table in Output 77.1.11 displays parameters for the sample size computation of the test for a normal mean.

Output 77.1.11 Sample Size Summary

Sample Size Summary

Test	Two-Sample Means
Mean Difference	1.2
Standard Deviation	2
Max Sample Size	131.3428
Expected Sample Size (Null Ref)	131.3428
Expected Sample Size (Alt Ref)	131.3428
Weight (Group A)	2
Weight (Group B)	1

The "Sample Sizes (N)" table in Output 77.1.12 displays the derived sample sizes, in both fractional and integer numbers. With the WEIGHT=2 option, the allocation ratio is 2 for the first group and 1 for the second group. With the resulting integer sample sizes, the corresponding information level is slightly larger than the level from the design. This can increase the power slightly if the integer sample size is used in the trial.

Output 77.1.12 Derived Sample Sizes

Sample Sizes (N)
Two-Sample Z Test for Mean Difference

	Fractional N			
Stage	N	N(Grp 1)	N(Grp 2)	Information
1	131.34	87.56	43.78	7.2968

Sample Sizes (N)
Two-Sample Z Test for Mean Difference

	Ceiling N			
Stage	N	N(Grp 1)	N(Grp 2)	Information
1	132	88	44	7.3333

Example 77.2: Creating a One-Sided O'Brien-Fleming Design

This example demonstrates a group sequential design for a clinical study. A clinic is conducting a study on the effect of vitamin C supplements in treating flu symptoms. The study groups consist of patients in the clinic with their first sign of flu symptoms within the last 24 hours. These individuals are randomly assigned to either the control group, which receives the placebo pills, or the treatment group, which receives large doses of vitamin C supplements. At the end of a five-day period, the flu symptoms of each individual are recorded.

Suppose that from past experience, 60% of individuals experiencing flu symptoms have the symptoms disappeared within five days. The clinic wants to detect a 75% symptoms disappearance with a high probability in the trial. A test that compares the proportions directly is to specify a null hypothesis $H_0 : \theta = p_a - p_b = 0$ with a Type I error probability level $\alpha = 0.025$, where p_a and p_b are the proportions of symptoms' disappearance in the treatment group and control group, respectively. A one-sided alternative $H_1 : \theta > 0$ is also specified with a power of $1 - \beta = 0.90$ at $H_1 : \theta = 0.15$.

For a one-sided fixed-sample design, the critical value for the standardized Z test statistic is given by $C_\alpha = \Phi^{-1}(1-\alpha) = 1.96$. That is, at the end of study, if the test statistic $z \geq C_\alpha$, then the null hypothesis is rejected and the efficacy of vitamin C supplements is declared. Otherwise, the null hypothesis is not rejected and the effect of vitamin C supplements is not significant.

To achieve a $1 - \beta = 0.90$ power at $H_1 : \theta = 0.15$ for a fixed-sample design, the information required is given by

$$I_0 = \frac{(\Phi^{-1}(1-\alpha) + \Phi^{-1}(1-\beta))^2}{0.15^2} = \frac{(1.96 + 1.28155)^2}{0.0225} = 466.99$$

With an equal sample size on the treatment and control groups, $N_a = N_b$, the sample size required for each group under H_1 is computed from the information I_0:

$$N_a = N_b = \left(p_{1a}\left(1 - p_{1a}\right) + p_{1b}\left(1 - p_{1b}\right)\right) I_0$$

where $p_{1a} = 0.75$ and $p_{1b} = 0.60$ are proportions in the treatment and control groups under H_1. That is,

$$N_a = N_b = (0.75 \times 0.25 + 0.6 \times 0.4) \times 466.99 = 199.64$$

Thus, 200 individuals are required for each group in the fixed-sample study. See the section "Test for the Difference between Two Binomial Proportions" on page 5862 for a detailed derivation of these required sample sizes.

Instead of a fixed-sample design for the trial, a group sequential design is used to stop the trial early for ethical concerns of possible harm or an unexpected strong efficacy outcome of the new drug. It can also save time and resources in the process. The following statements invoke the SEQDESIGN procedure and request a four-stage group sequential design that uses an O'Brien-Fleming method for normally distributed statistics. The design uses a one-sided alternative hypothesis H_1 with early stopping to reject or accept H_0.

```
ods graphics on;
proc seqdesign altref=0.15
               ;
   OneSidedOBrienFleming: design nstages=4
                          method=obf
                          alt=upper   stop=both
                          alpha=0.025 beta=0.10
                          ;
   samplesize model=twosamplefreq(nullprop=0.6 test=prop);
ods output Boundary=Bnd_Prop;
run;
ods graphics off;
```

In a sequential design, a hypothesis can be rejected, accepted, or continued to the next time point at each interim stage. The STOP=BOTH option specifies early stopping to reject or accept the null hypothesis. The "Design Information," "Method Information," and "Boundary Information" tables are displayed by default.

The "Design Information" table in Output 77.2.1 displays design specifications and derived statistics such as power and maximum information. With a specified alternative reference, ALTREF=0.15, the maximum information I_X is derived.

Output 77.2.1 Design Information

The SEQDESIGN Procedure
Design: OneSidedOBrienFleming

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Upper
Early Stop	Accept/Reject Null
Method	O'Brien-Fleming
Boundary Key	Both
Alternative Reference	0.15
Number of Stages	4
Alpha	0.025
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	107.6741
Max Information	502.8343
Null Ref ASN (Percent of Fixed Sample)	61.12891
Alt Ref ASN (Percent of Fixed Sample)	75.89782

The Max Information (Percent Fixed-Sample) is the ratio in percentage between the maximum information for the group sequential design and the information required for a corresponding fixed-sample design:

$$100 \times \frac{I_X}{I_0} = 100 \times \frac{502.83}{466.99} = 107.67$$

That is, if the group sequential trial does not stop at any interim stages, the information needed is 7.67% more than is needed for the corresponding fixed-sample design. For a two-sample test for binomial proportions, the information is proportional to the sample size. Thus, 7.67% more observations are needed for the group sequential trial.

The Null Ref ASN (Percent Fixed-Sample) is the ratio in percentage between the expected sample size required under the null hypothesis for the group sequential design and the sample size required for the corresponding fixed-sample design. With a ratio of 61.1%, the expected sample size for the group sequential trial under the null hypothesis is 61.1% of the sample size in the corresponding fixed-sample design.

Similarly, the Alt Ref ASN (Percent Fixed-Sample) is the ratio in percentage between the expected sample size required under the alternative hypothesis for the group sequential design and the sample size required for the corresponding fixed-sample design. With a ratio of 75.9%, the expected sample size for the group sequential trial under the alternative hypothesis is 75.9% of the sample size in the corresponding fixed-sample design.

For a one-sided design with an upper alternative and early stopping to reject or accept the null hypothesis, upper α and β boundaries are created. The "Method Information" table in Output 77.2.2 displays the Type I error probability α, the Type II error probability β, and the derived drift parameter. The drift parameter is the standardized reference improvement between the alternative and null hypotheses at the final stage. It is also the standardized alternative reference at the final stage if the null reference is zero.

Output 77.2.2 Method Information

Method Information

Boundary	Method	Alpha	Beta	Unified Family: Rho	Unified Family: Tau	Unified Family: C
Upper Alpha	O'Brien-Fleming	0.02500	.	0.5	0	1.9784
Upper Beta	O'Brien-Fleming	.	0.10000	0.5	0	1.3852

Method Information

Boundary	Alternative Reference	Drift
Upper Alpha	0.15	3.363595
Upper Beta	0.15	3.363595

With the METHOD=OBF option, the O'Brien-Fleming method is used for each boundary. The O'Brien-Fleming method is one of the unified family methods, and the "Method Information" table displays the corresponding parameter ρ in the unified family method. The table also displays the critical values $C_\alpha = 1.9784$ for the α boundary and $C_\beta = 1.3852$ for the β boundary. These critical values are used to create the boundary values. With the default BOUNDARYSCALE=STDZ option, the boundaries are displayed with the standardized Z statistic scale.

The "Boundary Information" table in Output 77.2.3 displays information level, alternative reference, and boundary values at each stage. The default BOUNDARYSCALE=STDZ option specifies that the standardized Z scale be used to display the alternative references and boundary values. The resulting standardized alternative reference at stage k is given by $\theta_1 \sqrt{I_k}$, where θ_1 is the alternative reference and I_k is the information level at stage k, $k = 1, 2, 3, 4$.

Output 77.2.3 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

Stage	Information Level: Proportion	Information Level: Actual	Information Level: N	Alternative Reference: Upper	Boundary Values: Upper: Beta	Boundary Values: Upper: Alpha
1	0.2500	125.7086	107.4808	1.68180	-1.08860	3.95679
2	0.5000	251.4171	214.9617	2.37842	0.41946	2.79788
3	0.7500	377.1257	322.4425	2.91296	1.31347	2.28446
4	1.0000	502.8343	429.9233	3.36360	1.97840	1.97840

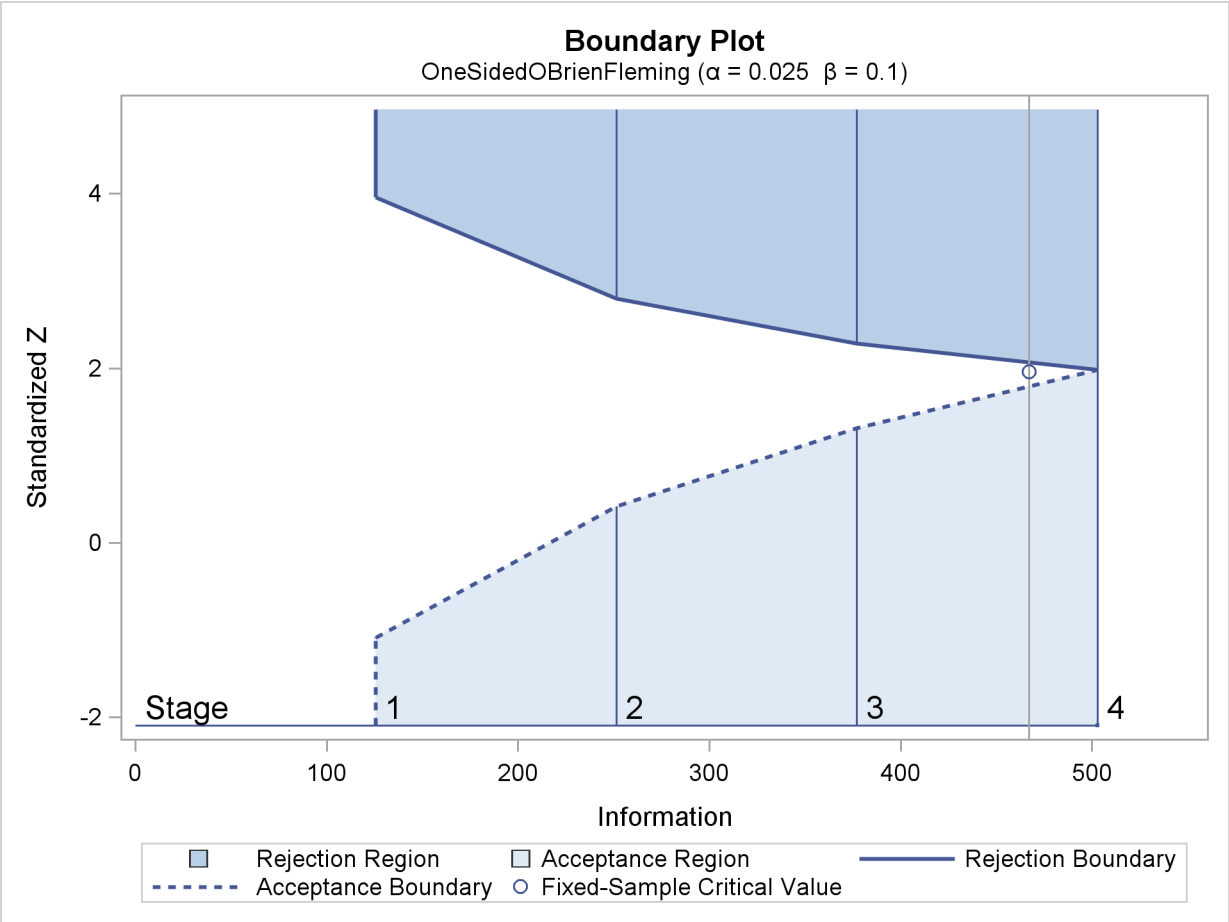
By default, equally spaced information levels are used. An information proportion is the proportion of maximum information available at each stage. With the derived maximum information, the actual information level at each stage is also displayed. With the SAMPLESIZE statement, the required sample size N is also displayed under the heading "Information Level."

At each interim stage, if the standardized Z test statistic is larger than or equal to the corresponding upper α boundary, then the hypothesis $H_0 : \theta = 0$ is rejected. If the test statistic is less than the corresponding upper β boundary, then the trial is stopped and the hypothesis H_0 is accepted. Otherwise, the process continues to the next stage. At the final stage, stage 4, the trial stops and the hypothesis H_0 is rejected if the standardized Z statistic $Z_4 \geq 1.9784$. Otherwise, the trial is accepted.

The ODS OUTPUT statement with the BOUNDARY=BND_PROP option creates an output data set that contains the resulting boundary information. After the actual data from the clinical trial are collected and analyzed at each stage with a procedure such as PROC GENMOD, the SEQTEST procedure is used to test the resulting statistics at stage 1 with the boundary information stored in the BOUND_PROP data set.

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 77.2.4.

Output 77.2.4 Boundary Plot



The horizontal axis indicates the information levels for the design. The stages are indicated by vertical lines with accompanying stage numbers. If at any stage a test statistic is in a rejection region, the trial stops and the hypothesis is rejected. If a test statistic is in an acceptance region, then the trial also stops and the hypothesis is accepted. If the statistic is not in a rejection region

or an acceptance region, the trial continues to the next stage. The boundary plot also displays the information level and critical value for the corresponding fixed-sample design.

The SEQDESIGN procedure derives the drift parameter $\theta_1\sqrt{I_X}$, where θ_1 is the alternative reference and I_X is the maximum information. If either θ_1 or I_X is specified, the other can be derived. With the SAMPLESIZE statement, the maximum information is used to compute the required sample size for the study.

The "Sample Size Summary" table in Output 77.2.5 displays parameters for the sample size computation. With the MODEL=TWOSAMPLEFREQ(NULLPROP=0.6 TEST=PROP) option in the SAMPLESIZE statement, the total sample size in each group for testing the difference between two proportions is computed. With the default REF=PROP option, the required sample sizes are computed under the alternative hypothesis. That is,

$$N_a = N_b = \left(p_{1a}\left(1 - p_{1a}\right) + p_{1b}\left(1 - p_{1b}\right)\right) I_X$$

where $p_{1b} = 0.60$ and $p_{1a} = p_{1b} + \theta_1 = 0.75$ are the proportions in the control and treatment groups, respectively, under the alternative hypothesis H_1. See the section "Test for the Difference between Two Binomial Proportions" on page 5862 for a detailed description of these parameters.

Output 77.2.5 Sample Size Summary

Sample Size Summary

Test	Two-Sample Proportions
Null Proportion	0.6
Proportion (Group A)	0.75
Test Statistic	Z for Proportion
Reference Proportions	Alt Ref
Max Sample Size	429.9233
Expected Sample Size (Null Ref)	244.0768
Expected Sample Size (Alt Ref)	303.0464

The "Sample Sizes (N)" table in Output 77.2.6 displays the required sample sizes at each stage, in both fractional and integer numbers. The derived fractional sample sizes are under the heading "Fractional N." These sample sizes are rounded up to integers under the heading "Ceiling N." In practice, integer sample sizes are used, and the resulting information levels increase slightly. Thus, 54, 108, 162, and 215 individuals are needed in each group for the four stages, respectively.

Output 77.2.6 Derived Sample Sizes

Sample Sizes (N)
Two-Sample Z Test for Proportion Difference

	Fractional N			
Stage	N	N(Grp 1)	N(Grp 2)	Information
1	107.48	53.74	53.74	125.7
2	214.96	107.48	107.48	251.4
3	322.44	161.22	161.22	377.1
4	429.92	214.96	214.96	502.8

Sample Sizes (N)
Two-Sample Z Test for Proportion Difference

	Ceiling N			
Stage	N	N(Grp 1)	N(Grp 2)	Information
1	108	54	54	126.3
2	216	108	108	252.6
3	324	162	162	378.9
4	430	215	215	502.9

Example 77.3: Creating Two-Sided Pocock and O'Brien-Fleming Designs

This example requests two 4-stage group sequential designs for normally distributed statistics with equally spaced information levels at all stages. One design uses Pocock's method and the other uses the O'Brien-Fleming method. With the default BOUNDARYSCALE=STDZ option, the output boundaries are displayed with the standardized normal Z scale.

In the following statements, the default ALT=TWOSIDED option specifies a null hypothesis with a two-sided alternative, and the default STOP=REJECT option indicates an early stop to reject the null hypothesis H_0:

```
proc seqdesign altref=0.4
               pss
               stopprob
               errspend
               ;
   TwoSidedPocock:         design nstages=4 method=poc;
   TwoSidedOBrienFleming: design nstages=4 method=obf;
   samplesize model=twosamplemean(stddev=0.8 weight=2);
run;
```

The "Design Information" table in Output 77.3.1 displays design specifications and derived statistics for the Pocock's design. With the specified ALTREF= option, the maximum information $I_X = 77.6984$ is also derived.

Output 77.3.1 Pocock Design Information

The SEQDESIGN Procedure
Design: TwoSidedPocock

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Reject Null
Method	Pocock
Boundary Key	Both
Alternative Reference	0.4
Number of Stages	4
Alpha	0.05
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	118.3143
Max Information	77.69844
Null Ref ASN (Percent of Fixed Sample)	115.6074
Alt Ref ASN (Percent of Fixed Sample)	69.74805

With the corresponding fixed-sample information

$$I_0 = \frac{(\Phi^{-1}(1-\alpha/2) + \Phi^{-1}(1-\beta))^2}{0.4^2} = \frac{(1.96 + 1.28155)^2}{0.16} = 65.6728$$

the fixed-sample information ratio is $77.6984/65.6728 = 1.1831$.

For a two-sided design with early stopping to reject the null hypothesis, lower and upper α boundaries are created. The "Method Information" table in Output 77.3.2 displays the α and β errors, alternative references, and derived drift parameters, which are the standardized alternative references at the final stage.

Output 77.3.2 Method Information

Method Information

Boundary	Method	Alpha	Beta	-------Unified Family------ Rho	Tau	C
Upper Alpha	Pocock	0.02500	0.10000	0	0	2.36129
Lower Alpha	Pocock	0.02500	0.10000	0	0	2.36129

Method Information

Boundary	Alternative Reference	Drift
Upper Alpha	0.4	3.525869
Lower Alpha	-0.4	-3.52587

With the METHOD=POC option, the Pocock method is used for each boundary. The Pocock method is one of the unified family methods, and the table also displays its corresponding parameters $\rho = 0$ as a unified family method and the derived parameters $C = 2.3613$ for the boundary values.

With the PSS option, the "Power and Expected Sample Sizes" table in Output 77.3.3 displays powers and expected sample sizes under various hypothetical references $\theta = c_i\theta_1$, where θ_1 is the alternative reference and c_i are values specified in the CREF= option. By default, $c_i = 0, 0.5, 1.0, 1.5$.

Output 77.3.3 Power and Expected Sample Size Information

Powers and Expected Sample Sizes
Reference = CRef * (Alt Reference)

CRef	Power	-Sample Size- Percent Fixed-Sample
0.0000	0.02500	115.6074
0.5000	0.34252	104.0615
1.0000	0.90000	69.7480
1.5000	0.99869	43.6600

Note that at $c_i = 0$, the null reference $\theta = 0$, and the power 0.025 corresponds to the one-sided Type I error probability 0.025. At $c_i = 1$, $\theta = \theta_1$, the power 0.9 is the power of the design. The expected sample sizes are displayed in a percentage scale to its corresponding fixed-sample size design. With the specified SAMPLESIZE statement, the expected sample sizes for the specified model in the SAMPLESIZE statement are also displayed.

With the STOPPROB option, the "Expected Cumulative Stopping Probabilities" table in Output 77.3.4 displays the expected cumulative stopping stage and cumulative stopping probability to reject the null hypothesis H_0 at each stage under various hypothetical references $\theta = c_i\theta_1$, where θ_1 is the alternative reference and c_i are values specified in the CREF= option. By default, $c_i = 0, 0.5, 1.0, 1.5$.

Output 77.3.4 Stopping Probabilities

Expected Cumulative Stopping Probabilities
Reference = CRef * (Alt Reference)

CRef	Expected Stopping Stage	Source	--------Stopping Probabilities-------- Stage_1	Stage_2	Stage_3	Stage_4
0.0000	3.908	Reject Null	0.01821	0.03155	0.04176	0.05000
0.5000	3.518	Reject Null	0.07005	0.15939	0.25242	0.34327
1.0000	2.358	Reject Null	0.27482	0.58074	0.78638	0.90002
1.5000	1.476	Reject Null	0.61145	0.92348	0.98900	0.99869

Note that at $c_i = 0$, the cumulative stopping probability to reject H_0 at the final stage is the overall Type I error probability 0.05. At $c_i = 1$, the alternative hypothesis $H_1 : \theta = \theta_1$, the cumulative

stopping probability to reject H_0 includes both the probability in the lower rejection region and the probability in the upper rejection region. This stopping probability to reject H_0 at the final stage, 0.90002, is slightly greater than the power $1 - \beta = 0.90$, which corresponds to the cumulative stopping probability in the upper rejection region only. See the section "Type I and Type II Errors" on page 5836 for a detailed description of the Type II error probability β.

The "Boundary Information" table in Output 77.3.5 displays the information level, alternative references, and boundary values at each stage. The default BOUNDARYSCALE=STDZ option specifies that the standardized Z scale be used to display the alternative references and boundary values. The resulting standardized alternative reference at stage k is given by $\theta_1 \sqrt{I_k}$, where θ_1 is the alternative reference and I_k is the information level at stage k, $k = 1, 2, 3, 4$.

Output 77.3.5 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

	Information Level			Alternative Reference	
Stage	Proportion	Actual	N	Lower	Upper
1	0.2500	19.42461	55.94288	-1.76293	1.76293
2	0.5000	38.84922	111.8858	-2.49317	2.49317
3	0.7500	58.27383	167.8286	-3.05349	3.05349
4	1.0000	77.69844	223.7715	-3.52587	3.52587

Boundary Information (Standardized Z Scale)
Null Reference = 0

	Boundary Values	
	Lower	Upper
Stage	Alpha	Alpha
1	-2.36129	2.36129
2	-2.36129	2.36129
3	-2.36129	2.36129
4	-2.36129	2.36129

By default, equally spaced information levels are used, and the procedure displays the output boundaries in terms of standardized statistics (BOUNDARYSCALE=STDZ). With the SAMPLESIZE statement, the required sample size N is also displayed under the heading "Information Level." With the Pocock method, the standardized Z boundary values are identical at all stages for each α boundary.

At each interim stage, the hypothesis of $H_0 : \theta = 0$ is rejected if the standardized normal test statistic $z \leq -2.36129$, the lower α boundary, or $z \geq 2.36129$, the upper α boundary. Otherwise, the trial continues to the next stage. At the final stage, stage 4, the trial stops and the hypothesis is rejected if the test statistic $|z_4| \geq 2.36129$. Otherwise, the hypothesis is accepted.

The "Error Spending Information" in Output 77.3.6 displays cumulative error spending at each stage for each boundary. It shows that more α errors are used in early stages than in later stages.

Output 77.3.6 Error Spending Information

Error Spending Information

Stage	-Information Level- Proportion	--------Cumulative Error Spending------- -------Lower------ Alpha	Beta	-------Upper------ Beta	Alpha
1	0.2500	0.00911	0.00002	0.00002	0.00911
2	0.5000	0.01577	0.00002	0.00002	0.01577
3	0.7500	0.02088	0.00002	0.00002	0.02088
4	1.0000	0.02500	0.10000	0.10000	0.02500

The "Sample Size Summary" table in Output 77.3.7 displays the specified parameters for the sample size computation of the two-sample test for mean difference.

Output 77.3.7 Sample Size Summary

Sample Size Summary

Test	Two-Sample Means
Mean Difference	0.4
Standard Deviation	0.8
Max Sample Size	223.7715
Expected Sample Size (Null Ref)	218.652
Expected Sample Size (Alt Ref)	131.9167
Weight (Group A)	2
Weight (Group B)	1

The "Sample Sizes (N)" table in Output 77.3.8 displays the derived sample sizes at each stage, in both fractional and integer numbers. With the WEIGHT=2 option, the allocation ratio is 2 for the first group and 1 for the second group. See the section "Test for the Difference between Two Normal Means" on page 5860 for the derivation of these sample sizes. With the fixed-sample information ratio 1.1831, the derived sample sizes in fractional numbers are derived by multiplying 1.1831 by the corresponding sample sizes in the fixed-sample design.

Output 77.3.8 Sample Sizes

Sample Sizes (N)
Two-Sample Z Test for Mean Difference

	Fractional N			
Stage	N	N(Grp 1)	N(Grp 2)	Information
1	55.94	37.30	18.65	19.4246
2	111.89	74.59	37.30	38.8492
3	167.83	111.89	55.94	58.2738
4	223.77	149.18	74.59	77.6984

Sample Sizes (N)
Two-Sample Z Test for Mean Difference

	Ceiling N			
Stage	N	N(Grp 1)	N(Grp 2)	Information
1	57	38	19	19.7917
2	113	75	38	39.4082
3	168	112	56	58.3333
4	225	150	75	78.1250

These fractional sample sizes are rounded up to integers under the heading "Ceiling N." When the resulting integer sample sizes are used, the corresponding information levels are slightly larger than the levels from the design. This can increase the power slightly if a trial uses these integer sample sizes.

Note that compared with other designs, a Pocock design can stop the trial early with a larger p-value. However, this might not be persuasive enough to make a new treatment widely accepted (Pocock and White, 1999).

The "Design Information" table in Output 77.3.9 displays design specifications and the derived statistics for the O'Brien-Fleming design. With the specified ALTREF= option, the maximum information $I_X = 67.1268$ is derived.

Output 77.3.9 O'Brien-Fleming Design Information

The SEQDESIGN Procedure
Design: TwoSidedOBrienFleming

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Reject Null
Method	O'Brien-Fleming
Boundary Key	Both
Alternative Reference	0.4
Number of Stages	4
Alpha	0.05
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	102.2163
Max Information	67.12682
Null Ref ASN (Percent of Fixed Sample)	101.5728
Alt Ref ASN (Percent of Fixed Sample)	76.7397

With the corresponding fixed-sample information

$$I_0 = \frac{(\Phi^{-1}(1-\alpha/2) + \Phi^{-1}(1-\beta))^2}{0.4^2} = \frac{(1.96 + 1.28155)^2}{0.16} = 65.6728$$

the fixed-sample information ratio is $67.1268/65.6728 = 1.022$. That is, the maximum information for the O'Brien-Fleming design is only 2.2% more than for the corresponding fixed-sample design.

The "Method Information" table in Output 77.3.10 displays the Type I α level and Type II β level. It also displays the derived drift parameter $\theta_1\sqrt{I_X}$, which is the standardized alternative reference at the final stage.

Output 77.3.10 Method Information

Method Information

				Unified Family		
Boundary	Method	Alpha	Beta	Rho	Tau	C
Upper Alpha	O'Brien-Fleming	0.02500	0.10000	0.5	0	2.02429
Lower Alpha	O'Brien-Fleming	0.02500	0.10000	0.5	0	2.02429

Method Information

Boundary	Alternative Reference	Drift
Upper Alpha	0.4	3.277238
Lower Alpha	-0.4	-3.27724

With the METHOD=OBF option, the O'Brien-Fleming method is used for each boundary. The O'Brien-Fleming method is one of the unified family methods, and the table also displays its corresponding parameters $\rho = 0.5$ as a unified family method and the derived parameter $C = 2.0243$ for the boundary values.

With the PSS option, the "Power and Expected Sample Sizes" table in Output 77.3.11 displays powers and expected sample sizes under various hypothetical references $\theta = c_i\theta_1$, where θ_1 is the alternative reference and c_i are values specified in the CREF= option.

Output 77.3.11 Power and Expected Sample Size Information

Powers and Expected Sample Sizes
Reference = CRef * (Alt Reference)

CRef	Power	-Sample Size- Percent Fixed-Sample
0.0000	0.02500	101.5728
0.5000	0.36495	96.3684
1.0000	0.90000	76.7397
1.5000	0.99821	57.2590

Compared with the corresponding Pocock design, the O'Brien-Fleming design has a smaller maximum sample size, and smaller expected sample sizes under hypothetical references $\theta = 0$ and $\theta = 0.5\,\theta_1$, but larger expected sample sizes under hypothetical references $\theta = \theta_1$ and $\theta = 1.5\,\theta_1$.

With the STOPPROB option, the "Expected Cumulative Stopping Probabilities" table in Output 77.3.12 displays the expected stopping stage and cumulative stopping probability to reject the null hypothesis at each stage under various hypothetical references $\theta = c_i\theta_1$, where θ_1 is the alternative reference and c_i are values specified in the CREF= option.

Output 77.3.12 Stopping Probabilities

Expected Cumulative Stopping Probabilities
Reference = CRef * (Alt Reference)

CRef	Expected Stopping Stage	Source	--------Stopping Probabilities------- Stage_1	Stage_2	Stage_3	Stage_4
0.0000	3.975	Reject Null	0.00005	0.00422	0.02091	0.05000
0.5000	3.771	Reject Null	0.00062	0.04430	0.18392	0.36515
1.0000	3.003	Reject Null	0.00798	0.29296	0.69603	0.90000
1.5000	2.241	Reject Null	0.05584	0.73031	0.97315	0.99821

Compared with the corresponding Pocock design, the O'Brien-Fleming design has smaller stopping probabilities in early stages under each hypothetical reference.

The "Boundary Information" table in Output 77.3.13 displays the boundary values for the design that uses the O'Brien-Fleming method. Compared with the Pocock method, the standardized statistics α boundary values derived from the O'Brien-Fleming method in absolute values are larger in

early stages and smaller in later stages. This makes the O'Brien-Fleming design less likely to reject the null hypothesis in early stages than the Pocock design. With the derived parameter $C = 2.0243$ for the α boundary, the α boundaries at stage j are computed as $C\sqrt{4/j}$, $j = 1, \ldots, 4$.

Output 77.3.13 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

	Information Level			Alternative Reference	
Stage	Proportion	Actual	N	Lower	Upper
1	0.2500	16.7817	48.33131	-1.63862	1.63862
2	0.5000	33.56341	96.66262	-2.31736	2.31736
3	0.7500	50.34511	144.9939	-2.83817	2.83817
4	1.0000	67.12682	193.3252	-3.27724	3.27724

Boundary Information (Standardized Z Scale)
Null Reference = 0

	Boundary Values	
	Lower	Upper
Stage	Alpha	Alpha
1	-4.04859	4.04859
2	-2.86278	2.86278
3	-2.33745	2.33745
4	-2.02429	2.02429

The "Error Spending Information" in Output 77.3.14 displays cumulative error spending at each stage for each boundary. With smaller α spending in early stages for the O'Brien-Fleming method, it also indicates that the O'Brien-Fleming design is less likely to reject the null hypothesis in early stages than the Pocock design.

Output 77.3.14 Error Spending Information

Error Spending Information

	Information Level	Cumulative Error Spending			
		Lower		Upper	
Stage	Proportion	Alpha	Beta	Beta	Alpha
1	0.2500	0.00003	0.00000	0.00000	0.00003
2	0.5000	0.00211	0.00000	0.00000	0.00211
3	0.7500	0.01046	0.00000	0.00000	0.01046
4	1.0000	0.02500	0.10000	0.10000	0.02500

The "Sample Size Summary" table in Output 77.3.15 displays the specified parameters for the sample size computation of the two-sample test for mean difference.

Output 77.3.15 Sample Size Summary

Sample Size Summary

Test	Two-Sample Means
Mean Difference	0.4
Standard Deviation	0.8
Max Sample Size	193.3252
Expected Sample Size (Null Ref)	192.1081
Expected Sample Size (Alt Ref)	145.1404
Weight (Group A)	2
Weight (Group B)	1

The "Sample Sizes (N)" table in Output 77.3.16 displays the derived sample sizes at each stage, in both fractional and integer numbers. With the fixed-sample information ratio 1.0222, the required sample sizes in fractional numbers are derived by multiplying 1.0222 by the corresponding sample sizes in the fixed-sample design.

Output 77.3.16 Derived Sample Sizes

Sample Sizes (N)
Two-Sample Z Test for Mean Difference

	Fractional N			
Stage	N	N(Grp 1)	N(Grp 2)	Information
1	48.33	32.22	16.11	16.7817
2	96.66	64.44	32.22	33.5634
3	144.99	96.66	48.33	50.3451
4	193.33	128.88	64.44	67.1268

Sample Sizes (N)
Two-Sample Z Test for Mean Difference

	Ceiling N			
Stage	N	N(Grp 1)	N(Grp 2)	Information
1	50	33	17	17.5313
2	98	65	33	34.1996
3	146	97	49	50.8669
4	194	129	65	67.5338

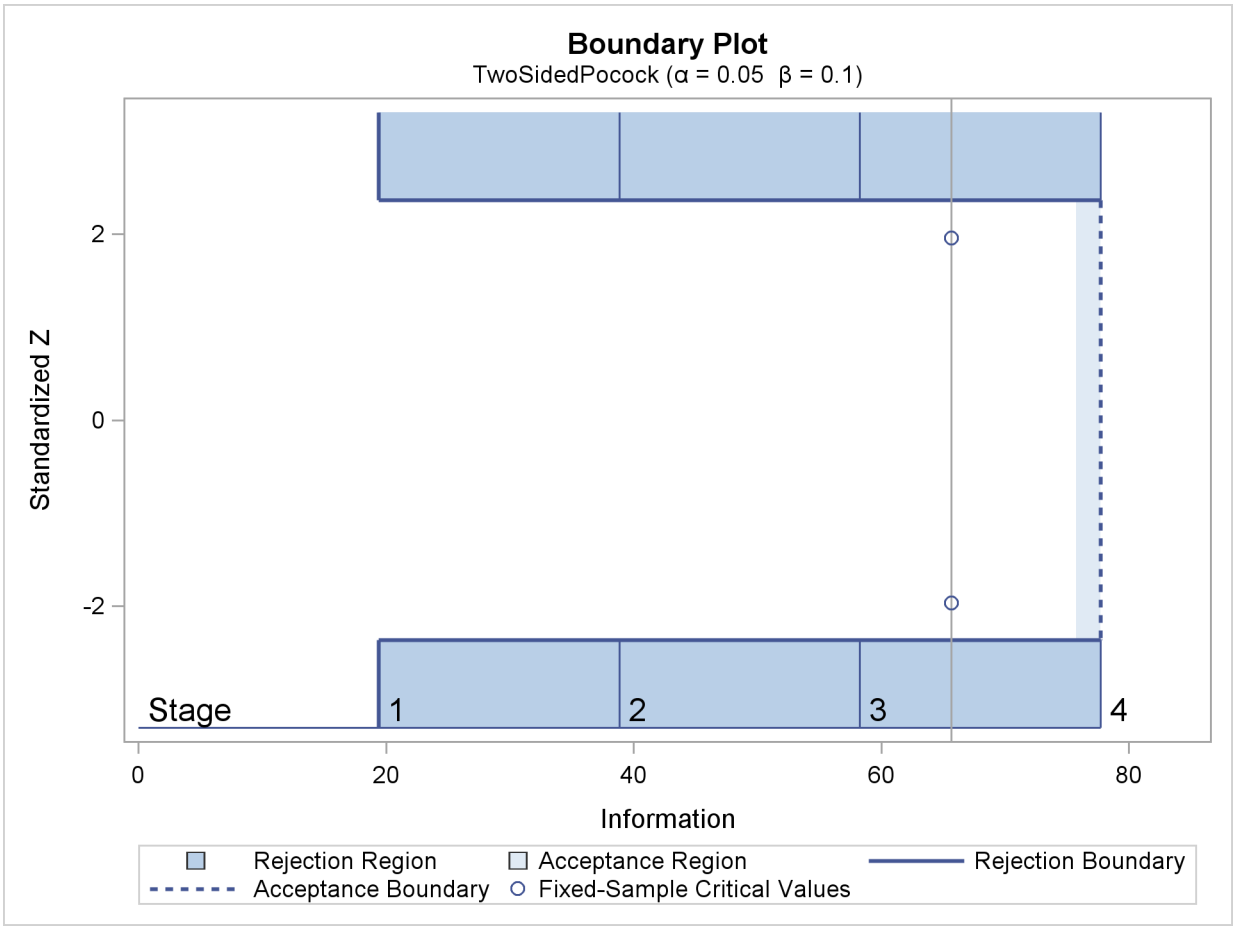
Example 77.4: Generating Graphics Display for Sequential Designs

This example creates the same group sequential design as in Example 77.3 and creates graphics by using ODS Graphics. The following statements request all available graphs in the SEQDESIGN procedure:

```
ods graphics on;
proc seqdesign altref=0.4
               plots=all
               ;
   TwoSidedPocock:         design nstages=4 method=poc;
   TwoSidedOBrienFleming: design nstages=4 method=obf;
run;
ods graphics off;
```

With the PLOTS=ALL option, a detailed boundary plot with the rejection region and acceptance region is displayed for the Pocock design, as shown in Output 77.4.1. With the default STOP=REJECT option, the rejection boundaries are also generated at interim stages.

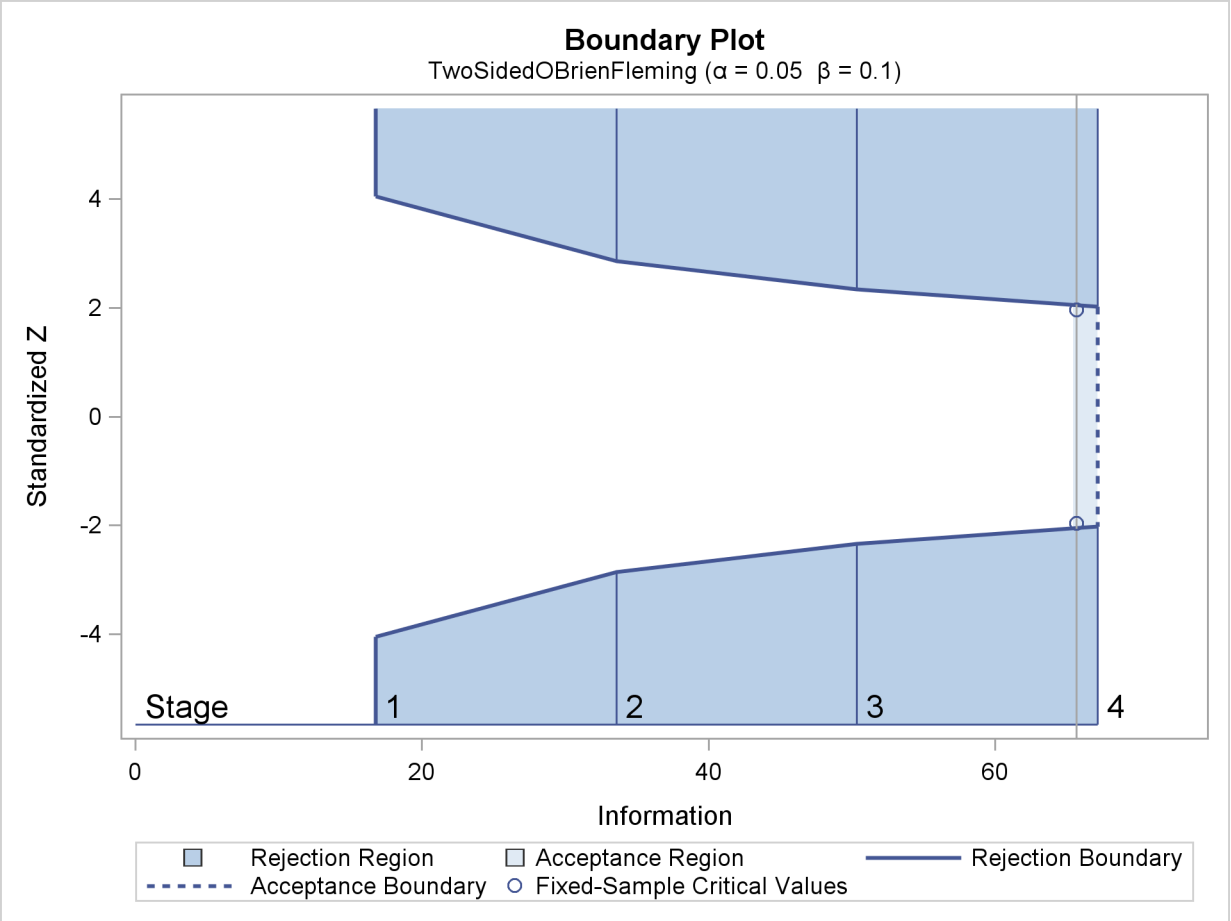
Output 77.4.1 Pocock Boundary Plot



The plot shows identical boundary values in each boundary in the standardized Z scale for the Pocock design. The information level and critical value for the corresponding fixed-sample design are also displayed.

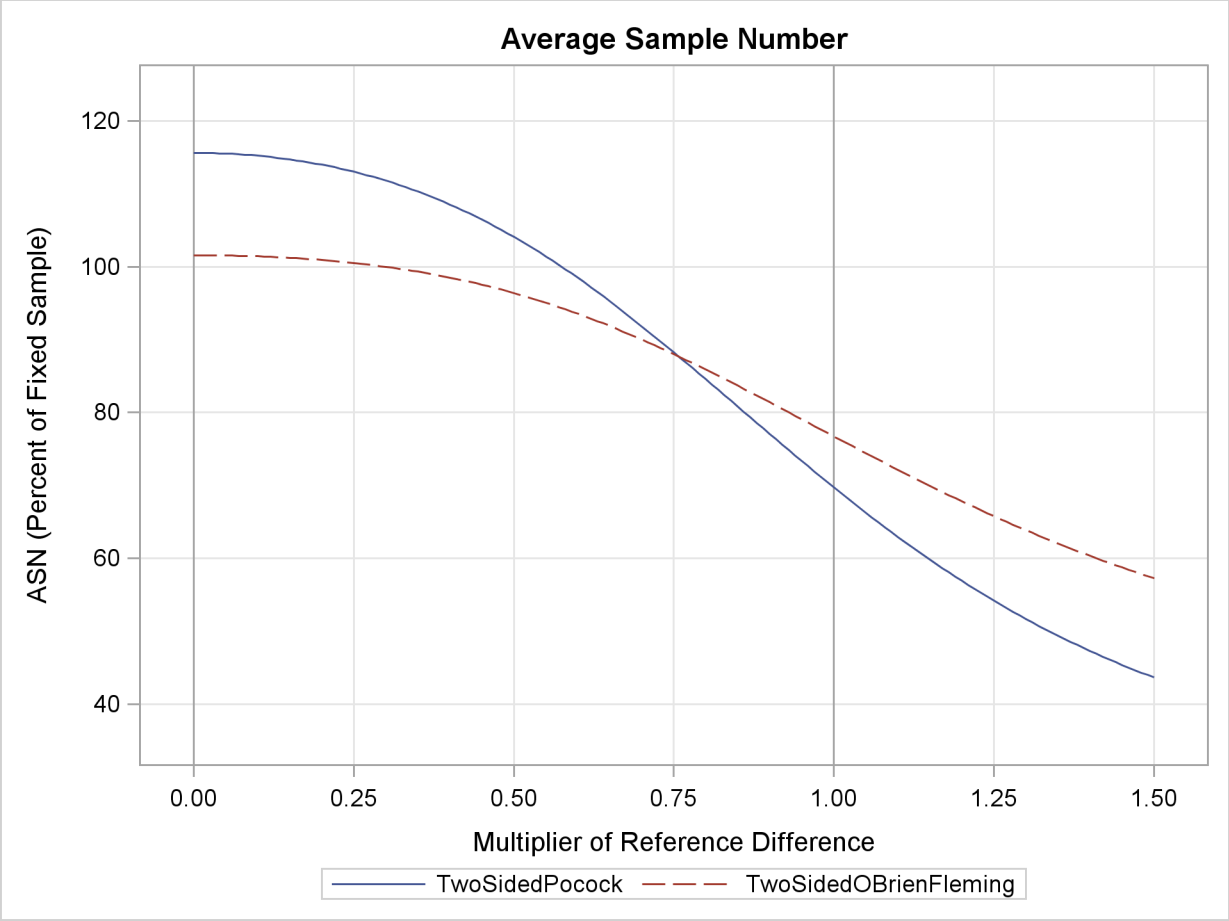
With the PLOTS=ALL option, a detailed boundary plot with the rejection region and acceptance region is also displayed for the O'Brien-Fleming design, as shown in Output 77.4.2. The plot shows that the rejection boundary values are decreasing as the trial advances in the standardized Z scale.

Output 77.4.2 O'Brien-Fleming Boundary Plot



With the PLOTS=ALL option, the procedure displays a plot of average sample numbers (expected sample sizes for nonsurvival data or expected numbers of events for survival data) under various hypothetical references for all designs simultaneously, as shown in Output 77.4.3. By default, the option CREF= $0, 0.01, 0.02, \ldots, 1.50$ and expected sample sizes under the hypothetical references $\theta = c_i\,\theta_1$ are displayed, where c_i are values specified in the CREF= option. These CREF= values are displayed on the horizontal axis.

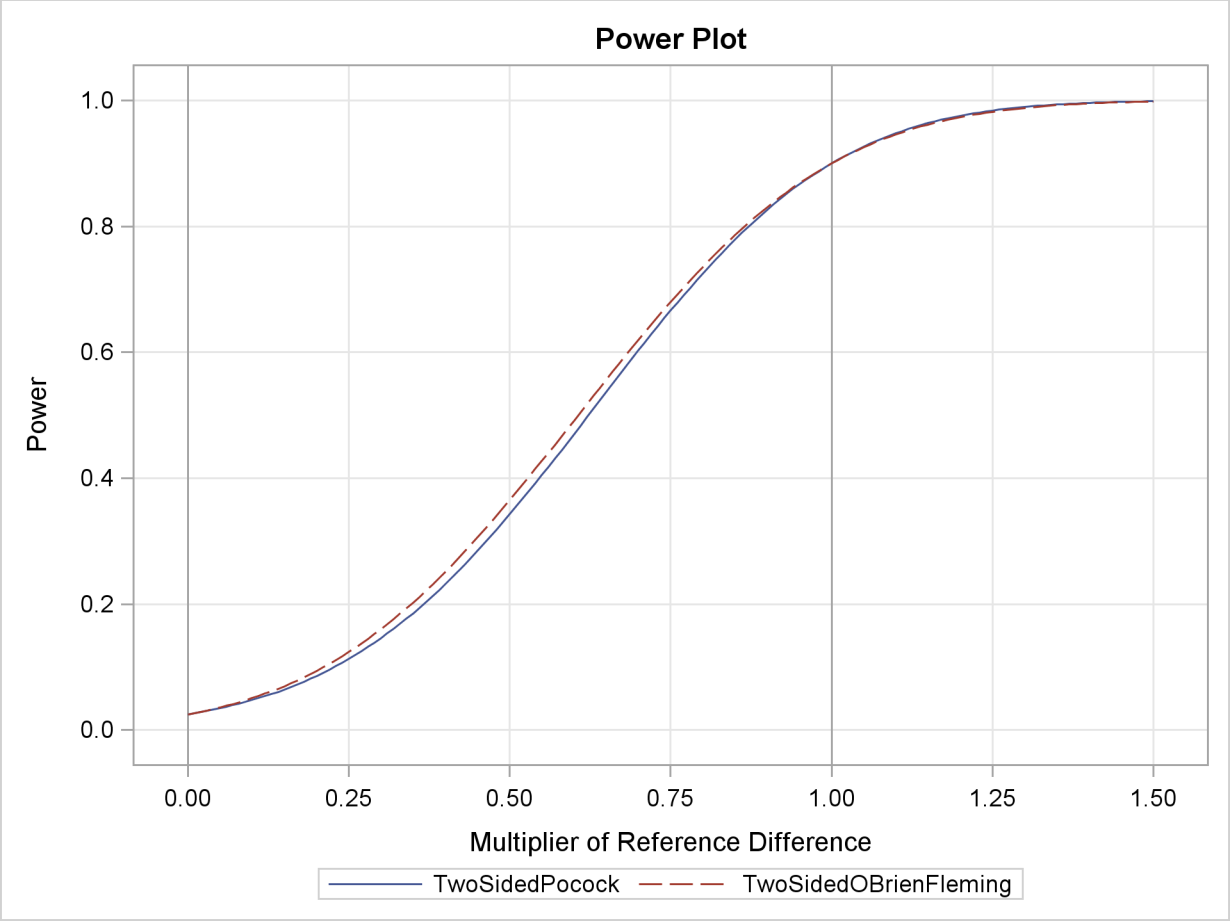
Output 77.4.3 ASN Plot



The plot shows that the Pocock design has a larger expected sample size than the O'Brien-Fleming design under the null hypothesis ($c_i = 0$) and has a smaller expected sample size under the alternative hypothesis ($c_i = 1$).

With the PLOTS=ALL option, the procedure displays a plot of the power curves under various hypothetical references for all designs simultaneously, as shown in Output 77.4.4. By default, the option CREF= $0, 0.01, 0.02, \ldots, 1.50$ and powers under hypothetical references $\theta = c_i\,\theta_1$ are displayed, where c_i are values specified in the CREF= option. These CREF= values are displayed on the horizontal axis.

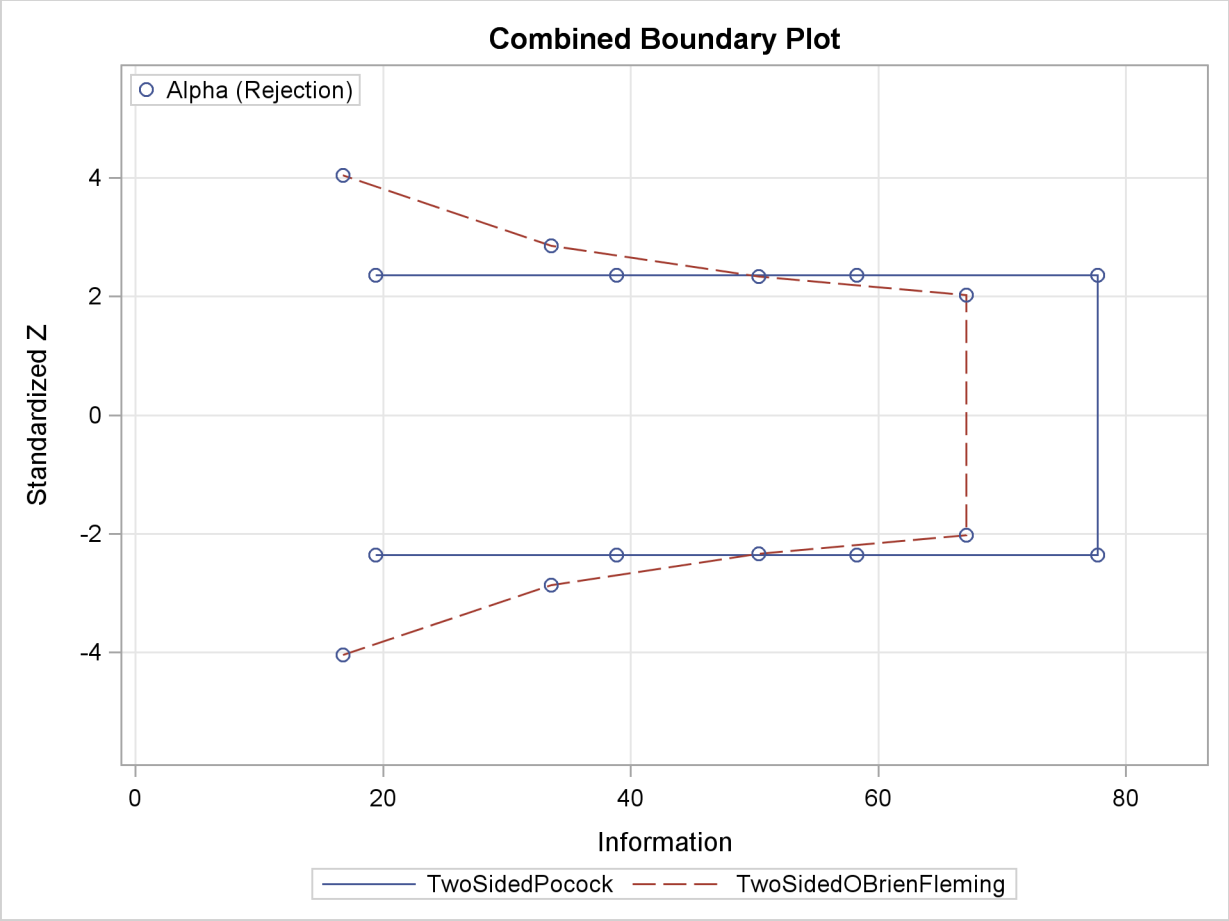
Output 77.4.4 Power Plot



Under the null hypothesis, $c_i = 0$, the power is 0.025, the upper Type I error probability. Under the alternative hypothesis, $c_i = 1$, the power is 0.9, one minus the Type II error probability. The plot shows only minor difference between the two designs.

With the PLOTS=ALL option, the procedure displays a plot of sequential boundaries for all designs simultaneously, as shown in Output 77.4.5. By default, the information levels are used on the horizontal axis.

Output 77.4.5 Combined Boundary Plot



The plot shows that the α boundary values (in absolute value) created from the O'Brien-Fleming method are greater in early stages and smaller in later stages than the boundary values from the Pocock method. The plot also shows that the information level in the Pocock design is larger than the corresponding level in the O'Brien-Fleming design at each stage.

With the PLOTS=ALL option, the procedure displays a plot of cumulative error spends for all boundaries in the designs simultaneously, as shown in Output 77.4.6. With a symmetric two-sided design, cumulative error spending is displayed only for the upper α boundary. The plot shows that for the upper α boundary, the O'Brien-Fleming method spends fewer errors in early stages and more errors in later stages than the corresponding Pocock method.

Output 77.4.6 Error Spending Plot



Example 77.5: Creating Designs Using Haybittle-Peto Methods

This example requests two 3-stage group sequential designs for normally distributed statistics. Each design uses a Haybittle-Peto method with a two-sided alternative and early stopping to reject the hypothesis. One design uses the specified interim boundary Z values and derives the final-stage boundary value for the specified α and β errors. The other design uses the specified boundary Z values and derives the overall α and β errors.

The following statements specify the interim boundary Z values and derive the final-stage boundary value for the specified $\alpha = 0.05$ and $\beta = 0.10$:

```
ods graphics on;
proc seqdesign altref=0.25
               errspend
               stopprob
               plots=errspend
               ;
   OneSidedPeto: design nstages=3
                 method=peto( z=3)
```

```
                    alt=upper   stop=reject
                    alpha=0.05  beta=0.10;
run;
ods graphics off;
```

The "Design Information" table in Output 77.5.1 displays design specifications and maximum information in percentage of its corresponding fixed-sample design.

Output 77.5.1 Haybittle-Peto Design Information

The SEQDESIGN Procedure
Design: OneSidedPeto

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Upper
Early Stop	Reject Null
Method	Haybittle-Peto
Boundary Key	Both
Alternative Reference	0.25
Number of Stages	3
Alpha	0.05
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	100.2466
Max Information	137.3592
Null Ref ASN (Percent of Fixed Sample)	100.1192
Alt Ref ASN (Percent of Fixed Sample)	87.35

The "Method Information" table in Output 77.5.2 displays the α and β errors and the derived drift parameter, which is the standardized alternative reference at the final stage.

Output 77.5.2 Method Information

Method Information

Boundary	Method	Alpha	Beta	Alternative Reference	Drift
Upper Alpha	Haybittle-Peto	0.05000	0.10000	0.25	2.930009

With the STOPPROB option, the "Expected Cumulative Stopping Probabilities" table in Output 77.5.3 displays the expected stopping stage and cumulative stopping probability to reject the null hypothesis at each stage under various hypothetical references $\theta = c_i\theta_1$, where θ_1 is the alternative reference and $c_i = 0, 0.5, 1, 1.5$ are the default values in the CREF= option.

Output 77.5.3 Stopping Probabilities

Expected Cumulative Stopping Probabilities
Reference = CRef * (Alt Reference)

CRef	Expected Stopping Stage	Source	----Stopping Probabilities--- Stage_1	Stage_2	Stage_3
0.0000	2.996	Reject Null	0.00135	0.00246	0.05000
0.5000	2.941	Reject Null	0.01561	0.04372	0.42762
1.0000	2.614	Reject Null	0.09538	0.29057	0.90000
1.5000	1.944	Reject Null	0.32185	0.73442	0.99698

The "Boundary Information" table in Output 77.5.4 displays information level, alternative references, and boundary values. The default BOUNDARYSCALE=STDZ option specifies that the standardized Z scale be used to display the alternative references and boundary values. The resulting standardized alternative reference at stage k is given by $\theta_1\sqrt{I_k}$, where θ_1 is the alternative reference and I_k is the information level at stage k, $k = 1, 2, 3$.

Output 77.5.4 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

Stage	---Information Level-- Proportion	Actual	-Alternative- --Reference-- Upper	-Boundary Values- ------Upper------ Alpha
1	0.3333	45.7864	1.69164	3.00000
2	0.6667	91.57281	2.39234	3.00000
3	1.0000	137.3592	2.93001	1.65042

At each interim stage, if the standardized statistic $z \geq 3$, the trial is stopped and the null hypothesis is rejected. If the statistic $z < 3$, the trial continues to the next stage. At the final stage, the null hypothesis is rejected if the statistic $z_3 > 1.65$. Otherwise, the hypothesis is accepted. Note that the boundary values at the final stage, 1.65, are close to the critical values 1.645 in the corresponding fixed-sample design.

The "Error Spending Information" in Output 77.5.5 displays cumulative error spending at each stage for each boundary. The stage 1 α spending 0.00135 corresponds to the one-sided p-value for a standardized Z statistic, $Z > 3$.

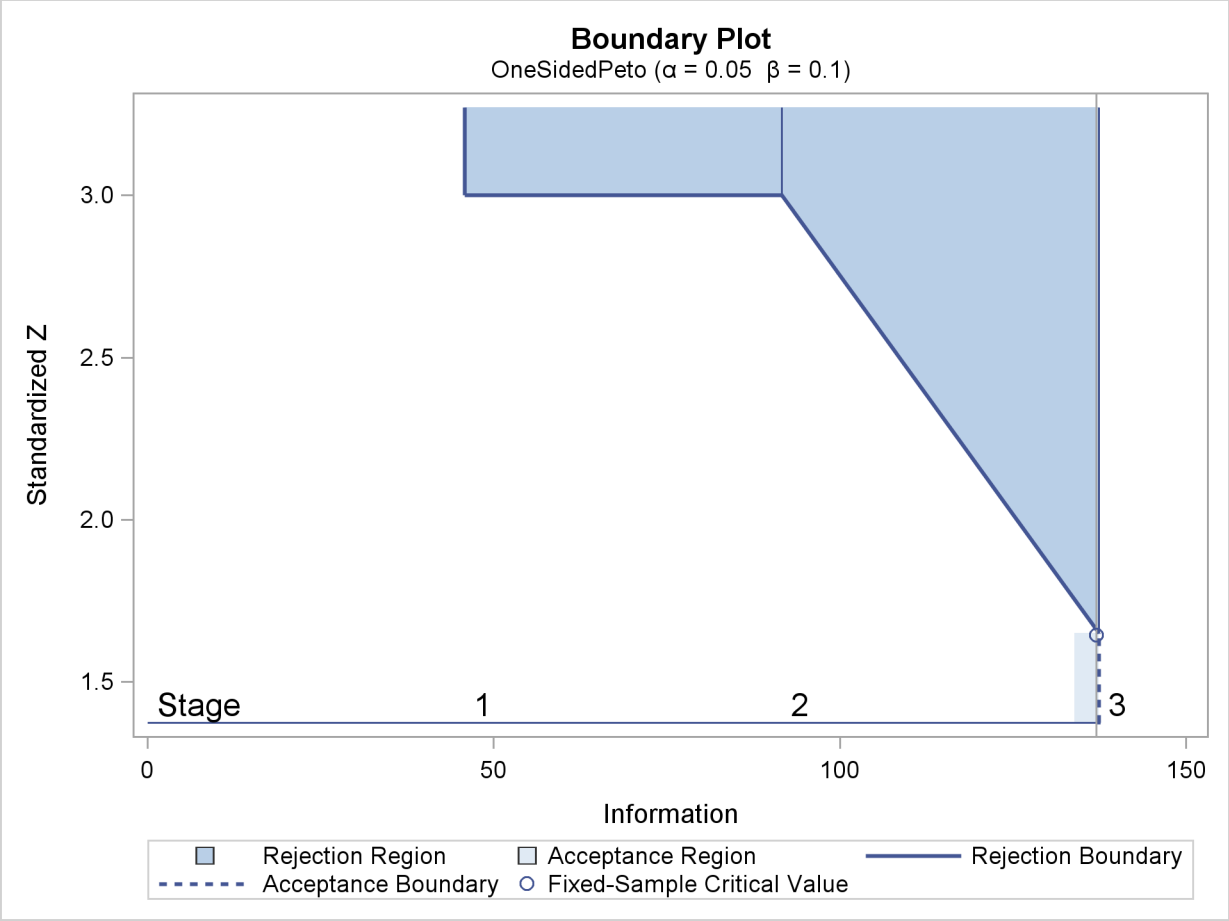
Output 77.5.5 Error Spending Information

Error Spending Information

Stage	-Information Level- Proportion	-Cumulative Error Spending- -----Upper----- Beta	Alpha
1	0.3333	0.00000	0.00135
2	0.6667	0.00000	0.00246
3	1.0000	0.10000	0.05000

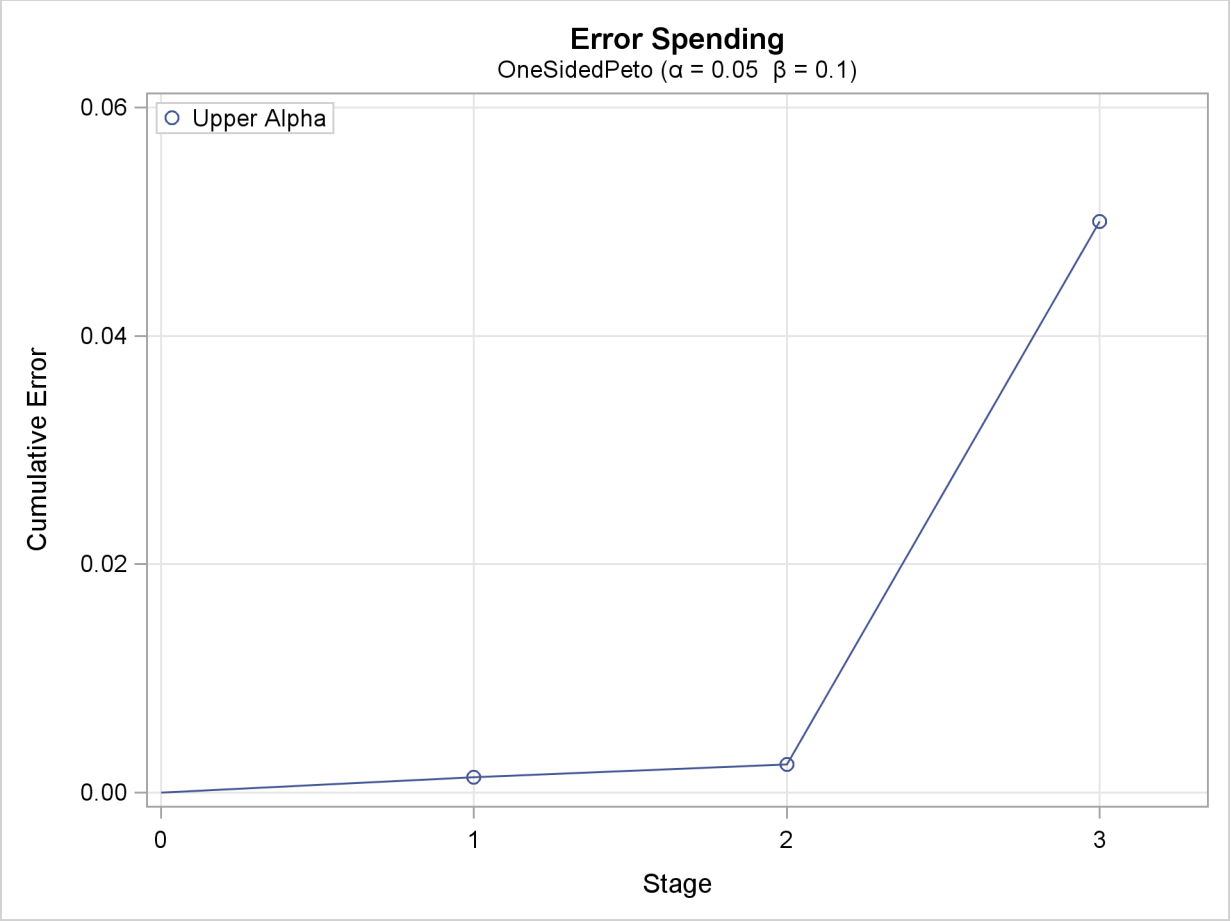
With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 77.5.6. With the STOP=REJECT option, the interim rejection boundaries are displayed.

Output 77.5.6 Boundary Plot



With the PLOTS=ERRSPEND option, the procedure displays a plot of error spending for each boundary, as shown in Output 77.5.7. The error spending values in the "Error Spending Information" in Output 77.5.4 are displayed in the plot. As expected, the error spending at each of the first two stages is small, with the standardized Z boundary value 3.

Output 77.5.7 Error Spending Plot



The following statements specify the boundary Z values and derive the α and β errors from these completely specified boundary values:

```
ods graphics on;
proc seqdesign altref=0.25
               maxinfo=200
               errspend
               stopprob
               plots=errspend
               ;
   OneSidedPeto: design nstages=3
                 method=peto(z=3 2.5 2)
                 alt=upper  stop=reject
                 boundarykey=none
                 ;
run;
ods graphics off;
```

The "Design Information" table in Output 77.5.8 displays design specifications and derived α and β error levels.

Output 77.5.8 Design Information

The SEQDESIGN Procedure
Design: OneSidedPeto

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Upper
Early Stop	Reject Null
Method	Haybittle-Peto
Boundary Key	None
Alternative Reference	0.25
Number of Stages	3
Alpha	0.02532
Beta	0.06035
Power	0.93965
Max Information (Percent of Fixed Sample)	101.6769
Max Information	200
Null Ref ASN (Percent of Fixed Sample)	101.3933
Alt Ref ASN (Percent of Fixed Sample)	73.74031

The "Method Information" table in Output 77.5.9 displays the α and β errors and the derived drift parameter for each boundary.

Output 77.5.9 Method Information

Method Information

Boundary	Method	Alpha	Beta	Alternative Reference	Drift
Upper Alpha	Haybittle-Peto	0.02532	0.06035	0.25	3.535534

With the STOPPROB option, the "Expected Cumulative Stopping Probabilities" table in Output 77.5.10 displays the expected stopping stage and cumulative stopping probability to reject the null hypothesis at each stage under various hypothetical references $\theta = c_i\theta_1$, where θ_1 is the alternative reference and $c_i = 0, 0.5, 1, 1.5$ are the default values in the CREF= option.

Output 77.5.10 Stopping Probabilities

Expected Cumulative Stopping Probabilities
Reference = CRef * (Alt Reference)

CRef	Expected Stopping Stage	Source	----Stopping Probabilities--- Stage_1	Stage_2	Stage_3
0.0000	2.992	Reject Null	0.00135	0.00702	0.02532
0.5000	2.826	Reject Null	0.02389	0.15030	0.41775
1.0000	2.176	Reject Null	0.16884	0.65544	0.93965
1.5000	1.508	Reject Null	0.52466	0.96708	0.99954

The "Boundary Information" table in Output 77.5.11 displays information level, alternative references, and boundary values.

Output 77.5.11 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

Stage	Information Level Proportion	Information Level Actual	Alternative Reference Upper	Boundary Values Upper Alpha
1	0.3333	66.66667	2.04124	3.00000
2	0.6667	133.3333	2.88675	2.50000
3	1.0000	200	3.53553	2.00000

The "Error Spending Information" in Output 77.5.12 displays cumulative error spending at each stage for each boundary. The first-stage α spending 0.00135 corresponds to the one-sided p-value for a standardized Z statistic, $Z > 3$.

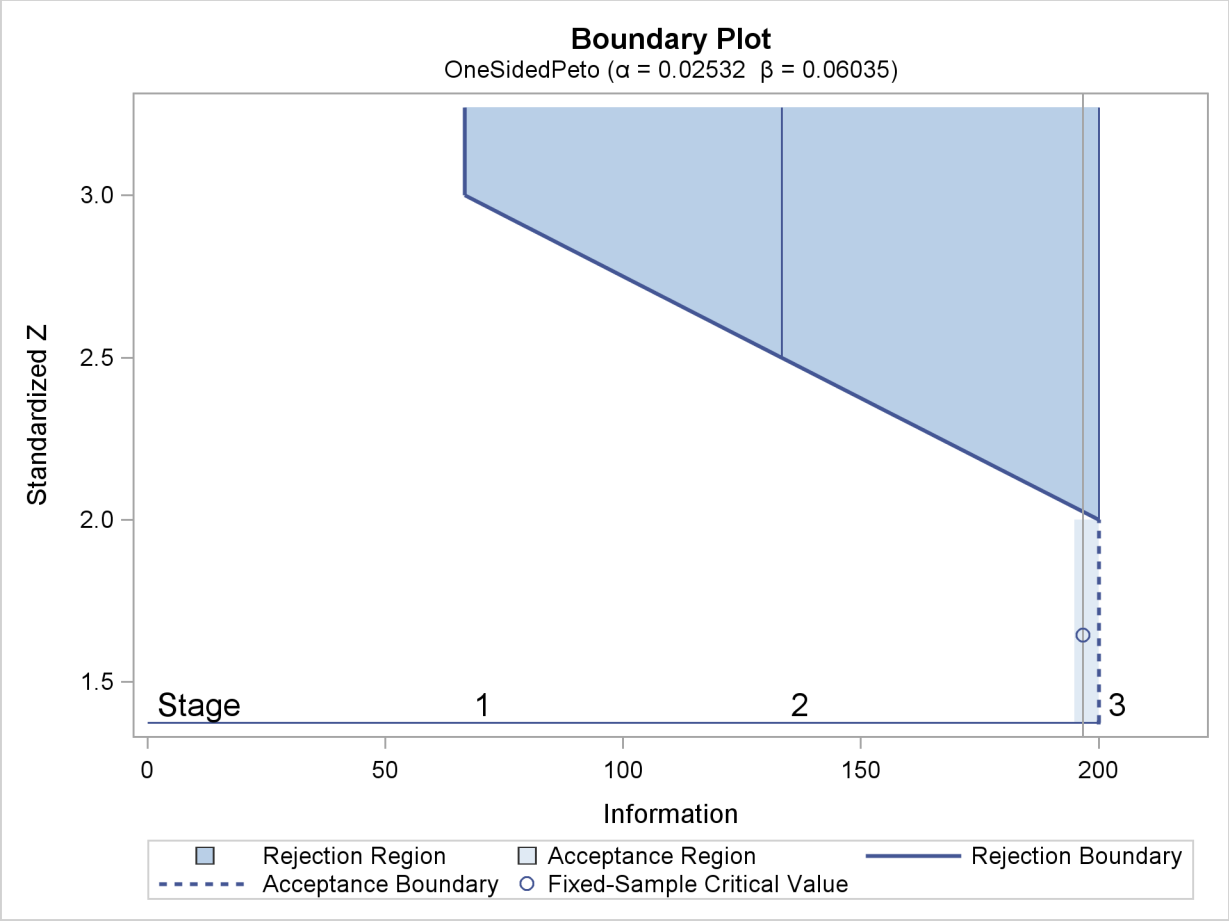
Output 77.5.12 Error Spending Information

Error Spending Information

Stage	Information Level Proportion	Cumulative Error Spending Upper Beta	Cumulative Error Spending Upper Alpha
1	0.3333	0.00000	0.00135
2	0.6667	0.00000	0.00702
3	1.0000	0.06035	0.02532

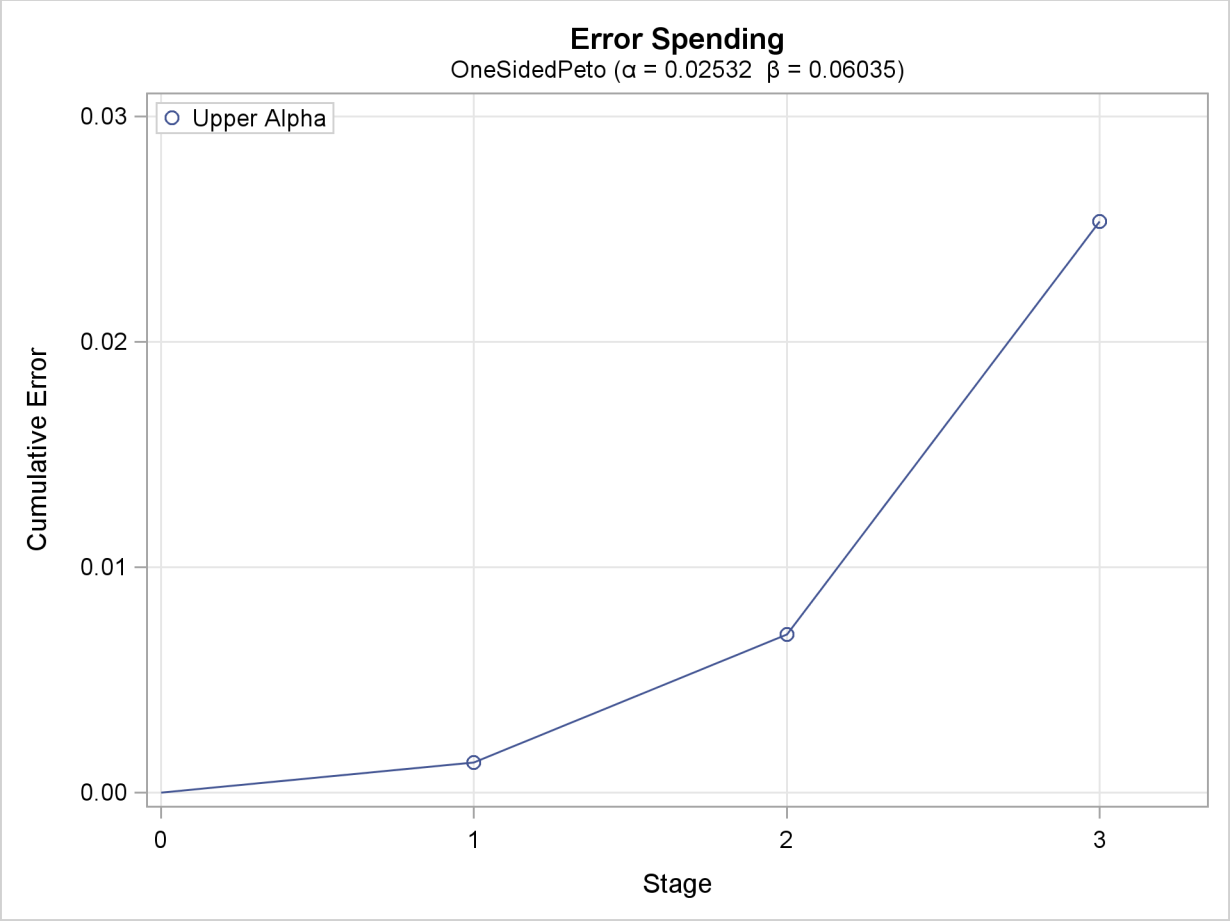
With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 77.5.13. With the STOP=REJECT option, the interim rejection boundaries are displayed.

Output 77.5.13 Boundary Plot



With the PLOTS=ERRSPEND option, the procedure displays a plot of error spending for each boundary, as shown in Output 77.5.14. The error spending values in the "Error Spending Information" table in Output 77.5.10 are displayed in the plot.

Output 77.5.14 Error Spending Plot



Example 77.6: Creating Designs with Various Stopping Criteria

This example requests three 5-stage group sequential designs for normally distributed statistics. Each design uses a triangular method with the specified one-sided upper alternative reference $\theta_1 = 0.2$. The resulting boundary values are displayed with the score scale. Note that these unified family triangular designs are different from Whitehead's triangular designs.

The following statements request three designs with different stopping criterion:

```
ods graphics on;
proc seqdesign altref=0.2
               bscale=score
               errspend
               plots=(combinedboundary errspend(hscale=info))
               ;
   StopToRejectAccept: design nstages=5 method=tri alt=upper stop=both;
   StopToReject:       design nstages=5 method=tri alt=upper stop=reject;
   StopToAccept:       design nstages=5 method=tri alt=upper stop=accept;
run;
ods graphics off;
```

The first design has early stopping to reject or accept the null hypothesis H_0.

The "Design Information" table in Output 77.6.1 displays design specifications and derived statistics. With the specified alternative reference, the maximum information is derived.

Output 77.6.1 Triangular Design Information

The SEQDESIGN Procedure
Design: StopToRejectAccept

Design Information

Statistic Distribution	Normal
Boundary Scale	Score
Alternative Hypothesis	Upper
Early Stop	Accept/Reject Null
Method	Triangular
Boundary Key	Both
Alternative Reference	0.2
Number of Stages	5
Alpha	0.05
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	140.0293
Max Information	299.797
Null Ref ASN (Percent of Fixed Sample)	59.11973
Alt Ref ASN (Percent of Fixed Sample)	66.94909

The "Method Information" table in Output 77.6.2 displays the α and β errors and the derived drift parameter, which is the standardized alternative reference at the final stage. The table also shows the corresponding parameters for a triangular method as a unified family method.

Output 77.6.2 Method Information

Method Information

				Unified Family		
Boundary	Method	Alpha	Beta	Rho	Tau	C
Upper Alpha	Triangular	0.05000	.	0.5	1	0.94394
Upper Beta	Triangular	.	0.10000	0.5	1	0.78753

Method Information

Boundary	Alternative Reference	Drift
Upper Alpha	0.2	3.46293
Upper Beta	0.2	3.46293

The "Boundary Information" table in Output 77.6.3 displays information level, alternative reference, and boundary values. With the specified BOUNDARYSCALE=SCORE option, the alternative reference and boundary values are displayed in the score statistic scale. With a score scale, the alternative reference is $\theta_1 I_k$, where θ_1 is the specified alternative reference and I_k is the information level at stage k, $k = 1, 2, \ldots, 5$.

Output 77.6.3 Boundary Information

Boundary Information (Score Scale)
Null Reference = 0

Stage	Information Level Proportion	Information Level Actual	Alternative Reference Upper	Boundary Values Upper Beta	Boundary Values Upper Alpha
1	0.2000	59.9594	11.99188	-4.37102	19.61274
2	0.4000	119.9188	23.98376	4.89371	22.88154
3	0.6000	179.8782	35.97564	14.15845	26.15033
4	0.8000	239.8376	47.96752	23.42318	29.41912
5	1.0000	299.797	59.95940	32.68791	32.68791

The "Error Spending Information" table in Output 77.6.4 displays cumulative error spending at each stage for each boundary.

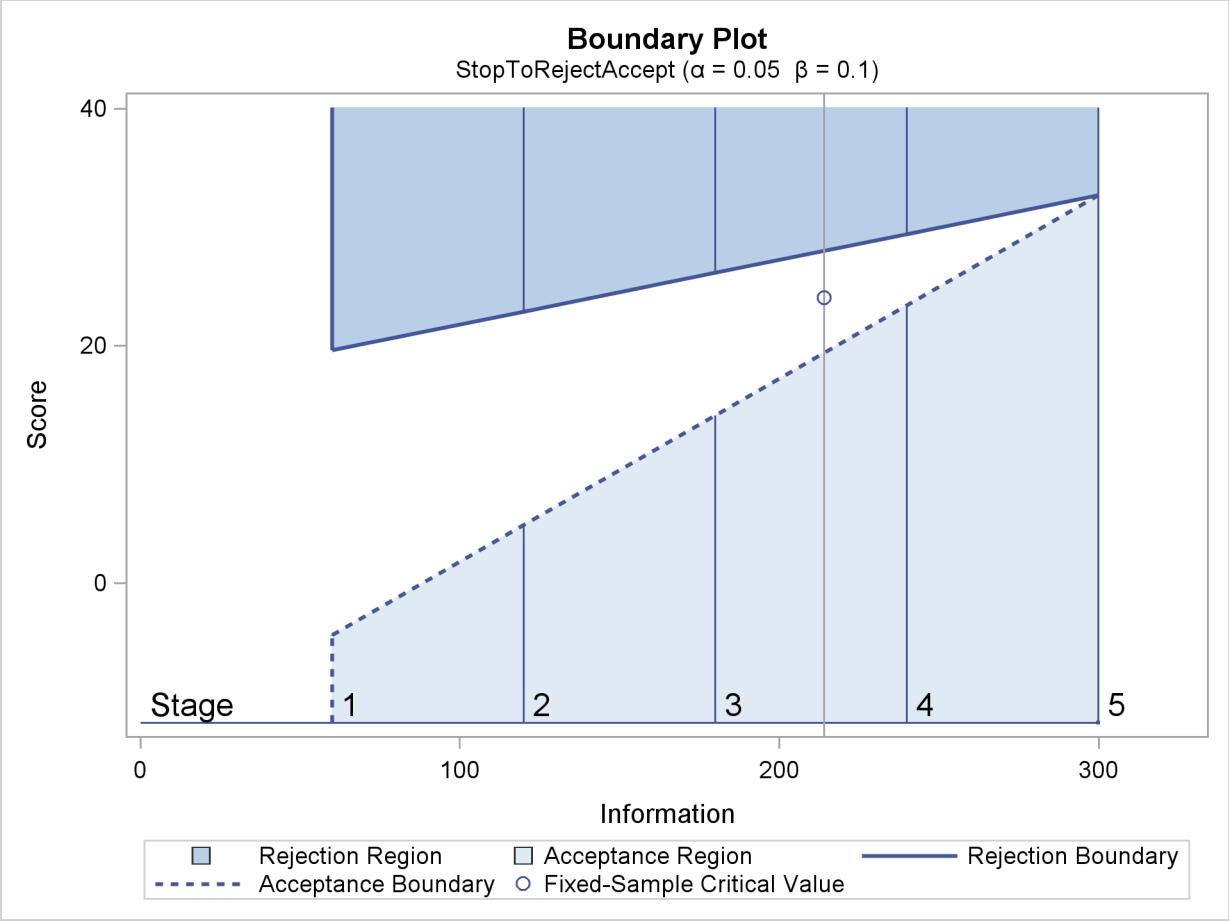
Output 77.6.4 Error Spending Information

Error Spending Information

Stage	Information Level Proportion	Cumulative Error Spending Upper Beta	Cumulative Error Spending Upper Alpha
1	0.2000	0.01729	0.00566
2	0.4000	0.04927	0.02138
3	0.6000	0.07611	0.03643
4	0.8000	0.09357	0.04641
5	1.0000	0.10000	0.05000

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 77.6.5. With the STOP=BOTH option, both the acceptance and rejection boundaries at interim stages are displayed. With the score scale, the acceptance and rejection boundaries are straight lines and form a triangular-shape continuation region.

Output 77.6.5 Boundary Plot with Score Statistics



The second design has early stopping only to reject the null hypothesis H_0.

The "Design Information" table in Output 77.6.6 displays design specifications and derived statistics. With the specified alternative reference, the maximum information is derived.

Output 77.6.6 Triangular Design Information

The SEQDESIGN Procedure
Design: StopToReject

Design Information

Statistic Distribution	Normal
Boundary Scale	Score
Alternative Hypothesis	Upper
Early Stop	Reject Null
Method	Triangular
Boundary Key	Both
Alternative Reference	0.2
Number of Stages	5
Alpha	0.05
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	113.4443
Max Information	242.8799
Null Ref ASN (Percent of Fixed Sample)	111.3399
Alt Ref ASN (Percent of Fixed Sample)	67.41968

The "Method Information" table in Output 77.6.7 displays the α and β errors and the derived drift parameter. The table also shows the corresponding parameters for a triangular method as a unified family method.

Output 77.6.7 Method Information

Method Information

				Unified Family		
Boundary	Method	Alpha	Beta	Rho	Tau	C
Upper Alpha	Triangular	0.05000	0.10000	0.5	1	0.9833

Method Information

Boundary	Alternative Reference	Drift
Upper Alpha	0.2	3.116921

The "Boundary Information" table in Output 77.6.8 displays information level, alternative reference, and boundary values. With the specified BOUNDARYSCALE=SCORE option, the alternative reference and boundary values are displayed in the score statistic scale.

Output 77.6.8 Boundary Information

Boundary Information (Score Scale)
Null Reference = 0

Stage	---Information Level-- Proportion	Actual	-Alternative- --Reference-- Upper	-Boundary Values- ------Upper------ Alpha
1	0.2000	48.57597	9.71519	18.38919
2	0.4000	97.15194	19.43039	21.45405
3	0.6000	145.7279	29.14558	24.51891
4	0.8000	194.3039	38.86078	27.58378
5	1.0000	242.8799	48.57597	30.64864

The "Error Spending Information" table in Output 77.6.9 displays cumulative error spending at each stage for each boundary.

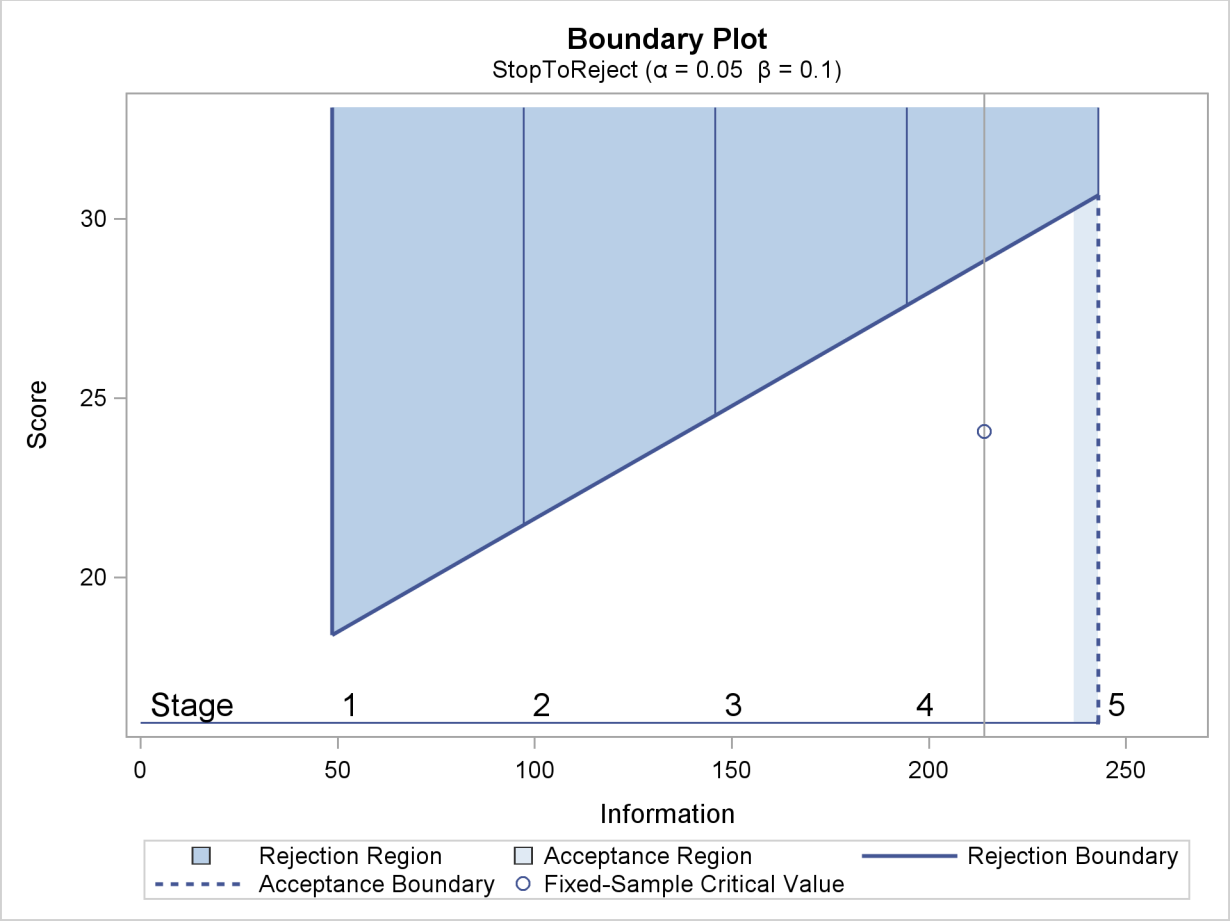
Output 77.6.9 Error Spending Information

Error Spending Information

Stage	-Information Level- Proportion	-Cumulative Error Spending- ----------Upper---------- Beta	Alpha
1	0.2000	0.00000	0.00416
2	0.4000	0.00000	0.01705
3	0.6000	0.00000	0.03027
4	0.8000	0.00000	0.04127
5	1.0000	0.10000	0.05000

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 77.6.10. For a triangular design, these rejection boundaries form a straight line with the score scale.

Output 77.6.10 Boundary Plot with Score Statistics



The third design has early stopping to accept the null hypothesis H_0.

The "Design Information" table in Output 77.6.11 displays design specifications and derived statistics. With the specified alternative reference, the maximum information is derived.

Output 77.6.11 Triangular Design Information

The SEQDESIGN Procedure
Design: StopToAccept

Design Information

Statistic Distribution	Normal
Boundary Scale	Score
Alternative Hypothesis	Upper
Early Stop	Accept Null
Method	Triangular
Boundary Key	Both
Alternative Reference	0.2
Number of Stages	5
Alpha	0.05
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	114.9925
Max Information	246.1945
Null Ref ASN (Percent of Fixed Sample)	57.83208
Alt Ref ASN (Percent of Fixed Sample)	110.2477

The "Method Information" table in Output 77.6.12 displays the α and β errors and the derived drift parameter. The table also shows the corresponding parameters for a triangular method as a unified family method.

Output 77.6.12 Method Information

Method Information

				Unified Family		
Boundary	Method	Alpha	Beta	Rho	Tau	C
Upper Beta	Triangular	0.05000	0.10000	0.5	1	0.82154

Method Information

Boundary	Alternative Reference	Drift
Upper Beta	0.2	3.138117

The "Boundary Information" table in Output 77.6.13 displays information level, alternative reference, and boundary values. With the specified BOUNDARYSCALE=SCORE option, the alternative reference and boundary values are displayed in the score statistic scale.

Output 77.6.13 Boundary Information

Boundary Information (Score Scale)
Null Reference = 0

Stage	Information Level Proportion	Information Level Actual	Alternative Reference Upper	Boundary Values Upper Beta
1	0.2000	49.2389	9.84778	-5.62074
2	0.4000	98.4778	19.69556	1.64895
3	0.6000	147.7167	29.54334	8.91865
4	0.8000	196.9556	39.39112	16.18834
5	1.0000	246.1945	49.23890	23.45803

The "Error Spending Information" table in Output 77.6.14 displays cumulative error spending at each stage for each boundary.

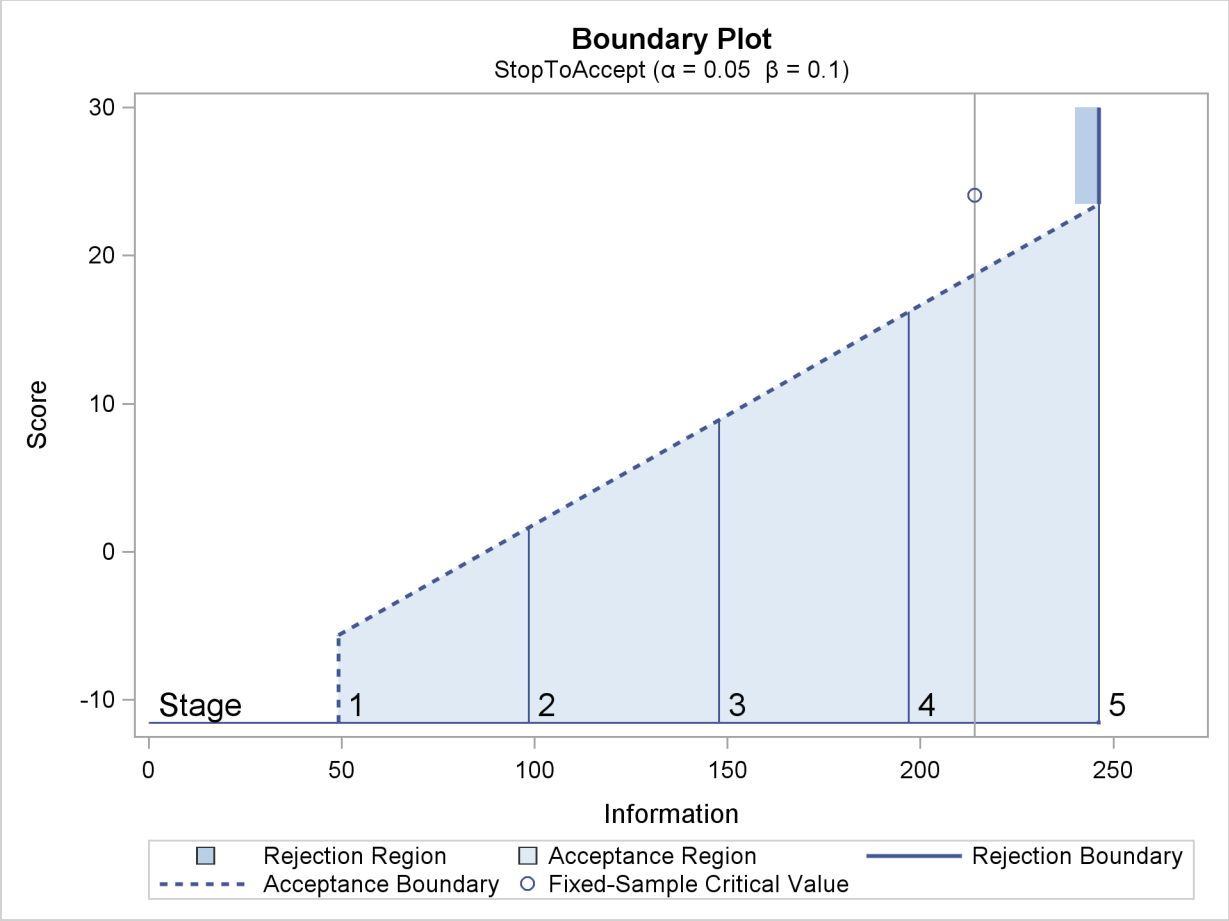
Output 77.6.14 Error Spending Information

Error Spending Information

Stage	Information Level Proportion	Cumulative Error Spending Upper Beta	Cumulative Error Spending Upper Alpha
1	0.2000	0.01375	0.00000
2	0.4000	0.04149	0.00000
3	0.6000	0.06594	0.00000
4	0.8000	0.08513	0.00000
5	1.0000	0.10000	0.05000

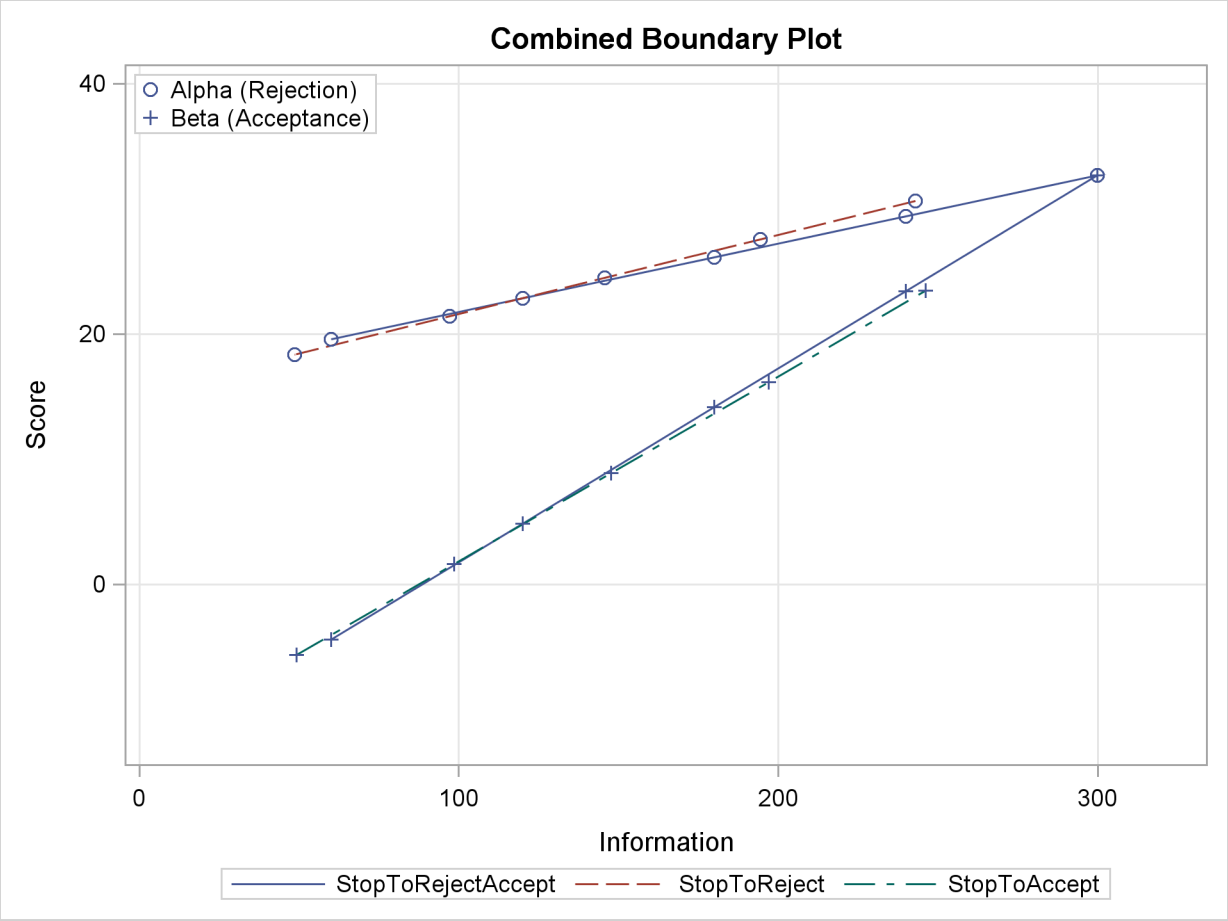
With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 77.6.15. For a triangular design, these rejection boundaries form a straight line with the score scale.

Output 77.6.15 Boundary Plot with Score Scale



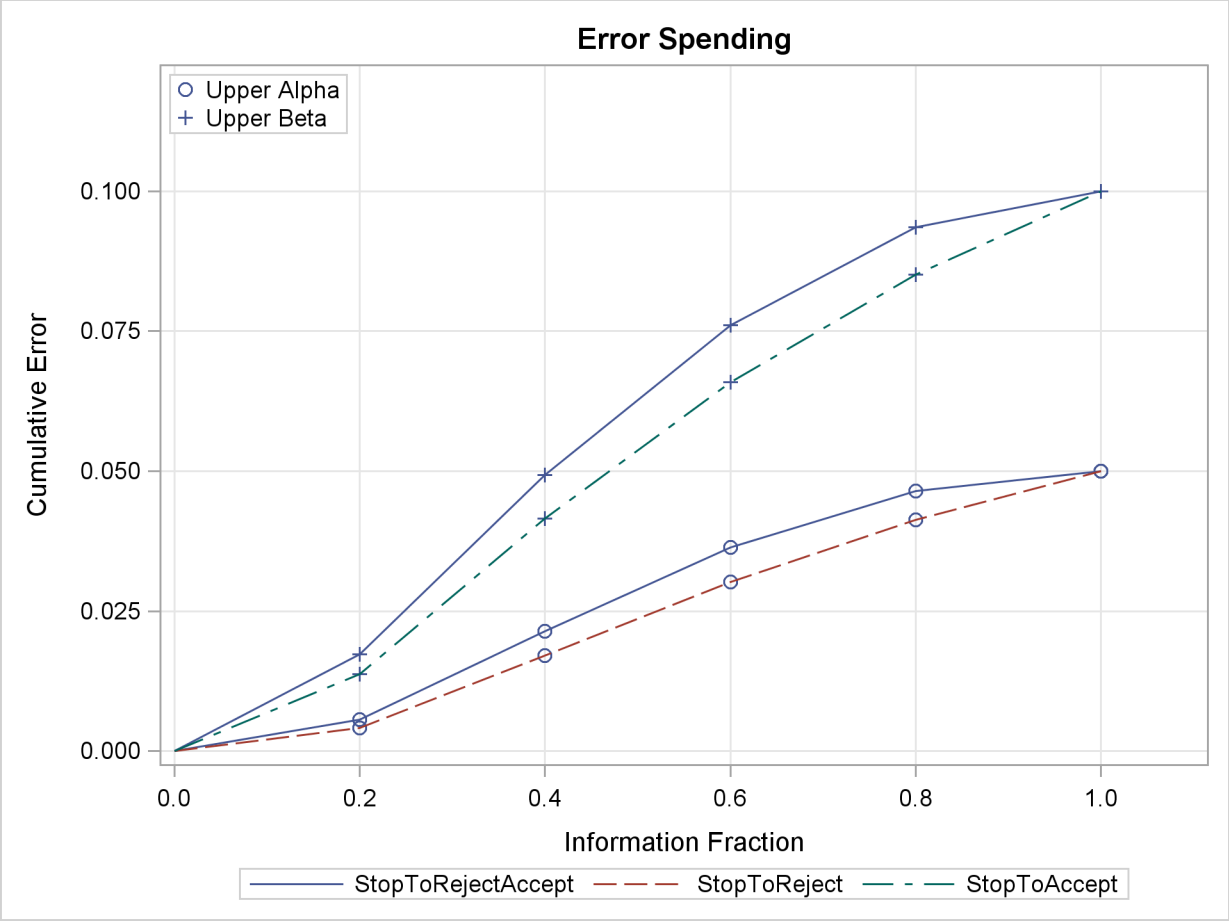
With the PLOTS=COMBINEDBOUNDARY option, a plot of the resulting sequential boundaries for all designs is displayed, as shown in Output 77.6.16. The plot shows that the design with early stopping to reject and to accept H_0 has larger maximum information than the other two designs.

Output 77.6.16 Combined Boundary Plot with Score Scale



With the PLOTS=ERRSPEND(HSCALE=INFO) option, the error spending plot is displayed with the information level on the horizontal axis, as shown in Output 77.6.17. The design with early stopping to reject or accept the null hypothesis H_0 has larger α spending and larger β spending in early stages than the other two designs.

Output 77.6.17 Error Spending Plot



Example 77.7: Creating Whitehead's Triangular Designs

This example requests three 4-stage Whitehead's triangular designs for normally distributed statistics. Each design has a one-sided alternative hypothesis with early stopping to reject or accept the null hypothesis H_0. Note that Whitehead's triangular designs are different from unified family triangular designs.

Suppose that a clinic is conducting a study of the effect of a new cancer treatment. The study consists of exposing mice to a carcinogen and randomly assigning them to either the control group or the treatment group. The event of interest is death from cancer induced by the carcinogen, and the response is the time from randomization to death.

Following the derivations in the section "Test for Two Survival Distributions with a Log-Rank Test" on page 5866, the hypothesis $H_0 : \theta = -\log(\lambda) = 0$ with an alternative hypothesis $H_1 : \theta = \theta_1 > 0$ is used, where λ is the hazard ratio between the treatment group and the control group.

Also suppose that from past experience, the median survival time for the control group is $t_0 = 20$ days, and the study wants to detect a $t_1 = 40$ days' median survival time with a 80% power in the trial. Assuming exponential survival functions for the two groups, the hazard rates can be computed

from

$$S_j(t_j) = e^{-h_j t_j} = \frac{1}{2}$$

where $j = 0, 1$.

Thus, with $h_0 = 0.0346574$ and $h_1 = 0.0173287$, the hazard ratio $\lambda_1 = h_1/h_0 = 1/2$, and the alternative reference is

$$\theta_1 = -\log(\lambda_1) = -\log(\frac{1}{2}) = 0.693147$$

The following statements invoke the SEQDESIGN procedure and specify three Whitehead's triangular designs:

```
ods graphics on;
proc seqdesign altref=0.693147
               bscale=score
               plots=combinedboundary
               ;
   BoundaryKeyNone:  design nstages=4
                            method=whitehead
                            boundarykey=none
                            alt=upper   stop=both
                            alpha=0.05  beta=0.20
                            ;
   BoundaryKeyAlpha: design nstages=4
                            method=whitehead
                            boundarykey=alpha
                            alt=upper   stop=both
                            alpha=0.05  beta=0.20
                            ;
   BoundaryKeyBeta:  design nstages=4
                            method=whitehead
                            boundarykey=beta
                            alt=upper   stop=both
                            alpha=0.05  beta=0.20
                            ;
run;
ods graphics off;
```

Whitehead methods with early stopping to reject or accept the null hypothesis create boundaries that approximately satisfy the Type I and Type II error probability specification. The BOUNDARYKEY=NONE option specifies no adjustment to the boundary value at the final stage to maintain either a Type I or a Type II error probability level.

The "Design Information" table in Output 77.7.1 displays design specifications and maximum information. Note that with the BOUNDARYKEY=NONE option, the derived errors $\alpha = 0.05071$ and $\beta = 0.19771$ are not the same as the specified errors $\alpha = 0.05$ and $\beta = 0.20$.

Output 77.7.1 Whitehead Design Information

The SEQDESIGN Procedure
Design: BoundaryKeyNone

Design Information

Statistic Distribution	Normal
Boundary Scale	Score
Alternative Hypothesis	Upper
Early Stop	Accept/Reject Null
Method	Whitehead
Boundary Key	None
Alternative Reference	0.693147
Number of Stages	4
Alpha	0.05071
Beta	0.19771
Power	0.80229
Max Information (Percent of Fixed Sample)	129.6815
Max Information	16.70639
Null Ref ASN (Percent of Fixed Sample)	62.48184
Alt Ref ASN (Percent of Fixed Sample)	73.82535

The "Method Information" table in Output 77.7.2 displays the derived α and β errors and the derived drift parameter. The derived errors $\alpha = 0.05071$ and $\beta = 0.19771$ are not exactly the same as the specified errors $\alpha = 0.05$ and $\beta = 0.20$ with the BOUNDARYKEY=NONE option.

Output 77.7.2 Method Information

Method Information

				Whitehead	
Boundary	Method	Alpha	Beta	Tau	C
Upper Alpha	Whitehead	0.05071	.	0.25	4.60517
Upper Beta	Whitehead	.	0.19771	0.25	4.60517

Method Information

Boundary	Alternative Reference	Drift
Upper Alpha	0.693147	2.833131
Upper Beta	0.693147	2.833131

The "Boundary Information" table in Output 77.7.3 displays information level, alternative reference, and boundary values. With the specified BOUNDARYSCALE=SCORE option, the alternative reference and boundary values are displayed with the score statistics scale.

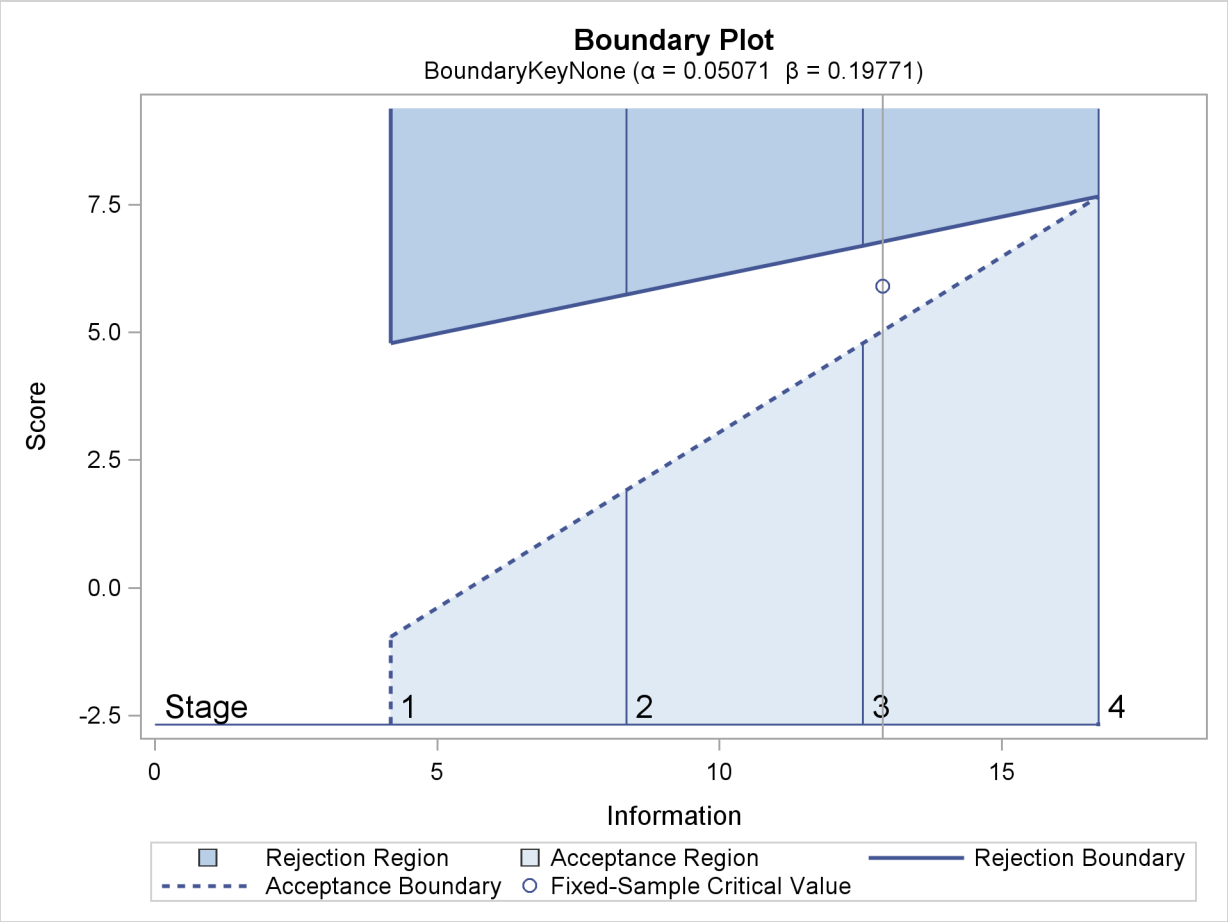
Output 77.7.3 Boundary Information

Boundary Information (Score Scale)
Null Reference = 0

Stage	---Information Level-- Proportion	Actual	-Alternative- --Reference-- Upper	-----Boundary Values---- ----------Upper---------- Beta	Alpha
1	0.2500	4.176597	2.89500	-0.95755	4.78775
2	0.5000	8.353195	5.78999	1.91510	5.74530
3	0.7500	12.52979	8.68499	4.78775	6.70285
4	1.0000	16.70639	11.57998	7.66039	7.66039

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 77.7.4.

Output 77.7.4 Boundary Plot



The second design uses the BOUNDARYKEY=ALPHA option to adjust the boundary value at the final stage to maintain the Type I error probability level.

The "Design Information" table in Output 77.7.5 displays design specifications and the derived maximum information. Note that with the BOUNDARYKEY=ALPHA option, the specified Type I error probability $\alpha = 0.05$ is maintained.

Output 77.7.5 Whitehead Design Information

The SEQDESIGN Procedure
Design: BoundaryKeyAlpha

Design Information

Statistic Distribution	Normal
Boundary Scale	Score
Alternative Hypothesis	Upper
Early Stop	Accept/Reject Null
Method	Whitehead
Boundary Key	Alpha
Alternative Reference	0.693147
Number of Stages	4
Alpha	0.05
Beta	0.20044
Power	0.79956
Max Information (Percent of Fixed Sample)	129.9894
Max Information	16.70639
Null Ref ASN (Percent of Fixed Sample)	62.6302
Alt Ref ASN (Percent of Fixed Sample)	74.00064

The "Method Information" table in Output 77.7.6 displays the specified and derived α and β errors and the derived drift parameter. The derived Type I error probability is the same as the specified $\alpha = 0.05$ and the derived Type II error probability $\beta = 0.20044$ is not the same as the specified $\beta = 0.20$ with the BOUNDARYKEY=ALPHA option.

Output 77.7.6 Method Information

Method Information

				------Whitehead-----	
Boundary	Method	Alpha	Beta	Tau	C
Upper Alpha	Whitehead	0.05000	.	0.25	4.60517
Upper Beta	Whitehead	.	0.20044	0.25	4.60517

Method Information

Boundary	Alternative Reference	Drift
Upper Alpha	0.693147	2.833131
Upper Beta	0.693147	2.833131

The "Boundary Information" table in Output 77.7.7 displays information level, alternative reference, and boundary values.

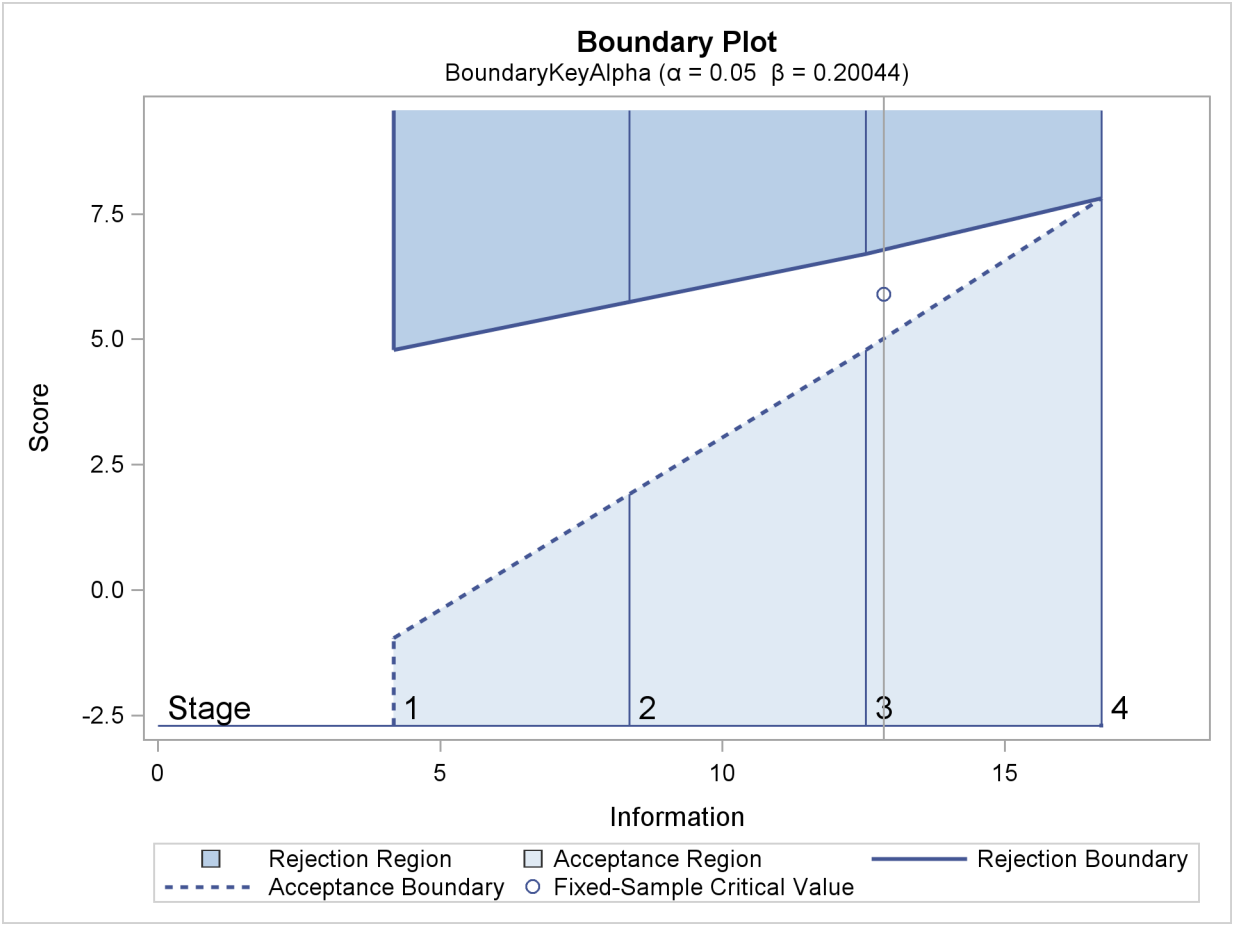
Output 77.7.7 Boundary Information

Boundary Information (Score Scale)
Null Reference = 0

Stage	---Information Level-- Proportion	Actual	-Alternative- --Reference-- Upper	-----Boundary Values---- ----------Upper--------- Beta	Alpha
1	0.2500	4.176597	2.89500	-0.95755	4.78775
2	0.5000	8.353195	5.78999	1.91510	5.74530
3	0.7500	12.52979	8.68499	4.78775	6.70285
4	1.0000	16.70639	11.57998	7.81300	7.81300

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 77.7.8.

Output 77.7.8 Boundary Plot



The third design specifies the BOUNDARYKEY=BETA option to derive the boundary values to maintain the Type II error probability level β.

The "Design Information" table in Output 77.7.9 displays design specifications and the derived maximum information. Note that with the BOUNDARYKEY=BETA option, the specified Type II error probability $\beta = 0.20$ is maintained.

Output 77.7.9 Whitehead Design Information

The SEQDESIGN Procedure
Design: BoundaryKeyBeta

Design Information

Statistic Distribution	Normal
Boundary Scale	Score
Alternative Hypothesis	Upper
Early Stop	Accept/Reject Null
Method	Whitehead
Boundary Key	Beta
Alternative Reference	0.693147
Number of Stages	4
Alpha	0.05011
Beta	0.2
Power	0.8
Max Information (Percent of Fixed Sample)	129.9364
Max Information	16.70639
Null Ref ASN (Percent of Fixed Sample)	62.60462
Alt Ref ASN (Percent of Fixed Sample)	73.97042

The "Method Information" table in Output 77.7.10 displays the α and β errors and the derived drift parameter. The derived Type II error probability is the same as the specified $\beta = 0.20$ and the derived Type I error probability $\alpha = 0.05011$ is not the same as the specified $\alpha = 0.05$ with the BOUNDARYKEY=BETA option.

Output 77.7.10 Method Information

Method Information

				------Whitehead-----	
Boundary	Method	Alpha	Beta	Tau	C
Upper Alpha	Whitehead	0.05011	.	0.25	4.60517
Upper Beta	Whitehead	.	0.20000	0.25	4.60517

Method Information

Boundary	Alternative Reference	Drift
Upper Alpha	0.693147	2.833131
Upper Beta	0.693147	2.833131

The "Boundary Information" table in Output 77.7.11 displays information level, alternative reference, and boundary values.

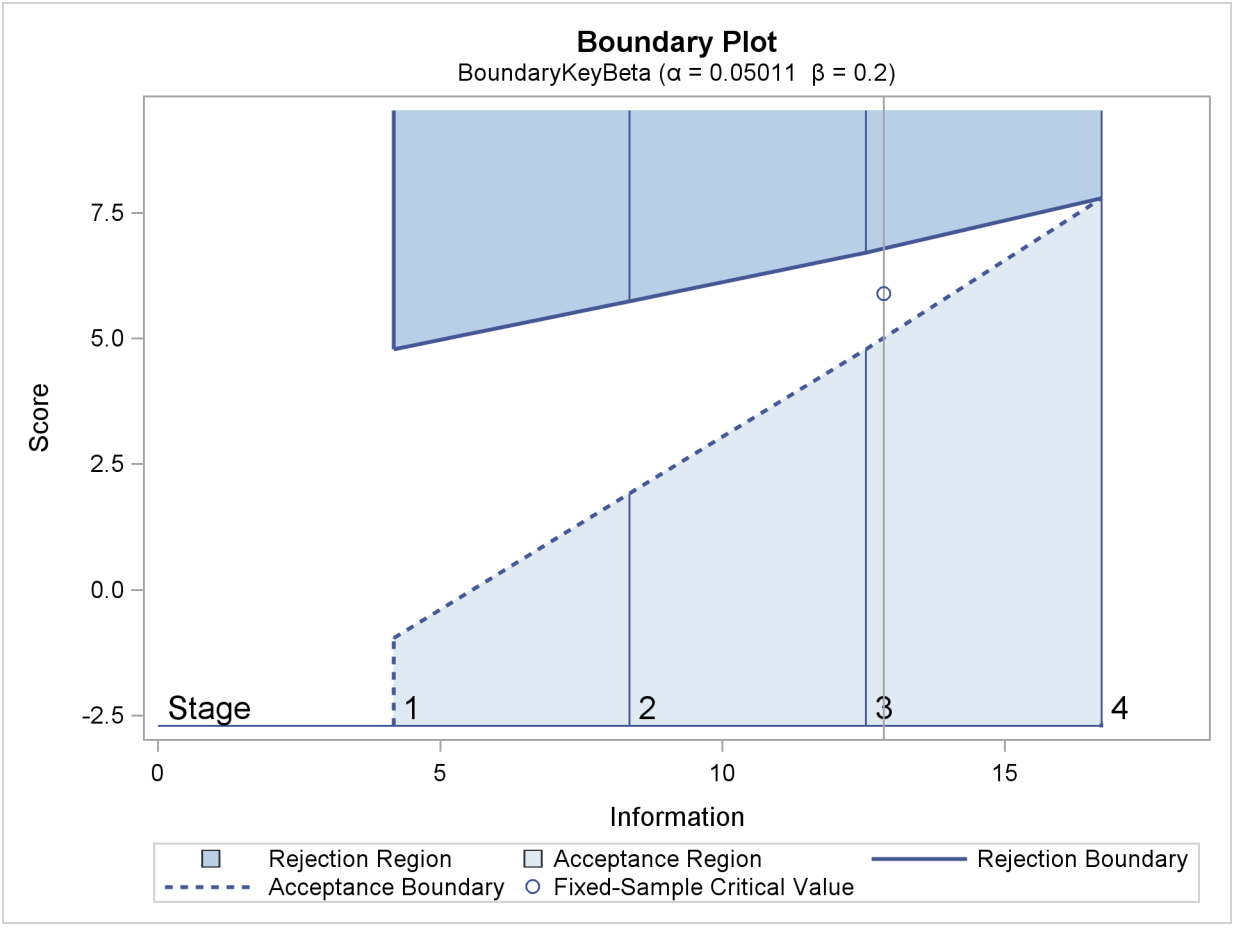
Output 77.7.11 Boundary Information

Boundary Information (Score Scale)
Null Reference = 0

	---Information Level--		-Alternative- --Reference--	-----Boundary Values---- ----------Upper---------	
Stage	Proportion	Actual	Upper	Beta	Alpha
1	0.2500	4.176597	2.89500	-0.95755	4.78775
2	0.5000	8.353195	5.78999	1.91510	5.74530
3	0.7500	12.52979	8.68499	4.78775	6.70285
4	1.0000	16.70639	11.57998	7.78899	7.78899

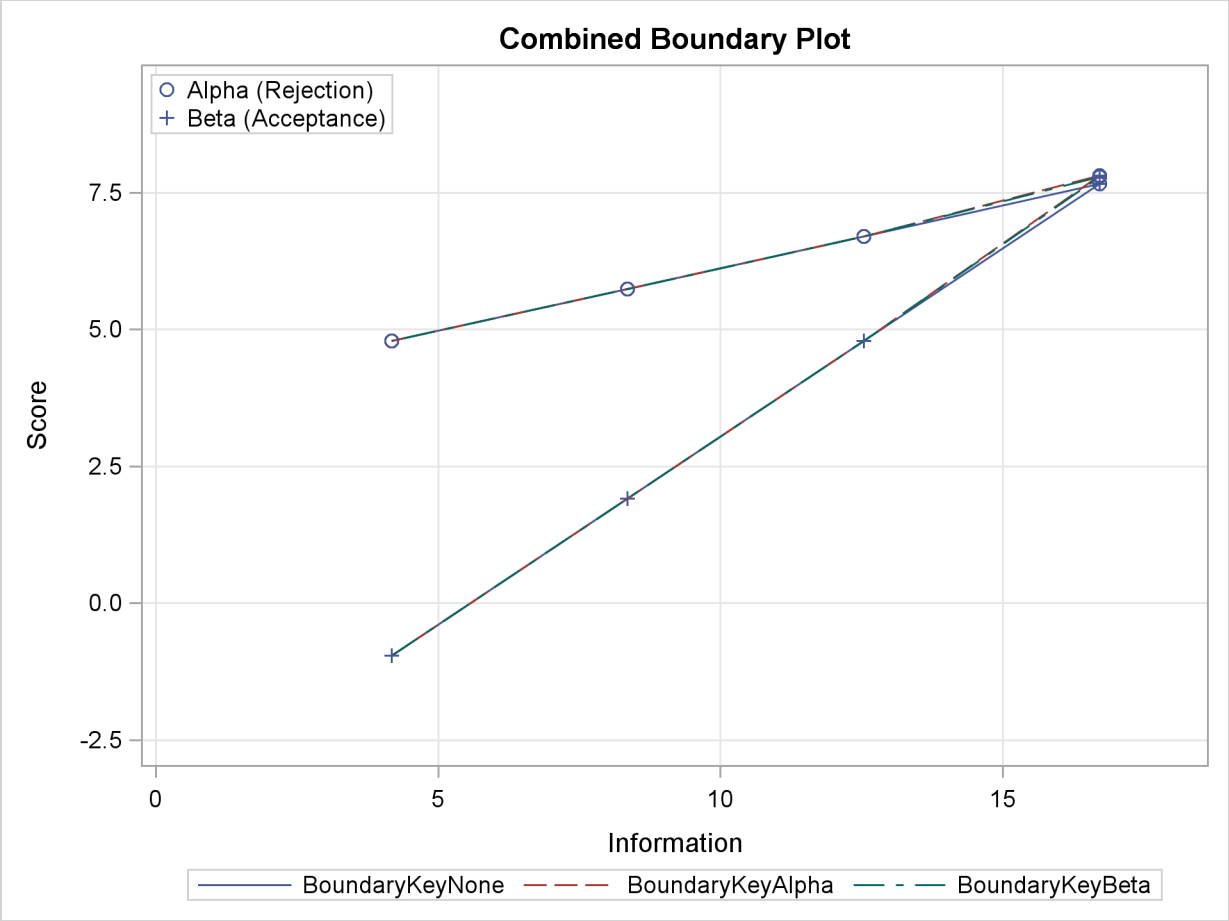
With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 77.7.12.

Output 77.7.12 Boundary Plot



With the PLOTS=COMBINEDBOUNDARY option, a combined plot of group sequential boundaries for all designs is displayed, as shown in Output 77.7.13. It shows that three designs are similar, with a slightly smaller boundary value at the final stage for the design with the BOUNDARYKEY=NONE option.

Output 77.7.13 Combined Boundary Plot



The following statements invoke the SEQDESIGN procedure and specify the SAMPLESIZE statement to derive required sample sizes for a log-rank test comparing two survival distributions for the treatment effect (Jennison and Turnbull 2000 pp. 77–79; Whitehead 1997, pp. 36–39):

```
proc seqdesign altref=0.693147
               bscale=score
               ;
  BoundaryKeyAlpha: design nstages=4
                           method=whitehead
                           boundarykey=alpha
                           alt=upper   stop=both
                           alpha=0.05  beta=0.20
                           ;
  samplesize model=twosamplesurvival
                  ( nullhazard=0.03466 accrate=10);
run;
```

The design is identical to the previous design with the BOUNDARYKEY=ALPHA option except with the addition of the sample size computation.

The "Sample Size Summary" table in Output 77.7.14 displays parameters for the sample size computation. Since the ACCTIME= option is not specified for the accrual time, the minimum and maximum accrual times are derived for the specified accrual rate.

Output 77.7.14 Sample Size Summary

The SEQDESIGN Procedure
Design: BoundaryKeyAlpha

Sample Size Summary

Test	Two-Sample Survival
Null Hazard Rate	0.03466
Hazard Rate (Group A)	0.01733
Hazard Rate (Group B)	0.03466
Hazard Ratio	0.5
log(Hazard Ratio)	-0.69315
Reference Hazards	Alt Ref
Accrual Rate	10
Min Accrual Time	6.682556
Min Sample Size	66.82556
Max Accrual Time	25.40111
Max Sample Size	254.0111
Max Number of Events	66.82556

If the ACCTIME=20 option is specified in the SAMPLESIZE statement, the "Sample Size Summary" table in Output 77.7.15 also displays the follow-up time and maximum sample size with the specified accrual time.

Output 77.7.15 Sample Size Summary

The SEQDESIGN Procedure
Design: WhiteheadKeyAlpha

Sample Size Summary

Test	Two-Sample Survival
Null Hazard Rate	0.03466
Hazard Rate (Group A)	0.01733
Hazard Rate (Group B)	0.03466
Hazard Ratio	0.5
log(Hazard Ratio)	-0.69315
Reference Hazards	Alt Ref
Accrual Rate	10
Accrual Time	20
Follow-up Time	6.474376
Total Time	26.47438
Max Number of Events	66.82556
Max Sample Size	200
Expected Sample Size (Null Ref)	161.5941
Expected Sample Size (Alt Ref)	172.4693

The "Number of Events (D) and Sample Sizes (N)" table in Output 77.7.16 displays the required time at each stage, in both fractional and integer numbers. The derived times under the heading "Fractional Time" are not integers. These times are rounded up to integers under the heading "Ceiling Time." The table also displays the numbers of events and sample sizes at each stage.

Output 77.7.16 Number of Events and Sample Sizes

Numbers of Events (D) and Sample Sizes (N)
Two-Sample Log-Rank Test

	Fractional Time						
Stage	D	D(Grp 1)	D(Grp 2)	Time	N	N(Grp 1)	N(Grp 2)
1	16.71	5.82	10.89	11.9867	119.87	59.93	59.93
2	33.41	11.84	21.57	17.3585	173.58	86.79	86.79
3	50.12	18.01	32.11	21.7480	200.00	100.00	100.00
4	66.83	24.46	42.37	26.4744	200.00	100.00	100.00

Numbers of Events (D) and Sample Sizes (N)
Two-Sample Log-Rank Test

	Fractional Time	Ceiling Time					
Stage	Information	D	D(Grp 1)	D(Grp 2)	Time	N	N(Grp 1)
1	4.1766	16.74	5.83	10.91	12	120.00	60.00
2	8.3532	35.73	12.68	23.04	18	180.00	90.00
3	12.5298	51.07	18.37	32.70	22	200.00	100.00
4	16.7064	68.55	25.14	43.41	27	200.00	100.00

Numbers of Events (D) and Sample Sizes (N)
Two-Sample Log-Rank Test

	Ceiling Time	
Stage	N(Grp 2)	Information
1	60.00	4.1854
2	90.00	8.9322
3	100.00	12.7667
4	100.00	17.1378

Example 77.8: Creating a One-Sided Error Spending Design

This example requests a five-stage, one-sided group sequential design for normally distributed statistics. The design uses an O'Brien-Fleming-type error spending function for the α boundary and a Pocock-type error spending function for the β boundary. The following statements request a one-sided design by using different α and β spending functions:

```
ods graphics on;
proc seqdesign altref=0.2  errspend
               pss(cref=0 0.5 1)
               stopprob(cref=0 0.5 1)
```

```
                plots=(asn power errspend)
                ;
   OneSidedErrorSpending: design nstages=5
                          method(alpha)=errfuncobf
                          method(beta)=errfuncpoc
                          alt=upper  stop=both
                          alpha=0.025
                          ;
run;
ods graphics off;
```

The "Design Information" table in Output 77.8.1 displays design specifications and the derived statistics. With the specified alternative reference, the maximum information is derived.

Output 77.8.1 Error Spending Method Design Information

The SEQDESIGN Procedure
Design: OneSidedErrorSpending

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Upper
Early Stop	Accept/Reject Null
Method	Error Spending
Boundary Key	Both
Alternative Reference	0.2
Number of Stages	5
Alpha	0.025
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	119.4278
Max Information	313.7196
Null Ref ASN (Percent of Fixed Sample)	50.35408
Alt Ref ASN (Percent of Fixed Sample)	78.77223

The "Method Information" table in Output 77.8.2 displays the α and β errors, alternative reference, and derived drift parameter, which is the standardized alternative reference at the final stage.

Output 77.8.2 Method Information

Method Information

Boundary	Method	Alpha	Beta	----Error Spending---- Function
Upper Alpha	Error Spending	0.02500	.	Approx O'Brien-Fleming
Upper Beta	Error Spending	.	0.10000	Approx Pocock

Method Information

Boundary	Alternative Reference	Drift
Upper Alpha	0.2	3.542426
Upper Beta	0.2	3.542426

With the STOPPROB option, the "Expected Cumulative Stopping Probabilities" table in Output 77.8.3 displays the expected stopping stage and cumulative stopping probability to reject the null hypothesis at each stage under various hypothetical references $\theta = c_i\theta_1$, where θ_1 is the alternative reference and c_i are values specified in the CREF= option.

Output 77.8.3 Stopping Probabilities

Expected Cumulative Stopping Probabilities
Reference = CRef * (Alt Reference)

CRef	Expected Stopping Stage	Source
0.0000	2.108	Reject Null
0.0000	2.108	Accept Null
0.0000	2.108	Total
0.5000	3.296	Reject Null
0.5000	3.296	Accept Null
0.5000	3.296	Total
1.0000	3.298	Reject Null
1.0000	3.298	Accept Null
1.0000	3.298	Total

Expected Cumulative Stopping Probabilities
Reference = CRef * (Alt Reference)

	Stopping Probabilities				
CRef	Stage_1	Stage_2	Stage_3	Stage_4	Stage_5
0.0000	0.00000	0.00039	0.00381	0.01221	0.02500
0.0000	0.38080	0.69133	0.86162	0.94170	0.97500
0.0000	0.38080	0.69173	0.86543	0.95391	1.00000
0.5000	0.00002	0.01265	0.09650	0.24465	0.38724
0.5000	0.13665	0.28063	0.41080	0.52230	0.61276
0.5000	0.13667	0.29328	0.50730	0.76695	1.00000
1.0000	0.00050	0.13209	0.52642	0.80390	0.90000
1.0000	0.02954	0.05231	0.07085	0.08648	0.10000
1.0000	0.03004	0.18440	0.59728	0.89039	1.00000

With the PSS option, the "Power and Expected Sample Sizes" table in Output 77.8.4 displays powers and expected sample sizes under various hypothetical references $\theta = c_i\theta_1$, where θ_1 is the alternative reference and $c_i = 0, 0.5, 1, 1.5$ are the default values in the CREF= option.

Output 77.8.4 Power and Expected Sample Size Information

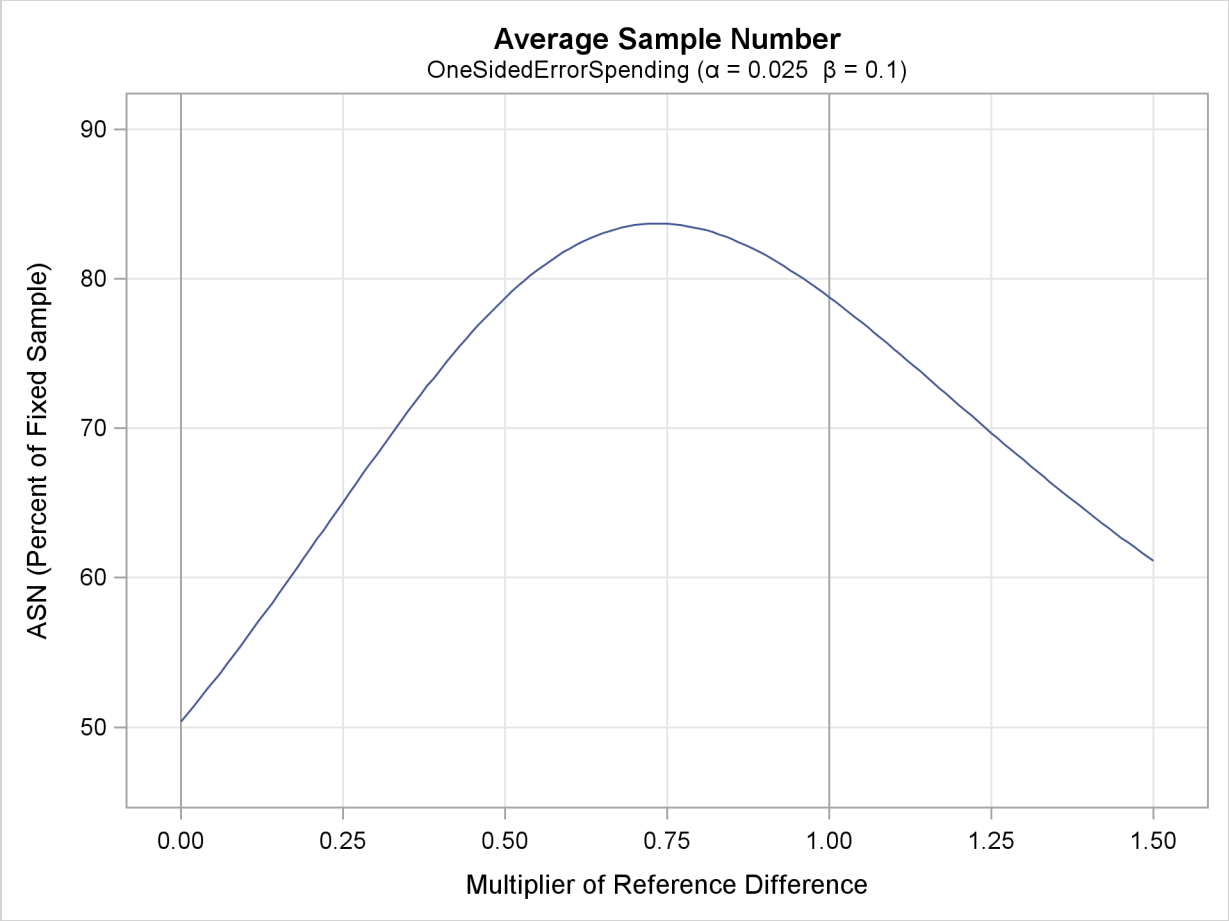
Powers and Expected Sample Sizes
Reference = CRef * (Alt Reference)

CRef	Power	-Sample Size- Percent Fixed-Sample
0.0000	0.02500	50.3541
0.5000	0.38724	78.7219
1.0000	0.90000	78.7722

With the PLOTS=ASN option, the procedure displays a plot of expected sample sizes under various hypothetical references, as shown in Output 77.8.5. By default, expected sample sizes under the

hypotheses $\theta = c_i\,\theta_1$, $c_i = 0, 0.01, 0.02, \ldots, 1.50$, are displayed, where θ_1 is the alternative reference.

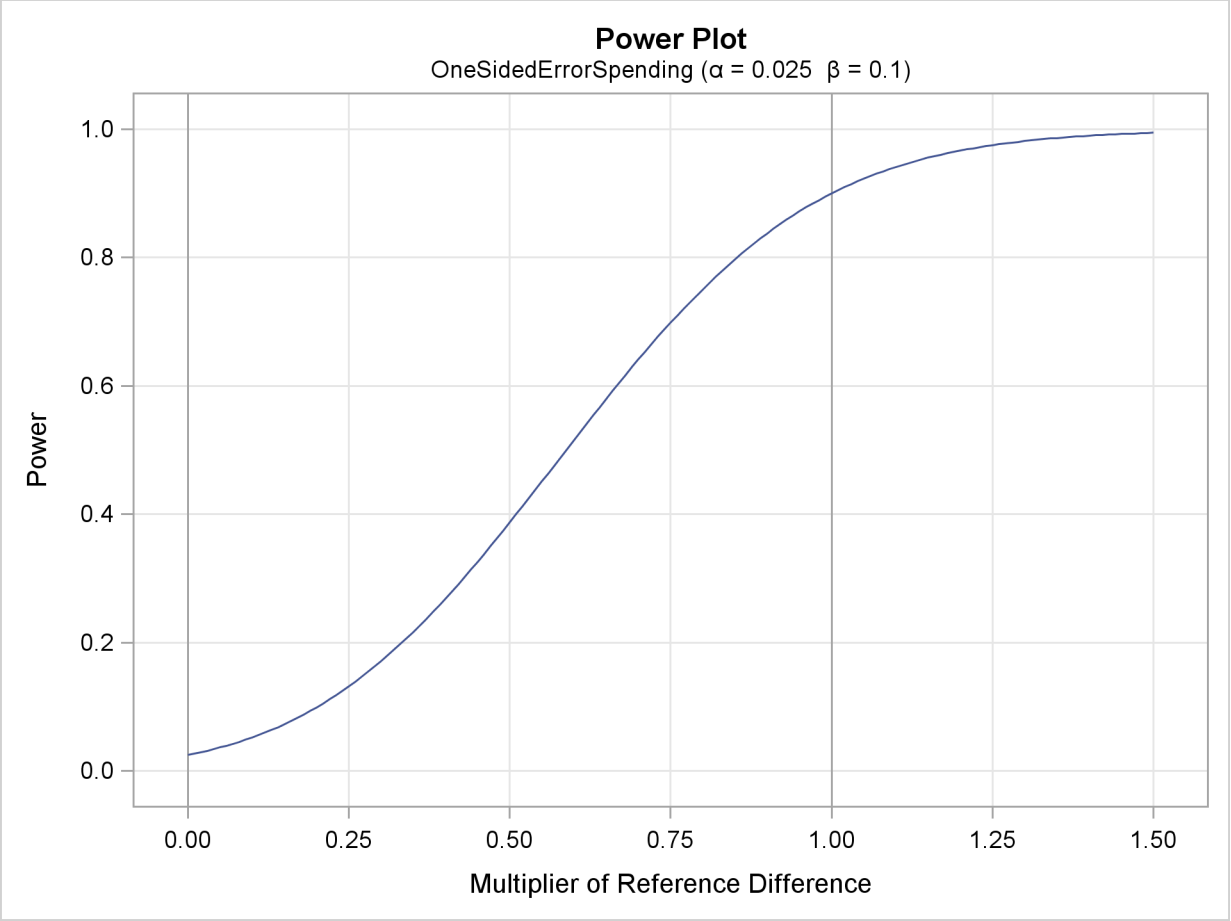
Output 77.8.5 ASN Plot



With the PLOTS=POWER option, the procedure displays a plot of the power curves under various hypothetical references for all designs simultaneously, as shown in Output 77.8.6. By default, the option CREF= $0, 0.01, 0.02, \ldots, 1.50$ and powers under hypothetical references $\theta = c_i\,\theta_1$ are displayed, where c_i are values specified in the CREF= option. These CREF= values are displayed on the horizontal axis.

Under the null hypothesis, $c_i = 0$, the power is 0.025, the upper Type I error probability. Under the alternative hypothesis, $c_i = 1$, the power is 0.9, one minus the Type II error probability. The plot shows only minor difference between the two designs.

Output 77.8.6 Power Plot



The "Boundary Information" table in Output 77.8.7 displays information level, alternative reference, and boundary values. By default, the alternative reference and boundary values are displayed with the default standardized Z scale. That is, the resulting standardized alternative reference at stage k is given by $\theta_1\sqrt{I_k}$, where θ_1 is the specified alternative reference and I_k is the information level at stage k, $k = 1, 2, \ldots, 5$.

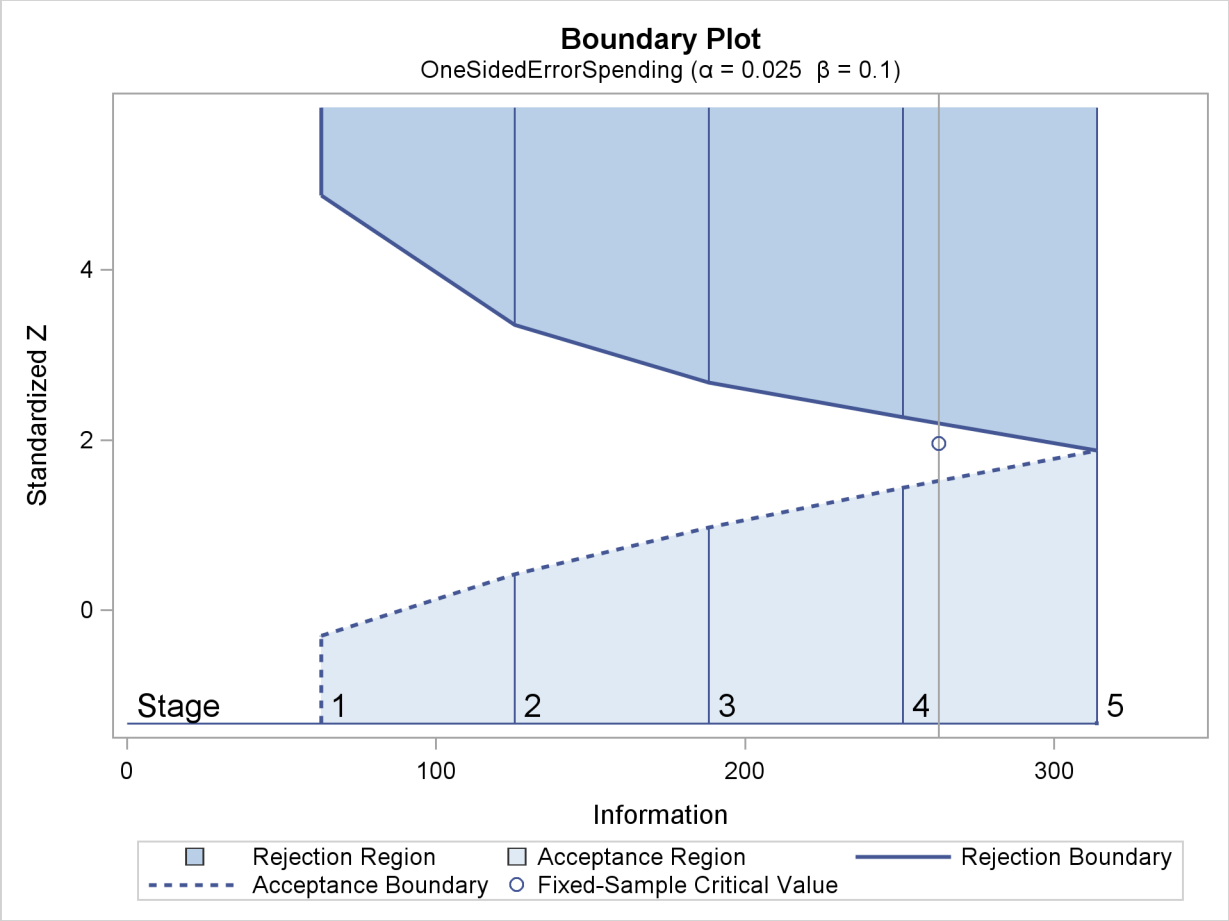
Output 77.8.7 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

	---Information Level--		-Alternative- --Reference--	-----Boundary Values---- ----------Upper---------	
Stage	Proportion	Actual	Upper	Beta	Alpha
1	0.2000	62.74393	1.58422	-0.30338	4.87688
2	0.4000	125.4879	2.24043	0.41667	3.35706
3	0.6000	188.2318	2.74395	0.97165	2.67766
4	0.8000	250.9757	3.16844	1.43627	2.26535
5	1.0000	313.7196	3.54243	1.87522	1.87522

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 77.8.8. This plot displays the boundary values in the "Boundary Information" table in Output 77.8.7.

Output 77.8.8 Boundary Plot



The "Error Spending Information" table in Output 77.8.9 displays cumulative error spending at each stage for each boundary.

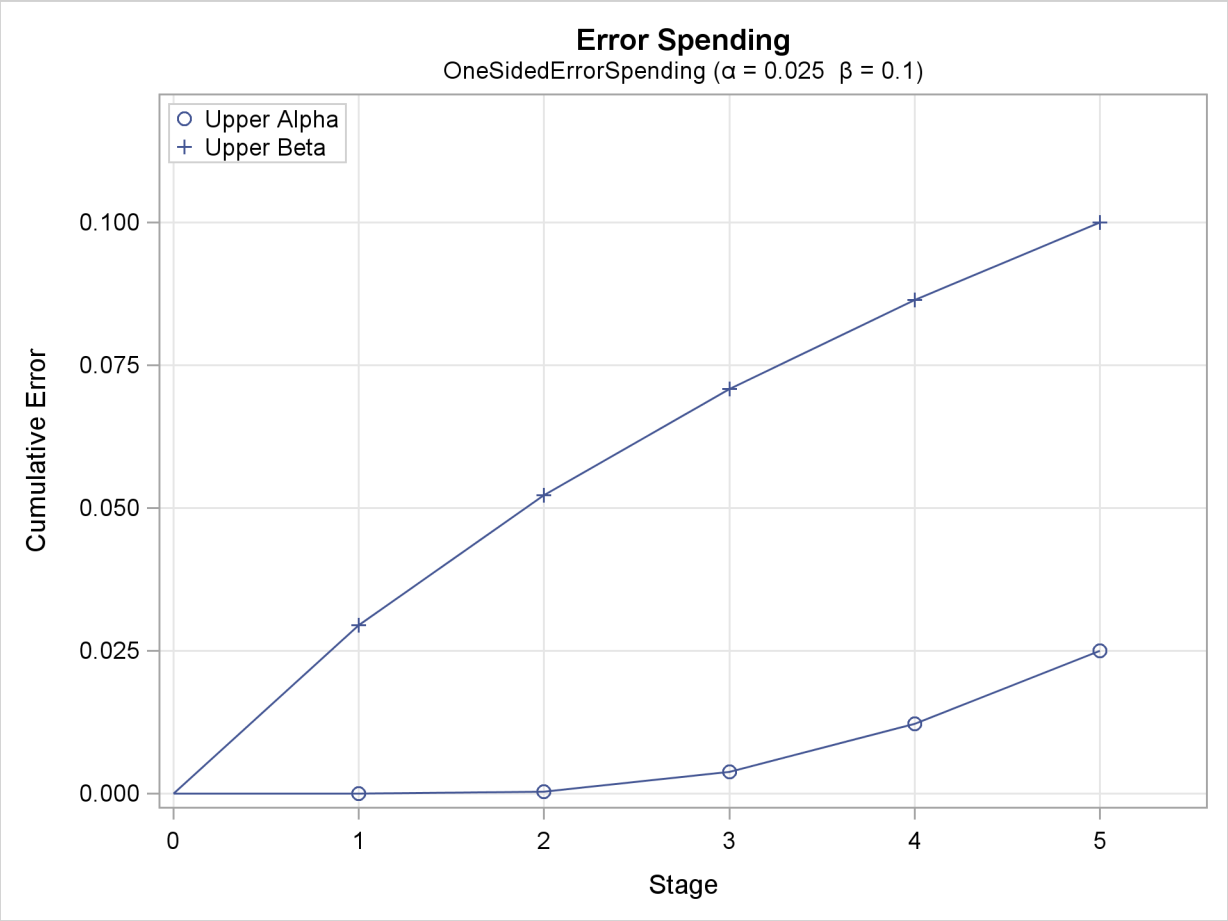
Output 77.8.9 Error Spending Information

Error Spending Information

Stage	-Information Level- Proportion	-Cumulative Error Spending- ----------Upper---------- Beta	Alpha
1	0.2000	0.02954	0.00000
2	0.4000	0.05231	0.00039
3	0.6000	0.07085	0.00381
4	0.8000	0.08648	0.01221
5	1.0000	0.10000	0.02500

With the PLOTS=ERRSPEND option, the procedure displays a plot of error spending for each boundary, as shown in Output 77.8.10. This plot displays the cumulative error spending at each stage in the "Error Spending Information" table in Output 77.8.9. The O'Brien-Fleming-type α spending function is conservative in early stages because it uses much less at early stages than in the later stages. In contrast, the Pocock-type β spending function uses more at early stages than in the later stages.

Output 77.8.10 Error Spending Plot



Example 77.9: Creating Designs with Various Number of Stages

This example requests three group sequential designs for normally distributed statistics. Each design uses the power family error spending function with the default power parameter $\rho = 2$. The specified error spending method is between the approximated Pocock method ($\rho = 1$) and the approximated O'Brien-Fleming method ($\rho = 3$) (Jennison and Turnbull 1999, p. 148). The three designs are identical except for the specified number of stages. The following statements request the group sequential design with the default standardized Z scale for the boundary values:

```
ods graphics on;
proc seqdesign plots=( asn
                       power
                       combinedboundary
                       errspend(hscale=info)
                       )
                ;
   TwoStageDesign:  design nstages=2
                    method=errfuncpow
                    alt=upper  stop=reject
                    ;
   FiveStageDesign: design nstages=5
                    method=errfuncpow
                    alt=upper  stop=reject
                    ;
   TenStageDesign:  design nstages=10
                    method=errfuncpow
                    alt=upper  stop=reject
                    ;
run;
ods graphics off;
```

The "Design Information" table in Output 77.9.1 displays design information for the two-stage design.

Output 77.9.1 Design Information

The SEQDESIGN Procedure
Design: TwoStageDesign

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Upper
Early Stop	Reject Null
Method	Error Spending
Boundary Key	Both
Number of Stages	2
Alpha	0.05
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	102.4167
Null Ref ASN (Percent of Fixed Sample)	101.7766
Alt Ref ASN (Percent of Fixed Sample)	79.81021

The "Boundary Information" table in Output 77.9.2 displays the information level, alternative reference, and boundary values with the default standardized normal Z scale. The resulting standardized alternative reference at stage k is given by $\theta_1\sqrt{I_k}$, where θ_1 is the alternative reference and I_k is the information level at stage k, $k = 1, 2$.

Output 77.9.2 Boundary Information in *Z* Scale

Boundary Information (Standardized Z Scale)
Null Reference = 0

Stage	-Information Level- Proportion	-Alternative- --Reference-- Upper	-Boundary Values- ------Upper------ Alpha
1	0.5000	2.09414	2.24140
2	1.0000	2.96156	1.69970

The "Design Information" table in Output 77.9.3 displays design information for the five-stage design. Compared with the two-stage design in Output 77.9.1, the maximum information increases from 102.42 to 105.62, and the average sample number under the alternative reference (Alt Ref ASN) decreases from 79.81 to 69.64.

Output 77.9.3 Design Information

The SEQDESIGN Procedure
Design: FiveStageDesign

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Upper
Early Stop	Reject Null
Method	Error Spending
Boundary Key	Both
Number of Stages	5
Alpha	0.05
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	105.6235
Null Ref ASN (Percent of Fixed Sample)	104.356
Alt Ref ASN (Percent of Fixed Sample)	69.64322

The "Boundary Information" table in Output 77.9.4 displays the information level, alternative reference, and boundary values with the default standardized normal Z scale.

Output 77.9.4 Boundary Information in Z Scale

Boundary Information (Standardized Z Scale)
Null Reference = 0

Stage	-Information Level- Proportion	-Alternative- --Reference-- Upper	-Boundary Values- ------Upper------ Alpha
1	0.2000	1.34502	2.87816
2	0.4000	1.90215	2.47023
3	0.6000	2.32965	2.20095
4	0.8000	2.69005	1.98182
5	1.0000	3.00756	1.79024

The "Design Information" table in Output 77.9.5 displays design information for the ten-stage design. Compared with the five-stage design in Output 77.9.3, the maximum information increases further from 105.62 to 107.26 and under the alternative reference, the average sample number decreases further from 69.64 to 66.36.

Output 77.9.5 Design Information

The SEQDESIGN Procedure
Design: TenStageDesign

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Upper
Early Stop	Reject Null
Method	Error Spending
Boundary Key	Both
Number of Stages	10
Alpha	0.05
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	107.256
Null Ref ASN (Percent of Fixed Sample)	105.7276
Alt Ref ASN (Percent of Fixed Sample)	66.35565

The "Boundary Information" table in Output 77.9.6 displays the information level, alternative reference, and boundary values with the default standardized normal Z scale.

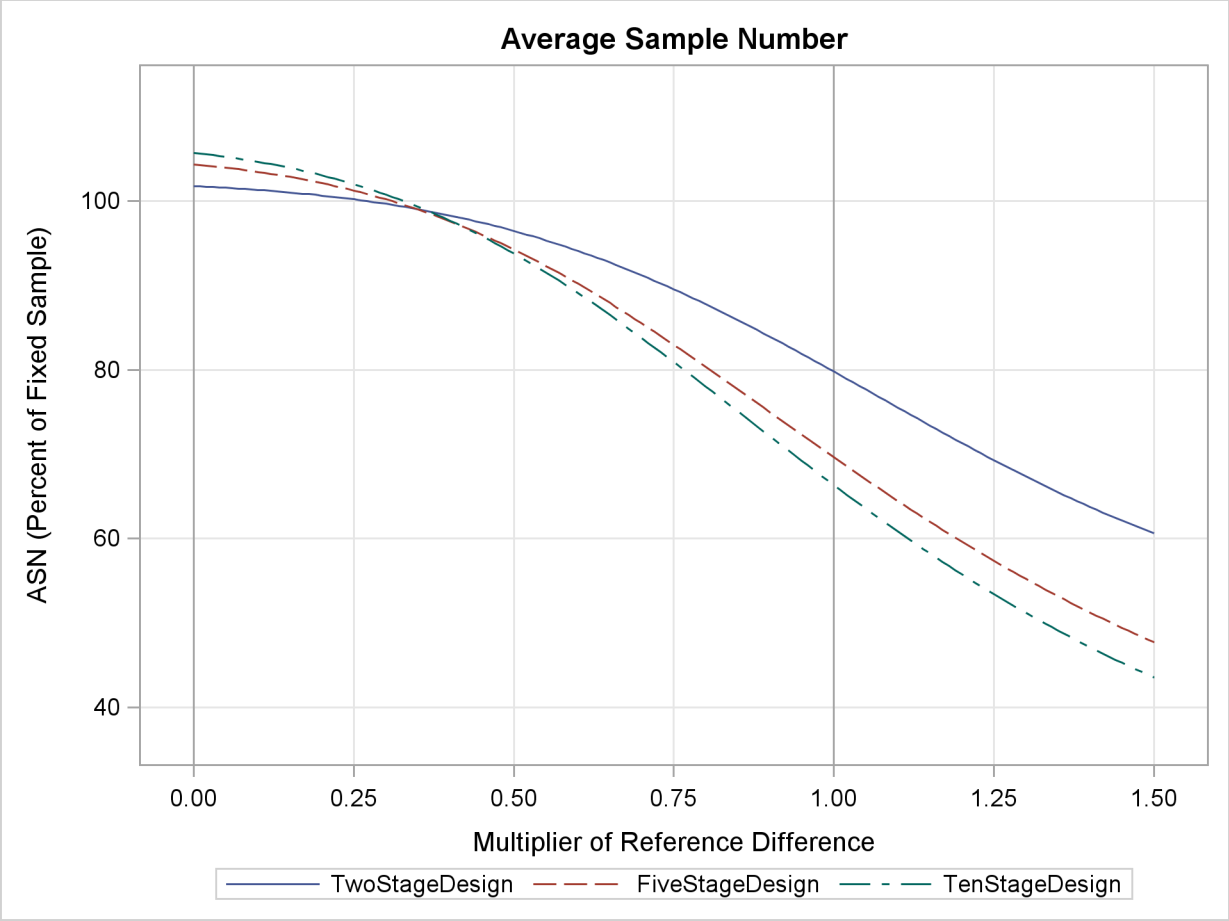
Output 77.9.6 Boundary Information in *Z* Scale

Boundary Information (Standardized Z Scale)
Null Reference = 0

Stage	-Information Level- Proportion	-Alternative- --Reference-- Upper	-Boundary Values- ------Upper------ Alpha
1	0.1000	0.95840	3.29053
2	0.2000	1.35538	2.94037
3	0.3000	1.65999	2.72115
4	0.4000	1.91679	2.54808
5	0.5000	2.14304	2.40114
6	0.6000	2.34758	2.27127
7	0.7000	2.53568	2.15359
8	0.8000	2.71076	2.04503
9	0.9000	2.87519	1.94355
10	1.0000	3.03072	1.84765

With the PLOTS=ASN option, the procedure displays a plot of average sample numbers under various hypothetical references for all designs simultaneously, as shown in Output 77.9.7. By default, the option CREF= $0, 0.01, 0.02, \ldots, 1.50$ and expected sample sizes under the hypothetical references $\theta = c_i\,\theta_1$ are displayed, where c_i are values specified in the CREF= option. These CREF= values are displayed on the horizontal axis.

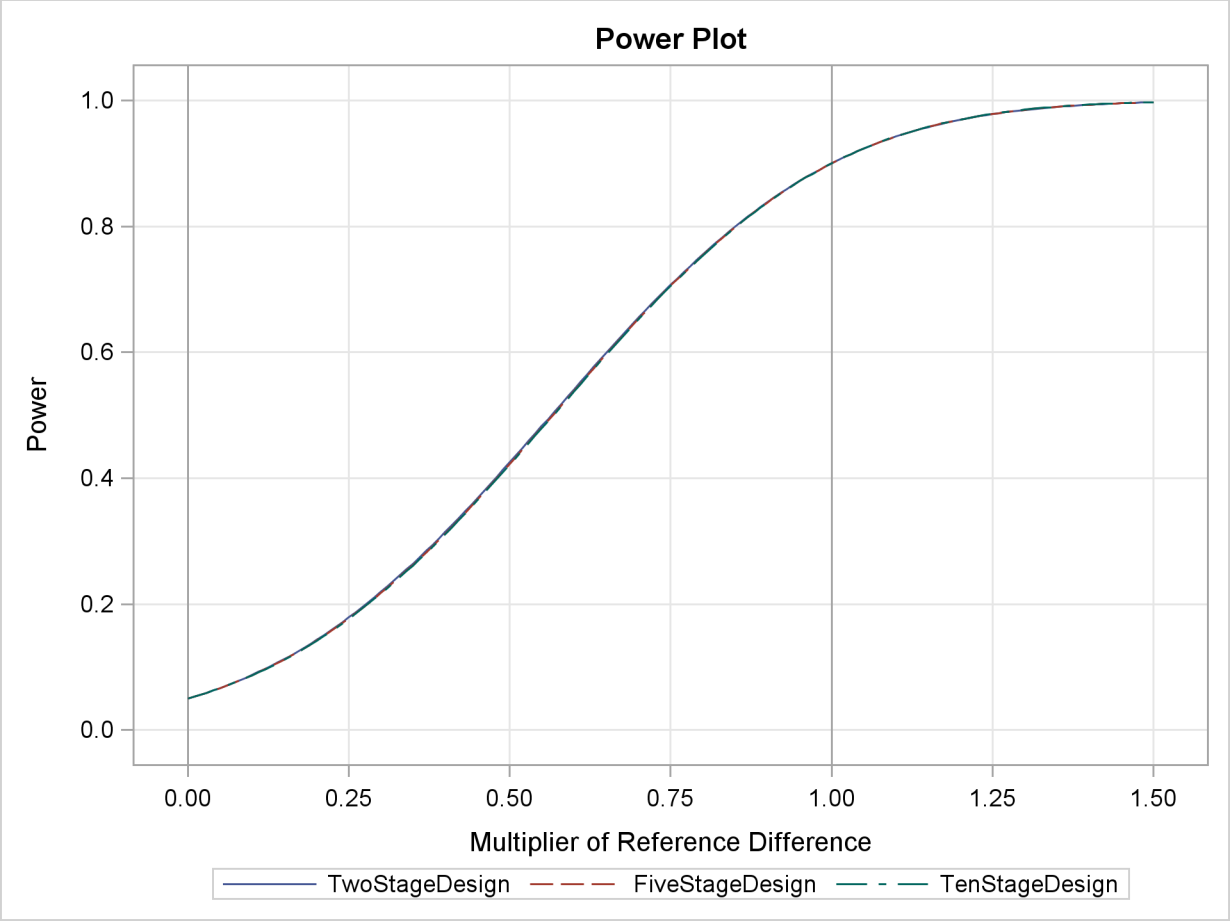
Output 77.9.7 ASN Plot



The plot shows that as the number of stages increases, the average sample number as a percentage of the fixed-sample design increases under the null hypothesis ($c_i = 0$) but decreases under the alternative hypothesis ($c_i = 1$).

With the PLOTS=POWER option, the procedure displays a plot of the power curves under various hypothetical references for all designs simultaneously, as shown in Output 77.9.8. By default, the option CREF= $0, 0.01, 0.02, \ldots, 1.50$ and powers under hypothetical references $\theta = c_i\,\theta_1$ are displayed, where c_i are values specified in the CREF= option. These CREF= values are displayed on the horizontal axis.

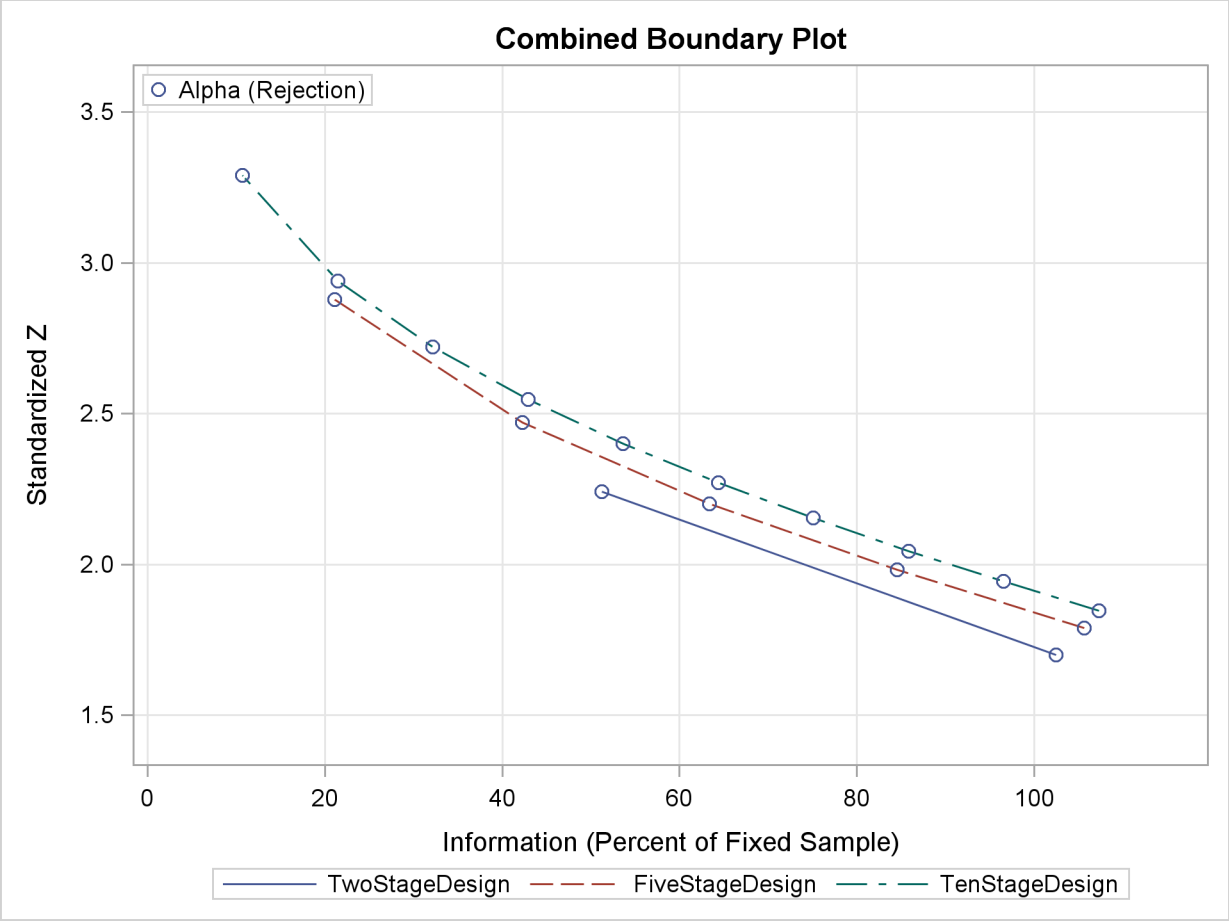
Output 77.9.8 Power Plot



Under the null hypothesis, $c_i = 0$, the power is 0.05, the upper Type I error probability. Under the alternative hypothesis, $c_i = 1$, the power is 0.9, one minus the Type II error probability. The plot shows only minor difference among the three designs.

With the PLOTS=COMBINEDBOUNDARY option, the procedure displays a plot of sequential boundaries for all designs simultaneously, as shown in Output 77.9.9. By default, the information levels are used on the horizontal axis. Since the maximum information is not available for the design, the percent information ratios with respect to the corresponding fixed-sample design are displayed in the plot.

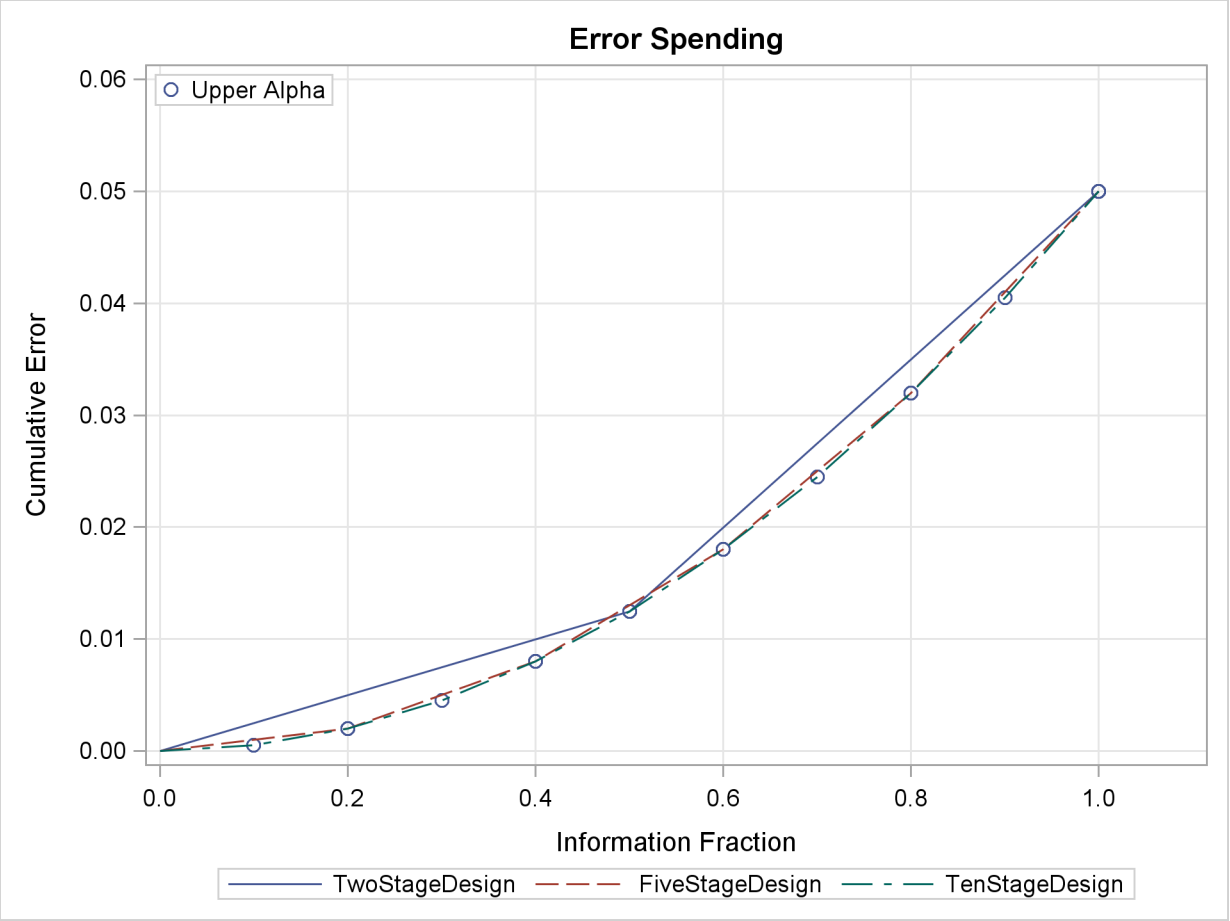
Output 77.9.9 Combined Boundary Plot



The plot shows that as the number of stages increases, the maximum information increases and the α boundary values also increase.

With the PLOTS=ERRSPEND(HSCALE=INFO) option, the procedure displays a plot of cumulative error spends for all boundaries in the designs simultaneously, as shown in Output 77.9.10.

Output 77.9.10 Error Spending Plot



The plot shows similar error spending for these three designs since all three designs are generated from the same power family error spending function.

Example 77.10: Creating Two-Sided Error Spending Designs with and without Overlapping Lower and Upper β Boundaries

This example requests two three-stage group sequential designs for normally distributed statistics. Each design uses a power family error spending function with a specified two-sided alternative hypothesis $H_1 : \theta_1 = \pm 0.2$ and early stopping only to accept the null hypothesis H_0.

The first design uses the BETAOVERLAP=NOADJUST option to derive acceptance boundary values without adjusting for the possible overlapping of the lower and upper β boundaries computed from the two corresponding one-sided tests. The second design uses the BETAOVERLAP=ADJUST option to test the overlapping of the β boundaries at each interim stage based on the two corresponding one-sided tests and then to set the β boundary values at the stage to missing if overlapping occurs at that stage.

The following statements request a two-sided design with the BETAOVERLAP=NOADJUST option:

```
ods graphics on;
proc seqdesign altref=0.2 errspend;
   design nstages=3
          method=errfuncpow
          alt=twosided  stop=accept
          betaoverlap=noadjust
          beta=0.09
          ;
run;
ods graphics off;
```

The "Design Information" table in Output 77.10.1 displays design specifications and the derived statistics for the first design. With the specified alternative reference $\theta_1 = 0.2$, the maximum information is derived.

Output 77.10.1 Design Information

The SEQDESIGN Procedure
Design: Design_1

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Accept Null
Method	Error Spending
Boundary Key	Both
Alternative Reference	0.2
Number of Stages	3
Alpha	0.05
Beta	0.09
Power	0.91
Max Information (Percent of Fixed Sample)	103.8789
Max Information	282.9328
Null Ref ASN (Percent of Fixed Sample)	79.20197
Alt Ref ASN (Percent of Fixed Sample)	102.1476

The "Boundary Information" table in Output 77.10.2 displays the information level, alternative reference, and boundary values. With a specified alternative reference θ_1, the maximum information is derived from the procedure, and the actual information level at each stage is displayed in the table. With the default BOUNDARYSCALE=STDZ option, the alternative reference in the standardized Z scale at stage k is given by $\theta_1\sqrt{I_k}$, where θ_1 is the specified alternative reference and I_k is the information level at stage k, $k = 1, 2, 3$.

Output 77.10.2 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

Stage	Information Level- Proportion	Information Level- Actual	Alternative Reference Lower	Alternative Reference Upper	Boundary Values Lower Beta	Boundary Values Upper Beta
1	0.3333	94.31094	-1.94228	1.94228	-0.08239	0.08239
2	0.6667	188.6219	-2.74679	2.74679	-0.90351	0.90351
3	1.0000	282.9328	-3.36412	3.36412	-1.92519	1.92519

The "Error Spending Information" table in Output 77.10.3 displays the cumulative error spending at each stage for each boundary.

Output 77.10.3 Error Spending Information

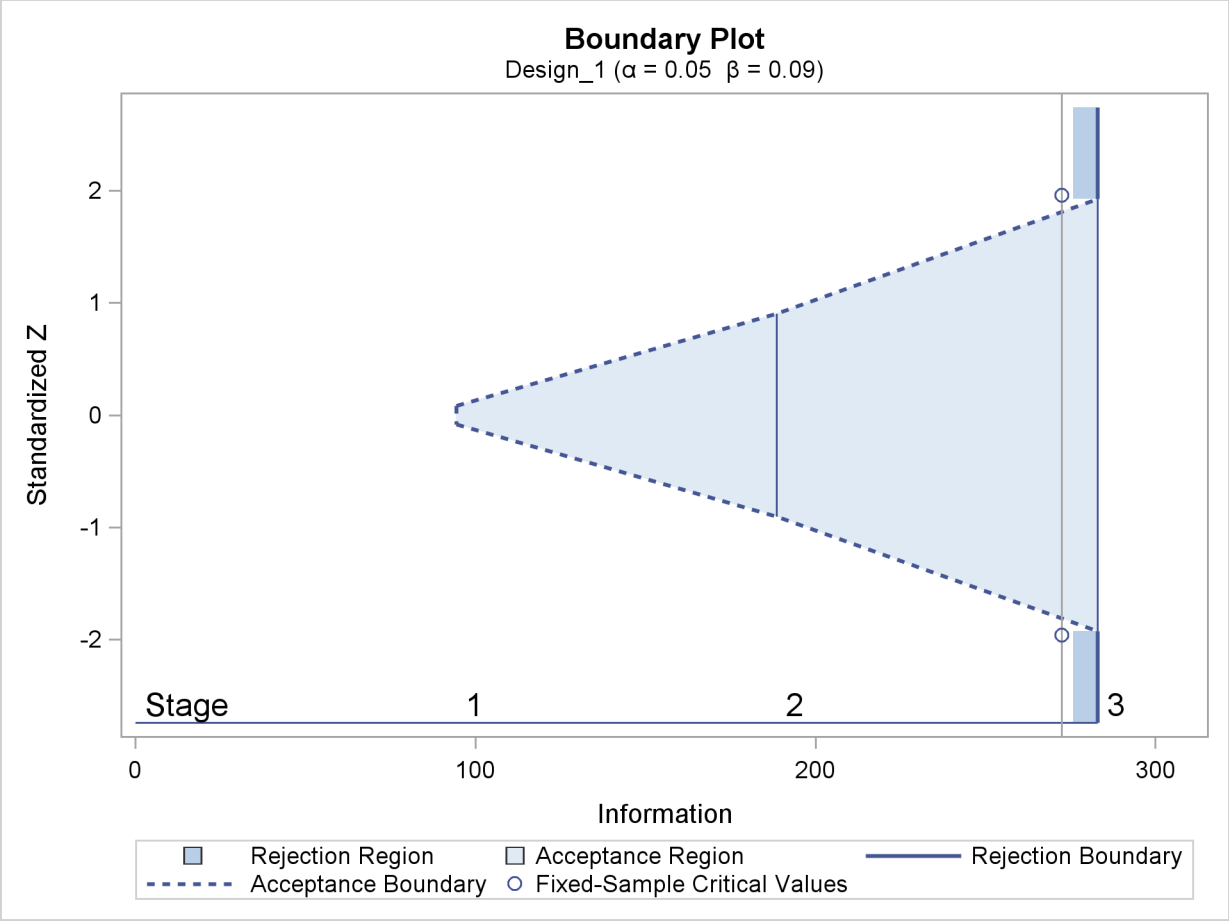
Error Spending Information

Stage	Information Level- Proportion	Cumulative Error Spending Lower Alpha	Cumulative Error Spending Lower Beta	Cumulative Error Spending Upper Beta	Cumulative Error Spending Upper Alpha
1	0.3333	0.00000	0.01000	0.01000	0.00000
2	0.6667	0.00000	0.04000	0.04000	0.00000
3	1.0000	0.02500	0.09000	0.09000	0.02500

With the STOP=ACCEPT option, the design does not stop at interim stages to reject H_0, and the α spending at each interim stage is zero. For the power family error spending function with the default parameter $\rho = 2$, the beta spending at stage 1 is $(1/3)^\rho \beta = (1/3)^2 0.09 = 0.01$, and the cumulative beta spending at stage 2 is $(2/3)^\rho \beta = (2/3)^2 0.09 = 0.04$.

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the acceptance and rejection regions is displayed by default, as shown in Output 77.10.4.

Output 77.10.4 Boundary Plot



The following statements request a two-sided design with the BETAOVERLAP=ADJUST option, which is the default:

```
ods graphics on;
proc seqdesign altref=0.2 errspend;
   design nstages=3
          method=errfuncpow
          alt=twosided
          stop=accept
          betaoverlap=adjust
          beta=0.09
          ;
run;
ods graphics off;
```

With the BETAOVERLAP=ADJUST option, the procedure first derives the usual β boundary values for the two-sided design and then checks for overlapping of the β boundaries for the two corresponding one-sided tests at each stage. If this type of overlapping occurs at a particular stage, the β boundary values for that stage are set to missing, the β spending values at that stage are reset to zero, and the β spending values at subsequent stages are adjusted proportionally.

The boundary values without adjusting for the possible overlapping of the two one-sided β boundaries are identical to the boundary values derived in the first design (with the BETAOVERLAP=NOADJUST option, as shown in Output 77.10.2). At stage 1, the upper β boundary value for the corresponding one-sided test is

$$\theta_1\sqrt{I_1} - \Phi^{-1}(1-\beta_1) = 0.2\sqrt{94.31094} - \Phi^{-1}(0.99) = 1.94228 - 2.32635 = -0.38407$$

where $\theta_1 = 0.2$ is the upper alternative reference, $I_1 = 94.31094$ is the information level at stage 1, and $\beta_1 = 0.01$ is the β spending at stage 1 (as shown in Output 77.10.3).

Similarly, the lower β boundary value for the corresponding one-sided test is computed as 0.38407. Since the upper β boundary value is less than the lower β boundary at stage 1, overlapping occurs, and so the β boundary values for the two-sided design are set to missing at stage 1.

With the β boundary values set to missing at stage 1 and the β spending $\beta'_1 = 0$ the β spending values at subsequent interim stages are adjusted proportionally. In this example, the adjusted β spending at stage 2 is computed as

$$\beta'_2 = \beta'_1 + \frac{\beta_2 - \beta_1}{\beta_3 - \beta_1}(\beta_3 - \beta'_1) = 0 + \frac{0.04 - 0.01}{0.09 - 0.01}\,0.09 = 0.03375$$

where β_k is the cumulative β spending at stage k before the adjustment, $k = 1, 2, 3$.

The "Design Information" table in Output 77.10.5 displays design specifications and derived statistics for the design.

Output 77.10.5 Design Information

The SEQDESIGN Procedure
Design: Design_1

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Accept Null
Method	Error Spending
Boundary Key	Both
Alternative Reference	0.2
Number of Stages	3
Alpha	0.05
Beta	0.09
Power	0.91
Max Information (Percent of Fixed Sample)	101.9388
Max Information	277.649
Null Ref ASN (Percent of Fixed Sample)	80.56408
Alt Ref ASN (Percent of Fixed Sample)	100.792

The "Boundary Information" table in Output 77.10.6 displays the information levels, alternative references, and boundary values.

Output 77.10.6 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

Stage	Information Level- Proportion	Information Level- Actual	Alternative- Reference- Lower	Alternative- Reference- Upper	Boundary Values- Lower- Beta	Boundary Values- Upper- Beta
1	0.3333	92.54967	-1.92405	1.92405	.	.
2	0.6667	185.0993	-2.72102	2.72102	-0.89469	0.89469
3	1.0000	277.649	-3.33256	3.33256	-1.93494	1.93494

The "Error Spending Information" table in Output 77.10.7 displays the cumulative error spending at each stage for each boundary.

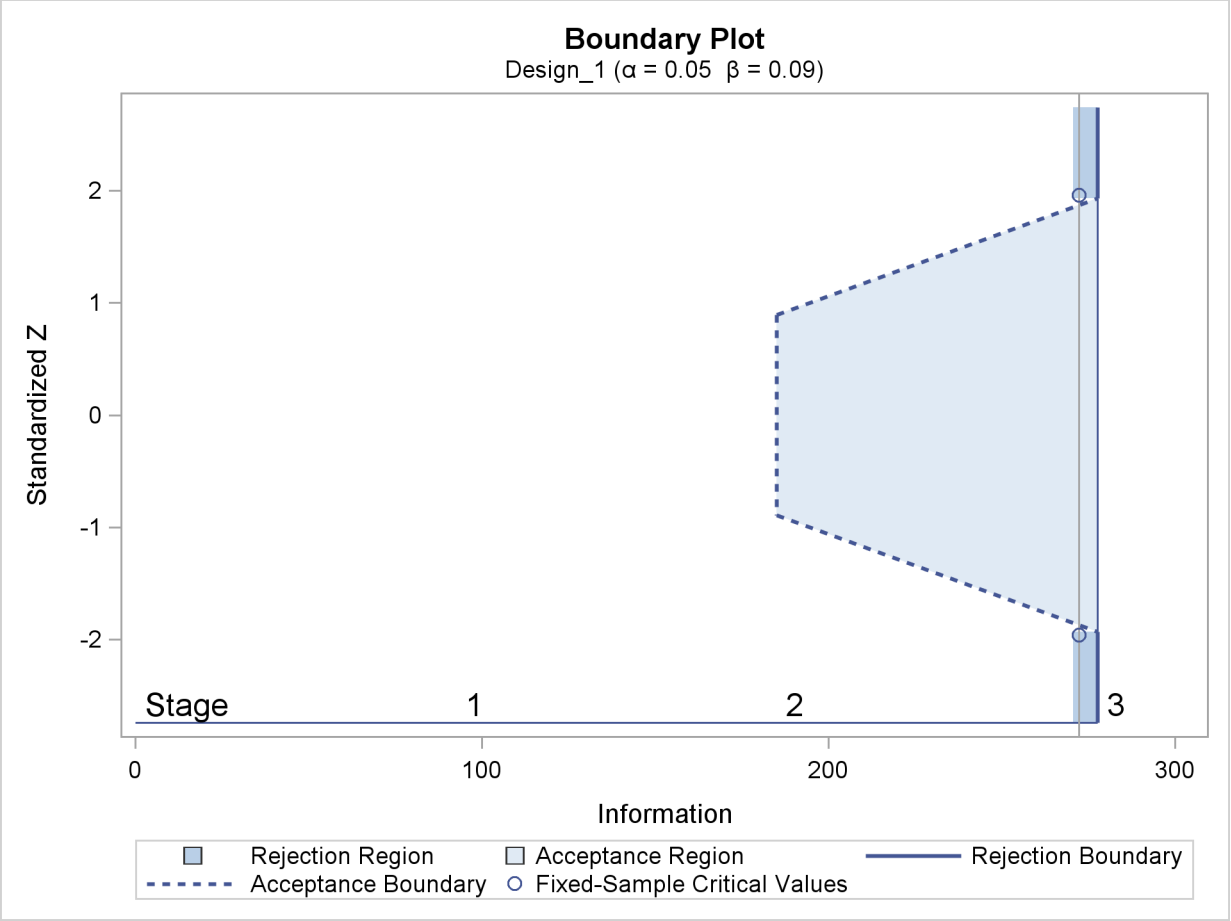
Output 77.10.7 Error Spending Information

Error Spending Information

Stage	Information Level- Proportion	Cumulative Error Spending- Lower- Alpha	Cumulative Error Spending- Lower- Beta	Cumulative Error Spending- Upper- Beta	Cumulative Error Spending- Upper- Alpha
1	0.3333	0.00000	0.00000	0.00000	0.00000
2	0.6667	0.00000	0.03375	0.03375	0.00000
3	1.0000	0.02500	0.09000	0.09000	0.02500

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the acceptance and rejection regions is displayed by default, as shown in Output 77.10.8.

Output 77.10.8 Boundary Plot



Example 77.11: Creating a Two-Sided Asymmetric Error Spending Design with Early Stopping to Reject H_0

This example requests a three-stage two-sided asymmetric group sequential design for normally distributed statistics.

The O'Brien-Fleming boundary can be approximated using a power family error spending function with parameter $\rho = 3$, and the Pocock boundary can be approximated using a power family error spending function with parameter $\rho = 1$ (Jennison and Turnbull 2000, p. 148). The following statements use the power family error spending function to creates a two-sided asymmetric design with early stopping to reject the null hypothesis H_0:

```
ods graphics on;
proc seqdesign altref=1.0
               pss(cref=0 0.5 1)
               stopprob(cref=0 0.5 1)
               errspend
               plots=(asn power errspend)
               ;
```

```
   TwoSidedErrorSpending: design nstages=3
                          method(upperalpha)=errfuncpow(rho=3)
                          method(loweralpha)=errfuncpow(rho=1)
                          info=cum(2 3 4)
                          alt=twosided
                          stop=reject
                          alpha=0.075(upper=0.025)
                          ;
run;
ods graphics off;
```

The design uses power family error spending functions with $\rho = 1$ for the lower α boundary and $\rho = 3$ for the upper α boundary. Thus, the design is conservative in the early stages and tends to stop the trials early only with a small p-value for the upper α boundary. The upper α level 0.025 is specified explicitly, and the lower α level is computed as $0.075 - 0.025 = 0.05$.

The "Design Information" table in Output 77.11.1 displays design specifications and the derived maximum information. Note that in order to attain the same information level for the asymmetric lower and upper boundaries, the derived power at the lower alternative 0.92963 is larger than the default 0.90.

Output 77.11.1 Design Information

The SEQDESIGN Procedure
Design: TwoSidedErrorSpending

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Reject Null
Method	Error Spending
Boundary Key	Both
Alternative Reference	1
Number of Stages	3
Alpha	0.075
Alpha (Lower)	0.05
Alpha (Upper)	0.025
Beta (Lower)	0.07037
Beta (Upper)	0.1
Power (Lower)	0.92963
Power (Upper)	0.9
Max Information (Percent of Fixed Sample)	102.4384
Max Information	10.76365
Null Ref ASN (Percent of Fixed Sample)	100.4877
Lower Alt Ref ASN (Percent of Fixed Sample)	64.8288
Upper Alt Ref ASN (Percent of Fixed Sample)	75.98778

The "Method Information" table in Output 77.11.2 displays the specified α and β error levels and the derived drift parameter. With the same information level used for the asymmetric lower and upper boundaries, only one of the β levels is maintained, and the other is derived to have the level less than or equal to the default level.

Output 77.11.2 Method Information

Method Information

Boundary	Method	Alpha	Beta	----Error Spending---- Function
Upper Alpha	Error Spending	0.02500	0.10000	Power (Rho=3)
Lower Alpha	Error Spending	0.05000	0.07037	Power (Rho=1)

Method Information

Boundary	Alternative Reference	Drift
Upper Alpha	1	3.280801
Lower Alpha	-1	-3.2808

With the STOPPROB(CREF=0 0.5 1) option, the "Expected Cumulative Stopping Probabilities" table in Output 77.11.3 displays the expected stopping stage and cumulative stopping probability to reject the null hypothesis H_0 at each stage under hypothetical references $\theta = 0$ (null hypothesis H_0), $\theta = 0.5\,\theta_1$, and $\theta = \theta_1$ (alternative hypothesis H_1), where θ_1 is the alternative reference.

Output 77.11.3 Stopping Probabilities

Expected Cumulative Stopping Probabilities
Reference = CRef * (Alt Reference)

CRef	Ref	Expected Stopping Stage	Source
0.0000	Lower Alt	2.924	Rej Null (Lower Alt)
0.0000	Lower Alt	2.924	Rej Null (Upper Alt)
0.0000	Lower Alt	2.924	Reject Null
0.5000	Lower Alt	2.456	Rej Null (Lower Alt)
0.5000	Lower Alt	2.456	Rej Null (Upper Alt)
0.5000	Lower Alt	2.456	Reject Null
1.0000	Lower Alt	1.531	Rej Null (Lower Alt)
1.0000	Lower Alt	1.531	Rej Null (Upper Alt)
1.0000	Lower Alt	1.531	Reject Null
0.0000	Upper Alt	2.924	Rej Null (Lower Alt)
0.0000	Upper Alt	2.924	Rej Null (Upper Alt)
0.0000	Upper Alt	2.924	Reject Null
0.5000	Upper Alt	2.758	Rej Null (Lower Alt)
0.5000	Upper Alt	2.758	Rej Null (Upper Alt)
0.5000	Upper Alt	2.758	Reject Null
1.0000	Upper Alt	1.967	Rej Null (Lower Alt)
1.0000	Upper Alt	1.967	Rej Null (Upper Alt)
1.0000	Upper Alt	1.967	Reject Null

Expected Cumulative Stopping Probabilities
Reference = CRef * (Alt Reference)

		----Stopping Probabilities---		
CRef	Ref	Stage_1	Stage_2	Stage_3
0.0000	Lower Alt	0.02500	0.03750	0.05000
0.0000	Lower Alt	0.00313	0.01055	0.02500
0.0000	Lower Alt	0.02813	0.04805	0.07500
0.5000	Lower Alt	0.21185	0.33190	0.45370
0.5000	Lower Alt	0.00005	0.00012	0.00021
0.5000	Lower Alt	0.21190	0.33202	0.45391
1.0000	Lower Alt	0.64054	0.82803	0.92963
1.0000	Lower Alt	0.00000	0.00000	0.00000
1.0000	Lower Alt	0.64054	0.82803	0.92963
0.0000	Upper Alt	0.02500	0.03750	0.05000
0.0000	Upper Alt	0.00313	0.01055	0.02500
0.0000	Upper Alt	0.02813	0.04805	0.07500
0.5000	Upper Alt	0.00090	0.00110	0.00120
0.5000	Upper Alt	0.05769	0.18269	0.36458
0.5000	Upper Alt	0.05860	0.18379	0.36578
1.0000	Upper Alt	0.00001	0.00001	0.00001
1.0000	Upper Alt	0.33926	0.69356	0.90000
1.0000	Upper Alt	0.33927	0.69357	0.90001

"Rej Null (Lower Alt)" and "Rej Null (Upper Alt)" under the heading "Source" indicate the probabilities of rejecting the null hypothesis for the lower alternative and for the upper alternative, respectively. "Reject Null" indicates the probability of rejecting the null hypothesis for either the lower or upper alternative.

Note that with the STOP=REJECT option, the cumulative stopping probability of accepting the null hypothesis H_0 at each interim stage is zero and is not displayed.

With the PSS(CREF=0 0.5 1.0) option, the "Power and Expected Sample Sizes" table in Output 77.11.4 displays powers and expected sample sizes under hypothetical references $\theta = 0$ (null hypothesis H_0), $\theta = 0.5\,\theta_1$, and $\theta = \theta_1$ (alternative hypothesis H_1), where θ_1 is the alternative reference. The expected sample sizes are displayed in a percentage scale relative to the corresponding fixed-sample size design.

Output 77.11.4 Power and Expected Sample Size Information

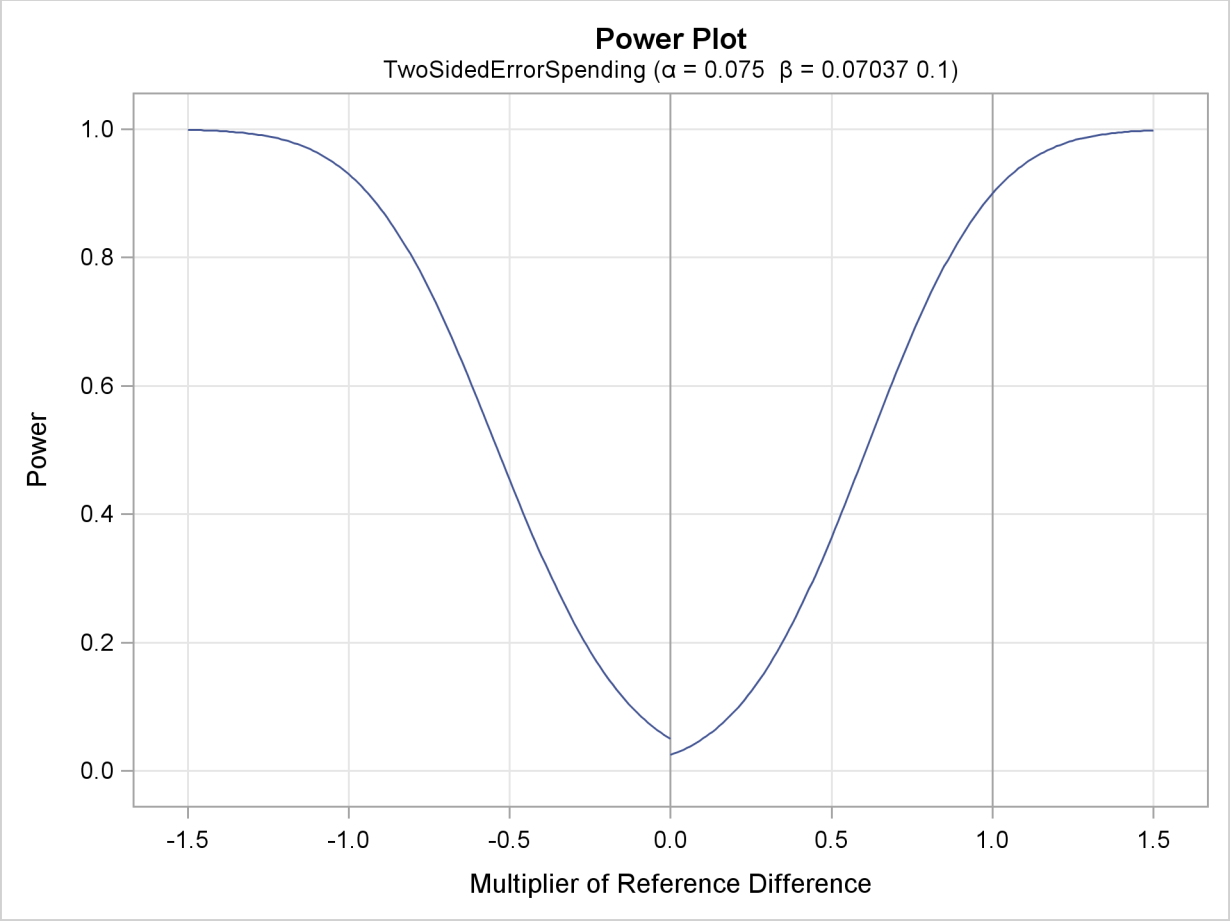
Powers and Expected Sample Sizes
Reference = CRef * (Alt Reference)

CRef	Ref	Power	-Sample Size- Percent Fixed-Sample
0.0000	Lower Alt	0.05000	100.4877
0.5000	Lower Alt	0.45370	88.5090
1.0000	Lower Alt	0.92963	64.8288
0.0000	Upper Alt	0.02500	100.4877
0.5000	Upper Alt	0.36458	96.2309
1.0000	Upper Alt	0.90000	75.9878

Note that at $c_i = 0$, the null reference $\theta = 0$, the power with the lower alternative is the lower α error 0.05, and the power with the upper alternative is the upper α error 0.025. At $c_i = 1$, the alternative reference $\theta = \theta_1$, the power with the upper alternative is the specified power 0.90, and the power with the lower alternative 0.92963 is greater than the specified power 0.90 because the same information level is used for these two asymmetric boundaries.

With the PLOTS=POWER option, the procedure displays a plot of the power curves under various hypothetical references, as shown in Output 77.11.5. By default, powers under the lower hypotheses $\theta = c_i\,\theta_{1l}$ and under the upper hypotheses $\theta = c_i\,\theta_{1u}$ are displayed for a two-sided asymmetric design, where $c_i = 0, 0.01, 0.02, \ldots, 1.50$ and $\theta_{1l} = -1$ and $\theta_{1u} = 1$ are the lower and upper alternative references, respectively.

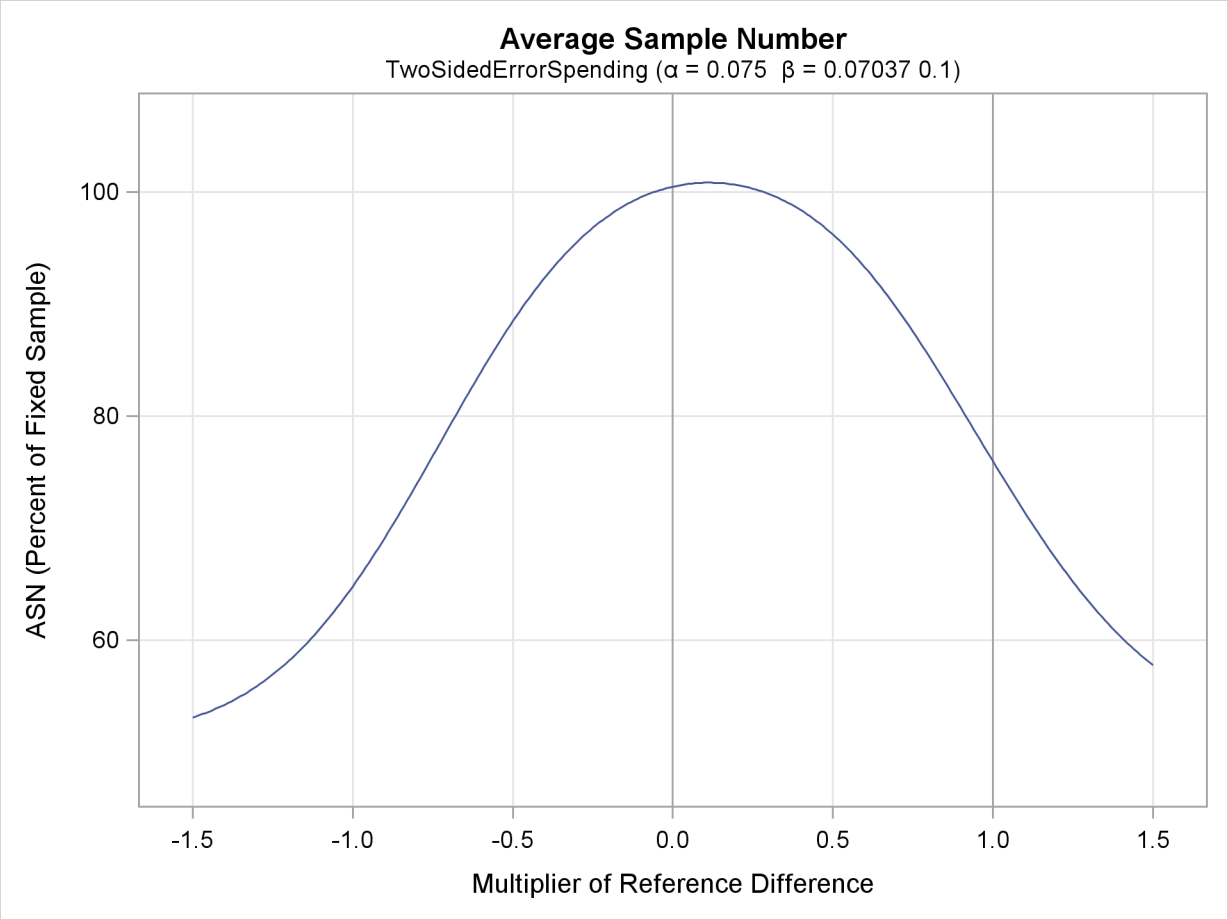
Output 77.11.5 Power Plot



The horizontal axis displays the multiplier of the reference difference. A positive multiplier corresponds to c_i for the upper alternative hypothesis, and a negative multiplier corresponds to $-c_i$ for the lower alternative hypothesis. For lower reference hypotheses, the power is the lower α error 0.05 under the null hypothesis ($c_i = 0$) and is 0.92963 under the alternative hypothesis ($c_i = 1$). For upper reference hypotheses, the power is the upper α error 0.025 under the null hypothesis ($c_i = 0$) and is 0.90 under the alternative hypothesis ($c_i = 1$).

With the PLOTS=ASN option, the procedure displays a plot of expected sample sizes under various hypothetical references, as shown in Output 77.11.6. By default, expected sample sizes under the lower hypotheses $\theta = c_i\,\theta_{1l}$ and under the upper hypotheses $\theta = c_i\,\theta_{1u}$, $c_i = 0, 0.01, 0.02, \ldots, 1.50$, are displayed for a two-sided asymmetric design, where $\theta_{1l} = -1$ and $\theta_{1u} = 1$ are the lower and upper alternative references, respectively.

Output 77.11.6 ASN Plot



The horizontal axis displays the multiplier of the reference difference. A positive multiplier corresponds to c_i for the upper alternative hypothesis and a negative multiplier corresponds to $-c_i$ for the lower alternative hypothesis.

The "Boundary Information" table in Output 77.11.7 displays the information levels, alternative references, and boundary values. The default BOUNDARYSCALE=STDZ option specifies that the standardized Z scale be used to display the alternative references and boundary values. The resulting standardized alternative references at stage k are given by $\pm\theta_1\sqrt{I_k}$, where θ_1 is the specified alternative reference and I_k is the information level at stage k, $k = 1, 2, 3$.

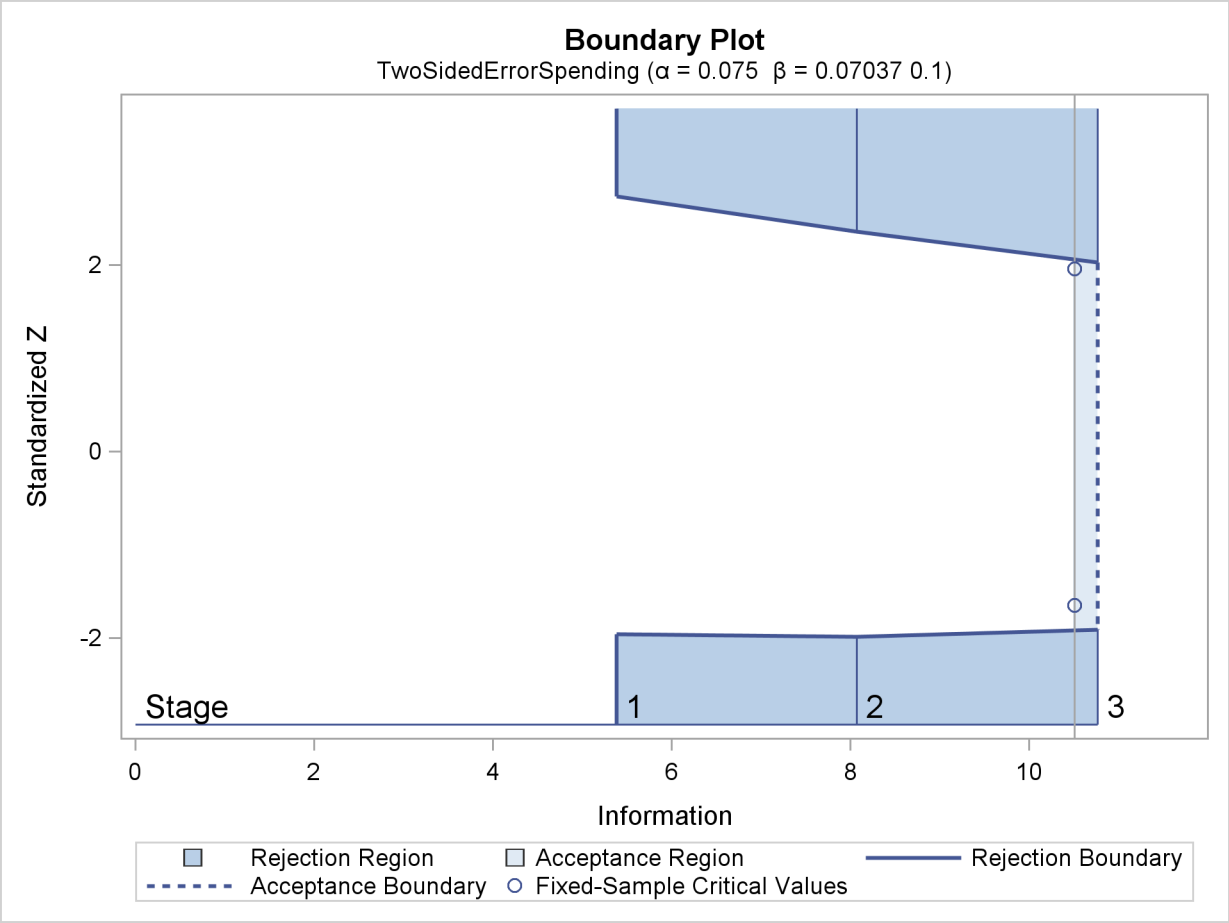
Output 77.11.7 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

Stage	--Information Level- Proportion	Actual	------Alternative----- -------Reference------ Lower	Upper	----Boundary Values--- ---Lower-- Alpha	---Upper-- Alpha
1	0.5000	5.381827	-2.31988	2.31988	-1.95996	2.73437
2	0.7500	8.07274	-2.84126	2.84126	-1.98394	2.35681
3	1.0000	10.76365	-3.28080	3.28080	-1.90855	2.02853

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 77.11.8.

Output 77.11.8 Boundary Plot



The "Error Spending Information" table in Output 77.11.9 displays the cumulative error spending at each stage for each boundary.

Output 77.11.9 Error Spending Information

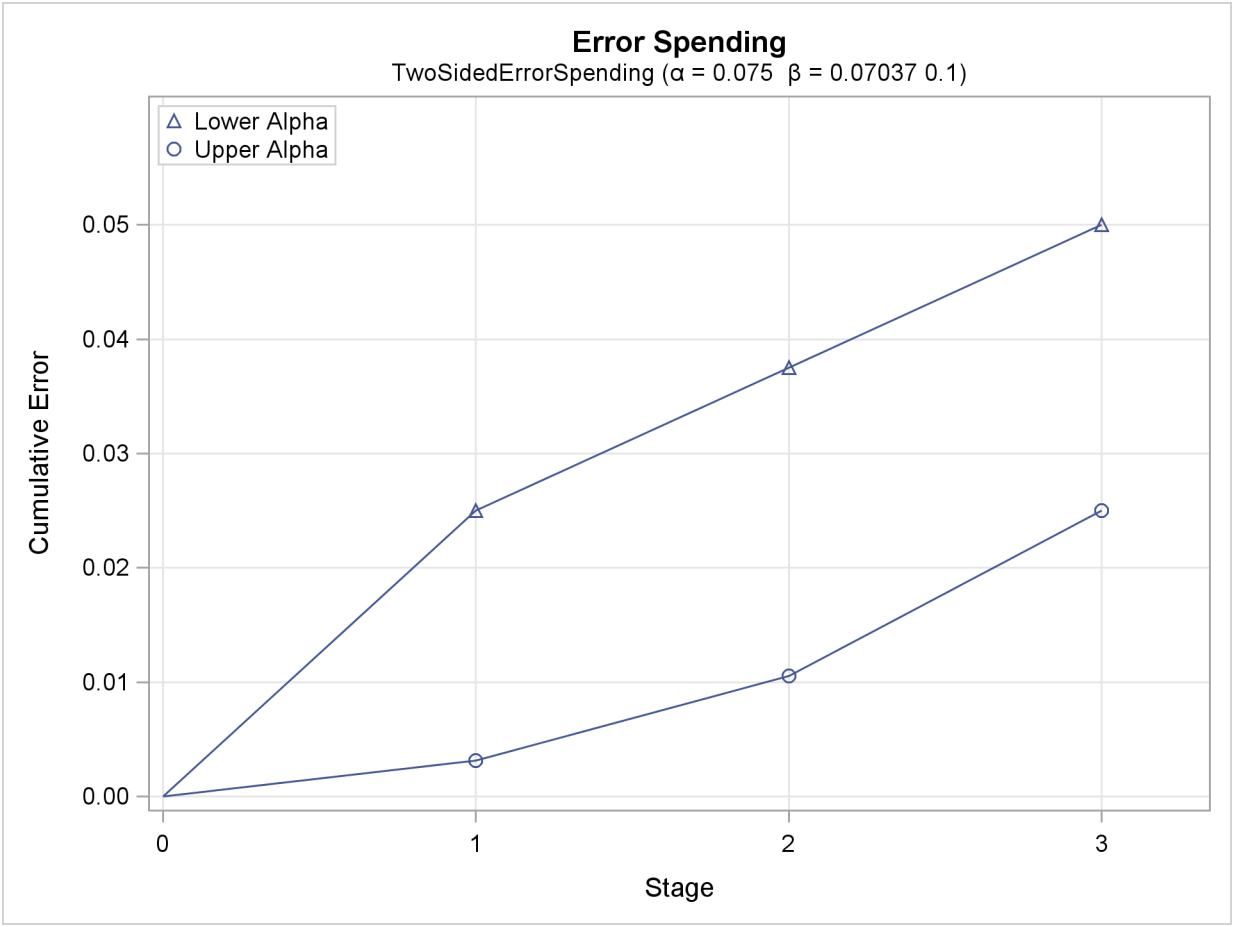
Error Spending Information

Stage	-Information Level- Proportion	--------Cumulative Error Spending------- -------Lower------ Alpha	Beta	-------Upper------ Beta	Alpha
1	0.5000	0.02500	0.00000	0.00001	0.00313
2	0.7500	0.03750	0.00000	0.00001	0.01055
3	1.0000	0.05000	0.07037	0.10000	0.02500

With the STOP=REJECT option, there is no early stopping to accept H_0, and the corresponding β spending at an interim stage is computed from the rejection region. For example, the upper β spending at stage 1 (0.00001) is the probability of rejecting H_0 for the lower alternative under the upper alternative reference.

With the PLOTS=ERRSPEND option, the procedure displays a plot of the cumulative error spending on each boundary at each stage, as shown in Output 77.11.10.

Output 77.11.10 Error Spending Plot



Example 77.12: Creating a Two-Sided Asymmetric Error Spending Design with Early Stopping to Reject or Accept H_0

This example requests a four-stage two-sided asymmetric group sequential design for normally distributed statistics. The O'Brien-Fleming boundary can be approximated by a gamma family error spending function with parameter $\gamma = -4$ or -5, and the Pocock boundary can be approximated with parameter $\gamma = 1$ (Hwang, Shih, and DeCani 1990, p. 1440). The following statements use the gamma error spending function with early stopping to reject or accept the null hypothesis H_0:

```
ods graphics on;
proc seqdesign altref=2
               pss(cref=0 0.5 1)
               stopprob(cref=0 1)
               errspend
               plots=(asn power errspend)
               ;
TwoSidedAsymmetric: design nstages=4
                    method=errfuncgamma(gamma=1)
                    method(beta)=errfuncgamma(gamma=-2)
                    method(upperalpha)=errfuncgamma(gamma=-5)
                    alt=twosided
                    stop=both
                    beta=0.1
                    ;
run;
ods graphics off;
```

The design uses gamma family error spending functions with $\gamma = -5$ for the upper α boundary, $\gamma = 1$ for the lower α boundary, and $\gamma = -2$ for the lower and upper β boundaries.

The "Design Information" table in Output 77.12.1 displays design specifications and the derived maximum information. Note that in order to attain the same information level for the asymmetric lower and upper boundaries, the derived power at the upper alternative 0.93655 is larger than the specified $1 - \beta = 0.90$.

Output 77.12.1 Design Information

The SEQDESIGN Procedure
Design: TwoSidedAsymmetric

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Accept/Reject Null
Method	Error Spending
Boundary Key	Both
Alternative Reference	2
Number of Stages	4
Alpha	0.05
Beta (Lower)	0.1
Beta (Upper)	0.06345
Power (Lower)	0.9
Power (Upper)	0.93655
Max Information (Percent of Fixed Sample)	104.0688
Max Information	3.162386
Null Ref ASN (Percent of Fixed Sample)	74.16654
Lower Alt Ref ASN (Percent of Fixed Sample)	59.10271
Upper Alt Ref ASN (Percent of Fixed Sample)	73.78797

The "Method Information" table in Output 77.11.2 displays the specified α and β error levels and the derived drift parameter. With the same information level used for the asymmetric lower and upper boundaries, only one of the β levels is maintained and the other is derived to have the level less than or equal to the specified level.

Output 77.12.2 Method Information

Method Information

Boundary	Method	Alpha	Beta	----Error Spending---- Function
Upper Alpha	Error Spending	0.02500	.	Gamma (Gamma=-5)
Upper Beta	Error Spending	.	0.06345	Gamma (Gamma=-2)
Lower Beta	Error Spending	.	0.10000	Gamma (Gamma=-2)
Lower Alpha	Error Spending	0.02500	.	Gamma (Gamma=1)

Method Information

Boundary	Alternative Reference	Drift
Upper Alpha	2	3.55662
Upper Beta	2	3.55662
Lower Beta	-2	-3.55662
Lower Alpha	-2	-3.55662

With the STOPPROB(CREF=0 1) option, the "Expected Cumulative Stopping Probabilities" table in Output 77.12.3 displays the expected stopping stage and cumulative stopping probabilities at each stage under the null reference $\theta = 0$ and under the alternative reference $\theta = \theta_1$.

Output 77.12.3 Stopping Probabilities

Expected Cumulative Stopping Probabilities
Reference = CRef * (Alt Reference)

CRef	Ref	Expected Stopping Stage	Source
0.0000	Lower Alt	2.851	Rej Null (Lower Alt)
0.0000	Lower Alt	2.851	Rej Null (Upper Alt)
0.0000	Lower Alt	2.851	Reject Null
0.0000	Lower Alt	2.851	Accept Null
0.0000	Lower Alt	2.851	Total
1.0000	Lower Alt	2.272	Rej Null (Lower Alt)
1.0000	Lower Alt	2.272	Rej Null (Upper Alt)
1.0000	Lower Alt	2.272	Reject Null
1.0000	Lower Alt	2.272	Accept Null
1.0000	Lower Alt	2.272	Total
0.0000	Upper Alt	2.851	Rej Null (Lower Alt)
0.0000	Upper Alt	2.851	Rej Null (Upper Alt)
0.0000	Upper Alt	2.851	Reject Null
0.0000	Upper Alt	2.851	Accept Null
0.0000	Upper Alt	2.851	Total
1.0000	Upper Alt	2.836	Rej Null (Lower Alt)
1.0000	Upper Alt	2.836	Rej Null (Upper Alt)
1.0000	Upper Alt	2.836	Reject Null
1.0000	Upper Alt	2.836	Accept Null
1.0000	Upper Alt	2.836	Total

Expected Cumulative Stopping Probabilities
Reference = CRef * (Alt Reference)

		Stopping Probabilities			
CRef	Ref	Stage_1	Stage_2	Stage_3	Stage_4
0.0000	Lower Alt	0.00875	0.01556	0.02087	0.02500
0.0000	Lower Alt	0.00042	0.00190	0.00704	0.02500
0.0000	Lower Alt	0.00917	0.01746	0.02791	0.05000
0.0000	Lower Alt	0.00000	0.30125	0.79354	0.95000
0.0000	Lower Alt	0.00917	0.31870	0.82145	1.00000
1.0000	Lower Alt	0.27499	0.58934	0.79601	0.90000
1.0000	Lower Alt	0.00000	0.00000	0.00000	0.00000
1.0000	Lower Alt	0.27499	0.58934	0.79601	0.90000
1.0000	Lower Alt	0.00000	0.01863	0.04935	0.10000
1.0000	Lower Alt	0.27499	0.60797	0.84536	1.00000
0.0000	Upper Alt	0.00875	0.01556	0.02087	0.02500
0.0000	Upper Alt	0.00042	0.00190	0.00704	0.02500
0.0000	Upper Alt	0.00917	0.01746	0.02791	0.05000
0.0000	Upper Alt	0.00000	0.30125	0.79354	0.95000
0.0000	Upper Alt	0.00917	0.31870	0.82145	1.00000
1.0000	Upper Alt	0.00002	0.00002	0.00002	0.00002
1.0000	Upper Alt	0.05945	0.33802	0.72323	0.93655
1.0000	Upper Alt	0.05947	0.33804	0.72325	0.93657
1.0000	Upper Alt	0.00000	0.01182	0.03131	0.06343
1.0000	Upper Alt	0.05947	0.34986	0.75456	1.00000

"Rej Null (Lower Alt)" and "Rej Null (Upper Alt)" under the heading "Source" indicate the probabilities of rejecting the null hypothesis for the lower alternative and for the upper alternative, respectively. "Reject Null" indicates the probability of rejecting the null hypothesis for either the lower or upper alternative, "Accept Null" indicates the probability of accepting the null hypothesis, and "Total" indicates the total probability of stopping the trial.

With the PSS(CREF=0 0.5 1.0) option, the "Power and Expected Sample Sizes" table in Output 77.12.4 displays powers and expected sample sizes under hypothetical references $\theta = 0$ (null hypothesis H_0), $\theta = 0.5\,\theta_1$, and $\theta = \theta_1$ (alternative hypothesis H_1), where θ_1 is the alternative reference. The expected sample sizes are displayed in a scale that indicates a percentage of its corresponding fixed-sample size design.

Output 77.12.4 Power and Expected Sample Size Information

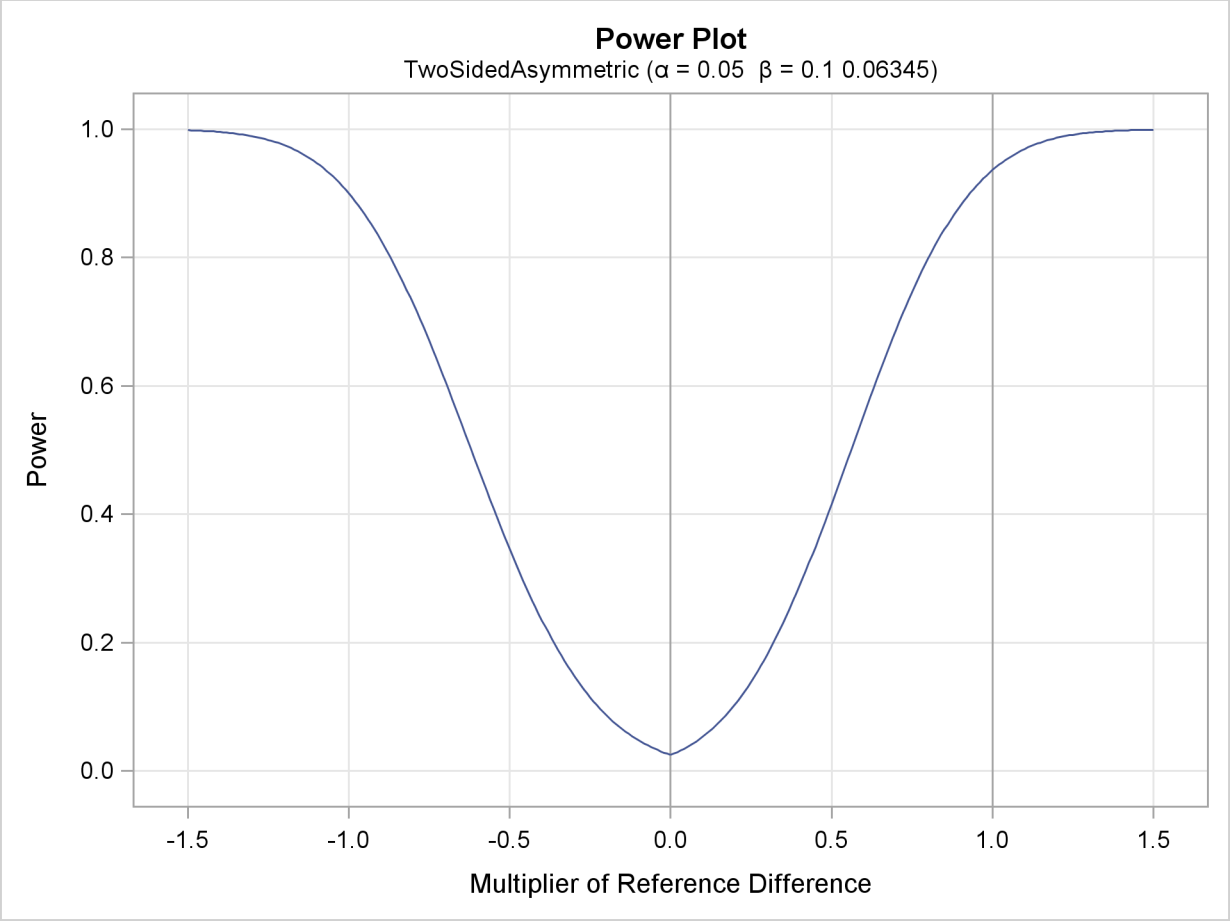
Powers and Expected Sample Sizes
Reference = CRef * (Alt Reference)

CRef	Ref	Power	-Sample Size- Percent Fixed-Sample
0.0000	Lower Alt	0.02500	74.1665
0.5000	Lower Alt	0.34601	75.8425
1.0000	Lower Alt	0.90000	59.1027
0.0000	Upper Alt	0.02500	74.1665
0.5000	Upper Alt	0.41647	85.3976
1.0000	Upper Alt	0.93655	73.7880

Note that at $c_i = 0$, the null reference $\theta = 0$, the power with the lower alternative is the lower α error 0.025, and the power with the upper alternative is the upper α error 0.025. At $c_i = 1$, the alternative reference $\theta = \theta_1$, the power with the lower alternative is the specified power 0.90, and the power with the upper alternative 0.93655 is greater than the specified power 0.90 because the same information level is used for these two asymmetric boundaries.

With the PLOTS=POWER option, the procedure displays a plot of the power curves under various hypothetical references, as shown in Output 77.12.5. By default, powers under the lower hypotheses $\theta = c_i\,\theta_{1l}$ and under the upper hypotheses $\theta = c_i\,\theta_{1u}$, are displayed for a two-sided asymmetric design, where $c_i = 0, 0.01, 0.02, \ldots, 1.50$ and $\theta_{1l} = -1$ and $\theta_{1u} = 1$ are the lower and upper alternative references, respectively.

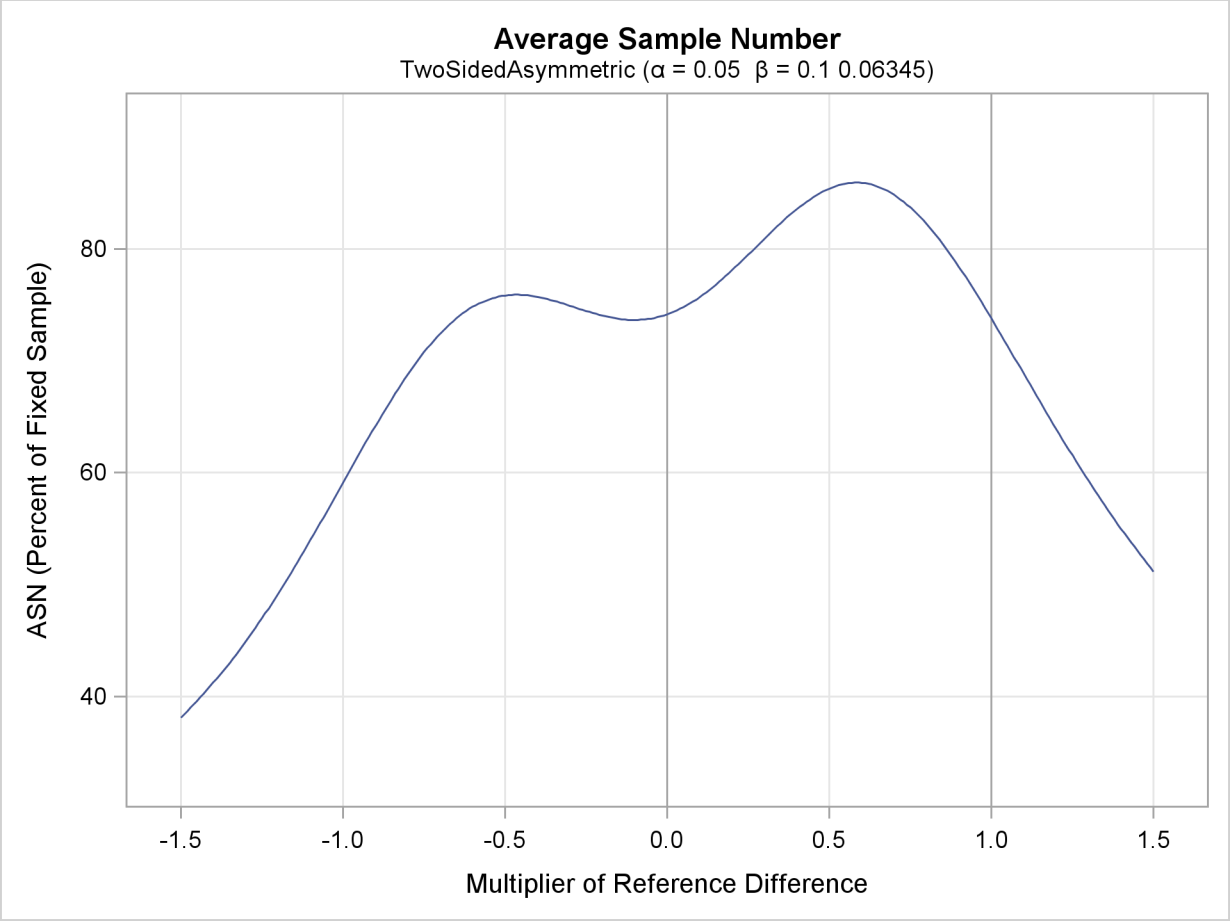
Output 77.12.5 Power Plot



The horizontal axis displays the multiplier of the reference difference. A positive multiplier corresponds to c_i for the upper alternative hypothesis, and a negative multiplier corresponds to $-c_i$ for the lower alternative hypothesis. For lower reference hypotheses, the power is the lower α error 0.025 under the null hypothesis ($c_i = 0$) and is 0.90 under the alternative hypothesis ($c_i = 1$). For upper reference hypotheses, the power is the upper α error 0.025 under the null hypothesis ($c_i = 0$) and is 0.93655 under the alternative hypothesis ($c_i = 1$).

With the PLOTS=ASN option, the procedure displays a plot of expected sample sizes under various hypothetical references, as shown in Output 77.12.6. By default, expected sample sizes under the lower hypotheses $\theta = c_i\,\theta_{1l}$ and under the upper hypotheses $\theta = c_i\,\theta_{1u}$ are displayed for a two-sided asymmetric design, where $c_i = 0, 0.01, 0.02, \ldots, 1.50$ and $\theta_{1l} = -1$ and $\theta_{1u} = 1$ are the lower and upper alternative references, respectively.

Output 77.12.6 ASN Plot



The horizontal axis displays the multiplier of the reference difference. A positive multiplier corresponds to c_i for the upper alternative hypothesis, and a negative multiplier corresponds to $-c_i$ for the lower alternative hypothesis.

By default, or when you specify BETAOVERLAP=ADJUST, the SEQDESIGN procedure first derives boundary values without adjusting for the possible overlapping of the two one-sided β boundaries based on two corresponding one-sided tests. Then the procedure checks for overlapping of the β boundaries at the interim stages. Since the two β boundaries overlap at stage 1, the β boundary values for stage 1 are set to missing, the β spending values at stage 1 are set to zero, and the β spending values at subsequent stages are adjusted proportionally.

The "Boundary Information" table in Output 77.12.7 displays the information levels, alternative references, and boundary values. The default BOUNDARYSCALE=STDZ option specifies that the standardized Z scale be used to display the alternative references and boundary values. The resulting standardized alternative references at stage k is given by $\pm\theta_1\sqrt{I_k}$, where θ_1 is the specified alternative reference and I_k is the information level at stage k, $k = 1, 2, 3, 4$.

Output 77.12.7 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

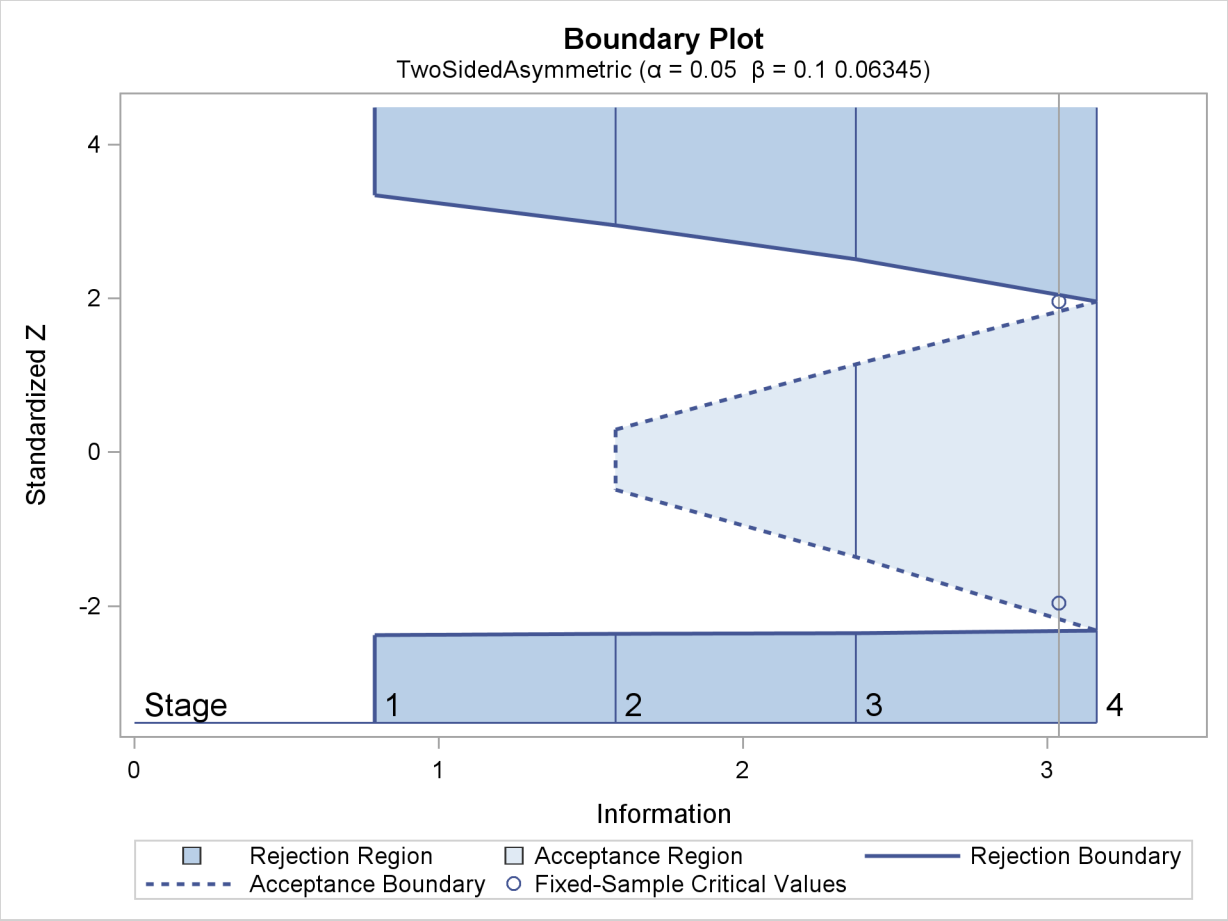
Stage	---Information Level-- Proportion	Actual	-------Alternative------ --------Reference------- Lower	Upper
1	0.2500	0.790597	-1.77831	1.77831
2	0.5000	1.581193	-2.51491	2.51491
3	0.7500	2.37179	-3.08012	3.08012
4	1.0000	3.162386	-3.55662	3.55662

Boundary Information (Standardized Z Scale)
Null Reference = 0

	----------Boundary Values---------- ----------Lower----------		----------Upper----------	
Stage	Alpha	Beta	Beta	Alpha
1	-2.37610	.	.	3.33772
2	-2.35714	-0.48408	0.29400	2.94871
3	-2.34861	-1.36183	1.13898	2.50473
4	-2.32105	-2.32105	1.95675	1.95675

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 77.12.8.

Output 77.12.8 Boundary Plot



The "Error Spending Information" in Output 77.12.9 displays the cumulative error spending at each stage for each boundary.

Output 77.12.9 Error Spending Information

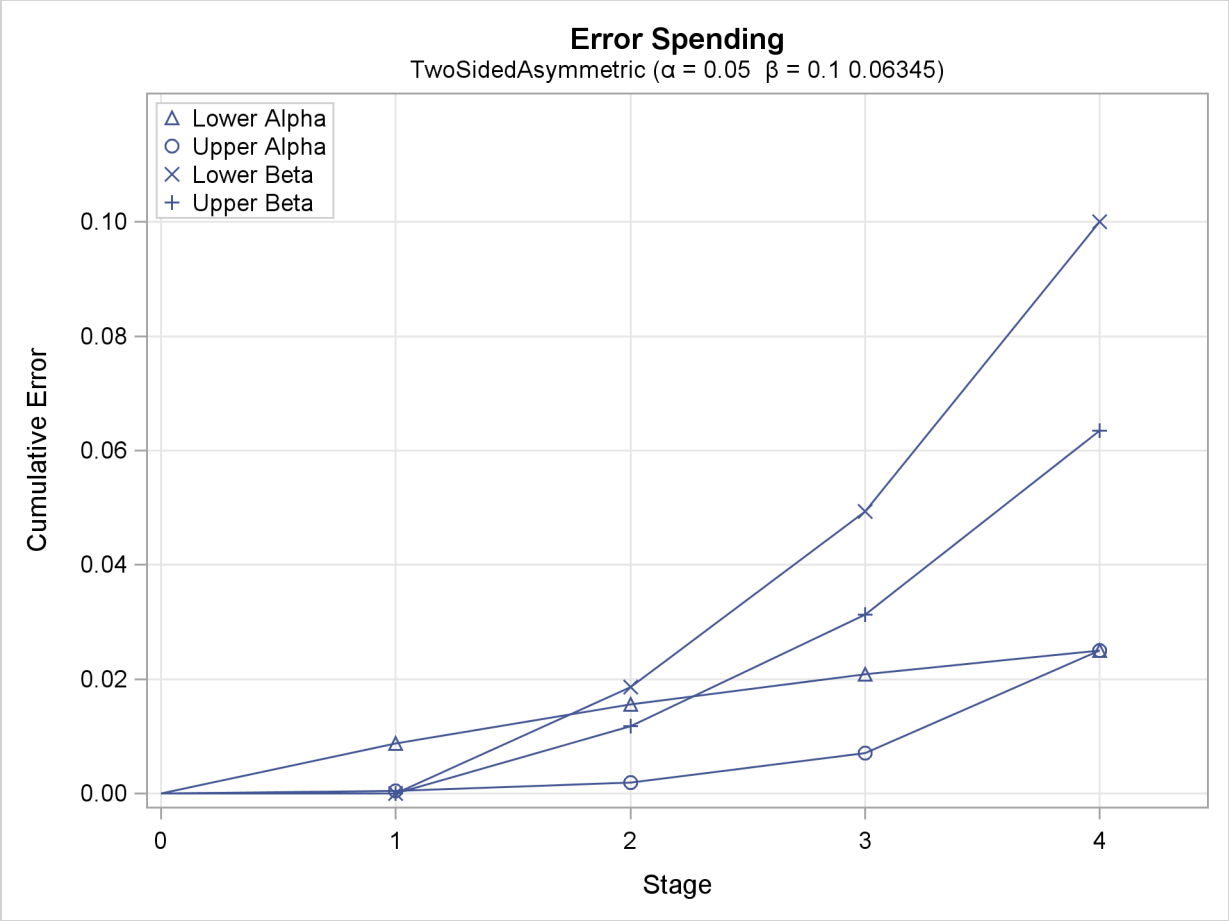
Error Spending Information

		Cumulative Error Spending			
	Information Level	Lower		Upper	
Stage	Proportion	Alpha	Beta	Beta	Alpha
1	0.2500	0.00875	0.00000	0.00002	0.00042
2	0.5000	0.01556	0.01863	0.01184	0.00190
3	0.7500	0.02087	0.04935	0.03132	0.00704
4	1.0000	0.02500	0.10000	0.06345	0.02500

With the β boundary values missing at stage 1, there is no early stopping to accept H_0 at stage 1, and the corresponding β spending at stage 1 is computed from the rejection region. For example, the upper β spending at stage 1 (0.00002) is the probability of rejecting H_0 for the lower alternative under the upper alternative reference.

With the PLOTS=ERRSPEND option, the procedure displays a plot of the cumulative error spending on each boundary at each stage, as shown in Output 77.12.10.

Output 77.12.10 Error Spending Plot



Acknowledgments

In addition to being shaped by the research literature listed in the section "References" on page 5979, the development of the SEQDESIGN and SEQTEST procedures has benefited significantly from the advice and expertise of the following researchers:

- Lu Cui, Eisai Medical Research
- Alex Dmitrienko, Eli Lilly
- Scott Emerson, University of Washington
- Gordon Lan, Johnson & Johnson
- Steve Snapinn, Amgen
- John Whitehead, University of Reading

The time and effort that these researchers have contributed is gratefully acknowledged.

References

Armitage, P., McPherson, C. K., and Rowe, B. C. (1969), "Repeated Significance Test on Accumulating Data," *Journal of the Royal Statistical Society, Series A*, 132, 235–244.

Chow, S. C. and Liu, J. P. (1998), *Design and Analysis of Clinical Trials, Concept and Methodologies,* New York: John Wiley & Sons.

Chow, S. C., Shao, J., and Wang, H. (2003), *Sample Size Calculations in Clinical Research,* Boca Raton, FL: CRC Press.

Cox, D. R. and Hinkley, D. V.(1974), *Theoretical Statistics,* London: Chapman & Hall.

DeMets, D. L., Furberg, C. D., and Friedman, L. M. (2006), *Data Monitoring in Clinical Trials,* New York: Springer.

Diggle, P. J., Heagerty, P., Liang, K. Y., and Zeger, S. L. (2002), *Analysis of Longitudinal Data*, Second Edition, New York: Oxford University Press.

Dmitrienko, A., Molenberghs, G., Chuang-Stein, C., and Offen, W. (2005), *Analysis of Clinical Trials Using SAS: A Practical Guide,* Cary, NC: SAS Institute.

Efron, B. and Hinkley, D. V.(1978), "Assessing the Accuracy of the Maximum Likelihood Estimator: Observed versus Expected Fisher Information," *Biometrika*, 65, 457–483.

Ellenberg, S. S., Fleming, T. R., and DeMets, D. L. (2003), *Data Monitoring Committees in Clinical Trials,* New York: John Wiley & Sons.

Emerson, S. S. (1996), "Statistical Packages for Group Sequential Methods," *The American Statistician*, 50, 183–192.

Emerson, S. S. and Fleming, T. R. (1989), "Symmetric Group Sequential Designs," *Biometrics*, 45, 905–923.

Emerson, S. S., Kittelson, J. M., and Gillen, D. L. (2005), "On the Use of Stochastic Curtailment in Group Sequential Clinical Trials," *UW Biostatistics Working Paper Series*, *http://www.bepress.com/uwbiostat/paper243*.

Food and Drug Administration (1998), "E9: Statistical Principles for Clinical Trials," *Federal Register*, 63 (179), 49583–49598.

Haybittle, J. L. (1971), "Repeated Assessment of Results in Clinical Trials of Cancer Treatment," *Brit. J. Radiology*, 44, 793–797.

Hsieh, F. Y. and Lavori, P. W. (2000), "Sample-Size Calculations for the Cox Proportional Hazards Regression Model with Nonbinary Covariates," *Controlled Clinical Trials*, 21, 552–560.

Hwang, I. K., Shih, W. J., and DeCani, J. S. (1990), "Group Sequential Designs Using a Family of Type I Error Probability Spending Functions," *Statistics in Medicine*, 9, 1439–1445.

Jennison, C. and Turnbull, B. W. (1990), "Statistical Approaches to Interim Monitoring of Medical Trials: A Review and Commentary," *Statistical Science*, 5, 299–317.

Jennison, C. and Turnbull, B. W. (2000), *Group Sequential Methods with Applications to Clinical Trials,* New York: Chapman & Hall.

Kalbfleisch, J. D. and Prentice, R. L. (1980), *The Statistical Analysis of Failure Time Data,* New York: John Wiley & Sons.

Kim, K. and DeMets, D. L. (1987), "Design and Analysis of Group Sequential Tests Based on the Type I Error Spending Rate Function," *Biometrika*, 74, 149–154.

Kim, K. and Tsiatis, A. A. (1990), "Study Duration for Clinical Trials with Survival Response and Early Stopping Rule," *Biometrics*, 46, 81–92.

Kittelson, J. M. and Emerson, S. S. (1999), "A Unifying Family of Group Sequential Test Designs," *Biometrics*, 55, 874–882.

Lan, K. K. G. and DeMets, D. L. (1983), "Discrete Sequential Boundaries for Clinical Trials," *Biometrika*, 70, 659–663.

Lan, K. K. G., Lachin, J. M, and Bautista, O. (2003), "Over-ruling a Group Sequential Boundary: A Stopping Rule versus a Guideline," *Statistics in Medicine*, 22, 3347–3355.

Lan, K. K. G., Simon, R., and Halperin, M. (1982), "Stochastically Curtailed Tests in Long-Term Clinical Trials," *Sequential Analysis*, 1, 207–219.

Lindgren, B. W. (1976), *Statistical Theory,* Third Edition, New York: Macmillan.

McCullagh, P. and Nelder, J. A. (1989), *Generalized Linear Models,* Second Edition, New York: Chapman & Hall/CRC.

Mehta, C. R. and Tsiatis, A. A. (2001), "Flexible Sample Size Considerations under Information Based Interim Monitoring," *Drug Information Journal*, 35, 1095–1112.

O'Brien, P. C. and Fleming, T. R. (1979), "A Multiple Testing Procedure for Clinical Trials," *Biometrics*, 35, 549–556.

O'Neill, R. T. (1994), "Interim Analysis, A Regulatory Perspective on Data Monitoring and Interim Analysis," *Statistics in the Pharmaceutical Industry,* Revised and Expanded Second Edition, ed. C. R. Buncher and J-Y Tsay, New York: Marcel Dekker, 285–290.

Pampallona, S. and Tsiatis, A. A. (1994), "Group Sequential Designs for One-Sided and Two-Sided Hypothesis Testing with Provision for Early Stopping in Favor of the Null Hypothesis," *J. Statist. Planning and Inference*, 42, 19–35.

Peto, R., Pike, M. C., et al. (1976), "Design and Analysis of Randomized Clinical Trials Requiring Prolonged Observation of Each Patient: I. Introduction and Design," *British Journal of Cancer*, 34, 585–612.

Pocock, S. J. (1977), "Group Sequential Methods in the Design and Analysis of Clinical Trials," *Biometrika*, 64, 191–199.

Pocock, S. J. (1982), "Interim Analyses for Randomized Clinical Trials: The Group Sequential Approach," *Biometrics*, 38, 153–162.

Pocock, S. J. and White, I. (1999), "Trials Stopped Early: Too Good to Be True?" *Lancet*, 353, 943–944.

Proschan, M. A., Lan, K. K. G., and Wittes, J. T. (2006), *Statistical Monitoring of Clinical Trials,* New York: Springer.

Rudser, K.D. and Emerson, S.S. (2007), "Implementing Type I & Type II Error Spending for Two-Sided Group Sequential Designs," *Contemporary Clinical Trials*, doi:10.1016/j.cct.2007.09.002.

Schoenfeld, D. A. (1983), "Sample-Size Formula for the Proportional-Hazards Regression Model," *Biometrics*, 39, 499–503.

Senn, S. (1997), *Statistical Issues in Drug Development,* New York: John Wiley & Sons.

Snapinn, S. M. (2000), "Noninferiority Trials," *Current Controlled Trials in Cardiovascular Medicine*, 1, 19–21.

Wang, S. K. and Tsiatis, A. A. (1987), "Approximately Optimal One-Parameter Boundaries for Group Sequential Trials," *Biometrics*, 43, 193–200.

Ware, J. H., Muller, J. E., and Braunwald, E. (1985), "The Futility Index: An Approach to the Cost-Effective Termination of Randomized Clinical Trials" *American Journal of Medicine*, 78, 635–643.

Whitehead, J. (1997), *The Design and Analysis of Sequential Clinical Trials,* Revised Second Edition, Chichester: John Wiley & Sons.

Whitehead, J. (2001), "Use of the Triangular Test in Sequential Clinical Trials," *Handbook of Statistics in Clinical Oncology,* ed. J. Crowley, New York: Marcel Dekker, 211–228.

Whitehead, J. and Jones, D. R. (1979), "The Analysis of Sequential Clinical Trials," *Biometrika*, 66, 443–452.

Whitehead, J. and Stratton, I. (1983), "Group Sequential Clinical Trials with Triangular Continuation Regions," *Biometrics*, 39, 227–236.

Chapter 78

The SEQTEST Procedure

Contents

Overview: SEQTEST Procedure **5984**
Getting Started: SEQTEST Procedure **5988**
Syntax: SEQTEST Procedure **6003**
PROC SEQTEST Statement 6003
Details: SEQTEST Procedure **6012**
Input Data Sets 6012
Boundary Variables 6015
Boundary Adjustments for Information Levels 6017
Boundary Adjustments for Minimum Error Spending 6019
Boundary Adjustments for Overlapping Lower and Upper β Boundaries . . 6020
Stochastic Curtailment 6021
Repeated Confidence Intervals 6022
Analysis after a Sequential Test 6024
Available Sample Space Orderings in a Sequential Test 6025
Applicable Tests and Sample Size Computation 6026
Table Output 6028
ODS Table Names 6032
Graphics Output 6032
ODS Graphics 6034
Examples: SEQTEST Procedure **6034**
Example 78.1: Testing the Difference between Two Proportions 6035
Example 78.2: Testing an Effect in a Regression Model 6048
Example 78.3: Testing an Effect with Early Stopping to Accept H_0 6064
Example 78.4: Testing a Binomial Proportion 6079
Example 78.5: Comparing Two Proportions with a Log Odds Ratio Test . . 6092
Example 78.6: Comparing Two Survival Distributions with a Log-Rank Test 6106
Example 78.7: Testing an Effect in a Proportional Hazards Regression Model 6121
Example 78.8: Testing an Effect in a Logistic Regression Model 6137
Acknowledgments **6151**
References **6151**

Overview: SEQTEST Procedure

The purpose of the SEQTEST procedure is to perform interim analyses for clinical trials. Clinical trials are experiments on human beings to demonstrate the efficacy and safety of new drugs or treatments. A simple example is a trial to test the effectiveness of a new drug in humans by comparing the outcomes in a group of patients who receive the new drug with the outcomes in a comparable group of patients who receive a placebo.

A clinical trial is conducted according to a plan called a *protocol*. A protocol details the objectives of the trial, the data collection process, and the analyses of the data. The protocol contains information such as a null hypothesis and an alternative hypothesis, a test statistic, the probability α of a Type I error (incorrectly rejecting the null hypothesis), the probability β of a Type II error (incorrectly accepting the null hypothesis), the sample size needed to attain a specified power (probability of correctly rejecting the null hypothesis) of $1 - \beta$ at an alternative reference, and critical values that are associated with the test statistic for hypothesis testing.

In a fixed-sample trial, data about all individuals are first collected and then examined at the end of the study. Most major trials have data safety monitoring boards or data monitoring committees that periodically monitor safety and efficacy data during the trial and recommend that a trial be stopped for safety concerns such as an unacceptable toxicity level. In certain rare situations, the board or committee might even recommend that a trial be stopped for efficacy. In contrast to a fixed-sample trial, a group sequential trial provides for interim analyses before the formal completion of the trial while maintaining the specified overall Type I and Type II error probability levels.

A group sequential trial is most useful in situations where it is important to monitor the trial to prevent unnecessary exposure of patients to an unsafe new drug, or alternatively to a placebo treatment if the new drug shows significant improvement. If a group sequential trial stops early, then it usually requires fewer participants than a corresponding fixed-sample trial.

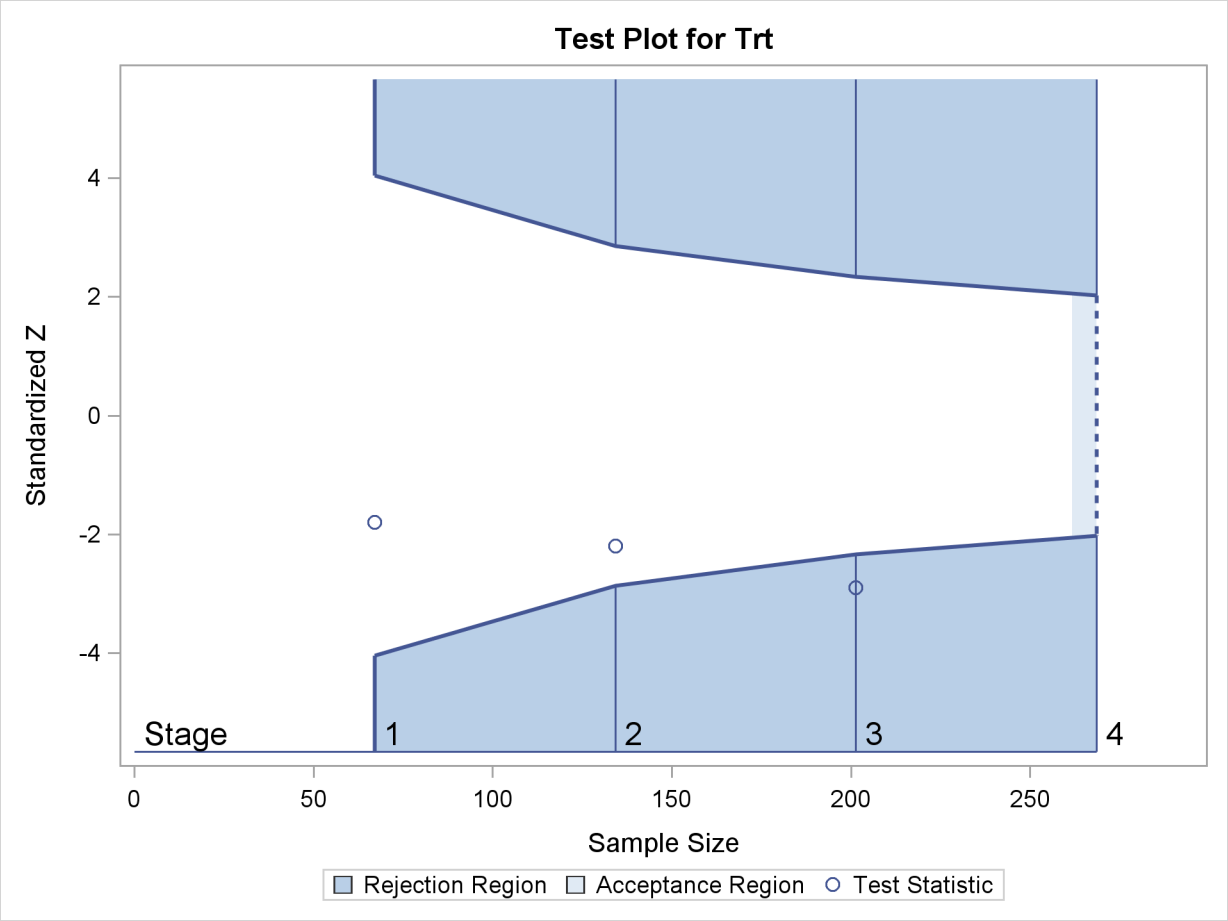
Thus, in most cases, if a group sequential trial stops early for safety of the new treatment, fewer patients will be exposed to the new treatment than in the fixed-sample trial. Also, if a trial stops early for efficacy of the new treatment, the new treatment will be available sooner than it would be in a fixed-sample trial. Furthermore, if a trial stops early, this can also save time and resources.

A group sequential design provides detailed specifications for a group sequential trial. In addition to the usual specification for a fixed-sample design, it provides the total number of stages (the number of interim stages plus a final stage) and a stopping criterion to reject, to accept, or to either reject or accept the null hypothesis at each interim stage. It also provides critical values and the sample size at each stage for the trial.

At each interim stage, the data collected at the current stage in addition to the data collected at previous stages are analyzed, and statistics such as a maximum likelihood test statistic and its associated standard error are computed. The test statistic is then compared with its corresponding critical values at the stage, and the trial is stopped or continued. If a trial continues to the final stage, the null hypothesis is either rejected or accepted. The critical values for each stage are chosen in such a way to maintain the overall α level, the overall β level, or both the overall α and β levels.

Figure 78.1 shows a two-sided symmetric group sequential trial that stops early to reject the null hypothesis that the parameter Trt is zero.

Figure 78.1 Sequential Plot for Two-Sided Test



The trial has four stages, which are indicated by vertical lines with accompanying stage numbers. With early stopping to reject the null hypothesis, the lower rejection boundary is constructed by connecting the lower critical values (boundary values) for the stages. Similarly, the upper rejection boundary is constructed by connecting the upper critical values for the stages. The horizontal axis indicates the sample size for the group sequential trial, and the vertical axis indicates the boundary values and test statistics on the standardized Z scale.

At each interim stage, if the standardized Z test statistic falls into a rejection region (the darker shaded areas in Figure 78.1), the trial stops and the null hypothesis is rejected. Otherwise, the trial continues to the next stage. At the final stage (stage 4), the trial is rejected if Z falls into a rejection region. Otherwise, the trial is accepted. In Figure 78.1, the test statistic does not fall into the rejection regions for stages 1 and 2, and so the trial continues to stage 3. At stage 3, the test statistic falls into the rejection region, and the null hypothesis is rejected.

A group sequential trial usually involves six steps:

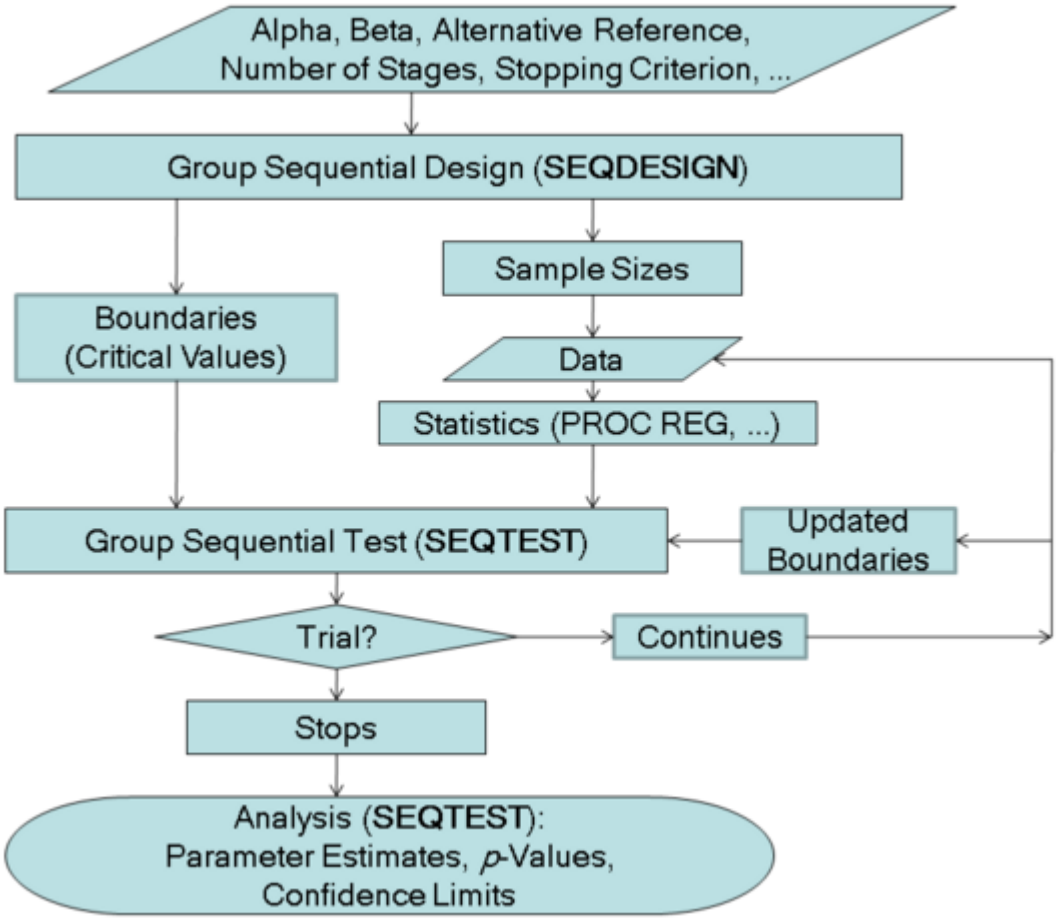
1. You specify the statistical details of the design, including the null and alternative hypotheses, a test statistic for the hypothesis test, the Type I and II error probabilities, a stopping criterion, the total number of stages, and the relative information level at each stage.
2. You compute the boundary values for the trial based on the specifications in Step 1. You also compute the sample size required at each stage for the specified hypothesis test.
3. At each stage, you collect additional data with the required sample sizes. The data available at each stage include the data collected at previous stages in addition to the data collected at the current stage.
4. At each stage, you analyze the available data with a procedure such as the REG procedure, and you compute the test statistic.
5. At each stage, you compare the test statistic with the corresponding boundary values. You stop the trial to reject or accept the hypothesis, or you continue the trial to the next stage. If you continue the trial to the final stage, you either accept or reject the hypothesis.
6. After the trial stops, you compute parameter estimates, confidence limits for the parameter, and a p-value for the hypothesis test.

You use the companion SEQDESIGN procedure at Step 2 to compute the boundary values and required sample sizes for the trial. You use the SEQTEST procedure at Step 5 to compare the test statistic with its boundary values. At stage 1, the boundary values are derived by using the boundary information tables created by the SEQDESIGN procedure. These boundary information tables are structured for input to the SEQTEST procedure. At each subsequent stage, the boundary values are derived by using the test information tables created by the SEQTEST procedure at the previous stage. These test information tables are also structured for input to the SEQTEST procedure. You also use the SEQTEST procedure at Step 6 to compute parameter estimates, confidence limits, and p-values after the trial stops.

Note that for some clinical trials, the information levels are derived from statistics based on individuals specified in the design plan and might not reach the target maximum information level. For example, if an estimate of the variance is used to compute the required sample size for a group sequential trial, the computed variance at each stage might not be the same as the estimated variance. Thus, instead of specifying the number of individuals in the protocol, the information level can be specified. You can then adjust the sample sizes with the updated variance estimate at interim stages to achieve the target maximum information level for the trial (Jennison and Turnbull 2000, p. 295).

The flowchart in Figure 78.2 summarizes the steps in a typical group sequential trial and the relevant SAS procedures.

Figure 78.2 Group Sequential Trial



Features of the SEQTEST Procedure

At each stage, the data are analyzed with a statistical procedure such as the REG procedure, and a test statistic and its associated information level are computed. The information level is the amount of information available about the unknown parameter. For a maximum likelihood statistic, the information level is the inverse of its variance.

At each stage, you use the SEQTEST procedure to compare the test statistic with its boundary values. At stage 1, the boundary values are derived by using the boundary information tables created by the SEQDESIGN procedure. At each subsequent stage, the boundary values are derived by using the test information tables created by the SEQTEST procedure at the previous stage.

If the observed information level does not match the corresponding information level in the BOUNDARY= data set, the SEQTEST procedure modifies the boundary values to adjust for new information levels at the current and subsequent stages. See the section "Boundary Adjustments for Information Levels" on page 6017 for a detailed description of these boundary adjustments.

Either you can specify the test statistic and its information level in the DATA= input data set, or you can specify the test statistic and its associated standard error in the PARMS= input data set. With

the PARMS= input data set, the information level for the test statistic is computed from its standard error. See the section "Input Data Sets" on page 6012 for a detailed description of these input data sets.

At the end of a trial, the parameter estimate is computed. The median unbiased estimate, confidence limits, and p-value depend on the specified sample space ordering. A sample space ordering specifies the ordering for test statistics that result in the stopping of a trial. That is, for all the statistics in the rejection region and in acceptance region, the SEQTEST procedure provides three different sample space orderings: the stagewise ordering uses counterclockwise ordering around the continuation region, the LR ordering uses the distance between the observed Z statistic z and its hypothetical value, and the MLE ordering uses the observed maximum likelihood estimate. See the section "Available Sample Space Orderings in a Sequential Test" on page 6025 for a detailed description of these orderings.

Output from the SEQTEST Procedure

In addition to the adjusted boundary values and test results for the group sequential trial, the SEQTEST procedure also computes the following quantities:

- average sample numbers (as percentages of the corresponding fixed-sample sizes for nonsurvival data or fixed-sample numbers of events for survival data) under various hypothetical references, including the null and alternative references
- stopping probabilities at each stage under various hypothetical references to indicate how likely it is that the trial will stop at that stage
- conditional power given the most recently observed statistic under specified hypothetical references
- predictive power given the most recently observed statistic
- repeated confidence intervals for the parameter from the observed statistic at each stage
- parameter estimate, p-value for hypothesis testing, and median and confidence limits for the parameter at the conclusion of a sequential trial

Getting Started: SEQTEST Procedure

The following example illustrates a clinical study that uses a two-sided O'Brien-Fleming design (O'Brien and Fleming 1979) to stop the trial early for ethical concerns about possible harm or for unexpectedly strong efficacy of the new drug.

Suppose that a pharmaceutical company is conducting a clinical trial to test the efficacy of a new cholesterol-lowering drug. The primary focus is low-density lipoprotein (LDL), the so-called bad cholesterol, which is a risk factor for coronary heart disease. LDL is measured in mg/dL, milligrams per deciliter of blood.

The trial consists of two groups of equally allocated patients with elevated LDL level: an experimental group given the new drug and a placebo control group. Suppose the changes in LDL level after the treatment for patients in the experimental and control groups are normally distributed with means μ_e and μ_c, respectively, and have a common variance σ^2. Then the null hypothesis of no effect for the new drug is $H_0 : \theta = \mu_e - \mu_c = 0$. Also suppose that the alternative reference $\theta = -10$ is the clinically meaningful difference that the trial should detect with a high probability (power), and that a good estimate of the standard deviation for the changes in LDL level is $\hat{\sigma} = 20$.

The following statements invoke the SEQDESIGN procedure and request a four-stage O'Brien-Fleming design for standardized normal test statistics:

```
ods graphics on;
proc seqdesign altref=-10.0;
  TwoSidedOBrienFleming: design nstages=4
                         method=obf
                         ;
  samplesize model=twosamplemean(stddev=20);
  ods output Boundary=Bnd_LDL;
run;
ods graphics off;
```

The ALTREF= option specifies the alternative reference, and the actual maximum information is derived in the SEQDESIGN procedure.

In the DESIGN statement, the label `TwoSidedOBrienFleming` identifies the design in the output tables. The default ALT=TWOSIDED option specifies a two-sided alternative hypothesis. The default STOP=REJECT option specifies early stopping in the interim stages only for the purpose of rejecting the null hypothesis. That is, at each interim stage, the trial will either be stopped to reject the null hypothesis or continue to the next stage.

The NSTAGES=4 option in the DESIGN statement specifies the total number of stages in the group sequential trial, including three interim stages and a final stage. In the SEQDESIGN procedure, the null hypothesis for the design is $H_0 : \theta = 0$. The default ALPHA=0.05 option specifies a Type I error probability $\alpha = 0.05$, and the default BETA=0.10 option specifies a Type II error probability $\beta = 0.10$, which corresponds to a power of $1-\beta = 0.90$ at the alternative reference $H_1 : \theta = -10$.

For a two-sided design with early stopping to reject the null hypothesis, there are two boundaries for the design: an upper α (rejection) boundary that consists of upper rejection critical values and a lower α boundary that consists of lower rejection critical values. Each boundary is a set of critical values, one from each stage. With the METHOD=OBF option in the DESIGN statement, the O'Brien-Fleming method is used for the two boundaries for the design; see Figure 78.5.

The SAMPLESIZE statement with the MODEL=TWOSAMPLEMEAN option uses the derived maximum information to compute required sample sizes for a two-sample test for mean difference.

The ODS OUTPUT statement with the BOUNDARY=BND_LDL option creates an output data set named BND_LDL which contains the resulting boundary information. At each stage of the trial, data are collected and analyzed with a statistical procedure, and a test statistic and its corresponding information level are computed.

In this example, you can use the REG procedure to compute the maximum likelihood estimate $\hat{\theta}$ for the drug effect and the corresponding standard error for $\hat{\theta}$. At stage 1, you can use the SEQTEST procedure to compare the test statistic with adjusted boundaries that are derived from the boundary information stored in the BOUND_LDL data set. At each subsequent stage, you can use the SEQTEST procedure to compare the test statistic with adjusted boundaries that are derived from the boundary information stored in the test information table that was created by the SEQTEST procedure at the previous stage. The test information tables are structured for input to the SEQTEST procedure.

At each interim stage, the trial will either be stopped to reject the null hypothesis or continue to the next stage. At the final stage, the null hypothesis is either rejected or accepted.

By default, the SEQDESIGN procedure derives boundary values with equally spaced information levels for all stages—that is, the same information increment between successive stages.

The "Design Information" table in Figure 78.3 displays design specifications and three derived statistics: the maximum information, the average sample number under the null hypothesis (Null Ref ASN), and the average sample number under the alternative hypothesis (Alt Ref ASN). Each statistic is expressed as a percentage of the identical statistic for the corresponding fixed-sample information. The average sample number is the expected sample size (for nonsurvival data) or expected number of events (for survival data). When you specify an alternative reference (in this case, ALTREF=−10), the actual maximum information 0.1074 is also computed. Note that for a symmetric two-sided design, the ALTREF=−10 option implies a lower alternative reference of −10 and an upper alternative reference of 10.

Figure 78.3 O'Brien-Fleming Design Information

The SEQDESIGN Procedure
Design: TwoSidedOBrienFleming

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Reject Null
Method	O'Brien-Fleming
Boundary Key	Both
Alternative Reference	-10
Number of Stages	4
Alpha	0.05
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	102.2163
Max Information	0.107403
Null Ref ASN (Percent of Fixed Sample)	101.5728
Alt Ref ASN (Percent of Fixed Sample)	76.7397

The "Boundary Information" table in Figure 78.4 displays the information level, the lower and upper alternative references, and the lower and upper boundary values at each stage. By default, the SEQDESIGN procedure uses equally spaced information levels for all stages.

Figure 78.4 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

Stage	Information Level Proportion	Information Level Actual	Information Level N	Alternative Reference Lower	Alternative Reference Upper
1	0.2500	0.026851	42.96116	-1.63862	1.63862
2	0.5000	0.053701	85.92233	-2.31736	2.31736
3	0.7500	0.080552	128.8835	-2.83817	2.83817
4	1.0000	0.107403	171.8447	-3.27724	3.27724

Boundary Information (Standardized Z Scale)
Null Reference = 0

Stage	Boundary Values Lower Alpha	Boundary Values Upper Alpha
1	-4.04859	4.04859
2	-2.86278	2.86278
3	-2.33745	2.33745
4	-2.02429	2.02429

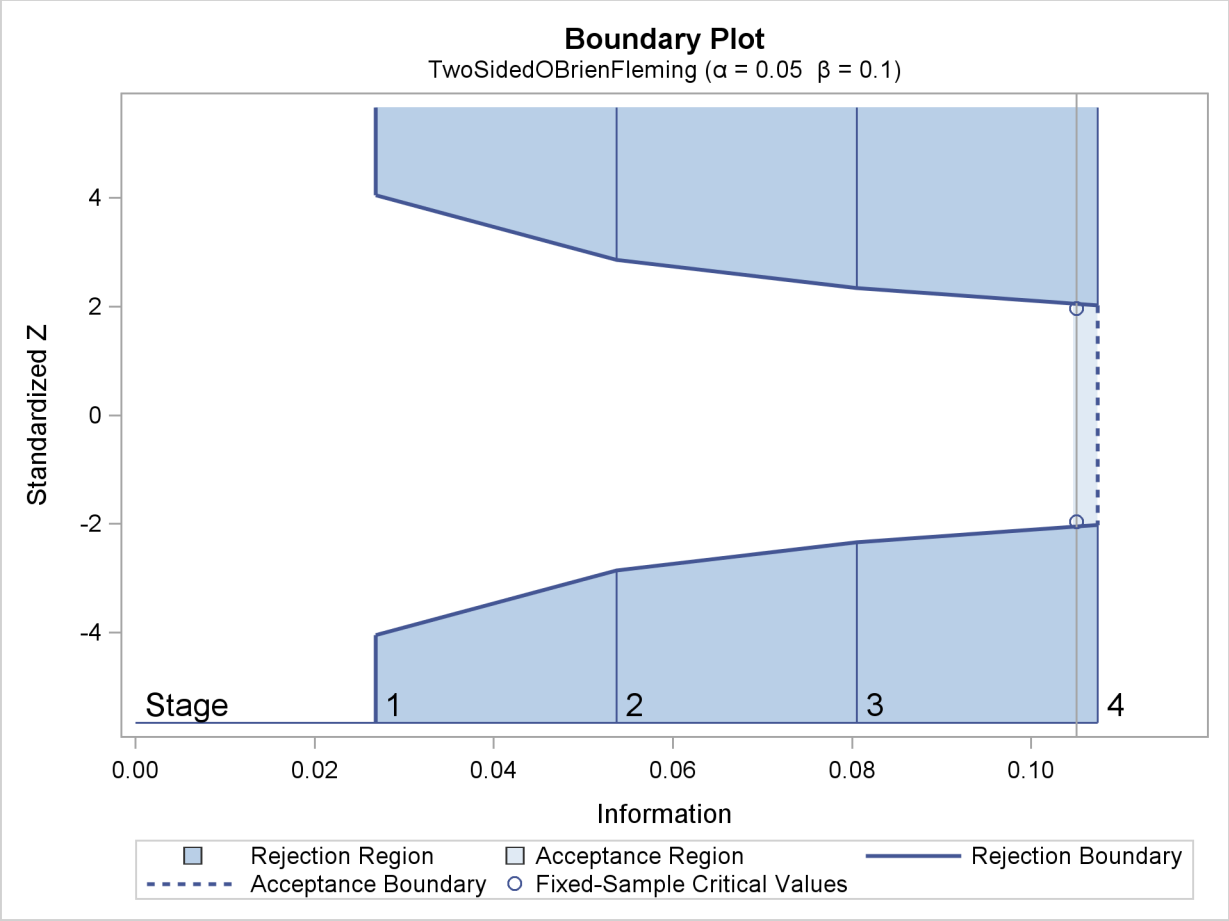
The information proportion is the proportion of maximum information available at each stage. With the BOUNDARYSCALE=STDZ option, the alternative references and boundary values are displayed with the standardized Z statistic scale. The alternative reference in the standardized Z scale at stage k is given by $\theta_1\sqrt{I_k}$, where θ_1 is the alternative reference and I_k is the information available at stage k, $k = 1, 2, 3, 4$.

In this example, a standardized Z statistic is computed by standardizing the parameter estimate of the effect in LDL level. A lower Z test statistic indicates a beneficial effect. Consequently, at each interim stage, if the standardized Z test statistic is less than or equal to the corresponding lower α boundary value, the hypothesis $H_0 : \theta = 0$ is rejected for efficacy. If the test statistic is greater than or equal to the corresponding upper α boundary value, the hypothesis H_0 is rejected for harmful effect. Otherwise, the process continues to the next stage. At the final stage (stage 4), the hypothesis H_0 is rejected for efficacy if the Z statistic is less than or equal to the corresponding lower α boundary value -2.0243, and the hypothesis H_0 is rejected for harmful effect if the Z statistic is greater than or equal to the corresponding upper α boundary value 2.0243. Otherwise, the hypothesis of no significant difference is accepted.

Note that in a typical trial, the actual information levels do not match the information levels specified in the design. Consequently, the SEQTEST procedure modifies the boundary values stored in the BND_LDL data set to adjust for these new information levels.

If the ODS GRAPHICS ON statement is specified, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Figure 78.5.

Figure 78.5 O'Brien-Fleming Boundary Plot



This boundary plot displays the boundary values in the "Boundary Information" table in Figure 78.4. The stages are indicated by vertical lines with accompanying stage numbers. The horizontal axis indicates the information levels for the stages. If a test statistic at an interim stage is in the rejection region (shaded area), the trial stops and the null hypothesis is rejected. If the statistic is not in any rejection region, the trial continues to the next stage. The plot also displays critical values for the corresponding fixed-sample design. The symbol "○" identifies the fixed-sample critical values of -1.96 and 1.96.

When you specify the SAMPLESIZE statement, the maximum information (either explicitly specified or derived in the SEQDESIGN procedure) is used to compute the required sample sizes for the study. The MODEL=TWOSAMPLEMEAN(STDDEV=20) option specifies the test for the difference between two normal means. See the section "Test for the Difference between Two Normal Means" in the chapter "The SEQDESIGN Procedure" for a detailed description of how these required sample sizes are calculated.

The "Sample Size Summary" table in Figure 78.6 displays the parameters for the sample size computation and the resulting maximum and expected sample sizes.

Figure 78.6 Required Sample Size Summary

Sample Size Summary

Test	Two-Sample Means
Mean Difference	-10
Standard Deviation	20
Max Sample Size	171.8447
Expected Sample Size (Null Ref)	170.7627
Expected Sample Size (Alt Ref)	129.0137

With the derived maximum information 0.1074 and the specified MODEL=TWOSAMPLEMEAN (STDDEV=20) option in the SAMPLESIZE statement, the total sample size in each group is

$$N_a = N_b = 2\sigma^2 I_X = 2 \times 20^2 \times 0.1074 = 85.92$$

The "Sample Sizes (N)" table in Figure 78.7 displays the required sample sizes at each stage for the trial, in both fractional and integer numbers. The derived fractional sample sizes are displayed under the heading "Fractional N." These sample sizes are rounded up to integers under the heading "Ceiling N." With the default WEIGHT=1 option in the SAMPLESIZE statement, the sample sizes for the two groups are equal for the two-sample test.

Figure 78.7 Required Sample Sizes

Sample Sizes (N)
Two-Sample Z Test for Mean Difference

	Fractional N			
Stage	N	N(Grp 1)	N(Grp 2)	Information
1	42.96	21.48	21.48	0.0269
2	85.92	42.96	42.96	0.0537
3	128.88	64.44	64.44	0.0806
4	171.84	85.92	85.92	0.1074

Sample Sizes (N)
Two-Sample Z Test for Mean Difference

	Ceiling N			
Stage	N	N(Grp 1)	N(Grp 2)	Information
1	44	22	22	0.0275
2	86	43	43	0.0538
3	130	65	65	0.0812
4	172	86	86	0.1075

In practice, integer sample sizes are used in the trial, and the resulting information levels increase slightly. Thus, each of the two groups needs 22, 43, 65, and 86 patients for the four stages, respectively.

Suppose that 22 patients are available in each group at stage 1 and that their measurements for LDL are saved in the data set LDL_1. Figure 78.8 lists the first 10 observations in the data set LDL_1.

Figure 78.8 Clinical Trial Data

First 10 Obs in the Trial Data

Obs	Trt	Ldl
1	0	33.33
2	1	-14.89
3	0	15.30
4	1	4.71
5	0	26.89
6	1	-48.74
7	0	-39.35
8	1	-8.13
9	0	-8.22
10	1	12.35

The variable Trt is an indicator variable with value 1 for patients in the treatment group and value 0 for patients in the placebo control group. The variable Ldl is the LDL level of these patients.

The following statements use the REG procedure to estimate the mean treatment difference and its associated standard error at stage 1:

```
proc reg data=LDL_1;
   model Ldl=Trt;
ods output ParameterEstimates=Parms_LDL1;
run;
```

The following statements create the data set for the mean treatment difference and its associated standard error as a PARMS= data set, which will subsequently serve as an input data set for PROC SEQTEST. Note that all of the variables are required for a PARMS= data set, as described in the section "PARMS < (TESTVAR= *variable*) > = SAS Data Set" on page 6014.

```
data Parms_LDL1;
   set Parms_LDL1;
   if Variable='Trt';
   _Scale_='MLE';
   _Stage_= 1;
   keep _Scale_ _Stage_ Variable Estimate StdErr;
run;

proc print data=Parms_LDL1;
   title 'Statistics Computed at Stage 1';
run;
```

Figure 78.9 displays the statistics computed at stage 1.

Figure 78.9 Statistics Computed at Stage 1

Statistics Computed at Stage 1

Obs	Variable	Estimate	StdErr	_Scale_	_Stage_
1	Trt	-2.52591	5.68572	MLE	1

Since the sample sizes derived are based on the estimated variance at the designing phase, the information level that corresponds to the test statistic at stage 1 is estimated by

$$I_1 = \frac{1}{s_1^2} = \frac{1}{5.686^2} = 0.0309$$

where s_1 is the standard error of the treatment estimate.

The following statements invoke the SEQTEST procedure to test for early stopping at stage 1:

```
ods graphics on;
proc seqtest Boundary=Bnd_LDL
             Parms(Testvar=Trt)=Parms_LDL1
             ;
ods output Test=Test_LDL1;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 1, which was generated in the SEQDESIGN procedure. The PARMS=PARMS_LDL1 option specifies the input data set PARMS_LDL1 that contains the test statistic and its associated standard error at stage 1, and the TESTVAR=TRT option identifies the test variable TRT in the data set.

The ODS OUTPUT statement with the TEST=TEST_LDL1 option creates an output data set named TEST_LDL1 which contains the updated boundary information for the test at stage 1. The data set also provides the boundary information that is needed for the group sequential test at the next stage. See the section "Boundary Adjustments for Information Levels" on page 6017 for details.

The "Design Information" table in Figure 78.10 displays design specifications. By default, the boundary values are modified for the updated information levels to maintain the Type I α level. The maximum information remains the same as in the original design, but the derived Type II error probability β and power $1 - \beta$ are slightly different with new information levels. With the updated power $1 - \beta$, the corresponding fixed-sample design is also updated.

Figure 78.10 Design Information

The SEQTEST Procedure

Design Information

BOUNDARY Data Set	WORK.BND_LDL
Data Set	WORK.PARMS_LDL1
Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Reject Null
Number of Stages	4
Alpha	0.05
Beta	0.10074
Power	0.89926
Max Information (Percent of Fixed Sample)	102.4815
Max Information	0.10740291
Null Ref ASN (Percent of Fixed Sample)	101.7765
Alt Ref ASN (Percent of Fixed Sample)	75.4928

The "Test Information" table in Figure 78.11 displays the boundary values for the test statistic with the default standardized Z scale. Since only the information level at stage 1 is available from the PARMS= data set, information levels at future interim stages are derived proportionally from the corresponding information levels in the original design.

Figure 78.11 Sequential Tests

Test Information (Standardized Z Scale)
Null Reference = 0

	Information Level		Alternative Reference		Boundary Values	
Stage	Proportion	Actual	Lower	Upper	Lower Alpha	Upper Alpha
1	0.2880	0.030934	-1.75879	1.75879	-3.39532	3.39532
2	0.5253	0.056423	-2.37536	2.37536	-2.77374	2.77374
3	0.7627	0.081913	-2.86205	2.86205	-2.32412	2.32412
4	1.0000	0.107403	-3.27724	3.27724	-2.03147	2.03147

Test Information (Standardized Z Scale)
Null Reference = 0

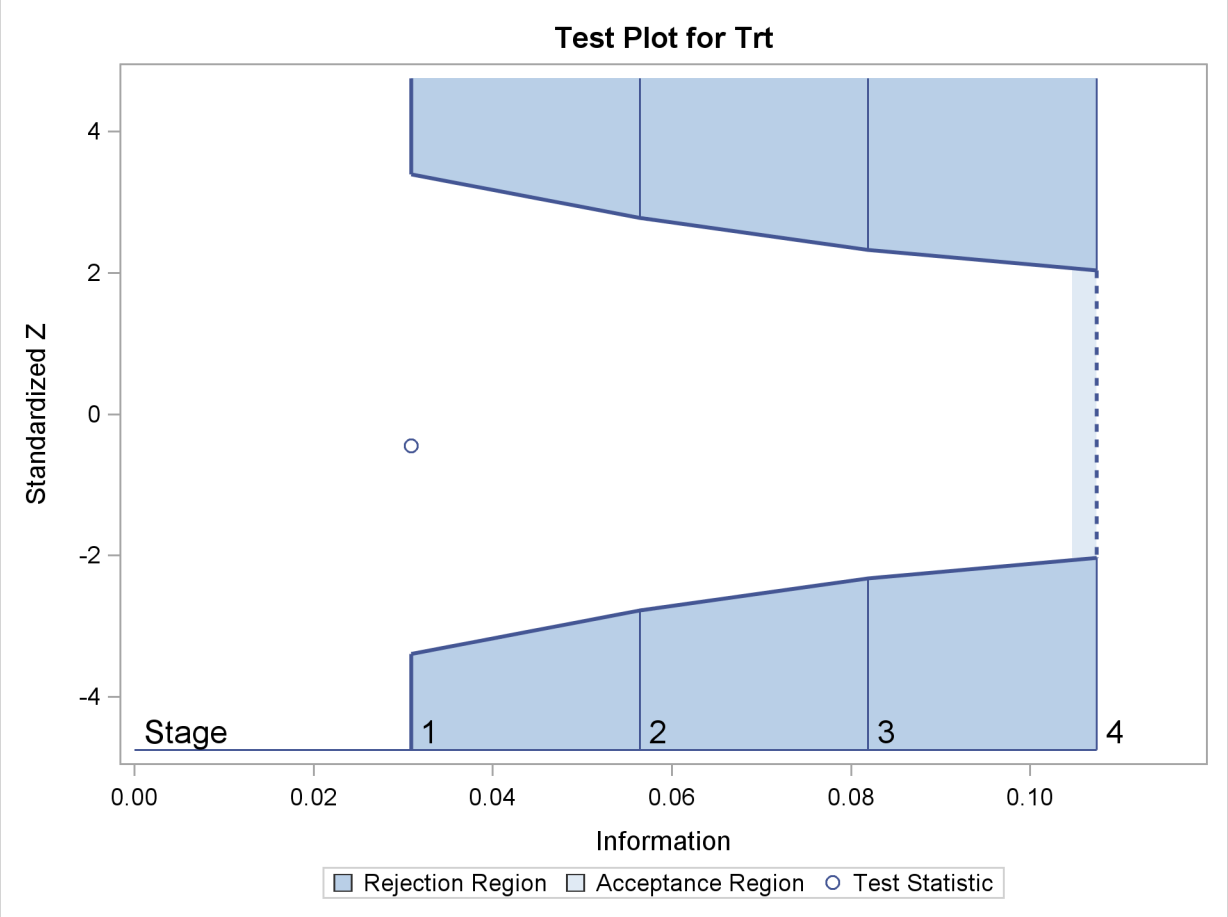
	Test Trt	
Stage	Estimate	Action
1	-0.44426	Continue
2	.	
3	.	
4	.	

At stage 1, the standardized Z statistic -0.44426 is between the lower and upper α boundary values, and so the trial continues to the next stage. With the observed information level at stage 1, $I_1 = 0.0309$ (which is not substantially different from the target information level at stage 1), the trial continues to the next stage without adjustment of the sample size according to the study plan.

Note that if an observed information level is different from its target level at an interim stage, the sample sizes at future stages might be adjusted to achieve the target maximum information level according to the study plan. For example, a study plan might modify the final sample size to achieve the target maximum information level if the observed information level is different from its target level by a specified amount at the interim stage. If the variance estimate is used to compute the required sample size for the group sequential trial, the study plan might use the current variance estimate to update the required sample size for the trial (Jennison and Turnbull 2000, p. 295).

If the ODS GRAPHICS ON statement is specified, a detailed test plot with the rejection and acceptance regions is displayed by default, as shown in Figure 78.12. This plot displays the boundary values in the "Test Information" table in Figure 78.11. The stages are indicated by vertical lines with accompanying stage numbers. The horizontal axis indicates the information levels for the stages. As expected, the test statistic is in the continuation region between the lower and upper α boundaries.

Figure 78.12 Sequential Test Plot



The following statements use the REG procedure with the data available at the first two stages to estimate the mean treatment difference and its associated standard error at stage 2:

```
proc reg data=LDL_2;
   model Ldl=Trt;
ods output ParameterEstimates=Parms_LDL2;
run;
```

The following statements create and display (in Figure 78.13) the data set for the mean treatment difference and its associated standard error:

```
data Parms_LDL2;
   set Parms_LDL2;
   if Variable='Trt';
   _Scale_='MLE';
   _Stage_= 2;
   keep _Scale_ _Stage_ Variable Estimate StdErr;
run;

proc print data=Parms_LDL2;
   title 'Statistics Computed at Stage 2';
run;
```

Figure 78.13 Statistics Computed at Stage 2

Statistics Computed at Stage 2

Obs	Variable	Estimate	StdErr	_Scale_	_Stage_
1	Trt	-8.37628	4.24405	MLE	2

Using the standard error for the treatment estimate available at stage 2, the information level that corresponds to the test statistic at stage 2 is estimated by

$$I_2 = \frac{1}{s_2^2} = \frac{1}{4.244^2} = 0.0555$$

where s_2 is the standard error of the treatment estimate at stage 2.

The following statements invoke the SEQTEST procedure to test for early stopping at stage 2:

```
proc seqtest Boundary=Test_LDL1
             Parms(Testvar=Trt)=Parms_LDL2
             ;
ods output Test=Test_LDL2;
run;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 2, which was generated by the SEQTEST procedure at the previous stage. The PARMS= option specifies the input data set that contains the test statistic and its associated standard error at stage 2, and the TESTVAR= option identifies the test variable in the data set.

The ODS OUTPUT statement with the TEST=TEST_LDL2 option creates an output data set named TEST_LDL2 which contains the updated boundary information for the test at stage 2. The data set also provides the boundary information that is needed for the group sequential test at the next stage.

The "Test Information" table in Figure 78.14 displays the boundary values for the test statistic with the default standardized Z scale.

Figure 78.14 Sequential Tests

The SEQTEST Procedure

Test Information (Standardized Z Scale)
Null Reference = 0

Stage	--Information Level- Proportion	Actual	------Alternative----- -------Reference------ Lower	Upper	----Boundary Values--- ---Lower-- Alpha	---Upper-- Alpha
1	0.2880	0.030934	-1.75879	1.75879	-3.39532	3.39532
2	0.5169	0.055519	-2.35624	2.35624	-2.78456	2.78456
3	0.7585	0.081461	-2.85413	2.85413	-2.32908	2.32908
4	1.0000	0.107403	-3.27724	3.27724	-2.03097	2.03097

Test Information (Standardized Z Scale)
Null Reference = 0

Stage	----------Test---------- ----------Trt---------- Estimate	Action
1	-0.44426	Continue
2	-1.97365	Continue
3	.	
4	.	

At stage 2, the standardized test statistic, $z = -8.37628/4.24405 = -1.97365$, is between its corresponding lower and upper α boundary values. Therefore, the trial continues to the next stage.

The following statements use the REG procedure with the data available at the first three stages to estimate the mean treatment difference and its associated standard error at stage 3:

```
proc reg data=LDL_3;
   model Ldl=Trt;
ods output ParameterEstimates=Parms_LDL3;
run;
```

The following statements create and display (in Figure 78.15) the data set for the mean treatment difference and its associated standard error:

```
data Parms_LDL3;
   set Parms_LDL3;
   if Variable='Trt';
   _Scale_='MLE';
   _Stage_= 3;
   keep _Scale_ _Stage_ Variable Estimate StdErr;
run;

proc print data=Parms_LDL3;
   title 'Statistics Computed at Stage 3';
run;
```

Figure 78.15 Statistics Computed at Stage 3

Statistics Computed at Stage 3

Obs	Variable	Estimate	StdErr	_Scale_	_Stage_
1	Trt	-9.21369	3.42149	MLE	3

The following statements invoke the SEQTEST procedure to test for early stopping at stage 3:

```
ods graphics on;
proc seqtest Boundary=Test_LDL2
             Parms(Testvar=Trt)=Parms_LDL3
             ;
ods output Test=Test_LDL3;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 2, which was generated by the SEQTEST procedure at the previous stage. The PARMS= option specifies the input data set that contains the test statistic and its associated standard error at stage 3, and the TESTVAR= option identifies the test variable in the data set.

The ODS OUTPUT statement with the TEST=TEST_LDL3 option creates an output data set named TEST_LDL3 which contains the updated boundary information for the test at stage 3. The data set also provides the boundary information that is needed for the group sequential test at the next stage.

The "Test Information" table in Figure 78.16 displays the boundary values for the test statistic with the default standardized Z scale.

Figure 78.16 Sequential Tests

The SEQTEST Procedure

Test Information (Standardized Z Scale)
Null Reference = 0

Stage	--Information Level- Proportion	Actual	------Alternative----- -------Reference------ Lower	Upper	----Boundary Values--- ---Lower-- Alpha	---Upper-- Alpha
1	0.2880	0.030934	-1.75879	1.75879	-3.39532	3.39532
2	0.5169	0.055519	-2.35624	2.35624	-2.78456	2.78456
3	0.7953	0.085422	-2.92271	2.92271	-2.25480	2.25480
4	1.0000	0.107403	-3.27724	3.27724	-2.04573	2.04573

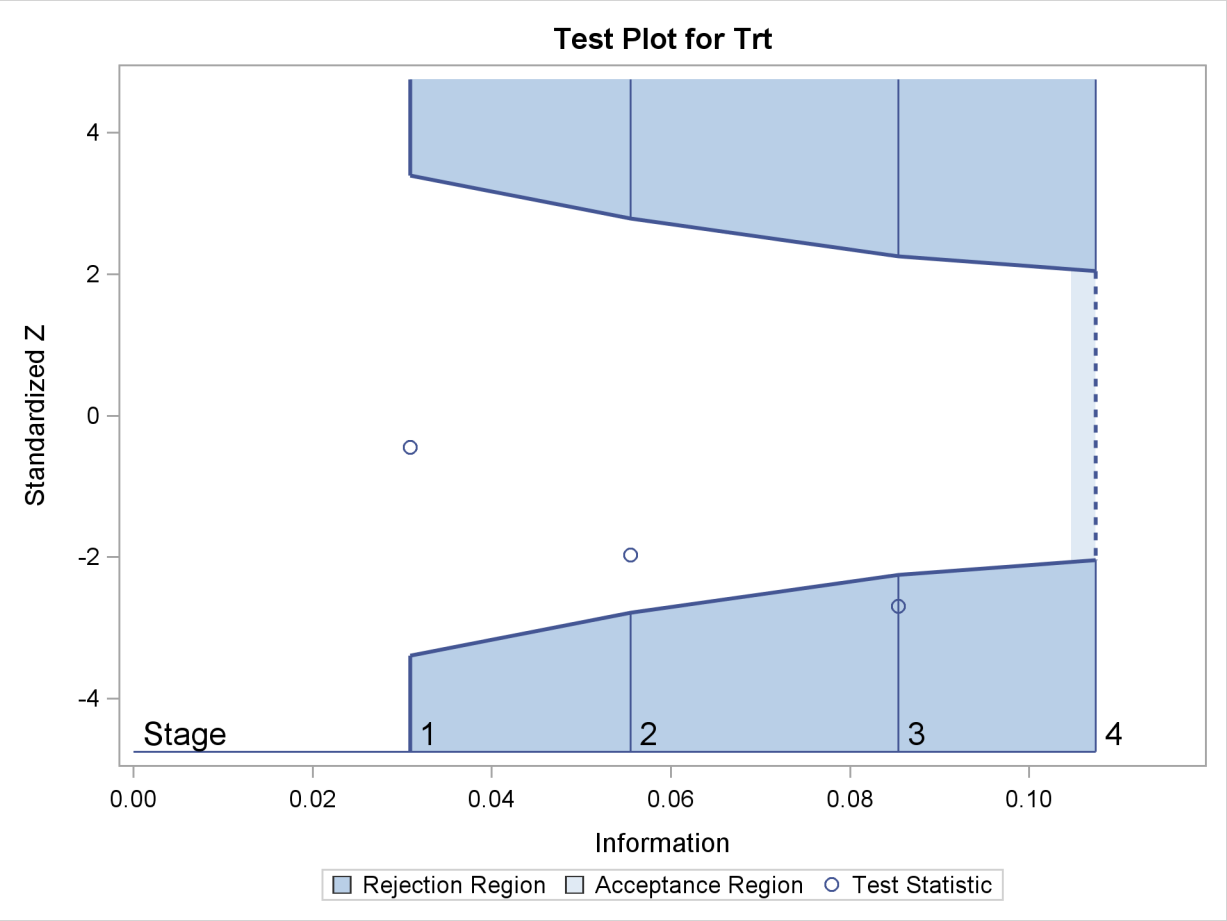
Test Information (Standardized Z Scale)
Null Reference = 0

Stage	----------Test---------- ----------Trt---------- Estimate	Action
1	-0.44426	Continue
2	-1.97365	Continue
3	-2.69289	Reject Null
4	.	

The sequential test stops at stage 3 to reject the null hypothesis for the lower alternative because the test statistic -2.69289 is less than the corresponding upper α boundary -2.25480. That is, the test demonstrates significant beneficial effect for the new drug.

The "Test Plot" displays boundary values for the design and the test statistic at the first three stages, as shown in Figure 78.17. It shows that the test statistic is in the "Rejection Region" below the lower α boundary at stage 3.

Figure 78.17 Sequential Test Plot



When a trial stops, the "Parameter Estimates" table in Figure 78.18 displays the stopping stage, parameter estimate, unbiased median estimate, confidence limits, and p-value under the null hypothesis $H_0 : \theta = 0$. As expected, the p-value 0.0108 is significant at the two-sided α level, $\alpha = 0.05$, and the confidence interval does not contain the value zero. The p-value, unbiased median estimate, and confidence limits depend on the ordering of the sample space (k, z), where k is the stage number and z is the standardized Z statistic. See the section "Analysis after a Sequential Test" on page 6024 for a detailed description of these statistics.

Figure 78.18 Parameter Estimates

Parameter Estimates
Stagewise Ordering

Parameter	Stopping Stage	MLE	p-Value for H0:Parm=0	Median Estimate
Trt	3	-9.213692	0.0108	-9.022891

Parameter Estimates
Stagewise Ordering

Parameter	95% Confidence Limits	
Trt	-15.79845	-2.13138

Syntax: SEQTEST Procedure

The following PROC SEQTEST statement is required for the SEQTEST procedure:

PROC SEQTEST < *options* > ;

PROC SEQTEST Statement

Table 78.1 summarizes the options in the PROC SEQTEST statement.

Table 78.1 Summary of PROC SEQTEST Options

Option	Description
Input Data Sets	
BOUNDARY=	specifies data set for boundary information
DATA=	specifies data set for parameter estimates and information levels
PARMS=	specifies data set for parameter estimates and standard errors
Boundaries	
BETAOVERLAP=	checks for overlapping of the lower and upper β boundaries at the current and subsequent interim stages in a two-sided design
BOUNDARYADJ=	specifies boundary adjustment for information levels
BOUNDARYKEY=	specifies boundary key to maintain Type I and II error probability levels
BOUNDARYSCALE=	specifies boundary scale
ERRSPENDMIN=	specifies minimum error spending for the boundaries
NSTAGES=	specifies number of stages

Table 78.1 *continued*

Option	Description
Test Statistics	
DATA(TESTVAR=)	specifies test variable in DATA= data set
PARMS(TESTVAR=)	specifies test variable in PARMS= data set
***p*-Values and Confidence Intervals**	
CIALPHA=	specifies significance levels for the confidence interval
CITYPE=	specifies types of confidence interval
ORDER=	specifies the ordering of the sample space used to derive the p-values and confidence limits
Table Output	
CONDPOWER	displays conditional powers
ERRSPEND	displays cumulative error spending at each stage
PREDPOWER	displays predictive powers
PSS	displays powers and expected sample sizes
RCI	displays repeated confidence intervals
STOPPROB	displays expected cumulative stopping probabilities
Graphics Output	
PLOTS=ASN	displays expected sample numbers plot
PLOTS=CONDPOWER	displays conditional powers plot
PLOTS=ERRSPEND	displays error spending plot
PLOTS=POWER	displays powers plot
PLOTS=RCI	displays repeated confidence intervals plot
PLOTS=TEST	displays boundary plot with test statistics

The BOUNDARY= option provides the information for the design and is required in the PROC SEQTEST statement. By default, the SEQTEST procedure displays tables of design information and test information. If the ODS GRAPHICS ON statement is specified, the procedure also displays a sequential test plot.

The following options can be used in the PROC SEQTEST statement. They are listed in alphabetical order.

BETAOVERLAP=ADJUST | NOADJUST

OVERLAP=ADJUST | NOADJUST

specifies whether to check for overlapping of the lower and upper β boundaries for the two corresponding one-sided tests at the current and subsequent interim stages. This option applies to two-sided designs with early stopping to accept H_0, or to either accept or reject H_0. This type of overlapping might result from a small β spending at an interim stage. When you specify BETAOVERLAP=ADJUST, the procedure checks for this type of overlapping at the current and subsequent interim stages. If such overlapping is found, the β boundaries for the two-sided design at that stage are set to missing, and the β spending values at subsequent stages are adjusted, as described in the section "Boundary Adjustments for Overlapping Lower and Upper β Boundaries" on page 6020.

You can specify BETAOVERLAP=NOADJUST to request that no adjustment be made. The default is BETAOVERLAP=ADJUST.

BOUNDARY=*SAS-data-set*

names the required SAS data set that contains the design boundary information. At stage 1, the data set is usually created from the "Boundary Information" table created by the SEQDESIGN procedure. At each subsequent stage, the data set is usually created from the "Test Information" table created by the SEQTEST procedure at the previous stage. The data set includes the variables _Scale_ for the boundary scale, _Stop_ for the stopping criterion, and _ALT_ for the type of alternative hypothesis. It also includes _Stage_ for the stage number, Info_Prop for the information proportion, and a set of the boundary variables from Bound_LA, Bound_LB, Bound_UB, and Bound_UA for boundary values at each stage.

The data set might also include _Info_ for the actual information level, NObs for the number of observation, and Events for the number of events required at each stage.

BOUNDARYADJ=*method*

BOUNDARYADJ(*boundary***)=***method*

ERRSPENDADJ=*method*

ERRSPENDADJ(*boundary***)=***method*

specifies methods for boundary adjustments if the observed information level at a stage does not match the information level in the design specified with the BOUNDARY= data set. The default is BOUNDARYADJ=ERRLINE.

The following options specify available methods for the boundary adjustment:

NONE

specifies that the cumulative error spending at each interim stage not be changed, even if the corresponding information level has been changed.

ERRLINE

specifies the linear interpolation method for the adjustment.

ERRFUNCGAMMA < (GAMMA= γ **) >**

specifies the gamma function method for the adjustment. The GAMMA= suboption specifies the γ parameter in the function, where $\gamma \leq 3$. The default is GAMMA=-2.

ERRFUNCOBF

specifies the approximate O'Brien-Fleming cumulative error spending function for the adjustment.

ERRFUNCPOC

specifies the approximate Pocock cumulative error spending function for the adjustment.

ERRFUNCPOW < (RHO= ρ **) >**

specifies the power function method for the adjustment. The RHO= suboption specifies the power parameter ρ in the function, where $\rho \geq 0.25$. The default is RHO=2.

See the section "Boundary Adjustments for Information Levels" on page 6017 for a detailed description of the available boundary adjustment methods in the SEQTEST procedure.

If a boundary adjustment method is used for all boundaries in the design, you can use the BOUNDARYADJ=*method* option to specify the method. Otherwise, you can use the following BOUNDARYADJ(*boundary*)=*method* options to specify different methods for the boundaries.

BOUNDARYADJ(ALPHA)=*method*

BOUNDARYADJ(REJECT)=*method*

specifies the adjustment method for the α (rejection) boundary of a one-sided design or the lower and upper α boundaries of a two-sided design.

BOUNDARYADJ(LOWERALPHA)=*method*

BOUNDARYADJ(LOWERREJECT)=*method*

specifies the adjustment method for the lower α boundary of a two-sided design.

BOUNDARYADJ(UPPERALPHA)=*method*

BOUNDARYADJ(UPPERREJECT)=*method*

specifies the adjustment method for the upper α boundary of a two-sided design.

BOUNDARYADJ(BETA)=*method*

BOUNDARYADJ(ACCEPT)=*method*

specifies the adjustment method for the β (acceptance) boundary of a one-sided design or the lower and upper β boundaries of a two-sided design.

BOUNDARYADJ(LOWERBETA)=*method*

BOUNDARYADJ(LOWERACCEPT)=*method*

specifies the adjustment method for the lower β boundary of a two-sided design.

BOUNDARYADJ(UPPERBETA)=*method*

BOUNDARYADJ(UPPERACCEPT)=*method*

specifies the adjustment method for the upper β boundary of a two-sided design.

BOUNDARYKEY=ALPHA | BETA | BOTH

specifies the boundary key to be maintained in the boundary adjustments. The BOUNDARYKEY=ALPHA option maintains the Type I α level and derives the Type II error probability, and the BOUNDARYKEY=BETA option maintains the Type II β level and derives the Type I error probability. The BOUNDARYKEY=BOTH option maintains both α and β levels simultaneously by deriving a new maximum information. The default is BOUNDARYKEY=ALPHA.

BOUNDARYSCALE=MLE | SCORE | STDZ | PVALUE

BSCALE=MLE | SCORE | STDZ | PVALUE

specifies the boundary scale to be displayed in the output boundary table and plot. The BOUNDARYSCALE=MLE, BOUNDARYSCALE=SCORE, BOUNDARYSCALE=STDZ, and BOUNDARYSCALE=PVALUE options correspond to the boundary with the maximum likelihood estimator scale, score statistic scale, standardized normal Z scale, and p-value scale, respectively. The default is BOUNDARYSCALE=STDZ.

With the BOUNDARYSCALE=MLE or BOUNDARYSCALE=SCORE option, either the MAXINFO= option must be specified or the _Info_ variable must be in the BOUNDARY= data set to provide the necessary information level at each stage to derive the boundary values. Usually, these values are obtained from analysis output in SAS procedures.

Note that for a two-sided design, the p-value scale displays the one-sided fixed-sample p-value under the null hypothesis with a lower alternative hypothesis.

CIALPHA=α **< (< LOWER=**α_l **> < UPPER=**α_u **>) >**

specifies the significance levels for the confidence interval, where $0 < \alpha < 1$, $0 < \alpha_l < 0.5$, and $0 < \alpha_u < 0.5$. The default is CIALPHA= 0.05.

For a lower confidence interval (CITYPE=LOWER), the CIALPHA=α option produces a $(1 - \alpha)$ lower confidence interval. For an upper confidence interval (CITYPE=UPPER), the CIALPHA=α option produces a $(1 - \alpha)$ upper confidence interval. The LOWER= and UPPER= suboptions are applicable only for a two-sided confidence interval (CITYPE=TWOSIDED). The LOWER= suboption specifies the lower significance level α_l and the upper significance level $\alpha_u = 1 - \alpha_l$. The UPPER= suboption specifies the upper significance level α_u and the lower significance level $\alpha_l = 1 - \alpha_u$. If both LOWER= and UPPER= suboptions are not specified, $\alpha_l = \alpha_u = \alpha/2$. The significance levels α_l and α_u are then used for the $(1 - \alpha_l)$ lower confidence limit and $(1 - \alpha_u)$ upper confidence limit, respectively.

CITYPE=LOWER | UPPER | TWOSIDED

specifies the type of confidence interval. The CITYPE=LOWER, CITYPE=UPPER, and CITYPE=TWOSIDED options correspond to the lower confidence interval, upper confidence interval, and two-sided confidence interval, respectively. The default is CITYPE=LOWER for the design with an upper alternative, CITYPE=UPPER for the design with a lower alternative, and CITYPE=TWOSIDED for the design with a two-sided alternative.

DATA < (TESTVAR=*variable***) >=***SAS-data-set*

names the SAS data set that contains the test statistic and its associated information level for the stage. The data set includes the stage variable _Stage_ and a variable to identify or derive the information level: _Info_ for the information level, NObs for the number of observation, or Events for the number of events. If the information level that corresponds to the test statistic is not available, the information level derived in the BOUNDARY= data set is used.

If the TESTVAR= option is specified, the data set also includes the test variable specified in the TESTVAR= option and the scale variable _Scale_ for the test statistic. Usually, these test variable values are obtained from analysis output in SAS procedures.

ERRSPENDMIN=*numbers*

ERRSPENDMIN(*boundary***)=***numbers*

specifies the minimum error spending values at the current observed and future interim stages for the boundaries specified in the BOUNDARYKEY= option. The default is ERRSPENDMIN=0.

If a set of numbers is used for each boundary in the design, you can use the ERRSPENDMIN=*numbers* option. Otherwise, you can use the following ERRSPENDMIN(*boundary*)=*numbers* options to specify different sets of minimum error spending values

for the boundaries. For a boundary, the error spending value at stage 1 is identical to its nominal p-value.

ERRSPENDMIN(ALPHA)=*numbers*

ERRSPENDMIN(REJECT)=*numbers*

specifies the minimum error spending values for the α boundary of a one-sided design or the lower and upper α boundaries of a two-sided design.

ERRSPENDMIN(LOWERALPHA)=*numbers*

ERRSPENDMIN(LOWERREJECT)=*numbers*

specifies the minimum error spending values for the lower α boundary of a two-sided design.

ERRSPENDMIN(UPPERALPHA)=*numbers*

ERRSPENDMIN(UPPERREJECT)=*numbers*

specifies the minimum error spending values for the upper α boundary of a two-sided design.

ERRSPENDMIN(BETA)=*numbers*

ERRSPENDMIN(ACCEPT)=*numbers*

specifies the minimum error spending values for the β boundary of a one-sided design or the lower and upper β boundaries of a two-sided design.

ERRSPENDMIN(LOWERBETA)=*numbers*

ERRSPENDMIN(LOWERACCEPT)=*numbers*

specifies the minimum error spending values for the lower β boundary of a two-sided design.

ERRSPENDMIN(UPPERBETA)=*numbers*

ERRSPENDMIN(UPPERACCEPT)=*numbers*

specifies the minimum error spending values for the upper β boundary of a two-sided design.

NSTAGES=*number*

specifies the number of stages for the clinical trial. The default is the number derived from the BOUNDARY= data set.

The specified NSTAGES= number might or might not be the same as the number derived in the BOUNDARY= data set. You can use the NSTAGES= option to set the next stage as the final stage to compute the conditional power, as described in the section "Conditional Power Approach" on page 6021.

ORDER=LR | MLE | STAGEWISE

specifies the ordering of the sample space (k, z), where k is the stage number and z is the observed standardized Z statistic. The ordering is used to derive the p-values for the observed (k, z) statistic and to create unbiased median estimate and confidence limits from the statistic. The ORDER=LR option specifies the LR ordering that compares the distances between observed standardized Z statistics and their corresponding hypothetical values, the ORDER=MLE option specifies the MLE ordering that compares values in the MLE scale,

and the ORDER=STAGEWISE specifies the stagewise ordering that uses counterclockwise ordering around the continuation region. The default is ORDER=STAGEWISE. See the section "Available Sample Space Orderings in a Sequential Test" on page 6025 for a detailed description of these sample space orderings.

PARMS < (TESTVAR=*variable***) > =***SAS-data-set*

names the SAS data set that contains the parameter estimate and its associated standard error for the stage. The data set includes the stage variable _Stage_, the test statistic Estimate, the standard error of the estimate StdErr, and the test statistic scale variable _Scale_. The standard error is are used to derive the information level. If the standard error is not available, the information level derived in the BOUNDARY= data set is used.

The data set also includes the variable Parameter, Effect, Variable, or Parm that contains the test variable specified in the TESTVAR= option. Usually, these test variable values are obtained from analysis output in SAS procedures.

Table Output Options

The following options can be used in the PROC SEQTEST statement to display additional table output. They are listed in alphabetical order.

CONDPOWER < (CREF=*numbers* **) >**

displays conditional powers given the most recently observed statistic under specified hypothetical references, where the numbers $c_i \geq 0$. In the SEQTEST procedure, the conditional power is the probability that the test statistic at the final stage would exceed the rejection critical value given the observed statistic.

If interim stages exist between the current stage and the final stage, the conditional power is not the conditional probability to reject the null hypothesis H_0. In this case, you can set the next stage as the final stage, and the conditional power is the conditional probability to reject H_0.

For a one-sided test, the powers are derived under the hypothetical references $\theta = \hat{\theta}$ and $\theta = c_i\theta_1$, where $\hat{\theta}$ is the observed statistic, θ_1 is the alternative reference, and c_i are the values specified in the CREF= option. For a two-sided test, the powers are derived under hypothetical references $\theta = \hat{\theta}$, $\theta = c_i\theta_{1l}$, and $\theta = c_i\theta_{1u}$, where θ_{1l} is the lower alternative reference and θ_{1u} is the upper alternative reference. The default is CREF= 0 0.5 1.0 1.5.

ERRSPEND

displays the error spending at each stage for each sequential boundary.

PREDPOWER

displays predictive powers given the most recently observed statistic. The predictive power is the posterior probability that the test statistic at the final stage would exceed the rejection critical value given the observed statistic and a prior distribution of the hypothetical reference. A noninformative prior is used in the procedure.

PSS < (CREF=*numbers* **) >**

displays powers and expected sample sizes under various hypothetical references, where the numbers $c_i \geq 0$.

For a one-sided design with the null reference $\theta_0 = 0$, the power and expected sample sizes under hypotheses $\theta = c_i\ \theta_1$ are displayed, where θ_1 is the alternative reference and c_i are the values specified in the CREF= option.

For a two-sided design, the power and expected sample sizes under hypotheses $\theta = c_i\theta_{1l}$ and $\theta = c_i\theta_{1u}$ are displayed, where θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively. The default is CREF= 0 0.5 1.0 1.5.

Note that for a symmetric two-sided design, only the power and expected sample sizes under hypotheses $\theta = c_i\ \theta_{1u}$ are derived.

RCI

displays repeated confidence intervals for the parameter from the observed statistic at each stage. Repeated confidence intervals include both rejection and acceptance confidence intervals.

With the STOP=REJECT or STOP=BOTH option, rejection confidence limits can be derived, and the null hypothesis $H_0 : \theta = 0$ is rejected if the lower rejection confidence limit is greater than 0 or the upper rejection confidence limit is less than 0.

With the STOP=ACCEPT or STOP=BOTH option, acceptance confidence limits can be derived, and the null hypothesis is accepted with alternative hypotheses $H_{1l} : \theta = \theta_{1l}$ and $H_{1u} : \theta = \theta_{1u}$ if the upper acceptance confidence limit is less than θ_{1u} and the lower acceptance confidence limit is greater than θ_{1l}.

STOPPROB < (CREF=*numbers* **) >**

displays expected cumulative stopping probabilities under various hypothetical references, where the numbers $c_i \geq 0$.

For a one-sided design, expected cumulative stopping probabilities at each stage under hypotheses $\theta = c_i\ \theta_1$ are displayed, where θ_1 is the alternative reference and c_i are the values specified in the CREF= option.

For a two-sided design, expected cumulative stopping probabilities at each stage under hypotheses $\theta = c_i\theta_{1l}$ and $\theta = c_i\theta_{1u}$ are displayed, where θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively. Note that for a symmetric two-sided design, only expected cumulative stopping probabilities under hypotheses $\theta = c_i\ \theta_{1u}$ are derived. The default is CREF= 0 0.5 1.0 1.5.

Graphics Output Options

This section describes the options for using ODS Graphics with the SEQTEST procedure to create plots. To request these plots, you must specify the ODS GRAPHICS ON statement in addition to the following options in the PROC SEQTEST statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

The following options can be used in the PROC SEQTEST statement to display plots with ODS Graphics. They are listed in alphabetical order.

PLOTS < (ONLY) > < = *plot-request* >

PLOTS < (ONLY) > < = (*plot-request* < . . . *plot-request* >) >

specifies options that control the details of the plots. The default is PLOTS=TEST. The global plot option ONLY suppresses the default plots and displays only plots specifically requested.

The plot request options include the following.

ALL

produces all appropriate plots.

ASN < (CREF=*numbers*) >

displays a plot of the average sample numbers (expected sample sizes for nonsurvival data or expected number of events for survival data) under various hypothetical references, where the numbers $c_i \geq 0$.

For a one-sided design, expected sample numbers under hypotheses $\theta = c_i\ \theta_1$ are displayed, where θ_1 is the alternative reference and c_i are the values specified in the CREF= option.

For a two-sided design, expected sample numbers under hypotheses $\theta = c_i \theta_{1l}$ and $\theta = c_i \theta_{1u}$ are displayed, where θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively. Note that for a symmetric two-sided design, only the average sample numbers under hypotheses $\theta = c_i\ \theta_{1u}$ are derived. The default is CREF= 0 to 1.5 by 0.01.

CONDPOWER < (CREF=*numbers*) >

displays a plot of conditional powers given the most recently observed statistic under specified hypothetical references, where the numbers $c_i \geq 0$. In the SEQTEST procedure, the conditional power is the probability that the test statistic at the final stage would exceed the rejection critical value given the observed statistic.

For a one-sided test, the powers are derived under hypothetical references $\theta = \hat{\theta}$ and $\theta = c_i \theta_1$, where $\hat{\theta}$ is the observed statistic, θ_1 is the alternative reference, and c_i are the values specified in the CREF= option. For a two-sided test, the powers are derived under hypothetical references $\theta = \hat{\theta}$, $\theta = c_i \theta_{1l}$, and $\theta = c_i \theta_{1u}$, where θ_{1l} is the lower alternative reference and θ_{1u} is the upper alternative reference. The default is CREF= 0 to 1.5 by 0.01.

ERRSPEND < (HSCALE=INFO | STAGE) >

displays a plot of the error spending for all sequential boundaries in the designs simultaneously. You can display the information level (HSCALE=INFO) or the stage number (HSCALE=STAGE) on the horizontal axis. With HSCALE=INFO, the information fractions are used in the plot. The default is HSCALE=STAGE.

NONE

suppresses all plots.

POWER < (CREF=*numbers*) >

displays a plot of the power curves under various hypothetical references, where the numbers $c_i \geq 0$.

For a one-sided design, powers under hypotheses $\theta = c_i\,\theta_1$ are displayed, where θ_1 is the alternative reference and c_i are the values specified in the CREF= option.

For a two-sided design, powers under hypotheses $\theta = c_i\theta_{1l}$ and $\theta = c_i\theta_{1u}$ are displayed, where θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively. Note that for a symmetric two-sided design, only powers under hypotheses $\theta = c_i\,\theta_{1u}$ are derived. The default is CREF= 0 to 1.5 by 0.01.

RCI

displays a plot of repeated confidence intervals. Repeated confidence intervals include both rejection and acceptance confidence intervals.

With the STOP=REJECT or STOP=BOTH option, rejection confidence limits can be derived and the null hypothesis $H_0 : \theta = 0$ is rejected if the lower rejection confidence limit is greater than 0 or the upper rejection confidence limit is less than 0.

With the STOP=ACCEPT or STOP=BOTH option, acceptance confidence limits can be derived and the null hypothesis is accepted with alternative hypotheses $H_{1l} : \theta = \theta_{1l}$ and $H_{1u} : \theta = \theta_{1u}$ if the upper acceptance confidence limit is less than θ_{1u} and the lower acceptance confidence limit is greater than θ_{1l}.

TEST < (HSCALE=INFO | SAMPLESIZE) >

displays a plot of the sequential boundaries and test variables. Either the information level (HSCALE=INFO) or the sample size (HSCALE=SAMPLESIZE) is displayed on the horizontal axis. The HSCALE=SAMPLESIZE option is applicable only if the sample size information is available in both the input BOUNDARY= data set and input DATA= data set. The stage number for each stage is displayed inside the plot. The default is HSCALE=INFO.

Details: SEQTEST Procedure

Input Data Sets

The BOUNDARY= data set option is required, and if neither the DATA= nor the PARMS= data set option is specified, the procedure derives statistics such as Type I and Type II error probabilities from the BOUNDARY= data set. The resulting boundaries are displayed with the scale specified in the BOUNDARYSCALE= option.

BOUNDARY= SAS Data Set

The BOUNDARY= data set provides the boundary information for the sequential test. At stage 1, the data set is usually created with an ODS OUTPUT statement from the "Boundary Information" table created by the SEQDESIGN procedure. At each subsequent stage, the data set is usually created with an ODS OUTPUT statement from the "Test Information" table that was created by the

SEQTEST procedure at the previous stage. See the section "Getting Started: SEQTEST Procedure" on page 5988 for an illustration of the BOUNDARY= data set option.

The BOUNDARY= data set contains the following variables:

- _Scale_, the boundary scale, with the value MLE for the maximum likelihood estimate, STDZ for the standardized Z, SCORE for the score statistic, or PVALUE for the nominal p-value. Note that for a two-sided design, the nominal p-value is the one-sided fixed-sample p-value under the null hypothesis with a lower alternative hypothesis.
- _Stop_, the stopping criterion, with the value REJECT for rejecting the null hypothesis H_0, ACCEPT for accepting H_0, or BOTH for both rejecting and accepting H_0
- _ALT_, the type of alternative hypothesis, with the value UPPER for an upper alternative, LOWER for a lower alternative, or TWOSIDED for a two-sided alternative
- _Stage_, the stage number
- the boundary variables, a subset of Bound_LA for lower α boundary, Bound_LB for lower β boundary, Bound_UB for upper β boundary, and Bound_UA for upper α boundary
- AltRef_L, the lower alternative reference, if ALT=LOWER or ALT=TWOSIDED
- AltRef_U, the upper alternative reference, if ALT=UPPER or ALT=TWOSIDED
- _InfoProp_, the information proportion at each stage

Optionally, the BOUNDARY= data set also contains the following variables:

- _Info_, the information level at each stage
- NObs, the required number of observations for nonsurvival data at each stage
- Events, the required number of events for survival data at each stage
- Parameter, the variable specified in the DATA(TESTVAR=) or PARMS(TESTVAR=) option
- Estimate, the parameter estimate

If the BOUNDARY= data set contains the variable Parameter for the test variable that is specified in the TESTVAR= option, and the variable Estimate for the test statistics, then these test statistics are also displayed in the output test information table and output test plot.

DATA < (TESTVAR= *variable*) > = SAS Data Set

The DATA= data set provides the test variable information for the current stage of the trial. Such data sets are usually created with an ODS OUTPUT statement by using a procedure such as PROC MEANS. See "Example 78.4: Testing a Binomial Proportion" on page 6079 for an illustration of the DATA= data set option.

The DATA= data set includes the following variables:

- _Stage_, the stage number
- _Scale_, the scale for the test statistic, with the value MLE for the maximum likelihood estimate, STDZ for the standardized Z, SCORE for the score statistic, or PVALUE for the nominal p-value
- _Info_, the information level
- NObs, the number of observations for nonsurvival data at each stage
- Events, the number of events for survival data at each stage
- test variable, specified in the TESTVAR= option, contains the test variable value in the scale specified in the _Scale_ variable

With the specified DATA= data set, the procedure derives boundary values from the information levels in the _Info_ variable. If the data set does not include the _Info_ variable, then the information levels are derived from the NObs or Events variable in the DATA= data set if the variable is also in the input BOUNDARY= data set. That is, the information level at stage k is computed as $I_k^* = I_k \times (n_k^*/n_k)$, where I_k and n_k are the information level and sample size at stage k in the BOUNDARY= data set and n_k^* is the sample size at stage k in the DATA= data set. Otherwise, the information levels from the BOUNDARY= data set are used.

If the TESTVAR= option is specified, the DATA= data set must also include the test variable for the test statistic and _Scale_ variable for the corresponding scale. Note that for a two-sided design, the nominal p-value is the one-sided fixed-sample p-value under the null hypothesis with a lower alternative hypothesis.

PARMS < (TESTVAR= *variable*) > = SAS Data Set

The PARMS= data set provides a parameter estimate and associated standard error for the current stage of the trial. Such data sets are usually created with an ODS OUTPUT statement by using procedures such as the GENMOD, GLM, LOGISTIC, and REG procedures. See the section "Getting Started: SEQTEST Procedure" on page 5988 for an illustration of the PARMS= data set option.

The PARMS= data set includes the following variables:

- _Stage_, the stage number
- _Scale_, the scale for the test statistic, with the value MLE for the maximum likelihood estimate, STDZ for the standardized Z, SCORE for the score statistic, or PVALUE for the nominal p-value
- Parameter, Effect, Variable, or Parm, which contains the variable specified in the TESTVAR= option
- Estimate, the parameter estimate
- StdErr, standard error of the parameter estimate

With the specified PARMS= data set, the information level is derived from the StdErr variable. For a score statistic, the information level I_k is the variance of the statistic, $\hat{s}_k^2$, where $\hat{s}_k$ is the standard error in the StdErr variable. Otherwise, the information level is the inverse of the variance of the statistic, $\hat{s}_k^{-2}$. If the data set does not include the StdErr variable, the information levels derived from the BOUNDARY= data set are used.

If the TESTVAR= option is specified, the PARMS= data set also includes the variable Parameter, Effect, Variable, or Parm for the test variable, Estimate for the test statistic, and _Scale_ variable for the corresponding scale. Note that for a two-sided design, the nominal p-value is the one-sided fixed-sample p-value under the null hypothesis with a lower alternative hypothesis.

Boundary Variables

The boundaries created in group sequential trials depend on the type of the alternative hypothesis and the early stopping criterion. Table 78.2 shows the boundaries created with various design specifications.

Table 78.2 Boundary Variables

Specifications		Boundary Variables			
Alternative		**Lower**		**Upper**	
Hypothesis	**Early Stopping**	**Alpha**	**Beta**	**Beta**	**Alpha**
Lower	Accept H_0		X		
	Reject H_0	X			
	Accept/Reject H_0	X	X		
Upper	Accept H_0			X	
	Reject H_0				X
	Accept/Reject H_0			X	X
Two-sided	Accept H_0		X	X	
	Reject H_0	X			X
	Accept/Reject H_0	X	X	X	X

Up to four different boundaries can be generated in a group sequential design:

- the upper α boundary, used to reject the null hypothesis in favor of an upper alternative hypothesis
- the upper β boundary, used to accept the null hypothesis with an upper alternative hypothesis
- the lower β boundary, used to accept the null hypothesis with a lower alternative hypothesis
- the lower α boundary, used to reject the null hypothesis in favor of a lower alternative hypothesis

For a two-sided design, the null hypothesis is accepted only if both the null hypothesis is accepted with an upper alternative hypothesis and the null hypothesis is accepted with a lower alternative hypothesis.

For a one-sided design with a lower alternative, only the lower boundaries are created. Similarly, for a one-sided design with an upper alternative, only the upper boundaries are created. For example, Figure 78.19 shows the boundary plot for a one-sided test with an upper alternative.

Figure 78.19 Boundary Plot for One-Sided Test

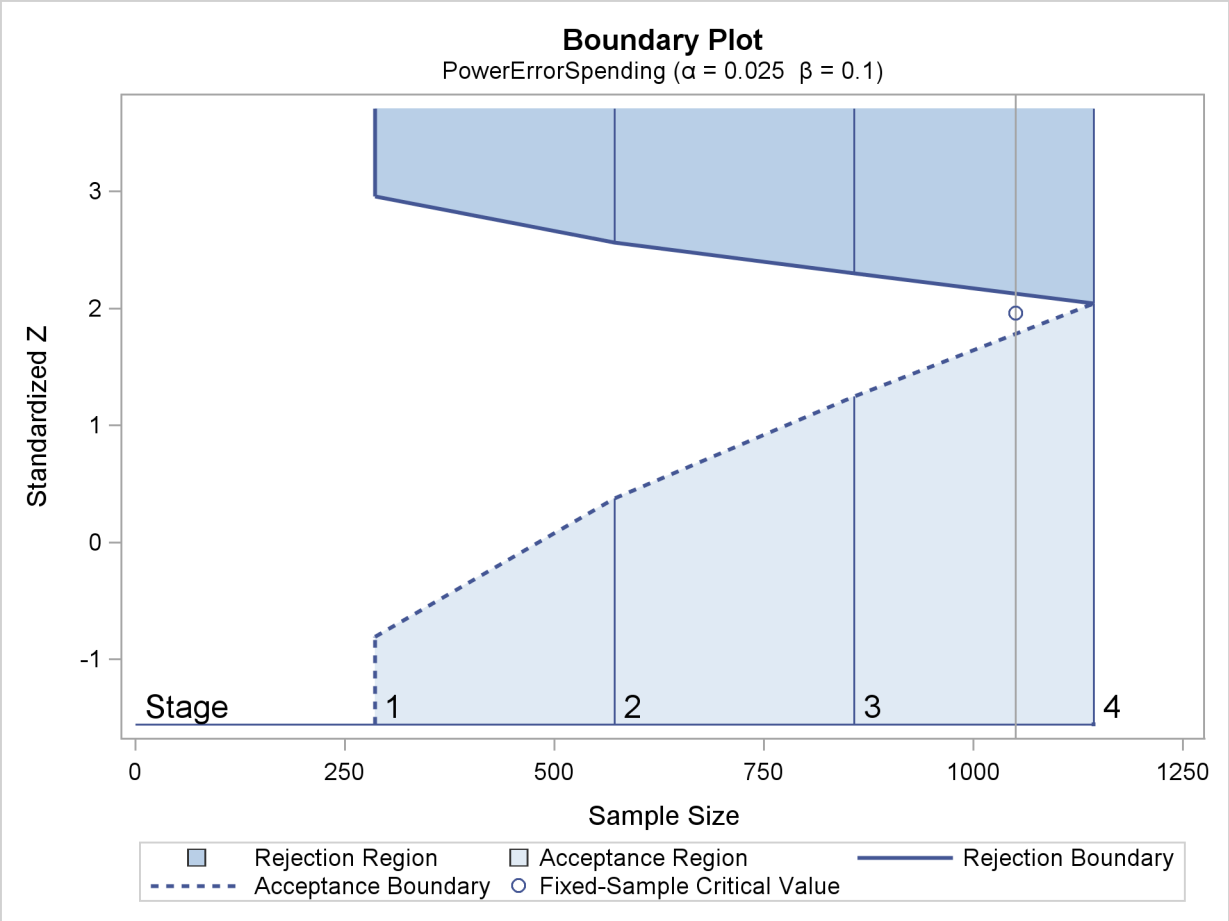


Figure 78.19 corresponds to a one-sided sequential design with early stopping to reject or accept the null hypothesis. For a sequential test with early stopping only to reject the null hypothesis, there are no acceptance boundary values at interim stages. The acceptance boundary value and its associated acceptance region are displayed only at the final stage. Similarly, for a sequential test with early stopping only to accept the null hypothesis, there are no rejection boundary values at interim stages. The rejection boundary value and its associated rejection region are displayed only at the final stage.

For a two-sided design, both the lower and upper boundaries are created. For a design with early stopping to reject the null hypothesis, α boundaries are created. Similarly, for a design with early stopping to accept the null hypothesis, β boundaries are created. For a design with early stopping to accept or reject the null hypothesis, both the α and β boundaries are created.

For example, Figure 78.20 shows the boundary plot for a two-sided test.

Figure 78.20 Boundary Plot for Two-Sided Test

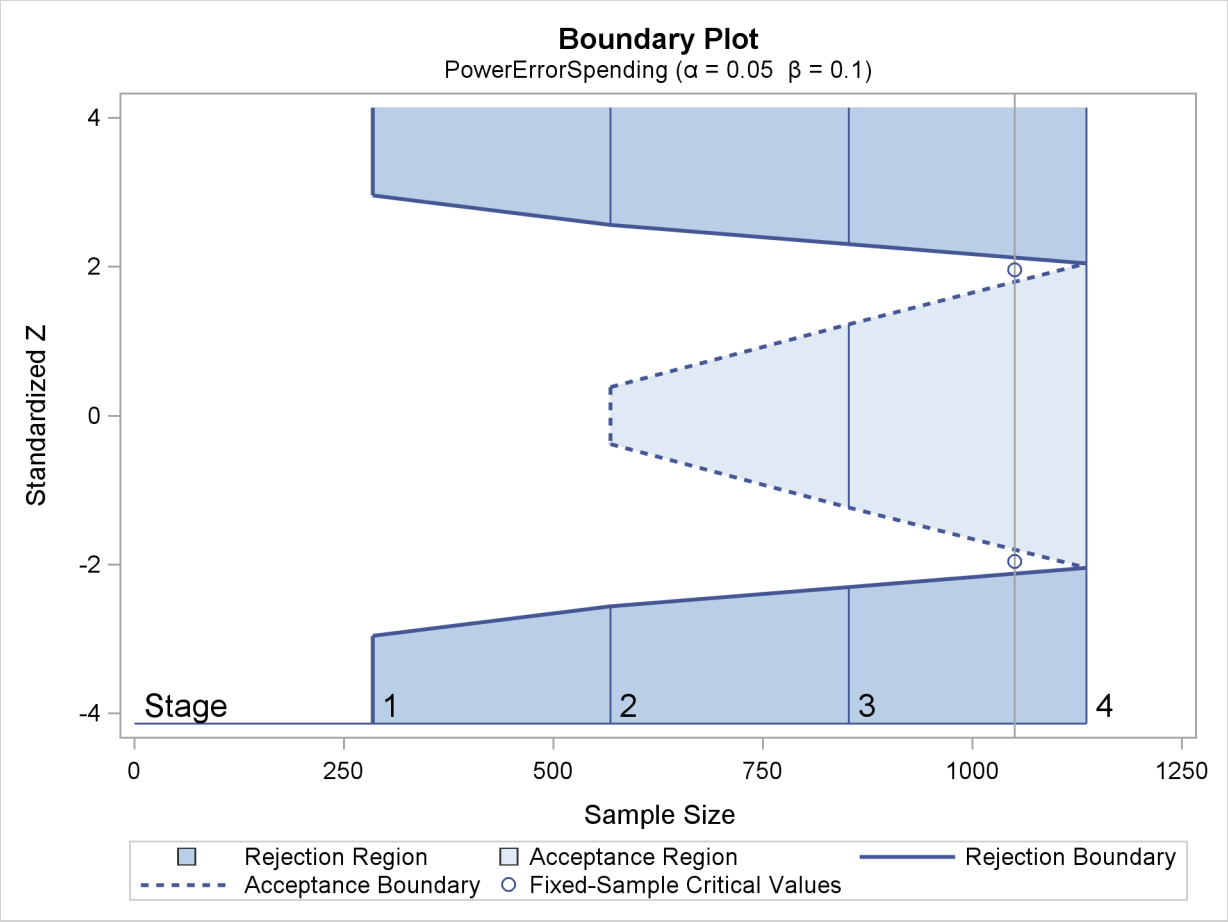


Figure 78.20 corresponds to a two-sided sequential design with early stopping to reject or accept the null hypothesis. For a sequential test with early stopping only to reject the null hypothesis, there are no acceptance boundary values at interim stages. The acceptance boundary value and its associated acceptance region are displayed only at the final stage. Similarly, for a sequential test with early stopping only to accept the null hypothesis, there are no rejection boundary values at interim stages. The rejection boundary value and its associated rejection region are displayed only at the final stage.

Boundary Adjustments for Information Levels

In a group sequential clinical trial, the information level for the observed test statistic generally does not match the corresponding information level in the BOUNDARY= data set. In this situation, the SEQTEST procedure first modifies the information levels at future interim stages to accommodate the information level for the test statistic and computes the error spending values at the current and future stages. The SEQTEST procedure then derives boundary values at the current and future stages for the trial with these new error spending values.

Denote the information level at stage k for the K-stage design stored in the BOUNDARY= data set by I_k, $k = 1, 2, \ldots, K$. Also denote the information level that corresponds to the test statistic at an interim stage k_0 by I'_{k_0}, $1 \le k_0 \le K-1$. Then for the updated design, the information level at stage k, $k = k_0 + 1, \ldots, K-1$, is computed as

$$I'_k = I'_{k_0} + (I_K - I'_{k_0}) \frac{I_k - I_{k_0}}{I_K - I_{k_0}}$$

Note that if $I'_{k_0} \ge I_K$, the information level at stage k_0 reaches the maximum information level in the design, the trial stops at stage k_0, and no future information levels are derived.

The BOUNDARYADJ= option provides various methods to compute error spending values for these new information levels at the current and future interim stages.

The BOUNDARYADJ=NONE option keeps the error spending the same at each stage. The BOUNDARYADJ=ERRLINE option uses a linear interpolation on the cumulative error spending in the design stored in the BOUNDARY= data set to derive the error spending for each unmatched information level (Kittelson and Emerson 1999, p. 882). That is, the cumulative error spending for an information level I is computed as

$$e(I) = \begin{cases} e_1 \left(\frac{I}{I_1}\right) & \text{if } I < I_1 \\ e_j + (\alpha_{j+1} - \alpha_j) \left(\frac{I - I_j}{I_{j+1} - I_j}\right) & \text{if } I_j \le I < I_{j+1} \\ e_K & \text{if } I \ge I_K \end{cases}$$

where $e_1, e_2, \ldots, e_K$ are the cumulative errors at the K stages of the design that is stored in the BOUNDARY= data set.

The BOUNDARYADJ=ERRFUNCPOC option uses Pocock-type cumulative error spending function (Lan and DeMets 1983):

$$E(t) = \begin{cases} 1 & \text{if } t \ge 1 \\ \log(1 + (e-1)t) & \text{if } 0 < t < 1 \\ 0 & \text{otherwise} \end{cases}$$

With an error level of α or β, the cumulative error spending for an information level I is $e(I) = \alpha\, E(I/I_K)$ or $e(I) = \beta\, E(I/I_K)$.

The BOUNDARYADJ=ERRFUNCOBF option uses O'Brien-Fleming-type cumulative error spending function (Lan and DeMets 1983):

$$E(t; a) = \begin{cases} 1 & \text{if } t \ge 1 \\ \frac{1}{a}\, 2\,(1 - \Phi(\frac{z_{(1-a/2)}}{\sqrt{t}})) & \text{if } 0 < t < 1 \\ 0 & \text{otherwise} \end{cases}$$

where a is either α for the α spending function or β for the β spending function, and Φ is the cumulative distribution function of the standardized Z statistic. That is, with an error level of α or β, the cumulative error spending for an information level I is $e(I) = \alpha\, E(I/I_K; \alpha)$ or $e(I) = \beta\, E(I/I_K; \beta)$.

The BOUNDARYADJ=ERRFUNCGAMMA option uses gamma cumulative error spending function (Hwang, Shih, and DeCani 1990):

$$E(t;\gamma) = \begin{cases} 1 & \text{if } t \geq 1 \\ \frac{1-e^{-\gamma t}}{1-e^{-\gamma}} & \text{if } 0 < t < 1, \gamma \neq 0 \\ t & \text{if } 0 < t < 1, \gamma = 0 \\ 0 & \text{otherwise} \end{cases}$$

where γ is the parameter γ specified in the GAMMA= option. That is, with an error level of α or β, the cumulative error spending for an information level I is $e(I) = \alpha\, E(I/I_K;\gamma)$ or $e(I) = \beta\, E(I/I_K;\gamma)$.

The BOUNDARYADJ=ERRFUNCPOW option uses power cumulative error spending function (Jennison and Turnbull 2000, p. 148):

$$E(t;\rho) = \begin{cases} 1 & \text{if } t \geq 1 \\ t^{\rho} & \text{if } 0 < t < 1 \\ 0 & \text{otherwise} \end{cases}$$

where ρ is the power parameter specified in the RHO= suboption. With an error level of α or β, the cumulative error spending for an information level I is $e(I) = \alpha\, E(I/I_K;\rho)$ or $e(I) = \beta\, E(I/I_K;\rho)$.

If the BOUNDARYKEY=BOTH option is specified, the maximum information (information level at the final stage) required for the trial might not be the same as the maximum information level stored in the BOUNDARY= data set. In this case, the information levels at future stages are adjusted proportionally, and the same error spending values that were computed based on the maximum information level stored in the BOUNDARY= data set are used to derive boundary values for the trial.

If an error spending function is used to create boundaries for the original design in the SEQDESIGN procedure, then in order to better maintain the design features throughout the group sequential trial, the same error spending function to create boundaries for the original design in the SEQDESIGN procedure should be used to modify boundaries in the SEQTEST procedure at each subsequent stage.

Boundary Adjustments for Minimum Error Spending

In a group sequential clinical trial, boundary values created from a design such as an O'Brien-Fleming design might be too conservative in early stages. Thus the trial is unlikely to stop in early stages. Lan and Demets (1983, p. 662) suggest truncating boundary values to a number such as 3.5 for the trial to have a reasonable probability of stopping at early stages. Instead of truncating boundary values by a specified number, the ERRSPENDMIN= option provides individual minimum error spending at each interim stage to stop the trial early.

For a K-stage trial, denote the derived cumulative error spending at stage k after adjusting for information levels by $e_k, k = 1, 2, \ldots, K$. Also denote the specified minimum error spending at interim stage k by $\epsilon_k, k = 1, 2, \ldots, K-1$. Then the cumulative error spending at stage 1 is $e'_1 = \max(e_1,\ \epsilon_1)$. If $e_1 < e'_1$, the error spending values at subsequent interim stages are adjusted

proportionally by

$$e'_j = e'_1 + \frac{e_j - e_1}{e_K - e_1}\,(e_K - e'_1)$$

for $j = 2, \ldots, K-1$.

The process is repeated at each subsequent interim stage. That is, at stage $k, k = 2, \ldots, K-1$, denote the updated cumulative β spending at stage j by e_j, $j = k, k+1, \ldots, K$. Then the cumulative error spending at stage k is $e'_k = \max(e_k,\ e'_{k-1} + \epsilon_k)$. If $e_k < e'_k$, the error spending values at subsequent interim stages are adjusted proportionally by

$$e'_j = e'_k + \frac{e_j - e_k}{e_K - e_k}\,(e_K - e'_k)$$

for $j = k+1, \ldots, K-1$.

Note that the ERRSPENDMIN= option is applicable only to the boundaries specified in the BOUNDARYKEY= option. That is, the ERRSPENDMIN= option is applicable to the α boundaries with BOUNDARYKEY=ALPHA or BOUNDARYKEY=BOTH, and it is applicable to the β boundaries with BOUNDARYKEY=BETA or BOUNDARYKEY=BOTH.

Boundary Adjustments for Overlapping Lower and Upper β Boundaries

In the SEQTEST procedure, the α and β spending values at the stages are used to derive the boundary values for the trial. For a two-sided design with early stopping to accept H_0, or to either reject or accept H_0, a zero β spending at an interim stage sets the β boundary values to missing. A small β spending at the current or subsequent interim stage might result in overlapping of the lower and upper β boundaries for the two corresponding one-sided tests. Specifically, this form of overlapping occurs at an interim stage k if the upper β boundary value that is derived from the one-sided test for the upper alternative is less than the lower β boundary value that is derived from the one-sided test for the lower alternative (Kittelson and Emerson 1999, pp. 881–882; Rudser and Emerson 2007, p. 6). You can use the BETAOVERLAP= option to specify how this type of overlapping is to be handled.

If BETAOVERLAP=ADJUST (which is the default) is specified, the procedure derives the boundary values for the two-sided design and then checks for overlapping of the two one-sided β boundaries at the current and subsequent interim stages. If overlapping occurs at a particular stage, the β boundary values for the two-sided design are set to missing (so the trial does not stop to accept the null hypothesis at this stage), and the β spending values at subsequent stages are adjusted proportionally as follows.

If the β boundary values are set to missing at stage k in a K-stage trial, the adjusted β spending value at stage k, e'_k, is updated for these missing β boundary values, and then the β spending values at subsequent stages are adjusted proportionally by

$$e'_j = e'_k + \frac{e_j - e_k}{e_K - e_k}\,(e_K - e'_k)$$

for $j = k+1, \ldots, K$, where e_j and e'_j are cumulative β spending values at stage j before and after the adjustment, respectively.

After all these adjusted β spending values are computed, the boundary values are then further modified for these adjusted β spending values.

If you specify BETAOVERLAP=NOADJUST, no adjustment is made when overlapping of one-sided β boundaries occurs.

Stochastic Curtailment

Lan, Simon, and Halperin (1982) introduce stochastic curtailment to stop a trial if, given current data, it is likely to predict the outcome of the trial with high probability. That is, a trial can be stopped to reject the null hypothesis H_0 if the conditional probability of rejecting H_0 under H_0, given current data in the analyses at the end of the trial, is greater than γ, where the constant γ should be between 0.5 and 1, and values of 0.8 or 0.9 are recommended (Jennison and Turnbull 2000, p. 206). Similarly, a trial can be stopped to accept the null hypothesis H_0 if the conditional probability of rejecting H_0 under the alternative hypothesis H_1, given current data in the analyses at the end of the trial, is less than γ.

The following two approaches for stochastic curtailment are available in the SEQTEST procedures: conditional power approach and predictive power approach. For each approach, the derived group sequential test is used as the reference test for rejection.

Conditional Power Approach

In the SEQTEST procedure, the conditional power at an interim stage k is the probability that the test statistic at the final stage (stage K) would exceed the rejection critical value (Cui, Hung, and Wang 1999, p. 854; Emerson, Kittelson, and Gillen 2005, p. 13). If there exist interim stages between the kth stage and the final stage, $k < K-1$, the conditional power is not the conditional probability to reject the null hypothesis H_0. In this case, you can set the next stage as the final stage, and the conditional power is the conditional probability to reject H_0.

The conditional distribution of Z_K given the observed statistic z_k at the kth stage and the hypothetical reference θ is

$$Z_K \,|\, (z_k, \theta) \sim N\left(z_k \ \Pi_k^{\frac{1}{2}} + \theta \ I_X^{\frac{1}{2}} (1 - \Pi_k) \,,\ 1 - \Pi_k \right)$$

where $\Pi_k = I_k / I_X$ is the fraction of information at the kth stage.

The power for the upper alternative, prob($Z_K > a_K \,|\, z_k, \theta$), is then given by

$$p_{ku}(\theta) = \Phi\left((1 - \Pi_k)^{-\frac{1}{2}} \left(z_k \ \Pi_k^{\frac{1}{2}} - a_K \right) \ + \ \theta \ I_X^{\frac{1}{2}} (1 - \Pi_k)^{\frac{1}{2}} \right)$$

where Φ is the cumulative distribution function of the standardized Z statistic and a_K is the upper critical value at the final stage.

Similarly, the power for the lower alternative, prob($Z_K < a\_K \,|\, z_k, \theta$), is

$$p_{kl}(\theta) = 1 - \Phi\left((1 - \Pi_k)^{-\frac{1}{2}} \left(z_k \ \Pi_k^{\frac{1}{2}} - a\_K \right) \ + \ \theta \ I_X^{\frac{1}{2}} (1 - \Pi_k)^{\frac{1}{2}} \right)$$

where a_K is the lower critical value at the final stage.

A special case of the conditional power is the futility index (Ware, Muller, and Braunwald, 1985). It is one minus the conditional power under $H_1 : \theta = \theta_1$:

$$1 - p_{ku}(\theta_1) \text{ or } 1 - p_{kl}(\theta_1)$$

That is, it is the probability of accepting the null hypothesis under the alternative hypothesis given current data. A high futility index indicates a small probability of success (rejecting H_0) given the current data.

If $\theta = \hat{\theta}_k = z_k I_k^{-\frac{1}{2}}$, the maximum likelihood estimate at stage k, the powers for the upper and lower alternatives can be simplified:

$$p_{ku}(\theta) = \Phi\left((1-\Pi_k)^{-\frac{1}{2}}\ (z_k\ \Pi_k^{-\frac{1}{2}} - a_K)\right)$$

$$p_{kl}(\theta) = 1 - \Phi\left((1-\Pi_k)^{-\frac{1}{2}}\ (z_k\ \Pi_k^{-\frac{1}{2}} - a\_K)\right)$$

Predictive Power Approach

The conditional power depends on the specified reference θ, which might be supported by the current data (Jennison and Turnbull 2000, p. 210). An alternative is to use the predictive power (Herson 1979), which is a weighted average of the conditional power over values of θ. Without prior knowledge about θ, then with $\hat{\theta} = z_k/\sqrt{I_k}$, the maximum likelihood estimate at stage k, the posterior distribution for θ (Jennison and Turnbull 2000, p. 211) is

$$\theta \mid Z_K \sim N\left(\frac{z_k}{\sqrt{I_k}},\ \frac{1}{I_k}\right)$$

Thus, the predictive power at stage k for the upper and lower alternatives can be derived as

$$p_{ku} = 1 - \Phi\left((1-\Pi_k)^{-\frac{1}{2}}\ (a_K\ \Pi_k^{\frac{1}{2}} - z_k)\right)$$

$$p_{kl} = \Phi\left((1-\Pi_k)^{-\frac{1}{2}}\ (a\_K\ \Pi_k^{\frac{1}{2}} - z_k)\right)$$

where a_K and a_K are the upper and lower critical values at the final stage.

Repeated Confidence Intervals

In a group sequential test, repeated confidence intervals for a parameter θ are defined as a sequence of intervals $(\hat{\theta}_{kl}, \hat{\theta}_{ku})$, $k = 1, 2, \ldots, K$, for which a simultaneous coverage probability is maintained (Jennison and Turnbull 2000, p. 189). That is, a $(1-\alpha)$ sequence of repeated confidence intervals has

$$\text{Prob}(\hat{\theta}_{kl} \le \theta \le \hat{\theta}_{ku}) = 1 - \alpha$$

These confidence limits $\hat{\theta}_{kl}$ and $\hat{\theta}_{ku}$ can be created from observed statistic and boundary values at each stage.

Two-Sided Repeated Confidence Intervals

Two sequences of repeated confidence intervals can be derived for a two-sided test. One is a $(1-\alpha_l-\alpha_u)$ rejection repeated confidence intervals $(\hat{\theta}_{kl}(\alpha), \hat{\theta}_{ku}(\alpha)), k = 1, 2, \ldots, K$, and the other is a $(1-\beta_l-\beta_u)$ acceptance repeated confidence intervals $(\hat{\theta}_{kl}(\beta), \hat{\theta}_{ku}(\beta)), k = 1, 2, \ldots, K$, where α_l and α_u are the lower and upper Type I error probabilities for the test and β_l and β_u are the lower and upper Type II error probabilities for the test (Jennison and Turnbull 2000, p. 196).

The rejection lower and upper repeated confidence limits at stage k are

$$\hat{\theta}_{kl}(\alpha) = \hat{\theta}_k - \frac{a_k}{\sqrt{I_k}} \qquad \hat{\theta}_{ku}(\alpha) = \hat{\theta}_k - \frac{a\_k}{\sqrt{I_k}}$$

The hypothesis is rejected for upper alternative if the lower limit $\hat{\theta}_{kl}(\alpha) > \theta_{0u}$ and is rejected for lower alternative if the upper limit $\hat{\theta}_{ku}(\alpha) < \theta_{0l}$. That is, the hypothesis is rejected if both θ_{0l} and θ_{0u} are not in a rejection repeated confidence interval $(\hat{\theta}_{kl}(\alpha), \hat{\theta}_{ku}(\alpha))$.

The acceptance lower and upper repeated confidence limits at stage k are

$$\hat{\theta}_{kl}(\beta) = \hat{\theta}_k + \left(\theta_{1l} - \frac{b\_k}{\sqrt{I_k}}\right) \qquad \hat{\theta}_{ku}(\beta) = \hat{\theta}_k + \left(\theta_{1u} - \frac{b_k}{\sqrt{I_k}}\right)$$

The hypothesis is accepted if the lower limit $\hat{\theta}_{kl}(\beta) > \theta_{1l}$ and the upper limit $\hat{\theta}_{ku}(\beta) < \theta_{1u}$. That is, a repeated confidence interval is contained in the interval $(\theta_{1l}, \theta_{1u})$.

One-Sided Repeated Confidence Intervals

Like the two-sided repeated confidence intervals, two sequences of repeated confidence intervals can be derived for a one-sided test. Suppose the one-sided test has an upper alternative θ_{1u}. Then one sequence of repeated confidence intervals is a $(1-\alpha_u)$ rejection repeated confidence intervals $(\hat{\theta}_{kl}(\alpha), \infty), k = 1, 2, \ldots, K$, and the other is a $(1-\beta_u)$ acceptance repeated confidence intervals $(-\infty, \hat{\theta}_{ku}(\beta)), k = 1, 2, \ldots, K$, where α_u and β_u are the upper Type I and Type II error probabilities for the test. Thus, a sequence of repeated confidence intervals with confidence level greater than or equal to $(1-\alpha_u-\beta_u)$ is given by $(\hat{\theta}_{kl}(\alpha), \hat{\theta}_{ku}(\beta))$.

The rejection lower repeated confidence limit and the acceptance upper repeated confidence limit at stage k are

$$\hat{\theta}_{kl}(\alpha) = \hat{\theta}_k - \left(\frac{a_k}{\sqrt{I_k}} - \theta_{0u}\right) \qquad \hat{\theta}_{ku}(\beta) = \hat{\theta}_k + \left(\theta_{1u} - \frac{b_k}{\sqrt{I_k}}\right)$$

The hypothesis is rejected if the lower limit $\hat{\theta}_{kl}(\alpha) > \theta_{0u}$. and it is accepted if the upper limit $\hat{\theta}_{ku}(\beta) < \theta_{1u}$.

Analysis after a Sequential Test

At the end of a trial, the hypothesis is either rejected or accepted. But the p-value, median, and confidence limits depend on the ordering the sample space (k, z), where k is the stage number and z is the standardized Z statistic.

Following the notations used in Jennison and Turnbull (2000, pp. 179–180), $(k', z') \succ (k, z)$ if (k', z') has a higher order or more extreme than (k, z). Then for a given ordering, the p-value, median, and confidence limits associated with the observed statistics (k, z) can be derived.

p-value

With the observed pair of statistics (k_0, z_0) when the trial is stopped, a one-sided upper p-value is computed as

$$\text{Prob}\{\, (k, z) \succeq (k_0, z_0) \,\}$$

A one-sided lower p-value is computed as

$$\text{Prob}\{\, (k, z) \preceq (k_0, z_0) \,\}$$

A two-sided p-value is twice the smaller of the lower and upper p-values.

Median Unbiased Estimate

With the observed pair (k_0, z_0), a median unbiased estimate θ_m is computed from

$$\text{Prob}\{\, (k, z) \succeq (k_0, z_0) \mid \theta_m \,\} = 0.50$$

Confidence Limits

With the observed pair (k_0, z_0), a lower $(1 - \alpha_l)$ confidence limit for θ, θ_l, is computed from

$$\text{Prob}\{\, (k, z) \succeq (k_0, z_0) \mid \theta_l \,\} = \alpha_l$$

Similarly, an upper $(1 - \alpha_u)$ confidence limit for θ, θ_u, is computed from

$$\text{Prob}\{\, (k, z) \preceq (k_0, z_0) \mid \theta_u \,\} = \alpha_u$$

Available Sample Space Orderings in a Sequential Test

At the end of a trial, the hypothesis is either rejected or accepted. Denote the stage number and the statistic at the end of a trial by a pair of statistics (k, z), where k is the stage number and z is the standardized Z statistic. Then an ordering on the sample space (k, z) is needed to derive the p-value, median, and confidence limits associated with the observed statistics (k^*, z^*).

The SEQTEST procedure provides the stagewise, LR, and MLE orderings. Refer to Jennison and Turnbull (2000 pp. 179–187) for a detailed description and comparison of these orderings.

Stagewise Ordering

If the continuation regions of a design are intervals, the stagewise ordering (Fairbanks and Madsen 1982; Tsiatis, Rosner, and Mehta 1984; Jennison and Turnbull 2000, pp. 179–180) uses counterclockwise ordering around the continuation region to compute the p-value, unbiased median estimate, and confidence limits. This ordering depends on the stopping region, stopping stage, and standardized statistic at the stopping stage. But it does not depend on information levels beyond the observed stage. For a one-sided design with an upper alternative, $(k', z') \succ (k, z)$ if one of the following criteria holds:

- $k' = k$ and $z' > z$
- $k' < k$ and $z' \geq a_{k'}$, the upper α boundary at stage k'
- $k' > k$ and $z < b_k$, the upper β boundary at stage k

Similar criteria can be derived for a one-sided design with a lower alternative.

For a two-sided design with early stopping to reject the null hypothesis, $(k', z') \succ (k, z)$ if one of the following criteria holds:

- $k' = k$ and $z' > z$
- $k' < k$ and $z' \geq a_{k'}$, the upper α boundary at stage k'
- $k' > k$ and $z \leq \_a_k$, the lower α boundary at stage k

Note that the stagewise ordering is not applicable for two-sided designs with early stopping to accept H_0 or to either accept or reject H_0, which might have two disjoint continuous intervals at each interim stage.

For a two-sided design with early stopping either to reject or to accept the null hypothesis, $(k', z') \succ (k, z)$ if one of the following criteria holds:

- $z' \geq a_{k'}$ and $z < b_k$
- $z' > \_b_{k'}$ and $z \leq \_a_k$

That is, each value in the continuation region is less extreme than each value in the upper rejection region and more extreme than each value in the lower rejection region. Then, combining with the ordering defined for a two-sided design with early stopping to reject the null hypothesis, the p-value, median, and confidence limits can be derived for the observed statistics in the lower or upper rejection region.

Thus, if the stagewise ordering is specified in the SEQTEST procedure for a two-sided design with early stopping to either reject or accept the null hypothesis, the stagewise ordering is used to derive these statistics only if the observed statistics is in the lower or upper rejection region. Otherwise, the LR ordering is used.

LR Ordering

The LR ordering (Chang 1989) depends on the observed standardized Z statistic z, information levels, and a specified hypothetical reference. For the LR ordering under a given hypothesis $H : \theta = \theta_g$, $(k', z') \succ (k, z)$ if

$$(z' - \theta_g \sqrt{I_{k'}}) > (z - \theta_g \sqrt{I_k})$$

Under the null hypothesis $H_0 : \theta = 0$, it reduces to

$$z' > z$$

and can be used to derive statistics under H_0, such as p-values.

The LR ordering is applicable to all designs if all information levels are available. But depending on the boundary shape, some observed statistics (k, z) in the rejection region might be less extreme than the statistics in the acceptance region. That is, the p-value for observed statistics in the rejection region might be greater than the significance level.

MLE Ordering

The MLE ordering (Emerson and Fleming 1990) depends only on the observed maximum likelihood estimate. $(k', z') \succ (k, z)$ if

$$\frac{z'}{\sqrt{I_{k'}}} > \frac{z}{\sqrt{I_k}}$$

The MLE ordering is applicable to all designs if all information levels are available.

Applicable Tests and Sample Size Computation

The SEQDESIGN procedure assumes that the data are from a multivariate normal distribution and the sequence of the standardized test statistics $\{Z_1, Z_2, \ldots, Z_K\}$ have the following canonical joint distribution:

- $Z_k \sim N\left(\theta\sqrt{I_k}, 1\right)$
- $\text{Cov}(Z_{k_1}, Z_{k_2}) = \sqrt{I_{k_1}/I_{k_2}} \quad , \quad 1 \le k_1 \le k_2 \le K$

where K is the total number of stages and I_k is the information available at stage k.

If the data are not from a normal distribution such as binomial distribution, then it is assumed that the test statistic is computed from a large sample such that the statistic has an approximately normal distribution.

In a clinical trial, the sample size required depends on the Type I error probability α, reference improvement θ_1, power $1-\beta$, and variance of the response variable. Given a null hypothesis $H_0 : \theta = 0$ with an upper alternative hypothesis $H_1 : \theta = \theta_1$, the information required for a fixed-sample test is given by

$$I_0 = \frac{(\Phi^{-1}(1-\alpha) + \Phi^{-1}(1-\beta))^2}{\theta_1^2}$$

where the parameter θ depends on the test specified in the clinical trial. For example, if you are comparing two binomial populations $H_0 : \theta = 0$, then $\theta = p_t - p_c$ is the difference between two proportions if the proportion difference statistic is used, and $\theta = \log\left(\frac{p_t(1-p_c)}{p_c(1-p_t)}\right)$, the log odds ratio for the two proportions if the log odds ratio statistic is used.

If the maximum likelihood estimate $\hat{\theta}$ from the likelihood function can be derived, then the asymptotic variance for $\hat{\theta}$ is $\text{Var}(\hat{\theta}) = 1/I$, where I is Fisher's information for θ.

The resulting statistic $\hat{\theta}$ corresponds to the MLE scale as specified in the BOUNDARYSCALE=MLE option in the PROC SEQDESIGN statement, $\hat{\theta}\sqrt{I}$ corresponds to the standardized Z scale (BOUNDARYSCALE=STDZ), and $\hat{\theta}I$ corresponds to the score scale (BOUNDARYSCALE=SCORE).

Alternatively, if the score statistic is derived, it can also be used as the test statistic and its asymptotic variance is given by Fisher's information.

For a group sequential trial, the maximum information I_X is derived in the SEQDESIGN procedure by using the specified α, β, and θ_1. With the maximum information

$$I_X = \frac{1}{\text{Var}(\hat{\theta})}$$

the sample size required for a specified test statistic in the trial can be evaluated or estimated from the known or estimated variance of the response variable. Note that different designs might produce different maximum information levels for the same hypothesis, and this in turn might require a different number of observations for the trial.

With a specified test statistic, the resulting information levels can be computed and then used to derive the required sample size. These tests include commonly used tests for normal means, binomial proportions, and survival distributions. See the section "Applicable Tests and Sample Size Computation" in "The SEQDESIGN Procedure" for a description of these tests.

Table Output

The SEQTEST procedure displays the "Design Information" and "Test Information" tables by default.

Conditional Power

The "Conditional Power Information" table displays the following information under a hypothetical reference:

- stopping stage
- MLE, observed maximum likelihood estimate
- conditional power under the hypothetical reference

For a one-sided test, the power are derived under hypothetical references $\theta = \hat{\theta}$ and $\theta = c_i\theta_1$, where $\hat{\theta}$ is the observed statistic, θ_1 is the alternative reference, and c_i are the values specified in the CREF= option. For a two-sided test, the power are derived under the hypothetical references $\theta = \hat{\theta}$, $\theta = c_i\theta_{1l}$, and $\theta = c_i\theta_{1u}$, where θ_{1l} is the lower alternative reference and θ_{1u} is the upper alternative reference. The default is CREF= 0 0.5 1.0 1.5.

Design Information

The "Design Information" table displays the design specifications and derived statistics. The derived Max Information (Percent Fixed-Sample) is the maximum information for the sequential design in percentage of the corresponding fixed-sample information.

The Null Ref ASN (Percent Fixed-Sample) is the average sample size required under the null hypothesis for the group sequential design in percentage of the corresponding fixed-sample design. Similarly, the Alt Ref ASN (Percent Fixed-Sample) is the average sample size required under the alternative reference for the group sequential design in percentage of the corresponding fixed-sample design.

Error Spending Information

The "Error Spending Information" table displays the following information at each stage:

- proportion of information
- actual information level, if the maximum information is either specified or derived
- cumulative error spending for each boundary

Parameter Estimates

The "Parameter Estimates" table displays the following information at the conclusion of a sequential trial:

- stopping stage
- parameter estimate
- median and confidence limits based on the specified ordering
- p-value for the hypothesis H_0 based on the specified ordering

Powers and Expected Sample Sizes

The "Powers and Expected Sample Sizes" table displays the following information under each of the specified hypothetical references $\theta = c_i\theta_1$, where θ_1 is the alternative reference and c_i are values specified in the CREF= option.

- coefficient c_i for the hypothetical references. The value $c_i = 0$ corresponds to the null hypothesis and $c_i = 1$ corresponds to the alternative hypothesis
- power
- expected sample size, as percentage of fixed-sample size

For a one-sided design, the power and expected sample sizes under the hypothetical references $\theta = c_i\theta_1$ are displayed.

For a two-sided symmetric design, the power and expected sample sizes under each of the hypothetical references $\theta = c_i\theta_{1u}$ are displayed, where θ_{1u} is the upper alternative reference.

For a two-sided asymmetric design, the power and expected sample sizes under each of the hypothetical references $\theta = c_i\theta_{1l}$ and $\theta = c_i\theta_{1u}$ are displayed, where θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively.

For a two-sided design, the power is the probability of correctly rejecting the null hypothesis for the correct alternative. Thus, under the null hypothesis, the displayed power corresponds to a one-sided Type I error probability level—that is, the lower α level or the upper α level.

The expected sample size as a percentage of the corresponding fixed-sample design is

$$100 \times \frac{\sum_{k=1}^{K} p_k\, I_k}{I_0}$$

where p_k is the stopping probability at stage k, $\sum_{k=1}^{K} p_k\, I_k$ is the expected information level, and I_0 is the information level for the fixed-sample design.

Predictive Power

The "Predictive Power Information" table displays the following information:

- stopping stage
- MLE, observed maximum likelihood estimate
- predictive power

Repeated Confidence Intervals

The "Repeated Confidence Intervals" table displays the following information for the observed statistic at each stage:

- information level
- parameter estimate
- rejection confidence limits. The null hypothesis is rejected for the upper alternative if the lower rejection confidence limit is greater than the null parameter value. Similarly, the null hypothesis is rejected for the lower alternative if the upper rejection confidence limit is less than the null parameter value.
- acceptance confidence limits. The upper alternative hypothesis is rejected if the upper acceptance confidence limit is less than the upper alternative value. Similarly, the lower alternative hypothesis is rejected if the lower acceptance confidence limit is greater than the upper alternative value. For a two-sided design, if both upper and lower alternative hypothesis are rejected, the null hypothesis is accepted.

Stopping Probabilities

The "Expected Cumulative Stopping Probabilities" table displays the following information under each of the specified hypothetical references $\theta = c_i\theta_1$, where c_i are values specified in the CREF= option, and θ_1 is the alternative reference:

- coefficient c_i for the hypothetical references. The value $c_i = 0$ corresponds to the null hypothesis, and $c_i = 1$ corresponds to the alternative hypothesis
- expected stopping stage
- source of the stopping probability: reject H_0 (with STOP=REJECT or STOP=BOTH), accept H_0 (with STOP=ACCEPT or STOP=BOTH), or either reject or accept H_0 (with STOP=BOTH)
- expected cumulative stopping probabilities at each stage

For a one-sided design, the expected cumulative stopping probabilities under the hypothetical references $\theta = c_i\theta_1$ are displayed.

For a two-sided design, the expected cumulative stopping probabilities under each of the hypothetical references $\theta = c_i\theta_{1l}$ and $\theta = c_i\theta_{1u}$ are displayed, where θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively.

Note that for a symmetric two-sided design, only the expected cumulative stopping probabilities under the hypothetical references $\theta = c_i\theta_{1u}$ are derived.

The expected stopping stage is given by $k_0 + d$ and is derived from the expected information level

$$\sum_{k=1}^{K} p_k\, I_k = I_{k_0} + d\,(I_{(k_0+1)} - I_{k_0})$$

where p_k is the stopping probability at stage k and $0 \leq d < 1$.

For equally spaced information levels, the expected stopping stage is reduced to the weighted average

$$\sum_{k=1}^{K} p_k\, k$$

Test Information

The "Test Information" table displays the following information at each stage:

- proportion of information
- actual information level, if the maximum information is available from the input BOUNDARY= data set
- alternative references with the specified statistic scale. If a p-value scale is specified, the standardized Z scale is used.
- boundary values with the specified statistic scale to reject or accept the null hypothesis

Note that implicitly, the test information table also contains variables for the boundary scale, stopping criterion, and type of alternative hypothesis. That is, if an ODS statement is used to save the table, the data set also contains the variables _Scale_ for the boundary scale, _Stop_ for the stopping criterion, and _ALT_ for the type of alternative hypothesis.

If the test variable is specified, the table also displays the following:

- test statistic
- resulting action of test statistic: continue to the next stage, accept the null hypothesis H_0, or reject H_0

ODS Table Names

PROC SEQTEST assigns a name to each table it creates. You must use these names to reference tables when using the Output Delivery System (ODS). These names are listed in Table 78.3. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 78.3 ODS Tables Produced by PROC SEQTEST

ODS Table Name	Description	Option
CondPower	Conditional power	CONDPOWER
Design	Design information	
ErrSpend	Error spending	ERRSPEND
ParameterEstimates	Parameter estimates	DATA(TESTVAR=) or PARMS(TESTVAR=)
PowerSampleSize	Power and expected sample sizes	PSS
PredPower	Predictive power	PREDPOWER
RepeatedCI	Repeated confidence intervals	RCI
StopProb	Stopping probabilities	STOPPROB
Test	Test statistics and boundary values	

Graphics Output

This section describes the use of ODS for creating graphics with the SEQTEST procedure. To request these graphs, you must specify the ODS GRAPHICS ON statement in addition to the associated graphics options in the PROC SEQTEST statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Sequential ASN Plot

The PLOTS=ASN option displays the average sample numbers (expected sample sizes for nonsurvival data or expected numbers of events for survival data) under various hypothetical references. The average sample numbers are connected for each design, and these connected curves for all designs are displayed in the "Sequential ASN Plot" graph.

For a one-sided design, average sample numbers under the hypothetical references $\theta = c_i\theta_1$ are displayed, where c_i are the values specified in the CREF= option and θ_1 is the alternative reference. The horizontal axis displays the c_i values of these hypothetical references.

For a two-sided design, average sample numbers under each of the hypothetical references $\theta = c_i\theta_{1l}$ and $\theta = c_i\theta_{1u}$ are displayed, where θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively. The horizontal axis displays $-c_i$ values for lower hypothetical references $\theta = c_i\theta_{1l}$ and c_i values for upper hypothetical references $\theta = c_i\theta_{1u}$. Note that for a symmetric two-sided design, only average sample numbers under the hypothetical references $\theta = c_i\theta_{1u}$ are derived.

If the trial stops after the sequential test, the hypothetical reference corresponding to the test statistic is also indicated in the plot.

Conditional Power Plot

The PLOTS=CONDPOWER option displays the conditional powers given the observed statistic under various hypothetical references. These powers are connected and are displayed in the "Conditional Power Plot" graph.

For a one-sided test, the power are derived under the hypothetical references $\theta = \hat{\theta}$ and $\theta = c_i\theta_1$, where $\hat{\theta}$ is the observed statistic, θ_1 is the alternative reference, and c_i are the values specified in the CREF= option. The horizontal axis displays these c_i values for hypothetical references.

For a two-sided test, the power are derived under hypothetical references $\theta = \hat{\theta}$, $\theta = c_i\theta_{1l}$, and $\theta = c_i\theta_{1u}$, where θ_{1l} is the lower alternative reference and θ_{1u} is the upper alternative reference. The horizontal axis displays $-c_i$ values for hypothetical references $\theta = c_i\theta_{1l}$ and c_i values for hypothetical references $\theta = c_i\theta_{1u}$.

If the trial stops after the sequential test, the hypothetical reference corresponding to the test statistic is also indicated in the plot.

Sequential Error Spending Plot

The PLOTS=ERRSPEND option displays the cumulative error spending at each stage on each boundary in the "Sequential Error Spending Plot" graph. A legend table uses the design labels to identify the curves for the corresponding design in the plot.

Sequential Power Plot

The PLOTS=POWER option displays the powers under various hypothetical references. The powers are connected for each design, and these connected curves for all designs are displayed in the "Sequential Power Plot" graph.

For a one-sided design, powers under hypothetical references $\theta = c_i\theta_1$ are displayed, where c_i are the values specified in the CREF= option and θ_1 is the alternative reference. The horizontal axis displays the c_i values of these hypothetical references.

For a two-sided design, powers under hypothetical references $\theta = c_i\theta_{1l}$ and $\theta = c_i\theta_{1u}$ are displayed, where θ_{1l} and θ_{1u} are the lower and upper alternative references, respectively. The horizontal axis displays $-c_i$ values for lower hypothetical references $\theta = c_i\theta_{1l}$ and c_i values for upper hypothetical references $\theta = c_i\theta_{1u}$. Note that for a symmetric two-sided design, only powers under hypothetical references $\theta = c_i\theta_{1u}$ are derived.

If the trial stops after the sequential test, the hypothetical reference corresponding to the test statistic is also indicated in the plot.

Repeated Confidence Intervals Plot

The PLOTS=RCI option displays repeated confidence intervals at each stage given the observed statistic at that stage. These repeated confidence intervals are displayed in the "Repeated Confidence Intervals Plot" graph.

Sequential Test Plot

The PLOTS=TEST option displays boundary values and test statistics in the "Test Plot" graph. The boundary values are connected for each boundary, and both the stage number and the information level at each stage are displayed. The legend table identifies the acceptance and rejection regions in the plot.

ODS Graphics

PROC SEQTEST assigns a name to each graph it creates. You can use these names to reference the graphs when using ODS. The names are listed in Table 78.4.

To request these graphs, you must specify the ODS GRAPHICS ON statement in addition to the options indicated in Table 78.4. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Table 78.4 ODS Graphics Produced by PROC SEQTEST

ODS Graph Name	Plot Description	Option
AsnPlot	Average sample numbers	PLOTS=ASN
CondPowerPlot	Conditional power curves	PLOTS=CONDPOWER
ErrSpendPlot	Error spending	PLOTS=ERRSPEND
PowerPlot	Power curves	PLOTS=POWER
RepeatedCIPlot	Repeated confidence intervals	PLOTS=RCI
TestPlot	Boundary values and test statistics	PLOTS=TEST

Examples: SEQTEST Procedure

The following examples perform group sequential tests with various designs and test statistics.

Four statistic scales are available for the input boundary values, the input test statistic, and the displayed test information in the SEQTEST procedure. These are the maximum likelihood estimator scale, score statistic scale, standardized normal Z scale, and p-value scale. There is a unique one-to-one transformation between any two of the scales, and you can use different scales for the input boundary values and input test statistic. These boundary values and test statistic are displayed with

the scale specified in the BOUNDARYSCALE= option in the SEQTEST procedure.

Example 78.1: Testing the Difference between Two Proportions

This example demonstrates group sequential tests that use an O'Brien-Fleming group sequential design. A clinic is studying the effect of vitamin C supplements in treating flu symptoms. The study consists of patients in the clinic who have exhibited the first sign of flu symptoms within the last 24 hours. These patients are randomly assigned to either the control group (which receives placebo pills) or the treatment group (which receives large doses of vitamin C supplements). At the end of a five-day period, the flu symptoms of each patient are recorded.

Suppose that you know from past experience that flu symptoms disappear in five days for 60% of patients who experience flu symptoms. The clinic would like to detect a 75% symptom disappearance with a high probability. A test that compares the proportions directly specifies the null hypothesis $H_0 : \theta = p_t - p_c = 0$ with a one-sided alternative $H_1 : \theta > 0$ and a power of 0.90 at $H_1 : \theta = 0.15$, where p_t and p_c are the proportions of symptom disappearance in the treatment group and control group, respectively.

The following statements invoke the SEQDESIGN procedure and request a four-stage group sequential design by using an O'Brien-Fleming method for normally distributed data. The design uses a one-sided alternative hypothesis with early stopping either to accept or reject the null hypothesis H_0. The BOUNDARYSCALE=MLE option uses the MLE scale to display statistics in the boundary table and boundary plots.

```
ods graphics on;
proc seqdesign altref=0.15
               boundaryscale=mle
               ;
   OBrienFleming: design method=obf
                         nstages=4
                         alt=upper
                         stop=both
                         alpha=0.025
                         ;
   samplesize model=twosamplefreq(nullprop=0.6 test=prop);
ods output Boundary=Bnd_Count;
run;
ods graphics off;
```

The ODS OUTPUT statement with the BOUNDARY=BND_COUNT option creates an output data set named BND_COUNT which contains the resulting boundary information for the subsequent sequential tests.

The "Design Information" table in Output 78.1.1 displays design specifications. With the specified alternative hypothesis $H_1 : \theta = 0.15$, the maximum information is derived to achieve a power of 0.90 at H_1. The derived fixed-sample information ratio 1.0767 is the maximum information needed for a group sequential design relative to its corresponding fixed-sample design.

Output 78.1.1 O'Brien-Fleming Design Information

The SEQDESIGN Procedure
Design: OBrienFleming

Design Information

Statistic Distribution	Normal
Boundary Scale	MLE
Alternative Hypothesis	Upper
Early Stop	Accept/Reject Null
Method	O'Brien-Fleming
Boundary Key	Both
Alternative Reference	0.15
Number of Stages	4
Alpha	0.025
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	107.6741
Max Information	502.8343
Null Ref ASN (Percent of Fixed Sample)	61.12891
Alt Ref ASN (Percent of Fixed Sample)	75.89782

The "Boundary Information" table in Output 78.1.2 displays the information level, alternative reference, and boundary values at each stage. With the BOUNDARYSCALE=MLE option, the SEQDESIGN procedure displays the output boundaries with the maximum likelihood estimator scale.

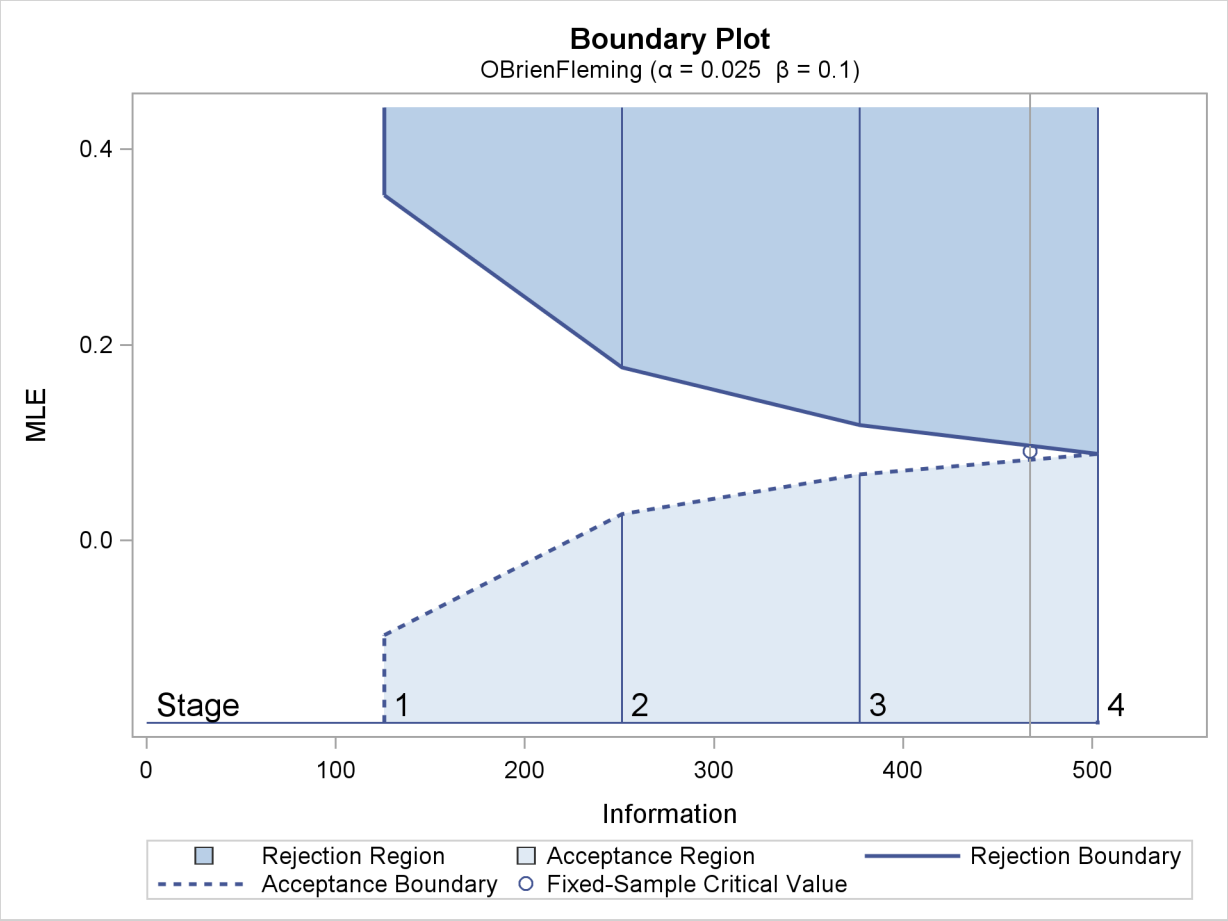
Output 78.1.2 O'Brien-Fleming Boundary Information

Boundary Information (MLE Scale)
Null Reference = 0

	Information Level			Alternative Reference	Boundary Values	
					Upper	
Stage	Proportion	Actual	N	Upper	Beta	Alpha
1	0.2500	125.7086	107.4808	0.15000	-0.09709	0.35291
2	0.5000	251.4171	214.9617	0.15000	0.02645	0.17645
3	0.7500	377.1257	322.4425	0.15000	0.06764	0.11764
4	1.0000	502.8343	429.9233	0.15000	0.08823	0.08823

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 78.1.3. The horizontal axis indicates the information levels for the design. The stages are indicated by vertical lines with accompanying stage numbers. If the test statistic at a stage is in a rejection region, the trial stops and the hypothesis is rejected. If the test statistic is in an acceptance region, then the trial also stops and the hypothesis is accepted. If the statistic is not in a rejection or an acceptance region, the trial continues to the next stage.

Output 78.1.3 O'Brien-Fleming Boundary Plot



The boundary plot also displays the information level and critical value for the corresponding fixed-sample design. The solid and dashed lines at the fixed-sample information level correspond to the rejection and acceptance lines, respectively.

With the SAMPLESIZE statement, the maximum information is used to derive the required sample size for the study. The "Sample Size Summary" table in Output 78.1.4 displays parameters for the sample size computation.

Output 78.1.4 Required Sample Size Summary

Sample Size Summary

Test	Two-Sample Proportions
Null Proportion	0.6
Proportion (Group A)	0.75
Test Statistic	Z for Proportion
Reference Proportions	Alt Ref
Max Sample Size	429.9233
Expected Sample Size (Null Ref)	244.0768
Expected Sample Size (Alt Ref)	303.0464

With the derived maximum information and the specified MODEL= option in the SAMPLESIZE statement, the total sample size in each group for testing the difference between two proportions under the alternative hypothesis is

$$N_1 = N_2 = \left(p_{1c}\left(1 - p_{1c}\right) + p_{1t}\left(1 - p_{1t}\right)\right) I_X$$

where $p_{1c} = 0.6$ and $p_{1t} = p_{1c} + \theta_1 = 0.75$. By default (or equivalently, if you specify REF=PROP), the required sample sizes are computed under the alternative hypothesis. See the section "Test for the Difference between Two Binomial Proportions" in the chapter "The SEQDESIGN Procedure" for a description of these parameters.

The "Sample Sizes (N)" table in Output 78.1.5 displays the required sample sizes at each stage, in both fractional and integer numbers. The derived sample sizes under the heading Fractional N which correspond to the design are not integers. These sample sizes are rounded up to integers under the heading Ceiling N. In practice, integer sample sizes are used, and the information levels increase slightly. Thus, 54, 108, 162, and 215 patients are needed in each group for the four stages, respectively.

Output 78.1.5 Required Sample Sizes

Sample Sizes (N)
Two-Sample Z Test for Proportion Difference

	Fractional N			
Stage	N	N(Grp 1)	N(Grp 2)	Information
1	107.48	53.74	53.74	125.7
2	214.96	107.48	107.48	251.4
3	322.44	161.22	161.22	377.1
4	429.92	214.96	214.96	502.8

Sample Sizes (N)
Two-Sample Z Test for Proportion Difference

	Ceiling N			
Stage	N	N(Grp 1)	N(Grp 2)	Information
1	108	54	54	126.3
2	216	108	108	252.6
3	324	162	162	378.9
4	430	215	215	502.9

Suppose the trial follows the study plan, and 54 patients are available in each group at stage 1. The data set count_1 contains these 108 patients. Output 78.1.6 lists the first 10 observations of the data set.

Output 78.1.6 Clinical Trial Data

First 10 Obs in the Trial Data

Obs	Trt	Resp
1	0	0
2	1	1
3	0	1
4	1	0
5	0	0
6	1	0
7	0	0
8	1	1
9	0	1
10	1	0

The Trt variable is a grouping variable with value 0 for a patient in the placebo control group and value 1 for a patient in the treatment group who is given vitamin C supplements. The Resp variable is an indicator variable with value 1 for a patient without flu symptoms after five days and value 0 for a patient with flu symptoms after five days.

The following statements use the GENMOD procedure to estimate the treatment effect at stage 1:

```
proc genmod data=count_1;
   model Resp= Trt;
ods output ParameterEstimates=Parms_Count1;
run;
```

Output 78.1.7 displays the treatment effect at stage 1.

Output 78.1.7 Stage 1 Treatment Difference

The GENMOD Procedure

Analysis Of Maximum Likelihood Parameter Estimates

Parameter	DF	Estimate	Standard Error	Wald 95% Confidence Limits		Wald Chi-Square	Pr > ChiSq
Intercept	1	0.6296	0.0627	0.5066	0.7526	100.68	<.0001
Trt	1	0.1111	0.0887	-0.0628	0.2850	1.57	0.2105
Scale	1	0.4611	0.0314	0.4035	0.5269		

NOTE: The scale parameter was estimated by maximum likelihood.

The test statistic is $\hat{\theta}_1 = \hat{p}_t - \hat{p}_c = 0.1111$, and its associated standard error is

$$\sqrt{\mathrm{Var}(\hat{\theta}_1)} = \sqrt{\frac{\hat{p}_c(1-\hat{p}_c)}{54} + \frac{\hat{p}_t(1-\hat{p}_t)}{54}} = 0.0887$$

The following statements create and display (in Output 78.1.8) the data set that contains the parameter estimate at stage 1, $\hat{\theta}_1 = 0.1111$, and its associated standard error $\sqrt{\text{Var}(\hat{\theta}_1)} = 0.0887$ which are used in the SEQTEST procedure:

```
data Parms_Count1;
   set Parms_Count1;
   if Parameter='Trt';
   _Scale_='MLE';
   _Stage_= 1;
   keep _Scale_ _Stage_ Parameter Estimate StdErr;
run;

proc print data=Parms_Count1;
   title 'Statistics Computed at Stage 1';
run;
```

Output 78.1.8 Statistics Computed at Stage 1

Statistics Computed at Stage 1

Obs	Parameter	Estimate	StdErr	_Scale_	_Stage_
1	Trt	0.1111	0.0887	MLE	1

The initial required sample sizes are derived with the proportions $p_c = 0.6$ and $p_t = 0.75$. If the observed proportions are different from these assumed values, or if the number of available patients is different from the original plan in one of the stages, then the information level that corresponds to the test statistic is estimated from

$$I_k = \frac{1}{\text{Var}(\hat{\theta}_k)}$$

The following statements invoke the SEQTEST procedure and test for early stopping at stage 1:

```
ods graphics on;
proc seqtest Boundary=Bnd_Count
             Parms(Testvar=Trt)=Parms_Count1
             errspendmin=0.001
             boundaryscale=mle
             errspend
             plots=errspend
             ;
ods output Test=Test_Count1;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 1, which was generated in the SEQDESIGN procedure. The PARMS=PARMS_COUNT1 option specifies the input data set PARMS_COUNT1 that contains the test statistic and its associated standard error at stage 1, and the TESTVAR=TRT option identifies the test variable TRT in the data set.

The O'Brien-Fleming design is conservative in early stages and might not be desirable in a clinical trial. The ERRSPENDMIN=0.001 option specifies the minimum error spending at each stage to be 0.001, and it might increase the corresponding nominal p-value in early stages for the trial. The BOUNDARYSCALE=MLE option uses the MLE scale to display test statistics in the boundary table and boundary plots.

The ODS OUTPUT statement with the TEST=TEST_COUNT1 option creates an output data set named TEST_COUNT1 which contains the updated boundary information for the test at stage 1. The data set also provides the boundary information that is needed for the group sequential test at the next stage.

The "Design Information" table in Output 78.1.9 displays design specifications. The derived statistics, such as the overall α and β levels, are derived from the specified maximum information and boundary values in the BOUNDARY= data set. Note that with a minor change in the information level at stage 1, the power also changes slightly from the original design.

Output 78.1.9 Design Information

The SEQTEST Procedure

Design Information

BOUNDARY Data Set	WORK.BND_COUNT
Data Set	WORK.PARMS_COUNT1
Statistic Distribution	Normal
Boundary Scale	MLE
Alternative Hypothesis	Upper
Early Stop	Accept/Reject Null
Number of Stages	4
Alpha	0.025
Beta	0.10154
Power	0.89846
Max Information (Percent of Fixed Sample)	108.2541
Max Information	502.834283
Null Ref ASN (Percent of Fixed Sample)	61.13266
Alt Ref ASN (Percent of Fixed Sample)	73.9393

With the ERRSPEND option, the "Error Spending Information" table in Output 78.1.10 displays cumulative error spending at each stage for each boundary. With the default BOUNDARYKEY=ALPHA and the specified ERRSPENDMIN=0.001 options, the α spending at each stage is at least 0.001.

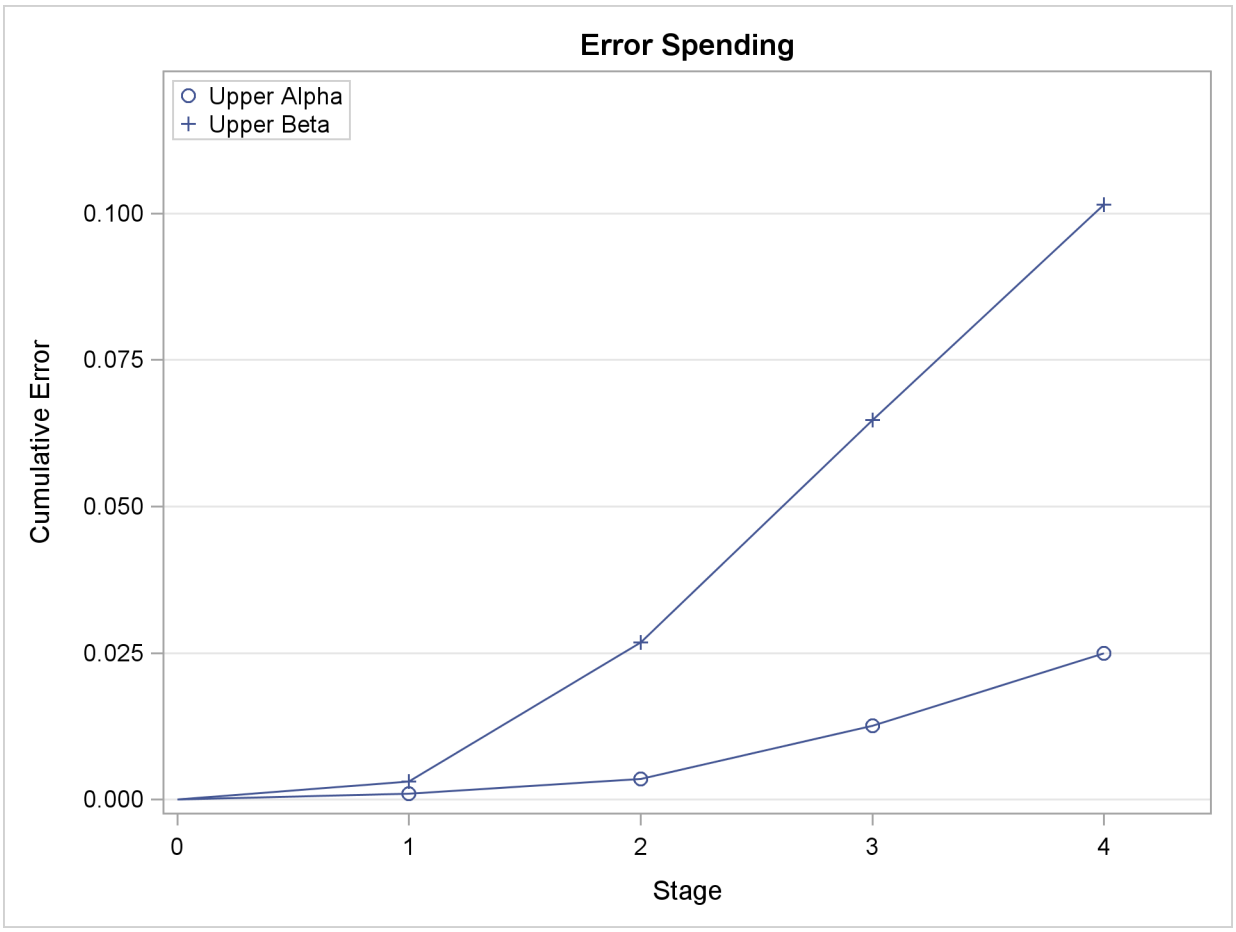
Output 78.1.10 Error Spending Information

Error Spending Information

Stage	---Information Level-- Proportion	Actual	-Cumulative Error Spending- ----------Upper---------- Beta	Alpha
1	0.2525	126.9871	0.00308	0.00100
2	0.5017	252.2695	0.02680	0.00349
3	0.7508	377.5519	0.06472	0.01258
4	1.0000	502.8343	0.10154	0.02500

With the PLOTS=ERRSPEND option, the procedure displays a plot of error spending for each boundary, as shown in Output 78.1.11. The error spending values in the "Error Spending Information" table in Output 78.1.10 are displayed in the plot.

Output 78.1.11 Error Spending Plot



The "Test Information" table in Output 78.1.12 displays the boundary values for the design, test statistic, and resulting action at each stage. With the BOUNDARYSCALE=MLE option, the maximum likelihood estimator scale is used for the test statistic and boundary values. The table shows

that the test statistic 0.1111 is between the upper α and β boundaries, so the trial continues to the next stage.

Output 78.1.12 Sequential Test

Test Information (MLE Scale)
Null Reference = 0

Stage	Information Level Proportion	Information Level Actual	Alternative Reference Upper	Boundary Values Upper Beta	Boundary Values Upper Alpha
1	0.2525	126.9871	0.15000	-0.09304	0.27423
2	0.5017	252.2695	0.15000	0.02724	0.17450
3	0.7508	377.5519	0.15000	0.06813	0.11782
4	1.0000	502.8343	0.15000	0.08876	0.08876

Test Information (MLE Scale)
Null Reference = 0

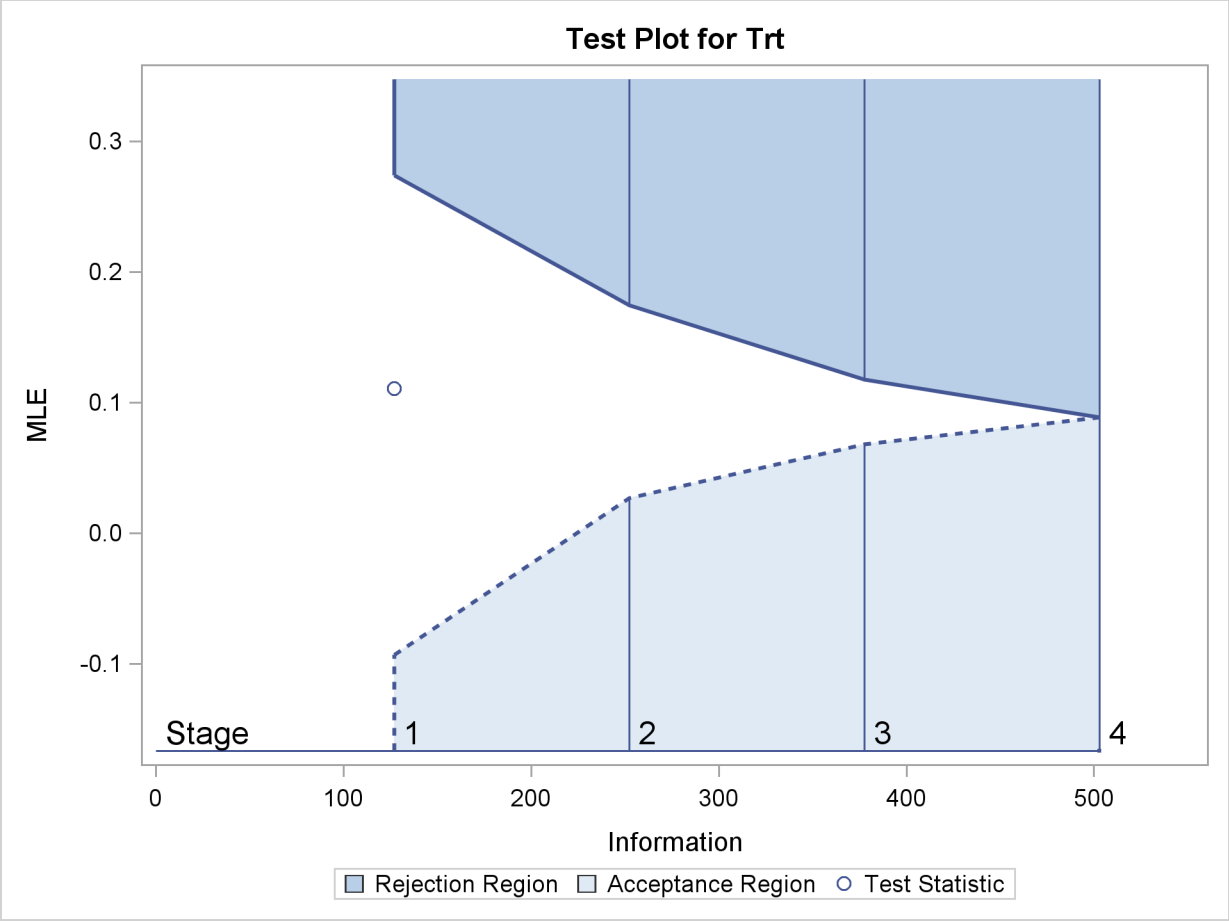
Stage	Test Trt Estimate	Test Trt Action
1	0.11111	Continue
2	.	
3	.	
4	.	

The information level at stage 1 is derived from the standard error,

$$I_1 = \frac{1}{\mathrm{Var}(\hat{\theta}_1)} = \frac{1}{(\mathrm{s.e.}(\hat{\theta}_1))^2}$$

With the default PLOTS=TEST option, the "Test Plot" graph displays boundary values of the design and the test statistic at stage 1, as shown in Output 78.1.13. It also shows that the observed statistic is in the continuation region.

Output 78.1.13 Sequential Test Plot



The observed information level at stage 1, $I_1 = 126.987$, is slightly larger than the target information level at the design. If an observed information level in the study is substantially different from its target level in the design, then the sample sizes should be adjusted in the subsequent stages to achieve the target information levels.

Suppose the trial continues to the next stage, and 108 patients are available in each group at stage 2. The data set COUNT_2 contains these 216 patients.

The following statements use the GENMOD procedure to estimate the treatment effect at stage 2:

```
proc genmod data=Count_2;
   model Resp= Trt;
ods output ParameterEstimates=Parms_Count2;
run;
```

The following statements create the parameter estimate at stage 2, $\hat{\theta}_2 = \hat{p}_2 - \hat{p}_1 = 0.1759$, and its associated standard error $\sqrt{\mathrm{Var}(\hat{\theta}_2)} = 0.0623$ into a test data set:

```
data Parms_Count2;
   set Parms_Count2;
   if Parameter='Trt';
```

```
   _Scale_='MLE';
   _Stage_= 2;
   keep _Scale_ _Stage_ Parameter Estimate StdErr;
run;

proc print data=Parms_Count2;
   title 'Statistics Computed at Stage 2';
run;
```

Output 78.1.14 displays the test statistics at stage 2.

Output 78.1.14 Statistics Computed at Stage 2

Statistics Computed at Stage 2

Obs	Parameter	Estimate	StdErr	_Scale_	_Stage_
1	Trt	0.1759	0.0623	MLE	2

The following statements invoke the SEQTEST procedure and test for early stopping at stage 2:

```
ods graphics on;
proc seqtest Boundary=Test_Count1
             Parms(Testvar=Trt)=Parms_Count2
             boundaryscale=mle
             ;
ods output Test=Test_Count2;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 2, which was generated by the SEQTEST procedure at the previous stage. The PARMS= option specifies the input data set that contains the test statistic and its associated standard error at stage 2, and the TESTVAR= option identifies the test variable in the data set.

The ODS OUTPUT statement with the TEST=TEST_COUNT2 option creates an output data set named TEST_COUNT2 which contains the updated boundary information for the test at stage 2. The data set also provides the boundary information that is needed for the group sequential test at the next stage.

The "Test Information" table in Output 78.1.15 displays the boundary values, test statistic, and resulting action at each stage. The table shows that the test statistic 0.17593 is larger than the corresponding upper alpha boundary value, so the trial stops to reject the hypothesis.

Output 78.1.15 Sequential Test

The SEQTEST Procedure

Test Information (MLE Scale)
Null Reference = 0

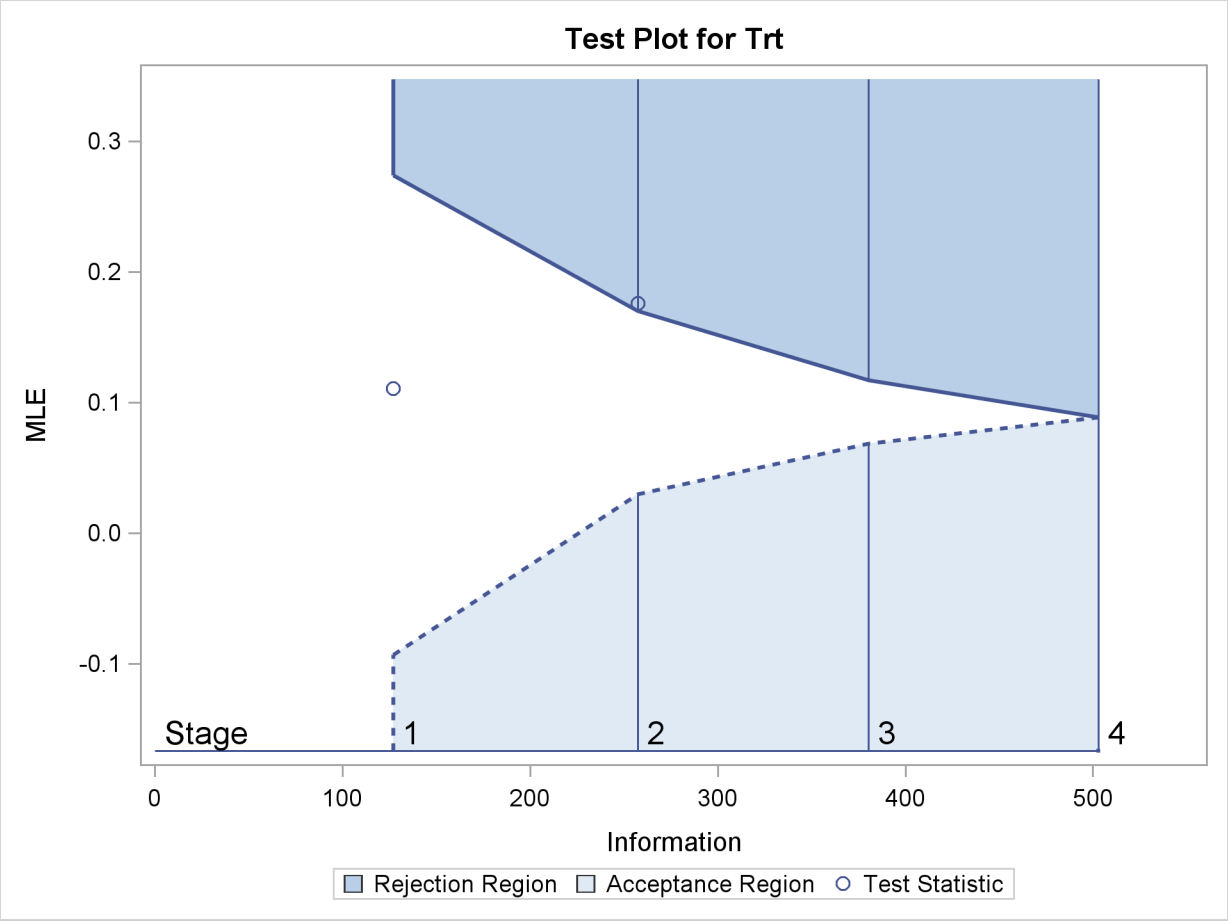
Stage	Information Level-- Proportion	Actual	-Alternative- --Reference-- Upper	Boundary Values---- ----------Upper--------- Beta	Alpha
1	0.2525	126.9871	0.15000	-0.09304	0.27423
2	0.5122	257.5571	0.15000	0.03024	0.17001
3	0.7561	380.1957	0.15000	0.06863	0.11722
4	1.0000	502.8343	0.15000	0.08882	0.08882

Test Information (MLE Scale)
Null Reference = 0

Stage	----------Test---------- ----------Trt---------- Estimate	Action
1	0.11111	Continue
2	0.17593	Reject Null
3	.	
4	.	

With the specified ODS GRAPHICS ON statement, the "Test Plot" is displayed by default, as shown in Output 78.1.16. The plot displays boundary values of the design and the test statistics at the first two stages. As expected, the test statistic at stage 2 is in the "Upper Rejection Region" above the upper alpha boundary.

Output 78.1.16 Sequential Test Plot



After a trial is stopped, the "Parameter Estimates" table in Output 78.1.17 displays the stopping stage and the maximum likelihood estimate of the parameter. It also displays the p-value, median estimate, and confidence limits for the parameter that correspond to the observed statistic by using the specified sample space ordering.

Output 78.1.17 Parameter Estimates

Parameter Estimates
Stagewise Ordering

Parameter	Stopping Stage	MLE	p-Value for H0:Parm=0	Median Estimate	Lower 95% CL
Trt	2	0.175926	0.0031	0.174462	0.07059

The MLE statistic at the stopping stage is the maximum likelihood estimate of the parameter and is biased. The computation of p-value, unbiased median estimate, and confidence limits depends on the ordering of the sample space (k, z), where k is the stage number and z is the observed standardized Z statistic. With the default ORDER=STAGEWISE option, the stagewise ordering

that uses counterclockwise ordering around the continuation region is used to compute the p-value, unbiased median estimate, and confidence limits. As expected, the p-value is less than 0.025, and the confidence interval does not contain the null reference zero. With the stagewise ordering, the p-value is computed as

$$P_{\theta=0}\left(Z_1 > a_1\right) + P_{\theta=0}\left(Z_2 > z_2 \mid \_a_1 < Z_1 < a_1\right)$$

where z_2 is the observed standardized Z statistic at stage 2, Z_1 is the standardized normal variate at stage 1, Z_2 is the standardized normal variate at stage 2, and $\_a_1$ and a_1 are the stage 1 lower and upper rejection boundary values, respectively.

See the section "Available Sample Space Orderings in a Sequential Test" on page 6025 for a detailed description of the stagewise ordering.

Example 78.2: Testing an Effect in a Regression Model

This example demonstrates a two-sided group sequential test that uses an error spending design with early stopping to reject the null hypothesis. A study is conducted to examine the effects of Age (years), Weight (kg), RunTime (time in minutes to run 1.5 miles), RunPulse (heart rate while running), and MaxPulse (maximum heart rate recorded while running) on Oxygen (oxygen intake rate, ml per kg body weight per minute). The primary interest is whether oxygen intake rate is associated with weight.

The hypothesis is tested using the following linear model:

$$\text{Oxygen} = \text{Age} + \text{Weight} + \text{RunTime} + \text{RunPulse} + \text{MaxPulse}$$

The null hypothesis is $H_0 : \beta_w = 0$, where β_w is the regression parameter for the variable Weight. Suppose that $\beta_w = 0.10$ is the reference improvement that should be detected at a 0.90 level. Then the maximum information I_X can be derived in the SEQDESIGN procedure.

Following the derivations in the section "Test for a Parameter in the Regression Model" in the chapter "The SEQDESIGN Procedure," the required sample size can be derived from

$$N = I_X \frac{\sigma_y^2}{\left(1 - r_x^2\right)\sigma_x^2}$$

where σ_y^2 is the variance of the response variable in the regression model, r_x^2 is the proportion of variance of Weight explained by other covariates, and σ_x^2 is the variance of Weight.

Further suppose that from past experience, $\sigma_y^2 = 5$, $r_x^2 = 0.10$, and $\sigma_x^2 = 64$. Then the required sample size can be derived using the SAMPLESIZE statement in the SEQDESIGN procedure.

The following statements invoke the SEQDESIGN procedure and request a three-stage group sequential design for normally distributed data to test the null hypothesis of a regression parameter $H_0 : \beta_w = 0$ against the alternative $H_1 : \beta_w \neq 0$:

```
ods graphics on;
proc seqdesign altref=0.10;
   OBFErrorFunction: design method=errfuncobf
                           nstages=3
                           info=cum(2 3 4)
                           ;
   samplesize model=reg(variance=5 xvariance=64 xrsquare=0.10);
ods output Boundary=Bnd_Fit;
run;
ods graphics off;
```

By default, the procedure uses a Type I error probability 0.05 and a Type II error probability 0.10. The ALTREF=0.10 option specifies a power of $1 - \beta = 0.90$ at the alternative hypothesis $H_1 : \beta_w = \pm 0.10$. The INFO=CUM(2 3 4) option specifies that the study perform the first interim analysis with information proportion $2/4 = 0.5$—that is, after half of the total observations are collected.

The ODS OUTPUT statement with the BOUNDARY=BND_FIT option creates an output data set named BND_FIT which contains the resulting boundary information for the subsequent sequential tests.

The "Design Information" table in Output 78.2.1 displays design specifications and derived statistics. Since the alternative reference is specified, the maximum information is derived.

Output 78.2.1 Error Spending Design Information

The SEQDESIGN Procedure
Design: OBFErrorFunction

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Reject Null
Method	Error Spending
Boundary Key	Both
Alternative Reference	0.1
Number of Stages	3
Alpha	0.05
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	101.8276
Max Information	1069.948
Null Ref ASN (Percent of Fixed Sample)	101.2587
Alt Ref ASN (Percent of Fixed Sample)	77.81586

The "Boundary Information" table in Output 78.2.2 displays information level, alternative reference, and boundary values at each stage.

Output 78.2.2 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

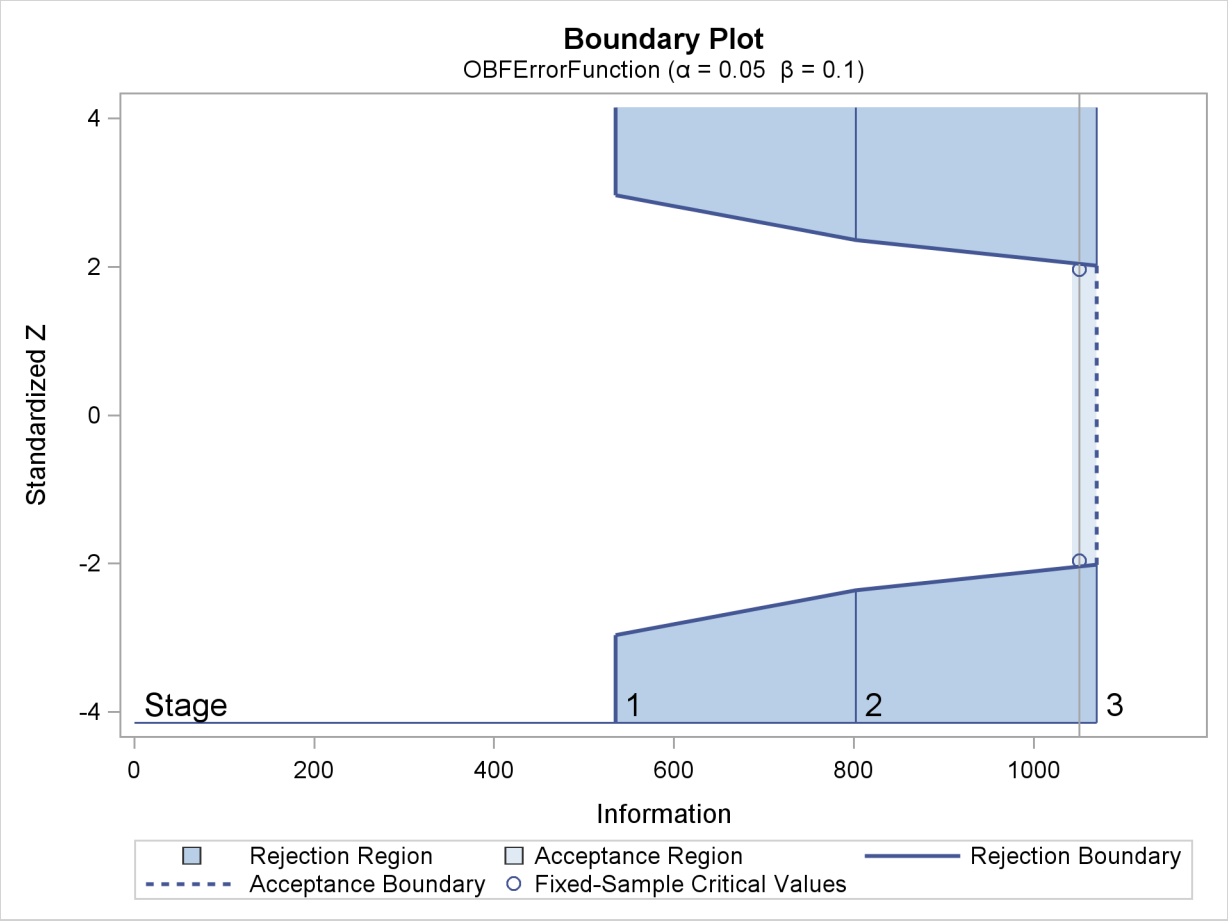
Stage	Information Level Proportion	Information Level Actual	Information Level N	Alternative Reference Lower	Alternative Reference Upper
1	0.5000	534.9738	46.43869	−2.31295	2.31295
2	0.7500	802.4606	69.65804	−2.83277	2.83277
3	1.0000	1069.948	92.87739	−3.27101	3.27101

Boundary Information (Standardized Z Scale)
Null Reference = 0

Stage	Boundary Values Lower Alpha	Boundary Values Upper Alpha
1	−2.96259	2.96259
2	−2.35902	2.35902
3	−2.01409	2.01409

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 78.2.3. The boundary plot also displays the information level and critical value for the corresponding fixed-sample design. This design has characteristics of an O'Brien-Fleming design; the probability for early stopping is low, and the maximum information and critical values at the final stage are similar to those of the corresponding fixed-sample design.

Output 78.2.3 Boundary Plot



With the MODEL=REG option in the SAMPLESIZE statement, the "Sample Size Summary" table in Output 78.2.4 displays the parameters for the sample size computation.

Output 78.2.4 Required Sample Size Summary

Sample Size Summary

Test	Reg Parameter
Parameter	0.1
Variance	5
X Variance	64
R Square (X)	0.1
Max Sample Size	92.87739
Expected Sample Size (Null Ref)	92.35845
Expected Sample Size (Alt Ref)	70.97617

The "Sample Sizes" table in Output 78.2.5 displays the required sample sizes for the group sequential clinical trial.

Output 78.2.5 Required Sample Sizes

Sample Sizes (N)
Z Test for Regression Parameter

Stage	------Fractional N----- N	Information	-------Ceiling N------- N	Information
1	46.44	535.0	47	541.4
2	69.66	802.5	70	806.4
3	92.88	1069.9	93	1071.4

Thus, 47, 70, and 93 individuals are needed in stages 1, 2, and 3, respectively. Since the sample sizes are derived from estimated values of σ_y^2, r_x^2, and σ_x^2, the actual information levels might not achieve the target information levels. Thus, instead of specifying sample sizes in the protocol, you can specify the maximum information levels. Then if an actual information level is much less than the target level, you can increase the sample sizes for the remaining stages to achieve the desired information levels and power.

Suppose that 47 individuals are available at stage 1. Output 78.2.6 lists the first 10 observations of the trial data.

Output 78.2.6 Clinical Trial Data

First 10 Obs in the Trial Data

Obs	Oxygen	Age	Weight	RunTime	Run Pulse	Max Pulse
1	54.5521	44	87.7676	11.6949	178.435	181.607
2	52.2821	40	75.4853	9.8872	184.433	183.667
3	62.1871	44	89.0638	8.7950	155.540	167.108
4	65.3269	42	67.7310	8.4577	162.926	173.877
5	59.9809	37	93.1902	9.3228	179.033	180.144
6	52.5588	47	75.9044	12.0385	177.753	175.033
7	51.7838	40	73.5422	11.6607	175.838	178.140
8	57.0024	43	81.2861	11.2219	160.963	171.770
9	48.0775	44	85.2290	13.1789	173.722	176.548
10	68.3357	38	80.2490	8.5066	171.824	184.011

The following statements use the REG procedure to estimate the slope β_w and its associated standard error at stage 1:

```
proc reg data=Fit_1;
   model Oxygen=Age Weight RunTime RunPulse MaxPulse;
   ods output ParameterEstimates=Parms_Fit1;
run;
```

The following statements create and display (in Output 78.2.7) the input data set that contains slope β_w and its associated standard error for the SEQTEST procedure:

```
data Parms_Fit1;
   set Parms_Fit1;
   if Variable='Weight';
   _Scale_='MLE';
   _Stage_= 1;
   keep _Scale_ _Stage_ Variable Estimate StdErr;
run;

proc print data=Parms_Fit1;
   title 'Statistics Computed at Stage 1';
run;
```

Output 78.2.7 Statistics Computed at Stage 1

Statistics Computed at Stage 1

Obs	Variable	Estimate	StdErr	_Scale_	_Stage_
1	Weight	0.03772	0.04345	MLE	1

The following statements invoke the SEQTEST procedure to test for early stopping at stage 1:

```
ods graphics on;
proc seqtest Boundary=Bnd_Fit
             Parms(Testvar=Weight)=Parms_Fit1
             boundaryadj=errfuncobf
             order=lr
             stopprob
             ;
ods output Test=Test_Fit1;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 1 (this data set was generated in the SEQDESIGN procedure). Recall that these boundary values were derived for the information levels specified with the INFO=CUM(2 3 4) option in the SEQDESIGN procedure. The PARMS=PARMS_FIT1 option specifies the input data set PARMS_FIT1 that contains the test statistic and its associated standard error at stage 1, and the TESTVAR=WEIGHT option identifies the test variable WEIGHT in the data set.

The ORDER=LR option uses the LR ordering to derive the p-value, the unbiased median estimate, and the confidence limits for the regression slope estimate. The boundaries are adjusted using an O'Brien-Fleming-type cumulative error spending function, as specified with the BOUNDARYADJ= option.

The ODS OUTPUT statement with the TEST=TEST_FIT1 option creates an output data set named TEST_FIT1 which contains the updated boundary information for the test at stage 1. The adjustment is needed because the observed information level is different from the information level in the BOUNDARY= data set. The data set TEST_FIT1 also provides the boundary information that is needed for the group sequential test at the next stage.

The "Design Information" table in Output 78.2.8 displays the design specifications. By default, the boundary values are modified for the new information levels to maintain the Type I α level. The maximum information remains the same as in the BOUNDARY= data set, but the derived Type II error probability β and power $1-\beta$ are different because of the new information level.

Output 78.2.8 Design Information

The SEQTEST Procedure

Design Information

BOUNDARY Data Set	WORK.BND_FIT
Data Set	WORK.PARMS_FIT1
Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Reject Null
Number of Stages	3
Alpha	0.05
Beta	0.09994
Power	0.90006
Max Information (Percent of Fixed Sample)	101.8057
Max Information	1069.94751
Null Ref ASN (Percent of Fixed Sample)	101.2416
Alt Ref ASN (Percent of Fixed Sample)	77.87607

With the STOPPROB option, the "Expected Cumulative Stopping Probabilities" table in Output 78.2.9 displays the expected stopping stage and cumulative stopping probability to reject the null hypothesis at each stage under various hypothetical references $\theta = c_i\theta_1$, where θ_1 is the alternative reference and $c_i = 0, 0.5, 1, 1.5$ by default. You can specify other values for c_i with the CREF= option.

Output 78.2.9 Stopping Probabilities

Expected Cumulative Stopping Probabilities
Reference = CRef * (Alt Reference)

CRef	Expected Stopping Stage	Source	----Stopping Probabilities--- Stage_1	Stage_2	Stage_3
0.0000	2.978	Reject Null	0.00289	0.01906	0.05000
0.5000	2.792	Reject Null	0.03373	0.17443	0.36566
1.0000	2.069	Reject Null	0.24884	0.68206	0.90006
1.5000	1.348	Reject Null	0.68172	0.97032	0.99820

The "Test Information" table in Output 78.2.10 displays the boundary values for the test statistic with the default standardized Z scale. The information level at stage 1 is derived from the standard error s_1 in the PARMS= data set,

$$I_1 = \frac{1}{s_1^2} = \frac{1}{0.043453^2} = 529.62$$

Output 78.2.10 Sequential Tests

Test Information (Standardized Z Scale)
Null Reference = 0

Stage	--Information Level- Proportion	Actual	------Alternative----- -------Reference------ Lower	Upper	----Boundary Values--- ---Lower-- Alpha	---Upper-- Alpha
1	0.4950	529.6232	-2.30135	2.30135	-2.97951	2.97951
2	0.7475	799.7853	-2.82805	2.82805	-2.36291	2.36291
3	1.0000	1069.948	-3.27101	3.27101	-2.01336	2.01336

Test Information (Standardized Z Scale)
Null Reference = 0

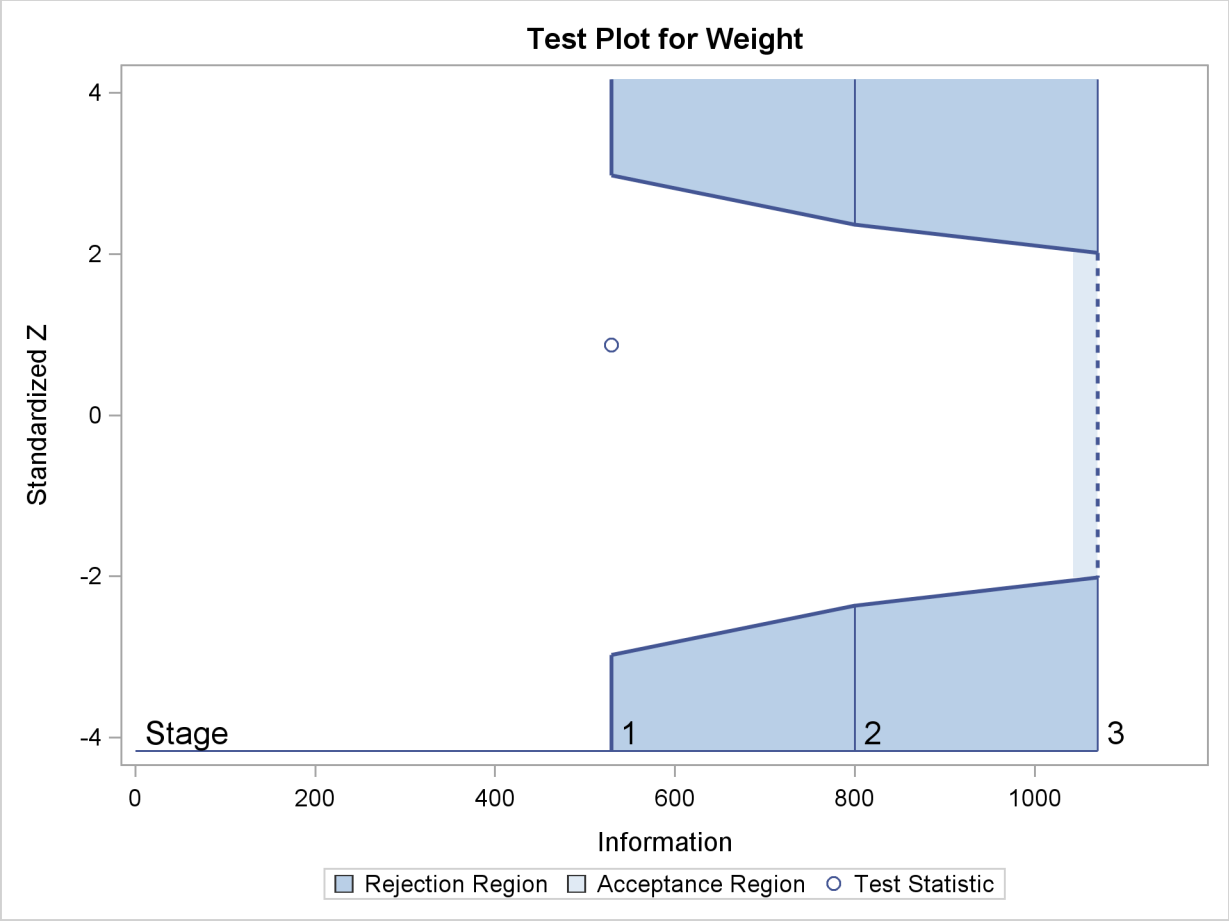
Stage	----------Test---------- ----------Weight--------- Estimate	Action
1	0.86798	Continue
2	.	
3	.	

The information levels at stages 2 and 3 are derived proportionally from the corresponding information levels in the BOUNDARY= data set. See the section "Boundary Adjustments for Information Levels" on page 6017 for details about how these information levels are computed.

At stage 1, the standardized Z statistic 0.86798 is between the lower α boundary -2.97951 and the upper α boundary 2.97951, so the trial continues to the next stage.

With the specified ODS GRAPHICS ON statement, a boundary plot with test statistics is displayed by default, as shown in Output 78.2.11. As expected, the test statistic is in the continuation region between the lower and upper α boundaries.

Output 78.2.11 Sequential Test Plot



The following statements use the REG procedure to estimate the slope β_w and its associated standard error at stage 2:

```
proc reg data=Fit_2;
   model Oxygen=Age Weight RunTime RunPulse MaxPulse;
   ods output ParameterEstimates=Parms_Fit2;
run;
```

Note that the data set Fit_2 contains both the data from stage 1 and the data from stage 2.

The following statements create and display (in Output 78.2.12) the input data set that contains slope β_w and its associated standard error at stage 2 for the SEQTEST procedure:

```
data Parms_Fit2;
   set Parms_Fit2;
   if Variable='Weight';
   _Scale_='MLE';
   _Stage_= 2;
   keep _Scale_ _Stage_ Variable Estimate StdErr;
run;
```

```
proc print data=Parms_Fit2;
   title 'Statistics Computed at Stage 2';
run;
```

Output 78.2.12 Statistics Computed at Stage 2

Statistics Computed at Stage 2

Obs	Variable	Estimate	StdErr	_Scale_	_Stage_
1	Weight	0.02932	0.03520	MLE	2

The following statements invoke the SEQTEST procedure to test for early stopping at stage 2:

```
proc seqtest Boundary=Test_Fit1
             Parms(Testvar=Weight)=Parms_Fit2
             boundaryadj=errfuncobf
             order=lr
             ;
ods output Test=Test_Fit2;
run;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 2, which was generated by the SEQTEST procedure at the previous stage. The PARMS= option specifies the input data set that contains the test statistic and its associated standard error at stage 2, and the TESTVAR= option identifies the test variable in the data set.

The ODS OUTPUT statement with the TEST=TEST_FIT2 option creates an output data set named TEST_FIT2 which contains the updated boundary information for the test at stage 2. The data set also provides the boundary information that is needed for the group sequential test at the next stage.

Since the data set PARMS_FIT2 does not contain the test information at stage 1, the information level at stage 1 in the TEST_FIT1 data set is used to generate boundary values for the test at stage 2.

Following the process at stage 1, the slope estimate is also between its corresponding lower and upper α boundary values, so the trial continues to the next stage.

The following statements use the REG procedure to estimate the slope β_w and its associated standard error at the final stage:

```
proc reg data=Fit_3;
   model Oxygen=Age Weight RunTime RunPulse MaxPulse;
   ods output ParameterEstimates=Parms_Fit3;
run;
```

The following statements create the input data set that contains slope β_w and its associated standard error at stage 3 for the SEQTEST procedure:

```
data Parms_Fit3;
   set Parms_Fit3;
   if Variable='Weight';
   _Scale_='MLE';
   _Stage_= 3;
   keep _Scale_ _Stage_ Variable Estimate StdErr;
run;
```

The following statements print (in Output 78.2.13) the test statistics at stage 3:

```
proc print data=Parms_Fit3;
   title 'Statistics Computed at Stage 3';
run;
```

Output 78.2.13 Statistics Computed at Stage 3

Statistics Computed at Stage 3

Obs	Variable	Estimate	StdErr	_Scale_	_Stage_
1	Weight	0.02189	0.03028	MLE	3

The following statements invoke the SEQTEST procedure to test the hypothesis:

```
ods graphics on;
proc seqtest Boundary=Test_Fit2
             Parms(testvar=Weight)=Parms_Fit3
             boundaryadj=errfuncobf
             order=lr
             pss
             plots=(asn power)
             ;
ods output Test=Test_Fit3;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 3, which was generated by the SEQTEST procedure at the previous stage. The PARMS= option specifies the input data set that contains the test statistic and its associated standard error at stage 3, and the TESTVAR= option identifies the test variable in the data set.

The ODS OUTPUT statement with the TEST=TEST_FIT3 option creates an output data set named TEST_FIT3 which contains the updated boundary information for the test at stage 3.

The "Design Information" table in Output 78.2.14 displays design specifications. By default, the boundary values are modified for the new information levels to maintain the Type I α level.

Output 78.2.14 Design Information

The SEQTEST Procedure

Design Information

BOUNDARY Data Set	WORK.TEST_FIT2
Data Set	WORK.PARMS_FIT3
Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Reject Null
Number of Stages	3
Alpha	0.05
Beta	0.09514
Power	0.90486
Max Information (Percent of Fixed Sample)	102.0102
Max Information	1090.63724
Null Ref ASN (Percent of Fixed Sample)	101.4122
Alt Ref ASN (Percent of Fixed Sample)	77.22139

The maximum information is derived from the standard error associated with the slope estimate at the final stage and is larger than the target level. The derived Type II error probability β and power $1-\beta$ are different because of the new information levels.

With the PSS option, the "Power and Expected Sample Sizes" table in Output 78.2.15 displays powers and expected mean sample sizes under various hypothetical references $\theta = c_i\theta_1$, where θ_1 is the alternative reference and $c_i = 0, 0.5, 1, 1.5$ by default. You can specify the c_i values with the CREF= option.

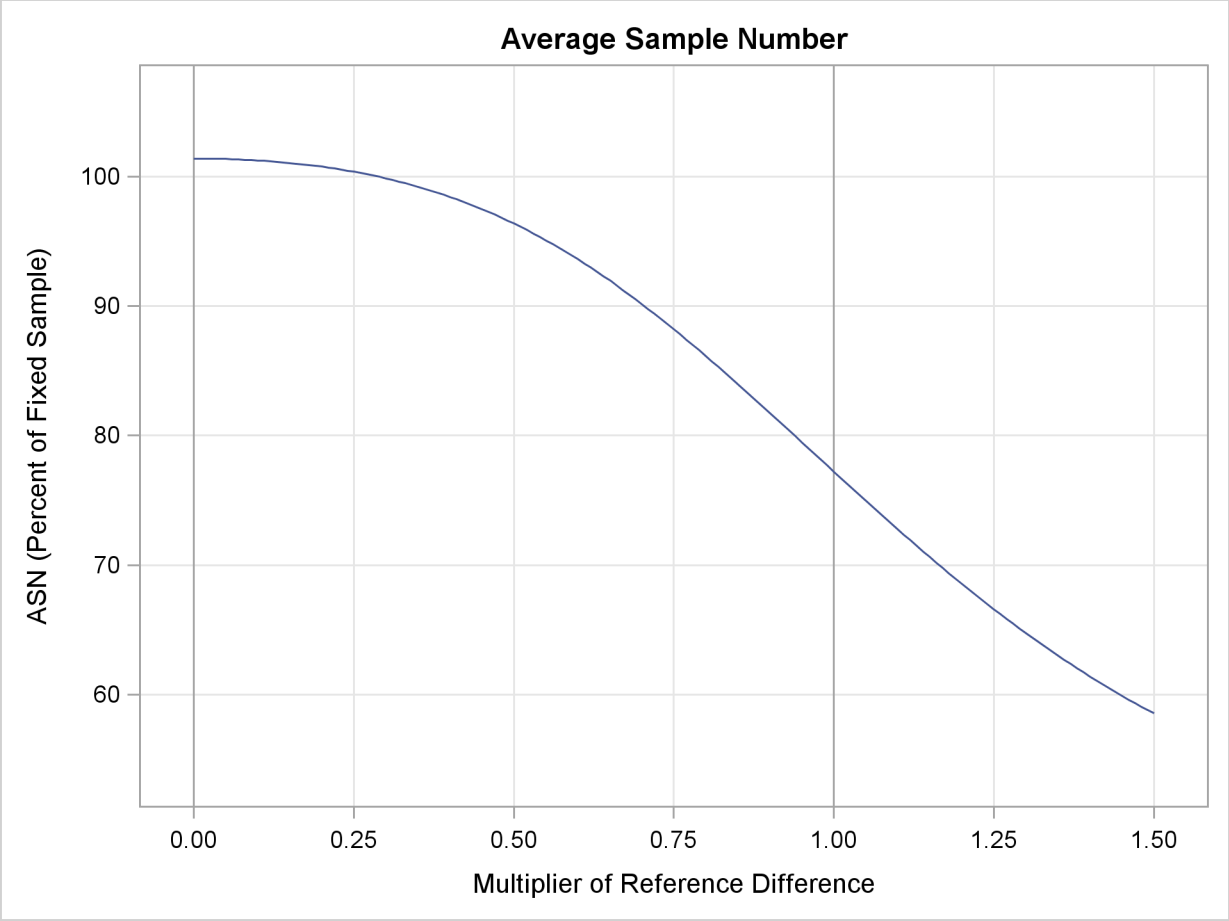
Output 78.2.15 Power and Expected Sample Size Information

Powers and Expected Sample Sizes
Reference = CRef * (Alt Reference)

CRef	Power	-Sample Size- Percent Fixed-Sample
0.0000	0.02500	101.4122
0.5000	0.37046	96.3754
1.0000	0.90486	77.2214
1.5000	0.99844	58.5301

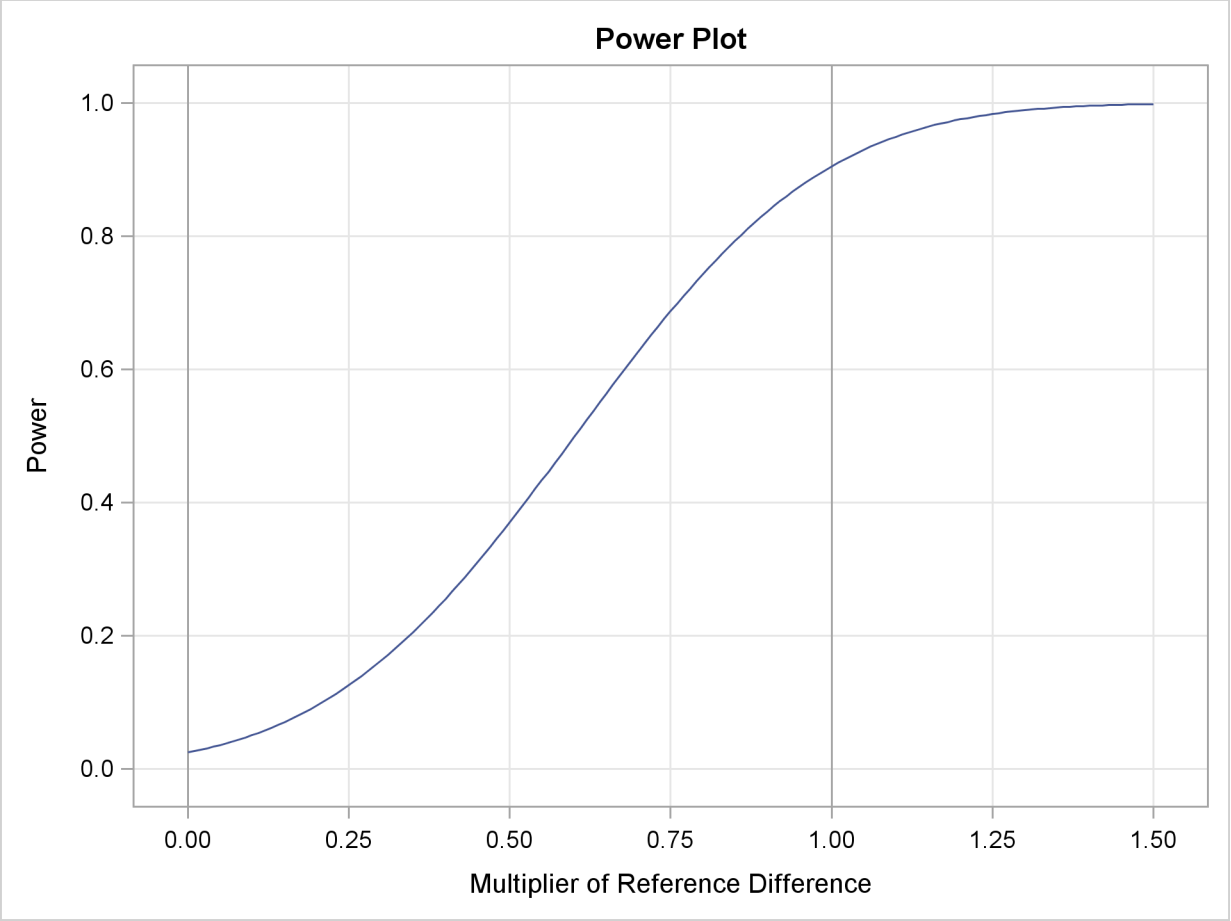
With the PLOTS=ASN option, the procedure displays a plot of expected sample sizes under various hypothetical references, as shown in Output 78.2.16. By default, expected sample sizes under the hypotheses $\theta = c_i\,\theta_1$, $c_i = 0, 0.01, 0.02, \ldots, 1.50$, are displayed, where θ_1 is the alternative reference.

Output 78.2.16 ASN Plot



With the PLOTS=POWER option, the procedure displays a plot of powers under various hypothetical references, as shown in Output 78.2.17. By default, powers under hypothetical references $\theta = c_i\,\theta_1$ are displayed, where $c_i = 0, 0.01, 0.02, \ldots, 1.50$ by default. You can specify c_i values with the CREF= option. The c_i values are displayed on the horizontal axis.

Output 78.2.17 Power Plot



Under the null hypothesis, $c_i = 0$, the power is 0.025, which is the upper Type I error probability. Under the alternative hypothesis, $c_i = 1$, the power is 0.90486, which is one minus the Type II error probability.

The "Test Information" table in Output 78.2.18 displays the boundary values for the test statistic with the default standardized Z scale. The information level at the current stage is derived from the standard error for the current stage in the PARMS= data set. At stage 3, the standardized slope estimate 0.72284 is still between the lower and upper α boundary values. Since it is the final stage, the trial stops to accept the null hypothesis that the variable Weight has no effect on the oxygen intake rate after adjusting for other covariates.

Output 78.2.18 Sequential Tests

Test Information (Standardized Z Scale)
Null Reference = 0

Stage	--Information Level- Proportion	Actual	------Alternative----- -------Reference------ Lower	Upper	----Boundary Values--- ---Lower-- Alpha	---Upper-- Alpha
1	0.4856	529.6232	-2.30135	2.30135	-2.97951	2.97951
2	0.7401	807.1954	-2.84112	2.84112	-2.34945	2.34945
3	1.0000	1090.637	-3.30248	3.30248	-2.01885	2.01885

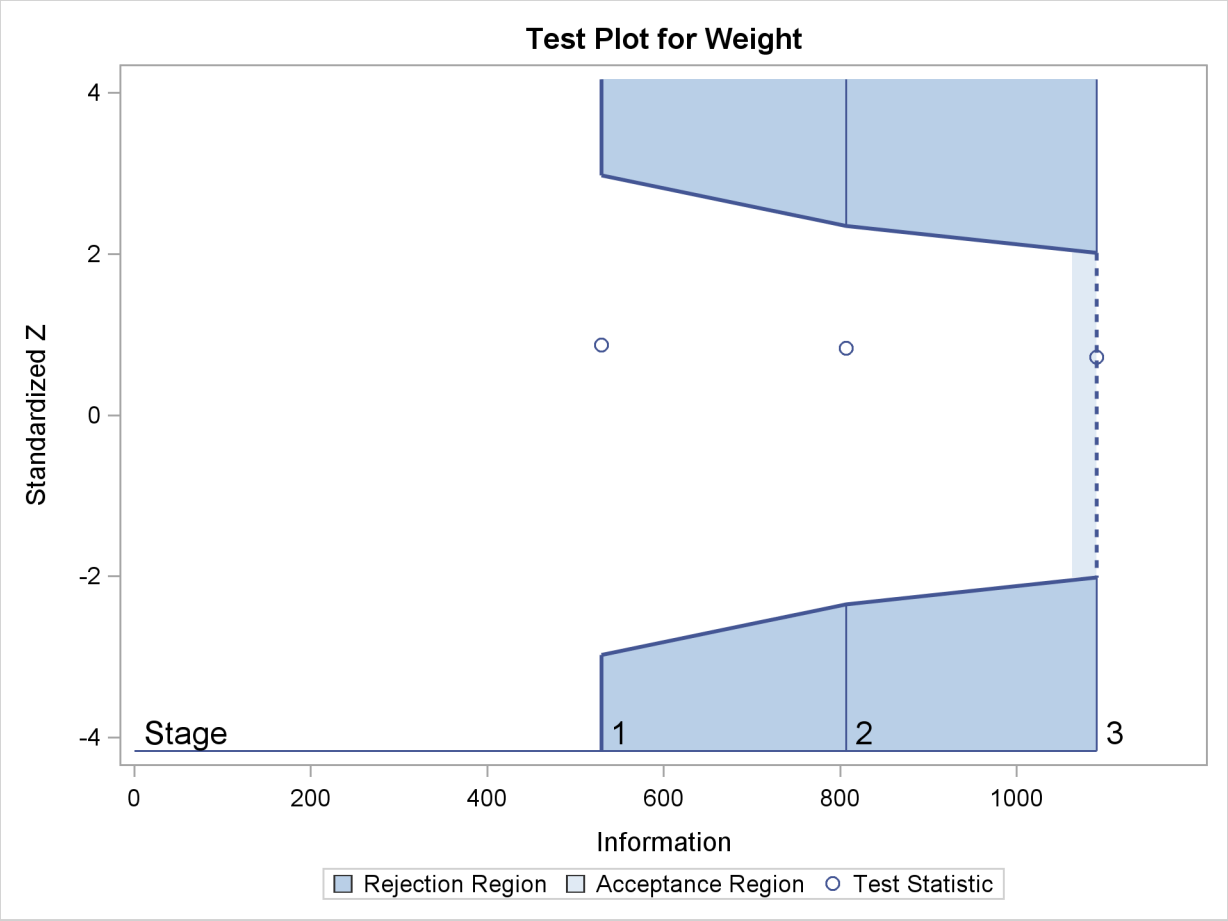
Test Information (Standardized Z Scale)
Null Reference = 0

Stage	----------Test---------- ----------Weight--------- Estimate	Action
1	0.86798	Continue
2	0.83305	Continue
3	0.72284	Accept Null

Since the data set FIT_3 contains the test information only at stage 3, the information levels at previous stages in the TEST_FIT2 data set are used to generate boundary values for the test.

With the specified ODS GRAPHICS ON statement, a boundary plot with test statistics is displayed by default, as shown in Output 78.2.19. As expected, the test statistic is in the acceptance region between the lower and upper α boundaries at the final stage.

Output 78.2.19 Sequential Test Plot



After a trial is stopped, the "Parameter Estimates" table in Output 78.2.20 displays the stopping stage, parameter estimate, unbiased median estimate, confidence limits, and the p-value under the null hypothesis $H_0 : \beta_w = 0$.

Output 78.2.20 Parameter Estimates

Parameter Estimates
LR Ordering

Parameter	Stopping Stage	MLE	p-Value for H0:Parm=0	Median Estimate
Weight	3	0.021888	0.4699	0.021884

Parameter Estimates
LR Ordering

Parameter	95% Confidence Limits	
Weight	-0.03747	0.08123

As expected, the p-value 0.4699 is not significant at the $\alpha = 0.05$ level, and the confidence interval does contain the value zero. The p-value, unbiased median estimate, and confidence limits depend on the ordering of the sample space (k, z), where k is the stage number and z is the standardized Z statistic. With the specified LR ordering, the p-values are computed with the ordering $(k', z') \succ (k, z)$ if $z' > z$. See the section "Available Sample Space Orderings in a Sequential Test" on page 6025 for a detailed description of the LR ordering.

Example 78.3: Testing an Effect with Early Stopping to Accept H_0

This example demonstrates a two-sided group sequential test that uses an error spending design with early stopping to accept the null hypothesis H_0. The example is similar to Example 78.2 but with early stopping to accept H_0.

A study is conducted to examine the effects of Age (years), Weight (kg), RunTime (time in minutes to run 1.5 miles), RunPulse (heart rate while running), and MaxPulse (maximum heart rate recorded while running) on Oxygen (oxygen intake rate, ml per kg body weight per minute). The primary interest is whether oxygen intake rate is associated with weight.

The hypothesis is tested using the following linear model:

$$\text{Oxygen} = \text{Age} + \text{Weight} + \text{RunTime} + \text{RunPulse} + \text{MaxPulse}$$

The null hypothesis is $H_0 : \beta_w = 0$, where β_w is the regression parameter for the variable Weight. Suppose that $\beta_w = 0.10$ is the reference improvement that should be detected at a 0.90 level. Then the maximum information I_X can be derived in the SEQDESIGN procedure.

Following the derivations in the section "Test for a Parameter in the Regression Model" in the chapter "The SEQDESIGN Procedure," the required sample size can be derived from

$$N = I_X \frac{\sigma_y^2}{(1 - r_x^2)\,\sigma_x^2}$$

where σ_y^2 is the variance of the response variable in the regression model, r_x^2 is the proportion of variance of Weight explained by other covariates, and σ_x^2 is the variance of Weight.

Further suppose that from past experience, $\sigma_y^2 = 5$, $r_x^2 = 0.10$, and $\sigma_x^2 = 64$. Then the required sample size can be derived using the SAMPLESIZE statement in the SEQDESIGN procedure.

The following statements invoke the SEQDESIGN procedure and request a three-stage group sequential design for normally distributed data to test the null hypothesis of a regression parameter $H_0 : \beta_w = 0$ against the alternative $H_1 : \beta_w \neq 0$:

```
ods graphics on;
proc seqdesign altref=0.10;
   OBFErrorFunction: design method=errfuncgamma
                            stop=accept
                            nstages=3
                            info=cum(2 3 4)
                            ;
```

```
   samplesize model=reg( variance=5 xvariance=64 xrsquare=0.10);
ods output Boundary=Bnd_Fit;
run;
ods graphics off;
```

By default, the procedure uses a Type I error probability 0.05 and a Type II error probability 0.10. The ALTREF=0.10 option specifies a power of $1 - \beta = 0.90$ at the alternative hypothesis $H_1 : \beta_w = \pm 0.10$. The INFO=CUM(2 3 4) option specifies that the study perform the first interim analysis with information proportion $2/4 = 0.5$—that is, after half of the total observations are collected.

The ODS OUTPUT statement with the BOUNDARY=BND_FIT option creates an output data set named BND_FIT which contains the resulting boundary information for the subsequent sequential tests.

The "Design Information" table in Output 78.3.1 displays design specifications and derived statistics. Since the alternative reference is specified, the maximum information is derived.

Output 78.3.1 Error Spending Design Information

The SEQDESIGN Procedure
Design: OBFErrorFunction

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Accept Null
Method	Error Spending
Boundary Key	Both
Alternative Reference	0.1
Number of Stages	3
Alpha	0.05
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	103.9245
Max Information	1091.972
Null Ref ASN (Percent of Fixed Sample)	75.00521
Alt Ref ASN (Percent of Fixed Sample)	101.8099

The "Boundary Information" table in Output 78.3.2 displays information level, alternative reference, and boundary values at each stage.

Output 78.3.2 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

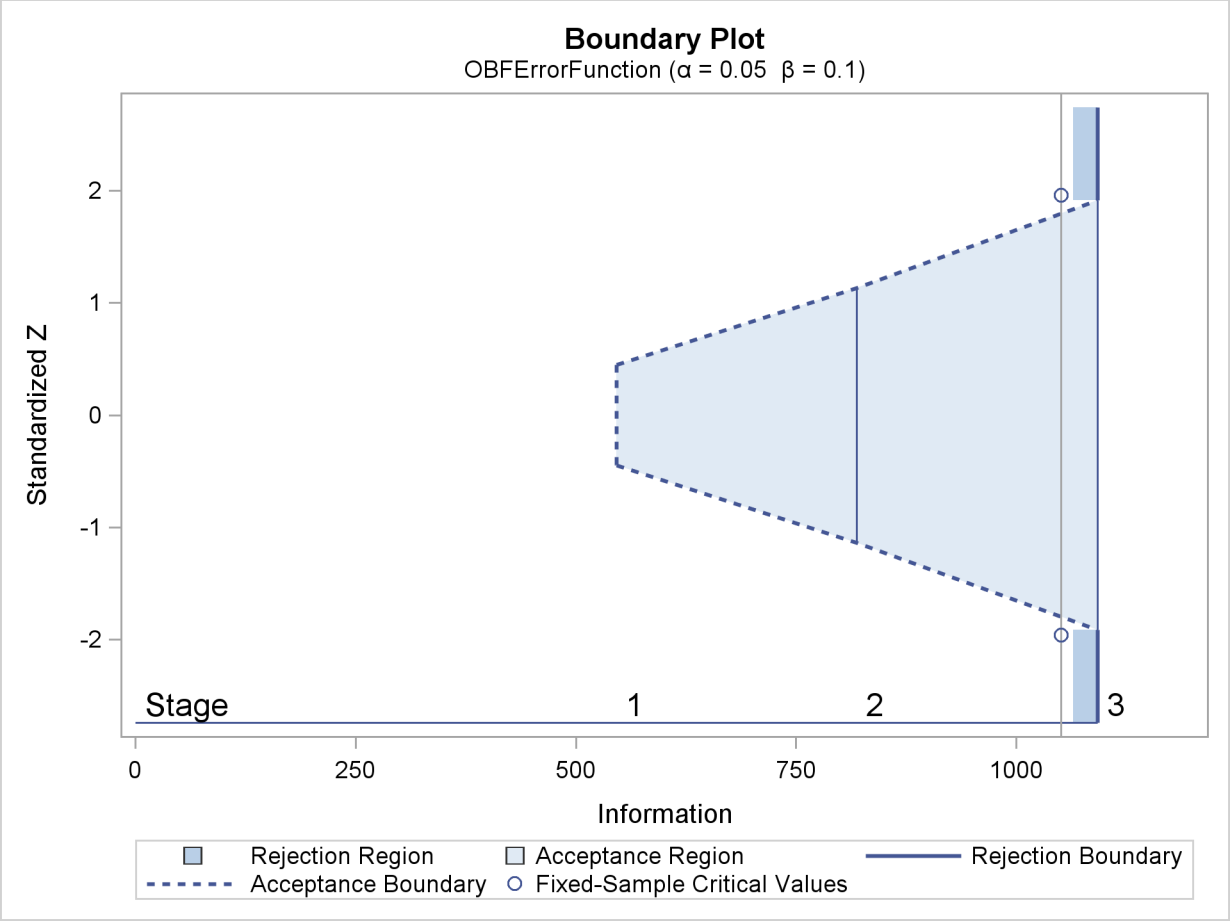
Stage	Information Level Proportion	Information Level Actual	Information Level N	Alternative Reference Lower	Alternative Reference Upper
1	0.5000	545.9862	47.39463	−2.33663	2.33663
2	0.7500	818.9792	71.09195	−2.86178	2.86178
3	1.0000	1091.972	94.78926	−3.30450	3.30450

Boundary Information (Standardized Z Scale)
Null Reference = 0

Stage	Boundary Values Lower Beta	Boundary Values Upper Beta
1	−0.44937	0.44937
2	−1.13583	1.13583
3	−1.91428	1.91428

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 78.3.3. The boundary plot also displays the information level and critical value for the corresponding fixed-sample design.

Output 78.3.3 Boundary Plot



With the MODEL=REG option in the SAMPLESIZE statement, the "Sample Size Summary" table in Output 78.3.4 displays the parameters for the sample size computation.

Output 78.3.4 Required Sample Size Summary

Sample Size Summary

Test	Reg Parameter
Parameter	0.1
Variance	5
X Variance	64
R Square (X)	0.1
Max Sample Size	94.78926
Expected Sample Size (Null Ref)	68.41207
Expected Sample Size (Alt Ref)	92.86057

The "Sample Sizes" table in Output 78.3.5 displays the required sample sizes for the group sequential clinical trial.

Output 78.3.5 Required Sample Sizes

Sample Sizes (N)
Z Test for Regression Parameter

Stage	Fractional N		Ceiling N	
	N	Information	N	Information
1	47.39	546.0	48	553.0
2	71.09	819.0	72	829.4
3	94.79	1092.0	95	1094.4

Thus, 48, 72, and 95 individuals are needed in stages 1, 2, and 3, respectively. Since the sample sizes are derived from estimated values of σ_y^2, r_x^2, and σ_x^2, the actual information levels might not achieve the target information levels. Thus, instead of specifying sample sizes in the protocol, you can specify the maximum information levels. Then if an actual information level is much less than the target level, you can increase the sample sizes for the remaining stages to achieve the desired information levels and power.

Suppose that 48 individuals are available at stage 1. Output 78.3.6 lists the first 10 observations of the trial data.

Output 78.3.6 Clinical Trial Data

First 10 Obs in the Trial Data

Obs	Oxygen	Age	Weight	RunTime	Run Pulse	Max Pulse
1	54.5521	44	87.7676	11.6949	178.435	181.607
2	52.2821	40	75.4853	9.8872	184.433	183.667
3	62.1871	44	89.0638	8.7950	155.540	167.108
4	65.3269	42	67.7310	8.4577	162.926	173.877
5	59.9809	37	93.1902	9.3228	179.033	180.144
6	52.5588	47	75.9044	12.0385	177.753	175.033
7	51.7838	40	73.5422	11.6607	175.838	178.140
8	57.0024	43	81.2861	11.2219	160.963	171.770
9	48.0775	44	85.2290	13.1789	173.722	176.548
10	68.3357	38	80.2490	8.5066	171.824	184.011

The following statements use the REG procedure to estimate the slope β_w and its associated standard error at stage 1:

```
proc reg data=Fit_1;
   model Oxygen=Age Weight RunTime RunPulse MaxPulse;
   ods output ParameterEstimates=Parms_Fit1;
run;
```

The following statements create and display (in Output 78.3.7) the input data set that contains slope β_w and its associated standard error for the SEQTEST procedure:

```
data Parms_Fit1;
   set Parms_Fit1;
   if Variable='Weight';
   _Scale_='MLE';
   _Stage_= 1;
   keep _Scale_ _Stage_ Variable Estimate StdErr;
run;

proc print data=Parms_Fit1;
   title 'Statistics Computed at Stage 1';
run;
```

Output 78.3.7 Statistics Computed at Stage 1

Statistics Computed at Stage 1

Obs	Variable	Estimate	StdErr	_Scale_	_Stage_
1	Weight	0.04660	0.04308	MLE	1

The following statements invoke the SEQTEST procedure to test for early stopping at stage 1:

```
ods graphics on;
proc seqtest Boundary=Bnd_Fit
             Parms(testvar=Weight)=Parms_Fit1
             boundaryadj=errfuncgamma
             stopprob
             order=lr
             ;
ods output Test=Test_Fit1;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 1, which was generated in the SEQDESIGN procedure. The PARMS=PARMS_FIT1 option specifies the input data set PARMS_FIT1 that contains the test statistic and its associated standard error at stage 1, and the TESTVAR=WEIGHT option identifies the test variable WEIGHT in the data set.

The ORDER=LR option uses the LR ordering to derive the p-value, the unbiased median estimate, and the confidence limits for the regression slope estimate. The boundaries are adjusted using gamma cumulative error spending function, as specified with the BOUNDARYADJ= option.

The ODS OUTPUT statement with the TEST=TEST_FIT1 option creates an output data set named TEST_FIT1 which contains the updated boundary information for the test at stage 1. The data set also provides the boundary information that is needed for the group sequential test at the next stage.

The "Design Information" table in Output 78.3.8 displays the design specifications. By default, the boundary values are modified for the new information levels to maintain the Type I α level. The maximum information remains the same as in the BOUNDARY= data set, but the derived Type II error probability β and power $1-\beta$ are different because of the new information level.

Output 78.3.8 Design Information

The SEQTEST Procedure

Design Information

BOUNDARY Data Set	WORK.BND_FIT
Data Set	WORK.PARMS_FIT1
Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Accept Null
Number of Stages	3
Alpha	0.05
Beta	0.10003
Power	0.89997
Max Information (Percent of Fixed Sample)	103.9351
Max Information	1091.97232
Null Ref ASN (Percent of Fixed Sample)	75.11777
Alt Ref ASN (Percent of Fixed Sample)	101.8186

With the STOPPROB option, the "Expected Cumulative Stopping Probabilities" table in Output 78.3.9 displays the expected stopping stage and the cumulative stopping probability of accepting the null hypothesis at each stage under various hypothetical references $\theta = c_i\theta_1$, where θ_1 is the alternative reference and $c_i = 0, 0.5, 1, 1.5$ by default. You can specify other values for c_i with the CREF= option.

Output 78.3.9 Stopping Probabilities

Expected Cumulative Stopping Probabilities
Reference = CRef * (Alt Reference)

CRef	Expected Stopping Stage	Source	----Stopping Probabilities--- Stage_1	Stage_2	Stage_3
0.0000	1.905	Accept Null	0.33294	0.76168	0.95000
0.5000	2.417	Accept Null	0.17674	0.40627	0.62824
1.0000	2.920	Accept Null	0.02635	0.05405	0.10003
1.5000	2.997	Accept Null	0.00109	0.00165	0.00242

The "Test Information" table in Output 78.3.10 displays the boundary values for the test statistic with the default standardized Z scale. The information level at stage 1 is derived from the standard error s_1 in the PARMS= data set,

$$I_1 = \frac{1}{s_1^2} = \frac{1}{0.04308^2} = 538.8$$

Output 78.3.10 Sequential Tests

Test Information (Standardized Z Scale)
Null Reference = 0

Stage	Information Level Proportion	Information Level Actual	Alternative Reference Lower	Alternative Reference Upper
1	0.4934	538.7887	-2.32118	2.32118
2	0.7467	815.3805	-2.85549	2.85549
3	1.0000	1091.972	-3.30450	3.30450

Test Information (Standardized Z Scale)
Null Reference = 0

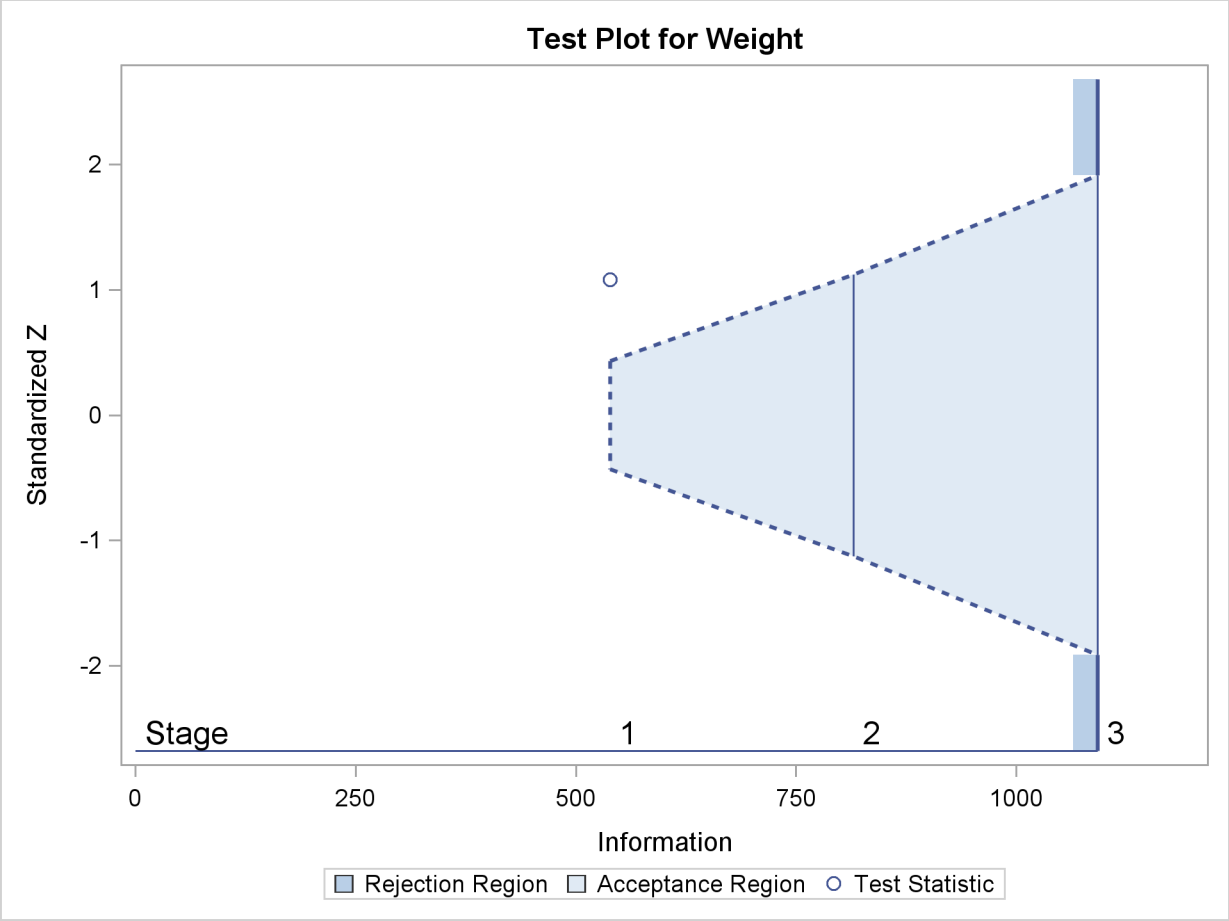
Stage	Boundary Values Lower Beta	Boundary Values Upper Beta	Test Weight Estimate	Test Weight Action
1	-0.43019	0.43019	1.08174	Continue
2	-1.12435	1.12435	.	
3	-1.91475	1.91475	.	

The information levels at stages 2 and 3 are derived proportionally from the corresponding information levels in the BOUNDARY= data set. See the section "Boundary Adjustments for Information Levels" on page 6017 for details about how these information levels are computed.

At stage 1, the standardized Z statistic 1.08174 is greater than the upper β boundary 0.43019, so the trial continues to the next stage.

With the specified ODS GRAPHICS ON statement, a boundary plot with test statistics is displayed by default, as shown in Output 78.3.11. As expected, the test statistic is in the continuation region.

Output 78.3.11 Sequential Test Plot



The following statements use the REG procedure to estimate the slope β_w and its associated standard error at stage 2:

```
proc reg data=Fit_2;
   model Oxygen=Age Weight RunTime RunPulse MaxPulse;
   ods output ParameterEstimates=Parms_Fit2;
run;
```

Note that the data set Fit_2 contains both the data from stage 1 and the data from stage 2,

The following statements create and display (in Output 78.3.12) the input data set that contains slope β_w and its associated standard error at stage 2 for the SEQTEST procedure:

```
data Parms_Fit2;
   set Parms_Fit2;
   if Variable='Weight';
   _Scale_='MLE';
   _Stage_= 2;
   keep _Scale_ _Stage_ Variable Estimate StdErr;
run;

proc print data=Parms_Fit2;
```

```
   title 'Statistics Computed at Stage 2';
run;
```

Output 78.3.12 Statistics Computed at Stage 2

Statistics Computed at Stage 2

Obs	Variable	Estimate	StdErr	_Scale_	_Stage_
1	Weight	0.02925	0.03490	MLE	2

The following statements invoke the SEQTEST procedure to test for early stopping at stage 2:

```
ods graphics on;
proc seqtest Boundary=Test_Fit1
             Parms(testvar=Weight)=Parms_Fit2
             boundaryadj=errfuncgamma
             order=lr
             pss
             plots=(asn power)
             ;
ods output Test=Test_Fit2;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 2, which was generated by the SEQTEST procedure at the previous stage. The PARMS= option specifies the input data set that contains the test statistic and its associated standard error at stage 2, and the TESTVAR= option identifies the test variable in the data set.

Since the data set PARMS_FIT2 does not contain the test information at stage 1, the information level at stage 1 in the TEST_FIT1 data set is used to generate boundary values for the test.

The ORDER=LR option uses the LR ordering to derive the p-value, unbiased median estimate, and confidence limits for the regression slope estimate.

The ODS OUTPUT statement with the TEST=TEST_FIT2 option creates an output data set named TEST_FIT2 which contains the updated boundary information for the test at stage 2. The data set also provides the boundary information that is needed for the group sequential test at the next stage.

The "Design Information" table in Output 78.3.13 displays design specifications. By default, the boundary values are modified for the new information levels to maintain the Type I α level.

Output 78.3.13 Design Information

The SEQTEST Procedure

Design Information

BOUNDARY Data Set	WORK.TEST_FIT1
Data Set	WORK.PARMS_FIT2
Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Accept Null
Number of Stages	3
Alpha	0.05
Beta	0.10009
Power	0.89991
Max Information (Percent of Fixed Sample)	103.9552
Max Information	1091.97232
Null Ref ASN (Percent of Fixed Sample)	75.18481
Alt Ref ASN (Percent of Fixed Sample)	101.8341

The derived Type II error probability β and power $1-\beta$ are different because of the new information levels.

With the PSS option, the "Power and Expected Sample Sizes" table in Output 78.3.14 displays powers and expected mean sample sizes under various hypothetical references $\theta = c_i\theta_1$, where θ_1 is the alternative reference and $c_i = 0, 0.5, 1, 1.5$ are the default values in the CREF= option.

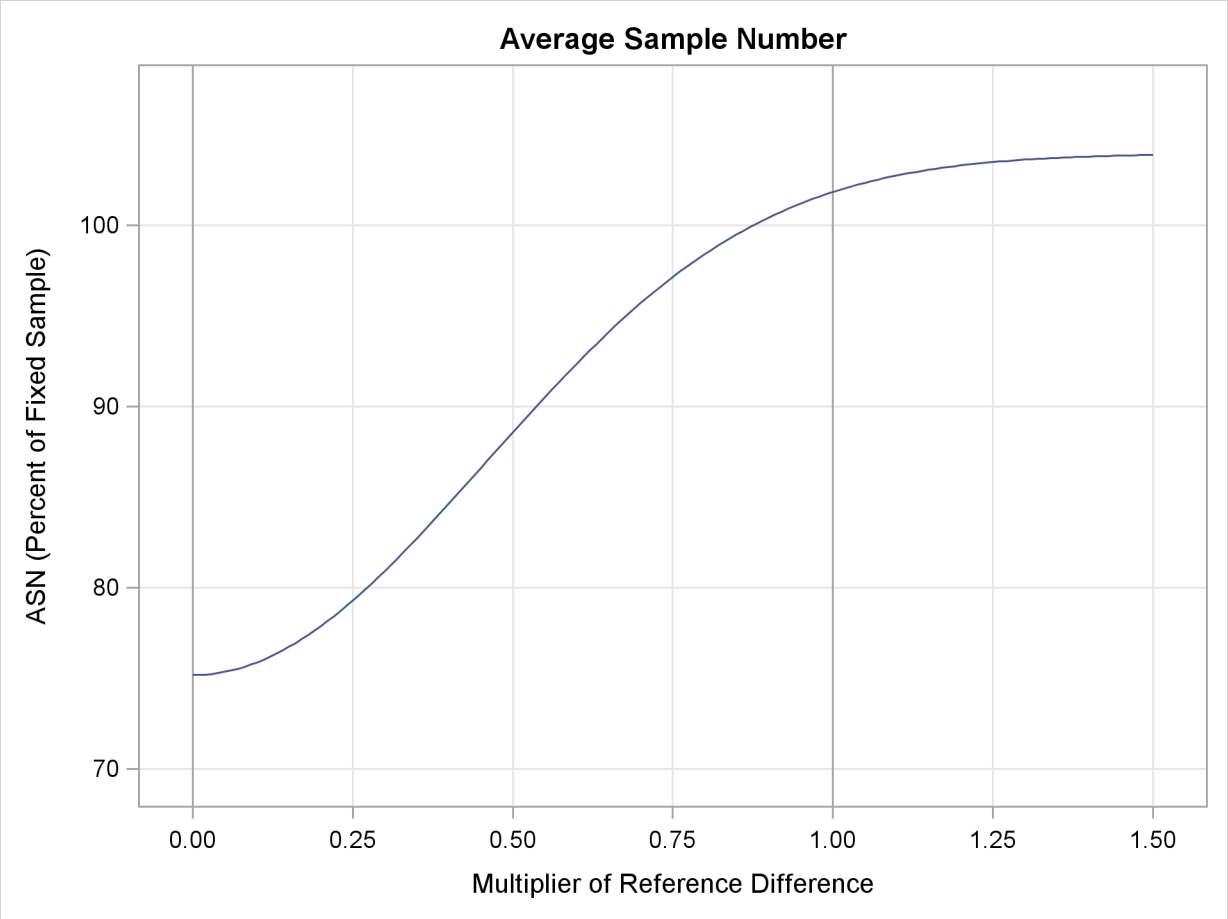
Output 78.3.14 Power and Expected Sample Size Information

Powers and Expected Sample Sizes
Reference = CRef * (Alt Reference)

CRef	Power	-Sample Size- Percent Fixed-Sample
0.0000	0.02500	75.1848
0.5000	0.37154	88.5985
1.0000	0.89991	101.8341
1.5000	0.99758	103.8830

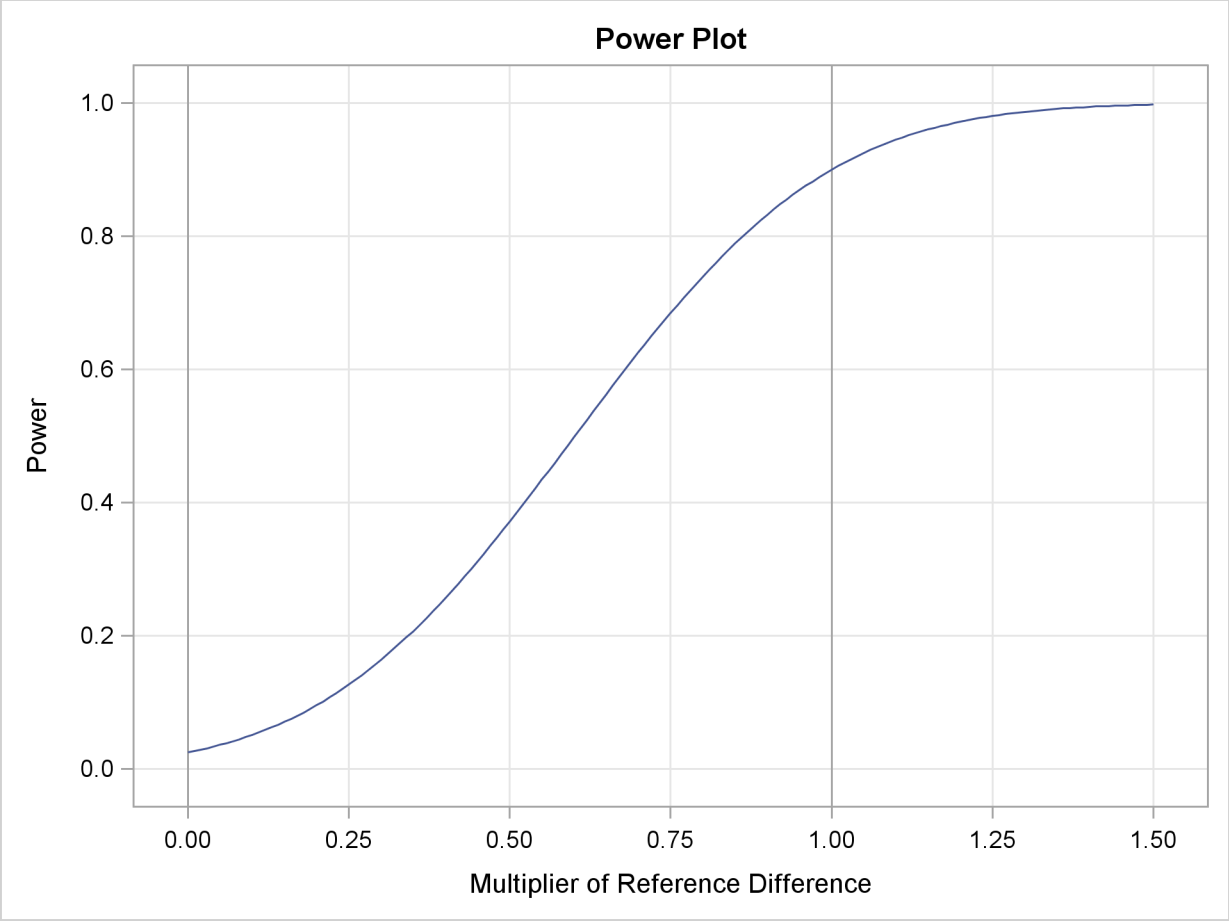
With the PLOTS=ASN option, the procedure displays a plot of expected sample sizes under various hypothetical references, as shown in Output 78.3.15. By default, expected sample sizes under the hypotheses $\theta = c_i\,\theta_1$, $c_i = 0, 0.01, 0.02, \ldots, 1.50$, are displayed, where θ_1 is the alternative reference.

Output 78.3.15 ASN Plot



With the PLOTS=POWER option, the procedure displays a plot of the power curves under various hypothetical references for all designs simultaneously, as shown in Output 78.3.16. By default, powers under hypothetical references $\theta = c_i\,\theta_1$ are displayed, where $c_i = 0, 0.01, 0.02, \ldots, 1.50$ by default. You can specify c_i values with the CREF= option. The c_i values are displayed on the horizontal axis.

Output 78.3.16 Power Plot



Under the null hypothesis, $c_i = 0$, the power is 0.025, which is the upper Type I error probability. Under the alternative hypothesis, $c_i = 1$, the power is 0.89991, which is one minus the Type II error probability, as displayed in the "Design Information" table in Output 78.3.13.

The "Test Information" table in Output 78.3.17 displays the boundary values for the test statistic with the default standardized Z scale. At stage 2, the standardized slope estimate 0.83805 is between the lower and upper β boundary values. The trial stops to accept the null hypothesis that the variable Weight has no effect on the oxygen intake rate after adjusting for other covariates.

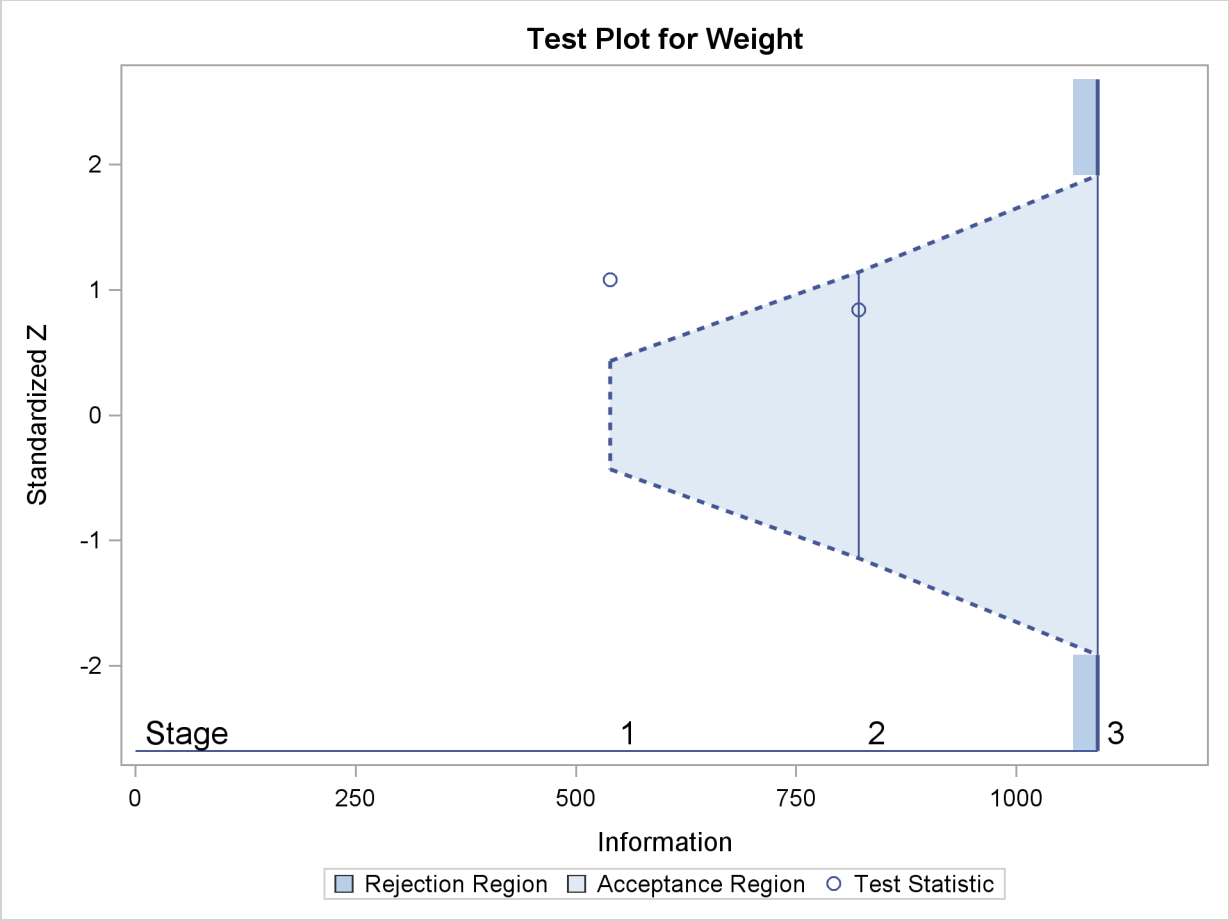
Output 78.3.17 Sequential Tests

```
                 Test Information (Standardized Z Scale)
                           Null Reference = 0

                                          -------Alternative------
                 ---Information Level--   --------Reference-------
      _Stage_    Proportion      Actual        Lower         Upper

            1        0.4934    538.7887     -2.32118       2.32118
            2        0.7517    820.8509     -2.86505       2.86505
            3        1.0000    1091.972     -3.30450       3.30450

                 Test Information (Standardized Z Scale)
                           Null Reference = 0

                -----Boundary Values----    -----------Test----------
                ---Lower--    ---Upper--    ----------Weight---------
      _Stage_         Beta          Beta      Estimate    Action

            1      -0.43019       0.43019       1.08174    Continue
            2      -1.14237       1.14237       0.83805    Accept Null
            3      -1.91409       1.91409             .
```

Since the data set PARMS_FIT2 contains the test information only at stage 2, the information level at stage 1 in the TEST_FIT1 data set is used to generate boundary values for the test.

With the specified ODS GRAPHICS ON statement, a boundary plot with test statistics is displayed by default, as shown in Output 78.3.18. As expected, the test statistic is in the acceptance region between the lower and upper α boundaries at the final stage.

Output 78.3.18 Sequential Test Plot



After a trial is stopped, the "Parameter Estimates" table in Output 78.3.19 displays the stopping stage, parameter estimate, unbiased median estimate, confidence limits, and the p-value under the null hypothesis $H_0 : \beta_w = 0$. As expected, the p-value 0.3056 is not significant at the $\alpha = 0.05$ level, and the confidence interval does contain the value zero. The p-value, unbiased median estimate, and confidence limits depend on the ordering of the sample space (k, z), where k is the stage number and z is the standardized Z statistic. With the specified LR ordering, the p-values are computed with the ordering $(k', z') \succ (k, z)$ if $z' > z$. See the section "Available Sample Space Orderings in a Sequential Test" on page 6025 for a detailed description of the LR ordering.

Output 78.3.19 Parameter Estimates

Parameter Estimates
LR Ordering

Parameter	Stopping Stage	MLE	p-Value for H0:Parm=0	Median Estimate
Weight	2	0.029251	0.3056	0.037078

Parameter Estimates
LR Ordering

Parameter	95% Confidence Limits	
Weight	-0.03368	0.10532

Example 78.4: Testing a Binomial Proportion

This example tests a binomial proportion by using a four-stage group sequential design. Suppose a supermarket is developing a new store-brand coffee. From past studies, the positive response for the current store-brand coffee from customers is around 60%. The store is interested in whether the new brand has a better positive response than the current brand.

A power family method is used for the group sequential trial with the null hypothesis $H_0 : p = p_0 = 0.60$ and a one-sided upper alternative with a power of 0.80 at $H_1 : p = 0.70$. To accommodate the zero null reference that is assumed in the SEQDESIGN procedure, an equivalent hypothesis $H_0 : \theta = 0$ with $H_1 : \theta = 0.10$ is used, where $\theta = p - p_0$. The following statements request a power family method with early stopping to reject the null hypothesis:

```
ods graphics on;
proc seqdesign altref=0.10
               boundaryscale=mle
               ;
   PowerFamily: design method=pow
                       nstages=4
                       alt=upper
                       beta=0.20
                       ;
   samplesize model=onesamplefreq( nullprop=0.6);
ods output Boundary=Bnd_Prop;
run;
ods graphics off;
```

The NULLPROP= option in the SAMPLESIZE statement specifies $p_0 = 0.60$ for the sample size computation. The ODS OUTPUT statement with the BOUNDARY=BND_PROP option creates an output data set named BND_PROP which contains the resulting boundary information for the subsequent sequential tests.

With the BOUNDARYSCALE=MLE option, the procedure displays the output boundaries in terms of the maximum likelihood estimates. The "Design Information" table in Output 78.4.1 displays design specifications and derived statistics. With the specified alternative reference $\theta_1 = p_1 - p_0 = 0.7 - 0.6 = 0.1$, the maximum information 670.38 is also derived.

Output 78.4.1 Design Information

The SEQDESIGN Procedure
Design: PowerFamily

Design Information

Statistic Distribution	Normal
Boundary Scale	MLE
Alternative Hypothesis	Upper
Early Stop	Reject Null
Method	Power Family
Boundary Key	Both
Alternative Reference	0.1
Number of Stages	4
Alpha	0.05
Beta	0.2
Power	0.8
Max Information (Percent of Fixed Sample)	108.4306
Max Information	670.3782
Null Ref ASN (Percent of Fixed Sample)	106.9276
Alt Ref ASN (Percent of Fixed Sample)	78.51072

The "Boundary Information" table in Output 78.4.2 displays the information level, alternative reference, and boundary values at each stage. With the STOP=REJECT option, only the rejection boundary values are displayed.

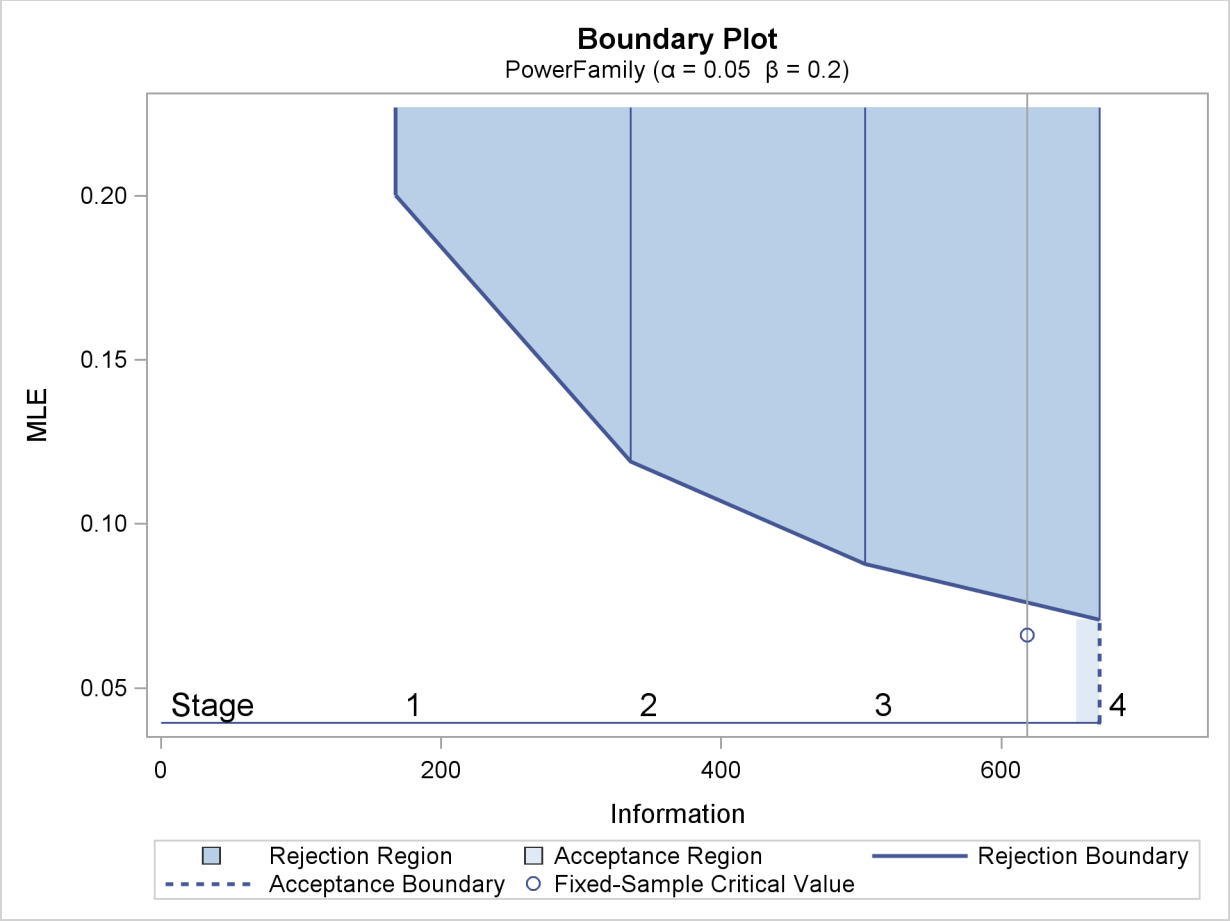
Output 78.4.2 Boundary Information

Boundary Information (MLE Scale)
Null Reference = 0

	Information Level			Alternative Reference	Boundary Values
					Upper
Stage	Proportion	Actual	N	Upper	Alpha
1	0.2500	167.5945	35.19485	0.10000	0.20018
2	0.5000	335.1891	70.38971	0.10000	0.11903
3	0.7500	502.7836	105.5846	0.10000	0.08782
4	1.0000	670.3782	140.7794	0.10000	0.07077

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 78.4.3.

Output 78.4.3 Boundary Plot



With the MODEL=ONESAMPLEFREQ option in the SAMPLESIZE statement, the "Sample Size Summary" table in Output 78.4.4 displays the parameters for the sample size computation.

Output 78.4.4 Required Sample Size Summary

Sample Size Summary

Test	One-Sample Proportion
Null Proportion	0.6
Proportion	0.7
Test Statistic	Z for Proportion
Reference Proportion	Alt Ref
Max Sample Size	140.7794
Expected Sample Size (Null Ref)	138.828
Expected Sample Size (Alt Ref)	101.9333

The "Sample Sizes" table in Output 78.4.5 displays the required sample sizes for the group sequential clinical trial.

Output 78.4.5 Required Sample Sizes

Sample Sizes (N)
One-Sample Z Test for Proportion

Stage	------Fractional N----- N	Information	-------Ceiling N------- N	Information
1	35.19	167.6	36	171.4
2	70.39	335.2	71	338.1
3	105.58	502.8	106	504.8
4	140.78	670.4	141	671.4

Thus, 36 customers are needed at stage 1, and 35 new customers are needed at each of the remaining stages. Suppose that 36 customers are available at stage 1. Output 78.4.6 lists the 10 observations in the data set count_1.

Output 78.4.6 Clinical Trial Data

First 10 Obs in the Trial Data

Obs	Resp
1	1
2	1
3	0
4	0
5	1
6	1
7	0
8	1
9	1
10	1

The Resp variable is an indicator variable with a value of 1 for a customer with a positive response and a value of 0 for a customer without a positive response.

The following statements use the MEANS procedure to compute the mean response at stage 1:

```
proc means data=Prop_1;
   var Resp;
   ods output Summary=Data_Prop1;
run;
```

The following statements create and display (in Output 78.4.7) the data set for the centered mean positive response, $\hat{p} - p_0$:

```
data Data_Prop1;
   set Data_Prop1;
   _Scale_='MLE';
   _Stage_= 1;
   NObs= Resp_N;
```

```
   PDiff= Resp_Mean - 0.6;
   keep _Scale_ _Stage_ NObs PDiff;
run;
proc print data=Data_Prop1;
   title 'Statistics Computed at Stage 1';
run;
```

Output 78.4.7 Statistics Computed at Stage 1

Statistics Computed at Stage 1

Obs	_Scale_	_Stage_	NObs	PDiff
1	MLE	1	36	-0.016667

The following statements invoke the SEQTEST procedure to test for early stopping at stage 1:

```
ods graphics on;
proc seqtest Boundary=Bnd_Prop
             Data(Testvar=PDiff)=Data_Prop1
             boundaryscale=mle
             ;
ods output Test=Test_Prop1;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 1, which was generated in the SEQDESIGN procedure. The DATA=DATA_PROP1 option specifies the input data set DATA_PROP1 that contains the test statistic and its associated sample size at stage 1, and the TESTVAR=PDIFF option identifies the test variable PDIFF in the data set.

The ODS OUTPUT statement with the TEST=TEST_PROP1 option creates an output data set named TEST_PROP1 which contains the updated boundary information for the test at stage 1. The data set also provides the boundary information that is needed for the group sequential test at the next stage.

The "Design Information" table in Output 78.4.8 displays design specifications. By default, when the boundary values are modified for the new information levels, the Type I error probability α is maintained. The power has been modified for the new information levels.

Output 78.4.8 Design Information

The SEQTEST Procedure

Design Information

BOUNDARY Data Set	WORK.BND_PROP
Data Set	WORK.DATA_PROP1
Statistic Distribution	Normal
Boundary Scale	MLE
Alternative Hypothesis	Upper
Early Stop	Reject Null
Number of Stages	4
Alpha	0.05
Beta	0.20015
Power	0.79985
Max Information (Percent of Fixed Sample)	108.4783
Max Information	670.378176
Null Ref ASN (Percent of Fixed Sample)	106.9684
Alt Ref ASN (Percent of Fixed Sample)	78.45899

The "Test Information" table in Output 78.4.9 displays the boundary values for the test statistic with the specified MLE scale.

Output 78.4.9 Sequential Tests

Test Information (MLE Scale)
Null Reference = 0

	Information Level			Alternative Reference	Boundary Values Upper
Stage	Proportion	Actual	N	Upper	Alpha
1	0.2557	171.4286	36	0.10000	0.19638
2	0.5038	337.7451	70.92647	0.10000	0.11845
3	0.7519	504.0616	105.8529	0.10000	0.08771
4	1.0000	670.3782	140.7794	0.10000	0.07082

Test Information (MLE Scale)
Null Reference = 0

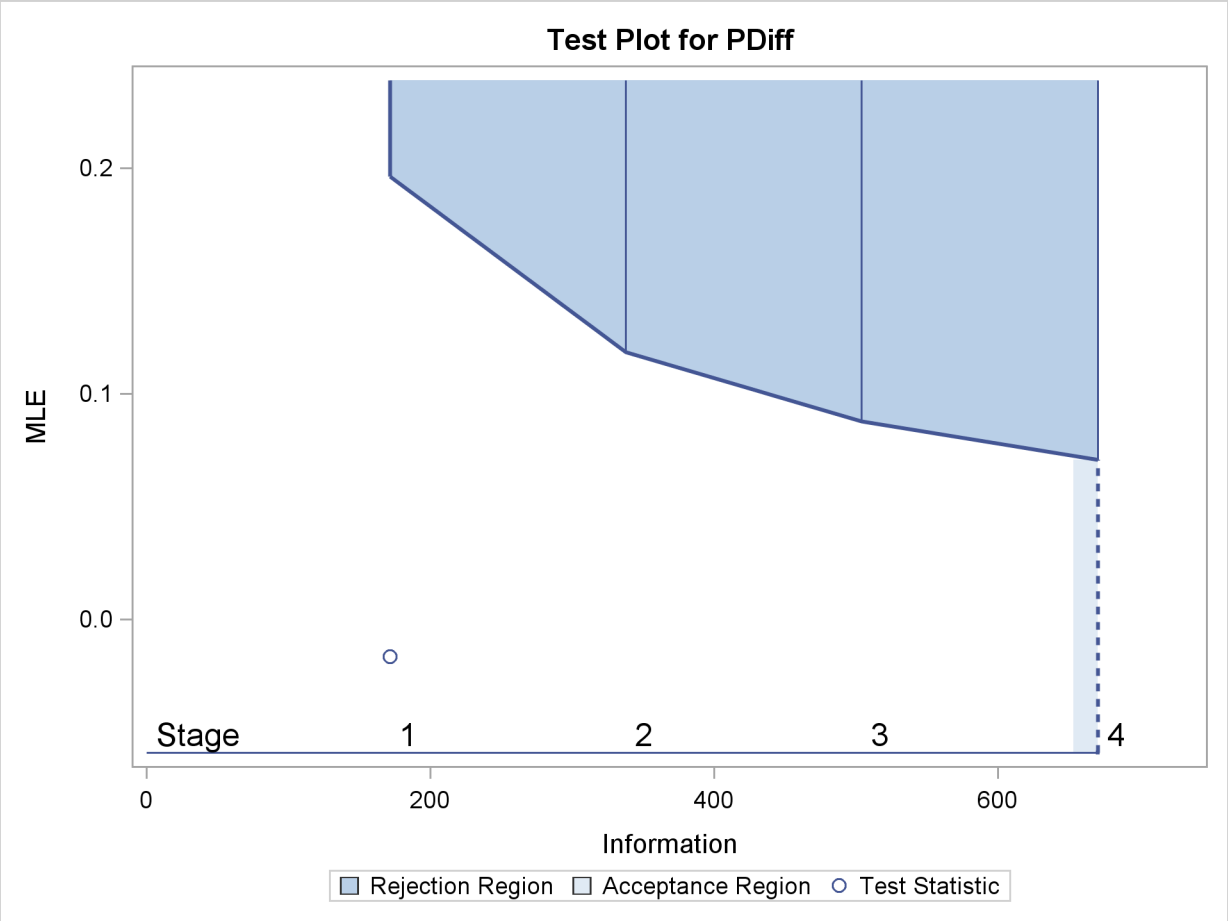
	Test PDiff	
Stage	Estimate	Action
1	-0.01667	Continue
2	.	
3	.	
4	.	

The information level at stage 1 is computed as $I_1^* = I_1 \times (n_1^*/n_1)$, where I_1 and n_1 are the information level and sample size at stage 1 in the BOUNDARY= data set, and $n_1^* = 36$ is the available sample size at stage 1. Since only the information level at stage 1 is derived from the DATA= data

set, the information levels at stages 2 and 3 are derived proportionally from the corresponding information levels in the original design. At stage 1, the statistic $\hat{\theta} = \hat{p} - p_0 = 0.58333 - 0.6 = -0.01667$ is less than the upper α boundary value 0.19638, so the trial continues to the next stage.

With the specified ODS GRAPHICS ON statement, a boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 78.4.10. As expected, the test statistic is in the continuation region.

Output 78.4.10 Sequential Test Plot



The following statements use the MEANS procedure to compute the mean response at stage 2:

```
proc means data=Prop_2;
   var Resp;
   ods output Summary=Data_Prop2;
run;
```

The following statements create and display (in Output 78.4.11) the data set for the centered mean positive response ($\hat{p} - p_0$) at stage 2:

```
data Data_Prop2;
   set Data_Prop2;
   _Scale_='MLE';
```

```
   _Stage_= 2;
   NObs= Resp_N;
   PDiff= Resp_Mean - 0.6;
   keep _Scale_ _Stage_ NObs PDiff;
run;

proc print data=Data_Prop2;
   title 'Statistics Computed at Stage 2';
run;
```

Output 78.4.11 Statistics Computed at Stage 2

Statistics Computed at Stage 2

Obs	_Scale_	_Stage_	NObs	PDiff
1	MLE	2	71	-0.064789

The following statements invoke the SEQTEST procedure to test for early stopping at stage 2:

```
ods graphics on;
proc seqtest Boundary=Test_Prop1
             Data(Testvar=PDiff)=Data_Prop2
             boundaryscale=mle
             condpower(cref=1)
             predpower
             plots=condpower
             ;
ods output test=Test_Prop2;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 2, which was generated by the SEQTEST procedure at the previous stage. The DATA= option specifies the input data set that contains the test statistic and its associated sample size at stage 2, and the TESTVAR= option identifies the test variable in the data set.

The ODS OUTPUT statement with the TEST=TEST_PROP2 option creates an output data set named TEST_PROP2 which contains the updated boundary information for the test at stage 2. The data set also provides the boundary information that is needed for the group sequential test at the next stage.

The CONDPOWER(CREF=1) option requests the conditional power with the observed statistic under the alternative hypothesis, in addition to the conditional power under the hypothetical reference $\theta = \hat{\theta}$, the MLE estimate. The PREDPOWER option requests the noninformative predictive power with the observed statistic.

The "Test Information" table in Output 78.4.12 displays the boundary values for the test statistic with the specified MLE scale. The test statistic $\hat{\theta} = -0.06479$ is less than the corresponding upper α boundary 0.11834, so the sequential test does not stop at stage 2 to reject the null hypothesis.

Output 78.4.12 Sequential Tests

The SEQTEST Procedure

Test Information (MLE Scale)
Null Reference = 0

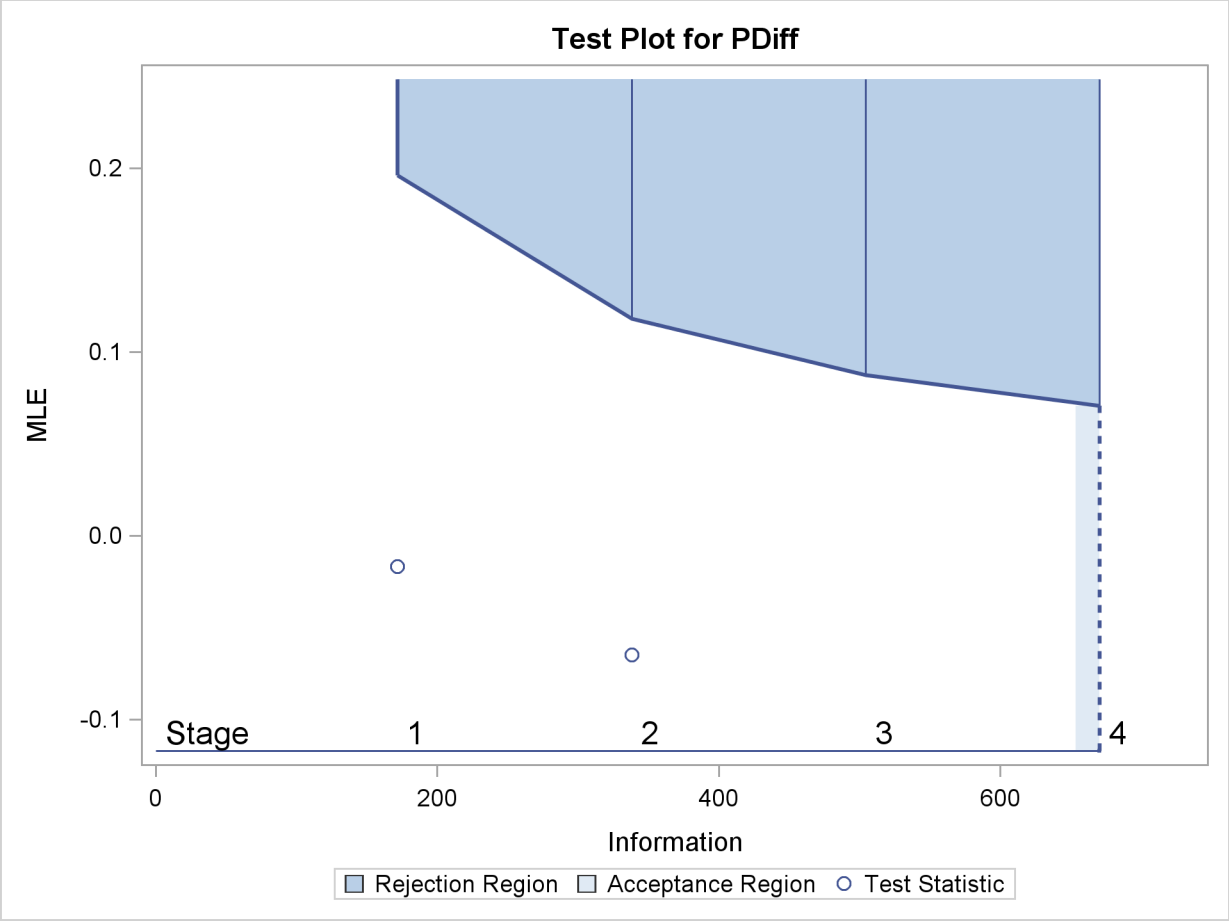
	Information Level			Alternative Reference	Boundary Values Upper	
Stage	Proportion	Actual	N	Upper	Upper	Alpha
1	0.2557	171.4286	36	0.10000		0.19638
2	0.5043	338.0952	71	0.10000		0.11834
3	0.7522	504.2367	105.8897	0.10000		0.08769
4	1.0000	670.3782	140.7794	0.10000		0.07082

Test Information (MLE Scale)
Null Reference = 0

	Test PDiff	
Stage	Estimate	Action
1	-0.01667	Continue
2	-0.06479	Continue
3	.	
4	.	

With the specified ODS GRAPHICS ON statement, the "Test Plot" displays boundary values of the design and the test statistic by default, as shown in Output 78.4.13. It also shows that the test statistic is in the "Continuation Region" below the upper α boundary value at stage 2.

Output 78.4.13 Sequential Test Plot



The predictive power is the probability to reject the null hypothesis under the posterior distribution with a noninformative prior given the observed statistic $\hat{\theta} = -0.06479$. The "Predictive Power Information" table in Output 78.4.14 indicates that the predictive power at $\hat{\theta} = -0.06479$ is 0.0002.

Output 78.4.14 Predictive Power

Predictive Power Information

Stopping Stage	MLE	Predictive Power
2	-0.06479	0.00020

The "Conditional Power Information" table in Output 78.4.15 displays conditional powers given the observed statistic under hypothetical references $\theta = \hat{\theta}$, the maximum likelihood estimate, and $\theta = \theta_1$. The constant c under CRef for the MLE is derived from $\hat{\theta} = c\theta_1$; that is, $c = \hat{\theta}/\theta_1 = -0.06479/0.1 = -0.6479$.

Output 78.4.15 Conditional Power

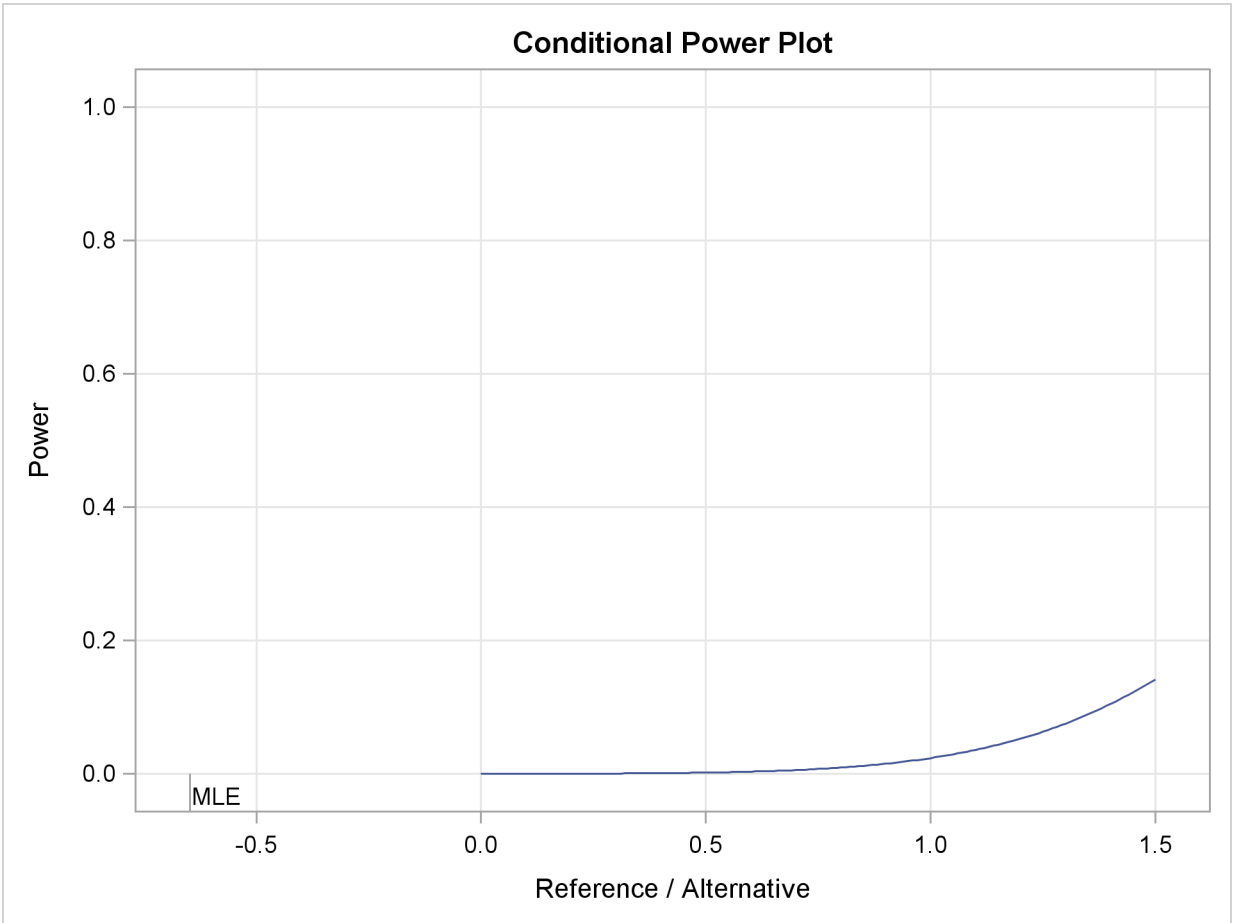
Conditional Power Information
Reference = CRef * (Alt Reference)

Stopping Stage	MLE	Reference Ref	Reference CRef	Conditional Power
2	−0.06479	MLE	−0.6479	0.00000
2	−0.06479	Alternative	1.0000	0.02367

The conditional power is the probability of rejecting the null hypothesis under these hypothetical references given the observed statistic $\hat{\theta} = -0.06479$. The table in Output 78.4.15 shows a weak conditional power of 0.02367 under the alternative hypothesis.

The "Conditional Power Plot" displays conditional powers given the observed statistic under various hypothetical references, as shown in Output 78.4.16. These references include $\theta = \hat{\theta}$, the maximum likelihood estimate, and $\theta = c_i\theta_1$, where θ_1 is the alternative reference and $c_i = 0, 0.01, \ldots, 1.50$ are constants that are specified in the CREF= option. Output 78.4.16 shows that the conditional power increases as c_i increases.

Output 78.4.16 Conditional Power Plot



With a predictive power 0.0002 and a conditional power of 0.02367 under H_1, the supermarket decides to stop the trial and accept the null hypothesis. That is, the positive response for the new store-brand coffee is not better than that for the current store-brand coffee.

Output 78.4.17 Predictive Power

Predictive Power Information

Stopping Stage	MLE	Predictive Power
2	-0.06479	0.00020

In the SEQTEST procedure, the conditional probability at an interim stage k is the probability that the test statistic at the final stage (stage 4) would exceed the rejection critical value. Since an interim stage exists between the current stage (stage 2) and the final stage, the conditional power is not the conditional probability to reject the null hypothesis H_0.

The following statements invoke the SEQTEST procedure to test for early stopping at stage 2. The NSTAGES=3 option sets the next stage as the final stage (stage 3). The CONDPOWER(CREF=1) option requests the conditional power with the observed statistic under the alternative hypothesis, in addition to the conditional power under the hypothetical reference $\theta = \hat{\theta}$, the MLE estimate.

```
proc seqtest Boundary=Test_Prop1
             Data(Testvar=PDiff)=Data_Prop2
             nstages=3
             boundaryscale=mle
             condpower(cref=1)
             ;
run;
```

The "Test Information" table in Output 78.4.18 displays the boundary values for the test statistic with the specified MLE scale, assuming that the next stage is the final stage.

Output 78.4.18 Sequential Tests

The SEQTEST Procedure

Test Information (MLE Scale)
Null Reference = 0

Stage	Information Level Proportion	Information Level Actual	Information Level N	Alternative Reference Upper	Boundary Values Upper Alpha
1	0.2557	171.4286	36	0.10000	0.19638
2	0.5043	338.0952	71	0.10000	0.11834
3	1.0000	670.3782	140.7794	0.10000	0.06730

Test Information (MLE Scale)
Null Reference = 0

Stage	Test PDiff Estimate	Test PDiff Action
1	-0.01667	Continue
2	-0.06479	Continue
3	.	

The "Conditional Power Information" table in Output 78.4.19 displays conditional powers given the observed statistic, assuming that the next stage is the final stage.

Output 78.4.19 Conditional Power

Conditional Power Information
Reference = CRef * (Alt Reference)

Stopping Stage	MLE	Reference Ref	Reference CRef	Conditional Power
2	-0.06479	MLE	-0.6479	0.00000
2	-0.06479	Alternative	1.0000	0.03188

The conditional power is the probability of rejecting the null hypothesis under these hypothetical references given the observed statistic $\hat{\theta} = -0.06479$. The table in Output 78.4.19 also shows a weak conditional power of 0.03188 under the alternative hypothesis.

Example 78.5: Comparing Two Proportions with a Log Odds Ratio Test

This example compares two binomial proportions by using a log odds ratio statistic in a five-stage group sequential test. A clinic is studying the effect of vitamin C supplements in treating flu symptoms. The study consists of patients in the clinic who exhibit the first sign of flu symptoms within the last 24 hours. These patients are randomly assigned to either the control group (which receives placebo pills) or the treatment group (which receives large doses of vitamin C supplements). At the end of a five-day period, the flu symptoms of each patient are recorded.

Suppose that you know from past experience that flu symptoms disappear in five days for 60% of patients who experience flu symptoms. The clinic would like to detect a 70% symptom disappearance with a high probability. A test that compares the proportions directly specifies the null hypothesis $H_0 : \theta = p_t - p_c = 0$ with a one-sided alternative $H_1 : \theta > 0$ and a power of 0.90 at $H_1 : \theta = 0.10$, where p_t and p_c are the proportions of symptom disappearance in the treatment group and control group, respectively. An alternative trial tests an equivalent hypothesis by using the log odds ratio statistics:

$$\theta = \log\left(\frac{(\frac{p_t}{1-p_t})}{(\frac{p_c}{1-p_c})}\right)$$

Then the null hypothesis is $H_0 : \theta = \theta_0 = 0$ and the alternative hypothesis is

$$H_1 : \theta = \theta_1 = \log\left(\frac{(\frac{0.70}{0.30})}{(\frac{0.6}{0.4})}\right) = 0.441833$$

The following statements invoke the SEQDESIGN procedure and request a five-stage group sequential design by using an error spending function method for normally distributed statistics. The design uses a two-sided alternative hypothesis with early stopping to reject the null hypothesis H_0.

```
ods graphics on;
proc seqdesign altref=0.441833
               boundaryscale=mle
               ;
   OneSidedErrorSpending: design method=errfuncpow
                                 nstages=5
                                 alt=upper
                                 stop=accept
                                 alpha=0.025;
   samplesize model=twosamplefreq( nullprop=0.6 test=logor);
ods output Boundary=Bnd_CSup;
run;
ods graphics off;
```

The ODS OUTPUT statement with the BOUNDARY=BND_CSUP option creates an output data set named BND_CSUP which contains the resulting boundary information for the subsequent sequential tests.

The "Design Information" table in Output 78.5.1 displays design specifications and derived statistics. With the specified alternative reference, the maximum information 56.30934 is derived.

Output 78.5.1 Design Information

The SEQDESIGN Procedure
Design: OneSidedErrorSpending

Design Information

Statistic Distribution	Normal
Boundary Scale	MLE
Alternative Hypothesis	Upper
Early Stop	Accept Null
Method	Error Spending
Boundary Key	Both
Alternative Reference	0.441833
Number of Stages	5
Alpha	0.025
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	104.6166
Max Information	56.30934
Null Ref ASN (Percent of Fixed Sample)	57.21399
Alt Ref ASN (Percent of Fixed Sample)	102.1058

The "Boundary Information" table in Output 78.5.2 displays information level, alternative reference, and boundary values at each stage. With the specified BOUNDARYSCALE=MLE option, the procedure displays the output boundaries in terms of the MLE scale.

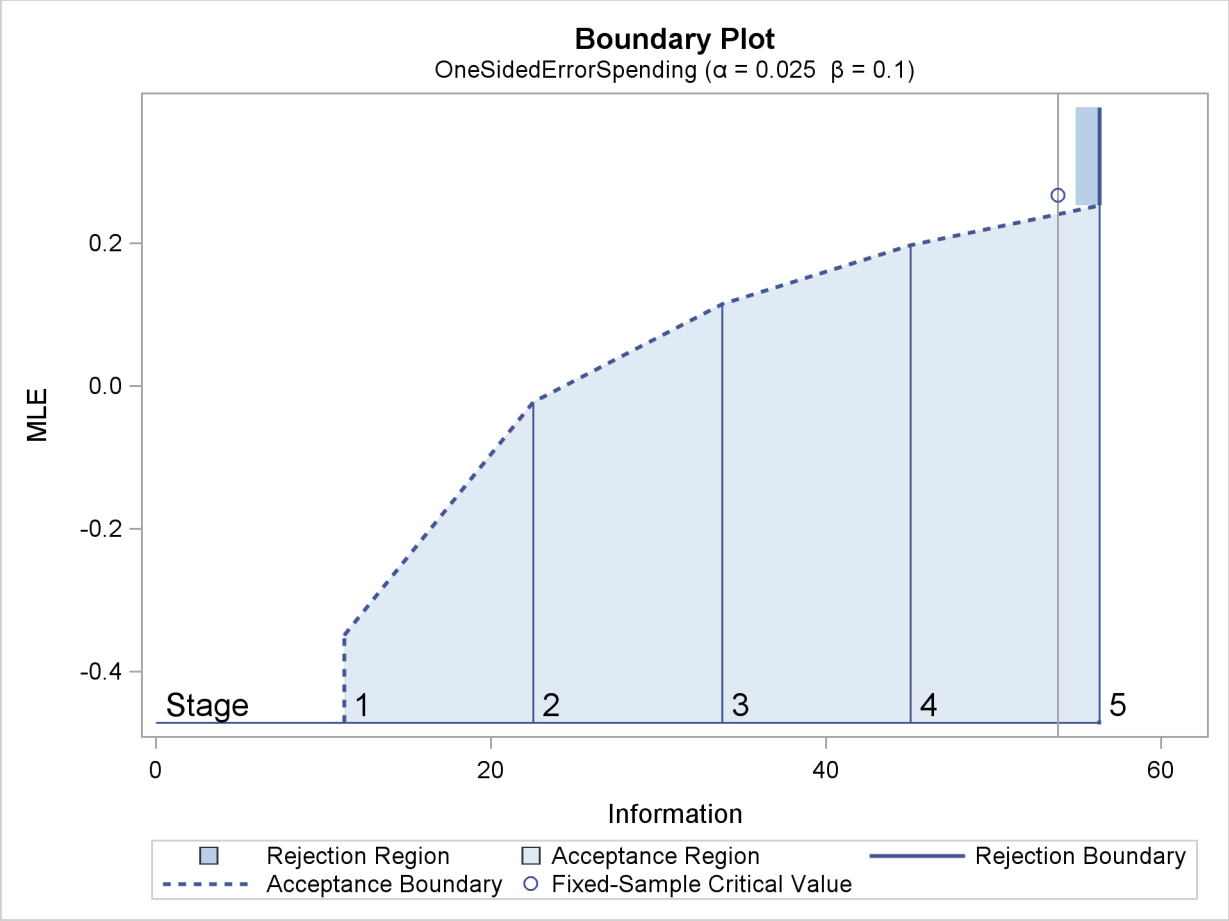
Output 78.5.2 Boundary Information

Boundary Information (MLE Scale)
Null Reference = 0

	Information Level			Alternative Reference	Boundary Values
Stage	Proportion	Actual	N	Upper	Upper Beta
1	0.2000	11.26187	201.1048	0.44183	−0.34844
2	0.4000	22.52374	402.2096	0.44183	−0.02262
3	0.6000	33.7856	603.3144	0.44183	0.11527
4	0.8000	45.04747	804.4192	0.44183	0.19708
5	1.0000	56.30934	1005.524	0.44183	0.25345

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 78.5.3.

Output 78.5.3 Boundary Plot



With the SAMPLESIZE statement, the "Sample Size Summary" table in Output 78.5.4 displays the parameters for the sample size computation.

Output 78.5.4 Sample Size Summary

Sample Size Summary

Test	Two-Sample Proportions
Null Proportion	0.6
Proportion (Group A)	0.7
Test Statistic	Log Odds Ratio
Reference Proportions	Alt Ref
Max Sample Size	1005.524
Expected Sample Size (Null Ref)	549.9132
Expected Sample Size (Alt Ref)	981.3914

The "Sample Sizes" table in Output 78.5.5 displays the required sample sizes for the group sequential clinical trial.

Output 78.5.5 Required Sample Sizes

Sample Sizes (N)
Two-Sample Log Odds Ratio Test for Proportion Difference

	Fractional N			
Stage	N	N(Grp 1)	N(Grp 2)	Information
1	201.10	100.55	100.55	11.2619
2	402.21	201.10	201.10	22.5237
3	603.31	301.66	301.66	33.7856
4	804.42	402.21	402.21	45.0475
5	1005.52	502.76	502.76	56.3093

Sample Sizes (N)
Two-Sample Log Odds Ratio Test for Proportion Difference

	Ceiling N			
Stage	N	N(Grp 1)	N(Grp 2)	Information
1	202	101	101	11.3120
2	404	202	202	22.6240
3	604	302	302	33.8240
4	806	403	403	45.1360
5	1006	503	503	56.3360

Thus, 101 new patients are needed in each group at stages 1, 2, and 4, and 100 new patients are needed in each group at stages 3 and 5. Suppose that 101 patients are available in each group at stage 1. Output 78.5.6 lists the 10 observations in the data set count_1.

Output 78.5.6 Clinical Trial Data

First 10 Obs in the Trial Data

Obs	TrtGrp	Resp
1	Control	1
2	C_Sup	0
3	Control	0
4	C_Sup	1
5	Control	1
6	C_Sup	1
7	Control	1
8	C_Sup	0
9	Control	0
10	C_Sup	1

The TrtGrp variable is a grouping variable with the value Control for a patient in the placebo control group and the value C_Sup for a patient in the treatment group who receives vitamin C supplements. The Resp variable is an indicator variable with the value 1 for a patient without flu symptoms after five days and the value 0 for a patient with flu symptoms after five days.

The following statements use the LOGISTIC procedure to compute the log odds ratio statistic and its associated standard error at stage 1:

```
proc logistic data=CSup_1 descending;
   class TrtGrp / param=ref;
   model Resp= TrtGrp;
   ods output ParameterEstimates=Parms_CSup1;
run;
```

The DESCENDING option is used to reverse the order for the response levels, so the LOGISTIC procedure is modeling the probability that Resp = 1.

The following statements create and display (in Output 78.5.7) the data set for the log odds ratio statistic and its associated standard error:

```
data Parms_CSup1;
   set Parms_CSup1;
   if Variable='TrtGrp' and ClassVal0='C_Sup';
   _Scale_='MLE';
   _Stage_= 1;
   keep _Scale_ _Stage_ Variable Estimate StdErr;
run;

proc print data=Parms_CSup1;
   title 'Statistics Computed at Stage 1';
run;
```

Output 78.5.7 Statistics Computed at Stage 1

Statistics Computed at Stage 1

Obs	Variable	Estimate	StdErr	_Scale_	_Stage_
1	TrtGrp	0.3247	0.2856	MLE	1

The following statements invoke the SEQTEST procedure to test for early stopping at stage 1:

```
ods graphics on;
proc seqtest Boundary=Bnd_CSup
             Parms(Testvar=TrtGrp)=Parms_CSup1
             boundaryadj=errfuncpow
             boundarykey=both
             boundaryscale=mle
             ;
ods output test=Test_CSup1; run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 1, which was generated in the SEQDESIGN procedure. The PARMS=PARMS_CSUP1 option specifies the input data set PARMS_CSUP1 that contains the test statistic and its associated standard error at stage 1, and the TESTVAR=TRTGRP option identifies the test variable TRTGRP in the data set.

The ODS OUTPUT statement with the TEST=TEST_CSUP1 option creates an output data set named TEST_CSUP1 which contains the updated boundary information for the test at stage 1. The data set also provides the boundary information that is needed for the group sequential test at the next stage.

The "Design Information" table in Output 78.5.8 displays design specifications. With the specified BOUNDARYKEY=BOTH option, the information levels and boundary values at future stages are modified to maintain both the α and β levels.

Output 78.5.8 Design Information

The SEQTEST Procedure

Design Information

BOUNDARY Data Set	WORK.BND_CSUP
Data Set	WORK.PARMS_CSUP1
Statistic Distribution	Normal
Boundary Scale	MLE
Alternative Hypothesis	Upper
Early Stop	Accept Null
Number of Stages	5
Alpha	0.025
Beta	0.1
Power	0.9
Max Information (Percent of Fixed Sample)	104.6673
Max Information	56.3361718
Null Ref ASN (Percent of Fixed Sample)	57.02894
Alt Ref ASN (Percent of Fixed Sample)	102.1369

The "Test Information" table in Output 78.5.9 displays the boundary values for the test statistic with the specified MLE scale. Since the information level at stage 1 is derived from the PARMS= data set and other information levels are not specified, equal increments are used at remaining stages. At stage 1, the MLE statistic 0.32474 is greater than the corresponding upper β boundary value -0.29906, so the sequential test continues to the next stage.

Output 78.5.9 Sequential Tests

Test Information (MLE Scale)
Null Reference = 0

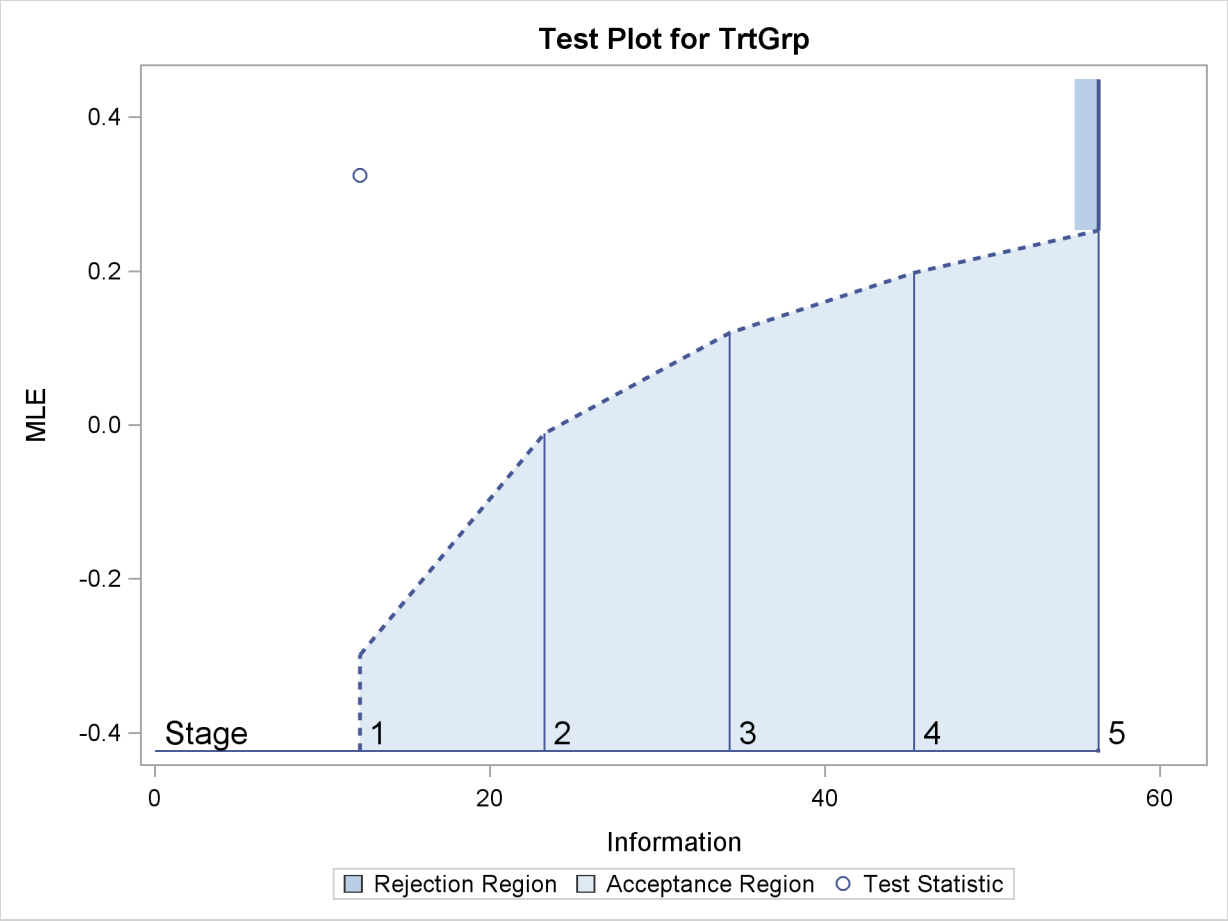
Stage	---Information Level-- Proportion	Actual	-Alternative- --Reference-- Upper
1	0.2176	12.26014	0.44183
2	0.4132	23.27914	0.44183
3	0.6088	34.29815	0.44183
4	0.8044	45.31716	0.44183
5	1.0000	56.33617	0.44183

Test Information (MLE Scale)
Null Reference = 0

Stage	-Boundary Values- ------Upper------ Beta	----------Test---------- ----------TrtGrp---------- Estimate	Action
1	-0.29906	0.32474	Continue
2	-0.01067	.	
3	0.11942	.	
4	0.19829	.	
5	0.25325	.	

With the specified ODS GRAPHICS ON statement, a boundary plot with the boundary values and test statistics is displayed, as shown in Output 78.5.10. As expected, the test statistic is in the continuation region.

Output 78.5.10 Sequential Test Plot



The following statements use the LOGISTIC procedure to compute the log odds ratio statistic and its associated standard error at stage 2:

```
proc logistic data=CSup_2 descending;
   class TrtGrp / param=ref;
   model Resp= TrtGrp;
   ods output ParameterEstimates=Parms_CSup2;
run;
```

The following statements create and display (in Output 78.5.11) the data set for the mean positive response and its associated standard error at stage 2:

```
data Parms_CSup2;
   set Parms_CSup2;
   if Variable='TrtGrp' and ClassVal0='C_Sup';
   _Scale_='MLE';
   _Stage_= 2;
   keep _Scale_ _Stage_ Variable Estimate StdErr;
run;
proc print data=Parms_CSup2;
   title 'Statistics Computed at Stage 2';
run;
```

Output 78.5.11 Statistics Computed at Stage 2

Statistics Computed at Stage 2

Obs	Variable	Estimate	StdErr	_Scale_	_Stage_
1	TrtGrp	0.2356	0.2073	MLE	2

The following statements invoke the SEQTEST procedure to test for early stopping at stage 2:

```
proc seqtest Boundary=Test_CSup1
             Parms( testvar=TrtGrp)=Parms_CSup2
             boundaryadj=errfuncpow
             boundarykey=both
             boundaryscale=mle
             ;
ods output Test=Test_CSup2;
run;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 2, which was generated by the SEQTEST procedure at the previous stage. The PARMS= option specifies the input data set that contains the test statistic and its associated standard error at stage 2, and the TESTVAR= option identifies the test variable in the data set.

The ODS OUTPUT statement with the TEST=CSUP_LDL2 option creates an output data set named CSUP_LDL2 which contains the updated boundary information for the test at stage 2. The data set also provides the boundary information that is needed for the group sequential test at the next stage.

The "Test Information" table in Output 78.5.12 displays the boundary values for the test statistic with the specified MLE scale. The test statistic 0.2356 is greater than the corresponding upper β boundary value -0.01068, so the sequential test continues to the next stage.

Output 78.5.12 Sequential Tests

The SEQTEST Procedure

Test Information (MLE Scale)
Null Reference = 0

Stage	Information Level Proportion	Information Level Actual	Alternative Reference Upper
1	0.2176	12.26014	0.44183
2	0.4132	23.27916	0.44183
3	0.6088	34.29799	0.44183
4	0.8044	45.31681	0.44183
5	1.0000	56.33563	0.44183

Test Information (MLE Scale)
Null Reference = 0

Stage	Boundary Values Upper Beta	Test TrtGrp Estimate	Test TrtGrp Action
1	−0.29906	0.32474	Continue
2	−0.01068	0.23560	Continue
3	0.11942	.	
4	0.19829	.	
5	0.25325	.	

Similar results are found at stages 3 and stage 4, so the trial continues to the final stage. The following statements use the LOGISTIC procedure to compute the log odds ratio statistic and its associated standard error at stage 5:

```
proc logistic data=CSup_5 descending;
   class TrtGrp / param=ref;
   model Resp= TrtGrp;
   ods output ParameterEstimates=Parms_CSup5;
run;
```

The following statements create and display (in Output 78.5.13) the data set for the log odds ratio statistic and its associated standard error at stage 5:

```
data Parms_CSup5;
   set Parms_CSup5;
   if Variable='TrtGrp' and ClassVal0='C_Sup';
   _Scale_='MLE';
   _Stage_= 5;
   keep _Scale_ _Stage_ Variable Estimate StdErr;
run;

proc print data=Parms_CSup5;
   title 'Statistics Computed at Stage 5';
run;
```

Output 78.5.13 Statistics Computed at Stage 5

Statistics Computed at Stage 5

Obs	Variable	Estimate	StdErr	_Scale_	_Stage_
1	TrtGrp	0.2043	0.1334	MLE	5

The following statements invoke the SEQTEST procedure to test for the hypothesis at stage 5:

```
ods graphics on;
proc seqtest Boundary=Test_CSup4
             Parms( testvar=TrtGrp)=Parms_CSup5
             boundaryadj=errfuncpow
             boundaryscale=mle
             cialpha=.025
             rci
             plots=rci
             ;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 5, which was generated by the SEQTEST procedure at the previous stage. The PARMS= option specifies the input data set that contains the test statistic and its associated standard error at stage 5, and the TESTVAR= option identifies the test variable in the data set. With the default BOUNDARYKEY=ALPHA option, the boundary value at stage 5 is derived to maintain the α level.

The "Test Information" table in Output 78.5.14 displays the boundary values for the test statistic with the specified MLE scale. The test statistic 0.2043 is less than the corresponding upper β boundary 0.25375, so the sequential test stops to accept the null hypothesis. That is, there is no reduction in duration of symptoms for the group receiving vitamin C supplements.

Output 78.5.14 Sequential Tests

The SEQTEST Procedure

Test Information (MLE Scale)
Null Reference = 0

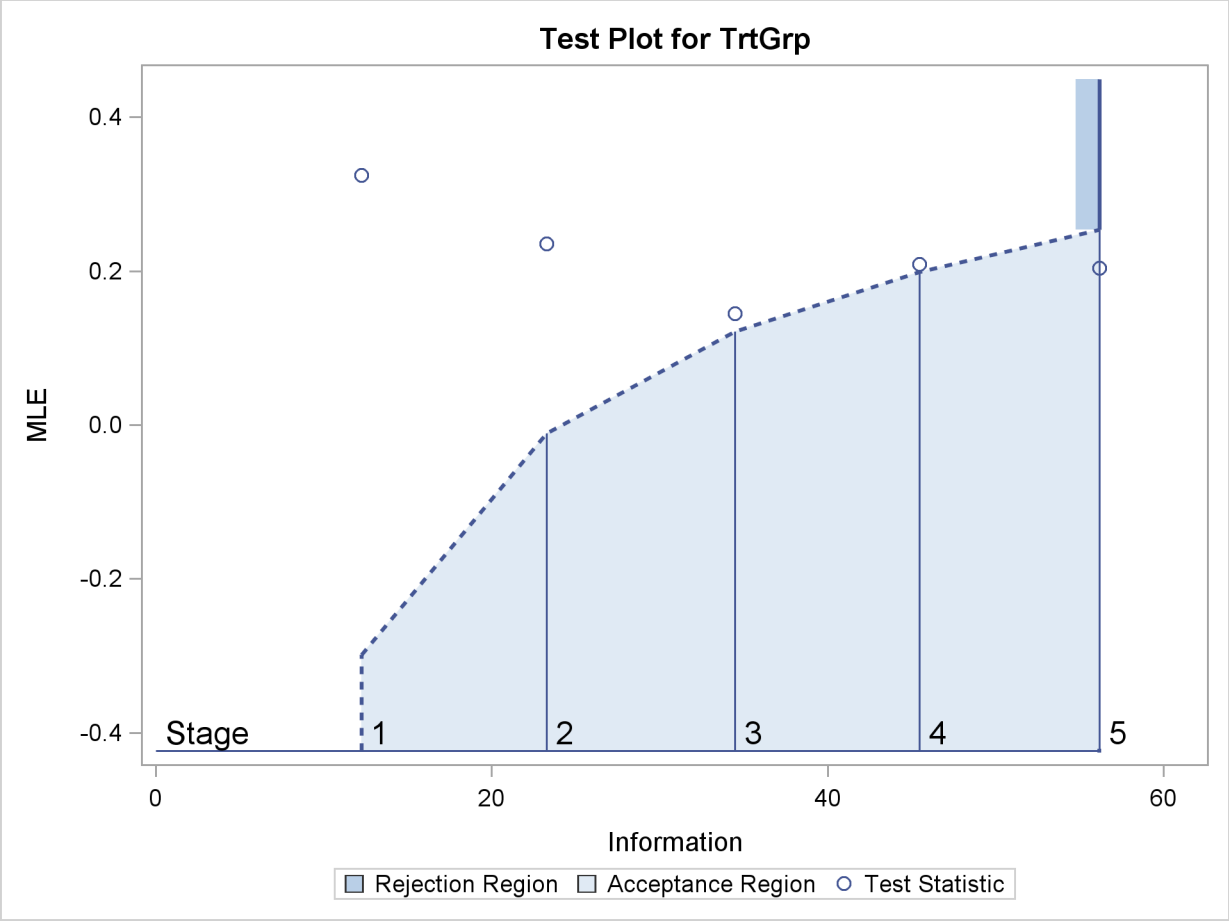
Stage	Information Level Proportion	Information Level Actual	Alternative Reference Upper
1	0.2183	12.26014	0.44183
2	0.4145	23.27916	0.44183
3	0.6141	34.48793	0.44183
4	0.8092	45.44685	0.44183
5	1.0000	56.16068	0.44183

Test Information (MLE Scale)
Null Reference = 0

Stage	Boundary Values Upper Beta	Test TrtGrp Estimate	Action
1	−0.29906	0.32474	Continue
2	−0.01068	0.23560	Continue
3	0.12134	0.14482	Continue
4	0.19899	0.20855	Continue
5	0.25375	0.20430	Accept Null

The "Test Plot" displays boundary values of the design and the test statistics, as shown in Output 78.5.15. It also shows that the test statistic is in the "Acceptance Region" at the final stage.

Output 78.5.15 Sequential Test Plot



After a trial is stopped, the "Parameter Estimates" table in Output 78.5.16 displays the stopping stage, parameter estimate, unbiased median estimate, confidence limits, and the p-value under the null hypothesis $H_0 : \theta = 0$. As expected, the p-value 0.0456 is not significant at $\alpha = 0.025$ level and the lower 97.5% confidence limit is less than the value $\theta_0 = 0$. The p-value, unbiased median estimate, and confidence limits depend on the ordering of the sample space (k, z), where k is the stage number and z is the standardized Z statistic.

Output 78.5.16 Parameter Estimates

Parameter Estimates
Stagewise Ordering

Parameter	Stopping Stage	MLE	p-Value for H0:Parm=0	Median Estimate	Lower 97.5% CL
TrtGrp	5	0.204303	0.0456	0.234494	-0.03712

Since the test is accepted at stage 5, the p-value computed by using the default stagewise ordering

can be expressed as

$$\alpha_u = P_{\theta=0}\left(z_5 < Z_5 \mid b_k < Z_k, k < 5\right)$$

where $z_5 = 1.53105$ is the test statistic at stage 5, Z_k is a standardized normal variate at stage k, and b_k is the upper β boundary value in the standardized Z scale at stage $k, k = 1, 2, \ldots, 5$.

With the RCI option, the "Repeated Confidence Intervals" table in Output 78.5.17 displays repeated confidence intervals for the parameter. For a one-sided test with an upper alternative hypothesis, since the upper acceptance repeated confidence limit 0.3924 at the final stage is less than the alternative reference 0.441833, the null hypothesis is accepted.

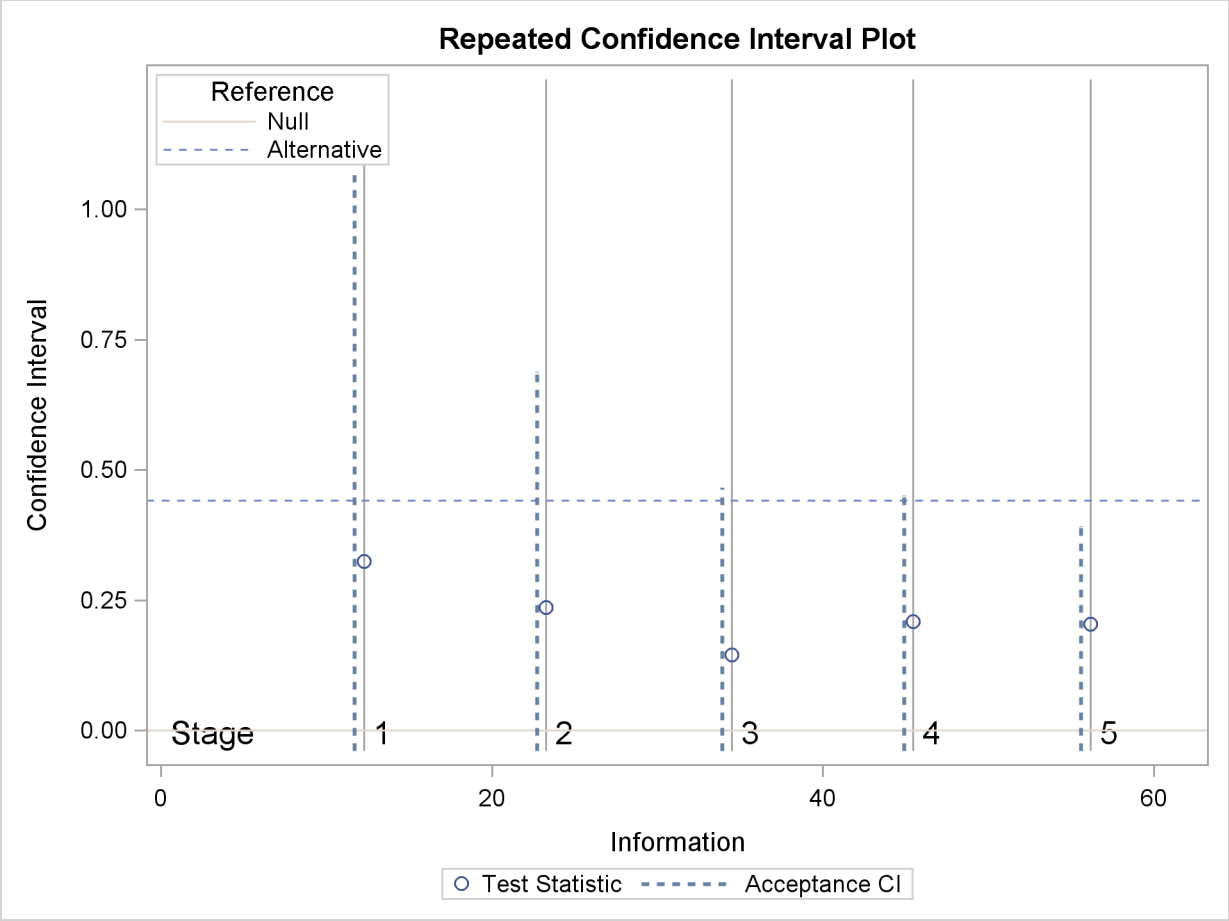
Output 78.5.17 Repeated Confidence Intervals

Repeated Confidence Intervals

Stage	Information Level	Parameter Estimate	-Acceptance Boundary- Upper 89.94% CL
1	12.2601	0.32474	1.0656
2	23.2792	0.23560	0.6881
3	34.4879	0.14482	0.4653
4	45.4468	0.20855	0.4514
5	56.1607	0.20430	0.3924

With the PLOTS=RCI option, the "Repeated Confidence Intervals Plot" displays repeated confidence intervals for the parameter, as shown in Output 78.5.18. It shows that the upper acceptance repeated confidence limit at the final stage is less than the alternative reference 0.441833. This implies that the study accepts the null hypothesis at the final stage.

Output 78.5.18 Repeated Confidence Intervals Plot



Example 78.6: Comparing Two Survival Distributions with a Log-Rank Test

This example requests a log-rank test that compares two survival distributions for the treatment effect (Jennison and Turnbull 2000, pp. 77–79; Whitehead 1997, pp. 36–39).

A clinic is studying the effect of a new cancer treatment. The study consists of mice exposed to a carcinogen and randomized to either the control group or the treatment group. The event of interest is the death from cancer induced by the carcinogen, and the response is the time from randomization to death.

Following the derivations in the section "Test for Two Survival Distributions with a Log-Rank Test" in the chapter "The SEQDESIGN Procedure," the hypothesis $H_0 : \theta = -\log(\lambda) = 0$ with an alternative hypothesis $H_1 : \theta = \theta_1 > 0$ can be used, where λ is the hazard ratio between the treatment group and control group.

Suppose that from past experience, the median survival time for the control group is $t_0 = 20$ weeks, and the study would like to detect a $t_1 = 40$ weeks median survival time with a 80% power in the trial. Assuming exponential survival functions for the two groups, the hazard rates can be computed

from

$$S_j(t_j) = e^{-h_j t_j} = \frac{1}{2}$$

where $j = 0, 1$.

Thus, with $h_0 = 0.03466$ and $h_1 = 0.01733$, the hazard ratio $\lambda_1 = h_1/h_0 = 1/2$ and the alternative hypothesis is

$$\theta_1 = -\log(\lambda_1) = -\log(\frac{1}{2}) = 0.69315$$

The following statements invoke the SEQDESIGN procedure and request a four-stage group sequential design for normally distributed data. The design uses a one-sided alternative hypothesis with early stopping to reject and to accept the null hypothesis H_0. Whitehead's triangular method is used to derive the boundaries.

```
ods graphics on;
proc seqdesign altref=0.69315
               boundaryscale=score
               ;
   OneSidedWhitehead: design method=whitehead
                             nstages=4
                             boundarykey=alpha
                             alt=upper  stop=both
                             beta=0.20;
   samplesize model=twosamplesurvival
                    ( nullhazard=0.03466
                      accrate=10);
run;
ods graphics off;
```

A Whitehead method creates boundaries that approximately satisfy the Type I and Type II error probability level specification. The BOUNDARYKEY=ALPHA option is used to adjust the boundary value at the last stage and to meet the specified Type I probability level.

The specified ACCRATE=10 option indicates that 10 mice will be accrued each week and the resulting minimum and maximum accrual times are displayed. With the BOUNDARYSCALE=SCORE option, the procedure displays the output boundaries with the score statistics.

The "Design Information" table in Output 78.6.1 displays design specifications and derived statistics.

Output 78.6.1 Design Information

The SEQDESIGN Procedure
Design: OneSidedWhitehead

Design Information

Statistic Distribution	Normal
Boundary Scale	Score
Alternative Hypothesis	Upper
Early Stop	Accept/Reject Null
Method	Whitehead
Boundary Key	Alpha
Alternative Reference	0.69315
Number of Stages	4
Alpha	0.05
Beta	0.20044
Power	0.79956
Max Information (Percent of Fixed Sample)	129.9894
Max Information	16.70624
Null Ref ASN (Percent of Fixed Sample)	62.6302
Alt Ref ASN (Percent of Fixed Sample)	74.00064

The "Boundary Information" table in Output 78.6.2 displays the information level, alternative reference, and boundary values at each stage.

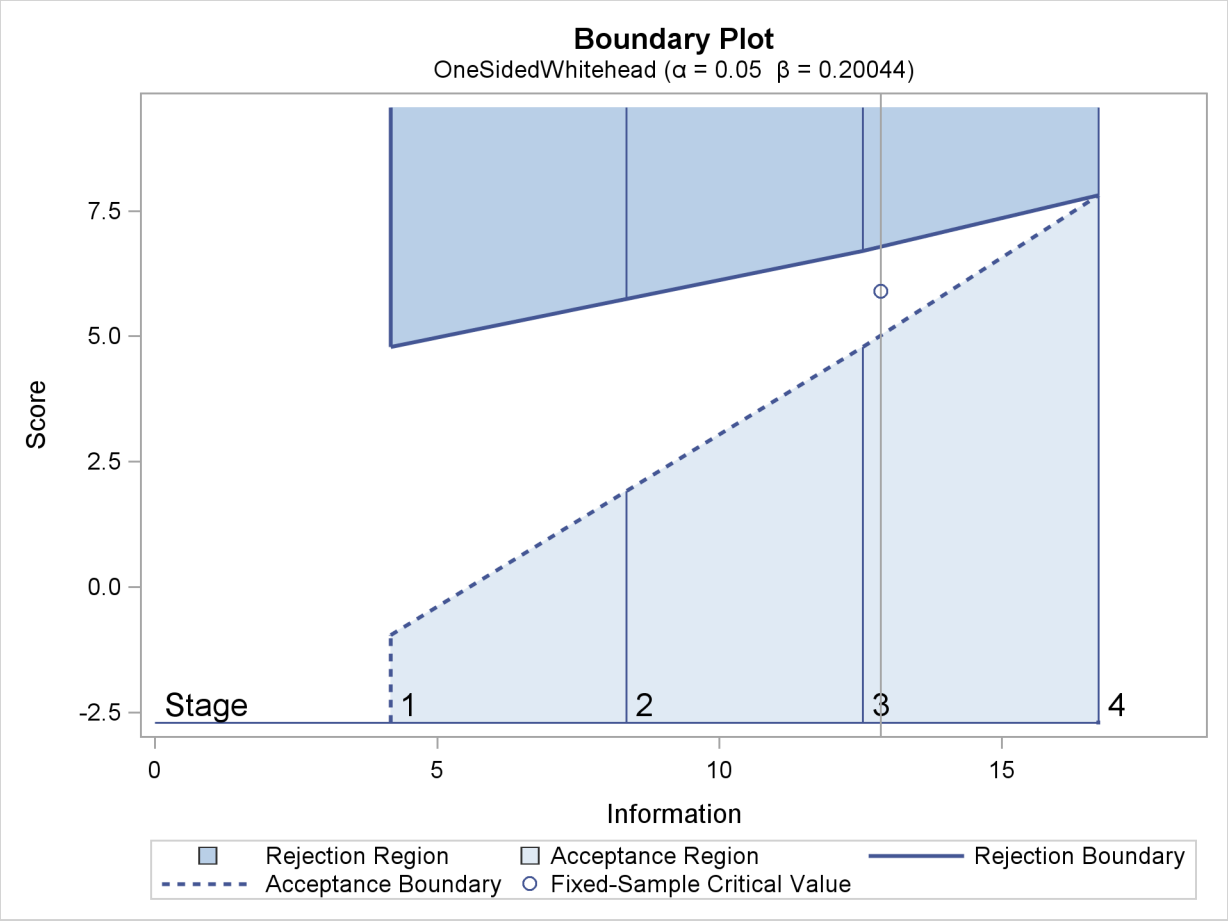
Output 78.6.2 Boundary Information

Boundary Information (Score Scale)
Null Reference = 0

	Information Level			Alternative Reference	Boundary Values	
					Upper	
Stage	Proportion	Actual	Events	Upper	Beta	Alpha
1	0.2500	4.176561	16.70624	2.89498	-0.95755	4.78773
2	0.5000	8.353122	33.41249	5.78997	1.91509	5.74527
3	0.7500	12.52968	50.11873	8.68495	4.78773	6.70282
4	1.0000	16.70624	66.82498	11.57993	7.81296	7.81296

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 78.6.3.

Output 78.6.3 Boundary Plot



With the MODEL=TWOSAMPLESURVIVAL option in the SAMPLESIZE statement, the "Sample Size Summary" table in Output 78.6.4 displays the parameters for the sample size computation.

Output 78.6.4 Required Sample Size Summary

Sample Size Summary

Test	Two-Sample Survival
Null Hazard Rate	0.03466
Hazard Rate (Group A)	0.01733
Hazard Rate (Group B)	0.03466
Hazard Ratio	0.499999
log(Hazard Ratio)	-0.69315
Reference Hazards	Alt Ref
Accrual Rate	10
Min Accrual Time	6.682498
Min Sample Size	66.82498
Max Accrual Time	25.401
Max Sample Size	254.01
Max Number of Events	66.82498

With a minimum accrual time of 6.6825 weeks and a maximum accrual time of 25.401 weeks, an accrual time of 20 weeks is used in the study.

The "Numbers of Events" table in Output 78.6.5 displays the required number of events for the group sequential clinical trial.

Output 78.6.5 Required Numbers of Events

Numbers of Events (D)
Two-Sample Log-Rank Test

Stage	D	Information
1	16.71	4.1766
2	33.41	8.3531
3	50.12	12.5297
4	66.82	16.7062

The following statements invoke the SEQDESIGN procedure and provide more detailed sample size information:

```
proc seqdesign altref=0.69315
               boundaryscale=score
               ;
   OneSidedWhitehead: design method=whitehead
                             nstages=4
                             boundarykey=alpha
                             alt=upper
                             stop=both
                             beta=0.20;
   samplesize model=twosamplesurvival
                    ( nullhazard=0.03466
                      accrate=10 acctime=20);
ods output Boundary=Bnd_Surv;
run;
```

The ODS OUTPUT statement with the BOUNDARY=BND_SURV option creates an output data set named BND_SURV which contains the resulting boundary information for the subsequent sequential tests.

With an accrual time of 20 weeks, the "Sample Size Summary" table in Output 78.6.6 displays the follow-up time for the trial.

Output 78.6.6 Required Sample Size Summary

The SEQDESIGN Procedure
Design: OneSidedWhitehead

Sample Size Summary

Test	Two-Sample Survival
Null Hazard Rate	0.03466
Hazard Rate (Group A)	0.01733
Hazard Rate (Group B)	0.03466
Hazard Ratio	0.499999
log(Hazard Ratio)	−0.69315
Reference Hazards	Alt Ref
Accrual Rate	10
Accrual Time	20
Follow-up Time	6.47422
Total Time	26.47422
Max Number of Events	66.82498
Max Sample Size	200
Expected Sample Size (Null Ref)	161.5937
Expected Sample Size (Alt Ref)	172.4689

The "Numbers of Events and Sample Sizes" table in Output 78.6.7 displays the required sample sizes for the group sequential clinical trial.

Output 78.6.7 Numbers of Events and Sample Sizes

Numbers of Events (D) and Sample Sizes (N)
Two-Sample Log-Rank Test

	Fractional Time						
Stage	D	D(Grp 1)	D(Grp 2)	Time	N	N(Grp 1)	N(Grp 2)
1	16.71	5.82	10.89	11.9866	119.87	59.93	59.93
2	33.41	11.84	21.57	17.3584	173.58	86.79	86.79
3	50.12	18.01	32.11	21.7479	200.00	100.00	100.00
4	66.82	24.46	42.37	26.4742	200.00	100.00	100.00

Numbers of Events (D) and Sample Sizes (N)
Two-Sample Log-Rank Test

	Fractional Time	Ceiling Time					
Stage	Information	D	D(Grp 1)	D(Grp 2)	Time	N	N(Grp 1)
1	4.1766	16.74	5.83	10.91	12	120.00	60.00
2	8.3531	35.73	12.68	23.04	18	180.00	90.00
3	12.5297	51.07	18.37	32.70	22	200.00	100.00
4	16.7062	68.55	25.14	43.41	27	200.00	100.00

Numbers of Events (D) and Sample Sizes (N)
Two-Sample Log-Rank Test

	Ceiling Time	
Stage	N(Grp 2)	Information
1	60.00	4.1854
2	90.00	8.9322
3	100.00	12.7667
4	100.00	17.1377

Thus the study will perform three interim analyses after 12, 18, and 22 weeks and a final analysis after 27 weeks if the study does not stop at any of the interim analyses.

Note that the SEQDESIGN procedure does not compute numbers of events or sample sizes for all statistical models. If the number of events or sample size for a fixed-sample design is available, then the MODEL=INPUTNEVENTS or MODEL=INPUTNOBS option can be used to input fixed-sample information. For example, with a required fixed-sample number of events 51.4073, the following SAMPLESIZE statement can be used to produce the same sample size results:

```
samplesize model=inputnevents
                ( d=51.47 sample=two
                  hazard=0.03466 0.01733
                  accrate=10 acctime=20);
```

Suppose that 120 mice are available after week 12 for the first interim analysis. Output 78.6.8 lists the 10 observations in the data set weeks_1.

Output 78.6.8 Clinical Trial Data

First 10 Obs in the Trial Data

Obs	Trt Gp	Event	Weeks
1	0	0	11
2	1	0	11
3	0	0	11
4	1	0	11
5	0	1	6
6	1	0	11
7	0	0	11
8	1	0	11
9	0	1	9
10	1	0	11

The TrtGp variable is a grouping variable with the value 0 for a mouse in the placebo control group and the value 1 for a mouse in the treatment group. The Weeks variable is the survival time variable measured in weeks and the Event variable is the censoring variable with the value 0 indicating censoring. That is, the values of Weeks are considered censored if the corresponding values of Event are 0; otherwise, they are considered as event times.

The following statements use the LIFETEST procedure to estimate the log-rank statistic at stage 1:

```
proc lifetest data=Surv_1;
   time Weeks*Event(0);
   test TrtGp;
ods output logunichisq=Parms_Surv1;
run;
```

The following statements create and display (in Output 78.6.9) the data set for the log-rank statistic and its associated standard error:

```
data Parms_Surv1;
   set Parms_Surv1(rename=(Statistic=Estimate));
   if Variable='TrtGp';
   _Scale_='Score';
   _Stage_= 1;
   keep Variable _Scale_ _Stage_ StdErr Estimate;
run;

proc print data=Parms_Surv1;
   title 'Statistics Computed at Stage 1';
run;
```

Output 78.6.9 Statistics Computed at Stage 1

Statistics Computed at Stage 1

Obs	Variable	Estimate	StdErr	_Scale_	_Stage_
1	TrtGp	3.2004	1.9979	Score	1

The following statements invoke the SEQTEST procedure to test for early stopping at stage 1:

```
ods graphics on;
proc seqtest Boundary=Bnd_Surv
             Parms(Testvar=TrtGp)=Parms_Surv1
             boundaryscale=score
             ;
ods output Test=Test_Surv1;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 1, which was generated in the SEQDESIGN procedure. The PARMS=PARMS_SURV1 option specifies the input data set PARMS_SURV1 that contains the test statistic and its associated standard error at stage 1, and the TESTVAR=TRTGP option identifies the test variable TRTGP in the data set.

The ODS OUTPUT statement with the TEST=TEST_SURV1 option creates an output data set named TEST_SURV1 which contains the updated boundary information for the test at stage 1. The data set also provides the boundary information that is needed for the group sequential test at the next stage.

The "Design Information" table in Output 78.6.10 displays design specifications. By default, when the boundary values are modified for the new information levels, the Type I α level is maintained. The maximum information and the power have been modified for the new information levels.

Output 78.6.10 Design Information

The SEQTEST Procedure

Design Information

BOUNDARY Data Set	WORK.BND_SURV
Data Set	WORK.PARMS_SURV1
Statistic Distribution	Normal
Boundary Scale	Score
Alternative Hypothesis	Upper
Early Stop	Accept/Reject Null
Number of Stages	4
Alpha	0.05
Beta	0.20007
Power	0.79993
Max Information (Percent of Fixed Sample)	129.854
Max Information	16.7062448
Null Ref ASN (Percent of Fixed Sample)	62.56841
Alt Ref ASN (Percent of Fixed Sample)	74.05609

The "Test Information" table in Output 78.6.11 displays the boundary values for the test statistic with the SCORE statistic scale. Since only the information level at stage 1 is specified in the DATA= data set, the information levels at subsequent stages are derived proportionally from the corresponding information levels in the original design. At stage 1, the score statistic 3.2004 is between the upper β boundary value -1.0407 and the upper α boundary value 4.7142, so the trial continues to the next stage.

Output 78.6.11 Sequential Tests

Test Information (Score Scale)
Null Reference = 0

	---Information Level--		-Alternative- --Reference--	-----Boundary Values---- ----------Upper---------	
Stage	Proportion	Actual	Upper	Beta	Alpha
1	0.2389	3.991698	2.76685	-1.04069	4.71422
2	0.4926	8.22988	5.70454	1.81950	5.72627
3	0.7463	12.46806	8.64224	4.73885	6.68621
4	1.0000	16.70624	11.57993	7.80104	7.80104

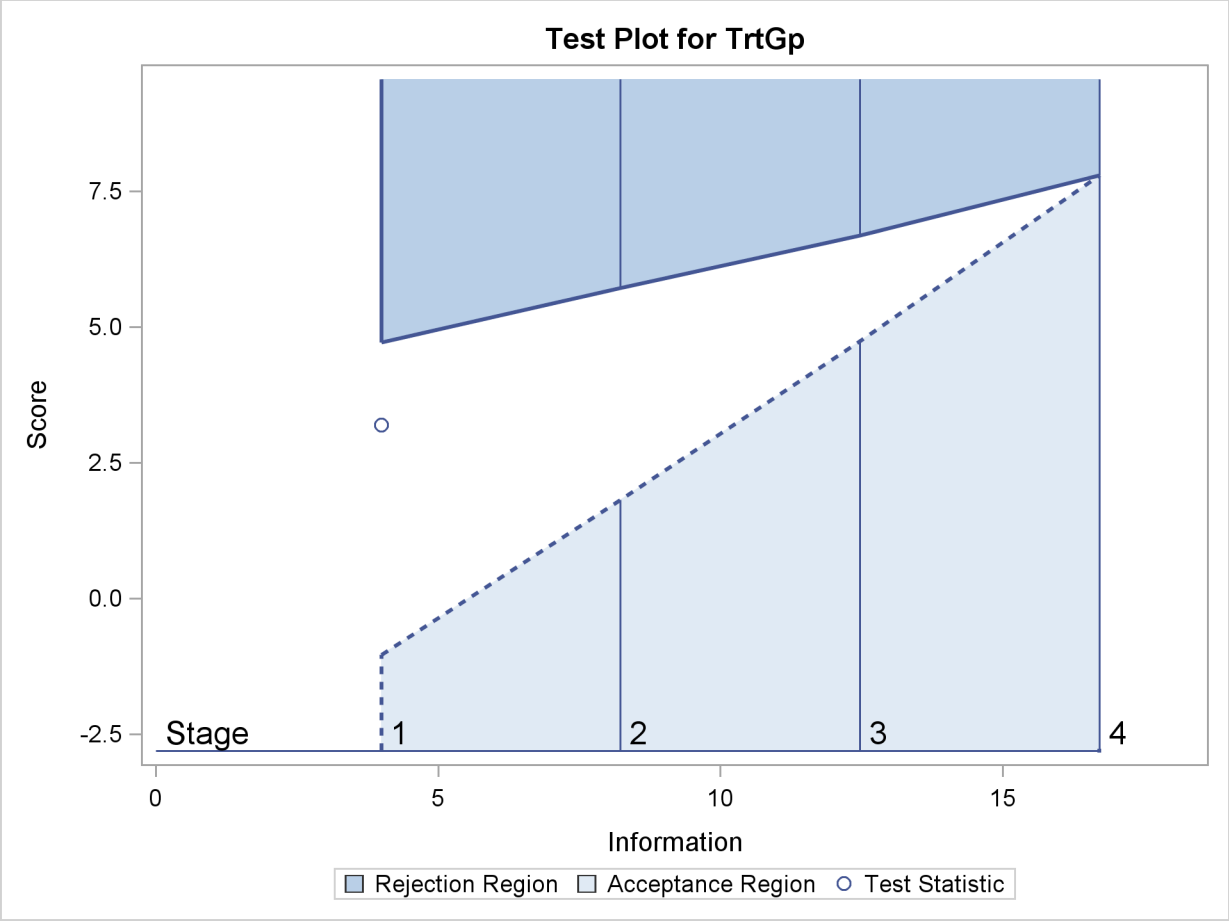
Test Information (Score Scale)
Null Reference = 0

	----------Test---------- ----------TrtGp----------	
Stage	Estimate	Action
1	3.20040	Continue
2	.	
3	.	
4	.	

Note that the observed information level 3.9917 corresponds to a proportion of 0.2389 in information level. If the observed information level is much smaller than the target proportion of 0.25, then you need to increase the accrual rate, accrual time, or follow-up time to achieve the target maximum information level for the trial. Scharfstein and Tsiatis (1998) use the simulation and bootstrap methods to modify the trial at interim stages to achieve the target maximum information level. These modifications should be specified in the study protocol or study plan before the study starts.

With the specified ODS GRAPHICS ON statement, a boundary plot with test statistics is displayed, as shown in Output 78.6.12. As expected, the test statistic is in the continuation region between the upper β and α boundary values.

Output 78.6.12 Sequential Test Plot



Note that the input DATA= option can also be used for the test statistics. For example, the following statements create and display (in Output 78.6.13) the data set for the log-rank statistic and its associated standard error after the LIFETEST procedure. Since the log-rank statistic is a score statistic, the corresponding information level is the variance of the statistic.

```
proc lifetest data=Surv_1;
   time Weeks*Event(0);
   test TrtGp;
ods output logunichisq=Parms_Surv1a;
run;

data Parms_Surv1a;
   set Parms_Surv1a(rename=(Statistic=TrtGp));
   keep _Scale_ _Stage_ _Info_ TrtGp;
   _Scale_='Score';
   _Stage_= 1;
   _Info_= StdErr * StdErr;
   if Variable='TrtGp';
run;
```

```
proc print data=Parms_Surv1a;
   title 'Statistics Computed at Stage 1';
run;
```

Output 78.6.13 Statistics Computed at Stage 1

Statistics Computed at Stage 1

Obs	TrtGp	_Scale_	_Stage_	_Info_
1	3.2004	Score	1	3.99170

The following statements can be used to invoke the SEQTEST procedure to test for early stopping at stage 1:

```
ods graphics on;
proc seqtest Boundary=Bnd_Surv
             Data(Testvar=TrtGp)=Parms_Surv1a
             boundaryscale=score
             ;
ods output Test=Test_Surv1;
run;
ods graphics off;
```

The following statements use the LIFETEST procedure to compute the log-rank statistic and its associated standard error at stage 2:

```
proc lifetest data=Surv_2;
   time Weeks*Event(0);
   test TrtGp;
ods output logunichisq=Parms_Surv2;
run;
```

The following statements create and display (in Output 78.6.14) the data set for the log-rank statistic and its associated standard error for each of the first two stages:

```
data Parms_Surv2;
   set Parms_Surv2 (rename=(Statistic=Estimate));
   if Variable='TrtGp';
   _Scale_='Score';
   _Stage_= 2;
   keep Variable _Scale_ _Stage_ StdErr Estimate;
run;

proc print data=Parms_Surv2;
   title 'Statistics Computed at Stage 2';
run;
```

Output 78.6.14 Statistics Computed at Stage 2

Statistics Computed at Stage 2

Obs	Variable	Estimate	StdErr	_Scale_	_Stage_
1	TrtGp	7.3136	2.9489	Score	2

The following statements invoke the SEQTEST procedure to test for early stopping at stage 2:

```
ods graphics on;
proc seqtest Boundary=Test_Surv1
             Parms(Testvar=TrtGp)=Parms_Surv2
             boundaryscale=score
             citype=lower
             ;
ods output Test=Test_Surv2;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 2, which was generated by the SEQTEST procedure at the previous stage. The PARMS= option specifies the input data set that contains the test statistic and its associated standard error at stage 2, and the TESTVAR= option identifies the test variable in the data set.

The ODS OUTPUT statement with the TEST=TEST_SURV2 option creates an output data set named TEST_SURV2 which contains the updated boundary information for the test at stage 2. The data set also provides the boundary information that is needed for the group sequential test at the next stage if the trial does not stop at the current stage.

The "Test Information" table in Output 78.6.15 displays the boundary values for the test statistic. The test statistic 7.31365 is larger than the corresponding upper α boundary 5.79886, so the study stops and rejects the null hypothesis. That is, there is evidence of reduction in hazard rate for the new treatment.

Output 78.6.15 Sequential Tests

The SEQTEST Procedure

Test Information (Score Scale)
Null Reference = 0

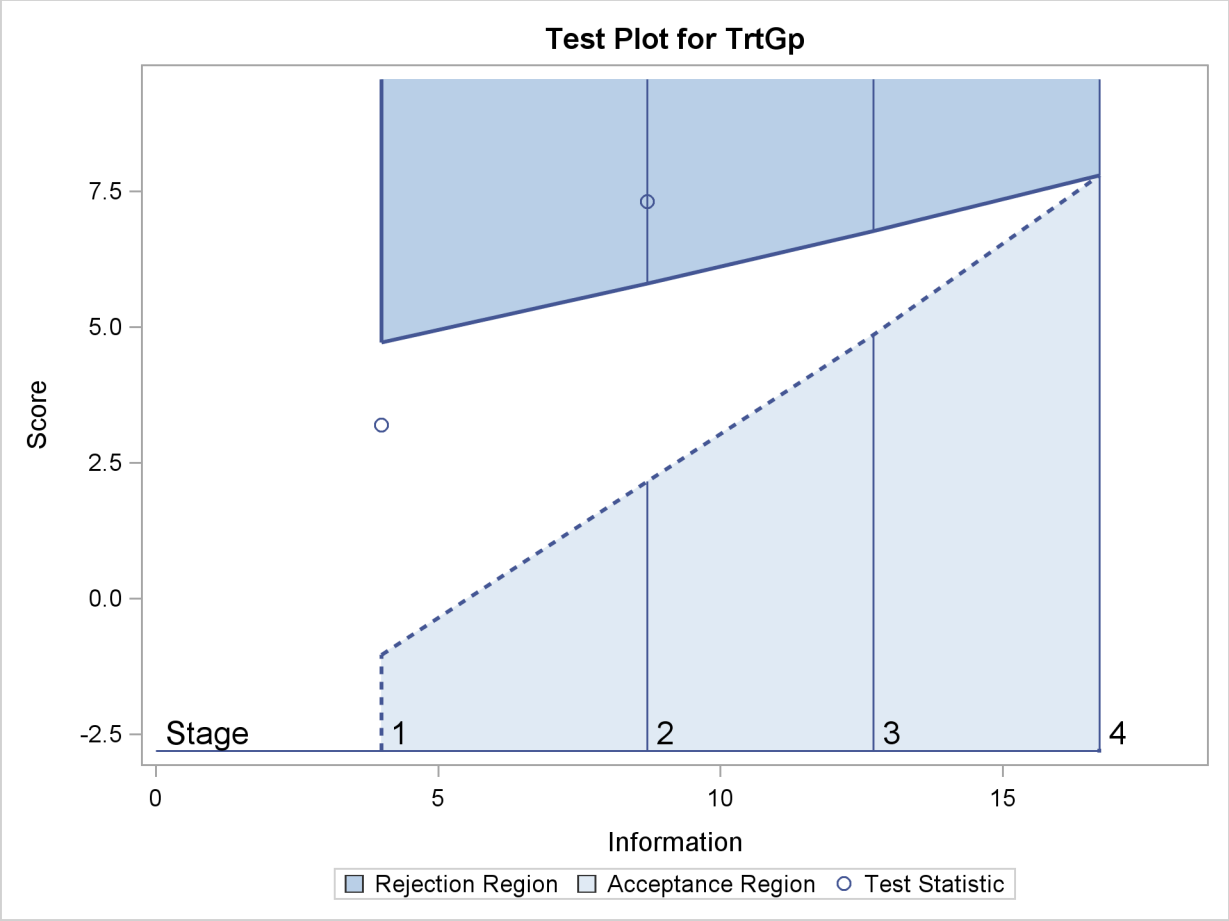
	---Information Level--		-Alternative- --Reference--	-----Boundary Values---- ----------Upper---------	
Stage	Proportion	Actual	Upper	Beta	Alpha
1	0.2389	3.991698	2.76685	-1.04069	4.71422
2	0.5205	8.696125	6.02772	2.15758	5.79895
3	0.7603	12.70118	8.80383	4.85836	6.77006
4	1.0000	16.70624	11.57993	7.79713	7.79713

Test Information (Score Scale)
Null Reference = 0

	----------Test---------- ----------TrtGp----------	
Stage	Estimate	Action
1	3.20040	Continue
2	7.31365	Reject Null
3	.	
4	.	

With the specified ODS GRAPHICS ON statement, the "Test Plot" displays boundary values of the design and the test statistic at the first two stages, as shown in Output 78.6.16. It also shows that the test statistic is in the "Rejection Region" above the upper α boundary value at stage 2.

Output 78.6.16 Sequential Test Plot



After the stopping of a trial, the "Parameter Estimates" table in Output 78.6.17 displays the stopping stage, parameter estimate, unbiased median estimate, confidence limits, and p-value under the null hypothesis $H_0 : \theta = 0$.

Output 78.6.17 Parameter Estimates

Parameter Estimates
Stagewise Ordering

Parameter	Stopping Stage	MLE	p-Value for H0:Parm=0	Median Estimate	Lower 95% CL
TrtGp	2	0.841024	0.0139	0.810328	0.21615

As expected, the p-value 0.0139 is significant at the $\alpha = 0.05$ level and the lower 95% confidence limit is larger than $\theta_0 = 0$. The p-value, unbiased median estimate, and lower confidence limit depend on the ordering of the sample space (k, z), where k is the stage number and z is the standardized Z statistic. With the specified stagewise ordering, the p-value is $p_1 + p_2$, where p_1 is the

α spending at stage 1,

$$p_1 = P_{\theta=0}(Z_1\sqrt{I_1} \geq 4.71422) = 0.00915$$

$$p_2 = P_{\theta=0}(Z_2\sqrt{I_2} \geq 7.31365 \mid -1.04069 < Z_1\sqrt{I_1} < 4.71422)$$

where Z_k is a standardized normal variate and I_k is the information level at stage k for $k = 1, 2$.

Example 78.7: Testing an Effect in a Proportional Hazards Regression Model

This example compares two survival distributions for the treatment effect. The example uses a power family method to generate two-sided asymmetric boundaries and then uses a proportional hazards regression model to test the hypothesis with a covariate.

A clinic is conducting a clinical study for the effect of a new cancer treatment. The study consists of mice exposed to a carcinogen and randomized to either the control group or the treatment group. The event of interest is the death from cancer induced by the carcinogen, and the response is the time from randomization to death.

Consider the proportional hazards regression model

$$h(t;\ \text{TrtGp},\ \text{Wgt}) = h_0(t)\ \exp(\beta_g\ \text{TrtGp}\ +\ \beta_w \text{Wgt})$$

where $h_0(t)$ is an arbitrary and unspecified baseline hazard function, TrtGp is the grouping variable for the two groups, Wgt is the initial weight of the mice, and β_g and β_w are the regression parameters associated with the variables TrtGp and Wgt, respectively. The grouping variable has the value 0 for each mouse in the control group and the value 1 for each mouse in the treatment group.

The hypothesis $H_0 : \beta_g = 0$ with an alternative hypothesis $H_1 : \beta_g \neq 0$ is used for the study.

Suppose that from past experience, the median survival time for the control group is $t_0 = 20$ weeks. The study would like to detect a $t_1 = 40$ weeks median survival time with a 80% power in the trial. Assuming exponential survival functions for the two groups, the hazard rates can be computed from

$$S_j(t_j) = e^{-h_j t_j} = \frac{1}{2}$$

where $j = 0, 1$.

Thus, with the hazard rates $h_0 = 0.03466$ and $h_1 = 0.01733$, the hazard ratio $\exp(\beta_g) = h_1/h_0 = 1/2$ and the alternative hypothesis

$$\beta_{g1} = \log(\frac{1}{2}) = -0.69315$$

Following the derivations in the section "Test for a Parameter in the Proportional Hazards Regression Model" in the chapter "The SEQDESIGN Procedure," the required number of events for testing a parameter in $\boldsymbol{\beta}$ is given by

$$D_X = I_X\ \frac{1}{(1 - r_x^2)\ \sigma_x^2}$$

where σ_x^2 is the variance of TrtGp and r_x^2 is the proportion of variance of TrtGp explained by the variable Wgt.

If the two groups have the same number of mice in the study, then the MLE of the variance is $\hat{\sigma}_x^2 = 0.25$. Further, if $r_x^2 = 0.10$, then you can specify the MODEL=PHREG(XVARIANCE=0.25 XRSQUARE=0.10) option in the SAMPLESIZE statement in the SEQDESIGN procedure to compute the required number of events and the individual number of events at each stage.

The following statements invoke the SEQDESIGN procedure and request a four-stage group sequential design for normally distributed data. The design uses a two-sided alternative hypothesis with early stopping to reject the null hypothesis H_0. A power family method is used to derive the boundaries.

```
ods graphics on;
proc seqdesign altref=0.69315;
   TwoSidedPowerFamily: design method=pow
                               nstages=4
                               alpha=0.075(lower=0.025)
                               beta=0.20;
   samplesize model=phreg( xvariance=0.25 xrsquare=0.10
                           hazard=0.02451 accrate=10);
run;
ods graphics off;
```

The ALPHA=0.075(LOWER=0.025) option specifies a lower α level 0.025 for the lower rejection boundary and an upper α level $0.05 = 0.075 - 0.025$ for the upper rejection boundary. The geometric average hazard $\sqrt{h_0 \times h_1} = \sqrt{0.03466 \times 0.01733} = 0.02451$ is used in the HAZARD= option in the SAMPLESIZE statement to compute the required sample size. The specified ACCRATE=10 option indicates that 10 mice will be accrued each week and the resulting minimum and maximum accrual times will be displayed.

The "Design Information" table in Output 78.7.1 displays the design specifications and the derived statistics.

Output 78.7.1 Design Information

The SEQDESIGN Procedure
Design: TwoSidedPowerFamily

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Reject Null
Method	Power Family
Boundary Key	Both
Alternative Reference	0.69315
Number of Stages	4
Alpha	0.075
Alpha (Lower)	0.025
Alpha (Upper)	0.05
Beta (Lower)	0.2
Beta (Upper)	0.12764
Power (Lower)	0.8
Power (Upper)	0.87236
Max Information (Percent of Fixed Sample)	106.468
Max Information	17.39288
Null Ref ASN (Percent of Fixed Sample)	104.3691
Lower Alt Ref ASN (Number of Events)	58.04014
Upper Alt Ref ASN (Number of Events)	52.05395

The "Boundary Information" table in Output 78.7.2 displays the information level, alternative reference, and boundary values at each stage. With the default BOUNDARYSCALE=STDZ option, the procedure displays the output boundaries with the standardized Z statistic.

Output 78.7.2 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

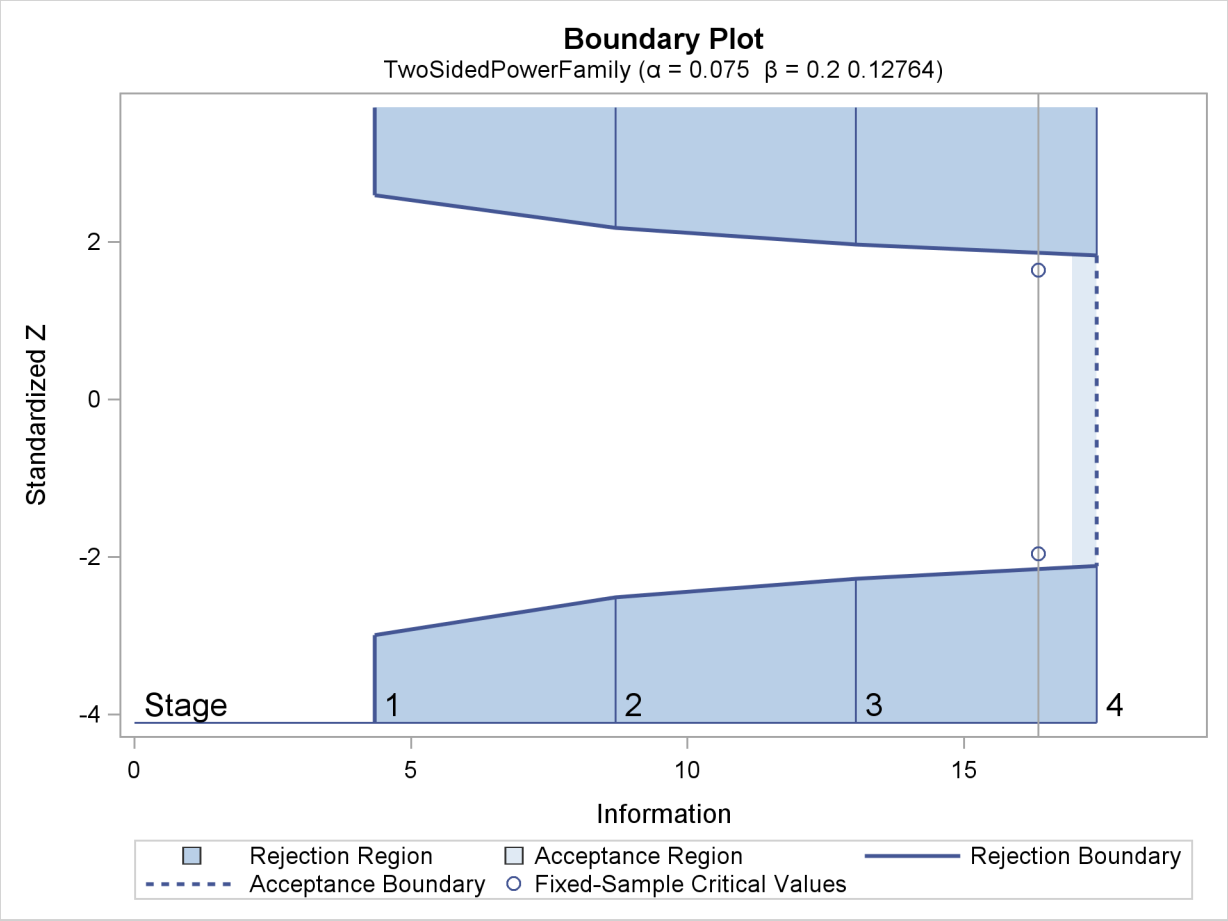
Stage	Information Level: Proportion	Information Level: Actual	Information Level: Events	Alternative Reference: Lower	Alternative Reference: Upper
1	0.2500	4.348221	19.32543	−1.44538	1.44538
2	0.5000	8.696441	38.65085	−2.04408	2.04408
3	0.7500	13.04466	57.97628	−2.50348	2.50348
4	1.0000	17.39288	77.3017	−2.89077	2.89077

Boundary Information (Standardized Z Scale)
Null Reference = 0

Stage	Boundary Values: Lower Alpha	Boundary Values: Upper Alpha
1	−2.98871	2.59149
2	−2.51320	2.17917
3	−2.27093	1.96910
4	−2.11334	1.83246

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed, as shown in Output 78.7.3.

Output 78.7.3 Boundary Plot



With the MODEL=PHREG option in the SAMPLESIZE statement, the "Sample Size Summary" table in Output 78.7.4 displays the parameters used in the sample size computation for the proportional hazards regression model.

Output 78.7.4 Required Sample Size Summary

Sample Size Summary

Test	PH Reg Parameter
Parameter	0.69315
X Variance	0.25
R Square (X)	0.1
Hazard Rate	0.02451
Accrual Rate	10
Min Accrual Time	7.73017
Min Sample Size	77.3017
Max Accrual Time	27.97872
Max Sample Size	279.7872
Max Number of Events	77.3017

With a minimum accrual time of 7.73 weeks and maximum accrual time of 27.98 weeks, an accrual time of 20 weeks is used in the study. The "Numbers of Events" table in Output 78.7.5 displays the required numbers of events for the group sequential clinical trial.

Output 78.7.5 Required Sample Sizes

Numbers of Events (D)
Z Test for PH Regression Parameter

Stage	D	Information
1	19.33	4.3482
2	38.65	8.6964
3	57.98	13.0447
4	77.30	17.3929

The following statements invoke the SEQDESIGN procedure and provide more detailed sample size information with a 20-week accrual time:

```
proc seqdesign altref=0.69315;
   TwoSidedPowerFamily: design method=pow
                               nstages=4
                               alpha=0.075(lower=0.025)
                               beta=0.20;
   samplesize model=phreg( xvariance=0.25 xrsquare=0.10
                           hazard=0.02451
                           accrate=10 acctime=20);
ods output Boundary=Bnd_Time;
run;
```

The ODS OUTPUT statement with the BOUNDARY=BND_TIME option creates an output data set named BND_TIME which contains the resulting boundary information for the subsequent sequential tests.

With an accrual time of 20 weeks, the "Sample Size Summary" table in Output 78.7.6 displays the follow-up time for the trial.

Output 78.7.6 Sample Size Summary

The SEQDESIGN Procedure
Design: TwoSidedPowerFamily

Sample Size Summary

Test	PH Reg Parameter
Parameter	0.69315
X Variance	0.25
R Square (X)	0.1
Hazard Rate	0.02451
Accrual Rate	10
Accrual Time	20
Follow-up Time	10.34195
Total Time	30.34195
Max Number of Events	77.3017
Max Sample Size	200
Expected Sample Size (Null Ref)	199.4282
Expected Sample Size (Alt Ref)	188.6561

The "Numbers of Events and Sample Sizes" table in Output 78.7.7 displays the required sample sizes for the group sequential clinical trial.

Output 78.7.7 Numbers of Events and Sample Sizes

Numbers of Events (D) and Sample Sizes (N)
Z Test for PH Regression Parameter

	Fractional Time			
Stage	D	Time	N	Information
1	19.33	13.2362	132.36	4.3482
2	38.65	19.1466	191.47	8.6964
3	57.98	24.3744	200.00	13.0447
4	77.30	30.3420	200.00	17.3929

Numbers of Events (D) and Sample Sizes (N)
Z Test for PH Regression Parameter

	Ceiling Time			
Stage	D	Time	N	Information
1	21.49	14	140.00	4.8359
2	41.90	20	200.00	9.4281
3	60.14	25	200.00	13.5309
4	79.26	31	200.00	17.8346

Thus, the study will perform three interim analyses after 14, 20, and 25 weeks and a final analysis after 31 weeks if the study does not stop at any of the interim analyses.

Suppose 140 mice are available for the first interim analysis after week 14. Output 78.7.8 lists the first 10 observations in the data set weeks_1.

Output 78.7.8 Clinical Trial Data

First 10 Obs in the Trial Data

Obs	Trt Gp	Event	Wgt	Weeks
1	0	0	22.1659	12
2	1	0	28.4458	12
3	0	0	26.2857	12
4	1	0	25.0283	12
5	0	0	21.5114	12
6	1	0	23.2240	12
7	0	1	22.6845	6
8	1	0	27.9292	12
9	0	0	22.5514	12
10	1	1	27.3793	11

The TrtGp variable is a grouping variable with the value 0 for a mouse in the placebo control group and the value 1 for a mouse in the treatment group.

The Weeks variable is the survival time variable measured in weeks and the Event variable is the censoring variable with the value 0 indicating censoring. That is, the values of Weeks are considered censored if the corresponding values of Event are 0; otherwise, they are considered as event times.

The following statements use the PHREG procedure to estimate the treatment effect after adjusting for the Wgt variable at stage 1:

```
proc phreg data=Time_1;
   model Weeks*Event(0)= TrtGp Wgt;
ods output parameterestimates=Parms_Time1;
run;
```

The following statements create and display (in Output 78.7.9) the data set for the treatment effect MLE statistic and its associated standard error. Note that for a MLE statistic, the inverse of the variance of the statistic is the information.

```
data Parms_Time1;
   set Parms_Time1;
   if Parameter='TrtGp';
   _Scale_='MLE';
   _Stage_= 1;
   keep _Scale_ _Stage_ Parameter Estimate StdErr;
run;

proc print data=Parms_Time1;
   title 'Statistics Computed at Stage 1';
run;
```

Output 78.7.9 Statistics Computed at Stage 1

Statistics Computed at Stage 1

Obs	Parameter	Estimate	StdErr	_Scale_	_Stage_
1	TrtGp	0.00836	0.46588	MLE	1

The following statements invoke the SEQTEST procedure to test for early stopping at stage 1:

```
ods graphics on;
proc seqtest Boundary=Bnd_Time
             Parms(Testvar=TrtGp)=Parms_Time1
             order=lr
             ;
ods output Test=Test_Time1;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 1, which was generated in the SEQDESIGN procedure. The PARMS=PARMS_TIME1 option specifies the input data set PARMS_TIME1 that contains the test statistic and its associated standard error at stage 1, and the TESTVAR=TRTGP option identifies the test variable TRTGP in the data set.

The ORDER=LR option uses the LR ordering to derive the p-value, the unbiased median estimate, and the confidence limits for the regression slope estimate.

The ODS OUTPUT statement with the TEST=TEST_TIME1 option creates an output data set named TEST_TIME1 which contains the updated boundary information for the test at stage 1. The data set also provides the boundary information that is needed for the group sequential test at the next stage.

The "Design Information" table in Output 78.7.10 displays design specifications. By default, when the boundary values are modified for the new information levels, the Type I α level is maintained. The maximum information and the power have been modified for the new information levels.

Output 78.7.10 Design Information

The SEQTEST Procedure

Design Information

BOUNDARY Data Set	WORK.BND_TIME
Data Set	WORK.PARMS_TIME1
Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Reject Null
Number of Stages	4
Alpha	0.075
Alpha (Lower)	0.025
Alpha (Upper)	0.05
Beta (Lower)	0.20048
Beta (Upper)	0.12795
Power (Lower)	0.79952
Power (Upper)	0.87205
Max Information (Percent of Fixed Sample)	106.5982
Max Information	17.3928828
Null Ref ASN (Percent of Fixed Sample)	104.4715
Lower Alt Ref ASN (Percent of Fixed Sample)	79.7886
Upper Alt Ref ASN (Percent of Fixed Sample)	71.53877

The "Test Information" table in Output 78.7.11 displays the boundary values for the test statistic with the MLE statistic scale.

Output 78.7.11 Sequential Tests

Test Information (Standardized Z Scale)
Null Reference = 0

	--Information Level-		------Alternative----- ------Reference------		----Boundary Values--- ---Lower--	
Stage	Proportion	Actual	Lower	Upper	Alpha	---Upper-- Alpha
1	0.2649	4.607347	-1.48783	1.48783	-2.92457	2.54086
2	0.5099	8.869192	-2.06428	2.06428	-2.50505	2.17290
3	0.7550	13.13104	-2.51175	2.51175	-2.27093	1.96941
4	1.0000	17.39288	-2.89077	2.89077	-2.11635	1.83531

Test Information (Standardized Z Scale)
Null Reference = 0

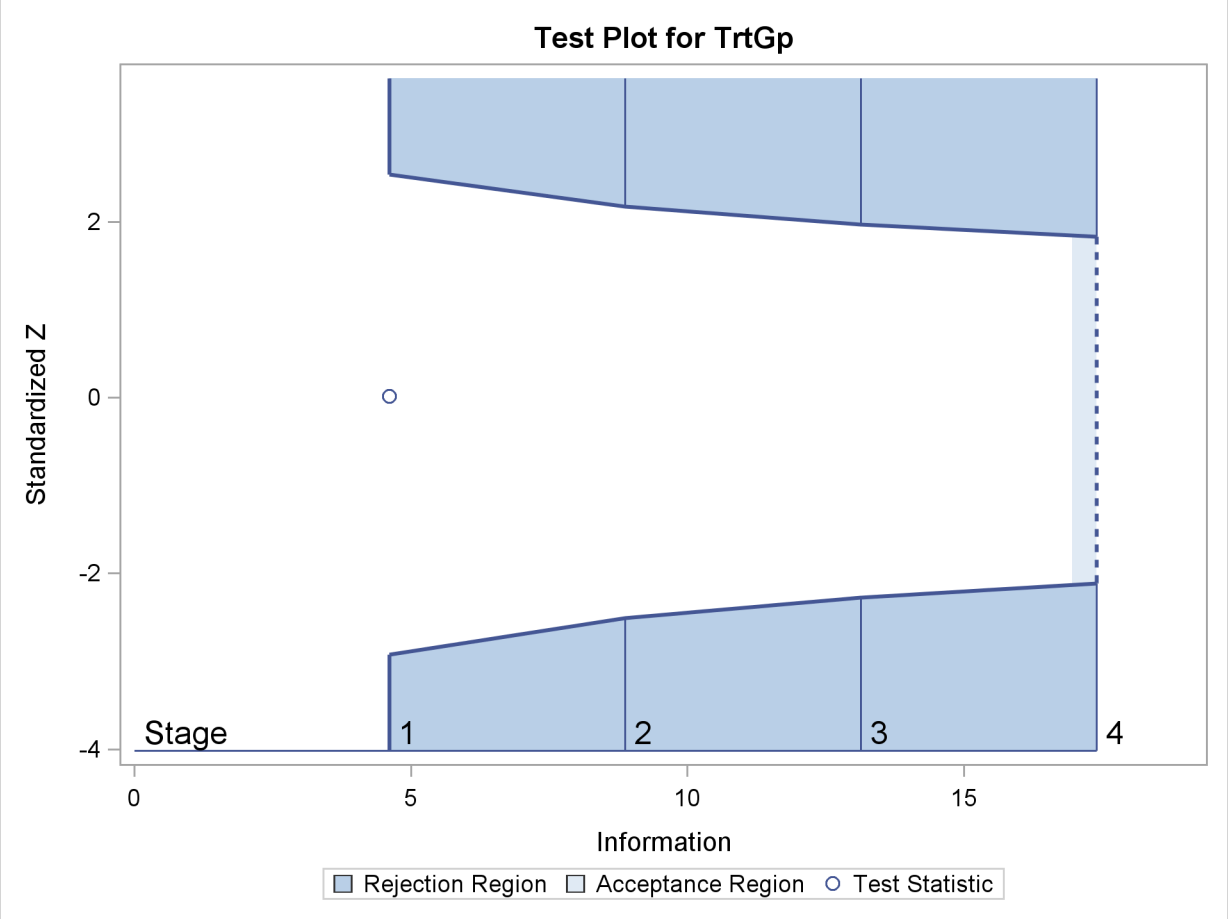
	----------Test---------- ----------TrtGp----------	
Stage	Estimate	Action
1	0.01795	Continue
2	.	
3	.	
4	.	

Since only the information level at stage 1 is derived in the PARMS= data set, the information levels at subsequent stages are derived proportionally from the corresponding information levels in the original design. At stage 1, the standardized Z statistic 0.01795 is between the lower and upper α boundary values of -2.92457 and 2.54086, so the trial continues to the next stage.

Note that the observed information level 4.6073 corresponds to a proportion of 0.2649 in the information level. If the observed information level is much smaller than the target proportion of 0.25, then you need to increase the accrual rate, accrual time, or follow-up time to achieve target information levels for subsequent stages. These modifications should be specified in the study plan before the study begins.

With the specified ODS GRAPHICS ON statement, a boundary plot with test statistics is displayed by default, as shown in Output 78.7.12. As expected, the test statistic is in the continuation region between the lower and upper α boundary values.

Output 78.7.12 Sequential Test Plot



The following statements use the PHREG procedure to compute the MLE statistic and its associated standard error at stage 2:

```
proc phreg data=Time_2;
   model Weeks*Event(0)= TrtGp Wgt;
```

```
ods output parameterestimates= Parms_Time2;
run;
```

The following statements create the data set for the MLE statistic and its associated standard error at stage 2:

```
data Parms_Time2;
   set Parms_Time2;
   if Parameter='TrtGp';
   _Scale_='MLE';
   _Stage_= 2;
   keep _Scale_ _Stage_ Parameter Estimate StdErr;
run;
```

The following statements invoke the SEQTEST procedure to test for early stopping at stage 2:

```
proc seqtest Boundary=Test_Time1
             Parms(Testvar=TrtGp)=Parms_Time2
             order=lr
             ;
ods output Test=Test_Time2;
run;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 2, which was generated by the SEQTEST procedure at the previous stage. The PARMS= option specifies the input data set that contains the test statistic and its associated standard error at stage 2, and the TESTVAR= option identifies the test variable in the data set.

The ODS OUTPUT statement with the TEST=TEST_TIME2 option creates an output data set named TEST_TIME2 which contains the updated boundary information for the test at stage 2. The data set also provides the boundary information that is needed for the group sequential test at the next stage.

The "Test Information" table in Output 78.7.13 displays the boundary values for the test statistic with the MLE statistic scale. At stage 2, the standardized Z statistic -0.43552 is between the lower α and upper boundary values, -2.47689 and 2.14819, respectively, so the trial continues to the next stage.

Output 78.7.13 Sequential Tests

The SEQTEST Procedure

Test Information (Standardized Z Scale)
Null Reference = 0

Stage	--Information Level- Proportion	Actual	------Alternative----- -------Reference------ Lower	Upper	----Boundary Values--- ---Lower-- Alpha	---Upper-- Alpha
1	0.2649	4.607347	-1.48783	1.48783	-2.92457	2.54086
2	0.5251	9.132918	-2.09475	2.09475	-2.47689	2.14819
3	0.7625	13.2629	-2.52433	2.52433	-2.26878	1.96770
4	1.0000	17.39288	-2.89077	2.89077	-2.12017	1.83880

Test Information (Standardized Z Scale)
Null Reference = 0

Stage	----------Test---------- ----------TrtGp---------- Estimate	Action
1	0.01795	Continue
2	-0.43552	Continue
3	.	
4	.	

Since the data set PARMS_Time2 contains the test information only at stage 2, the information level at stage 1 in the TEST_Time1 data set is used to generate boundary values for the test.

Similarly, the test statistic at stage 3 is also between its corresponding lower and upper α boundary values. The trial continues to the next stage.

The following statements use the PHREG procedure to compute the MLE statistic and its associated standard error at the final stage:

```
proc phreg data=Time_4;
   model Weeks*Event(0)= TrtGp Wgt;
ods output parameterestimates= Parms_Time4;
run;
```

The following statements create and display (in Output 78.7.14) the data set for the MLE statistic and its associated standard error at each stage of the study:

```
data Parms_Time4;
   set Parms_Time4;
   if Parameter='TrtGp';
   _Scale_='MLE';
   _Stage_= 4;
   keep _Scale_ _Stage_ Parameter Estimate StdErr;
run;

proc print data=Parms_Time4;
   title 'Statistics Computed at Stage 4';
run;
```

Output 78.7.14 Statistics Computed at Stage 4

Statistics Computed at Stage 4

Obs	Parameter	Estimate	StdErr	_Scale_	_Stage_
1	TrtGp	-0.04451	0.23971	MLE	4

The following statements invoke the SEQTEST procedure to test the hypothesis at stage 4:

```
ods graphics on;
proc seqtest Boundary=Test_Time3
             Parms(Testvar=TrtGp)=Parms_Time4
             order=lr
             ;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 4, which was generated by the SEQTEST procedure at the previous stage. The PARMS= option specifies the input data set that contains the test statistic and its associated standard error at stage 4, and the TESTVAR= option identifies the test variable in the data set.

The "Test Information" table in Output 78.7.15 displays the boundary values for the test statistic. The standardized test statistic -0.1857 is between the lower and upper α boundary values of -2.10447 and 1.82112, respectively, so the study stops and accepts the null hypothesis. That is, there is no evidence of reduction in hazard rate for the new treatment.

Output 78.7.15 Sequential Tests

The SEQTEST Procedure

Test Information (Standardized Z Scale)
Null Reference = 0

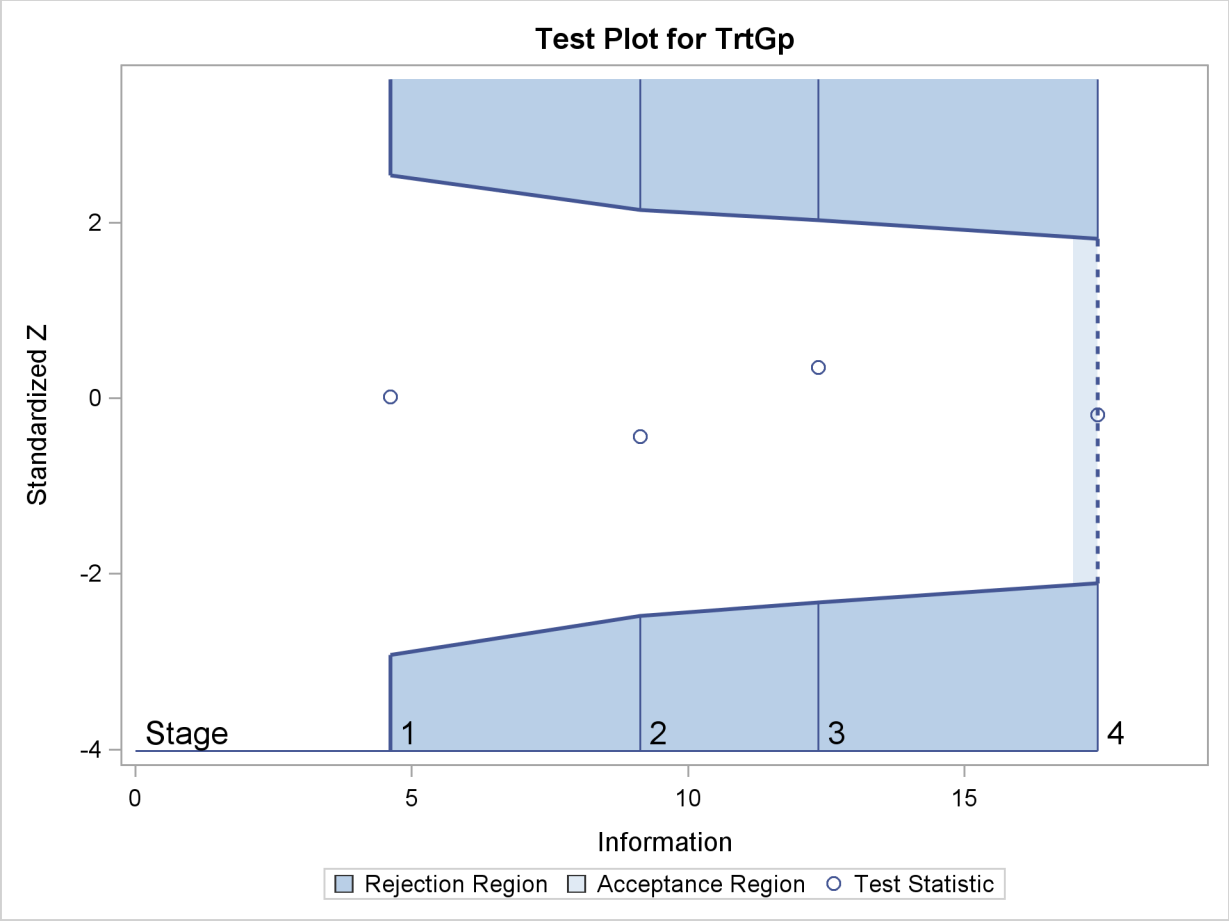
Stage	--Information Level- Proportion	Actual	------Alternative----- -------Reference------ Lower	Upper	----Boundary Values--- ---Lower-- Alpha	---Upper-- Alpha
1	0.2647	4.607347	-1.48783	1.48783	-2.92457	2.54086
2	0.5248	9.132918	-2.09475	2.09475	-2.47689	2.14819
3	0.7095	12.34753	-2.43566	2.43566	-2.32705	2.02634
4	1.0000	17.40274	-2.89159	2.89159	-2.10447	1.82112

Test Information (Standardized Z Scale)
Null Reference = 0

Stage	----------Test---------- ----------TrtGp---------- Estimate	Action
1	0.01795	Continue
2	-0.43552	Continue
3	0.34864	Continue
4	-0.18570	Accept Null

The "Test Plot" displays boundary values of the design and the test statistic at the first two stages, as shown in Output 78.7.16. It also shows that the test statistic is in the "Acceptance Region" between the lower and upper α boundary values at stage 4.

Output 78.7.16 Sequential Test Plot



After the stopping of a trial, the "Parameter Estimates" table in Output 78.7.17 displays the stopping stage, parameter estimate, unbiased median estimate, confidence limits, and p-value under the null hypothesis $H_0 : \theta = 0$.

Output 78.7.17 Parameter Estimates

Parameter Estimates
LR Ordering

Parameter	Stopping Stage	MLE	p-Value for H0:Parm=0	Median Estimate
TrtGp	4	-0.044514	0.8525	-0.044577

Parameter Estimates
LR Ordering

Parameter	95% Confidence Limits	
TrtGp	-0.51461	0.42538

As expected, the two-sided p-value 0.8525 is not significant at the lower $\alpha = 0.025$ level and the upper $\alpha = 0.05$ level, and the two-sided 95% confidence interval contains the null value zero. The p-value, unbiased median estimate, and lower confidence limit depend on the ordering of the sample space (k, z), where k is the stage number and z is the standardized Z statistic. With the specified LR ordering, the two-sided p-value is derived from the one-sided p-value

$$p_u = \sum_{k=1}^{4} P_{\theta=0}\left(Z_k \geq z_4 \mid \_a_{k'} < Z_{k'} < a_{k'}, k' < k\right)$$

where $z_4 = -0.1857$ is the observed test statistic at stage 4, Z_k is a standardized normal variate at stage k, and $\_a_{k'}$ and $a_{k'}$ are the stage k lower and upper rejection boundary values, respectively.

Thus,

$$p_u = \alpha_u + P_{\theta=0}\left(z_4 \leq Z_4 < a_4 \mid \_a_{k'} < Z_{k'} < a_{k'}, k' < 4\right)$$

where $\alpha_u = 0.05$ is the upper α level and $a_4 = 1.82112$.

Since $P_{\theta=0}\left(z_4 \leq Z_4 \leq a_4 \mid \_a_{k'} < Z_{k'} < a_{k'}, k' < 4\right) = 0.52374$, $p_u = 0.05 + 0.52374 = 0.57374$, which is greater than 0.50. Thus, the two-sided p-value is given by $2 \times (1.0 - p_u) = 0.8525$.

Example 78.8: Testing an Effect in a Logistic Regression Model

This example requests a two-sided test for the dose effect in a dose-response model (Whitehead 1997, pp. 262–263). Consider the logistic regression model

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1 \,\text{LDose}$$

where $p = \text{Prob}(\text{Resp} = 1 \,|\, \text{LDose})$ is the response probability to be modeled for the binary response Resp and LDose = log(Dose +1) is the covariate. The dose levels are 0 for the control group, and they are 1, 3, and 6 for the three treatment groups.

Following the derivations in the section "Test for a Parameter in the Logistic Regression Model" in the chapter "The SEQDESIGN Procedure," the required sample size can be derived from

$$N = I_X \, \frac{\sigma_y^2}{(1 - r_x^2)\,\sigma_x^2}$$

where σ_y^2 is the variance of the response variable in the logistic regression model, r_x^2 is the proportion of variance of LDose explained by other covariates, and σ_x^2 is the variance of LDose.

Since LDose is the only covariate in the model, $r_x^2 = 0$. For a logistic model, the variance σ^2 can be estimated by

$$\sigma_y^2 = \frac{1}{\hat{p}(1-\hat{p})}$$

where $\hat{p}$ is the estimated probability of the response variable Resp. Thus, the sample size can be computed as

$$N = I_X \frac{1}{p(1-p)} \frac{1}{\sigma_x^2}$$

The null hypothesis $H_0 : \beta_1 = 0$ corresponds to no treatment effect. Suppose that the alternative hypothesis $H_1 : \beta_1 = 0.5$ is the reference improvement that should be detected at a 0.90 level.

Note that $\beta_1 = 0.5$ corresponds to an odds ratio of 2 between the treatment group with dose level 3 and the control group. The log odds ratio between the two groups is

$$\log\left(\frac{p_t(1-p_c)}{(1-p_t)p_0}\right) = \log\left(\frac{p_t}{1-p_t}\right) - \log\left(\frac{p_c}{1-p_c}\right)$$

which corresponds to

$$(\beta_0 + \beta_1 \log(3+1)) - (\beta_0 + \beta_1 \log(1)) = \beta_1 \log(4) = \log(2)$$

If the same number of patients are assigned in each of the four groups, then the MLE of the variance of LDose is $\hat{\sigma}_x^2 = 0.5345$. Further, if the response rate is 0.40, then the required sample size can be derived using the SAMPLESIZE statement in the SEQDESIGN procedure.

The following statements invoke the SEQDESIGN procedure and request a three-stage group sequential design for normally distributed data. The design has a null hypothesis of no treatment effect $H_0 : \beta_1 = 0$ with early stopping to reject the null hypothesis with a two-sided alternative hypothesis $H_1 : \beta_1 = \pm 0.5$.

```
ods graphics on;
proc seqdesign altref=0.5;
   TwoSidedErrorSpending: design method=errfuncpow
                                method(loweralpha)=errfuncpow(rho=1)
                                method(upperalpha)=errfuncpow(rho=3)
                                nstages=3
                                stop=both;
   samplesize model=logistic( prop=0.4 xvariance=0.5345);
ods output Boundary=Bnd_Dose;
run;
ods graphics off;
```

The ODS OUTPUT statement with the BOUNDARY=BND_DOSE option creates an output data set named BND_DOSE which contains the resulting boundary information for the subsequent sequential tests.

The "Design Information" table in Output 78.8.1 displays design specifications and derived statistics. Since the alternative reference is specified, the maximum information 47.22445 is derived.

Output 78.8.1 Error Spending Design Information

The SEQDESIGN Procedure
Design: TwoSidedErrorSpending

Design Information

Statistic Distribution	Normal
Boundary Scale	Standardized Z
Alternative Hypothesis	Two-Sided
Early Stop	Accept/Reject Null
Method	Error Spending
Boundary Key	Both
Alternative Reference	0.5
Number of Stages	3
Alpha	0.05
Beta (Lower)	0.1
Beta (Upper)	0.07871
Power (Lower)	0.9
Power (Upper)	0.92129
Max Information (Percent of Fixed Sample)	103.7223
Max Information	47.22445
Null Ref ASN (Percent of Fixed Sample)	79.47628
Lower Alt Ref ASN (Sample Size)	234.1646
Upper Alt Ref ASN (Sample Size)	270.4058

The "Boundary Information" table in Output 78.8.2 displays the information level, alternative reference, and boundary values at each stage. With the default BOUNDARYSCALE=STDZ option, the boundary values are displayed with the standardized Z statistic scale.

Output 78.8.2 Boundary Information

Boundary Information (Standardized Z Scale)
Null Reference = 0

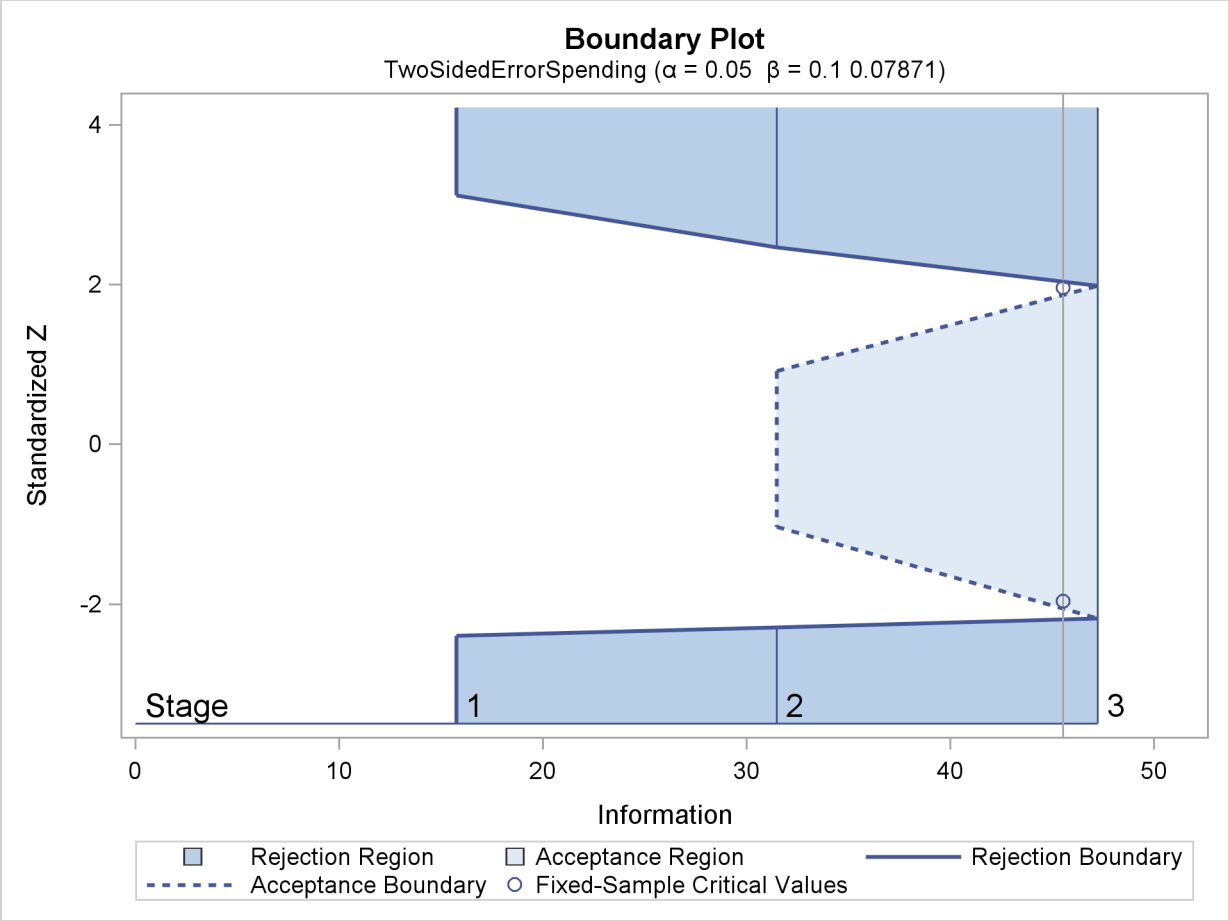
	Information Level			Alternative Reference	
Stage	Proportion	Actual	N	Lower	Upper
1	0.3333	15.74148	122.7119	-1.98378	1.98378
2	0.6667	31.48297	245.4238	-2.80548	2.80548
3	1.0000	47.22445	368.1357	-3.43600	3.43600

Boundary Information (Standardized Z Scale)
Null Reference = 0

	Boundary Values			
	Lower		Upper	
Stage	Alpha	Beta	Beta	Alpha
1	-2.39398	.	.	3.11302
2	-2.29380	-1.02812	0.91855	2.46193
3	-2.17479	-2.17479	1.98311	1.98311

With the specified ODS GRAPHICS ON statement, a detailed boundary plot with the rejection and acceptance regions is displayed by default, as shown in Output 78.8.3.

Output 78.8.3 Boundary Plot



With the SAMPLESIZE statement, the "Sample Size Summary" table in Output 78.8.4 displays the parameters for the sample size computation.

Output 78.8.4 Required Sample Size Summary

Sample Size Summary

Test	Logistic Reg Parameter
Parameter	0.5
Proportion	0.4
X Variance	0.5345
R Square (X)	0
Max Sample Size	368.1357
Expected Sample Size (Null Ref)	282.0807
Expected Sample Size (Alt Ref)	270.4058

The "Sample Sizes" table in Output 78.8.5 displays the required sample sizes for the group sequential clinical trial.

Output 78.8.5 Required Sample Sizes

Sample Sizes (N)
Z Test for Logistic Regression Parameter

	Fractional N		Ceiling N	
Stage	N	Information	N	Information
1	122.71	15.7415	123	15.7784
2	245.42	31.4830	246	31.5569
3	368.14	47.2245	369	47.3353

That is, 123 new patients are needed in each stage and the number is rounded up to 124 for each stage to have a multiple of four for the four dose levels in the trial. Note that since the sample sizes are derived from an estimated response probability and are rounded up, the actual information levels might not match the corresponding target information levels.

Output 78.8.6 lists the first 10 observations of the trial data.

Output 78.8.6 Clinical Trial Data

First 10 Obs in the Trial Data

Obs	Resp	Dose	LDose
1	0	0	0.00000
2	0	1	0.69315
3	1	3	1.38629
4	1	6	1.94591
5	1	0	0.00000
6	1	1	0.69315
7	1	3	1.38629
8	1	6	1.94591
9	0	0	0.00000
10	0	1	0.69315

The following statements use the LOGISTIC procedure to estimate the slope β_1 and its associated standard error at stage 1:

```
proc logistic data=Dose_1;
   model Resp(event='1')= LDose;
   ods output ParameterEstimates=Parms_Dose1;
run;
```

The following statements create and display (in Output 78.8.7) the input data set that contains slope β_1 and its associated standard error for the SEQTEST procedure:

```
data Parms_Dose1;
   set Parms_Dose1;
   if Variable='LDose';
   _Scale_='MLE';
   _Stage_= 1;
   keep _Scale_ _Stage_ Variable Estimate StdErr;
run;

proc print data=Parms_Dose1;
   title 'Statistics Computed at Stage 1';
run;
```

Output 78.8.7 Statistics Computed at Stage 1

Statistics Computed at Stage 1

Obs	Variable	Estimate	StdErr	_Scale_	_Stage_
1	LDose	0.5741	0.2544	MLE	1

The following statements invoke the SEQTEST procedure to test for early stopping at stage 1:

```
ods graphics on;
proc seqtest Boundary=Bnd_Dose
             Parms(testvar=LDose)=Parms_Dose1
             order=mle
             boundaryscale=mle
             ;
ods output Test=Test_Dose1;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 1, which was generated in the SEQDESIGN procedure. The PARMS=PARMS_DOSE1 option specifies the input data set PARMS_DOSE1 that contains the test statistic and its associated standard error at stage 1, and the TESTVAR=LDOSE option identifies the test variable LDOSE in the data set.

The ORDER=MLE option uses the MLE ordering to derive the p-value, the unbiased median estimate, and the confidence limits for the regression slope estimate.

The ODS OUTPUT statement with the TEST=TEST_DOSE1 option creates an output data set named TEST_DOSE1 which contains the updated boundary information for the test at stage 1. The data set also provides the boundary information that is needed for the group sequential test at the next stage.

The "Design Information" table in Output 78.8.8 displays design specifications. By default, the boundary values are modified for the new information levels to maintain the Type I α level. The maximum information remains the same as the design stored in the BOUNDARY= data set, but the derived Type II error probability β and power $1 - \beta$ are different because of the new information levels.

Output 78.8.8 Design Information

The SEQTEST Procedure

Design Information

BOUNDARY Data Set	WORK.BND_DOSE
Data Set	WORK.PARMS_DOSE1
Statistic Distribution	Normal
Boundary Scale	MLE
Alternative Hypothesis	Two-Sided
Early Stop	Accept/Reject Null
Number of Stages	3
Alpha	0.05
Beta (Lower)	0.09992
Beta (Upper)	0.07871
Power (Lower)	0.90008
Power (Upper)	0.92129
Max Information (Percent of Fixed Sample)	103.7231
Max Information	47.2244524
Null Ref ASN (Percent of Fixed Sample)	79.45049
Lower Alt Ref ASN (Percent of Fixed Sample)	66.05269
Upper Alt Ref ASN (Percent of Fixed Sample)	76.24189

The "Test Information" table in Output 78.8.9 displays the boundary values for the test statistic with the specified MLE scale.

Output 78.8.9 Sequential Tests

Test Information (MLE Scale)
Null Reference = 0

	Information Level		Alternative / Reference	
Stage	Proportion	Actual	Lower	Upper
1	0.3272	15.45062	-0.50000	0.50000
2	0.6636	31.33753	-0.50000	0.50000
3	1.0000	47.22445	-0.50000	0.50000

Test Information (MLE Scale)
Null Reference = 0

	Boundary Values				Test	
	Lower		Upper		LDose	
Stage	Alpha	Beta	Beta	Alpha	Estimate	Action
1	-0.61078	.	.	0.79337	0.57409	Continue
2	-0.40974	-0.18169	0.16222	0.44033	.	
3	-0.31633	-0.31633	0.28860	0.28860	.	

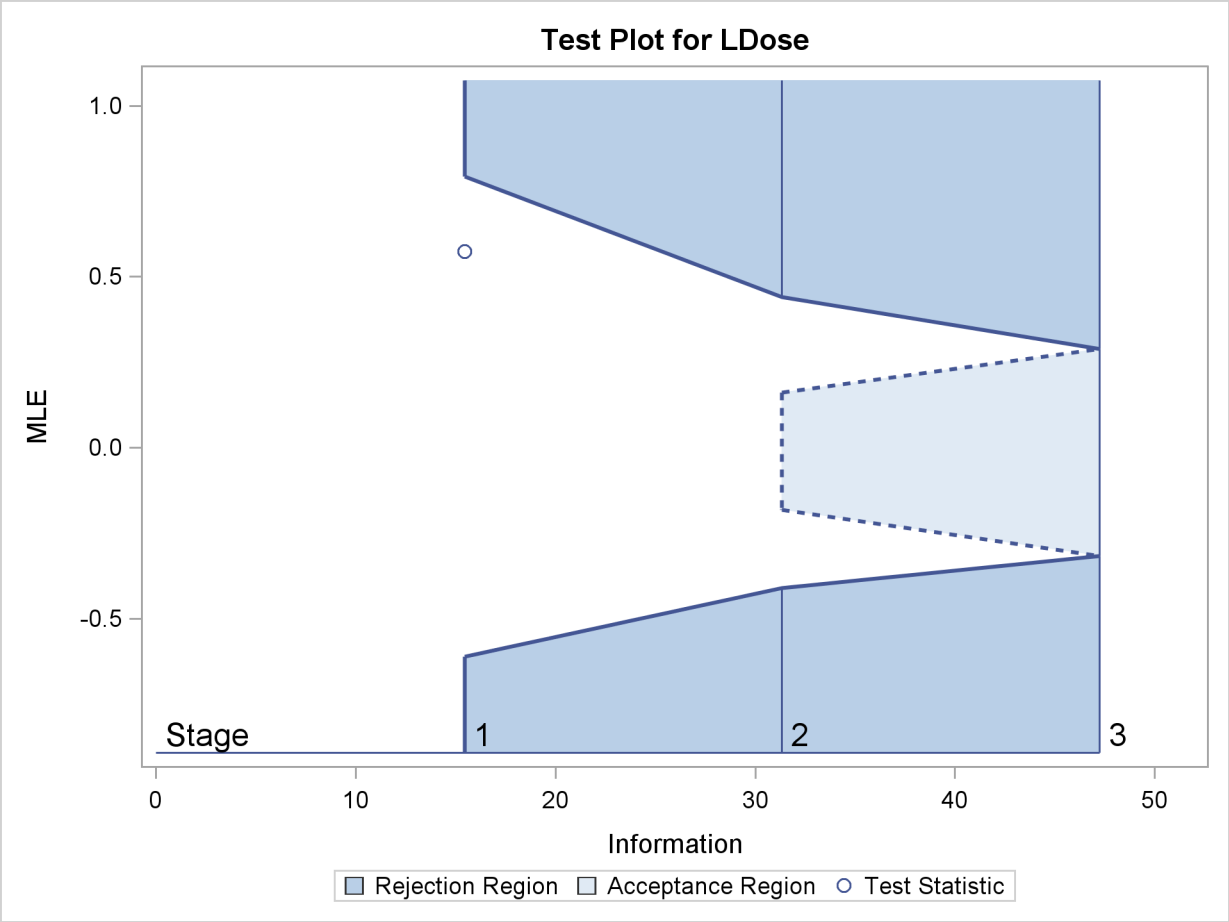
The information level at stage 1 is derived from the standard error s_1 in the PARMS= data set,

$$I_1 = \frac{1}{s_1^2} = \frac{1}{0.2544^2} = 15.45$$

The remaining interim information levels are derived proportionally from the corresponding information levels in the original design. At stage 1, the β boundary values are missing and there is no early stopping to accept H_0. The MLE statistic 0.57409 is between the lower and upper α boundaries, so the trial continues to the next stage.

With the specified ODS GRAPHICS ON statement, a boundary plot with the boundary values and test statistics is displayed, as shown in Output 78.8.10. As expected, the test statistic is in the continuation region below the upper alpha boundary.

Output 78.8.10 Sequential Test Plot



The following statements use the LOGISTIC procedure to estimate the slope β_1 and its associated standard error at stage 2:

```
proc logistic data=dose_2;
   model Resp(event='1')=LDose;
   ods output ParameterEstimates=Parms_Dose2;
run;
```

The following statements create and display (in Output 78.8.11) the input data set that contains slope β_1 and its associated standard error at stage 2 for the SEQTEST procedure:

```
data Parms_Dose2;
   set Parms_Dose2;
   if Variable='LDose';
   _Scale_='MLE';
   _Stage_= 2;
   keep _Scale_ _Stage_ Variable Estimate StdErr;
run;

proc print data=Parms_Dose2;
   title 'Statistics Computed at Stage 2';
run;
```

Output 78.8.11 Statistics Computed at Stage 2

Statistics Computed at Stage 2

Obs	Variable	Estimate	StdErr	_Scale_	_Stage_
1	LDose	0.5213	0.1788	MLE	2

The following statements invoke the SEQTEST procedure to test for early stopping at stage 2:

```
ods graphics on;
proc seqtest Boundary=Test_Dose1
             Parms(Testvar=LDose)=Parms_Dose2
             order=mle
             boundaryscale=mle
             rci
             plots=rci
             ;
ods output Test=Test_Dose2;
run;
ods graphics off;
```

The BOUNDARY= option specifies the input data set that provides the boundary information for the trial at stage 2, which was generated by the SEQTEST procedure at the previous stage. The PARMS= option specifies the input data set that contains the test statistic and its associated standard error at stage 2, and the TESTVAR= option identifies the test variable in the data set.

The ORDER=MLE option uses the MLE ordering to derive the p-value, unbiased median estimate, and confidence limits for the regression slope estimate.

The ODS OUTPUT statement with the TEST=TEST_DOSE2 option creates an output data set named TEST_DOSE2 which contains the updated boundary information for the test at stage 2. The data set also provides the boundary information that is needed for the group sequential test at the next stage.

The "Design Information" table in Output 78.8.12 displays design specifications. By default, the boundary values are modified for the new information levels to maintain the Type I α level.

Output 78.8.12 Design Information

The SEQTEST Procedure

Design Information

BOUNDARY Data Set	WORK.TEST_DOSE1
Data Set	WORK.PARMS_DOSE2
Statistic Distribution	Normal
Boundary Scale	MLE
Alternative Hypothesis	Two-Sided
Early Stop	Accept/Reject Null
Number of Stages	3
Alpha	0.05
Beta (Lower)	0.0999
Beta (Upper)	0.07871
Power (Lower)	0.9001
Power (Upper)	0.92129
Max Information (Percent of Fixed Sample)	103.7227
Max Information	47.2244524
Null Ref ASN (Percent of Fixed Sample)	79.44086
Lower Alt Ref ASN (Percent of Fixed Sample)	66.04641
Upper Alt Ref ASN (Percent of Fixed Sample)	76.22739

The information is derived from the standard error associated with the slope estimate at the final stage and is larger than the target level. The derived Type II error probability β and power $1-\beta$ are different because of the new information levels.

The "Test Information" table in Output 78.8.13 displays the boundary values for the test statistic with the specified MLE scale. The information levels are derived from the standard errors in the PARMS= data set. At stage 2, the slope estimate 0.52128 is larger than 0.44091, the upper α boundary value, the trial stops to reject the null hypothesis of no treatment effect.

Output 78.8.13 Sequential Tests

Test Information (MLE Scale)
Null Reference = 0

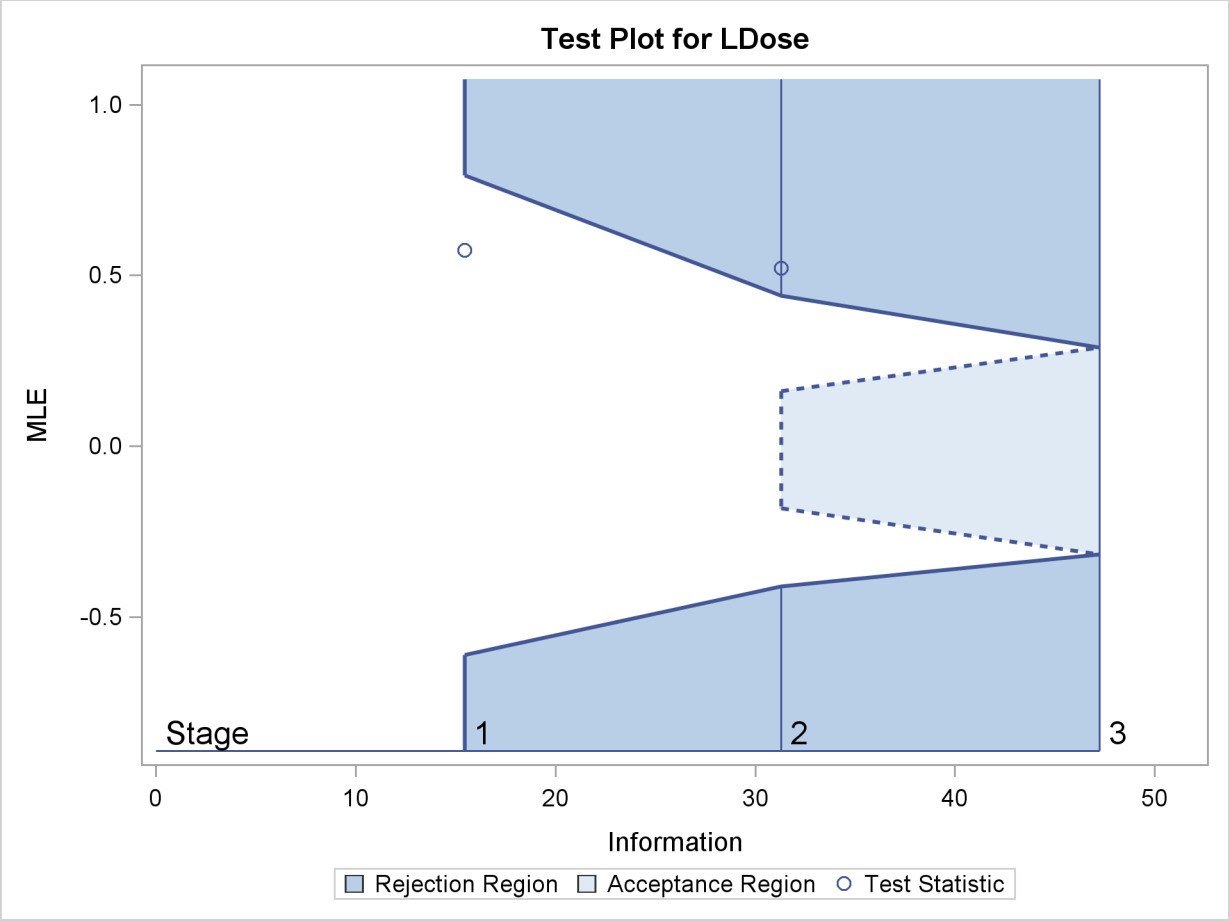
Stage	Information Level-- Proportion	Actual	Alternative------ Reference------- Lower	Upper
1	0.3272	15.45062	-0.50000	0.50000
2	0.6624	31.28346	-0.50000	0.50000
3	1.0000	47.22445	-0.50000	0.50000

Test Information (MLE Scale)
Null Reference = 0

	Boundary Values				Test	
	Lower		Upper		LDose	
Stage	Alpha	Beta	Beta	Alpha	Estimate	Action
1	-0.61078	.	.	0.79337	0.57409	Continue
2	-0.41028	-0.18112	0.16166	0.44091	0.52128	Reject Null
3	-0.31628	-0.31628	0.28861	0.28861	.	

With the specified ODS GRAPHICS ON statement, a boundary plot with the boundary values and test statistics is displayed, as shown in Output 78.8.14. As expected, the test statistic is above the upper α boundary in the upper rejection region at stage 2.

Output 78.8.14 Sequential Test Plot



After a trial is stopped, the "Parameter Estimates" table in Output 78.8.15 displays the stopping stage, parameter estimate, unbiased median estimate, confidence limits, and the p-value under the null hypothesis $H_0 : \beta_1 = 0$.

Output 78.8.15 Parameter Estimates

Parameter Estimates
MLE Ordering

Parameter	Stopping Stage	MLE	p-Value for H0:Parm=0	Median Estimate
LDose	2	0.521275	0.0050	0.502647

Parameter Estimates
MLE Ordering

Parameter	95% Confidence Limits	
LDose	0.15745	0.85154

With the ORDER=MLE option, the MLE ordering is used to compute the p-value, unbiased median estimate, and confidence limits. As expected, the p-value 0.005 is significant at the $\alpha = 0.05$ level and the confidence interval does not contain the null reference zero.

With the RCI option, the "Repeated Confidence Intervals" table in Output 78.8.16 displays repeated confidence intervals for the parameter. For a two-sided test, since the rejection lower repeated confidence limit 0.0804 is greater than the null reference zero, the trial is stopped to reject the hypothesis.

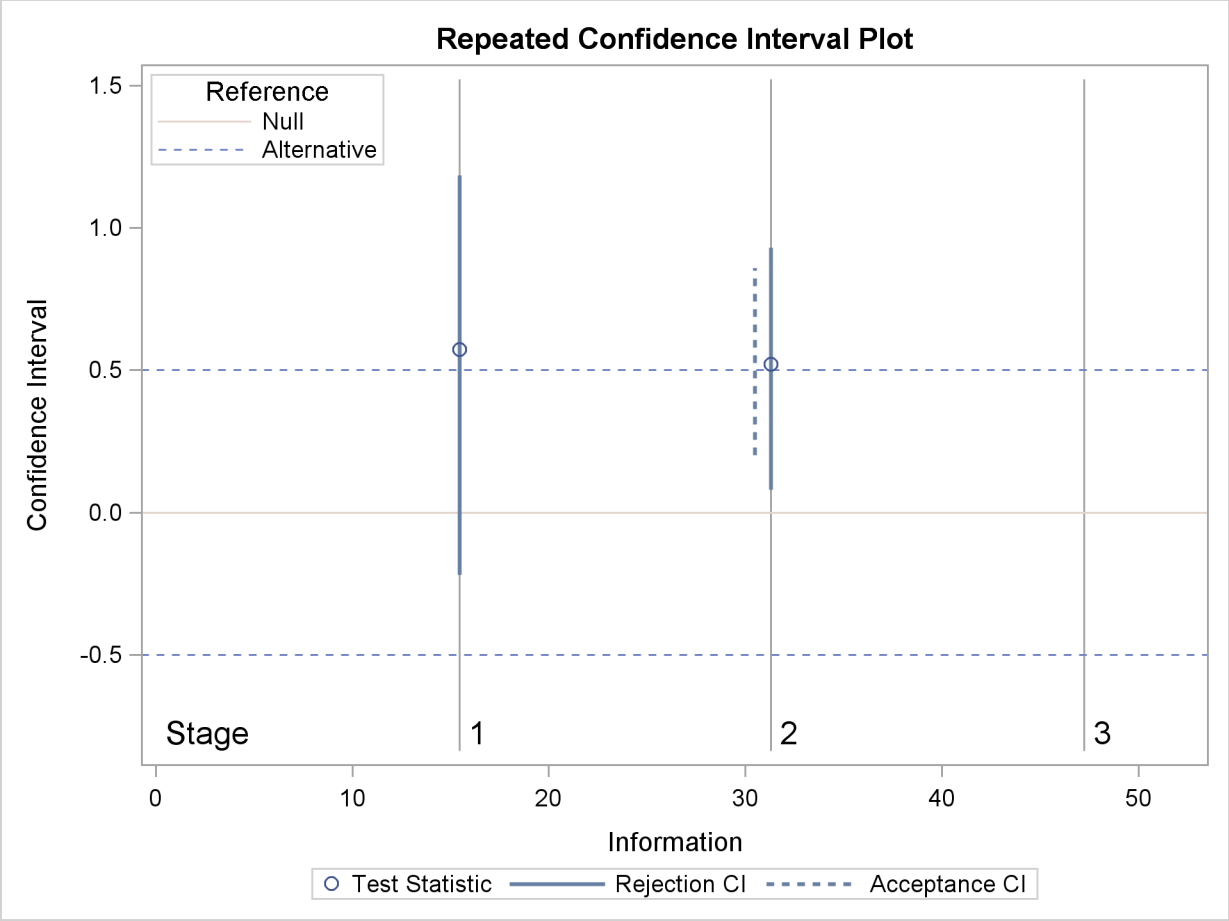
Output 78.8.16 Repeated Confidence Intervals

Repeated Confidence Intervals

Stage	Information Level	Parameter Estimate	-Rejection Boundary- Lower 97.5% Repeated CL	Upper 97.5% Repeated CL	-Acceptance Boundary- Lower 90.01% Repeated CL	Upper 92.13% Repeated CL
1	15.4506	0.57409	-0.2193	1.1849	.	.
2	31.2835	0.52128	0.0804	0.9316	0.2024	0.8596

With the PLOTS=RCI option, the "Repeated Confidence Intervals Plot" displays repeated confidence intervals for the parameter, as shown in Output 78.8.17. It shows that the null reference zero is inside the rejection repeated confidence interval at stage 1 but outside the rejection repeated confidence interval at stage 2. This implies that the study stops at stage 2 to reject the hypothesis.

Output 78.8.17 Repeated Confidence Intervals Plot



Note that the hypothesis is accepted if at any stage, the acceptance repeated confidence interval falls within the interval $(-0.5, 0.5)$ of the alternative references.

Acknowledgments

In addition to being shaped by the research literature listed in the section "References" on page 6151, the development of the SEQDESIGN and SEQTEST procedures has benefited significantly from the advice and expertise of the following researchers:

- Lu Cui, Eisai Medical Research
- Alex Dmitrienko, Eli Lilly
- Scott Emerson, University of Washington
- Gordon Lan, Johnson & Johnson
- Steve Snapinn, Amgen
- John Whitehead, University of Reading

We gratefully acknowledge the time and effort that these researchers have contributed.

References

Chang, M. N. (1989), "Confidence Intervals for a Normal Mean Following a Group Sequential Test," *Biometrics*, 45, 247–254.

Chang, M. N., Gould, A. L., and Snapinn, S. M. (1995), "P-values for Group Sequential Testing," *Biometrika*, 82, 650–654.

Chow, S. C. and Liu, J. P. (1998), *Design and Analysis of Clinical Trials, Concept and Methodologies,* New York: John Wiley & Sons.

Chow, S. C., Shao, J., and Wang, H. (2003), *Sample Size Calculations in Clinical Research,* Boca Raton, FL: CRC Press.

Cox, E. R. (1972), "Regression Models and Life-Tables (with Discussion)," *Journal of the Royal Statistical Society, B*, 34, 187–220.

Cui, L., Hung, H. M. J., and Wang, S. (1999), "Modification of Sample Size in Group Sequential Clinical Trials," *Biometrics*, 55, 853–857.

DeMets, D. L., Furberg, C. D., and Friedman, L. M. (2006), *Data Monitoring in Clinical Trials,* New York: Springer.

Dmitrienko, A., Molenberghs, G., Chuang-Stein, C., and Offen, W. (2005), *Analysis of Clinical Trials Using SAS: A Practical Guide,* Cary, NC: SAS Institute Inc.

Ellenberg, S. S., Fleming, T. R., and DeMets, D. L. (2003), *Data Monitoring Committees in Clinical Trials,* New York: John Wiley & Sons.

Emerson, S. S. (1996), "Statistical Packages for Group Sequential Methods," *The American Statistician*, 50, 183–192.

Emerson, S. S. and Fleming, T. R. (1989), "Symmetric Group Sequential Designs," *Biometrics*, 45, 905–923.

Emerson, S. S. and Fleming, T. R. (1990), "Parameter Estimation Following Group Sequential Hypothesis Testing," *Biometrika*, 77, 875–892.

Emerson, S. S., Kittelson, J. M., and Gillen, D. L. (2005), "On the Use of Stochastic Curtailment in Group Sequential Clinical Trials," *UW Biostatistics Working Paper Series*, *http://www.bepress.com/uwbiostat/paper243*.

Fairbanks, K. and Madsen, R. (1982), "P Values for Tests Using a Repeated Significance Test Design," *Biometrika*, 69, 69–74.

Food and Drug Administration (1998), "E9: Statistical Principles for Clinical Trials," *Federal Register*, 63 (179), 49583–49598.

Herson, J. (1979), "Predictive Probability Early Termination for Phase II Clinical Trials," *Biometrics*, 35, 775–783.

Hwang, I. K., Shih, W. J., and DeCani, J. S. (1990), "Group Sequential Designs Using a Family of Type I Error Probability Spending Functions," *Statistics in Medicine*, 9, 1439–1445.

Jennison, C. and Turnbull, B. W. (2000), *Group Sequential Methods with Applications to Clinical Trials,* New York: Chapman & Hall.

Kim, K. and Tsiatis, A. A. (1990), "Study Duration for Clinical Trials with Survival Response and Early Stopping Rule," *Biometrics*, 46, 81–92.

Kittelson, J. M. and Emerson, S. S. (1999), "A Unifying Family of Group Sequential Test Designs," *Biometrics*, 55, 874–882.

Lan, K. K. G. and DeMets, D. L. (1983), "Discrete Sequential Boundaries for Clinical Trials," *Biometrika*, 70, 659–663.

Lan, K. K. G., Lachin, J. M., and Bautista, O. (2003), "Over-ruling a Group Sequential Boundary: A Stopping Rule versus a Guideline," *Statistics in Medicine*, 22, 3347–3355.

Lan, K. K. G., Simon, R., and Halperin, M. (1982), "Stochastically Curtailed Tests in Long-Term Clinical Trials," *Sequential Analysis*, 1, 207–219.

McCullagh, P. and Nelder, J. A. (1989), *Generalized Linear Models,* Second Edition, New York: Chapman & Hall/CRC.

Mehta, C. R. and Tsiatis, A. A. (2001), "Flexible Sample Size Considerations under Information Based Interim Monitoring," *Drug Information Journal*, 35, 1095–1112.

O'Brien, P. C. and Fleming, T. R. (1979), "A Multiple Testing Procedure for Clinical Trials," *Biometrics*, 35, 549–556.

Proschan, M. A., Lan, K. K. G., and Wittes, J. T. (2006), *Statistical Monitoring of Clinical Trials,* New York: Springer.

Rosner, G. L. and Tsiatis, A. A. (1988), "Exact Confidence Intervals Following a Group Sequential Trial: A Comparison of Methods," *Biometrika*, 75, 723–729.

Rudser, K.D. and Emerson, S.S. (2007), "Implementing Type I & Type II Error Spending for Two-Sided Group Sequential Designs," *Contemporary Clinical Trials*, doi:10.1016/j.cct.2007.09.002.

Scharfstein, D. O. and Tsiatis, A. A. (1998), "The Use of Simulation and Bootstrap in Information-Based Group Sequential Studies," *Statistics in Medicine*, 17, 75–87.

Tsiatis, A. A. and Mehta C. R. (2003), "On the Inefficiency of the Adaptive Design for Monitoring Clinical Trials," *Biometrika*, 90, 367–378.

Tsiatis, A. A., Rosner G. L., and Mehta C. R. (1984), "Exact Confidence Intervals Following a Group Sequential Test," *Biometrics*, 40, 797–803.

Ware, J. H., Muller, J. E., and Braunwald, E. (1985), "The Futility Index: An Approach to the Cost-Effective Termination of Randomized Clinical Trials" *American Journal of Medicine*, 78, 635–643.

Whitehead, J. (1997), *The Design and Analysis of Sequential Clinical Trials,* Revised Second Edition, Chichester: John Wiley & Sons.

Chapter 79

The SIM2D Procedure

Contents

Overview: SIM2D Procedure **6155**
Introduction to Spatial Simulation 6156
Getting Started: SIM2D Procedure **6157**
Preliminary Spatial Data Analysis 6157
Investigating Variability by Simulation 6158
Syntax: SIM2D Procedure **6164**
PROC SIM2D Statement 6165
BY Statement 6169
COORDINATES Statement 6169
GRID Statement 6170
SIMULATE Statement 6171
MEAN Statement 6175
Details: SIM2D Procedure **6176**
Computational and Theoretical Details of Spatial Simulation 6176
Introduction 6176
Theoretical Development 6176
Computational Details 6179
Output Data Set 6180
Displayed Output 6181
ODS Table Names 6181
ODS Graphics 6181
Examples: SIM2D Procedure **6182**
Example 79.1: Simulation 6182
Simulating a Subregion for Economic Feasibility 6182
Implementation Using PROC SIM2D 6184
Example 79.2: Variability at Selected Locations 6187
References **6190**

Overview: SIM2D Procedure

The SIM2D procedure produces a spatial simulation for a Gaussian random field with a specified mean and covariance structure in two dimensions by using an LU decomposition technique.

The simulation can be conditional or unconditional. If it is conditional, a set of coordinates and associated field values are read from a SAS data set. The resulting simulation honors these data values.

You can specify the mean structure as a quadratic function in the coordinates. You can specify the covariance by naming the form and supplying the associated parameters.

PROC SIM2D can handle anisotropic and nested semivariogram models. Three covariance models are supported: Gaussian, exponential, and spherical. A single nugget effect is also supported.

You can specify the locations of simulation points in a GRID statement, or they can be read from a SAS data set. The grid specification is most suitable for a regular grid; the data set specification can handle any irregular pattern of points.

The SIM2D procedure writes the simulated values for each grid point to an output data set. The SIM2D procedure now uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For more information about the graphics available in PROC SIM2D, see the section "ODS Graphics" on page 6181.

Introduction to Spatial Simulation

The purpose of spatial simulation is to produce a set of partial realizations of a spatial random field (SRF) $Z(\boldsymbol{s}), \boldsymbol{s} \in D \subset \mathcal{R}^2$ in a way that preserves a specified mean $\mu(\boldsymbol{s}) = E[Z(\boldsymbol{s})]$ and covariance structure $C_z(\boldsymbol{s}_1 - \boldsymbol{s}_2) = \text{Cov}(Z(\boldsymbol{s}_1), Z(\boldsymbol{s}_2))$. The realizations are partial in the sense that they occur only at a finite set of locations $(\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_n)$. These locations are typically on a regular grid, but they can be arbitrary locations in the plane.

PROC SIM2D produces simulations for continuous processes in two dimensions by using the lower-upper (LU) decomposition method. In these simulations the possible values of the measured quantity $Z(\boldsymbol{s}_0)$ at location $\boldsymbol{s}_0 = (x_0, y_0)$ can vary continuously over a certain range. An additional assumption, needed for computational purposes, is that the spatial random field $Z(\boldsymbol{s})$ is Gaussian. The section "Details: SIM2D Procedure" on page 6176 provides more information about different types of spatial simulation and associated computational methods.

Spatial simulation is different from spatial prediction, where the emphasis is on predicting a point value at a given grid location. In this sense, spatial prediction is local. In contrast, spatial simulation is global; the emphasis is on the entire realization $(Z(\boldsymbol{s}_1), Z(\boldsymbol{s}_2), \cdots, Z(\boldsymbol{s}_n))$.

Given the correct mean $\mu(\boldsymbol{s})$ and covariance structure $C_z(\boldsymbol{s}_1 - \boldsymbol{s}_2)$, SRF quantities that are difficult or impossible to calculate in a spatial prediction context can easily be approximated by functions of multiple simulations.

Getting Started: SIM2D Procedure

Spatial simulation, just like spatial prediction, requires a model of spatial dependence, usually in terms of the covariance $C_z(\boldsymbol{h})$. For a given set of spatial data $Z(\boldsymbol{s}_i), i = 1, \cdots, n$, the covariance structure (both the form and parameter values) can be found by the VARIOGRAM procedure. This example uses the coal seam thickness data that are also used in the section "Getting Started: VARIOGRAM Procedure" on page 7519.

In this example, the data consist of coal seam thickness measurements (in feet) taken over an area of 100×100 (10^6 ft^2). The coordinates are offsets from a point in the southwest corner of the measurement area, with the north and east distances in units of thousands of feet.

Preliminary Spatial Data Analysis

The semivariogram analysis of the thick data set in "Example 95.1: Theoretical Semivariogram Model Fitting" on page 7570 of the VARIOGRAM procedure considered the spatial random field (SRF) $Z(\boldsymbol{s})$ of the thickness values to be free of surface trends. The expected value $E[Z(\boldsymbol{s})]$ is then a constant $\mu(\boldsymbol{s}) = \mu$, which suggests that you can work with the original thickness data rather than residuals from a trend surface fit. In fact, a reasonable approximation of the spatial process generating the coal seam data is given by

$$Z(\boldsymbol{s}) = \mu + \varepsilon(\boldsymbol{s})$$

where $\varepsilon(\boldsymbol{s})$ is a Gaussian SRF with Gaussian covariance structure

$$C_z(\boldsymbol{h}) = c_0 \exp\left(-\frac{h^2}{a_0^2}\right)$$

Note that the term "Gaussian" is used in two ways in this description. For a set of locations $\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_n$, the random vector

$$\boldsymbol{Z}(\boldsymbol{s}) = \begin{bmatrix} Z(\boldsymbol{s}_1) \\ Z(\boldsymbol{s}_2) \\ \vdots \\ Z(\boldsymbol{s}_n) \end{bmatrix}$$

has a multivariate Gaussian or normal distribution $N_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. The (i, j)th element of $\boldsymbol{\Sigma}$ is computed by $C_z(\boldsymbol{s}_i - \boldsymbol{s}_j)$, which happens to be a Gaussian functional form.

Any functional form for $C_z(\boldsymbol{h})$ that yields a valid covariance matrix $\boldsymbol{\Sigma}$ can be used. Both the functional form of $C_z(\boldsymbol{h})$ and the parameter values

$$\mu = 40.1173$$

$$c_0 = 7.2881$$

$$a_0 = 30.6239$$

are estimated by using PROC VARIOGRAM and PROC NLIN in "Example 95.1: Theoretical Semivariogram Model Fitting" on page 7570 in the VARIOGRAM procedure. Specifically, the expected value μ is reported in the VARIOGRAM procedure OUTV output data set, and the parameters c_0 and a_0 are estimates derived from a weighted least squares fit.

The choice of a Gaussian functional form for $C_z(\boldsymbol{h})$ is simply based on the data, and it is not at all crucial to the simulation. However, it *is* crucial to the simulation method used in PROC SIM2D that $Z(\boldsymbol{s})$ be a Gaussian SRF. For details, see the section "Computational and Theoretical Details of Spatial Simulation" on page 6176.

Investigating Variability by Simulation

The variability of $Z(\boldsymbol{s})$, as modeled by

$$Z(\boldsymbol{s}) = \mu + \varepsilon(\boldsymbol{s})$$

with the Gaussian covariance structure $C_z(\boldsymbol{h})$ found previously, is not obvious from the covariance model form and parameters. The variation around the mean of the surface is relatively small, making it difficult visually to pick up differences in surface plots of simulated realizations.

Instead, you can compute the mean for each location on a grid from a series of realizations in a simulation. Then, the standard deviation of all the simulated values at each grid location will provide you with a measure of the variability of $Z(\boldsymbol{s})$ for the given covariance structure. You can also investigate variations at selected grid points in more detail, as shown in the "Example 79.2: Variability at Selected Locations" on page 6187.

In the present example you will see how to use ODS Graphics with PROC SIM2D to investigate the mean and standard deviation of simulated values. In particular, you produce 5,000 realizations of a simulation with PROC SIM2D, where you specify the Gaussian model with the parameters found previously. You want the simulated data to pass through the simulated values, so first you define the data with the following DATA step:

```
data thick;
   input East North Thick @@;
   label Thick='Coal Seam Thickness';
   datalines;
    0.7  59.6  34.1   2.1  82.7  42.2   4.7  75.1  39.5
    4.8  52.8  34.3   5.9  67.1  37.0   6.0  35.7  35.9
    6.4  33.7  36.4   7.0  46.7  34.6   8.2  40.1  35.4
   13.3   0.6  44.7  13.3  68.2  37.8  13.4  31.3  37.8
   17.8   6.9  43.9  20.1  66.3  37.7  22.7  87.6  42.8
   23.0  93.9  43.6  24.3  73.0  39.3  24.8  15.1  42.3
   24.8  26.3  39.7  26.4  58.0  36.9  26.9  65.0  37.8
   27.7  83.3  41.8  27.9  90.8  43.3  29.1  47.9  36.7
   29.5  89.4  43.0  30.1   6.1  43.6  30.8  12.1  42.8
   32.7  40.2  37.5  34.8   8.1  43.3  35.3  32.0  38.8
   37.0  70.3  39.2  38.2  77.9  40.7  38.9  23.3  40.5
   39.4  82.5  41.4  43.0   4.7  43.3  43.7   7.6  43.1
   46.4  84.1  41.5  46.7  10.6  42.6  49.9  22.1  40.7
   51.0  88.8  42.0  52.8  68.9  39.3  52.9  32.7  39.2
   55.5  92.9  42.2  56.0   1.6  42.7  60.6  75.2  40.1
   62.1  26.6  40.1  63.0  12.7  41.8  69.0  75.6  40.1
   70.5  83.7  40.9  70.9  11.0  41.7  71.5  29.5  39.8
   78.1  45.5  38.7  78.2   9.1  41.7  78.4  20.0  40.8
   80.5  55.9  38.7  81.1  51.0  38.6  83.8   7.9  41.6
   84.5  11.0  41.5  85.2  67.3  39.4  85.5  73.0  39.8
   86.7  70.4  39.6  87.2  55.7  38.8  88.1   0.0  41.6
   88.4  12.1  41.3  88.4  99.6  41.2  88.8  82.9  40.5
   88.9   6.2  41.5  90.6   7.0  41.5  90.7  49.6  38.9
   91.5  55.4  39.0  92.9  46.8  39.1  93.4  70.9  39.7
   55.8  50.5  38.1  96.2  84.3  40.3  98.2  58.2  39.5
   ;
```

Since this is a conditional simulation, you can specify the OBSERV option in the PLOT option in PROC SIM2D to see the locations and values of the measured points in the area where you want to perform spatial simulations.

Furthermore, the MEANS suboption in the PLOT option specifies that you want to create a plot that shows the means of the simulated values across the region. The MEANS suboption with no other arguments produces a plot that shows the contours of the simulated means in the foreground and the gradient of the simulated standard deviations in the background.

You obtain these PROC SIM2D results at the nodes of an output grid that you specify according to your application needs. In the present analysis, a convenient area that encompasses all the Thick data points is a square with a side length of 100,000 feet. You define a regular grid for your simulation in this area. Assume a distance of 2,500 feet between grid nodes in both directions for a smooth contour plot. Based on this choice, your square grid has 41 nodes on each side. This means that PROC SIM2D computes the simulated values at a total of 1,681 grid points. You use the GRID statement of the PROC SIM2D to specify this grid.

The SIMULATE statement specifies the parameters of your simulation across the output grid. In particular, the VAR= option specifies the conditional simulation variable. The number of realizations in the simulation is specified with the NUMREAL option. The SEED= option specifies the seed for the simulation random number generator.

The spatial correlation model for the simulation is also specified in the SIMULATE statement. You specify the model type by using the FORM= option. The options SCALE= and RANGE= specify the covariance structure sill c_0 and range a_0 parameters, respectively, as discussed in the previous section.

Although it is not included in the original spatial structure, note that a minimal nugget effect is specified with the NUGGET= option to avoid singularity issues. Singularity can appear in the present example as a result of the combined use of the Gaussian covariance model and relatively short distances between nodes, data, or nodes and data in the simulation area.

These steps are implemented using the following DATA step and statements:

```
ods graphics on;

proc sim2d data=thick outsim=sim plot=(observ means);
   coordinates xc=East yc=North;
   simulate var=Thick numreal=5000 seed=79931
      scale=7.2881 range=30.6239 nugget=1e-8 form=gauss;
   mean 40.1173;
   grid x=0 to 100 by 2.5 y=0 to 100 by 2.5;
run;
```

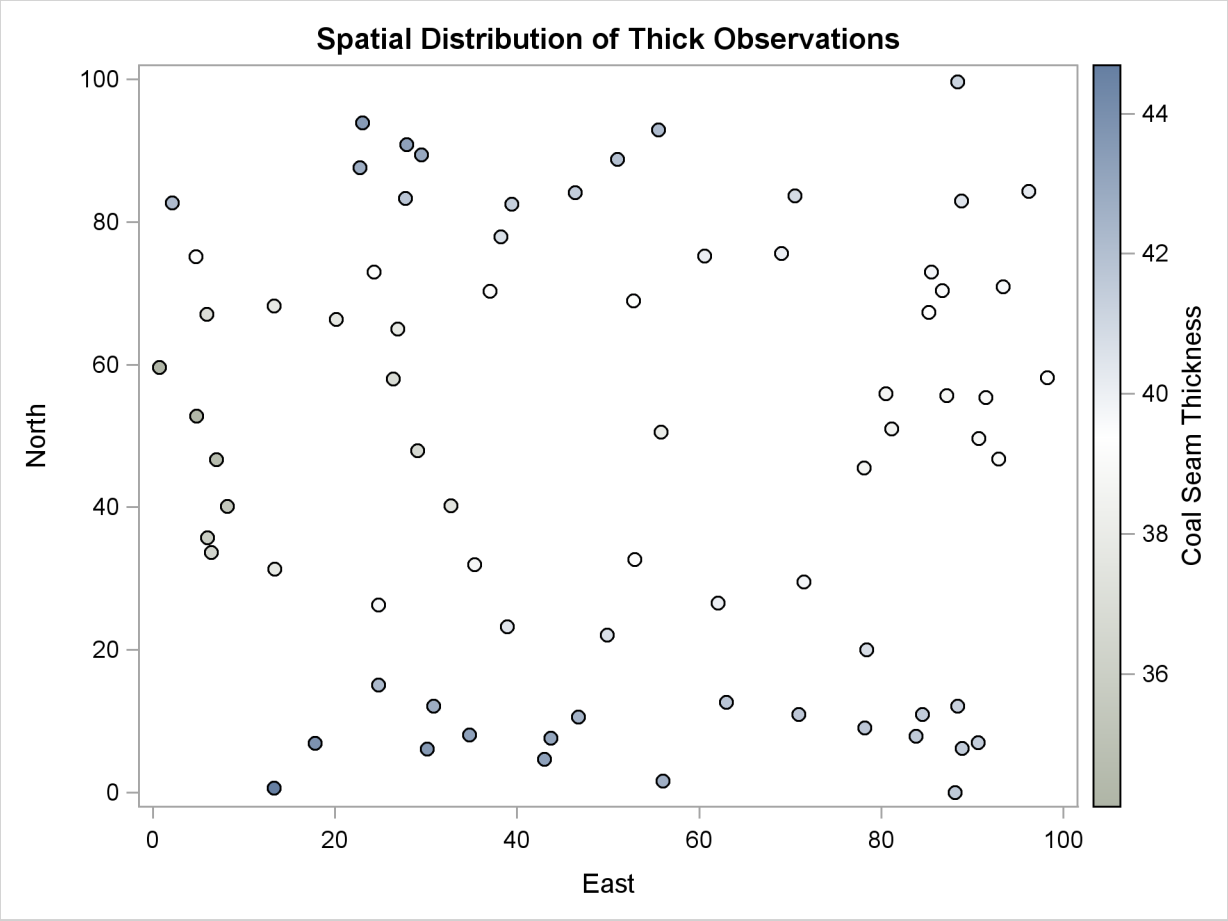
The table in Figure 79.1 shows the number of observations read and used in the conditional simulation. This table can provide you with useful information in case you have missing values in the input data.

Figure 79.1 Number of Observations for the thick Data Set

```
                    The SIM2D Procedure
          Simulation: Sim1, Dependent Variable: Thick

            Number of Observations Read          75
            Number of Observations Used          75
```

The sample locations are then plotted in Figure 79.2. The figure clearly shows some small-scale variation that is typical of spatial data.

Figure 79.2 Scatter Plot of the Observations Spatial Distribution



PROC SIM2D also produces the table shown in Figure 79.3, which contains information about the type of simulation you run and the number of realizations requested.

Figure 79.3 Simulation Analysis Information

Simulation Information

Type	**Conditional**
Number of Realizations	**5000**

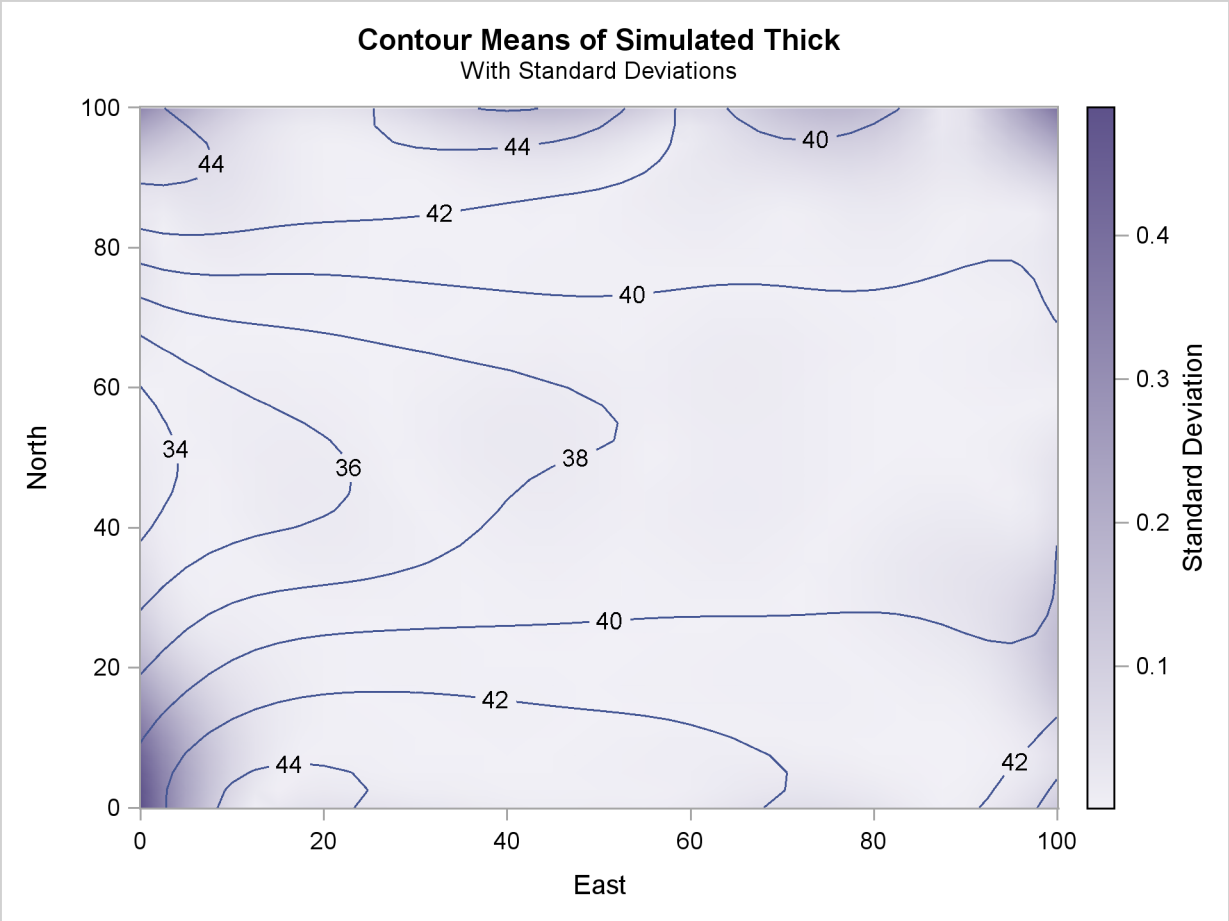
The table in Figure 79.4 displays the spatial correlation model information that will be used by PROC SIM2D for the current simulation. If applicable, the table provides the effective range, as well. This is the distance r_ϵ at which the covariance is 5% of its value at zero. Here you specified the Gaussian model, for which the effective range r_ϵ is $\sqrt{3}a_0$.

Figure 79.4 Simulation Covariance Model Information

Covariance Model Information	
Type	Gaussian
Sill	7.2881
Range	30.6239
Effective Range	53.042151
Nugget Effect	1E-8

Eventually, the SIM2D procedure produces the requested means plot shown in Figure 79.5. The contours of the mean of the simulated values show the average of the simulated realizations at each grid node, which is based on the given spatial structure characteristics. In this case, these means are also conditioned by the Thick observations across the region.

Figure 79.5 Contour Plot of Conditionally Simulated Coal Seam Thickness



Note also that the gradient showing the standard deviation of the simulated values at each grid node is generally small throughout the region. A few exceptions are evident close to the region borders. In these areas the simulated realizations depend on a limited amount of neighboring data. The simulation at these locations relies mainly on the underlying spatial structure.

In addition to the simulation analysis, you can use the PROC SIM2D output to obtain statistical information about the simulated values at selected locations. Let us assume that you would like some basic statistics for the extreme southwest point at (East=0, North=0) and the point (East=75, North=75) toward the northeast corner of the region. You use the following DATA step to select the realizations for these points from the OUTSIM= output data set:

```
data selected;
   set sim(where=((gxc=0 and gyc=0) or (gxc=75 and gyc=75)));
   label gxc = "X-coord";
   label gyc = "Y-coord";
run;
```

Then, you use PROC SORT to sort the selected data set entries and PROC MEANS to produce the simulation statistics for the selected points. The following statements yield the mean, standard deviation, and maximum values of the 5,000 realizations of the Thick values at each one of the selected locations:

```
proc sort data=selected;
   by gxc gyc;
run;

proc means data=selected Mean Std Max;
   class gxc gyc;
   ways 2;
   where (   ((gxc =  0) & (gyc =  0))
           | ((gxc = 75) & (gyc = 75)));
   var SValue;
run;

ods graphics off;
```

The requested statistics for the grid points (East=0, North=0) and (East=75,North=75) are shown in Figure 79.6.

Figure 79.6 Simulation Statistics at Grid Points (East=0, North=0) and (East=75, North=75)

```
                          The MEANS Procedure

          Analysis Variable : SVALUE Simulated Value at Grid Point

     X-coord        Y-coord    N Obs           Mean         Std Dev         Maximum
-------------------------------------------------------------------------------------
           0              0     5000     40.6064093       0.4900041      42.4131782

          75             75     5000     40.1099449       0.0020988      40.1187941
-------------------------------------------------------------------------------------
```

"Example 79.2: Variability at Selected Locations" on page 6187 shows you how to perform a simulation at a set of selected locations rather than a domain-wide grid, and how to obtain more detailed statistics from the simulation.

Syntax: SIM2D Procedure

The following statements are available in PROC SIM2D:

PROC SIM2D *options* ;
  BY *variables* ;
  COORDINATES *coordinate-variables* ;
  GRID *grid-options* ;
  SIMULATE *simulate-options* ;
  MEAN *mean-options* ;

The SIMULATE and MEAN statements are hierarchical; you can specify any number of SIMULATE statements, but you must specify at least one. If you specify a MEAN statement, it refers to the preceding SIMULATE statement. If you do not specify a MEAN statement, a zero-mean model is simulated.

You must specify a single COORDINATES statement to identify the x and y coordinate variables in the input data set when you perform a conditional simulation. You must also specify a single GRID statement to specify the grid information.

The following table outlines the options available in PROC SIM2D classified by function.

Table 79.1 Options Available in the SIM2D Procedure

Task	Statement	Option
Data Set Options		
Specify input data set	PROC SIM2D	DATA=
Specify grid data set	GRID	GDATA=
Write simulated values	PROC SIM2D	OUTSIM=
Specify plot display and options	PROC SIM2D	PLOTS=
Specify quadratic form data set	MEAN	QDATA=
Specify plot display and options	PROC SIM2D	PLOTS=
Declaring the Role of Variables		
Specify variables to define analysis subgroups	BY	
Specify the conditioning variable	SIMULATE	VAR=
Specify the x and y coordinate variables in the DATA= data set	COORDINATES	XC= YC=
Specify the x and y coordinate variables in the GDATA= data set	GRID	XC= YC=
Specify the constant coefficient variable in the QDATA= data set	MEAN	CONST=
Specify the linear x coefficient variable in the QDATA= data set	MEAN	CX=
Specify the linear y coefficient variable in the QDATA= data set	MEAN	CY=

Table 79.1 *continued*

Task	Statement	Option
Specify the quadratic x coefficient variable in the QDATA= data set	MEAN	CXX=
Specify the quadratic y coefficient variable in the QDATA= data set	MEAN	CYY=
Specify the quadratic xy coefficient variable in the QDATA= data set	MEAN	CXY=
Controlling the Simulation		
Specify the number of realizations	SIMULATE	NUMREAL=
Specify the seed value for the random generator	SIMULATE	SEED=
Controlling the Mean Quadratic Surface		
Specify the CONST term	MEAN	CONST=
Specify the linear x term	MEAN	CX=
Specify the linear y term	MEAN	CY=
Specify the quadratic x term	MEAN	CXX=
Specify the quadratic y term	MEAN	CYY=
Specify the quadratic cross term	MEAN	CXY=
Controlling the Semivariogram Model		
Specify a nugget effect	SIMULATE	NUGGET=
Specify a functional form	SIMULATE	FORM=
Specify nested functional forms	SIMULATE	FORM=($f_1, \cdots, f_k$)
Specify a range parameter	SIMULATE	RANGE=
Specify nested range parameters	SIMULATE	RANGE=($r_1, \cdots, r_k$)
Specify a scale parameter	SIMULATE	SCALE=
Specify nested scale parameters	SIMULATE	SCALE=($s_1, \cdots, s_k$)
Specify an angle for an anisotropic model	SIMULATE	ANGLE=
Specify nested angles	SIMULATE	ANGLE=($a_1, \cdots, a_k$)
Specify a minor-major axis ratio for an anisotropic model	SIMULATE	RATIO=
Specify nested minor-major axis ratios	SIMULATE	RATIO=($ra_1, \cdots, ra_k$)

PROC SIM2D Statement

PROC SIM2D *options* **;**

You can specify the following options with the PROC SIM2D statement.

DATA=*SAS-data-set*
specifies a SAS data set that contains the x and y coordinate variables and the VAR= variables that are used in the SIMULATE statements. This data set is required if you specify the BY

statement or if any of the SIMULATE statements are conditional—that is, if you specify the VAR= option in any of those. Otherwise, you do not need the DATA= option and this option is ignored if you specify it.

NARROW

restricts the variables included in the OUTSIM= data set. When you specify the NARROW option, only four variables are included. This option is useful when a large number of simulations are produced. Including only four variables reduces the memory required for the OUTSIM= data set. For details on the variables that are excluded with the NARROW option, see the section "Output Data Set" on page 6180.

NOPRINT

suppresses the normal display of results. The NOPRINT option is useful when you want only to create one or more output data sets with the procedure. Note that this option temporarily disables the Output Delivery System (ODS); see the section "ODS Graphics" on page 6181 for more information.

OUTSIM=*SAS-data-set*

specifies a SAS data set to store the simulation values, iteration number, simulate statement label, variable name, and grid location. For details, see the section "Output Data Set" on page 6180.

PLOTS <(*global-plot-option***)> <= *plot-request* <(***options***)>>**

PLOTS <(*global-plot-option***)> <= (***plot-request* <(***options***)>** **<... *plot-request* <(***options***)>>)>**

controls the plots produced through ODS Graphics. When you specify only one plot request, you can omit the parentheses around the plot request. Here are some examples:

```
plots=none
plots=observ
plots=(observ(outl) means)
plots=(means(fill=mean line=sd obs=grad) means(fill=sd))
```

You must enable ODS Graphics before requesting plots, for example, like this:

```
ods graphics on;

proc sim2d data=thick outsim=sim;
   coordinates xc=East yc=North;
   simulate var=Thick numreal=5000 seed=79931
      scale=7.2881 range=30.6239 form=gauss;
   mean 40.1173;
   grid x=0 to 100 by 2.5 y=0 to 100 by 2.5;
run;

ods graphics off;
```

For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." PROC SIM2D does not have default plots. This means that unless you have enabled ODS Graphics, specified the PLOTS= option, or specified PLOTS=ALL, then PROC SIM2D will produce no plots.

The following *global-plot-option* is available:

ONLY
produces only plots specifically requested.

The following individual *plot-requests* and plot *options* are available:

ALL
produces all appropriate plots. You can specify other options with ALL. For example, to request all appropriate plots and an additional plot of the simulation means, specify PLOTS=(ALL MEANS).

EQUATE
specifies that all appropriate plots be produced in a way that the axes coordinates have equal size units.

NONE
supresses all plots.

OBSERVATIONS < (*observations-plot-options*) >

OBSERV < (*observations-plot-options*) >

OBS < (*observations-plot-options*) >
produces the observed data plot. Only one observations plot will be created if you specify the OBSERVATIONS option more than once within a PLOTS option.

The OBSERVATIONS option has the following suboptions:

GRADIENT
specifies that observations be displayed as circles colored by the observed measurement.

OUTLINE
specifies that observations be displayed as circles with a border but with a completely transparent fill.

OUTLINEGRADIENT
is the same as OBSERVATIONS(GRADIENT) except that a border is shown around each observation.

SHOWMISSING
specifies that observations with missing values be displayed in addition to the observations with nonmissing values. By default, missing values locations are not shown on the plot. If you specify multiple instances of the OBSERVATIONS option, and you specify the SHOWMISSING suboption in any of those, then the resulting observations plot will display the observations with missing values.

For the GRADIENT, OUTLINE, and OUTLINEGRADIENT suboptions: The OUTLINEGRADIENT is the default suboption if you do not specify any of those three. If you specify multiple instances of the OBSERVATIONS option or multiple suboptions for OBSERVATIONS, then the resulting observations plot will honor the last specified GRADIENT, OUTLINE, or OUTLINEGRADIENT suboption.

MEANS < (*means-plot-options*) >

MEA < (*means-plot-options*) >

specifies that a plot of the means and standard deviations of the simulation realizations at each grid point be produced. You can specify the MEANS option multiple times in the same PLOTS option to request instances of plots with the following *means-plot-options*:

FILL=NONE | MEAN | SD

produces a surface plot for either the values of the means or the standard deviations. FILL=SD is the default; however, if you do not specify the FILL suboption the behavior depends on the LINE suboption as follows: If you specify LINE=NONE or do not specify the LINE suboption at all, then the FILL suboption will be set to its default value. If LINE=PRED or LINE=SE, then the FILL suboption will be set to the same value as the LINE suboption.

LINE=NONE | MEAN | SD

produces a contour line plot for either the values of the means or the standard deviations. LINE=MEAN is the default; however, if you do not specify the LINE suboption the behavior depends on the FILL suboption as follows: If you specify FILL=NONE or do not specify the FILL suboption at all, then the LINE suboption will be set to its default value. If FILL=PRED or FILL=SE, then the LINE suboption will be set to the same value as the FILL suboption.

OBS=*obs-options*

produces an overlaid scatter plot of the observations in addition to the specified contour plots. The following *obs-options* are available:

GRAD

specifies that observations be displayed as circles colored by the observed measurement. The same color gradient is used to display the means surface and the observations. The conditional simulation honors the observed values, so the means surface at the observation locations has the same color as the corresponding observations.

LINEGRAD

is the same as OBS=GRAD except that a border is shown around each observation.

NONE

specifies that no observations be displayed.

OUTL

specifies that observations be displayed as circles with a border but with a completely transparent fill.

OBS=NONE is the default.

If you specify multiple instances of the FILL, LINE, or OBS suboptions in the same MEANS option, then the resulting means plot will honor the last value specified for any of the suboptions.

Any combination where you specify FILL=NONE and LINE=NONE is not available.

The MEANS option requires that simulations be requested at more than one location on a rectangular grid. The plot is not available when you use the GRIDDATA= data set option in the GRID statement, because the GRIDDATA= option allows you to specify any coordinates for the simulation locations.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC SIM2D to obtain separate analyses on observations in groups defined by the BY variables. Hence, in PROC SIM2D it makes sense to use the BY statement when you request conditional simulations where observations are involved. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement options NOTSORTED or DESCENDING in the BY statement for the SIM2D procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

The use of the BY statement assumes that you have specified an input data set with the DATA= option in the PROC SIM2D statement. Note that in PROC SIM2D if you do not specify the VAR= option in the SIMULATE statement, then this is a request for unconditional simulation even if you have specified the DATA= option in the PROC SIM2D statement. Therefore, it is possible to specify the BY statement and request mixed types of simulations by specifying multiple SIMULATE statements in the same PROC SIM2D step.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

COORDINATES Statement

COORDINATES *coordinate-variables* **;**

The following two options give the name of the variables in the DATA= data set containing the values of the x and y coordinates of the conditioning data.

Only one COORDINATES statement is allowed, and it is applied to all SIMULATE statements that have the VAR= specification. In other words, it is assumed that all the VAR= variables in all SIMULATE statements have the same x and y coordinates.

You can abbreviate the COORDINATES statement as COORD.

XCOORD=*(variable-name)*

XC=*(variable-name)*

gives the name of the variable containing the x coordinate of the data in the DATA= data set.

YCOORD=*(variable-name)*

YC=*(variable-name)*

gives the name of the variable containing the y coordinate of the data locations in the DATA= data set.

GRID Statement

GRID *grid-options* **;**

The following options can be used to specify the grid of spatial locations at which to perform the simulations. A single GRID statement is required and is applied to all SIMULATE statements.

There are two basic methods for specifying the grid. You can specify the x and y coordinates explicitly, or they can be read from a SAS data set.

The options for the explicit specification of grid locations are as follows.

X=*number*

X=$x_1, \ldots, x_m$

X=x_1 *to* x_m

X=x_1 *to* x_m *by* δx

specifies the x coordinate of the grid locations.

Y=*number*

Y=$y_1, \ldots, y_m$

Y=y_1 *to* y_m

Y=y_1 *to* y_m *by* δy

specifies the y coordinate of the grid locations.

For example, the following two GRID statements are equivalent:

```
grid x=1,2,3,4,5  y=0,2,4,6,8,10;

grid x=1 to 5 y=0 to 10 by 2;
```

To specify grid locations from a SAS data set, you must provide the name of the data set and the variables containing the values of the x and y coordinates.

GRIDDATA=*SAS-data-set*

GDATA=*SAS-data-set*

specifies a SAS data set containing the x and y grid coordinates.

XCOORD=*(variable-name)*

XC=*(variable-name)*

gives the name of the variable containing the x coordinate of the grid locations in the GRIDDATA= data set.

YCOORD=*(variable-name)*

YC=*(variable-name)*

gives the name of the variable containing the y coordinate of the grid locations in the GRIDDATA= data set.

SIMULATE Statement

SIMULATE *simulate-options* **;**

The SIMULATE statement specifies details on the simulation and the covariance model used in the simulation. You can specify the following options with a SIMULATE statement, which can be abbreviated by SIM.

NUMREAL=*number*

NUMR=*number*

NR=*number*

specifies the number of realizations to produce for the spatial process specified by the covariance model. Note that the number of observations in the OUTSIM= data set contributed by a given SIMULATE statement is the product of the NUMREAL= value with the number of grid points. This can cause the OUTSIM= data set to become large even for moderate values of the NUMREAL= option.

VAR=*(variable-name)*

specifies the single numeric variable used as the conditioning variable in the simulation. In other words, the simulation is conditional on the values of the VAR= variable found in the DATA= data set. If you omit the VAR= option or if all observations of the VAR= variable are missing values, then the simulation is *unconditional*. Since multiple SIMULATE statements are allowed, you can perform both unconditional and conditional simulations with a single PROC SIM2D statement.

Covariance Model Specification

There are two ways to specify a semivariogram or covariance model. In the first method, you can specify the required parameters SCALE, RANGE, and FORM, and possibly the optional parameters NUGGET, ANGLE, and RATIO, explicitly in the SIMULATE statement.

In the second method, you can specify an MDATA= data set. This data set contains variables corresponding to the required SCALE, RANGE, and FORM parameters, and, optionally, variables for the NUGGET, ANGLE, and RATIO parameters.

The two methods are exclusive; either you specify all parameters explicitly, or they are all are read from the MDATA= data set.

ANGLE=*angle*

ANGLE=*(angle*$_1$*,...,angle*$_k$*)*

specifies the angle of the major axis for anisotropic models, measured in degrees clockwise from the N–S axis. In the case of a nested semivariogram model, you can specify an angle for each nesting. The default is ANGLE=0.

FORM=*form_spec*

FORM=*(form_spec*$_1$*, form_spec*$_2$*,...,form_spec*$_k$*)*

specifies the functional form or forms of the semivariogram model, where *form_spec* can take only the values SPHERICAL, EXPONENTIAL, and GAUSSIAN. The two ways of specifying the FORM= parameter allow specification of both nested and non-nested models.

The following abbreviations are permitted. For the spherical model, you can specify the *form_spec* as FORM=SPHERICAL, FORM=SPH, or FORM=S. For the exponential model, you can specify the *form_spec* as FORM=EXPONENTIAL, FORM=EXP, or FORM=E. For the Gaussian model, you can specify the *form_spec* as FORM=GAUSSIAN, FORM=GAUSS, or FORM=G.

MDATA=*SAS-data-set*

specifies the input data set that contains parameter values for the covariance or semivariogram model. The MDATA= data set must contain variables named SCALE, RANGE, and FORM, and it can optionally contain the variables NUGGET, ANGLE, and RATIO.

The FORM variables must be character, and they can assume the same values allowed in the explicit FORM= syntax described previously. The RANGE and SCALE variables must be numeric. The optional variables ANGLE, RATIO, and NUGGET must also be numeric if present.

The number of observations present in the MDATA= data set corresponds to the level of nesting of the covariance or semivariogram model. For example, to specify a non-nested model that uses a spherical covariance, an MDATA= data set might contain the following statements:

```
data md1;
   input scale range form $;
   datalines;
   25 10 SPH
run;
```

The PROC SIM2D statement to use the MDATA= specification is of the form shown in the following:

```
proc sim2d data=...;
   sim var=.... mdata=md1;
run;
```

This is equivalent to the following explicit specification of the covariance model parameters:

```
proc sim2d data=...;
   sim var=.... scale=25 range=10 form=sph;
run;
```

The following MDATA= data set is an example of an anisotropic nested model:

```
data md2;
   input scale range form $ nugget angle ratio;
   datalines;
   20 8 S  5 35 .7
   12 3 G  5 0  .8
   4  1 G  5 45 .5
   ;

proc sim2d data=...;
   sim var=.... mdata=md2;
run;
```

This is equivalent to the following explicit specification of the covariance model parameters:

```
proc sim2d data=...;
   sim var=.... scale=(20,12,4) range=(8,3,1) form=(S,G,G)
                angle=(35,0,45) ratio=(.7,.8,.5) nugget=5;
run;
```

This example is somewhat artificial in that it is usually hard to detect different anisotropy directions and ratios for different nestings by using an experimental semivariogram. Note that the NUGGET value is the same for all nestings. This is always the case; the nugget effect is a single additive term for all models. For further details, refer to the section "The Nugget Effect" on page 2954 in the KRIGE2D procedure.

The SIMULATE statement can be given a label. This is useful for identification in the OUTSIM= data set when multiple SIMULATE statements are specified. For example:

```
proc sim2d data=...;
   gauss1: sim var=.... form=gauss;
   mean ....;
   gauss2: sim var=.... form gauss;
   mean ....;
   exp1: sim var=.... form=exp;
   mean ....;
   exp2: sim var=.... form=exp;
   mean ....;
run;
```

In the OUTSIM= data set, the values "GAUSS1," "GAUSS2," "EXP1," and "EXP2" for the LABEL variable help to identify the realizations corresponding to the four SIMULATE statements. If you do not provide a label for a SIMULATE statement, a default label of SIM*n* is given, where *n* is the number of unlabeled SIMULATE statements seen so far.

NUGGET=*number*

specifies the nugget effect for the model. This effect is due to a discontinuity in the semivariogram as determined by plotting the sample semivariogram (refer to the section "The Nugget Effect" on page 2954 in the KRIGE2D procedure for details). For models without any nugget effect, the NUGGET= option is left out. The default is NUGGET=0.

RANGE=*range*

RANGE=*(range*$_1$*,. . . ,range*$_k$*)*

specifies the range parameter in the semivariogram models. In the case of a nested semivariogram model, you must specify a range for each nesting.

The range parameter is the divisor in the exponent in all supported models. It has the units of distance or distance squared for these models, and it is related to the correlation scale for the underlying spatial process.

Refer to the section "Theoretical Semivariogram Models" on page 2947 in the KRIGE2D procedure for details on how the RANGE= values are determined.

RATIO=*ratio*

RATIO=*(ratio*$_1$*,. . . ,ratio*$_k$*)*

specifies the ratio of the length of the minor axis to the length of the major axis for anisotropic models. The value of the RATIO= option must be between 0 and 1. In the case of a nested semivariogram model, you can specify a ratio for each nesting. The default is RATIO=1.

SCALE=*scale*

SCALE=*(scale*$_1$*,. . . ,scale*$_k$*)*

specifies the scale (or *sill*) parameter in semivariogram models. In the case of a nested semivariogram model, you must specify a scale for each nesting. The scale parameter is the multiplicative factor in all supported models; it has the same units as the variance of the VAR= variable.

Refer to the section "Theoretical Semivariogram Models" on page 2947 in the KRIGE2D procedure for details on how the SCALE= values are determined.

SEED=*seed value*

specifies the seed to use for the random number generator. The SEED= value has to be an integer.

SINGULAR=*number*

gives the singularity criteria for solving the set of linear equations involved in the computation of the mean and covariance of the conditional distribution associated with a given SIMULATE statement. The larger the value of the SINGULAR= option, the easier it is for the covariance matrix system to be declared singular. The default is SINGULAR=1E–8.

For more details on the use of the SINGULAR= option, see the section "Computational and Theoretical Details of Spatial Simulation" on page 6176.

MEAN Statement

MEAN *spec*$_1$,...,*spec*$_6$ **;**

MEAN QDATA= *SAS-data-set CONST=var*$_1$ *CX=var*$_2$ *CY=var*$_3$
CXX=var$_4$ *CYY=var*$_5$ *CXY=var*$_6$ **;**

MEAN QDATA= *SAS-data-set* **;**

A mean function $\mu(s)$ that is a quadratic in the coordinates can be written

$$\mu(s) = \mu(x, y) = \beta_0 + \beta_1 x + \beta_2 y + \beta_3 x^2 + \beta_4 y^2 + \beta_5 xy$$

The MEAN statement is used to specify the quadratic surface to use as the mean function for the simulated SRF. There are three ways to specify the MEAN statement. The MEAN statement allows the specification of the coefficients $\beta_0, \cdots, \beta_5$ either explicitly or through a QDATA= data set.

An example of an explicit specification is the following:

```
mean 1.4 + 2.5*x + 3.6*y + 0.47*x*x + 0.58*y*y + 0.69*x*y;
```

In this example, all terms have a nonzero coefficient. Any term with a zero coefficient is simply left out of the specification. For example,

```
mean 1.4;
```

is a valid quadratic form with all terms having zero coefficients except the constant term.

An equivalent way of specifying the mean function is through the QDATA= data set. For example, the MEAN statement

```
mean 1.4 + 2.5*x + 3.6*y + 0.47*x*x + 0.58*y*y + 0.69*x*y;
```

can be alternatively specified by the following DATA step and MEAN statement:

```
data q1;
   input c1 c2 c3 c4 c5 c6;
   datalines;
   1.4 2.5 3.6 0.47 0.58 0.69
run;

proc sim2d data=....;
   simulate ...;
   mean qdata=q1 const=c1 cx=c2 cy=c3 cxx=c4
            cyy=c5 cxy=c6;
run;
```

The QDATA= data set specifies the data set containing the coefficients. The parameters CONST=, CX=, CY=, CXX=, CYY=, and CYX= specify the variables in the QDATA= data set that correspond to the constant, linear x, linear y, and so on. For any coefficient not specified in this list, the

QDATA= data set is checked for the presence of variables with default names of CONST, CX, CY, CXX, CYY, and CXY. If these variables are present, their values are taken as the corresponding coefficients. Hence, you can rewrite the previous example as follows:

```
data q1;
   input const cx cy cxx cyy cxy;
   datalines;
   1.4 2.5 3.6 0.47 0.58 0.69
   ;

proc sim2d data=....;
   simulate ...;
   mean qdata=q1;
run;
```

If a given coefficient does not appear in the list or in the data set with the default name, a value of zero is assumed.

Details: SIM2D Procedure

Computational and Theoretical Details of Spatial Simulation

Introduction

There are a number of approaches to simulating spatial random fields or, more generally, simulating sets of dependent random variables. This includes sequential indicator methods, turning bands, and the Karhunen-Loeve Expansion. Refer to Christakos (1992, Chapter 8) and Deutsch and Journel (1992, Chapter V) for details.

A particularly simple method available for Gaussian spatial random fields is the LU decomposition method. This method is computationally efficient. For a given covariance matrix, the $LU = \mathbf{L}\mathbf{L}^T$ decomposition is computed once, and the simulation proceeds by repeatedly generating a vector of independent $N(0, 1)$ random variables and multiplying by the $\mathbf{L}$ matrix.

One problem with this technique is memory requirements; memory is required to hold the full data and grid covariance matrix in core. While this is especially limiting in the three-dimensional case, you can use PROC SIM2D, which handles only two-dimensional data, for moderately sized simulation problems.

Theoretical Development

It is a simple matter to produce an $N(0, 1)$ random number, and by stacking k $N(0, 1)$ random numbers in a column vector, you can obtain a vector with independent standard normal components

$\mathbf{W} \sim N_k(\mathbf{0}, \mathbf{I})$. The meaning of the terms *independence* and *randomness* in the context of a deterministic algorithm required for the generation of these numbers is subtle; refer to Knuth (1981, Chapter 3) for details.

Rather than $\mathbf{W} \sim N_k(\mathbf{0}, \mathbf{I})$, what is required is the generation of a vector $\mathbf{Z} \sim N_k(\mathbf{0}, \mathbf{C})$—that is,

$$\mathbf{Z} = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_k \end{bmatrix}$$

with covariance matrix

$$\mathbf{C} = \begin{pmatrix} C_{11} & C_{12} & \cdots & C_{1k} \\ C_{21} & C_{22} & \cdots & C_{2k} \\ & \ddots & & \\ C_{k1} & C_{k2} & \cdots & C_{kk} \end{pmatrix}$$

If the covariance matrix is symmetric and positive definite, it has a Cholesky root $\mathbf{L}$ such that $\mathbf{C}$ can be factored as

$$\mathbf{C} = \mathbf{L}\mathbf{L}^T$$

where $\mathbf{L}$ is lower triangular. Refer to Ralston and Rabinowitz (1978, Chapter 9, Section 3-3) for details. This vector $\mathbf{Z}$ can be generated by the transformation $\mathbf{Z} = \mathbf{L}\mathbf{W}$. Note that this is where the assumption of a Gaussian SRF is crucial. When $\mathbf{W} \sim N_k(\mathbf{0}, \mathbf{I})$, then $\mathbf{Z} = \mathbf{L}\mathbf{W}$ is also Gaussian. The mean of $\mathbf{Z}$ is

$$E(\mathbf{Z}) = \mathbf{L}(E(\mathbf{W})) = \mathbf{0}$$

and the variance is

$$\text{Var}(\mathbf{Z}) = \text{Var}(\mathbf{L}\mathbf{W}) = E(\mathbf{L}\mathbf{W}\mathbf{W}^T\mathbf{L}^T) = \mathbf{L}E(\mathbf{W}\mathbf{W}^T)\mathbf{L}^T = \mathbf{L}\mathbf{L}^T = \mathbf{C}$$

Consider now an SRF $Z(\boldsymbol{s}), \boldsymbol{s} \in D \subset \mathcal{R}^2$, with spatial covariance function $C(\boldsymbol{h})$. Fix locations $\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_k$, and let $\mathbf{Z}$ denote the random vector

$$\mathbf{Z} = \begin{bmatrix} Z(\boldsymbol{s}_1) \\ Z(\boldsymbol{s}_2) \\ \vdots \\ Z(\boldsymbol{s}_k) \end{bmatrix}$$

with corresponding covariance matrix

$$\mathbf{C}_z = \begin{pmatrix} C(\mathbf{0}) & C(\boldsymbol{s}_1 - \boldsymbol{s}_2) & \cdots & C(\boldsymbol{s}_1 - \boldsymbol{s}_k) \\ C(\boldsymbol{s}_2 - \boldsymbol{s}_1) & C(\mathbf{0}) & \cdots & C(\boldsymbol{s}_2 - \boldsymbol{s}_k) \\ & \ddots & & \\ C(\boldsymbol{s}_k - \boldsymbol{s}_1) & C(\boldsymbol{s}_k - \boldsymbol{s}_2) & \cdots & C(\mathbf{0}) \end{pmatrix}$$

Since this covariance matrix is symmetric and positive definite, it has a Cholesky root, and the $Z(\boldsymbol{s}_i), i = 1, \cdots, k$, can be simulated as described previously. This is how the SIM2D procedure implements unconditional simulation in the zero-mean case. More generally,

$$Z(\boldsymbol{s}) = \mu(\boldsymbol{s}) + \varepsilon(\boldsymbol{s})$$

with $\mu(\boldsymbol{s})$ being a quadratic form in the coordinates $\boldsymbol{s} = (x, y)$, and the $\varepsilon(\boldsymbol{s})$ being an SRF having the same covariance matrix $\mathbf{C}_z$ as previously. In this case, the $\mu(\boldsymbol{s}_i), i = 1, \cdots, k$, is computed once and added to the simulated vector $\varepsilon(\boldsymbol{s}_i), i = 1, \cdots, k$, for each realization.

For a conditional simulation, this distribution of

$$\mathbf{Z} = \begin{bmatrix} Z(\boldsymbol{s}_1) \\ Z(\boldsymbol{s}_2) \\ \vdots \\ Z(\boldsymbol{s}_k) \end{bmatrix}$$

must be conditioned on the observed data. The relevant general result concerning conditional distributions of multivariate normal random variables is the following. Let $\mathbf{X} \sim N_m(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where

$$\mathbf{X} = \begin{bmatrix} \mathbf{X}_1 \\ \mathbf{X}_2 \end{bmatrix}$$

$$\boldsymbol{\mu} = \begin{bmatrix} \boldsymbol{\mu}_1 \\ \boldsymbol{\mu}_2 \end{bmatrix}$$

$$\boldsymbol{\Sigma} = \begin{pmatrix} \boldsymbol{\Sigma}_{11} & \boldsymbol{\Sigma}_{12} \\ \boldsymbol{\Sigma}_{21} & \boldsymbol{\Sigma}_{22} \end{pmatrix}$$

The subvector $\mathbf{X}_1$ is $k \times 1$, $\mathbf{X}_2$ is $n \times 1$, $\boldsymbol{\Sigma}_{11}$ is $k \times k$, $\boldsymbol{\Sigma}_{22}$ is $n \times n$, and $\boldsymbol{\Sigma}_{12} = \boldsymbol{\Sigma}_{21}^T$ is $k \times n$, with $k + n = m$. The full vector $\mathbf{X}$ is partitioned into two subvectors, $\mathbf{X}_1$ and $\mathbf{X}_2$, and $\boldsymbol{\Sigma}$ is similarly partitioned into covariances and cross covariances.

With this notation, the distribution of $\mathbf{X}_1$ conditioned on $\mathbf{X}_2 = \boldsymbol{x}_2$ is $N_k(\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Sigma}})$, with

$$\tilde{\boldsymbol{\mu}} = \boldsymbol{\mu}_1 + \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)$$

and

$$\tilde{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}_{11} - \boldsymbol{\Sigma}_{12}\boldsymbol{\Sigma}_{22}^{-1}\boldsymbol{\Sigma}_{21}$$

Refer to Searle (1971, pp. 46–47) for details. The correspondence with the conditional spatial simulation problem is as follows. Let the coordinates of the observed data points be denoted $\tilde{\boldsymbol{s}}_1, \tilde{\boldsymbol{s}}_2, \cdots, \tilde{\boldsymbol{s}}_n$, with values $\tilde{z}_1, \tilde{z}_2, \cdots, \tilde{z}_n$. Let $\tilde{\mathbf{Z}}$ denote the random vector

$$\tilde{\mathbf{Z}} = \begin{bmatrix} Z(\tilde{\boldsymbol{s}}_1) \\ Z(\tilde{\boldsymbol{s}}_2) \\ \vdots \\ Z(\tilde{\boldsymbol{s}}_n) \end{bmatrix}$$

The random vector $\tilde{\mathbf{Z}}$ corresponds to $\mathbf{X}_2$, while $\mathbf{Z}$ corresponds to $\mathbf{X}_1$. Then $\left(\mathbf{Z} \mid \tilde{\mathbf{Z}} = \tilde{\mathbf{z}}\right) \sim N_k(\tilde{\boldsymbol{\mu}}, \tilde{\mathbf{C}})$ as in the previous distribution. The matrix

$$\tilde{\mathbf{C}} = \mathbf{C}_{11} - \mathbf{C}_{12}\mathbf{C}_{22}^{-1}\mathbf{C}_{21}$$

is again positive definite, so a Cholesky factorization can be performed.

The dimension n for $\tilde{\mathbf{Z}}$ is simply the number of nonmissing observations for the VAR= variable; the values $\tilde{z}_1, \tilde{z}_2, \cdots, \tilde{z}_n$ are the values of this variable. The coordinates $\tilde{\boldsymbol{s}}_1, \tilde{\boldsymbol{s}}_2, \cdots, \tilde{\boldsymbol{s}}_n$ are also found in the DATA= data set, with the variables corresponding to the x and y coordinates identified in the COORDINATES statement. Note that all VAR= variables use the same set of conditioning coordinates; this fixes the matrix $\mathbf{C}_{22}$ for all simulations.

The dimension k for $\mathbf{Z}$ is the number of grid points specified in the GRID statement. Since there is a single GRID statement, this fixes the matrix $\mathbf{C}_{11}$ for all simulations. Similarly, $\mathbf{C}_{12}$ is fixed.

The Cholesky factorization $\tilde{\mathbf{C}} = \mathbf{L}\mathbf{L}^T$ is computed once, as is the mean correction

$$\tilde{\boldsymbol{\mu}} = \boldsymbol{\mu}_1 + \mathbf{C}_{12}\mathbf{C}_{22}^{-1}(\boldsymbol{x}_2 - \boldsymbol{\mu}_2)$$

Note that the means $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are computed using the grid coordinates $\boldsymbol{s}_1, \boldsymbol{s}_2, \cdots, \boldsymbol{s}_k$, the data coordinates $\tilde{\boldsymbol{s}}_1, \tilde{\boldsymbol{s}}_2, \cdots, \tilde{\boldsymbol{s}}_n$, and the quadratic form specification from the MEAN statement. The simulation is now performed exactly as in the unconditional case. A $k \times 1$ vector of independent standard $N(0, 1)$ random variables is generated and multiplied by $\mathbf{L}$, and $\tilde{\boldsymbol{\mu}}$ is added to the transformed vector. This is repeated N times, where N is the value specified for the NR= option.

Computational Details

In the computation of $\tilde{\boldsymbol{\mu}}$ and $\boldsymbol{\Sigma}$ described in the previous section, the inverse $\boldsymbol{\Sigma}_{22}^{-1}$ is never actually computed; an equation of the form

$$\boldsymbol{\Sigma}_{22}\mathbf{A} = \mathbf{B}$$

is solved for $\mathbf{A}$ by using a modified Gaussian elimination algorithm that takes advantage of the fact that $\boldsymbol{\Sigma}_{22}$ is symmetric with constant diagonal $C_z(0)$ that is larger than all off-diagonal elements. The SINGULAR= option pertains to this algorithm. The value specified for the SINGULAR= option is scaled by $C_z(0)$ before comparison with the pivot element.

Memory Usage

For conditional simulations, the largest matrix held in core at any one time depends on the number of grid points and data points. Using the previous notation, the data-data covariance matrix $\mathbf{C}_{22}$ is $n \times n$, where n is the number of nonmissing observations for the VAR= variable in the DATA= data set. The grid-data cross covariance $\mathbf{C}_{12}$ is $n \times k$, where k is the number of grid points. The grid-grid covariance $\mathbf{C}_{11}$ is $k \times k$. The maximum memory required at any one time for storing these matrices is

$$\max\left(k(k+1), n(n+1) + 2(n \times k)\right) \times \text{sizeof(double)}$$

There are additional memory requirements that add to the total memory usage, but usually these matrix calculations dominate, especially when the number of grid points is large.

Output Data Set

The SIM2D procedure produces a single output data set: the OUTSIM=*SAS-data-set*. The OUTSIM= data set contains all the needed information to uniquely identify the simulated values.

The OUTSIM= data set contains the following variables:

- LABEL, which is the label for the current SIMULATE statement
- VARNAME, which is the name of the conditioning variable for the current SIMULATE statement
- _ITER_, which is the iteration number within the current SIMULATE statement
- GXC, which is the x coordinate for the current grid point
- GYC, which is the y coordinate for the current grid point
- SVALUE, which is the value of the simulated variable

If you specify the NARROW option in the PROC SIM2D statement, the LABEL and VARNAME variables are not included in the OUTSIM= data set. This option is useful in the case where the number of data points, grid points, and realizations are such that they generate a very large OUTSIM= data set. The size of the OUTSIM= data set is reduced when these variables are not included.

In the case of an unconditional simulation, the VARNAME variable is not included. In the case of mixed conditional and unconditional simulations (that is, when multiple SIMULATE statements are specified, and one or more contain a VAR= specification and one or more do *not* contain a VAR= specification), the VARNAME variable is included but is given a missing value for those observations corresponding to an unconditional simulation.

Displayed Output

In addition to the output data set, the SIM2D procedure produces output objects as well. The SIM2D procedure output objects are the following:

- a default "Number of Observations" table that displays the number of observations read from the input data set and the number of observations used in the analysis
- a map showing the spatial distribution of the observations of the current VAR variable in the SIMULATE statement, in the case of conditional simulations. The observations are displayed by default with circled markers whose color indicates the VAR value at the corresponding location.
- a default table for each SIMULATE statement that summarizes the simulation specifications
- a default table for each SIMULATE statement that shows the covariance model parameters for the corresponding simulation
- plots of the mean and standard deviation of the simulation realizations at each point of the specified output grid. You can produce more than one of these plots for every SIMULATE statement with styles that you can specify by using the available suboptions of the PLOTS=MEANS option.

ODS Table Names

Each table created by PROC SIM2D has a name associated with it, and you must use this name to reference the table when using ODS Graphics. These names are listed in Table 79.2.

Table 79.2 ODS Tables Produced by PROC SIM2D

ODS Table Name	Description	Statement	Option
ModelInfo	Parameters of the covariance model used in current simulation	PROC	default output
NObs	Number of observations read and used	PROC	default output
SimuInfo	General information about the simulation	PROC	default output

ODS Graphics

This section describes the use of the Output Delivery System (ODS) for creating graphics with the SIM2D procedure.

To request these graphs, you must specify the ODS GRAPHICS statement. For additional control of the graphics that are displayed, see the PLOTS= option in the section "PROC SIM2D Statement" on page 6165. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

ODS Graph Names

PROC SIM2D assigns a name to each graph it creates by using ODS Graphics. You can use these names to reference the graphs when using ODS Graphics. The names are listed in Table 79.3.

Table 79.3 ODS Graphics Produced by PROC SIM2D

ODS Graph Name	Plot Description	Statement	Option
MeansPlot	Contour plot of the simulated means, surface of the standard deviation, and outlines of the observation locations	PROC	PLOTS=MEANS
ObservationsPlot	Scatter plot of observed data and colored markers indicating observed values	PROC	PLOTS=OBSERV

To request this graph, you must specify the ODS GRAPHICS statement in addition to the statements indicated in Table 79.3. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Examples: SIM2D Procedure

Example 79.1: Simulation

Continuing with the coal seam thickness example from the section "Getting Started: SIM2D Procedure" on page 6157, this example asks a more complicated question. This question is economic in nature, and the (approximate) answer requires the use of simulation.

Simulating a Subregion for Economic Feasibility

The coal seam must be of a minimum thickness, called a *cutoff value*, for a mining operation to be profitable. Suppose that, for a subregion of the measured area, the cost of mining is higher than in the remaining areas due to the geology of the overburden. This higher cost results in a higher thickness cutoff value for the subregion. Suppose also that it is determined from a detailed cost analysis that at least 60% of the subregion must exceed a seam thickness of 39.7 feet for profitability.

How can you use the SRF model (μ and $C_z(\boldsymbol{s})$) and the measured seam thickness values $Z(\boldsymbol{s}_i), i = 1, \cdots, 75$, to determine, in some approximate way, if at least 60% of the subregion exceeds this minimum?

Spatial prediction does not appear to be helpful in answering this question. While it is easy to determine whether a predicted value at a location in the subregion is above the 39.7-feet cutoff value, it is not clear how to incorporate the standard error associated with the predicted value. The standard error is what characterizes the stochastic nature of the prediction (and the underlying SRF). It is clear that it must be included in any realistic approach to the problem.

A conditional simulation, on the other hand, seems to be a natural way of obtaining an approximate answer. By simulating the SRF on a sufficiently fine grid in the subregion, you can determine the proportion of grid points in which the mean value over realizations exceeds the 39.7-feet cutoff and compare it with the 60% value needed for profitability.

It is desirable in any simulation study that the quantity being estimated (in this case, the proportion exceeding the 39.7-feet cutoff) not depend on the number of simulations performed. For example, suppose that the maximum seam thickness is simulated. It is likely that the maximum value increases as the number of simulations performed increases. Hence, a simulation is not useful for such an estimate. A simulation is useful for determining the *distribution* of the maximum, but there are general theoretical results for such distributions, making such a simulation unnecessary. Refer to Leadbetter, Lindgren, and Rootzen (1983) for details.

In the case of simulating the proportion exceeding the 39.7-feet cutoff, it is expected that this quantity will settle down to a fixed value as the number of realizations increases. At a fixed grid point, the quantity being compared with the cutoff value is the mean over all simulated realizations; this mean value settles down to a fixed number as the number of realizations increases. In the same manner, the proportion of the grid where the mean values exceed the cutoff also becomes constant. This can be tested using PROC SIM2D.

A crucial, nonprovable assumption in applying SRF theory to the coal seam thickness data is that the values $Z(\boldsymbol{s}_i), i = 1, \cdots, 75$, represent a *single* realization from the set of all possible realizations consistent with the SRF model (μ and $C_z(\boldsymbol{h})$). A conditional simulation repeatedly produces other possible simulated realizations consistent with the model and data. However, the only concern of the mining company is this single unique realization. It is not concerned about similar coal fields to be mined sometime in the future; it might never see another coal field remotely similar to this one, or it might not be in business in the future.

Hence the proportion found by generating repeated simulated realizations must somehow relate back to the unique realization that is the coal field (seam thickness). This is done by interpreting the proportion found from a simulation to the spatial mean proportion for the unique realization. The term "spatial mean" is simply an appropriate integral over the fixed (but unknown) spatial function $z(\boldsymbol{s})$. (The SRF is denoted $Z(\boldsymbol{s})$; a particular realization, a deterministic function of the spatial coordinates, is denoted $z(\boldsymbol{s})$.)

This interpretation requires an ergodic assumption, which is also needed in the original estimation of $C_z(\boldsymbol{s})$. Refer to Cressie (1993, pp. 53–58) for a discussion of ergodicity and Gaussian SRFs.

Implementation Using PROC SIM2D

The subregion to be considered is the southeast corner of the field, which is a square region with a length of 40 distance units (in thousands of feet). First, you input the thickness data as the following DATA step shows:

```
data thick;
   input East North Thick @@;
   label Thick='Coal Seam Thickness';
   datalines;
 0.7  59.6  34.1   2.1  82.7  42.2   4.7  75.1  39.5
 4.8  52.8  34.3   5.9  67.1  37.0   6.0  35.7  35.9
 6.4  33.7  36.4   7.0  46.7  34.6   8.2  40.1  35.4
13.3   0.6  44.7  13.3  68.2  37.8  13.4  31.3  37.8
17.8   6.9  43.9  20.1  66.3  37.7  22.7  87.6  42.8
23.0  93.9  43.6  24.3  73.0  39.3  24.8  15.1  42.3
24.8  26.3  39.7  26.4  58.0  36.9  26.9  65.0  37.8
27.7  83.3  41.8  27.9  90.8  43.3  29.1  47.9  36.7
29.5  89.4  43.0  30.1   6.1  43.6  30.8  12.1  42.8
32.7  40.2  37.5  34.8   8.1  43.3  35.3  32.0  38.8
37.0  70.3  39.2  38.2  77.9  40.7  38.9  23.3  40.5
39.4  82.5  41.4  43.0   4.7  43.3  43.7   7.6  43.1
46.4  84.1  41.5  46.7  10.6  42.6  49.9  22.1  40.7
51.0  88.8  42.0  52.8  68.9  39.3  52.9  32.7  39.2
55.5  92.9  42.2  56.0   1.6  42.7  60.6  75.2  40.1
62.1  26.6  40.1  63.0  12.7  41.8  69.0  75.6  40.1
70.5  83.7  40.9  70.9  11.0  41.7  71.5  29.5  39.8
78.1  45.5  38.7  78.2   9.1  41.7  78.4  20.0  40.8
80.5  55.9  38.7  81.1  51.0  38.6  83.8   7.9  41.6
84.5  11.0  41.5  85.2  67.3  39.4  85.5  73.0  39.8
86.7  70.4  39.6  87.2  55.7  38.8  88.1   0.0  41.6
88.4  12.1  41.3  88.4  99.6  41.2  88.8  82.9  40.5
88.9   6.2  41.5  90.6   7.0  41.5  90.7  49.6  38.9
91.5  55.4  39.0  92.9  46.8  39.1  93.4  70.9  39.7
55.8  50.5  38.1  96.2  84.3  40.3  98.2  58.2  39.5
;
```

PROC SIM2D is run on the entire data set for conditioning, while the simulation grid covers only this subregion. It is convenient to be able to vary the seed, the grid increment, and the number of simulations performed. The following macro implements the computation of the percent area exceeding the cutoff value by using the seed, the grid increment, and the number of simulated realizations as macro arguments.

Within the macro, the data set produced by PROC SIM2D is transposed with PROC TRANSPOSE so that each grid location is a separate variable. The MEANS procedure is then used to average the simulated value at each grid point over all realizations. It is this average that is compared to the cutoff value. The last DATA step does the comparison, uses an appropriate loop to determine the percent of the grid locations that exceed this cutoff value, and writes the results to the listing file in the form of a report. This sequence is implemented with the following statements:

```
/* Construct macro for conditional simulation ------------------*/

%let cc0=7.2881;
%let aa0=30.6239;
%let ngt=1e-8;
%let form=gauss;
%let cut=39.7;

%macro area_sim(seed=,nr=,ginc=);

   %let ngrid=%eval(40/&ginc+1);
   %let tgrid=%eval(&ngrid*&ngrid);

   proc sim2d data=thick outsim=sim1;
      coordinates xc=east yc=north;
      simulate var=thick numreal=&nr seed=&seed
               scale=&cc0 range=&aa0 nugget=&ngt form=&form;
      mean 40.1173;
      grid x=60 to 100 by &ginc
           y= 0 to  40 by &ginc;
   run;

   proc transpose data=sim1 out=sim2 prefix=sims;
      by _iter_;
      var svalue;
   run;

   proc means data=sim2 noprint n mean;
      var sims1-sims&tgrid;
      output out=msim n=numsim mean=ms1-ms&tgrid;
   run;

   data _null_;
      file print;
      array simss ms1-ms&tgrid;
      set msim;

      cflag=0;
      do ss=1 to &tgrid;
         tempv=simss[ss];
         if simss[ss] > &cut then do;
            cflag + 1;
         end;
      end;

      area_per=100*(cflag/&tgrid);
      put // +5 'Conditional Simulation of Coal Seam'
                ' Thickness for Subregion';
      put / +5 'Subregion is South-East Corner 40 by 40 distance units';
      put / +5 "Seed:&seed" +2 "Grid Increment:&ginc";
      put / +5 "Total Number of Grid Points:&tgrid" +2
               "Number of Simulations:&nr";
```

```
      put / +5 "Percent of Subregion Exceeding Cutoff of %left(&cut) ft.:"
                +2 area_per 5.2;
   run;
%mend area_sim;
```

In the following statement, you invoke the macro three times. Each time, the macro is invoked with a different seed and combination of the grid increment and number of simulations. The macro is first invoked with a relatively coarse grid (grid increment of 10 distance units) and a small number of realizations (5). The output of this conditional simulation is shown in Output 79.1.1.

```
/* Execute macro for coarse grid ------------------------------*/
%area_sim(seed=12345,nr=5,ginc=10);
```

Output 79.1.1 Conditional Simulation of Coal Seam Thickness on a Coarse Grid

```
Conditional Simulation of Coal Seam Thickness for Subregion

Subregion is South-East Corner 40 by 40 distance units

Seed:12345  Grid Increment:10

Total Number of Grid Points:25  Number of Simulations:5

Percent of Subregion Exceeding Cutoff of 39.7 ft.:  76.00
```

The next invocation, in the following statement, uses a finer grid and 50 realizations. The output of the second conditional simulation is shown in Output 79.1.2.

```
/* Execute macro for fine grid and fewer simulations -----------*/
%area_sim(seed=54321,nr=50,ginc=1);
```

Output 79.1.2 Conditional Simulation of Coal Seam Thickness on a Fine Grid

```
Conditional Simulation of Coal Seam Thickness for Subregion

Subregion is South-East Corner 40 by 40 distance units

Seed:54321  Grid Increment:1

Total Number of Grid Points:1681  Number of Simulations:50

Percent of Subregion Exceeding Cutoff of 39.7 ft.:  76.74
```

The final invocation, in the following statement, uses the same grid increment and 500 realizations. The output of this conditional simulation is shown in Output 79.1.3.

```
/* Execute macro for fine grid and more simulations ------------*/
%area_sim(seed=655311,nr=500,ginc=1);
```

Output 79.1.3 Conditional Simulation of Coal Seam Thickness on a Fine Grid

```
Conditional Simulation of Coal Seam Thickness for Subregion

Subregion is South-East Corner 40 by 40 distance units

Seed:655311  Grid Increment:1

Total Number of Grid Points:1681  Number of Simulations:500

Percent of Subregion Exceeding Cutoff of 39.7 ft.:  76.68
```

The results from the preceding simulations indicate that 76% to 77% of the subregion exceeds the cutoff value.

Note also that the number of grid points in the simulation increases with the square of the decrease in the grid increment, leading to long CPU processing times. Increasing the number of realizations results in a linear increase in processing times. Hence, using as coarse a grid as possible allows for more realizations and experimentation with different seeds.

Example 79.2: Variability at Selected Locations

This example exhibits a more detailed investigation of the variation of simulated Thick values. You use the same thick data set from the section "Getting Started: SIM2D Procedure" on page 6157, and you are interested in the simulated values statistics at two selected grid points.

Specifically, you perform a simulation asking for 5,000 realizations (iterations) at two points of the region defined in the section "Preliminary Spatial Data Analysis" on page 6157. These are the extreme southwest point and a point toward the northeast corner of the region. Since you do not want to perform the simulation across the whole region, you need to produce a GDATA= data set to specify the coordinates of the selected points. These steps are implemented using the following DATA step and statements:

```
data thick;
   input East North Thick @@;
   label Thick='Coal Seam Thickness';
   datalines;
    0.7  59.6  34.1   2.1  82.7  42.2   4.7  75.1  39.5
    4.8  52.8  34.3   5.9  67.1  37.0   6.0  35.7  35.9
    6.4  33.7  36.4   7.0  46.7  34.6   8.2  40.1  35.4
   13.3   0.6  44.7  13.3  68.2  37.8  13.4  31.3  37.8
   17.8   6.9  43.9  20.1  66.3  37.7  22.7  87.6  42.8
   23.0  93.9  43.6  24.3  73.0  39.3  24.8  15.1  42.3
   24.8  26.3  39.7  26.4  58.0  36.9  26.9  65.0  37.8
   27.7  83.3  41.8  27.9  90.8  43.3  29.1  47.9  36.7
   29.5  89.4  43.0  30.1   6.1  43.6  30.8  12.1  42.8
   32.7  40.2  37.5  34.8   8.1  43.3  35.3  32.0  38.8
```

```
   37.0  70.3  39.2  38.2  77.9  40.7  38.9  23.3  40.5
   39.4  82.5  41.4  43.0   4.7  43.3  43.7   7.6  43.1
   46.4  84.1  41.5  46.7  10.6  42.6  49.9  22.1  40.7
   51.0  88.8  42.0  52.8  68.9  39.3  52.9  32.7  39.2
   55.5  92.9  42.2  56.0   1.6  42.7  60.6  75.2  40.1
   62.1  26.6  40.1  63.0  12.7  41.8  69.0  75.6  40.1
   70.5  83.7  40.9  70.9  11.0  41.7  71.5  29.5  39.8
   78.1  45.5  38.7  78.2   9.1  41.7  78.4  20.0  40.8
   80.5  55.9  38.7  81.1  51.0  38.6  83.8   7.9  41.6
   84.5  11.0  41.5  85.2  67.3  39.4  85.5  73.0  39.8
   86.7  70.4  39.6  87.2  55.7  38.8  88.1   0.0  41.6
   88.4  12.1  41.3  88.4  99.6  41.2  88.8  82.9  40.5
   88.9   6.2  41.5  90.6   7.0  41.5  90.7  49.6  38.9
   91.5  55.4  39.0  92.9  46.8  39.1  93.4  70.9  39.7
   55.8  50.5  38.1  96.2  84.3  40.3  98.2  58.2  39.5
   ;

data grid;
   input xc yc;
   datalines;
   0   0
   75  75
run;
```

Then, you run PROC SIM2D with the same parameters and characteristics as those in the section "Preliminary Spatial Data Analysis" on page 6157. This time, however, you ask for simulated values only at the two locations you specified in the previous DATA step. The following statements execute the requested simulation:

```
proc sim2d data=thick outsim=sim1;
   coordinates xc=East yc=North;
   simulate var=Thick numreal=5000 seed=79931
            scale=7.2881 range=30.6239 form=gauss;
   mean 40.1173;
   grid gdata=grid xc=xc yc=yc;
run;
```

Once the simulation is performed, summary statistics are computed for each of the specified grid points. In particular, you use PROC UNIVARIATE and a BY statement to request at these locations the quantiles and the extreme observations, as the following statements show:

```
proc sort data=sim1;
   by gxc gyc;
run;

proc univariate data=sim1;
   var svalue;
   by gxc gyc;
   ods select Quantiles ExtremeObs;
   title 'Simulation Statistics at Selected Grid Points';
run;
```

The summary statistics for the first grid point (East=0, North=0) are presented in Output 79.2.1.

Output 79.2.1 Simulation Statistics at Grid Point (East=0, North=0)

```
                Simulation Statistics at Selected Grid Points

------ X-coordinate of the grid point=0 Y-coordinate of the grid point=0 -------

                          The UNIVARIATE Procedure
                Variable:  SVALUE  (Simulated Value at Grid Point)

                          Quantiles (Definition 5)

                           Quantile      Estimate

                           100% Max       42.1810
                           99%            41.6986
                           95%            41.3633
                           90%            41.1890
                           75% Q3         40.9044
                           50% Median     40.5713
                           25% Q1         40.2191
                           10%            39.9463
                           5%             39.7552
                           1%             39.4201
                           0% Min         38.7473

------ X-coordinate of the grid point=0 Y-coordinate of the grid point=0 -------

                            Extreme Observations

                  ------Lowest-----        -----Highest-----

                     Value      Obs           Value      Obs

                   38.7473     2691         42.0656     1149
                   38.9080     1817         42.0805     3612
                   38.9449     3026         42.0980     3757
                   38.9867     2275         42.1782      135
                   39.0194     3100         42.1810     4536
```

Finally, Output 79.2.2 displays the summary statistics for the second grid point (East=75, North=75).

Output 79.2.2 Simulation Statistics at Grid Point (East=75, North=75)

```
                Simulation Statistics at Selected Grid Points

----- X-coordinate of the grid point=75 Y-coordinate of the grid point=75 ------

                           The UNIVARIATE Procedure
              Variable:  SVALUE  (Simulated Value at Grid Point)

                           Quantiles (Definition 5)

                            Quantile      Estimate

                            100% Max       40.1168
                            99%            40.1148
                            95%            40.1134
                            90%            40.1126
                            75% Q3         40.1114
                            50% Median     40.1101
                            25% Q1         40.1086
                            10%            40.1075
                            5%             40.1068
                            1%             40.1053
                            0% Min         40.1024

----- X-coordinate of the grid point=75 Y-coordinate of the grid point=75 ------

                             Extreme Observations

                  ------Lowest-----        -----Highest-----

                     Value      Obs           Value      Obs

                   40.1024     7176         40.1164     8980
                   40.1029     6262         40.1165     9272
                   40.1032     7383         40.1167     5676
                   40.1037     7156         40.1167     6514
                   40.1037     5643         40.1168     5329
```

References

Christakos, G. (1992), *Random Field Models in Earth Sciences*, New York: Academic Press.

Cressie, N. A. C. (1993), *Statistics for Spatial Data*, New York: John Wiley & Sons.

Deutsch, C. V. and Journel, A. G. (1992), *GSLIB: Geostatistical Software Library and User's Guide*, New York: Oxford University Press.

Knuth, D. E. (1981), *The Art of Computer Programming: Seminumerical Algorithms*, Vol. 2, Second Edition, Reading, MA: Addison-Wesley.

Leadbetter, M. R., Lindgren, G., and Rootzen, H. (1983), *Extremes and Related Properties of Random Sequences and Processes*, New York: Springer-Verlag.

Ralston, A. and Rabinowitz, P. (1978), *A First Course in Numerical Analysis*, New York: McGraw-Hill.

Searle, S. R. (1971), *Linear Models*, New York: John Wiley & Sons.

Chapter 80

The SIMNORMAL Procedure

Contents

Overview: SIMNORMAL Procedure **6193**
Getting Started: SIMNORMAL Procedure **6194**
Syntax: SIMNORMAL Procedure **6202**
PROC SIMNORMAL Statement 6202
BY Statement 6204
CONDITION Statement 6205
VAR Statement 6205
OUT= Output Data Set 6205
Computational Details: SIMNORMAL Procedure **6206**
Introduction 6206
Unconditional Simulation 6206
Conditional Simulation 6207
References **6209**

Overview: SIMNORMAL Procedure

The SIMNORMAL procedure can perform conditional and unconditional simulation for a set of correlated normal or Gaussian random variables.

The means, variances, and covariances (or correlations) are read from an input TYPE=CORR or TYPE=COV data set. This data set is typically produced by the CORR procedure. Conditional simulations are performed by appending a special observation, identified by the value of 'COND' for the _TYPE_ variable, which contains the conditioning value.

The output data set from PROC SIMNORMAL contains simulated values for each of the analysis variables. Optionally, the output data set also contains the seed stream and the values of the conditioning variables. PROC SIMNORMAL produces no printed output.

Getting Started: SIMNORMAL Procedure

The following example illustrates the use of PROC SIMNORMAL to generate variable values conditioned on a set of related or correlated variables.

Suppose you are given a sample of size 50 from ten normally distributed, correlated random variables, $IN_{1,i}, \cdots, IN_{5,i}, OUT_{1,i}, \cdots, OUT_{5,i}, i = 1, \cdots, 50$. The first five variables represent input variables for a chemical manufacturing process, and the last five are output variables.

First, the data are input and the correlation structure is determined by using PROC CORR, as in the following statements. The results are shown in Figure 80.1.

```
data a ;
   input in1-in5 out1-out5 ;
   datalines ;
 9.3500    10.0964     7.3177    10.3617    10.3444    9.4612
10.7443     9.9026     9.0144    11.7968

... more lines ...

 8.9174     9.9623     9.5742     9.9713
run ;

proc corr data=a cov nocorr outp=outcov ;
  var in1-in5 out1-out5 ;
run ;
```

Figure 80.1 Correlation of Chemical Process Variables

```
                          The CORR Procedure

  10  Variables:    in1      in2      in3      in4      in5      out1     out2
                    out3     out4     out5
```

Figure 80.1 *continued*

Covariance Matrix, DF = 49

	in1	in2	in3	in4	in5
in1	1.019198331	0.128086799	0.291646382	0.327014916	0.417546732
in2	0.128086799	1.056460818	0.143581799	0.095937707	0.104117743
in3	0.291646382	0.143581799	1.384051249	0.058853960	0.326107730
in4	0.327014916	0.095937707	0.058853960	1.023128678	0.347916864
in5	0.417546732	0.104117743	0.326107730	0.347916864	1.606858140
out1	0.097650713	0.056612934	0.093498839	0.022915645	0.360270318
out2	0.206698403	-0.121700731	0.078294087	0.125961491	0.297046593
out3	0.516271121	0.266581451	0.481576554	0.179627237	0.749212945
out4	0.118726106	0.092288067	0.057816322	0.075028230	0.220196337
out5	0.261770905	-0.020971411	0.259053423	0.078147576	0.349618466

Covariance Matrix, DF = 49

	out1	out2	out3	out4	out5
in1	0.097650713	0.206698403	0.516271121	0.118726106	0.261770905
in2	0.056612934	-0.121700731	0.266581451	0.092288067	-0.020971411
in3	0.093498839	0.078294087	0.481576554	0.057816322	0.259053423
in4	0.022915645	0.125961491	0.179627237	0.075028230	0.078147576
in5	0.360270318	0.297046593	0.749212945	0.220196337	0.349618466
out1	0.807007554	0.217285879	0.064816340	-0.053931448	0.037758721
out2	0.217285879	0.929455806	0.206825664	0.138551008	0.054039499
out3	0.064816340	0.206825664	1.837505268	0.292963975	0.165910481
out4	-0.053931448	0.138551008	0.292963975	0.832831377	-0.067396486
out5	0.037758721	0.054039499	0.165910481	-0.067396486	0.697717191

Simple Statistics

Variable	N	Mean	Std Dev	Sum	Minimum	Maximum
in1	50	10.18988	1.00955	509.49400	7.63500	12.58860
in2	50	10.10673	1.02784	505.33640	8.12580	13.78310
in3	50	10.14888	1.17646	507.44420	7.31770	12.40080
in4	50	10.03884	1.01150	501.94200	7.40490	11.99060
in5	50	10.22587	1.26762	511.29340	7.23350	12.93360
out1	50	9.85347	0.89834	492.67340	8.01220	12.24660
out2	50	9.96857	0.96408	498.42840	7.76420	12.09450
out3	50	10.29588	1.35555	514.79410	7.29660	13.74200
out4	50	10.15856	0.91260	507.92780	8.43090	12.45230
out5	50	10.26023	0.83529	513.01130	7.86060	11.96000

After the mean and correlation structure are determined, any subset of these variables can be simulated. Suppose you are interested in a particular function of the output variables for two sets of values of the input variables for the process. In particular, you are interested in the mean and variability of the following function over 500 runs of the process conditioned on each set of input values:

$$f(out_1, \cdots, out_5) = \frac{out_1 - out_3}{out_1 + out_2 + out_3 + out_4 + out_5}$$

Although the distribution of these quantities could be determined theoretically, it is simpler to perform a conditional simulation by using PROC SIMNORMAL.

To do this, you first append a _TYPE_='COND' observation to the covariance data set produced by PROC CORR for each group of input values:

```
data cond1 ;
      _TYPE_='COND' ;
      in1 = 8    ;
      in2 = 10.5 ;
      in3 = 12   ;
      in4 = 13.5 ;
      in5 = 14.4 ;
      output ;
run ;

data cond2 ;
      _TYPE_='COND' ;
      in1 = 15.4 ;
      in2 = 13.7 ;
      in3 = 11 ;
      in4 = 7.9 ;
      in5 = 5.5 ;
      output ;
run ;
```

Next, each of these conditioning observations is appended to a copy of the OUTP=OUTCOV data from the CORR procedure, as in the following statements. A new variable, INPUT, is added to distinguish the sets of input values. This variable is used as a BY variable in subseqent steps.

```
data outcov1 ;
   input=1 ;
   set outcov cond1 ;
run ;

data outcov2 ;
   input=2 ;
   set outcov cond2 ;
run ;
```

Finally, these two data sets are concatenated:

```
data outcov ;
  set outcov1 outcov2 ;
run ;
proc print data=outcov ;
where (_type_ ne 'COV') ;
run ;
```

Figure 80.2 shows the added observations.

Figure 80.2 OUTP= Data Set from PROC CORR with _TYPE_=COND Observations Appended

Obs	input	_TYPE_	_NAME_	in1	in2	in3	in4
1	1	COV	in1	1.0192	0.1281	0.2916	0.3270
2	1	COV	in2	0.1281	1.0565	0.1436	0.0959
3	1	COV	in3	0.2916	0.1436	1.3841	0.0589
4	1	COV	in4	0.3270	0.0959	0.0589	1.0231
5	1	COV	in5	0.4175	0.1041	0.3261	0.3479
6	1	COV	out1	0.0977	0.0566	0.0935	0.0229
7	1	COV	out2	0.2067	-0.1217	0.0783	0.1260
8	1	COV	out3	0.5163	0.2666	0.4816	0.1796
9	1	COV	out4	0.1187	0.0923	0.0578	0.0750
10	1	COV	out5	0.2618	-0.0210	0.2591	0.0781
11	1	MEAN		10.1899	10.1067	10.1489	10.0388
12	1	STD		1.0096	1.0278	1.1765	1.0115
13	1	N		50.0000	50.0000	50.0000	50.0000
14	1	COND		8.0000	10.5000	12.0000	13.5000
15	2	COV	in1	1.0192	0.1281	0.2916	0.3270
16	2	COV	in2	0.1281	1.0565	0.1436	0.0959
17	2	COV	in3	0.2916	0.1436	1.3841	0.0589
18	2	COV	in4	0.3270	0.0959	0.0589	1.0231
19	2	COV	in5	0.4175	0.1041	0.3261	0.3479
20	2	COV	out1	0.0977	0.0566	0.0935	0.0229
21	2	COV	out2	0.2067	-0.1217	0.0783	0.1260
22	2	COV	out3	0.5163	0.2666	0.4816	0.1796
23	2	COV	out4	0.1187	0.0923	0.0578	0.0750
24	2	COV	out5	0.2618	-0.0210	0.2591	0.0781
25	2	MEAN		10.1899	10.1067	10.1489	10.0388
26	2	STD		1.0096	1.0278	1.1765	1.0115
27	2	N		50.0000	50.0000	50.0000	50.0000
28	2	COND		15.4000	13.7000	11.0000	7.9000

Obs	in5	out1	out2	out3	out4	out5
1	0.4175	0.0977	0.2067	0.5163	0.1187	0.2618
2	0.1041	0.0566	-0.1217	0.2666	0.0923	-0.0210
3	0.3261	0.0935	0.0783	0.4816	0.0578	0.2591
4	0.3479	0.0229	0.1260	0.1796	0.0750	0.0781
5	1.6069	0.3603	0.2970	0.7492	0.2202	0.3496
6	0.3603	0.8070	0.2173	0.0648	-0.0539	0.0378
7	0.2970	0.2173	0.9295	0.2068	0.1386	0.0540
8	0.7492	0.0648	0.2068	1.8375	0.2930	0.1659
9	0.2202	-0.0539	0.1386	0.2930	0.8328	-0.0674
10	0.3496	0.0378	0.0540	0.1659	-0.0674	0.6977
11	10.2259	9.8535	9.9686	10.2959	10.1586	10.2602
12	1.2676	0.8983	0.9641	1.3555	0.9126	0.8353
13	50.0000	50.0000	50.0000	50.0000	50.0000	50.0000
14	14.4000	.	.	.	.	.
15	0.4175	0.0977	0.2067	0.5163	0.1187	0.2618
16	0.1041	0.0566	-0.1217	0.2666	0.0923	-0.0210
17	0.3261	0.0935	0.0783	0.4816	0.0578	0.2591
18	0.3479	0.0229	0.1260	0.1796	0.0750	0.0781
19	1.6069	0.3603	0.2970	0.7492	0.2202	0.3496
20	0.3603	0.8070	0.2173	0.0648	-0.0539	0.0378
21	0.2970	0.2173	0.9295	0.2068	0.1386	0.0540
22	0.7492	0.0648	0.2068	1.8375	0.2930	0.1659
23	0.2202	-0.0539	0.1386	0.2930	0.8328	-0.0674
24	0.3496	0.0378	0.0540	0.1659	-0.0674	0.6977
25	10.2259	9.8535	9.9686	10.2959	10.1586	10.2602
26	1.2676	0.8983	0.9641	1.3555	0.9126	0.8353
27	50.0000	50.0000	50.0000	50.0000	50.0000	50.0000
28	5.5000	.	.	.	.	.

You now run PROC SIMNORMAL, specifying the input data set and the VAR and COND variables. Note that you must specify a TYPE=COV or TYPE=CORR for the input data set. PROC CORR automatically assigns a TYPE=COV or TYPE=CORR attribute for the OUTP= data set. However, since the intermediate DATA steps that appended the _TYPE_='COND' observations turned off this attribute, an explicit TYPE=CORR in the DATA= option in the PROC SIMNORMAL statement is needed.

The specification of PROC SIMNORMAL now follows from the problem description. The condition variables are IN1–IN5, the analysis variables are OUT1–OUT5, and 500 realizations are required. A seed value can be chosen arbitrarily, or the system clock can be used. Note that in the following statements, the simulation is done for each of the values of the BY variable INPUT:

```
proc simnormal data=outcov(type=cov)
      out = osim
      numreal = 500
      seed = 33179
      ;
   by input ;
   var out1-out5 ;
   cond in1-in5 ;
   run;

data b;
   set osim ;
   denom = sum(of out1-out5) ;
   if abs(denom) < 1e-8 then ff = . ;
   else ff = (out1-out3)/denom ;
run ;
```

The DATA step that follows the simulation computes the function $f(out_1, \cdots, out_5)$; in the following statements the UNIVARIATE procedure computes the simple statistics for this function for each set of conditioning input values. This is shown in Figure 80.3, and Figure 80.4 shows the distribution of the function values for each set of input values by using the SGPANEL procedure.

```
proc univariate data=b ;
   by input ;
   var ff ;
run ;
title ;
proc sgpanel data=b ;
   panelby input ;
   REFLINE 0 / axis= x ;
   density ff ;
run ;
```

Figure 80.3 Simple Statistics for ff for Each Set of Input Values

```
---------------------------------- input=1 ------------------------------------

                          The UNIVARIATE Procedure
                               Variable:  ff

                                 Moments

        N                          500    Sum Weights               500
        Mean                -0.0134833    Sum Observations    -6.7416303
        Std Deviation       0.02830426    Variance            0.00080113
        Skewness            0.56773239    Kurtosis            1.31522925
        Uncorrected SS      0.49066351    Corrected SS        0.39976435
        Coeff Variation     -209.92145    Std Error Mean       0.0012658

---------------------------------- input=1 ------------------------------------

                          Basic Statistical Measures

              Location                     Variability

            Mean     -0.01348     Std Deviation            0.02830
            Median   -0.01565     Variance                0.0008011
            Mode        .         Range                     0.21127
                                  Interquartile Range       0.03618

---------------------------------- input=1 ------------------------------------

                          Tests for Location: Mu0=0

               Test            -Statistic-    -----p Value------

               Student's t     t  -10.6519    Pr > |t|    <.0001
               Sign            M      -106    Pr >= |M|   <.0001
               Signed Rank     S    -33682    Pr >= |S|   <.0001

---------------------------------- input=1 ------------------------------------

                          Quantiles (Definition 5)

                          Quantile         Estimate

                          100% Max       0.11268600
                          99%            0.07245656
                          95%            0.03270269
                          90%            0.02064338
                          75% Q3         0.00370322
                          50% Median    -0.01564850
                          25% Q1        -0.03247389
                          10%           -0.04716239
                          5%            -0.05572806
                          1%            -0.07201126
                          0% Min        -0.09858350
```

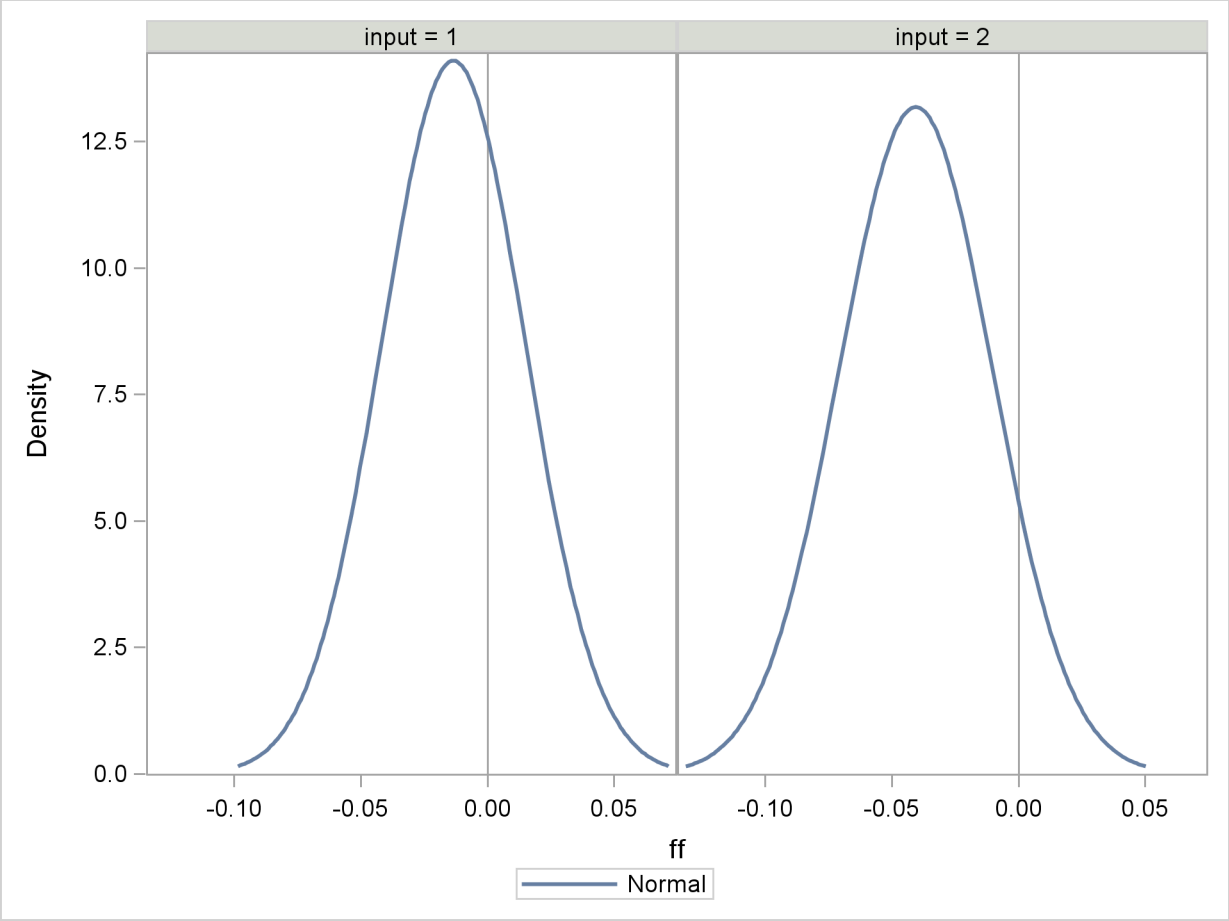
Figure 80.3 *continued*

```
---------------------------------- input=1 -----------------------------------

                              Extreme Observations

                 -------Lowest------        ------Highest------

                      Value      Obs             Value      Obs

                 -0.0985835      471         0.0750538       22
                 -0.0908179      472         0.0794747      245
                 -0.0802423       90         0.0840160       48
                 -0.0760645      249         0.1004812      222
                 -0.0756070      226         0.1126860       50

---------------------------------- input=2 -----------------------------------

                            The UNIVARIATE Procedure
                                 Variable:  ff

                                    Moments

        N                          500    Sum Weights                500
        Mean                -0.0405913    Sum Observations    -20.295631
        Std Deviation       0.03027008    Variance            0.00091628
        Skewness             0.1033062    Kurtosis            -0.1458848
        Uncorrected SS      1.28104777    Corrected SS         0.4572225
        Coeff Variation      -74.57289    Std Error Mean      0.00135372

---------------------------------- input=2 -----------------------------------

                           Basic Statistical Measures

                 Location                    Variability

             Mean     -0.04059     Std Deviation            0.03027
             Median   -0.04169     Variance               0.0009163
             Mode        .         Range                    0.18332
                                   Interquartile Range      0.04339

---------------------------------- input=2 -----------------------------------

                           Tests for Location: Mu0=0

                Test           -Statistic-    -----p Value------

                Student's t    t   -29.985    Pr > |t|    <.0001
                Sign           M      -203    Pr >= |M|   <.0001
                Signed Rank    S    -58745    Pr >= |S|   <.0001
```

Figure 80.3 *continued*

```
----------------------------------- input=2 -------------------------------------

                            Quantiles (Definition 5)

                            Quantile         Estimate

                            100% Max       0.06101208
                            99%            0.02693796
                            95%            0.01008202
                            90%           -0.00111776
                            75% Q3        -0.01847726
                            50% Median    -0.04169199
                            25% Q1        -0.06187039
                            10%           -0.07798499
                            5%            -0.08606522
                            1%            -0.11026564
                            0% Min        -0.12231183

----------------------------------- input=2 -------------------------------------

                               Extreme Observations

                    ------Lowest------        ------Highest------

                        Value      Obs             Value      Obs

                    -0.122312      937          0.0272906      688
                    -0.119884      980          0.0291769      652
                    -0.113512      920          0.0388217      670
                    -0.112345      523          0.0477261      845
                    -0.110497      897          0.0610121      632
```

Figure 80.4 Frequency Plot for ff for Each Set of Input Values



Syntax: SIMNORMAL Procedure

PROC SIMNORMAL *DATA=SAS-data-set* ;
<options>
VAR *variables* ;
BY *variables* ;
CONDITION *variables* ;

Both the PROC SIMNORMAL and VAR statements are required. The following statements can be used with the SIMNORMAL procedure:

PROC SIMNORMAL Statement

Table 80.1 summarizes the options in the PROC SIMNORMAL statement.

Table 80.1 Summary of PROC SIMNORMAL Statement Options

Option	Description
Specify Input and Output Data Sets	
DATA=	specifies input data set (TYPE=CORR, COV, and so on)
OUT=	creates output data set that contains simulated values
Seed Values	
SEED=	specifies seed value (integer)
SEEDBY	requests reinitialization of seed for each BY group
Control Contents of OUT= Data Set	
OUTSEED	requests seed values written to OUT= data set
OUTCOND	requests conditioning variable values written to OUT=data set
Control Number of Simulated Values	
NUMREAL=	specifies the number of realizations for each BY group written to the OUT= data set
Singularity Criteria	
SINGULAR1=	sets the singularity criterion for Cholesky decomposition
SINGULAR2=	sets the singularity criterion for covariance matrix sweeping

The following options can be used with the PROC SIMNORMAL statement.

DATA=*SAS-data-set*

specifies the input data set that must be a specially structured TYPE=CORR, COV, UCORR, UCOV, or SSCP SAS data set. If the DATA= option is omitted, the most recently created SAS data set is used.

SEED=*seed-value*

specifies the seed to use for the random number generator. If the SEED= value is omitted, the system clock is used. If the system clock is used, a note is written to the log; the note gives the seed value based on the system clock. In addition, the random seed stream is copied to the OUT= data set if the OUTSEED option is specified.

SEEDBY

specifies that the seed stream be reinitialized for each BY group. By default, a single random stream is used over all BY groups. If you specify SEEDBY, the random stream starts again at the initial seed value. This initial value is from the SEED= value that you specify. If you do not specify a SEED=value, the system clock generates this initial seed.

For example, suppose you had a TYPE=CORR data set with BY groups, and the mean, variances, and covariance or correlation values were identical for each BY group. Then if you specified SEEDBY, the simulated values in each BY group in the OUT= data set would be identical.

OUT=*SAS-data-set*

specifies a SAS data set in which to store the simulated values for the VAR variables. If you omit the OUT=option, the output data set is created and given a default name by using the DATA*n* convention.

See the section "OUT= Output Data Set" on page 6205 for details.

NUMREAL=*n*

specifies the number of realizations to generate. A value of NUMREAL=500 generates 500 observations in the OUT=dataset, or 500 observations within each BY group if a BY statement is given.

NUMREAL can be abbreviated as NUMR or NR.

OUTSEED

requests that the seed values be included in the OUT= data set. The variable Seed is added to the OUT= data set. The first value of Seed is the SEED= value specified in the PROC SIMNORMAL statement (or obtained from the system clock); subsequent values are produced by the random number generator.

OUTCOND

requests that the values of the conditioning variables be included in the OUT= data set. These values are constant for the data set or within a BY group. Note that specifying OUTCOND can greatly increase the size of the OUT= data set. This increase depends on the number of conditioning variables.

SINGULAR1=*number*

specifies the first singularity criterion, which is applied to the Cholesky decomposition of the covariance matrix. The SINGULAR1= value must be in the range $(0, 1)$. The default value is 10^{-8}. SINGULAR1 can be abbreviated SING1.

SINGULAR2=*number*

specifies the second singularity criterion, which is applied to the sweeping of the covariance or correlation matrix to obtain the conditional covariance. The SINGULAR2=option is applicable only when a CONDITION statement is given. The SINGULAR2= value must be in the range $(0, 1)$. The default value is 10^{-8}. SINGULAR2 can be abbreviated SING2.

BY Statement

BY *variables* **;**

A BY statement can be used with the SIMNORMAL procedure to obtain separate simulations for each covariance structure defined by the BY variables. When a BY statement appears, the procedure expects the input DATA= data set to be sorted in the order of the BY variables. If a CONDITION statement is used along with a BY statement, there must be a _TYPE_='COND' observation within each BY group. Note that if a BY statement is specified, the number of realizations specified by the NUMREAL= option are produced for each BY group.

CONDITION Statement

CONDITION | COND *variables* **;**

A CONDITION statement specifies the conditioning variables. The presence of a CONDITION statement requests that a conditional simulation be performed.

The lack of a CONDITIONAL statement simply means that an unconditional simulation for the VAR variables is to be performed.

If a CONDITION statement is given, the variables listed must be numeric variables in the DATA= data set. This requires a conditioning value for each of the CONDITION variables. This value is supplied by adding a _TYPE_='COND' observation for each CONDITION variable. Such observations are added to the DATA= data set by a DATA step.

Note that a data set created by the CORR procedure is automatically given the TYPE=COV, UCOV, CORR, or UCORR attribute, so you do not have to specify the TYPE= option in the DATA= option in the PROC SIMNORMAL statement. However, when adding the conditioning values by using a DATA step with a SET statement, you must use the TYPE=COV, UCOV, CORR, or UCORR attribute in the new data set. See the section "Getting Started: SIMNORMAL Procedure" on page 6194 for an example in which the TYPE is set.

VAR Statement

VAR *variables* **;**

Use the VAR statement to specify the analysis variables. Only numeric variables can be specified. If a VAR statement is not given, all numeric variables in the DATA= data set that are not in the CONDITION or BY statement are used.

OUT= Output Data Set

The SIMNORMAL procedure produces a single output data set: the OUT=*SAS-data-set*.

The OUT= data set contains the following variables:

- all variables listed in the VAR statement
- all variables listed in the BY statement, if one is given
- Rnum, which is the realization number within the current BY group
- Seed, which is current seed value, if the OUTSEED option is specified

- all variables listed in the CONDITION statement, if a CONDITION statement is given and the OUTCOND option is specified

The number of observations is determined by the value of the NUMREAL= option. If there are no BY groups, the number of observations in the OUT= data set is equal to the value of the NUMREAL= option. If there are BY groups, there are number of observations equals the value of the NUMREAL= option for each BY group.

Computational Details: SIMNORMAL Procedure

Introduction

There are a number of approaches to simulating a set of dependent random variables. In the context of spatial random fields, these include sequential indicator methods, turning bands, and the Karhunen-Loeve expansion. See Christakos (1992, Chapter 8) and Duetsch and Journel (1992, Chapter 5) for details.

In addition, there is the LU decomposition method, a particularly simple and computationally efficient for normal or Gaussian variates. For a given covariance matrix, the $LU = LL^T$ decomposition is computed once, and the simulation proceeds by repeatedly generating a vector of independent $N(0, 1)$ random variables and multiplying by the L matrix.

One problem with this technique is that memory is required to hold the covariance matrix of all the analysis and conditioning variables in core.

Unconditional Simulation

It is a simple matter to produce an $N(0, 1)$ random number, and by stacking k such numbers in a column vector you obtain a vector with independent standard normal components $W \sim N_k(0, I)$. The meaning of the terms *independence* and *randomness* in the context of a deterministic algorithm required for the generation of these numbers is somewhat subtle; see Knuth (1973, Vol. 2, Chapter 3) for a discussion of these issues.

Rather than $W \sim N_k(0, I)$, what is required is the generation of a vector $Z \sim N_k(0, V)$—that is,

$$Z = \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_k \end{bmatrix}$$

with covariance matrix

$$V = \begin{pmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1k} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2k} \\ & \ddots & & \\ \sigma_{k1} & \sigma_{k2} & \cdots & \sigma_{kk} \end{pmatrix}$$

where

$$\sigma_{ij} = \text{Cov}(Z_i, Z_j)$$

If the covariance matrix is symmetric and positive definite, it has a Cholesky root L such that V can be factored as

$$V = LL^T$$

where L is lower triangular. See Ralston and Rabinowitz (1978, Chapter 9, Section 3-3) for details. This vector Z can be generated by the transformation $Z = LW$. Note that this is where the assumption of multivariate normality is crucial. If $W \sim N_k(0, I_k)$, then $Z = LW$ is also normal or Gaussian. The mean of Z is

$$\text{E}(Z) = L(\text{E}(W)) = 0$$

and the variance is

$$\text{Var}(Z) = \text{Var}(LW) = \text{E}(LWW^TL^T) = L\text{E}(WW^T)L^T = LL^T = V$$

Finally, let $Y_k = Z_k + \mu_k$; that is, you add a mean term to each variable Z_k. The covariance structure of the $Y_k's$ remains the same. Unconditional simulation is done by simply repeatedly generating k $N(0, 1)$ random numbers, stacking them, and performing the transformation

$$W \longmapsto Z = LW \longmapsto Y = Z + \mu$$

Conditional Simulation

For a conditional simulation, this distribution of

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_k \end{bmatrix}$$

must be conditioned on the values of the CONDITION variables. The relevant general result concerning conditional distributions of multivariate normal random variables is the following. Let $X \sim N_m(\mu, \Sigma)$, where

$$X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$$

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}$$

$$\Sigma = \begin{pmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{pmatrix}$$

and where X_1 is $k \times 1$, X_2 is $n \times 1$, Σ_{11} is $k \times k$, Σ_{22} is $n \times n$, and $\Sigma_{12} = \Sigma_{21}^T$ is $k \times n$, with $k + n = m$. The full vector X has simply been partitioned into two subvectors, X_1 and X_2, and Σ has been similarly partitioned into covariances and cross covariances.

With this notation, the distribution of X_1 conditioned on $X_2 = x_2$ is $N_k(\tilde{\mu}, \tilde{\Sigma})$, with

$$\tilde{\mu} = \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2)$$

and

$$\tilde{\Sigma} = \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$$

See Searle (1971, pp. 46–47) for details.

Using the SIMNORMAL procedure corresponds with the conditional simulation as follows. Let $Y_1, \cdots, Y_k$ be the VAR variables as before (k is the number of variables in the VAR list). Let the mean vector for Y be denoted by $\mu_1 = E(Y)$. Let the CONDITION variables be denoted by $C_1, \cdots, C_n$ (where n is the number of variables in the COND list). Let the mean vector for C be denoted by $\mu_2 = E(C)$ and the conditioning values be denoted by

$$c = \begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{bmatrix}$$

Then stacking

$$X = \begin{bmatrix} Y \\ C \end{bmatrix}$$

the variance of X is

$$V = \mathrm{Var}(X) = \Sigma = \begin{pmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{pmatrix}$$

where $V_{11} = \text{Var}(Y)$, $V_{12} = \text{Cov}(Y, C)$, and $V_{22} = \text{Var}(C)$. By using the preceeding general result, the relevant covariance matrix is

$$\tilde{V} = V_{11} - V_{12}V_{22}^{-1}V_{21}$$

and the mean is

$$\tilde{\mu} = \mu_1 + V_{12}V_{22}^{-1}(c - \mu_2)$$

By using $\tilde{V}$ and $\tilde{\mu}$, simulating $(Y|C = c) \sim N_k(\tilde{\mu}, \tilde{V})$ now proceeds as in the unconditional case.

References

Christakos, G. (1992), *Random Field Models in Earth Sciences*, New York: Academic Press.

Deutsch, C. V. and Journel, A. G. (1992), *GSLIB: Geostatistical Software Library and User's Guide*, New York: Oxford University Press.

Knuth, D. E. (1973), *The Art of Computer Programming: Seminumerical Algorithms*, Reading, MA: Addison-Wesley.

Ralston, A. and Rabinowitz, P. (1978), *A First Course in Numerical Analysis*, Second Edition, New York: McGraw-Hill.

Searle, S. R. (1971), *Linear Models*, New York: John Wiley & Sons.

Chapter 81

The STDIZE Procedure

Contents

Overview: STDIZE Procedure **6211**
Getting Started: STDIZE Procedure **6212**
Syntax: STDIZE Procedure **6220**
PROC STDIZE Statement 6221
BY Statement 6225
FREQ Statement 6226
LOCATION Statement 6227
SCALE Statement 6227
VAR Statement 6227
WEIGHT Statement 6227
Details: STDIZE Procedure **6229**
Standardization Methods 6229
Computation of the Statistics 6230
Computing Quantiles 6232
Missing Values 6233
Output Data Sets 6234
Displayed Output 6235
ODS Table Names 6235
Example: STDIZE Procedure **6236**
Example 81.1: Standardization of Variables in Cluster Analysis 6236
References **6265**

Overview: STDIZE Procedure

The STDIZE procedure standardizes one or more numeric variables in a SAS data set by subtracting a location measure and dividing by a scale measure. A variety of location and scale measures are provided, including estimates that are resistant to outliers and clustering. Some of the well-known standardization methods such as mean, median, standard deviation, range, Huber's estimate, Tukey's biweight estimate, and Andrew's wave estimate are available in the STDIZE procedure.

In addition, you can multiply each standardized value by a constant and add a constant. Thus, the final output value is

$$result = add + multiply \times \frac{original - location}{scale}$$

where

result	= final output value
add	= constant to add (ADD= option)
multiply	= constant to multiply by (MULT= option)
original	= original input value
location	= location measure
scale	= scale measure

PROC STDIZE can also find quantiles in one pass of the data, a capability that is especially useful for very large data sets. With such data sets, the UNIVARIATE procedure might have high or excessive memory or time requirements.

Getting Started: STDIZE Procedure

The following example demonstrates how you can use the STDIZE procedure to obtain location and scale measures of your data.

In the following hypothetical data set, a random sample of grade twelve students is selected from a number of coeducational schools. Each school is classified as one of two types: Urban or Rural. There are 40 observations.

The variables are id (student identification), Type (type of school attended: 'urban'=urban area and 'rural'=rural area), and total (total assessment scores in History, Geometry, and Chemistry).

The following DATA step creates the SAS data set TotalScores.

```
data TotalScores;
   title 'High School Scores Data';
   input id Type $  total;
   datalines;
 1      rural        135
 2      rural        125
 3      rural        223
 4      rural        224
 5      rural        133
 6      rural        253
 7      rural        144
 8      rural        193
 9      rural        152
10      rural        178
11      rural        120
12      rural        180
```

```
13      rural        154
14      rural        184
15      rural        187
16      rural        111
17      rural        190
18      rural        128
19      rural        110
20      rural        217
21      urban        192
22      urban        186
23      urban         64
24      urban        159
25      urban        133
26      urban        163
27      urban        130
28      urban        163
29      urban        189
30      urban        144
31      urban        154
32      urban        198
33      urban        150
34      urban        151
35      urban        152
36      urban        151
37      urban        127
38      urban        167
39      urban        170
40      urban        123
;
```

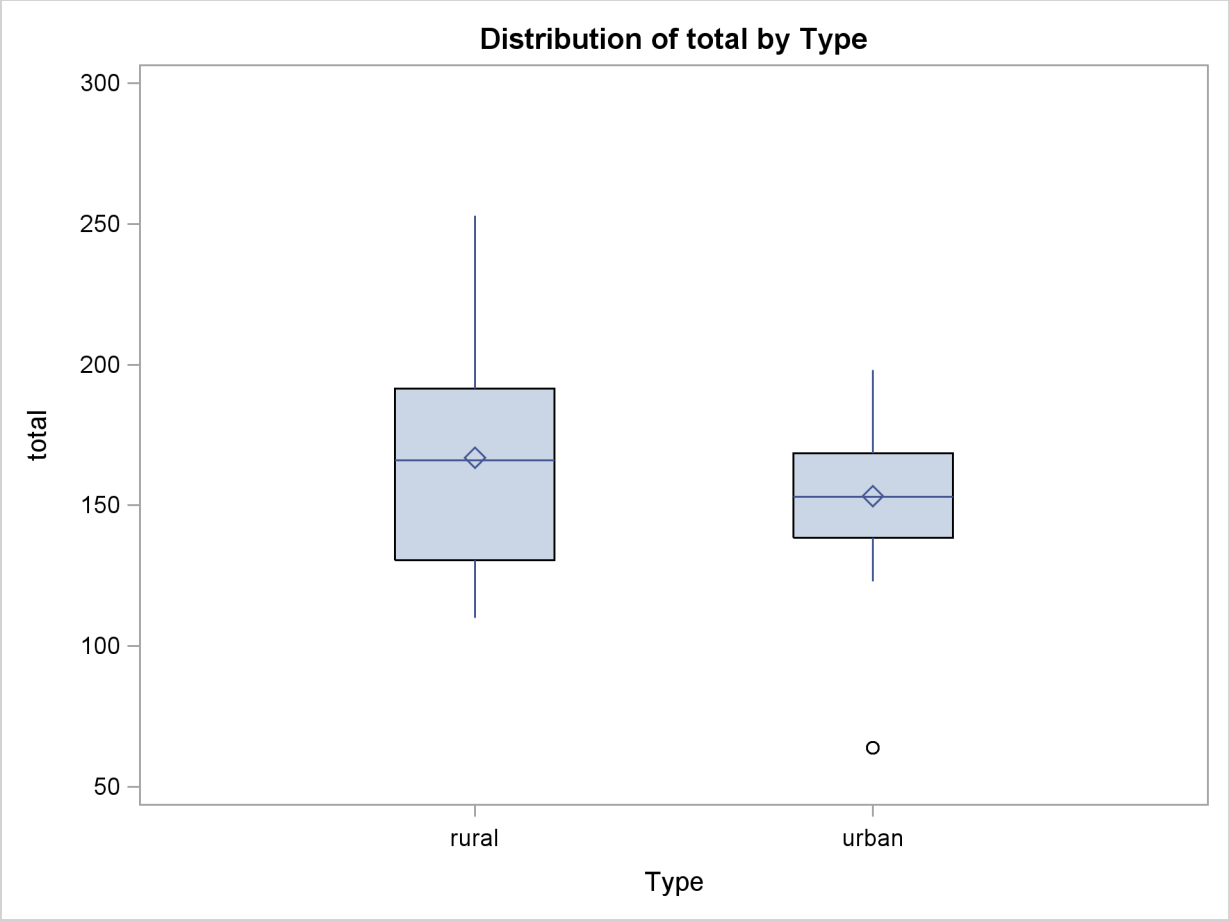
Suppose you now want to standardize the total scores in different types of schools prior to any further analysis. Before standardizing the total scores, you can use the box plot from PROC BOXPLOT to summarize the total scores for both types of schools.

To request this graph, you must specify the ODS GRAPHICS statement as follows. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

```
ods graphics on;
proc boxplot data=TotalScores;
   plot total*Type / boxstyle=schematic noserifs;
run;
ods graphics off;
```

The PLOT statement in the PROC BOXPLOT statement creates the schematic plots (without the serifs) when you specify **boxstyle=schematic noserifs**. Figure 81.1 displays a box plot for each type of school.

Figure 81.1 Schematic Plots from PROC BOXPLOT



Inspection reveals that one urban score is a low outlier. Also, if you compare the lengths of two box plots, there seems to be twice as much dispersion for the rural scores as for the urban scores.

The following PROC UNIVARIATE statement reports the information about the extreme values of the Score variable for each type of school:

```
proc univariate data=TotalScores;
   var total;
   by Type;
run;
```

Figure 81.2 displays the table from PROC UNIVARIATE for the lowest and highest five total scores for urban schools. The outlier (Obs = 3), marked in Figure 81.2 by the symbol '0', has a score of 64.

Figure 81.2 Table for Extreme Observations When Type=urban

High School Scores Data

Type=urban

The UNIVARIATE Procedure
Variable: total

Extreme Observations

----Lowest----		----Highest---	
Value	Obs	Value	Obs
64	23	170	39
123	40	186	22
127	37	189	29
130	27	192	21
133	25	198	32

The following PROC STDIZE procedure requests the METHOD=STD option for computing the location and scale measures:

```
proc stdize data=totalscores method=std pstat;
   title2 'METHOD=STD';
   var total;
   by Type;
run;
```

Figure 81.3 displays the table of location and scale measures from the PROC STDIZE statement. PROC STDIZE uses the sample mean as the location measure and the sample standard deviation as the scale measure for standardizing. The PSTAT option displays a table containing these two measures.

Figure 81.3 Location and Scale Measures Table When METHOD=STD

High School Scores Data
METHOD=STD

Type=rural

The STDIZE Procedure

Location and Scale Measures

Location = mean Scale = standard deviation

Name	Location	Scale	N
total	167.050000	41.956713	20

Figure 81.3 *continued*

```
                          High School Scores Data
                                 METHOD=STD

---------------------------------- Type=urban ----------------------------------

                             The STDIZE Procedure

              Location and Scale Measures

              Location = mean     Scale = standard deviation

              Name          Location            Scale           N

              total       153.300000        30.066768          20
```

The ratio of the scale of rural scores to the scale of urban scores is approximately 1.4 (41.96/30.07). This ratio is smaller than the dispersion ratio observed in the previous schematic plots.

The STDIZE procedure provides several location and scale measures that are resistant to outliers. The following statements invoke three different standardization methods and display the tables for the location and scale measures:

```
proc stdize data=totalscores method=mad pstat;
   title2 'METHOD=MAD';
   var total;
   by Type;
run;

proc stdize data=totalscores method=iqr pstat;
   title2 'METHOD=IQR';
   var total;
   by Type;
run;

proc stdize data=totalscores method=abw(4) pstat;
   title2 'METHOD=ABW(4)';
   var total;
   by Type;
run;
```

Figure 81.4 displays the table of location and scale measures when the standardization method is median absolute deviation (MAD). The location measure is the median, and the scale measure is the median absolute deviation from the median. The ratio of the scale of rural scores to the scale of urban scores is approximately 2.06 (32.0/15.5) and is close to the dispersion ratio observed in the previous schematic plots.

Figure 81.4 Location and Scale Measures Table When METHOD=MAD

```
                          High School Scores Data
                                 METHOD=MAD

---------------------------------- Type=rural ----------------------------------

                            The STDIZE Procedure

              Location and Scale Measures

           Location = median     Scale = median abs dev from median

              Name          Location            Scale            N

              total       166.000000        32.000000          20

                          High School Scores Data
                                 METHOD=MAD

---------------------------------- Type=urban ----------------------------------

                            The STDIZE Procedure

              Location and Scale Measures

           Location = median     Scale = median abs dev from median

              Name          Location            Scale            N

              total       153.000000        15.500000          20
```

Figure 81.5 displays the table of location and scale measures when the standardization method is IQR. The location measure is the median, and the scale measure is the interquartile range. The ratio of the scale of rural scores to the scale of urban scores is approximately 2.03 (61/30) and is, in fact, the dispersion ratio observed in the previous schematic plots.

Figure 81.5 Location and Scale Measures Table When METHOD=IQR

```
                          High School Scores Data
                                 METHOD=IQR

---------------------------------- Type=rural ----------------------------------

                            The STDIZE Procedure

              Location and Scale Measures

              Location = median     Scale = interquartile range

              Name          Location            Scale            N

              total       166.000000        61.000000          20
```

Figure 81.5 *continued*

```
                          High School Scores Data
                                METHOD=IQR

---------------------------------- Type=urban ----------------------------------

                            The STDIZE Procedure

              Location and Scale Measures

              Location = median     Scale = interquartile range

              Name          Location            Scale            N

              total       153.000000        30.000000          20
```

Figure 81.6 displays the table of location and scale measures when the standardization method is ABW, for which the location measure is the biweight 1-step M-estimate, and the scale measure is the biweight A-estimate. Note that the initial estimate for ABW is MAD. The following steps help to decide the value of the tuning constant:

1. For rural scores, the location estimate for MAD is 166.0, and the scale estimate for MAD is 32.0. The maximum of the rural scores is 253 (not shown), and the minimum is 110 (not shown). Thus, the tuning constant needs to be 3 so that it does not reject any observation that has a score between 110 to 253.

2. For urban scores, the location estimate for MAD is 153.0, and the scale estimate for MAD is 15.5. The maximum of the rural scores is 198, and the minimum (also an outlier) is 64. Thus, the tuning constant needs to be 4 so that it rejects the outlier (64) but includes the maximum (198) as an normal observation.

3. The maximum of the tuning constants, obtained in steps 1 and 2, is 4.

See Goodall (1983, Chapter 11) for details about the tuning constant. The ratio of the scale of rural scores to the scale of urban scores is approximately 2.06 (32.0/15.5). It is also close to the dispersion ratio observed in the previous schematic plots.

Figure 81.6 Location and Scale Measures Table When METHOD=ABW

```
                          High School Scores Data
                               METHOD=ABW(4)

---------------------------------- Type=rural ----------------------------------

                            The STDIZE Procedure

              Location and Scale Measures

     Location = biweight 1-step M-estimate     Scale = biweight A-estimate

              Name          Location            Scale            N

              total       162.889603        56.662855          20

                          High School Scores Data
                               METHOD=ABW(4)

---------------------------------- Type=urban ----------------------------------

                            The STDIZE Procedure

              Location and Scale Measures

     Location = biweight 1-step M-estimate     Scale = biweight A-estimate

              Name          Location            Scale            N

              total       156.014608        28.615980          20
```

The preceding analysis shows that METHOD=MAD, METHOD=IQR, and METHOD=ABW all provide better dispersion ratios than METHOD=STD does.

You can recompute the standard deviation after deleting the outlier from the original data set for comparison. The following statements create a data set NoOutlier that excludes the outlier from the TotalScores data set and invoke PROC STDIZE with METHOD=STD.

```
   data NoOutlier;
      set totalscores;
      if (total = 64) then delete;
   run;

proc stdize data=NoOutlier method=std pstat;
   title2 'after removing outlier, METHOD=STD';
   var total;
   by Type;
run;
```

Figure 81.7 displays the location and scale measures after deleting the outlier. The lack of resistance of the standard deviation to outliers is clearly illustrated: if you delete the outlier, the sample standard deviation of urban scores changes from 30.07 to 22.09. The new ratio of the scale of rural scores to the scale of urban scores is approximately 1.90 (41.96/22.09).

Figure 81.7 Location and Scale Measures Table When METHOD=STD without the Outlier

```
                          High School Scores Data
                      after removing outlier, METHOD=STD

---------------------------------- Type=rural ----------------------------------

                             The STDIZE Procedure

               Location and Scale Measures

               Location = mean     Scale = standard deviation

               Name          Location            Scale            N

               total       167.050000        41.956713          20

                          High School Scores Data
                      after removing outlier, METHOD=STD

---------------------------------- Type=urban ----------------------------------

                             The STDIZE Procedure

               Location and Scale Measures

               Location = mean     Scale = standard deviation

               Name          Location            Scale            N

               total       158.000000        22.088207          19
```

Syntax: STDIZE Procedure

The following statements are available in the STDIZE procedure:

PROC STDIZE < *options* > ;
  BY *variables* ;
  FREQ *variable* ;
  LOCATION *variables* ;
  SCALE *variables* ;
  VAR *variables* ;
  WEIGHT *variable* ;

The PROC STDIZE statement is required. The BY, LOCATION, FREQ, VAR, SCALE, and WEIGHT statements are described in alphabetical order following the PROC STDIZE statement.

PROC STDIZE Statement

PROC STDIZE < *options* > **;**

The PROC STDIZE statement invokes the procedure. You can specify the following options in the PROC STDIZE statement. Table 81.1 summarizes the options.

Table 81.1 Summary of PROC STDIZE Statement Options

Option	Description
Specify standardization methods	
METHOD=	specifies the name of the standardization method
INITIAL=	specifies the method for computing initial estimates for the A estimates
Unstandardize variables	
UNSTD	unstandardizes variables when you also specify the METHOD=IN option
Process missing values	
NOMISS	omits observations with any missing values from computation
MISSING=	specifies the method or a numeric value for replacing missing values
REPLACE	replaces missing data with zero in the standardized data
REPONLY	replaces missing data with the location measure (does not standardize the data)
Specify data set details	
DATA=	specifies the input data set
KEEPLEN	specifies that output variables inherit the length of the analysis variable
OUT=	specifies the output data set
OUTSTAT=	specifies the output statistic data set
Specify computational settings	
VARDEF=	specifies the variances divisor
NMARKERS=	specifies the number of markers when you also specify PCTLMTD=ONEPASS
MULT=	specifies the constant to multiply each value by after standardizing
ADD=	specifies the constant to add to each value after standardizing and multiplying by the value specified in the MULT= option
FUZZ=	specifies the relative fuzz factor for writing the output

Table 81.1 *continued*

Option	Description
Specify percentiles	
PCTLDEF=	specifies the definition of percentiles when you also specify the PCTLMTD=ORD_STAT option
PCTLMTD=	specifies the method used to estimate percentiles
PCTLPTS=	writes observations containing percentiles to the data set specified in the OUTSTAT= option
Normalize scale estimators	
NORM	normalizes the scale estimator to be consistent for the standard deviation of a normal distribution
SNORM	normalizes the scale estimator to have an expectation of approximately 1 for a standard normal distribution
Specify output	
PSTAT	displays the location and scale measures

These options and their abbreviations are described (in alphabetical order) in the remainder of this section.

ADD=*c*

specifies a constant, c, to add to each value after standardizing and multiplying by the value you specify in the MULT= option. The default value is 0.

DATA=*SAS-data-set*

specifies the input data set to be standardized. If you omit the DATA= option, the most recently created data set is used.

FUZZ=*c*

specifies the relative fuzz factor. The default value is 1E–14. For the OUT= data set, the score is computed as follows:

$$\text{if } |\text{Result}| < m \times c, \text{ then Result} = 0$$

where m is the constant specified in the MULT= option, or 1 if MULT= option is not specified.

For the OUTSTAT= data set and the Location and Scale table, the scale and location values are computed as follows:

$$\text{if Scale} < |\text{Location}| \times c, \text{ Scale} = 0$$

Otherwise,

$$\text{if } |\text{Location}| < m \times c, \text{ Location} = 0$$

INITIAL=*method*

specifies the method for computing initial estimates for the A estimates (ABW, AWAVE, and AHUBER). You cannot specify the following methods for initial estimates: INITIAL=ABW, INITIAL=AHUBER, INITIAL=AWAVE, and INITIAL=IN. The default is INITIAL=MAD.

KEEPLEN

specifies that output variables inherit the length of the analysis variable that PROC STDIZE uses to derive them. PROC STDIZE stores numbers in double-precision without this option.

Caution: The KEEPLEN option causes the output variables to permanently lose numeric precision by truncating or rounding the value. However, the precision of the output variables will match that of the input.

METHOD=*name*

specifies the name of the method for computing location and scale measures. Valid values for *name* are as follows: MEAN, MEDIAN, SUM, EUCLEN, USTD, STD, RANGE, MIDRANGE, MAXABS, IQR, MAD, ABW, AHUBER, AWAVE, AGK, SPACING, L, and IN.

For details about these methods, see the descriptions in the section "Standardization Methods" on page 6229. The default is METHOD=STD.

MISSING=*method* | *value*

specifies the method (or a numeric value) for replacing missing values. If you omit the MISSING= option, the REPLACE option replaces missing values with the location measure given by the METHOD= option. Specify the MISSING= option when you want to replace missing values with a different value. You can specify any name that is valid in the METHOD= option except the name IN. The corresponding location measure is used to replace missing values.

If a numeric value is given, the value replaces missing values after standardizing the data. However, you can specify the REPONLY option with the MISSING= option to suppress standardization for cases in which you want only to replace missing values.

MULT=*c*

specifies a constant, *c*, by which to multiply each value after standardizing. The default value is 1.

NMARKERS=*n*

specifies the number of markers used when you specify the one-pass algorithm (PCTLMTD=ONEPASS). The value *n* must be greater than or equal to 5. The default value is 105.

NOMISS

omits observations with missing values for any of the analyzed variables from calculation of the location and scale measures. If you omit the NOMISS option, all nonmissing values are used.

NORM

normalizes the scale estimator to be consistent for the standard deviation of a normal distribution when you specify the option METHOD=AGK, METHOD=IQR, METHOD=MAD, or METHOD=SPACING.

OUT=*SAS-data-set*

specifies the name of the SAS data set created by PROC STDIZE. The output data set is a copy of the DATA= data set except that the analyzed variables have been standardized.

Note that analyzed variables are those specified in the VAR statement or, if there is no VAR statement, all numeric variables not listed in any other statement. See the section "Output Data Sets" on page 6234 for more information.

If you want to create a permanent SAS data set, you must specify a two-level name. See *SAS Language Reference: Concepts* for more information about permanent SAS data sets.

If you omit the OUT= option, PROC STDIZE creates an output data set named according to the DATA*n* convention.

OUTSTAT=*SAS-data-set*
specifies the name of the SAS data set containing the location and scale measures and other computed statistics. See the section "Output Data Sets" on page 6234 for more information.

PCTLDEF=*percentiles*
specifies which of five definitions is used to calculate percentiles when you specify the option PCTLMTD=ORD_STAT. By default, PCTLDEF=5. Note that the option PCTLMTD=ONEPASS implies PCTLDEF=5. See the section "Computational Methods for the PCTLDEF= Option" on page 6232 for details about percentile definition.

You cannot use PCTLDEF= when you compute weighted quantiles.

PCTLMTD=ORD_STAT | ONEPASS | P2
specifies the method used to estimate percentiles. Specify the PCTLMTD=ORD_STAT option to compute the percentiles by the order statistics method.

The PCTLMTD=ONEPASS option modifies an algorithm invented by Jain and Chlamtac (1985). See the section "Computing Quantiles" on page 6232 for more details about this algorithm.

PCTLPTS=*n*
writes percentiles to the OUTSTAT= data set. Values of *n* can be any decimal number between 0 and 100, inclusive.

A requested percentile is identified by the _TYPE_ variable in the OUTSTAT= data set with a value of P*n*. For example, suppose you specify the option PCTLPTS=10, 30. The corresponding observations in the OUTSTAT= data set that contain the 10th and the 30th percentiles would then have values _TYPE_=P10 and _TYPE_=P30, respectively.

PSTAT
displays the location and scale measures.

REPLACE
replaces missing data with the value 0 in the standardized data (this value corresponds to the location measure before standardizing). To replace missing data by other values, see the preceding description of the MISSING= option. You cannot specify both the REPLACE and REPONLY options.

REPONLY
replaces missing data only; PROC STDIZE does not standardize the data. Missing values are replaced with the location measure unless you also specify the MISSING=*value* option, in

which case missing values are replaced with *value*. You cannot specify both the REPLACE and REPONLY options.

SNORM

normalizes the scale estimator to have an expectation of approximately 1 for a standard normal distribution when you specify the METHOD=SPACING option.

UNSTD

UNSTDIZE

unstandardizes variables when you specify the METHOD=IN(*ds*) option. The location and scale measures, along with constants for addition and multiplication that the unstandardization is based on, are identified by the _TYPE_ variable in the *ds* data set.

The *ds* data set must have a _TYPE_ variable and contain the following two observations: a _TYPE_= 'LOCATION' observation and a _TYPE_= 'SCALE' observation. The variable _TYPE_ can also contain the optional observations, 'ADD' and 'MULT'; if these observations are not found in the *ds* data set, the constants specified in the ADD= and MULT= options (or their default values) are used for unstandardization.

See the section "OUTSTAT= Data Set" on page 6234 for details about the statistics that each value of _TYPE_ represents. The formula used for unstandardization is as follows: If the final output value from the previous standardization is calculated as

$$result = add + multiply \times \frac{original - location}{scale}$$

The unstandardized variable is computed as

$$original = scale \times \frac{result - add}{multiply} + location$$

VARDEF=DF | N | WDF | WEIGHT | WGT

specifies the divisor to be used in the calculation of variances. By default, VARDEF=DF. The values and associated divisors are as follows.

Value	Divisor	Formula
DF	degrees of freedom	$n - 1$
N	number of observations	n
WDF	sum of weights minus 1	$(\sum_i w_i) - 1$
WEIGHT | WGT	sum of weights	$\sum_i w_i$

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC STDIZE to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC STDIZE. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

When you specify the option METHOD=IN(*ds*), the following rules are applied to BY-group processing:

- If the *ds* data set does not contain any of the BY variables, the entire DATA= data set is standardized by the location and scale measures (along with the constants for addition and multiplication) in the *ds* data set.
- If the *ds* data set contains some, but not all, of the BY variables or if some BY variables do not have the same type or length in the *ds* data set that they have in the DATA= data set, PROC STDIZE displays an error message and stops.
- If all of the BY variables appear in the *ds* data set with the same type and length as in the DATA= data set, each BY group in the DATA= data set is standardized using the location and scale measures (along with the constants for addition and multiplication) from the corresponding BY group in the *ds* data set. The BY groups in the *ds* data set must be in the same order in which they appear in the DATA= data set. All BY groups in the DATA= data set must also appear in the *ds* data set. If you do not specify the NOTSORTED option, some BY groups can appear in the *ds* data set but not in the DATA= data set; such BY groups are not used in standardizing data.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide.*

FREQ Statement

FREQ *variable* **;**

If one variable in the input data set represents the frequency of occurrence for other values in the observation, specify the variable name in a FREQ statement. PROC STDIZE treats the data set as if each observation appeared *n* times, where *n* is the value of the FREQ variable for the observation. Nonintegral values of the FREQ variable are truncated to the largest integer less than the FREQ value. If the FREQ variable has a value that is less than 1 or is missing, the observation is not used in the analysis.

NOTRUNCATE

NOTRUNC

specifies that frequency values are not truncated to integers.

The nonintegral values of the FREQ variable can be used for the following standardization methods: AGK, ABW, AHUBER, AWAVE, EUCLEN, IQR, L, MAD, MEAN, MEDIAN, SPACING, STD, SUM, and USTD. Note only when PCTLMTD=ORD_STAT is specified that the nonintegral frequency values will be used for the MAD, MEDIAN, or IQR method; if PCTLMTD=ONEPASS is specified, the NOTRUNCATE option is ignored.

LOCATION Statement

LOCATION *variables* **;**

The LOCATION statement specifies a list of numeric variables that contain location measures in the input data set specified by the METHOD=IN option.

SCALE Statement

SCALE *variables* **;**

The SCALE statement specifies the list of numeric variables containing scale measures in the input data set specified by the METHOD=IN option.

VAR Statement

VAR *variable* **;**

The VAR statement lists numeric variables to be standardized. If you omit the VAR statement, all numeric variables not listed in the BY, FREQ, and WEIGHT statements are used.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement specifies a numeric variable in the input data set with values that are used to weight each observation. Only one variable can be specified.

The WEIGHT variable values can be nonintegers. An observation is used in the analysis only if the value of the WEIGHT variable is greater than zero.

The WEIGHT variable applies only when you specify the following standardization methods: AGK, EUCLEN, IQR, L, MAD, MEAN, MEDIAN, STD, SUM, and USTD. Note that weights are used for the METHOD=MAD, MEDIAN, or IQR only when PCTLMTD=ORD_STAT is specified; if PCTLMTD=ONEPASS is specified, the WEIGHT statement is ignored.

PROC STDIZE uses the value of the WEIGHT variable to calculate the following statistics:

The sample mean and sample variances are computed as

$$\overline{x}_w = \sum_i w_i x_i / \sum_i w_i \quad \text{(sample mean)}$$

$$us_w^2 = \sum_i w_i x_i^2 / d \quad \text{(uncorrected sample variances)}$$

$$s_w^2 = \sum_i w_i (x_i - \overline{x}_w)^2 / d \quad \text{(sample variances)}$$

where w_i is the weight value of the ith observation, x_i is the value of the ith observation, and d is the divisor controlled by the VARDEF= option (see the VARDEF= option for details).

MEAN — the weighted mean, $\overline{x}_w$

SUM — the weighted sum, $\sum_i w_i x_i$

USTD — the weighted uncorrected standard deviation, $\sqrt{us_w^2}$

STD — the weighted standard deviation, $\sqrt{s_w^2}$

EUCLEN — the weighted Euclidean length, computed as the square root of the weighted uncorrected sum of squares:

$$\sqrt{\sum_i w_i x_i^2}$$

MEDIAN — the weighted median. See the section "Weighted Percentiles" on page 6233 for the formulas and descriptions.

MAD — the weighted median absolute deviation from the weighted median. See the section "Weighted Percentiles" on page 6233 for the formulas and descriptions.

IQR — the weighted median, 25th percentile, and the 75th percentile. See the section "Weighted Percentiles" on page 6233 for the formulas and descriptions.

AGK — the AGK estimate. This estimate is documented further in the ACECLUS procedure as the METHOD=COUNT option. See the discussion of the WEIGHT statement in Chapter 22, "The ACECLUS Procedure," for information about how the WEIGHT variable is applied to the AGK estimate.

L — the L_p estimate. This estimate is documented further in the FASTCLUS procedure as the LEAST= option. See the discussion of the WEIGHT statement in Chapter 34, "The FASTCLUS Procedure," for information about how the WEIGHT variable is used to compute weighted cluster means. Note that the number of clusters is always 1.

Details: STDIZE Procedure

Standardization Methods

The following table lists standardization methods and their corresponding location and scale measures available with the METHOD= option.

Table 81.2 Available Standardization Methods

Method	Location	Scale
MEAN	mean	1
MEDIAN	median	1
SUM	0	sum
EUCLEN	0	Euclidean length
USTD	0	standard deviation about origin
STD	mean	standard deviation
RANGE	minimum	range
MIDRANGE	midrange	range/2
MAXABS	0	maximum absolute value
IQR	median	interquartile range
MAD	median	median absolute deviation from median
ABW(c)	biweight 1-step M-estimate	biweight A-estimate
AHUBER(c)	Huber 1-step M-estimate	Huber A-estimate
AWAVE(c)	Wave 1-step M-estimate	Wave A-estimate
AGK(p)	mean	AGK estimate (ACECLUS)
SPACING(p)	mid-minimum spacing	minimum spacing
L(p)	L(p)	L(p)
IN(ds)	read from data set	read from data set

For METHOD=ABW(c), METHOD=AHUBER(c), or METHOD=AWAVE(c), c is a positive numeric tuning constant.

For METHOD=AGK(p), p is a numeric constant giving the proportion of pairs to be included in the estimation of the within-cluster variances.

For METHOD=SPACING(p), p is a numeric constant giving the proportion of data to be contained in the spacing.

For METHOD=L(p), p is a numeric constant greater than or equal to 1 specifying the power to which differences are to be raised in computing an L(p) or Minkowski metric.

For METHOD=IN(ds), ds is the name of a SAS data set that meets either of the following two conditions:

- contains a _TYPE_ variable. The observation that contains the location measure corresponds to the value _TYPE_= 'LOCATION', and the observation that contains the scale measure corresponds to the value _TYPE_= 'SCALE'. You can also use a data set created by the OUTSTAT= option from another PROC STDIZE statement as the *ds* data set. See the section "Output Data Sets" on page 6234 for the contents of the OUTSTAT= data set.
- contains the location and scale variables specified by the LOCATION and SCALE statements

PROC STDIZE reads in the location and scale variables in the *ds* data set by first looking for the _TYPE_ variable in the *ds* data set. If it finds this variable, PROC STDIZE continues to search for all variables specified in the VAR statement. If it does not find the _TYPE_ variable, PROC STDIZE searches for the location variables specified in the LOCATION statement and the scale variables specified in the SCALE statement.

The variable _TYPE_ can also contain the optional observations, 'ADD' and 'MULT'. If these observations are found in the *ds* data set, the values in the observation of _TYPE_ = 'MULT' are the multiplication constants, and the values in the observation of _TYPE_ = 'ADD' are the addition constants; otherwise, the constants specified in the ADD= and MULT= options (or their default values) are used.

For robust estimators, refer to Goodall (1983) and Iglewicz (1983). The MAD method has the highest breakdown point (50%), but it is somewhat inefficient. The ABW, AHUBER, and AWAVE methods provide a good compromise between breakdown and efficiency. The L(p) location estimates are increasingly robust as p drops from 2 (corresponding to least squares, or mean estimation) to 1 (corresponding to least absolute value, or median estimation). However, the L(p) scale estimates are not robust.

The SPACING method is robust to both outliers and clustering (Jannsen et al. 1995) and is, therefore, a good choice for cluster analysis or nonparametric density estimation. The mid-minimum spacing method estimates the mode for small p. The AGK method is also robust to clustering and more efficient than the SPACING method, but it is not as robust to outliers and takes longer to compute. If you expect g clusters, the argument to METHOD=SPACING or METHOD=AGK should be $\frac{1}{g}$ or less. The AGK method is less biased than the SPACING method for small samples. As a general guide, it is reasonable to use AGK for samples of size 100 or less and SPACING for samples of size 1000 or more, with the treatment of intermediate sample sizes depending on the available computer resources.

Computation of the Statistics

Formulas for statistics of METHOD=MEAN, METHOD=MEDIAN, METHOD=SUM, METHOD=USTD, METHOD=STD, METHOD=RANGE, and METHOD=IQR are given in the chapter "Elementary Statistics Procedures" (*Base SAS Procedures Guide*).

Note that the computations of median and upper and lower quartiles depend on the PCTLMTD= option.

The other statistics listed in Table 81.2, except for METHOD=IN, are described as follows:

EUCLEN — Euclidean length.
$\sqrt{\sum_{i=1}^{n} x_i^2}$, where x_i is the ith observation and n is the total number of observations in the sample.

L(p) — Minkowski metric. This metric is documented as the LEAST=p option in the PROC FASTCLUS statement of the FASTCLUS procedure (see Chapter 34, "The FASTCLUS Procedure").

If you specify METHOD=L(p) in the PROC STDIZE statement, your results are similar to those obtained from PROC FASTCLUS if you specify the LEAST=p option with MAXCLUS=1 (and use the default values of the MAXITER= option). The difference between the two types of calculations concerns the maximum number of iterations. In PROC STDIZE, it is a criterion for convergence on all variables; in PROC FASTCLUS, it is a criterion for convergence on a single variable.

The location and scale measures for L(p) are output to the OUTSEED= data set in PROC FASTCLUS.

MIDRANGE — (maximum + minimum)/2

ABW(c) — Tukey's biweight. Refer to Goodall (1983, pp. 376–378, p. 385) for the biweight 1-step M-estimate. Also refer to Iglewicz (1983, pp. 416–418) for the biweight A-estimate.

AHUBER(c) — Hubers. Refer to Goodall (1983, pp. 371–374) for the Huber 1-step M-estimate. Also refer to Iglewicz (1983, pp. 416–418) for the Huber A-estimate of scale.

AWAVE(c) — Andrews' wave. Refer to Goodall (1983, p. 376) for the Wave 1-step M-estimate. Also refer to Iglewicz (1983, pp. 416 –418) for the Wave A-estimate of scale.

AGK(p) — The noniterative univariate form of the estimator described by Art, Gnanadesikan, and Kettenring (1982).

The AGK estimate is documented in the section on the METHOD= option in the PROC ACECLUS statement of the ACECLUS procedure (also see the section "Background" on page 660 in Chapter 22, "The ACECLUS Procedure"). Specifying METHOD=AGK(p) in the PROC STDIZE statement is the same as specifying METHOD=COUNT and P=p in the PROC ACECLUS statement.

SPACING(p) — The absolute difference between two data values. The minimum spacing for a proportion p is the minimum absolute difference between two data values that contain a proportion p of the data between them. The mid-minimum spacing is the mean of these two data values.

Computing Quantiles

PROC STDIZE offers two methods for computing quantiles: the one-pass approach and the order-statistics approach (like that used in the UNIVARIATE procedure).

The one-pass approach used in PROC STDIZE modifies the P^2 algorithm for histograms proposed by Jain and Chlamtac (1985). The primary difference comes from the movement of markers. The one-pass method allows a marker to move to the right (or left) by more than one position (to the largest possible integer) as long as it does not result in two markers being in the same position. The modification is necessary in order to incorporate the FREQ variable.

You might obtain inaccurate results if you use the one-pass approach to estimate quantiles beyond the quartiles (that is, when you estimate quantiles < P25 or > P75). A large sample size (10,000 or more) is often required if the tail quantiles (quantiles <= P10 or >= P90) are requested. Note that, for variables with highly skewed or heavy-tailed distributions, tail quantile estimates might be inaccurate.

The order-statistics approach for estimating quantiles is faster than the one-pass method but requires that the entire data set be stored in memory. The accuracy in estimating the quantiles is comparable for both methods when the requested percentiles are between the lower and upper quartiles. The default is PCTLMTD=ORD_STAT if enough memory is available; otherwise, PCTLMTD=ONEPASS.

Computational Methods for the PCTLDEF= Option

You can specify one of five methods for computing quantile statistics when you use the order-statistics approach (PCTLMTD=ORD_STAT); otherwise, the PCTLDEF=5 method is used when you use the one-pass approach (PCTLMTD=ONEPASS).

Percentile Definitions Let n be the number of nonmissing values for a variable, and let $x_1, x_2, \ldots, x_n$ represent the ordered values of the variable. For the tth percentile, let $p = t/100$. In the following definitions numbered 1, 2, 3, and 5, let

$$np = j + g$$

where j is the integer part and g is the fractional part of np. For definition 4, let

$$(n+1)p = j + g$$

Given the preceding definitions, the tth percentile, y, is defined as follows:

PCTLDEF=1 weighted average at x_{np}

$$y = (1-g)x_j + gx_{j+1}$$

where x_0 is taken to be x_1

PCTLDEF=2 observation numbered closest to np

$$y = x_i$$

where i is the integer part of $np + 1/2$ if $g \neq 1/2$. If $g = 1/2$, then
$y = x_j$ if j is even, or
$y = x_{j+1}$ if j is odd

PCTLDEF=3 empirical distribution function

$$y = x_j \text{ if } g = 0$$

$$y = x_{j+1} \text{ if } g > 0$$

PCTLDEF=4 weighted average aimed at $x_{p(n+1)}$

$$y = (1 - g)x_j + gx_{j+1}$$

where x_{n+1} is taken to be x_n

PCTLDEF=5 empirical distribution function with averaging

$$y = (x_j + x_{j+1})/2 \text{ if } g = 0$$

$$y = x_{j+1} \text{ if } g > 0$$

Weighted Percentiles

When you specify a WEIGHT statement, or specify the NOTRUNCATE option in a FREQ statement, the percentiles are computed differently. The $100p$th weighted percentile y is computed from the empirical distribution function with averaging

$$y = \begin{cases} \frac{1}{2}(x_i + x_{i+1}) & \text{if } \sum_{j=1}^{i} w_j = pW \\ x_{i+1} & \text{if } \sum_{j=1}^{i} w_j < pW < \sum_{j=1}^{i+1} w_j \end{cases}$$

where w_i is the weight associated with x_i, and where $W = \sum_{i=1}^{n} w_i$ is the sum of the weights.

For PCTLMTD= ORD_STAT, the PCTLDEF= option is not applicable when a WEIGHT statement is used, or when a NOTRUNCATE option is specified in a FREQ statement. However, in this case, if all the weights are identical, the weighted percentiles are the same as the percentiles that would be computed without a WEIGHT statement and with PCTLDEF=5.

For PCTLMTD= ONEPASS, the quantile computation currently does not use any weights.

Missing Values

Missing values can be replaced by the location measure or by any specified constant (see the REPLACE option and the MISSING= option). You can also suppress standardization if you want only to replace missing values (see the REPONLY option).

If you specify the NOMISS option, PROC STDIZE omits observations with any missing values in the analyzed variables from computation of the location and scale measures.

Output Data Sets

OUT= Data Set

The output data set is a copy of the DATA= data set except that the analyzed variables have been standardized. Analyzed variables are those listed in the VAR statement or, if there is no VAR statement, all numeric variables not listed in any other statement.

OUTSTAT= Data Set

The new data set contains the following variables:

- the BY variables, if any
- _TYPE_, a character variable
- the analyzed variables

Each observation in the new data set contains a type of statistic as indicated by the _TYPE_ variable. The values of the _TYPE_ variable are as follows:

LOCATION	location measure of each variable
SCALE	scale measure of each variable
ADD	constant specified in the ADD= option. This value is the same for each variable.
MULT	constant specified in the MULT= option. This value is the same for each variable.
N	total number of nonmissing positive frequencies of each variable
NORM	norm measure of each variable. This observation is produced only when you specify the NORM option with METHOD=AGK, METHOD=IQR, METHOD=MAD, or METHOD=SPACING or when you specify the SNORM option with METHOD=SPACING.
NObsRead	number of physical records read
NObsUsed	number of physical records used in the analysis
NObsMiss	number of physical records containing missing values
P*n*	percentiles of each variable, as specified by the PCTLPTS= option. The argument n is any real number such that $0 \leq n \leq 100$
SumFreqsRead	sum of the frequency variable (or the sum of NObsUsed ones when there is no frequency variable) for all observations read
SumFreqsUsed	sum of the frequency variable (or the sum of NObsUsed ones when there is no frequency variable) for all observations used in the analysis

SumWeightsRead — sum of the weight variable (or the sum of NObsUsed ones when there is no weight variable) for all observations read

SumWeightsUsed — sum of the weight variable (or the sum of NObsUsed ones when there is no weight variable) for all observations used in the analysis

Displayed Output

If you specify the PSTAT option, PROC STDIZE displays the following statistics for each variable:

- the name of the variable, Name
- the location estimate, Location
- the scale estimate, Scale
- the norm estimate, Norm (when you specify the NORM option with METHOD=AGK, METHOD=IQR, METHOD=MAD, or METHOD=SPACING or when you specify the SNORM option with METHOD=SPACING)
- sum of nonmissing positive frequencies, N
- sum of nonmissing positive weights if the WEIGHT statement is specified, Sum of Weights

ODS Table Names

PROC STDIZE assigns a name to the single table it creates. You can use this name to reference the table when using the Output Delivery System (ODS) to select a table or create an output data set. This name is listed in Table 81.3. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 81.3 ODS Table Produced by PROC STDIZE

ODS Table Name	Description	Statement	Option
Statistics	Location and Scale Measures	PROC	PSTAT

Example: STDIZE Procedure

Example 81.1: Standardization of Variables in Cluster Analysis

To illustrate the effect of standardization in cluster analysis, this example uses the Fish data set described in the "Getting Started" section of Chapter 34, "The FASTCLUS Procedure." The numbers are measurements taken on 159 fish caught from the same lake (Laengelmavesi) near Tampere in Finland; this data set is available from the Data Archive of the *Journal of Statistics Education*. The complete data set is displayed in Chapter 82, "The STEPDISC Procedure."

The species (bream, parkki, pike, perch, roach, smelt, and whitefish), weight, three different length measurements (measured from the nose of the fish to the beginning of its tail, the notch of its tail, and the end of its tail), height, and width of each fish are recorded. The height and width are recorded as percentages of the third length variable.

Several new variables are created in the Fish data set: Weight3, Height, Width, and logLengthRatio. The weight of a fish indicates its size—a heavier pike tends to be larger than a lighter pike. To get a one-dimensional measure of the size of a fish, take the cubic root of the weight (Weight3). The variables Height, Width, Length1, Length2, and Length3 are rescaled in order to adjust for dimensionality. The logLengthRatio variable measures the tail length.

Because the new variables Weight3–logLengthRatio depend on the variable Weight, observations with missing values for Weight are not added to the data set. Consequently, there are 157 observations in the SAS data set Fish.

Before you perform a cluster analysis on coordinate data, it is necessary to consider scaling or transforming the variables since variables with large variances tend to have a larger effect on the resulting clusters than variables with small variances do.

This example uses three different approaches to standardize or transform the data prior to the cluster analysis. The first approach uses several standardization methods provided in the STDIZE procedure. However, since standardization is not always appropriate prior to the clustering (refer to Milligan and Cooper (1987) for a Monte Carlo study on various methods of variable standardization), the second approach performs the cluster analysis with no standardization. The third approach invokes the ACECLUS procedure to transform the data into a within-cluster covariance matrix.

The clustering is performed by the FASTCLUS procedure to find seven clusters. Note that the variables Length2 and Length3 are eliminated from this analysis since they both are significantly and highly correlated with the variable Length1. The correlation coefficients are 0.9958 and 0.9604, respectively. An output data set is created, and the FREQ procedure is invoked to compare the clusters with the species classification.

The DATA step is as follows, after the initial PROC FORMAT step:

```
proc format;
   value specfmt
      1='Bream'
      2='Roach'
      3='Whitefish'
      4='Parkki'
      5='Perch'
      6='Pike'
      7='Smelt';
data Fish (drop=HtPct WidthPct);
   title 'Fish Measurement Data';
   input Species Weight Length1 Length2 Length3 HtPct
         WidthPct @@;

   if Weight <= 0 or Weight=. then delete;
   Weight3=Weight**(1/3);
   Height=HtPct*Length3/(Weight3*100);
   Width=WidthPct*Length3/(Weight3*100);
   Length1=Length1/Weight3;
   Length2=Length2/Weight3;
   Length3=Length3/Weight3;
   logLengthRatio=log(Length3/Length1);

   format Species specfmt.;
   symbol = put(Species, specfmt2.);
datalines;
1  242.0 23.2 25.4 30.0 38.4 13.4 1  290.0 24.0 26.3 31.2 40.0 13.8
1  340.0 23.9 26.5 31.1 39.8 15.1 1  363.0 26.3 29.0 33.5 38.0 13.3
1  430.0 26.5 29.0 34.0 36.6 15.1 1  450.0 26.8 29.7 34.7 39.2 14.2
1  500.0 26.8 29.7 34.5 41.1 15.3 1  390.0 27.6 30.0 35.0 36.2 13.4
1  450.0 27.6 30.0 35.1 39.9 13.8 1  500.0 28.5 30.7 36.2 39.3 13.7
1  475.0 28.4 31.0 36.2 39.4 14.1 1  500.0 28.7 31.0 36.2 39.7 13.3
1  500.0 29.1 31.5 36.4 37.8 12.0 1     .  29.5 32.0 37.3 37.3 13.6
1  600.0 29.4 32.0 37.2 40.2 13.9 1  600.0 29.4 32.0 37.2 41.5 15.0
1  700.0 30.4 33.0 38.3 38.8 13.8 1  700.0 30.4 33.0 38.5 38.8 13.5
1  610.0 30.9 33.5 38.6 40.5 13.3 1  650.0 31.0 33.5 38.7 37.4 14.8
1  575.0 31.3 34.0 39.5 38.3 14.1 1  685.0 31.4 34.0 39.2 40.8 13.7
1  620.0 31.5 34.5 39.7 39.1 13.3 1  680.0 31.8 35.0 40.6 38.1 15.1
1  700.0 31.9 35.0 40.5 40.1 13.8 1  725.0 31.8 35.0 40.9 40.0 14.8

... more lines ...

7   19.7 13.2 14.3 15.2 18.9 13.6 7   19.9 13.8 15.0 16.2 18.1 11.6
;
```

The following macro, Std, standardizes the Fish data. The macro reads a single argument, mtd, which selects the METHOD= specification to be used in PROC STDIZE.

```
/*--- macro for standardization ---*/

%macro Std(mtd);
title2 "Data are standardized by PROC STDIZE with METHOD= &mtd";
   proc stdize data=fish out=sdzout method=&mtd;
      var Length1 logLengthRatio Height Width Weight3;
   run;
%mend Std;
```

The following macro, FastFreq, includes a PROC FASTCLUS statement for performing cluster analysis and a PROC FREQ statement for crosstabulating species with the cluster membership information that is derived from the previous PROC FASTCLUS statement. The macro reads a single argument, ds, which selects the input data set to be used in PROC FASTCLUS.

```
/*--- macro for clustering and crosstabulating ---*/
/*--- cluster membership with species           ---*/
%macro FastFreq(ds);
   proc fastclus data=&ds out=clust maxclusters=7 maxiter=100 noprint;
      var Length1 logLengthRatio Height Width Weight3;
   run;

   proc freq data=clust;
      tables species*cluster;
   run;
%mend FastFreq;
```

The following analysis (labeled 'Approach 1') includes 18 different methods of standardization followed by clustering. Since there is a large amount of output from this approach, only results from METHOD=STD, METHOD=RANGE, METHOD=AGK(0.14), and METHOD=SPACING(0.14) are shown. The following statements produce Output 81.1.1 through Output 81.1.4.

```
/*********************************************************/
/*                                                       */
/*     Approach 1: data are standardized by PROC STDIZE  */
/*                                                       */
/*********************************************************/

%Std(MEAN);
%FastFreq(sdzout);

%Std(MEDIAN);
%FastFreq(sdzout);

%Std(SUM);
%FastFreq(sdzout);

%Std(EUCLEN);
%FastFreq(sdzout);

%Std(USTD);
%FastFreq(sdzout);
```

```
%Std(STD);
%FastFreq(sdzout);

%Std(RANGE);
%FastFreq(sdzout);

%Std(MIDRANGE);
%FastFreq(sdzout);

%Std(MAXABS);
%FastFreq(sdzout);

%Std(IQR);
%FastFreq(sdzout);

%Std(MAD);
%FastFreq(sdzout);

%Std(AGK(.14));
%FastFreq(sdzout);

%Std(SPACING(.14));
%FastFreq(sdzout);

%Std(ABW(5));
%FastFreq(sdzout);

%Std(AWAVE(5));
%FastFreq(sdzout);

%Std(L(1));
%FastFreq(sdzout);

%Std(L(1.5));
%FastFreq(sdzout);

%Std(L(2));
%FastFreq(sdzout);
```

Output 81.1.1 Data Are Standardized by PROC STDIZE with METHOD=STD

Fish Measurement Data
Data are standardized by PROC STDIZE with METHOD= STD

The FREQ Procedure

Table of Species by CLUSTER

Species CLUSTER(Cluster)

Frequency / Percent / Row Pct / Col Pct	1	2	3	4	Total
Bream	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	34 21.66
Roach	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	19 12.10
Whitefish	0 0.00 0.00 0.00	2 1.27 33.33 10.53	0 0.00 0.00 0.00	1 0.64 16.67 7.69	6 3.82
Parkki	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	11 7.01
Total	17 10.83	19 12.10	13 8.28	13 8.28	157 100.00

(Continued)

Output 81.1.1 *continued*

Fish Measurement Data
Data are standardized by PROC STDIZE with METHOD= STD

The FREQ Procedure

Table of Species by CLUSTER

Species CLUSTER(Cluster)

Frequency / Percent / Row Pct / Col Pct	1	2	3	4	Total
Perch	0	17	0	12	56
	0.00	10.83	0.00	7.64	35.67
	0.00	30.36	0.00	21.43	
	0.00	89.47	0.00	92.31	
Pike	17	0	0	0	17
	10.83	0.00	0.00	0.00	10.83
	100.00	0.00	0.00	0.00	
	100.00	0.00	0.00	0.00	
Smelt	0	0	13	0	14
	0.00	0.00	8.28	0.00	8.92
	0.00	0.00	92.86	0.00	
	0.00	0.00	100.00	0.00	
Total	17	19	13	13	157
	10.83	12.10	8.28	8.28	100.00

(Continued)

Output 81.1.1 *continued*

```
                Fish Measurement Data
Data are standardized by PROC STDIZE with METHOD= STD

                The FREQ Procedure

            Table of Species by CLUSTER

Species     CLUSTER(Cluster)

Frequency |
Percent   |
Row Pct   |
Col Pct   |       5|       6|       7|  Total
----------+--------+--------+--------+
Bream     |      0 |     34 |      0 |     34
          |   0.00 |  21.66 |   0.00 |  21.66
          |   0.00 | 100.00 |   0.00 |
          |   0.00 | 100.00 |   0.00 |
----------+--------+--------+--------+
Roach     |      0 |      0 |     19 |     19
          |   0.00 |   0.00 |  12.10 |  12.10
          |   0.00 |   0.00 | 100.00 |
          |   0.00 |   0.00 |  38.00 |
----------+--------+--------+--------+
Whitefish |      0 |      0 |      3 |      6
          |   0.00 |   0.00 |   1.91 |   3.82
          |   0.00 |   0.00 |  50.00 |
          |   0.00 |   0.00 |   6.00 |
----------+--------+--------+--------+
Parkki    |     11 |      0 |      0 |     11
          |   7.01 |   0.00 |   0.00 |   7.01
          | 100.00 |   0.00 |   0.00 |
          | 100.00 |   0.00 |   0.00 |
----------+--------+--------+--------+
Total          11       34       50      157
              7.01    21.66    31.85   100.00
(Continued)
```

Output 81.1.1 *continued*

Fish Measurement Data
Data are standardized by PROC STDIZE with METHOD= STD

The FREQ Procedure

Table of Species by CLUSTER

Species	CLUSTER(Cluster)			
Frequency / Percent / Row Pct / Col Pct	5	6	7	Total
Perch	0	0	27	56
	0.00	0.00	17.20	35.67
	0.00	0.00	48.21	
	0.00	0.00	54.00	
Pike	0	0	0	17
	0.00	0.00	0.00	10.83
	0.00	0.00	0.00	
	0.00	0.00	0.00	
Smelt	0	0	1	14
	0.00	0.00	0.64	8.92
	0.00	0.00	7.14	
	0.00	0.00	2.00	
Total	11	34	50	157
	7.01	21.66	31.85	100.00

Output 81.1.2 Data Are Standardized by PROC STDIZE with METHOD=RANGE

Fish Measurement Data
Data are standardized by PROC STDIZE with METHOD= RANGE

The FREQ Procedure

Table of Species by CLUSTER

Species CLUSTER(Cluster)

Frequency / Percent / Row Pct / Col Pct	1	2	3	4	Total
Bream	0	0	34	0	34
	0.00	0.00	21.66	0.00	21.66
	0.00	0.00	100.00	0.00	
	0.00	0.00	100.00	0.00	
Roach	0	0	0	19	19
	0.00	0.00	0.00	12.10	12.10
	0.00	0.00	0.00	100.00	
	0.00	0.00	0.00	61.29	
Whitefish	0	0	0	3	6
	0.00	0.00	0.00	1.91	3.82
	0.00	0.00	0.00	50.00	
	0.00	0.00	0.00	9.68	
Parkki	0	0	0	0	11
	0.00	0.00	0.00	0.00	7.01
	0.00	0.00	0.00	0.00	
	0.00	0.00	0.00	0.00	
Total	17	14	34	31	157
	10.83	8.92	21.66	19.75	100.00

(Continued)

Output 81.1.2 *continued*

Fish Measurement Data
Data are standardized by PROC STDIZE with METHOD= RANGE

The FREQ Procedure

Table of Species by CLUSTER

Species	CLUSTER(Cluster)				
Frequency / Percent / Row Pct / Col Pct	1	2	3	4	Total
Perch	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	9 5.73 16.07 29.03	56 35.67
Pike	17 10.83 100.00 100.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	17 10.83
Smelt	0 0.00 0.00 0.00	14 8.92 100.00 100.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	14 8.92
Total	17 10.83	14 8.92	34 21.66	31 19.75	157 100.00

(Continued)

Output 81.1.2 *continued*

Fish Measurement Data
Data are standardized by PROC STDIZE with METHOD= RANGE

The FREQ Procedure

Table of Species by CLUSTER

Species CLUSTER(Cluster)

Frequency / Percent / Row Pct / Col Pct	5	6	7	Total
Bream	0	0	0	34
	0.00	0.00	0.00	21.66
	0.00	0.00	0.00	
	0.00	0.00	0.00	
Roach	0	0	0	19
	0.00	0.00	0.00	12.10
	0.00	0.00	0.00	
	0.00	0.00	0.00	
Whitefish	3	0	0	6
	1.91	0.00	0.00	3.82
	50.00	0.00	0.00	
	13.04	0.00	0.00	
Parkki	0	11	0	11
	0.00	7.01	0.00	7.01
	0.00	100.00	0.00	
	0.00	100.00	0.00	
Total	23	11	27	157
	14.65	7.01	17.20	100.00

(Continued)

Output 81.1.2 *continued*

Fish Measurement Data
Data are standardized by PROC STDIZE with METHOD= RANGE

The FREQ Procedure

Table of Species by CLUSTER

Species	CLUSTER(Cluster)			
Frequency / Percent / Row Pct / Col Pct	5	6	7	Total
Perch	20	0	27	56
	12.74	0.00	17.20	35.67
	35.71	0.00	48.21	
	86.96	0.00	100.00	
Pike	0	0	0	17
	0.00	0.00	0.00	10.83
	0.00	0.00	0.00	
	0.00	0.00	0.00	
Smelt	0	0	0	14
	0.00	0.00	0.00	8.92
	0.00	0.00	0.00	
	0.00	0.00	0.00	
Total	23	11	27	157
	14.65	7.01	17.20	100.00

Output 81.1.3 Data Are Standardized by PROC STDIZE with METHOD=AGK(0.14)

Fish Measurement Data
Data are standardized by PROC STDIZE with METHOD= AGK(.14)

The FREQ Procedure

Table of Species by CLUSTER

Species CLUSTER(Cluster)

Frequency / Percent / Row Pct / Col Pct	1	2	3	4	Total
Bream	0	0	34	0	34
	0.00	0.00	21.66	0.00	21.66
	0.00	0.00	100.00	0.00	
	0.00	0.00	100.00	0.00	
Roach	0	0	0	17	19
	0.00	0.00	0.00	10.83	12.10
	0.00	0.00	0.00	89.47	
	0.00	0.00	0.00	73.91	
Whitefish	0	0	0	3	6
	0.00	0.00	0.00	1.91	3.82
	0.00	0.00	0.00	50.00	
	0.00	0.00	0.00	13.04	
Parkki	11	0	0	0	11
	7.01	0.00	0.00	0.00	7.01
	100.00	0.00	0.00	0.00	
	100.00	0.00	0.00	0.00	
Total	11	14	34	23	157
	7.01	8.92	21.66	14.65	100.00

(Continued)

Output 81.1.3 *continued*

Fish Measurement Data
Data are standardized by PROC STDIZE with METHOD= AGK(.14)

The FREQ Procedure

Table of Species by CLUSTER

Species CLUSTER(Cluster)

Frequency / Percent / Row Pct / Col Pct	1	2	3	4	Total
Perch	0	0	0	3	56
	0.00	0.00	0.00	1.91	35.67
	0.00	0.00	0.00	5.36	
	0.00	0.00	0.00	13.04	
Pike	0	0	0	0	17
	0.00	0.00	0.00	0.00	10.83
	0.00	0.00	0.00	0.00	
	0.00	0.00	0.00	0.00	
Smelt	0	14	0	0	14
	0.00	8.92	0.00	0.00	8.92
	0.00	100.00	0.00	0.00	
	0.00	100.00	0.00	0.00	
Total	11	14	34	23	157
	7.01	8.92	21.66	14.65	100.00

(Continued)

Output 81.1.3 *continued*

Fish Measurement Data
Data are standardized by PROC STDIZE with METHOD= AGK(.14)

The FREQ Procedure

Table of Species by CLUSTER

Species	CLUSTER(Cluster)			
Frequency / Percent / Row Pct / Col Pct	5	6	7	Total
Bream	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	34 21.66
Roach	0 0.00 0.00 0.00	0 0.00 0.00 0.00	2 1.27 10.53 5.71	19 12.10
Whitefish	0 0.00 0.00 0.00	3 1.91 50.00 13.04	0 0.00 0.00 0.00	6 3.82
Parkki	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	11 7.01
Total	17 10.83	23 14.65	35 22.29	157 100.00

(Continued)

Output 81.1.3 *continued*

Fish Measurement Data
Data are standardized by PROC STDIZE with METHOD= AGK(.14)

The FREQ Procedure

Table of Species by CLUSTER

Species	CLUSTER(Cluster)			
Frequency / Percent / Row Pct / Col Pct	5	6	7	Total
Perch	0 0.00 0.00 0.00	20 12.74 35.71 86.96	33 21.02 58.93 94.29	56 35.67
Pike	17 10.83 100.00 100.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	17 10.83
Smelt	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	14 8.92
Total	17 10.83	23 14.65	35 22.29	157 100.00

Output 81.1.4 Data Are Standardized by PROC STDIZE with METHOD=SPACING(0.14)

Fish Measurement Data
Data are standardized by PROC STDIZE with METHOD= SPACING(.14)

The FREQ Procedure

Table of Species by CLUSTER

Species	CLUSTER(Cluster)				
Frequency / Percent / Row Pct / Col Pct	1	2	3	4	Total
Bream	0	0	0	0	34
	0.00	0.00	0.00	0.00	21.66
	0.00	0.00	0.00	0.00	
	0.00	0.00	0.00	0.00	
Roach	0	0	0	17	19
	0.00	0.00	0.00	10.83	12.10
	0.00	0.00	0.00	89.47	
	0.00	0.00	0.00	85.00	
Whitefish	3	0	0	3	6
	1.91	0.00	0.00	1.91	3.82
	50.00	0.00	0.00	50.00	
	13.04	0.00	0.00	15.00	
Parkki	0	0	11	0	11
	0.00	0.00	7.01	0.00	7.01
	0.00	0.00	100.00	0.00	
	0.00	0.00	100.00	0.00	
Total	23	17	11	20	157
	14.65	10.83	7.01	12.74	100.00

(Continued)

Output 81.1.4 *continued*

Fish Measurement Data
Data are standardized by PROC STDIZE with METHOD= SPACING(.14)

The FREQ Procedure

Table of Species by CLUSTER

Species	CLUSTER(Cluster)				
Frequency / Percent / Row Pct / Col Pct	1	2	3	4	Total
Perch	20	0	0	0	56
	12.74	0.00	0.00	0.00	35.67
	35.71	0.00	0.00	0.00	
	86.96	0.00	0.00	0.00	
Pike	0	17	0	0	17
	0.00	10.83	0.00	0.00	10.83
	0.00	100.00	0.00	0.00	
	0.00	100.00	0.00	0.00	
Smelt	0	0	0	0	14
	0.00	0.00	0.00	0.00	8.92
	0.00	0.00	0.00	0.00	
	0.00	0.00	0.00	0.00	
Total	23	17	11	20	157
	14.65	10.83	7.01	12.74	100.00

(Continued)

Output 81.1.4 *continued*

```
                    Fish Measurement Data
Data are standardized by PROC STDIZE with METHOD= SPACING(.14)

                      The FREQ Procedure

                Table of Species by CLUSTER

    Species     CLUSTER(Cluster)

    Frequency |
    Percent   |
    Row Pct   |
    Col Pct   |       5|       6|       7|  Total
    ----------+--------+--------+--------+
    Bream     |      0 |      0 |     34 |     34
              |   0.00 |   0.00 |  21.66 |  21.66
              |   0.00 |   0.00 | 100.00 |
              |   0.00 |   0.00 | 100.00 |
    ----------+--------+--------+--------+
    Roach     |      0 |      2 |      0 |     19
              |   0.00 |   1.27 |   0.00 |  12.10
              |   0.00 |  10.53 |   0.00 |
              |   0.00 |   5.26 |   0.00 |
    ----------+--------+--------+--------+
    Whitefish |      0 |      0 |      0 |      6
              |   0.00 |   0.00 |   0.00 |   3.82
              |   0.00 |   0.00 |   0.00 |
              |   0.00 |   0.00 |   0.00 |
    ----------+--------+--------+--------+
    Parkki    |      0 |      0 |      0 |     11
              |   0.00 |   0.00 |   0.00 |   7.01
              |   0.00 |   0.00 |   0.00 |
              |   0.00 |   0.00 |   0.00 |
    ----------+--------+--------+--------+
    Total          14       38       34      157
                 8.92    24.20    21.66   100.00
    (Continued)
```

Output 81.1.4 *continued*

```
                        Fish Measurement Data
        Data are standardized by PROC STDIZE with METHOD= SPACING(.14)

                          The FREQ Procedure

                      Table of Species by CLUSTER

     Species     CLUSTER(Cluster)

     Frequency |
     Percent   |
     Row Pct   |
     Col Pct   |       5|       6|       7|  Total
     ----------+--------+--------+--------+
     Perch     |      0 |     36 |      0 |     56
               |   0.00 |  22.93 |   0.00 |  35.67
               |   0.00 |  64.29 |   0.00 |
               |   0.00 |  94.74 |   0.00 |
     ----------+--------+--------+--------+
     Pike      |      0 |      0 |      0 |     17
               |   0.00 |   0.00 |   0.00 |  10.83
               |   0.00 |   0.00 |   0.00 |
               |   0.00 |   0.00 |   0.00 |
     ----------+--------+--------+--------+
     Smelt     |     14 |      0 |      0 |     14
               |   8.92 |   0.00 |   0.00 |   8.92
               | 100.00 |   0.00 |   0.00 |
               | 100.00 |   0.00 |   0.00 |
     ----------+--------+--------+--------+
     Total           14       38       34      157
                   8.92    24.20    21.66   100.00
```

The following analysis (labeled 'Approach 2') applies the cluster analysis directly to the original data. The following statements produce Output 81.1.5.

```
/*********************************************************/
/*                                                       */
/*         Approach 2: data are untransformed            */
/*                                                       */
/*********************************************************/

title2 'Data are untransformed';
%FastFreq(fish);
```

Output 81.1.5 Untransformed Data

Fish Measurement Data
Data are untransformed

The FREQ Procedure

Table of Species by CLUSTER

Species CLUSTER(Cluster)

Frequency / Percent / Row Pct / Col Pct	1	2	3	4	Total
Bream	13	0	0	0	34
	8.28	0.00	0.00	0.00	21.66
	38.24	0.00	0.00	0.00	
	44.83	0.00	0.00	0.00	
Roach	3	4	0	0	19
	1.91	2.55	0.00	0.00	12.10
	15.79	21.05	0.00	0.00	
	10.34	25.00	0.00	0.00	
Whitefish	3	0	0	0	6
	1.91	0.00	0.00	0.00	3.82
	50.00	0.00	0.00	0.00	
	10.34	0.00	0.00	0.00	
Parkki	2	3	0	0	11
	1.27	1.91	0.00	0.00	7.01
	18.18	27.27	0.00	0.00	
	6.90	18.75	0.00	0.00	
Total	29	16	10	15	157
	18.47	10.19	6.37	9.55	100.00

(Continued)

Output 81.1.5 *continued*

Fish Measurement Data
Data are untransformed

The FREQ Procedure

Table of Species by CLUSTER

Species	CLUSTER(Cluster)				
Frequency / Percent / Row Pct / Col Pct	1	2	3	4	Total
Perch	8	9	0	1	56
	5.10	5.73	0.00	0.64	35.67
	14.29	16.07	0.00	1.79	
	27.59	56.25	0.00	6.67	
Pike	0	0	10	0	17
	0.00	0.00	6.37	0.00	10.83
	0.00	0.00	58.82	0.00	
	0.00	0.00	100.00	0.00	
Smelt	0	0	0	14	14
	0.00	0.00	0.00	8.92	8.92
	0.00	0.00	0.00	100.00	
	0.00	0.00	0.00	93.33	
Total	29	16	10	15	157
	18.47	10.19	6.37	9.55	100.00

(Continued)

Output 81.1.5 *continued*

Fish Measurement Data
Data are untransformed

The FREQ Procedure

Table of Species by CLUSTER

Species CLUSTER(Cluster)

Frequency / Percent / Row Pct / Col Pct	5	6	7	Total
Bream	0	0	21	34
	0.00	0.00	13.38	21.66
	0.00	0.00	61.76	
	0.00	0.00	47.73	
Roach	12	0	0	19
	7.64	0.00	0.00	12.10
	63.16	0.00	0.00	
	30.77	0.00	0.00	
Whitefish	0	0	3	6
	0.00	0.00	1.91	3.82
	0.00	0.00	50.00	
	0.00	0.00	6.82	
Parkki	6	0	0	11
	3.82	0.00	0.00	7.01
	54.55	0.00	0.00	
	15.38	0.00	0.00	
Total	39	4	44	157
	24.84	2.55	28.03	100.00

(Continued)

Output 81.1.5 *continued*

Fish Measurement Data
Data are untransformed

The FREQ Procedure

Table of Species by CLUSTER

Species	CLUSTER(Cluster)			
Frequency / Percent / Row Pct / Col Pct	5	6	7	Total
Perch	20	0	18	56
	12.74	0.00	11.46	35.67
	35.71	0.00	32.14	
	51.28	0.00	40.91	
Pike	1	4	2	17
	0.64	2.55	1.27	10.83
	5.88	23.53	11.76	
	2.56	100.00	4.55	
Smelt	0	0	0	14
	0.00	0.00	0.00	8.92
	0.00	0.00	0.00	
	0.00	0.00	0.00	
Total	39	4	44	157
	24.84	2.55	28.03	100.00

The following analysis (labeled 'Approach 3') transforms the original data with the ACECLUS procedure and creates a TYPE=ACE output data set that is used as an input data set for the cluster analysis. The following statements produce Output 81.1.6.

```
/*********************************************************/
/*                                                       */
/*    Approach 3: data are transformed by PROC ACECLUS   */
/*                                                       */
/*********************************************************/

title2 'Data are transformed by PROC ACECLUS';
proc aceclus data=fish out=ace p=.02 noprint;
   var Length1 logLengthRatio Height Width Weight3;
run;
%FastFreq(ace);
```

Output 81.1.6 Data Are Transformed by PROC ACECLUS

Fish Measurement Data
Data are transformed by PROC ACECLUS

The FREQ Procedure

Table of Species by CLUSTER

Species CLUSTER(Cluster)

Frequency / Percent / Row Pct / Col Pct	1	2	3	4	Total
Bream	13	0	0	0	34
	8.28	0.00	0.00	0.00	21.66
	38.24	0.00	0.00	0.00	
	44.83	0.00	0.00	0.00	
Roach	3	4	0	0	19
	1.91	2.55	0.00	0.00	12.10
	15.79	21.05	0.00	0.00	
	10.34	25.00	0.00	0.00	
Whitefish	3	0	0	0	6
	1.91	0.00	0.00	0.00	3.82
	50.00	0.00	0.00	0.00	
	10.34	0.00	0.00	0.00	
Parkki	2	3	0	0	11
	1.27	1.91	0.00	0.00	7.01
	18.18	27.27	0.00	0.00	
	6.90	18.75	0.00	0.00	
Total	29	16	10	15	157
	18.47	10.19	6.37	9.55	100.00

(Continued)

Output 81.1.6 *continued*

Fish Measurement Data
Data are transformed by PROC ACECLUS

The FREQ Procedure

Table of Species by CLUSTER

Species	CLUSTER(Cluster)				
Frequency / Percent / Row Pct / Col Pct	1	2	3	4	Total
Perch	8	9	0	1	56
	5.10	5.73	0.00	0.64	35.67
	14.29	16.07	0.00	1.79	
	27.59	56.25	0.00	6.67	
Pike	0	0	10	0	17
	0.00	0.00	6.37	0.00	10.83
	0.00	0.00	58.82	0.00	
	0.00	0.00	100.00	0.00	
Smelt	0	0	0	14	14
	0.00	0.00	0.00	8.92	8.92
	0.00	0.00	0.00	100.00	
	0.00	0.00	0.00	93.33	
Total	29	16	10	15	157
	18.47	10.19	6.37	9.55	100.00

(Continued)

Output 81.1.6 *continued*

Fish Measurement Data
Data are transformed by PROC ACECLUS

The FREQ Procedure

Table of Species by CLUSTER

Species CLUSTER(Cluster)

Frequency / Percent / Row Pct / Col Pct	5	6	7	Total
Bream	0	0	21	34
	0.00	0.00	13.38	21.66
	0.00	0.00	61.76	
	0.00	0.00	47.73	
Roach	12	0	0	19
	7.64	0.00	0.00	12.10
	63.16	0.00	0.00	
	30.77	0.00	0.00	
Whitefish	0	0	3	6
	0.00	0.00	1.91	3.82
	0.00	0.00	50.00	
	0.00	0.00	6.82	
Parkki	6	0	0	11
	3.82	0.00	0.00	7.01
	54.55	0.00	0.00	
	15.38	0.00	0.00	
Total	39	4	44	157
	24.84	2.55	28.03	100.00

(Continued)

Output 81.1.6 *continued*

Fish Measurement Data
Data are transformed by PROC ACECLUS

The FREQ Procedure

Table of Species by CLUSTER

Species	CLUSTER(Cluster)			
Frequency / Percent / Row Pct / Col Pct	5	6	7	Total
Perch	20 12.74 35.71 51.28	0 0.00 0.00 0.00	18 11.46 32.14 40.91	56 35.67
Pike	1 0.64 5.88 2.56	4 2.55 23.53 100.00	2 1.27 11.76 4.55	17 10.83
Smelt	0 0.00 0.00 0.00	0 0.00 0.00 0.00	0 0.00 0.00 0.00	14 8.92
Total	39 24.84	4 2.55	44 28.03	157 100.00

Table 81.4 displays a table summarizing each classification results. In this table, the first column represents the standardization method, the second column represents the number of clusters that the seven species are classified into, and the third column represents the total number of observations that are misclassified.

Table 81.4 Summary of Clustering Results

Method of Standardization	Number of Clusters	Misclassification
MEAN	5	71
MEDIAN	5	71
SUM	6	51
EUCLEN	6	45
USTD	6	45
STD	5	33
RANGE	7	32
MIDRANGE	7	32
MAXABS	7	26
IQR	5	28
MAD	4	35
ABW(5)	6	34
AWAVE(5)	6	29
AGK(0.14)	7	28
SPACING(0.14)	7	25
L(1)	6	41
L(1.5)	5	33
L(2)	5	33
untransformed	5	71
PROC ACECLUS	5	71

Consider the results displayed in Output 81.1.1. In that analysis, the method of standardization is STD, and the number of clusters and the number of misclassifications are computed as shown in Table 81.5.

Table 81.5 Computations of Numbers of Clusters and Misclassification When Standardization Method Is STD

Species	Cluster Number	Misclassification in Each Species
Bream	6	0
Roach	7	0
Whitefish	7	3
Parkki	5	0
Perch	7	29
Pike	1	0
Smelt	3	1

In Output 81.1.1, the bream species is classified as cluster 6 since all 34 bream are categorized into cluster 6 with no misclassification. A similar pattern is seen with the roach, parkki, pike, and smelt species.

For the whitefish species, two fish are categorized into cluster 2, one fish is categorized into cluster 4, and three fish are categorized into cluster 7. Because the majority of this species is categorized

into cluster 7, it is recorded in Table 81.5 as being classified as cluster 7 with 3 misclassifications. A similar pattern is seen with the perch species: it is classified as cluster 7 with 29 misclassifications.

In summary, when the standardization method is STD, seven species of fish are classified into only five clusters and the total number of misclassified observations is 33.

The result of this analysis demonstrates that when variables are standardized by the STDIZE procedure with methods including RANGE, MIDRANGE, MAXABS, AGK(0.14), and SPACING(0.14), the FASTCLUS procedure produces the correct number of clusters and less misclassification than it does when other standardization methods are used. The SPACING method attains the best result, probably because the variables Length1 and Height both exhibit marked groupings (bimodality) in their distributions.

References

Art, D., Gnanadesikan, R., and Kettenring, R. (1982), "Data-Based Metrics for Cluster Analysis," *Utilitas Mathematica*, 21A, 75–99.

Goodall, C. (1983), "*M*-Estimators of Location: An Outline of Theory," in Hoaglin, D. C., Mosteller, M. and Tukey, J. W., eds., *Understanding Robust and Exploratory Data Analysis*, New York: John Wiley & Sons.

Iglewicz, B. (1983), "Robust Scale Estimators and Confidence Intervals for Location," in Hoaglin, D. C., Mosteller, M. and Tukey, J. W., eds., *Understanding Robust and Exploratory Data Analysis*, New York: John Wiley & Sons.

Jannsen, P., Marron, J. S., Veraverbeke, N., and Sarle, W.S. (1995), "Scale Measures for Bandwidth Selection," *J. of Nonparametric Statistics*, 5(4), 359–380.

Jain R. and Chlamtac I. (1985), "The P^2 Algorithm for Dynamic Calculation of Quantiles and Histograms Without Storing Observations," *Communications of the ACM*, 28(10), 1076–1085.

Journal of Statistics Education, "Fish Catch Data Set," http://www.amstat.org/publications/jse/jse_data_archive.html, accessed 4 December 1997.

Milligan, G. W. and Cooper, M. C. (1987), "A Study of Variable Standardization," *College of Administrative Science Working Paper Series*, 87–63, Columbus: Ohio State University.

Chapter 82

The STEPDISC Procedure

Contents

Overview: STEPDISC Procedure **6267**
Getting Started: STEPDISC Procedure **6269**
Syntax: STEPDISC Procedure **6273**
PROC STEPDISC Statement 6274
BY Statement 6278
CLASS Statement 6278
FREQ Statement 6279
VAR Statement 6279
WEIGHT Statement 6279
Details: STEPDISC Procedure **6280**
Missing Values 6280
Input Data Sets 6280
Computational Resources 6281
Displayed Output 6282
ODS Table Names 6284
Example: STEPDISC Procedure **6285**
Example 82.1: Performing a Stepwise Discriminant Analysis 6285
References **6292**

Overview: STEPDISC Procedure

Given a classification variable and several quantitative variables, the STEPDISC procedure performs a stepwise discriminant analysis to select a subset of the quantitative variables for use in discriminating among the classes. The set of variables that make up each class is assumed to be multivariate normal with a common covariance matrix. The STEPDISC procedure can use forward selection, backward elimination, or stepwise selection (Klecka 1980). The STEPDISC procedure is a useful prelude to further analyses with the CANDISC procedure or the DISCRIM procedure.

With PROC STEPDISC, variables are chosen to enter or leave the model according to one of two criteria:

- the significance level of an F test from an analysis of covariance, where the variables already chosen act as covariates and the variable under consideration is the dependent variable

- the squared partial correlation for predicting the variable under consideration from the CLASS variable, controlling for the effects of the variables already selected for the model

Forward selection begins with no variables in the model. At each step, PROC STEPDISC enters the variable that contributes most to the discriminatory power of the model as measured by Wilks' lambda, the likelihood ratio criterion. When none of the unselected variables meet the entry criterion, the forward selection process stops.

Backward elimination begins with all variables in the model except those that are linearly dependent on previous variables in the VAR statement. At each step, the variable that contributes least to the discriminatory power of the model as measured by Wilks' lambda is removed. When all remaining variables meet the criterion to stay in the model, the backward elimination process stops.

Stepwise selection begins, like forward selection, with no variables in the model. At each step, the model is examined. If the variable in the model that contributes least to the discriminatory power of the model as measured by Wilks' lambda fails to meet the criterion to stay, then that variable is removed. Otherwise, the variable not in the model that contributes most to the discriminatory power of the model is entered. When all variables in the model meet the criterion to stay and none of the other variables meet the criterion to enter, the stepwise selection process stops. Stepwise selection is the default method of variable selection.

It is important to realize that, in the selection of variables for entry, only one variable can be entered into the model at each step. The selection process does not take into account the relationships between variables that have not yet been selected. Thus, some important variables could be excluded in the process. Also, Wilks' lambda might not be the best measure of discriminatory power for your application. However, if you use PROC STEPDISC carefully, in combination with your knowledge of the data and careful cross validation, it can be a valuable aid in selecting variables for a discrimination model.

As with any stepwise procedure, it is important to remember that when many significance tests are performed, each at a level of, for example, 5% (0.05), the overall probability of rejecting at least one true null hypothesis is much larger than 5%. If you want to prevent including any variables that do not contribute to the discriminatory power of the model in the population, you should specify a very small significance level. In most applications, all variables considered have some discriminatory power, however small. To choose the model that provides the best discrimination by using the sample estimates, you need only to guard against estimating more parameters than can be reliably estimated with the given sample size.

Costanza and Afifi (1979) use Monte Carlo studies to compare alternative stopping rules that can be used with the forward selection method in the two-group multivariate normal classification problem. Five different numbers of variables, ranging from 10 to 30, are considered in the studies. The comparison is based on conditional and estimated unconditional probabilities of correct classification. They conclude that the use of a moderate significance level, in the range of 10 to 25 percent, often performs better than the use of a much larger or a much smaller significance level.

The significance level and the squared partial correlation criteria select variables in the same order, although they might select different numbers of variables. Increasing the sample size tends to increase the number of variables selected when you are using significance levels, but it has little effect on the number selected by using squared partial correlations.

See Chapter 10, "Introduction to Discriminant Procedures," for more information about discriminant analysis.

Getting Started: STEPDISC Procedure

The data in this example are measurements of 159 fish caught in Finland's lake Laengelmavesi; this data set is available from the Puranen. For each of the seven species (bream, roach, whitefish, parkki, perch, pike, and smelt) the weight, length, height, and width of each fish are tallied. Three different length measurements are recorded: from the nose of the fish to the beginning of its tail, from the nose to the notch of its tail, and from the nose to the end of its tail. The height and width are recorded as percentages of the third length variable. PROC STEPDISC will select a subset of the six quantitative variables that might be useful for differentiating between the fish species. This subset is used in conjunction with PROC CANDISC and PROC DISCRIM to develop discrimination models.

The following steps create the data set fish and use PROC STEPDISC to select a subset of potential discriminator variables. By default, PROC STEPDISC uses stepwise selection on all numeric variables that are not listed in other statements, and the significance levels for a variable to enter the subset and to stay in the subset are set to 0.15. The following statements produce Figure 82.1 through Figure 82.5:

```
title 'Fish Measurement Data';

proc format;
   value specfmt
      1='Bream'
      2='Roach'
      3='Whitefish'
      4='Parkki'
      5='Perch'
      6='Pike'
      7='Smelt';
run;

data fish (drop=HtPct WidthPct);
   input Species Weight Length1 Length2 Length3 HtPct WidthPct @@;
   Height=HtPct*Length3/100;
   Width=WidthPct*Length3/100;
   format Species specfmt.;
   datalines;
1  242.0 23.2 25.4 30.0 38.4 13.4 1  290.0 24.0 26.3 31.2 40.0 13.8
1  340.0 23.9 26.5 31.1 39.8 15.1 1  363.0 26.3 29.0 33.5 38.0 13.3
1  430.0 26.5 29.0 34.0 36.6 15.1 1  450.0 26.8 29.7 34.7 39.2 14.2

... more lines ...

7   19.7 13.2 14.3 15.2 18.9 13.6 7   19.9 13.8 15.0 16.2 18.1 11.6
;
```

```
proc stepdisc data=fish;
   class Species;
run;
```

PROC STEPDISC begins by displaying summary information about the analysis (see Figure 82.1). This information includes the number of observations with nonmissing values, the number of classes in the classification variable (specified by the CLASS statement), the number of quantitative variables under consideration, the significance criteria for variables to enter and to stay in the model, and the method of variable selection being used. The frequency of each class is also displayed.

Figure 82.1 Summary Information

Fish Measurement Data

The STEPDISC Procedure

The Method for Selecting Variables is STEPWISE

Total Sample Size	158	Variable(s) in the Analysis	6
Class Levels	7	Variable(s) Will Be Included	0
		Significance Level to Enter	0.15
		Significance Level to Stay	0.15

Number of Observations Read	159
Number of Observations Used	158

Class Level Information

Species	Variable Name	Frequency	Weight	Proportion
Bream	Bream	34	34.0000	0.215190
Parkki	Parkki	11	11.0000	0.069620
Perch	Perch	56	56.0000	0.354430
Pike	Pike	17	17.0000	0.107595
Roach	Roach	20	20.0000	0.126582
Smelt	Smelt	14	14.0000	0.088608
Whitefish	Whitefish	6	6.0000	0.037975

For each entry step, the statistics for entry are displayed for all variables not currently selected (see Figure 82.2). The variable selected to enter at this step (if any) is displayed, as well as all the variables currently selected. Next are multivariate statistics that take into account all previously selected variables and the newly entered variable.

Figure 82.2 Step 1: Variable HEIGHT Selected for Entry

Fish Measurement Data

The STEPDISC Procedure
Stepwise Selection: Step 1

Statistics for Entry, DF = 6, 151

Variable	R-Square	F Value	Pr > F	Tolerance
Weight	0.3750	15.10	<.0001	1.0000
Length1	0.6017	38.02	<.0001	1.0000
Length2	0.6098	39.32	<.0001	1.0000
Length3	0.6280	42.49	<.0001	1.0000
Height	0.7553	77.69	<.0001	1.0000
Width	0.4806	23.29	<.0001	1.0000

Variable Height will be entered.

Variable(s) That Have Been Entered

Height

Multivariate Statistics

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.244670	77.69	6	151	<.0001
Pillai's Trace	0.755330	77.69	6	151	<.0001
Average Squared Canonical Correlation	0.125888				

For each removal step (Figure 82.3), the statistics for removal are displayed for all variables currently entered. The variable to be removed at this step (if any) is displayed. If no variable meets the criterion to be removed and the maximum number of steps as specified by the MAXSTEP= option has not been attained, then the procedure continues with another entry step.

Figure 82.3 Step 2: No Variable Is Removed; Variable Length2 Added

Fish Measurement Data

The STEPDISC Procedure
Stepwise Selection: Step 2

Statistics for Removal, DF = 6, 151

Variable	R-Square	F Value	Pr > F
Height	0.7553	77.69	<.0001

No variables can be removed.

Figure 82.3 *continued*

Statistics for Entry, DF = 6, 150

Variable	Partial R-Square	F Value	Pr > F	Tolerance
Weight	0.7388	70.71	<.0001	0.4690
Length1	0.9220	295.35	<.0001	0.6083
Length2	0.9229	299.31	<.0001	0.5892
Length3	0.9173	277.37	<.0001	0.5056
Width	0.8783	180.44	<.0001	0.3699

Variable Length2 will be entered.

Variable(s) That Have Been Entered

Length2 Height

Multivariate Statistics

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.018861	157.04	12	300	<.0001
Pillai's Trace	1.554349	87.78	12	302	<.0001
Average Squared Canonical Correlation	0.259058				

The stepwise procedure terminates either when no variable can be removed and no variable can be entered or when the maximum number of steps as specified by the MAXSTEP= option has been attained. In this example at step 7 no variables can be either removed or entered (Figure 82.4). Steps 3 through 6 are not displayed in this document.

Figure 82.4 Step 7: No Variables Entered or Removed

Fish Measurement Data

The STEPDISC Procedure
Stepwise Selection: Step 7

Statistics for Removal, DF = 6, 146

Variable	Partial R-Square	F Value	Pr > F
Weight	0.4521	20.08	<.0001
Length1	0.2987	10.36	<.0001
Length2	0.5250	26.89	<.0001
Length3	0.7948	94.25	<.0001
Height	0.7257	64.37	<.0001
Width	0.5757	33.02	<.0001

No variables can be removed.

PROC STEPDISC ends by displaying a summary of the steps.

Figure 82.5 Step Summary

No further steps are possible.

Fish Measurement Data

The STEPDISC Procedure

Stepwise Selection Summary

Step	Number In	Entered	Removed	Partial R-Square	F Value	Pr > F	Wilks' Lambda	Pr < Lambda	Average Squared Canonical Correlation	Pr > ASCC
1	1	Height		0.7553	77.69	<.0001	0.24466983	<.0001	0.12588836	<.0001
2	2	Length2		0.9229	299.31	<.0001	0.01886065	<.0001	0.25905822	<.0001
3	3	Length3		0.8826	186.77	<.0001	0.00221342	<.0001	0.38427100	<.0001
4	4	Width		0.5775	33.72	<.0001	0.00093510	<.0001	0.45200732	<.0001
5	5	Weight		0.4461	19.73	<.0001	0.00051794	<.0001	0.49488458	<.0001
6	6	Length1		0.2987	10.36	<.0001	0.00036325	<.0001	0.51744189	<.0001

All the variables in the data set are found to have potential discriminatory power. These variables are used to develop discrimination models in both the CANDISC and DISCRIM procedure chapters.

Syntax: STEPDISC Procedure

The following statements are available in PROC STEPDISC:

PROC STEPDISC < *options* > ;
CLASS *variable* ;
BY *variables* ;
FREQ *variable* ;
VAR *variables* ;
WEIGHT *variable* ;

The BY, CLASS, FREQ, VAR, and WEIGHT statements are described after the PROC STEPDISC statement.

PROC STEPDISC Statement

PROC STEPDISC < *options* > **;**

The PROC STEPDISC statement invokes the STEPDISC procedure. The options listed in Table 82.1 are available in the PROC STEPDISC statement.

Table 82.1 STEPDISC Procedure Options

Option	Description
Input Data Set	
DATA=	specifies input SAS data set
Method Details	
MAXMACRO=	specifies maximum macro variable lists
METHOD=	specifies method
SINGULAR=	specifies singularity
Control Stepwise Selection	
SLENTRY=	specifies entry significance
SLSTAY=	specifies staying significance
PR2ENTRY=	specifies entry partial R square
PR2STAY=	specifies staying partial R square
INCLUDE=	forces inclusion of variables
MAXSTEP=	specifies maximum number of steps
START=	specifies variables to begin
STOP=	specifies number of variables in final model
Control Displayed Output	
ALL	displays all
BCORR	displays between correlations
BCOV	displays between covariances
BSSCP	displays between SSCPs
PCORR	displays pooled correlations
PCOV	displays pooled covariances
PSSCP	displays pooled SSCPs
SHORT	suppresses output
SIMPLE	displays descriptive statistics
STDMEAN	displays standardized class means
TCORR	displays total correlations
TCOV	displays total covariances
TSSCP	displays total SSCPs
WCORR	displays within correlations
WCOV	displays within covariances
WSSCP	displays within SSCPs

ALL

activates all of the display options.

BCORR

displays between-class correlations.

BCOV

displays between-class covariances. The between-class covariance matrix equals the between-class SSCP matrix divided by $n(c-1)/c$, where n is the number of observations and c is the number of classes. The between-class covariances should be interpreted in comparison with the total-sample and within-class covariances, not as formal estimates of population parameters.

BSSCP

displays the between-class SSCP matrix.

DATA=*SAS-data-set*

specifies the data set to be analyzed. The data set can be an ordinary SAS data set or one of several specially structured data sets created by statistical procedures available with SAS/STAT software. These specially structured data sets include TYPE=CORR, COV, CSSCP, and SSCP. If the DATA= option is omitted, the procedure uses the most recently created SAS data set.

INCLUDE=*n*

includes the first n variables in the VAR statement in every model. By default, INCLUDE=0.

MAXMACRO=*n*

specifies the maximum number of macro variables with independent variable lists to create. By default, MAXMACRO=100. PROC STEPDISC saves the list of selected variables in a macro variable, &_StdVar. Suppose your input variable list consists of x1-x10; then &_StdVar would be set to x1 x3 x4 x10 if, for example, the first, third, fourth, and tenth variables were selected for the model. This list can be used, for example, in a subsequent procedure's VAR statement as follows:

```
var &_stdvar;
```

With BY processing, one macro variable is created for each BY group, and the macro variables are indexed by the BY-group number. The MAXMACRO= option can be used to either limit or increase the number of these macro variables in processing data sets with many BY groups. The macro variables are created as follows:

With no BY processing, PROC STEPDISC creates the following:

_StdVar	selected variables
_StdVar1	selected variables
_StdNumBys	number of BY groups (1)
_StdNumMacroBys	number of _StdVar*i* macro variables actually made (1)

With BY processing, PROC STEPDISC creates the following:

_StdVar	selected variables for BY group 1
_StdVar1	selected variables for BY group 1
_StdVar2	selected variables for BY group 2
.	
.	
.	
_StdVar*m*	selected variables for BY group m, where a number is substituted for m
_StdNumBys	n, the number of BY groups
_StdNumMacroBys	the number m of _StdVar*i* macro variables actually made. This value might be less than _StdNumbys = n, and it is less than or equal to the MAXMACRO= value.

MAXSTEP=*n*

specifies the maximum number of steps. By default, MAXSTEP= two times the number of variables in the VAR statement.

METHOD=BACKWARD | BW

METHOD=FORWARD | FW

METHOD=STEPWISE | SW

specifies the method used to select the variables in the model. The BACKWARD method specifies backward elimination, FORWARD specifies forward selection, and STEPWISE specifies stepwise selection. By default, METHOD=STEPWISE.

PCORR

displays pooled within-class correlations (partial correlations based on the pooled within-class covariances).

PCOV

displays pooled within-class covariances.

PR2ENTRY=*p*

PR2E=*p*

specifies the partial R square for adding variables in the forward selection mode, where $p \leq 1$.

PR2STAY=*p*

PR2S=*p*

specifies the partial R square for retaining variables in the backward elimination mode, where $p \leq 1$.

PSSCP

displays the pooled within-class corrected SSCP matrix.

SHORT

suppresses the displayed output from each step.

SIMPLE

displays simple descriptive statistics for the total sample and within each class.

SINGULAR=*p*

specifies the singularity criterion for entering variables, where $0 < p < 1$. PROC STEPDISC precludes the entry of a variable if the squared multiple correlation of the variable with the variables already in the model exceeds $1 - p$. With more than one variable already in the model, PROC STEPDISC also excludes a variable if it would cause any of the variables already in the model to have a squared multiple correlation (with the entering variable and the other variables in the model) exceeding $1 - p$. By default, SINGULAR= 1E–8.

SLENTRY=*p*

SLE=*p*

specifies the significance level for adding variables in the forward selection mode, where $0 \leq p \leq 1$. The default value is 0.15.

SLSTAY=*p*

SLS=*p*

specifies the significance level for retaining variables in the backward elimination mode, where $0 \leq p \leq 1$. The default value is 0.15.

START=*n*

specifies that the first n variables in the VAR statement be used to begin the selection process. When you specify METHOD=FORWARD or METHOD=STEPWISE, the default value is 0; when you specify METHOD=BACKWARD, the default value is the number of variables in the VAR statement.

STDMEAN

displays total-sample and pooled within-class standardized class means.

STOP=*n*

specifies the number of variables in the final model. The STEPDISC procedure stops the selection process when a model with n variables is found. This option applies only when you specify METHOD=FORWARD or METHOD=BACKWARD. When you specify METHOD=FORWARD, the default value is the number of variables in the VAR statement; when you specify METHOD=BACKWARD, the default value is 0.

TCORR

displays total-sample correlations.

TCOV

displays total-sample covariances.

TSSCP

displays the total-sample corrected SSCP matrix.

WCORR
displays within-class correlations for each class level.

WCOV
displays within-class covariances for each class level.

WSSCP
displays the within-class corrected SSCP matrix for each class level.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC STEPDISC to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC STEPDISC. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variable* **;**

The values of the CLASS variable define the groups for analysis. Class levels are determined by the formatted values of the CLASS variable. The CLASS variable can be numeric or character. A CLASS statement is required.

FREQ Statement

FREQ *variable* **;**

If a variable in the data set represents the frequency of occurrence for the other values in the observation, include the name of the variable in a FREQ statement. The procedure then treats the data set as if each observation appears n times, where n is the value of the FREQ variable for the observation. The total number of observations is considered to be equal to the sum of the FREQ variable when the procedure determines degrees of freedom for significance probabilities.

If the value of the FREQ variable is missing or is less than one, the observation is not used in the analysis. If the value is not an integer, the value is truncated to an integer.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the quantitative variables eligible for selection. The default is all numeric variables not listed in other statements.

WEIGHT Statement

WEIGHT *variable* **;**

To use relative weights for each observation in the input data set, place the weights in a variable in the data set and specify the name in a WEIGHT statement. This is often done when the variance associated with each observation is different and the values of the WEIGHT variable are proportional to the reciprocals of the variances. If the value of the WEIGHT variable is missing or is less than zero, then a value of zero for the weight is assumed.

The WEIGHT and FREQ statements have a similar effect except that the WEIGHT statement does not alter the degrees of freedom.

Details: STEPDISC Procedure

Missing Values

Observations containing missing values are omitted from the analysis.

Input Data Sets

The input data set can be an ordinary SAS data set or one of several specially structured data sets created by statistical procedures available with SAS/STAT software. For more information about these data sets, see Appendix A, "Special SAS Data Sets." The BY variable in these data sets becomes the CLASS variable in PROC STEPDISC. These specially structured data sets include the following:

- TYPE=CORR data sets created by PROC CORR by using a BY statement
- TYPE=COV data sets created by PROC PRINCOMP by using both the COV option and a BY statement
- TYPE=CSSCP data sets created by PROC CORR by using the CSSCP option and a BY statement, where the OUT= data set is assigned TYPE=CSSCP with the TYPE= data set option
- TYPE=SSCP data sets created by PROC REG by using both the OUTSSCP= option and a BY statement

When the input data set is TYPE=CORR, TYPE=COV, or TYPE=CSSCP, the STEPDISC procedure reads the number of observations for each class from the observations with _TYPE_='N' and the variable means in each class from the observations with _TYPE_='MEAN'. The procedure then reads the within-class correlations from the observations with _TYPE_='CORR', the standard deviations from the observations with _TYPE_='STD' (data set TYPE=CORR), the within-class covariances from the observations with _TYPE_='COV' (data set TYPE=COV), or the within-class corrected sums of squares and crossproducts from the observations with _TYPE_='CSSCP' (data set TYPE=CSSCP).

When the data set does not include any observations with _TYPE_='CORR' (data set TYPE=CORR), _TYPE_='COV' (data set TYPE=COV), or _TYPE_='CSSCP' (data set TYPE=CSSCP) for each class, PROC STEPDISC reads the pooled within-class information from the data set. In this case, the STEPDISC procedure reads the pooled within-class correlations from the observations with _TYPE_='PCORR', the pooled within-class standard deviations from the observations with _TYPE_='PSTD' (data set TYPE=CORR), the pooled within-class covariances

from the observations with _TYPE_='PCOV' (data set TYPE=COV), or the pooled within-class corrected SSCP matrix from the observations with_TYPE_='PSSCP' (data set TYPE=CSSCP).

When the input data set is TYPE=SSCP, the STEPDISC procedure reads the number of observations for each class from the observations with _TYPE_='N', the sum of weights of observations from the variable INTERCEPT in observations with _TYPE_='SSCP' and _NAME_='INTERCEPT', the variable sums from the variable=*variablenames* in observations with _TYPE_='SSCP' and _NAME_='INTERCEPT', and the uncorrected sums of squares and crossproducts from the variable=*variablenames* in observations with _TYPE_='SSCP' and _NAME_=*variablenames*.

Computational Resources

In the following discussion, let

$$
\begin{aligned}
n &= \text{number of observations} \\
c &= \text{number of class levels} \\
v &= \text{number of variables in the VAR list} \\
l &= \text{length of the CLASS variable} \\
t &= v + c - 1
\end{aligned}
$$

Memory Requirements

The amount of memory in bytes for temporary storage needed to process the data is

$$c(4v^2 + 28v + 3l + 4c + 72) + 16v^2 + 92v + 4t^2 + 20t + 4l$$

Additional temporary storage of 72 bytes at each step is also required to store the results.

Time Requirements

The following factors determine the time requirements of a stepwise discriminant analysis:

- The time needed for reading the data and computing covariance matrices is proportional to nv^2. The STEPDISC procedure must also look up each class level in the list. This is faster if the data are sorted by the CLASS variable. The time for looking up class levels is proportional to a value ranging from n to $n \ln(c)$.
- The time needed for stepwise discriminant analysis is proportional to the number of steps required to select the set of variables in the discrimination model. The number of steps required depends on the data set itself and the selection method and criterion used in the procedure. Each forward or backward step takes time proportional to $(v + c)^2$.

Displayed Output

The displayed output from PROC STEPDISC includes the class level information table. For each level of the classification variable, the following information is provided: the output data set variable name, frequency sum, weight sum, and the proportion of the total sample.

The optional output from PROC STEPDISC includes the following:

The optional output includes the following:

- Within-class SSCP matrices for each group
- Pooled within-class SSCP matrix
- Between-class SSCP matrix
- Total-sample SSCP matrix
- Within-class covariance matrices for each group
- Pooled within-class covariance matrix
- Between-class covariance matrix, equal to the between-class SSCP matrix divided by $n(c-1)/c$, where n is the number of observations and c is the number of classes
- Total-sample covariance matrix
- Within-class correlation coefficients and Pr $> |r|$ to test the hypothesis that the within-class population correlation coefficients are zero
- Pooled within-class correlation coefficients and Pr $> |r|$ to test the hypothesis that the partial population correlation coefficients are zero
- Between-class correlation coefficients and Pr $> |r|$ to test the hypothesis that the between-class population correlation coefficients are zero
- Total-sample correlation coefficients and Pr $> |r|$ to test the hypothesis that the total population correlation coefficients are zero
- Simple statistics, including N (the number of observations), sum, mean, variance, and standard deviation for the total sample and within each class
- Total-sample standardized class means, obtained by subtracting the grand mean from each class mean and dividing by the total-sample standard deviation
- Pooled within-class standardized class means, obtained by subtracting the grand mean from each class mean and dividing by the pooled within-class standard deviation

At each step, the following statistics are displayed:

- for each variable considered for entry or removal: partial R-square, the squared (partial) correlation, the F statistic, and Pr $>$ F, the probability level, from a one-way analysis of covariance
- the minimum tolerance for entering each variable. A variable is entered only if its tolerance and the tolerances for all variables already in the model are greater than the value specified in the SINGULAR= option. The tolerance for the entering variable is $1 - R^2$ from regressing the entering variable on the other variables already in the model. The tolerance for a variable already in the model is $1 - R^2$ from regressing that variable on the entering variable and the other variables already in the model. With m variables already in the model, for each entering variable, $m + 1$ multiple regressions are performed by using the entering variable and each of the m variables already in the model as a dependent variable. These $m + 1$ tolerances are computed for each entering variable, and the minimum tolerance is displayed for each.

  The tolerance is computed by using the total-sample correlation matrix. It is customary to compute tolerance by using the pooled within-class correlation matrix (Jennrich 1977), but it is possible for a variable with excellent discriminatory power to have a high total-sample tolerance and a low pooled within-class tolerance. For example, PROC STEPDISC enters a variable that yields perfect discrimination (that is, produces a canonical correlation of one), but a program that uses pooled within-class tolerance does not.
- the variable label, if any
- the name of the variable chosen
- the variables already selected or removed
- Wilks' lambda and the associated F approximation with degrees of freedom and Pr $<$ F, the associated probability level after the selected variable has been entered or removed. Wilks' lambda is the likelihood ratio statistic for testing the hypothesis that the means of the classes on the selected variables are equal in the population (see the section "Multivariate Tests" on page 102 in Chapter 4, "Introduction to Regression Procedures.") Lambda is close to zero if any two groups are well separated.
- Pillai's trace and the associated F approximation with degrees of freedom and Pr $>$ F, the associated probability level after the selected variable has been entered or removed. Pillai's trace is a multivariate statistic for testing the hypothesis that the means of the classes on the selected variables are equal in the population (see the section "Multivariate Tests" on page 102 in Chapter 4, "Introduction to Regression Procedures").
- Average squared canonical correlation (ASCC). The ASCC is Pillai's trace divided by the number of groups minus 1. The ASCC is close to 1 if all groups are well separated and if all or most directions in the discriminant space show good separation for at least two groups.
- Summary to give statistics associated with the variable chosen at each step. The summary includes the following:
  - Step number
  - Variable entered or removed

- Number in, the number of variables in the model
- Partial R-square
- the F value for entering or removing the variable
- Pr $>$ F, the probability level for the F statistic
- Wilks' lambda
- Pr $<$ Lambda based on the F approximation to Wilks' lambda
- Average squared canonical correlation
- Pr $>$ ASCC based on the F approximation to Pillai's trace
- the variable label, if any

ODS Table Names

PROC STEPDISC assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 82.2 along with the PROC STEPDISC statement options needed to produce the table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 82.2 ODS Tables Produced by PROC STEPDISC

ODS Table Name	Description	Option
BCorr	Between-class correlations	BCORR
BCov	Between-class covariances	BCOV
BSSCP	Between-class SSCP matrix	BSSCP
Counts	Number of observations, variables, classes, df	default
CovDF	Nonprinting table of df for covariance matrices	any *COV option
Levels	Class level information	default
Messages	Entry/removal messages	default
Multivariate	Multivariate statistics	default
NObs	Number of observations	default
PCorr	Pooled within-class correlations	PCORR
PCov	Pooled within-class covariances	PCOV
PSSCP	Pooled within-class SSCP matrix	PSSCP
PStdMeans	Pooled standardized class means	STDMEAN
SimpleStatistics	Simple statistics	SIMPLE
Steps	Stepwise selection entry/removal	default
Summary	Stepwise selection summary	default
TCorr	Total-sample correlations	TCORR
TCov	Total-sample covariances	TCOV
TSSCP	Total-sample SSCP matrix	TSSCP
TStdMeans	Total standardized class means	STDMEAN
Variables	Variable lists	default
WCorr	Within-class correlations	WCORR
WCov	Within-class covariances	WCOV
WSSCP	Within-class SSCP matrices	WSSCP

Example: STEPDISC Procedure

Example 82.1: Performing a Stepwise Discriminant Analysis

The iris data published by Fisher (1936) have been widely used for examples in discriminant analysis and cluster analysis. The sepal length, sepal width, petal length, and petal width are measured in millimeters on 50 iris specimens from each of three species: *Iris setosa*, *I. versicolor*, and *I. virginica*.

```
title 'Fisher (1936) Iris Data';

proc format;
   value specname
      1='Setosa    '
      2='Versicolor'
      3='Virginica ';
run;

data iris;
   input SepalLength SepalWidth PetalLength PetalWidth
         Species @@;
   format Species specname.;
   label SepalLength='Sepal Length in mm.'
         SepalWidth ='Sepal Width in mm.'
         PetalLength='Petal Length in mm.'
         PetalWidth ='Petal Width in mm.';
   datalines;
50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3
63 28 51 15 3 46 34 14 03 1 69 31 51 23 3 62 22 45 15 2
59 32 48 18 2 46 36 10 02 1 61 30 46 14 2 60 27 51 16 2

... more lines ...

63 33 60 25 3 53 37 15 02 1
;
```

A stepwise discriminant analysis is performed by using stepwise selection.

In the PROC STEPDISC statement, the BSSCP and TSSCP options display the between-class SSCP matrix and the total-sample corrected SSCP matrix. By default, the significance level of an F test from an analysis of covariance is used as the selection criterion. The variable under consideration is the dependent variable, and the variables already chosen act as covariates. The following SAS statements produce Output 82.1.1 through Output 82.1.8:

```
%let _stdvar = ;
proc stepdisc data=iris bsscp tsscp;
   class Species;
   var SepalLength SepalWidth PetalLength PetalWidth;
run;
```

Output 82.1.1 Iris Data: Summary Information

```
                          Fisher (1936) Iris Data

                           The STEPDISC Procedure

                The Method for Selecting Variables is STEPWISE

    Total Sample Size     150          Variable(s) in the Analysis        4
    Class Levels            3          Variable(s) Will Be Included       0
                                       Significance Level to Enter     0.15
                                       Significance Level to Stay      0.15

                  Number of Observations Read            150
                  Number of Observations Used            150

                           Class Level Information

                     Variable
       Species       Name          Frequency       Weight    Proportion

       Setosa        Setosa               50      50.0000      0.333333
       Versicolor    Versicolor           50      50.0000      0.333333
       Virginica     Virginica            50      50.0000      0.333333
```

Output 82.1.2 Iris Data: Between-Class and Total-Sample SSCP Matrices

```
                          Fisher (1936) Iris Data

                           The STEPDISC Procedure

                         Between-Class SSCP Matrix

Variable    Label             SepalLength   SepalWidth  PetalLength   PetalWidth

Sepal       Sepal Length       6321.21333  -1995.26667  16524.84000   7127.93333
Length      in mm.
SepalWidth  Sepal Width       -1995.26667   1134.49333  -5723.96000  -2293.26667
            in mm.
Petal       Petal Length      16524.84000  -5723.96000  43710.28000  18677.40000
Length      in mm.
PetalWidth  Petal Width        7127.93333  -2293.26667  18677.40000   8041.33333
            in mm.
```

Output 82.1.2 *continued*

Total-Sample SSCP Matrix

Variable	Label	SepalLength	SepalWidth	PetalLength	PetalWidth
Sepal Length	Sepal Length in mm.	10216.83333	-632.26667	18987.30000	7692.43333
SepalWidth	Sepal Width in mm.	-632.26667	2830.69333	-4911.88000	-1812.42667
Petal Length	Petal Length in mm.	18987.30000	-4911.88000	46432.54000	19304.58000
PetalWidth	Petal Width in mm.	7692.43333	-1812.42667	19304.58000	8656.99333

In step 1, the tolerance is 1.0 for each variable under consideration because no variables have yet entered the model. The variable PetalLength is selected because its F statistic, 1180.161, is the largest among all variables.

Output 82.1.3 Iris Data: Stepwise Selection Step 1

Fisher (1936) Iris Data

The STEPDISC Procedure
Stepwise Selection: Step 1

Statistics for Entry, DF = 2, 147

Variable	Label	R-Square	F Value	Pr > F	Tolerance
SepalLength	Sepal Length in mm.	0.6187	119.26	<.0001	1.0000
SepalWidth	Sepal Width in mm.	0.4008	49.16	<.0001	1.0000
PetalLength	Petal Length in mm.	0.9414	1180.16	<.0001	1.0000
PetalWidth	Petal Width in mm.	0.9289	960.01	<.0001	1.0000

Variable PetalLength will be entered.

Variable(s) That Have Been Entered

PetalLength

Multivariate Statistics

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.058628	1180.16	2	147	<.0001
Pillai's Trace	0.941372	1180.16	2	147	<.0001
Average Squared Canonical Correlation	0.470686				

In step 2, with the variable PetalLength already in the model, PetalLength is tested for removal before a new variable is selected for entry. Since PetalLength meets the criterion to stay, it is used as a covariate in the analysis of covariance for variable selection. The variable SepalWidth is selected because its F statistic, 43.035, is the largest among all variables not in the model and because its associated tolerance, 0.8164, meets the criterion to enter. The process is repeated in steps 3 and 4. The variable PetalWidth is entered in step 3, and the variable SepalLength is entered in step 4.

Output 82.1.4 Iris Data: Stepwise Selection Step 2

Fisher (1936) Iris Data

The STEPDISC Procedure
Stepwise Selection: Step 2

Statistics for Removal, DF = 2, 147

Variable	Label	R-Square	F Value	Pr > F
PetalLength	Petal Length in mm.	0.9414	1180.16	<.0001

No variables can be removed.

Statistics for Entry, DF = 2, 146

Variable	Label	Partial R-Square	F Value	Pr > F	Tolerance
SepalLength	Sepal Length in mm.	0.3198	34.32	<.0001	0.2400
SepalWidth	Sepal Width in mm.	0.3709	43.04	<.0001	0.8164
PetalWidth	Petal Width in mm.	0.2533	24.77	<.0001	0.0729

Variable SepalWidth will be entered.

Variable(s) That Have Been Entered

SepalWidth PetalLength

Multivariate Statistics

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.036884	307.10	4	292	<.0001
Pillai's Trace	1.119908	93.53	4	294	<.0001
Average Squared Canonical Correlation	0.559954				

Output 82.1.5 Iris Data: Stepwise Selection Step 3

Fisher (1936) Iris Data

The STEPDISC Procedure
Stepwise Selection: Step 3

Statistics for Removal, DF = 2, 146

Variable	Label	Partial R-Square	F Value	Pr > F
SepalWidth	Sepal Width in mm.	0.3709	43.04	<.0001
PetalLength	Petal Length in mm.	0.9384	1112.95	<.0001

No variables can be removed.

Statistics for Entry, DF = 2, 145

Variable	Label	Partial R-Square	F Value	Pr > F	Tolerance
SepalLength	Sepal Length in mm.	0.1447	12.27	<.0001	0.1323
PetalWidth	Petal Width in mm.	0.3229	34.57	<.0001	0.0662

Variable PetalWidth will be entered.

Variable(s) That Have Been Entered

SepalWidth PetalLength PetalWidth

Multivariate Statistics

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.024976	257.50	6	290	<.0001
Pillai's Trace	1.189914	71.49	6	292	<.0001
Average Squared Canonical Correlation	0.594957				

Output 82.1.6 Iris Data: Stepwise Selection Step 4

Fisher (1936) Iris Data

The STEPDISC Procedure
Stepwise Selection: Step 4

Statistics for Removal, DF = 2, 145

Variable	Label	Partial R-Square	F Value	Pr > F
SepalWidth	Sepal Width in mm.	0.4295	54.58	<.0001
PetalLength	Petal Length in mm.	0.3482	38.72	<.0001
PetalWidth	Petal Width in mm.	0.3229	34.57	<.0001

No variables can be removed.

Statistics for Entry, DF = 2, 144

Variable	Label	Partial R-Square	F Value	Pr > F	Tolerance
SepalLength	Sepal Length in mm.	0.0615	4.72	0.0103	0.0320

Variable SepalLength will be entered.

All variables have been entered.

Multivariate Statistics

Statistic	Value	F Value	Num DF	Den DF	Pr > F
Wilks' Lambda	0.023439	199.15	8	288	<.0001
Pillai's Trace	1.191899	53.47	8	290	<.0001
Average Squared Canonical Correlation	0.595949				

Since no more variables can be added to or removed from the model, the procedure stops at step 5 and displays a summary of the selection process.

Output 82.1.7 Iris Data: Stepwise Selection Step 5

Fisher (1936) Iris Data

The STEPDISC Procedure
Stepwise Selection: Step 5

Statistics for Removal, DF = 2, 144

Variable	Label	Partial R-Square	F Value	Pr > F
SepalLength	Sepal Length in mm.	0.0615	4.72	0.0103
SepalWidth	Sepal Width in mm.	0.2335	21.94	<.0001
PetalLength	Petal Length in mm.	0.3308	35.59	<.0001
PetalWidth	Petal Width in mm.	0.2570	24.90	<.0001

No variables can be removed.

Output 82.1.8 Iris Data: Stepwise Selection Summary

No further steps are possible.

Fisher (1936) Iris Data

The STEPDISC Procedure

Stepwise Selection Summary

Step	Number In	Entered	Removed	Label	Partial R-Square	F Value	Pr > F	Wilks' Lambda	Pr < Lambda
1	1	PetalLength		Petal Length in mm.	0.9414	1180.16	<.0001	0.05862828	<.0001
2	2	SepalWidth		Sepal Width in mm.	0.3709	43.04	<.0001	0.03688411	<.0001
3	3	PetalWidth		Petal Width in mm.	0.3229	34.57	<.0001	0.02497554	<.0001
4	4	SepalLength		Sepal Length in mm.	0.0615	4.72	0.0103	0.02343863	<.0001

Step	Number In	Entered	Removed	Average Squared Canonical Correlation	Pr > ASCC
1	1	PetalLength		0.47068586	<.0001
2	2	SepalWidth		0.55995394	<.0001
3	3	PetalWidth		0.59495691	<.0001
4	4	SepalLength		0.59594941	<.0001

PROC STEPDISC automatically creates a list of the selected variables and stores it in a macro variable. You can submit the following statement to see the list of selected variables:

```
* print the macro variable list;
%put &_stdvar;
```

The macro variable _StdVar contains the following variable list:

```
SepalLength SepalWidth PetalLength PetalWidth
```

You could use this macro variable if you want to analyze these variables in subsequent steps as follows:

```
proc discrim data=iris;
   class Species;
   var &_stdvar;
run;
```

The results of this step are not shown.

References

Costanza, M. C. and Afifi, A. A. (1979), "Comparison of Stopping Rules in Forward Stepwise Discriminant Analysis," *Journal of the American Statistical Association*, 74, 777–785.

Fisher, R. A. (1936), "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, 7, 179–188.

Jennrich, R. I. (1977), "Stepwise Discriminant Analysis," in K. Enslein, A. Ralston, and H. Wilf, eds., *Statistical Methods for Digital Computers*, New York: John Wiley & Sons.

Klecka, W. R. (1980), *Discriminant Analysis*, Sage University Paper Series on Quantitative Applications in the Social Sciences, 07-019, Beverly Hills and London: Sage Publications.

Puranen, J. (1917), "Fish Catch data set (1917)," Journal of Statistics Education Data Archive, last accessed May 22, 2009.
URL http://www.amstat.org/publications/jse/jse_data_archive.html

Chapter 83

The SURVEYFREQ Procedure

Contents

Overview: SURVEYFREQ Procedure **6294**
Getting Started: SURVEYFREQ Procedure **6294**
Syntax: SURVEYFREQ Procedure **6301**
PROC SURVEYFREQ Statement 6302
BY Statement 6309
CLUSTER Statement 6310
REPWEIGHTS Statement 6310
STRATA Statement 6312
TABLES Statement 6312
WEIGHT Statement 6320
Details: SURVEYFREQ Procedure **6320**
Specifying the Sample Design 6320
Domain Analysis 6323
Missing Values 6323
Statistical Computations 6326
Variance Estimation 6326
Definitions and Notation 6328
Totals 6329
Covariance of Totals 6330
Proportions 6331
Row and Column Proportions 6332
Balanced Repeated Replication (BRR) 6334
The Jackknife 6337
Confidence Limits 6339
Degrees of Freedom 6340
Coefficient of Variation 6341
Design Effect 6341
Expected Weighted Frequency 6342
Risks and Risk Difference 6342
Odds Ratio and Relative Risks 6344
Rao-Scott Chi-Square Test 6346
Rao-Scott Likelihood Ratio Chi-Square Test 6350
Wald Chi-Square Test 6352
Wald Log-Linear Chi-Square Test 6354

Output Data Sets 6355
Displayed Output 6356
ODS Table Names 6362
Examples: SURVEYFREQ Procedure **6362**
Example 83.1: Two-Way Tables 6362
Example 83.2: Multiway Tables (Domain Analysis) 6366
Example 83.3: Output Data Sets 6368
References **6370**

Overview: SURVEYFREQ Procedure

The SURVEYFREQ procedure produces one-way to *n*-way frequency and crosstabulation tables from sample survey data. These tables include estimates of population totals, population proportions, and their standard errors. Confidence limits, coefficients of variation, and design effects are also available. The procedure provides a variety of options to customize the table display.

For one-way frequency tables, PROC SURVEYFREQ provides Rao-Scott chi-square goodness-of-fit tests, which are adjusted for the sample design. You can test a null hypothesis of equal proportions for a one-way frequency table, or you can input custom null hypothesis proportions for the test. For two-way tables, PROC SURVEYFREQ provides design-adjusted tests of independence, or no association, between the row and column variables. These tests include the Rao-Scott chi-square test, the Rao-Scott likelihood ratio test, the Wald chi-square test, and the Wald log-linear chi-square test. For 2×2 tables, PROC SURVEYFREQ computes estimates and confidence limits for risks (or row proportions), the risk difference, the odds ratio, and relative risks.

PROC SURVEYFREQ computes variance estimates based on the sample design used to obtain the survey data. The design can be a complex multistage survey design with stratification, clustering, and unequal weighting. PROC SURVEYFREQ provides a choice of variance estimation methods, which include Taylor series linearization, balanced repeated replication (BRR), and the jackknife.

Getting Started: SURVEYFREQ Procedure

The following example shows how you can use PROC SURVEYFREQ to analyze sample survey data. The example uses data from a customer satisfaction survey for a student information system (SIS), which is a software product that provides modules for student registration, class scheduling, attendance, grade reporting, and other functions.

The software company conducted a survey of school personnel who use the SIS. A probability sample of SIS users was selected from the study population, which included SIS users at middle schools and high schools in the three-state area of Georgia, South Carolina, and North Carolina. The sample design for this survey was a two-stage stratified design. A first-stage sample of schools

was selected from the list of schools in the three-state area that use the SIS. The list of schools, or the first-stage sampling frame, was stratified by state and by customer status (whether the school was a new user of the system or a renewal user). Within the first-stage strata, schools were selected with probability proportional to size and with replacement, where the size measure was school enrollment. From each sample school, five staff members were randomly selected to complete the SIS satisfaction questionnaire. These staff members included three teachers and two administrators or guidance department members.

The SAS data set SIS_Survey contains the survey results, as well as the sample design information needed to analyze the data. This data set includes an observation for each school staff member responding to the survey. The variable Response contains the staff member's response about overall satisfaction with the system.

The variable State contains the school's state, and the variable NewUser contains the school's customer status ('New Customer' or 'Renewal Customer'). These two variables determine the first-stage strata from which schools were selected. The variable School contains the school identification code and identifies the first-stage (primary) sampling units, or clusters. The variable SamplingWeight contains the overall sampling weight for each respondent. Overall sampling weights were computed from the selection probabilities at each stage of sampling and were adjusted for nonresponse.

Other variables in the data set SIS_Survey include SchoolType and Department. The variable SchoolType identifies the school as a high school or a middle school. The variable Department identifies the staff member as a teacher, or an administrator or guidance department member.

The following PROC SURVEYFREQ statements request a one-way frequency table for the variable Response:

```
title 'Student Information System Survey';
proc surveyfreq data=SIS_Survey;
   tables  Response;
   strata  State NewUser;
   cluster School;
   weight  SamplingWeight;
run;
```

The PROC SURVEYFREQ statement invokes the procedure and identifies the input data set to be analyzed. The TABLES statement requests a one-way frequency table for the variable Response. The table request syntax for PROC SURVEYFREQ is very similar to the table request syntax for PROC FREQ. This example shows a request for a single one-way table, but you can also request two-way tables and multiway tables. As in PROC FREQ, you can request more than one table in the same TABLES statement, and you can use multiple TABLES statements in the same invocation of the procedure.

The STRATA, CLUSTER, and WEIGHT statements provide sample design information for the procedure, so that the analysis is done according to the sample design used for the survey, and the estimates apply to the study population. The STRATA statement names the variables State and NewUser, which identify the first-stage strata. Note that the design for this example also includes stratification at the second stage of selection (by type of school personnel), but you specify only the first-stage strata for PROC SURVEYFREQ. The CLUSTER statement names the variable School, which identifies the clusters or primary sample units (PSUs). The WEIGHT statement names the sampling weight variable.

Figure 83.1 and Figure 83.2 display the output produced by PROC SURVEYFREQ, which includes the "Data Summary" table and the one-way table, "Table of Response." The "Data Summary" table is produced by default unless you specify the NOSUMMARY option. This table shows there are 6 strata, 370 clusters or schools, and 1850 observations or respondents in the SIS_Survey data set. The sum of the sampling weights is approximately 39,000, which estimates the total number of school personnel in the study area that use the SIS.

Figure 83.1 SIS_Survey Data Summary

Student Information System Survey

The SURVEYFREQ Procedure

Data Summary

Number of Strata	6
Number of Clusters	370
Number of Observations	1850
Sum of Weights	38899.6482

Figure 83.2 displays the one-way table of Response, which provides estimates of the population total (weighted frequency) and the population percentage for each category, or level, of the variable Response. The response level 'Very Unsatisfied' has a frequency of 304, which means that 304 sample respondents fall into this category. It is estimated that 17.17% of all school personnel in the study population fall into this category, and the standard error of this estimate is 1.29%. Note that the estimates apply to the population of all SIS users in the study area, as opposed to describing only the sample of 1850 respondents. The estimate of the total number of school personnel that are 'Very Unsatisfied' is 6,678, with a standard deviation of 502. The standard errors computed by PROC SURVEYFREQ are based on the multistage stratified design of the survey. This differs from some of the traditional analysis procedures, which assume the design is simple random sampling from an infinite population.

Figure 83.2 One-Way Table of Response

Table of Response

Response	Frequency	Weighted Frequency	Std Dev of Wgt Freq	Percent	Std Err of Percent
Very Unsatisfied	304	6678	501.61039	17.1676	1.2872
Unsatisfied	326	6907	495.94101	17.7564	1.2712
Neutral	581	12291	617.20147	31.5965	1.5795
Satisfied	455	9309	572.27868	23.9311	1.4761
Very Satisfied	184	3714	370.66577	9.5483	0.9523
Total	1850	38900	129.85268	100.000	

The following PROC SURVEYFREQ statements request confidence limits for the percentages and a chi-square goodness-of-fit test for the one-way table of Response:

```
proc surveyfreq data=SIS_Survey nosummary;
   tables  Response / cl nowt chisq;
   strata  State  NewUser;
   cluster School;
   weight  SamplingWeight;
 run;
```

The NOSUMMARY option in the PROC statement suppresses the "Data Summary" table. In the TABLES statement, the CL option requests confidence limits for the percentages in the one-way table. The NOWT option suppresses display of the weighted frequencies and their standard deviations. The CHISQ option requests a Rao-Scott chi-square goodness-of-fit test.

Figure 83.3 shows the one-way table of Response, which includes confidence limits for the percentages. The 95% confidence limits for the percentage of users that are 'Very Unsatisfied' are 14.64% and 19.70%. To change the α level of the confidence limits, which equals 5% by default, you can use the ALPHA= option. Like the other estimates and standard errors produced by PROC SURVEYFREQ, these confidence limit computations take into account the complex sample design of the survey, and the results apply to the entire study population.

Figure 83.3 Confidence Limits for Response Percentages

Student Information System Survey

The SURVEYFREQ Procedure

Table of Response

Response	Frequency	Percent	Std Err of Percent	95% Confidence Limits for Percent	
Very Unsatisfied	304	17.1676	1.2872	14.6364	19.6989
Unsatisfied	326	17.7564	1.2712	15.2566	20.2562
Neutral	581	31.5965	1.5795	28.4904	34.7026
Satisfied	455	23.9311	1.4761	21.0285	26.8338
Very Satisfied	184	9.5483	0.9523	7.6756	11.4210
Total	1850	100.000			

Figure 83.4 shows the chi-square goodness-of-fit results for the table of Response. The null hypothesis for this test is equal proportions for the levels of the one-way table. (To test a null hypothesis of specified proportions instead of equal proportions, you can use the TESTP= option to specify null hypothesis proportions.)

The chi-square test invoked by the CHISQ option is the Rao-Scott design-adjusted chi-square test, which takes the sample design into account and provides inferences for the entire study population. To produce the Rao-Scott chi-square statistic, PROC SURVEYFREQ first computes the usual Pearson chi-square statistic based on the weighted frequencies, and then adjusts this value with a design correction. An F approximation is also provided. For the table of Response, the F value is 30.0972 with a p-value of <0.0001, which indicates rejection of the null hypothesis of equal proportions for all response levels.

Figure 83.4 Chi-Square Goodness-of-Fit Test for Response

Rao-Scott Chi-Square Test	
Pearson Chi-Square	251.8105
Design Correction	2.0916
Rao-Scott Chi-Square	120.3889
DF	4
Pr > ChiSq	<.0001
F Value	30.0972
Num DF	4
Den DF	1456
Pr > F	<.0001

Sample Size = 1850

Continuing to analyze the SIS_Survey data, the following PROC SURVEYFREQ statements request a two-way table of SchoolType by Response:

```
proc surveyfreq data=SIS_Survey nosummary;
   tables  SchoolType * Response;
   strata  State NewUser;
   cluster School;
   weight  SamplingWeight;
run;
```

The STRATA, CLUSTER, and WEIGHT statements do not change from the one-way table analysis, because the sample design and the input data set are the same. These SURVEYFREQ statements request a different table but specify the same sample design information.

Figure 83.5 shows the two-way table produced for SchoolType by Response. The first variable named in the two-way table request, SchoolType, is referred to as the *row variable*, and the second variable, Response, is referred to as the *column variable*. Two-way tables display all column variable levels for each row variable level. This two-way table lists all levels of the column variable Response for each level of the row variable SchoolType, 'Middle School' and 'High School'. Also SchoolType = 'Total' shows the distribution of Response overall for both types of schools. And Response = 'Total' provides totals over all levels of response, for each type of school and overall. To suppress these totals, you can specify the NOTOTAL option.

Figure 83.5 Two-Way Table of SchoolType by Response

Student Information System Survey

The SURVEYFREQ Procedure

Table of SchoolType by Response

SchoolType	Response	Frequency	Weighted Frequency	Std Dev of Wgt Freq	Percent	Std Err of Percent
Middle School	Very Unsatisfied	116	2496	351.43834	6.4155	0.9030
	Unsatisfied	109	2389	321.97957	6.1427	0.8283
	Neutral	234	4856	504.20553	12.4847	1.2953
	Satisfied	197	4064	443.71188	10.4467	1.1417
	Very Satisfied	94	1952	302.17144	5.0193	0.7758
	Total	750	15758	1000	40.5089	2.5691
High School	Very Unsatisfied	188	4183	431.30589	10.7521	1.1076
	Unsatisfied	217	4518	446.31768	11.6137	1.1439
	Neutral	347	7434	574.17175	19.1119	1.4726
	Satisfied	258	5245	498.03221	13.4845	1.2823
	Very Satisfied	90	1762	255.67158	4.5290	0.6579
	Total	1100	23142	1003	59.4911	2.5691
Total	Very Unsatisfied	304	6678	501.61039	17.1676	1.2872
	Unsatisfied	326	6907	495.94101	17.7564	1.2712
	Neutral	581	12291	617.20147	31.5965	1.5795
	Satisfied	455	9309	572.27868	23.9311	1.4761
	Very Satisfied	184	3714	370.66577	9.5483	0.9523
	Total	1850	38900	129.85268	100.000	

By default, without any other TABLES statement options, a two-way table displays the frequency, the weighted frequency and its standard deviation, and the percentage and its standard error for each table cell, or combination of row and column variable levels. But there are several options available to customize your table display by adding more information or by suppressing some of the default information.

The following PROC SURVEYFREQ statements request a two-way table of SchoolType by Response that displays row percentages, and also request a chi-square test of association between the two variables:

```
proc surveyfreq data=SIS_Survey nosummary;
   tables  SchoolType * Response / row nowt chisq;
   strata  State NewUser;
   cluster School;
   weight  SamplingWeight;
run;
```

The ROW option in the TABLES statement requests row percentages, which give the distribution of Response within each level of the row variable SchoolType. The NOWT option suppresses display of the weighted frequencies and their standard deviations. The CHISQ option requests a Rao-Scott chi-square test of association between SchoolType and Response.

Figure 83.6 displays the two-way table of SchoolType by Response. For middle schools, it is estimated that 25.79% of school personnel are satisfied with the student information system and 12.39% are very satisfied. For high schools, these estimates are 22.67% and 7.61%, respectively.

Figure 83.7 displays the chi-square test results. The Rao-Scott chi-square statistic equals 9.04, and the corresponding F value is 2.26 with a p-value of 0.0605. This indicates an association between school type (middle school or high school) and satisfaction with the student information system at the 10% significance level.

Figure 83.6 Two-Way Table with Row Percentages

Student Information System Survey

The SURVEYFREQ Procedure

Table of SchoolType by Response

SchoolType	Response	Frequency	Percent	Std Err of Percent	Row Percent	Std Err of Row Percent
Middle School	Very Unsatisfied	116	6.4155	0.9030	15.8373	1.9920
	Unsatisfied	109	6.1427	0.8283	15.1638	1.8140
	Neutral	234	12.4847	1.2953	30.8196	2.5173
	Satisfied	197	10.4467	1.1417	25.7886	2.2947
	Very Satisfied	94	5.0193	0.7758	12.3907	1.7449
	Total	750	40.5089	2.5691	100.000	
High School	Very Unsatisfied	188	10.7521	1.1076	18.0735	1.6881
	Unsatisfied	217	11.6137	1.1439	19.5218	1.7280
	Neutral	347	19.1119	1.4726	32.1255	2.0490
	Satisfied	258	13.4845	1.2823	22.6663	1.9240
	Very Satisfied	90	4.5290	0.6579	7.6128	1.0557
	Total	1100	59.4911	2.5691	100.000	
Total	Very Unsatisfied	304	17.1676	1.2872		
	Unsatisfied	326	17.7564	1.2712		
	Neutral	581	31.5965	1.5795		
	Satisfied	455	23.9311	1.4761		
	Very Satisfied	184	9.5483	0.9523		
	Total	1850	100.000			

Figure 83.7 Chi-Square Test of No Association

Rao-Scott Chi-Square Test	
Pearson Chi-Square	18.7829
Design Correction	2.0766
Rao-Scott Chi-Square	9.0450
DF	4
Pr > ChiSq	0.0600
F Value	2.2613
Num DF	4
Den DF	1456
Pr > F	0.0605
Sample Size = 1850	

Syntax: SURVEYFREQ Procedure

The following statements are available in PROC SURVEYFREQ:

PROC SURVEYFREQ < *options* > ;
BY *variables* ;
CLUSTER *variables* ;
REPWEIGHTS *variables* < / *options* > ;
STRATA *variables* < / *option* > ;
TABLES *requests* < / *options* > ;
WEIGHT *variable* ;

The PROC SURVEYFREQ statement invokes the procedure, identifies the data set to be analyzed, and specifies the variance estimation method. The PROC SURVEYFREQ statement is required.

The TABLES statement specifies frequency or crosstabulation tables and requests tests and statistics for those tables. The STRATA statement lists the variables that form the strata in a stratified sample design. The CLUSTER statement specifies cluster identification variables in a clustered sample design. The WEIGHT statement names the sampling weight variable. The REPWEIGHTS statement names replicate weight variables for BRR or jackknife variance estimation. The BY statement requests completely separate analyses of groups defined by the BY variables.

All statements can appear multiple times except the PROC SURVEYFREQ statement and the WEIGHT statement, which can appear only once.

The rest of this section gives detailed syntax information for the BY, CLUSTER, REPWEIGHTS, STRATA, TABLES, and WEIGHT statements in alphabetical order after the description of the PROC SURVEYFREQ statement.

PROC SURVEYFREQ Statement

PROC SURVEYFREQ < *options* > **;**

The PROC SURVEYFREQ statement invokes the procedure. In this statement, you identify the data set to be analyzed, specify the variance estimation method, and provide sample design information. The DATA= option names the input data set to be analyzed. The VARMETHOD= option specifies the variance estimation method, which is the Taylor series method by default. For Taylor series variance estimation, you can include a finite population correction factor in the analysis by providing either the sampling rate or population total with the RATE= or TOTAL= option. If your design is stratified, with different sampling rates or totals for different strata, then you can input these stratum rates or totals in a SAS data set that contains the stratification variables.

You can specify the following options in the PROC SURVEYFREQ statement.

DATA=*SAS-data-set*
names the SAS data set to be analyzed by PROC SURVEYFREQ. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

MISSING
treats missing values as a valid (nonmissing) category for all categorical variables, which include TABLES, STRATA, and CLUSTER variables.

By default, if you do not specify the MISSING option, an observation is excluded from the analysis if it has a missing value for any STRATA or CLUSTER variable. Additionally, PROC SURVEYFREQ excludes an observation from a frequency or crosstabulation table if that observation has a missing value for any of the variables in the table request, unless you specify the MISSING option. For more information, see the section "Missing Values" on page 6323.

NOMCAR
includes observations with missing values of TABLES variables in the variance computation as *not missing completely at random* (NOMCAR) for Taylor series variance estimation. When you specify the NOMCAR option, PROC SURVEYFREQ computes variance estimates by analyzing the nonmissing values as a domain or subpopulation, where the entire population includes both nonmissing and missing domains. See the section "Missing Values" on page 6323 for details.

By default, PROC SURVEYFREQ completely excludes an observation from a frequency or crosstabulation table (and the corresponding variance computations) if that observation has a missing value for any of the variables in the table request, unless you specify the MISSING option. Note that the NOMCAR option has no effect when you specify the MISSING option, which treats missing values as a valid nonmissing level.

The NOMCAR option applies only to Taylor series variance estimation. The replication methods, which you request with the VARMETHOD=BRR and VARMETHOD=JACKKNIFE options, do not use the NOMCAR option.

NOSUMMARY

suppresses the display of the "Data Summary" table, which PROC SURVEYFREQ produces by default. For details about this table, see the section "Data Summary Table" on page 6356.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the order in which the values of the frequency and crosstabulation table variables are displayed. PROC SURVEYFREQ interprets values of the ORDER= option as follows:

DATA	orders values according to their order in the input data set.
FORMATTED	orders values by their formatted values (in ascending order). This order is operating-environment dependent.
FREQ	orders values by their descending frequency counts. The frequency count of a variable value is its (nonweighted) frequency of occurrence or sample size, and not its weighted frequency.
INTERNAL	orders values by their unformatted values, which yields the same order that the SORT procedure does. This order is operating-environment dependent.

By default, ORDER=INTERNAL.

PAGE

displays only one table per page. Otherwise, PROC SURVEYFREQ displays multiple tables per page as space permits.

RATE=*value* | *SAS-data-set*

R=*value* | *SAS-data-set*

specifies the sampling rate as a nonnegative *value*, or identifies an input data set that gives the stratum sampling rates in a variable named _RATE_. PROC SURVEYFREQ uses this information to compute a finite population correction for Taylor series variance estimation. The procedure does not use the RATE= option for BRR or jackknife variance estimation, which you request with the VARMETHOD=BRR or VARMETHOD=JACKKNIFE option.

If your sample design has multiple stages, you should specify the *first-stage sampling rate*, which is the ratio of the number of primary sampling units (PSUs) selected to the total number of PSUs in the population.

For a nonstratified sample design, or for a stratified sample design with the same sampling rate in all strata, you should specify a nonnegative *value* for the RATE= option. If your design is stratified with different sampling rates in different strata, then you should name a SAS data set that contains the stratification variables and the stratum sampling rates. See the section "Population Totals and Sampling Rates" on page 6322 for details.

The sampling rate *value* must be a nonnegative number. You can specify *value* as a number between 0 and 1. Or you can specify *value* in percentage form as a number between 1 and 100, and PROC SURVEYFREQ converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

If you do not specify the RATE= or TOTAL= option, then the Taylor series variance estimation does not include a finite population correction. You cannot specify both the TOTAL= option and the RATE= option in the same PROC SURVEYFREQ statement.

TOTAL=*value* | *SAS-data-set*

N=*value* | *SAS-data-set*

specifies the total number of primary sampling units (PSUs) in the study population as a positive *value*, or identifies an input data set that gives the stratum population totals in a variable named _TOTAL_. PROC SURVEYFREQ uses this information to compute a finite population correction for Taylor series variance estimation. The procedure does not use the TOTAL= option for BRR or jackknife variance estimation, which you request with the VARMETHOD=BRR or VARMETHOD=JACKKNIFE option.

For a nonstratified sample design, or for a stratified sample design with the same population total in all strata, you should specify a positive *value* for the TOTAL= option. If your sample design is stratified with different population totals in different strata, then you should name a SAS data set that contains the stratification variables and the stratum totals. See the section "Population Totals and Sampling Rates" on page 6322 for details.

If you do not specify the TOTAL= or RATE= option, then the Taylor series variance estimation does not include a finite population correction. You cannot specify both the TOTAL= option and the RATE= option in the same PROC SURVEYFREQ statement.

VARHEADER=LABEL | NAME | NAMELABEL

specifies the variable identification to use in the displayed output. By default VARHEADER=NAME, which displays variable names in the output. The VARHEADER= option affects the headers of the variable level columns in one-way frequency tables, crosstabulation tables, and the "Stratum Information" table. The VARHEADER= option also controls variable identification in the table headers. The following values are available for the VARHEADER= option:

LABEL	displays the variable label.
NAME	displays the variable name.
NAMELABEL	displays both the variable name and label, with the label in parentheses following the name—that is, *Name (Label)*.

VARMETHOD=BRR < (*method-options***) > | JACKKNIFE < (***method-options***) > | TAYLOR**

specifies the variance estimation method. VARMETHOD=TAYLOR requests the Taylor series method, which is the default if you do not specify the VARMETHOD= option or a REPWEIGHTS statement. VARMETHOD=BRR requests variance estimation by balanced repeated replication (BRR), and VARMETHOD=JACKKNIFE requests variance estimation by the delete-1 jackknife method.

For VARMETHOD=BRR and VARMETHOD=JACKKNIFE, you can specify *method-options* in parentheses following the variance method name. Table 83.1 summarizes the available *method-options*.

Table 83.1 Variance Estimation Options

VARMETHOD=	Variance Estimation Method	*Method-Options*
BRR	Balanced repeated replication	DFADJ FAY <=*value*> HADAMARD=*SAS-data-set* OUTWEIGHTS=*SAS-data-set* PRINTH REPS=*number*
JACKKNIFE	Jackknife	DFADJ OUTJKCOEFS=*SAS-data-set* OUTWEIGHTS=*SAS-data-set*
TAYLOR	Taylor series linearization	*None*

The following values are available for the VARMETHOD= option.

BRR < *method-options* > requests variance estimation by balanced repeated replication (BRR). The BRR method requires a stratified sample design with two primary sampling units (PSUs) in each stratum. If you specify the VARMETHOD=BRR option, you must also specify a STRATA statement unless you provide replicate weights with a REPWEIGHTS statement. See the section "Balanced Repeated Replication (BRR)" on page 6334 for details.

You can specify the following *method-options* in parentheses after the VARMETHOD=BRR option.

DFADJ

computes the degrees of freedom as the number of nonmissing strata for the individual table request. The degrees of freedom for VARMETHOD=BRR equal the number of strata, which by default is based on all valid observations in the data set. But if you specify the DFADJ method-option, PROC SURVEYFREQ excludes from the count any empty strata that occur when observations with missing values of the TABLES variables are removed from the analysis for that table.

See the section "Degrees of Freedom" on page 6340 for more information. See the section "Data Summary Table" on page 6356 for details about valid observations.

The DFADJ method-option has no effect when you specify the MISSING option, which treats missing values as a valid nonmissing level. The DFADJ method-option is not used when you specify the degrees of freedom in the DF= option in the TABLES statement.

The DFADJ method-option cannot be used when you provide replicate weights with a REPWEIGHTS statement. When you use a REPWEIGHTS statement, the degrees of freedom equal the number

of REPWEIGHTS variables (or replicates), unless you specify an alternative value in the DF= option in the REPWEIGHTS or TABLES statement.

FAY < =*value* >
requests Fay's method, which is a modification of the BRR method. See the section "Fay's BRR Method" on page 6335 for details.

You can specify the *value* of the Fay coefficient, which is used in converting the original sampling weights to replicate weights. The Fay coefficient must be a nonnegative number less than 1. By default, the value of the Fay coefficient equals 0.5.

HADAMARD=*SAS-data-set*
H=*SAS-data-set*
names a SAS data set that contains the Hadamard matrix for BRR replicate construction. If you do not provide a Hadamard matrix with the HADAMARD= method-option, PROC SURVEYFREQ generates an appropriate Hadamard matrix for replicate construction. See the sections "Balanced Repeated Replication (BRR)" on page 6334 and "Hadamard Matrix" on page 6336 for details.

If a Hadamard matrix of a given dimension exists, it is not necessarily unique. Therefore, if you want to use a specific Hadamard matrix, you must provide the matrix as a SAS data set in the HADAMARD=*SAS-data-set* method-option.

In the HADAMARD= input data set, each variable corresponds to a column of the Hadamard matrix, and each observation corresponds to a row of the matrix. You can use any variable names in the HADAMARD= data set. All values in the data set must equal either 1 or –1. You must ensure that the matrix you provide is indeed a Hadamard matrix—that is, $\mathbf{A}'\mathbf{A} = R\mathbf{I}$, where $\mathbf{A}$ is the Hadamard matrix of dimension R and $\mathbf{I}$ is an identity matrix. PROC SURVEYFREQ does not check the validity of the Hadamard matrix that you provide.

The HADAMARD= input data set must contain at least H variables, where H denotes the number of first-stage strata in your design. If the data set contains more than H variables, PROC SURVEYFREQ uses only the first H variables. Similarly, the HADAMARD= input data set must contain at least H observations.

If you do not specify the REPS= method-option, then the number of replicates is taken to be the number of observations in the HADAMARD= input data set. If you specify the number of replicates—for example, REPS=*nreps*—then the first *nreps* observations in the HADAMARD= data set are used to construct the replicates.

You can specify the PRINTH method-option to display the Hadamard matrix that the procedure uses to construct replicates for BRR.

OUTWEIGHTS=*SAS-data-set*

names a SAS data set to store the replicate weights that PROC SURVEYFREQ creates for BRR variance estimation. See the section "Balanced Repeated Replication (BRR)" on page 6334 for information about replicate weights. See the section "Replicate Weights Output Data Set" on page 6355 for details about the contents of the OUTWEIGHTS= data set.

The OUTWEIGHTS= method-option is not available when you provide replicate weights with a REPWEIGHTS statement.

PRINTH

displays the Hadamard matrix used to construct replicates for BRR. When you provide the Hadamard matrix in the HADAMARD= method-option, PROC SURVEYFREQ displays only the rows and columns that are actually used to construct replicates. See the sections "Balanced Repeated Replication (BRR)" on page 6334 and "Hadamard Matrix" on page 6336 for more information.

The PRINTH method-option is not available when you provide replicate weights with a REPWEIGHTS statement because the procedure does not use a Hadamard matrix in this case.

REPS=*number*

specifies the number of replicates for BRR variance estimation. The value of *number* must be an integer greater than 1.

If you do not provide a Hadamard matrix with the HADAMARD= method-option, the number of replicates should be greater than the number of strata and should be a multiple of 4. See the section "Balanced Repeated Replication (BRR)" on page 6334 for more information. If a Hadamard matrix cannot be constructed for the REPS= value that you specify, the value is increased until a Hadamard matrix of that dimension can be constructed. Therefore, it is possible for the actual number of replicates used to be larger than the REPS= value that you specify.

If you provide a Hadamard matrix with the HADAMARD= method-option, the value of REPS= must not be less than the number of rows in the Hadamard matrix. If you provide a Hadamard matrix and do not specify the REPS= method-option, the number of replicates equals the number of rows in the Hadamard matrix.

If you do not specify the REPS= or HADAMARD= method-option and do not include a REPWEIGHTS statement, the number of replicates equals the smallest multiple of 4 that is greater than the number of strata.

If you provide replicate weights with a REPWEIGHTS statement, the procedure does not use the REPS= method-option. With a REPWEIGHTS statement, the number of replicates equals the number of REPWEIGHTS variables.

JACKKNIFE | JK < *method-options* > requests variance estimation by the delete-1 jackknife method. See the section "The Jackknife" on page 6337 for details. If you provide replicate weights with a REPWEIGHTS statement, VARMETHOD=JACKKNIFE is the default variance estimation method.

You can specify the following *method-options* in parentheses after the VARMETHOD=JACKKNIFE option:

DFADJ

computes the degrees of freedom by using the number of nonmissing strata and clusters for the individual table request. The degrees of freedom for VARMETHOD=JACKKNIFE equal the number of clusters minus the number of strata, which by default is based on all valid observations in the data set. But if you specify the DFADJ method-option, PROC SURVEYFREQ excludes any empty strata or clusters that occur when observations with missing values of the TABLES variables are removed from the analysis for that table.

See the section "Degrees of Freedom" on page 6340 for more information. See the section "Data Summary Table" on page 6356 for details about valid observations.

The DFADJ method-option has no effect when you specify the MISSING option, which treats missing values as a valid nonmissing level. The DFADJ method-option is not used when you specify the degrees of freedom in the DF= option in the TABLES statement.

The DFADJ method-option cannot be used when you provide replicate weights with a REPWEIGHTS statement. When you include a REPWEIGHTS statement, the degrees of freedom equal the number of REPWEIGHTS variables (or replicates), unless you specify an alternative value in the DF= option in the REPWEIGHTS or TABLES statement.

OUTJKCOEFS=*SAS-data-set*

names a SAS data set to store the jackknife coefficients. See the section "The Jackknife" on page 6337 for information about jackknife coefficients. See the section "Jackknife Coefficients Output Data Set" on page 6355 for details about the contents of the OUTJKCOEFS= data set.

OUTWEIGHTS=*SAS-data-set*

names a SAS data set to store the replicate weights that PROC SURVEYFREQ creates for jackknife variance estimation. See the

section "The Jackknife" on page 6337 for information about replicate weights. See the section "Replicate Weights Output Data Set" on page 6355 for details about the contents of the OUTWEIGHTS= data set.

The OUTWEIGHTS= method-option is not available when you provide replicate weights with a REPWEIGHTS statement.

TAYLOR requests Taylor series variance estimation. This is the default method if you do not specify the VARMETHOD= option or a REPWEIGHTS statement. See the section "Taylor Series Variance Estimation" on page 6327 for details.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC SURVEYFREQ to obtain separate analyses of observations in groups defined by the BY variables. The *variables* are one or more variables in the input data set.

Note that using a BY statement provides completely separate analyses of the BY groups. It does not provide a statistically valid subpopulation or domain analysis, where the total number of units in the subpopulation is not known with certainty. You should include the domain variable(s) in your TABLES request to obtain domain analysis. See the section "Domain Analysis" on page 6323 for more details.

If you specify more than one BY statement, the procedure uses only the last BY statement and ignores any previous BY statements.

When you use a BY statement, the procedure expects the input data set to be sorted in order of the BY variables. If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the option NOTSORTED or DESCENDING in the BY statement. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLUSTER Statement

CLUSTER *variables* **;**

The CLUSTER statement names variables that identify the first-stage clusters, or primary sampling units (PSUs), in a clustered sample design. The combinations of categories of CLUSTER variables define the clusters in the sample. If there is a STRATA statement, clusters are nested within strata.

If your sample design has clustering at multiple stages, you should specify only the first-stage clusters, or PSUs, in the CLUSTER statement. See the section "Specifying the Sample Design" on page 6320 for more information.

If you provide replicate weights for BRR or jackknife variance estimation with a REPWEIGHTS statement, you do not need to specify a CLUSTER statement.

The CLUSTER *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric, but the procedure treats them as categorical variables. The formatted values of the CLUSTER variables determine the CLUSTER variable levels. Thus, you can use formats to group values into levels. See the discussion of the FORMAT procedure in the *Base SAS Procedures Guide* and the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

You can use multiple CLUSTER statements to specify CLUSTER variables. The procedure uses variables from all CLUSTER statements to create clusters.

REPWEIGHTS Statement

REPWEIGHTS *variables* < */ options* > **;**

The REPWEIGHTS statement names variables that provide replicate weights for BRR or jackknife variance estimation, which you request with the VARMETHOD=BRR or VARMETHOD=JACKKNIFE option in the PROC SURVEYFREQ statement. If you do not provide replicate weights for these methods by using a REPWEIGHTS statement, then PROC SURVEYFREQ constructs replicate weights for the analysis. See the sections "Balanced Repeated Replication (BRR)" on page 6334 and "The Jackknife" on page 6337 for information about replicate weights.

Each REPWEIGHTS variable should contain the weights for a single replicate, and the number of replicates equals the number of REPWEIGHTS variables. The REPWEIGHTS variables must be numeric, and the variable values must be nonnegative numbers.

If you provide replicate weights with a REPWEIGHTS statement, you do not need to specify a CLUSTER or STRATA statement. If you use a REPWEIGHTS statement and do not specify the VARMETHOD= option in the PROC SURVEYFREQ statement, the procedure uses VARMETHOD=JACKKNIFE by default.

If you specify a REPWEIGHTS statement but do not include a WEIGHT statement, PROC SURVEYFREQ uses the average of each observation's replicate weights as the observation's weight.

You can specify the following options in the REPWEIGHTS statement after a slash (/):

DF=*df*

specifies the degrees of freedom for the analysis. The value of *df* must be a positive number. By default, the degrees of freedom equals the number of REPWEIGHTS variables. See the section "Degrees of Freedom" on page 6340 for details.

PROC SURVEYFREQ uses the DF= value in computing confidence limits for proportions, totals, and other statistics. See the section "Confidence Limits" on page 6339 for details. PROC SURVEYFREQ also uses the DF= value in computing the denominator degrees of freedom for the *F* statistics in the Rao-Scott and Wald chi-square tests. See the sections "Rao-Scott Chi-Square Test" on page 6346, "Rao-Scott Likelihood Ratio Chi-Square Test" on page 6350, "Wald Chi-Square Test" on page 6352, and "Wald Log-Linear Chi-Square Test" on page 6354 for more information.

JKCOEFS=*value*

specifies the jackknife coefficient for VARMETHOD=JACKKNIFE. The coefficient *value* must be a nonnegative number less than one. See the section "The Jackknife" on page 6337 for details about jackknife coefficients.

You can use this option to specify a single value of the jackknife coefficient, which the procedure uses for all replicates. To specify different coefficients for different replicates, use the JKCOEFS=(*values*) or JKCOEFS=*SAS-data-set* option.

JKCOEFS=(*values***)**

specifies jackknife coefficients for VARMETHOD=JACKKNIFE, where each coefficient corresponds to an individual replicate identified by a REPWEIGHTS variable. You can separate *values* with blanks or commas. The coefficient *values* must be nonnegative numbers less than one. The number of *values* must equal the number of replicate weight variables named in the REPWEIGHTS statement. List these values in the same order in which you list the corresponding replicate weight variables in the REPWEIGHTS statement.

See the section "The Jackknife" on page 6337 for details about jackknife coefficients.

To specify different coefficients for different replicates, you can also use the JKCOEFS=*SAS-data-set* option. To specify a single jackknife coefficient for all replicates, use the JKCOEFS=*value* option.

JKCOEFS=*SAS-data-set*

names a SAS data set that contains the jackknife coefficients for VARMETHOD=JACKKNIFE. You provide the jackknife coefficients in the JKCOEFS= data set variable JKCoefficient. Each coefficient value must be a nonnegative number less than one. The observations in the JKCOEFS= data set should correspond to the replicates that are identified by the REPWEIGHTS variables. Arrange the coefficients or observations in the JKCOEFS= data set in the same order in which you list the corresponding replicate weight variables in the REPWEIGHTS statement. The number of observations in the JKCOEFS= data set must not be less than the number of REPWEIGHTS variables.

See the section "The Jackknife" on page 6337 for details about jackknife coefficients.

To specify different coefficients for different replicates, you can also use the JKCOEFS=*values* option. To specify a single jackknife coefficient for all replicates, use the JKCOEFS=*value* option.

STRATA Statement

STRATA *variables* < / *option* > **;**

The STRATA statement names variables that form the strata in a stratified sample design. The combinations of levels of STRATA variables define the strata in the sample, where strata are nonoverlapping subgroups that were sampled independently.

If your sample design has stratification at multiple stages, you should identify only the first-stage strata in the STRATA statement. See the section "Specifying the Sample Design" on page 6320 for more information.

If you provide replicate weights for BRR or jackknife variance estimation with a REPWEIGHTS statement, you do not need to specify a STRATA statement.

The STRATA *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric, but the procedure treats them as categorical variables. The formatted values of the STRATA variables determine the STRATA variable levels. Thus, you can use formats to group values into levels. See the discussion of the FORMAT procedure in the *Base SAS Procedures Guide* and the discussions of the FORMAT statement and SAS formats in Bookreflrdict.

You can specify the following option in the STRATA statement after a slash (/):

LIST

displays a "Stratum Information" table, which lists all strata together with the corresponding values of the STRATA variables. This table provides the number of observations and the number of clusters in each stratum, as well as the sampling fraction if you specify the RATE= or TOTAL= option. See the section "Stratum Information Table" on page 6356 for more information.

TABLES Statement

TABLES *requests* < / *options* > **;**

The TABLES statement requests one-way to *n*-way frequency and crosstabulation tables and statistics for these tables.

If you omit the TABLES statement, PROC SURVEYFREQ generates one-way frequency tables for all DATA= data set variables that are not listed in the other statements.

The following argument is required in the TABLES statement:

requests

specify the frequency and crosstabulation tables to produce. A *request* is composed of one variable name or several variable names separated by asterisks. To request a one-way frequency table, use a single variable. To request a two-way crosstabulation table, use an asterisk between two variables. To request a multiway table (an n-way table, where n>2), separate the desired variables with asterisks. The unique values of these variables form the rows, columns, and layers of the table.

For two-way tables to multiway tables, the values of the last variable form the crosstabulation table columns, while the values of the next-to-last variable form the rows. Each level (or combination of levels) of the other variables forms one layer. PROC SURVEYFREQ produces a separate crosstabulation table for each layer. For example, a specification of A*B*C*D in a TABLES statement produces k tables, where k is the number of different combinations of levels for A and B. Each table lists the levels for D (columns) within each level of C (rows).

You can use multiple TABLES statements in a single PROC SURVEYFREQ step. You can also specify any number of table requests in a single TABLES statement. To specify multiple table requests quickly, use a grouping syntax by placing parentheses around several variables and joining other variables or variable combinations. Table 83.2 shows some examples of grouping syntax.

Table 83.2 Grouping Syntax

Request	Equivalent to
tables A*(B C);	tables A*B A*C;
tables (A B)*(C D);	tables A*C B*C A*D B*D;
tables (A B C)*D;	tables A*D B*D C*D;
tables A – – C;	tables A B C;
tables (A – – C)*D;	tables A*D B*D C*D;

The TABLES statement variables are one or more variables from the DATA= input data set. These variables can be either character or numeric, but the procedure treats them as categorical variables. PROC SURVEYFREQ uses the formatted values of the TABLES variable to determine the categorical variable levels. So if you assign a format to a variable with a FORMAT statement, PROC SURVEYFREQ formats the values before dividing observations into the levels of a frequency or crosstabulation table. See the discussion of the FORMAT procedure in the *Base SAS Procedures Guide* and the discussions of the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary*.

The frequency or crosstabulation table lists the values of both character and numeric variables in ascending order based on internal (unformatted) variable values unless you change the order with the ORDER= option. To list the values in ascending order by formatted value, use ORDER=FORMATTED in the PROC SURVEYFREQ statement.

Without Options

If you request a frequency or crosstabulation table without specifying options, PROC SURVEYFREQ produces the following for each table level or cell:

- frequency, or sample size
- weighted frequency, which estimates the population total
- standard deviation of the weighted frequency
- percentage, which estimates the population proportion
- standard error of the percentage

The table displays weighted frequencies if your analysis includes a WEIGHT statement, or if you specify the WTFREQ option in the TABLES statement. The table also displays the number of observations with missing values. See the sections "One-Way Frequency Tables" on page 6357 and "Crosstabulation Tables" on page 6358 for more information.

Options

Table 83.3 lists the options available in the TABLES statement. Descriptions follow in alphabetical order.

Table 83.3 TABLES Statement Options

Option	Description
Control Statistical Analysis	
ALPHA=	sets the level for confidence limits
CHISQ	requests Rao-Scott chi-square test
CHISQ1	requests Rao-Scott modified chi-square test
DF=	specifies degrees of freedom
LRCHISQ	requests Rao-Scott likelihood ratio test
LRCHISQ1	requests Rao-Scott modified likelihood ratio test
OR	requests odds ratio and relative risks
RISK	requests risks and risk difference
TESTP=	specifies null proportions for one-way chi-square test
WCHISQ	requests Wald chi-square test
WLLCHISQ	requests Wald log-linear chi-square test
Control Additional Table Information	
CL	displays confidence limits for percentages
CLWT	displays confidence limits for weighted frequencies
COL	displays column percentages and standard errors
CV	displays coefficients of variation for percentages
CVWT	displays coefficients of variation for weighted frequencies
DEFF	displays design effects for percentages
EXPECTED	displays expected weighted frequencies (two-way tables)

Table 83.3 *continued*

Option	Description
ROW	displays row percentages and standard errors
VAR	displays variances for percentages
VARWT	displays variances for weighted frequencies
WTFREQ	displays totals and standard errors when there is no WEIGHT statement
Control Displayed Output	
NOCELLPERCENT	suppresses display of overall percentages
NOFREQ	suppresses display of frequency counts
NOPERCENT	suppresses display of all percentages
NOPRINT	suppresses display of tables but displays statistical tests
NOSPARSE	suppresses display of zero rows and columns
NOSTD	suppresses display of standard errors for all estimates
NOTOTAL	suppresses display of row and column totals
NOWT	suppresses display of weighted frequencies

You can specify the following options in a TABLES statement.

ALPHA=α

sets the level for confidence limits. The value of α must be between 0 and 1, and the default is 0.05. A confidence level of α produces $100(1-\alpha)\%$ confidence limits. The default of ALPHA=0.05 produces 95% confidence limits.

You request confidence limits for percentages with the CL option, and you request confidence limits for weighted frequencies with the CLWT option. See the section "Confidence Limits" on page 6339 for more information.

The ALPHA= option also applies to confidence limits for the risks and risk difference, which you request with the RISK option, and to confidence limits for the odds ratio and relative risks, which you request with the OR option. See the sections "Risks and Risk Difference" on page 6342 and "Odds Ratio and Relative Risks" on page 6344 for details.

CHISQ

requests the Rao-Scott chi-square test. This test applies a design effect correction to the Pearson chi-square statistic computed from the weighted frequencies. See the section "Rao-Scott Chi-Square Test" on page 6346 for details.

By default for one-way tables, the CHISQ option provides a design-based goodness-of-fit test for equal proportions. To compute the test for other null hypothesis proportions, specify the null proportions with the TESTP= option.

The CHISQ option uses proportion estimates to compute the design effect correction. To use null hypothesis proportions instead, specify the CHISQ1 option.

CHISQ1

requests the Rao-Scott modified chi-square test. This test applies a design effect correction to the Pearson chi-square statistic computed from the weighted frequencies, and bases the design

effect correction on null hypothesis proportions. See the section "Rao-Scott Chi-Square Test" on page 6346 for details. To compute the design effect correction from proportion estimates instead of null proportions, specify the CHISQ option.

By default for one-way tables, the CHISQ option provides a design-based goodness-of-fit test for equal proportions. To compute the test for other null hypothesis proportions, specify the null proportions with the TESTP= option.

CL

requests confidence limits for the percentages, or proportions, in the crosstabulation table. PROC SURVEYFREQ determines the confidence coefficient from the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits. See the section "Confidence Limits" on page 6339 for more information.

CLWT

requests confidence limits for the weighted frequencies, or totals, in the crosstabulation table. PROC SURVEYFREQ determines the confidence coefficient from the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits. See the section "Confidence Limits" on page 6339 for more information.

COL

displays the column percentage, or estimated proportion of the column total, for each cell in a two-way table. The COL option also provides the standard errors of the column percentages. See the section "Row and Column Proportions" on page 6332 for more information. This option has no effect for one-way tables.

CV

displays the coefficient of variation for each percentage, or proportion estimate, in the crosstabulation table. See the section "Coefficient of Variation" on page 6341 for more information.

CVWT

displays the coefficient of variation for each weighted frequency, or estimated total, in the crosstabulation table. See the section "Coefficient of Variation" on page 6341 for more information.

DEFF

displays the design effect for each overall proportion estimate in the crosstabulation table. See the section "Design Effect" on page 6341 for more information.

DF=*df*

specifies the degrees of freedom for the analysis. The value of *df* must be a nonnegative number. By default, PROC SURVEYFREQ computes the degrees of freedom as described in the section "Degrees of Freedom" on page 6340.

PROC SURVEYFREQ uses the DF= value in computing confidence limits for proportions, totals, and other statistics. See the section "Confidence Limits" on page 6339 for details. PROC SURVEYFREQ also uses the DF= value in computing the denominator degrees of freedom for the *F* statistics in the Rao-Scott and Wald chi-square tests. See the sections

"Rao-Scott Chi-Square Test" on page 6346, "Rao-Scott Likelihood Ratio Chi-Square Test" on page 6350, "Wald Chi-Square Test" on page 6352, and "Wald Log-Linear Chi-Square Test" on page 6354 for more information.

EXPECTED

displays expected weighted frequencies for the table cells in a two-way table. The expected frequencies are computed under the null hypothesis that the row and column variables are independent. See the section "Expected Weighted Frequency" on page 6342 for more information. This option has no effect for one-way tables.

LRCHISQ

requests the Rao-Scott likelihood ratio chi-square test. This test applies a design effect correction to the likelihood ratio chi-square statistic computed from the weighted frequencies. See the section "Rao-Scott Likelihood Ratio Chi-Square Test" on page 6350 for more information.

By default for one-way tables, the LRCHISQ option provides a design-based test for equal proportions. To compute the test for other null hypothesis proportions, specify the null proportions with the TESTP= option.

The LRCHISQ option uses proportion estimates to compute the design effect correction. To use null hypothesis proportions instead, specify the LRCHISQ1 option.

LRCHISQ1

requests the Rao-Scott modified likelihood ratio chi-square test. This test applies a design effect correction to the likelihood ratio chi-square statistic computed from the weighted frequencies, and bases the design effect correction on null hypothesis proportions. See the section "Rao-Scott Likelihood Ratio Chi-Square Test" on page 6350 for more information. To compute the design effect correction from proportion estimates instead of null proportions, specify the LRCHISQ option.

By default for one-way tables, the LRCHISQ option provides a design-based test for equal proportions. To compute the test for other null hypothesis proportions, specify the null proportions with the TESTP= option.

NOCELLPERCENT

suppresses the display of overall cell percentages in the crosstabulation table, as well as the standard errors of the percentages. The NOCELLPERCENT option does not suppress the display of row or column percentages, which you request with the ROW or COL option.

NOFREQ

suppresses the display of cell frequencies in the crosstabulation table. The NOFREQ option also suppresses the display of row, column, and overall table frequencies.

NOPERCENT

suppresses the display of all percentages in the crosstabulation table. The NOPERCENT option also suppresses the display of standard errors of the percentages. Use the NOCELLPERCENT option to suppress display of overall cell percentages but allow display of row or column percentages.

NOPRINT

suppresses the display of frequency and crosstabulation tables but displays all requested statistical tests. Note that this option disables the Output Delivery System (ODS) for the suppressed tables. For more information, see Chapter 20, "Using the Output Delivery System."

NOSPARSE

suppresses the display of variable levels with zero frequency in two-way tables. By default, the procedure displays all levels of the column variable within each level of the row variable, including any column variable levels with zero frequency for that row. For multiway tables, the procedure displays all levels of the row variable for each layer of the table by default, including any row variable levels with zero frequency for the layer.

NOSTD

suppresses the display of all standard errors in the crosstabulation table.

NOTOTAL

suppresses the display of row totals, column totals, and overall totals in the crosstabulation table.

NOWT

suppresses the display of weighted frequencies in the crosstabulation table. The NOWT option also suppresses the display of standard errors of the weighted frequencies.

OR

requests estimates of the odds ratio, the column 1 relative risk, and the column 2 relative risk for 2×2 tables. The OR option also provides confidence limits for these statistics. See the section "Odds Ratio and Relative Risks" on page 6344 for details.

To compute confidence limits for the odds ratio and relative risks, PROC SURVEYFREQ determines the confidence coefficient from the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits.

RISK

requests risk statistics for 2×2 tables. The RISK option also provides standard errors and confidence limits for these statistics. Risk statistics include the row 1 risk (or proportion), row 2 risk, overall risk, and risk difference. See the section "Risks and Risk Difference" on page 6342 for details.

The RISK option provides both column 1 and column 2 risks. To request only column 1 or column 2 risks, use the RISK1 or RISK2 option.

To compute confidence limits for the risks and risk difference, PROC SURVEYFREQ determines the confidence coefficient from the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits.

RISK1

requests column 1 risk statistics for 2×2 tables, together with their standard errors and confidence limits. Risk statistics include the row 1 risk (or proportion), row 2 risk, overall risk, and risk difference. See the section "Risks and Risk Difference" on page 6342 for details.

To compute confidence limits for the risks and risk difference, PROC SURVEYFREQ determines the confidence coefficient from the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits.

RISK2

requests column 2 risk statistics for 2×2 tables, together with their standard errors and confidence limits. Risk statistics include the row 1 risk (or proportion), row 2 risk, overall risk, and risk difference. See the section "Risks and Risk Difference" on page 6342 for details.

To compute confidence limits for the risks and risk difference, PROC SURVEYFREQ determines the confidence coefficient from the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits.

ROW

displays the row percentage, or estimated proportion of the row total, for each cell in a two-way table. The ROW option also provides the standard errors of the row percentages. See the section "Row and Column Proportions" on page 6332 for more information. This option has no effect for one-way tables.

TESTP=(*values*)

specifies null hypothesis proportions, or test percentages, for one-way chi-square tests. You can separate *values* with blanks or commas. Specify *values* in probability form as numbers between 0 and 1, where the proportions sum to 1. Or specify *values* in percentage form as numbers between 0 and 100, where the percentages sum to 100. PROC SURVEYFREQ treats the value 1 as the percentage form 1%. The number of TESTP= values must equal the number of variable levels in the one-way table. List these values in the same order in which the corresponding variable levels appear in the output.

When you specify the TESTP= option, PROC SURVEYFREQ displays the specified test percentages in the one-way frequency table. The TESTP= option has no effect for two-way tables.

PROC SURVEYFREQ uses the TESTP= values for all one-way chi-square tests you request in the TABLES statement. The available one-way chi-square tests include the Rao-Scott (Pearson) chi-square test and the Rao-Scott likelihood ratio chi-square test and their modified versions, which you request with the CHISQ, CHISQ1, LRCHISQ, and LRCHISQ1 options. See the sections "Rao-Scott Chi-Square Test" on page 6346 and "Rao-Scott Likelihood Ratio Chi-Square Test" on page 6350 for details.

VAR

displays the variance estimate for each percentage in the crosstabulation table. See the section "Proportions" on page 6331 for details. By default, PROC SURVEYFREQ displays the standard errors of the percentages.

VARWT

displays the variance estimate for each weighted frequency, or estimated total, in the crosstabulation table. See the section "Totals" on page 6329 for details. By default, PROC SURVEYFREQ displays the standard deviations of the weighted frequencies.

WCHISQ
requests the Wald chi-square test for two-way tables. See the section "Wald Chi-Square Test" on page 6352 for details.

WLLCHISQ
requests the Wald log-linear chi-square test for two-way tables. See the section "Wald Log-Linear Chi-Square Test" on page 6354 for details.

WTFREQ
displays totals (weighted frequencies) and their standard errors when you do not specify a WEIGHT or REPWEIGHTS statement. PROC SURVEYFREQ displays the weighted frequencies by default when you include a WEIGHT or REPWEIGHTS statement. Without a WEIGHT or REPWEIGHTS statement, PROC SURVEYFREQ assigns all observations a weight of one.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement names the variable that contains the sampling weights. This variable must be numeric, and the sampling weights must be positive numbers. If an observation has a weight that is nonpositive or missing, then the procedure omits that observation from the analysis. See the section "Missing Values" on page 6323 for more information. If you specify more than one WEIGHT statement, the procedure uses only the first WEIGHT statement and ignores the rest.

If you do not specify a WEIGHT statement but provide replicate weights with a REPWEIGHTS statement, PROC SURVEYFREQ uses the average of each observation's replicate weights as the observation's weight.

If you do not specify a WEIGHT statement or a REPWEIGHTS statement, PROC SURVEYFREQ assigns all observations a weight of one.

Details: SURVEYFREQ Procedure

Specifying the Sample Design

PROC SURVEYFREQ produces tables and statistics based on the sample design used to obtain the survey data. PROC SURVEYFREQ can be used for single-stage or multistage designs, with or without stratification, and with or without unequal weighting. To analyze your survey data with PROC SURVEYFREQ, you need to provide sample design information for the procedure. This information can include design strata, clusters, and sampling weights. You provide sample de-

sign information with the STRATA, CLUSTER, and WEIGHT statements, and with the RATE= or TOTAL= option in the PROC SURVEYFREQ statement.

If you provide replicate weights for BRR or jackknife variance estimation, you do not need to specify a STRATA or CLUSTER statement. Otherwise, you should specify STRATA and CLUSTER statements whenever your design includes stratification and clustering.

When there are clusters, or PSUs, in the sample design, the procedure estimates variance by using the PSUs, as described in the section "Statistical Computations" on page 6326. For a multistage sample design, the variance estimation depends only on the first stage of the sample design. So, the required input includes only first-stage cluster (PSU) and first-stage stratum identification. You do not need to input design information about any additional stages of sampling.

Stratification

If your sample design is stratified at the first stage of sampling, use the STRATA statement to name the variables that form the strata. The combinations of categories of STRATA variables define the strata in the sample, where strata are nonoverlapping subgroups that were sampled independently. If your sample design has stratification at multiple stages, you should identify only the first-stage strata in the STRATA statement.

If you use a REPWEIGHTS statement to provide replicate weights for BRR or jackknife variance estimation, you do not need to specify a STRATA statement. Otherwise, you should specify a STRATA statement whenever your design includes stratification. If you do not specify a STRATA statement or a REPWEIGHTS statement, then PROC SURVEYFREQ assumes there is no stratification at the first stage.

Clustering

If your sample design selects clusters at the first stage of sampling, use the CLUSTER statement to name the variables that identify the first-stage clusters, or primary sampling units (PSUs). The combinations of categories of CLUSTER variables define the clusters in the sample. If there is a STRATA statement, clusters are nested within strata. If your sample design has clustering at multiple stages, you should specify only the first-stage clusters, or PSUs, in the CLUSTER statement. PROC SURVEYFREQ assumes that each cluster defined by the CLUSTER statement variables represents a PSU in the sample, and that each observation belongs to one PSU.

If you use a REPWEIGHTS statement to provide replicate weights for BRR or jackknife variance estimation, you do not need to specify a CLUSTER statement. Otherwise, you should specify a CLUSTER statement whenever your design includes clustering at the first stage of sampling. If you do not specify a CLUSTER statement, then PROC SURVEYFREQ treats each observation as a PSU.

Weighting

If your sample design includes unequal weighting, use the WEIGHT statement to name the variable that contains the sampling weights. Sampling weights must be positive numbers. If an observation

has a weight that is nonpositive or missing, then the procedure omits that observation from the analysis. See the section "Missing Values" on page 6323 for more information.

If you do not specify a WEIGHT statement but include a REPWEIGHTS statement, PROC SURVEYFREQ uses the average of each observation's replicate weights as the observation's weight. If you do not specify a WEIGHT statement or a REPWEIGHTS statement, PROC SURVEYFREQ assigns all observations a weight of one.

Population Totals and Sampling Rates

To include a finite population correction (*fpc*) in Taylor series variance estimation, you can input either the sampling rate or the population total by using the RATE= or TOTAL= option in the PROC SURVEYFREQ statement. (You cannot specify both of these options in the same PROC SURVEYFREQ statement.) The RATE= and TOTAL= options apply only to Taylor series variance estimation. The procedure does not use a finite population correction for BRR or jackknife variance estimation.

If you do not specify the RATE= or TOTAL= option, the Taylor series variance estimation does not include a finite population correction. For fairly small sampling fractions, it is appropriate to ignore this correction. See Cochran (1977) and Kish (1965) for more information.

If your design has multiple stages of selection and you are specifying the RATE= option, you should input the first-stage sampling rate, which is the ratio of the number of PSUs in the sample to the total number of PSUs in the study population. If you are specifying the TOTAL= option for a multistage design, you should input the total number of PSUs in the study population.

For a nonstratified sample design, or for a stratified sample design with the same sampling rate or the same population total in all strata, you can use the RATE=*value* or TOTAL=*value* option. If your sample design is stratified with different sampling rates or population totals in different strata, use the RATE=*SAS-data-set* or TOTAL=*SAS-data-set* option to name a SAS data set that contains the stratum sampling rates or totals. This data set is called a *secondary data set*, as opposed to the *primary data set* that you specify with the DATA= option.

The secondary data set must contain all the stratification variables listed in the STRATA statement and all the variables in the BY statement. Furthermore, the BY groups must appear in the same order as in the primary data set. If there are formats associated with the STRATA variables and the BY variables, then the formats must be consistent in the primary and the secondary data sets. If you specify the TOTAL=*SAS-data-set* option, the secondary data set must have a variable named _TOTAL_ that contains the stratum population totals. If you specify the RATE=*SAS-data-set* option, the secondary data set must have a variable named _RATE_ that contains the stratum sampling rates. If the secondary data set contains more than one observation for any one stratum, the procedure uses the first value of _TOTAL_ or _RATE_ for that stratum and ignores the rest.

The *value* in the RATE= option or the values of _RATE_ in the secondary data set must be nonnegative numbers. You can specify *value* as a number between 0 and 1. Or you can specify *value* in percentage form as a number between 1 and 100, and PROC SURVEYFREQ converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

If you specify the TOTAL=*value* option, *value* must not be less than the sample size. If you provide stratum population totals in a secondary data set, these values must not be less than the corresponding stratum sample sizes.

Domain Analysis

PROC SURVEYFREQ provides domain analysis through its multiway table capability. *Domain analysis* refers to the computation of statistics for subpopulations, or domains, in addition to the computation of statistics for the entire study population. Formation of subpopulations can be unrelated to the sample design, and so the domain sample sizes can actually be random variables. Domain analysis takes this variability into account by using the entire sample in estimating the variance of domain estimates. Domain analysis is also known as subgroup analysis, subpopulation analysis, or subdomain analysis. For more information about domain analysis, see Lohr (1999), Cochran (1977), and Fuller et al. (1989).

To request domain analysis with PROC SURVEYFREQ, you should include the domain variable(s) in your TABLES statement request. For example, specifying DOMAIN * A * B in a TABLES statement produces separate two-way tables of A by B for each level of DOMAIN. If your domains are formed by more than one variable, you can specify DomainVariable_1 * DomainVariable_2 * A * B, for example, to obtain two-way tables of A by B for each domain formed by the different combinations of levels for DomainVariable_1 and DomainVariable_2. See Example 83.2 for an example of domain analysis.

If you specify DOMAIN * A in a TABLES statement, the values of the variable DOMAIN form the table rows. The two-way table lists levels of the variable A within each level of the row variable DOMAIN. Specify the ROW option in the TABLES statement to obtain the row percentages and their standard errors. This provides the one-way distribution of A for each domain, or level of the variable DOMAIN.

Including the domain variables in a TABLES statement request gives a different analysis from that obtained by using a BY statement, which provides completely separate analyses of the BY groups. The BY statement can also be used to analyze the data set by subgroups, but it is critical to note that this will *not* produce a valid domain analysis. The BY statement is appropriate only when the number of units in each subgroup is known with certainty; when the subgroup sample size is a random variable, include the domain variables in your TABLES statement request.

Missing Values

WEIGHT Variable

If an observation has a missing value or a nonpositive value for the WEIGHT variable, then PROC SURVEYFREQ excludes that observation from the analysis.

REPWEIGHTS Variables

If you provide replicate weights with a REPWEIGHTS statement for BRR or jackknife variance estimation, all REPWEIGHTS variable values must be nonmissing. Similarly, if you provide jackknife coefficients with the JKCOEFS= option in the REPWEIGHTS statement, all values of the

JKCoefficient variable must be nonmissing. The procedure does not perform the analysis when any replicate weight or jackknife coefficient value is missing.

STRATA and CLUSTER Variables

An observation is excluded from the analysis if it has a missing value for any STRATA or CLUSTER variable, unless you specify the MISSING option in the PROC SURVEYFREQ statement. If you specify the MISSING option, the procedure treats missing values as a valid (nonmissing) category for all categorical variables, which include STRATA variables, CLUSTER variables, and TABLES variables.

TABLES Variables

By default, PROC SURVEYFREQ excludes an observation from a crosstabulation table (and all associated analyses) if the observation has a missing value for any of the variables in the TABLES request, unless you specify the MISSING or NOMCAR option in the PROC SURVEYFREQ statement. When the procedure excludes observations with missing values from a table, it displays the total frequency of missing observations below the table.

If you specify the MISSING option, the procedure treats missing values as a valid (nonmissing) level for each TABLES variable. These levels are displayed in the crosstabulation table and included in computations of totals, percentages, and all other table statistics.

If you specify the NOMCAR option, which is available for Taylor series variance estimation, the procedure includes observations with missing values of TABLES variables in the variance computations. The NOMCAR option does not display missing levels in the crosstabulation table or compute percentages and totals for missing levels.

The NOMCAR Option

The NOMCAR option includes observations with missing values of TABLES variables in the variance computations as *not missing completely at random* (NOMCAR) for Taylor series variance estimation. By default, observations are completely excluded from the analysis if they have missing values for any of the variables in the current TABLES request. This default treatment is based on the assumption that the values are *missing completely at random* (MCAR), and assumes that the analysis results should not be substantially different between the missing and nonmissing groups. See the section "Analysis Considerations" on page 6326 for more information.

When you specify the NOMCAR option, PROC SURVEYFREQ computes variance estimates by analyzing the nonmissing values as a domain or subpopulation, where the entire population includes both nonmissing and missing domains.

Note that the NOMCAR option has no effect when you specify the MISSING option, which treats missing values as a valid nonmissing level. The NOMCAR option does not affect the inclusion of observations with missing values of the WEIGHT, CLUSTER, or STRATA variables. Observations

with missing values of the WEIGHT variable are always excluded from the analysis. Observations with missing values of the CLUSTER or STRATA variables are excluded unless you specify the MISSING option.

The NOMCAR option applies only to Taylor series variance estimation. The replication methods, which you request with the VARMETHOD=BRR and VARMETHOD=JACKKNIFE options, do not use the NOMCAR option.

Degrees of Freedom

PROC SURVEYFREQ computes degrees of freedom to obtain the *t*-percentile for confidence limits for proportions, totals, and other statistics. The procedure also uses degrees of freedom for the *F* statistics in the Rao-Scott and Wald chi-square tests. The degrees of freedom computation depends on the variance estimation method that you request. See the section "Degrees of Freedom" on page 6340 for details. Missing values can affect the degrees of freedom computation.

Taylor Series Variance Estimation

The degrees of freedom can depend on the number of clusters, the number of strata, and the number of observations. For Taylor series variance estimation, these numbers are based on the observations included in the analysis of the individual table. These numbers do not count observations excluded from the table due to missing values. If all values in a stratum are excluded from the analysis of a table as missing values, then that stratum is called an *empty stratum*. Empty strata are not counted in the total number of strata for the table. Similarly, empty clusters and missing observations are not included in the totals counts of cluster and observations use to compute the degrees of freedom for the analysis.

If you specify the MISSING option, missing values are treated as valid nonmissing levels and are included in computing degrees of freedom. If you specify the NOMCAR option for Taylor series variance estimation, observations with missing values of the TABLES variables are included in computing degrees of freedom.

Replicate-Based Variance Estimation

For BRR or jackknife variance estimation, by default PROC SURVEYFREQ computes the degrees of freedom by using all valid observations in the input data set. A valid observation is an observation that has a positive value of the WEIGHT variable and nonmissing values of the STRATA and CLUSTER variables unless you specify the MISSING option. See the section "Data Summary Table" on page 6356 for details about valid observations.

If you specify the DFADJ method-option for VARMETHOD=BRR or VARMETHOD=JACKKNIFE, the procedure computes the degrees of freedom based on the nonmissing observations included in the individual table analysis. This excludes any empty strata or clusters that occur when observations with missing values of the TABLES variables are removed from the analysis for that table.

Table Summary Output Data Set

For each table request, PROC SURVEYFREQ produces a nondisplayed ODS table, "Table Summary," which contains the number of (nonmissing) observations, strata, and clusters that are included in the analysis of the individual table. If there are missing observations, empty strata, or empty clusters excluded from the analysis, the "Table Summary" data set also contains this information. If you request any confidence limits or chi-square tests for the table, which require degrees of freedom, the "Table Summary" data set provides the degrees of freedom.

Due to missing values, the number of observations used for an individual table analysis can differ from the number of valid observations in the input data set, which is reported in the "Data Summary" table. Similarly, a difference can also occur for the number of clusters or strata. See Example 83.3 for more information about the "Table Summary" output data set.

If you specify the NOMCAR option for Taylor series variance estimation, the "Table Summary" data set reflects all observations used for variance estimation, which includes those observations with missing values of the TABLES variables.

Analysis Considerations

If you have missing values in your survey data for any reason (such as nonresponse), this can compromise the quality of your survey results. An observation without missing values is called a *complete respondent*, and an observation with missing values is called an *incomplete respondent*. If the complete respondents are different from the incomplete respondents with regard to a survey effect or outcome, then survey estimates will be biased and will not accurately represent the survey population. There are a variety of techniques in sample design and survey operations that can reduce nonresponse. After data collection is complete, you can use imputation to replace missing values with acceptable values, and you can use sampling weight adjustments to compensate for nonresponse. You should complete this data preparation and adjustment before you analyze your data with PROC SURVEYFREQ. See Cochran (1977), Kalton and Kaspyzyk (1986), and Brick and Kalton (1996) for more details.

Statistical Computations

Variance Estimation

PROC SURVEYFREQ provides a choice of variance estimation methods for complex survey data. In addition to the Taylor series linearization method, the procedures offer two replication-based or resampling methods—balanced repeated replication (BRR) and the delete-1 jackknife. These variance estimation methods usually give similar, satisfactory results (Lohr 1999; Särndal, Swensson, and Wretman 1992; Wolter 1985). The choice of a variance estimation method can depend on the sample design used, the sample design information available, the parameters to be estimated, and computational issues. See Lohr (1999) for more details.

Taylor Series Variance Estimation

The Taylor series linearization method can be used to estimate standard errors of proportions and other statistics for crosstabulation tables. For sample survey data, the proportion estimator is a ratio estimator formed from estimators of totals. For example, to estimate the proportion in a crosstabulation table cell, the procedure uses the ratio of the estimator of the cell total frequency to the estimator of the overall population total, where these totals are linear statistics computed from the survey data. The Taylor series expansion method obtains a first-order linear approximation for the ratio estimator and then uses the variance estimate for this approximation to estimate the variance of the estimate itself (Woodruff 1971; Fuller 1975). For more information about Taylor series variance estimation for sample survey data, see Lohr (1999), Särndal, Swensson, and Wretman (1992), Lee, Forthoffer, and Lorimor (1989), and Wolter (1985).

When there are clusters, or PSUs, in the sample design, the Taylor series method estimates variance from the variance among PSUs. When the design is stratified, the procedure combines stratum variance estimates to compute the overall variance estimate. For a multistage sample design, the variance estimation depends only on the first stage of the sample design. So the required input includes only first-stage cluster (PSU) and first-stage stratum identification. You do not need to input design information about any additional stages of sampling. This variance estimation method assumes that the first-stage sampling fraction is small, or the first-stage sample is drawn with replacement, as it often is in practice.

See the sections "Proportions" on page 6331, "Row and Column Proportions" on page 6332, "Risks and Risk Difference" on page 6342, and "Odds Ratio and Relative Risks" on page 6344 for details and formulas for Taylor series variance estimates.

Replication-Based Variance Estimation

Replication-based methods for variance estimation draw multiple replicates (or subsamples) from the full sample by following a specific resampling scheme. Commonly used resampling schemes include *balanced repeated replication* (BRR) and the *jackknife*. PROC SURVEYFREQ estimates the parameter of interest (a proportion, total, odds ratio, or other statistic) from each replicate, and then uses the variability among replicate estimates to estimate the overall variance of the parameter estimate. See Wolter (1985) and Lohr (1999) for more information.

The BRR variance estimation method requires a stratified sample design with two PSUs in each stratum. Each replicate is obtained by deleting one PSU per stratum according to the corresponding Hadamard matrix and adjusting the original weights of the remaining PSUs. The adjusted weights are called *replicate weights*. PROC SURVEYFREQ also provides Fay's method, which is a modification of the BRR method. See the section "Balanced Repeated Replication (BRR)" on page 6334 for details.

The jackknife method deletes one PSU at a time from the full sample to create replicates, and modifies the original weights to obtain replicate weights. The total number of replicates equals the number of PSUs. If the sample design is stratified, each stratum must contain at least two PSUs, and the jackknife is applied separately within each stratum. See the section "The Jackknife" on page 6337 for details.

Instead of having PROC SURVEYFREQ generate replicate weights for the analysis, you can input your own replicate weights with a REPWEIGHTS statement. This can be useful if you need to do multiple analyses with the same set of replicate weights, or if you have access to replicate weights instead of design information. See the section "Replicate Weights Output Data Set" on page 6355 for more information.

Definitions and Notation

For a stratified clustered sample design, define the following:

h	$=$	$1, 2, \ldots, H$	is the stratum number, with a total of H strata
i	$=$	$1, 2, \ldots, n_h$	is the cluster number within stratum h, with a total of n_h sample clusters in stratum h
j	$=$	$1, 2, \ldots, m_{hi}$	is the unit number within cluster i of stratum h, with a total of m_{hi} sample units from cluster i of stratum h
n	$=$	$\sum_{h=1}^{H} \sum_{i=1}^{n_h} m_{hi}$	is the total number of observations in the sample

f_h	$=$	first-stage sampling rate for stratum h
W_{hij}	$=$	sampling weight of unit j in cluster i of stratum h

The sampling rate f_h, which is used in Taylor series variance estimation, is the fraction of first-stage units (PSUs) selected for the sample. You can specify the stratum sampling rates with the RATE= option. Or if you specify population totals with the TOTAL= option, PROC SURVEYFREQ computes f_h as the ratio of stratum sample size to the stratum total, in terms of PSUs. See the section "Population Totals and Sampling Rates" on page 6322 for details. If you do not specify the RATE= option or the TOTAL= option, then the procedure assumes that the stratum sampling rates f_h are negligible and does not use a finite population correction when computing variances.

This notation is also applicable to other sample designs. For example, for a design without stratification, you can let $H = 1$; for a sample design without clustering, you can let $m_{hi} = 1$ for every h and i, which replaces clusters with individual sampling units.

For a two-way table representing the crosstabulation of two variables, define the following, where there are R levels of the row variable and C levels of the column variable:

r	$=$	$1, 2, \ldots, R$	is the row number, with a total of R rows
c	$=$	$1, 2, \ldots, C$	is the column number, with a total of C columns
N_{rc}			is the population total in row r and column c
$N_{r\cdot}$	$=$	$\sum_{c=1}^{C} N_{rc}$	is the total in row r
$N_{\cdot c}$	$=$	$\sum_{r=1}^{R} N_{rc}$	is the total in column c
N	$=$	$\sum_{r=1}^{R} \sum_{c=1}^{C} N_{rc}$	is the overall total

$$
\begin{aligned}
P_{rc} &= N_{rc} \,/\, N && \text{is the population proportion in row } r \text{ and column } c \\
P_{r\cdot} &= N_{r\cdot} \,/\, N && \text{is the proportion in row } r \\
P_{\cdot c} &= N_{\cdot c} \,/\, N && \text{is the proportion in column } c \\
P^{r}_{rc} &= N_{rc} \,/\, N_{r\cdot} && \text{is the row proportion for table cell } (r,c) \\
P^{c}_{rc} &= N_{rc} \,/\, N_{\cdot c} && \text{is the column proportion for table cell } (r,c)
\end{aligned}
$$

For a specified observation (identified by stratum, cluster, and unit number within the cluster), define the following to indicate whether or not that observation belongs to cell (r, c), row r and column c, of the two-way table, for $r = 1, 2, \ldots, R$ and $c = 1, 2, \ldots, C$:

$$
\delta_{hij}(r,c) = \begin{cases} 1 & \text{if observation } (hij) \text{ is in cell } (r,c) \\ 0 & \text{otherwise} \end{cases}
$$

Similarly, define the following functions to indicate the observation's row and column classification:

$$
\delta_{hij}(r\,\cdot) = \begin{cases} 1 & \text{if observation } (hij) \text{ is in row } r \\ 0 & \text{otherwise} \end{cases}
$$

$$
\delta_{hij}(\cdot\, c) = \begin{cases} 1 & \text{if observation } (hij) \text{ is in column } c \\ 0 & \text{otherwise} \end{cases}
$$

Totals

PROC SURVEYFREQ estimates population frequency totals for the specified crosstabulation tables, including totals for two-way table cells, rows, columns, and overall totals. The procedure computes the estimate of the total frequency in table cell (r, c) as the weighted frequency sum,

$$
\widehat{N}_{rc} = \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \delta_{hij}(r,c)\; W_{hij}
$$

Similarly, PROC SURVEYFREQ computes estimates of row totals, column totals, and overall totals as

$$
\widehat{N}_{r\cdot} = \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \delta_{hij}(r\,\cdot)\; W_{hij}
$$

$$
\widehat{N}_{\cdot c} = \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \delta_{hij}(\cdot\, c)\; W_{hij}
$$

$$
\widehat{N} = \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} W_{hij}
$$

PROC SURVEYFREQ estimates the variances of totals by using the variance estimation method that you request. If you request BRR variance estimation (by specifying the VARMETHOD=BRR

option in the PROC SURVEYFREQ statement), the procedure estimates the variances as described in the section "Balanced Repeated Replication (BRR)" on page 6334. If you request jackknife variance estimation (by specifying the VARMETHOD=JACKKNIFE option), the procedure estimates the variances as described in the section "The Jackknife" on page 6337.

If you do not specify the VARMETHOD= option or a REPWEIGHTS statement, the default variance estimation method is Taylor series, which you can also request with the VARMETHOD=TAYLOR option. Since totals are linear statistics, their variances can be estimated directly, without the approximation that is used for proportions and other nonlinear statistics. PROC SURVEYFREQ estimates the variance of the total frequency in table cell (r, c) as

$$\widehat{\mathrm{Var}}(\widehat{N}_{rc}) = \sum_{h=1}^{H} \widehat{\mathrm{Var}}_h(\widehat{N}_{rc})$$

where if $n_h > 1$,

$$\begin{aligned}
\widehat{\mathrm{Var}}_h(\widehat{N}_{rc}) &= \frac{n_h(1-f_h)}{n_h - 1} \sum_{i=1}^{n_h} (n_{rc}^{hi} - \bar{n}_{rc}^{h})^2 \\
n_{rc}^{hi} &= \sum_{j=1}^{m_{hi}} \delta_{hij}(r,c)\, W_{hij} \\
\bar{n}_{rc}^{h} &= \sum_{i=1}^{n_h} n_{rc}^{hi} \,/\, n_h
\end{aligned}$$

and if $n_h = 1$,

$$\widehat{\mathrm{Var}}_h(\widehat{N}_{rc}) = \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 \le h' \le H \end{cases}$$

The standard deviation of the total is computed as

$$\mathrm{Std}(\widehat{N}_{rc}) = \sqrt{\widehat{\mathrm{Var}}(\widehat{N}_{rc})}$$

The variances and standard deviations are computed in a similar manner for row totals, column totals, and overall table totals.

Covariance of Totals

The covariance matrix of the table cell totals $\widehat{N}_{rc}$ is an $rc \times rc$ matrix $\widehat{\mathbf{V}}(\widehat{\mathbf{N}})$, which contains the pairwise table cell covariances $\widehat{\mathrm{Cov}}(\widehat{N}_{rc}, \widehat{N}_{ab})$, for $r = 1, \cdots, R$; $c = 1, \cdots, C$; $a = 1, \cdots, R$; and $b = 1, \cdots, C$.

PROC SURVEYFREQ estimates the covariances by using the variance estimation method that you request. If you request BRR variance estimation (by specifying the VARMETHOD=BRR option in the PROC SURVEYFREQ statement), the procedure estimates the covariances by the method described in the section "Balanced Repeated Replication (BRR)" on page 6334. If you request

jackknife variance estimation (by specifying the VARMETHOD=JACKKNIFE option), the procedure uses the method described in the section "The Jackknife" on page 6337.

Otherwise (by default, or if you request the Taylor series method), PROC SURVEYFREQ estimates the covariance between total frequency estimates for table cells (r, c) and (a, b) as

$$\widehat{\mathrm{Cov}}(\widehat{N}_{rc},\ \widehat{N}_{ab}) = \sum_{h=1}^{H} \left(\frac{n_h(1-f_h)}{n_h - 1} \sum_{i=1}^{n_h} (n_{rc}^{hi} - \bar{n}_{rc}^{h})\ (n_{ab}^{hi} - \bar{n}_{ab}^{h}) \right)$$

Proportions

PROC SURVEYFREQ computes the estimate of the proportion in table cell (r, c) as the ratio of the estimated total for the table cell to the estimated overall total,

$$\begin{aligned} \widehat{P}_{rc} &= \widehat{N}_{rc} \ / \ \widehat{N} \\ &= \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \delta_{hij}(r,c)\ W_{hij} \ / \ \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} W_{hij} \end{aligned}$$

If you request BRR variance estimation (by specifying the VARMETHOD=BRR option in the PROC SURVEYFREQ statement), the procedure estimates the variances of proportion estimates as described in the section "Balanced Repeated Replication (BRR)" on page 6334. If you request jackknife variance estimation (by specifying the VARMETHOD=JACKKNIFE option), the procedure estimates the variances as described in the section "The Jackknife" on page 6337.

If you do not specify the VARMETHOD= option or a REPWEIGHTS statement, the default variance estimation method is Taylor series, which you can also request with the VARMETHOD=TAYLOR option. By using Taylor series linearization, the variance of a proportion estimate can be expressed as

$$\widehat{\mathrm{Var}}(\widehat{P}_{rc}) = \sum_{h=1}^{H} \widehat{\mathrm{Var}}_h(\widehat{P}_{rc})$$

where if $n_h > 1$,

$$\begin{aligned} \widehat{\mathrm{Var}}_h(\widehat{P}_{rc}) &= \frac{n_h(1-f_h)}{n_h - 1} \sum_{i=1}^{n_h} (e_{rc}^{hi} - \bar{e}_{rc}^{h})^2 \\ e_{rc}^{hi} &= \left(\sum_{j=1}^{m_{hi}} (\delta_{hij}(r,c) - \widehat{P}_{rc})\ W_{hij} \right) \ / \ \widehat{N} \\ \bar{e}_{rc}^{h} &= \sum_{i=1}^{n_h} e_{rc}^{hi} \ / \ n_h \end{aligned}$$

and if $n_h = 1$,

$$\widehat{\mathrm{Var}}_h(\widehat{P}_{rc}) = \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 \le h' \le H \end{cases}$$

The standard error of the proportion is computed as

$$\text{StdErr}(\widehat{P}_{rc}) = \sqrt{\widehat{\text{Var}}(\widehat{P}_{rc})}$$

Similarly, the estimate of the proportion in row r is

$$\widehat{P}_{r\cdot} = \widehat{N}_{r\cdot} \,/\, \widehat{N}$$

And its variance estimate is

$$\widehat{\text{Var}}(\widehat{P}_{r\cdot}) = \sum_{h=1}^{H} \widehat{\text{Var}}_h(\widehat{P}_{r\cdot})$$

where if $n_h > 1$,

$$\begin{aligned}
\widehat{\text{Var}}_h(\widehat{P}_{r\cdot}) &= \frac{n_h(1-f_h)}{n_h - 1} \sum_{i=1}^{n_h} (e_{r\cdot}^{hi} - \bar{e}_{r\cdot}^{h})^2 \\
e_{r\cdot}^{hi} &= \left(\sum_{j=1}^{m_{hi}} (\delta_{hij}(r\,\cdot) - \widehat{P}_{r\cdot})\, W_{hij} \right) \,/\, \widehat{N} \\
\bar{e}_{r\cdot}^{h} &= \sum_{i=1}^{n_h} e_{r\cdot}^{hi} \,/\, n_h
\end{aligned}$$

and if $n_h = 1$,

$$\widehat{\text{Var}}_h(\widehat{P}_{r\cdot}) = \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 \le h' \le H \end{cases}$$

The standard error of the proportion in row r is computed as

$$\text{StdErr}(\widehat{P}_{r\cdot}) = \sqrt{\widehat{\text{Var}}(\widehat{P}_{r\cdot})}$$

Computations for the proportion in column c are done in the same way.

Row and Column Proportions

PROC SURVEYFREQ computes the estimate of the row proportion for table cell (r, c) as the ratio of the estimated total for the table cell to the estimated total for row r,

$$\begin{aligned}
\widehat{P}_{rc}^{\,r} &= \widehat{N}_{rc} \,/\, \widehat{N}_{r\cdot} \\
&= \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \delta_{hij}(r,c)\, W_{hij} \;/\; \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} \delta_{hij}(r\,\cdot)\, W_{hij}
\end{aligned}$$

Similarly, PROC SURVEYFREQ estimates the column proportion for table cell (r, c) as the ratio of the estimated total for the table cell to the estimated total for column c,

$$\begin{aligned}\widehat{P}_{rc}^{\,c} &= \widehat{N}_{rc} \,/\, \widehat{N}_{\cdot c} \\ &= \sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} \delta_{hij}(r,c)\, W_{hij} \;/\; \sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} \delta_{hij}(\cdot\, c)\, W_{hij}\end{aligned}$$

If you request BRR variance estimation (VARMETHOD=BRR), PROC SURVEYFREQ estimates the variances of the row and column proportions as described in the section "Balanced Repeated Replication (BRR)" on page 6334. If you request jackknife variance estimation (VARMETHOD=JACKKNIFE), the procedure estimates the variances as described in the section "The Jackknife" on page 6337.

If you do not specify the VARMETHOD= option or a REPWEIGHTS statement, the default variance estimation method is Taylor series (VARMETHOD=TAYLOR). By using Taylor series linearization, the variance of the row proportion estimate can be expressed as

$$\widehat{\mathrm{Var}}(\widehat{P}_{rc}^{\,r}) = \sum_{h=1}^{H} \widehat{\mathrm{Var}}_h(\widehat{P}_{rc})$$

where if $n_h > 1$,

$$\begin{aligned}\widehat{\mathrm{Var}}_h(\widehat{P}_{rc}^{\,r}) &= \frac{n_h(1-f_h)}{n_h-1}\sum_{i=1}^{n_h}(g_{rc}^{hi} - \bar{g}_{rc}^{\,h})^2 \\ g_{rc}^{hi} &= \left(\sum_{j=1}^{m_{hi}}(\delta_{hij}(r,c) - \widehat{P}_{rc}^{\,r}\,\delta_{hij}(r\,\cdot))\, W_{hij}\right) \,/\, \widehat{N}_{r\cdot} \\ \bar{g}_{rc}^{\,h} &= \sum_{i=1}^{n_h} g_{rc}^{hi} \,/\, n_h\end{aligned}$$

and if $n_h = 1$,

$$\widehat{\mathrm{Var}}_h(\widehat{P}_{rc}^{\,r}) = \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 \le h' \le H \end{cases}$$

The standard error of the row proportion is computed as

$$\mathrm{StdErr}(\widehat{P}_{rc}^{\,r}) = \sqrt{\widehat{\mathrm{Var}}(\widehat{P}_{rc}^{\,r})}$$

The Taylor series variance estimate for the column proportion is computed as described previously for the row proportion, but with

$$g_{rc}^{hi} = \left(\sum_{j=1}^{m_{hi}}(\delta_{hij}(r,c) - \widehat{P}_{rc}^{\,c}\,\delta_{hij}(\cdot\, c))\, W_{hij}\right) \,/\, \widehat{N}_{\cdot c}$$

Balanced Repeated Replication (BRR)

If you specify the VARMETHOD=BRR option, then PROC SURVEYFREQ uses balanced repeated replication (BRR) for variance estimation. The BRR variance estimation method requires a stratified sample design with two PSUs in each stratum. You can provide replicate weights for BRR variance estimation by using a REPWEIGHTS statement, or the procedure can construct replicate weights for the analysis. PROC SURVEYFREQ estimates the parameter of interest (a proportion, total, odds ratio, or other statistic) from each replicate, and then uses the variability among replicate estimates to estimate the overall variance of the parameter estimate. See Wolter (1985) and Lohr (1999) for more information about BRR variance estimation.

If you do not provide replicate weights with a REPWEIGHTS statement, PROC SURVEYFREQ constructs replicates based on the stratified design with two PSUs in each stratum. This section describes replicate construction by the traditional BRR method. If you specify the FAY method-option for VARMETHOD=BRR, the procedure uses Fay's modified BRR method, which is described in the section "Fay's BRR Method" on page 6335.

With the traditional BRR method, each replicate is obtained by deleting one PSU per stratum according to the corresponding Hadamard matrix of dimension R, where R is the number of replicates. The number of replicates equals the smallest multiple of 4 that is greater than the number of strata H. Alternatively, you can specify the number of replicates with the REPS= method-option. If a Hadamard matrix cannot be constructed for the REPS= value that you specify, the value is increased until a Hadamard matrix of that dimension can be constructed. Therefore, it is possible for the actual number of replicates used to be larger than the REPS= value that you specify.

You can provide a Hadamard matrix for BRR replicate construction by using the HADAMARD= method-option. Otherwise, PROC SURVEYFREQ generates an appropriate Hadamard matrix. See the section "Hadamard Matrix" on page 6336 for more information. You can display the Hadamard matrix by specifying the PRINTH method-option.

PROC SURVEYFREQ constructs replicates by using the first H columns of the $R \times R$ Hadamard matrix, where H denotes the number of strata. The rth replicate ($r = 1, 2, \ldots, R$) is drawn from the full sample according to the rth row of the Hadamard matrix as follows:

- If the (r, h)th element of the Hadamard matrix equals 1, then the first PSU of stratum h is included in the rth replicate, and the second PSU of stratum h is excluded.
- If the (r, h)th element of the Hadamard matrix equals -1, then the second PSU of stratum h is included in the rth replicate, and the first PSU of stratum h is excluded.

For the PSUs included in replicate r, the original weights are doubled to form the replicate r weights. For the PSUs not included in replicate r, the replicate r weights equal zero. You can use the OUTWEIGHTS= method-option to store the replicate weights in a SAS data set. See the section "Replicate Weights Output Data Set" on page 6355 for details about the contents of the OUTWEIGHTS= data set. You can provide these replicate weights to the procedure for subsequent analyses by using a REPWEIGHTS statement.

Let θ denote the population parameter to be estimated—for example, a proportion, total, odds ratio, or other statistic. Let $\hat{\theta}$ denote the estimate of θ from the full sample, and let $\hat{\theta}_r$ denote the estimate

from the rth BRR replicate, which is computed by using the replicate weights. The BRR variance estimate for $\hat{\theta}$ is computed as

$$\widehat{V}(\hat{\theta}) = \frac{1}{R}\sum_{r=1}^{R}\left(\hat{\theta}_r - \hat{\theta}\right)^2$$

where R is the total number of replicates.

If a parameter cannot be estimated from some replicate(s), then the variance estimate is computed by using those replicates from which the parameter can be estimated. For example, suppose the parameter is a column proportion—the proportion of column j for table cell (i, j). If a replicate r contains no observations in column j, then the column j proportion is not estimable from replicate r. In this case, the BRR variance estimate is computed as

$$\widehat{V}(\hat{\theta}) = \frac{1}{R'}\sum_{r=1}^{R'}\left(\hat{\theta}_r - \hat{\theta}\right)^2$$

where the summation is over the replicates where the parameter θ is estimable, and R' is the number of those replicates.

Fay's BRR Method

If you specify the FAY method-option for VARMETHOD=BRR, then PROC SURVEYFREQ uses Fay's BRR method, which is a modification of the traditional BRR variance estimation method. As for traditional BRR, Fay's method requires a stratified sample design with two PSUs in each stratum. You can provide replicate weights by using a REPWEIGHTS statement, or the procedure can construct replicate weights for the analysis. PROC SURVEYFREQ estimates the parameter of interest (a proportion, total, odds ratio, or other statistic) from each replicate, and then uses the variability among replicate estimates to estimate the overall variance of the parameter estimate.

If you do not provide replicate weights with a REPWEIGHTS statement, PROC SURVEYFREQ constructs replicates based on the stratified design with two PSUs in each stratum. As for traditional BRR, the number of replicates R equals the smallest multiple of 4 that is greater than the number of strata H, or you can specify the number of replicates with the REPS= method-option. You can provide a Hadamard matrix for replicate construction by using the HADAMARD= method-option, or PROC SURVEYFREQ generates an appropriate Hadamard matrix.

The traditional BRR method constructs half-sample replicates by deleting one PSU per stratum according to the Hadamard matrix and doubling the original weights to form replicate weights. Fay's BRR method adjusts the original weights by a coefficient ϵ, where $0 \leq \epsilon < 1$. You can specify the value of ϵ with the FAY= method-option. If you do not specify the value of ϵ, PROC SURVEYFREQ uses $\epsilon = 0.5$ by default. See Judkins (1990) and Rao and Shao (1999) for information about the value of the Fay coefficient. When $\epsilon = 0$, Fay's method becomes the traditional BRR method. See Dippo, Fay, and Morganstein (1984), Fay (1989), and Judkins (1990) for more information.

PROC SURVEYFREQ constructs Fay BRR replicates by using the first H columns of the $R \times R$ Hadamard matrix, where H denotes the number of strata. The rth replicate ($r = 1, 2, \ldots, R$) is drawn from the full sample according to the rth row of the Hadamard matrix as follows:

- If the (r, h)th element of the Hadamard matrix equals 1, the sampling weight of the first PSU in stratum h is multiplied by ϵ, and the sampling weight of the second PSU is multiplied by $(2 - \epsilon)$ to form the rth replicate weights.
- If the (r, h)th element of the Hadamard matrix equals -1, then the sampling weight of the second PSU in stratum h is multiplied by ϵ, and the sampling weight of the first PSU is multiplied by $(2 - \epsilon)$ to form the rth replicate weights.

You can use the OUTWEIGHTS= method-option to store the replicate weights in a SAS data set. See the section "Replicate Weights Output Data Set" on page 6355 for details about the contents of the OUTWEIGHTS= data set. You can provide these replicate weights to the procedure for subsequent analyses by using a REPWEIGHTS statement.

Let θ denote the population parameter to be estimated—for example, a proportion, total, odds ratio, or other statistic. Let $\hat{\theta}$ denote the estimate of θ from the full sample, and let $\hat{\theta}_r$ denote the estimate from the rth BRR replicate, which is computed by using the replicate weights. The Fay BRR variance estimate for $\hat{\theta}$ is computed as

$$\widehat{V}(\hat{\theta}) = \frac{1}{R(1-\epsilon)^2} \sum_{r=1}^{R} \left(\hat{\theta}_r - \hat{\theta}\right)^2$$

where R is the total number of replicates and ϵ is the Fay coefficient.

If you request Fay's BRR method and also include a REPWEIGHTS statement, PROC SURVEYFREQ uses the replicate weights that you provide and includes the Fay coefficient ϵ in the denominator of the variance estimate in the preceding expression.

If a parameter cannot be estimated from some replicate(s), then the variance estimate is computed by using those replicates from which the parameter can be estimated. For example, suppose the parameter is a column proportion—the proportion of column j for table cell (i, j). If a replicate r contains no observations in column j, then the column j proportion is not estimable from replicate r. In this case, the BRR variance estimate is computed as

$$\widehat{V}(\hat{\theta}) = \frac{1}{R'(1-\epsilon)^2} \sum_{r=1}^{R'} \left(\hat{\theta}_r - \hat{\theta}\right)^2$$

where the summation is over the replicates where the parameter θ is estimable, and R' is the number of those replicates.

Hadamard Matrix

PROC SURVEYFREQ uses a Hadamard matrix to construct replicates for BRR variance estimation. You can provide a Hadamard matrix for replicate construction by using the HADAMARD= method-option for VARMETHOD=BRR. Otherwise, PROC SURVEYFREQ generates an appropriate Hadamard matrix. You can display the Hadamard matrix by specifying the PRINTH method-option.

A Hadamard matrix $\mathbf{A}$ of dimension R is a square matrix that has all elements equal to 1 or –1. A Hadamard matrix must satisfy the requirement that $\mathbf{A}'\mathbf{A} = R\mathbf{I}$, where $\mathbf{I}$ is an identity matrix. The dimension of a Hadamard matrix must equal 1, 2, or a multiple of 4.

For example, the following matrix is a Hadamard matrix of dimension $k = 8$:

$$\begin{array}{rrrrrrrr} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{array}$$

For BRR replicate construction, the dimension of the Hadamard matrix must be at least H, where H denotes the number of first-stage strata in your design. If a Hadamard matrix of a given dimension exists, it is not necessarily unique. Therefore, if you want to use a specific Hadamard matrix, you must provide the matrix as a SAS data set in the HADAMARD=*SAS-data-set* method-option. You must ensure that the matrix you provide is actually a Hadamard matrix; PROC SURVEYFREQ does not check the validity of your Hadamard matrix.

See the section "Balanced Repeated Replication (BRR)" on page 6334 and "Fay's BRR Method" on page 6335 for details about how the Hadamard matrix is used to construct replicates for BRR variance estimation.

The Jackknife

If you specify the VARMETHOD=JACKKNIFE option, PROC SURVEYFREQ uses the delete-1 jackknife method for variance estimation. The jackknife method can be used for stratified sample designs and for designs with no stratification. If your design is stratified, the jackknife method requires at least two PSUs in each stratum. You can provide replicate weights for jackknife variance estimation by using a REPWEIGHTS statement, or the procedure can construct replicate weights for the analysis. PROC SURVEYFREQ estimates the parameter of interest (a proportion, total, odds ratio, or other statistic) from each replicate, and then uses the variability among replicate estimates to estimate the overall variance of the parameter estimate. See Wolter (1985) and Lohr (1999) for more information about jackknife variance estimation.

If you do not provide replicate weights with a REPWEIGHTS statement, PROC SURVEYFREQ constructs the replicates. The number of replicates R equals the number of PSUs, and the procedure deletes one PSU from the full sample to form each replicate. The sampling weights are modified by the jackknife coefficient for the replicate to create the replicate weights.

If your design is not stratified (no STRATA statement), the jackknife coefficient has the same value for each replicate r. The jackknife coefficient equals

$$\alpha_r = \frac{R-1}{R} \quad \text{for } r = 1, 2, \ldots, R$$

where R is the total number of replicates (or total number of PSUs). For the PSUs included in a replicate, the replicate weights are computed by dividing the original sampling weights by the jackknife coefficient. For the deleted PSU, which is not included in the replicate, the replicate

weights equal zero. The replicate weight for the jth member of the ith PSU can be expressed as follows when the design is not stratified:

$$W_{ij}^{r} = \begin{cases} W_{ij}/\alpha_r & \text{if PSU } i \text{ is included in replicate } r \\ 0 & \text{otherwise} \end{cases}$$

where W_{ij} is the original sampling weight of unit (ij), r is the replicate number, and α_r is the jackknife coefficient.

If your design is stratified, the jackknife method requires at least two PSUs in each stratum. Let stratum $\tilde{h}_r$ be the stratum from which a PSU is deleted to form the rth replicate. Stratum $\tilde{h}_r$ is called the *donor stratum*. The jackknife coefficients are defined as

$$\alpha_r = \frac{n_{\tilde{h}_r} - 1}{n_{\tilde{h}_r}} \quad \text{for } r = 1, 2, \ldots, R$$

where $n_{\tilde{h}_r}$ is the total number of PSUs in the donor stratum for replicate r. For all strata other than the donor stratum, the replicate r weights equal the original sampling weights. For PSUs included from the donor stratum, the replicate weights are computed by dividing the original sampling weights by the jackknife coefficient. For the deleted PSU, which is not included in the replicate, the replicate weights equal zero. The replicate weight for the jth member of the ith PSU in stratum h can be expressed as

$$W_{hij}^{r} = \begin{cases} W_{hij} & \text{if } h \neq \tilde{h}_r \\ W_{hij}/\alpha_r & \text{if } h = \tilde{h}_r \quad \text{and PSU } (hi) \text{ is included in replicate } r \\ 0 & \text{if } h = \tilde{h}_r \quad \text{and PSU } (hi) \text{ is not included in replicate } r \end{cases}$$

You can use the OUTWEIGHTS= method-option to store the replicate weights in a SAS data set. You can also use the OUTJKCOEFS= method-option to store the jackknife coefficients in a SAS data set. See the sections "Jackknife Coefficients Output Data Set" on page 6355 and "Replicate Weights Output Data Set" on page 6355 for details about the contents of these output data sets. You can provide replicate weights and jackknife coefficients to the procedure for subsequent analyses by using a REPWEIGHTS statement. If you provide replicate weights but do not provide jackknife coefficients, PROC SURVEYFREQ uses $\alpha_r = (R-1)/R$ as the jackknife coefficient for all replicates.

Let θ denote the population parameter to be estimated—for example, a proportion, total, odds ratio, or other statistic. Let $\hat{\theta}$ denote the estimate of θ from the full sample, and let $\hat{\theta}_r$ be the estimate from the rth jackknife replicate, which is computed by using the replicate weights. The jackknife variance estimate for $\hat{\theta}$ is computed as

$$\widehat{V}(\hat{\theta}) = \sum_{r=1}^{R} \alpha_r \left(\hat{\theta}_r - \hat{\theta}\right)^2$$

where R is the total number of replicates and α_r is the jackknife coefficient for replicate r.

If a parameter cannot be estimated from some replicate(s), then the variance estimate is computed by using those replicates from which the parameter can be estimated. For example, suppose the

parameter is a column proportion—the proportion of column j for table cell (i, j). If a replicate r contains no observations in column j, then the column j proportion is not estimable from replicate r. In this case, the jackknife variance estimate is computed as

$$\widehat{V}(\hat{\theta}) = \frac{R}{R'} \sum_{r=1}^{R'} \alpha_r \left(\hat{\theta}_r - \hat{\theta}\right)^2$$

where the summation is over the replicates where the parameter θ is estimable, and R' is the number of those replicates.

Confidence Limits

If you specify the CL option in the TABLES statement, PROC SURVEYFREQ computes confidence limits for the proportions in the frequency and crosstabulation tables. The confidence coefficient is determined according to the value of the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits.

For the proportion in table cell (r, c), the confidence limits are computed as

$$\widehat{P}_{rc} \pm \left(t_{df,\alpha/2} \times \text{StdErr}(\widehat{P}_{rc})\right)$$

where $\widehat{P}_{rc}$ is the estimate of the proportion in table cell (r, c), $\text{StdErr}(\widehat{P}_{rc})$ is the standard error of the estimate, and $t_{df,\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the t distribution with df degrees of freedom calculated as described in the section "Degrees of Freedom" on page 6340. The confidence limits for row proportions and column proportions are computed similarly to the confidence limits for table cell proportions.

If you specify the CLWT option in the TABLES statement, PROC SURVEYFREQ computes confidence limits for the weighted frequencies, or totals, in the crosstabulation tables.

For the total in table cell (r, c), the confidence limits are computed as

$$\widehat{N}_{rc} \pm \left(t_{df,\alpha/2} \times \text{StdErr}(\widehat{N}_{rc})\right)$$

where $\widehat{N}_{rc}$ is the estimate of the total frequency in table cell (r, c), $\text{StdErr}(\hat{N}_{rc})$ is the standard error of the estimate, and $t_{df,\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the t distribution with df degrees of freedom calculated as described in the section "Degrees of Freedom" on page 6340. The confidence limits for row totals, column totals, and the overall total are computed similarly to the confidence limits for table cell totals.

For each table request, PROC SURVEYFREQ produces a nondisplayed ODS table, "Table Summary," which contains the number of observations, strata, and clusters that are included in the analysis of the requested table. When you request confidence limits, the "Table Summary" data set also contains the degrees of freedom df and the value of $t_{df,\alpha/2}$ used to compute the confidence limits. See Example 83.3 for more information about this output data set.

Degrees of Freedom

PROC SURVEYFREQ uses the degrees of freedom of the variance estimator to obtain the t-percentile for confidence limits for proportions, totals, and other statistics. The procedure also uses the degrees of freedom in computing the F statistics for the Rao-Scott and Wald chi-square tests.

PROC SURVEYFREQ computes the degrees of freedom based on the variance estimation method and the sample design. Alternatively, you can specify the degrees of freedom in the DF= option in the TABLES statement instead of having the procedure compute it.

For Taylor series variance estimation, PROC SURVEYFREQ calculates the degrees of freedom (df) as the number of clusters minus the number of strata. If there are no clusters, then df equals the number of observations minus the number of strata. If the design is not stratified, then df equals the number of clusters minus one. These numbers are based on the observations included in the analysis of the individual table request. These numbers do not count observations excluded from the table due to missing values. See the section "Missing Values" on page 6323 for details. If you specify the MISSING option, missing values are treated as valid nonmissing levels and are included in computing degrees of freedom. If you specify the NOMCAR option for Taylor series variance estimation, observations with missing values of the TABLES variables are included in computing degrees of freedom.

If you provide replicate weights with a REPWEIGHTS statement, the degrees of freedom equal the number of replicates, which is the number of REPWEIGHTS variables that you provide. Alternatively, you can specify the degrees of freedom in the DF= option in the REPWEIGHTS or TABLES statement.

For BRR variance estimation (when you do not use a REPWEIGHTS statement), PROC SURVEYFREQ calculates the degrees of freedom as the number of strata. PROC SURVEYFREQ bases the number of strata on all valid observations in the data set, unless you specify the DFADJ method-option for VARMETHOD=BRR. When you specify the DFADJ option, the procedure computes the degrees of freedom as the number of nonmissing strata for the individual table request. This excludes any empty strata that occur when observations with missing values of the TABLES variables are removed from the analysis for that table.

For jackknife variance estimation (when you do not use a REPWEIGHTS statement), PROC SURVEYFREQ calculates the degrees of freedom as the number of clusters minus the number of strata. If there are no clusters, then df equals the number of observations minus the number of strata. If the design is not stratified, then df equals the number of clusters minus one. For jackknife variance estimation, PROC SURVEYFREQ bases the number of strata and clusters on all valid observations in the data set, unless you specify the DFADJ method-option for VARMETHOD=JK. When you specify the DFADJ option, the procedure computes the degrees of freedom from the number of nonmissing strata and clusters for the individual table request. This excludes any empty strata or clusters that occur when observations with missing values of the TABLES variables are removed from the analysis for that table.

For each table request, PROC SURVEYFREQ produces a nondisplayed ODS table, "Table Summary," which contains the number of (nonmissing) observations, strata, and clusters that are included in the analysis of the table. If there are missing observations, empty strata, or empty clusters excluded from the analysis, the "Table Summary" data set also contains this information. If you re-

quest confidence limits or chi-square tests, which depend on the degrees of freedom of the variance estimator, the "Table Summary" data set provides the degrees of freedom df. See Example 83.3 for more information about this output data set.

Coefficient of Variation

If you specify the CV option in the TABLES statement, PROC SURVEYFREQ computes the coefficients of variation for the proportion estimates in the frequency and crosstabulation tables. The coefficient of variation is the ratio of the standard error to the estimate.

For the proportion in table cell (r, c), the coefficient of variation is computed as

$$\mathrm{CV}(\widehat{P}_{rc}) = \mathrm{StdErr}(\widehat{P}_{rc}) \;/\; \widehat{P}_{rc}$$

where $\widehat{P}_{rc}$ is the estimate of the proportion in table cell (r, c) and $\mathrm{StdErr}(\widehat{P}_{rc})$ is the standard error of the estimate. The coefficients of variation for row proportions and column proportions are computed similarly.

If you specify the CVWT option in the TABLES statement, PROC SURVEYFREQ computes the coefficients of variation for the weighted frequencies, or estimated totals, in the crosstabulation tables. For the total in table cell (r, c), the coefficient of variation is computed as

$$\mathrm{CV}(\widehat{N}_{rc}) = \mathrm{StdErr}(\widehat{N}_{rc}) \;/\; \widehat{N}_{rc}$$

where $\widehat{N}_{rc}$ is the estimate of the total in table cell (r, c) and $\mathrm{StdErr}(\widehat{N}_{rc})$ is the standard error of the estimate. The coefficients of variation for row totals, column totals, and the overall total are computed similarly.

Design Effect

If you specify the DEFF option in the TABLES statement, PROC SURVEYFREQ computes design effects for the overall proportion estimates in the frequency and crosstabulation tables. The design effect for an estimate is the ratio of the actual variance (estimated based on the sample design) to the variance of a simple random sample with the same number of observations. See Lohr (1999) and Kish (1965) for details.

For Taylor series variance estimation, PROC SURVEYFREQ computes the design effect for the proportion in table cell (r, c) as

$$\begin{aligned}
\mathrm{DEFF}(\widehat{P}_{rc}) &= \widehat{\mathrm{Var}}(\widehat{P}_{rc}) \;/\; \widehat{\mathrm{Var}}_{\mathrm{SRS}}(\widehat{P}_{rc}) \\
&= \widehat{\mathrm{Var}}(\widehat{P}_{rc}) \;/\; \left\{(1-f)\;\widehat{P}_{rc}\;(1-\widehat{P}_{rc})\;/\;(n-1)\right\}
\end{aligned}$$

where $\widehat{P}_{rc}$ is the estimate of the proportion in table cell (r, c), $\widehat{\mathrm{Var}}(\widehat{P}_{rc})$ is the variance of the estimate, f is the overall sampling fraction, and n is the number of observations in the sample.

For Taylor series variance estimation, PROC SURVEYFREQ determines the value of f, the overall sampling fraction, based on the RATE= or TOTAL= option. If you do not specify either of these

options, then PROC SURVEYFREQ assumes the value of f is negligible and does not use a finite population correction in the analysis, as described in the section "Population Totals and Sampling Rates" on page 6322. If you specify RATE=*value*, then PROC SURVEYFREQ uses this value as the overall sampling fraction f. If you specify TOTAL=*value*, then PROC SURVEYFREQ computes f as the ratio of the number of PSUs in the sample to the specified total.

If you specify stratum sampling rates with the RATE=*SAS-data-set* option, then PROC SURVEYFREQ computes stratum totals based on these stratum sampling rates and the number of sample PSUs in each stratum. The procedure sums the stratum totals to form the overall total, and computes f as the ratio of the number of sample PSUs to the overall total. Alternatively, if you specify stratum totals with the TOTAL=*SAS-data-set* option, then PROC SURVEYFREQ sums these totals to compute the overall total. The overall sampling fraction f is then computed as the ratio of the number of sample PSUs to the overall total.

For BRR and jackknife variance estimation, PROC SURVEYFREQ computes the design effect for the proportion in table cell (r, c) as

$$\begin{aligned} \mathrm{DEFF}(\widehat{P}_{rc}) &= \widehat{\mathrm{Var}}(\widehat{P}_{rc}) \,/\, \widehat{\mathrm{Var}}_{\mathrm{SRS}}(\widehat{P}_{rc}) \\ &= \widehat{\mathrm{Var}}(\widehat{P}_{rc}) \,/\, \left\{ \widehat{P}_{rc}\,(1-\widehat{P}_{rc}) \,/\, (n-1) \right\} \end{aligned}$$

where $\widehat{P}_{rc}$ is the estimate of the proportion in table cell (r, c), $\widehat{\mathrm{Var}}(\widehat{P}_{rc})$ is the variance of the estimate, and n is the number of observations in the sample. This computation does not include the overall sampling fraction.

Expected Weighted Frequency

If you specify the EXPECTED option in the TABLES statement, PROC SURVEYFREQ computes expected weighted frequencies for the table cells in two-way tables. The expected weighted frequencies are computed under the null hypothesis that the row and column variables are independent. The expected weighted frequency for table cell (r, c) equals

$$E_{rc} = \widehat{N}_{r\cdot}\, \widehat{N}_{\cdot c} \,/\, \widehat{N}$$

where $\widehat{N}_{r\cdot}$ is the estimated total for row r, $\widehat{N}_{\cdot c}$ is the estimated total for column c, and $\widehat{N}$ is the estimated overall total. Equivalently, the expected weighted frequency can be expressed as

$$E_{rc} = \widehat{P}_{r\cdot}\, \widehat{P}_{\cdot c}\, \widehat{N}$$

These expected values are used in the design-based chi-square tests of independence, as described in the sections "Rao-Scott Chi-Square Test" on page 6346 and "Wald Chi-Square Test" on page 6352.

Risks and Risk Difference

The RISK option provides estimates of risks (or binomial proportions) and risk differences for 2×2 tables, together with their standard errors and confidence limits. Risk statistics include the row 1 risk, row 2 risk, overall risk, and risk difference. If you specify the RISK option, PROC

SURVEYFREQ provides both column 1 and column 2 risks. You can request only column 1 (or only column 2) risks by specifying the RISK1 (or RISK2) option.

The column 1 risk for row 1 is the row 1 proportion for table cell $(1, 1)$. The column 1 risk estimate is computed as the ratio of the estimated total for table cell $(1, 1)$ to the estimated total for row 1,

$$\widehat{P}_{11}^{(1)} = \widehat{N}_{11} \,/\, \widehat{N}_{1\cdot}$$

where the total estimates are computed as described in the section "Totals" on page 6329. The column 1 risk for row 2 is the row 2 proportion for table cell $(2, 1)$, which is estimated as

$$\widehat{P}_{21}^{(2)} = \widehat{N}_{21} \,/\, \widehat{N}_{2\cdot}$$

The overall column 1 risk is the overall proportion in column 1, and its estimate is computed as

$$\widehat{P}_{\cdot 1} = \widehat{N}_{\cdot 1} \,/\, \widehat{N}$$

The column 2 risk estimates are computed similarly.

The row 1 and row 2 risks are the same as the row proportions for a 2×2 table, and their variances are computed as described in the section "Row and Column Proportions" on page 6332. The overall risk is the overall proportion in the column, and its variance computation is described in the section "Proportions" on page 6331. Confidence limits for the column 1 risk for row 1 are computed as

$$\widehat{P}_{11}^{(1)} \pm \left(t_{df,\alpha/2} \times \mathrm{StdErr}(\widehat{P}_{11}^{(1)}) \right)$$

where $\mathrm{StdErr}(\widehat{P}_{11}^{(1)})$ is the standard error of the risk estimate, and $t_{df,\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the t distribution with df degrees of freedom calculated as described in the section "Degrees of Freedom" on page 6340. The value of the confidence coefficient α is determined by the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits. Confidence limits for the other risks are computed similarly.

The risk difference is defined as the row 1 risk minus the row 2 risk. The estimate of the column 1 risk difference $\widehat{RD}_1$ is computed as

$$\begin{aligned} \widehat{RD}_1 &= \widehat{P}_{11}^{(1)} - \widehat{P}_{21}^{(2)} \\ &= \left(\widehat{N}_{11} \,/\, \widehat{N}_{1\cdot} \right) - \left(\widehat{N}_{21} \,/\, \widehat{N}_{2\cdot} \right) \end{aligned}$$

The column 2 risk difference is computed similarly.

PROC SURVEYFREQ estimates the variance of the risk difference by using the variance estimation method that you request. If you request BRR variance estimation (VARMETHOD=BRR), the procedure estimates the variance as described in the section "Balanced Repeated Replication (BRR)" on page 6334. If you request jackknife variance estimation (VARMETHOD=JACKKNIFE), the procedure estimates the variance as described in the section "The Jackknife" on page 6337.

If you do not specify the VARMETHOD= option or a REPWEIGHTS statement, the default variance estimation method is Taylor series (VARMETHOD=TAYLOR). By using Taylor series linearization, the variance estimate for the column 1 risk difference $\widehat{\mathrm{Var}}(\widehat{RD}_1)$ can be expressed as

$$\widehat{\mathrm{Var}}(\widehat{RD}_1) = \widehat{\mathbf{D}}\, \widehat{\mathbf{V}}(\widehat{\mathbf{X}})\, \widehat{\mathbf{D}}'$$

where $\widehat{\mathbf{V}}(\widehat{\mathbf{X}})$ is the covariance matrix of $\widehat{\mathbf{X}}$,

$$\widehat{\mathbf{X}} = (\ \widehat{N}_{11},\ \widehat{N}_{1\cdot},\ \widehat{N}_{21},\ \widehat{N}_{2\cdot}\ )$$

and $\widehat{\mathbf{D}}$ is an array containing the partial derivatives of the risk difference with respect to the elements of $\widehat{\mathbf{X}}$,

$$\widehat{\mathbf{D}} = (\ 1/\widehat{N}_{1\cdot},\ -\widehat{N}_{11}/\widehat{N}_{1\cdot}^{2},\ -1/\widehat{N}_{2\cdot},\ -\widehat{N}_{21}/\widehat{N}_{2\cdot}^{2}\ )$$

See Wolter (1985, pp. 239–242) for details. The variance estimate for the column 2 risk difference is computed similarly.

The standard error of the column 1 risk difference is

$$\text{StdErr}(\widehat{RD}_1) = \sqrt{\widehat{\text{Var}}(\widehat{RD}_1)}$$

Confidence limits for the column 1 risk difference are computed as

$$\widehat{RD}_1 \pm \left(t_{df,\alpha/2} \times \text{StdErr}(\widehat{RD}_1)\right)$$

where $t_{df,\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the t distribution with df degrees of freedom calculated as described in the section "Degrees of Freedom" on page 6340. The value of the confidence coefficient α is determined by the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits. Confidence limits for the column 2 risk difference are computed in the same way.

Odds Ratio and Relative Risks

The OR option provides estimates of the odds ratio, the column 1 relative risk, and the column 2 relative risk for 2×2 tables, together with their confidence limits.

Odds Ratio

For a 2×2 table, the odds of a positive (column 1) response in row 1 is N_{11}/N_{12}. Similarly, the odds of a positive response in row 2 is N_{21}/N_{22}. The odds ratio is formed as the ratio of the row 1 odds to the row 2 odds. The estimate of the odds ratio is computed as

$$\widehat{OR} = \frac{\widehat{N}_{11}\ /\ \widehat{N}_{12}}{\widehat{N}_{21}\ /\ \widehat{N}_{22}} = \frac{\widehat{N}_{11}\ \widehat{N}_{22}}{\widehat{N}_{12}\ \widehat{N}_{21}}$$

The value of the odds ratio can be any nonnegative number. When the row and column variables are independent, the true value of the odds ratio equals 1. An odds ratio greater than 1 indicates that the odds of a positive response are higher in row 1 than in row 2. An odds ratio less than 1 indicates that the odds of positive response are higher in row 2. The strength of association increases with the deviation from 1. See Stokes, Davis, and Koch (2000) and Agresti (2007) for details.

PROC SURVEYFREQ constructs confidence limits for the odds ratio by using the log transform. The $100(1-\alpha)\%$ confidence limits for the odds ratio are computed as

$$\left(\ \widehat{OR} \times \exp(-t_{df,\alpha/2}\ \sqrt{v}),\ \widehat{OR} \times \exp(t_{df,\alpha/2}\ \sqrt{v})\ \right)$$

where

$$v = \widehat{\mathrm{Var}}(\ln \widehat{OR}) = \widehat{\mathrm{Var}}(\widehat{OR}) \;/\; \widehat{OR}^{\,2}$$

is the estimate of the variance of the log odds ratio, and where $t_{df,\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the t distribution with df degrees of freedom. The computation of df is described in the section "Degrees of Freedom" on page 6340. The value of the confidence coefficient α is determined by the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits.

If you request BRR variance estimation (VARMETHOD=BRR), PROC SURVEYFREQ estimates the variance of the odds ratio as described in the section "Balanced Repeated Replication (BRR)" on page 6334. If you request jackknife variance estimation (VARMETHOD=JACKKNIFE), the procedure estimates the variance as described in the section "The Jackknife" on page 6337.

If you do not specify the VARMETHOD= option or a REPWEIGHTS statement, the default variance estimation method is Taylor series (VARMETHOD=TAYLOR). By using Taylor series linearization, the variance estimate for the odds ratio can be expressed as

$$\widehat{\mathrm{Var}}(\widehat{OR}) = \widehat{\mathbf{D}}\,\widehat{\mathbf{V}}(\widehat{\mathbf{N}})\,\widehat{\mathbf{D}}'$$

where $\widehat{\mathbf{V}}(\widehat{\mathbf{N}})$ is the covariance matrix of the estimates of the cell totals $\widehat{\mathbf{N}}$,

$$\widehat{\mathbf{N}} = (\,\widehat{N}_{11},\ \widehat{N}_{12},\ \widehat{N}_{21},\ \widehat{N}_{22}\,)$$

and $\widehat{\mathbf{D}}$ is an array containing the partial derivatives of the odds ratio with respect to the elements of $\widehat{\mathbf{N}}$. The section "Covariance of Totals" on page 6330 describes the computation of $\widehat{\mathbf{V}}(\widehat{\mathbf{N}})$. The array $\widehat{\mathbf{D}}$ is computed as

$$\begin{aligned}\widehat{\mathbf{D}} \;=\; (\;& \widehat{N}_{22}/\widehat{N}_{12}\widehat{N}_{21}, \quad -\widehat{N}_{11}\widehat{N}_{22}/\widehat{N}_{21}\widehat{N}_{12}^{\,2},\\ & -\widehat{N}_{11}\widehat{N}_{22}/\widehat{N}_{12}\widehat{N}_{21}^{\,2}, \quad \widehat{N}_{11}/\widehat{N}_{12}\widehat{N}_{21} \quad)\end{aligned}$$

See Wolter (1985, pp. 239–242) for more information.

Relative Risks

For a 2×2 table, the column 1 relative risk is the ratio of the column 1 risks for row 1 to row 2. As described in the section "Risks and Risk Difference" on page 6342, the column 1 risk for row 1 is the proportion of row 1 observations classified in column 1, and the column 1 risk for row 2 is the proportion of row 2 observations classified in column 1. The estimate of the column 1 relative risk is computed as

$$\widehat{RR}_1 = \frac{\widehat{N}_{11} \;/\; \widehat{N}_{1\cdot}}{\widehat{N}_{21} \;/\; \widehat{N}_{2\cdot}}$$

Similarly, the estimate of the column 2 relative risk is computed as

$$\widehat{RR}_2 = \frac{\widehat{N}_{12} \;/\; \widehat{N}_{1\cdot}}{\widehat{N}_{22} \;/\; \widehat{N}_{2\cdot}}$$

A relative risk greater than 1 indicates that the probability of positive response is greater in row 1 than in row 2. Similarly, a relative risk less than 1 indicates that the probability of positive response

is less in row 1 than in row 2. The strength of association increases with the deviation from 1. See Stokes, Davis, and Koch (2000) and Agresti (2007) for more information.

PROC SURVEYFREQ constructs confidence limits for the relative risk by using the log transform, which is similar to the odds ratio computations described previously. The $100(1-\alpha)\%$ confidence limits for the column 1 relative risk are computed as

$$\left(\widehat{RR}_1 \times \exp(-t_{df,\alpha/2}\sqrt{v}),\ \widehat{RR}_1 \times \exp(t_{df,\alpha/2}\sqrt{v}) \right)$$

where

$$v = \widehat{\mathrm{Var}}(\ln \widehat{RR}_1) = \widehat{\mathrm{Var}}(\widehat{RR}_1) \ / \ \widehat{RR}_1^2$$

is the estimate of the variance of the log column 1 relative risk, and where $t_{df,\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the t distribution with df degrees of freedom. The computation of df is described in the section "Degrees of Freedom" on page 6340. The value of the confidence coefficient α is determined by the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits.

If you request BRR variance estimation (VARMETHOD=BRR), PROC SURVEYFREQ estimates the variance of the column 1 relative risk as described in the section "Balanced Repeated Replication (BRR)" on page 6334. If you request jackknife variance estimation (VARMETHOD=JACKKNIFE), the procedure estimates the variance as described in the section "The Jackknife" on page 6337.

If you do not specify the VARMETHOD= option or a REPWEIGHTS statement, the default variance estimation method is Taylor series (VARMETHOD=TAYLOR). By using Taylor series linearization, the variance estimate for the column 1 relative risk can be expressed as

$$\widehat{\mathrm{Var}}(\widehat{RR}_1) = \widehat{\mathbf{D}}\ \widehat{\mathbf{V}}(\widehat{\mathbf{X}})\ \widehat{\mathbf{D}}'$$

where $\widehat{\mathbf{V}}(\widehat{\mathbf{X}})$ is the covariance matrix of $\widehat{\mathbf{X}}$,

$$\widehat{\mathbf{X}} = (\ \widehat{N}_{11},\ \widehat{N}_{1\cdot},\ \widehat{N}_{21},\ \widehat{N}_{2\cdot}\ )$$

and $\widehat{\mathbf{D}}$ is an array containing the partial derivatives of the column 1 relative risk with respect to the elements of $\widehat{\mathbf{X}}$,

$$\begin{aligned}\widehat{\mathbf{D}} \ = \ (\ & \widehat{N}_{2\cdot}/\widehat{N}_{21}\widehat{N}_{1\cdot}, \quad -\widehat{N}_{11}\widehat{N}_{2\cdot}/\widehat{N}_{21}\widehat{N}_{1\cdot}^2, \\ & -\widehat{N}_{11}\widehat{N}_{2\cdot}/\widehat{N}_{1\cdot}\widehat{N}_{21}^2, \quad \widehat{N}_{11}/\widehat{N}_{21}\widehat{N}_{1\cdot} \quad)\end{aligned}$$

See Wolter (1985, pp. 239–242) for more information.

Confidence limits for the column 2 relative risk are computed similarly.

Rao-Scott Chi-Square Test

The Rao-Scott chi-square test is a design-adjusted version of the Pearson chi-square test, which involves differences between observed and expected frequencies. For two-way tables, the null hypothesis for this test is no association between the row and column variables. For one-way tables,

the null hypothesis is equal proportions for the variable levels. Or you can specify null hypothesis proportions for one-way tables by using the TESTP= option.

Two forms of the design correction are available for the Rao-Scott tests. One form of the design correction uses the proportion estimates, and you request the corresponding Rao-Scott chi-square test with the CHISQ option. The other form of the design correction uses the null hypothesis proportions. You request this test, called the Rao-Scott modified chi-square test, with the CHISQ1 option.

See Lohr (1999), Thomas, Singh, and Roberts (1996), and Rao and Scott (1981, 1984, 1987) for details about design-adjusted chi-square tests.

Two-Way Tables

The Rao-Scott chi-square statistic is computed from the Pearson chi-square statistic and a design correction based on the design effects of the proportions. Under the null hypothesis of no association between the row and column variables, this statistic approximately follows a chi-square distribution with $(R-1)(C-1)$ degrees of freedom. PROC SURVEYFREQ also computes an F statistic that can provide a better approximation.

The Rao-Scott chi-square Q_{RS} is computed as

$$Q_{RS} = Q_P \;/\; D$$

where D is the design correction described in the section "Design Correction for Two-Way Tables" on page 6348, and Q_P is the Pearson chi-square based on the estimated totals. The Pearson chi-square is computed as

$$Q_P = (n/\widehat{N}) \sum_r \sum_c (\widehat{N}_{rc} - E_{rc})^2 \;/\; E_{rc}$$

where n is the sample size, $\widehat{N}$ is the estimated overall total, $\widehat{N}_{rc}$ is the estimated total for table cell (r, c), and E_{rc} is the expected total for table cell (r, c) under the null hypothesis of no association,

$$E_{rc} = \widehat{N}_{r\cdot}\, \widehat{N}_{\cdot c} \;/\; \widehat{N}$$

Under the null hypothesis of no association, the Rao-Scott chi-square Q_{RS} approximately follows a chi-square distribution with $(R-1)(C-1)$ degrees of freedom. A better approximation can be obtained by the F statistic,

$$F = Q_{RS} \;/\; (R-1)(C-1)$$

which has an F distribution with $(R-1)(C-1)$ and $(R-1)(C-1)\kappa$ degrees of freedom under the null hypothesis. The value κ is the degrees of freedom for the variance estimator and depends on the sample design and the variance estimation method. The section "Degrees of Freedom" on page 6340 describes the computation of κ.

Design Correction for Two-Way Tables

If you specify the CHISQ or LRCHISQ option, the design correction D is computed by using the estimated proportions as

$$D = \left\{ \sum_r \sum_c (1 - \widehat{P}_{rc}) \text{ DEFF}(\widehat{P}_{rc}) - \sum_r (1 - \widehat{P}_{r\cdot}) \text{ DEFF}(\widehat{P}_{r\cdot}) \right. \\ \left. - \sum_c (1 - \widehat{P}_{\cdot c}) \text{ DEFF}(\widehat{P}_{\cdot c}) \right\} / (R-1)(C-1)$$

where

$$\begin{aligned} \text{DEFF}(\widehat{P}_{rc}) &= \widehat{\text{Var}}(\widehat{P}_{rc}) / \text{Var}_{\text{SRS}}(\widehat{P}_{rc}) \\ &= \text{Var}(\widehat{P}_{rc}) / \left\{ (1-f)\, \widehat{P}_{rc}\, (1 - \widehat{P}_{rc}) / (n-1) \right\} \end{aligned}$$

as described in the section "Design Effect" on page 6341. $\widehat{P}_{rc}$ is the estimate of the proportion in table cell (r, c), $\widehat{\text{Var}}(\widehat{P}_{rc})$ is the variance of the estimate, f is the overall sampling fraction, and n is the number of observations in the sample. $\text{DEFF}(\widehat{P}_{r\cdot})$, the design effect for the estimate of the proportion in row r, and $\text{DEFF}(\widehat{P}_{\cdot c})$, the design effect for the estimate of the proportion in column c, are computed similarly.

If you specify the CHISQ1 or LRCHISQ1 option for the Rao-Scott modified test, the design correction uses the null hypothesis cell proportions instead of the estimated cell proportions. For two-way tables, the null hypothesis cell proportions are computed as the products of the corresponding row and column proportion estimates. The modified design correction D_0 (based on null hypothesis proportions) is computed as

$$D_0 = \left\{ \sum_r \sum_c (1 - P_{rc}^0) \text{ DEFF}_0(\widehat{P}_{rc}) - \sum_r (1 - \widehat{P}_{r\cdot}) \text{ DEFF}(\widehat{P}_{r\cdot}) \right. \\ \left. - \sum_c (1 - \widehat{P}_{\cdot c}) \text{ DEFF}(\widehat{P}_{\cdot c}) \right\} / (R-1)(C-1)$$

where

$$P_{rc}^0 = \widehat{P}_{r\cdot} \times \widehat{P}_{\cdot c}$$

and

$$\begin{aligned} \text{DEFF}_0(\widehat{P}_{rc}) &= \widehat{\text{Var}}(\widehat{P}_{rc}) / \text{Var}_{\text{SRS}}(P_{rc}^0) \\ &= \widehat{\text{Var}}(\widehat{P}_{rc}) / \left\{ (1-f)\, P_{rc}^0\, (1 - P_{rc}^0) / (n-1) \right\} \end{aligned}$$

One-Way Tables

For one-way tables, the Rao-Scott chi-square statistic provides a design-based goodness-of-fit test for equal proportions. Or if you specify null proportions with the TESTP= option, the Rao-Scott chi-square provides a design-based goodness-of-fit test for the specified proportions. Under the null hypothesis, the Rao-Scott chi-square statistic approximately follows a chi-square distribution with

$(C-1)$ degrees of freedom for a table with C levels. PROC SURVEYFREQ also computes an F statistic that can provide a better approximation.

The Rao-Scott chi-square Q_{RS} is computed as

$$Q_{RS} = Q_P \;/\; D$$

where D is the design correction described in the section "Design Correction for One-Way Tables" on page 6349, and Q_P is the Pearson chi-square based on the estimated totals. The Pearson chi-square is computed as

$$Q_P = (n/\widehat{N}) \sum_c (\widehat{N}_c - E_c)^2 \;/\; E_c$$

where n is the sample size, $\widehat{N}$ is the estimated overall total, $\widehat{N}_c$ is the estimated total for level c, and E_c is the expected total for level c under the null hypothesis. For the null hypothesis of equal proportions, the expected total for level c equals

$$E_c = \widehat{N} \;/\; C$$

For specified null proportions, the expected total for level c equals

$$E_c = \widehat{N} \;\times\; P_c^{\,0}$$

where $P_c^{\,0}$ is the null proportion for level c.

Under the null hypothesis, the Rao-Scott chi-square Q_{RS} approximately follows a chi-square distribution with $(C-1)$ degrees of freedom. A better approximation can be obtained by the F statistic,

$$F = Q_{RS} \;/\; (C-1)$$

which has an F distribution with $(C-1)$ and $(C-1)\kappa$ degrees of freedom under the null hypothesis. The value κ is the degrees of freedom for the variance estimator and depends on the sample design and the variance estimation method. The section "Degrees of Freedom" on page 6340 describes the computation of κ.

Design Correction for One-Way Tables

If you specify the CHISQ or LRCHISQ option, the design correction D is computed by using the estimated proportions as

$$D = \sum_c (1-\widehat{P}_c)\,\mathrm{DEFF}(\widehat{P}_c) \;/\; (C-1)$$

where

$$\begin{aligned}\mathrm{DEFF}(\widehat{P}_c) &= \widehat{\mathrm{Var}}(\widehat{P}_c) \;/\; \mathrm{Var}_{\mathrm{SRS}}(\widehat{P}_c) \\ &= \widehat{\mathrm{Var}}(\widehat{P}_c) \;/\; \left\{(1-f)\;\widehat{P}_c\;(1-\widehat{P}_c) \;/\; (n-1)\right\}\end{aligned}$$

as described in the section "Design Effect" on page 6341. $\widehat{P}_c$ is the proportion estimate for table level c, $\widehat{\mathrm{Var}}(\widehat{P}_c)$ is the variance of the estimate, f is the overall sampling fraction, and n is the number of observations in the sample.

If you specify the CHISQ1 or LRCHISQ1 option for the Rao-Scott modified test, the design correction uses the null hypothesis proportions—either equal proportions for all levels, or the proportions that you specify with the TESTP= option. The modified design correction D_0 is computed as

$$D_0 = \sum_c (1 - {P_c}^0)\ \mathrm{DEFF}_0(\widehat{P}_c)\ /\ (C-1)$$

where

$$\begin{aligned} \mathrm{DEFF}_0(\widehat{P}_c) &= \widehat{\mathrm{Var}}(\widehat{P}_c)\ /\ \mathrm{Var}_{\mathrm{SRS}}({P_c}^0) \\ &= \widehat{\mathrm{Var}}(\widehat{P}_c)\ /\ \left\{(1-f){P_c}^0\ (1-{P_c}^0)\ /\ (n-1)\right\} \end{aligned}$$

and ${P_c}^0 = 1/C$ for equal proportions, or ${P_c}^0$ equals the null proportion for level c if you specify the TESTP= option.

Rao-Scott Likelihood Ratio Chi-Square Test

The Rao-Scott likelihood ratio chi-square test is a design-adjusted version of the likelihood ratio test, which involves ratios between observed and expected frequencies. For two-way tables, the null hypothesis for this test is no association between the row and column variables. For one-way tables, the null hypothesis is equal proportions for the variable levels. Or you can specify null hypothesis proportions for one-way tables by using the TESTP= option.

Two forms of the design correction are available for the Rao-Scott tests. One form of the design correction uses the proportion estimates, and you request the corresponding Rao-Scott likelihood ratio test with the LRCHISQ option. The other form of the design correction uses the null hypothesis proportions. You request this test, called the Rao-Scott modified likelihood ratio test, with the LRCHISQ1 option.

See Lohr (1999), Thomas, Singh, and Roberts (1996), and Rao and Scott (1981, 1984, 1987) for details about design-adjusted chi-square tests.

Two-Way Tables

The Rao-Scott likelihood ratio statistic is computed from the likelihood ratio chi-square statistic and a design correction based on the design effects of the proportions. Under the null hypothesis of no association between the row and column variables, this statistic approximately follows a chi-square distribution with $(R-1)(C-1)$ degrees of freedom. PROC SURVEYFREQ also computes an F statistic that can provide a better approximation.

The Rao-Scott likelihood ratio chi-square G^2_{RS} is computed as

$$G^2_{RS} = G^2\ /\ D$$

where D is the design correction described in the section "Design Correction for Two-Way Tables" on page 6348, and G^2 is the likelihood ratio chi-square based on the estimated totals. The likelihood ratio chi-square is computed as

$$G^2 = 2\ (n/\widehat{N})\ \sum_r \sum_c \widehat{N}_{rc}\ \ln\left(\widehat{N}_{rc}\ /\ E_{rc}\right)$$

where n is the sample size, $\widehat{N}$ is the estimated overall total, $\widehat{N}_{rc}$ is the estimated total for table cell (r, c), and E_{rc} is the expected total for cell (r, c) under the null hypothesis of no association. The expected total for cell (r, c) equals

$$E_{rc} = \widehat{N}_{r\cdot}\ \widehat{N}_{\cdot c}\ /\ \widehat{N}$$

Under the null hypothesis of no association, the Rao-Scott likelihood ratio chi-square G^2_{RS} approximately follows a chi-square distribution with $(R-1)(C-1)$ degrees of freedom. A better approximation can be obtained by the F statistic,

$$F = G^2_{RS}\ /\ (R-1)(C-1)$$

which has an F distribution with $(R-1)(C-1)$ and $(R-1)(C-1)\kappa$ degrees of freedom under the null hypothesis. The value κ is the degrees of freedom for the variance estimator and depends on the sample design and the variance estimation method. The section "Degrees of Freedom" on page 6340 describes the computation of κ.

One-Way Tables

For one-way tables, the Rao-Scott likelihood ratio chi-square statistic provides a design-based goodness-of-fit test for equal proportions. Or if you specify null proportions with the TESTP= option, the Rao-Scott likelihood ratio chi-square provides a design-based goodness-of-fit test for the specified proportions. Under the null hypothesis, the Rao-Scott likelihood ratio statistic approximately follows a chi-square distribution with $(C-1)$ degrees of freedom for a table with C levels. PROC SURVEYFREQ also computes an F statistic that can provide a better approximation.

The Rao-Scott likelihood ratio chi-square G^2_{RS} is computed as

$$G^2_{RS} = G^2\ /\ D$$

where D is the design correction described in the section "Design Correction for One-Way Tables" on page 6349, and G^2 is the likelihood ratio chi-square based on the estimated totals. The likelihood ratio chi-square is computed as

$$G^2 = 2\ \left(\ n\ /\ \widehat{N}\ \right)\ \sum_c \widehat{N}_c\ \ln\left(\widehat{N}_c\ /\ E_c\right)$$

where n is the sample size, $\widehat{N}$ is the estimated overall total, $\widehat{N}_c$ is the estimated total for level c, and E_c is the expected total for level c under the null hypothesis. For the null hypothesis of equal proportions, the expected total for each level equals

$$E_c = \widehat{N}\ /\ C$$

For specified null proportions, the expected total for level c equals

$$E_c = \widehat{N} \times {P_c}^0$$

where ${P_c}^0$ is the null proportion for level c.

Under the null hypothesis of no association, the Rao-Scott likelihood ratio chi-square G^2_{RS} approximately follows a chi-square distribution with $(C-1)$ degrees of freedom. A better approximation can be obtained by the F statistic,

$$F = G^2_{RS} \; / \; (C-1)$$

which has an F distribution with $(C-1)$ and $(C-1)\kappa$ degrees of freedom under the null hypothesis, The value κ is the degrees of freedom for the variance estimator and depends on the sample design and the variance estimation method. The section "Degrees of Freedom" on page 6340 describes the computation of κ.

Wald Chi-Square Test

PROC SURVEYFREQ provides two Wald chi-square tests for independence of the row and column variables in a two-way table: a Wald chi-square test based on the difference between observed and expected weighted cell frequencies, and a Wald log-linear chi-square test based on the log odds ratios. These statistics test for independence of the row and column variables in two-way tables, taking into account the complex survey design. See Bedrick (1983), Koch, Freeman, and Freeman (1975), and Wald (1943) for information about Wald statistics and their applications to categorical data analysis.

For these two tests, PROC SURVEYFREQ computes the generalized Wald chi-square statistic, the corresponding Wald F statistic, and also an adjusted Wald F statistic for tables larger than 2×2. Under the null hypothesis of independence, the Wald chi-square statistic approximately follows a chi-square distribution with $(R-1)(C-1)$ degrees of freedom for large samples. However, it has been shown that this test can perform poorly in terms of actual significance level and power, especially for tables with a large number of cells or for samples with a relatively small number of clusters. See Thomas and Rao (1984 and 1985) and Lohr (1999) for more information. See Felligi (1980) and Hidiroglou, Fuller, and Hickman (1980) for information about the adjusted Wald F statistic. Thomas and Rao (1984) found that the adjusted Wald F statistic provides a more stable test than the chi-square statistic, although its power can be low when the number of sample clusters is not large. See also Korn and Graubard (1990) and Thomas, Singh, and Roberts (1996).

If you specify the WCHISQ option in the TABLES statement, PROC SURVEYFREQ computes a Wald test for independence in the two-way table based on the differences between the observed (weighted) cell frequencies and the expected frequencies.

Under the null hypothesis of independence of the row and column variables, the expected cell frequencies are computed as

$$E_{rc} = \widehat{N}_{r\cdot} \, \widehat{N}_{\cdot c} \; / \; \widehat{N}$$

where $\widehat{N}_{r\cdot}$ is the estimated total for row r, $\widehat{N}_{\cdot c}$ is the estimated total for column c, and $\widehat{N}$ is the estimated overall total, as described in the section "Expected Weighted Frequency" on page 6342. The null hypothesis that the population weighted frequencies equal the expected frequencies can be expressed as

$$H_0 \colon Y_{rc} = N_{rc} \; - \; E_{rc} \; = \; 0$$

for all $r = 1, \ldots (R-1)$ and $c = 1, \ldots (C-1)$. This null hypothesis can be stated equivalently in terms of cell proportions, with the expected cell proportions computed as the products of the marginal row and column proportions.

The generalized Wald chi-square statistic Q_W is computed as

$$Q_W = \widehat{\mathbf{Y}}' \, (\mathbf{H} \, \widehat{\mathbf{V}}(\widehat{\mathbf{N}}) \, \mathbf{H}')^{-1} \, \widehat{\mathbf{Y}}$$

where $\widehat{\mathbf{Y}}$ is the $(R-1)(C-1)$ array of differences between the observed and expected weighted frequencies $(\widehat{N}_{rc} - E_{rc})$, and $(\mathbf{H} \, \widehat{\mathbf{V}}(\widehat{\mathbf{N}}) \, \mathbf{H}')$ estimates the variance of $\widehat{\mathbf{Y}}$.

$\widehat{\mathbf{V}}(\widehat{\mathbf{N}})$ is the covariance matrix of the estimates $\widehat{N}_{rc}$, and its computation is described in the section "Covariance of Totals" on page 6330.

$\mathbf{H}$ is an $(R-1)(C-1)$ by RC matrix containing the partial derivatives of the elements of $\widehat{\mathbf{Y}}$ with respect to the elements of $\widehat{\mathbf{N}}$. The elements of $\mathbf{H}$ are computed as follows, where a denotes a row different from row r, and b denotes a column different from column c:

$$\begin{aligned}
\partial \widehat{Y}_{rc} / \partial \widehat{N}_{rc} &= 1 - \left(\widehat{N}_{r\cdot} + \widehat{N}_{\cdot c} - \widehat{N}_{\cdot c} \, \widehat{N}_{r\cdot} \, / \, \widehat{N} \right) / \, \widehat{N} \\
\partial \widehat{Y}_{rc} / \partial \widehat{N}_{ac} &= - \left(\widehat{N}_{r\cdot} - \widehat{N}_{r\cdot} \, \widehat{N}_{\cdot c} \, / \, \widehat{N} \right) / \, \widehat{N} \\
\partial \widehat{Y}_{rc} / \partial \widehat{N}_{rb} &= - \left(\widehat{N}_{\cdot c} - \widehat{N}_{r\cdot} \, \widehat{N}_{\cdot c} \, / \, \widehat{N} \right) / \, \widehat{N} \\
\partial \widehat{Y}_{rc} / \partial \widehat{Y}_{ab} &= \widehat{N}_{r\cdot} \, \widehat{N}_{\cdot c} \, / \, \widehat{N}^{\,2}
\end{aligned}$$

Under the null hypothesis of independence, the statistic Q_W approximately follows a chi-square distribution with $(R-1)(C-1)$ degrees of freedom for large samples.

PROC SURVEYFREQ computes the Wald F statistic as

$$F_W = Q_W \, / \, (R-1)(C-1)$$

Under the null hypothesis of independence, F_W approximately follows an F distribution with $(R-1)(C-1)$ numerator degrees of freedom. The denominator degrees of freedom are the degrees of freedom for the variance estimator and depend on the sample design and the variance estimation method. The section "Degrees of Freedom" on page 6340 describes the computation of the denominator degrees of freedom. Alternatively, you can specify the denominator degrees of freedom with the DF= option in the TABLES statement.

For tables larger than 2×2, PROC SURVEYFREQ also computes the adjusted Wald F statistic as

$$F_{Adj\_W} = \frac{s - k + 1}{k \, s} \, Q_W$$

where $k = (R-1)(C-1)$, and s is the degrees of freedom, which are computed as described in the section "Degrees of Freedom" on page 6340. Alternatively, you can specify the value of s with the DF= option in the TABLES statement. Note that for 2×2 tables, $k = (R-1)(C-1) = 1$, so the adjusted Wald F statistic equals the (unadjusted) Wald F statistic, with the same numerator and denominator degrees of freedom.

Under the null hypothesis, F_{Adj_W} approximately follows an F distribution with k numerator degrees of freedom and $(s - k + 1)$ denominator degrees of freedom.

Wald Log-Linear Chi-Square Test

If you specify the WLLCHISQ option in the TABLES statement, PROC SURVEYFREQ computes a Wald test for independence based on the log odds ratios. See the section "Wald Chi-Square Test" on page 6352 for more information about Wald tests.

For a two-way table of R rows and C columns, the Wald log-linear test is based on the $(R-1)(C-1)$ array of elements $\widehat{Y}_{rc}$,

$$\widehat{Y}_{rc} = log\widehat{N}_{rc} - log\widehat{N}_{rC} - log\widehat{N}_{Rc} + log\widehat{N}_{RC}$$

where $\widehat{N}_{rc}$ is the estimated total for table cell (r, c). The null hypothesis of independence between the row and column variables can be expressed as $H_0: Y_{rc} = 0$ for all $r = 1, \ldots (R-1)$ and $c = 1, \ldots (C-1)$. This null hypothesis can be stated equivalently in terms of cell proportions.

The generalized Wald log-linear chi-square statistic is computed as

$$Q_{WLL} = \widehat{\mathbf{Y}}' \, \widehat{\mathbf{V}}(\widehat{\mathbf{Y}})^{-1} \, \widehat{\mathbf{Y}}$$

where $\widehat{\mathbf{Y}}$ is the $(R-1)(C-1)$ array of the $\widehat{Y}_{rc}$, and $\widehat{\mathbf{V}}(\widehat{\mathbf{Y}})$ estimates the variance of $\widehat{\mathbf{Y}}$,

$$\widehat{\mathbf{V}}(\widehat{\mathbf{Y}}) = \mathbf{A} \, \mathbf{D}^{-1} \, \widehat{V}(\widehat{\mathbf{N}}) \, \mathbf{D}^{-1} \, \mathbf{A}'$$

where $\widehat{\mathbf{V}}(\widehat{\mathbf{N}})$ is the covariance matrix of the estimates $\widehat{N}_{rc}$, which is computed as described in the section "Covariance of Totals" on page 6330. $\mathbf{D}$ is a diagonal matrix with the estimated totals $\widehat{N}_{rc}$ on the diagonal, and $\mathbf{A}$ is the $(R-1)(C-1)$ by $RC \times RC$ linear contrast matrix.

Under the null hypothesis of independence, the statistic Q_{WLL} approximately follows a chi-square distribution with $(R-1)(C-1)$ degrees of freedom for large samples.

PROC SURVEYFREQ computes the Wald log-linear F statistic as

$$F_{WLL} = Q_{WLL} \; / \; (R-1)(C-1)$$

Under the null hypothesis of independence, F_{WLL} approximately follows an F distribution with $(R-1)(C-1)$ numerator degrees of freedom. PROC SURVEYFREQ computes the denominator degrees of freedom as described in the section "Degrees of Freedom" on page 6340. Alternatively, you can specify the denominator degrees of freedom with the DF= option in the TABLES statement.

For tables larger than 2×2, PROC SURVEYFREQ also computes the adjusted Wald log-linear F statistic as

$$F_{Adj\_WLL} = \frac{s-k+1}{k \; s} \, Q_{WLL}$$

where $k = (R-1)(C-1)$, and s is the denominator degrees of freedom computed as described in the section "Degrees of Freedom" on page 6340. Alternatively, you can specify the value of s with the DF= option in the TABLES statement. Note that for 2×2 tables, $k = (R-1)(C-1) = 1$, so the adjusted Wald F statistic equals the (unadjusted) Wald F statistic, with the same numerator and denominator degrees of freedom.

Under the null hypothesis, F_{Adj_WLL} approximately follows an F distribution with k numerator degrees of freedom and $(s-k+1)$ denominator degrees of freedom.

Output Data Sets

You can use the Output Delivery System to create a SAS data set from any piece of PROC SURVEYFREQ output. See the section "ODS Table Names" on page 6362 for more information.

PROC SURVEYFREQ also provides an output data set that stores the replicate weights for BRR or jackknife variance estimation and an output data set that stores the jackknife coefficients for jackknife variance estimation.

Replicate Weights Output Data Set

If you specify the OUTWEIGHTS= method-option for VARMETHOD=BRR or JACKKNIFE, PROC SURVEYFREQ stores the replicate weights in an output data set. The OUTWEIGHTS= output data set contains all observations used in the analysis or all valid observations in the DATA= input data set. (A valid observation is an observation that has a positive value of the WEIGHT variable. Valid observations must also have nonmissing values of the STRATA and CLUSTER variables, unless you specify the MISSING option. See the section "Data Summary Table" on page 6356 for details about valid observations.)

The OUTWEIGHTS= data set contains the following variables:

- all variables in the DATA= input data set
- RepWt_1, RepWt_2, . . ., RepWt_n, which are the replicate weight variables

where *n* is the total number of replicates in the analysis. Each replicate weight variable contains the replicate weights for the corresponding replicate. Replicate weights equal zero for those observations not included in the replicate.

After the procedure creates replicate weights for a particular input data set and survey design, you can use the OUTWEIGHTS= method-option to store these replicate weights and then use them again in subsequent analyses, either in PROC SURVEYFREQ or in the other survey procedures. You can use a REPWEIGHTS statement to provide replicate weights for the procedure.

Jackknife Coefficients Output Data Set

If you specify the OUTJKCOEFS= method-option for VARMETHOD=JACKKNIFE, PROC SURVEYFREQ stores the jackknife coefficients in an output data set. The OUTJKCOEFS= output data set contains one observation for each replicate. The OUTJKCOEFS= data set contains the following variables:

- Replicate, which is the replicate number for the jackknife coefficient
- JKCoefficient, which is the jackknife coefficient
- DonorStratum, which is the stratum of the PSU that was deleted to construct the replicate, if you specify a STRATA statement

After the procedure creates jackknife coefficients for a particular input data set and survey design, you can use the OUTJKCOEFS= method-option to store these coefficients and then use them again in subsequent analyses, either in PROC SURVEYFREQ or in the other survey procedures. You can use the JKCOEFS= option in the REPWEIGHTS statement to provide jackknife coefficients for the procedure.

Displayed Output

Data Summary Table

The "Data Summary" table provides information about the input data set and the sample design. PROC SURVEYFREQ displays this table unless you specify the NOSUMMARY option in the PROC SURVEYFREQ statement.

The "Data Summary" table displays the total number of valid observations. To be considered *valid*, an observation must have a nonmissing, positive sampling weight value if you specify a WEIGHT statement. If you do not specify the MISSING option, a valid observation must also have nonmissing values for all STRATA and CLUSTER variables. The number of valid observations can differ from the number of nonmissing observations for an individual table request, which the procedure displays in the frequency or crosstabulation tables. See the section "Missing Values" on page 6323 for more information.

PROC SURVEYFREQ displays the following information in the "Data Summary" table:

- Number of Strata, if you specify a STRATA statement
- Number of Clusters, if you specify a CLUSTER statement
- Number of Observations, which is the total number of valid observations
- Sum of Weights, which is the sum over all valid observations, if you specify a WEIGHT or REPWEIGHTS statement

Stratum Information Table

If you specify the LIST option in the STRATA statement, PROC SURVEYFREQ displays a "Stratum Information" table. This table provides the following information for each stratum:

- Stratum Index, which is a sequential stratum identification number
- STRATA variable(s), which lists the levels of STRATA variables for the stratum
- Number of Observations, which is the number of valid observations in the stratum
- Population Total for the stratum, if you specify the TOTAL= option

- Sampling Rate for the stratum, if you specify the TOTAL= or RATE= option. If you specify the TOTAL= option, the sampling rate is based on the number of valid observations in the stratum.
- Number of Clusters, which is the number of clusters in the stratum, if you specify a CLUSTER statement

Variance Estimation Table

If you specify the VARMETHOD=BRR, VARMETHOD=JACKKNIFE, or NOMCAR option in the PROC SURVEYFREQ statement, the procedure displays a "Variance Estimation" table. If you do not specify any of these options, the procedure creates a "Variance Estimation" table but does not display it. You can store this nondisplayed table in an output data set by using the Output Delivery System (ODS). See the section "ODS Table Names" on page 6362 for more information.

The "Variance Estimation" table provides the following information:

- Method, which is the variance estimation method—Taylor Series, Balanced Repeated Replication, or Jackknife
- Replicate Weights input data set name, if you provide replicate weights with a REPWEIGHTS statement
- Number of Replicates, for VARMETHOD=BRR or VARMETHOD=JACKKNIFE
- Hadamard Data Set name, if you specify the HADAMARD= method-option for VARMETHOD=BRR
- Fay Coefficient, if you specify the FAY method-option for VARMETHOD=BRR
- Missing Levels Included (MISSING), if you specify the MISSING option
- Missing Levels Included (NOMCAR), if you specify the NOMCAR option

Hadamard Matrix

If you specify the PRINTH method-option for VARMETHOD=BRR, PROC SURVEYFREQ displays the Hadamard matrix used to construct replicates for BRR variance estimation. If you provide a Hadamard matrix with the HADAMARD= method-option for VARMETHOD=BRR but the procedure does not use the entire matrix, the procedure displays only the rows and columns that are actually used to construct replicates.

One-Way Frequency Tables

PROC SURVEYFREQ displays one-way frequency tables for all one-way table requests in the TABLES statements, unless you specify the NOPRINT option in the TABLES statement. A one-way table shows the sample frequency distribution of a single variable, and provides estimates for its population distribution in terms of totals and proportions.

If you request a one-way table without specifying options, PROC SURVEYFREQ displays the following information for each level of the variable:

- Frequency count, which is the number of sample observations in the level
- Weighted Frequency, which estimates the population total for the level
- Standard Deviation of Weighted Frequency
- Percent, which estimates the population proportion for the level
- Standard Error of Percent

The one-way table displays weighted frequencies if your analysis includes a WEIGHT or REPWEIGHTS statement, or if you specify the WTFREQ option in the TABLES statement.

The one-way table also displays the Frequency Missing, or the number of observations with missing values.

You can suppress the frequency counts by specifying the NOFREQ option in the TABLES statement. Also, the NOWT option suppresses the weighted frequencies and their standard deviations. The NOPERCENT option suppresses the percentages and their standard errors. The NOSTD option suppresses the standard errors of the percentages and the standard deviations of the weighted frequencies. The NOTOTAL option suppresses the total row of the one-way table.

PROC SURVEYFREQ optionally displays the following information in a one-way table:

- Variance of Weighted Frequency, if you specify the VARWT option
- Confidence Limits for Weighted Frequency, if you specify the CLWT option
- Coefficient of Variation for Weighted Frequency, if you specify the CVWT option
- Test Percent, if you specify the TESTP= option
- Variance of Percent, if you specify the VAR option
- Confidence Limits for Percent, if you specify the CL option
- Coefficient of Variation for Percent, if you specify the CV option
- Design Effect for Percent, if you specify the DEFF option

Crosstabulation Tables

PROC SURVEYFREQ displays all table requests in the TABLES statements, unless you specify the NOPRINT option in the TABLES statement. For two-way to multiway crosstabulation tables, the values of the last variable in the table request form the table columns. The values of the next-to-last variable form the rows. Each level (or combination of levels) of the other variables forms one layer. PROC SURVEYFREQ produces a separate two-way crosstabulation table for each layer of a multiway table.

For each layer, the crosstabulation table displays the row and column variable names and values (or levels). Each two-way table lists levels of the column variable within each level of the row variable.

By default, the procedure displays all levels of the column variable within each level of the row variables, including any column variable levels with zero frequency for that row. For multiway tables, the procedure displays all levels of the row variable for each layer of the table by default, including any row levels with zero frequency for that layer. You can suppress the display of zero frequency levels by specifying the NOSPARSE option.

If you request a crosstabulation table without specifying options, the table displays the following information for each combination of variable levels, or table cell:

- Frequency, which is the number of sample observations in the table cell
- Weighted Frequency, which estimates the population total for the table cell
- Standard Deviation of Weighted Frequency
- Percent, which estimates the population proportion for the table cell
- Standard Error of Percent

The two-way table displays weighted frequencies if your analysis includes a WEIGHT or REPWEIGHTS statement, or if you specify the WTFREQ option in the TABLES statement.

The two-way table also displays the Frequency Missing, or the number of observations with missing values.

You can suppress the frequency counts by specifying the NOFREQ option in the TABLES statement. Also, the NOWT option suppresses the weighted frequencies and their standard deviations. The NOPERCENT option suppresses all percentages and their standard errors. The NOCELLPERCENT option suppresses overall cell percentages and their standard errors, but displays any other percentages (and standard errors) that you request, such as row or column percentages. The NOSTD option suppresses the standard errors of the percentages and the standard deviations of the weighted frequencies. The NOTOTAL option suppresses the row totals and column totals, as well as the overall total.

PROC SURVEYFREQ optionally displays the following information in a two-way table:

- Expected Weighted Frequency, if you specify the EXPECTED option
- Variance of Weighted Frequency, if you specify the VARWT option
- Confidence Limits for Weighted Frequency, if you specify the CLWT option
- Coefficient of Variation for Weighted Frequency, if you specify the CVWT option
- Variance of Percent, if you specify the VAR option
- Confidence Limits for Percent, if you specify the CL option
- Coefficient of Variation for Percent, if you specify the CV option

- Design Effect for Percent, if you specify the DEFF option
- Row Percent, which estimates the population proportion of the row total, if you specify the ROW option
- Standard Error of Row Percent, if you specify the ROW option
- Variance of Row Percent, if you specify the VAR option and the ROW option
- Confidence Limits for Row Percent, if you specify the CL option and the ROW option
- Coefficient of Variation for Row Percent, if you specify the CV option and the ROW option
- Column Percent, which estimates the population proportion of the column total, if you specify the COL option
- Standard Error of Column Percent, if you specify the COL option
- Variance of Column Percent, if you specify the VAR option and the COL option
- Confidence Limits for Column Percent, if you specify the CL option and the COL option
- Coefficient of Variation for Column Percent, if you specify the CV option and the COL option

Statistical Tests

If you specify the CHISQ option for the Rao-Scott chi-square test, the CHISQ1 option for the modified test, the LRCHISQ option for the Rao-Scott likelihood ratio chi-square test, or the LRCHISQ1 option for the modified test, PROC SURVEYFREQ displays the following information:

- Pearson Chi-Square, if you specify the CHISQ or CHISQ1 option
- Likelihood Ratio Chi-Square, if you specify the LRCHISQ or LRCHISQ1 option
- Design Correction
- Rao-Scott Chi-Square, if you specify the CHISQ or CHISQ1 option
- Rao-Scott Likelihood Ratio Chi-Square, if you specify the LRCHISQ or LRCHISQ1 option
- DF, which is the degrees of freedom for the chi-square test
- Pr > ChiSq, which is the *p*-value for the chi-square test
- F Value
- Num DF, which is the numerator degrees of freedom for F
- Den DF, which is the denominator degrees of freedom for F
- Pr > F, which is the *p*-value for the *F* test

If you specify the WCHISQ option for the Wald chi-square test or the WLLCHISQ option for the Wald log-linear chi-square test, PROC SURVEYFREQ displays the following information:

- Wald Chi-Square, if you specify the WCHISQ option
- Wald Log-Linear Chi-Square, if you specify the WLLCHISQ option
- F Value
- Num DF, which is the numerator degrees of freedom for F
- Den DF, which is the denominator degrees of freedom for F
- Pr > F, which is the *p*-value for the *F* test
- Adjusted F Value, for tables larger than 2×2
- Num DF, which is the numerator degrees of freedom for Adjusted F
- Den DF, which is the denominator degrees of freedom for Adjusted F
- Pr > Adj F, which is the *p*-value for the Adjusted *F* test

Risks and Risk Difference

If you specify the RISK option in the TABLES statement for a 2×2 table, PROC SURVEYFREQ displays "Column 1 Risk Estimates" and "Column 2 Risk Estimates" tables. You can display only column 1 or column 2 risks by specifying the RISK1 or RISK2 option, respectively.

The "Risk Estimates" table displays the following information for Row 1, Row 2, Total, and Difference:

- Row, which identifies the risk as Row 1, Row 2, Total, or Difference
- Risk estimate
- Standard Error
- Confidence Limits

In the "Column 1 Risk Estimates" table, the row 1 risk is the column 1 percentage of row 1. The row 2 risk is the column 1 percentage of row 2, and the total risk is the column 1 percentage of the entire table. The risk difference is the row 1 risk minus the row 2 risk. In the "Column 2 Risk Estimates" table, these computations are based on column 2.

Odds Ratio and Relative Risks

If you specify the OR option in the TABLES statement for a 2×2 table, PROC SURVEYFREQ displays the "Odds Ratio" table. This table includes the following information:

- Statistic, which identifies the statistic as the Odds Ratio, the Column 1 Relative Risk, or the Column 2 Relative Risk
- Estimate
- Confidence Limits

ODS Table Names

PROC SURVEYFREQ assigns a name to each table it creates. You can use these names to reference tables when using the Output Delivery System (ODS) to select tables and create output data sets. For more information about ODS, see Chapter 20, "Using the Output Delivery System." Table 83.4 lists the table names, along with the corresponding analysis options.

Table 83.4 ODS Tables Produced by PROC SURVEYFREQ

ODS Table Name	Description	Statement	Option
ChiSq	Chi-square test	TABLES	CHISQ
ChiSq1	Modified chi-square test	TABLES	CHISQ1
CrossTabs	Crosstabulation table	TABLES	(n-way table, $n > 1$)
HadamardMatrix	Hadamard matrix	PROC	VARMETHOD=BRR(PRINTH)
LRChiSq	Likelihood ratio test	TABLES	LRCHISQ
LRChiSq1	Modified likelihood ratio test	TABLES	LRCHISQ1
OddsRatio	Odds ratio and relative risks	TABLES	OR (2×2 table)
OneWay	One-way frequency table	PROC	(with no TABLES stmt)
		or TABLES	(one-way table)
Risk1	Column 1 risk estimates	TABLES	RISK or RISK1 (2×2 table)
Risk2	Column 2 risk estimates	TABLES	RISK or RISK2 (2×2 table)
StrataInfo	Stratum information	STRATA	LIST
Summary	Data summary	PROC	default
TableSummary	Table summary (not displayed)	TABLES	default
VarianceEstimation	Variance estimation	PROC	VARMETHOD=JK | BRR
			or NOMCAR
WChiSq	Wald chi-square test	TABLES	WCHISQ (two-way table)
WLLChiSq	Wald log-linear chi-square test	TABLES	WLLCHISQ (two-way table)

Examples: SURVEYFREQ Procedure

Example 83.1: Two-Way Tables

This example uses the SIS_Survey data set from the section "Getting Started: SURVEYFREQ Procedure" on page 6294. The data set contains results from a customer satisfaction survey for a student information system (SIS).

The following PROC SURVEYFREQ statements request a two-way table for Department by Response and customize the crosstabulation table display:

```
proc surveyfreq data=SIS_Survey;
   tables  Department * Response / cv deff nowt nostd nototal;
   strata  State NewUser / list;
   cluster School;
   weight  SamplingWeight;
run;
```

The TABLES statement requests a two-way table of Department by Response. The CV option requests coefficients of variation for the percentage estimates. The DEFF option requests design effects for the percentage estimates. The NOWT option suppresses display of the weighted frequencies, and the NOSTD option suppresses display of standard errors for the estimates. The NOTOTAL option suppresses the row totals, column totals, and overall totals.

The STRATA, CLUSTER, and WEIGHT statements provide sample design information for the procedure, so that the analysis is done according to the sample design used for the survey. The STRATA statement names the variables State and NewUser, which identify the first-stage strata. The LIST option in the STRATA statement requests a "Stratum Information" table. The CLUSTER statement names the variable School, which identifies the clusters or primary sampling units (PSUs). The WEIGHT statement names the sampling weight variable.

Output 83.1.1 displays the "Data Summary" and "Stratum Information" tables produced by PROC SURVEYFREQ. The "Stratum Information" table lists the six strata in the survey and shows the number of observations and the number of clusters, or schools, in each stratum.

Output 83.1.1 Data Summary and Stratum Information

Student Information System Survey

The SURVEYFREQ Procedure

Data Summary

Number of Strata	6
Number of Clusters	370
Number of Observations	1850
Sum of Weights	38899.6482

Stratum Information

Stratum Index	State	NewUser	Number of Obs	Number of Clusters
1	GA	Renewal Customer	315	63
2	GA	New Customer	355	71
3	NC	Renewal Customer	280	56
4	NC	New Customer	420	84
5	SC	Renewal Customer	210	42
6	SC	New Customer	270	54

Output 83.1.2 displays the two-way table of Department by Response. According to the TABLES statement options specified, this two-way table includes coefficients of variation and design effects for the percentage estimates, and it does not show the weighted frequencies or the standard errors of the estimates. It also does not show the row, column, and overall totals.

Output 83.1.2 Two-Way Table of Department by Response

Table of Department by Response

Department	Response	Frequency	Percent	CV for Percent	Design Effect
Faculty	Very Unsatisfied	209	13.4987	0.0865	2.1586
	Unsatisfied	203	13.0710	0.0868	2.0962
	Neutral	346	22.4127	0.0629	2.1157
	Satisfied	254	16.2006	0.0806	2.3232
	Very Satisfied	98	6.2467	0.1362	2.2842
Admin/Guidance	Very Unsatisfied	95	3.6690	0.1277	1.1477
	Unsatisfied	123	4.6854	0.1060	1.0211
	Neutral	235	9.1838	0.0700	0.9166
	Satisfied	201	7.7305	0.0756	0.8848
	Very Satisfied	86	3.3016	0.1252	0.9892

The following PROC SURVEYFREQ statements request a two-way table of Department by Response that includes row percentages, and also a Wald chi-square test of association between the two table variables:

```
proc surveyfreq data=SIS_Survey nosummary;
   tables Department * Response / row nowt wchisq;
   strata State NewUser;
   cluster School;
   weight SamplingWeight;
run;
```

Output 83.1.3 displays the two-way table. The row percentages show the distribution of Response for Department = 'Faculty' and for Department = 'Admin/Guidance'. This is equivalent to a domain or subpopulation analysis of Response, where the domains are Department = 'Faculty' and Department = 'Admin/Guidance'.

Output 83.1.4 displays the Wald chi-square test of association between Department and Response. The Wald chi-square is 11.44, and the corresponding adjusted F value is 2.84 with a p-value of 0.0243. This indicates a significant association between department (faculty or admin/guidance) and satisfaction with the student information system.

Output 83.1.3 Table of Department by Response with Row Percentages

Student Information System Survey

The SURVEYFREQ Procedure

Table of Department by Response

Department	Response	Frequency	Percent	Std Err of Percent	Row Percent	Std Err of Row Percent
Faculty	Very Unsatisfied	209	13.4987	1.1675	18.8979	1.6326
	Unsatisfied	203	13.0710	1.1350	18.2992	1.5897
	Neutral	346	22.4127	1.4106	31.3773	1.9705
	Satisfied	254	16.2006	1.3061	22.6805	1.8287
	Very Satisfied	98	6.2467	0.8506	8.7452	1.1918
	Total	1110	71.4297	0.1468	100.000	
Admin/Guidance	Very Unsatisfied	95	3.6690	0.4684	12.8419	1.6374
	Unsatisfied	123	4.6854	0.4966	16.3995	1.7446
	Neutral	235	9.1838	0.6430	32.1447	2.2300
	Satisfied	201	7.7305	0.5842	27.0579	2.0406
	Very Satisfied	86	3.3016	0.4133	11.5560	1.4466
	Total	740	28.5703	0.1468	100.000	
Total	Very Unsatisfied	304	17.1676	1.2872		
	Unsatisfied	326	17.7564	1.2712		
	Neutral	581	31.5965	1.5795		
	Satisfied	455	23.9311	1.4761		
	Very Satisfied	184	9.5483	0.9523		
	Total	1850	100.000			

Output 83.1.4 Wald Chi-Square Test

Wald Chi-Square Test

Chi-Square	11.4454
F Value	2.8613
Num DF	4
Den DF	364
Pr > F	0.0234
Adj F Value	2.8378
Num DF	4
Den DF	361
Pr > Adj F	0.0243

Sample Size = 1850

Example 83.2: Multiway Tables (Domain Analysis)

Continuing to use the SIS_Survey data set from the section "Getting Started: SURVEYFREQ Procedure" on page 6294, this example shows how to produce multiway tables. The following PROC SURVEYFREQ statements request a table of Department by SchoolType by Response for the student information system survey:

```
proc surveyfreq data=SIS_Survey;
   tables  Department * SchoolType * Response
           SchoolType * Response;
   strata  State NewUser;
   cluster School;
   weight  SamplingWeight;
run;
```

The TABLES statement requests a multiway table with SchoolType as the row variable, Response as the column variable, and Department as the layer variable. This request produces a separate two-way table of SchoolType by Response for each level of the variable Department. The TABLES statement also requests a two-way table of SchoolType by Response, which totals the multiway table over both levels of Department. As in the previous examples, the STRATA, CLUSTER, and WEIGHT statements provide sample design information, so that the analysis will be done according to the design used for this survey.

Output 83.2.1 displays the multiway table produced by PROC SURVEYFREQ, which includes a table of SchoolType by Response for Department = 'Faculty' and for Department = 'Admin/Guidance'. This is equivalent to a domain or subpopulation analysis of SchoolType by Response, where the domains are Department = 'Faculty' and Department = 'Admin/Guidance'.

Output 83.2.1 Multiway Table of Department by SchoolType by Response

Student Information System Survey

The SURVEYFREQ Procedure

Table of SchoolType by Response
Controlling for Department=Faculty

SchoolType	Response	Frequency	Weighted Frequency	Std Dev of Wgt Freq	Percent	Std Err of Percent
Middle School	Very Unsatisfied	74	1846	301.22637	6.6443	1.0838
	Unsatisfied	78	1929	283.11476	6.9428	1.0201
	Neutral	130	3289	407.80855	11.8369	1.4652
	Satisfied	113	2795	368.85087	10.0597	1.3288
	Very Satisfied	55	1378	261.63311	4.9578	0.9411
	Total	450	11237	714.97120	40.4415	2.5713
High School	Very Unsatisfied	135	3405	389.42313	12.2536	1.3987
	Unsatisfied	125	3155	384.56734	11.3563	1.3809
	Neutral	216	5429	489.37826	19.5404	1.7564
	Satisfied	141	3507	417.54773	12.6208	1.5040
	Very Satisfied	43	1052	221.59367	3.7874	0.7984
	Total	660	16549	719.61536	59.5585	2.5713
Total	Very Unsatisfied	209	5251	454.82598	18.8979	1.6326
	Unsatisfied	203	5085	442.39032	18.2992	1.5897
	Neutral	346	8718	550.81735	31.3773	1.9705
	Satisfied	254	6302	507.01711	22.6805	1.8287
	Very Satisfied	98	2430	330.97602	8.7452	1.1918
	Total	1110	27786	119.25529	100.000	

Output 83.2.1 *continued*

Table of SchoolType by Response
Controlling for Department=Admin/Guidance

SchoolType	Response	Frequency	Weighted Frequency	Std Dev of Wgt Freq	Percent	Std Err of Percent
Middle School	Very Unsatisfied	42	649.43427	133.06194	5.8435	1.1947
	Unsatisfied	31	460.35557	100.80158	4.1422	0.9076
	Neutral	104	1568	186.99946	14.1042	1.6804
	Satisfied	84	1269	165.71127	11.4142	1.4896
	Very Satisfied	39	574.93878	110.37243	5.1732	0.9942
	Total	300	4521	287.86832	40.6774	2.5801
High School	Very Unsatisfied	53	777.77725	136.41869	6.9983	1.2285
	Unsatisfied	92	1362	175.40862	12.2573	1.5806
	Neutral	131	2005	212.34804	18.0404	1.8990
	Satisfied	117	1739	190.07798	15.6437	1.7118
	Very Satisfied	47	709.37033	126.54394	6.3828	1.1371
	Total	440	6593	288.92483	59.3226	2.5801
Total	Very Unsatisfied	95	1427	182.28132	12.8419	1.6374
	Unsatisfied	123	1823	193.43045	16.3995	1.7446
	Neutral	235	3572	250.22739	32.1447	2.2300
	Satisfied	201	3007	226.82311	27.0579	2.0406
	Very Satisfied	86	1284	160.83434	11.5560	1.4466
	Total	740	11114	60.78850	100.000	

Example 83.3: Output Data Sets

PROC SURVEYFREQ uses the Output Delivery System (ODS) to create output data sets. This is a departure from older SAS procedures that provide OUTPUT statements for similar functionality. By using ODS, you can create a SAS data set from any piece of PROC SURVEYFREQ output. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

When selecting tables for ODS output data sets, you reference tables by their ODS table names. Each table created by PROC SURVEYFREQ is assigned a name. See the section "ODS Table Names" on page 6362 for a list of the table names provided by PROC SURVEYFREQ.

To save the one-way table of Response from Figure 83.3 in an output data set, use an ODS OUTPUT statement as follows:

```
proc surveyfreq data=SIS_Survey;
   tables  Response / cl nowt;
   ods output OneWay=ResponseTable;
   strata  State NewUser;
   cluster School;
   weight  SamplingWeight;
run;
```

Output 83.3.1 displays the output data set ResponseTable, which contains the one-way table of Response. This data set has six observations, and each of these observations corresponds to a row of the one-way table. The first five observations correspond to the five levels of Response, as they are ordered in the one-way table display, and the last observation corresponds to the overall total, which is the last row of the one-way table. The data set ResponseTable includes a variable corresponding to each column of the one-way table. For example, the variable Percent contains the percentage estimates, and the variables LowerCL and UpperCL contain the lower and upper confidence limits for the percentage estimates.

Output 83.3.1 ResponseTable Output Data Set

Obs	Table	Response	Frequency	Percent	StdErr	LowerCL	UpperCL
1	Table Response	Very Unsatisfied	304	17.1676	1.2872	14.6364	19.6989
2	Table Response	Unsatisfied	326	17.7564	1.2712	15.2566	20.2562
3	Table Response	Neutral	581	31.5965	1.5795	28.4904	34.7026
4	Table Response	Satisfied	455	23.9311	1.4761	21.0285	26.8338
5	Table Response	Very Satisfied	184	9.5483	0.9523	7.6756	11.4210
6	Table Response	.	1850	100.000	_	_	_

PROC SURVEYFREQ also creates a table summary that is not displayed. Some of the information in this table is similar to that contained in the "Data Summary" table, but the table summary describes the data used to analyze the specified table, while the data summary describes the entire input data set. Due to missing values, for example, the number of observations (or strata or clusters) used to analyze a particular table can differ from the number of observations (or strata or clusters) reported for the input data set in the "Data Summary" table. See the section "Missing Values" on page 6323 for more details. If you request confidence limits, the "Table Summary" table also contains the degrees of freedom and the t-value used to compute the confidence limits.

The following statements store the nondisplayed "Table Summary" table in the output data set ResponseSummary:

```
proc surveyfreq data=SIS_Survey;
   tables  Response / cl nowt;
   ods output TableSummary=ResponseSummary;
   strata  State NewUser;
   cluster School;
   weight  SamplingWeight;
run;
```

Output 83.3.2 displays the output data set ResponseSummary.

Output 83.3.2 ResponseSummary Output Data Set

Obs	Table	Number of Observations	Number of Strata	Number of Clusters	Degrees of Freedom	t Percentile
1	Table Response	1850	6	370	364	1.966503

References

Agresti, A. (2002), *Categorical Data Analysis*, Second Edition, New York: John Wiley & Sons.

Agresti, A. (2007), *An Introduction to Categorical Data Analysis*, Second Edition, New York: John Wiley & Sons.

Bedrick, E. J. (1983), "Adjusted Chi-Squared Tests for Cross-Classified Tables of Survey Data," *Biometrika*, 70, 591–596.

Brick, J. M. and Kalton, G. (1996), "Handling Missing Data in Survey Research," *Statistical Methods in Medical Research*, 5, 215–238.

Cochran, W. G. (1977), *Sampling Techniques*, Third Edition, New York: John Wiley & Sons.

Dippo, C. S., Fay, R. E., and Morganstein, D. H. (1984), "Computing Variances from Complex Samples with Replicate Weights," *Proceedings of the Survey Research Methods Section, ASA*, 489–494.

Fay, R. E. (1989), "Theory and Application of Replicate Weighting for Variance Calculations," *Proceedings of the Survey Research Methods Section, ASA*, 212–217.

Felligi, I. P. (1980), "Approximate Tests of Independence and Goodness of Fit Based on Stratified Multistage Samples," *Journal of the American Statistical Association*, 75, 261–268.

Fienberg, S. E. (1980), *The Analysis of Cross-Classified Data*, Second Edition, Cambridge, MA: MIT Press.

Fleiss, J. L. (1981), *Statistical Methods for Rates and Proportions*, Second Edition, New York: John Wiley & Sons.

Fuller, W. A. (1975), "Regression Analysis for Sample Survey," *Sankhyā*, 37 (3), Series C, 117–132.

Fuller, W. A., Kennedy, W., Schnell, D., Sullivan, G., and Park, H. J. (1989), *PC CARP*, Ames: Statistical Laboratory, Iowa State University.

Hansen, M. H., Hurwitz, W. N., and Madow, W. G. (1953), *Sample Survey Methods and Theory*, Volumes I and II, New York: John Wiley & Sons.

Hidiroglou, M. A., Fuller, W. A., and Hickman, R. D. (1980), *SUPER CARP*, Ames: Statistical Laboratory, Iowa State University.

Judkins, D. (1990), "Fay's Method for Variance Estimation," *Journal of Official Statistics*, 6, 223–239.

Kalton, G. (1983), *Introduction to Survey Sampling*, Sage University Paper series on Quantitative Applications in the Social Sciences, series no. 07-035, Beverly Hills, CA, and London: Sage Publications.

Kalton, G. and Kaspyzyk, D. (1986), "The Treatment of Missing Survey Data," *Survey Methodol-*

ogy, 12, 1–16.

Kish, L. (1965), *Survey Sampling*, New York: John Wiley & Sons.

Koch, G. G., Freeman, D. H., and Freeman, J. L. (1975), "Strategies in the Multivariate Analysis of Data from Complex Surveys," *International Statistical Review*, 43, 59–78.

Koch, G. G., Landis, J. R., Freeman, D. H., Freeman, J. L., and Lehnen, R. G. (1977), "A General Methodology for the Analysis of Experiments with Repeated Measurement of Categorical Data," *Biometrics*, 33, 133–158.

Korn, E. L. and Graubard, B. I. (1990), "Simultaneous Testing with Complex Survey Data: Use of Bonferroni *t*-Statistics," *The American Statistician*, 44, 270–276.

Lee, E. S., Forthoffer, R. N., and Lorimor, R. J. (1989), *Analyzing Complex Survey Data*, Sage University Paper series on Quantitative Applications in the Social Sciences, series no. 07-071, Beverly Hills, CA, and London: Sage Publications.

Levy, P. and Lemeshow, S. (1999), *Sampling of Populations, Methods and Applications*, Third Edition, New York: John Wiley & Sons.

Lohr, S. L. (1999), *Sampling: Design and Analysis*, Pacific Grove, CA: Duxbury Press.

Nathan, G. (1975), "Tests for Independence in Contingency Tables from Stratified Samples," *Sankhyā*, 37, Series C, 77–87.

Rao, J. N. K. and Scott, A. J. (1979), "Chi-Squared Tests for Analysis of Categorical Data from Complex Surveys," *Proceedings of the Survey Research Methods Section, ASA*, 58–66.

Rao, J. N. K. and Scott, A. J. (1981), "The Analysis of Categorical Data from Complex Surveys: Chi-Squared Tests for Goodness of Fit and Independence in Two-Way Tables," *Journal of the American Statistical Association*, 76, 221–230.

Rao, J. N. K. and Scott, A. J. (1984), "On Chi-Squared Tests for Multiway Contingency Tables with Cell Properties Estimated from Survey Data," *The Annals of Statistics*, 12, 46–60.

Rao, J. N. K. and Scott, A. J. (1987), "On Simple Adjustments to Chi-Square Tests with Survey Data," *The Annals of Statistics*, 15, 385–397.

Rao, J. N. K. and Shao, J. (1996), "On Balanced Half Sample Variance Estimation in Stratified Sampling," *Journal of the American Statistical Association*, 91, 343–348.

Rao, J. N. K. and Shao, J. (1999), "Modified Balanced Repeated Replication for Complex Survey Data," *Biometrika,* 86, 403–415.

Särndal, C. E., Swensson, B., and Wretman, J. (1992), *Model Assisted Survey Sampling*, New York: Springer-Verlag.

Satterthwaite, F. E. (1946), "An Approximate Distribution of Estimates of Variance Components," *Biometrics*, 2, 110–114.

Stokes, M. E., Davis, C. S., and Koch, G. G. (2000), *Categorical Data Analysis Using the SAS*

System, Second Edition, Cary, NC: SAS Institute Inc.

Thomas, D. R., and Rao, J. N. K. (1984), "A Monte Carlo Study of Exact Levels of Goodness-of-Fit Statistics under Cluster Sampling," *Proceedings of the Survey Research Methods Section, ASA*, 207–211.

Thomas, D. R., and Rao, J. N. K. (1985), "On the Power of Some Goodness-of-Fit Tests under Cluster Sampling," *Proceedings of the Survey Research Methods Section, ASA*, 291–296.

Thomas, D. R., Singh, A. C., and Roberts, G. R. (1996), "Tests of Independence on Two-Way Tables under Cluster Sampling: An Evaluation," *International Statistical Review*, 64, 295–311.

Wald, A. (1943), "Tests of Statistical Hypotheses Concerning Several Parameters When the Number of Observations Is Large," *Transactions of the American Mathematical Society*, 54, 426–482.

Wolter, K. M. (1985), *Introduction to Variance Estimation*, New York: Springer-Verlag.

Woodruff, R. S. (1971), "A Simple Method for Approximating the Variance of a Complicated Estimate," *Journal of the American Statistical Association,* 66, 411–414.

Chapter 84

The SURVEYLOGISTIC Procedure

Contents

Overview: SURVEYLOGISTIC Procedure **6375**
Getting Started: SURVEYLOGISTIC Procedure **6377**
Syntax: SURVEYLOGISTIC Procedure **6382**
PROC SURVEYLOGISTIC Statement 6383
BY Statement 6388
CLASS Statement 6389
CLUSTER Statement 6391
CONTRAST Statement 6391
DOMAIN Statement 6394
FREQ Statement 6395
MODEL Statement 6395
Response Variable Options 6396
Model Options 6397
OUTPUT Statement 6402
Details of the PREDPROBS= Option 6404
REPWEIGHTS Statement 6406
STRATA Statement 6407
TEST Statement 6408
UNITS Statement 6409
WEIGHT Statement 6409
Details: SURVEYLOGISTIC Procedure **6410**
Missing Values 6410
Model Specification 6411
Response Level Ordering 6411
CLASS Variable Parameterization 6412
Link Functions and the Corresponding Distributions 6415
Model Fitting 6416
Determining Observations for Likelihood Contributions 6416
Iterative Algorithms for Model Fitting 6417
Convergence Criteria 6418
Existence of Maximum Likelihood Estimates 6418
Model Fitting Statistics 6420
Generalized Coefficient of Determination 6421
INEST= Data Set 6421

Survey Design Information 6422
Specification of Population Totals and Sampling Rates 6422
Primary Sampling Units (PSUs) 6423
Logistic Regression Models and Parameters 6423
Notation 6423
Logistic Regression Models 6424
Likelihood Function 6426
Variance Estimation 6427
Taylor Series (Linearization) 6427
Balanced Repeated Replication (BRR) Method 6429
Fay's BRR Method 6430
Jackknife Method 6431
Hadamard Matrix 6432
Domain Analysis 6432
Hypothesis Testing and Estimation 6433
Score Statistics and Tests 6433
Testing the Parallel Lines Assumption 6433
Wald Confidence Intervals for Parameters 6434
Testing Linear Hypotheses about the Regression Coefficients 6434
Odds Ratio Estimation 6434
Rank Correlation of Observed Responses and Predicted Probabilities 6437
Linear Predictor, Predicted Probability, and Confidence Limits 6438
Cumulative Response Models 6438
Generalized Logit Model 6439
Output Data Sets 6439
OUT= Data Set in the OUTPUT Statement 6440
Replicate Weights Output Data Set 6440
Jackknife Coefficients Output Data Set 6441
Displayed Output 6441
Model Information 6441
Variance Estimation 6442
Data Summary 6443
Response Profile 6443
Class Level Information 6443
Stratum Information 6444
Maximum Likelihood Iteration History 6444
Score Test 6444
Model Fit Statistics 6444
Type 3 Analysis of Effects 6445
Analysis of Maximum Likelihood Estimates 6445
Odds Ratio Estimates 6446
Association of Predicted Probabilities and Observed Responses . . 6446
Wald Confidence Interval for Parameters 6446
Wald Confidence Interval for Odds Ratios 6446

Estimated Covariance Matrix 6446
Linear Hypotheses Testing Results 6447
Hadamard Matrix 6447
ODS Table Names 6447
Examples: SURVEYLOGISTIC Procedure **6448**
Example 84.1: Stratified Cluster Sampling 6448
Example 84.2: The Medical Expenditure Panel Survey (MEPS) 6455
References **6462**

Overview: SURVEYLOGISTIC Procedure

Categorical responses arise extensively in sample survey. Common examples of responses include the following:

- binary: for example, attended graduate school or not
- ordinal: for example, mild, moderate, and severe pain
- nominal: for example, ABC, NBC, CBS, FOX TV network viewed at a certain hour

Logistic regression analysis is often used to investigate the relationship between such discrete responses and a set of explanatory variables. See Binder (1981, 1983); Roberts, Rao, and Kumar (1987); Skinner, Holt, and Smith (1989); Morel (1989); and Lehtonen and Pahkinen (1995) for description of logistic regression for sample survey data.

For binary response models, the response of a sampling unit can take a specified value or not (for example, attended graduate school or not). Suppose $\mathbf{x}$ is a row vector of explanatory variables and π is the response probability to be modeled. The linear logistic model has the form

$$\text{logit}(\pi) \equiv \log\left(\frac{\pi}{1-\pi}\right) = \alpha + \mathbf{x}\boldsymbol{\beta}$$

where α is the intercept parameter and $\boldsymbol{\beta}$ is the vector of slope parameters.

The logistic model shares a common feature with the more general class of generalized linear models—namely, that a function $g = g(\mu)$ of the expected value, μ, of the response variable is assumed to be linearly related to the explanatory variables. Since μ implicitly depends on the stochastic behavior of the response, and since the explanatory variables are assumed to be fixed, the function g provides the link between the random (stochastic) component and the systematic (deterministic) component of the response variable. For this reason, Nelder and Wedderburn (1972) refer to $g(\cdot)$ as a link function. One advantage of the logit function over other link functions is that differences on the logistic scale are interpretable regardless of whether the data are sampled prospectively or retrospectively (McCullagh and Nelder 1989, Chapter 4). Other link functions that are widely

used in practice are the probit function and the complementary log-log function. The SURVEYLOGISTIC procedure enables you to choose one of these link functions, resulting in fitting a broad class of binary response models of the form

$$g(\pi) = \alpha + \mathbf{x}\boldsymbol{\beta}$$

For ordinal response models, the response Y of an individual or an experimental unit might be restricted to one of a usually small number of ordinal values, denoted for convenience by $1, \ldots, D, D+1$ ($D \geq 1$). For example, pain severity can be classified into three response categories as 1=mild, 2=moderate, and 3=severe. The SURVEYLOGISTIC procedure fits a common slopes cumulative model, which is a parallel lines regression model based on the cumulative probabilities of the response categories rather than on their individual probabilities. The cumulative model has the form

$$g(\Pr(Y \leq d \mid \mathbf{x})) = \alpha_d + \mathbf{x}\boldsymbol{\beta}, \quad 1 \leq d \leq D$$

where $\alpha_1, \ldots, \alpha_k$ are k intercept parameters and $\boldsymbol{\beta}$ is the vector of slope parameters. This model has been considered by many researchers. Aitchison and Silvey (1957) and Ashford (1959) employ a probit scale and provide a maximum likelihood analysis; Walker and Duncan (1967) and Cox and Snell (1989) discuss the use of the log-odds scale. For the log-odds scale, the cumulative logit model is often referred to as the *proportional odds* model.

For nominal response logistic models, where the $D+1$ possible responses have no natural ordering, the logit model can also be extended to a *generalized logit* model, which has the form

$$\log\left(\frac{\Pr(Y = i \mid \mathbf{x})}{\Pr(Y = D+1 \mid \mathbf{x})}\right) = \alpha_i + \mathbf{x}\boldsymbol{\beta}_i, \quad i = 1, \ldots, D$$

where the $\alpha_1, \ldots, \alpha_D$ are D intercept parameters and the $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_D$ are D vectors of parameters. These models were introduced by McFadden (1974) as the *discrete choice* model, and they are also known as *multinomial* models.

The SURVEYLOGISTIC procedure fits linear logistic regression models for discrete response survey data by the method of maximum likelihood. For statistical inferences, PROC SURVEYLOGISTIC incorporates complex survey sample designs, including designs with stratification, clustering, and unequal weighting.

The maximum likelihood estimation is carried out with either the Fisher scoring algorithm or the Newton-Raphson algorithm. You can specify starting values for the parameter estimates. The logit link function in the ordinal logistic regression models can be replaced by the probit function or the complementary log-log function.

Odds ratio estimates are displayed along with parameter estimates. You can also specify the change in the explanatory variables for which odds ratio estimates are desired.

Variances of the regression parameters and odds ratios are computed by using either the Taylor series (linearization) method or replication (resampling) methods to estimate sampling errors of estimators based on complex sample designs (Binder 1983; Särndal, Swensson, and Wretman 1992, Wolter 1985; Rao, Wu, and Yue 1992).

The SURVEYLOGISTIC procedure enables you to specify categorical variables (also known as CLASS variables) as explanatory variables. It also enables you to specify interaction terms in the same way as in the LOGISTIC procedure.

Like many procedures in SAS/STAT software that allow the specification of CLASS variables, the SURVEYLOGISTIC procedure provides a CONTRAST statement for specifying customized hypothesis tests concerning the model parameters. The CONTRAST statement also provides estimation of individual rows of contrasts, which is particularly useful for obtaining odds ratio estimates for various levels of the CLASS variables.

Getting Started: SURVEYLOGISTIC Procedure

The SURVEYLOGISTIC procedure is similar to the LOGISTIC procedure and other regression procedures in the SAS System. See Chapter 51, "The LOGISTIC Procedure," for general information about how to perform logistic regression by using SAS. PROC SURVEYLOGISTIC is designed to handle sample survey data, and thus it incorporates the sample design information into the analysis.

The following example illustrates how to use PROC SURVEYLOGISTIC to perform logistic regression for sample survey data.

In the customer satisfaction survey example in the section "Getting Started: SURVEYSELECT Procedure" on page 6615 of Chapter 87, "The SURVEYSELECT Procedure," an Internet service provider conducts a customer satisfaction survey. The survey population consists of the company's current subscribers from four states: Alabama (AL), Florida (FL), Georgia (GA), and South Carolina (SC). The company plans to select a sample of customers from this population, interview the selected customers and ask their opinions on customer service, and then make inferences about the entire population of subscribers from the sample data. A stratified sample is selected by using the probability proportional to size (PPS) method. The sample design divides the customers into strata depending on their types ('Old' or 'New') and their states (AL, FL, GA, SC). There are eight strata in all. Within each stratum, customers are selected and interviewed by using the PPS with replacement method, where the size variable is Usage. The stratified PPS sample contains 192 customers. The data are stored in the SAS data set SampleStrata. Figure 84.1 displays the first 10 observations of this data set.

Figure 84.1 Stratified PPS Sample (First 10 Observations)

Customer Satisfaction Survey
Stratified PPS Sampling
(First 10 Observations)

Obs	State	Type	Customer ID	Rating	Usage	Sampling Weight
1	AL	New	2178037	Unsatisfied	23.53	14.7473
2	AL	New	75375074	Unsatisfied	99.11	3.5012
3	AL	New	116722913	Satisfied	31.11	11.1546
4	AL	New	133059995	Neutral	52.70	19.7542
5	AL	New	216784622	Satisfied	8.86	39.1613
6	AL	New	225046040	Neutral	8.32	41.6960
7	AL	New	238463776	Satisfied	4.63	74.9483
8	AL	New	255918199	Unsatisfied	10.05	34.5405
9	AL	New	395767821	Extremely Unsatisfied	33.14	10.4719
10	AL	New	409095328	Satisfied	10.67	32.5295

In the SAS data set SampleStrata, the variable CustomerID uniquely identifies each customer. The variable State contains the state of the customer's address. The variable Type equals 'Old' if the customer has subscribed to the service for more than one year; otherwise, the variable Type equals 'New'. The variable Usage contains the customer's average monthly service usage, in hours. The variable Rating contains the customer's responses to the survey. The sample design uses an unequal probability sampling method, with the sampling weights stored in the variable SamplingWeight.

The following SAS statements fit a cumulative logistic model between the satisfaction levels and the Internet usage by using the stratified PPS sample.

```
title 'Customer Satisfaction Survey';
proc surveylogistic data=SampleStrata;
strata state type/list;
model Rating (order=internal) = Usage;
weight SamplingWeight;
run;
```

The PROC SURVEYLOGISTIC statement invokes the SURVEYLOGISTIC procedure. The STRATA statement specifies the stratification variables State and Type that are used in the sample design. The LIST option requests a summary of the stratification. In the MODEL statement, Rating is the response variable and Usage is the explanatory variable. The ORDER=internal is used for the response variable Rating to ask the procedure to order the response levels by using the internal numerical value (1–5) instead of the formatted character value. The WEIGHT statement specifies the variable SamplingWeight that contains the sampling weights.

The results of this analysis are shown in the following figures.

Figure 84.2 Stratified PPS Sample, Model Information

Customer Satisfaction Survey

The SURVEYLOGISTIC Procedure

Model Information

Data Set	WORK.SAMPLESTRATA	
Response Variable	Rating	
Number of Response Levels	5	
Stratum Variables	State	
	Type	
Number of Strata	8	
Weight Variable	SamplingWeight	Sampling Weight
Model	Cumulative Logit	
Optimization Technique	Fisher's Scoring	
Variance Adjustment	Degrees of Freedom (DF)	

PROC SURVEYLOGISTIC first lists the following model fitting information and sample design information in Figure 84.2:

- The link function is the logit of the cumulative of the lower response categories.
- The Fisher scoring optimization technique is used to obtain the maximum likelihood estimates for the regression coefficients.
- The response variable is Rating, which has five response levels.
- The stratification variables are State and Type.
- There are eight strata in the sample.
- The weight variable is SamplingWeight.
- The variance adjustment method used for the regression coefficients is the default degrees of freedom adjustment.

Figure 84.3 lists the number of observations in the data set and the number of observations used in the analysis. Since there is no missing value in this example, observations in the entire data set are used in the analysis. The sums of weights are also reported in this table.

Figure 84.3 Stratified PPS Sample, Number of Observations

Number of Observations Read	192
Number of Observations Used	192
Sum of Weights Read	13262.74
Sum of Weights Used	13262.74

The "Response Profile" table in Figure 84.4 lists the five response levels, their ordered values, and

their total frequencies and total weights for each category. Due to the ORDER=INTERNAL option for the response variable Rating, the category "Extremely Unsatisfied" has the Ordered Value 1, the category "Unsatisfied" has the Ordered Value 2, and so on.

Figure 84.4 Stratified PPS Sample, Response Profile

Response Profile

Ordered Value	Rating	Total Frequency	Total Weight
1	Extremely Unsatisfied	52	2067.1092
2	Unsatisfied	47	2148.7127
3	Neutral	47	3649.4869
4	Satisfied	38	2533.5379
5	Extremely Satisfied	8	2863.8888

Probabilities modeled are cumulated over the lower Ordered Values.

Figure 84.5 displays the output of the stratification summary. There are a total of eight strata, and each stratum is defined by the customer types within each state. The table also shows the number of customers within each stratum.

Figure 84.5 Stratified PPS Sample, Stratification Summary

Stratum Information

Stratum Index	State	Type	N Obs
1	AL	New	22
2		Old	24
3	FL	New	25
4		Old	22
5	GA	New	25
6		Old	25
7	SC	New	24
8		Old	25

Figure 84.6 shows the chi-square test for testing the proportional odds assumption. The test is highly significant, which indicates that the cumulative logit model might not adequately fit the data.

Figure 84.6 Stratified PPS Sample, Testing the Proportional Odds Assumption

Score Test for the Proportional Odds Assumption

Chi-Square	DF	Pr > ChiSq
911.1244	3	<.0001

Figure 84.7 shows the iteration algorithm converged to obtain the MLE for this example. The "Model Fit Statistics" table contains the Akaike information criterion (AIC), the Schwarz criterion (SC), and the negative of twice the log likelihood ($-2 \log L$) for the intercept-only model and the fitted model. AIC and SC can be used to compare different models, and the ones with smaller values are preferred.

Figure 84.7 Stratified PPS Sample, Model Fitting Information

Model Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	42099.954	41378.851
SC	42112.984	41395.139
-2 Log L	42091.954	41368.851

The table "Testing Global Null Hypothesis: BETA=0" in Figure 84.8 shows the likelihood ratio test, the efficient score test, and the Wald test for testing the significance of the explanatory variable (Usage). All tests are significant.

Figure 84.8 Stratified PPS Sample

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	723.1023	1	<.0001
Score	465.4939	1	<.0001
Wald	4.5212	1	0.0335

Figure 84.9 shows the parameter estimates of the logistic regression and their standard errors.

Figure 84.9 Stratified PPS Sample, Parameter Estimates

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept Extremely Unsatisfied	1	-2.0168	0.3988	25.5769	<.0001
Intercept Unsatisfied	1	-1.0527	0.3543	8.8292	0.0030
Intercept Neutral	1	0.1334	0.4189	0.1015	0.7501
Intercept Satisfied	1	1.0751	0.5794	3.4432	0.0635
Usage	1	0.0377	0.0178	4.5212	0.0335

Figure 84.10 displays the odds ratio estimate and its confidence limits.

Figure 84.10 Stratified PPS Sample, Odds Ratios

Odds Ratio Estimates			
Effect	Point Estimate	95% Wald Confidence Limits	
Usage	1.038	1.003	1.075

Syntax: SURVEYLOGISTIC Procedure

The following statements are available in PROC SURVEYLOGISTIC:

PROC SURVEYLOGISTIC < *options* > ;
BY *variables* ;
CLASS *variable* < *(v-options)* > < *variable* < *(v-options)* > ... > < / *v-options* > ;
CLUSTER *variables* ;
CONTRAST *'label'* *effect values* < *,... effect values* > < / *options* > ;
DOMAIN *variables* < *variable*∗*variable variable*∗*variable*∗*variable ...* > ;
FREQ *variable* ;
MODEL *events/trials* = < *effects* < / *options* > > ;
MODEL *variable* < *(v-options)* > = < *effects* > < / *options* > ;
OUTPUT < **OUT=***SAS-data-set* > < *options* > ;
REPWEIGHTS *variables* < / *options* > ;
STRATA *variables* < / *option* > ;
< *label:* > **TEST** *equation1* < , ... , *equationk* > < / *options* > ;
UNITS *independent1* = *list1* < ... *independentk* = *listk* > < / *options* > ;
WEIGHT *variable* < / *option* > ;

The PROC SURVEYLOGISTIC and MODEL statements are required. The CLASS, CLUSTER, STRATA, and CONTRAST statements can appear multiple times. You should use only one MODEL statement and one WEIGHT statement. The CLASS statement (if used) must precede the MODEL statement, and the CONTRAST statement (if used) must follow the MODEL statement.

The rest of this section provides detailed syntax information for each of the preceding statements, beginning with the PROC SURVEYLOGISTIC statement. The remaining statements are covered in alphabetical order.

PROC SURVEYLOGISTIC Statement

PROC SURVEYLOGISTIC < *options* > **;**

The PROC SURVEYLOGISTIC statement invokes the SURVEYLOGISTIC procedure and optionally identifies input data sets, controls the ordering of the response levels, and specifies the variance estimation method. The PROC SURVEYLOGISTIC statement is required.

ALPHA=*value*

sets the confidence level for confidence limits. The value of the ALPHA= option must be between 0 and 1, and the default value is 0.05. A confidence level of α produces $100(1-\alpha)\%$ confidence limits. The default of ALPHA=0.05 produces 95% confidence limits.

DATA=*SAS-data-set*

names the SAS data set containing the data to be analyzed. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

INEST=*SAS-data-set*

names the SAS data set that contains initial estimates for all the parameters in the model. BY-group processing is allowed in setting up the INEST= data set. See the section "INEST= Data Set" on page 6421 for more information.

MISSING

treats missing values as a valid (nonmissing) category for all categorical variables, which include CLASS, STRATA, CLUSTER, and DOMAIN variables.

By default, if you do not specify the MISSING option, an observation is excluded from the analysis if it has a missing value. For more information, see the section "Missing Values" on page 6410.

NAMELEN=*n*

specifies the length of effect names in tables and output data sets to be n characters, where n is a value between 20 and 200. The default length is 20 characters.

NOMCAR

requests that the procedure treat missing values in the variance computation as *not missing completely at random* (NOMCAR) for Taylor series variance estimation. When you specify the NOMCAR option, PROC SURVEYLOGISTIC computes variance estimates by analyzing the nonmissing values as a domain or subpopulation, where the entire population includes both nonmissing and missing domains. See the section "Missing Values" on page 6410 for more details.

By default, PROC SURVEYLOGISTIC completely excludes an observation from analysis if that observation has a missing value, unless you specify the MISSING option. Note that the NOMCAR option has no effect on a classification variable when you specify the MISSING option, which treats missing values as a valid nonmissing level.

The NOMCAR option applies only to Taylor series variance estimation. The replication methods, which you request with the VARMETHOD=BRR and VARMETHOD=JACKKNIFE options, do not use the NOMCAR option.

NOSORT

suppresses the internal sorting process to shorten the computation time if the data set is presorted by the STRATA and CLUSTER variables. By default, the procedure sorts the data by the STRATA variables if you use the STRATA statement; then the procedure sorts the data by the CLUSTER variables within strata. If your data are already stored by the order of STRATA and CLUSTER variables, then you can specify this option to omit this sorting process to reduce the usage of computing resources, especially when your data set is very large. However, if you specify this NOSORT option while your data are not presorted by STRATA and CLUSTER variables, then any changes in these variables creates a new stratum or cluster.

RATE=*value* | *SAS-data-set*

R=*value* | *SAS-data-set*

specifies the sampling rate as a nonnegative *value*, or specifies an input data set that contains the stratum sampling rates. The procedure uses this information to compute a finite population correction for Taylor series variance estimation. The procedure does not use the RATE= option for BRR or jackknife variance estimation, which you request with the VARMETHOD=BRR or VARMETHOD=JACKKNIFE option.

If your sample design has multiple stages, you should specify the *first-stage sampling rate*, which is the ratio of the number of PSUs selected to the total number of PSUs in the population.

For a nonstratified sample design, or for a stratified sample design with the same sampling rate in all strata, you should specify a nonnegative *value* for the RATE= option. If your design is stratified with different sampling rates in the strata, then you should name a SAS data set that contains the stratification variables and the sampling rates. See the section "Specification of Population Totals and Sampling Rates" on page 6422 for more details.

The *value* in the RATE= option or the values of _RATE_ in the secondary data set must be nonnegative numbers. You can specify *value* as a number between 0 and 1. Or you can specify *value* in percentage form as a number between 1 and 100, and PROC SURVEYLOGISTIC converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

If you do not specify the TOTAL= or RATE= option, then the Taylor series variance estimation does not include a finite population correction. You cannot specify both the TOTAL= and RATE= options.

TOTAL=*value* | *SAS-data-set*

N=*value* | *SAS-data-set*

specifies the total number of primary sampling units in the study population as a positive *value*, or specifies an input data set that contains the stratum population totals. The procedure uses this information to compute a finite population correction for Taylor series variance estimation. The procedure does not use the TOTAL= option for BRR or jackknife variance estimation, which you request with the VARMETHOD=BRR or VARMETHOD=JACKKNIFE option.

For a nonstratified sample design, or for a stratified sample design with the same population total in all strata, you should specify a positive *value* for the TOTAL= option. If your sample

design is stratified with different population totals in the strata, then you should name a SAS data set that contains the stratification variables and the population totals. See the section "Specification of Population Totals and Sampling Rates" on page 6422 for more details.

If you do not specify the TOTAL= or RATE= option, then the Taylor series variance estimation does not include a finite population correction. You cannot specify both the TOTAL= and RATE= options.

VARMETHOD=BRR <(*method-options*)>

VARMETHOD=JACKKNIFE | JK <(*method-options*)>

VARMETHOD=TAYLOR

specifies the variance estimation method. VARMETHOD=TAYLOR requests the Taylor series method, which is the default if you do not specify the VARMETHOD= option or the REPWEIGHTS statement. VARMETHOD=BRR requests variance estimation by balanced repeated replication (BRR), and VARMETHOD=JACKKNIFE requests variance estimation by the delete-1 jackknife method.

For VARMETHOD=BRR and VARMETHOD=JACKKNIFE you can specify *method-options* in parentheses. Table 84.1 summarizes the available *method-options*.

Table 84.1 Variance Estimation Method-Options

Keyword	Method	(*Method-Options*)
BRR	Balanced repeated replication	FAY <=*value*> HADAMARD | H=*SAS-data-set* OUTWEIGHTS=*SAS-data-set* PRINTH REPS=*number*
JACKKNIFE | JK	Jackknife	OUTJKCOEFS=*SAS-data-set* OUTWEIGHTS=*SAS-data-set*
TAYLOR	Taylor series	

Method-options must be enclosed in parentheses following the method keyword. For example:

```
varmethod=BRR(reps=60 outweights=myReplicateWeights)
```

The following values are available for the VARMETHOD= option:

BRR <(*method-options*)> requests balanced repeated replication (BRR) variance estimation. The BRR method requires a stratified sample design with two primary sampling units (PSUs) per stratum. See the section "Balanced Repeated Replication (BRR) Method" on page 6429 for more information.

You can specify the following *method-options* in parentheses following VARMETHOD=BRR.

FAY <=*value***>**

requests Fay's method, a modification of the BRR method, for variance estimation. See the section "Fay's BRR Method" on page 6430 for more information.

You can specify the *value* of the Fay coefficient, which is used in converting the original sampling weights to replicate weights. The Fay coefficient must be a nonnegative number less than 1. By default, the value of the Fay coefficient equals 0.5.

HADAMARD=*SAS-data-set*

H=*SAS-data-set*

names a SAS data set that contains the Hadamard matrix for BRR replicate construction. If you do not provide a Hadamard matrix with the HADAMARD= method-option, PROC SURVEYLOGISTIC generates an appropriate Hadamard matrix for replicate construction. See the sections "Balanced Repeated Replication (BRR) Method" on page 6429 and "Hadamard Matrix" on page 6432 for details.

If a Hadamard matrix of a given dimension exists, it is not necessarily unique. Therefore, if you want to use a specific Hadamard matrix, you must provide the matrix as a SAS data set in the HADAMARD=*SAS-data-set* method-option.

In the HADAMARD= input data set, each variable corresponds to a column of the Hadamard matrix, and each observation corresponds to a row of the matrix. You can use any variable names in the HADAMARD= data set. All values in the data set must equal either 1 or -1. You must ensure that the matrix you provide is indeed a Hadamard matrix—that is, $\mathbf{A}'\mathbf{A} = R\mathbf{I}$, where $\mathbf{A}$ is the Hadamard matrix of dimension R and $\mathbf{I}$ is an identity matrix. PROC SURVEYLOGISTIC does not check the validity of the Hadamard matrix that you provide.

The HADAMARD= input data set must contain at least H variables, where H denotes the number of first-stage strata in your design. If the data set contains more than H variables, the procedure uses only the first H variables. Similarly, the HADAMARD= input data set must contain at least H observations.

If you do not specify the REPS= method-option, then the number of replicates is taken to be the number of observations in the HADAMARD= input data set. If you specify the number of replicates—for example, REPS=*nreps*—then the first *nreps* observations in the HADAMARD= data set are used to construct the replicates.

You can specify the PRINTH option to display the Hadamard matrix that the procedure uses to construct replicates for BRR.

OUTWEIGHTS=*SAS-data-set*

names a SAS data set that contains replicate weights. See the section "Balanced Repeated Replication (BRR) Method" on page 6429 for information about replicate weights. See the section "Replicate Weights Output Data Set" on page 6440 for more details about the contents of the OUTWEIGHTS= data set.

The OUTWEIGHTS= method-option is not available when you provide replicate weights with the REPWEIGHTS statement.

PRINTH

displays the Hadamard matrix.

When you provide your own Hadamard matrix with the HADAMARD= method-option, only the rows and columns of the Hadamard matrix that are used by the procedure are displayed. See the sections "Balanced Repeated Replication (BRR) Method" on page 6429 and "Hadamard Matrix" on page 6432 for details.

The PRINTH method-option is not available when you provide replicate weights with the REPWEIGHTS statement because the procedure does not use a Hadamard matrix in this case.

REPS=*number*

specifies the number of replicates for BRR variance estimation. The value of *number* must be an integer greater than 1.

If you do not provide a Hadamard matrix with the HADAMARD= method-option, the number of replicates should be greater than the number of strata and should be a multiple of 4. See the section "Balanced Repeated Replication (BRR) Method" on page 6429 for more information. If a Hadamard matrix cannot be constructed for the REPS= value that you specify, the value is increased until a Hadamard matrix of that dimension can be constructed. Therefore, it is possible for the actual number of replicates used to be larger than the REPS= value that you specify.

If you provide a Hadamard matrix with the HADAMARD= method-option, the value of REPS= must not be less than the number of rows in the Hadamard matrix. If you provide a Hadamard matrix and do not specify the REPS= method-option, the number of replicates equals the number of rows in the Hadamard matrix.

If you do not specify the REPS= or HADAMARD= method-option and do not include a REPWEIGHTS statement, the number of replicates equals the smallest multiple of 4 that is greater than the number of strata.

If you provide replicate weights with the REPWEIGHTS statement, the procedure does not use the REPS= method-option. With a REPWEIGHTS statement, the number of replicates equals the number of REPWEIGHTS variables.

JACKKNIFE | JK <(*method-options*)> requests variance estimation by the delete-1 jackknife method. See the section "Jackknife Method" on page 6431 for details. If you provide replicate weights with a REPWEIGHTS statement, VARMETHOD=JACKKNIFE is the default variance estimation method.

You can specify the following *method-options* in parentheses following VARMETHOD=JACKKNIFE:

OUTWEIGHTS=*SAS-data-set*
names a SAS data set that contains replicate weights. "Jackknife Method" on page 6431 for information about replicate weights. See the section "Replicate Weights Output Data Set" on page 6440 for more details about the contents of the OUTWEIGHTS= data set.

The OUTWEIGHTS= method-option is not available when you provide replicate weights with the REPWEIGHTS statement.

OUTJKCOEFS=*SAS-data-set*
names a SAS data set that contains jackknife coefficients. See the section "Jackknife Coefficients Output Data Set" on page 6441 for more details about the contents of the OUTJKCOEFS= data set.

TAYLOR requests Taylor series variance estimation. This is the default method if you do not specify the VARMETHOD= option and if there is no REPWEIGHTS statement. See the section "Taylor Series (Linearization)" on page 6427 for more information.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC SURVEYLOGISTIC to obtain separate analyses on observations in groups defined by the BY variables.

Note that using a BY statement provides completely separate analyses of the BY groups. It does not provide a statistically valid subpopulation (or domain) analysis, where the total number of units in the subpopulation is not known with certainty.

When a BY statement appears, the procedure expects the input data sets to be sorted in the order of the BY variables. The *variables* are one or more variables in the input data set.

If you specify more than one BY statement, the procedure uses only the latest BY statement and ignores any previous ones.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Use the BY statement option NOTSORTED or DESCENDING. The NOTSORTED option does not mean that the data are unsorted, but rather that the data are arranged in groups (ac-

cording to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.

- Create an index of the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variable* < *(v-options)* > < *variable* < *(v-options)* > ... > < / *v-options* > **;**

The CLASS statement names the classification variables to be used in the analysis. The CLASS statement must precede the MODEL statement. You can specify various *v-options* for each variable by enclosing them in parentheses after the variable name. You can also specify global *v-options* for the CLASS statement by placing them after a slash (/). Global *v-options* are applied to all the variables specified in the CLASS statement. However, individual CLASS variable *v-options* override the global *v-options*.

CPREFIX= *n*

specifies that, at most, the first *n* characters of a CLASS variable name be used in creating names for the corresponding dummy variables. The default is $32 - \min(32, \max(2, f))$, where f is the formatted length of the CLASS variable.

DESCENDING

DESC

reverses the sorting order of the classification variable.

LPREFIX= *n*

specifies that, at most, the first *n* characters of a CLASS variable label be used in creating labels for the corresponding dummy variables.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of classification variables. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option might be useful when you use the CONTRAST statement. When the default ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values.

The following table shows how PROC SURVEYLOGISTIC interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent.

For more information about sorting order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

PARAM=*keyword*

specifies the parameterization method for the classification variable or variables. Design matrix columns are created from CLASS variables according to the following coding schemes; the default is PARAM=EFFECT.

EFFECT specifies effect coding

GLM specifies less-than-full-rank, reference cell coding; this option can be used only as a global option

ORDINAL specifies the cumulative parameterization for an ordinal CLASS variable

POLYNOMIAL | POLY specifies polynomial coding

REFERENCE | REF specifies reference cell coding

ORTHEFFECT orthogonalizes PARAM=EFFECT

ORTHORDINAL | ORTHOTHERM orthogonalizes PARAM=ORDINAL

ORTHPOLY orthogonalizes PARAM=POLYNOMIAL

ORTHREF orthogonalizes PARAM=REFERENCE

If PARAM=ORTHPOLY or PARAM=POLY, and the CLASS levels are numeric, then the ORDER= option in the CLASS statement is ignored, and the internal, unformatted values are used.

EFFECT, POLYNOMIAL, REFERENCE, ORDINAL, and their orthogonal parameterizations are full rank. The REF= option in the CLASS statement determines the reference level for EFFECT, REFERENCE, and their orthogonal parameterizations.

Parameter names for a CLASS predictor variable are constructed by concatenating the CLASS variable name with the CLASS levels. However, for the POLYNOMIAL and orthogonal parameterizations, parameter names are formed by concatenating the CLASS variable name and keywords that reflect the parameterization.

REF=*'level'* | *keyword*

specifies the reference level for PARAM=EFFECT or PARAM=REFERENCE. For an individual (but not a global) variable REF= *option*, you can specify the *level* of the variable to use as the reference level. For a global or individual variable REF= *option*, you can use one of the following *keywords*. The default is REF=LAST.

FIRST	designates the first ordered level as reference.
LAST	designates the last ordered level as reference.

CLUSTER Statement

CLUSTER *variables* **;**

The CLUSTER statement names variables that identify the clusters in a clustered sample design. The combinations of categories of CLUSTER variables define the clusters in the sample. If there is a STRATA statement, clusters are nested within strata.

If your sample design has clustering at multiple stages, you should identify only the first-stage clusters, or primary sampling units (PSUs), in the CLUSTER statement. See the section "Primary Sampling Units (PSUs)" on page 6423 for more information.

If you provide replicate weights for BRR or jackknife variance estimation with the REPWEIGHTS statement, you do not need to specify a CLUSTER statement.

The CLUSTER *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric. The formatted values of the CLUSTER variables determine the CLUSTER variable levels. Thus, you can use formats to group values into levels. See the FORMAT procedure in the *Base SAS Procedures Guide* and the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary* for more information.

You can use multiple CLUSTER statements to specify cluster variables. The procedure uses all variables from all CLUSTER statements to create clusters.

CONTRAST Statement

CONTRAST *'label' row-description* < *, ... , row-description* < */ options* > > **;**

where a *row-description* is defined as follows:

effect values < *, . . ., effect values* >

The CONTRAST statement provides a mechanism for obtaining customized hypothesis tests. It is similar to the CONTRAST statement in PROC LOGISTIC and PROC GLM, depending on the coding schemes used with any classification variables involved.

The CONTRAST statement enables you to specify a matrix, $\mathbf{L}$, for testing the hypothesis $\mathbf{L}\boldsymbol{\theta} = 0$, where $\boldsymbol{\theta}$ is the parameter vector. You must be familiar with the details of the model parameteriza-

tion that PROC SURVEYLOGISTIC uses (for more information, see the PARAM= option in the section "CLASS Statement" on page 6389). Optionally, the CONTRAST statement enables you to estimate each row, $\mathbf{l}_i\boldsymbol{\theta}$, of $\mathbf{L}\boldsymbol{\theta}$ and test the hypothesis $\mathbf{l}_i\boldsymbol{\theta} = 0$. Computed statistics are based on the asymptotic chi-square distribution of the Wald statistic.

There is no limit to the number of CONTRAST statements that you can specify, but they must appear after the MODEL statement.

The following parameters are specified in the CONTRAST statement:

label identifies the contrast on the output. A label is required for every contrast specified, and it must be enclosed in quotes.

effect identifies an effect that appears in the MODEL statement. The name INTERCEPT can be used as an effect when one or more intercepts are included in the model. You do not need to include all effects that are included in the MODEL statement.

values are constants that are elements of the **L** matrix associated with the effect. To correctly specify your contrast, it is crucial to know the ordering of parameters within each effect and the variable levels associated with any parameter. The "Class Level Information" table shows the ordering of levels within variables. The E option, described later in this section, enables you to verify the proper correspondence of *values* to parameters.

The rows of **L** are specified in order and are separated by commas. Multiple degree-of-freedom hypotheses can be tested by specifying multiple *row-descriptions*. For any of the full-rank parameterizations, if an effect is not specified in the CONTRAST statement, all of its coefficients in the **L** matrix are set to 0. If too many values are specified for an effect, the extra ones are ignored. If too few values are specified, the remaining ones are set to 0.

When you use effect coding (by default or by specifying PARAM=EFFECT in the CLASS statement), all parameters are directly estimable (involve no other parameters).

For example, suppose an effect that is coded CLASS variable A has four levels. Then there are three parameters $(\alpha_1, \alpha_2, \alpha_3)$ that represent the first three levels, and the fourth parameter is represented by

$$-\alpha_1 - \alpha_2 - \alpha_3$$

To test the first versus the fourth level of A, you would test

$$\alpha_1 = -\alpha_1 - \alpha_2 - \alpha_3$$

or, equivalently,

$$2\alpha_1 + \alpha_2 + \alpha_3 = 0$$

which, in the form $\mathbf{L}\boldsymbol{\theta} = 0$, is

$$\begin{bmatrix} 2 & 1 & 1 \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \end{bmatrix} = 0$$

Therefore, you would use the following CONTRAST statement:

```
contrast '1 vs. 4' A 2 1 1;
```

To contrast the third level with the average of the first two levels, you would test

$$\frac{\alpha_1 + \alpha_2}{2} = \alpha_3$$

or, equivalently,

$$\alpha_1 + \alpha_2 - 2\alpha_3 = 0$$

Therefore, you would use the following CONTRAST statement:

```
contrast '1&2 vs. 3' A 1 1 -2;
```

Other CONTRAST statements are constructed similarly. For example:

```
contrast '1 vs. 2    '  A  1 -1  0;
contrast '1&2 vs. 4  '  A  3  3  2;
contrast '1&2 vs. 3&4'  A  2  2  0;
contrast 'Main Effect'  A  1  0  0,
                        A  0  1  0,
                        A  0  0  1;
```

When you use the less-than-full-rank parameterization (by specifying PARAM=GLM in the CLASS statement), each row is checked for estimability. If PROC SURVEYLOGISTIC finds a contrast to be nonestimable, it displays missing values in corresponding rows in the results. PROC SURVEYLOGISTIC handles missing level combinations of classification variables in the same manner as PROC LOGISTIC. Parameters corresponding to missing level combinations are not included in the model. This convention can affect the way in which you specify the **L** matrix in your CONTRAST statement. If the elements of **L** are not specified for an effect that contains a specified effect, then the elements of the specified effect are distributed over the levels of the higher-order effect just as the LOGISTIC procedure does for its CONTRAST and ESTIMATE statements. For example, suppose that the model contains effects A and B and their interaction A*B. If you specify a CONTRAST statement involving A alone, the **L** matrix contains nonzero terms for both A and A*B, since A*B contains A.

The degrees of freedom is the number of linearly independent constraints implied by the CONTRAST statement—that is, the rank of **L**.

You can specify the following options after a slash (/).

ALPHA=*value*

sets the confidence level for confidence limits. The value of the ALPHA= option must be between 0 and 1, and the default value is 0.05. A confidence level of α produces $100(1-\alpha)\%$ confidence limits. The default of ALPHA=0.05 produces 95% confidence limits.

E

requests that the **L** matrix be displayed.

ESTIMATE=*keyword*

requests that each individual contrast (that is, each row, $\mathbf{l}_i\boldsymbol{\beta}$, of $\mathbf{L}\boldsymbol{\beta}$) or exponentiated contrast ($e^{\mathbf{l}_i\boldsymbol{\beta}}$) be estimated and tested. PROC SURVEYLOGISTIC displays the point estimate, its standard error, a Wald confidence interval, and a Wald chi-square test for each contrast. The significance level of the confidence interval is controlled by the ALPHA= option. You can estimate the contrast or the exponentiated contrast ($e^{\mathbf{l}_i\boldsymbol{\beta}}$), or both, by specifying one of the following *keywords*:

PARM	specifies that the contrast itself be estimated
EXP	specifies that the exponentiated contrast be estimated
BOTH	specifies that both the contrast and the exponentiated contrast be estimated

SINGULAR=*value*

tunes the estimability checking. If $\mathbf{v}$ is a vector, define ABS($\mathbf{v}$) to be the largest absolute value of the elements of $\mathbf{v}$. For a row vector $\mathbf{l}$ of the matrix $\mathbf{L}$, define

$$c = \begin{cases} \text{ABS}(\mathbf{l}) & \text{if ABS}(\mathbf{l}) > 0 \\ 1 & \text{otherwise} \end{cases}$$

If ABS($\mathbf{l} - \mathbf{lH}$) is greater than *c*value*, then $\mathbf{l}\boldsymbol{\beta}$ is declared nonestimable. The $\mathbf{H}$ matrix is the Hermite form matrix $\mathbf{I}_0^-\mathbf{I}_0$, where $\mathbf{I}_0^-$ represents a generalized inverse of the information matrix $\mathbf{I}_0$ of the null model. The *value* must be between 0 and 1; the default is 10^{-4}.

DOMAIN Statement

DOMAIN *variables < variable*variable variable*variable*variable ... >* **;**

The DOMAIN statement requests analysis for subpopulations, or domains, in addition to analysis for the entire study population. The DOMAIN statement names the variables that identify domains, which are called domain variables.

It is common practice to compute statistics for domains. The formation of these domains might be unrelated to the sample design. Therefore, the sample sizes for the domains are random variables. In order to incorporate this variability into the variance estimation, you should use a DOMAIN statement.

Note that a DOMAIN statement is different from a BY statement. In a BY statement, you treat the sample sizes as fixed in each subpopulation, and you perform analysis within each BY group independently.

You should use the DOMAIN statement on the entire data set to perform the domain analysis. Creating a new data set from a single domain and analyzing that with PROC SURVEYLOGISTIC yields inappropriate estimates of variance.

A domain variable can be either character or numeric. The procedure treats domain variables as categorical variables. If a variable appears by itself in a DOMAIN statement, each level of this variable determines a domain in the study population. If two or more variables are joined by asterisks (*),

then every possible combination of levels of these variables determines a domain. The procedure performs a descriptive analysis within each domain defined by the domain variables.

The formatted values of the domain variables determine the categorical variable levels. Thus, you can use formats to group values into levels. See the FORMAT procedure in the *Base SAS Procedures Guide* and the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary* for more information.

FREQ Statement

FREQ *variable* **;**

The *variable* in the FREQ statement identifies a variable that contains the frequency of occurrence of each observation. PROC SURVEYLOGISTIC treats each observation as if it appears n times, where n is the value of the FREQ variable for the observation. If it is not an integer, the frequency value is truncated to an integer. If the frequency value is less than 1 or missing, the observation is not used in the model fitting. When the FREQ statement is not specified, each observation is assigned a frequency of 1.

If you use the events/trials syntax in the MODEL statement, the FREQ statement is not allowed because the event and trial variables represent the frequencies in the data set.

MODEL Statement

MODEL *events/trials* = < *effects* < / *options* > > **;**

MODEL *variable* < *(v-options)* > = < *effects* > < / *options* > **;**

The MODEL statement names the response variable and the explanatory effects, including covariates, main effects, interactions, and nested effects; see the section "Specification of Effects" on page 2488 of Chapter 39, "The GLM Procedure," for more information. If you omit the explanatory variables, the procedure fits an intercept-only model. Model options can be specified after a slash (/).

Two forms of the MODEL statement can be specified. The first form, referred to as *single-trial* syntax, is applicable to binary, ordinal, and nominal response data. The second form, referred to as *events/trials* syntax, is restricted to the case of binary response data. The single-trial syntax is used when each observation in the DATA= data set contains information about only a single trial, such as a single subject in an experiment. When each observation contains information about multiple binary-response trials, such as the counts of the number of subjects observed and the number responding, then events/trials syntax can be used.

In the events/trials syntax, you specify two variables that contain count data for a binomial experiment. These two variables are separated by a slash. The value of the first variable, *events*, is the

number of positive responses (or events), and it must be nonnegative. The value of the second variable, *trials*, is the number of trials, and it must not be less than the value of *events*.

In the single-trial syntax, you specify one variable (on the left side of the equal sign) as the response variable. This variable can be character or numeric. Options specific to the response variable can be specified immediately after the response variable with parentheses around them.

For both forms of the MODEL statement, explanatory *effects* follow the equal sign. Variables can be either continuous or classification variables. Classification variables can be character or numeric, and they must be declared in the CLASS statement. When an effect is a classification variable, the procedure enters a set of coded columns into the design matrix instead of directly entering a single column containing the values of the variable.

Response Variable Options

You specify the following options by enclosing them in parentheses after the response variable.

DESCENDING
DESC

reverses the order of response categories. If both the DESCENDING and ORDER= options are specified, PROC SURVEYLOGISTIC orders the response categories according to the ORDER= option and then reverses that order. See the section "Response Level Ordering" on page 6411 for more detail.

EVENT='*category***' | *keyword***

specifies the event category for the binary response model. PROC SURVEYLOGISTIC models the probability of the event category. The EVENT= option has no effect when there are more than two response categories. You can specify the value (formatted if a format is applied) of the event category in quotes or you can specify one of the following keywords. The default is EVENT=FIRST.

FIRST	designates the first ordered category as the event
LAST	designates the last ordered category as the event

One of the most common sets of response levels is {0,1}, with 1 representing the event for which the probability is to be modeled. Consider the example where Y takes the values 1 and 0 for event and nonevent, respectively, and Exposure is the explanatory variable. To specify the value 1 as the event category, use the following MODEL statement:

```
model Y(event='1') = Exposure;
```

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of the response variable. By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent.

When the default ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values.

The following table shows the interpretation of the ORDER= values.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

For more information about sorting order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts* for more information.

REFERENCE='*category*' | *keyword*

REF='*category*' | *keyword*

specifies the reference category for the generalized logit model and the binary response model. For the generalized logit model, each nonreference category is contrasted with the reference category. For the binary response model, specifying one response category as the reference is the same as specifying the other response category as the event category. You can specify the value (formatted if a format is applied) of the reference category in quotes or you can specify one of the following keywords. The default is REF=LAST.

FIRST	designates the first ordered category as the reference
LAST	designates the last ordered category as the reference

Model Options

Model options can be specified after a slash (/). Table 84.2 summarizes the options available in the MODEL statement.

Table 84.2 MODEL Statement Options

Option	Description
Model Specification Options	
LINK=	Specifies link function
NOINT	Suppresses intercept(s)
OFFSET=	Specifies offset variable
Convergence Criterion Options	
ABSFCONV=	Specifies absolute function convergence criterion
FCONV=	Specifies relative function convergence criterion
GCONV=	Specifies relative gradient convergence criterion
XCONV=	Specifies relative parameter convergence criterion
MAXITER=	Specifies maximum number of iterations
NOCHECK	Suppresses checking for infinite parameters

Table 84.2 (continued)

Option	Description
RIDGING=	Specifies technique used to improve the log-likelihood function when its value is worse than that of the previous step
SINGULAR=	Specifies tolerance for testing singularity
TECHNIQUE=	Specifies iterative algorithm for maximization
Options for Adjustment to Variance Estimation	
VADJUST=	Chooses variance estimation adjustment method
Options for Confidence Intervals	
ALPHA=	Specifies α for the $100(1-\alpha)\%$ confidence intervals
CLPARM	Computes confidence intervals for parameters
CLODDS	Computes confidence intervals for odds ratios
Options for Display of Details	
CORRB	Displays correlation matrix
COVB	Displays covariance matrix
EXPB	Displays exponentiated values of estimates
ITPRINT	Displays iteration history
NODUMMYPRINT	Suppresses "Class Level Information" table
PARMLABEL	Displays parameter labels
RSQUARE	Displays generalized R^2
STB	Displays standardized estimates

The following list describes these options.

ABSFCONV=*value*

specifies the absolute function convergence criterion. Convergence requires a small change in the log-likelihood function in subsequent iterations:

$$|l^{(i)} - l^{(i-1)}| < value$$

where $l^{(i)}$ is the value of the log-likelihood function at iteration i. See the section "Convergence Criteria" on page 6418.

ALPHA=*value*

sets the level of significance α for $100(1-\alpha)\%$ confidence intervals for regression parameters or odds ratios. The value α must be between 0 and 1. By default, α is equal to the value of the ALPHA= option in the PROC SURVEYLOGISTIC statement, or $\alpha = 0.05$ if the ALPHA= option is not specified. This option has no effect unless confidence limits for the parameters or odds ratios are requested.

CLODDS

requests confidence intervals for the odds ratios. Computation of these confidence intervals is based on individual Wald tests. The confidence coefficient can be specified with the ALPHA= option.

See the section "Wald Confidence Intervals for Parameters" on page 6434 for more information.

CLPARM

requests confidence intervals for the parameters. Computation of these confidence intervals is based on the individual Wald tests. The confidence coefficient can be specified with the ALPHA= option.

See the section "Wald Confidence Intervals for Parameters" on page 6434 for more information.

CORRB

displays the correlation matrix of the parameter estimates.

COVB

displays the covariance matrix of the parameter estimates.

EXPB

EXPEST

displays the exponentiated values ($e^{\hat{\theta}_i}$) of the parameter estimates $\hat{\theta}_i$ in the "Analysis of Maximum Likelihood Estimates" table for the logit model. These exponentiated values are the estimated odds ratios for the parameters corresponding to the continuous explanatory variables.

FCONV=*value*

specifies the relative function convergence criterion. Convergence requires a small relative change in the log-likelihood function in subsequent iterations:

$$\frac{|l^{(i)} - l^{(i-1)}|}{|l^{(i-1)}| + 1\text{E}{-}6} < value$$

where $l^{(i)}$ is the value of the log likelihood at iteration i. See the section "Convergence Criteria" on page 6418 for details.

GCONV=*value*

specifies the relative gradient convergence criterion. Convergence requires that the normalized prediction function reduction is small:

$$\frac{\mathbf{g}^{(i)\prime}\mathbf{I}^{(i)}\mathbf{g}^{(i)}}{|l^{(i)}| + 1\text{E}{-}6} < value$$

where $l^{(i)}$ is the value of the log-likelihood function, $\mathbf{g}^{(i)}$ is the gradient vector, and $\mathbf{I}^{(i)}$ the (expected) information matrix. All of these functions are evaluated at iteration i. This is the default convergence criterion, and the default value is 1E−8. See the section "Convergence Criteria" on page 6418 for details.

ITPRINT

displays the iteration history of the maximum-likelihood model fitting. The ITPRINT option also displays the last evaluation of the gradient vector and the final change in the $-2\log L$.

LINK=*keyword*

L=*keyword*

specifies the link function that links the response probabilities to the linear predictors. You can specify one of the following keywords. The default is LINK=LOGIT.

CLOGLOG specifies the complementary log-log function. PROC SURVEYLOGISTIC fits the binary complementary log-log model for binary response and fits the cumulative complementary log-log model when there are more than two response categories. Aliases: CCLOGLOG, CCLL, CUMCLOGLOG.

GLOGIT specifies the generalized logit function. PROC SURVEYLOGISTIC fits the generalized logit model where each nonreference category is contrasted with the reference category. You can use the response variable option REF= to specify the reference category.

LOGIT specifies the cumulative logit function. PROC SURVEYLOGISTIC fits the binary logit model when there are two response categories and fits the cumulative logit model when there are more than two response categories. Aliases: CLOGIT, CUMLOGIT.

PROBIT specifies the inverse standard normal distribution function. PROC SURVEYLOGISTIC fits the binary probit model when there are two response categories and fits the cumulative probit model when there are more than two response categories. Aliases: NORMIT, CPROBIT, CUMPROBIT.

See the section "Link Functions and the Corresponding Distributions" on page 6415 for details.

MAXITER=*n*

specifies the maximum number of iterations to perform. By default, MAXITER=25. If convergence is not attained in *n* iterations, the displayed output created by the procedure contains results that are based on the last maximum likelihood iteration.

NOCHECK

disables the checking process to determine whether maximum likelihood estimates of the regression parameters exist. If you are sure that the estimates are finite, this option can reduce the execution time when the estimation takes more than eight iterations. For more information, see the section "Existence of Maximum Likelihood Estimates" on page 6418.

NODUMMYPRINT

suppresses the "Class Level Information" table, which shows how the design matrix columns for the CLASS variables are coded.

NOINT

suppresses the intercept for the binary response model or the first intercept for the ordinal response model.

OFFSET=*name*

names the offset variable. The regression coefficient for this variable is fixed at 1.

PARMLABEL

displays the labels of the parameters in the "Analysis of Maximum Likelihood Estimates" table.

RIDGING=ABSOLUTE | RELATIVE | NONE

specifies the technique used to improve the log-likelihood function when its value in the current iteration is less than that in the previous iteration. If you specify the RIDGING=ABSOLUTE option, the diagonal elements of the negative (expected) Hessian are inflated by adding the ridge value. If you specify the RIDGING=RELATIVE option, the diagonal elements are inflated by a factor of 1 plus the ridge value. If you specify the RIDGING=NONE option, the crude line search method of taking half a step is used instead of ridging. By default, RIDGING=RELATIVE.

RSQUARE

requests a generalized R^2 measure for the fitted model.

For more information, see the section "Generalized Coefficient of Determination" on page 6421.

SINGULAR=*value*

specifies the tolerance for testing the singularity of the Hessian matrix (Newton-Raphson algorithm) or the expected value of the Hessian matrix (Fisher scoring algorithm). The Hessian matrix is the matrix of second partial derivatives of the log likelihood. The test requires that a pivot for sweeping this matrix be at least this *value* times a norm of the matrix. Values of the SINGULAR= option must be numeric. By default, SINGULAR=10^{-12}.

STB

displays the standardized estimates for the parameters for the continuous explanatory variables in the "Analysis of Maximum Likelihood Estimates" table. The standardized estimate of θ_i is given by $\hat{\theta}_i/(s/s_i)$, where s_i is the total sample standard deviation for the ith explanatory variable and

$$s = \begin{cases} \pi/\sqrt{3} & \text{Logistic} \\ 1 & \text{Normal} \\ \pi/\sqrt{6} & \text{Extreme-value} \end{cases}$$

For the intercept parameters and parameters associated with a CLASS variable, the standardized estimates are set to missing.

TECHNIQUE=FISHER | NEWTON

TECH=FISHER | NEWTON

specifies the optimization technique for estimating the regression parameters. NEWTON (or NR) is the Newton-Raphson algorithm and FISHER (or FS) is the Fisher scoring algorithm. Both techniques yield the same estimates, but the estimated covariance matrices are slightly different except for the case where the LOGIT link is specified for binary response data. The default is TECHNIQUE=FISHER. See the section "Iterative Algorithms for Model Fitting" on page 6417 for details.

VADJUST=DF

VADJUST=MOREL <(*Morel-options***)>**

VADJUST=NONE

specifies an adjustment to the variance estimation for the regression coefficients.

By default, PROC SURVEYLOGISTIC uses the degrees of freedom adjustment VADJUST=DF.

If you do not want to use any variance adjustment, you can specify the VADJUST=NONE option. You can specify the VADJUST=MOREL option for the variance adjustment proposed by Morel (1989).

You can specify the following *Morel-options* within parentheses after the VADJUST=MOREL option.

ADJBOUND=ϕ

sets the upper bound coefficient ϕ in the variance adjustment. This upper bound must be positive. By default, the procedure uses $\phi = 0.5$. See the section "Adjustments to the Variance Estimation" on page 6428 for more details on how this upper bound is used in the variance estimation.

DEFFBOUND=δ

sets the lower bound of the estimated design effect in the variance adjustment. This lower bound must be positive. By default, the procedure uses $\delta = 1$. See the section "Adjustments to the Variance Estimation" on page 6428 for more details about how this lower bound is used in the variance estimation.

XCONV=*value*

specifies the relative parameter convergence criterion. Convergence requires a small relative parameter change in subsequent iterations:

$$\max_j |\delta_j^{(i)}| < value$$

where

$$\delta_j^{(i)} = \begin{cases} \theta_j^{(i)} - \theta_j^{(i-1)} & |\theta_j^{(i-1)}| < 0.01 \\ \dfrac{\theta_j^{(i)} - \theta_j^{(i-1)}}{\theta_j^{(i-1)}} & \text{otherwise} \end{cases}$$

and $\theta_j^{(i)}$ is the estimate of the jth parameter at iteration i. See the section "Iterative Algorithms for Model Fitting" on page 6417.

OUTPUT Statement

OUTPUT < **OUT=***SAS-data-set* > < *options* > **;**

The OUTPUT statement creates a new SAS data set that contains all the variables in the input data set and, optionally, the estimated linear predictors and their standard error estimates, the estimates of the cumulative or individual response probabilities, and the confidence limits for the cumulative probabilities. Formulas for the statistics are given in the section "Linear Predictor, Predicted Probability, and Confidence Limits" on page 6438.

If you use the single-trial syntax, the data set also contains a variable named _LEVEL_, which indicates the level of the response that the given row of output is referring to. For example, the value of the cumulative probability variable is the probability that the response variable is as large as the corresponding value of _LEVEL_. For details, see the section "OUT= Data Set in the OUTPUT Statement" on page 6440.

The estimated linear predictor, its standard error estimate, all predicted probabilities, and the confidence limits for the cumulative probabilities are computed for all observations in which the explanatory variables have no missing values, even if the response is missing. By adding observations with missing response values to the input data set, you can compute these statistics for new observations, or for settings of the explanatory variables not present in the data, without affecting the model fit.

OUT=*SAS-data-set*

names the output data set. If you omit the OUT= option, the output data set is created and given a default name by using the DATA*n* convention.

The statistic options in the OUTPUT statement specify the statistics to be included in the output data set and name the new variables that contain the statistics.

LOWER | L=*name*

names the variable that contains the lower confidence limits for π, where π is the probability of the event response if events/trials syntax or the single-trial syntax with binary response is specified; π is cumulative probability (that is, the probability that the response is less than or equal to the value of _LEVEL_) for a cumulative model; and π is the individual probability (that is, the probability that the response category is represented by the value of _LEVEL_) for the generalized logit model. See the ALPHA= option for information about setting the confidence level.

PREDICTED | P=*name*

names the variable that contains the predicted probabilities. For the events/trials syntax or the single-trial syntax with binary response, it is the predicted event probability. For a cumulative model, it is the predicted cumulative probability (that is, the probability that the response variable is less than or equal to the value of _LEVEL_); and for the generalized logit model, it is the predicted individual probability (that is, the probability of the response category represented by the value of _LEVEL_).

PREDPROBS=(*keywords***)**

requests individual, cumulative, or cross validated predicted probabilities. Descriptions of the *keywords* are as follows.

INDIVIDUAL | I requests the predicted probability of each response level. For a response variable Y with three levels, 1, 2, and 3, the individual probabilities are Pr(Y=1), Pr(Y=2), and Pr(Y=3).

CUMULATIVE | C requests the cumulative predicted probability of each response level. For a response variable Y with three levels, 1, 2, and 3, the cumulative probabilities are Pr(Y≤1), Pr(Y≤2), and Pr(Y≤3). The cumulative probability for the last response level always has the constant value of 1. For generalized logit models, the cumulative predicted probabilities are not computed and are set to missing.

CROSSVALIDATE | XVALIDATE | X requests the cross validated individual predicted probability of each response level. These probabilities are derived from the leave-one-out principle; that is, dropping the data of one subject and reestimating the parameter estimates. PROC SURVEYLOGISTIC uses a less expensive one-step approximation to compute the parameter estimates. This option is valid only for binary response models; for nominal and ordinal models, the cross validated probabilities are not computed and are set to missing.

See the section "Details of the PREDPROBS= Option" on page 6404 at the end of this section for further details.

STDXBETA=*name*
names the variable that contains the standard error estimates of XBETA (the definition of which follows).

UPPER | U=*name*
names the variable that contains the upper confidence limits for π, where π is the probability of the event response if events/trials syntax or single-trial syntax with binary response is specified; π is cumulative probability (that is, the probability that the response is less than or equal to the value of _LEVEL_) for a cumulative model; and π is the individual probability (that is, the probability that the response category is represented by the value of _LEVEL_) for the generalized logit model. See the ALPHA= option for information about setting the confidence level.

XBETA=*name*
names the variable that contains the estimates of the linear predictor $\alpha_i + \mathbf{x}\boldsymbol{\beta}$, where i is the corresponding ordered value of _LEVEL_.

You can specify the following option after a slash.

ALPHA=*value*
sets the level of significance α for $100(1-\alpha)\%$ confidence limits for the appropriate response probabilities. The value α must be between 0 and 1. By default, α is equal to the value of the ALPHA= option in the PROC SURVEYLOGISTIC statement, or 0.05 if the ALPHA= option is not specified.

Details of the PREDPROBS= Option

You can request any of the three given types of predicted probabilities. For example, you can request both the individual predicted probabilities and the cross validated probabilities by specifying PREDPROBS=(I X).

When you specify the PREDPROBS= option, two automatic variables _FROM_ and _INTO_ are included for the single-trial syntax and only one variable, _INTO_, is included for the events/trials syntax. The _FROM_ variable contains the formatted value of the observed response. The variable _INTO_ contains the formatted value of the response level with the largest individual predicted probability.

If you specify PREDPROBS=INDIVIDUAL, the OUTPUT data set contains k additional variables representing the individual probabilities, one for each response level, where k is the maximum number of response levels across all BY groups. The names of these variables have the form IP_*xxx*, where *xxx* represents the particular level. The representation depends on the following situations:

- If you specify the events/trials syntax, *xxx* is either Event or Nonevent. Thus, the variable that contains the event probabilities is named IP_Event and the variable containing the nonevent probabilities is named IP_Nonevent.
- If you specify the single-trial syntax with more than one BY group, *xxx* is 1 for the first ordered level of the response, 2 for the second ordered level of the response, and so forth, as given in the "Response Profile" table. The variable that contains the predicted probabilities Pr(Y=1) is named IP_1, where Y is the response variable. Similarly, IP_2 is the name of the variable containing the predicted probabilities Pr(Y=2), and so on.
- If you specify the single-trial syntax with no BY-group processing, *xxx* is the left-justified formatted value of the response level (the value can be truncated so that IP_*xxx* does not exceed 32 characters). For example, if Y is the response variable with response levels 'None,' 'Mild,' and 'Severe,' the variables representing individual probabilities Pr(Y='None'), Pr(Y='Mild'), and Pr(Y='Severe') are named IP_None, IP_Mild, and IP_Severe, respectively.

If you specify PREDPROBS=CUMULATIVE, the OUTPUT data set contains k additional variables that represent the cumulative probabilities, one for each response level, where k is the maximum number of response levels across all BY groups. The names of these variables have the form CP_*xxx*, where *xxx* represents the particular response level. The naming convention is similar to that given by PREDPROBS=INDIVIDUAL. The PREDPROBS=CUMULATIVE values are the same as those output by the PREDICT=keyword, but they are arranged in variables in each output observation rather than in multiple output observations.

If you specify PREDPROBS=CROSSVALIDATE, the OUTPUT data set contains k additional variables representing the cross validated predicted probabilities of the k response levels, where k is the maximum number of response levels across all BY groups. The names of these variables have the form XP_*xxx*, where *xxx* represents the particular level. The representation is the same as that given by PREDPROBS=INDIVIDUAL, except that for the events/trials syntax there are four variables for the cross validated predicted probabilities instead of two:

XP_EVENT_R1E is the cross validated predicted probability of an event when a current event trial is removed.

XP_NONEVENT_R1E is the cross validated predicted probability of a nonevent when a current event trial is removed.

XP_EVENT_R1N is the cross validated predicted probability of an event when a current nonevent trial is removed.

XP_NONEVENT_R1N is the cross validated predicted probability of a nonevent when a current nonevent trial is removed.

REPWEIGHTS Statement

REPWEIGHTS *variables* < / *options* > **;**

The REPWEIGHTS statement names variables that provide replicate weights for BRR or jackknife variance estimation, which you request with the VARMETHOD=BRR or VARMETHOD=JACKKNIFE option in the PROC SURVEYLOGISTIC statement. If you do not provide replicate weights for these methods by using a REPWEIGHTS statement, then the procedure constructs replicate weights for the analysis. See the sections "Balanced Repeated Replication (BRR) Method" on page 6429 and "Jackknife Method" on page 6431 for information about replicate weights.

Each REPWEIGHTS variable should contain the weights for a single replicate, and the number of replicates equals the number of REPWEIGHTS variables. The REPWEIGHTS variables must be numeric, and the variable values must be nonnegative numbers.

If you provide replicate weights with a REPWEIGHTS statement, you do not need to specify a CLUSTER or STRATA statement. If you use a REPWEIGHTS statement and do not specify the VARMETHOD= option in the PROC SURVEYLOGISTIC statement, the procedure uses VARMETHOD=JACKKNIFE by default.

If you specify a REPWEIGHTS statement but do not include a WEIGHT statement, the procedure uses the average of each observation's replicate weights as the observation's weight.

You can specify the following options in the REPWEIGHTS statement after a slash (/):

DF=*df*

specifies the degrees of freedom for the analysis. The value of *df* must be a positive number. By default, the degrees of freedom equals the number of REPWEIGHTS variables.

JKCOEFS=*value*

specifies a jackknife coefficient for VARMETHOD=JACKKNIFE. The coefficient *value* must be a nonnegative number less than one. See the section "Jackknife Method" on page 6431 for details about jackknife coefficients.

You can use this option to specify a single value of the jackknife coefficient, which the procedure uses for all replicates. To specify different coefficients for different replicates, use the JKCOEFS=*values* or JKCOEFS=*SAS-data-set* option.

JKCOEFS=*values*

specifies jackknife coefficients for VARMETHOD=JACKKNIFE, where each coefficient corresponds to an individual replicate identified by a REPWEIGHTS variable. You can separate *values* with blanks or commas. The coefficient *values* must be nonnegative numbers less than one. The number of *values* must equal the number of replicate weight variables named in the REPWEIGHTS statement. List these values in the same order in which you list the corresponding replicate weight variables in the REPWEIGHTS statement.

See the section "Jackknife Method" on page 6431 for details about jackknife coefficients.

To specify different coefficients for different replicates, you can also use the JKCOEFS=*SAS-data-set* option. To specify a single jackknife coefficient for all replicates, use the JKCOEFS=*value* option.

JKCOEFS=*SAS-data-set*

names a SAS data set that contains the jackknife coefficients for VARMETHOD=JACKKNIFE. You provide the jackknife coefficients in the JKCOEFS= data set variable JKCoefficient. Each coefficient value must be a nonnegative number less than one. The observations in the JKCOEFS= data set should correspond to the replicates that are identified by the REPWEIGHTS variables. Arrange the coefficients or observations in the JKCOEFS= data set in the same order in which you list the corresponding replicate weight variables in the REPWEIGHTS statement. The number of observations in the JKCOEFS= data set must not be less than the number of REPWEIGHTS variables.

See the section "Jackknife Method" on page 6431 for details about jackknife coefficients.

To specify different coefficients for different replicates, you can also use the JKCOEFS=*values* option. To specify a single jackknife coefficient for all replicates, use the JKCOEFS=*value* option.

STRATA Statement

STRATA | STRATUM *variables* < / *option* > **;**

The STRATA statement names variables that form the strata in a stratified sample design. The combinations of levels of STRATA variables define the strata in the sample.

If your sample design has stratification at multiple stages, you should identify only the first-stage strata in the STRATA statement. See the section "Survey Design Information" on page 6422 for more information.

If you provide replicate weights for BRR or jackknife variance estimation with the REPWEIGHTS statement, you do not need to specify a STRATA statement.

The STRATA *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric. The formatted values of the STRATA variables determine the levels. Thus, you can use formats to group values into levels. See the FORMAT procedure in the *Base SAS Procedures Guide* and the FORMAT statement and SAS formats in *SAS Language Reference: Concepts* for more information.

You can specify the following option in the STRATA statement after a slash (/):

LIST

displays a "Stratum Information" table, which includes values of the STRATA variables and sampling rates for each stratum. This table also provides the number of observations and number of clusters for each stratum and analysis variable. See the section "Displayed Output" on page 6441 for more details.

TEST Statement

< *label:* > **TEST** *equation1* < , *equation2, . . .* >< / *option* > **;**

The TEST statement tests linear hypotheses about the regression coefficients. The Wald test is used to jointly test the null hypotheses (H_0: $\mathbf{L}\boldsymbol{\theta} = \mathbf{c}$) specified in a single TEST statement. When $\mathbf{c} = \mathbf{0}$ you should specify a CONTRAST statement instead.

Each *equation* specifies a linear hypothesis (a row of the $\mathbf{L}$ matrix and the corresponding element of the $\mathbf{c}$ vector); multiple *equations* are separated by commas. The label, which must be a valid SAS name, is used to identify the resulting output and should always be included. You can submit multiple TEST statements.

The form of an *equation* is as follows:

term < ± *term ...* >< = ± *term* < ± *term ...* >>

where *term* is a parameter of the model, or a constant, or a constant times a parameter. For a binary response model, the intercept parameter is named INTERCEPT; for an ordinal response model, the intercept parameters are named INTERCEPT, INTERCEPT2, INTERCEPT3, and so on. When no equal sign appears, the expression is set to 0. The following illustrates possible uses of the TEST statement:

```
proc surveylogistic;
   model y= a1 a2 a3 a4;
   test1: test intercept + .5 * a2 = 0;
   test2: test intercept + .5 * a2;
   test3: test a1=a2=a3;
   test4: test a1=a2, a2=a3;
run;
```

Note that the first and second TEST statements are equivalent, as are the third and fourth TEST statements.

You can specify the following option in the TEST statement after a slash(/).

PRINT

displays intermediate calculations in the testing of the null hypothesis H_0: $\mathbf{L}\boldsymbol{\theta} = \mathbf{c}$. This includes $\mathbf{L}\widehat{\mathbf{V}}(\hat{\boldsymbol{\theta}})\mathbf{L}'$ bordered by $(\mathbf{L}\hat{\boldsymbol{\theta}} - \mathbf{c})$ and $[\mathbf{L}\widehat{\mathbf{V}}(\hat{\boldsymbol{\theta}})\mathbf{L}']^{-1}$ bordered by $[\mathbf{L}\widehat{\mathbf{V}}(\hat{\boldsymbol{\theta}})\mathbf{L}']^{-1}(\mathbf{L}\hat{\boldsymbol{\theta}} - \mathbf{c})$, where $\hat{\boldsymbol{\theta}}$ is the pseudo-estimator of $\boldsymbol{\theta}$ and $\widehat{\mathbf{V}}(\hat{\boldsymbol{\theta}})$ is the estimated covariance matrix of $\hat{\boldsymbol{\theta}}$.

For more information, see the section "Testing Linear Hypotheses about the Regression Coefficients" on page 6434.

UNITS Statement

UNITS *independent1 = list1 < ... independentk = listk >< / options >* **;**

The UNITS statement enables you to specify units of change for the continuous explanatory variables so that customized odds ratios can be estimated. An estimate of the corresponding odds ratio is produced for each unit of change specified for an explanatory variable. The UNITS statement is ignored for CLASS variables. If the CLODDS option is specified in the MODEL statement, the corresponding confidence limits for the odds ratios are also displayed.

The term *independent* is the name of an explanatory variable, and *list* represents a list of units of change, separated by spaces, that are of interest for that variable. Each unit of change in a list has one of the following forms:

- *number*
- SD or −SD
- *number* * SD

where *number* is any nonzero number and SD is the sample standard deviation of the corresponding independent variable. For example, $X = -2$ requests an odds ratio that represents the change in the odds when the variable X is decreased by two units. $X = 2*\text{SD}$ requests an estimate of the change in the odds when X is increased by two sample standard deviations.

You can specify the following option in the UNITS statement after a slash(/).

DEFAULT=*list*

gives a list of units of change for all explanatory variables that are not specified in the UNITS statement. Each unit of change can be in any of the forms described previously. If the DEFAULT= option is not specified, PROC SURVEYLOGISTIC does not produce customized odds ratio estimates for any explanatory variable that is not listed in the UNITS statement.

For more information, see the section "Odds Ratio Estimation" on page 6434.

WEIGHT Statement

WEIGHT *variable < / option >* **;**

The WEIGHT statement names the variable that contains the sampling weights. This variable must be numeric, and the sampling weights must be positive numbers. If an observation has a weight that is nonpositive or missing, then the procedure omits that observation from the analysis. See the section "Missing Values" on page 6410 for more information. If you specify more than one WEIGHT statement, the procedure uses only the first WEIGHT statement and ignores the rest.

If you do not specify a WEIGHT statement but provide replicate weights with a REPWEIGHTS statement, PROC SURVEYLOGISTIC uses the average of each observation's replicate weights as the observation's weight.

If you do not specify a WEIGHT statement or a REPWEIGHTS statement, PROC SURVEYLOGISTIC assigns all observations a weight of one.

Details: SURVEYLOGISTIC Procedure

Missing Values

If you have missing values in your survey data for any reason, such as nonresponse, this can compromise the quality of your survey results. If the respondents are different from the nonrespondents with regard to a survey effect or outcome, then survey estimates might be biased and cannot accurately represent the survey population. There are a variety of techniques in sample design and survey operations that can reduce nonresponse. After data collection is complete, you can use imputation to replace missing values with acceptable values, and/or you can use sampling weight adjustments to compensate for nonresponse. You should complete this data preparation and adjustment before you analyze your data with PROC SURVEYLOGISTIC. See Cochran (1977), Kalton and Kaspyzyk (1986), and Brick and Kalton (1996) for more information.

If an observation has a missing value or a nonpositive value for the WEIGHT or FREQ variable, then that observation is excluded from the analysis.

An observation is also excluded if it has a missing value for design variables such as STRATA variables, CLUSTER variables, and DOMAIN variables, unless missing values are regarded as a legitimate categorical level for these variables, as specified by the MISSING option.

By default, if an observation contains missing values for the response, offset, or any explanatory variables used in the independent effects, the observation is excluded from the analysis. This treatment is based on the assumption that the missing values are missing completely at random (MCAR). However, this assumption is not true sometimes. For example, evidence from other surveys might suggest that observations with missing values are systematically different from observations without missing values. If you believe that missing values are not missing completely at random, then you can specify the NOMCAR option to include these observations with missing values in the dependent variable and the independent variables in the variance estimation.

Whether or not the NOMCAR option is used, observations with missing or invalid values for WEIGHT, FREQ, STRATA, CLUSTER, or DOMAIN variables are always excluded, unless the MISSING option is also specified.

When you specify the NOMCAR option, the procedure treats observations with and without missing values for variables in the regression model as two different domains, and it performs a domain analysis in the domain of nonmissing observations.

If you use a REPWEIGHTS statement, all REPWEIGHTS variables must contain nonmissing values.

Model Specification

Response Level Ordering

Response level ordering is important because, by default, PROC SURVEYLOGISTIC models the probabilities of response levels with lower *Ordered Values*. Ordered Values, displayed in the "Response Profile" table, are assigned to response levels in ascending sorted order. That is, the lowest response level is assigned Ordered Value 1, the next lowest is assigned Ordered Value 2, and so on. For example, if your response variable Y takes values in $\{1, \ldots, D+1\}$, then the functions of the response probabilities modeled with the cumulative model are

$$\text{logit}(\Pr(Y \leq i|\mathbf{x})), i = 1, \ldots, D$$

and for the generalized logit model they are

$$\log\left(\frac{\Pr(Y = i|\mathbf{x})}{\Pr(Y = D+1|\mathbf{x})}\right), i = 1, \ldots, D$$

where the highest Ordered Value $Y = D+1$ is the reference level. You can change these default functions by specifying the EVENT=, REF=, DESCENDING, or ORDER= response variable options in the MODEL statement.

For binary response data with event and nonevent categories, the procedure models the function

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

where p is the probability of the response level assigned to Ordered Value 1 in the "Response Profiles" table. Since

$$\text{logit}(p) = -\text{logit}(1-p)$$

the effect of reversing the order of the two response levels is to change the signs of α and $\boldsymbol{\beta}$ in the model $\text{logit}(p) = \alpha + \mathbf{x}\boldsymbol{\beta}$.

If your event category has a higher Ordered Value than the nonevent category, the procedure models the nonevent probability. You can use response variable options to model the event probability. For example, suppose the binary response variable Y takes the values 1 and 0 for event and nonevent, respectively, and Exposure is the explanatory variable. By default, the procedure assigns Ordered Value 1 to response level Y=0, and Ordered Value 2 to response level Y=1. Therefore, the procedure models the probability of the nonevent (Ordered Value=1) category. To model the event probability, you can do the following:

- Explicitly state which response level is to be modeled by using the response variable option EVENT= in the MODEL statement:

```
model Y(event='1') = Exposure;
```

- Specify the response variable option DESCENDING in the MODEL statement:

```
model Y(descending)=Exposure;
```

- Specify the response variable option REF= in the MODEL statement as the nonevent category for the response variable. This option is most useful when you are fitting a generalized logit model.

```
model Y(ref='0') = Exposure;
```

- Assign a format to Y such that the first formatted value (when the formatted values are put in sorted order) corresponds to the event. For this example, Y=1 is assigned formatted value 'event' and Y=0 is assigned formatted value 'nonevent.' Since ORDER=FORMATTED by default, Ordered Value 1 is assigned to response level Y=1 so the procedure models the event.

```
proc format;
   value Disease 1='event' 0='nonevent';
run;
proc surveylogistic;
   format Y Disease.;
   model Y=Exposure;
run;
```

CLASS Variable Parameterization

Consider a model with one CLASS variable A with four levels: 1, 2, 5, and 7. Details of the possible choices for the PARAM= option follow.

EFFECT — Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of −1. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

	Design Matrix		
A	**A1**	**A2**	**A5**
1	1	0	0
2	0	1	0
5	0	0	1
7	−1	−1	−1

For CLASS main effects that use the EFFECT coding scheme, individual parameters correspond to the difference between the effect of each nonreference level and the average over all four levels.

GLM — As in PROC GLM, four columns are created to indicate group membership. The design matrix columns for A are as follows.

	Design Matrix			
A	**A1**	**A2**	**A5**	**A7**
1	1	0	0	0
2	0	1	0	0
5	0	0	1	0
7	0	0	0	1

For CLASS main effects that use the GLM coding scheme, individual parameters correspond to the difference between the effect of each level and the last level.

ORDINAL

Three columns are created to indicate group membership of the higher levels of the effect. For the first level of the effect (which for A is 1), all three dummy variables have a value of 0. The design matrix columns for A are as follows.

	Design Matrix		
A	**A2**	**A5**	**A7**
1	0	0	0
2	1	0	0
5	1	1	0
7	1	1	1

The first level of the effect is a control or baseline level.

For CLASS main effects that use the ORDINAL coding scheme, the first level of the effect is a control or baseline level; individual parameters correspond to the difference between effects of the current level and the preceding level. When the parameters for an ordinal main effect have the same sign, the response effect is monotonic across the levels.

POLYNOMIAL

POLY

Three columns are created. The first represents the linear term (x), the second represents the quadratic term (x^2), and the third represents the cubic term (x^3), where x is the level value. If the CLASS levels are not numeric, they are translated into 1, 2, 3, ... according to their sorting order. The design matrix columns for A are as follows.

	Design Matrix		
A	**APOLY1**	**APOLY2**	**APOLY3**
1	1	1	1
2	2	4	8
5	5	25	125
7	7	49	343

REFERENCE

REF

Three columns are created to indicate group membership of the nonreference levels. For the reference level, all three dummy variables have a value of 0. For instance, if the reference level is 7 (REF=7), the design matrix columns for A are as follows.

	Design Matrix		
A	A1	A2	A5
1	1	0	0
2	0	1	0
5	0	0	1
7	0	0	0

For CLASS main effects that use the REFERENCE coding scheme, individual parameters correspond to the difference between the effect of each nonreference level and the reference level.

ORTHEFFECT The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=EFFECT. The design matrix columns for A are as follows.

	Design Matrix		
A	AOEFF1	AOEFF2	AOEFF3
1	1.41421	−0.81650	−0.57735
2	0.00000	1.63299	−0.57735
5	0.00000	0.00000	1.73205
7	−1.41421	−0.81649	−0.57735

ORTHORDINAL

ORTHOTHERM The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=ORDINAL. The design matrix columns for A are as follows.

	Design Matrix		
A	AOORD1	AOORD2	AOORD3
1	−1.73205	0.00000	0.00000
2	0.57735	−1.63299	0.00000
5	0.57735	0.81650	−1.41421
7	0.57735	0.81650	1.41421

ORTHPOLY The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=POLY. The design matrix columns for A are as follows.

	Design Matrix		
A	AOPOLY1	AOPOLY2	AOPOLY5
1	−1.153	0.907	−0.921
2	−0.734	−0.540	1.473
5	0.524	−1.370	−0.921
7	1.363	1.004	0.368

ORTHREF The columns are obtained by applying the Gram-Schmidt orthogonalization to the columns for PARAM=REFERENCE. The design matrix columns for A are as follows.

	Design Matrix		
A	AOREF1	AOREF2	AOREF3
1	1.73205	0.00000	0.00000
2	−0.57735	1.63299	0.00000
5	−0.57735	−0.81650	1.41421
7	−0.57735	−0.81650	−1.41421

Link Functions and the Corresponding Distributions

Four link functions are available in the SURVEYLOGISTIC procedure. The logit function is the default. To specify a different link function, use the LINK= option in the MODEL statement. The link functions and the corresponding distributions are as follows:

- The **logit** function

$$g(p) = \log\left(\frac{p}{1-p}\right)$$

is the inverse of the cumulative logistic distribution function, which is

$$F(x) = \frac{1}{1+e^{-x}}$$

- The **probit** (or normit) function

$$g(p) = \Phi^{-1}(p)$$

is the inverse of the cumulative standard normal distribution function, which is

$$F(x) = \Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}z^2} dz$$

Traditionally, the probit function includes an additive constant 5, but throughout PROC SURVEYLOGISTIC, the terms probit and normit are used interchangeably, previously defined as $g(p)$.

- The **complementary log-log** function

$$g(p) = \log(-\log(1-p))$$

is the inverse of the cumulative extreme-value function (also called the Gompertz distribution), which is

$$F(x) = 1 - e^{-e^x}$$

- The **generalized logit** function extends the binary logit link to a vector of levels $(\pi_1, \ldots, \pi_{k+1})$ by contrasting each level with a fixed level

$$g(\pi_i) = \log\left(\frac{\pi_i}{\pi_{k+1}}\right) \quad i = 1, \ldots, k$$

The variances of the normal, logistic, and extreme-value distributions are not the same. Their respective means and variances are

Distribution	Mean	Variance
Normal	0	1
Logistic	0	$\pi^2/3$
Extreme-value	$-\gamma$	$\pi^2/6$

where γ is the Euler constant. In comparing parameter estimates that use different link functions, you need to take into account the different scalings of the corresponding distributions and, for the complementary log-log function, a possible shift in location. For example, if the fitted probabilities are in the neighborhood of 0.1 to 0.9, then the parameter estimates from using the logit link function should be about $\pi/\sqrt{3} \approx 1.8$ larger than the estimates from the probit link function.

Model Fitting

Determining Observations for Likelihood Contributions

If you use the events/trials syntax, each observation is split into two observations. One has the response value 1 with a frequency equal to the frequency of the original observation (which is 1 if the FREQ statement is not used) times the value of the *events* variable. The other observation has the response value 2 and a frequency equal to the frequency of the original observation times the value of (*trials* − *events*). These two observations have the same explanatory variable values and the same FREQ and WEIGHT values as the original observation.

For either the single-trial or the events/trials syntax, let j index all observations. In other words, for the single-trial syntax, j indexes the actual observations. And, for the events/trials syntax, j indexes the observations after splitting (as described previously). If your data set has 30 observations and you use the single-trial syntax, j has values from 1 to 30; if you use the events/trials syntax, j has values from 1 to 60.

Suppose the response variable in a cumulative response model can take on the ordered values $1, \ldots, k, k+1$, where k is an integer ≥ 1. The likelihood for the jth observation with ordered response value y_j and explanatory variables vector (row vectors) $\mathbf{x}_j$ is given by

$$L_j = \begin{cases} F(\alpha_1 + \mathbf{x}_j\boldsymbol{\beta}) & y_j = 1 \\ F(\alpha_i + \mathbf{x}_j\boldsymbol{\beta}) - F(\alpha_{i-1} + \mathbf{x}_j\boldsymbol{\beta}) & 1 < y_j = i \leq k \\ 1 - F(\alpha_k + \mathbf{x}_j\boldsymbol{\beta}) & y_j = k+1 \end{cases}$$

where $F(.)$ is the logistic, normal, or extreme-value distribution function; $\alpha_1, \ldots, \alpha_k$ are ordered intercept parameters; and $\boldsymbol{\beta}$ is the slope parameter vector.

For the generalized logit model, letting the $k+1$st level be the reference level, the intercepts $\alpha_1, \ldots, \alpha_k$ are unordered and the slope vector $\boldsymbol{\beta}_i$ varies with each logit. The likelihood for the jth observation with ordered response value y_j and explanatory variables vector $\mathbf{x}_j$ (row vectors) is given by

$$
\begin{aligned}
L_j &= \Pr(Y = y_j | \mathbf{x}_j) \\
&= \begin{cases} \dfrac{e^{\alpha_i + \mathbf{x}_j \boldsymbol{\beta}_i}}{1 + \sum_{i=1}^{k} e^{\alpha_i + \mathbf{x}_j \boldsymbol{\beta}_i}} & 1 \le y_j = i \le k \\ \dfrac{1}{1 + \sum_{i=1}^{k} e^{\alpha_i + \mathbf{x}_j \boldsymbol{\beta}_i}} & y_j = k+1 \end{cases}
\end{aligned}
$$

Iterative Algorithms for Model Fitting

Two iterative maximum likelihood algorithms are available in PROC SURVEYLOGISTIC to obtain the pseudo-estimate $\hat{\boldsymbol{\theta}}$ of the model parameter $\boldsymbol{\theta}$. The default is the Fisher scoring method, which is equivalent to fitting by iteratively reweighted least squares. The alternative algorithm is the Newton-Raphson method. Both algorithms give the same parameter estimates; the covariance matrix of $\hat{\boldsymbol{\theta}}$ is estimated in the section "Variance Estimation" on page 6427. For a generalized logit model, only the Newton-Raphson technique is available. You can use the TECHNIQUE= option to select a fitting algorithm.

Iteratively Reweighted Least Squares Algorithm (Fisher Scoring)

Let Y be the response variable that takes values $1, \ldots, k, k+1$ ($k \ge 1$). Let j index all observations and Y_j be the value of response for the jth observation. Consider the multinomial variable $\mathbf{Z}_j = (Z_{1j}, \ldots, Z_{kj})'$ such that

$$
Z_{ij} = \begin{cases} 1 & \text{if } Y_j = i \\ 0 & \text{otherwise} \end{cases}
$$

and $Z_{(k+1)j} = 1 - \sum_{i=1}^{k} Z_{ij}$. With π_{ij} denoting the probability that the jth observation has response value i, the expected value of $\mathbf{Z}_j$ is $\boldsymbol{\pi}_j = (\pi_{1j}, \ldots, \pi_{kj})'$, and $\pi_{(k+1)j} = 1 - \sum_{i=1}^{k} \pi_{ij}$. The covariance matrix of $\mathbf{Z}_j$ is $\mathbf{V}_j$, which is the covariance matrix of a multinomial random variable for one trial with parameter vector $\boldsymbol{\pi}_j$. Let $\boldsymbol{\theta}$ be the vector of regression parameters—for example, $\boldsymbol{\theta} = (\alpha_1, \ldots, \alpha_k, \boldsymbol{\beta}')'$ for cumulative logit model. Let $\mathbf{D}_j$ be the matrix of partial derivatives of $\boldsymbol{\pi}_j$ with respect to $\boldsymbol{\theta}$. The estimating equation for the regression parameters is

$$
\sum_j \mathbf{D}_j' \mathbf{W}_j (\mathbf{Z}_j - \boldsymbol{\pi}_j) = \mathbf{0}
$$

where $\mathbf{W}_j = w_j f_j \mathbf{V}_j^{-1}$, and w_j and f_j are the WEIGHT and FREQ values of the jth observation.

With a starting value of $\boldsymbol{\theta}^{(0)}$, the pseudo-estimate of $\boldsymbol{\theta}$ is obtained iteratively as

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} + (\sum_j \mathbf{D}_j' \mathbf{W}_j \mathbf{D}_j)^{-1} \sum_j \mathbf{D}_j' \mathbf{W}_j (\mathbf{Z}_j - \boldsymbol{\pi}_j)$$

where $\mathbf{D}_j$, $\mathbf{W}_j$, and $\boldsymbol{\pi}_j$ are evaluated at the ith iteration $\boldsymbol{\theta}^{(i)}$. The expression after the plus sign is the step size. If the log likelihood evaluated at $\boldsymbol{\theta}^{(i+1)}$ is less than that evaluated at $\boldsymbol{\theta}^{(i)}$, then $\boldsymbol{\theta}^{(i+1)}$ is recomputed by step-halving or ridging. The iterative scheme continues until convergence is obtained—that is, until $\boldsymbol{\theta}^{(i+1)}$ is sufficiently close to $\boldsymbol{\theta}^{(i)}$. Then the maximum likelihood estimate of $\boldsymbol{\theta}$ is $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(i+1)}$.

By default, starting values are zero for the slope parameters, and starting values are the observed cumulative logits (that is, logits of the observed cumulative proportions of response) for the intercept parameters. Alternatively, the starting values can be specified with the INEST= option.

Newton-Raphson Algorithm

Let

$$\mathbf{g} = \sum_j w_j f_j \frac{\partial l_j}{\partial \boldsymbol{\theta}}$$

$$\mathbf{H} = \sum_j -w_j f_j \frac{\partial^2 l_j}{\partial \boldsymbol{\theta}^2}$$

be the gradient vector and the Hessian matrix, where $l_j = \log L_j$ is the log likelihood for the jth observation. With a starting value of $\boldsymbol{\theta}^{(0)}$, the pseudo-estimate $\hat{\boldsymbol{\theta}}$ of $\boldsymbol{\theta}$ is obtained iteratively until convergence is obtained:

$$\boldsymbol{\theta}^{(i+1)} = \boldsymbol{\theta}^{(i)} + \mathbf{H}^{-1}\mathbf{g}$$

where $\mathbf{H}$ and $\mathbf{g}$ are evaluated at the ith iteration $\boldsymbol{\theta}^{(i)}$. If the log likelihood evaluated at $\boldsymbol{\theta}^{(i+1)}$ is less than that evaluated at $\boldsymbol{\theta}^{(i)}$, then $\boldsymbol{\theta}^{(i+1)}$ is recomputed by step-halving or ridging. The iterative scheme continues until convergence is obtained—that is, until $\boldsymbol{\theta}^{(i+1)}$ is sufficiently close to $\boldsymbol{\theta}^{(i)}$. Then the maximum likelihood estimate of $\boldsymbol{\theta}$ is $\hat{\boldsymbol{\theta}} = \boldsymbol{\theta}^{(i+1)}$.

Convergence Criteria

Four convergence criteria are allowed: ABSFCONV=, FCONV=, GCONV=, and XCONV=. If you specify more than one convergence criterion, the optimization is terminated as soon as one of the criteria is satisfied. If none of the criteria is specified, the default is GCONV=1E−8.

Existence of Maximum Likelihood Estimates

The likelihood equation for a logistic regression model does not always have a finite solution. Sometimes there is a nonunique maximum on the boundary of the parameter space, at infinity. The ex-

istence, finiteness, and uniqueness of pseudo-estimates for the logistic regression model depend on the patterns of data points in the observation space (Albert and Anderson 1984; Santner and Duffy 1986).

Consider a binary response model. Let Y_j be the response of the ith subject, and let $\mathbf{x}_j$ be the row vector of explanatory variables (including the constant 1 associated with the intercept). There are three mutually exclusive and exhaustive types of data configurations: complete separation, quasi-complete separation, and overlap.

Complete separation

There is a complete separation of data points if there exists a vector $\mathbf{b}$ that correctly allocates all observations to their response groups; that is,

$$\begin{cases} \mathbf{x}_j\mathbf{b} > 0 & Y_j = 1 \\ \mathbf{x}_j\mathbf{b} < 0 & Y_j = 2 \end{cases}$$

This configuration gives nonunique infinite estimates. If the iterative process of maximizing the likelihood function is allowed to continue, the log likelihood diminishes to zero, and the dispersion matrix becomes unbounded.

Quasi-complete separation

The data are not completely separable, but there is a vector $\mathbf{b}$ such that

$$\begin{cases} \mathbf{x}_j\mathbf{b} \geq 0 & Y_j = 1 \\ \mathbf{x}_j\mathbf{b} \leq 0 & Y_j = 2 \end{cases}$$

and equality holds for at least one subject in each response group. This configuration also yields nonunique infinite estimates. If the iterative process of maximizing the likelihood function is allowed to continue, the dispersion matrix becomes unbounded and the log likelihood diminishes to a nonzero constant.

Overlap

If neither complete nor quasi-complete separation exists in the sample points, there is an overlap of sample points. In this configuration, the pseudo-estimates exist and are unique.

Complete separation and quasi-complete separation are problems typically encountered with small data sets. Although complete separation can occur with any type of data, quasi-complete separation is not likely with truly continuous explanatory variables.

The SURVEYLOGISTIC procedure uses a simple empirical approach to recognize the data configurations that lead to infinite parameter estimates. The basis of this approach is that any convergence method of maximizing the log likelihood must yield a solution that gives complete separation, if such a solution exists. In maximizing the log likelihood, there is no checking for complete or quasi-complete separation if convergence is attained in eight or fewer iterations. Subsequent to the eighth iteration, the probability of the observed response is computed for each observation. If the probability of the observed response is one for all observations, there is a complete separation of data points and the iteration process is stopped. If the complete separation of data has not been determined and an observation is identified to have an extremely large probability ($\geq$0.95) of the observed response, there are two possible situations. First, there is overlap in the data set, and the observation is an atypical observation of its own group. The iterative process, if allowed to continue, stops

when a maximum is reached. Second, there is quasi-complete separation in the data set, and the asymptotic dispersion matrix is unbounded. If any of the diagonal elements of the dispersion matrix for the standardized observations vectors (all explanatory variables standardized to zero mean and unit variance) exceeds 5,000, quasi-complete separation is declared and the iterative process is stopped. If either complete separation or quasi-complete separation is detected, a warning message is displayed in the procedure output.

Checking for quasi-complete separation is less foolproof than checking for complete separation. The NOCHECK option in the MODEL statement turns off the process of checking for infinite parameter estimates. In cases of complete or quasi-complete separation, turning off the checking process typically results in the procedure failing to converge.

Model Fitting Statistics

Suppose the model contains s explanatory effects. For the jth observation, let $\hat{\pi}_j$ be the estimated probability of the observed response. The three criteria displayed by the SURVEYLOGISTIC procedure are calculated as follows:

- -2 log likelihood:

$$-2 \text{ Log L} = -2 \sum_j w_j f_j \log(\hat{\pi}_j)$$

  where w_j and f_j are the weight and frequency values, respectively, of the jth observation. For binary response models that use the events/trials syntax, this is equivalent to

$$-2 \text{ Log L} = -2 \sum_j w_j f_j \{r_j \log(\hat{\pi}_j) + (n_j - r_j) \log(1 - \hat{\pi}_j)\}$$

  where r_j is the number of events, n_j is the number of trials, and $\hat{\pi}_j$ is the estimated event probability.

- Akaike information criterion:

$$\text{AIC} = -2 \text{ Log L} + 2p$$

  where p is the number of parameters in the model. For cumulative response models, $p = k + s$, where k is the total number of response levels minus one, and s is the number of explanatory effects. For the generalized logit model, $p = k(s + 1)$.

- Schwarz criterion:

$$\text{SC} = -2 \text{ Log L} + p \log(\sum_j f_j)$$

  where p is the number of parameters in the model. For cumulative response models, $p = k + s$, where k is the total number of response levels minus one, and s is the number of explanatory effects. For the generalized logit model, $p = k(s + 1)$.

The −2 log likelihood statistic has a chi-square distribution under the null hypothesis (that all the explanatory effects in the model are zero), and the procedure produces a p-value for this statistic. The AIC and SC statistics give two different ways of adjusting the −2 log likelihood statistic for the number of terms in the model and the number of observations used.

Generalized Coefficient of Determination

Cox and Snell (1989, pp. 208–209) propose the following generalization of the coefficient of determination to a more general linear model:

$$R^2 = 1 - \left\{ \frac{L(\mathbf{0})}{L(\hat{\boldsymbol{\theta}})} \right\}^{\frac{2}{n}}$$

where $L(\mathbf{0})$ is the likelihood of the intercept-only model, $L(\hat{\boldsymbol{\theta}})$ is the likelihood of the specified model, and n is the sample size. The quantity R^2 achieves a maximum of less than 1 for discrete models, where the maximum is given by

$$R^2_{\max} = 1 - \{L(\mathbf{0})\}^{\frac{2}{n}}$$

Nagelkerke (1991) proposes the following adjusted coefficient, which can achieve a maximum value of 1:

$$\tilde{R}^2 = \frac{R^2}{R^2_{\max}}$$

Properties and interpretation of R^2 and $\tilde{R}^2$ are provided in Nagelkerke (1991). In the "Testing Global Null Hypothesis: BETA=0" table, R^2 is labeled as "RSquare" and $\tilde{R}^2$ is labeled as "Max-rescaled RSquare." Use the RSQUARE option to request R^2 and $\tilde{R}^2$.

INEST= Data Set

You can specify starting values for the iterative algorithm in the INEST= data set.

The INEST= data set contains one observation for each BY group. The INEST= data set must contain the intercept variables (named Intercept for binary response models and Intercept, Intercept2, Intercept3, and so forth, for ordinal response models) and all explanatory variables in the MODEL statement. If BY processing is used, the INEST= data set should also include the BY variables, and there must be one observation for each BY group. If the INEST= data set also contains the _TYPE_ variable, only observations with _TYPE_ value 'PARMS' are used as starting values.

Survey Design Information

Specification of Population Totals and Sampling Rates

To include a finite population correction (*fpc*) in Taylor series variance estimation, you can input either the sampling rate or the population total by using the RATE= or TOTAL= option in the PROC SURVEYLOGISTIC statement. (You cannot specify both of these options in the same PROC SURVEYLOGISTIC statement.) The RATE= and TOTAL= options apply only to Taylor series variance estimation. The procedure does not use a finite population correction for BRR or jackknife variance estimation.

If you do not specify the RATE= or TOTAL= option, the Taylor series variance estimation does not include a finite population correction. For fairly small sampling fractions, it is appropriate to ignore this correction. See Cochran (1977) and Kish (1965) for more information.

If your design has multiple stages of selection and you are specifying the RATE= option, you should input the first-stage sampling rate, which is the ratio of the number of PSUs in the sample to the total number of PSUs in the study population. If you are specifying the TOTAL= option for a multistage design, you should input the total number of PSUs in the study population. See the section "Primary Sampling Units (PSUs)" on page 6423 for more details.

For a nonstratified sample design, or for a stratified sample design with the same sampling rate or the same population total in all strata, you can use the RATE=*value* or TOTAL=*value* option. If your sample design is stratified with different sampling rates or population totals in different strata, use the RATE=*SAS-data-set* or TOTAL=*SAS-data-set* option to name a SAS data set that contains the stratum sampling rates or totals. This data set is called a *secondary data set*, as opposed to the *primary data set* that you specify with the DATA= option.

The secondary data set must contain all the stratification variables listed in the STRATA statement and all the variables in the BY statement. If there are formats associated with the STRATA variables and the BY variables, then the formats must be consistent in the primary and the secondary data sets. If you specify the TOTAL=*SAS-data-set* option, the secondary data set must have a variable named _TOTAL_ that contains the stratum population totals. Or if you specify the RATE=*SAS-data-set* option, the secondary data set must have a variable named _RATE_ that contains the stratum sampling rates. If the secondary data set contains more than one observation for any one stratum, then the procedure uses the first value of _TOTAL_ or _RATE_ for that stratum and ignores the rest.

The *value* in the RATE= option or the values of _RATE_ in the secondary data set must be non-negative numbers. You can specify *value* as a number between 0 and 1. Or you can specify *value* in percentage form as a number between 1 and 100, and PROC SURVEYLOGISTIC converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

If you specify the TOTAL=*value* option, *value* must not be less than the sample size. If you provide stratum population totals in a secondary data set, these values must not be less than the corresponding stratum sample sizes.

Primary Sampling Units (PSUs)

When you have clusters, or primary sampling units (PSUs), in your sample design, the procedure estimates variance from the variation among PSUs when the Taylor series variance method is used. See the section "Taylor Series (Linearization)" on page 6427 for more information.

BRR or jackknife variance estimation methods draw multiple replicates (or subsamples) from the full sample by following a specific resampling scheme. These subsamples are constructed by deleting PSUs from the full sample.

If you use a REPWEIGHTS statement to provide replicate weights for BRR or jackknife variance estimation, you do not need to specify a CLUSTER statement. Otherwise, you should specify a CLUSTER statement whenever your design includes clustering at the first stage of sampling. If you do not specify a CLUSTER statement, then PROC SURVEYLOGISTIC treats each observation as a PSU.

Logistic Regression Models and Parameters

The SURVEYLOGISTIC procedure fits a logistic regression model and estimates the corresponding regression parameters. Each model uses the link function you specified in the LINK= option in the MODEL statement. There are four types of model you can use with the procedure: cumulative logit model, complementary log-log model, probit model, and generalized logit model.

Notation

Let Y be the response variable with categories $1, 2, \ldots, D, D+1$. The p covariates are denoted by a p-dimension row vector $\mathbf{x}$.

For a stratified clustered sample design, each observation is represented by a row vector, $(w_{hij}, \mathbf{y}'_{hij}, y_{hij(D+1)}, \mathbf{x}_{hij})$, where

- $h = 1, 2, \ldots, H$ is the stratum index
- $i = 1, 2, \ldots, n_h$ is the cluster index within stratum h
- $j = 1, 2, \ldots, m_{hi}$ is the unit index within cluster i of stratum h
- w_{hij} denotes the sampling weight
- $\mathbf{y}_{hij}$ is a D-dimensional column vector whose elements are indicator variables for the first D categories for variable Y. If the response of the jth unit of the ith cluster in stratum h falls in category d, the dth element of the vector is one, and the remaining elements of the vector are zero, where $d = 1, 2, \ldots, D$.
- $y_{hij(D+1)}$ is the indicator variable for the $(D+1)$ category of variable Y
- $\mathbf{x}_{hij}$ denotes the k-dimensional row vector of explanatory variables for the jth unit of the ith cluster in stratum h. If there is an intercept, then $x_{hij1} \equiv 1$.

- $\tilde{n} = \sum_{h=1}^{H} n_h$ is the total number of clusters in the sample
- $n = \sum_{h=1}^{H} \sum_{i=1}^{n_h} m_{hi}$ is the total sample size

The following notations are also used:

- f_h denotes the sampling rate for stratum h
- $\boldsymbol{\pi}_{hij}$ is the expected vector of the response variable:

$$\begin{aligned} \boldsymbol{\pi}_{hij} &= E(\mathbf{y}_{hij}|\mathbf{x}_{hij}) \\ &= (\pi_{hij1}, \pi_{hij2}, \ldots, \pi_{hijD})' \\ \pi_{hij(D+1)} &= E(y_{hij(D+1)}|\mathbf{x}_{hij}) \end{aligned}$$

Note that $\pi_{hij(D+1)} = 1 - \mathbf{1}'\boldsymbol{\pi}_{hij}$, where $\mathbf{1}$ is a D-dimensional column vector whose elements are 1.

Logistic Regression Models

If the response categories of the response variable Y can be restricted to a number of ordinal values, you can fit cumulative probabilities of the response categories with a cumulative logit model, a complementary log-log model, or a probit model. Details of cumulative logit models (or proportional odds models) can be found in McCullagh and Nelder (1989). If the response categories of Y are nominal responses without natural ordering, you can fit the response probabilities with a generalized logit model. Formulation of the generalized logit models for nominal response variables can be found in Agresti (1990). For each model, the procedure estimates the model parameter $\boldsymbol{\theta}$ by using a pseudo-log-likelihood function. The procedure obtains the pseudo-maximum likelihood estimator $\hat{\boldsymbol{\theta}}$ by using iterations described in the section "Iterative Algorithms for Model Fitting" on page 6417 and estimates its variance described in the section "Variance Estimation" on page 6427.

Cumulative Logit Model

A cumulative logit model uses the **logit** function

$$g(t) = \log\left(\frac{t}{1-t}\right)$$

as the link function.

Denote the cumulative sum of the expected proportions for the first d categories of variable Y by

$$F_{hijd} = \sum_{r=1}^{d} \pi_{hijr}$$

for $d = 1, 2, \ldots, D$. Then the cumulative logit model can be written as

$$\log\left(\frac{F_{hijd}}{1 - F_{hijd}}\right) = \alpha_d + \mathbf{x}_{hij}\boldsymbol{\beta}$$

with the model parameters

$$\begin{aligned}\boldsymbol{\beta} &= (\beta_1, \beta_2, \ldots, \beta_k)' \\ \boldsymbol{\alpha} &= (\alpha_1, \alpha_2, \ldots, \alpha_D)', \quad \alpha_1 < \alpha_2 < \cdots < \alpha_D \\ \boldsymbol{\theta} &= (\boldsymbol{\alpha}', \boldsymbol{\beta}')'\end{aligned}$$

Complementary Log-Log Model

A complementary log-log model uses the **complementary log-log** function

$$g(t) = \log(-\log(1-t))$$

as the link function. Denote the cumulative sum of the expected proportions for the first d categories of variable Y by

$$F_{hijd} = \sum_{r=1}^{d} \pi_{hijr}$$

for $d = 1, 2, \ldots, D$. Then the complementary log-log model can be written as

$$\log(-\log(1 - F_{hijd})) = \alpha_d + \mathbf{x}_{hij}\boldsymbol{\beta}$$

with the model parameters

$$\begin{aligned}\boldsymbol{\beta} &= (\beta_1, \beta_2, \ldots, \beta_k)' \\ \boldsymbol{\alpha} &= (\alpha_1, \alpha_2, \ldots, \alpha_D)', \quad \alpha_1 < \alpha_2 < \cdots < \alpha_D \\ \boldsymbol{\theta} &= (\boldsymbol{\alpha}', \boldsymbol{\beta}')'\end{aligned}$$

Probit Model

A probit model uses the **probit** (or normit) function, which is the inverse of the cumulative standard normal distribution function,

$$g(t) = \Phi^{-1}(t)$$

as the link function, where

$$\Phi(t) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{t} e^{-\frac{1}{2}z^2} dz$$

Denote the cumulative sum of the expected proportions for the first d categories of variable Y by

$$F_{hijd} = \sum_{r=1}^{d} \pi_{hijr}$$

for $d = 1, 2, \ldots, D$. Then the probit model can be written as

$$F_{hijd} = \Phi(\alpha_d + \mathbf{x}_{hij}\boldsymbol{\beta})$$

with the model parameters

$$\begin{aligned}\boldsymbol{\beta} &= (\beta_1, \beta_2, \ldots, \beta_k)' \\ \boldsymbol{\alpha} &= (\alpha_1, \alpha_2, \ldots, \alpha_D)', \quad \alpha_1 < \alpha_2 < \cdots < \alpha_D \\ \boldsymbol{\theta} &= (\boldsymbol{\alpha}', \boldsymbol{\beta}')'\end{aligned}$$

Generalized Logit Model

For nominal response, a generalized logit model is to fit the ratio of the expected proportion for each response category over the expected proportion of a reference category with a logit link function.

Without loss of generality, let category $D+1$ be the reference category for the response variable Y. Denote the expected proportion for the dth category by π_{hijd} as in the section "Notation" on page 6423. Then the generalized logit model can be written as

$$\log\left(\frac{\pi_{hijd}}{\pi_{hij(D+1)}}\right) = \mathbf{x}_{hij}\boldsymbol{\beta}_d$$

for $d = 1, 2, \ldots, D$, with the model parameters

$$\begin{aligned}\boldsymbol{\beta}_d &= (\beta_{d1}, \beta_{d2}, \ldots, \beta_{dk})' \\ \boldsymbol{\theta} &= (\boldsymbol{\beta}_1', \boldsymbol{\beta}_2', \ldots, \boldsymbol{\beta}_D')'\end{aligned}$$

Likelihood Function

Let $\mathbf{g}(\cdot)$ be a link function such that

$$\boldsymbol{\pi} = \mathbf{g}(\mathbf{x}, \boldsymbol{\theta})$$

where $\boldsymbol{\theta}$ is a column vector for regression coefficients. The pseudo-log likelihood is

$$l(\boldsymbol{\theta}) = \sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}\left((\log(\boldsymbol{\pi}_{hij}))'\mathbf{y}_{hij} + \log(\pi_{hij(D+1)})y_{hij(D+1)}\right)$$

Denote the pseudo-estimator as $\hat{\boldsymbol{\theta}}$, which is a solution to the estimating equations:

$$\sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}\mathbf{D}_{hij}\left(\mathrm{diag}(\boldsymbol{\pi}_{hij}) - \boldsymbol{\pi}_{hij}\boldsymbol{\pi}_{hij}'\right)^{-1}(\mathbf{y}_{hij} - \boldsymbol{\pi}_{hij}) = \mathbf{0}$$

where $\mathbf{D}_{hij}$ is the matrix of partial derivatives of the link function $\mathbf{g}$ with respect to $\boldsymbol{\theta}$.

To obtain the pseudo-estimator $\hat{\boldsymbol{\theta}}$, the procedure uses iterations with a starting value $\boldsymbol{\theta}^{(0)}$ for $\boldsymbol{\theta}$. See the section "Iterative Algorithms for Model Fitting" on page 6417 for more details.

Variance Estimation

Due to the variability of characteristics among items in the population, researchers apply scientific sample designs in the sample selection process to reduce the risk of a distorted view of the population, and they make inferences about the population based on the information from the sample survey data. In order to make statistically valid inferences for the population, they must incorporate the sample design in the data analysis.

The SURVEYLOGISTIC procedure fits linear logistic regression models for discrete response survey data by using the maximum likelihood method. In the variance estimation, the procedure uses the Taylor series (linearization) method or replication (resampling) methods to estimate sampling errors of estimators based on complex sample designs, including designs with stratification, clustering, and unequal weighting (Binder (1981, 1983); Roberts, Rao, and Kumar (1987); Skinner, Holt, and Smith (1989); Binder and Roberts (2003); Morel (1989); Lehtonen and Pahkinen (1995); Woodruff (1971); Fuller (1975); Särndal, Swensson, and Wretman (1992); Wolter (1985); Rust (1985); Dippo, Fay, and Morganstein (1984); Rao and Shao (1999); Rao, Wu, and Yue (1992); and Rao and Shao (1996)).

You can use the VARMETHOD= option to specify a variance estimation method to use. By default, the Taylor series method is used. However, replication methods have recently gained popularity for estimating variances in complex survey data analysis. One reason for this popularity is the relative simplicity of replication-based estimates, especially for nonlinear estimators; another is that modern computational capacity has made replication methods feasible for practical survey analysis.

Replication methods draw multiple replicates (also called subsamples) from a full sample according to a specific resampling scheme. The most commonly used resampling schemes are the *balanced repeated replication* (BRR) method and the *jackknife* method. For each replicate, the original weights are modified for the PSUs in the replicates to create replicate weights. The parameters of interest are estimated by using the replicate weights for each replicate. Then the variances of parameters of interest are estimated by the variability among the estimates derived from these replicates. You can use the REPWEIGHTS statement to provide your own replicate weights for variance estimation.

The following sections provide details about how the variance-covariance matrix of the estimated regression coefficients is estimated for each variance estimation method.

Taylor Series (Linearization)

The Taylor series (linearization) method is the most commonly used method to estimate the covariance matrix of the regression coefficients for complex survey data. It is the default variance estimation method used by PROC SURVEYLOGISTIC.

Using the notation described in the section "Notation" on page 6423, the estimated covariance matrix of model parameters $\hat{\boldsymbol{\theta}}$ by the Taylor series method is

$$\widehat{V}(\hat{\boldsymbol{\theta}}) = \widehat{\mathbf{Q}}^{-1}\widehat{\mathbf{G}}\widehat{\mathbf{Q}}^{-1}$$

where

$$
\begin{aligned}
\widehat{\mathbf{Q}} &= \sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}\widehat{\mathbf{D}}_{hij}\left(\mathrm{diag}(\hat{\boldsymbol{\pi}}_{hij}) - \hat{\boldsymbol{\pi}}_{hij}\hat{\boldsymbol{\pi}}'_{hij}\right)^{-1}\widehat{\mathbf{D}}'_{hij} \\
\widehat{\mathbf{G}} &= \frac{n-1}{n-p}\sum_{h=1}^{H}\frac{n_h(1-f_h)}{n_h-1}\sum_{i=1}^{n_h}(\mathbf{e}_{hi\cdot} - \bar{\mathbf{e}}_{h\cdot\cdot})(\mathbf{e}_{hi\cdot} - \bar{\mathbf{e}}_{h\cdot\cdot})' \\
\mathbf{e}_{hi\cdot} &= \sum_{j=1}^{m_{hi}} w_{hij}\widehat{\mathbf{D}}_{hij}\left(\mathrm{diag}(\hat{\boldsymbol{\pi}}_{hij}) - \hat{\boldsymbol{\pi}}_{hij}\hat{\boldsymbol{\pi}}'_{hij}\right)^{-1}(\mathbf{y}_{hij} - \hat{\boldsymbol{\pi}}_{hij}) \\
\bar{\mathbf{e}}_{h\cdot\cdot} &= \frac{1}{n_h}\sum_{i=1}^{n_h}\mathbf{e}_{hi\cdot}
\end{aligned}
$$

and $\mathbf{D}_{hij}$ is the matrix of partial derivatives of the link function $\mathbf{g}$ with respect to $\boldsymbol{\theta}$ and $\widehat{\mathbf{D}}_{hij}$ and the response probabilities $\hat{\boldsymbol{\pi}}_{hij}$ are evaluated at $\hat{\boldsymbol{\theta}}$.

If you specify the TECHNIQUE=NEWTON option in the MODEL statement to request the Newton-Raphson algorithm, the matrix $\widehat{\mathbf{Q}}$ is replaced by the negative (expected) Hessian matrix when the estimated covariance matrix $\widehat{V}(\hat{\boldsymbol{\theta}})$ is computed.

Adjustments to the Variance Estimation

The factor $(n-1)/(n-p)$ in the computation of the matrix $\widehat{\mathbf{G}}$ reduces the small sample bias associated with using the estimated function to calculate deviations (Morel 1989; Hidiroglou, Fuller, and Hickman 1980). For simple random sampling, this factor contributes to the degrees-of-freedom correction applied to the residual mean square for ordinary least squares in which p parameters are estimated. By default, the procedure uses this adjustment in Taylor series variance estimation. It is equivalent to specifying the VADJUST=DF option in the MODEL statement. If you do not want to use this multiplier in the variance estimation, you can specify the VADJUST=NONE option in the MODEL statement to suppress this factor.

In addition, you can specify the VADJUST=MOREL option to request an adjustment to the variance estimator for the model parameters $\hat{\boldsymbol{\theta}}$, introduced by Morel (1989):

$$
\widehat{V}(\hat{\boldsymbol{\theta}}) = \widehat{\mathbf{Q}}^{-1}\widehat{\mathbf{G}}\widehat{\mathbf{Q}}^{-1} + \kappa\lambda\widehat{\mathbf{Q}}^{-1}
$$

where for given nonnegative constants δ and ϕ,

$$
\begin{aligned}
\kappa &= \max\left(\delta,\ p^{-1}\mathrm{tr}\left(\widehat{\mathbf{Q}}^{-1}\widehat{\mathbf{G}}\right)\right) \\
\lambda &= \min\left(\phi, \frac{p}{\tilde{n}-p}\right)
\end{aligned}
$$

The adjustment $\kappa\lambda\widehat{\mathbf{Q}}^{-1}$ does the following:

- reduces the small sample bias reflected in inflated Type I error rates
- guarantees a positive-definite estimated covariance matrix provided that $\widehat{\mathbf{Q}}^{-1}$ exists
- is close to zero when the sample size becomes large

In this adjustment, κ is an estimate of the design effect, which has been bounded below by the positive constant δ. You can use DEFFBOUND=δ in the VADJUST=MOREL option in the MODEL statement to specify this lower bound; by default, the procedure uses $\delta = 1$. The factor λ converges to zero when the sample size becomes large, and λ has an upper bound ϕ. You can use ADJBOUND=ϕ in the VADJUST=MOREL option in the MODEL statement to specify this upper bound; by default, the procedure uses $\phi = 0.5$.

Balanced Repeated Replication (BRR) Method

The balanced repeated replication (BRR) method requires that the full sample be drawn by using a stratified sample design with two primary sampling units (PSUs) per stratum. Let H be the total number of strata. The total number of replicates R is the smallest multiple of 4 that is greater than H. However, if you prefer a larger number of replicates, you can specify the REPS=*number* option. If a $number \times number$ Hadamard matrix cannot be constructed, the number of replicates is increased until a Hadamard matrix becomes available.

Each replicate is obtained by deleting one PSU per stratum according to the corresponding Hadamard matrix and adjusting the original weights for the remaining PSUs. The new weights are called replicate weights.

Replicates are constructed by using the first H columns of the $R \times R$ Hadamard matrix. The rth ($r = 1, 2, ..., R$) replicate is drawn from the full sample according to the rth row of the Hadamard matrix as follows:

- If the (r, h)th element of the Hadamard matrix is 1, then the first PSU of stratum h is included in the rth replicate and the second PSU of stratum h is excluded.
- If the (r, h)th element of the Hadamard matrix is -1, then the second PSU of stratum h is included in the rth replicate and the first PSU of stratum h is excluded.

The replicate weights of the remaining PSUs in each half sample are then doubled to their original weights. For more detail about the BRR method, see Wolter (1985) and Lohr (1999).

By default, an appropriate Hadamard matrix is generated automatically to create the replicates. You can request that the Hadamard matrix be displayed by specifying the VARMETHOD=BRR(PRINTH) method-option. If you provide a Hadamard matrix by specifying the VARMETHOD=BRR(HADAMARD=) method-option, then the replicates are generated according to the provided Hadamard matrix.

Let $\hat{\boldsymbol{\theta}}$ be the estimated regression coefficients from the full sample for $\boldsymbol{\theta}$, and let $\hat{\boldsymbol{\theta}}_r$ be the estimated regression coefficient from the rth replicate by using replicate weights. PROC SURVEYLOGISTIC estimates the covariance matrix of $\hat{\boldsymbol{\theta}}$ by

$$\widehat{\mathbf{V}}(\hat{\boldsymbol{\theta}}) = \frac{1}{R}\sum_{r=1}^{R}\left(\hat{\boldsymbol{\theta}}_r - \hat{\boldsymbol{\theta}}\right)\left(\hat{\boldsymbol{\theta}}_r - \hat{\boldsymbol{\theta}}\right)'$$

with H degrees of freedom, where H is the number of strata.

You can use the VARMETHOD=BRR(OUTWEIGHTS=) method-option to save the replicate weights into a SAS data set.

Fay's BRR Method

Fay's method is a modification of the BRR method, and it requires a stratified sample design with two primary sampling units (PSUs) per stratum. The total number of replicates R is the smallest multiple of 4 that is greater than the total number of strata H. However, if you prefer a larger number of replicates, you can specify the REPS= option.

For each replicate, Fay's method uses a Fay coefficient $0 \leq \epsilon < 1$ to impose a perturbation of the original weights in the full sample that is gentler than using only half samples, as in the traditional BRR method. The Fay coefficient $0 \leq \epsilon < 1$ can be optionally set by the FAY = ϵ method-option. By default, $\epsilon = 0.5$ if only the FAY method-option is used without specifying a value for ϵ (Judkins 1990; Rao and Shao 1999). When $\epsilon = 0$, Fay's method becomes the traditional BRR method. For more details, see Dippo, Fay, and Morganstein (1984), Fay (1984), Fay (1989), and Judkins (1990).

Let H be the number of strata. Replicates are constructed by using the first H columns of the $R \times R$ Hadamard matrix, where R is the number of replicates, $R > H$. The rth ($r = 1, 2, ..., R$) replicate is created from the full sample according to the rth row of the Hadamard matrix as follows:

- If the (r, h)th element of the Hadamard matrix is 1, then the full sample weight of the first PSU in stratum h is multiplied by ϵ and that of the second PSU is multiplied by $2-\epsilon$ to obtain the rth replicate weights.
- If the (r, h)th element of the Hadamard matrix is -1, then the full sample weight of the first PSU in stratum h is multiplied by $2-\epsilon$ and that of the second PSU is multiplied by ϵ to obtain the rth replicate weights.

You can use the VARMETHOD=BRR(OUTWEIGHTS=) method-option to save the replicate weights into a SAS data set.

By default, an appropriate Hadamard matrix is generated automatically to create the replicates. You can request that the Hadamard matrix be displayed by specifying the VARMETHOD=BRR(PRINTH) method-option. If you provide a Hadamard matrix by specifying the VARMETHOD=BRR(HADAMARD=) method-option, then the replicates are generated according to the provided Hadamard matrix.

Let $\hat{\boldsymbol{\theta}}$ be the estimated regression coefficients from the full sample for $\boldsymbol{\theta}$. Let $\hat{\boldsymbol{\theta}}_r$ be the estimated regression coefficient obtained from the rth replicate by using replicate weights. PROC SURVEYLOGISTIC estimates the covariance matrix of $\hat{\boldsymbol{\theta}}$ by

$$\widehat{\mathbf{V}}(\hat{\boldsymbol{\theta}}) = \frac{1}{R(1-\epsilon)^2} \sum_{r=1}^{R} \left(\hat{\boldsymbol{\theta}}_r - \hat{\boldsymbol{\theta}}\right)\left(\hat{\boldsymbol{\theta}}_r - \hat{\boldsymbol{\theta}}\right)'$$

with H degrees of freedom, where H is the number of strata.

Jackknife Method

The jackknife method of variance estimation deletes one PSU at a time from the full sample to create replicates. The total number of replicates R is the same as the total number of PSUs. In each replicate, the sample weights of the remaining PSUs are modified by the *jackknife coefficient* α_r. The modified weights are called replicate weights.

The jackknife coefficient and replicate weights are described as follows.

Without Stratification If there is no stratification in the sample design (no STRATA statement), the jackknife coefficients α_r are the same for all replicates:

$$\alpha_r = \frac{R-1}{R} \quad \text{where } r = 1, 2, ..., R$$

Denote the original weight in the full sample for the jth member of the ith PSU as w_{ij}. If the ith PSU is included in the rth replicate ($r = 1, 2, ..., R$), then the corresponding replicate weight for the jth member of the ith PSU is defined as

$$w_{ij}^{(r)} = w_{ij}/\alpha_r$$

With Stratification If the sample design involves stratification, each stratum must have at least two PSUs to use the jackknife method.

Let stratum $\tilde{h}_r$ be the stratum from which a PSU is deleted for the rth replicate. Stratum $\tilde{h}_r$ is called the *donor stratum.* Let $n_{\tilde{h}_r}$ be the total number of PSUs in the donor stratum $\tilde{h}_r$. The jackknife coefficients are defined as

$$\alpha_r = \frac{n_{\tilde{h}_r} - 1}{n_{\tilde{h}_r}}, \quad \text{where } r = 1, 2, ..., R$$

Denote the original weight in the full sample for the jth member of the ith PSU as w_{ij}. If the ith PSU is included in the rth replicate ($r = 1, 2, ..., R$), then the corresponding replicate weight for the jth member of the ith PSU is defined as

$$w_{ij}^{(r)} = \begin{cases} w_{ij} & \text{if } i\text{th PSU is not in the donor stratum } \tilde{h}_r \\ w_{ij}/\alpha_r & \text{if } i\text{th PSU is in the donor stratum } \tilde{h}_r \end{cases}$$

You can use the VARMETHOD=JACKKNIFE(OUTJKCOEFS=) method-option to save the jackknife coefficients into a SAS data set and use the VARMETHOD=JACKKNIFE(OUTWEIGHTS=) method-option to save the replicate weights into a SAS data set.

If you provide your own replicate weights with a REPWEIGHTS statement, then you can also provide corresponding jackknife coefficients with the JKCOEFS= option.

Let $\hat{\boldsymbol{\theta}}$ be the estimated regression coefficients from the full sample for $\boldsymbol{\theta}$. Let $\hat{\boldsymbol{\theta}}_r$ be the estimated regression coefficient obtained from the rth replicate by using replicate weights. PROC SURVEYLOGISTIC estimates the covariance matrix of $\hat{\boldsymbol{\theta}}$ by

$$\widehat{\mathbf{V}}(\hat{\boldsymbol{\theta}}) = \sum_{r=1}^{R} \alpha_r \left(\hat{\boldsymbol{\theta}}_r - \hat{\boldsymbol{\theta}}\right)\left(\hat{\boldsymbol{\theta}}_r - \hat{\boldsymbol{\theta}}\right)'$$

with $R - H$ degrees of freedom, where R is the number of replicates and H is the number of strata, or $R - 1$ when there is no stratification.

Hadamard Matrix

A Hadamard matrix $\mathbf{H}$ is a square matrix whose elements are either 1 or -1 such that

$$\mathbf{H}\mathbf{H}' = k\mathbf{I}$$

where k is the dimension of $\mathbf{H}$ and $\mathbf{I}$ is the identity matrix of order k. The order k is necessarily 1, 2, or a positive integer that is a multiple of 4.

For example, the following matrix is a Hadamard matrix of dimension $k = 8$:

$$\begin{array}{rrrrrrrr}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1
\end{array}$$

Domain Analysis

A DOMAIN statement requests the procedure to perform logistic regression analysis for each domain.

For a domain Ω, let I_Ω be the corresponding indicator variable:

$$I_\Omega(h,i,j) = \begin{cases} 1 & \text{if observation } (h,i,j) \text{ belongs to } \Omega \\ 0 & \text{otherwise} \end{cases}$$

Let

$$v_{hij} = w_{hij} I_\Omega(h,i,j) = \begin{cases} w_{hij} & \text{if observation } (h,i,j) \text{ belongs to } \Omega \\ 0 & \text{otherwise} \end{cases}$$

The regression in domain Ω uses v as the weight variable.

Hypothesis Testing and Estimation

Score Statistics and Tests

To understand the general form of the score statistics, let $\mathbf{g}(\boldsymbol{\theta})$ be the vector of first partial derivatives of the log likelihood with respect to the parameter vector $\boldsymbol{\theta}$, and let $\mathbf{H}(\boldsymbol{\theta})$ be the matrix of second partial derivatives of the log likelihood with respect to $\boldsymbol{\theta}$. That is, $\mathbf{g}(\boldsymbol{\theta})$ is the gradient vector, and $\mathbf{H}(\boldsymbol{\theta})$ is the Hessian matrix. Let $\mathbf{I}(\boldsymbol{\theta})$ be either $-\mathbf{H}(\boldsymbol{\theta})$ or the expected value of $-\mathbf{H}(\boldsymbol{\theta})$. Consider a null hypothesis H_0. Let $\hat{\boldsymbol{\theta}}$ be the MLE of $\boldsymbol{\theta}$ under H_0. The chi-square score statistic for testing H_0 is defined by

$$\mathbf{g}'(\hat{\boldsymbol{\theta}})\mathbf{I}^{-1}(\hat{\boldsymbol{\theta}})\mathbf{g}(\hat{\boldsymbol{\theta}})$$

It has an asymptotic χ^2 distribution with r degrees of freedom under H_0, where r is the number of restrictions imposed on $\boldsymbol{\theta}$ by H_0.

Testing the Parallel Lines Assumption

For an ordinal response, PROC SURVEYLOGISTIC performs a test of the parallel lines assumption. In the displayed output, this test is labeled "Score Test for the Equal Slopes Assumption" when the LINK= option is NORMIT or CLOGLOG. When LINK=LOGIT, the test is labeled as "Score Test for the Proportional Odds Assumption" in the output. This section describes the methods used to calculate the test.

For this test, the number of response levels, $D + 1$, is assumed to be strictly greater than 2. Let Y be the response variable taking values $1, \ldots, D, D + 1$. Suppose there are k explanatory variables. Consider the general cumulative model without making the parallel lines assumption:

$$g(\Pr(Y \leq d \mid \mathbf{x})) = (1, \mathbf{x})\boldsymbol{\theta}_d, \quad 1 \leq d \leq D$$

where $g(\cdot)$ is the link function, and $\boldsymbol{\theta}_d = (\alpha_d, \beta_{d1}, \ldots, \beta_{dk})'$ is a vector of unknown parameters consisting of an intercept α_d and k slope parameters $\beta_{k1}, \ldots, \beta_{kd}$. The parameter vector for this general cumulative model is

$$\boldsymbol{\theta} = (\boldsymbol{\theta}_1', \ldots, \boldsymbol{\theta}_D')'$$

Under the null hypothesis of parallelism H_0: $\beta_{1i} = \beta_{2i} = \cdots = \beta_{Di}, 1 \leq i \leq k$, there is a single common slope parameter for each of the s explanatory variables. Let $\beta_1, \ldots, \beta_k$ be the common slope parameters. Let $\hat{\alpha}_1, \ldots, \hat{\alpha}_D$ and $\hat{\beta}_1, \ldots, \hat{\beta}_D$ be the MLEs of the intercept parameters and the common slope parameters. Then, under H_0, the MLE of $\boldsymbol{\theta}$ is

$$\hat{\boldsymbol{\theta}} = (\hat{\boldsymbol{\theta}}_1', \ldots, \hat{\boldsymbol{\theta}}_D')' \quad \text{with} \quad \hat{\boldsymbol{\theta}}_d = (\hat{\alpha}_d, \hat{\beta}_1, \ldots, \hat{\beta}_k)', \quad 1 \le d \le D$$

and the chi-squared score statistic $\mathbf{g}'(\hat{\boldsymbol{\theta}})\mathbf{I}^{-1}(\hat{\boldsymbol{\theta}})\mathbf{g}(\hat{\boldsymbol{\theta}})$ has an asymptotic chi-square distribution with $k(D-1)$ degrees of freedom. This tests the parallel lines assumption by testing the equality of separate slope parameters simultaneously for all explanatory variables.

Wald Confidence Intervals for Parameters

Wald confidence intervals are sometimes called the normal confidence intervals. They are based on the asymptotic normality of the parameter estimators. The $100(1-\alpha)\%$ Wald confidence interval for θ_j is given by

$$\hat{\theta}_j \pm z_{1-\alpha/2}\hat{\sigma}_j$$

where $z_{1-\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the standard normal distribution, $\hat{\theta}_j$ is the pseudo-estimate of θ_j, and $\hat{\sigma}_j$ is the standard error estimate of $\hat{\theta}_j$ in the section "Variance Estimation" on page 6427.

Testing Linear Hypotheses about the Regression Coefficients

Linear hypotheses for $\boldsymbol{\theta}$ are expressed in matrix form as

$$H_0: \mathbf{L}\boldsymbol{\theta} = \mathbf{c}$$

where $\mathbf{L}$ is a matrix of coefficients for the linear hypotheses and $\mathbf{c}$ is a vector of constants. The vector of regression coefficients $\boldsymbol{\theta}$ includes slope parameters as well as intercept parameters. The Wald chi-square statistic for testing H_0 is computed as

$$\chi^2_W = (\mathbf{L}\hat{\boldsymbol{\theta}} - \mathbf{c})'[\mathbf{L}\hat{\mathbf{V}}(\hat{\boldsymbol{\theta}})\mathbf{L}']^{-1}(\mathbf{L}\hat{\boldsymbol{\theta}} - \mathbf{c})$$

where $\widehat{\mathbf{V}}(\hat{\boldsymbol{\theta}})$ is the estimated covariance matrix in the section "Variance Estimation" on page 6427. Under H_0, χ^2_W has an asymptotic chi-square distribution with r degrees of freedom, where r is the rank of $\mathbf{L}$.

Odds Ratio Estimation

Consider a dichotomous response variable with outcomes *event* and *nonevent*. Let a dichotomous risk factor variable X take the value 1 if the risk factor is present and 0 if the risk factor is absent. According to the logistic model, the log odds function, $g(X)$, is given by

$$g(X) \equiv \log\left(\frac{\Pr(\,event \mid X)}{\Pr(\,nonevent \mid X)}\right) = \beta_0 + \beta_1 X$$

The odds ratio ψ is defined as the ratio of the odds for those with the risk factor ($X = 1$) to the odds for those without the risk factor ($X = 0$). The log of the odds ratio is given by

$$\log(\psi) \equiv \log(\psi(X = 1, X = 0)) = g(X = 1) - g(X = 0) = \beta_1$$

The parameter, β_1, associated with X represents the change in the log odds from $X = 0$ to $X = 1$. So the odds ratio is obtained by simply exponentiating the value of the parameter associated with the risk factor. The odds ratio indicates how the odds of *event* change as you change X from 0 to 1. For instance, $\psi = 2$ means that the odds of an event when $X = 1$ are twice the odds of an event when $X = 0$.

Suppose the values of the dichotomous risk factor are coded as constants a and b instead of 0 and 1. The odds when $X = a$ become $\exp(\beta_0 + a\beta_1)$, and the odds when $X = b$ become $\exp(\beta_0 + b\beta_1)$. The odds ratio corresponding to an increase in X from a to b is

$$\psi = \exp[(b - a)\beta_1] = [\exp(\beta_1)]^{b-a} \equiv [\exp(\beta_1)]^c$$

Note that for any a and b such that $c = b - a = 1, \psi = \exp(\beta_1)$. So the odds ratio can be interpreted as the change in the odds for any increase of one unit in the corresponding risk factor. However, the change in odds for some amount other than one unit is often of greater interest. For example, a change of one pound in body weight might be too small to be considered important, while a change of 10 pounds might be more meaningful. The odds ratio for a change in X from a to b is estimated by raising the odds ratio estimate for a unit change in X to the power of $c = b - a$, as shown previously.

For a polytomous risk factor, the computation of odds ratios depends on how the risk factor is parameterized. For illustration, suppose that Race is a risk factor with four categories: White, Black, Hispanic, and Other.

For the effect parameterization scheme (PARAM=EFFECT) with White as the reference group, the design variables for Race are as follows.

	Design Variables		
Race	X_1	X_2	X_3
Black	1	0	0
Hispanic	0	1	0
Other	0	0	1
White	−1	−1	−1

The log odds for Black is

$$\begin{aligned} g(\text{Black}) &= \beta_0 + \beta_1(X_1 = 1) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0) \\ &= \beta_0 + \beta_1 \end{aligned}$$

The log odds for White is

$$\begin{aligned} g(\text{White}) &= \beta_0 + \beta_1(X_1 = -1) + \beta_2(X_2 = -1) + \beta_3(X_3 = -1)) \\ &= \beta_0 - \beta_1 - \beta_2 - \beta_3 \end{aligned}$$

Therefore, the log odds ratio of Black versus White becomes

$$\begin{aligned}\log(\psi(\text{Black, White})) &= g(\text{Black}) - g(\text{White}) \\ &= 2\beta_1 + \beta_2 + \beta_3\end{aligned}$$

For the reference cell parameterization scheme (PARAM=REF) with White as the reference cell, the design variables for race are as follows.

	Design Variables		
Race	X_1	X_2	X_3
Black	1	0	0
Hispanic	0	1	0
Other	0	0	1
White	0	0	0

The log odds ratio of Black versus White is given by

$$\begin{aligned}\log(\psi(\text{Black, White})) &= g(\text{Black}) - g(\text{White}) \\ &= (\beta_0 + \beta_1(X_1 = 1) + \beta_2(X_2 = 0)) + \beta_3(X_3 = 0)) - \\ &\quad (\beta_0 + \beta_1(X_1 = 0) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0)) \\ &= \beta_1\end{aligned}$$

For the GLM parameterization scheme (PARAM=GLM), the design variables are as follows.

	Design Variables			
Race	X_1	X_2	X_3	X_4
Black	1	0	0	0
Hispanic	0	1	0	0
Other	0	0	1	0
White	0	0	0	1

The log odds ratio of Black versus White is

$$\begin{aligned}\log(\psi(\text{Black, White})) &= g(\text{Black}) - g(\text{White}) \\ &= (\beta_0 + \beta_1(X_1 = 1) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0) + \beta_4(X_4 = 0)) - \\ &\quad (\beta_0 + \beta_1(X_1 = 0) + \beta_2(X_2 = 0) + \beta_3(X_3 = 0) + \beta_4(X_4 = 1)) \\ &= \beta_1 - \beta_4\end{aligned}$$

Consider the hypothetical example of heart disease among race in Hosmer and Lemeshow (2000, p. 51). The entries in the following contingency table represent counts.

	Race			
Disease Status	**White**	**Black**	**Hispanic**	**Other**
Present	5	20	15	10
Absent	20	10	10	10

The computation of odds ratio of Black versus White for various parameterization schemes is shown in Table 84.3.

Table 84.3 Odds Ratio of Heart Disease Comparing Black to White

	Parameter Estimates				
PARAM=	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	**Odds Ratio Estimates**
EFFECT	0.7651	0.4774	0.0719		$\exp(2 \times 0.7651 + 0.4774 + 0.0719) = 8$
REF	2.0794	1.7917	1.3863		$\exp(2.0794) = 8$
GLM	2.0794	1.7917	1.3863	0.0000	$\exp(2.0794) = 8$

Since the log odds ratio ($\log(\psi)$) is a linear function of the parameters, the Wald confidence interval for $\log(\psi)$ can be derived from the parameter estimates and the estimated covariance matrix. Confidence intervals for the odds ratios are obtained by exponentiating the corresponding confidence intervals for the log odd ratios. In the displayed output of PROC SURVEYLOGISTIC, the "Odds Ratio Estimates" table contains the odds ratio estimates and the corresponding 95% Wald confidence intervals computed by using the covariance matrix in the section "Variance Estimation" on page 6427. For continuous explanatory variables, these odds ratios correspond to a unit increase in the risk factors.

To customize odds ratios for specific units of change for a continuous risk factor, you can use the UNITS statement to specify a list of relevant units for each explanatory variable in the model. Estimates of these customized odds ratios are given in a separate table. Let (L_j, U_j) be a confidence interval for $\log(\psi)$. The corresponding lower and upper confidence limits for the customized odds ratio $\exp(c\beta_j)$ are $\exp(cL_j)$ and $\exp(cU_j)$, respectively, (for $c > 0$); or $\exp(cU_j)$ and $\exp(cL_j)$, respectively, (for $c < 0$). You use the CLODDS= option to request the confidence intervals for the odds ratios.

For a generalized logit model, odds ratios are computed similarly, except D odds ratios are computed for each effect, corresponding to the D logits in the model.

Rank Correlation of Observed Responses and Predicted Probabilities

The predicted mean score of an observation is the sum of the Ordered Values (shown in the "Response Profile" table) minus one, weighted by the corresponding predicted probabilities for that observation; that is, the predicted means score is $\sum_{d=1}^{D+1}(d-1)\hat{\pi}_d$, where $D+1$ is the number of response levels and $\hat{\pi}_d$ is the predicted probability of the dth (ordered) response.

A pair of observations with different observed responses is said to be *concordant* if the observation with the lower ordered response value has a lower predicted mean score than the observation with the higher ordered response value. If the observation with the lower ordered response value has a higher predicted mean score than the observation with the higher ordered response value, then the pair is *discordant*. If the pair is neither concordant nor discordant, it is a *tie*. Enumeration of the total numbers of concordant and discordant pairs is carried out by categorizing the predicted mean score into intervals of length $D/500$ and accumulating the corresponding frequencies of observations.

Let N be the sum of observation frequencies in the data. Suppose there are a total of t pairs with different responses, n_c of them are concordant, n_d of them are discordant, and $t - n_c - n_d$ of them

are tied. PROC SURVEYLOGISTIC computes the following four indices of rank correlation for assessing the predictive ability of a model:

$$
\begin{aligned}
c &= (n_c + 0.5(t - n_c - n_d))/t \\
\text{Somers' } D &= (n_c - n_d)/t \\
\text{Goodman-Kruskal Gamma} &= (n_c - n_d)/(n_c + n_d) \\
\text{Kendall's Tau-}a &= (n_c - n_d)/(0.5N(N-1))
\end{aligned}
$$

Note that c also gives an estimate of the area under the receiver operating characteristic (ROC) curve when the response is binary (Hanley and McNeil 1982).

For binary responses, the predicted mean score is equal to the predicted probability for Ordered Value 2. As such, the preceding definition of concordance is consistent with the definition used in previous releases for the binary response model.

Linear Predictor, Predicted Probability, and Confidence Limits

This section describes how predicted probabilities and confidence limits are calculated by using the pseudo-estimates (MLEs) obtained from PROC SURVEYLOGISTIC. For a specific example, see the section "Getting Started: SURVEYLOGISTIC Procedure" on page 6377. Predicted probabilities and confidence limits can be output to a data set with the OUTPUT statement.

Cumulative Response Models

For a row vector of explanatory variables $\mathbf{x}$, the linear predictor

$$\eta_i = g(\Pr(Y \le i \mid \mathbf{x})) = \alpha_i + \mathbf{x}\boldsymbol{\beta}, \quad 1 \le i \le k$$

is estimated by

$$\hat{\eta}_i = \hat{\alpha}_i + \mathbf{x}\hat{\boldsymbol{\beta}}$$

where $\hat{\alpha}_i$ and $\hat{\boldsymbol{\beta}}$ are the MLEs of α_i and $\boldsymbol{\beta}$. The estimated standard error of η_i is $\hat{\sigma}(\hat{\eta}_i)$, which can be computed as the square root of the quadratic form $(1, \mathbf{x}')\hat{\mathbf{V}}_{\mathbf{b}}(1, \mathbf{x}')'$, where $\hat{\mathbf{V}}_{\mathbf{b}}$ is the estimated covariance matrix of the parameter estimates. The asymptotic $100(1-\alpha)\%$ confidence interval for η_i is given by

$$\hat{\eta}_i \pm z_{\alpha/2}\hat{\sigma}(\hat{\eta}_i)$$

where $z_{\alpha/2}$ is the $100(1-\alpha/2)$ percentile point of a standard normal distribution.

The predicted value and the $100(1-\alpha)\%$ confidence limits for $\Pr(Y \le i \mid \mathbf{x})$ are obtained by back-transforming the corresponding measures for the linear predictor.

Link	Predicted Probability	$100(1-\alpha)$ Confidence Limits
LOGIT	$1/(1+e^{-\hat{\eta}_i})$	$1/(1+e^{-\hat{\eta}_i \pm z_{\alpha/2}\hat{\sigma}(\hat{\eta}_i)})$
PROBIT	$\Phi(\hat{\eta}_i)$	$\Phi(\hat{\eta}_i \pm z_{\alpha/2}\hat{\sigma}(\hat{\eta}_i))$
CLOGLOG	$1-e^{-e^{\hat{\eta}_i}}$	$1-e^{-e^{\hat{\eta}_i \pm z_{\alpha/2}\hat{\sigma}(\hat{\eta}_i)}}$

Generalized Logit Model

For a vector of explanatory variables $\boldsymbol{x}$, let π_i denote the probability of obtaining the response value i:

$$\pi_i = \begin{cases} \pi_{k+1} e^{\alpha_i + \mathbf{x}\boldsymbol{\beta}_i} & 1 \le i \le k \\ \dfrac{1}{1+\sum_{j=1}^{k} e^{\alpha_j + \mathbf{x}\boldsymbol{\beta}_j}} & i = k+1 \end{cases}$$

By the *delta method*,

$$\sigma^2(\pi_i) = \left(\frac{\partial \pi_i}{\partial \boldsymbol{\theta}}\right)' \mathbf{V}(\boldsymbol{\theta}) \frac{\partial \pi_i}{\partial \boldsymbol{\theta}}$$

A $100(1-\alpha)\%$ confidence level for π_i is given by

$$\hat{\pi}_i \pm z_{\alpha/2}\hat{\sigma}(\hat{\pi}_i)$$

where $\hat{\pi}_i$ is the estimated expected probability of response i and $\hat{\sigma}(\hat{\pi}_i)$ is obtained by evaluating $\sigma(\pi_i)$ at $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$.

Output Data Sets

You can use the Output Delivery System (ODS) to create a SAS data set from any piece of PROC SURVEYLOGISTIC output. See the section "ODS Table Names" on page 6447 for more information. For a more detailed description of using ODS, see Chapter 20, "Using the Output Delivery System."

PROC SURVEYLOGISTIC also provides an OUTPUT statement to create a data set that contains estimated linear predictors, the estimates of the cumulative or individual response probabilities, and their confidence limits.

If you use BRR or jackknife variance estimation, PROC SURVEYLOGISTIC provides an output data set that stores the replicate weights and an output data set that stores the jackknife coefficients for jackknife variance estimation.

OUT= Data Set in the OUTPUT Statement

The OUT= data set in the OUTPUT statement contains all the variables in the input data set along with statistics you request by using *keyword=name* options or the PREDPROBS= option in the OUTPUT statement. In addition, if you use the single-trial syntax and you request any of the XBETA=, STDXBETA=, PREDICTED=, LCL=, and UCL= options, the OUT= data set contains the automatic variable _LEVEL_. The value of _LEVEL_ identifies the response category upon which the computed values of XBETA=, STDXBETA=, PREDICTED=, LCL=, and UCL= are based.

When there are more than two response levels, only variables named by the XBETA=, STDXBETA=, PREDICTED=, LOWER=, and UPPER= options and the variables given by PREDPROBS=(INDIVIDUAL CUMULATIVE) have their values computed; the other variables have missing values. If you fit a generalized logit model, the cumulative predicted probabilities are not computed.

When there are only two response categories, each input observation produces one observation in the OUT= data set.

If there are more than two response categories and you specify only the PREDPROBS= option, then each input observation produces one observation in the OUT= data set. However, if you fit an ordinal (cumulative) model and specify options other than the PREDPROBS= options, each input observation generates as many output observations as one fewer than the number of response levels, and the predicted probabilities and their confidence limits correspond to the cumulative predicted probabilities. If you fit a generalized logit model and specify options other than the PREDPROBS= options, each input observation generates as many output observations as the number of response categories; the predicted probabilities and their confidence limits correspond to the probabilities of individual response categories.

For observations in which only the response variable is missing, values of the XBETA=, STDXBETA=, PREDICTED=, UPPER=, LOWER=, and PREDPROBS= options are computed even though these observations do not affect the model fit. This enables, for instance, predicted probabilities to be computed for new observations.

Replicate Weights Output Data Set

If you specify the OUTWEIGHTS= method-option for BRR or jackknife method in the VARMETHOD= option, PROC SURVEYLOGISTIC stores the replicate weights in an output data set. The OUTWEIGHTS= output data set contains all observations used in the analysis or all valid observations in the DATA= input data set. A valid observation is an observation that has a positive value of the WEIGHT variable. Valid observations must also have nonmissing values of the STRATA and CLUSTER variables, unless you specify the MISSING option.

The OUTWEIGHTS= data set contains the following variables:

- all variables in the DATA= input data set
- RepWt_1, RepWt_2, . . ., RepWt_n, which are the replicate weight variables

where n is the total number of replicates in the analysis. Each replicate weight variable contains the replicate weights for the corresponding replicate. Replicate weights equal zero for those observations not included in the replicate.

After the procedure creates replicate weights for a particular input data set and survey design, you can use the OUTWEIGHTS= method-option to store these replicate weights and then use them again in subsequent analyses, either in PROC SURVEYLOGISTIC or in the other survey procedures. You can use the REPWEIGHTS statement to provide replicate weights for the procedure.

Jackknife Coefficients Output Data Set

If you specify the OUTJKCOEFS= method-option for VARMETHOD=JACKKNIFE, PROC SURVEYLOGISTIC stores the jackknife coefficients in an output data set. The OUTJKCOEFS= output data set contains one observation for each replicate. The OUTJKCOEFS= data set contains the following variables:

- Replicate, which is the replicate number for the jackknife coefficient
- JKCoefficient, which is the jackknife coefficient
- DonorStratum, which is the stratum of the PSU that was deleted to construct the replicate, if you specify a STRATA statement

After the procedure creates jackknife coefficients for a particular input data set and survey design, you can use the OUTJKCOEFS= method-option to store these coefficients and then use them again in subsequent analyses, either in PROC SURVEYLOGISTIC or in the other survey procedures. You can use the JKCOEFS= option in the REPWEIGHTS statement to provide jackknife coefficients for the procedure.

Displayed Output

The SURVEYLOGISTIC procedure produces the following output.

Model Information

By default, PROC SURVEYLOGISTIC displays the following information in the "Model Information" table:

- name of the input Data Set
- name and label of the Response Variable if the single-trial syntax is used
- number of Response Levels
- name of the Events Variable if the events/trials syntax is used

- name of the Trials Variable if the events/trials syntax is used
- name of the Offset Variable if the OFFSET= option is specified
- name of the Frequency Variable if the FREQ statement is specified
- name(s) of the Stratum Variable(s) if the STRATA statement is specified
- total Number of Strata if the STRATA statement is specified
- name(s) of the Cluster Variable(s) if the CLUSTER statement is specified
- total Number of Clusters if the CLUSTER statement is specified
- name of the Weight Variable if the WEIGHT statement is specified
- Variance Adjustment method
- Upper Bound ADJBOUND parameter used in the VADJUST=MOREL(ADJBOUND=) option
- Lower Bound DEFFBOUND parameter used in the VADJUST=MOREL(DEFFBOUND=) option
- whether FPC (finite population correction) is used

Variance Estimation

By default, PROC SURVEYLOGISTIC displays the following variance estimation specification in the "Variance Estimation" table:

- variance estimation Method
- Variance Adjustment method
- Upper Bound ADJBOUND parameter used in the VADJUST=MOREL(ADJBOUND=) option
- Lower Bound DEFFBOUND parameter used in the VADJUST=MOREL(DEFFBOUND=) option
- whether FPC (finite population correction) is used
- Number of Replicates if you specify VARMETHOD=BRR or VARMETHOD=JACKKNIFE
- name of the Hadamard Data Set if you specify the VARMETHOD=BRR(HADAMARD=) method-option
- FAY coefficient used if you specify the VARMETHOD=BRR(FAY) method-option
- name of the Replicate Weights Dataset if the REPWEIGHTS statement is used
- whether Missing Levels are created for categorical variables by the MISSING option
- whether observations with Missing Values are included in the analysis due to the NOMCAR option

Data Summary

By default, PROC SURVEYLOGISTIC displays the following information for the entire data set:

- Number of Observations read from the input data set
- Number of Observations used in the analysis

If there is a DOMAIN statement, PROC SURVEYLOGISTIC also displays the following:

- Number of Observations in the current domain
- Number of Observations not in the current domain

If there is a FREQ statement, PROC SURVEYLOGISTIC also displays the following:

- Sum of Frequencies of all the observations read from the input data set
- Sum of Frequencies of all the observations used in the analysis

If there is a WEIGHT statement, PROC SURVEYLOGISTIC also displays the following:

- Sum of Weights of all the observations read from the input data set
- Sum of Weights of all the observations used in the analysis
- Sum of Weights of all the observations in the current domain, if DOMAIN statement is also specified.

Response Profile

By default, PROC SURVEYLOGISTIC displays a “Response Profile” table, which gives, for each response level, the ordered value (an integer between one and the number of response levels, inclusive); the value of the response variable if the single-trial syntax is used or the values “EVENT” and “NO EVENT” if the events/trials syntax is used; the count or frequency; and the sum of weights if the WEIGHT statement is specified.

Class Level Information

If you use a CLASS statement to name classification variables, PROC SURVEYLOGISTIC displays a "Class Level Information" table. This table contains the following information for each classification variable:

- Class, which lists each CLASS variable name

- Value, which lists the values of the classification variable. The values are separated by a white space character; therefore, to avoid confusion, you should not include a white space character within a classification variable value.
- Design Variables, which lists the parameterization used for the classification variables

Stratum Information

When you specify the LIST option in the STRATA statement, PROC SURVEYLOGISTIC displays a "Stratum Information" table, which provides the following information for each stratum:

- Stratum Index, which is a sequential stratum identification number
- STRATA variable(s), which lists the levels of STRATA variables for the stratum
- Population Total, if you specify the TOTAL= option
- Sampling Rate, if you specify the TOTAL= or RATE= option. If you specify the TOTAL= option, the sampling rate is based on the number of nonmissing observations in the stratum.
- N Obs, which is the number of observations
- number of Clusters, if you specify a CLUSTER statement

Maximum Likelihood Iteration History

The "Maximum Likelihood Iterative Phase" table gives the iteration number, the step size (in the scale of 1.0, 0.5, 0.25, and so on) or the ridge value, -2 log likelihood, and parameter estimates for each iteration. Also displayed are the last evaluation of the gradient vector and the last change in the -2 log likelihood. You need to use the ITPRINT option in the MODEL statement to obtain this table.

Score Test

The "Score Test" table displays the score test result for testing the parallel lines assumption, if an ordinal response model is fitted. If LINK=CLOGLOG or LINK=PROBIT, this test is labeled "Score Test for the Parallel Slopes Assumption." The proportion odds assumption is a special case of the parallel lines assumption when LINK=LOGIT. In this case, the test is labeled "Score Test for the Proportional Odds Assumption."

Model Fit Statistics

By default, PROC SURVEYLOGISTIC displays the following information in the "Model Fit Statistics" table:

- "Model Fit Statistics" and "Testing Global Null Hypothesis: BETA=0" tables, which give the various criteria (−2 Log L, AIC, SC) based on the likelihood for fitting a model with intercepts only and for fitting a model with intercepts and explanatory variables. If you specify the NOINT option, these statistics are calculated without considering the intercept parameters. The third column of the table gives the chi-square statistics and p-values for the −2 Log L statistic and for the Score statistic. These test the joint effect of the explanatory variables included in the model. The Score criterion is always missing for the models identified by the first two columns of the table. Note also that the first two rows of the Chi-Square column are always missing, since tests cannot be performed for AIC and SC.
- generalized R^2 measures for the fitted model if you specify the RSQUARE option in the MODEL statement

Type 3 Analysis of Effects

PROC SURVEYLOGISTIC displays the "Type III Analysis of Effects" table if the model contains an effect involving a CLASS variable. This table gives the degrees of freedom, the Wald Chi-square statistic, and the p-value for each effect in the model.

Analysis of Maximum Likelihood Estimates

By default, PROC SURVEYLOGISTIC displays the following information in the "Analysis of Maximum Likelihood Estimates" table:

- the degrees of freedom for Wald chi-square test
- maximum likelihood estimate of the parameter
- estimated standard error of the parameter estimate, computed as the square root of the corresponding diagonal element of the estimated covariance matrix
- Wald chi-square statistic, computed by squaring the ratio of the parameter estimate divided by its standard error estimate
- p-value of the Wald chi-square statistic with respect to a chi-square distribution with one degree of freedom
- standardized estimate for the slope parameter, given by $\hat{\beta}_i / (s/s_i)$, where s_i is the total sample standard deviation for the ith explanatory variable and

$$s = \begin{cases} \pi/\sqrt{3} & \text{logistic} \\ 1 & \text{normal} \\ \pi/\sqrt{6} & \text{extreme-value} \end{cases}$$

You need to specify the STB option in the MODEL statement to obtain these estimates. Standardized estimates of the intercept parameters are set to missing.

- value of ($e^{\hat{\beta}_i}$) for each slope parameter β_i if you specify the EXPB option in the MODEL statement. For continuous variables, this is equivalent to the estimated odds ratio for a one-unit change.
- label of the variable (if space permits) if you specify the PARMLABEL option in the MODEL statement. Due to constraints on the line size, the variable label might be suppressed in order to display the table in one panel. Use the SAS system option LINESIZE= to specify a larger line size to accommodate variable labels. A shorter line size can break the table into two panels, allowing labels to be displayed.

Odds Ratio Estimates

The "Odds Ratio Estimates" table displays the odds ratio estimates and the corresponding 95% Wald confidence intervals. For continuous explanatory variables, these odds ratios correspond to a unit increase in the risk factors.

Association of Predicted Probabilities and Observed Responses

The "Association of Predicted Probabilities and Observed Responses" table displays measures of association between predicted probabilities and observed responses, which include a breakdown of the number of pairs with different responses, and four rank correlation indexes: Somers' *D*, Goodman-Kruskal Gamma, and Kendall's Tau-*a*, and *c*.

Wald Confidence Interval for Parameters

The "Wald Confidence Interval for Parameters" table displays confidence intervals for all the parameters if you use the CLPARM option in the MODEL statement.

Wald Confidence Interval for Odds Ratios

The "Wald Confidence Interval for Odds Ratios" table displays confidence intervals for all the odds ratios if you use the CLODDS option in the MODEL statement.

Estimated Covariance Matrix

PROC SURVEYLOGISTIC displays the following information in the "Estimated Covariance Matrix" table:

- estimated covariance matrix of the parameter estimates if you use the COVB option in the MODEL statement
- estimated correlation matrix of the parameter estimates if you use the CORRB option in the MODEL statement

Linear Hypotheses Testing Results

The "Linear Hypothesis Testing" table gives the result of the Wald test for each TEST statement (if specified).

Hadamard Matrix

If you specify the VARMETHOD=BRR(PRINTH) method-option in the PROC statement, PROC SURVEYLOGISTIC displays the Hadamard matrix.

When you provide a Hadamard matrix with the VARMETHOD=BRR(HADAMARD=) method-option, the procedure displays only used rows and columns of the Hadamard matrix.

ODS Table Names

PROC SURVEYLOGISTIC assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 84.4. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 84.4 ODS Tables Produced by PROC SURVEYLOGISTIC

ODS Table Name	Description	Statement	Option
Association	Association of predicted probabilities and observed responses	MODEL	default
ClassLevelInfo	Class variable levels and design variables	MODEL	default (with CLASS vars)
CLOddsWald	Wald's confidence limits for odds ratios	MODEL	CLODDS
CLparmWald	Wald's confidence limits for parameters	MODEL	CLPARM
ContrastCoeff	L matrix from CONTRAST	CONTRAST	E
ContrastEstimate	Estimates from CONTRAST	CONTRAST	ESTIMATE=
ContrastTest	Wald test for CONTRAST	CONTRAST	default
ConvergenceStatus	Convergence status	MODEL	default
CorrB	Estimated correlation matrix of parameter estimators	MODEL	CORRB
CovB	Estimated covariance matrix of parameter estimators	MODEL	COVB
CumulativeModelTest	Test of the cumulative model assumption	MODEL	(ordinal response)
DomainSummary	Domain summary	DOMAIN	default
FitStatistics	Model fit statistics	MODEL	default

Table 84.4 *continued*

ODS Table Name	Description	Statement	Option
GlobalTests	Test for global null hypothesis	MODEL	default
HadamardMatrix	Hadamard matrix	PROC	PRINTH
IterHistory	Iteration history	MODEL	ITPRINT
LastGradient	Last evaluation of gradient	MODEL	ITPRINT
Linear	Linear combination	PROC	default
LogLikeChange	Final change in the log likelihood	MODEL	ITPRINT
ModelInfo	Model information	PROC	default
NObs	Number of observations	PROC	default
OddsEst	Adjusted odds ratios	UNITS	default
OddsRatios	Odds ratios	MODEL	default
ParameterEstimates	Maximum likelihood estimates of model parameters	MODEL	default
RSquare	R-square	MODEL	RSQUARE
ResponseProfile	Response profile	PROC	default
StrataInfo	Stratum information	STRATA	LIST
TestPrint1	$\mathbf{L}[\mathrm{cov}(\mathbf{b})]\mathbf{L}'$ and $\mathbf{Lb}-\mathbf{c}$	TEST	PRINT
TestPrint2	$\mathrm{Ginv}(\mathbf{L}[\mathrm{cov}(\mathbf{b})]\mathbf{L}')$ and $\mathrm{Ginv}(\mathbf{L}[\mathrm{cov}(\mathbf{b})]\mathbf{L}')(\mathbf{Lb}-\mathbf{c})$	TEST	PRINT
TestStmts	Linear hypotheses testing results	TEST	default
Type3	Type 3 tests of effects	MODEL	default (with CLASS variables)
VarianceEstimation	Variance estimation	PROC	default

By referring to the names of such tables, you can use the ODS OUTPUT statement to place one or more of these tables in output data sets.

Examples: SURVEYLOGISTIC Procedure

Example 84.1: Stratified Cluster Sampling

A market research firm conducts a survey among undergraduate students at a certain university to evaluate three new Web designs for a commercial Web site targeting undergraduate students at the university.

The sample design is a stratified sample where the strata are students' classes. Within each class,

300 students are randomly selected by using simple random sampling without replacement. The total number of students in each class in the fall semester of 2001 is shown in the following table:

Class	Enrollment
1 - Freshman	3,734
2 - Sophomore	3,565
3 - Junior	3,903
4 - Senior	4,196

This total enrollment information is saved in the SAS data set Enrollment by using the following SAS statements:

```
proc format ;
   value Class 1='Freshman' 2='Sophomore'
               3='Junior'   4='Senior';
run;
data Enrollment;
   format Class Class.;
   input Class _TOTAL_;
   datalines;
1 3734
2 3565
3 3903
4 4196
;
```

In the data set Enrollment, the variable _TOTAL_ contains the enrollment figures for all classes. They are also the population size for each stratum in this example.

Each student selected in the sample evaluates one randomly selected Web design by using the following scale:

1	dislike very much
2	dislike
3	neutral
4	like
5	like very much

The survey results are collected and shown in the following table, with the three different Web designs coded as A, B, and C.

Evaluation of New Web Designs						
		Rating Counts				
Strata	Design	1	2	3	4	5
Freshman	A	10	34	35	16	15
	B	5	6	24	30	25
	C	11	14	20	34	21
Sophomore	A	19	12	26	18	25
	B	10	18	32	23	26
	C	15	22	34	9	20
Junior	A	8	21	23	26	22
	B	1	4	15	33	47
	C	16	19	30	23	12
Senior	A	11	14	24	33	18
	B	8	15	25	30	22
	C	2	34	30	18	16

The survey results are stored in a SAS data set WebSurvey by using the following SAS statements:

```
proc format ;
   value Design 1='A' 2='B' 3='C';
   value Rating 1='dislike very much'
                2='dislike'
                3='neutral'
                4='like'
                5='like very much';
run;
data WebSurvey;
   format Class Class. Design Design. Rating Rating. ;
   do Class=1 to 4;
      do Design=1 to 3;
         do Rating=1 to 5;
           input Count @@;
           output;
         end;
      end;
   end;
   datalines;
10 34 35 16 15   8 21 23 26 22   5 10 24 30 21
 1 14 25 23 37  11 14 20 34 21  16 19 30 23 12
19 12 26 18 25  11 14 24 33 18  10 18 32 23 17
 8 15 35 30 12  15 22 34  9 20   2 34 30 18 16
;
data WebSurvey; set WebSurvey;
   if Class=1 then Weight=3734/300;
   if Class=2 then Weight=3565/300;
   if Class=3 then Weight=3903/300;
   if Class=4 then Weight=4196/300;
run;
```

The data set WebSurvey contains the variables Class, Design, Rating, Count, and Weight. The variable class is the stratum variable, with four strata: freshman, sophomore, junior, and senior. The variable Design specifies the three new Web designs: A, B, and C. The variable Rating contains students' evaluations of the new Web designs. The variable counts gives the frequency with which each Web design received each rating within each stratum. The variable weight contains the sampling weights, which are the reciprocals of selection probabilities in this example.

Output 84.1.1 shows the first 20 observations of the data set.

Output 84.1.1 Web Design Survey Sample (First 20 Observations)

Obs	Class	Design	Rating	Count	Weight
1	Freshman	A	dislike very much	10	12.4467
2	Freshman	A	dislike	34	12.4467
3	Freshman	A	neutral	35	12.4467
4	Freshman	A	like	16	12.4467
5	Freshman	A	like very much	15	12.4467
6	Freshman	B	dislike very much	8	12.4467
7	Freshman	B	dislike	21	12.4467
8	Freshman	B	neutral	23	12.4467
9	Freshman	B	like	26	12.4467
10	Freshman	B	like very much	22	12.4467
11	Freshman	C	dislike very much	5	12.4467
12	Freshman	C	dislike	10	12.4467
13	Freshman	C	neutral	24	12.4467
14	Freshman	C	like	30	12.4467
15	Freshman	C	like very much	21	12.4467
16	Sophomore	A	dislike very much	1	11.8833
17	Sophomore	A	dislike	14	11.8833
18	Sophomore	A	neutral	25	11.8833
19	Sophomore	A	like	23	11.8833
20	Sophomore	A	like very much	37	11.8833

The following SAS statements perform the logistic regression:

```
proc surveylogistic data=WebSurvey total=Enrollment;
   stratum Class;
   freq Count;
   class Design;
   model Rating (order=internal) = design ;
   weight Weight;
run;
```

The PROC statement invokes PROC SURVEYLOGISTIC. The TOTAL= option specifies the data set Enrollment, which contains the population totals in the strata. The population totals are used to calculate the finite population correction factor in the variance estimates. The response variable Rating is in the ordinal scale. A cumulative logit model is used to investigate the responses to the Web designs. In the MODEL statement, rating is the response variable, and Design is the effect in the regression model. The ORDER=INTERNAL option is used for the response variable Rating to sort the ordinal response levels of Rating by its internal (numerical) values rather than by the formatted values (for example, 'like very much'). Because the sample design involves stratified

simple random sampling, the STRATA statement is used to specify the stratification variable Class. The WEIGHT statement specifies the variable Weight for sampling weights.

The sample and analysis summary is shown in Output 84.1.2. There are five response levels for the Rating, with 'dislike very much' as the lowest ordered value. The regression model is modeling lower cumulative probabilities by using logit as the link function. Because the TOTAL= option is used, the finite population correction is included in the variance estimation. The sampling weight is also used in the analysis.

Output 84.1.2 Web Design Survey, Model Information

The SURVEYLOGISTIC Procedure

Model Information

Data Set	WORK.WEBSURVEY
Response Variable	Rating
Number of Response Levels	5
Frequency Variable	Count
Stratum Variable	Class
Number of Strata	4
Weight Variable	Weight
Model	Cumulative Logit
Optimization Technique	Fisher's Scoring
Variance Adjustment	Degrees of Freedom (DF)
Finite Population Correction	Used

Response Profile

Ordered Value	Rating	Total Frequency	Total Weight
1	dislike very much	116	1489.0733
2	dislike	227	2933.0433
3	neutral	338	4363.3767
4	like	283	3606.8067
5	like very much	236	3005.7000

Probabilities modeled are cumulated over the lower Ordered Values.

In Output 84.1.3, the score chi-square for testing the proportional odds assumption is 98.1957, which is highly significant. This indicates that the cumulative logit model might not adequately fit the data.

Output 84.1.3 Web Design Survey, Testing the Proportional Odds Assumption

Score Test for the Proportional Odds Assumption

Chi-Square	DF	Pr > ChiSq
98.1957	6	<.0001

An alternative model is to use the generalized logit model with the LINK=GLOGIT option, as shown in the following SAS statements:

```
proc surveylogistic data=WebSurvey total=Enrollment;
   stratum Class;
   freq Count;
   class Design;
   model Rating (ref='neutral') = Design /link=glogit;
   weight Weight;
run;
```

The REF='neutral' option is used for the response variable Rating to indicate that all other response levels are referenced to the level 'neutral.' The option LINK=GLOGIT option requests that the procedure fit a generalized logit model.

The summary of the analysis is shown in Output 84.1.4, which indicates that the generalized logit model is used in the analysis.

Output 84.1.4 Web Design Survey, Model Information

The SURVEYLOGISTIC Procedure

Model Information

Data Set	WORK.WEBSURVEY
Response Variable	Rating
Number of Response Levels	5
Frequency Variable	Count
Stratum Variable	Class
Number of Strata	4
Weight Variable	Weight
Model	Generalized Logit
Optimization Technique	Newton-Raphson
Variance Adjustment	Degrees of Freedom (DF)
Finite Population Correction	Used

Response Profile

Ordered Value	Rating	Total Frequency	Total Weight
1	dislike	227	2933.0433
2	dislike very much	116	1489.0733
3	like	283	3606.8067
4	like very much	236	3005.7000
5	neutral	338	4363.3767

Logits modeled use Rating='neutral' as the reference category.

Output 84.1.5 shows the parameterization for the main effect Design.

Output 84.1.5 Web Design Survey, Class Level Information

Class Level Information

Class	Value	Design Variables	
Design	A	1	0
	B	0	1
	C	−1	−1

The parameter and odds ratio estimates are shown in Output 84.1.6. For each odds ratio estimate, the 95% confidence limits shown in the table contain the value 1.0. Therefore, no conclusion about which Web design is preferred can be made based on this survey.

Output 84.1.6 Web Design Survey, Parameter and Odds Ratio Estimates

Analysis of Maximum Likelihood Estimates

Parameter		Rating	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept		dislike	1	−0.3964	0.0832	22.7100	<.0001
Intercept		dislike very much	1	−1.0826	0.1045	107.3889	<.0001
Intercept		like	1	−0.1892	0.0780	5.8888	0.0152
Intercept		like very much	1	−0.3767	0.0824	20.9223	<.0001
Design	A	dislike	1	−0.0942	0.1166	0.6518	0.4195
Design	A	dislike very much	1	−0.0647	0.1469	0.1940	0.6596
Design	A	like	1	−0.1370	0.1104	1.5400	0.2146
Design	A	like very much	1	0.0446	0.1130	0.1555	0.6933
Design	B	dislike	1	0.0391	0.1201	0.1057	0.7451
Design	B	dislike very much	1	0.2721	0.1448	3.5294	0.0603
Design	B	like	1	0.1669	0.1102	2.2954	0.1298
Design	B	like very much	1	0.1420	0.1174	1.4641	0.2263

Odds Ratio Estimates

Effect	Rating	Point Estimate	95% Wald Confidence Limits	
Design A vs C	dislike	0.861	0.583	1.272
Design A vs C	dislike very much	1.153	0.692	1.923
Design A vs C	like	0.899	0.618	1.306
Design A vs C	like very much	1.260	0.851	1.865
Design B vs C	dislike	0.984	0.659	1.471
Design B vs C	dislike very much	1.615	0.975	2.675
Design B vs C	like	1.218	0.838	1.768
Design B vs C	like very much	1.389	0.925	2.086

Example 84.2: The Medical Expenditure Panel Survey (MEPS)

The U.S. Department of Health and Human Services conducts the Medical Expenditure Panel Survey (MEPS) to produce national and regional estimates of various aspects of health care. The MEPS has a complex sample design that includes both stratification and clustering. The sampling weights are adjusted for nonresponse and raked with respect to population control totals from the Current Population Survey. See the MEPS Survey Background (2006) and Machlin, Yu, and Zodet (2005) for details.

In this example, the 1999 full-year consolidated data file HC-038 (MEPS HC-038, 2002) from the MEPS is used to investigate the relationship between medical insurance coverage and the demographic variables. The data can be downloaded directly from the Agency for Healthcare Research and Quality (AHRQ) Web site at http://www.meps.ahrq.gov/mepsweb/data_stats/download_data_files_detail.jsp?cboPufNumber=HC-038 in either ASCII format or SAS transport format. The Web site includes a detailed description of the data as well as the SAS program used to access and format it.

For this example, the SAS transport format data file for HC-038 is downloaded to 'C:H38.ssp' on a Windows-based PC. The instructions on the Web site lead to the following SAS statements for creating a SAS data set MEPS, which contains only the sample design variables and other variables necessary for this analysis.

```
proc format;
   value racex
      -9 = 'NOT ASCERTAINED'
      -8 = 'DK'
      -7 = 'REFUSED'
      -1 = 'INAPPLICABLE'
      1 = 'AMERICAN INDIAN'
      2 = 'ALEUT, ESKIMO'
      3 = 'ASIAN OR PACIFIC ISLANDER'
      4 = 'BLACK'
      5 = 'WHITE'
      91 = 'OTHER'
      ;
   value sex
      -9 = 'NOT ASCERTAINED'
      -8 = 'DK'
      -7 = 'REFUSED'
      -1 = 'INAPPLICABLE'
      1 = 'MALE'
      2 = 'FEMALE'
      ;
   value povcat9h
      1 = 'NEGATIVE OR POOR'
      2 = 'NEAR POOR'
      3 = 'LOW INCOME'
      4 = 'MIDDLE INCOME'
      5 = 'HIGH INCOME'
      ;
   value inscov9f
```

```
      1 = 'ANY PRIVATE'
      2 = 'PUBLIC ONLY'
      3 = 'UNINSURED'
      ;
   run;
libname mylib '';
filename in1 'H38.SSP';
proc xcopy in=in1 out=mylib import;
run;
data mylib.meps; set mylib.H38;
   label racex= sex= inscov99= povcat99=
      varstr99= varpsu99= perwt99f= totexp99=;
   format racex racex. sex sex.
      povcat99 povcat9h. inscov99 inscov9f.;
   keep inscov99 sex racex povcat99 varstr99
      varpsu99 perwt99f totexp99;
run;
```

There are a total of 24,618 observations in this SAS data set. Each observation corresponds to a person in the survey. The stratification variable is VARSTR99, which identifies the 143 strata in the sample. The variable VARPSU99 identifies the 460 PSUs in the sample. The sampling weights are stored in the variable PERWT99F. The response variable is the health insurance coverage indicator variable, INSCOV99, which has three values:

1	the person had any private insurance coverage any time during 1999
2	the person had only public insurance coverage during 1999
3	the person was uninsured during all of 1999

The demographic variables include gender (SEX), race (RACEX), and family income level as a percent of the poverty line (POVCAT99). The variable RACEX has five categories:

1	American Indian
2	Aleut, Eskimo
3	Asian or Pacific Islander
4	Black
5	White

The variable POVCAT99 is constructed by dividing family income by the applicable poverty line (based on family size and composition), with the resulting percentages grouped into five categories:

1	negative or poor (less than 100%)
2	near poor (100% to less than 125%)
3	low income (125% to less than 200%)
4	middle income (200% to less than 400%)
5	high income (greater than or equal to 400%)

The data set also contains the total health care expenditure in 1999, TOTEXP99, which is used as a covariate in the analysis.

Output 84.2.1 displays the first 30 observations of this data set.

Output 84.2.1 1999 Full-Year MEPS (First 30 Observations)

Obs	SEX	RACE	POVCAT99	INSCOV99	TOTEXP99	PERWT99F	VARSTR99	VARPSU99
1	MALE	WHITE	MIDDLE INCOME	PUBLIC ONLY	2735	14137.86	131	2
2	FEMALE	WHITE	MIDDLE INCOME	ANY PRIVATE	6687	17050.99	131	2
3	MALE	WHITE	MIDDLE INCOME	ANY PRIVATE	60	35737.55	131	2
4	MALE	WHITE	MIDDLE INCOME	ANY PRIVATE	60	35862.67	131	2
5	FEMALE	WHITE	MIDDLE INCOME	ANY PRIVATE	786	19407.11	131	2
6	MALE	WHITE	MIDDLE INCOME	ANY PRIVATE	345	18499.83	131	2
7	MALE	WHITE	MIDDLE INCOME	ANY PRIVATE	680	18499.83	131	2
8	MALE	WHITE	MIDDLE INCOME	ANY PRIVATE	3226	22394.53	136	1
9	FEMALE	WHITE	MIDDLE INCOME	ANY PRIVATE	2852	27008.96	136	1
10	MALE	WHITE	MIDDLE INCOME	ANY PRIVATE	112	25108.71	136	1
11	MALE	WHITE	MIDDLE INCOME	ANY PRIVATE	3179	17569.81	136	1
12	MALE	WHITE	MIDDLE INCOME	ANY PRIVATE	168	21478.06	136	1
13	FEMALE	WHITE	MIDDLE INCOME	ANY PRIVATE	1066	21415.68	136	1
14	MALE	WHITE	NEGATIVE OR POOR	PUBLIC ONLY	0	12254.66	125	1
15	MALE	WHITE	NEGATIVE OR POOR	ANY PRIVATE	0	17699.75	125	1
16	FEMALE	WHITE	NEGATIVE OR POOR	UNINSURED	0	18083.15	125	1
17	MALE	BLACK	NEGATIVE OR POOR	PUBLIC ONLY	230	6537.97	78	10
18	MALE	WHITE	LOW INCOME	UNINSURED	408	8951.36	95	2
19	FEMALE	WHITE	LOW INCOME	UNINSURED	0	11833.00	95	2
20	MALE	WHITE	LOW INCOME	UNINSURED	40	12754.07	95	2
21	FEMALE	WHITE	LOW INCOME	UNINSURED	51	14698.57	95	2
22	MALE	WHITE	LOW INCOME	UNINSURED	0	3890.20	92	19
23	FEMALE	WHITE	LOW INCOME	UNINSURED	610	5882.29	92	19
24	MALE	WHITE	LOW INCOME	PUBLIC ONLY	24	8610.47	92	19
25	FEMALE	BLACK	MIDDLE INCOME	UNINSURED	1758	0.00	64	1
26	MALE	BLACK	MIDDLE INCOME	PUBLIC ONLY	551	7049.70	64	1
27	MALE	BLACK	MIDDLE INCOME	ANY PRIVATE	65	34067.03	64	1
28	FEMALE	BLACK	NEGATIVE OR POOR	PUBLIC ONLY	0	9313.84	73	12
29	FEMALE	BLACK	NEGATIVE OR POOR	PUBLIC ONLY	10	14697.03	73	12
30	MALE	BLACK	NEGATIVE OR POOR	PUBLIC ONLY	0	4574.73	73	12

The following SAS statements fit a generalized logit model for the 1999 full-year consolidated MEPS data:

```
proc surveylogistic data=meps;
   stratum VARSTR99;
   cluster VARPSU99;
   weight PERWT99F;
   class SEX RACEX POVCAT99;
   model INSCOV99 = TOTEXP99 SEX RACEX POVCAT99 / link=glogit;
run;
```

The STRATUM statement specifies the stratification variable VARSTR99. The CLUSTER statement specifies the PSU variable VARPSU99. The WEIGHT statement specifies the sample weight variable PERWT99F. The demographic variables SEX, RACEX, and POVCAT99 are listed in the CLASS statement to indicate that they are categorical independent variables in the MODEL statement. In the MODEL statement, the response variable is INSCOV99, and the independent variables are TOTEXP99 along with the selected demographic variables. The LINK= option requests that the procedure fit the generalized logit model because the response variable INSCOV99 has nominal responses.

The results of this analysis are shown in the following outputs.

PROC SURVEYLOGISTIC lists the model fitting information and sample design information in Output 84.2.2.

Output 84.2.2 MEPS, Model Information

The SURVEYLOGISTIC Procedure

Model Information

Data Set	MYLIB.MEPS
Response Variable	INSCOV99
Number of Response Levels	3
Stratum Variable	VARSTR99
Number of Strata	143
Cluster Variable	VARPSU99
Number of Clusters	460
Weight Variable	PERWT99F
Model	Generalized Logit
Optimization Technique	Newton-Raphson
Variance Adjustment	Degrees of Freedom (DF)

Output 84.2.3 displays the number of observations and the total of sampling weights both in the data set and used in the analysis. Only the observations with positive person-level weight are used in the analysis. Therefore, 1,053 observations with zero person-level weights were deleted.

Output 84.2.3 MEPS, Number of Observations

Number of Observations Read	24618
Number of Observations Used	23565
Sum of Weights Read	2.7641E8
Sum of Weights Used	2.7641E8

Output 84.2.4 lists the three insurance coverage levels for the response variable INSCOV99. The "UNINSURED" category is used as the reference category in the model.

Output 84.2.4 MEPS, Response Profile

Response Profile

Ordered Value	INSCOV99	Total Frequency	Total Weight
1	ANY PRIVATE	16130	204403997
2	PUBLIC ONLY	4241	41809572
3	UNINSURED	3194	30197198

Logits modeled use INSCOV99='UNINSURED' as the reference category.

Output 84.2.5 shows the parameterization in the regression model for each categorical independent variable.

Output 84.2.5 MEPS, Classification Levels

Class Level Information

Class	Value	Design Variables			
SEX	FEMALE	1			
	MALE	-1			
RACEX	ALEUT, ESKIMO	1	0	0	0
	AMERICAN INDIAN	0	1	0	0
	ASIAN OR PACIFIC ISLANDER	0	0	1	0
	BLACK	0	0	0	1
	WHITE	-1	-1	-1	-1
POVCAT99	HIGH INCOME	1	0	0	0
	LOW INCOME	0	1	0	0
	MIDDLE INCOME	0	0	1	0
	NEAR POOR	0	0	0	1
	NEGATIVE OR POOR	-1	-1	-1	-1

Output 84.2.6 displays the parameter estimates and their standard errors.

Output 84.2.6 MEPS, Parameter Estimates

Analysis of Maximum Likelihood Estimates

Parameter		INSCOV99	DF	Estimate	Standard Error	Wald Chi-Square
Intercept		ANY PRIVATE	1	2.7703	0.2911	90.5749
Intercept		PUBLIC ONLY	1	1.9216	0.2663	52.0571
TOTEXP99		ANY PRIVATE	1	0.000215	0.000071	9.1900
TOTEXP99		PUBLIC ONLY	1	0.000241	0.000072	11.1514
SEX	FEMALE	ANY PRIVATE	1	0.1208	0.0248	23.7175
SEX	FEMALE	PUBLIC ONLY	1	0.1741	0.0308	31.9573
RACEX	ALEUT, ESKIMO	ANY PRIVATE	1	7.1457	1.0170	49.3692
RACEX	ALEUT, ESKIMO	PUBLIC ONLY	1	7.6303	0.9091	70.4522
RACEX	AMERICAN INDIAN	ANY PRIVATE	1	−2.0904	0.3402	37.7540
RACEX	AMERICAN INDIAN	PUBLIC ONLY	1	−1.8992	0.3682	26.6109
RACEX	ASIAN OR PACIFIC ISLANDER	ANY PRIVATE	1	−1.8055	0.2809	41.3276
RACEX	ASIAN OR PACIFIC ISLANDER	PUBLIC ONLY	1	−1.9914	0.2916	46.6388
RACEX	BLACK	ANY PRIVATE	1	−1.7517	0.2745	40.7079
RACEX	BLACK	PUBLIC ONLY	1	−1.7038	0.2509	46.1058
POVCAT99	HIGH INCOME	ANY PRIVATE	1	1.4560	0.0685	452.1835
POVCAT99	HIGH INCOME	PUBLIC ONLY	1	−0.6092	0.0903	45.5392
POVCAT99	LOW INCOME	ANY PRIVATE	1	−0.3066	0.0666	21.1762
POVCAT99	LOW INCOME	PUBLIC ONLY	1	−0.0239	0.0754	0.1007
POVCAT99	MIDDLE INCOME	ANY PRIVATE	1	0.6467	0.0587	121.1736
POVCAT99	MIDDLE INCOME	PUBLIC ONLY	1	−0.3496	0.0807	18.7732
POVCAT99	NEAR POOR	ANY PRIVATE	1	−0.8015	0.1076	55.4443
POVCAT99	NEAR POOR	PUBLIC ONLY	1	0.2985	0.0952	9.8308

Analysis of Maximum Likelihood Estimates

Parameter		INSCOV99	Pr > ChiSq
Intercept		ANY PRIVATE	<.0001
Intercept		PUBLIC ONLY	<.0001
TOTEXP99		ANY PRIVATE	0.0024
TOTEXP99		PUBLIC ONLY	0.0008
SEX	FEMALE	ANY PRIVATE	<.0001
SEX	FEMALE	PUBLIC ONLY	<.0001
RACEX	ALEUT, ESKIMO	ANY PRIVATE	<.0001
RACEX	ALEUT, ESKIMO	PUBLIC ONLY	<.0001
RACEX	AMERICAN INDIAN	ANY PRIVATE	<.0001
RACEX	AMERICAN INDIAN	PUBLIC ONLY	<.0001
RACEX	ASIAN OR PACIFIC ISLANDER	ANY PRIVATE	<.0001
RACEX	ASIAN OR PACIFIC ISLANDER	PUBLIC ONLY	<.0001
RACEX	BLACK	ANY PRIVATE	<.0001
RACEX	BLACK	PUBLIC ONLY	<.0001
POVCAT99	HIGH INCOME	ANY PRIVATE	<.0001
POVCAT99	HIGH INCOME	PUBLIC ONLY	<.0001
POVCAT99	LOW INCOME	ANY PRIVATE	<.0001
POVCAT99	LOW INCOME	PUBLIC ONLY	0.7510
POVCAT99	MIDDLE INCOME	ANY PRIVATE	<.0001
POVCAT99	MIDDLE INCOME	PUBLIC ONLY	<.0001
POVCAT99	NEAR POOR	ANY PRIVATE	<.0001
POVCAT99	NEAR POOR	PUBLIC ONLY	0.0017

Output 84.2.7 displays the odds ratio estimates and their standard errors.

Output 84.2.7 MEPS, Odds Ratios

Odds Ratio Estimates

Effect			INSCOV99	Point Estimate
TOTEXP99			ANY PRIVATE	1.000
TOTEXP99			PUBLIC ONLY	1.000
SEX	FEMALE vs MALE		ANY PRIVATE	1.273
SEX	FEMALE vs MALE		PUBLIC ONLY	1.417
RACEX	ALEUT, ESKIMO	vs WHITE	ANY PRIVATE	>999.999
RACEX	ALEUT, ESKIMO	vs WHITE	PUBLIC ONLY	>999.999
RACEX	AMERICAN INDIAN	vs WHITE	ANY PRIVATE	0.553
RACEX	AMERICAN INDIAN	vs WHITE	PUBLIC ONLY	1.146
RACEX	ASIAN OR PACIFIC ISLANDER	vs WHITE	ANY PRIVATE	0.735
RACEX	ASIAN OR PACIFIC ISLANDER	vs WHITE	PUBLIC ONLY	1.045
RACEX	BLACK	vs WHITE	ANY PRIVATE	0.776
RACEX	BLACK	vs WHITE	PUBLIC ONLY	1.394
POVCAT99	HIGH INCOME	vs NEGATIVE OR POOR	ANY PRIVATE	11.595
POVCAT99	HIGH INCOME	vs NEGATIVE OR POOR	PUBLIC ONLY	0.274
POVCAT99	LOW INCOME	vs NEGATIVE OR POOR	ANY PRIVATE	1.990
POVCAT99	LOW INCOME	vs NEGATIVE OR POOR	PUBLIC ONLY	0.492
POVCAT99	MIDDLE INCOME	vs NEGATIVE OR POOR	ANY PRIVATE	5.162
POVCAT99	MIDDLE INCOME	vs NEGATIVE OR POOR	PUBLIC ONLY	0.356
POVCAT99	NEAR POOR	vs NEGATIVE OR POOR	ANY PRIVATE	1.213
POVCAT99	NEAR POOR	vs NEGATIVE OR POOR	PUBLIC ONLY	0.680

Odds Ratio Estimates

95% Wald Confidence Limits	
1.000	1.000
1.000	1.000
1.155	1.403
1.255	1.598
483.788	>999.999
>999.999	>999.999
0.340	0.901
0.603	2.180
0.499	1.083
0.656	1.667
0.638	0.944
1.129	1.720
9.301	14.455
0.213	0.353
1.607	2.464
0.395	0.614
4.200	6.343
0.280	0.451
0.903	1.630
0.527	0.877

For example, after adjusting for the effects of sex, race, and total health care expenditures, a person with high income is estimated to be 11.595 times more likely than a poor person to choose private health care insurance over no insurance, but only 0.274 times as likely to choose public health insurance over no insurance.

References

Agresti, A. (1984), *Analysis of Ordinal Categorical Data*, New York: John Wiley & Sons.

Agresti, A. (1990), *Categorical Data Analysis*, New York: John Wiley & Sons.

Aitchison, J. and Silvey, S. D. (1957), "The Generalization of Probit Analysis to the Case of Multiple Responses," *Biometrika*, 44, 131–40.

Albert, A. and Anderson, J. A. (1984), "On the Existence of Maximum Likelihood Estimates in Logistic Regression Models," *Biometrika*, 71, 1–10.

Ashford, J. R. (1959), "An Approach to the Analysis of Data for Semi-quantal Responses in Biology Response," *Biometrics*, 15, 573–81.

Binder, D. A. (1981), "On the Variances of Asymptotically Normal Estimators from Complex Surveys," *Survey Methodology*, 7, 157–170.

Binder, D. A. (1983), "On the Variances of Asymptotically Normal Estimators from Complex Surveys," *International Statistical Review*, 51, 279–292.

Binder, D. A. and Roberts, G. R. (2003), "Design-Based and Model-Based Methods for Estimating Model Parameters," in *Analysis of Survey Data,* ed. C. Skinner and R. Chambers, New York: John Wiley & Sons.

Brick, J. M. and Kalton, G. (1996), "Handling Missing Data in Survey Research," *Statistical Methods in Medical Research*, 5, 215–238.

Cochran, W. G. (1977), *Sampling Techniques*, Third Edition, New York: John Wiley & Sons.

Collett, D. (1991), *Modelling Binary Data*, London: Chapman & Hall.

Cox, D. R. and Snell, E. J. (1989), *The Analysis of Binary Data*, Second Edition, London: Chapman & Hall.

Dippo, C. S., Fay, R. E., and Morganstein, D. H. (1984), "Computing Variances from Complex Samples with Replicate Weights," *Proceedings of the Survey Research Methods Section, ASA*, 489–494.

Fay, R. E. (1984), "Some Properties of Estimators of Variance Based on Replication Methods," *Proceedings of the Survey Research Methods Section, ASA*, 495–500.

Fay, R. E. (1989), "Theory and Application of Replicate Weighting for Variance Calculations,"

Proceedings of the Survey Research Methods Section, ASA, 212–217.

Freeman, D. H., Jr. (1987), *Applied Categorical Data Analysis*, New York: Marcel Dekker.

Fuller, W. A. (1975), "Regression Analysis for Sample Survey," *Sankhyā*, 37 (3), Series C, 117–132.

Hanley, J. A. and McNeil, B. J. (1982), "The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) Curve," *Radiology*, 143, 29–36.

Hidiroglou, M. A., Fuller, W. A., and Hickman, R. D. (1980), *SUPER CARP*, Ames: Statistical Laboratory, Iowa State University.

Hosmer, D. W, Jr. and Lemeshow, S. (2000), *Applied Logistic Regression*, Second Edition, New York: John Wiley & Sons.

Judkins, D. (1990), "Fay's Method for Variance Estimation," *Journal of Official Statistics*, 6, 223–239.

Kalton, G., and Kaspyzyk, D. (1986), "The Treatment of Missing Survey Data," *Survey Methodology*, 12, 1–16.

Kish, L. (1965), *Survey Sampling*, New York: John Wiley & Sons.

Korn, E. and Graubard B. (1999), *Analysis of Health Survey*, New York: John Wiley & Sons.

Lancaster, H. O. (1961), "Significance Tests in Discrete Distributions," *JASA*, 56, 223–234.

Lehtonen, R. and Pahkinen E. (1995), *Practical Methods for Design and Analysis of Complex Surveys*, Chichester: John Wiley & Sons.

Lohr, S. L. (1999), *Sampling: Design and Analysis*, Pacific Grove, CA: Duxbury Press.

Machlin, S., Yu, W., and Zodet, M. (2005), "Computing Standard Errors for MEPS Estimates," Agency for Healthcare Research and Quality, Rockville, MD [http://www.meps.ahrq.gov/mepsweb/survey_comp/standard_errors.jsp].

McCullagh, P. and Nelder, J. A. (1989), *Generalized Linear Models*, London: Chapman & Hall.

McFadden, D. (1974), "Conditional Logit Analysis of Qualitative Choice Behaviour," in *Frontiers in Econometrics*, ed. by P. Zarembka, New York: Academic Press.

MEPS HC-038: 1999 Full Year Consolidated Data File, October 2002, Agency for Healthcare Research and Quality, Rockville, MD [http://www.meps.ahrq.gov/mepsweb/data_stats/download_data_files_detail.jsp?cboPufNumber=HC-038].

MEPS Survey Background, September 2006, Agency for Healthcare Research and Quality, Rockville, MD [http://www.meps.ahrq.gov/mepsweb/about_meps/survey_back.jsp].

Morel, J. G. (1989) "Logistic Regression under Complex Survey Designs," *Survey Methodology*, 15, 203–223.

Nagelkerke, N. J. D. (1991), "A Note on a General Definition of the Coefficient of Determination," *Biometrika*, 78, 691–692.

Nelder, J. A. and Wedderburn, R. W. M. (1972), "Generalized Linear Models," *Journal of the Royal Statistical Society, Series A*, 135, 761–768.

Rao, J. N. K., and Shao, J. (1996), "On Balanced Half Sample Variance Estimation in Stratified Sampling," *Journal of the American Statistical Association,* 91, 343–348.

Rao, J. N. K. and Shao, J. (1999), "Modified Balanced Repeated Replication for Complex Survey Data," *Biometrika,* 86, 403–415.

Rao, J. N. K., Wu, C. F. J., and Yue, K. (1992), " Some Recent Work on Resampling Methods for Complex Surveys," *Survey Methodology,* 18, 209–217.

Roberts, G., Rao, J. N. K., and Kumar, S. (1987), "Logistic Regression Analysis of Sample Survey Data," *Biometrika*, 74, 1–12.

Rust, K. (1985), "Variance Estimation for Complex Estimators in Sample Surveys," *Journal of Official Statistics,* 1, 381–397.

Rust, K. and Kalton, G. (1987), "Strategies for Collapsing Strata for Variance Estimation," *Journal of Official Statistics,* 3, 69–81.

Santner, T. J. and Duffy, E. D. (1986), "A Note on A. Albert and J. A. Anderson's Conditions for the Existence of Maximum Likelihood Estimates in Logistic Regression Models," *Biometrika*, 73, 755–758.

Särndal, C. E., Swensson, B., and Wretman, J. (1992), *Model Assisted Survey Sampling*, New York: Springer-Verlag.

Skinner, C. J., Holt, D., and Smith, T. M. F. (1989), *Analysis of Complex Surveys*, New York: John Wiley & Sons.

Stokes, M. E., Davis, C. S., and Koch, G. G. (2000), *Categorical Data Analysis Using the SAS System*, Second Edition, Cary: SAS Institute.

Walker, S. H. and Duncan, D. B. (1967), "Estimation of the Probability of an Event as a Function of Several Independent Variables," *Biometrika*, 54, 167–179.

Wolter, K. M. (1985), *Introduction to Variance Estimation*, New York: Springer-Verlag.

Woodruff, R. S. (1971), "A Simple Method for Approximating the Variance of a Complicated Estimate," *Journal of the American Statistical Association*, 66, 411–414.

Chapter 85

The SURVEYMEANS Procedure

Contents

Overview: SURVEYMEANS Procedure **6466**
Getting Started: SURVEYMEANS Procedure **6467**
Simple Random Sampling 6467
Stratified Sampling 6469
Output Data Sets 6472
Syntax: SURVEYMEANS Procedure **6473**
PROC SURVEYMEANS Statement 6474
BY Statement 6483
CLASS Statement 6484
CLUSTER Statement 6485
DOMAIN Statement 6485
RATIO Statement 6486
REPWEIGHTS Statement 6488
STRATA Statement 6489
VAR Statement 6490
WEIGHT Statement 6490
Details: SURVEYMEANS Procedure **6491**
Missing Values 6491
Survey Data Analysis 6492
Specification of Population Totals and Sampling Rates 6492
Primary Sampling Units (PSUs) 6493
Domain Analysis 6493
Statistical Computations 6493
Definitions and Notation 6494
Mean 6495
Variance and Standard Error of the Mean 6496
Ratio 6497
t Test for the Mean 6498
Degrees of Freedom 6498
Confidence Limits for the Mean 6499
Coefficient of Variation 6500
Proportions 6500
Total 6500
Variance and Standard Deviation of the Total 6501

Confidence Limits for the Total 6501
Domain Statistics 6502
Quantiles 6504
Replication Methods for Variance Estimation 6506
Balanced Repeated Replication (BRR) Method 6506
Fay's BRR Method 6507
Jackknife Method 6508
Hadamard Matrix 6509
Output Data Sets 6509
Replicate Weights Output Data Set 6509
Jackknife Coefficients Output Data Set 6510
Rectangular and Stacking Structures in an Output Data Set 6510
Displayed Output 6512
Data and Sample Design Summary 6512
Class Level Information 6513
Stratum Information 6513
Variance Estimation 6513
Statistics 6514
Quantiles 6515
Domain Analysis 6516
Ratio Analysis 6516
Hadamard Matrix 6517
ODS Table Names 6517
Examples: SURVEYMEANS Procedure **6518**
Example 85.1: Stratified Cluster Sample Design 6518
Example 85.2: Domain Analysis 6522
Example 85.3: Ratio Analysis 6525
Example 85.4: Analyzing Survey Data with Missing Values 6526
Example 85.5: Variance Estimation Using Replication Methods 6528
References **6531**

Overview: SURVEYMEANS Procedure

The SURVEYMEANS procedure estimates characteristics of a survey population by using statistics computed from a survey sample. You can estimate statistics such as means, totals, proportions, quantiles, and ratios. PROC SURVEYMEANS also provides domain analysis, which computes estimates for subpopulations or domains. The procedure also estimates variances and confidence limits and performs *t* tests for these statistics. PROC SURVEYMEANS uses either the Taylor series (linearization) method or replication (subsampling) methods to estimate sampling errors of estimators based on complex sample designs. The sample design can be a complex survey sample design with stratification, clustering, and unequal weighting. See Lohr (1999), Särndal, Swensson, and Wretman (1992), and Wolter (1985) for more details.

Getting Started: SURVEYMEANS Procedure

This section demonstrates how you can use the SURVEYMEANS procedure to produce descriptive statistics from sample survey data. For a complete description of PROC SURVEYMEANS, see the section "Syntax: SURVEYMEANS Procedure" on page 6473. The section "Examples: SURVEYMEANS Procedure" on page 6518 provides more complicated examples to illustrate the applications of PROC SURVEYMEANS.

Simple Random Sampling

This example illustrates how you can use PROC SURVEYMEANS to estimate population means and proportions from sample survey data. The study population is a junior high school with a total of 4,000 students in grades 7, 8, and 9. Researchers want to know how much these students spend weekly for ice cream, on average, and what percentage of students spend at least $10 weekly for ice cream.

To answer these questions, 40 students were selected from the entire student population by using simple random sampling (SRS). Selection by simple random sampling means that all students have an equal chance of being selected and no student can be selected more than once. Each student selected for the sample was asked how much he or she spends for ice cream per week, on average. The SAS data set IceCream saves the responses of the 40 students:

```
data IceCream;
   input Grade Spending @@;
   if (Spending < 10) then Group='less';
   else Group='more';
   datalines;
7 7  7  7  8 12  9 10  7  1  7 10  7  3  8 20  8 19  7 2
7 2  9 15  8 16  7  6  7  6  7  6  9 15  8 17  8 14  9 8
9 8  9  7  7  3  7 12  7  4  9 14  8 18  9  9  7  2  7 1
7 4  7 11  9  8  8 10  8 13  7  2  9  6  9 11  7  2  7 9
;
```

The variable Grade contains a student's grade. The variable Spending contains a student's response regarding how much he spends per week for ice cream, in dollars. The variable Group is created to indicate whether a student spends at least $10 weekly for ice cream: Group='more' if a student spends at least $10, or Group='less' if a student spends less than $10.

You can use PROC SURVEYMEANS to produce estimates for the entire student population, based on this random sample of 40 students:

```
title1 'Analysis of Ice Cream Spending';
title2 'Simple Random Sample Design';
proc surveymeans data=IceCream total=4000;
   var Spending Group;
   run;
```

The PROC SURVEYMEANS statement invokes the procedure. The TOTAL=4000 option specifies the total number of students in the study population, or school. The procedure uses this total to adjust variance estimates for the effects of sampling from a finite population. The VAR statement names the variables to analyze, Spending and Group.

Figure 85.1 displays the results from this analysis. There are a total of 40 observations used in the analysis. The "Class Level Information" table lists the two levels of the variable Group. This variable is a character variable, and so PROC SURVEYMEANS provides a categorical analysis for it, estimating the relative frequency or proportion for each level. If you want a categorical analysis for a numeric variable, you can name that variable in the CLASS statement.

Figure 85.1 Analysis of Ice Cream Spending

Analysis of Ice Cream Spending
Simple Random Sample Design

The SURVEYMEANS Procedure

Data Summary

Number of Observations	40

Class Level Information

Class Variable	Levels	Values
Group	2	less more

Statistics

Variable	Level	N	Mean	Std Error of Mean	95% CL for Mean	
Spending		40	8.750000	0.845139	7.04054539	10.4594546
Group	less	23	0.575000	0.078761	0.41568994	0.7343101
	more	17	0.425000	0.078761	0.26568994	0.5843101

The "Statistics" table displays the estimates for each analysis variable. By default, PROC SURVEYMEANS displays the number of observations, the estimate of the mean, its standard error, and the 95% confidence limits for the mean. You can obtain other statistics by specifying the corresponding *statistic-keywords* in the PROC SURVEYMEANS statement.

The estimate of the average weekly ice cream expense is $8.75 for students at this school. The standard error of this estimate if $0.85, and the 95% confidence interval for weekly ice cream expense is from $7.04 to $10.46. The analysis variable Group is a character variable, and so PROC SURVEYMEANS analyzes it as categorical, estimating the relative frequency or proportion for each level or category. These estimates are displayed in the Mean column of the "Statistics" table. It is estimated that 57.5% of all students spend less than $10 weekly on ice cream, while 42.5% of the students spend at least $10 weekly. The standard error of each estimate is 7.9%.

Stratified Sampling

Suppose that the sample of students described in the previous section was actually selected by using stratified random sampling. In stratified sampling, the study population is divided into nonoverlapping strata, and samples are selected from each stratum independently.

The list of students in this junior high school was stratified by grade, yielding three strata: grades 7, 8, and 9. A simple random sample of students was selected from each grade. Table 85.1 shows the total number of students in each grade.

Table 85.1 Number of Students by Grade

Grade	Number of Students
7	1,824
8	1,025
9	1,151
Total	4,000

To analyze this stratified sample, you need to provide the population totals for each stratum to PROC SURVEYMEANS. The SAS data set StudentTotals contains the information from Table 85.1:

```
data StudentTotals;
   input Grade _total_; datalines;
7 1824
8 1025
9 1151
;
```

The variable Grade is the stratum identification variable, and the variable _TOTAL_ contains the total number of students for each stratum. PROC SURVEYMEANS requires you to use the variable name _TOTAL_ for the stratum population totals.

The procedure uses the stratum population totals to adjust variance estimates for the effects of sampling from a finite population. If you do not provide population totals or sampling rates, then the procedure assumes that the proportion of the population in the sample is very small, and the computation does not involve a finite population correction.

In a stratified sample design, when the sampling rates in the strata are unequal, you need to use sampling weights to reflect this information in order to produce an unbiased mean estimator. In this example, the appropriate sampling weights are reciprocals of the probabilities of selection. You can use the following DATA step to create the sampling weights:

```
data IceCream;
   set IceCream;
   if Grade=7 then Prob=20/1824;
   if Grade=8 then Prob=9/1025;
   if Grade=9 then Prob=11/1151;
   Weight=1/Prob;
```

When you use PROC SURVEYSELECT to select your sample, the procedure creates these sampling weights for you.

The following SAS statements perform the stratified analysis of the survey data:

```
title1 'Analysis of Ice Cream Spending';
title2 'Stratified Simple Random Sample Design';
proc surveymeans data=IceCream total=StudentTotals;
   stratum Grade / list;
   var Spending Group;
   weight Weight;
run;
```

The PROC SURVEYMEANS statement invokes the procedure. The DATA= option names the SAS data set IceCream as the input data set to be analyzed. The TOTAL= option names the data set StudentTotals as the input data set that contains the stratum population totals. Comparing this to the analysis in the section "Simple Random Sampling" on page 6467, notice that the TOTAL=StudentTotals option is used here instead of the TOTAL=4000 option. In this stratified sample design, the population totals are different for different strata, and so you need to provide them to PROC SURVEYMEANS in a SAS data set.

The STRATA statement identifies the stratification variable Grade. The LIST option in the STRATA statement requests that the procedure display stratum information. The WEIGHT statement tells the procedure that the variable Weight contains the sampling weights.

Figure 85.2 displays information about the input data set. There are three strata in the design and 40 observations in the sample. The categorical variable Group has two levels, 'less' and 'more.'

Figure 85.2 Data Summary

```
                    Analysis of Ice Cream Spending
                Stratified Simple Random Sample Design

                      The SURVEYMEANS Procedure

                            Data Summary

                 Number of Strata                   3
                 Number of Observations            40
                 Sum of Weights                  4000

                       Class Level Information

                 Class
                 Variable      Levels    Values

                 Group              2    less more
```

Figure 85.3 displays information for each stratum. The table displays a stratum index and the values of the STRATA variable. The stratum index identifies each stratum by a sequentially assigned number. For each stratum, the table gives the population total (total number of students), the sampling rate, and the sample size. The stratum sampling rate is the ratio of the number of students in the

sample to the number of students in the population for that stratum. The table also lists each analysis variable and the number of stratum observations for that variable. For categorical variables, the table lists each level and the number of sample observations in that level.

Figure 85.3 Stratum Information

Stratum Information

Stratum Index	Grade	Population Total	Sampling Rate	N Obs	Variable	Level
1	7	1824	1.10%	20	Spending Group	less
						more
2	8	1025	0.88%	9	Spending Group	less
						more
3	9	1151	0.96%	11	Spending Group	less
						more

Stratum Information

Stratum Index	Grade	Population Total	Sampling Rate	N Obs	Variable	N
1	7	1824	1.10%	20	Spending Group	20
						17
						3
2	8	1025	0.88%	9	Spending Group	9
						0
						9
3	9	1151	0.96%	11	Spending Group	11
						6
						5

Figure 85.4 shows the following:

- The estimate of average weekly ice cream expense is $9.14 for students in this school, with a standard error of $0.53, and a 95% confidence interval from $8.06 to $10.22.
- An estimate of 54.5% of all students spend less than $10 weekly on ice cream, and 45.5% spend more, with a standard error of 5.8%.

Figure 85.4 Analysis of Ice Cream Spending

Statistics

Variable	Level	N	Mean	Std Error of Mean	95% CL for Mean	
Spending		40	9.141298	0.531799	8.06377052	10.2188254
Group	less	23	0.544555	0.058424	0.42617678	0.6629323
	more	17	0.455445	0.058424	0.33706769	0.5738232

Output Data Sets

PROC SURVEYMEANS uses the Output Delivery System (ODS) to create output data sets. This is a departure from older SAS procedures that provide OUTPUT statements for similar functionality. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

For example, to save the "Statistics" table shown in Figure 85.4 in the previous section in an output data set, you use the ODS OUTPUT statement as follows:

```
title1 'Analysis of Ice Cream Spending';
title2 'Stratified Simple Random Sample Design';
proc surveymeans data=IceCream total=StudentTotals;
   stratum Grade / list;
   var Spending Group;
   weight Weight;
   ods output Statistics=MyStat;
run;
```

The statement

```
ods output Statistics=MyStat;
```

requests that the "Statistics" table that appears in Figure 85.4 be placed in a SAS data set MyStat.

The PRINT procedure displays observations of the data set MyStat:

```
proc print data=MyStat;
run;
```

Figure 85.5 displays the data set MyStat.

Figure 85.5 The Data Set MyStat

Analysis of Ice Cream Spending
Stratified Simple Random Sample Design

Obs	VarName	Var Level	N	Mean	StdErr	Lower CLMean	Upper CLMean
1	Spending		40	9.141298	0.531799	8.06377052	10.2188254
2	Group	less	23	0.544555	0.058424	0.42617678	0.6629323
3	Group	more	17	0.455445	0.058424	0.33706769	0.5738232

The section "ODS Table Names" on page 6517 gives the complete list of the tables produced by PROC SURVEYMEANS.

Syntax: SURVEYMEANS Procedure

The following statements are available in PROC SURVEYMEANS:

PROC SURVEYMEANS < *options* > < *statistic-keywords* > ;
BY *variables* ;
CLASS *variables* ;
CLUSTER *variables* ;
DOMAIN *variables* < *variable*∗*variable variable*∗*variable*∗*variable ...* > < */ option* > ;
RATIO < *'label'* > *variables / variables* ;
REPWEIGHTS *variables* < */ options* > ;
STRATA *variables* < */ option* > ;
VAR *variables* ;
WEIGHT *variable* ;

The PROC SURVEYMEANS statement invokes the procedure. It optionally names the input data sets, specifies statistics for the procedure to compute, and specifies the variance estimation method. The PROC SURVEYMEANS statement is required.

The VAR statement identifies the variables to be analyzed. The CLASS statement identifies those numeric variables that are to be analyzed as categorical variables. The STRATA statement lists the variables that form the strata in a stratified sample design. The CLUSTER statement specifies cluster identification variables in a clustered sample design. The DOMAIN statement lists the variables that define domains for subpopulation analysis. The RATIO statement requests ratio analysis for means or proportions of analysis variables. The WEIGHT statement names the sampling weight variable. The REPWEIGHTS statement names replicate weight variables for BRR or jackknife variance estimation. You can use a BY statement with PROC SURVEYMEANS to obtain separate analyses for groups defined by the BY variables.

All statements can appear multiple times except the PROC SURVEYMEANS statement and the WEIGHT statement, which can appear only once.

The rest of this section gives detailed syntax information for the BY, CLASS, CLUSTER, DOMAIN, RATIO, REPWEIGHTS, STRATA, VAR, and WEIGHT statements in alphabetical order after the description of the PROC SURVEYMEANS statement.

PROC SURVEYMEANS Statement

PROC SURVEYMEANS < *options* > *statistic-keywords* **;**

The PROC SURVEYMEANS statement invokes the procedure. In this statement, you identify the data set to be analyzed, specify the variance estimation method, and provide sample design information. The DATA= option names the input data set to be analyzed. The VARMETHOD= option specifies the variance estimation method, which is the Taylor series method by default. For Taylor series variance estimation, you can include a finite population correction factor in the analysis by providing either the sampling rate or population total with the RATE= or TOTAL= option. If your design is stratified, with different sampling rates or totals for different strata, then you can input these stratum rates or totals in a SAS data set that contains the stratification variables.

In the PROC SURVEYMEANS statement, you also can use *statistic-keywords* to specify statistics for the procedure to compute. Available statistics include the population mean and population total, together with their variance estimates and confidence limits. You can also request data set summary information and sample design information.

You can specify the following options in the PROC SURVEYMEANS statement.

ALPHA=*α*
sets the confidence level for confidence limits. The value of the ALPHA= option must be between 0 and 1, and the default value is 0.05. A confidence level of α produces $100(1-\alpha)\%$ confidence limits. The default of ALPHA=0.05 produces 95% confidence limits.

DATA=*SAS-data-set*
specifies the SAS data set to be analyzed by PROC SURVEYMEANS. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

MISSING
treats missing values as a valid (nonmissing) category for all categorical variables, which include CLASS, STRATA, CLUSTER, and DOMAIN variables.

By default, if you do not specify the MISSING option, an observation is excluded from the analysis if it has a missing value. For more information, see the section "Missing Values" on page 6491.

NOMCAR
requests that the procedure treat missing values in the variance computation as *not missing completely at random* (NOMCAR) for Taylor series variance estimation. When you specify the NOMCAR option, PROC SURVEYMEANS computes variance estimates by analyzing the nonmissing values as a domain or subpopulation, where the entire population includes both nonmissing and missing domains. See the section "Missing Values" on page 6491 for more details.

By default, PROC SURVEYMEANS completely excludes an observation from analysis if that observation has a missing value, unless you specify the MISSING option. Note that the NOMCAR option has no effect on a categorical variable when you specify the MISSING option, which treats missing values as a valid nonmissing level.

The NOMCAR option applies only to Taylor series variance estimation. The replication methods, which you request with the VARMETHOD=BRR and VARMETHOD=JACKKNIFE options, do not use the NOMCAR option.

NONSYMCL

requests nonsymmetric confidence limits for the quantiles when you request quantiles with PERCENTILE= or QUANTILE= option. See the section "Confidence Limits" on page 6505 for more details.

NOSPARSE

suppresses the display of analysis variables with zero frequency in the output. By default, the procedure displays all continuous variable and all levels of categorical variables.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the order in which the values of the categorical variables are to be reported. The following shows how PROC SURVEYMEANS interprets values of the ORDER= option:

DATA	orders values according to their order in the input data set.
FORMATTED	orders values by their formatted values. This order is operating environment dependent. By default, the order is ascending.
FREQ	order values by descending frequency count; the most observed value comes first in the order.
INTERNAL	orders values by their unformatted values, which yields the same order that the SORT procedure does. This order is operating environment dependent.

By default, ORDER=FORMATTED.

The ORDER= option applies to all the categorical variables. When the default ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values.

PERCENTILE=(*values*)

specifies percentiles you want the procedure to compute. You can separate values with blanks or commas. Each value must be between 0 and 100. You can also use the *statistic-keywords* DECILES, MEDIAN, Q1, Q3, and QUARTILES to request common percentiles.

PROC SURVEYMEANS uses Woodruff method (Dorfman and Valliant 1993; Särndal, Swensson, and Wretman 1992; Francisco and Fuller 1991) to estimate the variances of quantiles. BRR and jackknife variance estimation do not apply to the variance estimation for percentiles. See the section "Quantiles" on page 6504 for more details.

QUANTILE=(*values*)

specifies quantiles you want the procedure to compute. You can separate values with blanks

or commas. Each value must be between 0 and 1. You can also use the *statistic-keywords* DECILES, MEDIAN, Q1, Q3, and QUARTILES to request common quantiles.

PROC SURVEYMEANS uses Woodruff method (Dorfman and Valliant 1993; Särndal, Swensson, and Wretman 1992; Francisco and Fuller 1991) to estimate the variances of quantiles. BRR and jackknife variance estimation do not apply to the variance estimation for quantiles. See the section "Quantiles" on page 6504 for more details.

RATE=*value* | *SAS-data-set*

R=*value* | *SAS-data-set*

specifies the sampling rate as a nonnegative *value*, or specifies an input data set that contains the stratum sampling rates. The procedure uses this information to compute a finite population correction for Taylor series variance estimation. The procedure does not use the RATE= option for BRR or jackknife variance estimation, which you request with the VARMETHOD=BRR or VARMETHOD=JACKKNIFE option.

If your sample design has multiple stages, you should specify the *first-stage sampling rate*, which is the ratio of the number of PSUs selected to the total number of PSUs in the population.

For a nonstratified sample design, or for a stratified sample design with the same sampling rate in all strata, you should specify a nonnegative *value* for the RATE= option. If your design is stratified with different sampling rates in the strata, then you should name a SAS data set that contains the stratification variables and the sampling rates. See the section "Specification of Population Totals and Sampling Rates" on page 6492 for more details.

The *value* in the RATE= option or the values of _RATE_ in the secondary data set must be nonnegative numbers. You can specify *value* as a number between 0 and 1. Or you can specify *value* in percentage form as a number between 1 and 100, and PROC SURVEYMEANS converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

If you do not specify the TOTAL= or RATE= option, then the Taylor series variance estimation does not include a finite population correction. You cannot specify both the TOTAL= and RATE= options.

STACKING

requests that the procedure produce the output data sets by using a stacking table structure, which was the default before SAS 9. The new default is to produce a rectangular table structure in the output data sets.

A rectangular structure creates one observation for each analysis variable in the data set. A stacking structure creates only one observation in the output data set for all analysis variables.

The STACKING option affects the following tables:

- Domain
- Ratio
- Statistics
- StrataInfo

See the section "Rectangular and Stacking Structures in an Output Data Set" on page 6510 for more details.

TOTAL=*value* | *SAS-data-set*
N=*value* | *SAS-data-set*

specifies the total number of primary sampling units in the study population as a positive *value*, or specifies an input data set that contains the stratum population totals. The procedure uses this information to compute a finite population correction for Taylor series variance estimation. The procedure does not use the TOTAL= option for BRR or jackknife variance estimation, which you request with the VARMETHOD=BRR or VARMETHOD=JACKKNIFE option.

For a nonstratified sample design, or for a stratified sample design with the same population total in all strata, you should specify a positive *value* for the TOTAL= option. If your sample design is stratified with different population totals in the strata, then you should name a SAS data set that contains the stratification variables and the population totals. See the section "Specification of Population Totals and Sampling Rates" on page 6492 for more details.

If you do not specify the TOTAL= or RATE= option, then the Taylor series variance estimation does not include a finite population correction. You cannot specify both the TOTAL= and RATE= options.

statistic-keywords

specifies the statistics for the procedure to compute. If you do not specify any *statistic-keywords*, PROC SURVEYMEANS computes the NOBS, MEAN, STDERR, and CLM statistics by default.

The statistics produced depend on the type of the analysis variable. If you name a numeric variable in the CLASS statement, then the procedure analyzes that variable as a categorical variable. The procedure always analyzes character variables as categorical. See the section "CLASS Statement" on page 6484 for more information.

PROC SURVEYMEANS computes MIN, MAX, and RANGE for numeric variables but not for categorical variables. For numeric variables, the keyword MEAN produces the mean, but for categorical variables it produces the proportion in each category or level. Also, for categorical variables, the keyword NOBS produces the number of observations for each variable level, and the keyword NMISS produces the number of missing observations for each level. If you request the keyword NCLUSTER for a categorical variable, PROC SURVEYMEANS displays for each level the number of clusters with observations in that level. PROC SURVEYMEANS computes SUMWGT in the same way for both categorical and numeric variables, as the sum of the weights over all nonmissing observations.

PROC SURVEYMEANS performs univariate analysis, analyzing each variable separately. Thus the number of nonmissing and missing observations might not be the same for all analysis variables. See the section "Missing Values" on page 6491 for more information.

When you use a RATIO statement, available statistics computed for a ratio include N, NCLU, SUMWGT, RATIO, STDERR, DF, T, PROBT, and CLM, as shown in the following list. If no statistics are requested, the procedure computes the ratio and its standard error by default for a RATIO statement.

The valid *statistic-keywords* are as follows:

ALL	all statistics listed
CLM	$100(1-\alpha)\%$ two-sided confidence limits for the MEAN, where α is determined by the ALPHA= option; the default is $\alpha = 0.05$
CLSUM	$100(1-\alpha)\%$ two-sided confidence limits for the SUM, where α is determined by the ALPHA= option; the default is $\alpha = 0.05$
CV	coefficient of variation for MEAN
CVSUM	coefficient of variation for SUM
DECILES	the 10th, the 20th, ..., and the 90th percentiles including their standard errors and confidence limits
DF	degrees of freedom for the *t* test
LCLM	$100(1-\alpha)\%$ one-sided lower confidence limit of the MEAN, where α is determined by the ALPHA= option; the default is $\alpha = 0.05$
LCLSUM	$100(1-\alpha)\%$ one-sided lower confidence limit of the SUM, where α is determined by the ALPHA= option; the default is $\alpha = 0.05$
MAX	maximum value
MEAN	mean for a numeric variable, or the proportion in each category for a categorical variable
MEDIAN	median for a numeric variable
MIN	minimum value
NCLUSTER	number of clusters
NMISS	number of missing observations
NOBS	number of nonmissing observations
Q1	lower quartile
Q3	upper quartile
QUARTILES	lower quartile (25%th percentile), median (50%th percentile), and upper quartile (75%th percentile), including their standard errors and confidence limits
RANGE	range, MAX−MIN
STD	standard deviation of the SUM. When you request SUM, the procedure computes STD by default.
STDERR	standard error of the MEAN or RATIO. When you request MEAN or RATIO, the procedure computes STDERR by default.
SUM	weighted sum, $\sum w_i y_i$, or estimated population total when the appropriate sampling weights are used
SUMWGT	sum of the weights, $\sum w_i$

T	t-value and its corresponding p-value with DF degrees of freedom for $H_0 : \theta = 0$ where θ is a requested statistic
UCLM	$100(1-\alpha)\%$ one-sided upper confidence limit of the MEAN, where α is determined by the ALPHA= option; the default is $\alpha = 0.05$
UCLSUM	$100(1-\alpha)\%$ one-sided upper confidence limit of the SUM, where α is determined by the ALPHA= option; the default is $\alpha = 0.05$
VAR	variance of the MEAN or RATIO
VARSUM	variance of the SUM

See the section "Statistical Computations" on page 6493 for details about how PROC SURVEYMEANS computes these statistics.

VARMETHOD=BRR < (*method-options*) >

VARMETHOD=JACKKNIFE | JK < (*method-options*) >

VARMETHOD=TAYLOR

specifies the variance estimation method. VARMETHOD=TAYLOR requests the Taylor series method, which is the default if you do not specify the VARMETHOD= option or the REPWEIGHTS statement. VARMETHOD=BRR requests variance estimation by balanced repeated replication (BRR), and VARMETHOD=JACKKNIFE requests variance estimation by the delete-1 jackknife method.

For VARMETHOD=BRR and VARMETHOD=JACKKNIFE you can specify *method-options* in parentheses. Table 85.2 summarizes the available *method-options*.

Table 85.2 Variance Estimation Method-Options

Keyword	Method	(*Method-Options*)
BRR	Balanced repeated replication	DFADJ FAY <=*value*> HADAMARD | H=*SAS-data-set* OUTWEIGHTS=*SAS-data-set* PRINTH REPS=*number*
JACKKNIFE | JK	Jackknife	DFADJ OUTJKCOEFS=*SAS-data-set* OUTWEIGHTS=*SAS-data-set*
TAYLOR	Taylor series	

Method-options must be enclosed in parentheses following the method keyword. For example:

```
varmethod=BRR(reps=60 outweights=myReplicateWeights)
```

The following values are available for the VARMETHOD= option:

BRR <(*method-options*)> requests balanced repeated replication (BRR) variance estimation. The BRR method requires a stratified sample design with two primary sampling units (PSUs) per stratum. See the section "Balanced Repeated Replication (BRR) Method" on page 6506 for more information.

You can specify the following *method-options* in parentheses following VARMETHOD=BRR.

DFADJ

computes the degrees of freedom as the number of nonmissing strata for an analysis variable. The degrees of freedom for VARMETHOD=BRR equal the number of strata, which by default is based on all valid observations in the data set. But if you specify the DFADJ method-option, PROC SURVEYMEANS does not count any empty strata due to all observations containing missing values for an analysis variable.

See the section "Degrees of Freedom" on page 6498 for more information. See the section "Data and Sample Design Summary" on page 6512 for details about valid observations.

The DFADJ method-option has no effect on categorical variables when you specify the MISSING option, which treats missing values as a valid nonmissing level.

The DFADJ method-option cannot be used when you provide replicate weights with a REPWEIGHTS statement. When you use a REPWEIGHTS statement, the degrees of freedom equal the number of REPWEIGHTS variables (or replicates), unless you specify an alternative value in the DF= option in the REPWEIGHTS statement.

FAY <=*value***>**

requests Fay's method, a modification of the BRR method, for variance estimation. See the section "Fay's BRR Method" on page 6507 for more information.

You can specify the *value* of the Fay coefficient, which is used in converting the original sampling weights to replicate weights. The Fay coefficient must be a nonnegative number less than 1. By default, the value of the Fay coefficient equals 0.5.

HADAMARD=*SAS-data-set*

H=*SAS-data-set*

names a SAS data set that contains the Hadamard matrix for BRR replicate construction. If you do not provide a Hadamard matrix with the HADAMARD= method-option, PROC SURVEYMEANS generates an appropriate Hadamard matrix for replicate construction. See the sections "Balanced Repeated Replication (BRR)

Method" on page 6506 and "Hadamard Matrix" on page 6509 for details.

If a Hadamard matrix of a given dimension exists, it is not necessarily unique. Therefore, if you want to use a specific Hadamard matrix, you must provide the matrix as a SAS data set in the HADAMARD=*SAS-data-set* method-option.

In the HADAMARD= input data set, each variable corresponds to a column of the Hadamard matrix, and each observation corresponds to a row of the matrix. You can use any variable names in the HADAMARD= data set. All values in the data set must equal either 1 or -1. You must ensure that the matrix you provide is indeed a Hadamard matrix—that is, $\mathbf{A}'\mathbf{A} = R\mathbf{I}$, where $\mathbf{A}$ is the Hadamard matrix of dimension R and $\mathbf{I}$ is an identity matrix. PROC SURVEYMEANS does not check the validity of the Hadamard matrix that you provide.

The HADAMARD= input data set must contain at least H variables, where H denotes the number of first-stage strata in your design. If the data set contains more than H variables, the procedure uses only the first H variables. Similarly, the HADAMARD= input data set must contain at least H observations.

If you do not specify the REPS= method-option, then the number of replicates is taken to be the number of observations in the HADAMARD= input data set. If you specify the number of replicates—for example, REPS=*nreps*—then the first *nreps* observations in the HADAMARD= data set are used to construct the replicates.

You can specify the PRINTH option to display the Hadamard matrix that the procedure uses to construct replicates for BRR.

OUTWEIGHTS=*SAS-data-set*

names a SAS data set that contains replicate weights. See the section "Balanced Repeated Replication (BRR) Method" on page 6506 for information about replicate weights. See the section "Replicate Weights Output Data Set" on page 6509 for more details about the contents of the OUTWEIGHTS= data set.

The OUTWEIGHTS= method-option is not available when you provide replicate weights with the REPWEIGHTS statement.

PRINTH

displays the Hadamard matrix.

When you provide your own Hadamard matrix with the HADAMARD= method-option, only the rows and columns of the Hadamard matrix that are used by the procedure are displayed.

See the sections "Balanced Repeated Replication (BRR) Method" on page 6506 and "Hadamard Matrix" on page 6509 for details.

The PRINTH method-option is not available when you provide replicate weights with the REPWEIGHTS statement because the procedure does not use a Hadamard matrix in this case.

REPS=*number*

specifies the number of replicates for BRR variance estimation. The value of *number* must be an integer greater than 1.

If you do not provide a Hadamard matrix with the HADAMARD= method-option, the number of replicates should be greater than the number of strata and should be a multiple of 4. See the section "Balanced Repeated Replication (BRR) Method" on page 6506 for more information. If a Hadamard matrix cannot be constructed for the REPS= value that you specify, the value is increased until a Hadamard matrix of that dimension can be constructed. Therefore, it is possible for the actual number of replicates used to be larger than the REPS= value that you specify.

If you provide a Hadamard matrix with the HADAMARD= method-option, the value of REPS= must not be less than the number of rows in the Hadamard matrix. If you provide a Hadamard matrix and do not specify the REPS= method-option, the number of replicates equals the number of rows in the Hadamard matrix.

If you do not specify the REPS= or HADAMARD= method-option and do not include a REPWEIGHTS statement, the number of replicates equals the smallest multiple of 4 that is greater than the number of strata.

If you provide replicate weights with the REPWEIGHTS statement, the procedure does not use the REPS= method-option. With a REPWEIGHTS statement, the number of replicates equals the number of REPWEIGHTS variables.

JACKKNIFE | JK <(*method-options*)> requests variance estimation by the delete-1 jackknife method. See the section "Jackknife Method" on page 6508 for details. If you provide replicate weights with a REPWEIGHTS statement, VARMETHOD=JACKKNIFE is the default variance estimation method.

You can specify the following *method-options* in parentheses following VARMETHOD=JACKKNIFE:

DFADJ

computes the degrees of freedom as the number of nonmissing strata for an analysis variable. The degrees of freedom for VARMETHOD=BRR equal the number of strata, which by default is based on all valid observations in the data set. But if you specify the DFADJ method-option, PROC SURVEYMEANS does not

count any empty strata due to all observations containing missing values for an analysis variable.

See the section "Degrees of Freedom" on page 6498 for more information. See the section "Data and Sample Design Summary" on page 6512 for details about valid observations.

The DFADJ method-option has no effect on categorical variables when you specify the MISSING option, which treats missing values as a valid nonmissing level.

The DFADJ method-option cannot be used when you provide replicate weights with a REPWEIGHTS statement. When you use a REPWEIGHTS statement, the degrees of freedom equal the number of REPWEIGHTS variables (or replicates), unless you specify an alternative value in the DF= option in the REPWEIGHTS statement.

OUTWEIGHTS=*SAS-data-set*

names a SAS data set that contains replicate weights. See the section "Jackknife Method" on page 6508 for information about replicate weights. See the section "Replicate Weights Output Data Set" on page 6509 for more details about the contents of the OUTWEIGHTS= data set.

The OUTWEIGHTS= method-option is not available when you provide replicate weights with the REPWEIGHTS statement.

OUTJKCOEFS=*SAS-data-set*

names a SAS data set that contains jackknife coefficients. See the section "Jackknife Method" on page 6508 for information about jackknife coefficients. See the section "Jackknife Coefficients Output Data Set" on page 6510 for more details about the contents of the OUTJKCOEFS= data set.

TAYLOR

requests Taylor series variance estimation. This is the default method if you do not specify the VARMETHOD= option and if there is no REPWEIGHTS statement. See the section "Statistical Computations" on page 6493 for more information.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC SURVEYMEANS to obtain separate analyses on observations in groups defined by the BY variables.

Note that by using a BY statement provides completely separate analyses of the BY groups. It does not provide a statistically valid subpopulation or domain analysis, where the total number of units

in the subpopulation is not known with certainty. You should use the DOMAIN statement to obtain domain analysis.

When a BY statement appears, the procedure expects the input data sets to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If you specify more than one BY statement, the procedure uses only the latest BY statement and ignores any previous ones.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Use the BY statement option NOTSORTED or DESCENDING. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

The CLASS statement names variables to be analyzed as categorical variables. For categorical variables, PROC SURVEYMEANS estimates the proportion in each category or level, instead of the overall mean. PROC SURVEYMEANS always analyzes character variables as categorical. If you want categorical analysis for a numeric variable, you must include that variable in the CLASS statement.

The CLASS *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric. The formatted values of the CLASS variables determine the categorical variable levels. Thus, you can use formats to group values into levels. See the FORMAT procedure in the *Base SAS Procedures Guide* and the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary* for more information.

You can use multiple CLASS statements to specify categorical variables.

When you specify classification variables, you can use the SAS system option SUMSIZE= to limit (or to specify) the amount of memory that is available for data analysis. See the chapter on SAS system options in *SAS Language Reference: Dictionary* for a description of the SUMSIZE= option.

CLUSTER Statement

CLUSTER *variables* **;**

The CLUSTER statement names variables that identify the clusters in a clustered sample design. The combinations of categories of CLUSTER variables define the clusters in the sample. If there is a STRATA statement, clusters are nested within strata.

If you provide replicate weights for BRR or jackknife variance estimation with the REPWEIGHTS statement, you do not need to specify a CLUSTER statement.

If your sample design has clustering at multiple stages, you should identify only the first-stage clusters, or primary sampling units (PSUs), in the CLUSTER statement. See the section "Primary Sampling Units (PSUs)" on page 6493 for more information.

The CLUSTER *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric. The formatted values of the CLUSTER variables determine the CLUSTER variable levels. Thus, you can use formats to group values into levels. See the FORMAT procedure in the *Base SAS Procedures Guide* and the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary* for more information.

You can use multiple CLUSTER statements to specify cluster variables. The SURVEYMEANS procedure uses variables from all CLUSTER statements to create clusters.

DOMAIN Statement

DOMAIN *variables* < *variable*∗*variable variable*∗*variable*∗*variable ...* > < */ option* > **;**

The DOMAIN statement requests analysis for subpopulations, or domains, in addition to analysis for the entire study population. The DOMAIN statement names the variables that identify domains, which are called domain variables.

It is common practice to compute statistics for domains. The formation of these domains might be unrelated to the sample design. Therefore, the sample sizes for the domains are random variables. In order to incorporate this variability into the variance estimation, you should use a DOMAIN statement.

Note that a DOMAIN statement is different from a BY statement. In a BY statement, you treat the sample sizes as fixed in each subpopulation, and you perform analysis within each BY group independently. See the section "Domain Statistics" on page 6502 for more details.

You should use the DOMAIN statement on the entire data set to perform a domain analysis. Creating a new data set from a single domain and analyzing that with SURVEYMEANS yields inappropriate estimates of variance.

A domain variable can be either character or numeric. The procedure treats domain variables as categorical variables. If a variable appears by itself in a DOMAIN statement, each level of this variable determines a domain in the study population. If two or more variables are joined by asterisks (*),

then every possible combination of levels of these variables determines a domain. The procedure performs a descriptive analysis within each domain defined by the domain variables.

The formatted values of the domain variables determine the categorical variable levels. Thus, you can use formats to group values into levels. See the FORMAT procedure in the *Base SAS Procedures Guide* and the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary* for more information.

You can specify the following option in the DOMAIN statement after a slash (/):

DFADJ

computes the degrees of freedom by using the number of nonempty strata for an analysis variable in a domain.

Given a specific domain, it is possible that there is no sampling unit in the sample falling into a domain for a stratum or there is no valid observations from a stratum in a domain due to missing values. Therefore, a domain might contain empty strata. By default, the procedure counts these empty strata when calculating the degrees of freedom.

However, if you specify the DFADJ option, the procedure makes an adjustment by excluding these empty strata when compute the degrees of freedom. Note that prior to SAS 9.2, the procedure made this adjustment by default.

See the section "Degrees of Freedom" on page 6498 for more information. See the section "Data and Sample Design Summary" on page 6512 for details about valid observations.

The DFADJ option has no effect on categorical variables when you specify the MISSING option, which treats missing values as a valid nonmissing level.

RATIO Statement

RATIO < *'label'* > *variables* / *variables* **;**

The RATIO statement requests ratio analysis for means or proportions of analysis variables. A ratio statement names the variables whose means are used as numerators or denominators in a ratio. Variables appearing before the slash (/), called *numerator variables*, are used as numerators. Variables appearing after the slash (/), called *denominator variables*, are used as denominators. These *variables* can be any number of analysis variables, either continuous or categorical in the input data set, except those named in the BY, CLUSTER, REPWEIGHT, STRATA, and WEIGHT statements.

You can optionally specify a label for each RATIO statement to identify the ratios in the output. Labels must be enclosed in single quotes.

If a RATIO statement does not have any numerator variable or denominator variable specified, the RATIO statement is ignored.

The computation of ratios depends on whether the numerator and denominator variables are continuous or categorical.

For continuous variables, ratios are calculated with the mean of the variables. For example, for continuous variables X, Y, Z, and T, the following RATIO statement requests that the procedure analyze the ratios $\bar{x}/\bar{z}$, $\bar{x}/\bar{t}$, $\bar{y}/\bar{z}$, and $\bar{y}/\bar{t}$:

```
ratio x y / z t;
```

If a continuous variable appears as both a numerator and a denominator variable, the ratio of this variable itself is ignored.

For categorical variables, ratios are calculated with the proportions for the categories of a categorical variable. For example, if the categorical variable Gender has the values 'Male' and 'Female,' with the proportions p_m = Pr(Gender='Male') and p_f = Pr(Gender='Female'), and Y is a continuous variable, then the following RATIO statement requests that the procedure analyze the ratios p_m/p_f, p_f/p_m, $\bar{y}/p_m$, and $\bar{y}/p_f$:

```
ratio Gender y / Gender;
```

If a categorical variable appears as both a numerator and denominator variable, then the ratios of the proportions for all categories are computed, except the ratio of each category with itself.

You can have more than one RATIO statement. Each RATIO statement produces ratios independently by using its own numerator and denominator variables. Each RATIO statement also produces its own ratio analysis table.

Available statistics for a ratio are as follows:

- N, number of observations used to compute the ratio
- NCLU, number of clusters
- SUMWGT, sum of weights
- RATIO, ratio
- STDERR, standard error of ratio
- VAR, variance of ratio
- T, *t*-value of ratio
- PROBT, *p*-value of *t*
- DF, degrees of freedom of *t*
- CLM, two-sided confidence limits of ratio
- UCLM, one-sided upper confidence limit of ratio
- LCLM, one-sided lower confidence limit of ratio

The procedure calculates these statistics based on the *statistic-keywords* that you specified in the PROC SURVEYMEANS statement. If a *statistic-keyword* is not appropriate for a RATIO statement, that *statistic-keyword* is ignored. If no valid statistics are requested for a RATIO statement, the procedure computes the ratio and its standard error by default.

Note that ratios within a domain are currently not available.

When the means or proportions for the numerator and denominator variables in a ratio are calculated, an observation is excluded if it has a missing value in either the continuous numerator variable or the denominator variable. An observation with missing values is also excluded for the categorical numerator or denominator variables, unless the MISSING option is used.

When the denominator for a ratio is zero, then the value of the ratio is displayed as '–Infty', 'Infty', or a missing value, depending on whether the numerator is negative, positive, or zero, respectively; and the corresponding internal value is the special missing value '.M', the special missing value '.I', or the usual missing value, respectively.

REPWEIGHTS Statement

REPWEIGHTS *variables* < / *options* > **;**

The REPWEIGHTS statement names variables that provide replicate weights for BRR or jackknife variance estimation, which you request with the VARMETHOD=BRR or VARMETHOD=JACKKNIFE option in the PROC SURVEYMEANS statement. If you do not provide replicate weights for these methods by using a REPWEIGHTS statement, then the procedure constructs replicate weights for the analysis. See the sections "Balanced Repeated Replication (BRR) Method" on page 6506 and "Jackknife Method" on page 6508 for information about replicate weights.

Each REPWEIGHTS variable should contain the weights for a single replicate, and the number of replicates equals the number of REPWEIGHTS variables. The REPWEIGHTS variables must be numeric, and the variable values must be nonnegative numbers.

If you provide replicate weights with a REPWEIGHTS statement, you do not need to specify a CLUSTER or STRATA statement. If you use a REPWEIGHTS statement and do not specify the VARMETHOD= option in the PROC SURVEYMEANS statement, the procedure uses VARMETHOD=JACKKNIFE by default.

If you specify a REPWEIGHTS statement but do not include a WEIGHT statement, the procedure uses the average of each observation's replicate weights as the observation's weight.

You can specify the following options in the REPWEIGHTS statement after a slash (/):

DF=*df*

specifies the degrees of freedom for the analysis. The value of *df* must be a positive number. By default, the degrees of freedom equals the number of REPWEIGHTS variables. See the section "Degrees of Freedom" on page 6498 for details.

PROC SURVEYMEANS uses the DF= value in computing confidence limits for proportions, totals, and other statistics. See the section "Statistical Computations" on page 6493 for details.

JKCOEFS=*value*

specifies a jackknife coefficient for VARMETHOD=JACKKNIFE. The coefficient *value* must

be a nonnegative number less than one. See the section "Jackknife Method" on page 6508 for details about jackknife coefficients.

You can use this option to specify a single value of the jackknife coefficient, which the procedure uses for all replicates. To specify different coefficients for different replicates, use the JKCOEFS=*values* or JKCOEFS=*SAS-data-set* option.

JKCOEFS=*values*
specifies jackknife coefficients for VARMETHOD=JACKKNIFE, where each coefficient corresponds to an individual replicate identified by a REPWEIGHTS variable. You can separate *values* with blanks or commas. The coefficient *values* must be nonnegative numbers less than one. The number of *values* must equal the number of replicate weight variables named in the REPWEIGHTS statement. List these values in the same order in which you list the corresponding replicate weight variables in the REPWEIGHTS statement.

See the section "Jackknife Method" on page 6508 for details about jackknife coefficients.

To specify different coefficients for different replicates, you can also use the JKCOEFS=*SAS-data-set* option. To specify a single jackknife coefficient for all replicates, use the JKCOEFS=*value* option.

JKCOEFS=*SAS-data-set*
names a SAS data set that contains the jackknife coefficients for VARMETHOD=JACKKNIFE. You provide the jackknife coefficients in the JKCOEFS= data set variable JKCoefficient. Each coefficient value must be a nonnegative number less than one. The observations in the JKCOEFS= data set should correspond to the replicates that are identified by the REPWEIGHTS variables. Arrange the coefficients or observations in the JKCOEFS= data set in the same order in which you list the corresponding replicate weight variables in the REPWEIGHTS statement. The number of observations in the JKCOEFS= data set must not be less than the number of REPWEIGHTS variables.

See the section "Jackknife Method" on page 6508 for details about jackknife coefficients.

To specify different coefficients for different replicates, you can also use the JKCOEFS=*values* option. To specify a single jackknife coefficient for all replicates, use the JKCOEFS=*value* option.

STRATA Statement

STRATA *variables* < / *option* > **;**

The STRATA statement names variables that form the strata in a stratified sample design. The combinations of categories of STRATA variables define the strata in the sample.

If your sample design has stratification at multiple stages, you should identify only the first-stage strata in the STRATA statement. See the section "Survey Data Analysis" on page 6492 for more information.

If you provide replicate weights for BRR or jackknife variance estimation with the REPWEIGHTS statement, you do not need to specify a STRATA statement.

The STRATA *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric. The formatted values of the STRATA variables determine the levels. Thus, you can use formats to group values into levels. See the FORMAT procedure in the *Base SAS Procedures Guide* and the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary* for more information.

You can specify the following option in the STRATA statement after a slash (/):

LIST

displays a "Stratum Information" table, which includes values of the STRATA variables and sampling rates for each stratum. This table also provides the number of observations and number of clusters for each stratum and analysis variable. See the section "Displayed Output" on page 6512 for more details.

VAR Statement

VAR *variables* **;**

The VAR statement names the variables to be analyzed.

If you want a categorical analysis for a numeric variable, you must also name that variable in the CLASS statement. For categorical variables, PROC SURVEYMEANS estimates the proportion in each category or level, instead of the overall mean. Character variables are always analyzed as categorical variables. See the section "CLASS Statement" on page 6484 for more information.

When you specify a variable in a RATIO statement, but not in a VAR statement, the procedure includes this variable as an analysis variable.

If you do not specify a VAR statement, then PROC SURVEYMEANS analyzes all variables in the DATA= input data set, except those named in the BY, CLUSTER, REPWEIGHT, STRATA, and WEIGHT statements.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement names the variable that contains the sampling weights. This variable must be numeric, and the sampling weights must be positive numbers. If an observation has a weight that is nonpositive or missing, then the procedure omits that observation from the analysis. See the section "Missing Values" on page 6491 for more information. If you specify more than one WEIGHT statement, the procedure uses only the first WEIGHT statement and ignores the rest.

If you do not specify a WEIGHT statement but provide replicate weights with a REPWEIGHTS statement, PROC SURVEYMEANS uses the average of each observation's replicate weights as the observation's weight.

If you do not specify a WEIGHT statement or a REPWEIGHTS statement, PROC SURVEYMEANS assigns all observations a weight of one.

Details: SURVEYMEANS Procedure

Missing Values

If you have missing values in your survey data for any reason, such as nonresponse, this can compromise the quality of your survey results. If the respondents are different from the nonrespondents with regard to a survey effect or outcome, then survey estimates might be biased and cannot accurately represent the survey population. There are a variety of techniques in sample design and survey operations that can reduce nonresponse. After data collection is complete, you can use imputation to replace missing values with acceptable values, and/or you can use sampling weight adjustments to compensate for nonresponse. You should complete this data preparation and adjustment before you analyze your data with PROC SURVEYMEANS. See Cochran (1977); Kalton and Kaspyzyk (1986); and Brick and Kalton (1996) for more information.

If an observation has a missing value or a nonpositive value for the WEIGHT variable, then that observation is excluded from the analysis.

An observation is also excluded if it has a missing value for design variables such as STRATA variables, CLUSTER variables, and DOMAIN variables, unless missing values are regarded as a legitimate categorical level for these variables, as specified by the MISSING option.

If you specify the MISSING option in the PROC SURVEYMEANS statement, the procedure treats missing values of a categorical variable as a valid category.

By default, when computing statistics for an analysis variable, PROC SURVEYMEANS omits observations with missing values for that variable. The procedure computes statistics for each variable based only on observations that have nonmissing values for that variable. This treatment is based on the assumption that the missing values are missing completely at random (MCAR). However, this assumption is sometimes not true. For example, evidence from other surveys might suggest that observations with missing values are systematically different from observations without missing values. If you believe that missing values are not missing completely at random, then you can specify the NOMCAR option to let variance estimation include these observations with missing values in the analysis variables.

Whether or not the NOMCAR option is used, observations with missing or invalid values for WEIGHT, STRATA, CLUSTER, or DOMAIN variables are always excluded, unless the MISSING option is also specified.

When you specify the NOMCAR option, the procedure treats observations with and without missing values for analysis variables as two different domains, and it performs a domain analysis in the domain of nonmissing observations.

The procedure performs univariate analysis and analyzes each VAR variable separately. Thus, the number of missing observations might be different for different variables. You can specify the keyword NMISS in the PROC SURVEYMEANS statement to display the number of missing values for each analysis variable in the "Statistics" table.

If you use a REPWEIGHTS statement, all REPWEIGHTS variables must contain nonmissing values.

Survey Data Analysis

Specification of Population Totals and Sampling Rates

To include a finite population correction (*fpc*) in Taylor series variance estimation, you can input either the sampling rate or the population total by using the RATE= or TOTAL= option in the PROC SURVEYMEANS statement. (You cannot specify both of these options in the same PROC SURVEYMEANS statement.) The RATE= and TOTAL= options apply only to Taylor series variance estimation. The procedure does not use a finite population correction for BRR or jackknife variance estimation.

If you do not specify the RATE= or TOTAL= option, the Taylor series variance estimation does not include a finite population correction. For fairly small sampling fractions, it is appropriate to ignore this correction. See Cochran (1977) and Kish (1965) for more information.

If your design has multiple stages of selection and you are specifying the RATE= option, you should input the first-stage sampling rate, which is the ratio of the number of PSUs in the sample to the total number of PSUs in the study population. If you are specifying the TOTAL= option for a multistage design, you should input the total number of PSUs in the study population. See the section "Primary Sampling Units (PSUs)" on page 6493 for more details.

For a nonstratified sample design, or for a stratified sample design with the same sampling rate or the same population total in all strata, you can use the RATE=*value* or TOTAL=*value* option. If your sample design is stratified with different sampling rates or population totals in different strata, use the RATE=*SAS-data-set* or TOTAL=*SAS-data-set* option to name a SAS data set that contains the stratum sampling rates or totals. This data set is called a *secondary data set*, as opposed to the *primary data set* that you specify with the DATA= option.

The secondary data set must contain all the stratification variables listed in the STRATA statement and all the variables in the BY statement. If there are formats associated with the STRATA variables and the BY variables, then the formats must be consistent in the primary and the secondary data sets. If you specify the TOTAL=*SAS-data-set* option, the secondary data set must have a variable named _TOTAL_ that contains the stratum population totals. Or if you specify the RATE=*SAS-data-set* option, the secondary data set must have a variable named _RATE_ that contains the stratum sampling rates. If the secondary data set contains more than one observation for any one stratum, then the procedure uses the first value of _TOTAL_ or _RATE_ for that stratum and ignores the rest.

The *value* in the RATE= option or the values of _RATE_ in the secondary data set must be nonnegative numbers. You can specify *value* as a number between 0 and 1. Or you can specify *value*

in percentage form as a number between 1 and 100, and PROC SURVEYMEANS converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

If you specify the TOTAL=*value* option, *value* must not be less than the sample size. If you provide stratum population totals in a secondary data set, these values must not be less than the corresponding stratum sample sizes.

Primary Sampling Units (PSUs)

When you have clusters, or primary sampling units (PSUs), in your sample design, the procedure estimates variance from the variation among PSUs when the Taylor series variance method is used. See the section "Variance and Standard Error of the Mean" on page 6496 and the section "Variance and Standard Deviation of the Total" on page 6501 for more information.

BRR or jackknife variance estimation methods draw multiple replicates (or subsamples) from the full sample by following a specific resampling scheme. These subsamples are constructed by deleting PSUs from the full sample.

If you use a REPWEIGHTS statement to provide replicate weights for BRR or jackknife variance estimation, you do not need to specify a CLUSTER statement. Otherwise, you should specify a CLUSTER statement whenever your design includes clustering at the first stage of sampling. If you do not specify a CLUSTER statement, then PROC SURVEYMEANS treats each observation as a PSU.

Domain Analysis

It is common practice to compute statistics for subpopulations, or domains, in addition to computing statistics for the entire study population. Analysis for domains that uses the entire sample is called *domain analysis* (subgroup analysis, subpopulation analysis, subdomain analysis). The formation of these subpopulations of interest might be unrelated to the sample design. Therefore, the sample sizes for the subpopulations might actually be random variables.

In order to incorporate this variability into the variance estimation, you should use a DOMAIN statement. Note that using a BY statement provides completely separate analyses of the BY groups. It does not provide a statistically valid subpopulation or domain analysis, where the total number of units in the subpopulation is not known with certainty.

For more detailed information about domain analysis, see Kish (1965).

Statistical Computations

The SURVEYMEANS procedure uses the Taylor series (linearization) method or replication (resampling) methods to estimate sampling errors of estimators based on complex sample designs. For details see Wolter (1985); Lohr (1999); Kalton (1983); Hidiroglou, Fuller, and Hickman (1980); Fuller et al. (1989); Lee, Forthoffer, and Lorimor (1989); Cochran (1977); Kish (1965); Hansen,

Hurwitz, and Madow (1953); Rust (1985); Dippo, Fay, and Morganstein (1984); Rao and Shao (1999); Rao, Wu, and Yue (1992); and Rao and Shao (1996). You can use the VARMETHOD= option to specify a variance estimation method to use. By default, the Taylor series method is used.

The Taylor series method obtains a linear approximation for the estimator and then uses the variance estimate for this approximation to estimate the variance of the estimate itself (Woodruff 1971, Fuller 1975). When there are clusters, or PSUs, in the sample design, the procedure estimates variance from the variation among PSUs. When the design is stratified, the procedure pools stratum variance estimates to compute the overall variance estimate. For t tests of the estimates, the degrees of freedom equal the number of clusters minus the number of strata in the sample design.

For a multistage sample design, the Taylor series estimation depends only on the first stage of the sample design. Therefore, the required input includes only first-stage cluster (PSU) and first-stage stratum identification. You do not need to input design information about any additional stages of sampling. This variance estimation method assumes that the first-stage sampling fraction is small, or that the first-stage sample is drawn with replacement, as it often is in practice.

Quite often in complex surveys, respondents have unequal weights, which reflect unequal selection probabilities and adjustments for nonresponse. In such surveys, the appropriate sampling weights must be used to obtain valid estimates for the study population.

However, replication methods have recently gained popularity for estimating variances in complex survey data analysis. One reason for this popularity is the relative simplicity of replication-based estimates, especially for nonlinear estimators; another is that modern computational capacity has made replication methods feasible for practical survey analysis.

Replication methods draw multiple replicates (also called subsamples) from a full sample according to a specific resampling scheme. The most commonly used resampling schemes are the *balanced repeated replication* (BRR) method and the *jackknife* method. For each replicate, the original weights are modified for the PSUs in the replicates to create replicate weights. The population parameters of interest are estimated by using the replicate weights for each replicate. Then the variances of parameters of interest are estimated by the variability among the estimates derived from these replicates. You can use a REPWEIGHTS statement to provide your own replicate weights for variance estimation. For more information about using replication methods to analyze sample survey data, see the section "Replication Methods for Variance Estimation" on page 6506.

Definitions and Notation

For a stratified clustered sample design, together with the sampling weights, the sample can be represented by an $n \times (P + 1)$ matrix

$$\begin{aligned}(\mathbf{w}, \mathbf{Y}) &= \left(w_{hij}, \mathbf{y}_{hij}\right) \\ &= \left(w_{hij}, y_{hij}^{(1)}, y_{hij}^{(2)}, \ldots, y_{hij}^{(P)}\right)\end{aligned}$$

where

- $h = 1, 2, \ldots, H$ is the stratum index

- $i = 1, 2, \ldots, n_h$ is the cluster index within stratum h
- $j = 1, 2, \ldots, m_{hi}$ is the unit index within cluster i of stratum h
- $p = 1, 2, \ldots, P$ is the analysis variable number, with a total of P variables
- $n = \sum_{h=1}^{H} \sum_{i=1}^{n_h} m_{hi}$ is the total number of observations in the sample
- w_{hij} denotes the sampling weight for unit j in cluster i of stratum h
- $\mathbf{y}_{hij} = \left(y_{hij}^{(1)}, y_{hij}^{(2)}, \ldots, y_{hij}^{(P)}\right)$ are the observed values of the analysis variables for unit j in cluster i of stratum h, including both the values of numerical variables and the values of indicator variables for levels of categorical variables.

For a categorical variable C, let l denote the number of levels of C, and denote the level values as $c_1, c_2, \ldots, c_l$. Let $y^{(q)}$ ($q \in \{1, 2, \ldots, P\}$) be an indicator variable for the category $C = c_k$ ($k = 1, 2, \ldots, l$) with the observed value in unit j in cluster i of stratum h:

$$y_{hij}^{(q)} = I_{\{C=c_k\}}(h, i, j) = \begin{cases} 1 & \text{if } C_{hij} = c_k \\ 0 & \text{otherwise} \end{cases}$$

Note that the indicator variable $y_{hij}^{(q)}$ is set to missing when C_{hij} is missing. Therefore, the total number of analysis variables, P, is the total number of numerical variables plus the total number of levels of all categorical variables.

The sampling rate f_h for stratum h, which is used in Taylor series variance estimation, is the fraction of first-stage units (PSUs) selected for the sample. You can use the TOTAL= or RATE= option to input population totals or sampling rates. See the section "Specification of Population Totals and Sampling Rates" on page 6492 for details. If you input stratum totals, PROC SURVEYMEANS computes f_h as the ratio of the stratum sample size to the stratum total. If you input stratum sampling rates, PROC SURVEYMEANS uses these values directly for f_h. If you do not specify the TOTAL= or RATE= option, then the procedure assumes that the stratum sampling rates f_h are negligible, and a finite population correction is not used when computing variances. Replication methods specified by the VARMETHOD=BRR or the VARMETHOD=JACKKNIFE option do not use this finite population correction f_h.

Mean

When you specify the keyword MEAN, the procedure computes the estimate of the mean (mean per element) from the survey data. Also, the procedure computes the mean by default if you do not specify any *statistic-keywords* in the PROC SURVEYMEANS statement.

PROC SURVEYMEANS computes the estimate of the mean as

$$\widehat{\bar{Y}} = \left(\sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} w_{hij}\, y_{hij} \right) / \, w_{...}$$

where

$$w_{...} = \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} w_{hij}$$

is the sum of the weights over all observations in the sample.

Variance and Standard Error of the Mean

When you specify the keyword STDERR, the procedure computes the standard error of the mean. Also, the procedure computes the standard error by default if you specify the keyword MEAN, or if you do not specify any *statistic-keywords* in the PROC SURVEYMEANS statement. The keyword VAR requests the variance of the mean.

Taylor Series Method

When you use VARMETHOD=TAYLOR, or by default if you do not specify the VARMETHOD= option, PROC SURVEYMEANS uses the Taylor series method to estimate the variance of the mean $\widehat{\bar{Y}}$. The procedure computes the estimated variance as

$$\widehat{V}(\widehat{\bar{Y}}) = \sum_{h=1}^{H} \widehat{V_h}(\widehat{\bar{Y}})$$

where if $n_h > 1$,

$$\begin{aligned}
\widehat{V_h}(\widehat{\bar{Y}}) &= \frac{n_h(1-f_h)}{n_h - 1} \sum_{i=1}^{n_h} (e_{hi\cdot} - \bar{e}_{h\cdot\cdot})^2 \\
e_{hi\cdot} &= \left(\sum_{j=1}^{m_{hi}} w_{hij} \, (y_{hij} - \widehat{\bar{Y}}) \right) / \, w_{...} \\
\bar{e}_{h\cdot\cdot} &= \left(\sum_{i=1}^{n_h} e_{hi\cdot} \right) / \, n_h
\end{aligned}$$

and if $n_h = 1$,

$$\widehat{V_h}(\widehat{\bar{Y}}) = \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 \le h' \le H \end{cases}$$

Replication Methods

When you specify VARMETHOD=BRR or VARMETHOD=JACKKNIFE, the procedure computes the variance $\widehat{V}(\widehat{\bar{Y}})$ with replication methods by using the variability among replicate estimates to estimate the overall variance. See the section "Replication Methods for Variance Estimation" on page 6506 for more details.

Standard Error

The standard error of the mean is the square root of the estimated variance.

$$\text{StdErr}(\widehat{\bar{Y}}) = \sqrt{\widehat{V}(\widehat{\bar{Y}})}$$

Ratio

When you use a RATIO statement, the procedure produces statistics requested by the *statistic-keywords* in the PROC SURVEYMEANS statement.

Suppose that you want to calculate the ratio of variable Y over variable X. Let x_{hij} be the value of variable X for the jth member in cluster i in the hth stratum.

The ratio of Y over X is

$$\widehat{R} = \frac{\sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}\, y_{hij}}{\sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}\, x_{hij}}$$

PROC SURVEYMEANS uses the Taylor series method to estimate the variance of the ratio $\widehat{R}$ as

$$\widehat{V}(\widehat{R}) = \sum_{h=1}^{H} \widehat{V_h}(\widehat{R})$$

where if $n_h > 1$,

$$\begin{aligned}
\widehat{V_h}(\widehat{R}) &= \frac{n_h(1-f_h)}{n_h - 1} \sum_{i=1}^{n_h} (g_{hi\cdot} - \bar{g}_{h\cdot\cdot})^2 \\
g_{hi\cdot} &= \frac{\sum_{j=1}^{m_{hi}} w_{hij}\,(y_{hij} - x_{hij}\widehat{R})}{\sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}\, x_{hij}} \\
\bar{g}_{h\cdot\cdot} &= \left(\sum_{i=1}^{n_h} g_{hi\cdot}\right) / n_h
\end{aligned}$$

and if $n_h = 1$,

$$\widehat{V_h}(\widehat{R}) = \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 \le h' \le H \end{cases}$$

The standard error of the ratio is the square root of the estimated variance:

$$\text{StdErr}(\widehat{R}) = \sqrt{\widehat{V}(\widehat{R})}$$

When the denominator for a ratio is zero, then the value of the ratio is displayed as '–Infty', 'Infty', or a missing value, depending on whether the numerator is negative, positive, or zero, respectively; and the corresponding internal value is the special missing value '.M', the special missing value '.I', or the usual missing value, respectively.

t Test for the Mean

If you specify the keyword T, PROC SURVEYMEANS computes the t-value for testing that the population mean equals zero, $H_0 : \bar{Y} = 0$. The test statistic equals

$$t(\widehat{\bar{Y}}) = \widehat{\bar{Y}} \;/\; \mathrm{StdErr}(\widehat{\bar{Y}})$$

The two-sided p-value for this test is

$$\mathrm{Prob}(\,|T| > |t(\widehat{\bar{Y}})|\,)$$

where T is a random variable with the t distribution with df degrees of freedom.

Degrees of Freedom

PROC SURVEYMEANS computes degrees of freedom df to obtain the $100(1-\alpha)\%$ confidence limits for means, proportions, totals, ratios, and other statistics. The degrees of freedom computation depends on the variance estimation method that you request. Missing values can affect the degrees of freedom computation. See the section "Missing Values" on page 6491 for details.

Taylor Series Variance Estimation

For the Taylor series method, PROC SURVEYMEANS calculates the degrees of freedom for the t test as the number of clusters minus the number of strata. If there are no clusters, then the degrees of freedom equal the number of observations minus the number of strata. If the design is not stratified, then the degrees of freedom equal the number of PSUs minus one.

If in a stratum, all observations are excluded from the analysis due to missing values, then that stratum is called an *empty stratum*. Empty strata are not counted in the total number of strata for the table. Similarly, empty clusters and missing observations are not included in the totals counts of cluster and observations used to compute the degrees of freedom for the analysis.

If you specify the MISSING option, missing values are treated as valid nonmissing levels for a categorical variable and are included in computing degrees of freedom. If you specify the NOMCAR option for Taylor series variance estimation, observations with missing values for an analysis variable are included in computing degrees of freedom.

Replicate-Based Variance Estimation

When there is a REPWEIGHTS statement, the degrees of freedom equal the number of REPWEIGHTS variables, unless you specify an alternative in the DF= option in a REPWEIGHTS statement.

For BRR or jackknife variance estimation without a REPWEIGHT statement, by default PROC SURVEYMEANS computes the degrees of freedom by using all valid observations in the input data set. A valid observation is an observation that has a positive value of the WEIGHT variable and nonmissing values of the STRATA and CLUSTER variables unless you specify the MISSING option. See the section "Data and Sample Design Summary" on page 6512 for details about valid observations.

For BRR variance estimation (including Fay's method) without a REPWEIGHTS statement, PROC SURVEYMEANS calculates the degrees of freedom as the number of strata. PROC SURVEYMEANS bases the number of strata on all valid observations in the data set, unless you specify the DFADJ method-option for VARMETHOD=BRR. When you specify the DFADJ option, the procedure computes the degrees of freedom as the number of nonmissing strata for an analysis variable. This excludes any empty strata that occur when observations with missing values of that analysis variable are removed.

For jackknife variance estimation without a REPWEIGHTS statement, PROC SURVEYMEANS calculates the degrees of freedom as the number of clusters minus the number of strata. If there are no clusters, then df equals the number of observations minus the number of strata. If the design is not stratified, then df equals the number of clusters minus one. For jackknife variance estimation, PROC SURVEYMEANS bases the number of strata and clusters on all valid observations in the data set, unless you specify the DFADJ method-option for VARMETHOD=JK. When you specify the DFADJ option, the procedure computes the degrees of freedom from the number of nonmissing strata and clusters for an analysis variable. This excludes any empty strata or clusters that occur when observations with missing values of an analysis variable are removed.

The procedure displays the degrees of freedom for the t test if you specify the keyword DF in the PROC SURVEYMEANS statement.

Confidence Limits for the Mean

If you specify the keyword CLM, the procedure computes two-sided confidence limits for the mean. Also, the procedure includes the confidence limits by default if you do not specify any *statistic-keywords* in the PROC SURVEYMEANS statement.

The confidence coefficient is determined by the value of the ALPHA= option, which by default equals 0.05 and produces 95% confidence limits. The confidence limits are computed as

$$\hat{\bar{Y}} \pm \mathrm{StdErr}(\hat{\bar{Y}}) \cdot t_{df,\,\alpha/2}$$

where $\hat{\bar{Y}}$ is the estimate of the mean, $\mathrm{StdErr}(\hat{\bar{Y}})$ is the standard error of the mean, and $t_{df,\,\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the t distribution with df calculated as described in the section "*t* Test for the Mean" on page 6498.

If you specify the keyword UCLM, the procedure computes the one-sided upper $100(1-\alpha)\%$ confidence limit for the mean:

$$\hat{\bar{Y}} + \mathrm{StdErr}(\hat{\bar{Y}}) \cdot t_{df,\,\alpha}$$

If you specify the keyword LCLM, the procedure computes the one-sided lower $100(1-\alpha)\%$ confidence limit for the mean:

$$\hat{\bar{Y}} - \text{StdErr}(\hat{\bar{Y}}) \cdot t_{df,\,\alpha}$$

Coefficient of Variation

If you specify the keyword CV, PROC SURVEYMEANS computes the coefficient of variation, which is the ratio of the standard error of the mean to the estimated mean:

$$cv(\bar{Y}) = \text{StdErr}(\hat{\bar{Y}}) \;/\; \hat{\bar{Y}}$$

If you specify the keyword CVSUM, PROC SURVEYMEANS computes the coefficient of variation for the estimated total, which is the ratio of the standard deviation of the sum to the estimated total:

$$cv(Y) = \text{Std}(\hat{Y}) \;/\; \hat{Y}$$

Proportions

If you specify the keyword MEAN for a categorical variable, PROC SURVEYMEANS estimates the proportion, or relative frequency, for each level of the categorical variable. If you do not specify any *statistic-keywords* in the PROC SURVEYMEANS statement, the procedure estimates the proportions for levels of the categorical variables, together with their standard errors and confidence limits.

The procedure estimates the proportion in level c_k for variable C as

$$\hat{p} = \frac{\sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}\; y_{hij}^{(q)}}{\sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}}$$

where $y_{hij}^{(q)}$ is the value of the indicator function for level $C = c_k$, defined in the section "Definitions and Notation" on page 6494, and $y_{hij}^{(q)}$ equals 1 if the observed value of variable C equals c_k, and $y_{hij}^{(q)}$ equals 0 otherwise. Since the proportion estimator is actually an estimator of the mean for an indicator variable, the procedure computes its variance and standard error according to the method outlined in the section "Variance and Standard Error of the Mean" on page 6496. Similarly, the procedure computes confidence limits for proportions as described in the section "Confidence Limits for the Mean" on page 6499.

Total

If you specify the keyword SUM, the procedure computes the estimate of the population total from the survey data. The estimate of the total is the weighted sum over the sample:

$$\hat{Y} = \sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}\; y_{hij}$$

For a categorical variable level, $\hat{Y}$ estimates its total frequency in the population.

Variance and Standard Deviation of the Total

When you specify the keyword STD or the keyword SUM, the procedure estimates the standard deviation of the total. The keyword VARSUM requests the variance of the total.

Taylor Series Method

When you use VARMETHOD=TAYLOR, or by default, PROC SURVEYMEANS uses the Taylor series method to estimate the variance of the total as

$$\widehat{V}(\widehat{Y}) = \sum_{h=1}^{H} \widehat{V_h}(\widehat{Y})$$

where if $n_h > 1$,

$$\begin{aligned}
\widehat{V_h}(\widehat{Y}) &= \frac{n_h(1-f_h)}{n_h-1} \sum_{i=1}^{n_h} (y_{hi\cdot} - \bar{y}_{h\cdot\cdot})^2 \\
y_{hi\cdot} &= \sum_{j=1}^{m_{hi}} w_{hij}\, y_{hij} \\
\bar{y}_{h\cdot\cdot} &= \left(\sum_{i=1}^{n_h} y_{hi\cdot} \right) / n_h
\end{aligned}$$

and if $n_h = 1$,

$$\widehat{V_h}(\widehat{Y}) = \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 \le h' \le H \end{cases}$$

Replication Methods

When you specify VARMETHOD=BRR or VARMETHOD=JACKKNIFE option, the procedure computes the variance $\widehat{V}(\widehat{Y})$ with replication methods by measuring the variability among the estimates derived from these replicates. See the section "Replication Methods for Variance Estimation" on page 6506 for more details.

Standard Deviation

The standard deviation of the total equals

$$\text{Std}(\widehat{Y}) = \sqrt{\widehat{V}(\widehat{Y})}$$

Confidence Limits for the Total

If you specify the keyword CLSUM, the procedure computes confidence limits for the total. The confidence coefficient is determined by the value of the ALPHA= option, which by default equals

0.05 and produces 95% confidence limits. The confidence limits are computed as

$$\widehat{Y} \pm \mathrm{Std}(\widehat{Y}) \cdot t_{df,\,\alpha/2}$$

where $\widehat{Y}$ is the estimate of the total, $\mathrm{Std}(\widehat{Y})$ is the estimated standard deviation, and $t_{df,\,\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the t distribution with df calculated as described in the section "*t* Test for the Mean" on page 6498.

If you specify the keyword UCLSUM, the procedure computes the one-sided upper $100(1-\alpha)\%$ confidence limit for the sum:

$$\widehat{Y} + \mathrm{Std}(\widehat{Y}) \cdot t_{df,\,\alpha}$$

If you specify the keyword LCLSUM, the procedure computes the one-sided lower $100(1-\alpha)\%$ confidence limit for the sum:

$$\widehat{Y} - \mathrm{Std}(\widehat{Y}) \cdot t_{df,\,\alpha}$$

Domain Statistics

When you use a DOMAIN statement to request a domain analysis, the procedure computes the requested statistics for each domain.

For a domain D, let I_D be the corresponding indicator variable:

$$I_D(h,i,j) = \begin{cases} 1 & \text{if observation } (h,i,j) \text{ belongs to } D \\ 0 & \text{otherwise} \end{cases}$$

Let

$$v_{hij} = w_{hij} I_D(h,i,j) = \begin{cases} w_{hij} & \text{if observation } (h,i,j) \text{ belongs to } D \\ 0 & \text{otherwise} \end{cases}$$

The requested statistics for variable y in domain D are computed by using the new weights v.

Domain Mean

The estimated mean of y in the domain D is

$$\widehat{\bar{Y}_D} = \left(\sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} v_{hij}\, y_{hij} \right) / \, v_{...}$$

where

$$v_{...} = \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} v_{hij}$$

The variance of $\widehat{\bar{Y}_D}$ is estimated by

$$\widehat{V}(\widehat{\bar{Y}_D}) = \sum_{h=1}^{H} \widehat{V_h}(\widehat{\bar{Y}_D})$$

where if $n_h > 1$,

$$\begin{aligned}
\widehat{V_h}(\widehat{\bar{Y}_D}) &= \frac{n_h(1-f_h)}{n_h - 1} \sum_{i=1}^{n_h} (r_{hi\cdot} - \bar{r}_{h\cdot\cdot})^2 \\
r_{hi\cdot} &= \left(\sum_{j=1}^{m_{hi}} v_{hij} \ (y_{hij} - \widehat{\bar{Y}_D}) \right) / \ v_{\cdots} \\
\bar{r}_{h\cdot\cdot} &= \left(\sum_{i=1}^{n_h} r_{hi\cdot} \right) / \ n_h
\end{aligned}$$

and if $n_h = 1$,

$$\widehat{V_h}(\widehat{\bar{Y}_D}) = \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 \le h' \le H \end{cases}$$

Domain Total

The estimated total in domain D is

$$\widehat{Y}_D = \sum_{h=1}^{H} \sum_{i=1}^{n_h} \sum_{j=1}^{m_{hi}} v_{hij} \ y_{hij}$$

and its estimated variance is

$$\widehat{V}(\widehat{Y}_D) = \sum_{h=1}^{H} \widehat{V_h}(\widehat{Y}_D)$$

where if $n_h > 1$,

$$\begin{aligned}
\widehat{V_h}(\widehat{Y}_D) &= \frac{n_h(1-f_h)}{n_h - 1} \sum_{i=1}^{n_h} (z_{hi\cdot} - \bar{z}_{h\cdot\cdot})^2 \\
z_{hi\cdot} &= \sum_{j=1}^{m_{hi}} v_{hij} \ z_{hij} \\
\bar{z}_{h\cdot\cdot} &= \left(\sum_{i=1}^{n_h} z_{hi\cdot} \right) \ / \ n_h
\end{aligned}$$

and if $n_h = 1$,

$$\widehat{V_h}(\widehat{Y}_D) = \begin{cases} \text{missing} & \text{if } n_{h'} = 1 \text{ for } h' = 1, 2, \ldots, H \\ 0 & \text{if } n_{h'} > 1 \text{ for some } 1 \le h' \le H \end{cases}$$

Quantiles

Let Y be the variable of interest in a complex survey. Denote $F(t) = Pr(Y \leq t)$ as the cumulative distribution for Y.

For $0 < p < 1$, the pth quantile of the population cumulative distribution function is

$$Y_p = \inf\{y : F(y) \geq p\}$$

Estimate of Quantile

Let $\{y_{hij}, w_{hij}\}$ be the observed values for variable Y associated with sampling weights, where (h, i, j) are the stratum index, cluster index, and member index, respectively, as shown in the section "Definitions and Notation" on page 6494. Let $y_{(1)} < y_{(2)} < \ldots < y_{(d)}$ denote the sample order statistics for variable Y.

An estimate of quantile Y_p is

$$\hat{Y}_p = \begin{cases} y_{(1)} & \text{if } p < \hat{F}(y_{(1)}) \\ y_{(k)} + \dfrac{p - \hat{F}(y_{(k)})}{\hat{F}(y_{(k+1)}) - \hat{F}(y_{(k)})}(y_{(k+1)} - y_{(k)}) & \text{if } \hat{F}(y_{(k)}) \leq p < \hat{F}(y_{(k+1)}) \\ y_{(d)} & \text{if } p = 1 \end{cases}$$

where $\hat{F}(t)$ is the estimated cumulative distribution for Y:

$$\hat{F}(t) = \frac{\sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij} I(y_{hij} \leq t)}{\sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}}$$

and $I(\cdot)$ is the indicator function.

Standard Error

PROC SURVEYMEANS uses Woodruff method (Dorfman and Valliant 1993; Särndal, Swensson, and Wretman 1992; Francisco and Fuller 1991) to estimate the variances of quantiles. This method first constructs a confidence interval on a quantile. Then it uses the width of the confidence interval to estimate the standard error of a quantile.

In order to estimate the variance for $\hat{Y}_p$, first the procedure estimates the variance of the estimated distribution function $\hat{F}(\hat{Y}_p)$ by

$$\hat{V}(\hat{F}(\hat{Y}_p)) = \sum_{h=1}^{H} \frac{n_h(1 - f_h)}{n_h - 1} \sum_{i=1}^{n_h} (d_{hi\cdot} - \bar{d}_{h\cdot\cdot})^2$$

where

$$d_{hi\cdot} = \left(\sum_{j=1}^{m_{hi}} w_{hij} \left(I(y_{hij} \leq \hat{Y}_p) - \hat{F}(\hat{Y}_p)\right) \right) / w_{\ldots}$$

$$\bar{d}_{h\cdot\cdot} = \left(\sum_{i=1}^{n_h} d_{hi\cdot} \right) / n_h$$

Then $100(1-\alpha)\%$ confidence limits of $\hat{F}(\hat{Y}_p)$ can be constructed by

$$(\hat{p}_L, \quad \hat{p}_U) = \left(\hat{F}(\hat{Y}_p) - t_{df,\,\alpha/2}\sqrt{\hat{V}(\hat{F}(\hat{Y}_p))}, \quad \hat{F}(\hat{Y}_p) + t_{df,\,\alpha/2}\sqrt{\hat{V}(\hat{F}(\hat{Y}_p))}\right)$$

where $t_{df,\,\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the t distribution with df degrees of freedom, described in the section "Degrees of Freedom" on page 6498.

When $(\hat{p}_L, \hat{p}_U)$ is out of the range of [0,1], the procedure does not compute the standard error.

The $\hat{p}_L$th quantile is defined as

$$\hat{Y}_{\hat{p}_L} = \begin{cases} y_{(1)} & \text{if } \hat{p}_L < \hat{F}(y_{(1)}) \\ y_{(k_L)} + \dfrac{\hat{p}_L - \hat{F}(y_{(k_L)})}{\hat{F}(y_{(k_L+1)}) - \hat{F}(y_{(k_L)})}(y_{(k_L+1)} - y_{(k_L)}) & \text{if } \hat{F}(y_{(k_L)}) \le \hat{p}_L < \hat{F}(y_{(k_L+1)}) \\ y_{(d)} & \text{if } \hat{p}_L = 1 \end{cases}$$

and the $\hat{p}_U$th quantile is defined as

$$\hat{Y}_{\hat{p}_U} = \begin{cases} y_{(1)} & \text{if } \hat{p}_U < \hat{F}(y_{(1)}) \\ y_{(k_U)} + \dfrac{\hat{p}_U - \hat{F}(y_{(k_U)})}{\hat{F}(y_{(k_U+1)}) - \hat{F}(y_{(k_U)})}(y_{(k_U+1)} - y_{(k_U)}) & \text{if } \hat{F}(y_{(k_U)}) \le \hat{p}_U < \hat{F}(y_{(k_U+1)}) \\ y_{(d)} & \text{if } \hat{p}_U = 1 \end{cases}$$

The standard error of $\hat{Y}_p$ then is estimated by

$$sd(\hat{Y}_p) = \frac{\hat{Y}_{\hat{p}_U} - \hat{Y}_{\hat{p}_L}}{2t_{df,\,\alpha/2}}$$

where $t_{df,\,\alpha/2}$ is the $100(1-\alpha/2)$th percentile of the t distribution with df degrees of freedom.

Confidence Limits

Symmetric $100(1-\alpha)\%$ confidence limits are computed as

$$\left(\hat{Y}_p - sd(\hat{Y}_p) \cdot t_{df,\,\alpha/2}, \quad \hat{Y}_p + sd(\hat{Y}_p) \cdot t_{df,\,\alpha/2}\right)$$

If you specify the NONSYMCL option in the PROC statement, the procedure computes $100(1-\alpha)\%$ nonsymmetric confidence limits:

$$\left(\hat{Y}_{\hat{p}_L}, \quad \hat{Y}_{\hat{p}_U}\right)$$

Replication Methods for Variance Estimation

Recently replication methods have gained popularity for estimating variances in complex survey data analysis. One reason for this popularity is the relative simplicity of replication-based estimates, especially for nonlinear estimators; another is that modern computational capacity has made replication methods feasible for practical survey analysis. For details see Lohr (1999); Wolter (1985); Rust (1985); Dippo, Fay, and Morganstein (1984); Rao and Shao (1999); Rao, Wu, and Yue (1992); and Rao and Shao (1996).

Replication methods draw multiple replicates (also called subsamples) from a full sample according to a specific resampling scheme. The most commonly used resampling schemes are the *balanced repeated replication* (BRR) method and the *jackknife* method. For each replicate, the original weights are modified for the PSUs in the replicates to create replicate weights. The statistics of interest are estimated by using the replicate weights for each replicate. Then the variances of parameters of interest are estimated by the variability among the estimates derived from these replicates. You can use the REPWEIGHTS statement to provide your own replicate weights for variance estimation.

Balanced Repeated Replication (BRR) Method

The balanced repeated replication (BRR) method requires that the full sample be drawn by using a stratified sample design with two primary sampling units (PSUs) per stratum. Let H be the total number of strata. The total number of replicates R is the smallest multiple of 4 that is greater than H. However, if you prefer a larger number of replicates, you can specify the REPS=*number* option. If a $number \times number$ Hadamard matrix cannot be constructed, the number of replicates is increased until a Hadamard matrix becomes available.

Each replicate is obtained by deleting one PSU per stratum according to the corresponding Hadamard matrix and adjusting the original weights for the remaining PSUs. The new weights are called replicate weights.

Replicates are constructed by using the first H columns of the $R \times R$ Hadamard matrix. The rth ($r = 1, 2, ..., R$) replicate is drawn from the full sample according to the rth row of the Hadamard matrix as follows:

- If the (r, h)th element of the Hadamard matrix is 1, then the first PSU of stratum h is included in the rth replicate and the second PSU of stratum h is excluded.
- If the (r, h)th element of the Hadamard matrix is -1, then the second PSU of stratum h is included in the rth replicate and the first PSU of stratum h is excluded.

The replicate weights of the remaining PSUs in each half sample are then doubled to their original weights. For more details about the BRR method, see Wolter (1985) and Lohr (1999).

By default, an appropriate Hadamard matrix is generated automatically to create the replicates. You can request that the Hadamard matrix be displayed by specifying the VARMETHOD=BRR(PRINTH) method-option. If you provide a Hadamard matrix by specifying the VARMETHOD=BRR(HADAMARD=) method-option, then the replicates are generated according to the provided Hadamard matrix.

Suppose that θ is a population parameter of interest. Let $\hat{\theta}$ be the estimate from the full sample for θ. Let $\hat{\theta}_r$ be the estimate from the rth replicate subsample by using replicate weights. PROC SURVEYMEANS estimates the variance of $\hat{\theta}$ by

$$\widehat{V}(\hat{\theta}) = \frac{1}{R}\sum_{r=1}^{R}\left(\hat{\theta}_r - \hat{\theta}\right)^2$$

with H degrees of freedom, where H is the number of strata.

You can use the VARMETHOD=BRR(OUTWEIGHTS=) method-option to save the replicate weights into a SAS data set.

Fay's BRR Method

Fay's method is a modification of the BRR method, and it requires a stratified sample design with two primary sampling units (PSUs) per stratum. The total number of replicates R is the smallest multiple of 4 that is greater than the total number of strata H. However, if you prefer a larger number of replicates, you can specify the REPS= method-option.

For each replicate, Fay's method uses a Fay coefficient $0 \leq \epsilon < 1$ to impose a perturbation of the original weights in the full sample that is gentler than using only half samples, as in the traditional BRR method. The Fay coefficient $0 \leq \epsilon < 1$ can be optionally set by the FAY = ϵ method-option. By default, $\epsilon = 0.5$ if only the FAY method-option is used without specifying a value for ϵ (Judkins 1990; Rao and Shao 1999). When $\epsilon = 0$, Fay's method becomes the traditional BRR method. For more details, see Dippo, Fay, and Morganstein (1984), Fay (1984), Fay (1989), and Judkins (1990).

Let H be the number of strata. Replicates are constructed by using the first H columns of the $R \times R$ Hadamard matrix, where R is the number of replicates, $R > H$. The rth ($r = 1, 2, ..., R$) replicate is created from the full sample according to the rth row of the Hadamard matrix as follows:

- If the (r, h)th element of the Hadamard matrix is 1, then the full sample weight of the first PSU in stratum h is multiplied by ϵ and that of the second PSU is multiplied by $2-\epsilon$ to obtain the rth replicate weights.
- If the (r, h)th element of the Hadamard matrix is -1, then the full sample weight of the first PSU in stratum h is multiplied by $2-\epsilon$ and that of the second PSU is multiplied by ϵ to obtain the rth replicate weights.

You can use the VARMETHOD=BRR(OUTWEIGHTS=) method-option to save the replicate weights into a SAS data set.

By default, an appropriate Hadamard matrix is generated automatically to create the replicates. You can request that the Hadamard matrix be displayed by specifying the VARMETHOD=BRR(PRINTH) method-option. If you provide a Hadamard matrix by specifying the VARMETHOD=BRR(HADAMARD=) method-option, then the replicates are generated according to the provided Hadamard matrix.

Suppose that θ is a population parameter of interest. Let $\hat{\theta}$ be the estimate from the full sample for θ. Let $\hat{\theta}_r$ be the estimate from the rth replicate subsample by using replicate weights. PROC

SURVEYMEANS estimates the variance of $\hat{\theta}$ by

$$\widehat{V}(\hat{\theta}) = \frac{1}{R(1-\epsilon)^2} \sum_{r=1}^{R} \left(\hat{\theta}_r - \hat{\theta}\right)^2$$

with H degrees of freedom, where H is the number of strata.

Jackknife Method

The jackknife method of variance estimation deletes one PSU at a time from the full sample to create replicates. The total number of replicates R is the same as the total number of PSUs. In each replicate, the sample weights of the remaining PSUs are modified by the *jackknife coefficient* α_r. The modified weights are called replicate weights.

The jackknife coefficient and replicate weights are described as follows.

Without Stratification If there is no stratification in the sample design (no STRATA statement), the jackknife coefficients α_r are the same for all replicates:

$$\alpha_r = \frac{R-1}{R} \quad \text{where } r = 1, 2, ..., R$$

Denote the original weight in the full sample for the jth member of the ith PSU as w_{ij}. If the ith PSU is included in the rth replicate ($r = 1, 2, ..., R$), then the corresponding replicate weight for the jth member of the ith PSU is defined as

$$w_{ij}^{(r)} = w_{ij}/\alpha_r$$

With Stratification If the sample design involves stratification, each stratum must have at least two PSUs to use the jackknife method.

Let stratum $\tilde{h}_r$ be the stratum from which a PSU is deleted for the rth replicate. Stratum $\tilde{h}_r$ is called the *donor stratum.* Let $n_{\tilde{h}_r}$ be the total number of PSUs in the donor stratum $\tilde{h}_r$. The jackknife coefficients are defined as

$$\alpha_r = \frac{n_{\tilde{h}_r} - 1}{n_{\tilde{h}_r}} \quad \text{where } r = 1, 2, ..., R$$

Denote the original weight in the full sample for the jth member of the ith PSU as w_{ij}. If the ith PSU is included in the rth replicate ($r = 1, 2, ..., R$), then the corresponding replicate weight for the jth member of the ith PSU is defined as

$$w_{ij}^{(r)} = \begin{cases} w_{ij} & \text{if } i\text{th PSU is not in the donor stratum } \tilde{h}_r \\ w_{ij}/\alpha_r & \text{if } i\text{th PSU is in the donor stratum } \tilde{h}_r \end{cases}$$

You can use the VARMETHOD=JACKKNIFE(OUTJKCOEFS=) method-option to save the jackknife coefficients into a SAS data set and use the VARMETHOD=JACKKNIFE(OUTWEIGHTS=) method-option to save the replicate weights into a SAS data set.

If you provide your own replicate weights with a REPWEIGHTS statement, then you can also provide corresponding jackknife coefficients with the JKCOEFS= option.

Suppose that θ is a population parameter of interest. Let $\hat{\theta}$ be the estimate from the full sample for θ. Let $\hat{\theta}_r$ be the estimate from the rth replicate subsample by using replicate weights. PROC SURVEYMEANS estimates the variance of $\hat{\theta}$ by

$$\widehat{V}(\hat{\theta}) = \sum_{r=1}^{R} \alpha_r \left(\hat{\theta}_r - \hat{\theta}\right)^2$$

with $R - H$ degrees of freedom, where R is the number of replicates and H is the number of strata, or $R - 1$ when there is no stratification.

Hadamard Matrix

A Hadamard matrix $\mathbf{H}$ is a square matrix whose elements are either 1 or -1 such that

$$\mathbf{H}\mathbf{H}' = k\mathbf{I}$$

where k is the dimension of $\mathbf{H}$ and $\mathbf{I}$ is the identity matrix of order k. The order k is necessarily 1, 2, or a positive integer that is a multiple of 4.

For example, the following matrix is a Hadamard matrix of dimension $k = 8$:

$$\begin{array}{rrrrrrrr}
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1
\end{array}$$

Output Data Sets

You can use the Output Delivery System to create a SAS data set from any piece of PROC SURVEYMEANS output. See the section "ODS Table Names" on page 6517 for more information.

PROC SURVEYMEANS also provides an output data set that stores the replicate weights for BRR or jackknife variance estimation and an output data set that stores the jackknife coefficients for jackknife variance estimation.

Replicate Weights Output Data Set

If you specify the OUTWEIGHTS= method-option for BRR or jackknife method in the VARMETHOD= option, PROC SURVEYMEANS stores the replicate weights in an output data

set. The OUTWEIGHTS= output data set contains all observations used in the analysis or all valid observations in the DATA= input data set. (A valid observation is an observation that has a positive value of the WEIGHT variable. Valid observations must also have nonmissing values of the STRATA and CLUSTER variables, unless you specify the MISSING option. See the section "Data and Sample Design Summary" on page 6512 for details about valid observations.)

The OUTWEIGHTS= data set contains the following variables:

- all variables in the DATA= input data set
- RepWt_1, RepWt_2, . . ., RepWt_n, which are the replicate weight variables

where *n* is the total number of replicates in the analysis. Each replicate weight variable contains the replicate weights for the corresponding replicate. Replicate weights equal zero for those observations not included in the replicate.

After the procedure creates replicate weights for a particular input data set and survey design, you can use the OUTWEIGHTS= method-option to store these replicate weights and then use them again in subsequent analyses, either in PROC SURVEYMEANS or in the other survey procedures. You can use the REPWEIGHTS statement to provide replicate weights for the procedure.

Jackknife Coefficients Output Data Set

If you specify the OUTJKCOEFS= method-option for VARMETHOD=JACKKNIFE, PROC SURVEYMEANS stores the jackknife coefficients in an output data set. The OUTJKCOEFS= output data set contains one observation for each replicate. The OUTJKCOEFS= data set contains the following variables:

- Replicate, which is the replicate number for the jackknife coefficient
- JKCoefficient, which is the jackknife coefficient
- DonorStratum, which is the stratum of the PSU that was deleted to construct the replicate, if you specify a STRATA statement

After the procedure creates jackknife coefficients for a particular input data set and survey design, you can use the OUTJKCOEFS= method-option to store these coefficients and then use them again in subsequent analyses, either in PROC SURVEYMEANS or in the other survey procedures. You can use the JKCOEFS= option in the REPWEIGHTS statement to provide jackknife coefficients for the procedure.

Rectangular and Stacking Structures in an Output Data Set

When you use an ODS output statement to create SAS data sets for certain tables in PROC SURVEYMEANS, there are two types of table structure for a data set: *rectangular* and *stacking*. A rectangular structure creates one observation for each analysis variable in the data set. A stacking structure creates only one observation in the output data set for all analysis variables.

Before SAS 9, the stacking table structure, similar to the table structure in PROC MEANS, was the default in PROC SURVEYMEANS. Since SAS 9, the new default is to produce a rectangular table in the output data sets. You can use the STACKING option to request that the procedure produce the output data sets by using a stacking table structure.

The STACKING option affects the following tables:

- Domain
- Ratio
- Statistics
- StrataInfo

Figure 85.6 and Figure 85.7 shows these two structures for analyzing the following data set:

```
data new;
   input sex$ x;
   datalines;
M 12
F 5
M 13
F 23
F 11
;
```

The following statements request the default rectangular structure of the output data set for the statistics table:

```
proc surveymeans data=new mean;
   ods output statistics=rectangle;
run;
proc print data=rectangle;
run;
```

Figure 85.6 shows the rectangular structure.

Figure 85.6 Rectangular Structure in the Output Data Set

Rectangular Structure in the Output Data Set

Obs	Var Name	Var Level	Mean	StdErr
1	x		12.800000	2.905168
2	sex	F	0.600000	0.244949
3	sex	M	0.400000	0.244949

The following statements specify the STACKING option to request that the output data set have a stacking structure:

```
proc surveymeans data=new mean stacking;
   ods output statistics=stacking;
run;
proc print data=stacking;
run;
```

Figure 85.7 shows the stacking structure of the output data set for the statistics table requested by the STACKING option.

Figure 85.7 Stacking Structure in the Output Data Set Requested by the STACKING option

Stacking Structure in the Output Data Set

Obs	x	x_Mean	x_StdErr	sex_F	sex_F_Mean
1	x	12.800000	2.905168	sex=F	0.600000

Obs	sex_F_StdErr	sex_M	sex_M_Mean	sex_M_StdErr
1	0.244949	sex=M	0.400000	0.244949

Displayed Output

The SURVEYMEANS procedure produces the following output.

Data and Sample Design Summary

The "Data Summary" table provides information about the input data set and the sample design. This table displays the total number of valid observations, where an observation is considered *valid* if it has nonmissing values for all procedure variables other than the analysis variables—that is, for all specified STRATA, CLUSTER, and WEIGHT variables. This number might differ from the number of nonmissing observations for an individual analysis variable, which the procedure displays in the "Statistics" table. See the section "Missing Values" on page 6491 for more information.

PROC SURVEYMEANS displays the following information in the "Data Summary" table:

- Number of Strata, if you specify a STRATA statement
- Number of Clusters, if you specify a CLUSTER statement
- Number of Observations, which is the total number of valid observations
- Sum of Weights, which is the sum over all valid observations, if you specify a WEIGHT statement

Class Level Information

If you use a CLASS statement to name classification variables for categorical analysis, or if you list any character variables in the VAR statement, then PROC SURVEYMEANS displays a "Class Level Information" table. This table contains the following information for each classification variable:

- Class Variable, which lists each CLASS variable name
- Levels, which is the number of values or levels of the classification variable
- Values, which lists the values of the classification variable. The values are separated by a white space character; therefore, to avoid confusion, you should not include a white space character within a classification variable value.

Stratum Information

If you specify the LIST option in the STRATA statement, PROC SURVEYMEANS displays a "Stratum Information" table. This table displays the number of valid observations in each stratum, as well as the number of nonmissing stratum observations for each analysis variable. The "Stratum Information" table provides the following for each stratum:

- Stratum Index, which is a sequential stratum identification number
- STRATA variable(s), which lists the levels of STRATA variables for the stratum
- Population Total, if you specify the TOTAL= option
- Sampling Rate, if you specify the TOTAL= or RATE= option. If you specify the TOTAL= option, the sampling rate is based on the number of valid observations in the stratum.
- N Obs, which is the number of valid observations
- Variable, which lists each analysis variable name
- Levels, which identifies each level for categorical variables
- N, which is the number of nonmissing observations for the analysis variable
- Clusters, which is the number of clusters, if you specify a CLUSTER statement

Variance Estimation

If the variance method is not Taylor series or if the NOMCAR option is used, by default, PROC SURVEYMEANS displays the following variance estimation specifications in the "Variance Estimation" table:

- Method, which is the variance estimation method

- Number of Replicates, which is the number of replicates if you specify the VARMETHOD=BRR or VARMETHOD=JACKKNIFE option
- Hadamard Data Set, which is the name of the SAS data set for the HADAMARD matrix if you specify the VARMETHOD=BRR(HADAMARD=) method-option
- Fay Coefficient, which is the value of the FAY coefficient if you specify the VARMETHOD=BRR(FAY) method-option
- Replicate Weights Data Set, which is the name of the SAS data set that contains the replicate weights
- Missing Levels Included (MISSING), if you specify the MISSING option
- Missing Levels Include (NOMCAR), if you specify the NOMCAR option

Statistics

The "Statistics" table displays all of the statistics that you request with *statistic-keywords* in the PROC SURVEYMEANS statement, except DECILES, MEDIAN, Q1, Q3, and QUARTILES, which are displayed in the "Quantiles" table. If you do not specify any *statistic-keywords*, then by default this table displays the following information for each analysis variable: the sample size, the mean, the standard error of the mean, and the confidence limits for the mean. The "Statistics" table might contain the following information for each analysis variable, depending on which *statistic-keywords* you request:

- Variable name
- Variable Label
- Level, which identifies each level for categorical variables
- N, which is the number of nonmissing observations
- N Miss, which is the number of missing observations
- Minimum
- Maximum
- Range
- Number of Clusters
- Sum of Weights
- DF, which is the degrees of freedom for the t test
- Mean
- Std Error of Mean, which is the standard error of the mean

- Var of Mean, which is the variance of the mean
- t Value, for testing H_0 : population MEAN $= 0$
- Pr $>$ | t |, which is the two-sided p-value for the t test
- $100(1-\alpha)\%$ CL for Mean, which are two-sided confidence limits for the mean
- $100(1-\alpha)\%$ Upper CL for Mean, which is a one-sided upper confidence limit for the mean
- $100(1-\alpha)\%$ Lower CL for Mean, which is a one-sided lower confidence limit for the mean
- Coeff of Variation, which is the coefficients of variation for the mean and the sum
- Sum
- Std Dev, which is the standard deviation of the sum
- Var of Sum, which is the variance of the sum
- $100(1-\alpha)\%$ CL for Sum, which are two-sided confidence limits for the sum
- $100(1-\alpha)\%$ Upper CL for Sum, which is a one-sided upper confidence limit for the sum
- $100(1-\alpha)\%$ Lower CL for Sum, which is a one-sided lower confidence limit for the Sum

Quantiles

The "Quantiles" table displays all the quantiles that you request with either *statistic-keywords* such as DECILES, MEDIAN, Q1, Q3, and QUARTILES, or the PERCENTILE= option, or the QUANTILE= option in the PROC SURVEYMEANS statement.

The "Quantiles" table contains the following information for each quantile:

- Variable name
- Variable Label
- Percentile, which is the requested quantile in the format of %
- Percentile Label, which is the corresponding common name for a percentile if it exists—for example, *Median* for 50th percentile
- Estimate, which is the estimate for a requested quantile with respect to the population distribution
- StdErr, which is the standard error of the quantile
- $100(1-\alpha)\%$ Confidence Limits, which are two-sided confidence limits for the quantile

Domain Analysis

If you use a DOMAIN statement, the procedure displays statistics in each domain in a "Domain Analysis" table. A "Domain Analysis" table contains all the columns in the "Statistics" table, plus columns of domain variable values.

Note that depending on how you define the domains with domain variables, the procedure might produce more than one "Domain Analysis" table. For example, in the DOMAIN statement

```
domain A B*C*D A*C C;
```

you use four definitions to define domains:

- A: all the levels of A
- C: all the levels of C
- A*C: all the interactive levels of A and C
- B*C*D: all the interactive levels of B, C, and D

The procedure displays four "Domain Analysis" tables, one for each domain definition. If you use an ODS OUTPUT statement to create an output data set for domain analysis, the output data set contains a variable Domain whose values are these domain definitions.

Ratio Analysis

The "Ratio Analysis" table displays all of the statistics that you request with *statistic-keywords* in the PROC statement. If you do not specify any *statistic-keywords*, then by default this table displays the ratio and its standard error. The "Ratio Analysis" table might contain the following information for each ratio, depending on which *statistic-keywords* you request:

- Numerator, which identifies the numerator variable of the ratio
- Denominator, which identifies the denominator variable of the ratio
- N, which is the number of observations used in the ratio analysis
- number of Clusters
- Sum of Weights
- DF, which is the degrees of freedom for the t test
- Ratio
- Std Error of Ratio, which is the standard error of the ratio
- Var of Ratio, which is the variance of the ratio

- t Value, for testing H_0 : population RATIO $= 0$
- Pr $> |\ t\ |$, which is the two-sided p-value for the t test
- $100(1-\alpha)\%$ CL for Ratio, which are two-sided confidence limits for the Ratio
- Upper $100(1-\alpha)\%$ CL for Ratio, which are one-sided upper confidence limits for the Ratio
- Lower $100(1-\alpha)\%$ CL for Ratio, which are one-sided lower confidence limits for the Ratio

When you use the ODS OUTPUT statement to create an output data set, if you use labels for your RATIO statement, these labels are saved in the variable Ratio Statement in the output data set.

Hadamard Matrix

If you specify the VARMETHOD=BRR(PRINTH) method-option in the PROC statement, PROC SURVEYMEANS displays the Hadamard matrix.

When you provide a Hadamard matrix with the VARMETHOD=BRR(HADAMARD=) method-option, the procedure displays only used rows and columns of the Hadamard matrix.

ODS Table Names

PROC SURVEYMEANS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 85.3. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 85.3 ODS Tables Produced by PROC SURVEYMEANS

ODS Table Name	Description	Statement	Option
ClassVarInfo	Class level information	CLASS	default
Domain	Statistics in domains	DOMAIN	default
HadamardMatrix	Hadamard matrix	PROC	PRINTH
Ratio	Statistics for ratios	RATIO	default
Quantiles	Quantiles	PROC	default
Statistics	Statistics	PROC	default
StrataInfo	Stratum information	STRATA	LIST
Summary	Data summary	PROC	default
VarianceEstimation	Variance estimation	PROC	default

For example, the following statements create an output data set MyStrata, which contains the

"StrataInfo" table, and an output data set MyStat, which contains the "Statistics" table for the ice cream study discussed in the section "Stratified Sampling" on page 6469:

```
title1 'Analysis of Ice Cream Spending';
title2 'Stratified Simple Random Sample Design';
proc surveymeans data=IceCream total=StudentTotals;
   strata Grade / list;
   var Spending Group;
   weight Weight;
   ods output StrataInfo = MyStrata
              Statistics = MyStat;
run;
```

Examples: SURVEYMEANS Procedure

The section "Getting Started: SURVEYMEANS Procedure" on page 6467 contains examples of analyzing data from simple random sampling and stratified simple random sample designs. This section provides more examples that illustrate how to use PROC SURVEYMEANS.

Example 85.1: Stratified Cluster Sample Design

Consider the example in the section "Stratified Sampling" on page 6469. The study population is a junior high school with a total of 4,000 students in grades 7, 8, and 9. Researchers want to know how much these students spend weekly for ice cream, on the average, and what percentage of students spend at least $10 weekly for ice cream.

The example in the section "Stratified Sampling" on page 6469 assumes that the sample of students was selected using a stratified simple random sample design. This example shows analysis based on a more complex sample design.

Suppose that every student belongs to a study group and that study groups are formed within each grade level. Each study group contains between two and four students. Table 85.4 shows the total number of study groups for each grade.

Table 85.4 Study Groups and Students by Grade

Grade	Number of Study Groups	Number of Students
7	608	1,824
8	252	1,025
9	403	1,151
Total	617	4,000

It is quicker and more convenient to collect data from students in the same study group than to

collect data from students individually. Therefore, this study uses a stratified clustered sample design. The primary sampling units, or clusters, are study groups. The list of all study groups in the school is stratified by grade level. From each grade level, a sample of study groups is randomly selected, and all students in each selected study group are interviewed. The sample consists of eight study groups from the 7th grade, three groups from the 8th grade, and five groups from the 9th grade.

The SAS data set IceCreamStudy saves the responses of the selected students:

```
data IceCreamStudy;
   input Grade StudyGroup Spending @@;
   if (Spending < 10) then Group='less';
      else Group='more';
   datalines;
7  34  7     7  34  7    7 412  4     9  27 14
7  34  2     9 230 15    9  27 15     7 501  2
9 230  8     9 230  7    7 501  3     8  59 20
7 403  4     7 403 11    8  59 13     8  59 17
8 143 12     8 143 16    8  59 18     9 235  9
8 143 10     9 312  8    9 235  6     9 235 11
9 312 10     7 321  6    8 156 19     8 156 14
7 321  3     7 321 12    7 489  2     7 489  9
7  78  1     7  78 10    7 489  2     7 156  1
7  78  6     7 412  6    7 156  2     9 301  8
;
```

In the data set IceCreamStudy, the variable Grade contains a student's grade. The variable StudyGroup identifies a student's study group. It is possible for students from different grades to have the same study group number because study groups are sequentially numbered within each grade. The variable Spending contains a student's response regarding how much he spends per week for ice cream, in dollars. The variable GROUP indicates whether a student spends at least $10 weekly for ice cream. It is not necessary to store the data in order of grade and study group.

The SAS data set StudyGroup is created to provide PROC SURVEYMEANS with the sample design information shown in Table 85.4:

```
data StudyGroups;
   input Grade _total_; datalines;
7 608
8 252
9 403
;
```

The variable Grade identifies the strata, and the variable _TOTAL_ contains the total number of study groups in each stratum. As discussed in the section "Specification of Population Totals and Sampling Rates" on page 6492, the population totals stored in the variable _TOTAL_ should be expressed in terms of the primary sampling units (PSUs), which are study groups in this example. Therefore, the variable _TOTAL_ contains the total number of study groups for each grade, rather than the total number of students.

In order to obtain unbiased estimates, you create sampling weights by using the following SAS statements:

```
data IceCreamStudy;
   set IceCreamStudy;
   if Grade=7 then Prob=8/608;
   if Grade=8 then Prob=3/252;
   if Grade=9 then Prob=5/403;
   Weight=1/Prob;
```

The sampling weights are the reciprocals of the probabilities of selections. The variable Weight contains the sampling weights. Because the sampling design is clustered and all students from each selected cluster are interviewed, the sampling weights equal the inverse of the cluster (or study group) selection probabilities.

The following SAS statements perform the analysis for this sample design:

```
title1 'Analysis of Ice Cream Spending';
title2 'Stratified Clustered Sample Design';
proc surveymeans data=IceCreamStudy total=StudyGroups;
   strata Grade / list;
   cluster StudyGroup;
   var Spending Group;
   weight Weight;
run;
```

Output 85.1.1 provides information about the sample design and the input data set. There are three strata in the sample design, and the sample contains 16 clusters and 40 observations. The variable Group has two levels, 'less' and 'more.'

Output 85.1.1 Data Summary and Class Information

```
                    Analysis of Ice Cream Spending
                  Stratified Clustered Sample Design

                      The SURVEYMEANS Procedure

                             Data Summary

                  Number of Strata                   3
                  Number of Clusters                16
                  Number of Observations            40
                  Sum of Weights                 3162.6


                       Class Level Information

                  Class
                  Variable      Levels    Values

                  Group              2    less more
```

Output 85.1.2 displays information for each stratum. Since the primary sampling units in this design are study groups, the population totals shown in Output 85.1.2 are the total numbers of study groups for each stratum or grade. This differs from Output 85.3, which provides the population totals in

terms of students since students were the primary sampling units for that design. Output 85.1.2 also displays the number of clusters for each stratum and analysis variable.

Output 85.1.2 Stratum Information

Stratum Information

Stratum Index	Grade	Population Total	Sampling Rate	N Obs	Variable	Level
1	7	608	1.32%	20	Spending Group	less more
2	8	252	1.19%	9	Spending Group	less more
3	9	403	1.24%	11	Spending Group	less more

Stratum Information

Stratum Index	Grade	Population Total	Sampling Rate	N Obs	Variable	N
1	7	608	1.32%	20	Spending	20
					Group	17
						3
2	8	252	1.19%	9	Spending	9
					Group	0
						9
3	9	403	1.24%	11	Spending	11
					Group	6
						5

Stratum Information

Stratum Index	Grade	Population Total	Sampling Rate	N Obs	Variable	Clusters
1	7	608	1.32%	20	Spending	8
					Group	8
						3
2	8	252	1.19%	9	Spending	3
					Group	0
						3
3	9	403	1.24%	11	Spending	5
					Group	4
						4

Output 85.1.3 displays the estimates of the average weekly ice cream expenditure and the percentage of students spending at least $10 weekly for ice cream.

Output 85.1.3 Statistics

Statistics						
Variable	Level	N	Mean	Std Error of Mean	95% CL for Mean	
Spending		40	8.923860	0.650859	7.51776370	10.3299565
Group	less	23	0.561437	0.056368	0.43966057	0.6832130
	more	17	0.438563	0.056368	0.31678698	0.5603394

Example 85.2: Domain Analysis

Suppose that you are studying profiles of 800 top-performing companies to provide information about their impact on the economy. You are also interested in the company profiles within each market type. A sample of 66 companies is selected with unequal probability across market types. However, market type is not included in the sample design. Thus, the number of companies within each market type is a random variable in your sample. To obtain statistics within each market type, you should use domain analysis. The data of the 66 companies are saved in the following data set.

```
data Company;
   length Type $14;
   input Type$ Asset Sale Value Profit Employee Weight;
   datalines;
Other            2764.0  1828.0  1850.3   144.0   18.7   9.6
Energy          13246.2  4633.5  4387.7   462.9   24.3  42.6
Finance          3597.7   377.8    93.0    14.0    1.1  12.2
Transportation   6646.1  6414.2  2377.5   348.2   47.1  21.8
HiTech           1068.4  1689.8  1430.2    72.9    4.6   4.3
Manufacturing    1125.0  1719.4  1057.5    98.1   20.4   4.5
Other            1459.0  1241.4   452.7    24.5   20.1   5.5
Finance          2672.3   262.5   296.2    23.1    2.2   9.3
Finance           311.0   566.2   932.0    52.8    2.7   1.9
Energy           1148.6  1014.6   485.1    60.6    4.0   4.5
Finance          5327.0   572.4   372.9    25.2    4.2  17.7
Energy           1602.7   678.4   653.0    75.6    2.8   6.0
Energy           5808.8  1288.4  2007.0   318.8    5.9  19.2
Medical           268.8   204.4   820.9    45.6    3.7   1.8
Transportation   5222.6  2627.8  1910.0   245.6   22.8  17.4
Other             872.7  1419.4   939.3    69.7   12.2   3.7
Retail           4461.7  8946.8  4662.7   289.0  132.1  15.0
HiTech           6719.2  6942.0  8240.2   381.3   85.8  22.1
Retail            833.4  1538.8  1090.3    64.9   15.4   3.5
Finance           415.9   167.3  1126.8    56.8    0.7   2.2
HiTech            442.4  1139.9  1039.9    57.6   22.7   2.3
Other             801.5  1157.0   664.2    56.9   15.5   3.4
Finance          4954.8   468.8   366.4    41.7    3.0  16.5
Finance          2661.9   257.9   181.1    21.2    2.1   9.3
```

```
Finance          5345.8   530.1   337.4    36.4    4.3  17.8
Energy           3334.3  1644.7  1407.8   157.6    6.4  11.4
Manufacturing    1826.6  2671.7   483.2    71.3   25.3   6.7
Retail            618.8  2354.7   767.7    58.6   19.0   2.9
Retail           1529.1  6534.0   826.3    58.3   65.8   5.7
Manufacturing    4458.4  4824.5  3132.1    28.9   67.0  15.0
HiTech           5831.7  6611.1  9464.7   459.6   86.7  19.3
Medical          6468.3  4199.2  3170.4   270.1   59.5  21.3
Energy           1720.7   473.1   811.1    86.6    1.6   6.3
Energy           1679.7  1379.9   721.1    91.8    4.5   6.2
Retail           4018.2 16823.4  2038.3   178.1  162.0  13.6
Other             227.1   575.8  1083.8    62.6    1.9   1.6
Finance          3872.8   362.0   209.3    27.6    2.4  13.1
Retail           3359.3  4844.7  2651.4   224.1   75.6  11.5
Energy           1295.6   356.9   180.8   162.3    0.6   5.0
Energy           1658.0   626.6   688.0   126.0    3.5   6.1
Finance         12156.7  1345.5   680.7   106.6    9.4  39.2
HiTech           3982.6  4196.0  3946.8   313.9   64.3  13.5
Finance          8760.7   886.4  1006.9    90.0    7.5  28.5
Manufacturing    2362.2  3153.3  1080.0   137.0   25.2   8.4
Transportation   2499.9  3419.0   992.6    47.2   25.3   8.8
Energy           1430.4  1610.0   664.3    77.7    3.5   5.4
Energy          13666.5 15465.4  2736.7   411.4   26.6  43.9
Manufacturing    4069.3  4174.7  2907.6   289.2   38.2  13.7
Energy           2924.7   711.9  1067.8   146.7    3.4  10.1
Transportation   1262.1  1716.0   364.3    71.2   14.5   4.9
Medical           684.4   672.9   287.4    61.8    6.0   3.1
Energy           3069.3  1719.0  1439.0   196.4    4.9  10.6
Medical           246.5   318.8   924.1    43.8    3.1   1.7
Finance         11562.2  1128.5   580.4    64.2    6.7  37.3
Finance          9316.0  1059.4   816.5    95.9    8.0  30.2
Retail           1094.3  3848.0   563.3    29.4   44.7   4.4
Retail           1102.1  4878.3   932.4    65.2   47.3   4.4
HiTech            466.4   675.8   845.7    64.5    5.2   2.4
Manufacturing   10839.4  5468.7  1895.4   232.8   47.8  35.0
Manufacturing     733.5  2135.3    96.6    10.9    2.7   3.2
Manufacturing   10354.2 14477.4  5607.2   321.9  188.5  33.5
Energy           1902.1  2697.9   329.3    34.2    2.2   6.9
Other            2245.2  2132.2  2230.4   198.9    8.0   8.0
Transportation    949.4  1248.3   298.9    35.4   10.4   3.9
Retail           2834.4  2884.6   458.2    41.2   49.8   9.8
Retail           2621.1  6173.8  1992.7   183.7  115.1   9.2
;
```

For each company in your sample, the variables are defined as follows:

- Type identifies the type of market for the company.
- Asset contains the company's assets, in millions of dollars.
- Sale contains sales, in millions of dollars.
- Value contains the market value of the company, in millions of dollars.

- Profit contains the profit, in millions of dollars.
- Employee contains the number of employees, in thousands.
- Weight contains the sampling weight.

The following SAS statements use PROC SURVEYMEANS to perform the domain analysis, estimating means, and other statistics for the overall population and also for the subpopulations (or domain) defined by market type. The DOMAIN statement specifies Type as the domain variable:

```
title 'Top Companies Profile Study';
proc surveymeans data=Company total=800 mean sum;
   var Asset Sale Value Profit Employee;
   weight Weight;
   domain Type;
run;
```

Output 85.2.1 shows that there are 66 observations in the sample. The sum of the sampling weights equals 799.8, which is close to the total number of companies in the study population.

Output 85.2.1 Company Profile Study

Top Companies Profile Study

The SURVEYMEANS Procedure

Data Summary

Number of Observations	66
Sum of Weights	799.8

Statistics

Variable	Mean	Std Error of Mean	Sum	Std Dev
Asset	6523.488510	720.557075	5217486	1073829
Sale	4215.995799	839.132506	3371953	847885
Value	2145.935121	342.531720	1716319	359609
Profit	188.788210	25.057876	150993	30144
Employee	36.874869	7.787857	29493	7148.003298

The "Statistics" table in Output 85.2.1 displays the estimates of the mean and total for all analysis variables for the entire set of 800 companies, while Output 85.2.2 shows the mean and total estimates for each company type.

Output 85.2.2 Domain Analysis for Company Profile Study

Domain Analysis: Type

Type	Variable	Mean	Std Error of Mean	Sum	Std Dev
Energy	Asset	7868.302932	1941.699163	1449341	785962
	Sale	5419.679099	2416.214417	998305	673373
	Value	2249.297177	520.295162	414321	213580
	Profit	289.564658	52.512141	53338	25927
	Employee	14.151194	3.974697	2606.650000	1481.777769
Finance	Asset	7890.190264	1057.185336	1855773	704506
	Sale	829.210502	115.762531	195030	74436
	Value	565.068197	76.964547	132904	48156
	Profit	63.716837	10.099341	14986	5801.108513
	Employee	5.806293	0.811555	1365.640000	519.658410
HiTech	Asset	5031.959781	732.436967	321542	183302
	Sale	5464.292019	731.296997	349168	196013
	Value	6707.828482	1194.160584	428630	249154
	Profit	346.407042	42.299004	22135	12223
	Employee	70.766980	8.683595	4522.010000	2524.778281
Manufacturing	Asset	7403.004250	1454.921083	888361	492577
	Sale	7207.638833	2112.444703	864917	501679
	Value	2986.442750	799.121544	358373	196979
	Profit	211.933583	39.993255	25432	13322
	Employee	83.314333	31.089019	9997.720000	6294.309490
Medical	Asset	5046.570609	1218.444638	140799	131942
	Sale	3313.219713	758.216303	92439	85655
	Value	2561.614695	530.802245	71469	64663
	Profit	218.682796	44.051447	6101.250000	5509.560969
	Employee	46.518996	11.135955	1297.880000	1213.651734
Other	Asset	1850.250000	338.128984	58838	31375
	Sale	1620.784906	168.686773	51541	24593
	Value	1432.820755	297.869828	45564	24204
	Profit	115.089937	27.970560	3659.860000	2018.201371
	Employee	14.306604	2.313733	454.950000	216.327710
Retail	Asset	2939.845750	393.692369	235188	94605
	Sale	7395.453500	1746.187580	591636	263263
	Value	2103.863125	529.756409	168309	78304
	Profit	157.171875	31.734253	12574	5478.281027
	Employee	93.624000	15.726743	7489.920000	3093.832061
Transportation	Asset	4712.047359	888.954411	267644	163516
	Sale	4030.233275	1015.555708	228917	142669
	Value	1703.330282	313.841326	96749	58947
	Profit	224.762324	56.168925	12767	8287.585418
	Employee	30.946303	6.786270	1757.750000	1066.586615

Example 85.3: Ratio Analysis

Suppose you are interested in the profit per employee and the sale per employee among the 800 top-performing companies in the data in the previous example. The following SAS statements illustrate

how you can use PROC SURVEYMEANS to estimate these ratios:

```
title 'Ratio Analysis in Top Companies Profile Study';
proc surveymeans data=Company total=800 ratio;
   var Profit Sale Employee;
   weight Weight;
   ratio Profit Sale / Employee;
run;
```

The RATIO statement requests the ratio of the profit and the sales to the number of employees.

Output 85.3.1 shows the estimated ratios and their standard errors. Because the profit and the sales figures are in millions of dollars, and the employee numbers are in thousands, the profit per employee is estimated as $5,120 with a standard error of $1,059, and the sales per employee are $114,332 with a standard error of $20,503.

Output 85.3.1 Estimate Ratios

Ratio Analysis in Top Companies Profile Study

The SURVEYMEANS Procedure

Ratio Analysis

Numerator	Denominator	Ratio	Std Err
Sale	Employee	114.332497	20.502742
Profit	Employee	5.119698	1.058939

Example 85.4: Analyzing Survey Data with Missing Values

As described in the section "Missing Values" on page 6491, the SURVEYMEANS procedure excludes an observation from the analysis if it has a missing value for the analysis variable or a nonpositive value for the WEIGHT variable.

However, if there is evidence indicating that the nonrespondents are different from the respondents for your study, you can use the NOMCAR option to compute descriptive statistics among respondents while still counting the number of nonrespondents.

We use the ice cream example in the section "Stratified Sampling" on page 6469 to illustrate how to perform similar analysis when there are missing values.

Suppose that some of the students failed to provide the amounts spent on ice cream, as shown in the following data set, IceCream:

```
data IceCream;
   input Grade Spending @@;
   if Grade=7 then Prob=20/1824;
   if Grade=8 then Prob=9/1025;
   if Grade=9 then Prob=11/1151;
   Weight=1/Prob;
   datalines;
7 7  7  7  8  .  9 10  7  .  7 10  7  3  8 20  8 19  7 2
7 .  9 15  8 16  7  6  7  6  7  6  9 15  8 17  8 14  9 .
9 8  9  7  7  3  7 12  7  4  9 14  8 18  9  9  7  2  7 1
7 4  7 11  9  8  8  .  8 13  7  .  9  .  9 11  7  2  7 9
;
data StudentTotals;
   input Grade _total_; datalines;
7 1824
8 1025
9 1151
;
```

Considering the possibility that those students who did not respond spend differently than those students who did respond, you can use the NOMCAR option to request the analysis to treat the respondents as a domain rather than exclude the nonrespondents.

The following SAS statements produce the desired analysis:

```
title 'Analysis of Ice Cream Spending';
proc surveymeans data=IceCream total=StudentTotals nomcar mean sum;
   strata Grade;
   var Spending;
   weight Weight;
run;
```

Output 85.4.1 summarize the analysis including the variance estimation method.

Output 85.4.1 Analysis of Incomplete Ice Cream Data Excluding Observations with Missing Values

```
                    Analysis of Ice Cream Spending

                      The SURVEYMEANS Procedure

                              Data Summary

                  Number of Strata                   3
                  Number of Observations            40
                  Sum of Weights                  4000


                        Variance Estimation

                  Method                Taylor Series
                  Missing Values    Included (NOMCAR)
```

Output 85.4.2 shows the mean and total estimates when treating respondents as a domain in the student population. Although the point estimates are the same as the analysis without the NOMCAR option, for this particular example, the variance estimations are slightly higher when you assume that the missingness is not completely at random.

Output 85.4.2 Analysis of Incomplete Ice Cream Data Excluding Observations with Missing Values

Statistics

Variable	Mean	Std Error of Mean	Sum	Std Dev
Spending	9.770542	0.652347	32139	3515.126876

Example 85.5: Variance Estimation Using Replication Methods

In order to improve service, the San Francisco Municipal Railway (MUNI) conducts a survey to estimated passenger's average waiting time for MUNI's subway system.

The study uses a stratified cluster sample design. Each MUNI subway line is a stratum. The subway lines included in the study are 'J-Church,' 'K-Ingleside,' 'L-Taraval,' 'M-Ocean View,' 'N-Judah,' and the street car 'F-Market & Wharves.' The MUNI vehicles in service for these lines during a day are primary sampling units. Within each stratum, two vehicles (PSUs) are randomly selected. Then the waiting times of passengers for a selected MUNI vehicle are collected.

Table 85.5 shows the number of passengers that are interviewed in each of the selected MUNI vehicles.

Table 85.5 The Sample of the MUNI Waiting Time Study

MUNI Line	Vehicle	Number of Passengers
F-Market & Wharves	1	65
	2	102
J-Church	1	101
	2	142
K-Ingleside	1	145
	2	180
L-Taraval	1	135
	2	185
M-Ocean View	1	139
	2	203
N-Judah	1	306
	2	234

The collected data are saved in the SAS data set MUNIsurvey. The variable Line indicates which MUNI line a passenger is riding. The variable vehicle identifies the vehicle that a passenger is boarding. The variable Waittime is the time (in minutes) that a passenger waited. The variable weight contains the sampling weights, which are determined by selection probabilities within each stratum.

Output 85.5.1 displays the first 10 observations of the data set MUNIsurvey.

Output 85.5.1 First 10 Observations in the Data Set from the MUNI Subway Survey

MUNI subway passenger waiting time analysis

Obs	line	vehicle	passenger	waittime	weight
1	F-Market & Wharves	1	1	18	59
2	F-Market & Wharves	1	2	0	59
3	F-Market & Wharves	1	3	16	59
4	F-Market & Wharves	1	4	13	59
5	F-Market & Wharves	1	5	5	59
6	F-Market & Wharves	1	6	13	59
7	F-Market & Wharves	1	7	7	59
8	F-Market & Wharves	1	8	5	59
9	F-Market & Wharves	1	9	16	59
10	F-Market & Wharves	1	10	5	59

Using the VARMETHOD=BRR option, the following SAS statements analyze the MUNI subway survey by using the BRR method to estimate the variance:

```
title 'MUNI passenger waiting time analysis using BRR';
proc surveymeans data=MUNIsurvey mean varmethod=brr mean clm;
   strata line;
   cluster vehicle;
   var waittime;
   weight weight;
run;
```

The STRATUM variable is line, which corresponds to MUNI lines. The two clusters within each stratum are identified by the variable vehicle. The sampling weights are stored in the variable weight. The mean and confident limits of passenger waiting time (in minutes) are requested statistics.

Output 85.5.2 summarizes the data and indicates that the variance estimation method is BRR with 8 replicates.

Output 85.5.2 MUNI Passenger Waiting Time Analysis Using the BRR Method

MUNI passenger waiting time analysis using BRR

The SURVEYMEANS Procedure

Data Summary

Number of Strata	6
Number of Clusters	12
Number of Observations	1937
Sum of Weights	143040

Output 85.5.2 *continued*

```
                    Variance Estimation

              Method                          BRR
              Number of Replicates              8
```

Output 85.5.3 reports that the average passenger waiting time for a MUNI vehicle is 7.33 minutes, with an estimated standard of 0.24 minutes, using the BRR method. The 95% confident limits for the mean are estimated as 6.75 to 7.91 minutes.

Output 85.5.3 MUNI Passenger Waiting Time Analysis Using the BRR Method

```
                          Statistics

                                Std Error
Variable            Mean          of Mean       95% CL for Mean
-----------------------------------------------------------------
waittime        7.333012        0.237557    6.75172983 7.91429366
-----------------------------------------------------------------
```

Alternatively, the variance can be estimated using the jackknife method if the VARMETHOD=JACKKNIFE option is used. The following SAS statements analyze the MUNI subway survey by using the jackknife method to estimate the variance:

```
title 'MUNI passenger waiting time analysis using jackknife';
proc surveymeans data=MUNIsurvey mean varmethod=jackknife mean clm;
   strata line;
   cluster vehicle;
   var waittime;
   weight weight;
run;
```

Output 85.5.4 summarizes the data and indicates that the variance estimation method is jackknife with 12 replicates.

Output 85.5.4 MUNI Passenger Waiting Time Analysis Using the Jackknife Method

```
          MUNI passenger waiting time analysis using jackknife

                    The SURVEYMEANS Procedure

                          Data Summary

                Number of Strata                   6
                Number of Clusters                12
                Number of Observations          1937
                Sum of Weights                143040
```

Output 85.5.4 *continued*

Variance Estimation	
Method	Jackknife
Number of Replicates	12

Output 85.5.5 reports the statistics computed using the jackknife method. Although the average passenger waiting time remains the same (7.33 minutes), the standard error is slightly smaller 0.23 minutes when the jackknife method is used, as opposed to 0.24 minutes when the BRR method is used. The 95% confident limits are between 6.76 and 7.90 minutes when the jackknife method is used.

Output 85.5.5 MUNI Passenger Waiting Time Analysis Using the Jackknife Method

Statistics

Variable	Mean	Std Error of Mean	95% CL for Mean	
waittime	7.333012	0.232211	6.76481105	7.90121244

References

Brick, J. M. and Kalton, G. (1996), "Handling Missing Data in Survey Research," *Statistical Methods in Medical Research*, 5, 215–238.

Cochran, W. G. (1977), *Sampling Techniques*, Third Edition, New York: John Wiley & Sons.

Dippo, C. S., Fay, R. E., and Morganstein, D. H. (1984), "Computing Variances from Complex Samples with Replicate Weights," *Proceedings of the Survey Research Methods Section, ASA*, 489–494.

Dorfman, A. and Valliant, R. (1993), "Quantile Variance Estimators in Complex Surveys," *Proceedings of the Survey Research Methods Section, ASA*, 866–871.

Fay, R. E. (1984), "Some Properties of Estimators of Variance Based on Replication Methods," *Proceedings of the Survey Research Methods Section, ASA*, 495–500.

Fay, R. E. (1989), "Theory and Application of Replicate Weighting for Variance Calculations," *Proceedings of the Survey Research Methods Section, ASA*, 212–217.

Francisco, C. A. and Fuller, W. A. (1991), "Quantile Estimation with a Complex Survey Design," *Annals of Statistics*, 19, 454–469.

Fuller, W. A. (1975), "Regression Analysis for Sample Survey," *Sankhyā*, 37, Series C, Pt. 3, 117–

132.

Fuller, W. A., Kennedy, W., Schnell, D., Sullivan, G., and Park, H. J. (1989), *PC CARP*, Ames: Statistical Laboratory, Iowa State University.

Hansen, M. H., Hurwitz, W. N., and Madow, W. G. (1953), *Sample Survey Methods and Theory*, Volumes I and II, New York: John Wiley & Sons.

Hidiroglou, M. A., Fuller, W. A., and Hickman, R. D. (1980), *SUPER CARP*, Ames, IA: Statistical Laboratory, Iowa State University.

Judkins, D. (1990), "Fay's Method for Variance Estimation," *Journal of Official Statistics*, 6, 223–239.

Kalton, G. (1983), *Introduction to Survey Sampling*, Sage University Paper series on Quantitative Applications in the Social Sciences, series no. 07-035, Beverly Hills, CA, and London: Sage Publications.

Kalton, G., and Kaspyzyk, D. (1986), "The Treatment of Missing Survey Data," *Survey Methodology*, 12, 1–16.

Kish, L. (1965), *Survey Sampling*, New York: John Wiley & Sons.

Lee, E. S., Forthoffer, R. N., and Lorimor, R. J. (1989), *Analyzing Complex Survey Data*, Sage University Paper series on Quantitative Applications in the Social Sciences, series no. 07-071, Beverly Hills, CA, and London: Sage Publications.

Lohr, S. L. (1999), *Sampling: Design and Analysis*, Pacific Grove, CA: Duxbury Press.

Rao, J. N. K. and Shao, J. (1996), "On Balanced Half Sample Variance Estimation in Stratified Sampling," *Journal of the American Statistical Association,* 91, 343–348.

Rao, J. N. K. and Shao, J. (1999), "Modified Balanced Repeated Replication for Complex Survey Data," *Biometrika,* 86, 403–415.

Rao, J. N. K., Wu, C. F. J., and Yue, K. (1992), " Some Recent Work on Resampling Methods for Complex Surveys," *Survey Methodology,* 18, 209–217.

Rust, K. (1985), "Variance Estimation for Complex Estimators in Sample Surveys," *Journal of Official Statistics,* 1, 381–397.

Särndal, C. E., Swensson, B., and Wretman, J. (1992), *Model Assisted Survey Sampling*, New York: Springer-Verlag.

Wolter, K. M. (1985), *Introduction to Variance Estimation*, New York: Springer-Verlag.

Woodruff, R. S. (1971), "A Simple Method for Approximating the Variance of a Complicated Estimate," *Journal of the American Statistical Association,* 66, 411–414.

Chapter 86

The SURVEYREG Procedure

Contents

Overview: SURVEYREG Procedure **6535**
Getting Started: SURVEYREG Procedure **6535**
Simple Random Sampling 6535
Stratified Sampling 6538
Output Data Sets 6541
Syntax: SURVEYREG Procedure **6542**
PROC SURVEYREG Statement 6543
BY Statement 6549
CLASS Statement 6549
CLUSTER Statement 6550
CONTRAST Statement 6551
DOMAIN Statement 6552
ESTIMATE Statement 6553
MODEL Statement 6555
OUTPUT Statement 6556
REPWEIGHTS Statement 6558
STRATA Statement 6559
WEIGHT Statement 6560
Details: SURVEYREG Procedure **6560**
Missing Values 6560
Survey Design Information 6561
Specification of Population Totals and Sampling Rates 6561
Primary Sampling Units (PSUs) 6562
Computational Details 6563
Notation 6563
Regression Coefficients 6563
Variance Estimation 6564
Hadamard Matrix 6568
Degrees of Freedom 6568
Testing Effects 6570
Design Effect 6571
Stratum Collapse 6572
Sampling Rate of the Pooled Stratum from Collapse 6572
Contrasts 6573

Domain Analysis 6573
Computational Resources 6573
Output Data Sets 6574
OUT= Data Set Created by the OUTPUT Statement 6574
Replicate Weights Output Data Set 6575
Jackknife Coefficients Output Data Set 6575
Displayed Output 6576
Data Summary 6576
Design Summary 6576
Domain Summary 6577
Fit Statistics 6577
Variance Estimation 6577
Stratum Information 6578
Class Level Information 6578
X'X Matrix 6578
Inverse Matrix of X'X 6579
ANOVA for Dependent Variable 6579
Tests of Model Effects 6579
Estimated Regression Coefficients 6579
Covariance of Estimated Regression Coefficients 6580
Coefficients of Contrast 6580
Analysis of Contrasts 6580
Coefficients of Estimate 6580
Analysis of Estimable Functions 6580
Hadamard Matrix 6581
ODS Table Names 6581
Examples: SURVEYREG Procedure **6582**
Example 86.1: Simple Random Sampling 6582
Example 86.2: Simple Random Cluster Sampling 6585
Example 86.3: Regression Estimator for Simple Random Sample 6588
Example 86.4: Stratified Sampling 6589
Example 86.5: Regression Estimator for Stratified Sample 6595
Example 86.6: Stratum Collapse 6598
Example 86.7: Domain Analysis 6603
Example 86.8: Variance Estimate Using the Jackknife Method 6606
References **6610**

Overview: SURVEYREG Procedure

The SURVEYREG procedure performs regression analysis for sample survey data. This procedure can handle complex survey sample designs, including designs with stratification, clustering, and unequal weighting. The procedure fits linear models for survey data and computes regression coefficients and their variance-covariance matrix. The procedure also provides significance tests for the model effects and for any specified estimable linear functions of the model parameters. Using the regression model, the procedure can compute predicted values for the sample survey data.

PROC SURVEYREG computes the regression coefficient estimators by generalized least squares estimation using elementwise regression. The procedure assumes that the regression coefficients are the same across strata and primary sampling units (PSUs). To estimate the variance-covariance matrix for the regression coefficients, PROC SURVEYREG uses either the Taylor series (linearization) method or replication (resampling) methods to estimate sampling errors of estimators based on complex sample designs. For details see Woodruff (1971); Fuller (1975); Särndal, Swensson, and Wretman (1992); Wolter (1985); Rust (1985); Dippo, Fay, and Morganstein (1984); Rao and Shao (1999); Rao, Wu, and Yue (1992); and Rao and Shao (1996).

Getting Started: SURVEYREG Procedure

This section demonstrates how you can use PROC SURVEYREG to perform a regression analysis for sample survey data. For a complete description of the usage of PROC SURVEYREG, see the section "Syntax: SURVEYREG Procedure" on page 6542. The section "Examples: SURVEYREG Procedure" on page 6582 provides more detailed examples that illustrate the applications of PROC SURVEYREG.

Simple Random Sampling

Suppose that, in a junior high school, there are a total of 4,000 students in grades 7, 8, and 9. You want to know how household income and the number of children in a household affect students' average weekly spending for ice cream.

In order to answer this question, you draw a sample by using simple random sampling from the student population in the junior high school. You randomly select 40 students and ask them their average weekly expenditure for ice cream, their household income, and the number of children in their household. The answers from the 40 students are saved as the SAS data set IceCream:

```
data IceCream;
   input Grade Spending Income Kids @@;
   datalines;
7   7  39  2   7   7  38  1   8  12  47  1
9  10  47  4   7   1  34  4   7  10  43  2
7   3  44  4   8  20  60  3   8  19  57  4
7   2  35  2   7   2  36  1   9  15  51  1
8  16  53  1   7   6  37  4   7   6  41  2
7   6  39  2   9  15  50  4   8  17  57  3
8  14  46  2   9   8  41  2   9   8  41  1
9   7  47  3   7   3  39  3   7  12  50  2
7   4  43  4   9  14  46  3   8  18  58  4
9   9  44  3   7   2  37  1   7   1  37  2
7   4  44  2   7  11  42  2   9   8  41  2
8  10  42  2   8  13  46  1   7   2  40  3
9   6  45  1   9  11  45  4   7   2  36  1
7   9  46  1
;
```

In the data set IceCream, the variable Grade indicates a student's grade. The variable Spending contains the dollar amount of each student's average weekly spending for ice cream. The variable Income specifies the household income, in thousands of dollars. The variable Kids indicates how many children are in a student's family.

The following PROC SURVEYREG statements request a regression analysis:

```
title1 'Ice Cream Spending Analysis';
title2 'Simple Random Sample Design';
proc surveyreg data=IceCream total=4000;
   class Kids;
   model Spending = Income Kids / solution;
run;
```

The PROC SURVEYREG statement invokes the procedure. The TOTAL=4000 option specifies the total in the population from which the sample is drawn. The CLASS statement requests that the procedure use the variable Kids as a classification variable in the analysis. The MODEL statement describes the linear model that you want to fit, with Spending as the dependent variable and Income and Kids as the independent variables. The SOLUTION option in the MODEL statement requests that the procedure output the regression coefficient estimates.

Figure 86.1 displays the summary of the data, the summary of the fit, and the levels of the classification variable Kids. The "Fit Statistics" table displays the denominator degrees of freedom, which are used in F tests and t tests in the regression analysis.

Figure 86.1 Summary of Data

```
                    Ice Cream Spending Analysis
                    Simple Random Sample Design

                      The SURVEYREG Procedure

          Regression Analysis for Dependent Variable Spending

                            Data Summary

                  Number of Observations            40
                  Mean of Spending             8.75000
                  Sum of Spending            350.00000

                          Fit Statistics

                       R-square            0.8132
                       Root MSE            2.4506
                       Denominator DF          39

                        Class Level Information

                    Class
                    Variable      Levels    Values

                    Kids               4    1 2 3 4
```

Figure 86.2 displays the tests for model effects. The effect Income is significant in the linear regression model, while the effect Kids is not significant at the 5% level.

Figure 86.2 Testing Effects in the Regression

```
                          Tests of Model Effects

                  Effect       Num DF    F Value    Pr > F

                  Model             4     119.15    <.0001
                  Intercept         1     153.32    <.0001
                  Income            1     324.45    <.0001
                  Kids              3       0.92    0.4385

        NOTE: The denominator degrees of freedom for the F tests is 39.
```

The regression coefficient estimates and their standard errors and associated t tests are displayed in Figure 86.3.

Figure 86.3 Regression Coefficients

```
                    Estimated Regression Coefficients

                                    Standard
        Parameter      Estimate        Error    t Value    Pr > |t|

        Intercept    -26.084677    2.46720403     -10.57      <.0001
        Income         0.775330    0.04304415      18.01      <.0001
        Kids 1         0.897655    1.12352876       0.80      0.4292
        Kids 2         1.494032    1.24705263       1.20      0.2381
        Kids 3        -0.513181    1.33454891      -0.38      0.7027
        Kids 4         0.000000    0.00000000        .         .

NOTE: The denominator degrees of freedom for the t tests is 39.
      Matrix X'X is singular and a generalized inverse was used to solve the
      normal equations.  Estimates are not unique.
```

Stratified Sampling

Suppose that the previous student sample is actually drawn using a stratified sample design. The strata are grades in the junior high school: 7, 8, and 9. Within strata, simple random samples are selected. Table 86.1 provides the number of students in each grade.

Table 86.1 Students in Grades

Grade	Number of Students
7	1,824
8	1,025
9	1,151
Total	**4,000**

In order to analyze this sample by using PROC SURVEYREG, you need to input the stratification information by creating a SAS data set with the information in Table 86.1. The following SAS statements create such a data set called StudentTotals:

```
data StudentTotals;
   input Grade _TOTAL_;
   datalines;
7 1824
8 1025
9 1151
;
```

The variable Grade is the stratification variable, and the variable _TOTAL_ contains the total numbers of students in each stratum in the survey population. PROC SURVEYREG requires you to use the keyword _TOTAL_ as the name of the variable that contains the population total information.

In a stratified sample design, when the sampling rates in the strata are unequal, you need to use sampling weights to reflect this information. For this example, the appropriate sampling weights are the reciprocals of the probabilities of selection. You can use the following DATA step to create the sampling weights:

```
data IceCream;
   set IceCream;
   if Grade=7 then Prob=20/1824;
   if Grade=8 then Prob=9/1025;
   if Grade=9 then Prob=11/1151;
   Weight=1/Prob;
```

If you use PROC SURVEYSELECT to select your sample, PROC SURVEYSELECT creates these sampling weights for you.

The following statements demonstrate how you can fit a linear model while incorporating the sample design information (stratification):

```
title1 'Ice Cream Spending Analysis';
title2 'Stratified Simple Random Sample Design';
proc surveyreg data=IceCream total=StudentTotals;
   strata Grade /list;
   class Kids;
   model Spending = Income Kids / solution;
   weight Weight;
run;
```

Comparing these statements to those in the section "Simple Random Sampling" on page 6535, you can see how the TOTAL=StudentTotals option replaces the previous TOTAL=4000 option.

The STRATA statement specifies the stratification variable Grade. The LIST option in the STRATA statement requests that the stratification information be included in the output. The WEIGHT statement specifies the weight variable.

Figure 86.4 summarizes the data information, the sample design information, and the fit information. Note that, due to the stratification, the denominator degrees of freedom for F tests and t tests are 37, which is different from the analysis in Figure 86.1.

Figure 86.4 Summary of the Regression

```
                    Ice Cream Spending Analysis
                 Stratified Simple Random Sample Design

                      The SURVEYREG Procedure

          Regression Analysis for Dependent Variable Spending

                          Data Summary

                Number of Observations               40
                Sum of Weights                   4000.0
                Weighted Mean of Spending       9.14130
                Weighted Sum of Spending        36565.2
```

Figure 86.4 *continued*

```
                    Design Summary

              Number of Strata            3

                    Fit Statistics

                R-square            0.8219
                Root MSE            2.4185
                Denominator DF          37
```

For each stratum, Figure 86.5 displays the value of identifying variables, the number of observations (sample size), the total population size, and the calculated sampling rate or fraction.

Figure 86.5 Stratification and Classification Information

```
                    Stratum Information

     Stratum                                Population    Sampling
      Index        Grade       N Obs            Total        Rate

        1              7          20             1824      1.10%
        2              8           9             1025      0.88%
        3              9          11             1151      0.96%

                    Class Level Information

                  Class
                  Variable      Levels    Values

                  Kids               4    1 2 3 4
```

Figure 86.6 displays the tests for the significance of model effects under the stratified sample design. The Income effect is strongly significant, while the Kids effect is not significant at the 5% level.

Figure 86.6 Testing Effects

```
                    Tests of Model Effects

              Effect       Num DF    F Value    Pr > F

              Model             4     124.85    <.0001
              Intercept         1     150.95    <.0001
              Income            1     326.89    <.0001
              Kids              3       0.99    0.4081

     NOTE: The denominator degrees of freedom for the F tests is 37.
```

The regression coefficient estimates for the stratified sample, along with their standard errors and associated *t* tests, are displayed in Figure 86.7.

Figure 86.7 Regression Coefficients

Estimated Regression Coefficients

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	-26.086882	2.44108058	-10.69	<.0001
Income	0.776699	0.04295904	18.08	<.0001
Kids 1	0.888631	1.07000634	0.83	0.4116
Kids 2	1.545726	1.20815863	1.28	0.2087
Kids 3	-0.526817	1.32748011	-0.40	0.6938
Kids 4	0.000000	0.00000000	.	.

NOTE: The denominator degrees of freedom for the t tests is 37.
Matrix X'WX is singular and a generalized inverse was used to solve the normal equations. Estimates are not unique.

You can request other statistics and tests by using PROC SURVEYREG. You can also analyze data from a more complex sample design. The remainder of this chapter provides more detailed information.

Output Data Sets

You can use the OUTPUT statement to create a new SAS data set that contains the estimated linear predictors and their standard error estimates, the residuals from the linear regression, and the confidence limits for the predictors. See the section "OUTPUT Statement" on page 6556 for more details.

You can use the Output Delivery System (ODS) to create SAS data sets that capture the outputs from PROC SURVEYREG. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

For example, to save the "ParameterEstimates" table (Figure 86.7) in the previous section in an output data set, you use the ODS OUTPUT statement as follows:

```
title1 'Ice Cream Spending Analysis';
title2 'Stratified Simple Random Sample Design';
proc surveyreg data=IceCream total=StudentTotals;
   strata Grade /list;
   class Kids;
   model Spending = Income Kids / solution;
   weight Weight;
   ods output ParameterEstimates = MyParmEst;
   run;
```

The statement

```
ods output ParameterEstimates = MyParmEst;
```

requests that the "ParameterEstimates" table that appears in Figure 86.7 be placed into a SAS data set MyParmEst.

The PRINT procedure displays observations of the data set MyParmEst:

```
proc print data=MyParmEst;
run;
```

Figure 86.8 displays the observations in the data set MyParmEst.

Figure 86.8 The Data Set MyParmEst

Ice Cream Spending Analysis
Stratified Simple Random Sample Design

Obs	Parameter	Estimate	StdErr	DenDF	tValue	Probt
1	Intercept	-26.086882	2.44108058	37	-10.69	<.0001
2	Income	0.776699	0.04295904	37	18.08	<.0001
3	Kids 1	0.888631	1.07000634	37	0.83	0.4116
4	Kids 2	1.545726	1.20815863	37	1.28	0.2087
5	Kids 3	-0.526817	1.32748011	37	-0.40	0.6938
6	Kids 4	0.000000	0.00000000	37	.	.

The section "ODS Table Names" on page 6581 gives the complete list of the tables produced by PROC SURVEYREG.

Syntax: SURVEYREG Procedure

The following statements are available in PROC SURVEYREG:

PROC SURVEYREG < *options* > ;
BY *variables* ;
CLASS *variables* ;
CLUSTER *variables* ;
CONTRAST *'label' effect values* < *... effect values* > < / *options* > ;
DOMAIN *variables* < *variable*∗*variable variable*∗*variable*∗*variable ...* > ;
ESTIMATE *'label' effect values* < *... effect values* > < / *options* > ;
MODEL *dependent* = < *effects* > < / *options* > ;
OUTPUT < *keyword* < =*variable-name* > *... keyword* < =*variable-name* > > < / *option* > ;
REPWEIGHTS *variables* < / *options* > ;
STRATA *variables* < / *options* > ;
WEIGHT *variable* ;

The PROC SURVEYREG and MODEL statements are required. If your model contains classification effects, you must list the classification variables in a CLASS statement, and the CLASS

statement must precede the MODEL statement. If you use a CONTRAST statement or an ESTIMATE statement, the MODEL statement must precede the CONTRAST or ESTIMATE statement.

The CLASS, CLUSTER, STRATA, CONTRAST, and ESTIMATE statements can appear multiple times. You should use only one MODEL statement and one WEIGHT statement.

PROC SURVEYREG Statement

PROC SURVEYREG < *options* > **;**

The PROC SURVEYREG statement invokes the procedure. It optionally names the input data sets and specifies the variance estimation method.

You can specify the following options in the PROC SURVEYREG statement.

ALPHA=*α*

sets the confidence level for confidence limits. The value of the ALPHA= option must be between 0 and 1, and the default value is 0.05. A confidence level of α produces $100(1-\alpha)\%$ confidence limits. The default of ALPHA=0.05 produces 95% confidence limits.

DATA=*SAS-data-set*

specifies the SAS data set to be analyzed by PROC SURVEYREG. If you omit the DATA= option, the procedure uses the most recently created SAS data set.

MISSING

treats missing values as a valid (nonmissing) category for all categorical variables, which include CLASS, STRATA, CLUSTER, and DOMAIN variables.

By default, if you do not specify the MISSING option, an observation is excluded from the analysis if it has a missing value. For more information, see the section "Missing Values" on page 6560.

NOMCAR

requests that the procedure treat missing values in the variance computation as *not missing completely at random* (NOMCAR) for Taylor series variance estimation. When you specify the NOMCAR option, PROC SURVEYREG computes variance estimates by analyzing the nonmissing values as a domain or subpopulation, where the entire population includes both nonmissing and missing domains. See the section "Missing Values" on page 6560 for more details.

By default, PROC SURVEYREG completely excludes an observation from analysis if that observation has a missing value, unless you specify the MISSING option. Note that the NOMCAR option has no effect on a classification variable when you specify the MISSING option, which treats missing values as a valid nonmissing level.

The NOMCAR option applies only to Taylor series variance estimation. The replication methods, which you request with the VARMETHOD=BRR and VARMETHOD=JACKKNIFE options, do not use the NOMCAR option.

ORDER=DATA | FORMATTED | FREQ | INTERNAL

specifies the sorting order for the levels of classification variables. This ordering determines which parameters in the model correspond to each level in the data, so the ORDER= option might be useful when you use the CONTRAST statement. When the default ORDER=FORMATTED is in effect for numeric variables for which you have supplied no explicit format, the levels are ordered by their internal values.

The following table shows how PROC SURVEYREG interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order
INTERNAL	unformatted value

Note that the specified order also determines the ordering for levels of STRATA and CLUSTER variables.

By default, ORDER=FORMATTED. For FORMATTED and INTERNAL, the sort order is machine dependent.

For more information about sorting order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

RATE=*value* | *SAS-data-set*

R=*value* | *SAS-data-set*

specifies the sampling rate as a nonnegative *value*, or specifies an input data set that contains the stratum sampling rates. The procedure uses this information to compute a finite population correction for Taylor series variance estimation. The procedure does not use the RATE= option for BRR or jackknife variance estimation, which you request with the VARMETHOD=BRR or VARMETHOD=JACKKNIFE option.

If your sample design has multiple stages, you should specify the *first-stage sampling rate*, which is the ratio of the number of PSUs selected to the total number of PSUs in the population.

For a nonstratified sample design, or for a stratified sample design with the same sampling rate in all strata, you should specify a nonnegative *value* for the RATE= option. If your design is stratified with different sampling rates in the strata, then you should name a SAS data set that contains the stratification variables and the sampling rates. See the section "Specification of Population Totals and Sampling Rates" on page 6561 for more details.

The *value* in the RATE= option or the values of _RATE_ in the secondary data set must be nonnegative numbers. You can specify *value* as a number between 0 and 1. Or you can

specify *value* in percentage form as a number between 1 and 100, and PROC SURVEYREG converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

If you do not specify the TOTAL= or RATE= option, then the Taylor series variance estimation does not include a finite population correction. You cannot specify both the TOTAL= and RATE= options.

TOTAL=*value* | *SAS-data-set*

N=*value* | *SAS-data-set*

specifies the total number of primary sampling units in the study population as a positive *value*, or specifies an input data set that contains the stratum population totals. The procedure uses this information to compute a finite population correction for Taylor series variance estimation. The procedure does not use the TOTAL= option for BRR or jackknife variance estimation, which you request with the VARMETHOD=BRR or VARMETHOD=JACKKNIFE option.

For a nonstratified sample design, or for a stratified sample design with the same population total in all strata, you should specify a positive *value* for the TOTAL= option. If your sample design is stratified with different population totals in the strata, then you should name a SAS data set that contains the stratification variables and the population totals. See the section "Specification of Population Totals and Sampling Rates" on page 6561 for more details.

If you do not specify the TOTAL= or RATE= option, then the Taylor series variance estimation does not include a finite population correction. You cannot specify both the TOTAL= and RATE= options.

TRUNCATE

specifies that class levels should be determined using no more than the first 16 characters of the formatted values of the CLASS, STRATA, and CLUSTER variables. When formatted values are longer than 16 characters, you can use this option in order to revert to the levels as determined in releases before SAS 9.

VARMETHOD=BRR < (*method-options***) >**

VARMETHOD=JACKKNIFE | JK < (*method-options***) >**

VARMETHOD=TAYLOR

specifies the variance estimation method. VARMETHOD=TAYLOR requests the Taylor series method, which is the default if you do not specify the VARMETHOD= option or the REPWEIGHTS statement. VARMETHOD=BRR requests variance estimation by balanced repeated replication (BRR), and VARMETHOD=JACKKNIFE requests variance estimation by the delete-1 jackknife method.

For VARMETHOD=BRR and VARMETHOD=JACKKNIFE you can specify *method-options* in parentheses. Table 86.2 summarizes the available *method-options*.

Table 86.2 Variance Estimation Method-Options

Keyword	Method	(*Method-Options*)
BRR	Balanced repeated replication	FAY <=*value*> HADAMARD | H=*SAS-data-set* OUTWEIGHTS=*SAS-data-set* PRINTH REPS=*number*
JACKKNIFE | JK	Jackknife	OUTJKCOEFS=*SAS-data-set* OUTWEIGHTS=*SAS-data-set*
TAYLOR	Taylor series	

Method-options must be enclosed in parentheses following the method keyword. For example:

```
varmethod=BRR(reps=60 outweights=myReplicateWeights)
```

The following values are available for the VARMETHOD= option:

BRR <(*method-options*)> requests balanced repeated replication (BRR) variance estimation. The BRR method requires a stratified sample design with two primary sampling units (PSUs) per stratum. See the section "Balanced Repeated Replication (BRR) Method" on page 6565 for more information.

You can specify the following *method-options* in parentheses following VARMETHOD=BRR.

FAY <=*value***>**
requests Fay's method, a modification of the BRR method, for variance estimation. See the section "Fay's BRR Method" on page 6566 for more information.

You can specify the *value* of the Fay coefficient, which is used in converting the original sampling weights to replicate weights. The Fay coefficient must be a nonnegative number less than 1. By default, the value of the Fay coefficient equals 0.5.

HADAMARD=*SAS-data-set*

H=*SAS-data-set*
names a SAS data set that contains the Hadamard matrix for BRR replicate construction. If you do not provide a Hadamard matrix with the HADAMARD= method-option, PROC SURVEYREG generates an appropriate Hadamard matrix for replicate construction. See the sections "Balanced Repeated Replication (BRR) Method" on page 6565 and "Hadamard Matrix" on page 6568 for details.

If a Hadamard matrix of a given dimension exists, it is not necessarily unique. Therefore, if you want to use a specific Hadamard

matrix, you must provide the matrix as a SAS data set in the HADAMARD=*SAS-data-set* method-option.

In the HADAMARD= input data set, each variable corresponds to a column of the Hadamard matrix, and each observation corresponds to a row of the matrix. You can use any variable names in the HADAMARD= data set. All values in the data set must equal either 1 or -1. You must ensure that the matrix you provide is indeed a Hadamard matrix—that is, $\mathbf{A}'\mathbf{A} = R\mathbf{I}$, where $\mathbf{A}$ is the Hadamard matrix of dimension R and $\mathbf{I}$ is an identity matrix. PROC SURVEYREG does not check the validity of the Hadamard matrix that you provide.

The HADAMARD= input data set must contain at least H variables, where H denotes the number of first-stage strata in your design. If the data set contains more than H variables, the procedure uses only the first H variables. Similarly, the HADAMARD= input data set must contain at least H observations.

If you do not specify the REPS= method-option, then the number of replicates is taken to be the number of observations in the HADAMARD= input data set. If you specify the number of replicates—for example, REPS=*nreps*—then the first *nreps* observations in the HADAMARD= data set are used to construct the replicates.

You can specify the PRINTH option to display the Hadamard matrix that the procedure uses to construct replicates for BRR.

OUTWEIGHTS=*SAS-data-set*

names a SAS data set that contains replicate weights. See the section "Balanced Repeated Replication (BRR) Method" on page 6565 for information about replicate weights. See the section "Replicate Weights Output Data Set" on page 6575 for more details about the contents of the OUTWEIGHTS= data set.

The OUTWEIGHTS= method-option is not available when you provide replicate weights with the REPWEIGHTS statement.

PRINTH

displays the Hadamard matrix.

When you provide your own Hadamard matrix with the HADAMARD= method-option, only the rows and columns of the Hadamard matrix that are used by the procedure are displayed. See the sections "Balanced Repeated Replication (BRR) Method" on page 6565 and "Hadamard Matrix" on page 6568 for details.

The PRINTH method-option is not available when you provide replicate weights with the REPWEIGHTS statement because the procedure does not use a Hadamard matrix in this case.

REPS=*number*

specifies the number of replicates for BRR variance estimation. The value of *number* must be an integer greater than 1.

If you do not provide a Hadamard matrix with the HADAMARD= method-option, the number of replicates should be greater than the number of strata and should be a multiple of 4. See the section "Balanced Repeated Replication (BRR) Method" on page 6565 for more information. If a Hadamard matrix cannot be constructed for the REPS= value that you specify, the value is increased until a Hadamard matrix of that dimension can be constructed. Therefore, it is possible for the actual number of replicates used to be larger than the REPS= value that you specify.

If you provide a Hadamard matrix with the HADAMARD= method-option, the value of REPS= must not be less than the number of rows in the Hadamard matrix. If you provide a Hadamard matrix and do not specify the REPS= method-option, the number of replicates equals the number of rows in the Hadamard matrix.

If you do not specify the REPS= or HADAMARD= method-option and do not include a REPWEIGHTS statement, the number of replicates equals the smallest multiple of 4 that is greater than the number of strata.

If you provide replicate weights with the REPWEIGHTS statement, the procedure does not use the REPS= method-option. With a REPWEIGHTS statement, the number of replicates equals the number of REPWEIGHTS variables.

JACKKNIFE | JK <(*method-options*)> requests variance estimation by the delete-1 jackknife method. See the section "Jackknife Method" on page 6567 for details. If you provide replicate weights with a REPWEIGHTS statement, VARMETHOD=JACKKNIFE is the default variance estimation method.

You can specify the following *method-options* in parentheses following VARMETHOD=JACKKNIFE:

OUTWEIGHTS=*SAS-data-set*

names a SAS data set that contains replicate weights. See the section "Jackknife Method" on page 6567 for information about replicate weights. See the section "Replicate Weights Output Data Set" on page 6575 for more details about the contents of the OUTWEIGHTS= data set.

The OUTWEIGHTS= method-option is not available when you provide replicate weights with the REPWEIGHTS statement.

OUTJKCOEFS=*SAS-data-set*

names a SAS data set that contains jackknife coefficients. See the section "Jackknife Coefficients Output Data Set" on page 6575 for more details about the contents of the OUTJKCOEFS= data set.

TAYLOR requests Taylor series variance estimation. This is the default method if you do not specify the VARMETHOD= option and if there is no REPWEIGHTS statement. See the section "Taylor Series (Linearization)" on page 6564 for more information.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC SURVEYREG to obtain separate analyses on observations in groups defined by the BY variables.

Note that using a BY statement provides completely separate analyses of the BY groups. It does not provide a statistically valid subpopulation or domain analysis, where the total number of units in the subpopulation is not known with certainty. For more information about subpopulation analysis for sample survey data, see Cochran (1977).

When a BY statement appears, the procedure expects the input data sets to be sorted in order of the BY variables. If you specify more than one BY statement, the procedure uses only the latest BY statement and ignores any previous ones.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING. The NOTSORTED option does not mean that the data are unsorted, but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

The CLASS statement specifies the classification variables to be used in the model. Typical classification variables are TREATMENT, GENDER, RACE, GROUP, and REPLICATION. If you specify the CLASS statement, it must appear before the MODEL statement.

Classification variables can be either character or numeric. Class levels are determined from the formatted values of the CLASS variables. Thus, you can use formats to group values into levels.

See the FORMAT procedure in the *Base SAS Procedures Guide* and the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary* for more information.

By default, class levels are determined from the entire formatted values of the CLASS variables. Note that this represents a slight change from previous releases in the way in which class levels are determined. Prior to SAS 9, class levels were determined by using no more than the first 16 characters of the formatted values. If you want to revert to this previous behavior, you can use the TRUNCATE option in the PROC SURVEYREG statement.

You can use multiple CLASS statements to specify classification variables.

CLUSTER Statement

CLUSTER *variables* **;**

The CLUSTER statement specifies variables that identify clusters in a clustered sample design. The combinations of categories of CLUSTER variables define the clusters in the sample. If there is a STRATA statement, clusters are nested within strata.

If your sample design has clustering at multiple stages, you should identify only the first-stage clusters, or primary sampling units (PSUs), in the CLUSTER statement.

If you provide replicate weights for BRR or jackknife variance estimation with the REPWEIGHTS statement, you do not need to specify a CLUSTER statement.

The CLUSTER *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric. The formatted values of the CLUSTER variables determine the CLUSTER variable levels. Thus, you can use formats to group values into levels. See the FORMAT procedure in the *Base SAS Procedures Guide* and the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary* for more information.

By default, clusters are determined from the entire formatted values of the CLUSTER variables. Note that this represents a slight change from previous releases in the way in which clusters are determined. Prior to SAS 9, clusters were determined by using no more than the first 16 characters of the formatted values. If you want to revert to this previous behavior, you can use the TRUNCATE option in the PROC SURVEYREG statement.

You can use multiple CLUSTER statements to specify cluster variables. The procedure uses variables from all CLUSTER statements to create clusters.

CONTRAST Statement

CONTRAST *'label' effect values* < */ options* > **;**

CONTRAST *'label' effect values* < *. . . effect values* > < */ options* > **;**

The CONTRAST statement provides custom hypothesis tests for linear combinations of the regression parameters $H_0: \mathbf{L}\boldsymbol{\beta} = \mathbf{0}$, where $\mathbf{L}$ is the vector or matrix you specify and $\boldsymbol{\beta}$ is the vector of regression parameters. Thus, to use this feature, you must be familiar with the details of the model parameterization used by PROC SURVEYREG. For information about the parameterization, see the section "GLM Parameterization of Classification Variables and Effects" on page 369 in Chapter 18, "Shared Concepts and Topics."

Each term in the MODEL statement, called an *effect*, is a variable or a combination of variables. You can specify an effect with a variable name or a special notation by using variable names and operators. For more details about how to specify an effect, see the section "Specification of Effects" on page 2488 in Chapter 39, "The GLM Procedure."

For each CONTRAST statement, PROC SURVEYREG computes Wald's F test. The procedure displays this value with the degrees of freedom, and identifies it with the contrast label. The numerator degrees of freedom for Wald's F test equal rank($\mathbf{L}$). The denominator degrees of freedom equal the number of clusters (or the number of observations if there is no CLUSTER statement) minus the number of strata. Alternatively, you can use the DF= option in the MODEL statement to specify the denominator degrees of freedom.

You can specify any number of CONTRAST statements, but they must appear after the MODEL statement.

In the CONTRAST statement,

label	identifies the contrast in the output. A label is required for every contrast specified. Labels must be enclosed in single quotes.
effect	identifies an effect that appears in the MODEL statement. You can use the INTERCEPT keyword as an effect when an intercept is fitted in the model. You do not need to include all effects that are in the MODEL statement.
values	are constants that are elements of $\mathbf{L}$ associated with the effect.

You can specify the following options in the CONTRAST statement after a slash (/):

E

displays the entire coefficient $\mathbf{L}$ vector or matrix.

NOFILL

requests no filling in higher-order effects. When you specify only certain portions of $\mathbf{L}$, by default PROC SURVEYREG constructs the remaining elements from the context. (For more information, see the section "Specification of ESTIMATE Expressions" on page 2509 in Chapter 39, "The GLM Procedure.")

When you specify the NOFILL option, PROC SURVEYREG does not construct the remaining portions and treats the vector or matrix **L** as it is defined in the CONTRAST statement.

SINGULAR=*value*

tunes the estimability checking. If **v** is a vector, define ABS(**v**) to be the largest absolute value of the elements of **v**. For a row vector **l** of the matrix **L** , define

$$c = \begin{cases} \text{ABS}(\mathbf{l}) & \text{if ABS}(\mathbf{l}) > 0 \\ 1 & \text{otherwise} \end{cases}$$

If ABS(**l**−**lH**) is greater than *c*value*, then $\mathbf{l}\boldsymbol{\beta}$ is declared nonestimable. Here, **H** is the matrix $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$. The *value* must be between 0 and 1; the default is 10^{-4}.

As stated previously, the CONTRAST statement enables you to perform hypothesis tests $H_0: \mathbf{L}\boldsymbol{\beta} = 0$.

If the **L** matrix contains more than one contrast, then you can separate the rows of the **L** matrix with commas.

For example, for the model

```
proc surveyreg;
   class A B;
   model Y=A B;
run;
```

with A at 5 levels and B at 2 levels, the parameter vector is

$$(\mu \;\; \alpha_1 \;\; \alpha_2 \;\; \alpha_3 \;\; \alpha_4 \;\; \alpha_5 \;\; \beta_1 \;\; \beta_2)$$

To test the hypothesis that the pooled A linear and A quadratic effect is zero, you can use the following **L** matrix:

$$\mathbf{L} = \begin{bmatrix} 0 & -2 & -1 & 0 & 1 & 2 & 0 & 0 \\ 0 & 2 & -1 & -2 & -1 & 2 & 0 & 0 \end{bmatrix}$$

The corresponding CONTRAST statement is

```
contrast 'A Linear & Quadratic'
         a -2 -1  0  1  2,
         a  2 -1 -2 -1  2;
```

DOMAIN Statement

DOMAIN *variables* < *variable** *variable variable** *variable** *variable ...* > **;**

The DOMAIN statement requests analysis for subpopulations, or domains, in addition to analysis for the entire study population. The DOMAIN statement names the variables that identify domains, which are called domain variables.

It is common practice to compute statistics for domains. The formation of these domains might be unrelated to the sample design. Therefore, the sample sizes for the domains are random variables. In order to incorporate this variability into the variance estimation, you should use a DOMAIN statement.

Note that a DOMAIN statement is different from a BY statement. In a BY statement, you treat the sample sizes as fixed in each subpopulation, and you perform analysis within each BY group independently.

You should use the DOMAIN statement on the entire data set to perform the domain analysis. Creating a new data set from a single domain and analyzing that with SURVEYREG yields inappropriate estimates of variance.

A domain variable can be either character or numeric. The procedure treats domain variables as categorical variables. If a variable appears by itself in a DOMAIN statement, each level of this variable determines a domain in the study population. If two or more variables are joined by asterisks (*), then every possible combination of levels of these variables determines a domain. The procedure performs a descriptive analysis within each domain defined by the domain variables.

The formatted values of the domain variables determine the categorical variable levels. Thus, you can use formats to group values into levels. See the FORMAT procedure in the *Base SAS Procedures Guide* and the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary* for more information.

ESTIMATE Statement

ESTIMATE *'label' effect values < / options >* **;**

ESTIMATE *'label' effect values < ... effect values > < / options >* **;**

You can use an ESTIMATE statement to estimate a linear function of the regression parameters by multiplying a row vector $\mathbf{l}$ by the parameter estimate vector $\hat{\boldsymbol{\beta}}$.

Each term in the MODEL statement, called an *effect*, is a variable or a combination of variables. You can specify an effect with a variable name or with a special notation by using variable names and operators.

For more details about how to specify an effect, see the section "Specification of Effects" on page 2488 in Chapter 39, "The GLM Procedure."

PROC SURVEYREG checks the linear function for estimability. (See the SINGULAR= option).

The procedure displays the estimate $\mathbf{l}\hat{\boldsymbol{\beta}}$ along with its standard error and t test. If you specify the CLPARM option in the MODEL statement, PROC SURVEYREG also displays confidence limits for the linear function. By default, the degrees of freedom for the t test equal the number of clusters (or the number of observations if there is no CLUSTER statement) minus the number of strata. Alternatively, you can specify the degrees of freedom with the DF= option in the MODEL statement.

You can specify any number of ESTIMATE statements, but they must appear after the MODEL statement.

In the ESTIMATE statement,

label identifies the linear function **l** in the output. A label is required for every function specified. Labels must be enclosed in single quotes.

effect identifies an effect that appears in the MODEL statement. You can use the INTERCEPT keyword as an effect when an intercept is fitted in the model. You do not need to include all effects that are in the MODEL statement.

values are constants that are elements of the vector **l** associated with the effect. For example, the following statement forms an estimate that is the difference between the parameters estimated for the first and second levels of the CLASS variable A:

```
estimate 'A1 vs A2' A  1 -1;
```

You can specify the following options in the ESTIMATE statement after a slash (/):

DIVISOR=*value*

specifies a value by which to divide all coefficients so that fractional coefficients can be entered as integers.

For example, you can use

```
estimate '1/3(A1+A2) - 2/3A3' a 1 1 -2 / divisor=3;
```

instead of

```
estimate '1/3(A1+A2) - 2/3A3' a 0.33333 0.33333 -0.66667;
```

E

displays the entire coefficient vector **l**.

NOFILL

requests no filling in higher-order effects. When you specify only certain portions of the vector **l**, by default PROC SURVEYREG constructs the remaining elements from the context. (See the section "Specification of ESTIMATE Expressions" on page 2509 in Chapter 39, "The GLM Procedure.") When you specify the NOFILL option, PROC SURVEYREG does not construct the remaining portions and treats the vector **l** as it is defined in the ESTIMATE statement.

SINGULAR=*value*

tunes the estimability checking. If **v** is a vector, define ABS(**v**) to be the largest absolute value of the elements of **v**. For a row vector **l**, define

$$c = \begin{cases} \text{ABS}(\mathbf{l}) & \text{if ABS}(\mathbf{l}) > 0 \\ 1 & \text{otherwise} \end{cases}$$

If ABS($\mathbf{l}-\mathbf{lH}$) is greater than c*$value$, then $\mathbf{l}\boldsymbol{\beta}$ is declared nonestimable. Here, $\mathbf{H}$ is the matrix $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$. The *value* must be between 0 and 1; the default is 10^{-4}.

MODEL Statement

MODEL *dependent* = < *effects* > < / *options* > **;**

The MODEL statement specifies the dependent (response) variable and the independent (regressor) variables or effects. The dependent variable must be numeric. Each term in a MODEL statement, called an *effect*, is a variable or a combination of variables. You can specify an effect with a variable name or with special notation by using variable names and operators. For more information about how to specify an effect, see the section "Specification of Effects" on page 2488 in Chapter 39, "The GLM Procedure."

Only one MODEL statement is allowed for each PROC SURVEYREG statement. If you specify more than one MODEL statement, the procedure uses the first model and ignores the rest.

You can specify the following options in the MODEL statement after a slash (/).

ADJRSQ
requests the procedure compute the adjusted multiple R-square.

ANOVA
requests the ANOVA table be produced in the output. By default, the ANOVA table is not printed in the output.

CLPARM
requests confidence limits for the parameter estimates. The SURVEYREG procedure determines the confidence coefficient by using the ALPHA= option, which by default equals 0.05 and produces 95% confidence bounds. The CLPARM option also requests confidence limits for all the estimable linear functions of regression parameters in the ESTIMATE statements.

Note that when there is a CLASS statement, you need to use the SOLUTION option with the CLPARM option to obtain the parameter estimates and their confidence limits.

COVB
displays the estimated covariance matrix of the estimated regression estimates.

DEFF
displays design effects for the regression coefficient estimates.

DF=*value*
specifies the denominator degrees of freedom for the F tests and the degrees of freedom for the t tests. The default is the number of clusters (or the number of observations if there is no CLUSTER statement) minus the number of actual strata. The number of actual strata equals the number of strata in the data before collapsing minus the number of strata collapsed plus 1. See the section "Stratum Collapse" on page 6572 for details about "collapsing of strata."

I | INVERSE
displays the inverse or the generalized inverse of the $\mathbf{X}'\mathbf{X}$ matrix. When there is a WEIGHT variable, the procedure displays the inverse or the generalized inverse of the $\mathbf{X}'\mathbf{W}\mathbf{X}$ matrix, where $\mathbf{W}$ is the diagonal matrix constructed from WEIGHT variable values.

NOINT
omits the intercept from the model.

PARMLABEL
displays the labels of the parameters in the "Estimated Regression Coefficients" table.

SINGULAR=*value*
tunes the estimability checking. If $\mathbf{v}$ is a vector, define ABS($\mathbf{v}$) to be the largest absolute value of the elements of $\mathbf{v}$. For a row vector $\mathbf{l}$ of the matrix $\mathbf{L}$, define

$$c = \begin{cases} \text{ABS}(\mathbf{l}) & \text{if ABS}(\mathbf{l}) > 0 \\ 1 & \text{otherwise} \end{cases}$$

If ABS($\mathbf{l}-\mathbf{lH}$) is greater than c*$value$, then $\mathbf{l}\boldsymbol{\beta}$ is declared nonestimable. Here, $\mathbf{H}$ is the matrix $(\mathbf{X}'\mathbf{X})^{-}\mathbf{X}'\mathbf{X}$. The *value* must be between 0 and 1; the default is 10^{-4}.

SOLUTION
displays a solution to the normal equations, which are the parameter estimates. The SOLUTION option is useful only when you use a CLASS statement. If you do not specify a CLASS statement, PROC SURVEYREG displays parameter estimates by default. But if you specify a CLASS statement, PROC SURVEYREG does not display parameter estimates unless you also specify the SOLUTION option.

VADJUST=DF | NONE
specifies whether to use degrees of freedom adjustment $(n-1)/(n-p)$ in the computation of the matrix $\mathbf{G}$ for the variance estimation. If you do not specify the VADJUST= option, by default, PROC SURVEYREG uses the degrees-of-freedom adjustment that is equivalent to the VARADJ=DF option. If you do not want to use this variance adjustment, you can specify the VADJUST=NONE option.

X | XPX
displays the $\mathbf{X}'\mathbf{X}$ matrix, or the $\mathbf{X}'\mathbf{W}\mathbf{X}$ matrix when there is a WEIGHT variable, where $\mathbf{W}$ is the diagonal matrix constructed from WEIGHT variable values. The X option also displays the crossproducts vector $\mathbf{X}'\mathbf{y}$ or $\mathbf{X}'\mathbf{W}\mathbf{y}$.

OUTPUT Statement

OUTPUT < **OUT=***SAS-data-set* > < *keyword* < =*variable-name* > ... *keyword* < =*variable-name* > > < / *option* > ;

The OUTPUT statement creates a new SAS data set that contains all the variables in the input data set and, optionally, the estimated linear predictors and their standard error estimates, the residuals from the linear regression, and the confidence limits for the predictors.

You can specify the following options in the OUTPUT statement.

OUT=*SAS-data-set*
gives the name of the new output data set. By default, the procedure uses the DATA*n* convention to name the new data set.

keyword < =variable-name >

specifies the statistics to include in the output data set and names the new variables that contain the statistics. You can specify a keyword for each desired statistic (see the following list of keywords). Optionally, you can name a statistic by providing a variable name followed an equal sign to contain the statistic. For example,

```
output out=myOutDataSet p=myPredictor;
```

creates a SAS data set myOutDataSet that contains the predicted values in the variable myPredictor.

The keywords allowed and the statistics they represent are as follows:

LCLM | L — lower bound of a $100(1-\alpha)\%$ confidence interval for the expected value (mean) of the predicted value. The α level is equal to the value of the ALPHA= option in the OUTPUT statement or, if this option is not specified, to the ALPHA= option in the PROC SURVEYREG statement. If neither of these options is set, then $\alpha = 0.05$ by default, resulting in the lower bound for a 95% confidence interval. If no variable name is given for this keyword, the default variable name is _LCLM_.

PREDICTED | PRED | P — predicted values. If no variable name is given for this keyword, the default variable name is _PREDICTED_.

RESIDUAL | R — residuals, calculated as ACTUAL − PREDICTED. If no variable name is given for this keyword, the default variable name is _RESIDUAL_.

STDP | STD — standard error of the mean predicted value. If no variable name is given for this keyword, the default variable name is _STD_.

UCLM | U — upper bound of a $100(1-\alpha)\%$ confidence interval for the expected value (mean) of the predicted value. The α level is equal to the value of the ALPHA= option in the OUTPUT statement or, if this option is not specified, to the ALPHA= option in the PROC SURVEYREG statement. If neither of these options is set, then $\alpha = 0.05$ by default, resulting in the upper bound for a 95% confidence interval. If no variable name is given for this keyword, the default variable name is _UCLM_.

The following option is available in the OUTPUT statement and is specified after a slash(/):

ALPHA=α

specifies the level of significance α for $100(1-\alpha)\%$ confidence intervals. By default, α is equal to the value of the ALPHA= option in the PROC SURVEYREG statement or 0.05 if that option is not specified. You can use values between 0 and 1.

REPWEIGHTS Statement

REPWEIGHTS *variables* < / *options* > **;**

The REPWEIGHTS statement names variables that provide replicate weights for BRR or jackknife variance estimation, which you request with the VARMETHOD=BRR or VARMETHOD=JACKKNIFE option in the PROC SURVEYREG statement. If you do not provide replicate weights for these methods by using a REPWEIGHTS statement, then the procedure constructs replicate weights for the analysis. See the sections "Balanced Repeated Replication (BRR) Method" on page 6565 and "Jackknife Method" on page 6567 for information about replicate weights.

Each REPWEIGHTS variable should contain the weights for a single replicate, and the number of replicates equals the number of REPWEIGHTS variables. The REPWEIGHTS variables must be numeric, and the variable values must be nonnegative numbers.

If you provide replicate weights with a REPWEIGHTS statement, you do not need to specify a CLUSTER or STRATA statement. If you use a REPWEIGHTS statement and do not specify the VARMETHOD= option in the PROC SURVEYREG statement, the procedure uses VARMETHOD=JACKKNIFE by default.

If you specify a REPWEIGHTS statement but do not include a WEIGHT statement, the procedure uses the average of each observation's replicate weights as the observation's weight.

You can specify the following options in the REPWEIGHTS statement after a slash (/):

DF=*df*

specifies the denominator degrees of freedom for various tests in the regression analysis. The value of *df* must be a positive number. See the section "Degrees of Freedom" on page 6568 for details. By default, the denominator degrees of freedom equals the number of REPWEIGHTS variables.

JKCOEFS=*value*

specifies a jackknife coefficient for VARMETHOD=JACKKNIFE. The coefficient *value* must be a nonnegative number less than one. See the section "Jackknife Method" on page 6567 for details about jackknife coefficients.

You can use this option to specify a single value of the jackknife coefficient, which the procedure uses for all replicates. To specify different coefficients for different replicates, use the JKCOEFS=*values* or JKCOEFS=*SAS-data-set* option.

JKCOEFS=*values*

specifies jackknife coefficients for VARMETHOD=JACKKNIFE, where each coefficient corresponds to an individual replicate identified by a REPWEIGHTS variable. You can separate *values* with blanks or commas. The coefficient *values* must be nonnegative numbers less than one. The number of *values* must equal the number of replicate weight variables named in the REPWEIGHTS statement. List these values in the same order in which you list the corresponding replicate weight variables in the REPWEIGHTS statement.

See the section "Jackknife Method" on page 6567 for details about jackknife coefficients.

To specify different coefficients for different replicates, you can also use the JKCOEFS=*SAS-data-set* option. To specify a single jackknife coefficient for all replicates, use the JKCOEFS=*value* option.

JKCOEFS=*SAS-data-set*

names a SAS data set that contains the jackknife coefficients for VARMETHOD=JACKKNIFE. You provide the jackknife coefficients in the JKCOEFS= data set variable JKCoefficient. Each coefficient value must be a nonnegative number less than one. The observations in the JKCOEFS= data set should correspond to the replicates that are identified by the REPWEIGHTS variables. Arrange the coefficients or observations in the JKCOEFS= data set in the same order in which you list the corresponding replicate weight variables in the REPWEIGHTS statement. The number of observations in the JKCOEFS= data set must not be less than the number of REPWEIGHTS variables.

See the section "Jackknife Method" on page 6567 for details about jackknife coefficients.

To specify different coefficients for different replicates, you can also use the JKCOEFS=*values* option. To specify a single jackknife coefficient for all replicates, use the JKCOEFS=*value* option.

STRATA Statement

STRATA *variables* < */options* > **;**

The STRATA statement specifies variables that form the strata in a stratified sample design. The combinations of categories of STRATA variables define the strata in the sample.

If your sample design has stratification at multiple stages, you should identify only the first-stage strata in the STRATA statement. See the section "Specification of Population Totals and Sampling Rates" on page 6561 for more information.

If you provide replicate weights for BRR or jackknife variance estimation with the REPWEIGHTS statement, you do not need to specify a STRATA statement.

The STRATA *variables* are one or more variables in the DATA= input data set. These variables can be either character or numeric. By default, strata are determined from the entire formatted values of the STRATA variables. Note that this represents a slight change from previous releases in the way in which strata are determined. Prior to SAS 9, strata were determined by using no more than the first 16 characters of the formatted values. If you want to revert to this previous behavior, you can use the TRUNCATE option in the PROC SURVEYREG statement.

Thus, you can use formats to group values into levels. See the FORMAT procedure in the *Base SAS Procedures Guide* and the FORMAT statement and SAS formats in *SAS Language Reference: Dictionary* for more information.

You can use multiple STRATA statements to specify stratum variables.

You can specify the following options in the STRATA statement after a slash (/):

LIST

displays a "Stratum Information" table, which includes values of the STRATA variables, and the number of observations, number of clusters, population total, and sampling rate for each stratum. This table also displays stratum collapse information.

NOCOLLAPSE

prevents the procedure from collapsing, or combining, strata that have only one sampling unit for the Taylor series variance estimation. By default, the procedure collapses strata that contain only one sampling unit for the Taylor series method. See the section "Stratum Collapse" on page 6572 for details.

WEIGHT Statement

WEIGHT | WGT *variable* **;**

The WEIGHT statement names the variable that contains the sampling weights. This variable must be numeric, and the sampling weights must be positive numbers. If an observation has a weight that is nonpositive or missing, then the procedure omits that observation from the analysis. See the section "Missing Values" on page 6560 for more information. If you specify more than one WEIGHT statement, the procedure uses only the first WEIGHT statement and ignores the rest.

If you do not specify a WEIGHT statement but provide replicate weights with a REPWEIGHTS statement, PROC SURVEYREG uses the average of each observation's replicate weights as the observation's weight.

If you do not specify a WEIGHT statement or a REPWEIGHTS statement, PROC SURVEYREG assigns all observations a weight of one.

Details: SURVEYREG Procedure

Missing Values

If you have missing values in your survey data for any reason, such as nonresponse, this can compromise the quality of your survey results. If the respondents are different from the nonrespondents with regard to a survey effect or outcome, then survey estimates might be biased and cannot accurately represent the survey population. There are a variety of techniques in sample design and survey operations that can reduce nonresponse. After data collection is complete, you can use imputation to replace missing values with acceptable values, and/or you can use sampling weight adjustments to compensate for nonresponse. You should complete this data preparation and adjustment before you analyze your data with PROC SURVEYREG. See Cochran (1977), Kalton and Kaspyzyk (1986), and Brick and Kalton (1996) for more information.

If an observation has a missing value or a nonpositive value for the WEIGHT variable, then that observation is excluded from the analysis.

An observation is also excluded if it has a missing value for design variables such as STRATA variables, CLUSTER variables, and DOMAIN variables, unless missing values are regarded as a legitimate categorical level for these variables, as specified by the MISSING option.

By default, if an observation contains missing values for the dependent variable or for any variable used in the independent effects, the observation is excluded from the analysis. This treatment is based on the assumption that the missing values are missing completely at random (MCAR). However, this assumption sometimes is not true. For example, evidence from other surveys might suggest that observations with missing values are systematically different from observations without missing values. If you believe that missing values are not missing completely at random, then you can specify the NOMCAR option to include these observations with missing values in the dependent variable and the independent variables in the variance estimation.

Whether or not the NOMCAR option is used, observations with missing or invalid values for WEIGHT, STRATA, CLUSTER, or DOMAIN variables are always excluded, unless the MISSING option is also specified.

When you specify the NOMCAR option, the procedure treats observations with and without missing values for variables in the regression model as two different domains, and it performs a domain analysis in the domain of nonmissing observations.

If you use a REPWEIGHTS statement, all REPWEIGHTS variables must contain nonmissing values.

Survey Design Information

Specification of Population Totals and Sampling Rates

To include a finite population correction (*fpc*) in Taylor series variance estimation, you can input either the sampling rate or the population total by using the RATE= or TOTAL= option in the PROC SURVEYREG statement. (You cannot specify both of these options in the same PROC SURVEYREG statement.) The RATE= and TOTAL= options apply only to Taylor series variance estimation. The procedure does not use a finite population correction for BRR or jackknife variance estimation.

If you do not specify the RATE= or TOTAL= option, the Taylor series variance estimation does not include a finite population correction. For fairly small sampling fractions, it is appropriate to ignore this correction. See Cochran (1977) and Kish (1965) for more information.

If your design has multiple stages of selection and you are specifying the RATE= option, you should input the first-stage sampling rate, which is the ratio of the number of PSUs in the sample to the total number of PSUs in the study population. If you are specifying the TOTAL= option for a multistage design, you should input the total number of PSUs in the study population. See the section "Primary Sampling Units (PSUs)" on page 6562 for more details.

For a nonstratified sample design, or for a stratified sample design with the same sampling rate or the same population total in all strata, you can use the RATE=*value* or TOTAL=*value* option. If your sample design is stratified with different sampling rates or population totals in different strata, use the RATE=*SAS-data-set* or TOTAL=*SAS-data-set* option to name a SAS data set that contains the stratum sampling rates or totals. This data set is called a *secondary data set*, as opposed to the *primary data set* that you specify with the DATA= option.

The secondary data set must contain all the stratification variables listed in the STRATA statement and all the variables in the BY statement. If there are formats associated with the STRATA variables and the BY variables, then the formats must be consistent in the primary and the secondary data sets. If you specify the TOTAL=*SAS-data-set* option, the secondary data set must have a variable named _TOTAL_ that contains the stratum population totals. Or if you specify the RATE=*SAS-data-set* option, the secondary data set must have a variable named _RATE_ that contains the stratum sampling rates. If the secondary data set contains more than one observation for any one stratum, then the procedure uses the first value of _TOTAL_ or _RATE_ for that stratum and ignores the rest.

The *value* in the RATE= option or the values of _RATE_ in the secondary data set must be nonnegative numbers. You can specify *value* as a number between 0 and 1. Or you can specify *value* in percentage form as a number between 1 and 100, and PROC SURVEYREG converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

If you specify the TOTAL=*value* option, *value* must not be less than the sample size. If you provide stratum population totals in a secondary data set, these values must not be less than the corresponding stratum sample sizes.

Primary Sampling Units (PSUs)

When you have clusters, or primary sampling units (PSUs), in your sample design, the procedure estimates variance from the variation among PSUs when the Taylor series variance method is used. See the section "Variance Estimation" on page 6564 for more information.

BRR or jackknife variance estimation methods draw multiple replicates (or subsamples) from the full sample by following a specific resampling scheme. These subsamples are constructed by deleting PSUs from the full sample.

If you use a REPWEIGHTS statement to provide replicate weights for BRR or jackknife variance estimation, you do not need to specify a CLUSTER statement. Otherwise, you should specify a CLUSTER statement whenever your design includes clustering at the first stage of sampling. If you do not specify a CLUSTER statement, then PROC SURVEYREG treats each observation as a PSU.

Computational Details

Notation

For a stratified clustered sample design, observations are represented by an $n \times (p+2)$ matrix

$$(\mathbf{w}, \mathbf{y}, \mathbf{X}) = (w_{hij}, y_{hij}, \mathbf{x}_{hij})$$

where

- $\mathbf{w}$ denotes the sampling weight vector
- $\mathbf{y}$ denotes the dependent variable
- $\mathbf{X}$ denotes the $n \times p$ design matrix. (When an effect contains only classification variables, the columns of $\mathbf{X}$ that correspond this effect contain only 0s and 1s; no reparameterization is made.)
- $h = 1, 2, \ldots, H$ is the stratum index
- $i = 1, 2, \ldots, n_h$ is the cluster index within stratum h
- $j = 1, 2, \ldots, m_{hi}$ is the unit index within cluster i of stratum h
- p is the total number of parameters (including an intercept if the INTERCEPT effect is included in the MODEL statement)
- $n = \sum_{h=1}^{H} \sum_{i=1}^{n_h} m_{hi}$ is the total number of observations in the sample

Also, f_h denotes the sampling rate for stratum h. You can use the TOTAL= or RATE= option to input population totals or sampling rates. See the section "Specification of Population Totals and Sampling Rates" on page 6561 for details. If you input stratum totals, PROC SURVEYREG computes f_h as the ratio of the stratum sample size to the stratum total. If you input stratum sampling rates, PROC SURVEYREG uses these values directly for f_h. If you do not specify the TOTAL= or RATE= option, then the procedure assumes that the stratum sampling rates f_h are negligible, and a finite population correction is not used when computing variances.

Regression Coefficients

PROC SURVEYREG solves the normal equations $\mathbf{X}'\mathbf{W}\mathbf{X}\boldsymbol{\beta} = \mathbf{X}'\mathbf{W}\mathbf{y}$ by using a modified sweep routine that produces a generalized (g2) inverse $(\mathbf{X}'\mathbf{W}\mathbf{X})^{-}$ and a solution (Pringle and Rayner 1971)

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-}\mathbf{X}'\mathbf{W}\mathbf{y}$$

where $\mathbf{W}$ is the diagonal matrix constructed from WEIGHT variable values.

For models with class variables, there are more design matrix columns than there are degrees of freedom (df) for the effect. Thus, there are linear dependencies among the columns. In this case, the parameters are not estimable; there is an infinite number of least squares solutions. PROC

SURVEYREG uses a generalized (g2) inverse to obtain values for the estimates. The solution values are not displayed unless you specify the SOLUTION option in the MODEL statement. The solution has the characteristic that estimates are zero whenever the design column for that parameter is a linear combination of previous columns. (In strict terms, the solution values should not be called estimates.) With this full parameterization, hypothesis tests are constructed to test linear functions of the parameters that are estimable.

Variance Estimation

PROC SURVEYREG uses the Taylor series method or replication (resampling) methods to estimate sampling errors of estimators based on complex sample designs (Woodruff (1971); Fuller (1975); Fuller, Kennedy, Schnell, Sullivan, and Park (1989); Särndal, Swensson, and Wretman (1992); Wolter (1985); Rust (1985); Dippo, Fay, and Morganstein (1984); Rao and Shao (1999); Rao, Wu, and Yue (1992); and Rao and Shao (1996)). You can use the VARMETHOD= option to specify a variance estimation method to use. By default, the Taylor series method is used. However, replication methods have recently gained popularity for estimating variances in complex survey data analysis. One reason for this popularity is the relative simplicity of replication-based estimates, especially for nonlinear estimators; another is that modern computational capacity has made replication methods feasible for practical survey analysis.

Replication methods draw multiple replicates (also called subsamples) from a full sample according to a specific resampling scheme. The most commonly used resampling schemes are the *balanced repeated replication* (BRR) method and the *jackknife* method. For each replicate, the original weights are modified for the PSUs in the replicates to create replicate weights. The parameters of interest are estimated by using the replicate weights for each replicate. Then the variances of parameters of interest are estimated by the variability among the estimates derived from these replicates. You can use the REPWEIGHTS statement to provide your own replicate weights for variance estimation.

The following sections provide details about how the variance-covariance matrix of the estimated regression coefficients is estimated for each variance estimation method.

Taylor Series (Linearization)

The Taylor series (linearization) method is the most commonly used method to estimate the covariance matrix of the regression coefficients for complex survey data. It is the default variance estimation method used by PROC SURVEYREG.

Use the notation described in the section "Notation" on page 6563 to denote the residuals from the linear regression as

$$\mathbf{r} = \mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}$$

with r_{hij} as its elements. Let the $p \times p$ matrix $\mathbf{G}$ be defined as

$$\mathbf{G} = \frac{n-1}{n-p} \sum_{h=1}^{H} \frac{n_h(1-f_h)}{n_h-1} \sum_{i=1}^{n_h} (\mathbf{e}_{hi\cdot} - \bar{\mathbf{e}}_{h\cdot\cdot})'(\mathbf{e}_{hi\cdot} - \bar{\mathbf{e}}_{h\cdot\cdot})$$

where

$$\begin{aligned}
\mathbf{e}_{hij} &= w_{hij} r_{hij} \mathbf{x}_{hij} \\
\mathbf{e}_{hi\cdot} &= \sum_{j=1}^{m_{hi}} \mathbf{e}_{hij} \\
\bar{\mathbf{e}}_{h\cdot\cdot} &= \frac{1}{n_h} \sum_{i=1}^{n_h} \mathbf{e}_{hi\cdot}
\end{aligned}$$

The Taylor series estimate of the covariance matrix of $\hat{\boldsymbol{\beta}}$ is

$$\widehat{\mathbf{V}}(\hat{\boldsymbol{\beta}}) = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-}\mathbf{G}(\mathbf{X}'\mathbf{W}\mathbf{X})^{-}$$

The factor $(n-1)/(n-p)$ in the computation of the matrix $\mathbf{G}$ reduces the small sample bias associated with using the estimated function to calculate deviations (Hidiroglou, Fuller, and Hickman 1980). For simple random sampling, this factor contributes to the degrees of freedom correction applied to the residual mean square for ordinary least squares in which p parameters are estimated. By default, the procedure use this adjustment in the variance estimation. If you do not want to use this multiplier in variance estimation, you can specify the VADJUST=NONE option in the MODEL statement to suppress this factor.

Balanced Repeated Replication (BRR) Method

The balanced repeated replication (BRR) method requires that the full sample be drawn by using a stratified sample design with two primary sampling units (PSUs) per stratum. Let H be the total number of strata. The total number of replicates R is the smallest multiple of 4 that is greater than H. However, if you prefer a larger number of replicates, you can specify the REPS=*number* option. If a $number \times number$ Hadamard matrix cannot be constructed, the number of replicates is increased until a Hadamard matrix becomes available.

Each replicate is obtained by deleting one PSU per stratum according to the corresponding Hadamard matrix and adjusting the original weights for the remaining PSUs. The new weights are called replicate weights.

Replicates are constructed by using the first H columns of the $R \times R$ Hadamard matrix. The rth ($r = 1, 2, ..., R$) replicate is drawn from the full sample according to the rth row of the Hadamard matrix as follows:

- If the (r, h)th element of the Hadamard matrix is 1, then the first PSU of stratum h is included in the rth replicate and the second PSU of stratum h is excluded.
- If the (r, h)th element of the Hadamard matrix is -1, then the second PSU of stratum h is included in the rth replicate and the first PSU of stratum h is excluded.

The replicate weights of the remaining PSUs in each half sample are then doubled to their original weights. For more detail about the BRR method, see Wolter (1985) and Lohr (1999).

By default, an appropriate Hadamard matrix is generated automatically to create the replicates. You can request that the Hadamard matrix be displayed by specifying the VARMETHOD=BRR(PRINTH) method-option. If you provide a Hadamard matrix by specifying the VARMETHOD=BRR(HADAMARD=) method-option, then the replicates are generated according to the provided Hadamard matrix.

Let $\hat{\boldsymbol{\beta}}$ be the estimated regression coefficients from the full sample for $\boldsymbol{\beta}$, and let $\hat{\boldsymbol{\beta}}_r$ be the estimated regression coefficient from the rth replicate by using replicate weights. PROC SURVEYREG estimates the covariance matrix of $\hat{\boldsymbol{\beta}}$ by

$$\widehat{\mathbf{V}}(\hat{\boldsymbol{\beta}}) = \frac{1}{R}\sum_{r=1}^{R}\left(\hat{\boldsymbol{\beta}}_r - \hat{\boldsymbol{\beta}}\right)\left(\hat{\boldsymbol{\beta}}_r - \hat{\boldsymbol{\beta}}\right)'$$

with H degrees of freedom, where H is the number of strata.

You can use the VARMETHOD=BRR(OUTWEIGHTS=) method-option to save the replicate weights into a SAS data set.

Fay's BRR Method

Fay's method is a modification of the BRR method, and it requires a stratified sample design with two primary sampling units (PSUs) per stratum. The total number of replicates R is the smallest multiple of 4 that is greater than the total number of strata H. However, if you prefer a larger number of replicates, you can specify the REPS= option.

For each replicate, Fay's method uses a Fay coefficient $0 \leq \epsilon < 1$ to impose a perturbation of the original weights in the full sample that is gentler than using only half samples, as in the traditional BRR method. The Fay coefficient $0 \leq \epsilon < 1$ can be optionally set by the FAY = ϵ method-option. By default, $\epsilon = 0.5$ if only the FAY method-option is used without specifying a value for ϵ (Judkins 1990; Rao and Shao 1999). When $\epsilon = 0$, Fay's method becomes the traditional BRR method. For more details, see Dippo, Fay, and Morganstein (1984), Fay (1984), Fay (1989), and Judkins (1990).

Let H be the number of strata. Replicates are constructed by using the first H columns of the $R \times R$ Hadamard matrix, where R is the number of replicates, $R > H$. The rth ($r = 1, 2, ..., R$) replicate is created from the full sample according to the rth row of the Hadamard matrix as follows:

- If the (r, h)th element of the Hadamard matrix is 1, then the full sample weight of the first PSU in stratum h is multiplied by ϵ and that of the second PSU is multiplied by $2-\epsilon$ to obtain the rth replicate weights.
- If the (r, h)th element of the Hadamard matrix is -1, then the full sample weight of the first PSU in stratum h is multiplied by $2-\epsilon$ and that of the second PSU is multiplied by ϵ to obtain the rth replicate weights.

You can use the VARMETHOD=BRR(OUTWEIGHTS=) method-option to save the replicate weights into a SAS data set.

By default, an appropriate Hadamard matrix is generated automatically to create the replicates. You can request that the Hadamard matrix be displayed by specifying the VARMETHOD=BRR(PRINTH) method-option. If you provide a Hadamard matrix by specifying the VARMETHOD=BRR(HADAMARD=) method-option, then the replicates are generated according to the provided Hadamard matrix.

Let $\hat{\boldsymbol{\beta}}$ be the estimated regression coefficients from the full sample for $\boldsymbol{\beta}$. Let $\hat{\boldsymbol{\beta}}_r$ be the estimated regression coefficient obtained from the rth replicate by using replicate weights. PROC SURVEYREG estimates the covariance matrix of $\hat{\boldsymbol{\beta}}$ by

$$\widehat{\mathbf{V}}(\hat{\boldsymbol{\beta}}) = \frac{1}{R(1-\epsilon)^2} \sum_{r=1}^{R} \left(\hat{\boldsymbol{\beta}}_r - \hat{\boldsymbol{\beta}}\right)\left(\hat{\boldsymbol{\beta}}_r - \hat{\boldsymbol{\beta}}\right)'$$

with H degrees of freedom, where H is the number of strata.

Jackknife Method

The jackknife method of variance estimation deletes one PSU at a time from the full sample to create replicates. The total number of replicates R is the same as the total number of PSUs. In each replicate, the sample weights of the remaining PSUs are modified by the *jackknife coefficient* α_r. The modified weights are called replicate weights.

The jackknife coefficient and replicate weights are described as follows.

Without Stratification If there is no stratification in the sample design (no STRATA statement), the jackknife coefficients α_r are the same for all replicates:

$$\alpha_r = \frac{R-1}{R} \quad \text{where } r = 1, 2, ..., R$$

Denote the original weight in the full sample for the jth member of the ith PSU as w_{ij}. If the ith PSU is included in the rth replicate ($r = 1, 2, ..., R$), then the corresponding replicate weight for the jth member of the ith PSU is defined as

$$w_{ij}^{(r)} = w_{ij}/\alpha_r$$

With Stratification If the sample design involves stratification, each stratum must have at least two PSUs to use the jackknife method.

Let stratum $\tilde{h}_r$ be the stratum from which a PSU is deleted for the rth replicate. Stratum $\tilde{h}_r$ is called the *donor stratum.* Let $n_{\tilde{h}_r}$ be the total number of PSUs in the donor stratum $\tilde{h}_r$. The jackknife coefficients are defined as

$$\alpha_r = \frac{n_{\tilde{h}_r} - 1}{n_{\tilde{h}_r}} \quad \text{where } r = 1, 2, ..., R$$

Denote the original weight in the full sample for the jth member of the ith PSU as w_{ij}. If the ith PSU is included in the rth replicate ($r = 1, 2, ..., R$), then the corresponding replicate weight for the jth member of the ith PSU is defined as

$$w_{ij}^{(r)} = \begin{cases} w_{ij} & \text{if } i\text{th PSU is not in the donor stratum } \tilde{h}_r \\ w_{ij}/\alpha_r & \text{if } i\text{th PSU is in the donor stratum } \tilde{h}_r \end{cases}$$

You can use the VARMETHOD=JACKKNIFE(OUTJKCOEFS=) method-option to save the jackknife coefficients into a SAS data set and use the VARMETHOD=JACKKNIFE(OUTWEIGHTS=) method-option to save the replicate weights into a SAS data set.

If you provide your own replicate weights with a REPWEIGHTS statement, then you can also provide corresponding jackknife coefficients with the JKCOEFS= option.

Let $\hat{\boldsymbol{\beta}}$ be the estimated regression coefficients from the full sample for $\boldsymbol{\beta}$. Let $\hat{\boldsymbol{\beta}}_r$ be the estimated regression coefficient obtained from the rth replicate by using replicate weights. PROC SURVEYREG estimates the covariance matrix of $\hat{\boldsymbol{\beta}}$ by

$$\widehat{\mathbf{V}}(\hat{\boldsymbol{\beta}}) = \sum_{r=1}^{R} \alpha_r \left(\hat{\boldsymbol{\beta}}_r - \hat{\boldsymbol{\beta}}\right)\left(\hat{\boldsymbol{\beta}}_r - \hat{\boldsymbol{\beta}}\right)'$$

with $R - H$ degrees of freedom, where R is the number of replicates and H is the number of strata, or $R - 1$ when there is no stratification.

Hadamard Matrix

A Hadamard matrix $\mathbf{H}$ is a square matrix whose elements are either 1 or -1 such that

$$\mathbf{H}\mathbf{H}' = k\mathbf{I}$$

where k is the dimension of $\mathbf{H}$ and $\mathbf{I}$ is the identity matrix of order k. The order k is necessarily 1, 2, or a positive integer that is a multiple of 4.

For example, the following matrix is a Hadamard matrix of dimension $k = 8$:

$$\begin{array}{rrrrrrrr} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{array}$$

Degrees of Freedom

PROC SURVEYREG produces tests for the significance of model effects, regression parameters, estimable functions specified in the ESTIMATE statement, and contrasts specified in the CONTRAST

statement. It computes all these tests taking into account the sample design. The degrees of freedom for these tests differ from the degrees of freedom for the ANOVA table, which does not consider the sample design.

Denominator Degrees of Freedom

The denominator *df* refers to the denominator degrees of freedom for *F* tests and to the degrees of freedom for *t* tests in the analysis.

For the Taylor series method, the denominator *df* equals the number of clusters minus the actual number of strata. If there are no clusters, the denominator *df* equals the number of observations minus the actual number of strata. The *actual number of strata* equals the following:

- one, if there is no STRATA statement
- the number of strata in the sample, if there is a STRATA statement but the procedure does not collapse any strata
- the number of strata in the sample after collapsing, if there is a STRATA statement and the procedure collapses strata that have only one sampling unit

Alternatively, you can specify your own denominator *df* by using the DF= option in the MODEL statement.

For the BRR method (including Fay's method) without a REPWEIGHTS statement, the denominator *df* equals the number of strata.

For the jackknife method without a REPWEIGHTS statement, the denominator *df* is equal to the number of replicates minus the *actual number of strata*.

When there is a REPWEIGHTS statement, the denominator *df* equals the number of REPWEIGHTS variables, unless you specify an alternative in the DF= option in a REPWEIGHTS statement.

Numerator Degrees of Freedom

The numerator *df* refers to the numerator degrees of freedom for the Wald *F* statistic associated with an effect or with a contrast. The procedure computes the Wald *F* statistic for an effect as a Type III test; that is, the test has the following properties:

- The hypothesis for an effect does not involve parameters of other effects except for containing effects (which it must involve to be estimable).
- The hypotheses to be tested are invariant to the ordering of effects in the model.

See the section "Testing Effects" on page 6570 for more information. The numerator *df* for the Wald *F* statistic for a contrast is the rank of the **L** matrix that defines the contrast.

Testing Effects

For each effect in the model, PROC SURVEYREG computes an $\mathbf{L}$ matrix such that every element of $\mathbf{L}\boldsymbol{\beta}$ is estimable; the $\mathbf{L}$ matrix has the maximum possible rank associated with the effect. To test the effect, the procedure uses the Wald F statistic for the hypothesis H_0: $\mathbf{L}\boldsymbol{\beta} = 0$. The Wald F statistic equals

$$F_{\text{Wald}} = \frac{(\mathbf{L}\hat{\boldsymbol{\beta}})'(\mathbf{L}'\widehat{\mathbf{V}}\mathbf{L})^{-1}(\mathbf{L}\hat{\boldsymbol{\beta}})}{\text{rank}(\mathbf{L})}$$

with numerator degrees of freedom equal to rank($\mathbf{L}$) and denominator degrees of freedom equal to the number of clusters minus the number of strata (unless you have specified the denominator degrees of freedom with the DF= option in the MODEL statement; see the section "Denominator Degrees of Freedom" on page 6569). It is possible that the $\mathbf{L}$ matrix cannot be constructed for an effect, in which case that effect is not testable. For more information about how the matrix $\mathbf{L}$ is constructed, see the discussion in Chapter 15, "The Four Types of Estimable Functions."

Analysis of Variance (ANOVA)

PROC SURVEYREG produces an analysis of variance table for the model specified in the MODEL statement. This table is identical to the one produced by the GLM procedure for the model. PROC SURVEYREG computes ANOVA table entries by using the sampling weights, but not the sample design information about stratification and clustering.

The degrees of freedom (*df*) displayed in the ANOVA table are the same as those in the ANOVA table produced by PROC GLM. The Total DF is the total degrees of freedom used to obtain the regression coefficient estimates. The Total DF equals the total number of observations minus 1 if the model includes an intercept. If the model does not include an intercept, the Total DF equals the total number of observations. The Model DF equals the degrees of freedom for the effects in the MODEL statement, not including the intercept. The Error DF equals the Total DF minus the Model DF.

Multiple R-Square

PROC SURVEYREG computes a multiple R-square for the weighted regression as

$$R^2 = 1 - \frac{\text{SS}_{error}}{\text{SS}_{total}}$$

where SS_{error} is the error sum of squares in the ANOVA table

$$\text{SS}_{error} = \mathbf{r}'\mathbf{W}\mathbf{r}$$

and SS_{total} is the total sum of squares

$$\text{SS}_{total} = \begin{cases} \mathbf{y}'\mathbf{W}\mathbf{y} & \text{if no intercept} \\ \mathbf{y}'\mathbf{W}\mathbf{y} - \left(\sum_{h=1}^{H}\sum_{i=1}^{n_h}\sum_{j=1}^{m_{hi}} w_{hij}y_{hij}\right)^2 / w_{\cdots} & \text{otherwise} \end{cases}$$

where $w_{\cdots}$ is the sum of the sampling weights over all observations.

Adjusted R-Square

If you specify the option ADJRSQ in the MODEL statement, PROC SURVEYREG computes an multiple R-square adjusted as the weighted regression as

$$\text{ADJRSQ} = \begin{cases} 1 - \dfrac{n(1-R^2)}{n-p} & \text{if no intercept} \\ 1 - \dfrac{(n-1)(1-R^2)}{n-p} & \text{otherwise} \end{cases}$$

where R^2 is the multiple R-square.

Root Mean Square Errors

PROC SURVEYREG computes the square root of mean square errors as

$$\sqrt{\text{MSE}} = \sqrt{n \ \text{SS}_{error} \ / \ (n-p) \ w_{..}}$$

where $w_{..}$ is the sum of the sampling weights over all observations.

Design Effect

If you specify the DEFF option in the MODEL statement, PROC SURVEYREG calculates the design effects for the regression coefficients. The design effect of an estimate is the ratio of the actual variance to the variance computed under the assumption of simple random sampling:

$$\text{DEFF} = \frac{\text{variance under the sample design}}{\text{variance under simple random sampling}}$$

See Kish (1965, p. 258) for more details. PROC SURVEYREG computes the numerator as described in the section "Variance Estimation" on page 6564. And the denominator is computed under the assumption that the sample design is simple random sampling, with no stratification and no clustering.

To compute the variance under the assumption of simple random sampling, PROC SURVEYREG calculates the sampling rate as follows. If you specify both sampling weights and sampling rates (or population totals) for the analysis, then the sampling rate under simple random sampling is calculated as

$$f_{\text{SRS}} = n \ / \ w_{..}$$

where n is the sample size and $w_{..}$ (the sum of the weights over all observations) estimates the population size. If the sum of the weights is less than the sample size, f_{SRS} is set to zero. If you specify sampling rates for the analysis but not sampling weights, then PROC SURVEYREG computes the sampling rate under simple random sampling as the average of the stratum sampling rates:

$$f_{\text{SRS}} = \frac{1}{H} \sum_{h=1}^{H} f_h$$

If you do not specify sampling rates (or population totals) for the analysis, then the sampling rate under simple random sampling is assumed to be zero:

$$f_{\text{SRS}} = 0$$

Stratum Collapse

If there is only one sampling unit in a stratum, then PROC SURVEYREG cannot estimate the variance for this stratum for the Taylor series method. To estimate stratum variances, by default the procedure collapses, or combines, those strata that contain only one sampling unit. If you specify the NOCOLLAPSE option in the STRATA statement, PROC SURVEYREG does not collapse strata and uses a variance estimate of zero for any stratum that contains only one sampling unit.

Note that stratum collapse only applies to Taylor series variance estimation (the default method, also specified by VARMETHOD=TAYLOR). The procedure does not collapse strata for BRR or jackknife variance estimation, which you request with the VARMETHOD=BRR or VARMETHOD=JACKKNIFE option.

If you do not specify the NOCOLLAPSE option for the Taylor series method, PROC SURVEYREG collapses strata according to the following rules. If there are multiple strata that contain only one sampling unit each, then the procedure collapses, or combines, all these strata into a new pooled stratum. If there is only one stratum with a single sampling unit, then PROC SURVEYREG collapses that stratum with the preceding stratum, where strata are ordered by the STRATA variable values. If the stratum with one sampling unit is the first stratum, then the procedure combines it with the following stratum.

If you specify stratum sampling rates by using the RATE=*SAS-data-set* option, PROC SURVEYREG computes the sampling rate for the new pooled stratum as the weighted average of the sampling rates for the collapsed strata. See the section "Computational Details" on page 6563 for details. If the specified sampling rate equals 0 for any of the collapsed strata, then the pooled stratum is assigned a sampling rate of 0. If you specify stratum totals by using the TOTAL=*SAS-data-set* option, PROC SURVEYREG combines the totals for the collapsed strata to compute the sampling rate for the new pooled stratum.

Sampling Rate of the Pooled Stratum from Collapse

Assuming that PROC SURVEYREG collapses single-unit strata $h_1, h_2, \ldots, h_c$ into the pooled stratum, the procedure calculates the sampling rate for the pooled stratum as

$$f_{\text{Pooled Stratum}} = \begin{cases} 0 & \text{if any of } f_{h_l} = 0 \text{ where } l = 1, 2, \ldots, c \\ \left(\sum_{l=1}^{c} n_{h_l} f_{h_l}^{-1} \right)^{-1} \sum_{l=1}^{c} n_{h_l} & \text{otherwise} \end{cases}$$

Contrasts

You can use the CONTRAST statement to perform custom hypothesis tests. If the hypothesis is testable in the univariate case, the Wald F statistic for $H_0 : \mathbf{L}\boldsymbol{\beta} = 0$ is computed as

$$F_{\text{Wald}} = \frac{(\mathbf{L}_{\text{Full}}\hat{\boldsymbol{\beta}})'(\mathbf{L}'_{\text{Full}}\widehat{\mathbf{V}}\mathbf{L}_{\text{Full}})^{-1}(\mathbf{L}_{\text{Full}}\hat{\boldsymbol{\beta}})}{\text{rank}(\mathbf{L})}$$

where $\mathbf{L}$ is the contrast vector or matrix you specify, $\boldsymbol{\beta}$ is the vector of regression parameters, $\hat{\boldsymbol{\beta}} = (\mathbf{X}'\mathbf{W}\mathbf{X})^{-}\mathbf{X}'\mathbf{W}\mathbf{Y}$, $\widehat{\mathbf{V}}$ is the estimated covariance matrix of $\hat{\boldsymbol{\beta}}$, rank($\mathbf{L}$) is the rank of $\mathbf{L}$, and $\mathbf{L}_{\text{Full}}$ is a matrix such that

- $\mathbf{L}_{\text{Full}}$ has the same number of columns as $\mathbf{L}$
- $\mathbf{L}_{\text{Full}}$ has full row rank
- the rank of $\mathbf{L}_{\text{Full}}$ equals the rank of the $\mathbf{L}$ matrix
- all rows of $\mathbf{L}_{\text{Full}}$ are estimable functions
- the Wald F statistic computed using the $\mathbf{L}_{\text{Full}}$ matrix is equivalent to the Wald F statistic computed by using the $\mathbf{L}$ matrix with any row deleted that is a linear combination of previous rows

If $\mathbf{L}$ is a full-rank matrix and all rows of $\mathbf{L}$ are estimable functions, then $\mathbf{L}_{\text{Full}}$ is the same as $\mathbf{L}$. It is possible that $\mathbf{L}_{\text{Full}}$ matrix cannot be constructed for contrasts in a CONTRAST statement, in which case the contrasts are not testable.

Domain Analysis

A DOMAIN statement requests that the procedure perform regression analysis for each domain.

For a domain D, let I_D be the corresponding indicator variable:

$$I_D(h,i,j) = \begin{cases} 1 & \text{if observation } (h,i,j) \text{ belongs to domain } D \\ 0 & \text{otherwise} \end{cases}$$

Let

$$v_{hij} = w_{hij} I_D(h,i,j) = \begin{cases} w_{hij} & \text{if observation } (h,i,j) \text{ belongs to domain } D \\ 0 & \text{otherwise} \end{cases}$$

The regression in domain D uses v as the weight variable.

Computational Resources

Due to the nature of survey data analysis, the SURVEYREG procedure requires more memory compared to analysis that uses the same regression model in the GLM procedure. For details about

the amount of memory related to the modeling, see the section "Computational Resources" on page 2544 in Chapter 39, "The GLM Procedure."

The memory needed in the SURVEYREG procedure to handle the survey design is described as follows.

Let

- H be the total number of strata
- n_c be the total clusters in your sample across all H strata if it is a cluster sampling
- p be the total number of parameters in the model

The memory needed (in bytes) is

$$48H + 8pH + 4p(p+1)H$$

For a cluster sampling, the additional memory needed (in bytes) is

$$48H + 8pH + 4p(p+1)H + 4p(p+1)n_c + 16n_c$$

Other small amounts of additional memory are also used by the SURVEYREG procedure. However, when you have large number of clusters or strata, or large number of parameters in your model, the memory described previously dominates the total memory usage in the procedure.

Output Data Sets

You can use the Output Delivery System (ODS) to create a SAS data set from any piece of PROC SURVEYREG output. See the section "ODS Table Names" on page 6581 for more information. For a more detailed description of using ODS, see Chapter 20, "Using the Output Delivery System."

PROC SURVEYREG also provides an OUTPUT statement to create a data set that contains estimated linear predictors and their standard error estimates, the residuals from the linear regression, and the confidence limits for the predictors.

If you use BRR or jackknife variance estimation, PROC SURVEYREG provides an output data set that stores the replicate weights and an output data set that stores the jackknife coefficients for jackknife variance estimation.

OUT= Data Set Created by the OUTPUT Statement

The OUTPUT statement produces an output data set that contains the following:

- all original data from the SAS data set input to PROC SURVEYREG
- the new variables corresponding to the diagnostic measures specified with statistics keywords in the OUTPUT statement (PREDICTED=, RESIDUAL=, and so on)

When any independent variable in the analysis (including all classification variables) is missing for an observation, then all new variables that correspond to diagnostic measures are missing for the observation in the output data set.

When a dependent variable in the analysis is missing for an observation, then the residual variable that corresponds to R is also missing in the output data set. However, the variables corresponding to LCLM, P, STDP, and UCLM are not missing.

Replicate Weights Output Data Set

If you specify the OUTWEIGHTS= method-option for BRR or jackknife method in the VARMETHOD= option, PROC SURVEYREG stores the replicate weights in an output data set. The OUTWEIGHTS= output data set contains all observations used in the analysis or all valid observations in the DATA= input data set. A valid observation is an observation that has a positive value of the WEIGHT variable. Valid observations must also have nonmissing values of the STRATA and CLUSTER variables, unless you specify the MISSING option.

The OUTWEIGHTS= data set contains the following variables:

- all variables in the DATA= input data set
- RepWt_1, RepWt_2, . . ., RepWt_n, which are the replicate weight variables

where n is the total number of replicates in the analysis. Each replicate weight variable contains the replicate weights for the corresponding replicate. Replicate weights equal zero for those observations not included in the replicate.

After the procedure creates replicate weights for a particular input data set and survey design, you can use the OUTWEIGHTS= method-option to store these replicate weights and then use them again in subsequent analyses, either in PROC SURVEYREG or in the other survey procedures. You can use the REPWEIGHTS statement to provide replicate weights for the procedure.

Jackknife Coefficients Output Data Set

If you specify the OUTJKCOEFS= method-option for VARMETHOD=JACKKNIFE, PROC SURVEYREG stores the jackknife coefficients in an output data set. The OUTJKCOEFS= output data set contains one observation for each replicate. The OUTJKCOEFS= data set contains the following variables:

- Replicate, which is the replicate number for the jackknife coefficient
- JKCoefficient, which is the jackknife coefficient

- DonorStratum, which is the stratum of the PSU that was deleted to construct the replicate, if you specify a STRATA statement

After the procedure creates jackknife coefficients for a particular input data set and survey design, you can use the OUTJKCOEFS= method-option to store these coefficients and then use them again in subsequent analyses, either in PROC SURVEYREG or in the other survey procedures. You can use the JKCOEFS= option in the REPWEIGHTS statement to provide jackknife coefficients for the procedure.

Displayed Output

The SURVEYREG procedure produces the following output.

Data Summary

By default, PROC SURVEYREG displays the following information in the "Data Summary" table:

- Number of Observations, which is the total number of observations used in the analysis, excluding observations with missing values
- Sum of Weights, if you specify a WEIGHT statement
- Mean of the dependent variable in the MODEL statement, or Weighted Mean if you specify a WEIGHT statement
- Sum of the dependent variable in the MODEL statement, or Weighted Sum if you specify a WEIGHT statement

Design Summary

When you specify a CLUSTER statement or a STRATA statement, the procedure displays a "Design Summary" table, which provides the following sample design information:

- Number of Strata, if you specify a STRATA statement
- Number of Strata Collapsed, if the procedure collapses strata
- Number of Clusters, if you specify a CLUSTER statement
- Overall Sampling Rate used to calculate the design effect, if you specify the DEFF option in the MODEL statement

Domain Summary

By default, PROC SURVEYREG displays the following information in the "Domain Summary" table:

- Number of Observations, which is the total number of observations used in the analysis
- total number of observations in the current domain
- total number of observations not in the current domain
- Sum of Weights for the observations in the current domain, if you specify a WEIGHT statement

Fit Statistics

By default, PROC SURVEYREG displays the following regression statistics in the "Fit Statistics" table:

- R-square for the regression
- Root MSE, which is the square root of the mean square error
- Denominator DF, which is the denominator degrees of freedom for the F tests and also the degrees of freedom for the t tests produced by the procedure

Variance Estimation

If the variance method is not Taylor series (see the section "Variance Estimation" on page 6564) or if the NOMCAR option is used, by default, PROC SURVEYREG displays the following variance estimation specifications the "Variance Estimation" table:

- Method, which is the variance estimation method
- Number of Replicates, which is the number of replicates if you specify the VARMETHOD=BRR or VARMETHOD=JACKKNIFE option
- Hadamard Data Set, which is the name of the SAS data set for the HADAMARD matrix if you specify the VARMETHOD=BRR(HADAMARD=) method-option in the VARMETHOD=BRR option
- Fay Coefficient, which is the value of the FAY coefficient if you specify the VARMETHOD=BRR(FAY) method-option
- Replicate Weights Dataset, which is the name of the SAS data set that contains the replicate weights
- Missing Levels, which indicates whether a missing category for categorical variables is created depending on whether you specify the MISSING option

- Missing Values, which indicates whether observations with missing values are included in the analysis, depending on whether you specify the NOMCAR option

Stratum Information

When you specify the LIST option in the STRATA statement, PROC SURVEYREG displays a "Stratum Information" table, which provides the following information for each stratum:

- Stratum Index, which is a sequential stratum identification number
- STRATA variable(s), which lists the levels of STRATA variables for the stratum
- Population Total, if you specify the TOTAL= option
- Sampling Rate, if you specify the TOTAL= option or the RATE= option. If you specify the TOTAL= option, the sampling rate is based on the number of nonmissing observations in the stratum.
- N Obs, which is the number of observations
- number of Clusters, if you specify a CLUSTER statement
- Collapsed, which has the value 'Yes' if the stratum is collapsed with another stratum before analysis

If PROC SURVEYREG collapses strata, the "Stratum Information" table also displays stratum information for the new, collapsed stratum. The new stratum has a Stratum Index of 0 and is labeled 'Pooled.'

Class Level Information

If you use a CLASS statement to name classification variables, PROC SURVEYREG displays a "Class Level Information" table. This table contains the following information for each classification variable:

- Class Variable, which lists each CLASS variable name
- Levels, which is the number of values or levels of the classification variable
- Values, which lists the values of the classification variable. The values are separated by a white space character; therefore, to avoid confusion, you should not include a white space character within a classification variable value.

X'X Matrix

If you specify the XPX option in the MODEL statement, PROC SURVEYREG displays the $\mathbf{X}'\mathbf{X}$ matrix. When there is a WEIGHT variable, the procedure displays the $\mathbf{X}'\mathbf{W}\mathbf{X}$ matrix. This option also displays the crossproducts vector $\mathbf{X}'\mathbf{y}$ or $\mathbf{X}'\mathbf{W}\mathbf{y}$, where $\mathbf{y}$ is the response vector (dependent variable).

Inverse Matrix of X'X

If you specify the INV option in the MODEL statement, PROC SURVEYREG displays the inverse or the generalized inverse of the $\mathbf{X}'\mathbf{X}$ matrix. When there is a WEIGHT variable, the procedure displays the inverse or the generalized inverse of the $\mathbf{X}'\mathbf{W}\mathbf{X}$ matrix.

ANOVA for Dependent Variable

If you specify the ANOVA option in the model statement, PROC SURVEYREG displays an analysis of variance table for the dependent variable. This table is identical to the ANOVA table displayed by the GLM procedure.

Tests of Model Effects

By default, PROC SURVEYREG displays a "Tests of Model Effects" table, which provides Wald's F test for each effect in the model. The table contains the following information for each effect:

- Effect, which is the effect name
- Num DF, which is the numerator degrees of freedom for Wald's F test
- F Value, which is Wald's F statistic
- Pr > F, which is the significance probability corresponding to the F Value

A footnote displays the denominator degrees of freedom, which is the same for all effects.

Estimated Regression Coefficients

PROC SURVEYREG displays the "Estimated Regression Coefficients" table by default when there is no CLASS statement. Also, the procedure displays this table when you specify a CLASS statement and also specify the SOLUTIONS option in the MODEL statement. This table contains the following information for each regression parameter:

- Parameter, which identifies the effect or regressor variable
- Estimate, which is the estimate of the regression coefficient
- Standard Error, which is the standard error of the estimate
- t Value, which is the t statistic for testing H_0: Parameter $= 0$
- Pr > | t |, which is the two-sided significance probability corresponding to the t Value

Covariance of Estimated Regression Coefficients

When you specify the COVB option in the MODEL statement, PROC SURVEYREG displays the "Covariance of Estimated Regression Coefficients" matrix.

Coefficients of Contrast

When you specify the E option in a CONTRAST statement, PROC SURVEYREG displays a "Coefficients of Contrast" table for the contrast. You can use this table to check the coefficients you specified in the CONTRAST statement. Also, this table gives a note for a nonestimable contrast.

Analysis of Contrasts

If you specify a CONTRAST statement, PROC SURVEYREG produces an "Analysis of Contrasts" table, which displays Wald's F test for the contrast. If you use more than one CONTRAST statement, the procedure displays all results in the same table. The "Analysis of Contrasts" table contains the following information for each contrast:

- Contrast, which is the label of the contrast
- Num DF, which is the numerator degrees of freedom for Wald's F test
- F Value, which is Wald's F statistic for testing H_0: Contrast $= 0$
- Pr > F, which is the significance probability corresponding to the F Value

Coefficients of Estimate

When you specify the E option in an ESTIMATE statement, PROC SURVEYREG displays a "Coefficients of Estimate" table for the linear function of the regression parameters in the ESTIMATE statement. You can use this table to check the coefficients you specified in the ESTIMATE statement. Also, this table gives a note for a nonestimable function.

Analysis of Estimable Functions

If you specify an ESTIMATE statement, PROC SURVEYREG checks the function for estimability. If the function is estimable, PROC SURVEYREG produces an "Analysis of Estimable Functions" table, which displays the estimate and the corresponding t test. If you use more than one ESTIMATE statement, the procedure displays all results in the same table. The table contains the following information for each estimable function:

- Parameter, which is the label of the function
- Estimate, which is the estimate of the estimable linear function

- Standard Error, which is the standard error of the estimate
- t Value, which is the t statistic for testing H_0: Estimable Function $= 0$
- Pr > | t |, which is the two-sided significance probability corresponding to the t Value

Hadamard Matrix

If you specify the VARMETHOD=BRR(PRINTH) method-option in the PROC statement, PROC SURVEYREG displays the Hadamard matrix.

When you provide a Hadamard matrix with the VARMETHOD=BRR(HADAMARD=) method-option, the procedure displays only used rows and columns of the Hadamard matrix.

ODS Table Names

PROC SURVEYREG assigns a name to each table it creates. You can use these names to reference the table when by using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 86.3. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 86.3 ODS Tables Produced by PROC SURVEYREG

ODS Table Name	Description	Statement	Option
ANOVA	ANOVA for dependent variable	MODEL	ANOVA
ClassVarInfo	Class level information	CLASS	default
ContrastCoef	Coefficients of contrast	CONTRAST	E
Contrasts	Analysis of contrasts	CONTRAST	default
CovB	Covariance of estimated regression coefficients	MODEL	COVB
DataSummary	Data summary	PROC	default
DesignSummary	Design summary	STRATA | CLUSTER	default
DomainSummary	Domain summary	DOMAIN	default
Effects	Tests of model effects	MODEL	defect
EstimateCoef	Coefficients of estimate	ESTIMATE	E
Estimates	Analysis of estimable functions	ESTIMATE	default
FitStatistics	Fit statistics	MODEL	default
HadamardMatrix	Hadamard matrix	PROC	PRINTH
InvXPX	Inverse matrix of $\mathbf{X}'\mathbf{X}$	MODEL	INV
ParameterEstimates	Estimated regression coefficients	MODEL	SOLUTION
StrataInfo	Stratum information	STRATA	LIST

Table 86.3 (continued)

ODS Table Name	Description	Statement	Option
VarianceEstimation	Variance estimation	PROC	default
XPX	$\mathbf{X}'\mathbf{X}$ matrix	MODEL	XPX

By referring to the names of such tables, you can use the ODS OUTPUT statement to place one or more of these tables in output data sets.

For example, the following statements create an output data set MyStrata, which contains the "StrataInfo" table, an output data set MyParmEst, which contains the "ParameterEstimates" table, and an output data set Cov, which contains the "CovB" table for the ice cream study discussed in the section "Stratified Sampling" on page 6538:

```
title1 'Ice Cream Spending Analysis';
title2 'Stratified Simple Random Sample Design';
proc surveyreg data=IceCream total=StudentTotals;
   strata Grade /list;
   class Kids;
   model Spending = Income Kids / solution covb;
   weight Weight;
   ods output StrataInfo = MyStrata
              ParameterEstimates = MyParmEst
              CovB = Cov;
run;
```

Note that the option CovB is specified in the MODEL statement in order to produce the covariance matrix table.

Examples: SURVEYREG Procedure

Example 86.1: Simple Random Sampling

This example investigates the relationship between the labor force participation rate (LFPR) of women in 1968 and 1972 in large cities in the United States. A simple random sample of 19 cities is drawn from a total of 200 cities. For each selected city, the LFPRs are recorded and saved in a SAS data set Labor. In the following DATA step, LFPR in 1972 is contained in the variable LFPR1972, and the LFPR in 1968 is identified by the variable LFPR1968:

```
data Labor;
   input City $ 1-16 LFPR1972 LFPR1968;
   datalines;
New York         .45     .42
Los Angeles      .50     .50
Chicago          .52     .52
Philadelphia     .45     .45
Detroit          .46     .43
San Francisco    .55     .55
Boston           .60     .45
Pittsburgh       .49     .34
St. Louis        .35     .45
Connecticut      .55     .54
Washington D.C.  .52     .42
Cincinnati       .53     .51
Baltimore        .57     .49
Newark           .53     .54
Minn/St. Paul    .59     .50
Buffalo          .64     .58
Houston          .50     .49
Patterson        .57     .56
Dallas           .64     .63
;
```

Assume that the LFPRs in 1968 and 1972 have a linear relationship, as shown in the following model:

$$\text{LFPR1972} = \beta_0 + \beta_1 * \text{LFPR1968} + \text{error}$$

You can use PROC SURVEYREG to obtain the estimated regression coefficients and estimated standard errors of the regression coefficients. The following statements perform the regression analysis:

```
title 'Study of Labor Force Participation Rates of Women';
proc surveyreg data=Labor total=200;
   model LFPR1972 = LFPR1968;
run;
```

Here, the TOTAL=200 option specifies the finite population total from which the simple random sample of 19 cities is drawn. You can specify the same information by using the sampling rate option RATE=0.095 (19/200=.095).

Output 86.1.1 summarizes the data information and the fit information.

Output 86.1.1 Summary of Regression Using Simple Random Sampling

Study of Labor Force Participation Rates of Women

The SURVEYREG Procedure

Regression Analysis for Dependent Variable LFPR1972

Data Summary

Number of Observations	19
Mean of LFPR1972	0.52684
Sum of LFPR1972	10.01000

Fit Statistics

R-square	0.3970
Root MSE	0.05657
Denominator DF	18

Output 86.1.2 presents the significance tests for the model effects and estimated regression coefficients. The F tests and t tests for the effects in the model are also presented in these tables.

Output 86.1.2 Regression Coefficient Estimates

Tests of Model Effects

Effect	Num DF	F Value	Pr > F
Model	1	13.84	0.0016
Intercept	1	4.63	0.0452
LFPR1968	1	13.84	0.0016

NOTE: The denominator degrees of freedom for the F tests is 18.

Estimated Regression Coefficients

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	0.20331056	0.09444296	2.15	0.0452
LFPR1968	0.65604048	0.17635810	3.72	0.0016

NOTE: The denominator degrees of freedom for the t tests is 18.

From the regression performed by PROC SURVEYREG, you obtain a positive estimated slope for the linear relationship between the LFPR in 1968 and the LFPR in 1972. The regression coefficients are all significant at the 5% level. The effects Intercept and LFPR1968 are significant in the model at the 5% level. In this example, the F test for the overall model without intercept is the same as the effect LFPR1968.

Example 86.2: Simple Random Cluster Sampling

This example illustrates the use of regression analysis in a simple random cluster sample design. The data are from Särndal, Swensson, and Wretman (1992, p. 652).

A total of 284 Swedish municipalities are grouped into 50 clusters of neighboring municipalities. Five clusters with a total of 32 municipalities are randomly selected. The results from the regression analysis in which clusters are used in the sample design are compared to the results of a regression analysis that ignores the clusters. The linear relationship between the population in 1975 and in 1985 is investigated.

The 32 selected municipalities in the sample are saved in the data set Municipalities:

```
data Municipalities;
   input Municipality Cluster Population85 Population75;
   datalines;
205   37    5    5
206   37   11   11
207   37   13   13
208   37    8    8
209   37   17   19
  6    2   16   15
  7    2   70   62
  8    2   66   54
  9    2   12   12
 10    2   60   50
 94   17    7    7
 95   17   16   16
 96   17   13   11
 97   17   12   11
 98   17   70   67
 99   17   20   20
100   17   31   28
101   17   49   48
276   50    6    7
277   50    9   10
278   50   24   26
279   50   10    9
280   50   67   64
281   50   39   35
282   50   29   27
283   50   10    9
284   50   27   31
 52   10    7    6
 53   10    9    8
 54   10   28   27
 55   10   12   11
 56   10  107  108
;
```

The variable Municipality identifies the municipalities in the sample; the variable Cluster indicates the

cluster to which a municipality belongs; and the variables Population85 and Population75 contain the municipality populations in 1985 and in 1975 (in thousands), respectively. A regression analysis is performed by PROC SURVEYREG with a CLUSTER statement:

```
title1 'Regression Analysis for Swedish Municipalities';
title2 'Cluster Simple Random Sampling';
proc surveyreg data=Municipalities total=50;
   cluster Cluster;
   model Population85=Population75;
run;
```

The TOTAL=50 option specifies the total number of clusters in the sampling frame.

Output 86.2.1 displays the data and design summary. Since the sample design includes clusters, the procedure displays the total number of clusters in the sample in the "Design Summary" table.

Output 86.2.1 Regression Analysis for Simple Random Cluster Sampling

```
Regression Analysis for Swedish Municipalities
        Cluster Simple Random Sampling

          The SURVEYREG Procedure

Regression Analysis for Dependent Variable Population85

                Data Summary

Number of Observations            32
Mean of Population85           27.50000
Sum of Population85           880.00000

              Design Summary

Number of Clusters                5
```

Output 86.2.2 displays the fit statistics and regression coefficient estimates. In the "Estimated Regression Coefficients" table, the estimated slope for the linear relationship is 1.05, which is significant at the 5% level; but the intercept is not significant. This suggests that a regression line crossing the original can be established between populations in 1975 and in 1985.

Output 86.2.2 Regression Analysis for Simple Random Cluster Sampling

```
                Fit Statistics

R-square                0.9860
Root MSE                3.0488
Denominator DF               4
```

Output 86.2.2 *continued*

Estimated Regression Coefficients

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	-0.0191292	0.89204053	-0.02	0.9839
Population75	1.0546253	0.05167565	20.41	<.0001

NOTE: The denominator degrees of freedom for the t tests is 4.

The CLUSTER statement is necessary in PROC SURVEYREG in order to incorporate the sample design. If you do not specify a CLUSTER statement in the regression analysis, as in the following statements, the standard deviation of the regression coefficients are incorrectly estimated.

```
title1 'Regression Analysis for Swedish Municipalities';
title2 'Simple Random Sampling';
proc surveyreg data=Municipalities total=284;
   model Population85=Population75;
run;
```

The analysis ignores the clusters in the sample, assuming that the sample design is a simple random sampling. Therefore, the TOTAL= option specifies the total number of municipalities, which is 284.

Output 86.2.3 displays the regression results ignoring the clusters. Compared to the results in Output 86.2.1, the regression coefficient estimates are the same. However, without using clusters, the regression coefficients have a smaller variance estimate, as in Output 86.2.3. By using clusters in the analysis, the estimated regression coefficient for effect Population75 is 1.05, with the estimated standard error 0.05, as displayed in Output 86.2.1; without using the clusters, the estimate is 1.05, but with the estimated standard error 0.04, as displayed in Output 86.2.3. To estimated the variance of the regression coefficients correctly, you should include the clustering information in the regression analysis.

Output 86.2.3 Regression Analysis for Simple Random Sampling

Regression Analysis for Swedish Municipalities
Simple Random Sampling

The SURVEYREG Procedure

Regression Analysis for Dependent Variable Population85

Data Summary

Number of Observations	32
Mean of Population85	27.50000
Sum of Population85	880.00000

Output 86.2.3 *continued*

```
                          Fit Statistics

                      R-square            0.9860
                      Root MSE            3.0488
                      Denominator DF          31

                  Estimated Regression Coefficients

                                    Standard
Parameter          Estimate          Error    t Value    Pr > |t|

Intercept       -0.0191292     0.67417606      -0.03      0.9775
Population75     1.0546253     0.03668414      28.75      <.0001

NOTE: The denominator degrees of freedom for the t tests is 31.
```

Example 86.3: Regression Estimator for Simple Random Sample

Using auxiliary information, you can construct the regression estimators to provide more accurate estimates of the population characteristics that are of interest. With ESTIMATE statements in PROC SURVEYREG, you can specify a regression estimator as a linear function of the regression parameters to estimate the population total. This example illustrates this application, by using the data in the previous example.

In this sample, a linear model between the Swedish populations in 1975 and in 1985 is established:

$$\text{Population85} = \alpha + \beta * \text{Population75} + \text{error}$$

Assuming that the total population in 1975 is known to be 8200 (in thousands), you can use the ESTIMATE statement to predict the 1985 total population by using the following statements:

```
title1 'Regression Analysis for Swedish Municipalities';
title2 'Estimate Total Population';
proc surveyreg data=Municipalities total=50;
   cluster Cluster;
   model Population85=Population75;
   estimate '1985 population' Intercept 284 Population75 8200;
run;
```

Since each observation in the sample is a municipality and there is a total of 284 municipalities in Sweden, the coefficient for Intercept (α) in the ESTIMATE statement is 284 and the coefficient for Population75 (β) is the total population in 1975 (8.2 million).

Output 86.3.1 displays the regression results and the estimation of the total population. By using the linear model, you can predict the total population in 1985 to be 8.64 million, with a standard error of 0.26 million.

Output 86.3.1 Use the Regression Estimator to Estimate the Population Total

```
Regression Analysis for Swedish Municipalities
Estimate Total Population

The SURVEYREG Procedure

Regression Analysis for Dependent Variable Population85

Analysis of Estimable Functions

                                    Standard
Parameter            Estimate         Error    t Value    Pr > |t|

1985 population    8642.49485    258.558613      33.43      <.0001

NOTE: The denominator degrees of freedom for the t tests is 4.
```

Example 86.4: Stratified Sampling

This example illustrates the use of the SURVEYREG procedure to perform a regression in a stratified sample design. Consider a population of 235 farms producing corn in Nebraska and Iowa. You are interested in the relationship between corn yield (CornYield) and total farm size (FarmArea).

Each state is divided into several regions, and each region is used as a stratum. Within each stratum, a simple random sample with replacement is drawn. A total of 19 farms is selected by using a stratified simple random sample. The sample size and population size within each stratum are displayed in Table 86.4.

Table 86.4 Number of Farms in Each Stratum

			Number of Farms	
Stratum	**State**	**Region**	**Population**	**Sample**
1	Iowa	1	100	3
2		2	50	5
3		3	15	3
4	Nebraska	1	30	6
5		2	40	2
Total			**235**	**19**

Three models for the data are considered:

- Model I — Common intercept and slope:

$$\text{Corn Yield} = \alpha + \beta * \text{Farm Area}$$

- Model II — Common intercept, different slope:

$$\text{Corn Yield} = \begin{cases} \alpha + \beta_{\text{Iowa}} * \text{Farm Area} & \text{if the farm is in Iowa} \\ \alpha + \beta_{\text{Nebraska}} * \text{Farm Area} & \text{if the farm is in Nebraska} \end{cases}$$

- Model III — Different intercept and different slope:

$$\text{Corn Yield} = \begin{cases} \alpha_{\text{Iowa}} + \beta_{\text{Iowa}} * \text{Farm Area} & \text{if the farm is in Iowa} \\ \alpha_{\text{Nebraska}} + \beta_{\text{Nebraska}} * \text{Farm Area} & \text{if the farm is in Nebraska} \end{cases}$$

Data from the stratified sample are saved in the SAS data set Farms. In the data set Farms, the variable Weight represents the sampling weight. In the following DATA step, the sampling weights are the reciprocals of selection probabilities:

```
data Farms;
   input State $ Region FarmArea CornYield Weight;
   datalines;
Iowa     1 100  54 33.333
Iowa     1  83  25 33.333
Iowa     1  25  10 33.333
Iowa     2 120  83 10.000
Iowa     2  50  35 10.000
Iowa     2 110  65 10.000
Iowa     2  60  35 10.000
Iowa     2  45  20 10.000
Iowa     3  23   5  5.000
Iowa     3  10   8  5.000
Iowa     3 350 125  5.000
Nebraska 1 130  20  5.000
Nebraska 1 245  25  5.000
Nebraska 1 150  33  5.000
Nebraska 1 263  50  5.000
Nebraska 1 320  47  5.000
Nebraska 1 204  25  5.000
Nebraska 2  80  11 20.000
Nebraska 2  48   8 20.000
;
```

The information about population size in each stratum is saved in the SAS data set StratumTotals:

```
data StratumTotals;
   input State $ Region _TOTAL_;
   datalines;
Iowa     1 100
Iowa     2  50
Iowa     3  15
Nebraska 1  30
Nebraska 2  40
;
```

Using the sample data from the data set Farms and the control information data from the data set StratumTotals, you can fit Model I by using PROC SURVEYREG with the following statements:

```
title1 'Analysis of Farm Area and Corn Yield';
title2 'Model I: Same Intercept and Slope';
proc surveyreg data=Farms total=StratumTotals;
   strata State Region / list;
   model CornYield = FarmArea / covB;
   weight Weight;
run;
```

Output 86.4.1 displays the data summary and stratification information fitting Model I. The sampling rates are automatically computed by the procedure based on the sample sizes and the population totals in strata.

Output 86.4.1 Data Summary and Stratum Information Fitting Model I

Analysis of Farm Area and Corn Yield
Model I: Same Intercept and Slope

The SURVEYREG Procedure

Regression Analysis for Dependent Variable CornYield

Data Summary

Number of Observations	19
Sum of Weights	234.99900
Weighted Mean of CornYield	31.56029
Weighted Sum of CornYield	7416.6

Design Summary

Number of Strata	5

Fit Statistics

R-square	0.3882
Root MSE	20.6422
Denominator DF	14

Stratum Information

Stratum Index	State	Region	N Obs	Population Total	Sampling Rate
1	Iowa	1	3	100	3.00%
2		2	5	50	10.0%
3		3	3	15	20.0%
4	Nebraska	1	6	30	20.0%
5		2	2	40	5.00%

Output 86.4.2 displays tests of model effects and the estimated regression coefficients.

Output 86.4.2 Estimated Regression Coefficients and the Estimated Covariance Matrix

Tests of Model Effects

Effect	Num DF	F Value	Pr > F
Model	1	21.74	0.0004
Intercept	1	4.93	0.0433
FarmArea	1	21.74	0.0004

NOTE: The denominator degrees of freedom for the F tests is 14.

Estimated Regression Coefficients

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	11.8162978	5.31981027	2.22	0.0433
FarmArea	0.2126576	0.04560949	4.66	0.0004

NOTE: The denominator degrees of freedom for the t tests is 14.

Covariance of Estimated Regression Coefficients

	Intercept	FarmArea
Intercept	28.300381277	-0.146471538
FarmArea	-0.146471538	0.0020802259

Alternatively, you can assume that the linear relationship between corn yield (CornYield) and farm area (FarmArea) is different among the states (Model II). In order to analyze the data by using this model, you create auxiliary variables FarmAreaNE and FarmAreaIA to represent farm area in different states:

$$\text{FarmAreaNE} = \begin{cases} 0 & \text{if the farm is in Iowa} \\ \text{FarmArea} & \text{if the farm is in Nebraska} \end{cases}$$

$$\text{FarmAreaIA} = \begin{cases} \text{FarmArea} & \text{if the farm is in Iowa} \\ 0 & \text{if the farm is in Nebraska} \end{cases}$$

The following statements create these variables in a new data set called FarmsByState and use PROC SURVEYREG to fit Model II:

```
data FarmsByState; set Farms;
   if State='Iowa' then do;
      FarmAreaIA=FarmArea ; FarmAreaNE=0; end;
   else do;
      FarmAreaIA=0 ; FarmAreaNE=FarmArea; end;
run;
```

The following statements perform the regression by using the new data set FarmsByState. The analysis uses the auxiliary variables FarmAreaIA and FarmAreaNE as the regressors:

```
title1 'Analysis of Farm Area and Corn Yield';
title2 'Model II: Same Intercept, Different Slopes';
proc SURVEYREG data=FarmsByState total=StratumTotals;
   strata State Region;
   model CornYield = FarmAreaIA FarmAreaNE / covB;
   weight Weight;
run;
```

Output 86.4.3 displays the fit statistics and parameter estimates. The estimated slope parameters for each state are quite different from the estimated slope in Model I. The results from the regression show that Model II fits these data better than Model I.

Output 86.4.3 Regression Results from Fitting Model II

```
                    Analysis of Farm Area and Corn Yield
                 Model II: Same Intercept, Different Slopes

                         The SURVEYREG Procedure

            Regression Analysis for Dependent Variable CornYield

                              Fit Statistics

                        R-square            0.8158
                        Root MSE           11.6759
                        Denominator DF          14

                    Estimated Regression Coefficients

                                   Standard
 Parameter       Estimate          Error    t Value    Pr > |t|

 Intercept     4.04234816    3.80934848       1.06      0.3066
 FarmAreaIA    0.41696069    0.05971129       6.98      <.0001
 FarmAreaNE    0.12851012    0.02495495       5.15      0.0001

NOTE: The denominator degrees of freedom for the t tests is 14.

          Covariance of Estimated Regression Coefficients

                    Intercept        FarmAreaIA        FarmAreaNE

Intercept       14.511135861      -0.118001232      -0.079908772
FarmAreaIA      -0.118001232      0.0035654381      0.0006501109
FarmAreaNE      -0.079908772      0.0006501109      0.0006227496
```

For Model III, different intercepts are used for the linear relationship in two states. The following statements illustrate the use of the NOINT option in the MODEL statement associated with the CLASS statement to fit Model III:

```
title2 'Model III: Different Intercepts and Slopes';
proc SURVEYREG data=FarmsByState total=StratumTotals;
   strata State Region;
   class State;
   model CornYield = State FarmAreaIA FarmAreaNE / noint covB solution;
   weight Weight;
run;
```

The model statement includes the classification effect State as a regressor. Therefore, the parameter estimates for effect State present the intercepts in two states.

Output 86.4.4 displays the regression results for fitting Model III, including parameter estimates, and covariance matrix of the regression coefficients. The estimated covariance matrix shows a lack of correlation between the regression coefficients from different states. This suggests that Model III might be the best choice for building a model for farm area and corn yield in these two states.

However, some statistics remain the same under different regression models—for example, Weighted Mean of CornYield. These estimators do not rely on the particular model you use.

Output 86.4.4 Regression Results for Fitting Model III

Analysis of Farm Area and Corn Yield
Model III: Different Intercepts and Slopes

The SURVEYREG Procedure

Regression Analysis for Dependent Variable CornYield

Fit Statistics

R-square	0.9300
Root MSE	11.9810
Denominator DF	14

Estimated Regression Coefficients

Parameter	Estimate	Standard Error	t Value	Pr > |t|
State Iowa	5.27797099	5.27170400	1.00	0.3337
State Nebraska	0.65275201	1.70031616	0.38	0.7068
FarmAreaIA	0.40680971	0.06458426	6.30	<.0001
FarmAreaNE	0.14630563	0.01997085	7.33	<.0001

NOTE: The denominator degrees of freedom for the t tests is 14.

Covariance of Estimated Regression Coefficients

	State Iowa	State Nebraska	FarmAreaIA	FarmAreaNE
State Iowa	27.790863033	0	-0.205517205	0
State Nebraska	0	2.8910750385	0	-0.027354011
FarmAreaIA	-0.205517205	0	0.0041711265	0
FarmAreaNE	0	-0.027354011	0	0.0003988349

Example 86.5: Regression Estimator for Stratified Sample

This example uses the corn yield data from the previous example to illustrate how to construct a regression estimator for a stratified sample design.

As in Example 86.3, by incorporating auxiliary information into a regression estimator, the procedure can produce more accurate estimates of the population characteristics that are of interest. In this example, the sample design is a stratified sample design. The auxiliary information is the total farm areas in regions of each state, as displayed in Table 86.5. You want to estimate the total corn yield by using this information under the three linear models given in Example 86.4.

Table 86.5 Information for Each Stratum

			Number of Farms		
Stratum	**State**	**Region**	**Population**	**Sample**	**Total Farm Area**
1	Iowa	1	100	3	
2		2	50	5	13,200
3		3	15	3	
4	Nebraska	1	30	6	8,750
5		2	40	2	
Total			**235**	**19**	**21,950**

The regression estimator to estimate the total corn yield under Model I can be obtained by using PROC SURVEYREG with an ESTIMATE statement:

```
title1 'Estimate Corn Yield from Farm Size';
title2 'Model I: Same Intercept and Slope';
proc surveyreg data=Farms total=StratumTotals;
   strata State Region / list;
   class  State Region;
   model  CornYield = FarmArea State*Region /solution;
   weight Weight;
   estimate 'Estimate of CornYield under Model I'
            INTERCEPT 235 FarmArea 21950
            State*Region 100 50 15 30 40 /e;
run;
```

To apply the constraint in each stratum that the weighted total number of farms equals to the total number of farms in the stratum, you can include the strata as an effect in the MODEL statement, effect State*Region. Thus, the CLASS statement must list the STRATA variables, State and Region, as classification variables. The following ESTIMATE statement specifies the regression estimator, which is a linear function of the regression parameters:

```
estimate 'Estimate of CornYield under Model I'
         INTERCEPT 235 FarmArea 21950
         State*Region 100 50 15 30 40 /e;
```

This linear function contains the total for each explanatory variable in the model. Because the sampling units are farms in this example, the coefficient for Intercept in the ESTIMATE statement is

the total number of farms (235); the coefficient for FarmArea is the total farm area listed in Table 86.5 (21950); and the coefficients for effect State*Region are the total number of farms in each strata (as displayed in Table 86.5).

Output 86.5.1 displays the results of the ESTIMATE statement. The regression estimator for the total of CornYield in Iowa and Nebraska is 7464 under Model I, with a standard error of 927.

Output 86.5.1 Regression Estimator for the Total of CornYield under Model I

```
                    Estimate Corn Yield from Farm Size
                    Model I: Same Intercept and Slope

                        The SURVEYREG Procedure

            Regression Analysis for Dependent Variable CornYield

                     Analysis of Estimable Functions

                                                  Standard
 Parameter                              Estimate       Error  t Value  Pr > |t|

 Estimate of CornYield under Model I  7463.52329  926.841541     8.05    <.0001

       NOTE: The denominator degrees of freedom for the t tests is 14.
```

Under Model II, a regression estimator for totals can be obtained by using the following statements:

```
title1 'Estimate Corn Yield from Farm Size';
title2 'Model II: Same Intercept, Different Slopes';
proc surveyreg data=FarmsByState total=StratumTotals;
   strata State Region;
   class  State Region;
   model  CornYield = FarmAreaIA FarmAreaNE
                      state*region /solution;
   weight Weight;
   estimate 'Total of CornYield under Model II'
            INTERCEPT 235 FarmAreaIA 13200 FarmAreaNE 8750
            State*Region 100 50 15 30 40 /e;
run;
```

In this model, you also need to include strata as a fixed effect in the MODEL statement. Other regressors are the auxiliary variables FarmAreaIA and FarmAreaNE (defined in Example 86.4). In the following ESTIMATE statement, the coefficient for Intercept is still the total number of farms; and the coefficients for FarmAreaIA and FarmAreaNE are the total farm area in Iowa and Nebraska, respectively, as displayed in Table 86.5. The total number of farms in each strata are the coefficients for the strata effect:

```
   estimate 'Total of CornYield under Model II'
            INTERCEPT 235 FarmAreaIA 13200 FarmAreaNE 8750
            State*Region 100 50 15 30 40 /e;
```

Output 86.5.2 displays that the results of the regression estimator for the total of corn yield in two states under Model II is 7580 with a standard error of 859. The regression estimator under Model II has a slightly smaller standard error than under Model I.

Output 86.5.2 Regression Estimator for the Total of CornYield under Model II

```
                    Estimate Corn Yield from Farm Size
                  Model II: Same Intercept, Different Slopes

                          The SURVEYREG Procedure

             Regression Analysis for Dependent Variable CornYield

                      Analysis of Estimable Functions

                                                  Standard
Parameter                              Estimate        Error   t Value   Pr > |t|

Total of CornYield under Model II   7580.48657   859.180439      8.82     <.0001

        NOTE: The denominator degrees of freedom for the t tests is 14.
```

Finally, you can apply Model III to the data and estimate the total corn yield. Under Model III, you can also obtain the regression estimators for the total corn yield for each state. Three ESTIMATE statements are used in the following statements to create the three regression estimators:

```
title1 'Estimate Corn Yield from Farm Size';
title2 'Model III: Different Intercepts and Slopes';
proc SURVEYREG data=FarmsByState total=StratumTotals;
   strata State Region;
   class  State Region;
   model  CornYield = state FarmAreaIA FarmAreaNE
      State*Region /noint solution;
   weight Weight;
   estimate 'Total CornYield in Iowa under Model III'
             State 165 0 FarmAreaIA 13200 FarmAreaNE  0
             State*region 100 50 15  0  0 /e;
   estimate 'Total CornYield in Nebraska under Model III'
             State 0 70 FarmAreaIA 0 FarmAreaNE 8750
             State*Region 0 0 0 30 40 /e;
   estimate 'Total CornYield in both states under Model III'
             State 165 70 FarmAreaIA 13200 FarmAreaNE 8750
             State*Region 100 50 15 30 40 /e;
run;
```

The fixed effect State is added to the MODEL statement to obtain different intercepts in different states, by using the NOINT option. Among the ESTIMATE statements, the coefficients for explanatory variables are different depending on which regression estimator is estimated. For example, in the ESTIMATE statement

```
   estimate 'Total CornYield in Iowa under Model III'
             State 165 0 FarmAreaIA 13200 FarmAreaNE  0
             State*region 100 50 15  0  0 /e;
```

the coefficients for the effect State are 165 and 0, respectively. This indicates that the total number of farms in Iowa is 165 and the total number of farms in Nebraska is 0, because the estimation is the total corn yield in Iowa only. Similarly, the total numbers of farms in three regions in Iowa are

used for the coefficients of the strata effect State*Region, as displayed in Table 86.5.

Output 86.5.3 displays the results from the three regression estimators by using Model III. Since the estimations are independent in each state, the total corn yield from both states is equal to the sum of the estimated total of corn yield in Iowa and Nebraska, $6246 + 1334 = 7580$. This regression estimator is the same as the one under Model II. The variance of regression estimator of the total corn yield in both states is the sum of variances of regression estimators for total corn yield in each state. Therefore, it is not necessary to use Model III to obtain the regression estimator for the total corn yield unless you need to estimate the total corn yield for each individual state.

Output 86.5.3 Regression Estimator for the Total of CornYield under Model III

```
                    Estimate Corn Yield from Farm Size
                 Model III: Different Intercepts and Slopes

                          The SURVEYREG Procedure

             Regression Analysis for Dependent Variable CornYield

                      Analysis of Estimable Functions

                                                          Standard
Parameter                                      Estimate       Error  t Value

Total CornYield in Iowa under Model III      6246.10697  851.272372     7.34
Total CornYield in Nebraska under Model III  1334.37961  116.302948    11.47
Total CornYield in both states under Model III 7580.48657 859.180439     8.82

                      Analysis of Estimable Functions

          Parameter                                    Pr > |t|

          Total CornYield in Iowa under Model III        <.0001
          Total CornYield in Nebraska under Model III    <.0001
          Total CornYield in both states under Model III <.0001

       NOTE: The denominator degrees of freedom for the t tests is 14.
```

Example 86.6: Stratum Collapse

In a stratified sample, it is possible that some strata might have only one sampling unit. When this happens, PROC SURVEYREG collapses the strata that contain a single sampling unit into a pooled stratum. For more detailed information about stratum collapse, see the section "Stratum Collapse" on page 6572.

Suppose that you have the following data:

```
data Sample;
   input Stratum X Y W;
   datalines;
10 0 0 5
10 1 1 5
11 1 1 10
11 1 2 10
12 3 3 16
33 4 4 45
14 6 7 50
12 3 4 16
;
```

The variable Stratum is again the stratification variable, the variable X is the independent variable, and the variable Y is the dependent variable. You want to regress Y on X. In the data set Sample, both Stratum=33 and Stratum=14 contain one observation. By default, PROC SURVEYREG collapses these strata into one pooled stratum in the regression analysis.

To input the finite population correction information, you create the SAS data set StratumTotals:

```
data StratumTotals;
   input Stratum _TOTAL_;
   datalines;
10 10
11 20
12 32
33 40
33 45
14 50
15  .
66 70
;
```

The variable Stratum is the stratification variable, and the variable _TOTAL_ contains the stratum totals. The data set StratumTotals contains more strata than the data set Sample. Also in the data set StratumTotals, more than one observation contains the stratum totals for Stratum=33:

```
33 40
33 45
```

PROC SURVEYREG allows this type of input. The procedure simply ignores strata that are not present in the data set Sample; for the multiple entries of a stratum, the procedure uses the first observation. In this example, Stratum=33 has the stratum total _TOTAL_=40.

The following SAS statements perform the regression analysis:

```
title1 'Stratified Sample with Single Sampling Unit in Strata';
title2 'With Stratum Collapse';
proc SURVEYREG data=Sample total=StratumTotals;
   strata Stratum/list;
   model Y=X;
   weight W;
run;
```

Output 86.6.1 shows that there are a total of five strata in the input data set and two strata are collapsed into a pooled stratum. The denominator degrees of freedom is 4, due to the collapse (see the section "Denominator Degrees of Freedom" on page 6569).

Output 86.6.1 Summary of Data and Regression

```
                Stratified Sample with Single Sampling Unit in Strata
                                With Stratum Collapse

                              The SURVEYREG Procedure

                    Regression Analysis for Dependent Variable Y

                                    Data Summary

                       Number of Observations             8
                       Sum of Weights             157.00000
                       Weighted Mean of Y           4.31210
                       Weighted Sum of Y          677.00000

                                  Design Summary

                     Number of Strata                       5
                     Number of Strata Collapsed             2

                                  Fit Statistics

                           R-square            0.9564
                           Root MSE            0.5111
                           Denominator DF           4
```

Output 86.6.2 displays the stratification information, including stratum collapse. Under the column Collapsed, the fourth stratum (Stratum=14) and the fifth (Stratum=33) are marked as 'Yes,' which indicates that these two strata are collapsed into the pooled stratum (Stratum Index=0). The sampling rate for the pooled stratum is 2% (see the section "Sampling Rate of the Pooled Stratum from Collapse" on page 6572).

Output 86.6.2 Stratification Information

```
                                Stratum Information

    Stratum                                                Population    Sampling
     Index      Collapsed      Stratum        N Obs           Total        Rate

       1                           10            2              10       20.0%
       2                           11            2              20       10.0%
       3                           12            2              32       6.25%
       4         Yes               14            1              50       2.00%
       5         Yes               33            1              40       2.50%

       0         Pooled                          2              90       2.22%

NOTE: Strata with only one observation are collapsed into the stratum with
      Stratum Index "0".
```

Output 86.6.3 displays the parameter estimates and the tests of the significance of the model effects.

Output 86.6.3 Parameter Estimates and Effect Tests

```
                    Tests of Model Effects

          Effect       Num DF    F Value    Pr > F

          Model             1     173.01    0.0002
          Intercept         1       0.00    0.9961
          X                 1     173.01    0.0002

NOTE: The denominator degrees of freedom for the F tests is 4.

              Estimated Regression Coefficients

                                Standard
 Parameter      Estimate          Error    t Value    Pr > |t|

 Intercept    0.00179469    0.34306373       0.01      0.9961
 X            1.12598708    0.08560466      13.15      0.0002

NOTE: The denominator degrees of freedom for the t tests is 4.
```

Alternatively, if you prefer not to collapse strata with a single sampling unit, you can specify the NOCOLLAPSE option in the STRATA statement:

```
title1 'Stratified Sample with Single Sampling Unit in Strata';
title2 'Without Stratum Collapse';
proc SURVEYREG data=Sample total=StratumTotals;
   strata Stratum/list nocollapse;
model Y = X;
weight W;
run;
```

Output 86.6.4 does not contain the stratum collapse information displayed in Output 86.6.1, and the denominator degrees of freedom are 3 instead of 4.

Output 86.6.4 Summary of Data and Regression

```
          Stratified Sample with Single Sampling Unit in Strata
                        Without Stratum Collapse

                        The SURVEYREG Procedure

               Regression Analysis for Dependent Variable Y

                              Data Summary

                  Number of Observations             8
                  Sum of Weights             157.00000
                  Weighted Mean of Y           4.31210
                  Weighted Sum of Y          677.00000
```

Output 86.6.4 *continued*

Design Summary	
Number of Strata	5
Fit Statistics	
R-square	0.9564
Root MSE	0.5111
Denominator DF	3

In Output 86.6.5, although the fourth stratum and the fifth stratum contain only one observation, no stratum collapse occurs.

Output 86.6.5 Stratification Information

Stratum Information

Stratum Index	Stratum	N Obs	Population Total	Sampling Rate
1	10	2	10	20.0%
2	11	2	20	10.0%
3	12	2	32	6.25%
4	14	1	50	2.00%
5	33	1	40	2.50%

As a result of not collapsing strata, the standard error estimates of the parameters, shown in Output 86.6.6, are different from those in Output 86.6.3, as are the tests of the significance of model effects.

Output 86.6.6 Parameter Estimates and Effect Tests

Tests of Model Effects

Effect	Num DF	F Value	Pr > F
Model	1	347.27	0.0003
Intercept	1	0.00	0.9962
X	1	347.27	0.0003

NOTE: The denominator degrees of freedom for the F tests is 3.

Output 86.6.6 *continued*

Estimated Regression Coefficients

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	0.00179469	0.34302581	0.01	0.9962
X	1.12598708	0.06042241	18.64	0.0003

NOTE: The denominator degrees of freedom for the t tests is 3.

Example 86.7: Domain Analysis

Recall that in the section "Getting Started: SURVEYREG Procedure" on page 6535, you collected a stratified simple random sample from a junior high school to examine how household income and the number of children in a household affect students' average weekly spending for ice cream. You can also use the same sample to estimate the average weekly spending among male and female students, respectively. This is often called domain analysis (subgroup analysis). You can use PROC SURVEYREG to perform domain analysis as in this example. The data set follows:

```
data IceCreamDataDomain;
   input Grade Spending Income Gender$ @@;
   datalines;
7   7  39  M   7   7  38  F   8  12  47  F
9  10  47  M   7   1  34  M   7  10  43  M
7   3  44  M   8  20  60  F   8  19  57  M
7   2  35  M   7   2  36  F   9  15  51  F
8  16  53  F   7   6  37  F   7   6  41  M
7   6  39  M   9  15  50  M   8  17  57  F
8  14  46  M   9   8  41  M   9   8  41  F
9   7  47  F   7   3  39  F   7  12  50  M
7   4  43  M   9  14  46  F   8  18  58  M
9   9  44  F   7   2  37  F   7   1  37  M
7   4  44  M   7  11  42  M   9   8  41  M
8  10  42  M   8  13  46  F   7   2  40  F
9   6  45  F   9  11  45  M   7   2  36  F
7   9  46  F
;
data IceCreamDataDomain;
   set IceCreamDataDomain;
   if Grade=7 then Prob=20/1824;
   if Grade=8 then Prob=9/1025;
   if Grade=9 then Prob=11/1151;
   Weight=1/Prob;
```

In the data set IceCreamDataDomain, the variable Grade indicates a student's grade, which is the stratification variable. The variable Spending contains the dollar amount of each student's average weekly spending for ice cream. The variable Income specifies the household income, in thousands

of dollars. The variable Gender indicates a student's gender. The sampling weights are created by using the reciprocals of the probabilities of selection, as follows:

```
data StudentTotals;
   input Grade _TOTAL_;
   datalines;
7 1824
8 1025
9 1151
;
```

In the data set StudentTotals, the variable Grade is the stratification variable, and the variable _TOTAL_ contains the total numbers of students in the strata in the survey population.

The following statements demonstrate how you can estimate the average spending in the subgroup of male students:

```
title1 'Ice Cream Spending Analysis';
title2 'Domain Analysis by Gender';
proc surveyreg data=IceCreamDataDomain total=StudentTotals;
   strata Grade;
   model Spending = Income;
   domain Gender;
run;
```

Output 86.7.1 gives a summary of the domains.

Output 86.7.1 Domain Analysis Summary

```
                    Ice Cream Spending Analysis
                     Domain Analysis by Gender

                      The SURVEYREG Procedure

                              Gender=F

            Domain Regression Analysis for Variable Spending

                            Domain Summary

          Number of Observations                          40
          Number of Observations in Domain                19
          Number of Observations Not in Domain            21
          Mean of Spending                           8.94737
          Sum of Spending                          170.00000
```

Output 86.7.1 *continued*

```
                    Ice Cream Spending Analysis
                     Domain Analysis by Gender

                      The SURVEYREG Procedure

                              Gender=M

           Domain Regression Analysis for Variable Spending

                            Domain Summary

          Number of Observations                          40
          Number of Observations in Domain                21
          Number of Observations Not in Domain            19
          Mean of Spending                           8.57143
          Sum of Spending                          180.00000
```

Output 86.7.2 shows that parameter estimates for the model within each domain.

Output 86.7.2 Parameter Estimates within Domain

```
                    Ice Cream Spending Analysis
                     Domain Analysis by Gender

                      The SURVEYREG Procedure

                              Gender=F

           Domain Regression Analysis for Variable Spending

                  Estimated Regression Coefficients

                                  Standard
   Parameter      Estimate          Error    t Value    Pr > |t|

   Intercept    -23.897418     2.38307272     -10.03      <.0001
   Income         0.737649     0.04973471      14.83      <.0001

  NOTE: The denominator degrees of freedom for the t tests is 37.
```

Output 86.7.2 *continued*

```
                    Ice Cream Spending Analysis
                     Domain Analysis by Gender

                      The SURVEYREG Procedure

                              Gender=M

           Domain Regression Analysis for Variable Spending

                  Estimated Regression Coefficients

                                  Standard
     Parameter      Estimate         Error    t Value    Pr > |t|

     Intercept    -23.342282    2.11458083     -11.04      <.0001
     Income         0.730052    0.04587826      15.91      <.0001

   NOTE: The denominator degrees of freedom for the t tests is 37.
```

Example 86.8: Variance Estimate Using the Jackknife Method

This example uses the stratified sample in the section "Stratified Sampling" on page 6538 to illustrate how to estimate the variances with replication methods.

As shown in the section "Stratified Sampling," the selected sample is saved in the SAS data set IceCream. The variable Grade that indicates a student's grade is the stratification variable. The variable Spending contains the dollar amount of each student's average weekly spending for ice cream. The variable Income specifies the household income, in thousands of dollars. The variable Kids indicates how many children are in a student's family. The variable Weight contains sampling weights.

In this example, we use the jackknife method to estimate the variance, saving the replicate weights generated by the procedure into a SAS data set:

```
title1 'Ice Cream Spending Analysis';
title2 'Use the jackknife method to estimate the variance';
proc surveyreg data=IceCream
   varmethod=JACKKNIFE(outweights=JKWeights);
   strata Grade;
   class Kids;
   model Spending = Income Kids / solution;
   weight Weight;
run;
```

The option VARMETHOD=JACKKNIFE requests the procedure to estimate the variance by using the jackknife method. The option OUTWEIGHTS=*JKWeights* provides a SAS data set named JKWeights that contains the replicate weights used in the computation.

Output 86.8.1 shows the summary of the data and the variance estimation method. There are a total of 40 replicates generated by the procedure.

Output 86.8.1 Variance Estimation Using the Jackknife Method

```
                    Ice Cream Spending Analysis
          Use the jackknife method to estimate the variance

                      The SURVEYREG Procedure

         Regression Analysis for Dependent Variable Spending

                           Data Summary

              Number of Observations                40
              Sum of Weights                     4000.0
              Weighted Mean of Spending         9.14130
              Weighted Sum of Spending          36565.2

                           Design Summary

                   Number of Strata              3

                        Variance Estimation

                 Method                    Jackknife
                 Number of Replicates             40
```

Output 86.8.2 displays the parameter estimates and their standard errors, as well as the tests of model effects that use the jackknife method.

Output 86.8.2 Variance Estimation Using the Jackknife Method

```
                        Tests of Model Effects

              Effect       Num DF    F Value    Pr > F

              Model             4     110.48    <.0001
              Intercept         1     133.30    <.0001
              Income            1     289.16    <.0001
              Kids              3       0.90    0.4525

     NOTE: The denominator degrees of freedom for the F tests is 37.
```

Output 86.8.2 *continued*

Estimated Regression Coefficients

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	-26.086882	2.58771182	-10.08	<.0001
Income	0.776699	0.04567521	17.00	<.0001
Kids 1	0.888631	1.12799263	0.79	0.4358
Kids 2	1.545726	1.25598146	1.23	0.2262
Kids 3	-0.526817	1.42555453	-0.37	0.7138
Kids 4	0.000000	0.00000000	.	.

NOTE: The denominator degrees of freedom for the t tests is 37.
Matrix X'WX is singular and a generalized inverse was used to solve the normal equations. Estimates are not unique.

Output 86.8.3 prints the first 6 observation in the output data set JKWeights, which contains the replicate weights.

The data set JKWeights contains all the variable in the data set IceCream, in addition to the replicate weights variables named RepWt_1, RepWt_2, ..., RepWt_40.

For example, the first observation (student) from stratum Grade=7 is deleted to create the first replicate. Therefore, stratum Grade=7 is the donor stratum for the first replicate, and the corresponding replicate weights are saved in the variable RepWt_1.

Because the first observation is deleted in the first replicate, RepWt_1=0 for the first observation. For observations from strata other than the donor stratum Grade=7, their replicate weights remain the same as in the variable Weight, while the rest of the observations in stratum Grade=7 are multiplied by the reciprocal of the corresponding jackknife coefficient, 0.95 for the first replicate.

Output 86.8.3 The Jackknife Replicate Weights for the First 6 Observations

The Jackknife Weights for the First 6 Obs

Obs	Grade	Spending	Income	Kids	Prob	Weight	RepWt_1	RepWt_2	RepWt_3	RepWt_4
1	7	7	39	2	0.010965	91.200	0.000	96.000	91.200	91.200
2	7	7	38	1	0.010965	91.200	96.000	0.000	91.200	91.200
3	8	12	47	1	0.008780	113.889	113.889	113.889	0.000	113.889
4	9	10	47	4	0.009557	104.636	104.636	104.636	104.636	0.000
5	7	1	34	4	0.010965	91.200	96.000	96.000	91.200	91.200
6	7	10	43	2	0.010965	91.200	96.000	96.000	91.200	91.200

Obs	RepWt_5	RepWt_6	RepWt_7	RepWt_8	RepWt_9	RepWt_ 10	RepWt_ 11	RepWt_ 12	RepWt_ 13
1	96.000	96.000	96.000	91.200	91.200	96.000	96.000	91.200	91.200
2	96.000	96.000	96.000	91.200	91.200	96.000	96.000	91.200	91.200
3	113.889	113.889	113.889	128.125	128.125	113.889	113.889	113.889	128.125
4	104.636	104.636	104.636	104.636	104.636	104.636	104.636	115.100	104.636
5	0.000	96.000	96.000	91.200	91.200	96.000	96.000	91.200	91.200
6	96.000	0.000	96.000	91.200	91.200	96.000	96.000	91.200	91.200

Obs	RepWt_ 14	RepWt_ 15	RepWt_ 16	RepWt_ 17	RepWt_ 18	RepWt_ 19	RepWt_ 20	RepWt_ 21	RepWt_ 22
1	96.000	96.000	96.000	91.200	91.200	91.200	91.200	91.200	91.200
2	96.000	96.000	96.000	91.200	91.200	91.200	91.200	91.200	91.200
3	113.889	113.889	113.889	113.889	128.125	128.125	113.889	113.889	113.889
4	104.636	104.636	104.636	115.100	104.636	104.636	115.100	115.100	115.100
5	96.000	96.000	96.000	91.200	91.200	91.200	91.200	91.200	91.200
6	96.000	96.000	96.000	91.200	91.200	91.200	91.200	91.200	91.200

Obs	RepWt_ 23	RepWt_ 24	RepWt_ 25	RepWt_ 26	RepWt_ 27	RepWt_ 28	RepWt_ 29	RepWt_ 30	RepWt_ 31
1	96.000	96.000	96.000	91.200	91.200	91.200	96.000	96.000	96.000
2	96.000	96.000	96.000	91.200	91.200	91.200	96.000	96.000	96.000
3	113.889	113.889	113.889	113.889	128.125	113.889	113.889	113.889	113.889
4	104.636	104.636	104.636	115.100	104.636	115.100	104.636	104.636	104.636
5	96.000	96.000	96.000	91.200	91.200	91.200	96.000	96.000	96.000
6	96.000	96.000	96.000	91.200	91.200	91.200	96.000	96.000	96.000

Obs	RepWt_ 32	RepWt_ 33	RepWt_ 34	RepWt_ 35	RepWt_ 36	RepWt_ 37	RepWt_ 38	RepWt_ 39	RepWt_ 40
1	96.000	91.200	91.200	91.200	96.000	91.200	91.200	96.000	96.000
2	96.000	91.200	91.200	91.200	96.000	91.200	91.200	96.000	96.000
3	113.889	113.889	128.125	128.125	113.889	113.889	113.889	113.889	113.889
4	104.636	115.100	104.636	104.636	104.636	115.100	115.100	104.636	104.636
5	96.000	91.200	91.200	91.200	96.000	91.200	91.200	96.000	96.000
6	96.000	91.200	91.200	91.200	96.000	91.200	91.200	96.000	96.000

References

Brick, J. M. and Kalton, G. (1996), "Handling Missing Data in Survey Research," *Statistical Methods in Medical Research*, 5, 215–238.

Cochran, W. G. (1977), *Sampling Techniques*, Third Edition, New York: John Wiley & Sons.

Dippo, C. S., Fay, R. E., and Morganstein, D. H. (1984), "Computing Variances from Complex Samples with Replicate Weights," *Proceedings of the Survey Research Methods Section, ASA*, 489–494.

Fay, R. E. (1984), "Some Properties of Estimators of Variance Based on Replication Methods," *Proceedings of the Survey Research Methods Section, ASA*, 495–500.

Fay, R. E. (1989), "Theory and Application of Replicate Weighting for Variance Calculations," *Proceedings of the Survey Research Methods Section, ASA*, 212–217.

Fuller, W. A. (1975), "Regression Analysis for Sample Survey," *Sankhyā*, 37 (3), Series C, 117–132.

Fuller, W. A., Kennedy, W., Schnell, D., Sullivan, G., and Park, H. J. (1989), *PC CARP*, Ames: Statistical Laboratory, Iowa State University.

Hidiroglou, M. A., Fuller, W. A., and Hickman, R. D. (1980), *SUPER CARP*, Ames: Statistical Laboratory, Iowa State University.

Judkins, D. (1990), "Fay's Method for Variance Estimation," *Journal of Official Statistics*, 6, 223–239.

Kalton, G., and Kaspyzyk, D. (1986), "The Treatment of Missing Survey Data," *Survey Methodology*, 12, 1–16.

Kish, L. (1965), *Survey Sampling*, New York: John Wiley & Sons.

Lohr, S. L. (1999), *Sampling: Design and Analysis*, Pacific Grove, CA: Duxbury Press.

Pringle, R. M. and Rayner, A. A. (1971), *Generalized Inverse Matrices with Applications to Statistics*, New York: Hafner Publishing.

Rao, J. N. K., and Shao, J. (1996), "On Balanced Half Sample Variance Estimation in Stratified Sampling," *Journal of the American Statistical Association,* 91, 343–348.

Rao, J. N. K. and Shao, J. (1999), "Modified Balanced Repeated Replication for Complex Survey Data," *Biometrika,* 86, 403–415.

Rao, J. N. K., Wu, C. F. J., and Yue, K. (1992), " Some Recent Work on Resampling Methods for Complex Surveys," *Survey Methodology,* 18, 209–217.

Särndal, C. E., Swensson, B., and Wretman, J. (1992), *Model Assisted Survey Sampling*, New York: Springer-Verlag.

Rust, K. (1985), "Variance Estimation for Complex Estimators in Sample Surveys," *Journal of Official Statistics,* 1, 381–397.

Wolter, K. M. (1985), *Introduction to Variance Estimation*, New York: Springer-Verlag.

Woodruff, R. S. (1971), "A Simple Method for Approximating the Variance of a Complicated Estimate," *Journal of the American Statistical Association*, 66, 411–414.

Chapter 87

The SURVEYSELECT Procedure

Contents

Overview: SURVEYSELECT Procedure **6614**
Getting Started: SURVEYSELECT Procedure **6615**
Simple Random Sampling 6616
Stratified Sampling 6618
Stratified Sampling with Control Sorting 6622
Syntax: SURVEYSELECT Procedure **6623**
PROC SURVEYSELECT Statement 6623
CONTROL Statement 6639
ID Statement 6639
SIZE Statement 6639
STRATA Statement 6640
Details: SURVEYSELECT Procedure **6644**
Missing Values 6644
Sorting by CONTROL Variables 6644
Sample Selection Methods 6645
Simple Random Sampling 6646
Unrestricted Random Sampling 6646
Systematic Random Sampling 6647
Sequential Random Sampling 6647
PPS Sampling without Replacement 6648
PPS Sampling with Replacement 6650
PPS Systematic Sampling 6650
PPS Sequential Sampling 6651
Brewer's PPS Method 6652
Murthy's PPS Method 6653
Sampford's PPS Method 6653
Sample Size Allocation 6654
Proportional Allocation 6654
Optimal Allocation 6655
Neyman Allocation 6656
Secondary Input Data Set 6656
Sample Output Data Set 6657
Allocation Output Data Set 6660
Displayed Output 6661

ODS Table Names 6663
Examples: SURVEYSELECT Procedure **6664**
Example 87.1: Replicated Sampling 6664
Example 87.2: PPS Selection of Two Units per Stratum 6667
Example 87.3: PPS (Dollar-Unit) Sampling 6670
Example 87.4: Proportional Allocation 6673
References **6676**

Overview: SURVEYSELECT Procedure

The SURVEYSELECT procedure provides a variety of methods for selecting probability-based random samples. The procedure can select a simple random sample or can sample according to a complex multistage sample design that includes stratification, clustering, and unequal probabilities of selection. With probability sampling, each unit in the survey population has a known, positive probability of selection. This property of probability sampling avoids selection bias and enables you to use statistical theory to make valid inferences from the sample to the survey population.

To select a sample with PROC SURVEYSELECT, you input a SAS data set that contains the sampling frame or list of units from which the sample is to be selected. You also specify the selection method, the desired sample size or sampling rate, and other selection parameters. PROC SURVEYSELECT selects the sample and produces an output data set that contains the selected units, their selection probabilities, and their sampling weights. When you select a sample in multiple stages, you invoke the procedure separately for each stage of selection, inputting the frame and selection parameters for each current stage.

PROC SURVEYSELECT provides methods for both equal probability sampling and probability proportional to size (PPS) sampling. In equal probability sampling, each unit in the sampling frame, or in a stratum, has the same probability of being selected for the sample. In PPS sampling, a unit's selection probability is proportional to its size measure. For details about probability sampling methods, see Lohr (1999), Kish (1965, 1987), Kalton (1983), and Cochran (1977).

PROC SURVEYSELECT provides the following equal probability sampling methods:

- simple random sampling
- unrestricted random sampling (with replacement)
- systematic random sampling
- sequential random sampling

This procedure also provides the following probability proportional to size (PPS) methods:

- PPS sampling without replacement
- PPS sampling with replacement
- PPS systematic sampling
- PPS algorithms for selecting two units per stratum
- sequential PPS sampling with minimum replacement

The procedure uses fast, efficient algorithms for these sample selection methods. Thus, it performs well even for large input data sets or sampling frames, which might occur in practice for large-scale sample surveys.

PROC SURVEYSELECT can perform stratified sampling by selecting samples independently within the specified strata, or nonoverlapping subgroups of the survey population. Stratification controls the distribution of the sample size in the strata. It is widely used in practice toward meeting a variety of survey objectives. For example, with stratification you can ensure adequate sample sizes for subgroups of interest, including small subgroups, or you can use stratification toward improving the precision of the overall estimates. When you use a systematic or sequential selection method, PROC SURVEYSELECT can also sort by control variables within strata for the additional control of implicit stratification.

For stratified sampling, PROC SURVEYSELECT provides survey design methods to allocate the total sample size among the strata. Available allocation methods include proportional, Neyman, and optimal allocation. Optimal allocation maximizes the estimation precision within the available resources, taking into account stratum sizes, costs, and variances.

PROC SURVEYSELECT provides replicated sampling, where the total sample is composed of a set of replicates, and each replicate is selected in the same way. You can use replicated sampling to study variable nonsampling errors, such as variability in the results obtained by different interviewers. You can also use replication to compute standard errors for the combined sample estimates.

Getting Started: SURVEYSELECT Procedure

In this example, an Internet service provider conducts a customer satisfaction survey. The survey population consists of the company's current subscribers. The company plans to select a sample of customers from this population, interview the selected customers, and then make inferences about the entire survey population from the sample data.

The SAS data set Customers contains the sampling frame, which is the list of units in the survey population. The sample of customers will be selected from this sampling frame. The data set Customers is constructed from the company's customer database. It contains one observation for each customer, with a total of 13,471 observations.

The following PROC PRINT statements display the first 10 observations of the data set Customers and produce Figure 87.1:

```
title1 'Customer Satisfaction Survey';
title2 'First 10 Observations';
proc print data=Customers(obs=10);
run;
```

Figure 87.1 Customers Data Set (First 10 Observations)

Customer Satisfaction Survey
First 10 Observations

Obs	CustomerID	State	Type	Usage
1	416-87-4322	AL	New	839
2	288-13-9763	GA	Old	224
3	339-00-8654	GA	Old	2451
4	118-98-0542	GA	New	349
5	421-67-0342	FL	New	562
6	623-18-9201	SC	New	68
7	324-55-0324	FL	Old	137
8	832-90-2397	AL	Old	1563
9	586-45-0178	GA	New	615
10	801-24-5317	SC	New	728

In the SAS data set Customers, the variable CustomerID uniquely identifies each customer. The variable State contains the state of the customer's address. The company has customers in four states: Georgia (GA), Alabama (AL), Florida (FL), and South Carolina (SC). The variable Type equals 'Old' if the customer has subscribed to the service for more than one year; otherwise, the variable Type equals 'New'. The variable Usage contains the customer's average monthly service usage, in minutes.

The following sections illustrate the use of PROC SURVEYSELECT for probability sampling with three different designs for the customer satisfaction survey. All three designs are one-stage, with customers as the sampling units. The first design is simple random sampling without stratification. In the second design, customers are stratified by state and type, and the sample is selected by simple random sampling within strata. In the third design, customers are sorted within strata by usage, and the sample is selected by systematic random sampling within strata.

Simple Random Sampling

The following PROC SURVEYSELECT statements select a probability sample of customers from the Customers data set by using simple random sampling:

```
title1 'Customer Satisfaction Survey';
title2 'Simple Random Sampling';
proc surveyselect data=Customers
   method=srs n=100 out=SampleSRS;
run;
```

The PROC SURVEYSELECT statement invokes the procedure. The DATA= option names the SAS data set Customers as the input data set from which to select the sample. The METHOD=SRS option specifies simple random sampling as the sample selection method. In simple random sampling, each unit has an equal probability of selection, and sampling is without replacement. Without-replacement sampling means that a unit cannot be selected more than once. The N=100 option specifies a sample size of 100 customers. The OUT= option stores the sample in the SAS data set named SampleSRS.

Figure 87.2 displays the output from PROC SURVEYSELECT, which summarizes the sample selection. A sample of 100 customers is selected from the data set Customers by simple random sampling. With simple random sampling and no stratification in the sample design, the selection probability is the same for all units in the sample. In this sample, the selection probability for each customer equals 0.007423, which is the sample size (100) divided by the population size (13,471). The sampling weight equals 134.71 for each customer in the sample, where the weight is the inverse of the selection probability. If you specify the STATS option, PROC SURVEYSELECT includes the selection probabilities and sampling weights in the output data set. (This information is always included in the output data set for more complex designs.)

The random number seed is 39647. PROC SURVEYSELECT uses this number as the initial seed for random number generation. Because the SEED= option is not specified in the PROC SURVEYSELECT statement, the seed value is obtained by using the time of day from the computer's clock. You can specify SEED=39647 to reproduce this sample.

Figure 87.2 Sample Selection Summary

```
                    Customer Satisfaction Survey
                       Simple Random Sampling

                     The SURVEYSELECT Procedure

                Selection Method    Simple Random Sampling

                   Input Data Set            CUSTOMERS
                   Random Number Seed            39647
                   Sample Size                     100
                   Selection Probability      0.007423
                   Sampling Weight              134.71
                   Output Data Set           SAMPLESRS
```

The sample of 100 customers is stored in the SAS data set SampleSRS. PROC SURVEYSELECT does not display this output data set. The following PROC PRINT statements display the first 20 observations of SampleSRS:

```
title1 'Customer Satisfaction Survey';
title2 'Sample of 100 Customers, Selected by SRS';
title3 '(First 20 Observations)';
proc print data=SampleSRS(obs=20);
run;
```

Figure 87.3 displays the first 20 observations of the output data set SampleSRS, which contains the sample of customers. This data set includes all the variables from the DATA= input data set Customers. If you do not want to include all variables, you can use the ID statement to specify which variables to copy from the input data set to the output (sample) data set.

Figure 87.3 Customer Sample (First 20 Observations)

Customer Satisfaction Survey
Sample of 100 Customers, Selected by SRS
(First 20 Observations)

Obs	CustomerID	State	Type	Usage
1	036-89-0212	FL	New	74
2	045-53-3676	AL	New	411
3	050-99-2380	GA	Old	167
4	066-93-5368	AL	Old	1232
5	082-99-9234	FL	New	90
6	097-17-4766	FL	Old	131
7	110-73-1051	FL	Old	102
8	111-91-6424	GA	New	247
9	127-39-4594	GA	New	61
10	162-50-3866	FL	New	100
11	162-56-1370	FL	New	224
12	167-21-6808	SC	New	60
13	168-02-5189	AL	Old	7553
14	174-07-8711	FL	New	284
15	187-03-7510	SC	New	21
16	190-78-5019	GA	New	185
17	200-75-0054	GA	New	224
18	201-14-1003	GA	Old	3437
19	207-15-7701	GA	Old	24
20	211-14-1373	AL	Old	88

Stratified Sampling

In this section, stratification is added to the sample design for the customer satisfaction survey. The sampling frame, or list of all customers, is stratified by State and Type. This divides the sampling frame into nonoverlapping subgroups formed from the values of the State and Type variables. Samples are then selected independently within the strata.

PROC SURVEYSELECT requires that the input data set be sorted by the STRATA variables. The following PROC SORT statements sort the Customers data set by the stratification variables State and Type:

```
proc sort data=Customers;
   by State Type;
run;
```

The following PROC FREQ statements display the crosstabulation of the Customers data set by State and Type:

```
title1 'Customer Satisfaction Survey';
title2 'Strata of Customers';
proc freq data=Customers;
   tables State*Type;
run;
```

Figure 87.4 presents the table of State by Type for the 13,471 customers. There are four states and two levels of Type, forming a total of eight strata.

Figure 87.4 Stratification of Customers by State and Type

```
                    Customer Satisfaction Survey
                        Strata of Customers

                          The FREQ Procedure

                       Table of State by Type

State     Type

Frequency|
Percent  |
Row Pct  |
Col Pct  |New     |Old     |  Total
---------+--------+--------+
AL       |   1238 |    706 |   1944
         |   9.19 |   5.24 |  14.43
         |  63.68 |  36.32 |
         |  14.43 |  14.43 |
---------+--------+--------+
FL       |   2170 |   1370 |   3540
         |  16.11 |  10.17 |  26.28
         |  61.30 |  38.70 |
         |  25.29 |  28.01 |
---------+--------+--------+
GA       |   3488 |   1940 |   5428
         |  25.89 |  14.40 |  40.29
         |  64.26 |  35.74 |
         |  40.65 |  39.66 |
---------+--------+--------+
SC       |   1684 |    875 |   2559
         |  12.50 |   6.50 |  19.00
         |  65.81 |  34.19 |
         |  19.63 |  17.89 |
---------+--------+--------+
Total        8580     4891    13471
            63.69    36.31   100.00
```

The following PROC SURVEYSELECT statements select a probability sample of customers from the Customers data set according to the stratified sample design:

```
title1 'Customer Satisfaction Survey';
title2 'Stratified Sampling';
proc surveyselect data=Customers
      method=srs n=15
      seed=1953 out=SampleStrata;
   strata State Type;
run;
```

The STRATA statement names the stratification variables State and Type. In the PROC SURVEYSELECT statement, the METHOD=SRS option specifies simple random sampling. The N=15 option specifies a sample size of 15 customers for each stratum. If you want to specify different sample sizes for different strata, you can use the N=*SAS-data-set* option to name a secondary data set that contains the stratum sample sizes. The SEED=1953 option specifies '1953' as the initial seed for random number generation.

Figure 87.5 displays the output from PROC SURVEYSELECT, which summarizes the sample selection. A total of 120 customers are selected.

Figure 87.5 Sample Selection Summary

```
                    Customer Satisfaction Survey
                        Stratified Sampling

                     The SURVEYSELECT Procedure

                Selection Method    Simple Random Sampling
                Strata Variables    State
                                    Type

                   Input Data Set            CUSTOMERS
                   Random Number Seed             1953
                   Stratum Sample Size              15
                   Number of Strata                  8
                   Total Sample Size               120
                   Output Data Set         SAMPLESTRATA
```

The following PROC PRINT statements display the first 30 observations of the output data set SampleStrata:

```
title1 'Customer Satisfaction Survey';
title2 'Sample Selected by Stratified Design';
title3 '(First 30 Observations)';
proc print data=SampleStrata(obs=30);
run;
```

Figure 87.6 displays the first 30 observations of the output data set SampleStrata, which contains the sample of 120 customers, 15 customers from each of the eight strata. The variable SelectionProb contains the selection probability for each customer in the sample. Because customers are selected with equal probability within strata in this design, the selection probability equals the stratum sample size (15) divided by the stratum population size. The selection probabilities differ from stratum to stratum because the stratum population sizes differ. The selection probability for each customer in the first stratum (State='AL' and Type='New') is 0.012116, and the selection probability for customers in the second stratum is 0.021246. The variable SamplingWeight contains the sampling weights, which are computed as inverse selection probabilities.

Figure 87.6 Customer Sample (First 30 Observations)

Customer Satisfaction Survey
Sample Selected by Stratified Design
(First 30 Observations)

Obs	State	Type	CustomerID	Usage	Selection Prob	Sampling Weight
1	AL	New	002-26-1498	1189	0.012116	82.5333
2	AL	New	070-86-8494	106	0.012116	82.5333
3	AL	New	121-28-6895	76	0.012116	82.5333
4	AL	New	131-79-7630	265	0.012116	82.5333
5	AL	New	211-88-4991	108	0.012116	82.5333
6	AL	New	222-81-3742	83	0.012116	82.5333
7	AL	New	238-46-3776	278	0.012116	82.5333
8	AL	New	370-01-0671	123	0.012116	82.5333
9	AL	New	407-07-5479	1580	0.012116	82.5333
10	AL	New	550-90-3188	177	0.012116	82.5333
11	AL	New	582-40-9610	46	0.012116	82.5333
12	AL	New	672-59-9114	66	0.012116	82.5333
13	AL	New	848-60-3119	28	0.012116	82.5333
14	AL	New	886-83-4909	170	0.012116	82.5333
15	AL	New	993-31-7677	64	0.012116	82.5333
16	AL	Old	124-60-0495	80	0.021246	47.0667
17	AL	Old	128-54-9590	56	0.021246	47.0667
18	AL	Old	204-05-4017	17	0.021246	47.0667
19	AL	Old	210-68-8704	4363	0.021246	47.0667
20	AL	Old	239-75-4343	430	0.021246	47.0667
21	AL	Old	317-70-6496	452	0.021246	47.0667
22	AL	Old	365-37-1340	21	0.021246	47.0667
23	AL	Old	399-78-7900	108	0.021246	47.0667
24	AL	Old	404-90-6273	824	0.021246	47.0667
25	AL	Old	421-04-8548	1332	0.021246	47.0667
26	AL	Old	604-48-0587	16	0.021246	47.0667
27	AL	Old	774-04-0162	318	0.021246	47.0667
28	AL	Old	849-66-4156	79	0.021246	47.0667
29	AL	Old	937-69-9106	182	0.021246	47.0667
30	AL	Old	985-09-8691	24	0.021246	47.0667

Stratified Sampling with Control Sorting

The next sample design for the customer satisfaction survey uses stratification by State, as well as control sorting by Type and Usage within State. After stratification and control sorting, customers are selected by systematic random sampling within strata. Selection by systematic sampling, together with control sorting before selection, spreads the sample uniformly over the range of type and usage values within each stratum or state. The following PROC SURVEYSELECT statements select a probability sample of customers from the Customers data set according to this design:

```
title1 'Customer Satisfaction Survey';
title2 'Stratified Sampling with Control Sorting';
proc surveyselect data=Customers
      method=sys rate=.02
      seed=1234 out=SampleControl;
   strata State;
   control Type Usage;
run;
```

The STRATA statement names the stratification variable State. The CONTROL statement names the control variables Type and Usage. In the PROC SURVEYSELECT statement, the METHOD=SYS option requests systematic random sampling. The RATE=.02 option specifies a sampling rate of 2% for each stratum. The SEED=1234 option specifies the initial seed for random number generation.

Figure 87.7 displays the output from PROC SURVEYSELECT, which summarizes the sample selection. A sample of 271 customers is selected by using systematic random sampling within strata determined by State. The sampling frame Customers is sorted by control variables Type and Usage within strata. The type of sorting is serpentine, which is the default when SORT=NEST is not specified. See the section "Sorting by CONTROL Variables" on page 6644 for a description of serpentine sorting. The sorted data set replaces the input data set. (To leave the input data set unsorted and store the sorted input data in another data set, use the OUTSORT= option.) The output data set SampleControl contains the sample of customers.

Figure 87.7 Sample Selection Summary

```
                    Customer Satisfaction Survey
                Stratified Sampling with Control Sorting

                     The SURVEYSELECT Procedure

          Selection Method      Systematic Random Sampling
          Strata Variable       State
          Control Variables     Type
                                Usage
          Control Sorting       Serpentine

               Input Data Set                 CUSTOMERS
               Random Number Seed                  1234
               Stratum Sampling Rate               0.02
               Number of Strata                       4
               Total Sample Size                    271
               Output Data Set            SAMPLECONTROL
```

Syntax: SURVEYSELECT Procedure

The following statements are available in PROC SURVEYSELECT:

PROC SURVEYSELECT *options* ;
  STRATA *variables* < / *options* > ;
  CONTROL *variables* ;
  SIZE *variable* ;
  ID *variables* ;

The PROC SURVEYSELECT statement invokes the procedure and optionally identifies input and output data sets. It also specifies the selection method, the sample size, and other sample design parameters. The SURVEYSELECT statement is required.

The SIZE statement identifies the variable that contains the size measures of the sampling units. It is required for any selection method that is probability proportional to size (PPS).

The remaining statements are optional. The STRATA statement identifies a variable or set of variables that stratify the input data set. When you specify a STRATA statement, PROC SURVEYSELECT selects samples independently from the strata formed by the STRATA variables. The STRATA statement also provides options to allocate the total sample size among the strata.

The CONTROL statement identifies variables for ordering units within strata. It can be used for systematic and sequential sampling methods. The ID statement identifies variables to copy from the input data set to the output data set of selected units.

The rest of this section gives detailed syntax information about the CONTROL, ID, SIZE, and STRATA statements in alphabetical order after the description of the PROC SURVEYSELECT statement.

PROC SURVEYSELECT Statement

PROC SURVEYSELECT *options* ;

The PROC SURVEYSELECT statement invokes the procedure and optionally identifies input and output data sets. If you do not name a DATA= input data set, the procedure selects the sample from the most recently created SAS data set. If you do not name an OUT= output data set to contain the sample of selected units, the procedure still creates an output data set and names it according to the DATA*n* convention.

The PROC SURVEYSELECT statement also specifies the sample selection method, the sample size, and other sample design parameters. If you do not specify a selection method, PROC SURVEYSELECT uses simple random sampling (METHOD=SRS) if there is no SIZE statement. If you do specify a SIZE statement and do not specify a selection method, PROC SURVEYSELECT uses probability proportional to size selection without replacement (METHOD=PPS). You must specify the sample size or sampling rate unless you request a method that selects two units from each stratum (METHOD=PPS_BREWER or METHOD=PPS_MURTHY).

You can use the SAMPSIZE=*n* option to specify the sample size, or you can use the SAMPSIZE=*SAS-data-set* option to name a secondary input data set that contains stratum sample sizes. You can also specify stratum sampling rates, minimum size measures, maximum size measures, and certainty size measures in the secondary input data set. See the descriptions of the SAMPSIZE=, SAMPRATE=, MINSIZE=, MAXSIZE=, CERTSIZE=, and CERTSIZE=P= options for more information. You can name only one secondary input data set in each invocation of the procedure. See the section "Secondary Input Data Set" on page 6656 for details.

Table 87.1 lists the options available with the PROC SURVEYSELECT statement. Descriptions follow in alphabetical order.

Table 87.1 PROC SURVEYSELECT Statement Options

Task	Options
Specify the input data set	DATA=
Specify output data sets	OUT= OUTSORT=
Suppress displayed output	NOPRINT
Specify selection method	METHOD=
Specify sample size	SAMPSIZE= SELECTALL
Specify sampling rate	SAMPRATE= NMIN= NMAX=
Specify number of replicates	REPS=
Adjust size measures	MINSIZE= MAXSIZE=
Specify certainty size measures	CERTSIZE= CERTSIZE=P=
Specify sorting type	SORT=
Specify random number seed	SEED=
Control OUT= contents	JTPROBS OUTALL OUTHITS OUTSEED OUTSIZE STATS

You can specify the following options in the PROC SURVEYSELECT statement.

CERTSIZE

requests certainty selection, where the certainty size values are provided in the secondary input data set. Use the CERTSIZE option when you have already named the secondary data set in another option, such as the SAMPSIZE=*SAS-data-set* option. See the section "Secondary Input Data Set" on page 6656 for details.

In certainty selection, PROC SURVEYSELECT automatically selects all sampling units with size measures greater than or equal to the stratum certainty size values. After identifying the certainty units, PROC SURVEYSELECT selects the remainder of the sample according to the method specified in the METHOD= option. The CERTSIZE option is available for METHOD=PPS and METHOD=PPS_SAMPFORD.

You provide the stratum certainty size values in the secondary input data set variable _CERTSIZE_. Each certainty size value must be a positive number. The variable Certain in the OUT= data set identifies the certainty selections, which have selection probabilities equal to 1.

If you want to specify a single certainty size value for all strata, you can use the CERTSIZE=*certain* option.

CERTSIZE=*certain*

specifies the certainty size value. PROC SURVEYSELECT automatically selects all sampling units with size measures greater than or equal to the value *certain*. After identifying the certainty units, PROC SURVEYSELECT selects the remainder of the sample according to the method specified in the METHOD= option. The CERTSIZE= option is available for METHOD=PPS and METHOD=PPS_SAMPFORD.

The value of *certain* must be a positive number. The variable Certain in the OUT= data set identifies the certainty selections, which have selection probabilities equal to 1.

If you request a stratified sample design with the STRATA statement and specify the CERTSIZE=*certain* option, PROC SURVEYSELECT uses the value *certain* for all strata. If you do not want to use the same certainty size for all strata, use the CERTSIZE=*SAS-data-set* option to specify a certainty size value for each stratum.

CERTSIZE=*SAS-data-set*

names a SAS data set that contains certainty size values for the strata. PROC SURVEYSELECT automatically selects all sampling units with size measures greater than or equal to the stratum certainty size values. After identifying the certainty units, PROC SURVEYSELECT selects the remainder of the sample according to the method specified in the METHOD= option. The CERTSIZE= option is available for METHOD=PPS and METHOD=PPS_SAMPFORD.

You provide the stratum certainty size values in the CERTSIZE= data set variable _CERTSIZE_. Each certainty size value must be a positive number. The variable Certain in the OUT= data set identifies the certainty selections, which have selection probabilities equal to 1.

The CERTSIZE= input data set should contain all the STRATA variables, with the same type and length as in the DATA= data set. The STRATA groups should appear in the same order in the CERTSIZE= data set as in the DATA= data set. The CERTSIZE= data set must include a variable named _CERTSIZE_ that contains the certainty size value for each stratum. The CERTSIZE= data set is a secondary input data set. See the section "Secondary Input Data Set" on page 6656 for details. You can name only one secondary input data set in each invocation of the procedure.

If you want to specify a single certainty size value for all strata, you can use the CERTSIZE=*certain* option.

CERTSIZE=P

requests certainty proportion selection, where the stratum certainty proportions are provided in the secondary input data set. Use the CERTSIZE=P option when you have already named the secondary data set in another option, such as the SAMPSIZE=*SAS-data-set* option. See the section "Secondary Input Data Set" on page 6656 for details.

In certainty proportion selection, PROC SURVEYSELECT automatically selects all sampling units with size measures greater than or equal to the stratum certainty proportion of the total stratum size. The procedure repeats this process with the remaining units until no more certainty units are selected. After identifying the certainty units, PROC SURVEYSELECT selects the remainder of the sample according to the method specified in the METHOD= option. The CERTSIZE=P option is available for METHOD=PPS and METHOD=PPS_SAMPFORD.

You provide the stratum certainty proportions in the secondary input data set variable _CERTP_. Each certainty proportion must be a positive number. You can specify a proportion value as a number between 0 and 1. Or you can specify a proportion value in percentage form as a number between 1 and 100, and PROC SURVEYSELECT converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

The variable Certain in the OUT= data set identifies the certainty selections, which have selection probabilities equal to 1.

If you want to specify a single certainty proportion for all strata, you can use the CERTSIZE=P=*p* option.

CERTSIZE=P=*p*

specifies the certainty proportion. PROC SURVEYSELECT automatically selects all sampling units with size measures greater than or equal to the proportion p of the total stratum size. The procedure repeats this process with the remaining units until no more certainty units are selected. After identifying the certainty units, PROC SURVEYSELECT selects the remainder of the sample according to the method specified in the METHOD= option. The CERTSIZE=P= option is available for METHOD=PPS and METHOD=PPS_SAMPFORD.

The value of p must be a positive number. You can specify p as a number between 0 and 1. Or you can specify p in percentage form as a number between 1 and 100, and PROC SURVEYSELECT converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

The variable Certain in the OUT= data set identifies the certainty selections, which have selection probabilities equal to 1.

If you request a stratified sample design with the STRATA statement and specify the CERTSIZE=P=p option, PROC SURVEYSELECT uses the certainty proportion p for all strata. If you do not want to use the same certainty proportion for all strata, use the CERTSIZE=P=*SAS-data-set* option to specify a certainty proportion for each stratum.

CERTSIZE=P=*SAS-data-set*

names a SAS data set that contains the certainty proportions for the strata. PROC SURVEYSELECT automatically selects all sampling units with size measures greater than

or equal to the certainty proportion of the total stratum size. The procedure repeats this process with the remaining units until no more certainty units are selected. After identifying the certainty units, PROC SURVEYSELECT selects the remainder of the sample according to the method specified in the METHOD= option. The CERTSIZE=P= option is available for METHOD=PPS and METHOD=PPS_SAMPFORD.

You provide the stratum certainty proportions in the CERTSIZE=P= data set variable _CERTP_. Each certainty proportion must be a positive number. You can specify a proportion value as a number between 0 and 1. Or you can specify a proportion value in percentage form as a number between 1 and 100, and PROC SURVEYSELECT converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

The variable Certain in the OUT= data set identifies the certainty selections, which have selection probabilities equal to 1.

The CERTSIZE=P= input data set should contain all the STRATA variables, with the same type and length as in the DATA= data set. The STRATA groups should appear in the same order in the CERTSIZE=P= data set as in the DATA= data set. The CERTSIZE=P= data set must include a variable named _CERTP_ that contains the certainty proportion for each stratum. The CERTSIZE=P= data set is a secondary input data set. See the section "Secondary Input Data Set" on page 6656 for details. You can name only one secondary input data set in each invocation of the procedure.

If you want to specify a single certainty proportion for all strata, you can use the CERTSIZE=P=*p* option.

DATA=*SAS-data-set*

names the SAS data set from which PROC SURVEYSELECT selects the sample. If you omit the DATA= option, the procedure uses the most recently created SAS data set. In sampling terminology, the input data set is the *sampling frame*, or list of units from which the sample is selected.

JTPROBS

includes joint probabilities of selection in the OUT= output data set. This option is available for the following probability proportional to size selection methods: METHOD=PPS, METHOD=PPS_SAMPFORD, and METHOD=PPS_WR. By default, PROC SURVEYSELECT outputs joint selection probabilities for METHOD=PPS_BREWER and METHOD=PPS_MURTHY, which select two units per stratum.

For details about computation of joint selection probabilities for a particular sampling method, see the method description in the section "Sample Selection Methods" on page 6645. For more information about the contents of the output data set, see the section "Sample Output Data Set" on page 6657.

MAXSIZE

requests adjustment of size measures according to the stratum maximum size values provided in the secondary input data set. Use the MAXSIZE option when you have already named the secondary input data set in another option, such as the SAMPSIZE=*SAS-data-set* option. See the section "Secondary Input Data Set" on page 6656 for details.

You provide the stratum maximum size values in the secondary input data set variable _MAXSIZE_. Each maximum size value must be a positive number.

When a size measure exceeds the specified maximum value for its stratum, PROC SURVEYSELECT adjusts the size measure downward to equal the maximum size value. The variable AdjustedSize in the OUT= data set contains the adjusted size measures.

The MAXSIZE option is available when you use a SIZE statement for probability proportional to size selection and a STRATA statement.

If you want to specify a single maximum size value for all strata, you can use the MAXSIZE=*max* option.

MAXSIZE=*max*

specifies the maximum size value. The value of *max* must be a positive number.

When any size measure exceeds the value *max*, PROC SURVEYSELECT adjusts the size measure downward to equal *max*. The variable AdjustedSize in the OUT= data set contains the adjusted size measures.

The MAXSIZE=*max* option is available when you use a SIZE statement for selection with probability proportional to size.

If you request a stratified sample design with the STRATA statement and specify the MAXSIZE=*max* option, PROC SURVEYSELECT uses the maximum size *max* for all strata. If you do not want to use the same maximum size for all strata, use the MAXSIZE=*SAS-data-set* option to specify a maximum size value for each stratum.

MAXSIZE=*SAS-data-set*

names a SAS data set that contains the maximum size values for the strata. You provide the stratum maximum size values in the MAXSIZE= data set variable _MAXSIZE_. Each maximum size value must be a positive number.

When any size measure exceeds the maximum size value for its stratum, PROC SURVEYSELECT adjusts the size measure downward to equal the maximum size value. The variable AdjustedSize in the OUT= data set contains the adjusted size measures.

The MAXSIZE=*SAS-data-set* option is available when you use a SIZE statement for probability proportional to size selection and a STRATA statement for stratified selection.

The MAXSIZE= input data set should contain all the STRATA variables, with the same type and length as in the DATA= data set. The STRATA groups should appear in the same order in the MAXSIZE= data set as in the DATA= data set. The MAXSIZE= data set must include a variable named _MAXSIZE_ that contains the maximum size value for each stratum. The MAXSIZE= data set is a secondary input data set. See the section "Secondary Input Data Set" on page 6656 for details. You can name only one secondary input data set in each invocation of the procedure.

If you want to specify a single maximum size value for all strata, you can use the MAXSIZE=*max* option.

METHOD=*name*

M=*name*

specifies the method for sample selection. If you do not specify the METHOD= option, by default, PROC SURVEYSELECT uses simple random sampling (METHOD=SRS) if there is no SIZE statement. If you specify a SIZE statement, the default selection method is probability proportional to size without replacement (METHOD=PPS).

Valid values for *name* are as follows:

PPS

requests selection with probability proportional to size and without replacement. See the section "PPS Sampling without Replacement" on page 6648 for details. If you specify METHOD=PPS, you must name the size measure variable in the SIZE statement.

PPS_BREWER

BREWER

requests selection according to Brewer's method. Brewer's method selects two units from each stratum with probability proportional to size and without replacement. See the section "Brewer's PPS Method" on page 6652 for details. If you specify METHOD=PPS_BREWER, you must name the size measure variable in the SIZE statement. You do not need to specify the sample size with the SAMPSIZE= option, because Brewer's method selects two units from each stratum.

PPS_MURTHY

MURTHY

requests selection according to Murthy's method. Murthy's method selects two units from each stratum with probability proportional to size and without replacement. See the section "Murthy's PPS Method" on page 6653 for details. If you specify METHOD=PPS_MURTHY, you must name the size measure variable in the SIZE statement. You do not need to specify the sample size with the SAMPSIZE= option, because Murthy's method selects two units from each stratum.

PPS_SAMPFORD

SAMPFORD

requests selection according to Sampford's method. Sampford's method selects units with probability proportional to size and without replacement. See the section "Sampford's PPS Method" on page 6653 for details. If you specify METHOD=PPS_SAMPFORD, you must name the size measure variable in the SIZE statement.

PPS_SEQ

CHROMY

requests sequential selection with probability proportional to size and with minimum replacement. This method is also known as Chromy's method. See the section "PPS Sequential Sampling" on page 6651 for details. If you specify METHOD=PPS_SEQ, you must name the size measure variable in the SIZE statement.

PPS_SYS

requests systematic selection with probability proportional to size. See the section "PPS Systematic Sampling" on page 6650 for details. If you specify METHOD=PPS_SYS, you must name the size measure variable in the SIZE statement.

PPS_WR

requests selection with probability proportional to size and with replacement. See the section "PPS Sampling with Replacement" on page 6650 for details. If you specify METHOD=PPS_WR, you must name the size measure variable in the SIZE statement.

SEQ

requests sequential selection according to Chromy's method. If you specify METHOD=SEQ and do not specify a size measure variable with the SIZE statement, PROC SURVEYSELECT uses sequential zoned selection with equal probability and without replacement. See the section "Sequential Random Sampling" on page 6647 for details. If you specify METHOD=SEQ and also name a size measure variable in the SIZE statement, PROC SURVEYSELECT uses METHOD=PPS_SEQ, which is sequential selection with probability proportional to size and with minimum replacement. See the section "PPS Sequential Sampling" on page 6651 for more information.

SRS

requests simple random sampling, which is selection with equal probability and without replacement. See the section "Simple Random Sampling" on page 6646 for details. METHOD=SRS is the default if you do not specify the METHOD= option and also do not specify a SIZE statement.

SYS

requests systematic random sampling. If you specify METHOD=SYS and do not specify a size measure variable with the SIZE statement, PROC SURVEYSELECT uses systematic selection with equal probability. See the section "Systematic Random Sampling" on page 6647 for more information. If you specify METHOD=SYS and also name a size measure variable in the SIZE statement, PROC SURVEYSELECT uses METHOD=PPS_SYS, which is systematic selection with probability proportional to size. See the section "PPS Systematic Sampling" on page 6650 for details.

URS

requests unrestricted random sampling, which is selection with equal probability and with replacement. See the section "Unrestricted Random Sampling" on page 6646 for details.

MINSIZE

requests adjustment of size measures according to the stratum minimum size values provided in the secondary input data set. Use the MINSIZE option when you have already named the secondary input data set in another option, such as the SAMPSIZE=*SAS-data-set* option. See the section "Secondary Input Data Set" on page 6656 for details.

You provide the stratum minimum size values in the secondary input data set variable _MINSIZE_. Each minimum size value must be a positive number.

When a size measure is less than the specified minimum value for its stratum, PROC SURVEYSELECT adjusts the size measure upward to equal the minimum size value. The variable AdjustedSize in the OUT= data set contains the adjusted size measures.

The MINSIZE option is available when you specify a SIZE statement for probability proportional to size selection and a STRATA statement.

If you want to specify a single minimum size value for all strata, you can use the MINSIZE=*min* option.

MINSIZE=*min*

specifies the minimum size value. The value of *min* must be a positive number.

When any size measure is less than the value *min*, PROC SURVEYSELECT adjusts the size measure upward to equal *min*. The variable AdjustedSize in the OUT= data set contains the adjusted size measures.

The MINSIZE=*min* option is available when you specify a SIZE statement for selection with probability proportional to size.

If you request a stratified sample design with the STRATA statement and specify the MINSIZE=*min* option, PROC SURVEYSELECT uses the minimum size *min* for all strata. If you do not want to use the same minimum size for all strata, use the MINSIZE=*SAS-data-set* option to specify a minimum size value for each stratum.

MINSIZE=*SAS-data-set*

names a SAS data set that contains the minimum size values for the strata. You provide the stratum minimum size values in the MINSIZE= data set variable _MINSIZE_. Each minimum size value must be a positive number.

When any size measure is less than the minimum size value for its stratum, PROC SURVEYSELECT adjusts the size measure upward to equal the minimum size measure. The variable AdjustedSize in the OUT= data set contains the adjusted size measures.

The MINSIZE=*SAS-data-set* option is available when you specify a SIZE statement for probability proportional to size selection and a STRATA statement.

The MINSIZE= input data set should contain all the STRATA variables, with the same type and length as in the DATA= data set. The STRATA groups should appear in the same order in the MINSIZE= data set as in the DATA= data set. The MINSIZE= data set must include a variable named _MINSIZE_ that contains the minimum size measure for each stratum. The MINSIZE= data set is a secondary input data set. See the section "Secondary Input Data Set" on page 6656 for details. You can name only one secondary input data set in each invocation of the procedure.

If you want to specify a single minimum size measure for all strata, you can use the MINSIZE=*min* option.

NMAX=*n*

specifies the maximum stratum sample size *n* for the SAMPRATE= option. When you specify the SAMPRATE= option, PROC SURVEYSELECT calculates the stratum sample size by multiplying the total number of units in the stratum by the specified sampling rate. If this sample size is greater than the value NMAX=*n*, then PROC SURVEYSELECT selects only *n* units.

The maximum sample size *n* must be a positive integer. The NMAX= option is available only with the SAMPRATE= option, which can be used with equal probability selection methods (METHOD=SRS, METHOD=URS, METHOD=SYS, and METHOD=SEQ).

NMIN=*n*

specifies the minimum stratum sample size *n* for the SAMPRATE= option. When you specify the SAMPRATE= option, PROC SURVEYSELECT calculates the stratum sample size by multiplying the total number of units in the stratum by the specified sampling rate. If this sample size is less than the value NMIN=*n*, then PROC SURVEYSELECT selects *n* units.

The minimum sample size *n* must be a positive integer. The NMIN= option is available only with the SAMPRATE= option, which can be used with equal probability selection methods (METHOD=SRS, METHOD=URS, METHOD=SYS, and METHOD=SEQ).

NOPRINT

suppresses the display of all output. You can use the NOPRINT option when you want only to create an output data set. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 20, "Using the Output Delivery System."

OUT=*SAS-data-set*

names the output data set that contains the sample. If you omit the OUT= option, the data set is named DATA*n*, where *n* is the smallest integer that makes the name unique.

The output data set contains the units selected for the sample, as well as design information and selection statistics, depending on the selection method and output options you specify. See descriptions of the options JTPROBS, OUTHITS, OUTSEED, OUTSIZE, and STATS, which specify information to include in the output data set. See the section "Sample Output Data Set" on page 6657 for details about the contents of the output data set.

By default, the output data set contains only those units selected for the sample. To include all observations from the input data set in the output data set, use the OUTALL option.

By default, the output data set includes one observation for each unit selected. When the unit is selected multiple times, which can occur when you use with-replacement or with-minimum-replacement selection methods, the OUT= data set variable NumberHits contains the number of hits or selections for each unit. To produce a separate observation for each hit or selection, specify the OUTHITS option.

If you specify the NOSAMPLE option in the STRATA statement, PROC SURVEYFREQ allocates the total sample size among the strata but does not select the sample. In this case, the OUT= data set contains the allocated sample sizes. See the section "Allocation Output Data Set" on page 6660 for details.

OUTALL

includes all observations from the input data set in the output data set. By default, the output data set includes only those observations selected for the sample. When you specify the OUTALL option, the output data set includes all observations from DATA= and also contains a variable to indicate each observation's selection status. The variable Selected equals 1 for an observation selected for the sample, and equals 0 for an observation not selected. For information about the contents of the output data set, see the section "Sample Output Data Set" on page 6657.

The OUTALL option is available only for equal probability selection methods (METHOD=SRS, METHOD=URS, METHOD=SYS, and METHOD=SEQ).

OUTHITS

includes a separate observation in the output data set for each selection when the same unit is selected more than once. A unit can be selected more than once only by methods that select with replacement or with minimum replacement, which include METHOD=URS, METHOD=PPS_WR, METHOD=PPS_SYS, and METHOD=PPS_SEQ.

By default, the output data set contains one observation for each selected unit, even if it is selected more than once, and the variable NumberHits contains the number of hits or selections for that unit. See the section "Sample Output Data Set" on page 6657 for details about the contents of the output data set.

The OUTHITS option is available for selection methods that select with replacement or with minimum replacement (METHOD=URS, METHOD=PPS_WR, METHOD=PPS_SYS, and METHOD=PPS_SEQ).

OUTSEED

includes the initial seed for each stratum in the output data set. The variable InitialSeed contains the stratum initial seeds. See the section "Sample Output Data Set" on page 6657 for details about the contents of the output data set.

To reproduce the same sample for any stratum in a subsequent execution of PROC SURVEYSELECT, you can specify the same stratum initial seed with the SEED=*SAS-data-set* option, along with the same sample selection parameters. See the section "Sample Selection Methods" on page 6645 for information about initial seeds and random number generation in PROC SURVEYSELECT.

The "Sample Selection Summary" table displays the initial random number seed for the entire sample selection, which is the same as the initial seed for the first stratum when the design is stratified. To reproduce the entire sample, you can specify this same seed value in the SEED= option, along with the same sample selection parameters.

OUTSIZE

includes additional design and sampling frame parameters in the output data set. If you specify the OUTSIZE option, PROC SURVEYSELECT includes the sample size or sampling rate in the output data set. When you specify the OUTSIZE option and also specify the SIZE statement, the procedure outputs the size measure total for the sampling frame. If you do not specify the SIZE statement, the procedure outputs the total number of sampling units in the frame. Also, PROC SURVEYSELECT includes the minimum size measure if you specify

the MINSIZE= option, the maximum size measure if you specify the MAXSIZE= option, and the certainty size measure if you specify the CERTSIZE= option.

If you have a stratified design, the output data set includes the stratum-level values of these parameters. Otherwise, the output data set includes the overall population-level values.

For information about the contents of the output data set, see the section "Sample Output Data Set" on page 6657.

OUTSORT=*SAS-data-set*

names an output data set that contains the sorted input data set. This option is available when you specify a CONTROL statement for systematic or sequential selection methods (METHOD=SYS, METHOD=PPS_SYS, METHOD=SEQ, and METHOD=PPS_SEQ). PROC SURVEYSELECT sorts the input data set by the CONTROL variables within strata before selecting the sample.

If you specify CONTROL variables but do not name an output data set with the OUTSORT= option, then the sorted data set replaces the input data set.

REPS=*nreps*

specifies the number of sample replicates. The value of *nreps* must be a positive integer.

When you specify the REPS= option, PROC SURVEYSELECT selects *nreps* independent samples, each with the same specified sample size or sampling rate and the same sample design. The variable Replicate in the OUT= data set contains the sample replicate number.

You can use replicated sampling to provide a simple method of variance estimation for any form of statistic, as well as to evaluate variable nonsampling errors such as interviewer differences. See Lohr (1999), Wolter (1985), Kish (1965, 1987), and Kalton (1983) for information about replicated sampling.

SAMPRATE=*r*

RATE=*r*

specifies the sampling rate, which is the proportion of units to select for the sample. The sampling rate *r* must be a positive number. You can specify *r* as a number between 0 and 1. Or you can specify *r* in percentage form as a number between 1 and 100, and PROC SURVEYSELECT converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

The SAMPRATE= option is available only for equal probability selection methods (METHOD=SRS, METHOD=URS, METHOD=SYS, and METHOD=SEQ). For systematic random sampling (METHOD=SYS), PROC SURVEYSELECT uses the inverse of the sampling rate *r* as the interval. See the section "Systematic Random Sampling" on page 6647 for details. For other selection methods, PROC SURVEYSELECT converts the sampling rate *r* to the sample size before selection by multiplying the total number of units in the stratum or frame by the sampling rate and rounding up to the nearest integer.

If you request a stratified sample design with the STRATA statement and specify the SAMPRATE=*r* option, PROC SURVEYSELECT uses the sampling rate *r* for each stratum. If you do not want to use the same sampling rate for each stratum, use the SAMPRATE=(*values*) option or the SAMPRATE=*SAS-data-set* option to specify a sampling rate for each stratum.

SAMPRATE=(*values*)

RATE=(*values*)

specifies stratum sampling rates. You can separate *values* with blanks or commas. The number of SAMPRATE= values must equal the number of strata in the input data set.

List the stratum sampling rate values in the order in which the strata appear in the input data set. When you use the SAMPRATE=(*values*) option, the input data set must be sorted by the STRATA variables in ascending order. You cannot use the DESCENDING or NOTSORTED option in the STRATA statement.

Each stratum sampling rate value must be a positive number. You can specify a rate value as a number between 0 and 1. Or you can specify a rate value in percentage form as a number between 1 and 100, and PROC SURVEYSELECT converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

The SAMPRATE= option is available only for equal probability selection methods (METHOD=SRS, METHOD=URS, METHOD=SYS, and METHOD=SEQ). For systematic random sampling (METHOD=SYS), PROC SURVEYSELECT uses the inverse of the stratum sampling rate as the interval for the stratum. See the section "Systematic Random Sampling" on page 6647 for details about systematic sampling. For other selection methods, PROC SURVEYSELECT converts the stratum sampling rate to a stratum sample size before selection by multiplying the total number of units in the stratum by the sampling rate and rounding up to the nearest integer.

SAMPRATE=*SAS-data-set*

RATE=*SAS-data-set*

names a SAS data set that contains stratum sampling rates. The SAMPRATE= data set should have a variable _RATE_ that contains the sampling rate for each stratum.

Each sampling rate value must be a positive number. You can specify each value as a number between 0 and 1. Or you can specify a value in percentage form as a number between 1 and 100, and PROC SURVEYSELECT converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

The SAMPRATE= input data set should contain all the STRATA variables, with the same type and length as in the DATA= data set. The STRATA groups should appear in the same order in the SAMPRATE= data set as in the DATA= data set.

The SAMPRATE= option is available only for equal probability selection methods (METHOD=SRS, METHOD=URS, METHOD=SYS, and METHOD=SEQ). For systematic random sampling (METHOD=SYS), PROC SURVEYSELECT uses the inverse of the stratum sampling rate as the interval for the stratum. See the section "Systematic Random Sampling" on page 6647 for details. For other selection methods, PROC SURVEYSELECT converts the stratum sampling rate to the stratum sample size before selection by multiplying the total number of units in the stratum by the sampling rate and rounding up to the nearest integer.

SAMPSIZE=*n*

N=*n*

specifies the sample size, which is the number of units to select for the sample. The sample size *n* must be a positive integer. For selection methods that select without replacement, the sample size *n* must not exceed the number of units in the input data set.

If you specify the ALLOC= option in the STRATA statement, PROC SURVEYSELECT allocates the total sample size among the strata according to the allocation method you request in the ALLOC= option. In this case, SAMPSIZE=*n* specifies the total sample size to be allocated among the strata.

Otherwise, if you specify the SAMPSIZE=*n* option and request a stratified sample design with the STRATA statement, PROC SURVEYSELECT selects *n* units from each stratum. For methods that select without replacement, the sample size *n* must not exceed the number of units in any stratum. If you do not want to select the same number of units from each stratum, use the SAMPSIZE=(*values*) option or the SAMPSIZE=*SAS-data-set* option to specify a sample size for each stratum.

For without-replacement selection methods, by default, PROC SURVEYSELECT does not allow you to specify a stratum sample size that is greater than the total number of units available in the stratum. If you specify the SELECTALL option, PROC SURVEYSELECT selects all stratum units when the stratum sample size exceeds the number of units in the stratum.

SAMPSIZE=(*values***)**

N=(*values***)**

specifies sample sizes for the strata. You can separate *values* with blanks or commas. The number of SAMPSIZE= values must equal the number of strata in the input data set.

List the stratum sample size values in the order in which the strata appear in the input data set. When you use the SAMPSIZE=(*values*) option, the input data set must be sorted by the STRATA variables in ascending order. You cannot use the DESCENDING or NOTSORTED option in the STRATA statement.

Each stratum sample size value must be a positive integer. For without-replacement selection methods, by default, PROC SURVEYSELECT does not allow you to specify a stratum sample size that is greater than the total number of units available in the stratum. If you specify the SELECTALL option, PROC SURVEYSELECT selects all stratum units when the stratum sample size exceeds the number of units in the stratum.

SAMPSIZE=*SAS-data-set*

N=*SAS-data-set*

names a SAS data set that contains the sample sizes for the strata.

You provide the stratum sample sizes in the SAMPSIZE= input data set variable named _NSIZE_ or SampleSize. Each stratum sample size value must be a positive integer.

The SAMPSIZE= input data set should contain all the STRATA variables, with the same type and length as in the DATA= data set. The STRATA groups should appear in the same order in the SAMPSIZE= data set as in the DATA= data set. The SAMPSIZE= data set is a secondary

input data set. See the section "Secondary Input Data Set" on page 6656 for details. You can name only one secondary input data set in each invocation of the procedure.

For without-replacement selection methods, by default, PROC SURVEYSELECT does not allow you to specify a stratum sample size that is greater than the total number of units available in the stratum. If you specify the SELECTALL option, PROC SURVEYSELECT selects all stratum units when the stratum sample size exceeds the number of units in the stratum.

SEED=*number*

specifies the initial seed for random number generation. The SEED= value must be a positive integer. If you do not specify the SEED= option, or if the SEED= value is negative or zero, PROC SURVEYSELECT uses the time of day from the computer's clock to obtain the initial seed. See the section "Sample Selection Methods" on page 6645 for more information.

Whether or not you specify the SEED= option, PROC SURVEYSELECT displays the value of the initial seed in the "Sample Selection Summary" table. If you need to reproduce the same sample in a subsequent execution of PROC SURVEYSELECT, you can specify this same seed value in the SEED= option, along with the same sample selection parameters, and PROC SURVEYSELECT will reproduce the sample.

If you request a stratified sample design with the STRATA statement, you can use the SEED=*SAS-data-set* option to specify an initial seed for each stratum. Otherwise, PROC SURVEYSELECT generates random numbers continuously across strata from the random number stream initialized by the SEED= value, as described in the section "Sample Selection Methods" on page 6645.

You can use the OUTSEED option to include the stratum initial seeds in the output data set.

SEED=*SAS-data-set*

names a SAS data set that contains initial seeds for the strata. You provide the stratum seeds in the SEED= input data set variable _SEED_ or InitialSeed.

The initial seed values must be positive integers. If the initial seed value for the first stratum is not a positive integer, PROC SURVEYSELECT uses the time of day from the computer's clock to obtain the initial seed. If the initial seed value for a subsequent stratum is not a positive integer, PROC SURVEYSELECT continues to use the random number stream already initialized by the seed for the previous stratum. See the section "Sample Selection Methods" on page 6645 for more information.

The SEED= input data set should contain all the STRATA variables, with the same type and length as in the DATA= data set. The STRATA groups should appear in the same order in the SEED= data set as in the DATA= data set. The SEED= data set is a secondary input data set. See the section "Secondary Input Data Set" on page 6656 for details. You can name only one secondary input data set in each invocation of the procedure.

You can use the OUTSEED option to include the stratum initial seeds in the output data set.

Whether or not you specify the SEED= option, PROC SURVEYSELECT displays the value of the initial seed in the "Sample Selection Summary" table. If you need to reproduce the same sample in a subsequent execution of PROC SURVEYSELECT, you can specify this

same seed value in the SEED= option, along with the same sample selection parameters, and PROC SURVEYSELECT will reproduce the sample.

If you specify initial seeds by strata with the SEED=*SAS-data-set* option, you can reproduce the same sample in a subsequent execution of PROC SURVEYSELECT by specifying these same stratum initial seeds, along with the same sample selection parameters. If you need to reproduce the same sample for only a subset of the strata, you can use the same initial seeds for those strata in the subset.

SELECTALL

requests that PROC SURVEYSELECT select all stratum units when the stratum sample size exceeds the total number of units in the stratum. By default, PROC SURVEYSELECT does not allow you to specify a stratum sample size that is greater than the total number of units in the stratum, unless you are using a with-replacement selection method.

The SELECTALL option is available for the following without-replacement selection methods: METHOD=SRS, METHOD=SYS, METHOD=SEQ, METHOD=PPS, and METHOD=PPS_SAMPFORD.

The SELECTALL option is not available for with-replacement selection methods, with-minimum-replacement methods, or those PPS methods that select two units per stratum.

SORT=NEST | SERP

specifies the type of sorting by CONTROL variables. The option SORT=NEST requests nested sorting, and SORT=SERP requests hierarchic serpentine sorting. The default is SORT=SERP. See the section "Sorting by CONTROL Variables" on page 6644 for descriptions of serpentine and nested sorting. Where there is only one CONTROL variable, the two types of sorting are equivalent.

This option is available when you specify a CONTROL statement for systematic or sequential selection methods (METHOD=SYS, METHOD=PPS_SYS, METHOD=SEQ, and METHOD=PPS_SEQ). When you specify a CONTROL statement, PROC SURVEYSELECT sorts the input data set by the CONTROL variables within strata before selecting the sample.

When you specify a CONTROL statement, you can also use the OUTSORT= option to name an output data set that contains the sorted input data set. Otherwise, if you do not specify the OUTSORT= option, then the sorted data set replaces the input data set.

STATS

includes selection probabilities and sampling weights in the OUT= output data set for equal probability selection methods when you do not specify a STRATA statement. This option is available for the following equal probability selection methods: METHOD=SRS, METHOD=URS, METHOD=SYS, and METHOD=SEQ. For PPS selection methods and stratified designs, the output data set contains selection probabilities and sampling weights by default. For more information about the contents of the output data set, see the section "Sample Output Data Set" on page 6657.

CONTROL Statement

CONTROL *variables* **;**

The CONTROL statement names variables for sorting the input data set. The CONTROL variables can be character or numeric.

PROC SURVEYSELECT sorts the input data set by the CONTROL variables before selecting the sample. If you also specify a STRATA statement, PROC SURVEYSELECT sorts by CONTROL variables within strata. Control sorting is available for systematic and sequential selection methods (METHOD=SYS, METHOD=PPS_SYS, METHOD=SEQ, and METHOD=PPS_SEQ).

By default (or if you specify the SORT=SERP option), PROC SURVEYSELECT uses hierarchic serpentine sorting by the CONTROL variables. If you specify the SORT=NEST option, the procedure uses nested sorting. For more information about serpentine and nested sorting, see the section "Sorting by CONTROL Variables" on page 6644.

You can use the OUTSORT= option to name an output data set that contains the sorted input data set. If you do not specify the OUTSORT= option when you use the CONTROL statement, then the sorted data set replaces the input data set.

ID Statement

ID *variables* **;**

The ID statement names variables from the DATA= input data set to be included in the OUT= data set of selected units. If there is no ID statement, PROC SURVEYSELECT includes all variables from the DATA= data set in the OUT= data set. The ID variables can be character or numeric.

SIZE Statement

SIZE *variable* **;**

The SIZE statement names one and only one size measure variable, which contains the sampling unit size measures that are used for selection with probability proportional to size. The SIZE variable must be numeric. When the value of an observation's SIZE variable is missing or nonpositive, that observation is excluded from the sample selection.

The SIZE statement is required for all PPS selection methods, which include METHOD=PPS, METHOD=PPS_BREWER, METHOD=PPS_MURTHY, METHOD=PPS_SAMPFORD, METHOD=PPS_SEQ, METHOD=PPS_SYS, and METHOD=PPS_WR. For details about how size measures are used, see the descriptions of PPS methods in the section "Sample Selection Methods" on page 6645.

Note that an observation's size measure, specified in the SIZE statement and used for PPS selection, is not the same as the sample size. The sample size is the number of units to select for the sample, and you can specify the sample size with the SAMPSIZE= option.

STRATA Statement

STRATA *variables* < / *options* > **;**

You can specify a STRATA statement to partition the input data set into nonoverlapping groups defined by the STRATA variables. PROC SURVEYSELECT then selects independent samples from these strata, according to the selection method and design parameters specified in the PROC SURVEYSELECT statement. For information about the use of stratification in sample design, see Lohr (1999), Kalton (1983), Kish (1965, 1987), and Cochran (1977).

The *variables* are one or more variables in the input data set. The STRATA variables function much like BY variables, and PROC SURVEYSELECT expects the input data set to be sorted in order of the STRATA variables.

If you specify a CONTROL statement, or if you specify METHOD=PPS, the input data set must be sorted in ascending order of the STRATA variables. This means you cannot use the STRATA option NOTSORTED or DESCENDING when you specify a CONTROL statement or METHOD=PPS.

If your input data set is not sorted by the STRATA variables in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with the STRATA variables in a BY statement.
- Specify the option NOTSORTED or DESCENDING in the STRATA statement for the SURVEYSELECT procedure (when you do not specify a CONTROL statement or METHOD=PPS). The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the STRATA variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the STRATA variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

Allocation Options

The STRATA options request allocation of the total sample size among the strata. You can specify the total sample size in the SAMPSIZE= option in the PROC SURVEYSELECT statement. When you request allocation with the ALLOC= option in the STRATA statement, PROC SURVEYSELECT allocates the total sample size among the strata according to the allocation method you name. You can request proportional allocation (ALLOC=PROP), optimal allocation

(ALLOC=OPTIMAL), or Neyman allocation (ALLOC=NEYMAN). See the section "Sample Size Allocation" on page 6654 for details about these methods.

Instead of requesting that PROC SURVEYSELECT compute the sample allocation, you can directly specify the allocation proportions by using the ALLOC=(*values*) option or the ALLOC=*SAS-data-set* option. Then PROC SURVEYSELECT allocates the total sample size among the strata according to the proportions you specify.

By default, PROC SURVEYSELECT computes the allocation of the total sample size among the strata and then selects the sample by using the allocated sample sizes. If you specify the NOSAMPLE option, PROC SURVEYSELECT computes the allocation but does not select the sample. In this case the OUT= output data set contains the stratum sample sizes computed according to the specified allocation method. See the section "Allocation Output Data Set" on page 6660 for details.

You can specify the following options in the STRATA statement.

ALLOC=*name*

specifies the method for allocating the total sample size among the strata. The following values for *name* are available:

PROPORTIONAL

PROP

requests proportional allocation, which allocates the total sample size in proportion to the stratum sizes, where the stratum size is the number of sampling units in the stratum. See the section "Sample Size Allocation" on page 6654 for details.

OPTIMAL

OPT

requests optimal allocation, which allocates the total sample size among the strata in proportion to stratum sizes, stratum variances, and stratum costs. See the section "Sample Size Allocation" on page 6654 for more information. If you specify ALLOC=OPTIMAL, you must provide the stratum variances with the VAR, VAR=(*values*), or VAR=*SAS-data-set* option. You must provide the stratum costs with the COST, COST=(*values*), or COST=*SAS-data-set* option.

NEYMAN

requests Neyman allocation, which allocates the total sample size among the strata in proportion to the stratum sizes and variances. See the section "Sample Size Allocation" on page 6654 for more information. If you specify ALLOC=NEYMAN, you must provide the stratum variances with the VAR, VAR=(*values*), or VAR=*SAS-data-set* option.

ALLOC=(*values***)**

lists stratum allocation proportions. You can separate *values* with blanks or commas.

Each allocation proportion specifies the percent of the total sample size to allocate to the corresponding stratum. The number of ALLOC= values must equal the number of strata in the input data set. The sum of the allocation proportions must equal 1.

Each allocation proportion must be a positive number. You can specify each value as a number between 0 and 1. Or you can specify a value in percentage form as a number between 1 and

100, and PROC SURVEYSELECT converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

List the allocation proportions in the order in which the strata appear in the input data set. If you use the ALLOC=(*values*) option, the input data set must be sorted by the STRATA variables in ascending order. You cannot use the DESCENDING or NOTSORTED option in the STRATA statement.

ALLOC=*SAS-data-set*

names a SAS data set that contains stratum allocation proportions. You provide the stratum allocation proportions in the ALLOC= data set variable _ALLOC_.

Each allocation proportion specifies the percent of the total sample size to allocate to the corresponding stratum. The sum of the allocation proportions must equal 1.

Each allocation proportion must be a positive number. You can specify the value as a number between 0 and 1. Or you can specify the value in percentage form as a number between 1 and 100, and PROC SURVEYSELECT converts that number to a proportion. The procedure treats the value 1 as 100%, and not the percentage form 1%.

The ALLOC= data set should contain all the STRATA variables, with the same type and length as in the DATA= input data set. The STRATA groups should appear in the same order in the ALLOC= data set as in the DATA= data set. The ALLOC= data set is a secondary input data set. See the section "Secondary Input Data Set" on page 6656 for details. You can name only one secondary data set in each invocation of the procedure.

COST

indicates that stratum costs are included in the secondary input data set. Use the COST option when you have already named the secondary input data set in another option, such as the VAR=*SAS-data-set* option. You provide the stratum costs in the secondary input data set variable _COST_.

A stratum cost represents the per-unit cost, or the survey cost of a single unit in the stratum. Each stratum cost must be a positive number. Cost values are required if you specify the ALLOC=OPTIMAL option.

COST=(*values***)**

specifies stratum costs, which are required if you specify the ALLOC=OPTIMAL option. You can separate *values* with blanks or commas.

A stratum cost represents the per-unit cost, or the survey cost of a single unit in the stratum. Each stratum cost must be a positive number.

The number of COST= values must equal the number of strata in the input data set. List the stratum costs in the order in which the strata appear in the input data set. If you use the COST=*values* option, the input data set must be sorted by the STRATA variables in ascending order. You cannot use the DESCENDING or NOTSORTED option in the STRATA statement.

COST=*SAS-data-set*

names a SAS data set that contains the stratum costs. You provide the stratum costs in the COST= data set variable _COST_.

A stratum cost represents the per-unit cost, or the survey cost of a single unit in the stratum. Each stratum cost must be a positive number. Stratum costs are required if you specify the ALLOC=OPTIMAL option.

The COST= data set should contain all the STRATA variables, with the same type and length as in the DATA= input data set. The STRATA groups should appear in the same order in the COST= data set as in the DATA= data set. The COST= data set is a secondary input data set. See the section "Secondary Input Data Set" on page 6656 for details. You can name only one secondary input data set in each invocation of the procedure.

NOSAMPLE

requests that SURVEYSELECT allocate the total sample size among the strata but not select the sample. When you specify the NOSAMPLE option, the OUT= output data set contains the stratum sample sizes that PROC SURVEYSELECT computes. See the section "Allocation Output Data Set" on page 6660 for details.

VAR

indicates that stratum variances are included in the secondary input data set. Use the VAR option when you have already named the secondary input data set in another option, such as the COST=*SAS-data-set* option. You provide the stratum variances in the secondary input data set variable _VAR_.

Each stratum variance must be a positive number. Stratum variances are required if you specify ALLOC=OPTIMAL or ALLOC=NEYMAN.

VAR=(*values*)

lists stratum variances, which are required if you specify ALLOC=OPTIMAL or ALLOC=NEYMAN. You can separate *values* with blanks or commas.

Each stratum variance must be a positive number. The number of VAR= values must equal the number of strata in the input data set. List the stratum variances in the order in which the strata appear in the input data set. If you use the VAR=(*values*) option, the input data set must be sorted by the STRATA variables in ascending order. You cannot use the DESCENDING or NOTSORTED option in the STRATA statement.

VAR=*SAS-data-set*

names a SAS data set that contains the stratum variances. You provide the stratum variances in the VAR= data set variable _VAR_.

Each stratum variance must be a positive number. Stratum variances are required if you specify ALLOC=OPTIMAL or ALLOC=NEYMAN.

The VAR= data set should contain all the STRATA variables, with the same type and length as in the DATA= input data set. The STRATA groups should appear in the same order in the VAR= data set as in the DATA= data set. The VAR= data set is a secondary input data set. See the section "Secondary Input Data Set" on page 6656 for details. You can name only one secondary input data set in each invocation of the procedure.

Details: SURVEYSELECT Procedure

Missing Values

If an observation has a missing or nonpositive value for the SIZE variable, PROC SURVEYSELECT excludes that observation from the sample selection. The procedure provides a log note that gives the number of observations omitted due to missing or nonpositive size measures.

PROC SURVEYSELECT treats missing STRATA variable values like any other STRATA variable value. The missing values form a separate stratum.

If a value of _NSIZE_ (or SampleSize) is missing in the SAMPSIZE= secondary input data set, then PROC SURVEYSELECT cannot select a sample from that stratum. Similarly, if other secondary data set variables have missing values for a stratum, a sample cannot be selected from that stratum. These variables include _NRATE_, _MINSIZE_, _MAXSIZE_, _CERTSIZE_, and _CERTP_. See the section "Secondary Input Data Set" on page 6656 for details.

If a value of _ALLOC_, _VAR_, or _COST_ is missing in the secondary input data set, PROC SURVEYSELECT cannot compute the sample allocation.

Sorting by CONTROL Variables

If you specify a CONTROL statement, PROC SURVEYSELECT sorts the input data set by the CONTROL variables before selecting the sample. If you also specify a STRATA statement, the procedure sorts by CONTROL variables within strata. Sorting by CONTROL variables is available for systematic and sequential selection methods, which include METHOD=SYS, METHOD=PPS_SYS, METHOD=SEQ, and METHOD=PPS_SEQ. Sorting provides additional control over the distribution of the sample, giving some benefits of proportionate stratification.

By default, the sorted data set replaces the input data set. Or you can use the OUTSORT= option to name an output data set that contains the sorted input data set.

PROC SURVEYSELECT provides two types of sorting: hierarchic serpentine sorting and nested sorting. By default (or if you specify the SORT=SERP option), the procedure uses serpentine sorting. If you specify the SORT=NEST option, then the procedure sorts by the CONTROL variables according to nested sorting. These two types of sorting are equivalent when there is only one CONTROL variable.

If you request nested sorting, PROC SURVEYSELECT sorts observations in the same order as PROC SORT does for an ascending sort by the CONTROL variables. See the chapter "The SORT Procedure" in the *Base SAS Procedures Guide* for more information. PROC SURVEYSELECT sorts within strata if you also specify a STRATA statement. The procedure first arranges the input observations in ascending order of the first CONTROL variable. Then within each level of the

first control variable, the procedure arranges the observations in ascending order of the second CONTROL variable. This continues for all CONTROL variables specified.

In hierarchic serpentine sorting, PROC SURVEYSELECT sorts by the first CONTROL variable in ascending order. Then within the first level of the first CONTROL variable, the procedure sorts by the second CONTROL variable in ascending order. Within the second level of the first CONTROL variable, the procedure sorts by the second CONTROL variable in descending order. Sorting by the second CONTROL variable continues to alternate between ascending and descending sorting throughout all levels of the first CONTROL variable. If there is a third CONTROL variable, the procedure sorts by that variable within levels formed from the first two CONTROL variables, again alternating between ascending and descending sorting. This continues for all CONTROL variables specified. This sorting algorithm minimizes the change from one observation to the next with respect to the CONTROL variable values, thus making nearby observations more similar. For more information about serpentine sorting, see Chromy (1979) and Williams and Chromy (1980).

Sample Selection Methods

PROC SURVEYSELECT provides a variety of methods for selecting probability-based random samples. With probability sampling, each unit in the survey population has a known, positive probability of selection. This property of probability sampling avoids selection bias and enables you to use statistical theory to make valid inferences from the sample to the survey population. See Lohr (1999), Kish (1965, 1987), Kalton (1983), and Cochran (1977) for more information about probability sampling.

In equal probability sampling, each unit in the sampling frame, or in a stratum, has the same probability of being selected for the sample. PROC SURVEYSELECT provides the following methods that select units with equal probability: simple random sampling, unrestricted random sampling, systematic random sampling, and sequential random sampling. In simple random sampling, units are selected *without replacement*, which means that a unit cannot be selected more than once. Both systematic and sequential equal probability sampling are also without replacement. In unrestricted random sampling, units are selected *with replacement*, which means that a unit can be selected more than once. In with-replacement sampling, the *number of hits* refers to the number of times a unit is selected.

In probability proportional to size (PPS) sampling, a unit's selection probability is proportional to its size measure. PROC SURVEYSELECT provides the following methods that select units with probability proportional to size (PPS): PPS sampling without replacement, PPS sampling with replacement, PPS systematic sampling, PPS sequential sampling, Brewer's method, Murthy's method, and Sampford's method. PPS sampling is often used in cluster sampling, where you select clusters (or groups of sampling units) of varying size in the first stage of selection. For example, clusters might be schools, hospitals, or geographical areas, and the final sampling units might be students, patients, or citizens. Cluster sampling can provide efficiencies in frame construction and other survey operations. See Lohr (1999), Kalton (1983), Kish (1965), and the other references cited in the following sections for more information.

All the probability sampling methods provided by PROC SURVEYSELECT use random numbers in their selection algorithms, as described in the following sections and in the references cited. PROC

SURVEYSELECT uses a uniform random number function to generate streams of pseudo-random numbers from an initial starting point, or *seed*. You can use the SEED= option to specify the initial seed. If you do not specify the SEED= option, PROC SURVEYSELECT uses the time of day from the computer's clock to obtain the initial seed. PROC SURVEYSELECT generates uniform random numbers according to the method of Fishman and Moore (1982), which uses a prime modulus multiplicative generator with modulus 2^{31} and multiplier 397204094. PROC SURVEYSELECT uses the same uniform random number generator as the RANUNI function. For more information about the RANUNI function, see *SAS Language Reference: Dictionary*.

The following sections give detailed descriptions of the sample selection methods available in PROC SURVEYSELECT. In these sections, n_h denotes the sample size (the number of units in the sample) for stratum h, and N_h denotes the population size (number of units in the population) for stratum h, for $h = 1, 2, \ldots, H$. When the sample design is not stratified, n denotes the sample size, and N denotes the population size. For PPS sampling, M_{hi} represents the size measure for unit i in stratum h, $M_{h\cdot}$ is the total of all size measures for the population of stratum h, and $Z_{hi} = M_{hi}/M_{h\cdot}$ is the relative size of unit i in stratum h.

Simple Random Sampling

The method of simple random sampling (METHOD=SRS) selects units with equal probability and without replacement. Each possible sample of n different units out of N has the same probability of being selected. The selection probability for each individual unit equals n/N. When you request stratified sampling with a STRATA statement, PROC SURVEYSELECT selects samples independently within strata. The selection probability for a unit in stratum h equals n_h/N_h for stratified simple random sampling.

By default, PROC SURVEYSELECT uses Floyd's ordered hash table algorithm for simple random sampling. This algorithm is fast, efficient, and appropriate for large data sets. See Bentley and Floyd (1987) and Bentley and Knuth (1986) for details.

If there is not enough memory available for Floyd's algorithm, PROC SURVEYSELECT switches to the sequential algorithm of Fan, Muller, and Rezucha (1962), which requires less memory but might require more time to select the sample. When PROC SURVEYSELECT uses the alternative sequential algorithm, it writes a note to the log. To request the sequential algorithm, even if enough memory is available for Floyd's algorithm, you can specify METHOD=SRS2 in the PROC SURVEYSELECT statement.

Unrestricted Random Sampling

The method of unrestricted random sampling (METHOD=URS) selects units with equal probability and with replacement. Because units are selected with replacement, a unit can be selected for the sample more than once. The expected number of selections or hits for each unit equals n/N when sampling without stratification. For stratified sampling, the expected number of hits for a unit in stratum h equals n_h/N_h. Note that the expected number of hits exceeds one when the sample size n is greater than the population size N.

For unrestricted random sampling, by default, the output data set contains one observation for each distinct unit selected for the sample, together with a variable NumberHits that gives the number of times the observation was selected. But if you specify the OUTHITS option, then the output data set contains a separate observation for each selection, so that a unit selected three times, for example, is represented by three observations in the output data set. For information about the contents of the output data set, see the section "Sample Output Data Set" on page 6657.

Systematic Random Sampling

The method of systematic random sampling (METHOD=SYS) selects units at a fixed interval throughout the sampling frame or stratum after a random start. If you specify the sample size (or the stratum sample sizes) with the SAMPSIZE= option, PROC SURVEYSELECT uses a fractional interval to provide exactly the specified sample size. The interval equals N/n, or N_h/n_h for stratified sampling. The selection probability for each unit equals n/N, or n_h/N_h for stratified sampling. If you specify the sampling rate (or the stratum sampling rates) with the SAMPRATE= option, PROC SURVEYSELECT uses the inverse of the rate as the interval for systematic selection. The selection probability for each unit equals the specified rate.

Systematic random sampling controls the distribution of the sample by spreading it throughout the sampling frame or stratum at equal intervals, thus providing implicit stratification. You can use the CONTROL statement to order the input data set by the CONTROL variables before sample selection. If you also use a STRATA statement, PROC SURVEYSELECT sorts by the CONTROL variables within strata. If you do not specify a CONTROL statement, PROC SURVEYSELECT applies systematic selection to the observations in the order in which they appear in the input data set.

Sequential Random Sampling

If you specify the option METHOD=SEQ and do not include a SIZE statement, PROC SURVEYSELECT uses the equal probability version of Chromy's method for sequential random sampling. This method selects units sequentially with equal probability and without replacement. See Chromy (1979) and Williams and Chromy (1980) for details. See the section "PPS Sequential Sampling" on page 6651 for a description of Chromy's PPS selection method.

Sequential random sampling controls the distribution of the sample by spreading it throughout the sampling frame or stratum, thus providing implicit stratification according to the order of units in the frame or stratum. You can use the CONTROL statement to sort the input data set by the CONTROL variables before sample selection. If you also use a STRATA statement, PROC SURVEYSELECT sorts by the CONTROL variables within strata. By default (or if you specify the SORT=SERP option), the procedure uses hierarchic serpentine ordering for sorting. If you specify the SORT=NEST option, the procedure uses nested sorting. See the section "Sorting by CONTROL Variables" on page 6644 for descriptions of serpentine and nested sorting. If you do not specify a CONTROL statement, PROC SURVEYSELECT applies sequential selection to the observations in the order in which they appear in the input data set.

Following Chromy's method of sequential selection, PROC SURVEYSELECT randomly chooses a starting unit from the entire stratum (or frame, if the design is not stratified). With this unit as the

first one, the procedure treats the stratum units as a closed loop. This is done so that all pairwise (joint) selection probabilities are positive and an unbiased variance estimator can be obtained. The procedure numbers units sequentially from the random start to the end of the stratum and then continues from the beginning of the stratum until all units are numbered.

Beginning with the randomly chosen starting unit, PROC SURVEYSELECT accumulates the expected number of selections or hits, where the expected number of selections $\mathrm{E}[S_{hi}]$ equals n_h/N_h for all units i in stratum h. The procedure computes

$$I_{hi} = \mathrm{Int}\Big(\sum_{j=1}^{i} \mathrm{E}[S_{hj}]\Big) = \mathrm{Int}(i\, n_h/N_h)$$

$$F_{hi} = \mathrm{Frac}\Big(\sum_{j=1}^{i} \mathrm{E}[S_{hj}]\Big) = \mathrm{Frac}(i\, n_h/N_h)$$

where Int denotes the integer part of the number, and Frac denotes the fractional part.

Considering each unit sequentially, Chromy's method determines whether unit i is selected by comparing the total number of selections for the first $i-1$ units,

$$T_{h(i-1)} = \sum_{j=1}^{i-1} S_{hj}$$

with the value of $I_{h(i-1)}$.

If $T_{h(i-1)} = I_{h(i-1)}$, Chromy's method determines whether or not unit i is selected as follows. If $F_{hi} = 0$ or $F_{h(i-1)} > F_{hi}$, then unit i is selected with certainty. Otherwise, unit i is selected with probability

$$(F_{hi} - F_{h(i-1)})/(1 - F_{h(i-1)})$$

If $T_{h(i-1)} = I_{h(i-1)} + 1$, Chromy's method determines whether or not unit i is selected as follows. If $F_{hi} = 0$ or $F_{hi} > F_{h(i-1)}$, then the unit is not selected. Otherwise, unit i is selected with probability

$$F_{hi}/F_{h(i-1)}$$

PPS Sampling without Replacement

If you specify the option METHOD=PPS, PROC SURVEYSELECT selects units with probability proportional to size and without replacement. The selection probability for unit i in stratum h equals $n_h Z_{hi}$, where n_h is the sample size for stratum h, and Z_{hi} is the relative size of unit i in stratum h. The relative size equals $M_{hi}/M_{h.}$, which is the ratio of the size measure for unit i in stratum h (M_{hi}) to the total of all size measures for stratum h ($M_{h.}$).

Because selection probabilities cannot exceed 1, the relative size for each unit must not exceed $1/n_h$ for METHOD=PPS. This requirement can be expressed as $Z_{hi} <= 1/n_h$, or equivalently, $M_{hi} <= M_{h.}/n_h$. If your size measures do not meet this requirement, you can adjust the size

measures by using the MAXSIZE= or MINSIZE= option. Or you can request certainty selection for the larger units by using the CERTSIZE= or CERTSIZE=P= option. Alternatively, you can use a selection method that does not have this relative size restriction, such as PPS with minimum replacement (METHOD=PPS_SEQ).

PROC SURVEYSELECT uses the Hanurav-Vijayan algorithm for PPS selection without replacement. Hanurav (1967) introduced this algorithm for the selection of two units per stratum, and Vijayan (1968) generalized it for the selection of more than two units. The algorithm enables computation of joint selection probabilities and provides joint selection probability values that usually ensure nonnegativity and stability of the Sen-Yates-Grundy variance estimator. See Fox (1989), Golmant (1990), and Watts (1991) for details.

Notation in the remainder of this section drops the stratum subscript h for simplicity, but selection is still done independently within strata if you specify a stratified design. For a stratified design, n now denotes the sample size for the current stratum, N denotes the stratum population size, and M_i denotes the size measure for unit i in the stratum. If the design is not stratified, this notation applies to the entire sampling frame.

According to the Hanurav-Vijayan algorithm, PROC SURVEYSELECT first orders units within the stratum in ascending order by size measure, so that $M_1 \leq M_2 \leq \ldots \leq M_N$. Then the procedure selects the PPS sample of n observations as follows:

1. The procedure randomly chooses one of the integers $1, 2, \ldots, n$ with probability $\theta_1, \theta_2, \ldots, \theta_n$, where

   $$\theta_i = n(Z_{N-n+i+1} - Z_{N-n+i})(T + iZ_{N-n+1})/T$$

   where $Z_j = M_j/M$ and

   $$T = \sum_{j=1}^{N-n} Z_j$$

   By definition, $Z_{N+1} = 1/n$ to ensure that $\sum_{i=1}^{n} \theta_i = 1$.

2. If i is the integer selected in step 1, the procedure includes the last $(n-i)$ units of the stratum in the sample, where the units are ordered by size measure as described previously. The procedure then selects the remaining i units according to steps 3 through 6.

3. The procedure defines new normed size measures for the remaining $(N-n+i)$ stratum units that were not selected in steps 1 and 2:

   $$Z_j^* = \begin{cases} Z_j/(T + iZ_{N-n+1}) & \text{for } j = 1, \ldots, N-n+1 \\ Z_{N-n+1}/(T + iZ_{N-n+1}) & \text{for } j = N-n+2, \ldots, N-n+i \end{cases}$$

4. The procedure selects the next unit from the first $(N-n+1)$ stratum units with probability proportional to $a_j(1)$, where

   $$\begin{aligned} a_1(1) &= iZ_1^* \\ a_j(1) &= iZ_j^* \textstyle\prod_{k=1}^{j-1}\big(1-(i-1)\,P_k\big) \quad \text{for } j = 2, \ldots, N-n+1 \end{aligned}$$

   and

   $$P_k = M_k/(M_{k+1} + M_{k+2} + \cdots + M_{N-n+i})$$

5. If stratum unit j_1 is the unit selected in step 4, then the procedure selects the next unit from units $(j_1 + 1)$ through $(N - n + 2)$ with probability proportional to $a_j(2, j_1)$, where

$$a_{j_1+1}(2, j_1) = (i-1)Z^*_{j_1+1}$$

$$a_j(2, j_1) = (i-1)Z^*_j \prod_{k=j_1+1}^{j-1} \left(1 - (i-2)P_k\right) \quad \text{for} \quad j = j_1 + 2, \ldots, N - n + 2$$

6. The procedure repeats step 5 until all n sample units are selected.

If you specify the JTPROBS option, PROC SURVEYSELECT computes the joint selection probabilities for all pairs of selected units in each stratum. The joint selection probability for units i and j in the stratum equals

$$P_{(ij)} = \sum_{r=1}^{n} \theta_r K_{ij}^{(r)}$$

where

$$K_{ij} = \begin{cases} 1 & N - n + r < i \le N - 1 \\ rZ_{N-n+1}/(T + rZ_{N-n+1}) & N - n < i \le N - n + r, \quad j > N - n + r \\ rZ_i/(T + rZ_{N-n+1}) & 1 \le i \le N - n, \quad j > N - n + r \\ \pi_{ij}^{(r)} & j \le N - n + r \end{cases}$$

$$\pi_{ij}^{(r)} = \frac{r(r-1)}{2} P_i Z_j \prod_{k=1}^{i-1} (1 - P_k)$$

$$P_k = M_k/(M_{k+1} + M_{k+2} + \cdots + M_{N-n+r})$$

PPS Sampling with Replacement

If you specify the option METHOD=PPS_WR, PROC SURVEYSELECT selects units with probability proportional to size and with replacement. The procedure makes n_h independent random selections from the stratum of N_h units, selecting with probability $Z_{hi} = M_{hi}/M_{h.}$. Because units are selected with replacement, a unit can be selected for the sample more than once. The expected number of selections or hits for unit i in stratum h equals $n_h Z_{hi}$. If you specify the JTPROBS option, PROC SURVEYSELECT computes the joint expected number of hits for all pairs of selected units in each stratum. The joint expected number of hits for units i and j in stratum h equals

$$P_{h(ij)} = \begin{cases} n_h(n_h - 1)Z_{hi}Z_{hj} & \text{for} \quad j \ne i \\ n_h(n_h - 1)Z_{hi}Z_{hi}/2 & \text{for} \quad j = i \end{cases}$$

PPS Systematic Sampling

If you specify the option METHOD=PPS_SYS, PROC SURVEYSELECT selects units by systematic random sampling with probability proportional to size. Systematic sampling selects

units at a fixed interval throughout the stratum or sampling frame after a random start. PROC SURVEYSELECT uses a fractional interval to provide exactly the specified sample size. The interval equals $M_{h\cdot}/n_h$ for stratified sampling and M/n for sampling without stratification. Depending on the sample size and the values of the size measures, it might be possible for a unit to be selected more than once. The expected number of selections or hits for unit i in stratum h equals $n_h M_{hi}/M_{h\cdot} = n_h Z_{hi}$. See Cochran (1977, pp. 265–266) and Madow (1949) for details.

Systematic random sampling controls the distribution of the sample by spreading it throughout the sampling frame or stratum at equal intervals, thus providing implicit stratification. You can use the CONTROL statement to order the input data set by the CONTROL variables before sample selection. If you also use a STRATA statement, PROC SURVEYSELECT sorts by the CONTROL variables within strata. If you do not specify a CONTROL statement, PROC SURVEYSELECT applies systematic selection to the observations in the order in which they appear in the input data set.

PPS Sequential Sampling

If you specify the option METHOD=PPS_SEQ, PROC SURVEYSELECT uses Chromy's method of sequential random sampling. See Chromy (1979) and Williams and Chromy (1980) for details. Chromy's method selects units sequentially with probability proportional to size and with minimum replacement. Selection *with minimum replacement* means that the actual number of hits for a unit can equal the integer part of the expected number of hits for that unit, or the next largest integer. This can be compared to selection *without replacement*, where each unit can be selected only once, so the number of hits can equal 0 or 1. The other alternative is selection *with replacement*, where there is no restriction on the number of hits for each unit, so the number of hits can equal $0, 1, \cdots, n_h$, where n_h is the stratum sample size.

Sequential random sampling controls the distribution of the sample by spreading it throughout the sampling frame or stratum, thus providing implicit stratification according to the order of units in the frame or stratum. You can use the CONTROL statement to sort the input data set by the CONTROL variables before sample selection. If you also use a STRATA statement, PROC SURVEYSELECT sorts by the CONTROL variables within strata. By default (or if you specify the SORT=SERP option), the procedure uses hierarchic serpentine ordering to sort the sampling frame by the CONTROL variables within strata. If you specify the SORT=NEST option, the procedure uses nested sorting. See the section "Sorting by CONTROL Variables" on page 6644 for descriptions of serpentine and nested sorting. If you do not specify a CONTROL statement, PROC SURVEYSELECT applies sequential selection to the observations in the order in which they appear in the input data set.

According to Chromy's method of sequential selection, PROC SURVEYSELECT first chooses a starting unit randomly from the entire stratum, with probability proportional to size. The procedure uses this unit as the first one and treats the stratum observations as a closed loop. This is done so that all pairwise (joint) expected number of hits are positive and an unbiased variance estimator can be obtained. The procedure numbers observations sequentially from the random start to the end of the stratum and then continues from the beginning of the stratum until all units are numbered.

Beginning with the randomly chosen starting unit, Chromy's method partitions the ordered stratum sampling frame into n_h zones of equal size. There is one selection from each zone and a total of

n_h selections or hits, although fewer than n_h distinct units might be selected. Beginning with the random start, the procedure accumulates the expected number of hits and computes

$$\mathrm{E}[S_{hi}] = n_h Z_{hi}$$

$$I_{hi} = \mathrm{Int}\Big(\sum_{j=1}^{i} \mathrm{E}[S_{hj}]\Big)$$

$$F_{hi} = \mathrm{Frac}\Big(\sum_{j=1}^{i} \mathrm{E}[S_{hj}]\Big)$$

where $\mathrm{E}[S_{hi}]$ represents the expected number of hits for unit i in stratum h, Int denotes the integer part of the number, and Frac denotes the fractional part.

Considering each unit sequentially, Chromy's method determines the actual number of hits for unit i by comparing the total number of hits for the first $i-1$ units,

$$T_{h(i-1)} = \sum_{j=1}^{i-1} S_{hj}$$

with the value of $I_{h(i-1)}$.

If $T_{h(i-1)} = I_{h(i-1)}$, Chromy's method determines the total number of hits for the first i units as follows. If $F_{hi} = 0$ or $F_{h(i-1)} > F_{hi}$, then $T_{hi} = I_{hi}$. Otherwise, $T_{hi} = I_{hi} + 1$ with probability

$$(F_{hi} - F_{h(i-1)})/(1 - F_{h(i-1)})$$

And the number of hits for unit i equals $T_{hi} - T_{h(i-1)}$.

If $T_{h(i-1)} = I_{h(i-1)} + 1$, Chromy's method determines the total number of hits for the first i units as follows. If $F_{hi} = 0$, then $T_{hi} = I_{hi}$. If $F_{hi} > F_{h(i-1)}$, then $T_{hi} = I_{hi} + 1$. Otherwise, $T_{hi} = I_{hi} + 1$ with probability

$$F_{hi}/F_{h(i-1)}$$

Brewer's PPS Method

Brewer's method (METHOD=PPS_BREWER) selects two units from each stratum, with probability proportional to size and without replacement. The selection probability for unit i in stratum h equals $2M_{hi}/M_{h.} = 2Z_{hi}$.

Brewer's algorithm first selects a unit with probability

$$\frac{Z_{hi}(1 - Z_{hi})}{D_h(1 - 2Z_{hi})}$$

where

$$D_h = \sum_{i=1}^{N_h} \frac{Z_{hi}(1 - Z_{hi})}{1 - 2Z_{hi}}$$

Then a second unit is selected from the remaining units with probability

$$\frac{Z_{hj}}{1 - Z_{hi}}$$

where unit i is the first unit selected. The joint selection probability for units i and j in stratum h equals

$$P_{h(ij)} = \frac{2Z_{hi}Z_{hj}}{D_h}\left(\frac{1 - Z_{hi} - Z_{hj}}{(1 - 2Z_{hi})(1 - 2Z_{hj})}\right)$$

Brewer's method requires that the relative size Z_{hi} be less than 0.5 for all units. See Cochran (1977, pp. 261–263) and Brewer (1963) for details. Brewer's method yields the same selection probabilities and joint selection probabilities as Durbin's method. See Cochran (1977) and Durbin (1967) for details.

Murthy's PPS Method

Murthy's method (METHOD=PPS_MURTHY) selects two units from each stratum, with probability proportional to size and without replacement. The selection probability for unit i in stratum h equals

$$P_{hi} = Z_{hi}\big(1 + K - (Z_{hi}/(1 - Z_{hi}))\big)$$

where $Z_{hi} = M_{hi}/M_{h\cdot}$ and

$$K = \sum_{j=1}^{N}\big(Z_{hj}/(1 - Z_{hj})\big)$$

Murthy's algorithm first selects a unit with probability Z_{hi}. Then a second unit is selected from the remaining units with probability $Z_{hj}/(1 - Z_{hi})$, where unit i is the first unit selected. The joint selection probability for units i and j in stratum h equals

$$P_{h(ij)} = Z_{hi}Z_{hj}\left(\frac{2 - Z_{hi} - Z_{hj}}{(1 - Z_{hi})(1 - Z_{hj})}\right)$$

See Cochran (1977, pp. 263–265) and Murthy (1957) for details.

Sampford's PPS Method

Sampford's method (METHOD=PPS_SAMPFORD) is an extension of Brewer's method that selects more than two units from each stratum, with probability proportional to size and without replacement. The selection probability for unit i in stratum h equals $n_h M_{hi}/M_{h\cdot} = n_h Z_{hi}$.

Sampford's method first selects a unit from stratum h with probability Z_{hi}. Then subsequent units are selected with probability proportional to

$$\lambda_i = \frac{Z_{hi}}{1 - n_h\, Z_{hi}}$$

and with replacement. If the same unit appears more than once in the sample of size n_h, then Sampford's algorithm rejects that sample and selects a new sample. The sample is accepted if it contains n_h distinct units.

If you specify the JTPROBS option, PROC SURVEYSELECT computes the joint selection probabilities for all pairs of selected units in each stratum. The joint selection probability for units i and j in stratum h equals

$$P_{h(ij)} = K_h \lambda_i \lambda_j \sum_{t=2}^{n_h} \Big(t - n_h \left(P_{hi} + P_{hj}\right) L_{n_h - t}(ij) \Big) / n_h^{t-2}$$

where

$$L_m = \sum_{S(m)} \lambda_{i_1} \lambda_{i_2} \cdots \lambda_{i_m}$$

and $S(m)$ denotes all possible samples of size m, for $m = 1, 2, \ldots, N_h$. The sum $L_m(ij)$ is defined similarly to L_m but sums over all possible samples of size m that do not include units i and j, and

$$K_h = \left(\sum_{t=1}^{n_h} t \; L_{n_h - t} / n_h^t \right)^{-1}$$

Sampford's method requires that the relative size Z_{hi} be less than $1/n_h$ for all units. See Cochran (1977, pp. 262–263) and Sampford (1967) for details.

Sample Size Allocation

If you specify the ALLOC= option in the STRATA statement, PROC SURVEYSELECT allocates the total sample size among the strata according to the method you request. You specify the total sample size in the SAMPSIZE=*n* option in the PROC SURVEYSELECT statement.

PROC SURVEYSELECT provides proportional allocation (ALLOC=PROP), optimal allocation (ALLOC=OPTIMAL), and Neyman allocation (ALLOC=NEYMAN). See Lohr (1999), Kish (1965), and Cochran (1977) for more information about these allocation methods. Alternatively, you can directly specify the allocation proportions by using the ALLOC=(*values*) option or the ALLOC=*SAS-data-set* option. Then PROC SURVEYSELECT allocates the total sample size among the strata according to the proportions that you specify.

Proportional Allocation

When you specify the ALLOC=PROP option, PROC SURVEYSELECT allocates the total sample size among the strata in proportion to the stratum sizes, where the stratum size is the number of sampling units in the stratum. The allocation proportion of the total sample size for stratum h equals

$$f_h^* = N_h / N$$

where N_h is the number of sampling units in stratum h and N is the total number of sampling units for all strata. Based on this allocation proportion, the target sample size for stratum h is

$$n_h^* = f_h^* \times n$$

where n is the total sample size that you specify in the SAMPSIZE= option.

The target sample size values, n_h^*, might not be integers, but the stratum sample sizes must be integers. PROC SURVEYSELECT uses a rounding algorithm to convert the n_h^* to integer values n_h and maintain the requested total sample size n. The rounding algorithm includes the restriction that all values of n_h must be at least 1, so that at least one unit will be selected from each stratum. For without-replacement selection methods, PROC SURVEYSELECT also requires that each stratum sample size must not exceed the total number of sampling units in the stratum, $n_h <= N_h$. If a target stratum sample size exceeds the number of units in the stratum, PROC SURVEYSELECT allocates the maximum number of units, N_h, to the stratum, and then allocates the remaining total sample size proportionally among the remaining strata.

PROC SURVEYSELECT provides the target allocation proportions f_h^* in the output data set variable AllocProportion. The variable ActualProportion contains the actual proportions for the allocated sample sizes n_h. For stratum h, the actual proportion is computed as

$$f_h = n_h/n$$

where n_h is the allocated sample size for stratum h, and n is the total sample size. The actual proportions f_h can differ from the target allocation proportions f_h^* due to rounding and the restrictions that $n_h >= 1$ and $n_h <= N_h$.

Optimal Allocation

When you specify the ALLOC=OPTIMAL option, PROC SURVEYSELECT allocates the total sample size among the strata in proportion to stratum sizes, stratum costs, and stratum variances. Optimal allocation minimizes the overall variance for a specified cost, or equivalently minimizes the overall cost for a specified variance. See Lohr (1999), Cochran (1977) and Kish (1965) for details. For optimal allocation, the proportion of the total sample size for stratum h is computed as

$$f_h^* = \frac{N_h S_h}{\sqrt{C_h}} \Big/ \sum_{i=1}^{H} \frac{N_i S_i}{\sqrt{C_i}}$$

where N_h is the number of sampling units in stratum h, S_h is the standard deviation within stratum h, C_h is the unit cost within stratum h, and H is the total number of strata. The target sample size for stratum h is $n_h^* = f_h^* \times n$, where n is the total sample size. As for proportional allocation, the values of n_h^* are converted to integer sample sizes n_h by using a rounding algorithm that requires the sum of the stratum sample sizes to equal n. The final sample sizes n_h are also required to be at least 1, and the final sample sizes must not exceed the stratum sizes for without-replacement selection methods.

Neyman Allocation

When you specify the ALLOC=NEYMAN option, PROC SURVEYSELECT allocates the total sample size among the strata in proportion to stratum sizes and stratum variances. Neyman allocation is a special case of optimal allocation, where the costs per unit are the same for all strata. For Neyman allocation, the proportion of the total sample size for stratum h is computed as

$$f_h^* = N_h S_h / \sum_{i=1}^{H} N_i S_i$$

The target sample size for stratum h is $n_h^* = f_h^* \times n$. The n_h^* are converted to integer sample sizes n_h by using a rounding algorithm that requires the sum of the stratum sizes to equal n. The final sample sizes n_h are required to be at least 1, and the final sample sizes must not exceed the stratum sizes for without-replacement selection methods.

Secondary Input Data Set

The primary input data set for PROC SURVEYSELECT is the DATA= data set, which contains the list of units from which the sample is selected. You can use a secondary input data set to provide stratum-level design and selection information, such as sample sizes or rates, certainty size values, or stratum costs. This secondary input data set is sometimes called the SAMPSIZE= input data set. You can provide stratum sample sizes in the _NSIZE_ (or SampleSize) variable in the SAMPSIZE= data set.

The secondary input data set must contain all the STRATA variables, with the same type and length as in the DATA= data set. The STRATA groups should appear in the same order in the secondary data set as in the DATA= data set. You can name only one secondary data set in each invocation of the procedure.

You must name the secondary input data set in the appropriate PROC SURVEYSELECT or STRATA option, and use the designated variable name to provide the stratum-level values. For example, if you want to provide stratum-level costs for sample allocation, you name the secondary data set in the COST=*SAS-data-set* option in the STRATA statment. The data set must include the stratum costs in a variable named _COST_. You can use the secondary input data set for more than one option if it is appropriate for your design. For example, the secondary data set can include both stratum costs and stratum variances, which are required for optimal allocation (ALLOC=OPTIMAL).

Instead of using a separate secondary input data set, you can include secondary information in the DATA= data set along with the sampling frame. When you include secondary information in the DATA= data set, name the DATA= data set in the appropriate options, and include the required variables in the DATA= data set.

Table 87.2 lists the available secondary data set variables, together with their descriptions and the corresponding options.

Table 87.2 PROC SURVEYSELECT Secondary Data Set Variables

Variable	Description	Statement	Option
ALLOC	Allocation proportion	STRATA	ALLOC=
CERTP	Certainty proportion	PROC	CERTSIZE=P=
CERTSIZE	Certainty size	PROC	CERTSIZE=
COST	Cost	STRATA	COST=
MAXSIZE	Maximum size	PROC	MAXSIZE=
MINSIZE	Minimum size	PROC	MINSIZE=
NSIZE	Sample size	PROC	SAMPSIZE=
RATE	Sampling rate	PROC	SAMPRATE=
SEED	Random number seed	PROC	SEED=
VAR	Variance	STRATA	VAR=

Sample Output Data Set

Unless you specify the NOSAMPLE option in the STRATA statement, PROC SURVEYSELECT selects a sample and creates a SAS data set that contains the sample of selected units. If you specify the NOSAMPLE option, PROC SURVEYSELECT allocates the total sample size among the strata but does not select the sample. When you specify the NOSAMPLE option, the output data set contains the allocated sample sizes. See the section "Allocation Output Data Set" on page 6660 for details.

You can specify the name of the sample output data set in the OUT= option in the PROC SURVEYSELECT statement. If you omit the OUT= option, the data set is named DATA*n*, where *n* is the smallest integer that makes the name unique.

By default, the output data set contains one observation for each unit selected for the sample. But if you specify the OUTALL option, the output data set includes all observations from the input data set. With OUTALL, the output data set also contains a variable to indicate each observation's selection status. The variable Selected equals 1 for an observation selected for the sample, and equals 0 for an observation not selected. The OUTALL option is available only for equal probability selection methods.

By default, the output data set contains one observation for each selected unit, even if the unit is selected more than once, and the variable NumberHits contains the number of hits or selections for that unit. A unit might be selected more than once if you use a with-replacement or with-minimum-replacement selection method (METHOD=URS, METHOD=PPS_WR, METHOD=PPS_SYS, or METHOD=PPS_SEQ). If you specify the OUTHITS option, the output data set contains a separate observation for each hit or selection.

The output data set contains design information and selection statistics, depending on the selection method and output options you specify. The output data set can include the following variables:

- Selected, which indicates whether or not the observation is selected for the sample. This variable is included if you specify the OUTALL option. Selected equals 1 for an observation selected for the sample or 0 for an observation not selected.

- STRATA variables, which you specify in the STRATA statement
- Replicate, which is the sample replicate number. This variable is included when you request replicated sampling with the REPS= option.
- ID variables, which you name in the ID statement
- CONTROL variables, which you specify in the CONTROL statement
- Zone, which is the selection zone. This variable is included for METHOD=PPS_SEQ.
- SIZE variable, which you specify in the SIZE statement
- AdjustedSize, which is the adjusted size measure. This variable is included if you request adjusted sizes with the MINSIZE= or MAXSIZE= option.
- Certain, which indicates certainty selection. This variable is included if you specify the CERTSIZE= or CERTSIZE=P= option. Certain equals 1 for units included with certainty because their size measures exceed the certainty size value or the certainty proportion; otherwise, Certain equals 0.
- NumberHits, which is the number of hits or selections. This variable is included for selection methods that are with replacement or with minimum replacement (METHOD=URS, METHOD=PPS_WR, METHOD=PPS_SYS, and METHOD=PPS_SEQ).

The output data set includes the following variables if you request a PPS selection method or if you specify the STATS option for other methods:

- ExpectedHits, which is the expected number of hits or selections. This variable is included for selection methods that are with replacement or with minimum replacement, and so might select the same unit more than once (METHOD=URS, METHOD=PPS_WR, METHOD=PPS_SYS, and METHOD=PPS_SEQ).
- SelectionProb, which is the probability of selection. This variable is included for selection methods that are without replacement.
- SamplingWeight, which is the sampling weight. This variable equals the inverse of ExpectedHits or SelectionProb.

For METHOD=PPS_BREWER and METHOD=PPS_MURTHY, which select two units from each stratum with probability proportional to size, the output data set contains the following variable:

- JtSelectionProb, which is the joint probability of selection for the two units selected from the stratum

If you specify the JTPROBS option to compute joint probabilities of selection for METHOD=PPS or METHOD=PPS_SAMPFORD, then the output data set contains the following variables:

- Unit, which is an identification variable that numbers the selected units sequentially within each stratum

- JtProb_1, JtProb_2, JtProb_3, . . . , where the variable JtProb_1 contains the joint probability of selection for the current unit and unit 1. Similarly, JtProb_2 contains the joint probability of selection for the current unit and unit 2, and so on.

If you specify the JTPROBS option for METHOD=PPS_WR, then the output data set contains the following variables:

- Unit, which is an identification variable that numbers the selected units sequentially within each stratum
- JtHits_1, JtHits_2, JtHits_3, . . . , where the variable JtHits_1 contains the joint expected number of hits for the current unit and unit 1. Similarly, JtHits_2 contains the joint expected number of hits for the current unit and unit 2, and so on.

If you specify the OUTSIZE option, the output data set contains the following variables. If you specify a STRATA statement, the output data set includes stratum-level values of these variables. Otherwise, the output data set contains population-level values of these variables.

- MinimumSize, which is the minimum size measure specified with the MINSIZE= option. This variable is included if you specify the MINSIZE= option.
- MaximumSize, which is the maximum size measure specified with the MAXSIZE= option. This variable is included if you specify the MAXSIZE= option.
- CertaintySize, which is the certainty size measure specified with the CERTSIZE= option. This variable is included if you specify the CERTSIZE= option.
- CertaintyProp, which is the certainty proportion specified with the CERTSIZE=P= option. This variable is included if you specify the CERTSIZE=P= option.
- Total, which is the total number of sampling units in the stratum. This variable is included if there is no SIZE statement.
- TotalSize, which is the total of size measures in the stratum. This variable is included if there is a SIZE statement.
- TotalAdjSize, which is the total of adjusted size measures in the stratum. This variable is included if you specify a SIZE statement and if you request adjusted sizes with the MAXSIZE= or MINSIZE= option.
- SamplingRate, which is the sampling rate. This variable is included if you specify the SAMPRATE= option.
- SampleSize, which is the sample size. This variable is included if you specify the SAMPSIZE= option, or if you specify METHOD=PPS_BREWER or METHOD=PPS_MURTHY, which selects two units from each stratum.

If you specify the OUTSEED option, the output data set contains the following variable:

- InitialSeed, which is the initial seed for the stratum.

If you specify the ALLOC= option in the STRATA statement, the output data set contains the following variables:

- Total, which is the total number of sampling units in the stratum
- Variance, which is the stratum variance. This variable is included if you specify the VAR, VAR=(*values*), or VAR=*SAS-data-set* option for ALLOC=OPTIMAL or ALLOC=NEYMAN.
- Cost, which is the stratum cost. This variable is included if you specify the COST, COST=(*values*), or COST=*SAS-data-set* option for ALLOC=OPTIMAL.
- AllocProportion, which is the target allocation proportion, or the proportion of the total sample size to allocate to the stratum. PROC SURVEYSELECT computes this proportion by using the specified allocation method.
- SampleSize, which is the sample size allocated to the stratum
- ActualProportion, which is the actual proportion allocated to the stratum. The value of ActualProportion equals the allocated stratum sample size divided by the total sample size. This value can differ from the target AllocProportion due to rounding and other restrictions. See the section "Sample Size Allocation" on page 6654 for details.

Allocation Output Data Set

When you specify the NOSAMPLE option in the STRATA statement, PROC SURVEYSELECT allocates the total sample size among the strata but does not select the sample. In this case, the OUT= data set contains the allocated sample sizes.

You can specify the name of the allocation output data set with the OUT= option in the PROC SURVEYSELECT statement. If you omit the OUT= option, the data set is named DATA*n*, where *n* is the smallest integer that makes the name unique.

The allocation output data set can include the following variables:

- STRATA variables, which you specify in the STRATA statement
- Total, which is the total number of sampling units in the stratum
- Variance, which is the stratum variance. This variable is included if you specify the VAR, VAR=(*values*), or VAR=*SAS-data-set* option for ALLOC=OPTIMAL or ALLOC=NEYMAN.

- Cost, which is the stratum cost. This variable is included if you specify the COST, COST=(*values*), or COST=*SAS-data-set* option for ALLOC=OPTIMAL.
- AllocProportion, which is the target allocation proportion, or the proportion of the total sample size to allocate to the stratum. PROC SURVEYSELECT computes this proportion by using the specified allocation method.
- SampleSize, which is the sample size allocated to the stratum
- ActualProportion, which is the actual proportion allocated to the stratum. The value of ActualProportion equals the allocated stratum sample size divided by the total sample size. This value can differ from the target AllocProportion due to rounding and other restrictions. See the section "Sample Size Allocation" on page 6654 for details.

Displayed Output

By default, PROC SURVEYSELECT displays two tables that summarize the sample selection, the "Sample Selection Method" table and the "Sample Selection Summary" table.

If you request sample allocation but no sample selection, PROC SURVEYSELECT displays two tables that summarize the allocation, the "Sample Allocation Method" table and the "Sample Allocation Summary" table.

You can suppress display of these tables by specifying the NOPRINT option.

PROC SURVEYSELECT creates an output data set that contains the units selected for the sample. Or if you request sample allocation but no sample selection, PROC SURVEYSELECT creates an output data set that contains the sample size allocation results. (See the sections "Sample Output Data Set" on page 6657 and "Allocation Output Data Set" on page 6660 for information about these output data sets.) The procedure does not display the output data set that it creates. Use PROC PRINT, PROC REPORT, or any other SAS reporting tool to display the output data set.

PROC SURVEYSELECT displays the following information in the "Sample Selection Method" table:

- Selection Method
- Size Measure variable, if you specify a SIZE statement
- Minimum Size Measure, if you specify the MINSIZE= option
- Maximum Size Measure, if you specify the MAXSIZE= option
- Certainty Size Measure, if you specify the CERTSIZE= option
- Certainty Proportion, if you specify the CERTSIZE=P= option
- Strata Variables, if you specify a STRATA statement
- Control Variables, if you specify a CONTROL statement

- type of Control Sorting, Serpentine or Nested, if you specify a CONTROL statement
- type of Allocation, if you specify the ALLOC= option in the STRATA statement

PROC SURVEYSELECT displays the following information in the "Sample Selection Summary" table:

- Input Data Set name
- Sorted Data Set name, if you specify the OUTSORT= option
- Random Number Seed
- Sample Size or Stratum Sample Size, if you specify the SAMPSIZE=*n* option
- Sample Size Data Set, if you specify the SAMPSIZE=*SAS-data-set* option
- Sampling Rate or Stratum Sampling Rate, if you specify the SAMPRATE=*r* option
- Sampling Rate Data Set, if you specify the SAMPRATE=*SAS-data-set* option
- Minimum Sample Size or Stratum Minimum Sample Size, if you specify the NMIN= option with the SAMPRATE= option
- Maximum Sample Size or Stratum Maximum Sample Size, if you specify the NMAX= option with the SAMPRATE= option
- Allocation Input Data Set name, if you specify the ALLOC=*SAS-data-set* option in the STRATA statement
- Variance Input Data Set name, if you specify the VAR=*SAS-data-set* option in the STRATA statement.
- Cost Input Data Set name, if you specify the COST=*SAS-data-set* option in the STRATA statement.
- Selection Probability, if you specify METHOD=SRS, METHOD=SYS, or METHOD=SEQ and do not specify a STRATA statement
- Expected Number of Hits, if you specify METHOD=URS and do not specify a STRATA statement
- Sampling Weight for equal probability selection methods (METHOD=SRS, METHOD=URS, METHOD=SYS, METHOD=SEQ) if you do not specify a STRATA statement
- Number of Strata, if you specify a STRATA statement
- Number of Replicates, if you specify the REPS= option
- Total Sample Size, if you specify a STRATA statement or the REPS= option
- Output Data Set name

If you specify the NOSAMPLE option in the STRATA statement, PROC SURVEYSELECT allocates the total sample among the strata but does not select the sample. When you specify the NOSAMPLE option, PROC SURVEYSELECT displays the "Sample Allocation Method" table, which includes the following information:

- Allocation method
- Strata Variables

When you specify the NOSAMPLE option in the STRATA statement, PROC SURVEYSELECT also displays the "Sample Allocation Summary" table, which includes the following information:

- Input Data Set name
- Allocation Input Data Set name, if you specify the ALLOC=*SAS-data-set* option in the STRATA statement
- Variance Input Data Set name, if you specify the VAR=*SAS-data-set* option in the STRATA statement
- Cost Input Data Set name, if you specify the COST=*SAS-data-set* option in the STRATA statement
- Number of Strata
- Total Sample Size
- Allocation Output Data Set name

ODS Table Names

PROC SURVEYSELECT assigns a name to each table it creates. You can use these names to reference tables when using the Output Delivery System (ODS) to select tables and create output data sets. For more information about ODS, see Chapter 20, "Using the Output Delivery System." Table 87.3 lists the table names.

Table 87.3 ODS Tables Produced by PROC SURVEYSELECT

ODS Table Name	Description	Statement	Option
Method	Sample selection method	PROC	default
Summary	Sample selection summary	PROC	default

Examples: SURVEYSELECT Procedure

Example 87.1: Replicated Sampling

This example uses the Customers data set from the section "Getting Started: SURVEYSELECT Procedure" on page 6615. The data set Customers contains an Internet service provider's current subscribers, and the service provider wants to select a sample from this population for a customer satisfaction survey.

This example illustrates replicated sampling, which selects multiple samples from the survey population according to the same design. You can use replicated sampling to provide a simple method of variance estimation, or to evaluate variable nonsampling errors such as interviewer differences. See Lohr (1999), Wolter (1985), Kish (1965, 1987), and Kalton (1983) for information about replicated sampling.

This design includes four replicates, each with a sample size of 50 customers. The sampling frame is stratified by State and sorted by Type and Usage within strata. Customers are selected by sequential random sampling with equal probability within strata. The following PROC SURVEYSELECT statements select a probability sample of customers from the Customers data set by using this design:

```
title1 'Customer Satisfaction Survey';
title2 'Replicated Sampling';
proc surveyselect data=Customers
      method=seq n=(8 12 20 10)
      reps=4
      seed=40070 out=SampleRep;
   strata State;
   control Type Usage;
run;
```

The STRATA statement names the stratification variable State. The CONTROL statement names the control variables Type and Usage. In the PROC SURVEYSELECT statement, the METHOD=SEQ option requests sequential random sampling. The REPS=4 option specifies four replicates of this sample. The N=(8 12 20 10) option lists the stratum sample sizes for each replicate. The N= option lists the stratum sample sizes in the same order as the strata appear in the Customers data set, which has been sorted by State. The sample size of eight customers corresponds to the first stratum, State = 'AL'. The sample size 12 corresponds to the next stratum, State = 'FL', and so on. The SEED=40070 option specifies '40070' as the initial seed for random number generation.

Output 87.1.1 displays the output from PROC SURVEYSELECT, which summarizes the sample selection. A total of 200 customers is selected in four replicates. PROC SURVEYSELECT selects each replicate by using sequential random sampling within strata determined by State. The sampling frame Customers is sorted by the control variables Type and Usage within strata, according to hierarchic serpentine sorting. The output data set SampleRep contains the sample.

Output 87.1.1 Sample Selection Summary

```
                    Customer Satisfaction Survey
                        Replicated Sampling

                     The SURVEYSELECT Procedure

Selection Method      Sequential Random Sampling
                      With Equal Probability
Strata Variable       State
Control Variables     Type
                      Usage
Control Sorting       Serpentine

          Input Data Set            CUSTOMERS
          Random Number Seed            40070
          Number of Strata                  4
          Number of Replicates              4
          Total Sample Size               200
          Output Data Set           SAMPLEREP
```

The following PROC PRINT statements display the selected customers for the first stratum, State = 'AL', from the output data set SampleRep:

```
title1 'Customer Satisfaction Survey';
title2 'Sample Selected by Replicated Design';
title3 '(First Stratum)';
proc print data=SampleRep;
   where State = 'AL';
run;
```

Output 87.1.2 displays the 32 sample customers of the first stratum (State = 'AL') from the output data set SampleRep, which includes the entire sample of 200 customers. The variable SelectionProb contains the selection probability, and SamplingWeight contains the sampling weight. Because customers are selected with equal probability within strata in this design, all customers in the same stratum have the same selection probability. These selection probabilities and sampling weights apply to a single replicate, and the variable Replicate contains the sample replicate number.

Output 87.1.2 Customer Sample (First Stratum)

Customer Satisfaction Survey
Sample Selected by Replicated Design
(First Stratum)

Obs	State	Replicate	CustomerID	Type	Usage	Selection Prob	Sampling Weight
1	AL	1	882-37-7496	New	572	.004115226	243
2	AL	1	581-32-5534	New	863	.004115226	243
3	AL	1	980-29-2898	Old	571	.004115226	243
4	AL	1	172-56-4743	Old	128	.004115226	243
5	AL	1	998-55-5227	Old	35	.004115226	243
6	AL	1	625-44-3396	New	60	.004115226	243
7	AL	1	627-48-2509	New	114	.004115226	243
8	AL	1	257-66-6558	New	172	.004115226	243
9	AL	2	622-83-1680	New	22	.004115226	243
10	AL	2	343-57-1186	New	53	.004115226	243
11	AL	2	976-05-3796	New	110	.004115226	243
12	AL	2	859-74-0652	New	303	.004115226	243
13	AL	2	476-48-1066	New	839	.004115226	243
14	AL	2	109-27-8914	Old	2102	.004115226	243
15	AL	2	743-25-0298	Old	376	.004115226	243
16	AL	2	722-08-2215	Old	105	.004115226	243
17	AL	3	668-57-7696	New	200	.004115226	243
18	AL	3	300-72-0129	New	471	.004115226	243
19	AL	3	073-60-0765	New	656	.004115226	243
20	AL	3	526-87-0258	Old	672	.004115226	243
21	AL	3	726-61-0387	Old	150	.004115226	243
22	AL	3	632-29-9020	Old	51	.004115226	243
23	AL	3	417-17-8378	New	56	.004115226	243
24	AL	3	091-26-2366	New	93	.004115226	243
25	AL	4	336-04-1288	New	419	.004115226	243
26	AL	4	827-04-7407	New	650	.004115226	243
27	AL	4	317-70-6496	Old	452	.004115226	243
28	AL	4	002-38-4582	Old	206	.004115226	243
29	AL	4	181-83-3990	Old	33	.004115226	243
30	AL	4	675-34-7393	New	47	.004115226	243
31	AL	4	228-07-6671	New	65	.004115226	243
32	AL	4	298-46-2434	New	161	.004115226	243

Example 87.2: PPS Selection of Two Units per Stratum

This example describes hospital selection for a survey by using PROC SURVEYSELECT. A state health agency plans to conduct a statewide survey of a variety of different hospital services. The agency plans to select a probability sample of individual discharge records within hospitals by using a two-stage sample design. First-stage units are hospitals, and second-stage units are patient discharges during the study period. Hospitals are stratified first according to geographic region and then by rural/urban type and size of hospital. Two hospitals are selected from each stratum with probability proportional to size.

The data set HospitalFrame contains all hospitals in the first geographical region of the state:

```
data HospitalFrame;
   input Hospital$ Type$ SizeMeasure @@;
   if (SizeMeasure < 20) then Size='Small ';
      else if (SizeMeasure < 50) then Size='Medium';
      else Size='Large ';
   datalines;
034 Rural  0.870   107 Rural  1.316
079 Rural  2.127   223 Rural  3.960
236 Rural  5.279   165 Rural  5.893
086 Rural  0.501   141 Rural 11.528
042 Urban  3.104   124 Urban  4.033
006 Urban  4.249   261 Urban  4.376
195 Urban  5.024   190 Urban 10.373
038 Urban 17.125   083 Urban 40.382
259 Urban 44.942   129 Urban 46.702
133 Urban 46.992   218 Urban 48.231
026 Urban 61.460   058 Urban 65.931
119 Urban 66.352
;
```

In the SAS data set HospitalFrame, the variable Hospital identifies the hospital. The variable Type equals 'Urban' if the hospital is located in an urban area, and 'Rural' otherwise. The variable SizeMeasure contains the hospital's size measure, which is constructed from past data on service utilization for the hospital together with the desired sampling rates for each service. This size measure reflects the amount of relevant survey information expected from the hospital. See Drummond et al. (1982) for details about this type of size measure. The variable Size equals 'Small', 'Medium', or 'Large', depending on the value of the hospital's size measure.

The following PROC PRINT statements display the data set Hospital Frame and produce Output 87.2.1:

```
title1 'Hospital Utilization Survey';
title2 'Sampling Frame, Region 1';
proc print data=HospitalFrame;
run;
```

Output 87.2.1 Sampling Frame

Hospital Utilization Survey
Sampling Frame, Region 1

Obs	Hospital	Type	Size Measure	Size
1	034	Rural	0.870	Small
2	107	Rural	1.316	Small
3	079	Rural	2.127	Small
4	223	Rural	3.960	Small
5	236	Rural	5.279	Small
6	165	Rural	5.893	Small
7	086	Rural	0.501	Small
8	141	Rural	11.528	Small
9	042	Urban	3.104	Small
10	124	Urban	4.033	Small
11	006	Urban	4.249	Small
12	261	Urban	4.376	Small
13	195	Urban	5.024	Small
14	190	Urban	10.373	Small
15	038	Urban	17.125	Small
16	083	Urban	40.382	Medium
17	259	Urban	44.942	Medium
18	129	Urban	46.702	Medium
19	133	Urban	46.992	Medium
20	218	Urban	48.231	Medium
21	026	Urban	61.460	Large
22	058	Urban	65.931	Large
23	119	Urban	66.352	Large

The following PROC SURVEYSELECT statements select a probability sample of hospitals from the HospitalFrame data set by using a stratified design with PPS selection of two units from each stratum:

```
title1 'Hospital Utilization Survey';
title2 'Stratified PPS Sampling';
proc surveyselect data=HospitalFrame
      method=pps_brewer
      seed=48702 out=SampleHospitals;
   size SizeMeasure;
   strata Type Size notsorted;
run;
```

The STRATA statement names the stratification variables Type and Size. The NOTSORTED option specifies that observations with the same STRATA variable values are grouped together but are not necessarily sorted in alphabetical or increasing numerical order. In the HospitalFrame data set, Size = 'Small' precedes Size = 'Medium'.

In the PROC SURVEYSELECT statement, the METHOD=PPS_BREWER option requests sample selection by Brewer's method, which selects two units per stratum with probability proportional to size. The SEED=48702 option specifies '48702' as the initial seed for random number generation.

The SIZE statement names SizeMeasure as the size measure variable. It is not necessary to specify the sample size with the N= option, because Brewer's method always selects two units from each stratum.

Output 87.2.2 displays the output from PROC SURVEYSELECT. A total of eight hospitals were selected from the four strata. The data set SampleHospitals contains the selected hospitals.

Output 87.2.2 Sample Selection Summary

Hospital Utilization Survey
Stratified PPS Sampling

The SURVEYSELECT Procedure

Selection Method	Brewer's PPS Method
Size Measure	SizeMeasure
Strata Variables	Type
	Size

Input Data Set	HOSPITALFRAME
Random Number Seed	48702
Stratum Sample Size	2
Number of Strata	4
Total Sample Size	8
Output Data Set	SAMPLEHOSPITALS

The following PROC PRINT statements display the sample hospitals and produce Output 87.2.3:

```
title1 'Hospital Utilization Survey';
title2 'Sample Selected by Stratified PPS Design';
proc print data=SampleHospitals;
run;
```

Output 87.2.3 Sample Hospitals

Hospital Utilization Survey
Sample Selected by Stratified PPS Design

Obs	Type	Size	Hospital	Size Measure	Selection Prob	Sampling Weight	Jt Selection Prob
1	Rural	Small	079	2.127	0.13516	7.39868	0.01851
2	Rural	Small	236	5.279	0.33545	2.98106	0.01851
3	Urban	Small	006	4.249	0.17600	5.68181	0.01454
4	Urban	Small	195	5.024	0.20810	4.80533	0.01454
5	Urban	Medium	133	46.992	0.41357	2.41795	0.11305
6	Urban	Medium	218	48.231	0.42448	2.35584	0.11305
7	Urban	Large	026	61.460	0.63445	1.57617	0.31505
8	Urban	Large	058	65.931	0.68060	1.46929	0.31505

The variable SelectionProb contains the selection probability for each hospital in the sample. The variable JtSelectionProb contains the joint probability of selection for the two sample hospitals in the same stratum. The variable SamplingWeight contains the sampling weight component for this first stage of the design. The final-stage weight components, which correspond to patient record selection within hospitals, can be multiplied by the hospital weight components to obtain the overall sampling weights.

Example 87.3: PPS (Dollar-Unit) Sampling

A small company wants to audit employee travel expenses in an effort to improve the expense reporting procedure and possibly reduce expenses. The company does not have resources to examine all expense reports and wants to use statistical sampling to objectively select expense reports for audit.

The data set TravelExpense contains the dollar amount of all employee travel expense transactions during the past month:

```
data TravelExpense;
   input ID$ Amount @@;
   if (Amount < 500) then Level='1_Low ';
      else if (Amount > 1500) then Level='3_High';
      else Level='2_Avg ';
   datalines;
110  237.18   002  567.89   234  118.50
743   74.38   411 1287.23   782  258.10
216  325.36   174  218.38   568 1670.80
302  134.71   285 2020.70   314   47.80
139 1183.45   775  330.54   425  780.10
506  895.80   239  620.10   011  420.18
672  979.66   142  810.25   738  670.85
192  314.58   243   87.50   263 1893.40
496  753.30   332  540.65   486 2580.35
614  230.56   654  185.60   308  688.43
784  505.14   017  205.48   162  650.42
289 1348.34   691   30.50   545 2214.80
517  940.35   382  217.85   024  142.90
478  806.90   107  560.72
;
```

In the SAS data set TravelExpense, the variable ID identifies the travel expense report. The variable Amount contains the dollar amount of the reported expense. The variable Level equals '1_Low', '2_Avg', or '3_High', depending on the value of Amount.

In the sample design for this audit, expense reports are stratified by Level. This ensures that each of these expense levels is included in the sample and also permits a disproportionate allocation of the sample, selecting proportionately more of the expense reports from the higher levels. Within strata, the sample of expense reports is selected with probability proportional to the amount of the expense, thus giving a greater chance of selection to larger expenses. In auditing terms, this is known as monetary-unit sampling. See Wilburn (1984) for details.

PROC SURVEYSELECT requires that the input data set be sorted by the STRATA variables. The following PROC SORT statements sort the TravelExpense data set by the stratification variable Level.

```
proc sort data=TravelExpense;
   by Level;
run;
```

Output 87.3.1 displays the sampling frame data set TravelExpense, which contains 41 observations.

Output 87.3.1 Sampling Frame

Travel Expense Audit

Obs	ID	Amount	Level
1	110	237.18	1_Low
2	234	118.50	1_Low
3	743	74.38	1_Low
4	782	258.10	1_Low
5	216	325.36	1_Low
6	174	218.38	1_Low
7	302	134.71	1_Low
8	314	47.80	1_Low
9	775	330.54	1_Low
10	011	420.18	1_Low
11	192	314.58	1_Low
12	243	87.50	1_Low
13	614	230.56	1_Low
14	654	185.60	1_Low
15	017	205.48	1_Low
16	691	30.50	1_Low
17	382	217.85	1_Low
18	024	142.90	1_Low
19	002	567.89	2_Avg
20	411	1287.23	2_Avg
21	139	1183.45	2_Avg
22	425	780.10	2_Avg
23	506	895.80	2_Avg
24	239	620.10	2_Avg
25	672	979.66	2_Avg
26	142	810.25	2_Avg
27	738	670.85	2_Avg
28	496	753.30	2_Avg
29	332	540.65	2_Avg
30	308	688.43	2_Avg
31	784	505.14	2_Avg
32	162	650.42	2_Avg
33	289	1348.34	2_Avg
34	517	940.35	2_Avg
35	478	806.90	2_Avg
36	107	560.72	2_Avg
37	568	1670.80	3_High
38	285	2020.70	3_High
39	263	1893.40	3_High
40	486	2580.35	3_High
41	545	2214.80	3_High

The following PROC SURVEYSELECT statements select a probability sample of expense reports from the TravelExpense data set by using the stratified design with PPS selection within strata:

```
title1 'Travel Expense Audit';
title2 'Stratified PPS (Dollar-Unit) Sampling';
proc surveyselect data=TravelExpense
      method=pps n=(6 10 4)
      seed=47279 out=AuditSample;
   size Amount;
   strata Level;
run;
```

The STRATA statement names the stratification variable Level. The SIZE statement specifies the size measure variable Amount. In the PROC SURVEYSELECT statement, the METHOD=PPS option requests sample selection with probability proportional to size and without replacement. The N=(6 10 4) option specifies the stratum sample sizes, listing the sample sizes in the same order as the strata appear in the TravelExpense data set. The sample size of 6 corresponds to the first stratum, Level = '1_Low'; the sample size of 10 corresponds to the second stratum, Level = '2_Avg'; and 4 corresponds to the last stratum, Level = '3_High'. The SEED=47279 option specifies '47279' as the initial seed for random number generation.

Output 87.3.2 displays the output from PROC SURVEYSELECT. A total of 20 expense reports are selected for audit. The data set AuditSample contains the sample of travel expense reports.

Output 87.3.2 Sample Selection Summary

```
                    Travel Expense Audit
                Stratified PPS (Dollar-Unit) Sampling

                     The SURVEYSELECT Procedure

              Selection Method    PPS, Without Replacement
              Size Measure        Amount
              Strata Variable     Level

                  Input Data Set        TRAVELEXPENSE
                  Random Number Seed            47279
                  Number of Strata                  3
                  Total Sample Size                20
                  Output Data Set          AUDITSAMPLE
```

The following PROC PRINT statements display the audit sample, which is shown in Output 87.3.3:

```
title1 'Travel Expense Audit';
title2 'Sample Selected by Stratified PPS Design';
proc print data=AuditSample;
run;
```

Output 87.3.3 Audit Sample

Travel Expense Audit
Sample Selected by Stratified PPS Design

Obs	Level	ID	Amount	Selection Prob	Sampling Weight
1	1_Low	654	185.60	0.31105	3.21489
2	1_Low	017	205.48	0.34437	2.90385
3	1_Low	382	217.85	0.36510	2.73896
4	1_Low	614	230.56	0.38640	2.58797
5	1_Low	782	258.10	0.43256	2.31183
6	1_Low	775	330.54	0.55396	1.80518
7	2_Avg	784	505.14	0.34623	2.88823
8	2_Avg	332	540.65	0.37057	2.69853
9	2_Avg	002	567.89	0.38924	2.56909
10	2_Avg	239	620.10	0.42503	2.35278
11	2_Avg	738	670.85	0.45981	2.17479
12	2_Avg	496	753.30	0.51633	1.93676
13	2_Avg	425	780.10	0.53470	1.87022
14	2_Avg	478	806.90	0.55307	1.80810
15	2_Avg	672	979.66	0.67148	1.48925
16	2_Avg	139	1183.45	0.81116	1.23280
17	3_High	568	1670.80	0.64385	1.55316
18	3_High	263	1893.40	0.72963	1.37056
19	3_High	285	2020.70	0.77869	1.28421
20	3_High	486	2580.35	0.99435	1.00568

Example 87.4: Proportional Allocation

This example uses the Customers data set from the section "Getting Started: SURVEYSELECT Procedure" on page 6615. The data set Customers contains an Internet service provider's current subscribers, and the service provider wants to select a sample from this population for a customer satisfaction survey. This example illustrates proportional allocation, which allocates the total sample size among the strata in proportion to the strata sizes.

The section "Getting Started: SURVEYSELECT Procedure" on page 6615 gives an example of stratified sampling, where the list of customers is stratified by State and Type. Figure 87.4 displays the strata in a table of State by Type for the 13,471 customers. There are four states and two levels of Type, forming a total of eight strata. A sample of 15 customers was selected from each stratum by using the following PROC SURVEYSELECT statements:

```
title1 'Customer Satisfaction Survey';
title2 'Stratified Sampling';
proc surveyselect data=Customers
      method=srs n=15
      seed=1953 out=SampleStrata;
   strata State Type;
run;
```

The STRATA statement names the stratification variables State and Type. In the PROC SURVEYSELECT statement, the N=15 option specifies a sample size of 15 customers for each stratum.

Instead of specifying the number of customers to select from each stratum, you can specify the total sample size and request allocation of the total sample size among the strata. The following PROC SURVEYSELECT statements request proportional allocation, which allocates the total sample size in proportion to the stratum sizes:

```
title1 'Customer Satisfaction Survey';
title2 'Proportional Allocation';
proc surveyselect data=Customers
      n=1000 out=SampleSizes;
   strata State Type / alloc=prop nosample;
run;
```

The STRATA statement names the stratification variables State and Type. In the STRATA statement, the ALLOC=PROP option requests proportional allocation. The NOSAMPLE option requests that no sample be selected after the procedure computes the sample size allocation. In the PROC SURVEYSELECT statement, the N=1000 option specifies a total sample size of 1000 customers to be allocated among the strata.

Output 87.4.1 displays the output from PROC SURVEYSELECT, which summarizes the sample allocation. The total sample size of 1000 is allocated among the eight strata by using proportional allocation. The allocated sample sizes are stored in the SAS data set SampleSizes.

Output 87.4.1 Proportional Allocation Summary

```
                    Customer Satisfaction Survey
                      Proportional Allocation

                     The SURVEYSELECT Procedure

                  Allocation          Proportional
                  Strata Variables    State
                                      Type

              Input Data Set                   CUSTOMERS
              Number of Strata                         8
              Total Sample Size                     1000
              Allocation Output Data Set     SAMPLESIZES
```

The following PROC PRINT statements display the allocation output data set SampleSizes, which is shown in Output 87.4.2:

```
title1 'Customer Satisfaction Survey';
title2 'Proportional Allocation';
proc print data=SampleSizes;
run;
```

Output 87.4.2 Stratum Sample Sizes

Customer Satisfaction Survey
Proportional Allocation

Obs	State	Type	Total	Alloc Proportion	Sample Size	Actual Proportion
1	AL	New	1238	0.09190	92	0.092
2	AL	Old	706	0.05241	52	0.052
3	FL	New	2170	0.16109	161	0.161
4	FL	Old	1370	0.10170	102	0.102
5	GA	New	3488	0.25893	259	0.259
6	GA	Old	1940	0.14401	144	0.144
7	SC	New	1684	0.12501	125	0.125
8	SC	Old	875	0.06495	65	0.065

The output data set SampleSizes includes one observation for each of the eight strata, which are identified by the stratification variables State and Type. The variable Total contains the number of sampling units in the stratum, and the variable AllocProportion contains the proportion of the total sample size to allocate to the stratum. The variable SampleSize contains the allocated stratum sample size. For the first stratum (State='AL' and Type='New'), the total number of sampling units is 1238 customers, the allocation proportion is 0.09190, and the allocated sample size is 92 customers. The sum of the allocated sample sizes equals the requested total sample size of 1000 customers.

The output data set also includes the variable ActualProportion, which contains actual stratum proportions of the total sample size. The actual proportion for a stratum equals the stratum sample size divided by the total sample size. For the first stratum (State='AL' and Type='New'), the actual proportion is 0.092, while the allocation proportion is 0.09190. The target sample sizes computed from the allocation proportions are often not integers, and PROC SURVEYSELECT uses a rounding algorithm to obtain integer sample sizes and maintain the requested total sample size. Due to rounding and other restrictions, the actual proportions can differ from the target allocation proportions. See the section "Sample Size Allocation" on page 6654 for details.

If you want to use the allocated sample sizes in a later invocation of PROC SURVEYSELECT, you can name the allocation data set in the N=*SAS-data-set* option, as shown in the following PROC SURVEYSELECT statements:

```
title1 'Customer Satisfaction Survey';
title2 'Stratified Sampling';
proc surveyselect data=Customers
      method=srs n=SampleSizes
      seed=1953 out=SampleStrata;
   strata State Type;
run;
```

References

Bentley, J. L. and Floyd, R. (1987), "A Sample of Brilliance," *Communications of the Association for Computing Machinery*, 30, 754–757.

Bentley, J. L. and Knuth, D. (1986), "Literate Programming," *Communications of the Association for Computing Machinery*, 29, 364–369.

Brewer, K. W. R. (1963), "A Model of Systematic Sampling with Unequal Probabilities," *Australian Journal of Statistics*, 5, 93–105.

Chromy, J. R. (1979), "Sequential Sample Selection Methods," *Proceedings of the American Statistical Association, Survey Research Methods Section*, 401–406.

Cochran, W. G. (1977), *Sampling Techniques*, Third Edition, New York: John Wiley & Sons.

Drummond, D., Lessler, J., Watts, D., and Williams, S. (1982), "A Design for Achieving Prespecified Levels of Representation for Multiple Domains in Health Record Samples," *Proceedings of the Fourth Conference on Health Survey Research Methods*, DHHS Publication No. (PHS) 84-3346, Washington, DC: National Center for Health Services Research, 233–248.

Durbin, J. (1967), "Design of Multi-stage Surveys for the Estimation of Sampling Errors," *Applied Statistics*, 16, 152–164.

Fan, C. T., Muller, M. E., and Rezucha, I. (1962), "Development of Sampling Plans by Using Sequential (Item by Item) Selection Techniques and Digital Computers," *Journal of the American Statistical Association*, 57, 387–402.

Fishman, G. S. and Moore, L. R. (1982), "A Statistical Evaluation of Multiplicative Congruential Generators with Modulus $(2^{31} - 1)$," *Journal of the American Statistical Association*, 77, 129–136.

Fox, D. R. (1989), "Computer Selection of Size-Biased Samples," *The American Statistician*, 43(3), 168–171.

Golmant, J. (1990), "Correction: Computer Selection of Size-Biased Samples," *The American Statistician*, 44(2), 194.

Hanurav, T. V. (1967), "Optimum Utilization of Auxiliary Information: π_{ps} Sampling of Two Units from a Stratum," *Journal of the Royal Statistical Society, Series B*, 29, 374–391.

Kalton, G. (1983), *Introduction to Survey Sampling*, Sage University Paper series on Quantitative Applications in the Social Sciences, series no. 07-035, Beverly Hills, CA and London: Sage Publications.

Kish, L. (1965), *Survey Sampling*, New York: John Wiley & Sons.

Kish, L. (1987), *Statistical Design for Research*, New York: John Wiley & Sons.

Lohr, S. L. (1999), *Sampling: Design and Analysis*, Pacific Grove, CA: Duxbury Press.

Madow, W. G. (1949), "On the Theory of Systematic Sampling, II," *Annals of Mathematical Statistics*, 20, 333–354.

McLeod, A. I. and Bellhouse, D. R. (1983), "A Convenient Algorithm for Drawing a Simple Random Sample," *Applied Statistics*, 32, 182–183.

Murthy, M. N. (1957), "Ordered and Unordered Estimators in Sampling without Replacement," *Sankhyā*, 18, 379–390.

Murthy, M. N. (1967), *Sampling Theory and Methods*, Calcutta: Statistical Publishing Society.

Sampford, M. R. (1967), "On Sampling without Replacement with Unequal Probabilities of Selection," *Biometrika*, 54, 499–513.

Vijayan, K. (1968), "An Exact π_{ps} Sampling Scheme: Generalization of a Method of Hanurav," *Journal of the Royal Statistical Society, Series B*, 30, 556–566.

Watts, D. L. (1991), "Correction: Computer Selection of Size-Biased Samples," *The American Statistician*, 45(2), 172.

Wilburn, A. J. (1984), *Practical Statistical Sampling for Auditors*, New York: Marcel Dekker.

Williams, R. L. and Chromy, J. R. (1980), "SAS Sample Selection Macros," *Proceedings of the Fifth Annual SAS Users Group International Conference*, 5, 392–396.

Wolter, K. M. (1985), *Introduction to Variance Estimation*, New York: Springer-Verlag.

Chapter 88

The TCALIS Procedure (Experimental)

Contents

Overview: TCALIS Procedure **6681**
A Guide to the PROC TCALIS Documentation **6685**
Guide to the Basic Skills Level 6686
Guide to the Intermediate Skills Level 6691
Guide to the Advanced Skills Level 6692
Reference Topics 6694
Changes and Enhancement from PROC CALIS **6695**
New Features 6695
Inactive Statements and Options 6699
New Statements and Options 6700
Changes in ODS Table Names 6701
Getting Started: TCALIS Procedure **6704**
A Structural Equation Example 6704
A Factor Model Example 6710
Direct Covariance Structures Analysis 6712
Which Modeling Language? 6714
Syntax: TCALIS Procedure **6715**
Classes of Statements 6716
Single-Group Analysis Syntax 6719
Multiple-Group Multiple-Model Analysis Syntax 6720
PROC TCALIS Statement 6721
BASEMODEL Statement 6745
BOUNDS Statement 6746
BY Statement 6746
COV Statement 6747
DETERM Statement 6750
EFFPART Statement 6751
FACTOR Statement 6752
FITINDEX Statement 6760
FREQ Statement 6763
GROUP Statement 6764
LINCON Statement 6766
LINEQS Statement 6766
LISMOD Statement 6772

LMTESTS Statement 6776
MATRIX Statement 6784
MEAN Statement 6788
MODEL Statement 6789
MSTRUCT Statement 6792
NLINCON Statement 6794
NLOPTIONS Statement 6795
OUTFILES Statement 6796
PARAMETERS Statement 6798
PARTIAL Statement 6798
PATH Statement 6799
PCOV Statement 6802
PVAR Statement 6803
RAM Statement 6805
REFMODEL Statement 6811
RENAMEPARM Statement 6812
SAS Programming Statements 6814
SIMTEST Statement 6814
STD Statement 6815
STRUCTEQ Statement 6817
TESTFUNC Statement 6817
VAR Statement 6818
WEIGHT Statement 6818
Details: TCALIS Procedure **6819**
Input Data Sets 6819
Output Data Sets 6823
The LINEQS Model 6839
The LISMOD Model and Submodels 6847
The RAM Model 6855
The FACTOR Model 6863
The PATH Model 6870
The MSTRUCT Model 6877
Naming Variables and Parameters 6880
Setting Constraints on Parameters 6882
Automatic Variable Selection 6887
Estimation Criteria 6888
Relationships among Estimation Criteria 6893
Gradient, Hessian, Information Matrix, and Approximate Standard Errors . . 6894
Counting the Degrees of Freedom 6897
Assessment of Fit 6900
Total, Direct, and Indirect Effects 6913
Standardized Solutions 6915
Modification Indices 6917
Missing Values 6919

Measures of Multivariate Kurtosis 6919
Initial Estimates 6922
Use of Optimization Techniques 6923
Computational Problems 6932
Displayed Output 6935
ODS Table Names 6939
ODS Graphics 6952
Examples: TCALIS Procedure **6953**
Example 88.1: Path Analysis: Stability of Alienation 6953
Example 88.2: Simultaneous Equations with Mean Structures and Reciprocal Paths 6967
Example 88.3: A Direct Covariance Structures Model 6972
Example 88.4: Confirmatory Factor Analysis: Cognitive Abilities 6976
Example 88.5: Testing Equality of Two Covariance Matrices Using a Multiple-Group Analysis 6987
Example 88.6: Illustrating Various General Modeling Languages 6991
Example 88.7: Fitting a Latent Growth Curve Model 6999
Example 88.8: Higher-Order and Hierarchical Factor Models 7005
Example 88.9: Linear Relations among Factor Loadings 7021
Example 88.10: A Multiple-Group Model for Purchasing Behavior 7030
References **7052**

Overview: TCALIS Procedure

Structural equation modeling is an important statistical tool in social and behavioral sciences. Structural equations express relationships among a system of variables that can be either observed variables (manifest variables) or unobserved hypothetical variables (latent variables). For an introduction to latent variable models, see Loehlin (2004), Bollen (1989b), Everitt (1984), or Long (1983); and for manifest variables with measurement errors, see Fuller (1987).

In structural models, as opposed to functional models, all variables are taken to be random rather than having fixed levels. For maximum likelihood (default) and generalized least squares estimation in PROC TCALIS, the random variables are assumed to have an approximately multivariate normal distribution. Nonnormality, especially high kurtosis, can produce poor estimates and grossly incorrect standard errors and hypothesis tests, even in large samples. Consequently, the assumption of normality is much more important than in models with nonstochastic exogenous variables. You should remove outliers and consider transformations of nonnormal variables before using PROC TCALIS with maximum likelihood (default) or generalized least squares estimation. If the number of observations is sufficiently large, Browne's asymptotically distribution-free (ADF) estimation method can be used.

You can use the TCALIS procedure to estimate parameters and test hypotheses for constrained and unconstrained problems in various situations, including but not limited to the following:

- exploratory and confirmatory factor analysis of any order
- linear measurement-error models or regression with errors in variables
- multiple and multivariate linear regression
- multiple-group structural equation modeling with mean and covariance structures
- path analysis and causal modeling
- simultaneous equation models with reciprocal causation
- structured covariance and mean matrices

To specify models in PROC TCALIS, you can use a variety of modeling languages:

- FACTOR—supports the input of factor-variable relations
- LINEQS—like the EQS program (Bentler 1995), uses equations to describe variable relationships
- LISMOD—utilizes LISREL (Jöreskog and Sörbom 1985) model matrices for defining models
- MSTRUCT—supports direct parameterization in the mean and covariance matrices
- PATH—provides an intuitive causal path specification interface
- RAM—utilizes the formulation of the reticular action model (McArdle and McDonald 1984)
- REFMODEL—provides a quick way for model referencing and respecification

Various modeling languages are provided to suit a wide range of researchers' background and modeling philosophy. However, statistical situations might arise when one modeling language is more convenient than the others. This will be discussed in the section "Which Modeling Language?" on page 6714.

In addition to basic model specification, you can set various parameter constraints in PROC TCALIS. Equality constraints on parameters can be achieved by simply giving the same parameter names in different parts of the model. Boundary, linear, and nonlinear constraints are supported as well. If parameters in the model are dependent on additional parameters, you can define the dependence by using the PARAMETERS and the SAS programming statements.

Before the data are analyzed, researchers might be interested in studying some statistical properties of the data. PROC TCALIS can provide the following statistical summary of the data:

- covariance and mean matrices and their properties
- descriptive statistics like means, standard deviations, univariate skewness, and kurtosis measures
- multivariate measures of kurtosis
- weight matrix and its descriptive properties

After a model is fitted and accepted by the researcher, PROC TCALIS can provide the following supplementary statistical analysis:

- computing squared multiple correlations and determination coefficients
- direct and indirect effects partitioning with standard error estimates
- model modification tests such as Lagrange multiplier and Wald tests
- computing fit summary indices
- computing predicted moments of the model
- residual analysis
- factor rotations
- standardized solutions with standard errors
- testing parametric functions, individually or simultaneously

When fitting a model, you need to choose an estimation method. The following estimation methods are supported in the TCALIS procedure:

- diagonally weighted least squares (DWLS, with optional weight matrix input)
- generalized least squares (GLS, with optional weight matrix input)
- maximum likelihood (ML, for multivariate normal data); this is the default method
- unweighted least squares (ULS)
- weighted least squares or asymptotically distribution-free method (WLS or ADF, with optional weight matrix input)

Estimation methods implemented in PROC TCALIS do not exhaust all alternatives in the field. For example, partial least squares (PLS) is not implemented. See the section "Estimation Criteria" on page 6888 for details about estimation criteria used in PROC TCALIS. Note that there is a SAS/STAT procedure called PROC PLS, which employs the partial least squares technique but for a different class of models than those of PROC TCALIS. For general path analysis with latent variables, consider using PROC TCALIS.

All estimation methods need some starting values for the parameter estimates. You can provide starting values for any parameters. If there is any estimate without a starting value provided, PROC TCALIS determines the starting value by using one or any combination of the following methods:

- approximate factor analysis
- default initial values
- instrumental variable method
- matching observed moments of exogenous variables
- McDonald's (McDonald and Hartmann 1992) method
- ordinary least squares estimation
- random number generation, if a seed is provided
- two-stage least squares estimation

Although no methods for initial estimates are completely foolproof, the initial estimation methods provided by PROC TCALIS behave reasonably well in most common applications.

With initial estimates, PROC TCALIS will iterate the solutions so as to achieve the optimum solution as defined by the estimation criterion. This is a process known as optimization. Because numerical problems can occur in any optimization process, the TCALIS procedure offers several optimization algorithms so that you can choose alternative algorithms when the one being used fails. The following optimization algorithms are supported in PROC TCALIS:

- Levenberg-Marquardt algorithm (Moré, 1978)
- trust-region algorithm (Gay 1983)
- Newton-Raphson algorithm with line search
- ridge-stabilized Newton-Raphson algorithm
- various quasi-Newton and dual quasi-Newton algorithms: Broyden-Fletcher-Goldfarb-Shanno and Davidon-Fletcher-Powell, including a sequential quadratic programming algorithm for processing nonlinear equality and inequality constraints
- various conjugate gradient algorithms: automatic restart algorithm of Powell (1977), Fletcher-Reeves, Polak-Ribiere, and conjugate descent algorithm of Fletcher (1980)

In addition to the ability to save output tables as data sets by using the ODS OUTPUT statement, PROC TCALIS supports the following types of output data sets so that you can save your analysis results for later use:

- OUTEST= data sets for storing parameter estimates and their covariance estimates
- OUTFIT= data sets for storing fit indices and some pertinent modeling information
- OUTMODEL= data sets for storing model specifications and final estimates
- OUTSTAT= data sets for storing descriptive statistics, residuals, predicted moments, and latent variable scores regression coefficients
- OUTWGT= data sets for storing the weight matrices used in the modeling

The OUTEST=, OUTMODEL=, and OUTWGT= data sets can be used as input data sets for subsequent analyses. That is, in addition to the input data provided by the DATA= option, PROC TCALIS supports the following input data sets for various purposes in the analysis:

- INEST= data sets for providing initial parameter estimates. An INEST= data set could be an OUTEST= data set created from a previous analysis.
- INMODEL= data sets for providing model specifications and initial estimates. An INMODEL= data set could be an OUTMODEL= data set created from a previous analysis.
- INWGT= data sets for providing the weight matrices. An INWGT= data set could be an OUTWGT= data set created from a previous analysis.

The TCALIS procedure uses ODS Graphics to create graphs as part of its output. High-quality residual histograms are available in PROC TCALIS. See Chapter 21, "Statistical Graphics Using ODS," for general information about ODS Graphics. See the section "ODS Graphics" on page 6952 and the PLOTS= option on page 6741 for specific information about the statistical graphics available with the TCALIS procedure.

A Guide to the PROC TCALIS Documentation

The TCALIS procedure fits structural equation models using a variety of modeling languages. This chapter provides documentation for all of them. Additionally, some sections provide introductions to the model specification, the theory behind the software, and other technical details. While some introductory material and examples are provided, this chapter is not a textbook for structural equation modeling and related topics. For didactic treatment of structural equation models with latent variables, see the books by Bollen (1989b) and Loehlin (2004). Therefore, reading this chapter sequentially is not a good strategy for learning about PROC TCALIS. This section provides a guide or "road map" to the rest of the PROC TCALIS chapter starting with the basics and continuing

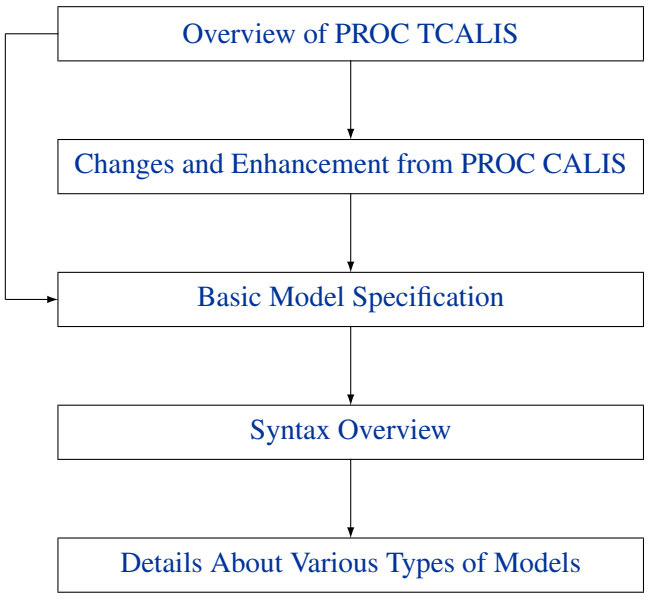
through more advanced topics. Many sections assume that you already have a basic understanding of structural equation modeling.

The following table shows three different skills levels of using the TCALIS procedure (basic, intermediate, and advanced) and their milestones:

Level	Milestone
Basic	You are able to specify simple models, but might make mistakes
Intermediate	You are able to specify more sophisticated models with few syntactic and semantic mistakes
Advanced	You are able to use the advanced options provided by PROC TCALIS

In the next three sections, each skills level is discussed, followed by an introductory section of the reference topics that are not covered in any of the skills levels.

Guide to the Basic Skills Level



Overview of PROC TCALIS

The section "Overview: TCALIS Procedure" on page 6681 gives you an overview of the PROC TCALIS procedure and an overall picture of the TCALIS procedure but without the details.

Changes and Enhancement from PROC CALIS

Read the section "Changes and Enhancement from PROC CALIS" on page 6695 if you have previous experience with PROC CALIS, the predecessor of PROC TCALIS. If you have not used PROC CALIS previously, you may skip this section.

Basic Model Specification

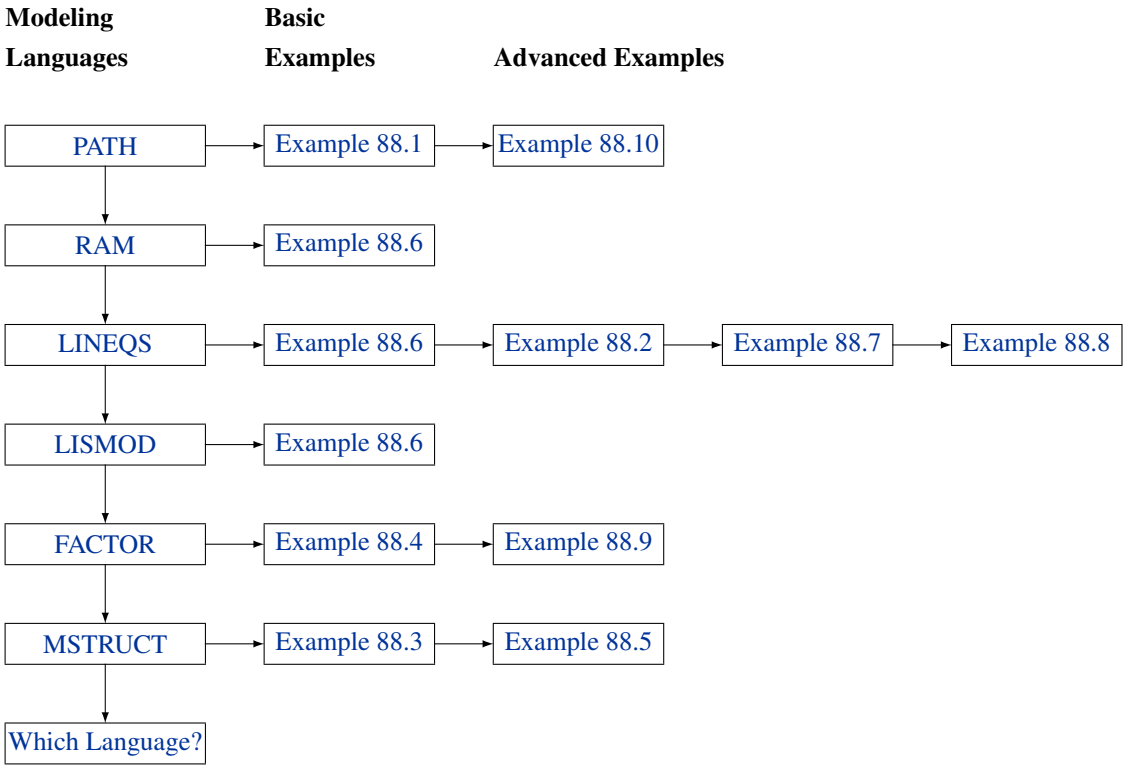
The structural equation example in the section "Getting Started: TCALIS Procedure" on page 6704 provides the best starting point for learning the basic model specification. You learn how to represent your theory by using a path diagram and then translate the diagram into the PATH model for PROC TCALIS to analyze. Because the PATH modeling language is new, this example is useful whether or not you have previous experience with PROC CALIS. The PATH model is specified in the section "PATH Model" on page 6706. The corresponding results are shown and discussed in Example 88.1.

After you learn about the PATH modeling language and an example of its application, you can either continue to learn more modeling languages or skip to the syntax overview. However, you do not need to learn all of the modeling languages in PROC TCALIS. Any one of the modeling languages (LINEQS, LISMOD, PATH, or RAM) is sufficient for specifying a very wide class of structural equation models. PROC TCALIS provides different kinds of modeling languages because different researchers might have previously learned different modeling languages or approaches.

You can go to the next step for an overview of the PROC TCALIS syntax and learn other modeling languages at a later time. See "Getting Started: TCALIS Procedure" on page 6704 for a brief overview of the various modeling languages. In these examples, model specifications are shown and explained, but no analysis results are discussed. Each subsection discusses the same model but uses a different modeling language. In the subsections "LISMOD Model" on page 6709, "LINEQS Model" on page 6708, and "RAM Model" on page 6707, you learn how to specify a structural equation model equivalently by using the LINEQS, LISMOD, and RAM modeling languages, respectively. In the subsection "A Factor Model Example" on page 6710, you learn how to specify a confirmatory factor model by using the FACTOR modeling language. In the subsection "Direct Covariance Structures Analysis" on page 6712, you learn how to specify the covariance structures directly on the covariance matrix by using the MSTRUCT modeling language.

After studying the examples in the "Getting Started: TCALIS Procedure" section, you can strengthen your understanding of the various modeling languages by studying more examples such as those in section "Examples: TCALIS Procedure" on page 6953. Unlike the examples in the "Getting Started: TCALIS Procedure" section, examples in the "Examples: TCALIS Procedure" section include the analysis results, in addition to the explanations of the model specifications. The following figure provides a guide to studying the examples in the "Examples: TCALIS Procedure" section.

Figure 88.1 A Guide for Studying the Examples



In this guide, the various modeling languages are listed in sequence. However, only the PATH modeling language is a prerequisite of other modeling languages. You are not required to learn the languages in the order presented in this guide. For example, you can skip the RAM modeling language and learn the LINEQS model specification after you finish the PATH model language. Similarly, you can learn the LISMOD language without going through the RAM or LINEQS modeling languages if you are familiar with the traditional Keesling-Wiley-Jöreskog measurement and structural models (Keesling 1972; Wiley 1973; Jöreskog 1973).

You can learn each modeling language by studying the examples in the "Getting Started: TCALIS Procedure" sections and in the "Examples: TCALIS Procedure" sections. Examples are listed for each modeling language with the basic examples listed before the more advanced examples. The basic examples are required, and the more advanced examples can be skimmed in the first reading. You might need to revisit the more advanced examples as your understanding grows. A summary of the examples is presented next.

Example 88.1. Path Analysis: Stability of Alienation

In this example, you learn how to specify a simple PATH model and interpret the basic estimation results. The results are shown in considerable detail. The output and analyses include: a model summary, initial model specification, initial estimation method, optimization history and results, residual analyses, residual graphics, estimation results, squared multiple correlations, and standardized results.

Example 88.2. Simultaneous Equations with Mean Structures and Reciprocal Paths

In this econometric example, you learn how to specify models using the LINEQS modeling language. This example also illustrates the specification of reciprocal effects, the simultaneous analysis of the mean and covariance structures, the setting of bounds for parameters, and the definitions of meta-parameters by using the PARAMETERS and SAS programming statements. You also learn how to shorten your output results by using some global display options such as the PSHORT and NOSTAND options in the PROC TCALIS statement.

Example 88.3. A Direct Covariance Structures Model

In this example, you fit your covariance structures directly on the covariance matrix by using the MSTRUCT modeling language. You also learn how to use the FITINDEX statement to create a customized model fit summary and how to save the fit summary statistics into an external file.

Example 88.4. Confirmatory Factor Analysis: Cognitive Abilities

Confirmatory factor analysis is illustrated in this example by using the FACTOR modeling language. In addition, you use the MODIFICATION option in the PROC TCALIS statement to compute LM test indices for model modifications.

Example 88.5. Testing Equality of Two Covariance Matrices Using A Multiple-Group Analysis

In this example, a simple multiple-group analysis is illustrated by the MSTRUCT modeling language. You also learn how to use the **ods select** statement to customize your printed output.

Example 88.6. Illustrating Various General Modeling Languages

This example extends Example 88.1, which uses the PATH modeling language, and shows how to use the other general modeling languages: RAM, LINEQS, and LISMOD. These modeling languages enable you to specify the same path model as in Example 88.1 and get equivalent results. This example shows the connections between the general modeling languages supported in PROC TCALIS. A good understanding of Example 88.1 is a prerequisite for this example.

Example 88.7. Fitting a Latent Growth Curve Model

This is an advanced example that illustrates the use of structural equation modeling techniques for fitting latent growth curve models. In this example, you learn how to specify random intercepts and random slopes by using the LINEQS modeling language. In addition to the modeling of the covariance structures, you also learn how to specify the mean structure parameters.

Example 88.8. Higher-Order and Hierarchical Factor Models

This is an advanced example for confirmatory factor analysis. It involves the specifications of higher-order and hierarchical factor models. Because higher-order factor models cannot be specified by the FACTOR modeling language, you need to use the LINEQS model specification instead.

A second-order factor model and a bifactor model are fit. Linear constraints on parameters are illustrated by using the PARAMETERS and the SAS programming statements. Relationships between the second-order factor model and the bifactor model are numerically illustrated.

Example 88.9. Linear Relations Among Factor Loadings

This is an advanced example of a first-order confirmatory factor analysis by using the FACTOR modeling language. In this example, you learn how to use the PARAMETERS and the SAS programming statements to set up dependent parameters in your model. You also learn how to specify the correlation structures for a specific confirmatory factor model.

Example 88.10. A Multiple-group Model for Purchasing Behavior

This is a sophisticated example of analyzing a path model. The PATH modeling language is used. In this example, a two-group analysis of mean and covariance structures is conducted. You learn how to use a SAS macro to organize your model specification, the REFMODEL statement to reference properly defined models, and the SIMTEST statement to test a priori simultaneous hypotheses.

Once you are familiar with various modeling languages, you might wonder which modeling language should be used in a given situation. The section "Which Modeling Language?" on page 6714 provides some guidelines and suggestions.

Syntax Overview

Now that you have some basic understanding of the modeling languages of PROC TCALIS, you can look at the global structure of PROC TCALIS specifications. The section "Syntax: TCALIS Procedure" on page 6715 shows the syntactic structure of PROC TCALIS. However, reading the "Syntax: TCALIS Procedure" section sequentially might not be a good strategy. The statements used in PROC TCALIS are classified in the section "Classes of Statements" on page 6716. Understanding this section is a prerequisite for understanding single-group and multiple-group analyses in PROC TCALIS. Syntax for single-group analyses is described in the section "Single-Group Analysis Syntax" on page 6719, and syntax for multiple-group analyses is described in the section "Multiple-Group Multiple-Model Analysis Syntax" on page 6720.

You might also want to get an overview of the options on the PROC TCALIS statement in the section "PROC TCALIS Statement" on page 6721. However, you can skip the detailed listing of the available options on the PROC TCALIS statement. Most of these details serve as references, so they are consulted only when needed. You can just read the summary tables for the available options on the PROC TCALIS statement in the following subsections:

- "Data Set Options" on page 6721
- "Model and Estimation Options" on page 6722
- "Options for Fit Statistics" on page 6722
- "Options for Statistical Analysis" on page 6723

- "Global Display Options" on page 6724
- "Optimization Options" on page 6725

Details About Various Types of Models

Several subsections in the section "Details: TCALIS Procedure" on page 6819 will help you gain a deeper understanding of the various types of modeling languages:

Language	Section
FACTOR:	"The FACTOR Model" on page 6863
LINEQS:	"The LINEQS Model" on page 6839
LISMOD:	"The LISMOD Model and Submodels" on page 6847
MSTRUCT:	"The MSTRUCT Model" on page 6877
PATH:	"The PATH Model" on page 6870
RAM:	"The RAM Model" on page 6855

The specification techniques you learn from the examples cover only parts of the modeling language. A more complete treatment of the modeling languages is covered in these subsections. In addition, you can also learn the mathematical models, model restrictions, and default parameterization of all supported modeling languages in these subsections.

Guide to the Intermediate Skills Level

At the intermediate level, you learn to minimize your mistakes in model specification and to establish more sophisticated modeling techniques. The following topics in the "Details: TCALIS Procedure" section or elsewhere can help:

- "Naming Variables and Parameters" on page 6880

  This section summarizes the naming rules and conventions for variable and parameter names in specifying models.

- "Setting Constraints on Parameters" on page 6882

  This section covers various techniques of constraining parameters in model specifications.

- "Automatic Variable Selection" on page 6887

  This section discusses how PROC TCALIS treats variables in the models and variables in the data sets. It also discusses situations where the specification of the VAR statement specification is deemed necessary.

- "Computational Problems" on page 6932

  This section discusses computational problems that occur quite commonly in structural equation modeling. It also discusses some possible remedies of the computational problem.

- "Missing Values" on page 6919

  This small section describes the default treatment of missing values.

- REFMODEL statement on page 6811 and RENAMEPARM statement on page 6812

  These statements are useful when you need to make references to well-defined models when specifying a "new" model. See Example 88.10 for an application.

Revisit topics and examples covered at the basic level, as needed, to help you better understand the topics at the intermediate level.

Guide to the Advanced Skills Level

At the advanced level, you learn to use the advanced data analysis and output control tools supported by PROC TCALIS.

Advanced Data Analysis Tools

The following advanced data analysis topics are discussed:

- Assessment of fit

  The section "Assessment of Fit" on page 6900 presents the fit indices used in PROC TCALIS. However, the more important topics covered in this section are about how model fit indices are organized and used, how residuals can be used to gauge the fitting of individual parts of the model, and how the coefficients of determination are defined for equations.

  To customize your fit summary table, you can use the options on the FITINDEX statement.

- Effect partitioning

  The section "Total, Direct, and Indirect Effects" on page 6913 discusses the total, direct, and indirect effects and their computations. The stability coefficient of reciprocal causation is also defined.

  To customize the effect analysis, you can use the EFFPART statement.

- Counting and adjusting degrees of freedom

  The section "Counting the Degrees of Freedom" on page 6897 describes how PROC TCALIS computes model fit degrees of freedom and how you can use some options on the PROC TCALIS statement to make degrees-of-freedom adjustments.

  To adjust the model fit degrees of freedom, you can use the DFREDUCE= and NOADJDF options in the PROC TCALIS statement.

- Standardized solutions

  Standardization schemes used in PROC TCALIS are described and discussed in the section "Standardized Solutions" on page 6915.

Standardized solutions are displayed by default. You can turn them off by using the NOSTAND option of the PROC TCALIS statement.

- Model modifications

  In the section "Modification Indices" on page 6917, modification indices such as Lagrange multiplier test indices and Wald statistics are defined and discussed. These indices can be used either to enhance your model fit or to make your model more precise.

  To limit the modification process only to those parameters of interest, you can use the LMTESTS statement to customize the sets of LM tests conducted on potential parameters.

- A Priori Parametric Function Testing

  You can use the TESTFUNC statement to test a priori hypotheses individually. You can use the SIMTEST statement to test a priori hypotheses simultaneously.

Advanced Output Control Tools

To be more effective in presenting your analysis results, you need to be more sophisticated in controlling your output. Some customization tools have been discussed in the previous section "Advanced Data Analysis Tools" on page 6692 and might have been mentioned in the examples included in the basic and the intermediate levels. In the following topics, these output control tools are presented in a more organized way so that you can have a systematic study scheme of these tools.

- Global output control tools in PROC TCALIS

  You can control output displays in PROC TCALIS either by the global display options or by the individual output printing options. Each global display option typically controls more than one output display, while each individual output display controls only one output display. The global display options can both enable and suppress output displays, and they can also alter the format of the output.

  See the ALL, PRINT, PSHORT, PSUMMARY, and NOPRINT options for ways to control the appearances of the output. See the section "Global Display Options" on page 6724 for details about the global display options and their relationships with the individual output display options. Also see the ORDERALL, ORDERGROUPS, ORDERMODELS, ORDERSPEC, PARMNAME, PRIMAT, NOORDERSPEC, NOPARMNAME, NOSTAND, and NOSE options which control the output formats.

- Customized analysis tools in PROC TCALIS

  Many individual output displays in PROC TCALIS can be customized via specific options or statements. If you do not use these customization tools, the default output will usually contain a large number of displays or displays with very large dimensions. These customized analysis tools are as follows:

  – The ON=, OFF=, ON(ONLY)= options in the FITINDEX statement enable you to select individual or groups of model fit indices or modeling information to display. You can still save the information of *all* fit indices in an external file by using the OUTFIT= option.

  - The EFFPART statement enables you to customize the effect analysis. You display only those effects of substantive interest.
  - The LMTESTS statement enables you to customize the sets of LM tests of interest. You test only those potential parameters that are theoretically and substantively possible.

- Output selection and destinations by the ODS system

  This kind of output control is used not only for PROC TCALIS, but is used for all procedures that support the ODS system. The most common uses include output selection and output destinations assignment. You use the ODS SELECT statement together with the ODS table names or graph names to select particular output displays. See the section "ODS Table Names" on page 6939 for these names in PROC TCALIS.

  The default output destination of PROC TCALIS is the listing destination. You can add or change the destinations by using statements such as **`ods html`** (for html output), **`ods rtf`** (for rich text output), and so on. For details, see Chapter 20, "Using the Output Delivery System."

Reference Topics

Some topics in the "Details: TCALIS Procedure" section are intended primarily for references—you consult them only when you encounter specific problems in the PROC TCALIS modeling or when you need to know the very fine technical details in certain special situations. Many of these reference topics in the "Details: TCALIS Procedure" section are not required for practical applications of structural equation modeling. The following technical topics are discussed:

- Measures of multivariate kurtosis and skewness

  This is covered in the section "Measures of Multivariate Kurtosis" on page 6919.

- Estimation criteria and the mathematical functions for estimation

  The section "Estimation Criteria" on page 6888 presents formulas for various estimation criteria. The relationships among these criteria are shown in the section "Relationships among Estimation Criteria" on page 6893. To optimize an estimation criterion, you usually need its gradient and Hessian functions. These functions are detailed in the section "Gradient, Hessian, Information Matrix, and Approximate Standard Errors" on page 6894, where you can also find information about the computation of the standard error estimates in PROC TCALIS.

- Initial estimation

  Initial estimates are necessary for all kinds of iterative optimization techniques. They are described in section "Initial Estimates" on page 6922.

- Use of optimization techniques

  Optimization techniques are covered in section "Use of Optimization Techniques" on page 6923. See this section if you need to fine-tune the optimization.

- Output displays and control

  The output displays in PROC TCALIS are listed in the section "Displayed Output" on page 6935. General requirements for the displays are also shown.

  With the ODS system, each table and graph has a name, which can be used on the ODS OUTPUT or ODS SELECT statement. See the section "ODS Table Names" on page 6939 for the ODS table and graph names.

- Input and output files

  PROC TCALIS supports several input and output data files for data, model information, weight matrices, estimates, fit indices, and estimation and descriptive statistics. The uses and the structures of these input and output data files are described in the sections "Input Data Sets" on page 6819 and "Output Data Sets" on page 6823.

Changes and Enhancement from PROC CALIS

PROC TCALIS was modified from PROC CALIS. PROC TCALIS is not a simple functional enhancement of PROC CALIS. The basic computational architecture of PROC TCALIS is quite different from that of PROC CALIS. Hence, some discrepancies between the two procedures are less apparent. There are also some temporary discrepancies due to issues that are still being resolved.

In this section, new functionalities in PROC TCALIS are described, followed by some lists for comparing the options and statements available in PROC CALIS and PROC TCALIS. Finally, ODS table names in PROC CALIS are listed with their corresponding new names in PROC TCALIS.

New Features

There are several notably new features in PROC TCALIS:

- new modeling languages
- multiple-group analysis
- improved mean structures analysis
- general parametric functions testing
- customizable fit summary table
- improved standardized results
- improved effects analysis
- customizable Lagrange multiplier tests

- more rotation options in exploratory factor analysis
- input order respecting
- improved OUTRAM= data set format

These new functionalities are outlined in the following sections.

New Modeling Languages in PROC TCALIS

To accommodate various modeling backgrounds and philosophies of researchers, more modeling languages are supported in PROC TCALIS. Three new modeling languages are provided: LISMOD, MSTRUCT, and PATH.

The LISMOD modeling language is a matrix-based parameter specification method modified from the LISREL model developed by Jöreskog and Sörbom. You can specify parameters as matrix entries by using the LISMOD language. For details, see the LISMOD statement on page 6772.

The MSTRUCT modeling language is also a matrix-based parameter specification method. You can specify parameters directly in the structured mean and covariance matrices in this language. For details, see the MSTRUCT statement on page 6792.

The PATH modeling language provides a tool to specify causal relations among variables by using paths (represented by arrows). It is especially suitable for path analysis, although it can also be applied to general structural models. For details, see the PATH statement on page 6799.

Although the FACTOR and RAM modeling languages are not new, their syntax has been changed for easier specifications. The FACTOR and RAM modeling languages are both matrix-based specification methods in PROC CALIS. When you specify parameters in your model, you must use row and column numbers of the model matrices to refer to the variables involved. However, in PROC TCALIS these matrix-based languages are replaced by the more intuitive approach. For confirmatory factor analysis in PROC TCALIS, you can specify the factors to variables paths (or loadings) by using the factor and variable names directly in the FACTOR statement. For the RAM specification in PROC TCALIS, you no longer need to implicitly assume a certain order for the variables and factors in matrices, as you do in PROC CALIS. Also, you no longer need to use the matrix numbers to refer to the types of parameter specifications. In each parameter specification of the RAM statement in PROC TCALIS, the variables and the parameter type involved are specified directly by using the variable names or meaningful keywords for parameter types.

To provide a quick and easy way to specify similar models, the REFMODEL statement is provided in PROC TCALIS. Using this statement, you can specify a new model by referring to another well-defined model. Supporting options PARM_PREFIX= and PARM_SUFFIX= and the RENAMEPARM statement enable you to change parameter names efficiently.

Multiple-Group Analysis

You can do multiple-group analysis in PROC TCALIS. Groups can also be fitted by multiple models simultaneously. You can use multiple GROUP statements to define independent groups of data.

Within the scope of each GROUP statement, you can set group-specific attributes and options for the associated group. See the GROUP statement on page 6764 for details.

You can use multiple MODEL statements to define models and the groups they fit. Within the scope of each MODEL statement, you specify your model by using one of the modeling languages provided by PROC TCALIS. You can use different modeling languages for different models. You can also set model-specific analysis and options for the model within the scope of a MODEL statement. See the MODEL statement on page 6789 for details.

Improved Mean Structures Analysis

In PROC CALIS, the mean structures are analyzed by means of augmented uncorrected moment matrices. This approach is an viable option only for maximum likelihood estimation. Often, this approach creates some interpretation problems in standardized results, R-square calculations, and so on. It is also difficult to set the mean parameters by using this approach.

In PROC TCALIS, the mean structures are analyzed directly as a term in the objective function being optimized. This method is applicable to all estimation methods and yields more interpretable results. You can use the MEANSTR option in the PROC TCALIS or MODEL statements to specify the analysis of mean structures explicitly. Alternatively, when you specify Intercept terms in LINEQS models, parameters in the MEAN statements, or parameters in the intercept or mean vector in the MATRIX statements, the mean structures of the model will be analyzed automatically.

As a result of the improved mean structures analysis, AUG, NOINT, UCOV, and UCORR options are obsolete in PROC TCALIS.

General Parametric Function Testing

You can test any differentiable parametric functions separately or simultaneously by using the TESTFUNC and the SIMTEST statements. A parametric function can be either a parameter in the model or a computed function defined by the SAS Programming statements. PROC TCALIS will analytically generate the necessary partial derivatives for computing the test statistics.

Customizable Fit Summary Table

In PROC TCALIS, you can customize the display of the fit summary table by selecting a subset of the fit indices to display. See the FITINDEX statement on page 6760 for details. You can also choose a particular type of chi-square correction for model fit chi-square statistics. A new OUTFIT= option enables you to store the fit indices in an external data set.

Improved Standardized Results

Standardized parameter estimates with standard errors are provided by default in PROC TCALIS. You can turn off the printing of standard error estimates by the NOSE option. To suppress the printing of the entire standardized results, you can use the NOSTAND option.

The standardized results in PROC TCALIS are somewhat different from that of PROC CALIS. In particular, in PROC TCALIS path coefficients attached to error terms will remain equal to 1 after standardization. The error variances are rescaled appropriately so as to maintain mathematical consistency. In contrast, after standardization PROC CALIS will make all error variances equal to 1 and the path coefficients attached to error terms will not be 1 in general. For interpretation, this is not desirable because error terms, by nature, should be a non-deterministic term added without modification (that is, multiplied by a path coefficient) to the deterministic terms in an equation. In this sense, the standardized method in PROC TCALIS is more interpretable.

Improved Effect Analysis

There are several improvements regarding the effects partitioning in PROC TCALIS. First, standardized effects are displayed in addition to unstandardized effects. Second, standard error estimates are provided for the standardized and unstandardized effects. Third, you can customize the effects analysis by using the EFFPART statement. This will enable you to display only those effects of interest. See the EFFPART statement on page 6751 statement for details.

Customizable Lagrange Multiplier Tests

In PROC TCALIS, you can set your own regions of the parameter space for the Lagrange multiplier (LM) tests. In the LMTESTS statement, you define sets of parameter regions. In each set, you include the regions of interest. In the output, LM statistics ranked within sets are displayed. The parameter that improves the model fit the most appears first. You can also set other display options in the LMTESTS statement.

More Rotation Options in Exploratory Factor Analysis

PROC TCALIS provides more orthogonal and oblique rotation options for exploratory factor analytic solutions. See the ROTATE= option in the FACTOR statement on page 6752 for details.

Input Order Respecting

When you use the LINEQS statement, PROC TCALIS will display equations in the order you specify in the input. The terms within each equation are also ordered the same way you specify them. Unfortunately, PROC CALIS does not have these properties. PROC CALIS might display equations and terms in a certain order that is not consistent with the input.

PROC TCALIS also respects the order of parameter specification in the following statements:

- COV
- MEAN
- PATH

- PCOV
- PVAR
- RAM
- STD

If you want to order the specification by parameter types, you can use the ORDERSPEC option.

PROC TCALIS also respects order when displaying model and group results. By default, the output results for models or groups follow the order of your input. By using the ORDERMODELS and ORDERGROUPS options, the output results for models or groups are ordered by the model or group numbers provided in the specification. The ORDERALL option combines all these ordering options.

Improved OUTRAM= Data Set Format

The OUTRAM= data sets in PROC CALIS stores the model specifications in terms of the RAM model matrix entries, even if the original model is specified by the LINEQS or FACTOR modeling language. The problem is that the modeler who did not write the original code in the RAM modeling language might not understand the contents of the OUTRAM= data set. This inconsistency is eliminated in PROC TCALIS by means of the new OUTMODEL= option (although you can still use the OUTRAM= option for the same purpose). In the OUTMODEL= data sets, different types of models would have different types of observations. The types of observations resemble closely the original modeling language used. See the OUTMODEL= option and the section "OUTMODEL= SAS-data-set" on page 6827 for more details.

Inactive Statements and Options

Inactive statements and options are those available in PROC CALIS, but not available in PROC TCALIS.

The COSAN statement is currently inactive in PROC TCALIS. COSAN model specification might be available in future releases of PROC TCALIS.

The following options are currently inactive in PROC TCALIS, but might be added back in future releases:

Option	Description	TCALIS Treatment
FDCODE	uses the finite difference method for partial derivatives	uses analytic differentiations by default
HESSALG	specifies the algorithm for computing the Hessian	
NODIAG	suppresses the fitting of the diagonal elements in the covariance or correlation matrices	
OUTJAC	outputs the Jacobian pattern in an external file	
PJACPAT	prints the Jacobian pattern	

The following options are inactive in PROC TCALIS, and there is no plan to put these options back:

Option	Description	TCALIS Treatment
AUGMENT	uses the augmented moment matrices with mean structure analysis	specifies the mean structures in models directly
NOINT	uses the uncorrected covariance or correlation matrices for analyses	eliminates this option
PREDET	displays the predetermined elements in moment matrices	eliminates this display
PVEC	displays the final estimates, std errors, t-values, and gradients in separate vector forms	uses the PRINT or PALL option to display these results in a single table
UCORR	analyzes the uncorrected correlation matrix	eliminates this option
UCOV	analyzes the uncorrected covariance matrix	eliminates this option

Note that even though the UCORR and UCOV options are not supported in PROC TCALIS, you can still provide SAS data sets with TYPE=UCORR or TYPE=UCOV in the DATA= option. PROC TCALIS will utilize the information provided in the data set to compute the corrected covariance or correlation matrix for analysis.

New Statements and Options

The following are new statements in the TCALIS procedure:

Statement	Description
DETERM on page 6750	sets groups of variables for computing the determination coefficients
EFFPART on page 6729	displays and partitions the effects in the model
FITINDEX on page 6760	controls the fit summary output
GROUP on page 6764	defines groups and controls the corresponding data processing
LISMOD on page 6772	specifies a LISREL model
LMTESTS on page 6776	defines the Lagrange multiplier test regions
MEAN on page 6788	specifies the mean parameters
MODEL on page 6789	defines models and controls the corresponding modeling and analysis options
MSTRUCT on page 6792	specifies the direct covariance and mean model
OUTFILES on page 6796	controls multiple output data sets
PATH on page 6799	specifies the PATH model
PCOV on page 6802	specifies the covariance and partial covariance parameters
PVAR on page 6803	specifies the variance and partial variance parameters
REFMODEL on page 6811	specifies the base model for references
RENAMEPARM on page 6812	renames parameters
SIMTEST on page 6814	defines simultaneous parametric function tests
TESTFUNC on page 6817	tests individual parametric functions

The following are new options available in the PROC TCALIS statement:

Option	Description
CHICORRECT= on page 6727	specifies the chi-square correction factor
INMODEL= on page 6730	specifies the data set that contains the input model information
INWGTINV on page 6731	specifies that the INWGT= data set contains the inverse of the weight matrix
MEANSTR on page 6733	analyzes the mean structures
NOORDERSPEC on page 6735	displays the parameter specifications and results according to the input order
NOPARMNAME on page 6735	suppresses the display of parameter names
NOSTAND on page 6735	suppresses the display of standardized results
ORDERALL on page 6738	displays the specifications and results by the group numbers, model numbers, and parameter types
ORDERGROUPS on page 6738	displays the specifications and results for groups by the group numbers
ORDERMODELS on page 6738	displays the specifications and results for models by the model numbers
ORDERSPEC on page 6738	displays the specifications and results for parameters by the parameter types
OUTFIT= on page 6739	specifies the output data set for storing the fit statistics
OUTMODEL= on page 6739	specifies the output data set for storing the model specification and results
PARMNAME on page 6740	displays the parameter names in output

Changes in ODS Table Names

PROC CALIS assigns an ODS table name to each output it creates. Not all of these ODS table names are preserved in PROC TCALIS. In the following table, ODS table names for the CALIS procedure are listed together with the corresponding new ODS table name(s) in PROC TCALIS. Some of these ODS table names have been changed in PROC TCALIS, while others are unchanged or deleted.

CALIS ODS Table Name	TCALIS Table Name(s)
AddParms	(unchanged)
AsymStdRes	(unchanged)
AveAsymStdRes	(unchanged)
AveNormRes	(unchanged)
AveRawRes	(unchanged)
AveVarStdRes	(unchanged)
ContKurtosis	(unchanged)
ConvergenceStatus	(unchanged)
CorrExog	LINEQSCovExogStd
CorrParm	(no longer available)
CovMat	(no longer available)

CALIS ODS Table Name	TCALIS Table Name(s)
DependParms	DependParmsStart or DependParmsResults
Determination	(unchanged)
DistAsymStdRes	(unchanged)
DistNormRes	(unchanged)
DistVarStdRes	(unchanged)
EndogenousVar	LINEQSVariables or RAMVariables
EstCovExog	LINEQSCovExog
Estimates	(no longer available)
EstLatentEq	LINEQSEq
EstManifestEq	LINEQSEq
EstParms	LINEQSBeta, LINEQSBetaStd, LINEQSGamma, LINEQSGammaStd, LINEQSPhi, LINEQSPhiStd, FACTCov, FACTErrVar, FACTErrVarStd, FACTLoadings, FACTRotCov, FACTRotErrVar, or FACTRotErrVarStd
EstVarExog	LINEQSVarExog
ExogenousVar	LINEQSVariables
FactCorrExog	FACTCovStd or FACTRotCovStd
FactScoresCoef	FACTScoresRegCoef
Fit	(unchanged)
GenModInfo	(no longer available)
Gradient	ParameterEstimatesResults
InCorr	(unchanged)
InCorrDet	(unchanged)
InCov	(unchanged)
InCovDet	(unchanged)
InCovExog	LINEQSCovExogInit
Indirect Effects	(unchanged)
Information	(unchanged)
InitEstimates	(no longer available)
InitParms	LINEQSBetaInit, LINEQSGammaInit, LINEQSPhiInit, FACTCovInit, FACTErrVarInit, or FACTLoadingsInit
InitRAMEstimates	(no longer available)
InLatentEq	LINEQSEqInit
InManifestEq	LINEQSEqInit
InSymmetric	(no longer available)
InVarExog	LINEQSVarExogInit
IterHist	(unchanged)
IterStart	(unchanged)
IterStop	(unchanged)
Jacobian	(no longer available)
Kurtosis	(unchanged)
LagrangeBoundary	(unchanged)
LagrangeEquality	(unchanged)
LatentScoreCoef	LatentScoresRegCoef
ModelStatement	(no longer available)

CALIS ODS Table Name	TCALIS Table Name(s)
ModIndices	(replaced by various ODS tables with the LM prefix and matrix name suffixes)
NormRes	(unchanged)
PredetElements	(no longer available)
PredModel	PredCorr, or PredCov
PredModelDet	PredCorrDet, or PredCovDet
PredMomentLatent	PredCovLatent
PredMomentManLat	PredCovLatMan
ProblemDescription	(unchanged)
RAMCorrExog	RAMListStd
RAMEstimates	RAMList
RAMStdEstimates	RAMListStd
RankAsymStdRes	(unchanged)
RankLagrange	(replaced by various ODS tables with the LMRank prefix and matrix name suffixes)
RankNormRes	(unchanged)
RankRawRes	(unchanged)
RankVarStdRes	(unchanged)
RawRes	(unchanged)
RotatedLoadings	FACTRotLoadings
Rotation	FACTRotMat
SetCovExog	(no longer available)
SimpleStatistics	(unchanged)
SqMultCorr	(unchanged)
Stability	(unchanged)
StdErrs	ParameterEstimatesResults
StdLatentEq	LINEQSEqStd
StdLoadings	FACTLoadingsStd, or FACTRotLoadingsStd
StdManifestEq	LINEQSEqStd
StructEq	(no longer available)
SumSqDif	(no longer available)
TotalEffects	(unchanged)
tValues	ParameterEstimatesResults
VarSelection	(no longer available)
VarStdRes	(unchanged)
WaldTest	(unchanged)
Weights	(unchanged)
WeightsDet	(unchanged)

Getting Started: TCALIS Procedure

A Structural Equation Example

This example from Wheaton et al. (1977) illustrates the basic uses of the TCALIS procedure and the relationships among the LINEQS, LISMOD, PATH, and RAM modeling languages. Different structural models for these data are analyzed in Jöreskog and Sörbom (1985) and in Bentler (1985, p. 28). The data contain the following six (manifest) variables collected from 932 people in rural regions of Illinois:

Anomie67:	Anomie 1967
Powerless67:	Powerlessness 1967
Anomie71:	Anomie 1971
Powerless71:	Powerlessness 1971
Education:	Education level (years of schooling)
SEI:	Duncan's socioeconomic index (SEI)

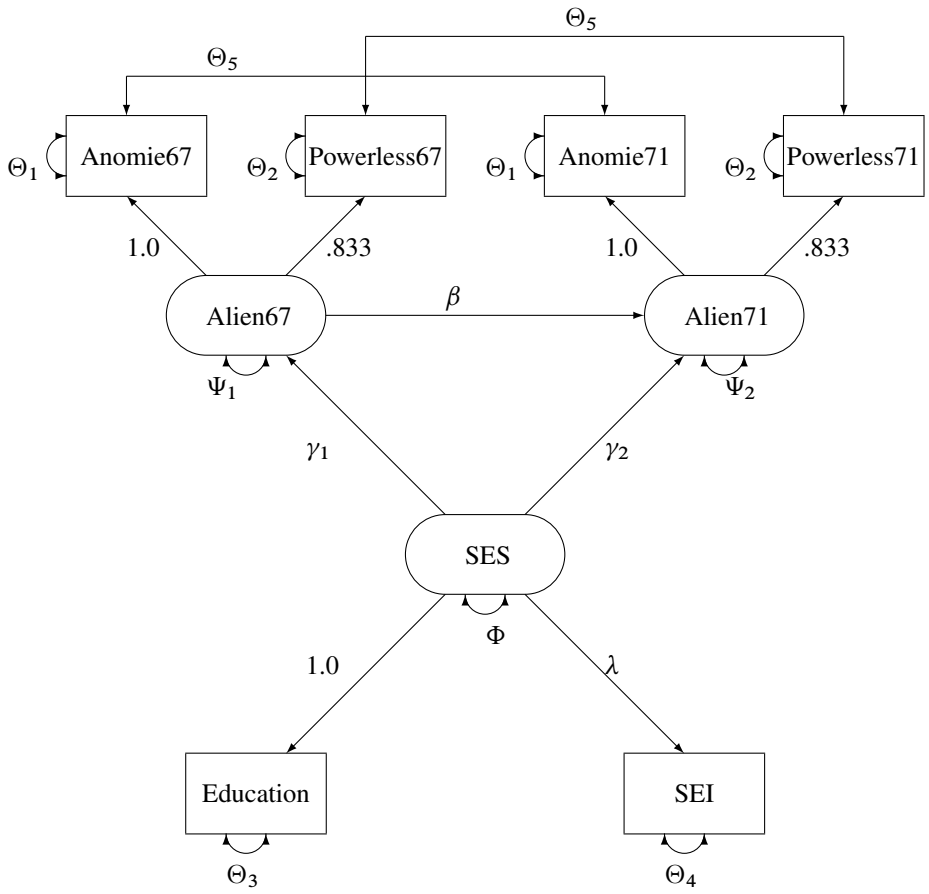
The covariance matrix of the these six variables is stored in the data set named Wheaton.

It is assumed that anomie and powerlessness are indicators of an alienation factor and that education and SEI are indicators for a socioeconomic status (SES) factor. Hence, the analysis contains three latent variables (factors):

Alien67:	Alienation 1967
Alien71:	Alienation 1971
SES:	Socioeconomic status (SES)

The following path diagram shows the structural model used in Bentler (1985, p. 29) and slightly modified in Jöreskog and Sörbom (1985, p. 56):

Figure 88.2 Path Diagram of Stability and Alienation Example



In the path diagram shown in Figure 88.2, regressions of variables are represented by one-headed arrows. Regression coefficients are indicated along these one-headed arrows. Variances and covariances among the variables are represented by two-headed arrows. Error variances and covariances are also represented by two-headed arrows. This scheme of representing paths, variances and covariances, and error variances and covariances (McArdle 1988; McDonald 1985) is helpful in translating the path diagram to the PATH or RAM model input in the TCALIS procedure.

PATH Model

Specification by using the PATH modeling language is direct and intuitive in PROC TCALIS once a path diagram is drawn. The following statements specify the path diagram almost intuitively:

```
proc tcalis nobs=932 data=Wheaton;
   path
      Anomie67     <-  Alien67   1.0,
      Powerless67  <-  Alien67   0.833,
      Anomie71     <-  Alien71   1.0,
      Powerless71  <-  Alien71   0.833,
      Education    <-  SES       1.0,
      SEI          <-  SES       lambda,
      Alien67      <-  SES       gamma1,
      Alien71      <-  SES       gamma2,
      Alien71      <-  Alien67   beta;
   pvar
      Anomie67     = theta1,
      Powerless67  = theta2,
      Anomie71     = theta1,
      Powerless71  = theta2,
      Education    = theta3,
      SEI          = theta4,
      Alien67      = psi1,
      Alien71      = psi2,
      SES          = phi;
   pcov
      Anomie67    Anomie71    = theta5,
      Powerless67 Powerless71 = theta5;
run;
```

In the PROC TCALIS statement, you specify Wheaton as the input data set, which contains the covariance matrix of the variables.

In the PATH model specification, all the one-headed arrows in the path diagram are represented as path entries in the PATH statement, with entries separated by commas. In each path entry, you specify a pair of variables and the direction of the path (either <– or –>), followed by a path coefficient, which is either a fixed constant or a parameter with a name in the specification.

All the two-headed arrows each with the same source and destination are represented as entries in the PVAR statement, with entries separated by commas. In the PVAR statement, you specify the variance or error (or partial) variance parameters. In each entry, you specify a variable and then a parameter name or a fixed parameter value. If the variable involved is exogenous in the model (serves only as a predictor; never being pointed at by one-headed arrows), you are specifying a variance parameter for an exogenous variable in the PVAR statement. Otherwise, you are specifying an error variance (or a partial variance) parameter for an endogenous variable.

All other two-headed arrows are represented as entries in the PCOV statement, with entries separated by commas. In the PCOV statement, you specify the covariance or error (or partial) covariance parameters. In each entry, you specify a pair of variables and then a parameter name or a fixed parameter value. If both variables involved in an entry are exogenous, you are specifying a covariance parameter. If both variables involved in an entry are endogenous, you are specifying an error (or partial) covariance parameter. When one variable is exogenous and the other is endogenous in an

entry, you are specifying a partial covariance parameter that can be interpreted as the covariance between the exogenous variable and the error of the endogenous variable.

See Example 88.1 for the results of the current PATH model analysis. For more information about the PATH modeling language, see the section "The PATH Model" on page 6870 and the PATH statement on page 6799.

RAM Model

The PATH modeling language is not the only specification method that you can use to represent the path diagram. You can also use the RAM, LINEQS or LISMOD modeling language to represent the diagram equivalently.

The RAM model specification in PROC TCALIS resembles that of the PATH model, as shown in the following statements:

```
proc tcalis nobs=932 data=Wheaton;
   ram
      path  Anomie67     <-  Alien67   1.0,
      path  Powerless67  <-  Alien67   0.833,
      path  Anomie71     <-  Alien71   1.0,
      path  Powerless71  <-  Alien71   0.833,
      path  Education    <-  SES       1.0,
      path  SEI          <-  SES       lambda,
      path  Alien67      <-  SES       gamma1,
      path  Alien71      <-  SES       gamma2,
      path  Alien71      <-  Alien67   beta,
      pvar  Anomie67                   theta1,
      pvar  Powerless67                theta2,
      pvar  Anomie71                   theta1,
      pvar  Powerless71                theta2,
      pvar  Education                  theta3,
      pvar  SEI                        theta4,
      pvar  Alien67                    psi1,
      pvar  Alien71                    psi2,
      pvar  SES                        phi,
      pcov  Anomie67    Anomie71       theta5,
      pcov  Powerless67 Powerless71    theta5;
run;
```

In the RAM statement, you specify a list of entries for parameters, with entries separated by commas. In each entry, you specify the type of parameter (PATH, PVAR, or PCOV in the code), the associated variable or pair of variables and the path direction if applicable, and then a parameter name or a fixed parameter value. The types of parameters you specify in this RAM model are for path coefficients, variances or partial variances, and covariances or partial covariances. They bear the same meanings as those in the PATH model specified previously. The RAM model specification is therefore quite similar to the PATH model specification—except that in the RAM model you put all parameter specification in the same list under the RAM statement, whereas you specify different types of parameters separately under different statements in the PATH model.

See Example 88.6 for partial results of the current RAM model analysis. For more information about the RAM modeling language, see the section "The RAM Model" on page 6855 and the RAM statement on page 6805.

LINEQS Model

The LINEQS modeling language uses equations to specify functional relationships among variables, as shown in the following statements:

```
proc tcalis nobs=932 data=Wheaton;
   lineqs
      Anomie67     = 1.0    f_Alien67 + E1,
      Powerless67  = 0.833  f_Alien67 + E2,
      Anomie71     = 1.0    f_Alien71 + E3,
      Powerless71  = 0.833  f_Alien71 + E4,
      Education    = 1.0    f_SES     + E5,
      SEI          = lambda f_SES     + E6,
      f_Alien67    = gamma1 f_SES     + D1,
      f_Alien71    = gamma2 f_SES     + beta f_Alien67 + D2;
   std
      E1           = theta1,
      E2           = theta2,
      E3           = theta1,
      E4           = theta2,
      E5           = theta3,
      E6           = theta4,
      D1           = psi1,
      D2           = psi2,
      f_SES        = phi;
   cov
      E1  E3       = theta5,
      E2  E4       = theta5;
run;
```

In the LINEQS statement, equations are separated by commas. In each equation, you specify an endogenous variable on the left-hand side, and then predictors and path coefficients on the right-hand side of the equal side. The set of equations specified in this LINEQS model is equivalent to the system of paths specified in the preceding PATH (or RAM) model. However, there are some notable differences between the LINEQS and the PATH specifications.

First, in the LINEQS modeling language you must specify the error terms explicitly as exogenous variables. For example, E1, E2, and D1 are error terms in the specification. In the PATH (or RAM) modeling language, you do not need to specify error terms explicitly.

Second, equations specified in the LINEQS modeling language are oriented by the endogenous variables. Each endogenous variable can appear on the left-hand side of an equation only **once** in the LINEQS statement. All the corresponding predictor variables must then be specified on the right-hand side of the equation. For example, f_Alien71 is predicted from f_Alien67 and f_SES in the last equation of the LINEQS statement. In the PATH or RAM modeling language, however, you would specify the same functional relationships in two separate paths.

Third, you must follow some naming conventions for latent variables when using the LINEQS modeling language. The names of latent variables that are not errors or disturbances must start with an ‘f’ or ‘F’. Also, the names of the error variables must start with ‘e’ or ‘E’ and the names of the disturbance variables must start with ‘d’ or ‘D’. For example, variables Alien67, Alien71, and SES serve as latent factors in the previous PATH or RAM model specification. To comply

with the naming conventions, these variables are named with an extra prefix 'f_' in the LINEQS model specification—that is, f_Alien67, f_Alien71, and f_SES, respectively. In addition, because of the naming conventions of the LINEQS modeling language, E1–E6 serve as error terms and D1–D1 serve as disturbances in the specification.

A consequence of explicit specification of error terms in the LINEQS statement is that the partial variance and partial covariance concepts used in the PATH and RAM modeling languages are no longer needed. They are replaced by the variances or covariances of the error terms or disturbances. Errors and disturbances are exogenous variables by nature. Hence, in terms of variance and covariance specification, they are treated exactly the same way as other non-error exogenous variables in the LINEQS modeling language. That is, variance parameters for all exogenous variables, including errors and disturbances, are specified in the STD statement, and covariance parameters among exogenous variables, including errors and disturbances, are specified in COV statement.

See Example 88.6 for partial results of the current LINEQS model analysis. For more information about the LINEQS modeling language, see the section "The LINEQS Model" on page 6839 and the LINEQS statement on page 6766.

LISMOD Model

The LISMOD language is quite different from the LINEQS, PATH, and RAM modeling languages. In the LISMOD specification, you define parameters as entries in model matrices, as shown in the following statements:

```
proc tcalis nobs=932 data=Wheaton;
   lismod
      yvar = Anomie67 Powerless67 Anomie71 Powerless71,
      xvar = Education SEI,
      etav = Alien67 Alien71,
      xiv  = SES;
   matrix _LAMBDAY_ [1,1] = 1.0,
                    [2,1] = 0.833,
                    [3,2] = 1.0,
                    [4,2] = 0.833;
   matrix _LAMBDAX_ [1,1] = 1.0,
                    [2,1] = lambda;
   matrix _GAMMA_   [1,1] = gamma1,
                    [2,1] = gamma2;
   matrix _BETA_    [2,1] = beta;
   matrix _THETAY_  [1,1] = theta1,
                    [2,2] = theta2,
                    [3,3] = theta1,
                    [4,4] = theta2,
                    [3,1] = theta5,
                    [4,2] = theta5;
   matrix _THETAX_  [1,1] = theta3,
                    [2,2] = theta4;
   matrix _PSI_     [1,1] = psi1,
                    [2,2] = psi2;
   matrix _PHI_     [1,1] = phi;
run;
```

In the LISMOD statement, you specify the lists of variables in the model. In the MATRIX statements, you specify the parameters in the LISMOD model matrices. Each MATRIX statement contains the matrix name of interest and then locations of the parameters, followed by the parameter names or fixed parameter values. It would be difficult to explain the LISMOD specification here without better knowledge about the formulation of the mathematical model. For this purpose, see the section "The LISMOD Model and Submodels" on page 6847 and the LISMOD statement on page 6772. See also Example 88.6 for partial results of the current LISMOD model analysis.

A Factor Model Example

In addition to the general modeling languages such as PATH, RAM, LINEQS, and LISMOD, the TCALIS procedure provides a specialized language for factor analysis. In the FACTOR modeling language, you can specify either exploratory or confirmatory factor models. For exploratory factor models, you can specify the number of factors, factor extraction method, and rotation algorithm, among many other options. For confirmatory factor models, you can specify the variable-factor relationships, factor variances and covariances, and the error variances.

For example, the following is an exploratory factor model fitted to the Wheaton et al. (1977) data by using PROC TCALIS:

```
proc tcalis nobs=932 data=Wheaton;
   factor n=2 rotate=varimax;
run;
```

In this model, you want to get the varimax-rotated solution with two factors. By default, the factor extraction method is maximum likelihood (METHOD=ML). Maximum likelihood exploratory factor analysis by PROC TCALIS can also be done equivalently by the FACTOR procedure, as shown in the following statements for the Wheaton et al. (1977) data:

```
proc factor nobs=932 data=Wheaton n=2 rotate=varimax method=ml;
run;
```

Note that METHOD=ML is necessary because maximum likelihood is not the default method in PROC FACTOR.

Whereas you can use either the TCALIS or FACTOR procedure to fit certain exploratory factor models, you can only use the TCALIS procedure to fit confirmatory factor models. In a confirmatory factor model, you are assumed to have some prior knowledge about the variable-factor relations. For example, in your substantive theory, some observed variables are not related to certain factors in the model. The following statements illustrate the specification of a confirmatory factor model for Wheaton et al. (1977) data:

```
proc tcalis nobs=932 data=Wheaton;
   factor
      Alien67 -> Anomie67 Powerless67    = 1.0 load1,
      Alien71 -> Anomie71 Powerless71    = 1.0 load2,
      SES     -> Education SEI           = 1.0 load3;
   pvar
      Alien67      = phi11,
      Alien71      = phi22,
      SES          = phi33,
      Anomie67     = theta1,
      Powerless67  = theta2,
      Anomie71     = theta3,
      Powerless71  = theta4,
      Education    = theta5,
      SEI          = theta6;
   cov
      Alien71 Alien67 = phi21,
      SES     Alien67 = phi31,
      SES     Alien71 = phi32;
run;
```

Unlike the model fitted by the PATH, RAM, LINEQS, or LISMOD modeling language in previous sections, the confirmatory factor model considered here is purely a measurement model—that is, there are no functional relationships among factors in the model (beyond the covariances among factors) and hence it is a different model. In the FACTOR statement, you specify factors on the left-hand side of the entries, followed by arrows and the manifest variables that are related to the factors. On the right-hand side of the entries, you specify either parameter names or fixed parameter values for the corresponding factor loadings. In this example, there are three factors with three loadings to estimate. In the PVAR statement, you specify the parameters for factor variances and error variances of manifest variables. In the COV statement, you specify the factor covariances. As compared with the PATH, RAM, LINEQS, or LISMOD, the factor modeling language has more restrictions on parameters. These restrictions are listed as follows:

- factor-factor paths and variable-to-factor paths are not allowed
- error covariances and factor-error covariances are not allowed

For more information about exploratory and confirmatory factor models and the FACTOR modeling language, see the section "The FACTOR Model" on page 6863 or the FACTOR statement on page 6752.

Direct Covariance Structures Analysis

Previous examples are concerned with the implied covariance structures from the functional relationships among manifest and latent variables. In some cases, direct modeling of the covariance structures is not only possible, but indeed more convenient. The MSTRUCT modeling language in PROC TCALIS is designed for this purpose. Consider the following four variables from the Wheaton et al. (1977) data set:

Anomie67:	Anomie 1967
Powerless67:	Powerlessness 1967
Anomie71:	Anomie 1971
Powerless71:	Powerlessness 1971

The covariance structures are hypothesized as follows:

$$\Sigma = \begin{pmatrix} \phi_1 & \theta_1 & \theta_2 & \theta_1 \\ \theta_1 & \phi_2 & \theta_1 & \theta_3 \\ \theta_2 & \theta_1 & \phi_1 & \theta_1 \\ \theta_1 & \theta_3 & \theta_1 & \phi_2 \end{pmatrix}$$

where:

ϕ_1:	Variance of Anomie
ϕ_2:	Variance of Powerlessness
θ_1:	Covariance between Anomie and Powerlessness
θ_2:	Covariance between Anomie measures
θ_3:	Covariance between Powerlessness measures

In the hypothesized covariance structures, the variances of Anomie and Powerlessness measures are assumed to stay constant over the two time points. Their covariances are also independent of the time of measurements. To test the tenability of this covariance structure model, you can use the following statements of the MSTRUCT modeling language:

```
proc tcalis nobs=932 data=Wheaton;
   mstruct
      var = Anomie67 Powerless67 Anomie71 Powerless71;
   matrix _COV_ [1,1] = phi1,
                [2,2] = phi2,
                [3,3] = phi1,
                [4,4] = phi2,
                [2,1] = theta1,
                [3,1] = theta2,
                [3,2] = theta1,
                [4,1] = theta1,
                [4,2] = theta3,
                [4,3] = theta1;
run;
```

In the MSTRUCT statement, you specify the list of variables of interest with the VAR= option. The order of the variables in the list will be the order in the hypothesized covariance matrix. Next, you use the MATRIX _COV_ statement to specify the parameters in the covariance matrix. The specification is a direct translation from the hypothesized covariance matrix. For example, the [1,1] element of the covariance matrix is fitted by the free parameter phi1. Depending on the hypothesized model, you can also specify fixed constants for the elements in the covariance matrix. If an element in the covariance matrix is not specified by either a parameter name or a constant, it is assumed to be a fixed zero.

The analysis of this model is carried out in Example 88.3.

The MSTRUCT modeling language appears to be more restrictive than any of the other modeling languages discussed, in regard to the following limitations:

- It does not explicitly support latent variables in modeling.
- It does not explicitly support modeling of linear functional relations among variables (for example, paths).

However, these limitations are more apparent than real. In PROC TCALIS, the parameters defined in models can be dependent. These dependent parameters can be defined further as functions of other parameters in the PARAMETERS and the SAS programming statements. With these capabilities, it is possible to fit structural models with latent variables and with linear functional relations by using the MSTRUCT modeling language. However, this requires a certain level of sophistication in statistical knowledge and in programming. Therefore, it is recommended that the MSTRUCT modeling language be used only when the covariance and mean structures are modeled directly.

For more information about the MSTRUCT modeling language, see the section "The MSTRUCT Model" on page 6877 and the MSTRUCT statement on page 6792.

Which Modeling Language?

Various modeling languages are supported in PROC TCALIS because researchers are trained in or adhere to different schools of modeling. Different modeling languages reflect different modeling terminology and philosophies. The statistical and mathematical consequences by using these various modeling languages, however, might indeed be the same. In other words, you can use more than one modeling languages for certain types of models without affecting the statistical analysis. Given the choices, which modeling language is preferred? There are two guidelines for this:

- Use the modeling language that you are most familiar with.
- Use the most specialized modeling language whenever it is possible.

The first guideline calls for researchers' knowledge about a particular modeling language. Use the language you know the best. For example, some researchers might find equation input language like LINEQS the most suitable, while others might feel more comfortable using matrix input language like LISMOD.

The second guideline depends on the nature of the model at hand. For example, to specify a factor-analysis model in the TCALIS procedure, the specialized FACTOR language, instead of the LISMOD language, is recommended. Using a more specialized the modeling language is less error-prone. In addition, using a specialized language like FACTOR in this case amounts to giving the TCALIS procedure additional information about the specific mathematical properties of the model. This additional information is used to enhance computational efficiency and to provide more specialized results. Another example is fitting an equi-covariance model. You can simply use the MSTRUCT model specification, in which you specify the same parameter for all off-diagonal elements of the covariance elements. This is direct and intuitive. Alternatively, you could tweak a LINEQS model that would predict the same covariance for all variables. However, this is indirect and error-prone, especially for novice modelers.

In PROC TCALIS, the FACTOR and MSTRUCT modeling languages are considered more specialized, while other languages are more general in applications. Whenever possible, you should use the more specialized languages.

Syntax: TCALIS Procedure

PROC TCALIS < *options* > ;
- **BASEMODEL** *model number* < / *options* > ;
- **BOUNDS** *boundary constraints* ;
- **BY** *variables* ;
- **COV** *covariance parameters* ;
- **DETERM** *variables* < *label* > ;
- **EFFPART** *effects* ;
- **FACTOR** < *factor options* > ;
- **FITINDEX** < *options* > ;
- **FREQ** *variable* ;
- **GROUP** *group number* < / *group options* > ;
- **LINCON** *linear constraints* ;
- **LINEQS** *model equations* ;
- **LISMOD** *variable lists* ;
- **LMTESTS** < *options* > ;
- **MATRIX** *matrix-name parameters-in-matrix* ;
- **MEAN** *mean parameters* ;
- **MODEL** *model number* < / *model options* > ;
- **MSTRUCT** *variable list* ;
- **NLINCON** *nonlinear constraints* ;
- **NLOPTIONS** *optimization options* ;
- **OUTFILES** *output files organization* ;
- **PARAMETERS** *parameters* ;
- **PARTIAL** *variables* ;
- **PATH** *path list* ;
- **PCOV** *partial covariance parameters* ;
- **PVAR** *partial variance parameters* ;
- **RAM** *ram list* ;
- **REFMODEL** *model number* < / *options* > ;
- **RENAMEPARM** *parameter renaming* ;
- **SIMTEST** *simultaneous tests definitions* ;
- **STD** *variance parameters* ;
- **STRUCTEQ** *set of variables* < *label* > ;
- **TESTFUNC** *parametric functions* ;
- **VAR** *variables* ;
- **WEIGHT** *variable* ;
- **SAS Programming statements** ;

Classes of Statements

To better understand the syntax of PROC TCALIS, it is useful to classify the statements into classes. These classes of statements are described in the following sections.

PROC TCALIS Statement

is the main statement that invokes the TCALIS procedure. You can specify options for input and output data sets, printing, statistical analysis, and computations in this statement. The options specified in the PROC TCALIS statement will propagate to all groups and models, but are superseded by the options specified in the individual GROUP or MODEL statements.

GROUP Statement

signifies the beginning of a group specification. A group in the TCALIS procedure is an independent sample of observations. You can specify options for input and output data sets, printing, and statistical computations in this statement. Some of these group options in the GROUP statement can also be specified in the MODEL or PROC TCALIS statement, but the options specified in the GROUP statement supersede those specified in the MODEL or PROC TCALIS statement for the group designated in the GROUP statement. For group options that are available in both of the GROUP and PROC TCALIS statements, see the section "Options Available in the GROUP and PROC TCALIS Statements" on page 6765. For group options that are available in the GROUP, MODEL, and PROC TCALIS statements, see the section "Options Available in GROUP, MODEL, and PROC TCALIS Statements" on page 6765. If no GROUP statement is used, a single-group analysis is assumed. The group options for a single-group analysis are specified in the PROC TCALIS statement.

The GROUP statement can be followed by subsidiary group specification statements, which specify further data processing procedures for the group designated in the GROUP statement.

Subsidiary Group Specification Statements

are for specifying additional data processing attributes for the input data. These statements are summarized in the following table:

Statement	Description
FREQ on page 6763	specifies the frequency variable for the input observations
PARTIAL on page 6798	specifies the partial variables
VAR on page 6818	specifies the set of variables in analysis
WEIGHT on page 6818	specifies the weight variable for the input observations

These statements can be used after the PROC TCALIS statement or each GROUP statement. Again, the specifications within the scope of the GROUP statement supersede those specified after the PROC TCALIS statement for the group designated in the GROUP statement.

MODEL Statement

signifies the beginning of a model specification. In the MODEL statement, you can specify the fitted groups, input and output data sets for model specification or estimates, printing options, statistical analysis, and computational options. Some of the options in the MODEL statement can also be specified in the PROC TCALIS statement. These options are called model options. Model options specified in the MODEL statement supersede those specified in the PROC TCALIS statement. For model options that are available in both of the MODEL and PROC TCALIS statements, see the section "Options Available in the MODEL and PROC TCALIS Statements" on page 6791. If no MODEL statement is used, a single model is assumed and the model options are specified in the PROC TCALIS statement.

Some of the options in the MODEL statement can also be specified in the GROUP statement. These options are called group options. The group options in the MODEL statement are transferred to the groups being fitted, but they are superseded by the group options specified in the associated GROUP statement. For group options that are available in the GROUP and the MODEL statements, see the section "Options Available in GROUP, MODEL, and PROC TCALIS Statements" on page 6765.

The MODEL statement itself does not define the model being fitted to the data; the main and subsidiary model specification statements that follow immediately after the MODEL statement do. These statements are described in the next two sections.

Main Model Specification Statements

are for specifying the type of the modeling language and the main features of the model. These statements are summarized in the following table:

Statement	Description
FACTOR on page 6752	specifies confirmatory or exploratory factor models
LINEQS on page 6766	specifies models by using linear equations
LISMOD on page 6772	specifies models in terms of LISREL-like model matrices
MSTRUCT on page 6792	specifies parameters directly in the mean and covariance matrices
PATH on page 6799	specifies models by using the causal paths of variables
RAM on page 6805	specifies models by using RAM-like lists of parameters
REFMODEL on page 6811	specifies a base model from which the target model is modified

You can use one of these statements for specifying one model. Each statement in the list represents a particular type of modeling language. After the main model specification statement, you might need to add subsidiary model specification statements, as described in the following section, to complete the model specification.

Subsidiary Model Specification Statements

are used to supplement the model specification. They are specific to the types of the modeling languages invoked by the main model specification statements, as shown in the following table:

Statement	Specification	Modeling Languages
COV on page 6747	covariance parameters	FACTOR, LINEQS
MATRIX on page 6784	parameters in matrices	LISMOD, MSTRUCT
MEAN on page 6788	mean or intercept parameters	FACTOR, LINEQS, PATH
PCOV on page 6802	(partial) covariance parameters	PATH
PVAR on page 6803	(partial) variance parameters	FACTOR, PATH
RENAMEPARM on page 6812	new parameters by renaming	REFMODEL
STD on page 6815	variance parameters	LINEQS

Notice that the RAM modeling language does not have any subsidiary model specification statements, because all model specification can be done in the RAM statement.

Model Analysis Statements

are used to request specific statistical analysis, as shown in the following table:

Statement	Analysis
DETERM on page 6750	sets variable groups for computing the determination coefficients; same as the STRUCTEQ statement
EFFPART on page 6729	displays and partitions the effects in the model
FITINDEX on page 6760	controls the fit summary output
LMTESTS on page 6776	defines the Lagrange multiplier test regions
SIMTEST on page 6814	defines simultaneous parametric function tests
STRUCTEQ on page 6750	sets variable groups for computing the determination coefficients; same as the DETERM statement
TESTFUNC on page 6817	tests individual parametric functions

Notice that the DETERM and the STRUCTEQ statements function exactly the same way.

Optimization Statements

are used to define additional parameters and parameter constraints, to fine-tune the optimization techniques, or to set the printing options in optimization, as shown in the following table:

Statement	Description
BOUNDS on page 6746	defines the bounds of parameters
LINCON on page 6766	defines the linear constraints of parameters
NLINCON on page 6794	defines the nonlinear constraints of parameters
NLOPTIONS on page 6795	sets the optimization techniques and printing options

Other Statements

that are not listed in preceding sections are summarized in the following table:

Statement	Description
BY on page 6746	fits a model to different groups separately
OUTFILES on page 6796	controls multiple output data sets
PARAMETERS on page 6798	defines additional parameters or superparameters
SAS programming statements on page 6814	define parameters or functions

Note that SAS programming statements include the ARRAY statement and the mathematical statements for defining parameter interdependence.

Single-Group Analysis Syntax

PROC TCALIS < *options* > ;
subsidiary group specification statements ;
main model specification statement ;
subsidiary model specification statements ;
model analysis statements ;
optimization statements ;
other statements ;

In a single-group analysis, there is only one group and one model. Because all model or group specifications are unambiguous, the MODEL and GROUP statements are not necessary. The order of the statements is not important for parsing purposes, although you might still like to order them in a particular way to aid understanding. Notice that the OUTFILES statement is not necessary in single-group analyses, as it is designed for multiple-group situations. Output file options in a single-group analysis can be specified in the PROC TCALIS statement.

Multiple-Group Multiple-Model Analysis Syntax

PROC TCALIS < *options* > ;
  subsidiary group specification statements ;
  model analysis statements ;
  GROUP *1* < / *group options* > ;
    subsidiary group specification statements ;
  GROUP *2* < / *group options* > ;
    subsidiary group specification statements ;
  MODEL *1* < / *model options* > ;
    main model specification statement ;
    subsidiary model specification statements ;
    model analysis statements ;
  MODEL *2* < / *model options* > ;
    main model specification statement ;
    subsidiary model specification statements ;
    model analysis statements ;
  optimization statements ;
  other statements ;

The multiple uses of the GROUP and the MODEL statements characterize the multiple-group multiple-model analysis. Unlike the single-group analysis, the order of some statements in a multiple-group multiple-model syntax is important for parsing purposes.

A GROUP statement signifies the beginning of a group specification block and designates a group number for the group. The scope of a GROUP statement extends to the subsequent subsidiary group specification statements until another MODEL or GROUP statement is encountered. In the preceding syntax, GROUP 1 and GROUP 2 have separate blocks of subsidiary group specification statements. By using additional GROUP statements, you can add as many groups as your situation calls for. Subsidiary group specification statements declared before the first GROUP statement are in the scope of the PROC TCALIS statement. This means that these subsidiary group specification statements are applied globally to all groups unless they are respecified locally within the scopes of individual GROUP statements.

A MODEL statement signifies the beginning of a model specification block and designates a model number for the model. The scope of a MODEL statement extends to the subsequent main and subsidiary model specification statements until another MODEL or GROUP statement is encountered. In the preceding syntax, MODEL 1 and MODEL 2 have separate blocks of main and subsidiary model specification statements. By using additional MODEL statements, you can add as many models as your situation calls for. If you use at least one MODEL statement, any main and subsidiary model specification statements declared before the first MODEL statement are ignored.

Some model analysis statements are also bounded by the scope of the MODEL statements. These statements are: DETERM, EFFPART, LMTESTS, and STRUCTEQ. These statements are applied only locally to the model in which they belong. To apply these statements globally to all models, put these statements before the first MODEL statement.

Other model analysis statements are not bounded by the scope of the MODEL statements. These statements are: FITINDEX, SIMTEST, and TESTFUNC. Because these statements are not model-specific, you can put these statements anywhere in a PROC TCALIS run.

Optimization and other statements are not bounded by the scope of either the GROUP or MODEL statements. You can specify them anywhere between the PROC TCALIS and the run statements without affecting the parsing of the models and the groups. For clarity of presentation, they are shown as last statement block in the syntax. Notice that the BY statement is not supported in a multiple-group setting.

PROC TCALIS Statement

PROC TCALIS < *options* > **;**

This statement invokes the procedure. There are many options in the PROC TCALIS statement. These options, together with brief descriptions, are classified into different categories in the next few sections. An alphabetical listing of these options with more details then follows.

Data Set Options

You can use the following options to specify input and output data sets:

Option	Description
DATA=	inputs the data
INEST=	inputs the initial values and constraints
INMODEL=	inputs the model specifications
INWGT=	inputs the weight matrix
OUTEST=	outputs the estimates and their covariance matrix
OUTFIT=	outputs the fit indices
OUTMODEL=	outputs the model specifications
OUTSTAT=	outputs the statistical results
OUTWGT=	outputs the weight matrix

Model and Estimation Options

You can use these options to specify details about estimation, models, and computations:

Option	Description
CORRELATION	analyzes correlation matrix
COVARIANCE	analyzes covariance matrix
DEMPHAS=	emphasizes the diagonal entries
EDF=	defines number of observations by the number of error degrees of freedom
INWGTINV	specifies that the INWGT= data set contains the inverse of the weight matrix
MEANSTR	analyzes the mean structures
METHOD=	specifies the estimation method
NOBS=	defines the number of observations
RANDOM=	specifies the seed for randomly generated initial values
RDF=	defines nobs by the number of regression df
RIDGE=	specifies the ridge factor for the covariance matrix
START=	specifies a constant for initial values
VARDEF=	specifies the variance divisor
WPENALTY=	specifies the penalty weight to fit correlations
WRIDGE=	specifies the ridge factor for the weight matrix

Options for Fit Statistics

You can use these options to modify the default behavior of fit index computations and display and to specify output file for fit indices:

Option	Description
ALPHAECV=	specifies the α level for computing the confidence interval of ECV (Browne and Cudeck 1993)
ALPHARMS=	specifies the α level for computing the confidence interval of RMSEA (Steiger and Lind 1980)
CHICORRECT=	specifies the chi-square correction factor
CLOSEFIT=	defines the close fit value
DFREDUCE=	reduces the degrees of freedom for model fit chi-square test
NOADJDF	requests no degrees-of-freedom adjustment be made for active constraints
NOINDEXTYPE	suppresses the printing of fit index types
OUTFIT=	specifies the output data set for storing fit indices

These options can also be specified in the FITINDEX statement. However, to control the display of individual fit indices, you must use the ON= and OFF= options of the FITINDEX statement.

Options for Statistical Analysis

You can use these options to request specific statistical analysis and display and to set the parameters for statistical analysis:

Option	Description
ASYCOV=	specifies the formula for computing asymptotic covariances
BIASKUR	computes the skewness and kurtosis without bias corrections
EFFPART | TOTEFF	displays total, direct, and indirect effects
G4=	specifies the algorithm for computing standard errors
KURTOSIS	computes and displays kurtosis
MODIFICATION	computes modification indices
NOMOD	suppresses modification indices
NOSTAND	suppresses the standardized output
NOSTDERR	suppresses standard error computations
PCORR	displays analyzed and estimated moment matrix
PCOVES	displays the covariance matrix of estimates
PDETERM	computes the determination coefficients
PESTIM	prints parameter estimates
PINITIAL	prints initial pattern and values
PLATCOV	computes the latent variable covariances and score coefficients
PLOTS=	specifies ODS Graphics selection
PWEIGHT	displays the weight matrix
RESIDUAL=	specifies the type of residuals being computed
SIMPLE	prints univariate statistics
SLMW=	specifies the probability limit for Wald tests
STDERR	computes the standard errors

Global Display Options

There are two different kinds of global display options: one is for selecting output; the other is for controlling the format or order of output.

You can use the following options to select printed output:

Option	Description
NOPRINT	suppresses the displayed output
PALL	displays all displayed output (ALL)
PRINT	adds default displayed output
PSHORT	reduces default output (SHORT)
PSUMMARY	displays fit summary only (SUMMARY)

In contrast to individual output printing options described in the section "Options for Statistical Analysis" on page 6723, the global display options typically control more than one output or analysis. The relations between these two types of options are summarized in the following table:

Options	PALL	PRINT	default	PSHORT	PSUMMARY
fit indices	*	*	*	*	*
linear dependencies	*	*	*	*	*
PESTIM	*	*	*	*	
iteration history	*	*	*	*	
PINITIAL	*	*	*		
SIMPLE	*	*	*		
STDERR	*	*	*		
RESIDUAL	*	*			
KURTOSIS	*	*			
PLATCOV	*	*			
TOTEFF	*	*			
PCORR	*				
MODIFICATION	*				
PWEIGHT	*				
PCOVES					
PDETERM					
PRIMAT					

Each column in the table represents a global display option. An "*" in the column means that the individual output or analysis option listed in the corresponding row turns on when the global display option in the corresponding column is specified.

Note that the column labeled with "default" is for default printing. If the NOPRINT option is not specified, a default set of output is displayed. The PRINT and PALL options add to the default output, while the PSHORT and PSUMMARY options reduce from the default output.

Note also that the PCOVES, PDETERM, and PRIMAT options cannot be turned on by any global display options. They must be specified individually.

The following global display options are for controlling formats and order of the output:

Option	Description
NOORDERSPEC	displays model specifications and results according to the input order
NOPARMNAME	suppresses the printing of parameter names in results
ORDERALL	orders all output displays according to the model numbers, group numbers, and parameter types
ORDERGROUPS	orders the group output displays according to the group numbers
ORDERMODELS	orders the model output displays according to the model numbers
ORDERSPEC	orders the model output displays according to the parameter types within each model
PARMNAME	displays parameter names in model specifications and results
PRIMAT	displays estimation results in matrix form

Optimization Options

You can use the following options to control the behavior of the optimization. Most of these options are also available in the NLOPTIONS statement.

Option	Description
ASINGULAR=	specifies the absolute singularity criterion for inverting the information matrix
COVSING=	specifies the singularity tolerance of the information matrix
FCONV=	specifies the relative function convergence criterion
GCONV=	specifies the gradient convergence criterion
INSTEP=	specifies the initial step length (RADIUS=, SALPHA=)
LINESEARCH=	specifies the line-search method
LSPRECISION=	specifies the line-search precision (SPRECISION=)
MAXFUNC=	specifies the maximum number of function calls
MAXITER=	specifies the maximum number of iterations
MSINGULAR=	specifies the relative M singularity of the information matrix
OMETHOD | TECHNIQUE=	specifies the minimization method
SINGULAR=	specifies the singularity criterion for matrix inversion
UPDATE=	specifies the update method for some optimization techniques
VSINGULAR=	specifies the relative V singularity of information matrix

Listing of PROC TCALIS Statement Options

ALPHAECV=α

specifies a $(1-\alpha)100\%$ confidence interval ($0 \leq \alpha \leq 1$) for the Browne and Cudeck (1993) expected cross-validation index (ECVI). The default value is $\alpha = 0.1$, which corresponds to a 90% confidence interval for the ECVI.

ALPHARMS=α

specifies a $(1-\alpha)100\%$ confidence interval ($0 \leq \alpha \leq 1$) for the Steiger and Lind (1980) root mean square error of approximation (RMSEA) coefficient (see Browne and Du Toit 1992). The default value is $\alpha = 0.1$, which corresponds to a 90% confidence interval for the RMSEA.

ASINGULAR | ASING=*r*

specifies an absolute singularity criterion r ($r > 0$), for the inversion of the information matrix, which is needed to compute the covariance matrix. The default value for r or *ASING* is the square root of the smallest positive double precision value.

When inverting the information matrix, the following singularity criterion is used for the diagonal pivot $d_{j,j}$ of the matrix:

$$|d_{j,j}| \leq \max(ASING, VSING * |H_{j,j}|, MSING * \max(|H_{1,1}|, \ldots, |H_{n,n}|))$$

where *VSING* and *MSING* are the specified values in the VSINGULAR= and MSINGULAR= options, respectively, and $H_{j,j}$ is the j-th diagonal element of the information matrix. Note that in many cases a normalized matrix $\mathbf{D}^{-1}\mathbf{H}\mathbf{D}^{-1}$ is decomposed (where $D^2 = diag(H)$), and the singularity criteria are modified correspondingly.

ASYCOV | ASC=*name*

specifies the formula for asymptotic covariances used in the weight matrix $\mathbf{W}$ for WLS and DWLS estimation. The ASYCOV option is effective only if METHOD= WLS or METHOD=DWLS and no INWGT= input data set is specified. The following formulas are implemented:

BIASED: Browne's (1984) formula (3.4)
biased asymptotic covariance estimates; the resulting weight matrix is at least positive semidefinite. This is the default for analyzing a covariance matrix.

UNBIASED: Browne's (1984) formula (3.8)
asymptotic covariance estimates corrected for bias; the resulting weight matrix can be indefinite (that is, can have negative eigenvalues), especially for small N.

CORR: Browne and Shapiro's (1986) formula (3.2)
(identical to DeLeeuw's (1983) formulas (2,3,4)) the asymptotic variances of the diagonal elements are set to the reciprocal of the value r specified by the WPENALTY= option (default: $r = 100$). This formula is the default for analyzing a correlation matrix.

CAUTION: Using the WLS and DWLS methods with the ASYCOV=CORR option assumes that you are fitting a correlation (rather than a covariance) structure. Because the fixed diagonal of a correlation matrix for some models does not contribute to the model's degrees of freedom, you can specify the DFREDUCE=*i* option to reduce the degrees of freedom by the number of manifest variables used in the model.

See the section "Counting the Degrees of Freedom" on page 6897 for more information.

BIASKUR
computes univariate skewness and kurtosis by formulas uncorrected for bias.

See the section "Measures of Multivariate Kurtosis" on page 6919 for more information.

CHICORRECT | CHICORR= *name* | *c*
specifies a correction factor c for the chi-square statistics for model fit. You can specify a *name* for a built-in correction factor or a value between 0 and 1 as the CHICORRECT= value. The model fit chi-square statistic is computed as:

$$\chi^2 = (1 - c)(N - k)F$$

where N is the total number of observations, k is the number of independent groups, and F is the optimized function value.

Valid *names* for the CHICORRECT= value are as follows:

EQVARCOV | COMPSYM specifies the correction factor due to Box (1949) for testing equal variances and equal covariances in a covariance matrix. The correction factor is:

$$c = \frac{p(p+1)^2(2p-3)}{6n(p-1)(p^2+p-4)}$$

where p represents the number of variables and $n = (N - 1)$, with N denoting the number of observations in a single group analysis.

CIRCULARITY | CIRCULAR | TYPEH | SPHERICITY specifies the correction factor due to Mauchly (1940) for testing circularity or Huynh and Feldt Type H matrix. The correction factor is:

$$c = \frac{2p^2 - 3p + 3}{6n(p-1)}$$

where p represents the number of variables and $n = (N - 1)$, with N denoting the number of observations in a single group analysis.

EQCOVMAT specifies the correction factor due to Box (1949) for testing equality of covariance matrices. The correction factor is:

$$c = \frac{2p^2 + 3p - 1}{6(p+1)(k-1)}\left(\sum_{i=1}^{k}\frac{1}{n_i} - \frac{1}{\sum_{i=1}^{k} n_i}\right)$$

where p represents the number of variables, k represents the number of groups, and $n_i = (N_i - 1)$, with N_i denoting the number of observations in the i-th group.

CLOSEFIT=*p*

defines the criterion value p for indicating a close fit. The smaller the better fit. The default value for close fit is .05.

CORRELATION | CORR

analyzes the correlation matrix, instead of the default covariance matrix. See the COVARIANCE option for more details.

COVARIANCE | COV

analyzes the covariance matrix. Because this is also the default analysis in PROC TCALIS, you can simply omit this option when you analyze covariance rather than correlation matrices. If the DATA= input data set is a TYPE=CORR data set (containing a correlation matrix and standard deviations), the default COV option means that the covariance matrix is computed and analyzed.

Unlike many other SAS/STAT procedures (for example, the FACTOR procedure) that analyze correlation matrices by default, PROC TCALIS uses a different default because statistical theories of structural equation modeling or covariance structure analysis are mostly developed for covariance matrices. You must use the CORR option if correlation matrices are analyzed.

COVSING=*r*

specifies a nonnegative threshold r, which determines whether the eigenvalues of the information matrix are considered to be zero. If the inverse of the information matrix is found to be singular (depending on the VSINGULAR=, MSINGULAR=, ASINGULAR=, or SINGULAR= option), a generalized inverse is computed using the eigenvalue decomposition of the singular matrix. Those eigenvalues smaller than r are considered to be zero. If a generalized inverse is computed and you do not specify the NOPRINT option, the distribution of eigenvalues is displayed.

DATA=*SAS-data-set*

specifies an input data set that can be an ordinary SAS data set or a specially structured TYPE=CORR, TYPE=COV, TYPE=UCORR, TYPE=UCOV, TYPE=SSCP, or TYPE=FACTOR SAS data set, as described in the section "Input Data Sets" on page 6819. If the DATA= option is omitted, the most recently created SAS data set is used.

DEMPHAS | DE=*r*

changes the initial values of all variance parameters by the relationship:

$$s_{new} = r(|s_{old}| + 1)$$

where s_{new} is the new initial value and s_{old} is the original initial value. The initial values of all variance parameters should always be nonnegative to generate positive definite predicted model matrices in the first iteration. By using values of $r > 1$, for example, $r = 2$, $r = 10$, and so on, you can increase these initial values to produce predicted model matrices with high positive eigenvalues in the first iteration. The DEMPHAS= option is effective independent of the way the initial values are set; that is, it changes the initial values set in the model specification as well as those set by an INMODEL= data set and those automatically generated for the FACTOR, LINEQS, LISMOD, PATH, or RAM models. It also affects the initial values set by the START= option, which uses, by default, DEMPHAS=100 if a covariance matrix is analyzed and DEMPHAS=10 for a correlation matrix.

DFREDUCE | DFRED=*i*

reduces the degrees of freedom of the model fit χ^2 test by i. In general, the number of degrees of freedom is the total number of nonredundant elements in all moment matrices minus the number of parameters, t. Because negative values of i are allowed, you can also increase the number of degrees of freedom by using this option. If the DFREDUCE= option is used in a correlation structure analysis, PROC TCALIS does not additionally reduce the degrees of freedom by the number of constant elements in the diagonal of the predicted model matrix, which is otherwise done automatically. See the section "Counting the Degrees of Freedom" on page 6897 for more information.

EDF | DFE=*n*

makes the effective number of observations $n + 1$. You can also use the NOBS= option to specify the number of observations.

EFFPART | PARTEFF | TOTEFF | TE

computes and displays total, direct, and indirect effects for the unstandardized and standardized estimation results. Standard errors for the effects are also computed. Note that this displayed output is not automatically included in the output generated by the PALL option.

Note also that in some situations computations of total effects and their partitioning are not appropriate. While total and indirect effects must converge in recursive models (models with no cyclic paths among variables), they do not always converge in nonrecursive models. When total or indirect effects do not converge, it is not appropriate to partition the effects. Therefore, before partitioning the total effects, the convergence criterion must be met. To check the convergence of the effects, PROC TCALIS computes and displays the "stability coefficient of reciprocal causation"— that is, the largest modulus of the eigenvalues of the $\boldsymbol{\beta}$ matrix, which is the square matrix that contains the path coefficients of all endogenous variables in the model. Stability coefficients less than one provide a necessary and sufficient condition for the convergence of the total and the indirect effects. Otherwise, PROC TCALIS does not show results for total effects and their partitioning. See the section "Stability Coefficient of Reciprocal Causation" on page 6915 for more information about the computation of the stability coefficient.

FCONV | FTOL=*r*

specifies the relative function convergence criterion. The optimization process is terminated when the relative difference of the function values of two consecutive iterations is smaller than the specified value of r; that is,

$$\frac{|f(x^{(k)}) - f(x^{(k-1)})|}{\max(|f(x^{(k-1)})|, FSIZE)} \leq r$$

where *FSIZE* can be defined by the FSIZE= option in the NLOPTIONS statement. The default value is $r = 10^{-FDIGITS}$, where *FDIGITS* either can be specified in the NLOPTIONS statement or is set by default to $-\log_{10}(\epsilon)$, where ϵ is the machine precision.

G4=*i*

instructs that the algorithm to compute the approximate covariance matrix of parameter estimates used for computing the approximate standard errors and modification indices when the information matrix is singular. If the number of parameters t used in the model you analyze

is smaller than the value of i, the time-expensive Moore-Penrose (G4) inverse of the singular information matrix is computed by eigenvalue decomposition. Otherwise, an inexpensive pseudo (G1) inverse is computed by sweeping. By default, $i = 60$.

See the section "Estimation Criteria" on page 6888 for more details.

GCONV | GTOL=*r*

specifies the relative gradient convergence criterion.

Termination of all techniques (except the CONGRA technique) requires the following normalized predicted function reduction to be smaller than r. That is,

$$\frac{[g(x^{(k)})]'[\mathbf{G}^{(k)}]^{-1}g(x^{(k)})}{\max(|f(x^{(k)})|, FSIZE)} \leq r$$

where *FSIZE* can be defined by the FSIZE= option in the NLOPTIONS statement. For the CONGRA technique (where a reliable Hessian estimate $\mathbf{G}$ is not available),

$$\frac{\| g(x^{(k)}) \|_2^2 \quad \| s(x^{(k)}) \|_2}{\| g(x^{(k)}) - g(x^{(k-1)}) \|_2 \max(|f(x^{(k)})|, FSIZE)} \leq r$$

is used. The default value is $r = 10^{-8}$.

INEST | INVAR | ESTDATA=*SAS-data-set*

specifies an input data set that contains initial estimates for the parameters used in the optimization process and can also contain boundary and general linear constraints on the parameters. Typical applications of this option are to specify an OUTEST= data set from a previous PROC TCALIS analysis. The initial estimates are taken from the values of the PARMS observation in the INEST= data set.

INMODEL | INRAM=*SAS-data-set*

specifies an input data set that contains in RAM list form all information needed to specify an analysis model. The INMODEL= data set is described in the section "Input Data Sets" on page 6819. Typically, this input data set is an OUTMODEL= data set (possibly modified) from a previous PROC TCALIS analysis. If you use an INMODEL= data set to specify the analysis model, you should not use any of the main or subsidiary model specification statements, but you can use the BOUNDS and PARAMETERS statements and program statements. If the INMODEL= option is omitted, you should define the analysis model with one of the main model specification statements.

INSTEP=*r*

For highly nonlinear objective functions, such as the EXP function, the default initial radius of the trust-region algorithms (TRUREG, DBLDOG, and LEVMAR) or the default step length of the line-search algorithms can produce arithmetic overflows. If an arithmetic overflow occurs, specify decreasing values of $0 < r < 1$ such as INSTEP=1E−1, INSTEP=1E−2, INSTEP=1E−4, and so on, until the iteration starts successfully.

- For trust-region algorithms (TRUREG, DBLDOG, and LEVMAR), the INSTEP option specifies a positive factor for the initial radius of the trust region. The default initial trust-region radius is the length of the scaled gradient, and it corresponds to the default radius factor of $r = 1$.

- For line-search algorithms (NEWRAP, CONGRA, and QUANEW), INSTEP specifies an upper bound for the initial step length for the line search during the first five iterations. The default initial step length is $r = 1$.

For more details, see the section "Computational Problems" on page 6932.

INWGT< (INV) >=*SAS-data-set*

specifies an input data set that contains the weight matrix **W** used in generalized least squares (GLS), weighted least squares (WLS, ADF), or diagonally weighted least squares (DWLS) estimation, if you do not specify the INV option at the same time. The weight matrix must be positive definite because its inverse must be defined in the computation of the objective function. If the weight matrix **W** defined by an INWGT= data set is not positive definite, it can be ridged using the WRIDGE= option. See the section "Estimation Criteria" on page 6888 for more information. If you specify the INWGT(INV)= option, the INWGT= data set contains the inverse of the weight matrix, rather than the weight matrix itself. Specifying the INWGT(INV)= option is equivalent to specifying the INWGT= and INWGTINV options simultaneously. With the INWGT(INV)= specification, the input matrix is not required to be positive definite. See the INWGTINV option for more details. If no INWGT= data set is specified, default settings for the weight matrices are used in the estimation process. The INWGT= data set is described in the section "Input Data Sets" on page 6819. Typically, this input data set is an OUTWGT= data set from a previous PROC TCALIS analysis.

INWGTINV

specifies that the INWGT= data set contains the inverse of the weight matrix, rather than the weight matrix itself. This option is effective only with an input weight matrix specified in the INWGT= data set and with the generalized least squares (GLS), weighted least squares (WLS or ADF), or diagonally weighted least squares (DWLS) estimation. With this option, the input matrix provided in the INWGT= data set is not required to be positive definite. Also, the ridging requested by the WRIDGE= option is ignored when you specify the INWGTINV option.

KURTOSIS | KU

computes and displays univariate kurtosis and skewness, various coefficients of multivariate kurtosis, and the numbers of observations that contribute most to the normalized multivariate kurtosis. See the section "Measures of Multivariate Kurtosis" on page 6919 for more information. Using the KURTOSIS option implies the SIMPLE display option. This information is computed only if the DATA= data set is a raw data set, and it is displayed by default if the PRINT option is specified. The multivariate least squares kappa and the multivariate mean kappa are displayed only if you specify METHOD=WLS and the weight matrix is computed from an input raw data set. All measures of skewness and kurtosis are corrected for the mean. Using the BIASKUR option displays the biased values of univariate skewness and kurtosis.

LINESEARCH | LIS | SMETHOD | SM=*i*

specifies the line-search method for the CONGRA, QUANEW, and NEWRAP optimization techniques. Refer to Fletcher (1980) for an introduction to line-search techniques. The value of i can be any integer between 1 and 8, inclusively; the default is $i = 2$.

LIS=1 specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is similar to one used by the Harwell subroutine library.

LIS=2 specifies a line-search method that needs more function calls than gradient calls for quadratic and cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the LSPRECISION= option.

LIS=3 specifies a line-search method that needs the same number of function and gradient calls for cubic interpolation and cubic extrapolation; this method is implemented as shown in Fletcher (1987) and can be modified to an exact line search by using the LSPRECISION= option.

LIS=4 specifies a line-search method that needs the same number of function and gradient calls for stepwise extrapolation and cubic interpolation.

LIS=5 specifies a line-search method that is a modified version of LIS=4.

LIS=6 specifies golden-section line search (Polak 1971), which uses only function values for linear approximation.

LIS=7 specifies bisection line search (Polak 1971), which uses only function values for linear approximation.

LIS=8 specifies the Armijo line-search technique (Polak 1971), which uses only function values for linear approximation.

LSPRECISION | LSP=*r*

SPRECISION | SP=*r*

specifies the degree of accuracy that should be obtained by the line-search algorithms LIS=2 and LIS=3. Usually an imprecise line search is inexpensive and successful. For more difficult optimization problems, a more precise and more expensive line search might be necessary (Fletcher 1980, p.22). The second (default for NEWRAP, QUANEW, and CONGRA) and third line-search methods approach exact line search for small LSPRECISION= values. If you have numerical problems, you should decrease the LSPRECISION= value to obtain a more precise line search. The default LSPRECISION= values are displayed in the following table.

OMETHOD=	UPDATE=	LSP default
QUANEW	DBFGS, BFGS	$r = 0.4$
QUANEW	DDFP, DFP	$r = 0.06$
CONGRA	all	$r = 0.1$
NEWRAP	no update	$r = 0.9$

For more details, refer to Fletcher (1980, pp. 25–29).

MAXFUNC | MAXFU=*i*

specifies the maximum number i of function calls in the optimization process. The default values are displayed in the following table.

OMETHOD=	MAXFUNC default
LEVMAR, NEWRAP, NRRIDG, TRUREG	$i = 125$
DBLDOG, QUANEW	$i = 500$
CONGRA	$i = 1000$

The default is used if you specify MAXFUNC=0. The optimization can be terminated only after completing a full iteration. Therefore, the number of function calls that is actually performed can exceed the number that is specified by the MAXFUNC= option.

MAXITER | MAXIT=*i* < *n* >

specifies the maximum number i of iterations in the optimization process. The default values are displayed in the following table.

OMETHOD=	MAXITER default
LEVMAR, NEWRAP, NRRIDG, TRUREG	$i = 50$
DBLDOG, QUANEW	$i = 200$
CONGRA	$i = 400$

The default is used if you specify MAXITER=0 or if you omit the MAXITER option.

The optional second value n is valid only for OMETHOD=QUANEW with nonlinear constraints. It specifies an upper bound n for the number of iterations of an algorithm and reduces the violation of nonlinear constraints at a starting point. The default is n=20. For example, specifying

```
maxiter= . 0
```

means that you do not want to exceed the default number of iterations during the main optimization process and that you want to suppress the feasible point algorithm for nonlinear constraints.

MEANSTR

invokes the analysis of mean structures. By default, no mean structures are analyzed. You can also analyze the mean structures by specifying any mean or intercept parameters in your model without using the MEANSTR option.

METHOD | MET | M=*name*

specifies the method of parameter estimation. The default is METHOD=ML. Valid values for *name* are as follows:

ML | M | MAX
: performs normal-theory maximum-likelihood parameter estimation. The ML method requires a nonsingular covariance or correlation matrix.

GLS | G
: performs generalized least squares parameter estimation. If no INWGT= data set is specified, the GLS method uses the inverse sample covariance or correlation matrix as the weight matrix **W**. Therefore, METHOD=GLS requires a nonsingular covariance or correlation matrix.

WLS | W | ADF	performs weighted least squares parameter estimation. If no INWGT= data set is specified, the WLS method uses the inverse matrix of estimated asymptotic covariances of the sample covariance or correlation matrix as the weight matrix **W**. In this case, the WLS estimation method is equivalent to Browne's asymptotically distribution-free estimation (Browne 1982, 1984). The WLS method requires a nonsingular weight matrix.
DWLS | D	performs diagonally weighted least squares parameter estimation. If no INWGT= data set is specified, the DWLS method uses the inverse diagonal matrix of asymptotic variances of the input sample covariance or correlation matrix as the weight matrix **W**. The DWLS method requires a nonsingular diagonal weight matrix.
ULS | LS | U	performs unweighted least squares parameter estimation.
LSML | LSM | LSMAX	performs unweighted least squares followed by normal-theory maximum-likelihood parameter estimation.
LSGLS | LSG	performs unweighted least squares followed by generalized least squares parameter estimation.
LSWLS | LSW | LSADF	performs unweighted least squares followed by weighted least squares parameter estimation.
LSDWLS | LSD	performs unweighted least squares followed by diagonally weighted least squares parameter estimation.
NONE | NO	uses no estimation method. This option is suitable for checking the validity of the input information and for displaying the model matrices and initial values.

MODIFICATION | MOD

computes and displays Lagrange multiplier (LM) test indices for constant parameter constraints, equality parameter constraints, and active boundary constraints, as well as univariate and multivariate Wald test indices. The modification indices are not computed in the case of unweighted or diagonally weighted least squares estimation.

The Lagrange multiplier test (Bentler 1986; Lee 1985; Buse 1982) provides an estimate of the χ^2 reduction that results from dropping the constraint. For constant parameter constraints and active boundary constraints, the approximate change of the parameter value is displayed also. You can use this value to obtain an initial value if the parameter is allowed to vary in a modified model. See the section "Modification Indices" on page 6917 for more information.

Relying solely on the LM tests to modify your model can lead to unreliable models that capitalize purely on sampling errors. See MacCallum, Roznowski, and Necowitz (1992) for the use of LM tests.

MSINGULAR | MSING=*r*

specifies a relative singularity criterion r ($r > 0$) for the inversion of the information matrix, which is needed to compute the covariance matrix. If you do not specify the SINGULAR= option, the default value for r or *MSING* is 1E−12; otherwise, the default value is 1E−4 × *SING*, where *SING* is the specified SINGULAR= value.

When inverting the information matrix, the following singularity criterion is used for the diagonal pivot $d_{j,j}$ of the matrix:

$$|d_{j,j}| \leq \max(ASING, VSING * |H_{j,j}|, MSING * \max(|H_{1,1}|, \ldots, |H_{n,n}|))$$

where *ASING* and *VSING* are the specified values of the ASINGULAR= and VSINGULAR= options, respectively, and $H_{j,j}$ is the j-th diagonal element of the information matrix. Note that in many cases a normalized matrix $\mathbf{D}^{-1}\mathbf{H}\mathbf{D}^{-1}$ is decomposed (where $D^2 = diag(H)$), and the singularity criteria are modified correspondingly.

NOADJDF
turns off the automatic adjustment of degrees of freedom when there are active constraints in the analysis. When the adjustment is in effect, most fit statistics and the associated probability levels will be affected. This option should be used when you believe that the active constraints observed in the current sample will have little chance to occur in repeated sampling. See the section "Adjustment of Degrees of Freedom" on page 6899 for more discussion on the issue.

NOBS= *nobs*
specifies the number of observations. If the DATA= input data set is a raw data set, *nobs* is defined by default to be the number of observations in the raw data set. The NOBS= and EDF= options override this default definition. You can use the RDF= option to modify the *nobs* specification. If the DATA= input data set contains a covariance, correlation, or scalar product matrix, you can specify the number of observations either by using the NOBS=, EDF=, and RDF= options in the PROC TCALIS statement or by including a _TYPE_='N' observation in the DATA= input data set.

NOINDEXTYPE
disables the display of index types in the fit summary table.

NOMOD
suppresses the computation of modification indices. The NOMOD option is useful in connection with the PALL option because it saves computing time.

NOORDERSPEC
prints the model results in the order they appear in the input specifcations. This is the default printing behavior. You can use this option to reset the default at a local model when the ORDERSPEC option might have been set in a statement at a more global level.

NOPARMNAME
suppresses the printing of parameter names in the model results. The default is to print the parameter names.

NOPRINT | NOP
suppresses the displayed output. Note that this option temporarily disables the Output Delivery System (ODS). See Chapter 20, "Using the Output Delivery System," for more information.

NOSTAND
suppresses the printing of standardized results. The default is to print the standardized results.

NOSTDERR | NOSE

specifies that standard errors should not be computed. Standard errors are not computed for unweighted least squares (ULS) or diagonally weighted least squares (DWLS) estimation. In general, standard errors are computed even if the STDERR display option is not used (for file output).

OMETHOD | OM=*name*

TECHNIQUE | TECH=*name*

specifies the optimization method or technique. Because there is no single nonlinear optimization algorithm available that is clearly superior (in terms of stability, speed, and memory) for all applications, different types of optimization methods or techniques are provided in the TCALIS procedure. The optimization method or technique is specified by using one of the following *names* in the OMETHOD= option:

CONGRA | CG chooses one of four different conjugate-gradient optimization algorithms, which can be more precisely defined with the UPDATE= option and modified with the LINESEARCH= option. The conjugate-gradient techniques need only $O(t)$ memory compared to the $O(t^2)$ memory for the other three techniques, where t is the number of parameters. On the other hand, the conjugate-gradient techniques are significantly slower than other optimization techniques and should be used only when memory is insufficient for more efficient techniques. When you choose this option, UPDATE=PB by default. This is the default optimization technique if there are more than 400 parameters to estimate.

DBLDOG | DD performs a version of double dogleg optimization, which uses the gradient to update an approximation of the Cholesky factor of the Hessian. This technique is, in many aspects, very similar to the dual quasi-Newton method, but it does not use line search. The implementation is based on Dennis and Mei (1979) and Gay (1983).

LEVMAR | LM | MARQUARDT performs a highly stable (but for large problems, memory- and time-consuming) Levenberg-Marquardt optimization technique, a slightly improved variant of the Moré (1978) implementation. This is the default optimization technique if there are fewer than 40 parameters to estimate.

NEWRAP | NRA performs a usually stable (but for large problems, memory- and time-consuming) Newton-Raphson optimization technique. The algorithm combines a line-search algorithm with ridging, and it can be modified with the LINESEARCH= option. In releases prior to Release 6.11, this option invokes the NRRIDG option.

NRRIDG | NRR | NR | NEWTON performs a usually stable (but for large problems, memory- and time-consuming) Newton-Raphson optimization technique. This algorithm does not perform a line search. Since OMETHOD=NRRIDG uses an orthogonal decomposition of the approximate Hessian, each iteration of OMETHOD=NRRIDG can be slower than that of OMETHOD=NEWRAP, which works with

Cholesky decomposition. However, usually OMETHOD=NRRIDG needs fewer iterations than OMETHOD=NEWRAP.

QUANEW | QN
: chooses one of four different quasi-Newton optimization algorithms that can be more precisely defined with the UPDATE= option and modified with the LINESEARCH= option. If boundary constraints are used, these techniques sometimes converge slowly. When you choose this option, UPDATE=DBFGS by default. If nonlinear constraints are specified in the NLINCON statement, a modification of Powell's (1982a, 1982b) VMCWD algorithm is used, which is a sequential quadratic programming (SQP) method. This algorithm can be modified by specifying VERSION=1, which replaces the update of the Lagrange multiplier estimate vector μ to the original update of Powell (1978a, 1978b) that is used in the VF02AD algorithm. This can be helpful for applications with linearly dependent active constraints. The QUANEW technique is the default optimization technique if there are nonlinear constraints specified or if there are more than 40 and fewer than 400 parameters to estimate. The QUANEW algorithm uses only first-order derivatives of the objective function and, if available, of the nonlinear constraint functions.

TRUREG | TR
: performs a usually very stable (but for large problems, memory- and time-consuming) trust-region optimization technique. The algorithm is implemented similar to Gay (1983) and Moré and Sorensen (1983).

NONE | NO
: does not perform any optimization. This option is similar to METHOD=NONE, but OMETHOD=NONE also computes and displays residuals and goodness-of-fit statistics. If you specify METHOD=ML, METHOD=LSML, METHOD=GLS, METHOD=LSGLS, METHOD=WLS, or METHOD=LSWLS, this option enables computing and displaying (if the display options are specified) of the standard error estimates and modification indices corresponding to the input parameter estimates.

For fewer than 40 parameters ($t < 40$), OMETHOD=LEVMAR (Levenberg-Marquardt) is the default optimization technique. For $40 \leq t < 400$, OMETHOD=QUANEW (quasi-Newton) is the default method, and for $t \geq 400$, OMETHOD=CONGRA (conjugate gradient) is the default method. Each optimization method or technique can be modified in various ways. See the section "Use of Optimization Techniques" on page 6923 for more details.

UPDATE | UPD=*name*

specifies the update method for the quasi-Newton or conjugate-gradient optimization technique.

For OMETHOD=CONGRA, the following updates can be used:

PB
: performs the automatic restart update method of Powell (1977) and Beale (1972). This is the default.

FR
: performs the Fletcher-Reeves update (Fletcher 1980, p. 63).

PR
: performs the Polak-Ribiere update (Fletcher 1980, p. 66).

CD performs a conjugate-descent update of Fletcher (1987).

For OMETHOD=DBLDOG, the following updates (Fletcher 1987) can be used:

DBFGS performs the dual Broyden, Fletcher, Goldfarb, and Shanno (BFGS) update of the Cholesky factor of the Hessian matrix. This is the default.

DDFP performs the dual Davidon, Fletcher, and Powell (DFP) update of the Cholesky factor of the Hessian matrix.

For OMETHOD=QUANEW, the following updates (Fletcher 1987) can be used:

BFGS performs original BFGS update of the inverse Hessian matrix. This is the default for earlier releases.

DFP performs the original DFP update of the inverse Hessian matrix.

DBFGS performs the dual BFGS update of the Cholesky factor of the Hessian matrix. This is the default.

DDFP performs the dual DFP update of the Cholesky factor of the Hessian matrix.

ORDERALL
prints the model and group results in the order of the model or group numbers, starting from the smallest number. It also arrange some model results by the parameter types. In effect, this option turns on the ORDERGROUPS, ORDERMODELS, and ORDERSPEC options. The ORDERALL is not a default option. By default, the printing of the results follow the order of the input specifications.

ORDERGROUPS | ORDERG
prints the group results in the order of the group numbers, starting from the smallest number. The default behavior, however, is to print the group results in the order they appear in the input specifications.

ORDERMODELS | ORDERMO
prints the model results in the order of the model numbers, starting from the smallest number. The default behavior, however, is to print the model results in the order they appear in the input specifications.

ORDERSPEC
arranges some model results by the types of parameters. The default behavior, however, is to print the results in the order they appear in the input specifications.

OUTEST=*SAS-data-set*
creates an output data set that contains the parameter estimates, their gradient, Hessian matrix, and boundary and linear constraints. For METHOD=ML, METHOD=GLS, and METHOD=WLS, the OUTEST= data set also contains the information matrix, the approximate covariance matrix of the parameter estimates ((generalized) inverse of information matrix), and approximate standard errors. If linear or nonlinear equality or active inequality constraints are present, the Lagrange multiplier estimates of the active constraints, the projected Hessian, and the Hessian of the Lagrange function are written to the data set.

See the section "OUTEST= SAS-data-set" on page 6823 for a description of the OUTEST= data set. If you want to create a permanent SAS data set, you must specify a two-level name. Refer to the chapter titled "SAS Data Files" in *SAS Language Reference: Concepts* for more information about permanent data sets.

OUTFIT=*SAS-data-set*
creates an output data set that contains the values of the fit indices. See the section "OUTFIT= SAS-data-set" on page 6837 for details.

OUTMODEL | OUTRAM=*SAS-data-set*
creates an output data set that contains the model information for the analysis, the parameter estimates, and their standard errors. An OUTMODEL= data set can be used as an input INMODEL= data set in a subsequent analysis by PROC TCALIS. The OUTMODEL= data set also contains a set of fit indices; the section "OUTMODEL= SAS-data-set" on page 6827 provides more details. If you want to create a permanent SAS data set, you must specify a two-level name.

Refer to the chapter titled "SAS Data Files" in *SAS Language Reference: Concepts* for more information about permanent data sets.

OUTSTAT=*SAS-data-set*
creates an output data set that contains the BY group variables, the analyzed covariance or correlation matrices, and the predicted and residual covariance or correlation matrices of the analysis. You can specify the correlation or covariance matrix in an OUTSTAT= data set as an input DATA= data set in a subsequent analysis by PROC TCALIS. See the section "OUTSTAT= SAS-data-set" on page 6832 for a description of the OUTSTAT= data set. If the model contains latent variables, this data set also contains the predicted covariances between latent and manifest variables and the latent variable score regression coefficients (see the PLATCOV option on page 6741). If the FACTOR statement is used, the OUTSTAT= data set also contains the rotated and unrotated factor loadings, the unique variances, the matrix of factor correlations, the transformation matrix of the rotation, and the matrix of standardized factor loadings.

You can use the latent variable score regression coefficients with PROC SCORE to compute factor scores.

If you want to create a permanent SAS data set, you must specify a two-level name.

Refer to the chapter titled "SAS Data Files" in *SAS Language Reference: Concepts* for more information about permanent data sets.

OUTWGT=*SAS-data-set*
creates an output data set that contains the elements of the weight matrix $\mathbf{W}$ or the its inverse $\mathbf{W}^{-1}$ used in the estimation process. The inverse of the weight matrix is output only when you specify an INWGT= data set with the INWGT= and INWGTINV options (or the INWGT(INV)= option alone) in the same analysis. As a result, the entries in the INWGT= and OUTWGT= data sets are consistent. In other situations where the weight matrix is computed by the procedure or obtained from the OUTWGT= data set without the INWGTINV option, the weight matrix is output in the OUTWGT= data set. Furthermore, if the weight matrix is computed by the procedure, the OUTWGT= data set contains the elements of the weight matrix on which the WRIDGE= and the WPENALTY= options are applied.

You cannot create an OUTWGT= data set with an unweighted least squares or maximum likelihood estimation. The weight matrix is defined only in the GLS, WLS (ADF), or DWLS fit function. An OUTWGT= data set can be used as an input INWGT= data set in a subsequent analysis by PROC TCALIS. See the section "OUTWGT= SAS-data-set" on page 6836 for the description of the OUTWGT= data set. If you want to create a permanent SAS data set, you must specify a two-level name.

Refer to the chapter titled "SAS Data Files" in *SAS Language Reference: Concepts* for more information about permanent data sets.

PALL | ALL

displays all optional output except the output generated by the PCOVES and PDETERM options.

CAUTION: The PALL option includes the very expensive computation of the modification indices. If you do not really need modification indices, you can save computing time by specifying the NOMOD option in addition to the PALL option.

PARMNAME

prints the parameter names in the model results. This is the default behavior. You can use this option to reset the default at a local MODEL statement when the NOPARMNAME option might have been set in a statement at a more global level.

PCORR | CORR

displays the covariance or correlation matrix that is analyzed and the predicted model covariance or correlation matrix.

PCOVES | PCE

displays the following:

- the information matrix
- the approximate covariance matrix of the parameter estimates (generalized inverse of the information matrix)
- the approximate correlation matrix of the parameter estimates

The covariance matrix of the parameter estimates is not computed for estimation methods ULS and DWLS. This displayed output is not included in the output generated by the PALL option.

PDETERM | PDE

displays three coefficients of determination: the determination of all equations (DETAE), the determination of the structural equations (DETSE), and the determination of the manifest variable equations (DETMV). These determination coefficients are intended to be global means of the squared multiple correlations for different subsets of model equations and variables. The coefficients are displayed only when you specify a FACTOR, LINEQS, LISMOD, PATH, or RAM model, but they are displayed for all five estimation methods: ULS, GLS, ML, WLS, and DWLS.

You can use the STRUCTEQ statement to define which equations are structural equations. If you do not use the STRUCTEQ statement, PROC TCALIS uses its own default definition to identify structural equations.

The term "structural equation" is not defined in a unique way. The LISREL program defines the structural equations by the user-defined BETA matrix. In PROC TCALIS, the default definition of a structural equation is an equation that has a dependent left-side variable that appears at least once on the right side of another equation, or an equation that has at least one right-side variable that appears at the left side of another equation. Therefore, PROC TCALIS sometimes identifies more equations as structural equations than the LISREL program does.

PESTIM | PES
displays the parameter estimates. In some cases, this includes displaying the standard errors and t values.

PINITIAL | PIN
displays the model specification with initial estimates and the vector of initial values.

PLATCOV | PLC
displays the following:

- the estimates of the covariances among the latent variables
- the estimates of the covariances between latent and manifest variables
- the latent variable score regression coefficients

The estimated covariances between latent and manifest variables and the latent variable score regression coefficients are written to the OUTSTAT= data set. You can use the score coefficients with PROC SCORE to compute factor scores.

PLOTS | PLOT < = *plot-request* >
PLOTS | PLOT < = (*plot-request* < ... *plot-request* >) >
specifies the ODS graphical plots. Currently, the only available ODS graphical plots in PROC TCALIS are for residual histograms. Also, when the residual histograms are requested, the bar charts of residual tallies are suppressed. To display these bar charts with the residual histograms, you must use the RESIDUAL(TALLY) option.

When you specify only one *plot-request*, you can omit the parentheses around the *plot-request*. For example:

PLOTS=ALL
PLOTS=RESIDUALS

The following table shows the available *plot-requests*:

Plot-request	Plot Description
ALL	all available plots
NONE	no ODS graphical plots
RESIDUALS	distribution of residuals

To request the ODS graphical plots, you must also specify the `ods graphics on` statement. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

PRIMAT | PMAT
displays parameter estimates, approximate standard errors, and *t* values in matrix form if you specify the analysis model using the RAM or LINEQS statement.

PRINT | PRI
adds the options KURTOSIS, RESIDUAL, PLATCOV, and TOTEFF to the default output.

PSHORT | SHORT | PSH
excludes the output produced by the PINITIAL, SIMPLE, and STDERR options from the default output.

PSUMMARY | SUMMARY | PSUM
displays the fit assessment table only.

PWEIGHT | PW
displays the weight matrix **W** used in the estimation. The weight matrix is displayed after the WRIDGE= and the WPENALTY= options are applied to it. However, if you specify an INWGT= data set by the INWGT= and INWGTINV options (or the INWGT(INV)= option alone) in the same analysis, this option displays the elements of the inverse of the weight matrix.

RADIUS=*r*
is an alias for the INSTEP= option for Levenberg-Marquardt minimization.

RANDOM =*i*
specifies a positive integer as a seed value for the pseudo-random number generator to generate initial values for the parameter estimates for which no other initial value assignments in the model definitions are made. Except for the parameters in the diagonal locations of the central matrices in the model, the initial values are set to random numbers in the range $0 \leq r \leq 1$. The values for parameters in the diagonals of the central matrices are random numbers multiplied by 10 or 100. See the section "Initial Estimates" on page 6922 for more information.

RDF | DFR=*n*
makes the effective number of observations the actual number of observations minus the RDF= value. The degree of freedom for the intercept should not be included in the RDF= option. If you use PROC TCALIS to compute a regression model, you can specify RDF= *number-of-regressor-variables* to get approximate standard errors equal to those computed by PROC REG.

RESIDUAL | RES <(TALLY | TALLIES)> <= NORM | VARSTAND | ASYSTAND>
displays the raw and normalized residual covariance matrix, the rank order of the largest residuals, and a bar chart of the residual tallies. This information is displayed by default when you specify the PRINT option.

Three types of normalized or standardized residual matrices can be chosen with the RESIDUAL= specification.

RESIDUAL= NORM	normalized residuals
RESIDUAL= VARSTAND	variance standardized residuals

RESIDUAL= ASYSTAND	asymptotically standardized residuals

When ODS graphical plots of residuals are also requested, the bar charts of residual tallies are suppressed. They are replaced with high quality graphical histograms showing residual distributions. If you still want to display the bar charts in this situation, use the RESIDUAL(TALLY) or RESIDUAL(TALLY)= option.

See the section "Assessment of Fit" on page 6900 for more details.

RIDGE< =*r* >
defines a ridge factor r for the diagonal of the covariance or correlation matrix $\mathbf{S}$ that is analyzed. The matrix $\mathbf{S}$ is transformed to:

$$\mathbf{S} \longrightarrow \tilde{\mathbf{S}} = \mathbf{S} + r(\mathrm{diag}(\mathbf{S}))$$

If you do not specify r in the RIDGE option, PROC TCALIS tries to ridge the covariance or correlation matrix $\mathbf{S}$ so that the smallest eigenvalue is about 10^{-3}. Because the weight matrix in the GLS method is the same as the observed covariance or correlation matrix, the RIDGE= option also applies to the weight matrix for the GLS estimation, unless you input the weight matrix by the INWGT= option.

CAUTION: The covariance or correlation matrix in the OUTSTAT= output data set does not contain the ridged diagonal.

SALPHA=*r*
is an alias for the INSTEP= option for line-search algorithms.

SIMPLE | S
displays means, standard deviations, skewness, and univariate kurtosis if available. This information is displayed when you specify the PRINT option. If the KURTOSIS option is specified, the SIMPLE option is set by default.

SINGULAR | SING =*r*
specifies the singularity criterion r $(0 < r < 1)$ used, for example, for matrix inversion. The default value is the square root of the relative machine precision or, equivalently, the square root of the largest double precision value that, when added to 1, results in 1.

SLMW=*r*
specifies the probability limit used for computing the stepwise multivariate Wald test. The process stops when the univariate probability is smaller than r. The default value is $r = 0.05$.

SPRECISION | SP=*r*
is an alias for the LSPRECISION= option.

START =*r*
specifies initial estimates for parameters as multiples of the r value. In all TCALIS models, you can supply initial estimates individually as parenthesized values after each parameter name. Unspecified initial estimates are usually computed by various reasonable initial estimation methods in PROC TCALIS. If none of the initialization methods is able to compute all the unspecified initial estimates, then the remaining unspecified initial estimates are set to

r, $10|r|$, or $100|r|$. For variance parameters, $100|r|$ is used for covariance structure analyses and $10|r|$ is used for correlation structure analyses. For other types of parameters, r is used. The default value is $r = 0.5$. If the DEMPHAS= option is used, the initial values of the variance parameters are multiplied by the value specified in the DEMPHAS= option. See the section "Initial Estimates" on page 6922 for more information.

STDERR | SE

displays approximate standard errors if estimation methods other than unweighted least squares (ULS) or diagonally weighted least squares (DWLS) are used (and the NOSTDERR option is not specified). If you specify neither the STDERR nor the NOSTDERR option, the standard errors are computed for the OUTMODEL= data set. This information is displayed by default when you specify the PRINT option.

VARDEF= DF | N | WDF | WEIGHT | WGT

specifies the divisor used in the calculation of covariances and standard deviations. The default value is VARDEF=DF. The values and associated divisors are displayed in the following table, where k is the number of partial variables specified in the PARTIAL statement. When a WEIGHT statement is used, w_j is the value of the WEIGHT variable in the jth observation, and the summation is performed only over observations with positive weight.

Value	Description	Divisor
DF	degrees of freedom	$N - k - 1$
N	number of observations	N
WDF	sum of weights DF	$\sum_j^N w_j - k - 1$
WEIGHT | WGT	sum of weights	$\sum_j^N w_j$

VSINGULAR | VSING=*r*

specifies a relative singularity criterion r ($r > 0$) for the inversion of the information matrix, which is needed to compute the covariance matrix. If you do not specify the SINGULAR= option, the default value for r or *VSING* is 1E−8; otherwise, the default value is *SING*, which is the specified SINGULAR= value.

When inverting the information matrix, the following singularity criterion is used for the diagonal pivot $d_{j,j}$ of the matrix:

$$|d_{j,j}| \leq \max(ASING, VSING * |H_{j,j}|, MSING * \max(|H_{1,1}|, \ldots, |H_{n,n}|))$$

where *ASING* and *MSING* are the specified values of the ASINGULAR= and MSINGULAR= options, respectively, and $H_{j,j}$ is the j-th diagonal element of the information matrix. Note that in many cases a normalized matrix $\mathbf{D}^{-1}\mathbf{H}\mathbf{D}^{-1}$ is decomposed (where $D^2 = diag(H)$), and the singularity criteria are modified correspondingly.

WPENALTY | WPEN=*r*

specifies the penalty weight $r \geq 0$ for the WLS and DWLS fit of the diagonal elements of a correlation matrix (constant 1s). The criterion for weighted least squares estimation of a correlation structure is

$$\mathbf{F}_{WLS} = \sum_{i=2}^{n}\sum_{j=1}^{i-1}\sum_{k=2}^{n}\sum_{l=1}^{k-1} w^{ij,kl}(s_{ij} - c_{ij})(s_{kl} - c_{kl}) + r\sum_{i}^{n}(s_{ii} - c_{ii})^2$$

where r is the penalty weight specified by the WPENALTY=r option and the $w^{ij,kl}$ are the elements of the inverse of the reduced $(n(n-1)/2) \times (n(n-1)/2)$ weight matrix that contains only the nonzero rows and columns of the full weight matrix $\mathbf{W}$. The second term is a penalty term to fit the diagonal elements of the correlation matrix. The default value is 100. The reciprocal of this value replaces the asymptotic variance corresponding to the diagonal elements of a correlation matrix in the weight matrix $\mathbf{W}$, and it is effective only with the ASYCOV=CORR option. The often used value $r = 1$ seems to be too small in many cases to fit the diagonal elements of a correlation matrix properly. The default WPENALTY= value emphasizes the importance of the fit of the diagonal elements in the correlation matrix. You can decrease or increase the value of r if you want to decrease or increase the importance of the diagonal elements fit. This option is effective only with the WLS or DWLS estimation method and the analysis of a correlation matrix.

See the section "Estimation Criteria" on page 6888 for more details.

CAUTION: If you input the weight matrix by the INWGT= option, the WPENALTY= option is ignored.

WRIDGE=*r*

defines a ridge factor r for the diagonal of the weight matrix $\mathbf{W}$ used in GLS, WLS, or DWLS estimation. The weight matrix $\mathbf{W}$ is transformed to

$$\mathbf{W} \longrightarrow \tilde{\mathbf{W}} = \mathbf{W} + r(\text{diag}(\mathbf{W}))$$

The WRIDGE= option is applied on the weight matrix before the following actions occur:

- the WPENALTY= option is applied on it
- the weight matrix is written to the OUTWGT= data set
- the weight matrix is displayed

CAUTION: If you input the weight matrix by the INWGT= option, the OUTWGT= data set will contain the same weight matrix without the ridging requested by the WRIDGE= option. This ensures that the entries in the INWGT= and OUTWGT= data sets are consistent. The WRIDGE= option is ignored if you input the inverse of the weight matrix by the INWGT= and INWGTINV options (or the INWGT(INV)= option alone).

BASEMODEL Statement

REFMODEL | BASEMODEL *model_number* < / *options* > **;**

where *model_number* represents a model number between 1 and 9999.

See the REFMODEL statement on page 6811 for details.

BOUNDS Statement

BOUNDS *constraint* < , *constraint* . . . > **;**

where *constraint* represents
< *number operator* > *parameter-list* < *operator number* >

You can use the BOUNDS statement to define boundary constraints for any independent parameter that has its name specified in the main or subsidiary model specification statements, the PARAMETERS statement, or the INMODEL= data set. You cannot define boundary constraints for dependent parameters created in SAS programming statements or elsewhere.

Valid operators are <=, <, >=, >, and = (or, equivalently, LE, LT, GE, GT, and EQ). The following is an example of the BOUNDS statement:

```
bounds         0.   <= a1-a9 x    <= 1. ,
              -1.   <= c2-c5             ,
                       b1-b10 y   >= 0. ;
```

You must separate boundary constraints with a comma, and you can specify more than one BOUNDS statement. The feasible region for a parameter is the intersection of all boundary constraints specified for that parameter; if a parameter has a maximum lower boundary constraint larger than its minimum upper bound, the parameter is set equal to the minimum of the upper bounds.

The active set strategies made available in PROC TCALIS treat strict inequality constraints < or > as if they were just inequality constriants <= or >=. For example, if you require x be strictly bigger than zero so as to prevent an undefined value for $y = log(x)$, specifying the following statement is insufficient:

```
BOUNDS x > 0;
```

Specify the following statement instead:

```
BOUNDS x > 1E-8;
```

If the TCALIS procedure encounters negative variance estimates during the minimization process, serious convergence problems can occur. You can use the BOUNDS statement to constrain these parameters to nonnegative values. Although allowing negative values for variances might lead to a better model fit with smaller χ^2 value, it adds difficulties in interpretation.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC TCALIS to obtain separate analyses on observations in groups defined by the BY variables in the input data set. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the TCALIS procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

The BY statement is not supported if you define more than one group by using the GROUP statements.

For more information about the BY statement, refer to the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, refer to the discussion in the *SAS Procedures Guide.*

COV Statement

COV *assignment* < , *assignment* . . . > **;**

where *assignment* represents
variables < ∗ *variables2* > = *parameter-spec*

The COV statement is a subsidiary model specification statement for the FACTOR and LINEQS models. In the LINEQS model, the COV statement defines the covariances among the exogenous variables, including errors and disturbances. In the FACTOR model, the COV statement defines the factor covariances. In each *assignment* of the COV statement, you specify variables in the *variables* and the *variables2* lists, followed by the covariance parameter specification in the *parameter-spec* list. Covariances with only values given are fixed constants. Covariances with names given are free parameters to estimate. You can also specify initial values for these free parameters by putting the parenthesized initial values after the parameter names. The *assignment*s in the COV statement must be separated by commas. You can use only one COV statement in each model specification.

Consider a LINEQS model with exogenous variables V1, V2, V3, and V4. The following is a COV statement specification of the LINEQS model:

```
cov V2 V1 = 0.3,
    V3 V1 = phi1 (0.4),
    V3 V2 = phi2,
    V4 V3 = phi2;
```

In this statement, you specify **cov(V2,V1)** as a fixed constant at 0.3, **cov(V3,V1)** as parameter phi1 with an initial value at 0.4, **cov(V3,V2)** and **cov(V4,V3)** as a free parameter phi2 without an initial value.

Note that the *variables* and *variables2* lists on the left-hand side of the equal sign of the COV statement should contain only names of exogenous variables. In LINEQS models, exogenous variables

can be either observed or latent (including errors and disturbances). However, in the FACTOR models, only covariances among latent factors can be specified.

If the right-hand-side parameter list is longer than the left-hand-side variable list, the right-hand-side list is shortened to the length of the variable list. If the right-hand-side list is shorter than the variable list, the right-hand-side list is filled with repetitions of the last item in the list.

You can use one of two alternatives to specify covariance parameters. The first alternative uses only one variable list and refers to all distinct pairs of variables within the list. The second alternative uses two variable lists separated by an asterisk and refers to all pairs of variables among the two lists.

Within-List Covariances

Using k variable names in the *variables* list on the left-hand side of an equal sign in a COV statement means that the parameter list (*parameter-spec*) on the right-hand side refers to all $k(k-1)/2$ distinct variable pairs. Order is very important. The order relation between the left-hand-side variable pairs and the right-hand-side parameter list is illustrated by the following within-list covariances specification in the COV statement:

```
cov E1-E4 = PHI1-PHI6 ;
```

This specification is equivalent to the following specification:

```
cov E2 E1 = phi1,
    E3 E1 = phi2, E3 E2 = phi3,
    E4 E1 = phi4, E4 E2 = phi5, E4 E3 = phi6;
```

Because **cov(a,b)** is the same as **cov(b,a)**, you need to specify only one of them for a covariance parameter of any two variables.

When you use prefix names on the right-hand sides, you do not have to count the exact number of parameters. For example, the following statement generates distinct parameter names by appending distinct numbers to the prefix phi:

```
cov E1-E4 = phi__ ;
```

Between-List Covariances

Using k_1 and k_2 variable names in the two lists (separated by an asterisk) on the left-hand side of an equal sign in a COV statement means that the covariance parameter list on the right-hand side refers to all $k_1 \times k_2$ distinct variable pairs. Again, order is very important. The order relation between the left-hand-side variable pairs and the right-hand-side parameter list is illustrated by the following between-list covariances specification:

```
cov E1 E2 * E3 E4 = phi1-phi4;
```

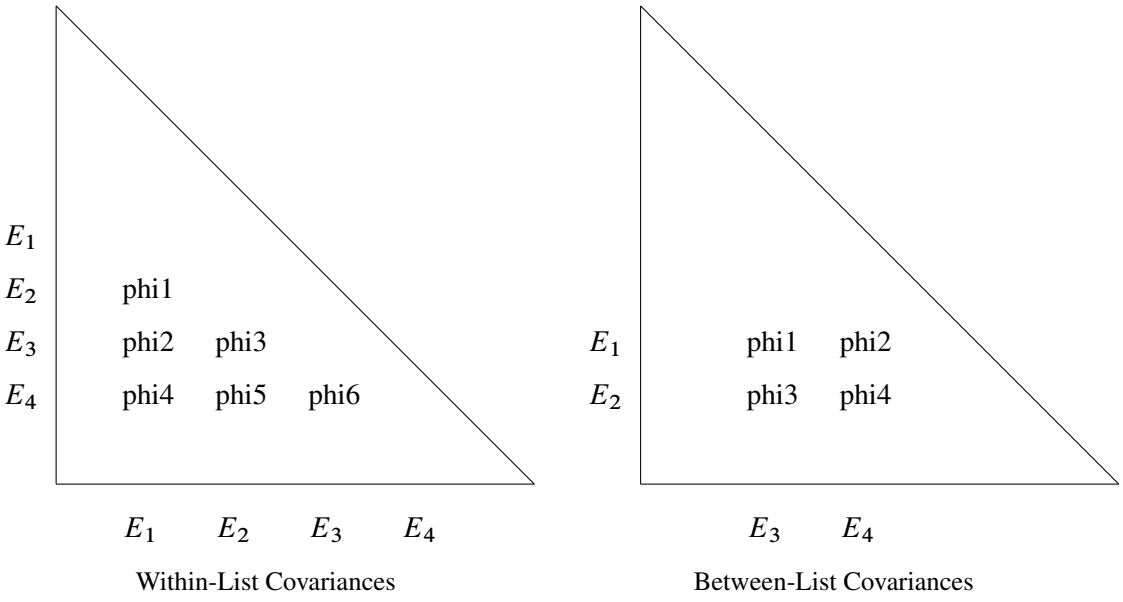
This is equivalent to the following specification:

```
cov  E1 E3 = phi1, E1 E4 = phi2,
     E2 E3 = phi3, E2 E4 = phi4;
```

Using prefix names on the right-hand sides lets you achieve the same purpose without counting the number of parameters. That is, you can specify the between-list covariances equivalently by the following statement:

```
cov  E1 E2 * E3 E4 = phi__ ;
```

Figure 88.3 Within-List and Between-List Covariances



Automatic Covariance Parameters among Exogenous Manifest Variables

In LINEQS models, all covariances among exogenous manifest variables are free parameters to estimate unless you specify otherwise in the COV statement. The parameter names for these covariances are generated by PROC TCALIS with the prefix _Add and appended with integer suffixes. If you want any of these covariances be fixed values, you must specify them explicitly in the COV statement.

Emphasis must be put on the **manifest variables** for automatic covariance parameters generation. Covariaces among exogenous latent variables by default are fixed zeros unless they are specified otherwise in the COV statement. The same applies to any covariance between an exogenous latent variable and an exogenous manifest variable. Only among the set of exogenous mainfest variables are the covariance parameters automatically generated if they are not specified explicitly in the COV statement. This is different than the case of variance parameters. Variances of **all** exogenous variables are automatically generated if they are not explicitly specified in the STD statement.

Modifying a Parameter Specification from a Reference Model

If you define a new LINEQS model by using a reference (or base) model in the REFMODEL statement, in some situations you might want to modify some parameter specifications from the COV statement of the reference model before transferring the specifications to the new model. If you want to change a particular COV parameter specification from the reference model, you can simply put the corresponding parameter location with a new parameter specification in the new model. If you want to delete a particular parameter location from the reference model, you can put that parameter location with a missing parameter value in the new model. For example, if the covariance between variables V1 and V2 are defined in the reference model and you do not want this parameter location in your new model, you can use the following statement to delete the parameter location in the new model:

```
cov  V1 V2 = .;
```

Note that the missing value syntax is valid only when you use with the REFMODEL statement. See the section "Modifying a LINEQS Model from a Reference Model" on page 6769 for a more detailed example of LINEQS model respecification.

In LINEQS models, all covariances among exogenous manifest variables are free parameters to estimate unless you specify otherwise in the COV statement. This is also true for LINEQS models specified under the REFMODEL statement. In the reference model, covariance parameters generated by PROC TCALIS are named with the _Add prefix and appended with integer suffixes. These parameter specifications in the reference model do **not** transfer to the new model. Only after the new model is resolved from the reference model, the REFMODEL statement options, the RENAMEPARM statement, and model respecifications do the remaining unspecified covariances among exogenous manifest variables generate additional covariance parameters with the _Add prefix and integer suffixes. In this way, the covariance parameters in the new model are not constrained to be the same as the corresponding parameters in the reference model. If you want any of these covariance parameters to be constrained across the models, you must specify them explicitly in the COV statement of the reference model so that the covariance specification is transferred to the new model.

DETERM Statement

DETERM | STRUCTEQ *variables* < / *option* > **;**

where *option* represents:
LABEL | NAME = *name*

The DETERM statement is used to compute the determination coefficient of the listed dependent *variables* in the model. The precursor of the DETERM statement is the STRUCTEQ statement, which enables you to define the list of the dependent variables of the structural equations. Because the term *structural equation* is not defined in a unique way, a more generic concept of determination coefficients is revealed by the DETERM statement.

You can specify the DETERM statement as many times as you want for computing determination

coefficients for the sets of dependent *variables* of interest. You can label each set of dependent variables by using the LABEL= option. Note that you cannot use the DETERM statement in an MSTRUCT model because there are no dependent variables in this type of model.

EFFPART Statement

EFFPART *effect* < , *effect*> **;**

where *effect* represents:
variables < *direction variables2* >
and *direction* is the direction of the effect, as indicated by one of the following: –>, >, <–, or <.

In the EFFPART statement, you select those effects you want to analyze by partitioning the total effects into direct and indirect effects, with estimated standard errors. The EFFPART or TOTEFF option of the PROC TCALIS statement also enables you to analyze effects. The difference is that the EFFPART or TOTEFF option displays effects on *all* endogenous variables, while the EFFPART statement shows only the effects of interest. In addition, the EFFPART statement enables you to arrange the effects in any way you like. Hence, the EFFPART statement offers a more precise and organized way to present various results of effects.

The EFFPART statement supports the following three types of effect specifications:

- > or –> direction
  Example:

  ```
  effpart X1 X3-X5 -> Y1 Y2;
  ```

  This will display *four* separate tables, respectively for the effects of X1, X3, X4, and X5 on Y1 and Y2. Each table contains the total, direct, and indirect effects of an X-variable on the two Y-variables.

- < or <– direction
  Example:

  ```
  effpart Y1 Y2 <- X1 X3-X5;
  ```

  This will display *two* separate tables, respectively for the effects on Y1 and Y2, by X1, X3, X4, and X5. Each table contains the total, direct, and indirect effects of the four X-variables on a Y-variable. Certainly, the results produced from this statement are essentially the same as the previous statement. The difference is about the organization of the effects in the tables.

- no direction
  Example:

  ```
  effpart Y1 Y2 X1-X3;
  ```

  In this case, variables on the list are analyzed one by one to determine the nature of the effects. If a variable has nonzero effects on any other variables in the model, a table of the total, direct, and indirect effects of the variable on those variables is displayed. If a variable is

endongenous, a table of total, direct, and indirect effects of those variables that have nonzero effects on the variable is displayed. Note that an endogenous variable in a model might also have effects on other endogenous variables. Therefore, the two cases mentioned are not mutually exclusive—a variable listed in the EFFPART statement might yield two tables for effect analysis.

FACTOR Statement

FACTOR < *EFA_options* | *CFA_spec* > **;**

where *EFA_options* are exploratory factor analysis options, *CFA_spec* is confirmatory factor analysis specification defined as:

factor-variables-relation < , *factor-variables-relation* > ...

and each *factor-variables-relation* is defined as:

factor –> *variables* = *parameter-spec*

In the FACTOR statement, you can specify either *EFA_options*, *CFA_spec*, or neither of these. However, you cannot specify both *EFA_options* and *CFA_spec* at the same time.

If an *EFA_option* (exploratory factor analysis option) is specified immediately after the FACTOR statement, or neither *EFA_options* nor *factor-variable-relations* are specified in the FACTOR statement, an exploratory factor model is analyzed. Otherwise, specifying an *CFA_spec* (confirmatory factor analysis specification) invokes a confirmatory factor model. These two types of models are discussed in the next two sections.

Exploratory Factor Analysis

For the exploratory factor model with orthogonal factors, PROC TCALIS assumes the following model structures for the population covariance or correlation matrix $\mathbf{C}$:

$$\mathbf{C} = \mathbf{F}\mathbf{F}' + \mathbf{U}$$

where $\mathbf{F}$ is the factor loading matrix and $\mathbf{U}$ is a diagonal matrix of error variances. In this section, p denotes the number of manifest variables corresponding to the rows and columns of matrix $\mathbf{C}$, and n denotes the number of factors (or components, if the COMPONENT option is specified in the FACTOR statement) corresponding to the columns of the factor loading matrix $\mathbf{F}$. While the number of manifest variables is set automatically by the number of variables in the VAR statement or in the input data set, the number of factors can be set by the N= option in the FACTOR statement.

The unrestricted exploratory factor model is not identified because any orthogonal rotated factor loading matrix $\tilde{\mathbf{F}} = \mathbf{F}\boldsymbol{\Theta}$ satisfies the same model structures as $\mathbf{F}$ does, where $\boldsymbol{\Theta}$ is any orthogonal matrix so that $\boldsymbol{\Theta}'\boldsymbol{\Theta} = \boldsymbol{\Theta}\boldsymbol{\Theta}' = \mathbf{I}$. Mathematically, the covariance or correlation structures can be expressed as:

$$\mathbf{C} = \tilde{\mathbf{F}}\tilde{\mathbf{F}}' + \mathbf{U} = \mathbf{F}\boldsymbol{\Theta}\,\boldsymbol{\Theta}'\mathbf{F}' + \mathbf{U} = \mathbf{F}\mathbf{F}' + \mathbf{U}$$

To obtain an identified orthogonal factor solution as a starting point, the $n(n-1)/2$ elements in the upper triangle of $\mathbf{F}$ are constrained to zeros in PROC TCALIS. Initial estimates for factor loadings and unique variances are computed by an algebraic method of approximate factor analysis. Given the initial estimates, final estimates are obtained through the iterative optimization of an objective function, which depends on the estimation method specified in the METHOD= option (default with ML—maximum likelihood) of the PROC TCALIS statement.

To make the factor solution more interpretable, you can use the ROTATE= option in the FACTOR statement to obtain a rotated factor loading matrix with a "simple" pattern. Rotation can be orthogonal or oblique. The rotated factors remain uncorrelated after an orthogonal rotation but would be correlated after an oblique rotation. The model structures of an oblique solution are expressed in the following equation:

$$\mathbf{C} = \tilde{\mathbf{F}}\mathbf{P}\tilde{\mathbf{F}}' + \mathbf{U}$$

where $\tilde{\mathbf{F}}$ is the rotated factor loading matrix and $\mathbf{P}$ is a symmetric matrix for factor correlations. See the sections "The FACTOR Model" on page 6863 and "Exploratory Factor Analysis Models" on page 6865 for more details about exploratory factor models.

You can also do exploratory factor analysis by the more dedicated FACTOR procedure. Even though extensive comparisons of the factor analysis capabilities between the FACTOR and TCALIS procedures are not attempted here, some general points can be made here. In general, the FACTOR procedure provides more factor analysis options than the TCALIS procedure does, although both procedures have some unique factor analysis features that are not shared by the other. PROC TCALIS requires more computing time and memory than PROC FACTOR because it is designed for more general structural estimation problems and is not able to exploit all the special properties of the unconstrained factor analysis model. For maximum likelihood analysis, you can use either PROC FACTOR (with METHOD=ML, which is not the default method in PROC FACTOR) or PROC TCALIS. Because the initial unrotated factor solution obtained by PROC FACTOR uses a different set of identification constraints than that of PROC TCALIS, you would observe different initial ML factor solutions for the procedures. Nonetheless, the initial solutions by both procedures are statistically equivalent.

The following exploratory factor analysis options are available in the FACTOR statement.

COMPONENT | COMP

computes a component analysis instead of a factor analysis (the diagonal matrix $\mathbf{U}$ in the model is set to 0). Note that the rank of $\mathbf{FF}'$ is equal to the number n of components in $\mathbf{F}$. If n is smaller than the number of variables in the moment matrix $\mathbf{C}$, the matrix of predicted model values is singular and maximum likelihood estimates for $\mathbf{F}$ cannot be computed. You should compute ULS estimates in this case.

HEYWOOD | HEY

constrains the diagonal elements of $\mathbf{U}$ to be nonnegative. Equivalently, you can constrain these elements to positive values by the BOUNDS statement.

GAMMA=*p*

specifies the orthomax weight used with the option ROTATE=ORTHOMAX. Alternatively, you can use ROTATE=ORTHOMAX(*p*) with *p* representing the orthomax weight. There is no restriction on valid values for the orthomax weight, although the most common values are

between 0 and the number of variables. The default GAMMA= value is one, resulting in the varimax rotation.

N=*n*

specifies the number of first-order factors or components. The number of factors (n) should not exceed the number of manifest variables (p) in the analysis. For the saturated model with $n = p$, the COMP option should generally be specified for $\mathbf{U} = 0$; otherwise, $df < 0$. For $n = 0$ no factor loadings are estimated, and the model is $\mathbf{C} = \mathbf{U}$, with $\mathbf{U} = diag$. By default, $n = 1$.

NORM< = KAISER | NONE >

Kaiser-normalizes the rows of the factor pattern for rotation. NORM=KAISER, which is the default, has exactly the same effect as NORM. You can turn off the normalization by NORM=NONE.

RCONVERGE=*p*

RCONV=*p*

specifies the convergence criterion for rotation cycles. Rotation stops when the scaled change of the simplicity function value is less than the RCONVERGE= value. The default convergence criterion is:

$$|f_{new} - f_{old}|/K < \epsilon$$

where f_{new} and f_{old} are simplicity function values of the current cycle and the previous cycle, respectively, $K = max(1, |f_{old}|)$ is a scaling factor, and ϵ is 1E–9 by default and is modified by the RCONVERGE= value.

RITER=*i*

specifies the maximum number of cycles i for factor rotation. The default i is the greater of 10 times the number of variables and 100.

ROTATE | R=*name*

specifies an orthogonal or oblique rotation of the initial factor solution. Although ROTATE=PRINCIPAL is actually not a rotation method, it is put here for convenience. By default, ROTATE=NONE.

Valid *names* for orthogonal rotations are as follows:

BIQUARTIMAX | BIQMAX specifies orthogonal biquartimax rotation. This corresponds to the specification ROTATE=ORTHOMAX(0.5).

EQUAMAX | E specifies orthogonal equamax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=*n*/2.

FACTORPARSIMAX | FPA specifies orthogonal factor parsimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=*n*.

NONE | N specifies that no rotation be performed, leaving the original orthogonal solution.

ORTHCF(*p1*,*p2*) | ORCF(*p1*,*p2*) specifies the orthogonal Crawford-Ferguson rotation (Crawford and Ferguson 1970) with the weights *p1* and *p2* for variable parsimony and factor parsimony, respectively. See the definitions of weights in Chapter 33, "The FACTOR Procedure."

ORTHGENCF(*p1*,*p2*,*p3*,*p4*) | ORGENCF(*p1*,*p2*,*p3*,*p4*) specifies the orthogonal generalized Crawford-Ferguson rotation (Jennrich 1973), with the four weights *p1*, *p2*, *p3*, and *p4*. For the definitions of these weights, see the section "Simplicity Functions for Rotations" on page 1572 in Chapter 33, "The FACTOR Procedure."

ORTHOMAX<(*p1*)> | ORMAX<(*p1*)> specifies the orthomax rotation (see Harman 1976) with orthomax weight *p1*. If ROTATE=ORTHOMAX is used, the default *p1* value is 1 unless specified otherwise in the GAMMA= option. Alternatively, ROTATE=ORTHOMAX(*p1*) specifies *p1* as the orthomax weight or the GAMMA= value. For the definitions of the orthomax weight, see the section "Simplicity Functions for Rotations" on page 1572 in Chapter 33, "The FACTOR Procedure."

PARSIMAX | PA specifies orthogonal parsimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with

$$\text{GAMMA} = \frac{p \times (n-1)}{p+n-2}$$

PRINCIPAL | PC specifies a principal axis rotation. If ROTATE=PRINCIPAL is used with a factor rather than a component model, the following rotation is performed:

$$\mathbf{F}_{new} = \mathbf{F}_{old}\mathbf{T}, \quad \text{with} \quad \mathbf{F}'_{old}\mathbf{F}_{old} = \mathbf{T}\mathbf{\Lambda}\mathbf{T}'$$

where the columns of matrix $\mathbf{T}$ contain the eigenvectors of $\mathbf{F}'_{old}\mathbf{F}_{old}$.

QUARTIMAX | QMAX | Q specifies orthogonal quartimax rotation. This corresponds to the specification ROTATE=ORTHOMAX(0).

VARIMAX | V specifies orthogonal varimax rotation. This corresponds to the specification ROTATE=ORTHOMAX with GAMMA=1.

Valid *name*s for oblique rotations are as follows:

BIQUARTIMIN | BIQMIN specifies biquartimin rotation. It corresponds to the specification ROTATE=OBLIMIN(.5) or ROTATE=OBLIMIN with TAU=.5.

COVARIMIN | CVMIN specifies covarimin rotation. It corresponds to the specification ROTATE=OBLIMIN(1) or ROTATE=OBLIMIN with TAU=1.

OBBIQUARTIMAX | OBIQMAX specifies oblique biquartimax rotation.

OBEQUAMAX | OE specifies oblique equamax rotation.

OBFACTORPARSIMAX | OFPA specifies oblique factor parsimax rotation.

OBLICF(*p1*,*p2*) | OBCF(*p1*,*p2*) specifies the oblique Crawford-Ferguson rotation (Crawford and Ferguson 1970) with the weights *p1* and *p2* for variable parsimony and factor parsimony, respectively. For the definitions of these weights, see the section "Simplicity Functions for Rotations" on page 1572 in Chapter 33, "The FACTOR Procedure."

OBLIGENCF(*p1*,*p2*,*p3*,*p4*) | OBGENCF(*p1*,*p2*,*p3*,*p4*) specifies the oblique generalized Crawford-Ferguson rotation (Jennrich 1973) with the four weights *p1*, *p2*, *p3*, and *p4*. For the definitions of these weights, see the section "Simplicity Functions for Rotations" on page 1572 in Chapter 33, "The FACTOR Procedure."

OBLIMIN<(*p1*)> | OBMIN<(*p1*)> specifies the oblimin rotation with oblimin weight *p1*. If ROTATE=OBLIMIN is used, the default *p1* value is zero unless specified otherwise in the TAU= option. Alternatively, ROTATE=OBLIMIN(*p1*) specifies *p1* as the oblimin weight or the TAU= value. For the definitions of the oblimin weight, see the section "Simplicity Functions for Rotations" on page 1572 in Chapter 33, "The FACTOR Procedure."

OBPARSIMAX | OPA specifies oblique parsimax rotation.

OBQUARTIMAX | OQMAX specifies oblique quartimax rotation. This is the same as the QUARTIMIN method.

OBVARIMAX | OV specifies oblique varimax rotation.

QUARTIMIN | QMIN specifies quartimin rotation. It is the same as the oblique quartimax method. It also corresponds to the specification ROTATE=OBLIMIN(0) or ROTATE=OBLIMIN with TAU=0.

TAU=*p*

specifies the oblimin weight used with the option ROTATE=OBLIMIN. Alternatively, you can use ROTATE=OBLIMIN(*p*) with *p* representing the oblimin weight. There is no restriction on valid values for the oblimin weight, although for practical purposes a negative or zero value is recommended. The default TAU= value is 0, resulting in the quartimin rotation.

Confirmatory Factor Analysis

FACTOR *factor-variable-relations* **;**
PVAR *partial variance parameters* **;**
COV *covariance parameters* **;**
MEAN *mean parameters* **;**

If you specify at least one *factor-variable-relation*, a confirmatory factor analysis is assumed. Whenever applicable, you can use the PVAR, COV, and MEAN statements to specify the variance, partial variance, covariance, and mean parameters. The model structures for the covariance matrix **C** of the confirmatory factor model are described in the following equation:

$$\mathbf{C} = \mathbf{FPF}' + \mathbf{U}$$

where **F** is the factor loading matrix, **P** is a symmetric matrix for factor correlations, and **U** is a diagonal matrix of error variances.

If the mean structures are also analyzed, the model structures for the mean vector $\boldsymbol{\mu}$ of the confirmatory factor model are described in the following equation:

$$\boldsymbol{\mu} = \boldsymbol{\alpha} + \mathbf{F}\boldsymbol{\nu}$$

where $\boldsymbol{\alpha}$ is the intercept vector for the observed variables and $\boldsymbol{\nu}$ is the vector for factor means. See the sections "The FACTOR Model" on page 6863 and "Confirmatory Factor-Analysis Models" on page 6866 for more details about confirmatory factor models.

You can use the PVAR statement to specify factor variance parameters in the diagonal of matrix **P**, or the error variances in the diagonal of matrix **U**. These variances, by default, are automatically added as free parameters in the factor model even if you do not specify them. However, if you want to set constraints on these parameters, it is useful to define them explicitly by using the PVAR statement. For example, you might like to set some variances to fixed constants, or you might want to set equality constraints by using the same parameter name at different parameter locations in your model.

You can use the COV statement to specify factor covariance parameters. These factor covariances are the off-diagonal elements of matrix **P**. Unlike the factor variances, factor covariances are assumed to be fixed zeros by default. Note that you cannot use the COV statement to specify the error covariances—they are always fixed zeros in the confirmatory factor-analysis model.

You can use the MEAN statement to specify the means of the factors in vector $\boldsymbol{\nu}$ and the intercepts of the manifest variables in vector $\boldsymbol{\alpha}$. All the factor means and intercepts are fixed zeros if you do not specify them explicitly.

Suppose that you want to fit a factor model with nine manifest variables V1-V9. In this factor model, you assume a general factor and three group-factors, with three variables defining each group-factor. The following FACTOR statement shows the specification of such a model:

```
factor
   g_factor -> V1-V9 (9 * g_load__),
   factor_a -> V1-V3 (3 * load_a__),
   factor_b -> V4-V6 (3 * load_b__),
   factor_c -> V7-V9 (3 * load_c__);
```

In this example, you are fitting a factor matrix of the following form:

	g_factor	factor_a	factor_b	factor_c
V1	g_load1	load_a1		
V2	g_load2	load_a2		
V3	g_load3	load_a3		
V4	g_load4		load_b1	
V5	g_load5		load_b2	
V6	g_load6		load_b3	
V7	g_load7			load_c1
V8	g_load8			load_c2
V9	g_load9			load_c3

The blanks in the factor matrix denotes fixed zeros. Because you do not specify the covariances by using the COV statement, g_factor, factor_a, factor_b, and factor_c are all uncorrelated in the model.

Modifying a FACTOR Model from a Reference Model

In this section, it is assumed that you use a REFMODEL statement within the scope of a MODEL statement and the reference model (or base model) is a factor model, either exploratory or confirmatory. The reference model is referred to as the old model, while the model makes reference to this old model is referred to as the new model. If the new model is not intended to be an exact copy of the old FACTOR model, you can use the following extended FACTOR modeling language to make modifications from the old model before transferring the specifications to the new model.

Before discussing the syntax for modifying factor models, there are two cases where using the REFMODEL statement for defining new factor models are not recommended. First, if your old model is an exploratory factor-analysis model, then specification by using the FACTOR modeling language in the new model will replace the old model completely. In this case, the use of the REFMODEL statement is superfluous and should be avoided. Second, if your old model is a confirmatory factor-analysis model, then specification of an exploratory factor model by using the FACTOR statement in the new model will also replace the old model completely. Again, in this case the use of the REFMODEL statement is superfluous and should be avoided.

The nontrivial case where you might find the REFMODEL statement useful is when you modify an old confirmatory factor model to form a new confirmatory factor model. This nontrivial case is the focus of discussion in the remaining of the section.

The extended FACTOR modeling language for modifying model specification bears the same syntax as that of the ordinary FACTOR modeling language (see the section "Confirmatory Factor Analysis" on page 6756). The syntax is:

FACTOR *factor-variable-relation* **;**
PVAR *partial variance parameters* **;**
COV *covariance parameters* **;**
MEAN *mean parameters* **;**

The new model is formed by integrating with the old model in the following ways:

Duplication: If you do not specify in the new model a parameter location that exists in the old model, the old parameter specification is duplicated in the new model.

Addition: If you specify in the new model a parameter location that does not exist in the old model, the new parameter specification is added in the new model.

Deletion: If you specify in the new model a parameter location that also exists in the old model and the new parameter is denoted by the missing value '.', the old parameter specification is not copied into the new model.

Replacement: If you specify in the new model a parameter location that also exists in the old model and the new parameter is not denoted by the missing value '.', the new parameter specification replaces the old one in the new model.

For example, consider the following two-group analysis:

```
proc tcalis;
   group 1 / data=d1;
   group 2 / data=d2;
   model 1 / group=1;
      factor
         F1 -> V1-V3 = 1. load1 load2,
         F2 -> V4-V6 = 1. load3 load4,
         F3 -> V7-V9 = 1. load5 load6;
      cov
         F1 F2 = c12,
         F2 F3 = c23;
      pvar
         F1-F3 = c1-c3,
         V1-V9 = ev1-ev9;
   model 2 / group=2;
      refmodel 1;
      factor
         F1 -> V1 = loada,
         F2 -> V4 = loadb,
         F3 -> V7 = loadc;
      cov
         F1 F2 = .,
         F1 F3 = c13;
   run;
```

In this specification, you specify model 2 by referring to model 1 in the REFMODEL statement; model 2 is the new model which refers to the old model, model 1. Because the PVAR statement is not used in new model, all variance and partial variance parameter specifications in the PVAR statement of the old model are duplicated in the new model. The covariance parameter c23 for covariance between F2 and F3 in the COV statement of the old model is also duplicated in the new model. Similarly, loading parameters load1–load6 for some specific factor matrix locations are duplicated from the old model to the new model.

In the new model, there is an additional parameter specification that does not exist in the old model. In the COV statement of the new model, covariance parameter c13 for the covariance between F1 and F3 is added.

In the same statement, the covariance between F1 and F2 is denoted by the missing value '.'. This means that this parameter location in the old model should not be included in the new model. The consequence of this deletion from the old model is that the covariance between F1 and F2 is a fixed zero in the new model.

Finally, the three new loading specifications in the FACTOR statement of the new model replaces the fixed ones in the old model. They are now free parameters loada, loadb, and loadc in the new model.

FITINDEX Statement

FITINDEX *option* < *option* . . . > **;**

You can use the FITINDEX statement to set the options for computing and displaying the fit indices, or to output the fit indices. All but the OFF= and ON= options of the FITINDEX statement are also available in the PROC TCALIS statement. The options set in the FITINDEX statement will overwrite those set in the PROC TCALIS statement.

For the listing of fit indices and their definitions, see the section "Overall Model Fit Indices" on page 6903. Note that not all fit indices are available with all estimation methods, which is specified by the METHOD= option of the PROC TCALIS statement. See the section "Fit Indices and Estimation Methods" on page 6910 for more details.

The options of the FITINDEX statement are as follows:

ALPHAECV=α

specifies a $(1-\alpha)100\%$ confidence interval ($0 \leq \alpha \leq 1$) for the Browne and Cudeck (1993) expected cross validation index (ECVI). See the ALPHAECV= option of the PROC TCALIS statement on page 6726.

ALPHARMS=α

specifies a $(1-\alpha)100\%$ confidence interval ($0 \leq \alpha \leq 1$) for the Steiger and Lind (1980) root mean square error of approximation (RMSEA) coefficient. See the ALPHARMS= option of the PROC TCALIS statement on page 6726.

CHICORRECT | CHICORR = *name* **|** *c*

specifies a correction factor c for the chi-square statistics for model fit. See the CHICORRECT= option of the PROC TCALIS statement on page 6727.

CLOSEFIT=*p*

defines the criterion value p for indicating a close fit. See the CLOSEFIT= option of the PROC TCALIS statement on page 6728.

DFREDUCE=*i*

reduces the degrees of freedom of the χ^2 test by i. See the DFREDUCE= option of the PROC TCALIS statement on page 6728.

NOADJDF

turns off the automatic adjustment of degrees of freedom when there are active constraints in the analysis. See the NOADJDF option of the PROC TCALIS statement on page 6735.

NOINDEXTYPE

disables the display of index types in the fit summary table. See the NOINDEXTYPE option of the PROC TCALIS statement on page 6735.

OFF | OFFLIST= [*names***] | {***names***}**

turns off the printing of one or more fit indices or modeling information as indicated by *names*, where a *name* represents a fit index, a group of fit indices, or modeling information.

Names must be specified inside a pair of parentheses and separated by spaces. By default, all fit indices are printed. See the ON= option for the value of *names*.

ON | ONLIST < (ONLY) > = [*names***] | {***names***}**

turns on the printing of one or more fit indices or modeling information as indicated by *names*, where a *name* represents a fit index, a group of fit indices, or modeling information. *Names* must be specified inside a pair of parentheses and separated by spaces. Because all fit indices and modeling information are printed by default, using an ON= list alone is redundant. When both ON= and OFF= lists are specified, the ON= list will override the OFF= list for those fit indices or modeling information that appear on both lists. If an ON(ONLY)= list is used, only those fit indices or modeling information specified in the list will be printed. Effectively, an ON(ONLY)= list is the same as the specification with an ON= list with the same selections and an OFF=ALL list in the FITINDEX statement.

Output Control of Fit Index Groups and Modeling Information Group

You can use the following *names* to refer to the groups of fit indices or modeling information available in PROC TCALIS:

ABSOLUTE	Absolute or stand-alone fit indices that measures the model fit without using a baseline model.
ALL	All fit indices available in PROC TCALIS.
INCREMENTAL	Incremental fit indices that measure model fit by comparing with a baseline model.
MODELINFO	General modeling information including sample size, number of variables, number of variables, and so on.
PARSIMONY	Fit indices that take model parsimony into account.

Output Control of Modeling Information

You can use the following *names* to refer to the individual modeling information available in PROC TCALIS:

INDCHI	Chi-square statistic for the independence model.
INDCHIDF	Degrees of freedom of the chi-square statistic for the independence model.
NACTCON	Number of active constraints.
NMOMENTS	Number of elements in the moment matrices being modeled.
NOBS	Number of observations assumed in the analysis.
NPARMS	Number of independent parameters.
NVAR	Number of variables.

Output Control of Absolute Fit Indices
You can use the following *names* to refer to the individual absolute fit indices available in PROC TCALIS:

CHISQ	Chi-square statistic for model fit.
CN	Hoelter's critical N.
CONTRIBUTION	Percentage contribution to the chi-square value for multiple-group analyses.
DF	Degrees of freedom for the chi-square test for model fit.
ELLIPTIC	Elliptical chi-square statistic for ML and GLS methods in single-group analyses without mean structures.
FUNCVAL	Optimized function value.
GFI	Goodness-of-fit index by Jöreskog and Sörbom.
PROBCHI	P-value of the chi-square statistic for model fit.
PROBELLIPTIC	P-value of the elliptical chi-square statistic.
RMSR	Root mean square residual.
SRMSR	Standardized root mean square residual.
ZTEST	Z-test of Wilson and Hilferty.

Output Control of Parsimonious Fit Indices
You can use the following *names* to refer to the individual parsimonious fit indices available in PROC TCALIS:

AGFI	Adjusted GFI.
AIC	Akaike information criterion.
CAIC	Bozdogan corrected AIC.
CENTRALITY	McDonald centrality measure.
ECVI	Expected cross-validation index.
LL_ECVI	Lower confidence limit for ECVI.
LL_RMSEA	Lower confidence limit for RMSEA.
PGFI	Parsimonious GFI.
PROBCLFIT	Probability of close fit.
RMSEA	Root mean squares of error approximation.
SBC	Schwarz Bayesian criterion.
UL_ECVI	Upper confidence limit for ECVI .
UL_RMSEA	Upper confidence limit for RMSEA.

Output Control of Incremental Fit Indices
You can use the following *names* to refer to the individual incremental fit indices available in PROC TCALIS:

BENTLERCFI	Bentler comparative fit index.
BENTLERNFI	Bentler-Bonett normed fit index.
BENTLERNNFI	Bentler-Bonett non-normed fit index.
BOLLENNFI	Bollen normed fit index (Rho1).
BOLLENNNFI	Bollen non-normed fit index (Delta2).
PNFI	James et al. parsimonious normed fit index.

OUTFIT=*SAS-data-set*
creates an output data set containing the values of the fit indices. This is the same as the OUTFIT= option of the PROC TCALIS statement on page 6739. See the section "OUTFIT= SAS-data-set" on page 6837 for details.

FREQ Statement

FREQ *variable* **;**

If one variable in your data set represents the frequency of occurrence for the other values in the observation, specify the variable's name in a FREQ statement. PROC TCALIS then treats the data set as if each observation appears n_i times, where n_i is the value of the FREQ variable for observation i. Only the integer portion of the value is used. If the value of the FREQ variable is less than 1 or is missing, that observation is not included in the analysis. The total number of observations is considered to be the sum of the FREQ values. You can use only one FREQ statement within the scope of each GROUP or the PROC TCALIS statement.

GROUP Statement

GROUP *i* < / *options* > **;**

where *i* is an assigned group number between 1 and 9999, inclusively.

The GROUP statement signifies the beginning of a group specification block and designates a group number for the group. All subsidiary group specification statements after a GROUP statement belong in that group until another MODEL or GROUP statement is used. The subsidiary group specification statements refer to one of the following four statements:

- FREQ statement on page 6763
- PARTIAL statement on page 6798
- VAR statement on page 6818
- WEIGHT statement on page 6818

For example, consider the following statements:

```
proc tcalis;
   var X1-X4;
   group 1 / label='Women' data=women_data;
      freq  Z;
   group 2 / label='Men' data=men_data;
      partial P;
   model 1 / group = 1-2;
      factor N=1; /* One factor exploratory factor analysis */
   run;
```

In the GROUP statements, two groups are defined. Group 1, labeled as 'Women', refers to the data set women_data. Group 2, labeled as 'Men', refers to the data set men_data. Both groups are fitted by an exploratory factor model defined in model 1, as indicated in the GROUP= option of the MODEL statement. While the frequency variable Z defined in the FREQ statement is applicable only to group 1, the partial variable P defined in the PARTIAL statement is applicable only to group 2. However, the VAR statement, which appears before the definitions of both groups, applies globally to both group 1 and group 2. Therefore, variables X1–X4 are the analysis variables in the two groups.

You can set group-specific *options* in each GROUP statement. All but one (that is, the LABEL= option) of these *options* are also available in the MODEL and PROC TCALIS statements. If you set these group-specific *options* in the PROC TCALIS statement, they will apply to all groups unless you respecify them in the GROUP statement. If you set these group-specific *options* in the MODEL statement, they will apply to all groups that are fitted by the associated model. In general, the group-specific options are transfered from the PROC TCALIS statement to the MODEL statements (if present) and then to the fitted groups. In the transferring process, options are overwritten by the newer ones. If you want to apply some group-specific options to a particular group only, you should set those options in the GROUP statement corresponding to that group.

Option Available in the GROUP Statement Only

LABEL | NAME=*name*

specifies a label for the current group. You can use any valid SAS names up to 256 characters for labels. You can also use quote strings for the labels. This option can be specified only in the GROUP statement, not the PROC TCALIS statement.

Options Available in the GROUP and PROC TCALIS Statements

These options are available in the GROUP and PROC TCALIS statements:

Option	Description
DATA= on page 6728	specifies the input data set
INWGT= on page 6731	specifies the data set that contains the weight matrix
OUTSTAT= on page 6739	specifies the data set for storing the statistical results
OUTWGT= on page 6739	specifies the data set for storing the weight matrix

See the section "Listing of PROC TCALIS Statement Options" on page 6726 for more details about these options. If you specify these options in the PROC TCALIS statement, they are transferred to *all* GROUP statements. They might be overwritten by the respecifications in the individual GROUP statements.

Options Available in GROUP, MODEL, and PROC TCALIS Statements

These options are available in the GROUP, MODEL and PROC TCALIS statements:

Option	Description
BIASKUR on page 6727	computes the skewness and kurtosis without bias corrections
EDF= on page 6729	defines nobs by the number of error df
KURTOSIS on page 6731	computes and displays kurtosis
NOBS= on page 6735	defines the number of observations (nobs)
PCORR on page 6740	displays analyzed and estimated moment matrix
PLOTS= on page 6741	specifies ODS Graphics selection
PWEIGHT on page 6818	displays the weight matrix
RDF | DFR= on page 6742	defines nobs by the number of regression df
RESIDUAL | RES on page 6742	computes the default residuals
RESIDUAL | RES= on page 6742	specifies the type of residuals
RIDGE on page 6743	specifies the ridge factor for covariance matrix
SIMPLE on page 6743	prints univariate statistics
VARDEF= on page 6744	specifies variance divisor
WPENALTY= on page 6744	specifies the penalty weight to fit correlations
WRIDGE= on page 6745	specifies the ridge factor for the weight matrix

If you specify these options in the PROC TCALIS statement, they are transferred to all MODEL statements. These options are overwritten by the respecifications in the individual MODEL statements. After these options are resolved in a given MODEL statement, they are transferred further to the GROUP statements of which the associated groups are fitted by the model. Again, these options might be overwritten by the respecifications in the individual GROUP statements.

LINCON Statement

LINCON *constraint* < , *constraint* . . . > **;**

where *constraint* represents one of the following:

- *number operator linear-term*
- *linear-term operator number*

and *linear-term* is

< + | − >< *coefficient* ∗ > *parameter* << + | − >< *coefficient* ∗ > *parameter* . . . >

The LINCON statement specifies a set of linear equality or inequality constraints of the following form:

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 1, \ldots, m$$

The constraints must be separated by commas. Each linear constraint i in the statement consists of a linear combination $\sum_j a_{ij} x_j$ of a subset of the n parameters x_j, $j = 1, \ldots, n$, and a constant value b_i separated by a comparison operator. Valid operators are <=, <, >=, >, and = or, equivalently, LE, LT, GE, GT, and EQ. PROC TCALIS cannot enforce the strict inequalities < or >. Note that the coefficients a_{ij} in the linear combination must be constant numbers and must be followed by an asterisk and the name of a parameter (that is, listed in the PARMS, main, or subsidiary model specification statements). The following is an example of the LINCON statement that sets a linear constraint on parameters x1 and x2:

```
lincon       x1 + 3 * x2 <= 1;
```

Although you can easily express boundary constraints in LINCON statements, for many applications it is much more convenient to specify both the BOUNDS and the LINCON statements in the same PROC TCALIS call.

LINEQS Statement

LINEQS *equation* < , *equation* . . . > **;**

where *equation* represents:

dependent = *term* < + *term* . . . >

and each *term* represents one of the following:

- *coefficient-name* < *(number)* > *variable-name*
- *prefix-name* < *(number)* > *variable-name*
- < *number* > *variable-name*

The LINEQS statement is a main model specification statement that invokes the LINEQS modeling language. You can specify at most one LINEQS statement in a model, within the scope of either the PROC TCALIS statement or a MODEL statement. To completely specify a LINEQS model, you

might need to add some subsidiary model specification statements. The following is the syntax for the LINEQS modeling language:

LINEQS *model equations* **;**
STD *partial variance parameters* **;**
COV *covariance parameters* **;**
MEAN *mean parameters* **;**

In the LINEQS statement, you use equations to specify the linear functional relations among manifest or latent variables. Equations in the LINEQS statement are separated by commas.

In the STD statement, you specify the variance parameters. In the COV statement, you specify the covariance parameters. In the MEAN statement, you specify the mean parameters. For details of these subsidiary model specification statements, see the syntax of these statements.

In the LINEQS statement, in addition to the functional relations among variables, you specify the coefficient parameters of interest in the equations. There are four types of parameters you can specify in equations, as shown in the following example:

```
lineqs
   V1 = 1.       F1 + E1,
   V2 = b2       F1 + E2,
   V3 = b2       F1 + E3,
   V4 = b4 (.4) F1 + E4;
```

In this example, you have manifest variables V1–V4, which are related to a latent factor, denoted by F1, as specified in the equations. In each equation, you have one outcome variable (V-variable), one predictor variable (F1), and one error variable (E-variable). The following four types of parameters have been specified:

- free or constrained parameters with starting values

  The coefficient parameter b4 in the fourth equation is a free parameter to estimate. A starting value for a parameter can be specified inside parentheses following the parameter name. In this case, the starting value for b4 is 0.4. As shown in the equations, it is not constrained with any other coefficient parameters because b4 is specified at only one location.

- free or constrained parameters without starting values

  The coefficient parameter b2 in the second and the third equations is a parameter to estimate. No starting value is given for the parameter. With the same parameter name b2, the path coefficients with predictor variable F1 in the second and the third equation are thus implicitly constrained to be equal.

- explicitly fixed parameter values

  Any fixed constant preceding a predictor variable is a fixed parameter (for example, the constant coefficient 1 for F1 in the first equation).

- implicitly fixed parameter values

  If there is neither a parameter nor a constant preceding a predictor variable in an equation, the coefficient associated with that predictor variable is assumed to be a fixed parameter at one.

For example, the coefficients preceding all error variables (E-variables) are fixed at 1. Any coefficients associated with unspecified predictor variables in equations are fixed zeros.

Parameters with no starting values will be initialized by various heuristic and effective methods in PROC TCALIS. See the section "Initial Estimates" on page 6922 for details.

If your model contains many unconstrained parameters and it is too cumbersome to find different parameter names, you can specify all those parameters by the same prefix name. A prefix name is a short name called "root" followed by double underscores '__'. Whenever a prefix name is encountered, the TCALIS procedure generates a parameter name by appending a unique integer to the root. Hence, the prefix name should have few characters so that the generated parameter name is not longer than thirty-two characters. To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names.

Certainly, coefficient parameters are only a subset of parameters of a model. Other parameters like variance, covariance and mean parameters should be specified in the subsidiary model statements for the LINEQS modeling language—that is, the COV, MEAN, and STD statements.

Representing Latent Variables in the LINEQS Model

Because latent variables are widely used in structural equation modeling, PROC TCALIS needs a way to identify different types of latent variables specified in the LINEQS model. This is accomplished by following some naming conventions for the latent variables. See the section "Naming Variables in the LINEQS Model" on page 6840 for details about these naming rules. Essentially, latent factors (systematic sources) must start with letter 'F' or 'f'. Error terms must start with letter 'E', 'e', 'D', or 'd'. Prefix 'E' or 'e' represents the error term of an endogenous manifest variable. Prefix 'D' or 'd' represent represents disturbance (or error) term of an endogenous latent variable. Although 'D' and 'E' variables are conceptually different, for modeling purposes 'D' and 'E' prefixes are not distinguished in PROC TCALIS. Essentially, only the distinction between latent factors (systematic sources) and errors or disturbances (unsystematic sources) is critical in specifying a proper model analyzed by PROC TCALIS. Manifest variables in PROC TCALIS do not need to follow additional naming rules beyond those required by the general SAS system—they only need to have references in the input data set.

Types of Variables and Semantic Rules of Equations

Depending on their roles in the system of equations, variables in a LINEQS model can be classified into endogenous or exogenous. An endogenous variable is a variable that serves as an outcome variable (left-hand side of an equation) in one of the equations. All other variables are exogenous variables, including those manifest variables that never appear in any places of any equation but are specified in the VAR statement.

The syntactic rules for equations are far from sufficient to define the system of equations that the LINEQS model would accept. You must also observe the following semantic rules:

- Only manifest or latent variables can be endogenous. This means that you cannot specify any

error or disturbances variables on the left-hand side of equations. This also means that error and disturbance variables are always exogenous in the LINEQS model.

- An endogenous variable appearing on the left-hand side of an equation cannot appear on the left-hand side of another equation. In other words, you have to specify all the predictors for an endogenous variable in a single equation.
- An endogenous variable appearing on the left-hand side of an equation cannot appear on the right-hand side of the same equation.
- Each equation must contain one and only one **unique** error term, be it an E-variable for manifest outcome variable or a D-variable for latent outcome variable. If, indeed, you want to specify an equation without an error term, you might equivalently set the variance of the error term to a fixed zero in the STD statement.

Mean Structures in Equations

To fit a LINEQS model with mean structures, you can specify the MEANSTR option in the PROC TCALIS or the associated MODEL statement. Alternatively, you can use the Intercept variable in equations for including intercept terms or the MEAN statement to specify the mean parameters. Conceptually, the Intercept variable is a special "variable" containing the value one for each observation. You do not need to have this variable in your input data set, nor do you need to generate it in the DATA step. It serves as a notational convenience in the LINEQS modeling language. The actual intercept parameter is expressed as a coefficient parameter with the intercept variable. For example, consider the following LINEQS model specification.

```
lineqs
   V1 = a1 (10) Intercept +          F1 + E1,
   V2 =                    + b2      F1 + E2,
   V3 = a2      Intercept + b2       F1 + E3,
   V4 = a2      Intercept + b4 (.4)  F1 + E4;
```

In the first equation, a1, with a starting value at 10, is the intercept parameter for V1. The intercept parameters for V3 and V4 are constrained to be the same. They are named a2 in the equations. Even though there is no Intercept variable specified in the equation for endogenous variable V2, its intercept parameter is assumed to be a fixed zero. This is because once you use at least one Intercept variable in an equation or you specify at least one mean parameter in the MEAN statement, mean structure analysis is activated and all mean and intercept parameters have default values at zero, unless you specify parameter names or other fixed values explicitly in the equations or in the MEAN statement.

Modifying a LINEQS Model from a Reference Model

In this section, it is assumed that you use a REFMODEL statement within the scope of a MODEL statement and the reference model (or base model) is a LINEQS model. The reference model will be referred to as the old model, while the model being defined is referred to as the new model. If the new model is not intended to be an exact copy of the old model, you can use the following extended

LINEQS modeling language to make modifications within the scope of the MODEL statement for the new model.

The syntax of the extended LINEQS modeling language is the same as that of the ordinary LINEQS modeling language (see the section "LINEQS Statement" on page 6766):

LINEQS *model equations* **;**
STD *partial variance parameters* **;**
COV *covariance parameters* **;**
MEAN *mean parameters* **;**

The new model is formed by integrating with the old model in the following ways:

Duplication: If you do not specify in the new model an equation with an outcome variable (that is, a variable on the left of the equal sign) that exists in the old model, the equation with that outcome variable in the old model is duplicated in the new model. For specifications other than the LINEQS statement, if you do not specify in the new model a parameter location that exists in the old model, the old parameter specification is duplicated in the new model.

Addition: If you specify in the new model an equation with an outcome variable that does not exist as an outcome variable in the equations of the old model, the equation is added in the new model. For specifications other than the LINEQS statement, if you specify in the new model a parameter location that does not exist in the old model, the new parameter specification is added in the new model.

Deletion: If you specify in the new model an equation with an outcome variable that also exists as an outcome variable in an equation of the old model and you specify the missing value '.' as the only term on the right-hand side of the equation in the new model, the equation with the same outcome variable in the old model is not copied into the new model. For specifications other than the LINEQS statement, if you specify in the new model a parameter location that also exists in the old model and the new parameter is denoted by the missing value '.', the old parameter specification is not copied into the new model.

Replacement: If you specify in the new model an equation with an outcome variable that also exists as an outcome variable in an equation of the model and the right-hand side of the equation in the new model is not denoted by the missing value '.', the new equation replaces the old equation with the same outcome variable in the new model. For specifications other than the LINEQS statement, if you specify in the new model a parameter location that also exists in the old model and the new parameter is not denoted by the missing value '.', the new parameter specification replaces the old one in the new model.

For example, in the following two-group analysis you specify model 2 by referring to model 1 in the REFMODEL statement.

```
proc tcalis;
   group 1 / data=d1;
   group 2 / data=d2;
   model 1 / group=1;
      lineqs
         V1   =      1 F1   + E1,
         V2   =  load1 F1   + E2,
         V3   =  load2 F1   + E3,
         F1   =     b1 V4   + b2 V5 + b3 V6 + D1;
      std
         E1-E3  = ve1-ve3,
         D1     = vd1,
         V4-V6  = phi4-phi6;
      cov
         E1 E2 = cve12;
   model 2 / group=2;
      refmodel 1;
      lineqs
         V3   = load1 F1 + E3;
      cov
         E1 E2 = .,
         E2 E3 = cve23;
   run;
```

Model 2 is the new model which refers to the old model, model 1. This example illustrates the four types of model integration:

- Duplication: All equations, except the one with outcome variable V3, in the old model are duplicated in the new model. All specifications in the STD and COV statements, except the covariance between E1 and E2, in the old model are also duplicated in the new model.
- Addition: The parameter cve23 for the covariance between E2 and E3 is added in the new model.
- Deletion: The specification of covariance between E1 and E2 in the old model is not copied into the new model, as indicated by the missing value '.' specified in the new model.
- Replacement: The equation with V3 as the outcome variable in the old model is replaced with a new equation in the model. The new equation uses parameter load1 so that it is now constrained to be the same as the regression coefficient in the equation with V2 as the outcome variable.

LISMOD Statement

LISMOD < *var_lists* > **;**

where *var_lists* represent one or more of the following:

- YVAR | YV | Y = *variables*
- XVAR | XV | X = *variables*
- ETAVAR | ETAV | ETA = *variables*
- XIVAR | XIV | XI | KSIVAR | KSIV | KSI = *variables*

LISMOD stands for LISrel MODeling, where LISREL is the program developed by Jöreskog and Sörbom (1988). Like the original implementation of LISREL, LISMOD uses a matrix specification interface. To complete the LISMOD specification, you might need to add as many MATRIX statements as needed, as shown in the following statement structure for the LISMOD model:

LISMOD *variable lists* **;**
MATRIX *matrix-name parameters-in-matrix* **;**
Repeat the MATRIX statement as needed **;**

The matrix-name in the MATRIX statement should be one of the twelve model matrices in LISMOD, as listed in the following:

- Matrices in structural model: _ALPHA_, _KAPPA_, _BETA_, _GAMMA_, _PHI_, _PSI_
- Matrices in measurement model for y-variables: _NUY_, _LAMBDAY_, _THETAY_
- Matrices in measurement model for x-variables: _NUX_, _LAMBDAX_, _THETAX_

See the section "Model Matrices in the LISMOD Model" on page 6849 for definitions of these matrices and their roles in the LISMOD modeling language. See the MATRIX statement on page 6784 for the details of parameter specification.

In the LISMOD statement, you can set the following four lists of variables:

- YVAR= list is for manifest variables y that are directly related to the endogenous latent eta-variables. Variables in the list are called y-variables.
- XVAR= list is for manifest variables x that are directly related to the exogenous latent ksi-variables. Variables in the list are called x-variables.
- ETAVAR= list is for endogenous latent variables η. Variables in the list are called eta-variables.
- XIVAR= list is for exogenous latent variables ξ. Variables in the list are called xi- or ksi-variables.

The order of variables in the lists of the LISMOD statement is important. The order is used to define the variable order in rows and columns of the LISMOD model matrices.

Depending on the model of interest, you might not need to set all the lists of variables. When some variable lists are not specified, the full model reduces to specialized submodels. However, to be a proper submodel in the LISMOD modeling language, it is necessary (but not sufficient) that at least one of the yvar= or xvar= lists is defined. See the section "LISMOD Submodels" on page 6851 for the details about LISMOD submodels that PROC TCALIS can handle.

An example of a LISMOD model specification is shown as follows:

```
proc tcalis;
   lismod xvar=x1-x3 yvar=y1-y6 xivar=xi etavar=eta1-eta2;
   matrix _LAMBDAY_   [,1]  = 1. load3 load4,
                      [,2]  = 0. 0. 0. 1. load5 load6;
   matrix _THETAY_    [1,1] = ey1-ey3,
                      [2,1] = cey;
   matrix _LAMBDAX_   [,]   = 1. load1 load2;
   matrix _THETAX_    [1,1] = 3*ex;
   matrix _GAMMA_     [,1]  = beta1 beta2;
   matrix _PHI_       [1,1] = phi;
   matrix _PSI_       [1,1] = psi1-psi2;
run;
```

In this example, you have three x-variables x1–x3, six y-variables y1–y6, one ξ-variable xi, and two η-variables eta1–eta2. The numbers of variables in these lists define the dimensions of the LISMOD model matrices. For example, matrix _LAMBDAY_ is 6×2, with y1–y6 as the row variables and eta1–eta2 as the column variables. Matrix _THETAX_ is 3×3, with x1–x3 as the row and column variables. In the MATRIX statements, you specify parameters in the elements of the matrices. After the matrix name, you specify in the parentheses '[' and ']' the starting row and column numbers of the first element to be parameterized. After the equal sign, you specify fixed or free parameters for the matrix elements.

Depending on how you specify the starting row and column numbers, the parameter specification might proceed differently. See the MATRIX statement on page 6784 for a detailed description. In this example, the first specification of the parameters in the _LAMBDAY_ matrix starts from [,1]—meaning that it starts from the first column and will proceed downwards. As a result, the [1,1] element is a fixed constant 1.0, the [2,1] element is a free parameter called load3, and the [3,1] element is a free parameter called load4. Similarly, in the second specification in the _LAMBDAY_ matrix, the [1,2], [2,2], [3,2], and [4,2] elements take constant values 0, 0, 0, and 1, respectively, and the [5,2] and [6,2] elements are free parameters load5 and load6, respectively.

You can also specify the parameters of a row using similar notation. For example, with the notation [2,] for the starting row and column numbers, specification proceeds to the left with the same second row in the matrix.

If you have both starting row and column numbers specified, such as that in the first specification in matrix _THETAY_, the parameter specification starts from [1,1] and proceeds to the next row and column numbers—that is [2,2], [3,3], and so on. This results in specifying the diagonal elements of matrix _THETAY_ as free parameters ey1, ey2, and ey3.

With the notation [,], there are no starting row and column numbers specified. Specification will start from the first valid element in the matrix and proceeds row-wise for all valid elements in the matrix. For example, in the matrix _LAMBDAX_ statement, the [1,1] element of matrix _LAMBDAX_ is a fixed constant 1, and the [1,2] and [1,3] elements are free parameters load1 and load2, respectively.

Modifying a LISMOD Model from a Reference Model

In this section, it is assumed that you use a REFMODEL statement within the scope of a MODEL statement and the reference model (or base model) is also a LISMOD model. The reference model is referred to as the old model, while the model that makes reference to this old model is referred to as the new model. If the new model is not intended to be an exact copy of the old model, you can use the following extended LISMOD modeling language to make modifications within the scope of the MODEL statement for the new model. The syntax is similar to, but not exactly the same as, the ordinary LISMOD modeling language (see the section "LISMOD Statement" on page 6772). The respecification syntax for a LISMOD model is shown as follows:

LISMOD ;
  MATRIX *matrix-name parameters-in-matrix* ;
  Repeat the MATRIX statement as needed ;

First, in the respecification you should not put any variable lists in the LISMOD statement. The reason is that the parameter respecifications in the new model make reference to the variable lists of the old model. Therefore, the variable lists in the new model are implicitly assumed to be exactly the same as those in the old model. Because of this, the LISMOD statement is entirely optional for the respecification in the new model.

Second, you can use MATRIX *matrix-name* statements to modify the old model by using the same syntax as in the LISMOD modeling language. The *matrix-name* can be one of the twelve possible LISMOD matrices. In addition, in the respecification syntax you can use the missing value '.' to drop a parameter specification from the old model.

The new model is formed by integrating with the old model in the following ways:

Duplication: If you do not specify in the new model a parameter location that exists in the old model, the old parameter specification is duplicated in the new model.

Addition: If you specify in the new model a parameter location that does not exist in the old model, the new parameter specification is used in the new model.

Deletion: If you specify in the new model a parameter location that also exists in the old model and the new parameter is denoted by the missing value '.', the old parameter specification is not copied into the new model.

Replacement: If you specify in the new model a parameter location that also exists in the old model and the new parameter is not denoted by the missing value '.', the new parameter specification replaces the old one in the new model.

For example, in the following two-group analysis you specify model 2 by referring to model 1 in the REFMODEL statement.

```
proc tcalis;
   group 1 / data=d1;
   group 2 / data=d2;
   model 1 / group=1;
      lismod xvar=X1-X3 yvar=Y1-Y6 xivar=xi etavar=eta1-eta2;
      matrix _LAMBDAY_   [,1]  = 1. load3 load4,
                         [,2]  = 0. 0. 0. 1. load5 load6;
      matrix _THETAY_    [1,1] = ey1-ey3,
                         [2,1] = cey;
      matrix _LAMBDAX_   [,]   = 1. load1 load2;
      matrix _THETAX_    [1,1] = 3*ex;
      matrix _GAMMA_     [,1]  = beta1 beta2;
      matrix _PHI_       [1,1] = phi;
      matrix _PSI_       [1,1] = psi1-psi2;
   model 2 / group=2;
      refmodel 1;
      matrix _THETAY_    [2,1] = .;
      matrix _THETAX_    [1,1] = ex1-ex3;
      matrix _BETA_      [2,1] = beta;
   run;
```

In this example, model 2 is the new model which refers to the old model, model 1. It illustrates the four types of model integration:

- Duplication: All parameter locations and specifications in the old model are duplicated in the new model, except the [2,1] element in matrix _THETAY_ and the diagonal of matrix _THETAX_, which are being modified in the new model.
- Addition: The _BETA_[2,1] parameter location is added with a new parameter beta in the new model. This indicates that *eta1* is a predictor variable of *eta2* in the new model, but not in the old model.
- Deletion: Because the missing value '.' is used for the parameter value, the _THETAY_[2,1] parameter location is no longer defined as a free parameter in the new model. In the old model, the same location is defined by the free parameter cey.
- Replacement: The diagonal elements of the _THETAX_ matrix in the new model are now defined by three distinct parameters ex1–ex3. This replaces the old specification where a single constrained parameter ex is applied to all the diagonal elements in the _THETAX_ matrix.

LMTESTS Statement

LMTESTS *option* < *option . . .* > **;**

where *option* represents one of the following:

- *display-option*
- *test-set*

and *test-set* represents one of the following:

- *set-name* = [*regions*]
- *set-name* = { *regions* }

where *set-name* is the name of the set of Lagrange multiplier (LM) tests defined by the *regions* that follow after the equal sign and *regions* are keywords denoting specific sets of parameters in the model.

You can use the LMTESTS statement to set *display-options* or to customize the *test-sets* for the LM tests. The LMTESTS statement is one of the model analysis statements. It can be used within the scope of the TCALIS statement so that the options will apply to all models. It can also be used within the scope of each MODEL statement so that the options will apply only locally. Therefore, different models within a TCALIS run can have very different LMTESTS *options*.

The LM Tests Display Options

The following are the *display-options* for the LM tests:

DEFAULT
conducts the default sets of LM tests for freeing fixed parameters in the model. This option is used when you need to reset the default sets of LM tests in the local model. For example, you might have turned off the default LM tests by using the NODEFAULT option in the LMTESTS statement within the scope of PROC TCALIS statement. However, for the model under the scope of a particular MODEL statement, you can use this DEFAULT option in the local LMTESTS statement to turn on the default LM tests again.

MAXRANK
sets the maximum number of rankings within a set of LM tests. The actual number of test rankings might be smaller because the number of possible LM tests within a set might be smaller than the maximum number requested.

NODEFAULT
turns off the default sets of LM tests for freeing fixed parameters in the model. As a result, only the customized LM tests defined in the *test-sets* of the LMTESTS statement are conducted and displayed. Note that the LM tests for equality and active boundary constraints are not turned off by this option. If you specify this option in the LMTESTS statement within the scope of the PROC TCALIS statement, it will propagate to all models.

NORANK
turns off the ranking of the LM tests. Ranking of the LM tests is done automatically when

the model modification indices are requested. The NORANK option is ignored if you also set the MAXRANK option.

LMMAT
prints the sets of LM tests in matrix form, in addition to the normal LM test results.

The Customized Sets of LM Tests: Syntax of the Test-sets

In addition to the *display-options*, you can define customized sets of LM tests as *test-sets* in the LMTESTS statement. You can define as many *test-sets* as you like. Ranking of the LM tests will be done individually for each *test-set*. For example, the following LMTESTS statement requests that the default sets of LM tests not be conducted by the NODEFAULT option. Instead, two customized *test-sets* are defined.

```
lmtests nodefault MyFirstSet=[ALL] MySecondSet=[COVEXOG COVERR];
```

The first customized set MyFirstSet pulls all possible parameter locations together for the LM test ranking (ALL keyword). The second customized set MySecondSet pulls only the covariances among exogenous variables (COVEXOG keyword) and among errors (COVERR keyword) together for the LM test ranking.

Two different kinds of *regions* for LM tests are supported in PROC TCALIS: matrix-based or non-matrix-based.

The matrix-based *regions* can be used if you are familiar with the matrix representations of various types of models. Note that defining *test-sets* by using matrix-based *regions* does not mean that LM tests are printed in matrix format. It means only that the parameter locations within the specified matrices are included into the specific *test-sets* for LM test ranking. For matrix output of LM tests, use the LMMAT option in the LMTESTS statement.

Non-matrix-based *regions* do not assume the knowledge of the model matrices. They are easier to use in most situations. In addition, non-matrix-based *regions* can cover special subsets of parameter locations that cannot be defined by model matrices and submatrices. For example, because of the compartmentalization according to independent and dependent variables in the LINEQS model matrices, the sets of LM tests defined by the LINEQS matrix-based *regions* are limited. For example, you cannot use any matrix-based *regions* to request LM tests for new paths to existing independent variables in the LINEQS model. Such a matrix does not exist in the original specification. However, you can use the non-matrix based *region* NEWENDO to refer to these new paths.

The *regions* for parameter locations are specified by keywords in the LMTESTS statement. Because the *regions* are specific to the types of models, they are described separately for each model type in the following.

The LM Test Regions for FACTOR Models

Keywords for Matrix-Based Regions

FACTERRV | FACTERRV
specifies the error variances.

FACTFCOR | FACTFCOR
specifies the covariances among factors.

FACTINTE | FACTINTE
specifies the intercepts.

FACTLOAD | FACTLOAD
specifies the factor loadings.

FACTMEAN | FACTMEAN
specifies the factor means.

See the section "Model Matrices in the FACTOR Model" on page 6864 for definitions of these FACTOR model matrices.

Keywords for Non-Matrix-Based Regions

ALL
specifies all parameter locations.

COV
specifies the covariances among factors.

COVERR
specifies the covariances among errors.

COVFACT | COVLV
specifies the covariances among factors.

FIRSTMOMENTS
specifies the means of factors and the intercepts.

INTERCEPTS
specifies the intercepts.

LOADINGS
specifies the factor loadings.

MEANS | MEAN
specifies the means of factors.

The LM Test Regions for LINEQS Models

Keywords for Matrix-Based Regions

EQSALPHA | EQSALPHA
specifies the intercepts of dependent variables.

EQSBETA | EQSBETA
specifies effects of dependent variables on dependent variables.

EQSGAMMA | _EQSGAMMA_SUB_ | EQSGAMMA | EQSGAMMASUB
specifies the effects of independent variables (excluding errors) on dependent variables. Because effects of errors on dependent variables are restricted to ones in the LINEQS model, LM tests on _EQSGAMMA_ and _EQSGAMMA_SUB_ (submatrix of _EQSGAMMA_) are the same.

EQSNU | _EQSNU_SUB_ | EQSNU | EQSNUSUB
specifies the means of independent variables (excluding errors). Because means of errors are restricted to zero in the LINEQS model, LM tests on _EQSNU_ and _EQSNU_SUB_ (submatrix of _EQSNU_) are the same.

EQSPHI | EQSPHI
specifies variances and covariances among all independent variables, including errors.

EQSPHI11 | EQSPHI11
specifies variances and covariances among independent variables, excluding errors.

EQSPHI21 | EQSPHI21
specifies covariances between errors and disturbances with other independent variables.

EQSPHI22 | EQSPHI22
specifies variances and covariances among errors and disturbances.

See the section "Matrix Representation of the LINEQS Model" on page 6841 for definitions of these model matrices and submatrices.

Keywords for Non-Matrix-Based Regions

ALL
specifies all possible parameter locations.

COV
specifies all covariances among independent variables, including errors and disturbances.

COVERR
specifies covariances among errors or disturbances.

COVEXOG
specifies covariances among independent variables, excluding errors and disturbances.

COVEXOGERR
specifies covariances of errors and disturbances with other independent variables.

COVLV | COVFACT
specifies covariances among latent variables (excluding errors and disturbances).

COVMV | COVOV
specifies covariance among independent manifest variables.

EQUATION | EQUATIONS
specifies all possible linear relationships among variables.

FIRSTMOMENTS
specifies means and intercepts.

INTERCEPTS | INTERCEPT
specifies intercepts of dependent variables.

LV–>LV
specifies all possible effects of latent factors on latent factors.

LV–>MV | MV<–LV
specifies all possible effects of latent factors on manifest variables.

LV<–MV | MV–>LV
specifies all possible effects of manifest variables on latent factors.

MEANS | MEAN
specifies the means of independent factors.

MV–>MV
specifies all possible effects of manifest variables on manifest variables.

NEWDEP | NEWENDO
specifies effects of other variables on the independent variables in the original model.

PATHS | PATH
specifies all possible linear relationships among variables.

The LM Test Regions for LISMOD Models

Keywords for Matrix-Based Regions

ALPHA | ALPHA
specifies the _ALPHA_ matrix.

BETA | BETA
specifies the _BETA_ matrix.

GAMMA | GAMMA
specifies the _GAMMA_ matrix.

KAPPA | KAPPA
specifies the _KAPPA_ matrix.

LAMBDA | LAMBDA
specifies the _LAMBDAX_ and _LAMBDAY_ matrices.

LAMBDAX | LAMBDAX
specifies the _LAMBDAX_ matrix.

LAMBDAY | LAMBDAY
specifies the _LAMBDAY_ matrix.

_NU_ | NU
specifies the _NUX_ and _NUY_ matrices.

NUX | NUX
specifies the _NUX_ matrix.

NUY | NUY
specifies the _NUY_ matrix.

PHI | PHI
specifies the _PHI_ matrix.

PSI | PSI
specifies the _PSI_ matrix.

THETA | THETA
specifies the _THETAX_ and _THETAY_ matrices.

THETAX | THETAX
specifies the _THETAX_ matrix.

THETAY | THETAY
specifies the _THETAY_ matrix.

Keywords for Non-Matrix-Based Regions

ALL
specifies all model matrices.

COV
specifies all covariance parameters in _THETAY_, _THETAX_, _PHI_, and _PSI_.

COVERR
specifies all covariances for errors or disturbances in _THETAY_, _THETAX_, and _PSI_.

COVFACT | COVLV
specifies all covariances among latent factors in _PHI_ when the ξ-variables exist, and in _PSI_ when the η-variables exist without the presence of the ξ-variables.

FIRSTMOMENTS
specifies all intercepts and means in _NUY_, _NUX_, _ALPHA_, and _KAPPA_.

INTERCEPTS | INTERCEPT
specifies all intercepts in _NUY_, _NUX_, and _ALPHA_.

LOADING | LOADINGS
specifies the coefficients in _LAMBDAY_ and _LAMBDAX_.

LV–>LV
specifies the effects of latent variables on latent variables. Depending on the type of LISMOD model, the _BETA_ and _GAMMA_ might be involved.

LV–>MV | MV<–LV
specifies the effects of latent variables on manifest variables. Depending on the type of LISMOD model, the _LAMBDAY_, _LAMBDAX_, and _GAMMA_ matrices might be involved.

MEANS | MEAN
specifies the mean parameters. Depending on the type of LISMOD model, the _ALPHA_ and _KAPPA_ matrices might be involved.

MV–>MV
specifies effects of manifest variables on manifest variables. Depending on the type of LISMOD model, the _BETA_ and _GAMMA_ matrices might be involved.

PATHS | PATH
specifies all path coefficients. Depending on the type of LISMOD model, the _LAMBDAY_, _LAMBDAX_, _BETA_, and _GAMMA_ matrices might be involved.

The LM Test Regions for MSTRUCT Models

Keywords for Matrix-Based Regions

MSTRUCTCOV | _COV_ | MSTRUCTCOV
specifies the _MSTRUCTCOV_ or _COV_ matrix.

MSTRUCTMEAN | _MEAN_ | MSTRUCTMEAN
specifies the _MSTRUCTMEAN_ or _MEAN_vector.

Keywords for Non-Matrix-Based Regions

ALL
specifies the _MSTRUCTCOV_ (or _COV_) and _MSTRUCTMEAN_ (or _MEAN_) matrices.

COV
specifies the _MSTRUCTCOV_ or _COV_ matrix.

MEANS | MEAN
specifies the _MSTRUCTMEAN_ or _MEAN_ matrix.

The LM Test Regions for PATH and RAM Models

Keywords for Matrix-Based Regions

RAMA | _A_ | RAMA
specifies the _RAMA_ matrix.

RAMALPHA | RAMALPHA
specifies the _RAMALPHA_ matrix.

RAMBETA | RAMBETA
specifies the _RAMBETA_ matrix.

RAMGAMMA | RAMGAMMA
specifies the _RAMGAMMA_ matrix.

RAMNU | RAMNU
specifies the _RAMNU_ matrix.

RAMP | _P_ | RAMP
specifies the _RAMP_ matrix.

RAMP11 | RAMP11
specifies the _RAMP11_ matrix.

RAMP21 | RAMP21
specifies the _RAMP21_ matrix.

RAMP22 | RAMP22
specifies the _RAMP22_ matrix.

RAMW | _W_ | RAMW
specifies the _RAMW_ vector.

Keywords for Non-Matrix-Based Regions

ALL
specifies all possible parameter locations.

ARROWS | ARROW
specifies all possible paths (that is, the entries in the _RAMA_ matrix).

COV
specifies all covariances and partial covariances (that is, the entries in the _RAMP_ matrix).

COVERR
specifies partial covariances among endogenous variables (that is, the entries in the _RAMP11_ matrix).

COVEXOG
specifies covariances among exogenous variables (that is, the entries in the _RAMP22_ matrix).

COVEXOGERR
specifies partial covariances of endogenous variables with exogenous variables (that is, the entries in the _RAM21_ matrix).

COVLV | COVFACT
specifies covariance among latent factors (that is, entries in _RAMP11_ pertaining to latent variables).

COVMV | COVOV
specifies covariance among manifest variables (that is, entries in _RAMP11_ pertaining to manifest variables).

FIRSTMOMENTS
specifies means or intercepts (that is, entries in _RAMW_ vector).

INTERCEPTS | INTERCEPT
specifies intercepts for endogenous variables (that is, entries in _RAMALPHA_ vector).

LV–>LV
specifies effects of latent variables on latent variables.

LV–>MV | MV<–LV
specifies effects of latent variables on manifest variables.

LV<–MV | MV–>LV
specifies effects of manifest variables on latent variables.

MEANS | MEAN
specifies the means of exogenous variables (that is, entries in the _RAMNU_ vector).

MV–>MV
specifies effects of manifest variables on manifest variables.

NEWENDO
specifies new paths to the exogenous variables in the original model.

PATHS | PATH
specifies all possible paths (that is, the entries in the _RAMA_ matrix).

MATRIX Statement

MATRIX *matrix-name* < *location* > = *parameter-spec* < , *location* = *parameter-spec* . . . > **;**

In MATRIX statements, you specify fixed constants or free parameters for the elements of the model matrix referred by the *matrix-name*. The *location* indicates the starting row and column numbers of the matrix being specified and the *parameter-spec* is a list of free or fixed parameters for the elements in the matrix. You can also assign initial values for the free parameters in the *parameter-spec* list.

The MATRIX statement is a subsidiary model specification statement of the LISMOD and MSTRUCT modeling languages. You might need to use the MATRIX statements as many times as needed for specifying your model. However, you can use the MATRIX statement at most once for each distinct model matrix.

Valid Matrix Names for the LISMOD modeling language

There are 12 model matrices in the LISMOD model and they correspond to the following *matrix-names* for the LISMOD modeling language:

- matrices in the measurement model for the y-variables

  LAMBDAY the matrix of regression coefficients of the y-variables on the eta-variables

  NUY the vector of intercept terms of the y-variables

  THETAY the error covariance matrix for the y-variables

- matrices in the measurement model for the x-variables

  LAMBDAX the matrix of regression coefficients of the x-variables on the ksi-variables

  NUX the vector of intercept terms of the x-variables

  THETAX the error covariance matrix for the x-variables

- matrices in the structural model

  ALPHA the vector of intercept terms of the eta-variables

  BETA the matrix of regression coefficients of the eta-variables on the eta-variables

  GAMMA the matrix of regression coefficients of the eta-variables on the ksi-variables

  KAPPA the mean vector for the ksi-variables

  PHI the covariance matrix for the ksi-variables

  PSI the error covariance matrix for the eta-variables

Valid Matrix Names for the MSTRUCT Modeling Language

the following *matrix-names* are valid for the MSTRUCT modeling language:

COV the covariance matrix

MEAN the mean vector

Specifying Locations in Model Matrices

There are four different ways to specify the starting *location* of the parameter matrix in question. Each way of specification leads to different continuation direction for assigning the parameters specified in the *parameter-spec* list. Assuming that there are n ($n \geq 1$) parameters specified in the *parameter-spec* list, the four different parameter assignment schemes due to different ways of starting *location* specification are described as follows:

[*i* , *j*] The elements in the *parameter-spec* list correspond to the diagonally continued matrix elements [*i,j*], [*i*+1,*j*+1], ..., [*i*+*n*-1,*j*+*n*-1]. The number of elements is defined by the length of the *parameter-spec* and is eventually terminated by the matrix boundaries. If the *parameter-spec* list contains just one element (constant or variable), then it is assigned to the matrix element [*i,j*].

[*i* ,] In this type of row-wise specification, the elements in the *parameter-spec* list correspond to the horizontally continued matrix elements [*i,1*], [*i,2*], ..., [*i,n*]. For a rectangular or non-symmetric square matrix with *k* columns, the number of elements in *parameter-spec* should not exceed *k*. For a symmetric matrix, because you need to specify lower triangle elements only (upper triangular elements are duplicated from the lower triangle elements automatically), the number of elements in *parameter-spec* list for this type of row-wise specification should not exceed *i*.

[, *j*] In this type of column-wise specification, the elements in the *parameter-spec* list correspond to the vertically continued matrix elements [*i,j*], [*i*+1,*j*], ..., [*i*+*n*-1,*j*]. For a rectangular or nonsymmetric square matrix, the value of *i* is 1. For a symmetric matrix, because you need to specify the lower triangular elements only (upper triangular elements are duplicated from the lower triangle elements automatically), the value of *i* in this type of column-wise specification is *j*.

[,] Unspecified location: The elements of the *parameter-spec* list are assigned to all valid matrix positions starting at the element [1,1] and continuing row-wise, until reaching the last element of the *parameter-spec* list or the last valid matrix position. For a symmetric matrix, the valid positions are the elements in the lower triangle since the other triangle receives the symmetric allocation automatically.

When there is no starting *location* specified, it is equivalent to using [,] for an *unspecified location*.

Specifying Parameters in Model Matrices

The *parameter-spec* list contains numeric values or parameter names, or both, that are assigned to the matrix elements starting at a specified position and proceeding in a specified direction, which are determined by the way of the starting *location* specification.

The syntax for the *parameter-spec* list is the same as the *parameter-spec* list for the STD statement. There are three types of specification in the *parameter-spec* list, as described in the following.

- Parameter names alone represent parameters in the model without starting values.
- Parameter names followed by numbers inside a pair of parentheses represent parameters in the model with starting values provided in the parentheses.
- Numbers alone represent fixed constant values for the corresponding matrix elements.

For example, consider the following *parameter-spec* list.

```
0. 1. A2-A5 (1.4 1.9 2.5) 5.
```

The first two matrix elements are assigned with fixed values 0 and 1, respectively. The next element is a parameter named A2 with no initial value provided. The next three matrix elements are assigned with parameters A3, A4, and A5 with initial values 1.4, 1.9, and 2.5, respectively. The next matrix element is assigned with a constant 5. At first glance, the initial values in the parentheses are seemingly assigned to A2, A3, and A4, respectively. However, because A2–A5 is a shorthand of four consecutive parameters, the preceding *parameter-spec* list is equivalent to:

```
0. 1. A2 A3 A4 A5 (1.4 1.9 2.5) 5.
```

With this explicit specification, it is clearer to see that the initial values in the parentheses are assigned to parameters A3–A5, respectively.

If your model contains many unconstrained parameters and it is too cumbersome to find different parameter names, you can specify all those parameters by the same prefix name. A prefix name is a short name called "root" followed by double underscores '__'. Whenever a prefix name is encountered, the TCALIS procedure generates a parameter name by appending a unique integer to the root. Hence, the prefix name should have few characters so that the generated parameter name is not longer than thirty-two characters. For example, consider the following *parameter-spec* list.

```
0. 1. 4 * A__ (1.4 1.9 2.5) 5.
```

If the prefix name A__ has not been specified previously, then A1, A2, A3, and A4 are generated by the **4 * A__** syntax. However, if the prefix name A__ has been specified, say, twice, previously, then A3, A4, A5, and A6 are generated by the same **4 * A__** syntax.

To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names. For example, consider the following specification.

```
matrix _mean_ = B1 B2 2 * B__;
```

If the prefix name B__ has not been specified previously, then B1 and B2 are generated for the third and the fourth elements, respectively, of the mean vector. However, these two parameter names are the same as those assigned explicitly to the first and the second elements of the mean vectors, resulting to two equality constraints that might not have been intended.

Modifying a Parameter Specification from a Reference Model

If you define a new LISMOD or MSTRUCT model by using a reference (or base) model in the REFMODEL statement, in some situations you might want to modify some parameter specifications from the MATRIX statement of the reference model before transferring the specifications to the new model. If you want to change a particular MATRIX parameter specification from the reference model, you can simply put the corresponding parameter location with a new parameter specification in the new model. If you want to delete a particular parameter location from the reference model, you can put that parameter location with the missing value '.' in the new model. For example, suppose you are defining a new LISMOD model by using the REFMODEL statement and the covariance between variables F1 and F2 is defined as a fixed or free parameter in the reference model. If you do not want this parameter specification to be copied into your new model, you can use the following specification in the new model:

```
matrix _PHI_  F1 F2 = .;
```

Notice that the missing value syntax is valid only when you use the REFMODEL statement. See the section "Modifying a LISMOD Model from a Reference Model" on page 6774 for a more detailed example of LISMOD model respecification. See the section "Modifying an MSTRUCT Model from a Reference Model" on page 6793 for a more detailed example of MSTRUCT model respecification.

MEAN Statement

MEAN *assignment* < , *assignment* . . . > **;**

where *assignment* represents:
variables = *parameter-spec*

The MEAN statement invokes the fitting of the mean structures of the model. The MEAN statement is one of the subsidiary model specification statements for supplementing modeling specification. You can use the MEAN statements to specify mean parameters in the FACTOR, LINEQS, and PATH modeling languages. You can also use the MEAN statement to specify the intercept parameters in the FACTOR or PATH modeling languages. The syntax of the MEAN statement is essentially the same as that of the STD or PVAR statement.

In each *assignment* of the MEAN statement, you put the *variables* on the left-hand side of the equal sign and specify the parameters on the right-hand side *parameter-spec*. Like elsewhere in the PROC TCALIS syntax, in *parameter-spec* you can put numbers for fixed values or parameter names for free or constrained parameters. Parameter names are optionally followed by numbers in parentheses for starting values. See the STD statement on page 6815 for examples.

In the FACTOR or PATH model, the variables in each *assignment* can be exogenous or endogenous. If a variable in the *variables* list is exogenous, the corresponding parameter is the mean parameter for the variable. Otherwise, the corresponding parameter is the intercept parameter for the variable. In the LINEQS model, however, only non-error type exogenous variables (that is, not E- or D- variables) can be used in the MEAN statement. The parameters are for the means of the exogenous variables. You cannot specify the intercept parameters in the MEAN statement of the LINEQS modeling language. Instead, you should specify intercepts in the equations of the LINEQS statement.

Automatic Mean Parameters among Exogenous Variables

If the names of any exogenous manifest variables in the LINEQS model are not included in the MEAN statement, the means of those exogenous manifest variables are assumed to be parameters to estimate. These parameters will be named automatically with the prefix _Add and appended with integer suffixes. However, all other mean parameters other than those for exogenous manifest variables are fixed zeros by default unless they are specified otherwise in the MEAN statement. In addition, all intercept parameters are assumed to be fixed zeros, unless they are specified otherwise in the LINEQS statement.

Modifying a Parameter Specification from a Reference Model

If you define a new FACTOR, LINEQS, or PATH model by using a reference (or base) model in the REFMODEL statement, in some situations you might want to modify some parameter specifications from the MEAN statement of the reference model before transferring the specifications to the new model. If you want to change a particular MEAN parameter specification from the reference model, you can simply put the corresponding parameter location with a new parameter specification in the new model. If you want to delete a particular parameter location from the reference model, you can put that parameter location with the missing value '.' in the new model. For example, suppose you are defining a new LINEQS model by using the REFMODEL statement and the mean of variable F1 is defined as a fixed or free parameter in the reference model. If you do not want this parameter specification to be copied into your new model, you can use the following specification in the new model:

```
mean F1 = .;
```

Notice that the missing value syntax is valid only when you use with the REFMODEL statement. See the section "Modifying a FACTOR Model from a Reference Model" on page 6758 for a more detailed example of FACTOR model respecification. See the section "Modifying a LINEQS Model from a Reference Model" on page 6769 for a more detailed example of LINEQS model respecification. See the section "Modifying a PATH Model from a Reference Model" on page 6800 for a more detailed example of PATH model respecification.

In LINEQS models, all means of exogenous manifest variables are free parameters to estimate unless you specify otherwise in the MEAN statement (see the section "Automatic Mean Parameters among Exogenous Variables" on page 6788). This is also true for LINEQS models specified under the REFMODEL statement. In the reference model, mean parameters generated by PROC TCALIS are named with the _Add prefix and appended with integer suffixes. These parameter specifications in the reference model do **not** transfer to the new model. Only after the new model is resolved from the reference model, the REFMODEL statement options, the RENAMEPARM statement options, and model respecifications are the additional mean parameters with the _Add prefix generated for the remaining unspecified means of exogenous manifest variables. In this way, the generated mean parameters for the exogenous mainfest variables in the new model are not constrained to be the same as the corresponding means in the reference model. If you want any of these mean parameters to be constrained across the models, you must specify them explicitly in the MEAN statement of the reference model so that the constrained mean specifications are transferred to the new model.

MODEL Statement

MODEL *i* < / *options* > **;**

where *i* is an assigned model number between 1 and 9999, inclusively.

A MODEL statement signifies the beginning of a model specification block and designates a model number for the model. All main and subsidiary model specification statements after a MODEL statement belong in that model until another MODEL or GROUP statement is encountered.

The MODEL statement itself does not serve the purpose of model specification, which is actually done by the main and subsidiary model specification statements that follow it. The MODEL statement serves as a "place-holder" of specification of a single model. It also makes the reference to a model easier with an assigned model number. For example, consider the following statements:

```
proc tcalis;
   group 1 / data=women_data;
   group 2 / data=men_data;
   model 1 / group=1 label='Women Model';
      {model 1 specification here}
   model 2 / group=2 label='Men Model';
      {model 2 specification here}
   run;
```

This example illustrates a two-group analysis with two models. One is model 1 labeled as 'Women Model' in a MODEL statement. Another is model 2 labeled as 'Men Model' in another MODEL statement. The two groups, group 1 and group 2, as defined in two separate GROUP statements, are fitted by model 1 and model 2, respectively, as indicated by the GROUP= option of the MODEL statements. Within the scope of model 1, you provide model specification statements by using the main and subsidiary model specification statements. Usually, one of the following main model specification statements is used: FACTOR, LINEQS, LISMOD, MSTRUCT, PATH, RAM, or REFMODEL. Similarly, you provide another set of model specification statements within the scope of model 2.

Hence, for an analysis with a single group, the use of the MODEL statement is not necessary because the model that fits the group is unambiguous.

You can set model-specific *options* in each MODEL statement. All but two of these *options* are also available in the PROC TCALIS statement. If you set these *options* in the PROC TCALIS statement, they will apply to all models, unless you respecify them in the local MODEL statements. If you want to apply some *options* only to a particular model, set these *options* in the MODEL statement corresponding to that model.

You can also set group-specific *options* in the MODEL statement. These group *options* will apply to the groups that are specified in GROUP= option of the MODEL statement. See the section "Options Available in the GROUP and PROC TCALIS Statements" on page 6765 for a detailed descriptions of these group *options*.

Option Available in MODEL Statement Only

LABEL | NAME=*name*
specifies a label for the model. You can use any valid SAS names up to 256 characters for labels. You can also use quote strings for labels.

GROUP | GROUPS=*int-list*
specifies a list of integers *int-list* of group numbers representing the groups to be fitted by the model.

Options Available in the MODEL and PROC TCALIS Statements

These options are available in the MODEL and PROC TCALIS statements. If you specify these options in the PROC TCALIS statement, they are transferred to all MODEL statements. These options might be overwritten by the respecifications in the local MODEL statements.

Option	Description
CORRELATION on page 6728	analyzes correlation matrix
COVARIANCE on page 6728	analyzes covariance matrix
DEMPHAS= on page 6728	emphasizes the diagonal entries
EFFPART | TOTEFF on page 6729	displays total, direct, and indirect effects
INEST= on page 6730	specifies the data set that contains the initial values and constraints
INMODEL | INRAM= on page 6730	specifies the data set that contains the model specifications
MEANSTR on page 6733	analyzes the mean structures
MODIFICATION on page 6734	computes modification indices
NOMOD on page 6735	suppresses modification indices
NOORDERSPEC on page 6735	displays model specifications and results according to the input order
NOPARMNAME on page 6735	suppresses the printing of parameter names in results
NOSTAND on page 6735	suppresses the standardized output
NOSTDERR on page 6735	suppresses standard error computations
ORDERSPEC on page 6738	orders the model output displays according to the parameter types within each model
OUTEST= on page 6738	specifies the data set that outputs the estimates and their covariance matrix
OUTMODEL | OUTRAM= on page 6739	specifies the output data set for storing the model specification and results
PARMNAME on page 6740	displays parameter names in model specifications and results
PDETERM on page 6740	computes the determination coefficients
PESTIM on page 6741	prints parameter estimates
PINITIAL on page 6741	prints initial pattern and values
PLATCOV on page 6741	computes the latent variable covariances and scoring coefficients
PRIMAT on page 6741	displays results in matrix forms
STDERR on page 6744	computes the standard errors

Options Available in MODEL, GROUP, and PROC TCALIS Statements

Some options in the GROUP statement can also be specified in the MODEL statements. Group options that are specified the MODEL statements are transferred to the GROUP statements that define the groups fitted by the associated models in the MODEL statements. This is a little bit more convenient than setting the common group options individually in the GROUP statements for all fitted groups by a model. See the section "Options Available in GROUP, MODEL, and PROC TCALIS Statements" on page 6765 for a reference of these options.

MSTRUCT Statement

MSTRUCT *VAR* = variables **;**

MSTRUCT stands for matrix structures. As opposed to other modeling languages, in which the mean and covariance structures are implied from paths, equations, or complicated model matrix computations, the MSTRUCT language is for direct structured mean and covariance models.

In the MSTRUCT statement, you define the list of variables. To complete the specification, you must use additional MATRIX statements to specify the parameters:

MSTRUCT *variables* **;**
  MATRIX _COV_ *parameters-in-matrix* **;**
  MATRIX _MEAN_ *parameters-in-matrix* **;**

You use the MATRIX _COV_ statement to specify the covariance and variance parameters in the structured covariance matrix. When applicable, you use the MATRIX _MEAN_ statement to specify the parameters in the structured mean vector. Each of these matrices can be specified no more than once within a model. See the MATRIX statement on page 6784 for details.

The order of variables in the *var_list* of the MSTRUCT statement is important. It is used to refer to the row and column variables of the _COV_ and the _MEAN_ matrices. The variables specified in the list should be present in the input data set intended for the MSTRUCT model. With direct mean and covariance structures on the observed variables, no latent variables are explicitly involved in the MSTRUCT modeling language. However, this does not mean that the MSTRUCT modeling language cannot handle latent variable models. With additional specifications in the PARAMETERS and the SAS programming statements, it is possible to fit certain latent variable models by using the MSTRUCT modeling language. Despite this, the code might get too complicated and error-prone. Hence, using the MSTRUCT modeling language for latent variable modeling is not recommended to novice users. The LINEQS, LISMOD, PATH, or RAM modeling language should be considered first for latent variable modeling.

Modifying an MSTRUCT Model from a Reference Model

In this section, it is assumed that you use a REFMODEL statement within the scope of a MODEL statement and the reference model (or base model) is also an MSTRUCT model. The reference model will be referred to as the old model, while the model that makes reference to the old model is referred to as the new model. If the new model is not intended to be an exact copy of the old model, you can use the following extended MSTRUCT modeling language to make modifications within the scope of the MODEL statement for the new model. The syntax is similar to, but not exactly the same as, the ordinary MSTRUCT modeling language, as in the section "MSTRUCT Statement" on page 6792. The syntax for respecifying or modifying an MSTRUCT model takes the following form:

MSTRUCT ;
  MATRIX _COV_ *parameters-in-matrix* ;
  MATRIX _MEAN_ *parameters-in-matrix* ;

First, in the respecification, you should not put any VAR= list in the MSTRUCT statement, as you would do for specifying the original base model. The reason is that parameter respecifications in the new model make reference to the variables in the VAR= list of the old model. Therefore, the VAR= list in the new model is implicitly assumed to be exactly the same as that in the old model. This renders the specification of a VAR= list of the MSTRUCT statement of the new model unnecessary. Because the VAR= option is the only possible option in the MSTRUCT statement, it also implies that the entire MSTRUCT statement is only optional for the new model.

Second, you can use the MATRIX _COV_ and MATRIX _MEAN_ statements to modify from the old model by using the same syntax as in ordinary MSTRUCT modeling language. In addition, in the respecification syntax, you can use the missing value '.' to drop a parameter location from the old model.

The new model is formed by integrating with the old model in the following ways:

Duplication: If you do not specify in the new model a parameter location that exists in the old model, the old parameter specification is duplicated in the new model.

Addition: If you specify in the new model a parameter location that does not exist in the old model, the new parameter specification is used in the new model.

Deletion: If you specify in the new model a parameter location that also exists in the old model and the new parameter is denoted by the missing value '.', the old parameter specification is not copied into the new model.

Replacement: If you specify in the new model a parameter location that also exists in the old model and the new parameter is not denoted by the missing value '.', the new parameter specification replaces the old one in the new model.

For example, consider the following statements for a two-group analysis:

```
proc tcalis;
   group 1 / data=d1;
   group 2 / data=d2;
   model 1 / group=1;
      mstruct var=V1-V6;
      matrix _COV_ [1,1] = 6*vparm (8.),
                   [2,]  = cv21,
                   [3,]  = cv31,
                   [4,]  = cv41 cv42 cv43,
                   [5,]  = cv51 cv52 cv53 cv54,
                   [6,]  = cv61 cv62 cv63 cv64 cv65;
   model 2 / group=2;
      refmodel 1;
      matrix _COV_ [2,]  = 3.,
                   [3,2] = cv32,
                   [4,]  = . . . ,
                   [5,]  = . . . ,
                   [6,]  = . . . ;
   run;
```

In these statements, you specify model 2 by referring to model 1 in the REFMODEL statement. Hence, model 2 is called the new model that refers to the old model, model 1. Because they are not respecified in the new model, all parameters on the diagonal of the covariance matrix are duplicated from the old model for the new model. Similarly, parameter locations associated with the cv54, cv64, and cv65 parameters are also duplicated in the new model.

An added parameter in the new model is cv32 for the covariance between V3 and V2. This parameter location is not specified in the old model.

In the new model, parameters for the covariances between the variable sets {V1 V2 V3} and {V4 V5 V6} are all deleted from the old model. The corresponding parameter locations for these covariances are given missing values '.' in the new model, indicating that they are no longer free parameters as in the old model. Deleting these parameters amounts to setting the corresponding covariances to fixed zeros in the new model.

Finally, covariance between V2 and V1 is changed from a free parameter cv21 in the old model to a fixed constant 3 in the new model. This illustrates the replacement rule of the respecification syntax.

NLINCON Statement

NLINCON | **NLC** *constraint* < , *constraint* . . . > **;**

where *constraint* represents one of the following:

- *number operator variable-list number operator*
- *variable-list operator number*
- *number operator variable-list*

You can specify nonlinear equality and inequality constraints with the NLINCON or NLC statement. The QUANEW optimization subroutine is used when you specify nonlinear constraints by using the NLINCON statement.

The syntax of the NLINCON statement is similar to that of the BOUNDS statement, except that the NLINCON statement must contain the names of variables that are defined in the program statements and are defined as continuous functions of parameters in the model. They must not be confused with the variables in the data set.

As with the BOUNDS statement, one- or two-sided constraints are allowed in the NLINCON statement; equality constraints must be one sided. Valid operators are <=, <, >=, >, and = (or, equivalently, LE, LT, GE, GT, and EQ).

PROC TCALIS cannot enforce the strict inequalities < or > but instead treats them as <= and >=, respectively. The listed nonlinear constraints must be separated by commas. The following is an example of the NLINCON statement that constrains the nonlinear parametric function $x_1 * x_1 + u_1$ to a fixed value of 1:

```
nlincon    xx = 1;
xx = x1 * x1 + u1;
```

Note that x1 and u1 are parameters defined in the model. The following three NLINCON statements, which require xx1, xx2, and xx3 to be between zero and ten, are equivalent:

```
nlincon  0. <= xx1-xx3,
               xx1-xx3 <= 10;
nlincon 0. <= xx1-xx3 <= 10.;
nlincon 10. >= xx1-xx3 >= 0.;
```

NLOPTIONS Statement

NLOPTIONS *options* **;**

Many options that are available in SAS/OR PROC NLP can be specified for the optimization subroutines in PROC TCALIS by using the NLOPTIONS statement. The NLOPTIONS statement provides more displayed and file output control on the results of the optimization process, and it permits the same set of termination criteria as in PROC NLP. These are more technical options that you might not need to specify in most cases.

Several statistical procedures support the use of NLOPTIONS statement. The syntax of NLOPTIONS statement is common to all these procedures and can be found in the section "Nonlinear Optimization: The NLOPTIONS Statement" on page 391 in Chapter 18, "Shared Concepts and Topics."

See the section "Use of Optimization Techniques" on page 6923 for more information about the use of optimization techniques in PROC TCALIS.

OUTFILES Statement

OUTFILES | **OUTFILE** *file_option* < *file_option . . .* > **;**

where *file_option* represents one of the following:

- OUTMODEL | OUTRAM= *file_name* [MODEL= *int_list* < , *int_list* >]
- OUTSTAT= *file_name* [GROUP= *int_list* < , *int_list* >]
- OUTWGT= *file_name* [GROUP= *int_list* < , *int_list* >]

with *file_name* representing an output file name and *int_list* representing list of model or group numbers

Use the OUTFILES statement when you need to output multiple-group or multiple-model information to output files in a complex way. In each OUTFILES statement, each possible *file_option* should appear no more than once. However, as needed, you can use the OUTFILES statement more than once. For example, suppose you want to create two OUTWGT= files for different sets of groups. You can specify the OUTFILES statement twice, as shown in the following specification:

```
outfiles outwgt=file1 [group=1,2];
outfiles outwgt=file2 [group=3,4];
```

In the first OUTFILES statement, the weights for groups 1 and 2 are output to the file file1. In the second OUTFILES statement, the weights for groups 3 and 4 are output to the file file2.

When the OUTMODEL=, OUTSTAT=, or OUTWGT= option is intended for **all** groups or models, you can simply specify the option in the PROC TCALIS statement. Only when you need to output the group (model) information from **more than one** group (model), **but not all** groups (models), to a single output file does the use the OUTFILES statement become necessary. For example, consider the following specification:

```
proc tcalis method=gls;
   outfiles outmodel=outmodel [model=1,3]
            outwgt=outwgt [group=1,2]
            outstat=outstat [group=2,3];
   group 1 / data=g1;
   group 2 / data=g2;
   group 3 / data=g3 outwgt=outwgt3;
   model 1 / group=1;
      factor N=3;
   model 2 / group=2;
      factor N=2;
   model 3 / group=3;
      factor N=3;
   run;
```

You fit three different factor models to three groups: model 1 for group 1, model 2 for group 2, and model 3 for group 3. In the OUTFILES statement, you output model information from models 1 and 3 to an output file named outmodel, weight matrices from groups 1 and 2 to outwgt, and statistics from groups 2 and 3 to outstat. In each of these output files, you have information from more than one (but not all) groups or models. In the GROUP statement for group 3, you have another OUTWGT= file named outwgt3 for group 3 alone.

Note that you cannot specify the preceding output file organization by using the following statements:

```
proc tcalis method=gls;
   group 1 / data=g1 outwgt=outwgt;
   group 2 / data=g2 outwgt=outwgt outstat=outstat;
   group 3 / data=g3 outwgt=outwgt3 outstat=outstat;
   model 1 / group=1 outmodel=outmodel;
      factor N=3;
   model 2 / group=2;
      factor N=2;
   model 3 / group=3 outmodel=outmodel;
      factor N=3;
   run;
```

This specification will not work because SAS forbids the repeated specification of the same output file in the same PROC TCALIS run. That is, you cannot specify OUTWGT=outwgt, OUTSTAT=outstat, or OUTMODEL=outmodel more than once in the PROC TCALIS run without causing file invocation problems (however, multiple specification of the same input file is not a problem).

The OUTFILES statement is intended for arranging output files in a complex way. The use of the OUTFILES statement is unnecessary in the following situations:

- If you have a single-sample analysis, you do not need to use the GROUP statement. As a result, you can simply use the OUTSTAT= or OUTWGT= options in the PROC TCALIS statement for specifying the output destinations. Therefore, the OUTFILES statement is not needed.
- If you have a single model in your analysis, you do not need to use the MODEL statement. As a result, you can simply use the OUTMODEL= options in the PROC TCALIS statement for specifying the output destination. Therefore, the OUTFILES statement is not needed.
- If you have multiple groups or multiple models in your analysis and information for all groups or models is output to the same file, you do not need to use the OUTFILES statement. You can simply use the OUTSTAT=, OUTWGT=, or OUTMODEL= options in the PROC TCALIS statement because the output file information is automatically propagated from the PROC TCALIS statement to the groups or models.
- If you have multiple groups or multiple models in your analysis and each group or model has a unique output data file destination (including cases where some groups or models might not have any output files), you do not need to use the OUTFILES statement. You can simply specify the OUTSTAT=, OUTWGT=, or OUTMODEL= options in the GROUP or MODEL statements.

PARAMETERS Statement

PARAMETERS | PARMS *parameter(s)* << = > *number(s)* >
<< , > *parameter(s)* << = > *number(s)* > . . . > ;

The PARAMETERS statement defines additional parameters that are not specified in your models. You can specify more than one PARAMETERS statement. The *parameters* can be followed by an equal sign and a number list. The values of the *numbers* list are assigned as initial values to the preceding parameters in the *parameters* list. For example, each of the following statements assigns the initial values ALPHA=.5 and BETA=-.5 for the parameters used in SAS programming statements:

```
parameters alfa beta=.5 -.5;
parameters alfa beta (.5 -.5);
parameters alfa beta .5 -.5;
parameters alfa=.5 beta (-.5);
```

The number of parameters and the number of values do not have to match. When there are fewer values than parameter names, either the RANDOM= or START= option is used. When there are more values than parameter names, the extra values are dropped. Parameters listed in the PARAMETERS statement can be assigned initial values by program statements or by the START= or RANDOM= option in the PROC TCALIS statement.

Do not confuse the PARAMETERS statement with the VAR statement. While you specify the parameters of the model in the PARAMETERS statement, you specify analysis variables in the VAR statement. See the VAR statement on page 6818 for more details.

CAUTION: The OUTMODEL= or OUTMODEL= data sets do not contain any information about the PARAMETERS statement or the SAS programming statements.

PARTIAL Statement

PARTIAL *variables* ;

If you want the analysis to be based on a partial correlation or covariance matrix, use the PARTIAL statement to list the variables used to partial out the variables in the analysis. You can specify only one PARTIAL statement within the scope of each GROUP or PROC TCALIS statement.

PATH Statement

PATH *path* < , *path* . . . > **;**

where *path* represents:
variable direction variable parameter-spec
and *direction* represents one of the following: –>, >, <–, <

In the PATH statement, you specify the paths in your model. You can specify at most one PATH statement in a model, within the scope of either the PROC TCALIS statement or a MODEL statement. To complete the PATH model specifications, you might need to add some subsidiary model specification statements. The following is the syntax for the PATH modeling language:

PATH *path lists* **;**
PVAR *partial variance parameters* **;**
PCOV *partial covariance parameters* **;**
MEAN *mean parameters* **;**

In the PATH statement, you specify paths of variables in your model, excluding paths from the errors or disturbances. Paths in the PATH statement are separated by commas. In the PVAR statement, you specify parameters for either variances or partial variances. In the PCOV statement, you specify parameters for either covariances or partial covariances. In the MEAN statement, you specify parameters for either means or intercepts. For details about these subsidiary model specification statements, refer to the syntax of the individual statements.

In each path, you have one outcome variable and one predictor variable. The variables in a path can be manifest or latent. The specified direction determines the predictor and outcome variables. The *variable* being aimed at is the outcome variable and the other *variable* is the predictor variable. The path coefficient parameter is specified after each path in *parameter-spec*. For example, in the following statement you specify a model with four paths, each with predictor variable F1 and one outcome manifest variable.

```
PATH
   V1  <-  F1    1,
   V2  <-  F1    b2,
   V3  <-  F1    b2,
   V4  <-  F1    b4 (.4);
```

As shown in the example, the *parameter-spec* is one of the following three types:

- fixed path coefficients

  The *parameter-spec* for the first path to V1 is a fixed path coefficient at 1.

- free or constrained parameters without starting values

  The *parameter-specs* for the second and third paths to V2 and V3, respectively, are the same path coefficient parameter b2 with no starting value given. Because the same parameter name is used for the two path coefficients, the coefficients are constrained in the model.

- free or constrained parameters with starting values

  The *parameter-spec* for the last path to V4 is a free parameter named b4 with a starting value at 0.4.

If an outcome variable has more than one predictor variable, you have more than one path specification for the outcome variable. The following PATH statement shows an extension of the previous specification.

```
PATH
   V1  <-  F1    1,
   V3  ->  V1    b1 (.2),
   V2  <-  F1    b2,
   V3  <-  F1    b2,
   V4  <-  F1    b4 (.4);
```

As compared with the previous example, an additional path is specified in second path specification of the current PATH statement. In the specification `V3` is a predictor of `V1`, in addition to the predictor `F1` in the first path specification. The additional path `V3 –> V1` has a path coefficient parameter named `b1`, with a starting value at 0.2.

Note that because the display of path results follows the way you specify the paths, you should arrange the order of the input paths in the most desirable way.

Modifying a PATH Model from a Reference Model

In this section, it is assumed that you use a REFMODEL statement within the scope of a MODEL statement and the reference model (or base model) is also a PATH model. The reference model is referred to as the old model, while the model that makes reference to this old model is referred to as the new model. If the new model is not intended to be an exact copy of the old model, you can use the following extended PATH modeling language to make modifications within the scope of the MODEL statement for the new model. The syntax is the same as the ordinary PATH modeling language, as in the section "PATH Statement" on page 6799. The respecification syntax for a PATH model is as follows:

PATH *path lists* **;**
  PVAR *partial variance parameters* **;**
  PCOV *partial covariance parameters* **;**
  MEAN *mean parameters* **;**

The new model is formed by integrating with the old model in the following ways:

Duplication: If you do not specify in the new model a parameter location that exists in the old model, the old parameter specification is duplicated in the new model.

Addition: If you specify in the new model a parameter location that does not exist in the old model, the new parameter specification is used in the new model.

Deletion: If you specify in the new model a parameter location that also exists in the old model and the new parameter is denoted by the missing value '.', the old parameter specification is not copied into the new model.

Replacement: If you specify in the new model a parameter location that also exists in the old model and the new parameter is not denoted by the missing value '.', the new parameter specification replaces the old one in the new model.

For example, consider the following specification of a two-group analysis:

```
proc tcalis;
   group 1 / data=d1;
   group 2 / data=d2;
   model 1 / group=1;
      path
         V1 <- F1   1.,
         V2 <- F1   load1,
         V3 <- F1   load2,
         F1 <- V4   b1,
         F1 <- V5   b2,
         F1 <- V6   b3;
      pvar
         E1-E3    = ve1-ve3,
         F1       = vd1,
         V5-V6    = phi4-phi6;
      pcov
         V1 V2    = cve12;
   model 2 / group=2;
      refmodel 1;
      path
         V3 <- F1   load1,
      pcov
         V1 V2    =  .,
         V2 V3    = cve23;
   run;
```

You specify model 2 by referring to model 1 in the REFMODEL statement. Model 2 is the new model that refers to the old model, model 1. This example illustrates the four types of model integration rules for the new model:

- Duplication: All parameter specifications, except for the partial covariance between V1 and V2 and the V3 <– F1 path in the old model, are duplicated in the new model.
- Addition: The parameter cve23 for the partial covariance between V2 and V3 is added in the new model because there is no corresponding specification in the old model.
- Deletion: The specification of partial covariance between V1 and V2 in the old model is not copied into the new model, as indicated by the missing value '.' specified in the new model.
- Replacement: The new path V3 <– F1 replaces the same path in the old model with parameter load1 for the path coefficient. Thus, in the new model paths V3 <– F1 and v2 <– F1 are now constrained to have the same path coefficient parameter load1.

PCOV Statement

PCOV *assignment* < , *assignment* . . . > **;**

where *assignment* represents:

variables < ∗ *variables2* > = *parameter-spec*

The PCOV statement defines the covariance or partial covariance parameters in the PATH model. You can use the PCOV statement with the PATH modeling language only. On the left-hand side of each assignment, variables involved are listed in the variables and variables2 lists. In the parameter-spec list, covariance or partial covariance parameters are specified. Names in the parameter-spec represent covariance or partial covariance parameters to estimate. Numbers enclosed in parentheses in the parameter-spec list are initial values for the parameters. Numbers not enclosed in parentheses in the parameter-spec list are fixed values for covariances or partial covariances. The *assignment*s in the PCOV statement must be separated by commas. You can specify only one PCOV statement in each model specification.

The syntax of the PCOV statement is the same as the COV statement. Refer to the COV statement on page 6747 for details about specifying within- and between- list (partial) covariances.

The concept behind the PCOV statement, however, is broader than that of the COV statement. The PCOV statement supports the partial covariance parameter specification in addition to the covariance parameter specification. The COV statement supports only the latter type of parameters. This discrepancy is also reflected from the sets of *variables* and *variables2* you can use in the PCOV statement. In the COV statement, variables on the left-hand side of an *assignment* must be exogenous. However, in the PCOV statement, both exogenous and endogenous variables can be specified. If both variables are exogenous in a specification of the PCOV statement, you are defining a covariance parameter between the variables. If both variables are endogenous, you are defining a partial covariance parameter between of the variables. This partial covariance is usually interpreted as error covariance between the two endogenous variables. If one variable is exogenous while the other is endogenous, you are defining a covariance parameter between the exogenous variable and the error term for the endogenous variable.

Automatic Covariance Parameters among Exogenous Manifest Variables

In the PATH model, all covariances among exogenous manifest variables are automatically free parameters to estimate unless you specify otherwise in the PCOV statement. The parameter names for these covariances will be generated by PROC TCALIS with the prefix _Add, appended with unique integer for each occurrence. If you want any of these covariances to be fixed values, you must specify them explicitly in the PCOV statement.

By default covariances among exogenous latent variables are fixed zeros unless they are specified otherwise in the PCOV statement. The same applies to any partial covariance among endogenous variables and to any partial covariance between an exogenous latent variable and an exogenous manifest variable.

This is different from the case of partial variance parameters specified in the PVAR statement. Variances of **all** exogenous variables and partial variances of **all** endogenous variables are automatically free parameters if they are not specified otherwise in the PVAR statement.

Modifying a Parameter Specification from a Reference Model

If you define a new PATH model by using a reference (or base) model in the REFMODEL statement, in some situations you might want to modify some parameter specifications from the PCOV statement of the reference model before transferring the specifications to the new model. If you want to change a particular PCOV parameter specification from the reference model, you can simply put the corresponding parameter location with a new parameter specification in the new model. If you want to delete a particular parameter location from the reference model, you can put that parameter location with the missing value '.' in the new model. For example, suppose you are defining a new PATH model by using the REFMODEL statement and the covariance between variables F1 and V2 is defined as a fixed or free parameter in the reference model. If you do not want this parameter specification to be copied into your new model, you can use the following specification in the new model:

```
PCOV F1 V2 = .;
```

Note that the missing value syntax is valid only when used with the REFMODEL statement. See the section "Modifying a PATH Model from a Reference Model" on page 6800 for a more detailed example of PATH model respecification.

In the PATH model, all covariances among exogenous manifest variables are free parameters to estimate unless you specify otherwise in the PCOV statement (see the section "Automatic Covariance Parameters among Exogenous Manifest Variables" on page 6802). This is also true for PATH models specified under the REFMODEL statement. In the reference model, covariance parameters generated by PROC TCALIS are named with the _Add prefix and appended with integer suffixes. These parameter specifications in the reference model do **not** transfer to the new model. Only after the new model is resolved from the reference model, the REFMODEL statement options, the RENAMEPARM statement, and the model respecifications are the additional covariance parameters with the _Add prefix generated for the unspecified covariances among exogenous manifest variables. In this way, the covariance parameters in the new model are not constrained to be the same as the corresponding parameters in the reference model. If you want any of these covariance parameters constrained across the models, you must specify them explicitly in the PCOV statement of the reference model so that the covariance specifications are transferred to the new model.

PVAR Statement

PVAR *assignment* < , *assignment . . .* > **;**

where *assignment* represents:
variables = *parameter-spec*

The PVAR statement defines the variance or partial variance parameters in FACTOR and PATH models. In each *assignment* in the PVAR statement, you specify *variables* on the left-hand side and the *parameter-spec* on the right-hand side of an equal sign. The *assignments* in the PVAR statement must be separated by commas. The *variables* listed on the left-hand side of the equal sign are any variables defined in the model. If a variable in the list is an exogenous variable, then the correspond-

ing parameter in the *parameter-spec* list is a variance parameter of the variable. If a variable in the list is an endogenous variable, then the corresponding parameter in the *parameter-spec* is a partial variance parameter of the variable. In most cases, the partial variance of an endogenous variable can be interpreted as the error variance for the variable.

The syntax of the PVAR statement is exactly the same as that of the STD statement. As in the STD statement, in the PVAR statement you can specify free or constrained parameters with or without starting or fixed values. You can also use prefixes to generate parameter names. Refer to the STD statement on page 6815 for details and examples.

The difference between the PVAR and the STD statements is conceptual in nature. In the STD statement, you can specify only variance parameters for exogenous variables. However, in the PVAR statement, both exogenous and endogenous variables can be put on the *variables* list. This treatment is needed in the FACTOR and PATH modeling languages because error variables in these models are not explicitly named.

Automatic Variance and Partial Variance Parameters among Variables

If the name of a variable (manifest or latent) in the FACTOR or PATH model does not appear in the PVAR statement, the variance or partial variance of the variable is assumed to be a parameter to estimate. These parameters will be named automatically with the prefix _Add and appended with integer suffixes. Therefore, the entire PVAR statement is optional in the sense that all variance and partial variance parameters will be generated automatically for the FACTOR or PATH model in PROC TCALIS. However, if you need to constrain these parameters, or assign fixed values, the PVAR statement specification must be used.

Modifying a Parameter Specification from a Reference Model

If you define a new FACTOR or PATH model by using a reference (or base) model in the REFMODEL statement, in some situations you might want to modify some parameter specifications from the PVAR statement of the reference model before transferring the specifications to the new model. If you want to change a particular PVAR parameter specification from the reference model, you can simply put the corresponding parameter location with a new parameter specification in the new model. If you want to delete a particular parameter location from the reference model, you can put that parameter location with the missing value '.' in the new model. For example, suppose you are defining a new PATH model by using the REFMODEL statement and the partial variance of variable `V2` is defined as a fixed or free parameter in the reference model. If you do not want this parameter specification to be copied into your new model, you can use the following specification in the new model:

```
pvar V2 = .;
```

Note that the missing value syntax is valid only when you use the REFMODEL statement. See the section "Modifying a FACTOR Model from a Reference Model" on page 6758 for a more detailed example of FACTOR model respecification. See the section "Modifying a PATH Model from a Reference Model" on page 6800 for a more detailed example of PATH model respecification.

In confirmatory factor or PATH models, all variances or partial variances among variables are free parameters to estimate unless you specify otherwise in the PVAR statement (see the section "Automatic Variance and Partial Variance Parameters among Variables" on page 6804). This is also true for confirmatory factor and LINEQS models specified under the REFMODEL statement. In the reference model, (partial) variance parameters generated by PROC TCALIS are named with the _Add prefix and appended with integer suffixes. These parameter specifications in the reference model do **not** transfer to the new model. Only after the new model is resolved from the reference model, the REFMODEL statement options, the RENAMEPARM statement options, and model re-specifications do the remaining unspecified (partial) variances of variables generate additional (partial) variance parameters with the _Add prefix and integer suffixes. In this way, the (partial) variance parameters in the new model are not constrained to be the same as the corresponding parameters in the reference model. If you want any of these (partial) variance parameters to be constrained across the models, you must specify them explicitly in the PVAR statement of the reference model so that the (partial) variance specification is transferred to the new model.

RAM Statement

RAM *list-entry* < , *list-entry* . . . > **;**

where *list-entry* represents:
parameter-type variable1 < *direction* > < *variable2* > *parameter-spec*
and *direction* can be one of the following: –>, >, <–, <

RAM stands for the reticular action model, developed by McArdle(1980). In the RAM statement, you specify various kinds of parameters in the model. In each *list-entry*, you specify the type of parameter in *parameter-type*, variables involved in *variable1* and, if applicable, in *variable2*, path direction in *direction*, parameter name and value in *parameter-spec*. *List-entries* must be separated by commas.

Types of RAM Parameters

The first keyword in each RAM *list-entry* is the parameter type, represented as *parameter-type* in the syntax. You can use one of the following keywords (not case-sensitive) for *parameter-type*:

_A_ or _RAMA_	a parameter in the A matrix of the RAM model (see the PATH keyword)
_P_ or _RAMP_	a parameter in the P matrix of the RAM model (see the PCOV and PVAR keywords)
_W_ or _RAMW_	a parameter in the vector for means and intercepts (see the MEAN and INTERCEPT keywords)
INTERCEPT	the intercept of an endogenous variable, manifest or latent
MEAN	the mean of an exogenous variable, manifest of latent
PATH	the path coefficient for a pair of outcome and predictor variables
PCOV	the partial covariance or covariance between a pair of variables
PVAR	the partial variance or variance of a variable

Keywords _A_, _P_, and _W_ refer to the matrix elements of the RAM model matrices _A_, _P_, and _W_, respectively. See the section "Model Matrices in the RAM Model" on page 6856 for detailed descriptions of these matrices. Elements in matrix _A_ are path coefficients; therefore the _A_ keyword is the same as the PATH keyword for specifying these matrix elements. Elements in matrix _P_ are either (partial) variances or (partial) covariances; therefore the _P_ keyword is the same as the PVAR or PCOV keyword for specifying these matrix elements. Elements in matrix _W_ are either means or intercepts; therefore the _W_ keyword is the same as the MEAN or INTERCEPT keyword for specifying these matrix elements.

Path Coefficients Parameters with Keyword _A_ or PATH

To specify the paths and the associated coefficients, you can use either the _A_ or PATH keyword for *parameter-type*. A pair of variables must be specified in *variable1* and *variable2*. The *direction* must also be inserted between the pair of variables with the PATH keyword. The following is an example of the list-entries for path coefficients:

```
path    a -> b    x1,
path    c <- b    x2,
```

The first path is from variable a to variable b with a path coefficient parameter named x1. The second path is from variable b to variable c with a path coefficient parameter named x2. However, if you use the _A_ keyword, you must not specify the *direction*, as it is implicitly assumed to be an arrow pointing to left, <–. For example, the following specification using the _A_ keyword is equivalent to the previous specification.

```
_A_    b   a     x1,
_A_    c   b     x2,
```

Partial Variance and Variance Parameters with Keyword _P_ or PVAR

To specify the variance or partial variance parameters, you can use the _P_ or PVAR keyword for *parameter-type*. If you use _P_, *variable1* and *variable2* must be the same variable name. If you use PVAR, only *variable1* needs to be specified. Specification for *direction* should be omitted. If the variable is exogenous, you are defining a variance parameter. Otherwise, you are defining a partial variance parameter. Partial variance in the RAM model can be interpreted as the error variance for an endogenous variable.

For example, assume that b is an exogenous variable in your model. In the following specification, you define the variance of b as a parameter named v1, with an initial estimate of 1.0:

```
PVAR    b     v1 (1.0),
```

This specification is equivalent to the following specification by using the _P_ keyword for *parameter-type*:

```
_P_   b  b    v1 (1.0),
```

Partial Covariance and Covariance Parameters with Keyword _P_ or PCOV

To specify the covariance or partial covariance parameters, you can use the _P_ or PCOV keyword for *parameter-type*. You must specify both *variable1* and *variable2*, where *variable1* and *variable2* are different variable names. Specification for *direction* should be omitted. Whereas it is a semantic error if *variable1* and *variable2* are the same name with the PCOV keyword, it is permissible to do so with the _P_ keyword. However, for the latter you are supposed to define a parameter location for the variance or partial variance of a variable, instead of the covariance or partial covariance parameter that is considered in this section.

The following shows a specification of a covariance parameter:

```
PCOV   a  b    v12 (0.5),
```

If both a and b are exogenous variables in your model, the preceding code specifies the covariance between a and b as a free parameter named v12, with a starting value at 0.5. Equivalently, you can use the _P_ keyword, as shown in the following specification:

```
_P_   a  b    v12 (0.5),
```

Note that if both a and b are both endogenous variables in your model, you can still use the preceding specification. But the interpretation is different. For endogenous a and b, v12 is treated as a partial covariance parameter. In almost all situations, this partial covariance is interpreted as the error covariance for variables a and b. Or, the model assumes correlated errors associated with variables a and b.

In fact, you can also specify the partial covariance between an exogenous variable and an endogenous variable. Suppose that a is exogenous and b is endogenous in your model. The preceding specifications are for specifying the covariance parameter between a and the error term for b. This is certainly a cumbersome interpretation. Fortunately, covariances between exogenous variables and error variables are considered inelegant in most statistical models. This kind of specification should be rare in practice.

Means and Intercepts with Keyword _W_, INTERCEPT, or MEAN

To specify the mean or intercept parameters, you can use the _W_, INTERCEPT, or MEAN keyword for *parameter-type*. You must specify *variable1* in the *list-entry*. Specifications for *variable2* and *direction* are not applicable and should be omitted. With the MEAN keyword, *variable1* is supposed to be an exogenous variable. With the INTERCEPT keyword, *variable1* is supposed to be an endogenous variable. With the _W_ keyword, however, *variable1* can be either exogenous or endogenous. In this case, whether a mean or an intercept parameter is being specified is determined by the TCALIS procedure.

For example, in the following specification the mean of f1 is a free parameter named mean1 while the intercept for a is a constant value 2.0:

```
mean        f1    mean1,
intercept    a    2.0,
```

Equivalently, you can specify using the _W_ keyword, as shown in the following:

```
_W_          f1    mean1,
_W_           a    2.0,
```

Variable Names for *variable1* and *variable2*

In each *list-entry*, you have to specify a variable name for *variable1* and possibly a variable name for *variable2* also. PROC TCALIS needs to distinguish names for observed and latent variables in your model. Variable names that are referenced in the input data set are for observed variables; otherwise, they are for latent variables. Therefore, you can make up any latent variable names in the RAM model specification, as long as these names follow the general SAS naming conventions and do not match any of the names in the input data set. Notice that in the RAM model specification, all latent variable names are for the latent factors, and names for error or disturbance terms are never needed. Error variances and covariances can be specified using the corresponding endogenous variables names as references in the *list-entries* with the PCOV or PVAR keyword for parameter-type. Hence, creating names for error or disturbance terms is not necessary in the RAM model specification.

Parameter Names and Initial Values

After you specify the type of parameter in *parameter-type*, the associated variables in *variable1* and *variable2* (when applicable), and the *direction* for paths (if applicable), the last piece information of *list-entry* is *parameter-spec*, which can be one of the following three types:

- fixed constants

  For example, in the following specification the intercept of variable a is fixed at 2.0.

  ```
  intercept    a    2.0,
  ```

- free or constrained parameter names without starting values

  For example, in the following specification the mean of variable f1 is a parameter named mean1.

  ```
  mean         f1    mean1,
  ```

  The starting value of mean1 is generated by PROC TCALIS.

- free or constrained parameter names with starting values

  For example, in the following specification the variance (if b is exogenous in the model) or error variance (if b is endogenous in the model) of variable b is a parameter named v1, with a starting value 1.0.

  ```
  PVAR         b     v1 (1.0),
  ```

Automatic Variance and Partial Variance Parameters

If the partial variance or variance parameter for any variable (manifest or latent) in the RAM model is not explicitly specified in the RAM *list-entries*, it will be added automatically as a free parameter. Each parameter is named with the prefix _Add and appended with a unique integer. Therefore, PVAR *parameter-type* entries for variables are optional when they are free and unconstrained parameters. However, if you need to set constraints on these parameters or to assign fixed parameter values, PVAR *list-entries* for the corresponding variables are necessary.

Automatic Covariance Parameters

All covariances among exogenous *manifest* variables in the RAM model are free parameters automatically unless specified otherwise in the *list-entries* with the PCOV keyword for *parameter-type*. Each automatic covariance parameter is named by PROC TCALIS with the _Add prefix and appended with an unique integer. If you want any of these covariances to be fixed values or constrained via name references, you must specify them explicitly in the PCOV *list-entries*.

Covariances among exogenous *latent* variables by default are fixed zeros unless specified otherwise in the PCOV *list-entries*. The same applies to any partial covariance among endogenous variables and to any partial covariance between an exogenous latent variable and an exogenous manifest variable.

Automatic Mean Parameters

All means of exogenous manifest variables in the RAM model are free parameters automatically unless specified otherwise in the MEAN *list-entries*. Each automatic mean parameter is named by PROC TCALIS with the _Add prefix and appended with an unique integer. If you want any of these means be fixed values or constrained via name references, you must specify them explicitly in the MEAN *list-entries*.

Modifying a RAM Model from a Reference Model

In this section, it is assumed that you use a REFMODEL statement within the scope of a MODEL statement and that the reference model (or base model) is also a RAM model. The reference model will be referred to as the old model, while the model that makes reference to this old model is referred to as the new model. If the new model is not intended to be an exact copy of the old model, you can use the following extended RAM modeling language to make modifications on the model specification. The syntax for modifications is the same as the ordinary RAM modeling language (see the section "RAM Statement" on page 6805). The syntax for respecifying/modifying the RAM model is as follows.

RAM *list-entry* < *, list-entry . . .* > **;**

where *list-entry* represents:

parameter-type variable1 < *direction* > < *variable2* > *parameter-spec*

and *direction* represents one of the following: –>, >, <–, <

The new model is formed by integrating with the old model in the following ways:

Duplication: If you do not specify in the new model a parameter location that exists in the old model, the old parameter specification is duplicated in the new model.

Addition: If you specify in the new model a parameter location that does not exist in the old model, the new parameter specification is added to the new model.

Deletion: If you specify in the new model a parameter location that also exists in the old model and the new parameter is denoted by the missing value '.', the old parameter specification is not copied into the new model.

Replacement: If you specify in the new model a parameter location that also exists in the old model and the new parameter is not denoted by the missing value '.', the new parameter specification replaces the old one in the new model.

For example, consider the following two-group analysis:

```
proc tcalis;
   group 1 / data=d1;
   group 2 / data=d2;
   model 1 / group=1;
      ram
         path V1 <- F1   1.,
         path V2 <- F1   load1,
         path V3 <- F1   load2,
         path F1 <- V4   b1,
         path F1 <- V5   b2
         path F1 <- V6   b3
         pvar E1         ve1,
         pvar E2         ve2,
         pvar E3         ve3,
         pvar F1         vd1,
         pvar V4         phi4,
         pvar V5         phi5,
         pvar V6         phi6,
         pcov V1 V2      cve12;
   model 2 / group=2;
      refmodel 1;
      ram
         path V3 <- F1   load1,
         pcov V1 V2      .    ,
         pcov V2 V3      cve23;
   run;
```

In this example, you specify model 2 by referring to model 1 in the REFMODEL statement. Model 2 is the new model which refers to the old model, model 1. This example illustrates the four types of model integration process by PROC TCALIS:

- Duplication: All parameter specifications, except for the partial covariance between V1 and V2, and the V3 <– F1 path in the old model are duplicated in the new model.
- Addition: The parameter cve23 for the partial covariance between V2 and V3 is added in the new model.
- Deletion: The specification of partial covariance between V1 and V2 in the old model is not copied into the new model, as indicated by the missing value '.' in the specification.
- Replacement: The new path V3 <– F1 replaces the same path in the old model with another parameter for the path coefficient. Thus, in the new model paths V3 <– F1 and V2 <– F1 are constrained to have the same path coefficient parameter load1.

In the RAM model, partial variances or variances of all variables, means of exogenous mainfest variables, and the covariances among exogenous mainfest variables are automatically free parameters unless you specify otherwise in the *list-entries* of the RAM statement. See the sections "Automatic Variance and Partial Variance Parameters" on page 6809, "Automatic Covariance Parameters" on page 6809, and "Automatic Mean Parameters" on page 6809 for more details. This is also true for RAM models specified under the REFMODEL statement. In the reference model, parameters generated by PROC TCALIS are named with the _Add prefix and appended with integer suffixes. These automatically generated parameters in the reference model do **not** transfer to the new model. Only after the new model is resolved from the reference model, the REFMODEL statement options, the RENAMEPARM statement, and the model respecification are the automatic parameters of the new RAM model generated. In this way, the generated parameters in the new model are not constrained to be the same as the corresponding parameters in the reference model. If you want any of these parameters be constrained across the models, you must specify them explicitly in the *list-entries* of the RAM statement of the reference model so that these specifications are transferred to the new model via the REFMODEL statement.

REFMODEL Statement

REFMODEL | **BASEMODEL** *model_number* < / *options* > **;**

where *model_number* represents a model number between 1 and 9999, inclusively, and *options* are for renaming parameters.

The REFMODEL statement is not a modeling language itself. It is a tool for referencing and modifying models. It is classified into one of the modeling languages because its role is similar to other modeling languages.

REFMODEL *model_number* < / *options* > **;**
  RENAMEPARM *parameter renaming* **;**
  main model specification statement **;**
  subsidiary model specification statements **;**

In the REFMODEL statement, you put the *model_number* of the model you are making reference to. The reference model must be well-defined in the same PROC TCALIS run. In the *options*, you

can rename all the parameters in the reference model by adding prefix or suffix so that the current model has a new set of parameters.

The following options are available in the REFMODEL statement:

ALLNEWPARMS
appends to the parameter names in the reference model with _mdl and then an integer suffix denoting the model number of the current model. For example, if qq is a parameter in the reference model for a current model with model number 3, then this option creates qq_mdl3 as a new parameter name.

PARM_PREFIX=*prefix*
inserts to all parameter names in the reference model with the *prefix* provided. For example, if qq is a parameter in the reference model for a current model, then **PARM_PREFIX=pre_** creates pre_qq as a new parameter name.

PARM_SUFFIX=*suffix*
appends to all parameter names in the reference model with the *suffix* provided. For example, if qq is a parameter in the reference model for a current model, then **PARM_SUFFIX=_suf** creates qq_suf as a new parameter name.

Instead of renaming all parameters, you can also rename parameters individually by using the RENAMEPARM statement within the scope of the REFMODEL statement.

You can also add the main and subsidiary model specification statements to modify a particular part from the reference model. For example, you might like to add or delete some equations or paths, or to change a fixed parameter to a free parameter or vice versa in the new model. All can be done in the respecification in the main and subsidiary model specification statements within the scope of the MODEL statement to which the REFMODEL statement belongs. Naturally, the modeling language used in respecification must be the same as that of the reference model. See the individual statements for modeling languages for the syntax of respecification. Note that when you respecify models by using the main and subsidiary model specification statements together with the RENAMEPARM statement or the REFMODEL options for changing parameter names, the parameter name changes occur after respecifications.

RENAMEPARM Statement

RENAMEPARM *assignment* < , *assignment . . .* > **;**

where *assignment* represents:
old_parameters = *parameter-spec*

You can use the RENAMEPARM statement to rename parameters or to change the types of parameters of a reference model so that new parameters are transferred to the new model in question. The RENAMEPARM statement is a subsidiary model specification statement that should be used together with the REFMODEL statement. The syntax of the RENAMEPARM statement is similar to that of the STD statement—except that in the RENAMEPARM statement, you put parameter names on the left-hand side of equal signs, whereas you put variable names on the left-hand side in

the STD statement. You can use no more than one RENAMEPARM statement within the scope of each REFMODEL statement.

In the REFMODEL statement, you transfer all the model specification information from a base model to the new model being specified. The RENAMEPARM statement enables you to modify the parameter names or types in the base model before transferring them to the new model. For example, in the following example, you define model 2, which is a new model, by referring it to model 1, the base model, in the REFMODEL statement.

```
model 1;
   lineqs
      V1 =    F1 + E1,
      V2 = b2 F1 + E2,
      V3 = b3 F1 + E3,
      V4 = b4 F1 + E4;
   std F1 = vF1,
      E1-E4 = ve1-ve4;
model 2;
   refmodel 1;
   renameparm ve1-ve4=new1-new4, b2=newb2(.2), b4=.3;
```

Basically, the LINEQS model specification in model 1 is transferred to model 2. In addition, you redefine some parameters in the base model by using the RENAMEPARM statement. This example illustrates two kinds of modifications that the RENAMEPARM statement can do:

- creating new parameters in the new model

  The error variances for E1–E4 in model 2 are different from those defined in model 1 because new parameters new1–new4 are now used. Parameter b2 is renamed as newb2 with a starting value at 0.2 in model 2. So the two models have distinct path coefficients for the F1-to-V2 path.

- changing free parameters into fixed constants

  By using the specification b4=.3 in the RENAMEPARM statement, b4 is no longer a free parameter in model 2. The path coefficient for the F1-to-V4 path in model 2 is now fixed at 0.3.

The RENAMEPARM statement is handy when you have just few parameters to change in the reference model defined by the REFMODEL statement. However, when there are a lot of parameters to modify, the RENAMEPARM statement might not be very efficient. For example, to make all parameters unique to the current model, you might consider using the ALLNEWPARMS, PARM_PREFIX, or PARM_SUFFIX option in the REFMODEL statement.

SAS Programming Statements

You can use SAS programming statements to define dependent parameters, parametric functions, and equality constraints among parameters.

Several statistical procedures support the use of SAS programming statements. The syntax of SAS programming statements are common to all these procedures and can be found in the section "Programming Statements" on page 410 in Chapter 18, "Shared Concepts and Topics."

SIMTEST Statement

SIMTEST *sim_test* < *sim_test . . .* > **;**

where *sim_test* represents one of the following:

- *test_name* = [*functions*]
- *test_name* = { *functions* }

and *functions* are either parameters in the model or parametric functions computed in the SAS programming statements.

When the estimates in a model are asymptotically multivariate-normal, continuous and differentiable functions of the estimates are also multivariate-normally distributed. In the SIMTEST statement, you can test these parametric functions simultaneously. The null hypothesis for the simultaneous tests is assumed to have the following form:

$$H_0 : h_1(\theta) = 0, h_2(\theta) = 0, \ldots$$

where θ is the set of model parameters (independent or dependent) in the analysis and each $h_i()$ is a continuous and differentiable function of the model parameters.

To test parametric functions simultaneously in the SIMTEST statement, you first assign a name for the simultaneously test in *test_name*. Then you put the parametric functions for the simultaneous test inside a pair of parentheses: either the '{' and '}' pair, or the '[' and ']' pair. For example, if θ_1, θ_2, θ_3, and θ_4 are parameters in the model and you want to test the equality of θ_1 and θ_2 and the equality of θ_3 and θ_4 simultaneously, you can use the following statements:

```
simtest
   Equality_test = [t1_t2_diff t3_t4_diff];
t1_t2_diff   = theta1 - theta2;
t3_t4_diff   = theta3 - theta4;
```

In the SIMTEST statement, you test two functions t1_t2_diff and t3_t4_diff simultaneously in the test named Equality_test. The two parametric functions t1_t2_diff and t3_t4_diff are computed in the SAS programming statements as differences of some parameters in the model.

See also the TESTFUNC statement on page 6817 for testing parametric functions individually.

STD Statement

STD *assignment* < , *assignment* . . . > **;**

where *assignment* represents:

variables = *parameter-spec*

The STD statement defines the variance parameters in the LINEQS models. STD, instead of VAR, is used as the statement name for defining variance parameters because it avoids conflicts with the VAR statement for specifying variables for analysis. You can specify no more than one STD statement with each LINEQS statement.

In each *assignment* of the STD statement, you specify the variance parameters of the *variables*. The *variables* are specified on the left-hand side of the equal sign of each *assignment*, while the parameter in the *parameter-spec* list on the right-hand side of the equal sign might be one of the following types:

- free or constrained parameters without starting values

  Names in the *parameter-spec* list denote parameters to estimate. They might be free or constrained parameters.

- free or constrained parameters with starting values

  Names followed by numbers inside a pair of parentheses in the *parameter-spec* list denote free or constrained parameters with starting values given by the parenthesized numbers.

- fixed parameter values

  Fixed constants not inside a pair of parentheses in the *parameter-spec* list denote fixed parameter values.

The *assignments* in the STD statement must be separated by commas. The *variables* listed on the left-hand side of the equal sign should contain only names of exogenous variables—that is, variance parameters in the LINEQS model are defined only for exogenous mainfest or latent variables, errors, and disturbances.

If the *parameter-spec* list is longer than the *variables* list, the *parameter-spec* list is shortened to the length of the *variables* list. If the *parameter-spec* is shorter than the *variables* list, the *parameter-spec* list is filled with repetitions of the last item in the list.

The *parameter-spec* list can contain prefix names. A prefix name is a short name called "root" followed by two underscores, '__'. A prefix name without a root (that is, as in '__') has its root replaced with parm. Whenever a prefix name is encountered, the TCALIS procedure generates a parameter name by appending a unique integer to the root of the prefix name. The root of the prefix name should have few characters so that the generated parameter name is not longer than 32 characters. To avoid unintentional equality constraints, the prefix names should not coincide with explicitly defined parameter names.

To illustrate various specification in the STD statement, consider the following statement:

```
std E1-E6 = vp1 GEN__ (1. 2.) GEN__ 4. vp5 ;
```

You specify the variance of E1 as a free parameter vp1 with a starting value at 1. The variance of E2 is a parameter named GEN1, with a starting value at 2. The variance of E3 is a parameter named GEN2, without a starting value provided. The variance of E4 is fixed at 4. The variance of E5 is parameter vp5, without a starting value given. Finally, the variance of E6 is denoted also by parameter vp5 because the *parameter-spec* list is shorter than the *variables* list so that the last member in the *parameter-spec* list is repeated.

Automatic Variance Parameters among Exogenous Variables

If the names of any exogenous variables (manifest, latent, or errors) in the LINEQS model are not included in the STD statement, the variances of those exogenous variables are assumed to be parameters to estimate. These parameters will be named automatically with the prefix _Add and appended with a unique integer in each name generation.

Modifying a Parameter Specification from a Reference Model

If you define a new LINEQS model by using a reference (or base) model in the REFMODEL statement, in some situations you might want to modify some parameter specifications from the STD statement of the reference model before transferring the specifications to the new model. If you want to change a particular STD parameter specification from the reference model, you can simply put the corresponding parameter location with a new parameter specification in the new model. If you want to delete a particular parameter location from the reference model, you can put that parameter location with the missing value '.' in the new model. For example, suppose you are defining a new LINEQS model by using the REFMODEL statement and the variance of variable V1 is defined as a fixed or free parameter in the reference model. If you do not want this parameter specification to be copied into your new model, you can use the following specification in the new model:

```
std V1 = .;
```

Notice that the missing value syntax is valid only when you use the REFMODEL statement. For a more detailed example of the LINEQS model respecification (see the section "Modifying a LINEQS Model from a Reference Model" on page 6769).

In the LINEQS model, all variances among exogenous variables are free parameters to estimate unless you specify otherwise in the STD statement (see the section "Automatic Variance Parameters among Exogenous Variables" on page 6816). This is also true for LINEQS models specified under the REFMODEL statement. In the reference model, variance parameters generated by PROC TCALIS are named with the _Add prefix and appended with integer suffixes. These parameter specifications in the reference model do **not** transfer to the new model. Only after the new model is resolved from the reference model, the REFMODEL statement options, the RENAMEPARM statement options, and the model respecifications are the unspecified variances of exogenous variables generated as free parameters named with the _Add prefix and integer suffixes. In this way, the variance parameters in the new model are not constrained to be the same as the corresponding parameters in the reference model. If you want any of these variance parameters constrained across the models, you must specify them explicitly in the STD statement of the reference model so that the variance specification is transferred to the new model.

STRUCTEQ Statement

STRUCTEQ *variables* < / *label* > **;**

where *label* represents:
LABEL | NAME = *name*

The STRUCTEQ statement functions exactly the same as the DETERM statement.

TESTFUNC Statement

TESTFUNC *functions* **;**

where *functions* are either parameters in the model or parametric functions computed in the SAS programming statements.

When the estimates in a model are asymptotically multivariate-normal, any continuous and differentiable function of the estimates is also normally distributed. In the TESTFUNC statement, you can test these parametric functions using z-tests. The form of the null hypothesis is as follows:

$H_0 : h(\theta) = 0$

where θ is the set of model parameters (independent or dependent) in the analysis and $h()$ is a continuous and differentiable function of the model parameters.

For example, if θ_1, θ_2, and θ_3 are parameters in the model, and you want to test whether θ_1 and θ_2 are the same and whether θ_3 is the same as the average of θ_1 and θ_2, you can use the following statements:

```
testfunc   t1_t2_diff t3_t1t2_diff;
t1_t2_diff   = theta1 - theta2;
t3_t1t2_diff = theta3 - (theta1 + theta2)/2;
```

In the TESTFUNC statement, you test two functions: t1_t2_diff and t3_t1t2_diff. These two functions are defined in the SAS programming statements that follow after the TESTFUNC statement. Thus, t1_t2_diff represents the difference between θ_1 and θ_2, and t3_t1t2_diff represents the difference between θ_3 and the average of θ_1 and θ_2.

See the SIMTEST statement if you want to test several null hypotheses simultaneously.

VAR Statement

VAR *variables* **;**

The VAR statement lists the numeric variables to be analyzed. It is one of the subsidiary group specification statements. You can use the VAR statement no more than once within the scope of each GROUP or the PROC TCALIS statement.

Variables in the VAR statement of the associated group are included in the corresponding model if the model is one of the following types: LINEQS, PATH, or RAM. Any VAR variables that are not mentioned in the model specification by using the main and subsidiary model specification statements are treated as exogenous variables in the model. This also means that for these models the use of the VAR statement in the associated groups is not necessary when all observed variables intended for analysis are already mentioned in the main and subsidiary model specification statements. Given the model specification, PROC TCALIS finds the set of observed variables for analysis automatically. You need the VAR statement only when you want to force the inclusion of those observed variables not mentioned in the model specification.

For the FACTOR model, the VAR variables of the associated groups are included in the model only for an exploratory factor analysis. The VAR statement is ignored for a confirmatory factor analysis, in which the observed variables can be defined only through the *factor-variable-relations* in the FACTOR statement.

The VAR statement of the associated group is ignored when the corresponding model is of either the LISMOD or MSTRUCT type. The observed variables for these two types of models must be defined in the variable lists of the LISMOD or MSTRUCT statement. For the LISMOD model, each observed variable for analysis is defined in either the YVAR= or XVAR= option of the LISMOD statement. For the MSTRUCT model, all observed variables for analysis are defined in the VAR= option of the MSTRUCT statement.

The VAR statement should not be confused with the PARAMETERS statement. In the PARAMETERS statement, you specify additional *parameters* in the model. Parameters are population quantities that characterize the functional relationships, variations, or covariation among variables. Unfortunately, parameters are sometimes referred to as *variables* in the optimization context. You have to make sure that all variables specified in the VAR statement correspond to the variables in the input data set, while the parameters specified in the PARAMETERS statement are population quantities that characterize distributions of the variables and their relationships.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement specifies the weight variable for the observations. It is one of the subsidiary group specification statements. You can use the WEIGHT statement no more than once within the scope of each GROUP statement or the PROC TCALIS statement.

Weighting is often done when the error variance associated with each observation is different and the values of the weight variable are proportional to the reciprocals of the variances. The WEIGHT and FREQ statements have a similar effect, except the WEIGHT statement does not alter the number of observations unless VARDEF=WGT or VARDEF=WDF. An observation is used in the analysis only if the WEIGHT variable is greater than 0 and is not missing.

Details: TCALIS Procedure

Input Data Sets

You can use four different kinds of input data sets in the TCALIS procedure, and you can use them simultaneously. The DATA= data set contains the data to be analyzed, and it can be an ordinary SAS data set containing raw data or a special TYPE=COV, TYPE=UCOV, TYPE=CORR, TYPE=UCORR, TYPE=SSCP, or TYPE=FACTOR data set containing previously computed statistics. The INEST= data set specifies an input data set that contains initial estimates for the parameters used in the optimization process, and it can also contain boundary and general linear constraints on the parameters. If the model does not change too much, you can use an OUTEST= data set from a previous PROC TCALIS analysis; the initial estimates are taken from the values of the _TYPE_=PARMS observation. The INMODEL= or INRAM= data set contains information of the analysis models (except for user-written programming statements). Often the INMODEL= data set is created as the OUTMODEL= data set from a previous PROC TCALIS analysis. See the section "OUTMODEL= SAS-data-set" on page 6827 for the structure of both OUTMODEL= and INMODEL= data sets. Using the INWGT= data set enables you to read in the weight matrix $\mathbf{W}$ that can be used in generalized least squares, weighted least squares, or diagonally weighted least squares estimation.

DATA= SAS-data-set

A TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set can be created by the CORR procedure or various other procedures. It contains means, standard deviations, the sample size, the covariance or correlation matrix, and possibly other statistics depending on which procedure is used.

If your data set has many observations and you plan to run PROC TCALIS several times, you can save computer time by first creating a TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set and using it as input to PROC TCALIS.

For example, assuming that PROC TCALIS is first run with an OUTMODEL=MODEL option, you can run the following statements in subsequent analyses with the same model in the first run:

```
* create TYPE=COV data set;
proc corr cov nocorr data=raw outp=cov(type=cov);
run;
* analysis using correlations;
proc tcalis corr data=cov inmodel=model;
run;
* analysis using covariances;
proc tcalis data=cov inmodel=model;
run;
```

Most procedures automatically set the TYPE= option of an output data set appropriately. However, the CORR procedure sets TYPE=CORR unless an explicit TYPE= option is used. Thus, (TYPE=COV) is needed in the preceding PROC CORR request, since the output data set is a covariance matrix. If you use a DATA step with a SET statement to modify this data set, you must declare the TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR attribute in the new data set.

You can use a VAR statement with PROC TCALIS when reading a TYPE=COV, TYPE=UCOV, TYPE=CORR, TYPE=UCORR, or TYPE=SSCP data set to select a subset of the variables or change the order of the variables.

CAUTION: Problems can arise from using the CORR procedure when there are missing data. By default, PROC CORR computes each covariance or correlation from all observations that have values present for the pair of variables involved ("pairwise deletion"). The resulting covariance or correlation matrix can have negative eigenvalues. A correlation or covariance matrix with negative eigenvalues is recognized as a singular matrix in PROC TCALIS, and you cannot compute (default) generalized least squares or maximum likelihood estimates. You can specify the RIDGE option to ridge the diagonal of such a matrix to obtain a positive definite data matrix. If the NOMISS option is used with the CORR procedure, observations with any missing values are completely omitted from the calculations ("listwise deletion"), and there is no possibility of negative eigenvalues (but there is still a chance for a singular matrix).

PROC TCALIS can also create a TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set that includes all the information needed for repeated analyses.

If the data set DATA=RAW does not contain missing values, the following statements should give the same PROC TCALIS results as the previous example:

```
* using correlations;
proc tcalis corr data=raw outstat=cov inmodel=model;
run;
* using covariances;
proc tcalis data=cov inmodel=model;
run;
```

You can create a TYPE=COV, TYPE=UCOV, TYPE=CORR, TYPE=UCORR, or TYPE=SSCP data set in a DATA step. Be sure to specify the TYPE= option in parentheses after the data set name in the DATA statement and include the _TYPE_ and _NAME_ variables. If you want to analyze the

covariance matrix but your DATA= data set is a TYPE=CORR or TYPE=UCORR data set, you should include an observation with _TYPE_=STD giving the standard deviation of each variable. By default, PROC TCALIS analyzes the recomputed covariance matrix even when a TYPE=CORR data set is provided, as shown in the following statements:

```
data correl(type=corr);
   input _type_ $ _name_ $ X1-X3;
   datalines;
std    .    4.   2.   8.
corr   X1   1.0   .    .
corr   X2    .7 1.0   .
corr   X3    .5   .4 1.0
;
proc tcalis inmodel=model;
run;
```

INEST= SAS-data-set

You can use the INEST= (or INVAR=) input data set to specify the initial values of the parameters used in the optimization and to specify boundary constraints and the more general linear constraints that can be imposed on these parameters.

The variables of the INEST= data set must correspond to the following:

- a character variable _TYPE_ that indicates the type of the observation
- *n* numeric variables with the parameter names used in the specified PROC TCALIS model
- the BY variables that are used in a DATA= input data set
- a numeric variable _RHS_ (right-hand side); needed only if linear constraints are used
- additional variables with names corresponding to constants used in the programming statements

The content of the _TYPE_ variable defines the meaning of the observation of the INEST= data set. PROC TCALIS recognizes observations with the following _TYPE_ specifications.

PARMS — specifies initial values for parameters that are defined in the model statements of PROC TCALIS. The _RHS_ variable is not used. Additional variables can contain the values of constants that are referred to in programming statements. At the beginning of each run of PROC TCALIS, the values of the constants are read from the PARMS observation for initializing the constants in the SAS programming statements.

UPPERBD | UB — specifies upper bounds with nonmissing values. The use of a missing value indicates that no upper bound is specified for the parameter. The _RHS_ variable is not used.

LOWERBD | LB — specifies lower bounds with nonmissing values. The use of a missing value indicates that no lower bound is specified for the parameter. The _RHS_ variable is not used.

LE | <= | < — specifies the linear constraint $\sum_j a_{ij}x_j \leq b_i$. The n parameter values contain the coefficients a_{ij}, and the _RHS_ variable contains the right-hand-side b_i. The use of a missing value indicates a zero coefficient a_{ij}.

GE | >= | > — specifies the linear constraint $\sum_j a_{ij}x_j \geq b_i$. The n parameter values contain the coefficients a_{ij}, and the _RHS_ variable contains the right-hand-side b_i. The use of a missing value indicates a zero coefficient a_{ij}.

EQ | = — specifies the linear constraint $\sum_j a_{ij}x_j = b_i$. The n parameter values contain the coefficients a_{ij}, and the _RHS_ variable contains the right-hand-side b_i. The use of a missing value indicates a zero coefficient a_{ij}.

The constraints specified in the INEST=, INVAR=, or ESTDATA= data set are added to the constraints specified in BOUNDS and LINCON statements.

You can use an OUTEST= data set from a PROC TCALIS run as an INEST= data set in a new run. However, be aware that the OUTEST= data set also contains the boundary and general linear constraints specified in the previous run of PROC TCALIS. When you are using this OUTEST= data set without changes as an INEST= data set, PROC TCALIS adds the constraints from the data set to the constraints specified by a BOUNDS and LINCON statement. Although PROC TCALIS automatically eliminates multiple identical constraints, you should avoid specifying the same constraint a second time.

INMODEL= SAS-data-set

This data set is usually created in a previous run of PROC TCALIS. It is useful if you want to reanalyze a problem in a different way such as using a different estimation method. You can alter an existing OUTMODEL= data set in the DATA step to create the INMODEL= data set that describes a modified model. See the section "OUTMODEL= SAS-data-set" on page 6827 for more details about the INMODEL= data set.

INWGT= SAS-data-set

This data set enables you to specify a weight matrix other than the default matrix for the generalized, weighted, and diagonally weighted least squares estimation methods. If you also specify the INWGTINV option (or use the INWGT(INV)= option), the INWGT= data set is assumed to contain the inverse of the weight matrix, rather than the weight matrix itself. The specification of any INWGT= data set for unweighted least squares or maximum likelihood estimation is ignored. For generalized and diagonally weighted least squares estimation, the INWGT= data set must contain a _TYPE_ and a _NAME_ variable as well as the manifest variables used in the analysis. The value of the _NAME_ variable indicates the row index i of the weight w_{ij}. For weighted least squares, the INWGT= data set must contain _TYPE_, _NAME_, _NAM2_, and _NAM3_ variables as well as the manifest variables used in the analysis. The values of the _NAME_, _NAM2_, and _NAM3_ variables indicate the three indices i, j, k of the weight $w_{ij,kl}$. You can store information other than the

weight matrix in the INWGT= data set, but only observations with _TYPE_=WEIGHT are used to specify the weight matrix **W**. This property enables you to store more than one weight matrix in the INWGT= data set. You can then run PROC TCALIS with each of the weight matrices by changing only the _TYPE_ observation in the INWGT= data set with an intermediate DATA step.

See the section "OUTWGT= SAS-data-set" on page 6836 for more details about the INWGT= data set.

Output Data Sets

OUTEST= SAS-data-set

The OUTEST= (or OUTVAR=) data set is of TYPE=EST and contains the final parameter estimates, the gradient, the Hessian, and boundary and linear constraints. For METHOD=ML, METHOD=GLS, and METHOD=WLS, the OUTEST= data set also contains the approximate standard errors, the information matrix (crossproduct Jacobian), and the approximate covariance matrix of the parameter estimates ((generalized) inverse of the information matrix). If there are linear or nonlinear equality or active inequality constraints at the solution, the OUTEST= data set also contains Lagrange multipliers, the projected Hessian matrix, and the Hessian matrix of the Lagrange function.

The OUTEST= data set can be used to save the results of an optimization by PROC TCALIS for another analysis with either PROC TCALIS or another SAS procedure. Saving results to an OUTEST= data set is advised for expensive applications that cannot be repeated without considerable effort.

The OUTEST= data set contains the BY variables, two character variables _TYPE_ and _NAME_, t numeric variables corresponding to the parameters used in the model, a numeric variable _RHS_ (right-hand side) that is used for the right-hand-side value b_i of a linear constraint or for the value $f = f(x)$ of the objective function at the final point x^* of the parameter space, and a numeric variable _ITER_ that is set to zero for initial values, set to the iteration number for the OUTITER output, and set to missing for the result output.

The _TYPE_ observations in Table 88.2 are available in the OUTEST= data set, depending on the request.

Table 88.2 _TYPE_ Observations in the OUTEST= Data Set

TYPE	Description
ACTBC	if there are active boundary constraints at the solution x^*, three observations indicate which of the parameters are actively constrained, as follows:

NAME	Description
GE	indicates the active lower bounds
LE	indicates the active upper bounds
EQ	indicates the active masks

COV	contains the approximate covariance matrix of the parameter estimates; used in computing the approximate standard errors.
COVRANK	contains the rank of the covariance matrix of the parameter estimates.
CRPJ_LF	contains the Hessian matrix of the Lagrange function (based on CRPJAC).
CRPJAC	contains the approximate Hessian matrix used in the optimization process. This is the inverse of the information matrix.
EQ	if linear constraints are used, this observation contains the ith linear constraint $\sum_j a_{ij}x_j = b_i$. The parameter variables contain the coefficients a_{ij}, $j = 1, \ldots, n$, the _RHS_ variable contains b_i, and _NAME_=ACTLC or _NAME_=LDACTLC.
GE	if linear constraints are used, this observation contains the ith linear constraint $\sum_j a_{ij}x_j \geq b_i$. The parameter variables contain the coefficients a_{ij}, $j = 1, \ldots, n$, and the _RHS_ variable contains b_i. If the constraint i is active at the solution x^*, then _NAME_=ACTLC or _NAME_=LDACTLC.
GRAD	contains the gradient of the estimates.
GRAD_LF	contains the gradient of the Lagrange function. The _RHS_ variable contains the value of the Lagrange function.
HESSIAN	contains the Hessian matrix.
HESS_LF	contains the Hessian matrix of the Lagrange function (based on HESSIAN).
INFORMAT	contains the information matrix of the parameter estimates (only for METHOD=ML, METHOD=GLS, or METHOD=WLS).
INITGRAD	contains the gradient of the starting estimates.
INITIAL	contains the starting values of the parameter estimates.

Table 88.2 *continued*

TYPE	Description
JACNLC	contains the Jacobian of the nonlinear constraints evaluated at the final estimates.
LAGM BC	contains Lagrange multipliers for masks and active boundary constraints.

NAME	Description
GE	indicates the active lower bounds
LE	indicates the active upper bounds
EQ	indicates the active masks

LAGM LC	contains Lagrange multipliers for linear equality and active inequality constraints in pairs of observations containing the constraint number and the value of the Lagrange multiplier.

NAME	Description
LEC_NUM	number of the linear equality constraint
LEC_VAL	corresponding Lagrange multiplier value
LIC_NUM	number of the linear inequality constraint
LIC_VAL	corresponding Lagrange multiplier value

LAGM NLC	contains Lagrange multipliers for nonlinear equality and active inequality constraints in pairs of observations that contain the constraint number and the value of the Lagrange multiplier.

NAME	Description
NLEC_NUM	number of the nonlinear equality constraint
NLEC_VAL	corresponding Lagrange multiplier value
NLIC_NUM	number of the linear inequality constraint
NLIC_VAL	corresponding Lagrange multiplier value

LE	if linear constraints are used, this observation contains the ith linear constraint $\sum_j a_{ij} x_j \leq b_i$. The parameter variables contain the coefficients a_{ij}, $j = 1, \ldots, n$, and the _RHS_ variable contains b_i. If the constraint i is active at the solution x^*, then _NAME_=ACTLC or _NAME_=LDACTLC.
LOWERBD | LB	if boundary constraints are used, this observation contains the lower bounds. Those parameters not subjected to lower bounds contain missing values. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.

Table 88.2 *continued*

TYPE	Description
NACTBC	all parameter variables contain the number n_{abc} of active boundary constraints at the solution x^*. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.
NACTLC	all parameter variables contain the number n_{alc} of active linear constraints at the solution x^* that are recognized as linearly independent. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.
NLC_EQ NLC_GE NLC_LE	contains values and residuals of nonlinear constraints. The _NAME_ variable is described as follows:

NAME	Description
NLC	inactive nonlinear constraint
NLCACT	linear independent active nonlinear constraint
NLCACTLD	linear dependent active nonlinear constraint

NLDACTBC	contains the number of active boundary constraints at the solution x^* that are recognized as linearly dependent. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.
NLDACTLC	contains the number of active linear constraints at the solution x^* that are recognized as linearly dependent. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.
NOBS	contains the number of observations.
PARMS	contains the final parameter estimates. The _RHS_ variable contains the value of the objective function.
PCRPJ_LF	contains the projected Hessian matrix of the Lagrange function (based on CRPJAC).
PHESS_LF	contains the projected Hessian matrix of the Lagrange function (based on HESSIAN).
PROJCRPJ	contains the projected Hessian matrix (based on CRPJAC).
PROJGRAD	if linear constraints are used in the estimation, this observation contains the $n - n_{act}$ values of the projected gradient $g_Z = Z'g$ in the variables corresponding to the first $n - n_{act}$ parameters. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.
PROJHESS	contains the projected Hessian matrix (based on HESSIAN).
STDERR	contains approximate standard errors (only for METHOD=ML, METHOD=GLS, or METHOD=WLS).

Table 88.2 *continued*

TYPE	Description
TERMINAT	the _NAME_ variable contains the name of the termination criterion.
UPPERBD | UB	if boundary constraints are used, this observation contains the upper bounds. Those parameters not subjected to upper bounds contain missing values. The _RHS_ variable contains a missing value, and the _NAME_ variable is blank.

If the technique specified by the OMETHOD= option cannot be performed (for example, no feasible initial values can be computed or the function value or derivatives cannot be evaluated at the starting point), the OUTEST= data set can contain only some of the observations (usually only the PARMS and GRAD observations).

OUTMODEL= SAS-data-set

The OUTMODEL= (or OUTRAM=) data set is of TYPE=CALISMDL and contains the model specification and the computed parameter estimates and standard error estimates. This data set is intended to be reused as an INMODEL= data set to specify good initial values in a subsequent analysis by PROC TCALIS.

The OUTMODEL= data set contains the following variables:

- the BY variables, if any
- an _MDLNUM_ variable for model numbers, if used
- a character variable _TYPE_, which takes various values indicating the type of model specification
- a character variable _NAME_, which indicates the model type, parameter name, or variable name
- a character variable _VAR1_, which is the name or number of the first variable in the specification
- a character variable _VAR2_, which is the name or number of the second variable in the specification
- a numerical variable _ESTIM_ for the final estimate of the parameter location
- a numerical variable _STDERR_ for the standard error estimate of the parameter location

Each observation (record) of the OUTMODEL= data set contains a piece of information regarding the model specification. Depending on the type of the specification indicated by the value of the _TYPE_ variable, the meanings of _NAME_, _VAR1_, and _VAR2_ differ. The following tables summarize the meanings of the _NAME_, _VAR1_, and _VAR2_ variables for each value of the _TYPE_ variable, given the type of the model.

FACTOR Models

Value of _TYPE_	Specification Type	_NAME_	_VAR1_	_VAR2_
MDLTYPE	model type	model type		
ADDPVAR	added (partial) variance	parameter	variable	
COV	covariance	parameter	first variable	second variable
LOADING	factor loading	parameter	manifest variable	factor variable
MEAN	mean or intercept	parameter	variable	
PVAR	(partial) variance	parameter	variable	

For factor models, the value of the _NAME_ variable is either EFACTOR (exploratory factor model) or CFACTOR (confirmatory factor model) for the _TYPE_=MDLTYPE observation. Other observations specify the parameters in the model. The _NAME_ values for these observations are the parameter names. Any observation with a _TYPE_ value of COV or LOADING is for specifying a parameter associated with two variables. The _VAR1_ and _VAR2_ values of such an observation indicate the variables involved. Any observation with a _TYPE_ value of ADDPVAR, MEAN, or PVAR is for specifying a parameter associated with a single variable. The _VAR1_ value of such an observation indicates the variable involved, while the _VAR2_ value is not used.

LINEQS Models

Value of _TYPE_	Specification Type	_NAME_	_VAR1_	_VAR2_
MDLTYPE	model type	model type		
ADDCOV	added covariance	parameter	first variable	second variable
ADDMEAN	added mean	parameter	variable	
ADDSTD	added variance	parameter	variable	
COV	covariance	parameter	first variable	second variable
EQUATION	path coefficient	parameter	outcome variable	predictor variable
MEAN	mean	parameter	variable	
STD	variance	parameter	variable	

The value of the _NAME_ variable is LINEQS for the _TYPE_=MDLTYPE observation. Other observations specify the parameters in the model. The _NAME_ values for these observations are the parameter names. Any observation with a _TYPE_ value of ADDCOV, COV, or EQUATION is for specifying a parameter associated with two variables. The _VAR1_ and _VAR2_ values of such an observation indicate the variables involved. Any observation with a _TYPE_ value of ADDMEAN, ADDSTD, MEAN, or STD is for specifying a parameter associated with a single variable. The _VAR1_ value of such an observation indicates the variables involved, while the _VAR2_ value is not used.

LISMOD Models

Value of _TYPE_	Specification Type	_NAME_	_VAR1_	_VAR2_
MDLTYPE	model type	model type		
XVAR	x-variable	variable	variable number	
YVAR	y-variable	variable	variable number	
ETAVAR	eta-variable	variable	variable number	
XIVAR	xi-variable	variable	variable number	
ADDKAPPA	added _KAPPA_ entry	parameter	row number	
ADDPHI	added _PHI_ entry	parameter	row number	column number
ADDPSI	added _PSI_ entry	parameter	row number	column number
ADDTHETAX	added _THETAX_ entry	parameter	row number	column number
ADDTHETAY	added _THETAY_ entry	parameter	row number	column number
ALPHA	_ALPHA_ entry	parameter	row number	
BETA	_BETA_ entry	parameter	row number	column number
GAMMA	_BETA_ entry	parameter	row number	column number
KAPPA	_KAPPA_ entry	parameter	row number	
LAMBDAX	_LAMBDAX_ entry	parameter	row number	column number
LAMBDAY	_LAMBDAY_ entry	parameter	row number	column number
NUX	_NUX_ entry	parameter	row number	
NUY	_NUY_ entry	parameter	row number	
PHI	_PHI_ entry	parameter	row number	column number
PSI	_PSI_ entry	parameter	row number	column number
THETAX	_THETAX_ entry	parameter	row number	column number
THETAY	_THETAY_ entry	parameter	row number	column number

The value of the _NAME_ variable is LISMOD for the _TYPE_=MDLTYPE observation. Other observations specify either the variables or the parameters in the model.

Observations with _TYPE_ values equal to XVAR, YVAR, ETAVAR, and XIVAR indicate the variables on the respective lists in the model. The _NAME_ variable of these observations stores the names of the variables, while the _VAR1_ variable stores the variable numbers on the respective list. The variable numbers in this data set are not arbitrary—that is, they define the variable orders in the rows and columns of the LISMOD model matrices. The _VAR2_ variable of these observations is not used.

All other observations in this data set specify the parameters in the model. The _NAME_ values of these observations are the parameter names. The corresponding _VAR1_ and _VAR2_ values of these observations indicate the row and column locations of the parameters in the LISMOD model matrices specified in the _TYPE_ variable. For example, when the value of _TYPE_ is ADDPHI or PHI, the parameter specified is located in the _PHI_ matrix, with its row and column numbers indicated by the _VAR1_ and _VAR2_ values, respectively. Some observations for specifying parameters do not have values in the _VAR2_ variable. This means that the associated LISMOD matrices are vectors so that the column numbers are always 1 for these observations.

MSTRUCT Models

Value of _TYPE_	Specification Type	_NAME_	_VAR1_	_VAR2_
MDLTYPE	model type	model type		
VAR	variable	variable	variable number	
COVMAT	covariance	parameter	row number	column number
MEANVEC	mean	parameter	row number	

The value of the _NAME_ variable is MSTRUCT for the _TYPE_=MDLTYPE observation. Other observations specify either the variables or the parameters in the model. Observations with _TYPE_ values equal to VAR indicate the variables in the model. The _NAME_ variable of these observations stores the names of the variables, while the _VAR1_ variable stores the variable numbers on the variable list. The variable numbers in this data set are not arbitrary—that is, they define the variable orders in the rows and columns of the mean and covariance matrices. The _VAR2_ variable of these observations is not used.

All other observations in this data set specify the parameters in the model. The _NAME_ values of these observations are the parameter names. The corresponding _VAR1_ and _VAR2_ values of these observations indicate the row and column locations of the parameters in the mean or covariance matrix, as specified in the _TYPE_ model. For example, when the value of _TYPE_ is COVMAT, the parameter specified is located in the covariance matrix, with its row and column numbers indicated by the _VAR1_ and _VAR2_ values, respectively. For observations with _TYPE_=MEANVEC, the _VAR2_ variable is not used because the column numbers are always 1 for parameters in the mean vector.

PATH Models

Value of _TYPE_	Specification Type	_NAME_	_VAR1_	_VAR2_
MDLTYPE	model type	model type		
ADDPCOV	added (partial) covariance	parameter	first variable	second variable
ADDMEAN	added mean	parameter	variable	
ADDPVAR	added (partial) variance	parameter	variable	
LEFT	path coefficient	parameter	outcome variable	predictor variable
MEAN	mean	parameter	variable	
PCOV	(partial) covariance	parameter	first variable	second variable
PVAR	(partial) variance	parameter	variable	
RIGHT	path coefficient	parameter	predictor variable	outcome variable

The value of the _NAME_ variable is PATH for the _TYPE_=MDLTYPE observation. Other observations specify the parameters in the model. The _NAME_ values for these observations are the parameter names. Any observation with a _TYPE_ value of ADDPCOV, LEFT, PCOV, or RIGHT is for specifying a parameter associated with two variables. The _VAR1_ and _VAR2_ values of such an observation indicate the variables involved. Any observation with a _TYPE_ value of ADDMEAN, ADDPVAR, ADDSTD, MEAN, or PVAR is for specifying a parameter associated with a single variable. The _VAR1_ value of such an observation indicates the variable involved, while the _VAR2_ value is not used.

RAM Models

Value of _TYPE_	Specification Type	_NAME_	_VAR1_	_VAR2_
MDLTYPE	model type	model type		
_A_	matrix _A_ entry	parameter	first variable	second variable
_P_	matrix _P_ entry	parameter	first variable	second variable
_W_	vector _W_ entry	parameter	variable	
ADDMEAN	added mean	parameter	variable	
ADDPCOV	added (partial) covariance	parameter	first variable	second variable
ADDPVAR	added (partial) variance	parameter	variable	
INTERCEP	intercept	parameter	variable	
LEFT	path coefficient	parameter	outcome variable	predictor variable
MEAN	mean	parameter	variable	
PCOV	(partial) covariance	parameter	first variable	second variable
PVAR	(partial) variance	parameter	variable	
RIGHT	path coefficient	parameter	predictor variable	outcome variable

The value of the _NAME_ variable is RAM for the _TYPE_=MDLTYPE observation. Other observations specify the parameters in the model. The _NAME_ values for these observations are for the parameter names. Any observation with a _TYPE_ value of _A_, _P_, ADDPCOV, LEFT, PCOV, or RIGHT is for specifying a parameter associated with two variables. The _VAR1_ and _VAR2_ values of such an observation indicate the variables involved. Any observation with a _TYPE_ value of _W_, ADDMEAN, ADDPVAR, INTERCEP, MEAN, or PVAR is for specifying a parameter associated with a single variable. The _VAR1_ value of such an observation indicates the variable involved, while the _VAR2_ value is not used.

Reading an OUTMODEL= Data Set As an INMODEL= Data Set in Subsequent Analyses

When the OUTMODEL= data set is treated as an INMODEL= data set in subsequent analyses, you need to pay attention to those observations with _TYPE_ values prefixed by "ADD" (for example, ADDCOV, ADDMEAN, or ADDSTD). These observations represent parameter locations added automatically by PROC TCALIS in a previous run. Because the context of the new analyses might be different, these observations for added parameter locations might not be suitable in the new runs any more. Hence, these observations are *not* read as input model information. Fortunately, after reading the INMODEL= specification in the new analyses, TCALIS analyzes the new model specification again. It then adds an appropriate set of parameters in the new context when necessary.

OUTSTAT= SAS-data-set

The OUTSTAT= data set is similar to the TYPE=COV, TYPE=UCOV, TYPE=CORR, or TYPE=UCORR data set produced by the CORR procedure. The OUTSTAT= data set contains the following variables:

- the BY variables, if any
- the _GPNUM_ variable for groups numbers, if used in the analysis
- two character variables, _TYPE_ and _NAME_
- the manifest and the latent variables analyzed

The OUTSTAT= data set contains the following information (when available) in the observations:

- the mean and standard deviation
- the skewness and kurtosis (if the DATA= data set is a raw data set and the KURTOSIS option is specified)
- the number of observations
- if the WEIGHT statement is used, sum of the weights
- the correlation or covariance matrix to be analyzed
- the predicted correlation or covariance matrix
- the standardized or normalized residual correlation or covariance matrix
- if the model contains latent variables, the predicted covariances between latent and manifest variables and the latent variable (or factor) score regression coefficients (see the PLATCOV option on page 6741)

In addition, for FACTOR models the OUTSTAT= data set contains:

- the unrotated factor loadings, the error variances, and the matrix of factor correlations
- the standardized factor loadings and factor correlations
- the rotation matrix, rotated factor loadings, and factor correlations
- standardized rotated factor loadings and factor correlations

If effects are analyzed, the OUTSTAT= data set also contains:

- direct, indirect, and total effects and their standard error estimates
- standardized direct, indirect, and total effects and their standard error estimates

Each observation in the OUTSTAT= data set contains some type of statistic as indicated by the _TYPE_ variable. The values of the _TYPE_ variable are shown in the following tables:

Basic Descriptive Statistics

Value of _TYPE_	Contents
CORR	correlations analyzed
COV	covariances analyzed
KURTOSIS	univariate kurtosis
MEAN	means
N	sample size
SKEWNESS	univariate skewness
STD	standard deviations
SUMWGT	sum of weights (if the WEIGHT statement is used)

For the _TYPE_=CORR or COV observations, the _NAME_ variable contains the name of the manifest variable that corresponds to each row for the covariance or correlation. For other observations, _NAME_ is blank.

Predicted Moments and Residuals

value of _TYPE_	Contents
METHOD=NONE	
ASRES	asymptotically standardized residuals
NRES	normalized residuals
PRED	predicted moments
RES	raw residuals
SRES	variance standardized residuals
METHOD=DWLS	
DWLSASRS	DWLS asymptotically standardized residuals
DWLSNRES	DWLS normalized residuals
DWLSPRED	DWLS predicted moments
DWLSRES	DWLS raw residuals
DWLSSRES	DWLS variance standardized residuals
METHOD=GLS	
GLSASRES	GLS asymptotically standardized residuals
GLSNRES	GLS normalized residuals
GLSPRED	GLS predicted moments
GLSRES	GLS raw residuals
GLSSRES	GLS variance standardized residuals
METHOD=ML	
MAXASRES	ML asymptotically standardized residuals
MAXNRES	ML normalized residuals
MAXPRED	ML predicted moments
MAXRES	ML raw residuals
MAXSRES	ML variance standardized residuals
METHOD=ULS	
ULSASRES	ULS asymptotically standardized residuals
ULSNRES	ULS normalized residuals
ULSPRED	ULS predicted moments
ULSRES	ULS raw residuals
ULSSRES	ULS variance standardized residuals
METHOD=WLS	
WLSASRES	WLS asymptotically standardized residuals
WLSNRES	WLS normalized residuals
WLSPRED	WLS predicted moments
WLSRES	WLS raw residuals
WLSSRES	WLS variance standardized residuals

For residuals or predicted moments of means, the _NAME_ variable is a fixed value denoted by _Mean_. For residuals or predicted moments for covariances or correlations, the _NAME_ variable is used for names of variables.

Effects and Latent Variable Scores Regression Coefficients

Value of _TYPE_	Contents
Unstandardized Effects	
DEFFECT	direct effects
DEFF_SE	standard error estimates for direct effects
IEFFECT	indirect effects
IEFF_SE	standard error estimates for indirect effects
TEFFECT	total effects
TEFF_SE	standard error estimates for total effects
Standardized Effects	
SDEFF	standardized direct effects
SDEFF_SE	standard error estimates for standardized direct effects
SIEFF	standardized indirect effects
SIEFF_SE	standard error estimates for standardized indirect effects
STEFF	standardized total effects
STEFF_SE	standard error estimates for standardized total effects
Latent Variable Scores Coefficients	
LSSCORE	latent variable (or factor) scores regression coefficients for ULS method
SCORE	latent variable (or factor) scores regression coefficients other than ULS method

For latent variable or factor scores coefficients, the _NAME_ variable contains factor or latent variables in the observations. For other observations, the _NAME_ variable contains manifest or latent variable names.

You can use the latent variable score regression coefficients with PROC SCORE to compute factor scores. If the analyzed matrix is a covariance rather than a correlation matrix, the _TYPE_=STD observation is not included in the OUTSTAT= data set. In this case, the standard deviations can be obtained from the diagonal elements of the covariance matrix. Dropping the _TYPE_=STD observation prevents PROC SCORE from standardizing the observations before computing the factor scores.

Factor Analysis Results

Value of _TYPE_	Contents
ERRVAR	error variances
FCORR	factor correlations or covariances
LOADINGS	unrotated factor loadings
RFCORR	rotated factor correlations or covariances
RLOADING	rotated factor loadings
ROTMAT	rotation matrix
STDFCORR	standardized factor correlations
STDLOAD	standardized factor loadings
STDRFCOR	standardized rotated factor correlations or covariances
STDRLOAD	standardized rotated factor loadings

For the _TYPE_=ERRVAR observation, the _NAME_ variable is blank. For all other observations, the _NAME_ variable contains factor names.

OUTWGT= SAS-data-set

You can create an OUTWGT= data set that is of TYPE=WEIGHT and contains the weight matrix used in generalized, weighted, or diagonally weighted least squares estimation. The OUTWGT= data set contains the weight matrix on which the WRIDGE= and the WPENALTY= options are applied. However, if you input the inverse of the weight matrix with the INWGT= and INWGTINV options (or the INWGT(INV)= option alone) in the same analysis, the OUTWGT= data set contains the same elements of the inverse of the weight matrix. For unweighted least squares or maximum likelihood estimation, no OUTWGT= data set can be written. The weight matrix used in maximum likelihood estimation is dynamically updated during optimization. When the ML solution converges, the final weight matrix is the same as the predicted covariance or correlation matrix, which is included in the OUTSTAT= data set (observations with _TYPE_ =MAXPRED).

For generalized and diagonally weighted least squares estimation, the weight matrices $\mathbf{W}$ of the OUTWGT= data set contain all elements w_{ij}, where the indices i and j correspond to all manifest variables used in the analysis. Let $varnam_i$ be the name of the ith variable in the analysis. In this case, the OUTWGT= data set contains n observations with the variables shown in the following table:

Variable	Contents
TYPE	WEIGHT (character)
NAME	name of variable $varnam_i$ (character)
$varnam_1$	weight w_{i1} for variable $varnam_1$ (numeric)
$\vdots$	$\vdots$
$varnam_n$	weight w_{in} for variable $varnam_n$ (numeric)

For weighted least squares estimation, the weight matrix $\mathbf{W}$ of the OUTWGT= data set contains only the nonredundant elements $w_{ij,kl}$. In this case, the OUTWGT= data set contains $n(n+1)(2n+1)/6$ observations with the variables shown in the following table:

Variable	Contents
TYPE	WEIGHT (character)
NAME	name of variable $varnam_i$ (character)
NAM2	name of variable $varnam_j$ (character)
NAM3	name of variable $varnam_k$ (character)
$varnam_1$	weight $w_{ij,k1}$ for variable $varnam_1$ (numeric)
$\vdots$	$\vdots$
$varnam_n$	weight $w_{ij,kn}$ for variable $varnam_n$ (numeric)

Symmetric redundant elements are set to missing values.

OUTFIT= SAS-data-set

You can create an OUTFIT= data set that is of TYPE=CALISFIT and contains the values of the fit indices of your analysis. If you use two estimation methods such as LSML or LSWLS, the fit indices are for the second analysis. An OUTFIT=data set contains the following variables:

- a character variable _TYPE_ for the types of fit indices
- a character variable _INDEX_ for the names of the fit indices
- a numerical variable _VALUE_ for the numerical values of the fit indices
- a character variable _PRINT_ for the character-formatted fit index values.

The possible values of _TYPE_ are:

ModelInfo:	basic modeling statistics and information
Absolute:	standalone fit indices
Parsimony:	fit indices that take model parsimony into account
Incremental:	fit indices that are based on comparison with a baseline model

Possible Values of _INDEX_ When _TYPE_=ModelInfo

Value of _INDEX_	Description
N Observations	number of observations use in the analysis
N Variables	number of variables
N Moments	number of mean or covariance elements
N Parameters	number of parameters
N Active Constraints	number of active constraints in the solution
Independence Model Chi-Square	chi-square value for the independence model
Independence Model Chi-Square DF	the degrees of freedom for the independence model chi-square

Possible Values of _INDEX_ When _TYPE_=Absolute

Value of _INDEX_	Description
Fit Function	fit function value
Percent Contribution to Chi-Square	percentage contribution to the chi-square value
Chi-Square	model chi-square value
Chi-Square DF	the degrees of freedom for model chi-square test
Pr > Chi-Square	the probability of obtaining a larger chi-square than the observed value
Elliptic Corrected Chi-Square	the elliptic-corrected chi-square value
Pr > Elliptic Corr. Chi-Square	the probability of obtaining a larger elliptic-corrected chi-square value
Z-test of Wilson and Hilferty	Z-test of Wilson and Hilferty
Hoelter Critical N	the N value that will make a significant chi-square when multiplied to the fit function value
Root Mean Square Residual (RMSR)	root mean square residual
Standardized RMSR (SRMSR)	standardized root mean square residual
Goodness of Fit Index (GFI)	Jöreskog and Sörbom goodness-of-fit index

Possible Values of _INDEX_ When _TYPE_=Parsimony

Value of _INDEX_	Description
Adjusted GFI (AGFI)	goodness-of-fit index adjusted for the degrees of freedom of the model
Parsimonious GFI	Mulaik et al. modification of the GFI
RMSEA Estimate	Steiger and Lind (1980) root mean square error approximation
RMSEA Lower **r**% Confidence Limit	the lower **r**%[1] confidence limit for RMSEA
RMSEA Upper **r**% Confidence Limit	the upper **r**%[1] confidence limit for RMSEA
Probability of Close Fit	Browne and Cudeck (1993) test of close fit
ECVI Estimate	expected cross-validation index
ECVI Lower **r**% Confidence Limit	the lower **r**%[2] confidence limit for ECVI
ECVI Upper **r**% Confidence Limit	the upper **r**%[2] confidence limit for ECVI
Akaike Information Criterion	Akaike information criterion
Bozdogan CAIC	Bozdogan (1987) consistent AIC
Schwarz Bayesian Criterion	Schwarz (1978) Bayesian criterion
McDonald Centrality	McDonald and Hartmann (1992) measure of centrality

1. The value of **r** is one minus the ALPHARMS= value. By default, **r**=90.
2. The value of **r** is one minus the ALPHAECV= value. By default, **r**=90.

Possible Values of _INDEX_ When _TYPE_=Incremental

Value of _INDEX_	Description
Bentler Comparative Fit Index	Bentler (1995) comparative fit index
Bentler-Bonett NFI	Bentler and Bonett (1980) normed fit index
Bentler-Bonett Non-normed Index	Bentler and Bonett (1980) non-normed fit index
Bollen Normed Index Rho1	Bollen normed ρ_1
Bollen Non-normed Index Delta2	Bollen non-normed δ_2
James et al. Parsimonious NFI	James, Mulaik, and Brett (1982) parsimonious normed fit index

The LINEQS Model

The LINEQS modeling language is adapted from the EQS program by Bentler (1995). The statistical models that LINEQS or EQS analyzes are essentially the same as other general modeling languages such as LISMOD, RAM, and PATH in PROC TCALIS. However, the terminology and approach of the LINEQS or EQS modeling language are different from other languages. They are based on the theoretical model developed by Bentler and Weeks (1980). For convenience, models analyzed using the LINEQS modeling language will be called LINEQS models. It is noted that these so-called LINEQS models can also be analyzed by other general modeling languages in PROC TCALIS.

In the LINEQS (or the original EQS) model, relationships among variables are represented by a system of equations. For example:

$$Y_1 = a_0 + a_1 X_1 + a_2 X_2 + E_1$$

$$Y_2 = b_0 + b_1 X_1 + b_2 Y_1 + E_2$$

On the left-hand side of each equation, an outcome variable is hypothesized to be a linear function of one or more predictor variables and an error, which are all specified on the right-hand side of the equation. The parameters specified in an equation are the effects (or regression coefficients) of the predictor variables. For example, in the preceding equations, Y_1 and Y_2 are outcome variables; E_1 and E_2 are error variables; a_1, a_2, b_1, and b_2 are effect parameters (or regression coefficients); and a_0 and b_0 are intercept parameters. Variables X_1 and X_2 serve as predictors in the first equation, while variables X_1 and Y_1 serve as predictors in the second equation.

This is almost the same representation as in multiple regression models. However, the LINEQS model entails more. It supports a system of equations that can also include latent variables, measurement errors, and correlated errors.

Types of Variables in the LINEQS Model

The distinction between dependent and independent variables is important in the LINEQS model.

A variable is dependent if it appears on the left-hand side of an equation in the model. A dependent variable might be observed (manifest) or latent. It might or might not appear on the right-hand

side of other equations, but it cannot appear on the left-hand sides of two or more equations. Error variables cannot be dependent in the LINEQS model.

A variable in the LINEQS model is independent if it is not dependent. Independent variables can be observed (manifest) or latent. All error variables must be independent in the LINEQS model.

Dependent variables are also referred to as endogenous variables. These names are interchangeable. Similarly, independent variables are interchangeable with exogenous variables.

Whereas an outcome variable in any equation must be a dependent variable, a predictor variable in an equation is not necessarily an independent variable in the entire LINEQS model. For example, Y_1 is a predictor variable in the second equation of the preceding example, but it is a dependent variable in the LINEQS model. In sum, the predictor-outcome nature of a variable is determined within a single equation, while the exogenous-endogenous (independent-dependent) nature of variable is determined within the entire system of equations.

In addition to the dependent-independent variable distinction, variables in the LINEQS model are distinguished according to whether they are observed in the data. Variables that are observed in research are called observed or manifest variables. Hypothetical variables that are not observed in the LINEQS model are latent variables.

Two types of latent variables should be distinguished: one is error variables; the other is non-error variables. An error variable is unique to an equation. It serves as the unsystematic source of effect for the outcome variable in an equation. If the outcome variable in the equation is latent, the corresponding error variable is also called disturbance. In contrast, non-error or systematic latent variables are called factors. Factors are unmeasured hypothetical constructs in your model. They are systematic sources that explain or describe functional relationships in your model.

Both manifest variables and latent factors can be dependent or independent. However, error or disturbance terms must be independent (or exogenous) variables in your model.

Naming Variables in the LINEQS Model

Whether a variable in each equation is an outcome or a predictor variable is prescribed by the modeler. Whether a variable is independent or dependent can be determined by analyzing the entire system of equations in the model. Whether a variable is observed or latent can be determined if it is referenced in your data set. However, whether a latent variable serves as a factor or an error can be determined only if you provide the specific information.

To distinguish latent factors from errors and both from manifest variables, the following rules for naming variables in the LINEQS model are followed:

- Manifest variables are referenced in the input data set. You use their names in the LINEQS model specification directly. There is no additional naming rule for the manifest variables in the LINEQS model beyond those required by the SAS system.
- Latent factor variables must start with letter F or f (for factor).
- Error variables must start with letter E or e (for error), or D or d (for disturbance). Although you might enforce the use of D- (or d-) variables for disturbances, it is not required. For flexibility, disturbance variables can also start with letter E or e in the LINEQS model.

- The names of latent variables, errors, and disturbances (F-, E-, and D- variables) should not coincide with the names of manifest variables.
- You should not use Intercept as a name for any variable. This name is reserved for intercept specification in LINEQS model equations.

Matrix Representation of the LINEQS Model

As a programming language, the LINEQS model uses equations to describes relationships among variables. But as a mathematical model, the LINEQS model is more conveniently described by matrix terms. In this section, the LINEQS matrix model is described.

Suppose in a LINEQS model that there are n_i independent variables and n_d dependent variables. The vector of the independent variables is denoted by ξ, in the order of manifest variables, latent factors, and error variables. The vector of dependent variables is denoted by η, in the order of manifest variables and latent factors. The LINEQS model matrices are defined as follows:

α $(n_d \times 1)$:	intercepts of dependent variables
β $(n_d \times n_d)$:	effects of dependent variables (in columns) on dependent variables (in rows)
γ $(n_d \times n_i)$:	effects of independent variables (in columns) on dependent variables (in rows)
Φ $(n_i \times n_i)$:	covariance matrix of independent variables
ν $(n_i \times 1)$:	means of independent variables

The model equation of the LINEQS model is:

$$\eta = \alpha + \beta\eta + \gamma\xi$$

Assuming that $(I - \beta)$ is invertible, under the model the covariance matrix of all variables $(\eta', \xi')'$ is structured as:

$$\Sigma_a = \begin{pmatrix} (I-\beta)^{-1}\gamma\Phi\gamma'(I-\beta)^{-1\prime} & (I-\beta)^{-1}\gamma\Phi \\ \Phi\gamma'(I-\beta)^{-1\prime} & \Phi \end{pmatrix}$$

The mean vector of all variables $(\eta', \xi')'$ is structured as:

$$\mu_a = \begin{pmatrix} (I-\beta)^{-1}(\alpha + \gamma\nu) \\ \nu \end{pmatrix}$$

As is shown in the structured covariance and mean matrices, the means and covariances of independent variables are direct model parameters in ν and Φ; whereas the means and covariances of dependent variables are functions of various model matrices and hence functions of model parameters.

The covariance and mean structures of all observed variables are obtained by selecting the elements in Σ_a and μ_a. Mathematically, define a selection matrix G of dimensions $n \times (n_d + n_i)$, where n is the number of observed variables in the model. The selection matrix G contains zeros and ones as its elements. Each row of G has exactly one nonzero element at the position that corresponds to the

location of an observed row variable in Σ_a or μ_a. With each row of G selecting a distinct observed variable, the structured covariance matrix of all observed variables is represented by the following:

$$\Sigma = G\Sigma_a G'$$

The structured mean vector of all observed variables is represented by:

$$\mu = G\mu_a$$

Partitions of Some LINEQS Model Matrices and Their Restrictions

There are some restrictions in some of the LINEQS model matrices. Although these restrictions do not affect the derivation of the covariance and mean structures, they are enforced in the LINEQS model specification.

Model Restrictions on the β Matrix

The diagonal of the β matrix must be zeros. This prevents the direct regression of dependent variables on themselves. Hence, in the LINEQS statement you cannot specify the same variable on both the left-hand and the right-hand sides of the same equation.

Partitions of the γ Matrix and the Associated Model Restrictions

The columns of the γ matrix refer to the variables in ξ, in the order of mainfest variables, latent factors, and error variables. In the LINEQS model, the following partition of the γ matrix is assumed:

$$\gamma = \begin{pmatrix} \gamma_0 & E \end{pmatrix}$$

where γ_0 is an $n_d \times (n_i - n_d)$ matrix for the effects of independent manifest variables and latent factors on the dependent variables and E is an $n_d \times n_d$ permutation matrix for the effects of errors on the dependent variables.

The dimension of submatrix E is $n_d \times n_d$ because in the LINEQS model each dependent variable signifies an equation with an error term. In addition, because E is a permutation matrix (which is formed by exchanging rows of an identity matrix of the same order), the partition of the γ matrix ensures that each dependent variable is associated with a **unique** error term and that the effect of each error term on its associated dependent variable is 1.

As a result of the error term restriction, in the LINEQS statement you must specify a unique error term in each equation. The coefficient associated with the error term can only be a fixed value at one, either explicitly (with 1.0 inserted immediately before the error term) or implicitly (with no coefficient specified).

Partitions of the ν Vector and the Associated Model Restrictions

The ν vector contains the means of independent variables, in the order of the manifest, latent factor, and error variables. In the LINEQS model, the following partition of the ν vector is assumed:

$$\nu = \begin{pmatrix} \nu_0 \\ 0 \end{pmatrix}$$

where ν_0 is an $(n_i - n_d) \times 1$ vector for the means of independent manifest variables and latent factors and 0 is a null vector of dimension n_d for the means of errors or disturbances. Again, the dimension of the null vector is n_d because each dependent variable is associated uniquely with an error term. This partition restricts the means of errors or disturbances to zeros.

Hence, when specifying a LINEQS model, you cannot specify the means of errors (or disturbances) as free parameter or fixed values other than zero in the MEAN statement.

Partitions of the Φ matrix

The Φ matrix is for the covariances of the independent variables, in the order of the manifest, latent factor, and error variables. The following partition of the Φ matrix is assumed:

$$\Phi = \begin{pmatrix} \Phi_{11} & \Phi_{21}' \\ \Phi_{21} & \Phi_{22} \end{pmatrix}$$

where Φ_{11} is an $(n_i - n_d) \times (n_i - n_d)$ covariance matrix for the independent manifest variables and latent factors, Φ_{22} is an $n_d \times n_d$ covariance matrix for the errors, and Φ_{21} is an $n_d \times (n_i - n_d)$ covariance matrix for the errors with other independent variables in the LINEQS model. Because Φ is symmetric, Φ_{11} and Φ_{22} are also symmetric.

There are actually no model restrictions placed on the submatrices of the partition. However, in most statistical applications, errors represent unsystematic sources of effects and therefore they are not to be correlated with other systematic sources. This implies that submatrix Φ_{21} is a null matrix. However, Φ_{21} being null is not enforced in the LINEQS model specification. If you ever specify a covariance between an error variable and a non-error independent variable in the COV statement, as a workaround trick or otherwise, you should provide your own theoretical justifications.

Summary of Matrices and Submatrices in the LINEQS Model

Let n_d be the number of dependent variables and n_i be the number of independent variables. The names, roles, and dimensions of the LINEQS model matrices and submatrices are summarized in the following table:

Matrix	Name	Description	Dimensions
Model Matrices			
α	_EQSALPHA_	intercepts of dependent variables	$n_d \times 1$
β	_EQSBETA_	effects of dependent (column) variables on dependent (row) variables	$n_d \times n_d$
γ	_EQSGAMMA_	effects of independent (column) variables on dependent (row) variables	$n_d \times n_i$
ν	_EQSNU_	means of independent variables	$n_i \times 1$
Φ	_EQSPHI_	covariance matrix of independent variables	$n_i \times n_i$
Submatrices			
γ_0	_EQSGAMMA_SUB_	effects of independent variables, excluding errors, on dependent variables	$n_d \times (n_i - n_d)$
ν_0	_EQSNU_SUB_	means of independent variables, excluding errors	$(n_i - n_d) \times 1$
Φ_{11}	_EQSPHI11_	covariance matrix of independent variables, excluding errors	$(n_i - n_d) \times (n_i - n_d)$
Φ_{21}	_EQSPHI21_	covariances of errors with other independent variables	$n_d \times (n_i - n_d)$
Φ_{22}	_EQSPHI22_	covariance matrix of errors	$n_d \times n_d$

Specification of the LINEQS Model

Specification in Equations

In the LINEQS statement, you specify intercepts and effect parameters (or regression coefficients) along with the variable relationships in equations. In terms of model matrices, you specify the α vector and the β and γ matrices in the LINEQS statement without using any matrix language.

For example:

$$Y = b_0 + b_1 X_1 + b_2 F_2 + E_1$$

In this equation, you specify Y as an outcome variable, X_1 and F_2 as predictor variables, and E_1 as an error variable. The parameters in the equation are the intercept b_0 and the path coefficients (or effects) b_1 and b_2.

This kind of model equation is specified in the LINEQS statement. For example, the previous equation translates into the following LINEQS statement specification:

```
lineqs Y = b0 Intercept + b1 X1 + b2 F2 + E1;
```

If the mean structures of the model are not of interest, the intercept term can be omitted. The specification becomes:

```
lineqs Y =  b1 X1 + b2 F2 + E1;
```

See the LINEQS statement on page 6766 for the details in syntax.

Because of the LINEQS model restrictions (see the section "Partitions of Some LINEQS Model Matrices and Their Restrictions" on page 6842), you must also follow these rules when specifying LINEQS model equations:

- A dependent variable can appear only on the left-hand side of an equation once. In other words, you must put all predictor variables for a dependent variable in one equation. This is different from some econometric models where a dependent variable can appear on the left-hand sides of two equations to represent an equilibrium point. This limitation, however, can be resolved by reparameterization in some cases. See Example 88.2.
- A dependent variable that appears on the left-hand side of an equation cannot appear on the right-hand side of the same equation. If you measure the same characteristic at different time points and the previous measurement serves as a predictor of the next measurement, you should use different variable names for the measurements so as to comply with this rule.
- An error term must be specified in each equation and must be unique. The same error name cannot appear in two or more equations. When an equation is truly intended to have no error term, it should be represented equivalently in the LINEQS equation by introducing an error term with zero variance (specified in the STD statement).
- The regression coefficient or effect associated with an error term must be fixed at one (1.0). This is done automatically by omitting any fixed constants or parameters associated with the error terms. Inserting a parameter or a fixed value other than one immediately before an error term is not allowed.

Mean, Variance, and Covariance Parameter Specification

In addition to the intercept and effect parameters specified in equations, the means, variances, and covariances among all independent variables are parameters in the LINEQS model. An exception is that the means of all error variables are restricted to fixed zeros in the LINEQS model. To specify the mean, variance, and covariance parameters, you use the MEAN, STD, and the COV statements, respectively.

The means, variances, and covariances among dependent variables are not parameters themselves in the model. Rather, they are complex functions of the model parameters. See the section "Matrix Representation of the LINEQS Model" on page 6841 for mathematical details.

Default Parameters in the LINEQS Model

Model-restricted values in the LINEQS model include the zero direct effects of any variables on themselves (that is, the diagonal of β matrix contains zeros only), the predetermined effects of error variables (that is, each element of the E submatrix in γ is always either one or zero) and the predetermined means of error variables (that is, ν elements pertaining to error means are always zero). These fixed values are always enforced and cannot be specified differently. All other elements or locations in the LINEQS model matrices can be specified as parameters (free, constrained, or fixed) in the LINEQS and the subsidiary model specification statements, as described in the previous section.

If a location (or an element) in a LINEQS model matrix is not model-restricted and you do not specify a parameter for it, a default parameter will be assumed for this location. There are two types of default parameters. One is automatic free parameters and the other is fixed zeros.

Automatic Free Parameters

In the LINEQS model, automatic free parameters are generated for the variances of all independent variables (manifest variables, latent factors, and latent errors), the covariances among all manifest independent variables (but not latent factors nor errors), and the means of all manifest independent variables if the mean structures are modeled. The name of each automatic free parameter is generated with the _Add prefix and appended with a unique integer.

In terms of LINEQS model matrices, automatic parameters are applied to all diagonal elements of the Φ_{11} submatrix, to all off-diagonal elements of the Φ_{11} submatrix pertaining to the manifest independent variable in rows and in columns, to the diagonal elements of the Φ_{22} submatrix, and to the elements in the ν_0 subvector pertaining to the manifest independent variables in rows.

Default Fixed Zeros

Unspecified locations (elements) of LINEQS model matrices that are neither model-restricted nor automatic free parameters will be fixed at zeros by default.

Rationale of the Default Parameters in the LINEQS Model

To explain the automatic free parameters in the LINEQS model, note that manifest independent variables are explanatory variables in your model. Their means, variances, and covariances are not functions of other parameters in the model. Rather, together with other parameters, they explain the means, variances, and covariances of the dependent variables. Therefore, means, variances, and covariances of manifest independent variables should be saturated as free parameters unless you are also testing a theoretical pattern of these parameters. Otherwise, failing to do this might add unnecessary constraints to your model.

The same argument applies to the latent independent variables. However, because latent variable means, variances, and covariances are not observed, the treatment is different. Only the variances of latent independent variables (factors or errors) are automatic free parameters. The means and covariances among latent independent variables are fixed zeros by default.

You can override the default parameters by manual specification in the STD, COV, and MEAN statements. Manual specification of these parameters becomes necessary when you are testing theoretical covariance or mean patterns or when these parameters are constrained with other parameters in your model. For example, when you test a fixed covariance matrix pattern, you have to manually specify the fixed values for the variances of, and covariances among, exogenous manifest variables in the STD and COV statements.

The LISMOD Model and Submodels

As a statistical model, the LISMOD modeling language is derived from the LISREL model proposed by Jöreskog and others (see Keesling 1972; Wiley 1973; Jöreskog 1973). But as a computer language, the LISMOD modeling language is quite different from the LISREL program. To maintain the consistence of specification syntax within the TCALIS procedure, the LISMOD modeling language departs from the original LISREL programming language. In addition, to make the programming a little easier, some terminological changes from LISREL are made in LISMOD.

For brevity, models specified by the LISMOD modeling language are called LISMOD models, although it is noted that you can also specify these so-called LISMOD models by other general modeling languages supported in PROC TCALIS.

The following descriptions of LISMOD models are basically the same as that of the original LISREL models. The main modifications are the names for the model matrices.

The LISMOD model is described by three component models. The first one is the structural equation model that describes the relationships among latent constructs or factors. The other two are measurement models that relate latent factors to manifest variables.

Structural Equation Model

The structural equation model for latent factors is:

$$\eta = \alpha + \beta\eta + \Gamma\xi + \zeta$$

where:
η is a random vector of n_η endogenous latent factors
ξ is a random vector of n_ξ exogenous latent factors
ζ is a random vector of errors
α is a vector of intercepts
β is a matrix of regression coefficients of η variables on other η variables
Γ is a matrix of regression coefficients of η on ξ

There are some assumptions in the structural equation model. To prevent a random variable in η from regressing directly on itself, the diagonal elements of β are assumed to be zeros. Also, $(I - \beta)^{-1}$ is assumed to be nonsingular, and ζ is uncorrelated with ξ.

The covariance matrix of ζ is denoted by Ψ and its expected value is a null vector. The covariance matrix of ξ is denoted by Φ and its expected value is denoted by κ.

Because variables in the structural equation model are not observed, to analyze the model these latent variables must somehow relate to the manifest variables. The measurement models, which will be discussed in the subsequent sections, provide such relations.

Measurement Model for y

$$y = \nu_y + \Lambda_y \eta + \epsilon$$

where:
y is a random vector of n_y manifest variables
ϵ is a random vector of errors for y
ν_y is a vector of intercepts for y
Λ_y is a matrix of regression coefficients of y on η

It is assumed that ϵ is uncorrelated with either η or ζ. The covariance matrix of ϵ is denoted by Θ_y and its expected value is the null vector.

Measurement Model for x

$$x = \nu_x + \Lambda_x \xi + \delta$$

where:
x is a random vector of n_x manifest variables
δ is a random vector of errors for x
ν_x is a vector of intercepts for x
Λ_x is a matrix of regression coefficients of x on ξ

It is assumed that δ is uncorrelated with ξ, ϵ, or ζ. The covariance matrix of δ is denoted by Θ_x and its expected value is a null vector.

Covariance and Mean Structures

Under the structural and measurement equations and the model assumptions, the covariance structures of the manifest variables $(y', x')'$ are expressed as:

$$\Sigma = \begin{pmatrix} \Lambda_y (I-\beta)^{-1}(\Gamma\Phi\Gamma' + \Psi)(I-\beta')^{-1}\Lambda_y' + \Theta_y & \Lambda_y (I-\beta)^{-1}\Gamma\Phi\Lambda_x' \\ \Lambda_x \Phi\Gamma'(I-\beta')^{-1}\Lambda_y' & \Lambda_x \Phi\Lambda_x' + \Theta_x \end{pmatrix}$$

The mean structures of the manifest variables $(y', x')'$ are expressed as:

$$\mu = \begin{pmatrix} \nu_y + \Lambda_y (I-\beta)^{-1}(\alpha + \Gamma\kappa) \\ \nu_x + \Lambda_x \kappa \end{pmatrix}$$

Model Matrices in the LISMOD Model

The parameters of the LISMOD model are elements in the model matrices, which are summarized as follows:

Matrix	Name	Description	Dimensions	Row Variables	Column Variables
α	_ALPHA_	intercepts for η	$n_\eta \times 1$	η (ETAVAR=)	N/A
β	_BETA_	effects of η on η	$n_\eta \times n_\eta$	η (ETAVAR=)	η (ETAVAR=)
Γ	_GAMMA_	effects of ξ on η	$n_\eta \times n_\xi$	η (ETAVAR=)	ξ (XIVAR=)
Ψ	_PSI_	error covariance matrix for η	$n_\eta \times n_\eta$	η (ETAVAR=)	η (ETAVAR=)
Φ	_PHI_	covariance matrix for ξ	$n_\xi \times n_\xi$	ξ (XIVAR=)	ξ (XIVAR=)
κ	_KAPPA_	mean vector for ξ	$n_\xi \times 1$	ξ (XIVAR=)	N/A
ν_y	_NUY_	intercepts for y	$n_y \times 1$	y (YVAR=)	N/A
Λ_y	_LAMBDAY_	effects of η on y	$n_y \times n_\eta$	y (YVAR=)	η (ETAVAR=)
Θ_y	_THETAY_	error covariance matrix for y	$n_y \times n_y$	y (YVAR=)	y (YVAR=)
ν_x	_NUX_	intercepts for x	$n_x \times 1$	x (XVAR=)	N/A
Λ_x	_LAMBDAX_	effects of ξ on x	$n_x \times n_\xi$	x (XVAR=)	ξ (XIVAR=)
Θ_x	_THETAX_	error covariance matrix for x	$n_x \times n_x$	x (XVAR=)	x (XVAR=)

There are twelve model matrices in the LISMOD model. Not all of them are used in all situations. See the section "LISMOD Submodels" on page 6851 for details. In the table, each model matrix is given a name in the column Name, followed by a brief description of the parameters in the matrix, the dimensions, and the row and column variables being referred to. In the second column of the table, the LISMOD matrix names are used in the MATRIX statements when specifying the LISMOD model. In the last two columns of the table, following the row or column variables are the variable list (for example, ETAVAR=, YVAR=, and so on) in parentheses. These lists are used in the LISMOD statement for defining variables.

Specification of the LISMOD Model

The LISMOD specification is characterized by two tasks. The first task is to define the variables in the model. The second task is to specify the parameters in the LISMOD model matrices.

Specifying Variables

The first task is accomplished in the LISMOD statement. In the LISMOD statement, you define the lists of variables of interest: YVAR=, XVAR=, ETAVAR=, and XIVAR= lists, respectively for the y-variables, x-variables, η-variables, and the ξ-variables. While you provide the names of variables in these lists, you also define implicitly the numbers of four types of variables: n_y, n_x, n_η, and n_ξ. The variables on the YVAR= and XVAR= lists are manifest variables and therefore must be present in the analyzed data set. The variables on the ETAVAR= and XIVAR= lists are latent factors, the names of which are assigned by the researcher to represent their roles in the substantive theory. Once these lists are defined, the dimensions of the model matrices are also defined by the number of variables on various lists. In addition, the variable orders on the lists are referred to by the row and column variables of the model matrices.

Specifying Parameters in Model Matrices

The second task is accomplished by the MATRIX statements. In each MATRIX statement, you specify the model matrix by using the matrix names described in the previous table. Then you specify the parameters (free or fixed) in the locations of the model matrix. You can use as many MATRIX statements as needed for defining your model. But each model matrix can only be specified in one MATRIX statement and each MATRIX statement is used for specifying one model matrix.

An Example

In the section "LISMOD Model" on page 6709, the LISMOD modeling language is used to specify the model described in the section "A Structural Equation Example" on page 6704. In the LISMOD statement, you define four lists of variables, as shown in the following statement:

```
lismod
   yvar  = Anomie67 Powerless67 Anomie71 Powerless71,
   xvar  = Education SEI,
   etav  = Alien67 Alien71,
   xivar = SES;
```

Endogenous latent factors are specified in the ETAVAR= list. Exogenous latent factors are specified in the XIVAR= list. In this case, Alien67 and Alien71 are the η-variables, and SES is the only ξ-variable in the model. Manifest variables that are indicators of endogenous latent factors in η are specified in the YVAR= list. In this case, they are the Anomie and Powerless variables, measured in two different time points. Manifest variables that are indicators of exogenous latent factors in ξ are specified in the XVAR= list. In this case, they are the Education and the SEI variables. Implicitly, the dimensions of the model matrices are defined by these lists already; that is, $n_y = 4$, $n_x = 2$, $n_\eta = 2$, and $n_\xi = 1$.

The MATRIX statements are used to specify parameters in the model matrices. For example, in the following statement you define the _LAMBDAX_ (Λ_x) matrix with two nonzero entries:

```
matrix _LAMBDAX_ [1,1] = 1.0,
                 [2,1] = lambda;
```

The first parameter location is for [1,1], which is the effect of SES (the first variable in the XIVAR= list) on Education (the first element in the XVAR= list). A fixed value of 1.0 is specified there. The second parameter location is for [2,1], which is the effect of SES (the first variable in the XIVAR= list) on SEI (the second variable in the XVAR= list). A parameter named lambda without initial value is specified there.

Another example is shown as follows:

```
matrix _THETAY_  [1,1] = theta1,
                 [2,2] = theta2,
                 [3,3] = theta1,
                 [4,4] = theta2,
                 [3,1] = theta5,
                 [4,2] = theta5;
```

In this MATRIX statement, the error variances and covariances (that is, the Θ_y matrix) for the y-variables are specified. The diagonal elements of the _THETAY_ matrix are specified by parameters theta1, theta2, theta1, and theta2, respectively, for the four y-variables Anomie67, Powerless67, Anomie71, and Powerless71. By using the same parameter name theta1, the error variances for Anomie67 and Anomie71 are implicitly constrained. Similarly, the error variances for Powerless67 and Powerless71 are also implicitly constrained. Two more parameter locations are specified. The error covariance between Anomie67 and Anomie71 and the error covariance between Powerless67 and Powerless71 are both represented by the parameter theta5. Again, this is an implicit constraint on the covariances. All other unspecified elements in the _THETAY_ matrix are treated as fixed zeros.

In this example, no parameters are specified for matrices _ALPHA_, _KAPPA_, _NUY_, or _NUX_. Therefore, mean structures are not modeled.

LISMOD Submodels

It is not necessary to specify all four lists of variables in the LISMOD statement. When some lists are unspecified in the LISMOD statement, PROC TCALIS will analyze submodels derived logically from the specified lists of variables. For example, if only y- and x- variable lists are specified, the submodel being analyzed would be a multivariate regression model with manifest variables only. Not all combinations of lists will lead to meaningful submodels, however. To determine whether and how a submodel (which is formed by a certain combination of variable lists) can be analyzed, the following three principles in the LISMOD modeling language are applied:

- Submodels with at least one of the YVAR= and XVAR= lists are required.
- Submodels that have an ETAVAR= list but no YVAR= list cannot be analyzed.
- When a submodel has a YVAR= (an XVAR=) list but without an ETAVAR= (a XIVAR=) list, it is assumed that the set of y-variables (x-variables) is equivalent to the η-variables (ξ-variables). Hereafter, this principle is referred to as an equivalence interpretation.

Apparently, the third principle is the same as the situation where the latent factors η (or ξ) are perfectly measured by the manifest variables y (or x). That is, in such a perfect measurement model,

Λ_y (Λ_x) is an identity matrix and Θ_y (Θ_x) and ν_y (ν_x) are both null. This can be referred to as a perfect measurement interpretation. However, the equivalence interpretation stated in the last principle presumes that there are actually no measurement equations at all. This is important because under the equivalence interpretation, matrices Λ_y (Λ_x), Θ_y (Θ_x) and ν_y (ν_x) are non-existent rather than fixed quantities, which is assumed under the perfect measurement interpretation. Hence, the x-variables are treated as *exogenous* variables with the equivalence interpretation, but they are still treated as *endogenous* with the perfect measurement interpretation. Ultimately, whether x-variables are treated as exogenous or endogenous will affect the default or automatic parameterization. See the section "Default Parameters in the LISMOD Model" on page 6854 for more details.

By using these three principles, the models and submodels that PROC TCALIS analyzes are summarized in the following table, followed by detailed descriptions of these models and submodels.

	Presence of Lists	Description	Model Equations	Non-fixed Model Matrices
Presence of Both x and y Variables				
1	YVAR=, ETAVAR=, XVAR=, XIVAR=	full model	$y = \nu_y + \Lambda_y\eta + \epsilon$ $x = \nu_x + \Lambda_x\xi + \delta$ $\eta = \alpha + \beta\eta + \Gamma\xi + \zeta$	ν_y,Λ_y,Θ_y ν_x,Λ_x,Θ_x,κ,Φ α ,β,Γ,Ψ
2	YVAR=, XVAR=, XIVAR=	full model with $y \equiv \eta$	$x = \nu_x + \Lambda_x\xi + \delta$ $y = \alpha + \beta y + \Gamma\xi + \zeta$	ν_x,Λ_x,Θ_x,κ,Φ α,β,Γ,Ψ
3	YVAR=, ETAVAR=, XVAR=	full model with $x \equiv \xi$	$y = \nu_y + \Lambda_y\eta + \epsilon$ $\eta = \alpha + \beta\eta + \Gamma x + \zeta$	ν_y,Λ_y,Θ_y α ,β,Γ,Ψ,κ,Φ
4	YVAR=, XVAR=	regression $(y \equiv \eta)$ $(x \equiv \xi)$	$y = \alpha + \beta y + \Gamma x + \zeta$, or $(I - \beta)^{-1} y = \alpha + \Gamma x + \zeta$	α,β,Γ,Ψ,κ,Φ
Presence of x Variables and Absence of y Variables				
5	XVAR=, XIVAR=	factor model for x	$x = \nu_x + \Lambda_x\xi + \delta$	ν_x,Λ_x,Θ_x,κ,Φ
6	XVAR=	x-structures $(x \equiv \xi)$		κ,Φ
Presence of y Variables and Absence of x Variables				
7	YVAR=, ETAVAR=	factor model for y	$y = \nu_y + \Lambda_y\eta + \epsilon$ $\eta = \alpha + \beta\eta + \zeta$	ν_y,Λ_y,Θ_y α,β,Ψ
8	YVAR=	y-structures $(y \equiv \eta)$	$y = \alpha + \beta y + \zeta$, or $(I - \beta)^{-1} y = \alpha + \zeta$	α,β,Ψ
9	YVAR=, ETAVAR=, XIVAR=	second-order factor model	$y = \nu_y + \Lambda_y\eta + \epsilon$ $\eta = \alpha + \beta\eta + \Gamma\xi + \zeta$	ν_y,Λ_y,Θ_y α,β,Γ,Ψ,κ,Φ
10	YVAR=, XIVAR=	factor model $(y \equiv \eta)$	$y = \alpha + \beta y + \Gamma\xi + \zeta$, or $(I - \beta)^{-1} y = \alpha + \Gamma\xi + \zeta$	α,β,Γ,Ψ,κ,Φ

Models 1, 2, 3, and 4 — Presence of Both x and y Variables

Submodels 1, 2, 3, and 4 are characterized by the presence of both x- and y- variables in the model. Model 1 is in fact the full model with the presence of all four types of variables. All twelve model matrices are parameter matrices in this model.

Depending on the absence of the latent factor lists, manifest variables can replace the role of the latent factors in models 2–4. For example, the absence of the ETAVAR= list in model 2 means y is equivalent to η ($y \equiv \eta$). Consequently, you cannot, nor do you need to, use the MATRIX statement to specify parameters in the _LAMBDAY_, _THETAY_, or _NUY_ matrices under this model. Similarly, because x is equivalent to ξ ($x \equiv \xi$) in model 3, you cannot, nor do you need to, use the MATRIX statement to specify the parameters in the _LAMBDAX_, _THETAX_, or _NUX_ matrices. In model 4, y is equivalent to η ($y \equiv \eta$) and x is equivalent to ξ ($x \equiv \xi$). None of the six model matrices in the measurement equations are defined in the model. Matrices in which you can specify parameters by using the MATRIX statement are listed in the last column of the table.

Describing model 4 as a regression model is a simplification. Because y can regress on itself in the model equation, the regression description is not totally accurate for model 4. Nonetheless, if β is a null matrix, the equation describes a multivariate regression model with outcome variables y and predictor variables x. This model is the TYPE 2A model in LISREL VI (Jöreskog and Sörbom, 1985).

You should also be aware of the changes in meaning of the model matrices when there is an equivalence between latent factors and manifest variables. For example, in model 4 the Φ and κ are now the covariance matrix and mean vector, respectively, of manifest variables x, while in model 1 (the complete model) these matrices are of the latent factors ξ.

Models 5 and 6 — Presence of x Variables and Absence of y Variables

Models 5 and 6 are characterized by the presence of the x-variables and the absence of y-variables.

Model 5 is simply a factor model for measured variables x, with Λ_x representing the factor loading matrix, Θ_x the error covariance matrix, and Φ the factor covariance matrix. If mean structures are modeled, κ represents the factor means and ν_x is the intercept vector. This is the TYPE 1 submodel in LISREL VI (Jöreskog and Sörbom, 1985).

Model 6 is a special case where there is no model equation. You specify the mean and covariance structures (in κ and Φ, respectively) for the manifest variables x directly. The x-variables are treated as exogenous variables in this case. Because this submodel uses direct mean and covariance structures for measured variables, it can also be handled more easily by the MSTRUCT modeling language. See the MSTRUCT statement and the section "The MSTRUCT Model" on page 6877 for more details.

Note that because η-variables cannot exist in the absence of y-variables (see one of the three aforementioned principles in deriving submodels), adding the ETAVAR= list alone to these two submodels does not generate new submodels that can be analyzed by PROC TCALIS.

Models 7, 8 ,9 and 10—Presence of y Variables and Absence of x Variables

Models 7–10 are characterized by the presence of the y-variables and the absence of x-variables.

Model 7 is a factor model for y-variables (TYPE 3B submodel in LISREL VI). It is similar to model 5, but with regressions among latent factors allowed. When β is null, model 7 functions the same as model 5. It becomes a factor model for y-variables, with Λ_y representing the factor loading matrix, Θ_y the error covariance matrix, Ψ the factor covariance matrix, α the factor means, and ν_y the intercept vector.

Model 8 (TYPE 2B submodel in LISREL VI) is a model for studying the mean and covariance structures of y-variables, with regression among y-variables allowed. When β is null, the mean structures of y are specified in α and the covariance structures are specified in Ψ. This is similar to model 6. However, there is an important distinction. In model 6, the x variables are treated as exogenous (no model equation at all). But the y-variables are treated as endogenous in model 8 (with or without $\beta = 0$). Consequently, the default parameterization would be different for these two submodels. See the section "Default Parameters in the LISMOD Model" on page 6854 for details about the default parameterization.

Model 9 represents a modified version of second-order factor model for y. It would be a standard second-order factor model when β is null. This is the TYPE 3A submodel in LISREL VI. With β being null, η represents the first-order factors and ξ represents the second-order factors. The first- and second- order factor loading matrices are Λ_y and Γ, respectively.

Model 10 is another form of factor model when β is null, with factors represented by ξ and manifest variables represented by y. However, if β is indeed a null matrix in applications, you might want to use model 5, in which the factor model specification is more direct and intuitive.

Default Parameters in the LISMOD Model

When a model matrix is defined in a LISMOD model (or submodel), you can specify fixed values or parameters for the elements in the matrix. See the MATRIX statement on page 6784 for the syntax of specification. All other unspecified elements in the model matrices will be set by default. There are two types of default parameters: one is automatic free parameters; the other is fixed zeros.

Automatic Free Parameters

Automatic additions of free parameters are done to ensure the model is properly parameterized. Each automatic parameters is prefixed with _Add and appended with an unique integer. The following rules are used to add parameters in the LISMOD model matrices:

- _THETAX_, _THETAY_, _PSI_, and _PHI_ matrices: If a matrix in this group is defined in a LISMOD submodel or full model, its diagonal elements are automatically represented by free parameters unless they are specified explicitly in the MATRIXstatements. This ensures that these covariance matrices will have nonzero variances.
- _PHI_ matrix: For LISMOD submodels with XVAR= list specified but XIVAR= unspecified, all the off-diagonal elements of the _PHI_ are automatic free parameters. In this case, because ξ and x are equivalent, its covariance matrix Φ is the covariance matrix of **exogenous**

manifest variables. Because exogenous covariances among manifest variables are not predicted as functions of any other parameters in the LISMOD model, they should be saturated by free parameters in the model unless there are theoretical reasons not to do so (for example, when you are testing a fixed pattern of these covariances). Failure to treat covariances among mainfest variables as free parameters is equivalent to fixing these values to constants, which leads to an overly restrictive model.

- _KAPPA_ matrix: For LISMOD submodels with XVAR= list specified but XIVAR= unspecified, all the elements of the _KAPPA_ vector are automatic free parameters if the mean structures are fitted. The reason for this is the same as the previous rule for adding covariance parameters among exogenous manifest variables. Exogenous manifest variables serve as explanatory variables in your model, and their means are not predicted by relationships with other variables. Unless you are also testing hypothesized mean values for the exogenous manifest variables, in general you should allow these means to be free parameters in the LISMOD model.

These three rules for automatic free parameters are in line with the treatments in the LINEQS, RAM, and PATH models. See, for example, the section "Default Parameters in the LINEQS Model" on page 6846 and the section "Rationale of the Default Parameters in the LINEQS Model" on page 6846 for details about the treatments of automatic free parameters.

Default Fixed Zeros

Matrix elements that are unspecified and are not automatic free parameters will be fixed at zeros by default.

The RAM Model

The RAM modeling language is adapted from the basic RAM model developed by McArdle (1980). For brevity, models specified by the RAM modeling language are called RAM models, although it is noted that you can also specify these so-called RAM models by other general modeling languages supported in PROC TCALIS.

Types of Variables in the RAM Model

A variable in the RAM model is manifest if it is observed and is defined in the input data set. A variable in the RAM model is latent if it is not manifest. Because error variables are not explicitly named in the RAM model, all latent variables in the RAM model are considered as factors (non-error-type latent variables).

A variable in the RAM model is endogenous if it ever serves as an outcome variable in the RAM model. That is, an endogenous variable has at least one path (or an effect) from another variable in the model. A variable is exogenous if it is not endogenous. Endogenous variables are also referred to as dependent variables, while exogenous variables are also referred to as independent variables.

In the RAM model, exogenous/endogenous and latent/manifest distinctions for variables are not essential to the definitions of model matrices, although they are useful for conceptual understanding when the model matrices are partitioned.

Naming Variables in the RAM Model

Manifest variables in the RAM model are referenced in the input data set. Their names must not be longer than 32 characters. There is no further restrictions beyond those required by the SAS system.

Latent variables in the RAM model are those not being referenced in the input data set. Their names must not be longer than 32 characters. Unlike the LINEQS model, you do not need to use any specific prefix (for example, 'F' or 'f') for the latent factor names. The reason is that error or disturbance variables in the RAM model are not named explicitly in the RAM model. Thus, any variable names that are not referenced in the input data set are for latent factors.

As a general naming convention, you should not use Intercept as either a manifest or latent variable name.

Model Matrices in the RAM Model

In terms of the number of model matrices involved, the RAM model is the simplest among all the general structural equations models supported by PROC TCALIS. Essentially, there are only three model matrices in the RAM model: one for the interrelationships among variables, one for the variances and covariances, and one for the means and intercepts. These matrices are discussed in the following.

Matrix A $(n_a \times n_a)$: Effects of Column Variables on Row Variables

The row and column variables of matrix A are the set of manifest and latent variables in the RAM model. Unlike the LINEQS model, the set of latent variables in the RAM model matrix does not include the error or disturbance variables. Each entry or element in the A matrix represents an effect of the associated column variable on the associated row variable or a path coefficient from the associated column variable to the associated row variable. A zero entry means an absence of a path or an effect.

The pattern of matrix A determines whether a variable is endogenous or exogenous. A variable in the RAM model is endogenous if its associated row in the A matrix has at least one nonzero entry. Any other variable in the RAM model is exogenous.

Mathematically, you do not need to arrange the set of variables for matrix A in a particular order, as long as the order of variables is the same for the rows and the columns. However, arranging the variables according to whether they are endogenous or exogenous is useful to show the partitions of the model matrices and certain mathematical properties. See the section "Partitions of the RAM Model Matrices and Some Restrictions" on page 6858 for details.

Matrix P ($n_a \times n_a$): Variances, Covariances, Partial Variances, and Partial Covariances

The row and column variables of matrix P refer to the same set of manifest and latent variables defined in the RAM model matrix A. The diagonal entries of P contain variances or partial variances of variables. If a variable is exogenous, then the corresponding diagonal element in the P matrix represents its variance. Otherwise, the corresponding diagonal element in the P matrix represents its partial variance. This partial variance is an unsystematic source of variance not explained by the interrelationships of variables in the model. In most cases, you can interpret a partial variance as the error variance for an endogenous variable.

The off-diagonal elements of P contain covariances or partial covariances among variables. An off-diagonal element in P associated with exogenous row and column variables represents covariance between the two exogenous variables. An off-diagonal element in P associated with endogenous row and column variables represents **partial** covariance between the two variables. This partial covariance is unsystematic, in the sense that it is not explained by the interrelationships of variables in the model. In most cases, you can interpret a partial covariance as the error covariance between the two endogenous variables involved. An off-diagonal element in P associated with one exogenous variable and one endogenous variable in the row and column represents the covariance between the exogenous variable and the error of the endogenous variable. While this interpretation sounds a little awkward and inelegant, this kind of covariance, fortunately, is rare in most applications.

Vector W ($n_a \times 1$): Intercepts and Means

The row variables of vector W refer to the same set of manifest and latent variables defined in the RAM model matrix A. Elements in W represent either intercepts or means. An element in W associated with an exogenous row variable represents the mean of the variable. An element in W associated with an endogenous row variable represents the intercept term for the variable.

Covariance and Mean Structures

Assuming that $(I - A)$ is invertible, where I is the identity matrix of the same dimension as A, the structured covariance matrix of all variables (including latent variables) in the RAM model is shown as follows:

$$\Sigma_a = (I - A)^{-1} P (I - A)^{-1\prime}$$

The structured mean vector of all variables is shown as follows:

$$\mu_a = (I - A)^{-1} W$$

The covariance and mean structures of all manifest variables are obtained by selecting the elements in Σ_a and μ_a. This can be achieved by defining a selection matrix G of dimensions $n \times n_a$, where n is the number of manifest variables in the model. The selection matrix G contains zeros and ones as its elements. Each row of G has exactly one nonzero element at the position that corresponds to the location of a manifest row variable in Σ_a or μ_a. With each row of G selecting a distinct manifest variable, the structured covariance matrix of all manifest variables is expressed as the following:

$$\Sigma = G \Sigma_a G'$$

The structured mean vector of all observed variables is expressed as the following:

$$\mu = G \mu_a$$

Partitions of the RAM Model Matrices and Some Restrictions

There are some model restrictions in the RAM model matrices. Although these restrictions do not affect the derivation of the covariance and mean structures, they are enforced in the RAM model specification.

For convenience, it is useful to assume that n_a variables are arranged in the order of n_d endogenous (or dependent) variables and the n_i exogenous (independent) variables in the rows and columns of the model matrices.

Model Restrictions on the A Matrix

The A matrix is partitioned as:

$$A = \begin{pmatrix} \beta & \gamma \\ 0 & 0 \end{pmatrix}$$

where β is an $n_d \times n_d$ matrix for paths or effects from (column) endogenous variables to (row) endogenous variables and γ is an $n_d \times n_i$ matrix for paths or effects from (column) exogenous variables to (row) endogenous variables.

As shown in the matrix partitions, there are four submatrices. The two at the lower parts are seemingly structured to zeros. However, this should not be interpreted as restrictions imposed by the model. The zero submatrices are artifacts created by the exogenous-endogenous arrangement of the row and column variables. The only restriction on the A matrix is that the diagonal elements must all be zeros. This implies that the diagonal elements of the submatrix β are also zeros. This restriction prevents a direct path from any endogenous variable to itself. There are no restrictions on the pattern of γ.

Partition of the P Matrix

The P matrix is partitioned as:

$$P = \begin{pmatrix} P_{11} & P_{21}' \\ P_{21} & P_{22} \end{pmatrix}$$

where P_{11} is an $n_d \times n_d$ partial covariance matrix for the endogenous variables, P_{22} is an $n_i \times n_i$ covariance matrix for the exogenous variables, and P_{21} is an $n_i \times n_d$ covariance matrix between the exogenous variables and the error terms for the endogenous variables. Because P is symmetric, P_{11} and P_{22} are also symmetric.

There are virtually no model restrictions placed on these submatrices. However, in most statistical application, errors for endogenous variables represent unsystematic sources of effects and therefore they are not to be correlated with other systematic sources such as the exogenous variables in the RAM model. This means that in most practical applications P_{21} would be a null matrix, although this is not enforced in PROC TCALIS.

Partition of the W Vector

The W vector is partitioned as:

$$W = \begin{pmatrix} \alpha \\ \nu \end{pmatrix}$$

where α is an $n_d \times 1$ vector for intercepts of the endogenous variables and ν is an $n_i \times 1$ vector for the means of the exogenous variables. There is no model restriction on these subvectors.

Summary of Matrices and Submatrices in the RAM Model

Let n_a be the total number of manifest and latent variables in the RAM model. Of these n_a variables, n_d are endogenous and n_i are exogenous. Suppose that the rows and columns of the RAM model matrices A and P and the rows of W are arranged in the order of n_d endogenous variables and then n_i exogenous variables. The names, roles, and dimensions of the RAM model matrices and submatrices are summarized in the following table:

Matrix	Name	Description	Dimensions
Model Matrices			
A	_A_ or _RAMA_	effects of column variables on row variables, or paths from the column variables to the row variables	$n_a \times n_a$
P	_P_ or _RAMP_	(partial) variances and covariances	$n_a \times n_a$
W	_W_ or _RAMW_	intercepts and means	$n_a \times 1$
Submatrices			
β	_RAMBETA_	effects of endogenous variables on endogenous variables	$n_d \times n_d$
γ	_RAMGAMMA_	effects of exogenous variables on endogenous variables	$n_d \times n_i$
P_{11}	_RAMP11_	error variances and covariances for endogenous variables	$n_d \times n_d$
P_{21}	_RAMP21_	covariances between exogenous variables and error terms for endogenous variables	$n_d \times n_i$
P_{22}	_RAMP22_	variances and covariances for exogenous variables	$n_i \times n_i$
α	_RAMALPHA_	intercepts for endogenous variables	$n_d \times 1$
ν	_RAMNU_	means for exogenous variables	$n_i \times 1$

Specification of the RAM Model

In PROC CALIS, the RAM model specification is a matrix-oriented modeling language. That is, you have to define the row and column variables for the model matrices and specify the parameters in terms of matrix entries. This is not the case in PROC TCALIS. You no longer need to specify the RAM model parameters as matrix entries. Instead, you specify the model by using the variable names directly. Representing the RAM model matrices is done internally by PROC TCALIS.

To specify the RAM model parameters, you use the RAM statement to list each parameter location. There are three types of parameter listings, which correspond to the elements in the three model matrices. The three types of parameter listings are described in the following.

(1) Specification of Effects or Paths in Model Matrix A

If in your model there is a path from V2 to V1 and the associated effect parameter is named parm1 with 0.5 as the starting value, you can use one of the following list-entries of the RAM statement to specify this parameter location:

```
_A_      V1     V2   parm1  (0.5),
path     V1 <-  V2   parm1  (0.5),
path     V2 ->  V1   parm1  (0.5),
```

(2) Specification of (Partial) Variances and (Partial) Covariances in Model Matrix P

If V2 is an exogenous variable in the RAM model and you want to specify its variance as a parameter named parm2 with 10 as the starting value, you can use one of the following list-entries of the RAM statement:

```
_P_      V2     V2   parm2  (10.),
pvar     V2          parm2  (10.),
```

If V1 is an endogenous variable in the RAM model and you want to specify its partial variance or error variance as a parameter named parm3 with 5 as the starting value, you can use one of the following list-entries of the RAM statement:

```
_P_      V1     V1   parm3  (5.),
pvar     V1          parm3  (5.),
```

Notice that this partial variance specification has the same format as that for the variance parameter of an exogenous variable.

If you want to specify (partial) covariance between two variables V3 and V4 as a parameter named parm4 with 5 as the starting value, you can use one of the following list-entries of the RAM statement:

```
_P_      V3     V4   parm4  (5.),
pcov     V3     V4   parm4  (5.),
```

(3) Specification of Means and Intercepts in Model Matrix _W_

Means and intercepts are specified when the mean structures of the model are of interest. If V5 is an exogenous variable and you want to specify its mean as a parameter named parm5 with 11 as the starting value, you can use one of the following list-entries of the RAM statement:

```
_W_      V5    parm5  (11.),
mean     V5    parm5  (11.),
```

If V6 is an endogenous variable and you want to specify its intercept as a parameter named parm6 with 7 as the starting value, you can use one of the following list-entries of the RAM statement:

```
_W_          V6    parm6  (7.),
intercept    V6    parm6  (7.),
```

Note that this specification bears the same format as that for the mean parameter of an exogenous variable.

Specifying Parameters without Initial Values

If you do not have any knowledge about the initial value for a parameter, you can omit the initial value specification and let PROC TCALIS to compute it. For example, you can just provide the parameter locations and parameter names as shown in the following specifications:

```
_A_          V1     V2   parm1,
intercept    V6          parm6,
```

Specifying Fixed Parameter Values

If you want to specify a fixed parameter value, you do not need to provide the parameter name. Instead, you just provide the fixed value without parentheses at the end of the list-entry. The following shows examples for specifying fixed parameters:

```
_PCOV_       V3   V4     5.,
mean         V5          11.,
```

A Complete RAM Model Specification Example

Essentially, you can specify all parameters and their locations of the RAM model in the RAM statement. No subsidiary model specification statements are needed in the RAM model—that is, all information regarding the model is specified in the RAM statement. Each parameter location specification is a list-entry in the RAM statement, and these list-entries are separated by commas. For example,

```
ram
   path     V1 <-  V2   parm1  (.5),
   path     V1 <-  V3   parm2      ,
   pvar     V1          errv   (1.),
   pvar     V2          parm3  (10.),
   pvar     V3          parm4  (10.),
   pcov     V3     V2   parm5  (5.);
```

Notice that although the order of the list-entries is not critical to the model specification, it is the same order that is used in output displays. See the RAM statement on page 6805 for more details about the syntax.

Default Parameters in the RAM Model

The effect of any variable on itself is restricted to zero in the RAM model. In other words, it is invalid to specify:

```
ram
   path     V1  <-  V1  parm;
```

In matrix terms, you cannot have any nonzero diagonal element in the β submatrix. All other elements in the RAM model matrices can be specified as free or fixed parameters.

If an element in a model matrix is not subject to model restrictions and is not specified explicitly, a default parameter will be applied to the location. There are two types of default parameters: one is automatic free parameter; and the other is fixed zero.

Automatic Free Parameters

The set of automatic parameters in the RAM model is essentially derived in the same way as those in the LINEQS model. That is, in the LINEQS model the set of automatic parameters includes the variances of and covariances among manifest exogenous variables, the means of manifest exogenous variables when the mean structures are modeled, and the variances of all exogenous latent variables (factors and errors). All these automatic free parameters are applied similarly to the RAM model.

The only difficulty about this generalization is the definition of error variances in the RAM model. Unlike the LINEQS model, error terms in the RAM model are not named explicitly. The question is what the error variances in the LINEQS model correspond to in the RAM model. As described previously, error variances in the RAM model are recast as the partial variances of the corresponding endogenous variables. As a result, the set of automatic parameters in the RAM model includes the variances or partial variances of all variables—that is, variances when the variables are exogenous, partial variances when the variables are endogenous.

As in the LINEQS model, the reason for automatic parameter generation is to safeguard a proper RAM model specification. See the section "Rationale of the Default Parameters in the LINEQS Model" on page 6846 for a more detailed discussion.

In terms of the RAM model matrices, the set of automatic parameters in the RAM model includes:

- all P matrix diagonal elements, which are either variances of exogenous variables (manifest or latent) or partial variances of endogenous variables (manifest or latent)
- those P matrix off-diagonal elements with their rows and columns corresponding to manifest exogenous variables
- those W matrix elements with their rows corresponding to manifest exogenous variables, provided that the mean structures are modeled

Each of these automatic mean, intercept, variance, partial variance, covariance, and partial covariance parameters is named with the prefix _Add, and appended with an unique integer.

Default Fixed Zeros

All unspecified parameter locations that are neither model-restricted nor automatic free parameters in the RAM model matrices are fixed zeros by default.

Relating the RAM Model to the PATH Model

The mathematical model for the RAM and the PATH modeling languages is the same. As a result, the two modeling languages are also quite similar in their syntax. See the section "Relating the PATH Model to the RAM Model" on page 6876 for details.

The FACTOR Model

The FACTOR modeling language is used for specifying exploratory and confirmatory factor-analysis models. You can use other general modeling languages such as LINEQS, LISMOD, PATH, and RAM to specify a factor model. But the FACTOR modeling language is more convenient for specifying factor models and is more specialized in outputting factor-analytic results. For convenience, models specified by the FACTOR modeling language are called FACTOR models.

Types of Variables in the FACTOR Model

Each variable in the FACTOR model is either manifest or latent. Manifest variables are those variables that are measured in the research. They must be present in the input data set. Latent variables are not measured. Each latent variable in the FACTOR model can either be a factor or an error term.

Factors are unmeasured hypothetical constructs for explaining the covariances among manifest variables, while errors are the unique parts of the manifest variables that are not explained by the (common) factors.

In the FACTOR model, all manifest variables are endogenous, which means that they are predicted from the latent variables. In contrast, all latent variables in the FACTOR model are exogenous, which means that they serve as predictors only.

Naming Variables in the FACTOR Model

Manifest variables in the FACTOR model are referenced in the input data set. In the FACTOR model specification, you use their names as they appear in the input data set. Manifest variable names must not be longer than 32 characters. There are no further restrictions on these names beyond those required by the SAS system.

Error variables in the FACTOR model are not named explicitly, although they are assumed in the model. You can name latent factors only in confirmatory FACTOR models. Factor names must not be longer than 32 characters and must be distinguishable from the manifest variable names in the

same analysis. You do not need to name factors in exploratory FACTOR models, however. Latent factors named Factor1, Factor2, and so on are generated automatically in exploratory FACTOR models.

Model Matrices in the FACTOR Model

Suppose in the FACTOR model that there are p manifest variables and n factors. The FACTOR model matrices are summarized as follows:

Matrix F $(p \times n)$: Factor Loading Matrix

The rows of F represent the p manifest variables, while the columns represent the n factors. Each row of F contains the factor loadings of a variable on all factors in the model.

Matrix P $(n \times n)$: Factor Covariance Matrix

The P matrix is a symmetric matrix for the variances of and covariances among the n factors.

Matrix U $(p \times p)$: Error Covariance Matrix

The U matrix represents an $p \times p$ diagonal matrix for the error variances for the manifest variables. Elements in this matrix are the parts of variances of the manifest variables that are not explained by the common factors. Note that all off-diagonal elements of U are fixed zeros in the FACTOR model.

Vector a $(p \times 1)$: Intercepts

If the mean structures are analyzed, vector a represents the intercepts of the manifest variables.

Vector v $(n \times 1)$: Factor Means

If the mean structures are analyzed, vector v represents the means of the factors.

Matrix Representation of the FACTOR Model

Let y be a $p \times 1$ vector of manifest variables, ξ be an $n \times 1$ vector of latent factors, and e be a $p \times 1$ vector of errors. The factor model is written as:

$$y = a + F\xi + e$$

With the model matrix definitions in the previous section, the covariance matrix Σ $(p \times p)$ of manifest variables is structured as:

$$\Sigma = FPF' + U$$

The mean vector μ $(p \times p)$ of manifest variables is structured as:

$$\mu = a + Fv$$

Exploratory Factor Analysis Models

So far confirmatory and exploratory models are not distinguished in deriving the covariance and mean structures. These two types of models are now distinguished in terms of the required structures or restrictions in model matrices.

Traditionally, exploratory factor analysis is applied when the relationships of manifest variables with factors have not been well-established in research. All manifest variables are allowed to have nonzero loadings on the factors in the model. First, factors are extracted and an initial solution is obtained. Then, for ease of interpretation a final factor solution is usually derived by rotating the factor space. Factor-variable relationships are determined by interpreting the final factor solution. This is different from the confirmatory factor analysis in which the factor-variable relationships are prescribed and to be confirmed.

Nonetheless, in terms of initial model fitting, exploratory factor analysis is not different from confirmatory factor analysis. In PROC TCALIS, the initial exploratory factor solution is obtained from a specific confirmatory factor model with restricted model matrices, which are described as follows:

- The factor loading matrix F has $p \times (n-1)/2$ fixed zeros at the upper triangle portion of the matrix.
- The factor covariance matrix P is an identity matrix, which means that factors are not correlated.
- The error covariance matrix U is a diagonal matrix.
- The mean structures are not modeled. That is, no specification in a or v vector is allowed.

For example, for an analysis with nine variables and three factors, the relevant model matrices of an exploratory FACTOR model have the following patterns:

$$F = \begin{pmatrix} * & 0 & 0 \\ * & * & 0 \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \\ * & * & * \end{pmatrix}$$

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

and

$$U = \begin{pmatrix} * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & * & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & * & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & * \end{pmatrix}$$

where * denotes free parameters in the model matrices.

It is emphasized that most of these patterns are put on the initial factor solutions only. If an initial factor solution is rotated afterward, only the pattern of the error covariance matrix U remains unchanged. That is, it is still a diagonal matrix after rotation. In general, rotating a factor solution will eliminate the fixed zero pattern in the upper triangle of the factor loading matrix F. If you apply an orthogonal rotation, the factor covariance matrix P will not change. It is an identity matrix before and after rotation. However, if you apply an oblique rotation, in general the rotated factor covariance matrix P will not be an identity matrix and the off-diagonal elements will not be zeros.

Confirmatory Factor-Analysis Models

In confirmatory FACTOR models, there are no imposed patterns on the F, P, a, and v model matrices. All elements in these model matrices can be specified. However, for model identification, you might need to specify some factor loadings or factor variances as constants.

The only model restriction in confirmatory FACTOR models is placed on U, which must be a diagonal matrix, as in exploratory FACTOR models too.

For example, for a confirmatory factor analysis with nine variables and three factors, you might specify the following patterns for the model matrices:

$$F = \begin{pmatrix} 1 & 0 & 0 \\ * & 0 & 0 \\ * & 0 & 0 \\ 0 & 1 & 0 \\ 0 & * & 0 \\ 0 & * & 0 \\ 0 & 0 & 1 \\ 0 & 0 & * \\ 0 & 0 & * \end{pmatrix}$$

$$P = \begin{pmatrix} * & * & * \\ * & * & * \\ * & * & * \end{pmatrix}$$

and

$$U = \begin{pmatrix} * & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & * & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & * & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & * & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & * & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & * & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & * & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & * & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & * \end{pmatrix}$$

where * denotes free parameters in the model matrices.

In this confirmatory factor model, mean structures are not modeled. In addition, there are some distinctive features that underscore the differences between confirmatory and exploratory models:

- Factor loading matrix F contains mostly zero elements and few nonzero free parameters, a pattern which is seen in most confirmatory factor models. In contrast, in exploratory factor models most elements in the F matrix are nonzero parameters.
- Factor loading matrix F contains fixed values of ones. These fixed values are used for model identification purposes (that is, identifying the scales of the latent variables). In general, you always have to make sure that your confirmatory factor models are identified by putting fixed values in appropriate parameter locations in the model matrices. However, this is not a concern in exploratory FACTOR models because identification has been ensured by imposing certain patterns on the model matrices.
- The nonzero off-diagonal parameters in the factor covariance matrix P indicate that correlated factors are hypothesized in the confirmatory factor model. This cannot be the case with the initial model of exploratory FACTOR models, where the P matrix must be an identity matrix before rotation.

Summary of Matrices in the FACTOR Model

Let p be the number of manifest variables and n be the number of factors in the FACTOR model. The names, roles, and dimensions of the FACTOR model matrices are shown in the following table:

Matrix	Name	Description	Dimensions
F	_FACTLOAD_	factor loading matrix	$p \times n$
P	_FACTFCOR_	factor covariance matrix	$n \times n$
U	_FACTERRV_	error covariance matrix	$p \times p$
a	_FACTINTE_	intercepts	$p \times 1$
v	_FACTMEAN_	factor means	$n \times 1$

Specification of the Exploratory Factor Model

Because all initial model matrices of exploratory FACTOR models are predefined in PROC TCALIS, you do not need to specify any other parameters in the model matrices. To obtain desired factor solutions, you can use various options for exploratory factor analysis in the FACTOR statement. These options are the *EFA_options* in the FACTOR statement. Two main types of *EFA_options* are shown as follows:

- options for factor extraction: COMPONENT, HEYWOOD, N=.
- options for factor rotation: GAMMA=, NORM=, RCONVERGE=, RITER=, ROTATE=, TAU=.

For example, the following statement requests that three factors are extracted, followed by a varimax rotation of the initial factor solution:

```
factor n=3 rotate=varimax;
```

See the FACTOR statement on page 6752 for details about the *EFA_options*.

Specification of the Confirmatory Factor Model

To specify a confirmatory FACTOR model, you specify the factor-variable relationships in the FACTOR statement, the factor variances and error variances in the PVAR statement, the factor covariances in the COV statement, and the means and intercepts in the MEAN statement.

Specification of Factor-Variable Relationships

The *CFA_spec* in the FACTOR statement is for specifying the factor-variables relationships. For example, in the following statement you specify three factors F1, F2, and F3 that are related to different clusters of observed variables V1–V9:

```
factor
   F1  -> V1-V3  = 1. parm1 (.4) parm2 (.4),
   F2  -> V4-V6  = 1. parm3 parm4,
   F3  -> V7-V9  = 1. parm5 parm6 (.3);
```

In the specification, variable V1 has a fixed loading of 1.0 on F1. Variables V2 and V3 have loadings on F1 also. These two loadings are free parameters named parm1 and parm2, respectively. Initial estimates can be set in parentheses after the free parameters. For example, both parm1 and parm2 have initial values at 0.4. Similarly, relationships of factor F2 with V4–V6, and of factor F3 with V7–V9 are defined in the same FACTOR statement. Providing initial estimates for parameters is optional. In this example, parm3, parm4, and parm5 are all free parameters without initial values provided. PROC TCALIS can determine appropriate initial estimates for these parameters. See the descriptions about *CFA_spec* in the FACTOR statement on page 6752 for more details about the syntax.

Specification of Factor Variances and Error Variances

You can specify the factor variances and error variances in the PVAR statement. For example, consider the following statement:

```
pvar F1–F3  = fvar1–fvar3,
     V1–V9  = evar1–evar9 (9*10.);
```

In the PVAR statement, you specify the variances of factors F1, F2, and F3 as free parameters fvar1, fvar2, and fvar3, respectively; and the error variances for manifest variables V1–V9 as free parameters evar1–evar9, respectively. Each of the error variance parameters is given a starting value at 10. See the PVAR statement on page 6803 for more details about the syntax.

Specification of Factor Covariances

You can specify the factor covariances in the COV statement. For example, you specify the covariances among factors F1, F2, and F3 in the following statement:

```
cov F1 F2  = cov12,
    F1 F3  = cov13,
    F2 F3  = cov23;
```

The covariance parameters are named cov12, cov13, and cov23, respectively. They represent the lower triangular elements of the factor covariance matrix P. See the COV statement on page 6802 for more details about the syntax.

Specification of Means and Intercepts

If mean structures are of interest, you can also specify the factor means and the intercepts for the manifest variables in the MEAN statement. For example, consider the following statement:

```
mean F1–F3 = fmean1–fmean3,
     V1–V9 = 9*12.;
```

In this statement, you specify the factor means of F1, F2, and F3 as free parameters fmean1, fmean2, and fmean3, respectively; and the intercepts for variables V1–V9 as fixed parameters at 12. See the MEAN statement on page 6788 for more details about the syntax.

Default Parameters in the FACTOR Model

In the initial exploratory FACTOR model, all fixed and free parameters of the model are prescribed or set by default. These prescribed or default parameters include a fixed pattern for the factor loading matrix F, a diagonal pattern for the error variance matrix U, and an identity matrix for factor covariance matrix P. See the section "Exploratory Factor Analysis Models" on page 6865 for more details about the patterns of the exploratory FACTOR model.

In the confirmatory FACTOR model, the error covariance matrix U is restricted to be a diagonal matrix. In other words, covariances of errors are restricted to zeros. Other than that, you can specify any other parameter locations in the all model matrices. These includes all the parameter locations

in the factor loading matrix F, the factor covariance matrix P, the diagonal of the error covariance matrix U, the factor mean vector v, and the intercept vector a. If any of these parameter locations are not specified, default parameters are applied. There are two types of default parameters: one is automatic free parameter, and the other is fixed zero.

Automatic Free Parameters

The set of automatic free parameters in the confirmatory FACTOR model are:

- the variances of factors; that is, the diagonal elements of the factor covariance matrix P
- the error variances for the manifest variables; that is, the diagonal elements of the error covariance matrix U

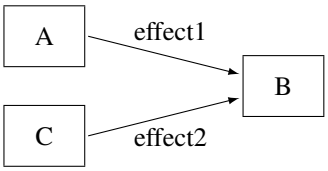
Each automatic free parameter is named with prefix _Add and appended with a unique integer. The reason for automatic parameter generation is to safeguard a proper FACTOR model specification. See the section "Rationale of the Default Parameters in the LINEQS Model" on page 6846 for a more detailed explanation.

Default Fixed Zeros

All unspecified parameter locations that are neither model-imposed specifications nor automatic free parameters in the confirmatory FACTOR model matrices are fixed zeros by default.

The PATH Model

The PATH modeling language is supported in PROC TCALIS as a more intuitive modeling tool. It is designed so that specification by using the PATH modeling language translates effortlessly from the path diagram. For example, given the following simple path diagram:



you can use the PATH statement to specify the paths easily:

```
path    A -> B   effect1,
        C -> B   effect2;
```

In the first entry of the PATH statement, the path A –> B is specified. The associated path coefficient or effect is named effect1. Similarly, in the second entry, the C –> B path is specified with effect2 as the associated effect parameter. In addition to the path coefficients or effects in the path diagram,

you can also specify other types of parameters by using the PVAR and PCOV statements. See the section "A Structural Equation Example" on page 6704 for a more detailed example of the PATH model specification.

Despite its simple representation of the path diagram, the PATH modeling language is general enough to handle a wide class of structural models that can also be handled by other general modeling languages such as LINEQS, LISMOD, or RAM. For brevity, models specified by the PATH modeling language are called PATH models.

Types of Variables in the PATH Model

When you specify the paths in the PATH model, you typically use arrows (such as <– or –>) to denote "causal" paths. For example, in the preceding path diagram or the PATH statement, you specify that B is an outcome variable with predictors A and C, respectively in two paths. An outcome variable is the variable being pointed to in a path specification, while the predictor variable is the one where the arrow starts from.

Whereas the outcome–predictor relationship describes the roles of variables in each single path, the endogenous–exogenous relationship describes the roles of variables in the entire system of paths. In a system of path specification, a variable is endogenous if it is pointed to by at least one arrow or it serves as an outcome variable in a path at least once. Otherwise, it is exogenous. In the preceding path diagram, for example, variable B is endogenous and both variables A and C are exogenous. Note that although any variable that serves as an outcome variable at least in one path must be endogenous, it does not mean that all endogenous variables must serve only as outcome variables in all paths. An endogenous variable in a model might also serve as a predictor variable in a path. For example, variable B in the following PATH statement is an endogenous variable, and it serves as an outcome variable the first path but as a predictor variable in the second path.

```
path    A -> B   effect1,
        B -> C   effect2;
```

A variable is a manifest or observed variable in the PATH model if it is measured and exists in the input data set. Otherwise, it is a latent variable. Because error variables are not explicitly defined in the PATH modeling language, all latent variables that are named in the PATH model are *factors*, which are considered to be the systematic source of effects in the model. Any manifest variable in the PATH model can be endogenous or exogenous. This same is true for any latent factor in the PATH model.

Because you do not name error variables in the PATH model, you do not need to specify paths from errors to any endogenous variables. Error terms are implicitly assumed for all endogenous variables in the PATH model. If error variables are not named, how can one specify the error variance parameters? In the PATH model, error variances are expressed equivalently as partial variances for the associated endogenous variables. These partial variances are set by default in the PATH modeling language. Therefore, you do not need to specify error variance parameters explicitly unless constraints on these parameters are required in the model. You can use the PVAR statement to specify the error variance or partial variance parameters explicitly.

Naming Variables in the PATH Model

Manifest variables in the PATH model are referenced in the input data set. Their names must not be longer than 32 characters. There is no further restrictions beyond those required by the SAS system. You use the names of manifest variables directly in the PATH model specification.

Because you do not name error variables in the PATH model, all latent variables named in the PATH model specification are factors (non-errors). Factor names in the PATH model must not be longer than 32 characters, and they should be different from the manifest variables. Unlike the LINEQS model, you do not need to use any specific prefix for the latent factor names in the PATH model.

As a general naming convention, you should not use Intercept as either a manifest or latent variable name.

Specification of the PATH Model

(1) Specification of Effects or Paths

You specify the "causal" paths or linear functional relationships among variables in the PATH statement. For example, in your model there is a path from v2 to v1 and the effect parameter is named parm1 with a starting value at 0.5, you can use either of these specifications:

```
path     v1 <-  v2   parm1  (0.5);
path     v2 ->  v1   parm1  (0.5);
```

If you have more than one path in your model, path specifications should be separated by commas, as shown in the following PATH statement:

```
path
   v1 <-  v2   parm1  (0.5),
   v2 <-  v3   parm2  (0.3);
```

Because the PATH statement can be used only once in each model specification, all paths in the model must be specified together in a single PATH statement. See the PATH statement on page 6799 for more details about the syntax.

(2) Specification of Variances and Partial (Error) Variances

If v2 is an exogenous variable in the PATH model and you want to specify its variance as a parameter named parm2 with a starting value at 10, you can use the following PVAR statement specification:

```
pvar     v2  = parm2 (10.);
```

If v1 is an endogenous variable in the PATH model and you want to specify its partial variance or error variance as a parameter named parm3 with a starting value at 5.0, you can also use the following PVAR statement specification:

```
pvar     v1 = parm3 (5.0);
```

Therefore, the PVAR statement can be used for both exogenous and endogenous variables. When a variable in the statement is exogenous (which can be automatically determined by PROC TCALIS), you are specifying the variance parameter of the variable. Otherwise, you are specifying the partial or error variance for an endogenous variable.

If you have more than one variance or partial variance parameters to specify in your model, you can put a variable list on the left-hand side of the equal sign, and a parameter list on the right-hand side, as shown in the following PVAR statement specification:

```
pvar
   v1 v2 v3 = parm1 (0.5) parm2 parm3;
```

In the specification, variance or partial variance parameters for variables v1–v3 are parm1, parm2, and parm3, respectively. Only parm1 is given an initial value at 0.5. The initial values for other parameters are generated by PROC TCALIS.

You can also separate the specifications into several entries in the PVAR statement. Entries should be separated by commas. For example, the preceding specification is equivalent to the following specification:

```
pvar
   v1 = parm1 (0.5),
   v2 = parm2,
   v3 = parm3;
```

Because the PVAR statement can be used only once in each model specification, all variance and partial variance parameters in the model must be specified together in a single PVAR statement. See the PVAR statement on page 6803 for more details about the syntax.

(3) Specification of Covariances and Partial Covariances

If you want to specify (partial) covariance between two variables v3 and v4 as a parameter named parm4 with a starting value at 3, you can use the following PCOV statement specification:

```
pcov  v3  v4 = parm4 (5.);
```

Whether parm4 is a covariance or partial covariance parameter depends on the variables types of v3 and v4. If both v3 and v4 are exogenous variables (manifest or latent), parm4 is a covariance parameter between v3 and v4. If both v3 and v4 are endogenous variables (manifest or latent), parm4 is a parameter for the covariance between the errors for v3 and v4. In other words, it is a partial covariance or error covariance parameter for v3 and v4.

A less common case is when one of the variables is exogenous and the other is endogenous. In this case, parm4 is a parameter for the partial covariance between the endogenous variable and the exogenous variable, or the covariance between the error for the endogenous variable and the exogenous variable. Fortunately, such covariances are relatively uncommon in statistical modeling. Their uses confuse the roles of systematic and unsystematic sources in the model and lead to difficulties in interpretations. Therefore, you should almost always avoid this kind of partial covariances.

Like the syntax of the PVAR statement, you can specify a list of (partial) covariance parameters in the PCOV statement. For example, consider the following statement:

```
pcov
   v1 v2 = parm4,
   v1 v3 = parm5,
   v2 v3 = parm6;
```

In the specification, three (partial) covariance parameters parm4, parm5, and parm6 are specified, respectively, for the variable pairs (v1,v2), (v1,v3), and (v2,v3). Entries for (partial) covariance specification are separated by commas.

Because the PCOV statement can only be used once in each model specification, all covariance and partial covariance parameters in the model must be specified together in a single PCOV statement. See the PCOV statement on page 6802 for more details about the syntax.

(4) Specification of Means and Intercepts

Means and intercepts are specified when the mean structures of the model are of interest. You can specify mean and intercept parameters in the MEAN statement. For example, consider the following statement:

```
mean     V5 = parm5   (11.);
```

If V5 is an exogenous variable (which is determined by PROC TCALIS automatically), you are specifying parm5 as the mean parameter of V5. If V5 is an endogenous variable, you are specifying parm5 as the intercept parameter for V5.

Because each named variable in the PATH model is either exogenous or endogenous (exclusively), each variable in the PATH model would have either a mean or an intercept parameter (but not both) to specify in the MEAN statement. Like the syntax of the PVAR statement, you can specify a list of mean or intercept parameters in the MEAN statement. For example, in the following statement you specify a list of mean or intercept parameters for variables v1-v4:

```
mean
   v1-v4 = parm6-parm9;
```

This specification is equivalent to the following specification with four entries of parameter specifications:

```
mean
   v1 = parm6,
   v2 = parm7,
   v3 = parm8,
   v4 = parm9;
```

Again, entries in the MEAN statement must be separated by commas, as shown in the preceding statement.

Because the MEAN statement can only be used once in each model specification, all mean and intercept parameters in the model must be specified together in a single MEAN statement. See the MEAN statement on page 6788 for more details about the syntax.

Specifying Parameters without Initial Values

If you do not have any knowledge about the initial value for a parameter, you can omit the initial value specification and let PROC TCALIS compute it. For example, you can just provide the parameter locations and parameter names as in the following specification:

```
path    v1 <- v2   parm1;
   pvar v2 = parm2,
        v1 = parm3;
```

Specifying Fixed Parameter Values

If you want to specify a fixed parameter value, you do not need to provide a parameter name. Instead, you provide the fixed value (without parentheses) in the specification.

For example, the path coefficient for the path is fixed at 1.0 and the (partial) variance of F1 is also fixed at 1.0.

```
path    v1 <- F1  1.;
   pvar
        F1 = 1.;
```

A Complete PATH Model Specification Example

To show a more complete PATH model specification, the RAM model example in the section "A Complete RAM Model Specification Example" on page 6861 is translated into the PATH model specification in the following statements:

```
path     v1 <-  v2   parm1  (.5),
         v1 <-  v3   parm2      ;
   pvar  v1 = errv   (1.),
         v2 = parm3  (10.),
         v3 = parm4  (10.);
   pcov  v3 v2 =  parm5  (5.);
```

The PATH model specification is very much the same as the RAM model specification. All paths specified in the RAM model are translated into specification in the PATH statement. All other specification are translated into the PVAR, PCOV, and MEAN statements.

Default Parameters in the PATH Model

The treatment of restricted and default parameters in the PATH model is essentially the same as that of the RAM model (see the section "Default Parameters in the RAM Model" on page 6862).

The PATH model does not allow the specification of the effect of any variable on itself. In other words, it is invalid to specify the following:

```
path     v1  <-  v1  parm;
```

The coefficient for such a path is always zero, meaning that the path should not exist in any PATH model. Other than this restriction, any other parameters supported by the PATH model can be specified in the PATH, PVAR, PCOV, and the MEAN statements. If a parameter location in the PATH model is not specified, a default parameter will be applied to that location. There are two types of default parameters: one is automatic free parameters, and the other is fixed zeros.

Automatic Free Parameters

The set of automatic free parameters in the PATH model is derived essentially in the same way as in the RAM or LINEQS model. That is, automatic free parameters of the PATH model include:

- the variances or partial (or error) variances of all variables, manifest or latent. Consequently, all possible variance or partial variance parameters that can be specified in the PVAR statement are automatic free parameters unless explicitly specified otherwise.
- the means of exogenous manifest variables when the mean structures are modeled. That is, all mean parameters pertaining to exogenous manifest variables that can be specified in the MEAN statement are automatic free parameters unless explicitly specified otherwise.
- the covariances among all exogenous manifest latent variables. That is, all covariance parameters pertaining to all possible pairs of exogenous manifest variables that can be specified in the PCOV statement are automatic free parameters unless explicitly specified otherwise.

The reason for automatic parameter generation is to safeguard a proper PATH model specification. See the section "Rationale of the Default Parameters in the LINEQS Model" on page 6846 for a more detailed explanation.

An automatic parameter name is generated by PROC TCALIS for each of these automatic mean, intercept, variance, partial variance, covariance, and partial covariance parameters. Each automatic parameter name is prefixed with _Add and appended with a unique integer.

Default Fixed Zeros

All unspecified parameter locations that are neither set by model-restricted values nor generated with automatic free parameters in the PATH model are fixed zeros by default.

Relating the PATH Model to the RAM Model

Mathematically, the PATH model is essentially the RAM model. You can consider the PATH model to share exactly the same set of model matrices as in the RAM model. See the section "Model Matrices in the RAM Model" on page 6856 and the section "Summary of Matrices and Submatrices in the RAM Model" on page 6859 for details about the RAM model matrices. In the RAM model, the A matrix contains effects or path coefficients for describing relationships among variables. In the PATH model, you specify these effect or coefficient parameters in the PATH statement. The P matrix in the RAM model contains (partial) variance and (partial) covariance parameters. In the PATH model, you use the PVAR and PCOV statements to specify these parameters. The W vector

in the RAM model contains the mean and intercept parameters, while in the PATH model you use the MEAN statement to specify these parameters. By using these model matrices in the PATH model, the covariance and mean structures are derived in the same way as that of the RAM model. See the section "The RAM Model" on page 6855 for derivations of the model structures.

Because the mathematical model behind the PATH and the RAM modeling languages are essentially the same, it is no wonder that the PATH and the RAM syntax for model specification resemble to each other. That is, all path specifications in the PATH statement translate into the PATH *list-entries* of the RAM statement. All PVAR, PCOV, and MEAN specifications of the PATH model translate into the PVAR, PCOV, and MEAN (or INTERCEPT) *list-entries*, respectively, in the RAM statement.

The MSTRUCT Model

In contrast to other modeling languages where the mean and covariance structures are implied from the specification of equations, paths, variable-factor relations, mean parameters, variance parameters, or covariance parameters, the MSTRUCT modeling language is supported in PROC TCALIS for modeling mean and covariance structures directly.

A simple example for using the MSTRUCT modeling language is the testing of a covariance model with equal variances and covariances. Suppose that a variable was measured five times in an experiment. The covariance matrix of these five measurements is hypothesized to have the following structure:

$$\Sigma = \Sigma(\theta)$$

where

$$\theta = (\phi, \tau)$$

and

$$\Sigma(\theta) = \begin{pmatrix} \phi & \tau & \tau & \tau & \tau \\ \tau & \phi & \tau & \tau & \tau \\ \tau & \tau & \phi & \tau & \tau \\ \tau & \tau & \tau & \phi & \tau \\ \tau & \tau & \tau & \tau & \phi \end{pmatrix}$$

For model structures that are hypothesized directly on the covariance matrix, the MSTRUCT modeling language is the most convenient to use. You can also use other general modeling languages such as LINEQS, PATH, or RAM to fit the same model structures, but the specification is less straightforward and more error-prone. For convenience, models specified using the MSTRUCT modeling language are called MSTRUCT models.

Model Matrices in the MSTRUCT Model

Suppose that there are p observed variables. The two model matrices, their names, their roles, and their dimensions are summarized in the following table:

Matrix	Name	Description	Dimensions
Σ	_COV_ or _MSTRUCTCOV_	structured covariance matrix	$p \times p$
μ	_MEAN_ or _MSTRUCTMEAN_	structured mean vector	$p \times 1$

Specification of the MSTRUCT Model

Specifying Variables

In the MSTRUCT statement, you specify the list of p manifest variables of interest in the VAR= list. For example, you specify v1–v5 as the variables analyzed in your MSTRUCT model by this statement:

```
mstruct VAR= v1 v2 v3 v4 v5;
```

See the MSTRUCT statement on page 6792 for details about the syntax.

The manifest variables in the VAR= list must be referenced in the input set. The number of variables in the VAR= list determines the dimensions of the _COV_ and the _MEAN_ matrices in the model. In addition, the order of variables determines the order of row and column variables in the model matrices.

Specifying Parameters in Model Matrices

Denote the parameter vector in the MSTRUCT model as θ. The dimension of θ depends on your hypothesized model. In the preceding example, θ contains two parameters in ϕ and τ. You can use the MATRIX statement to specify these parameters in the _COV_ matrix:

```
matrix _COV_ [1,1] = 5*phi,  /* phi for all diagonal elements */
             [2, ] = tau,    /* tau for all off-diagonal elements */
             [3, ] = 2*tau,
             [4, ] = 3*tau,
             [5, ] = 4*tau;
```

In this MATRIX statement, the five diagonal elements, starting from the [1,1] element of the covariance matrix, are fitted by the phi parameter. The specification 5*phi is a short-hand for specifying phi five times, respectively for each of the five diagonal elements in the covariance matrix. All other lower triangular elements are fitted by the tau parameter, as shown in the MATRIX statement. For example, with [3,] the elements starting from the first element of the third row of the _COV_ matrix are parameterized by the tau parameter. The specification 2*tau repeats the specification two times, meaning that the [3,1] and [3,2] elements are both fitted by the same parameter tau. Similarly, all lower triangular elements (not including the diagonal elements) of the _COV_ matrix are fitted by the tau parameter. The specification of the upper triangular elements (diagonal excluded) of the _COV_ matrix is not needed because the _COV_ matrix is symmetric. The specification in

the lower triangular elements is transferred automatically to the upper triangular elements. See the MATRIX statement on page 6784 for details about the syntax.

Default Parameters in the MSTRUCT Model

You can specify either fixed values or parameters (with or without initial values) for the elements in the _COV_ and _MEAN_ model matrices. If some elements are not specified, default parameters are applied. There are two types of default parameters: one is automatic free parameters, and the other is fixed zeros. They are applied in different situations.

Automatic Free Parameters

If no MATRIX statements are used to specify parameters in your model matrices, automatic free parameters are generated for all elements in the _COV_ matrix. For example, if you do not use the MATRIX statement to specify any element in the _COV_ matrix, free parameters are generated for all variances and covariances in the _COV_ matrix in the model. Similarly, free parameters are generated for the _MEAN_ vector when the mean structures are invoked by the MEANSTR option in the PROC TCALIS or MODEL statement. Each automatic free parameter is named with a prefix _Add and appended with a unique integer.

Default Fixed Zeros

If you specify **at least one** parameter (fixed or free) for an element in a model matrix using the MATRIX statement, no automatic free parameters will be generated for that matrix. All unspecified locations in the _COV_ and _MEAN_ model matrices are fixed zeros by default.

How and Why the Default Parameters Are Treated Differently in the MSTRUCT Model

Notice that the default parameter treatment in the MSTRUCT model is different from other types of models. For example, in the LINEQS model variances of and covariances among exogenous manifest variables are all automatic free parameters. These automatic free parameters are generated by PROC TCALIS. In the MSTRUCT model, however, automatic free parameters are either generated for all elements in a model matrix (when nothing is specified for the matrix or when the MATRIX statement is not used for the MSTRUCT model) or not generated at all (when at least one element in the _COV_ matrix is specified).

The reason is that there is no distinction between exogenous (independent) and endogenous (dependent) variables in the MSTRUCT model. Automatic free parameters for variances of, and covariances among, exogenous manifest variables are thus irrelevant.

Because of the particular default parameter treatment, when fitting an MSTRUCT model you must make sure that each diagonal element in your _COV_ matrix is set as a free, constrained, or fixed parameter, in accordance with your theoretical model. If you specify some elements in the model matrix but omit the specification of some diagonal elements, the default zero variances would lead to a nonpositive definite _COV_ model matrix, making the model fitting problematic.

Naming Variables and Parameters

Follow these rules when you name your variables:

- The usual naming conventions of the SAS system.
- Variable names are not more than 32 characters.
- When you create latent variable names, make sure that they are not used in the input data set that is being analyzed.
- For the LINEQS model, error or disturbance variables must start with 'E', 'e', 'D', or 'd.'
- For the LINEQS model, non-error-type latent variables (that is, factors) must start with 'F' or 'f.'
- For modeling languages other than LINEQS, names for errors or disturbances are not needed. As a result, you do not need to distinguish latent factors from errors or disturbances by using particular prefixes. Variable names that are not referenced in the analyzed data set are supposed to be latent factors.
- You should not use Intercept (case-insensitive) as a variable name in your data set or as a latent variable name in your model.

Follow these rules when you name your parameters:

- The usual naming conventions of the SAS system.
- Parameter names are not more than 32 characters.
- Use a prefix-name when you want to generate new parameter names automatically. A prefix-name contains a short string of characters called "root," followed by double underscores '__'. Each occurrence of such a prefix-name generates a new parameter name by replacing the two trailing underscores with a unique integer. For example, occurrences of Gen__ generate Gen1, Gen2, . . . etc.
- A special prefix-name is the one without a root—that is, it contains only double underscores '__'. Occurrences of '__' generate _Parm1, _Parm2, and so on.
- Automatic parameter names are generated by PROC TCALIS for automatic free parameters in the model. The purpose of adding these parameters is to safeguard ill-defined models. Automatic parameter names start with the prefix _Add and appended with unique integers. For example, _Add1, _Add2, and so on.
- Avoid using parameter names that start with either _, _Add, or _Parm. These names might get confused with the generated names by PROC TCALIS. The confusions might lead to unintended constraints to your model if the parameter names you use match those generated by PROC TCALIS exactly.

- Avoid using parameter names that are roots of prefix-names. For example, you should not use Gen as a parameter name if Gen__ is also used in the same model specification. Although violation of this rule might not distort the model specification, it might cause ambiguities and confusion.

Finally, in PROC TCALIS parameter names and variable names are not distinguished by explicit declarations. That is, a valid SAS name can be used as a parameter name or a variable name in any model supported by PROC TCALIS. Whether a name in a model specification is for a parameter or a variable is determined by the syntactic structure. For example, consider the following path specification:

```
proc tcalis;
   path
      a -> b  c;
   run;
```

PROC TCALIS parses the path specification according to the syntactic structure of the PATH statement and determines that a and b are variable names and c is a parameter name. Consider another specification as follows:

```
proc tcalis;
   path
      a -> b  b;
   run;
```

This is a syntactically correct specification. Variables a and b are defined in a path relationship with a path coefficient parameter also named b. While such a name conflict between parameter and variable names would not confuse PROC TCALIS in terms of model specification and fitting, it would create unnecessary confusions in programming and result interpretations. Hence, using parameter names that match variable names exactly is a bad practice and should be avoided.

Setting Constraints on Parameters

The TCALIS procedure offers a very flexible way to constrain parameters. There are two main methods for specifying constraints. One is explicit specification by using specially designed statements for constraints. The other is implicit specification by using the SAS programming statements.

Explicit Specification of Constraints

Explicit constraints can be set in the following ways:

- specifying boundary constraints on independent parameters in the BOUNDS statement
- specifying general linear equality and inequality constraints on independent parameters in the LINCON statement
- specifying general nonlinear equality and inequality constraints on parametric functions in the NLINCON statement

BOUNDS Statement

You can specify one-sided or two-sided boundaries on independent parameters in the BOUNDS statement. For example, in the following statement you constrain parameter var1 to be nonnegative and parameter effect to be between 0 and 5.

```
bounds   var1 >= 0,
         0. <= effect <= 5.;
```

Note that if your upper and lower bounds are the same for a parameter, it effectively sets a fixed value for that parameter. As a result, PROC TCALIS will reduce the number of independent parameters by one automatically. Note also that only independent parameters are allowed to be bounded in the BOUNDS statement.

LINCON Statement

You can specify equality or inequality linear constraints on independent parameters in the LINCON statement. For example, in the following statement you specify a linear inequality constraint on parameters beta1, beta2, and beta3 and an equality constraint on parameters gamma1 and gamma2.

```
lincon   beta1 - .5 * beta2 - .5 * beta3  >= 0.,
         gamma1 - gamma2 = 0.;
```

In the inequality constraint, beta1 is set to be at least as large as the average of beta2 and beta3. In the equality constraint, gamma1 and gamma2 are set to be equal. Note that in PROC TCALIS a nonredundant linear equality constraint on independent parameters effectively reduces the number of parameters by one.

NLINCON Statement

You can specify equality or inequality nonlinear constraints for parameters in the NLINCON statement. While you can only constrain the independent parameters in the BOUNDS and the LINCON statements, you can constrain any of the following in the NLINCON statement:

- independent parameters
- dependent parameters
- parametric functions computed by the SAS programming statements

For example, consider the following statements:

```
nlincon
    IndParm >= 0,           /* constraint 1 */
    0 <= DepParm <= 10,     /* constraint 2 */
    ParmFunc1 >= 3,         /* constraint 3 */
    0 <=  ParmFunc2 <= 8;   /* constraint 4 */

/* SAS Programming statements in the following */
DepParm   = Indparm1 + IndParm5;
ParmFunc1 = IndParm1 - .5 * IndParm2 - .5 * IndParm3;
ParmFunc2 = (IndParm1 - 7.)**2 + SQRT(DepParm * IndParm4) * ParmFunc1;
```

You specify four nonlinear constraints by using the NLINCON statement. Labeled in a comment as "constraint 1" is a one-sided boundary constraint for independent parameter IndParm. Labeled in a comment as "constraint 2" is a two-sided boundary constraint for dependent parameter DepParm. Labeled in a comment as "constraint 3" is a one-sided inequality constraint on parametric function named ParmFunc1. Finally, labeled in a comment as "constraint 4" is a two-sided inequality constraint on parametric function named ParmFunc2. Parametric functions ParmFunc1 and ParmFunc2 are defined and computed in the SAS programming statements after the NLINCON statement specification.

Constraint 1 could have been set in the BOUNDS statement because it is just a simple boundary constraint on an independent parameter. Constraint 3 could have been set in the LINCON statement because the definition of ParmFunc1 in a SAS programming statement shows that it is a linear function of independent parameters. The purpose of including these special cases of "nonlinear constraints" in this example is to show the flexibility of the NLINCON statement. However, whenever possible, the BOUNDS or the LINCON statement specification should be considered first because computationally they are more efficient than the equivalent specification in the NLINCON statement.

Specification in the NLINCON statement becomes necessary when you want to constrain dependent parameters or nonlinear parametric functions. For example, constraint 2 is a two-sided boundary constraint on the dependent parameter DepParm, which is defined as a linear function of independent parameters in a SAS programming statement. Constraints on dependent parameters are not allowed in the BOUNDS statement. Constraint 4 is a two-sided inequality constraint on the nonlinear parametric function ParmFunc2, which is defined as a nonlinear function of other parametric functions and parameters in the SAS programming statements. Again, you cannot use the LINCON statement to specify nonlinear constraints.

Implicit Constraint Specification

An implicit way to specify constraints is to use your own SAS programming statements together with the PARAMETERS statement to express special properties of the parameter estimates. This kind of reparameterization tool is also present in McDonald's COSAN implementation (McDonald 1978) but is considerably easier to use in the TCALIS procedure. PROC TCALIS is able to compute analytic first- and second-order derivatives that you would have to specify using the COSAN program.

Some traditional ways to enforce parameter constraints by using reparameterization or parameter transformation (McDonald 1980) are considered in the following:

- **one-sided boundary constraints of the form:**

  $$q \geq a \qquad \text{or} \qquad q \leq b$$

  where the parameter of interest is q, which should be at least as large as (or at most as small as) a given constant value a (or b). This inequality constraint can be expressed as an equality constraint:

  $$q = a + x^2 \qquad \text{or} \qquad q = b - x^2$$

  in which the new parameter x is unconstrained.

  For example, inequality constraint $q \geq 7$ can be accomplished by the following statements:

  ```
  parameters  x (0.);
  q = 7 + x * x;
  ```

  In this specification, you essentially redefine q as a parametric function of x, which is not constrained and has a starting value at 0.

- **two-sided boundary constraints of the form:**

  $$a \leq q \leq b$$

  where the parameter of interest is q, which should be located between two given constant values a and b, with $a < b$. This inequality constraint can be expressed as the following equality constraint:

  $$q = a + (b - a)\frac{\exp(x)}{1 + \exp(x)}$$

  where the new parameter x is unconstrained.

  For example, to implement $1 \leq q \leq 5$ in PROC TCALIS, you can use the following statements:

  ```
  parameters x (0.);
  u = exp(x);
  q = 1 + 4 * u / (1 + u);
  ```

In this specification, q becomes a dependent parameter which is nonlinearly related to independent parameter x, which is an independent parameter defined in the PARAMETERS statement with a starting value of 0.

- **one-sided order constraints of the form:**

$$q_1 \leq q_2, \quad q_1 \leq q_3, \quad \ldots, \quad q_1 \leq q_k$$

where $q_1, \ldots, q_k$ are the parameters of interest. These inequality constraints can be expressed as the following set of equality constraints:

$$q_1 = x_1, \quad q_2 = x_1 + x_2^2, \quad \ldots, \quad q_k = x_1 + x_k^2$$

where the new parameters $x_1, \ldots, x_k$ are unconstrained.

For example, to implement $q_1 \leq q_2$, $q_1 \leq q_3$, and $q_1 \leq q_4$ simultaneously, you can use the following statements:

```
parameters x1-x4 (4*0.);
q1 = x1;
q2 = x1 + x2 * x2;
q3 = x1 + x3 * x3;
q4 = x1 + x4 * x4;
```

In this specification, you essentially redefine q_1–q_4 as dependent parameters that are functions of x_1–x_4, which are defined as independent parameters in the PARAMETERS statement with starting values of zeros. No constraints on x_i's are needed. The way that q_i's are computed in the SAS programming statements guarantees the required order constraints on q_i's are satisfied.

- **two-sided order constraints of the form:**

$$q_1 \leq q_2 \leq q_3 \leq \ldots \leq q_k$$

These inequality constraints can be expressed as the following set of equality constraints:

$$q_1 = x_1, \quad q_2 = q_1 + x_2^2, \quad \ldots \quad q_k = q_{k-1} + x_k^2$$

where the new parameters $x_1, \ldots, x_k$ are unconstrained.

For example, to implement $q_1 \leq q_2 \leq q_3 \leq q_4$ simultaneously, you can use the following statements:

```
parameters x1-x4 (4*0.);
q1 = x1;
q2 = q1 + x2 * x2;
q3 = q2 + x3 * x3;
q4 = q3 + x4 * x4;
```

In this specification, you redefine q_1–q_4 as dependent parameters that are functions of x_1–x_4, which are defined as independent parameters in the PARAMETERS statement. Each x_i has a starting value of zero without being constrained in estimation. The order relation of q_i's are satisfied by the way they are computed in the SAS programming statements.

- **linear equation constraints of the form:**

$$\sum_{i}^{k} b_i q_i = a$$

where q_i's are the parameters of interest, b_i's are constant coefficients, a is a constant, and k is an integer greater than one. This linear equation can be expressed as the following system of equations with unconstrained new parameters $x_1, x_2, \ldots, x_k$:

$$\begin{aligned} q_i &= x_i / b_i \qquad (i < k) \\ q_k &= (a - \sum_{j}^{k-1} x_j)/b_k \end{aligned}$$

For example, consider the following linear constraint on independent parameters q_1–q_3:

$$3q_1 + 2q_2 - 5q_3 = 8$$

You can use the following statements to implement the linear constraint:

```
parameters x1-x2 (2*0.);
q1 = x1 / 3;
q2 = x2 / 2;
q3 = -(8 - x1 - x2) / 5;
```

In this specification, q_1–q_3 become dependent parameters that are functions of $x1$ and $x2$. The linear constraint on q_1 and q_3 are satisfied by the way they are computed. In addition, after reparameterization the number of independent parameters drops to two.

Refer to McDonald (1980) and Browne (1982) for further notes on reparameterization techniques.

Explicit or Implicit Specification of Constraints?

Explicit and implicit constraint techniques differ in their specifications and lead to different computational steps in optimizing a solution. The explicit constraint specification that uses the supported statements incurs additional computational routines within the optimization steps. In contrast, the implicit reparameterization method does not incur additional routines for evaluating constraints during the optimization. Rather, it changes the constrained problem to a non-constrained one. This costs more in computing function derivatives and in storing parameters.

If the optimization problem is small enough to apply the Levenberg-Marquardt or Newton-Raphson algorithm, use the BOUNDS and the LINCON statements to set explicit boundary and linear constraints. If the problem is so large that you must use a quasi-Newton or conjugate gradient algorithm, reparameterization techniques might be more efficient than the BOUNDS and LINCON statements.

Automatic Variable Selection

When you specify your model, you use the main and subsidiary model statements to define variable relationships and parameters. PROC TCALIS checks the variables mentioned in these statements against the variable list of the input data set. If a variable in your model is also found in your data set, PROC TCALIS knows that it is a manifest variable. Otherwise, it is either a latent variable or an invalid variable.

To save computational resources, only manifest variables defined in your model are automatically selected for analysis. For example, even if you have 100 variables in your input data set, only a covariance matrix of 10 manifest variables is computed for the analysis of the model if only 10 variables are selected for analysis.

In some special circumstances, the automatic variable selection performed for the analysis might be a drawback. For example, if you are interested in modification indices connected to some of the variables that are not used in the model, automatic variable selection in the specification stage will exclude those variables from consideration in computing modification indices. Fortunately, a little trick can be done. You can use the VAR statement to include as many exogenous manifest variables as needed. Any variables in the VAR statement that are defined in the input data set but are not used in the main and subsidiary model specification statements are included in the model as exogenous manifest variables.

For example, the first three steps in a stepwise regression analysis of the Werner Blood Chemistry data (Jöreskog and Sörbom 1988, p. 111) can be performed as follows:

```
proc tcalis data=dixon method=gls nobs=180 print mod;
   var    x1-x7;
   lineqs y = e;
   std    e = var;
run;
proc tcalis data=dixon method=gls nobs=180 print mod;
   var    x1-x7;
   lineqs y = g1 x1 + e;
   std    e = var;
run;
proc tcalis data=dixon method=gls nobs=180 print mod;
   var    x1-x7;
   lineqs y = g1 x1 + g6 x6 + e;
   std    e = var;
run;
```

In the first analysis, no independent manifest variables are included in the regression equation for dependent variable y. However, x1–x7 are specified in the VAR statement so that in computing the Lagrange multiplier tests these variables would be treated as potential predictors in the regression equation for dependent variable y. Similarly, in the next analysis, x1 is already a predictor in the regression equation, while x2–x7 are treated as potential predictors in the LM tests. In the last analysis, x1 and x6 are predictors in the regression equation, while other x-variables are treated as potential predictors in the LM tests.

Estimation Criteria

The following five estimation methods are available in PROC TCALIS:

- unweighted least squares (ULS)
- generalized least squares (GLS)
- normal-theory maximum likelihood (ML)
- weighted least squares (WLS, ADF)
- diagonally weighted least squares (DWLS)

Default weight matrices $\mathbf{W}$ are computed for GLS, WLS, and DWLS estimation. You can also provide your own weight matrices by using an INWGT= data set.

PROC TCALIS does not implement all estimation methods in the field. As mentioned in the section "Overview: TCALIS Procedure" on page 6681, partial least squares (PLS) is not implemented. The PLS method is developed under less restrictive statistical assumptions. It circumvents some computational and theoretical problems encountered by the existing estimation methods in PROC TCALIS; however, PLS estimates are less efficient in general. When the statistical assumptions of PROC TCALIS are tenable (for example, large sample size, correct distributional assumptions, and so on), ML, GLS, or WLS methods yield better estimates than the PLS method. Note that there is a SAS/STAT procedure called PROC PLS that employs the partial least squares technique, but for a different class of models than those of PROC TCALIS. For example, in a PROC TCALIS model each latent variable is typically associated with only a subset of manifest variables (predictor or outcome variables). However, in PROC PLS latent variables are not prescribed with subsets of manifest variables. Rather, they are extracted from linear combinations of all manifest predictor variables. Therefore, for general path analysis with latent variables you should use PROC TCALIS.

ULS, GLS, and ML Discrepancy functions

In each estimation method, the parameter vector is estimated iteratively by a nonlinear optimization algorithm that minimizes a discrepancy function F, which is also known as the fit function in the literature. With p denoting the number of manifest variables, $\mathbf{S}$ the sample $p \times p$ covariance matrix for a sample with size N, $\bar{\boldsymbol{x}}$ the $p \times 1$ vector of sample means, $\boldsymbol{\Sigma}$ the fitted covariance matrix, and $\boldsymbol{\mu}$ the vector of fitted means, the discrepancy function for unweighted least squares (ULS) estimation is:

$$F_{ULS} = 0.5Tr[(\mathbf{S} - \boldsymbol{\Sigma})^2] + (\bar{\boldsymbol{x}} - \boldsymbol{\mu})'(\bar{\boldsymbol{x}} - \boldsymbol{\mu})$$

The discrepancy function for generalized least squares estimation (GLS) is:

$$F_{GLS} = 0.5Tr[(\mathbf{W}^{-1}(\mathbf{S} - \boldsymbol{\Sigma}))^2] + (\bar{\boldsymbol{x}} - \boldsymbol{\mu})'\mathbf{W}^{-1}(\bar{\boldsymbol{x}} - \boldsymbol{\mu})$$

By default, $\mathbf{W} = \mathbf{S}$ is assumed so that F_{GLS} is the normal theory generalized least squares discrepancy function.

The discrepancy function for normal-theory maximum likelihood estimation (ML) is:

$$F_{ML} = Tr(\mathbf{S}\mathbf{\Sigma}^{-1}) - p + ln(|\mathbf{\Sigma}|) - ln(|\mathbf{S}|) + (\bar{\boldsymbol{x}} - \boldsymbol{\mu})'\mathbf{\Sigma}^{-1}(\bar{\boldsymbol{x}} - \boldsymbol{\mu})$$

In each of the discrepancy functions, $\mathbf{S}$ and $\bar{\boldsymbol{x}}$ are considered to be given and $\mathbf{\Sigma}$ and $\boldsymbol{\mu}$ are functions of model parameter vector $\mathbf{\Theta}$. That is:

$$F = F(\mathbf{\Sigma}(\mathbf{\Theta}), \boldsymbol{\mu}(\mathbf{\Theta}); \mathbf{S}, \bar{\boldsymbol{x}})$$

Estimating $\mathbf{\Theta}$ by using a particular estimation method amounts to choosing a vector $\boldsymbol{\theta}$ that minimizes the corresponding discrepancy function F.

When the mean structures are not modeled or when the mean model is saturated by parameters, the last term of each fit function vanishes. That is, they become:

$$F_{ULS} = 0.5Tr[(\mathbf{S} - \mathbf{\Sigma})^2]$$

$$F_{GLS} = 0.5Tr[(\mathbf{W}^{-1}(\mathbf{S} - \mathbf{\Sigma}))^2]$$

$$F_{ML} = Tr(\mathbf{S}\mathbf{\Sigma}^{-1}) - p + ln(|\mathbf{\Sigma}|) - ln(|\mathbf{S}|)$$

If, instead of being a covariance matrix, $\mathbf{S}$ is a correlation matrix in the discrepancy functions, $\mathbf{\Sigma}$ would naturally be interpreted as the fitted correlation matrix. Although whether $\mathbf{S}$ is a covariance or correlation matrix makes no difference in minimizing the discrepancy functions, correlational analyses that use these functions are problematic because of the following issues:

- The diagonal of the fitted correlation matrix $\mathbf{\Sigma}$ might contain values other than ones, which violates the requirement of being a correlation matrix.
- Whenever available, standard errors computed for correlation analysis in PROC TCALIS are straightforward generalizations of those of covariance analysis. In very limited cases these standard errors are good approximations. However, in general they are not even asymptotically correct.
- The model fit chi-square statistic for correlation analysis might not follow the theoretical distribution, thus making model fit testing difficult.

Despite these issues in correlation analysis, if your primary interest is to obtain the estimates in the correlation models, you might still find PROC TCALIS results for correlation analysis useful.

PROC TCALIS is primarily developed for the analysis of covariance structures, and hence COVARIANCE is the default option in the procedure. Depending on the nature of research, you can add the mean structures in the analysis. However, for the analysis of correlation structures, you can not add the mean structures for modeling in PROC TCALIS.

WLS and ADF Discrepancy Functions

Another important discrepancy function to consider is the weighted least squares (WLS) function. Let $u = (s, \bar{\boldsymbol{x}})$ be a $p(p+3)/2$ vector containing all nonredundant elements in the sample covariance matrix $\mathbf{S}$ and sample mean vector $\bar{\boldsymbol{x}}$, with $s = vecs(\mathbf{S})$ representing the vector of the

$p(p+1)/2$ lower triangle elements of the symmetric matrix $\mathbf{S}$, stacking row by row. Similarly, let $\eta = (\sigma, \mu)$ be a $p(p+3)/2$ vector containing all nonredundant elements in the fitted covariance matrix $\mathbf{\Sigma}$ and the fitted mean vector $\boldsymbol{\mu}$, with $\sigma = vecs(\mathbf{\Sigma})$ representing the vector of the $p(p+1)/2$ lower triangle elements of the symmetric matrix $\mathbf{\Sigma}$.

The WLS discrepancy function is:

$$F_{WLS} = (u-\eta)'\mathbf{W}^{-1}(u-\eta)$$

where $\mathbf{W}$ is a positive definite symmetric weight matrix with $(p(p+3)/2)$ rows and columns. Because η is a function of model parameter vector $\mathbf{\Theta}$ under the structural model, you can write the WLS function as:

$$F_{WLS} = (u-\eta(\mathbf{\Theta}))'\mathbf{W}^{-1}(u-\eta(\mathbf{\Theta}))$$

Suppose that u converges to $\eta_o = (\sigma_o, \mu_o)$ with increasing sample size, where σ_o and μ_o denote the population covariance matrix and mean vector, respectively. By default, the WLS weight matrix $\mathbf{W}$ in PROC TCALIS is computed from the raw data as a consistent estimate of the asymptotic covariance matrix $\mathbf{\Gamma}$ of $\sqrt{N}(u-\eta_o)$, with $\mathbf{\Gamma}$ partitioned as

$$\Gamma = \begin{pmatrix} \mathbf{\Gamma}_{ss} & \mathbf{\Gamma}'_{\bar{x}s} \\ \mathbf{\Gamma}_{\bar{x}s} & \mathbf{\Gamma}_{\bar{x}\bar{x}} \end{pmatrix}$$

where $\mathbf{\Gamma}_{ss}$ denotes the $(p(p+1)/2) \times (p(p+1)/2)$ asymptotic covariance matrix for $\sqrt{N}(s-\sigma_o)$, $\mathbf{\Gamma}_{\bar{x}\bar{x}}$ denotes the $p \times p$ asymptotic covariance matrix for $\sqrt{N}(\bar{x}-\mu_o)$, and $\mathbf{\Gamma}_{\bar{x}s}$ denotes the $p \times (p(p+1)/2)$ asymptotic covariance matrix between $\sqrt{N}(\bar{x}-\mu_o)$ and $\sqrt{N}(s-\sigma_o)$.

To compute the default weight matrix $\mathbf{W}$ as a consistent estimate of $\mathbf{\Gamma}$, define a similar partition of the weight matrix $\mathbf{W}$ as:

$$\mathbf{W} = \begin{pmatrix} \mathbf{W}_{ss} & \mathbf{W}'_{\bar{x}s} \\ \mathbf{W}_{\bar{x}s} & \mathbf{W}_{\bar{x}\bar{x}} \end{pmatrix}$$

Each of the submatrices in the partition can now be computed from the raw data. First, define the biased sample covariance for variables i and j as:

$$\mathbf{t}_{ij} = \frac{1}{N}\sum_{r=1}^{N}(x_{ri}-\bar{x}_i)(x_{rj}-\bar{x}_j)$$

and the sample fourth-order central moment for variables i, j, k, and l as:

$$\mathbf{t}_{ij,kl} = \frac{1}{N}\sum_{r=1}^{N}(x_{ri}-\bar{x}_i)(x_{rj}-\bar{x}_j)(x_{rk}-\bar{x}_k)(x_{rl}-\bar{x}_l)$$

The submatrices in $\mathbf{W}$ are computed by:

$$[\mathbf{W}_{ss}]_{ij,kl} = \mathbf{t}_{ij,kl} - \mathbf{t}_{ij}\mathbf{t}_{kl}$$

$$[\mathbf{W}_{\bar{x}s}]_{i,kl} = \frac{1}{N}\sum_{r=1}^{N}(x_{ri}-\bar{x}_i)(x_{rk}-\bar{x}_k)(x_{rl}-\bar{x}_l)$$

$$[\mathbf{W}_{\bar{x}\bar{x}}]_{ij} = \mathbf{t}_{ij}$$

Assuming the existence of finite eighth-order moments, this default weight matrix $\mathbf{W}$ is a consistent but biased estimator of the asymptotic covariance matrix $\mathbf{\Gamma}$.

By using the ASYCOV option, you can use Browne's (1984, formula (3.8)) unbiased estimator of $\mathbf{\Gamma}_{ss}$ as:

$$\begin{aligned}[\mathbf{W}_{ss}]_{ij,kl} &= \frac{N(N-1)}{(N-2)(N-3)}(\mathbf{t}_{ij,kl} - \mathbf{t}_{ij}\mathbf{t}_{kl}) \\ &\quad -\frac{N}{(N-2)(N-3)}(\mathbf{t}_{ik}\mathbf{t}_{jl} + \mathbf{t}_{il}\mathbf{t}_{jk} - \frac{2}{N-1}\mathbf{t}_{ij}\mathbf{t}_{kl})\end{aligned}$$

There is no guarantee that $\mathbf{W}_{ss}$ computed this way is positive semidefinite. However, the second part is of order $O(N^{-1})$ and does not destroy the positive semidefinite first part for sufficiently large N. For a large number of independent observations, default settings of the weight matrix $\mathbf{W}$ result in asymptotically distribution-free parameter estimates with unbiased standard errors and a correct χ^2 test statistic (Browne 1982, 1984).

With the default weight matrix $\mathbf{W}$ computed by PROC TCALIS, the WLS estimation is also called as the asymptotically distribution-free (ADF) method. In fact, as options in PROC TCALIS, METHOD=WLS and METHOD=ADF are totally equivalent, even though WLS in general might include cases with special weight matrices other than the default weight matrix.

When the mean structures are not modeled, the WLS discrepancy function is still the same quadratic form statistic. However, with only the elements in covariance matrix being modeled, the dimensions of u and η are both reduced to $p(p+1)/2 \times 1$, and the dimension of the weight matrix is now $(p(p+1)/2) \times (p(p+1)/2)$. That is, the WLS discrepancy function for covariance structure models is:

$$F_{WLS} = (s-\sigma)'\mathbf{W}_{ss}^{-1}(s-\sigma)$$

If $\mathbf{S}$ is a correlation rather than a covariance matrix, the default setting of the $\mathbf{W}_{ss}$ is a consistent estimator of the asymptotic covariance matrix $\mathbf{\Gamma}_{ss}$ of $\sqrt{N}(s-\sigma_o)$ (Browne and Shapiro 1986; DeLeeuw 1983), with s and σ_o representing vectors of sample and population correlations, respectively. Elementwise, $\mathbf{W}_{ss}$ is expressed as:

$$\begin{aligned}[\mathbf{W}_{ss}]_{ij,kl} &= r_{ij,kl} - \frac{1}{2}r_{ij}(r_{ii,kl} + r_{jj,kl}) - \frac{1}{2}r_{kl}(r_{kk,ij} + r_{ll,ij}) \\ &\quad +\frac{1}{4}r_{ij}r_{kl}(r_{ii,kk} + r_{ii,ll} + r_{jj,kk} + r_{jj,ll})\end{aligned}$$

where

$$r_{ij} = \frac{\mathbf{t}_{ij}}{\sqrt{\mathbf{t}_{ii}\mathbf{t}_{jj}}}$$

and

$$r_{ij,kl} = \frac{\mathbf{t}_{ij,kl}}{\sqrt{\mathbf{t}_{ii}\mathbf{t}_{jj}\mathbf{t}_{kk}\mathbf{t}_{ll}}}$$

The asymptotic variances of the diagonal elements of a correlation matrix are 0. That is,

$$[\mathbf{W}_{ss}]_{ii,ii} = 0$$

for all i. Therefore, the weight matrix computed this way is always singular. In this case, the discrepancy function for weighted least squares estimation is modified to:

$$\begin{aligned} F_{WLS} &= \sum_{i=2}^{n}\sum_{j=1}^{i-1}\sum_{k=2}^{n}\sum_{l=1}^{k-1}[\mathbf{W}_{ss}]^{ij,kl}([\mathbf{S}]_{ij}-[\mathbf{\Sigma}]_{ij})([\mathbf{S}]_{kl}-[\mathbf{\Sigma}]_{kl}) \\ &\quad +r\sum_{i}^{n}([\mathbf{S}]_{ii}-[\mathbf{\Sigma}]_{ii})^2 \end{aligned}$$

where r is the penalty weight specified by the WPENALTY=r option and the $[\mathbf{W}_{ss}]^{ij,kl}$ are the elements of the inverse of the reduced $(n(n-1)/2)\times(n(n-1)/2)$ weight matrix that contains only the nonzero rows and columns of the full weight matrix $\mathbf{W}_{ss}$.

The second term is a penalty term to fit the diagonal elements of the correlation matrix $\mathbf{S}$. The default value of $r = 100$ can be decreased or increased by the WPENALTY= option. The often used value of $r = 1$ seems to be too small in many cases to fit the diagonal elements of a correlation matrix properly.

Note that when you model correlation structures, no mean structures can be modeled simultaneously in the same model.

DWLS Discrepancy Functions

Storing and inverting the huge weight matrix $\mathbf{W}$ in WLS estimation requires considerable computer resources. A compromise is found by implementing the diagonally weighted least squares (DWLS) method that uses only the diagonal of the weight matrix $\mathbf{W}$ from the WLS estimation in the following discrepancy function:

$$\begin{aligned} F_{DWLS} &= (u-\eta)'[\mathrm{diag}(\mathbf{W})]^{-1}(u-\eta) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{i}[\mathbf{W}_{ss}]^{-1}_{ij,ij}([\mathbf{S}]_{ij}-[\mathbf{\Sigma}]_{ij})^2+\sum_{i=1}^{n}[\mathbf{W}_{\bar{x}\bar{x}}]^{-1}_{ii}(\bar{x}_i-\boldsymbol{\mu}_i)^2 \end{aligned}$$

When only the covariance structures are modeled, the discrepancy function becomes:

$$F_{DWLS} = \sum_{i=1}^{n}\sum_{j=1}^{i}[\mathbf{W}_{ss}]^{-1}_{ij,ij}([\mathbf{S}]_{ij}-[\mathbf{\Sigma}]_{ij})^2$$

For correlation models, the discrepancy function is:

$$F_{DWLS} = \sum_{i=2}^{n}\sum_{j=1}^{i-1}[\mathbf{W}_{ss}]^{-1}_{ij,ij}([\mathbf{S}]_{ij}-[\mathbf{\Sigma}]_{ij})^2+r\sum_{i=1}^{n}([\mathbf{S}]_{ii}-[\mathbf{\Sigma}]_{ii})^2$$

where r is the penalty weight specified by the WPENALTY=r option. Note that no mean structures can be modeled simultaneously with correlation structures when using the DWLS method.

As the statistical properties of DWLS estimates are still not known, standard errors for estimates are not computed for the DWLS method.

Input Weight Matrices

In GLS, WLS, or DWLS estimation you can change from the default settings of weight matrices $\mathbf{W}$ by using an INWGT= data set. The TCALIS procedure requires a positive definite weight matrix that has positive diagonal elements.

Multiple-Group Discrepancy Function

Suppose that there are k independent groups in the analysis and N_1, N_2, ..., N_k are the sample sizes for the groups. The overall discrepancy function $F(\mathbf{\Theta})$ is expressed as a weighted sum of individual discrepancy functions F_i's for the groups:

$$F(\mathbf{\Theta}) = \sum_{i=1}^{k} t_i F_i(\mathbf{\Theta})$$

where

$$t_i = \frac{N_i - 1}{N - k}$$

is the weight of the discrepancy function for group i, and

$$N = \sum_{i=1}^{k} N_i$$

is the total number of observations in all groups. In PROC TCALIS, all discrepancy function F_i's in the overall discrepancy function must belong to the same estimation method. You cannot specify different estimation methods for the groups in a multiple-group analysis. However, you can fit covariance and mean structures to a group but fit covariance structures only to another group.

Relationships among Estimation Criteria

If only the covariance or correlation structures are considered, the five estimation functions, F_{ULS}, F_{GLS}, F_{ML}, F_{WLS}, and F_{DWLS}, belong to the following two groups:

- The functions F_{ULS}, F_{GLS}, and F_{ML} take into account all n^2 elements of the symmetric residual matrix $\mathbf{S} - \mathbf{\Sigma}$. This means that the off-diagonal residuals contribute twice to the discrepancy function F, as lower and as upper triangle elements.

- The functions F_{WLS} and F_{DWLS} take into account only the $n(n+1)/2$ lower triangular elements of the symmetric residual matrix $\mathbf{S}-\mathbf{\Sigma}$. This means that the off-diagonal residuals contribute to the discrepancy function F only once.

The F_{DWLS} function used in PROC TCALIS differs from that used by the LISREL 7 program. Formula (1.25) of the LISREL 7 manual (Jöreskog and Sörbom 1988, p. 23) shows that LISREL groups the F_{DWLS} function in the first group by taking into account all n^2 elements of the symmetric residual matrix $\mathbf{S}-\mathbf{\Sigma}$.

- Relationship between DWLS and WLS:
  PROC TCALIS: The F_{DWLS} and F_{WLS} discrepancy functions deliver the same results for the special case that the weight matrix $\mathbf{W} = \mathbf{W}_{ss}$ used by WLS estimation is a diagonal matrix.
  LISREL 7: This is not the case.

- Relationship between DWLS and ULS:
  LISREL 7: The F_{DWLS} and F_{ULS} estimation functions deliver the same results for the special case that the diagonal weight matrix $\mathbf{W} = \mathbf{W}_{ss}$ used by DWLS estimation is an identity matrix.
  PROC TCALIS: To obtain the same results with F_{DWLS} and F_{ULS} estimation, set the diagonal weight matrix $\mathbf{W} = \mathbf{W}_{ss}$ used in DWLS estimation to:

$$[\mathbf{W}_{ss}]_{ik,ik} = \begin{cases} 1. & \text{if } i = k \\ 0.5 & \text{otherwise} \end{cases} \qquad (k \le i)$$

  Because the reciprocal elements of the weight matrix are used in the discrepancy function, the off-diagonal residuals are weighted by a factor of 2.

Gradient, Hessian, Information Matrix, and Approximate Standard Errors

For a single-sample setting with a discrepancy function $F = F(\mathbf{\Sigma}(\mathbf{\Theta}), \boldsymbol{\mu}(\mathbf{\Theta}); \mathbf{S}, \bar{\boldsymbol{x}})$, the gradient is defined as the first partial derivatives of the discrepancy function with respect to the model parameters $\mathbf{\Theta}$:

$$g(\mathbf{\Theta}) = \frac{\partial}{\partial \mathbf{\Theta}} F(\mathbf{\Theta})$$

The Hessian is defined as the second partial derivatives of the discrepancy function with respect to the model parameters $\mathbf{\Theta}$:

$$H(\mathbf{\Theta}) = \frac{\partial^2}{\partial \mathbf{\Theta} \partial \mathbf{\Theta}'} F(\mathbf{\Theta})$$

Suppose that the mean and covariance structures fit perfectly with $\mathbf{\Theta} = \mathbf{\Theta}_o$ in the population. The information matrix is defined as:

$$I(\mathbf{\Theta}_o) = \frac{1}{2}\mathcal{E}(H(\mathbf{\Theta}_o))$$

where the expectation $\mathcal{E}(\cdot)$ is taken over the sampling space of $\mathbf{S}, \bar{\boldsymbol{x}}$.

The information matrix plays a significant role in statistical theory. Under certain regularity conditions, the inverse of the information matrix $I^{-1}(\mathbf{\Theta}_o)$ is the asymptotic covariance matrix for $\sqrt{N}(\hat{\mathbf{\Theta}} - \mathbf{\Theta}_o)$, where N denotes the sample size and $\hat{\mathbf{\Theta}}$ is an estimator.

In practice, $\mathbf{\Theta}_o$ is never known and can only be estimated. The information matrix is therefore estimated by the so-called empirical information matrix:

$$I(\hat{\mathbf{\Theta}}) = \frac{1}{2}H(\hat{\mathbf{\Theta}})$$

which is evaluated at the values of the sample estimates $\hat{\mathbf{\Theta}}$. Notice that this empirical information matrix, rather than the unknown $I(\mathbf{\Theta}_o)$, is the "information matrix" displayed in PROC TCALIS output.

Taking the inverse of the empirical information matrix with sample size adjustment, PROC TCALIS approximates the estimated covariance matrix of $\hat{\mathbf{\Theta}}$ by:

$$((N-1)I(\hat{\mathbf{\Theta}}))^{-1} = ((N-1)\frac{1}{2}H(\hat{\mathbf{\Theta}}))^{-1} = \frac{2}{N-1}H^{-1}(\hat{\mathbf{\Theta}})$$

Approximate standard errors for $\hat{\mathbf{\Theta}}$ can then be computed as the square roots of the diagonal elements of the estimated covariance matrix. The theory about the empirical information matrix, the approximate covariance matrix of the parameter estimates, and the approximate standard errors applies to all but the ULS and DWLS estimation methods. Standard errors are therefore not computed with the ULS and DWLS estimation methods.

If a given Hessian or information matrix is singular, PROC TCALIS offers two ways to compute a generalized inverse of the matrix and, therefore, two ways to compute approximate standard errors of implicitly constrained parameter estimates, t values, and modification indices. Depending on the G4= specification, either a Moore-Penrose inverse or a G2 inverse is computed. The expensive Moore-Penrose inverse computes an estimate of the null space by using an eigenvalue decomposition. The cheaper G2 inverse is produced by sweeping the linearly independent rows and columns and zeroing out the dependent ones.

Multiple-Group Extensions

In the section "Multiple-Group Discrepancy Function" on page 6893, the overall discrepancy function for multiple-group analysis is defined. The same notation is applied here. To begin with, the overall discrepancy function $F(\mathbf{\Theta})$ is expressed as a weighted sum of individual discrepancy functions F_i's for the groups as follows:

$$F(\mathbf{\Theta}) = \sum_{i=1}^{k} t_i F_i(\mathbf{\Theta})$$

where

$$t_i = \frac{N_i - 1}{N - k}$$

is the weight for group i,

$$N = \sum_{i=1}^{k} N_i$$

is the total sample size, and N_i is the sample size for group i.

The gradient $g(\boldsymbol{\Theta})$ and the Hessian $H(\boldsymbol{\Theta})$ are now defined as weighted sum of individual functions. That is,

$$g(\boldsymbol{\Theta}) = \sum_{i=1}^{k} t_i g_i(\boldsymbol{\Theta}) = \sum_{i=1}^{k} t_i \frac{\partial}{\partial \boldsymbol{\Theta}} F_i(\boldsymbol{\Theta})$$

and

$$H(\boldsymbol{\Theta}) = \sum_{i=1}^{k} t_i H_i(\boldsymbol{\Theta}) = \sum_{i=1}^{k} t_i \frac{\partial^2}{\partial \boldsymbol{\Theta} \partial \boldsymbol{\Theta}'} F_i(\boldsymbol{\Theta})$$

Suppose that the mean and covariance structures fit perfectly with $\boldsymbol{\Theta} = \boldsymbol{\Theta}_o$ in the population. If each t_i converges to a fixed constant τ_i ($\tau_i > 0$) with increasing total sample size, the information matrix can be written as:

$$I(\boldsymbol{\Theta}_o) = \frac{1}{2} \sum_{i=1}^{k} \tau_i \mathcal{E}(H_i(\boldsymbol{\Theta}_o))$$

To approximate this information matrix, an empirical counterpart is used:

$$I(\hat{\boldsymbol{\Theta}}) = \frac{1}{2} \sum_{i=1}^{k} t_i H_i(\hat{\boldsymbol{\Theta}})$$

which is evaluated at the values of the sample estimates $\hat{\boldsymbol{\Theta}}$. Again, this empirical information matrix, rather than the unknown $I(\boldsymbol{\Theta}_o)$, is the "information matrix" output in PROC TCALIS results.

Taking the inverse of the empirical information matrix with sample size adjustment, PROC TCALIS approximates the estimated covariance matrix of $\hat{\boldsymbol{\Theta}}$ in multiple-group analysis by:

$$((N-k) I(\hat{\boldsymbol{\Theta}}))^{-1} = ((N-k) \frac{1}{2} H(\hat{\boldsymbol{\Theta}}))^{-1} = \frac{2}{N-k} \sum_{i=1}^{k} t_i H_i^{-1}(\hat{\boldsymbol{\Theta}})$$

Approximate standard errors for $\hat{\boldsymbol{\Theta}}$ can then be computed as the square roots of the diagonal elements of the estimated covariance matrix. Again, for ULS and DWLS estimation, the theory does not apply and so there are no standard errors computed in these cases.

Testing Rank Deficiency in the Approximate Covariance Matrix for Parameter Estimates

When computing the approximate covariance matrix and hence the standard errors for the parameter estimates, inversion of the scaled information matrix or Hessian matrix is involved. The numerical condition of the information matrix can be very poor in many practical applications, especially for the analysis of unscaled covariance data. The following four-step strategy is used for the inversion of the information matrix.

1. The inversion (usually of a normalized matrix $\mathbf{D}^{-1}\mathbf{H}\mathbf{D}^{-1}$) is tried using a modified form of the Bunch and Kaufman (1977) algorithm, which allows the specification of a different singularity criterion for each pivot. The following three criteria for the detection of rank loss in the information matrix are used to specify thresholds:

   - *ASING* specifies absolute singularity.
   - *MSING* specifies relative singularity depending on the whole matrix norm.
   - *VSING* specifies relative singularity depending on the column matrix norm.

   If no rank loss is detected, the inverse of the information matrix is used for the covariance matrix of parameter estimates, and the next two steps are skipped.

2. The linear dependencies among the parameter subsets are displayed based on the singularity criteria.

3. If the number of parameters t is smaller than the value specified by the G4= option (the default value is 60), the Moore-Penrose inverse is computed based on the eigenvalue decomposition of the information matrix. If you do not specify the NOPRINT option, the distribution of eigenvalues is displayed, and those eigenvalues that are set to zero in the Moore-Penrose inverse are indicated. You should inspect this eigenvalue distribution carefully.

4. If PROC TCALIS did not set the right subset of eigenvalues to zero, you can specify the COVSING= option to set a larger or smaller subset of eigenvalues to zero in a further run of PROC TCALIS.

Counting the Degrees of Freedom

When fitting covariance and mean structure models, the population moments are hypothesized to be functions of model parameters $\boldsymbol{\Theta}$. The population moments refer to the first-order moments (means) and the second-order central moments (variances of and covariances among the variables). Usually, the number of nonredundant population moments is larger than the number of model parameters for a structural model. The difference between the two is the degrees of freedom (df) of your model.

Formally, define a multiple-group situation where you have k independent groups in your model. The set of variables in each group might be different so that you have $p_1, p_2, \ldots, p_k$ mainfest or observed variables for the k groups. It is assumed that the primary interest is to study the covariance structures. The inclusion of mean structures is optional for each of these groups. Define

$\delta_1, \delta_2, \cdots, \delta_k$ as zero-one indicators of the mean structures for the groups. If δ_i takes the value of one, it means that the mean structures of group i is modeled. The total number of nonredundant elements in the moment matrices is thus computed by:

$$q = \sum_{i=1}^{k} (p_i(p_i + 1)/2 + \delta_i p_i)$$

The first term in the summation represents the number of lower triangular elements in the covariance or correlation matrix, while the second term represents the number of elements in the mean matrix. Let t be the total number of independent parameters **in the model**. The degrees of freedom is:

$$df = q - (t - c)$$

where c represents the number of linear equality constraints imposed on the independent parameters in the model. In effect, the $(t-c)$ expression means that each nonredundant linear equality constraint reduces one independent parameter.

Counting the Number of Independent Parameters

To count the number of independent parameters in the model, first you have to distinguish them from the dependent parameters. Dependent parameters are expressed as functions of other parameters in the SAS programming statements. That is, a parameter is dependent if it appears at the left-hand side of the equal sign in a SAS programming statement.

A parameter is independent if it is not dependent. An independent parameter can be specified in the main or subsidiary model specification statements or the PARMS statement, or it is generated automatically by PROC TCALIS as additional parameters. Quite intuitively, all independent parameter specified in the main or subsidiary model specification statements are independent parameters **in the model**. All automatic parameters added by PROC TCALIS are also independent parameters **in the model**.

Intentionally or not, some independent parameters specified in the PARMS statement might not be counted as independent parameters in the model. Independent parameters in the PARMS statement belong in the model only when they are used to define at least one dependent parameter specified in the main or subsidiary model specification statements. This restriction eliminates the counting of superfluous independent parameters which have no bearing of model specification.

Note that when counting the number of independent parameters, you are counting the number of distinct independent parameter names but not the number of distinct parameter locations for independent parameters. For example, consider the following statement for defining the error variances in a LINEQS model:

```
std    E1-E3 = vare1 vare2 vare3;
```

You define three variance parameter locations with three independent parameters vare1, vare2, and vare3. However, in the following specification:

```
std    E1-E3 = vare vare vare;
```

you still have three variance parameter locations to define, but the number of independent parameter is only one, which is the parameter named vare.

Counting the Number of Linear Equality Constraints

The linear equality constraints refer to those specified in the BOUNDS or LINCON statement. For example, consider the following specification:

```
bounds  3 <= parm01 <= 3;
lincon  3 * parm02 + 2 * parm03 = 12;
```

In the BOUNDS statement, parm01 is constrained to a fixed number 3, and in the LINCON statement, parm02 and parm03 are constrained linearly. In effect, these two statements reduce two independent parameters from the model. In the degrees of freedom formula, the value of c is 2 for this example.

Adjustment of Degrees of Freedom

In some cases, computing degrees of freedom for model fit is not so straightforward. Two important cases are considered in the following.

The first case is when you set linear inequality or boundary constraints in your model, and these inequality or boundary constraints become "active" in your final solution. For example, you might have set inequality boundary and linear constraints as:

```
bounds  0 <= var01;
lincon  3 * beta1 + 2 * beta2 >= 7;
```

The optimal solution occurs at the boundary point so that you observe in the final solution the following two equalities:

```
var01 = 0,
3 * beta1 + 2 * beta2 = 7
```

These two active constraints reduce the number of independent parameters of your original model. As a result, PROC TCALIS will automatically increase the degrees of freedom by the number of active linear constraints. Adjusting degrees of freedom not only affects the significance of the model fit chi-square statistic, but it also affects the computation of many fit statistics and indices. Refer to Dijkstra (1992) for a discussion of the validity of statistical inferences with active boundary constraints.

Automatically adjusting df in such a situation might not be totally justified in all cases. Statistical estimation is subject to sampling fluctuation. Active constraints might not occur when fitting the same model in new samples. If the researcher believes that those linear inequality and boundary constraints have a small chance of becoming active in repeated sampling, it might be more suitable to turn off the automatic adjustment by using the NOADJDF option in the PROC TCALIS statement.

Another case where you need to pay attention to the computation of degrees of freedom is when you fit correlation models. The degrees-of-freedom calculation in PROC TCALIS applies mainly to models with covariance structures with or without mean structures. When you model correlation structures, the degrees of freedom calculation in PROC TCALIS is a straightforward generalization

of the covariance structures. It does not take the fixed ones at the diagonal elements of the sample correlation matrix into account. Some might argue that with correlation structures, the degrees of freedom should be reduced by the total number of diagonal elements in the correlation matrices in the model. While PROC TCALIS does not do this automatically, you can use the DFREDUCE=*i* option to specify the adjustment, where i can be any positive or negative integer. The df value is reduced by the DFREDUCE= value.

A Different Type of Degrees of Freedom

The degrees of freedom for model fitting has to be distinguished from another type of degrees of freedom. In a regression problem, the number of degrees of freedom for the error variance estimate is the number of observations in the data set minus the number of parameters. The NOBS=, DFR= (RDF=), and DFE= (EDF=) options refer to degrees of freedom in this sense. However, these values are not related to the degrees of freedom for the model fit statistic. The NOBS=, DFR=, and DFE= options should be used in PROC TCALIS to specify the effective number of observations in the input data set only.

Assessment of Fit

In PROC TCALIS, there are three main tools for assessing model fit:

- residuals for the fitted means or covariances
- overall model fit indices
- squared multiple correlations and determination coefficients

This section contains a collection of formulas for these assessment tools. The following notation is used:

- N for the total sample size
- k for the total number of independent groups in analysis
- p for the number of manifest variables
- t for the number of parameters to estimate
- $\boldsymbol{\Theta}$ for the t-vector of parameters, $\hat{\boldsymbol{\Theta}}$ for the estimated parameters
- $\mathbf{S} = (s_{ij})$ for the $p \times p$ input covariance or correlation matrix
- $\bar{\boldsymbol{x}} = (\bar{x}_i)$ for the p-vector of sample means
- $\hat{\boldsymbol{\Sigma}} = \boldsymbol{\Sigma}(\hat{\boldsymbol{\Theta}}) = (\hat{\sigma}_{ij})$ for the predicted covariance or correlation matrix
- $\hat{\boldsymbol{\mu}} = (\hat{\mu}_i)$ for the predicted mean vector

- δ for indicating the modeling of the mean structures
- $\mathbf{W}$ for the weight matrix
- f_{min} for the minimized function value of the fitted model
- d_{min} for the degrees of freedom of the fitted model

In multiple-group analyses, subscripts are used to distinguish independent groups or samples. For example, $N_1, N_2, \ldots, N_r, \ldots, N_k$ denote the sample sizes for k groups. Similarly, notation such as p_r, $\mathbf{S}_r$, $\bar{\boldsymbol{x}}_r$, $\hat{\boldsymbol{\Sigma}}_r$, $\hat{\boldsymbol{\mu}}_r$, δ_r, and $\mathbf{W}_r$ is used for multiple-group situations.

Residuals

Residuals indicate how well each entry or element in the mean or covariance matrix is fitted. Large residuals indicate bad fit.

PROC TCALIS computes four types of residuals and writes them to the OUTSTAT= data set when requested.

- **raw residuals**

  $$s_{ij} - \hat{\sigma}_{ij}, \quad \bar{x}_i - \hat{\mu}_i$$

  for the covariance and mean residuals, respectively. The raw residuals are displayed whenever the PALL, PRINT, or RESIDUAL option is specified.

- **variance standardized residuals**

  $$\frac{s_{ij} - \hat{\sigma}_{ij}}{\sqrt{s_{ii}s_{jj}}}, \quad \frac{\bar{x}_i - \hat{\mu}_i}{\sqrt{s_{ii}}}$$

  for the covariance and mean residuals, respectively. The variance standardized residuals are displayed when you specify one of the following:

  - the PALL, PRINT, or RESIDUAL option and METHOD=NONE, METHOD=ULS, or METHOD=DWLS
  - RESIDUAL=VARSTAND

  The variance standardized residuals are equal to those computed by the EQS 3 program (Bentler 1995).

- **asymptotically standardized residuals**

  $$\frac{s_{ij} - \hat{\sigma}_{ij}}{\sqrt{v_{ij,ij}}}, \quad \frac{\bar{x}_i - \hat{\mu}_i}{\sqrt{u_{ii}}}$$

  for the covariance and mean residuals, respectively; with

  $$v_{ij,ij} = (\hat{\boldsymbol{\Gamma}}_1 - \mathbf{J}_1 \hat{\mathrm{Cov}}(\hat{\boldsymbol{\Theta}})\mathbf{J}_1')_{ij,ij}$$

$$u_{ii} = (\hat{\mathbf{\Gamma}}_2 - \mathbf{J}_2 \hat{\mathrm{Cov}}(\hat{\mathbf{\Theta}})\mathbf{J}_2')_{ii}$$

where $\hat{\mathbf{\Gamma}}_1$ is the $p^2 \times p^2$ estimated asymptotic covariance matrix of sample covariances, $\hat{\mathbf{\Gamma}}_2$ is the $p \times p$ estimated asymptotic covariance matrix of sample means, $\mathbf{J}_1$ is the $p^2 \times t$ Jacobian matrix $d\,\mathbf{\Sigma}/d\,\mathbf{\Theta}$, $\mathbf{J}_2$ is the $p \times t$ Jacobian matrix $d\,\boldsymbol{\mu}/d\,\mathbf{\Theta}$, and $\hat{Cov}(\hat{\mathbf{\Theta}})$ is the $t \times t$ estimated covariance matrix of parameter estimates, all evaluated at the sample moments and estimated parameter values. See the next section for the definitions of $\hat{\mathbf{\Gamma}}_1$ and $\hat{\mathbf{\Gamma}}_2$. Asymptotically standardized residuals are displayed when one of the following conditions is met:

- The PALL, the PRINT, or the RESIDUAL option is specified, and METHOD=ML, METHOD=GLS, or METHOD=WLS, and the expensive information and Jacobian matrices are computed for some other reason.
- RESIDUAL= ASYSTAND is specified.

The asymptotically standardized residuals are equal to those computed by the LISREL 7 program (Jöreskog and Sörbom 1988) except for the denominator in the definition of matrix $\hat{\mathbf{\Gamma}}_1$.

- **normalized residuals**

$$\frac{s_{ij} - \hat{\sigma}_{ij}}{\sqrt{(\hat{\mathbf{\Gamma}}_1)_{ij,ij}}}, \quad \frac{\bar{x}_i - \hat{\mu}_i}{\sqrt{(\hat{\mathbf{\Gamma}}_2)_{ii}}}$$

for the covariance and mean residuals, respectively; with $\hat{\mathbf{\Gamma}}_1$ as the $p^2 \times p^2$ estimated asymptotic covariance matrix of sample covariances; and $\hat{\mathbf{\Gamma}}_2$ as the $p \times p$ estimated asymptotic covariance matrix of sample means.

Diagonal elements of $\hat{\mathbf{\Gamma}}_1$ and $\hat{\mathbf{\Gamma}}_2$ are defined for the following methods:

- **GLS**: $(\hat{\mathbf{\Gamma}}_1)_{ij,ij} = \frac{1}{(N-1)}(s_{ii}s_{jj} + s_{ij}^2)$ and $(\hat{\mathbf{\Gamma}}_2)_{ii} = \frac{1}{(N-1)}s_{ii}$
- **ML**: $(\hat{\mathbf{\Gamma}}_1)_{ij,ij} = \frac{1}{(N-1)}(\hat{\sigma}_{ii}\hat{\sigma}_{jj} + \hat{\sigma}_{ij}^2)$ and $(\hat{\mathbf{\Gamma}}_2)_{ii} = \frac{1}{(N-1)}\hat{\sigma}_{ii}$
- **WLS**: $(\hat{\mathbf{\Gamma}}_1)_{ij,ij} = \frac{1}{(N-1)}W_{ij,ij}$ and $(\hat{\mathbf{\Gamma}}_2)_{ii} = \frac{1}{(N-1)}s_{ii}$

where $\mathbf{W}$ in the **WLS** method is the weight matrix for the second-order moments.

Normalized residuals are displayed when one of the following conditions is met:

- The PALL, PRINT, or RESIDUAL option is specified, and METHOD=ML, METHOD=GLS, or METHOD=WLS, and the expensive information and Jacobian matrices are **not** computed for some other reasons.
- RESIDUAL=NORM is specified.

The normalized residuals are equal to those computed by the LISREL VI program (Jöreskog and Sörbom 1985) except for the definition of the denominator in computing matrix $\hat{\mathbf{\Gamma}}_1$.

For estimation methods that are not "best" generalized least squares estimators (Browne 1982, 1984), such as METHOD=NONE, METHOD=ULS, or METHOD=DWLS, the assumption of an asymptotic covariance matrix $\mathbf{\Gamma}_1$ of sample covariances does not seem to be appropriate. In this

case, the normalized residuals should be replaced by the more relaxed variance standardized residuals. Computation of asymptotically standardized residuals requires computing the Jacobian and information matrices. This is computationally very expensive and is done only if the Jacobian matrix has to be computed for some other reasons—that is, if at least one of the following items is true:

- The default, PRINT, or PALL displayed output is requested, and neither the NOMOD nor NOSTDERR option is specified.
- Either the MODIFICATION (included in PALL), PCOVES, or STDERR (included in default, PRINT, and PALL output) option is requested or RESIDUAL=ASYSTAND is specified.
- The LEVMAR or NEWRAP optimization technique is used.
- An OUTMODEL= data set is specified without using the NOSTDERR option.
- An OUTEST= data set is specified without using the NOSTDERR option.

Since normalized residuals use an overestimate of the asymptotic covariance matrix of residuals (the diagonals of $\mathbf{\Gamma}_1$ and $\mathbf{\Gamma}_2$), the normalized residuals cannot be larger than the asymptotically standardized residuals (which use the diagonal of the form $\mathbf{\Gamma} - \mathbf{J}\hat{\mathrm{C}}\mathrm{ov}(\hat{\mathbf{\Theta}})\mathbf{J}'$).

Together with the residual matrices, the values of the average residual, the average off-diagonal residual, and the rank order of the largest values are displayed. The distributions of the normalized and standardized residuals are displayed also.

Overall Model Fit Indices

Instead of assessing the model fit by looking at a number of residuals of the fitted moments, an overall model fit index measures model fit by a single number. Although an overall model fit index is precise and easy to use, there are indeed many choices of overall fit indices. Unfortunately, researchers do not always have a consensus on the best set of indices to use in all occasions.

PROC TCALIS produces a large number of overall model fit indices in the fit summary table. If you prefer to display only a subset of these fit indices, you can use the ONLIST(ONLY)= option of the FITINDEX statement to customize the fit summary table.

Fit indices are classified into three classes in the fit summary table of PROC TCALIS:

- absolute or standalone Indices
- parsimony indices
- incremental indices

Absolute or Standalone Indices

These indices are constructed so that they measure model fit without comparing with a baseline model and without taking the model complexity into account. They measure the absolute fit of the model.

- **fit function or discrepancy function**
  The fit function or discrepancy function F is minimized during the optimization. See the section "Estimation Criteria" on page 6888 for definitions of various discrepancy functions available in PROC TCALIS. For a multiple-group analysis, the fit function can be written as a weighted average of discrepancy functions for k independent groups as:

  $$F = \sum_{r=1}^{k} a_r F_r$$

  where $a_r = \frac{(N_j - 1)}{(N-k)}$ and F_r are the group weight and the discrepancy function for the r-th group, respectively. Notice that although the groups are assumed to be independent in the model, in general F_r's are not independent when F is being minimized. The reason is that F_r's might have shared parameters in $\mathbf{\Theta}$ during estimation.

  The minimized function value of F will be denoted as f_{min}, which is always positive, with small values indicating good fit.

- **χ^2 test statistic**
  For the ML, GLS, and the WLS estimation, the overall χ^2 measure for testing model fit is:

  $$\chi^2 = (N - k) * f_{min}$$

  where f_{min} is the function value at the minimum, N is the total sample size, and k is the number of independent groups. The associated degrees of freedom is denoted by d_{min}.

  For the ML estimation, this gives the likelihood ratio test statistic of the specified structural model in the null hypothesis against an unconstrained saturated model in the alternative hypothesis. The χ^2 test is valid only if the observations are independent and identically distributed, the analysis is based on the nonstandardized sample covariance matrix $\mathbf{S}$, and the sample size N is sufficiently large (Browne 1982; Bollen 1989b; Jöreskog and Sörbom 1985). For ML and GLS estimates, the variables must also have an approximately multivariate normal distribution.

  In the output fit summary table of PROC TCALIS, the notation "Prob > Chi-Square" means "the probability of obtaining a greater χ^2 value than the observed value under the null hypothesis." This probability is also known as the p-value of the chi-square test statistic.

- **adjusted χ^2 value (Browne 1982)**
  If the variables are p-variate elliptic rather than normal and have significant amounts of multivariate kurtosis (leptokurtic or platykurtic), the χ^2 value can be adjusted to:

  $$\chi^2_{ell} = \frac{\chi^2}{\eta_2}$$

  where η_2 is the multivariate relative kurtosis coefficient.

- **Z-test (Wilson and Hilferty 1931)**
  The Z-test of Wilson and Hilferty assumes a p-variate normal distribution:

  $$Z = \frac{\sqrt[3]{\frac{\chi^2}{d}} - (1 - \frac{2}{9d})}{\sqrt{\frac{2}{9d}}}$$

  where d is the degrees of freedom of the model. Refer to McArdle (1988) and Bishop, Fienberg, and Holland (1977, p. 527) for an application of the Z-test.

- **critical N index (Hoelter 1983)**

$$\text{CN} = \frac{\chi^2_{crit}}{f_{min}} + 1$$

where χ^2_{crit} is the critical chi-square value for the given d degrees of freedom and probability $\alpha = 0.05$. Refer to Bollen (1989b, p. 277). Hoelter (1983) suggests that CN should be at least 200; however, Bollen (1989b) notes that the CN value might lead to an overly pessimistic assessment of fit for small samples.

- **root mean square residual (RMR)**
For a single-group analysis, the RMR is the root of the mean of the squared residuals:

$$\text{RMR} = \sqrt{\frac{2}{p(p+1+2\delta)}[\sum_{i}^{p}\sum_{j}^{i}(s_{ij}-\hat{\sigma}_{ij})^2 + \delta\sum_{i}^{p}(\bar{x}_i-\hat{\mu}_i)^2]}$$

For multiple-group analysis, PROC TCALIS computes the root mean square residual RMR_r for each group first. To obtain an overall RMR measure for the analysis, individual RMR_r's are weighted by the group weights $a_r = \frac{N_r-1}{N-k}$. That is,

$$\text{overall RMR} = \sqrt{\sum_{r=1}^{k} a_r \text{RMR}_r^2}$$

- **standardized root mean square residual (SRMR)**
For a single-group analysis, the SRMR is the root of the mean of the squared standardized residuals:

$$\text{SRMR} = \sqrt{\frac{2}{p(p+1+2\delta)}[\sum_{i}^{p}\sum_{j}^{i}\frac{(s_{ij}-\hat{\sigma}_{ij})^2}{s_{ii}s_{jj}} + \delta\sum_{i}^{p}\frac{(\bar{x}_i-\hat{\mu}_i)^2}{s_{ii}}]}$$

Similar to the calculation of the overall RMR, an overall measure of SRMR in a multiple-group analysis is a weighted average of the individual SRMR's. That is, with $a_r = \frac{N_r-1}{N-k}$

$$\text{overall SRMR} = \sqrt{\sum_{r=1}^{k} a_r \text{SRMR}_r^2}$$

- **goodness-of-fit index (GFI)**
For a single-group analysis, the goodness-of-fit index for the ULS, GLS, and ML estimation methods is:

$$GFI = 1 - \frac{Tr((\mathbf{W}^{-1}(\mathbf{S}-\hat{\boldsymbol{\Sigma}}))^2) + \delta(\bar{\boldsymbol{x}}-\hat{\boldsymbol{\mu}})'\mathbf{W}^{-1}(\bar{\boldsymbol{x}}-\hat{\boldsymbol{\mu}})}{Tr((\mathbf{W}^{-1}\mathbf{S})^2) + \delta\bar{\boldsymbol{x}}'\mathbf{W}^{-1}\bar{\boldsymbol{x}}}$$

with $\mathbf{W} = I$ for ULS, $\mathbf{W} = \mathbf{S}$ for GLS, and $\mathbf{W} = \hat{\boldsymbol{\Sigma}}$. For WLS and DWLS estimation,

$$GFI = 1 - \frac{(u-\hat{\eta})'\mathbf{W}^{-1}(u-\hat{\eta})}{u'\mathbf{W}^{-1}u}$$

where u is the vector of observed moments and $\hat{\eta}$ is the vector of fitted moments. When the mean structures are modeled, vectors u and $\hat{\eta}$ contains all the nonredundant elements vecs(**S**) in the covariance matrix and all the means. That is,

$$u = (\text{vecs}'(\mathbf{S}), \bar{\boldsymbol{x}}')', \quad \hat{\eta} = (\text{vecs}'(\hat{\boldsymbol{\Sigma}}), \hat{\boldsymbol{\mu}}')'$$

and the symmetric weight matrix **W** is of dimension $p \times (p+3)/2$. When the mean structures are not modeled, vectors u and $\hat{\eta}$ contains all the nonredundant elements vecs(**S**) in the covariance matrix only. That is,

$$u = \text{vecs}(\mathbf{S}), \quad \hat{\eta} = \text{vecs}(\hat{\boldsymbol{\Sigma}})$$

and the symmetric weight matrix **W** is of dimension $p \times (p+1)/2$. In addition, for the DWLS estimation, **W** is a diagonal matrix.

For a constant weight matrix **W**, the goodness-of-fit index is 1 minus the ratio of the minimum function value and the function value before any model has been fitted. The GFI should be between 0 and 1. The data probably do not fit the model if the GFI is negative or much larger than 1.

For a multiple-group analysis, individual GFI_r's are computed for groups. The overall measure is a weighted average of individual GFI_r's, using weight $a_r = \frac{N_r - 1}{N - k}$. That is,

$$\text{overall GFI} = \sum_{r=1}^{k} a_r \text{GFI}_r$$

Parsimony Indices

These indices are constructed so that the model complexity is taken into account when assessing model fit. In general, models with more parameters (fewer degrees of freedom) are penalized.

- **adjusted goodness-of-fit index (AGFI)**
  The AGFI is the GFI adjusted for the degrees of freedom d of the model,

  $$\text{AGFI} = 1 - \frac{c}{d}(1 - \text{GFI})$$

  where

  $$c = \sum_{r=1}^{k} \frac{p_k(p_k + 1 + 2\delta_k)}{2}$$

  computes the total number of elements in the covariance matrices and mean vectors for modeling. For single-group analyses, the AGFI corresponds to the GFI in replacing the total sum of squares by the mean sum of squares.

  CAUTION:

  - Large p and small d can result in a negative AGFI. For example, GFI= 0.90, p= 19, and d= 2 result in an AGFI of −8.5.

- AGFI is not defined for a saturated model, due to division by $d = 0$.
- AGFI is not sensitive to losses in d.

The AGFI should be between 0 and 1. The data probably do not fit the model if the AGFI is negative or much larger than 1. For more information, refer to Mulaik et al. (1989).

- **parsimonious goodness-of-fit index (PGFI)**
The PGFI (Mulaik et al. 1989) is a modification of the GFI that takes the parsimony of the model into account:

$$\text{PGFI} = \frac{d_{min}}{d_0}\text{GFI}$$

where d_{min} is the model degrees of freedom and d_0 is the degrees of freedom for the independence model. See the section "Incremental Indices" on page 6909 for the definition of independence model. The PGFI uses the same parsimonious factor as the parsimonious normed Bentler-Bonett index (James, Mulaik, and Brett 1982).

- **RMSEA index (Steiger and Lind 1980; Steiger 1998)**
The Steiger and Lind (1980) (see also Steiger 1998) root mean square error approximation (RMSEA) coefficient is:

$$\epsilon = \sqrt{k}\sqrt{\max(\frac{f_{min}}{d_{min}} - \frac{1}{(N-k)}, 0)}$$

The lower and upper limits of the $(1-\alpha)$%-confidence interval are computed using the cumulative distribution function of the noncentral chi-squared distribution $\Phi(x|\lambda, d)$. With $x = (N-k)\,f_{min}$, λ_L satisfying $\Phi(x|\lambda_L, d_{min}) = 1-\frac{\alpha}{2}$, and λ_U satisfying $\Phi(x|\lambda_U, d_{min}) = \frac{\alpha}{2}$:

$$(\epsilon_{\alpha_L}; \epsilon_{\alpha_U}) = (\sqrt{k}\sqrt{\frac{\lambda_L}{(N-k)d_{min}}}; \sqrt{k}\sqrt{\frac{\lambda_U}{(N-k)d_{min}}})$$

Refer to Browne and Du Toit (1992) for more details. The size of the confidence interval can be set by the option ALPHARMS=α, $0 \le \alpha \le 1$. The default is $\alpha = 0.1$, which corresponds to the 90% confidence interval for the RMSEA.

- **probability for test of close fit (Browne and Cudeck 1993)**
The traditional exact χ^2 test hypothesis $H_0{:}\,\epsilon = 0$ is replaced by the null hypothesis of close fit $H_0{:}\,\epsilon \le 0.05$ and the exceedance probability P is computed as:

$$P = 1 - \Phi(x|\lambda^*, d_{min})$$

where $x = (N-k)\,f_{min}$ and $\lambda^* = 0.05^2(N-k)d_{min}/k$. The null hypothesis of close fit is rejected if P is smaller than a pre-specified level (for example, $P < 0.05$).

- **ECVI: expected cross validation index (Browne and Cudeck 1993)**
The following formulas for ECVI are limited to the case of single-sample analysis without mean structures. For other cases, ECVI is not defined in PROC TCALIS. For GLS and WLS, the estimator c of the ECVI is linearly related to AIC, Akaike's Information Criterion (Akaike 1974; 1987):

$$c = f_{min} + \frac{2t}{N-1}$$

For ML estimation, c_{ML} is used:

$$c_{ML} = f_{min} + \frac{2t}{N-p-2}$$

For GLS and WLS, the confidence interval $(c_L; c_U)$ for ECVI is computed using the cumulative distribution function $\Phi(x|\lambda, d_{min})$ of the noncentral chi-squared distribution,

$$(c_L; c_U) = (\frac{\lambda_L + p(p+1)/2 + t}{(N-1)}; \frac{\lambda_U + p(p+1)/2 + t}{(N-1)})$$

with $x = (N-1)\,f_{min}$, $\Phi(x|\lambda_U, d_{min}) = \frac{\alpha}{2}$, and $\Phi(x|\lambda_L, d_{min}) = 1 - \frac{\alpha}{2}$.
For ML, the confidence interval $(c_L^*; c_U^*)$ for ECVI is:

$$(c_L^*; c_U^*) = (\frac{\lambda_L^* + p(p+1)/2 + t}{N-p-2}; \frac{\lambda_U^* + p(p+1)/2 + t}{N-p-2})$$

where $x = (N-p-2)\,f_{min}$, $\Phi(x|\lambda_U^*, d_{min}) = \frac{\alpha}{2}$ and $\Phi(x|\lambda_L^*, d_{min}) = 1 - \frac{\alpha}{2}$. Refer to Browne and Cudeck (1993). The size of the confidence interval can be set by the option ALPHAECV=α, $0 \leq \alpha \leq 1$. The default is $\alpha = 0.1$, which corresponds to the 90% confidence interval for the ECVI.

- **Akaike's information criterion (AIC) (Akaike 1974; 1987)**
  This is a criterion for selecting the best model among a number of candidate models. The model that yields the smallest value of AIC is considered the best.

  $$\text{AIC} = \chi^2 - 2d_{min}$$

- **consistent Akaike's information criterion (CAIC) (Bozdogan 1987)**
  This is another criterion, similar to AIC, for selecting the best model among alternatives. The model that yields the smallest value of CAIC is considered the best. CAIC is preferred by some people to AIC or the χ^2 test.

  $$\text{CAIC} = \chi^2 - (ln(N) + 1)d_{min}$$

- **Schwarz's Bayesian criterion (SBC) (Schwarz 1978; Sclove 1987)**
  This is another criterion, similar to AIC, for selecting the best model. The model that yields the smallest value of SBC is considered the best. SBC is preferred by some people to AIC or the χ^2 test.

  $$\text{SBC} = \chi^2 - ln(N)d_{min}$$

- **McDonald's measure of centrality (McDonald and Hartmann 1992)**

  $$\text{CENT} = \exp(-\frac{(\chi^2 - d_{min})}{2N})$$

Incremental Indices

These indices are constructed so that the model fit is assessed through the comparison with a baseline model. The baseline model is usually the independence model where all covariances among manifest variables are assumed to be zeros. The only parameters in the independence model are the diagonals of covariance matrix. If modeled, the mean structures are saturated in the independence model. For multiple-group analysis, the overall independence model consists of component independence models for each group.

In the following, let f_0 and d_0 denote the minimized discrepancy function value and the associated degrees of freedom, respectively, for the independence model; and f_{min} and d_{min} denote the minimized discrepancy function value and the associated degrees of freedom, respectively, for the model being fitted in the null hypothesis.

- **Bentler comparative fit index (Bentler 1995)**

$$\text{CFI} = 1 - \frac{\max((N-k)f_{min} - d_{min}, 0)}{\max((N-k)f_0 - d_0, 0)}$$

- **Bentler-Bonett normed fit index (NFI) (Bentler and Bonett 1980)**

$$\Delta = \frac{f_0 - f_{min}}{f_0}$$

  Mulaik et al. (1989) recommend the parsimonious weighted form called parsimonious normed fit index (PNFI) (James, Mulaik, and Brett 1982).

- **Bentler-Bonett nonnormed coefficient (Bentler and Bonett 1980)**

$$\rho = \frac{f_0/d_0 - f_{min}/d_{min}}{f_0/d_0 - 1/(N-k)}$$

  Refer to Tucker and Lewis (1973).

- **normed index ρ_1 (Bollen 1986)**

$$\rho_1 = \frac{f_0/d_0 - f_{min}/d_{min}}{f_0/d_0}$$

  ρ_1 is always less than or equal to 1; $\rho_1 < 0$ is unlikely in practice. Refer to the discussion in Bollen (1989a).

- **nonnormed index Δ_2 (Bollen 1989a)**

$$\Delta_2 = \frac{f_0 - f_{min}}{f_0 - \frac{d_{min}}{(N-k)}}$$

  is a modification of Bentler and Bonett's Δ that uses d and "lessens the dependence" on N. Refer to the discussion in Bollen (1989b). Δ_2 is identical to Mulaik et al.'s (1989) IFI2 index.

- **parsimonious normed fit index (James, Mulaik, and Brett 1982)**
  The PNFI is a modification of Bentler-Bonett's normed fit index that takes parsimony of the model into account,

  $$\text{PNFI} = \frac{d_{min}}{d_0}\frac{(f_0 - f_{min})}{f_0}$$

  The PNFI uses the same parsimonious factor as the parsimonious GFI of Mulaik et al. (1989).

Fit Indices and Estimation Methods

Note that not all fit indices are reasonable or appropriate for all estimation methods set by the METHOD= option of the PROC TCALIS statement. The availability of fit indices is summarized as follows:

- Adjusted (elliptic) chi-square and its probability are available only for METHOD=ML or GLS and with the presence of raw data input.
- For METHOD=ULS or DWLS, probability of the chi-square value, RMSEA and its confidence intervals, probability of close fit, ECVI and its confidence intervals, critical N index, Z-test, AIC, CAIC, SBC, and measure of centrality are not appropriate and therefore not displayed.

Individual Fit Indices for Multiple Groups

When you compare the fits of individual groups in a multiple-group analysis, you can examine the residuals of the groups to gauge which group is fitted better than the others. While examining residuals is good for knowing specific locations with inadequate fit, summary measures like fit indices for individual groups would be more convenient for overall comparisons among groups.

Although the overall fit function is a weighted sum of individual fit functions for groups, these individual functions are not statistically independent. Therefore, in general you cannot partition the degrees of freedom or χ^2 value according to the groups. This eliminates the possibility of breaking down those fit indices that are functions of degrees of freedom or χ^2 for group comparison purposes. Bearing this fact in mind, PROC TCALIS computes only a limited number of descriptive fit indices for individual groups.

- **fit function**
  The overall fit function is:

  $$F = \sum_{r=1}^{k} a_r F_r$$

  where $a_r = \frac{(N_j - 1)}{(N - k)}$ and F_r are the group weight and the discrepancy function for group r, respectively. The value of unweighted fit function F_r for the r-th group is denoted by:

  $$f_r$$

  This f_r value provides a measure of fit in the r-th group without taking the sample size into account. The large the f_r, the worse the fit for the group.

- **percentage contribution to the chi-square**
  The percentage contribution of group r to the chi-square is:

  $$\text{percentage contribution} = a_r f_r / f_{min} \times 100\%$$

  where f_r is the value of F_r with F minimized at the value f_{min}. This percentage value provides a descriptive measure of fit of the moments in group r, weighted by its sample size. The group with the largest percentage contribution accounts for the most lack of fit in the overall model.

- **root mean square residual (RMR)**
  For the r-th group, the total number of moments being modeled is:

  $$g = \frac{p_r(p_r + 1 + 2\delta_r)}{2}$$

  where p_r is the number of variables and δ_r is the indicator variable of the mean structures in the r-th group. The root mean square residual for the r-th group is:

  $$\text{RMR}_r = \sqrt{\frac{1}{g}[\sum_i^{p_r}\sum_j^{i}([\mathbf{S}_r]_{ij} - [\hat{\boldsymbol{\Sigma}}_r]_{ij})^2 + \delta_r \sum_i^{p_r}([\bar{\boldsymbol{x}}_r]_i - [\hat{\boldsymbol{\mu}}_r]_i)^2]}$$

- **standardized root mean square residual (SRMR)**
  For the r-th group, the standardized root mean square residual is:

  $$\text{SRMR} = \sqrt{\frac{1}{g}[\sum_i^{p_r}\sum_j^{i}\frac{([\mathbf{S}_r]_{ij} - [\hat{\boldsymbol{\Sigma}}_r]_{ij})^2}{[\mathbf{S}_r]_{ii}[\mathbf{S}_r]_{jj}} + \delta_r \sum_i^{p_r}\frac{([\bar{\boldsymbol{x}}_r]_i - [\hat{\boldsymbol{\mu}}_r]_i)^2}{[\mathbf{S}_r]_{ii}}]}$$

- **goodness-of-fit index (GFI)**
  For the ULS, GLS, and ML estimation, the goodness-of-fit index (GFI) for the r-th group is:

  $$GFI = 1 - \frac{Tr((\mathbf{W}_r^{-1}(\mathbf{S}_r - \hat{\boldsymbol{\Sigma}}_r))^2) + \delta_r(\bar{\boldsymbol{x}}_r - \hat{\boldsymbol{\mu}}_r)'\mathbf{W}_r^{-1}(\bar{\boldsymbol{x}}_r - \hat{\boldsymbol{\mu}}_r)}{Tr((\mathbf{W}_r^{-1}\mathbf{S}_r)^2) + \delta_r \bar{\boldsymbol{x}}_r'\mathbf{W}_r^{-1}\bar{\boldsymbol{x}}_r}$$

  with $\mathbf{W}_r = I$ for ULS, $\mathbf{W}_r = \mathbf{S}_r$ for GLS, and $\mathbf{W}_r = \hat{\boldsymbol{\Sigma}}_r$. For the WLS and DWLS estimation,

  $$GFI = 1 - \frac{(u_r - \hat{\eta}_r)'\mathbf{W}_r^{-1}(u_r - \hat{\eta}_r)}{u_r'\mathbf{W}_r^{-1}u_r}$$

  where u_r is the vector of observed moments and $\hat{\eta}_r$ is the vector of fitted moments for the r-th group ($r = 1, \ldots, k$).

  When the mean structures are modeled, vectors u_r and $\hat{\eta}_r$ contain all the nonredundant elements $\text{vecs}(\mathbf{S}_r)$ in the covariance matrix and all the means, and $\mathbf{W}_r$ is the weight matrix for covariances and means. When the mean structures are not modeled, u_r, $\hat{\eta}_r$, and $\mathbf{W}_r$ contain elements pertaining to the covariance elements only. Basically, formulas presented here are the same as the case for a single-group GFI. The only thing added here is the subscript r to denote individual group measures.

- **Bentler-Bonnett normed fit index (NFI)**
  For the r-th group, the Bentler-Bonnett NFI is:

$$\Delta_r = \frac{f_{0r} - f_r}{f_{0r}}$$

  where f_{0r} is the function value for fitting the independence model to the r-th group. The larger the value of Δ_r, the better is the fit for the group. Basically, the formula here is the same as the overall Bentler-Bonnet NFI. The only difference is that the subscript r is added to denote individual group measures.

Squared Multiple Correlations and Determination Coefficients

In the section, squared multiple correlations for endogenous variables are defined. Squared multiple correlation is computed for all of these five estimation methods: ULS, GLS, ML, WLS, and DWLS. These coefficients are also computed as in the LISREL VI program of Jöreskog and Sörbom (1985). The DETAE, DETSE, and DETMV determination coefficients are intended to be multivariate generalizations of the squared multiple correlations for different subsets of variables. These coefficients are displayed only when you specify the PDETERM option.

- **R^2 values corresponding to endogenous variables**

$$R^2 = 1 - \frac{\widehat{\mathrm{Evar}}(y)}{\widehat{\mathrm{Var}}(y)}$$

  where y denotes an endogenous variable, $\widehat{\mathrm{Var}}(y)$ denotes its variance, and $\widehat{\mathrm{Evar}}(y)$ denotes its error (or unsystematic) variance. The variance and error variance are estimated under the model.

- **total determination of all equations**

$$\mathrm{DETAE} = 1 - \frac{|\widehat{\mathrm{Ecov}}(\mathbf{y}, \boldsymbol{\eta})|}{|\widehat{\mathrm{Cov}}(\mathbf{y}, \boldsymbol{\eta})|}$$

  where the $\mathbf{y}$ vector denotes all manifest dependent variables, the $\boldsymbol{\eta}$ vector denotes all latent dependent variables, $\widehat{\mathrm{Cov}}(\mathbf{y}, \boldsymbol{\eta})$ denotes the covariance matrix of $\mathbf{y}$ and $\boldsymbol{\eta}$, and $\widehat{\mathrm{Ecov}}(\mathbf{y}, \boldsymbol{\eta})$ denotes the error covariance matrix of $\mathbf{y}$ and $\boldsymbol{\eta}$. The covariance matrices are estimated under the model.

- **total determination of latent equations**

$$\mathrm{DETSE} = 1 - \frac{|\widehat{\mathrm{Ecov}}(\boldsymbol{\eta})|}{|\widehat{\mathrm{Cov}}(\boldsymbol{\eta})|}$$

  where the $\boldsymbol{\eta}$ vector denotes all latent dependent variables, $\widehat{\mathrm{Cov}}(\boldsymbol{\eta})$ denotes the covariance matrix of $\boldsymbol{\eta}$, and $\widehat{\mathrm{Ecov}}(\boldsymbol{\eta})$ denotes the error covariance matrix of $\boldsymbol{\eta}$. The covariance matrices are estimated under the model.

- **total determination of the manifest equations**

$$\text{DETMV} = 1 - \frac{|\widehat{\text{Ecov}}(\mathbf{y})|}{|\widehat{\text{Cov}}(\mathbf{y})|}$$

where the $\mathbf{y}$ vector denotes all manifest dependent variables, $\widehat{\text{Cov}}(\mathbf{y})$ denotes the covariance matrix of $\mathbf{y}$, $\widehat{\text{Ecov}}(\mathbf{y})$ denotes the error covariance matrix of $\mathbf{y}$, and $|A|$ denotes the determinant of matrix A. All the covariance matrices in the formula are estimated under the model.

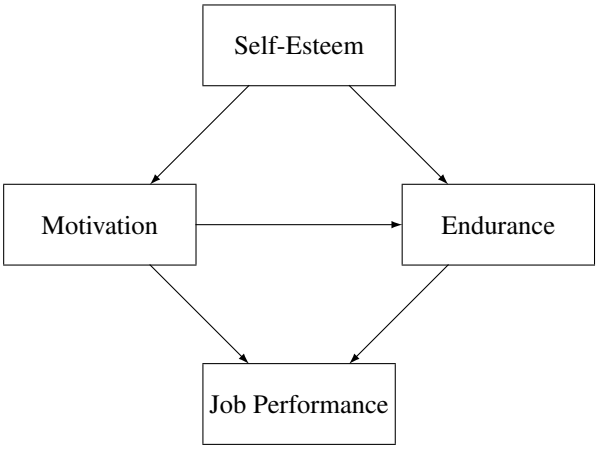
You can also use the DETERM statement to request the computations of determination coefficients for any subsets of dependent variables.

Total, Direct, and Indirect Effects

Most structural equation models involve the specification of the effects of variables on each other. Whenever you specify equations in the LINEQS model, paths in the PATH model, path coefficient parameters in the RAM model, variable-factor relations in the FACTOR model, or regression coefficients in model matrices of the LISMOD model, you are specifying direct effects of predictor variables on outcome variables. All direct effects are represented by the associated regression coefficients, either fixed or free, in the specifications. You can examine the direct effect estimates easily in the output for model estimation.

However, direct effects are not the only effects that are important. In some cases, the indirect effects or total effects are of interest too. For example, suppose Self-Esteem is an important factor of Job Performance in your theory. Although it does not have a direct effect on Job Performance, it affects Job Performance through its influences on Motivation and Endurance. Also, Motivation has a direct effect on Endurance in your theory. The following path diagram summarizes such a theory:

Figure 88.4 Direct and Indirect Effects of Self-Esteem on Job Performance



Clearly, each path in the diagram represents a direct effect of a predictor variable on an outcome variable. Less apparent are the total and indirect effects implied by the same path diagram. Despite

this, interesting theoretical questions regarding the total and indirect effects can be raised in such a model. For example, even though there is no direct effect of Self-Esteem on Job Performance, what is its indirect effect on Job Performance? In addition to its direct effect on Job Performance, Motivation also has an indirect effect on Job Performance via its effect on Endurance. So, what is the total effect of Motivation on Job Performance and what portion of this total effect is indirect? The TOTEFF option of the TCALIS statement and the EFFPART statement are designed to address these questions. By using the TOTEFF option or the EFFPART statement, PROC TCALIS can compute the total, direct, and indirect effects of any sets of predictor variables on any sets of outcome variables. In this section, formulas for computing these effects are presented.

Formulas for Computing Total, Direct and Indirect Effects

No matter which modeling language is used, variables in a model can be classified into three groups. The first group is the so-called dependent variables, which serve as outcome variables at least once in the model. The other two groups consist of the remaining independent variables, which never serve as outcome variables in the model. The second group consists of independent variables that are unsystematic sources such as error and disturbance variables. The third group consists of independent variables that are systematic sources only.

Any variable, no matter which group it falls into, can have effects on the first group of variables. By definition, however, effects of variables in the first group on the other two groups do not exist. Because the effects of unsystematic sources in the second group are treated as residual effects on the first group of dependent variables, these effects are trivial in the sense that they always serve as direct effects only. That is, the effects from the second group of unsystematic sources partition trivially—total effects are always the same as the direct effects for this group. Therefore, for the purpose of effect analysis or partitioning, only the first group (dependent variables) and the third group (systematic independent variables) are considered.

Define u to be the set of n_u dependent variables in the first group and w to be the set of n_w systematic independent variables in the third group. Variables in both groups can be manifest or latent. All variables in the effect analysis is thus represented by the vector $(u', w')'$.

The $(n_u + n_w) \times (n_u + n_w)$ matrix $\mathbf{D}$ of **direct** effects refers to the path coefficients from all column variables to the row variables. This matrix is represented by:

$$\mathbf{D} = \begin{pmatrix} \beta & \gamma \\ 0 & 0 \end{pmatrix}$$

where β is an $(n_u \times n_u)$ matrix for direct effects of dependent variables on dependent variables and γ is an $(n_u \times n_w)$ matrix for direct effects of systematic independent variables on dependent variables. By definition, there should not be any direct effects on independent variables, and therefore the lower submatrices of $\mathbf{D}$ are null. In addition, by model restrictions the diagonal elements of matrix β must be zeros.

Correspondingly, the $(n_u + n_w) \times (n_u + n_w)$ matrix $\mathbf{T}$ of **total** effects of column variables on the row variables is computed by:

$$\mathbf{T} = \begin{pmatrix} (I - \beta)^{-1} - I & (I - \beta)^{-1}\gamma \\ 0 & 0 \end{pmatrix}$$

Finally, the $(n_u + n_w) \times (n_u + n_w)$ matrix **M** of **indirect** effects of column variables on the row variables is computed by the difference between **T** and **D** as:

$$\mathbf{M} = \begin{pmatrix} (I-\beta)^{-1} - I - \beta & (I-\beta)^{-1}\gamma - \gamma \\ 0 & 0 \end{pmatrix}$$

In PROC TCALIS, any subsets of **D**, **T**, and **M** can be requested via the specification in the EFFPART statement. All you need to do is to specify the sets of column variables (variables that have effects on others) and row variables (variables that receive the effects, direct or indirect). Specifications of the column and row variables are done conveniently by specifying variable names—no matrix terminology is needed. This feature is very handy if you have some focused subsets of effects that you want to analyze a priori. See the EFFPART statement on page 6751 for details about specifications.

Stability Coefficient of Reciprocal Causation

For recursive models (that is, models without cyclical paths of effects), using the preceding formulas for computing the total effect and the indirect effect is appropriate without further restrictions. However, for non-recursive models (that is, models with reciprocal effects or cyclical effects) the appropriateness of the preceding formulas for effect computations is restricted to situations with the convergence of the total effects.

A necessary and sufficient condition for the convergence of total effects (with or without cyclical paths) is when all eigenvalues, complex or real, of the $\boldsymbol{\beta}$ matrix fall into a unit circle (see Bentler and Freeman 1983). Equivalently, define the stability coefficient of reciprocal causation as the largest length (modulus) of the eigenvalues of the $\boldsymbol{\beta}$ matrix. A stability coefficient less than one would ensure that all eigenvalues, complex or real, of the $\boldsymbol{\beta}$ matrix fall into a unit circle. Hence, stability coefficient that is less than one is a necessary and sufficient condition for the convergence of the total effects, which in turn substantiates the appropriateness of total and indirect effect computations. Whenever effect analysis or partitioning is requested, PROC TCALIS will check the appropriateness of effect computations by evaluating the stability coefficient of reciprocal causation. If the stability coefficient is greater than one, computations of the total and indirect effects will not be done.

Standardized Solutions

Standardized solutions are useful when you want to compare parameter values that are measured on quite different scales. PROC TCALIS provides standardized solutions routinely. In standardizing a solution, parameters are classified into five groups:

- **path coefficients, regression coefficients, or direct effects**
  With each parameter α in this group, there is an associated outcome variable and a predictor variable. Denote the predicted variance of the outcome variable by σ_o^2 and the variance of the predictor variable by σ_p^2, the standardized parameter α^* is:

  $$\alpha^* = \alpha \frac{\sigma_p}{\sigma_o}$$

- **fixed ones for the path coefficients attached to error or disturbance terms**
  These fixed values are unchanged in standardization.

- **variances and covariances among exogenous variables, excluding errors and disturbances**
  Let σ_{ij} be the covariance between variables i and j. In this notation, σ_{ii} is the variance of variable i. The standardized covariance σ^*_{ij} is:

  $$\sigma^*_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}}$$

  When $i = j$, σ^*_{ii} takes the value of 1 for all i. Also, σ^*_{ij} is the correlation between the i-th and j-th variables.

- **variances and covariances among errors or disturbances**
  Denote the error covariance parameter as θ_{ij} so that θ_{ii} represents the variance parameter of error variable i. Associated with each error or disturbance variable i is a unique outcome variable. Let the variance of such an outcome variable be σ_{ii}. In the standardized solution, the error covariance θ_{ij} is rescaled as:

  $$\theta^*_{ij} = \frac{\theta_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}}$$

  Notice that when $i = j$, θ^*_{ii} is not standardized to 1 in general. In fact, the error (disturbance) variance is simply rescaled by the reciprocal of the variance of the associated dependent variable. As a result, the rescaled error (disturbance) variance represents the proportion of variation of the dependent variable due to the unsystematic source. By the same token, θ^*_{ij} does not represent the correlation between errors i and j. It is a rescaled covariance of the errors involved.

- **intercepts and means of variables**
  These parameters are fixed zeros in the standardized solution.

While formulas for the standardized solution are useful in computing the parameter values in the standardized solution, it is conceptually more useful to explain how variables are being transformed in the standardization process. The following provides a summary of the transformation process:

- Observed and latent variables, excluding errors or disturbances, are centered and then divided by their corresponding standard deviations. Therefore, in the standardized solution, all these variables will have variance equal to 1. In other words, these variables are truly standardized.

- Errors or disturbances are divided by the standard deviations of the corresponding outcome variables. In the standardized solution, these variables will not have variance equal to 1 in general. However, the rescaled error variances represent the proportion of unexplained or unsystematic variance of the corresponding outcome variables. Therefore, errors or disturbances in the standardized solution are simply rescaled but not standardized.

Standardized total, direct, and indirect effects are computed using formulas presented in the section "Total, Direct, and Indirect Effects" on page 6913, but with the standardized parameter values substituted into the formulas.

Although parameter values associated with different scales are made more comparable in the standardized solution, a precaution should be mentioned. In the standardized solution, the original constraints on parameters in the unstandardized solution are usually lost. These constraints, however, might underscore some important theoretical position that needs to be maintained in the model. Destroying these constraints in the standardized solution means that interpretations or comparisons of parameter values in the standardized solution are made without maintaining the original theoretical position. You must judge whether such a departure from the original constraints poses conceptual difficulties for interpreting the standardized solution.

Modification Indices

While fitting structural equation models is mostly a confirmatory analytic procedure, it does not prevent you from exploring what might have been a better model given the data. After fitting your theoretical structural equation model, you might want to modify the original model in order to do one of the following:

- add free parameters to improve the model fit significantly
- reduce the number of parameters without affecting the model fit too much

The first kind of model modification can be achieved by using the Lagrange multiplier (LM) test indices. Parameters that have the largest LM indices would increase the model fit the most. In general, adding more parameters to your model improves the overall model fit, as measured by those absolute or standalone fit indices (see the section "Overall Model Fit Indices" on page 6903 for more details). However, adding parameters liberally makes your model more prone to sampling errors. It also makes your model more complex and less interpretable in most cases. A disciplined use of LM test indices is highly recommended. In addition to the model fit improvement indicated by the LM test indices, you should also consider the theoretical significance when adding particular parameters. See Example 88.10 for an illustration of the use of LM test indices for improving model fit.

The second kind of model modification can be achieved by using the Wald statistics. Parameters that are not significant in your model may be removed from the model without affecting the model fit too much. In general, removing parameters from your model decreases the model fit, as measured by those absolute or standalone fit indices (see the section "Overall Model Fit Indices" on page 6903 for more details). However, for just a little sacrifice in model fit, removing non-significant parameters increases the simplicity and precision of your model, which is the virtue that any modeler should look for.

Whether adding parameters by using the LM test indices or removing unnecessary parameters by the Wald statistics, you should not treat your modified model as if it were your original hypothesized model. That is, you should not publish your modified model as if it were hypothesized a priori. It is perfectly fine to use modification indices to gain additional insights for future research. But if you want to publish your modified model together with your original model, you should report the modification process that leads to your modified model. Theoretical justifications of the modified model should be supplemented if you want to make strong statements to support your modified

model. Whenever possible, the best practice is to show reasonable model fit of the modified model with new data.

To modify your model either by LM test indices or Wald statistics, you can use the MODIFICATION or MOD option in the PROC TCALIS statement. To customize the LM tests by setting specific regions of parameters, you can use the LMTESTS statements. PROC TCALIS computes and displays the following default set of modification indices:

- **univariate Lagrange multiplier (LM) test indices for parameters in the model**
  These are second-order approximations of the decrease in the χ^2 value that would result from allowing the *fixed parameter values* in the model to be freed to estimate. LM test indices are ranked within their own parameter regions in the model. The ones that suggest biggest model improvements (that is, largest χ^2 drop) are ranked first. Depending on the type of your model, the set of possible parameter regions varies. For example, in a RAM model, modification indices are ranked in three different parameter regions for the covariance structures: path coefficients, variances of and covariances among exogenous variables, and the error variances and covariances. In addition to the value of the Lagrange multiplier, the corresponding p-value ($df = 1$) and the approximate change of the parameter value are displayed.

  If you use the LMMAT option in the LMTESTS statement, LM test indices are shown as elements in model matrices. Not all elements in a particular model matrix will have LM test indices. Elements that are already free parameters in the model do not have LM test indices. Instead, the parameter names are shown. Elements that are model restricted values (for example, direct path from a variable to itself must be zero) are labeled Excluded in the matrix output. When you customize your own regions of LM tests, some elements might also be excluded from a custom set of LM tests. These elements are also labeled as Excluded in the matrix output. If an LM test for freeing a parameter would result in a singular information matrix, the corresponding element in the matrix is labeled as Singular.

- **univariate Lagrange multiplier test indices for releasing equality constraints**
  These are second-order approximations of the decrease in the χ^2 value that would result from the release of *equality constraints*. Multiple equality constraints containing $n > 2$ parameters are tested successively in n steps, each assuming the release of one of the equality-constrained parameters. The expected change of the parameter values of the separated parameter and the remaining parameter cluster are displayed, too.

- **univariate Lagrange multiplier test indices for releasing active boundary constraints**
  These are second-order approximations of the decrease in the χ^2 value that would result from the release of the *active boundary constraints* specified in the BOUNDS statement.

- **stepwise multivariate Wald statistics for constraining free parameters to 0**
  These are second-order approximations of the increases in χ^2 value that would result from constraining free parameters to zero in a stepwise fashion. In each step, the parameter that would lead to the smallest increase in the multivariate χ^2 value is set to 0. Besides the multivariate χ^2 value and its p-value, the univariate increments are also displayed. The process stops when the univariate p-value is smaller than the specified value in the SLMW= option, of which the default value is 0.05.

All of the preceding tests are approximations. You can often obtain more accurate tests by actually fitting different models and computing likelihood ratio tests. For more details about the Wald and the Lagrange multiplier test, refer to MacCallum (1986), Buse (1982), Bentler (1986), or Lee (1985). Note that relying solely on the LM tests to modify your model can lead to unreliable models that capitalize purely on sampling errors. See MacCallum, Roznowski, and Necowitz (1992) for the use of LM tests.

For large model matrices, the computation time for the default modification indices can considerably exceed the time needed for the minimization process.

The modification indices are not computed for unweighted least squares or diagonally weighted least squares estimation.

Missing Values

If the DATA= data set contains raw data (rather than a covariance or correlation matrix), observations with missing values for any variables in the analysis are omitted from the computations. If a covariance or correlation matrix is read, missing values are allowed as long as every pair of variables has at least one nonmissing value.

Measures of Multivariate Kurtosis

In many applications, the manifest variables are not even approximately multivariate normal. If this happens to be the case with your data set, the default generalized least squares and maximum likelihood estimation methods are not appropriate, and you should compute the parameter estimates and their standard errors by an asymptotically distribution-free method, such as the WLS estimation method. If your manifest variables are multivariate normal, then they have a zero relative multivariate kurtosis, and all marginal distributions have zero kurtosis (Browne 1982). If your DATA= data set contains raw data, PROC TCALIS computes univariate skewness and kurtosis and a set of multivariate kurtosis values. By default, the values of univariate skewness and kurtosis are corrected for bias (as in PROC UNIVARIATE), but using the BIASKUR option enables you to compute the uncorrected values also. The values are displayed when you specify the PROC TCALIS statement option KURTOSIS.

In the following formulas, N denotes the sample size and p denotes the number of variables.

- **corrected variance for variable** z_j

$$\sigma_j^2 = \frac{1}{N-1}\sum_{i}^{N}(z_{ij} - \bar{z}_j)^2$$

- **uncorrected univariate skewness for variable** z_j

$$\gamma_{1(j)} = \frac{N \sum_i^N (z_{ij} - \bar{z}_j)^3}{\sqrt{N[\sum_i^N (z_{ij} - \bar{z}_j)^2]^3}}$$

- **corrected univariate skewness for variable** z_j

$$\gamma_{1(j)} = \frac{N}{(N-1)(N-2)} \frac{\sum_i^N (z_{ij} - \bar{z}_j)^3}{\sigma_j^3}$$

- **uncorrected univariate kurtosis for variable** z_j

$$\gamma_{2(j)} = \frac{N \sum_i^N (z_{ij} - \bar{z}_j)^4}{[\sum_i^N (z_{ij} - \bar{z}_j)^2]^2} - 3$$

- **corrected univariate kurtosis for variable** z_j

$$\gamma_{2(j)} = \frac{N(N+1)}{(N-1)(N-2)(N-3)} \frac{\sum_i^N (z_{ij} - \bar{z}_j)^4}{\sigma_j^4} - \frac{3(N-1)^2}{(N-2)(N-3)}$$

- **Mardia's multivariate kurtosis**

$$\gamma_2 = \frac{1}{N} \sum_i^N [(z_i - \bar{z})' S^{-1} (z_i - \bar{z})]^2 - p(p+2)$$

- **relative multivariate kurtosis**

$$\eta_2 = \frac{\gamma_2 + p(p+2)}{p(p+2)}$$

- **normalized multivariate kurtosis**

$$\kappa_0 = \frac{\gamma_2}{\sqrt{8p(p+2)/N}}$$

- **Mardia based kappa**

$$\kappa_1 = \frac{\gamma_2}{p(p+2)}$$

- **mean scaled univariate kurtosis**

$$\kappa_2 = \frac{1}{3p} \sum_j^p \gamma_{2(j)}$$

- **adjusted mean scaled univariate kurtosis**

$$\kappa_3 = \frac{1}{3p}\sum_{j}^{p}\gamma^*_{2(j)}$$

with

$$\gamma^*_{2(j)} = \begin{cases} \gamma_{2(j)} & , \quad \text{if} \quad \gamma_{2(j)} > \frac{-6}{p+2} \\ \frac{-6}{p+2} & , \quad \text{otherwise} \end{cases}$$

If variable Z_j is normally distributed, the uncorrected univariate kurtosis $\gamma_{2(j)}$ is equal to 0. If Z has an p-variate normal distribution, Mardia's multivariate kurtosis γ_2 is equal to 0. A variable Z_j is called *leptokurtic* if it has a positive value of $\gamma_{2(j)}$ and is called *platykurtic* if it has a negative value of $\gamma_{2(j)}$. The values of κ_1, κ_2, and κ_3 should not be smaller than the following lower bound (Bentler 1985):

$$\hat{\kappa} \geq \frac{-2}{p+2}$$

PROC TCALIS displays a message if κ_1, κ_2, or κ_3 falls below the lower bound.

If weighted least squares estimates (METHOD=WLS or METHOD=ADF) are specified and the weight matrix is computed from an input raw data set, the TCALIS procedure computes two more measures of multivariate kurtosis.

- **multivariate mean kappa**

$$\kappa_4 = \frac{1}{m}\sum_{i}^{p}\sum_{j}^{i}\sum_{k}^{j}\sum_{l}^{k}\hat{\kappa}_{ij,kl} - 1$$

where

$$\hat{\kappa}_{ij,kl} = \frac{s_{ij,kl}}{s_{ij}s_{kl} + s_{ik}s_{jl} + s_{il}s_{jk}}$$

and $m = p(p+1)(p+2)(p+3)/24$ is the number of elements in the vector $s_{ij,kl}$ (Bentler 1985).

- **multivariate least squares kappa**

$$\kappa_5 = \frac{s_4' s_2}{s_2' s_2} - 1$$

where s_2 is the vector of the elements in the denominator of $\hat{\kappa}$ (Bentler 1985) and s_4 is the vector of the $s_{ij,kl}$, which is defined as:

$$s_{ij,kl} = \frac{1}{N}\sum_{r=1}^{N}(z_{ri} - \bar{z}_i)(z_{rj} - \bar{z}_j)(z_{rk} - \bar{z}_k)(z_{rl} - \bar{z}_l)$$

The occurrence of significant nonzero values of Mardia's multivariate kurtosis γ_2 and significant amounts of some of the univariate kurtosis values $\gamma_{2(j)}$ indicate that your variables are not multivariate normal distributed. Violating the multivariate normality assumption in (default) generalized least squares and maximum likelihood estimation usually leads to the wrong approximate standard errors and incorrect fit statistics based on the χ^2 value. In general, the parameter estimates are more stable against violation of the normal distribution assumption. For more details, refer to Browne (1974, 1982, 1984).

Initial Estimates

Each optimization technique requires a set of initial values for the parameters. To avoid local optima, the initial values should be as close as possible to the globally optimal solution. You can check for local optima by running the analysis with several different sets of initial values; the RANDOM= option in the PROC TCALIS statement is useful in this regard.

Except for the case of exploratory FACTOR model, you can specify initial estimates manually for all different types of models. If you do not specify some of the initial estimates and the RANDOM= option is not used, PROC TCALIS will use a combination of good strategic methods to compute initial estimates for your model.

These initial estimation methods are used in PROC TCALIS:

- two-stage least squares estimation
- instrumental variable method (Hägglund 1982; Jennrich 1987)
- approximate factor analysis method
- ordinary least squares estimation
- estimation method of McDonald (McDonald and Hartmann 1992)
- observed moments of manifest exogenous variables

The choice of initial estimation methods is dependent on the data and on the model. In general, it is difficult to tell in advance which initial estimation methods will be used for a given analysis. However, PROC TCALIS displays the methods used to obtain initial estimates in the output.

Poor initial values can cause convergence problems, especially with maximum likelihood estimation. Sufficiently large positive initial values for variance estimates (as compared with the covariance estimates) might help prevent a nonnegative definite initial predicted covariance model matrix from happening. If maximum likelihood estimation fails to converge, it might help to use METHOD=LSML, which uses the final estimates from an unweighted least squares analysis as initial estimates for maximum likelihood. Or you can fit a slightly different but better-behaved model and produce an OUTMODEL= data set, which can then be modified in accordance with the original model and used as an INMODEL= data set to provide initial values for another analysis.

If you are analyzing a covariance or scalar product matrix, be sure to take into account the scales of the variables. The default initial values might be inappropriate when some variables have extremely large or small variances.

Use of Optimization Techniques

No algorithm for optimizing general nonlinear functions exists that can always find the global optimum for a general nonlinear minimization problem in a reasonable amount of time. Since no single optimization technique is invariably superior to others, PROC TCALIS provides a variety of optimization techniques that work well in various circumstances. However, you can devise problems for which none of the techniques in PROC TCALIS can find the correct solution. All optimization techniques in PROC TCALIS use $O(n^2)$ memory except the conjugate gradient methods, which use only $O(n)$ of memory and are designed to optimize problems with many parameters.

The PROC TCALIS statement NLOPTIONS can be especially helpful for tuning applications with nonlinear equality and inequality constraints on the parameter estimates. Some of the options available in NLOPTIONS can also be invoked as PROC TCALIS options. The NLOPTIONS statement can specify almost the same options as the SAS/OR NLP procedure.

Nonlinear optimization requires the repeated computation of the following:

- the function value (optimization criterion)
- the gradient vector (first-order partial derivatives)
- for some techniques, the (approximate) Hessian matrix (second-order partial derivatives)
- values of linear and nonlinear constraints
- the first-order partial derivatives (Jacobian) of nonlinear constraints

For the criteria used by PROC TCALIS, computing the gradient takes more computer time than computing the function value, and computing the Hessian takes *much* more computer time and memory than computing the gradient, especially when there are many parameters to estimate. Unfortunately, optimization techniques that do not use the Hessian usually require many more iterations than techniques that do use the (approximate) Hessian, and so they are often slower. Techniques that do not use the Hessian also tend to be less reliable (for example, they might terminate at local rather than global optima).

The available optimization techniques are displayed in the following table and can be chosen by the OMETHOD=$name$ option.

OMETHOD=	Optimization Technique
LEVMAR	Levenberg-Marquardt method
TRUREG	trust-region method
NEWRAP	Newton-Raphson method with line search
NRRIDG	Newton-Raphson method with ridging
QUANEW	quasi-Newton methods (DBFGS, DDFP, BFGS, DFP)
DBLDOG	double-dogleg method (DBFGS, DDFP)
CONGRA	conjugate gradient methods (PB, FR, PR, CD)

The following table shows, for each optimization technique, which derivatives are needed (first-order or second-order) and what kind of constraints (boundary, linear, or nonlinear) can be imposed on the parameters.

	Derivatives		Constraints		
OMETHOD=	First Order	Second Order	Boundary	Linear	Nonlinear
LEVMAR	x	x	x	x	-
TRUREG	x	x	x	x	-
NEWRAP	x	x	x	x	-
NRRIDG	x	x	x	x	-
QUANEW	x	-	x	x	x
DBLDOG	x	-	x	x	-
CONGRA	x	-	x	x	-

The Levenberg-Marquardt, trust-region, and Newton-Raphson techniques are usually the most reliable, work well with boundary and general linear constraints, and generally converge after a few iterations to a precise solution. However, these techniques need to compute a Hessian matrix in each iteration. Computing the approximate Hessian in each iteration can be very time- and memory-consuming, especially for large problems (more than 60 or 100 parameters, depending on the computer used). For large problems, a quasi-Newton technique, especially with the BFGS update, can be far more efficient.

For a poor choice of initial values, the Levenberg-Marquardt method seems to be more reliable.

If memory problems occur, you can use one of the conjugate gradient techniques, but they are generally slower and less reliable than the methods that use second-order information.

There are several options to control the optimization process. You can specify various termination criteria. You can specify the GCONV= option to specify a relative gradient termination criterion. If there are active boundary constraints, only those gradient components that correspond to inactive constraints contribute to the criterion. When you want very precise parameter estimates, the GCONV= option is useful. Other criteria that use relative changes in function values or parameter estimates in consecutive iterations can lead to early termination when active constraints cause small steps to occur. The small default value for the FCONV= option helps prevent early termination. Using the MAXITER= and MAXFUNC= options enables you to specify the maximum number of iterations and function calls in the optimization process. These limits are especially useful in combination with the INMODEL= and OUTMODEL= options; you can run a few iterations at a time, inspect the results, and decide whether to continue iterating.

Nonlinearly Constrained QN Optimization

The algorithm used for nonlinearly constrained quasi-Newton optimization is an efficient modification of Powell's (1978a, 1978b, 1982a, 1982b) Variable Metric Constrained WatchDog (VMCWD) algorithm. A similar but older algorithm (VF02AD) is part of the Harwell library. Both VMCWD and VF02AD use Fletcher's VE02AD algorithm (also part of the Harwell library) for positive definite quadratic programming. The PROC TCALIS QUANEW implementation uses a quadratic programming subroutine that updates and downdates the approximation of the Cholesky factor when the active set changes. The nonlinear QUANEW algorithm is not a feasible point algorithm, and the value of the objective function might not necessarily decrease (minimization) or increase (maximization) monotonically. Instead, the algorithm tries to reduce a linear combination of the objective function and constraint violations, called the *merit function*.

The following are similarities and differences between this algorithm and VMCWD:

- A modification of this algorithm can be performed by specifying VERSION=1, which replaces the update of the Lagrange vector μ with the original update of Powell (1978a, 1978b), which is used in VF02AD. This can be helpful for some applications with linearly dependent active constraints.
- If the VERSION= option is not specified or VERSION=2 is specified, the evaluation of the Lagrange vector μ is performed in the same way as Powell (1982a, 1982b) describes.
- Instead of updating an approximate Hessian matrix, this algorithm uses the dual BFGS (or DFP) update that updates the Cholesky factor of an approximate Hessian. If the condition of the updated matrix gets too bad, a restart is done with a positive diagonal matrix. At the end of the first iteration after each restart, the Cholesky factor is scaled.
- The Cholesky factor is loaded into the quadratic programming subroutine, automatically ensuring positive definiteness of the problem. During the quadratic programming step, the Cholesky factor of the projected Hessian matrix $\mathbf{Z}_k'\mathbf{G}\mathbf{Z}_k$ and the QT decomposition are updated simultaneously when the active set changes. Refer to Gill et al. (1984) for more information.
- The line-search strategy is very similar to that of Powell (1982a, 1982b). However, this algorithm does not call for derivatives during the line search; hence, it generally needs fewer derivative calls than function calls. The VMCWD algorithm always requires the same number of derivative and function calls. It was also found in several applications of VMCWD that Powell's line-search method sometimes uses steps that are too long during the first iterations. In those cases, you can use the INSTEP= option specification to restrict the step length α of the first iterations.
- The watchdog strategy is similar to that of Powell (1982a, 1982b). However, this algorithm does not return automatically after a fixed number of iterations to a former better point. A return here is further delayed if the observed function reduction is close to the expected function reduction of the quadratic model.
- Although Powell's termination criterion still is used (as FCONV2), the QUANEW implementation uses two additional termination criteria (GCONV and ABSGCONV).

This algorithm is automatically invoked when you specify the NLINCON statement. The nonlinear QUANEW algorithm needs the Jacobian matrix of the first-order derivatives (constraints normals) of the constraints:

$$(\nabla c_i) = (\frac{\partial c_i}{\partial x_j}), \quad i = 1, \ldots, nc, j = 1, \ldots, n$$

where nc is the number of nonlinear constraints for a given point x.

You can specify two update formulas with the UPDATE= option:

- UPDATE=DBFGS performs the dual BFGS update of the Cholesky factor of the Hessian matrix. This is the default.
- UPDATE=DDFP performs the dual DFP update of the Cholesky factor of the Hessian matrix.

This algorithm uses its own line-search technique. All options and parameters (except the INSTEP= option) controlling the line search in the other algorithms do not apply here. In several applications, large steps in the first iterations are troublesome. You can specify the INSTEP= option to impose an upper bound for the step size α during the first five iterations. The values of the LCSINGULAR=, LCEPSILON=, and LCDEACT= options (which control the processing of linear and boundary constraints) are valid only for the quadratic programming subroutine used in each iteration of the nonlinear constraints QUANEW algorithm.

Optimization and Iteration History

The optimization and iteration histories are displayed by default because it is important to check for possible convergence problems. The optimization history includes the following summary of information about the initial state of the optimization:

- the number of constraints that are active at the starting point, or more precisely, the number of constraints that are currently members of the working set. If this number is followed by a plus sign, there are more active constraints, of which at least one is temporarily released from the working set due to negative Lagrange multipliers.
- the value of the objective function at the starting point
- if the (projected) gradient is available, the value of the largest absolute (projected) gradient element
- for the TRUREG and LEVMAR subroutines, the initial radius of the trust region around the starting point

The optimization history ends with some information concerning the optimization result:

- the number of constraints that are active at the final point, or more precisely, the number of constraints that are currently members of the working set. If this number is followed by a plus sign, there are more active constraints, of which at least one is temporarily released from the working set due to negative Lagrange multipliers.
- the value of the objective function at the final point
- if the (projected) gradient is available, the value of the largest absolute (projected) gradient element
- other information specific to the optimization technique

The iteration history generally consists of one line of displayed output containing the most important information for each iteration.

The iteration history always includes the following:

- the iteration number
- the number of iteration restarts
- the number of function calls
- the number of active constraints
- the value of the optimization criterion
- the difference between adjacent function values
- the maximum of the absolute gradient components that correspond to inactive boundary constraints

An apostrophe trailing the number of active constraints indicates that at least one of the active constraints is released from the active set due to a significant Lagrange multiplier.

For the Levenberg-Marquardt technique (LEVMAR), the iteration history also includes the following information:

- an asterisk trailing the iteration number when the computed Hessian approximation is singular and consequently ridged with a positive lambda value. If all or the last several iterations show a singular Hessian approximation, the problem is not sufficiently identified. Thus, there are other locally optimal solutions that lead to the same optimum function value for different parameter values. This implies that standard errors for the parameter estimates are not computable without the addition of further constraints.
- the value of the Lagrange multiplier (lambda). This value is 0 when the optimum of the quadratic function approximation is inside the trust region (a trust-region-scaled Newton step

can be performed) and is greater than 0 when the optimum of the quadratic function approximation is located at the boundary of the trust region (the scaled Newton step is too long to fit in the trust region and a quadratic constraint optimization is performed). Large values indicate optimization difficulties. For a nonsingular Hessian matrix, the value of lambda should go to 0 during the last iterations, indicating that the objective function can be well approximated by a quadratic function in a small neighborhood of the optimum point. An increasing lambda value often indicates problems in the optimization process.

- the value of the ratio ρ (rho) between the actually achieved difference in function values and the predicted difference in the function values on the basis of the quadratic function approximation. Values much less than 1 indicate optimization difficulties. The value of the ratio ρ indicates the goodness of the quadratic function approximation. In other words, $\rho << 1$ means that the radius of the trust region has to be reduced; a fairly large value of ρ means that the radius of the trust region does not need to be changed. And a value close to or larger than 1 means that the radius can be increased, indicating a good quadratic function approximation.

For the Newton-Raphson technique (NRRIDG), the iteration history also includes the following information:

- the value of the ridge parameter. This value is 0 when a Newton step can be performed, and it is greater than 0 when either the Hessian approximation is singular or a Newton step fails to reduce the optimization criterion. Large values indicate optimization difficulties.
- the value of the ratio ρ (rho) between the actually achieved difference in function values and the predicted difference in the function values on the basis of the quadratic function approximation. Values much less than 1.0 indicate optimization difficulties.

For the Newton-Raphson with line-search technique (NEWRAP), the iteration history also includes the following information:

- the step size α (alpha) computed with one of the line-search algorithms
- the slope of the search direction at the current parameter iterate. For minimization, this value should be significantly negative. Otherwise, the line-search algorithm has difficulty reducing the function value sufficiently.

For the trust-region technique (TRUREG), the iteration history also includes the following information:

- an asterisk after the iteration number when the computed Hessian approximation is singular and consequently ridged with a positive lambda value.
- the value of the Lagrange multiplier (lambda). This value is zero when the optimum of the quadratic function approximation is inside the trust region (a trust-region-scaled Newton step can be performed) and is greater than zero when the optimum of the quadratic function approximation is located at the boundary of the trust region (the scaled Newton step is too

long to fit in the trust region and a quadratically constrained optimization is performed). Large values indicate optimization difficulties. As in Gay (1983), a negative lambda value indicates the special case of an indefinite Hessian matrix (the smallest eigenvalue is negative in minimization).

- the value of the radius Δ of the trust region. Small trust-region radius values combined with large lambda values in subsequent iterations indicate optimization problems.

For the quasi-Newton (QUANEW) and conjugate gradient (CONGRA) techniques, the iteration history also includes the following information:

- the step size (alpha) computed with one of the line-search algorithms
- the descent of the search direction at the current parameter iterate. This value should be significantly smaller than 0. Otherwise, the line-search algorithm has difficulty reducing the function value sufficiently.

Frequent update restarts (rest) of a quasi-Newton algorithm often indicate numerical problems related to required properties of the approximate Hessian update, and they decrease the speed of convergence. This can happen particularly if the ABSGCONV= termination criterion is too small—that is, when the requested precision cannot be obtained by quasi-Newton optimization. Generally, the number of automatic restarts used by conjugate gradient methods are much higher.

For the nonlinearly constrained quasi-Newton technique, the iteration history also includes the following information:

- the maximum value of all constraint violations,

$$\text{conmax} = \max(|c_i(x)| : c_i(x) < 0)$$

- the value of the predicted function reduction used with the GCONV and FCONV2 termination criteria,

$$\text{pred} = |g(x^{(k)})s(x^{(k)})| + \sum_{i=1}^{m} |\lambda_i c_i(x^{(k)})|$$

- the step size α of the quasi-Newton step. Note that this algorithm works with a special line-search algorithm.
- the maximum element of the gradient of the Lagrange function,

$$\begin{aligned} \text{lfgmax} &= \nabla_x L(x^{(k)}, \lambda^{(k)}) \\ &= \nabla_x f(x^{(k)}) - \sum_{i=1}^{m} \lambda_i^{(k)} \nabla_x c_i(x^{(k)}) \end{aligned}$$

For the double dogleg technique, the iteration history also includes the following information:

- the parameter λ of the double-dogleg step. A value $\lambda = 0$ corresponds to the full (quasi) Newton step.
- the slope of the search direction at the current parameter iterate. For minimization, this value should be significantly negative.

Line-Search Methods

In each iteration k, the (dual) quasi-Newton, hybrid quasi-Newton, conjugate gradient, and Newton-Raphson minimization techniques use iterative line-search algorithms that try to optimize a linear, quadratic, or cubic approximation of the nonlinear objective function f of n parameters x along a feasible descent search direction $s^{(k)}$ as follows:

$$f(x^{(k+1)}) = f(x^{(k)} + \alpha^{(k)} s^{(k)})$$

by computing an approximately optimal scalar $\alpha^{(k)} > 0$. Since the outside iteration process is based only on the approximation of the objective function, the inside iteration of the line-search algorithm does not have to be perfect. Usually, it is satisfactory that the choice of α significantly reduces (in a minimization) the objective function. Criteria often used for termination of line-search algorithms are the Goldstein conditions (Fletcher 1987).

Various line-search algorithms can be selected by using the LIS= option on page 6731. The line-search methods LIS=1, LIS=2, and LIS=3 satisfy the left-hand-side and right-hand-side Goldstein conditions (Fletcher 1987). When derivatives are available, the line-search methods LIS=6, LIS=7, and LIS=8 try to satisfy the right-hand-side Goldstein condition; if derivatives are not available, these line-search algorithms use only function calls.

The line-search method LIS=2 seems to be superior when function evaluation consumes significantly less computation time than gradient evaluation. Therefore, LIS=2 is the default value for Newton-Raphson, (dual) quasi-Newton, and conjugate gradient optimizations.

Restricting the Step Length

Almost all line-search algorithms use iterative extrapolation techniques that can easily lead to feasible points where the objective function f is no longer defined (resulting in indefinite matrices for ML estimation) or is difficult to compute (resulting in floating point overflows). Therefore, PROC TCALIS provides options that restrict the step length or trust region radius, especially during the first main iterations.

The inner product $g's$ of the gradient g and the search direction s is the slope of $f(\alpha) = f(x + \alpha s)$ along the search direction s with step length α. The default starting value $\alpha^{(0)} = \alpha^{(k,0)}$ in each line-search algorithm ($\min_{\alpha>0} f(x + \alpha s)$) during the main iteration k is computed in three steps:

1. Use either the difference $df = |f^{(k)} - f^{(k-1)}|$ of the function values during the last two consecutive iterations or the final stepsize value α^- of the previous iteration $k-1$ to compute a first value $\alpha_1^{(0)}$.

- Using the DAMPSTEP<r> option:

$$\alpha_1^{(0)} = \min(1, r\alpha^-)$$

The initial value for the new step length can be no larger than r times the final step length α^- of the previous iteration. The default is $r = 2$.

- Not using the DAMPSTEP option:

$$\alpha_1^{(0)} = \begin{cases} step & \text{if } 0.1 \leq step \leq 10 \\ 10 & \text{if } step > 10 \\ 0.1 & \text{if } step < 0.1 \end{cases}$$

with

$$step = \begin{cases} df/|g's| & \text{if } |g's| \geq \epsilon \max(100df, 1) \\ 1 & \text{otherwise} \end{cases}$$

This value of $\alpha_1^{(0)}$ can be too large and can lead to a difficult or impossible function evaluation, especially for highly nonlinear functions such as the EXP function.

2. During the first five iterations, the second step enables you to reduce $\alpha_1^{(0)}$ to a smaller starting value $\alpha_2^{(0)}$ using the INSTEP=r option:

$$\alpha_2^{(0)} = \min(\alpha_1^{(0)}, r)$$

After more than five iterations, $\alpha_2^{(0)}$ is set to $\alpha_1^{(0)}$.

3. The third step can further reduce the step length by

$$\alpha_3^{(0)} = \min(\alpha_2^{(0)}, \min(10, u))$$

where u is the maximum length of a step inside the feasible region.

The INSTEP=r option lets you specify a smaller or larger radius of the trust region used in the first iteration by the trust-region, double-dogleg, and Levenberg-Marquardt algorithms. The default initial trust region radius is the length of the scaled gradient (Moré 1978). This default length for the initial trust region radius corresponds to the default radius factor of $r = 1$. This choice is successful in most practical applications of the TRUREG, DBLDOG, and LEVMAR algorithms. However, for bad initial values used in the analysis of a covariance matrix with high variances or for highly nonlinear constraints (such as using the EXP function) in your SAS programming statements, the default start radius can result in arithmetic overflows. If this happens, you can try decreasing values of INSTEP=r ($0 < r < 1$), until the iteration starts successfully. A small factor r also affects the trust region radius of the next steps because the radius is changed in each iteration by a factor $0 < c \leq 4$ depending on the ρ ratio. Reducing the radius corresponds to increasing the ridge parameter λ that produces smaller steps directed closer toward the gradient direction.

Computational Problems

First Iteration Overflows

Analyzing a covariance matrix that includes high variances in the diagonal and using bad initial estimates for the parameters can easily lead to arithmetic overflows in the first iterations of the minimization algorithm. The line-search algorithms that work with cubic extrapolation are especially sensitive to arithmetic overflows. If this occurs with quasi-Newton or conjugate gradient minimization, you can specify the INSTEP= option to reduce the length of the first step. If an arithmetic overflow occurs in the first iteration of the Levenberg-Marquardt algorithm, you can specify the INSTEP= option to reduce the trust region radius of the first iteration. You also can change the minimization technique or the line-search method. If none of these help, you can consider doing the following:

- scaling the covariance matrix
- providing better initial values
- changing the model

No Convergence of Minimization Process

If convergence does not occur during the minimization process, perform the following tasks:

- If there are *negative variance estimates*, you can do either of the following:
  - Specify the BOUNDS statement to obtain nonnegative variance estimates.
  - Specify the HEYWOOD option, if the FACTOR statement is specified.
- Change the estimation method to obtain a better set of initial estimates. For example, if you use METHOD=ML, you can do either of the following:
  - Change to METHOD=LSML.
  - Run some iterations with METHOD=DWLS or METHOD=GLS, write the results in an OUTMODEL= data set, and use the results as initial values specified by an INMODEL= data set in a second run with METHOD=ML.
- Change the optimization technique. For example, if you use the default OMETHOD=LEVMAR, you can do either of the following:
  - Change to OMETHOD=QUANEW or to OMETHOD=NEWRAP.
  - Run some iterations with OMETHOD=CONGRA, write the results in an OUTMODEL= data set, and use the results as initial values specified by an INMODEL= data set in a second run with a different OMETHOD= technique.

- Change or modify the update technique or the line-search algorithm or both when using OMETHOD=QUANEW or OMETHOD=CONGRA. For example, if you use the default update formula and the default line-search algorithm, you can do any or all of the following:
  - Change the update formula with the UPDATE= option.
  - Change the line-search algorithm with the LIS= option.
  - Specify a more precise line search with the LSPRECISION= option, if you use LIS=2 or LIS=3.
- Add more iterations and function calls by using the MAXIT= and MAXFU= options.
- Change the initial values. For many categories of model specifications, PROC TCALIS computes an appropriate set of initial values automatically. However, for some of the model specifications (for example, structural equations with latent variables on the left-hand side and manifest variables on the right-hand side), PROC TCALIS might generate very obscure initial values. In these cases, you have to set the initial values yourself.
  - Increase the initial values of the variance parameters by one of the following ways:
    - Set the variance parameter values in the model specification manually.
    - Use the DEMPHAS= option to increase all initial variance parameter values.
  - Use a slightly different, but more stable, model to obtain preliminary estimates.
  - Use additional information to specify initial values, for example, by using other SAS software like the FACTOR, REG, SYSLIN, and MODEL (SYSNLIN) procedures for the modified, unrestricted model case.

Unidentified Model

The parameter vector $\boldsymbol{\Theta}$ in the structural model

$$\boldsymbol{\Sigma} = \boldsymbol{\Sigma}(\boldsymbol{\Theta})$$

is said to be identified in a parameter space G, if

$$\boldsymbol{\Sigma}(\boldsymbol{\Theta}) = \boldsymbol{\Sigma}(\tilde{\boldsymbol{\Theta}}), \quad \tilde{\boldsymbol{\Theta}} \in G$$

implies $\boldsymbol{\Theta} = \tilde{\boldsymbol{\Theta}}$. The parameter estimates that result from an unidentified model can be very far from the parameter estimates of a very similar but identified model. They are usually machine dependent. Do not use parameter estimates of an unidentified model as initial values for another run of PROC TCALIS.

Singular Predicted Covariance Model Matrix

Sometimes you might inadvertently specify models with singular predicted covariance model matrices (for example, by fixing diagonal elements to zero). In such cases, you cannot compute maximum likelihood estimates (the ML function value F is not defined). Since singular predicted covariance model matrices can also occur temporarily in the minimization process, PROC TCALIS tries in such

cases to change the parameter estimates so that the predicted covariance model matrix becomes positive definite. This process does not always work well, especially if there are fixed instead of free diagonal elements in the predicted covariance model matrices. A famous example where you cannot compute ML estimates is a component analysis with fewer components than given manifest variables. See the section "FACTOR Statement" on page 6752 for more details. If you continue to obtain a singular predicted covariance model matrix after changing initial values and optimization techniques, then your model might be specified so that ML estimates cannot be computed.

Saving Computing Time

For large models, the most computing time is needed to compute the modification indices. If you do not really need the Lagrange multipliers or multiple Wald test indices (the univariate Wald test indices are the same as the t values), using the NOMOD option can save a considerable amount of computing time.

Predicted Covariance Matrices with Negative Eigenvalues

A covariance matrix cannot have negative eigenvalues, since a negative eigenvalue means that some linear combination of the variables has negative variance. PROC TCALIS displays a warning if the predicted covariance matrix has negative eigenvalues but does not actually compute the eigenvalues. Sometimes this warning can be triggered by 0 or very small positive eigenvalues that appear negative because of numerical error. If you want to be sure that the predicted covariance matrix you are fitting can be considered to be a variance-covariance matrix, you can use the SAS/IML command *VAL=EIGVAL(U)* to compute the vector *VAL* of eigenvalues of matrix **U**.

Negative R^2 Values

The estimated squared multiple correlations R^2 of the endogenous variables are computed using the estimated error variances:

$$R_i^2 = 1 - \frac{\widehat{var(\zeta_i)}}{\widehat{var(\eta_i)}}$$

When $\widehat{var(\zeta_i)} > \widehat{var(\eta_i)}$, R_i^2 is negative. This might indicate poor model fit or R^2 is an inappropriate measure for the model. For the latter case, for example, negative R^2 might be due to cyclical (nonrecursive) paths in the model so that the R^2 interpretation is not appropriate.

Displayed Output

The output of PROC TCALIS includes the following:

- a list of basic modeling information such as: the data set, the number of records read and used in the raw data set, the number of observations assumed by the statistical analysis, and the model type. When a multiple-group analysis is specified, the groups and their corresponding models are listed. This output assumes at least the PSHORT option.
- a list of all variables in the models. This output is displayed by default or by the PINITIAL option. It will not be displayed when you use the PSHORT or the PSUMMARY option.

  Depending on the modeling language, the variable lists vary, as shown in the following:

  – FACTOR: a list of the variables and the factors
  – LINEQS, PATH, and RAM: a list of the endogenous and exogenous variables specified in the model
  – LISMOD: a list of x-, y-, ξ-, and η- variables specified in the model
  – MSTRUCT: a list of the manifest variables specified in the model
- initial model specification. This output is displayed by default or by the PINITIAL option. It will not be displayed when you use the PSHORT or the PSUMMARY option.

  Depending on the modeling language, the sets of output vary, as shown in the following:

  – FACTOR: factor loading matrix, factor covariance matrix, intercepts, factor means, and error variances as specified initially in the model. The initial values for free parameters, the fixed values, and the parameter names are also displayed.
  – LINEQS: linear equations, variance and covariance parameters, and mean parameters as specified initially in the model. The initial values for free parameters, the fixed values, and the parameter names are also displayed.
  – LISMOD: all model matrices as specified initially in the model. The initial values for free parameters, the fixed values, and the parameter names are also displayed.
  – MSTRUCT: initial covariance matrix and mean vectors, with parameter names and initial values displayed.
  – PATH: the path list, variance and covariance parameters, intercept and mean parameters as specified initially in the model. The initial values for free parameters, the fixed values, and the parameter names are also displayed.
  – RAM: a list of parameters, their types, names, and initial values.
- mean and standard deviation of each manifest variable if you specify the SIMPLE option, as well as skewness and kurtosis if the DATA= data set is a raw data set and you specify the KURTOSIS option.

- various coefficients of multivariate kurtosis and the numbers of observations that contribute most to the normalized multivariate kurtosis if the DATA= data set is a raw data set and the KURTOSIS option is used or you specify at least the PRINT option. See the section "Measures of Multivariate Kurtosis" on page 6919 for more information.

- covariance or correlation matrix to be analyzed and the value of its determinant if you specify the output option PCORR or PALL. A zero determinant indicates a singular data matrix. In this case, the generalized least squares estimates with default weight matrix **S** and maximum likelihood estimates cannot be computed.

- the weight matrix **W** or its inverse is displayed if GLS, WLS, or DWLS estimation is used and you specify the PWEIGHT or PALL option.

- initial estimation methods for generating initial estimates. This output is displayed by default. It will not be displayed when you use the PSHORT or the PSUMMARY option.

- vector of parameter names and initial values and gradients. This output is displayed by default, unless you specify the PSUMMARY or NOPRINT option.

- special features of the optimization technique chosen if you specify at least the PSHORT option.

- optimization history if at least the PSHORT option is specified. For more details, see the section "Use of Optimization Techniques" on page 6923.

- specific output requested by options in the NLOPTIONS statement; for example, parameter estimates, gradient, constraints, projected gradient, Hessian, projected Hessian, Jacobian of nonlinear constraints, estimated covariance matrix of parameter estimates, and information matrix. Note that the estimated covariance of parameter estimates and the information matrix are not printed for the ULS and DWLS estimation methods.

- fit summary table with various model fit test statistics or fit indices, and some basic modeling information. For the listing of fit indices and their definitions, see the section "Overall Model Fit Indices" on page 6903. Note that for ULS and DWLS estimation methods, many of those fit indices that are based on model fit χ^2 are not displayed. See the section "Overall Model Fit Indices" on page 6903 for details. This output can be suppressed by the NOPRINT option.

- fit comparison for multiple-group analysis. See the section "Individual Fit Indices for Multiple Groups" on page 6910 for the fit indices for group comparison. This output can be suppressed by the NOPRINT option.

- the predicted covariance matrix and its determinant and mean vector, if you specify the output option PCORR or PALL.

- residual and normalized residual matrix if you specify the RESIDUAL option or at least the PRINT option. The variance standardized or asymptotically standardized residual matrix can be displayed also. The average residual and the average off-diagonal residual are also displayed. Note that normalized or asymptotically standardized residuals are not applicable for the ULS and DWLS estimation methods.

  See the section "Residuals" on page 6901 for more details.

- rank order of the largest normalized residuals if you specify the RESIDUAL option or at least the PRINT option.
- bar chart of the normalized residuals if you specify the RESIDUAL option or at least the PRINT option.
- plotting of smoothed density functions of residuals if you request the ODS Graphics by the PLOTS= option.
- equations of linear dependencies among the parameters used in the model specification if the information matrix is recognized as singular at the final solution.
- the estimation results and the standardized results. Except for ULS or DWLS estimates, the approximate standard errors and t values are also displayed. This output is displayed by default or if you specify the PESTIM option or at least the PSHORT option.

  Depending on the modeling language, the sets of output vary, as shown in the following:

  - FACTOR: factor loading matrix, rotation matrix, rotated factor loading matrix (if rotation requested), factor covariance matrix, intercepts, factor means, and error variances in the model. Factor rotation matrix is printed for the unstandardized solution.
  - LINEQS: linear equations, variance and covariance parameters, and mean parameters in the model.
  - LISMOD: all model matrices in the model.
  - MSTRUCT: covariance matrix and mean vectors.
  - PATH: the path list, variance and covariance parameters, intercept and mean parameters.
  - RAM: a list of parameters, their types, names, and initial values.
- squared multiple correlations table which displays the error variance, total variance, and the squared multiple correlation of each endogenous variable in the model. The total variances are the diagonal elements of the predicted covariance matrix. This output is displayed if you specify the PESTIM option or at least the PSHORT option.
- the total determination of all equations, the total determination of the latent equations, and the total determination of the manifest equations if you specify the PDETERM or the PALL option. See the section "Assessment of Fit" on page 6900 for more details. If you specify subsets of variables in the DETERM statements, the corresponding determination coefficients will also be shown. If one of the determinants in the formula for computing the determination coefficient is zero, the corresponding coefficient is displayed as the missing value '.'.
- the matrix of estimated covariances among the latent variables in the model if you specify the PLATCOV option or at least the PRINT option.
- the matrix of estimated covariances between latent and manifest variables in the model if you specify the PLATCOV option or at least the PRINT option.
- the vector of estimated means for the latent and manifest variables in the model if you specify the PLATCOV option or at least the PRINT option.

- the matrix **FSR** of latent variable scores regression coefficients if you specify the PLATCOV option or at least the PRINT option. The **FSR** matrix is a generalization of Lawley and Maxwell's (1971, p.109) factor scores regression matrix,

  $$\mathbf{FSR} = \hat{\mathbf{\Sigma}}_{yx}\hat{\mathbf{\Sigma}}_{xx}^{-1}$$

  where $\hat{\mathbf{\Sigma}}_{xx}$ is the $p \times p$ predicted covariance matrix among manifest variables and $\hat{\mathbf{\Sigma}}_{yx}$ is the $m \times p$ matrix of the predicted covariances between latent and manifest variables, with p being the number of manifest variables, and m being the number of latent variables. You can multiply the observed values by this matrix to estimate the scores of the latent variables used in your model.

- stability coefficient of reciprocal causation if you request the effect analysis by using the EFFPART or TOTEFF option, and you must not use the NOPRINT option.

- the matrices for the total, direct, and indirect effects if you specify the EFFPART or TOTEFF option or at least the PRINT option, and you must not use the NOPRINT option. Unstandardized and standardized effects are printed in separate tables. Standard errors for the all estimated effects are also included in the output. Additional tables for effects are available if you request specialized effect analysis in the EFFPART statements.

- the matrix of rotated factor loadings and the orthogonal transformation matrix if you specify the ROTATE= and PESTIM options or at least the PSHORT options. This output is available for the FACTOR models.

- factor scores regression matrix, if you specify the PESTIM option or at least the PSHORT option. The determination of manifest variables is displayed only if you specify the PDETERM option.

- univariate Lagrange multiplier indices if you specify the MODIFICATION (or MOD) or the PALL option. The value of a Lagrange multiplier (LM) index indicates the approximate drop in χ^2 when the corresponding fixed parameter in the original model is freely estimated. The corresponding probability (with $df = 1$) and the estimated change of the parameter value are printed. Ranking of the LM indices is automatically done for prescribed parameter subsets of the original model. The LM indices with greatest improvement of χ^2 model fit appear in the beginning of the ranking list. Note that LM indices are not applicable to the ULS and the DWLS estimation methods. See the section "Modification Indices" on page 6917 for more detail.

- matrices of univariate Lagrange multiplier (LM) indices if you specify the MODIFICATION (or MOD) or the PALL option, and the LMMAT option in the LMTESTS statement. These matrices are predefined in PROC TCALIS, or you can specify them in the LMTESTS statements. If releasing a fixed parameter in the matrix would result in a singular information matrix, the string 'Singular' is displayed instead of the Lagrange multiplier index. If a fixed entry in the matrix is restricted by the model (for example, fixed ones for coefficients associated with error terms) or being excluded in the specified subsets in the LMTESTS statement, the string 'Excluded' is displayed. Note that matrices for LM indices are not printed for the ULS and the DWLS estimation methods. See the section "Modification Indices" on page 6917 for more detail.

- univariate Lagrange multiplier test indices for releasing equality constraints if you specify the MODIFICATION (or MOD) or the PALL option. Note that this output is not applicable to the ULS and the DWLS estimation methods. See the section "Modification Indices" on page 6917 for more detail.
- univariate Lagrange multiplier test indices for releasing active boundary constraints specified by the BOUNDS statement if you specify the MODIFICATION (or MOD) or the PALL option. Note that this output is not applicable to the ULS and the DWLS estimation methods. See the section "Modification Indices" on page 6917 for more detail.
- the stepwise multivariate Wald test for constraining estimated parameters to zero constants if the MODIFICATION (or MOD) or the PALL option is specified and the univariate probability is larger than the value specified in the PMW= option (default PMW=0.05). Note that this output is not applicable to the ULS and the DWLS estimation methods. See the section "Modification Indices" on page 6917 for more details.

ODS Table Names

PROC TCALIS assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

There are numerous ODS tables in the TCALIS procedure. The conditions for these ODS tables to display vary a lot. For convenience in presentation, the ODS tables for the PROC TCALIS procedure are organized in the following categories:

- ODS tables for descriptive statistics and residual analysis
- ODS tables for model specification and results
- ODS tables for supplementary model analysis
- ODS tables for modification indices
- ODS tables for optimization control and results

Many ODS tables are displayed when you set either a specialized option in a certain statement or a global display option in the PROC TCALIS statement. Rather than requesting displays by setting specialized options separately, you can request a group of displays by using a global display option.

There are five global display levels, represented by five options: PALL (highest), PRINT, *default*, PSHORT, and PSUMMARY. The higher the level, the more output requested. The *default* printing level is in effect when you do not specify any other global printing options in the PROC TCALIS statement. See the section "Global Display Options" on page 6724 for details.

In the following description of ODS tables whenever applicable, the lowest level of global printing options for an ODS table to print is listed. It is understood that global printing options at higher levels can also be used. For example, if PSHORT is the global display option to print an ODS table, you can also use PALL, PRINT, or *default*.

ODS Tables for Descriptive Statistics and Residual Analysis

These ODS tables are group-oriented, meaning that each group has its own set of tables in the output. To display these tables in your output, you can set a specialized option in either the PROC TCALIS or GROUP statement. If the specialized option is set in the PROC TCALIS statement, it will apply to all groups. If the option is set in the GROUP statement, it will apply to the associated group only. Alternatively, you can set a global printing option in the PROC TCALIS statement to print these tables. Either a specialized or a global printing option is sufficient to print the tables.

Table Names for Descriptive Statistics

ODS Table Name	Description	Specialized Option	Global Display Option
ContKurtosis	Contributions to kurtosis from observations	KURTOSIS	PRINT
InCorr	Input correlation matrix	PCORR	PALL
InCorrDet	Determinant of the input correlation matrix	PCORR	PALL
InCov	Input covariance matrix	PCORR	PALL
InCovDet	Determinant of the input covariance matrix	PCORR	PALL
InMean	Input mean vector	PCORR	PALL
Kurtosis	Kurtosis, with raw data input	KURTOSIS	PRINT
PredCorr	Predicted correlation matrix	PCORR	PALL
PredCorrDet	Determinant of the predicted correlation matrix	PCORR	PALL
PredCov	Predicted covariance matrix	PCORR	PALL
PredCovDet	Determinant of the predicted covariance matrix	PCORR	PALL
PredMean	Predicted mean vector	PCORR	PALL
SimpleStatistics	Simple statistics, with raw data input	SIMPLE	default
Weights	Weight matrix	PWEIGHT	PALL
WeightsDet	Determinant of the weight matrix	PWEIGHT	PALL

Table Names for Residual Displays

ODS Table Name	Description	Specialized Option	Global Display Option
AsymStdRes	Asymptotically standardized residual matrix	RESIDUAL=ASYSTAND[1]	PALL
AveAsymStdRes	Average of absolute asymptotically standardized residual values	RESIDUAL=ASYSTAND[1]	PALL
AveNormRes	Average of absolute normalized residual values	RESIDUAL=NORM[1]	PALL
AveRawRes	Average of absolute raw residual values	RESIDUAL	PALL
AveVarStdRes	Average of absolute variance standardized residual values	RESIDUAL=VARSTAND[1]	PALL
DistAsymStdRes	Distribution of asymptotically standardized residuals	RESIDUAL=ASYSTAND[1]	PALL
DistNormRes	Distribution of normalized residuals	RESIDUAL=NORM[1]	PALL
DistRawRes	Distribution of raw residuals	RESIDUAL	PALL
DistVarStdRes	Distribution of variance standardized residuals	RESIDUAL=VARSTAND[1]	PALL
NormRes	Normalized residual matrix	RESIDUAL=NORM[1]	PALL
RawRes	Raw residual matrix	RESIDUAL[2]	PALL
RankAsymStdRes	Rank order of asymptotically standardized residuals	RESIDUAL=ASYSTAND[1]	PALL
RankNormRes	Rank order of normalized residuals	RESIDUAL=NORM[1]	PALL
RankRawRes	Rank order of raw residuals	RESIDUAL	PALL
RankVarStdRes	Rank order of variance standardized residuals	RESIDUAL=VARSTAND[1]	PALL
VarStdRes	Variance standardized residual matrix	RESIDUAL=VARSTAND[1]	PALL

1. In effect, the RESIDUAL= option specifies the RESIDUAL option and the type of residuals requested after the equal sign. For example, if you set RESIDUAL=ASYSTAND, asymptotically standardized residuals are requested, in addition to the printing of the tables enabled by the RESIDUAL option. In some cases, the RESIDUAL= option cannot be honored due to the specific estimation method or data type used. When this occurs, PROC TCALIS will determine the appropriate sets of normalized or standardized residuals to display. A warning message with an explanation will be printed.
2. Raw residuals are also printed for correlation analysis even if RESIDUAL or PALL is not specified.

ODS Tables for Model Specification and Results

Some ODS tables of this group are model-oriented. Others are not. Model-oriented ODS tables are printed for each model, while others are printed no more than once no matter how many models you have.

Non-Model-Oriented ODS Tables

The ODS tables that are not model-oriented are listed in the following table:

ODS Table Name	Description	Global Display Option	Additional Specification Required
AddParms	Estimates for additional parameters	PSHORT	PARAMETERS statement
AddParmsInit	Initial values for additional parameters	PSHORT	PARAMETERS statement
Fit	Fit summary	PSUMMARY	
GroupFit	Fit comparison among groups	PSUMMARY	Multiple groups
ModelingInfo	General modeling information	PSHORT	
ModelSummary	Summary of models and their labels and types	PSHORT	Multiple models[1]
ParmFunc	Parametric function testing	PSHORT	TESTFUNC statement
SimTest	Simultaneous tests of parametric functions	PSHORT	SIMTEST statement

1. This table is displayed when you have multiple models that have labels specified by the LABEL= option, or when you define a model with more than a single level of reference by using the REFMODEL option. Otherwise, the ModelingInfo table contains all pertinent information regarding the models in the analysis.

Model-Oriented ODS Tables

These ODS tables are model-oriented, meaning that each model has its own set of ODS tables in the output. There are three types of model specification and results printing in PROC TCALIS: initial specification, (unstandardized) estimated model results, and standardized model results. To distinguish these three types of ODS tables, different suffixes for the ODS table names are used. An "Init" suffix indicates initial specification, while a "Std" suffix indicates standardized solutions. All other tables are for unstandardized solutions.

These ODS tables require some specialized options to print. If you set the specialized option in the PROC TCALIS statement, it applies to all models. If you set the specialized option in the MODEL statement, it applies to the associated model only. Alternatively, to print **all** these ODS tables, you can use the PSHORT or any higher level global printing option in the PROC TCALIS statement. Either a specialized or a global printing option is sufficient to print these ODS tables. The following is a summary of the specialized and global printing options for these three types of ODS tables:

Type of ODS Tables	Table Name Suffix	Specialized Option	Global Display Option
Initial specification	Init	PINITIAL	PSHORT
Unstandardized solutions	(none)	PESTIM	PSHORT
Standardized solutions	Std	PESTIM, and NOSTAND not used	PSHORT

In the following list of ODS tables, the prefixes of the ODS table names indicate the modeling language required for the ODS tables to print. The last column of the list indicates whether the PRIMAT option is needed to print the corresponding ODS tables in matrix formats. You can use the PRIMAT option either in the PROC TCALIS or MODEL statement. If you want matrix output for all models, set this option in the PROC TCALIS statement. If you want matrix output for a specific model, set this option in the associated MODEL statement only.

ODS Table Name	Description	Additional Option
FACTORCov	Estimated factor covariances	
FACTORCovInit	Initial factor covariances	
FACTORCovStd	Factor correlations	
FACTORErrVar	Estimated error variances	
FACTORErrVarInit	Initial error variances	
FACTORErrVarStd	Standardized results for error variances	
FACTORIntercepts	Estimated intercepts	
FACTORInterceptsInit	Initial intercepts	
FACTORLoadings	Estimated factor loadings	
FACTORLoadingsInit	Initial factor loadings	
FACTORLoadingsStd	Standardized factor loadings	
FACTORMeans	Estimated factor means	
FACTORMeansInit	Initial factor means	
FACTORRotCov	Estimated rotated factor covariances	
FACTORRotCovStd	Rotated factor correlations	
FACTORRotErrVar	Error variances in rotated solution	
FACTORRotErrVarStd	Standardized results for error variances in rotated solution	
FACTORRotLoadings	Rotated factor loadings	
FACTORRotLoadingsStd	Standardized rotated factor loadings	
FACTORRotMat	Rotation matrix	
FACTORScoresRegCoef	Factor scores regression coefficients	
FACTORVariables	Variables in the analysis	
LINEQSAlpha	Estimated intercept vector	PRIMAT
LINEQSAlphaInit	Initial intercept vector	PRIMAT
LINEQSBeta	Estimated _EQSBETA_ matrix	PRIMAT
LINEQSBetaInit	Initial _EQSBETA_ matrix	PRIMAT
LINEQSBetaStd	Standardized results for _EQSBETA_ matrix	PRIMAT
LINEQSCovExog	Estimated covariances among exogenous variables	
LINEQSCovExogInit	Initial covariances among exogenous variables	
LINEQSCovExogStd	Standardized results for covariances among exogenous variables	
LINEQSEq	Estimated equations	
LINEQSEqInit	Initial equations	
LINEQSEqStd	Standardized equations	
LINEQSGamma	Estimated _EQSGAMMA_ matrix	PRIMAT
LINEQSGammaInit	Initial _EQSGAMMA_ matrix	PRIMAT

ODS Table Name	Description	Additional Option
LINEQSGammaStd	Standardized results for _EQSGAMMA_ matrix	PRIMAT
LINEQSMeans	Estimated means for exogenous variables	
LINEQSMeansInit	Initial means for exogenous variables	
LINEQSNu	Estimated mean vector	PRIMAT
LINEQSNuInit	Initial mean vector	PRIMAT
LINEQSPhi	Estimated _EQSPHI_ matrix	PRIMAT
LINEQSPhiInit	Initial _EQSPHI_ matrix	PRIMAT
LINEQSPhiStd	Standardized results for _EQSPHI_ matrix	PRIMAT
LINEQSVarExog	Estimated variances of exogenous variables	
LINEQSVarExogInit	Initial variances of exogenous variables	
LINEQSVarExogStd	Standardized results for variances of exogenous variables	
LINEQSVariables	Exogenous and endogenous variables	
LISMODAlpha	Estimated _ALPHA_ vector	
LISMODAlphaInit	Initial _ALPHA_ vector	
LISMODBeta	Estimated _BETA_ matrix	
LISMODBetaInit	Initial _BETA_ matrix	
LISMODBetaStd	Standardized _BETA_ matrix	
LISMODGamma	Estimated _GAMMA_ matrix	
LISMODGammaInit	Initial _GAMMA_ matrix	
LISMODGammaStd	Standardized _GAMMA_ matrix	
LISMODKappa	Estimated _KAPPA_ vector	
LISMODKappaInit	Initial _KAPPA_ vector	
LISMODLambdaX	Estimated _LAMBDAX_ matrix	
LISMODLambdaXInit	Initial _LAMBDAX_ matrix	
LISMODLambdaXStd	Standardized _LAMBDAX_ matrix	
LISMODLambdaY	Estimated _LAMBDAY_ matrix	
LISMODLambdaYInit	Initial _LAMBDAY_ matrix	
LISMODLambdaYStd	Standardized _LAMBDAY_ matrix	
LISMODNuX	Estimated _NUX_ vector	
LISMODNuXInit	Initial _NUX_ vector	
LISMODNuY	Estimated _NUY_ vector	
LISMODNuYInit	Initial _NUY_ vector	
LISMODPhi	Estimated _PHI_ matrix	
LISMODPhiInit	Initial _PHI_ matrix	
LISMODPhiStd	Standardized _PHI_ matrix	
LISMODPsi	Estimated _PSI_ matrix	
LISMODPsiInit	Initial _PSI_ matrix	
LISMODPsiStd	Standardized _PSI_ matrix	
LISMODThetaX	Estimated _THETAX_ matrix	
LISMODThetaXInit	Initial _THETAX_ matrix	
LISMODThetaXStd	Standardized _THETAX_ matrix	
LISMODThetaY	Estimated _THETAY_ matrix	
LISMODThetaYInit	Initial _THETAY_ matrix	
LISMODThetaYStd	Standardized _THETAY_ matrix	

ODS Table Name	Description	Additional Option
LISMODVariables	Variables in the model	
MSTRUCTCov	Estimated _COV_ matrix	
MSTRUCTCovInit	Initial _COV_ matrix	
MSTRUCTCovStd	Standardized _COV_ matrix	
MSTRUCTMean	Estimated _MEAN_ vector	
MSTRUCTMeanInit	Initial _MEAN_ vector	
MSTRUCTVariables	Variables in the model	
PATHCovErrors	Estimated error covariances	
PATHCovErrorsInit	Initial error covariances	
PATHCovErrorsStd	Standardized error covariances	
PATHCovVarErr	Estimated covariances bewteen exogenous variables and errors	
PATHCovVarErrInit	Initial covariances bewteen exogenous variables and errors	
PATHCovVarErrStd	Standardized results for covariances bewteen exogenous variables and errors	
PATHCovVars	Estimated covariances among exogenous variables	
PATHCovVarsInit	Initial covariances among exogenous variables	
PATHCovVarsStd	Standardized results for covariances among exogenous variables	
PATHList	Estimated path list	
PATHListInit	Initial path list	
PATHListStd	Standardized path list	
PATHMeansIntercepts	Estimated intercepts	
PATHMeansInterceptsInit	Initial intercepts	
PATHVariables	Exogenous and endogenous variables	
PATHVarParms	Estimated variances or error variances	
PATHVarParmsInit	Initial variances or error variances	
PATHVarParmsStd	Standardized results for variances or error variances	
RAMAMat	Estimated _A_ matrix	PRIMAT
RAMAMatInit	Initial _A_ matrix	PRIMAT
RAMAMatStd	Standardized results of _A_ matrix	PRIMAT
RAMList	List of RAM estimates	
RAMListInit	List of initial RAM estimates	
RAMListStd	Standardized results for RAM estimates	
RAMPMat	Estimated _P_ matrix	PRIMAT
RAMPMatInit	Initial _P_ matrix	PRIMAT
RAMPMatStd	Standardized results of _P_ matrix	PRIMAT
RAMVariables	Exogenous and endogenous variables	
RAMWVec	Estimated mean and intercept vector	PRIMAT
RAMWVecInit	Initial mean and intercept vector	PRIMAT

ODS Tables for Supplementary Model Analysis

These ODS tables are model-oriented. They are printed for each model in your analysis. To display these ODS tables, you can set some specialized **options** in either the PROC TCALIS or MODEL statement. If the specialized options are used in the PROC TCALIS statement, they apply to all models. If the specialized options are used in the MODEL statement, they apply to the associated model only. For some of these ODS tables, certain specialized **statements** for the model might also enable the printing. Alternatively, you can use the global printing options in the PROC TCALIS statement to print these ODS tables. Either a specialized option (or statement) or a global printing option is sufficient to print a particular ODS table.

ODS Table Name	Description	Specialized Option or Statement	Global Display Option
Determination	Coefficients of determination	PDETERM DETERM[4]	default
DirectEffects	Direct effects	TOTEFF[1]	PRINT
DirectEffectsStd	Standardized direct effects	TOTEFF[1,3]	PRINT
EffectsOf	Effects of the listed variables	EFFPART[2]	PRINT
EffectsOn	Effects on the listed variables	EFFPART[2]	PRINT
IndirectEffects	Indirect effects	TOTEFF[1]	PRINT
IndirectEffectsStd	Standardized indirect effects	TOTEFF[1,3]	PRINT
LatentScoresRegCoef	Latent variable scores regression coefficients	PLATCOV	PRINT
PredCovLatent	Predicted covariances among latent variables	PLATCOV	PRINT
PredCovLatMan	Predicted covariances between latent and manifest variables	PLATCOV	PRINT
PredMeanLatent	Predicted means of latent variables	PLATCOV	PRINT
SqMultCorr	Squared multiple correlations	PESTIM	PSHORT
Stability	Stability coefficient of reciprocal causation	PDETERM, DETERM[4]	default
StdEffectsOf	Standardized effects of the listed variables	EFFPART[2,3]	PSHORT
StdEffectsOn	Standardized effects on the listed variables	EFFPART[2,3]	PSHORT
TotalEffects	Total effects	TOTEFF[1]	PRINT
TotalEffectsStd	Standardized total effects	TOTEFF[1,3]	PRINT

1. This refers to the TOTEFF or EFFPART option in the PROC TCALIS or MODEL statement.
2. This refers to the EFFPART statement specifications.
3. NOSTAND option must not be specified in the MODEL or PROC TCALIS statement.
4. PDETERM is an option specified in the PROC TCALIS or MODEL statement, while DETERM is a statement name.

ODS Tables for Model Modification Indices

To print the ODS tables for model modification indices, you can use the MODIFICATION option in either the PROC TCALIS or MODEL statement. When this option is set in the PROC TCALIS statement, it applies to all models. When this option is set in the MODEL statement, it applies to the associated model only. Alternatively, you can also use the PALL option in the PROC TCALIS statement to print these ODS tables.

If the NOMOD option is set in the PROC TCALIS statement, these ODS tables are not printed for all models, unless the MODIFICATION is respecified in the individual MODEL statements. If the NOMOD option is set in the MODEL statement, then the ODS tables for modification do not print for the associated model.

For convenience in presentation, three different classes of ODS tables for model modifications are described in the following. First, ODS tables for ranking of LM indices are the default printing when the MODIFICATION option is specified. Second, ODS tables for LM indices in matrix forms require an additional option to print. Last, ODS tables for other modification indices, including the Wald test indices, require specific data-analytic conditions to print. While the first two classes of ODS tables are model-oriented (that is, each model has its own sets of output), the third one is not.

ODS Table Names for Ranking of LM indices

Rankings of LM indices in separate regions of parameter space are the default printing format. You can turn off this default printing by the NODEFAULT option in the LMTESTS statement for models. If you want to print matrices of LM test indices rather than the rankings of LM test indices, you can use the NORANK or MAXRANK=0 option in the LMTESTS statement.

These ODS tables are specific to the types of modeling languages used. This is noted in the last column of the following table.

ODS Table Name	Description	Model
LMRankCov	Covariances among variables	MSTRUCT
LMRankCovErr	Covariances among errors	LINEQS
LMRankCovErrofVar	Covariances among errors of variables	RAM or PATH
LMRankCovExog	Covariances among existing exogenous variables	LINEQS, PATH, or RAM
LMRankCovFactors	Covariance among factors	FACTOR
LMRankErrorVar	Error variances	FACTOR
LMRankFactMeans	Factor means	FACTOR
LMRankIntercepts	Intercepts	FACTOR, LINEQS, PATH, or RAM
LMRankLoadings	Factor loadings	FACTOR
LMRankMeans	Means of existing variables	LINEQS, MSTRUCT, PATH, or RAM
LMRankPaths	All possible paths in the model	PATH or RAM

ODS Table Name	Description	Model
LMRankPathsFromEndo	Paths from existing endogenous variables	LINEQS
LMRankPathsFromExog	Paths from existing exogenous variables	LINEQS
LMRankPathsNewEndo	Paths to existing exogenous variables	LINEQS
LMRankLisAlpha	LISMOD _ALPHA_	LISMOD
LMRankLisBeta	LISMOD _BETA_	LISMOD
LMRankLisGamma	LISMOD _GAMMA_	LISMOD
LMRankLisKappa	LISMOD _KAPPA_	LISMOD
LMRankLisLambdaX	LISMOD _LAMBDAX_	LISMOD
LMRankLisLambdaY	LISMOD _LAMBDAY_	LISMOD
LMRankLisNuX	LISMOD _NUX_	LISMOD
LMRankLisNuY	LISMOD _NUY_	LISMOD
LMRankLisPhi	LISMOD _PHI_	LISMOD
LMRankLisPsi	LISMOD _PSI_	LISMOD
LMRankLisThetaX	LISMOD _THETAX_	LISMOD
LMRankLisThetaY	LISMOD _THETAY_	LISMOD
LMRankMySet	Customized sets of new parameters defined in LMTESTS statements	any model

ODS Table Names for Lagrange Multiplier Tests in Matrix Form

To print matrices of LM test indices for a model, you must also use the LMMAT option in the LMTESTS statement for the model. This option itself is not sufficient for **all** the matrices of LM tests to print. Only those matrices associated with the LM test indices that are selected (by default or by customized sets) in the LMTESTS statement will be printed.

These ODS tables are specific to the types of modeling languages used in the model. This is noted in the last column of the following table.

ODS Table Name	Description	Model
LMEqsAlpha	_EQSALPHA_ vector	LINEQS
LMEqsBeta	_EQSBETA_ matrix	LINEQS
LMEqsGammaSub	_EQSGAMMA_ matrix, excluding entries with error variables in columns	LINEQS
LMEqsNewDep	New rows for expanding _EQSBETA_ and _EQSGAMMA_ matrices	LINEQS
LMEqsNuSub	_EQSNU_ vector, excluding fixed zero means for error variables	LINEQS
LMEqsPhi	_EQSPHI_ matrix	LINEQS
LMEqsPhi11	Upper left partition (exogenous variances and covariances) of the _EQSPHI_ matrix	LINEQS
LMEqsPhi21	Lower left partition (error variances and covariances) of the _EQSPHI_ matrix	LINEQS
LMEqsPhi22	Lower right partition (error variances and covariances) of the _EQSPHI_ matrix	LINEQS

ODS Table Name	Description	Model
LMFactErrv	Vector of error variances	FACTOR
LMFactFcov	Factor covariance matrix	FACTOR
LMFactInte	Intercept vector	FACTOR
LMFactLoad	Factor loading matrix	FACTOR
LMFactMean	Factor mean vector	FACTOR
LMLisAlpha	LISMOD _ALPHA_ vector	LISMOD
LMLisBeta	LISMOD _BETA_ matrix	LISMOD
LMLisGamma	LISMOD _GAMMA_ matrix	LISMOD
LMLisKappa	LISMOD _KAPPA_ vector	LISMOD
LMLisLambdaX	LISMOD _LAMBDAX_ matrix	LISMOD
LMLisLambdaY	LISMOD _LAMBDAY_ matrix	LISMOD
LMLisNuX	LISMOD _NUX_ vector	LISMOD
LMLisNuY	LISMOD _NUY_ vector	LISMOD
LMLisPhi	LISMOD _PHI_ matrix	LISMOD
LMLisPsi	LISMOD _PSI_ matrix	LISMOD
LMLisThetaX	LISMOD _THETAX_ matrix	LISMOD
LMLisThetaY	LISMOD _THETAY_ matrix	LISMOD
LMMstructCov	Covariance matrix	MSTRUCT
LMMstructMean	Mean vector	MSTRUCT
LMRamA	_RAMA_ matrix	RAM
LMRamALower	Lower partitions of the _RAMA_ matrix	RAM
LMRamAUpper	Upper partitions of the _RAMA_ matrix	RAM
LMRamAlpha	_RAMALPHA_ matrix	RAM
LMRamBeta	Upper left partition of the _RAMA_ matrix	RAM
LMRamGamma	Upper right partition of the _RAMA_ matrix	RAM
LMRamNu	_RAMNU_ matrix	RAM
LMRamP	_RAMP_ matrix	RAM
LMRamP11	Upper left partition of the _RAMP_ matrix	RAM
LMRamP21	Lower left partition of the _RAMP_ matrix	RAM
LMRamP22	Lower right partition of the _RAMP_ matrix	RAM
LMRamW	_RAMW_ matrix	RAM

ODS Table Names for Other Modification Indices

The following table shows the ODS tables for the remaining modification indices.

ODS Table Name	Description	Additional Requirement
LagrangeBoundary	LM tests for active boundary constraints	Presence of active boundary constraints
LagrangeDepParmEquality	LM tests for equality constraints in dependent parameters	Presence of equality constraints in dependent parameters
LagrangeEquality	LM tests for equality constraints	Presence of equality constraints in independent parameters
WaldTest	Wald tests for testing existing parameters equaling zeros	At least one insignificant parameter value

ODS Table for Optimization Control and Results

To display the ODS tables for optimization control and results, you must specify any of the following global display options in the PROC TCALIS statement: PSHORT, PRINT, PALL, or default (that is, NOPRINT is not specified). Also, you must not use the NOPRINT option in the NLOPTIONS statement. For some of these tables, you must also specify additional options, either in the PROC TCALIS or the NLOPTIONS statement. Some restrictions might apply. Additional options and restrictions are noted in the last column.

ODS Table Name	Description	Additional Option Required or Restriction
CovParm	Covariances of parameters	PCOVES[1] or PALL[2], restriction[3]
ConvergenceStatus	Convergence status	
DependParmsResults	Final dependent parameter estimates	Restriction[4]
DependParmsStart	Initial dependent parameter estimates	Restriction[4]
Information	Information matrix	PCOVES[1] or PALL[2], restriction[3]
InitEstMethod	Initial estimation methods	
InputOptions	Optimization options	PALL[2]
IterHist	Iteration history	
IterStart	Iteration start	
IterStop	Iteration stop	
Lagrange	First and second order Lagrange multipliers	PALL[2]
LinCon	Linear constraints	PALL[2], restriction[5]
LinConDel	Deleted constraints	PALL[2], restriction[5]
LinConSol	Linear constraints evaluated at solution	PALL[2], restriction[5]
LinDep	Linear dependencies of parameter estimates	Restriction[6]
ParameterEstimatesResults	Final estimates	
ParameterEstimatesStart	Initial estimates	
ProblemDescription	Problem description	
ProjGrad	Projected gradient	PALL[2]

1. PCOVES option is specified in the PROC TCALIS statement.
2. PALL option is specified in the NLOPTIONS statement.
3. Estimation method must not be ULS or DWLS.
4. Existence of dependent parameters.
5. Linear equality or boundary constraints are imposed.
6. Existence of parameter dependencies during optimization, but not due to model specification.

ODS Graphics

To request graphics with PROC TCALIS, you must first enable ODS Graphics by specifying the **ods graphics on** statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. In the following table, ODS graph names and the options to display the graphs are listed.

ODS Graph Name	Plot Description	PLOTS= Option
AsymStdResidualHistogram	Asymptotically standardized residuals	PLOTS=RESIDUALS and RESIDUAL=ASYMSTD, METHOD= is not ULS or DWLS
NormResidualHistogram	Normalized residuals	PLOTS=RESIDUALS and RESIDUAL=NORM
RawResidualHistogram	Raw residuals	PLOTS=RESIDUALS
VarStdResidualHistogram	Variance standardized residuals	PLOTS=RESIDUALS and RESIDUAL=VARSTD

Examples: TCALIS Procedure

Example 88.1: Path Analysis: Stability of Alienation

The following covariance matrix from Wheaton et al. (1977) has served to illustrate the performance of several implementations for the analysis of structural equation models. Two different models have been analyzed by an early implementation of LISREL and are mentioned in Jöreskog (1978). You can also find a more detailed discussion of these models in the LISREL VI manual (Jöreskog and Sörbom 1985). A slightly modified model for this covariance matrix is included in the EQS 2.0 manual (Bentler 1985, p. 28). The path diagram of this model is displayed in Figure 88.2. The same model is reanalyzed here by PROC TCALIS. However, for the analysis with the EQS implementation, the SEI variable is rescaled by a factor of 0.1 to make the matrix less ill-conditioned. Since the Levenberg-Marquardt or Newton-Raphson optimization techniques are used with PROC CALIS, rescaling the data matrix is not necessary and, therefore, is not done here. The results reported here reflect the estimates based on the original covariance matrix.

```
title "Stability of Alienation";
title2 "Data Matrix of WHEATON, MUTHEN, ALWIN & SUMMERS (1977)";
data Wheaton(TYPE=COV);
   _type_ = 'cov';
   input _name_ $ 1-11 Anomie67 Powerless67 Anomie71 Powerless71
                       Education SEI;
   label Anomie67='Anomie (1967)' Powerless67='Powerlessness (1967)'
         Anomie71='Anomie (1971)' Powerless71='Powerlessness (1971)'
         Education='Education'    SEI='Occupational Status Index';
   datalines;
Anomie67       11.834     .        .        .       .        .
Powerless67     6.947    9.364     .        .       .        .
Anomie71        6.819    5.091   12.532     .       .        .
Powerless71     4.783    5.028    7.495    9.986    .        .
Education      -3.839   -3.889   -3.841   -3.625   9.610     .
SEI           -21.899  -18.831  -21.748  -18.775  35.522  450.288
;

ods graphics on;

proc tcalis nobs=932 data=Wheaton plots=residuals;
   path
      Anomie67     <-  Alien67   1.0,
      Powerless67  <-  Alien67   0.833,
      Anomie71     <-  Alien71   1.0,
      Powerless71  <-  Alien71   0.833,
      Education    <-  SES       1.0,
      SEI          <-  SES       lambda,
      Alien67      <-  SES       gamma1,
      Alien71      <-  SES       gamma2,
      Alien71      <-  Alien67   beta;
   pvar
      Anomie67     = theta1,
      Powerless67  = theta2,
      Anomie71     = theta1,
      Powerless71  = theta2,
      Education    = theta3,
      SEI          = theta4,
      Alien67      = psi1,
      Alien71      = psi2,
      SES          = phi;
   pcov
      Anomie67    Anomie71    = theta5,
      Powerless67 Powerless71 = theta5;
run;

ods graphics off;
```

Since no METHOD= option has been used, maximum likelihood estimates are computed by default. In this example, no global display options are specified. This means that PROC TCALIS prints the default set of results.

PROC TCALIS can produce a high-quality residual histogram that is useful for showing the distribution of residuals. To request the residual histogram, you must first enable ODS Graphics by specifying the **ods graphics on** statement, as shown in the preceding statements before the PROC TCALIS statement. Then, the residual histogram is requested by the **plots=residuals** option in the PROC TCALIS statement.

Output 88.1.1 displays the modeling information and variables in the analysis.

Output 88.1.1 Model Specification and Variables

```
                        Stability of Alienation
                        PATH Model Specification

                          The TCALIS Procedure
          Covariance Structure Analysis: Model and Initial Values

                          Modeling Information

                     Data Set             WORK.WHEATON
                     N Obs                932
                     Model Type           PATH

                          Variables in the Model

Endogenous     Manifest    Anomie67  Anomie71  Education  Powerless67
                           Powerless71  SEI
               Latent      Alien67  Alien71
Exogenous      Manifest
               Latent      SES

                  Number of Endogenous Variables = 8
                  Number of Exogenous Variables  = 1
```

Output 88.1.1 shows that the data set Wheaton was used with 932 observations. The model is specified with the PATH modeling language. Variables in the model are classified into different categories according to their roles. All manifest variables are endogenous in the model. Also, three latent variables are hypothesized in the model: Alien67, Alien71, and SES. While Alien67 and Alien71 are endogenous, SES is exogenous in the model.

Output 88.1.2 echoes the initial specification of the PATH model.

Output 88.1.2 Initial Estimates

Initial Estimates for PATH List

Path			Parameter	Estimate
Anomie67	<-	Alien67		1.00000
Powerless67	<-	Alien67		0.83300
Anomie71	<-	Alien71		1.00000
Powerless71	<-	Alien71		0.83300
Education	<-	SES		1.00000
SEI	<-	SES	lambda	.
Alien67	<-	SES	gamma1	.
Alien71	<-	SES	gamma2	.
Alien71	<-	Alien67	beta	.

Initial Estimates for Variance Parameters

Variance Type	Variable	Parameter	Estimate
Error	Anomie67	theta1	.
	Powerless67	theta2	.
	Anomie71	theta1	.
	Powerless71	theta2	.
	Education	theta3	.
	SEI	theta4	.
	Alien67	psi1	.
	Alien71	psi2	.
Exogenous	SES	phi	.

Initial Estimates for Covariances Among Errors

Error of	Error of	Parameter	Estimate
Anomie67	Anomie71	theta5	.
Powerless67	Powerless71	theta5	.

In Output 88.1.2, numerical values for estimates are the initial values you input in the model specification. If the associated parameter name for a numerical estimate is blank, it means that the estimate is a fixed value, which would not be changed in the estimation. For example, the first five paths have fixed path coefficients with the fixed values given. For numerical estimates with parameter names given, the numerical values serve as initial values, which would be changed during the estimation. In Output 88.1.2, you actually do not have this kind of specification. All free parameters specified in the model are with missing initial values, denoted by '.'. For example, lambda, gamma1, theta1, and psi1, among others, are free parameters without initial values given. The initial values of these parameters will be generated by PROC TCALIS automatically.

You can examine this output to ensure that the desired model is being analyzed. PROC TCALIS outputs the initial specifications or the estimation results in the order you specify in the model, unless you use reordering options such as ORDERSPEC and ORDERALL. Therefore, the input order of specifications is important—it determines how your output would look.

Simple descriptive statistics are displayed in Output 88.1.3.

Output 88.1.3 Descriptive Statistics

Simple Statistics

	Variable	Mean	Std Dev
Anomie67	Anomie (1967)	0	3.44006
Powerless67	Powerlessness (1967)	0	3.06007
Anomie71	Anomie (1971)	0	3.54006
Powerless71	Powerlessness (1971)	0	3.16006
Education	Education	0	3.10000
SEI	Occupational Status Index	0	21.21999

Because the input data set contains only the covariance matrix, the means of the manifest variables are assumed to be zero. Note that this has no impact on the estimation, unless a mean structure model is being analyzed.

Initial estimates are necessary in all kinds of optimization problems. They could be provided by users or automatically generated by PROC TCALIS. As shown in Output 88.1.2, you did not provide any initial estimates for all free parameters. PROC TCALIS uses a combination of well-behaved mathematical methods to complete the initial estimation. The initial estimation methods for the current analysis are shown in Output 88.1.4.

Output 88.1.4 Optimization Starting Point

Initial Estimation Methods

1 Instrumental Variables Method
2 McDonald Method
3 Two-Stage Least Squares

Optimization Start
Parameter Estimates

N	Parameter	Estimate	Gradient
1	lambda	4.99508	-0.00206
2	gamma1	-0.62322	-0.04069
3	gamma2	-0.20437	-0.03816
4	beta	0.66589	0.03789
5	theta1	3.51433	-0.00409
6	theta2	3.65991	0.01182
7	theta3	2.49860	-0.00578
8	theta4	272.85274	0.0000194
9	psi1	5.57764	-0.00217
10	psi2	3.79636	-0.00935
11	phi	7.11140	0.00108
12	theta5	0.45298	-0.06463

Value of Objective Function = 0.0365979443

In this example, instrumental Variable Method, method by McDonald and Hartmann, and two-stage least squares method have been used for initial estimation. In the same output, the vector of initial parameter estimates and their gradients are also shown. The initial objective function value is 0.0366.

Output 88.1.5 displays the optimization information, including technical details, iteration history and convergence status.

Output 88.1.5 Optimization

```
            Parameter Estimates                                12
            Functions (Observations)                           21

                              Optimization Start

 Active Constraints                      0    Objective Function    0.0365979443
 Max Abs Gradient Element    0.0646338767    Radius                           1

                                                                       Actual
                                                       Max Abs           Over
         Rest    Func      Act    Objective  Obj Fun Gradient            Pred
 Iter    arts   Calls      Con     Function   Change  Element  Lambda   Change

    1       0       4        0      0.01453   0.0221  0.00142       0    1.013
    2       0       6        0      0.01448 0.000046 0.000249       0    1.001
    3       0       8        0      0.01448 1.007E-7 4.717E-6       0    1.006

                             Optimization Results

Iterations                            3  Function Calls                      11
Jacobian Calls                        5  Active Constraints                   0
Objective Function         0.0144844814  Max Abs Gradient Element   4.7172823E-6
Lambda                                0  Actual Over Pred Change    1.0060912391
Radius                      0.001390392

Convergence criterion (ABSGCONV=0.00001) satisfied.
```

The convergence status is important for the validity of your solution. In most cases, you should interpret your results only when the solution is converged. In this example, you obtain a converged solution, as shown in the message at the bottom of the table. The final objective function value is 0.01448, which is the minimized function value during the optimization. If problematic solutions such as nonconvergence are encountered, PROC TCALIS issues an error message.

The fit summary statistics are displayed in Output 88.1.6. By default, PROC TCALIS displays all available fit indices and modeling information.

Output 88.1.6 Fit Summary

Fit Summary		
Modeling Info	N Observations	932
	N Variables	6
	N Moments	21
	N Parameters	12
	N Active Constraints	0
	Independence Model Chi-Square	2131.4327
	Independence Model Chi-Square DF	15
Absolute Index	Fit Function	0.0145
	Chi-Square	13.4851
	Chi-Square DF	9
	Pr > Chi-Square	0.1419
	Z-Test of Wilson & Hilferty	1.0754
	Hoelter Critical N	1170
	Root Mean Square Residual (RMSR)	0.2281
	Standardized RMSR (SRMSR)	0.0150
	Goodness of Fit Index (GFI)	0.9953
Parsimony Index	Adjusted GFI (AGFI)	0.9890
	Parsimonious GFI	0.5972
	RMSEA Estimate	0.0231
	RMSEA Lower 90% Confidence Limit	.
	RMSEA Upper 90% Confidence Limit	0.0470
	Probability of Close Fit	0.9705
	ECVI Estimate	0.0405
	ECVI Lower 90% Confidence Limit	.
	ECVI Upper 90% Confidence Limit	0.0556
	Akaike Information Criterion	-4.5149
	Bozdogan CAIC	-57.0509
	Schwarz Bayesian Criterion	-48.0509
	McDonald Centrality	0.9976
Incremental Index	Bentler Comparative Fit Index	0.9979
	Bentler-Bonett NFI	0.9937
	Bentler-Bonett Non-normed Index	0.9965
	Bollen Normed Index Rho1	0.9895
	Bollen Non-normed Index Delta2	0.9979
	James et al. Parsimonious NFI	0.5962

First, the fit summary table starts with some basic modeling information, as shown in Output 88.1.6. You can check the number of observations, number of variables, number of moments being fitted, number of parameters, number of active constraints in the solution, and the independent model chi-square and its degrees of freedom in this modeling information category. Next, three types of fit indices are shown: absolute, parsimony, and incremental.

The absolute indices are fit measures that are interpreted without referring to any baseline model and without adjusting model parsimony. The chi-square test statistic is the best-known absolute index in this category. In this example, the p-value of the chi-square is 0.1419, which is larger than the conventional 0.05 value. From a statistical hypothesis testing point of view, this model cannot be rejected. The Z-test of Wilson and Hilferty is also insignificant at $\alpha = .05$, which echoes the result of the chi-square test. Other absolute indices can also be consulted. Although it seems that there are no clear conventional levels for these indices to indicate an acceptable model fit, you can always use these indices to compare the relative fit between different models.

Next, the parsimony fit indices take the model parsimony into account. These indices adjust the model fit by the degrees of freedom (or the number of the parameters) of the model in certain ways. The advantage of these indices is that merely increasing the number of parameters in the model might not necessarily improve the model fit indicated by these indices. Models with large numbers of parameters are penalized. There is no uniform way to interpret all these indices. However, for the relatively well-known RMSEA estimate, by convention values under 0.05 indicate good fit. The RMSEA value for this example is 0.0231, and so this is a very good fit. For interpretations of other parsimony indices, you can consult the original articles for these indices.

Last, the incremental fit indices are computed based on comparison with a baseline model, usually the independence model where all mainfest variables are assumed to be uncorrelated. The independence model fit statistic is shown under the Modeling Info category of the same fit summary table. In this example, the model fit chi-square of the independence model is 2131.43, with 15 degrees of freedom. The incremental indices show how well the hypothesized model improves over the independence model for the data. Various incremental fit indices have been proposed. In the fit summary table, there are six of such fit indices. Large values for these indices are desired. It has been suggested that values larger than .9 for these indices indicate acceptable model fit. In this example, all incremental indices but James et al. parsimonious NFI show that the hypothesized model fits well.

There is no consensus as to which fit index is the best to judge model fit. Probably, with artificial data and model, all fit indices can be shown defective in some aspects of measuring model fit. Conventional wisdom is to look at all fit indices and determine whether the majority of them are close to the desirable ranges of values. In this example, almost all fit indices are good, and so it is safe to conclude that the model fits well.

Nowadays, most researchers do pay less attention to the model fit chi-square statistic because it tends to reject all meaningful models with minimum departures from the truth. Although the model fit chi-square test statistic is an impeccable statistical inference tool when the underlying statistical assumptions are satisfied, for practical purposes it is just too powerful to accept any useful and reasonable models with only tiny imperfections. Some fit indices are more popular than others. Standardized RMSR, RMSEA estimate, adjusted AGFI, and Bentler's comparative fit index are frequently reported in empirical research for judging model fit. In this example, all these measures show good model fit of the hypothesized model. While there are certainly legitimate reasons why these fit indices are more popular than others, they are out of the current scope of discussion.

PROC TCALIS can perform a detailed residual analysis. Large residuals might indicate misspecification of the model. In Output 88.1.7, raw residuals are reported and ranked.

Output 88.1.7 Raw Residuals and Ranking

Raw Residual Matrix

		Anomie67	Powerless67	Anomie71
Anomie67	Anomie (1967)	−0.06997	0.03642	−0.01116
Powerless67	Powerlessness (1967)	0.03642	0.01261	0.15600
Anomie71	Anomie (1971)	−0.01116	0.15600	−0.08381
Powerless71	Powerlessness (1971)	−0.15200	0.01135	−0.00854
Education	Education	0.32892	−0.41712	0.22464
SEI	Occupational Status Index	0.47786	−0.19108	0.07976

Raw Residual Matrix

		Powerless71	Education	SEI
Anomie67	Anomie (1967)	−0.15200	0.32892	0.47786
Powerless67	Powerlessness (1967)	0.01135	−0.41712	−0.19108
Anomie71	Anomie (1971)	−0.00854	0.22464	0.07976
Powerless71	Powerlessness (1971)	0.14067	−0.23832	−0.59248
Education	Education	−0.23832	0	4.03148E−6
SEI	Occupational Status Index	−0.59248	4.03148E−6	0.0000221

Average Absolute Residual	0.153940
Average Off-diagonal Absolute Residual	0.195044

Rank Order of the 10 Largest Raw Residuals

Var1	Var2	Residual
SEI	Powerless71	−0.59248
SEI	Anomie67	0.47786
Education	Powerless67	−0.41712
Education	Anomie67	0.32892
Education	Powerless71	−0.23832
Education	Anomie71	0.22464
SEI	Powerless67	−0.19108
Anomie71	Powerless67	0.15600
Powerless71	Anomie67	−0.15200
Powerless71	Powerless71	0.14067

Because of the differential scaling of the variables, it is usually more useful to examine the standardized residuals instead. In Output 88.1.8, for example, the table for the 10 largest asymptotically standardized residuals is displayed.

Output 88.1.8 Asymptotically Standardized Residuals and Ranking

Asymptotically Standardized Residual Matrix

		Anomie67	Powerless67	Anomie71
Anomie67	Anomie (1967)	-0.30882	0.52686	-0.05619
Powerless67	Powerlessness (1967)	0.52686	0.05464	0.87613
Anomie71	Anomie (1971)	-0.05619	0.87613	-0.35460
Powerless71	Powerlessness (1971)	-0.86507	0.05735	-0.12169
Education	Education	2.55338	-2.76371	1.69781
SEI	Occupational Status Index	0.46484	-0.17015	0.07009

Asymptotically Standardized Residual Matrix

		Powerless71	Education	SEI
Anomie67	Anomie (1967)	-0.86507	2.55338	0.46484
Powerless67	Powerlessness (1967)	0.05735	-2.76371	-0.17015
Anomie71	Anomie (1971)	-0.12169	1.69781	0.07009
Powerless71	Powerlessness (1971)	0.58521	-1.55750	-0.49608
Education	Education	-1.55750	0	0
SEI	Occupational Status Index	-0.49608	0	0

Average Standardized Residual	0.646672
Average Off-diagonal Standardized Residual	0.818456

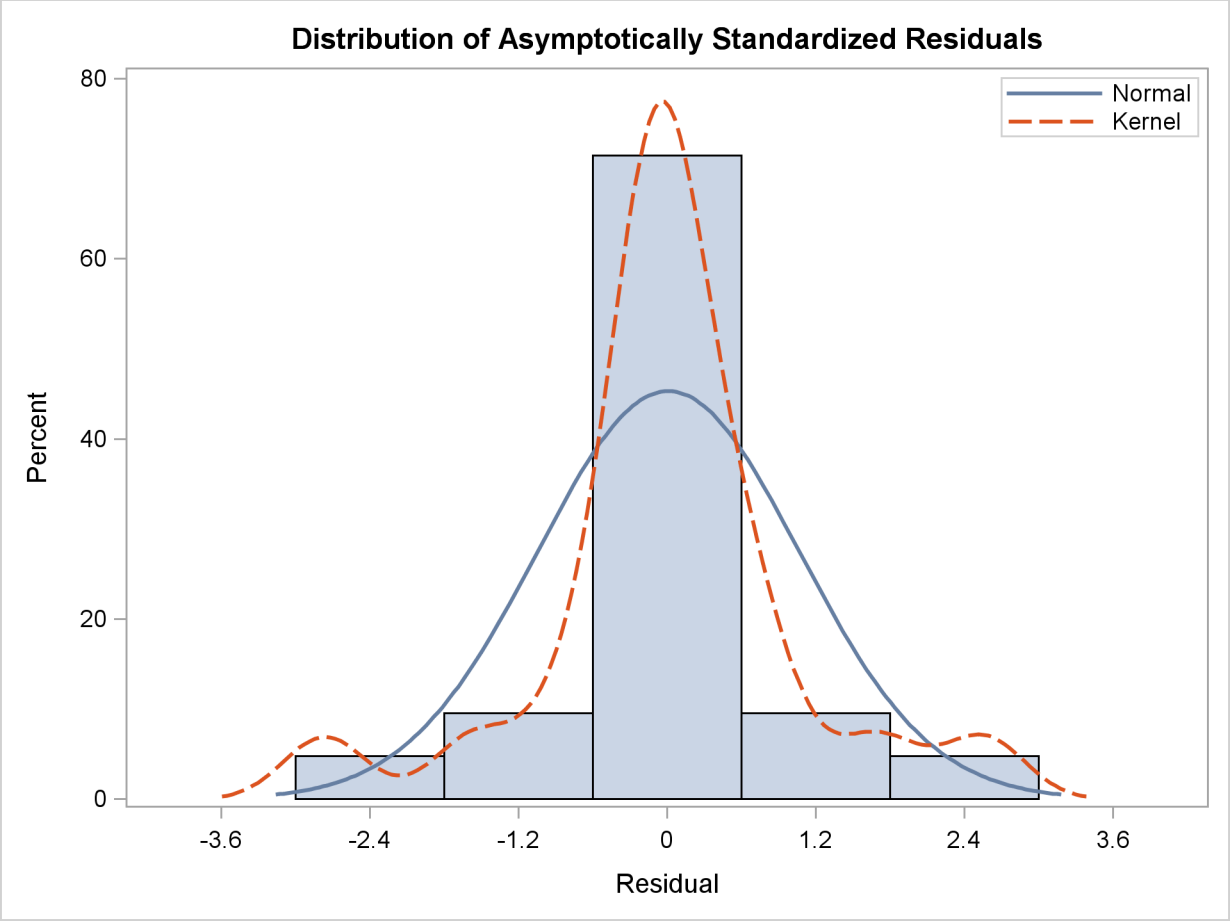
Rank Order of the 10 Largest Asymptotically Standardized Residuals

Var1	Var2	Residual
Education	Powerless67	-2.76371
Education	Anomie67	2.55338
Education	Anomie71	1.69781
Education	Powerless71	-1.55750
Anomie71	Powerless67	0.87613
Powerless71	Anomie67	-0.86507
Powerless71	Powerless71	0.58521
Powerless67	Anomie67	0.52686
SEI	Powerless71	-0.49608
SEI	Anomie67	0.46484

The model performs the poorest concerning the covariances of Education with all measures of Powerless and Anomie. This might suggest a misspecification of the functional relationships of Education with other variables in the model. However, because the model fit is quite good, such a possible misspecification should not be a serious concern in the analysis.

The histogram of the asymptotically standardized residuals is displayed in Output 88.1.9, which also shows the normal and kernel approximations.

Output 88.1.9 Distribution of Asymptotically Standardized Residuals



The residual distribution looks quite symmetrical. It shows a small to medium departure from the normal distribution, as evidenced by the discrepancies between the kernel and the normal distribution curves.

Estimation results are shown in Output 88.1.10.

Output 88.1.10 Estimation Results

PATH List

Path			Parameter	Estimate	Standard Error	t Value
Anomie67	<-	Alien67		1.00000		
Powerless67	<-	Alien67		0.83300		
Anomie71	<-	Alien71		1.00000		
Powerless71	<-	Alien71		0.83300		
Education	<-	SES		1.00000		
SEI	<-	SES	lambda	5.36883	0.43371	12.37880
Alien67	<-	SES	gamma1	-0.62994	0.05634	-11.18092
Alien71	<-	SES	gamma2	-0.24086	0.05489	-4.38836
Alien71	<-	Alien67	beta	0.59312	0.04678	12.67884

Output 88.1.10 *continued*

Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	Anomie67	theta1	3.60796	0.20092	17.95717
	Powerless67	theta2	3.59488	0.16448	21.85563
	Anomie71	theta1	3.60796	0.20092	17.95717
	Powerless71	theta2	3.59488	0.16448	21.85563
	Education	theta3	2.99366	0.49861	6.00398
	SEI	theta4	259.57639	18.31151	14.17559
	Alien67	psi1	5.67046	0.42301	13.40500
	Alien71	psi2	4.51479	0.33532	13.46394
Exogenous	SES	phi	6.61634	0.63914	10.35190

Covariances Among Errors

Error of	Error of	Parameter	Estimate	Standard Error	t Value
Anomie67	Anomie71	theta5	0.90580	0.12167	7.44472
Powerless67	Powerless71	theta5	0.90580	0.12167	7.44472

The paths, variances and partial (or error) variances, and covariances and partial covariances are shown. When you have fixed parameters such as the first five path coefficients in the output, the standard errors and t values are all blanks. For free or constrained estimates, standard errors and t values are computed. Researchers in structural equation modeling usually use the value 2 as an approximate critical value for the observed t values. The reason is that the estimates are asymptotically normal, and so the two-sided critical point with $\alpha = 0.05$ is 1.96, which is close to 2. Using this criterion, all estimates shown in Output 88.1.10 are significantly different from zero, supporting the presence of these parameters in the model.

Squared multiple correlations are shown in Output 88.1.11.

Output 88.1.11 Squared Multiple Correlations

Squared Multiple Correlations

Variable	Error Variance	Total Variance	R-Square
Anomie67	3.60796	11.90397	0.6969
Anomie71	3.60796	12.61581	0.7140
Education	2.99366	9.61000	0.6885
Powerless67	3.59488	9.35139	0.6156
Powerless71	3.59488	9.84533	0.6349
SEI	259.57639	450.28798	0.4235
Alien67	5.67046	8.29601	0.3165
Alien71	4.51479	9.00786	0.4988

For each endogenous variable in the model, the corresponding squared multiple correlation is computed by:

$$1 - \frac{\text{error variance}}{\text{total variance}}$$

In regression analysis, this is the percentage of explained variance of the endogenous variable by the predictors. However, this interpretation is complicated or even uninterpretable when your structural equation model has correlated errors or reciprocal casual relations. In these situations, it is not uncommon to see negative R-squares. Negative R-squares do not necessarily mean that your model is wrong or the model prediction is weak. Rather, the R-square interpretation is questionable in these situations.

When your variables are measured on different scales, comparison of path coefficients cannot be made directly. For example, in Output 88.1.10, the path coefficient for path Education <– SES is fixed at one, while the path coefficient for path SEI <– SES is 5.369. It would be very simple-minded to conclude that the effect of SES on SEI is larger than that SES on Education. Because SEI and Education are measured on different scales, direct comparison of the corresponding path coefficients is simply inappropriate.

In alleviating this problem, some might resort to the standardized solution for a better comparison. In a standardized solution, because the variances of manifest variables and systematic predictors are all standardized to ones, you hope the path coefficients are more comparable. In this example, PROC TCALIS standardizes your results in Output 88.1.12.

Output 88.1.12 Standardized Results

Standardized Results for PATH List

Path			Parameter	Estimate	Standard Error	t Value
Anomie67	<-	Alien67		0.83481	0.01093	76.35313
Powerless67	<-	Alien67		0.78459	0.01163	67.47756
Anomie71	<-	Alien71		0.84499	0.01031	81.97956
Powerless71	<-	Alien71		0.79678	0.01107	71.96263
Education	<-	SES		0.82975	0.03172	26.15990
SEI	<-	SES	lambda	0.65079	0.03019	21.55331
Alien67	<-	SES	gamma1	-0.56257	0.03456	-16.27961
Alien71	<-	SES	gamma2	-0.20642	0.04483	-4.60430
Alien71	<-	Alien67	beta	0.56920	0.04066	14.00001

Output 88.1.12 *continued*

Standardized Results for Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	Anomie67	theta1	0.30309	0.01825	16.60309
	Powerless67	theta2	0.38442	0.01825	21.06948
	Anomie71	theta1	0.28599	0.01742	16.41782
	Powerless71	theta2	0.36514	0.01764	20.69424
	Education	theta3	0.31152	0.05264	5.91822
	SEI	theta4	0.57647	0.03930	14.66804
	Alien67	psi1	0.68352	0.03888	17.57968
	Alien71	psi2	0.50121	0.03321	15.08974
Exogenous	SES	phi	1.00000		

Standardized Results for Covariances Among Errors

Error of	Error of	Parameter	Estimate	Standard Error	t Value
Anomie67	Anomie71	theta5	0.07391	0.01013	7.29574
Powerless67	Powerless71	theta5	0.09440	0.01274	7.40922

Now, the standardized path coefficient for path Education <– SES is 0.830, while the standardized path coefficient for path SEI <– SES is 0.651. So the standardized effect of SES on SEI is actually smaller than that of SES on Education.

Furthermore, in PROC TCALIS the standardized estimates are computed with standard error estimates and t values so that you can make statistical inferences on the standardized estimates as well.

PROC TCALIS might differ from other software in its standardization scheme. Unlike other software that might standardize the path coefficients that attach to the error terms (unsystematic sources), PROC TCALIS keeps these path coefficients at ones (not shown in the output). Unlike other software that might also standardize the corresponding error variances to ones, the error variances in the standardized solution of PROC TCALIS are rescaled so as to keep the mathematical consistency of the model.

Essentially, in PROC TCALIS only variances of mainfest and non-error-type latent variables are standardized to ones. The error variances are rescaled, but not standardized. For example, in the standardized solution shown in Output 88.1.12, the error variances for all endogenous variables are not ones (see the middle portion of the output). Only the variance for the latent variable SES is standardized to one. See the section "Standardized Solutions" on page 6915 for the logic of the standardization scheme adopted by PROC TCALIS.

In appearance, the standardized solution is like a correlational analysis on the standardized manifest variables with standardized exogenous latent factors. Unfortunately, this statement is oversimplified, if not totally inappropriate. In standardizing a solution, the implicit equality constraints are likely destroyed. In this example, the unstandardized error variances for Anomie67 and Anomie71 are both 3.608, represented by a common parameter theta1. However, after standardization, these

error variances have different values at 0.303 and 0.286, respectively. In addition, fixed parameter values are no longer fixed in a standardized solution (for example, the first five paths in the current example). The issue of standardization is common to all other SEM software and beyond the current discussion. PROC TCALIS provides the standardized solution so that users can interpret the standardized estimates whenever they find them appropriate.

Example 88.2: Simultaneous Equations with Mean Structures and Reciprocal Paths

The supply-and-demand food example of Kmenta (1971, pp. 565, 582) is used to illustrate PROC TCALIS for the estimation of intercepts and coefficients of simultaneous equations in econometrics. The model is specified by two simultaneous equations containing two endogenous variables Q and P, and three exogenous variables D, F, and Y:

$$Q_t(demand) = \alpha_1 + \beta_1 P_t + \gamma_1 D_t$$

$$Q_t(supply) = \alpha_2 + \beta_2 P_t + \gamma_2 F_t + \gamma_3 Y_t$$

for $t = 1, \ldots, 20$.

To analyze this model in PROC TCALIS, the second equation needs to be written in another form. For instance, in the LINEQS model each endogenous variable must appear on the left-hand side of exactly one equation. To satisfy this requirement, you can rewrite the second equation as an equation for P_t as:

$$P_t = -\frac{\alpha_2}{\beta_2} + \frac{1}{\beta_2} Q_t - \frac{\gamma_2}{\beta_2} F_t - \frac{\gamma_3}{\beta_2} Y_t$$

or, equivalently reparameterized as:

$$P_t = \theta_1 + \theta_2 Q_t + \theta_3 F_t + \theta_4 Y_t$$

where

$$\theta_1 = -\frac{\alpha_2}{\beta_2}, \quad \theta_2 = \frac{1}{\beta_2}, \quad \theta_3 = -\frac{\gamma_2}{\beta_2}, \quad \theta_4 = -\frac{\gamma_3}{\beta_2}$$

This new equation for P_t together with the first equation for Q_t suggest the following LINEQS model specification in PROC TCALIS:

```
title 'Food example of KMENTA(1971, p.565 & 582)';
data food;
  input Q P D F Y;
  label  Q='Food Consumption per Head'
         P='Ratio of Food Prices to General Price'
         D='Disposable Income in Constant Prices'
         F='Ratio of Preceding Years Prices'
         Y='Time in Years 1922-1941';
datalines;
  98.485  100.323   87.4   98.0   1
  99.187  104.264   97.6   99.1   2
 102.163  103.435   96.7   99.1   3
 101.504  104.506   98.2   98.1   4
 104.240   98.001   99.8  110.8   5
 103.243   99.456  100.5  108.2   6
 103.993  101.066  103.2  105.6   7
  99.900  104.763  107.8  109.8   8
 100.350   96.446   96.6  108.7   9
 102.820   91.228   88.9  100.6  10
  95.435   93.085   75.1   81.0  11
  92.424   98.801   76.9   68.6  12
  94.535  102.908   84.6   70.9  13
  98.757   98.756   90.6   81.4  14
 105.797   95.119  103.1  102.3  15
 100.225   98.451  105.1  105.0  16
 103.522   86.498   96.4  110.5  17
  99.929  104.016  104.4   92.5  18
 105.223  105.769  110.7   89.3  19
 106.232  113.490  127.1   93.0  20
;
proc tcalis data=food pshort nostand;
   lineqs
      Q = alpha1 Intercept + beta1  P  + gamma1 D + E1,
      P = theta1 Intercept + theta2 Q  + theta3 F + theta4 Y + E2;
   std
      E1-E2 = eps1-eps2;
   cov
      E1-E2 = eps3;
   bounds
      eps1-eps2 >= 0. ;
run;
```

The LINEQS modeling language is used in this example because its specification is similar to the original equations. In the LINEQS statement, you essentially input the two model equations for Q and P. Parameters for intercepts and regression coefficients are also specified in the equations. Note that Intercept in the two equations is treated as a special variable that contains ones for all observations. Intercept is not a variable in the data set, nor do you need to create such a variable in your data set. Hence, the variable Intercept does not represent the intercept parameter itself. Instead, the intercept parameters for the two equations are the coefficients attached to Intercept. In this example, the intercept parameters are alpha1 and theta1, respectively, in the two equations. As required, error terms E1 and E2 are added to complete the equation specification.

In the STD statement, you specify eps1 and eps2, respectively, for the variance parameters of the error terms. In the COV, you specify eps3 for the covariance parameter between the error terms. In the BOUNDS statement, you set lower bounds for the error variances so that estimates of eps1 and eps2 would be nonnegative.

In this example, the PSHORT and the NOSTAND options are used in the PROC TCALIS statement. The PSHORT option suppresses a large amount of the output. For example, initial estimates are not printed and simple descriptive statistics and standard errors are not computed. The NOSTAND option suppresses the printing of the standardized results. Because the default printing in PROC TCALIS might produce a large amount of output, using these printing options make your output more concise and readable. Whenever appropriate, it is recommended that you consider using these printing options.

The estimated equations are shown in Output 88.2.1.

Output 88.2.1 Linear Equations

```
                          Linear Equations

Q       =   93.6193*Intercept + -0.2295*P         +  0.3100*D
Std Err      7.5748 alpha1       0.0923 beta1        0.0448 gamma1
t Value     12.3592             -2.4856              6.9186

               +  1.0000 E1



                          Linear Equations

P       =    -218.9*Intercept +  4.2140*Q         + -0.9305*F
Std Err       137.7 theta1       1.7540 theta2       0.3960 theta3
t Value     -1.5897              2.4025             -2.3500

               + -1.5579*Y         +  1.0000 E2
                  0.6650 theta4
                 -2.3429
```

The estimates of intercepts and regression coefficients are shown directly in the equations. Any number in an equation followed by an asterisk is an estimate. For the estimates in equations, the parameter names are shown underneath the associated variables. Any number in an equation not followed by an asterisk is a fixed value. For example, the value 1.0000 attached to the error term in each of the output equation is fixed. Also, for fixed coefficients there are no parameter names underneath the associated variables.

All but the intercept estimates in the equation for predicting P are statistically significant at $\alpha = 0.05$ (when using an approximate critical value of 2). The t ratio for theta1 is -1.590, which implies that this intercept might have been zero in the population. However, because you have reparameterized the original model to use the LINEQS model specification, transformed parameters like theta1 in this model might not be of primary interest. Therefore, you might not need to pay any attention to the significance of the theta1 estimate. There is a way to use the original econometric parameters to specify the LINEQS model. It is discussed in the later part of this example.

Estimates for variance, covariance, and mean parameters are shown in Output 88.2.2.

Output 88.2.2 Variance, Covariance, and Mean Parameters

Estimates for Variances of Exogenous Variables

Variable Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	E1	eps1	3.51274	1.20204	2.92233
	E2	eps2	105.06749	83.89446	1.25238
Observed	D	_Add1	139.96029	45.40911	3.08221
	F	_Add2	161.51355	52.40192	3.08221
	Y	_Add3	35.00000	11.35550	3.08221

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
E1	E2	eps3	-18.87270	8.77951	-2.14963
F	D	_Add4	74.02539	38.44699	1.92539
Y	D	_Add5	22.99211	16.90102	1.36040
Y	F	_Add6	-21.58158	17.94544	-1.20262

Mean Parameters

Variable Type	Variable	Parameter	Estimate	Standard Error	t Value
Observed	D	_Add7	97.53500	2.71410	35.93643
	F	_Add8	96.62500	2.91560	33.14071
	Y	_Add9	10.50000	1.35724	7.73628

Parameters with a name prefix _Add are added automatically by PROC TCALIS. These parameters are added as free parameters to complete the model specification. In PROC TCALIS, variances and covariances among the set of exogenous manifest variables must be parameters. You either specify them explicitly or let the TCALIS procedure to add them. If you need to constrain or to fix these parameters, then you must specify them explicitly. When your model also fits the mean structures, the same principle applies to the means of the exogenous manifest variables. In this example, because variables D, F, and Y are all exogenous manifest variables, their associated means, variances and covariances must be parameters in the model.

The squared multiple correlations for the equations are shown in Output 88.2.3.

Output 88.2.3 Squared Multiple Correlations

Squared Multiple Correlations

Variable	Error Variance	Total Variance	R-Square
Q	3.51274	14.11128	0.7511
P	105.06749	35.11850	-1.9918

For endogenous variable P, the R-square is -1.9918, which is obviously an invalid value. In fact, because there are correlated errors (between E1 and E2) and reciprocal paths (paths to and from Q and P), the model departs from the regular assumptions of multiple regression analysis. As a result, you should not interpret the R-squares for this example.

If you are interested in estimating the parameters in the original econometric model (that is, α_2, β_2, γ_2, and γ_3), the previous reparameterized LINEQS model does not serve your purpose well enough. However, using the relations between these original parameters with the θ parameters in the reparameterized LINEQS model, you can set up some "super-parameters" in the LINEQS model as follows:

```
proc tcalis data=Food pshort nostand;
   lineqs
      Q = alpha1 Intercept + beta1  P  + gamma1 D + E1,
      P = theta1 Intercept + theta2 Q  + theta3 F + theta4 Y + E2;
   std
      E1-E2 = eps1-eps2;
   cov
      E1-E2 = eps3;
   bounds
      eps1-eps2 >= 0. ;
   parameters alpha2 (50.) beta2 gamma2 gamma3 (3*.25);
      theta1  = -alpha2 / beta2;
      theta2  = 1 / beta2;
      theta3  = -gamma2 / beta2;
      theta4  = -gamma3 / beta2;
run;
```

In this new specification, only the PARAMETERS statement and the SAS programming statements following it are new. In the PARAMETERS statement, you define super-parameters alpha2, beta2, gamma2, and gamma3, and put initial values for them in parentheses. These parameters are the original econometric parameters of interest. The SAS programming statements that follow the PARAMETERS statement are used to define the functional relationships of the super-parameters with the parameters in the LINEQS model. Consequently, in this new specification, theta1, theta2, theta3, and theta4 are no longer independent parameters in the model, as they are in the previous reparameterized model. Instead, alpha2, beta2, gamma2, and gamma3 are independent parameters in this new specification. By fitting this new model, you get the same set of estimates as those in the previous LINEQS model. In addition, you get estimates of the super-parameters, as shown in Output 88.2.4.

Output 88.2.4 Additional Parameters

Additional Parameters

Type	Parameter	Estimate	Standard Error	t Value
Independent	alpha2	51.94452	11.70002	4.43969
	beta2	0.23731	0.09877	2.40262
	gamma2	0.22082	0.04161	5.30695
	gamma3	0.36971	0.07060	5.23649

You can now interpret the results in terms of the original econometric parameterization. As shown in Output 88.2.4, all these estimates are significant, despite the fact that one of the transformed parameter estimates in the linear equations of the LINEQS model is not. You can obtain almost equivalent results by applying the SAS/ETS procedure SYSLIN on this problem.

Example 88.3: A Direct Covariance Structures Model

In the section "Direct Covariance Structures Analysis" on page 6712, the MSTRUCT modeling language is used to specify a model with direct covariance structures. In the model, four variables from the data set of Wheaton et al. (1977) are used. The analysis is carried out in this example to investigate the tenability of the hypothesized covariance structures.

The four variables used are: Anomie67, Powerless67, Anomie71, and Powerless71. The hypothesized covariance matrix is structured as:

$$\Sigma = \begin{pmatrix} \phi_1 & \theta_1 & \theta_2 & \theta_1 \\ \theta_1 & \phi_2 & \theta_l & \theta_3 \\ \theta_2 & \theta_1 & \phi_1 & \theta_1 \\ \theta_1 & \theta_3 & \theta_1 & \phi_2 \end{pmatrix}$$

where:

ϕ_1: variance of anomie

ϕ_2: variance of powerlessness

θ_1: covariance between anomie and powerlessness

θ_2: covariance between anomie measures

θ_3: covariance between powerlessness measures

In this example, you hypothesize the covariance structures directly, as opposed to those models with implied covariance structures from path models (see Example 88.1), structural equations (see Example 88.2), or other types of models. The basic assumption of the direct covariance structures in this example is that Anomie and Powerless were invariant over the measurement periods employed. This implies that the time of measurement did not change the variances and covariances of the

measures. Therefore, both Anomie67 and Anomie71 have the same variance parameter ϕ_1, and both Powerless67 and Powerless71 have the same variance parameter ϕ_2. These two parameters, ϕ_1 and ϕ_2, are hypothesized on the diagonal of the covariance matrix Σ. In the same structured covariance matrix, θ_1 represents the covariance between Anomie and Powerless, without regard to the time of measurement. The θ_2 parameter represents the covariance between the Anomie measures, or the reliability of the Anomie measure. Similarly, the θ_2 parameter represents the covariance between the Powerless measures, or the reliability of the Anomie measure.

As explained in the section "Direct Covariance Structures Analysis" on page 6712, you can use the MSTRUCT modeling language to specify the hypothesized covariance structures directly, as shown in the following statements:

```
proc tcalis nobs=932 data=Wheaton psummary;
   fitindex on(only)=[chisq df probchi] outfit=savefit;
   mstruct
      var = Anomie67 Powerless67 Anomie71 Powerless71;
   matrix _COV_ [1,1] = phi1,
                [2,2] = phi2,
                [3,3] = phi1,
                [4,4] = phi2,
                [2,1] = theta1,
                [3,1] = theta2,
                [3,2] = theta1,
                [4,1] = theta1,
                [4,2] = theta3,
                [4,3] = theta1;
run;
```

In the MSTRUCT statement you specify the variables in the VAR= list. The order of variables in this VAR= list is assumed to be the same as that in the row and column of the hypothesized covariance matrix. Next, in the MATRIX statement you specify parameters as entries in the hypothesized covariance matrix _COV_. Only the lower diagonal elements need to be specified because covariance matrices, by nature, are symmetric. Redundant specification of the upper triangular elements are unnecessary as PROC TCALIS has the information accounted for. You can also set initial estimates by putting parenthesized numbers after the parameter names. But in this example you let PROC TCALIS determine all the initial estimates.

In the PROC TCALIS statement, the PSUMMARY option is used. As a global display option, this option suppresses a lot of displayed output and requests only the fit summary table be printed. This way you can eliminate quite a lot of displayed output that is not of your primary interest. In this example, the specification of the covariance structures is straightforward, and you do not need any output regarding the initial estimation or standardized solution. Suppose that you are not even concerned with the estimates of the parameters because you are not yet sure if this model is good enough for the data. All you want to know at this stage is whether the hypothesized covariance structures fit the data well. Therefore, the PSUMMARY option would serve your purpose well in this example.

In fact, even the fit summary table can be trimmed down quite a bit if you only want to look at certain specific fit indices. In the FITINDEX statement of this example, the ON(ONLY)= option turns on the printing of the model fit chi-square, its df, and p-value only. This does not mean that you must lose the information of all other fit indices. In addition to the printed output, you can save

all fit indices in an output data set. To this end, you can use the OUTFIT= option in the FITINDEX statement. In this example, you save the results of all fit indices in a SAS data set called savefit.

The entire printed output is shown in Output 88.3.1.

Output 88.3.1 Testing Direct Covariance Structures

Fit Summary

Chi-Square	221.5798
Chi-Square DF	5
Pr > Chi-Square	0.0000

As you can see, the displayed output is very precise. It contains only a fit summary table with three statistics. The p-value for the mode fit chi-square test indicates that the hypothesized structures should be rejected at $\alpha = 0.05$. Therefore, this rather restrictive direct covariance structure model does not fit the data well. A less restrictive covariance structure model with more parameters is needed to better account for the variances and covariances of the variables.

All fit indices are saved in the savefit data set. To view it, you can use the following statement:

```
proc print data=savefit;
run;
```

All indices, their types and values are shown in Output 88.3.2.

Output 88.3.2 Saved Fit Indices

Analysis of Direct Covariance Structures
Testing Model by the MSTRUCT Language

Obs	_TYPE_	FitIndex	Fit Value	PrintChar
1	ModelInfo	N Observations	932.00	932
2	ModelInfo	N Variables	4.00	4
3	ModelInfo	N Moments	10.00	10
4	ModelInfo	N Parameters	5.00	5
5	ModelInfo	N Active Constraints	0.00	0
6	ModelInfo	Independence Model Chi-Square	1563.94	1563.9442
7	ModelInfo	Independence Model Chi-Square DF	6.00	6
8	Absolute	Fit Function	0.24	0.2380
9	Absolute	Chi-Square	221.58	221.5798
10	Absolute	Chi-Square DF	5.00	5
11	Absolute	Pr > Chi-Square	0.00	0.0000
12	Absolute	Elliptic Corrected Chi-Square	.	.
13	Absolute	Pr > Elliptic Corr. Chi-Square	.	.
14	Absolute	Z-Test of Wilson & Hilferty	12.25	12.2533
15	Absolute	Hoelter Critical N	48.00	48
16	Absolute	Root Mean Square Residual (RMSR)	0.76	0.7649
17	Absolute	Standardized RMSR (SRMSR)	0.07	0.0701
18	Absolute	Goodness of Fit Index (GFI)	0.90	0.9036
19	Parsimony	Adjusted GFI (AGFI)	0.81	0.8071
20	Parsimony	Parsimonious GFI	0.75	0.7530
21	Parsimony	RMSEA Estimate	0.22	0.2157
22	Parsimony	RMSEA Lower 90% Confidence Limit	0.19	0.1920
23	Parsimony	RMSEA Upper 90% Confidence Limit	0.24	0.2404
24	Parsimony	Probability of Close Fit	0.00	0.0000
25	Parsimony	ECVI Estimate	0.25	0.2488
26	Parsimony	ECVI Lower 90% Confidence Limit	0.20	0.2003
27	Parsimony	ECVI Upper 90% Confidence Limit	0.31	0.3053
28	Parsimony	Akaike Information Criterion	211.58	211.5798
29	Parsimony	Bozdogan CAIC	182.39	182.3932
30	Parsimony	Schwarz Bayesian Criterion	187.39	187.3932
31	Parsimony	McDonald Centrality	0.89	0.8903
32	Incremental	Bentler Comparative Fit Index	0.86	0.8610
33	Incremental	Bentler-Bonett NFI	0.86	0.8583
34	Incremental	Bentler-Bonett Non-normed Index	0.83	0.8332
35	Incremental	Bollen Normed Index Rho1	0.83	0.8300
36	Incremental	Bollen Non-normed Index Delta2	0.86	0.8611
37	Incremental	James et al. Parsimonious NFI	0.72	0.7153

The results of various fit indices from this output data set confirm that the hypothesized model does not fit the data well.

As an aside, it is noted with some shorthand notation, the specification of the MSTRUCT model parameters that use the MATRIX statements can be made a little more precise for the current example. This is shown as follows:

```
proc tcalis nobs=932 data=Wheaton psummary;
   fitindex on(only)=[chisq df probchi] outfit=savefit;
   mstruct
      var = Anomie67 Powerless67 Anomie71 Powerless71;
   matrix _COV_ [1,1] = phi1 phi2 phi1 phi2;
                [2, ] = theta1,
                [3, ] = theta2 theta1,
                [4, ] = theta1 theta3 theta1;
run;
```

In the first entry of the MATRIX statement, the notation [1,1] represents that the parameter list specified after the equal sign starts with the [1,1] element of the _COV_ matrix and proceeds down the diagonal. In the next three entries, the notations [2,], [3,], and [4,] represent that parameter lists start with the first elements of the second, third, and fourth rows, respectively, and proceed to the next (right) elements on the same rows. See the syntax of the MATRIX statement on page 6784 for more details about this kind of shorthand notation.

Example 88.4: Confirmatory Factor Analysis: Cognitive Abilities

In this example, cognitive abilities of 64 students from a middle school were measured. The fictitious data contain nine cognitive test scores. Three of the scores were for reading skills, three others were for math skills, and the remaining three were for writing skills. The covariance matrix for the nine variables was obtained. A confirmatory factor analysis with three factors was conducted. The following is the input data set and the PROC TCALIS specification for the analysis:

```
title "Confirmatory Factor Analysis Using the FACTOR Modeling Language";
title2 "Cognitive Data";
data cognitive1(type=cov);
   _type_='cov';
   input _name_ $ reading1 reading2 reading3 math1 math2 math3
         writing1 writing2 writing3;
   datalines;
reading1 83.024    .      .      .      .      .      .      .      .
reading2 50.924 108.243   .      .      .      .      .      .      .
reading3 62.205  72.050 99.341   .      .      .      .      .      .
math1    22.522  22.474 25.731 82.214   .      .      .      .      .
math2    14.157  22.487 18.334 64.423 96.125   .      .      .      .
math3    22.252  20.645 23.214 49.287 58.177 88.625   .      .      .
writing1 33.433  42.474 41.731 25.318 14.254 27.370 90.734   .      .
writing2 24.147  20.487 18.034 22.106 26.105 22.346 53.891 96.543   .
writing3 13.340  20.645 23.314 19.387 28.177 38.635 55.347 52.999 98.445
;
```

```
proc tcalis data=cognitive1 nobs=64 modification;
   factor
      Read_Factor   -> reading1-reading3  = load1-load3,
      Math_Factor   -> math1-math3        = load4-load6,
      Write_Factor  -> writing1-writing3  = load7-load9;
   pvar
      Read_Factor Math_Factor Write_Factor = 3 * 1.,
      reading1-reading3 math1-math3 writing1-writing3 = errvar1-errvar9;
run;
```

In the PROC TCALIS statement, the number of observations is specified with the NOBS= option. The MODIFICATION option in the same statement requests model modification indices be computed.

The FACTOR modeling language is the most handy tool for specifying confirmatory factor models. The FACTOR statement is used to invoke this modeling language. Entries in the FACTOR statement are for specifying factor-variables relationships and are separated by commas. In each entry, you first specify a latent factor, followed by the right arrow sign –>. Then you specify the observed variables that have nonzero loadings on the factor. Next, an equal sign is used to signify the specification of the loading parameters that follow. The loading parameters can be names (parameters without initial estimates), numbers (fixed values), or names followed by parenthesized numbers (parameter with initial values). In this example, there are three factors: Read_Factor, Math_Factor, and Write_Factor. These factors have simple cluster structures with the nine observed variables. Each observed variable has only one loading on exactly one factor, yielding a total of nine loading parameters named load1–load9. No initial estimates are specified for them. They are computed by PROC TCALIS.

In the PVAR statement, you specify the variances of the factors and the error variances for the observed variables. The factor variances in this model are fixed at ones for identification purposes. The error variances for the observed variables are free parameters without initial estimates, named errvar1–errvar9, respectively.

The covariances of the factors are not specified in the model, meaning that the factors are uncorrelated by default. Certainly, this might not be reasonable. But for illustration purposes, this uncorrelated factor model is fitted. With the MODIFICATION option in the PROC TCALIS statement, LM (Lagrange Multiplier) tests are conducted. The results of LM tests can suggest the inclusion of additional parameters for a better model fit. If the uncorrelated factor model is indeed unreasonable, it is shown in the results of the LM tests.

In Output 88.4.1, the initial model specification is echoed in matrix form. The observed variables and factors are also displayed.

Output 88.4.1 Uncorrelated Factor Model Specification

Variables in the Model

Variables	reading1	reading2	reading3	math1	math2	math3
	writing1	writing2	writing3			
Factors	Read_Factor	Math_Factor	Write_Factor			

Number of Variables = 9
Number of Factors = 3

Initial Factor Loading Matrix

	Read_Factor	Math_Factor	Write_Factor
reading1	. [load1]	0	0
reading2	. [load2]	0	0
reading3	. [load3]	0	0
math1	0	. [load4]	0
math2	0	. [load5]	0
math3	0	. [load6]	0
writing1	0	0	. [load7]
writing2	0	0	. [load8]
writing3	0	0	. [load9]

Initial Factor Covariance Matrix

	Read_Factor	Math_Factor	Write_Factor
Read_Factor	1.0000	0	0
Math_Factor	0	1.0000	0
Write_Factor	0	0	1.0000

Output 88.4.1 *continued*

Initial Error Variances

Variable	Parameter	Estimate
reading1	errvar1	.
reading2	errvar2	.
reading3	errvar3	.
math1	errvar4	.
math2	errvar5	.
math3	errvar6	.
writing1	errvar7	.
writing2	errvar8	.
writing3	errvar9	.

In the table for initial factor loading matrix, the nine loading parameters are shown to have simple cluster relations with the factors. In the table for initial factor covariance matrix, the diagonal matrix shows that the factors are not correlated. The diagonal elements are fixed at ones so that this matrix is also a correlation matrix for the factors. In the table for initial error variances, the nine variance parameters are shown. No initial estimates were specified, as indicated by the missing values '.'.

In Output 88.4.2, initial estimates are generated by the instrumental variable method and the McDonald method.

Output 88.4.2 Optimization of the Uncorrelated Factor Model: Initial Estimates

Initial Estimation Methods

1	Instrumental Variables Method
2	McDonald Method

Output 88.4.2 *continued*

Optimization Start
Parameter Estimates

N	Parameter	Estimate	Gradient
1	load1	7.15372	0.00851
2	load2	7.80225	−0.00170
3	load3	8.70856	−0.00602
4	load4	7.68637	0.00272
5	load5	8.01765	−0.01096
6	load6	7.05012	0.00932
7	load7	8.76776	−0.0009955
8	load8	5.96161	−0.01335
9	load9	7.23168	0.01665
10	errvar1	31.84831	−0.00179
11	errvar2	47.36790	0.0003461
12	errvar3	23.50199	0.00257
13	errvar4	23.13374	−0.0008384
14	errvar5	31.84224	0.00280
15	errvar6	38.92075	−0.00167
16	errvar7	13.86035	−0.00579
17	errvar8	61.00217	0.00115
18	errvar9	46.14784	−0.00300

Value of Objective Function = 0.9103815918

These initial estimates turn out to be pretty good, in the sense that only three more iterations are needed to converge to the maximum likelihood estimates and the final function value 0.784 does not change much from the initial function value 0.910, as shown in Output 88.4.3.

Output 88.4.3 Optimization of the Uncorrelated Factor Model: Iteration Summary

Iter	Rest arts	Func Calls	Act Con	Objective Function	Obj Fun Change	Max Abs Gradient Element	Lambda	Actual Over Pred Change
1	0	4	0	0.78792	0.1225	0.00175	0	0.932
2	0	6	0	0.78373	0.00419	0.000037	0	1.051
3	0	8	0	0.78373	5.087E−7	3.715E−9	0	1.001

Optimization Results

Iterations	3	Function Calls	11
Jacobian Calls	5	Active Constraints	0
Objective Function	0.783733415	Max Abs Gradient Element	3.7146571E−9
Lambda	0	Actual Over Pred Change	1.0006660673
Radius	0.0025042942		

Convergence criterion (ABSGCONV=0.00001) satisfied.

The fit summary is shown in Output 88.4.4.

Output 88.4.4 Fit of the Uncorrelated Factor Model

Fit Summary

Modeling Info	N Observations	64
	N Variables	9
	N Moments	45
	N Parameters	18
	N Active Constraints	0
	Independence Model Chi-Square	272.0467
	Independence Model Chi-Square DF	36
Absolute Index	Fit Function	0.7837
	Chi-Square	49.3752
	Chi-Square DF	27
	Pr > Chi-Square	0.0054
	Z-Test of Wilson & Hilferty	2.5474
	Hoelter Critical N	53
	Root Mean Square Residual (RMSR)	19.5739
	Standardized RMSR (SRMSR)	0.2098
	Goodness of Fit Index (GFI)	0.8555
Parsimony Index	Adjusted GFI (AGFI)	0.7592
	Parsimonious GFI	0.6416
	RMSEA Estimate	0.1147
	RMSEA Lower 90% Confidence Limit	0.0617
	RMSEA Upper 90% Confidence Limit	0.1646
	Probability of Close Fit	0.0271
	ECVI Estimate	1.4630
	ECVI Lower 90% Confidence Limit	1.2069
	ECVI Upper 90% Confidence Limit	1.8687
	Akaike Information Criterion	-4.6248
	Bozdogan CAIC	-89.9146
	Schwarz Bayesian Criterion	-62.9146
	McDonald Centrality	0.8396
Incremental Index	Bentler Comparative Fit Index	0.9052
	Bentler-Bonett NFI	0.8185
	Bentler-Bonett Non-normed Index	0.8736
	Bollen Normed Index Rho1	0.7580
	Bollen Non-normed Index Delta2	0.9087
	James et al. Parsimonious NFI	0.6139

Using the chi-square model test criterion, the uncorrelated factor model should be rejected at $\alpha = 0.05$. The RMSEA estimate is 0.1147, which is not indicative of a good fit according to Browne and Cudeck (1993). Other indices might suggest only a marginal good fit. For example, Bentler's comparative fit index and Bollen non-normed index delta2 are both above 0.90. However, many other do not attain this 0.90 level. For example, adjusted GFI is only 0.759. It is thus safe to conclude that there could be some improvements on the model fit.

The MODIFICATION option in the PROC TCALIS statement has been used to request for computing the LM test indices for model modifications. The results are shown in Output 88.4.5.

Output 88.4.5 Lagrange Multiplier Tests

Rank Order of the 10 Largest LM Stat for Factor Loadings

Variable	Factor	LM Stat	Pr > ChiSq	Parm Change
writing1	Read_Factor	9.76596	0.0018	2.95010
math3	Write_Factor	3.58077	0.0585	1.89703
math1	Read_Factor	2.15312	0.1423	1.17976
writing3	Math_Factor	1.87637	0.1707	1.41298
math3	Read_Factor	1.02954	0.3103	0.95427
reading2	Write_Factor	0.91230	0.3395	0.99933
writing2	Math_Factor	0.86221	0.3531	0.95672
reading1	Write_Factor	0.63403	0.4259	0.73916
math1	Write_Factor	0.55602	0.4559	0.63906
reading2	Math_Factor	0.55362	0.4568	0.74628

Rank Order of the 3 Largest LM Stat for Covariances of Factors

Var1	Var2	LM Stat	Pr > ChiSq	Parm Change
Write_Factor	Read_Factor	8.95268	0.0028	0.44165
Write_Factor	Math_Factor	7.07904	0.0078	0.40132
Math_Factor	Read_Factor	4.61896	0.0316	0.30411

Rank Order of the 10 Largest LM Stat for Error Variances and Covariances

Error of	Error of	LM Stat	Pr > ChiSq	Parm Change
writing1	math2	5.45986	0.0195	-13.16822
writing1	math1	5.05573	0.0245	12.32431
writing3	math3	3.93014	0.0474	13.59149
writing3	math1	2.83209	0.0924	-9.86342
writing2	reading1	2.56677	0.1091	10.15901
writing2	math2	1.94879	0.1627	8.40273
writing2	reading3	1.75181	0.1856	-7.82777
writing3	reading1	1.57978	0.2088	-7.97915
writing1	reading2	1.34894	0.2455	7.77158
writing2	math3	1.11704	0.2906	-7.23762

Three different tables for ranking the LM test results are shown. In the first table, the new loading parameters that would improve the model fit the most are shown first. For example, in the first row a new factor loading of writing1 on the Read_Factor is suggested to improve the model fit the most. The LM Stat value is 9.77. This is an approximation of the chi-square drop if this parameter was included in the model. The Pr > ChiSq value of 0.0018 indicates a significant improvement of model fit at $\alpha = 0.05$. Nine more new loading parameters are suggested in the table, with less and less statistical significance in the change of model fit chi-square. Note that these approximate chi-squares are one-at-a-time chi-square changes. That means that the overall chi-square drop is not a simple sum of individual chi-square changes when you include two or more new parameters in the modified model.

The other two tables in Output 88.4.5 shows the new parameters in factor covariances, error variances, or error covariances that would result in a better model fit. The table for the new parameters of the factor covariance matrix indicates that adding each of the covariances among factors might lead to a statistically significant improvement in model fit. This confirms with the initial argument that uncorrelated factors might not reasonable in this case—it fails to explain the covariances among observed variables through the correlations among latent factors. The largest LM Stat value in this table is 8.95, which is smaller than that of the largest LM Stat for the factor loading parameters. Despite this, it is more reasonable to add the covariance parameters among factors first to determine whether that improves the model fit. To do this, you need to add the COV statement for specifying the covariances among factors to the original code. The following statements are used to specify the modified factor model with covariances among factors:

```
proc tcalis data=cognitive1 nobs=64;
   factor
      Read_Factor   -> reading1-reading3  = load1-load3,
      Math_Factor   -> math1-math3        = load4-load6,
      Write_Factor  -> writing1-writing3  = load7-load9;
   pvar
      Read_Factor Math_Factor Write_Factor = 3 * 1.,
      reading1-reading3 math1-math3 writing1-writing3 = errvar1-errvar9;
   cov Read_Factor Math_Factor Write_Factor = fcov1-fcov3;
run;
```

The fit summary is shown in Output 88.4.6.

Output 88.4.6 Fit of the Correlated Factor Model

Fit Summary

Modeling Info	N Observations	64
	N Variables	9
	N Moments	45
	N Parameters	21
	N Active Constraints	0
	Independence Model Chi-Square	272.0467
	Independence Model Chi-Square DF	36
Absolute Index	Fit Function	0.4677
	Chi-Square	29.4667
	Chi-Square DF	24
	Pr > Chi-Square	0.2031
	Z-Test of Wilson & Hilferty	0.8320
	Hoelter Critical N	79
	Root Mean Square Residual (RMSR)	5.7038
	Standardized RMSR (SRMSR)	0.0607
	Goodness of Fit Index (GFI)	0.9109
Parsimony Index	Adjusted GFI (AGFI)	0.8330
	Parsimonious GFI	0.6073
	RMSEA Estimate	0.0601
	RMSEA Lower 90% Confidence Limit	.
	RMSEA Upper 90% Confidence Limit	0.1244
	Probability of Close Fit	0.3814
	ECVI Estimate	1.2602
	ECVI Lower 90% Confidence Limit	.
	ECVI Upper 90% Confidence Limit	1.5637
	Akaike Information Criterion	-18.5333
	Bozdogan CAIC	-94.3465
	Schwarz Bayesian Criterion	-70.3465
	McDonald Centrality	0.9582
Incremental Index	Bentler Comparative Fit Index	0.9768
	Bentler-Bonett NFI	0.8917
	Bentler-Bonett Non-normed Index	0.9653
	Bollen Normed Index Rho1	0.8375
	Bollen Non-normed Index Delta2	0.9780
	James et al. Parsimonious NFI	0.5945

The model fit chi-square value is 29.27, which is about 20 less than the model with uncorrelated factors. The p-value is 0.20, indicating a fairly satisfactory model fit. The RMSEA value is 0.06, which is close to 0.05, a value recommended as an indication of good model fit by Browne and Cudeck (1993). More fit indices that do not attain the 0.9 level with the uncorrelated factor model now have values close to or above 0.9. These include the goodness-of-fit index (GFI), McDonald centrality, Bentler-Bonnet NFI, and Bentler-Bonnet non-normed index. By all counts, the correlated factor model is a much better fit than the uncorrelated factor model.

In Output 88.4.7, the estimation results for factor loadings are shown. All these loadings are statistically significant, indicating non-chance relationships with the factors.

Output 88.4.7 Estimation of the Factor Loading Matrix

Factor Loading Matrix: Estimate/StdErr/t-value

	Read_Factor	Math_Factor	Write_Factor
reading1	6.7657 1.0459 6.4689 [load1]	0	0
reading2	7.8579 1.1890 6.6090 [load2]	0	0
reading3	9.1344 1.0712 8.5269 [load3]	0	0
math1	0	7.5488 1.0128 7.4536 [load4]	0
math2	0	8.4401 1.0838 7.7874 [load5]	0
math3	0	6.8194 1.0910 6.2506 [load6]	0
writing1	0	0	7.9677 1.1254 7.0797 [load7]
writing2	0	0	6.8742 1.1986 5.7350 [load8]
writing3	0	0	7.0949 1.2057 5.8844 [load9]

In Output 88.4.8, the factor covariance matrix is shown. Because the diagonal elements are all ones, the off-diagonal elements are correlations among factors. The correlations range from 0.30–0.5. These factors are moderately correlated.

Output 88.4.8 Estimation of the Correlations of Factors

Factor Covariance Matrix: Estimate/StdErr/t-value

	Read_Factor	Math_Factor	Write_Factor
Read_Factor	1.0000	0.3272 0.1311 2.4955 [fcov1]	0.4810 0.1208 3.9813 [fcov2]
Math_Factor	0.3272 0.1311 2.4955 [fcov1]	1.0000	0.3992 0.1313 3.0417 [fcov3]
Write_Factor	0.4810 0.1208 3.9813 [fcov2]	0.3992 0.1313 3.0417 [fcov3]	1.0000

In Output 88.4.9, the error variances for variables are shown.

Output 88.4.9 Estimation of the Error Variances

Error Variances

Variable	Parameter	Estimate	Standard Error	t Value
reading1	errvar1	37.24939	8.33997	4.46637
reading2	errvar2	46.49695	10.69869	4.34604
reading3	errvar3	15.90447	9.26097	1.71737
math1	errvar4	25.22889	7.72269	3.26685
math2	errvar5	24.89032	8.98327	2.77074
math3	errvar6	42.12110	9.20362	4.57658
writing1	errvar7	27.24965	10.36489	2.62903
writing2	errvar8	49.28881	11.39812	4.32429
writing3	errvar9	48.10684	11.48868	4.18733

All t values except the one for reading3 are bigger than 2, a value close to a critical t-value at $\alpha = 0.05$. This means that the error variance for reading3 could have been zero in the population, or it could have been nonzero but the current sample just has this nonsignificant value by chance (that is, a Type 2 error). Further research is needed to confirm either way.

In addition to the parameter estimation results, PROC TCALIS also outputs supplementary results that could be useful for interpretations. In Output 88.4.10, the squared multiple correlations and the factor scores regression coefficients are shown.

Output 88.4.10 Supplementary Estimation Results

Squared Multiple Correlations

Variable	Error Variance	Total Variance	R-Square
reading1	37.24939	83.02400	0.5513
reading2	46.49695	108.24300	0.5704
reading3	15.90447	99.34100	0.8399
math1	25.22889	82.21400	0.6931
math2	24.89032	96.12500	0.7411
math3	42.12110	88.62500	0.5247
writing1	27.24965	90.73400	0.6997
writing2	49.28881	96.54300	0.4895
writing3	48.10684	98.44500	0.5113

Factor Scores Regression Coefficients

	Read_Factor	Math_Factor	Write_Factor
reading1	0.02001	0.0006807	0.00198
reading2	0.01861	0.0006334	0.00185
reading3	0.06326	0.00215	0.00628
math1	0.00112	0.04035	0.00281
math2	0.00127	0.04572	0.00318
math3	0.0006068	0.02183	0.00152
writing1	0.00319	0.00274	0.05128
writing2	0.00152	0.00131	0.02446
writing3	0.00161	0.00138	0.02587

The percentages of variance for the observed variables that can be explained by the factors are shown in the R-Square column of the table for squared multiple correlations (R-squares). These R-squares can be interpreted meaningfully because there is no reciprocal relationships among variables or correlated errors in the model. All estimates of R-squares are bounded between 0 and 1.

In the table for factor scores regression coefficients, entries are coefficients for the variables you can use to create the factor scores. The larger the coefficient, the more influence of the corresponding variable for creating the factor scores. It makes intuitive sense to see the cluster pattern of these coefficients—the reading measures are more important to create the latent variable scores of Read_Factor and so on.

Example 88.5: Testing Equality of Two Covariance Matrices Using a Multiple-Group Analysis

You can use PROC TCALIS to do multiple-group or multiple-sample analysis. The groups in the analysis must be independent. In this example, a relatively simple multiple-group analysis is carried out. The covariance matrices of two independent groups are tested for equality. Hence, individual

covariance matrices are actually not structured. Rather, they are constrained to be the same under the null hypothesis. See Example 88.10 for a more sophisticated example of multiple-group analysis.

In this example, a reaction time experiment was conducted on two groups of individuals. One group ($N = 20$) was considered to be an expert group with prior training related to the tasks of the experiment. Another group ($N = 18$) was a control group without prior training. Three tasks of dexterity were administered to all individuals. These tasks differed by their required complexity levels of body skills. They were labeled as high, medium, and low complexities.

Apparently, the differential performance of the two groups under different task complexities was the primary research objective. In this example, however, you are interested in testing whether the groups have the same covariance matrix for the tasks. Equality of covariance matrices might be an essential assumption in some statistical tests for comparing group means. In this example, you will use PROC TCALIS to see the tenability of such an assumption. The covariance matrices for the two groups are stored in the data sets Expert and Novice, as shown in the following:

```
data expert(type=cov);
   input _type_ $ _name_ $ high medium low;
   datalines;
COV   high    5.88     .       .
COV   medium  2.88    7.16     .
COV   low     3.12    4.44    8.14
;

data novice(type=cov);
   input _type_ $ _name_ $ high medium low;
   datalines;
COV   high    6.42     .       .
COV   medium  1.24    8.25     .
COV   low     4.26    2.75    7.99
;
```

These data sets are read into the analysis through the GROUP statements in the following PROC TCALIS specification:

```
proc tcalis;
   group 1 / data=expert nobs=20 label="Expert";
   group 2 / data=novice nobs=18 label="Novice";
   model 1 / groups=1,2;
      mstruct
         var=high medium low;
         matrix _COV_ [1,1] = v_high v_medium v_low,
                      [2,]  = cov21,
                      [3,]  = cov31 cov32;
   fitindex NoIndexType On(only)=[chisq df probchi]
            chicorrect=eqcovmat;
   ods select ModelingInfo MSTRUCTVariables MSTRUCTCovInit Fit;
run;
```

The first GROUP statement defines group 1 for the expert group. The second GROUP statement defines group 2 for the novice group. The NOBS= option is used in both statements to provide the number of observations of these groups. The LABEL= option is used in these statements to provide meaningful group labels.

The MODEL statement defines MODEL 1. In the analysis, this model fits both groups 1 and 2, as indicated by the GROUPS= option of the statement. An MSTRUCT model for MODEL 1 is defined immediately afterward. Three variables, high, medium, and low, are specified in the VAR= option of the MSTRUCT statement. These three variables are also the row and column variables in the _COV_ matrix defined in the MATRIX statement.

In the first entry of the MATRIX statement, the diagonal elements of the _COV_ matrix are specified. Starting with the element [1,1] (and implicitly followed by [2,2] and [3,3]), variances for the three variables are parameters named v_high, v_medium, and v_low, respectively. In the second entry of the MATRIX statement, parameter specification starts with (and also ends with) the first element of the second row (that is, [2,1]) of the _COV_ matrix. The covariance between the variables high and medium is specified as cov21. In the last entry of the MATRIX statement, parameter specification starts with the first element of the third row (that is, [3,1] and then [3,2]) of the _COV_ matrix. The covariance between the variables high and low is specified as cov31 and the covariance between the variables medium and low is specified as cov32. No initial values are provided in this example. PROC TCALIS generates initial values by some reasonable methods.

There are six parameters in the covariance matrix _COV_, which also has six nonredundant elements in its general form as a symmetric matrix. Hence, model 1 is saturated. If this model were to fit a single group, it would have indicated a perfect model fit with zero chi-square value. Such a model is quite trivial and not of interest. But with two independent groups being fitted by this single model, it becomes a nontrivial model in which you test the equality of the two covariance matrices. Having this main purpose in mind, you use options in the FITINDEX statement to extract the relevant results that could answer your question. First, you use the NOINDEXTYPE option to suppress the printing of the index types in the fit summary table. Then, you use the ON(ONLY)= option to specify the fit indices printed in the fit summary table. In this example, only the model fit chi-square statistic, degrees of freedom, and the probability value of the test are requested. Finally, you use the CHICORRECT=EQCOVMAT option to request a chi-square correction for the test of equality of covariance matrices. This correction is due to Box (1949) and is implemented in PROC TCALIS as a built-in chi-square correction option.

In addition, because you are not interested in all displayed output, you use the ODS SELECT statement to display only those output (or ODS tables) of interest. In this example, only the modeling information, the variables involved, the initial covariance matrix specification, and the fit summary table are printed. All output in PROC TCALIS are named as an ODS table. To locate a particular output in PROC TCALIS, you must know the corresponding ODS table names. See the section "ODS Table Names" on page 6939 for a listing of ODS table names used by PROC TCALIS.

Output 88.5.1 displays some information regarding the basic model setup.

Output 88.5.1 Modeling Information and Initial Specification

Modeling Information

Group	Label	Data Set	N Obs	Model	Type
1	Expert	WORK.EXPERT	20	Model 1	MSTRUCT
2	Novice	WORK.NOVICE	18	Model 1	MSTRUCT

Output 88.5.1 *continued*

Model 1. Variables in the Model

high medium low

Number of Variables = 3

Model 1. Initial MSTRUCT _COV_ Matrix

	high	medium	low
high	. [v_high]	. [cov21]	. [cov31]
medium	. [cov21]	. [v_medium]	. [cov32]
low	. [cov31]	. [cov32]	. [v_low]

The modeling information table summarizes some basic information about the two groups. Both of them are fitted by model 1. The next table shows the variables involved: high, medium, and low. The order of variables in this table is the same as that of the row and column variables of the covariance model matrix, which is shown next. The parameters for the entries in the covariance matrix are shown. The names of parameters are displayed in parentheses. No initial estimates are given as input, as indicated by the missing value '.'.

The fit summary table is shown in Output 88.5.2, which has been much simplified for the current example due to the uses of some options in the FITINDEX statement.

Output 88.5.2 Model Fit

Fit Summary

Chi-Square	2.4924
Chi-Square DF	6
Pr > Chi-Square	0.8693

As shown in Table 88.5.2, the chi-square test statistic is 2.4924. With six degrees of freedom, the test statistic is not significant at $\alpha = 0.01$. Therefore, the fitted model is supported, which means that the equality of the covariance matrices of the groups is tenable.

Example 88.6: Illustrating Various General Modeling Languages

In PROC TCALIS, you can use many different modeling languages to specify the same model. The choice of modeling language depends on personal preferences and the purposes of the analysis. See the section "Which Modeling Language?" on page 6714 for guidance. In this example, the data and the model in Example 88.1 are used to illustrate how a particular model can be specified by various general modeling languages.

RAM Model Specification

In Example 88.1, the PATH modeling language was used. The PATH modeling language is closely related to the RAM modeling language. The following RAM model specification shows similar syntax to that of the PATH model specification.

```
proc tcalis nobs=932 data=Wheaton;
   ram
      path Anomie67     <-  Alien67   1.0,
      path Powerless67  <-  Alien67   0.833,
      path Anomie71     <-  Alien71   1.0,
      path Powerless71  <-  Alien71   0.833,
      path Education    <-  SES       1.0,
      path SEI          <-  SES       lambda,
      path Alien67      <-  SES       gamma1,
      path Alien71      <-  SES       gamma2,
      path Alien71      <-  Alien67   beta,
      pvar Anomie67                   theta1,
      pvar Powerless67                theta2,
      pvar Anomie71                   theta1,
      pvar Powerless71                theta2,
      pvar Education                  theta3,
      pvar SEI                        theta4,
      pvar Alien67                    psi1,
      pvar Alien71                    psi2,
      pvar SES                        phi,
      pcov Anomie67       Anomie71    theta5,
      pcov Powerless67    Powerless71 theta5;
run;
```

In the RAM model, a list of parameter locations are specified in the RAM statement. In each entry of the list, the parameter type, the variables involved, and the parameter are specified in order. These entries are separated by commas.

The parameter type is listed first in each list entry. In this example, three parameter types are specified: PATH, PVAR, and PCOV. These types correspond to the PATH, PVAR, and PCOV statements in the PATH model specification. There are other parameter types you can use for the RAM model specification, but they are not shown in this example. See the RAM statement on page 6805 for more details.

The variables involved are specified after the parameter type in each list entry. Depending on the parameter type, the number of variables required is either one or two. For entries with the PATH

parameter type, two variables must be specified—one for the outcome and one for the predictor. In this example, the two variables involved in the PATH entries are separated by arrows "<–" that indicate the prediction directions. When the arrows are not specified, the first variable is assumed to be the outcome variable, and the second one is the predictor variable. For the PVAR entries, which stand for partial variance, only one variable is required after the parameter type specification. For the PCOV entries, which stand for partial covariance, two variables are required. The order of these two variables in the PCOV entries is not important, however.

The last specification in each list entry is the parameter. The parameter can either be a fixed value (a numerical value alone), a free parameter without an initial value (a name alone), or a free parameter with an initial value (a name followed by a parenthesized numerical value). In this example, you specify five fixed values for the path coefficients (or effects) in the first five entries. In the next four paths and all other PVAR and PCOV entries, you specify free parameters without initial values.

When comparing with the PATH model specification, the RAM specification is quite similar. Certainly, the model fit would be the same because both specifications are mathematically equivalent. Unlike the PATH model that uses different tables for displaying different types of estimates, all estimates of the RAM model are displayed in a single table as shown in Output 88.6.1.

Output 88.6.1 RAM Model Estimates

RAM Pattern and Estimates

Type	Var1		Var2	Parameter	Estimate	Standard Error	t Value
Path	Anomie67	<-	Alien67		1.00000		
	Powerless67	<-	Alien67		0.83300		
	Anomie71	<-	Alien71		1.00000		
	Powerless71	<-	Alien71		0.83300		
	Education	<-	SES		1.00000		
	SEI	<-	SES	lambda	5.36883	0.43371	12.37880
	Alien67	<-	SES	gamma1	-0.62994	0.05634	-11.18092
	Alien71	<-	SES	gamma2	-0.24086	0.05489	-4.38836
	Alien71	<-	Alien67	beta	0.59312	0.04678	12.67884
Partial Var	Anomie67			theta1	3.60796	0.20092	17.95717
	Powerless67			theta2	3.59488	0.16448	21.85563
	Anomie71			theta1	3.60796	0.20092	17.95717
	Powerless71			theta2	3.59488	0.16448	21.85563
	Education			theta3	2.99366	0.49861	6.00398
	SEI			theta4	259.57639	18.31151	14.17559
	Alien67			psi1	5.67046	0.42301	13.40500
	Alien71			psi2	4.51479	0.33532	13.46394
	SES			phi	6.61634	0.63914	10.35190
Partial Cov	Anomie67		Anomie71	theta5	0.90580	0.12167	7.44472
	Powerless67		Powerless71	theta5	0.90580	0.12167	7.44472

These RAM model estimates match the set of estimates using the PATH model specification.

LINEQS Model Specification

Another way to specify the model in Example 88.1 is to use the LINEQS modeling language, which is shown in the following:

```
proc tcalis nobs=932 data=Wheaton;
   lineqs
      Anomie67     = 1.0    f_Alien67 + E1,
      Powerless67  = 0.833  f_Alien67 + E2,
      Anomie71     = 1.0    f_Alien71 + E3,
      Powerless71  = 0.833  f_Alien71 + E4,
      Education    = 1.0    f_SES     + E5,
      SEI          = lambda f_SES     + E6,
      f_Alien67    = gamma1 f_SES     + D1,
      f_Alien71    = gamma2 f_SES     + beta f_Alien67 + D2;
   std
      E1           = theta1,
      E2           = theta2,
      E3           = theta1,
      E4           = theta2,
      E5           = theta3,
      E6           = theta4,
      D1           = psi1,
      D2           = psi2,
      f_SES        = phi;
   cov
      E1  E3       = theta5,
      E2  E4       = theta5;
run;
```

In the LINEQS model, all paths in the PATH model are now specified in equations. In each equation, you list an outcome variable on the left-hand-side of the equation and all its predictors (including the error or disturbance variable) on the right-hand-side of the equation. Because each outcome variable can only be specified in exactly one equation, the number of equations in the LINEQS model and the number of paths in the corresponding PATH model do not match necessarily. In this example, there are eight equations in the LINEQS statement, but there would be nine paths specified in the corresponding PATH model.

In addition, in the LINEQS model, you need to follow a convention of naming latent variables. Names for latent variables that are neither errors nor disturbances must start with either 'F' or 'f.' Names for errors must start with either 'E' or 'e,' while names for disturbances must start with either 'D' or 'd.' Recall that in the PATH model specification, no such convention is imposed. For example, f_Alien67, f_Alien71, and f_SES are latent factors in the LINEQS model. They are not error terms, and so they must start with the 'f' prefix. However, this prefix is not needed in the PATH model. Furthermore, there are no explicit error terms that need to be specified in the PATH model, let alone specific prefixes for the error terms.

The PVAR statement in the PATH model is now replaced with the STD statement in the LINEQS model, and the PCOV statement with the COV statement. Recall that the PVAR and PCOV statements in the PATH model are for the partial variance and partial covariance specifications. The partial variance or covariance concepts are used in the PATH or RAM model specification because error terms are not named explicitly. But for the LINEQS model, errors or disturbances are

named explicitly as **exogenous** variables so that the partial variance or covariance concepts are no longer necessary. In this example, the variances of the errors ("E"-variables) and disturbances ("D"-variables) specified in the STD statement of the LINEQS model correspond to the partial variances of the endogenous variables specified in the PVAR statement of the PATH model. Similarly, covariances of errors specified in the COV statement of the LINEQS model correspond to the partial covariances of endogenous variables specified in the PCOV statement of the PATH model. The estimation results of the LINEQS model are shown in Output 88.6.2. Again, they are essentially the same estimates obtained from the PATH model specified in Example 88.1.

Output 88.6.2 LINEQS Model Estimates

Linear Equations

```
Anomie67     =    1.0000 f_Alien67 +  1.0000 E1
Powerless67  =    0.8330 f_Alien67 +  1.0000 E2
Anomie71     =    1.0000 f_Alien71 +  1.0000 E3
Powerless71  =    0.8330 f_Alien71 +  1.0000 E4
Education    =    1.0000 f_SES     +  1.0000 E5
SEI          =    5.3688*f_SES     +  1.0000 E6
Std Err           0.4337 lambda
t Value          12.3788
f_Alien67    =   -0.6299*f_SES     +  1.0000 D1
Std Err           0.0563 gamma1
t Value         -11.1809
f_Alien71    =   -0.2409*f_SES     +  0.5931*f_Alien67 +  1.0000 D2
Std Err           0.0549 gamma2        0.0468 beta
t Value          -4.3884              12.6788
```

Estimates for Variances of Exogenous Variables

Variable Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	E1	theta1	3.60796	0.20092	17.95717
	E2	theta2	3.59488	0.16448	21.85563
	E3	theta1	3.60796	0.20092	17.95717
	E4	theta2	3.59488	0.16448	21.85563
	E5	theta3	2.99366	0.49861	6.00398
	E6	theta4	259.57639	18.31151	14.17559
Disturbance	D1	psi1	5.67046	0.42301	13.40500
	D2	psi2	4.51479	0.33532	13.46394
Latent	f_SES	phi	6.61634	0.63914	10.35190

Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
E1	E3	theta5	0.90580	0.12167	7.44472
E2	E4	theta5	0.90580	0.12167	7.44472

LISMOD Specification

You can also specify general structural models by using the LISMOD modeling language. See the section "The LISMOD Model and Submodels" on page 6847 for details.

Four types of variables must be recognized before you can specify a LISMOD model. The η-variables (eta-variables) are latent factors that are endogenous, or predicted by other latent factors. The ξ-variables (xi-variables) are exogenous latent variables that are not predicted by any other variables. The y-variables are manifest variables that are indicators of the η-variables, and the x-variables are manifest variables that are indicators of the ξ-variables. In this example, Alien67 and Alien71 are the η-variables, and SES is the ξ-variable in the model. Manifest indicators for Alien67 and Alien71 include Anomie67, Powerless67, Anomie71, and Powerless71, which are the y-variables. Manifest indicators for SES include Education and SEI, which are the x-variables.

After defining these four types of variables, the parameters of the model are defined as entries in the model matrices. The _LAMBDAY_, _LAMBDAX_, _GAMMA_, and _BETA_ are matrices for the path coefficients or effects. The _THETAY, _THETAX_, _PSI_, and _PHI_ are matrices for the variances and covariances.

The following is the LISMOD specification for the model in Example 88.1:

```
proc tcalis nobs=932 data=Wheaton;
   lismod
      yvar   = Anomie67 Powerless67 Anomie71 Powerless71,
      xvar   = Education SEI,
      etavar = Alien67  Alien71,
      xivar  = SES;
   matrix _LAMBDAY_
      [1,1]  = 1,
      [2,1]  = 0.833,
      [3,2]  = 1,
      [4,2]  = 0.833;
   matrix _LAMBDAX_
      [1,1]  = 1,
      [2,1]  = lambda;
   matrix _GAMMA_
      [1,1]  = gamma1,
      [2,1]  = gamma2;
   matrix _BETA_
      [2,1]  = beta;
   matrix _THETAY_
      [1,1]  = theta1-theta2 theta1-theta2,
      [3,1]  = theta5,
      [4,2]  = theta5;
   matrix _THETAX_
      [1,1]  = theta3-theta4;
   matrix _PSI_
      [1,1]  = psi1-psi2;
   matrix _PHI_
      [1,1]  = phi;
run;
```

In the LISMOD statement, you specify the four lists of variables in the model. The orders of the variables in these lists are not arbitrary. They are the orders applied to the row and column variables in the model matrices, of which the parameter locations are specified.

The estimated model is divided into three conceptual parts. The first part is the measurement model that relates the η-variables with the y-variables, as shown in Output 88.6.3:

Output 88.6.3 LISMOD Model Measurement Model for the Eta-Variables

LAMBDAY Matrix: Estimate/StdErr/t-value

	Alien67	Alien71
Anomie67	1.0000	0
Powerless67	0.8330	0
Anomie71	0	1.0000
Powerless71	0	0.8330

THETAY Matrix: Estimate/StdErr/t-value

	Anomie67	Powerless67	Anomie71	Powerless71
Anomie67	3.6080 0.2009 17.9572 [theta1]	0	0.9058 0.1217 7.4447 [theta5]	0
Powerless67	0	3.5949 0.1645 21.8556 [theta2]	0	0.9058 0.1217 7.4447 [theta5]
Anomie71	0.9058 0.1217 7.4447 [theta5]	0	3.6080 0.2009 17.9572 [theta1]	0
Powerless71	0	0.9058 0.1217 7.4447 [theta5]	0	3.5949 0.1645 21.8556 [theta2]

The _LAMBDAY_ matrix contains the coefficients or effects of the η-variables on the y-variables. All these estimates are fixed constants as specified. The _THETAY_ matrix contains the error

variances and covariances for the y-variables. Three free parameters are located in this matrix: theta1, theta2, and theta5.

The second part of the estimated model is the measurement model that relates the ξ-variable with the x-variables, as shown in Output 88.6.4:

Output 88.6.4 LISMOD Model Measurement Model for the Xi-Variables

```
_LAMBDAX_ Matrix: Estimate/StdErr/t-value

                              SES

        Education          1.0000




        SEI                5.3688
                           0.4337
                          12.3788
                         [lambda]

_THETAX_ Matrix: Estimate/StdErr/t-value

                 Education             SEI

Education          2.9937               0
                   0.4986
                   6.0040
                  [theta3]

SEI                      0        259.5764
                                   18.3115
                                   14.1756
                                  [theta4]
```

The _LAMBDAX_ matrix contains the coefficients or effects of the ξ-variable SES on the x-variables. The effect of SES on Education is fixed at one. The effect of SES on SEI is represented by the free parameter lambda, which is estimated at 5.3688. The _THETAX_ matrix contains the error variances and covariances for the x-variables. Two free parameters are located in this matrix: theta3 and theta4.

The last part of the estimated model is the structural model that relates the latent variables η and ξ, as shown in Output 88.6.5:

Output 88.6.5 LISMOD Structural Model for the Latent Variables

```
                    _BETA_ Matrix: Estimate/StdErr/t-value

                                  Alien67        Alien71

                    Alien67             0              0



                    Alien71        0.5931              0
                                   0.0468
                                  12.6788
                                   [beta]

                    _GAMMA_ Matrix: Estimate/StdErr/t-value

                                                    SES

                           Alien67                -0.6299
                                                   0.0563
                                                 -11.1809
                                                 [gamma1]

                           Alien71                -0.2409
                                                   0.0549
                                                  -4.3884
                                                 [gamma2]

                    _PSI_ Matrix: Estimate/StdErr/t-value

                                  Alien67        Alien71

                    Alien67        5.6705              0
                                   0.4230
                                  13.4050
                                   [psi1]

                    Alien71             0         4.5148
                                                  0.3353
                                                 13.4639
                                                  [psi2]

                     _PHI_ Matrix: Estimate/StdErr/t-value

                                                  SES

                              SES                6.6163
                                                 0.6391
                                                10.3519
                                                  [phi]
```

The _BETA_ matrix contains effects of η-variables on themselves. In our example, there is only one such effect. The effect of Alien67 on Alien71 is represented by the free parameter beta. The

GAMMA matrix contains effects of the ξ-variable, which is SES in this example, on the η-variables Alien67 on Alien71. These effects are represented by free parameters gamma1 and gamma2. The _PSI_ matrix contains the error variances and covariances in the structural model. In this example, psi1 and psi2 are two free parameters for the error variances. Finally, the _PHI_ matrix is the covariance matrix for the ξ-variables. In this example, there is only one ξ-variable so that this matrix contains only the estimated variance of SES. This variance is represented by the parameter phi.

The estimates obtained from fitting the LISMOD model are the same as those from fitting the equivalent PATH, RAM, or LINEQS model. To some researchers the LISMOD modeling language might be more familiar, while for others modeling languages such as PATH, RAM, or LINEQS are more convenient to use.

Example 88.7: Fitting a Latent Growth Curve Model

Latent factors in structural equation modeling are constructed to represent important unobserved hypothetical constructs. However, with some manipulations latent factors can also represent random effects in models. In this example, a simple latent growth curve model is considered. You use latent factors to represent the random intercepts and slopes in the latent growth curve model.

Sixteen individuals were invited to a training program that was designed to boost self-confidence. During the training, the individuals' confidence levels were measured at five time points: initially and four more times separated by equal intervals. The data are stored in the following SAS data set:

```
data growth;
   input y1 y2 y3 y4 y5;
   datalines;
17.6  21.4  25.6  32.1  37.7
13.2  14.3  18.9  20.3  25.4
11.6  13.5  17.4  22.1  39.6
10.7  11.1  13.2  18.2  21.4
18.7  23.7  28.6  31.5  34.0
18.3  19.2  20.5  23.2  25.9
 9.2  13.5  17.8  19.2  21.1
18.3  23.5  27.9  30.2  34.6
11.2  15.6  20.8  22.7  30.4
17.0  22.9  26.9  31.9  35.6
10.4  13.6  18.0  25.6  29.3
17.7  19.0  22.5  28.5  30.7
14.5  19.4  21.1  28.8  31.5
20.0  21.4  28.9  30.2  35.6
14.6  19.3  21.7  28.5  32.0
11.7  15.2  19.1  23.7  28.7
;
```

First, consider a simple linear regression model for the confidence levels at time t due to training. That is,

$$y_t = \alpha + \beta T_t + e_t$$

where y_t represents the confidence level at time t ($t = 1, 2, \ldots, 5$), α represents the intercept, β represents the slope or the effect of training, T_t represents the fixed time point at t ($T_1 = 0$ and $T_i = T_{i-1} + 1$), and e_t is the error term at time t.

This simple linear regression assumes that the effect of training (slope) and the intercept are constants for the individuals. However, individual differences are rules rather than exceptions. It is thus more reasonable to argue that an index i for individuals should be added to the intercept and slope in the model. As a result, the following individualized regression model is derived:

$$y_{it} = \alpha_i + \beta_i T_t + e_t$$

where $i = 1, 2, \ldots, 16$. In this model, individuals are assumed to have different intercepts and slopes (regression coefficients). Note that theoretically e_t could also be "individualized" as e_{ti} in the model. But this is not done because such a model would be unnecessarily complicated without gaining additional insights in return.

Unfortunately, this individualized model with individual intercepts and slopes cannot be estimated directly. If you treat each α_i and β_i as fixed parameters, you are going to have too many parameters for the model to be identified or estimable. A workable solution is to treat α and β in the original linear regression model as random variables instead. That is, the latent growth curve model of interest is as follows:

$$y_t = \alpha + \beta T_t + e_t$$

where (α, β) is bivariate normal with unknown means, variances, and covariance. Therefore, instead of having 16 intercepts and 16 slopes to estimate in the individualized regression model, the final latent growth curve model has to estimate only two means, two variances and one covariance in the bivariate distribution of (α, β).

To use PROC TCALIS to fit this latent growth curve model, the random intercept and effect are treated as if they were covarying latent factors. To make them stand out more as latent variables, the random intercept and slope are renamed as f_α and f_β in the following structural equation:

$$y_t = f_\alpha + T_t f_\beta + e_t$$

where f_α and f_β are bivariate-normal latent variables. This model assumes that the error distribution is time dependent (with the index t). A simpler version is to make this error term invariant over time, which is then represented by the following model with constrained error variances:

$$y_t = f_\alpha + T_t f_\beta + e$$

This constrained model is considered first. The LINEQS modeling language is used to specify this constrained model, as shown in the following statements.

```
proc tcalis method=ml data=growth nostand noparmname;
   lineqs
      y1 = f_alpha               + e1,
      y2 = f_alpha  +  1 f_beta + e2,
      y3 = f_alpha  +  2 f_beta + e3,
      y4 = f_alpha  +  3 f_beta + e4,
      y5 = f_alpha  +  4 f_beta + e5;
   std
      f_alpha   = parm__,
      f_beta    = parm__,
      e1-e5     = 5 * evar;
   cov
      f_alpha f_beta = parm__;
   mean
      f_alpha   = parm__,
      f_beta    = parm__;
   fitindex on(only)=[chisq df probchi];
run;
```

In the LINEQS model specification, f_alpha and f_beta are treated as latent factors representing the random intercept and random slope, respectively. The f_ prefix for latent factors is not arbitrary but is required as a convention in the LINEQS modeling language. See the sections "Naming Variables in the LINEQS Model" on page 6840 and "Naming Variables and Parameters" on page 6880 for details.

At $T_1 = 0$, y_1 represents the initial confidence measurement so that it is not subject to the random effect f_beta. The next four measurements y_2, y_3, y_4, and y_5 are measured at time points T_2, T_3, T_4, and T_5, respectively. These are fixed time points with constant values 1, 2, 3, and 4, respectively, in the equations of the LINEQS statement.

The means, variances and covariances of f_alpha and f_beta are parameters in the model. The variances of these two latent variables are specified in the STD statement, while their covariance is specified in the COV statement. The means of f_alpha and f_beta are specified in the MEAN statement. Notice that the parameter name parm__ is used for all these parameters. Parameter names ending with two underscores ('__') are used as generic names for generating new parameter names. Each time such a generic name is parsed, PROC TCALIS replaces the underscores with an unique integer. Therefore, in the various locations of parm__ in the LINEQS model, the actual parameter names registered with PROC TCALIS are parm1, parm2, ..., parm5. For example, the variance parameters of f_alpha and f_beta are named parm1 and parm2, respectively, and the mean parameter of f_beta is named parm5.

The error variances for e1–e5 are also specified in the STD statement. Using the shorthand notation 5 *, the parameter name evar is repeated five times for the five error variances. This implicitly constrains the error variances for e1–e5 to be equal.

Special printing options are used in this example. In the PROC TCALIS statement, the NOSTAND option is specified because standardized solution is not of interest. The reason is that $y1$-$y5$ were already measured on comparable scales, making standardization unnecessary for interpretations. Another printing option specified is the NOPARMNAME option in the PROC TCALIS statement.

This option suppresses the printing of parameter names in the output for estimation. This makes the output look more precise when you do not need direct references to the parameter names. Still another printing option used is the ON(ONLY)= option of the FITINDEX statement. This option trims down the display of fit indices to include only those listed in the option. See the FITINDEX statement on page 6760 for details.

The fit summary table is shown in Output 88.7.1.

Output 88.7.1 Random Intercepts and Effects with Constrained Error Variances: Model Fit

Fit Summary

Chi-Square	31.4310
Chi-Square DF	14
Pr > Chi-Square	0.0048

In Output 88.7.1, the chi-square value in the fit summary table is 31.431 ($df = 14$, $p < 0.01$), which is a statistically significant result that might indicate a poor model fit. Despite that, it is illustrative to continue to look at the main estimation results, which are shown in the following table.

Output 88.7.2 Estimation of Random Intercepts and Effects with Constrained Error Variances

Estimates for Variances of Exogenous Variables

Variable Type	Variable	Estimate	Standard Error	t Value
Latent	f_alpha	13.89140	5.81540	2.38873
	f_beta	0.80742	0.42198	1.91342
Error	e1	3.32185	0.70031	4.74342
	e2	3.32185	0.70031	4.74342
	e3	3.32185	0.70031	4.74342
	e4	3.32185	0.70031	4.74342
	e5	3.32185	0.70031	4.74342

Covariances Among Exogenous Variables

Var1	Var2	Estimate	Standard Error	t Value
f_alpha	f_beta	-0.35281	1.13815	-0.30998

Mean Parameters

Variable Type	Variable	Estimate	Standard Error	t Value
Latent	f_alpha	14.15875	1.02906	13.75890
	f_beta	4.04813	0.27563	14.68665

In Output 88.7.2, the estimated variance of the random intercept α, which is represented by the variance estimate of the latent factor f_alpha, is 13.891 ($t = 2.389$). In the next row of the same table, the variance estimate of the random effect β, which is represented by the variance estimate of the latent factor f_beta, is 0.807 ($t = 1.913$).

The covariance of the random intercept and the random effect is shown in the next table for "Covariances Among Exogenous Variables." A negative estimate of -0.353 is shown. This means that the initial self-confidence level and the boosting effect of training are negatively correlated. The higher the initial self-confidence level, the smaller the training effect.

In the last table for the "Mean Parameters," the estimated mean of the random intercept is 14.159, which is an estimate of the averaged initial self-confidence level. The estimated mean of random effect is 4.048, which is an estimate of the averaged training effect.

Given that the model does not fit that well, perhaps you should not take the interpretations of these estimates so seriously. Knowing that the distribution of the errors might have been time-dependent, you now try to improve the fit of the model by relaxing the constraint about common error variances. This can be done by using the following specification:

```
proc tcalis method=ml data=growth nostand noparmname;
   lineqs
      y1 = f_alpha              + e1,
      y2 = f_alpha  +  1 f_beta + e2,
      y3 = f_alpha  +  2 f_beta + e3,
      y4 = f_alpha  +  3 f_beta + e4,
      y5 = f_alpha  +  4 f_beta + e5;
   std
      f_alpha   = parm__,
      f_beta    = parm__,
      e1-e5     = 5 * parm__;
   cov
      f_alpha f_beta = parm__;
   mean
      f_alpha   = parm__,
      f_beta    = parm__;
   fitindex on(only)=[chisq df probchi];
run;
```

In this new specification, there is only one change in the STD statement from the previous specification. That is, you now use five different parameters for the five error variances, as highlighted in the following:

```
   e1-e5   = 5 * parm__;
```

This specification is equivalent to putting parm__ five times for the five error variance parameters, which is shown as follows:

```
   e1-e5   = parm__ parm__ parm__ parm__ parm__;
```

The model fit summary is shown in Output 88.7.3.

Output 88.7.3 Random Intercepts and Effects with Unconstrained Error Variances: Model Fit

Fit Summary

Chi-Square	11.6250
Chi-Square DF	10
Pr > Chi-Square	0.3109

As you can see from the fit summary table, the chi-square for the unconstrained model is 11.625 ($df = 10$, $p > .10$). This indicates an acceptable model fit. The chi-square difference test can also be conducted for testing the previous constrained model against this new model. The chi-square difference is $19.81 = 31.431 - 11.625$. With df=4, this chi-square difference value is statistically significant at α=0.01, indicating a significant improvement of model fit by using the unconstrained model.

Estimation results are shown in Output 88.7.4.

Output 88.7.4 Estimation of Random Intercepts and Effects with Unconstrained Error Variances

Estimates for Variances of Exogenous Variables

Variable Type	Variable	Estimate	Standard Error	t Value
Latent	f_alpha	14.70071	5.66943	2.59298
	f_beta	0.45059	0.29867	1.50867
Error	e1	2.81712	1.35332	2.08164
	e2	0.32213	0.46118	0.69848
	e3	1.94429	0.86824	2.23935
	e4	1.88569	1.21306	1.55448
	e5	14.65193	5.99354	2.44462

Covariances Among Exogenous Variables

Var1	Var2	Estimate	Standard Error	t Value
f_alpha	f_beta	0.35291	0.90366	0.39054

Mean Parameters

Variable Type	Variable	Estimate	Standard Error	t Value
Latent	f_alpha	14.03046	1.01534	13.81851
	f_beta	3.96793	0.22612	17.54781

The estimation results for the unconstrained model present a slightly different picture than the constrained model. While the estimates for the means and variances of the random intercept and the random training effect look similar in both models, estimates of the covariance between the random intercept and the random training effect are quite different in the two models. The covariance esti-

mate is negative (−0.353) in the constrained model, but it is positive (0.353) in the unconstrained model. However, because the covariance estimates are not statistically significant in both models ($t = -0.310$ and 0.391, respectively), you wonder whether the current data are showing strong evidence that supports one way or another. To get a clearer picture, perhaps more data should be collected and the model should be fit again to check for the significance of the covariance between the random intercept and slope. If the covariance estimate is still not significant, future models might have this covariance fixed at 0 in analysis.

Example 88.8: Higher-Order and Hierarchical Factor Models

In this example, confirmatory higher-order and hierarchical factor models are fitted by PROC TCALIS.

In higher-order factor models, factors are at different levels. The higher-order factors explain the relationships among factors at the next lower level, in the same way that the first-order factors explain the relationships among manifest variables. For example, in a two-level higher factor model you have nine manifest variables V1–V9 with three first-order factors F1–F3. The first-order factor pattern of the model might appear like the following:

	F1	F2	F3
V1	x		
V2	x		
V3	x		
V4		x	
V5		x	
V6		x	
V7			x
V8			x
V9			x

where each "x" marks a nonzero factor loading and all other unmarked entries are fixed zeros in the model. To explain the correlations among the first-order factors, a second-order factor F4 is hypothesized with the following second-order factor pattern:

	F4
F1	x
F2	x
F3	x

If substantiated by your theory, you might have higher-order factor models with more than two levels.

In hierarchical factor models, all factors are at the same (first-order) level but are different in their clusters of manifest variables related. Using the terminology of Yung, Thissen, and McLeod (1999), factors in hierarchical factor models are classified into "layers." The factors in the first layer partition the manifest variables into clusters so that each factor has a distinct cluster of related manifest

variables. This part of the factor pattern of the hierarchical factor model is similar to that of the first-order factor model for manifest variables. The next layer of factors in the hierarchical factor model again partitions the manifest variables into clusters. However, this time each cluster contains at least two clusters of manifest variables that are formed in the previous layer. For example, the following is a factor pattern of a confirmatory hierarchical factor model with two layers:

	First Layer			Second Layer
	F1	F2	F3	F4
V1	x			x
V2	x			x
V3	x			x
V4		x		x
V5		x		x
V6		x		x
V7			x	x
V8			x	x
V9			x	x

F1–F3 are first-layer factors and F4 is the only second-layer factor. This special kind of two-layer hierarchical pattern is also known as a bifactor solution. In a bifactor solution, there are two classes of factors—group factors and a general factor. For example, in the preceding hierarchical factor pattern F1–F3 are group factors for different abilities and F4 is a general factor such as "intelligence" (see, for example, Holzinger and Swineford 1937). See Mulaik and Quartetti (1997) for more examples and distinctions among various types of hierarchical factor models. Certainly, if substantiated by your theory, hierarchical factor models with more than two layers are possible.

In this example, you use PROC TCALIS to fit these two types of confirmatory factor models. First, you fit a second-order factor model to a real data set. Then you fit a bifactor model to the same data set. In the final section of this example, an informal account of the relationship between the higher-order and hierarchical factor models is attempted. Techniques for constraining parameters using PROC TCALIS are also shown. This final section might be too technical in the first reading. Interested readers are referred to articles by Mulaik and Quartetti (1997), Schmid and Leiman (1957), and Yung, Thissen, and McLeod (1999), for more details.

A Second-Order Factor Analysis Model

In this section, a second-order confirmatory factor analysis model is applied to a correlation matrix of Thurstone reported by McDonald (1985). The correlation matrix is read into a SAS data set in the following statements:

```
data Thurst(type=corr);
title "Example of THURSTONE resp. McDONALD (1985, p.57, p.105)";
   _type_ = 'corr'; input _name_ $ V1-V9;
   label V1='Sentences' V2='Vocabulary' V3='Sentence Completion'
         V4='First Letters' V5='Four-letter Words' V6='Suffices'
         V7='Letter series' V8='Pedigrees' V9='Letter Grouping';
   datalines;
V1  1.       .      .      .      .      .      .      .      .
V2   .828   1.      .      .      .      .      .      .      .
V3   .776   .779   1.      .      .      .      .      .      .
V4   .439   .493    .460  1.      .      .      .      .      .
V5   .432   .464    .425   .674  1.      .      .      .      .
V6   .447   .489    .443   .590   .541  1.      .      .      .
V7   .447   .432    .401   .381   .402   .288  1.      .      .
V8   .541   .537    .534   .350   .367   .320   .555  1.      .
V9   .380   .358    .359   .424   .446   .325   .598   .452  1.
;
```

Variables in this data set are measures of cognitive abilities. Three factors are assumed for these nine variable V1–V9. These three factors are the first-order factors in the analysis. A second-order factor is also assumed to explain the correlations among the three first-order factors.

The following statements define a second-order factor model by using the LINEQS modeling language.

```
proc tcalis corr data=Thurst method=max nobs=213 nose nostand;
   lineqs
      V1      = X11 Factor1                                  + E1,
      V2      = X21 Factor1                                  + E2,
      V3      = X31 Factor1                                  + E3,
      V4      =             X42 Factor2                      + E4,
      V5      =             X52 Factor2                      + E5,
      V6      =             X62 Factor2                      + E6,
      V7      =                         X73 Factor3          + E7,
      V8      =                         X83 Factor3          + E8,
      V9      =                         X93 Factor3          + E9,
      Factor1 =                                   L1g FactorG + E10,
      Factor2 =                                   L2g FactorG + E11,
      Factor3 =                                   L3g FactorG + E12;
   std
      FactorG   = 1. ,
      E1-E12    = U1-U9 W1-W3;
   bounds
      0. <= U1-U9;
   fitindex ON(ONLY)=[chisq df probchi];
   /* SAS Programming Statements: Dependent parameter definitions */
      W1  = 1. - L1g * L1g;
      W2  = 1. - L2g * L2g;
      W3  = 1. - L3g * L3g;
run;
```

In the first nine equations of the LINEQS statement, variables V1–V3 are manifest indicators of latent factor Factor1, variables V4–V6 are manifest indicators of latent factor Factor2, and variables

V7–V9 are manifest indicators of latent factor Factor3. In the last three equations of the LINEQS statement, the three first-order factors Factor1–Factor3 are predicted by a common source: FactorG. Hence, Factor1–Factor3 are correlated due to the common source FactorG in the model.

An error term is added to each equation in the LINEQS statement. These error terms E1–E12 are needed because the factors are not assumed to be perfect predictors of the corresponding outcome variables.

In the STD statement, you specify variance parameters for all independent or exogenous variables in the model: FactorG, and E1–E12. The variance of FactorG is fixed at one for identification. Variances for E1–E9 are given parameter names U1–U9, respectively. Variances for E10–E12 are given parameter names W1–W3, respectively. Note that for model identification purposes, W1–W3 are defined as dependent parameters in the SAS programming statements. That is,

$$W_i = 1. - L_{ig}^2 \quad (i = 1, 2, 3)$$

These dependent parameter definitions ensure that the variances for Factor1–Factor3 are fixed at ones for identification.

In the BOUNDS statement, you specify that variance parameters U1–U9 must be positive in the solution.

In addition to the statements for model specification, options are used to control the output. In the PROC TCALIS statement, the NOSE and NOSTAND options suppress the display of standard errors and standardized results. In the FITINDEX statement, the ON(ONLY)= option requests only the model fit chi-square and its associated degrees of freedom and p-value be shown in the fit summary table. Using options in PROC TCALIS to reduce the amount the of printout is a good practice. It makes your output more focused, as you output only what you need in a particular situation.

In Output 88.8.1, parameters and their initial values, gradients, and bounds are shown.

Output 88.8.1 Parameters in the Model

Optimization Start
Parameter Estimates

N	Parameter	Estimate	Gradient	Lower Bound	Upper Bound
1	X11	1.00000	0.13476	.	.
2	X21	1.01408	0.17327	.	.
3	X31	0.95518	0.12174	.	.
4	X42	1.00000	0.22548	.	.
5	X52	0.96603	0.21304	.	.
6	X62	0.88305	0.19782	.	.
7	X73	1.00000	0.21041	.	.
8	X83	1.03403	0.39324	.	.
9	X93	0.91752	0.19880	.	.
10	L1g	0.75060	−0.57492	.	.
11	L2g	0.64268	−0.50975	.	.
12	L3g	0.60919	−0.56538	.	.
13	U1	0.18879	0.14837	0	.
14	U2	0.16579	0.08989	0	.
15	U3	0.25988	−0.03231	0	.
16	U4	0.33068	0.20120	0	.
17	U5	0.37538	0.09124	0	.
18	U6	0.47808	−0.03595	0	.
19	U7	0.44813	0.20918	0	.
20	U8	0.40994	−0.12469	0	.
21	U9	0.53541	0.05959	0	.

Value of Objective Function = 0.5693888709

The Number of Dependent Parameters is 3

N	Parameter	Estimate
22	W1	0.43660
23	W2	0.58697
24	W3	0.62889

The first table contains all the independent parameters. There are twenty-one in total. Parameters W1–W3, which are defined in the SAS programming statements, are shown in the next table for dependent parameters. Their initial values are computed as functions of initial independent parameters.

Output 88.8.2 shows information about optimization—iteration history and the convergence status.

Output 88.8.2 Optimization

Parameter Estimates	21
Functions (Observations)	45
Lower Bounds	9
Upper Bounds	0

Output 88.8.2 *continued*

Optimization Start

Active Constraints	0	Objective Function	0.5693888709
Max Abs Gradient Element	0.5749163348	Radius	1.8533033852

Iter	Rest arts	Func Calls	Act Con	Objective Function	Obj Fun Change	Max Abs Gradient Element	Lambda	Actual Over Pred Change
1	0	5	0	0.38684	0.1825	0.5158	3.214	1.174
2	0	9	0	0.18706	0.1998	0.1003	0	1.181
3	0	11	0	0.18039	0.00667	0.0273	0	0.987
4	0	13	0	0.18020	0.000192	0.00581	0	0.881
5	0	15	0	0.18017	0.000023	0.00295	0	0.967
6	0	17	0	0.18017	3.08E-6	0.000686	0	1.083
7	0	19	0	0.18017	4.606E-7	0.000379	0	1.195
8	0	21	0	0.18017	7.365E-8	0.000096	0	1.283
9	0	23	0	0.18017	1.228E-8	0.000054	0	1.342
10	0	25	0	0.18017	2.098E-9	0.000018	0	1.377
11	0	27	0	0.18017	3.63E-10	8.561E-6	0	1.397

Optimization Results

Iterations	11	Function Calls	30
Jacobian Calls	13	Active Constraints	0
Objective Function	0.1801712146	Max Abs Gradient Element	8.5605681E-6
Lambda	0	Actual Over Pred Change	1.3969225014
Radius	0.0000572561		

Convergence criterion (GCONV=1E-8) satisfied.

First, there are 21 independent parameters in the optimization by using 45 "Functions (Observations)." The so-called functions refer to the moments in the model that are structured with parameters. Nine lower bounds, which are specified for the error variance parameters, are specified in the optimization. The next table for iteration history shows that the optimization stops in 11 iterations. The notes at the bottom of table show that the solution converges without problems.

In the fit summary table shown in Output 88.8.3, the chi-square model fit value is 38.196, with df=24, and p=0.033. This indicates a satisfactory model fit.

Output 88.8.3 Fit Summary

Fit Summary

Chi-Square	38.1963
Chi-Square DF	24
Pr > Chi-Square	0.0331

The fitted equations with final estimates are shown in Output 88.8.4. Interpretations of these loadings are not done here. The last table in this output shows various variance estimates. These estimates are classified by whether they are for the latent variables, error variables, or disturbance variables.

Output 88.8.4 Estimation Results

Linear Equations

```
V1      =    0.9047*Factor1 +  1.0000 E1
                    X11
V2      =    0.9138*Factor1 +  1.0000 E2
                    X21
V3      =    0.8561*Factor1 +  1.0000 E3
                    X31
V4      =    0.8358*Factor2 +  1.0000 E4
                    X42
V5      =    0.7972*Factor2 +  1.0000 E5
                    X52
V6      =    0.7026*Factor2 +  1.0000 E6
                    X62
V7      =    0.7808*Factor3 +  1.0000 E7
                    X73
V8      =    0.7202*Factor3 +  1.0000 E8
                    X83
V9      =    0.7035*Factor3 +  1.0000 E9
                    X93
Factor1 =    0.8221*FactorG +  1.0000 E10
                    L1g
Factor2 =    0.7818*FactorG +  1.0000 E11
                    L2g
Factor3 =    0.8150*FactorG +  1.0000 E12
                    L3g
```

Estimates for Variances of Exogenous Variables

Variable Type	Variable	Parameter	Estimate
Latent	FactorG		1.00000
Error	E1	U1	0.18150
	E2	U2	0.16493
	E3	U3	0.26713
	E4	U4	0.30150
	E5	U5	0.36450
	E6	U6	0.50642
	E7	U7	0.39033
	E8	U8	0.48137
	E9	U9	0.50510
Disturbance	E10	W1	0.32420
	E11	W2	0.38879
	E12	W3	0.33576

For illustration purposes, you might check whether the model constraints put on the variances of Factor1–Factor3 are honored (although this should have been taken care of in the optimization). For example, the variance of Factor1 should be:

$$1 = (\text{Loading on FactorG})^2 + \text{Variance of E10}$$

Extracting the estimates from the output, you have:

$$(0.8221)^2 + 0.32420 = 1.0000$$

So, this model constraint is verified.

A Bifactor Model

A bifactor model (or a hierarchical factor model with two layers) for the same data set is now considered. In this model, the same set of factors as in the preceding higher-order factor model are used. The most notable difference is that the second-order factor FactorG in the higher-order factor model is no longer a factor of the first-order factors Factor1–Factor3. Instead, FactorG, like Factor1–Factor3, now also serves as a factor of the observed variable V1–V9. Unlike Factor1–Factor3, FactorG is considered to be a *general* factor in the sense that *all* observed variables have direct functional relationships with it. In contrast, Factor1–Factor3 are *group* factors in the sense that each of them has a direct functional relationship with only one group of *observed* variables. Because of the coexistence of a general factor and group factors at the same factor level, such a hierarchical model is also called a bifactor model.

The bifactor model is specified in the following statements:

```
proc tcalis corr data=Thurst method=max nobs=213 nose nostand;
   lineqs
      V1 = X11 Factor1                        + X1g FactorG + E1,
      V2 = X21 Factor1                        + X2g FactorG + E2,
      V3 = X31 Factor1                        + X3g FactorG + E3,
      V4 =            X42 Factor2             + X4g FactorG + E4,
      V5 =            X52 Factor2             + X5g FactorG + E5,
      V6 =            X62 Factor2             + X6g FactorG + E6,
      V7 =                       X73 Factor3  + X7g FactorG + E7,
      V8 =                       X83 Factor3  + X8g FactorG + E8,
      V9 =                       X93 Factor3  + X9g FactorG + E9;
   std
      Factor1-Factor3 = 3 * 1.,
      FactorG         = 1. ,
      E1-E9           = U1-U9;
   bounds
      0. <= U1-U9;
   fitindex ON(ONLY)=[chisq df probchi];
run;
```

In the LINEQS statement, there are only nine equations for the manifest variables in the model. Unlike the second-order factor model fitted previously, Factor1–Factor3 are no longer functionally related to FactorG and therefore there are no equations with Factor1–Factor3 as outcome variables.

All factors in the bifactor model are uncorrelated. The factor variances are all fixed at 1 in the STD statement. The variance parameters for E1–E9 are named U1–U9, respectively. The BOUNDS statement, again, is specified so that only positive estimates are accepted for error variance estimates. Like the previous PROC TCALIS run, options are specified in the PROC TCALIS and the FITINDEX statements to reduce the amount of default output.

There are more parameters in this model than in the preceding higher-order factor model, as shown in Output 88.8.5, which shows the optimization information.

Output 88.8.5 Optimization

Parameter Estimates	27
Functions (Observations)	45
Lower Bounds	9
Upper Bounds	0

Optimization Start

Active Constraints	0	Objective Function	0.8380304146
Max Abs Gradient Element	2.4076251809	Radius	20.596787596

Iter	Rest arts	Func Calls	Act Con	Objective Function	Obj Fun Change	Max Abs Gradient Element	Lambda	Actual Over Pred Change
1	0	5	0	0.70566	0.1324	0.4851	0.00140	0.148
2	0	7	0	0.30090	0.4048	0.3269	0	1.292
3	0	9	0	0.17403	0.1269	0.2947	0	0.985
4	0	11	0	0.11759	0.0564	0.0677	0	1.190
5	0	13	0	0.11455	0.00304	0.0267	0	1.043
6	0	15	0	0.11426	0.000285	0.00242	0	1.153
7	0	17	0	0.11423	0.000027	0.00168	0	1.394
8	0	19	0	0.11423	5.552E-6	0.000478	0	1.413
9	0	21	0	0.11423	1.154E-6	0.000335	0	1.420
10	0	23	0	0.11423	2.405E-7	0.000105	0	1.427
11	0	25	0	0.11423	5.016E-8	0.000068	0	1.432
12	0	27	0	0.11423	1.047E-8	0.000023	0	1.436
13	0	29	0	0.11423	2.184E-9	0.000014	0	1.439
14	0	31	0	0.11423	4.56E-10	4.909E-6	0	1.442

Optimization Results

Iterations	14	Function Calls	34
Jacobian Calls	16	Active Constraints	0
Objective Function	0.1142278162	Max Abs Gradient Element	4.9090342E-6
Lambda	0	Actual Over Pred Change	1.4423534599
Radius	0.0002294218		

Convergence criterion (GCONV=1E-8) satisfied.

There are 27 parameters in the bifactor model: nine for the loadings on the group factors Factor1–Factor3, nine for the loadings on the general factor FactorG, and nine for the variances of errors E1–E9. The optimization converges in 14 iterations without problems.

A fit summary table is shown in Output 88.8.6

Output 88.8.6 Fit Summary

```
                    Fit Summary

          Chi-Square            24.2163
          Chi-Square DF              18
          Pr > Chi-Square        0.1481
```

The fit of this model is quite good. The chi-square value is 24.216, with df=18 and p=0.148. This is expected because the bifactor model has more parameters than the second-order factor model, which already has a good fit with fewer parameters.

Estimation results are shown in Output 88.8.7. Estimates are left uninterpreted because they are not the main interest of this example.

Output 88.8.7 Estimation Results

```
                    Linear Equations

V1 =   -0.4879*Factor1 +  0.7679*FactorG +  1.0000 E1
              X11               X1g
V2 =   -0.4523*Factor1 +  0.7909*FactorG +  1.0000 E2
              X21               X2g
V3 =   -0.4045*Factor1 +  0.7536*FactorG +  1.0000 E3
              X31               X3g
V4 =    0.6140*Factor2 +  0.6084*FactorG +  1.0000 E4
              X42               X4g
V5 =    0.5058*Factor2 +  0.5973*FactorG +  1.0000 E5
              X52               X5g
V6 =    0.3943*Factor2 +  0.5718*FactorG +  1.0000 E6
              X62               X6g
V7 =   -0.7273*Factor3 +  0.5669*FactorG +  1.0000 E7
              X73               X7g
V8 =   -0.2468*Factor3 +  0.6623*FactorG +  1.0000 E8
              X83               X8g
V9 =   -0.4091*Factor3 +  0.5300*FactorG +  1.0000 E9
              X93               X9g
```

Output 88.8.7 *continued*

Estimates for Variances of Exogenous Variables

Variable Type	Variable	Parameter	Estimate
Latent	Factor1		1.00000
	Factor2		1.00000
	Factor3		1.00000
	FactorG		1.00000
Error	E1	U1	0.17236
	E2	U2	0.16984
	E3	U3	0.26848
	E4	U4	0.25281
	E5	U5	0.38735
	E6	U6	0.51757
	E7	U7	0.14966
	E8	U8	0.50039
	E9	U9	0.55175

One might ask whether this bifactor (hierarchical) model provides a significantly better fit than the previous second-order model. Can one use a chi-square difference test for nested models to answer this question? The answer is yes.

Although it is not obvious that the previous second-order factor model is nested within the current bifactor model, a general nested relationship between the higher-order factor and the hierarchical factor model is formally proved by Yung, Thissen, and McLeod (1999). Therefore, a chi-square difference test can be conducted using the following DATA step:

```
data _null_;
   df0 = 24; chi0 = 38.1963;
   df1 = 18; chi1 = 24.2163;
   diff = chi0-chi1;
   p = 1.-probchi(chi0-chi1,df0-df1);
   put 'Chi-square difference = ' diff;
   put 'p-value = ' p;
run;
```

The results are shown in the following:

Output 88.8.8 Chi-square Difference Test

```
Chi-square difference = 13.98
p-value = 0.0298603746
```

The chi-square difference is 13.98, with df=6 and p=0.02986. If α-level is set at 0.05, the bifactor model indicates a significantly better fit. But if α-level is set at 0.01, statistically the two models fit equally well to the data.

In the next section, it is demonstrated that the second-order factor model is indeed nested within the bifactor model, and hence the chi-square test conducted in the previous section is justified. In

addition, through the demonstration of the nested relationship between the two classes of models, you can see how some parameter constraints in structural equation model can be set up in PROC TCALIS.

For some practical researchers, the technical details involved in the next section might not be of interest and therefore could be skipped.

A Constrained Bifactor Model and Its Equivalence to the Second-Order Factor Model

To demonstrate that the second-order factor model is indeed nested within the bifactor model, a constrained bifactor model is fitted in this section. This constrained bifactor model is essentially the same as the preceding bifactor model, but with additional constraints on the factor loadings. Hence, the constrained bifactor model is nested within the unconstrained bifactor model.

Furthermore, if it can be shown that the constrained bifactor model is equivalent to the previous second-order factor, then the second-order factor model must also be nested within the unconstrained bifactor model. As a result, it justifies the chi-square difference test conducted in the previous section.

The construction of such a constrained bifactor model is based on Yung, Thissen, and McLeod (1999). In the following statements, a constrained bifactor model is specified.

```
proc tcalis corr data=Thurst method=max nobs=213 nose nostand;
   lineqs
      V1 = X11 Factor1                          + X1g FactorG + E1,
      V2 = X21 Factor1                          + X2g FactorG + E2,
      V3 = X31 Factor1                          + X3g FactorG + E3,
      V4 =              X42 Factor2             + X4g FactorG + E4,
      V5 =              X52 Factor2             + X5g FactorG + E5,
      V6 =              X62 Factor2             + X6g FactorG + E6,
      V7 =                          X73 Factor3 + X7g FactorG + E7,
      V8 =                          X83 Factor3 + X8g FactorG + E8,
      V9 =                          X93 Factor3 + X9g FactorG + E9;
   std
      Factor1-Factor3 = 3 * 1.,
      FactorG         = 1. ,
      E1-E9           = U1-U9;
   bounds
      0. <= U1-U9;
   fitindex ON(ONLY)=[chisq df probchi];
   parameters p1 (.5) p2 (.5) p3 (.5);
   /* Proportionality constraints */
   X1g = p1 * X11;
   X2g = p1 * X21;
   X3g = p1 * X31;
   X4g = p2 * X42;
   X5g = p2 * X52;
   X6g = p2 * X62;
   X7g = p3 * X73;
   X8g = p3 * X83;
   X9g = p3 * X93;
run;
```

In this constrained model, a PARAMETERS statement and nine SAS programming statements are added to the previous bifactor model. In the PARAMETERS statement, three new independent parameters are added: p1, p2, and p3. These parameters represent the proportions that constrain the factor loadings of the observed variables on the group factors Factor1–Factor3 and the general factor FactorG. They are all free parameters and have initial values at 0.5. The next nine SAS programming statements represent the proportionality constraints imposed. For example, X1g–X3g are now dependent parameters expressed as functions of p1, X11, X21, and X31. Adding three new parameters (in the PARAMETERS statement) and redefining nine original parameters as dependent (in the SAS programming statements) is equivalent to adding six ($= 9 - 3$) constraints to the original bifactor model. Mathematically, the additional statements in specifying the constrained bifactor model realizes the following six constraints:

$$\frac{\text{X1g}}{\text{X11}} = \frac{\text{X2g}}{\text{X21}} = \frac{\text{X3g}}{\text{X31}}$$

$$\frac{\text{X4g}}{\text{X42}} = \frac{\text{X5g}}{\text{X52}} = \frac{\text{X6g}}{\text{X62}}$$

$$\frac{\text{X7g}}{\text{X73}} = \frac{\text{X8g}}{\text{X83}} = \frac{\text{X9g}}{\text{X93}}$$

Obviously, with these six constraints the current constrained bifactor model is nested within the unconstrained version. What remains to be shown is that this constrained bifactor model is indeed equivalent to the previous second-order factor model. If so, the second-order factor model is also nested within the unconstrained bifactor model. One evidence for the equivalence of the current constrained bifactor model and the second-order factor model is from the fit summary table shown in Output 88.8.10. But first, it is also useful to consider the optimization information of the constrained bifactor model, which is shown in Output 88.8.9.

Output 88.8.9 Optimization

Parameter Estimates			21
Functions (Observations)			45
Lower Bounds			9
Upper Bounds			0
Optimization Start			
Active Constraints	0	Objective Function	12.813623548
Max Abs Gradient Element	0.2163121033	Radius	1

Output 88.8.9 *continued*

Iter	Rest arts	Func Calls	Act Con	Objective Function	Obj Fun Change	Max Abs Gradient Element	Lambda	Actual Over Pred Change
1*	0	5	0	11.77997	1.0337	0.1754	6.390	1.107
2*	0	7	0	11.40277	0.3772	0.1812	0.827	1.032
3*	0	9	0	10.63174	0.7710	0.1945	0.362	1.096
4*	0	11	0	8.97007	1.6617	0.2262	0.152	1.275
5*	0	13	0	5.18770	3.7824	0.4950	0.0283	1.862
6*	0	15	0	3.90387	1.2838	0.6547	0.781	1.289
7*	0	17	0	2.17113	1.7327	0.8946	0.781	1.228
8*	0	19	0	0.43097	1.7402	0.3567	0.195	1.532
9	0	21	0	0.19151	0.2395	0.0822	0	1.252
10	0	23	0	0.18095	0.0106	0.0143	0	1.135
11	0	25	0	0.18024	0.000713	0.00632	0	1.190
12	0	27	0	0.18018	0.000059	0.00260	0	1.354
13	0	29	0	0.18017	8.179E-6	0.000970	0	1.385
14	0	31	0	0.18017	1.232E-6	0.000387	0	1.390
15	0	33	0	0.18017	1.882E-7	0.000149	0	1.391
16	0	35	0	0.18017	2.886E-8	0.000059	0	1.392
17	0	37	0	0.18017	4.431E-9	0.000023	0	1.392
18	0	39	0	0.18017	6.8E-10	9.004E-6	0	1.392

Optimization Results

Iterations	18	Function Calls	42
Jacobian Calls	20	Active Constraints	0
Objective Function	0.1801712146	Max Abs Gradient Element	9.004196E-6
Lambda	0	Actual Over Pred Change	1.3918856965
Radius	0.0001807581		

Convergence criterion (GCONV=1E-8) satisfied.

As shown Output 88.8.9, there are 21 independent parameters in the constrained bifactor model for the 45 "Functions (Observations)." These numbers match those of the second-order factor model exactly. The optimization shows some problems in initial iterations. The iteration numbers with asterisks indicate that the Hessian matrix is not positive definite in those iterations. But as long as the final converged iteration is not marked with this asterisk, the problems exhibited in early iterations do not raise any concern, as in the current case. Next, the fit summary is shown in Output 88.8.10.

Output 88.8.10 Model Fit

Fit Summary

Chi-Square	38.1963
Chi-Square DF	24
Pr > Chi-Square	0.0331

In Output 88.8.10, the chi-square value in the fit summary table is 38.196, with df=24, and p=0.033. Again, these numbers match those of the second-order factor model exactly. These matches (same model fit with the same number of parameters) are necessary (but not sufficient) to show that the constrained bifactor model is equivalent to the second-order model. Stronger evidence is now presented.

In Output 88.8.11, estimation results of the constrained bifactor model are shown.

Output 88.8.11 Estimation Results

Linear Equations

```
V1 =   -0.5151*Factor1 +  0.7437*FactorG +  1.0000 E1
              X11              X1g
V2 =   -0.5203*Factor1 +  0.7512*FactorG +  1.0000 E2
              X21              X2g
V3 =   -0.4874*Factor1 +  0.7038*FactorG +  1.0000 E3
              X31              X3g
V4 =    0.5211*Factor2 +  0.6534*FactorG +  1.0000 E4
              X42              X4g
V5 =    0.4971*Factor2 +  0.6232*FactorG +  1.0000 E5
              X52              X5g
V6 =    0.4381*Factor2 +  0.5493*FactorG +  1.0000 E6
              X62              X6g
V7 =    0.4524*Factor3 +  0.6364*FactorG +  1.0000 E7
              X73              X7g
V8 =    0.4173*Factor3 +  0.5869*FactorG +  1.0000 E8
              X83              X8g
V9 =    0.4076*Factor3 +  0.5734*FactorG +  1.0000 E9
              X93              X9g
```

Estimates for Variances of Exogenous Variables

Variable Type	Variable	Parameter	Estimate
Latent	Factor1		1.00000
	Factor2		1.00000
	Factor3		1.00000
	FactorG		1.00000
Error	E1	U1	0.18150
	E2	U2	0.16493
	E3	U3	0.26713
	E4	U4	0.30150
	E5	U5	0.36450
	E6	U6	0.50642
	E7	U7	0.39033
	E8	U8	0.48138
	E9	U9	0.50510

According to Yung, Thissen, and McLeod (1999), two models are equivalent if there is a one-to-one correspondence of the parameters in the models. This fact is illustrated for the constrained bifactor model and the second-order factor model.

First, the error variances for E1–E9 in the second-order factor model are transformed directly (using an identity map) to that of the bifactor models. The nine error variances in Output 88.8.4 for the second-order factor model match those of the constrained bifactor model exactly in Output 88.8.11. In addition, the variances of factors are fixed at one in both models.

The error variances and the factor loadings at both factor levels in Output 88.8.4 for the second-order factor model are now transformed to yield the loading estimates in the constrained bifactor model. Denote P_1 as the first-order factor loading matrix, P_2 as the second-order factor loading matrix, and U_1^2 be the matrix of variances for disturbances. That is, for the second-order factor model,

$$P_1 = \begin{pmatrix} 0.9047 & 0 & 0 \\ 0.9138 & 0 & 0 \\ 0.8561 & 0 & 0 \\ 0 & 0.8358 & 0 \\ 0 & 0.7972 & 0 \\ 0 & 0.7026 & 0 \\ 0 & 0 & 0.7808 \\ 0 & 0 & 0.7202 \\ 0 & 0 & 0.7035 \end{pmatrix}$$

$$P_2 = \begin{pmatrix} 0.8221 \\ 0.7818 \\ 0.8150 \end{pmatrix}$$

$$U_1^2 = \begin{pmatrix} 0.3242 & 0 & 0 \\ 0 & 0.3888 & 0 \\ 0 & 0 & 0.3358 \end{pmatrix}$$

According to Yung, Thissen, and McLeod (1999), the transformation to obtain the estimates in the equivalent constrained bifactor model is:

$$\begin{aligned} L_1 &= P_1 U_1 \\ L_2 &= P_1 P_2 \end{aligned}$$

where L_1 is the matrix of the first-layer factor loadings (that is, loadings on group factors Factor1–Factor3), and L_2 is the matrix of the second-layer factor loadings (that is, loadings on FactorG) in the constrained bifactor model. Carrying out the matrix calculations for L_1 and L_2 shows that:

$$L_1 = \begin{pmatrix} 0.5151 & 0 & 0 \\ 0.5203 & 0 & 0 \\ 0.4875 & 0 & 0 \\ 0 & 0.5212 & 0 \\ 0 & 0.4971 & 0 \\ 0 & 0.4381 & 0 \\ 0 & 0 & 0.4525 \\ 0 & 0 & 0.4173 \\ 0 & 0 & 0.4077 \end{pmatrix}$$

$$L_2 = \begin{pmatrix} 0.7438 \\ 0.7512 \\ 0.7038 \\ 0.6534 \\ 0.6232 \\ 0.5493 \\ 0.6364 \\ 0.5870 \\ 0.5734 \end{pmatrix}$$

With very minor numerical differences and ignorable sign changes, these transformation results match the estimated loadings observed in Output 88.8.11 for the constrained bifactor model. Therefore, the second-order factor model is shown to be equivalent to the constrained bifactor model, and hence is nested within the unconstrained bifactor model.

Example 88.9: Linear Relations among Factor Loadings

In this example, a confirmatory factor-analysis model with linear constraints on loadings is specified by the FACTOR modeling language of PROC TCALIS. SAS programming statements are used to set the constraints. In the context of the current example, the differences between fitting covariance structures and correlation structures will also be discussed.

The correlation matrix of six variables from Kinzer and Kinzer (N=326) is used by Guttman (1957) as an example that yields an approximate simplex. McDonald (1980) uses this data set as an example of factor analysis where he assumes that the loadings on the second factor are linear functions of the loadings on the first factor. Let **B** be the factor loading matrix containing the two factors and six variables so that:

$$\mathbf{B} = \begin{pmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \\ b_{51} & b_{52} \\ b_{61} & b_{62} \end{pmatrix}$$

and

$$b_{j2} = \alpha + \beta b_{j1}, \quad j = 1, \ldots, 6$$

The correlation structures are represented by:

$$\mathbf{P} = \mathbf{B}\mathbf{B}' + \Psi$$

where $\Psi = \text{diag}(\psi_{11}, \psi_{22}, \psi_{33}, \psi_{44}, \psi_{55}, \psi_{66})$ represents the diagonal matrix of unique variances for the variables.

With parameters α and β being unconstrained, McDonald (1980) has fitted an underidentified model with seven degrees of freedom. Browne (1982) imposes the following identification condition:

$$\beta = -1$$

In this example, Browne's identification condition is imposed. The following is the specification of the confirmatory factor model using the FACTOR modeling language.

```
data kinzer(type=corr);
title "Data Matrix of Kinzer & Kinzer, see GUTTMAN (1957)";
   _type_ = 'corr';
   input _name_ $ var1-var6;
   datalines;
var1  1.00   .     .     .     .     .
var2   .51  1.00   .     .     .     .
var3   .46   .51  1.00   .     .     .
var4   .46   .47   .54  1.00   .     .
var5   .40   .39   .49   .57  1.00   .
var6   .33   .39   .47   .45   .56  1.00
  ;

proc tcalis data=kinzer nobs=326 nose;
   factor
      factor1 -> var1-var6 = b11 b21 b31 b41 b51 b61 (6 *.6),
      factor2 -> var1-var6 = b12 b22 b32 b42 b52 b62;
   pvar
      factor1-factor2 = 2 * 1.,
      var1-var6       = psi1-psi6 (6 *.3);
   parameters alpha (1.);
   /* SAS Programming Statements to define dependent parameters */
   b12 = alpha - b11;
   b22 = alpha - b21;
   b32 = alpha - b31;
   b42 = alpha - b41;
   b52 = alpha - b51;
   b62 = alpha - b61;
   fitindex on(only)=[chisq df probchi];
run;
```

In the FACTOR statement, you specify two factors, named factor1 and factor2, for the variables. In this model, all manifest variables have nonzero loadings on the two factors. These loading parameters are specified after the equal signs and are named with the prefix 'b.' Initial estimates are given in the parentheses. For example, in the first entry of factor-variable relations, you specify loadings of var1–var6 on factor1. All these loadings are free parameters and are given the same initial value 6. Loadings of var1–var6 on factor2 are also defined as parameters in the same statement. However, they are dependent parameters because they are functionally dependent on other independent parameters, as shown in the SAS programming statements afterwards. Initial values for these dependent parameters are thus unnecessary.

In the PVAR statement, the factor variances are fixed at ones, while the error variances for the variables are free parameters named psi1–psi6. Again, initial estimates are provided in the parentheses that follow after the parameter names.

An additional parameter alpha is specified in the PARAMETERS statement with an initial value of 1. Then, six SAS programming statements are used to define the loadings on the second factor as functions of the loadings on the first factor. Lastly, the FITINDEX statement is used to trim the results in the fit summary table.

In the specification, there are twelve loadings in the FACTOR statement and six unique variances in the PVAR statement. Adding the parameter alpha in the list, there are 19 parameters in total. However, the loading parameters are not all independent of each other. As defined in the SAS programming statements, six loadings are dependent. This reduces the number of free parameters to 13. Hence the degrees of freedom for the model is $8 = 21 - 13$. Notice that the factor variances are fixed at 1 and covariances among factors are fixed zeros by default.

In Output 88.9.1, a concise fit summary table is shown. The chi-square test statistic of model fit is 10.337 with df=8 (p=0.242). This indicates a good model fit.

Output 88.9.1 Fit of the Correlation Structures

Fit Summary

Chi-Square	10.3374
Chi-Square DF	8
Pr > Chi-Square	0.2421

The estimated factor loading matrix is presented in Output 88.9.2, and the estimated error variances and the estimate for alpha are presented in Output 88.9.3.

Output 88.9.2 Loading Estimates

Factor Loading Matrix

	factor1	factor2
var1	0.3609	0.6174
	[b11]	[b12]
var2	0.3212	0.6571
	[b21]	[b22]
var3	0.4859	0.4923
	[b31]	[b32]
var4	0.5745	0.4038
	[b41]	[b42]
var5	0.7985	0.1797
	[b51]	[b52]
var6	0.6736	0.3046
	[b61]	[b62]

Output 88.9.3 Unique Variances and the Additional Parameter

Error Variances

Variable	Parameter	Estimate
var1	psi1	0.53036
var2	psi2	0.44986
var3	psi3	0.48756
var4	psi4	0.47278
var5	psi5	0.31125
var6	psi6	0.53815

Additional Parameters

Type	Parameter	Estimate
Independent	alpha	0.97825

All these estimates are essentially the same as reported in Browne (1982). Notice that there are no standard error estimates in the output, as requested by the NOSE option in the PROC TCALIS statement. Standard error estimates are not of interest in this example.

In fitting the preceding factor model, wrong covariance structures rather than the intended correlation structures have been specified. As pointed out by Browne (1982), fitting such covariance structures directly is not entirely appropriate for analyzing correlations. For example, when fitting the correlation structures, the diagonal elements of $\mathbf{P}$ must always be fixed ones. This fact has never been enforced in the preceding specification. A simple check of the estimates will illustrate the problem. In Output 88.9.2, the loading estimates of VAR1 on the two factors are 0.3609 and 0.6174, respectively. In Output 88.9.3, the error variance estimate for VAR1 is 0.53036. The fitted variance of VAR1 can therefore be computed by the following equation:

$$\text{fitted variance} = 0.3609^2 + 0.6174^2 + 0.53036 = 1.0418$$

This fitted value is quite a bit off from 1.00, as required for the standardized variance of VAR1.

Fortunately, even though the wrong covariance structure model has been analyzed, the preceding analysis is not completely useless. For the current confirmatory factor model, according to Browne (1982) the estimates obtained from fitting the wrong covariance structure model are still consistent (as if they were estimating the population parameters in the correlation structures). However, the chi-square test statistic as reported previously is not correct.

Note that using the CORR option in the PROC TCALIS statement will not solve the problem. By specifying the CORR option you merely request PROC TCALIS to use the correlation matrix directly as a covariance matrix in the objective function for model fitting. It still would not constrain the fitting of the diagonal elements to 1 during estimation.

In the next section, a solution to the stated problem is suggested. It is not claimed that this is the only solution or the best solution. Alternative treatments of the problem are possible.

Fitting the Correct Correlation Structures

This main idea of this solution is to embed the intended correlation structures (with correct constraints on the diagonal elements of the correlation matrix) into a covariance structure model so that the estimation methods of PROC TCALIS can be applied legitimately to the specially constructed covariance structures.

First, the issue of the fixed ones on the diagonal of the correlation structure model is addressed. That is, the diagonal elements of the correlation structures represented by $(\mathbf{BB}' + \Psi)$ must be fitted by ones. This can be accomplished by constraining the error variances as dependent parameters of the loadings, as shown in the following:

$$\Psi_{jj} = 1. - b_{j1}^2 - b_{j2}^2, \quad j = 1, \ldots, 6$$

Other constraints might also serve the purpose, but the ones just presented are the most convenient and intuitive.

Now, due to the fact that discrepancy functions used in PROC TCALIS are derived for covariance matrices rather than correlation matrices, PROC TCALIS is essentially set up for analyzing covariance structures (with or without mean structures), but not correlation structures. Hence, the statistical theory behind PROC TCALIS applies to covariance structure analysis, but it might not generalize to correlation structure analysis in all situations. Despite that, with some manipulations PROC TCALIS can fit the correct correlation structures to the current data without compromising the statistical theory. These manipulations are now discussed. Recall that the correlation structures are represented by:

$$\mathbf{P} = \mathbf{BB}' + \Psi$$

As before, in the $\mathbf{B}$ matrix, there are six linear constraints on the factor loadings. In addition, the diagonal elements of $(\mathbf{BB}' + \Psi)$ are constrained to ones, as done by defining the error variances as dependent parameters of the loadings in the preceding equation. To analyze the correlation structures by using PROC TCALIS, a covariance structure model with such correlation structures embedded is now specified. That is, the covariance structure to be fitted by PROC TCALIS is as follows:

$$\mathbf{\Sigma} = \mathbf{DPD}' = \mathbf{D}(\mathbf{BB}' + \Psi)\mathbf{D}'$$

where $\mathbf{D}$ is a 6 x 6 diagonal matrix containing the population standard deviations for the manifest variables. Theoretically, it is legitimate that you analyze this covariance structure model for studying the embedded correlation structures. In addition, it does not matter whether your input matrix is a correlation or covariance matrix, or any rescaled covariance matrix (by multiplying any variables by any positive constants). You would get correct results if you could somehow specify these covariance structures correctly in PROC TCALIS. However, there seems to be nowhere in PROC TCALIS that you can specify the diagonal matrix $\mathbf{D}$ for the population standard deviations. So what can one do with this formulation? The answer is to rewrite the covariance structure model in a form similar to the usual confirmatory factor model, as presented in the following.

Let $\mathbf{T} = \mathbf{DB}$ and $\mathbf{K} = \mathbf{D}\Psi\mathbf{D}'$. The covariance structure model of interest can now be rewritten as:

$$\mathbf{\Sigma} = \mathbf{TT}' + \mathbf{K}$$

This form of covariance structures implies a confirmatory factor model with factor loading matrix $\mathbf{T}$ and error covariance matrix $\mathbf{K}$. This confirmatory factor model can certainly be specified using

the FACTOR modeling language, in much the same way you specify a confirmatory factor model in the preceding section. However, because you are actually more interested in estimating the basic set of parameters in matrices **B** and Ψ of the embedded correlation structures, you would define the model parameters as functions of this basic set of parameters of interest. This can be accomplished by using the PARAMETERS and the SAS programming statements.

All in all, you can use the following statements to set up such a confirmatory factor model with the desired correlation structures embedded.

```
proc tcalis data=Kinzer nobs=326 nose;
   factor
      factor1 -> var1-var6 = t11 t21 t31 t41 t51 t61,
      factor2 -> var1-var6 = t12 t22 t32 t42 t52 t62;
   pvar
      factor1-factor2 = 2 * 1.,
      var1-var6       = k1-k6;
   parameters alpha (1.) d1-d6 (6 * 1.)
              b11 b21 b31 b41 b51 b61 (6 *.6),
              b12 b22 b32 b42 b52 b62
              psi1-psi6;
   /* SAS Programming Statements */
   /* 12 Constraints on Correlation structures */
   b12  = alpha - b11;
   b22  = alpha - b21;
   b32  = alpha - b31;
   b42  = alpha - b41;
   b52  = alpha - b51;
   b62  = alpha - b61;
   psi1 = 1. - b11 * b11 - b12 * b12;
   psi2 = 1. - b21 * b21 - b22 * b22;
   psi3 = 1. - b31 * b31 - b32 * b32;
   psi4 = 1. - b41 * b41 - b42 * b42;
   psi5 = 1. - b51 * b51 - b52 * b52;
   psi6 = 1. - b61 * b61 - b62 * b62;
   /* Defining Covariance Structure Parameters */
   t11  = d1 * b11;
   t21  = d2 * b21;
   t31  = d3 * b31;
   t41  = d4 * b41;
   t51  = d5 * b51;
   t61  = d6 * b61;
   t12  = d1 * b12;
   t22  = d2 * b22;
   t32  = d3 * b32;
   t42  = d4 * b42;
   t52  = d5 * b52;
   t62  = d6 * b62;
   k1   = d1 * d1 * psi1;
   k2   = d2 * d2 * psi2;
   k3   = d3 * d3 * psi3;
   k4   = d4 * d4 * psi4;
   k5   = d5 * d5 * psi5;
   k6   = d6 * d6 * psi6;
   fitindex on(only)=[chisq df probchi];
run;
```

First, you notice that specifications in the FACTOR and the PVAR statements are essentially unchanged from the previous specification, except that the parameters are named differently here to reflect different model matrices. In the current specification, the factor loading parameters in matrix **T** are named with prefix 't,' and the error variance parameters in matrix **K** are named with prefix 'k.' Specification of these parameters reflects the covariance structures. As you see in the last block of the SAS programming statements statements, all these parameters are functions of the correlation structure parameters in **B**, Ψ, and **D**.

Next, in the PARAMETERS statement, all correlation structure parameters are defined with initial values provided. These are the parameters of interest: alpha is used to define dependencies among loadings, d's are the population standard deviations, b's are the loading parameters, and psi's are the error variance parameters. There are 25 parameters specified in this statement, but not all of them are free or independent.

In the first block of SAS programming statements, parameter dependencies or constraints on the correlation structures are specified. The first six statements realize the required linear relations among the factor loadings:

$$b_{j2} = \alpha - b_{j1}, \quad j = 1, \ldots, 6$$

The next six statements constrain the error variances so as to ensure that an embedded correlation structure model is being fitted. That is, each error variance is dependent on the corresponding loadings, as prescribed by the following equation:

$$\Psi_{jj} = 1. - b_{j1}^2 - b_{j2}^2, \quad j = 1, \ldots, 6$$

These twelve constraints reduce the number of independent parameters to 13, as expected.

The next block of SAS programming statements are essentially for relating the correlation structure parameters to the covariance structures that are specified in the FACTOR and the PVAR statements. These SAS programming statements realize the required relations: $\mathbf{T} = \mathbf{DB}$ and $\mathbf{K} = \mathbf{D}\Psi\mathbf{D}'$, but in non-matrix forms:

$$t_{ji} = d_j b_{ji} \quad (j = 1, \ldots, 6; \quad i = 1, 2)$$

$$k_{jj} = d_j d_j \Psi_{jj} \quad (j = 1, \ldots, 6)$$

where d_j denotes the j-th diagonal element of **D**.

The fit summary is presented in Output 88.9.4. The chi-square test statistic is 14.63 with df=8 (p=0.067). This shows that the previous chi-square test based on fitting a wrong covariance structure model is indeed questionable.

Output 88.9.4 Model Fit of the Correlation Structures

Fit Summary

Chi-Square	14.6269
Chi-Square DF	8
Pr > Chi-Square	0.0668

Estimates of the loadings and error variances are presented in Output 88.9.5. These estimates are for the covariance structure model with loading matrix **T** and error covariance matrix **K**. They are rescaled versions of the correlation structure parameters and are not of primary interest themselves.

Output 88.9.5 Estimates of Loadings and Error Variances

Factor Loading Matrix

	factor1	factor2
var1	0.3448	0.6367
	[t11]	[t12]
var2	0.3200	0.6512
	[t21]	[t22]
var3	0.4873	0.4778
	[t31]	[t32]
var4	0.5703	0.3948
	[t41]	[t42]
var5	0.7741	0.1964
	[t51]	[t52]
var6	0.6778	0.3126
	[t61]	[t62]

Factor Covariance Matrix

	factor1	factor2
factor1	1.0000	0
factor2	0	1.0000

Error Variances

Variable	Parameter	Estimate
var1	k1	0.49119
var2	k2	0.46780
var3	k3	0.51597
var4	k4	0.50070
var5	k5	0.35505
var6	k6	0.47685

The parameter estimates of the embedded correlation structures are shown in Output 88.9.6 as "additional" parameters.

Output 88.9.6 Estimates of Correlation Structure Parameters

Additional Parameters

Type	Parameter	Estimate
Independent	alpha	0.97400
	d1	1.00771
	d2	0.99712
	d3	0.99078
	d4	0.99085
	d5	0.99640
	d6	1.01687
	b11	0.34217
	b21	0.32095
	b31	0.49179
	b41	0.57553
	b51	0.77686
	b61	0.66659
Dependent	b12	0.63183
	b22	0.65305
	b32	0.48222
	b42	0.39848
	b52	0.19714
	b62	0.30742
	psi1	0.48371
	psi2	0.47051
	psi3	0.52561
	psi4	0.50998
	psi5	0.35762
	psi6	0.46116

Except for the population standard deviation parameter d's, all other parameters estimated in the current model can be compared with those from the previous fitting of an incorrect covariance structure model. Although estimates in the current model do not differ very much from those in the previous specification, it is at least reassuring that they are obtained from fitting a correctly specified covariance structure model with the intended correlation structures embedded.

Example 88.10: A Multiple-Group Model for Purchasing Behavior

In this example, data were collected from customers who made purchases from a retail company during years 2002 and 2003. A two-group structural equation model is fitted to the data.

The variables are:

Spend02:	total purchase amount in 2002
Spend03:	total purchase amount in 2003
Courtesy:	rating of the courtesy of the customer service
Responsive:	rating of the responsiveness of the customer service
Helpful:	rating of the helpfulness of the customer service
Delivery:	rating of the timeliness of the delivery
Pricing:	rating of the product pricing
Availability:	rating of the product availability
Quality:	rating of the product quality

For the ratings scales, nine-point scales were used. Customers could respond 1 to 9 on these scales, with 1 representing "extremely unsatisfied" and 9 representing "extremely satisfied." Data were collected from two different regions, which will be labeled as Region 1 ($N = 378$) and Region 2 ($N = 423$), respectively. The ratings were collected at the end of year 2002 so that they represent customers' purchasing experience in year 2002.

The central questions of the study are:

- How does the overall customer service affect the current purchases and predict future purchases?
- How does the overall product quality affect the current purchases and predict future purchases?
- Do current purchases predict future purchases?
- Do the two regions have different structural models for predicting the purchases?

In stating these research questions, you use several constructs that might or might not correspond to objective measurements. Current and future purchases are certainly measurable directly by the spending of the customers. That is, because customer service and product satisfaction/qualities were surveyed between 2002 and 2003, Spend02 would represent current purchases and Spend03 would represent future purchases in the study. Both variables Spend02 and Spend03 are objective measurements without measurement errors. All you need to do is to extract the information from the transaction records. But how about hypothetical constructs like customer service quality and product quality? How would you measure them in the model?

In measuring these hypothetical constructs, one might ask customers' perception about the service or product quality directly in a single question. A simple survey with two questions about the customer service and product qualities could then be what you need. These questions are called indicators (or indicator variables) of the underlying constructs. However, using just one indicator (question) for each of these hypothetical constructs would be quite unreliable—that is, measurement errors might dominate in the data collection process. Therefore, multiple indicators are usually recommended for measuring such hypothetical constructs.

There are two main advantages of using multiple indicators for hypothetical constructs. The first one is conceptual and the other is statistical/mathematical.

First, hypothetical constructs might conceptually be multifaceted themselves. Measuring a hypothetical construct by a single indicator does not capture the full meaning of the construct. For example, the product quality might refer to the durability of the product, the outlook of the product, the pricing of the product, and the availability of product, among others. The customer service quality might refer to the politeness of the customer service, the timeliness of the delivery, and the responsiveness of customer services, among others. Therefore, multiple indicators for a single hypothetical construct might be necessary if you want to cover the multifaceted aspects of a given hypothetical construct.

Second, from a statistical point of view, the reliability would be higher if you combine correlated indicators for a construct than if you use a single indicator only. Therefore, combining correlated indicators would lead to more accurate and reliable results.

One way to combine correlated indicators is to use a simple sum of them to represent the underlying hypothetical construct. However, a better way is to use the structural equation modeling technique that represents each indicator (variable) as a function of the underlying hypothetical construct plus an error term. In structural equation modeling, hypothetical constructs are constructed as latent factors, which are unobserved systematic (that is, non-error) variables. Theoretically, latent factors are free from measurement errors, and so the estimation through the structural equation modeling technique is more accurate than if you just use simple sums of indicators to represent hypothetical constructs. Therefore, structural equation modeling approach is the method of the choice in the current analysis.

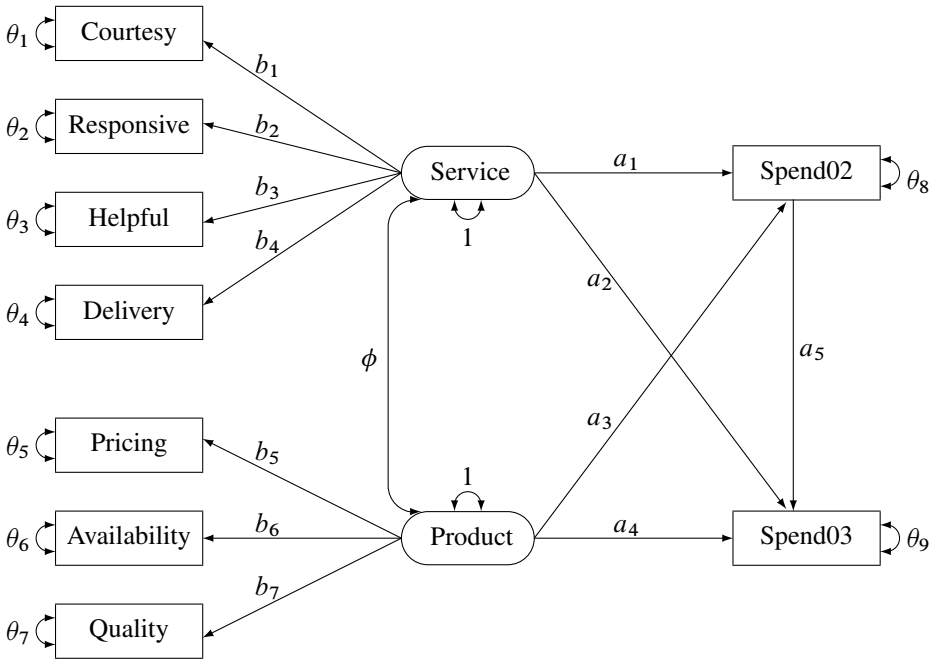
In practice, you must also make sure that there are enough indicators for the identification of the underlying latent factor, and hence the identification of the entire model. Necessary and sufficient rules for identification are very complicated to describe and are out of the scope of the current discussion (however, see Bollen 1989b for discussions of identification rules for various classes of structural equation models). Some simple rules of thumb may be useful as a guidance. For example, for unconstrained situations, you should at least have three indicators (variables) measured for a latent factor. Unfortunately, these rules of thumb do not guarantee identification in every case.

In this example, Service and Product are latent factors in the structural equation model that represents service and product qualities, respectively. In the model, these two latent factors are reflected by the ratings of the customers. Ratings on the Courtesy, Responsive, Helpful, and Delivery scales are indicators of Service. Ratings on the Pricing, Availability, and Quality scales are indicators of Product (that is, product quality).

A Path Diagram

A path diagram is used to represent the structural equation model for the purchase behavior. In Figure 88.5, observed or manifest variables are represented by rectangles, and latent variables are represented by ovals. As mentioned, two latent variables or factors, Service and Product are created as overall measures of customer service and product qualities, respectively.

Figure 88.5 Path Diagram of Purchasing Behavior



The left part of the diagram represents the measurement model of the latent variables. The Service factor has four indicators: Courtesy, Responsive, Helpful, and Delivery. The path coefficients to these observed variables from the Service factor are b_1, b_2, b_3, and b_4, respectively. Similarly, the Product variable has three indicators: Pricing, Availability, and Quality, with path coefficients b_5, b_6, and b_7, respectively.

The two latent factors are predictors of the purchase amounts Spend02 and Spend03. In addition, Spend02 also serves as a predictor of Spend03. Path coefficients or effects for this part of functional relationships are represented by a1–a5 in the diagram.

Each variable in the path diagram has a variance parameter. For endogenous or dependent variables, which serve as outcome variables in the model, the variance parameters are the error variances that are not accounted for by the predictors. For example, in the current model all observed variables are endogenous variables. The double-headed arrows tagged on these variables represent error variances. In the diagram, θ_1 to θ_9 are the names of these error variance parameters. For exogenous or independent variables, which never serve as outcome variables in the model, the variance parameters are the (total) variances of these variables. For example, in the diagram the double-headed arrows tagged on Service and Product represent the variances of these two latent variables. In the current model, both of these variances are fixed at one.

When the double-headed arrows are pointing to two variables, they represent covariances in the path diagram. For example, in Figure 88.5 the covariance between Service and Product is represented by the parameter ϕ.

A Macro for the Basic Path Model

For the moment, it is hypothesized that both Region 1 and Region 2 data are fitted by the same model as shown in Figure 88.5. Once the path diagram is drawn, it is readily translated into the PATH modeling language. See the PATH statement on page 6799 for details about how to use the PATH modeling language to specify structural equation models. A macro for the path model in Figure 88.5 using the PATH modeling language is shown in the following:

```
%macro BasepathModel;
path
   Service -> Spend02          a1,
   Service -> Spend03          a2,
   Product -> Spend02          a3,
   Product -> Spend03          a4,
   Spend02 -> Spend03          a5,
   Service -> Courtesy         b1,
   Service -> Responsive       b2,
   Service -> Helpful          b3,
   Service -> Delivery         b4,
   Product -> Pricing          b5,
   Product -> Availability     b6,
   Product -> Quality          b7;
pvar
   Courtesy Responsive Helpful
   Delivery Pricing
   Availability Quality = theta01-theta07,
   Spend02 = theta08,
   Spend03 = theta09,
   Service Product = 2 * 1.;
pcov
   Service Product = phi;
mean
   Service Product = 2 * 0.,
   Spend02         = InterSpend02,
   Spend03         = InterSpend03,
   Courtesy Responsive Helpful
   Delivery Pricing
   Availability Quality  = intercept01-intercept07;
%mend;
```

The name of this macro is BasePathModel. Defining a model by a SAS macro is not essential to run PROC TCALIS. You could have inserted the PATH model code directly into a PROC TCALIS run. The reason for using a macro here is to facilitate the presentation of the subsequent code so that the coding structure is made clearer. In the BasePathModel macro, the PATH statement captures all the path coefficient specification as well as the direction of the paths in the model. The first five paths are for defining how Spend02 and Spend03 are predicted from the latent variables Service, Product, and Spend02. The next seven paths are for defining the measurement model.

The PVAR statement captures the specification of the error variances and the variances of exogenous variables. The PCOV statement captures the specification of covariance between the two latent variables in the model.

The MEAN statement captures the specification of means and intercepts. Unlike some other representation schemes proposed by various researchers in the field, the mean parameters are not depicted in Figure 88.5. The reason is that representing the mean and intercept parameters in the path diagram would usually obscure the "causal" paths, which are of primary interest. In addition, it is a simple matter to specify the mean and intercept parameters in the MEAN statement without the help of a path diagram when you follow these principles:

- Each variable in the path diagram has a mean parameter that can be specified in the MEAN statement.
- For exogenous variables, the specifications in the MEAN statement are for the means of these variables.
- For endogenous variables, the specifications in the MEAN statement are for the intercepts of these variables.
- For variables that are not specified in the MEAN statement, their means or intercepts are fixed zeros by default.
- The total number of mean parameters should not exceed the number of observed variables.

Because all nine observed variables are endogenous in the model, their specification in the MEAN statement are for nine intercepts: intercept01–intercept07, InterSpend02, and InterSpend03. The means of the latent variables Service and Product are also specified in the MEAN statement as fixed zeros.

A Restrictive Model with Invariant Mean and Covariance Structures

The following shows the SAS data set being analyzed and the PROC TCALIS specification of a restrictive mean and covariance structure model:

```
data region1(type=cov);
   input _type_ $6. _name_ $12. Spend02 Spend03 Courtesy Responsive
         Helpful Delivery Pricing Availability Quality;
   datalines;
COV   Spend02      14.428  2.206  0.439 0.520 0.459 0.498 0.635 0.642 0.769
COV   Spend03       2.206 14.178  0.540 0.665 0.560 0.622 0.535 0.588 0.715
COV   Courtesy      0.439  0.540  1.642 0.541 0.473 0.506 0.109 0.120 0.126
COV   Responsive    0.520  0.665  0.541 2.977 0.582 0.629 0.119 0.253 0.184
COV   Helpful       0.459  0.560  0.473 0.582 2.801 0.546 0.113 0.121 0.139
COV   Delivery      0.498  0.622  0.506 0.629 0.546 3.830 0.120 0.132 0.145
COV   Pricing       0.635  0.535  0.109 0.119 0.113 0.120 2.152 0.491 0.538
COV   Availability  0.642  0.588  0.120 0.253 0.121 0.132 0.491 2.372 0.589
COV   Quality       0.769  0.715  0.126 0.184 0.139 0.145 0.538 0.589 2.753
MEAN     .        183.500 301.921 4.312 4.724 3.921 4.357 6.144 4.994 5.971
;

data region2(type=cov);
   input _type_ $6. _name_ $12. Spend02 Spend03 Courtesy Responsive
         Helpful Delivery Pricing Availability Quality;
   datalines;
COV   Spend02       14.489   2.193 0.442 0.541 0.469 0.508 0.637 0.675 0.769
COV   Spend03        2.193  14.168 0.542 0.663 0.574 0.623 0.607 0.642 0.732
COV   Courtesy       0.442   0.542 3.282 0.883 0.477 0.120 0.248 0.283 0.387
COV   Responsive     0.541   0.663 0.883 2.717 0.477 0.601 0.421 0.104 0.105
COV   Helpful        0.469   0.574 0.477 0.477 2.018 0.507 0.187 0.162 0.205
COV   Delivery       0.508   0.623 0.120 0.601 0.507 2.999 0.179 0.334 0.099
COV   Pricing        0.637   0.607 0.248 0.421 0.187 0.179 2.512 0.477 0.423
COV   Availability   0.675   0.642 0.283 0.104 0.162 0.334 0.477 2.085 0.675
COV   Quality        0.769   0.732 0.387 0.105 0.205 0.099 0.423 0.675 2.698
MEAN     .         156.250 313.670 2.412 2.727 5.224 6.376 7.147 3.233 5.119
;

proc tcalis maxiter=1000 omethod=nrr;
   group 1 / data=region1 label="Region 1" nobs=378;
   group 2 / data=region2 label="Region 2" nobs=423;
   model 1 / group=1,2;
      %BasepathModel
   run;
```

In the PROC TCALIS specification, you use the GROUP statements to specify the data for the two regions. Using the DATA= options in the GROUP statements, you assign the Region 1 data to group 1 and the Region 2 data to group 2. You label the two groups by the LABEL= options. Because the number of observations are not defined in the data sets, you use the NOBS= options in the GROUP statements to provide this information.

In the MODEL statement, you specify in the GROUP= option that both Groups 1 and 2 are fitted by the same model—model 1. Next, the BasePathModel macro is included for defining the model

for the two groups. Equivalently, you can use two MODEL statements to define two models for the two groups. This way you must also constrain each parameter in the two models. But using a single MODEL statement for the two groups is a short cut for defining such a restrictive model with total parameter invariance among groups.

The OMETHOD=NRR (Newton-Raphson ridge optimization) is used because the default Levenberg-Marquardt optimization did not converge in 1000 iterations.

In Output 88.10.1, a summary of modeling information is presented. Each group is listed with its associated data set, number of observations, and its corresponding model and the model type. In the current analysis, the same model is fitted to both groups. Next, a table for the types of variables is presented. As intended, all nine observed (manifest) variables are endogenous and all latent variables are exogenous in the model.

Output 88.10.1 Modeling Information and Variables

Modeling Information

Group	Label	Data Set	N Obs	Model	Type
1	Region 1	WORK.REGION1	378	Model 1	PATH
2	Region 2	WORK.REGION2	423	Model 1	PATH

Model 1. Variables in the Model

Endogenous	Manifest	Availability Courtesy Delivery Helpful Pricing Quality Responsive Spend02 Spend03
	Latent	
Exogenous	Manifest	
	Latent	Product Service

Number of Endogenous Variables = 9
Number of Exogenous Variables = 2

The optimization converges in 201 iterations. The fit summary table is presented in Output 88.10.2.

Output 88.10.2 Fit Summary

Fit Summary		
Modeling Info	N Observations	801
	N Variables	9
	N Moments	108
	N Parameters	31
	N Active Constraints	0
	Independence Model Chi-Square	399.7468
	Independence Model Chi-Square DF	72
Absolute Index	Fit Function	3.5310
	Chi-Square	2821.2798
	Chi-Square DF	77
	Pr > Chi-Square	0.0000
	Z-Test of Wilson & Hilferty	43.2651
	Hoelter Critical N	29
	Root Mean Square Residual (RMSR)	28.2211
	Standardized RMSR (SRMSR)	2.1370
	Goodness of Fit Index (GFI)	0.9996
Parsimony Index	Adjusted GFI (AGFI)	0.9995
	Parsimonious GFI	1.0690
	RMSEA Estimate	0.2987
	RMSEA Lower 90% Confidence Limit	0.2893
	RMSEA Upper 90% Confidence Limit	0.3082
	Probability of Close Fit	.
	Akaike Information Criterion	2667.2798
	Bozdogan CAIC	2229.4685
	Schwarz Bayesian Criterion	2306.4685
	McDonald Centrality	0.1803
Incremental Index	Bentler Comparative Fit Index	-7.3732
	Bentler-Bonett NFI	-6.0577
	Bentler-Bonett Non-normed Index	-6.8295
	Bollen Normed Index Rho1	-5.5994
	Bollen Non-normed Index Delta2	-7.5029
	James et al. Parsimonious NFI	-6.4783

The model chi-square statistic is 2818.732. With df=77 and $p < .0001$, the null hypothesis for the mean and covariance structures is rejected. All incremental fit indices are negative. These negative indices indicate a bad model fit, as compared with the independence model. The same fact can be deduced by comparing the chi-square values of the independence model and the structural model defined. The independence model has five degrees of freedom less (five parameters more) than the structural model but the chi-square value is only 399.747, much less than the model fit chi-square value of 2818.132. Because variables in social and behavioral sciences are almost always expected to correlate with each other, a structural model that explains relationships even worse than the independence model is deemed inappropriate for the data. The RMSEA for the structural model is .2985, which also indicates a bad model fit. However, GFI, AGFI, and Parsimonious GFI indicate good model fit, which is a little surprising given the fact that all other indices indicate the opposite and the overall model is pretty restrictive in the first place.

There is a warning in the output:

```
WARNING: Model 1. Although all predicted variances for the
         latent variables are positive, the corresponding
         predicted covariance matrix is not positive definite.
         It has one negative eigenvalue.
```

PROC TCALIS routinely checks the properties of the predicted covariance matrix. It will issue warnings when there are problems. In this case, the predicted covariance matrix for the latent variables is not positive definite and has one negative eigenvalue. The same fact can also be deduced from the output:

Output 88.10.3 Covariance of Latent Variables

Model 1. Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
Service	Product	phi	-0.78911	0.05542	-14.23750

In Output 88.10.3, the covariance between Service and Product is estimated at -1.115. Recall that the variances of these two variables are fixed at ones. Therefore, the predicted covariance matrix for the latent variables is:

$$\begin{pmatrix} 1.000 & -1.115 \\ -1.115 & 1.000 \end{pmatrix}$$

This matrix is certainly not a proper covariance matrix, as the covariance is larger in magnitude than the variances. The eigenvalues for this matrix are 2.115 and $-.115$, indicating a negative definite covariance matrix for the latent variables.

A Model with Unconstrained Parameters for the Two Regions

With all the bad model fit indications and the problematic predicted covariance matrix for the latent variables, you might conclude that an overly restricted model has been fit. Region 1 and Region 2 might not share exactly the same set of parameters. How about fitting a model at the other extreme with all parameters unconstrained for the two groups (regions)? Such a model can be easily specified, as shown in the following statements.

```
proc tcalis omethod=nrr;
   group 1 / data=region1 label="Region 1" nobs=378;
   group 2 / data=region2 label="Region 2" nobs=423;
   model 1 / groups=1;
      %BasepathModel
   model 2 / groups=2;
      refmodel 1/ AllNewParms;
   run;
```

Unlike the previous specification, in the current specification group 2 is now fitted by a new model labeled as model 2. This model is based on model 1, as specified in REFMODEL statement. The ALLNEWPARMS option in the REFMODEL statement request that all parameters specified in model 1 be renamed so that they become new parameters. As a result, this specification gives different sets of estimates for model 1 and model 2, although both models have the same path structures and a comparable set of parameters.

The optimization converges in 10 iterations without problems. The fit summary table is displayed in Output 88.10.4. The chi-square statistic is 29.613 (df= 46, $p = .97$). The theoretical model is not rejected. Many other measures of fit also indicate very good model fit. For example, GFI, AGFI, Bentler CFI, Bentler-Bonett NFI, Bollen non-normed index delta2 are all close to one, and RMSEA is close to zero.

Output 88.10.4 Fit Summary

Fit Summary

Modeling Info	N Observations	801
	N Variables	9
	N Moments	108
	N Parameters	62
	N Active Constraints	0
	Independence Model Chi-Square	399.7468
	Independence Model Chi-Square DF	72
Absolute Index	Fit Function	0.0371
	Chi-Square	29.6131
	Chi-Square DF	46
	Pr > Chi-Square	0.9710
	Z-Test of Wilson & Hilferty	-1.8950
	Hoelter Critical N	1697
	Root Mean Square Residual (RMSR)	0.0670
	Standardized RMSR (SRMSR)	0.0220
	Goodness of Fit Index (GFI)	1.0000
Parsimony Index	Adjusted GFI (AGFI)	1.0000
	Parsimonious GFI	0.6389
	RMSEA Estimate	0.0000
	RMSEA Lower 90% Confidence Limit	.
	RMSEA Upper 90% Confidence Limit	.
	Probability of Close Fit	1.0000
	Akaike Information Criterion	-62.3869
	Bozdogan CAIC	-323.9365
	Schwarz Bayesian Criterion	-277.9365
	McDonald Centrality	1.0103
Incremental Index	Bentler Comparative Fit Index	1.0000
	Bentler-Bonett NFI	0.9259
	Bentler-Bonett Non-normed Index	1.0783
	Bollen Normed Index Rho1	0.8840
	Bollen Non-normed Index Delta2	1.0463
	James et al. Parsimonious NFI	0.5916

Notice that because there are no constraints between the two models for the groups, you might have fit the two sets of data by the respective models separately and gotten exactly the same results as in the current analysis. For example, you get two model fit chi-square values from separate analyses. Adding up these two chi-squares will give you the same overall chi-square as in Output 88.10.4.

PROC TCALIS also provides a table for comparing relative model fit of the groups. In Output 88.10.5, basic modeling information and some measures of fit for the two groups are shown along with the corresponding overall measures.

Output 88.10.5 Fit Comparison among Groups

Fit Comparison Among Groups

		Overall	Region 1	Region 2
Modeling Info	N Observations	801	378	423
	N Variables	9	9	9
	N Moments	108	54	54
	N Parameters	62	31	31
	N Active Constraints	0	0	0
	Independence Model Chi-Square	399.7468	173.4482	226.2986
	Independence Model Chi-Square DF	72	36	36
Fit Index	Fit Function	0.0371	0.0023	0.0681
	Percent Contribution to Chi-Square	100	3	97
	Root Mean Square Residual (RMSR)	0.0670	0.0172	0.0907
	Standardized RMSR (SRMSR)	0.0220	0.0057	0.0298
	Goodness of Fit Index (GFI)	1.0000	1.0000	1.0000
	Bentler-Bonett NFI	0.9259	0.9950	0.8730

When you examine the results of this table, the first thing you have to realize is that in general the group statistics are not independent. For example, although the overall chi-square statistic can be written as the weighted sum of fit functions of the groups, in general it does not imply that the individual terms are statistically independent. In the current two-group analysis, the overall chi-square is written as:

$$T = (N_1 - 1) f_1 + (N_2 - 1) f_2$$

where N_1 and N_2 are sample sizes for the groups, f_1 and f_2 are the discrepancy functions for the groups. Even though T is chi-square distributed under the null hypothesis, in general the individual terms $(N_1 - 1) f_1$ and $(N_2 - 1) f_2$ are not chi-square distributed under the same null hypothesis. So when you compare the group fits by using the statistics in Output 88.10.5, you should treat those as descriptive measures only.

The current model is a special case where f_1 and f_2 are actually independent of each other. The reason is that there are no constrained parameters for the models fitted to the two groups. This would imply that the individual terms $(N_1 - 1) f_1$ and $(N_2 - 1) f_2$ are chi-squared distributed under the null hypothesis. Nonetheless, this fact is not important to our group comparison by using the descriptive statistics in Output 88.10.5. The values of f_1 and f_2 are shown in the row labeled "Fit Function." Group 1 (Region 1) is fitted better by its model ($f_1 = .0023$) than is group 2 (Region 2) by its model ($f_2 = .0681$). Next, the percentage contributions to the overall chi-square statistic for the two groups are shown. Group 1 contributes only 3% ($= (N_1 - 1) f_1 / T \times 100\%$) while Group 2 contributes 97%. Other measures like RMSR, SRMSR, and Bentler-Bonett NFI show that group 1 data are fitted better. The GFI's show equal fits for the two groups, however.

Despite a very good fit, the current model is not intended to be the final model. It was fitted mainly for illustration purposes. For multiple-group analysis, cross-group constraints are of primary

interest and should be explored whenever appropriate. The first fitting with all model parameters constrained for the groups has been shown to be too restrictive, while the current model with no cross-group constraints fits very well–so well that it might have overfit unnecessarily. A multiple-group model between these extremes is now explored. The following specification is for such an intermediate model with cross-group constraints.

```
proc tcalis omethod=nrr modification;
   group 1 / data=region1 label="Region 1" nobs=378;
   group 2 / data=region2 label="Region 2" nobs=423;
   model 1 / groups=1;
      %BasepathModel
   model 2 / groups=2;
      refmodel 1;
      mean
         Spend02 Spend03 = G2_InterSpend02 G2_InterSpend03,
         Courtesy Responsive Helpful
         Delivery Pricing Availability
         Quality = G2_intercept01-G2_intercept07;
   simtest
      SpendDiff       = (Spend02Diff Spend03Diff)
      MeasurementDiff = (CourtesyDiff ResponsiveDiff
                         HelpfulDiff DeliveryDiff
                         PricingDiff AvailabilityDiff
                         QualityDiff);
   Spend02Diff      = G2_InterSpend02 - InterSpend02;
   Spend03Diff      = G2_InterSpend03 - InterSpend03;
   CourtesyDiff     = G2_intercept01  - intercept01;
   ResponsiveDiff   = G2_intercept02  - intercept02;
   HelpfulDiff      = G2_intercept03  - intercept03;
   DeliveryDiff     = G2_intercept04  - intercept04;
   PricingDiff      = G2_intercept05  - intercept05;
   AvailabilityDiff = G2_intercept06  - intercept06;
   QualityDiff      = G2_intercept07  - intercept07;
   run;
```

In this specification, Region 1 is fitted by model 1, which is essentially unchanged from the previous analyses. Region 2 is fitted by model 2, which, again, is modified from model 1, as specified in the REFMODEL statement. The modification being made is the addition of unique mean parameters. In the MEAN statement, nine new parameters are specified. All these parameters are prefixed with G2_ to be distinguished from the parameters in model 1. Parameters G2_InterSpend02 and G2_InterSpend03 are intercepts for variables Spend02 and Spend03, respectively. Parameters G2_intercept01–G2_intercept07 are intercepts for the indicator measures of the latent variables. Because the latent variables in the model have means at fixed zeros, the intercept parameters G2_intercept01–G2_intercept07 and G2_InterSpend02 are also the means for the corresponding variables. To summarize, in this multiple-group model the covariance structures for the two regions are constrained to be the same, while the means structures are allowed to be unconstrained.

Additional statistics or tests are requested in the current PROC TCALIS run. The MODIFICATION option in the PROC TCALIS statement requests the Lagrange Multiplier tests and Wald tests be conducted. The Lagrange Multiplier tests provide information about which constrained or fixed parameters could be freed or added so as to improve the overall model fit. The Wald tests provide

information about which existing parameters could be fixed at zeros (or eliminated) without significantly affecting the overall model fit. These tests will be discussed with more details when the results are presented.

In the SIMTEST statement, two simultaneous tests are requested. The first simultaneous test is named SpendDiff, which includes two parametric functions Spend02Diff and Spend03Diff. The second simultaneous test is named MeasurementDiff, which includes seven parametric functions: CourtesyDiff, ResponsiveDiff, HelpfulDiff, DeliveryDiff, PricingDiff, AvailabilityDiff, and QualityDiff. The null hypothesis of these simultaneous tests is of the form:

$$H_0 : t_i = 0 \quad (i = 1 \ldots k)$$

where k is the number of parametric functions within the simultaneous test. In the current analysis, the component parametric functions are defined in the SAS programming statements, which are shown in the last block of the specification. Essentially, all these parametric functions represent the differences of the mean or intercept parameters between the two models for groups. The first simultaneous test is intended to test whether the mean or intercept parameters in the structural models are the same, while the second simultaneous test is intended to test whether the mean parameters in the measurement models are the same.

The fit summary table is shown in Output 88.10.6.

Output 88.10.6 Fit Summary

Fit Summary

Modeling Info	N Observations	801
	N Variables	9
	N Moments	108
	N Parameters	40
	N Active Constraints	0
	Independence Model Chi-Square	399.7468
	Independence Model Chi-Square DF	72
Absolute Index	Fit Function	0.1346
	Chi-Square	107.5461
	Chi-Square DF	68
	Pr > Chi-Square	0.0016
	Z-Test of Wilson & Hilferty	2.9452
	Hoelter Critical N	657
	Root Mean Square Residual (RMSR)	0.1577
	Standardized RMSR (SRMSR)	0.0678
	Goodness of Fit Index (GFI)	1.0000
Parsimony Index	Adjusted GFI (AGFI)	0.9999
	Parsimonious GFI	0.9444
	RMSEA Estimate	0.0382
	RMSEA Lower 90% Confidence Limit	0.0237
	RMSEA Upper 90% Confidence Limit	0.0514
	Probability of Close Fit	0.9275
	Akaike Information Criterion	-28.4539
	Bozdogan CAIC	-415.0924
	Schwarz Bayesian Criterion	-347.0924
	McDonald Centrality	0.9756
Incremental Index	Bentler Comparative Fit Index	0.8793
	Bentler-Bonett NFI	0.7310
	Bentler-Bonett Non-normed Index	0.8722
	Bollen Normed Index Rho1	0.7151
	Bollen Non-normed Index Delta2	0.8808
	James et al. Parsimonious NFI	0.6904

The chi-square value is 107.55 (df=68, p=0.0016), which is statistically significant. The null hypothesis of the mean and covariance structures is rejected if α-level at 0.01 or larger is chosen. However, in practical structural equation modeling, the chi-square test is not the only criterion, or even an important criterion, for evaluating model fit. The RMSEA estimate for the current model is .0382, which indicates a good fit. The probability level of close fit is .9969, indicating that a good population fit (RMSEA $<$ 0.05) hypothesis cannot be rejected. GFI, AGFI, and parsimonious GFI all indicate good fit. However, the incremental indices show only respectable model fit.

Comparison of the model fit to the groups is shown in Output 88.10.7.

Output 88.10.7 Fit Comparison among Groups

Fit Comparison Among Groups

		Overall	Region 1	Region 2
Modeling Info	N Observations	801	378	423
	N Variables	9	9	9
	N Moments	108	54	54
	N Parameters	40	31	31
	N Active Constraints	0	0	0
	Independence Model Chi-Square	399.7468	173.4482	226.2986
	Independence Model Chi-Square DF	72	36	36
Fit Index	Fit Function	0.1346	0.1261	0.1422
	Percent Contribution to Chi-Square	100	44	56
	Root Mean Square Residual (RMSR)	0.1577	0.1552	0.1599
	Standardized RMSR (SRMSR)	0.0678	0.0792	0.0557
	Goodness of Fit Index (GFI)	1.0000	1.0000	1.0000
	Bentler-Bonett NFI	0.7310	0.7260	0.7348

Looking at the percentage contribution to the chi-square, the Region 2 fitting shows a worse fit. However, this might be due to the larger sample size in Region 2. When comparing the fit of the two regions by using RMSR, which does not take the sample size into account, the fitting of two groups are about the same. The standardized RMSR even shows that Region 2 is fitted better. So, it seems to be safe to conclude that the models fit almost equally well (or badly) for the two regions.

Constrained parameter estimates for the two regions are shown in Output 88.10.8.

Output 88.10.8 Estimates of Path Coefficients and Other Covariance Parameters

Model 1. PATH List

Path			Parameter	Estimate	Standard Error	t Value
Service	->	Spend02	a1	0.37475	0.21318	1.75795
Service	->	Spend03	a2	0.53851	0.20840	2.58401
Product	->	Spend02	a3	0.80372	0.21939	3.66347
Product	->	Spend03	a4	0.59879	0.22144	2.70409
Spend02	->	Spend03	a5	0.08952	0.03694	2.42326
Service	->	Courtesy	b1	0.72418	0.07989	9.06482
Service	->	Responsive	b2	0.90452	0.08886	10.17972
Service	->	Helpful	b3	0.64969	0.07683	8.45574
Service	->	Delivery	b4	0.64473	0.09021	7.14677
Product	->	Pricing	b5	0.63452	0.07916	8.01600
Product	->	Availability	b6	0.76737	0.08265	9.28516
Product	->	Quality	b7	0.79716	0.08922	8.93470

Output 88.10.8 *continued*

Model 1. Variance Parameters

Variance Type	Variable	Parameter	Estimate	Standard Error	t Value
Error	Courtesy	theta01	1.98374	0.13169	15.06379
	Responsive	theta02	2.02152	0.16159	12.51005
	Helpful	theta03	1.96535	0.12263	16.02727
	Delivery	theta04	2.97542	0.17049	17.45184
	Pricing	theta05	1.93952	0.12326	15.73583
	Availability	theta06	1.63156	0.13067	12.48646
	Quality	theta07	2.08849	0.15329	13.62464
	Spend02	theta08	13.47066	0.71842	18.75051
	Spend03	theta09	13.02883	0.68682	18.96966
Exogenous	Service		1.00000		
	Product		1.00000		

Model 1. Covariances Among Exogenous Variables

Var1	Var2	Parameter	Estimate	Standard Error	t Value
Service	Product	phi	0.33725	0.07061	4.77599

All parameter estimates but one are statistically significant at $\alpha = 0.05$. The parameter a1, which represents the path coefficient from Service to Spend02, has a t-value of 1.76. This is only marginally significant. Although all these results bear the title of model 1, these estimates are the same for model 2, of which the corresponding results are not shown here.

The mean and intercept parameters for the two models (regions) are shown in Output 88.10.9.

Output 88.10.9 Estimates of Means and Intercepts

Model 1. Means and Intercepts

Type	Variable	Parameter	Estimate	Standard Error	t Value
Mean	Service		0		
	Product		0		
Intercept	Spend02	InterSpend02	183.50000	0.19585	936.95628
	Spend03	InterSpend03	285.49480	6.78127	42.10048
	Courtesy	intercept01	4.31200	0.08157	52.86519
	Responsive	intercept02	4.72400	0.08679	54.43096
	Helpful	intercept03	3.92100	0.07958	49.27201
	Delivery	intercept04	4.35700	0.09484	45.93968
	Pricing	intercept05	6.14400	0.07882	77.94992
	Availability	intercept06	4.99400	0.07674	65.07315
	Quality	intercept07	5.97100	0.08500	70.24543

Output 88.10.9 *continued*

Model 2. Means and Intercepts

Type	Variable	Parameter	Estimate	Standard Error	t Value
Mean	Service		0		
	Product		0		
Intercept	Spend02	G2_InterSpend02	156.25000	0.18511	844.09015
	Spend03	G2_InterSpend03	299.68311	5.77478	51.89515
	Courtesy	G2_intercept01	2.41200	0.07709	31.28628
	Responsive	G2_intercept02	2.72700	0.08203	33.24350
	Helpful	G2_intercept03	5.22400	0.07522	69.45319
	Delivery	G2_intercept04	6.37600	0.08964	71.12697
	Pricing	G2_intercept05	7.14700	0.07450	95.93427
	Availability	G2_intercept06	3.23300	0.07254	44.57020
	Quality	G2_intercept07	5.11900	0.08034	63.71500

All the mean and intercept estimates are statistically significant at $\alpha = 0.01$. Except for the fixed zero means for Service and Product, a quick glimpse of these mean and intercepts estimates shows a quite different pattern for the two models. Do these estimates truly differ beyond chance? The simultaneous tests of these parameter estimates shown in Output 88.10.10 can confirm this.

In Output 88.10.10, there are two simultaneous tests, as requested in the original statements.

Output 88.10.10 Simultaneous Tests

Simultaneous Tests

Simultaneous Test	Parametric Function	Function Value	DF	Chi-Square	p Value
SpendDiff			2	10458	<.0001
	Spend02Diff	-27.25000	1	10225	<.0001
	Spend03Diff	14.18831	1	185.86725	<.0001
MeasurementDiff			7	1610	<.0001
	CourtesyDiff	-1.90000	1	286.58605	<.0001
	ResponsiveDiff	-1.99700	1	279.63659	<.0001
	HelpfulDiff	1.30300	1	141.59942	<.0001
	DeliveryDiff	2.01900	1	239.35318	<.0001
	PricingDiff	1.00300	1	85.52567	<.0001
	AvailabilityDiff	-1.76100	1	278.09360	<.0001
	QualityDiff	-0.85200	1	53.06240	<.0001

The first one is SpendDiff, which tests simultaneously the hypotheses that:

$$H_0 : \text{G2\_InterSpend02 - InterSpend02} = 0$$
$$H_0 : \text{G2\_InterSpend03 - InterSpend03} = 0$$

The exceedingly large chi-square value 10548 suggests the composite null hypothesis is false. Individual tests for these hypotheses suggest each of these hypotheses should be rejected. The chi-square values for individual tests are 10225 and 185.97, respectively.

Similarly, the simultaneous and individual tests of the intercepts in the measurement model suggest that the two models (groups) differ significantly in the means of the measured variables. Region 2 has significantly higher means in variables Helpful, Delivery, and Pricing, but significantly lower means in variables Courtesy, Responsive, Availability, and Quality.

Now you are ready to answer the main research questions asked. The overall customer service (Service) does affect the future purchase (Spend03), but not the current purchase (Spend02), as the path coefficient a1 is only marginally significant. This perhaps is an artifact because the rating was done after the purchases in 2002. That is, purchases in 2002 had been done before the impression about customer service was fully formed. However, this argument cannot explain why overall customer service (Service) also shows a strong and significant relationship with purchases in 2002 (Spend02). Nonetheless, customer service and product quality do affect the future purchases (Spend03) in an expected way, even after partialling out the effect of the previous purchase amount (Spend02). Apart from the mean differences of the variables, the common measurement and prediction (or structural) models fit the two regions very well.

Because the current model fits well and most parts of fitting meet your expectations, you might accept this model without looking for further improvement. Nonetheless, for illustration purposes, it would be useful to consider the LM test results. In Output 88.10.11, ranked LM statistics for the path coefficients in model 1 and model 2 are shown.

Output 88.10.11 LM Tests for Path Coefficients

Model 1. Rank Order of the 10 Largest LM Stat for Path Relations

To	From	LM Stat	Pr > ChiSq	Parm Change
Service	Courtesy	11.15249	0.0008	−0.17145
Service	Helpful	3.09038	0.0788	0.09431
Service	Delivery	2.59511	0.1072	0.07504
Courtesy	Responsive	1.75943	0.1847	−0.07730
Delivery	Courtesy	1.66721	0.1966	0.08669
Helpful	Courtesy	1.62005	0.2031	0.07277
Courtesy	Product	1.48928	0.2223	−0.14815
Service	Product	0.83498	0.3608	−0.12327
Responsive	Helpful	0.76664	0.3813	−0.05625
Product	Helpful	0.53020	0.4665	−0.03831

Output 88.10.11 *continued*

Model 2. Rank Order of the 10 Largest LM Stat for Path Relations

To	From	LM Stat	Pr > ChiSq	Parm Change
Delivery	Courtesy	16.91167	<.0001	−0.26641
Service	Courtesy	9.11235	0.0025	0.15430
Courtesy	Delivery	8.12091	0.0044	−0.12989
Courtesy	Responsive	8.03954	0.0046	0.16215
Pricing	Responsive	5.48406	0.0192	0.10424
Courtesy	Product	4.39347	0.0361	0.24412
Courtesy	Quality	3.52147	0.0606	0.08672
Service	Delivery	3.20160	0.0736	−0.08281
Service	Helpful	2.97015	0.0848	−0.09198
Responsive	Pricing	2.91498	0.0878	0.08943

Path coefficients that lead to better improvement (larger chi-square drop) are shown first in the tables. For example, the first path coefficient suggested to be freed in model 1 is the Service <– Courtesy path. The associated p-value is 0.0008 and the estimated change of parameter value is −0.171. The second path coefficient is for the Service <– Helpful path, but it is not significant at the 0.05 level. So, is it good to add the Service <– Courtesy path to model 1, based on the LM test results? The answer is that it depends on your applications and the theoretical and practical implications. For instance, the Service –> Courtesy path, which is a part of the measurement model, is already specified in model 1. Even though the LM test statistic shows a significant drop of model fit chi-square, adding the Service <– Courtesy path might destroy the measurement model and lead to problematic interpretations. In this case, it is wise not to add the Service <– Courtesy path, which is suggested by the LM test results.

LM tests for the path coefficients in model 2 are shown in the bottom of Output 88.10.11. Quite a few of these tests suggest significant improvements in model fit. Again, you are cautioned against adding these paths blindly.

LM tests for the error variances and covariances are shown in Output 88.10.12.

Output 88.10.12 LM Tests for Error Covariances

Model 1. Rank Order of the 10 Largest LM Stat for Error Variances and Covariances

Error of	Error of	LM Stat	Pr > ChiSq	Parm Change
Responsive	Helpful	1.26589	0.2605	−0.15774
Delivery	Courtesy	0.70230	0.4020	0.12577
Helpful	Courtesy	0.50167	0.4788	0.09103
Quality	Availability	0.47993	0.4885	−0.09739
Quality	Pricing	0.45925	0.4980	0.09449
Responsive	Availability	0.25734	0.6120	0.05965
Helpful	Availability	0.24811	0.6184	−0.05413
Responsive	Pricing	0.23748	0.6260	−0.05911
Spend02	Availability	0.19634	0.6577	−0.13200
Responsive	Courtesy	0.18212	0.6696	0.06201

Model 2. Rank Order of the 10 Largest LM Stat for Error Variances and Covariances

Error of	Error of	LM Stat	Pr > ChiSq	Parm Change
Delivery	Courtesy	16.00996	<.0001	−0.57408
Responsive	Pricing	4.89190	0.0270	0.25403
Helpful	Delivery	3.33480	0.0678	0.25299
Delivery	Availability	2.79513	0.0946	0.20656
Responsive	Availability	2.16944	0.1408	−0.16421
Quality	Courtesy	2.14952	0.1426	0.17094
Responsive	Courtesy	2.12832	0.1446	0.20604
Quality	Pricing	2.00978	0.1563	−0.19154
Quality	Availability	1.99477	0.1578	0.19459
Responsive	Quality	1.88736	0.1695	−0.16963

Using $\alpha = 0.05$, you might consider adding two pairs of correlated errors in model 2. The first pair is for Delivery and Courtesy, which has a p-value less than 0.0001. The second pair is Pricing and Responsive, which has a p-value of 0.027. Again, adding correlated errors (in the PCOV statement) should not be a pure statistical consideration. You should also consider theoretical and practical implications.

LM tests for other subsets of parameters are also conducted. Some subsets do not have parameters that can be freed and so they are not shown here. Other subsets are simply not shown here for conserving space.

PROC TCALIS ranks and outputs the LM test results for some default subsets of parameters. You have seen the subsets for path coefficients and correlated errors in the two previous outputs. Some other LM test results are not shown. With this kind of default LM output, there could be a huge amount of modification indices to look at. Fortunately, you can limit the LM test results to any subsets of potential parameters that you might be interested in. With your substantive knowledge, you can define such meaningful subsets of potential parameters by using the LMTESTS statement. The LM test indices and rankings will then be done for each predefined subset of potential parameters.

With these customized LM results, you can limit your attention to consider only those meaningful parameters to be added. See the LMTESTS statement on page 6776 for details.

The next group of LM tests is for releasing implicit equality constraints in your model, as shown in Output 88.10.13.

Output 88.10.13 LM Tests for Equality Constraints

Lagrange Multiplier Statistics for Releasing Equality Constraints

	Released Parameter						Changes	
							Original	Released
Parm	Model	Type	Var1	Var2	LM Stat	Pr > ChiSq	Parm	Parm
a1	1	DV_IV	Spend02	Service	0.01554	0.9008	-0.0213	0.0238
	2	DV_IV	Spend02	Service	0.01554	0.9008	0.0238	-0.0213
a2	1	DV_IV	Spend03	Service	0.01763	0.8944	-0.0222	0.0248
	2	DV_IV	Spend03	Service	0.01763	0.8944	0.0248	-0.0222
a3	1	DV_IV	Spend02	Product	0.0003403	0.9853	-0.00321	0.00355
	2	DV_IV	Spend02	Product	0.0003403	0.9853	0.00355	-0.00321
a4	1	DV_IV	Spend03	Product	0.00176	0.9665	0.00714	-0.00802
	2	DV_IV	Spend03	Product	0.00176	0.9665	-0.00802	0.00714
a5	1	DV_DV	Spend03	Spend02	0.0009698	0.9752	-0.00100	0.00112
	2	DV_DV	Spend03	Spend02	0.0009698	0.9752	0.00112	-0.00100
b1	1	DV_IV	Courtesy	Service	19.17225	<.0001	0.2851	-0.3191
	2	DV_IV	Courtesy	Service	19.17225	<.0001	-0.3191	0.2851
b2	1	DV_IV	Responsive	Service	0.21266	0.6447	-0.0304	0.0341
	2	DV_IV	Responsive	Service	0.21266	0.6447	0.0341	-0.0304
b3	1	DV_IV	Helpful	Service	4.60629	0.0319	-0.1389	0.1555
	2	DV_IV	Helpful	Service	4.60629	0.0319	0.1555	-0.1389
b4	1	DV_IV	Delivery	Service	3.59763	0.0579	-0.1508	0.1687
	2	DV_IV	Delivery	Service	3.59763	0.0579	0.1687	-0.1508
b5	1	DV_IV	Pricing	Product	0.50974	0.4753	0.0468	-0.0524
	2	DV_IV	Pricing	Product	0.50974	0.4753	-0.0524	0.0468
b6	1	DV_IV	Availability	Product	0.57701	0.4475	-0.0457	0.0512
	2	DV_IV	Availability	Product	0.57701	0.4475	0.0512	-0.0457
b7	1	DV_IV	Quality	Product	0.00566	0.9400	-0.00511	0.00574
	2	DV_IV	Quality	Product	0.00566	0.9400	0.00574	-0.00511
theta01	1	COVERR	Courtesy	Courtesy	45.24725	<.0001	0.7204	-0.8064
	2	COVERR	Courtesy	Courtesy	45.24725	<.0001	-0.8064	0.7204
theta02	1	COVERR	Responsive	Responsive	1.73499	0.1878	-0.1555	0.1740
	2	COVERR	Responsive	Responsive	1.73499	0.1878	0.1740	-0.1555
theta03	1	COVERR	Helpful	Helpful	11.13266	0.0008	-0.3448	0.3860
	2	COVERR	Helpful	Helpful	11.13266	0.0008	0.3860	-0.3448
theta04	1	COVERR	Delivery	Delivery	4.99097	0.0255	-0.3364	0.3766
	2	COVERR	Delivery	Delivery	4.99097	0.0255	0.3766	-0.3364
theta05	1	COVERR	Pricing	Pricing	2.86428	0.0906	0.1729	-0.1936
	2	COVERR	Pricing	Pricing	2.86428	0.0906	-0.1936	0.1729
theta06	1	COVERR	Availability	Availability	2.53147	0.1116	-0.1494	0.1672
	2	COVERR	Availability	Availability	2.53147	0.1116	0.1672	-0.1494
theta07	1	COVERR	Quality	Quality	0.07328	0.7866	-0.0315	0.0352
	2	COVERR	Quality	Quality	0.07328	0.7866	0.0352	-0.0315
theta08	1	COVERR	Spend02	Spend02	0.00214	0.9631	0.0304	-0.0340
	2	COVERR	Spend02	Spend02	0.00214	0.9631	-0.0340	0.0304
theta09	1	COVERR	Spend03	Spend03	0.0001773	0.9894	-0.00842	0.00946
	2	COVERR	Spend03	Spend03	0.0001773	0.9894	0.00946	-0.00842
phi	1	COVEXOG	Service	Product	0.87147	0.3505	0.0605	-0.0678
	2	COVEXOG	Service	Product	0.87147	0.3505	-0.0678	0.0605

Recall that the measurement and the prediction models for the two regions are constrained to be the same by model referencing (that is, the REFMODEL statement). Output 88.10.13 shows you which parameter can be unconstrained so that your overall model fit might improve. For example, if you unconstrain the first parameter a1 for the two models, the expected chi-square drop (LM Stat) is about 0.0155, which is not significant ($p = .9008$). The associated parameter changes are small too. However, if you consider unconstraining parameter b1, the expected drop of chi-square is 19.17 ($p < 0.0001$). There are two rows for this parameter. Each row represents a parameter location to be released from the equality constraint. Consider the first row first. If you rename the coefficient for the Courtesy <– Service path in model 1 to a new parameter, say "new" (while keeping b1 as the parameter for the Courtesy <– Service path in model 2), and fit the model again, the new estimate of b1 will be 0.2851 larger than the previous b1 estimate. The estimate of "new" would be 0.3191 less than the previous b1 estimate. The second row for the b1 parameter shows similar but reflected results. It is for renaming the parameter location in model 2. For this example each equality constraint has exactly two locations, one for model 1 and one for model 2. That is the reason why you always observe reflected results for freeing the locations successively. Reflected results are not the case if you have equality constraints with more than two parameter locations.

Another example of big expected improvement of model fit is by freeing the constrained variances of Courtesy among the two models. The corresponding row to look at is the row with parameter theta01, where the parameter type is labeled "COVERR" and the values for the Var1 and Var2 are both "Courtesy." The LM statistic is 45.247, which is a significant chi-square drop if you free either parameter locations. If you choose to rename the error variance for Courtesy in model 1, the new theta01 estimate will be 0.8064 smaller than the original theta01 estimate. The new estimate of the error variance for Courtesy in model 2 will be 0.7204 larger than the previous theta01 estimate. Finally, the constrained parameter theta03, which is the error variance parameter for Helpful in both models, is also a potential constraint that can be released with a significant model fit improvement.

In addition to the LM statistics for suggesting ways to improve model fit, PROC TCALIS also computes the Wald tests to show which parameters can be constrained to zero without jeopardizing the model fit significantly. The Wald test results are shown in Output 88.10.14.

Output 88.10.14 Wald Tests

Stepwise Multivariate Wald Test

	------Cumulative Statistics-----			--Univariate Increment--	
Parm	Chi-Square	DF	Pr > ChiSq	Chi-Square	Pr > ChiSq
a1	3.09039	1	0.0788	3.09039	0.0788

In Output 88.10.14, you see that a1 is suggested to be a fixed zero parameter (or eliminated from the model) by the Wald test. Fixing this parameter to zero (or dropping the Spend02 <– Service path from the model) is expected to increase the model fit chi-square by 3.090 (p=.079), which is only marginally significant at $\alpha = .05$.

As is the case for the LM test statistics, you should not automatically adhere to the suggestions by the Wald statistics. Substantive and theoretical considerations should always be considered when determining whether a parameter should be added or dropped.

References

Akaike, H. (1974), "A New Look at the Statistical Identification Model," *IEEE Transactions on Automatic Control*, 19, 716–723.

Akaike, H. (1987), "Factor Analysis and AIC," *Psychometrika*, 52, 317–332.

Beale, E. M. L. (1972), "A Derivation of Conjugate Gradients," in *Numerical Methods for Nonlinear Optimization*, ed. F. A. Lootsma, London: Academic Press.

Bentler, P. M. (1985), *Theory and Implementation of EQS: A Structural Equations Program*, Manual for Program Version 2.0, Los Angeles: BMDP Statistical Software.

Bentler, P. M. (1986), *Lagrange Multiplier and Wald Tests for EQS and EQS/PC*, Los Angeles: BMDP Statistical Software.

Bentler, P. M. (1995), *EQS, Structural Equations Program Manual*, Program Version 5.0, Encino, CA: Multivariate Software.

Bentler, P. M. and Bonett, D. G. (1980), "Significance Tests and Goodness of Fit in the Analysis of Covariance Structures," *Psychological Bulletin*, 88, 588–606.

Bentler, P. M. and Freeman, E. H. (1983), "Test for Stability in Linear Structural Equation Systems," *Psychometrika*, 48, 143–145.

Bentler, P. M. and Weeks, D. G. (1980), "Linear Structural Equations with Latent Variables," *Psychometrika*, 45, 289–308.

Bishop, Y. M. M. , Fienberg, S. E., and Holland, P. W. (1977), *Discrete Multivariate Analysis: Theory and Practice*, Cambridge and London: MIT Press.

Bollen, K. A. (1986), "Sample Size and Bentler and Bonett's Nonnormed Fit Index," *Psychometrika*, 51, 375–377.

Bollen, K. A. (1989a), "A New Incremental Fit Index for General Structural Equation Models," *Sociological Methods and Research*, 17, 303–316.

Bollen, K. A. (1989b), *Structural Equations with Latent Variables*, New York: John Wiley & Sons.

Box, G. E. P. (1949), "A General Distribution Theory for a Class of Likelihood Criteria," *Biometrika*, 36, 317–346.

Bozdogan, H. (1987), "Model Selection and Akaike's Information Criterion (AIC): The General Theory and its Analytical Extensions," *Psychometrika*, 52, 345–370.

Browne, M. W. (1974), "Generalized Least Squares Estimators in the Analysis of Covariance Structures," *South African Statistical Journal*, 8, 1–24.

Browne, M. W. (1982), "Covariance Structures," in *Topics in Multivariate Analyses*, ed. D. M. Hawkins, New York: Cambridge University Press.

Browne, M. W. (1984), "Asymptotically Distribution-Free Methods for the Analysis of Covariance Structures," *British Journal of Mathematical and Statistical Psychology*, 37, 62–83.

Browne, M. W. and Cudeck, R. (1993), "Alternative Ways of Assessing Model Fit," in *Testing Structural Equation Models*, ed. K. A. Bollen and S. Long, Newbury Park, CA: Sage Publications.

Browne, M. W. and Du Toit, S. H. C. (1992), "Automated Fitting of Nonstandard Models," *Multivariate Behavioral Research*, 27, 269–300.

Browne, M. W. and Shapiro, A. (1986), "The Asymptotic Covariance Matrix of Sample Correlation Coefficients under General Conditions," *Linear Algebra and Its Applications*, 82, 169–176.

Bunch, J. R. and Kaufman, K. (1977), "Some Stable Methods for Calculating Inertia and Solving Symmetric Linear Systems," *Mathematics of Computation*, 31, 162–179.

Buse, A. (1982), "The Likelihood Ratio, Wald, and Lagrange Multiplier Tests: An Expository Note," *The American Statistician*, 36, 153–157.

Chamberlain, R. M., Powell, M. J. D., Lemarechal, C., and Pedersen, H. C. (1982), "The Watchdog Technique for Forcing Convergence in Algorithms for Constrained Optimization," *Mathematical Programming*, 16, 1–17.

Crawford, C. B. and Ferguson, G. A. (1970), "A General Rotation Criterion and Its Use in Orthogonal Rotation," *Psychometrika,* 35, 321–332.

DeLeeuw, J. (1983), "Models and Methods for the Analysis of Correlation Coefficients," *Journal of Econometrics*, 22, 113–137.

Dennis, J. E. and Mei, H. H. W. (1979), "Two New Unconstrained Optimization Algorithms Which Use Function and Gradient Values," *Journal of Optimization Theory and Applications*, 28, 453–482.

Dijkstra, T. K. (1992), "On Statistical Inference with Parameter Estimates on the Boundary of the Parameter Space," *British Journal of Mathematical and Statistical Psychology*, 45, 289–309.

Everitt, B. S. (1984), *An Introduction to Latent Variable Methods*, London: Chapman & Hall.

Fletcher, R. (1980), *Practical Methods of Optimization*, Vol. 1, Chichester: John Wiley & Sons.

Fletcher, R. (1987), *Practical Methods of Optimization*, Second Edition, Chichester: John Wiley & Sons.

Fuller, W. A. (1987), *Measurement Error Models*, New York: John Wiley & Sons.

Gay, D. M. (1983), "Subroutines for Unconstrained Minimization," *ACM Transactions on Mathematical Software*, 9, 503–524.

Gill, E. P., Murray, W., Saunders, M. A., and Wright, M. H. (1984), "Procedures for Optimization Problems with a Mixture of Bounds and General Linear Constraints," *ACM Transactions on Mathematical Software*, 10, 282–298.

Guttman, L. (1957), "Empirical Verification of the Radex Structure of Mental Abilities and Personality Traits," *Educational and Psychological Measurement*, 17, 391–407.

Hägglund, G. (1982), "Factor Analysis by Instrumental Variable Methods," *Psychometrika*, 47, 209–222.

Harman, H. H. (1976), *Modern Factor Analysis,* Third Edition, Chicago: University of Chicago Press.

Hoelter, J. W. (1983), "The Analysis of Covariance Structures: Goodness-of-Fit Indices," *Sociological Methods and Research*, 11, 325–344.

Holzinger, K. J. and Swineford, F. (1937), "The Bi-Factor Method," *Psychometrika*, 2, 41–54.

James, L. R., Mulaik, S. A., and Brett, J. M. (1982), *Causal Analysis: Assumptions, Models, and Data*, Beverly Hills, CA: Sage Publications.

Jennrich, R. I. (1973), "Standard Errors for Obliquely Rotated Factor Loadings," *Psychometrika,* 38, 593–604.

Jennrich, R. I. (1987), "Tableau Algorithms for Factor Analysis by Instrumental Variable Methods," *Psychometrika*, 52, 469–476.

Jöreskog, K. G. (1973), "A General Method for Estimating a Linear Structural Equation System," in *Structural Equation Models in the Social Sciences*, ed. A. S. Goldberger and O. D. Duncan, New York: Seminar Press.

Jöreskog, K. G. (1978), "Structural Analysis of Covariance and Correlation Matrices," *Psychometrika*, 43, 443–477.

Jöreskog, K. G. and Sörbom, D. (1985), *LISREL VI; Analysis of Linear Structural Relationships by Maximum Likelihood, Instrumental Variables, and Least Squares*, Uppsala: University of Uppsala.

Jöreskog, K. G. and Sörbom, D. (1988), *LISREL 7: A Guide to the Program and Applications*, Chicago: SPSS Inc.

Keesling, J. W. (1972), "Maximum Likelihood Approaches to Causal Analysis," Ph. D. dissertation, University of Chicago, 1972.

Kmenta, J. (1971), *Elements of Econometrics*, New York: Macmillan.

Lawley, D. N. and Maxwell, A. E. (1971), *Factor Analysis as a Statistical Method*, New York: American Elsevier.

Lee, S. Y. (1985), "On Testing Functional Constraints in Structural Equation Models," *Biometrika*, 72, 125–131.

Loehlin, J. C. (2004), *Latent Variable Models, An Introduction to Factor, Path, and Structural Analysis (4th ed.)*, Hillsdale, NJ: Lawrence Erlbaum Associates.

Long, J. S. (1983), *Covariance Structure Models, an Introduction to LISREL*, Beverly Hills, CA: Sage Publications.

MacCallum, R. (1986), "Specification Searches in Covariance Structure Modeling," *Psychological Bulletin*, 100, 107–120.

MacCallum, R. C., Roznowski, M., and Necowitz, L. B. (1992), "Model Modification in Covariance Structure Analysis: The Problem of Capitalization on Chance," *Psychological Bulletin*, 111, 490–504.

Mauchly, J. W. (1940), "Significance Test for Sphericity of a Normal N-Variate Distribution," *Annals of Mathematical Statistics*, 11, 204–209.

McArdle, J. J. (1980), "Causal Modeling Applied to Psychonomic Systems Simulation," *Behavior Research Methods & Instrumentation*, 12, 193–209.

McArdle, J. J. (1988), "Dynamic but Structural Equation Modeling of Repeated Measures Data," in *The Handbook of Multivariate Experimental Psychology*, ed. J. R. Nesselroade and R. B. Cattell, New York: Plenum Press.

McArdle, J. J. and McDonald, R. P. (1984), "Some Algebraic Properties of the Reticular Action Model," *British Journal of Mathematical and Statistical Psychology*, 37, 234–251.

McDonald, R. P. (1978), "A Simple Comprehensive Model for the Analysis of Covariance Structures," *British Journal of Mathematical and Statistical Psychology*, 31, 59–72.

McDonald, R. P. (1980), "A Simple Comprehensive Model for the Analysis of Covariance Structures: Some Remarks on Applications," *British Journal of Mathematical and Statistical Psychology*, 33, 161–183.

McDonald, R. P. (1985), *Factor Analysis and Related Methods*, Hillsdale, NJ: Lawrence Erlbaum Associates.

McDonald, R. P. (1989), "An Index of Goodness-of-Fit Based on Noncentrality," *Journal of Classification*, 6, 97–103.

McDonald, R. P. and Hartmann, W. (1992), "A Procedure for Obtaining Initial Values of Parameters in the RAM Model," *Multivariate Behavioral Research*, 27, 57–176.

Moré, J. J. (1978), "The Levenberg-Marquardt Algorithm: Implementation and Theory," in *Numerical Analysis–Dundee 1977*, ed. G. A. Watson, Lecture Notes in Mathematics 630, Berlin: Springer-Verlag.

Moré, J. J. and Sorensen, D. C. (1983), "Computing a Trust-Region Step," *SIAM Journal on Scientific and Statistical Computing*, 4, 553–572.

Mulaik, S. A., James, L. R., Van Alstine, J., Bennett, N., Lind, S., and Stilwell, C. D. (1989), "Evaluation of Goodness-of-Fit Indices for Structural Equation Models," *Psychological Bulletin*, 105, 430–445.

Mulaik, S. A. and Quartetti, D. A. (1997). "First Order or Higher Order General Factor," *Structural Equation Modeling*, 4, 193–211.

Polak, E. (1971), *Computational Methods in Optimization*, New York: Academic Press.

Powell, J. M. D. (1977), "Restart Procedures for the Conjugate Gradient Method," *Mathematical Programming*, 12, 241–254.

Powell, J. M. D. (1978a), "A Fast Algorithm for Nonlinearly Constraint Optimization Calculations," in *Numerical Analysis, Dundee 1977, Lecture Notes in Mathematics 630*, ed. G. A. Watson, Berlin: Springer Verlag, 144–175.

Powell, J. M. D. (1978b), "Algorithms for Nonlinear Constraints That Use Lagrangian Functions," *Mathematical Programming*, 14, 224–248.

Powell, M. J. D. (1982a), "Extensions to Subroutine VF02AD," in *Systems Modeling and Optimization, Lecture Notes In Control and Information Sciences 38*, ed. R. F. Drenick and F. Kozin, Berlin: Springer Verlag, 529–538.

Powell, J. M. D. (1982b), "VMCWD: A Fortran Subroutine for Constrained Optimization," *DAMTP 1982/NA4*, Cambridge, England.

Schmid, J. and Leiman, J. M. (1957), "The Development of Hierarchical Factor Solutions," *Psychometrika*, 22, 53–61.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Sclove, L. S. (1987), "Application of Model-Selection Criteria to Some Problems in Multivariate Analysis," *Psychometrika*, 52, 333–343.

Steiger, J. H. (1998), "A Note on Multiple Sample Extensions of the RMSEA Fit Index," *Structural Equation Modeling*, 5, 411–419.

Steiger, J. H. and Lind, J. C. (1980), "Statistically Based Tests for the Number of Common Factors," paper presented at the annual meeting of the Psychometric Society, Iowa City, IA.

Swaminathan, H. (1974), "A General Factor Model for the Description of Change," Report LR-74-9, Laboratory of Psychometric and Evaluative Research, University of Massachusetts.

Tucker, L. R. and Lewis, C. (1973), "A Reliability Coefficient for Maximum Likelihood Factor Analysis," *Psychometrika*, 38, 1–10.

Wheaton, B., Muthèn, B., Alwin, D. F., and Summers, G. F. (1977), "Assessing Reliability and Stability in Panel Models," in *Sociological Methodology*, ed. D. R. Heise, San Francisco: Jossey Bass.

Wiley, D. E. (1973), "The Identification Problem for Structural Equation Models with Unmeasured Variables," in *Structural Equation Models in the Social Sciences*, ed. A. S. Goldberger and O. D. Duncan, New York: Seminar Press, 69–83.

Wilson, E. B. and Hilferty, M. M. (1931), "The Distribution of Chi-Square," *Proceeding of the National Academy of Science*, 17, 694.

Yung, Y. F., Thissen, D., and McLeod, L. D. (1999), "On the Relationship Between the Higher-order Factor Model and the Hierarchical Factor Model," *Psychometrika*, 64, 113–128.

Chapter 89

The TPSPLINE Procedure

Contents

Overview: TPSPLINE Procedure **7057**
Penalized Least Squares Estimation 7058
PROC TPSPLINE with Large Data Sets 7059
Getting Started: TPSPLINE Procedure **7060**
Syntax: TPSPLINE Procedure **7068**
PROC TPSPLINE Statement 7068
BY Statement 7068
FREQ Statement 7069
ID Statement 7069
MODEL Statement 7069
SCORE Statement 7071
OUTPUT Statement 7072
Details: TPSPLINE Procedure **7073**
Computational Formulas 7073
ODS Table Names 7076
Examples: TPSPLINE Procedure **7077**
Example 89.1: Partial Spline Model Fit 7077
Example 89.2: Spline Model with Higher-Order Penalty 7079
Example 89.3: Multiple Minima of the GCV Function 7084
Example 89.4: Large Data Set Application 7089
Example 89.5: Computing a Bootstrap Confidence Interval 7094
References **7103**

Overview: TPSPLINE Procedure

The TPSPLINE procedure uses the penalized least squares method to fit a nonparametric regression model. It computes thin-plate smoothing splines to approximate smooth multivariate functions observed with noise. The TPSPLINE procedure allows great flexibility in the possible form of the regression surface. In particular, PROC TPSPLINE makes no assumptions of a parametric form for the model. The generalized cross validation (GCV) function can be used to select the amount of smoothing.

The TPSPLINE procedure complements the methods provided by the standard SAS regression procedures such as the GLM, REG, and NLIN procedures. These procedures can handle most situations in which you specify the regression model and the model is known up to a fixed number of parameters. However, when you have no prior knowledge about the model, or when you know that the data cannot be represented by a model with a fixed number of parameters, you can use the TPSPLINE procedure to model the data.

The TPSPLINE procedure uses the penalized least squares method to fit the data with a flexible model in which the number of effective parameters can be as large as the number of unique design points. Hence, as the sample size increases, the model space increases as well, enabling the thin-plate smoothing spline to fit more complicated situations.

The main features of the TPSPLINE procedure are as follows:

- provides penalized least squares estimates
- supports the use of multidimensional data
- supports multiple SCORE statements
- fits both semiparametric models and nonparametric models
- provides options for handling large data sets
- supports multiple dependent variables
- enables you to choose a particular model by specifying the model degrees of freedom or smoothing parameter

Penalized Least Squares Estimation

Penalized least squares estimation provide a way to balance fitting the data closely and avoiding excessive roughness or rapid variation. A penalized least squares estimate is a surface that minimizes the penalized least squares over the class of all surfaces satisfying sufficient regularity conditions.

Define $\mathbf{x}_i$ as a d-dimensional covariate vector from an $n \times d$ matrix $\mathbf{X}$, $\mathbf{z}_i$ as a p-dimensional covariate vector, and y_i as the observation associated with $(\mathbf{x}_i, \mathbf{z}_i)$. Assuming that the relation between $\mathbf{z}_i$ and y_i is linear but the relation between $\mathbf{x}_i$ and y_i is unknown, you can fit the data by using a semiparametric model as follows:

$$y_i = f(\mathbf{x}_i) + \mathbf{z}_i \boldsymbol{\beta} + \epsilon_i$$

where f is an unknown function that is assumed to be reasonably smooth, $\epsilon_i, i = 1, \cdots, n$, are independent, zero-mean random errors, and $\boldsymbol{\beta}$ is a p-dimensional unknown parametric vector.

This model consists of two parts. The $\mathbf{z}_i \boldsymbol{\beta}$ is the parametric part of the model, and the $\mathbf{z}_i$ are the regression variables. The $f(\mathbf{x}_i)$ is the nonparametric part of the model, and the $\mathbf{x}_i$ are the smoothing variables.

The ordinary least squares method estimates $f(\mathbf{x}_i)$ and $\boldsymbol{\beta}$ by minimizing the quantity:

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i) - \mathbf{z}_i\boldsymbol{\beta})^2$$

However, the functional space of $f(\mathbf{x})$ is so large that you can always find a function f that interpolates the data points. In order to obtain an estimate that fits the data well and has some degree of smoothness, you can use the penalized least squares method.

The penalized least squares function is defined as

$$S_\lambda(f) = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i) - \mathbf{z}_i\boldsymbol{\beta})^2 + \lambda J_2(f)$$

where $J_2(f)$ is the penalty on the roughness of f and is defined, in most cases, as the integral of the square of the second derivative of f.

The first term measures the goodness of fit and the second term measures the smoothness associated with f. The λ term is the smoothing parameter, which governs the trade-off between smoothness and goodness of fit. When λ is large, it more heavily penalizes rougher fits. Conversely, a small value of λ puts more emphasis on the goodness of fit.

The estimate f_λ is selected from a reproducing kernel Hilbert space, and it can be represented as a linear combination of a sequence of basis functions. Hence, the final estimates of f can be written as

$$\hat{f}_\lambda(\mathbf{x}_i) = \theta_0 + \sum_{j=1}^{d}\theta_j\mathbf{x}_{ij} + \sum_{j=1}^{p}\delta_j B_j(\mathbf{x}_j)$$

where B_j is the basis function, which depends on where the data $\mathbf{x}_i$ are located, and $\boldsymbol{\theta} = \{\theta_0, \ldots, \theta_d\}$ and $\boldsymbol{\delta} = \{\delta_1, \ldots, \delta_p\}$ are the coefficients that need to be estimated.

For a fixed λ, the coefficients $(\boldsymbol{\theta}, \boldsymbol{\delta}, \boldsymbol{\beta})$ can be estimated by solving an $n \times n$ system.

The smoothing parameter can be chosen by minimizing the generalized cross validation (GCV) function.

If you write

$$\hat{\mathbf{y}} = \mathbf{A}(\lambda)\mathbf{y}$$

then $\mathbf{A}(\lambda)$ is referred to as the *hat* or *smoothing* matrix, and the GCV function $GCV(\lambda)$ is defined as

$$GCV(\lambda) = \frac{(1/n)\|(\mathbf{I} - \mathbf{A}(\lambda))\mathbf{y}\|^2}{[(1/n)\mathrm{tr}(\mathbf{I} - \mathbf{A}(\lambda))]^2}$$

PROC TPSPLINE with Large Data Sets

The calculation of the penalized least squares estimate is computationally intensive. The amount of memory and CPU time needed for the analysis depends on the number of unique design points, which corresponds to the number of unknown parameters to be estimated.

You can specify the D= option in the MODEL statement to reduce the number of unknown parameters. The option groups design points by the specified range (see the D= option on page 7070).

PROC TPSPLINE selects one design point from the group and treats all observations in the group as replicates of that design point. Calculation of the thin-plate smoothing spline estimates is based on the reprocessed data. The way to choose the design point from a group depends on the order of the data. Hence, different orders of input data might result in different estimates.

This option, by combining several design points into one, reduces the number of unique design points, thereby approximating the original data. The value you specify for the D= option determines the width of the range used to group the data.

Getting Started: TPSPLINE Procedure

The following example demonstrates how you can use the TPSPLINE procedure to fit a semiparametric model.

Suppose that y is a continuous variable and x1 and x2 are two explanatory variables of interest. To fit a bivariate thin-plate spline model, you can use a MODEL statement similar to that used in many regression procedures in the SAS System:

```
proc tpspline;
    model y = (x1 x2);
run;
```

The TPSPLINE procedure can fit semiparametric models; the parentheses in the preceding MODEL statement separate the smoothing variables from the regression variables. The following statements illustrate this syntax:

```
proc tpspline;
    model y = z1 (x1 x2);
run;
```

This model assumes a linear relation with z1 and an unknown functional relation with x1 and x2.

If you want to fit several responses by using the same explanatory variables, you can save computation time by using the multiple responses feature in the MODEL statement. For example, if y1 and y2 are two response variables, the following MODEL statement can be used to fit two models. Separate analyses are then performed for each response variable.

```
proc tpspline;
    model y1 y2 = (x1 x2);
run;
```

The following example illustrates the use of PROC TPSPLINE. The data are from Bates et al. (1987).

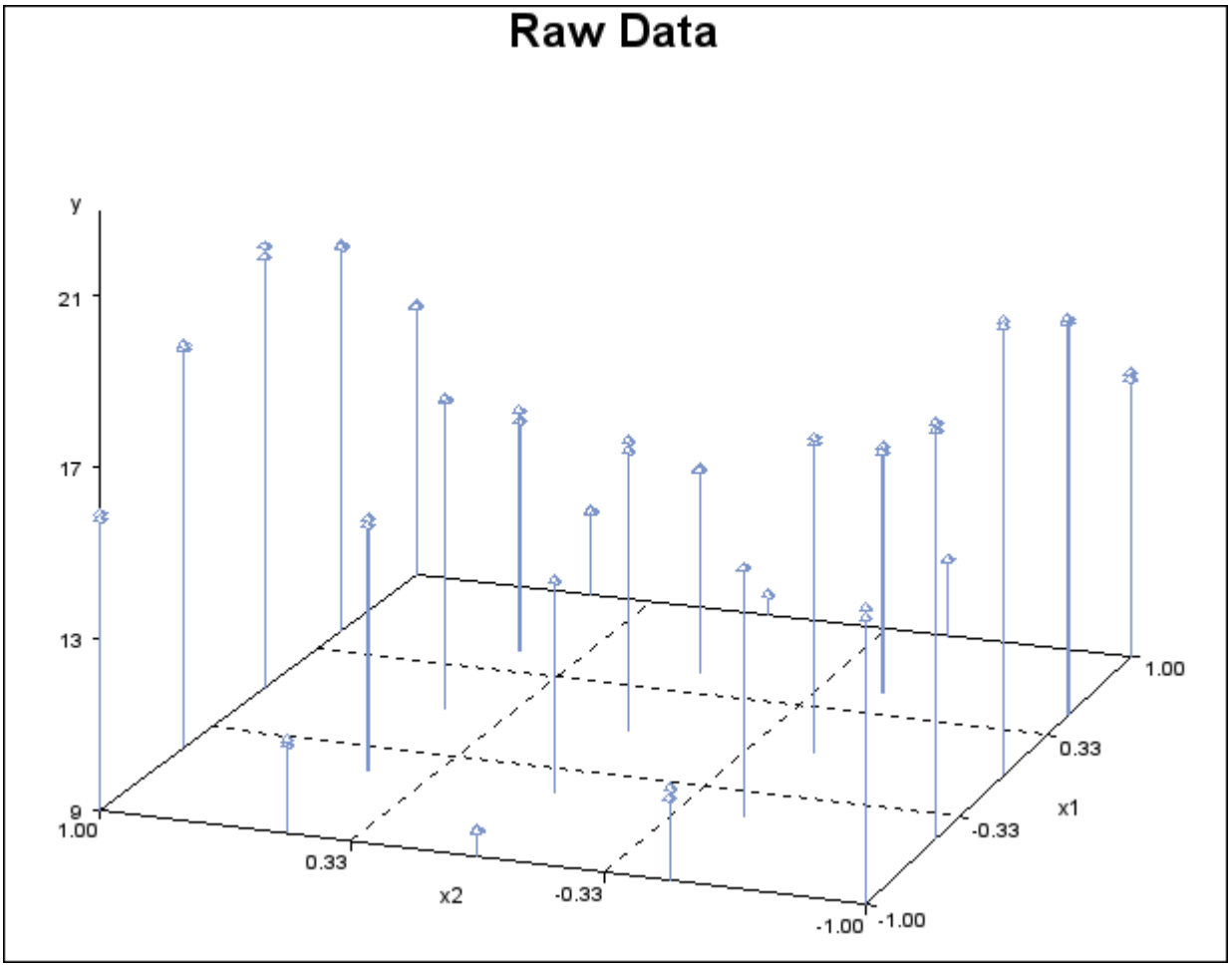
```
data Measure;
    input x1 x2 y @@;
datalines;
-1.0 -1.0   15.54483570   -1.0 -1.0    15.76312613
 -.5 -1.0   18.67397826    -.5 -1.0    18.49722167
  .0 -1.0   19.66086310     .0 -1.0    19.80231311
  .5 -1.0   18.59838649     .5 -1.0    18.51904737
 1.0 -1.0   15.86842815    1.0 -1.0    16.03913832
-1.0  -.5   10.92383867   -1.0  -.5    11.14066546
 -.5  -.5   14.81392847    -.5  -.5    14.82830425
  .0  -.5   16.56449698     .0  -.5    16.44307297
  .5  -.5   14.90792284     .5  -.5    15.05653924
 1.0  -.5   10.91956264    1.0  -.5    10.94227538
-1.0   .0    9.61492010   -1.0   .0     9.64648093
 -.5   .0   14.03133439    -.5   .0    14.03122345
  .0   .0   15.77400253     .0   .0    16.00412514
  .5   .0   13.99627680     .5   .0    14.02826553
 1.0   .0    9.55700164    1.0   .0     9.58467047
-1.0   .5   11.20625177   -1.0   .5    11.08651907
 -.5   .5   14.83723493    -.5   .5    14.99369172
  .0   .5   16.55494349     .0   .5    16.51294369
  .5   .5   14.98448603     .5   .5    14.71816070
 1.0   .5   11.14575565    1.0   .5    11.17168689
-1.0  1.0   15.82595514   -1.0  1.0    15.96022497
 -.5  1.0   18.64014953    -.5  1.0    18.56095997
  .0  1.0   19.54375504     .0  1.0    19.80902641
  .5  1.0   18.56884576     .5  1.0    18.61010439
 1.0  1.0   15.86586951    1.0  1.0    15.90136745
;
```

The data set Measure contains three variables x1, x2, and y. Suppose that you want to fit a surface by using the variables x1 and x2 to model the response y. The variables x1 and x2 are spaced evenly on a $[-1 \times 1] \times [-1 \times 1]$ square, and the response y is generated by adding a random error to a function $f(x_1, x_2)$. The raw data are plotted by using the G3D procedure. In order to visualize those replicates, half of the data are shifted a little bit by adding a small value (0.001) to x1 values, as in the following statements:

```
data Measure1;
   set Measure;
run;

proc sort data=Measure1;
   by x2 x1;
run;

data Measure1;
   set Measure1;
   if mod(_N_, 2) = 0 then x1=x1+0.001;
run;

proc g3d data=Measure1;
   scatter x2*x1=y /size=.5
                    zmin=9 zmax=21
                    zticknum=4;
   title "Raw Data";
run;
```

Figure 89.1 displays the raw data.

Figure 89.1 Plot of Data Set MEASURE



The following statements invoke the TPSPLINE procedure, by using the Measure data set as input. In the MODEL statement, the x1 and x2 variables are listed as smoothing variables. The LOGNLAMBDA= option returns a list of GCV values with $\log_{10}(n\lambda)$ ranging from -4 to -2. The OUTPUT statement creates the data set estimate to contain the predicted values and the 95% upper and lower confidence limits.

```
proc tpspline data=Measure;
   model y=(x1 x2) /lognlambda=(-4 to -2 by 0.1);
   output out=estimate pred uclm lclm;
run;

proc print data=estimate;
run;
```

The results of this analysis are displayed in the following figures. Figure 89.2 shows that the data set Measure contains 50 observations with 25 unique design points. The final model contains no parametric regression terms and 2 smoothing variables. The order derivative in the penalty is 2 by default, and the dimension of polynomial space is 3. See the section "Computational Formulas" on page 7073 for definitions.

The GCV values are listed along with the $\log_{10}$ of $n\lambda$. The value of $\log_{10}(n\lambda)$ that minimizes the GCV function is around -3.5. The final thin-plate smoothing spline estimate is based on LOGNLAMBDA = -3.4762. The residual sum of squares is 0.246110, and the degrees of freedom are 24.593203. The standard deviation, defined as $RSS/(\text{tr}(\mathbf{I}-\mathbf{A}))$, is 0.098421. The predictions and 95% confidence limits are displayed in Figure 89.3.

Figure 89.2 Output from PROC TPSPLINE

Raw Data

The TPSPLINE Procedure
Dependent Variable: y

Summary of Input Data Set

Number of Non-Missing Observations	50
Number of Missing Observations	0
Unique Smoothing Design Points	25

Summary of Final Model

Number of Regression Variables	0
Number of Smoothing Variables	2
Order of Derivative in the Penalty	2
Dimension of Polynomial Space	3

Figure 89.2 *continued*

GCV Function

log10(n*Lambda)	GCV
-4.000000	0.019215
-3.900000	0.019183
-3.800000	0.019148
-3.700000	0.019113
-3.600000	0.019082
-3.500000	0.019064*
-3.400000	0.019074
-3.300000	0.019135
-3.200000	0.019286
-3.100000	0.019584
-3.000000	0.020117
-2.900000	0.021015
-2.800000	0.022462
-2.700000	0.024718
-2.600000	0.028132
-2.500000	0.033165
-2.400000	0.040411
-2.300000	0.050614
-2.200000	0.064699
-2.100000	0.083813
-2.000000	0.109387

Note: * indicates minimum GCV value.

Summary Statistics
of Final Estimation

log10(n*Lambda)	-3.4762
Smoothing Penalty	2558.1432
Residual SS	0.2461
Tr(I-A)	25.4068
Model DF	24.5932
Standard Deviation	0.0984

Figure 89.3 Data Set ESTIMATE

Raw Data

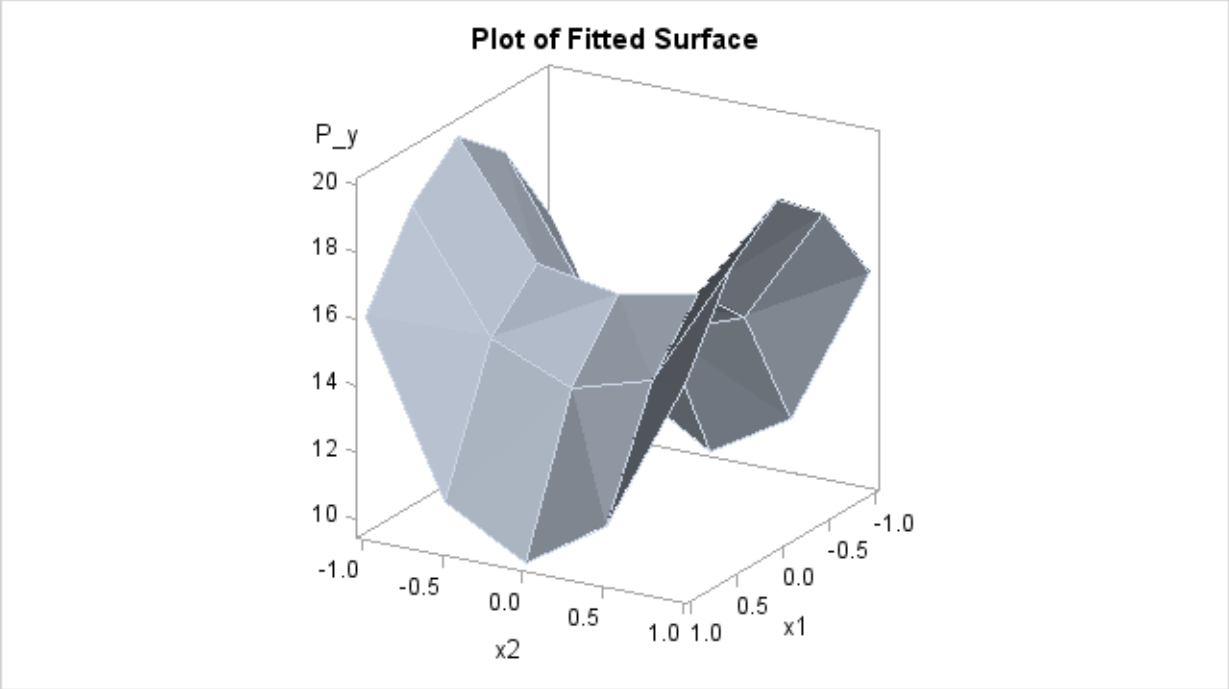
Obs	x1	x2	y	P_y	LCLM_y	UCLM_y
1	-1.0	-1.0	15.5448	15.6474	15.5115	15.7832
2	-1.0	-1.0	15.7631	15.6474	15.5115	15.7832
3	-0.5	-1.0	18.6740	18.5783	18.4430	18.7136
4	-0.5	-1.0	18.4972	18.5783	18.4430	18.7136
5	0.0	-1.0	19.6609	19.7270	19.5917	19.8622
6	0.0	-1.0	19.8023	19.7270	19.5917	19.8622
7	0.5	-1.0	18.5984	18.5552	18.4199	18.6905
8	0.5	-1.0	18.5190	18.5552	18.4199	18.6905
9	1.0	-1.0	15.8684	15.9436	15.8077	16.0794
10	1.0	-1.0	16.0391	15.9436	15.8077	16.0794
11	-1.0	-0.5	10.9238	11.0467	10.9114	11.1820
12	-1.0	-0.5	11.1407	11.0467	10.9114	11.1820
13	-0.5	-0.5	14.8139	14.8246	14.6896	14.9597
14	-0.5	-0.5	14.8283	14.8246	14.6896	14.9597
15	0.0	-0.5	16.5645	16.5102	16.3752	16.6452
16	0.0	-0.5	16.4431	16.5102	16.3752	16.6452
17	0.5	-0.5	14.9079	14.9812	14.8461	15.1162
18	0.5	-0.5	15.0565	14.9812	14.8461	15.1162
19	1.0	-0.5	10.9196	10.9497	10.8144	11.0850
20	1.0	-0.5	10.9423	10.9497	10.8144	11.0850
21	-1.0	0.0	9.6149	9.6372	9.5019	9.7724
22	-1.0	0.0	9.6465	9.6372	9.5019	9.7724
23	-0.5	0.0	14.0313	14.0188	13.8838	14.1538
24	-0.5	0.0	14.0312	14.0188	13.8838	14.1538
25	0.0	0.0	15.7740	15.8822	15.7472	16.0171
26	0.0	0.0	16.0041	15.8822	15.7472	16.0171
27	0.5	0.0	13.9963	14.0006	13.8656	14.1356
28	0.5	0.0	14.0283	14.0006	13.8656	14.1356
29	1.0	0.0	9.5570	9.5769	9.4417	9.7122
30	1.0	0.0	9.5847	9.5769	9.4417	9.7122
31	-1.0	0.5	11.2063	11.1614	11.0261	11.2967
32	-1.0	0.5	11.0865	11.1614	11.0261	11.2967
33	-0.5	0.5	14.8372	14.9182	14.7831	15.0532
34	-0.5	0.5	14.9937	14.9182	14.7831	15.0532
35	0.0	0.5	16.5549	16.5386	16.4036	16.6736
36	0.0	0.5	16.5129	16.5386	16.4036	16.6736
37	0.5	0.5	14.9845	14.8549	14.7199	14.9900
38	0.5	0.5	14.7182	14.8549	14.7199	14.9900
39	1.0	0.5	11.1458	11.1727	11.0374	11.3080
40	1.0	0.5	11.1717	11.1727	11.0374	11.3080
41	-1.0	1.0	15.8260	15.8851	15.7493	16.0210
42	-1.0	1.0	15.9602	15.8851	15.7493	16.0210
43	-0.5	1.0	18.6401	18.5946	18.4593	18.7299
44	-0.5	1.0	18.5610	18.5946	18.4593	18.7299
45	0.0	1.0	19.5438	19.6729	19.5376	19.8081
46	0.0	1.0	19.8090	19.6729	19.5376	19.8081
47	0.5	1.0	18.5688	18.5832	18.4478	18.7185
48	0.5	1.0	18.6101	18.5832	18.4478	18.7185
49	1.0	1.0	15.8659	15.8761	15.7402	16.0120
50	1.0	1.0	15.9014	15.8761	15.7402	16.0120

The fitted surface is plotted with PROC TEMPLATE and PROC SGRENDER as follows:

```
data pred_half;
   set estimate;
      if mod(_N_,2)=0 then output;
run;

proc template;
    define statgraph surface;
    dynamic _X _Y _Z _T;
    begingraph /designheight=360;
        entrytitle _T;
        layout overlay3d/rotate=120 cube=false
                         xaxisopts=(label="x1")
                         yaxisopts=(label="x2")
                         zaxisopts=(label="P_y");
                         surfaceplotparm x=_X y=_Y z=_Z;
        endlayout;
    endgraph;
    end;
run;

ods graphics on;

proc sgrender data=pred_half template=surface;
    dynamic _X='x1' _Y='x2' _Z='y' _T='Plot of Fitted Surface';
run;
```

The resulting plot is displayed in Figure 89.4.

Figure 89.4 Plot of PROC TPSPLINE Fit of Data Set MEASURE

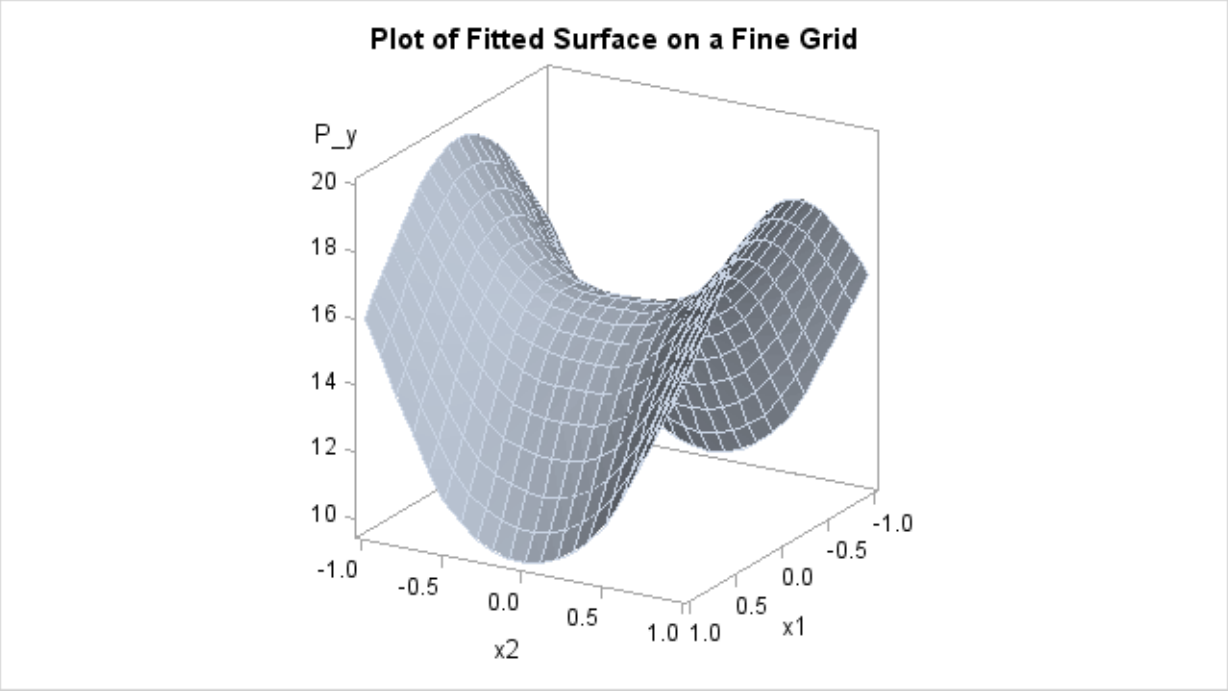


Because the data in the data set Measure are very sparse, the fitted surface is not smooth. To produce a smoother surface, the following statements generate the data set pred in order to obtain a finer grid. The SCORE statement evaluates the fitted surface at those new design points.

```
data pred;
   do x1=-1 to 1 by 0.1;
      do x2=-1 to 1 by 0.1;
         output;
      end;
   end;
run;

proc tpspline data=measure;
   model y=(x1 x2)/lognlambda=(-4 to -2 by 0.1);
   score data=pred out=predy;
run;

proc sgrender data=predy template=surface;
    dynamic _X='x1' _Y='x2' _Z='P_y' _T='Plot of Fitted Surface on a Fine Grid';
run;

ods graphics off;
```

The surface plot based on the finer grid is displayed in Figure 89.5. The plot shows that a parametric model with quadratic terms of x1 and x2 provides a reasonable fit to the data.

Figure 89.5 Plot of TPSPLINE Fit



Syntax: TPSPLINE Procedure

The following statements are available in PROC TPSPLINE:

PROC TPSPLINE < *option* > **;**
MODEL *dependents* = < *variables* > *(variables)* < */options* > **;**
SCORE *data=SAS-data-set out=SAS-data-set* **;**
OUTPUT < *out=SAS-data-set* > *keyword* ··· *keyword* **;**
BY *variables* **;**
FREQ *variable* **;**
ID *variables* **;**

The syntax in PROC TPSPLINE is similar to that of other regression procedures in the SAS System. The PROC TPSPLINE and MODEL statements are required. The SCORE statement can appear multiple times; all other statements appear only once.

The statements available for PROC TPSPLINE are described in alphabetical order after the description of the PROC TPSPLINE statement.

PROC TPSPLINE Statement

PROC TPSPLINE < *option* > **;**

The PROC TPSPLINE statement invokes the procedure. You can specify the following option.

DATA=*SAS-data-set*
specifies the SAS data set to be read by PROC TPSPLINE. The default value is the most recently created data set.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC TPSPLINE to obtain separate analysis on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the TPSPLINE procedure. The NOTSORTED option does not mean that the data are unsorted

but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.

- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see the discussion in *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the discussion in the *Base SAS Procedures Guide*.

FREQ Statement

FREQ *variable* **;**

If one variable in your input data set represents the frequency of occurrence for other values in the observation, specify the variable's name in a FREQ statement. PROC TPSPLINE treats the data as if each observation appears n times, where n is the value of the FREQ variable for the observation. If the value of the FREQ variable is less than one, the observation is not used in the analysis. Only the integer portion of the value is used.

ID Statement

ID *variables* **;**

The variables in the ID statement are copied from the input data set to the OUT= data set. If you omit the ID statement, only the variables used in the MODEL statement and requested statistics are included in the output data set.

MODEL Statement

MODEL *dependents = < regression variables > (smoothing variables) < /options >* **;**

The MODEL statement specifies the dependent variables, the independent regression variables, which are listed with no parentheses, and the independent smoothing variables, which are listed inside parentheses.

The regression variables are optional. At least one smoothing variable is required, and it must be listed after the regression variables. No variables can be listed in both the regression variable list and the smoothing variable list.

If you specify more than one dependent variable, PROC TPSPLINE calculates a thin-plate smoothing spline estimate for each dependent variable, by using the regression variables and smoothing variables specified on the right side.

If you specify regression variables, PROC TPSPLINE fits a semiparametric model by using the regression variables as the linear part of the model.

You can specify the following options in the MODEL statement.

ALPHA=*number*

specifies the significance level α of the confidence limits on the final thin-plate smoothing spline estimate when you request confidence limits to be included in the output data set. Specify *number* as a value between 0 and 1. The default value is 0.05. See the section "OUTPUT Statement" on page 7072 for more information about the OUTPUT statement.

DF=*number*

specifies the degrees of freedom of the thin-plate smoothing spline estimate, defined as

$$\mathrm{df} = \mathrm{tr}(\mathbf{A}(\lambda))$$

where $\mathbf{A}(\lambda)$ is the *hat* matrix. Specify *number* as a value between zero and the number of unique design points n_q. Smaller df values cause more penalty on the roughness and thus smoother fits.

DISTANCE=*number*

D=*number*

defines a range such that if the L_∞ distance between two data points $(\mathbf{x}_i, \mathbf{z}_i)$ and $(\mathbf{x}_j, \mathbf{z}_j)$ satisfies

$$\|\mathbf{x}_i - \mathbf{x}_j\|_\infty \leq D/2$$

then these data points are treated as replicates, where $\mathbf{x}_i$ are the smoothing variables and $\mathbf{z}_i$ are the regression variables.

You can use the DISTANCE= option to reduce the number of unique design points by treating nearby data as replicates. This can be useful when you have a large data set. Larger DISTANCE= option values cause fewer n_q points. The default value is 0.

PROC TPSPLINE uses the DISTANCE= value to group points as follows: The data are first sorted by the smoothing variables in the order in which they appear in the MODEL statement. The first point in the sorted data becomes the first unique point. Subsequent points have their x-values set equal to that point until the first point where the maximum distance in one dimension is larger than $D/2$. This point becomes the next unique point, and so on. Because of this sequential processing, the set of unique points differs depending on the order of the smoothing variables in the MODEL statement.

For example, with a model that has two smoothing variables (x1,x2), the data are first sorted by x1 and x2 (in that order), and then uniqueness is assessed sequentially. The first point in the sorted data $\mathbf{x}_1 = (\mathsf{x1}_1, \mathsf{x2}_1)$ becomes the first unique point, $\mathbf{u}_1 = (\mathsf{u1}_1, \mathsf{u2}_1)$. Subsequent points $\mathbf{x}_i = (\mathsf{x1}_i, \mathsf{x2}_i)$ are set equal to $\mathbf{u}_1$ until the algorithm comes to a point with $\max(|\mathsf{x1}_i - \mathsf{u1}_1|, |\mathsf{x2}_i - \mathsf{u2}_1|) > D/2$. This point becomes the second unique point $\mathbf{u}_2$, and we proceed from there.

LAMBDA0=*number*

specifies the smoothing parameter, λ_0, to be used in the thin-plate smoothing spline estimate. By default, PROC TPSPLINE uses the λ parameter that minimizes the GCV function for the final fit. The LAMBDA0= value must be positive. Larger λ_0 values cause smoother fits.

LAMBDA=*list-of-values*

specifies a set of values for the λ parameter. PROC TPSPLINE returns a GCV value for each λ point that you specify. You can use the LAMBDA= option to study the GCV function curve for a set of values for λ. All values listed in the LAMBDA= option must be positive.

LOGNLAMBDA0=*number*

LOGNL0=*number*

specifies the smoothing parameter λ_0 on the $\log_{10}(n\lambda)$ scale. If you specify both the LOGNL0= and LAMBDA0= options, only the value provided by the LOGNL0= option is used. Larger $\log_{10}(n\lambda_0)$ values cause smoother fits. By default, PROC TPSPLINE uses the λ parameter that minimizes the GCV function for the estimate.

LOGNLAMBDA=*list-of-values*

LOGNL=*list-of-values*

specifies a set of values for the λ parameter on the $\log_{10}(n\lambda)$ scale. PROC TPSPLINE returns a GCV value for each λ point that you specify. You can use the LOGNLAMBDA= option to study the GCV function curve for a set of λ values. If you specify both the LOGNL= and LAMBDA= options, only the list of values provided by the LOGNL= option is used.

In some cases, the LOGNL= option might be preferred over the LAMBDA= option. Because the LAMBDA= value must be positive, a small change in that value can result in a major change in the GCV value. If you instead specify λ on the $\log_{10}(n\lambda)$ scale, the allowable range is enlarged to include negative values. Thus, the GCV function is less sensitive to changes in LOGNLAMBDA.

The DF= option, LAMBDA0= option, and LOGNLAMBDA0= option all specify exact smoothness of a nonparametric fit. If you want to fit a model with specified smoothness, the DF= option is preferable to the other two options because $(0, n_q)$, the range of df, is much smaller in length than $(0, \infty)$ of λ and $(-\infty, \infty)$ of $\log_{10}(n\lambda)$.

M=*number*

specifies the order of the derivative in the penalty term. The M= value must be a positive integer. The default value is $\max(2, \text{int}(d/2) + 1)$, where d is the number of smoothing variables.

SCORE Statement

SCORE *DATA=SAS-data-set* *OUT=SAS-data-set* **;**

The SCORE statement calculates predicted values for a new data set. If you have multiple data sets to predict, you can specify multiple SCORE statements. You must use a SCORE statement for each data set.

The following keywords must be specified in the SCORE statement.

DATA=*SAS-data-set*
specifies the input SAS data set containing the smoothing variables **x** and regression variables **z**. The predicted response ($\hat{y}$) value is computed for each (**x**, **z**) pair. The data set must include all independent variables specified in the MODEL statement.

OUT=*SAS-data-set*
specifies the name of the SAS data set to contain the predictions.

OUTPUT Statement

OUTPUT *OUT=SAS-data-set <keyword ··· keyword >* **;**

The OUTPUT statement creates a new SAS data set containing diagnostic measures calculated after fitting the model.

You can request a variety of diagnostic measures that are calculated for each observation in the data set. The new data set contains the variables specified in the MODEL statement in addition to the requested variables. If no *keyword* is present, the data set contains only the predicted values.

Details on the specifications in the OUTPUT statement are as follows.

OUT=*SAS-data-set*
specifies the name of the new data set to contain the diagnostic measures. This specification is required.

keyword
specifies the statistics to include in the output data set. The names of the new variables that contain the statistics are formed by using a prefix of one or more characters to identify the statistic, followed by an underscore (_), followed by the dependent variable name.

For example, suppose that you have two dependent variables—say, y1 and y2—and you specify the keywords PRED, ADIAG, and UCLM. The output SAS data set will contain the following variables:

- P_y1 and P_y2
- ADIAG_y1 and ADIAG_y2
- UCLM_y1 and UCLM_y2

The keywords and the statistics they represent are as follows:

RESID | R	residual values, calculated as ACTUAL − PREDICTED
PRED	predicted values
STD	standard error of the mean predicted value

UCLM	upper limit of the confidence interval for the expected value of the dependent variables. By default, PROC TPSPLINE computes 95% confidence limits.
LCLM	lower limit of the confidence interval for the expected value of the dependent variables. By default, PROC TPSPLINE computes 95% confidence limits.
ADIAG	diagonal element of the hat matrix associated with the observation
COEF	coefficients arranged in the order of $(\theta_0, \theta_1, \cdots, \theta_d, \delta_1, \cdots \delta_{n_q})$, where n_q is the number of unique data points. This option can be used only when there is only one dependent variable in the model.

Details: TPSPLINE Procedure

Computational Formulas

The theoretical foundations for the thin-plate smoothing spline are described in Duchon (1976, 1977) and Meinguet (1979). Further results and applications are given in Wahba and Wendelberger (1980), Hutchinson and Bischof (1983), and Seaman and Hutchinson (1985).

Suppose that $\mathcal{H}_m$ is a space of functions whose partial derivatives of total order m are in $L_2(E^d)$, where E^d is the domain of $\mathbf{x}$.

Now, consider the data model

$$y_i = f(\mathbf{x}_i) + \epsilon_i, i = 1, \ldots, n$$

where $f \in \mathcal{H}_m$.

Using the notation from the section "Penalized Least Squares Estimation" on page 7058, for a fixed λ, estimate f by minimizing the penalized least squares function

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i) - \mathbf{z}_i\boldsymbol{\beta})^2 + \lambda J_m(f)$$

$\lambda J_m(f)$ is the penalty term to enforce smoothness on f. There are several ways to define $J_m(f)$. For the thin-plate smoothing spline, with $\mathbf{X}$ of dimension d, define $J_m(f)$ as

$$J_m(f) = \int_{-\infty}^{\infty}\cdots\int_{-\infty}^{\infty}\sum\frac{m!}{\alpha_1!\cdots\alpha_d!}\left(\frac{\partial^m f}{\partial x_1{}^{\alpha_1}\cdots\partial x_d{}^{\alpha_d}}\right)^2 \mathrm{d}x_1\cdots\mathrm{d}x_d$$

where $\sum_i \alpha_i = m$. Under this definition, $J_m(f)$ will give zero penalty to some functions. The space that is spanned by the set of polynomials contributing zero penalty is called the polynomial space. The dimension of the polynomial space M is a function of dimension d and order m of the smoothing penalty.

When $d = 2$ and $m = 2$, $J_m(f)$ is as follows:

$$J_2(f) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left(\left(\frac{\partial^2 f}{\partial {x_1}^2}\right)^2 + 2\left(\frac{\partial^2 f}{\partial x_1 \partial x_2}\right)^2 + \left(\frac{\partial^2 f}{\partial {x_2}^2}\right)^2\right) \mathrm{d}x_1 \mathrm{d}x_2$$

and $M = \text{sizeof}(\{1, x_1, x_2\}) = 3$.

In general, m and d must satisfy the condition that $2m - d > 0$. For the sake of simplicity, the formulas and equations that follow assume $m = 2$. See Wahba (1990) and Bates et al. (1987) for more details.

Duchon (1976) showed that f_λ can be represented as

$$f_\lambda(\mathbf{x}_i) = \theta_0 + \sum_{j=1}^{d} \theta_j \mathbf{x}_{i\,j} + \sum_{j=1}^{n} \delta_j E_2(\mathbf{x}_i - \mathbf{x}_j)$$

where $E_2(\mathbf{s}) = \frac{1}{2^3\pi}\|\mathbf{s}\|^2 \log(\|\mathbf{s}\|)$ for $d = 2$. For derivations of $E_2(\mathbf{s})$ for other values of d, see Villalobos and Wahba (1987).

If you define $\mathbf{K}$ with elements $\mathbf{K}_{ij} = E_2(\mathbf{x}_i - \mathbf{x}_j)$ and $\mathbf{T}$ with elements $\mathbf{T}_{ij} = (\mathbf{X}_{ij})$, the goal is to find coefficients β, θ, and δ that minimize

$$S_\lambda(\beta, \theta, \delta) = \frac{1}{n}\|\mathbf{y} - \mathbf{T}\theta - \mathbf{K}\delta - \mathbf{Z}\beta\|^2 + \lambda \delta^T \mathbf{K} \delta$$

A unique solution is guaranteed if the matrix $\mathbf{T}$ is of full rank and $\delta^T \mathbf{K} \delta \geq 0$.

If $\alpha = \begin{pmatrix} \theta \\ \beta \end{pmatrix}$ and $\mathbf{X} = (\mathbf{T}\ \mathbf{Z})$, the expression for S_λ becomes

$$\frac{1}{n}\|\mathbf{y} - \mathbf{X}\alpha - \mathbf{K}\delta\|^2 + \lambda \delta^T \mathbf{K} \delta$$

The coefficients α and δ can be obtained by solving

$$\begin{aligned} (\mathbf{K} + n\lambda \mathbf{I}_n)\delta + \mathbf{X}\alpha &= \mathbf{y} \\ \mathbf{X}^T \delta &= \mathbf{0} \end{aligned}$$

To compute α and δ, let the QR decomposition of $\mathbf{X}$ be

$$\mathbf{X} = (\mathbf{Q}_1\ \mathbf{Q}_2)\begin{pmatrix} \mathbf{R} \\ \mathbf{0} \end{pmatrix}$$

where $(\mathbf{Q}_1\ \mathbf{Q}_2)$ is an orthogonal matrix and $\mathbf{R}$ is an upper triangular, with $\mathbf{X}^T\mathbf{Q}_2 = 0$ (Dongarra et al. 1979).

Since $\mathbf{X}^T\delta = 0$, δ must be in the column space of $\mathbf{Q}_2$. Therefore, δ can be expressed as $\delta = \mathbf{Q}_2\gamma$ for a vector γ. Substituting $\delta = \mathbf{Q}_2\gamma$ into the preceding equation and multiplying through by $\mathbf{Q}_2^T$ gives

$$\mathbf{Q}_2^T(\mathbf{K} + n\lambda\mathbf{I})\mathbf{Q}_2\gamma = \mathbf{Q}_2^T\mathbf{y}$$

or

$$\delta = \mathbf{Q}_2\gamma = \mathbf{Q}_2[\mathbf{Q}_2^T(\mathbf{K} + n\lambda\mathbf{I})\mathbf{Q}_2]^{-1}\mathbf{Q}_2^T\mathbf{y}$$

The coefficient α can be obtained by solving

$$\mathbf{R}\alpha = \mathbf{Q}_1^T[\mathbf{y} - (\mathbf{K} + n\lambda\mathbf{I})\delta]$$

The influence matrix $\mathbf{A}(\lambda)$ is defined as

$$\hat{\mathbf{y}} = \mathbf{A}(\lambda)\mathbf{y}$$

and has the form

$$\mathbf{A}(\lambda) = \mathbf{I} - n\lambda\mathbf{Q}_2[\mathbf{Q}_2^T(K + n\lambda\mathbf{I})\mathbf{Q}_2]^{-1}\mathbf{Q}_2^T$$

Similar to the regression case, and if you consider the trace of $\mathbf{A}(\lambda)$ as the degrees of freedom for the model and the trace of $(\mathbf{I} - \mathbf{A}(\lambda))$ as the degrees of freedom for the error, the estimate σ^2 can be represented as

$$\hat{\sigma}^2 = \frac{RSS(\lambda)}{\text{tr}(\mathbf{I} - \mathbf{A}(\lambda))}$$

where $RSS(\lambda)$ is the residual sum of squares. Theoretical properties of these estimates have not yet been published. However, good numerical results in simulation studies have been described by several authors. For more information, see O'Sullivan and Wong (1987), Nychka (1986a, 1986b, 1988), and Hall and Titterington (1987).

Confidence Intervals

Viewing the spline model as a Bayesian model, Wahba (1983) proposed Bayesian confidence intervals for smoothing spline estimates as follows:

$$\hat{f}_\lambda(\mathbf{x}_i) \pm z_{\alpha/2}\sqrt{\hat{\sigma}^2 a_{ii}(\lambda)}$$

where $a_{ii}(\lambda)$ is the ith diagonal element of the $\mathbf{A}(\lambda)$ matrix and $z_{\alpha/2}$ is the $1 - \alpha/2$ quantile of the standard normal distribution. The confidence intervals are interpreted as intervals "across the function" as opposed to pointwise intervals.

Suppose that you fit a spline estimate that consists of a true function f and a random error term, ϵ_i, to experimental data. In repeated experiments, it is likely that about $100(1-\alpha)\%$ of the confidence intervals cover the corresponding true values, although some values are covered every time and other values are not covered by the confidence intervals most of the time. This effect is more pronounced when the true surface or surface has small regions of particularly rapid change.

Smoothing Parameter

The quantity λ is called the smoothing parameter, which controls the balance between the goodness of fit and the smoothness of the final estimate.

A large λ heavily penalizes the mth derivative of the function, thus forcing $f^{(m)}$ close to 0. A small λ places less of a penalty on rapid change in $f^{(m)}(\mathbf{x})$, resulting in an estimate that tends to interpolate the data points.

The smoothing parameter greatly affects the analysis, and it should be selected with care. One method is to perform several analyses with different values for λ and compare the resulting final estimates.

A more objective way to select the smoothing parameter λ is to use the "leave-out-one" cross validation function, which is an approximation of the predicted mean squares error. A generalized version of the leave-out-one cross validation function is proposed by Wahba (1990) and is easy to calculate. This generalized cross validation (GCV) function is defined as

$$GCV(\lambda) = \frac{(1/n)\|(\mathbf{I} - \mathbf{A}(\lambda))\mathbf{y}\|^2}{[(1/n)\mathrm{tr}(\mathbf{I} - \mathbf{A}(\lambda))]^2}$$

The justification for using the GCV function to select λ relies on asymptotic theory. Thus, you cannot expect good results for very small sample sizes or when there is not enough information in the data to separate the model from the error component. Simulation studies suggest that for independent and identically distributed Gaussian noise, you can obtain reliable estimates of λ for n greater than 25 or 30. Note that, even for large values of n (say, $n \geq 50$), in extreme Monte Carlo simulations there might be a small percentage of unwarranted extreme estimates in which $\hat{\lambda} = 0$ or $\hat{\lambda} = \infty$ (Wahba 1983). Generally, if σ^2 is known to within an order of magnitude, the occasional extreme case can be readily identified. As n gets larger, the effect becomes weaker.

The GCV function is fairly robust against nonhomogeneity of variances and non-Gaussian errors (Villalobos and Wahba 1987). Andrews (1988) has provided favorable theoretical results when variances are unequal. However, this selection method is likely to give unsatisfactory results when the errors are highly correlated.

The GCV value might be suspect when λ is extremely small because computed values might become indistinguishable from zero. In practice, calculations with $\lambda = 0$ or λ near 0 can cause numerical instabilities resulting in an unsatisfactory solution. Simulation studies have shown that a λ with $\log_{10}(n\lambda) > -8$ is small enough that the final estimate based on this λ almost interpolates the data points. A GCV value based on a $\lambda \leq 10^{-8}$ might not be accurate.

ODS Table Names

PROC TPSPLINE assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets.

These names are listed in the following table. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 89.1 ODS Tables Produced by PROC TPSPLINE

ODS Table Name	Description	Statement	Option
DataSummary	Data summary	PROC	default
FitSummary	Fit parameters and fit summary	PROC	default
FitStatistics	Model fit statistics	PROC	default
GCVFunction	GCV table	MODEL	LOGNLAMBDA, LAMBDA

By referring to the names of such tables, you can use the ODS OUTPUT statement to place one or more of these tables in output data sets.

For example, the following statements create an output data set named FitStats containing the FitStatistics table, an output data set named DataInfo containing the DataSummary table, an output data set named ModelInfo containing the FitSummary table, and an output data set named GCVFunc containing the GCVFunction table.

```
proc tpspline data=Melanoma;
   model Incidences=Year /LOGNLAMBDA=(-4 to 0 by 0.2);
   ods output FitStatistics = FitStats
              DataSummary   = DataInfo
              FitSummary    = ModelInfo
              GCVFunction   = GCVFunc;
run;
```

Examples: TPSPLINE Procedure

Example 89.1: Partial Spline Model Fit

This example analyzes the data set Measure that was introduced in the section "Getting Started: TPSPLINE Procedure" on page 7060. That analysis determined that the final estimated surface can be represented by a quadratic function for one or both of the independent variables. This example illustrates how you can use PROC TPSPLINE to fit a partial spline model. The data set Measure is fit by using the following model:

$$y = \beta_0 + \beta_1 x_1 + \beta x_1^2 + f(x_2)$$

The model has a parametric component (associated with the x_1 variable) and a nonparametric component (associated with the x_2 variable). The following statements fit a partial spline model:

```
data Measure;
   set Measure;
   x1sq = x1*x1;
run;

data pred;
   do x1=-1 to 1 by 0.1;
      do x2=-1 to 1 by 0.1;
         x1sq = x1*x1;
         output;
      end;
   end;
run;

proc tpspline data= measure;
   model y = x1 x1sq (x2);
   score data = pred
         out  = predy;
run;
```

Output 89.1.1 displays the results from these statements.

Output 89.1.1 Output from PROC TPSPLINE

```
                              Raw Data

                        The TPSPLINE Procedure
                      Dependent Variable:  y

                      Summary of Input Data Set

                Number of Non-Missing Observations    50
                Number of Missing Observations         0
                Unique Smoothing Design Points         5


                       Summary of Final Model

                Number of Regression Variables         2
                Number of Smoothing Variables          1
                Order of Derivative in the Penalty     2
                Dimension of Polynomial Space          4


                         Summary Statistics
                        of Final Estimation

                   log10(n*Lambda)            -2.2374
                   Smoothing Penalty         205.3461
                   Residual SS                 8.5821
                   Tr(I-A)                    43.1534
                   Model DF                    6.8466
                   Standard Deviation          0.4460
```

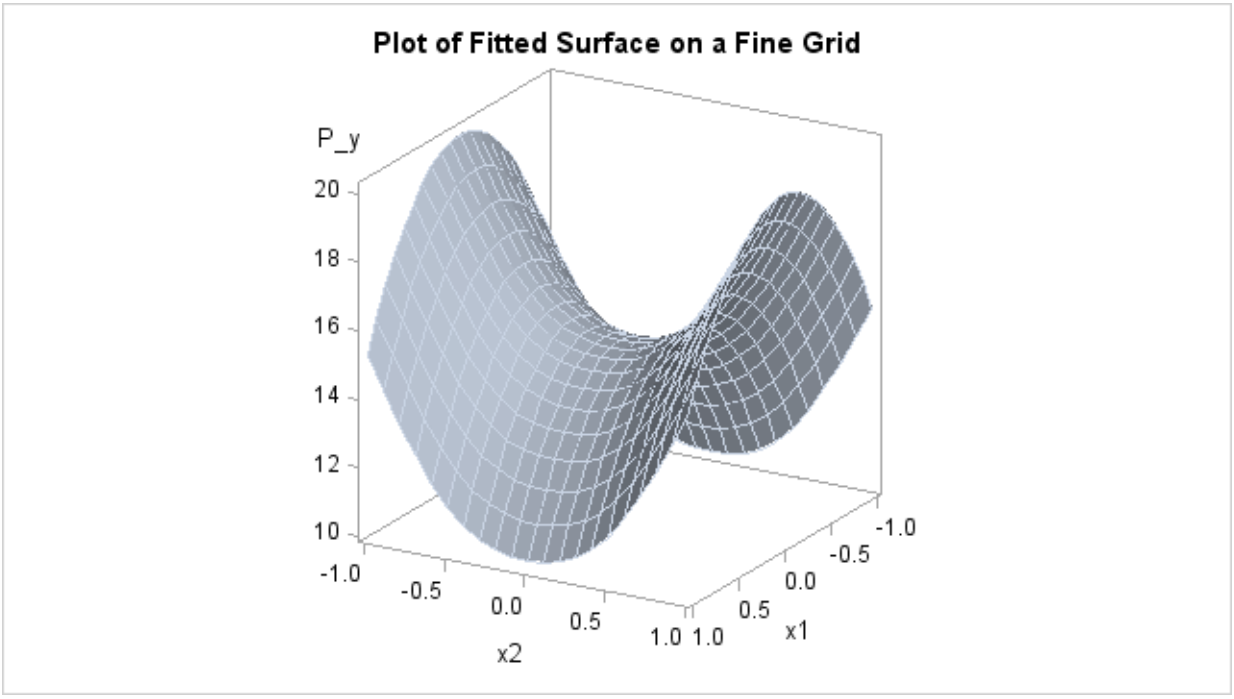
As displayed in Output 89.1.1, there are five unique design points for the smoothing variable x_2 and two regression variables in the model (x_1, x_1^2). The dimension of the polynomial space is sizeof$(\{1, x_1, x_1^2, x_2\}) = 4$. The standard deviation of the estimate is much larger than the one based on the model with both x_1 and x_2 as smoothing variables (0.445954 compared to 0.098421). One of the many possible explanations might be that the number of unique design points of the smoothing variable is too small to warrant an accurate estimate for $f(x_2)$.

The following statements produce a surface plot for the partial spline model by using the surface template defined in page 7066:

```
ods graphics on;

proc sgrender data=predy template=surface;
    dynamic _X='x1' _Y='x2' _Z='P_y' _T='Plot of Fitted Surface on a Fine Grid';
run;

ods graphics off;
```

The surface displayed in Output 89.1.2 is similar to the one estimated by using the full nonparametric model (displayed in Output 89.5).

Output 89.1.2 Plot of PROC TPSPLINE Fit from the Partial Spline Model



Example 89.2: Spline Model with Higher-Order Penalty

This example continues the analysis of the data set Measure to illustrate how you can use PROC TPSPLINE to fit a spline model with a higher-order penalty term. Spline models with high-order penalty terms move low-order polynomial terms into the polynomial space. Hence, there is no penalty for these terms, and they can vary without constraint.

As shown in the previous analyses, the final model for the data set Measure must include quadratic terms for both x_1 and x_2. This example fits the following model:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_1^2 + \beta_3 x_2 + \beta_4 x_2^2 + \beta_5 x_1 x_2 + f(x_1, x_2)$$

The model includes quadratic terms for both variables, although it differs from the usual linear model. The nonparametric term $f(x_1, x_2)$ explains the variation of the data unaccounted for by a simple quadratic surface.

To modify the order of the derivative in the penalty term, specify the M= option. The following statements specify the option M=3 in order to include the quadratic terms in the polynomial space:

```
data Measure;
   set Measure;
   x1sq = x1*x1;
   x2sq = x2*x2;
   x1x2 = x1*x2;
;

proc tpspline data= Measure;
   model y = (x1 x2) / m=3;
   score data = pred
         out = predy;
run;
```

Output 89.2.1 displays the results from these statements.

Output 89.2.1 Output from PROC TPSPLINE with M=3

```
                              Raw Data

                        The TPSPLINE Procedure
                      Dependent Variable:  y

                      Summary of Input Data Set

          Number of Non-Missing Observations     50
          Number of Missing Observations          0
          Unique Smoothing Design Points         25


                       Summary of Final Model

          Number of Regression Variables         0
          Number of Smoothing Variables          2
          Order of Derivative in the Penalty     3
          Dimension of Polynomial Space          6
```

Output 89.2.1 *continued*

Summary Statistics of Final Estimation	
log10(n*Lambda)	-3.7831
Smoothing Penalty	2092.4495
Residual SS	0.2731
Tr(I-A)	29.1716
Model DF	20.8284
Standard Deviation	0.0968

The model contains six terms in the polynomial space (sizeof($\{1, x_1, x_1^2, x_1x_2, x_2, x_2^2\}$) = 6). Compare Output 89.2.1 with Output 89.1.1: the LOGNLAMBDA value and the smoothing penalty differ significantly. Note that, in general, these terms are not directly comparable for different models. The final estimate based on this model is close to the estimate based on the model by using the default, M=2.

In the following statements, the REG procedure fits a quadratic surface model to the data set Measure:

```
proc reg data= Measure;
   model y = x1 x1sq x2 x2sq x1x2;
run;
```

The results are displayed in Output 89.2.2.

Output 89.2.2 Quadratic Surface Model: The REG Procedure

Raw Data

The REG Procedure
Model: MODEL1
Dependent Variable: y

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	443.20502	88.64100	436.33	<.0001
Error	44	8.93874	0.20315		
Corrected Total	49	452.14376			

Root MSE	0.45073	R-Square	0.9802
Dependent Mean	15.08548	Adj R-Sq	0.9780
Coeff Var	2.98781		

Output 89.2.2 *continued*

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	14.90834	0.12519	119.09	<.0001
x1	1	0.01292	0.09015	0.14	0.8867
x1sq	1	-4.85194	0.15237	-31.84	<.0001
x2	1	0.02618	0.09015	0.29	0.7729
x2sq	1	5.20624	0.15237	34.17	<.0001
x1x2	1	-0.04814	0.12748	-0.38	0.7076

The REG procedure produces slightly different results. To fit a similar model with PROC TPSPLINE, you can use a MODEL statement specifying the degrees of freedom with the DF= option. You can also use a large value for the LOGNLAMBDA0= option to force a parametric model fit.

Because there is one degree of freedom for each of the terms intercept, x1, x2, x1sq, x2sq, and x1x2, the DF=6 option is used as follows:

```
proc tpspline data=measure;
   model  y=(x1 x2) /m=3 df=6 lognlambda=(-4 to 1 by 0.2);
   score data = pred
         out = predy;
run;
```

The results are displayed in Output 89.2.3. PROC TPSPLINE displays the list of GCV values for comparison.

Output 89.2.3 Output from PROC TPSPLINE Using M=3 and DF=6

Raw Data

The TPSPLINE Procedure
Dependent Variable: y

Summary of Input Data Set

Number of Non-Missing Observations	50
Number of Missing Observations	0
Unique Smoothing Design Points	25

Summary of Final Model

Number of Regression Variables	0
Number of Smoothing Variables	2
Order of Derivative in the Penalty	3
Dimension of Polynomial Space	6

Output 89.2.3 *continued*

GCV Function

log10(n*Lambda)	GCV
−4.000000	0.016330
−3.800000	0.016051*
−3.600000	0.016363
−3.400000	0.017770
−3.200000	0.021071
−3.000000	0.027496
−2.800000	0.038707
−2.600000	0.056292
−2.400000	0.080613
−2.200000	0.109714
−2.000000	0.139642
−1.800000	0.166338
−1.600000	0.187437
−1.400000	0.202625
−1.200000	0.212871
−1.000000	0.219512
−0.800000	0.223727
−0.600000	0.226377
−0.400000	0.228041
−0.200000	0.229085
0	0.229740
0.200000	0.230153
0.400000	0.230413
0.600000	0.230576
0.800000	0.230680
1.000000	0.230745

Note: * indicates minimum GCV value.

Summary Statistics of Final Estimation

log10(n*Lambda)	2.3830
Smoothing Penalty	0.0000
Residual SS	8.9384
Tr(I−A)	43.9997
Model DF	6.0003
Standard Deviation	0.4507

The final estimate is based on 6.000330 degrees of freedom because there are already 6 degrees of freedom in the polynomial space and the search range for λ is not large enough (in this case, setting DF=6 is equivalent to setting $\lambda = \infty$).

The standard deviation and RSS (Output 89.2.3) are close to the sum of squares for the error term and the root MSE from the linear regression model (Output 89.2.2), respectively.

For this model, the optimal LOGNLAMBDA is around -3.8, which produces a standard deviation estimate of 0.096765 (see Output 89.2.1) and a GCV value of 0.016051, while the model specifying DF=6 results in a LOGNLAMBDA larger than 1 and a GCV value larger than 0.23074. The nonparametric model, based on the GCV, should provide better prediction, but the linear regression model can be more easily interpreted.

Example 89.3: Multiple Minima of the GCV Function

The data in this example represent the deposition of sulfate (SO_4) at 179 sites in the 48 contiguous states of the United States in 1990. Each observation records the latitude and longitude of the site as well as the SO_4 deposition at the site measured in grams per square meter (g/m^2).

You can use PROC TPSPLINE to fit a surface that reflects the general trend and that reveals underlying features of the data, which are shown in the following DATA step:

```
data so4;
input latitude longitude so4 @@;
datalines;
32.45833  87.24222 1.403 34.28778  85.96889 2.103
33.07139 109.86472 0.299 36.07167 112.15500 0.304
31.95056 112.80000 0.263 33.60500  92.09722 1.950

... more lines ...

43.87333 104.19222 0.306 44.91722 110.42028 0.210
45.07611  72.67556 2.646
;
data pred;
   do latitude = 25 to 47 by 1;
      do longitude = 68 to 124 by 1;
         output;
      end;
   end;
run;
```

The preceding statements create the SAS data set so4 and the data set pred in order to make predictions on a regular grid. The following statements fit a surface for SO_4 deposition. The ODS OUTPUT statement creates a data set called GCV to contain the GCV values for LOGNLAMBDA in the range from -6 to 1.

```
proc tpspline data=so4;
   ods output GCVFunction=gcv;
   model so4 = (latitude longitude) /lognlambda=(-6 to 1 by 0.1);
   score data=pred out=prediction1;
run;
```

Partial output from these statements is displayed in Output 89.3.1 and Output 89.3.2.

Output 89.3.1 Partial Output from PROC TPSPLINE for Data Set SO_4

Raw Data

The TPSPLINE Procedure
Dependent Variable: so4

Summary of Input Data Set

Number of Non-Missing Observations	179
Number of Missing Observations	0
Unique Smoothing Design Points	179

Summary of Final Model

Number of Regression Variables	0
Number of Smoothing Variables	2
Order of Derivative in the Penalty	2
Dimension of Polynomial Space	3

Output 89.3.2 Partial Output from PROC TPSPLINE for Data Set SO_4

Summary Statistics
of Final Estimation

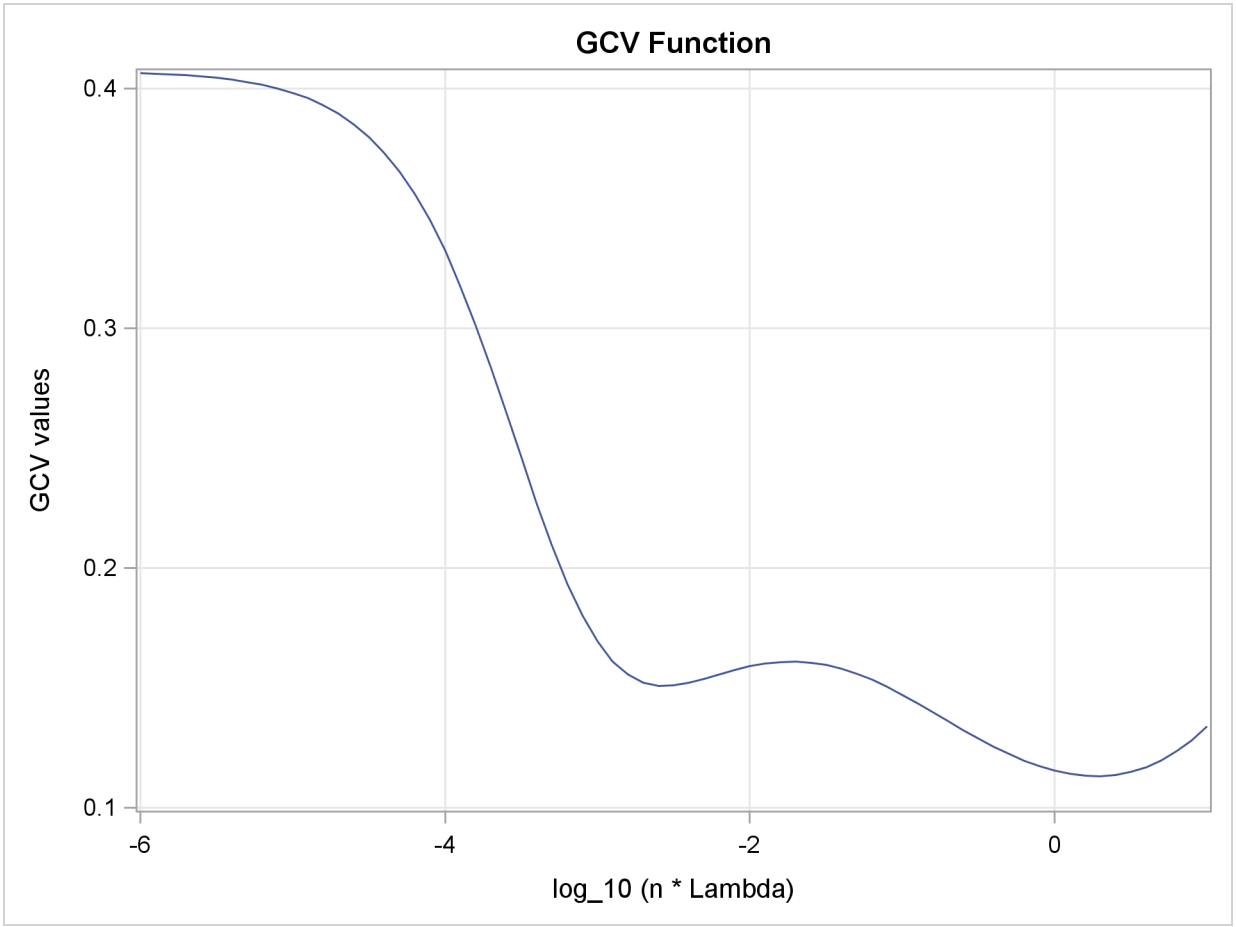
log10(n*Lambda)	0.2770
Smoothing Penalty	2.4588
Residual SS	12.4450
Tr(I-A)	140.2750
Model DF	38.7250
Standard Deviation	0.2979

The following statements use PROC TEMPLATE to produce Output 89.3.3:

```
proc template;
define statgraph plotgcv;
begingraph;
   entrytitle "GCV Function";
   layout overlay/xaxisopts=(griddisplay=on label="log_10 (n * Lambda)")
                  yaxisopts=(griddisplay=on label="GCV values");
                  seriesplot x=lognlambda y=gcv;
   endlayout;
endgraph;
end;
run;

ods graphics on;

proc sgrender data=gcv template=plotgcv;run;
```

Output 89.3.3 displays the plot of the GCV function versus NLAMBDA in $\log_{10}$ scale. The GCV function has two minima. PROC TPSPLINE locates the global minimum at 0.277005. The plot also displays a local minimum located around -2.56. Note that the TPSPLINE procedure might not always find the global minimum, although it did in this case.

Output 89.3.3 GCV Function of SO_4 Data Set



The following analysis specifies the option LOGNLAMBDA0=-2.56. The output is displayed in Output 89.3.4.

```
proc tpspline data=so4;
   model so4 = (latitude longitude) /lognlambda0=-2.56;
   score data=pred out=prediction2;
run;
```

Output 89.3.4 Output from PROC TPSPLINE for Data Set SO_4 with LOGNLAMBDA=−2.56

Raw Data

The TPSPLINE Procedure
Dependent Variable: so4

Summary of Input Data Set

Number of Non-Missing Observations	179
Number of Missing Observations	0
Unique Smoothing Design Points	179

Summary of Final Model

Number of Regression Variables	0
Number of Smoothing Variables	2
Order of Derivative in the Penalty	2
Dimension of Polynomial Space	3

Summary Statistics of Final Estimation

log10(n*Lambda)	-2.5600
Smoothing Penalty	177.2144
Residual SS	0.0438
Tr(I-A)	7.2086
Model DF	171.7914
Standard Deviation	0.0779

The smoothing penalty in Output 89.3.4 is much larger than that displayed in Output 89.3.1. The estimate in Output 89.3.1 uses a large λ value and, therefore, the surface is smoother than the estimate by using LOGNLAMBDA=−2.56 (Output 89.3.4).

The estimate based on LOGNLAMBDA=−2.56 has a larger value for the degrees of freedom, and it has a much smaller standard deviation.

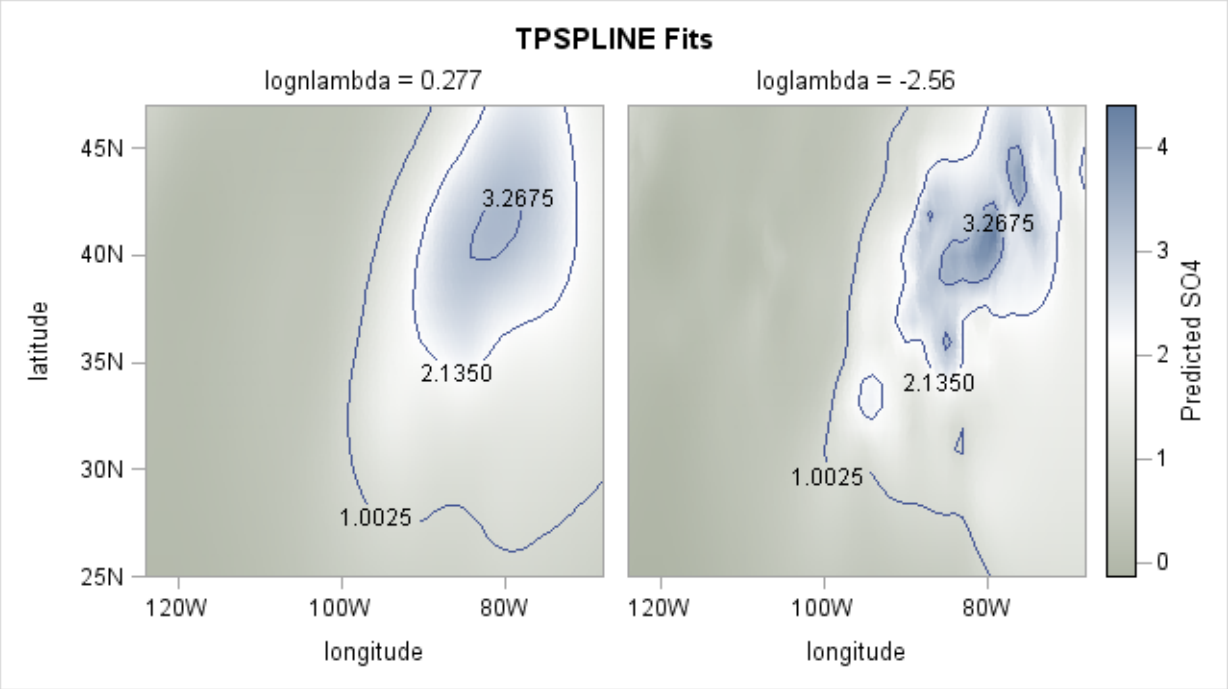
However, a smaller standard deviation in nonparametric regression does not necessarily mean that the estimate is good: a small λ value always produces an estimate closer to the data and, therefore, a smaller standard deviation.

The following statements use PROC TEMPLATE to define a template to produce two contour plots of the estimates by the two minima. A graph of two contour plots is then rendered by the SGRENDER procedure.

```
proc template;
define statgraph two_contour;
begingraph /designheight=360;
   entrytitle "TPSPLINE Fits";
   layout lattice / columns=2 rows=1
                    columndatarange=unionall
                    rowdatarange=unionall
                    columngutter=10;
      rowaxes;
         rowaxis / offsetmin=0 offsetmax=0 display=all
                   linearopts=(thresholdmin=0 thresholdmax=0);
      endrowaxes;

      layout overlay/
         xaxisopts=(offsetmin=0 offsetmax=0
                    linearopts=(thresholdmin=0 thresholdmax=0))
         yaxisopts=(display=none);
         entry "lognlambda = 0.277"/location=outside valign=top
               textattrs=graphlabeltext;
         scatterplot x=longitude y=latitude /primary=true markerattrs=(size=0);
         contourplotparm x=longitude y=latitude z=P_so4/
                         name="cont1" nlevels=5
                         contourtype=gradient;
         contourplotparm x=longitude y=latitude z=P_so4/
                         contourtype=labeledline nlevels=5;
      endlayout;

      layout overlay/
         xaxisopts=(offsetmin=0 offsetmax=0
                    linearopts=(thresholdmin=0 thresholdmax=0))
         yaxisopts=(display=none);
         entry "loglambda = -2.56"/location=outside valign=top
               textattrs=graphlabeltext;
         scatterplot x=longitude y=latitude /primary=true markerattrs=(size=0);
         contourplotparm x=longitude y=latitude z=P_so4_b/
                         name="cont2" nlevels=5
                         contourtype=gradient;
         contourplotparm x=longitude y=latitude z=P_so4_b/
                         contourtype=labeledline nlevels=5;
      endlayout;

      row2headers;
         continuouslegend "cont1" / title="Predicted SO4" pad=(left=10);
      endrow2headers;

   endlayout;
endgraph;
end;
run;
```

Compare the two estimates by examining the contour plots of both estimates (Output 89.3.5).

Output 89.3.5 Contour Plot of TPSPLINE Estimates with Different Lambdas



As the contour plots show, the estimate with LOGNLAMBDA=0.277 might represent the underlying trend, while the estimate with the LOGNLAMBDA=–2.56 is very rough and might be modeling the noise component.

Example 89.4: Large Data Set Application

This example illustrates how you can use the D= option to decrease the computation time needed by the TPSPLINE procedure. Note that, while the D= option can be helpful in decreasing computation time for large data sets, it might produce unexpected results when used with small data sets.

The following statements generate the data set large:

```
data large;
   do x=-5 to 5 by 0.02;
      y=5*sin(3*x)+1*rannor(57391);
      output;
   end;
run;
```

The data set large contains 501 observations with one independent variable x and one dependent variable y. The following statements invoke PROC TPSPLINE to produce a thin-plate smoothing spline estimate and the associated 99% confidence interval. The output statistics are saved in the data set fit1.

```
proc tpspline data=large;
   model y  =(x) /lognlambda=(-5 to -1 by 0.2) alpha=0.01;
   output out=fit1 pred lclm uclm;
run;
```

The results from this MODEL statement are displayed in Output 89.4.1.

Output 89.4.1 Output from PROC TPSPLINE without the D= Option

Raw Data

The TPSPLINE Procedure
Dependent Variable: y

Summary of Input Data Set

Number of Non-Missing Observations	501
Number of Missing Observations	0
Unique Smoothing Design Points	501

Summary of Final Model

Number of Regression Variables	0
Number of Smoothing Variables	1
Order of Derivative in the Penalty	2
Dimension of Polynomial Space	2

GCV Function

log10(n*Lambda)	GCV
-5.000000	1.258653
-4.800000	1.228743
-4.600000	1.205835
-4.400000	1.188371
-4.200000	1.174644
-4.000000	1.163102
-3.800000	1.152627
-3.600000	1.142590
-3.400000	1.132700
-3.200000	1.122789
-3.000000	1.112755
-2.800000	1.102642
-2.600000	1.092769
-2.400000	1.083779
-2.200000	1.076636
-2.000000	1.072763*
-1.800000	1.074636
-1.600000	1.087152
-1.400000	1.120339
-1.200000	1.194023
-1.000000	1.344213

Note: * indicates minimum GCV value.

Output 89.4.1 *continued*

```
                    Summary Statistics
                   of Final Estimation

          log10(n*Lambda)            -1.9483
          Smoothing Penalty        9953.7066
          Residual SS               475.0984
          Tr(I-A)                   471.0861
          Model DF                   29.9139
          Standard Deviation          1.0042
```

The following statements specify an identical model, but with the additional specification of the D= option. The estimates are obtained by treating nearby points as replicates.

```
proc tpspline data=large;
   model y  =(x) /lognlambda=(-5 to -1 by 0.2) d=0.05 alpha=0.01;
   output out=fit2 pred lclm uclm;
run;
```

The output is displayed in Output 89.4.2.

Output 89.4.2 Output from PROC TPSPLINE with the D= Option

```
                          Raw Data

                     The TPSPLINE Procedure
                   Dependent Variable:  y

                   Summary of Input Data Set

          Number of Non-Missing Observations    501
          Number of Missing Observations          0
          Unique Smoothing Design Points        251

                    Summary of Final Model

           Number of Regression Variables        0
           Number of Smoothing Variables         1
           Order of Derivative in the Penalty    2
           Dimension of Polynomial Space         2
```

Output 89.4.2 *continued*

GCV Function

log10(n*Lambda)	GCV
-5.000000	1.306536
-4.800000	1.261692
-4.600000	1.226881
-4.400000	1.200060
-4.200000	1.179284
-4.000000	1.162776
-3.800000	1.149072
-3.600000	1.137120
-3.400000	1.126220
-3.200000	1.115884
-3.000000	1.105766
-2.800000	1.095730
-2.600000	1.085972
-2.400000	1.077066
-2.200000	1.069954
-2.000000	1.066076*
-1.800000	1.067929
-1.600000	1.080419
-1.400000	1.113564
-1.200000	1.187172
-1.000000	1.337252

Note: * indicates minimum GCV value.

Summary Statistics of Final Estimation

log10(n*Lambda)	-1.9477
Smoothing Penalty	9943.5618
Residual SS	472.1424
Tr(I-A)	471.0901
Model DF	29.9099
Standard Deviation	1.0011

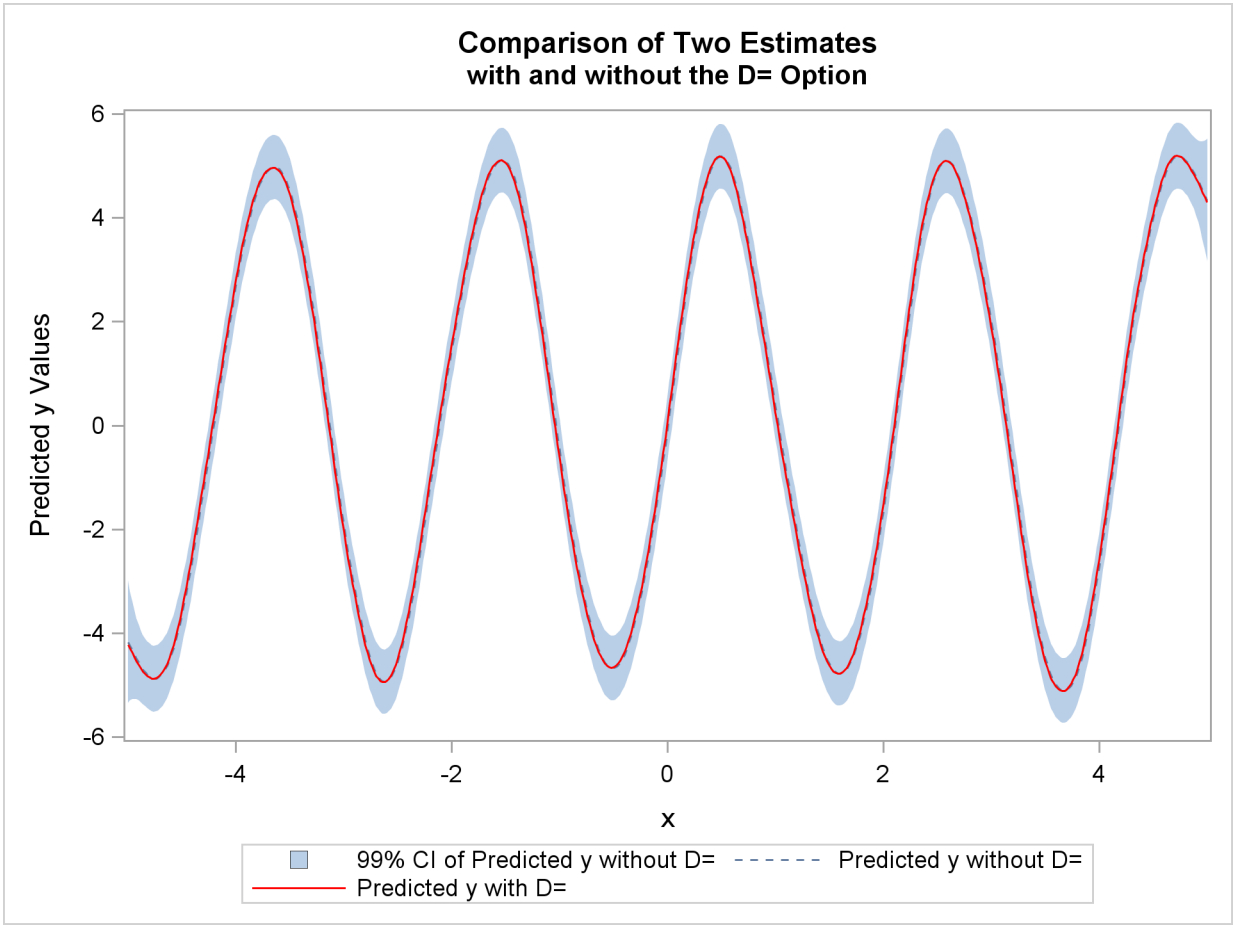
The difference between the two estimates is minimal. However, the CPU time for the second MODEL statement is only about $1/7$ of the CPU time used in the first model fit.

The following statements produce a plot for comparison of the two estimates:

```
data fit2;
   set fit2;
   P1_y     = P_y;
   LCLM1_y  = LCLM_y;
   UCLM1_y  = UCLM_y;
   drop P_y LCLM_y UCLM_y;

proc sort data=fit1;
   by x y;
proc sort data=fit2;
   by x y;

data comp;
   merge fit1 fit2;
      by x y;
   label p1_y   ="Yhat1" p_y="Yhat0"
         lclm_y ="Lower CL"
         uclm_y ="Upper CL";

ods graphics on;

proc sgplot data=comp;
   title "Comparison of Two Estimates";
   title2 "with and without the D= Option";

   yaxis label="Predicted y Values";
   xaxis label="x";

   band x=x lower=lclm_y upper=uclm_y /name="range"
                     legendlabel="99% CI of Predicted y without D=";
   series x=x y=P_y/ name="P_y" legendlabel="Predicted y without D="
                     lineattrs=graphfit(thickness=1px pattern=shortdash);
   series x=x y=P1_y/ name="P1_y" legendlabel="Predicted y with D="
                     lineattrs=graphfit(thickness=1px color=red);
   discretelegend "range" "P_y" "P1_y";
run;

ods graphics off;
```

The estimates from fit1 and fit2 are displayed in Output 89.4.3 with the 99% confidence interval from the fit1 output data set.

Output 89.4.3 Comparison of Two PROC TPSPLINE Fits with and without the D= Option



Example 89.5: Computing a Bootstrap Confidence Interval

This example illustrates how you can construct a bootstrap confidence interval by using the multiple responses feature in PROC TPSPLINE.

Numerous epidemiological observations have indicated that exposure to solar radiation is an important factor in the etiology of melanoma. The following data present age-adjusted melanoma incidences for 37 years from the Connecticut Tumor Registry (Houghton, Flannery, and Viola 1980). The data are analyzed by Ramsay and Silverman (1997).

```
data melanoma;
   input  year incidences @@;
datalines;
1936    0.9   1937   0.8  1938   0.8  1939   1.3
1940    1.4   1941   1.2  1942   1.7  1943   1.8
1944    1.6   1945   1.5  1946   1.5  1947   2.0
1948    2.5   1949   2.7  1950   2.9  1951   2.5
1952    3.1   1953   2.4  1954   2.2  1955   2.9
1956    2.5   1957   2.6  1958   3.2  1959   3.8
1960    4.2   1961   3.9  1962   3.7  1963   3.3
1964    3.7   1965   3.9  1966   4.1  1967   3.8
1968    4.7   1969   4.4  1970   4.8  1971   4.8
1972    4.8
;
```

The variable incidences records the number of melanoma cases per 100,000 people for the years 1936 to 1972. The following model fits the data and requests a 90% Bayesian confidence interval along with the estimate:

```
proc tpspline data=melanoma;
   model incidences = (year) /alpha = 0.1;
   output out = result pred uclm lclm;
run;
```

The output is displayed in Output 89.5.1

Output 89.5.1 Output from PROC TPSPLINE for the MELANOMA Data

```
                    Comparison of Two Estimates
                  with and without the D= Option

                       The TPSPLINE Procedure
                 Dependent Variable:  incidences

                      Summary of Input Data Set

              Number of Non-Missing Observations    37
              Number of Missing Observations         0
              Unique Smoothing Design Points        37


                      Summary of Final Model

              Number of Regression Variables        0
              Number of Smoothing Variables         1
              Order of Derivative in the Penalty    2
              Dimension of Polynomial Space         2
```

Output 89.5.1 *continued*

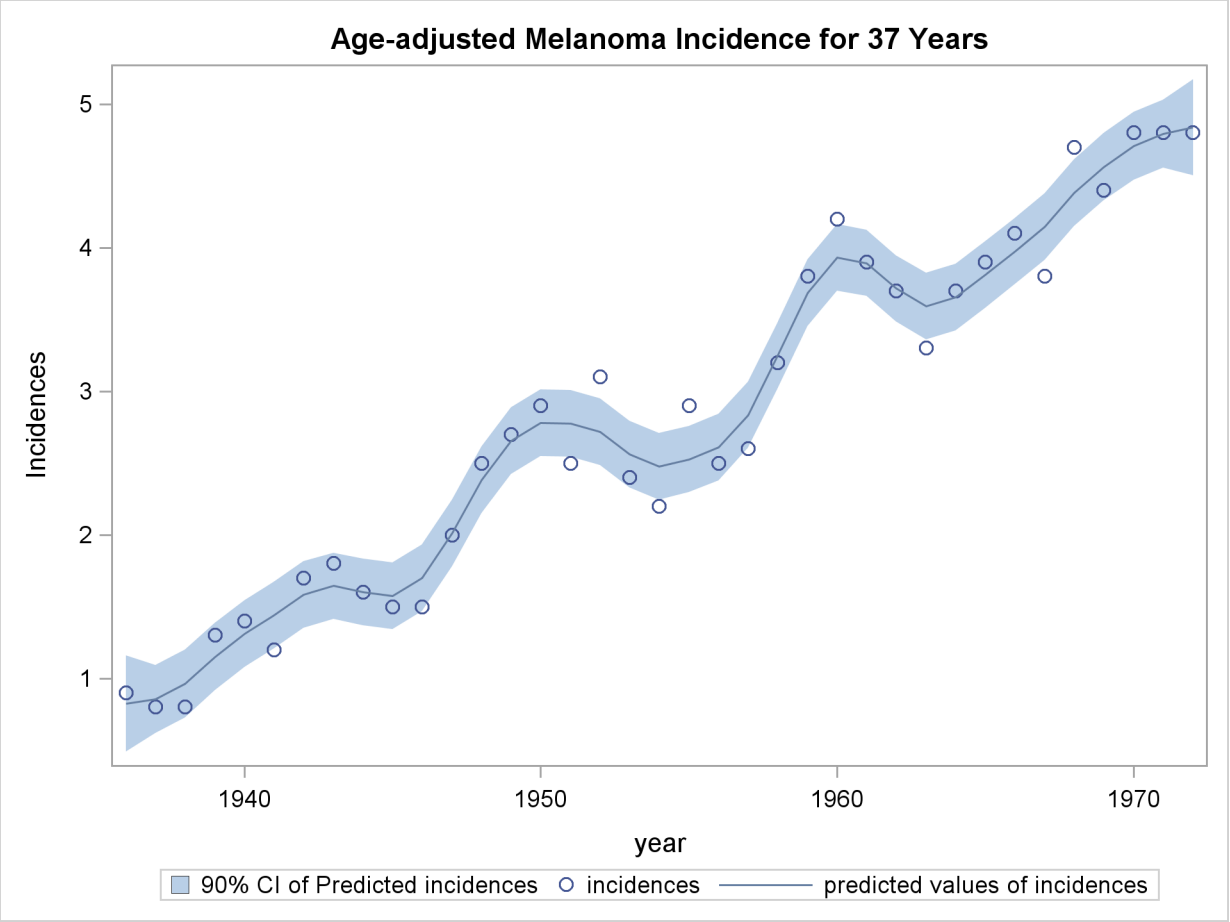
Summary Statistics of Final Estimation	
log10(n*Lambda)	-0.0607
Smoothing Penalty	0.5171
Residual SS	1.2243
Tr(I-A)	22.5852
Model DF	14.4148
Standard Deviation	0.2328

The following statements produce a plot of the estimated curve:

```
ods graphics on;

proc sgplot data=result;
   title "Age-adjusted Melanoma Incidence for 37 Years";

   xaxis label="year";
   yaxis label="Incidences";

   band x=year lower=lclm_incidences upper=uclm_incidences/name="range"
               legendlabel="90% CI of Predicted incidences";
   scatter x=year y=incidences/name="obs" legendlabel="incidences";
   series x=year y=p_incidences/name="pred"
               legendlabel="predicted values of incidences"
               lineattrs=graphfit(thickness=1px);

   discretelegend "range" "obs" "pred";
run;
```

The estimated curve is displayed with 90% confidence interval bands in Output 89.5.2. The number of melanoma incidences exhibits a periodic pattern and increases over the years. The periodic pattern is related to sunspot activity and the accompanying fluctuations in solar radiation.

Output 89.5.2 PROC TPSPLINE Estimate and 90% Confidence Interval of Data Set MELANOMA



Wang and Wahba (1995) compare several bootstrap confidence intervals to Bayesian confidence intervals for smoothing splines. Both bootstrap and Bayesian confidence intervals are across-the-curve intervals, not pointwise intervals. They concluded that bootstrap confidence intervals work as well as Bayesian intervals concerning average coverage probability. Additionally, bootstrap confidence intervals appear to be better for small sample sizes. Based on their simulation, the "percentile-t interval" bootstrap interval performs better than the other types of bootstrap intervals.

Suppose that $\hat{f}_{\hat{\lambda}}$ and $\hat{\sigma}$ are the estimates of f and σ from the data. Assume that $\hat{f}_{\hat{\lambda}}$ is the "true" f, and generate the bootstrap sample as follows:

$$y_i^* = \hat{f}_{\hat{\lambda}}(\mathbf{x}_i) + \epsilon_i^*, \quad i = 1, \cdots, n$$

where $\boldsymbol{\epsilon}^* = (\epsilon_1^*, \cdots, \epsilon_n^*)^T \sim N(0, \hat{\sigma}\mathbf{I})$. Denote $f_{\hat{\lambda}}^*(\mathbf{x}_i)$ as the random variable of the bootstrap estimate at $\mathbf{x}_i$. Repeat this process K times, so that at each point $\mathbf{x}_i$, you have K bootstrap estimates $\hat{f}_{\hat{\lambda}}(\mathbf{x}_i)$ or K realizations of $f_{\hat{\lambda}}^*(\mathbf{x}_i)$. For each fixed $\mathbf{x}_i$, consider the following statistic D_i^*, which is similar to the Student's t statistic:

$$D_i^* = \left(f_{\hat{\lambda}}^*(\mathbf{x}_i) - \hat{f}_{\hat{\lambda}}(\mathbf{x}_i)\right) / \hat{\sigma}_i{}^*$$

where $\hat{\sigma}_i{}^*$ is the estimate of $\hat{\sigma}$ based on the ith bootstrap sample.

Suppose $\chi_{\alpha/2}$ and $\chi_{1-\alpha/2}$ are the lower and upper $\alpha/2$ points, respectively, of the empirical distribution of D_i^*. The $(1-\alpha)100\%$ bootstrap confidence interval is defined as

$$\left(\hat{f}_{\hat{\lambda}}(\mathbf{x}_i) - \chi_{1-\alpha/2}\hat{\sigma},\ \ \hat{f}_{\hat{\lambda}}(\mathbf{x}_i) - \chi_{\alpha/2}\hat{\sigma}\right)$$

Bootstrap confidence intervals are easy to interpret and can be used with any distribution. However, because they require K model fits, their construction is computationally intensive.

The multiple dependent variables feature in PROC TPSPLINE enables you to fit multiple models with the same independent variables. The procedure calculates the matrix decomposition part of the calculations only once, regardless of the number of dependent variables in the model. These calculations are responsible for most of the computing time used by the TPSPLINE procedure. This feature is particularly useful when you need to generate a bootstrap confidence interval.

To construct a bootstrap confidence interval, perform the following tasks:

- Fit the data by using PROC TPSPLINE and obtain estimates $\hat{f}_{\hat{\lambda}}(\mathbf{x}_i)$ and $\hat{\sigma}$.
- Generate K bootstrap samples based on $\hat{f}_{\hat{\lambda}}(\mathbf{x}_i)$ and $\hat{\sigma}$.
- Fit the K bootstrap samples with the TPSPLINE procedure to obtain estimates of $\hat{f}^*_{\hat{\lambda}}(\mathbf{x}_i)$ and $\hat{\sigma}_i^*$.
- Compute D_i^* and the values $\chi_{\alpha/2}$ and $\chi_{1-\alpha/2}$.

The following statements illustrate this process:

```
proc tpspline data=melanoma;
   model incidences = (year) /alpha = 0.05;
   output out = result pred uclm lclm;
run;
```

The output from the initial PROC TPSPLINE analysis is displayed in Output 89.5.3. The data set result contains the predicted values and confidence limits from the analysis.

Output 89.5.3 Output from PROC TPSPLINE for the MELANOMA Data

```
          Age-adjusted Melanoma Incidence for 37 Years

                The TPSPLINE Procedure
          Dependent Variable:  incidences

               Summary of Input Data Set

  Number of Non-Missing Observations    37
  Number of Missing Observations         0
  Unique Smoothing Design Points        37
```

Output 89.5.3 *continued*

Summary of Final Model

Number of Regression Variables	0
Number of Smoothing Variables	1
Order of Derivative in the Penalty	2
Dimension of Polynomial Space	2

Summary Statistics of Final Estimation

log10(n*Lambda)	-0.0607
Smoothing Penalty	0.5171
Residual SS	1.2243
Tr(I-A)	22.5852
Model DF	14.4148
Standard Deviation	0.2328

The following statements illustrate how you can obtain a bootstrap confidence interval for the Melanoma data set. The following statements create the data set bootstrap. The observations are created with information from the preceding PROC TPSPLINE execution; as displayed in Output 89.5.3, $\hat{\sigma} = 0.232823$. The values of $\hat{f}_{\hat{\lambda}}(\mathbf{x}_i)$ are stored in the data set result in the variable P_incidence.

```
data bootstrap;
   set result;
   array y{1070} y1-y1070;
   do i=1 to 1070;
      y{i} = p_incidences + 0.232823*rannor(123456789);
   end;
   keep y1-y1070 p_incidences year;
run;

proc tpspline data=bootstrap;
   ods output FitStatistics=FitResult;
   id p_incidences;
   model y1-y1070 = (year);
   output out=result2;
run;
```

The DATA step generates 1,070 bootstrap samples based on the previous estimate from PROC TPSPLINE. For this data set, some of the bootstrap samples result in λs (selected by the GCV function) that cause problematic behavior. Thus, an additional 70 bootstrap samples are generated.

The ODS listing destination is closed before PROC TPSPLINE is invoked. The model fits all the y1...y1070 variables as dependent variables, and the models are fit for all bootstrap samples simultaneously. The output data set result2 contains the variables year, y1...y1070, p_y1...p_y1070, and p_incidences.

The ODS OUTPUT statement writes the FitStatistics table to the data set FitResult. The data set FitResult contains the two variables Parameter and Value. The FitResult data set is used in subsequent calculations for D_i^*.

In the data set FitResult, there are 63 estimates with a standard deviation of zero, suggesting that the estimates provide perfect fits of the data and are caused by $\hat{\lambda}$s that are approximately equal to zero. For small sample sizes, there is a positive probability that the λ chosen by the GCV function will be zero (Wang and Wahba 1995).

In the following steps, these cases are removed from the bootstrap samples as "bad" samples: they represent failure of the GCV function.

The following SAS statements manipulate the data set FitResult, retaining the standard deviations for all bootstrap samples and merging FitResult with the data set result2, which contains the estimates for bootstrap samples. In the final data set boot, the D_i^* statistics are calculated.

```
data FitResult;
   set FitResult;
   if Parameter="Standard Deviation";
   keep Value;
run;

proc transpose data=FitResult out=sd prefix=sd;

data result2;
   if _N_ = 1 then set sd;
   set result2;

data boot;
   set result2;
   array y{1070}  p_y1-p_y1070;
   array sd{1070} sd1-sd1070;
   do i=1 to 1070;
      if sd{i} > 0 then do;
         d = (y{i} - P_incidences)/sd{i};
         obs = _N_;
         output;
      end;
   end;
   keep d obs P_incidences year;
run;
```

The following SAS statements retain the first 1,000 bootstrap samples and calculate the values $\chi_{\alpha/2}$ and $\chi_{1-\alpha/2}$ with $\alpha = 0.1$.

```
proc sort data=boot;
   by obs;
run;

data boot;
   set boot;
      by obs;
   retain n;

   if first.obs then n=1;
      else n=n+1;
   if n > 1000 then delete;
run;

proc sort data=boot;
   by obs d;
run;

data chi1 chi2 ;
   set boot;
   if (_N_ = (obs-1)*1000+50)  then output chi1;
   if (_N_ = (obs-1)*1000+950) then output chi2;
run;

proc sort data=result;
   by year;
run;

proc sort data=chi1;
   by year;
run;

proc sort data=chi2;
   by year;
run;

data result;
   merge result
      chi1(rename=(d=chi05))
      chi2(rename=(d=chi95));
   keep year incidences P_incidences lower upper
        LCLM_incidences UCLM_incidences;

   lower = -chi95*0.232823 + P_incidences;
   upper = -chi05*0.232823 + P_incidences;

   label  lower="Lower 90% CL (Bootstrap)"
          upper="Upper 90% CL (Bootstrap)"
          lclm_incidences="Lower 90% CL (Bayesian)"
          uclm_incidences="Upper 90% CL (Bayesian)";
run;
```

The data set result contains the variables year and incidences, the PROC TPSPLINE estimate P_incidences, and the 90% Bayesian and 90% bootstrap confidence intervals.

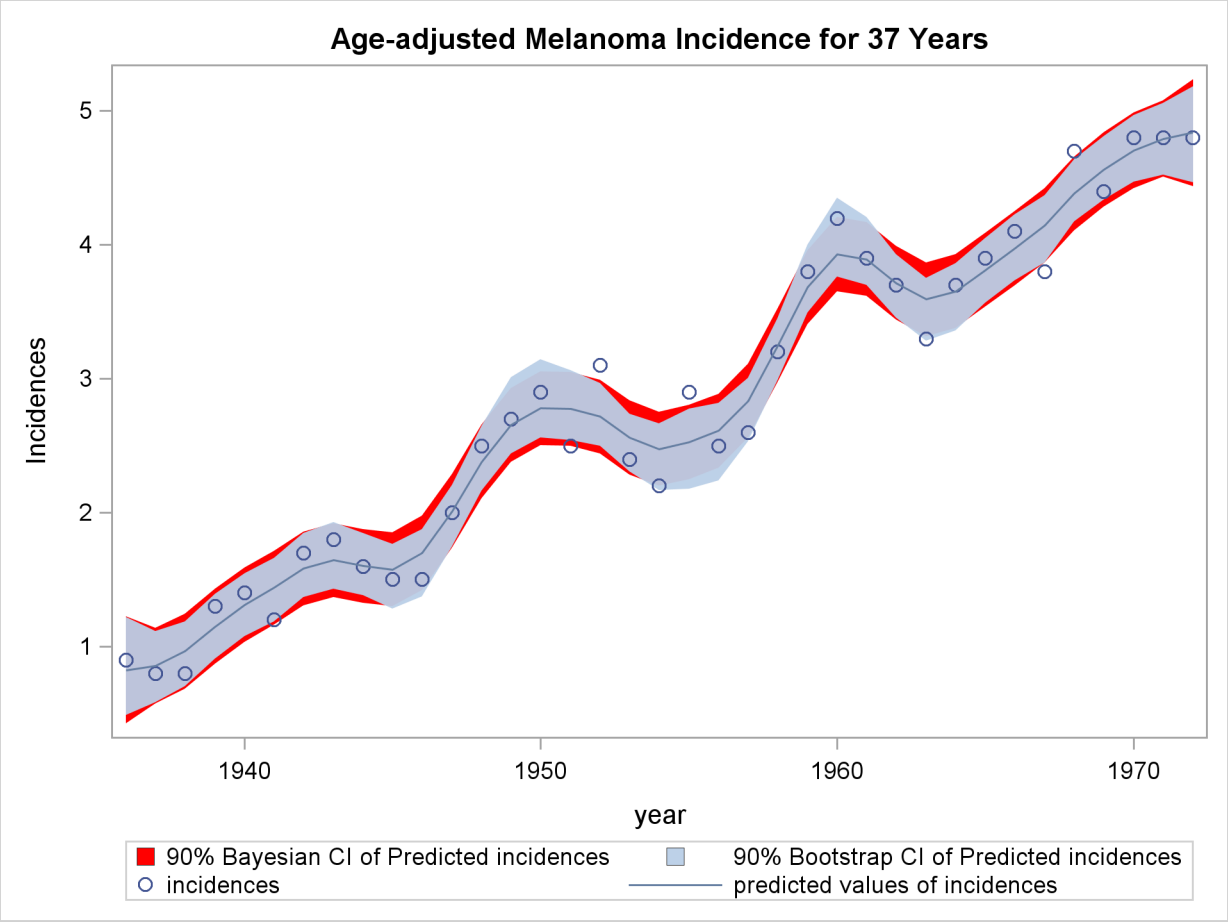
The following statements produce Output 89.5.4:

```
proc sgplot data=result;
   title "Age-adjusted Melanoma Incidence for 37 Years";

   xaxis label="year";
   yaxis label="Incidences";

   band x=year lower=lclm_incidences upper=uclm_incidences/name="bayesian"
               legendlabel="90% Bayesian CI of Predicted incidences"
               fillattrs=(color=red);
   band x=year lower=lower upper=upper/name="bootstrap"
               legendlabel="90% Bootstrap CI of Predicted incidences"
               transparency=0.05;
   scatter x=year y=incidences/name="obs" legendlabel="incidences";
   series x=year y=p_incidences/name="pred"
               legendlabel="predicted values of incidences"
               lineattrs=graphfit(thickness=1px);

   discretelegend "bayesian" "bootstrap" "obs" "pred";
run;

ods graphics off;
```

Output 89.5.4 displays the plot of the variable incidences, the predicted values, and the Bayesian and bootstrap confidence intervals.

The plot shows that the bootstrap confidence interval is similar to the Bayesian confidence interval. However, the Bayesian confidence interval is symmetric around the estimates, while the bootstrap confidence interval is not.

Output 89.5.4 Comparison of Bayesian and Bootstrap Confidence Interval for Data Set MELANOMA



References

Andrews, D (1988), *Asymptotic Optimality of GC_L, Cross-Validation, and GCV in Regression with Heteroscedastic Errors*, manuscript, Cowles Foundation, Yale University, New Haven, CT.

Bates, D., Lindstrom, M., Wahba, G. and Yandell, B. (1987), "GCVPACK-Routines for Generalized Cross Validation," *Comm. Statist. B-Simulation Comput.*, 16, 263–297.

Dongarra, J., Bunch, J., Moler, C. and Steward, G. (1979), *Linpack Users' Guide*, Philadelphia: Society for Industrial and Applied Mathematics.

Duchon, J. (1976), "Fonctions-Spline et Esperances Conditionnelles de Champs Gaussiens," *Ann. Sci. Univ. Clermont Ferrand II Math.*, 14, 19–27.

Duchon, J. (1977), "Splines Minimizing Rotation-Invariant Semi-Norms in Sobolev Spaces," in *Constructive Theory of Functions of Several Variables,* eds. W. Schempp and K. Zeller, 85–100.

Hall, P. and Titterington, D. (1987), "Common Structure of Techniques for Choosing Smoothing Parameters in Regression Problems," *J. Roy. Statist. Soc. Ser. B*, 49, 184–198.

Houghton, A. N., Flannery, J. and Viola, M. V. (1980), "Malignant Melanoma in Connecticut and Denmark," *International Journal of Cancer*, 25, 95–104.

Hutchinson, M. and Bischof, R. (1983), "A New Method for Estimating the Spatial Distribution of Mean Seasonal and Annual Rainfall Applied to the Hunter Valley," *New South Wales, Aust. Met. Mag.*, 31, 179–184.

Meinguet, J. (1979), "Multivariate Interpolation at Arbitrary Points Made Simple," *J. Appl. Math. Phys. (ZAMP)*, 5, 439–468.

Nychka, D. (1986a), "The Average Posterior Variance of a Smoothing Spline and a Consistent Estimate of the Mean Square Error," Tech. Report 168, The Institute of Statistics, North Carolina State University, Raleigh, NC.

Nychka, D. (1986b), "A Frequency Interpretation of Bayesian "Confidence" Interval for Smoothing Splines," Tech. Report 169, The Institute of Statistics, North Carolina State University, Raleigh, NC.

Nychka, D. (1988), "Bayesian Confidence Intervals for Smoothing Splines," *J. Amer. Statist. Assoc.*, 83, 1134–1143.

O'Sullivan, F. and Wong, T. (1987), "Determining a Function Diffusion Coefficient in the Heat Equation," Tech. Report 98, Department of Statistics, University of California, Berkeley.

Ramsay, J. and Silverman, B (1997), *Functional Data Analysis*, New York: Springer-Verlag.

Seaman, R. and Hutchinson, M. (1985), "Comparative Real Data Tests of Some Objective Analysis Methods by Withholding," *Aust. Met. Mag.*, 33, 37–46.

Villalobos, M. and Wahba, G. (1987), "Inequality Constrained Multivariate Smoothing Splines with Application to the Estimation of Posterior Probabilities," *J. Amer. Statist. Assoc.*, 82 239–248.

Wahba, G. (1983), "Bayesian 'Confidence Intervals' for the Cross Validated Smoothing Spline," *J. Roy. Statist. Soc. Ser. B*, 45, 133–150.

Wahba, G., (1990), *Spline Models for Observational Data,* Philadelphia: Society for Industrial and Applied Mathematics.

Wahba, G. and Wendelberger, J. (1980), "Some New Mathematical Methods for Variational Objective Analysis Using Splines and Cross Validation," *Monthly Weather Rev.*, 108, 1122–1145.

Wang, Y. and Wahba, G. (1995), "Bootstrap Confidence Intervals for Smoothing Splines and Their Comparison to Bayesian Confidence Intervals," *J. Statistical Computation and Simulation*, 51, 263–279.

Chapter 90

The TRANSREG Procedure

Contents

Overview: TRANSREG Procedure **7106**
Getting Started: TRANSREG Procedure **7109**
Fitting a Curve through a Scatter Plot 7109
Main-Effects ANOVA 7126
Syntax: TRANSREG Procedure **7129**
PROC TRANSREG Statement 7130
BY Statement 7137
FREQ Statement 7138
ID Statement 7138
MODEL Statement 7138
OUTPUT Statement 7168
WEIGHT Statement 7179
Details: TRANSREG Procedure **7179**
Model Statement Usage 7179
Box-Cox Transformations 7181
Using Splines and Knots 7192
Scoring Spline Variables 7204
Linear and Nonlinear Regression Functions 7209
Simultaneously Fitting Two Regression Functions 7213
Penalized B-Splines 7219
Smoothing Splines 7222
Smoothing Splines Changes and Enhancements 7227
Iteration History Changes and Enhancements 7230
ANOVA Codings 7230
Missing Values 7249
Missing Values, UNTIE, and Hypothesis Tests 7250
Controlling the Number of Iterations 7251
Using the REITERATE Algorithm Option 7252
Avoiding Constant Transformations 7253
Constant Variables 7253
Character OPSCORE Variables 7253
Convergence and Degeneracies 7254
Implicit and Explicit Intercepts 7254
Passive Observations 7255

Point Models 7255
Redundancy Analysis 7256
Optimal Scaling 7259
OPSCORE, MONOTONE, UNTIE, and LINEAR Transformations 7260
SPLINE and MSPLINE Transformations 7261
Specifying the Number of Knots 7262
SPLINE, BSPLINE, and PSPLINE Comparisons 7264
Hypothesis Tests 7264
Output Data Set 7267
OUTTEST= Output Data Set 7275
Computational Resources 7276
Unbalanced ANOVA without CLASS Variables 7277
Hypothesis Tests for Simple Univariate Models 7278
Hypothesis Tests with Monotonicity Constraints 7284
Hypothesis Tests with Dependent Variable Transformations 7286
Hypothesis Tests with One-Way ANOVA 7289
Using the DESIGN Output Option 7293
Discrete Choice Experiments: DESIGN, NORESTORE, NOZERO 7297
Centering 7298
Displayed Output 7299
ODS Table Names 7299
ODS Graphics 7301
Examples: TRANSREG Procedure **7307**
Example 90.1: Transformation Regression of Exhaust Emissions Data . . . 7307
Example 90.2: Box-Cox Transformations 7316
Example 90.3: Penalized B-Spline 7323
Example 90.4: Nonmetric Conjoint Analysis of Tire Data 7329
Example 90.5: Metric Conjoint Analysis of Tire Data 7333
Example 90.6: Preference Mapping of Automobile Data 7345
References **7351**

Overview: TRANSREG Procedure

The TRANSREG (transformation regression) procedure fits linear models, optionally with smooth, spline, Box-Cox, and other nonlinear transformations of the variables. You can use PROC TRANSREG to fit a curve through a scatter plot or fit multiple curves, one for each level of a classification variable. You can also constrain the functions to be parallel or monotone or have the same intercept. PROC TRANSREG can be used to code experimental designs and classification variables prior to their use in other analyses.

The TRANSREG procedure fits many types of linear models, including the following:

- ordinary regression and ANOVA
- metric and nonmetric conjoint analysis (Green and Wind 1975; de Leeuw, Young, and Takane 1976)
- linear models with Box-Cox (1964) transformations of the dependent variables
- regression with a smooth (Reinsch 1967), spline (de Boor 1978; van Rijckevorsel 1982), monotone spline (Winsberg and Ramsay 1980), or penalized B-spline (Eilers and Marx 1996) fit function
- metric and nonmetric vector and ideal point preference mapping (Carroll 1972)
- simple, multiple, and multivariate regression with variable transformations (Young, de Leeuw, and Takane 1976; Winsberg and Ramsay 1980; Breiman and Friedman 1985)
- redundancy analysis (Stewart and Love 1968) with variable transformations (Israels 1984)
- canonical correlation analysis with variable transformations (van der Burg and de Leeuw 1983)
- response surface regression (Meyers 1976; Khuri and Cornell 1987) with variable transformations

The data set can contain variables measured on nominal, ordinal, interval, and ratio scales (Siegel 1956). You can specify any mix of these variable types for the dependent and independent variables. PROC TRANSREG can do the following:

- transform nominal variables by scoring the categories to minimize squared error (Fisher 1938), or treat nominal variables as classification variables
- transform ordinal variables by monotonically scoring the ordered categories so that order is weakly preserved (adjacent categories can be merged) and squared error is minimized. Ties can be optimally untied or left tied (Kruskal 1964). Ordinal variables can also be transformed to ranks.
- transform interval and ratio scale of measurement variables linearly or nonlinearly with spline (de Boor 1978; van Rijckevorsel 1982), monotone spline (Winsberg and Ramsay 1980), penalized B-spline (Eilers and Marx 1996), smooth (Reinsch 1967), or Box-Cox (Box and Cox 1964) transformations. In addition, logarithmic, exponential, power, logit, and inverse trigonometric sine transformations are available.

Transformations produced by the PROC TRANSREG multiple regression algorithm, requesting spline transformations, are often similar to transformations produced by the ACE smooth regression method of Breiman and Friedman (1985). However, ACE does not explicitly optimize a loss function (de Leeuw 1986), while PROC TRANSREG explicitly minimizes a squared-error criterion.

PROC TRANSREG extends the ordinary general linear model by providing optimal variable transformations that are iteratively derived. PROC TRANSREG iterates until convergence, alternating two major steps: finding least-squares estimates of the model parameters given the current scoring of the data, and finding least-squares estimates of the scoring parameters given the current set of model parameters. This is called the method of alternating least squares (Young 1981).

For more background on alternating least-squares optimal scaling methods and transformation regression methods, see Young, de Leeuw, and Takane (1976), Winsberg and Ramsay (1980), Young (1981), Gifi (1990), Schiffman, Reynolds, and Young (1981), van der Burg and de Leeuw (1983), Israels (1984), Breiman and Friedman (1985), and Hastie and Tibshirani (1986). (These are just a few of the many relevant sources.)

There have been a number of enhancements to PROC TRANSREG with this release, and a few changes. The most notable enhancement is the addition of graphical displays produced through ODS Graphics. Now, fit, transformation, residual, and other plots are created when you enable ODS Graphics. See the sections "Fitting a Curve through a Scatter Plot" on page 7109, "Box-Cox Transformations" on page 7181, "Using Splines and Knots" on page 7192, "Linear and Nonlinear Regression Functions" on page 7209, "Simultaneously Fitting Two Regression Functions" on page 7213, "Smoothing Splines" on page 7222, and "ODS Graphics" on page 7301 for more information. See Example 90.1, Example 90.2, Example 90.3, and Example 90.6 for examples. The PBSPLINE transformation (penalized B-spline) is new with this release. It can be used to fit curves through a scatter plot with an automatic selection of the smoothing parameter. See the sections "Fitting a Curve through a Scatter Plot" on page 7109 and "Penalized B-Splines" on page 7219, and see Example 90.3 for an example. How the results of the SMOOTH transformation are used in PROC TRANSREG has changed with this release. In particular, some aspects of the syntax along with the coefficients, degrees of freedom, and predicted values have changed. See the section "Smoothing Splines Changes and Enhancements" on page 7227 for more information about the changes, and see the NSR algorithm option for a way to restore the old functionality. Also, the iteration history table is not printed in certain models, such as those with smoothing splines and penalized B-splines, where it is known that no iterations are necessary. See the section "Iteration History Changes and Enhancements" on page 7230 for more information.

New options include the following:
ORTHOGONAL *t-option* – requests an orthogonal-contrast coding
STANDORTH *t-option* – requests a standardized-orthogonal coding
PLOTS= PROC statement option – ODS Graphics selection
NSR *a-option* – no smoothing spline model restrictions
TSUFFIX= *a-option* – shortens transformed variable labels
PBSPLINE *transform* – penalized B-splines

New options for penalized B-splines include the following:
AIC *t-option* – Akaike's information criterion
AICC *t-option* – corrected AIC
CV *t-option* – cross validation criterion
GCV *t-option* – generalized cross validation criterion
RANGE *t-option* – LAMBDA= specifies a range, not a list
SBC *t-option* – Schwarz's Bayesian criterion

Changed options include the following:
LAMBDA= *t-option* – smoothing parameter list or range, now used with PBSPLINE
EVENLY= *t-option* – evenly spaced interior knots, now creates evenly spaced exterior knots as well
SMOOTH *transform* – new *df* calculations, printed output
SSPLINE *transform* – new *df* calculations, just like SMOOTH

Getting Started: TRANSREG Procedure

This section provides several examples that illustrate a few of the more basic features of PROC TRANSREG.

Fitting a Curve through a Scatter Plot

PROC TRANSREG can fit curves through data and detect nonlinear relationships among variables. This example uses a subset of the data from an experiment in which nitrogen oxide emissions from a single cylinder engine are measured for various combinations of fuel and equivalence ratio (Brinkman 1981). The following statements create the SAS data set:

```
title 'Gasoline and Emissions Data';

data Gas;
   input Fuel :$8. EqRatio NOx @@;
   label EqRatio = 'Equivalence Ratio'
         NOx     = 'Nitrogen Oxide Emissions';
   datalines;
Ethanol  0.907 3.741 Ethanol  0.761 2.295 Ethanol  1.108 1.498
Ethanol  1.016 2.881 Ethanol  1.189 0.760 Ethanol  1.001 3.120
Ethanol  1.231 0.638 Ethanol  1.123 1.170 Ethanol  1.042 2.358

... more lines ...

Ethanol  0.655 1.900
;
```

The next step fits a spline or curve through the data and displays the regression results. For information about splines and knots, see the sections "Smoothing Splines" on page 7222, "Linear and Nonlinear Regression Functions" on page 7209, "Simultaneously Fitting Two Regression Functions" on page 7213, and "Using Splines and Knots" on page 7192, as well as Example 90.1. The following statements produce Figure 90.1:

```
ods graphics on;

* Request a Spline Transformation of Equivalence Ratio;
proc transreg data=Gas solve ss2 plots=(transformation obp residuals);
   model identity(nox) = spline(EqRatio / nknots=4);
run;
```

The SOLVE algorithm option, or *a-option*, requests a direct solution for both the transformation and the parameter estimates. For many models, PROC TRANSREG with the SOLVE *a-option* can produce exact results without iteration. The SS2 (Type II sums of squares) *a-option* requests regression and ANOVA results. The PLOTS= option requests plots of the variable transformations, a plot of the observed values by the predicted values, and a plot of the residuals. The dependent variable NOx was specified with an IDENTITY transformation, which means that it will not be transformed, just as in ordinary regression. The independent variable EqRatio, in contrast, is transformed by using a cubic spline with four knots. The NKNOTS= option is known as a transformation option, or *t-option*. Graphical results are enabled with the two ODS statements that precede the PROC TRANSREG step, and the results are terminated with the final two ODS statements. The results are shown in Figure 90.1 through Figure 90.5.

Figure 90.1 Iteration, ANOVA, and Regression Results

```
                    Gasoline and Emissions Data

                       The TRANSREG Procedure

                  Dependent Variable Identity(NOx)
                      Nitrogen Oxide Emissions


              Number of Observations Read           112
              Number of Observations Used           110

      TRANSREG MORALS Algorithm Iteration History for Identity(NOx)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        0    1.04965    3.46121     0.00917
        1    0.00000    0.00000     0.82429      0.81512    Converged

Algorithm converged.
```

Figure 90.1 *continued*

The TRANSREG Procedure Hypothesis Tests for Identity(NOx)
Nitrogen Oxide Emissions

Univariate ANOVA Table Based on the Usual Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	7	180.0951	25.72788	68.36	<.0001
Error	102	38.3891	0.37636		
Corrected Total	109	218.4842			

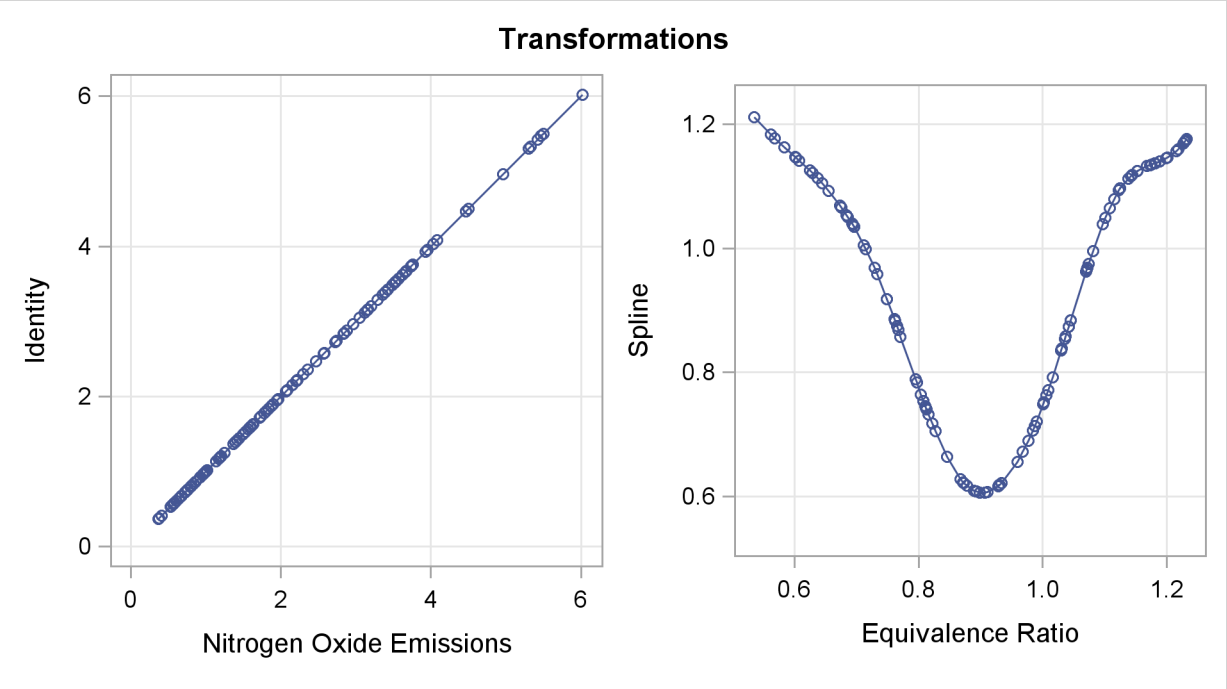
Root MSE	0.61348	R-Square	0.8243
Dependent Mean	2.25022	Adj R-Sq	0.8122
Coeff Var	27.26334		

Univariate Regression Table Based on the Usual Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Pr > F	Label
Intercept	1	8.3165407	324.065	324.065	861.04	<.0001	Intercept
Spline(EqRatio)	7	-6.5740158	180.095	25.728	68.36	<.0001	Equivalence Ratio

PROC TRANSREG increases the squared multiple correlation from the original value of 0.00917 to 0.82429. Iteration 0 shows the fit before the data are transformed, and iteration 1 shows the fit after the transformation, which was directly solved for in the initial iteration. The change values for iteration 0 show the change from the original EqRatio variable to the transformed EqRatio variable. For this model, no improvement on the initial solution is possible, so in iteration 1, all change values are zero. The ANOVA and regression results show that you are fitting a model with 7 model parameters, 4 knots plus a degree 3 or cubic spline. The overall model fit is identical to the test for the spline transformation, since there is only one term in the model besides the intercept, and the results are significant at the 0.0001 level. The transformations are shown next in Figure 90.2.

Figure 90.2 Transformations



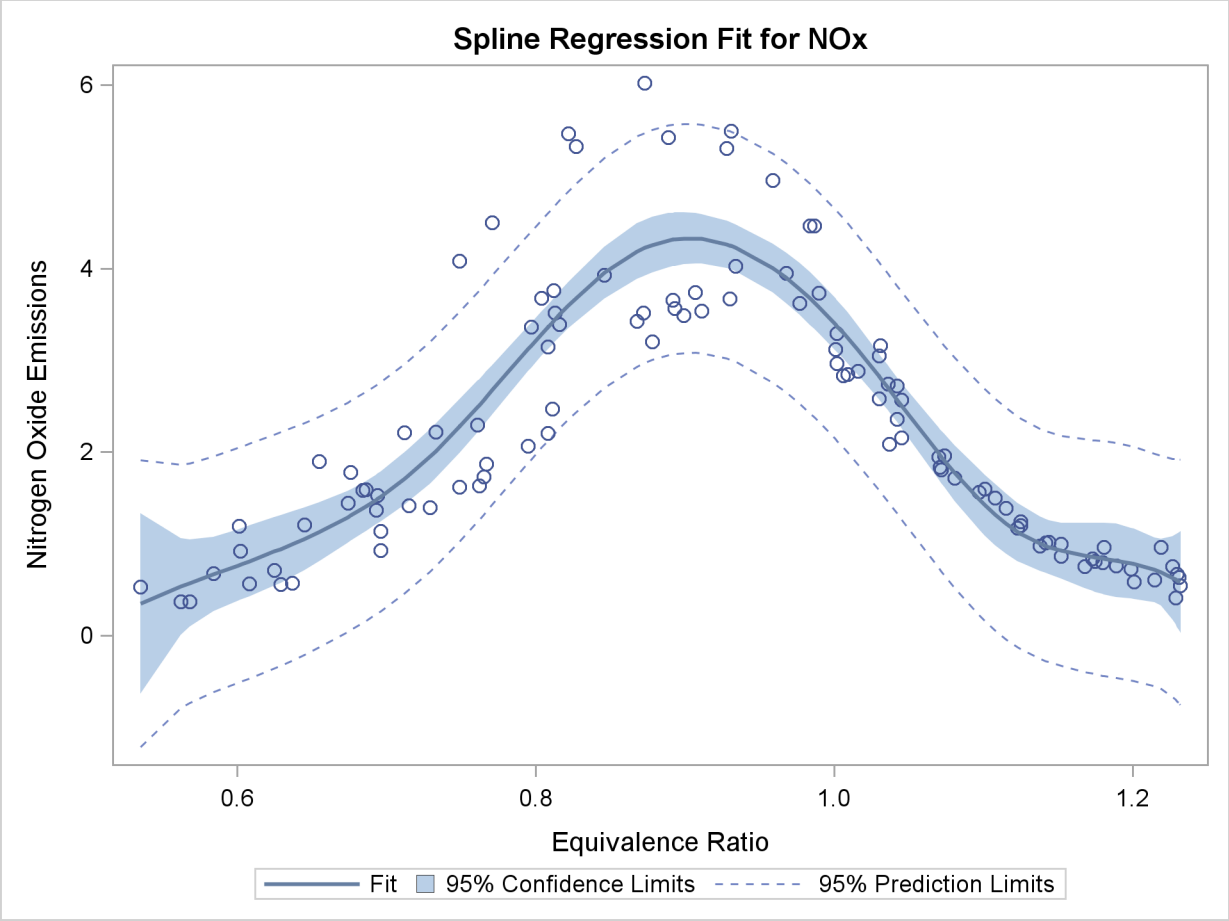
The transformation plots show the identity transformation of NOx and the nonlinear spline transformation of EqRatio. These plots are requested with the PLOTS=TRANSFORMATION option. The plot on the left shows that NOx is unchanged, which is always the case with the IDENTITY transformation. In contrast, the spline transformation of EqRatio is nonlinear. It is this nonlinear transformation of EqRatio that accounts for the increase in fit that is shown in the iteration history table.

Figure 90.3 Residuals



The residuals plot in Figure 90.3 shows the residuals as a function of the transformed independent variable.

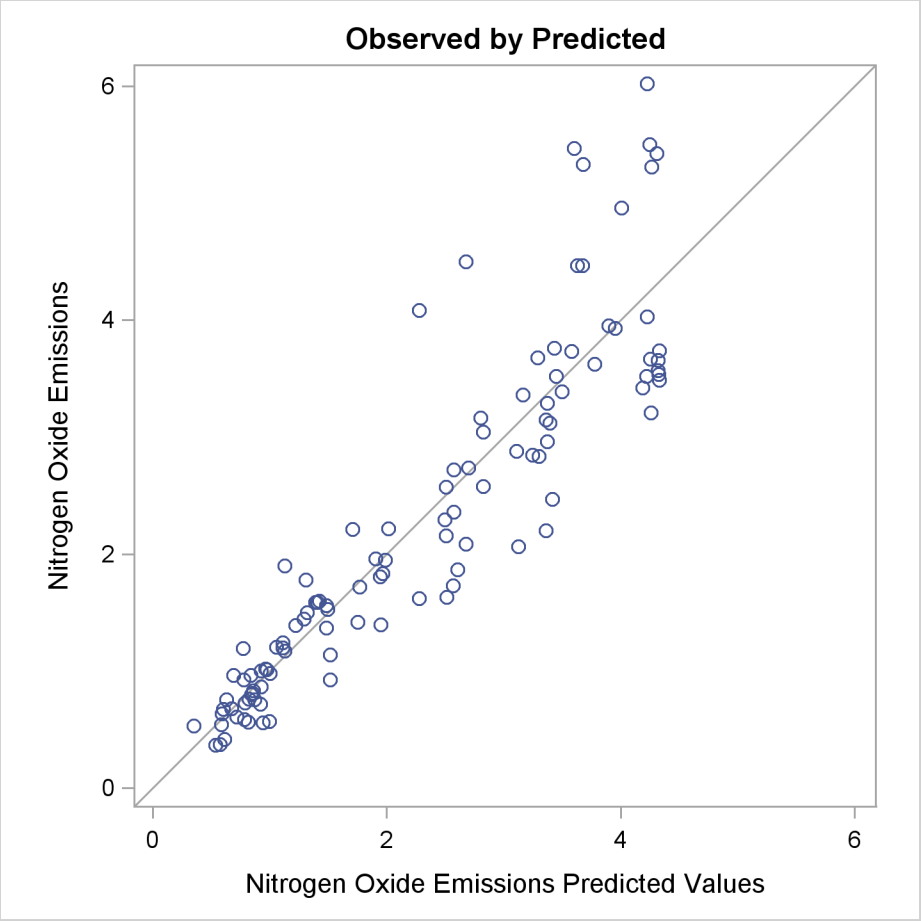
Figure 90.4 Fitting a Curve through a Scatter Plot



The "Spline Regression Fit" plot in Figure 90.4 displays the nonlinear regression function plotted through the original data, along with 95% confidence and prediction limits. This plot clearly shows that nitrous oxide emissions are largest in the middle range of equivalence ratio, 0.08 to 1.0, and are much lower for the extreme values of equivalence ratio, such as around 0.6 and 1.2.

This plot is produced by default when ODS Graphics is enabled and when there is an IDENTITY dependent variable and one non-CLASS independent variable. The plot consists of an ordinary scatter plot of NOx plotted as a function of EqRatio. It also contains the predicted values of NOx, which are a function of the spline transformation of EqRatio (or TEqRatio shown previously), and are plotted as a function of EqRatio. Similarly, it contains confidence limits based on NOx and TEqRatio.

Figure 90.5 Observed by Predicted



The "Observed by Predicted" values plot in Figure 90.5 displays the dependent variable plotted as a function of the regression predicted values along with a linear regression line, which for this plot always has a slope of 1. This plot was requested with the OBP or OBSERVEDBYPREDICTED suboption in the PLOTS= option. The residual differences between the transformed data and the regression line show how well the nonlinearly transformed data fit a linear-regression model. The residuals look mostly random; however, they are larger for larger values of NOx, suggesting that maybe this is not the optimal model. You can also see this by examining the fit of the function through the original scatter plot in Figure 90.4. Near the middle of the function, the residuals are much larger. You can refit the model, this time requesting separate functions for each type of fuel. You can request the original scatter plot, without any regression information and before the variables are transformed, by specifying the SCATTER suboption in the PLOTS= option.

These next statements fit an additive model with separate functions for each of the different fuels. The statements produce Figure 90.6 through Figure 90.9.

```
* Separate Curves and Intercepts;
proc transreg data=Gas solve ss2 additive plots=(transformation obp);
   model identity(nox) = class(Fuel / zero=none) |
                         spline(EqRatio / nknots=4 after);
run;
```

The ADDITIVE *a-option* requests an additive model, where the regression coefficients are absorbed into the transformations, and so the final regression coefficients are all one. The specification CLASS(Fuel / ZERO=NONE) recodes fuel into a set of three binary variables, one for each of the three fuels in this data set. The vertical bar between the CLASS and SPLINE specifications request both main effects and interactions. For this model, it requests both a separate intercept and a separate spline function for each fuel. The original two variables, Fuel and EqRatio, are replaced by six variables—three binary intercept terms and three spline variables. The three spline variables are zero when their corresponding intercept binary variable is zero, and nonzero otherwise. The nonzero parts are optimally transformed by the analysis. The AFTER *t-option* specified with the SPLINE transformation specifies that the four knots should be selected independently for each of the three spline transformations, *after* EqRatio is crossed with the CLASS variable. Alternatively, and by default, the knots are chosen by examining EqRatio before it is crossed with the CLASS variable, and the same knots are used for all three transformations. The results are shown in Figure 90.6.

Figure 90.6 Iteration, ANOVA, and Regression Results

```
                    Gasoline and Emissions Data

                      The TRANSREG Procedure

                  Dependent Variable Identity(NOx)
                      Nitrogen Oxide Emissions


                      Class Level Information

              Class    Levels    Values

              Fuel          3    82rongas Ethanol Gasohol

               Number of Observations Read           112
               Number of Observations Used           110
               Implicit Intercept Model

     TRANSREG MORALS Algorithm Iteration History for Identity(NOx)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        0    0.12476    1.13866     0.18543
        1    0.00000    0.00000     0.95870      0.77327    Converged

Algorithm converged.


    Hypothesis Test Iterations Excluding Spline(Fuel82rongasEqRatio)
     TRANSREG MORALS Algorithm Iteration History for Identity(NOx)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        0    0.00000    0.00000     0.80234
        1    0.00000    0.00000     0.80234      -.00000    Converged
```

Figure 90.6 *continued*

```
Algorithm converged.

     Hypothesis Test Iterations Excluding Spline(FuelEthanolEqRatio)
      TRANSREG MORALS Algorithm Iteration History for Identity(NOx)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        0    0.00000    0.00000     0.48801
        1    0.00000    0.00000     0.48801      -.00000    Converged

Algorithm converged.

     Hypothesis Test Iterations Excluding Spline(FuelGasoholEqRatio)
      TRANSREG MORALS Algorithm Iteration History for Identity(NOx)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        0    0.00000    0.00000     0.80052
        1    0.00000    0.00000     0.80052      -.00000    Converged

Algorithm converged.

        The TRANSREG Procedure Hypothesis Tests for Identity(NOx)
                        Nitrogen Oxide Emissions


       Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                 Sum of        Mean
   Source                DF      Squares      Square    F Value    Pr > F

   Model                 23     209.4613    9.107012      86.80    <.0001
   Error                 86       9.0229    0.104918
   Corrected Total      109     218.4842

             Root MSE           0.32391    R-Square    0.9587
             Dependent Mean     2.25022    Adj R-Sq    0.9477
             Coeff Var         14.39461
```

Figure 90.6 *continued*

Univariate Regression Table Based on the Usual Degrees of Freedom

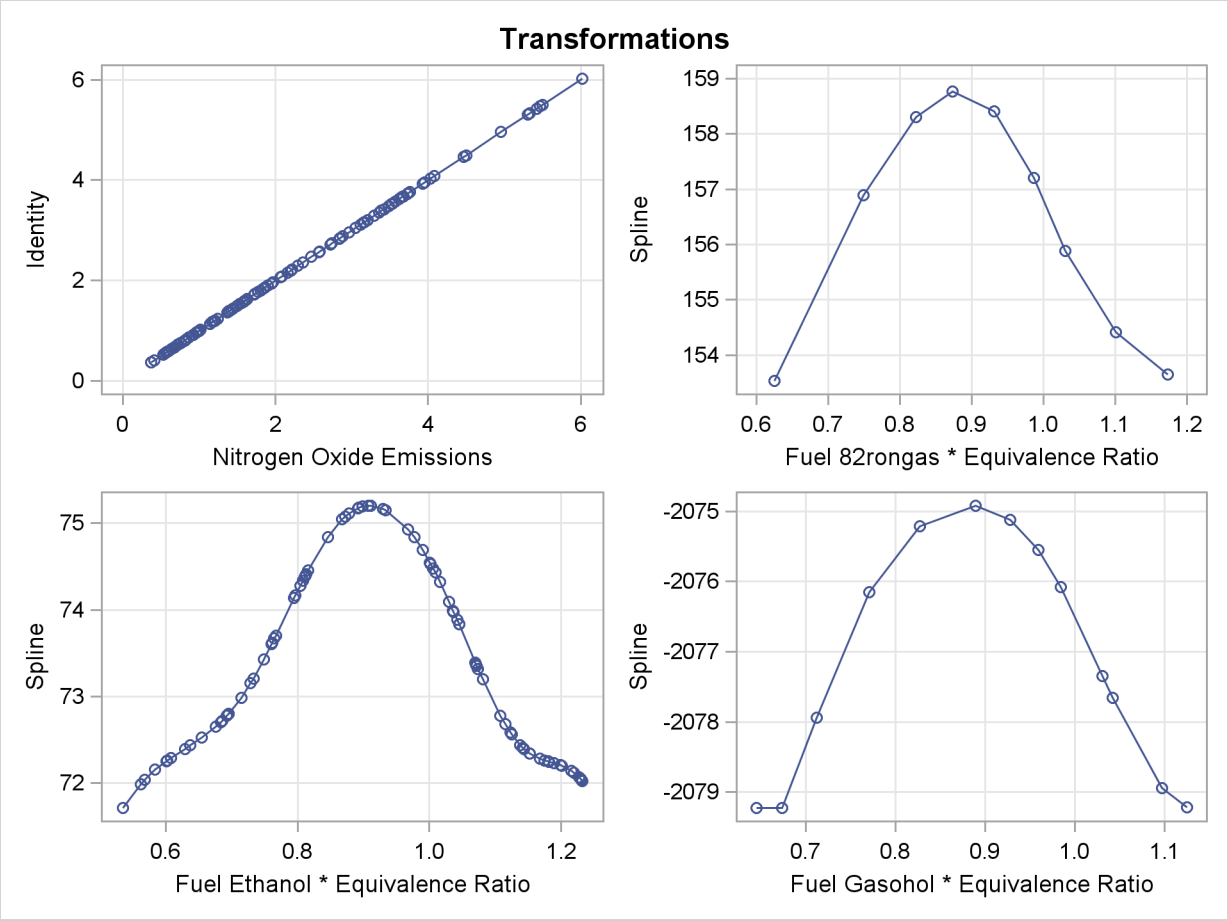
Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Pr > F
Class.Fuel82rongas	1	1.00000000	32.634	32.6338	311.04	<.0001
Class.FuelEthanol	1	1.00000000	97.406	97.4058	928.40	<.0001
Class.FuelGasohol	1	1.00000000	34.672	34.6720	330.47	<.0001
Spline(Fuel82rongasEq Ratio)	7	1.00000000	34.162	4.8803	46.52	<.0001
Spline(FuelEthanolEq Ratio)	7	1.00000000	102.840	14.6914	140.03	<.0001
Spline(FuelGasoholEq Ratio)	7	1.00000000	34.561	4.9372	47.06	<.0001

Variable	DF	Label
Class.Fuel82rongas	1	Fuel 82rongas
Class.FuelEthanol	1	Fuel Ethanol
Class.FuelGasohol	1	Fuel Gasohol
Spline(Fuel82rongasEq Ratio)	7	Fuel 82rongas * Equivalence Ratio
Spline(FuelEthanolEq Ratio)	7	Fuel Ethanol * Equivalence Ratio
Spline(FuelGasoholEq Ratio)	7	Fuel Gasohol * Equivalence Ratio

ZERO=SUM and ZERO=NONE coefficient tests are not exact when there are iterative transformations. Those tests are performed holding all transformations fixed, and so are generally liberal.

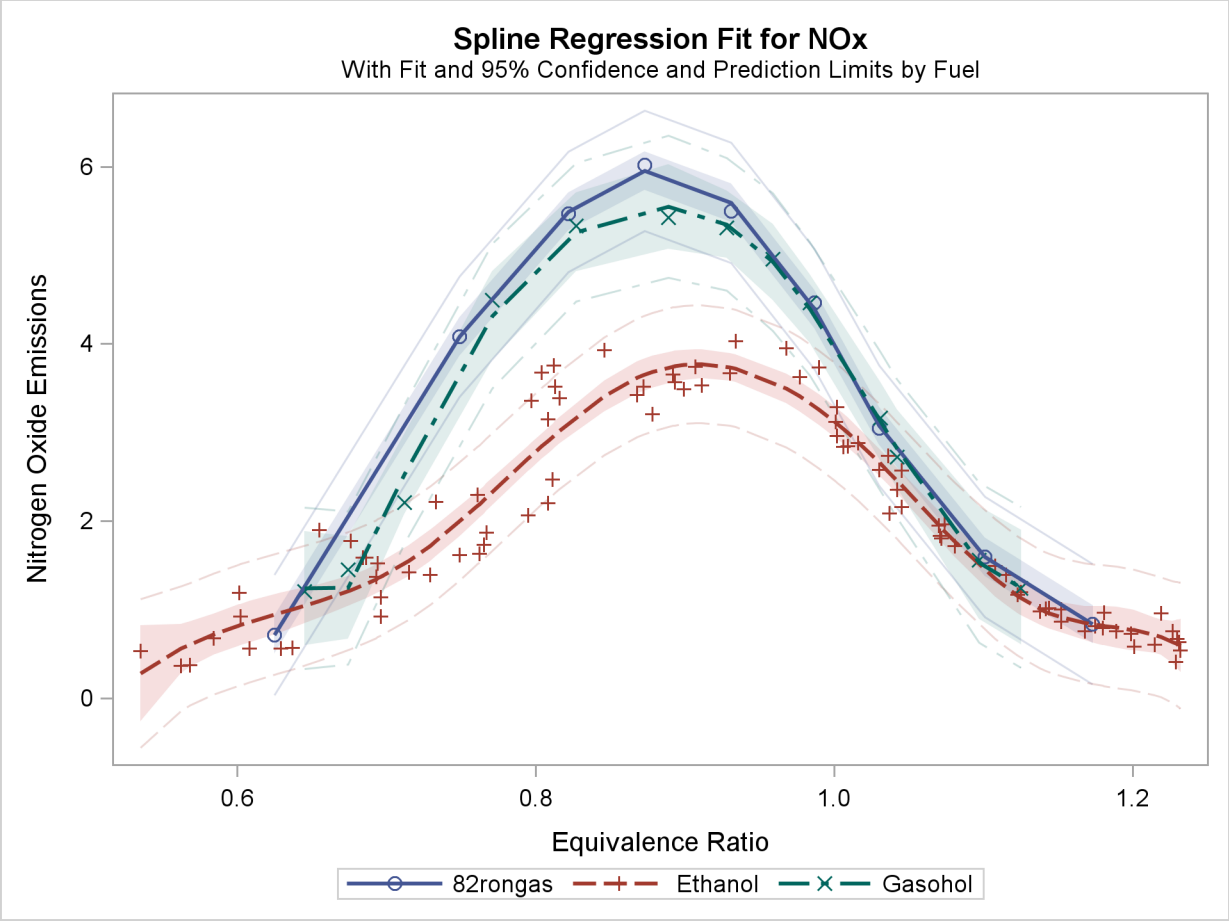
The first iteration history table in Figure 90.6 hows that PROC TRANSREG increases the squared multiple correlation from the original value of 0.18543 to 0.95870. The remaining iteration histories pertain to PROC TRANSREG's process of comparing models to test hypotheses. The important thing to look for is convergence in all of the tables.

Figure 90.7 Transformations



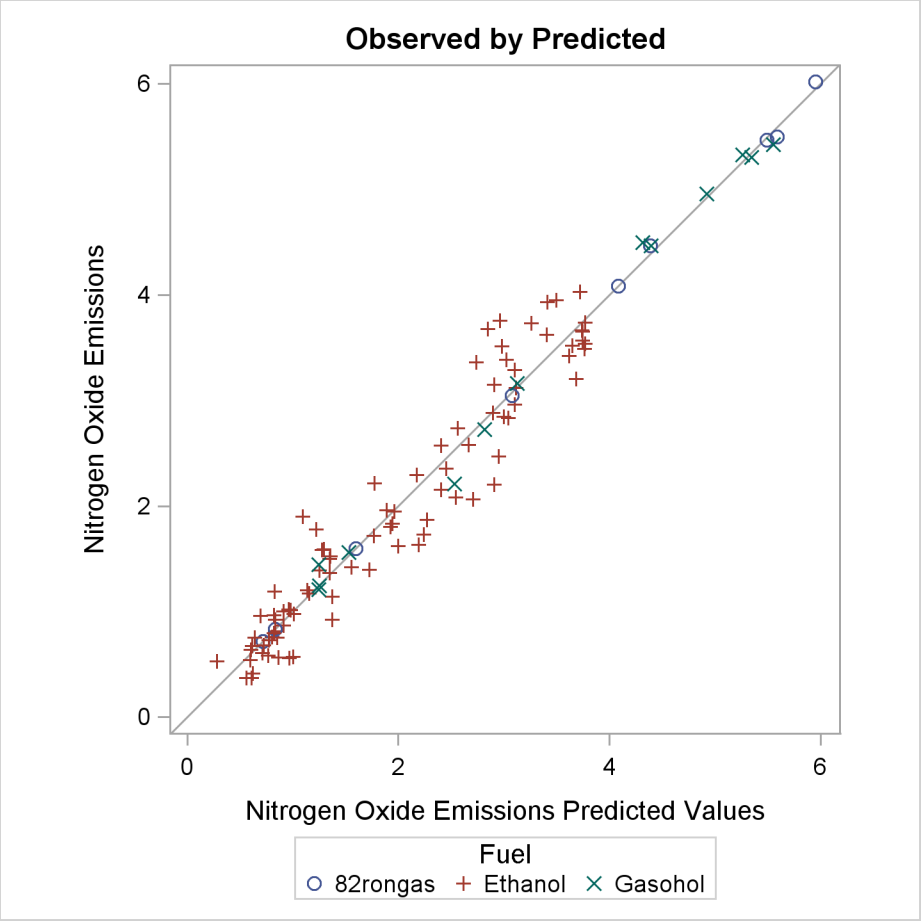
The transformations, shown in Figure 90.7, show that for all three groups, the transformation of EqRatio is approximately quadratic.

Figure 90.8 Fitting Curves through a Scatter Plot



The fit plot, shown in Figure 90.8, shows that there are in fact three distinct functions in the data. The increase in fit over the previous model comes from individually fitting each group instead of providing an aggregate fit.

Figure 90.9 Observed by Predicted



The residuals in the observed by predicted plot displayed in Figure 90.9 are much better for this analysis.

You could fit a model that is "in between" the two models shown previously. This next model provides for separate intercepts for each group, but calls for a common function. There are still three functions, one per group, but their shapes are the same, and they are equidistant or parallel. This model is requested by omitting the vertical bar so that separate intercepts are requested, but not separate curves within each group. The following statements fit the separate intercepts model and create Figure 90.10:

```
* Separate Intercepts;
proc transreg data=Gas solve ss2 additive;
   model identity(nox) = class(Fuel / zero=none)
                         spline(EqRatio / nknots=4);
run;
```

The ANOVA table and fit plot are shown in Figure 90.10.

Figure 90.10 Separate Intercepts Only

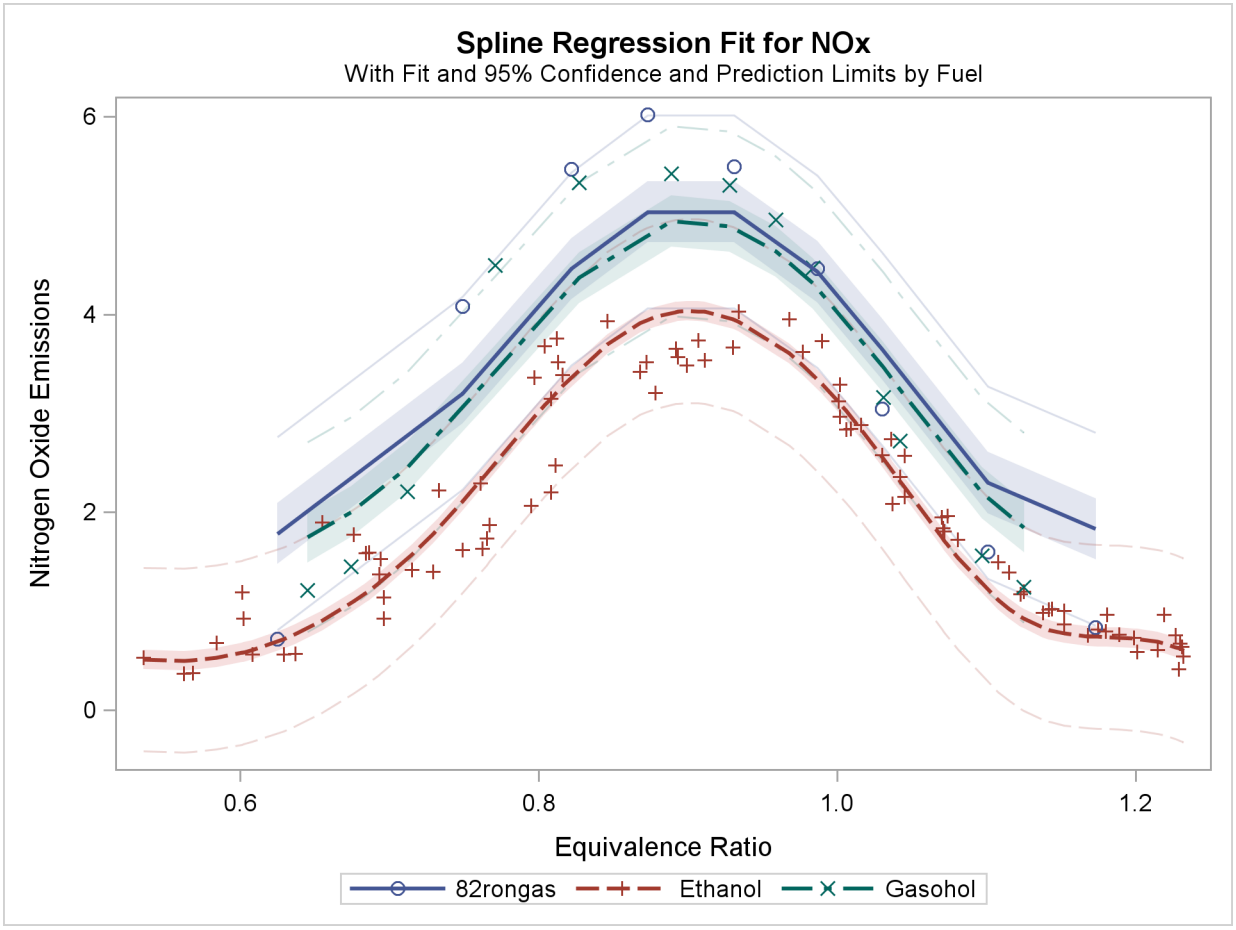
Gasoline and Emissions Data

The TRANSREG Procedure

Univariate ANOVA Table Based on the Usual Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	9	196.7548	21.86165	100.61	<.0001
Error	100	21.7294	0.21729		
Corrected Total	109	218.4842			

Figure 90.10 *continued*



Now, squared multiple correlation is 0.9005, which is smaller than the model with the unconstrained separate curves, but larger than the model with only one curve. Because of the restrictions on the shapes, these curves do not track the data as well as the previous model. However, this model is more parsimonious with many fewer parameters.

There are other ways to fit curves through scatter plots in PROC TRANSREG. For example, you could use smoothing splines or penalized B-splines, as is illustrated next. The following statements fit separate curves through each group by using penalized B-splines and produce Figure 90.11:

```
* Separate Curves and Intercepts with Penalized B-Splines;
proc transreg data=Gas ss2 plots=transformation lprefix=0;
   model identity(nox) = class(Fuel / zero=none) * pbspline(EqRatio);
run;

ods graphics off;
```

This example asks for a separate penalized B-spline transformation, PBSPLINE, of equivalence ratio for each type of fuel. The LPREFIX=0 *a-option* is specified in the PROC statement so that zero characters of the CLASS variable name (Fuel) are used in constructing the labels for the coded variables. The result is label components like "Ethanol" instead of the more redundant "Fuel Ethanol". The results of this analysis are shown in Figure 90.11.

Figure 90.11 Penalized B-Splines

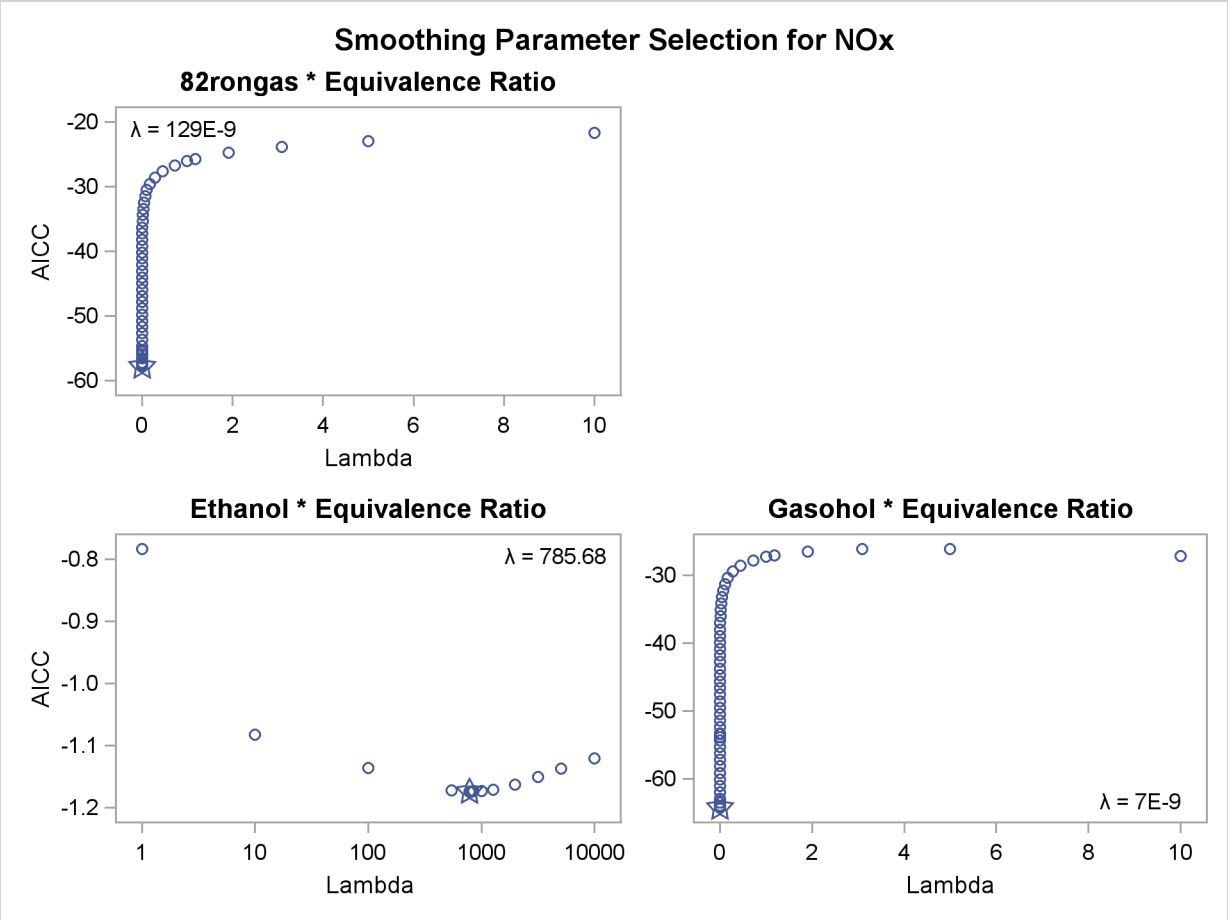


Figure 90.11 *continued*

```
                    Dependent Variable Identity(NOx)
                        Nitrogen Oxide Emissions


                           Class Level Information

                 Class    Levels    Values

                 Fuel          3    82rongas Ethanol Gasohol

                  Number of Observations Read           112
                  Number of Observations Used           110
                  Implicit Intercept Model

       TRANSREG Univariate Algorithm Iteration History for Identity(NOx)

              Iteration    Average    Maximum
                 Number     Change     Change    Note
              -----------------------------------------------
                      1    0.00000    0.00000    Converged

   Algorithm converged.

         The TRANSREG Procedure Hypothesis Tests for Identity(NOx)
                        Nitrogen Oxide Emissions


         Univariate ANOVA Table, Penalized B-Spline Transformation

                                  Sum of        Mean
     Source               DF     Squares      Square    F Value    Pr > F

     Model            33.194    211.4818    6.371106      68.97    <.0001
     Error            75.806      7.0024    0.092373
     Corrected Total     109    218.4842

               Root MSE           0.30393    R-Square    0.9680
               Dependent Mean     2.25022    Adj R-Sq    0.9539
               Coeff Var         13.50663

                      Penalized B-Spline Transformation

Variable                      DF Coefficient   Lambda     AICC Label

Pbspline(Fuel82rongasEq    9.000       1.000 1.287E-7 -57.7841 82rongas *
Ratio)                                                        Equivalence
                                                              Ratio
Pbspline(FuelEthanolEq    12.194       1.000    785.7  -1.1736 Ethanol *
Ratio)                                                        Equivalence
                                                              Ratio
Pbspline(FuelGasoholEq    13.000       1.000 7.019E-9 -64.2961 Gasohol *
Ratio)                                                        Equivalence
                                                              Ratio
```

Figure 90.11 *continued*

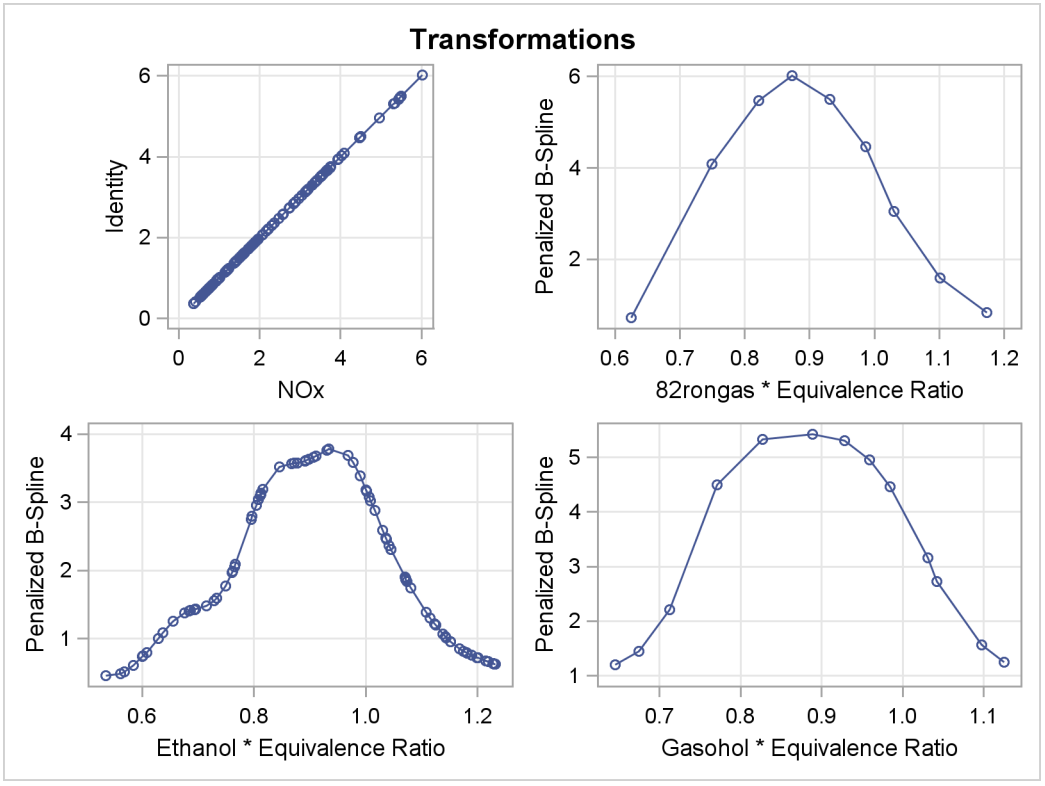
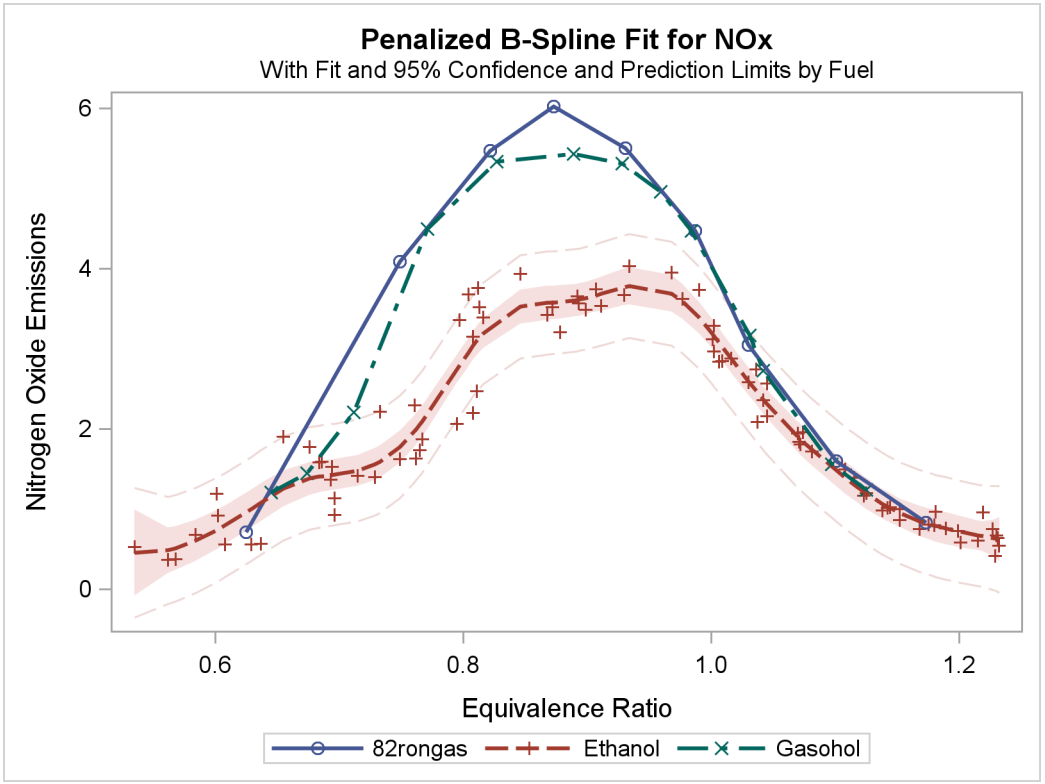


Figure 90.11 *continued*



With penalized B-splines, the degrees of freedom are based on the trace of the transformation hat matrix and are typically not integers. The first panel of plots shows AICC as a function of lambda, the smoothing parameter. The smoothing parameter is automatically chosen, and since the smoothing parameters range from essentially 0 to almost 800, it is clear that some functions are smoother than others. The plots of the criterion (AICC in this example) as a function of lambda use a linear scale for the horizontal axis when the range of lambdas is small, as in the first and third plot, and a log scale when the range is large, as in the second plot. The transformation for equivalence ratio for Ethanol required more smoothing than for the other two fuels. All three have an overall quadratic shape, but for Ethanol, the function more closely follows the smaller variations in the data. You could get similar results with SPLINE by using more knots.

For other examples of curve fitting by using PROC TRANSREG, see the sections "Smoothing Splines" on page 7222, "Linear and Nonlinear Regression Functions" on page 7209, "Simultaneously Fitting Two Regression Functions" on page 7213, and "Using Splines and Knots" on page 7192, as well as Example 90.3. These examples include cases where multiple curves are fit through scatter plots with multiple groups. Special cases include linear models with separate slopes and separate intercepts. Many constraints on the slopes, curves, and intercepts are possible.

Main-Effects ANOVA

This example shows how to use PROC TRANSREG to code and fit a main-effects ANOVA model. PROC TRANSREG has very extensive and versatile options for coding or creating so-called dummy variables. PROC TRANSREG is commonly used to code classification variables before they are used for analysis in other procedures. See the sections "Using the DESIGN Output Option" on page 7293 and "Discrete Choice Experiments: DESIGN, NORESTORE, NOZERO" on page 7297. In this example, the input data set contains the dependent variables `y`, factors `x1` and `x2`, and 12 observations. PROC TRANSREG can be useful for coding even before running procedures with a CLASS statement because of its detailed options that enable you to control how the coded variable names and labels are constructed. The following statements perform a main-effects ANOVA and display the results in Figure 90.12 and Figure 90.13:

```
title 'Introductory Main-Effects ANOVA Example';

data a;
   input y x1 $ x2 $;
   datalines;
8 a a
7 a a
4 a b
3 a b
5 b a
4 b a
2 b b
1 b b
8 c a
7 c a
5 c b
2 c b
;
```

```
* Fit a main-effects ANOVA model with 1, 0, -1 coding;
proc transreg ss2;
   model identity(y) = class(x1 x2 / effects);
   output coefficients replace;
run;

* Display TRANSREG output data set;
proc print label;
   format intercept -- x2a 5.2;
run;
```

The SS2 *a-option* requests results based on Type II sums of squares. The simple ANOVA model is fit by designating y as an IDENTITY variable, which specifies no transformation. The independent variables are specified with a CLASS expansion, which replaces them with coded variables. There are $(3-1)+(2-1)=3$ coded variables created by the CLASS specification, since the two CLASS variables have 3 and 2 different values or levels. In this case, the EFFECTS *t-option* is specified. This option requests an *effects coding* (displayed in Figure 90.13), which is also called a deviations from means or 0, 1, –1 coding. The OUTPUT statement requests an output data set with the data and coded variables. The COEFFICIENTS output option, or *o-option*, adds the parameter estimates and marginal means to the data set. The REPLACE *o-option* specifies that the transformed variables should replace the original variables in the output data set. The output data set variable names are the same as the original variable name. In an example like this, there are no nonlinear transformations; the transformed variables are the same as the original variables. The REPLACE *o-option* is used to eliminate unnecessary and redundant transformed variables from the output data set. The results of the PROC TRANSREG step are shown in Figure 90.12.

Figure 90.12 ANOVA Example Output from PROC TRANSREG

```
                Introductory Main-Effects ANOVA Example

                      The TRANSREG Procedure

                   Dependent Variable Identity(y)


                      Class Level Information

                    Class    Levels    Values

                    x1            3    a b c

                    x2            2    a b

                Number of Observations Read           12
                Number of Observations Used           12
```

Figure 90.12 *continued*

The TRANSREG Procedure Hypothesis Tests for Identity(y)

Univariate ANOVA Table Based on the Usual Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	57.00000	19.00000	19.83	0.0005
Error	8	7.66667	0.95833		
Corrected Total	11	64.66667			

Root MSE	0.97895	R-Square	0.8814
Dependent Mean	4.66667	Adj R-Sq	0.8370
Coeff Var	20.97739		

Univariate Regression Table Based on the Usual Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Pr > F	Label
Intercept	1	4.6666667	261.333	261.333	272.70	<.0001	Intercept
Class.x1a	1	0.8333333	4.167	4.167	4.35	0.0705	x1 a
Class.x1b	1	-1.6666667	16.667	16.667	17.39	0.0031	x1 b
Class.x2a	1	1.8333333	40.333	40.333	42.09	0.0002	x2 a

Figure 90.12 shows the ANOVA results, fit statistics, and regression tables. The output data set, with the coded design, parameter estimates and means, is shown in Figure 90.13. For more information about PROC TRANSREG for ANOVA and other codings, see the section "ANOVA Codings" on page 7230.

Figure 90.13 Output Data Set from PROC TRANSREG

Introductory Main-Effects ANOVA Example

Obs	_TYPE_	_NAME_	y	Intercept	x1 a	x1 b	x2 a	x1	x2
1	SCORE	ROW1	8	1.00	1.00	0.00	1.00	a	a
2	SCORE	ROW2	7	1.00	1.00	0.00	1.00	a	a
3	SCORE	ROW3	4	1.00	1.00	0.00	-1.00	a	b
4	SCORE	ROW4	3	1.00	1.00	0.00	-1.00	a	b
5	SCORE	ROW5	5	1.00	0.00	1.00	1.00	b	a
6	SCORE	ROW6	4	1.00	0.00	1.00	1.00	b	a
7	SCORE	ROW7	2	1.00	0.00	1.00	-1.00	b	b
8	SCORE	ROW8	1	1.00	0.00	1.00	-1.00	b	b
9	SCORE	ROW9	8	1.00	-1.00	-1.00	1.00	c	a
10	SCORE	ROW10	7	1.00	-1.00	-1.00	1.00	c	a
11	SCORE	ROW11	5	1.00	-1.00	-1.00	-1.00	c	b
12	SCORE	ROW12	2	1.00	-1.00	-1.00	-1.00	c	b
13	M COEFFI	y	.	4.67	0.83	-1.67	1.83		
14	MEAN	y	.	.	5.50	3.00	6.50		

The output data set has three kinds of observations, identified by values of _TYPE_ as follows:

- When _TYPE_='SCORE', the observation contains the following information about the dependent and independent variables:
  - y is the original dependent variable.
  - x1 and x2 are the independent classification variables, and the Intercept through x2 a columns contain the main-effects design matrix that PROC TRANSREG creates. The variable names are Intercept, x1a, x1b, and x2a. Their labels are shown in the listing.
- When _TYPE_='M COEFFI', the observation contains coefficients of the final linear model (parameter estimates).
- When _TYPE_='MEAN', the observation contains the marginal means.

The observations with _TYPE_='SCORE' form the score or data partition of the output data set, and the observations with _TYPE_='M COEFFI' and _TYPE_='MEAN' form the output statistics partition of the output data set.

Syntax: TRANSREG Procedure

The following statements are available in PROC TRANSREG:

```
PROC TRANSREG <DATA=SAS-data-set>
                <PLOTS=(plot-requests)>
                <OUTTEST=SAS-data-set> <a-options> <o-options> ;
  MODEL  <transform(dependents   </ t-options>)>
         <transform(dependents   </ t-options>) . . . =>
           transform(independents </ t-options>)
         <transform(independents </ t-options>) . . . > </ a-options> ;
  OUTPUT <OUT=SAS-data-set> <o-options> ;
  ID variables ;
  FREQ variable ;
  WEIGHT variable ;
  BY variables ;
```

To use PROC TRANSREG, you need both the PROC TRANSREG and MODEL statements. To produce an OUT= output data set, the OUTPUT statement is required. PROC TRANSREG enables you to specify the same options in more than one statement. All of the MODEL statement *a-options* (algorithm options) and all of the OUTPUT statement *o-options* (output options) can also be specified in the PROC TRANSREG statement. You can abbreviate all *a-options*, *o-options*, and *t-options* (transformation options) to their first three letters. This is a special feature of PROC TRANSREG and is not generally true of other SAS/STAT procedures. See Table 90.1 for a list of options available in the PROC TRANSREG statement.

The PROC TRANSREG statement starts the TRANSREG procedure. Optionally, this statement identifies an input and an OUTTEST= data set, specifies the algorithm and other computational details, requests displayed output, and controls the contents of the OUT= data set (which is created with the OUTPUT statement). The DATA= and OUTTEST= options can appear only in the PROC TRANSREG statement. All *a-options* and *o-options* are described in the sections on either the MODEL or OUTPUT statement, in which these options can also be specified.

The rest of this section provides detailed syntax information for each of the preceding statements, beginning with the PROC TRANSREG statement. The remaining statements are described in alphabetical order.

PROC TRANSREG Statement

PROC TRANSREG < **DATA=***SAS-data-set* >
< **PLOTS=***(plot-requests)* >
< **OUTTEST=***SAS-data-set* > < *a-options* > < *o-options* > **;**

The PROC TRANSREG statement invokes the TRANSREG procedure. Optionally, this statement identifies an input and an OUTTEST= data set, specifies the algorithm and other computational details, requests displayed output, and controls the contents of the OUT= data set (which is created with the OUTPUT statement). The DATA=, OUTTEST=, and PLOTS= options can appear only in the PROC TRANSREG statement. The options listed in Table 90.1 are available in the PROC TRANSREG statement. The *a-options* are also available in the MODEL statement, and the *o-options* are also available in the OUTPUT statement.

Table 90.1 Options Available in the PROC TRANSREG Statement

Option	Description
Data Set Options (PROC Statement)	
DATA=	specifies input SAS data set
OUTTEST=	specifies output test statistics data set
ODS Graphics (PROC Statement)	
PLOTS=	specifies ODS Graphics selection
Input Control (PROC or MODEL)	
REITERATE	restarts the iterations
TYPE=	specifies input observation type
Method and Iterations (PROC or MODEL)	
CCONVERGE=	specifies minimum criterion change
CONVERGE=	specifies minimum data change
MAXITER=	specifies maximum number of iterations
METHOD=	specifies iterative algorithm
NCAN=	specifies number of canonical variables
NSR	specifies no restrictions on smoothing models
SINGULAR=	specifies singularity criterion
SOLVE	attempts direct solution instead of iteration

Table 90.1 *continued*

Option	Description
Missing Data Handling (PROC or MODEL)	
INDIVIDUAL	fits each model individually (METHOD=MORALS)
MONOTONE=	includes monotone special missing values
NOMISS	excludes observations with missing values
UNTIE=	unties special missing values
Intercept and CLASS Variables (PROC or MODEL)	
CPREFIX=	specifies CLASS coded variable name prefix
LPREFIX=	specifies CLASS coded variable label prefix
NOINT	specifies no intercept or centering
ORDER=	specifies order of CLASS variable levels
REFERENCE=	controls output of reference levels
SEPARATORS=	controls CLASS coded variable label separators
Control Displayed Output (PROC or MODEL)	
ALPHA=	specifies confidence limits alpha
CL	displays parameter estimate confidence limits
DETAIL	displays model specification details
HISTORY	displays iteration histories
NOPRINT	suppresses displayed output
PBOXCOXTABLE	prints the Box-Cox log likelihood table
RSQUARE	displays the R square
SHORT	suppresses the iteration histories
SS2	displays regression results
TEST	displays ANOVA table
TSUFFIX=	shortens transformed variable labels
UTILITIES	displays conjoint part-worth utilities
Standardization (PROC or MODEL)	
ADDITIVE	fits additive model
NOZEROCONSTANT	does not zero constant variables
TSTANDARD=	specifies transformation standardization
Predicted Values, Residuals, Scores (PROC or OUTPUT)	
CANONICAL	outputs canonical scores
CLI	outputs individual confidence limits
CLM	outputs mean confidence limits
DESIGN=	specifies design matrix coding
DREPLACE	replaces dependent variables
IREPLACE	replaces independent variables
LEVERAGE	outputs leverage
NORESTOREMISSING	does not restore missing values
NOSCORES	suppresses output of scores
PREDICTED	outputs predicted values
REDUNDANCY=	outputs redundancy variables
REPLACE	replaces all variables

Table 90.1 *continued*

Option	Description
RESIDUALS	outputs residuals
Output Data Set Coefficients (PROC or OUTPUT)	
COEFFICIENTS	outputs coefficients
COORDINATES=	outputs ideal point coordinates
MEANS	outputs marginal means
MREDUNDANCY	outputs redundancy analysis coefficients
Output Data Set Variable Name Prefixes (PROC or OUTPUT)	
ADPREFIX=	specifies dependent variable approximations
AIPREFIX=	specifies independent variable approximations
CDPREFIX=	specifies canonical dependent variables
CILPREFIX=	specifies conservative individual lower CL
CIPREFIX=	specifies canonical independent variables
CIUPREFIX=	specifies conservative-individual-upper CL
CMLPREFIX=	specifies conservative-mean-lower CL
CMUPREFIX=	specifies conservative-mean-upper CL
DEPENDENT=	specifies METHOD=MORALS untransformed dependent
LILPREFIX=	specifies liberal-individual-lower CL
LIUPREFIX=	specifies liberal-individual-upper CL
LMLPREFIX=	specifies liberal-mean-lower CL
LMUPREFIX=	specifies liberal-mean-upper CL
PPREFIX=	specifies predicted values
RDPREFIX=	specifies residuals
RPREFIX=	specifies redundancy variables
TDPREFIX=	specifies transformed dependents
TIPREFIX=	specifies transformed independents
Macros Variables (PROC or OUTPUT)	
MACRO	creates macro variables
Other Options (PROC or OUTPUT)	
APPROXIMATIONS	outputs dependent and independent approximations
CCC	outputs canonical correlation coefficients
CEC	outputs canonical elliptical point coordinates
CPC	outputs canonical point coordinates
CQC	outputs canonical quadratic point coordinates
DAPPROXIMATIONS	outputs approximations to transformed dependents
IAPPROXIMATIONS	outputs approximations to transformed independents
MEC	outputs elliptical point coordinates
MPC	outputs point coordinates
MQC	outputs quadratic point coordinates
MRC	outputs multiple regression coefficients

DATA=*SAS-data-set*

specifies the SAS data set to be analyzed. If you do not specify the DATA= option, PROC TRANSREG uses the most recently created SAS data set. The data set must be an ordinary SAS data set; it cannot be a special TYPE= data set.

OUTTEST=*SAS-data-set*

specifies an output data set to contain hypothesis tests results. When you specify the OUTTEST= option, the data set contains ANOVA results. When you specify the SS2 *a-option*, regression tables are also output. When you specify the UTILITIES *o-option*, conjoint analysis part-worth utilities are also output. For more information about the OUTTEST= data set, see the section "OUTTEST= Output Data Set" on page 7275.

PLOTS <(*global-plot-options***)> <= *plot-request* <(options)>>**

PLOTS <(*global-plot-options***)> <= (***plot-request* <(options)> **<... *plot-request* <(options)>>)>**

controls the plots produced through ODS Graphics. When you specify only one plot request, you can omit the parentheses around the plot request. Here are some examples:

```
plots=none
plots=(residuals transformation)
plots(unpack)=boxcox
plots(unpack)=(transformation boxcox(p=0))
plots=(residuals(unpack) transformation(dep unp) boxcox(t rmse))
```

You must enable ODS Graphics before requesting plots, for example, like this:

```
ods graphics on;

proc transreg plots=all;
   model identity(y) = pbspline(x);
run;

ods graphics off;
```

For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." If you have enabled ODS Graphics but do not specify the PLOTS= option, then PROC TRANSREG produces a default set of plots. The fit, scatter, residual, and observed-by-predicted plots are available with METHOD=MORALS and also with METHOD=UNIVARIATE when there is only one dependent variable. When no method is specified and there is more than one dependent variable, and when regression plots are requested, the default method is set to METHOD=MORALS. When there is more than one dependent variable, when METHOD= is not specified, or when METHOD=MORALS is specified and PLOTS=ALL is specified, the plots that are produced might be different from those you would see with METHOD=UNIVARIATE and PLOTS=ALL. Certain plots appear by default when ODS Graphics is enabled and certain combinations of options are specified. The Box-Cox $F = t^2$ and log-likelihood plots appear when a BOXCOX dependent variable transformation is specified. The regression fit plot appears for models with a single dependent variable that is not transformed (for example, IDENTITY(y)), a single quantitative independent variable that might or might not be transformed, and at most one CLASS independent variable. Preference mapping plots appear when the COORDINATES *o-option* is used.

The global plot options include the following:

INTERPOLATE

INT

uses observations that are excluded from the analysis for interpolation in the fit and transformation plots. By default, observations with zero weight are excluded from all plots. These include observations with a zero, negative, or missing weight or frequency and observations excluded due to missing and invalid values. You can specify PLOTS(INTERPOLATE)=(*plot-requests*) to include some of these observations in the plots. You might want to use this option, for example, with sparse data sets to show smoother functions over the range of the data (see the section "The PLOTS(INTERPOLATE) Option" on page 7302). Observations with missing values in CLASS variables are excluded from the plots even when PLOTS(INTERPOLATE) is specified.

ONLY

ONL

suppresses the default plots. Only plots specifically requested are displayed.

UNPACKPANEL

UNPACK

UNP

suppresses paneling. By default, multiple plots can appear in some output panels. Specify UNPACKPANEL to get each plot in a separate panel. You can specify PLOTS(UNPACKPANEL) to unpack the default plots. You can also specify UNPACKPANEL as a suboption with TRANSFORMATION, RESIDUALS, PBSPLINE, and BOXCOX.

The plot requests include the following:

ALL

produces all appropriate plots. You can specify other options with ALL; for example, to request all plots and unpack only the residuals, specify PLOTS=(ALL RES(UNP)).

BOXCOX < (options) >

BOX < (options) >

requests a display of the results of the Box-Cox transformation. These results are displayed by default when there is a Box-Cox transformation. The BOXCOX plot request has the following options:

P=*n*

adds t or $F = t^2$ curves to the legend for the functions where $p(t) < n$, where t is the t statistic corresponding to the optimal lambda. You can specify P=0 to suppress the legend and P=1 to see all curves in the legend. The default value comes from the BOXCOX(variable / ALPHA=p) specification, which by default is 0.05.

RMSE

RMS

plots the root mean square error as a function of lambda.

T

plots t statistics rather than $F = t^2$ statistics.

UNPACKPANEL

UNPACK

UNP

plots the t or $F = t^2$ and log-likelihood plots in separate panels.

FIT < (options) >

requests a regression fit plot. This plot is produced by default whenever it is appropriate. It is produced when the dependent variable is specified with the IDENTITY *transform*, and when there is one quantitative independent variable (for example, IDENTITY for linear fit or SPLINE or one of the other transformations for a nonlinear fit) and at most one CLASS variable. When there is a CLASS variable, separate fits are produced within levels based on your model. You would specify the FIT plot request only to specify a FIT option or with the ONLY global plot option. The FIT plot request has the following options:

NOCLM

suppresses the confidence limits in regression fit plots.

NOCLI

suppresses the individual prediction limits in regression fit plots.

NOOBS

suppresses the observations showing only the fit function and optionally the confidence and prediction limits.

NONE

suppresses all plots.

OBSERVEDBYPREDICTED

OBP

OBS

plots the transformed dependent variable as a function of the regression predicted values.

PBSPLINE < (UNPACKPANEL) >

PBS < (UNPACK) >

requests the penalized B-spline criterion plots. You would specify the PBSPLINE plot request only to specify a PBSPLINE option or with the ONLY global plot option. The PBSPLINE plot request has the following option:

UNPACKPANEL

UNPACK

UNP

plots each criterion plot in a separate panel.

PREFMAP

PRE

plots ideal point or vector preference mapping results when either two IDENTITY or two POINT independent variables are specified along with the COORDINATES option.

RESIDUALS < (options) >

RES < (options) >

plots the residuals as a function of each of the transformed independent variables, except coded CLASS variables. The RESIDUALS plot request has the following options:

CLASS

CLA

plots the residuals as a function of each of the transformed independent variables, including coded CLASS variables. Note that the ALL plot request, which you use to request all plots, specifies the RESIDUALS plot request without the CLASS option.

UNPACKPANEL

UNPACK

UNP

plots the residuals in separate plots, not several per panel.

SMOOTH

SMO

adds a LOESS smooth function to the residuals plots.

SCATTER

SCA

plots the scatter plot of observed data, before the transformations, for models with a single quantitative dependent variable, a single quantitative independent variable, and at most one CLASS independent variable.

TRANSFORMATION < (options) >

TRA < (options) >

plots the variable transformations. The TRANSFORMATION plot request has the following options:

DEPENDENTS

DEP

plots only the dependent variable transformations.

INDEPENDENTS

IND

plots only the independent variable transformations.

UNPACKPANEL

UNPACK

UNP

plots the transformations in separate plots, not several per panel.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC TRANSREG to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC TRANSREG. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

FREQ Statement

FREQ *variable* **;**

If one variable in the input data set represents the frequency of occurrence for other values in the observation, specify the variable's name in a FREQ statement. PROC TRANSREG then treats the data set as if each observation appeared *n* times, where *n* is the value of the FREQ variable for the observation. Noninteger values of the FREQ variable are truncated to the largest integer less than the FREQ value. The observation is used in the analysis only if the value of the FREQ statement variable is greater than or equal to 1.

ID Statement

ID *variables* **;**

The ID statement includes additional character or numeric variables in the OUT= data set. The variables must be contained in the input data set. The first variable is used to label points in PREFMAP plots. These variables are also used in some plots as tip variables.

MODEL Statement

MODEL < *transform*(*dependents* < / *t-options* >) >
< *transform*(*dependents* < / *t-options* >) . . . =>
transform(*independents* < / *t-options* >)
< *transform*(*independents* < / *t-options* >) . . . > < / *a-options* > **;**

The MODEL statement specifies the dependent and independent variables (*dependents* and *independents*, respectively) and specifies the transformation (*transform*) to apply to each variable. Only one MODEL statement can appear in PROC TRANSREG. The *t-options* are transformation options, and the *a-options* are algorithm options. The *t-options* provide details for the transformation; these depend on the *transform* chosen. The *t-options* are listed after a slash in the parentheses that enclose the variable list (either *dependents* or *independents*). The *a-options* control the algorithm used, details of iteration, details of how the intercept and coded variables are generated, and displayed output details. The *a-options* are listed after the entire model specification (the *dependents*, *independents*, transformations, and *t-options*) and after a slash. You can also specify the algorithm options in the PROC TRANSREG statement. When you specify the DESIGN *o-option*, *dependents* and an equal sign are not required. The operators *, |, and @ from the GLM procedure are available for interactions with the CLASS expansion and the IDENTITY transformation. They are used as follows:

```
   Class(a * b ...
         c | d ...
         e | f ... @ n)
Identity(a * b ...
         c | d ...
         e | f ... @ n)
```

In addition, transformations and spline expansions can be crossed with classification variables as follows:

transform`(var) * class(group)`
transform`(var) | class(group)`

See the section "Types of Effects" on page 2488 in Chapter 39, "The GLM Procedure," for a description of the @, *, and | operators and see the section "Model Statement Usage" on page 7179 for information about how to use these operators in PROC TRANSREG. Note that nesting is not implemented in PROC TRANSREG.

The next three sections discuss the transformations available (*transforms*) (see the section "Families of Transformations" on page 7140), the transformation options (*t-options*) (see the section "Transformation Options (t-options)" on page 7147), and the algorithm options (*a-options*) (see the section "Algorithm Options (a-options)" on page 7158).

Families of Transformations

In the MODEL statement, *transform* specifies a transformation in one of the following five families:

Variable expansions	preprocess the specified variables, replacing them with more variables.
Nonoptimal transformations	preprocess the specified variables, replacing each one with a single new nonoptimal, nonlinear transformation.
Nonlinear fit transformations	preprocess the specified variable, replacing it with a smooth transformation, fitting one or more nonlinear functions through a scatter plot.
Optimal transformations	replace the specified variables with new, iteratively derived optimal transformation variables that fit the specified model better than the original variable (except for contrived cases where the transformation fits the model exactly as well as the original variable).
Other transformations	are the IDENTITY and SSPLINE transformations. These do not fit into the preceding categories.

The transformations and expansions listed in Table 90.2 are available in the MODEL statement.

Table 90.2 Transformation Families

Transformation	Description
Variable Expansions	
BSPLINE	B-spline basis
CLASS	set of coded variables
EPOINT	elliptical response surface
POINT	circular response surface & PREFMAP
PSPLINE	piecewise polynomial basis
QPOINT	quadratic response surface
Nonoptimal Transformations	
ARSIN	inverse trigonometric sine
EXP	exponential
LOG	logarithm
LOGIT	logit
POWER	raises variables to specified power
RANK	transforms to ranks
Nonlinear Fit Transformations	
BOXCOX	Box-Cox
PBSPLINE	penalized B-splines
SMOOTH	noniterative smoothing spline
Optimal Transformations	
LINEAR	linear
MONOTONE	monotonic, ties preserved
MSPLINE	monotonic B-spline

Table 90.2 *continued*

Transformation	Description
OPSCORE	optimal scoring
SPLINE	B-spline
UNTIE	monotonic, ties not preserved
Other Transformations	
IDENTITY	identity, no transformation
SSPLINE	iterative smoothing spline

You can use any transformation with either dependent or independent variables (except the SMOOTH and PBSPLINE transformations, which can be used only with independent variables, and BOXCOX, which can be used only with dependent variables). However, the variable expansions are usually more appropriate for independent variables.

The *transform* is followed by a variable (or list of variables) enclosed in parentheses. Here is an example:

```
model log(y) = class(x);
```

This example finds a LOG transformation of y and performs a CLASS expansion of x. Optionally, depending on the *transform*, the parentheses can also contain *t-options*, which follow the variables and a slash. Here is an example:

```
model identity(y) = spline(x1 x2 / nknots=3);
```

The preceding statement finds SPLINE transformations of x1 and x2. The NKNOTS= *t-option* used with the SPLINE transformation specifies three knots. The `identity(y)` transformation specifies that y is not to be transformed.

The rest of this section provides syntax details for members of the five families of transformations listed at the beginning of this section. The *t-options* are discussed in the section "Transformation Options (t-options)" on page 7147.

Variable Expansions

PROC TRANSREG performs variable expansions before iteration begins. Variable expansions expand the original variables into a typically larger set of new variables. The original variables are those that are listed in parentheses after *transform*, and they are sometimes referred to by the name of the *transform*. For example, in CLASS(x1 x2), x1 and x2 are sometimes referred to as CLASS expansion variables or simply CLASS variables, and the expanded variables are referred to as coded or sometimes "dummy" variables. Similarly, in POINT(Dim1 Dim2), Dim1 and Dim2 are sometimes referred to as POINT variables.

The resulting variables are not transformed by the iterative algorithms after the initial preprocessing. Observations with missing values for these types of variables are excluded from the analysis.

The POINT, EPOINT, and QPOINT variable expansions are used in preference mapping analyses (also called PREFMAP, external unfolding, ideal point regression) (Carroll 1972) and for response

surface regressions. These three expansions create circular, elliptical, and quadratic response or preference surfaces (see the section "Point Models" on page 7255 and Example 90.6). The CLASS variable expansion is used for main-effects ANOVA.

The following list provides syntax and details for the variable expansion *transforms*.

BSPLINE

BSP

expands each variable to a B-spline basis. You can specify the DEGREE=, KNOTS=, NKNOTS=, and EVENLY= *t-options* with the BSPLINE expansion. When DEGREE=n (3 by default) with k knots (0 by default), $n + k + 1$ variables are created. In addition, the original variable appears in the OUT= data set before the ID variables. For example, **`bspline(x)`** expands x into x_0 x_1 x_2 x_3 and outputs x as well. The x_: variables contain the B-spline basis vectors (which are the same basis vectors that the SPLINE and MSPLINE transformations use internally). The columns of the BSPLINE expansion sum to a column of ones, so an implicit intercept model is fit when the BSPLINE expansion is specified. If you specify the BSPLINE expansion for more than one variable, the model is less than full rank. Variables specified in a BSPLINE expansion must be numeric, and they are typically continuous. See the sections "SPLINE and MSPLINE Transformations" on page 7261 and "SPLINE, BSPLINE, and PSPLINE Comparisons" on page 7264 for more information about B-splines.

CLASS

CLA

expands the variables to a set of coded or "dummy" variables. PROC TRANSREG uses the values of the formatted variables to determine class membership. The specification **`class(x1 x2)`** fits a simple main-effects model, **`class(x1 | x2)`** fits a main-effects and interactions model, and **`class(x1|x2|x3|x4@2 x1*x2*x3)`** fits a model with all main effects, all two-way interactions, and one three-way interaction. Variables specified with the CLASS expansion can be either character or numeric; numeric variables should be discrete. See the section "ANOVA Codings" on page 7230 for more information about CLASS variables. See the section "Model Statement Usage" on page 7179 for information about how to use the operators @, *, and | in PROC TRANSREG.

EPOINT

EPO

expands the variables for an elliptical response surface regression or for an elliptical ideal point regression. Specify the COORDINATES *o-option* to output PREFMAP ideal elliptical point model coordinates to the OUT= data set. Each axis of the ellipse (or ellipsoid) is oriented in the same direction as one of the variables. The EPOINT expansion creates a new variable for each original variable. The value of each new variable is the square of each observed value for the corresponding original variable. The regression analysis then uses both sets of variables (original and squared). Variables specified with the EPOINT expansion must be numeric, and they are typically continuous. See the section "Point Models" on page 7255 and Example 90.6 for more information about point models.

POINT

POI

expands the variables for a circular response surface regression or for a circular ideal point regression. Specify the COORDINATES *o-option* to output PREFMAP ideal point model coordinates to the OUT= data set. The POINT expansion creates a new variable having a value for each observation that is the sum of squares of all the POINT variables. This new variable is added to the set of variables and is used in the regression analysis. For more information about ideal point regression, see Carroll (1972). Variables specified with the POINT expansion must be numeric, and they are typically continuous. See the section "Point Models" on page 7255 and Example 90.6 for more information about point models.

PSPLINE

PSP

expands each variable to a piecewise polynomial basis. You can specify the DEGREE=, KNOTS=, NKNOTS=, and EVENLY *t-options* with PSPLINE. When DEGREE=n (3 by default) with k knots (0 by default), $n + k$ variables are created. In addition, the original variable appears in the OUT= data set before the ID variables. For example, **`pspline(x / nknots=1)`** expands x into x_1 x_2 x_3 x_4 and outputs x as well. Unlike BSPLINE, an intercept is not implicit in the columns of PSPLINE. Variables specified with the PSPLINE expansion must be numeric, and they are typically continuous. See the sections "SPLINE, BSPLINE, and PSPLINE Comparisons" on page 7264 and "Using Splines and Knots" on page 7192 for more information about splines. Also see Smith (1979) for a good introduction to piecewise polynomial splines.

QPOINT

QPO

expands the variables for a quadratic response surface regression or for a quadratic ideal point regression. Specify the COORDINATES *o-option* to output PREFMAP quadratic ideal point model coordinates to the OUT= data set. For m QPOINT variables, $m(m+1)/2$ new variables are created containing the squares and crossproducts of the original variables. The regression analysis uses both sets (original and crossed). Variables specified with the QPOINT expansion must be numeric, and they are typically continuous. See the section "Point Models" on page 7255 and Example 90.6 for more information about point models.

Nonoptimal Transformations

The nonoptimal transformations, like the variable expansions, are computed before the iterative algorithm begins. Nonoptimal transformations create a single new transformed variable that replaces the original variable. The new variable is not transformed by the subsequent iterative algorithms (except for a possible linear transformation with missing value estimation). The following list provides syntax and details for nonoptimal variable transformations.

ARSIN

ARS

finds an inverse trigonometric sine transformation. Variables specified in the ARSIN *transform* must be numeric and in the interval ($-1.0 \leq x \leq 1.0$), and they are typically continuous.

EXP

exponentiates variables (x is transformed to a^x). To specify the value of a, use the PARAMETER= *t-option*. By default, a is the mathematical constant $e = 2.718\ldots$. Variables specified with the EXP *transform* must be numeric, and they are typically continuous.

LOG

transforms variables to logarithms (x is transformed to $\log_a(x)$). To specify the base of the logarithm, use the PARAMETER= *t-option*. The default is a natural logarithm with base $e = 2.718\ldots$. Variables specified with the LOG *transform* must be numeric and positive, and they are typically continuous.

LOGIT

finds a logit transformation on the variables. The logit of x is $\log(x/(1-x))$. Unlike other transformations, LOGIT does not have a three-letter abbreviation. Variables specified with the LOGIT *transform* must be numeric and in the interval $(0.0 < x < 1.0)$, and they are typically continuous.

POWER
POW

raises variables to a specified power (x is transformed to x^a). You must specify the power parameter a by specifying the PARAMETER= *t-option* following the variables. Here is an example:

```
power(variable / parameter=number)
```

You can use POWER for squaring variables (PARAMETER=2), reciprocal transformations (PARAMETER=–1), square roots (PARAMETER=0.5), and so on. Variables specified with the POWER *transform* must be numeric, and they are typically continuous.

RANK
RAN

transforms variables to ranks. Ranks are averaged within ties. The smallest input value is assigned the smallest rank. Variables specified in the RANK *transform* must be numeric.

Nonlinear Fit Transformations

Nonlinear fit transformations, like nonoptimal transformations, are computed before the iterative algorithm begins. Nonlinear fit transformations create a single new transformed variable that replaces the original variable and provides one or more smooth functions through a scatter plot. The new variable is not transformed by the subsequent iterative algorithms. The nonlinear fit transformations, unlike the nonoptimal transformations, use information in the other variables in the model to find the transformations. The nonlinear fit transformations, unlike the optimal transformations, do not minimize a squared-error criterion. The following list provides syntax and details for nonoptimal variable transformations.

BOXCOX
BOX

finds a Box-Cox (1964) transformation of the specified variables. The BOXCOX transformation can be used only with dependent variables. The ALPHA=, CLL=, CONVENIENT,

GEOMETRICMEAN, LAMBDA=, and PARAMETER= *t-options* can be used with the BOXCOX transformation. Variables specified in the BOXCOX *transform* must be numeric, and they are typically continuous. See the section "Box-Cox Transformations" on page 7181 and Example 90.2 for more information about Box-Cox transformations.

PBSPLINE

PBS

is a noniterative penalized B-spline transformation (Eilers and Marx 1996). The PBSPLINE transformation can be used only with independent variables. By default with PBSPLINE, a cubic spline is fit with 100 evenly spaced knots, three evenly spaced exterior knots, and a difference matrix of order three (DEGREE=3 NKNOTS=100 EVENLY=3 PARAMETER=3). Variables specified in the PBSPLINE *transform* must be numeric, and they are typically continuous. See the section "Penalized B-Splines" on page 7219 and Example 90.3 for more information about penalized B-splines.

SMOOTH

SMO

is a noniterative smoothing spline transformation (Reinsch 1967). You can specify the smoothing parameter with either the SM= or the PARAMETER= *t-option*. The default smoothing parameter is SM=0. The SMOOTH transformation can be used only with independent variables. Variables specified with the SMOOTH *transform* must be numeric, and they are typically continuous. How the results of the SMOOTH transformation are used in PROC TRANSREG has changed with this release. In particular, some aspects of the syntax along the coefficients and predicted values have changed. See the NSR *a-option* for a way to restore the old functionality. See the sections "Smoothing Splines" on page 7222 and "Smoothing Splines Changes and Enhancements" on page 7227 for more information about smoothing splines.

Optimal Transformations

Optimal transformations are iteratively derived. Missing values for these types of variables can be optimally estimated (see the section "Missing Values" on page 7249). The following list provides syntax and details for optimal transformations.

LINEAR

LIN

finds an optimal linear transformation of each variable. For variables with no missing values, the transformed variable is the same as the original variable. For variables with missing values, the transformed nonmissing values have a different scale and origin than the original values. Variables specified in the LINEAR *transform* must be numeric. See the section "OPSCORE, MONOTONE, UNTIE, and LINEAR Transformations" on page 7260 for more information about optimal scaling.

MONOTONE

MON

finds a monotonic transformation of each variable, with the restriction that ties are preserved. The Kruskal (1964) secondary least-squares monotonic transformation is used. This transformation weakly preserves order and category membership (ties). Variables specified with the MONOTONE *transform* must be numeric, and they are typically discrete. See the section "OPSCORE, MONOTONE, UNTIE, and LINEAR Transformations" on page 7260 for more information about optimal scaling.

MSPLINE

MSP

finds a monotonically increasing B-spline transformation with monotonic coefficients (de Boor 1978; de Leeuw 1986) of each variable. You can specify the DEGREE=, KNOTS=, NKNOTS=, and EVENLY= *t-options* with MSPLINE. By default, PROC TRANSREG fits a quadratic spline with no knots. Variables specified with the MSPLINE *transform* must be numeric, and they are typically continuous. See the section "SPLINE and MSPLINE Transformations" on page 7261 for more information about monotone splines.

OPSCORE

OPS

finds an optimal scoring of each variable. The OPSCORE transformation assigns scores to each class (level) of the variable. The Fisher (1938) optimal scoring method is used. Variables specified with the OPSCORE *transform* can be either character or numeric; numeric variables should be discrete. See the sections "Character OPSCORE Variables" on page 7253 and "OPSCORE, MONOTONE, UNTIE, and LINEAR Transformations" on page 7260 for more information about optimal scaling.

SPLINE

SPL

finds a B-spline transformation (de Boor 1978) of each variable. By default, PROC TRANSREG fits a cubic spline with no knots. You can specify the DEGREE=, KNOTS=, NKNOTS=, and EVENLY= *t-options* with SPLINE. Variables specified with the SPLINE *transform* must be numeric, and they are typically continuous. See the sections "SPLINE and MSPLINE Transformations" on page 7261, "Specifying the Number of Knots" on page 7262, and "SPLINE, BSPLINE, and PSPLINE Comparisons" on page 7264, and "Using Splines and Knots" on page 7192 for more information about splines.

UNTIE

UNT

finds a monotonic transformation of each variable without the restriction that ties are preserved. PROC TRANSREG uses the Kruskal (1964) primary least-squares monotonic transformation method. This transformation weakly preserves order but not category membership (it might untie some previously tied values). Variables specified with the UNTIE *transform* must be numeric, and they are typically discrete. See the section "OPSCORE, MONOTONE, UNTIE, and LINEAR Transformations" on page 7260 for more information about optimal scaling.

Other Transformations

IDENTITY
IDE

specifies variables that are not changed by the iterations. Typically, the IDENTITY transformation is used with a simple variable list, such as **identity(x1-x5)**. However, you can also specify interaction terms. For example, **identity(x1 | x2)** creates x1, x2, and the product x1*x2; and **identity(x1 | x2 | x3)** creates x1, x2, x1*x2, x3, x1*x3, x2*x3, and x1*x2*x3. See the section "Model Statement Usage" on page 7179 for information about how to use the operators @, *, and | in PROC TRANSREG. Variables specified in the IDENTITY *transform* must be numeric.

The IDENTITY transformation is used for variables when no transformation and no missing data estimation are desired. However, the REFLECT *t-option*, the ADDITIVE *a-option*, and the TSTANDARD=Z, and TSTANDARD=CENTER options can linearly transform all variables, including IDENTITY variables, after the iterations. Observations with missing values in IDENTITY variables are excluded from the analysis, and no optimal scores are computed for missing values in IDENTITY variables.

SSPLINE
SSP

finds an iterative smoothing spline transformation of each variable. The SSPLINE transformation does not generally minimize squared error. You can specify the smoothing parameter with either the SM= *t-option* or the PARAMETER= *t-option*. The default smoothing parameter is SM=0. Variables specified with the SSPLINE *transform* must be numeric, and they are typically continuous. How the results of the SSPLINE transformation are used in PROC TRANSREG has changed with this release to be consistent with the changes to SMOOTH. In particular, some aspects of the syntax along the coefficients and predicted values have changed. See the section "Smoothing Splines Changes and Enhancements" on page 7227 for more information about the changes, and see the NSR *a-option* for a way to restore the old functionality.

Transformation Options (t-options)

If you use a nonoptimal, nonlinear fit, optimal, or other transformation, you can use *t-options*, which specify additional details of the transformation. The *t-options* are specified within the parentheses that enclose variables and are listed after a slash. You can use *t-options* with both the dependent and the independent variables. Here is an example of using just one *t-option*:

```
proc transreg;
   model identity(y)=spline(x / nknots=3);
   output;
run;
```

The preceding statements find an optimal variable transformation (SPLINE) of the independent variable, and they use a *t-option* to specify the number of knots (NKNOTS=). The following is a more complex example:

```
proc transreg;
   model mspline(y / nknots=3)=class(x1 x2 / effects);
   output;
run;
```

These statements find a monotone spline transformation (MSPLINE with three knots) of the dependent variable and perform a CLASS expansion with effects coding of the independents.

The *t-options* listed in Table 90.3 are available in the MODEL statement.

Table 90.3 Transformation Options

Option	Description
Nonoptimal Transformation	
ORIGINAL	uses original mean and variance
Parameter Specification	
PARAMETER=	specifies miscellaneous parameters
SM=	specifies smoothing parameter
Penalized B-Spline	
AIC	uses Akaike's information criterion
AICC	uses corrected AIC
CV	uses cross validation criterion
GCV	uses generalized cross validation criterion
LAMBDA=	specifies smoothing parameter list or range
RANGE	specifies a LAMBDA= range, not a list
SBC	uses Schwarz's Bayesian criterion
Spline	
DEGREE=	specifies the degree of the spline
EVENLY=	spaces the knots evenly
EXKNOTS=	specifies exterior knots
KNOTS=	specifies the interior knots or break points
NKNOTS=	creates *n* knots
CLASS Variable	
CPREFIX=	specifies CLASS coded variable name prefix
DEVIATIONS	specifies a deviations-from-means coding
EFFECTS	specifies a deviations-from-means coding
LPREFIX=	specifies CLASS coded variable label prefix
ORDER=	specifies order of CLASS variable levels
ORTHOGONAL	specifies an orthogonal-contrast coding
SEPARATORS=	specifies CLASS coded variable label separators
STANDORTH	specifies a standardized-orthogonal coding
ZERO=	controls reference levels
Box-Cox	
ALPHA=	specifies confidence interval alpha
CLL=	specifies convenient lambda list
CONVENIENT	uses a convenient lambda

Table 90.3 *continued*

Option	Description
GEOMETRICMEAN	scales transformation using geometric mean
LAMBDA=	specifies power parameter list
Other t-options	
AFTER	specifies operations occur after the expansion
CENTER	specifies center before the analysis begins
NAME=	renames variables
REFLECT	reflects the variable around the mean
TSTANDARD=	specifies transformation standardization
Z	standardizes before the analysis begins

The following sections discuss the *t-options* available for nonoptimal, nonlinear fit, optimal, and other transformations.

Nonoptimal Transformation t-options

ORIGINAL

ORI

matches the variable's final mean and variance to the mean and variance of the original variable. By default, the mean and variance are based on the transformed values. The ORIGINAL *t-option* is available for all of the nonoptimal transformations.

Parameter t-options

PARAMETER=*number*

PAR=*number*

specifies the transformation parameter. The PARAMETER= *t-option* is available for the BOXCOX, EXP, LOG, POWER, SMOOTH, SSPLINE, and PBSPLINE transformations. For BOXCOX, the parameter is the value to add to each value of the variable before a Box-Cox transformation. For EXP, the parameter is the value to be exponentiated; for LOG, the parameter is the base value; and for POWER, the parameter is the power. For SMOOTH and SSPLINE, the parameter is the raw smoothing parameter. (See the SM= option for an alternative way to specify the smoothing parameter.) The default for the PARAMETER= *t-option* for the BOXCOX transformation is 0 and for the LOG and EXP transformations is $e = 2.718\ldots$. The default parameter for SMOOTH and SSPLINE is computed from SM=0. For the POWER transformation, you must specify the PARAMETER= *t-option*; there is no default. For PBSPLINE, the parameter is the order of the difference matrix, which provides some control over the smoothness of the transformation. The default order parameter with PBSPLINE is the maximum of the DEGREE= *t-option*, and 1. With PBSPLINE, the default is DEGREE=3 and PARAMETER=3, which works well for most problems.

SM=*n*

specifies a smoothing parameter in the range 0 to 100, just like PROC GPLOT uses. For example, SM=50 in PROC TRANSREG is equivalent to I=SM50 in the SYMBOL statement with PROC GPLOT. You can specify the SM= *t-option* only with the SMOOTH and SSPLINE transformations. The smoothness of the function increases as the value of the smoothing parameter increases. By default, SM=0.

Spline t-options

The following *t-options* are available with the SPLINE, MSPLINE and PBSPLINE transformations and with the PSPLINE and BSPLINE expansions.

DEGREE=*n*

DEG=*n*

specifies the degree of the spline transformation. The degree must be a nonnegative integer. The defaults are DEGREE=3 for SPLINE, PSPLINE, and BSPLINE variables and DEGREE=2 for MSPLINE variables.

The polynomial degree should be a small integer, usually 0, 1, 2, or 3. Larger values are rarely useful. If you have any doubt as to what degree to specify, use the default.

EVENLY<=*n***>**

EVE<=*n***>**

is used with the NKNOTS= *t-option* to space the knots evenly. The differences between adjacent knots are constant.

If you specify NKNOTS=k and EVENLY, k knots are created at

$$\text{minimum} + i((\text{maximum} - \text{minimum})/(k+1))$$

for $i = 1, \ldots, k$. Here is an example:

```
spline(x / nknots=2 evenly)
```

When the variable x has a minimum of 4 and a maximum of 10, then the two interior knots are 6 and 8. Without the EVENLY *t-option*, the NKNOTS= *t-option* places knots at percentiles, so the knots are not evenly spaced. By default for the BSPLINE expansion and the SPLINE and MSPLINE transformations, the smaller exterior knots are all the same and all just a little smaller than the minimum. Similarly, by default, the larger exterior knots are all the same and all just a little larger than the maximum. However, if you specify EVENLY=*n*, then the n exterior knots are evenly spaced as well. The number of exterior knots must be greater than or equal to the degree. You can specify values larger than the degree when you want to interpolate slightly beyond the range or your data. The exterior knots must be less than the minimum or greater than the maximum; hence the knots across all sets are not precisely equally spaced. For example, with data ranging from 0 to 10, and with EVENLY=3 and NKNOTS=4, the first exterior knots are –4.000000000001, –2.000000000001, and –0.000000000001, the interior knots are 2, 4, 6, and 8, and the second exterior knots are 10.000000000001, 12.000000000001, and 14.000000000001.

With the BSPLINE and PSPLINE expansions and the SPLINE and MSPLINE transformations, evenly spaced knots are not the default. With the PBSPLINE transformation, evenly spaced interior and exterior knots are the default. If you want unevenly spaced knots with PBSPLINE, you must use the KNOTS= *t-option*.

EXKNOTS=*number-list*

EXK=*number-list*

specifies exterior knots for SPLINE and MSPLINE transformations and BSPLINE expansions. Usually, this *t-option* is not needed; PROC TRANSREG automatically picks suitable exterior knots. The only time you need to use this option is when you want to ensure that the exact same basis is used for different splines, such as when you apply coefficients from one spline transformation to a variable in a different data set (see the section "Scoring Spline Variables" on page 7204).

Specify one or two values. If the minimum EXKNOTS= value is less than the minimum data value, it is used as the exterior knot. If the maximum EXKNOTS= value is greater than the maximum data value, it is used as the exterior knot. Otherwise these values are ignored. When EXKNOTS= is specified with the CENTER or Z *t-options*, the knots apply to the original variable, not to the centered or standardized variable.

The B-spline transformations and expansions use a knot list consisting of exterior knots (values just smaller than the minimum), the specified (interior) knots, and exterior knots (values just larger than the minimum). You can use the DETAIL *a-option* to see all of these knots. If you use different exterior knots, you get different but equivalent B-spline bases. You can specify exterior knots in either the KNOTS= or EXKNOTS= *t-options*; however, for the BSPLINE expansion, the KNOTS= *t-option* creates extra all-zero basis columns, whereas the EXKNOTS= *t-option* gives you the correct basis. See the EVENLY= *t-option* for an alternative way to specify exterior knots.

KNOTS=*number-list* | *n* **TO** *m* **BY** *p*

KNO=*number-list* | *n* **TO** *m* **BY** *p*

specifies the interior knots or break points. By default, there are no knots. The first time you specify a value in the knot list, it indicates a discontinuity in the nth (from DEGREE=n) derivative of the transformation function at the value of the knot. The second mention of a value indicates a discontinuity in the $(n-1)$th derivative of the transformation function at the value of the knot. Knots can be repeated any number of times for decreasing smoothness at the break points, but the values in the knot list can never decrease.

You cannot use the KNOTS= *t-option* with the NKNOTS= *t-option*. You should keep the number of knots small (see the section "Specifying the Number of Knots" on page 7262).

NKNOTS=*n*

NKN=*n*

creates n knots, the first at the $100/(n+1)$ percentile, the second at the $200/(n+1)$ percentile, and so on. Knots are always placed at data values; there is no interpolation. For example, if NKNOTS=3, knots are placed at the 25th percentile, the median, and the 75th percentile. You can use the EVENLY= *t-option* along with NKNOTS= to get evenly spaced knots. By default, with the BSPLINE and PSPLINE expansions and the SPLINE and MSPLINE transformations, NKNOTS=0. By default, with the PBSPLINE transformation, NKNOTS=100.

The value specified for the NKNOTS= *t-option* must be ≥ 0.

You cannot use the NKNOTS= *t-option* with the KNOTS= *t-option*.

You should keep the number of knots small (see the section "Specifying the Number of Knots" on page 7262).

Penalized B-Spline t-options

The following *t-options* are available with the PBSPLINE transformation.

AIC

specifies that the procedure should select the smoothing parameter, λ, that minimizes the (Akaike 1973) information criterion (AIC). By default, the (AICC) criterion is minimized.

AICC

specifies that the procedure should select the smoothing parameter, λ, that minimizes the corrected Akaike information criterion (Hurvich, Simonoff, and Tsai 1998). This is the default criterion unless the AIC, CV, GCV, or SBC *t-option* is specified.

CV

specifies that the procedure should select the smoothing parameter, λ, that minimizes the cross validation criterion (CV). By default, the (AICC) criterion is minimized.

GCV

specifies that the procedure should select the smoothing parameter, λ, that minimizes the generalized cross validation criterion (Craven and Wahba 1979). By default, the (AICC) criterion is minimized.

LAMBDA=*number-list*

LAM=*number-list*

specifies a list of penalized B-spline smoothing parameters. By default, PROC TRANSREG considers lambdas in the range 0 to 1E6. Alternatively, you can specify the RANGE *t-option* with LAMBDA=, such as LAMBDA=1E3 1E5 RANGE, to only consider lambdas in a narrower range. Note that the algorithm might not actually evaluate the criterion at the minimum and maximum if it does not have to. In particular, it avoids evaluating the criterion at LAMBDA=0 (no smoothing) unless it is the only LAMBDA= value specified. You can also specify a list of lambdas, such as LAMBDA=1 TO 10, and the procedure selects the best lambda from the list. In all cases, the lambda that minimizes the specified criterion (or AICC by default) is chosen.

RANGE

RAN

specifies that the LAMBDA= *t-option* specifies two lambdas that define a range of values, from which an optimal lambda is selected. By default, PROC TRANSREG considers lambdas in the range 0 to 1E6.

SBC

specifies that the procedure should select the smoothing parameter, λ, that minimizes Schwarz's Bayesian criterion (Schwarz 1978; Judge et al. 1980). By default, the (AICC) criterion is minimized.

Class Variable t-options

CPREFIX=*n* | *number-list*

CPR=*n* | *number-list*

specifies the number of first characters of a CLASS expansion variable's name to use in constructing names for coded variables. When you specify CPREFIX= as an *a-option* or an *o-option*, it specifies the default for all CLASS variables. When you specify CPREFIX= as a *t-option*, it overrides the default only for selected variables. A different CPREFIX= value can be specified for each CLASS variable by specifying the CPREFIX=number-list *t-option*, like the ZERO=*formatted-value t-option*.

DEVIATIONS

DEV

requests a deviations-from-means coding of CLASS variables. The coded design matrix has values of 0, 1, and –1 for reference levels. This coding is referred to as "deviations-from-means," "effects," "center-point," or "full-rank" coding. For example, here is the coding for two-, three-, four-, and five-level factors:

	Number of Levels									
	Two	Three		Four			Five			
a	1	1	0	1	0	0	1	0	0	0
b	-1	0	1	0	1	0	0	1	0	0
c		-1	-1	0	0	1	0	0	1	0
d				-1	-1	-1	0	0	0	1
e							-1	-1	-1	-1

EFFECTS

EFF

See the DEVIATIONS *t-option*.

LPREFIX=*n* | *number-list*

LPR=*n* | *number-list*

specifies the number of first characters of a CLASS expansion variable's label (or name if no label is specified) to use in constructing labels for the coded variables. When you specify LPREFIX= as an *a-option* or an *o-option*, it specifies the default for all CLASS variables. When you specify LPREFIX= as a *t-option*, it overrides the default only for selected variables. A different LPREFIX= value can be specified for each CLASS variable by specifying the LPREFIX=number-list *t-option*, like the ZERO=*formatted-value t-option*.

ORDER=DATA | FREQ | FORMATTED | INTERNAL

ORD=DAT | FRE | FOR | INT

specifies the order in which the CLASS variable levels are to be reported. The default is ORDER=INTERNAL. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine dependent. When you specify ORDER= as an *a-option* or an *o-option*, it specifies the default ordering for all CLASS variables. When you specify ORDER= as a *t-option*, it overrides the default ordering only for selected variables. You can specify a different ORDER= value for each CLASS specification.

ORTHOGONAL

ORT

requests an orthogonal-contrast coding of CLASS variables. For example, here is the orthogonal-contrast coding for two-, three-, four-, and five-level factors:

	Number of Levels									
	Two	Three		Four			Five			
a	1	1	-1	1	-1	-1	1	-1	-1	-1
b	-1	0	2	0	2	-1	0	2	-1	-1
c		-1	-1	0	0	3	0	0	3	-1
d				-1	-1	-1	0	0	0	4
e							-1	-1	-1	-1

The sum of the coded values within each column is zero, all columns within a factor are orthogonal, and the ith column represents a contrast between the ith level and the combination of all preceding levels and the last level. The $\mathbf{X}$ matrix is orthogonal and $\mathbf{X}'\mathbf{X}$ is diagonal with this coding only if the experimental design is orthogonal.

SEPARATORS='*string-1***' <'***string-2***'>**

SEP='*string-1***' <'***string-2***'>**

specifies separators for creating CLASS expansion variable labels. By default, SEPARATORS=' ' * ' ("blank" and "blank asterisk blank"). When you specify SEPARATORS= as an *a-option* or an *o-option*, it specifies the default separators for all CLASS variables. When you specify SEPARATORS= as a *t-option*, it overrides the default only for selected variables. You can specify a different SEPARATORS= value for each CLASS specification.

STANDORTH

STA

ORTHEFFECT

requests a standardized-orthogonal coding of CLASS variables. For example, here is the standardized-orthogonal coding for two-, three-, four-, and five-level factors:

	Number of Levels									
	Two	Three		Four			Five			
a	1	1.22	-0.71	1.41	-0.82	-0.58	1.58	-0.91	-0.65	-0.50
b	-1	0.00	1.41	0.00	1.63	-0.58	0.00	1.83	-0.65	-0.50
c		-1.22	-0.71	0.00	0.00	1.73	0.00	0.00	1.94	-0.50
d				-1.41	-0.82	-0.58	0.00	0.00	0.00	2.00
e							-1.58	-0.91	-0.65	-0.50

The sum of the coded values within each column is zero, the sum of squares of the coded values within each column is equal to the number of levels, all columns within a factor are orthogonal, and the *i*th column represents a contrast between the *i*th level and the combination of all preceding levels and the last level. The $\mathbf{X}$ matrix is orthogonal and $\mathbf{X}'\mathbf{X}$ is diagonal ($\mathbf{X}'\mathbf{X} = n\mathbf{I}$, the number of observations times an identity matrix) with this coding only if the experimental design is orthogonal.

ZERO=FIRST | LAST | NONE | SUM

ZER=FIR | LAS | NON | SUM

ZERO='*formatted-value***' <'***formatted-value***' ... >**

is used with CLASS variables. The default is ZERO=LAST.

The specification CLASS(variable / ZERO=FIRST) sets to missing the coded variable for the first of the sorted categories, implying a zero coefficient for that category.

The specification CLASS(variable / ZERO=LAST) sets to missing the coded variable for the last of the sorted categories, implying a zero coefficient for that category.

The specification CLASS(variable / ZERO='*formatted-value*') sets to missing the coded variable for the category with a formatted value that matches '*formatted-value*', implying a zero coefficient for that category. With ZERO=*formatted-value*, the first formatted value applies to the first variable in the specification, the second formatted value applies to the next variable that was not previously mentioned, and so on. For example, `class(a a*b b b*c c / zero='x' 'y' 'z')` specifies that the reference level for a is 'x', for b is 'y', and for c is 'z'. With ZERO='*formatted-value*', the procedure first looks for exact matches between the formatted values and the specified value. If none are found, leading blanks are stripped from both and the values are compared again. If zero or two or more matches are found, warnings are issued.

The specifications ZERO=FIRST, ZERO=LAST, and ZERO='*formatted-value*' are used for reference cell models. The Intercept parameter estimate is the marginal mean for the reference cell, and the other marginal means are obtained by adding the intercept to the coded variable coefficients.

The specification CLASS(variable / ZERO=NONE) sets to missing none of the coded variables. The columns of the expansion sum to a column of ones, so an implicit intercept model is fit. If you specify ZERO=NONE for more than one variable, the model is less than full rank. In the model `model identity(y) = class(x / zero=none)`, the coefficients are cell means.

The specification CLASS(variable / ZERO=SUM) sets to missing none of the coded variables, and the coefficients for the coded variables created from the variable sum to 0. This creates a less-than-full-rank model, but the coefficients are uniquely determined due to the sum-to-zero constraint.

In the presence of iterative transformations, hypothesis tests for ZERO=NONE and ZERO=SUM levels are not exact; they are liberal because a model with an explicit intercept is fit inside the iterations. There is no provision for adjusting the transformations while setting to 0 a parameter that is redundant given the explicit intercept and the other parameters.

Box-Cox t-options

The following *t-options* are available only with the BOXCOX transformation of the dependent variable (see the section "Box-Cox Transformations" on page 7181 and Example 90.2).

ALPHA=*p*

ALP=*p*

specifies the Box-Cox alpha for the confidence interval for the power parameter. By default, ALPHA=0.05.

CLL=*number-list*

specifies the Box-Cox convenient lambda list. When the confidence interval for the power parameter includes one of the values in this list, PROC TRANSREG reports it and can optionally use the convenient power parameter instead of the more optimal power parameter. The default is CLL=1.0 0.0 0.5 –1.0 –0.5 2.0 –2.0 3.0 –3.0. By default, a linear transformation is preferred over log, square root, inverse, inverse square root, quadratic, inverse quadratic, cubic, and inverse cubic. If you specify the CONVENIENT *t-option*, then PROC TRANSREG uses the first convenient power parameter in the list that is in the confidence interval. For example, if the optimal power parameter is 0.25 and 0.0 is in the confidence interval but not 1.0, then the convenient power parameter is 0.0.

CONVENIENT

CON

specifies that a power parameter from the CLL= *t-option* list is to be used for the final transformation instead of the LAMBDA= *t-option* value if a CLL= value is in the confidence interval. See the CLL= *t-option* for more information about its usage.

GEOMETRICMEAN

GEO

divides the Box-Cox transformation by $\dot{y}^{\lambda-1}$, where $\dot{y}$ is the geometric mean of the variable to be transformed. This form of the Box-Cox transformation essentially converts the transformation back to original units, and hence it permits direct comparison of the residual sums of squares for models with different power parameters.

LAMBDA=*number-list*

LAM=*number-list*

specifies a list of Box-Cox power parameters. The default is LAMBDA=–3 TO 3 BY 0.25. PROC TRANSREG tries each power parameter in the list and picks the best one. However, when the CONVENIENT *t-option* is specified, PROC TRANSREG chooses a convenient value from the confidence interval instead of the optimal value. For example, if the optimal power parameter is 0.25 and 0.0 is in the confidence interval but not 1.0, then the convenient power parameter 0.0 (log transformation) is chosen instead of the more optimal parameter 0.25. See the CLL= *t-option* for more information about its usage.

Other t-options

AFTER
AFT

requests that certain operations occur after the expansion. This *t-option* affects the NKNOTS= *t-option* when the SPLINE or MSPLINE transformation is crossed with a CLASS specification. For example, if the original spline variable (1 2 3 4 5 6 7 8 9) is expanded into the three variables (1 2 3 0 0 0 0 0 0), (0 0 0 4 5 6 0 0 0), and (0 0 0 0 0 0 7 8 9), then, by default, NKNOTS=1 would use the overall median of 5 as the knot for all three variables. When you specify the AFTER *t-option*, the knots for the three variables are 2, 5, and 8. Note that the structural zeros are ignored when the internal knot list is created, but they are not ignored for the exterior knots.

You can also specify the AFTER *t-option* with the RANK, SMOOTH, and PBSPLINE transformations. The following specifications compute ranks and smooth transformations within groups, after crossing, ignoring the structural zeros:

```
class(x / zero=none) | rank(z / after)
class(x / zero=none) | smooth(z / after)
```

CENTER
CEN

centers the variables before the analysis begins (in contrast to the TSTANDARD=CENTER option, which centers after the analysis ends). The CENTER *t-option* can be used instead of running PROC STANDARD before PROC TRANSREG (see the section "Centering" on page 7298). When the KNOTS= *t-option* is specified with CENTER, the knots apply to the original variable, not to the centered variable. PROC TRANSREG centers the knots.

NAME=(*variable-list*)
NAM=(*variable-list*)

renames variables as they are used in the MODEL statement. This *t-option* lets you use a variable more than once.

For example, if x is a character variable, then the following step stores both the original character variable x and a numeric variable xc that contains category numbers in the OUT= data set:

```
proc transreg data=a;
   model identity(y) = opscore(x / name=(xc));
   output;
   id x;
run;
```

With the CLASS and IDENTITY transformations, which can contain interaction effects, the first name applies to the first variable in the specification, the second name applies to the next variable that was not previously mentioned, and so on. For example, **identity(a a * b b b * c c / name=(g h i))** specifies that the new name for a is g, for b is h, and for c is i. The same assignment is used for the (not useful) specification **identity(a a b b c c / name=(g h i))**. For all *transforms* other than CLASS and IDENTITY (all those in which interactions are not supported), repeated variables are not handled specially. For example,

spline(a a b b c c / name=(a g b h c i)) creates six variables: a copy of a named a, another copy of a named g, a copy of b named b, another copy of b named h, a copy of c named c, and another copy of c named i.

REFLECT

REF

reflects the transformation

$$y = -(y - \bar{y}) + \bar{y}$$

after the iterations are completed and before the final standardization and results calculations. This *t-option* is particularly useful with the dependent variable in a conjoint analysis. When the dependent variable consists of ranks with the most preferred combination assigned 1.0, the REFLECT *t-option* reflects the transformation so that positive utilities mean high preference. (See Example 90.4.)

TSTANDARD=CENTER | NOMISS | ORIGINAL | Z

TST=CEN | NOM | ORI | Z

specifies the standardization of the transformed variables for the hypothesis tests and in the OUT= data set (see the section "Centering" on page 7298). By default, TSTANDARD=ORIGINAL. When you specify TSTANDARD= as an *a-option* or an *o-option*, it determines the default standardization for all variables. When you specify TSTANDARD= as a *t-option*, it overrides the default standardization only for selected variables. You can specify a different TSTANDARD= value for each transformation. For example, to perform a redundancy analysis with standardized dependent variables, specify the following:

```
model identity(y1-y4 / tstandard=z) = identity(x1-x10);
```

Z

centers and standardizes the variables to variance one before the analysis begins (in contrast to the TSTANDARD=Z option, which standardizes after the analysis ends). The Z *t-option* can be used instead of running PROC STANDARD before PROC TRANSREG (see the section "Centering" on page 7298). When the KNOTS= *t-option* is specified with Z, the knots apply to the original variable, not to the standardized variable. PROC TRANSREG standardizes the knots.

Algorithm Options (a-options)

This section discusses the options that can appear in the PROC TRANSREG or MODEL statement as *a-options*. They are listed after the entire model specification and after a slash. Here is an example:

```
proc transreg;
   model spline(y / nknots=3)=log(x1 x2 / parameter=2)
         / nomiss maxiter=50;
   output;
run;
```

In the preceding statements, NOMISS and MAXITER= are *a-options*. (SPLINE and LOG are *transforms*, and NKNOTS= and PARAMETER= are *t-options*.) The statements find a spline transformation with 3 knots on y and a base 2 logarithmic transformation on x1 and x2. The NOMISS *a-option* excludes all observations with missing values, and the MAXITER= *a-option* specifies the maximum number of iterations.

The *a-options* listed in Table 90.4 are available in the PROC TRANSREG or MODEL statement.

Table 90.4 Options Available in the PROC TRANSREG or MODEL Statement

Option	Description
Input Control	
REITERATE	restarts iterations
TYPE=	specifies input observation type
Method and Iterations	
CCONVERGE=	specifies minimum criterion change
CONVERGE=	specifies minimum data change
MAXITER=	specifies maximum number of iterations
METHOD=	specifies iterative algorithm
NCAN=	specifies number of canonical variables
NSR	specifies no restrictions on smoothing models
SINGULAR=	specifies singularity criterion
SOLVE	attempts direct solution instead of iteration
Missing Data Handling	
INDIVIDUAL	fits each model individually (METHOD=MORALS)
MONOTONE=	includes monotone special missing values
NOMISS	excludes observations with missing values
UNTIE=	unties special missing values
Intercept and CLASS Variables	
CPREFIX=	specifies CLASS coded variable name prefix
LPREFIX=	specifies CLASS coded variable label prefix
NOINT	specifies no intercept or centering
ORDER=	specifies order of CLASS variable levels
REFERENCE=	controls output of reference levels
SEPARATORS=	specifies CLASS coded variable label separators
Control Displayed Output	
ALPHA=	specifies confidence limits alpha
CL	displays parameter estimate confidence limits
DETAIL	displays model specification details
HISTORY	displays iteration histories
NOPRINT	suppresses displayed output
PBOXCOXTABLE	prints the Box-Cox log likelihood table
RSQUARE	displays the R square
SHORT	suppresses the iteration histories
SS2	displays regression results
TEST	displays ANOVA table

Table 90.4 *continued*

Option	Description
TSUFFIX=	shortens transformed variable labels
UTILITIES	displays conjoint part-worth utilities
Standardization	
ADDITIVE	fits additive model
NOZEROCONSTANT	does not zero constant variables
TSTANDARD=	specifies transformation standardization

The following list provides details about these *a-options*. The *a-options* are available in the PROC TRANSREG or MODEL statement.

ADDITIVE

ADD

creates an additive model by multiplying the values of each independent variable (after the TSTANDARD= standardization) by that variable's corresponding multiple regression coefficient. This process scales the independent variables so that the predicted-values variable for the final dependent variable is simply the sum of the final independent variables. An additive model is a univariate multiple regression model. As a result, the ADDITIVE *a-option* is not valid if METHOD=CANALS, or if METHOD=REDUNDANCY or METHOD=UNIVARIATE with more than one dependent variable.

ALPHA=*number*

ALP=*number*

specifies the level of significance for all of the confidence limits. By default, ALPHA=0.05.

CCONVERGE=*n*

CCO=*n*

specifies the minimum change in the criterion being optimized (squared multiple correlation for METHOD=MORALS and METHOD=UNIVARIATE, average squared multiple correlation for METHOD=REDUNDANCY, average squared canonical correlation for METHOD=CANALS) that is required to continue iterating. By default, CCONVERGE=0.0.

CL

requests confidence limits on the parameter estimates in the displayed output.

CONVERGE=*n*

CON=*n*

specifies the minimum average absolute change in standardized variable scores that is required to continue iterating. By default, CONVERGE=0.00001. Average change is computed over only those variables that can be transformed by the iterations; that is, all LINEAR, OPSCORE, MONOTONE, UNTIE, SPLINE, MSPLINE, and SSPLINE variables and nonoptimal transformation variables with missing values.

CPREFIX=*n*

CPR=*n*

specifies the number of first characters of a CLASS expansion variable's name to use in constructing names for coded variables. Coded variable names are constructed from the first n characters of the CLASS expansion variable's name and the first $32 - n$ characters of the formatted CLASS expansion variable's value. For example, if the variable ClassVariable has values 1, 2, and 3, then, by default, the coded variables are named ClassVariable1, ClassVariable2, and ClassVariable3. However, with CPREFIX=5, the coded variables are named Class1, Class2, and Class3. When CPREFIX=0, coded variable names are created entirely from the CLASS expansion variable's formatted values. Valid values range from –1 to 31, where –1 indicates the default calculation and 0 to 31 are the number of prefix characters to use. The default, –1, sets n to 32 – min(32, max(2, *fl*)), where *fl* is the format length. When you specify CPREFIX= as an *a-option* or an *o-option*, it specifies the default for all CLASS variables. When you specify CPREFIX= as a *t-option*, it overrides the default only for selected variables.

DETAIL

DET

reports on details of the model specification. For example, it reports the knots and coefficients for splines, reference levels for CLASS variables, Box-Cox results, the smoothing parameter, and so on. The DETAIL option can take two optional suboptions, NOCOEFFICIENTS and NOKNOTS (or NOC and NOK). To suppress knots from the details listing, specify DETAIL(NOKNOTS). To suppress coefficients from the details listing, specify DETAIL(NOCOEFFICIENTS). To suppress both knots and coefficients from the details listing, specify DETAIL(NOKNOTS NOCOEFFICIENTS).

SOLVE

SOL

DUMMY

DUM

provides a canonical initialization. When there are no monotonicity constraints, when there is at most one canonical variable in each set, and when there is enough available memory, PROC TRANSREG (with the SOLVE *a-option*) can usually directly solve for the optimal solution in only one iteration. The initialization iteration is number 0, which is slower and uses more memory than other iterations. However, for some models, specifying the SOLVE *a-option* can greatly decrease the amount of time required to find the optimal transformations. During iteration 0, each variable is replaced by an expanded variable and the model is fit to the larger, expanded set of variables. For example, an OPSCORE variable is expanded into coded (or "dummy") variables, as if CLASS were specified, and a SPLINE variable is expanded into a B-spline basis, as if BSPLINE were specified. Then for each expanded variable, the results of iteration zero are constructed by multiplying the expanded basis times the $\boldsymbol{\beta}$ subvector to get the optimal transformation. This *a-option* can be useful even in models where a direct solution is not possible, because it provides good initial transformations of all the variables.

HISTORY

HIS

displays the iteration histories even when the NOPRINT *a-option* is specified.

INDIVIDUAL

IND

fits each model for each dependent variable individually. This means, for example, that when INDIVIDUAL is specified, missing values in one dependent variable will not cause that observation to be deleted for the other models with the other dependent variables. In contrast, by default, missing values in any variable in any model can cause the observation to be deleted for all models. The INDIVIDUAL *a-option* can be specified only with METHOD=MORALS.

This *a-option* also affects the order of the output. By default, the number of observations table is printed once at the beginning of the output. With INDIVIDUAL, a number of observations table appears for each model.

LPREFIX=*n*

LPR=*n*

specifies the number of first characters of a CLASS expansion variable's label (or name if no label is specified) to use in constructing labels for coded variables. Coded variable labels are constructed from the first n characters of the CLASS expansion variable's name and the first $127 - n$ characters of the formatted CLASS expansion variable's value. Valid values range from –1 to 127. Values of 0 to 127 specify the number of name or label characters to use. The default is –1, which specifies that PROC TRANSREG should pick a value depending on the length of the prefix and the formatted class value. When you specify LPREFIX= as an *a-option* or an *o-option*, it determines the default for all CLASS variables. When you specify LPREFIX= as a *t-option*, it overrides the default only for selected variables.

MAXITER=*n*

MAX=*n*

specifies the maximum number of iterations (see the section "Controlling the Number of Iterations" on page 7251). By default, MAXITER=30. You can specify MAXITER=0 to save time when no transformations are requested.

METHOD=CANALS | MORALS | REDUNDANCY | UNIVARIATE

MET=CAN | MOR | RED | UNI

specifies the iterative algorithm. By default, METHOD=UNIVARIATE, unless you specify options that cannot be handled by the UNIVARIATE algorithm. Specifically, the default is METHOD=MORALS for the following situations:

- if you specify LINEAR, OPSCORE, MONOTONE, UNTIE, SPLINE, MSPLINE, or SSPLINE transformations for the independent variables
- if you specify the ADDITIVE *a-option* with more than one dependent variable
- if you specify the IAPPROXIMATIONS *o-option*
- if you specify the INDIVIDUAL *a-option*
- if ODS Graphics is enabled, regression plots are produced, and there is more than one dependent variable

CANALS specifies canonical correlation with alternating least squares. This jointly transforms all dependent and independent variables to maximize the average of the first *n* squared canonical correlations, where *n* is the value of the NCAN= *a-option*.

MORALS specifies multiple optimal regression with alternating least squares. This transforms each dependent variable, along with the set of independent variables, to maximize the squared multiple correlation.

REDUNDANCY jointly transforms all dependent and independent variables to maximize the average of the squared multiple correlations (see the section "Redundancy Analysis" on page 7256).

UNIVARIATE transforms each dependent variable to maximize the squared multiple correlation, while the independent variables are not transformed.

MONOTONE=*two-letters*

MON=*two-letters*

specifies the first and last special missing value in the list of those special missing values to be estimated with within-variable order and category constraints. By default, there are no order constraints on missing value estimates. The *two-letters* value must consist of two letters in alphabetical order. For example, MONOTONE=DF means that the estimate of .D must be less than or equal to the estimate of .E, which must be less than or equal to the estimate of .F; no order constraints are placed on estimates of ._, .A through .C, and .G through .Z. For details, see the section "Missing Values" on page 7249.

NCAN=*n*

NCA=*n*

specifies the number of canonical variables to use in the METHOD=CANALS algorithm. By default, NCAN=1. The value of the NCAN= *a-option* must be ≥ 1.

When canonical coefficients and coordinates are included in the OUT= data set, the NCAN= *a-option* also controls the number of rows of the canonical coefficient matrices in the data set. If you specify an NCAN= value larger than the minimum of the number of dependent variables and the number of independent variables, PROC TRANSREG displays a warning and sets the NCAN= *a-option* to the maximum value.

NOINT

NOI

omits the intercept from the OUT= data set and suppresses centering of data. You cannot specify the NOINT *a-option* with iterative transformations since there is no provision for optimal scaling without an intercept. The NOINT *a-option* can be specified only when there is no implicit intercept and when all of the data in a BY group absolutely will not change during the iterations.

NOMISS

NOM

excludes all observations with missing values from the analysis, but does not exclude them from the OUT= data set. If you omit the NOMISS *a-option*, PROC TRANSREG simultaneously computes the optimal transformations of the nonmissing values and estimates the missing values that minimize squared error. For details, see the section "Missing Values" on page 7249.

Casewise deletion of observations with missing values occurs when the NOMISS *a-option* is specified, when there are missing values in expansions, when there are missing values in METHOD=UNIVARIATE independent variables, when there are weights less than or equal to 0, or when there are frequencies less than 1. Excluded observations are output with a blank value for the _TYPE_ variable, and they have a weight of 0. They do not contribute to the analysis but are scored and transformed as *supplementary* or passive observations.

See the section "Passive Observations" on page 7255 for more information about excluded observations.

NOPRINT

NOP

suppresses the display of all output unless you specify the HISTORY *a-option*. The NOPRINT *a-option* without the HISTORY *a-option* disables the Output Delivery System (ODS), including ODS Graphics, for the duration of the procedure run. The NOPRINT *a-option* with the HISTORY *a-option* disables all output except the iteration history, again including ODS Graphics, for the duration of the procedure run. For more information, see Chapter 20, "Using the Output Delivery System."

NOZEROCONSTANT

NOZERO

NOZ

specifies that constant variables are expected and should not be zeroed. By default, constant variables are zeroed. This option is useful when PROC TRANSREG is used to code experimental designs for discrete choice models (see the section "Discrete Choice Experiments: DESIGN, NORESTORE, NOZERO" on page 7297). When these designs are very large, it might be more efficient to use the DESIGN=*n* *a-option*. It might be that attributes are constant within a block of *n* observations, so you need to specify the NOZEROCONSTANT *a-option* to get the correct results. You can specify this option in the PROC TRANSREG, MODEL, and OUTPUT statements.

NSR

specifies that no restrictions are placed on the use of SMOOTH and SSPLINE and the ordinary least squares is used to find the coefficients and predicted values. By default, only certain types of models can be specified with SMOOTH and ordinary least squares is not used to find the coefficients and predicted values. See the section "Smoothing Splines Changes and Enhancements" on page 7227 for more information about the NSR option and smooth transformations.

ORDER=DATA | FREQ | FORMATTED | INTERNAL

ORD=DAT | FRE | FOR | INT

specifies the order in which the CLASS variable levels are to be reported. The default is ORDER=INTERNAL. For ORDER=FORMATTED and ORDER=INTERNAL, the sort order is machine dependent. When you specify ORDER= as an *a-option* or an *o-option*, it determines the default ordering for all CLASS variables. When you specify ORDER= as a *t-option*, it overrides the default ordering only for selected variables.

DATA	sorts by order of appearance in the input data set.
FORMATTED	sorts by formatted value.
FREQ	sorts by descending frequency count; levels with the most observations appear first.
INTERNAL	sorts by unformatted value.

PBOXCOXTABLE

PBO

prints the Box-Cox table with the log likelihood displayed as a function of lambda. The important information in this table is displayed in the Box-Cox plot, so when ODS Graphics is in effect and the plot is produced, the table is not produced by default. When ODS Graphics is not in effect or when the plot is not produced, the table is produced by default. Specify the PBOXCOXTABLE option if you want to see the table in addition to the plot.

REFERENCE=NONE | MISSING | ZERO

REF=NON | MIS | ZER

specifies how reference levels of CLASS variables are to be treated. The options are REFERENCE=NONE, the default, in which reference levels are suppressed; REFERENCE=MISSING, in which reference levels are displayed and output with missing values; and REFERENCE=ZERO, in which reference levels are displayed and output with zeros. You can specify the REFERENCE= option in the PROC TRANSREG, MODEL, or OUTPUT statement, and you can specify it independently for the OUT= data set and the displayed output. When you specify it in only one statement, it sets the option for both the displayed output and the OUT= data set.

REITERATE

REI

enables PROC TRANSREG to use previous transformations as starting points. The REITERATE *a-option* affects only variables that are iteratively transformed (specified as LINEAR, OPSCORE, MONOTONE, UNTIE, SPLINE, MSPLINE, and SSPLINE). For iterative transformations, the REITERATE *a-option* requests a search in the input data set for a variable that consists of the value of the TDPREFIX= or TIPREFIX= *o-option* followed by the original variable name. If such a variable is found, it is used to provide the initial values for the first iteration. The final transformation is a member of the transformation family defined by the original variable, not the transformation family defined by the initialization variable. See the section "Using the REITERATE Algorithm Option" on page 7252 for more information about the REITERATE option.

RSQUARE

RSQ

prints a table with only the model R square.

SEPARATORS='*string-1***' <'***string-2***'>**

SEP='*string-1***' <'***string-2***'>**

specifies separators for creating CLASS expansion variable labels. By default, SEPARATORS=' ' * ' ("blank" and "blank asterisk blank"). The first value is used to separate variable names and values in interactions. The second value is used to separate interaction components. For example, the label for the coded variable for the A=1 and B=2 cell is, by default, 'A 1 * B 2'. If SEPARATORS='=' 'x' is specified, then the label is 'A=1xB=2'. When you specify SEPARATORS= as an *a-option* or an *o-option*, it determines the default separators for all CLASS variables. When you specify SEPARATORS= as a *t-option*, it overrides the default only for selected variables.

SHORT

SHO

suppresses the iteration histories.

SINGULAR=*n*

SIN=*n*

specifies the largest value within rounding error of zero. By default, SINGULAR=1E–12. PROC TRANSREG uses the value of the SINGULAR= *a-option* for checking $1 - R^2$ when constructing full-rank matrices of predictor variables, checking denominators before dividing, and so on. PROC TRANSREG computes the regression coefficients by sweeping with rational pivoting.

SS2

produces a regression table based on Type II sums of squares. Tests of the contribution of each transformation to the overall model are displayed and output to the OUTTEST= data set when you specify the OUTTEST= option. When you specify the SS2 *a-option*, the TEST *a-option* is automatically specified for you. See the section "Hypothesis Tests" on page 7264 for more information about the TEST and SS2 options. You can suppress the variable labels in the regression tables by specifying the NOLABEL option in the OPTIONS statement.

TEST

TES

generates an ANOVA table. PROC TRANSREG tests the null hypothesis that the vector of scoring coefficients for all of the transformations is zero. See the section "Hypothesis Tests" on page 7264 for more information about the TEST option.

TSUFFIX=*n*

TSU=*n*

specifies the number of characters in "Transformation" to append to variable labels for transformed variables. By default, all characters are used.

TSTANDARD=CENTER | NOMISS | ORIGINAL | Z
TST=CEN | NOM | ORI | Z

specifies the standardization of the transformed variables for the hypothesis tests and in the OUT= data set. By default, TSTANDARD=ORIGINAL. When you specify TSTANDARD= as an *a-option* or an *o-option*, it determines the default standardization for all variables. When you specify TSTANDARD= as a *t-option*, it overrides the default standardization only for selected variables.

CENTER centers the output variables to mean zero, but the variances are the same as the variances of the input variables.

NOMISS sets the means and variances of the transformed variables in the OUT= data set, computed over all output values that correspond to nonmissing values in the input data set, to the means and variances computed from the nonmissing observations of the original variables. The TSTANDARD=NOMISS specification is useful with missing data. When a variable is linearly transformed, the final variable contains the original nonmissing values and the missing value estimates. In other words, the nonmissing values are unchanged. If your data have no missing values, TSTANDARD=NOMISS and TSTANDARD=ORIGINAL produce the same results.

ORIGINAL sets the means and variances of the transformed variables to the means and variances of the original variables. This is the default.

Z standardizes the variables to mean zero, variance one.

The final standardization is affected by other options. If you also specify the ADDITIVE *a-option*, the TSTANDARD= option specifies an intermediate step in computing the final means and variances. The final independent variables, along with their means and standard deviations, are scaled by the regression coefficients, creating an additive model with all coefficients equal to one.

For nonoptimal variable transformations, the means and variances of the original variables are actually the means and variances of the nonlinearly transformed variables, unless you specify the ORIGINAL nonoptimal *t-option* in the MODEL statement. For example, if a variable x with no missing values is specified as LOG, then, by default, the final transformation of x is simply the log of x, not the log of x standardized to the mean of x and variance of x.

TYPE='*text***'|***name***
TYP='***text***'|***name***

specifies the valid value for the _TYPE_ variable in the input data set. If PROC TRANSREG finds an input _TYPE_ variable, it uses only observations with a _TYPE_ value that matches the TYPE= value. This enables a PROC TRANSREG OUT= data set containing coefficients to be used as input to PROC TRANSREG without requiring a WHERE statement to exclude the coefficients. If a _TYPE_ variable is not in the data set, all observations are used. The default is TYPE='SCORE', so if you do not specify the TYPE= *a-option*, only observations with _TYPE_='SCORE' are used. Do not confuse this *a-option* with the data set TYPE= option. The DATA= data set must be an ordinary SAS data set.

PROC TRANSREG displays a note when it reads observations with blank values of _TYPE_, but it does not automatically exclude those observations. Data sets created by the

TRANSREG and PRINQUAL procedures have blank _TYPE_ values for those observations that were excluded from the analysis due to nonpositive weights, nonpositive frequencies, or missing data. When these observations are read again, they are excluded for the same reason that they were excluded from their original analysis, not because their _TYPE_ value is blank.

UNTIE=*two-letters*

UNT=*two-letters*

specifies the first and last special missing values in the list of those special missing values that are to be estimated with within-variable order constraints but no category constraints. The *two-letters* value must consist of two letters in alphabetical order. By default, there are category constraints but no order constraints on special missing value estimates. For details, see the sections "Missing Values" on page 7249 and "Optimal Scaling" on page 7259.

UTILITIES

UTI

produces a table of the part-worth utilities from a conjoint analysis. Utilities, their standard errors, and the relative importance of each factor are displayed and output to the OUTTEST= data set when you specify the OUTTEST= option. When you specify the UTILITIES *a-option*, the TEST *a-option* is automatically specified for you. See Example 90.4 and Example 90.5 for more information about conjoint analysis.

OUTPUT Statement

OUTPUT *OUT=SAS-data-set* < *o-options* > **;**

The OUTPUT statement creates a new SAS data set that contains coefficients, marginal means, and information about the original and transformed variables. The information about original and transformed variables composes the score partition of the data set; observations have _TYPE_='SCORE'. The coefficients and marginal means compose the coefficient partition of the data set; observations have _TYPE_='M COEFFI' or _TYPE_='MEAN'. Other values of _TYPE_ are possible; for details, see "_TYPE_ and _NAME_ Variables" later in this chapter. For details about data set structure, see the section "Output Data Set" on page 7267. To specify the name of the output data set, use the OUT= option.

OUT=*SAS-data-set*

specifies the output data set for the data, transformed data, predicted values, residuals, scores, coefficients, and so on. When you use an OUTPUT statement but do not use the OUT= specification, PROC TRANSREG creates a data set and uses the DATA*n* convention. If you want to create a permanent SAS data set, you must specify a two-level name (see "SAS Files" in *SAS Language Reference: Concepts* and "Introduction to DATA Step Processing" in the *Base SAS Procedures Guide* for details).

To control the contents of the data set and variable names, use one or more of the *o-options*. You can also specify these options in the PROC TRANSREG statement.

Output Options (o-options)

The options listed in Table 90.5 are available in the OUTPUT statement. These options include the OUT= option and all of the *o-options*. Many of the statistics created in the OUTPUT statement are exactly the same as statistics created by PROC REG. More details are given in the sections "Predicted and Residual Values" on page 5531, "Model Fit and Diagnostic Statistics" on page 5557 in Chapter 73, "The REG Procedure," and Chapter 4, "Introduction to Regression Procedures."

Table 90.5 Options Available in the OUTPUT Statement

Option	Description
Identify output data set	
OUT=	outputs data set
Predicted Values, Residuals, Scores	
CANONICAL	outputs canonical scores
CLI	outputs individual confidence limits
CLM	outputs mean confidence limits
DESIGN=	specifies design matrix coding
DREPLACE	replaces dependent variables
IREPLACE	replaces independent variables
LEVERAGE	outputs leverage
NORESTOREMISSING	does not restore missing values
NOSCORES	suppresses output of scores
PREDICTED	outputs predicted values
REDUNDANCY=	outputs redundancy variables
REPLACE	replaces all variables
RESIDUALS	outputs residuals
Output Data Set Coefficients	
COEFFICIENTS	outputs coefficients
COORDINATES=	outputs ideal point coordinates
MEANS	outputs marginal means
MREDUNDANCY	outputs redundancy analysis coefficients
Output Data Set Variable Name Prefixes	
ADPREFIX=	specifies dependent variable approximations
AIPREFIX=	specifies independent variable approximations
CDPREFIX=	specifies canonical dependent variables
CILPREFIX=	specifies conservative individual lower CL
CIPREFIX=	specifies canonical independent variables
CIUPREFIX=	specifies conservative-individual-upper CL
CMLPREFIX=	specifies conservative-mean-lower CL
CMUPREFIX=	specifies conservative-mean-upper CL
DEPENDENT=	specifies METHOD=MORALS untransformed dependent
LILPREFIX=	specifies liberal-individual-lower CL
LIUPREFIX=	specifies liberal-individual-upper CL
LMLPREFIX=	specifies liberal-mean-lower CL
LMUPREFIX=	specifies liberal-mean-upper CL

Table 90.5 *continued*

Option	Description
RDPREFIX=	specifies residuals
PPREFIX=	specifies predicted values
RPREFIX=	specifies redundancy variables
TDPREFIX=	specifies transformed dependents
TIPREFIX=	specifies transformed independents
Macros Variables	
MACRO	creates macro variables
Other Options	
APPROXIMATIONS	outputs dependent and independent approximations
CCC	outputs canonical correlation coefficients
CEC	outputs canonical elliptical point coordinates
CPC	outputs canonical point coordinates
CQC	outputs canonical quadratic point coordinates
DAPPROXIMATIONS	outputs approximations to transformed dependents
IAPPROXIMATIONS	outputs approximations to transformed independents
MEC	outputs elliptical point coordinates
MPC	outputs point coordinates
MQC	outputs quadratic point coordinates
MRC	outputs multiple regression coefficients

For the coefficients partition, the COEFFICIENTS, COORDINATES, and MEANS *o-options* provide the coefficients that are appropriate for your model. For more explicit control of the coefficient partition, use the options that control details and prefixes. The following list provides details about these options.

ADPREFIX=*name*

ADP=*name*

specifies a prefix for naming the dependent variable predicted values. The default is ADPREFIX=P when you specify the PREDICTED *o-option*; otherwise, it is ADPREFIX=A. When you specify the ADPREFIX= *o-option*, the PREDICTED *o-option* is automatically specified for you. The ADPREFIX= *o-option* is the same as the PPREFIX= *o-option*.

AIPREFIX=*name*

AIP=*name*

specifies a prefix for naming the independent variable approximations. The default is AIPREFIX=A. When you specify the AIPREFIX= *o-option*, the IAPPROXIMATIONS *o-option* is automatically specified for you.

APPROXIMATIONS

APPROX

APP

is equivalent to specifying both the DAPPROXIMATIONS and the IAPPROXIMATIONS *o-options*. If you specify METHOD=UNIVARIATE, then the APPROXIMATIONS *o-option* specifies only the DAPPROXIMATIONS *o-option*.

CANONICAL

CAN

outputs canonical variables to the OUT= data set. When you specify METHOD=CANALS, the CANONICAL *o-option* is automatically specified for you. The CDPREFIX= *o-option* specifies a prefix for naming the dependent canonical variables (default Cand), and the CIPREFIX= *o-option* specifies a prefix for naming the independent canonical variables (default Cani).

CCC

outputs canonical correlation coefficients to the OUT= data set.

CDPREFIX=*name*

CDP=*name*

provides a prefix for naming the canonical dependent variables. The default is CDPREFIX=Cand. When you specify the CDPREFIX= *o-option*, the CANONICAL *o-option* is automatically specified for you.

CEC

outputs canonical elliptical point model coordinates to the OUT= data set.

CILPREFIX=*name*

CIL=*name*

specifies a prefix for naming the conservative-individual-lower confidence limits. The default prefix is CIL. When you specify the CILPREFIX= *o-option*, the CLI *o-option* is automatically specified for you.

CIPREFIX=*name*

CIP=*name*

provides a prefix for naming the canonical independent variables. The default is CIPREFIX=Cani. When you specify the CIPREFIX= *o-option*, the CANONICAL *o-option* is automatically specified for you.

CIUPREFIX=*name*

CIU=*name*

specifies a prefix for naming the conservative-individual-upper confidence limits. The default prefix is CIU. When you specify the CIUPREFIX= *o-option*, the CLI *o-option* is automatically specified for you.

CLI

outputs individual confidence limits to the OUT= data set. The names of the confidence limits variables are constructed from the original dependent variable names and the prefixes specified in the following *o-options*: LILPREFIX= (default LIL for liberal individual lower), CILPREFIX= (default CIL for conservative individual lower), LIUPREFIX= (default LIU for liberal individual upper), and CIUPREFIX= (default CIU for conservative individual upper). When there are no monotonicity constraints, the liberal and conservative limits are the same.

CLM

outputs mean confidence limits to the OUT= data set. The names of the confidence limits variables are constructed from the original dependent variable names and the prefixes specified in

the following *o-options*: LMLPREFIX= (default LML for liberal mean lower), CMLPREFIX= (default CML for conservative mean lower), LMUPREFIX= (default LMU for liberal mean upper), and CMUPREFIX= (default CMU for conservative mean upper). When there are no monotonicity constraints, the liberal and conservative limits are the same.

CMLPREFIX=*name*

CML=*name*

specifies a prefix for naming the conservative-mean-lower confidence limits. The default prefix is CML. When you specify the CMLPREFIX= *o-option*, the CLM *o-option* is automatically specified for you.

CMUPREFIX=*name*

CMU=*name*

specifies a prefix for naming the conservative-mean-upper confidence limits. The default prefix is CMU. When you specify the CMUPREFIX= *o-option*, the CLM *o-option* is automatically specified for you.

COEFFICIENTS

COE

outputs either multiple regression coefficients or raw canonical coefficients to the OUT= data set. If you specify METHOD=CANALS (in the MODEL or PROC TRANSREG statement), then the COEFFICIENTS *o-option* outputs the first *n* canonical variables, where *n* is the value of the NCAN= *a-option* (specified in the MODEL or PROC TRANSREG statement). Otherwise, the COEFFICIENTS *o-option* includes multiple regression coefficients in the OUT= data set. In addition, when you specify the CLASS expansion for any independent variable, the COEFFICIENTS *o-option* also outputs marginal means.

COORDINATES<=*n*>

COO<=*n*>

outputs either ideal point or vector model coordinates for preference mapping to the OUT= data set. When METHOD=CANALS, these coordinates are computed from canonical coefficients; otherwise, the coordinates are computed from multiple regression coefficients. For details, see the section "Point Models" on page 7255.

When ODS Graphics is enabled and vector model coordinates are requested, a plot is produced with points for each row and vectors for each column. If the vectors are plotted based on the actual computed coordinates, then often the vectors are short. A better graphical display is produced when the vectors are stretched. The absolute lengths of each vector can optionally be changed by specifying COORDINATES=*n*. Then the vector coordinates are all multiplied by *n*. Usually, *n* is a value such as 2, 2.5, or 3. The default is 2.5. Specify COORDINATES=1 if you want to see the vectors without any stretching. The relative lengths of the different vectors are important and interpretable, and these are preserved by the stretching.

CPC

outputs canonical point model coordinates to the OUT= data set.

CQC

outputs canonical quadratic point model coordinates to the OUT= data set.

DAPPROXIMATIONS

DAP

outputs the approximations of the transformed dependent variables to the OUT= data set. These are the target values for the optimal transformations. With METHOD=UNIVARIATE and METHOD=MORALS, the dependent variable approximations are the ordinary predicted values from the linear model. The names of the approximation variables are constructed from the ADPREFIX= *o-option* (default A) and the original dependent variable names. For ordinary predicted values, use the PREDICTED *o-option* instead of the DAPPROXIMATIONS *o-option*, since the PREDICTED *o-option* uses a more relevant prefix ("P" instead of "A") and a more relevant variable label suffix ("Predicted Values" instead of "Approximations").

DESIGN<=*n*>

DES<=*n*>

specifies that your primary goal is design matrix coding, not analysis. Specifying the DESIGN *o-option* makes the procedure run faster. The DESIGN *o-option* sets the default method to UNIVARIATE and the default MAXITER= value to zero. It suppresses computing the regression coefficients, unless they are needed for some other option. Furthermore, when the DESIGN *o-option* is specified, the MODEL statement is not required to have an equal sign. When no MODEL statement equal sign is specified, all variables are considered independent variables, all options that require dependent variables are ignored, and the IREPLACE *o-option* is automatically specified for you.

You can use DESIGN=*n* for coding very large data sets, where *n* is the number of observations to code at one time. For example, to code a data set with a large number of observations, you can specify DESIGN=100 or DESIGN=1000 to process the data set in blocks of 100 or 1000 observations. If you specify the DESIGN *o-option* rather than DESIGN=*n*, PROC TRANSREG tries to process all observations at once, which might not work with very large data sets. Specify the NOZEROCONSTANT *a-option* with DESIGN=*n* to ensure that constant variables within blocks are not zeroed. See the sections "Using the DESIGN Output Option" on page 7293 and "Discrete Choice Experiments: DESIGN, NORESTORE, NOZERO" on page 7297 for more information about the DESIGN option.

DEPENDENT=*name*

DEP=*name*

specifies the untransformed dependent variable for OUT= data sets with METHOD=MORALS when there is more than one dependent variable. The default is DEPENDENT=_DEPEND_.

DREPLACE

DRE

replaces the original dependent variables with the transformed dependent variables in the OUT= data set. The names of the transformed variables in the OUT= data set correspond to the names of the original dependent variables in the input data set. By default, both the original dependent variables and the transformed dependent variables (with names constructed from the TDPREFIX= (default T) *o-option* and the original dependent variable names) are included in the OUT= data set.

IAPPROXIMATIONS

IAP

outputs the approximations of the transformed independent variables to the OUT= data set.

These are the target values for the optimal transformations. The names of the approximation variables are constructed from the AIPREFIX= *o-option* (default A) and the original independent variable names. When you specify the AIPREFIX= *o-option*, the IAPPROXIMATIONS *o-option* is automatically specified for you. The IAPPROXIMATIONS *o-option* is not valid when METHOD=UNIVARIATE.

IREPLACE

IRE

replaces the original independent variables with the transformed independent variables in the OUT= data set. The names of the transformed variables in the OUT= data set correspond to the names of the original independent variables in the input data set. By default, both the original independent variables and the transformed independent variables (with names constructed from the TIPREFIX= *o-option* (default T) and the original independent variable names) are included in the OUT= data set.

LEVERAGE<=*name***>**

LEV<=*name***>**

creates a variable with the specified name in the OUT= data set that contains leverages. Specifying the LEVERAGE *o-option* is equivalent to specifying LEVERAGE=Leverage.

LILPREFIX=*name*

LIL=*name*

specifies a prefix for naming the liberal-individual-lower confidence limits. The default prefix is LIL. When you specify the LILPREFIX= *o-option*, the CLI *o-option* is automatically specified for you.

LIUPREFIX=*name*

LIU=*name*

specifies a prefix for naming the liberal-individual-upper confidence limits. The default prefix is LIU. When you specify the LIUPREFIX= *o-option*, the CLI *o-option* is automatically specified for you.

LMLPREFIX=*name*

LML=*name*

specifies a prefix for naming the liberal-mean-lower confidence limits. The default prefix is LML. When you specify the LMLPREFIX= *o-option*, the CLM *o-option* is automatically specified for you.

LMUPREFIX=*name*

LMU=*name*

specifies a prefix for naming the liberal-mean-upper confidence limits. The default prefix is LMU. When you specify the LMUPREFIX= *o-option*, the CLM *o-option* is automatically specified for you.

MACRO(*keyword=name...***)**

MAC(*keyword=name...***)**

creates macro variables. Most of the options available within the MACRO *o-option* are rarely

needed. By default, PROC TRANSREG creates a macro variable named _TrgInd with a complete list of independent variables created by the procedure. When PROC TRANSREG is being used for design matrix creation prior to running a procedure without a CLASS statement, this macro provides a convenient way to use the results from PROC TRANSREG. For example, a PROC LOGISTIC step that uses a design matrix coded by PROC TRANSREG can use the following MODEL statement:

```
model y=&_trgind;
```

PROC TRANSREG, also by default, creates a macro variable named _TrgIndN, which contains the number of variables in the _TrgInd list. These macro variables can be used in an ARRAY statement as follows:

```
array indvars[&_trgindn] &_trgind;
```

See the sections "Using the DESIGN Output Option" on page 7293 and "Discrete Choice Experiments: DESIGN, NORESTORE, NOZERO" on page 7297 for examples of using the default macro variables.

The available *keywords* are as follows.

DN=*name* specifies the name of a macro variable that contains the number of dependent variables. By default, a macro variable named _TrgDepN is created. This is the number of variables in the DL= list and the number of macro variables created by the DV= and DE= specifications.

IN=*name* specifies the name of a macro variable that contains the number of independent variables. By default, a macro variable named _TrgIndN is created. This is the number of variables in the IL= list and the number of macro variables created by the IV= and IE= specifications.

DL=*name* specifies the name of a macro variable that contains the list of the dependent variables. By default, a macro variable named _TrgDep is created. These are the variable names of the final transformed variables in the OUT= data set. For example, if there are three dependent variables, y1–y3, then _TrgDep contains, by default, Ty1 Ty2 Ty3 (or y1 y2 y3 if you specify the REPLACE *o-option*).

IL=*name* specifies the name of a macro variable that contains the list of the independent variables. By default, a macro variable named _TrgInd is created. These are the variable names of the final transformed variables in the OUT= data set. For example, if there are three independent variables, x1–x3, then _TrgInd contains, by default, Tx1 Tx2 Tx3 (or x1 x2 x3 if you specify the REPLACE *o-option*).

DV=*prefix* specifies a prefix for creating a list of macro variables, each of which contains one dependent variable name. For example, if there are three dependent variables, y1–y3, and you specify **macro(dv=Dep)**, then three macro variables, Dep1, Dep2, and Dep3, are created, containing Ty1, Ty2, and Ty3, respectively (or y1, y2, and y3 if you specify the REPLACE *o-option*). By default, no list is created.

IV=*prefix* specifies a prefix for creating a list of macro variables, each of which contains one independent variable name. For example, if there are three independent variables, x1–x3, and you specify **macro(iv=Ind)**, then three macro variables, Ind1, Ind2, and Ind3, are created, containing Tx1, Tx2, and TX3, respectively (or x1, x2, and x3 if you specify the REPLACE *o-option*). By default, no list is created.

DE=*prefix* specifies a prefix for creating a list of macro variables, each of which contains one dependent variable effect. This list shows the origin of each model term. Each effect consists of two or more parts, and each part consists of a value in 32 columns followed by a blank. For example, if you specify **macro(de=d)**, then a macro variable d1 is created for **identity(y)**. The d1 macro variable is shown next, wrapped onto two lines:

```
4                                TY
IDENTITY                         Y
```

The first part is the number of parts (4), the second part is the transformed variable name, the third part is the transformation, and the last part is the input variable name. By default, no list is created.

IE=*prefix* specifies a prefix for creating a list of macro variables, each of which contains one independent variable effect. This list shows the origin of each model term. Each effect consists of two or more parts, and each part consists of a value in 32 columns followed by a blank. For example, if you specify **macro(ie=I)**, then three macro variables, I1, I2, and I3, are created for **class(x1 | x2)** when both x1 and x2 have values of 1 and 2. These macro variables are shown next, with extra white space removed:

```
5     Tx11     CLASS    x1   1
5     Tx21     CLASS    x2   1
8     Tx11x21  CLASS    x1   1      CLASS    x2   1
```

For CLASS variables, the formatted level appears after the variable name. The first two effects are the main effects, and the last is the interaction term. By default, no list is created.

MEANS

MEA

outputs marginal means for CLASS variable expansions to the OUT= data set.

MEC

outputs multiple regression elliptical point model coordinates to the OUT= data set.

MPC

outputs multiple regression point model coordinates to the OUT= data set.

MQC

outputs multiple regression quadratic point model coordinates to the OUT= data set.

MRC

outputs multiple regression coefficients to the OUT= data set.

MREDUNDANCY

MRE

outputs multiple redundancy analysis coefficients to the OUT= data set.

NORESTOREMISSING

NORESTORE

NOR

specifies that missing values should not be restored when the OUT= data set is created. By default, the coded CLASS variable contains a row of missing values for observations in which the CLASS variable is missing. When you specify the NORESTOREMISSING *o-option*, these observations contain a row of zeros instead. This is useful when PROC TRANSREG is used to code experimental designs for discrete choice models and there is a constant alternative indicated by a missing value.

NOSCORES

NOS

excludes original variables, transformed variables, predicted values, residuals, and scores from the OUT= data set. You can use the NOSCORES *o-option* with various other options to create an OUT= data set that contains only a coefficient partition (for example, a data set consisting entirely of coefficients and coordinates).

PREDICTED

PRE

P

outputs predicted values, which for METHOD=UNIVARIATE and METHOD=MORALS are the ordinary predicted values from the linear model, to the OUT= data set. The names of the predicted values' variables are constructed from the PPREFIX= *o-option* (default P) and the original dependent variable names. When you specify the PPREFIX= *o-option*, the PREDICTED *o-option* is automatically specified for you.

PPREFIX=*name*

PDPREFIX=*name*

PDP=*name*

specifies a prefix for naming the dependent variable predicted values. The default is PPREFIX=P when you specify the PREDICTED *o-option*; otherwise, it is PPREFIX=A. When you specify the PPREFIX= *o-option*, the PREDICTED *o-option* is automatically specified for you. The PPREFIX= *o-option* is the same as the ADPREFIX= *o-option*.

RDPREFIX=*name*

RDP=*name*

specifies a prefix for naming the residual (dependent) variables to the OUT= data set. The default is RDPREFIX=R. When you specify the RDPREFIX= *o-option*, the RESIDUALS *o-option* is automatically specified for you.

REDUNDANCY< =STANDARDIZE | UNSTANDARDIZE >
RED< =STA | UNS >

outputs redundancy variables to the OUT= data set, either standardized or unstandardized. Specifying the REDUNDANCY *o-option* is the same as specifying REDUNDANCY=STANDARDIZE. The results of the REDUNDANCY *o-option* depends on the TSTANDARD= option. You must specify TSTANDARD=Z to get results based on standardized data. The TSTANDARD= option controls how the data that go into the redundancy analysis are scaled, and REDUNDANCY=STANDARDIZE|UNSTANDARDIZE controls how the redundancy variables are scaled. The REDUNDANCY *o-option* is automatically specified for you when you specify the METHOD=REDUNDANCY *a-option*. The RPREFIX= *o-option* specifies a prefix (default Red) for naming the redundancy variables.

REFERENCE=NONE | MISSING | ZERO
REF=NON | MIS | ZER

specifies how reference levels of CLASS variables are to be treated. The options are REFERENCE=NONE, the default, in which reference levels are suppressed; REFERENCE=MISSING, in which reference levels are displayed and output with missing values; and REFERENCE=ZERO, in which reference levels are displayed and output with zeros. You can specify the REFERENCE= option in the PROC TRANSREG, MODEL, or OUTPUT statement, and you can specify it independently for the OUT= data set and the displayed output. When you specify it in only one statement, it sets the option for both the displayed output and the OUT= data set.

REPLACE
REP

is equivalent to specifying both the DREPLACE and the IREPLACE *o-options*.

RESIDUALS
RES
R

outputs the differences between the transformed dependent variables and their predicted values. The names of the residual variables are constructed from the RDPREFIX= *o-option* (default R) and the original dependent variable names.

RPREFIX=*name*
RPR=*name*

provides a prefix for naming the redundancy variables. The default is RPREFIX=Red. When you specify the RPREFIX= *o-option*, the REDUNDANCY *o-option* is automatically specified for you.

TDPREFIX=*name*
TDP=*name*

specifies a prefix for naming the transformed dependent variables. By default, TDPREFIX=T. The TDPREFIX= *o-option* is ignored when you specify the DREPLACE *o-option*.

TIPREFIX=*name*
TIP=*name*

specifies a prefix for naming the transformed independent variables. By default, TIPREFIX=T. The TIPREFIX= *o-option* is ignored when you specify the IREPLACE *o-option*.

WEIGHT Statement

WEIGHT *variable* **;**

When you use a WEIGHT statement, a weighted residual sum of squares is minimized. The WEIGHT statement has no effect on degrees of freedom or number of observations, but the weights affect most other calculations. The observation is used in the analysis only if the value of the WEIGHT statement variable is greater than 0.

Details: TRANSREG Procedure

Model Statement Usage

MODEL < *transform(dependents* < / *t-options* >)>
< *transform(dependents* < / *t-options* >) . . . =>
transform(independents < / *t-options* >)
< *transform(independents* < / *t-options* >) . . . > < / *a-options* > **;**

Here are some examples of model statements:

- linear regression

```
model identity(y) = identity(x);
```

- a linear model with a nonlinear regression function

```
model identity(y) = spline(x / nknots=5);
```

- multiple regression

```
model identity(y) = identity(x1-x5);
```

- multiple regression with nonlinear transformations

```
model spline(y / nknots=3) = spline(x1-x5 / nknots=3);
```

- multiple regression with nonlinear but monotone transformations

```
model mspline(y / nknots=3) = mspline(x1-x5 / nknots=3);
```

- multivariate multiple regression

```
model identity(y1-y4) = identity(x1-x5);
```

- canonical correlation

```
model identity(y1-y4) = identity(x1-x5) / method=canals;
```

- redundancy analysis

```
model identity(y1-y4) = identity(x1-x5) / method=redundancy;
```

- preference mapping, vector model (Carroll 1972)

```
model identity(Attrib1-Attrib3) = identity(Dim1-Dim2);
```

- preference mapping, ideal point model (Carroll 1972)

```
model identity(Attrib1-Attrib3) = point(Dim1-Dim2);
```

- preference mapping, ideal point model, elliptical (Carroll 1972)

```
model identity(Attrib1-Attrib3) = epoint(Dim1-Dim2);
```

- preference mapping, ideal point model, quadratic (Carroll 1972)

```
model identity(Attrib1-Attrib3) = qpoint(Dim1-Dim2);
```

- metric conjoint analysis

```
model identity(Subj1-Subj50) = class(a b c d e f / zero=sum);
```

- nonmetric conjoint analysis

```
model monotone(Subj1-Subj50) = class(a b c d e f / zero=sum);
```

- main effects, two-way interaction

```
model identity(y) = class(a|b);
```

- less-than-full-rank model—main effects and two-way interaction are constrained to sum to zero

```
model identity(y) = class(a|b / zero=sum);
```

- main effects and all two-way interactions

```
model identity(y) = class(a|b|c@2);
```

- main effects and all two- and three-way interactions

```
model identity(y) = class(a|b|c);
```

- main effects and only the b*c two-way interaction

```
model identity(y) = class(a b c b*c);
```

- seven main effects, three two-way interactions

```
model identity(y) = class(a b c d e f g a*b a*c a*d);
```

- deviations-from-means (effects or $(1, 0, -1)$) coding, with an a reference level of '1' and a b reference level of '2'

```
model identity(y) = class(a|b / deviations zero='1' '2');
```

- cell-means coding (implicit intercept)

```
model identity(y) = class(a*b / zero=none);
```

- reference cell model

```
model identity(y) = class(a|b / zero='1' '1');
```

- reference line with change in line parameters

```
model identity(y) = class(a) | identity(x);
```

- reference curve with change in curve parameters

```
model identity(y) = class(a) | spline(x);
```

- separate curves and intercepts

```
model identity(y) = class(a / zero=none) | spline(x);
```

- quantitative effects with interaction

```
model identity(y) = identity(x1 | x2);
```

- separate quantitative effects with interaction within each cell

```
model identity(y) = class(a * b / zero=none) | identity(x1 | x2);
```

Box-Cox Transformations

Box-Cox (1964) transformations are used to find potentially nonlinear transformations of a dependent variable. The Box-Cox transformation has the form

$$\begin{array}{ll} (y^{\lambda} - 1)/\lambda & \lambda \neq 0 \\ \log(y) & \lambda = 0 \end{array}$$

This family of transformations of the positive dependent variable y is controlled by the parameter λ. Transformations linearly related to square root, inverse, quadratic, cubic, and so on are all

special cases. The limit as λ approaches 0 is the log transformation. More generally, Box-Cox transformations of the following form can be fit:

$$\begin{array}{ll} ((y+c)^{\lambda}-1)/(\lambda g) & \lambda \neq 0 \\ \log(y+c)/g & \lambda = 0 \end{array}$$

By default, $c = 0$. The parameter c can be used to rescale y so that it is strictly positive. By default, $g = 1$. Alternatively, g can be $\dot{y}^{\lambda-1}$, where $\dot{y}$ is the geometric mean of y.

The BOXCOX transformation in PROC TRANSREG can be used to perform a Box-Cox transformation of the dependent variable. You can specify a list of power parameters by using the LAMBDA= *t-option*. By default, LAMBDA=–3 TO 3 BY 0.25. The procedure chooses the optimal power parameter by using a maximum likelihood criterion (Draper and Smith 1981, pp. 225–226). You can specify the PARAMETER=*c* transformation option when you want to shift the values of y, usually to avoid negatives. To divide by $\dot{y}^{\lambda-1}$, specify the GEOMETRICMEAN *t-option*.

Here are three examples of using the LAMBDA= *t-option*:

```
model BoxCox(y / lambda=0) = identity(x1-x5);
model BoxCox(y / lambda=-2 to 2 by 0.1) = identity(x1-x5);
model BoxCox(y) = identity(x1-x5);
```

Here is the first example:

```
model BoxCox(y / lambda=0) = identity(x1-x5);
```

LAMBDA=0 specifies a Box-Cox transformation with a power parameter of 0. Since a single value of 0 was specified for LAMBDA=, there is no difference between the following models:

```
model BoxCox(y / lambda=0) = identity(x1-x5);
model log(y) = identity(x1-x5);
```

Here is the second example:

```
model BoxCox(y / lambda=-2 to 2 by 0.1) = identity(x1-x5);
```

LAMBDA= specifies a list of power parameters. PROC TRANSREG tries each power parameter in the list and picks the best transformation. A maximum likelihood approach (Draper and Smith 1981, pp. 225–226) is used. With Box-Cox transformations, PROC TRANSREG finds the transformation before the usual iterations begin. Note that this is quite different from PROC TRANSREG's usual approach of iteratively finding optimal transformations with ordinary and alternating least squares. It is analogous to SMOOTH and PBSPLINE, which also find transformations before the iterations begin based on a criterion other than least squares.

Here is the third example:

```
model BoxCox(y) = identity(x1-x5);
```

The default LAMBDA= list of –3 TO 3 BY 0.25 is used.

The procedure prints the optimal power parameter, a confidence interval on the power parameter (based on the ALPHA= *t-option*), a "convenient" power parameter (selected from the CLL= *t-option* list), and the log likelihood for each power parameter tried (see Example 90.2).

To illustrate how Box-Cox transformations work, data were generated from the model

$$y = e^{x+\epsilon}$$

where $\epsilon \sim \mathrm{N}(0, 1)$. The transformed data can be fit with a linear model

$$\log(y) = x + \epsilon$$

The following statements produce Figure 90.14 through Figure 90.15:

```
title 'Basic Box-Cox Example';

data x;
   do x = 1 to 8 by 0.025;
      y = exp(x + normal(7));
      output;
   end;
run;

ods graphics on;

title2 'Default Options';

proc transreg data=x test;
   model BoxCox(y) = identity(x);
run;
```

Figure 90.14 Basic Box-Cox Example, Default Output

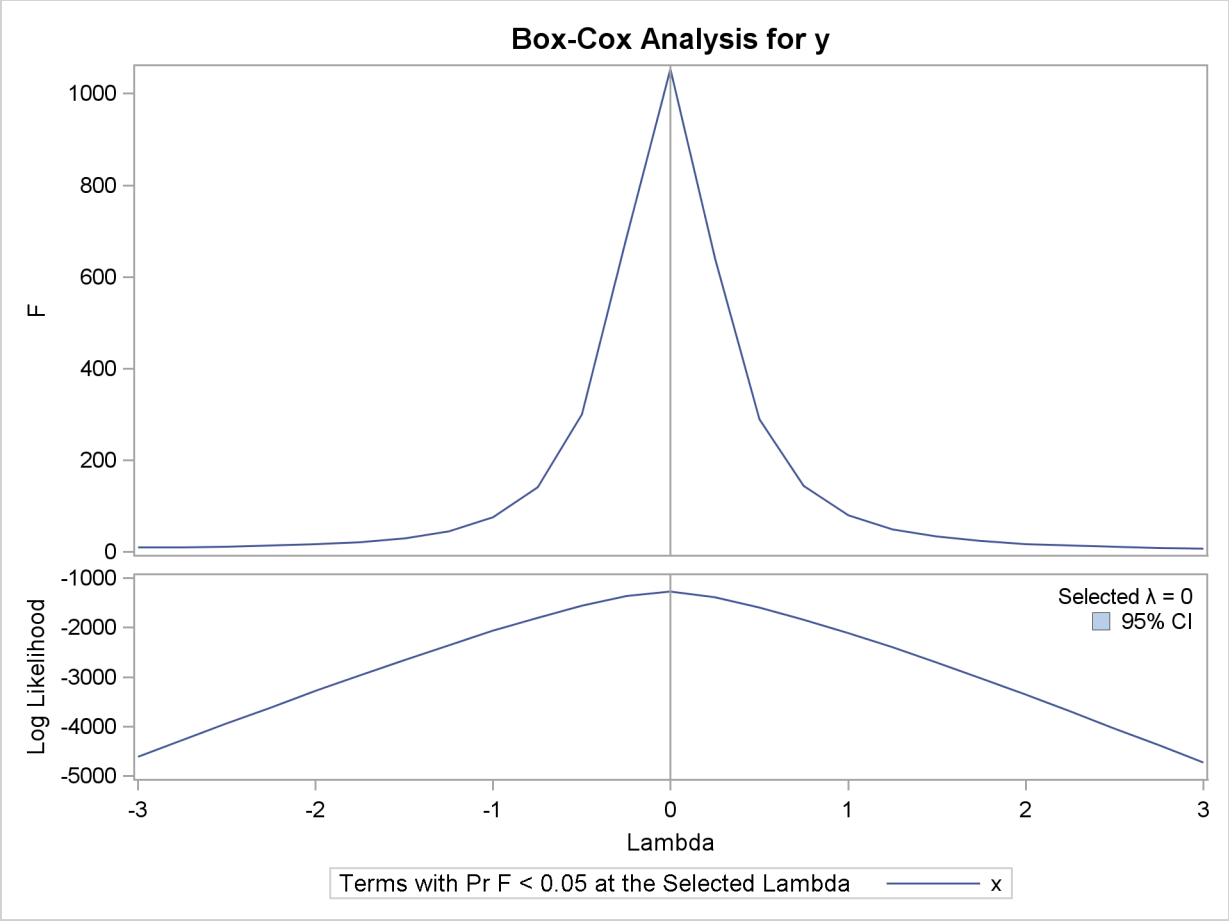


Figure 90.14 shows that PROC TRANSREG correctly selects the log transformation $\lambda = 0$, with a narrow confidence interval. The $F = t^2$ plot shows that F is at its largest in the vicinity of the optimal Box-Cox transformation.

The rest of the output, which contains the ANOVA results, is shown in Figure 90.15.

Figure 90.15 Basic Box-Cox Example, Default Output

```
                    Dependent Variable BoxCox(y)


              Number of Observations Read            281
              Number of Observations Used            281
```

Figure 90.15 *continued*

The TRANSREG Procedure Hypothesis Tests for BoxCox(y)

Univariate ANOVA Table Based on the Usual Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Liberal p
Model	1	1145.884	1145.884	1053.66	>= <.0001
Error	279	303.421	1.088		
Corrected Total	280	1449.305			

The above statistics are not adjusted for the fact that the dependent variable was transformed and so are generally liberal.

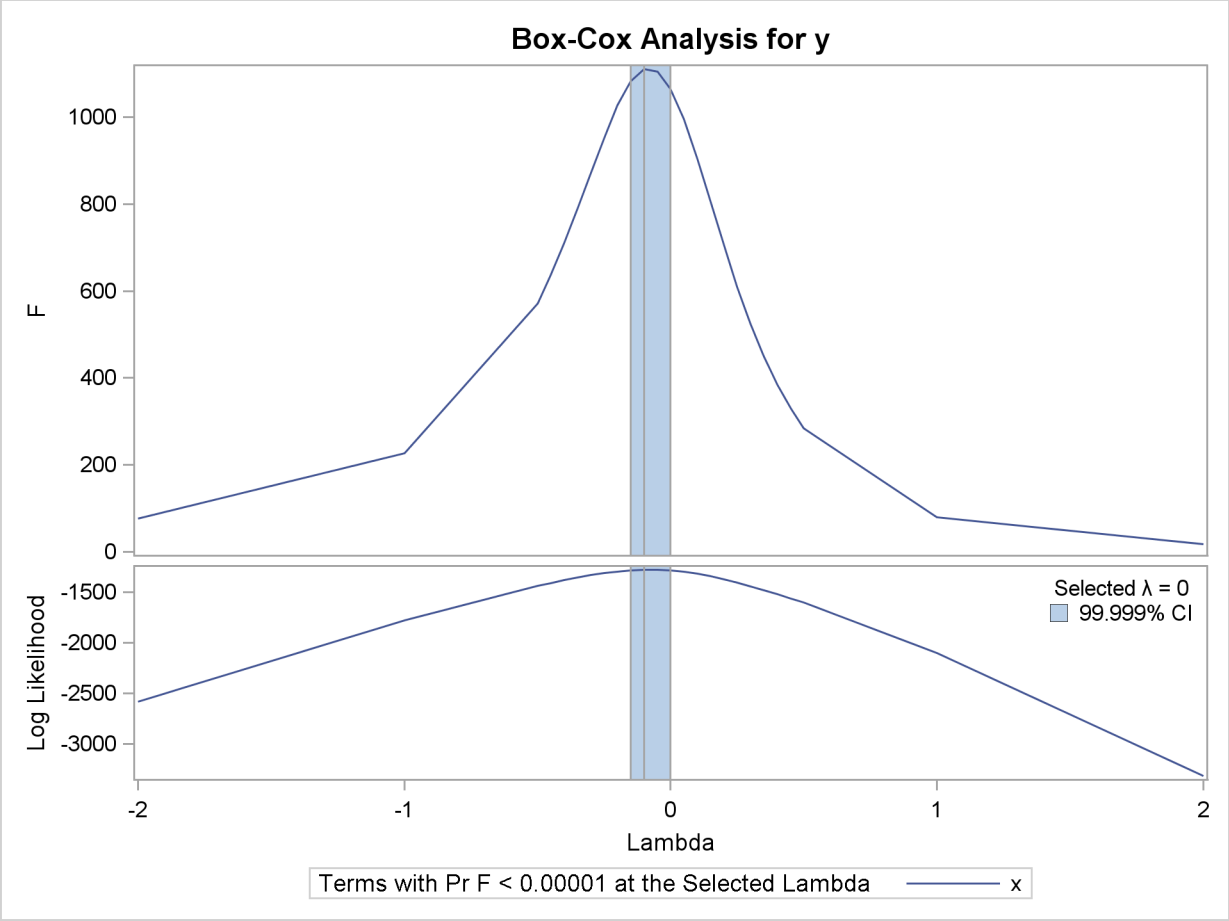
Root MSE	1.04285	R-Square	0.7906
Dependent Mean	4.49653	Adj R-Sq	0.7899
Coeff Var	23.19225	Lambda	0.0000

This next example uses several options. The LAMBDA= *t-option* specifies power parameters sparsely from –2 to –0.5 and 0.5 to 2 just to get the general shape of the log-likelihood function in that region. Between –0.5 and 0.5, more power parameters are tried. The CONVENIENT *t-option* is specified so that if a power parameter like $\lambda = 1$ or $\lambda = 0$ is found in the confidence interval, it is used instead of the optimal power parameter. PARAMETER=2 is specified to add 2 to each y before performing the transformations. ALPHA=0.00001 specifies a wide confidence interval.

These next statements perform the Box-Cox analysis and produce Figure 90.16 and Figure 90.17:

```
title2 'Several Options Demonstrated';

proc transreg data=x ss2 details
              plots=(transformation(dependent) scatter
                     observedbypredicted);
   model BoxCox(y / lambda=-2 -1 -0.5 to 0.5 by 0.05 1 2
                    convenient parameter=2 alpha=0.00001) =
         identity(x);
run;
```

Figure 90.16 Basic Box-Cox Example, Several Options Demonstrated



The results in Figure 90.16 and Figure 90.17 show that the optimal power parameter is –0.1, but 0 is in the confidence interval, and hence a log transformation is chosen. The actual Box-Cox transformation, the original scatter plot, and observed by predicted values plot are shown in Figure 90.17.

Figure 90.17 Basic Box-Cox Example, Several Options Demonstrated

```
                    Dependent Variable BoxCox(y)


              Number of Observations Read            281
              Number of Observations Used            281
```

Figure 90.17 *continued*

Model Statement Specification Details

Type	DF	Variable	Description	Value
Dep	1	BoxCox(y)	Lambda Used	0
			Lambda	-0.1
			Log Likelihood	-1280.1
			Conv. Lambda	0
			Conv. Lambda LL	-1287.7
			CI Limit	-1289.9
			Alpha	0.00001
			Parameter	2
			Options	Convenient Lambda Used
Ind	1	Identity(x)	DF	1

The TRANSREG Procedure Hypothesis Tests for BoxCox(y)

Univariate ANOVA Table Based on the Usual Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Liberal p
Model	1	999.438	999.4381	1064.82	>= <.0001
Error	279	261.868	0.9386		
Corrected Total	280	1261.306			

The above statistics are not adjusted for the fact that the dependent variable was transformed and so are generally liberal.

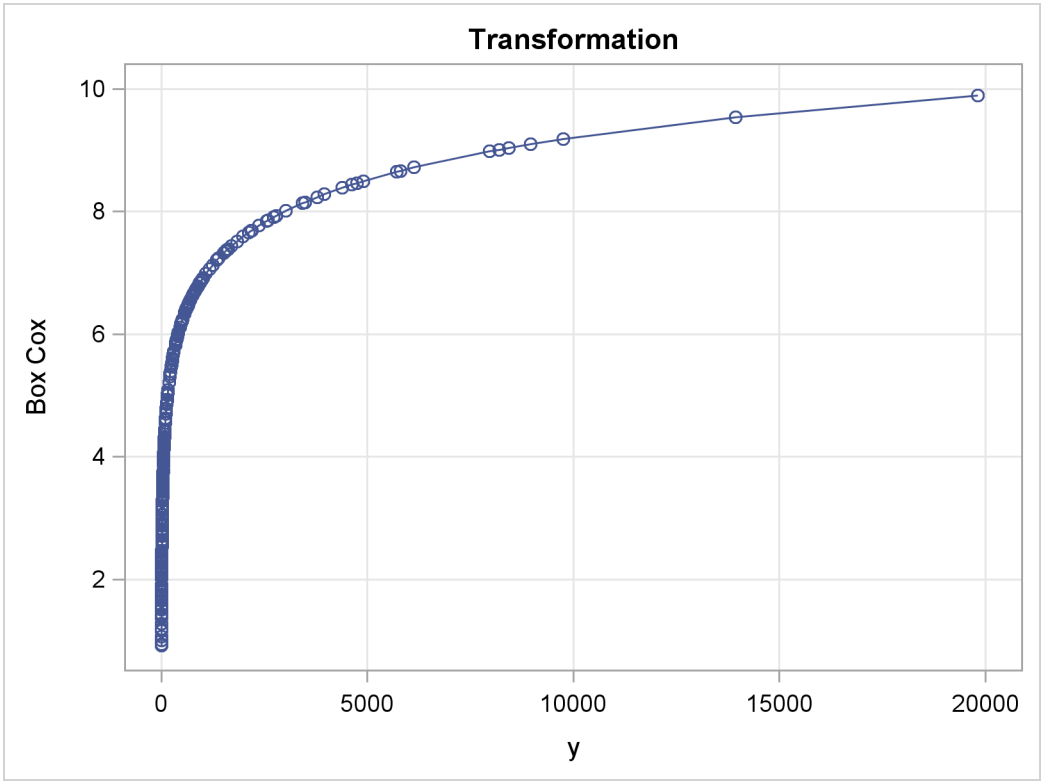
Root MSE	0.96881	R-Square	0.7924
Dependent Mean	4.61429	Adj R-Sq	0.7916
Coeff Var	20.99591	Lambda	0.0000

Univariate Regression Table Based on the Usual Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Liberal p
Intercept	1	0.42939328	8.746	8.746	9.32	>= 0.0025
Identity(x)	1	0.92997620	999.438	999.438	1064.82	>= <.0001

The above statistics are not adjusted for the fact that the dependent variable was transformed and so are generally liberal.

Figure 90.17 *continued*



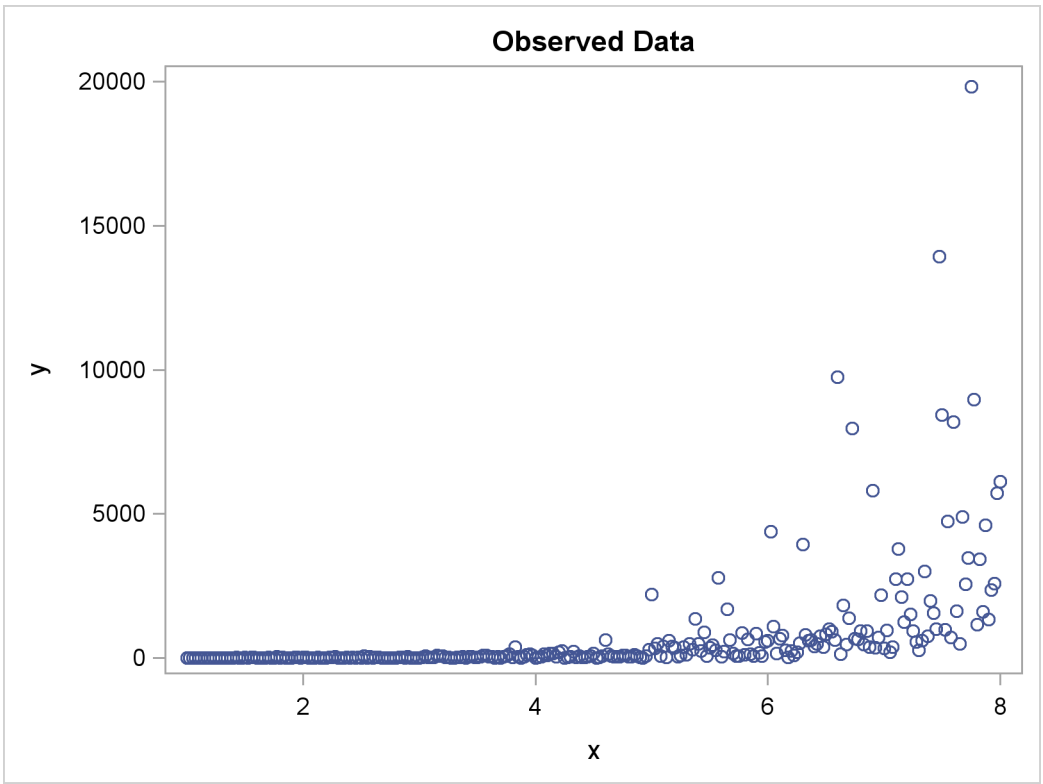
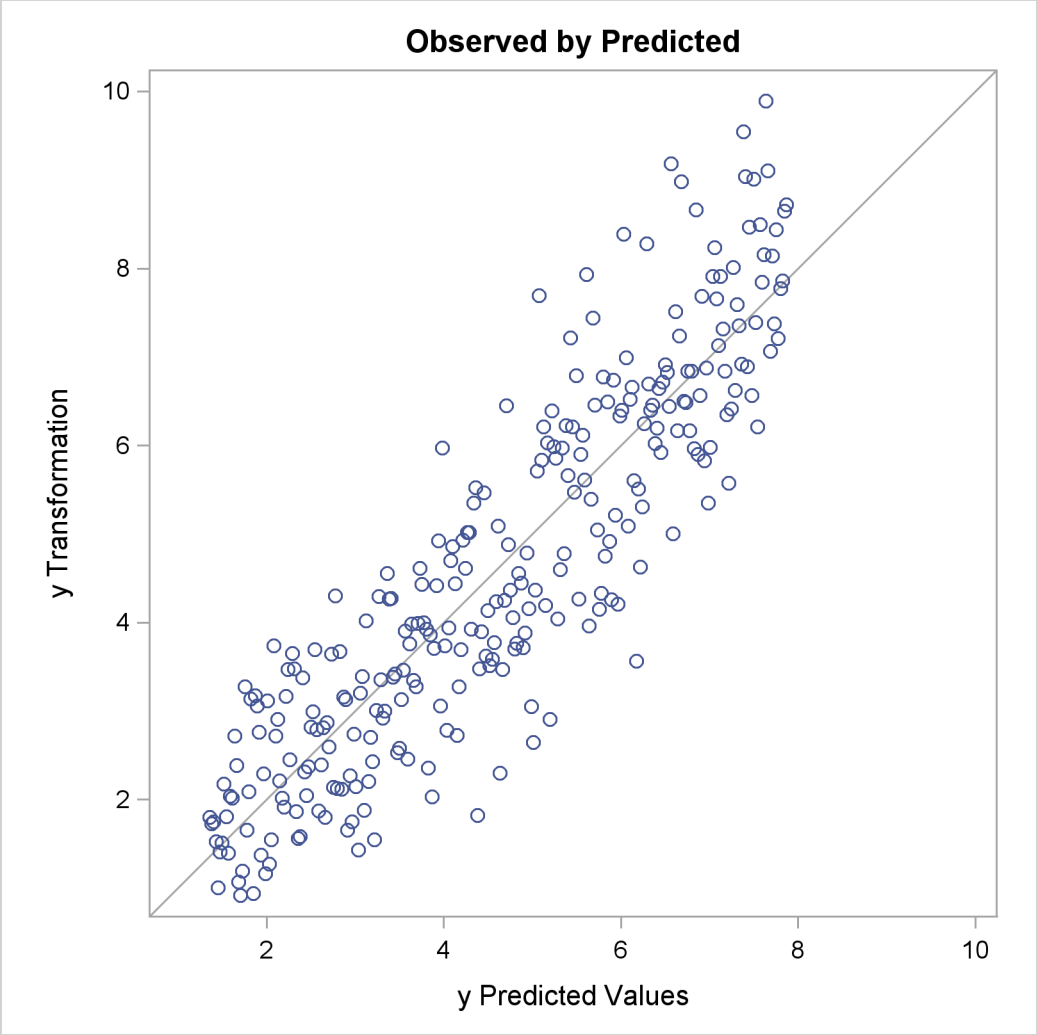


Figure 90.17 *continued*



The next example shows how to find a Box-Cox transformation without an independent variable. This seeks to normalize the univariate histogram. This example generates 500 random observations from a lognormal distribution. In addition, a constant variable z is created that is all zero. This is because PROC TRANSREG requires some independent variable to be specified, even if it is constant. Two options are specified in the PROC TRANSREG statement. MAXITER=0 is specified because the Box-Cox transformation is performed before any iterations are begun. No iterations are needed since no other work is required. The NOZEROCONSTANT *a-option* (which can be abbreviated NOZ) is specified so that PROC TRANSREG does not print any warnings when it encounters the constant independent variable. The MODEL statement asks for a Box-Cox transformation of y and an IDENTITY transformation (which does nothing) of the constant variable z. Finally, PROC UNIVARIATE is run to show a histogram of the original variable y, and the Box-Cox transformation, Ty. The following statements fit the univariate Box-Cox model and produce Figure 90.18:

```
title 'Univariate Box-Cox';

data x;
   call streaminit(17);
   z = 0;
   do i = 1 to 500;
      y = rand('lognormal');
      output;
   end;
run;

proc transreg maxiter=0 nozeroconstant;
   model BoxCox(y) = identity(z);
   output;
run;

proc univariate noprint;
   histogram y ty;
run;

ods graphics off;
```

The PROC TRANSREG results in Figure 90.18 show that zero is chosen for lambda, so a log transformation is chosen. The first histogram shows that the original data are skewed, but a log transformation makes the data appear much more nearly normal.

Figure 90.18 Box-Cox with No Independent Variable

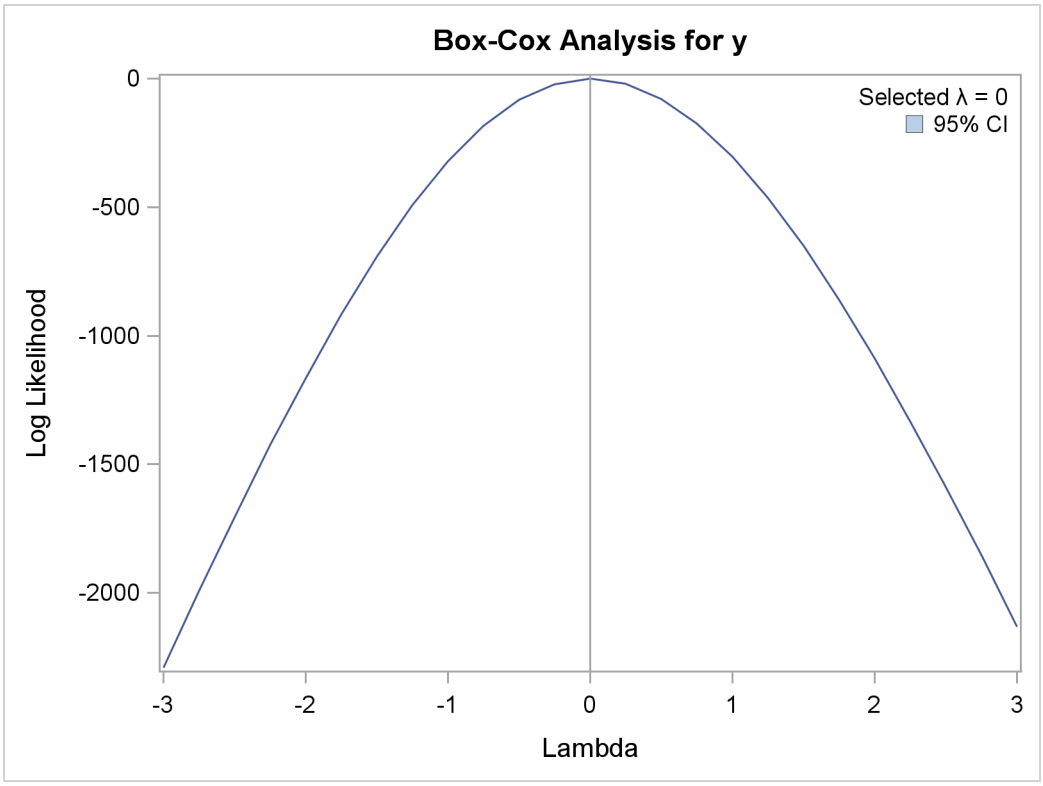
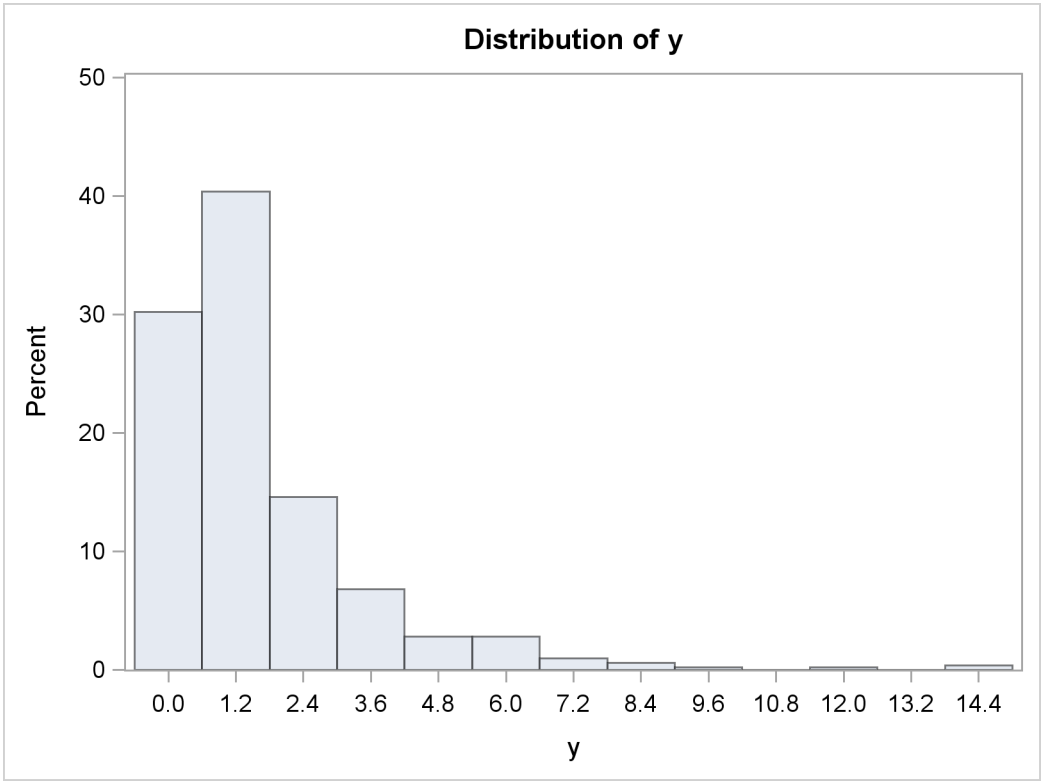
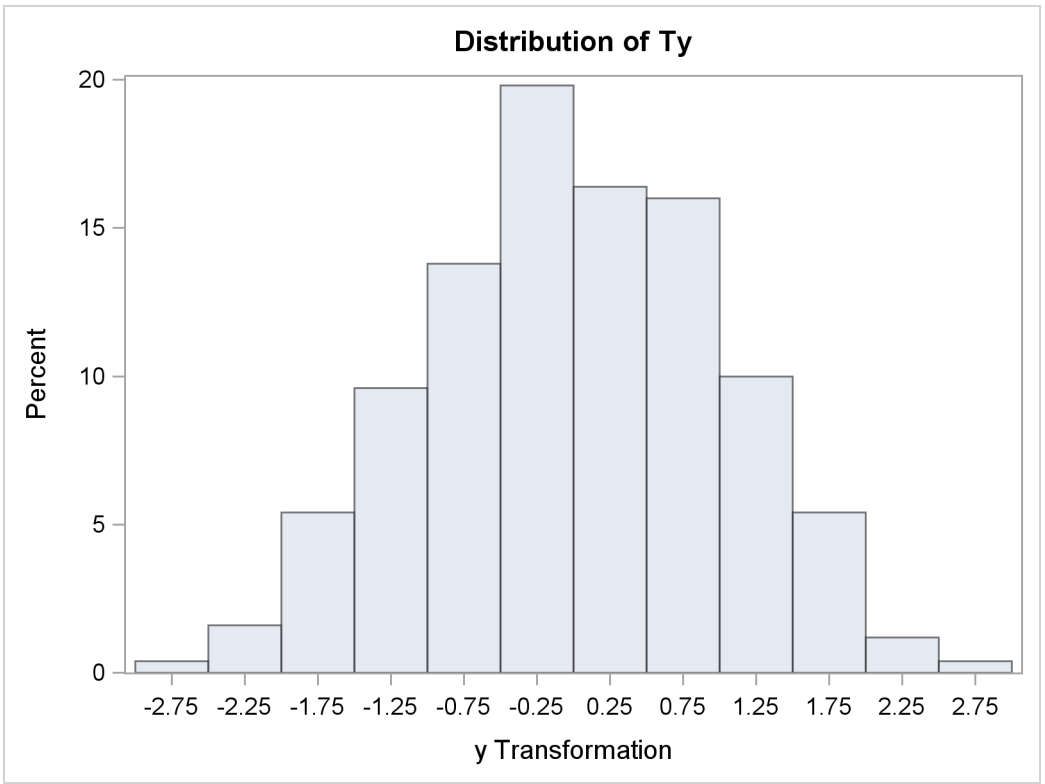


Figure 90.18 *continued*





Using Splines and Knots

This section illustrates some properties of splines. *Splines* are curves, and they are usually required to be continuous and smooth. Splines are usually defined as piecewise polynomials of degree n with function values and first $n-1$ derivatives that agree at the points where they join. The abscissa or X-axis values of the join points are called *knots*. The term "spline" is also used for polynomials (splines with no knots) and piecewise polynomials with more than one discontinuous derivative. Splines with no knots are generally smoother than splines with knots, which are generally smoother than splines with multiple discontinuous derivatives. Splines with few knots are generally smoother than splines with many knots; however, increasing the number of knots usually increases the fit of the spline function to the data. Knots give the curve freedom to bend to more closely follow the data. See Smith (1979) for an excellent introduction to splines.

In this section, an artificial data set is created with a variable y that is a discontinuous function of x. (See Figure 90.20.) Notice that the function has four unconnected parts, each of which is a curve. Notice too that there is an overall quadratic trend—that is, ignoring the shapes of the individual curves, at first the y values tend to decrease as x increases, then y values tend to increase. While these artificial data are clearly not realistic, their distinct pattern helps illustrate how splines work. The following statements create the data set, fit a simple linear regression model, and produce Figure 90.19 through Figure 90.20:

```
title 'An Illustration of Splines and Knots';

* Create in y a discontinuous function of x.;

data a;
   x = -0.000001;
   do i = 0 to 199;
      if mod(i, 50) = 0 then do;
         c = ((x / 2) - 5)**2;
         if i = 150 then c = c + 5;
         y = c;
      end;
      x = x + 0.1;
      y = y - sin(x - c);
      output;
   end;
run;

ods graphics on;

title2 'A Linear Regression Fit';
proc transreg data=a plots=scatter rsquare;
   model identity(y) = identity(x);
run;
```

The R square for the linear regression is 0.1006. The linear fit results in Figure 90.19 show the predicted values of y given x. It can clearly be seen in Figure 90.19 that the linear regression model is not appropriate for these data.

Figure 90.19 A Linear Regression Fit

```
An Illustration of Splines and Knots
A Linear Regression Fit

The TRANSREG Procedure

The TRANSREG Procedure Hypothesis Tests for Identity(y)

R-Square      0.1006
```

Figure 90.19 *continued*

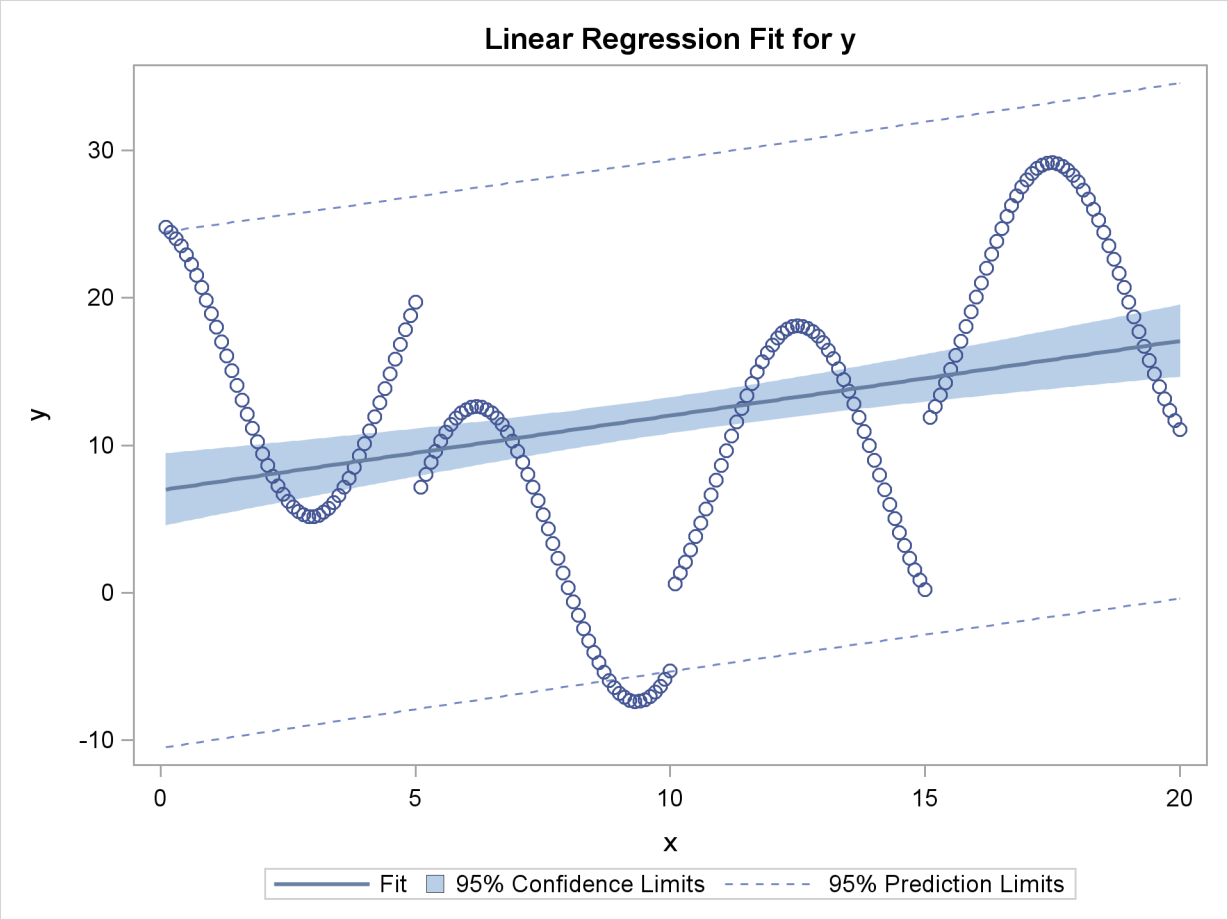
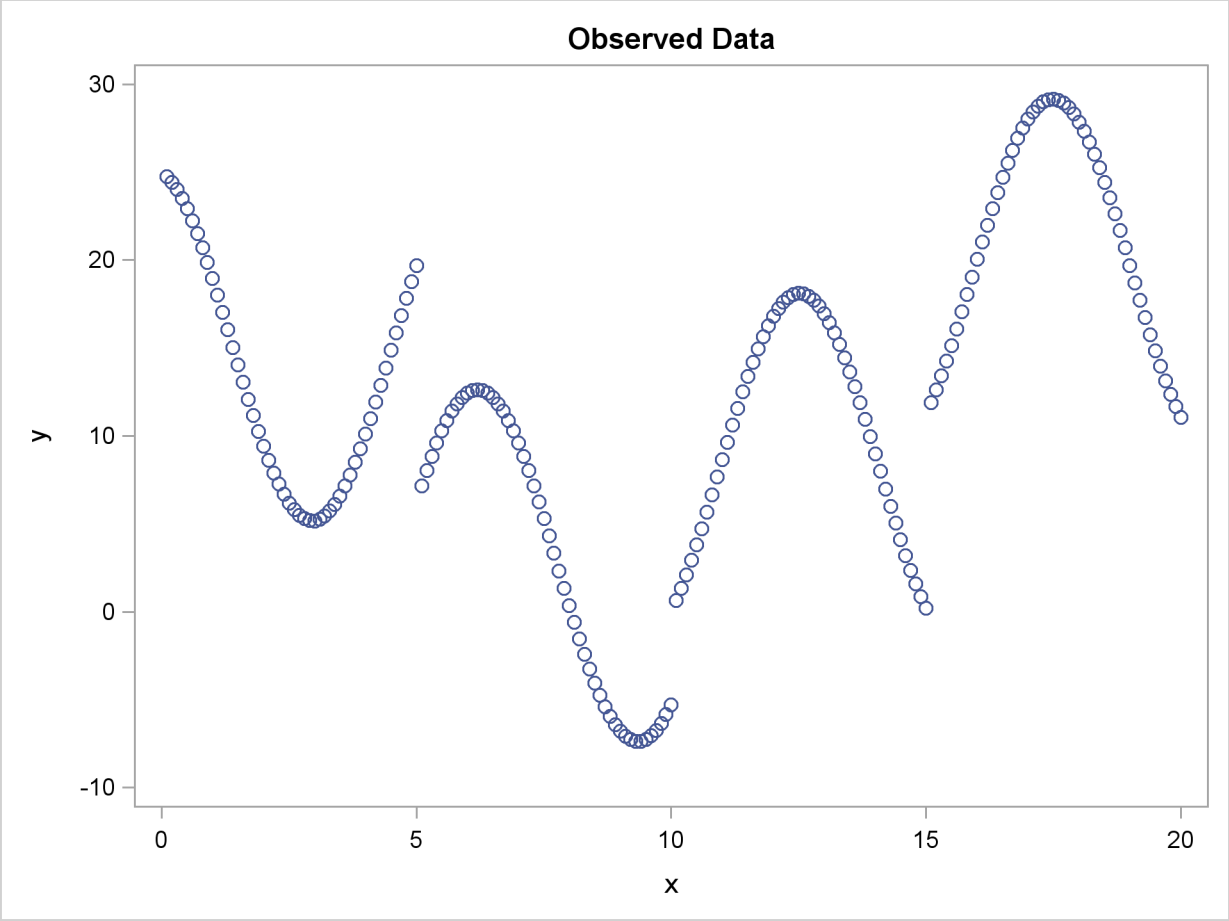


Figure 90.20 The Original Scatter Plot



The next PROC TRANSREG step finds a degree-two spline transformation with no knots, which is a quadratic polynomial. The spline is a weighted sum of a single constant, a single straight line, and a single quadratic curve. The following statements perform the quadratic analysis and produce Figure 90.21:

```
title2 'A Quadratic Polynomial Fit';

proc transreg data=A;
   model identity(y)=spline(x / degree=2);
run;
```

The R square in Figure 90.21 increases from 0.10061, which is the linear fit value from before, to 0.40720. The plot shows that the quadratic regression function does not fit any of the individual curves well, but it does follow the overall trend in the data. Since the overall trend is quadratic, if you were to fit a degree-three spline with no knots (not shown) would increase R square by only a small amount.

Figure 90.21 A Quadratic Polynomial Fit

An Illustration of Splines and Knots
A Quadratic Polynomial Fit

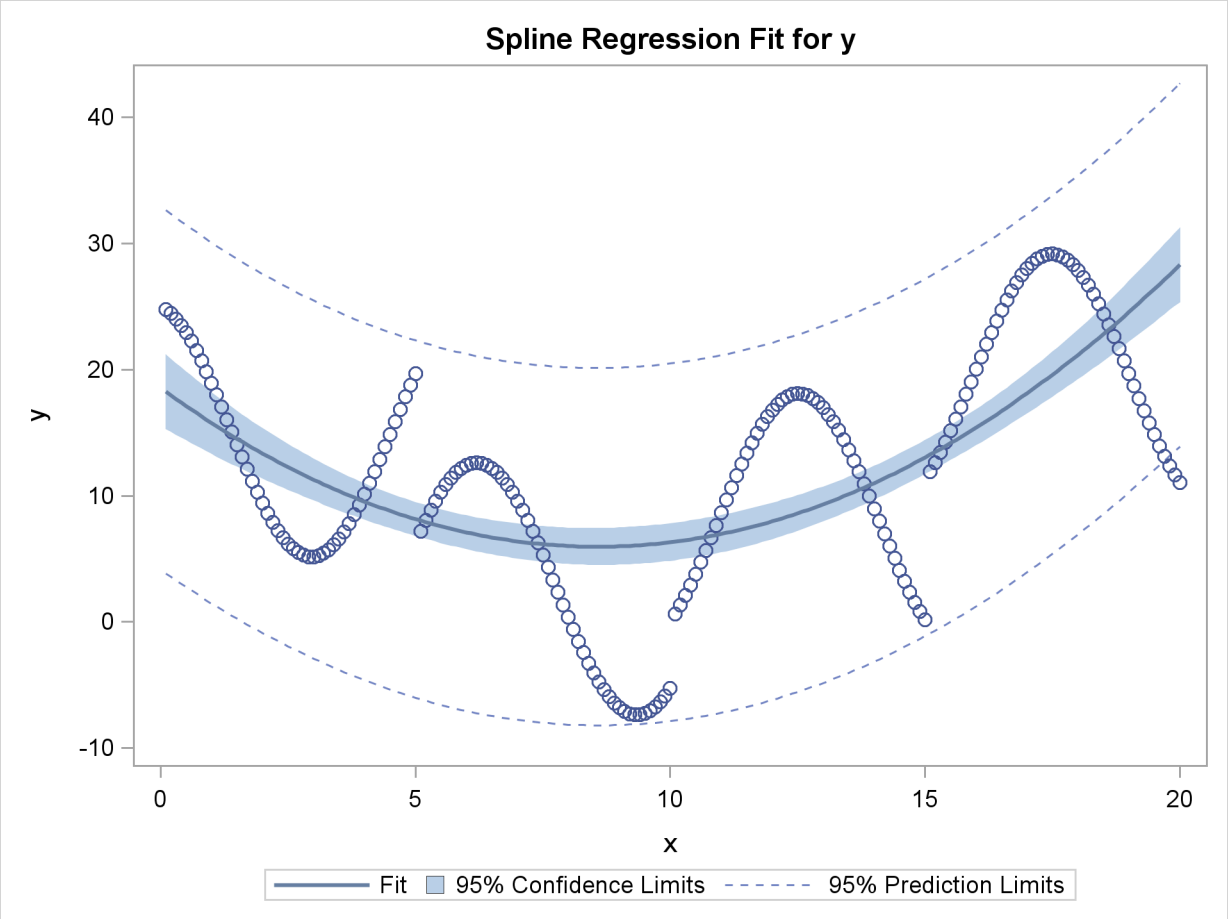
The TRANSREG Procedure

TRANSREG MORALS Algorithm Iteration History for Identity(y)

Iteration Number	Average Change	Maximum Change	R-Square	Criterion Change	Note
1	0.82127	2.77121	0.10061		
2	0.00000	0.00000	0.40720	0.30659	Converged

Algorithm converged.

Figure 90.21 *continued*



The next step uses the default degree of three, for a piecewise cubic polynomial, and requests knots at the known break points, x=5, 10, and 15. This requests a spline that is continuous, has continuous first and second derivatives, and has a third derivative that is discontinuous at 5, 10, and 15. The spline is a weighted sum of a single constant, a single straight line, a single quadratic curve, a cubic curve for the portion of x less than 5, a different cubic curve for the portion of x between 5 and 10, a different cubic curve for the portion of x between 10 and 15, and another cubic curve for the portion of x greater than 15. The following statements fit the spline model and produce Figure 90.22:

```
title2 'A Cubic Spline Fit with Knots at X=5, 10, 15';

proc transreg data=a;
   model identity(y) = spline(x / knots=5 10 15);
run;
```

The new R square in Figure 90.22 is 0.61730. The plot shows that the spline is less smooth than the quadratic polynomial and follows the data more closely than the quadratic polynomial.

Figure 90.22 A Cubic Spline Fit

An Illustration of Splines and Knots
A Cubic Spline Fit with Knots at X=5, 10, 15

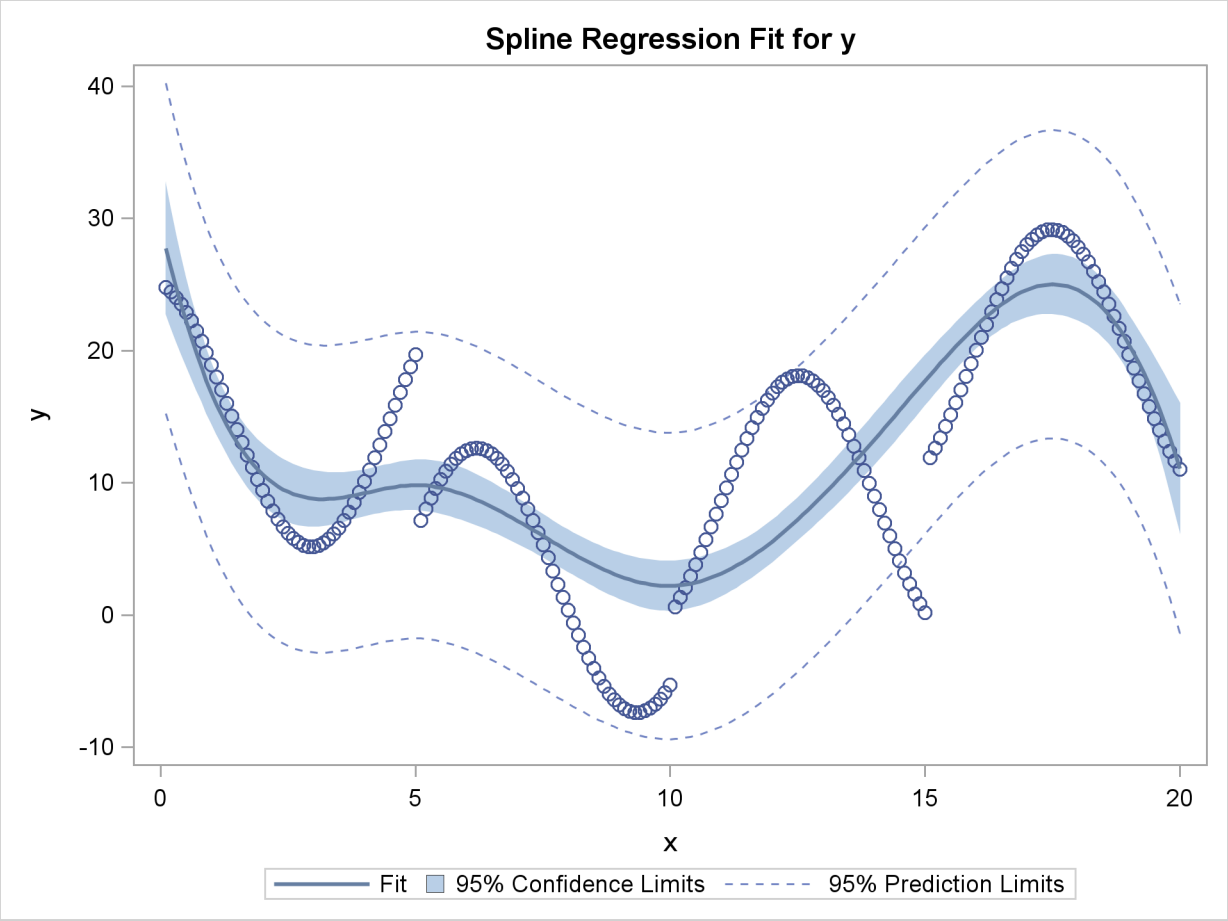
The TRANSREG Procedure

TRANSREG MORALS Algorithm Iteration History for Identity(y)

Iteration Number	Average Change	Maximum Change	R-Square	Criterion Change	Note
1	0.85367	3.88449	0.10061		
2	0.00000	0.00000	0.61730	0.51670	Converged

Algorithm converged.

Figure 90.22 *continued*



The same model could be fit with a DATA step and PROC REG, as follows:

```
data b;              /* A is the data set used by transreg */
   set a(keep=x y);
   x1=x;                        /* x                        */
   x2=x**2;                     /* x squared                */
   x3=x**3;                     /* x cubed                  */
   x4=(x> 5)*((x-5)**3);        /* change in x**3 after  5 */
   x5=(x>10)*((x-10)**3);       /* change in x**3 after 10 */
   x6=(x>15)*((x-15)**3);       /* change in x**3 after 15 */
run;

proc reg;
   model y=x1-x6;
run; quit;
```

The output from these previous statements is not displayed. The assignment statements and comments show how you can construct terms that can be used to fit the same model.

In the next step, each knot is repeated three times, so the first, second, and third derivatives are discontinuous at x=5, 10, and 15, but the spline is continuous at the knots. The spline is a weighted sum of the following:

- a single constant
- a line for the portion of x less than 5
- a quadratic curve for the portion of x less than 5
- a cubic curve for the portion of x less than 5
- a different line for the portion of x between 5 and 10
- a different quadratic curve for the portion of x between 5 and 10
- a different cubic curve for the portion of x between 5 and 10
- a different line for the portion of x between 10 and 15
- a different quadratic curve for the portion of x between 10 and 15
- a different cubic curve for the portion of x between 10 and 15
- another line for the portion of x greater than 15
- another quadratic curve for the portion of x greater than 15
- another cubic curve for the portion of x greater than 15

The spline is continuous since there is not a separate constant or separate intercept in the formula for the spline for each knot. The following statements perform this analysis and produce Figure 90.23:

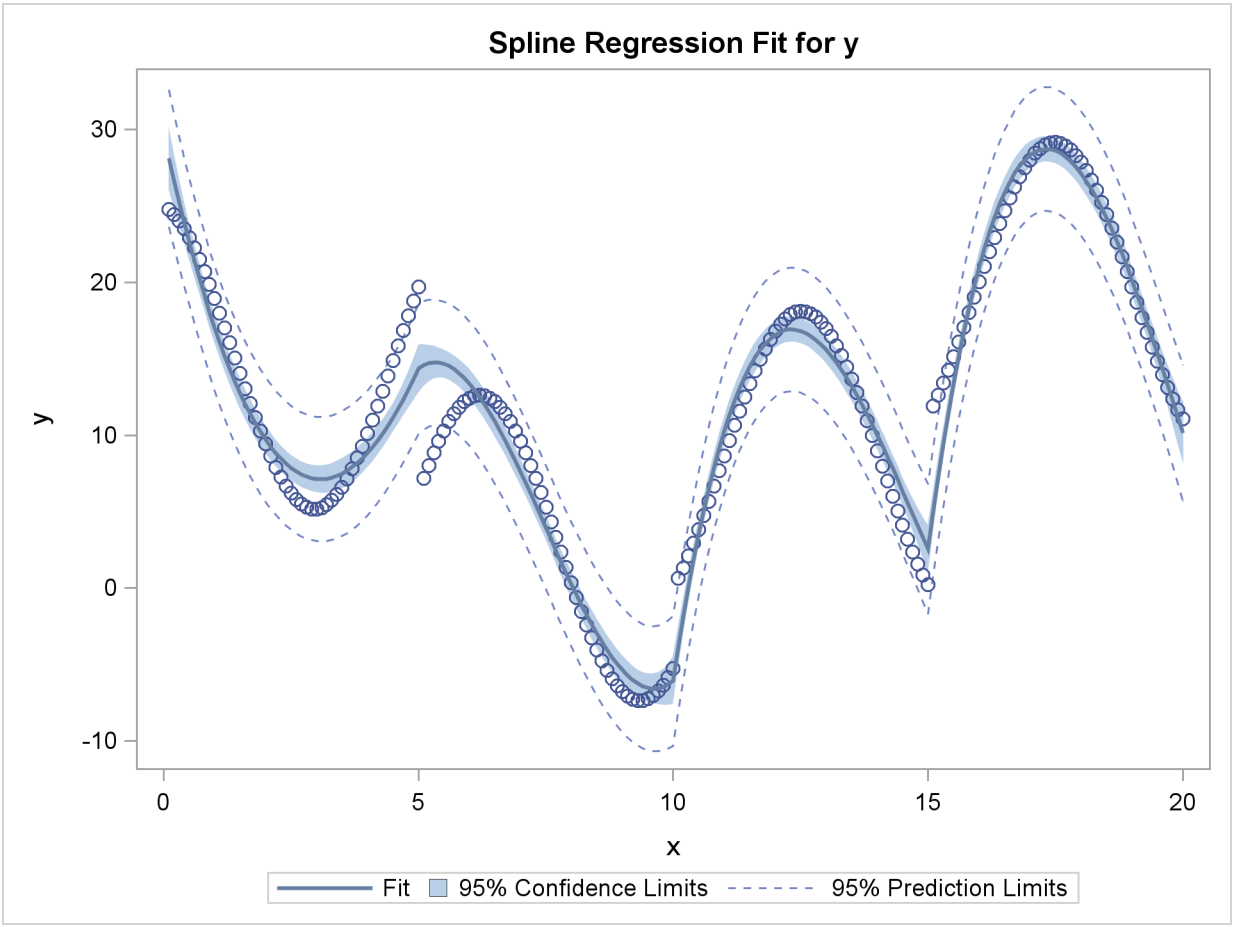
```
title3 'First - Third Derivatives Discontinuous at X=5, 10, 15';

proc transreg data=a;
   model identity(y) = spline(x / knots=5 5 5 10 10 10 15 15 15);
run;
```

Now the R square in Figure 90.23 is 0.95542, and the spline closely follows the data, except at the knots.

Figure 90.23 Spline with Discontinuous Derivatives

```
                An Illustration of Splines and Knots
             A Cubic Spline Fit with Knots at X=5, 10, 15
          First - Third Derivatives Discontinuous at X=5, 10, 15

                          The TRANSREG Procedure

        TRANSREG MORALS Algorithm Iteration History for Identity(y)

Iteration     Average     Maximum                  Criterion
   Number      Change      Change     R-Square       Change     Note
-------------------------------------------------------------------------
        1     0.92492     3.50038      0.10061
        2     0.00000     0.00000      0.95542      0.85481    Converged

Algorithm converged.
```

Figure 90.23 *continued*



The same model could be fit with a DATA step and PROC REG, as follows:

```
data b;
   set a(keep=x y);
   x1=x;                          /* x                          */
   x2=x**2;                       /* x squared                  */
   x3=x**3;                       /* x cubed                    */
   x4=(x>5)   * (x- 5);           /* change in x    after  5 */
   x5=(x>10)  * (x-10);           /* change in x    after 10 */
   x6=(x>15)  * (x-15);           /* change in x    after 15 */
   x7=(x>5)   * ((x-5)**2);       /* change in x**2 after  5 */
   x8=(x>10)  * ((x-10)**2);      /* change in x**2 after 10 */
   x9=(x>15)  * ((x-15)**2);      /* change in x**2 after 15 */
   x10=(x>5)  * ((x-5)**3);       /* change in x**3 after  5 */
   x11=(x>10) * ((x-10)**3);      /* change in x**3 after 10 */
   x12=(x>15) * ((x-15)**3);      /* change in x**3 after 15 */
run;

proc reg;
   model y=x1-x12;
run; quit;
```

The output from these previous statements is not displayed. The assignment statements and comments show how you can construct terms that can be used to fit the same model.

Each knot is repeated four times in the next step. Now the spline function is discontinuous at the knots, and it can follow the data more closely. The following statements perform this analysis and produce Figure 90.24:

```
title3 'Discontinuous Function and Derivatives';

proc transreg data=a;
   model identity(y) = spline(x / knots=5 5 5 5 10 10 10 10
                                     15 15 15 15);
run;
```

Now the R square in Figure 90.24 is 0.99254. In this step, each separate curve is approximated by a cubic polynomial (with no knots within the separate polynomials). (Note, however, that the separate functions are connected in the plot, because PROC TRANSREG cannot currently produce separate functions for a model like this. Usually, you would use a CLASS variable to get separate functions.)

Figure 90.24 Discontinuous Spline Fit

An Illustration of Splines and Knots
A Cubic Spline Fit with Knots at X=5, 10, 15
Discontinuous Function and Derivatives

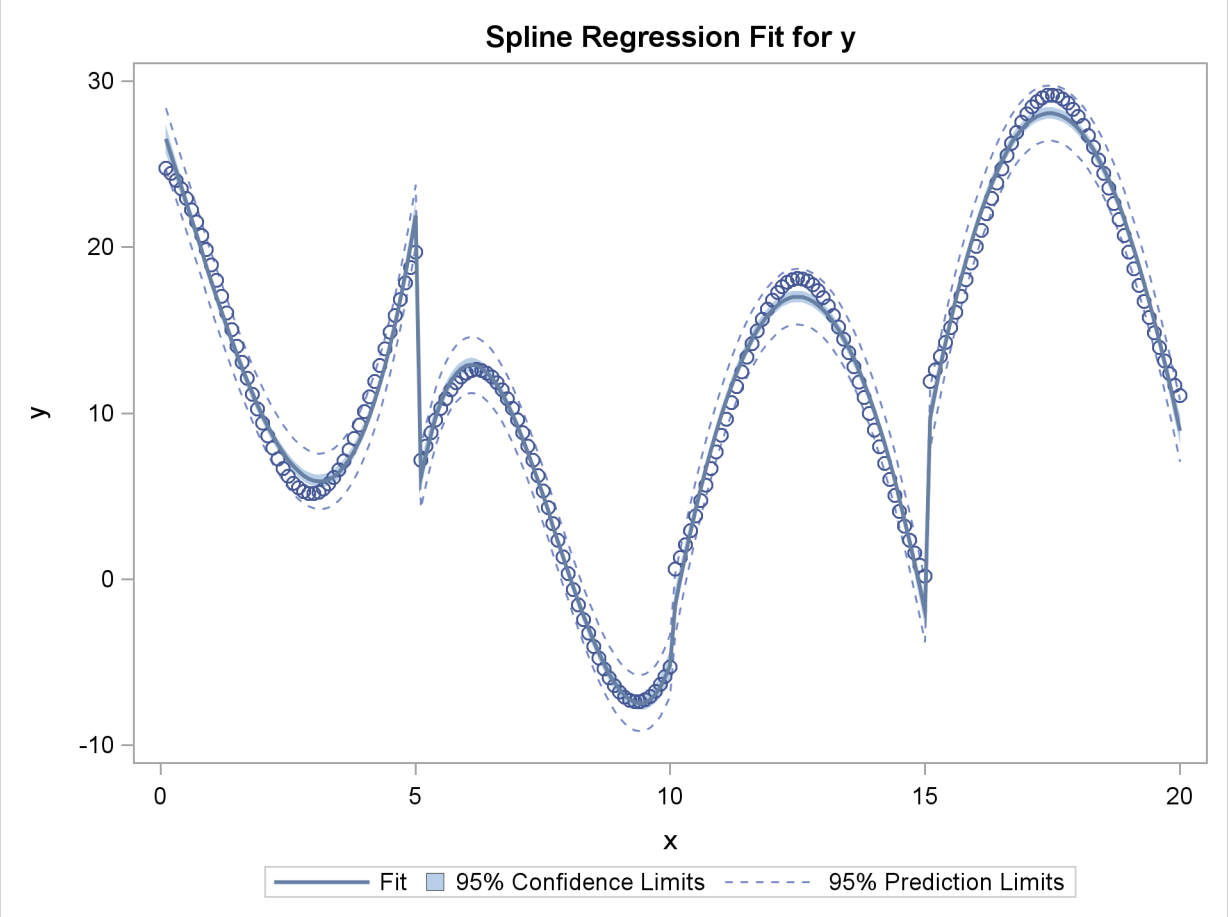
The TRANSREG Procedure

TRANSREG MORALS Algorithm Iteration History for Identity(y)

Iteration Number	Average Change	Maximum Change	R-Square	Criterion Change	Note
1	0.90271	3.29184	0.10061		
2	0.00000	0.00000	0.99254	0.89193	Converged

Algorithm converged.

Figure 90.24 *continued*



To solve this problem with a DATA step and PROC REG, you would need to create all of the variables in the preceding DATA step (the B data set for the piecewise polynomial with discontinuous third derivatives), plus the following three variables:

```
x13=(x >  5);   /* intercept change after  5 */
x14=(x > 10);   /* intercept change after 10 */
x15=(x > 15);   /* intercept change after 15 */
```

The next two examples use the NKNOTS= *t-option* to specify the number of knots but not their location. NKNOTS=4 places knots at the quintiles, whereas NKNOTS=9 places knots at the deciles. The spline and its first two derivatives are continuous. The following statements produce Figure 90.25 and Figure 90.26:

```
title3 'Four Knots';

proc transreg data=a;
   model identity(y) = spline(x / nknots=4);
run;

title3 'Nine Knots';

proc transreg data=a;
   model identity(y) = spline(x / nknots=9);
run;

ods graphics off;
```

The R-square values displayed in Figure 90.25 and Figure 90.26 are 0.74450 and 0.95256, respectively. Even though the knots are not optimally placed, the spline can closely follow the data with NKNOTS=9.

Figure 90.25 Spline Fit with Knots at the Quintiles

```
                    An Illustration of Splines and Knots
                 A Cubic Spline Fit with Knots at X=5, 10, 15
                                Four Knots

                            The TRANSREG Procedure

          TRANSREG MORALS Algorithm Iteration History for Identity(y)

  Iteration     Average     Maximum                 Criterion
     Number      Change      Change     R-Square        Change     Note
  ----------------------------------------------------------------------
          1     0.90305     4.46027      0.10061
          2     0.00000     0.00000      0.74450       0.64389    Converged

  Algorithm converged.
```

Figure 90.25 *continued*

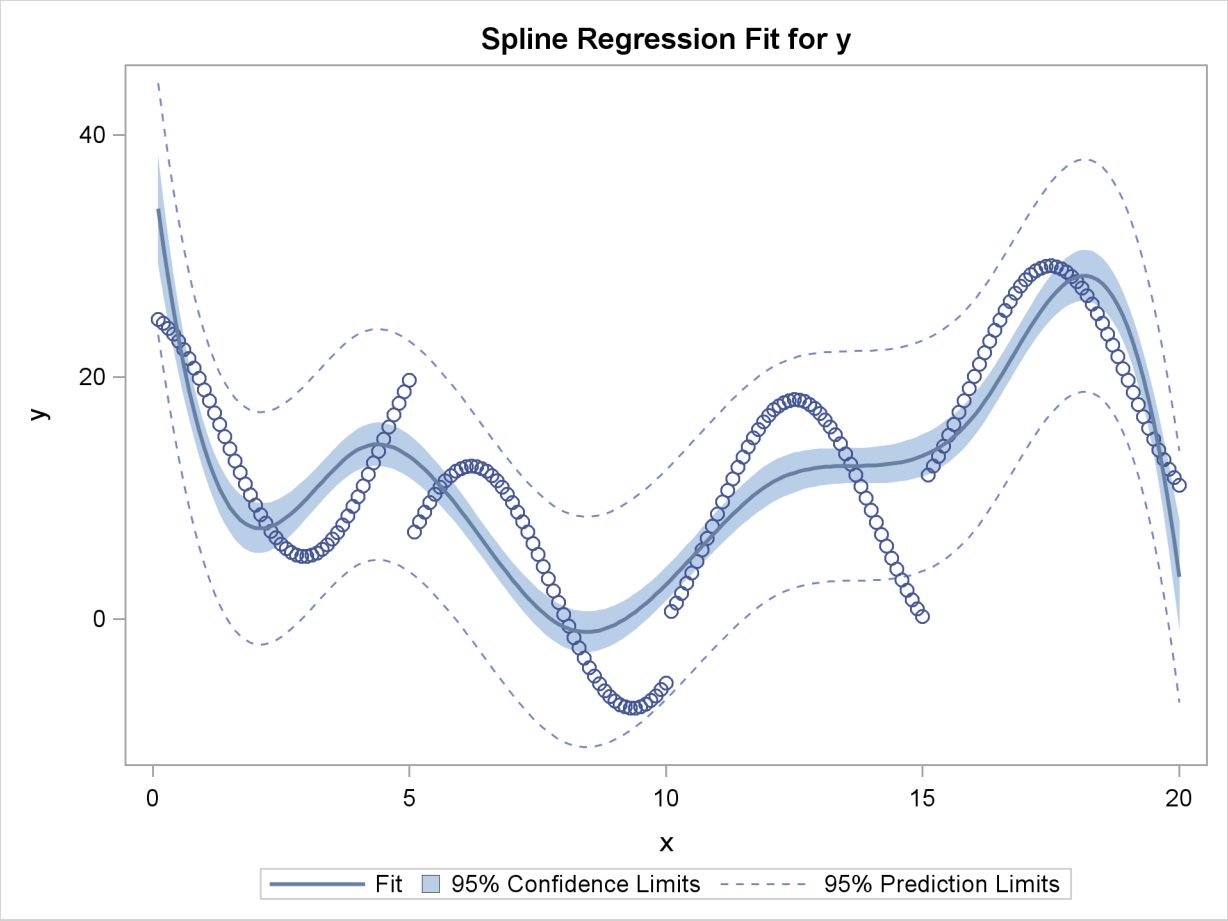


Figure 90.26 Spline Fit with Knots at the Deciles

An Illustration of Splines and Knots
A Cubic Spline Fit with Knots at X=5, 10, 15
Nine Knots

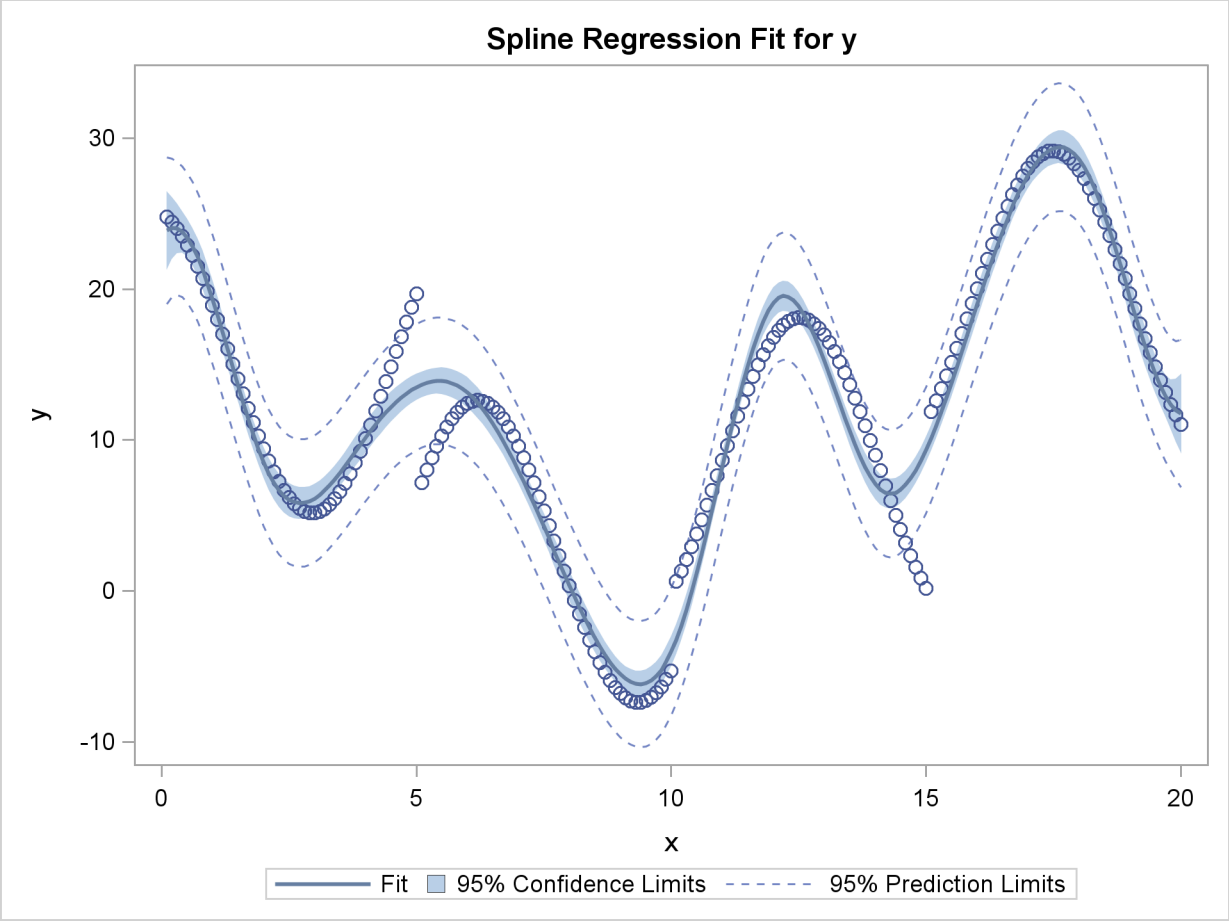
The TRANSREG Procedure

TRANSREG MORALS Algorithm Iteration History for Identity(y)

Iteration Number	Average Change	Maximum Change	R-Square	Criterion Change	Note
1	0.94832	3.03488	0.10061		
2	0.00000	0.00000	0.95256	0.85196	Converged

Algorithm converged.

Figure 90.26 *continued*



Scoring Spline Variables

This section shows you how to find spline transformations of variables in one data set and apply the same transformations to variables in another data set. This is illustrated with artificial data. In these data sets, the variable y is approximately a linear function of nonlinear transformations of the variables x, w, and z. The model is fit using data set X, and those results are used to score data set Z. The following statements create the two data sets:

```
title 'An Illustration of Splines and Knots';
title2 'Scoring Spline Variables';

data x;
   do i = 1 to 5000;
      w = normal(7);
      x = normal(7);
      z = normal(7);
      y = w * w + log(5 + x) + sin(z) + normal(7);
      output;
   end;
run;
```

```
data z;
   do i = 1 to 5000;
      w = normal(1);
      x = normal(1);
      z = normal(1);
      y = w * w + log(5 + x) + sin(z) + normal(1);
      output;
   end;
run;
```

First, you run PROC TRANSREG to fit the transformation regression model asking for spline transformations of the three independent variables. You must use the EXKNOTS= *t-option*, because you need to use the same knots, both interior and exterior, with both data sets. By default, the exterior knots will be different if the minima and maxima are different in the two data sets, so you get the wrong results if you do not specify the EXKNOTS= *t-option* with values less than the minima and greater than the maxima of the six x, y, and w variables. If the ranges in all three pairs were different, you would need separate spline transformation for each variable with different knot and exterior knot specifications. The following statements fit the spline model:

```
proc transreg data=x solve details ss2;
   ods output splinecoef=c;
   model identity(y) = spline(w x z / knots=-1.5 to 1.5 by 0.5
                                      exknots=-5 5);
   output out=d;
run;
```

The results of this step are not displayed. The nonprinting "SplineCoef" table is output to a SAS data set. This data set contains the coefficients that were used to get the spline transformations and can be used to transform variables in other data sets. These coefficients are also in the details table. However, in the "SplineCoef" table, they are in a form directly suitable for use with PROC SCORE.

The next step reads the second input data set, Z, and generates an output data set with the B-spline basis for each of the variables:

```
proc transreg data=z design;
   model bspl(w x z / knots=-1.5 to 1.5 by 0.5 exknots=-5 5);
   output out=b;
run;
```

Note that the same interior and exterior knots are used in both of the previous steps. The next three steps score the B-spline bases created in the previous step by using the coefficients generated in the first PROC TRANSREG step. PROC SCORE is run once for each SPLINE variable in the statements that follow:

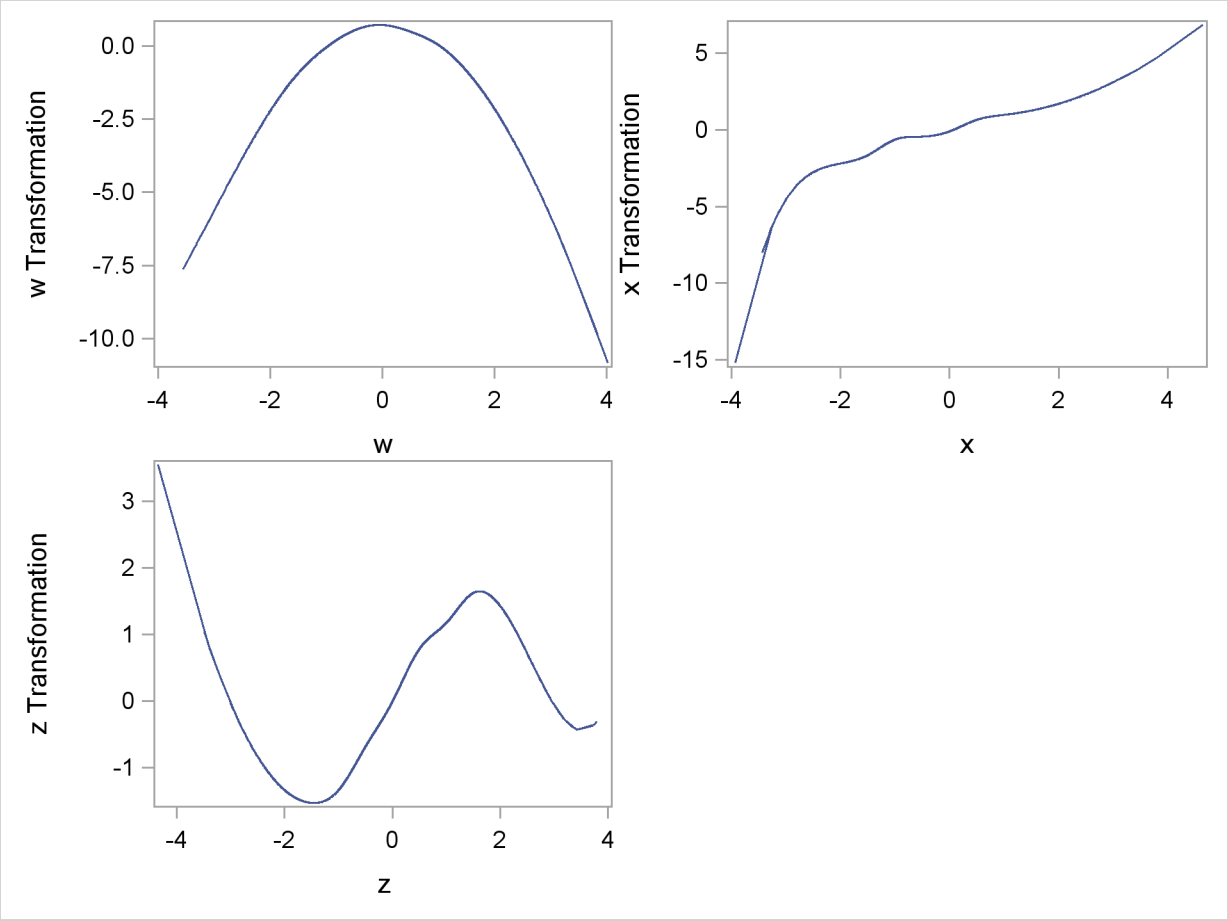
```
proc score data=b score=c out=o1(rename=(spline=bw w=nw));
   var w:;
run;

proc score data=b score=c out=o2(rename=(spline=bx x=nx));
   var x:;
run;

proc score data=b score=c out=o3(rename=(spline=bz z=nz));
   var z:;
run;
```

The following steps merge the three transformations with the original data and plot the results:

```
data all;
   merge d(keep=w x z tw tx tz) o1(keep=nw bw)
         o2(keep=nx bx) o3(keep=nz bz);
run;

proc template;
   define statgraph twobytwo;
   begingraph;
      layout lattice / rows=2 columns=2;
            layout overlay;
               seriesplot y=tw x=w  / connectorder=xaxis;
               seriesplot y=bw x=nw / connectorder=xaxis;
            endlayout;
            layout overlay;
               seriesplot y=tx x=x  / connectorder=xaxis;
               seriesplot y=bx x=nx / connectorder=xaxis;
            endlayout;
            layout overlay;
               seriesplot y=tz x=z  / connectorder=xaxis;
               seriesplot y=bz x=nz / connectorder=xaxis;
            endlayout;
         endlayout;
      endgraph;
   end;
run;

proc sgrender data=all template=twobytwo;
run;
```

The plots in Figure 90.27 show that the two transformations for each variable, original and scored, are the same function. The two functions in each plot are on top of each other and are indistinguishable. Furthermore, PROC TRANSREG found the functional forms that were used to generate the data: quadratic for w, log for x, and sine for z.

Figure 90.27 Scoring Spline Variables Example



The next statements show how to run PROC TRANSREG, output the interior and exterior knots to an output data set with ODS, extract the knots, and use them in a DATA step to re-create the B-spline basis that PROC TRANSREG makes. In practice, you would never need to use a DATA step to make the B-spline basis since PROC TRANSREG does it automatically. The following statements show how you could do it yourself:

```
data x;
   input x @@;
   datalines;
1 2 3 4 5 6 7 8 9 10
;
ods output details=d;
proc transreg details design;
   model bspline(x / nkn=3);
   output out=y;
run;
```

```
%let k = 0;
data d;
   set d;
   length d $ 20;
   retain d ' ';
   if description ne ' ' then d = description;
   if d = 'Degree' then call symput('d', compress(formattedvalue));
   if d = 'Number of Knots'
      then call symput('k', compress(formattedvalue));
   if index(d, 'Knots') and not index(d, 'Number');
   keep d numericvalue;
run;

%let nkn = %eval(&d * 2 + &k); /* total number of knots   */
%let nb  = %eval(&d + 1 + &k); /* number of cols in basis */

proc transpose data=d out=k(drop=_name_) prefix=Knot; run;

proc print; format k: 20.16; run;

data b(keep=x:);
   if _n_ = 1 then set k; /* read knots from transreg */
   array k[&nkn] knot1-knot&nkn;        /* knots      */
   array b[&nb] x_0 - x_%eval(&nb - 1); /* basis      */
   array w[%eval(2 * &d)];              /* work       */
   set x;
   do i = 1 to &nb; b[i] = 0; end;

   * find the index of first knot greater than current data value;
   do ki = 1 to &nkn while(k[ki] le x); end;
   kki = ki - &d - 1;

   * make the basis;
   b[1 + kki] = 1;
   do j = 1 to &d;
      w[&d + j] = k[ki + j - 1] - x;
      w[j] = x - k[ki - j];
      s = 0;
      do i = 1 to j;
         t = w[&d + i] + w[j + 1 - i];
         if t ne 0.0 then t = b[i + kki] / t;
         b[i + kki] = s + w[&d + i] * t;
         s = w[j + 1 - i] * t;
      end;
      b[j + 1 + kki] = s;
   end;
run;

proc compare data=y(keep=x:) compare=b
   criterion=1e-12 note nosummary;
   title3 "should be no differences";
run;
```

The output from these steps is not shown. There are several things to note about the DATA step. It produces the same basis as PROC TRANSREG only because it uses exactly the same interior and exterior knots. The exterior knots (0.999999999999 and 10.000000000001) are just slightly smaller than 1 (the minimum in x) and just slightly greater than 10 (the maximum in x). Both exterior knots appear in the list three times, because a cubic (degree 3) polynomial was requested. The complete knot list is: 0.999999999999 0.999999999999 0.999999999999 3 6 8 10.000000000001 10.000000000001 10.000000000001. The exterior knots do not have any particular interpretation, but they are needed by the algorithm to construct the proper basis. The construction method computes differences between each value and the nearby knots. The algorithm that makes the B-spline basis is not very obvious, particularly compared to the polynomial spline basis. However, the B-spline basis is much better behaved numerically than a polynomial-spline basis, so that is why it is used.

Linear and Nonlinear Regression Functions

This section shows how to use PROC TRANSREG in simple regression (one dependent variable and one independent variable) to find the optimal regression line, a nonlinear but monotone regression function, and a nonlinear and nonmonotone regression function. To find a linear regression function, specify the IDENTITY transformation of the independent variable. For a monotone curve, specify the MSPLINE transformation of the independent variable. To relax the monotonicity constraint, specify the SPLINE transformation. You can get more flexibility in spline functions by specifying knots. The more knots you specify, the more freedom the function has to follow minor variations in the data. This example uses artificial data. While these artificial data are clearly not realistic, their distinct pattern helps illustrate how splines work. The following statements generate the data and produce Figure 90.28 through Figure 90.31:

```
title 'Linear and Nonlinear Regression Functions';

* Generate an Artificial Nonlinear Scatter Plot;
data a;
   do i = 1 to 500;
      x = i / 2.5;
      y = -((x/50)-1.5)**2 + sin(x/8) + sqrt(x)/5 + 2*log(x) + cos(x);
      x = x / 21;
      if y > 2 then output;
   end;
run;

ods graphics on;
ods select fitplot(persist);

title2 'Linear Regression';

proc transreg data=a;
   model identity(y)=identity(x);
run;
```

```
title2 'A Monotone Regression Function';

proc transreg data=a;
   model identity(y)=mspline(x / nknots=9);
run;

title2 'A Nonlinear Regression Function';

proc transreg data=a;
   model identity(y)=spline(x / nknots=9);
run;

title2 'A Nonlinear Regression Function, 100 Knots';

proc transreg data=a;
   model identity(y)=spline(x / nknots=100);
run;

ods graphics off;
ods select all;
```

Figure 90.28 Linear Regression

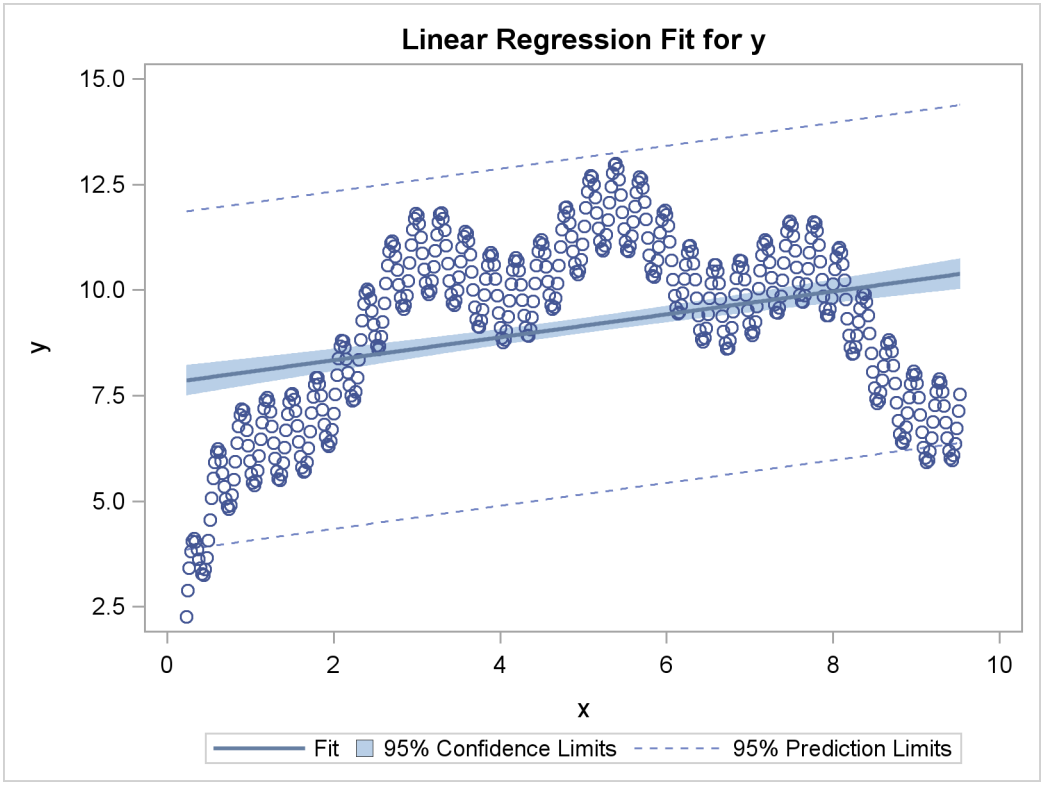


Figure 90.29 A Monotone Regression Function

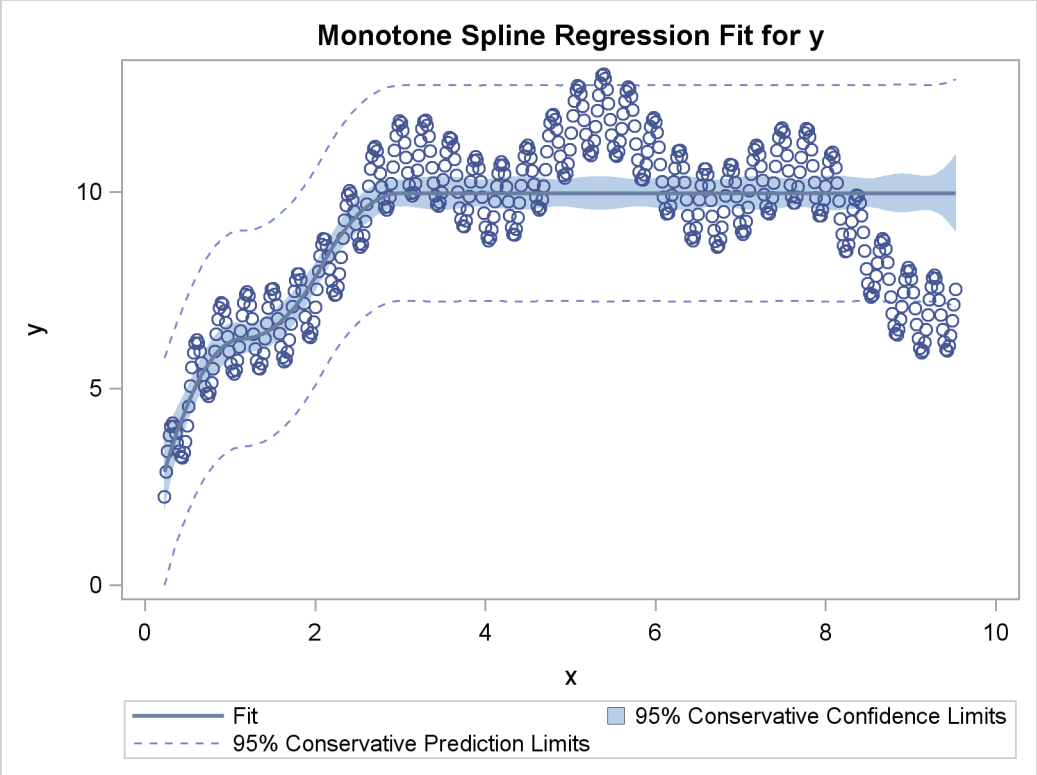


Figure 90.30 A Nonlinear Regression Function

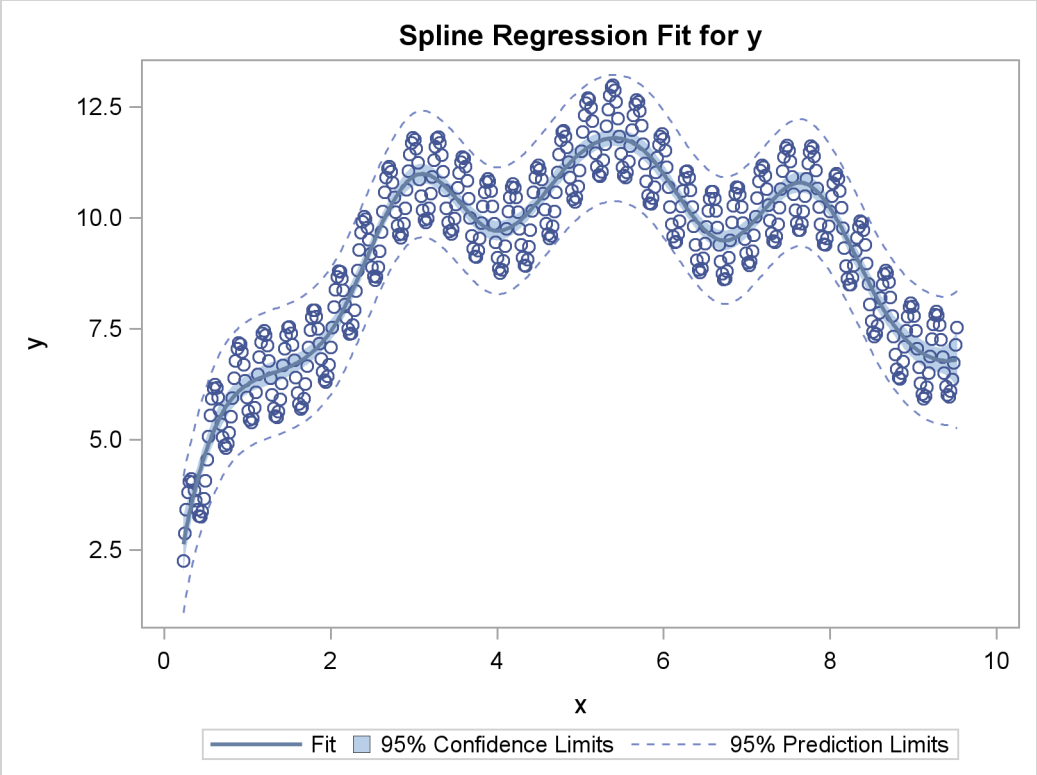
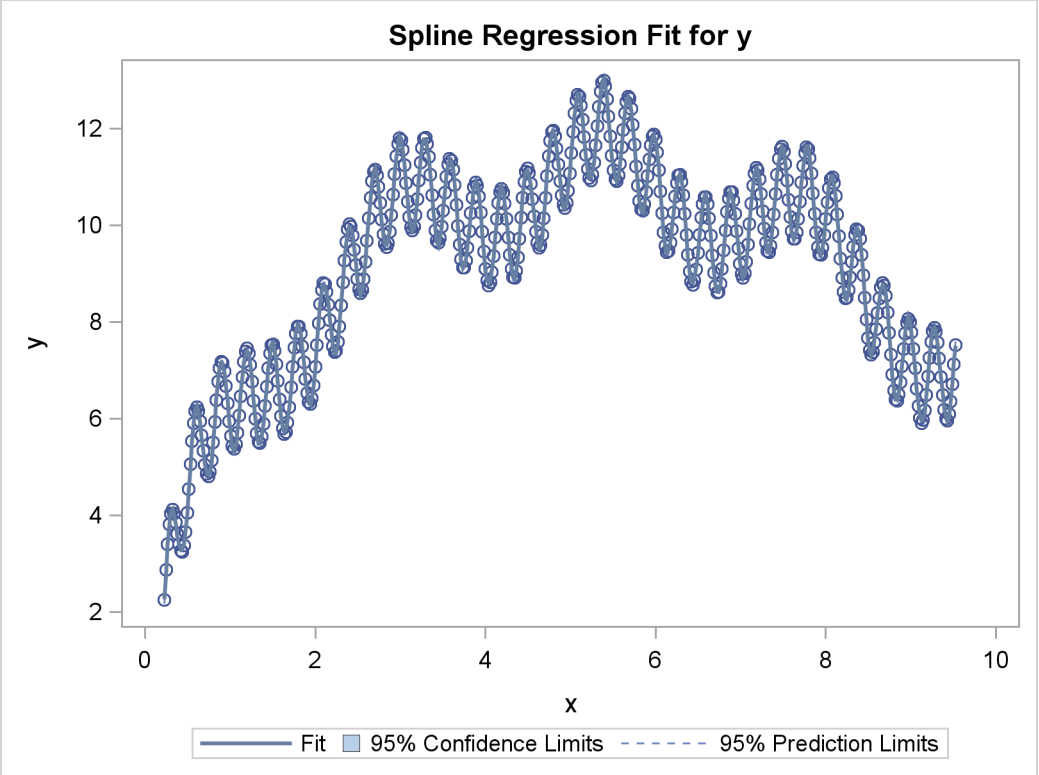


Figure 90.31 A Less-Smooth Nonlinear Regression Function



The squared correlation is only 0.15 for the linear regression in Figure 90.28. Clearly, a simple linear regression model is not appropriate for these data. By relaxing the constraints placed on the regression line, the proportion of variance accounted for increases from 0.15 (linear) to 0.61 (monotone in Figure 90.29) to 0.90 (nonlinear but smooth in Figure 90.30) to almost 1.0 with 100 knots (nonlinear and not very smooth in Figure 90.28). Relaxing the linearity constraint permits the regression function to bend and more closely follow the right portion of the scatter plot. Relaxing the monotonicity constraint permits the regression function to follow the periodic portion of the left side of the plot more closely. The nonlinear MSPLINE transformation is a quadratic spline with knots at the deciles. The first nonlinear nonmonotonic SPLINE transformation is a cubic spline with knots at the deciles.

Different knots and different degrees would produce slightly different results. The two nonlinear regression functions could be closely approximated by simpler piecewise linear regression functions. The monotone function could be approximated by a two-piece line with a single knot at the elbow. The first nonmonotone function could be approximated by a six-piece function with knots at the five elbows.

With this type of problem (one dependent variable with no missing values that is not transformed and one independent variable that is nonlinearly transformed), PROC TRANSREG always iterates exactly twice (although only one iteration is necessary). The first iteration reports the R square for the linear regression line and finds the optimal transformation of x. Since the data change in the first iteration, a second iteration is performed, which reports the R square for the final nonlinear regression function, and zero data change. The predicted values, which are a linear function of the optimal transformation of x, contain the Y coordinates for the nonlinear regression function.

The variance of the predicted values divided by the variance of y is the R square for the fit of the nonlinear regression function. When x is monotonically transformed, the transformation of x is always monotonically increasing, but the predicted values increase if the correlation is positive and decrease for negative correlations.

Simultaneously Fitting Two Regression Functions

One application of ordinary multiple regression is fitting two or more regression lines through a single scatter plot. With PROC TRANSREG, this application can easily be generalized to fit separate or parallel curves. To illustrate, consider a data set with two groups and a group membership variable g that has the value 1 for one group and 2 for the other group. The data set also has a continuous independent variable x and a continuous dependent variable y. When g is crossed with x, the variables g1x and g2x both have a large partition of zeros. For this reason, the KNOTS= *t-option* is specified instead of the NKNOTS= *t-option*. (The latter would put a number of knots in the partition of zeros.) The following example generates an artificial data set with two curves. While these artificial data are clearly not realistic, their distinct pattern helps illustrate how fitting simultaneous regression functions works. The following statements generate data and show how PROC TRANSREG fits lines, curves, and monotone curves through a scatter plot:

```
title 'Separate Curves, Separate Intercepts';

data a;
   do x = -2 to 3 by 0.025;
      g = 1;
      y = 8*(x*x + 2*cos(x*6)) + 15*normal(7654321);
      output;
      g = 2;
      y = 4*(-x*x + 4*sin(x*4)) - 40 + 15*normal(7654321);
      output;
   end;
run;

ods graphics on;
ods select fitplot(persist);

title 'Parallel Lines, Separate Intercepts';

proc transreg data=a solve;
   model identity(y)=class(g) identity(x);
run;

title 'Parallel Monotone Curves, Separate Intercepts';

proc transreg data=a;
   model identity(y)=class(g) mspline(x / knots=-1.5 to 2.5 by 0.5);
run;
```

```
title 'Parallel Curves, Separate Intercepts';

proc transreg data=a solve;
   model identity(y)=class(g) spline(x / knots=-1.5 to 2.5 by 0.5);
run;

title 'Separate Slopes, Same Intercept';

proc transreg data=a;
   model identity(y)=class(g / zero=none) * identity(x);
run;

title 'Separate Monotone Curves, Same Intercept';

proc transreg data=a;
   model identity(y) = class(g / zero=none) *
                       mspline(x / knots=-1.5 to 2.5 by 0.5);
run;

title 'Separate Curves, Same Intercept';

proc transreg data=a solve;
   model identity(y) = class(g / zero=none) *
                       spline(x / knots=-1.5 to 2.5 by 0.5);
run;

title 'Separate Slopes, Separate Intercepts';

proc transreg data=a;
   model identity(y) = class(g / zero=none) | identity(x);
run;

title 'Separate Monotone Curves, Separate Intercepts';

proc transreg data=a;
   model identity(y) = class(g / zero=none) |
                       mspline(x / knots=-1.5 to 2.5 by 0.5);
run;

title 'Separate Curves, Separate Intercepts';

proc transreg data=a solve;
   model identity(y) = class(g / zero=none) |
                       spline(x / knots=-1.5 to 2.5 by 0.5);
run;

ods graphics off;
ods select all;
```

The previous statements produce Figure 90.32 through Figure 90.40. Only the fit plots are generated and displayed.

Figure 90.32 Parallel Lines, Separate Intercepts

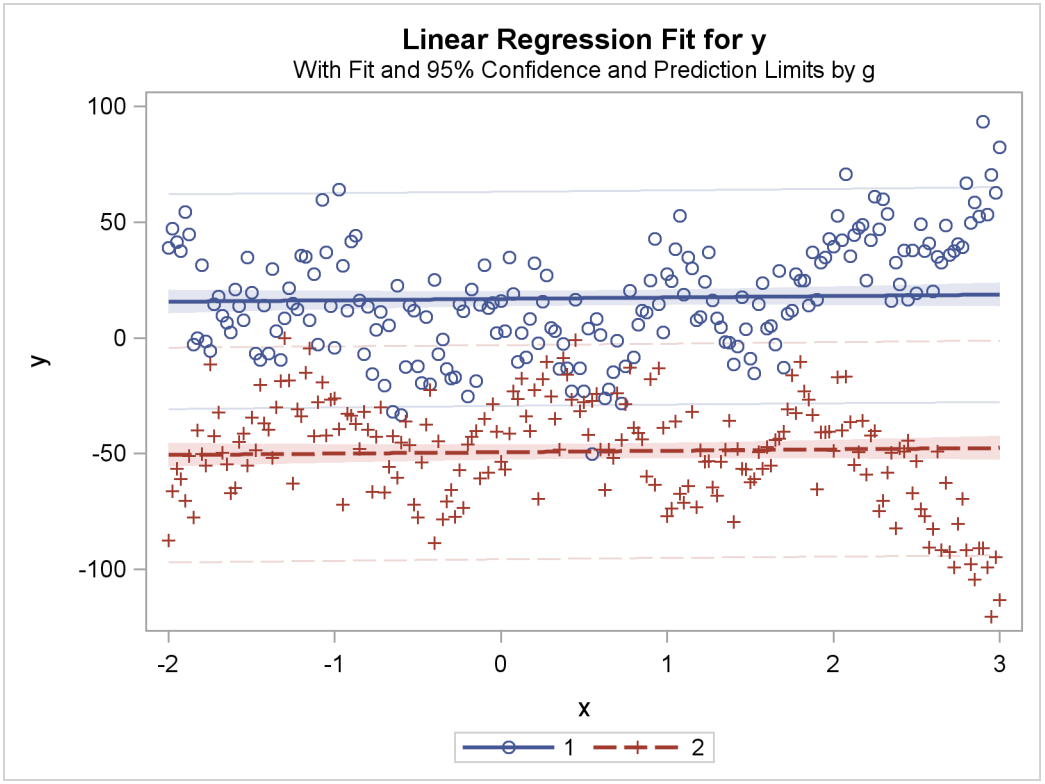


Figure 90.33 Parallel Monotone Curves, Separate Intercepts

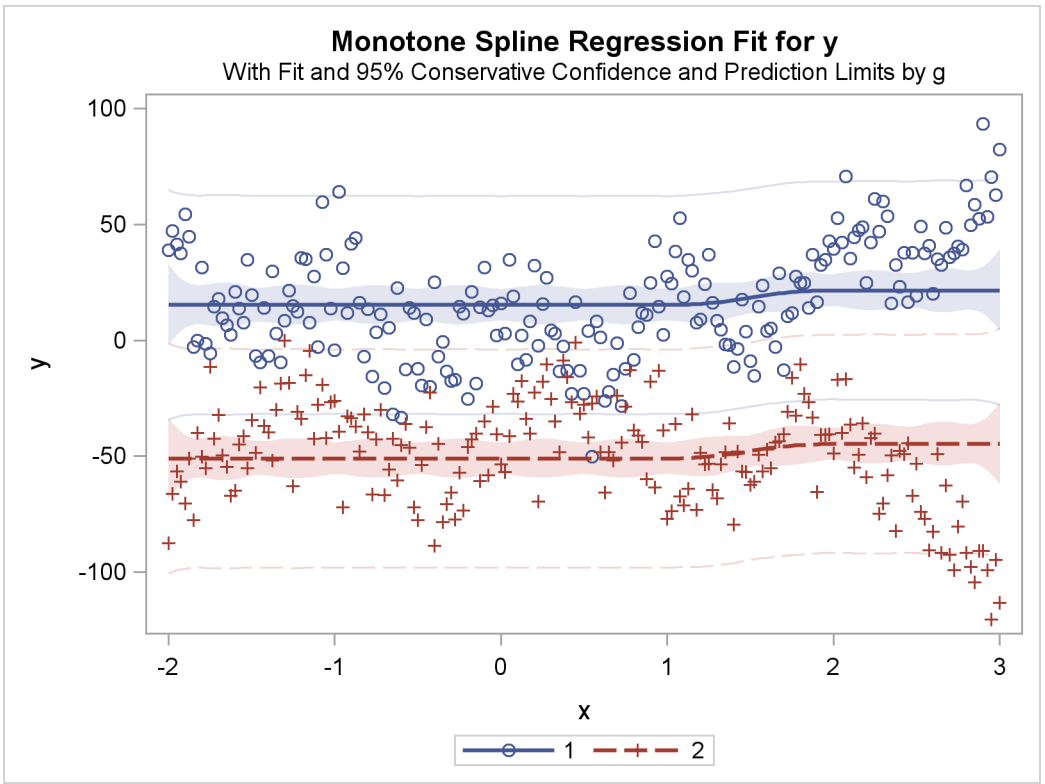


Figure 90.34 Parallel Curves, Separate Intercepts

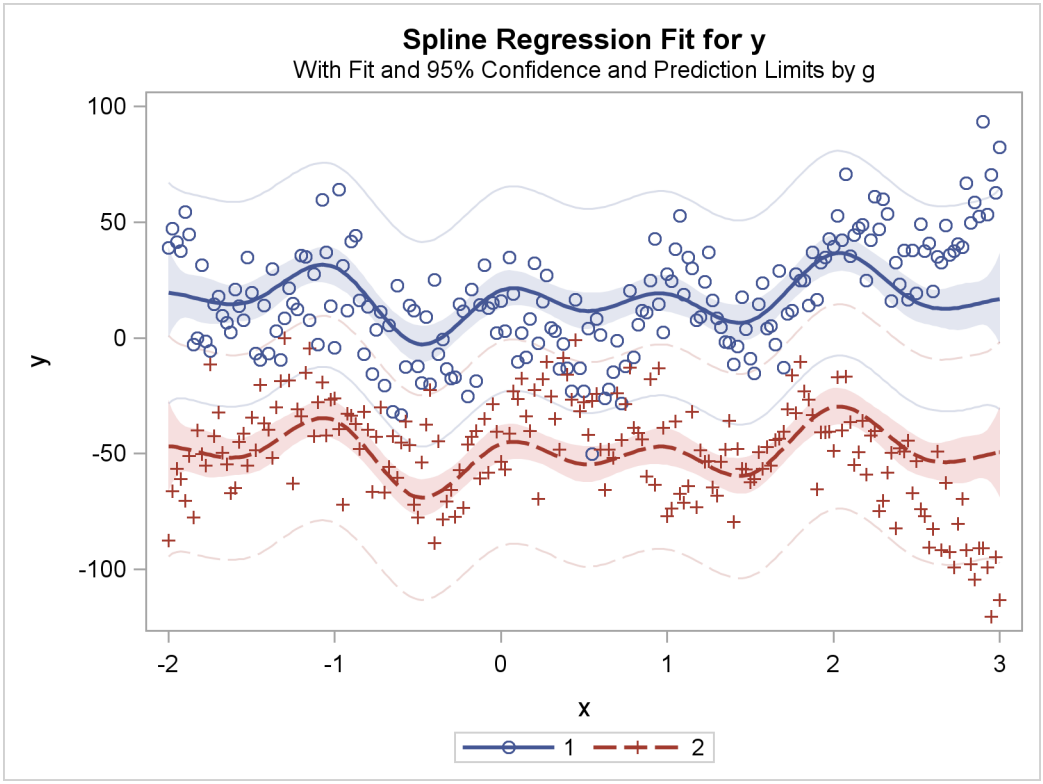


Figure 90.35 Separate Slopes, Same Intercept

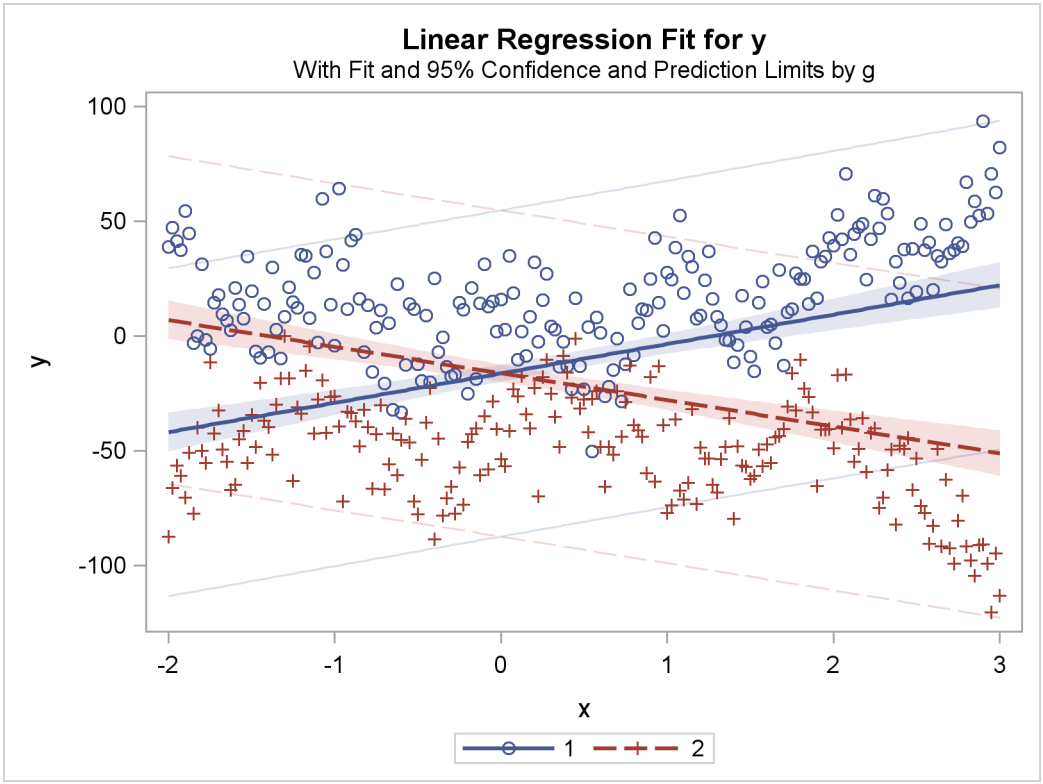


Figure 90.36 Separate Monotone Curves, Same Intercept

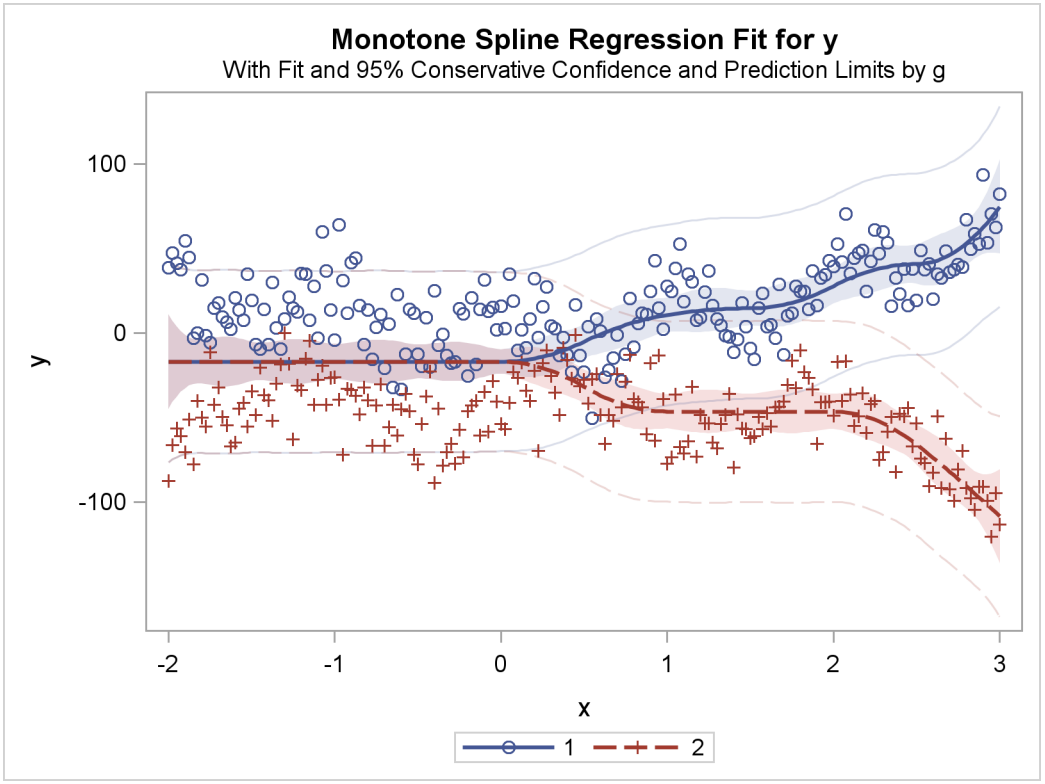


Figure 90.37 Separate Curves, Same Intercept

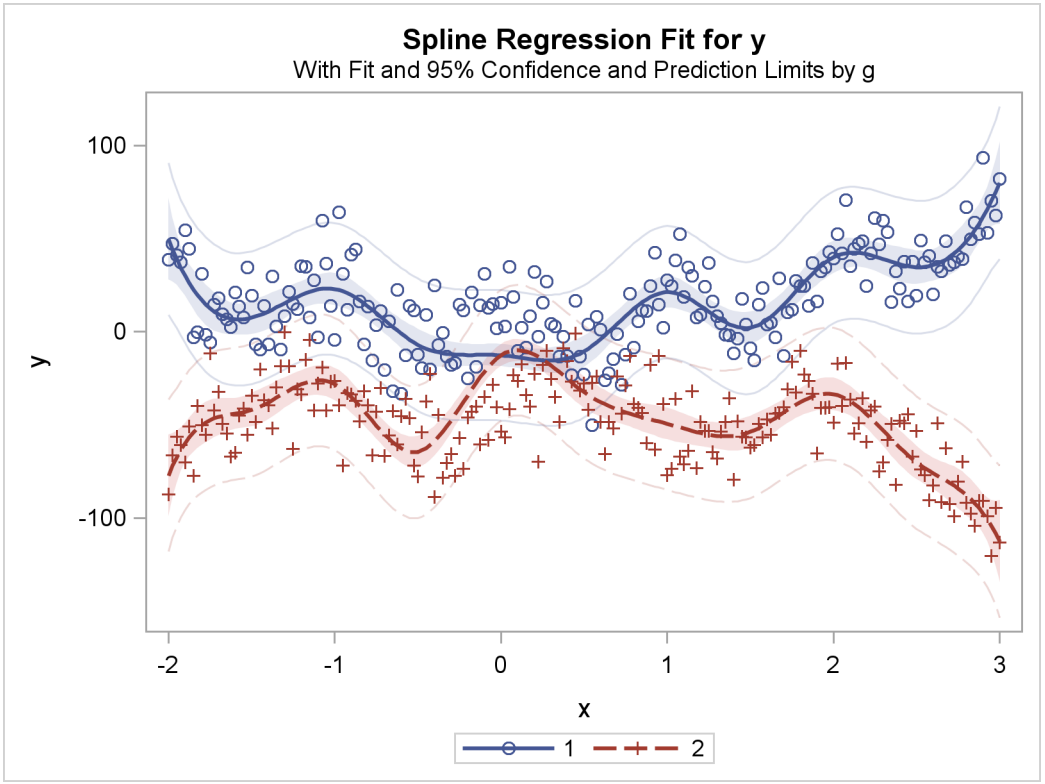


Figure 90.38 Separate Slopes, Separate Intercepts

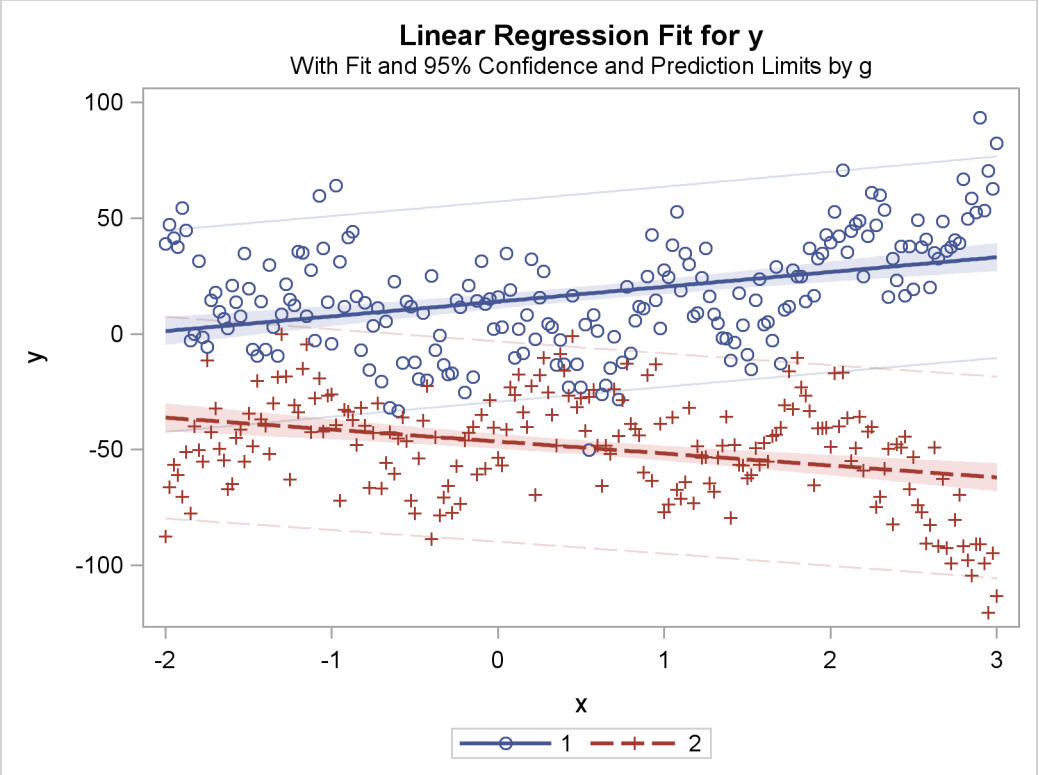


Figure 90.39 Separate Monotone Curves, Separate Intercepts

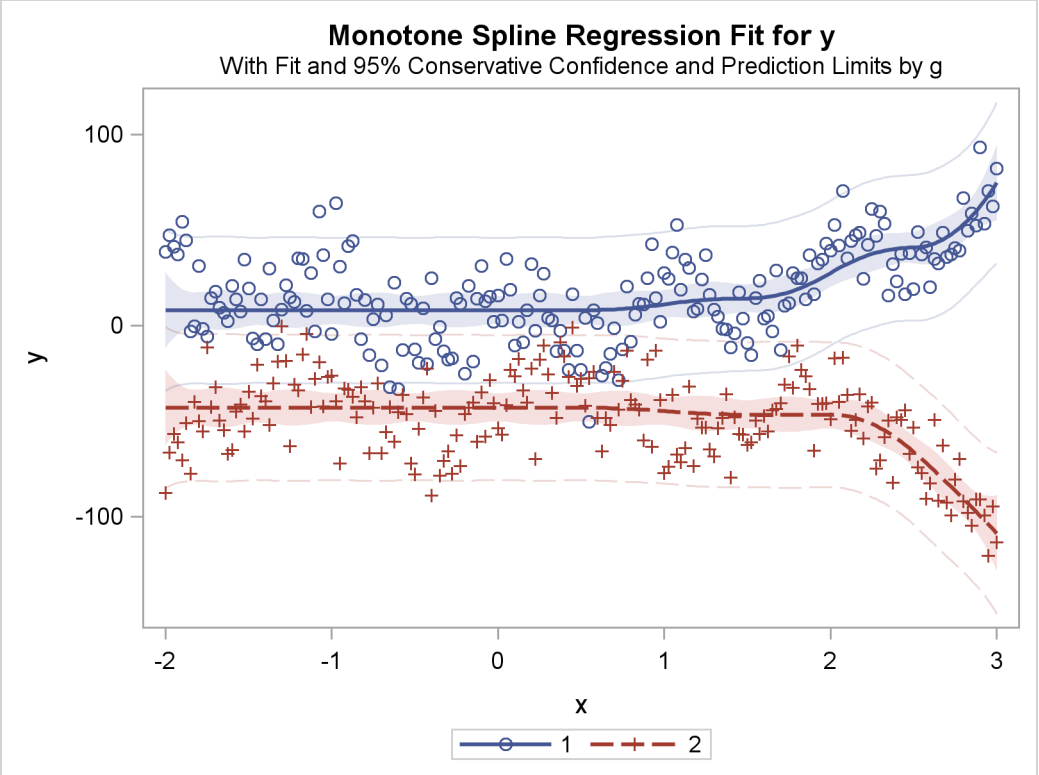
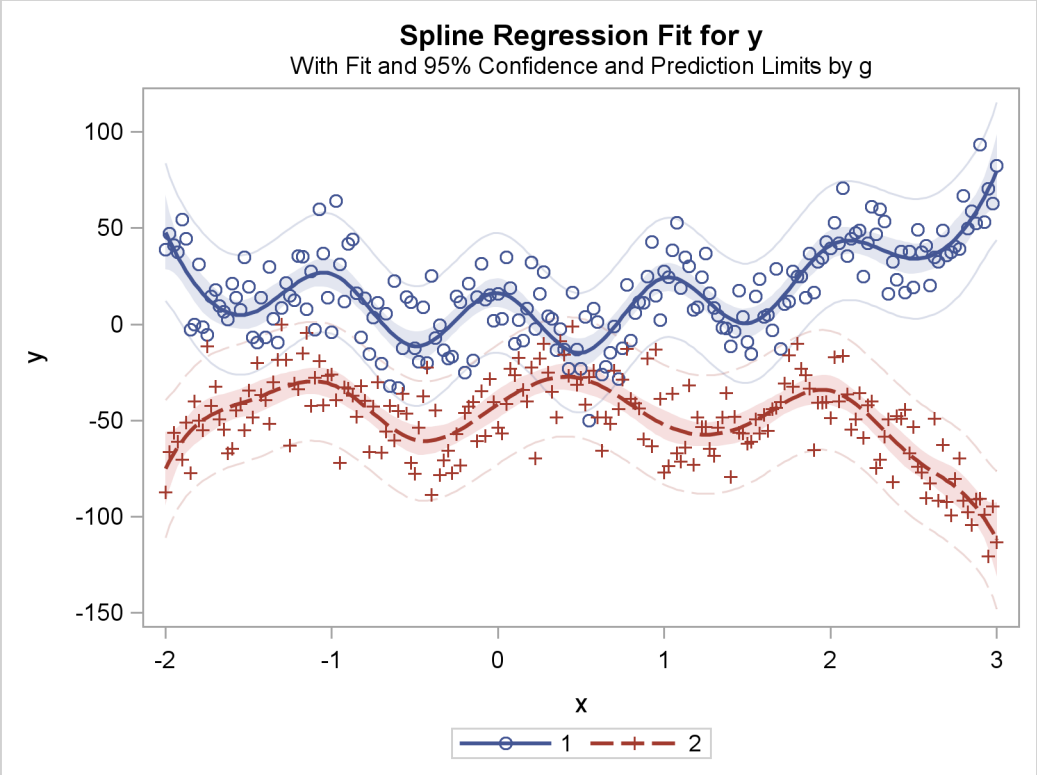


Figure 90.40 Separate Curves, Separate Intercepts



Penalized B-Splines

You can use penalized B-splines (Eilers and Marx 1996) to fit a smooth curve through a scatter plot with an automatic selection of the smoothing parameter. See Example 90.3 for an example. With penalized B-splines, you can find a transformation that minimizes any of the following criteria: CV, GCV, AIC, AICC, or SBC. These criteria are all functions of λ. For many problems, all of these criteria produce nearly identical results. However, for some problems, the choice of criterion can have a large effect. When the default results are not satisfactory, try the other criteria. Information criteria such as AIC and AICC are defined in different ways in the statistical literature, and these differences can be seen in different SAS procedures. Typically, the definitions differ only by a positive (additive or multiplicative) constant, so they are equivalent, and each of the definitions of the same criterion produces the same selection of λ. The definitions that PROC TRANSREG uses match the definitions that PROC REG uses. The penalized B-spline matrices, statistics, and criteria are defined as follows:

n	number of observations
$\mathbf{y}$	dependent variable
$\mathbf{W}$	diagonal matrix of observation weights
w_i	weight for the ith observation
$\mathbf{B}$	B-spline basis for the independent variable
λ	nonnegative smoothing parameter
$\mathbf{D}$	difference matrix, penalizes lack of smoothness

$\mathbf{H} = \mathbf{B}(\mathbf{B}'\mathbf{W}\mathbf{B} + \lambda\mathbf{D}'\mathbf{D})^{-1}\mathbf{B}'$	hat matrix
h_{ii}	ith diagonal element of $\mathbf{H}$
$\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$	penalized B-spline transformation of $\mathbf{y}$
$\text{SSE} = \sum_{i=1}^{n} w_i(y_i - \hat{y}_i)^2$	error sum of squares
$t = \sum_{i=1}^{n} w_i h_{ii}$	weighted trace of $\mathbf{H}$
$\sum_{i=1}^{n} w_i \left(\frac{y_i - \hat{y}_i}{1 - h_{ii}}\right)^2$	CV - cross validation criterion
$\sum_{i=1}^{n} w_i \left(\frac{y_i - \hat{y}_i}{n - t}\right)^2$	GCV - generalized cross validation criterion
$n \log(\text{SSE}/n) + 2t$	AIC - Akaike's information criterion
$1 + \log(\text{SSE}/n) + \frac{2(t+1)}{n-t-2}$	AICC - corrected AIC (default)
$n \log(\text{SSE}/n) + t \log(n)$	SBC - Schwarz's Bayesian criterion

For more information about constructing the B-spline basis, see Example 90.65 and the section "Using Splines and Knots" on page 7192. The nonzero elements of **D**, order 1 are (1 –1), order 2 are (1 –2 1), order 3 (the default) are (1 –3 3 –1), order 4 are (1 –4 6 –4 1), and so on. The nonzero elements for each order are made from the nonzero elements from the preceding order by subtraction: $\mathbf{d}'_{i+1} = (\mathbf{d}'_i \quad 0) - (0 \quad \mathbf{d}'_i)$. Within an order, the first nonzero element of row i is in column i—that is, each row of **D** is made from the preceding row by shifting the nonzero elements to the right one position. For example, with $k = 4$ knots, order $o = 3$, and degree $d = 3$, **D** is the $((d + 1 + k - o) \times (d + 1 + k))$ matrix:

$$\begin{bmatrix} 1 & -3 & 3 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -3 & 3 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -3 & 3 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -3 & 3 & -1 & 0 \\ 0 & 0 & 0 & 0 & 1 & -3 & 3 & -1 \end{bmatrix} \quad \text{where} \quad \begin{array}{rcccc} & 1 & -1 & 0 & \\ - & 0 & 1 & -1 & \\ \hline & 1 & -2 & 1 & 0 \\ - & 0 & 1 & -2 & 1 \\ \hline & 1 & -3 & 3 & -1 \end{array}$$

The weighted trace of the hat matrix, $t = \sum_{i=1}^{n} w_i h_{ii}$, provides an estimate of the number of parameters needed to find the transformation and is used in *df* calculations. Note, however, that in some cases, particularly with error-free or nearly error-free data, this value can be *much* larger than you might expect. You might be able to directly create a function by using SPLINE or BSPLINE with many fewer parameters that fits essentially just as well as the penalized B-spline function.

By default with PBSPLINE, a cubic spline is fit with 100 evenly spaced knots, three evenly spaced exterior knots, and a difference matrix of order three. Options are specified as follows:

PBSPLINE(x / DEGREE=3 NKNOTS=100 EVENLY=3 PARAMETER=3). By default, PROC TRANSREG searches for an optimal lambda in the range 0 to 1E6 by using parabolic interpolation and Brent's (Brent 1973; Press et al. 1989) method. Alternatively, you can specify a lambda range or a list of lambdas by using the LAMBDA= option. Be aware, however, LAMBDA=0 and values near zero might cause numerical problems including floating point errors. Also be aware that larger lambdas might cause numerical problems—for example, the error sum of squares for the model, $\Sigma(y - \hat{y})^2$, might be greater than the total sum of squares, $\Sigma(y - \bar{y})^2$—implying that the model with the transformation fits less well than simply predicting by using the mean. When this happens, you will see this message: ERROR: Degenerate transformation with PBSPLINE.

You can fit a single curve through a scatter plot (y × x) as follows:

```
model identity(y) = pbspline(x);
```

Alternatively, you can fit multiple curves through a scatter plot, one for each level of Group, as follows:

```
model identity(y) = class(group / zero=none) * pbspline(x);
```

There are several options for how the smoothing parameter, λ, is chosen. Usually, you do not specify the smoothing parameter, λ, and you let PROC TRANSREG choose λ for you by minimizing one of the information or cross validation criteria. By default, PROC TRANSREG first considers ranges defined by $\lambda = 0$ and $\lambda = 1, 10, 100, 1000, 10,000, 100,000, 1,000,000$. If it finds a range that includes the minimum, it stops and does not consider larger λ values. Then it performs further searches in that range. For example, if the initial evaluations at $\lambda = 1$ and $\lambda = 10$ show that there is at least a local minimum in the range 0 to 10, then larger values are not considered. Note that the zero smoothing case, $\lambda = 0$, provides a boundary on the range even though the criterion is not evaluated at $\lambda = 0$. The criterion is not evaluated at $\lambda = 0$ unless LAMBDA=0 is the only value specified. Also note that the default approach is not the same as specifying the options LAMBDA=0 1E6 RANGE. When a range of values is specified, along with the RANGE *t-option*, PROC TRANSREG does not try to find smaller ranges based on powers of 10.

PROC TRANSREG avoids evaluating the criterion for LAMBDA= values at or near zero unless you force it to consider them. This is because zero smoothing is rarely interesting and the results are numerically unstable. Values of λ at or near zero often result in predicted values that are far outside the range of the data, particularly with interpolation and x values that do not appear in the data set. Also, zero smoothing is prone to numerical problems including floating point errors. This is particularly true when there is a small number of observations, a large number of knots, a high degree, or a perfect or near perfect fit. If you force PROC TRANSREG to evaluate the criterion at or near $\lambda = 0$, you can easily get bad results.

Note that when some observations appear more than once, such as when you have the kind of data where you can use a FREQ statement, then you should consider directly specifying lambda based on a preliminary analysis, ignoring the frequencies. Alternatively, specify a range of λ values, such as LAMBDA=0.1 1E6 RANGE, that steers λ away from values near zero. With the default lambda list, a cross validation criterion does not perform well in choosing a smoothing parameter with replicated data. Leaving one observation out of the computations changes the frequency for that observation from one positive integer to the next smaller positive integer, so in some sense, the point corresponding to that observation is never really left out of any computations. The resulting fit will be undersmoothed unless you specify a larger λ.

Smoothing Splines

You can use PROC TRANSREG to plot and output to a SAS data set the same smoothing spline function that the GPLOT procedure creates. You request a smoothing spline transformation by specifying SMOOTH in the MODEL statement. The smoothing parameter can be specified with either the SM= or the PARAMETER= *o-option*. The results are saved in the independent variable transformation (for example, Tx, when the independent variable is x) and the predicted values variable (for example, Py, when the dependent variable is y). The smooth regression function is displayed through PROC TRANSREG and ODS Graphics in Figure 90.41.

While you would normally display the results by using only PROC TRANSREG and ODS Graphics, you can also use PROC GPLOT to verify that the two procedures produce the same results. PROC GPLOT produces Figure 90.42. The PROC GPLOT plot request **y * x = 1** displays the data as stars. The specification **y * x = 2** with I=SM50 requests the smooth curve through the scatter plot. It is overlaid with **Py * x = 3**, which displays with large dots the smooth function created by PROC TRANSREG. The following statements produce Figure 90.41 and Figure 90.42:

```
title h=1.5 'Smoothing Splines';

ods graphics on;

data x;
   do x = 1 to 100 by 2;
      do rep = 1 to 3;
         y = log(x) + sin(x / 10) + normal(7);
         output;
      end;
   end;
run;

proc transreg;
   model identity(y) = smooth(x / sm=50);
   output p;
run;

proc gplot;
   axis1 minor=none label=(angle=90 rotate=0);
   axis2 minor=none;
   symbol1 color=blue v=circle i=none;  /* data              */
   symbol2 color=blue v=none   i=sm50;  /* gplot's smooth    */
   symbol3 color=red  v=dot    i=none;  /* transreg's smooth */
   plot y*x=1 y*x=2 py*x=3 / overlay haxis=axis2 vaxis=axis1 frame;
run; quit;
```

Note in Figure 90.42 that the smoothed values from PROC TRANSREG, shown by the large dots, exactly fall on the curve produced by PROC GPLOT.

Figure 90.41 Smoothing Spline Displayed with ODS Graphics

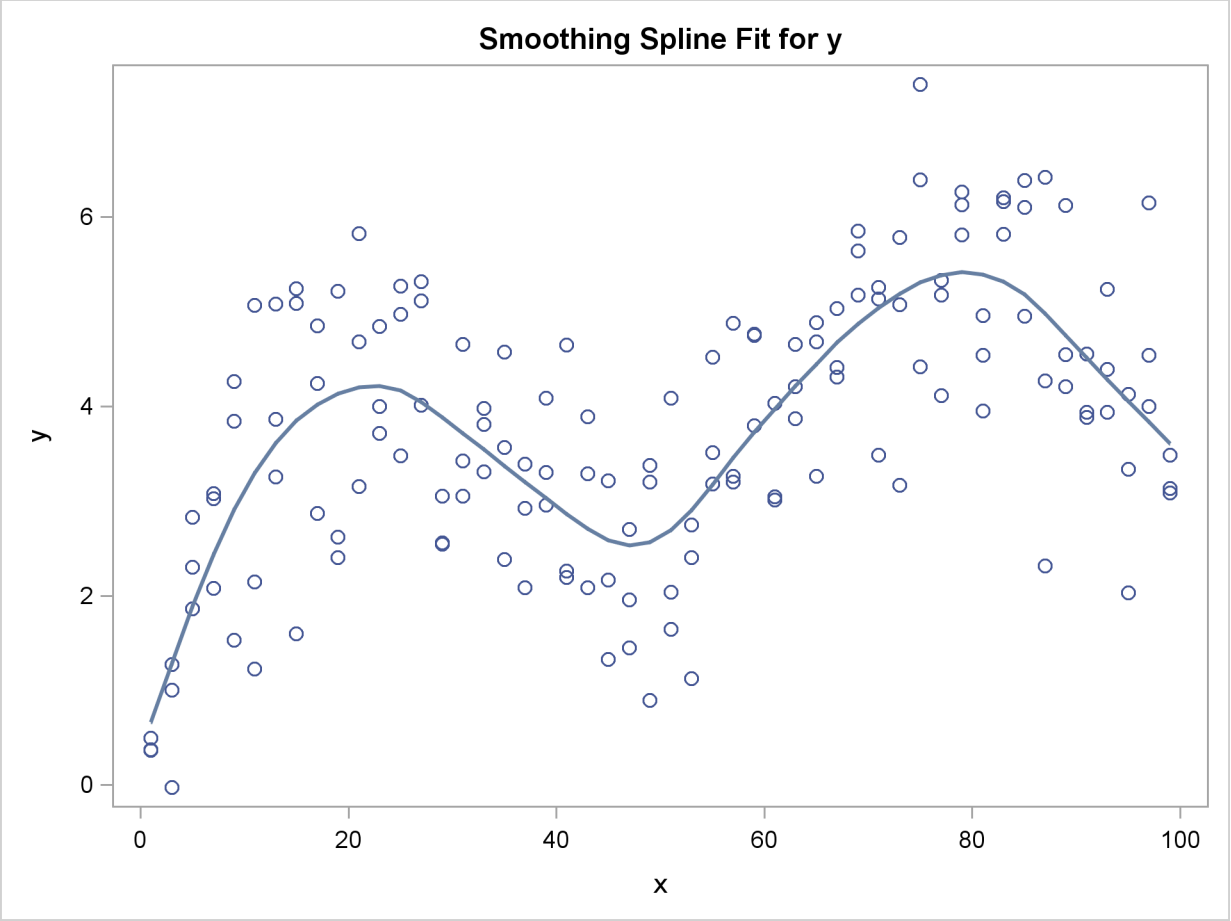
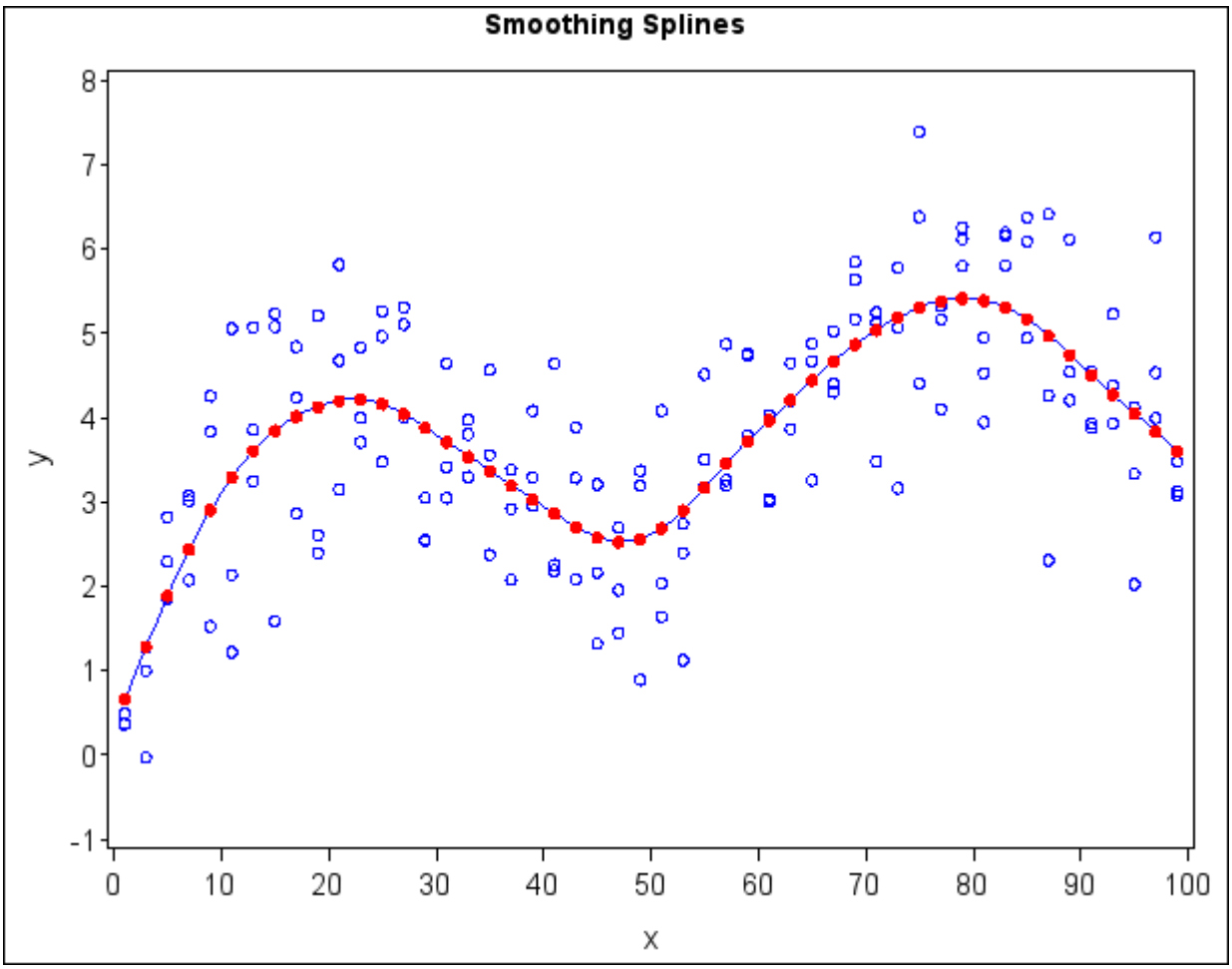


Figure 90.42 Smoothing Spline Created by PROC TRANSREG and PROC GPLOT, Overlaid



You can plot multiple nonlinear functions, one for each of several groups as defined by the levels of a CLASS variable. When you cross a SMOOTH variable with a CLASS variable, specify ZERO=NONE with the CLASS expansion. The following statements create artificial data and produce Figure 90.43:

```
title2 'Two Groups';

data x;
   do x = 1 to 100;
      Group = 1;
      do rep = 1 to 3;
         y = log(x) + sin(x / 10) + normal(7);
         output;
      end;
      group = 2;
      do rep = 1 to 3;
         y = -log(x) + cos(x / 10) + normal(7);
         output;
      end;
   end;
run;
```

```
proc transreg ss2 data=x;
   model identity(y) = class(group / zero=none) *
                       smooth(x / sm=50);
   output p;
run;

ods graphics off;
```

The ANOVA table in Figure 90.43 shows the overall model fit. The degrees of freedom are based on the trace of the transformation hat matrix, and are typically not integers. The "Smooth Transformation" table reports the degrees of freedom for each term, which includes an intercept for each group; the regression coefficients, which are always 1 with smoothing splines; the 0 to 100 smoothing parameter (like the one PROC GPLOT uses); the actual computed smoothing parameter; and the name and label for each term.

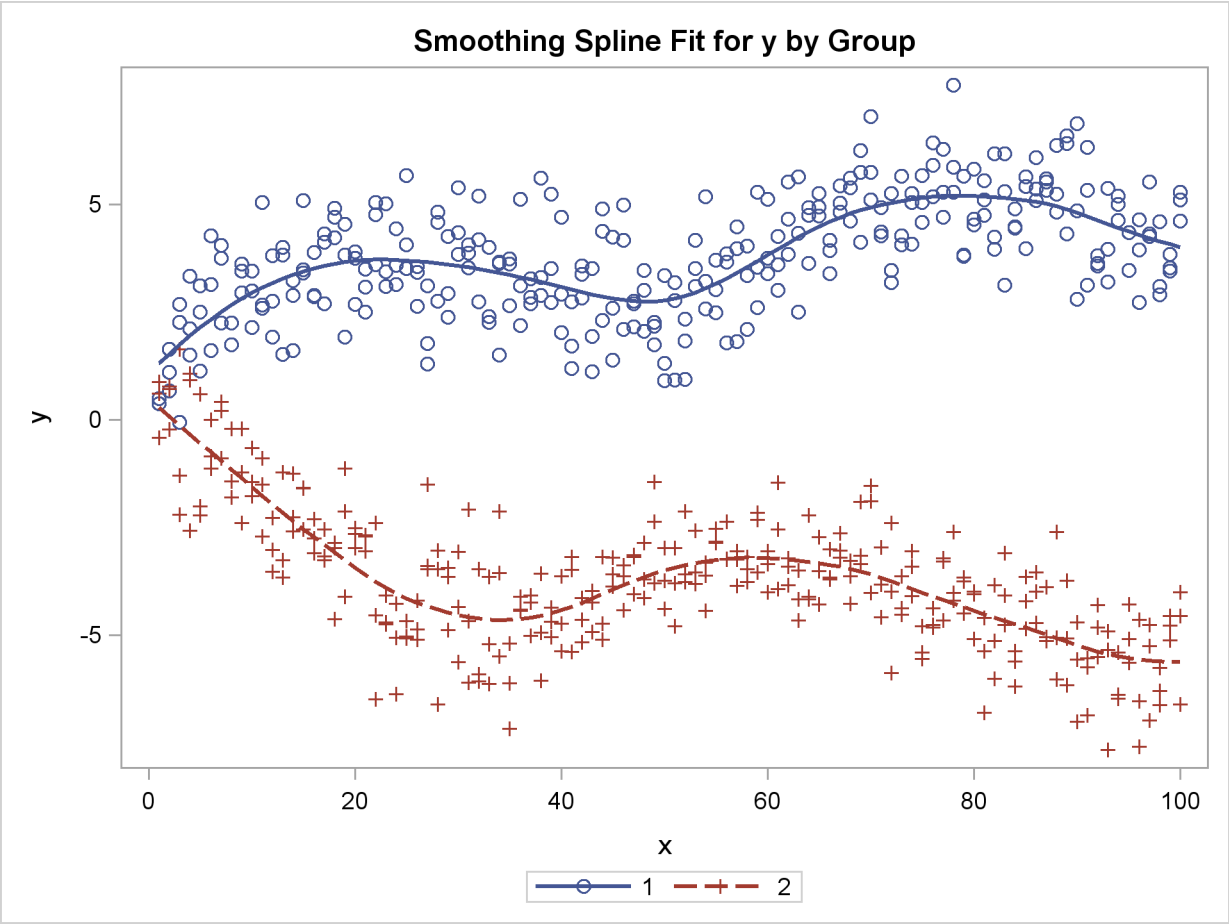
Figure 90.43 Smoothing Spline Example 2

```
                          Smoothing Splines
                              Two Groups

                        The TRANSREG Procedure

                     Dependent Variable Identity(y)


                        Class Level Information

                      Class    Levels    Values

                      Group         2    1 2

                 Number of Observations Read           600
                 Number of Observations Used           600
                 Implicit Intercept Model

          The TRANSREG Procedure Hypothesis Tests for Identity(y)


               Univariate ANOVA Table, Smooth Transformation

                                  Sum of        Mean
    Source               DF      Squares      Square    F Value    Pr > F

    Model            16.794    9195.493    547.5365     562.03    <.0001
    Error            582.21     567.195      0.9742
    Corrected Total     599    9762.688

              Root MSE           0.98702    R-Square    0.9419
              Dependent Mean     0.03651    Adj R-Sq    0.9402
              Coeff Var       2703.13908
```

Figure 90.43 *continued*

Smooth Transformation					
Variable	DF	Coefficient	SM	Parameter	Label
Smooth(Group1x)	8.8971	1.000	50	2405.265	Group 1 * x
Smooth(Group2x)	8.8971	1.000	50	2405.265	Group 2 * x

Figure 90.43 *continued*



The SMOOTH transformation is valid only with independent variables. Typically, it is used only, as in the two preceding examples, in models with a single dependent variable, a single independent variable, and optionally, a single classification variable that is crossed with the independent variable. The various standardization options such as TSTANDARD=, CENTER, Z, and REFLECT are by default not permitted when the SMOOTH transformation is part of the model.

The SMOOTH transformation can also be used in other ways, but only when you specify the NSR *a-option*. The requirement that you specify the NSR *a-option* is new with this release. (See the section "Smoothing Splines Changes and Enhancements" on page 7227.) When you specify the NSR *a-option*, and there are multiple independent variables designated as SMOOTH, PROC TRANSREG

tries to smooth the *i*th independent variable by using the *i*th dependent variable as a target. When there are more independent variables than dependent variables, the last dependent variable is reused as often as is necessary. For example, consider the following statements:

```
proc transreg nsr;
   model identity(y1-y3) = smooth(x1-x5);
   run;
```

Smoothing is based on the pairs (y1, x1), (y2, x2), (y3, x3), (y3, x4), and (y3, x5).

The SMOOTH transformation is a noniterative transformation. The smoothing of each variable occurs before the iterations begin. In contrast, SSPLINE provides an iterative smoothing spline transformation. It does not generally minimize squared error; hence, divergence is possible with SSPLINE.

Smoothing Splines Changes and Enhancements

The SMOOTH or smoothing spline transformation is the same as it has always been. However, how the results of the transformation are processed in PROC TRANSREG has changed with this release. In particular, some aspects of the syntax along the coefficients and predicted values have changed. The new behavior was required to make the smoothing splines work properly with ODS Graphics and to make SMOOTH work consistently with the new PBSPLINE (penalized B-spline; see the section "Penalized B-Splines" on page 7219) capabilities. However, you can use the new NSR *a-option*, if you want the old functionality. Here are two typical uses of the SMOOTH transformation:

```
proc transreg;
   model identity(y) = smooth(x / sm=50);
   output p;
run;

proc transreg;
   model identity(y) = class(group / zero=none) * smooth(x / sm=50);
   output p;
run;
```

For the first model, the variable x is smoothly transformed by using a smoothing parameter of SM=50, and the results are stored in the transformed variable Tx. The second model has two groups of observations corresponding to group=1 and Group=2. Separate curves are fit through each group. The results for the first group are stored in the transformed variable TGroup1x, and the results for the second group are stored in the transformed variable TGroup2x. The predicted values are stored in Py. In the first case, Py = Tx, and in the second case, Py = TGroup1x + TGroup2x. These represent the two standard usages of the SMOOTH transformation, and you can use ODS Graphics to display fit plots with a single or multiple smooth functions. For the first model, which is the most typical usage, the syntax has not changed, nor has the transformed variable. For the second model, the syntax has slightly changed, but the transformed variables have not. The details of the syntax changes are discussed later in this section. The primary change involves what happens after the SMOOTH transformation is found. Now, by default, ordinary least squares (OLS) is no longer used to find the coefficients when there are smooth transformations, and in the iteration history table the OLS R square is no longer produced.

Here is some background for the change. The first three of the four models shown next have much in common:

```
model identity(y) = smooth(x / sm=50);
model identity(y) = rank(x);
model identity(y) = log(x);
model identity(y) = spline(x);
```

Previously, the SMOOTH, RANK, and LOG transformations all requested that PROC TRANSREG preprocess the data, nonlinearly transforming x before using OLS to fit a model to the preprocessed results. All of these first three transformations of x are nonoptimal in the sense that none of them is based in any way on the OLS regression model that follows the preprocessing of the data. In contrast, the fourth model requests a spline transformation. In this model, both the nonlinear transformation and the final regression model seek to minimize the same OLS criterion. Some PROC TRANSREG transformations, such as SPLINE, MSPLINE, OPSCORE, MONOTONE, and so on, seek to minimize squared error, whereas others, such as SMOOTH, LOG, EXP, and RANK, do not. For the latter, the data are simply preprocessed before analysis. There is a philosophical difference, however, between SMOOTH and the nonoptimal transformations. The SMOOTH and PBSPLINE transformations use the dependent variable and a model (but not OLS) to compute the transformation, whereas LOG, EXP, RANK, and the other nonoptimal transformations do not. A log transformation, for example, would be the same, regardless of context, whereas the SMOOTH and PBSPLINE transformations depend on the model.

The principal change to SMOOTH, with this release of PROC TRANSREG, involves making PROC TRANSREG aware of the underlying smoothing spline model. This makes SMOOTH and PBSPLINE perform similarly, and less like LOG, EXP, RANK, and the other nonoptimal transformations. Previously, if you specified SMOOTH and then examined the regression coefficients, you would probably get an intercept very close to but not exactly 0, and the remaining coefficients would be very close to but not exactly 1. This is because PROC TRANSREG was using OLS to find the coefficients. This has changed. Now, PROC TRANSREG recognizes that the SMOOTH transformation has an implicit intercept (see the section "Implicit and Explicit Intercepts" on page 7254); hence there is no separate intercept. Furthermore, now the other parameters are exactly 1, which are the correct parameters for the non-OLS smoothing spline model. Hence, the predicted values are now the sum of the transformed variables. When there is no CLASS variable, the predicted values exactly match the transformed variable. The SMOOTH transformation is no longer a form of preprocessing; it now changes the nature of the model from OLS to a true smoothing-spline model. If you still want the old behavior, preprocessing and then OLS, you can get the old default functionality by specifying the NSR *a-option*.

The new, default functionality assumes that you either want to fit a smooth function through the data or fit separate functions, one for each level of a CLASS variable. It also recognizes the smoothing-spline model as a model with an implicit intercept. For these reasons, the syntax for models with a CLASS variable has slightly changed, as is shown next:

```
proc transreg nsr; /* old */
   model identity(y) = class(group / zero=none) |
                       smooth(x / after sm=50);
   output p;
run;

proc transreg; /* new */
   model identity(y) = class(group / zero=none) *
                       smooth(x / sm=50);
   output p;
run;
```

Previously, the AFTER *t-option* was required when you wanted to fit separate and independent functions within each group. This *t-option* specifies that PROC TRANSREG should find the smoothing spline transformations *after* it crosses the independent variable with the CLASS variable. Previously, by default, PROC TRANSREG found an overall smooth transformation and then crossed it with the CLASS variable, which is probably not what you want. You can still specify the AFTER *t-option*, but now it is assumed with CLASS * SMOOTH. If you specify AFTER without the NSR *a-option*, PROC TRANSREG suppresses the note that AFTER is assumed. It does not affect the model. If you do not want AFTER to be in effect by default, you must specify the NSR *a-option*. Also previously, you typically needed to specify the vertical bar instead of the asterisk to cross the CLASS and SMOOTH variables. The difference is that the bar adds both crossed variables and separate group intercepts to the model, whereas the asterisk adds only the crossed variables to the model. Since the SMOOTH transformation is now recognized as providing an implicit intercept, you should use the asterisk and not the vertical bar.

The default behavior of the SMOOTH transformation needed to change for several reasons. SMOOTH was originally provided as nothing more than a way to get PROC GPLOT's smoothing splines into an output data set in the transformed variables. However, with new enhancements to PROC TRANSREG such as ODS Graphics and PBSPLINE, the old method for SMOOTH did not fit well. The old method produced predicted values that were not the correct values to plot in order to show the smoothing spline fit. Now, with this change, ODS Graphics can always plot the predicted values. Also, PBSPLINE is new with this release; it and SMOOTH are similar in spirit, and for both, OLS results are not truly appropriate. Previously, PROC TRANSREG fit linear models, linear models with nonlinearly preprocessed variables, and linear models with optimal nonlinear transformations that minimized squared error. Now it also has the ability to fit non-OLS models for scatter plot smoothing.

One aspect of the SMOOTH transformation has unconditionally changed with this release. Previously, PROC TRANSREG did not evaluate the effective degrees of freedom by examining the trace of the transformation hat matrix. It simply used the number of categories in the *df* calculations, which for continuous variables is the number of observations. This made it impossible to get a sensible ANOVA test for the overall fit. With this release, the degrees of freedom are always based on the trace. This *df* change also affects the SSPLINE transformation, which finds a smooth transformation by using the same algorithm as SMOOTH. The difference is that the SMOOTH transformation occurs once, as an analysis preprocessing step, whereas SSPLINE transformations occur iteratively and in the body of the alternating least-squares algorithm.

Iteration History Changes and Enhancements

With this release, PROC TRANSREG no longer always prints an iteration history table by default, and in some cases, the table it prints is not the same as it was previously. This change is due to the increasing use of PROC TRANSREG with transformations that are not based on alternating least squares. Here is some background for the change. PROC TRANSREG's processing can be divided into three steps. In the first step, the data are read and certain transformations, such as SMOOTH, PBSPLINE, BOXCOX, RANK, LOG and the other nonoptimal transformations, are performed. These transformations are not based on OLS. In the second step, the alternating-least-squares iterations are performed according to METHOD=UNIVARIATE, MORALS, REDUNDANCY, or CANALS. It is in the second step that the alternating-least-squares transformations (SPLINE, MSPLINE, MONOTONE, OPSCORE, LINEAR, and UNTIE) are iteratively found. In the third step, the results are displayed. In some cases, the results are appropriately based on using the method of OLS applied to the optimally transformed variables. In other cases, such as with smoothing splines and penalized B-splines, OLS-based results are not appropriate. Furthermore, for many of these types of models, nothing changes in the iterations, so the computations needed to realize that nothing changes are not needed, nor is the iteration history table.

With this release, the iteration history is not printed for models where it is known that nothing will change in the iterations. Suppose the NOMISS option is specified or there are no missing data. If METHOD=UNIVARIATE, if there are no iterative transformations (SPLINE, MSPLINE, MONOTONE, OPSCORE, LINEAR, and UNTIE), and if the MAXITER= option is not specified, then by default, an iteration history table is not produced. If you want to see an iteration history, there are many things you can do, such as specifying MAXITER=, changing the method to MORALS, or changing IDENTITY to LINEAR.

With models with smoothing splines or penalized B-splines, the iteration history will not contain an R square. This is because the iterations are based on the method of alternating least squares, but the smoothing splines and penalized B-splines are not based on a least-squares model. Hence, an ordinary R square in the iterations, based on a computed intercept, which is typically not exactly zero, and a computed slope, which is typically not exactly one, will not be exactly the same as the correct R square, which is based on an intercept and slope of zero and one. The final reported results include the correct R square in the fit statistics table after the ANOVA table. If you want to see only the correct R square from the results, without the iteration history, you can specify the new RSQUARE option.

ANOVA Codings

This section illustrates several different codings of classification variables and hence several different ways of fitting two-way ANOVA models to some data. Each example fits an ANOVA model, displays the ANOVA table and parameter estimates, and displays the coded design matrix. Note throughout that the ANOVA tables and R squares are identical for all of the models, showing that the codings are equivalent. For each model, the parameter estimates are stated as a function of the cell means. The formulas are appropriate for a design such as this one, which is balanced and

orthogonal (every level and every pair of levels occurs equally often). They will not work with unequal frequencies. Since this data set has $3 \times 2 = 6$ cells, the full-rank codings all have six parameters. The following statements create the input data set, and display it in Figure 90.44:

```
title 'Two-Way ANOVA Models';

data x;
   input a b @@;
   do i = 1 to 2; input y @@; output; end;
   drop i;
   datalines;
1 1   16 14         1 2   15 13
2 1    1  9         2 2   12 20
3 1   14  8         3 2   18 20
;

proc print label;
run;
```

Figure 90.44 Input Data Set

Two-Way ANOVA Models

Obs	a	b	y
1	1	1	16
2	1	1	14
3	1	2	15
4	1	2	13
5	2	1	1
6	2	1	9
7	2	2	12
8	2	2	20
9	3	1	14
10	3	1	8
11	3	2	18
12	3	2	20

The following statements fit a cell-means model and produce Figure 90.45 and Figure 90.46:

```
proc transreg data=x ss2 short;
   title2 'Cell-Means Model';
   model identity(y) = class(a * b / zero=none);
   output replace;
run;

proc print label;
run;
```

Figure 90.45 Cell-Means Model

Two-Way ANOVA Models
Cell-Means Model

The TRANSREG Procedure

Dependent Variable Identity(y)

Class Level Information

Class	Levels	Values
a	3	1 2 3
b	2	1 2

Number of Observations Read	12
Number of Observations Used	12
Implicit Intercept Model	

The TRANSREG Procedure Hypothesis Tests for Identity(y)

Univariate ANOVA Table Based on the Usual Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	234.6667	46.93333	3.20	0.0946
Error	6	88.0000	14.66667		
Corrected Total	11	322.6667			

Root MSE	3.82971	R-Square	0.7273
Dependent Mean	13.33333	Adj R-Sq	0.5000
Coeff Var	28.72281		

Univariate Regression Table Based on the Usual Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Pr > F	Label
Class.a1b1	1	15.0000000	450.000	450.000	30.68	0.0015	a 1 * b 1
Class.a1b2	1	14.0000000	392.000	392.000	26.73	0.0021	a 1 * b 2
Class.a2b1	1	5.0000000	50.000	50.000	3.41	0.1144	a 2 * b 1
Class.a2b2	1	16.0000000	512.000	512.000	34.91	0.0010	a 2 * b 2
Class.a3b1	1	11.0000000	242.000	242.000	16.50	0.0066	a 3 * b 1
Class.a3b2	1	19.0000000	722.000	722.000	49.23	0.0004	a 3 * b 2

The parameter estimates are

$$\begin{aligned}
\hat{\mu}_{11} &= \overline{y}_{11} = 15 \\
\hat{\mu}_{12} &= \overline{y}_{12} = 14 \\
\hat{\mu}_{21} &= \overline{y}_{21} = 5 \\
\hat{\mu}_{22} &= \overline{y}_{22} = 16 \\
\hat{\mu}_{31} &= \overline{y}_{31} = 11 \\
\hat{\mu}_{32} &= \overline{y}_{32} = 19
\end{aligned}$$

Figure 90.46 Cell-Means Model, Design Matrix

Two-Way ANOVA Models
Cell-Means Model

Obs	_TYPE_	_NAME_	y	Intercept	a 1 * b 1	a 1 * b 2	a 2 * b 1	a 2 * b 2	a 3 * b 1	a 3 * b 2	a	b
1	SCORE	ROW1	16	.	1	0	0	0	0	0	1	1
2	SCORE	ROW2	14	.	1	0	0	0	0	0	1	1
3	SCORE	ROW3	15	.	0	1	0	0	0	0	1	2
4	SCORE	ROW4	13	.	0	1	0	0	0	0	1	2
5	SCORE	ROW5	1	.	0	0	1	0	0	0	2	1
6	SCORE	ROW6	9	.	0	0	1	0	0	0	2	1
7	SCORE	ROW7	12	.	0	0	0	1	0	0	2	2
8	SCORE	ROW8	20	.	0	0	0	1	0	0	2	2
9	SCORE	ROW9	14	.	0	0	0	0	1	0	3	1
10	SCORE	ROW10	8	.	0	0	0	0	1	0	3	1
11	SCORE	ROW11	18	.	0	0	0	0	0	1	3	2
12	SCORE	ROW12	20	.	0	0	0	0	0	1	3	2

The next model is a reference cell model, and the default reference cell is the last cell, which in this case is the (3,2) cell. The following statements fit a reference cell model and produce Figure 90.47 and Figure 90.48:

```
proc transreg data=x ss2 short;
   title2 'Reference Cell Model, (3,2) Reference Cell';
   model identity(y) = class(a | b);
   output replace;
run;

proc print label;
run;
```

Figure 90.47 Reference Cell Model, (3,2) Reference Cell

Two-Way ANOVA Models
Reference Cell Model, (3,2) Reference Cell

The TRANSREG Procedure

Dependent Variable Identity(y)

Class Level Information

Class	Levels	Values
a	3	1 2 3
b	2	1 2

Number of Observations Read	12
Number of Observations Used	12

The TRANSREG Procedure Hypothesis Tests for Identity(y)

Univariate ANOVA Table Based on the Usual Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	234.6667	46.93333	3.20	0.0946
Error	6	88.0000	14.66667		
Corrected Total	11	322.6667			

Root MSE	3.82971	R-Square	0.7273
Dependent Mean	13.33333	Adj R-Sq	0.5000
Coeff Var	28.72281		

Univariate Regression Table Based on the Usual Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Pr > F	Label
Intercept	1	19.0000000	722.000	722.000	49.23	0.0004	Intercept
Class.a1	1	-5.0000000	25.000	25.000	1.70	0.2395	a 1
Class.a2	1	-3.0000000	9.000	9.000	0.61	0.4632	a 2
Class.b1	1	-8.0000000	64.000	64.000	4.36	0.0817	b 1
Class.a1b1	1	9.0000000	40.500	40.500	2.76	0.1476	a 1 * b 1
Class.a2b1	1	-3.0000000	4.500	4.500	0.31	0.5997	a 2 * b 1

The parameter estimates are

$$
\begin{aligned}
\hat{\mu}_{32} &= \overline{y}_{32} = 19 \\
\hat{\alpha}_1 &= \overline{y}_{12} - \overline{y}_{32} = 14 - 19 = -5 \\
\hat{\alpha}_2 &= \overline{y}_{22} - \overline{y}_{32} = 16 - 19 = -3 \\
\hat{\beta}_1 &= \overline{y}_{31} - \overline{y}_{32} = 11 - 19 = -8 \\
\hat{\gamma}_{11} &= \overline{y}_{11} - (\hat{\mu}_{32} + \hat{\alpha}_1 + \hat{\beta}_1) = 15 - (19 + -5 + -8) = 9 \\
\hat{\gamma}_{21} &= \overline{y}_{21} - (\hat{\mu}_{32} + \hat{\alpha}_2 + \hat{\beta}_1) = 5 - (19 + -3 + -8) = -3
\end{aligned}
$$

Figure 90.48 Reference Cell Model, (3,2) Reference Cell, Design Matrix

Two-Way ANOVA Models
Reference Cell Model, (3,2) Reference Cell

Obs	_TYPE_	_NAME_	y	Intercept	a 1	a 2	b 1	a 1 * b 1	a 2 * b 1	a	b
1	SCORE	ROW1	16	1	1	0	1	1	0	1	1
2	SCORE	ROW2	14	1	1	0	1	1	0	1	1
3	SCORE	ROW3	15	1	1	0	0	0	0	1	2
4	SCORE	ROW4	13	1	1	0	0	0	0	1	2
5	SCORE	ROW5	1	1	0	1	1	0	1	2	1
6	SCORE	ROW6	9	1	0	1	1	0	1	2	1
7	SCORE	ROW7	12	1	0	1	0	0	0	2	2
8	SCORE	ROW8	20	1	0	1	0	0	0	2	2
9	SCORE	ROW9	14	1	0	0	1	0	0	3	1
10	SCORE	ROW10	8	1	0	0	1	0	0	3	1
11	SCORE	ROW11	18	1	0	0	0	0	0	3	2
12	SCORE	ROW12	20	1	0	0	0	0	0	3	2

The next model is a deviations-from-means model. This coding is also called effects coding. The default reference cell is the last cell (3,2). The following statements produce Figure 90.49 and Figure 90.50:

```
proc transreg data=x ss2 short;
   title2 'Deviations from Means, (3,2) Reference Cell';
   model identity(y) = class(a | b / deviations);
   output replace;
run;

proc print label;
run;
```

Figure 90.49 Deviations-from-Means Model, (3,2) Reference Cell

Two-Way ANOVA Models
Deviations from Means, (3,2) Reference Cell

The TRANSREG Procedure

Dependent Variable Identity(y)

Class Level Information

Class	Levels	Values
a	3	1 2 3
b	2	1 2

Number of Observations Read	12
Number of Observations Used	12

The TRANSREG Procedure Hypothesis Tests for Identity(y)

Univariate ANOVA Table Based on the Usual Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	234.6667	46.93333	3.20	0.0946
Error	6	88.0000	14.66667		
Corrected Total	11	322.6667			

Root MSE	3.82971	R-Square	0.7273
Dependent Mean	13.33333	Adj R-Sq	0.5000
Coeff Var	28.72281		

Univariate Regression Table Based on the Usual Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Pr > F	Label
Intercept	1	13.3333333	2133.33	2133.33	145.45	<.0001	Intercept
Class.a1	1	1.1666667	8.17	8.17	0.56	0.4837	a 1
Class.a2	1	-2.8333333	48.17	48.17	3.28	0.1199	a 2
Class.b1	1	-3.0000000	108.00	108.00	7.36	0.0349	b 1
Class.a1b1	1	3.5000000	73.50	73.50	5.01	0.0665	a 1 * b 1
Class.a2b1	1	-2.5000000	37.50	37.50	2.56	0.1609	a 2 * b 1

The parameter estimates are

$$
\begin{aligned}
\hat{\mu} &= \overline{y} = 13.3333 \\
\hat{\alpha}_1 &= (\overline{y}_{11} + \overline{y}_{12})/2 - \overline{y} = (15 + 14)/2 - 13.3333 = 1.1667 \\
\hat{\alpha}_2 &= (\overline{y}_{21} + \overline{y}_{22})/2 - \overline{y} = (5 + 16)/2 - 13.3333 = -2.8333 \\
\hat{\beta}_1 &= (\overline{y}_{11} + \overline{y}_{21} + \overline{y}_{31})/3 - \overline{y} = (15 + 5 + 11)/3 - 13.3333 = -3 \\
\hat{\gamma}_{11} &= \overline{y}_{11} - (\overline{y} + \hat{\alpha}_1 + \hat{\beta}_1) = 15 - (13.3333 + 1.1667 + -3) = 3.5 \\
\hat{\gamma}_{21} &= \overline{y}_{21} - (\overline{y} + \hat{\alpha}_2 + \hat{\beta}_1) = 5 - (13.3333 + -2.8333 + -3) = -2.5
\end{aligned}
$$

Figure 90.50 Deviations-from-Means Model, (3,2) Reference Cell, Design Matrix

Two-Way ANOVA Models
Deviations from Means, (3,2) Reference Cell

Obs	_TYPE_	_NAME_	y	Intercept	a 1	a 2	b 1	a 1 * b 1	a 2 * b 1	a	b
1	SCORE	ROW1	16	1	1	0	1	1	0	1	1
2	SCORE	ROW2	14	1	1	0	1	1	0	1	1
3	SCORE	ROW3	15	1	1	0	-1	-1	0	1	2
4	SCORE	ROW4	13	1	1	0	-1	-1	0	1	2
5	SCORE	ROW5	1	1	0	1	1	0	1	2	1
6	SCORE	ROW6	9	1	0	1	1	0	1	2	1
7	SCORE	ROW7	12	1	0	1	-1	0	-1	2	2
8	SCORE	ROW8	20	1	0	1	-1	0	-1	2	2
9	SCORE	ROW9	14	1	-1	-1	1	-1	-1	3	1
10	SCORE	ROW10	8	1	-1	-1	1	-1	-1	3	1
11	SCORE	ROW11	18	1	-1	-1	-1	1	1	3	2
12	SCORE	ROW12	20	1	-1	-1	-1	1	1	3	2

The next model is a less-than-full-rank model. The parameter estimates are constrained to sum to zero within each effect. The following statements produce Figure 90.51 and Figure 90.52:

```
proc transreg data=x ss2 short;
   title2 'Less-Than-Full-Rank Model';
   model identity(y) = class(a | b / zero=sum);
   output replace;
run;

proc print label;
run;
```

Figure 90.51 Less-Than-Full-Rank Model

Two-Way ANOVA Models
Less-Than-Full-Rank Model

The TRANSREG Procedure

Dependent Variable Identity(y)

Class Level Information

Class	Levels	Values
a	3	1 2 3
b	2	1 2

Number of Observations Read	12
Number of Observations Used	12

The TRANSREG Procedure Hypothesis Tests for Identity(y)

Univariate ANOVA Table Based on the Usual Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	234.6667	46.93333	3.20	0.0946
Error	6	88.0000	14.66667		
Corrected Total	11	322.6667			

Root MSE	3.82971	R-Square	0.7273
Dependent Mean	13.33333	Adj R-Sq	0.5000
Coeff Var	28.72281		

Figure 90.51 *continued*

Univariate Regression Table Based on the Usual Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Pr > F	Label
Intercept	1	13.3333333	2133.33	2133.33	145.45	<.0001	Intercept
Class.a1	1	1.1666667	8.17	8.17	0.56	0.4837	a 1
Class.a2	1	-2.8333333	48.17	48.17	3.28	0.1199	a 2
Class.a3	1	1.6666667	16.67	16.67	1.14	0.3274	a 3
Class.b1	1	-3.0000000	108.00	108.00	7.36	0.0349	b 1
Class.b2	1	3.0000000	108.00	108.00	7.36	0.0349	b 2
Class.a1b1	1	3.5000000	73.50	73.50	5.01	0.0665	a 1 * b 1
Class.a1b2	1	-3.5000000	73.50	73.50	5.01	0.0665	a 1 * b 2
Class.a2b1	1	-2.5000000	37.50	37.50	2.56	0.1609	a 2 * b 1
Class.a2b2	1	2.5000000	37.50	37.50	2.56	0.1609	a 2 * b 2
Class.a3b1	1	-1.0000000	6.00	6.00	0.41	0.5461	a 3 * b 1
Class.a3b2	1	1.0000000	6.00	6.00	0.41	0.5461	a 3 * b 2

The sum of the regression table DF's, minus one for the intercept, will be greater than the model df when there are ZERO=SUM constraints.

The parameter estimates are

$$
\begin{aligned}
\hat{\mu} &= \overline{y} = 13.3333 \\
\hat{\alpha}_1 &= (\overline{y}_{11} + \overline{y}_{12})/2 - \overline{y} = (15 + 14)/2 - 13.3333 = 1.1667 \\
\hat{\alpha}_2 &= (\overline{y}_{21} + \overline{y}_{22})/2 - \overline{y} = (5 + 16)/2 - 13.3333 = -2.8333 \\
\hat{\alpha}_3 &= (\overline{y}_{31} + \overline{y}_{32})/2 - \overline{y} = (11 + 19)/2 - 13.3333 = 1.6667 \\
\hat{\beta}_1 &= (\overline{y}_{11} + \overline{y}_{21} + \overline{y}_{31})/3 - \overline{y} = (15 + 5 + 11)/3 - 13.3333 = -3 \\
\hat{\beta}_2 &= (\overline{y}_{12} + \overline{y}_{22} + \overline{y}_{32})/3 - \overline{y} = (14 + 16 + 19)/3 - 13.3333 = 3 \\
\hat{\gamma}_{11} &= \overline{y}_{11} - (\overline{y} + \hat{\alpha}_1 + \hat{\beta}_1) = 15 - (13.3333 + 1.1667 + -3) = 3.5 \\
\hat{\gamma}_{12} &= \overline{y}_{12} - (\overline{y} + \hat{\alpha}_1 + \hat{\beta}_2) = 14 - (13.3333 + 1.1667 + 3) = -3.5 \\
\hat{\gamma}_{21} &= \overline{y}_{21} - (\overline{y} + \hat{\alpha}_2 + \hat{\beta}_1) = 5 - (13.3333 + -2.8333 + -3) = -2.5 \\
\hat{\gamma}_{22} &= \overline{y}_{22} - (\overline{y} + \hat{\alpha}_2 + \hat{\beta}_2) = 16 - (13.3333 + -2.8333 + 3) = 2.5 \\
\hat{\gamma}_{31} &= \overline{y}_{31} - (\overline{y} + \hat{\alpha}_3 + \hat{\beta}_1) = 11 - (13.3333 + 1.6667 + -3) = -1 \\
\hat{\gamma}_{32} &= \overline{y}_{32} - (\overline{y} + \hat{\alpha}_3 + \hat{\beta}_2) = 19 - (13.3333 + 1.6667 + 3) = 1
\end{aligned}
$$

The constraints are

$$\alpha_1 + \alpha_2 + \alpha_3 \equiv \beta_1 + \beta_2 \equiv 0$$

$$\gamma_{11} + \gamma_{12} \equiv \gamma_{21} + \gamma_{22} \equiv \gamma_{31} + \gamma_{32} \equiv \gamma_{11} + \gamma_{21} + \gamma_{31} \equiv \gamma_{12} + \gamma_{22} + \gamma_{32} \equiv 0$$

Only four of the five interaction constraints are needed. The fifth constraint is implied by the other four. (Given a 2×3 table with four marginal sum-to-zero constraints, you can freely fill in only two cells. The values in the other four cells are determined from the first two cells and the constraints.) A full-rank model has six estimable parameters. This less-than-full-rank model has one parameter for the intercept, two for the first main effect (plus one more as determined by the first constraint),

one for the second main effect (plus one more as determined by the second constraint), and two for the interactions (plus four more as determined by the next four constraints). Six of the twelve parameters are determined given the other six and the constraints. Notice that $\hat{\mu}, \hat{\alpha}_1, \hat{\alpha}_2, \hat{\beta}_1, \hat{\gamma}_{11}$, and $\hat{\gamma}_{21}$ match the corresponding estimates from the effects coding.

Figure 90.52 Less-Than-Full-Rank Model, Design Matrix

Two-Way ANOVA Models
Less-Than-Full-Rank Model

Obs	_TYPE_	_NAME_	y	Intercept	a 1	a 2	a 3	b 1
1	SCORE	ROW1	16	1	1	0	0	1
2	SCORE	ROW2	14	1	1	0	0	1
3	SCORE	ROW3	15	1	1	0	0	0
4	SCORE	ROW4	13	1	1	0	0	0
5	SCORE	ROW5	1	1	0	1	0	1
6	SCORE	ROW6	9	1	0	1	0	1
7	SCORE	ROW7	12	1	0	1	0	0
8	SCORE	ROW8	20	1	0	1	0	0
9	SCORE	ROW9	14	1	0	0	1	1
10	SCORE	ROW10	8	1	0	0	1	1
11	SCORE	ROW11	18	1	0	0	1	0
12	SCORE	ROW12	20	1	0	0	1	0

Obs	b 2	a 1 * b 1	a 1 * b 2	a 2 * b 1	a 2 * b 2	a 3 * b 1	a 3 * b 2	a	b
1	0	1	0	0	0	0	0	1	1
2	0	1	0	0	0	0	0	1	1
3	1	0	1	0	0	0	0	1	2
4	1	0	1	0	0	0	0	1	2
5	0	0	0	1	0	0	0	2	1
6	0	0	0	1	0	0	0	2	1
7	1	0	0	0	1	0	0	2	2
8	1	0	0	0	1	0	0	2	2
9	0	0	0	0	0	1	0	3	1
10	0	0	0	0	0	1	0	3	1
11	1	0	0	0	0	0	1	3	2
12	1	0	0	0	0	0	1	3	2

The next model is a reference cell model, but this time the reference cell is the first cell (1,1). The following statements produce Figure 90.53 and Figure 90.54:

```
proc transreg data=x ss2 short;
   title2 'Reference Cell Model, (1,1) Reference Cell';
   model identity(y) = class(a | b / zero=first);
   output replace;
run;

proc print label;
run;
```

Figure 90.53 Reference Cell Model, (1,1) Reference Cell

Two-Way ANOVA Models
Reference Cell Model, (1,1) Reference Cell

The TRANSREG Procedure

Dependent Variable Identity(y)

Class Level Information

Class	Levels	Values
a	3	1 2 3
b	2	1 2

Number of Observations Read	12
Number of Observations Used	12

The TRANSREG Procedure Hypothesis Tests for Identity(y)

Univariate ANOVA Table Based on the Usual Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	234.6667	46.93333	3.20	0.0946
Error	6	88.0000	14.66667		
Corrected Total	11	322.6667			

Root MSE	3.82971	R-Square	0.7273
Dependent Mean	13.33333	Adj R-Sq	0.5000
Coeff Var	28.72281		

Univariate Regression Table Based on the Usual Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Pr > F	Label
Intercept	1	15.000000	450.000	450.000	30.68	0.0015	Intercept
Class.a2	1	-10.000000	100.000	100.000	6.82	0.0401	a 2
Class.a3	1	-4.000000	16.000	16.000	1.09	0.3365	a 3
Class.b2	1	-1.000000	1.000	1.000	0.07	0.8027	b 2
Class.a2b2	1	12.000000	72.000	72.000	4.91	0.0686	a 2 * b 2
Class.a3b2	1	9.000000	40.500	40.500	2.76	0.1476	a 3 * b 2

The parameter estimates are

$$
\begin{aligned}
\hat{\mu}_{11} &= \overline{y}_{11} = 15 \\
\hat{\alpha}_2 &= \overline{y}_{21} - \overline{y}_{11} = 5 - 15 = -10 \\
\hat{\alpha}_3 &= \overline{y}_{31} - \overline{y}_{11} = 11 - 15 = -4 \\
\hat{\beta}_2 &= \overline{y}_{12} - \overline{y}_{11} = 14 - 15 = -1 \\
\hat{\gamma}_{22} &= \overline{y}_{22} - (\hat{\mu}_{11} + \hat{\alpha}_2 + \hat{\beta}_2) = 16 - (15 + -10 + -1) = 12 \\
\hat{\gamma}_{32} &= \overline{y}_{32} - (\hat{\mu}_{11} + \hat{\alpha}_3 + \hat{\beta}_2) = 19 - (15 + -4 + -1) = 9
\end{aligned}
$$

Figure 90.54 Reference Cell Model, (1,1) Reference Cell, Design Matrix

Two-Way ANOVA Models
Reference Cell Model, (1,1) Reference Cell

Obs	_TYPE_	_NAME_	y	Intercept	a 2	a 3	b 2	a 2 * b 2	a 3 * b 2	a	b
1	SCORE	ROW1	16	1	0	0	0	0	0	1	1
2	SCORE	ROW2	14	1	0	0	0	0	0	1	1
3	SCORE	ROW3	15	1	0	0	1	0	0	1	2
4	SCORE	ROW4	13	1	0	0	1	0	0	1	2
5	SCORE	ROW5	1	1	1	0	0	0	0	2	1
6	SCORE	ROW6	9	1	1	0	0	0	0	2	1
7	SCORE	ROW7	12	1	1	0	1	1	0	2	2
8	SCORE	ROW8	20	1	1	0	1	1	0	2	2
9	SCORE	ROW9	14	1	0	1	0	0	0	3	1
10	SCORE	ROW10	8	1	0	1	0	0	0	3	1
11	SCORE	ROW11	18	1	0	1	1	0	1	3	2
12	SCORE	ROW12	20	1	0	1	1	0	1	3	2

The next model is a deviations-from-means model, but this time the reference cell is the first cell (1,1). This coding is also called effects coding. The following statements produce Figure 90.55 and Figure 90.56:

```
proc transreg data=x ss2 short;
   title2 'Deviations from Means, (1,1) Reference Cell';
   model identity(y) = class(a | b / deviations zero=first);
   output replace;
run;

proc print label;
run;
```

Figure 90.55 Deviations-from-Means Model, (1,1) Reference Cell

Two-Way ANOVA Models
Deviations from Means, (1,1) Reference Cell

The TRANSREG Procedure

Dependent Variable Identity(y)

Class Level Information

Class	Levels	Values
a	3	1 2 3
b	2	1 2

Number of Observations Read	12
Number of Observations Used	12

The TRANSREG Procedure Hypothesis Tests for Identity(y)

Univariate ANOVA Table Based on the Usual Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	234.6667	46.93333	3.20	0.0946
Error	6	88.0000	14.66667		
Corrected Total	11	322.6667			

Root MSE	3.82971	R-Square	0.7273
Dependent Mean	13.33333	Adj R-Sq	0.5000
Coeff Var	28.72281		

Univariate Regression Table Based on the Usual Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Pr > F	Label
Intercept	1	13.3333333	2133.33	2133.33	145.45	<.0001	Intercept
Class.a2	1	-2.8333333	48.17	48.17	3.28	0.1199	a 2
Class.a3	1	1.6666667	16.67	16.67	1.14	0.3274	a 3
Class.b2	1	3.0000000	108.00	108.00	7.36	0.0349	b 2
Class.a2b2	1	2.5000000	37.50	37.50	2.56	0.1609	a 2 * b 2
Class.a3b2	1	1.0000000	6.00	6.00	0.41	0.5461	a 3 * b 2

The parameter estimates are

$$
\begin{aligned}
\hat{\mu} &= \overline{y} = 13.3333 \\
\hat{\alpha}_2 &= (\overline{y}_{21} + \overline{y}_{22})/2 - \overline{y} = (5+16)/2 - 13.3333 = -2.8333 \\
\hat{\alpha}_3 &= (\overline{y}_{31} + \overline{y}_{32})/2 - \overline{y} = (11+19)/2 - 13.3333 = 1.6667 \\
\hat{\beta}_2 &= (\overline{y}_{12} + \overline{y}_{22} + \overline{y}_{32})/3 - \overline{y} = (14+16+19)/3 - 13.3333 = 3 \\
\hat{\gamma}_{22} &= \overline{y}_{22} - (\overline{y} + \hat{\alpha}_2 + \hat{\beta}_2) = 16 - (13.3333 + -2.8333 + 3) = 2.5 \\
\hat{\gamma}_{32} &= \overline{y}_{32} - (\overline{y} + \hat{\alpha}_3 + \hat{\beta}_2) = 19 - (13.3333 + 1.6667 + 3) = 1
\end{aligned}
$$

Notice that all of the parameter estimates match the corresponding estimates from the less-than-full-rank coding.

Figure 90.56 Deviations-from-Means Model, (1,1) Reference Cell, Design Matrix

Two-Way ANOVA Models
Deviations from Means, (1,1) Reference Cell

Obs	_TYPE_	_NAME_	y	Intercept	a 2	a 3	b 2	a 2 * b 2	a 3 * b 2	a	b
1	SCORE	ROW1	16	1	-1	-1	-1	1	1	1	1
2	SCORE	ROW2	14	1	-1	-1	-1	1	1	1	1
3	SCORE	ROW3	15	1	-1	-1	1	-1	-1	1	2
4	SCORE	ROW4	13	1	-1	-1	1	-1	-1	1	2
5	SCORE	ROW5	1	1	1	0	-1	-1	0	2	1
6	SCORE	ROW6	9	1	1	0	-1	-1	0	2	1
7	SCORE	ROW7	12	1	1	0	1	1	0	2	2
8	SCORE	ROW8	20	1	1	0	1	1	0	2	2
9	SCORE	ROW9	14	1	0	1	-1	0	-1	3	1
10	SCORE	ROW10	8	1	0	1	-1	0	-1	3	1
11	SCORE	ROW11	18	1	0	1	1	0	1	3	2
12	SCORE	ROW12	20	1	0	1	1	0	1	3	2

The following statements fit a model with an orthogonal-contrast coding and produce Figure 90.57 and Figure 90.58:

```
proc transreg data=x ss2 short;
   title2 'Orthogonal Contrast Coding';
   model identity(y) = class(a | b / orthogonal);
   output replace;
run;

proc print label;
run;
```

Figure 90.57 Orthogonal-Contrast Coding

Two-Way ANOVA Models
Orthogonal Contrast Coding

The TRANSREG Procedure

Dependent Variable Identity(y)

Class Level Information

Class	Levels	Values
a	3	1 2 3
b	2	1 2

Number of Observations Read	12
Number of Observations Used	12

The TRANSREG Procedure Hypothesis Tests for Identity(y)

Univariate ANOVA Table Based on the Usual Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	234.6667	46.93333	3.20	0.0946
Error	6	88.0000	14.66667		
Corrected Total	11	322.6667			

Root MSE	3.82971	R-Square	0.7273
Dependent Mean	13.33333	Adj R-Sq	0.5000
Coeff Var	28.72281		

Univariate Regression Table Based on the Usual Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Pr > F	Label
Intercept	1	13.3333333	2133.33	2133.33	145.45	<.0001	Intercept
Class.a1	1	-0.2500000	0.50	0.50	0.03	0.8596	a 1
Class.a2	1	-1.4166667	48.17	48.17	3.28	0.1199	a 2
Class.b1	1	-3.0000000	108.00	108.00	7.36	0.0349	b 1
Class.a1b1	1	2.2500000	40.50	40.50	2.76	0.1476	a 1 * b 1
Class.a2b1	1	-1.2500000	37.50	37.50	2.56	0.1609	a 2 * b 1

The parameter estimates are

$$
\begin{aligned}
\hat{\mu} &= \overline{y} = 13.3333 \\
\hat{\alpha}_1 &= ((\overline{y}_{11} + \overline{y}_{12}) - (\overline{y}_{31} + \overline{y}_{32}))/4 = ((15 + 14) - (11 + 19))/4 = -0.25 \\
\hat{\alpha}_2 &= ((\overline{y}_{21} + \overline{y}_{22}) - (\overline{y}_{11} + \overline{y}_{12} + \overline{y}_{31} + \overline{y}_{32})/2)/6 \\
&= ((5 + 16) - (15 + 14 + 11 + 19)/2)/6 = -1.417 \\
\hat{\beta}_1 &= ((\overline{y}_{11} + \overline{y}_{21} + \overline{y}_{31}) - (\overline{y}_{12} + \overline{y}_{22} + \overline{y}_{32}))/6 \\
&= ((15 + 5 + 11) - (14 + 16 + 19))/6 = -3 \\
\hat{\gamma}_{11} &= (\overline{y}_{11} - \overline{y}_{12} - \overline{y}_{31} + \overline{y}_{32})/4 = (15 - 14 - 11 + 19)/4 = 2.25 \\
\hat{\gamma}_{21} &= ((-\overline{y}_{11} + \overline{y}_{12} - \overline{y}_{31} + \overline{y}_{32})/2 + (\overline{y}_{21} - \overline{y}_{22}))/6 \\
&= ((-15 + 14 - 11 + 19)/2 + (5 - 16))/6 = -1.25
\end{aligned}
$$

Figure 90.58 Orthogonal-Contrast Coding, Design Matrix

Two-Way ANOVA Models
Orthogonal Contrast Coding

Obs	_TYPE_	_NAME_	y	Intercept	a 1	a 2	b 1	a 1 * b 1	a 2 * b 1	a	b
1	SCORE	ROW1	16	1	1	-1	1	1	-1	1	1
2	SCORE	ROW2	14	1	1	-1	1	1	-1	1	1
3	SCORE	ROW3	15	1	1	-1	-1	-1	1	1	2
4	SCORE	ROW4	13	1	1	-1	-1	-1	1	1	2
5	SCORE	ROW5	1	1	0	2	1	0	2	2	1
6	SCORE	ROW6	9	1	0	2	1	0	2	2	1
7	SCORE	ROW7	12	1	0	2	-1	0	-2	2	2
8	SCORE	ROW8	20	1	0	2	-1	0	-2	2	2
9	SCORE	ROW9	14	1	-1	-1	1	-1	-1	3	1
10	SCORE	ROW10	8	1	-1	-1	1	-1	-1	3	1
11	SCORE	ROW11	18	1	-1	-1	-1	1	1	3	2
12	SCORE	ROW12	20	1	-1	-1	-1	1	1	3	2

The following statements fit a model with a standardized-orthogonal coding and produce Figure 90.59 and Figure 90.60:

```
proc transreg data=x ss2 short;
   title2 'Standardized-Orthogonal Coding';
   model identity(y) = class(a | b / standorth);
   output replace;
run;

proc print label;
run;
```

Figure 90.59 Standardized-Orthogonal Coding

Two-Way ANOVA Models
Standardized-Orthogonal Coding

The TRANSREG Procedure

Dependent Variable Identity(y)

Class Level Information

Class	Levels	Values
a	3	1 2 3
b	2	1 2

Number of Observations Read	12
Number of Observations Used	12

The TRANSREG Procedure Hypothesis Tests for Identity(y)

Univariate ANOVA Table Based on the Usual Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	5	234.6667	46.93333	3.20	0.0946
Error	6	88.0000	14.66667		
Corrected Total	11	322.6667			

Root MSE	3.82971	R-Square	0.7273
Dependent Mean	13.33333	Adj R-Sq	0.5000
Coeff Var	28.72281		

Univariate Regression Table Based on the Usual Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Pr > F	Label
Intercept	1	13.3333333	2133.33	2133.33	145.45	<.0001	Intercept
Class.a1	1	-0.2041241	0.50	0.50	0.03	0.8596	a 1
Class.a2	1	-2.0034692	48.17	48.17	3.28	0.1199	a 2
Class.b1	1	-3.0000000	108.00	108.00	7.36	0.0349	b 1
Class.a1b1	1	1.8371173	40.50	40.50	2.76	0.1476	a 1 * b 1
Class.a2b1	1	-1.7677670	37.50	37.50	2.56	0.1609	a 2 * b 1

The parameter estimates are

$$
\begin{aligned}
\hat{\mu} &= \overline{y} = 13.3333 \\
\hat{\alpha}_1 &= (((\overline{y}_{11} + \overline{y}_{12}) - (\overline{y}_{31} + \overline{y}_{32}))/4) \times \sqrt{2/3} \\
&= (((15 + 14) - (11 + 19))/4) \times \sqrt{2/3} = -0.2041 \\
\hat{\alpha}_2 &= (((\overline{y}_{21} + \overline{y}_{22}) - (\overline{y}_{11} + \overline{y}_{12} + \overline{y}_{31} + \overline{y}_{32})/2)/6) \times \sqrt{6/3} \\
&= (((5 + 16) - (15 + 14 + 11 + 19)/2)/6) \times \sqrt{6/3} = -2.0035 \\
\hat{\beta}_1 &= (((\overline{y}_{11} + \overline{y}_{21} + \overline{y}_{31}) - (\overline{y}_{12} + \overline{y}_{22} + \overline{y}_{32}))/6) \times \sqrt{2/2} \\
&= (((15 + 5 + 11) - (14 + 16 + 19))/6) \times \sqrt{2/2} = -3 \\
\hat{\gamma}_{11} &= ((\overline{y}_{11} - \overline{y}_{12} - \overline{y}_{31} + \overline{y}_{32})/4) \times \sqrt{2/3} \times \sqrt{2/2} \\
&= ((15 - 14 - 11 + 19)/4) \times \sqrt{2/3} \times \sqrt{2/2} = 1.8371 \\
\hat{\gamma}_{21} &= (((-\overline{y}_{11} + \overline{y}_{12} - \overline{y}_{31} + \overline{y}_{32})/2 + (\overline{y}_{21} - \overline{y}_{22}))/6) \times \sqrt{6/3} \times \sqrt{2/2} \\
&= (((-15 + 14 - 11 + 19)/2 + (5 - 16))/6) \times \sqrt{6/3} \times \sqrt{2/2} = -1.7678
\end{aligned}
$$

The numerators in the square roots are sums of squares of the coded values for the unstandardized-orthogonal codings, and the denominators are the numbers of levels. These terms convert the estimates from the orthogonal contrast coding to the standardized-orthogonal coding. The term $\sqrt{2/2}$, which is 1 and could be dropped, is included in the preceding formulas to show the general pattern. Notice the regression tables for the orthogonal-contrast coding and the standardized-orthogonal coding. Some of the coefficients are different, but the rest of the table is the same since the coded variables for the two models differ only by a constant.

Figure 90.60 Standardized-Orthogonal Coding, Design Matrix

Two-Way ANOVA Models
Standardized-Orthogonal Coding

Obs	_TYPE_	_NAME_	y	Intercept	a 1	a 2	b 1	a 1 * b 1	a 2 * b 1	a b
1	SCORE	ROW1	16	1	1.22474	-0.70711	1	1.22474	-0.70711	1 1
2	SCORE	ROW2	14	1	1.22474	-0.70711	1	1.22474	-0.70711	1 1
3	SCORE	ROW3	15	1	1.22474	-0.70711	-1	-1.22474	0.70711	1 2
4	SCORE	ROW4	13	1	1.22474	-0.70711	-1	-1.22474	0.70711	1 2
5	SCORE	ROW5	1	1	0.00000	1.41421	1	0.00000	1.41421	2 1
6	SCORE	ROW6	9	1	0.00000	1.41421	1	0.00000	1.41421	2 1
7	SCORE	ROW7	12	1	0.00000	1.41421	-1	0.00000	-1.41421	2 2
8	SCORE	ROW8	20	1	0.00000	1.41421	-1	0.00000	-1.41421	2 2
9	SCORE	ROW9	14	1	-1.22474	-0.70711	1	-1.22474	-0.70711	3 1
10	SCORE	ROW10	8	1	-1.22474	-0.70711	1	-1.22474	-0.70711	3 1
11	SCORE	ROW11	18	1	-1.22474	-0.70711	-1	1.22474	0.70711	3 2
12	SCORE	ROW12	20	1	-1.22474	-0.70711	-1	1.22474	0.70711	3 2

Missing Values

PROC TRANSREG can estimate missing values, with or without category or monotonicity constraints, so that the regression model fit is optimized. Several approaches to missing data handling are provided. All observations with missing values in IDENTITY, CLASS, POINT, EPOINT, QPOINT, SMOOTH, PBSPLINE, PSPLINE, and BSPLINE variables are excluded from the analysis. When METHOD=UNIVARIATE (specified in the PROC TRANSREG or MODEL statement), observations with missing values in any of the independent variables are excluded from the analysis. When you specify the NOMISS *a-option*, observations with missing values in the other analysis variables are excluded. Otherwise, missing data are estimated, and the variable means are the initial estimates.

You can specify the LINEAR, OPSCORE, MONOTONE, UNTIE, SPLINE, MSPLINE, SSPLINE, LOG, LOGIT, POWER, ARSIN, BOXCOX, RANK, and EXP transformations in any combination with nonmissing values, ordinary missing values, and special missing values, as long as the nonmissing values in each variable have positive variance. No category or order restrictions are placed on the estimates of ordinary missing values. You can force missing value estimates within a variable to be identical by using special missing values (see "DATA Step Processing" in *SAS Language Reference: Concepts*. You can specify up to 27 categories of missing values, in which within-category estimates must be the same, by coding the missing values with ._ and .A through .Z.

You can also specify an ordering of some missing value estimates. You can use the MONOTONE= *a-option* in the PROC TRANSREG or MODEL statement to indicate a range of special missing values (a subset of the list from .A to .Z) with estimates that must be weakly ordered within each variable in which they appear. For example, if MONOTONE=AI, the nine classes, .A, .B, . . . , .I, are monotonically scored and optimally scaled just as MONOTONE transformation values are scored. In this case, category but not order restrictions are placed on the missing values ._ and .J through .Z. You can also use the UNTIE= *a-option* (in the PROC TRANSREG or MODEL statement) to indicate a range of special missing values with estimates that must be weakly ordered within each variable in which they appear but can be untied.

The missing value estimation facilities enable you to have partitioned or mixed-type variables. For example, a variable can be considered part nominal and part ordinal. Nominal classes of otherwise ordinal variables are coded with special missing values. This feature can be useful with survey research. The class "unfamiliar with the product" in the variable "Rate your preference for 'Brand X' on a 1 to 9 scale, or if you are unfamiliar with the product, check 'unfamiliar with the product"' is an example. You can code "unfamiliar with the product" as a special missing value, such as .A. The 1s to 9s can be monotonically transformed, while no monotonic restrictions are placed on the quantification of the "unfamiliar with the product" class.

A variable specified for a LINEAR transformation, with special missing values and ordered categorical missing values, can be part interval, part ordinal, and part nominal. A variable specified for a MONOTONE transformation can have two independent ordinal parts. A variable specified for an UNTIE transformation can have an ordered categorical part and an ordered part without category restrictions. Many other mixes are possible.

Missing Values, UNTIE, and Hypothesis Tests

PROC TRANSREG can estimate missing data and monotonically transform variables while untying tied values. Estimates of ordinary missing values (.) are all permitted to be different. Analyses with UNTIE transformations, the UNTIE= *a-option*, and ordinary missing data estimation are all prone to degeneracy problems. Consider the following example. A perfect fit is found by collapsing all observations except the one with two missing values into a single value in y and x1. The following statements produce Figure 90.61:

```
title 'Missing Data';

data x;
   input y x1 x2 @@;
   datalines;
1 3 7    8 3 9    1 8 6    . . 9    3 3 9
8 5 1    6 7 3    2 7 2    1 8 2    . 9 1
;
proc transreg solve;
   model linear(y) = linear(x1 x2);
   output;
run;

proc print;
run;
```

Figure 90.61 Missing Values Example

Missing Data

Obs	_TYPE_	_NAME_	y	Ty	Intercept	x1	x2	TIntercept	Tx1	Tx2
1	SCORE	ROW1	1	2.7680	1	3	7	1	5.1233	7
2	SCORE	ROW2	8	2.7680	1	3	9	1	5.1233	9
3	SCORE	ROW3	1	2.7680	1	8	6	1	5.1233	6
4	SCORE	ROW4	.	12.5878	1	.	9	1	12.7791	9
5	SCORE	ROW5	3	2.7680	1	3	9	1	5.1233	9
6	SCORE	ROW6	8	2.7680	1	5	1	1	5.1233	1
7	SCORE	ROW7	6	2.7680	1	7	3	1	5.1233	3
8	SCORE	ROW8	2	2.7680	1	7	2	1	5.1233	2
9	SCORE	ROW9	1	2.7680	1	8	2	1	5.1233	2
10	SCORE	ROW10	.	2.7680	1	9	1	1	5.1233	1

Generally, the use of ordinary missing data estimation, the UNTIE transformation, and the UNTIE= *a-option* should be avoided, particularly with hypothesis tests. With these options, parameters are estimated based on only a single observation, and they can exert tremendous influence over the results. Each of these parameters has one model degree of freedom associated with it, so small or zero error degrees of freedom can also be a problem.

Controlling the Number of Iterations

Several *a-options* in the PROC TRANSREG or MODEL statement control the number of iterations performed. Iteration terminates when any one of the following conditions is satisfied:

- The number of iterations equals the value of the MAXITER= *a-option*.
- The average absolute change in variable scores from one iteration to the next is less than the value of the CONVERGE= *a-option*.
- The criterion change is less than the value of the CCONVERGE= *a-option*.

You can specify negative values for either convergence *a-option* if you want to define convergence only in terms of the other option. The criterion change can become negative when the data have converged, so it is numerically impossible, within machine precision, to increase the criterion. Usually, a negative criterion change is the result of very small amounts of rounding error, since the algorithms are (usually) convergent. However, there are cases where a negative criterion change is a sign of divergence, which is not necessarily an error. When you specify an SSPLINE transformation or the REITERATE or SOLVE *a-option*, divergence is perfectly normal.

When there are no monotonicity constraints and there is only one canonical variable in each set, PROC TRANSREG (with the SOLVE *a-option*) can usually find the optimal solution in only one iteration. (There are no monotonicity constraints when none of the following is specified: MONOTONE, MSPLINE, or UNTIE transformation or the UNTIE= or MONOTONE= *a-option*. There is only one canonical variable in each set when METHOD=MORALS or METHOD=UNIVARIATE, or when METHOD=REDUNDANCY with only one dependent variable, or when METHOD=CANALS and NCAN=1.)

The initialization iteration is number 0. When there are no monotonicity constraints and there is only one canonical variable in each set, the next iteration shows no change, and iteration stops. At least two iterations (0 and 1) are performed with the SOLVE *a-option* even if nothing changes in iteration 0. The MONOTONE, MSPLINE, and UNTIE variables are not transformed by the canonical initialization. Note that divergence with the SOLVE *a-option*, particularly in the second iteration, is not an error. The initialization iteration is slower and uses more memory than other iterations. However, for many models, specifying the SOLVE *a-option* can greatly decrease the amount of time required to find the optimal transformations.

You can increase the number of iterations to ensure convergence by increasing the value of the MAXITER= *a-option* and decreasing the value of the CONVERGE= *a-option*. Since the average absolute change in standardized variable scores seldom decreases below 1E–11, you should not specify a value for the CONVERGE= *a-option* less than 1E–8 or 1E–10. Most of the data changes occur during the first few iterations, but the data can still change after 50 or even 100 iterations. You can try different combinations of values for the CONVERGE= and MAXITER= *a-options* to ensure convergence without extreme overiteration. If the data do not converge with the default specifications, try CONVERGE=1E–8 and MAXITER=50, or CONVERGE=1E–10 and MAXITER=200. Note that you can specify the REITERATE *a-option* to start iterating where the previous analysis stopped.

Using the REITERATE Algorithm Option

You can use the REITERATE *a-option* to perform additional iterations when PROC TRANSREG stops before the data have adequately converged. For example, suppose that you execute the following step:

```
proc transreg data=a;
   model mspline(y) = mspline(x1-x5);
   output out=b coefficients;
run;
```

If the transformations do not converge in the default 30 iterations, you can perform more iterations without repeating the first 30 iterations, as follows:

```
proc transreg data=b reiterate;
   model mspline(y) = mspline(x1-x5);
   output out=b coefficients;
run;
```

Note that a WHERE statement is not necessary to exclude the coefficient observations. They are automatically excluded because their _TYPE_ value is not SCORE.

You can also use the REITERATE *a-option* to specify starting values other than the original values for the transformations. Providing alternate starting points might help avoid local optima. Here are two examples:

```
proc transreg data=a;
   model rank(y) = rank(x1-x5);
   output out=b;
run;

proc transreg data=b reiterate;
   /* Use ranks as the starting point. */
   model mspline(y) = mspline(x1-x5);
   output out=c coefficients;
run;

data b;
   set a;
   array tx[6] ty tx1-tx5;
   do j = 1 to 6;
      tx[j] = normal(7);
   end;
run;

proc transreg data=b reiterate;
   /* Use a random starting point. */
   model mspline(y) = mspline(x1-x5);
   output out=c coefficients;
run;
```

Note that divergence with the REITERATE *a-option*, particularly in the second iteration, is not an error since the initial transformation is not required to be a valid member of the transformation family. When you specify the REITERATE *a-option*, the iteration does not terminate when the criterion change is negative during the first 10 iterations.

Avoiding Constant Transformations

There are times when the optimal scaling produces a constant transformed variable. This can happen with the MONOTONE, UNTIE, and MSPLINE transformations when the target is negatively correlated with the original input variable. It can happen with all transformations when the target is uncorrelated with the original input variable. When this happens, the procedure modifies the target to avoid a constant transformation. This strategy avoids certain nonoptimal solutions.

If the transformation is monotonic and a constant transformed variable results, the procedure multiplies the target by –1 and tries the optimal scaling again. If the transformation is not monotonic or if the multiplication by –1 did not help, the procedure tries using a random target. If the transformation is still constant, the previous nonconstant transformation is retained. When a constant transformation is avoided by any strategy, this message is displayed: "A constant transformation was avoided for *name*."

With extreme collinearity, small amounts of rounding error might interact with the instability of the coefficients to produce target vectors that are not positively correlated with the original scaling. If a regression coefficient for a variable is zero, the formula for the target for that variable contains a zero divide. In a multiple regression model, after many iterations, one independent variable can be scaled the same way as the current scaling of the dependent variable, so the other independent variables have coefficients of zero. When the constant transformation warning appears, you should interpret your results with extreme caution, and recheck your model.

Constant Variables

Constant and almost constant variables are zeroed and ignored. When constant variables are expected and should not be zeroed, specify the NOZEROCONSTANT *a-option*.

Character OPSCORE Variables

Character OPSCORE variables are replaced by a numeric variable containing category numbers before the iterations, and the character values are discarded. Only the first eight characters are considered in determining category membership. If you want the original character variable in the output data set, give it a different name in the OPSCORE specification (OPSCORE(x / name=(x2)) and name the original variable in the ID statement (ID x;).

Convergence and Degeneracies

When you specify the SSPLINE transformation, divergence is normal. The rest of this section assumes that you did not specify SSPLINE. For all the methods available in PROC TRANSREG, the algorithms are convergent, in terms of both the criterion being optimized and the parameters being estimated. The value of the criterion being maximized (squared multiple correlation, average squared multiple correlation, or average squared canonical correlation) can, theoretically, never decrease from one iteration to the next. The values of the parameters being solved for (the scores and weights of the transformed variables) become stable after sufficient iteration.

In practice, the criterion being maximized can decrease with overiteration. When the statistic has very nearly reached its maximum, further iterations might report a decrease in the criterion in the last few decimal places. This is a normal result of very small amounts of rounding error. By default, iteration terminates when this occurs because, by default, CCONVERGE=0.0. Specifying CCONVERGE=–1, an impossible change, turns off this check for convergence.

Even though the algorithms are convergent, they might not converge to a global optimum. Also, under extreme circumstances, the solution might degenerate. Because two points always form a straight line, the algorithms sometimes try to reach this degenerate optimum. This sometimes occurs when one observation is an ordinal outlier (when one observation has the extreme rank on all variables). The algorithm can reach an optimal solution that ties all other categories producing two points. Similar results can occur when there are many missing values. More generally, whenever there are very few constraints on the scoring of one or more points, degeneracies can be a problem. In a well-behaved analysis, the maximum data change, average data change, and criterion change all decrease at a rapid rate with each iteration. When the rate of change increases for several iterations, the solution might be degenerating.

Implicit and Explicit Intercepts

Depending on several options, the model intercept is nonzero, zero, or implicit, or there is no intercept. Ordinarily, the model contains an explicit nonzero intercept, and the Intercept variable in the OUT= data set contains ones. When TSTANDARD=CENTER or TSTANDARD=Z is specified, the model contains an explicit, zero intercept and the Intercept variable contains zeros. When METHOD=CANALS, the model is fit with centered variables and the Intercept variable is set to missing.

If you specify CLASS with ZERO=NONE or BSPLINE for one or more independent variables, and TSTANDARD=NOMISS or TSTANDARD=ORIGINAL (the default), an implicit intercept model is fit. The intercept is implicit in a set of the independent variables since there exists a set of independent variables the sum of which is a column of ones. All statistics are mean corrected. The implicit intercept is not an option; it is implied by the model. Specifying SMOOTH or PBSPLINE also implies an implicit intercept model.

With METHOD=CANALS, the Intercept variable contains the *canonical intercept* for canonical coefficients observations: $\hat{\beta}_0 = \bar{\mathbf{y}}'\hat{\boldsymbol{\alpha}} - \bar{\mathbf{x}}'\hat{\boldsymbol{\beta}}$ where $\mathbf{Y}\hat{\boldsymbol{\alpha}} \approx \mathbf{X}\hat{\boldsymbol{\beta}}$.

Passive Observations

Observations can be excluded from the analysis for several reasons; these include zero weight; zero frequency; missing values in variables designated IDENTITY, CLASS, POINT, EPOINT, QPOINT, SMOOTH, PBSPLINE, PSPLINE, or BSPLINE; and missing values with the NOMISS *a-option* specified. These observations are passive in that they do not contribute to determining transformations, R square, sums of squares, degrees of freedom, and so on. However, some information can be computed for them. For example, if no independent variable values are missing, predicted values and redundancy variable values can both be computed. Residuals can be computed for observations with a nonmissing dependent and nonmissing predicted value. Canonical variables for dependent variables can be computed when no dependent variables are missing; canonical variables for independent variables can be computed when no independent variables are missing, and so on. Passive observations in the OUT= data set have a blank value for _TYPE_.

Point Models

The expanded set of independent variables generated from the POINT, EPOINT, and QPOINT expansions can be used to perform ideal point regressions (Carroll 1972) and compute ideal point coordinates for plotting in a biplot (Gabriel 1981). The three types of ideal point coordinates can all be described as transformed coefficients. Assume that m independent variables are specified in one of the three point expansions. Let $\mathbf{b}'$ be a $1 \times m$ row vector of coefficients for these variables and one of the dependent variables. Let $\mathbf{R}$ be a matrix created from the coefficients of the extra variables. When coordinates are requested with the MPC, MEC, or MQC *o-option*, $\mathbf{b}'$ and $\mathbf{R}$ are created from multiple regression coefficients. When coordinates are requested with the CPC, CEC, or CQC *o-option*, $\mathbf{b}'$ and $\mathbf{R}$ are created from canonical coefficients.

If you specify the POINT expansion in the MODEL statement, $\mathbf{R}$ is an $m \times m$ identity matrix times the coefficient for the sums of squares (_ISSQ_) variable. If you specify the EPOINT expansion, $\mathbf{R}$ is an $m \times m$ diagonal matrix of coefficients from the squared variables. If you specify the QPOINT expansion, $\mathbf{R}$ is an $m \times m$ symmetric matrix of coefficients from the squared variables on the diagonal and crossproduct variables off the diagonal. The MPC, MEC, MQC, CPC, CEC, and CQC ideal point coordinates are defined as $-0.5\mathbf{b}'\mathbf{R}^{-1}$. When $\mathbf{R}$ is singular, the ideal point coordinates are infinitely far away and are set to missing, so you should try a simpler version of the model. The version that is simpler than the POINT model is the vector model, where no extra variables are created. In the vector model, designate all independent variables as IDENTITY. Then draw vectors from the origin to the COEFFICIENTS points.

Typically, when you request ideal point coordinates, the MODEL statement should consist of a single transformation for the dependent variables (usually IDENTITY, MONOTONE, or MSPLINE) and a single expansion for the independent variables (one of POINT, EPOINT, or QPOINT).

Redundancy Analysis

Redundancy analysis (Stewart and Love 1968) is a principal component analysis of multivariate regression predicted values. These first steps show the redundancy analysis results produced by PROC TRANSREG. The specification TSTANDARD=Z sets all variables to mean zero and variance one. METHOD=REDUNDANCY specifies redundancy analysis and outputs the redundancy variables to the OUT= data set. The MREDUNDANCY *o-option* outputs two sets of redundancy analysis coefficients to the OUT= data set. The following statements produce Figure 90.62:

```
title 'Redundancy Analysis';

data x;
   input y1-y3 x1-x4;
   datalines;
 6  8  8 15 18 26 27
 1 12 16 18  9 20  8
 5  6 15 20 17 29 31
 6  9 15 14 10 16 22
 7  5 12 14  6 13  9
 3  6  7  2 14 26 22
 3  5  9 13 18 10 22
 6  3 11  3 15 22 29
 6  3  7 10 20 21 27
 7  5  9  8 10 12 18
;

proc transreg data=x tstandard=z method=redundancy;
   model identity(y1-y3) = identity(x1-x4);
   output out=red mredundancy replace;
run;

proc print data=red(drop=Intercept);
   format _numeric_ 4.1;
run;
```

Figure 90.62 Redundancy Analysis Example

Redundancy Analysis

Obs	_TYPE_	_NAME_	y1	y2	y3	x1	x2	x3	x4	Red1	Red2	Red3
1	SCORE	ROW1	0.5	0.6	−0.8	0.6	0.9	1.0	0.7	0.2	−0.5	−0.9
2	SCORE	ROW2	−2.0	2.1	1.5	1.1	−1.0	0.1	−1.7	1.6	−1.5	0.4
3	SCORE	ROW3	0.0	−0.1	1.2	1.4	0.7	1.5	1.2	1.0	0.8	−1.3
4	SCORE	ROW4	0.5	1.0	1.2	0.4	−0.8	−0.5	0.1	0.5	1.7	0.1
5	SCORE	ROW5	1.0	−0.4	0.3	0.4	−1.6	−1.0	−1.6	1.0	0.1	0.9
6	SCORE	ROW6	−1.0	−0.1	−1.1	−1.6	0.1	1.0	0.1	−0.8	−0.9	1.4
7	SCORE	ROW7	−1.0	−0.4	−0.6	0.2	0.9	−1.5	0.1	−1.0	−0.4	−1.3
8	SCORE	ROW8	0.5	−1.2	0.0	−1.5	0.3	0.4	1.0	−1.2	0.8	0.7
9	SCORE	ROW9	0.5	−1.2	−1.1	−0.3	1.3	0.2	0.7	−1.0	−0.9	−0.8
10	SCORE	ROW10	1.0	−0.4	−0.6	−0.6	−0.8	−1.1	−0.4	−0.4	0.8	0.7
11	M REDUND	Red1	.	.	.	0.7	−0.6	0.4	−0.1	.	.	.
12	M REDUND	Red2	.	.	.	0.3	−1.5	−0.6	1.9	.	.	.
13	M REDUND	Red3	.	.	.	−0.7	−0.7	0.3	−0.3	.	.	.
14	R REDUND	x1	.	.	.	.	.	.	.	0.8	−0.0	−0.6
15	R REDUND	x2	.	.	.	.	.	.	.	−0.6	−0.2	−0.7
16	R REDUND	x3	.	.	.	.	.	.	.	0.1	−0.2	−0.1
17	R REDUND	x4	.	.	.	.	.	.	.	−0.5	0.3	−0.5

The _TYPE_='SCORE' observations of the Red1–Red3 variables contain the redundancy variables. The nonmissing "M REDUND" values are coefficients for predicting the redundancy variables from the independent variables. The nonmissing "R REDUND" values are coefficients for predicting the independent variables from the redundancy variables.

The next steps show how to generate the same results manually. The data set is standardized, predicted values are computed, and principal components of the predicted values are computed. The following statements produce the redundancy variables, shown in Figure 90.63:

```
proc standard data=x out=std m=0 s=1;
   title2 'Manually Generate Redundancy Variables';
run;

proc reg noprint data=std;
   model y1-y3 = x1-x4;
   output out=p p=ay1-ay3;
run; quit;

proc princomp data=p cov noprint std out=p;
   var ay1-ay3;
run;

proc print data=p(keep=Prin:);
   format _numeric_ 4.1;
run;
```

Figure 90.63 Redundancy Analysis Example

Redundancy Analysis
Manually Generate Redundancy Variables

Obs	Prin1	Prin2	Prin3
1	0.2	−0.5	−0.9
2	1.6	−1.5	0.4
3	1.0	0.8	−1.3
4	0.5	1.7	0.1
5	1.0	0.1	0.9
6	−0.8	−0.9	1.4
7	−1.0	−0.4	−1.3
8	−1.2	0.8	0.7
9	−1.0	−0.9	−0.8
10	−0.4	0.8	0.7

The following statements produce the coefficients for predicting the redundancy variables from the independent variables, shown in Figure 90.64:

```
proc reg data=p outest=redcoef noprint;
   title2 'Manually Create Redundancy Coefficients';
   model Prin1-Prin3 = x1-x4;
run; quit;

proc print data=redcoef(keep=x1-x4);
   format _numeric_ 4.1;
run;
```

Figure 90.64 Redundancy Analysis Example

Redundancy Analysis
Manually Create Redundancy Coefficients

Obs	x1	x2	x3	x4
1	0.7	−0.6	0.4	−0.1
2	0.3	−1.5	−0.6	1.9
3	−0.7	−0.7	0.3	−0.3

The following statements produce the coefficients for predicting the independent variables from the redundancy variables, shown in Figure 90.65:

```
proc reg data=p outest=redcoef2 noprint;
   title2 'Manually Create Other Coefficients';
   model x1-x4 = prin1-prin3;
run; quit;

proc print data=redcoef2(keep=Prin1-Prin3);
   format _numeric_ 4.1;
run;
```

Figure 90.65 Redundancy Analysis Example

Redundancy Analysis
Manually Create Other Coefficients

Obs	Prin1	Prin2	Prin3
1	0.8	−0.0	−0.6
2	−0.6	−0.2	−0.7
3	0.1	−0.2	−0.1
4	−0.5	0.3	−0.5

Optimal Scaling

An alternating least-squares optimal scaling algorithm can be divided into two major stages. The first major stage estimates the parameters of the linear model. These parameters are used to create the predicted values or target for each variable that can be transformed. Each target minimizes squared error (as explained in the discussion of the algorithms in *SAS Technical Report R-108*). The definition of the target depends on many factors, such as whether a variable is independent or dependent, which algorithm is used (for example, regression, redundancy, CANALS, or principal components), and so on. The definition of the target is independent of the transformation family you specify for the variable. However, the target values for a variable typically do not fit the prescribed transformation family for the variable. They might not have the right category structure; they might not have the right order; they might not be a linear combination of the columns of a B-spline basis; and so on.

The second major stage is optimal scaling. Optimal scaling can be defined as a possibly constrained, least-squares regression problem. When you specify an optimal transformation, or when missing data are estimated for any variable, the full representation of the variable is not simply a vector; it is a matrix with more than one column. The optimal scaling phase finds the vector that is a linear combination of the columns of this matrix that is closest to the target (in terms of minimum squared error), among those that do not violate any of the constraints imposed by the transformation family. Optimal scaling methods are independent of the data analysis method that generated the target. In all cases, optimal scaling can be accomplished by creating a design matrix based on the original scaling of the variable and the transformation family specified for that variable. The optimally scaled variable is a linear combination of the columns of the design matrix. The coefficients of the linear combination are found by using (possibly constrained) least squares. Many optimal scaling problems are solved without actually constructing design and projection matrices. The next two sections describe the algorithms used by PROC TRANSREG for optimal scaling. The first section discusses optimal scaling for OPSCORE, MONOTONE, UNTIE, and LINEAR transformations, including how missing values are handled. The second section addresses SPLINE and MSPLINE transformations.

OPSCORE, MONOTONE, UNTIE, and LINEAR Transformations

Two vectors of information are needed to produce the optimally scaled variable: the initial variable scaling vector **x** and the target vector **y**. For convenience, both vectors are first sorted on the values of the initial scaling vector. If you request an UNTIE transformation, the target vector is sorted within ties in the initial scaling vector. The normal SAS collating sequence for missing and nonmissing values is used. Sorting simply permits the constraints to be specified in terms of relationships among adjoining coefficients. The sorting process partitions **x** and **y** into missing and nonmissing parts $(\mathbf{x}'_m\mathbf{x}'_n)'$, and $(\mathbf{y}'_m\mathbf{y}'_n)'$.

Next, PROC TRANSREG determines category membership. Every ordinary missing value (.) forms a separate category. (Three ordinary missing values form three categories.) Every special missing value within the range specified in the UNTIE= *a-option* forms a separate category. (If UNTIE= BC and there are three .B and two .C missing values, five categories are formed from them.) For all other special missing values, a separate category is formed for each different value. (If there are four .A missing values, one category is formed from them.)

Each distinct nonmissing value forms a separate category for OPSCORE and MONOTONE transformations (1 1 1 2 2 3 form three categories). Each nonmissing value forms a separate category for all other transformations (1 1 1 2 2 3 form six categories). When category membership is determined, category means are computed. Here is an example:

x:	**(. . .A .A .B 1 1 1 2 2 3 3 3 4)′**
y:	**(5 6 2 4 2 1 2 3 4 6 4 5 6 7)′**
OPSCORE and MONOTONE means:	**(5 6 3 2 2 5 5 7)′**
other means:	**(5 6 3 2 1 2 3 4 6 4 5 6 7)′**

The category means are the coefficients of a category indicator design matrix. The category means are the Fisher (1938) optimal scores. For MONOTONE and UNTIE transformations, order constraints are imposed on the category means for the nonmissing partition by merging categories that are out of order. The algorithm checks upward until an order violation is found, and then averages downward until the order violation is averaged away. (The average of $\bar{x}_1$ computed from n_1 observations and $\bar{x}_2$ computed from n_2 observations is $(n_1\bar{x}_1 + n_2\bar{x}_2)/(n_1 + n_2)$.) The MONOTONE algorithm (Kruskal 1964, secondary approach to ties) for this example with means for the nonmissing values (2 5 5 7)′ would do the following checks: $2 < 5$: OK, $5 = 5$: OK, $5 < 7$: OK. The means are in the proper order, so no work is needed.

The UNTIE transformation (Kruskal 1964, primary approach to ties) uses the same algorithm on the means of the nonmissing values (1 2 3 4 6 4 5 6 7)′ but with different results for this example: $1 < 2$: OK, $2 < 3$: OK, $3 < 4$: OK, $4 < 6$: OK, $6 > 4$: average 6 and 4 and replace 6 and 4 by the average. The new means of the nonmissing values are (1 2 3 4 5 5 5 6 7)′. The check resumes: $4 < 5$: OK, $5 = 5$: OK, $5 = 5$: OK, $5 < 6$: OK, $6 < 7$: OK. If some of the special missing values are ordered, the upward-checking, downward-averaging algorithm is applied to them also, independently of the other missing and nonmissing partitions. When the means conform to any required category or order constraints, an optimally scaled vector is produced from the means. The following example results from a MONOTONE transformation:

x:	(. . .A .A .B 1 1 1 2 2 3 3 3 4)′
y:	(5 6 2 4 2 1 2 3 4 6 4 5 6 7)′
result:	(5 6 3 3 2 2 2 2 5 5 5 5 5 7)′

The upward-checking, downward-averaging algorithm is equivalent to creating a category indicator design matrix, solving for least-squares coefficients with order constraints, and then computing the linear combination of design matrix columns.

For the optimal transformation LINEAR and for nonoptimal transformations, missing values are handled as just described. The nonmissing target values are regressed onto the matrix defined by the nonmissing initial scaling values and an intercept. In this example, the target vector $y_n = (1\ 2\ 3\ 4\ 6\ 4\ 5\ 6\ 7)'$ is regressed onto the design matrix

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 2 & 2 & 3 & 3 & 3 & 4 \end{bmatrix}'$$

Although only a linear transformation is performed, the effect of a linear regression optimal scaling is not eliminated by the later standardization step (unless the variable has no missing values). In the presence of missing values, the linear regression is necessary to minimize squared error.

SPLINE and MSPLINE Transformations

The missing portions of variables subjected to SPLINE or MSPLINE transformations are handled the same way as for OPSCORE, MONOTONE, UNTIE, and LINEAR transformations (see the previous section). The nonmissing partition is handled by first creating a B-spline basis of the specified degree with the specified knots for the nonmissing partition of the initial scaling vector and then regressing the target onto the basis. The optimally scaled vector is a linear combination of the B-spline basis vectors. Ordinary least-squares regression coefficients are used. An algorithm for generating the B-spline basis is given in de Boor (1978, pp. 134–135). B-splines are both a computationally accurate and efficient way of constructing a basis for piecewise polynomials; however, they are not the most natural method of describing splines.

Consider an initial scaling vector $x = (1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\ 9)'$ and a degree-three spline with interior knots at 3.5 and 6.5. The B-spline basis for the transformation is the left matrix, and the natural piecewise polynomial spline basis is the right matrix.

B-Spline Basis

$$\begin{bmatrix} 1.000 & 0.000 & 0.000 & 0.000 & 0 & 0 \\ 0.216 & 0.608 & 0.167 & 0.009 & 0 & 0 \\ 0.008 & 0.458 & 0.461 & 0.073 & 0 & 0 \\ 0 & 0.172 & 0.585 & 0.241 & 0.001 & 0 \\ 0 & 0.037 & 0.463 & 0.463 & 0.037 & 0 \\ 0 & 0.001 & 0.241 & 0.585 & 0.172 & 0 \\ 0 & 0 & 0.073 & 0.461 & 0.458 & 0.008 \\ 0 & 0 & 0.009 & 0.167 & 0.608 & 0.216 \\ 0 & 0 & 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix}$$

Piecewise Polynomial Splines

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 2 & 4 & 8 & 0 & 0 \\ 1 & 3 & 9 & 27 & 0 & 0 \\ 1 & 4 & 16 & 64 & 0.125 & 0 \\ 1 & 5 & 25 & 125 & 3.375 & 0 \\ 1 & 6 & 36 & 216 & 15.625 & 0 \\ 1 & 7 & 49 & 343 & 42.875 & 0.125 \\ 1 & 8 & 64 & 512 & 91.125 & 3.375 \\ 1 & 9 & 81 & 729 & 166.375 & 15.625 \end{bmatrix}$$

The two matrices span the same column space. The natural basis has an intercept, a linear term, a quadratic term, a cubic term, and two more terms since there are two interior knots. These terms are generated (for knot k and **x** element x) by the formula $(x-k)^3 \times I_{(x>k)}$. The indicator variable $I_{(x>k)}$ evaluates to 1.0 if x is greater than k and to 0.0 otherwise. If knot k had been repeated, there would be a $(x-k)^2 \times I_{(x>k)}$ term also. Notice that the fifth column makes no contribution to the curve before 3.5, makes zero contribution at 3.5 (the transformation is continuous), and makes an increasing contribution beyond 3.5. The same pattern of results holds for the last term with knot 6.5. The coefficient of the fifth column represents the change in the cubic portion of the curve after 3.5. The coefficient of the sixth column represents the change in the cubic portion of the curve after 6.5.

The numbers in the B-spline basis do not have a simple interpretation like the numbers in the natural piecewise polynomial basis. The B-spline basis has a diagonally banded structure. The band shifts one column to the right after every knot. The number of entries in each row that can potentially be nonzero is one greater than the degree. The elements within a row always sum to one. The B-spline basis is accurate because of the smallness of the numbers and the lack of extreme collinearity inherent in the natural polynomials. B-splines are efficient because PROC TRANSREG can take advantage of the sparseness of the B-spline basis when it accumulates crossproducts. The number of required multiplications and additions to accumulate the crossproduct matrix does not increase with the number of knots but does increase with the degree of the spline, so it is much more computationally efficient to increase the number of knots than to increase the degree of the polynomial.

MSPLINE transformations are handled like SPLINE transformations except that constraints are placed on the coefficients to ensure monotonicity. When the coefficients of the B-spline basis are monotonically increasing, the transformation is monotonically increasing. When the polynomial degree is two or less, monotone coefficient splines, integrated splines (Winsberg and Ramsay 1980), and the general class of all monotone splines are equivalent.

Specifying the Number of Knots

Keep the number of knots small (usually less than 10, although you can specify more). A degree-three spline with nine knots, one at each decile, can closely follow a large variety of curves. Each spline transformation of degree p with q knots fits a model with $p+q$ parameters. The total number of parameters should be much less than the number of observations. Usually in regression analyses, it is recommended that there be at least five or ten observations for each parameter in order to get stable results. For example, when spline transformations of degree three with nine knots are requested for six variables, the number of observations in the data set should be at least 5 or 10 times 72 (since $6 \times (3+9)$ is the total number of parameters). The overall model can also have a parameter for the intercept and one or more parameters for each nonspline variable in the model.

Increasing the number of knots gives the spline more freedom to bend and follow the data. Increasing the degree also gives the spline more freedom, but to a lesser extent. Specifying a large number of knots is much better than increasing the degree beyond three.

When you specify NKNOTS=q for a variable with n observations, then each of the $q+1$ segments of the spline contains $n/(q+1)$ observations on the average. When you specify KNOTS=number-list,

make sure that there is a reasonable number of observations in each interval.

The following statements find a cubic polynomial transformation of x and no transformation of y:

```
proc transreg;
   model identity(y)$=spline(x);
   output;
run;
```

The following statements find a cubic-spline transformation for x that consists of the weighted sum of a single constant, a single straight line, a quadratic curve for the portion of the variable less than 3.0, a different quadratic curve for the portion greater than 3.0 (since the 3.0 knot is repeated), and a different cubic curve for each of the intervals: (minimum to 1.5), (1.5 to 2.4), (2.4 to 3.0), (3.0 to 4.0), and (4.0 to maximum):

```
proc transreg;
   model identity(y)=spline(x / knots=1.5 2.4 3.0 3.0 4.0);
   output;
run;
```

The transformation is continuous everywhere, its first derivative is continuous everywhere, its second derivative is continuous everywhere except at 3.0, and its third derivative is continuous everywhere except at 1.5, 2.4, 3.0, and 4.0.

The following statements find a quadratic spline transformation that consists of a polynomial $\text{x\_t} = b_0 + b_1\text{x} + b_2\text{x}^2$ for the range ($\text{x} < 3.0$) and a completely different polynomial $\text{x\_t} = b_3 + b_4\text{x} + b_5\text{x}^2$ for the range ($\text{x} > 3.0$):

```
proc transreg;
   model identity(y)=spline(x / knots=3 3 3 degree=2);
   output;
run;
```

The two curves are not required to be continuous at 3.0.

The following statements categorize y into 10 intervals and find a step-function transformation:

```
proc transreg;
   model identity(y)=spline(x / degree=0 nknots=9);
   output;
run;
```

One aspect of this transformation family is unlike all other optimal transformation families. The initial scaling of the data does not fit the restrictions imposed by the transformation family. This is because the initial variable can be continuous, but a discrete step-function transformation is sought. Zero-degree spline variables are categorized before the first iteration.

The following statements find a continuous, piecewise linear transformation of x:

```
proc transreg;
   model identity(y)=spline(x / degree=1 nknots=8);
   output;
run;
```

SPLINE, BSPLINE, and PSPLINE Comparisons

SPLINE is a transformation. It takes a variable as input and produces a transformed variable as output. Internally, with SPLINE, a B-spline basis is used to find the transformation, which is a linear combination of the columns of the B-spline basis. However, with SPLINE, the basis is not made available in any output.

BSPLINE is an expansion. It takes a variable as input and produces more than one variable as output. The output variables are the same B-spline basis that is used internally by SPLINE.

PSPLINE is an expansion. It takes a variable as input and produces more than one variable as output. The difference between PSPLINE and BSPLINE is that PSPLINE produces a piecewise polynomial, whereas BSPLINE produces a B-spline. A matrix consisting of a piecewise polynomial basis and an intercept spans the same space as the B-spline matrix, but the basis vectors are quite different. The numbers in the piecewise polynomials can get quite large; the numbers in the B-spline basis range between 0 and 1. There are many more zeros in the B-spline basis.

Interchanging SPLINE, BSPLINE, and PSPLINE should have no effect on the fit of the overall model except for the fact that PSPLINE is much more prone to numerical problems. Similarly, interchanging a CLASS expansion and an OPSCORE transformation should have no effect on the fit of the overall model.

Hypothesis Tests

PROC TRANSREG has a set of options for testing hypotheses in models with a single dependent variable. The TEST *a-option* produces an ANOVA table. It tests the null hypothesis that the vector of coefficients for all of the transformations is zero. The SS2 *a-option* produces a regression table with Type II tests of the contribution of each transformation to the overall model. In some cases, exact tests are provided; in other cases, the tests are approximate, liberal, or conservative.

There are two reasons why it is typically not appropriate to test hypotheses by using the output from PROC TRANSREG as input to other procedures such as the REG procedure. First, PROC REG has no way of determining how many degrees of freedom were used for each transformation. Second, the Type II sums of squares for the tests of the individual regression coefficients are not correct for the transformation regression model because PROC REG, as it evaluates the effect of each variable, cannot change the transformations of the other variables. PROC TRANSREG uses the correct degrees of freedom and sums of squares.

In an ordinary univariate linear model, there is one parameter for each independent variable, including the intercept. In the transformation regression model, many of the "variables" are used internally in the bases for the transformations. Each basis column has one parameter or *scoring* coefficient, and each linearly independent column has one model degree of freedom associated with it. Coefficients applied to transformed variables, *model coefficients*, do not enter into the degrees-of-freedom calculations. They are byproducts of the standardizations and can be absorbed into the transformations by specifying the ADDITIVE *a-option*. The word *parameter* is reserved for model and scoring coefficients that have a degree of freedom associated with them.

For expansions, there is one model parameter for each variable created by the expansion (except for all missing CLASS columns and expansions that have an implicit intercept). Each IDENTITY variable has one model parameter. If there are m POINT variables, they expand to $m + 1$ variables and hence have $m + 1$ model parameters. For m EPOINT variables, there are $2m$ model parameters. For m QPOINT variables, there are $m(m + 3)/2$ model parameters. If a variable with m categories is designated CLASS, there are $m - 1$ parameters. For BSPLINE and PSPLINE variables of DEGREE=n with NKNOTS=k, there are $n + k$ parameters. Note that one of the $n + k + 1$ BSPLINE columns and one of the m CLASS(variable / ZERO=NONE) columns are not counted due to the implicit intercept.

There are scoring parameters for missing values in nonexcluded observations. Each ordinary missing value (.) has one scoring parameter. Each different special missing value (._ and .A through .Z) within each variable has one scoring parameter. Missing values specified in the UNTIE= and MONOTONE= options follow the rules for UNTIE and MONOTONE transformations, which are described later in this chapter.

For all nonoptimal transformations (LOG, LOGIT, ARSIN, POWER, EXP, RANK, BOXCOX), there is one parameter per variable in addition to any missing value scoring parameters.

For SPLINE, OPSCORE, and LINEAR transformations, the number of scoring parameters is the number of basis columns that are used internally to find the transformations minus 1 for the intercept. The number of scoring parameters for SPLINE variables is the same as the number of model parameters for BSPLINE and PSPLINE variables. If DEGREE=n and NKNOTS=k, there are $n + k$ scoring parameters. The number of scoring parameters for OPSCORE, SMOOTH, and SSPLINE variables is the same as the number of model parameters for CLASS variables. If there are m categories, there are $m - 1$ scoring parameters. There is one parameter for each LINEAR variable. For SPLINE, OPSCORE, LINEAR, MONOTONE, UNTIE, and MSPLINE transformations, missing value scoring parameters are computed as described previously with the nonoptimal transformations.

The number of scoring parameters for MONOTONE, UNTIE, and MSPLINE transformations is less precise than for SPLINE, OPSCORE, and LINEAR transformations. One way of handling a MONOTONE transformation is to treat it as if it were the same as an OPSCORE transformation. If there are m categories, there are $m - 1$ potential scoring parameters. However, there are typically fewer than $m - 1$ unique parameter estimates, since some of those $m - 1$ scoring parameter estimates might be tied during the optimal scaling to impose the order constraints. Imposing ties on the scoring parameter estimates is equivalent to fitting a model with fewer parameters. So there are two available scoring parameter counts: $m - 1$ and a smaller number that is determined during the analysis. Using $m - 1$ as the model degrees of freedom for MONOTONE variables (treating OPSCORE and MONOTONE transformations the same way) is *conservative*, since the MONOTONE scoring

parameter estimates are more restricted than the OPSCORE scoring parameter estimates. Using the smaller count (the number of scoring parameter estimates that are different, minus 1 for the intercept) in the model degrees of freedom is *liberal*, since the data and the model together are being used to determine the number of parameters. PROC TRANSREG reports tests that use both liberal and conservative degrees of freedom to provide lower and upper bounds on the "true" p-values.

For the UNTIE transformation, the conservative scoring parameter count is the number of distinct observations, whereas the liberal scoring parameter count is the number of scoring parameter estimates that are different, minus 1 for the intercept. Hence, when you specify UNTIE, conservative tests have zero error degrees of freedom unless there are replicated observations.

For MSPLINE variables of DEGREE=n and NKNOTS=k, the conservative scoring parameter count is $n + k$, whereas the liberal parameter count is the number of scoring parameter estimates that are different, minus 1 for the intercept. A liberal degrees of freedom of 1 does not necessarily imply a linear transformation. It implies only that n plus k minus the number of ties imposed equals 1. An example of a one-degree-of-freedom nonlinear transformation is a two-piece linear transformation in which the slope of one piece is 0.

The number of scoring parameters is determined during each iteration. After the last iteration, enough information is available for the TEST *a-option* to produce an ANOVA table that reports the overall fit of the model. If you specify the SS2 *a-option*, further iterations are necessary to test the contribution of each transformation to the overall model.

The liberal tests do not compensate for overparameterization. For example, requesting a spline transformation with k knots when a linear transformation will suffice results in "liberal" tests that are actually conservative because too many degrees of freedom are being used for the transformations. To avoid this problem, use as few knots as possible.

In ordinary multiple regression, an F test of the null hypothesis that the coefficient for variable x_j is zero can be constructed by comparing two linear models. One model is the full model with all parameters, and the other is a reduced model that has all parameters except the parameter for variable x_j. The difference between the model sum of squares for the full model and the model sum of squares for the reduced model is the Type II sum of squares for the test of the null hypothesis that the coefficient for variable x_j is 0. The numerator of the F test has one degree of freedom. The mean square error for the full model is the denominator of the F test of variable x_j. Note that the estimates of the coefficients for the two models are not usually the same. When variable x_j is removed, the coefficients for the other variables change to compensate for the removal of x_j. In a transformation regression model, the transformations of the other variables must be permitted to change and the numerator degrees of freedom are not always ones. It is not correct to simply let the model coefficients for the transformed variables change and apply the new model coefficients to the old transformations computed with the old scoring parameter estimates. In a transformation regression model, further iteration is needed to test each transformation, because all the scoring parameter estimates for other variables must be permitted to change to test the effect of variable x_j. This can be quite time-consuming for a large model if the SOLVE *a-option* cannot be used to solve directly for the transformations.

Output Data Set

The OUT= output data set can contain a great deal of information; however, in most cases, the output data set contains a small portion of the entire range of available information.

Output Data Set Examples

This section provides three brief examples, illustrating some typical OUT= output data sets. See the section "Output Data Set Contents" on page 7272 for a complete list of the contents of the OUT= data set.

The first example shows the output data set from a two-way ANOVA model. The following statements produce Figure 90.66:

```
title 'ANOVA Output Data Set Example';

data ReferenceCell;
   input y x1 $ x2 $;
   datalines;
11  a  a
12  a  a
10  a  a
 4  a  b
 5  a  b
 3  a  b
 5  b  a
 6  b  a
 4  b  a
 2  b  b
 3  b  b
 1  b  b
;

* Fit Reference Cell Two-Way ANOVA Model;
proc transreg data=ReferenceCell;
   model identity(y) = class(x1 | x2);
   output coefficients replace predicted residuals;
run;

* Print the Results;
proc print;
run;

proc contents position;
   ods select position;
run;
```

Figure 90.66 ANOVA Example Output Data Set Contents

ANOVA Output Data Set Example

Obs	_TYPE_	_NAME_	y	Py	Ry	Intercept	x1a	x2a	x1ax2a	x1	x2
1	SCORE	ROW1	11	11	0	1	1.0	1	1	a	a
2	SCORE	ROW2	12	11	1	1	1.0	1	1	a	a
3	SCORE	ROW3	10	11	-1	1	1.0	1	1	a	a
4	SCORE	ROW4	4	4	0	1	1.0	0	0	a	b
5	SCORE	ROW5	5	4	1	1	1.0	0	0	a	b
6	SCORE	ROW6	3	4	-1	1	1.0	0	0	a	b
7	SCORE	ROW7	5	5	0	1	0.0	1	0	b	a
8	SCORE	ROW8	6	5	1	1	0.0	1	0	b	a
9	SCORE	ROW9	4	5	-1	1	0.0	1	0	b	a
10	SCORE	ROW10	2	2	0	1	0.0	0	0	b	b
11	SCORE	ROW11	3	2	1	1	0.0	0	0	b	b
12	SCORE	ROW12	1	2	-1	1	0.0	0	0	b	b
13	M COEFFI	y	.	.	.	2	2.0	3	4		
14	MEAN	y	.	.	.	.	7.5	8	11		

ANOVA Output Data Set Example

The CONTENTS Procedure

Variables in Creation Order

#	Variable	Type	Len	Label
1	_TYPE_	Char	8	
2	_NAME_	Char	32	
3	y	Num	8	
4	Py	Num	8	y Predicted Values
5	Ry	Num	8	y Residuals
6	Intercept	Num	8	Intercept
7	x1a	Num	8	x1 a
8	x2a	Num	8	x2 a
9	x1ax2a	Num	8	x1 a * x2 a
10	x1	Char	32	
11	x2	Char	32	

The _TYPE_ variable indicates observation type: score, multiple regression coefficient (parameter estimates), and marginal means. The _NAME_ variable contains the default observation labels, "ROW1", "ROW2", and so on, and contains the dependent variable name (y) for the remaining observations. If you specify an ID statement, _NAME_ contains the values of the first ID variable for score observations. The y variable is the dependent variable, Py contains the predicted values, Ry contains the residuals, and the variables Intercept through x1ax2a contain the design matrix. The x1 and x2 variables are the original CLASS variables.

The next example shows the contents of the output data set from fitting a curve through a scatter plot. The following statements produce Figure 90.67:

```
title 'Output Data Set for Curve Fitting Example';

data a;
   do x = 1 to 100;
      y = log(x) + sin(x / 10) + normal(7);
      output;
   end;
run;

proc transreg;
   model identity(y) = spline(x / nknots=9);
   output predicted out=b;
run;

proc contents position;
   ods select position;
run;
```

Figure 90.67 Predicted Values Example Output Data Set Contents

Output Data Set for Curve Fitting Example

The CONTENTS Procedure

Variables in Creation Order

#	Variable	Type	Len	Label
1	_TYPE_	Char	8	
2	_NAME_	Char	32	
3	y	Num	8	
4	Ty	Num	8	y Transformation
5	Py	Num	8	y Predicted Values
6	Intercept	Num	8	Intercept
7	x	Num	8	
8	TIntercept	Num	8	Intercept Transformation
9	Tx	Num	8	x Transformation

The OUT= data set contains _TYPE_ and _NAME_ variables. Since no coefficients or coordinates are requested, all observations are _TYPE_='SCORE'. The y variable is the original dependent variable, Ty is the transformed dependent variable, Py contains the predicted values, x is the original independent variable, and Tx is the transformed independent variable. The data set also contains an Intercept and transformed intercept TIntercept variable. (In this case, the transformed intercept is the same as the intercept. However, if you specify the TSTANDARD= and ADDITIVE options, these are not always the same.)

The following example shows the results from specifying METHOD=MORALS when there is more than one dependent variable:

```
title 'METHOD=MORALS Output Data Set Example';

data x;
   input y1 y2 x1 $ x2 $;
   datalines;
11 1 a a
10 4 b a
 5 2 a b
 5 9 b b
 4 3 c c
 3 6 b a
 1 8 a b
;

* Fit Reference Cell Two-Way ANOVA Model;
proc transreg data=x noprint solve;
   model spline(y1 y2) = opscore(x1 x2 / name=(n1 n2));
   output coefficients predicted residuals;
   id x1 x2;
run;

* Print the Results;
proc print;
run;

proc contents position;
   ods select position;
run;
```

These statements produce Figure 90.68.

Figure 90.68 METHOD=MORALS Rolled Output Data Set

Obs	_DEPVAR_	_TYPE_	_NAME_	_DEPEND_	T_DEPEND_	P_DEPEND_	R_DEPEND_
1	Spline(y1)	SCORE	a	11	13.1600	11.1554	2.00464
2	Spline(y1)	SCORE	b	10	6.1931	6.8835	-0.69041
3	Spline(y1)	SCORE	a	5	2.4467	4.7140	-2.26724
4	Spline(y1)	SCORE	b	5	2.4467	0.4421	2.00464
5	Spline(y1)	SCORE	c	4	4.2076	4.2076	0.00000
6	Spline(y1)	SCORE	b	3	5.5693	6.8835	-1.31422
7	Spline(y1)	SCORE	a	1	4.9766	4.7140	0.26261
8	Spline(y1)	M COEFFI	y1	.	.	.	.
9	Spline(y2)	SCORE	a	1	-0.5303	-0.5199	-0.01043
10	Spline(y2)	SCORE	b	4	5.5487	4.5689	0.97988
11	Spline(y2)	SCORE	a	2	3.8940	4.5575	-0.66347
12	Spline(y2)	SCORE	b	9	9.6358	9.6462	-0.01043
13	Spline(y2)	SCORE	c	3	5.6210	5.6210	0.00000
14	Spline(y2)	SCORE	b	6	3.5994	4.5689	-0.96945
15	Spline(y2)	SCORE	a	8	5.2314	4.5575	0.67390
16	Spline(y2)	M COEFFI	y2	.	.	.	.

Obs	Intercept	n1	n2	TIntercept	Tn1	Tn2	x1	x2
1	1	0	0	1.0000	0.06711	-0.09384	a	a
2	1	1	0	1.0000	1.51978	-0.09384	b	a
3	1	0	1	1.0000	0.06711	1.32038	a	b
4	1	1	1	1.0000	1.51978	1.32038	b	b
5	1	2	2	1.0000	0.23932	1.32038	c	c
6	1	1	0	1.0000	1.51978	-0.09384	b	a
7	1	0	1	1.0000	0.06711	1.32038	a	b
8	.	.	.	10.9253	-2.94071	-4.55475	y1	y1
9	1	0	0	1.0000	0.03739	-0.09384	a	a
10	1	1	0	1.0000	1.51395	-0.09384	b	a
11	1	0	1	1.0000	0.03739	1.32038	a	b
12	1	1	1	1.0000	1.51395	1.32038	b	b
13	1	2	2	1.0000	0.34598	1.32038	c	c
14	1	1	0	1.0000	1.51395	-0.09384	b	a
15	1	0	1	1.0000	0.03739	1.32038	a	b
16	.	.	.	-0.3119	3.44636	3.59024	y2	y2

Figure 90.68 *continued*

METHOD=MORALS Output Data Set Example

The CONTENTS Procedure

Variables in Creation Order

#	Variable	Type	Len	Label
1	_DEPVAR_	Char	42	Dependent Variable Transformation(Name)
2	_TYPE_	Char	8	
3	_NAME_	Char	32	
4	_DEPEND_	Num	8	Dependent Variable
5	T_DEPEND_	Num	8	Dependent Variable Transformation
6	P_DEPEND_	Num	8	Dependent Variable Predicted Values
7	R_DEPEND_	Num	8	Dependent Variable Residuals
8	Intercept	Num	8	Intercept
9	n1	Num	8	
10	n2	Num	8	
11	TIntercept	Num	8	Intercept Transformation
12	Tn1	Num	8	n1 Transformation
13	Tn2	Num	8	n2 Transformation
14	x1	Char	32	
15	x2	Char	32	

If you specify METHOD=MORALS with multiple dependent variables, PROC TRANSREG performs separate univariate analyses and stacks the results in the OUT= data set. For this example, the results of the first analysis are in the partition designated by _DEPVAR_='Spline(y1)' and the results of the second analysis are in the partition designated by _DEPVAR_='Spline(y2)', which are the transformation and dependent variable names. Each partition has _TYPE_='SCORE' observations for the variables and a _TYPE_='M COEFFI' observation for the coefficients. In this example, an ID variable is specified, so the _NAME_ variable contains the formatted values of the first ID variable. Since both dependent variables have to go into the same column, the dependent variable is given a new name, _DEPEND_. The dependent variable transformation is named T_DEPEND_, the predicted values variable is named P_DEPEND_, and the residuals variable is named R_DEPEND_.

The independent variables are character OPSCORE variables. By default, PROC TRANSREG replaces character OPSCORE variables with category numbers and discards the original character variables. To avoid this, the input variables are renamed from x1 and x2 to n1 and n2 and the original x1 and x2 are added to the data set as ID variables. The n1 and n2 variables contain the initial values for the OPSCORE transformations, and the Tn1 and Tn2 variables contain optimal scores. The data set also contains an Intercept and transformed intercept TIntercept variable. The regression coefficients are in the transformation columns, which also contain the variables to which they apply.

Output Data Set Contents

Table 90.7 summarizes the various matrices that can result from PROC TRANSREG processing and that appear in the OUT= data set. The exact contents of an OUT= data set depends on many options.

Table 90.7 PROC TRANSREG OUT= Data Set Contents

TYPE	Contents	Options, Default Prefix
SCORE	dependent variables	DREPLACE not specified
SCORE	independent variables	IREPLACE not specified
SCORE	transformed dependent variables	default, TDPREFIX=T
SCORE	transformed independent variables	default, TIPREFIX=T
SCORE	predicted values	PREDICTED, PPREFIX=P
SCORE	residuals	RESIDUALS, RDPREFIX=R
SCORE	leverage	LEVERAGE, LEVERAGE=Leverage
SCORE	lower individual confidence limits	CLI, LILPREFIX=LIL, CILPREFIX=CIL
SCORE	upper individual confidence limits	CLI, LIUPREFIX=LIU, CIUPREFIX=CIU
SCORE	lower mean confidence limits	CLM, LMLPREFIX=LML, CMLPREFIX=CML
SCORE	upper mean confidence limits	CLM, LMUPREFIX=LMU, CMUPREFIX=CMU
SCORE	dependent canonical variables	CANONICAL, CDPREFIX=Cand
SCORE	independent canonical variables	CANONICAL, CIPREFIX=Cani
SCORE	redundancy variables	REDUNDANCY, RPREFIX=Red
SCORE	ID, CLASS, BSPLINE variables	ID, CLASS, BSPLINE,
SCORE	independent variables approximations	IAPPROXIMATIONS, AIPREFIX=A
M COEFFI	multiple regression coefficients	COEFFICIENTS, MRC
C COEFFI	canonical coefficients	COEFFICIENTS, CCC
MEAN	marginal means	COEFFICIENTS, MEANS
M REDUND	multiple redundancy coefficients	MREDUNDANCY
R REDUND	multiple redundancy coefficients	MREDUNDANCY
M POINT	point coordinates	COORDINATES or MPC, POINT
M EPOINT	elliptical point coordinates	COORDINATES or MEC, EPOINT
M QPOINT	quadratic point coordinates	COORDINATES or MQC, QPOINT
C POINT	canonical point coordinates	COORDINATES or CPC, POINT
C EPOINT	canonical elliptical point coordinates	COORDINATES or CEC, EPOINT
C QPOINT	canonical quadratic point coordinates	COORDINATES or CQC, QPOINT

The independent and dependent variables are created from the original input data. Several potential differences exist between these variables and the actual input data. An intercept variable can be added, new variables can be added for POINT, EPOINT, QPOINT, CLASS, IDENTITY, PSPLINE, and BSPLINE variables, and category numbers are substituted for character OPSCORE variables. These matrices are not always what is input to the first iteration. After the expanded data set is stored for inclusion in the output data set, several things happen to the data before they are input to the first iteration: column means are substituted for missing values; zero-degree SPLINE and MSPLINE variables are transformed so that the iterative algorithms get step-function data as input, which conform to the zero-degree transformation family restrictions; and the nonoptimal transformations are performed.

Details for the UNIVARIATE Method

When you specify METHOD=UNIVARIATE (in the MODEL or PROC TRANSREG statement), PROC TRANSREG can perform several analyses, one for each dependent variable. While each dependent variable can be transformed, their independent variables are not transformed. The OUT= data set optionally contains all of the _TYPE_='SCORE' observations, optionally followed by coefficients or coordinates.

Details for the MORALS Method

When you specify METHOD=MORALS (in the MODEL or PROC TRANSREG statement), successive analyses are performed, one for each dependent variable. Each analysis transforms one dependent variable and the entire set of the independent variables. All information for the first dependent variable (scores then, optionally, coefficients) appears first. Then all information for the second dependent variable (scores then, optionally, coefficients) appears next. This arrangement is repeated for all dependent variables.

Details for the CANALS and REDUNDANCY Methods

For METHOD=CANALS and METHOD=REDUNDANCY (specified in either the MODEL or PROC TRANSREG statement), one analysis is performed that simultaneously transforms all dependent and independent variables. The OUT= data set optionally contains all of the _TYPE_='SCORE' observations, optionally followed by coefficients or coordinates.

Variable Names

As shown in the preceding examples, some variables in the output data set directly correspond to input variables, and some are created. All original optimal and nonoptimal transformation variable names are unchanged.

The names of the POINT, QPOINT, and EPOINT expansion variables are also left unchanged, but new variables are created. When independent POINT variables are present, the sum-of-squares variable _ISSQ_ is added to the output data set. For each EPOINT and QPOINT variable, a new squared variable is created by appending "_2". For example, Dim1 and Dim2 are expanded into Dim1, Dim2, Dim1_2, and Dim2_2. In addition, for each pair of QPOINT variables, a new crossproduct variable is created by combining the two names—for example, Dim1Dim2.

The names of the CLASS variables are constructed from original variable names and levels. Lengths are controlled by the CPREFIX= *a-option*. For example, when x1 and x2 both have values of 'a' and 'b', CLASS(x1 | x2 / ZERO=NONE) creates x1 main-effect variable names x1a x1b, x2 main-effect variable names x2a x2b, and interaction variable names x1ax2a x1ax2b x1bx2a x1bx2b.

PROC TRANSREG then uses these variable names when creating the transformed, predicted, and residual variable names by affixing the relevant prefix and dropping extra characters if necessary.

METHOD=MORALS Variable Names

When you specify METHOD=MORALS and only one dependent variable is present, the output data set is structured exactly as if METHOD=REDUNDANCY (see the section "Details for the CANALS and REDUNDANCY Methods" on page 7274). When more than one dependent variable is present, the dependent variables are output in the variable _DEPEND_, transformed dependent variables are output in the variable T_DEPEND_, predicted values are output in the variable P_DEPEND_, and residuals are output in the variable R_DEPEND_. You can partition the data set into BY groups, one per dependent variable, by referring to the character variable _DEPVAR_, which contains the original dependent variable names and transformations.

Duplicate Variable Names

When the same name is generated from multiple variables in the OUT= data set, new names are created by appending '2', '3', or '4', and so on, until a unique name is created. For 32-character names, the last character is replaced with a numeric suffix until a unique name is created. For example, if there are two output variables that otherwise would be named x, then x and x2 are created instead. If there are two output variables that otherwise would be named ThisIsAThirtyTwoCharacterVarName, then ThisIsAThirtyTwoCharacterVarName and ThisIsAThirtyTwoCharacterVarNam2 are created instead.

OUTTEST= Output Data Set

The OUTTEST= data set contains hypothesis test results. The OUTTEST= data set always contains ANOVA results. When you specify the SS2 *a-option*, regression tables are also output. When you specify the UTILITIES *a-option*, conjoint analysis part-worth utilities are also output. The OUTTEST= data set has the following variables:

DEPVAR is a 42-character variable that contains the dependent variable transformation and name.

TYPE is an 8-character variable that contains the table type. The first character is "U" for univariate or "M" for multivariate. The second character is blank. The third character is "A" for ANOVA, "2" for Type II sum of squares, or "U" for UTILITIES. The fourth character is blank. The fifth character is "L" for liberal tests, "C" for conservative tests, or "U" for the usual tests.

Title is an 80-character variable that contains the table title.

Variable is a 42-character variable that contains the independent variable transformations and names for regression tables and blanks for ANOVA tables.

Coefficient contains the multiple regression coefficients for regression tables and underscore special missing values for ANOVA tables.

Statistic is a 24-character variable that contains the names for statistics in other variables, such as Value.

Value contains multivariate test statistics and all other information that does not fit in one of the other columns including R square, dependent mean, adjusted R

	square, and coefficient of variation. Whenever Value is not an underscore special missing value, the Statistic variable describes the contents of the Value variable.
NumDF	contains numerator degrees of freedom for F tests.
DenDF	contains denominator degrees of freedom for F tests.
SSq	contains sums of squares.
MeanSquare	contains mean squares.
F	contains F statistics.
NumericP	contains the p-value for the F statistic, stored in a numeric variable.
P	is a 9-character variable that contains the formatted p-value for the F statistic, including the appropriate ~, <=, >=, or blank symbols.
LowerLimit	contains lower confidence limits on the parameter estimates.
UpperLimit	contains upper confidence limits on the parameter estimates.
StdError	contains standard errors. For SS2 and UTILITIES tables, standard errors are output for each coefficient with one degree of freedom.
Importance	contains the relative importance of each factor for UTILITIES tables.
Label	is a 256-character variable that contains variable labels.

There are several possible tables in the OUTTEST= data set corresponding to combinations of univariate and multivariate tests; ANOVA and regression results; and liberal, conservative, and the usual tests. Each table is composed of only a subset of the variables. Numeric variables contain underscore special missing values when they are not a column in a table. Ordinary missing values (.) appear in variables that are part of a table when a nonmissing value cannot be produced. For example, the F is missing for a test with zero degrees of freedom.

Computational Resources

This section provides information about the computational resources required to use PROC TRANSREG.

Let

n = number of observations
q = number of expanded independent variables
r = number of expanded dependent variables
k = maximum spline degree
p = maximum number of knots

More than $56(q+r)$ plus the maximum of the data matrix size, the optimal scaling work space, and the covariance matrix size bytes of array space are required. The data matrix size is $8n(q+r)$

bytes. The optimal scaling work space requires less than $8(6n + (p + k + 2)(p + k + 11))$ bytes. The covariance matrix size is $4(q + r)(q + r + 1)$ bytes.

PROC TRANSREG tries to store the original and transformed data in memory. If there is not enough memory, a utility data set is used, potentially resulting in a large increase in execution time. The amount of memory for the preceding data formulas is an underestimate of the amount of memory needed to handle most problems. These formulas give the absolute minimum amount of memory required. If a utility data set is used, and if memory can be used with perfect efficiency, then roughly the amount of memory stated previously is needed. In reality, most problems require at least two or three times the minimum.

PROC TRANSREG sorts the data once. The sort time is roughly proportional to $(q + r)n^{3/2}$.

One regression analysis per iteration is required to compute model parameters (or two canonical correlation analyses per iteration for METHOD=CANALS). The time required to accumulate the crossproducts matrix is roughly proportional to $n(q + r)^2$. The time required to compute the regression coefficients is roughly proportional to q^3.

Each optimal scaling is a multiple regression problem, although some transformations are handled with faster special-case algorithms. The number of regressors for the optimal scaling problems depends on the original values of the variable and the type of transformation. For each monotone spline transformation, an unknown number of multiple regressions is required to find a set of coefficients that satisfies the constraints. The B-spline basis is generated twice for each SPLINE and MSPLINE transformation for each iteration. The time required to generate the B-spline basis is roughly proportional to nk^2.

Unbalanced ANOVA without CLASS Variables

This section illustrates that an analysis of variance model can be formulated as a simple regression model with optimal scoring. The purpose of the example is to explain one aspect of how PROC TRANSREG works, not to propose an alternative way of performing an analysis of variance.

Finding the overall fit of a large, unbalanced analysis of variance model can be handled as an optimal scoring problem without creating large, sparse design matrices. For example, consider an unbalanced full main-effects and interactions ANOVA model with six factors. Assume that a SAS data set is created with factor-level indicator variables `c1` through `c6` and dependent variable `y`. If each factor level consists of nonblank single characters, you can create a cell indicator in a DATA step with the statement as follows:

```
x=compress(c1||c2||c3||c4||c5||c6);
```

The following statements optimally score `x` (by using the OPSCORE transformation) and do not transform `y`:

```
proc transreg;
   model identity(y)=opscore(x);
   output;
run;
```

The final R square reported is the R square for the full analysis of variance model. This R square is the same R square that would be reported by both of the following PROC GLM steps:

```
proc glm;
   class x;
   model y=x;
run;

proc glm;
   class c1-c6;
   model y=c1|c2|c3|c4|c5|c6;
run;
```

PROC TRANSREG optimally scores the classes of x, within the space of a single variable with values linearly related to the cell means, so the full ANOVA problem is reduced to a simple regression problem with an optimal independent variable. PROC TRANSREG requires only one iteration to find the optimal scoring of x but, by default, performs a second iteration, which reports no data changes.

Hypothesis Tests for Simple Univariate Models

If the dependent variable has one parameter (IDENTITY, LINEAR with no missing values, and so on) and if there are no monotonicity constraints, PROC TRANSREG fits univariate models, which can also be fit with a DATA step and PROC REG. This is illustrated with the following artificial data set:

```
data htex;
   do i = 0.5 to 10 by 0.5;
      x1 = log(i);
      x2 = sqrt(i) + sin(i);
      x3 = 0.05 * i * i + cos(i);
      y  = x1 - x2 + x3 + 3 * normal(7);
      x1 = x1 + normal(7);
      x2 = x2 + normal(7);
      x3 = x3 + normal(7);
      output;
   end;
run;
```

Both PROC TRANSREG and PROC REG are run to fit the same polynomial regression model as follows:

```
proc transreg data=htex ss2 short;
   title 'Fit a Polynomial Regression Model with PROC TRANSREG';
   model identity(y) = spline(x1);
run;
```

```
data htex2;
   set htex;
   x1_1 = x1;
   x1_2 = x1 * x1;
   x1_3 = x1 * x1 * x1;
run;

proc reg;
   title 'Fit a Polynomial Regression Model with PROC REG';
   model y = x1_1 - x1_3;
run; quit;
```

The ANOVA and regression tables from PROC TRANSREG are displayed in Figure 90.69. The ANOVA and regression tables from PROC REG are displayed in Figure 90.70. The SHORT *a-option* is specified with PROC TRANSREG to suppress the iteration history.

Figure 90.69 ANOVA and Regression Output from PROC TRANSREG

```
              Fit a Polynomial Regression Model with PROC TRANSREG

                            The TRANSREG Procedure

                        Dependent Variable Identity(y)


                  Number of Observations Read            20
                  Number of Observations Used            20

   Identity(y)
   Algorithm converged.

            The TRANSREG Procedure Hypothesis Tests for Identity(y)


         Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                  Sum of        Mean
      Source               DF     Squares      Square    F Value    Pr > F

      Model                 3      5.8365     1.94550       0.14    0.9329
      Error                16    218.3073    13.64421
      Corrected Total      19    224.1438

                Root MSE           3.69381    R-Square    0.0260
                Dependent Mean     0.85490    Adj R-Sq   -0.1566
                Coeff Var        432.07258
```

Figure 90.69 *continued*

Univariate Regression Table Based on the Usual Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Pr > F
Intercept	1	1.4612767	18.8971	18.8971	1.38	0.2565
Spline(x1)	3	−0.3924013	5.8365	1.9455	0.14	0.9329

Figure 90.70 ANOVA and Regression Output from PROC REG

Fit a Polynomial Regression Model with PROC REG

The REG Procedure
Model: MODEL1
Dependent Variable: y

Number of Observations Read	20
Number of Observations Used	20

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	3	5.83651	1.94550	0.14	0.9329
Error	16	218.30729	13.64421		
Corrected Total	19	224.14380			

Root MSE	3.69381	R-Square	0.0260
Dependent Mean	0.85490	Adj R-Sq	−0.1566
Coeff Var	432.07258		

Parameter Estimates

Variable	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	1	1.22083	1.47163	0.83	0.4190
x1_1	1	0.79743	1.75129	0.46	0.6550
x1_2	1	−0.49381	1.50449	−0.33	0.7470
x1_3	1	0.04422	0.32956	0.13	0.8949

The PROC TRANSREG regression table differs in several important ways from the parameter estimate table produced by PROC REG. The REG procedure displays standard errors and t statistics. PROC TRANSREG displays Type II sums of squares, mean squares, and F statistics. The difference is because the numerator degrees of freedom are not always 1, so t tests are not uniformly appropriate. When the degrees of freedom for variable x_j is 1, the following relationships hold between the standard errors (s_{β_j}) and the Type II sums of squares (SS_j):

$$s_{\beta_j} = (\hat{\beta}_j^2 / F_j)^{1/2}$$

and

$$SS_j = \hat{\beta}_j^2 \times MSE / s_{\beta_j}^2$$

PROC TRANSREG does not provide tests of the individual terms that go into the transformation. (However, it could if BSPLINE or PSPLINE had been specified instead of SPLINE.) The test of **spline(x1)** is the same as the test of the overall model. The intercepts are different due to the different numbers of variables and their standardizations.

In the next example, both x1 and x2 are transformed in the first PROC TRANSREG step, and PROC TRANSREG is used instead of a DATA step to create the polynomials for PROC REG. Both PROC TRANSREG and PROC REG fit the same polynomial regression model. The following statements run PROC TRANSREG and PROC REG and produce Figure 90.71 and Figure 90.72:

```
title 'Two-Variable Polynomial Regression';

proc transreg data=htex ss2 solve;
   model identity(y) = spline(x1 x2);
run;

proc transreg noprint data=htex maxiter=0;
   /* Use PROC TRANSREG to prepare input to PROC REG */
   model identity(y) = pspline(x1 x2);
   output out=htex2;
run;

proc reg data=htex2;
   model y = x1_1-x1_3 x2_1-x2_3;
   test x1_1, x1_2, x1_3;
   test x2_1, x2_2, x2_3;
run; quit;
```

Figure 90.71 Two-Variable Polynomial Regression Output from PROC TRANSREG

```
                    Two-Variable Polynomial Regression

                         The TRANSREG Procedure

                       Dependent Variable Identity(y)


                  Number of Observations Read            20
                  Number of Observations Used            20

          TRANSREG MORALS Algorithm Iteration History for Identity(y)

   Iteration     Average     Maximum                 Criterion
      Number      Change      Change     R-Square       Change     Note
   ---------------------------------------------------------------------
           0     0.69502     4.73421      0.08252
           1     0.00000     0.00000      0.17287      0.09035    Converged
```

Figure 90.71 *continued*

```
Algorithm converged.

            Hypothesis Test Iterations Excluding Spline(x1)
       TRANSREG MORALS Algorithm Iteration History for Identity(y)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        0    0.03575    0.32390     0.15097
        1    0.00000    0.00000     0.15249      0.00152    Converged

Algorithm converged.

            Hypothesis Test Iterations Excluding Spline(x2)
       TRANSREG MORALS Algorithm Iteration History for Identity(y)

Iteration    Average    Maximum                Criterion
   Number     Change     Change    R-Square       Change    Note
-------------------------------------------------------------------------
        0    0.45381    1.43736     0.00717
        1    0.00000    0.00000     0.02604      0.01886    Converged

Algorithm converged.

         The TRANSREG Procedure Hypothesis Tests for Identity(y)


       Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                Sum of        Mean
   Source                DF     Squares      Square    F Value    Pr > F

   Model                  6     38.7478     6.45796       0.45    0.8306
   Error                 13    185.3960    14.26123
   Corrected Total       19    224.1438

             Root MSE           3.77640    R-Square     0.1729
             Dependent Mean     0.85490    Adj R-Sq    -0.2089
             Coeff Var        441.73431

    Univariate Regression Table Based on the Usual Degrees of Freedom

                                    Type II
                                     Sum of       Mean
Variable         DF    Coefficient    Squares     Square    F Value    Pr > F

Intercept         1      3.5437125    35.2282    35.2282       2.47    0.1400
Spline(x1)        3      0.3644562     4.5682     1.5227       0.11    0.9546
Spline(x2)        3     -1.3551738    32.9112    10.9704       0.77    0.5315
```

There are three iteration histories: one for the overall model and two for the two independent variables. The first PROC TRANSREG iteration history shows the R square of 0.17287 for the

fit of the overall model. The second is for the following model:

```
model identity(y) = spline(x2);
```

This model excludes **spline(x1)**. The third iteration history is for the following model:

```
model identity(y) = spline(x1);
```

This model excludes **spline(x2)**. The difference between the first and second R square times the total sum of squares is the model sum of squares for **spline(x1)**:

$$(0.17287 - 0.15249) \times 224.143800 = 4.568165$$

The difference between the first and third R square times the total sum of squares is the model sum of squares for **spline(x2)**:

$$(0.17287 - 0.02604) \times 224.143800 = 32.911247$$

Figure 90.72 displays the PROC REG results. The TEST statement in PROC REG tests the null hypothesis that the vector of parameters for x1_1 x1_2 x1_3 is zero. This is the same test as the **spline(x1)** test used by PROC TRANSREG. Similarly, the PROC REG test that the vector of parameters for x2_1 x2_2 x2_3 is zero is the same as the PROC TRANSREG SPLINE(x2) test. So for models with no monotonicity constraints and no dependent variable transformations, PROC TRANSREG provides little more than a different packaging of standard least-squares methodology.

Figure 90.72 Two-Variable Polynomial Regression Output from PROC REG

Two-Variable Polynomial Regression

The REG Procedure
Model: MODEL1
Dependent Variable: y

Number of Observations Read	20
Number of Observations Used	20

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	6	38.74775	6.45796	0.45	0.8306
Error	13	185.39605	14.26123		
Corrected Total	19	224.14380			

Root MSE	3.77640	R-Square	0.1729
Dependent Mean	0.85490	Adj R-Sq	-0.2089
Coeff Var	441.73431		

Figure 90.72 *continued*

Parameter Estimates

Variable	Label	DF	Parameter Estimate	Standard Error	t Value	Pr > |t|
Intercept	Intercept	1	10.77824	7.55244	1.43	0.1771
x1_1	x1 1	1	0.40112	1.81024	0.22	0.8281
x1_2	x1 2	1	0.25652	1.66023	0.15	0.8796
x1_3	x1 3	1	-0.11639	0.36775	-0.32	0.7567
x2_1	x2 1	1	-14.07054	12.50521	-1.13	0.2809
x2_2	x2 2	1	5.95610	5.97952	1.00	0.3374
x2_3	x2 3	1	-0.80608	0.87291	-0.92	0.3726

Two-Variable Polynomial Regression

The REG Procedure
Model: MODEL1

Test 1 Results for Dependent Variable y

Source	DF	Mean Square	F Value	Pr > F
Numerator	3	1.52272	0.11	0.9546
Denominator	13	14.26123		

Two-Variable Polynomial Regression

The REG Procedure
Model: MODEL1

Test 2 Results for Dependent Variable y

Source	DF	Mean Square	F Value	Pr > F
Numerator	3	10.97042	0.77	0.5315
Denominator	13	14.26123		

Hypothesis Tests with Monotonicity Constraints

Now consider a model with monotonicity constraints. This model has no counterpart in PROC REG. The following statements fit a monotone-spline model and produce Figure 90.73:

```
title 'Monotone Splines';

proc transreg data=htex ss2 short;
   model identity(y) = mspline(x1-x3 / nknots=3);
run;
```

The SHORT *a-option* is specified to suppress the iteration histories. Two ANOVA tables are displayed—one by using liberal degrees of freedom and one by using conservative degrees of freedom. All sums of squares and the R squares are the same for both tables. What differs are the degrees of freedom and statistics that use degrees of freedom. The liberal test has 8 model degrees of freedom and 11 error degrees of freedom, whereas the conservative test has 15 model degrees of freedom and only 4 error degrees of freedom. The "true" p-value is between 0.8462 and 0.9997, so clearly you would fail to reject the null hypothesis. Unfortunately, results are not always this clear. (See Figure 90.73.)

Figure 90.73 Monotone Spline Transformations

Monotone Splines

The TRANSREG Procedure

Dependent Variable Identity(y)

Number of Observations Read	20
Number of Observations Used	20

Identity(y)
Algorithm converged.

The TRANSREG Procedure Hypothesis Tests for Identity(y)

Univariate ANOVA Table Based on Liberal Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Liberal p
Model	8	58.0534	7.25667	0.48	>= 0.8462
Error	11	166.0904	15.09913		
Corrected Total	19	224.1438			

Root MSE	3.88576	R-Square	0.2590
Dependent Mean	0.85490	Adj R-Sq	-0.2799
Coeff Var	454.52581		

Univariate ANOVA Table Based on Conservative Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Conservative p
Model	15	58.0534	3.87022	0.09	<= 0.9997
Error	4	166.0904	41.52261		
Corrected Total	19	224.1438			

Root MSE	6.44380	R-Square	0.2590
Dependent Mean	0.85490	Adj R-Sq	-2.5197
Coeff Var	753.74578		

Figure 90.73 *continued*

Univariate Regression Table Based on Liberal Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Liberal p
Intercept	1	4.8687676	54.7372	54.7372	3.63	>= 0.0834
Mspline(x1)	2	−0.6886834	12.1943	6.0972	0.40	>= 0.6773
Mspline(x2)	3	−1.8237319	46.3155	15.4385	1.02	>= 0.4199
Mspline(x3)	3	0.8646155	24.6840	8.2280	0.54	>= 0.6616

Univariate Regression Table Based on Conservative Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Conservative p
Intercept	1	4.8687676	54.7372	54.7372	1.32	<= 0.3149
Mspline(x1)	5	−0.6886834	12.1943	2.4389	0.06	<= 0.9959
Mspline(x2)	5	−1.8237319	46.3155	9.2631	0.22	<= 0.9344
Mspline(x3)	5	0.8646155	24.6840	4.9368	0.12	<= 0.9809

Hypothesis Tests with Dependent Variable Transformations

PROC TRANSREG can also provide approximate tests of hypotheses when the dependent variable is transformed, but the output is more complicated. When a dependent variable has more than one degree of freedom, the problem becomes multivariate. Hypothesis tests are performed in the context of a multivariate linear model with the number of dependent variables equal to the number of scoring parameters for the dependent variable transformation. The transformation regression model with a dependent variable transformation differs from the usual multivariate linear model in two important ways. First, the usual assumption of multivariate normality is always violated. This fact is simply ignored. This is one reason why all hypothesis tests in the presence of a dependent variable transformation should be considered approximate at best. Multivariate normality is assumed even though it is known that the assumption is violated.

The second difference concerns the usual multivariate test statistics: Pillai's trace, Wilks' lambda, Hotelling-Lawley trace, and Roy's greatest root. The first three statistics are defined in terms of all the squared canonical correlations. Here, there is only one linear combination (the transformation), and hence only one squared canonical correlation of interest, which is equal to the R square. It might seem that Roy's greatest root, which uses only the largest squared canonical correlation, is the only statistic of interest. Unfortunately, Roy's greatest root is very liberal and provides only a lower bound on the *p*-value. Approximate upper bounds are provided by adjusting the other three statistics for the one linear combination case. Wilks' lambda, Pillai's trace, and Hotelling-Lawley trace are a conservative adjustment of the usual statistics.

These statistics are normally defined in terms of the squared canonical correlations, which are the eigenvalues of the matrix $\mathbf{H}(\mathbf{H}+\mathbf{E})^{-1}$, where $\mathbf{H}$ is the hypothesis sum-of-squares matrix and

E is the error sum-of-squares matrix. Here the R square is used for the first eigenvalue, and all other eigenvalues are set to 0 since only one linear combination is used. Degrees of freedom are computed assuming that all linear combinations contribute to the lambda and trace statistics, so the *F* tests for those statistics are conservative. The *p*-values for the liberal and conservative statistics provide approximate lower and upper bounds on *p*. In practice, the adjusted Pillai's trace is very conservative—perhaps too conservative to be useful. Wilks' lambda is less conservative, and the Hotelling-Lawley trace seems to be the least conservative. The conservative statistics and the liberal Roy's greatest root provide a bound on the true *p*-value. Unfortunately, they sometimes report a bound of 0.0001 and 1.0000.

The following example has a dependent variable transformation and produces Figure 90.74:

```
title 'Transform Dependent and Independent Variables';

proc transreg data=htex ss2 solve short;
   model spline(y) = spline(x1-x3);
run;
```

The univariate results match Roy's greatest root results. Clearly, the proper action is to fail to reject the null hypothesis. However, as stated previously, results are not always this clear.

Figure 90.74 Transform Dependent and Independent Variables

```
                Transform Dependent and Independent Variables

                           The TRANSREG Procedure

                         Dependent Variable Spline(y)


                  Number of Observations Read            20
                  Number of Observations Used            20

   Spline(y)
   Algorithm converged.

            The TRANSREG Procedure Hypothesis Tests for Spline(y)


        Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                  Sum of          Mean
    Source                 DF     Squares        Square    F Value    Liberal p

    Model                   9    110.8822    12.32025         1.09    >= 0.4452
    Error                  10    113.2616    11.32616
    Corrected Total        19    224.1438

    The above statistics are not adjusted for the fact that the dependent
    variable was transformed and so are generally liberal.

                Root MSE           3.36544    R-Square    0.4947
                Dependent Mean     0.85490    Adj R-Sq    0.0399
                Coeff Var        393.66234
```

Figure 90.74 *continued*

Adjusted Multivariate ANOVA Table Based on the Usual Degrees of Freedom

Dependent Variable Scoring Parameters=3 S=3 M=2.5 N=3

Statistic	Value	F Value	Num DF	Den DF	p
Wilks' Lambda	0.505308	0.23	27	24.006	<= 0.9998
Pillai's Trace	0.494692	0.22	27	30	<= 0.9999
Hotelling-Lawley Trace	0.978992	0.26	27	11.589	<= 0.9980
Roy's Greatest Root	0.978992	1.09	9	10	>= 0.4452

The Wilks' Lambda, Pillai's Trace, and Hotelling-Lawley Trace statistics are a conservative adjustment of the normal statistics. Roy's Greatest Root is liberal. These statistics are normally defined in terms of the squared canonical correlations which are the eigenvalues of the matrix H*inv(H+E). Here the R-Square is used for the first eigenvalue and all other eigenvalues are set to zero since only one linear combination is used. Degrees of freedom are computed assuming all linear combinations contribute to the Lambda and Trace statistics, so the F tests for those statistics are conservative. The p values for the liberal and conservative statistics provide approximate lower and upper bounds on p. A liberal test statistic with conservative degrees of freedom and a conservative test statistic with liberal degrees of freedom yield at best an approximate p value, which is indicated by a "~" before the p value.

Univariate Regression Table Based on the Usual Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Liberal p
Intercept	1	6.9089087	117.452	117.452	10.37	>= 0.0092
Spline(x1)	3	-1.0832321	32.493	10.831	0.96	>= 0.4504
Spline(x2)	3	-2.1539191	45.251	15.084	1.33	>= 0.3184
Spline(x3)	3	0.4779207	10.139	3.380	0.30	>= 0.8259

The above statistics are not adjusted for the fact that the dependent variable was transformed and so are generally liberal.

Figure 90.74 *continued*

Adjusted Multivariate Regression Table Based on the Usual Degrees of Freedom

Variable	Coefficient	Statistic	Value	F Value	Num DF	Den DF	p
Intercept	6.9089087	Wilks' Lambda	0.49092	2.77	3	8	0.1112
		Pillai's Trace	0.50908	2.77	3	8	0.1112
		Hotelling-Lawley Trace	1.036993	2.77	3	8	0.1112
		Roy's Greatest Root	1.036993	2.77	3	8	0.1112
Spline(x1)	-1.0832321	Wilks' Lambda	0.777072	0.24	9	19.621	<= 0.9840
		Pillai's Trace	0.222928	0.27	9	30	<= 0.9787
		Hotelling-Lawley Trace	0.286883	0.24	9	9.8113	<= 0.9784
		Roy's Greatest Root	0.286883	0.96	3	10	>= 0.4504
Spline(x2)	-2.1539191	Wilks' Lambda	0.714529	0.32	9	19.621	<= 0.9572
		Pillai's Trace	0.285471	0.35	9	30	<= 0.9494
		Hotelling-Lawley Trace	0.399524	0.33	9	9.8113	<= 0.9424
		Roy's Greatest Root	0.399524	1.33	3	10	>= 0.3184
Spline(x3)	0.4779207	Wilks' Lambda	0.917838	0.08	9	19.621	<= 0.9998
		Pillai's Trace	0.082162	0.09	9	30	<= 0.9996
		Hotelling-Lawley Trace	0.089517	0.07	9	9.8113	<= 0.9997
		Roy's Greatest Root	0.089517	0.30	3	10	>= 0.8259

These statistics are adjusted in the same way as the multivariate statistics above.

Hypothesis Tests with One-Way ANOVA

One-way ANOVA models are fit with either an explicit or implicit intercept. In implicit intercept models, the ANOVA table of PROC TRANSREG is the correct table for a model with an intercept, and the regression table is the correct table for a model that does not have a separate explicit intercept. The PROC TRANSREG implicit intercept ANOVA table matches the PROC REG table when the NOINT *a-option* is not specified, and the PROC TRANSREG implicit intercept regression table matches the PROC REG table when the NOINT *a-option* is specified. The following statements illustrate this relationship and produce Figure 90.75:

```
data oneway;
   input y x $;
   datalines;
0 a
1 a
2 a
7 b
8 b
9 b
3 c
4 c
5 c
;

title 'Implicit Intercept Model';

proc transreg ss2 data=oneway short;
   model identity(y) = class(x / zero=none);
   output out=oneway2;
run;

proc reg data=oneway2;
   model y = xa xb xc;          /* Implicit Intercept ANOVA      */
   model y = xa xb xc / noint; /* Implicit Intercept Regression */
run; quit;
```

Figure 90.75 Implicit Intercept Model

```
                          Implicit Intercept Model

                           The TRANSREG Procedure

                        Dependent Variable Identity(y)


                           Class Level Information

                         Class    Levels    Values

                         x             3    a b c

                    Number of Observations Read                 9
                    Number of Observations Used                 9
                    Implicit Intercept Model
```

Figure 90.75 *continued*

The TRANSREG Procedure Hypothesis Tests for Identity(y)

Univariate ANOVA Table Based on the Usual Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	74.00000	37.00000	37.00	0.0004
Error	6	6.00000	1.00000		
Corrected Total	8	80.00000			

Root MSE	1.00000	R-Square	0.9250
Dependent Mean	4.33333	Adj R-Sq	0.9000
Coeff Var	23.07692		

Univariate Regression Table Based on the Usual Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Pr > F	Label
Class.xa	1	1.00000000	3.000	3.000	3.00	0.1340	x a
Class.xb	1	8.00000000	192.000	192.000	192.00	<.0001	x b
Class.xc	1	4.00000000	48.000	48.000	48.00	0.0004	x c

Implicit Intercept Model

The REG Procedure
Model: MODEL1
Dependent Variable: y

Number of Observations Read	9
Number of Observations Used	9

Analysis of Variance

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	2	74.00000	37.00000	37.00	0.0004
Error	6	6.00000	1.00000		
Corrected Total	8	80.00000			

Root MSE	1.00000	R-Square	0.9250
Dependent Mean	4.33333	Adj R-Sq	0.9000
Coeff Var	23.07692		

Figure 90.75 *continued*

```
NOTE: Model is not full rank. Least-squares solutions for the parameters are
      not unique. Some statistics will be misleading. A reported DF of 0 or B
      means that the estimate is biased.
NOTE: The following parameters have been set to 0, since the variables are a
      linear combination of other variables as shown.


                          xc =  Intercept - xa - xb

                             Parameter Estimates

                                Parameter      Standard
  Variable    Label       DF      Estimate         Error   t Value   Pr > |t|

  Intercept   Intercept    B       4.00000       0.57735      6.93     0.0004
  xa          x a          B      -3.00000       0.81650     -3.67     0.0104
  xb          x b          B       4.00000       0.81650      4.90     0.0027
  xc          x c          0             0             .         .         .

                           Implicit Intercept Model

                              The REG Procedure
                                Model: MODEL2
                            Dependent Variable: y

                   Number of Observations Read           9
                   Number of Observations Used           9

             NOTE: No intercept in model. R-Square is redefined.

                             Analysis of Variance

                                    Sum of           Mean
 Source                   DF        Squares         Square    F Value    Pr > F

 Model                     3      243.00000       81.00000      81.00    <.0001
 Error                     6        6.00000        1.00000
 Uncorrected Total         9      249.00000

             Root MSE              1.00000    R-Square     0.9759
             Dependent Mean        4.33333    Adj R-Sq     0.9639
             Coeff Var            23.07692

                             Parameter Estimates

                                Parameter      Standard
  Variable    Label       DF      Estimate         Error   t Value   Pr > |t|

  xa          x a          1       1.00000       0.57735      1.73     0.1340
  xb          x b          1       8.00000       0.57735     13.86     <.0001
  xc          x c          1       4.00000       0.57735      6.93     0.0004
```

Using the DESIGN Output Option

This example uses PROC TRANSREG and the DESIGN *o-option* to prepare an input data set with classification variables for the LOGISTIC procedure. The DESIGN *o-option* specifies that the goal is design matrix creation, not analysis. When you specify DESIGN, dependent variables are not required. The DEVIATIONS (or EFFECTS) *t-option* requests a deviations-from-means $(1, 0, -1)$ coding of the classification variables, which is the same coding the CATMOD procedure uses. PROC TRANSREG automatically creates a macro variable &_TrgInd that contains the list of independent variables created. This macro is used in the PROC LOGISTIC MODEL statement. (See Figure 90.76.) For comparison, the same analysis is also performed with PROC CATMOD. The following statements create Figure 90.76:

```
title 'Using PROC TRANSREG to Create a Design Matrix';

data a;
   do y = 1, 2;
      do a = 1 to 4;
         do b = 1 to 3;
            w = ceil(uniform(1) * 10 + 10);
            output;
         end;
      end;
   end;
run;

proc transreg data=a design;
   model class(a b / deviations);
   id y w;
   output out=coded;
run;

proc print;
   title2 'PROC TRANSREG Output Data Set';
run;

title2 'PROC LOGISTIC with Classification Variables';

proc logistic;
   freq w;
   model y = &_trgind;
run;

title2 'PROC CATMOD Should Produce the Same Results';

proc catmod data=a;
   model y = a b;
   weight w;
run;
```

Figure 90.76 The PROC TRANSREG Design Matrix

Using PROC TRANSREG to Create a Design Matrix
PROC LOGISTIC with Classification Variables

The LOGISTIC Procedure

Model Information

Data Set	WORK.CODED
Response Variable	y
Number of Response Levels	2
Frequency Variable	w
Model	binary logit
Optimization Technique	Fisher's scoring

Number of Observations Read	24
Number of Observations Used	24
Sum of Frequencies Read	375
Sum of Frequencies Used	375

Response Profile

Ordered Value	y	Total Frequency
1	1	188
2	2	187

Probability modeled is y=1.

Model Convergence Status

Convergence criterion (GCONV=1E-8) satisfied.

Model Fit Statistics

Criterion	Intercept Only	Intercept and Covariates
AIC	521.858	524.378
SC	525.785	547.939
-2 Log L	519.858	512.378

Testing Global Null Hypothesis: BETA=0

Test	Chi-Square	DF	Pr > ChiSq
Likelihood Ratio	7.4799	5	0.1873
Score	7.4312	5	0.1905
Wald	7.3356	5	0.1969

Figure 90.76 *continued*

Analysis of Maximum Likelihood Estimates

Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	−0.00040	0.1044	0.0000	0.9969
a1	1	−0.0802	0.1791	0.2007	0.6542
a2	1	0.2001	0.1800	1.2363	0.2662
a3	1	−0.1350	0.1819	0.5514	0.4578
b1	1	−0.2392	0.1500	2.5436	0.1107
b2	1	0.3433	0.1474	5.4223	0.0199

Odds Ratio Estimates

Effect	Point Estimate	95% Wald Confidence Limits	
a1	0.923	0.650	1.311
a2	1.222	0.858	1.738
a3	0.874	0.612	1.248
b1	0.787	0.587	1.056
b2	1.410	1.056	1.882

Association of Predicted Probabilities and Observed Responses

Percent Concordant	54.0	Somers' D	0.163
Percent Discordant	37.8	Gamma	0.177
Percent Tied	8.2	Tau-a	0.082
Pairs	35156	c	0.581

Using PROC TRANSREG to Create a Design Matrix
PROC CATMOD Should Produce the Same Results

The CATMOD Procedure

Data Summary

Response	y	Response Levels	2
Weight Variable	w	Populations	12
Data Set	A	Total Frequency	375
Frequency Missing	0	Observations	24

Figure 90.76 *continued*

Population Profiles

Sample	a	b	Sample Size
1	1	1	31
2	1	2	31
3	1	3	34
4	2	1	26
5	2	2	33
6	2	3	37
7	3	1	36
8	3	2	29
9	3	3	28
10	4	1	26
11	4	2	35
12	4	3	29

Response Profiles

Response	y
1	1
2	2

Maximum Likelihood Analysis

Maximum likelihood computations converged.

Maximum Likelihood Analysis of Variance

Source	DF	Chi-Square	Pr > ChiSq
Intercept	1	0.00	0.9969
a	3	1.50	0.6823
b	2	5.64	0.0597
Likelihood Ratio	6	2.81	0.8329

Analysis of Maximum Likelihood Estimates

Parameter		Estimate	Standard Error	Chi-Square	Pr > ChiSq
Intercept		-0.00040	0.1044	0.00	0.9969
a	1	-0.0802	0.1791	0.20	0.6542
	2	0.2001	0.1800	1.24	0.2662
	3	-0.1350	0.1819	0.55	0.4578
b	1	-0.2392	0.1500	2.54	0.1107
	2	0.3434	0.1474	5.42	0.0199

Discrete Choice Experiments: DESIGN, NORESTORE, NOZERO

A discrete choice experiment is constructed consisting of four product brands, each available at three different prices, \$1.49, \$1.99, \$2.49. In addition, each choice set contains a constant "other" alternative available at \$1.49. In the fifth choice set, price is constant. PROC TRANSREG is used to code the design, and the PHREG procedure fits the multinomial logit choice model (not shown). See http://support.sas.com/resources/papers/tnote/tnote_marketresearch.html (Kuhfeld 2005) for more information about discrete choice modeling and the multinomial logit model; look for the latest "Discrete Choice" report. The following statements produce Figure 90.77:

```
title 'Choice Model Coding';

data design;
   array p[4];
   input p1-p4 @@;
   set = _n_;
   do brand = 1 to 4;
      price = p[brand];
      output;
   end;
   brand = .; price = 1.49; output; /* constant alternative */
   keep set brand price;
   datalines;
1.49 1.99 1.49 1.99 1.99 1.99 2.49 1.49 1.99 1.49 1.99 1.49
1.99 1.49 2.49 1.99 1.49 1.49 1.49 1.49 2.49 1.49 1.99 2.49
1.49 1.49 2.49 2.49 2.49 2.49 1.49 1.49 1.49 2.49 2.49 1.99
2.49 2.49 2.49 1.49 1.99 2.49 1.49 2.49 2.49 1.99 2.49 2.49
2.49 1.49 1.49 1.99 1.49 1.99 1.99 1.49 2.49 1.99 1.99 1.99
1.99 1.99 1.49 2.49 1.99 2.49 1.99 1.99 1.49 2.49 1.99 2.49
;

proc transreg data=design design norestoremissing nozeroconstant;
   model class(brand / zero=none) identity(price);
   output out=coded;
   by set;
run;

proc print data=coded(firstobs=21 obs=25);
   var set brand &_trgind;
run;
```

In the interest of space, only the fifth choice set is displayed in Figure 90.77.

Figure 90.77 The Fifth Choice Set

Choice Model Coding

Obs	set	brand	brand1	brand2	brand3	brand4	price
21	5	1	1	0	0	0	1.49
22	5	2	0	1	0	0	1.49
23	5	3	0	0	1	0	1.49
24	5	4	0	0	0	1	1.49
25	5	.	0	0	0	0	1.49

For the constant alternative (Brand = .), the brand coding is a row of zeros due to the NORESTOREMISSING *o-option*, and Price is a constant \$1.49 (instead of 0) due to the NOZEROCONSTANT.

The data set was coded by choice set (BY set;). This is a small problem. With very large problems, it might be necessary to restrict the number of observations that are coded at one time so that the procedure uses less time and memory. Coding by choice set is one option. When coding is performed after the data are merged in, coding by subject and choice set combinations is another option. Alternatively, you can specify DESIGN=*n*, where *n* is the number of observations to code at one time. For example, you can specify DESIGN=100 or DESIGN=1000 to process the data set in blocks of 100 or 1000 observations. Specify the NOZEROCONSTANT *a-option* to ensure that constant variables within blocks are not zeroed. When you specify DESIGN=*n*, or perform coding after the data are merged in, specify the dependent variable and any other variables needed for analysis as ID variables.

Centering

You can use transformation options to center and standardize the variables in several ways. For example, the following MODEL statement creates three independent variables, x, x^2, and x^3:

```
model identity(y) = pspline(x);
```

The variables are not centered.

When the CENTER *t-option* is specified, as in the following statement, the independent variable is centered before squaring and cubing:

```
model identity(y) = pspline(x / center);
```

The three independent variables are $x - \bar{x}$, $(x - \bar{x})^2$, and $(x - \bar{x})^3$.

Since operations such as squaring occur after the centering, the resulting variables are not always centered. The CENTER *t-option* is particularly useful with polynomials since centering before squaring and cubing can help reduce collinearity and numerical problems. For example, if one of your variables is year, with values all greater than 1900, squaring and cubing without centering first will create variables that are all essentially perfectly correlated.

When the TSTANDARD=CENTER *t-option* is specified, as in the following model, the three independent variables are squared and cubed and then centered:

```
model identity(y) = pspline(x / tstandard=center);
```

The three independent variables are $x - \bar{x}$, $x^2 - \overline{x^2}$, and $x^3 - \overline{x^3}$.

You can specify both the CENTER and TSTANDARD=CENTER *t-options* to center the variables, then square and cube them, and then center the results, as in the following statement:

```
model identity(y) = pspline(x / center tstandard=center);
```

The three independent variables are $x - \bar{x}$, $(x - \bar{x})^2 - \overline{(x - \bar{x})^2}$, and $(x - \bar{x})^3 - \overline{(x - \bar{x})^3}$.

Displayed Output

The display options control the amount of displayed output. The displayed output can contain the following:

- an iteration history and convergence status table (by default when there are iterations)
- an ANOVA table when the TEST, SS2, or UTILITIES *a-option* is specified
- a regression table when the SS2 *a-option* is specified
- conjoint analysis part-worth utilities when the UTILITIES *a-option* is specified
- model details when the DETAIL *a-option* is specified
- a multivariate ANOVA table when the dependent variable is transformed and the TEST or SS2 *a-option* is specified
- a multivariate regression table when the dependent variable is transformed and it is specified
- liberal and conservative ANOVA, multivariate ANOVA, regression, and multivariate regression tables when there is a MONOTONE, UNTIE, or MSPLINE transformation and the TEST or SS2 *a-option* is specified

ODS Table Names

PROC TRANSREG assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 90.8. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 90.8 ODS Tables Produced by PROC TRANSREG

ODS Table Name	Description	Statement & Option
ANOVA	ANOVA	MODEL/PROC, TEST/SS2
BoxCox	Box-Cox transformation results	MODEL, BOXCOX
CANALS	CANALS iteration history	MODEL/PROC, METHOD=CANALS
ClassLevels	ANOVA	MODEL/PROC, TEST/SS2
Coef	Regression results	MODEL/PROC, SS2
ConservANOVA	ANOVA, *1	MODEL/PROC, TEST/SS2
ConservCoef	Regression results, *1	MODEL/PROC, SS2
ConservFitStatistics	Fit statistics, *1	MODEL/PROC, TEST/SS2
ConservMVANOVA	Multivariate ANOVA, *1, *2	MODEL/PROC, TEST/SS2
ConservMVCoef	Multivariate regression results, *1, *2	MODEL/PROC, SS2
ConservUtilities	Conjoint analysis utilities, *1	MODEL/PROC, UTILITIES
ConvergenceStatus	Convergence status	default
Details	Model Details	MODEL/PROC, DETAIL
Equation	Linear dependency equation	less-than-full-rank model
FitStatistics	Fit statistics like R square	MODEL/PROC, TEST/SS2
Footnotes	Iteration history footnotes	default
LiberalANOVA	ANOVA, *1	MODEL/PROC, TEST/SS2
LiberalCoef	Regression results, *1	MODEL/PROC, SS2
LiberalFitStatistics	Fit statistics, *1	MODEL/PROC, TEST/SS2
LiberalMVANOVA	Multivariate ANOVA, *1, *2	MODEL/PROC, TEST/SS2
LiberalMVCoef	Multivariate regression results, *1, *2	MODEL/PROC, SS2
LiberalUtilities	Conjoint analysis utilities, *1	MODEL/PROC, UTILITIES
MORALS	MORALS iteration history	MODEL/PROC, METHOD=MORALS
MVANOVA	Multivariate ANOVA, *2	MODEL/PROC, TEST/SS2
MVCoef	Multivariate regression results, *2	MODEL/PROC, SS2
NObs	ANOVA	MODEL/PROC, TEST/SS2
PBSplineCriteria	Penalized B-spline criteria (nonprinting)	MODEL, PBSPLINE
RSquare	R square	MODEL/PROC, RSQUARE
Redundancy	Redundancy iteration history	MODEL/PROC, METHOD=REDUNDANCY
SplineCoef	Spline coefficients (nonprinting)	MODEL, SPLINE/MSPLINE
TestIterations	Hypothesis test iterations iteration history	MODEL/PROC, SS2
Univariate	Univariate iteration history	MODEL/PROC, METHOD=UNIVARIATE
Utilities	Conjoint analysis utilities	MODEL/PROC, UTILITIES

*1. Liberal and conservative test tables are produced when a MONOTONE, UNTIE, or MSPLINE transformation is requested.

*2. Multivariate tables are produced when the dependent variable is iteratively transformed.

ODS Graphics

To request graphics with PROC TRANSREG, you must first enable ODS Graphics by specifying the ODS GRAPHICS ON statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. Some graphs are produced by default; other graphs are produced by using statements and options. You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC TRANSREG generates are listed in Table 90.9, along with the required statements and options.

Table 90.9 ODS Graphics Produced by PROC TRANSREG

ODS Graph Name	Plot Description	Statement & Option
BoxCoxFPlot	Box-Cox $F = t^2$	MODEL & PROC, BOXCOX *transform* & PLOTS(UNPACK)
BoxCoxLogLikePlot	Box-Cox Log Likelihood	MODEL & PROC, BOXCOX *transform* & PLOTS(UNPACK)
BoxCoxPlot	Box-Cox t or $F = t^2$ & Log Likelihood	MODEL, BOXCOX *transform*
BoxCoxtPlot	Box-Cox t	MODEL & PROC, BOXCOX *transform* & PLOTS(UNPACK)=BOXCOX(T)
FitPlot	Simple Regression and Separate Group Regressions	MODEL, a dependent variable that is not transformed, one non-CLASS independent variable, and at most one CLASS variable
ObservedByPredicted	Dependent Variable by Predicted Values	MODEL, PLOTS=OBSERVEDBYPREDICTED
PBSPlineCritPlot	Penalized B-Spline Criterion Plot	MODEL, PBSPLINE *transform*
PrefMapVecPlot	Preference Mapping Vector Plot	MODEL & PROC, IDENTITY *transform* & COORDINATES
PrefMapIdealPlot	Preference Mapping Ideal Point Plot	MODEL & PROC, POINT expansion & COORDINATES
ResidualPlot	Residuals	PROC, PLOTS=RESIDUALS
RMSEPlot	Box-Cox Root Mean Square Error	MODEL & PROC, BOXCOX *transform* & PLOTS=BOXCOX(RMSE)
ScatterPlot	Scatter Plot of Observed Data	MODEL, one non-CLASS independent variable, and at most one CLASS variable, PLOTS=SCATTER
TransformationPlot	Variable Transformations	PROC, PLOTS=TRANSFORMATION

The PLOTS(INTERPOLATE) Option

This section illustrates one use of the PLOTS(INTERPOLATE) option for use with ODS Graphics. The data set has two groups of observations, c = 1 and c = 2. Each group is sparse, having only five observations, so the plots of the transformations and fit functions are not smooth. A second DATA step adds additional observations to the data set, over the range of x, with y missing. These observations do not contribute to the analysis, but they are used in computations of transformed and predicted values. The resulting plots are much smoother in the latter case than in the former. The other results of the analysis are the same. The following statements produce Figure 90.78 and Figure 90.79:

```
title 'Smoother Interpolation with PLOTS(INTERPOLATE)';

data a;
   input c y x;
   output;
   datalines;
1 1 1
1 2 2
1 4 3
1 6 4
1 7 5
2 3 1
2 4 2
2 5 3
2 4 4
2 5 5
;

ods graphics on;

proc transreg data=a plots=(tran fit) ss2;
   model ide(y) = pbs(x) * class(c / zero=none);
   run;

data b;
   set a end=eof;
   output;
   if eof then do;
      y = .;
      do x = 1 to 5 by 0.05;
         c = 1; output;
         c = 2; output;
         end;
      end;
   run;

proc transreg data=b plots(interpolate)=(tran fit) ss2;
   model ide(y) = pbs(x) * class(c / zero=none);
   run;

ods graphics off;
```

The results with no interpolation are shown in Figure 90.78. The transformation and fit functions are not at all smooth. The results with interpolation are shown in Figure 90.79. The transformation and fit functions are smooth in Figure 90.79, because there are intermediate points to plot.

Figure 90.78 No Interpolation

Smoother Interpolation with PLOTS(INTERPOLATE)

The TRANSREG Procedure

Univariate ANOVA Table, Penalized B-Spline Transformation

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	9	28.90000	3.211111	Infty	<.0001
Error	12E-10	0.00000	0.000000		
Corrected Total	9	28.90000			

Root MSE	0	R-Square	1.0000
Dependent Mean	4.10000	Adj R-Sq	1.0000
Coeff Var	0		

Penalized B-Spline Transformation

Variable	DF	Coefficient	Lambda	AICC	Label
Pbspline(xc1)	5.0000	1.000	2.642E-7	-66.4281	x * c 1
Pbspline(xc2)	5.0000	1.000	2.516E-7	-60.6430	x * c 2

Figure 90.78 *continued*

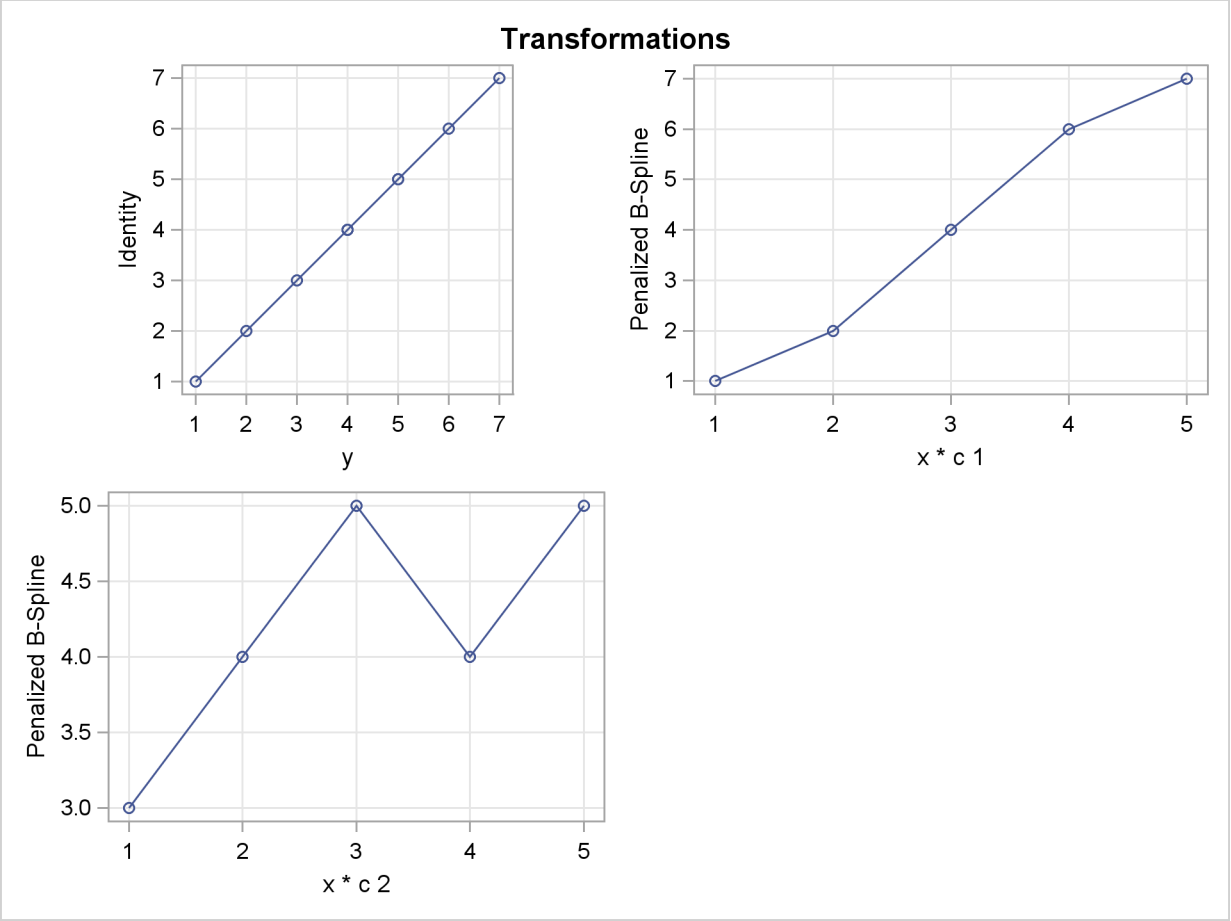


Figure 90.78 *continued*

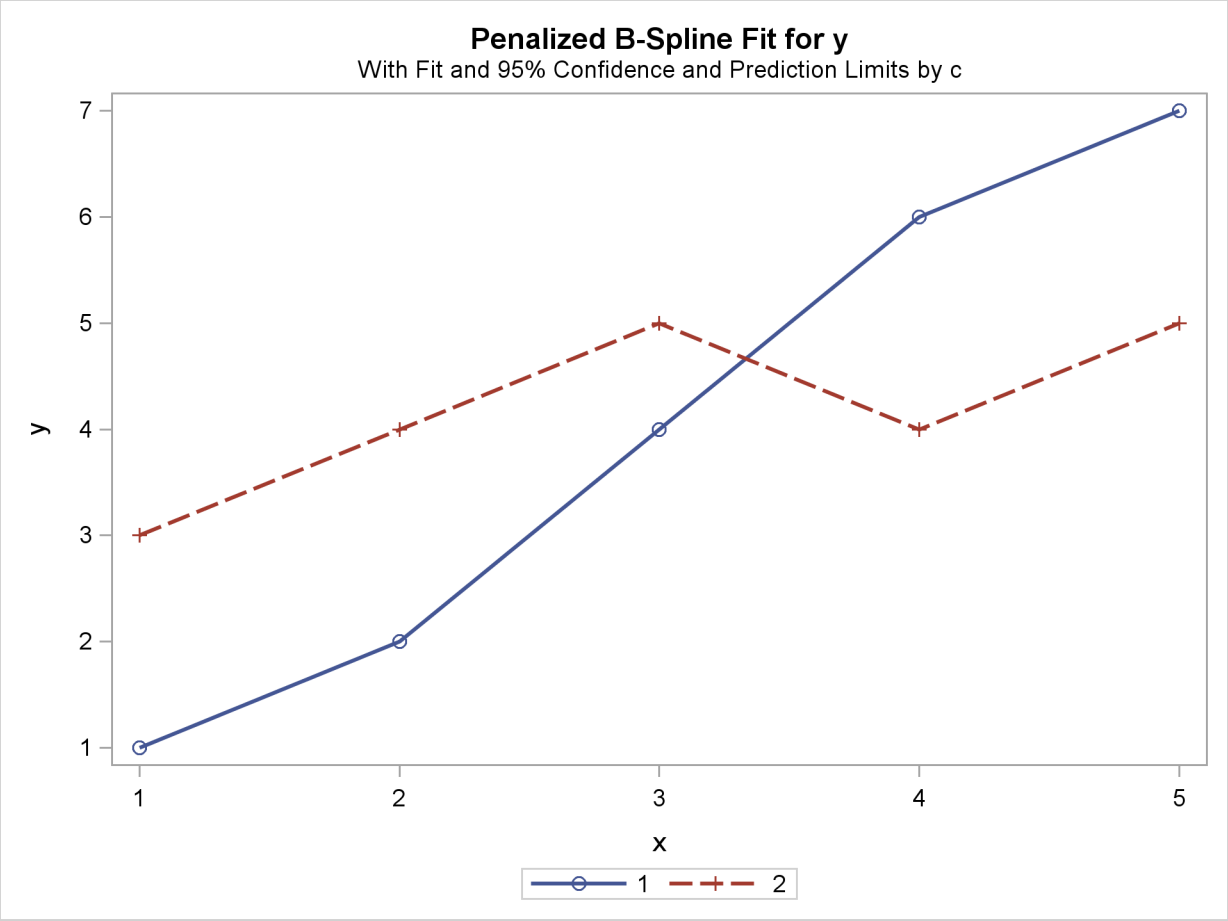


Figure 90.79 Interpolation with PLOTS(INTERPOLATE)

Smoother Interpolation with PLOTS(INTERPOLATE)

The TRANSREG Procedure

Univariate ANOVA Table, Penalized B-Spline Transformation

Source	DF	Sum of Squares	Mean Square	F Value	Pr > F
Model	9	28.90000	3.211111	Infty	<.0001
Error	12E-10	0.00000	0.000000		
Corrected Total	9	28.90000			

Root MSE	0	R-Square	1.0000
Dependent Mean	4.10000	Adj R-Sq	1.0000
Coeff Var	0		

Figure 90.79 *continued*

Penalized B-Spline Transformation

Variable	DF	Coefficient	Lambda	AICC	Label
Pbspline(xc1)	5.0000	1.000	2.642E-7	-66.4281	x * c 1
Pbspline(xc2)	5.0000	1.000	2.516E-7	-60.6430	x * c 2

Figure 90.79 *continued*

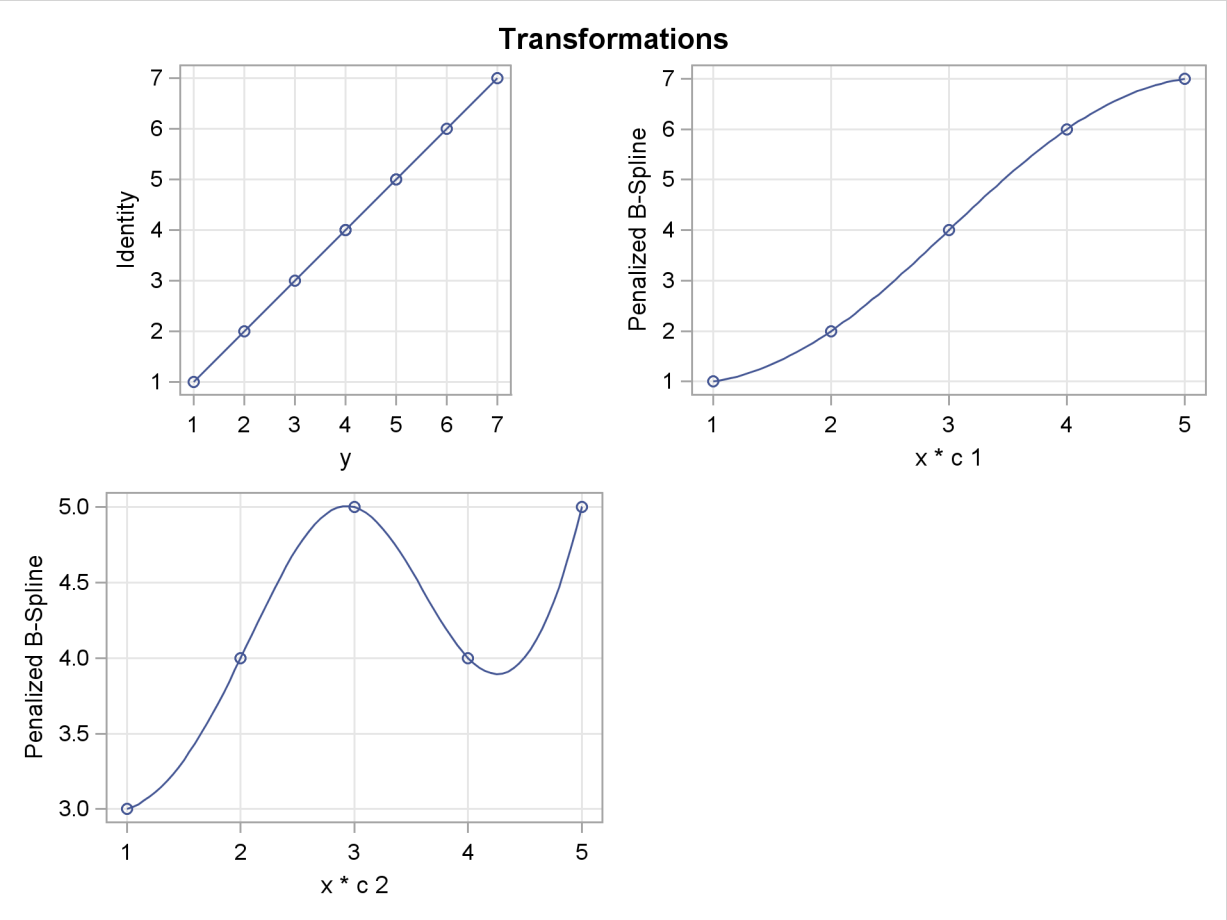
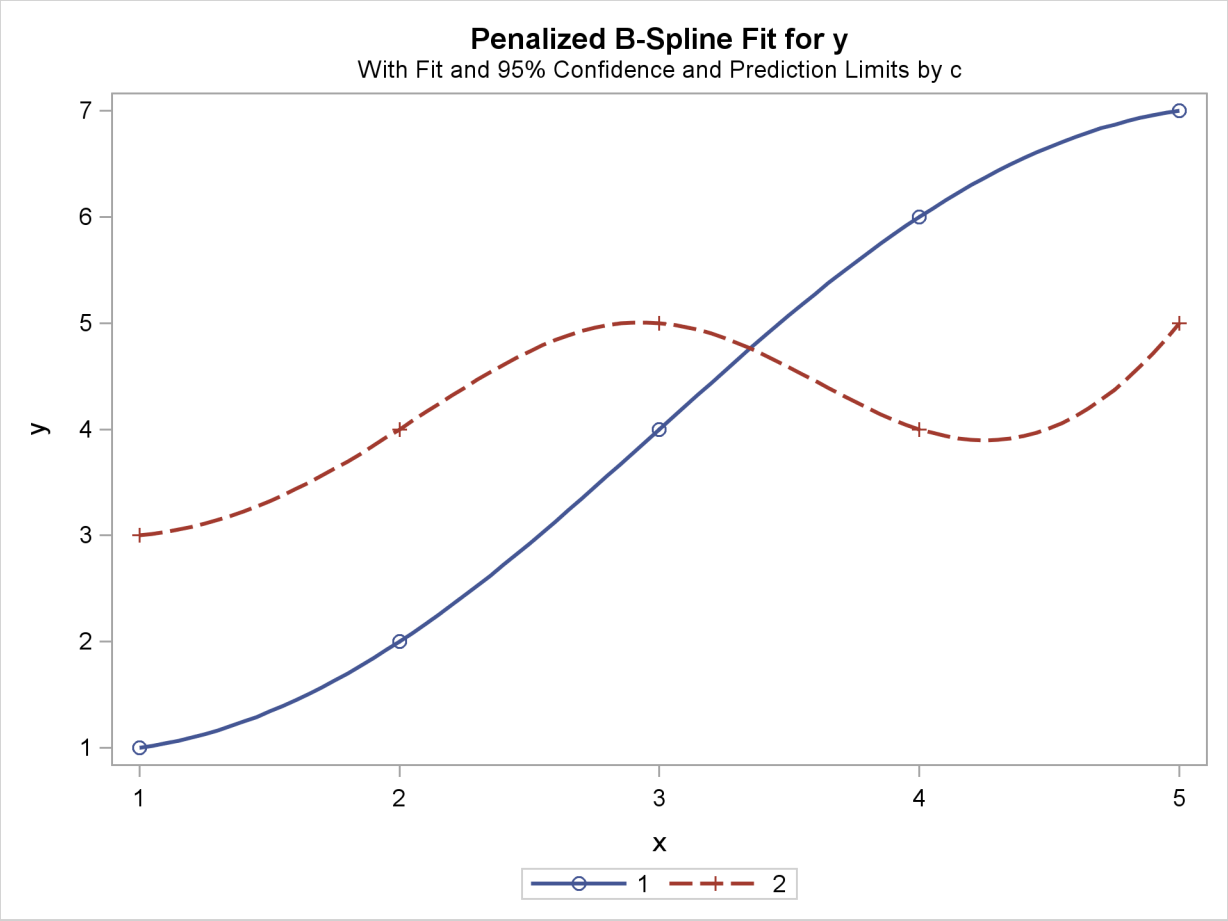


Figure 90.79 *continued*



Examples: TRANSREG Procedure

Example 90.1: Transformation Regression of Exhaust Emissions Data

In this example, the data are from an experiment in which nitrogen oxide emissions from a single cylinder engine are measured for various combinations of fuel, compression ratio, and equivalence ratio. The data are provided by Brinkman (1981).

The equivalence ratio and nitrogen oxide variables are continuous and numeric, so spline transformations of these variables are requested. The spline transformation of the dependent variable is restricted to be monotonic. Each spline is degree three with nine knots (one at each decile) in order to give PROC TRANSREG a great deal of freedom in finding transformations. The compression ratio variable has only five discrete values, so an optimal scoring is requested with monotonicity constraints. The character variable Fuel is nominal, so it is optimally scored without any monotonicity constraints. Observations with missing values are excluded with the NOMISS *a-option*. The following statements produce Output 90.1.1:

```
title 'Gasoline Example';

data Gas;
   input Fuel :$8. CpRatio EqRatio NOx @@;
   label Fuel    = 'Fuel'
         CpRatio = 'Compression Ratio (CR)'
         EqRatio = 'Equivalence Ratio (PHI)'
         NOx     = 'Nitrogen Oxide (NOx)';
   datalines;
Ethanol  12.0 0.907 3.741 Ethanol  12.0 0.761 2.295
Ethanol  12.0 1.108 1.498 Ethanol  12.0 1.016 2.881
Ethanol  12.0 1.189 0.760 Ethanol   9.0 1.001 3.120

... more lines ...

94%Eth    7.5 1.075 2.147
;
ods graphics on;

title2 'Iteratively Estimate NOx, CpRatio, EqRatio, and Fuel';

* Fit the Nonparametric Model;
proc transreg data=Gas solve test nomiss plots=all;
   ods exclude where=(_path_ ? 'MV');
   model mspline(NOx / nknots=9) = spline(EqRatio / nknots=9)
                                   monotone(CpRatio) opscore(Fuel);
run;
```

Output 90.1.1 Transformation Regression Example: The Nonparametric Model

```
                          Gasoline Example
              Iteratively Estimate NOx, CpRatio, EqRatio, and Fuel

                          The TRANSREG Procedure

                     Dependent Variable Mspline(NOx)
                           Nitrogen Oxide (NOx)


                  Number of Observations Read           171
                  Number of Observations Used           169
```

Output 90.1.1 *continued*

TRANSREG MORALS Algorithm Iteration History for Mspline(NOx)

Iteration Number	Average Change	Maximum Change	R-Square	Criterion Change	Note
0	0.41900	3.80550	0.05241		
1	0.11984	0.83327	0.91028	0.85787	
2	0.03727	0.17688	0.93981	0.02953	
3	0.02795	0.10880	0.94969	0.00987	
4	0.02088	0.07279	0.95382	0.00413	
5	0.01530	0.05031	0.95582	0.00201	
6	0.01130	0.03922	0.95688	0.00106	
7	0.00852	0.03197	0.95748	0.00060	
8	0.00657	0.02531	0.95783	0.00035	
9	0.00510	0.01975	0.95805	0.00022	
10	0.00398	0.01534	0.95818	0.00013	
11	0.00314	0.01200	0.95827	0.00009	
12	0.00250	0.00953	0.95832	0.00005	
13	0.00199	0.00752	0.95836	0.00003	
14	0.00159	0.00594	0.95838	0.00002	
15	0.00127	0.00470	0.95839	0.00001	
16	0.00102	0.00373	0.95840	0.00001	
17	0.00081	0.00297	0.95841	0.00001	
18	0.00065	0.00237	0.95841	0.00000	
19	0.00052	0.00189	0.95841	0.00000	
20	0.00042	0.00151	0.95842	0.00000	
21	0.00033	0.00120	0.95842	0.00000	
22	0.00027	0.00096	0.95842	0.00000	
23	0.00021	0.00077	0.95842	0.00000	
24	0.00017	0.00061	0.95842	0.00000	
25	0.00014	0.00049	0.95842	0.00000	
26	0.00011	0.00039	0.95842	0.00000	
27	0.00009	0.00031	0.95842	0.00000	
28	0.00007	0.00025	0.95842	0.00000	
29	0.00006	0.00020	0.95842	0.00000	
30	0.00005	0.00016	0.95842	0.00000	Not Converged

WARNING: Failed to converge, however criterion change is less than 0.0001.

The TRANSREG Procedure Hypothesis Tests for Mspline(NOx)
Nitrogen Oxide (NOx)

Univariate ANOVA Table Based on the Usual Degrees of Freedom

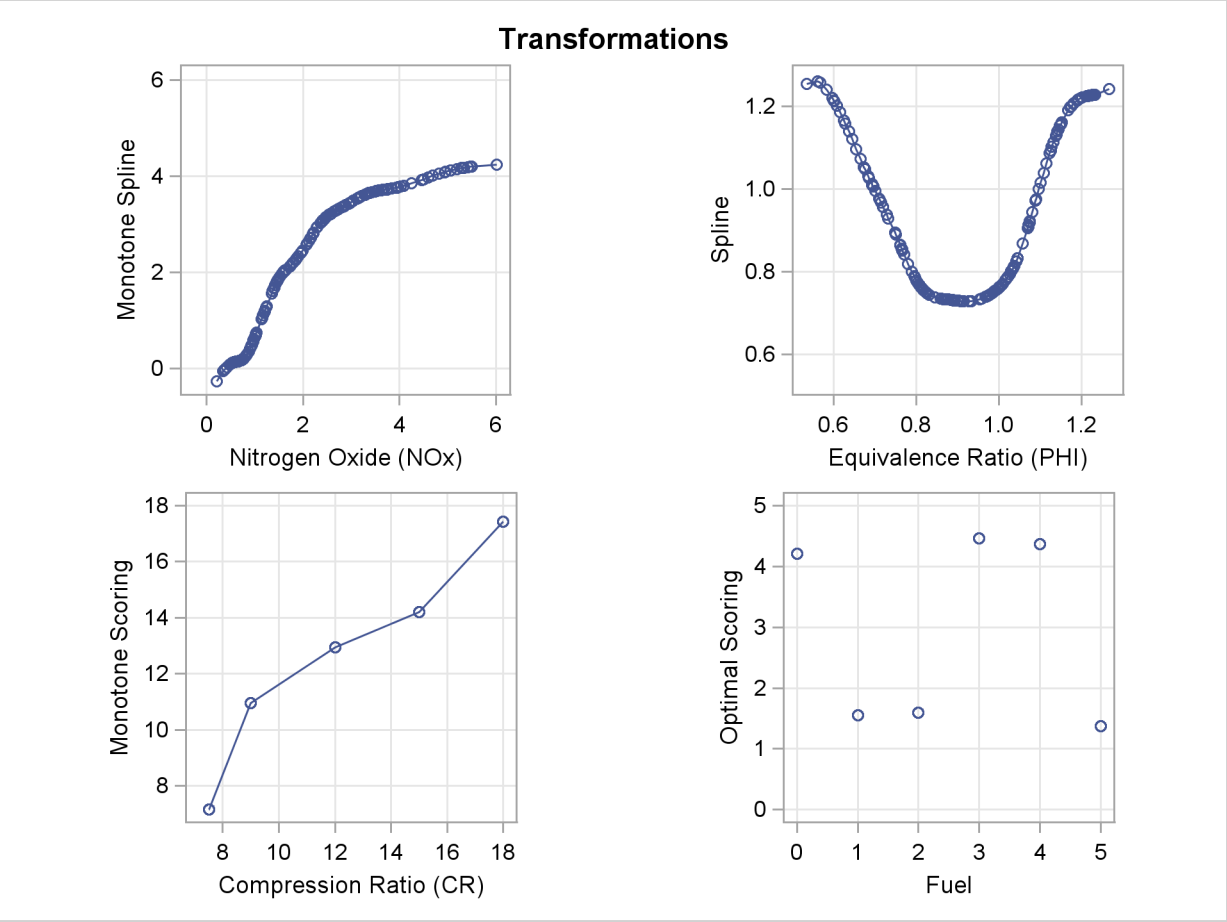
Source	DF	Sum of Squares	Mean Square	F Value	Liberal p
Model	21	326.0176	15.52465	161.35	>= <.0001
Error	147	14.1443	0.09622		
Corrected Total	168	340.1619			

The above statistics are not adjusted for the fact that the dependent variable was transformed and so are generally liberal.

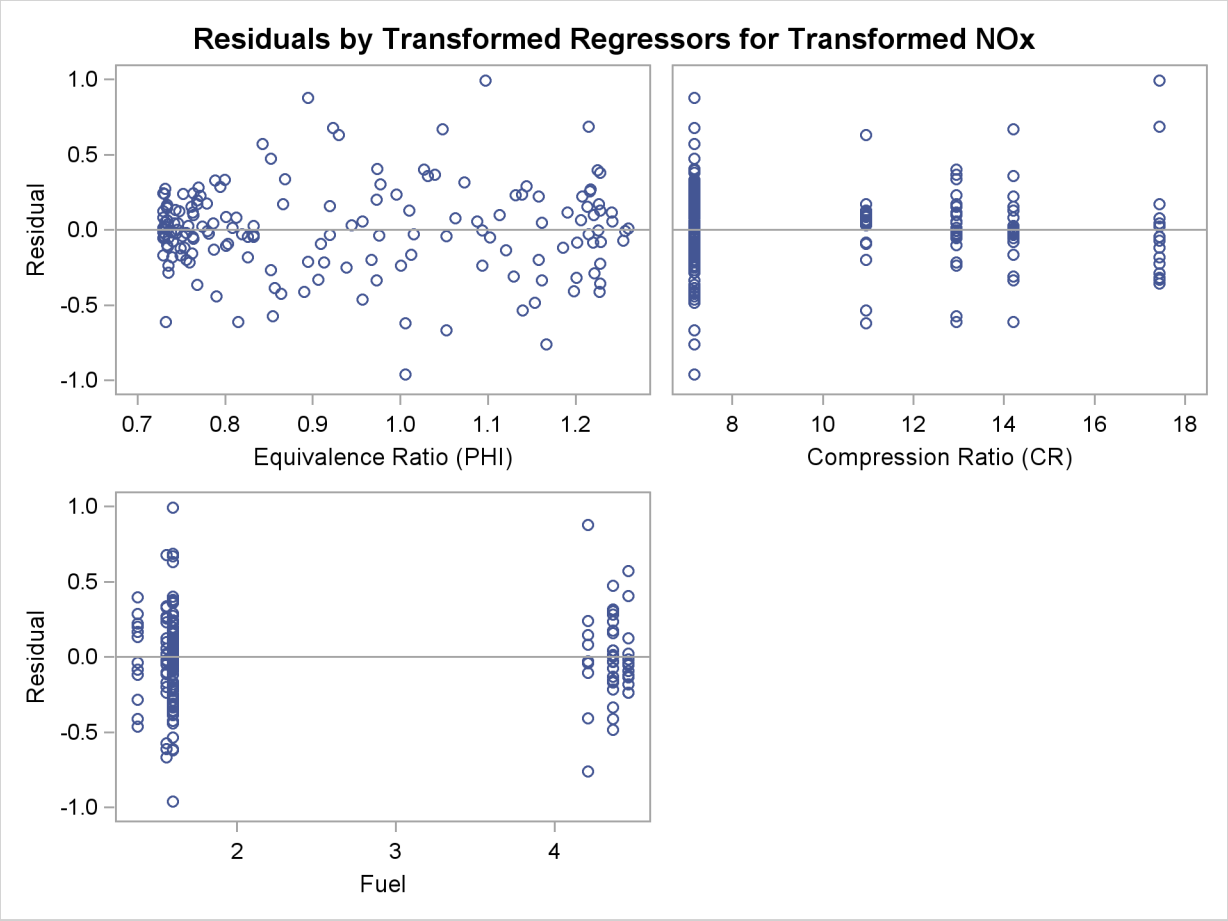
Output 90.1.1 *continued*

Root MSE	0.31019	R-Square	0.9584
Dependent Mean	2.34593	Adj R-Sq	0.9525
Coeff Var	13.22262		

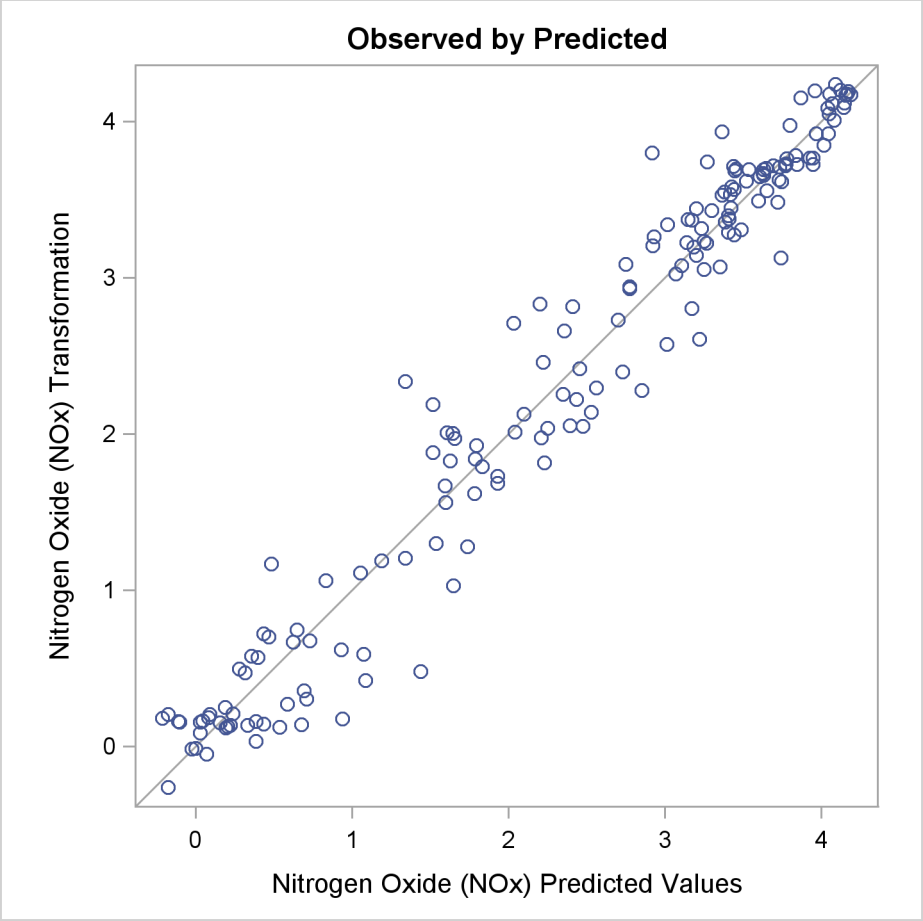
Output 90.1.1 *continued*



Output 90.1.1 *continued*



Output 90.1.1 *continued*



The squared multiple correlation for the initial model is approximately 0.05. PROC TRANSREG increases the R square to over 0.95 by transforming the variables. The transformation plots show how each variable is transformed. The transformation of compression ratio (TCpRatio) is nearly linear. The transformation of equivalence ratio (TEqRatio) is nearly parabolic. It can be seen from this plot that the optimal transformation of equivalence ratio is nearly uncorrelated with the original scoring. This suggests that the large increase in R square is due to this transformation. The transformation of nitrogen oxide (TNOx) is similar to a log transformation. The final plot shows the transformed dependent variable plotted as a function of the predicted values. This plot is reasonably linear, showing that the nonlinearities in the data are being accounted for fairly well by the TRANSREG model.

These results suggest the parametric model

$$\begin{aligned}\log(\text{NOX}) &= b_0 + b_1 \times \text{EqRatio} + b_2 \times \text{EqRatio}^2 + b_3 \times \text{CpRatio} \\ &\quad + \sum_j b_j \text{class}_j(\text{Fuel}) + \text{error}\end{aligned}$$

You can perform this analysis with PROC TRANSREG. The following statements produce Output 90.1.2:

```
title2 'Now fit log(NOx) = b0 + b1*EqRatio + b2*EqRatio**2 +';
title3 'b3*CpRatio + Sum b(j)*Fuel(j) + Error';

*-Fit the Parametric Model Suggested by the Nonparametric Analysis-;
proc transreg data=Gas solve ss2 short nomiss plots=all;
   model log(NOx) = pspline(EqRatio / deg=2) identity(CpRatio)
                    opscore(Fuel);
run;

ods graphics off;
```

Output 90.1.2 Transformation Regression Example: The Parametric Model

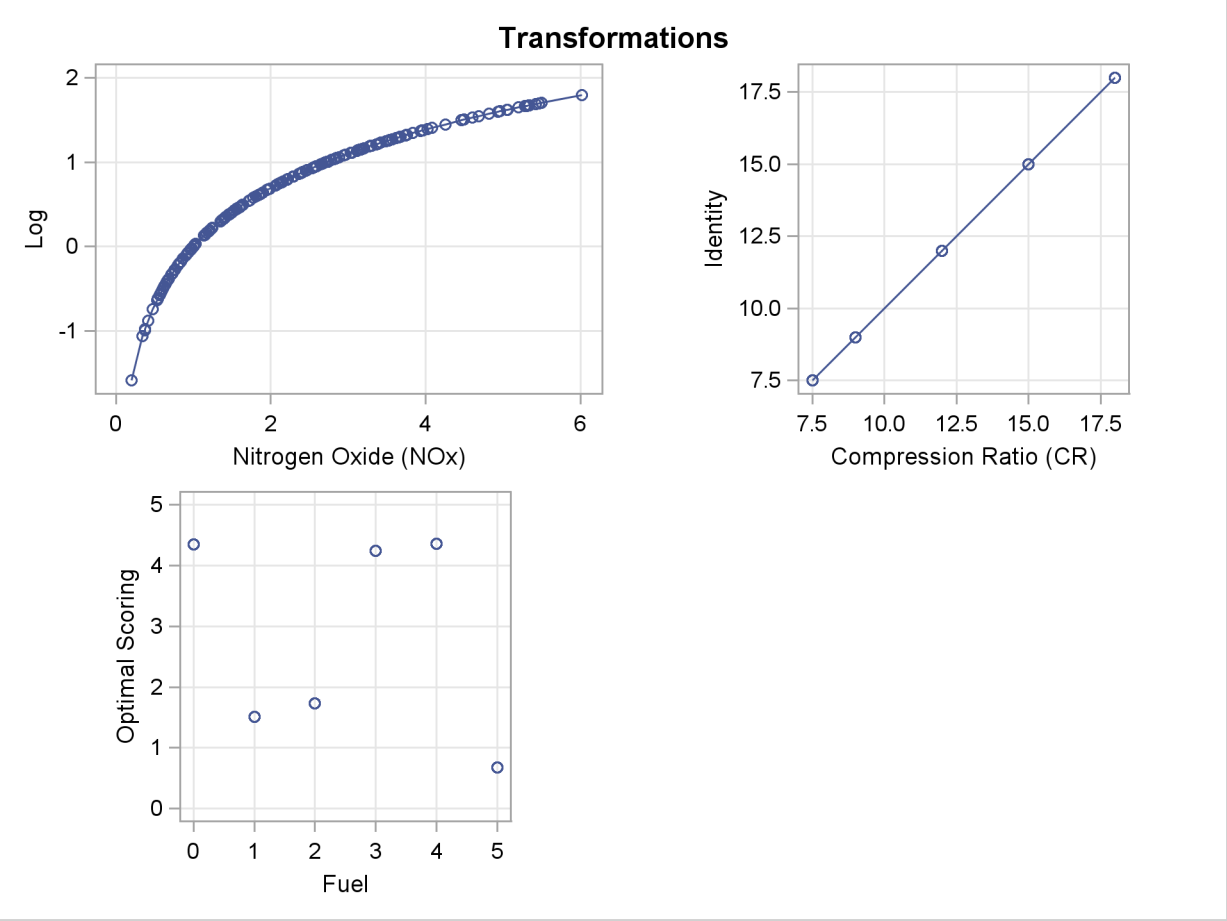
```
                              Gasoline Example
             Now fit log(NOx) = b0 + b1*EqRatio + b2*EqRatio**2 +
                    b3*CpRatio + Sum b(j)*Fuel(j) + Error

                            The TRANSREG Procedure

                         Dependent Variable Log(NOx)
                             Nitrogen Oxide (NOx)


                  Number of Observations Read            171
                  Number of Observations Used            169

   Log(NOx)
   Algorithm converged.

             The TRANSREG Procedure Hypothesis Tests for Log(NOx)
                             Nitrogen Oxide (NOx)


         Univariate ANOVA Table Based on the Usual Degrees of Freedom

                                  Sum of        Mean
     Source               DF     Squares      Square    F Value    Pr > F

     Model                 8    79.33838    9.917298     213.09    <.0001
     Error               160     7.44659    0.046541
     Corrected Total     168    86.78498

               Root MSE           0.21573    R-Square    0.9142
               Dependent Mean     0.63130    Adj R-Sq    0.9099
               Coeff Var         34.17294
```

Output 90.1.2 *continued*

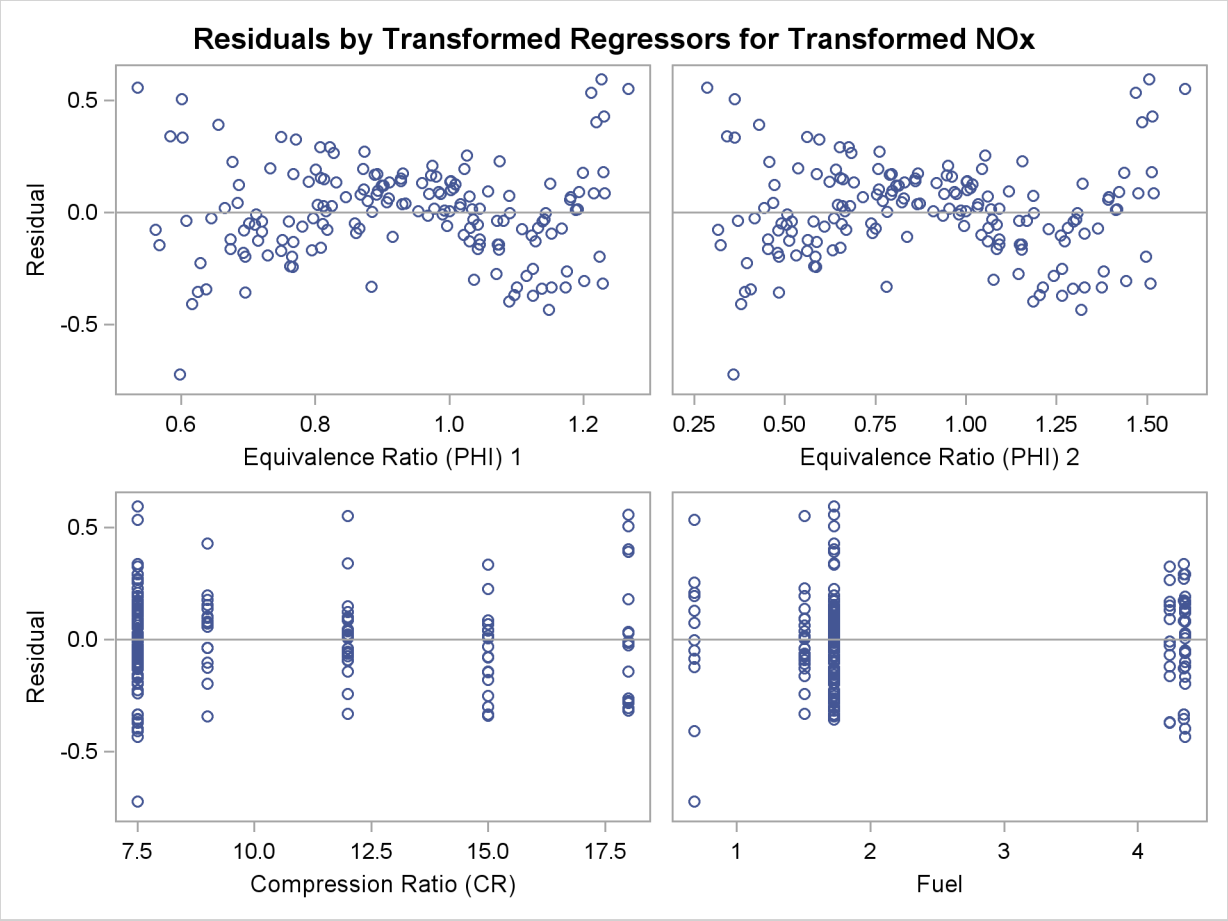
Univariate Regression Table Based on the Usual Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Pr > F	Label
Intercept	1	-15.274649	57.1338	57.1338	1227.60	<.0001	Intercept
Pspline.EqRatio_1	1	35.102914	62.7478	62.7478	1348.22	<.0001	Equivalence Ratio (PHI) 1
Pspline.EqRatio_2	1	-19.386468	64.6430	64.6430	1388.94	<.0001	Equivalence Ratio (PHI) 2
Identity(CpRatio)	1	0.032058	1.4445	1.4445	31.04	<.0001	Compression Ratio (CR)
Opscore(Fuel)	5	0.158388	5.5619	1.1124	23.90	<.0001	Fuel

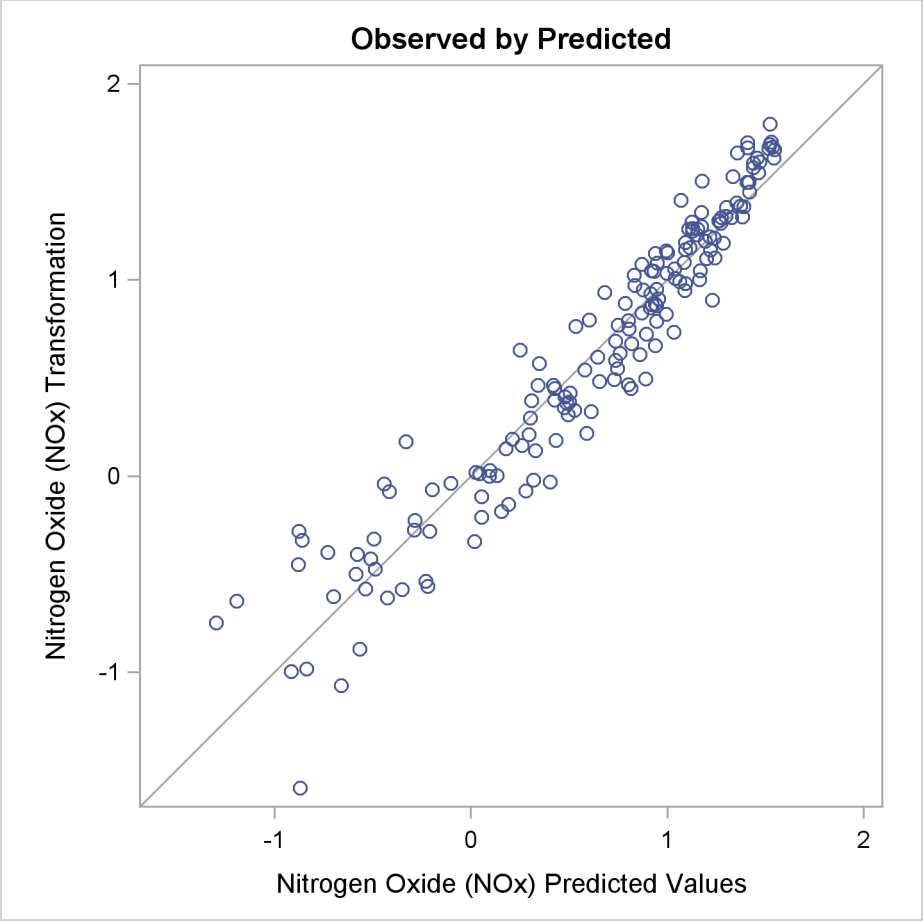
Output 90.1.2 *continued*



Output 90.1.2 *continued*



Output 90.1.2 *continued*



The LOG transformation computes the natural log. The PSPLINE expansion expands EqRatio into a linear term, EqRatio, and a squared term, EqRatio^2. An identity transformation of CpRatio and an optimal scoring of Fuel is requested. These should provide a good parametric operationalization of the optimal transformations. The final model has an R square of 0.91 (smaller than before since the model has fewer parameters, but still quite good).

Example 90.2: Box-Cox Transformations

This example shows Box-Cox transformations with a yarn failure data set. For more information about Box-Cox transformations, including using a Box-Cox transformation in a model with no independent variable, to normalize the distribution of the data, see the section "Box-Cox Transformations" on page 7181. In this example, a simple 3^3 design was used to study the effects of different factors on the failure of a yarn manufacturing process. The design factors are as follows:

- the length of test specimens of yarn, with levels of 250, 300, and 350 mm
- the amplitude of the loading cycle, with levels of 8, 9, and 10 mmd

- the load with levels of 40, 45, and 50 grams

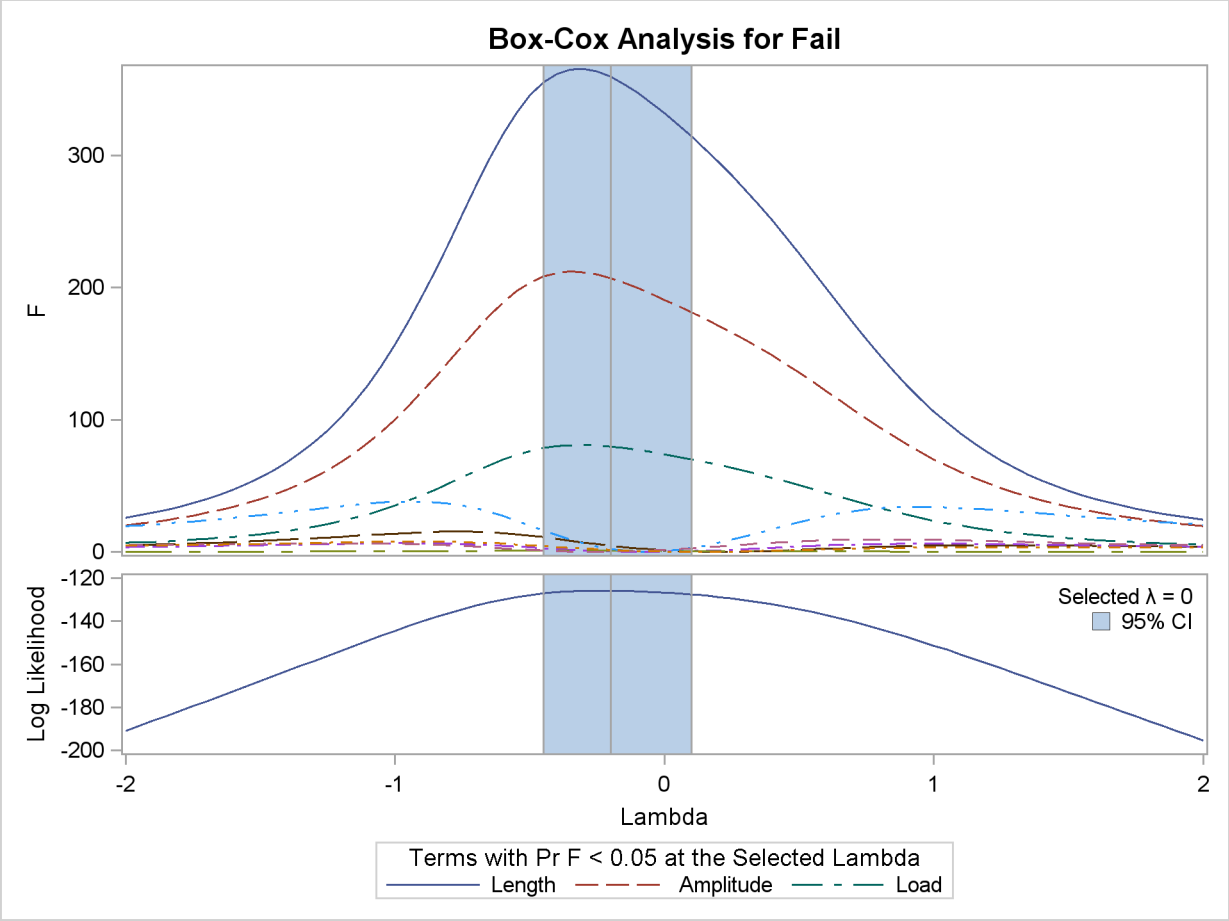
The measured response was time (in cycles) until failure. However, you could just as well have measured the inverse of time until failure (in other words, the failure rate). Hence, the correct metric with which to analyze the response is not apparent. You can use PROC TRANSREG to find an optimum power transformation for the analysis. The following statements create the input SAS data set:

```
title 'Yarn Strength';

proc format;
   value a -1 =   8 0 =   9 1 =  10;
   value l -1 = 250 0 = 300 1 = 350;
   value o -1 =  40 0 =  45 1 =  50;
run;

data yarn;
   input Fail Amplitude Length Load @@;
   format amplitude a. length l. load o.;
   label fail = 'Time in Cycles until Failure';
   datalines;
 674 -1 -1 -1    370 -1 -1  0    292 -1 -1  1    338  0 -1 -1
 266  0 -1  0    210  0 -1  1    170  1 -1 -1    118  1 -1  0
  90  1 -1  1   1414 -1  0 -1   1198 -1  0  0    634 -1  0  1
1022  0  0 -1    620  0  0  0    438  0  0  1    442  1  0 -1
 332  1  0  0    220  1  0  1   3636 -1  1 -1   3184 -1  1  0
2000 -1  1  1   1568  0  1 -1   1070  0  1  0    566  0  1  1
1140  1  1 -1    884  1  1  0    360  1  1  1
;
```

PROC TRANSREG is run to find the Box-Cox transformation. The lambda list is –2 TO 2 BY 0.05, which produces 81 lambdas, and a convenient lambda is requested. This many power parameters makes a nice graphical display with plenty of detail around the confidence interval. In the interest of space, only part of this table is displayed. The independent variables are designated with the QPOINT expansion. QPOINT, for quadratic point model, gets its name from PROC TRANSREG's ideal point modeling capabilities, which process variables for a response surface analysis. What QPOINT does is create a set of independent variables consisting of the following: the m original variables (Length Amplitude Load), the m original variables squared (Length_2 Amplitude_2 Load_2), and the $m \times (m-1)/2 = 3$ pairs of products between the m variables (LengthAmplitude LengthLoad AmplitudeLoad). The following statements produce Output 90.2.1:

```
ods graphics on;

proc transreg details data=yarn ss2
              plots=(transformation(dependent) obp);
   model BoxCox(fail / convenient lambda=-2 to 2 by 0.05) =
         qpoint(length amplitude load);
run;

ods graphics off;
```

Output 90.2.1 Box-Cox Yarn Data



Output 90.2.1 *continued*

```
Dependent Variable BoxCox(Fail)
  Time in Cycles until Failure


Number of Observations Read            27
Number of Observations Used            27
```

Output 90.2.1 *continued*

Model Statement Specification Details

Type	DF	Variable	Description	Value
Dep	1	BoxCox(Fail)	Lambda Used	0
			Lambda	-0.2
			Log Likelihood	-125.9
			Conv. Lambda	0
			Conv. Lambda LL	-126.7
			CI Limit	-127.8
			Alpha	0.05
			Options	Convenient Lambda Used
			Label	Time in Cycles until Failure
Ind	1	Qpoint.Length	DF	1
Ind	1	Qpoint.Amplitude	DF	1
Ind	1	Qpoint.Load	DF	1
Ind	1	Qpoint.Length_2	DF	1
Ind	1	Qpoint.Amplitude_2	DF	1
Ind	1	Qpoint.Load_2	DF	1
Ind	1	Qpoint.LengthAmplitude	DF	1
Ind	1	Qpoint.LengthLoad	DF	1
Ind	1	Qpoint.AmplitudeLoad	DF	1

The TRANSREG Procedure Hypothesis Tests for BoxCox(Fail)
Time in Cycles until Failure

Univariate ANOVA Table Based on the Usual Degrees of Freedom

Source	DF	Sum of Squares	Mean Square	F Value	Liberal p
Model	9	22.56498	2.507220	66.73	>= <.0001
Error	17	0.63871	0.037571		
Corrected Total	26	23.20369			

The above statistics are not adjusted for the fact that the dependent variable was transformed and so are generally liberal.

Root MSE	0.19383	R-Square	0.9725
Dependent Mean	6.33466	Adj R-Sq	0.9579
Coeff Var	3.05987	Lambda	0.0000

Output 90.2.1 *continued*

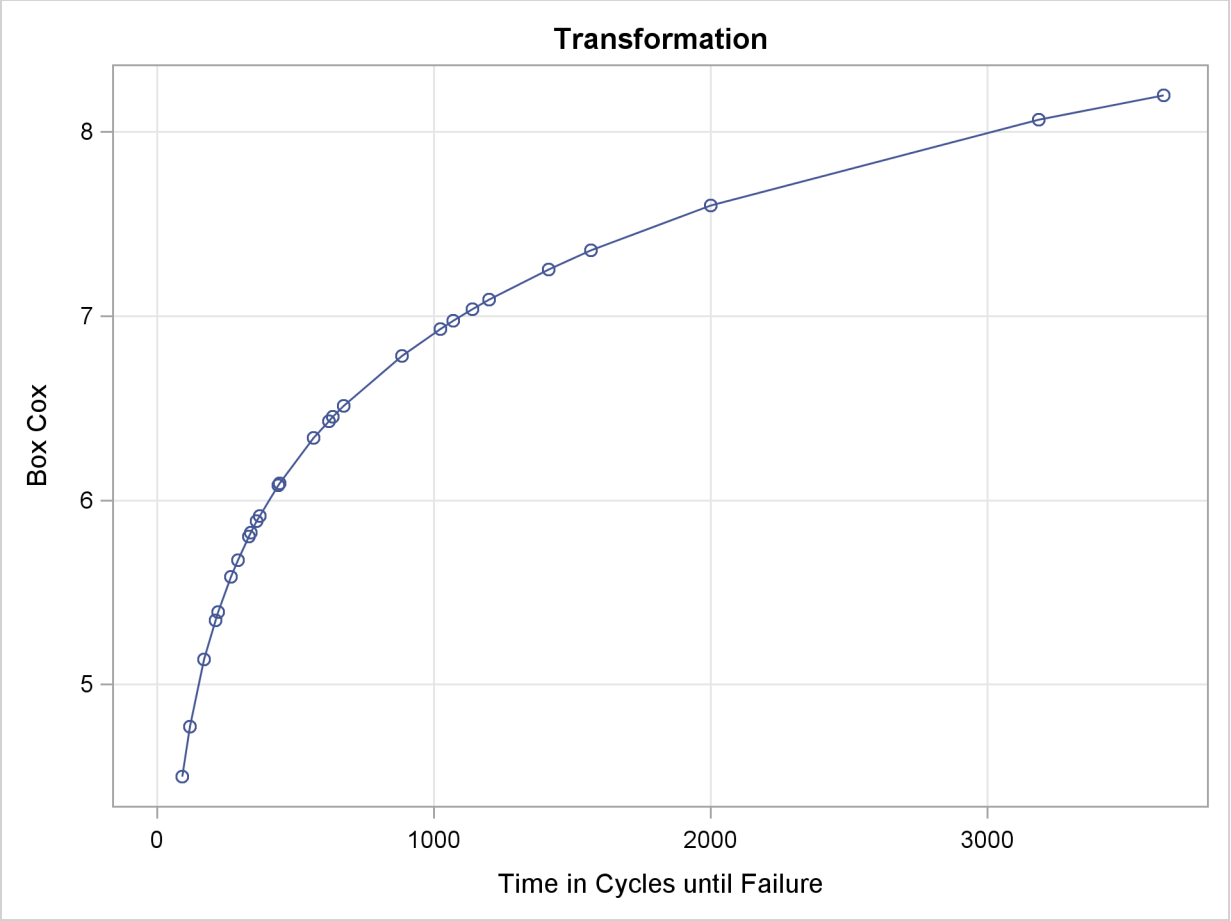
Univariate Regression Table Based on the Usual Degrees of Freedom

Variable	DF	Coefficient	Type II Sum of Squares	Mean Square	F Value	Liberal p
Intercept	1	6.4206207	159.008	159.008	4232.19	>= <.0001
Qpoint.Length	1	0.8323842	12.472	12.472	331.94	>= <.0001
Qpoint.Amplitude	1	−0.6309916	7.167	7.167	190.75	>= <.0001
Qpoint.Load	1	−0.3924940	2.773	2.773	73.80	>= <.0001
Qpoint.Length_2	1	−0.0856974	0.044	0.044	1.17	>= 0.2939
Qpoint.Amplitude_2	1	0.0242183	0.004	0.004	0.09	>= 0.7633
Qpoint.Load_2	1	−0.0674555	0.027	0.027	0.73	>= 0.4058
Qpoint.LengthAmplitude	1	−0.0382414	0.018	0.018	0.47	>= 0.5035
Qpoint.LengthLoad	1	−0.0684146	0.056	0.056	1.49	>= 0.2381
Qpoint.AmplitudeLoad	1	−0.0208340	0.005	0.005	0.14	>= 0.7142

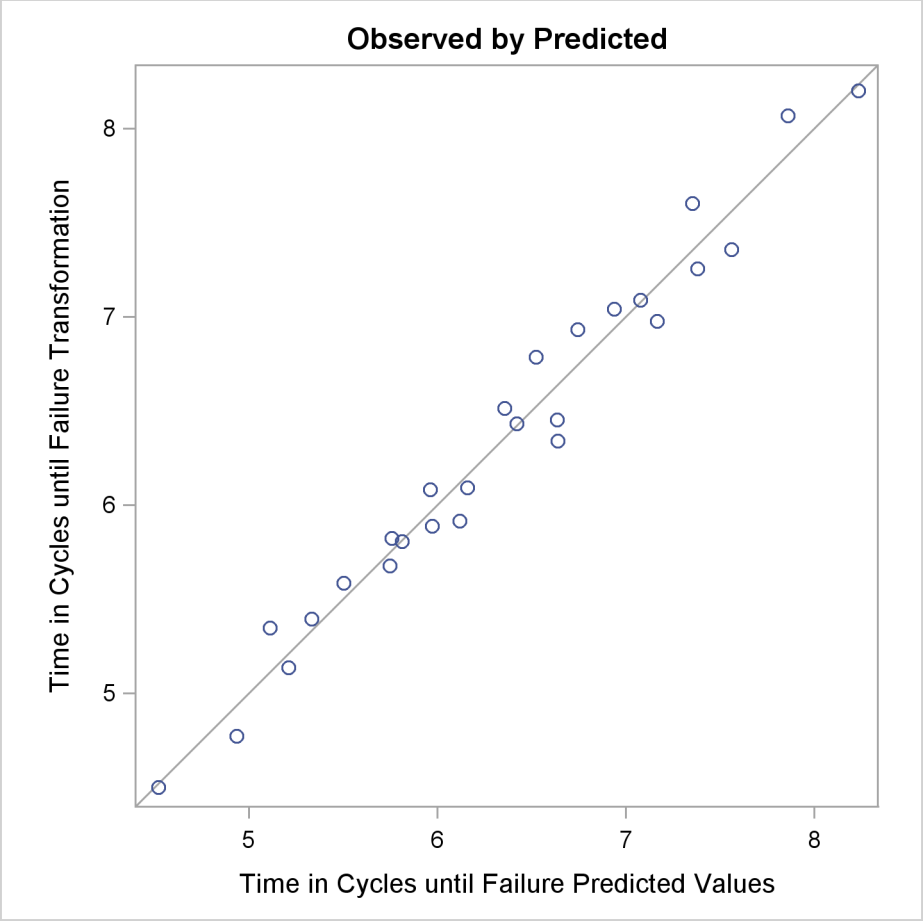
Variable	DF	Label
Intercept	1	Intercept
Qpoint.Length	1	Length
Qpoint.Amplitude	1	Amplitude
Qpoint.Load	1	Load
Qpoint.Length_2	1	Length_2
Qpoint.Amplitude_2	1	Amplitude_2
Qpoint.Load_2	1	Load_2
Qpoint.LengthAmplitude	1	LengthAmplitude
Qpoint.LengthLoad	1	LengthLoad
Qpoint.AmplitudeLoad	1	AmplitudeLoad

The above statistics are not adjusted for the fact that the dependent variable was transformed and so are generally liberal.

Output 90.2.1 *continued*



Output 90.2.1 *continued*



The optimal power parameter is –0.20, but since 0.0 is in the confidence interval, and since the CONVENIENT *t-option* was specified, the procedure chooses a log transformation. The $F = t^2$ plot shows in the vicinity of the optimal Box-Cox transformation, the parameters for the three original variables (Length Amplitude Load), particularly Length, are significant and the others become essentially zero.

Example 90.3: Penalized B-Spline

The following data set contains measurements of monthly averaged atmospheric pressure differences between Easter Island and Darwin, Australia, for a period of 168 months (National Institute of Standards and Technology 1998):

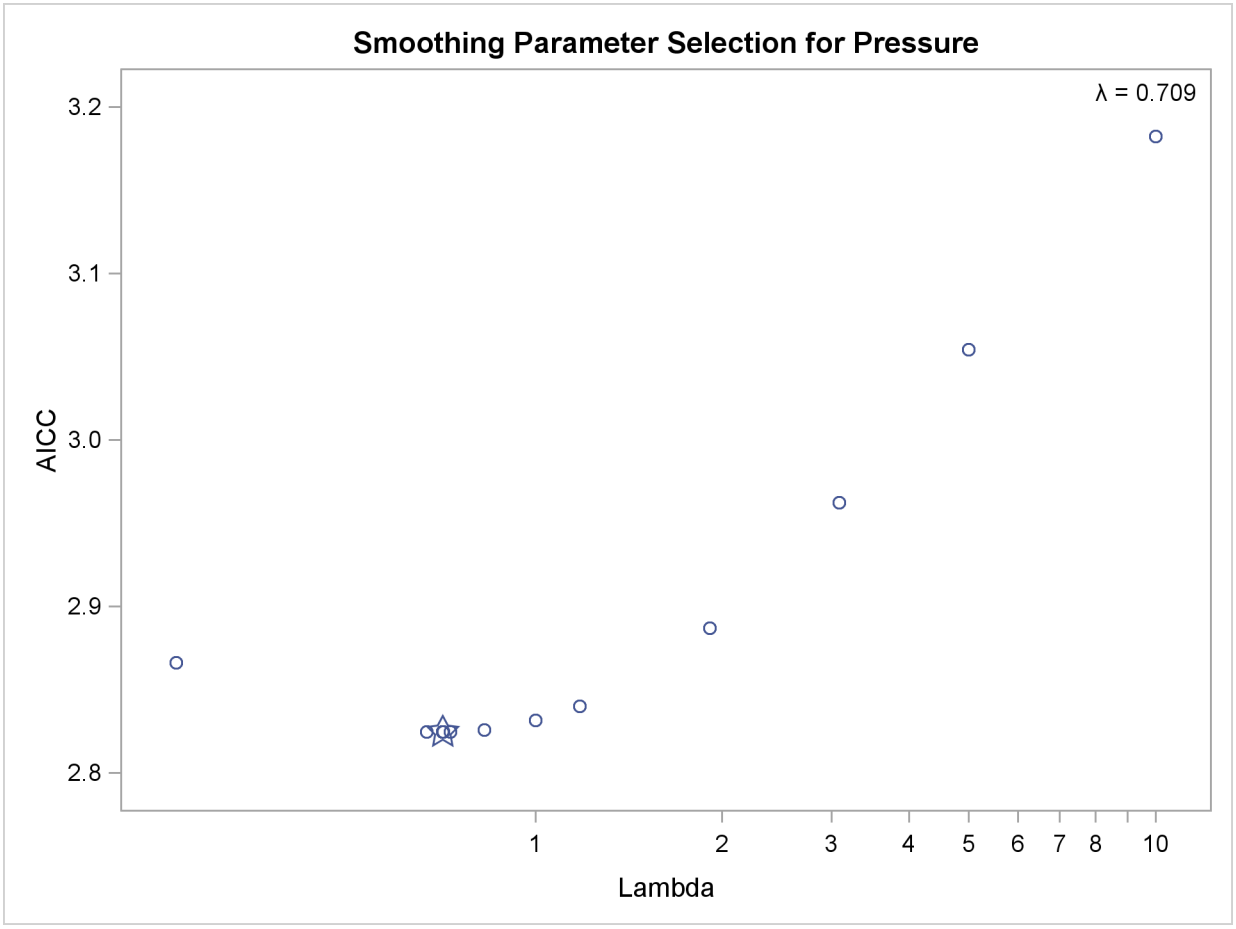
```
title 'Atmospheric Pressure Changes Between'
      ' Easter Island & Darwin, Australia';

data ENSO;
   input Pressure @@;
   Year = _n_ / 12;
   datalines;
12.9  11.3  10.6  11.2  10.9   7.5   7.7  11.7
12.9  14.3  10.9  13.7  17.1  14.0  15.3   8.5
 5.7   5.5   7.6   8.6   7.3   7.6  12.7  11.0
12.7  12.9  13.0  10.9  10.4  10.2   8.0  10.9
13.6  10.5   9.2  12.4  12.7  13.3  10.1   7.8
 4.8   3.0   2.5   6.3   9.7  11.6   8.6  12.4
10.5  13.3  10.4   8.1   3.7  10.7   5.1  10.4
10.9  11.7  11.4  13.7  14.1  14.0  12.5   6.3
 9.6  11.7   5.0  10.8  12.7  10.8  11.8  12.6
15.7  12.6  14.8   7.8   7.1  11.2   8.1   6.4
 5.2  12.0  10.2  12.7  10.2  14.7  12.2   7.1
 5.7   6.7   3.9   8.5   8.3  10.8  16.7  12.6
12.5  12.5   9.8   7.2   4.1  10.6  10.1  10.1
11.9  13.6  16.3  17.6  15.5  16.0  15.2  11.2
14.3  14.5   8.5  12.0  12.7  11.3  14.5  15.1
10.4  11.5  13.4   7.5   0.6   0.3   5.5   5.0
 4.6   8.2   9.9   9.2  12.5  10.9   9.9   8.9
 7.6   9.5   8.4  10.7  13.6  13.7  13.7  16.5
16.8  17.1  15.4   9.5   6.1  10.1   9.3   5.3
11.2  16.6  15.6  12.0  11.5   8.6  13.8   8.7
 8.6   8.6   8.7  12.8  13.2  14.0  13.4  14.8
;
```

You can fit a curve through these data by using a penalized B-spline (Eilers and Marx 1996) function and the following statements:

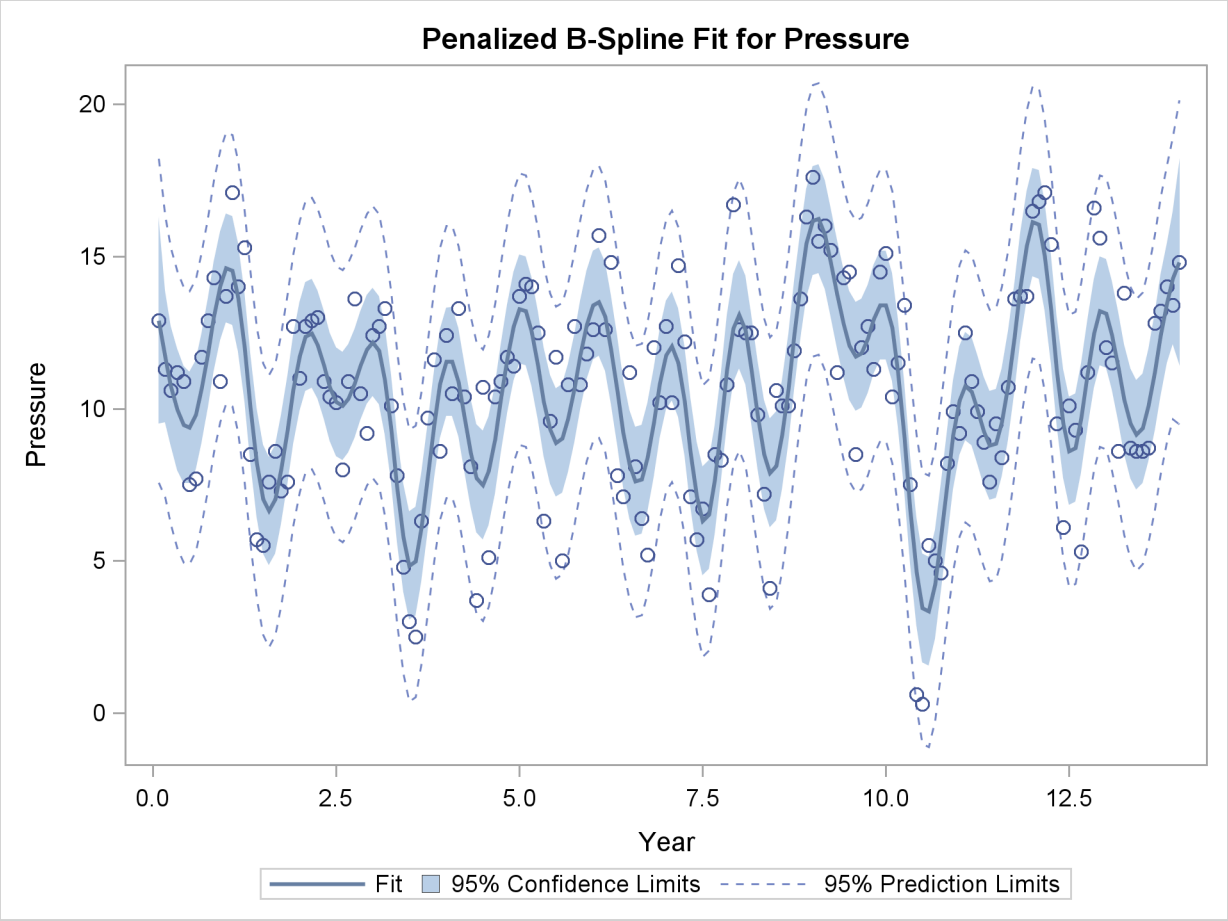
```
ods graphics on;

proc transreg data=enso;
   model identity(pressure) = pbspline(year);
run;
```

The dependent variable Pressure is specified along with an IDENTITY transformation, so Pressure is analyzed as is, with no transformations. The independent variable Year is specified with a PBSPLINE transformation, so a penalized B-spline model is fit. By default, a DEGREE=3 B-spline basis is used along with 100 evenly spaced knots and three evenly spaced exterior knots on each side of the data. The penalized spline function is typically much smoother than you would get by using a SPLINE transformation or a BSPLINE expansion since changes in the coefficients of the basis are penalized to make a smoother fit. The output is shown next in Output 90.3.1.

Output 90.3.1 Change in Atmospheric Pressure, AICC



Output 90.3.1 *continued*

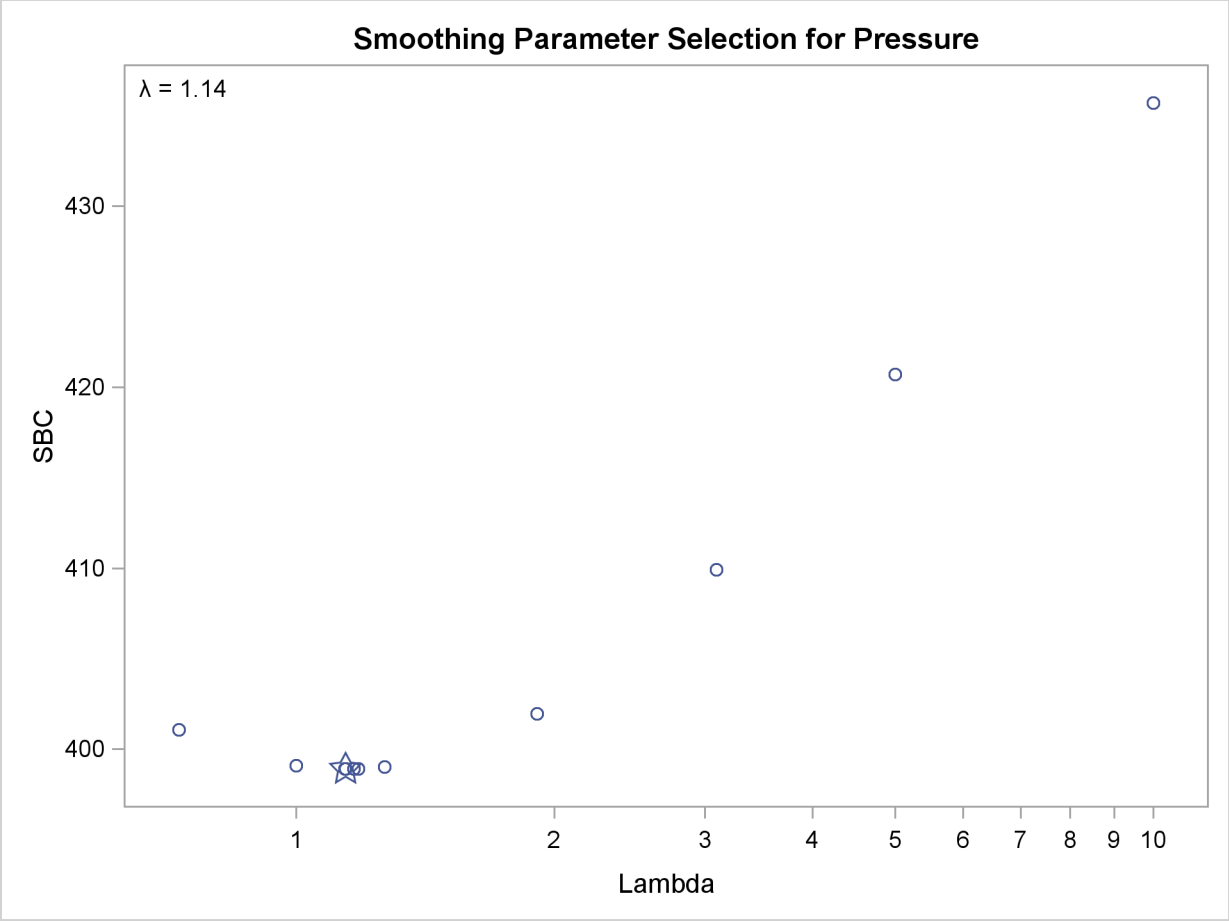


The results show a yearly cycle of pressure change. The procedure chose a smoothing parameter of $\lambda = 0.709$. With data such as these, with many peaks and valleys, it might be useful to perform another analysis, this time asking for a smoother plot. The Schwarz Bayesian criterion (SBC) is sometimes a better choice than the default criterion when you want a smoother plot. The following PROC TRANSREG step requests a penalized B-spline analysis minimizing the SBC criterion:

```
proc transreg data=enso;
   model identity(pressure) = pbspline(year / sbc);
run;
```

The plot of SBC as a function of λ is shown in Output 90.3.2.

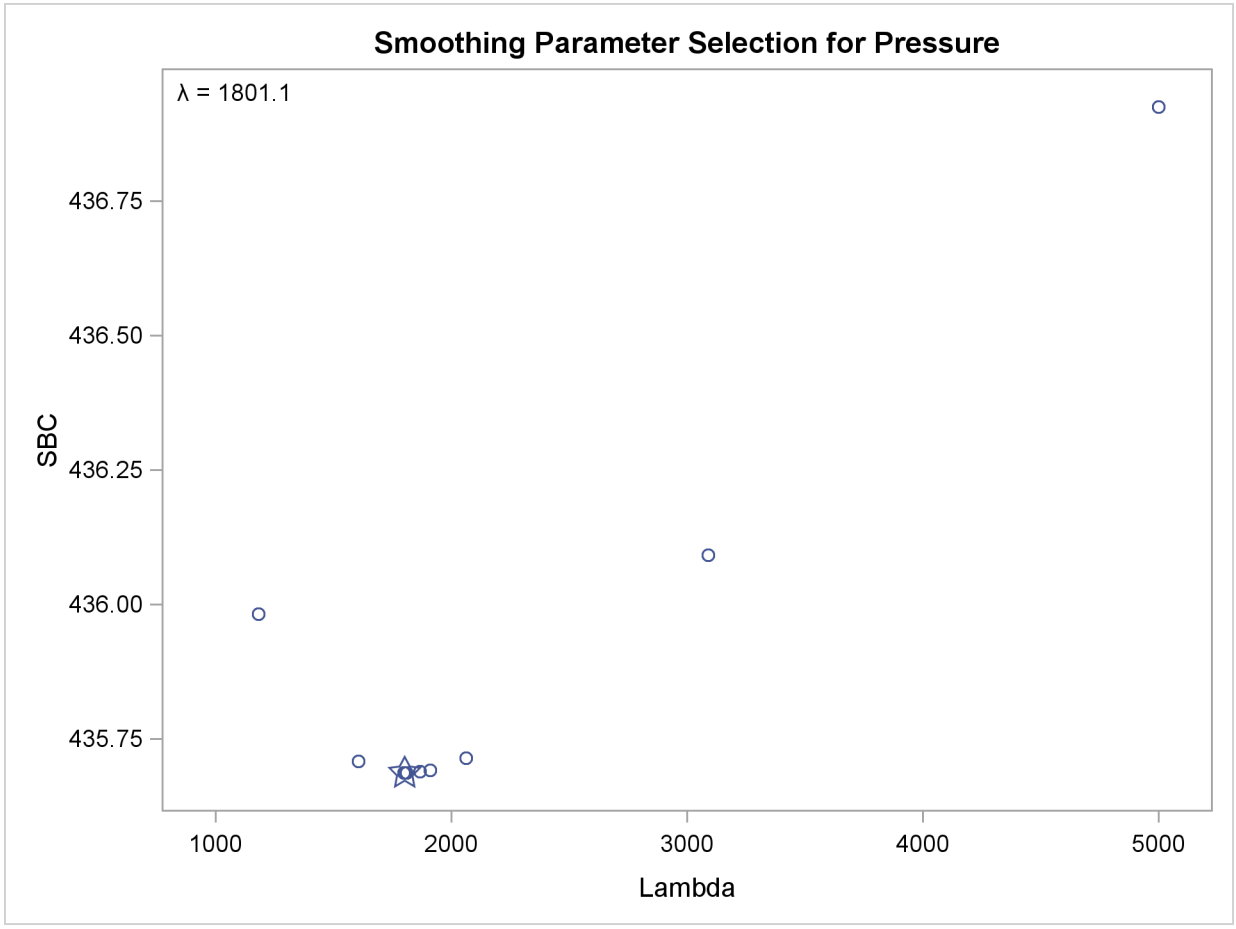
Output 90.3.2 Change in Atmospheric Pressure, SBC



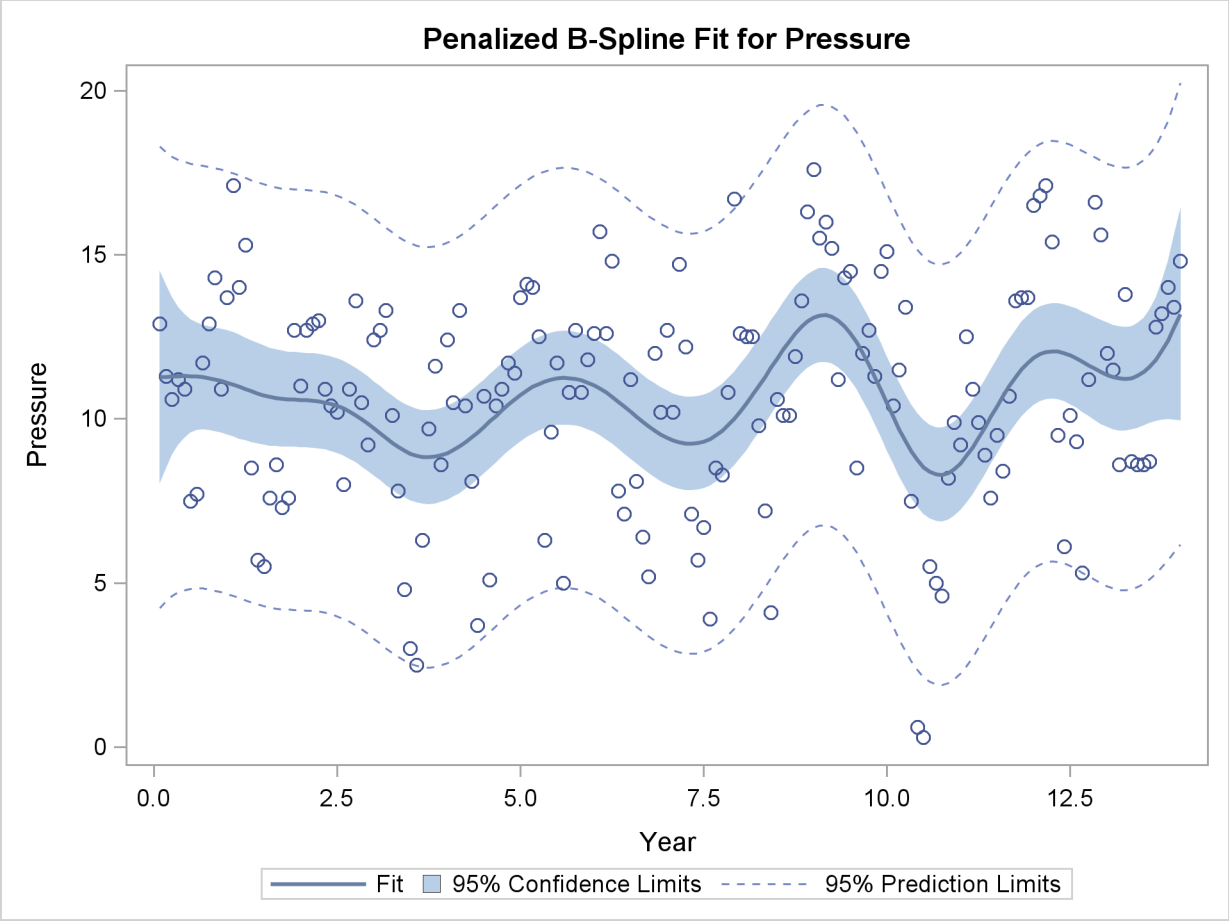
The fit plot (not shown) is essentially the same as the one shown in Output 90.3.1 due to the similar choice of smoothing parameters: $\lambda = 0.709$ versus $\lambda = 1.14$. You can analyze these data again, this time forcing PROC TRANSREG to consider only larger smoothing parameters. The specification LAMBDA=2 10000 RANGE eliminates from consideration the two lambdas that you previously saw and considers only $2 \leq \lambda \leq 10,000$. The following statements produce Output 90.3.3:

```
proc transreg data=enso;
   model identity(pressure) = pbspline(year / sbc lambda=2 10000 range);
run;
```

Output 90.3.3 Change in Atmospheric Pressure, SBC, Lambda > 1



Output 90.3.3 *continued*



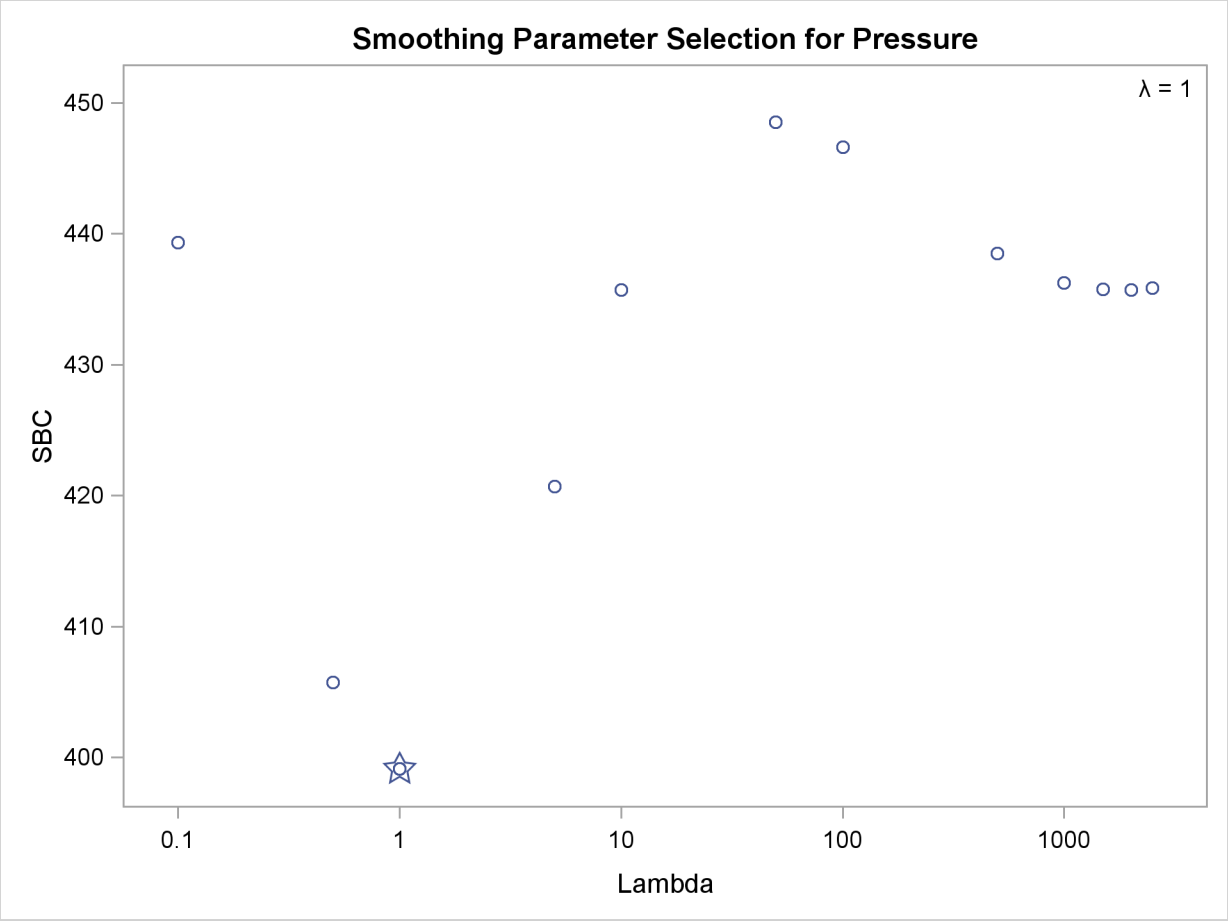
The results clearly show that there is a local minimum in the SBC(λ) function at $\lambda = 1801.1$. Using this lambda results in a much smoother regression function with a longer cycle than you saw previously. This second cycle can be identified as the periodic warming of the Pacific Ocean known as "El Niño." The SBC(λ) function has at least two minima since there are at least two trends in the data. In the first analysis, PROC TRANSREG found what is probably the globally optimal solution, and in the second set of analyses, with a little nudging away from the global optimum, it found a very interesting locally optimal solution.

You can specify a list of lambdas to see SBC as a function of lambda over the range that includes both minima as follows:

```
proc transreg data=enso;
   model identity(pressure) = pbspline(year / sbc lambda=.1 .5 1 5
                                          10 50 100 500 to 2500 by 500);
   run;

ods graphics off;
```

The plot of SBC as a function of λ is shown in Output 90.3.4.

Output 90.3.4 Change in Atmospheric Pressure, SBC, Over the Range of Both Minima



Example 90.4: Nonmetric Conjoint Analysis of Tire Data

This example uses PROC TRANSREG to perform a nonmetric conjoint analysis of tire preference data. Conjoint analysis decomposes rank-ordered evaluation judgments of products or services into components based on qualitative product attributes. For each level of each attribute of interest, a numerical "part-worth utility" value is computed. The sum of the part-worth utilities for each product is an estimate of the utility for that product. The goal is to compute part-worth utilities such that the product utilities are as similar as possible to the original rank ordering. (This example is a greatly simplified introductory example.)

The stimuli for the experiment are 18 hypothetical tires. The stimuli represent different brands (Goodstone, Pirogi, Machismo),[1] prices ($69.99, $74.99, $79.99), expected tread life (50,000, 60,000, 70,000 miles), and road hazard insurance plans (Yes, No). There are $3 \times 3 \times 3 \times 2 = 54$ possible combinations. From these, 18 combinations are selected that form an efficient experimental design for a main-effects model. The combinations are then ranked from 1 (most preferred) to 18 (least preferred). In this simple example, there is one set of rankings. A real conjoint study would have many more.

[1] In real conjoint experiments, real brand names would be used.

First, the FORMAT procedure is used to specify the meanings of the factor levels, which are entered as numbers in the DATA step along with the ranks. PROC TRANSREG is used to perform the conjoint analysis. A maximum of 50 iterations is requested. The specification **monotone(Rank / reflect)** in the MODEL statement requests that the dependent variable Rank should be monotonically transformed and reflected so that positive utilities mean high preference. The variables Brand, Price, Life, and Hazard are designated as CLASS variables, and the part-worth utilities are constrained by ZERO=SUM to sum to zero within each factor. The UTILITIES *a-option* displays the conjoint analysis results.

The importance column of the utilities table shows that price is the most important attribute in determining preference (57%), followed by expected tread life (18%), brand (15%), and road hazard insurance (10%). Looking at the utilities table for the maximum part-worth utility within each attribute, you see from the results that the most preferred combination is Pirogi brand tires, at $69.99, with a 70,000-mile expected tread life and road hazard insurance. This product is not actually in the data set. The sum of the part-worth utilities for this combination is as follows:

$$20.64 = 9.50 + 1.90 + 5.87 + 2.41 + 0.96$$

The following statements produce Output 90.4.1.

```
title 'Nonmetric Conjoint Analysis of Ranks';

proc format;
   value BrandF
               1 = 'Goodstone'
               2 = 'Pirogi   '
               3 = 'Machismo ';
   value PriceF
               1 = '$69.99'
               2 = '$74.99'
               3 = '$79.99';
   value LifeF
               1 = '50,000'
               2 = '60,000'
               3 = '70,000';
   value HazardF
               1 = 'Yes'
               2 = 'No ';
run;
```

```
data Tires;
   input Brand Price Life Hazard Rank;
   format Brand BrandF9. Price PriceF9. Life LifeF6. Hazard HazardF3.;
   datalines;
1 1 2 1  3
1 1 3 2  2
1 2 1 2 14
1 2 2 2 10
1 3 1 1 17
1 3 3 1 12
2 1 1 2  7
2 1 3 2  1
2 2 1 1  8
2 2 3 1  5
2 3 2 1 13
2 3 2 2 16
3 1 1 1  6
3 1 2 1  4
3 2 2 2 15
3 2 3 1  9
3 3 1 2 18
3 3 3 2 11
;

proc transreg maxiter=50 utilities short;
   ods select TestsNote ConvergenceStatus FitStatistics Utilities;
   model monotone(Rank / reflect) =
         class(Brand Price Life Hazard / zero=sum);
   output ireplace predicted;
run;

proc print label;
   var Rank TRank PRank Brand Price Life Hazard;
   label PRank = 'Predicted Ranks';
run;
```

Output 90.4.1 Simple Conjoint Analysis

```
                    Nonmetric Conjoint Analysis of Ranks

                          The TRANSREG Procedure

   Monotone(Rank)
   Algorithm converged.

          The TRANSREG Procedure Hypothesis Tests for Monotone(Rank)


               Root MSE           0.49759    R-Square     0.9949
               Dependent Mean     9.50000    Adj R-Sq     0.9913
               Coeff Var          5.23783
```

Output 90.4.1 *continued*

Utilities Table Based on the Usual Degrees of Freedom

Label	Utility	Standard Error	Importance (% Utility Range)	Variable
Intercept	9.5000	0.11728		Intercept
Brand Goodstone	−1.1718	0.16586	15.463	Class.BrandGoodstone
Brand Pirogi	1.8980	0.16586		Class.BrandPirogi
Brand Machismo	−0.7262	0.16586		Class.BrandMachismo
Price $69.99	5.8732	0.16586	56.517	Class.Price_69_99
Price $74.99	−0.5261	0.16586		Class.Price_74_99
Price $79.99	−5.3471	0.16586		Class.Price_79_99
Life 50,000	−1.2350	0.16586	18.361	Class.Life50_000
Life 60,000	−1.1751	0.16586		Class.Life60_000
Life 70,000	2.4101	0.16586		Class.Life70_000
Hazard Yes	0.9588	0.11728	9.659	Class.HazardYes
Hazard No	−0.9588	0.11728		Class.HazardNo

The standard errors are not adjusted for the fact that the dependent variable was transformed and so are generally liberal (too small).

Output 90.4.1 *continued*

Nonmetric Conjoint Analysis of Ranks

Obs	Rank	Rank Transformation	Predicted Ranks	Brand	Price	Life	Hazard
1	3	14.4462	13.9851	Goodstone	$69.99	60,000	Yes
2	2	15.6844	15.6527	Goodstone	$69.99	70,000	No
3	14	5.7229	5.6083	Goodstone	$74.99	50,000	No
4	10	5.7229	5.6682	Goodstone	$74.99	60,000	No
5	17	2.6699	2.7049	Goodstone	$79.99	50,000	Yes
6	12	5.7229	6.3500	Goodstone	$79.99	70,000	Yes
7	7	14.4462	15.0774	Pirogi	$69.99	50,000	No
8	1	18.7699	18.7225	Pirogi	$69.99	70,000	No
9	8	11.1143	10.5957	Pirogi	$74.99	50,000	Yes
10	5	14.4462	14.2408	Pirogi	$74.99	70,000	Yes
11	13	5.7229	5.8346	Pirogi	$79.99	60,000	Yes
12	16	3.8884	3.9170	Pirogi	$79.99	60,000	No
13	6	14.4462	14.3708	Machismo	$69.99	50,000	Yes
14	4	14.4462	14.4307	Machismo	$69.99	60,000	Yes
15	15	5.7229	6.1139	Machismo	$74.99	60,000	No
16	9	11.1143	11.6166	Machismo	$74.99	70,000	Yes
17	18	1.1905	1.2330	Machismo	$79.99	50,000	No
18	11	5.7229	4.8780	Machismo	$79.99	70,000	No

Example 90.5: Metric Conjoint Analysis of Tire Data

This example, which is more detailed than the previous one, uses PROC TRANSREG to perform a metric conjoint analysis of tire preference data. Conjoint analysis can be used to decompose preference ratings of products or services into components based on qualitative product attributes. For each level of each attribute of interest, a numerical "part-worth utility" value is computed. The sum of the part-worth utilities for each product is an estimate of the utility for that product. The goal is to compute part-worth utilities such that the product utilities are as similar as possible to the original ratings. Metric conjoint analysis, as shown in this example, fits an ordinary linear model directly to data assumed to be measured on an interval scale. Nonmetric conjoint analysis, as shown in Example 90.4, finds an optimal monotonic transformation of original data before fitting an ordinary linear model to the transformed data.

This example has three parts. In the first part, an experimental design is created. In the second part, a DATA step creates descriptions of the stimuli for the experiment. The third part of the example performs the conjoint analyses.

The stimuli for the experiment are 18 hypothetical tires. The stimuli represent different brands (Goodstone, Pirogi, Machismo),[2] prices ($69.99, $74.99, $79.99), expected tread life (50,000, 60,000, 70,000 miles), and road hazard insurance plans (Yes, No).

For a conjoint study such as this, you need to create an experimental design with 3 three-level factors, 1 two-level factor, and 18 combinations or *runs*. The easiest way to get this design is with the %MktEx autocall macro. The %MktEx macro requires you to specify the number of levels of each of the four factors, followed by N=18, the number of runs. Specifying a random number seed, while not strictly necessary, helps ensure that the design is reproducible. The %MktLab macro assigns the actual factor names instead of the default names x1, x2, and so on, and it assigns formats to the factor levels. The %MktEval macro helps you evaluate the design. It shows how correlated or independent the factors are, how often each factor level appears in the design, how often each pair occurs for every factor pair, and how often each product profile or run occurs in the design. See http://support.sas.com/resources/papers/tnote/tnote_marketresearch.html (Kuhfeld 2005) for more information about experimental design and conjoint analysis; look for the latest "Conjoint Analysis" report. The following statements create, evaluate, and display the design:

```
title 'Tire Study, Experimental Design';

proc format;
   value BrandF
             1 = 'Goodstone'
             2 = 'Pirogi   '
             3 = 'Machismo ';
   value PriceF
             1 = '$69.99'
             2 = '$74.99'
             3 = '$79.99';
   value LifeF
             1 = '50,000'
```

[2] In real conjoint experiments, real brand names would be used.

```
                2 = '60,000'
                3 = '70,000';
   value HazardF
                1 = 'Yes'
                2 = 'No ';
run;

%mktex(3 3 3 2, n=18, seed=448)

%mktlab(vars=Brand Price Life Hazard, out=sasuser.TireDesign,
        statements=format Brand BrandF9. Price PriceF9.
                   Life LifeF6. Hazard HazardF3.)

%mkteval;

proc print data=sasuser.TireDesign;
run;
```

The %MktEx macro (Kuhfeld 2005) output displayed in Output 90.5.1 shows you that the design is 100% efficient, which means it is orthogonal and balanced. The %MktEval macro output displayed in Output 90.5.2 shows you that all of the factors are uncorrelated or orthogonal, the design is balanced (each level occurs once), and every pair of factor levels occurs equally often (again showing that the design is orthogonal). The *n*-way frequencies show that each product profile occurs once (there are no duplicates). The design is shown in Output 90.5.3. The design is automatically randomized (the profiles were sorted into a random order and the original levels are randomly reassigned). Orthogonality, balance, randomization, and other design concepts are discussed in detail in Kuhfeld (2005), in the "Experimental Design, Efficiency, Coding, and Choice Designs" report.

Output 90.5.1 Tire Study, Design Efficiency

Tire Study, Experimental Design

Algorithm Search History

Tire Study, Experimental Design

Design	Row,Col	Current D-Efficiency	Best D-Efficiency	Notes
1	Start	100.0000	100.0000	Tab
1	End	100.0000		

Output 90.5.1 *continued*

```
                    Tire Study, Experimental Design

                         The OPTEX Procedure

                       Class Level Information

                        Class  Levels  Values

                        x1        3     1 2 3
                        x2        3     1 2 3
                        x3        3     1 2 3
                        x4        2     1 2

                    Tire Study, Experimental Design

                                                                  Average
                                                                Prediction
Design                                                          Standard
Number      D-Efficiency      A-Efficiency      G-Efficiency       Error
-------------------------------------------------------------------------
     1       100.0000          100.0000          100.0000         0.6667
```

Output 90.5.2 Tire Study, Design Evaluation

```
                    Tire Study, Experimental Design
                 Canonical Correlations Between the Factors
           There are 0 Canonical Correlations Greater Than 0.316

                          Brand    Price    Life    Hazard

                Brand     1        0        0        0
                Price     0        1        0        0
                Life      0        0        1        0
                Hazard    0        0        0        1

                    Tire Study, Experimental Design
                         Summary of Frequencies
           There are 0 Canonical Correlations Greater Than 0.316

                              Frequencies

            Brand             6 6 6
            Price             6 6 6
            Life              6 6 6
            Hazard            9 9
            Brand Price       2 2 2 2 2 2 2 2 2
            Brand Life        2 2 2 2 2 2 2 2 2
            Brand Hazard      3 3 3 3 3 3
            Price Life        2 2 2 2 2 2 2 2 2
            Price Hazard      3 3 3 3 3 3
            Life Hazard       3 3 3 3 3 3
            N-Way             1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

Output 90.5.3 Tire Study, Design

Tire Study, Experimental Design

Obs	Brand	Price	Life	Hazard
1	Pirogi	$79.99	50,000	No
2	Machismo	$79.99	60,000	No
3	Machismo	$74.99	70,000	Yes
4	Machismo	$74.99	50,000	No
5	Goodstone	$74.99	60,000	Yes
6	Pirogi	$69.99	60,000	Yes
7	Goodstone	$69.99	50,000	Yes
8	Machismo	$69.99	50,000	Yes
9	Pirogi	$74.99	60,000	Yes
10	Pirogi	$74.99	50,000	No
11	Goodstone	$79.99	60,000	No
12	Goodstone	$69.99	70,000	No
13	Pirogi	$79.99	70,000	Yes
14	Goodstone	$74.99	70,000	No
15	Machismo	$69.99	60,000	No
16	Machismo	$79.99	70,000	Yes
17	Pirogi	$69.99	70,000	No
18	Goodstone	$79.99	50,000	Yes

Next, the questionnaires are printed and given to the subjects, who are asked to rate the tires.

The following statements produce Output 90.5.4:

```
data _null_;
   title;
   set sasuser.TireDesign;
   file print;
   if mod(_n_,4) eq 1 then do;
      put _page_;
      put +55 'Subject _________';
   end;
   length hazardstring $ 7.;
   if put(hazard, hazardf3.) = 'Yes'
      then hazardstring = 'with';
      else hazardstring = 'without';

   s = 3 + (_n_ >= 10);
   put // _n_ +(-1) ') For your next tire purchase, '
          'how likely are you to buy this product?'
       // +s Brand 'brand tires at ' Price +(-1) ','
       /  +s 'with a ' Life 'tread life guarantee, '
       /  +s 'and ' hazardstring 'road hazard insurance.'
       // +s 'Definitely Would                  Definitely Would'
       /  +s 'Not Purchase                             Purchase'
       // +s '1     2     3     4     5     6     7     8     9 ';
run;
```

This output in Output 90.5.4 is abbreviated in the interest of conserving space; the statements actually produce stimuli for all combinations.

Output 90.5.4 Conjoint Analysis, Stimuli Descriptions

```
                                                    Subject _________


1) For your next tire purchase, how likely are you to buy this product?

   Pirogi brand tires at $79.99,
   with a 50,000 tread life guarantee,
   and without road hazard insurance.

   Definitely Would                 Definitely Would
   Not Purchase                             Purchase

   1     2     3     4     5     6     7     8     9


2) For your next tire purchase, how likely are you to buy this product?

   Machismo brand tires at $79.99,
   with a 60,000 tread life guarantee,
   and without road hazard insurance.

   Definitely Would                 Definitely Would
   Not Purchase                             Purchase

   1     2     3     4     5     6     7     8     9


3) For your next tire purchase, how likely are you to buy this product?

   Machismo brand tires at $74.99,
   with a 70,000 tread life guarantee,
   and with road hazard insurance.

   Definitely Would                 Definitely Would
   Not Purchase                             Purchase

   1     2     3     4     5     6     7     8     9


4) For your next tire purchase, how likely are you to buy this product?

   Machismo brand tires at $74.99,
   with a 50,000 tread life guarantee,
   and without road hazard insurance.

   Definitely Would                 Definitely Would
   Not Purchase                             Purchase

   1     2     3     4     5     6     7     8     9
```

The third part of the example performs the conjoint analyses. The DATA step reads the data. Only the ratings are entered, one row per subject. Real conjoint studies have many more subjects than five. The TRANSPOSE procedure transposes this (5×18) data set into an (18×5) data set that can be merged with the factor level data set sasuser.TireDesign. The next DATA step does the merge. The PRINT procedure displays the input data set.

PROC TRANSREG fits the five individual conjoint models, one for each subject. The UTILITIES *a-option* displays the conjoint analysis results. The SHORT *a-option* suppresses the iteration histories, OUTTEST=UTILS creates an output data set with all of the conjoint results, and the SEPARATORS= option requests that the labels constructed for each category contain two blanks between the variable name and the level value. The ODS SELECT statement is used to limit the displayed output. The MODEL statement specifies IDENTITY for the ratings, which specifies a metric conjoint analysis—the ratings are not transformed. The variables Brand, Price, Life, and Hazard are designated as CLASS variables, and the part-worth utilities are constrained to sum to zero within each factor.

The following statements produce Output 90.5.5:

```
title 'Tire Study, Data Entry, Preprocessing';

data Results;
   input (c1-c18) (1.);
   datalines;
233279766526376493
124467885349168274
262189456534275794
184396375364187754
133379775526267493
;

* Create an Object by Subject Data Matrix;
proc transpose data=Results out=Results(drop=_name_) prefix=Subj;
run;

* Merge the Factor Levels with the Data Matrix;
data Both;
   merge sasuser.TireDesign Results;
run;

proc print;
   title2 'Data Set for Conjoint Analysis';
run;

title 'Tire Study, Individual Conjoint Analyses';

* Fit Each Subject Individually;
proc transreg data=Both utilities short outtest=utils separators='  ';
   ods select TestsNote FitStatistics Utilities;
   model identity(Subj1-Subj5) =
         class(Brand Price Life Hazard / zero=sum);
run;
```

The output contains two tables per subject, one with overall fit statistics and one with the conjoint analysis results.

Output 90.5.5 Conjoint Analysis

Tire Study, Data Entry, Preprocessing
Data Set for Conjoint Analysis

Obs	Brand	Price	Life	Hazard	Subj1	Subj2	Subj3	Subj4	Subj5
1	Pirogi	$79.99	50,000	No	2	1	2	1	1
2	Machismo	$79.99	60,000	No	3	2	6	8	3
3	Machismo	$74.99	70,000	Yes	3	4	2	4	3
4	Machismo	$74.99	50,000	No	2	4	1	3	3
5	Goodstone	$74.99	60,000	Yes	7	6	8	9	7
6	Pirogi	$69.99	60,000	Yes	9	7	9	6	9
7	Goodstone	$69.99	50,000	Yes	7	8	4	3	7
8	Machismo	$69.99	50,000	Yes	6	8	5	7	7
9	Pirogi	$74.99	60,000	Yes	6	5	6	5	5
10	Pirogi	$74.99	50,000	No	5	3	5	3	5
11	Goodstone	$79.99	60,000	No	2	4	3	6	2
12	Goodstone	$69.99	70,000	No	6	9	4	4	6
13	Pirogi	$79.99	70,000	Yes	3	1	2	1	2
14	Goodstone	$74.99	70,000	No	7	6	7	8	6
15	Machismo	$69.99	60,000	No	6	8	5	7	7
16	Machismo	$79.99	70,000	Yes	4	2	7	7	4
17	Pirogi	$69.99	70,000	No	9	7	9	5	9
18	Goodstone	$79.99	50,000	Yes	3	4	4	4	3

Output 90.5.5 *continued*

Tire Study, Individual Conjoint Analyses

The TRANSREG Procedure

The TRANSREG Procedure Hypothesis Tests for Identity(Subj1)

Root MSE	1.34164	R-Square	0.8043
Dependent Mean	5.00000	Adj R-Sq	0.6674
Coeff Var	26.83282		

Output 90.5.5 *continued*

Utilities Table Based on the Usual Degrees of Freedom

Label		Utility	Standard Error	Importance (% Utility Range)	Variable
Intercept		5.0000	0.31623		Intercept
Brand	Goodstone	0.3333	0.44721	20.833	Class.BrandGoodstone
Brand	Pirogi	0.6667	0.44721		Class.BrandPirogi
Brand	Machismo	−1.0000	0.44721		Class.BrandMachismo
Price	$69.99	2.1667	0.44721	54.167	Class.Price_69_99
Price	$74.99	0.0000	0.44721		Class.Price_74_99
Price	$79.99	−2.1667	0.44721		Class.Price_79_99
Life	50,000	−0.8333	0.44721	16.667	Class.Life50_000
Life	60,000	0.5000	0.44721		Class.Life60_000
Life	70,000	0.3333	0.44721		Class.Life70_000
Hazard	Yes	0.3333	0.31623	8.333	Class.HazardYes
Hazard	No	−0.3333	0.31623		Class.HazardNo

Tire Study, Individual Conjoint Analyses

The TRANSREG Procedure

The TRANSREG Procedure Hypothesis Tests for Identity(Subj2)

Root MSE	0.56765	R-Square	0.9710
Dependent Mean	4.94444	Adj R-Sq	0.9506
Coeff Var	11.48049		

Utilities Table Based on the Usual Degrees of Freedom

Label		Utility	Standard Error	Importance (% Utility Range)	Variable
Intercept		4.9444	0.13380		Intercept
Brand	Goodstone	1.2222	0.18922	25.658	Class.BrandGoodstone
Brand	Pirogi	−0.9444	0.18922		Class.BrandPirogi
Brand	Machismo	−0.2778	0.18922		Class.BrandMachismo
Price	$69.99	2.8889	0.18922	65.132	Class.Price_69_99
Price	$74.99	−0.2778	0.18922		Class.Price_74_99
Price	$79.99	−2.6111	0.18922		Class.Price_79_99
Life	50,000	−0.2778	0.18922	7.895	Class.Life50_000
Life	60,000	0.3889	0.18922		Class.Life60_000
Life	70,000	−0.1111	0.18922		Class.Life70_000
Hazard	Yes	0.0556	0.13380	1.316	Class.HazardYes
Hazard	No	−0.0556	0.13380		Class.HazardNo

Output 90.5.5 *continued*

Tire Study, Individual Conjoint Analyses

The TRANSREG Procedure

The TRANSREG Procedure Hypothesis Tests for Identity(Subj3)

Root MSE	2.48104	R-Square	0.3902
Dependent Mean	4.94444	Adj R-Sq	-0.0367
Coeff Var	50.17832		

Utilities Table Based on the Usual Degrees of Freedom

Label		Utility	Standard Error	Importance (% Utility Range)	Variable
Intercept		4.9444	0.58479		Intercept
Brand	Goodstone	0.0556	0.82701	18.261	Class.BrandGoodstone
Brand	Pirogi	0.5556	0.82701		Class.BrandPirogi
Brand	Machismo	-0.6111	0.82701		Class.BrandMachismo
Price	$69.99	1.0556	0.82701	31.304	Class.Price_69_99
Price	$74.99	-0.1111	0.82701		Class.Price_74_99
Price	$79.99	-0.9444	0.82701		Class.Price_79_99
Life	50,000	-1.4444	0.82701	41.739	Class.Life50_000
Life	60,000	1.2222	0.82701		Class.Life60_000
Life	70,000	0.2222	0.82701		Class.Life70_000
Hazard	Yes	0.2778	0.58479	8.696	Class.HazardYes
Hazard	No	-0.2778	0.58479		Class.HazardNo

Tire Study, Individual Conjoint Analyses

The TRANSREG Procedure

The TRANSREG Procedure Hypothesis Tests for Identity(Subj4)

Root MSE	1.90321	R-Square	0.6185
Dependent Mean	5.05556	Adj R-Sq	0.3514
Coeff Var	37.64598		

Output 90.5.5 *continued*

Utilities Table Based on the Usual Degrees of Freedom

Label		Utility	Standard Error	Importance (% Utility Range)	Variable
Intercept		5.0556	0.44859		Intercept
Brand	Goodstone	0.6111	0.63440	36.885	Class.BrandGoodstone
Brand	Pirogi	−1.5556	0.63440		Class.BrandPirogi
Brand	Machismo	0.9444	0.63440		Class.BrandMachismo
Price	$69.99	0.2778	0.63440	12.295	Class.Price_69_99
Price	$74.99	0.2778	0.63440		Class.Price_74_99
Price	$79.99	−0.5556	0.63440		Class.Price_79_99
Life	50,000	−1.5556	0.63440	49.180	Class.Life50_000
Life	60,000	1.7778	0.63440		Class.Life60_000
Life	70,000	−0.2222	0.63440		Class.Life70_000
Hazard	Yes	0.0556	0.44859	1.639	Class.HazardYes
Hazard	No	−0.0556	0.44859		Class.HazardNo

Tire Study, Individual Conjoint Analyses

The TRANSREG Procedure

The TRANSREG Procedure Hypothesis Tests for Identity(Subj5)

Root MSE	1.36219	R-Square	0.8162
Dependent Mean	4.94444	Adj R-Sq	0.6875
Coeff Var	27.54987		

Utilities Table Based on the Usual Degrees of Freedom

Label		Utility	Standard Error	Importance (% Utility Range)	Variable
Intercept		4.9444	0.32107		Intercept
Brand	Goodstone	0.2222	0.45406	9.023	Class.BrandGoodstone
Brand	Pirogi	0.2222	0.45406		Class.BrandPirogi
Brand	Machismo	−0.4444	0.45406		Class.BrandMachismo
Price	$69.99	2.5556	0.45406	67.669	Class.Price_69_99
Price	$74.99	−0.1111	0.45406		Class.Price_74_99
Price	$79.99	−2.4444	0.45406		Class.Price_79_99
Life	50,000	−0.6111	0.45406	15.789	Class.Life50_000
Life	60,000	0.5556	0.45406		Class.Life60_000
Life	70,000	0.0556	0.45406		Class.Life70_000
Hazard	Yes	0.2778	0.32107	7.519	Class.HazardYes
Hazard	No	−0.2778	0.32107		Class.HazardNo

The next steps summarize the results. Three tables are displayed, showing the following: all of the importance values, the average importance, and the part-worth utilities. The first DATA step selects the importance information from the UTILS data set. The final assignment statement stores just the variable name from the label, relying on the fact that the separator is two blanks. PROC TRANSPOSE creates the data set of importances, one row per subject, and PROC PRINT displays the results. The MEANS procedure displays the average importance of each attribute across the subjects. The next DATA step selects the part-worth utilities information from the UTILS data set. PROC TRANSPOSE creates the data set of utilities, one row per subject, and PROC PRINT displays the results. The following statements produce Output 90.5.6:

```
title 'Tire Study Results';

* Gather the Importance Values;
data Importance;
   set utils(keep=_depvar_ Importance Label);
   if n(Importance);
   label = substr(label, 1, index(label, '  '));
run;

proc transpose out=Importance2(drop=_:);
   by _depvar_;
   id Label;
run;

proc print;
   title2 'Importance Values';
run;

proc means;
   title2 'Average Importance';
run;

* Gather the Part-Worth Utilities;
data Utilities;
   set utils(keep=_depvar_ Coefficient Label);
   if n(Coefficient);
run;

proc transpose out=Utilities2(drop=_:);
   by _depvar_;
   id Label;
   idlabel Label;
run;

proc print label;
   title2 'Utilities';
run;
```

Output 90.5.6 Summary of Conjoint Analysis Results

Tire Study Results
Importance Values

Obs	Brand	Price	Life	Hazard
1	20.8333	54.1667	16.6667	8.33333
2	25.6579	65.1316	7.8947	1.31579
3	18.2609	31.3043	41.7391	8.69565
4	36.8852	12.2951	49.1803	1.63934
5	9.0226	67.6692	15.7895	7.51880

Output 90.5.6 *continued*

Tire Study Results
Average Importance

The MEANS Procedure

Variable	N	Mean	Std Dev	Minimum	Maximum
Brand	5	22.1319800	10.2301014	9.0225564	36.8852459
Price	5	46.1133697	23.7391251	12.2950820	67.6691729
Life	5	26.2540671	18.0547195	7.8947368	49.1803279
Hazard	5	5.5005832	3.6989117	1.3157895	8.6956522

Output 90.5.6 *continued*

Tire Study Results
Utilities

Obs	Intercept	Brand Goodstone	Brand Pirogi	Brand Machismo	Price $69.99	Price $74.99
1	5.00000	0.33333	0.66667	-1.00000	2.16667	0.00000
2	4.94444	1.22222	-0.94444	-0.27778	2.88889	-0.27778
3	4.94444	0.05556	0.55556	-0.61111	1.05556	-0.11111
4	5.05556	0.61111	-1.55556	0.94444	0.27778	0.27778
5	4.94444	0.22222	0.22222	-0.44444	2.55556	-0.11111

Obs	Price $79.99	Life 50,000	Life 60,000	Life 70,000	Hazard Yes	Hazard No
1	-2.16667	-0.83333	0.50000	0.33333	0.33333	-0.33333
2	-2.61111	-0.27778	0.38889	-0.11111	0.05556	-0.05556
3	-0.94444	-1.44444	1.22222	0.22222	0.27778	-0.27778
4	-0.55556	-1.55556	1.77778	-0.22222	0.05556	-0.05556
5	-2.44444	-0.61111	0.55556	0.05556	0.27778	-0.27778

Based on the importance values, price is the most important attribute for some of the respondents, but expected tread life is most important for others. On the average, price is most important, followed by expected tread life and brand. Road hazard insurance is less important. Each of the brands is preferred by some of the respondents. All respondents preferred a lower price over a higher price, a longer tread life, and road hazard insurance.

Example 90.6: Preference Mapping of Automobile Data

This example uses PROC TRANSREG to perform a preference mapping (PREFMAP) analysis (Carroll 1972) of automobile preference data after a PROC PRINQUAL principal component analysis. The PREFMAP analysis is a response surface regression that locates ideal points for each dependent variable in a space defined by the independent variables.

The data are ratings obtained from 25 judges of their preference for each of 17 automobiles. The ratings were made on a scale of zero (very weak preference) to nine (very strong preference). These judgments were made in 1980 about that year's products. There are two character variables that indicate the manufacturer and model of the automobile. The data set also contains three ratings: miles per gallon (MPG), projected reliability (Reliability), and quality of the ride (Ride). These ratings are on a scale of one (bad) to five (good). PROC PRINQUAL creates an OUT= data set containing standardized principal component scores (Prin1 and Prin2), along with the ID variables Model, MPG, Reliability, and Ride.

While this data set contains all of the information needed for the subsequent preference mapping, you can make slightly more informative plots by adding new variable labels to the principal component score variables. The default labels are 'Component 1', 'Component 2', and so on. These are by necessity rather generic since they are created before any data are read, and they must be appropriate across BY groups when a BY variable is specified. In contrast, the MDPREF plot in PROC PRINQUAL has axis labels of the form 'Component 1 (43.54%)' and 'Component 2 (23.4%)' that show the proportion of variance accounted for by each component. You can create an output data set from the MDPREF plot by using the ODS OUTPUT statement and then use only the label information from it to reset the labels in the output data set from PROC PRINQUAL. In the DATA PLOT step, the SET statement for the MD data set is specified before the SET statement for the PRESULTS data set. The `if 0` ensures that no data are actually read from it, but nevertheless the properties of the Prin1 and Prin2 variables including the variable labels are set based on the properties of those variables in the MD data set.

The first PROC TRANSREG step fits univariate regression models for MPG and Reliability. All variables are designated IDENTITY. A vector drawn in the plot of Prin1 and Prin2 from the origin to the point defined by an attribute's regression coefficients approximately shows how the autos differ on that attribute. See Carroll (1972) for more information. The Prin1 and Prin2 columns of the TResult1 OUT= data set contain the automobile coordinates (_Type_='SCORE' observations) and endpoints of the MPG and Reliability vectors (_Type_='M COEFFI' observations).

The second PROC TRANSREG step fits a univariate regression model with Ride designated IDENTITY, and Prin1 and Prin2 designated POINT. The POINT expansion creates an additional independent variable _ISSQ_, which contains the sum of Prin1 squared and Prin2 squared. The OUT= data set TResult2 contains no _Type_='SCORE' observations, only ideal point (_Type_='M

POINT') coordinates for Ride. The coordinates of both the vectors and the ideal points are output by specifying COORDINATES in the OUTPUT statement in PROC TRANSREG.

A vector model is used for MPG and Reliability because perfectly efficient and reliable automobiles do not exist in the data set. The ideal points for MPG and Reliability are far removed from the plot of the automobiles. It is more likely that an ideal point for quality of the ride is in the plot, so an ideal point model is used for the ride variable. See Carroll (1972) and Schiffman, Reynolds, and Young (1981) for discussions of the vector model and point models (including the EPOINT and QPOINT versions of the point model that are not used in this example). For the vector model, the default coordinates stretch factor of 2.5 was used. This extends the vectors by a factor of 2.5 from their standard lengths, making a better graphical display. Sometimes the default vectors are short and near the origin, and they look better when they are extended.

The following statements produce Output 90.6.1 through Output 90.6.5:

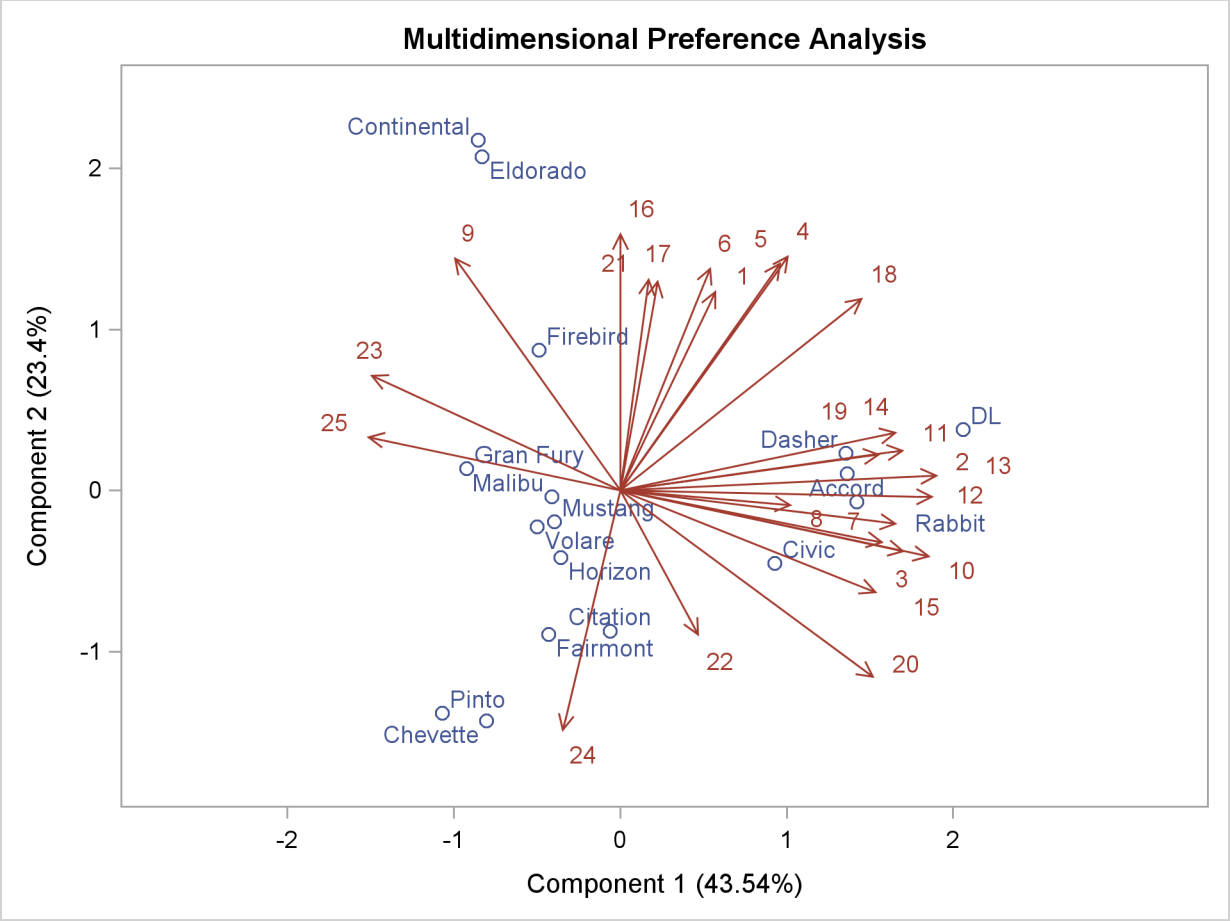
```
title 'Preference Ratings for Automobiles Manufactured in 1980';

options validvarname=any;

data CarPreferences;
   input Make $ 1-10 Model $ 12-22 @25 ('1'n-'25'n) (1.)
         MPG Reliability Ride;
   datalines;
Cadillac   Eldorado     800799049124050897109380 9 3 2 4
Chevrolet  Chevette     005120042345104300351569 8 5 3 2
Chevrolet  Citation     405330581416164354474779 5 4 1 5
Chevrolet  Malibu       602740072312134554566865 8 3 3 4
Ford       Fairmont     202400671502144353064865 5 3 3 4
Ford       Mustang      500719770502110185065755 5 3 2 2
Ford       Pinto        002100030303020150051407 8 4 1 1
Honda      Accord       595689760969995299897507 8 5 5 3
Honda      Civic        483670950748885256776507 5 5 5 3
Lincoln    Continental  700899059223040996209190 9 2 4 5
Plymouth   Gran Fury    700600043410110733345870 8 2 1 5
Plymouth   Horizon      300500563546130244467565 5 4 3 3
Plymouth   Volare       400500361402160275447655 5 2 1 3
Pontiac    Firebird     010789561320120695826590 7 1 1 5
Volkswagen Dasher       485869650887779537789500 0 5 3 4
Volkswagen Rabbit       485850970969579548788500 0 5 4 3
Volvo      DL           998999890999998798991900 0 4 5 5
;

ods graphics on;

* Compute Coordinates for a 2-Dimensional Scatter Plot of Automobiles;
proc prinqual data=CarPreferences out=PResults(drop='1'n-'25'n)
              n=2 replace standard scores mdpref=2;
   id Model MPG Reliability Ride;
   transform identity('1'n-'25'n);
   title2 'Multidimensional Preference (MDPREF) Analysis';
   ods output mdprefplot=md;
run;
```

```
options validvarname=v7;

title2 'Preference Mapping (PREFMAP) Analysis';

* Add the Labels from the Plot to the Results Data Set;
data plot;
   if 0 then set md(keep=prin:);
   set presults;
   run;

* Compute Endpoints for MPG and Reliability Vectors;
proc transreg data=plot rsquare;
   Model identity(MPG Reliability)=identity(Prin1 Prin2);
   output tstandard=center coordinates replace out=TResult1;
   id Model;
run;

* Compute Ride Ideal Point Coordinates;
proc transreg data=plot rsquare;
   Model identity(Ride)=point(Prin1 Prin2);
   output tstandard=center coordinates replace noscores out=TResult2;
   id Model;
run;

proc print;
run;
ods graphics off;
```

Output 90.6.1 Preference Ratings Example Output

```
            Preference Ratings for Automobiles Manufactured in 1980
                 Multidimensional Preference (MDPREF) Analysis

                              The PRINQUAL Procedure

                        PRINQUAL MTV Algorithm Iteration History

  Iteration     Average     Maximum      Proportion     Criterion
     Number      Change      Change     of Variance        Change     Note
  ------------------------------------------------------------------------
          1     0.00000     0.00000         0.66946                 Converged

   Algorithm converged.
```

Output 90.6.2 MDPREF Plot

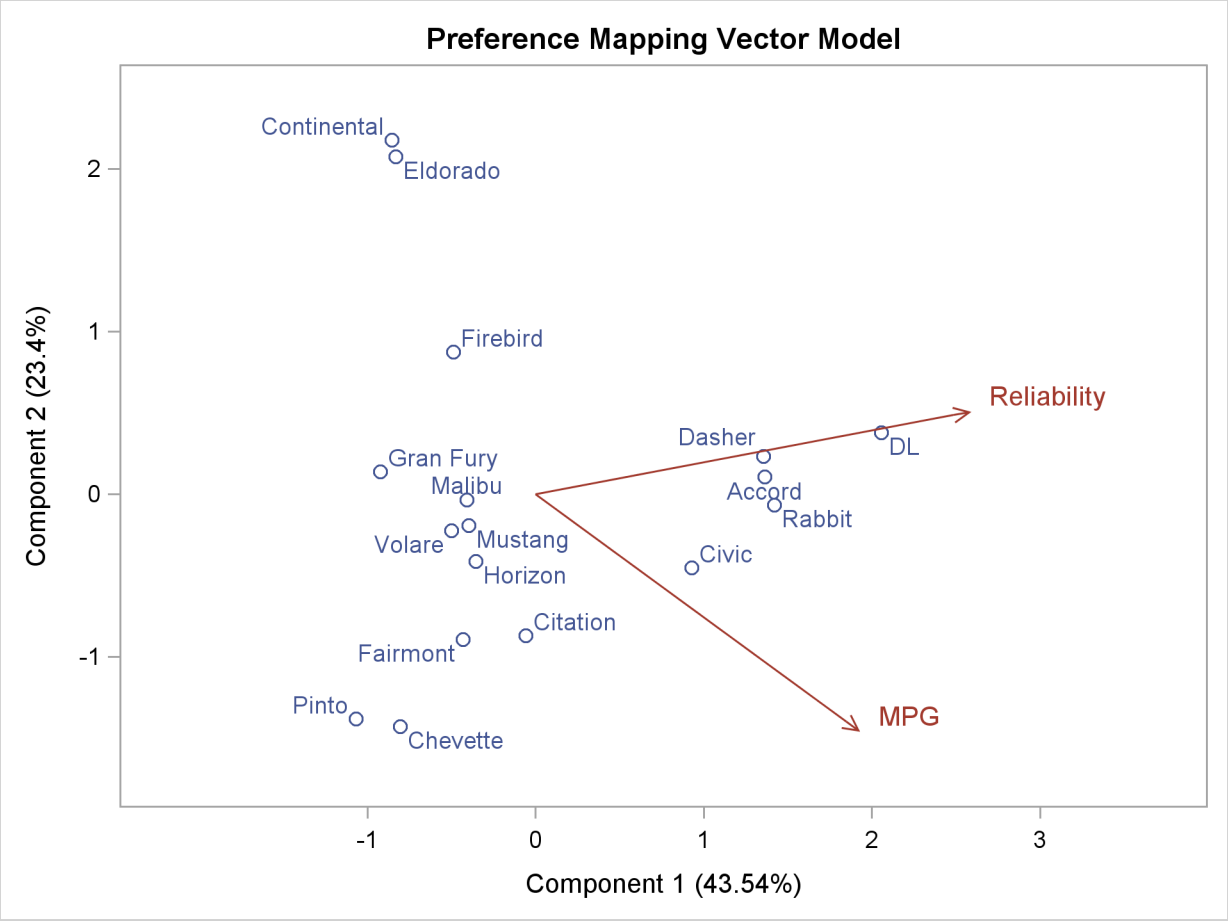


Output 90.6.3 shows that an unreliable-to-reliable direction extends from the left and slightly below the origin to the right and slightly above the origin. The Japanese and European automobiles are rated, on the average, as more reliable. A low MPG to good MPG direction extends from the top left of the plot to the bottom right. The smaller automobiles, on the average, get better gas mileage.

Output 90.6.3 Preference Mapping Vector Plot

```
Preference Ratings for Automobiles Manufactured in 1980
Preference Mapping (PREFMAP) Analysis

The TRANSREG Procedure

The TRANSREG Procedure Hypothesis Tests for Identity(MPG)

R-Square      0.5720

The TRANSREG Procedure Hypothesis Tests for Identity(Reliability)

R-Square      0.5086
```

Output 90.6.3 *continued*

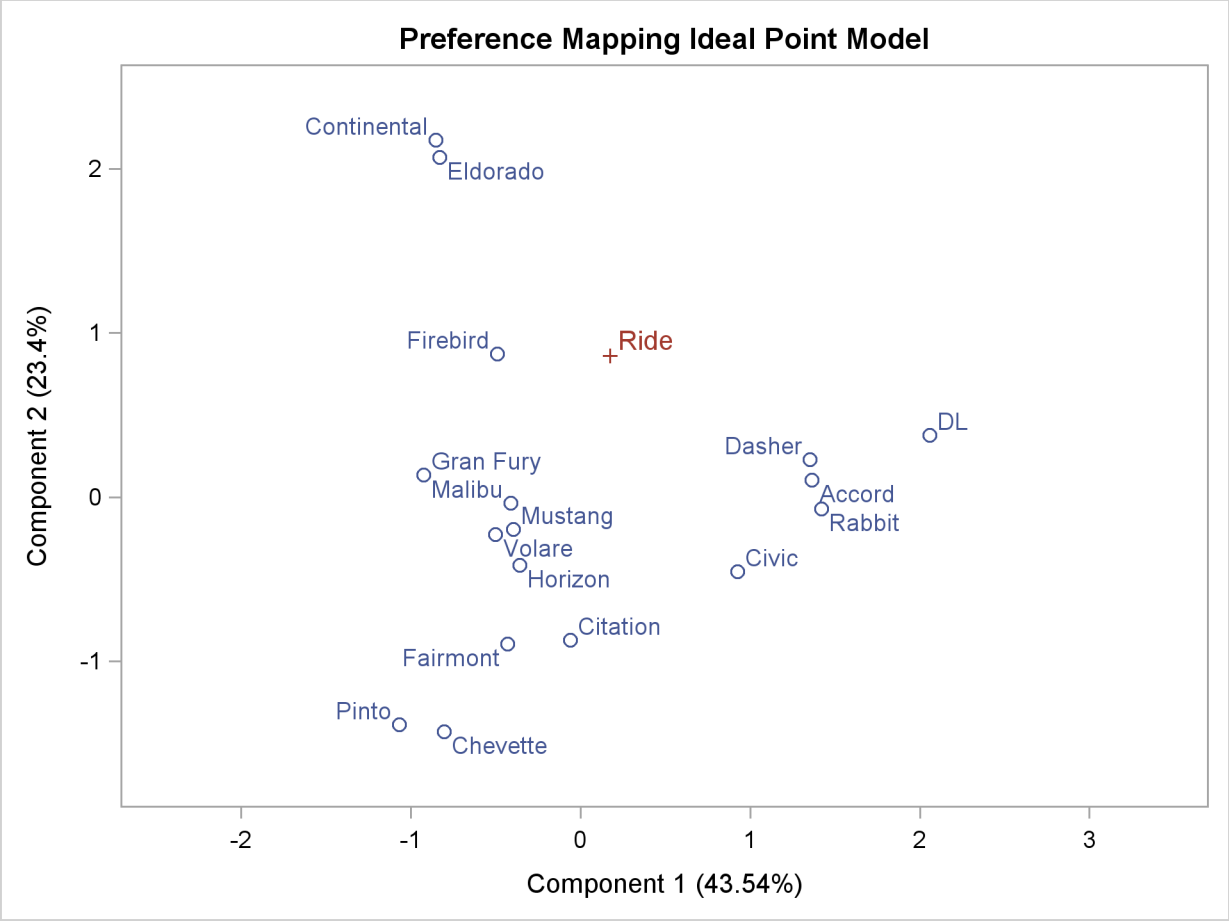


The ideal point for Ride in Output 90.6.4 is in the top, just right of the center of the plot. Automobiles near the Ride ideal point tend to have a better ride than automobiles far away. It can be seen from the R squares that none of these ratings perfectly fits the model, so all of the interpretations are approximate.

Output 90.6.4 Preference Mapping Ideal Point Plot

```
Preference Ratings for Automobiles Manufactured in 1980
Preference Mapping (PREFMAP) Analysis

The TRANSREG Procedure

The TRANSREG Procedure Hypothesis Tests for Identity(Ride)

R-Square      0.3780
```

Output 90.6.4 *continued*



The Ride point is a "negative-negative" ideal point. The point models assume that small ratings mean the object (automobile) is similar to the rating name and large ratings imply dissimilarity to the rating name. Because the opposite scoring is used, the interpretation of the Ride point must be reversed to a negative ideal point (bad ride). However, the coefficient for the _ISSQ_ variable in Output 90.6.5 is negative, so the interpretation is reversed again, back to the original interpretation.

Output 90.6.5 Preference Mapping Ideal Point Coefficients

Preference Ratings for Automobiles Manufactured in 1980
Preference Mapping (PREFMAP) Analysis

Obs	_TYPE_	_NAME_	Ride	Intercept	Prin1	Prin2	_ISSQ_	Model
1	M POINT	Ride	.	.	0.49461	2.46539	-0.17448	Ride

References

Akaike, H. (1973), "Information Theory and an Extension of the Maximum Likelihood Principle," in Petrov and Csaki, eds., *Proceedings of the Second International Symposium on Information Theory*, 267–281.

Box, G. E. P. and Cox, D. R. (1964), "An Analysis of Transformations," *Journal of the Royal Statistics Society, Series B*, 26, 211–234.

Breiman, L. and Friedman, J. H. (1985), "Estimating Optimal Transformations for Multiple Regression and Correlation," *Journal of the American Statistical Association*, 77, 580–619, with discussion.

Brent, R. P. (1973), *Algorithms for Minimization without Derivatives*, Englewood Cliffs, NJ: Prentice Hall, chapter 5.

Brinkman, N. D. (1981), "Ethanol Fuel—A Single-Cylinder Engine Study of Efficiency and Exhaust Emissions," *Society of Automotive Engineers Transactions*, 90, 1410–1424.

Carroll, J. D. (1972), "Individual Differences and Multidimensional Scaling," in R. N. Shepard, A. K. Romney, and S. B. Nerlove, eds., *Multidimensional Scaling: Theory and Applications in the Behavioral Sciences (Volume 1)*, New York: Seminar Press.

Craven, P. and Wahba, G. (1979), "Smoothing Noisy Data with Spline Functions," *Numerical Mathematics*, 31, 377–403.

de Boor, C. (1978), *A Practical Guide to Splines*, New York: Springer Verlag.

de Leeuw, J. (1986), *Regression with Optimal Scaling of the Dependent Variable*, Leiden: Department of Data Theory, University of Leiden.

de Leeuw, J., Young, F. W., and Takane, Y. (1976), "Additive Structure in Qualitative Data: An Alternating Least Squares Approach with Optimal Scaling Features," *Psychometrika*, 41, 471–503.

Draper, N. R. and Smith, H. (1981), *Applied Regression Analysis*, Second Edition, New York: John Wiley & Sons.

Eilers, P. H. C. and Marx, B. D. (1996), "Flexible Smoothing with *B*-Splines and Penalties," *Statistical Science*, 11, 89–121, with discussion.

Fisher, R. A. (1938), *Statistical Methods for Research Workers*, Tenth Edition, Edinburgh: Oliver & Boyd.

Gabriel, K. R. (1981), "Biplot Display of Multivariate Matrices for Inspection of Data and Diagnosis," in V. Barnett, ed., *Interpreting Multivariate Data*, London: John Wiley & Sons.

Gifi, A. (1990), *Nonlinear Multivariate Analysis*, New York: John Wiley & Sons.

Green, P. E. and Wind, Y. (1975), "New Way to Measure Consumers' Judgments," *Harvard Business Review*.

Hastie, T. and Tibshirani, R. (1986), "Generalized Additive Models," *Statistical Science*, 3, 297–318.

Hurvich, C. M., Simonoff, J. S., and Tsai, C. L. (1998), "Smoothing Parameter Selection in Nonparametric Regression Using an Improved Akaike Information Criterion," *Journal of the Royal Statistical Society B*, 60, 271–293.

Israels, A. Z. (1984), "Redundancy Analysis for Qualitative Variables," *Psychometrika*, 49, 331–346.

Judge, G. G., Griffiths, W. E., Hill, R. C., and Lee, T.-C. (1980), *The Theory and Practice of Econometrics*, New York: John Wiley & Sons.

Khuri, A. I. and Cornell, J. A. (1987), *Response Surfaces*, New York: Marcel Dekker.

Kruskal, J. B. (1964), "Nonmetric Multidimensional Scaling by Optimizing Goodness of Fit to a Nonmetric Hypothesis," *Psychometrika*, 29, 1–27.

Kuhfeld, W. F. (2005), *Marketing Research Methods in SAS*, Technical report, SAS Institute Inc., http://support.sas.com/resources/papers/tnote/tnote_marketresearch.html.

Meyers, R. H. (1976), *Response Surface Methodology*, Blacksburg, VA: Virginia Polytechnic Institute and State University.

National Institute of Standards and Technology (1998), *Statistical Reference Data Sets*, last accessed April 4, 2007.
URL http://www.nist.gov/itl/div898/strd

Press, W. H., Flannery, B. P., Teukoisky, S. A., and Vetterling, W. T. (1989), *Numerical Recipes in PASCAL*, Cambridge: Cambridge University Press.

Reinsch, C. H. (1967), "Smoothing by Spline Functions," *Numerische Mahematik*, 10, 177–183.

SAS Institute Inc. (1993), *Algorithms for the PRINQUAL and TRANSREG Procedures*, SAS Technical Report R-108, Cary, NC: SAS Institute Inc, http://support.sas.com/publishing/pubcat/techreports/59040.pdf.

Schiffman, S. S., Reynolds, M. L., and Young, F. W. (1981), *Introduction to Multidimensional Scaling*, New York: Academic Press.

Schwarz, G. (1978), "Estimating the Dimension of a Model," *Annals of Statistics*, 6, 461–464.

Siegel, S. (1956), *Nonparametric Statistics*, New York: McGraw-Hill.

Smith, P. L. (1979), "Splines as a Useful and Convenient Statistical Tool," *The American Statistician*, 33, 57–62.

Stewart, D. K. and Love, W. A. (1968), "A General Canonical Correlation Index," *Psychological Bulletin*, 70, 160–163.

van der Burg, E. and de Leeuw, J. (1983), "Non-linear Canonical Correlation," *British Journal of Mathematical and Statistical Psychology*, 36, 54–80.

van Rijckevorsel, J. (1982), "Canonical Analysis with B-Splines," in H. Caussinus, P. Ettinger, and R. Tomassone, eds., *COMPUSTAT 1982, Part I*, Vienna: Physica Verlag.

Winsberg, S. and Ramsay, J. O. (1980), "Monotonic Transformations to Additivity Using Splines," *Biometrika*, 67, 669–674.

Young, F. W. (1981), "Quantitative Analysis of Qualitative Data," *Psychometrika*, 46, 357–388.

Young, F. W., de Leeuw, J., and Takane, Y. (1976), "Regression with Qualitative and Quantitative Variables: An Alternating Least Squares Approach with Optimal Scaling Features," *Psychometrika*, 41, 505–529.

Chapter 91

The TREE Procedure

Contents

Overview: TREE Procedure **7355**
Getting Started: TREE Procedure **7356**
Syntax: TREE Procedure **7361**
PROC TREE Statement 7361
BY Statement 7367
COPY Statement 7367
FREQ Statement 7368
HEIGHT Statement 7368
ID Statement 7368
NAME Statement 7368
PARENT Statement 7369
Details: TREE Procedure **7369**
Missing Values 7369
Output Data Set 7369
Displayed Output 7370
ODS Table Names 7370
Examples: TREE Procedure **7371**
Example 91.1: Mammals' Teeth 7371
Example 91.2: Iris Data 7381
References **7389**

Overview: TREE Procedure

The TREE procedure produces a tree diagram, also known as a *dendrogram* or *phenogram*, from a data set created by the CLUSTER or VARCLUS procedure that contains the results of hierarchical clustering as a tree structure. The TREE procedure uses the data set to produce a diagram of the tree structure in the style of Johnson (1967), with the root at the top. Alternatively, the diagram can be oriented horizontally, with the root at the left. Any numeric variable in the output data set can be used to specify the heights of the clusters. PROC TREE can also create an output data set containing a variable to indicate the disjoint clusters at a specified level in the tree.

Tree diagrams are discussed in the context of cluster analysis by Duran and Odell (1974), Hartigan (1975), and Everitt (1980). Knuth (1973) provides a general treatment of tree diagrams in computer programming.

The literature on tree diagrams contains a mixture of botanical and genealogical terminology. The objects that are clustered are *leaves*. The cluster containing all objects is the *root*. A cluster containing at least two objects but not all of them is a *branch*. The general term for leaves, branches, and roots is *node*. If a cluster A is the union of clusters B and C, then A is the *parent* of B and C, and B and C are *children* of A. A leaf is thus a node with no children, and a root is a node with no parent. If every cluster has at most two children, the tree diagram is a *binary tree*. The CLUSTER procedure always produces binary trees. The VARCLUS procedure can produce tree diagrams with clusters that have many children.

Getting Started: TREE Procedure

The TREE procedure creates tree diagrams from a SAS data set containing the tree structure. You can create this type of data set with the CLUSTER or VARCLUS procedure. See Chapter 29, "The CLUSTER Procedure," and Chapter 93, "The VARCLUS Procedure," for more information.

In the following example, the VARCLUS procedure is used to divide a set of variables into hierarchical clusters and to create the SAS data set containing the tree structure. The TREE procedure then generates the tree diagrams.

The following data, from Hand et al. (1994), represent the amount of protein consumed from nine food groups for each of 25 European countries. The nine food groups are red meat (RedMeat), white meat (WhiteMeat), eggs (Eggs), milk (Milk), fish (Fish), cereal (Cereal), starch (Starch), nuts (Nuts), and fruits and vegetables (FruVeg).

The following SAS statements create the data set Protein:

```
data Protein;
   input Country $15. RedMeat WhiteMeat Eggs Milk
      Fish Cereal Starch Nuts FruVeg;
   datalines;
Albania         10.1  1.4  0.5   8.9  0.2  42.3  0.6  5.5  1.7
Austria          8.9 14.0  4.3  19.9  2.1  28.0  3.6  1.3  4.3
Belgium         13.5  9.3  4.1  17.5  4.5  26.6  5.7  2.1  4.0
Bulgaria         7.8  6.0  1.6   8.3  1.2  56.7  1.1  3.7  4.2
Czechoslovakia   9.7 11.4  2.8  12.5  2.0  34.3  5.0  1.1  4.0
Denmark         10.6 10.8  3.7  25.0  9.9  21.9  4.8  0.7  2.4
E Germany        8.4 11.6  3.7  11.1  5.4  24.6  6.5  0.8  3.6
Finland          9.5  4.9  2.7  33.7  5.8  26.3  5.1  1.0  1.4
France          18.0  9.9  3.3  19.5  5.7  28.1  4.8  2.4  6.5
Greece          10.2  3.0  2.8  17.6  5.9  41.7  2.2  7.8  6.5
Hungary          5.3 12.4  2.9   9.7  0.3  40.1  4.0  5.4  4.2
Ireland         13.9 10.0  4.7  25.8  2.2  24.0  6.2  1.6  2.9
Italy            9.0  5.1  2.9  13.7  3.4  36.8  2.1  4.3  6.7
Netherlands      9.5 13.6  3.6  23.4  2.5  22.4  4.2  1.8  3.7
```

```
Norway          9.4  4.7  2.7  23.3  9.7  23.0  4.6  1.6  2.7
Poland          6.9 10.2  2.7  19.3  3.0  36.1  5.9  2.0  6.6
Portugal        6.2  3.7  1.1   4.9 14.2  27.0  5.9  4.7  7.9
Romania         6.2  6.3  1.5  11.1  1.0  49.6  3.1  5.3  2.8
Spain           7.1  3.4  3.1   8.6  7.0  29.2  5.7  5.9  7.2
Sweden          9.9  7.8  3.5   4.7  7.5  19.5  3.7  1.4  2.0
Switzerland    13.1 10.1  3.1  23.8  2.3  25.6  2.8  2.4  4.9
UK             17.4  5.7  4.7  20.6  4.3  24.3  4.7  3.4  3.3
USSR            9.3  4.6  2.1  16.6  3.0  43.6  6.4  3.4  2.9
W Germany      11.4 12.5  4.1  18.8  3.4  18.6  5.2  1.5  3.8
Yugoslavia      4.4  5.0  1.2   9.5  0.6  55.9  3.0  5.7  3.2
;
run;
```

The data set Protein contains the character variable Country and the nine numeric variables representing the food groups. The `$15.` in the INPUT statement specifies that the variable Country is a character variable with a length of 15.

The following statements cluster the variables in the data set Protein. The OUTTREE= option creates an output data set named Tree to contain the tree structure. The CENTROID option specifies the centroid clustering method, and the MAXCLUSTERS= option specifies that the largest number of clusters desired is four. The NOPRINT option suppresses the display of the output. The VAR statement specifies that all numeric variables (RedMeat—FruVeg) are used by the procedure.

```
proc varclus data=Protein outtree=Tree centroid maxclusters=4 noprint;
   var RedMeat--FruVeg;
run;
```

The output data set Tree, created by the OUTTREE= option in the previous statements, contains the following variables:

NAME	the name of the cluster
PARENT	the parent of the cluster
NCL	the number of clusters
VAREXP	the amount of variance explained by the cluster
PROPOR	the proportion of variance explained by the clusters at the current level of the tree diagram
MINPRO	the minimum proportion of variance explained by a cluster
MAXEIGEN	the maximum second eigenvalue of a cluster

The following statements produce tree diagrams of the clusters created by PROC VARCLUS:

```
proc tree data=tree;
proc tree data=tree lineprinter;
```

PROC TREE is invoked twice. In the first invocation, the tree diagram is presented using the default graphical output. In the second invocation, the LINEPRINTER option specifies line printer output.

Figure 91.1 displays the default graphical representation of the tree diagram.

Figure 91.1 Graphical Tree Diagram from PROC TREE

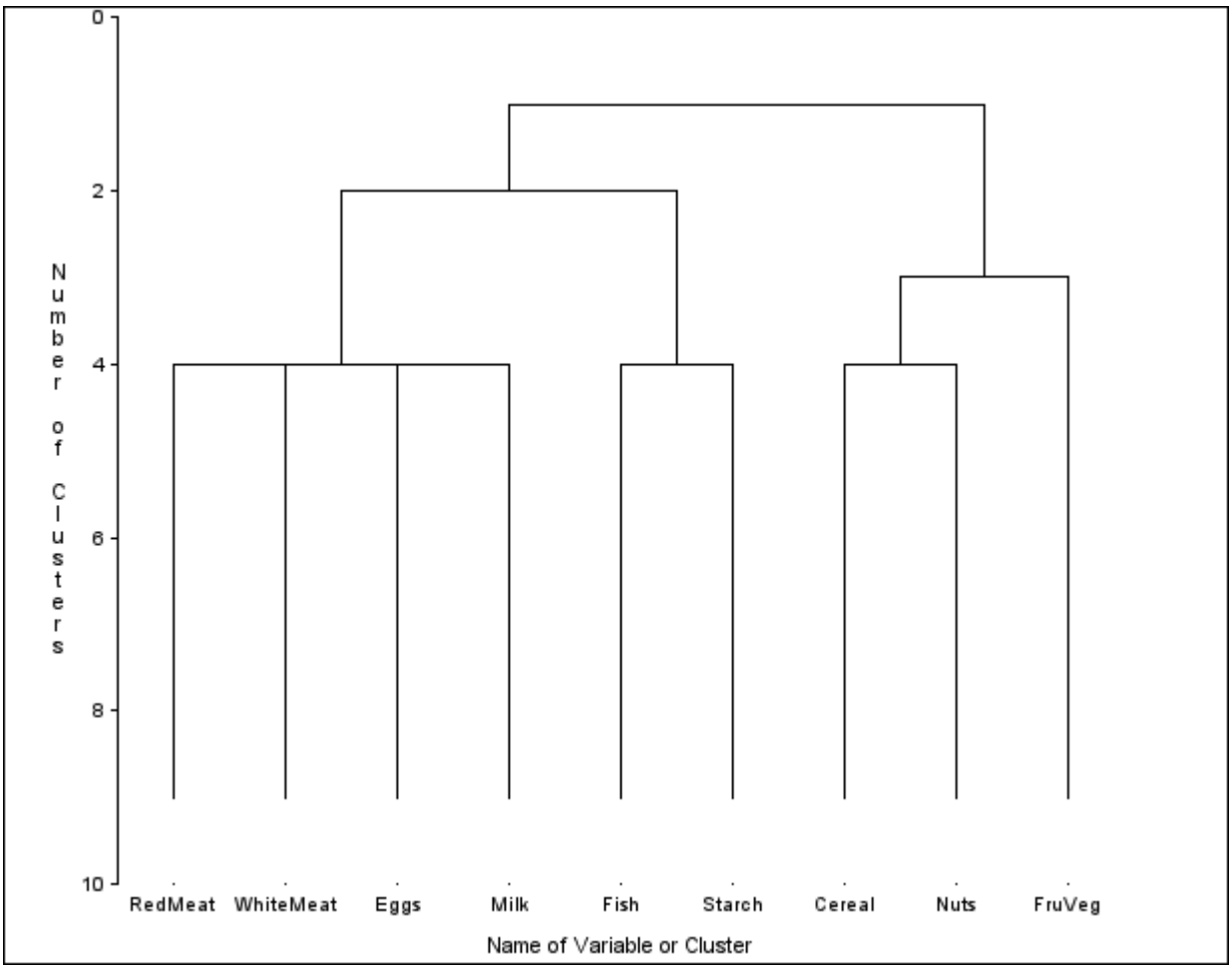
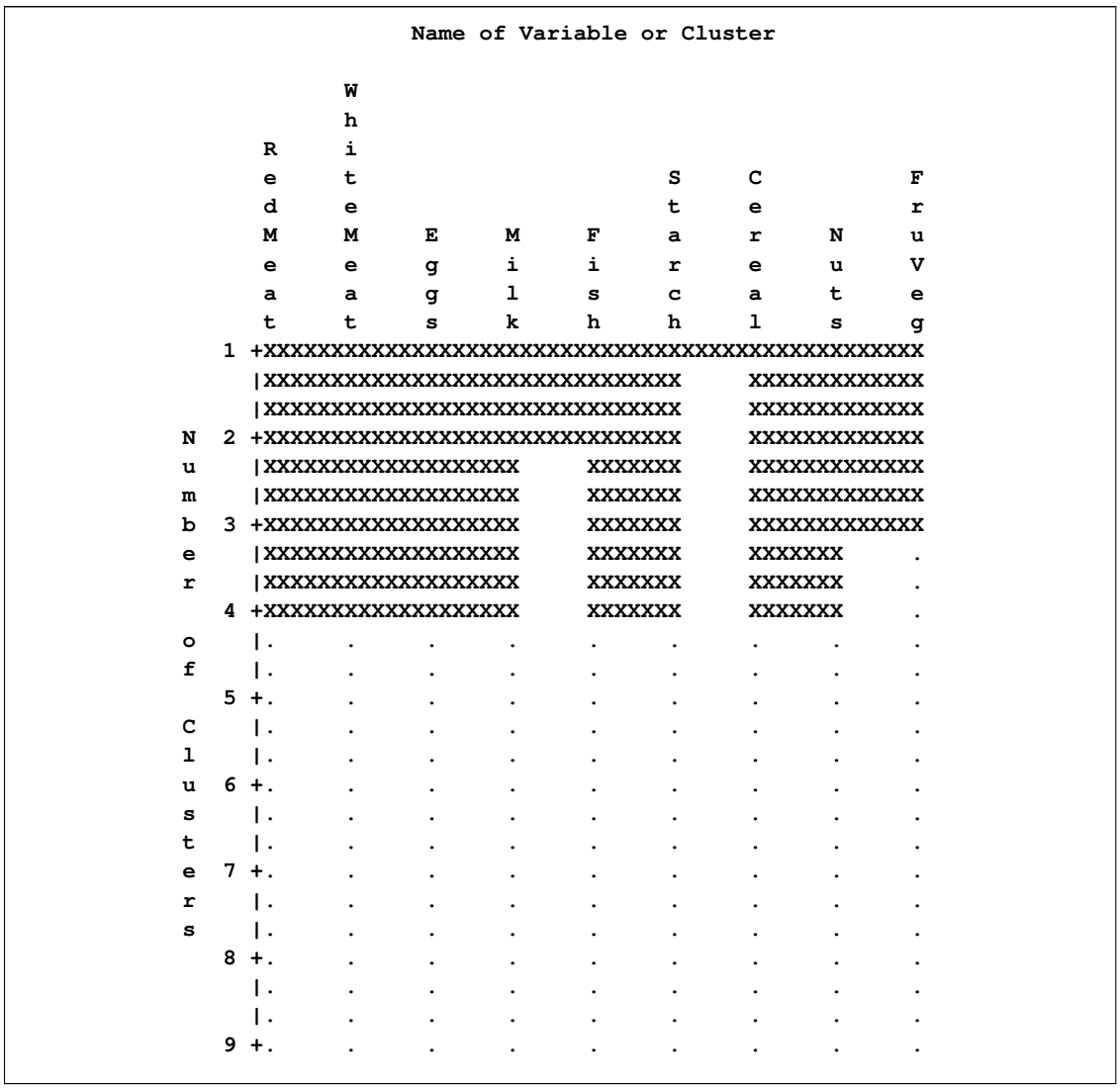


Figure 91.2 displays the same information as Figure 91.1 by using line printer output.

Figure 91.2 Line Printer Representation of the Tree Diagram



In each diagram the name of the cluster is displayed on the horizontal axis and the number of clusters is displayed on the vertical or height axis.

As you look up from the bottom of either diagram, clusters are progressively joined until a single, all-encompassing cluster is formed at the top (or root) of the tree. Clusters exist at each level of the diagram. For example, at the level where the diagram indicates three clusters, the clusters are as follows:

- Cluster 1: RedMeat WhiteMeat Eggs Milk
- Cluster 2: Fish Starch
- Cluster 3: Cereal Nuts FruVeg

As you proceed up the diagram one level, the number of clusters is two:

- Cluster 1: RedMeat WhiteMeat Eggs Milk Fish Starch
- Cluster 2: Cereal Nuts FruVeg

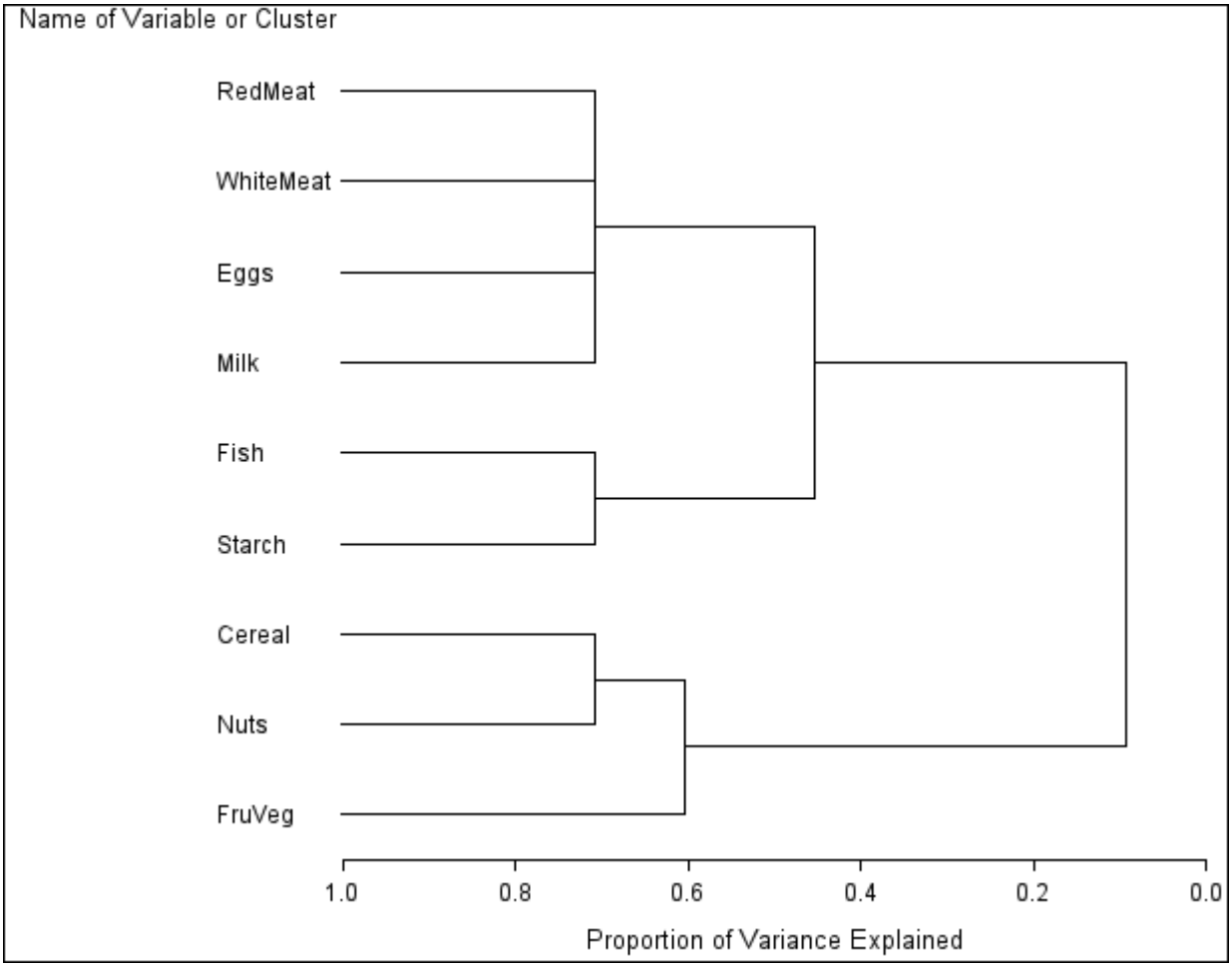
The following statements illustrate how you can specify the numeric variable defining the height of each node (cluster) in the tree:

```
axis1 order=(0 to 1 by 0.2);
proc tree data=Tree horizontal haxis=axis1;
   height _PROPOR_;
run;
```

The AXIS1 statement ORDER= option specifies the data values in the order in which they are to appear on the axis. The HORIZONTAL option in the PROC TREE statement orients the tree diagram horizontally. The HAXIS option specifies that the AXIS1 statement be used to customize the appearance of the horizontal axis. The HEIGHT statement specifies the variable _PROPOR_ (the proportion of variance explained) as the height variable.

The resulting tree diagram is shown in Figure 91.3.

Figure 91.3 Horizontal Tree Diagram with _PROPOR_ as the HEIGHT Variable



As you look from left to right in the diagram, objects and clusters are progressively joined until a single, all-encompassing cluster is formed at the right (or root) of the tree.

Clusters exist at each level of the diagram, represented by horizontal line segments. Each vertical line segment represents a point where leaves and branches are connected into progressively larger clusters.

For example, three clusters are formed at the leftmost point along the axis where three horizontal line segments exist. At that point, where a vertical line segment connects the Cereal-Nuts and FruVeg clusters, the proportion of variance explained is about 0.6 (_PROPOR_ = 0.6). At the next clustering level the variables Fish and Starch are clustered with variables RedMeat through Milk, resulting in a total of two clusters. The proportion of variance explained is about 0.45 at that point.

Syntax: TREE Procedure

The TREE procedure is invoked by the following statements:

```
PROC TREE < options > ;
    NAME variables ;
    HEIGHT variable ;
    PARENT variables ;
    BY variables ;
    COPY variables ;
    FREQ variable ;
    ID variable ;
```

If the input data set has been created by CLUSTER or VARCLUS, the only statement required is the PROC TREE statement. The BY, COPY, FREQ, HEIGHT, ID, NAME, and PARENT statements are described after the PROC TREE statement.

PROC TREE Statement

```
PROC TREE < options > ;
```

The PROC TREE statement starts the TREE procedure.

The options that can appear in the PROC TREE statement are summarized in Table 91.1.

Table 91.1 PROC TREE Statement Options

Option	Description
Data Sets	
DATA=	specifies input data set
DOCK=	specifies that small clusters not be counted in OUT= data set

Table 91.1 *continued*

Option	Description
LEVEL=	defines disjoint cluster in OUT= data set
NCLUSTERS=	specifies number of clusters in OUT= data set
OUT=	specifies output data set
ROOT=	displays root of a subtree
Cluster Heights	
HEIGHT=	specifies variable for the height axis
DISSIMILAR	specifies that height values indicate dissimilarity
SIMILAR	specifies that height values indicate similarity
Horizontal Trees	
HORIZONTAL	specifies that height axis be horizontal
Sort Order	
DESCENDING	reverses sort order
SORT	sorts children by HEIGHT variable
Displayed Output	
INC=	specifies increment between tick values
LINEPRINTER	displays tree by using line printer graphics
LIST	displays all nodes in tree
MAXHEIGHT=	specifies maximum value on axis
MINHEIGHT=	specifies minimum value on axis
NOPRINT	suppresses display of tree
NTICK=	specifies number of tick intervals
Graphics	
CFRAME=	specifies color of the frame
DESCRIPTION=	specifies catalog description
GOUT=	specifies catalog name
HAXIS=	customizes horizontal axis
HORDISPLAY=	displays horizontal tree with leaves on right
HPAGES=	specifies number of pages to expand tree horizontally
LINES=	specifies line color and thickness, dots at nodes
NAME=	specifies name of graph in catalog
VAXIS=	customizes vertical axis
VPAGES=	specifies number of pages to expand tree vertically
Line Printer Graphics	
PAGES=	specifies number of pages
POS=	specifies number of column positions
SPACES=	specifies number of spaces between objects
TICKPOS=	specifies number of column positions between ticks
FILLCHAR=	specifies fill character between unjoined leaves
JOINCHAR=	specifies character displayed between joined leaves
LEAFCHAR=	specifies character representing clusters with no children
TREECHAR=	specifies character representing clusters with children

CFRAME=*color*

specifies a color for the frame, which is the rectangle bounded by the axes.

DATA=*SAS-data-set*

specifies the input data set defining the tree. If you omit the DATA= option, the most recently created SAS data set is used.

DESCENDING

DES

reverses the sorting order for the SORT option.

DESCRIPTION=*entry-description*

specifies a description for the graph in the GOUT= catalog. The default is "Proc Tree Graph Output."

DISSIMILAR

DIS

specifies that the values of the HEIGHT variable are dissimilarities; that is, a large height value means that the clusters are very dissimilar or far apart.

If neither the SIMILAR nor the DISSIMILAR option is specified, PROC TREE attempts to infer from the data whether the height values are similarities or dissimilarities. If PROC TREE cannot tell this from the data, it issues an error message and does not display a tree diagram.

DOCK=*n*

causes observations in the OUT= data set assigned to output clusters with a frequency of *n* or less to be given missing values for the output variables CLUSTER and CLUSNAME. If the NCLUSTERS= option is also specified, DOCK= also prevents clusters with a frequency of *n* or less from being counted toward the number of clusters requested by the NCLUSTERS= option. By default, DOCK=0.

FILLCHAR=*'c'*

FC=*'c'*

specifies the character to display between leaves that are not joined into a cluster. The character should be enclosed in single quotes. The default is a blank. The LINEPRINTER option must also be specified.

GOUT=<*libref.*>*member-name*

specifies the catalog in which the generated graph is stored. The default is WORK.GSEG.

HAXIS=AXIS*n*

specifies that the AXIS*n* statement used to customize the appearance of the horizontal axis.

HEIGHT=*name*

H=*name*

specifies certain conventional variables to be used for the height axis of the tree diagram. For many situations, the only option you need is the HEIGHT= option. Valid values for *name* and their meanings are as follows:

HEIGHT | H specifies the _HEIGHT_ variable.

LENGTH | L	defines the height of each node as its path length from the root. This can also be interpreted as the number of ancestors of the node.
MODE | M	specifies the _MODE_ variable.
NCL | N	specifies the _NCL_ (number of clusters) variable.
RSQ | R	specifies the _RSQ_ variable.

See also the section "HEIGHT Statement" on page 7368, which can specify any variable in the input data set to be used for the height axis. In rare cases, you might need to specify either the DISSIMILAR option or the SIMILAR option.

HORDISPLAY=RIGHT

specifies that the graph be oriented horizontally, with the leaf nodes on the right side, when the HORIZONTAL option is also specified. By default, the leaf nodes are on the left side.

HORIZONTAL

HOR

displays the tree diagram with the height axis oriented horizontally. The leaf nodes are on the side specified in the HORDISPLAY= option. If you do not specify the HORIZONTAL option, the height axis is vertical, with the root at the top. When the tree takes up more than one page, horizontal orientation can make the tree diagram considerably easier to read.

HPAGES=*n1*

specifies that the original graph be enlarged to cover $n1$ pages. If you also specify the VPAGES=$n2$ option, the original graph is enlarged to cover $n1 \times n2$ graphs. For example, if HPAGES=2 and VPAGES=3, then the original graph is generated, followed by $2 \times 3 = 6$ more graphs. In these six graphs, the original is enlarged by a factor of 2 in the horizontal direction and by a factor of 3 in the vertical direction. The graphs are generated in left-to-right and top-to-bottom order.

INC=*n*

specifies the increment between tick values on the height axis. If the HEIGHT variable is _NCL_, the default is usually 1, although a different value can be specified for consistency with other options. For any other HEIGHT variable, the default is some power of 10 times 1, 2, 2.5, or 5.

JOINCHAR=*'c'*

JC=*'c'*

specifies the character displayed between leaves that are joined into a cluster. The character should be enclosed in single quotes. The default is 'X'. The LINEPRINTER option must also be specified.

LEAFCHAR=*'c'*

LC=*'c'*

specifies the character used to represent clusters having no children. The character should be enclosed in single quotes. The default is a period. The LINEPRINTER option must also be specified.

LEVEL=*n*

specifies the level of the tree defining disjoint clusters for the OUT= data set. The LEVEL= option also causes only clusters between the root and a height of n to be displayed. The clusters in the output data set are those that exist at a height of n on the tree diagram. For example, if the HEIGHT variable is _NCL_ (number of clusters) and LEVEL=5 is specified, then the OUT= data set contains five disjoint clusters. If the HEIGHT variable is _RSQ_ (R^2) and LEVEL=0.9 is specified, then the OUT= data set contains the smallest number of clusters that yields an R^2 of at least 0.9.

LINEPRINTER

specifies that the tree diagram be displayed using line printer graphics.

LINES=(<COLOR=*color***> <WIDTH=***n***> <DOTS>)**

specifies the color and the thickness of the lines of the tree, and whether a dot is drawn at each leaf node. Note that if the frame and the lines are specified to be the same color, PROC TREE selects a different color for the lines.

LIST

lists all the nodes in the tree, displaying the height, parent, and children of each node.

MAXHEIGHT=*n*

MAXH=*n*

specifies the maximum value displayed on the height axis.

MINHEIGHT=*n*

MINH=*n*

specifies the minimum value displayed on the height axis.

NAME=*name*

specifies the entry name for the generated graph in the GOUT= catalog. Note that each time another graph is generated with the same name, the name is modified by appending a number to make it unique.

NCLUSTERS=*n*

NCL=*n*

N=*n*

specifies the number of clusters desired in the OUT= data set. The number of clusters obtained might not equal the number specified if (1) there are fewer than n leaves in the tree, (2) there are more than n unconnected trees in the data set, (3) a multiway tree does not contain a level with the specified number of clusters, or (4) the DOCK= option eliminates too many clusters.

The NCLUSTERS= option uses the _NCL_ variable to determine the order in which the clusters are formed. If there is no _NCL_ variable, the height variable (as determined by the HEIGHT statement or HEIGHT= option) is used instead.

NTICK=*n*

specifies the number of tick intervals on the height axis. The default depends on the values of other options.

NOPRINT

suppresses the display of the tree. Specify the NOPRINT option if you want only to create an OUT= data set.

OUT=*SAS-data-set*

creates an output data set containing one observation for each object in the tree or subtree being processed and variables called CLUSTER and CLUSNAME showing cluster membership at any specified level in the tree. If you specify the OUT= option, you must also specify either the NCLUSTERS= or LEVEL= option in order to define the output partition level. If you want to create a permanent SAS data set, you must specify a two-level name (see "SAS Data Files" in *SAS Language Reference: Concepts*).

PAGES=*n*

specifies the number of pages over which the tree diagram (from root to leaves) is to extend. The default is 1. The LINEPRINTER option must also be specified.

POS=*n*

specifies the number of column positions on the height axis. The default depends on the value of the PAGES= option, the orientation of the tree diagram, and the values specified by the PAGESIZE= and LINESIZE= options. The LINEPRINTER option must also be specified.

ROOT=*'name'*

specifies the value of the NAME variable for the root of a subtree to be displayed if you do not want to display the entire tree. If you also specify the OUT= option, the output data set contains only objects belonging to the subtree specified by the ROOT= option.

SIMILAR

SIM

specifies that the values of the HEIGHT variable represent similarities; that is, a large height value means that the clusters are very similar or close together.

If neither the SIMILAR nor the DISSIMILAR option is specified, PROC TREE attempts to infer from the data whether the height values are similarities or dissimilarities. If PROC TREE cannot tell this from the data, it issues an error message and does not display a tree diagram.

SORT

sorts the children of each node by the HEIGHT variable, in the order of cluster formation. See the DESCENDING option on page 7363 for details.

SPACES=*s*

S=*s*

specifies the number of spaces between objects in the output. The default depends on the number of objects, the orientation of the tree diagram, and the values specified by the PAGESIZE= and LINESIZE= options. The LINEPRINTER option must also be specified.

TICKPOS=*n*

specifies the number of column positions per tick interval on the height axis. The default value is usually between 5 and 10, although a different value can be specified for consistency with other options.

TREECHAR=*'c'*

TC=*'c'*

specifies the character used to represent clusters with children. The character should be enclosed in single quotes. The default is 'X'. The LINEPRINTER option must also be specified.

VAXIS=AXIS*n*

specifies that the AXIS*n* statement be used to customize the appearance of the vertical axis.

VPAGES=*n2*

specifies that the original graph be enlarged to cover $n2$ pages. If you also specify the HPAGES=$n1$ option, the original graph is enlarged to cover $n1 \times n2$ pages. For example, if HPAGES=2 and VPAGES=3, then the original graph is generated, followed by $2 \times 3 = 6$ more graphs. In these six graphs, the original is enlarged by a factor of 2 in the horizontal direction and by a factor of 3 in the vertical direction. The graphs are generated in left-to-right and top-to-bottom order.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC TREE to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the TREE procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

COPY Statement

COPY *variables* **;**

The COPY statement specifies one or more character or numeric variables to be copied to the OUT= data set.

FREQ Statement

FREQ *variable* **;**

The FREQ statement specifies one numeric variable that tells how many clustering observations belong to the cluster. If the FREQ statement is omitted, PROC TREE uses the variable _FREQ_ to specify the number of observations per cluster. If neither the FREQ statement nor the _FREQ_ variable is present, each leaf is assumed to represent one clustering observation, and the frequency for each internal node is found by summing the frequencies of its children.

HEIGHT Statement

HEIGHT *variable* **;**

The HEIGHT statement specifies the name of a numeric variable to define the height of each node (cluster) in the tree. The height variable can also be specified by the HEIGHT= option in the PROC TREE statement. If both the HEIGHT statement and the HEIGHT= option are omitted, PROC TREE uses the variable _HEIGHT_. If the data set does not contain _HEIGHT_, PROC TREE uses the variable _NCL_. If _NCL_ is not present either, the height of each node is defined to be its path length from the root.

ID Statement

ID *variable* **;**

The ID variable is used to identify the objects (leaves) in the tree on the output. The ID variable can be a character or numeric variable of any length. If the ID statement is omitted, the variable in the NAME statement is used instead. If both the ID and NAME statements are omitted, PROC TREE uses the variable _NAME_. If the _NAME_ variable is not found in the data set, PROC TREE issues an error message and stops. The ID variable is copied to the OUT= data set.

NAME Statement

NAME *variable* **;**

The NAME statement specifies a character or numeric variable identifying the node represented by each observation. The NAME variable and the PARENT variable jointly define the tree structure. If the NAME statement is omitted, PROC TREE uses the variable _NAME_. If the _NAME_ variable is not found in the data set, PROC TREE issues an error message and stops.

PARENT Statement

PARENT *variable* **;**

The PARENT statement specifies a character or numeric variable identifying the node in the tree that is the parent of each observation. The PARENT variable must have the same formatted length as the NAME variable. If the PARENT statement is omitted, PROC TREE uses the variable _PARENT_. If the _PARENT_ variable is not found in the data set, PROC TREE issues an error message and stops.

Details: TREE Procedure

Missing Values

An observation with a missing value for the NAME variable is omitted from processing. If the PARENT variable has a missing value but the NAME variable is present, the observation is treated as the root of a tree. A data set can contain several roots and, hence, several trees.

Missing values of the HEIGHT variable are set to upper or lower bounds determined from the nonmissing values under the assumption that the heights are monotonic with respect to the tree structure.

Missing values of the FREQ variable are inferred from nonmissing values where possible; otherwise, they are treated as zero.

Output Data Set

The OUT= data set contains one observation for each leaf in the tree or subtree being processed. The variables are as follows:

- the BY variables, if any
- the ID variable, or the NAME variable if the ID statement is not used
- the COPY variables
- a numeric variable CLUSTER taking values from 1 to c, where c is the number of disjoint clusters. The cluster to which the first observation belongs is given the number 1, the cluster to which the next observation belongs that does not belong to cluster 1 is given the number 2, and so on.

- a character variable CLUSNAME giving the value of the NAME variable of the cluster to which the observation belongs

The CLUSTER and CLUSNAME variables are missing if the corresponding leaf has a nonpositive frequency.

Displayed Output

The displayed output from the TREE procedure includes the following:

- the names of the objects in the tree
- the height axis
- the tree diagram. The leaves are displayed at the bottom of the graph. Horizontal lines connect the leaves into branches, while the topmost horizontal line indicates the root.

  If the LINEPRINTER option is specified, the root (the cluster containing all the objects) is indicated by a solid line of the character specified by the TREECHAR= option (the default character is 'X'). At each level of the tree, clusters are shown by unbroken lines of the TREECHAR= symbol with the FILLCHAR= symbol (the default is a blank) separating the clusters. The LEAFCHAR= symbol (the default character is a period) represents single-member clusters.

By default, the tree diagram is oriented with the height axis vertical and the object names at the top of the diagram. If the HORIZONTAL option is specified, then the height axis is horizontal and the object names are on the left.

ODS Table Names

PROC TREE assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 91.2. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 91.2 ODS Tables Produced by PROC TREE

ODS Table Name	Description	Statement	Option
TreeListing	Listing of all nodes in the tree	PROC	LIST

Examples: TREE Procedure

Example 91.1: Mammals' Teeth

The following statements produce a data set containing the numbers of different kinds of teeth for a variety of mammals:

```
data teeth;
   title 'Mammals'' Teeth';
   input mammal $ 1-16 @21 (v1-v8) (1.);
   label V1='Right Top Incisors'
         V2='Right Bottom Incisors'
         V3='Right Top Canines'
         V4='Right Bottom Canines'
         V5='Right Top Premolars'
         V6='Right Bottom Premolars'
         V7='Right Top Molars'
         V8='Right Bottom Molars';
   datalines;
Brown Bat           23113333
Mole                32103333
Silver Hair Bat     23112333
Pigmy Bat           23112233
House Bat           23111233
Red Bat             13112233
Pika                21002233
Rabbit              21003233
Beaver              11002133
Groundhog           11002133
Gray Squirrel       11001133
House Mouse         11000033
Porcupine           11001133
Wolf                33114423
Bear                33114423
Raccoon             33114432
Marten              33114412
Weasel              33113312
Wolverine           33114412
Badger              33113312
River Otter         33114312
Sea Otter           32113312
Jaguar              33113211
Cougar              33113211
Fur Seal            32114411
Sea Lion            32114411
Grey Seal           32113322
Elephant Seal       21114411
Reindeer            04103333
Elk                 04103333
```

```
Deer                 04003333
Moose                04003333
;
run;
```

The following statements use the CLUSTER procedure to cluster the mammals by average linkage and the TREE procedure to produce a horizontal tree diagram that uses the average-linkage distance as its height axis:

```
proc cluster method=average std pseudo noeigen outtree=tree;
   id mammal;
   var v1-v8;
run;

proc tree horizontal;
run;
```

Output 91.1.1 displays the information about how the clusters are joined. For example, the cluster history shows that the observations Wolf and Bear form cluster 29, which is merged with Raccoon to form cluster 11.

Output 91.1.1 Output from PROC CLUSTER

```
                              Mammals' Teeth

                          The CLUSTER Procedure
                       Average Linkage Cluster Analysis

          The data have been standardized to mean 0 and variance 1

          Root-Mean-Square Total-Sample Standard Deviation           1

           Root-Mean-Square Distance Between Observations           4
```

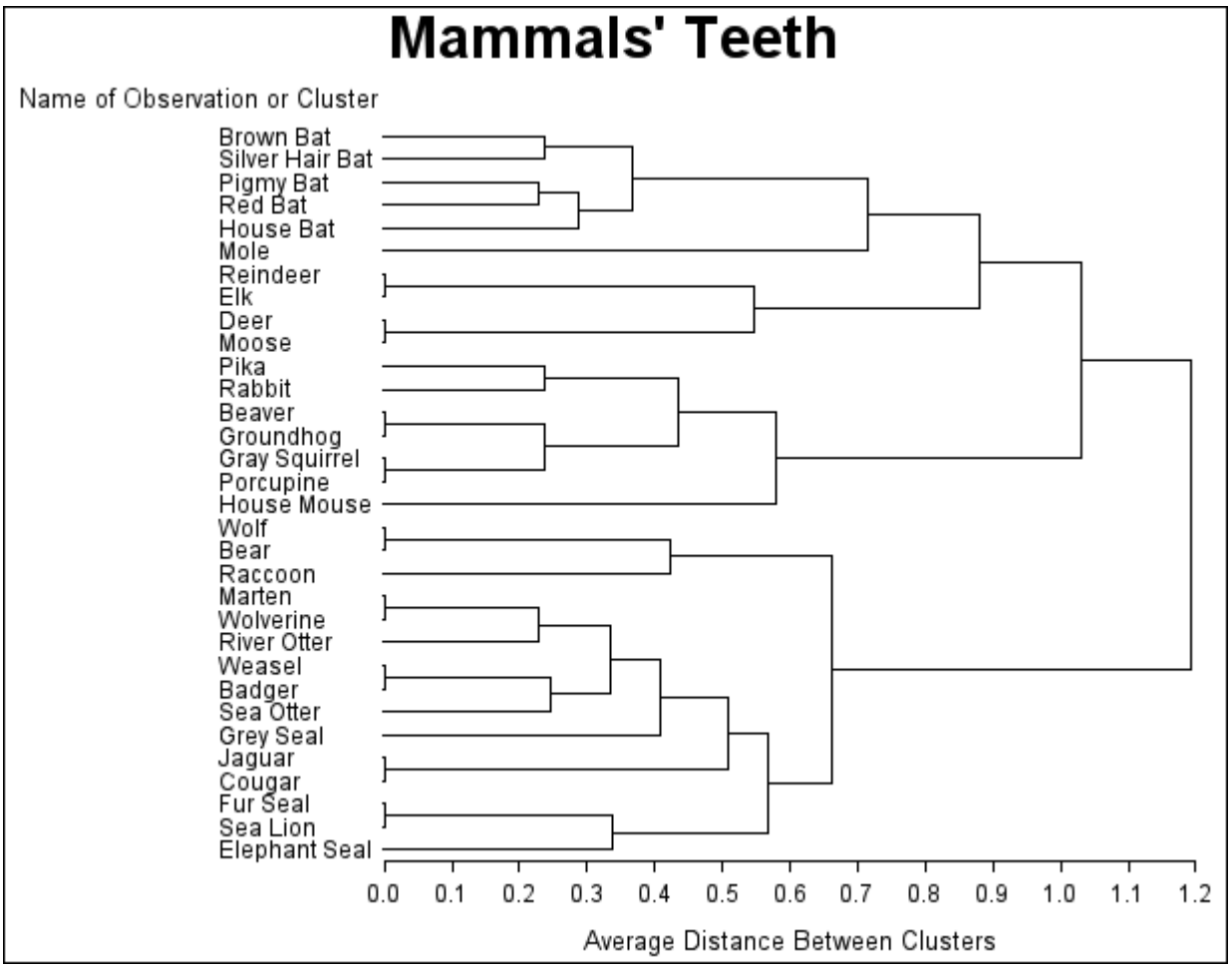
Output 91.1.1 *continued*

Cluster History

NCL	----------Clusters Joined----------		FREQ	PSF	PST2	Norm RMS Dist	Tie
31	Beaver	Groundhog	2	.	.	0	T
30	Gray Squirrel	Porcupine	2	.	.	0	T
29	Wolf	Bear	2	.	.	0	T
28	Marten	Wolverine	2	.	.	0	T
27	Weasel	Badger	2	.	.	0	T
26	Jaguar	Cougar	2	.	.	0	T
25	Fur Seal	Sea Lion	2	.	.	0	T
24	Reindeer	Elk	2	.	.	0	T
23	Deer	Moose	2	.	.	0	
22	Pigmy Bat	Red Bat	2	281	.	0.2289	
21	CL28	River Otter	3	139	.	0.2292	
20	CL31	CL30	4	83.2	.	0.2357	T
19	Brown Bat	Silver Hair Bat	2	76.7	.	0.2357	T
18	Pika	Rabbit	2	73.2	.	0.2357	
17	CL27	Sea Otter	3	67.4	.	0.2462	
16	CL22	House Bat	3	62.9	1.7	0.2859	
15	CL21	CL17	6	47.4	6.8	0.3328	
14	CL25	Elephant Seal	3	45.0	.	0.3362	
13	CL19	CL16	5	40.8	3.5	0.3672	
12	CL15	Grey Seal	7	38.9	2.8	0.4078	
11	CL29	Raccoon	3	38.0	.	0.423	
10	CL18	CL20	6	34.5	10.3	0.4339	
9	CL12	CL26	9	30.0	7.3	0.5071	
8	CL24	CL23	4	28.7	.	0.5473	
7	CL9	CL14	12	25.7	7.0	0.5668	
6	CL10	House Mouse	7	28.3	4.1	0.5792	
5	CL11	CL7	15	26.8	6.9	0.6621	
4	CL13	Mole	6	31.9	7.2	0.7156	
3	CL4	CL8	10	31.0	12.7	0.8799	
2	CL3	CL6	17	27.8	16.1	1.0316	
1	CL2	CL5	32	.	27.8	1.1938	

Output 91.1.2 shows the corresponding tree diagram.

Output 91.1.2 Tree Diagram of Mammal Teeth Clusters



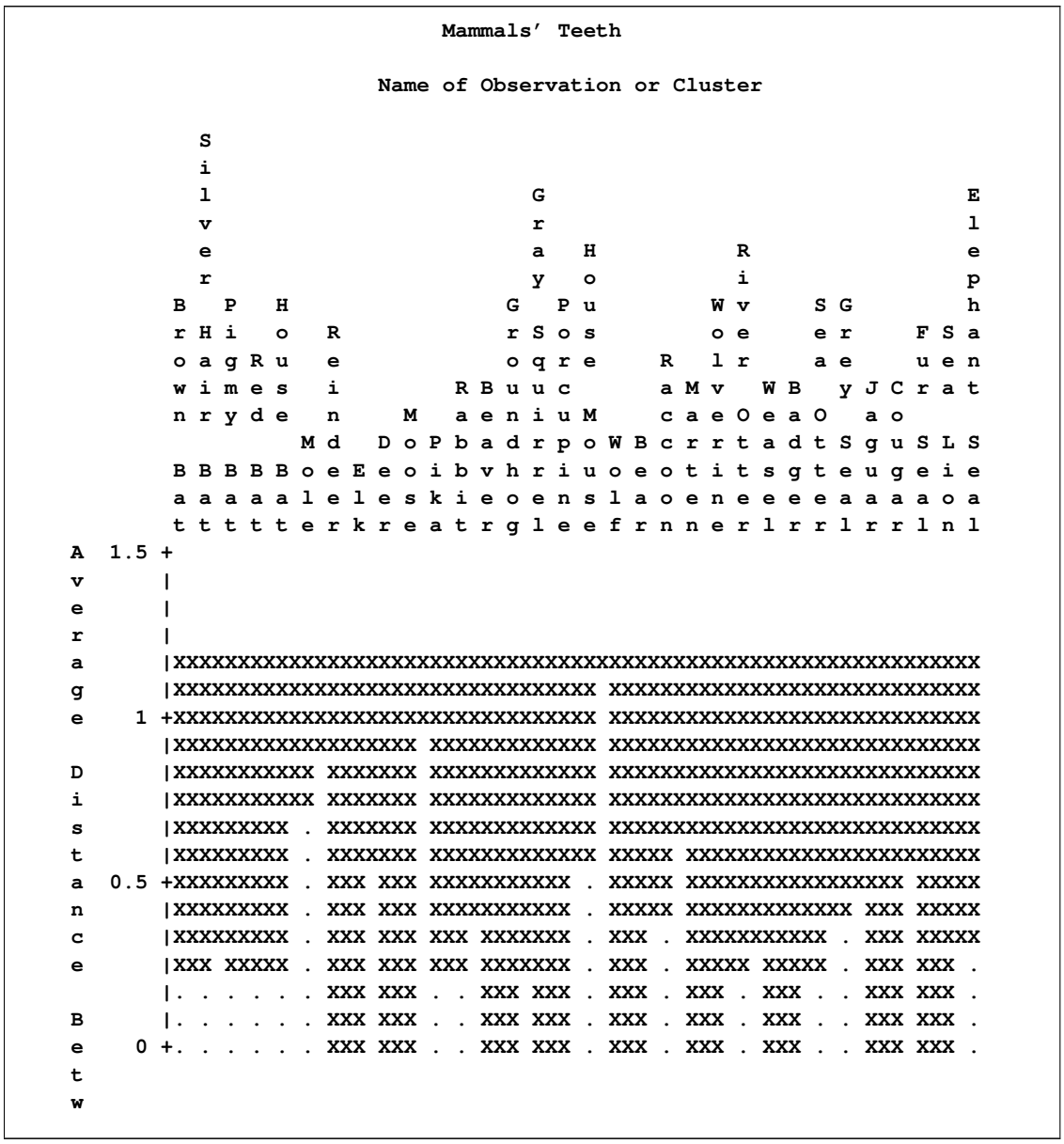
As you view the diagram in Output 91.1.2 from left to right, objects and clusters are progressively joined until a single, all-encompassing cluster is formed at the right (or root) of the tree. Clusters exist at each level of the diagram, and every vertical line connects leaves and branches into progressively larger clusters. For example, the five bats form a cluster at the 0.6 level, while the next cluster consists only of the mole. The observations `Reindeer`, `Elk`, `Deer`, and `Moose` form the next cluster at the 0.6 level, the mammals `Pika` through `House Mouse` are in the fourth cluster, The observations `Wolf`, `Bear`, and `Raccoon` form the fifth cluster, while the last cluster contains the observations `Marten` through `Elephant Seal`.

The following statements create the same tree with line printer graphics in a vertical orientation:

```
proc tree lineprinter;
run;
```

The tree is displayed in Output 91.1.3.

Output 91.1.3 PROC TREE with the LINEPRINTER Option



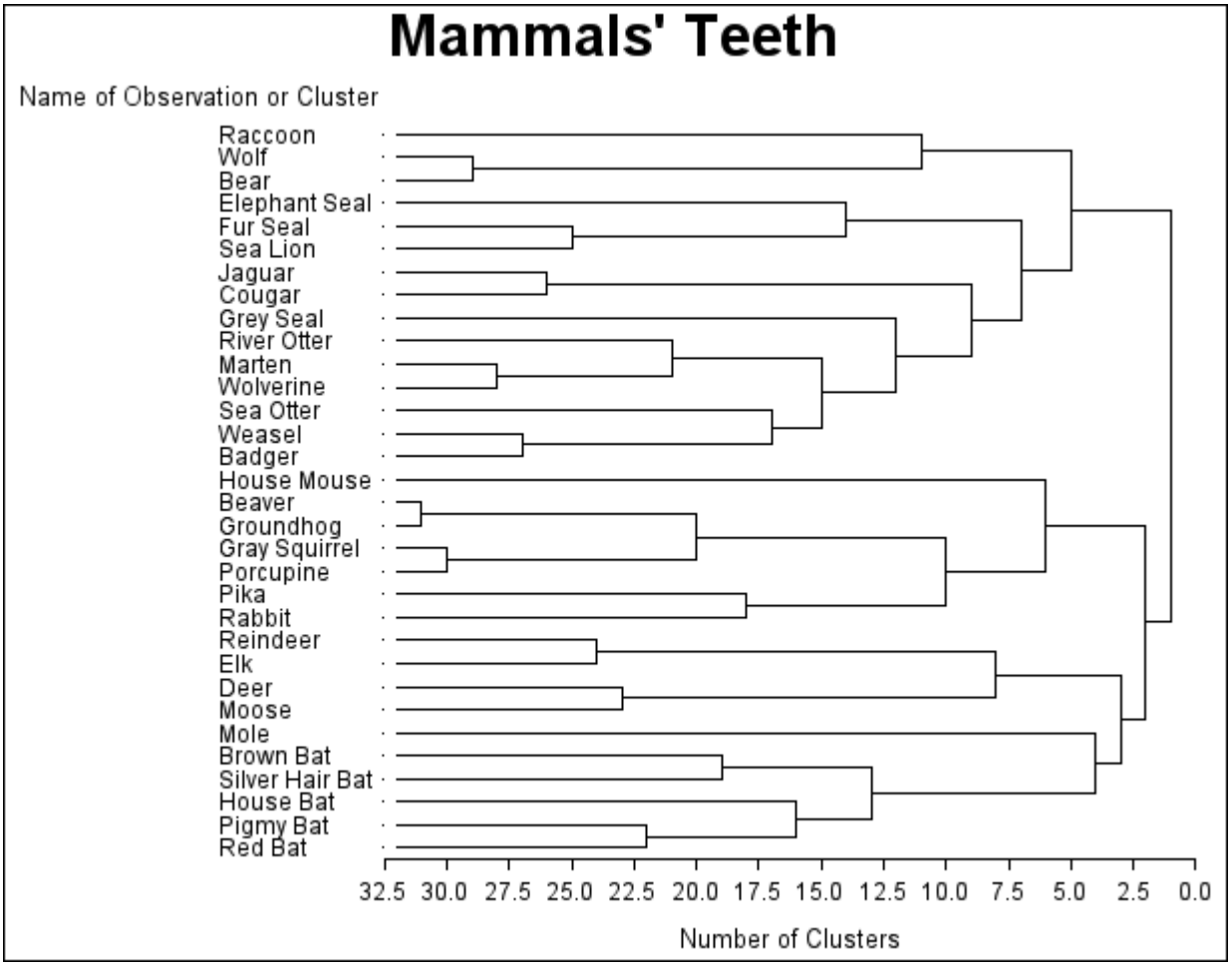
As you look up from the bottom of the diagram, objects and clusters are progressively joined until a single, all-encompassing cluster is formed at the top (or root) of the root. Clusters exist at each level of the diagram. For example, the unbroken line of Xs at the leftmost side of the 0.6 level indicates that the five bats have formed a cluster. The next cluster is represented by a period because it contains only one mammal, `Mole`. `Reindeer`, `Elk`, `Deer`, and `Moose` form the next cluster, indicated by Xs again. The mammals `Pika` through `House Mouse` are in the fourth cluster. The observations `Wolf`, `Bear`, and `Raccoon` form the fifth cluster, while the last cluster contains the observations `Marten` through `Elephant Seal`.

The next statements sort the clusters at each branch in order of formation and use the number of clusters as the height axis:

```
proc tree sort height=n horizontal;
run;
```

The resulting tree is displayed in Output 91.1.4.

Output 91.1.4 PROC TREE with SORT and HEIGHT= Options



Because the CLUSTER procedure always produces binary trees, the number of internal (root and branch) nodes in the tree is one less than the number of leaves. Therefore 31 clusters are formed from the 32 mammals in the input data set. These are represented by the 31 vertical line segments in the tree diagram, each at a different value along the horizontal axis.

As you examine the tree from left to right, the first vertical line segment is where `Beaver` and `Groundhog` are clustered and the number of clusters is 31. The next cluster is formed from `Gray Squirrel` and `Porcupine`. The third contains `Wolf` and `Bear`. Note how the tree graphically displays the clustering order information that was presented in tabular form by the CLUSTER procedure in Output 91.1.1.

The same clusters as in Output 91.1.2 and Output 91.1.3 can be seen at the six-cluster level of the tree diagram in Output 91.1.4, although the SORT and HEIGHT= options make them appear in a different order.

The following statements create these six clusters and save the result in the output data set part:

```
proc tree noprint out=part nclusters=6;
   id mammal;
   copy v1-v8;
proc sort;
   by cluster;
```

The PROC TREE statement produces no output but creates an output data set indicating the cluster to which each observation belongs at the six-cluster level in the tree. The following statements print part, with the results shown in Output 91.1.5:

```
proc print label uniform;
   id mammal;
   var v1-v8;
   format v1-v8 1.;
   by cluster;
run;
```

Output 91.1.5 PROC TREE OUT= Data Set

Mammals' Teeth

CLUSTER=1

mammal	Right Top Incisors	Right Bottom Incisors	Right Top Canines	Right Bottom Canines
Beaver	1	1	0	0
Groundhog	1	1	0	0
Gray Squirrel	1	1	0	0
Porcupine	1	1	0	0
Pika	2	1	0	0
Rabbit	2	1	0	0
House Mouse	1	1	0	0

mammal	Right Top Premolars	Right Bottom Premolars	Right Top Molars	Right Bottom Molars
Beaver	2	1	3	3
Groundhog	2	1	3	3
Gray Squirrel	1	1	3	3
Porcupine	1	1	3	3
Pika	2	2	3	3
Rabbit	3	2	3	3
House Mouse	0	0	3	3

Output 91.1.5 *continued*

Mammals' Teeth

CLUSTER=2

mammal	Right Top Incisors	Right Bottom Incisors	Right Top Canines	Right Bottom Canines
Wolf	3	3	1	1
Bear	3	3	1	1
Raccoon	3	3	1	1

mammal	Right Top Premolars	Right Bottom Premolars	Right Top Molars	Right Bottom Molars
Wolf	4	4	2	3
Bear	4	4	2	3
Raccoon	4	4	3	2

CLUSTER=3

mammal	Right Top Incisors	Right Bottom Incisors	Right Top Canines	Right Bottom Canines
Marten	3	3	1	1
Wolverine	3	3	1	1
Weasel	3	3	1	1
Badger	3	3	1	1

mammal	Right Top Premolars	Right Bottom Premolars	Right Top Molars	Right Bottom Molars
Marten	4	4	1	2
Wolverine	4	4	1	2
Weasel	3	3	1	2
Badger	3	3	1	2

Output 91.1.5 *continued*

Mammals' Teeth

CLUSTER=3 (continued)

mammal	Right Top Incisors	Right Bottom Incisors	Right Top Canines	Right Bottom Canines
Jaguar	3	3	1	1
Cougar	3	3	1	1
Fur Seal	3	2	1	1
Sea Lion	3	2	1	1
River Otter	3	3	1	1
Sea Otter	3	2	1	1
Elephant Seal	2	1	1	1
Grey Seal	3	2	1	1

mammal	Right Top Premolars	Right Bottom Premolars	Right Top Molars	Right Bottom Molars
Jaguar	3	2	1	1
Cougar	3	2	1	1
Fur Seal	4	4	1	1
Sea Lion	4	4	1	1
River Otter	4	3	1	2
Sea Otter	3	3	1	2
Elephant Seal	4	4	1	1
Grey Seal	3	3	2	2

Output 91.1.5 *continued*

Mammals' Teeth

CLUSTER=4

mammal	Right Top Incisors	Right Bottom Incisors	Right Top Canines	Right Bottom Canines
Reindeer	0	4	1	0
Elk	0	4	1	0
Deer	0	4	0	0
Moose	0	4	0	0

mammal	Right Top Premolars	Right Bottom Premolars	Right Top Molars	Right Bottom Molars
Reindeer	3	3	3	3
Elk	3	3	3	3
Deer	3	3	3	3
Moose	3	3	3	3

CLUSTER=5

mammal	Right Top Incisors	Right Bottom Incisors	Right Top Canines	Right Bottom Canines
Pigmy Bat	2	3	1	1
Red Bat	1	3	1	1
Brown Bat	2	3	1	1

mammal	Right Top Premolars	Right Bottom Premolars	Right Top Molars	Right Bottom Molars
Pigmy Bat	2	2	3	3
Red Bat	2	2	3	3
Brown Bat	3	3	3	3

Output 91.1.5 *continued*

Mammals' Teeth

CLUSTER=5
(continued)

mammal	Right Top Incisors	Right Bottom Incisors	Right Top Canines	Right Bottom Canines
Silver Hair Bat	2	3	1	1
House Bat	2	3	1	1

mammal	Right Top Premolars	Right Bottom Premolars	Right Top Molars	Right Bottom Molars
Silver Hair Bat	2	3	3	3
House Bat	1	2	3	3

CLUSTER=6

mammal	Right Top Incisors	Right Bottom Incisors	Right Top Canines	Right Bottom Canines
Mole	3	2	1	0

mammal	Right Top Premolars	Right Bottom Premolars	Right Top Molars	Right Bottom Molars
Mole	3	3	3	3

Example 91.2: Iris Data

Fisher (1936)'s iris data give sepal and petal dimensions for three different species of iris. The data are clustered by *k*th-nearest-neighbor density linkage by using the CLUSTER procedure with K=8. Observations are identified by species (`Setosa`, `Versicolor`, or `Virginica`) in the tree diagram, which is oriented with the height axis horizontal. The following statements produce the results shown in Output 91.2.1 and Output 91.2.2:

```
proc format;
   value specname
      1='Setosa    '
      2='Versicolor'
      3='Virginica ';
run;

data iris;
   title 'Fisher (1936) Iris Data';
   input SepalLength SepalWidth PetalLength PetalWidth
         Species @@;
   format Species specname.;
   label SepalLength='Sepal Length in mm.'
         SepalWidth ='Sepal Width in mm.'
         PetalLength='Petal Length in mm.'
         PetalWidth ='Petal Width in mm.';
   symbol = put(species, specname10.);
   datalines;
50 33 14 02 1 64 28 56 22 3 65 28 46 15 2 67 31 56 24 3
63 28 51 15 3 46 34 14 03 1 69 31 51 23 3 62 22 45 15 2
59 32 48 18 2 46 36 10 02 1 61 30 46 14 2 60 27 51 16 2
65 30 52 20 3 56 25 39 11 2 65 30 55 18 3 58 27 51 19 3
68 32 59 23 3 51 33 17 05 1 57 28 45 13 2 62 34 54 23 3
77 38 67 22 3 63 33 47 16 2 67 33 57 25 3 76 30 66 21 3
49 25 45 17 3 55 35 13 02 1 67 30 52 23 3 70 32 47 14 2
64 32 45 15 2 61 28 40 13 2 48 31 16 02 1 59 30 51 18 3
55 24 38 11 2 63 25 50 19 3 64 32 53 23 3 52 34 14 02 1
49 36 14 01 1 54 30 45 15 2 79 38 64 20 3 44 32 13 02 1
67 33 57 21 3 50 35 16 06 1 58 26 40 12 2 44 30 13 02 1
77 28 67 20 3 63 27 49 18 3 47 32 16 02 1 55 26 44 12 2
50 23 33 10 2 72 32 60 18 3 48 30 14 03 1 51 38 16 02 1
61 30 49 18 3 48 34 19 02 1 50 30 16 02 1 50 32 12 02 1
61 26 56 14 3 64 28 56 21 3 43 30 11 01 1 58 40 12 02 1
51 38 19 04 1 67 31 44 14 2 62 28 48 18 3 49 30 14 02 1
51 35 14 02 1 56 30 45 15 2 58 27 41 10 2 50 34 16 04 1
46 32 14 02 1 60 29 45 15 2 57 26 35 10 2 57 44 15 04 1
50 36 14 02 1 77 30 61 23 3 63 34 56 24 3 58 27 51 19 3
57 29 42 13 2 72 30 58 16 3 54 34 15 04 1 52 41 15 01 1
71 30 59 21 3 64 31 55 18 3 60 30 48 18 3 63 29 56 18 3
49 24 33 10 2 56 27 42 13 2 57 30 42 12 2 55 42 14 02 1
49 31 15 02 1 77 26 69 23 3 60 22 50 15 3 54 39 17 04 1
66 29 46 13 2 52 27 39 14 2 60 34 45 16 2 50 34 15 02 1
44 29 14 02 1 50 20 35 10 2 55 24 37 10 2 58 27 39 12 2
47 32 13 02 1 46 31 15 02 1 69 32 57 23 3 62 29 43 13 2
74 28 61 19 3 59 30 42 15 2 51 34 15 02 1 50 35 13 03 1
56 28 49 20 3 60 22 40 10 2 73 29 63 18 3 67 25 58 18 3
49 31 15 01 1 67 31 47 15 2 63 23 44 13 2 54 37 15 02 1
56 30 41 13 2 63 25 49 15 2 61 28 47 12 2 64 29 43 13 2
51 25 30 11 2 57 28 41 13 2 65 30 58 22 3 69 31 54 21 3
54 39 13 04 1 51 35 14 03 1 72 36 61 25 3 65 32 51 20 3
61 29 47 14 2 56 29 36 13 2 69 31 49 15 2 64 27 53 19 3
68 30 55 21 3 55 25 40 13 2 48 34 16 02 1 48 30 14 01 1
45 23 13 03 1 57 25 50 20 3 57 38 17 03 1 51 38 15 03 1
```

```
55 23 40 13 2 66 30 44 14 2 68 28 48 14 2 54 34 17 02 1
51 37 15 04 1 52 35 15 02 1 58 28 51 24 3 67 30 50 17 2
63 33 60 25 3 53 37 15 02 1
;

title 'Fisher (1936) Iris Data';
proc cluster data=iris method=twostage print=10
             outtree=tree k=8 noeigen;
   var SepalLength SepalWidth PetalLength PetalWidth;
   copy Species;
run;

proc tree data=tree horizontal lineprinter pages=1 maxh=10;
   id species;
run;
```

The PAGES=1 option specifies that the tree diagram extend over one page from tree to root. Since the HORIZONTAL option is also specified, the horizontal extent of the diagram is one page. The number of vertical pages required for the diagram is dictated by the number of leaves in the tree.

The MAXH=10 limits the values displayed on the height axis to a maximum of 10. This prunes the tree diagram so that only the portion from the leaves to level 10 is displayed. You can see this pruning effect in Output 91.2.2.

Output 91.2.1 Clustering of Fisher's Iris Data

Fisher (1936) Iris Data

The CLUSTER Procedure
Two-Stage Density Linkage Clustering

K = 8

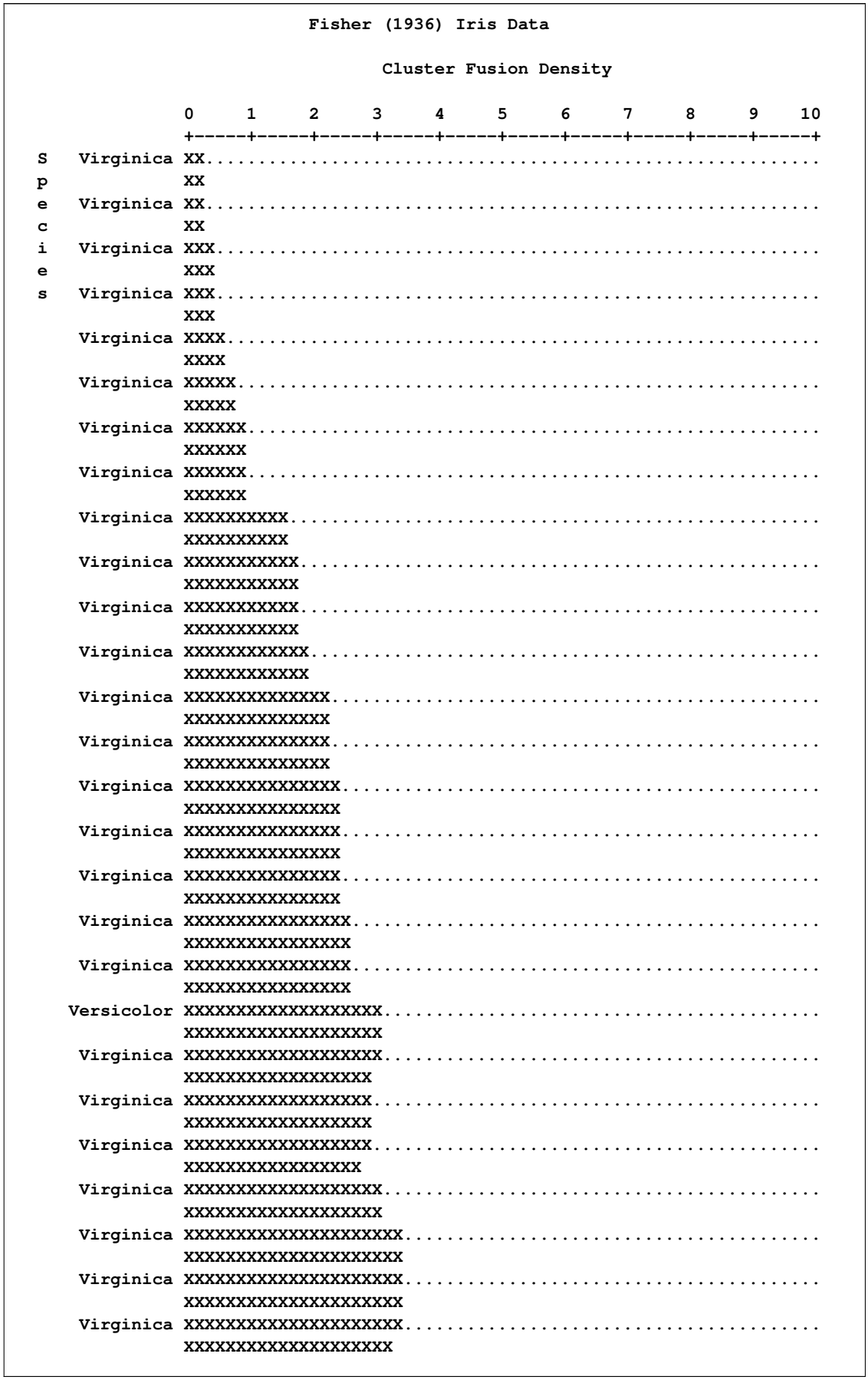
Root-Mean-Square Total-Sample Standard Deviation 10.69224

Cluster History

NCL	--Clusters Joined--		FREQ	Normalized Fusion Density	Maximum Density in Each Cluster Lesser	Greater	Tie
10	CL11	OB98	49	0.2879	0.1479	8.3678	
9	CL13	OB24	45	0.2802	0.2005	3.5156	
8	CL10	OB25	50	0.2699	0.1372	8.3678	
7	CL8	OB121	51	0.2586	0.1372	8.3678	
6	CL9	OB45	46	0.1412	0.0832	3.5156	
5	CL6	OB39	47	0.107	0.0605	3.5156	
4	CL5	OB21	48	0.0969	0.0541	3.5156	
3	CL4	OB90	49	0.0715	0.0370	3.5156	
2	CL7	CL3	100	2.6277	3.5156	8.3678	

3 modal clusters have been formed.

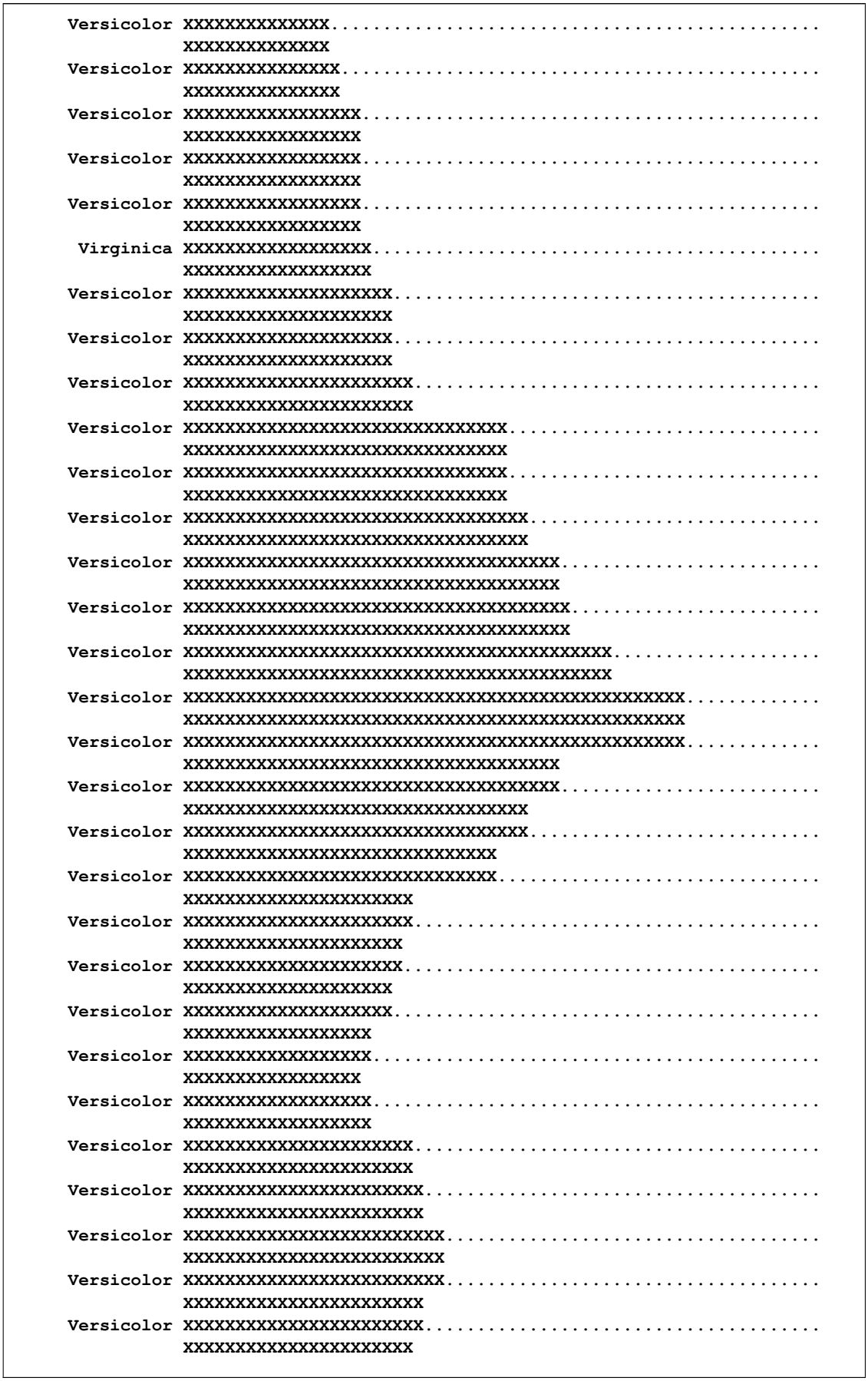
Output 91.2.2 Horizontal Tree for Fisher's Iris Data



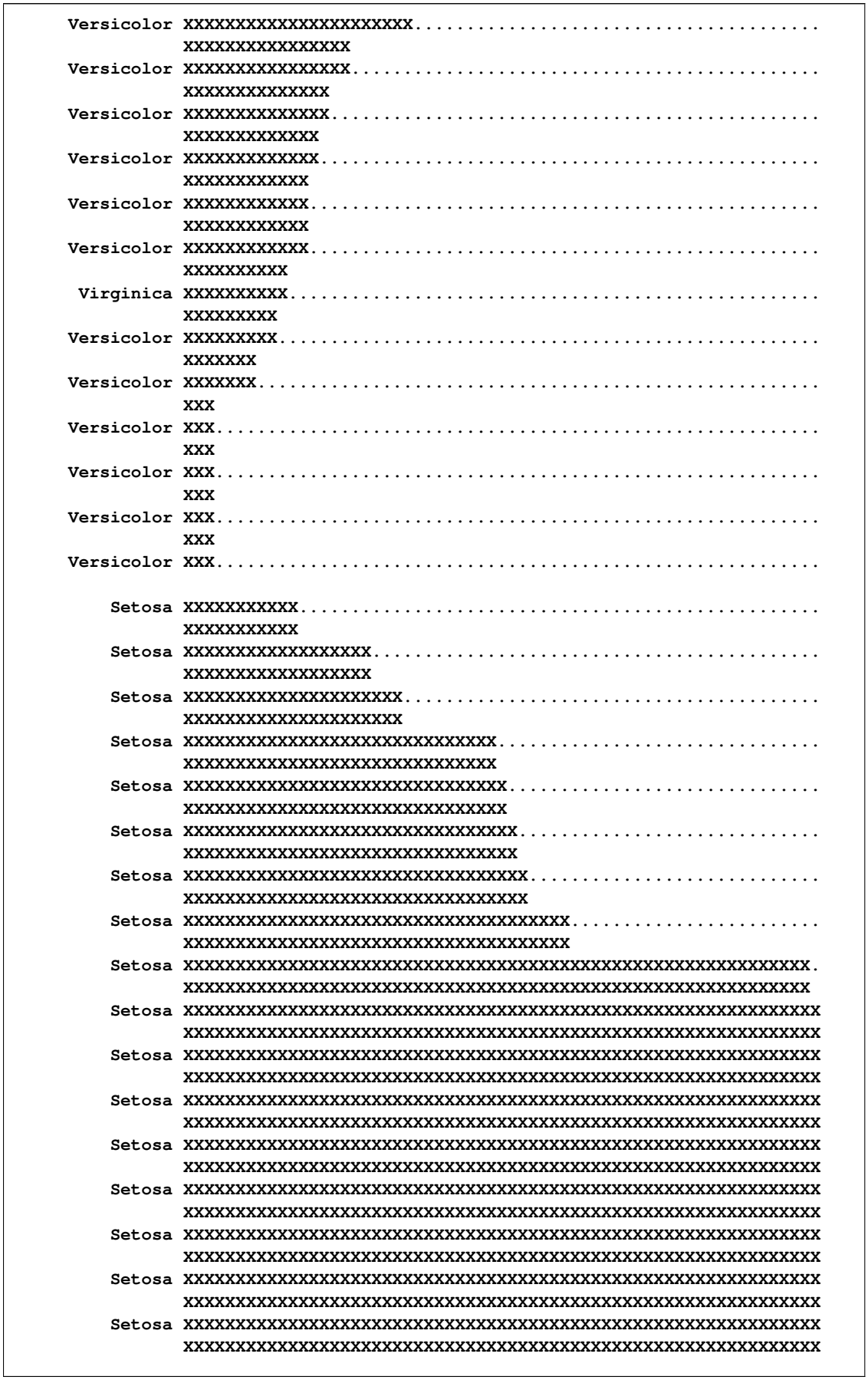
Output 91.2.2 *continued*

```
 Virginica XXXXXXXXXXXXXXXXXXX.....................................
           XXXXXXXXXXXXXXXXXXX
 Virginica XXXXXXXXXXXXXXXXXXX.....................................
           XXXXXXXXXXXXXXXXXX
 Virginica XXXXXXXXXXXXXXXXXX.....................................
           XXXXXXXXXXXXXXXX
 Virginica XXXXXXXXXXXXXXXX.....................................
           XXXXXXXXXXXXXXX
Versicolor XXXXXXXXXXXXXXX.....................................
           XXXXXXXXXXXXXXX
 Virginica XXXXXXXXXXXXXXX.....................................
           XXXXXXXXXXXXXX
 Virginica XXXXXXXXXXXXXX.....................................
           XXXXXXXXXXXXXX
 Virginica XXXXXXXXXXXXXX.....................................
           XXXXXXXXXXXXX
 Virginica XXXXXXXXXXXXX.....................................
           XXXXXXXXXXXX
 Virginica XXXXXXXXXXXX.....................................
           XXXXXXXXXXXX
 Virginica XXXXXXXXXXXX.....................................
           XXXXXXXXXXX
 Virginica XXXXXXXXXXX.....................................
           XXXXXXXXXX
 Virginica XXXXXXXXXX.....................................
           XXXXXXXX
 Virginica XXXXXXXX.....................................
           XXXXXXXX
 Virginica XXXXXXXX.....................................
           XXXXXX
 Virginica XXXXXX.....................................
           XXXXXX
 Virginica XXXXXX.....................................
           XXXXX
 Virginica XXXXX.....................................
           XXXX
 Virginica XXXX.....................................
           XXXX
 Virginica XXXX.....................................
           XXX
 Virginica XXX.....................................
           XX
 Virginica XX.....................................
           X
 Virginica X.....................................
           X
 Virginica XXX.....................................
           XXX
Versicolor XXXXX.....................................
           XXXXX
Versicolor XXXXXXXX.....................................
           XXXXXXXX
Versicolor XXXXXXXXX.....................................
           XXXXXXXXX
Versicolor XXXXXXXXXX.....................................
           XXXXXXXXXX
Versicolor XXXXXXXXXX.....................................
           XXXXXXXXXX
Versicolor XXXXXXXXXXX.....................................
           XXXXXXXXXXX
Versicolor XXXXXXXXXXXX.....................................
           XXXXXXXXXXXX
```

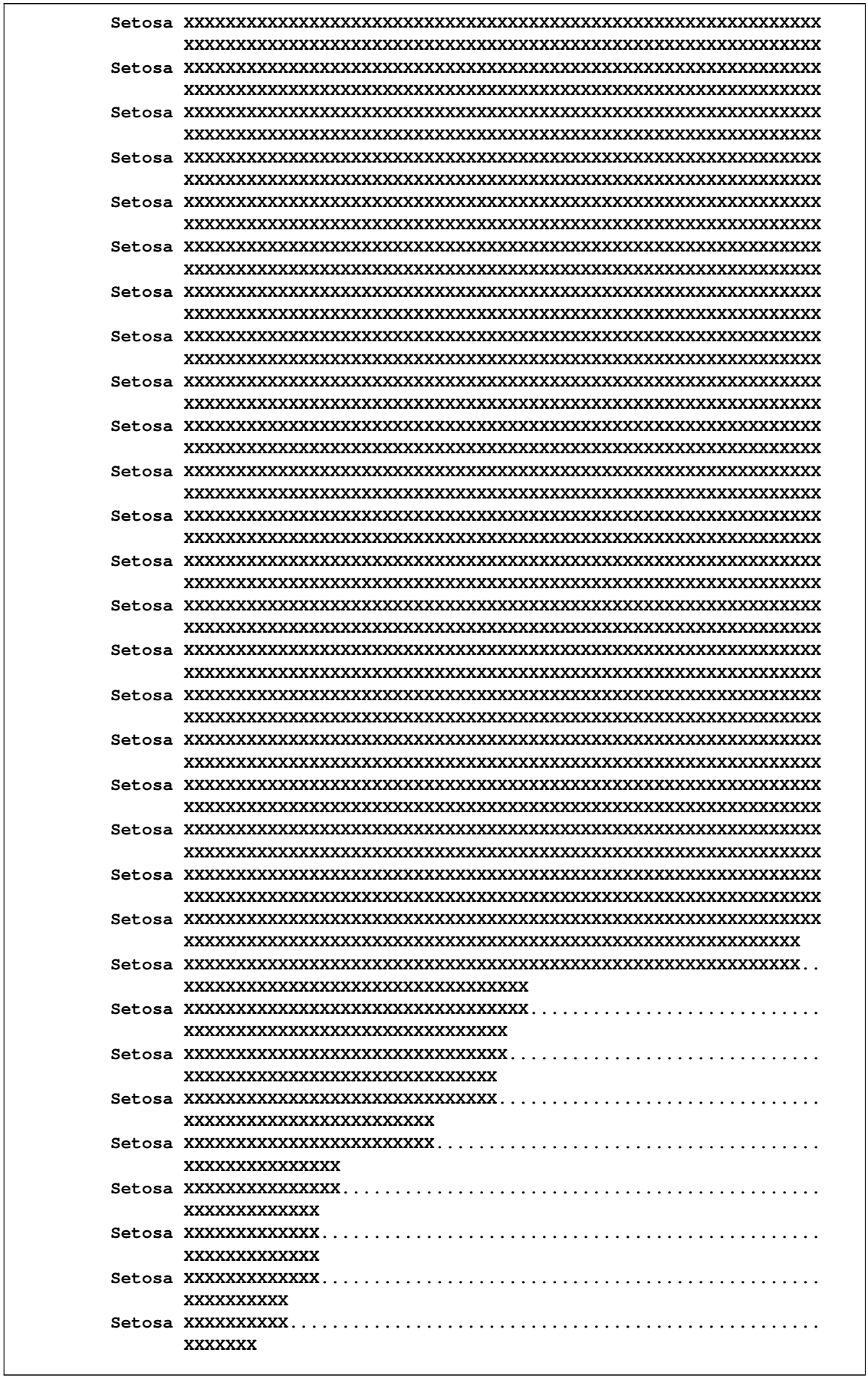
Output 91.2.2 *continued*



Output 91.2.2 *continued*



Output 91.2.2 *continued*


Setosa
Setosa
Setosa

Output 91.2.2 *continued*

```
Setosa XXXXXXX.....................................................
       XXXX
Setosa XXXX........................................................
       XXX
Setosa XXX.........................................................
```

References

Duran, B. S. and Odell, P. L. (1974), *Cluster Analysis*, New York: Springer-Verlag.

Everitt, B. S. (1980), *Cluster Analysis*, Second Edition, London: Heineman Educational Books.

Fisher, R. A. (1936), "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, 7, 179–188.

Hand, D. J., Daly, F., Lunn, A. D., McConway, K. J., and Ostrowski, E. (1994), *A Handbook of Small Data Sets*, London: Chapman & Hall.

Hartigan, J. A. (1975), *Clustering Algorithms*, New York: John Wiley & Sons.

Johnson, S. C. (1967), "Hierarchical Clustering Schemes," *Psychometrika*, 32, 241–254.

Knuth, D. E. (1973), *The Art of Computer Programming, Volume 1, Fundamental Algorithms*, Reading, MA: Addison-Wesley.

Chapter 92

The TTEST Procedure

Contents

Overview: TTEST Procedure **7392**
Getting Started: TTEST Procedure **7393**
One-Sample *t* Test 7393
Comparing Group Means 7396
Syntax: TTEST Procedure **7400**
PROC TTEST Statement 7401
BY Statement 7408
CLASS Statement 7408
FREQ Statement 7409
PAIRED Statement 7409
VAR Statement 7410
WEIGHT Statement 7410
Details: TTEST Procedure **7411**
Input Data Set of Statistics 7411
Missing Values 7411
Computational Methods 7411
Common Notation 7412
Arithmetic and Geometric Means 7412
Coefficient of Variation 7413
One-Sample Design 7413
Paired Design 7415
Two-Independent-Sample Design 7416
AB/BA Crossover Design 7421
TOST Equivalence Test 7422
Displayed Output 7423
ODS Table Names 7426
ODS Graphics 7427
ODS Graph Names 7427
Interpreting Graphs 7428
Examples: TTEST Procedure **7431**
Example 92.1: Using Summary Statistics to Compare Group Means 7431
Example 92.2: One-Sample Comparison with the FREQ Statement 7434
Example 92.3: Paired Comparisons 7437
Example 92.4: AB/BA Crossover Design 7442

Example 92.5: Equivalence Testing with Lognormal Data 7452
References 7457

Overview: TTEST Procedure

The TTEST procedure performs *t* tests and computes confidence limits for one sample, paired observations, two independent samples, and the AB/BA crossover design. Two-sided, TOST (two one-sided test) equivalence, and upper and lower one-sided hypotheses are supported for means, mean differences, and mean ratios for either normal or lognormal data.

Table 92.1 summarizes the designs, analysis criteria, hypotheses, and distributional assumptions supported in the TTEST procedure, along with the syntax used to specify them.

Table 92.1 Features Supported in the TTEST Procedure

Feature	Syntax
Design	
One-sample	VAR statement
Paired	PAIRED statement
Two-independent-sample	CLASS statement, VAR statement
AB/BA crossover	VAR / CROSSOVER
Analysis Criterion	
Mean difference	PROC TTEST TEST=DIFF
Mean ratio	PROC TTEST TEST=RATIO
Hypothesis	
Two-sided	PROC TTEST SIDES=2
Equivalence	PROC TTEST TOST (< *lower* , > *upper*)
Lower one-sided	PROC TTEST SIDES=L
Upper one-sided	PROC TTEST SIDES=U
Distribution	
Normal	PROC TTEST DIST=NORMAL
Lognormal	PROC TTEST DIST=LOGNORMAL

FREQ and WEIGHT statements are available. Data can be input in the form of observations or, in certain cases, summary statistics. Output includes summary statistics; confidence limits for means, standard deviations, and coefficients of variation; hypothesis tests; and a variety of graphical displays, including histograms, densities, box plots, confidence intervals, Q-Q plots, profiles, and agreement plots.

PROC TTEST now uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific informa-

tion about the statistical graphics available with the TTEST procedure, see the PLOTS option in the PROC TTEST statement and the section "ODS Graphics" on page 7427.

Getting Started: TTEST Procedure

One-Sample *t* Test

A one-sample *t* test can be used to compare a sample mean to a given value. This example, taken from Huntsberger and Billingsley (1989, p. 290), tests whether the mean length of a certain type of court case is more than 80 days by using 20 randomly chosen cases. The data are read by the following DATA step:

```
data time;
   input time @@;
   datalines;
 43  90  84  87  116   95  86   99   93  92
121  71  66  98   79  102  60  112  105  98
;
run;
```

The only variable in the data set, time, is assumed to be normally distributed. The trailing at signs (@@) indicate that there is more than one observation on a line. The following statements invoke PROC TTEST for a one-sample *t* test:

```
ods graphics on;

proc ttest h0=80 plots(showh0) sides=u alpha=0.1;
   var time;
run;

ods graphics off;
```

The ODS GRAPHICS statement requests graphical output. The VAR statement indicates that the time variable is being studied, while the H0= option specifies that the mean of the time variable should be compared to the null value 80 rather than the default of 0. The PLOTS(SHOWH0) option requests that this null value be displayed on all relevant graphs. The SIDES=U option reflects the focus of the research question, namely whether the mean court case length is *greater than* 80 days, rather than *different than* 80 days (in which case you would use the default SIDES=2 option). The ALPHA=0.1 option requests 90% confidence intervals rather than the default 95% confidence intervals. The output is displayed in Figure 92.1.

Figure 92.1 One-Sample *t* Test Results

The TTEST Procedure

Variable: time

N	Mean	Std Dev	Std Err	Minimum	Maximum
20	89.8500	19.1456	4.2811	43.0000	121.0

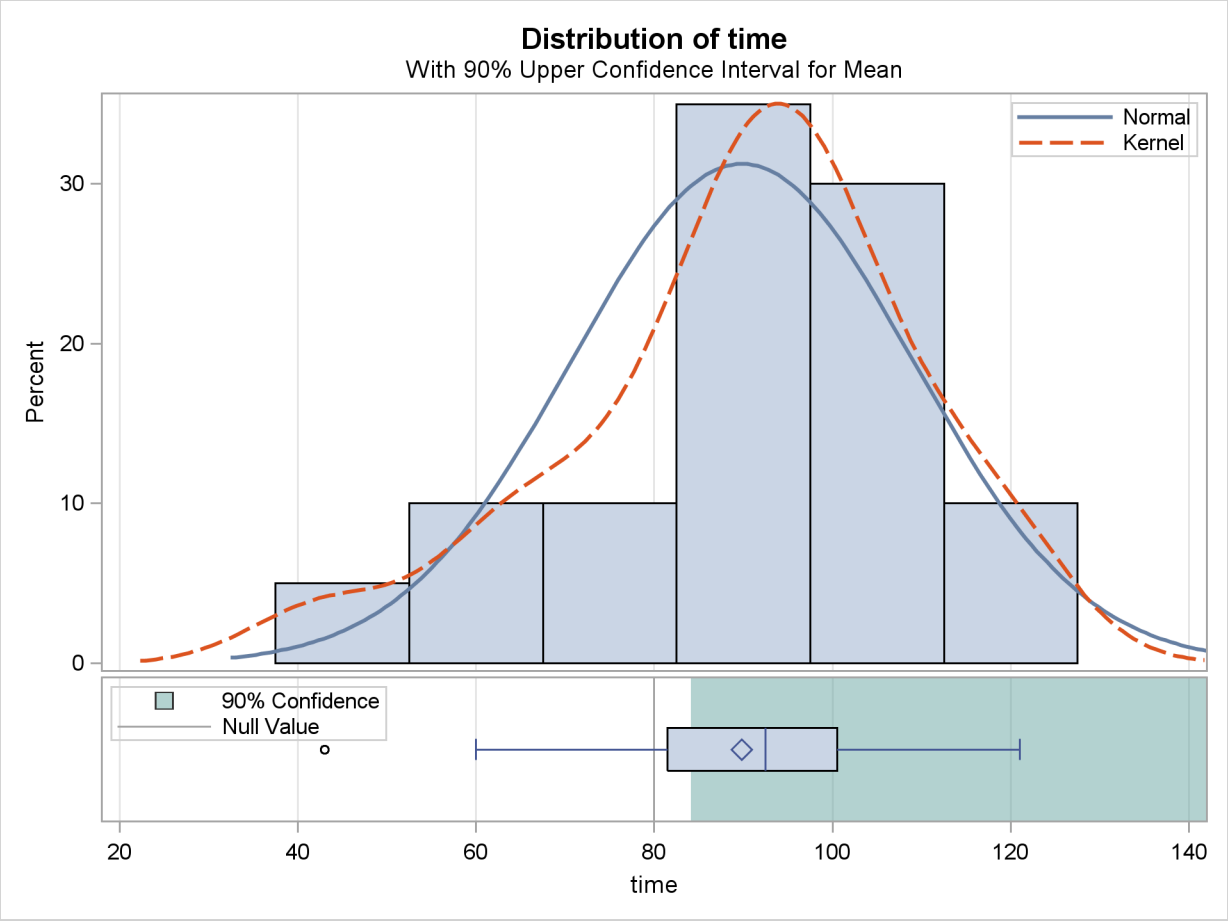
Mean	90% CL Mean		Std Dev	90% CL Std Dev	
89.8500	84.1659	Infty	19.1456	15.2002	26.2374

DF	t Value	Pr > t
19	2.30	0.0164

Summary statistics appear at the top of the output. The sample size (N), mean, standard deviation, and standard error are displayed with the minimum and maximum values of the time variable. The 90% confidence limits for the mean and standard deviation are shown next. Due to the SIDES=U option, the interval for the mean is an upper one-sided interval with a finite lower bound (84.1659 days). The limits for the standard deviation are the equal-tailed variety, per the default CI=EQUAL option in the PROC TTEST statement. At the bottom of the output are the degrees of freedom, t statistic value, and p-value for the t test. At the 10% α level, this test indicates that the mean length of the court cases is significantly greater than from 80 days ($t = 2.30$, $p = 0.0164$).

The summary panel in Figure 92.2 shows a histogram with overlaid normal and kernel densities, a box plot, the 90% confidence interval for the mean, and the null value of 80 days.

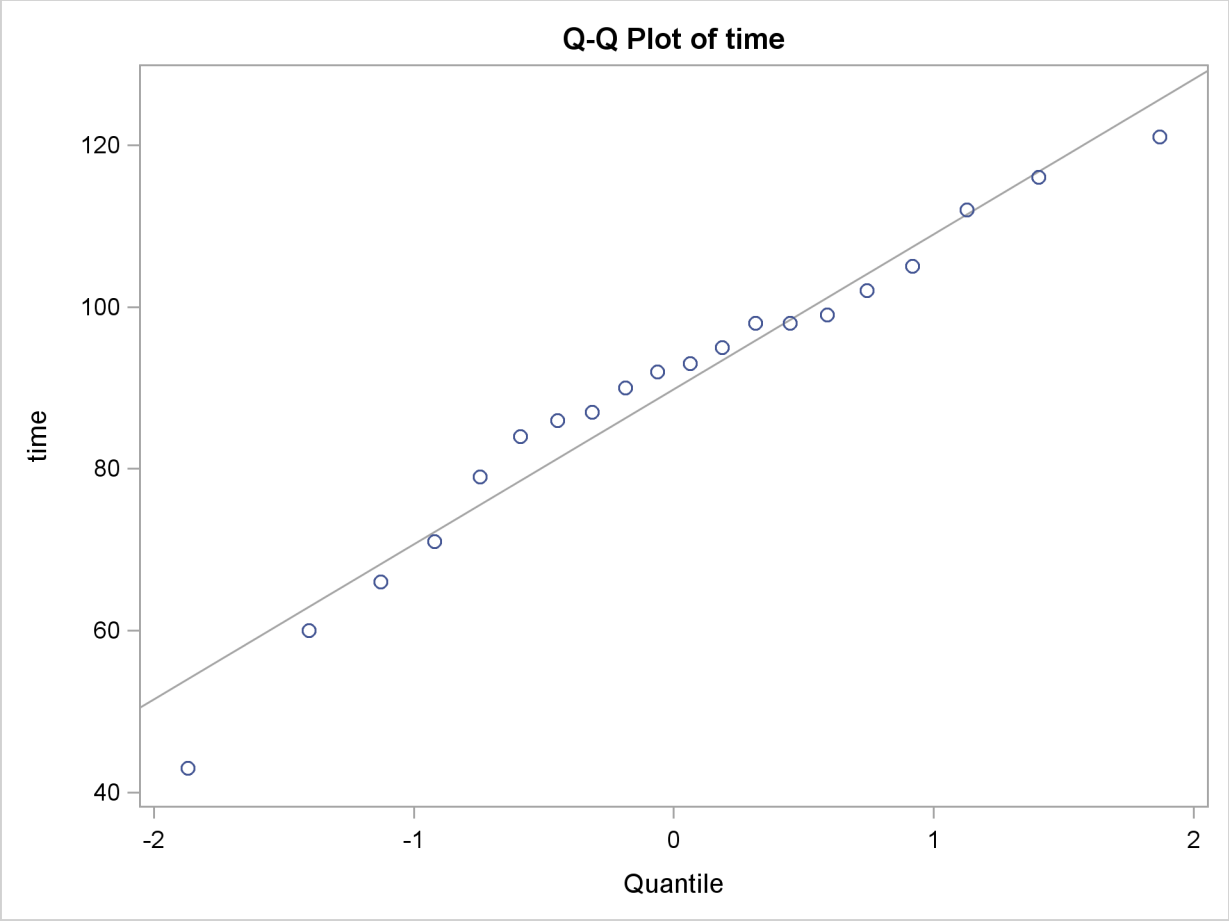
Figure 92.2 Summary Panel



The confidence interval excludes the null value, consistent with the rejection of the null hypothesis at $\alpha = 0.1$.

The Q-Q plot in Figure 92.3 assesses the normality assumption.

Figure 92.3 Q-Q Plot



The curvilinear shape of the Q-Q plot suggests a possible slight deviation from normality. You could use the UNIVARIATE procedure with the NORMAL option to numerically check the normality assumptions.

Comparing Group Means

If you want to compare values obtained from two different groups, and if the groups are independent of each other and the data are normally or lognormally distributed in each group, then a group *t* test can be used. Examples of such group comparisons include the following:

- test scores for two third-grade classes, where one of the classes receives tutoring
- fuel efficiency readings of two automobile nameplates, where each nameplate uses the same fuel
- sunburn scores for two sunblock lotions, each applied to a different group of people
- political attitude scores of males and females

In the following example, the golf scores for males and females in a physical education class are compared. The sample sizes from each population are equal, but this is not required for further analysis. The scores are thought to be approximately normally distributed within gender. The data are read by the following statements:

```
data scores;
   input Gender $ Score @@;
   datalines;
f 75  f 76  f 80  f 77  f 80  f 77  f 73
m 82  m 80  m 85  m 85  m 78  m 87  m 82
;
run;
```

The dollar sign ($) following Gender in the INPUT statement indicates that Gender is a character variable. The trailing at signs (@@) enable the procedure to read more than one observation per line.

You can use a group t test to determine whether the mean golf score for the men in the class differs significantly from the mean score for the women. If you also suspect that the distributions of the golf scores of males and females have unequal variances, then you might want to specify the COCHRAN option in order to use the Cochran approximation (in addition to the Satterthwaite approximation, which is included by default). The following statements invoke PROC TTEST for the case of unequal variances, along with both types of confidence limits for the pooled standard deviation.

```
ods graphics on;

proc ttest cochran ci=equal umpu;
   class Gender;
   var Score;
run;

ods graphics off;
```

The ODS GRAPHICS statement requests graphical output. The CLASS statement contains the variable that distinguishes the groups being compared, and the VAR statement specifies the response variable to be used in calculations. The COCHRAN option produces p-values for the unequal variance situation by using the Cochran and Cox (1950) approximation. Equal-tailed and uniformly most powerful unbiased (UMPU) confidence intervals for σ are requested by the CI= option. Output from these statements is displayed in Figure 92.4 through Figure 92.7.

Figure 92.4 Simple Statistics

The TTEST Procedure

Variable: Score

Gender	N	Mean	Std Dev	Std Err	Minimum	Maximum
f	7	76.8571	2.5448	0.9619	73.0000	80.0000
m	7	82.7143	3.1472	1.1895	78.0000	87.0000
Diff (1-2)		-5.8571	2.8619	1.5298		

Simple statistics for the two populations being compared, as well as for the difference of the means between the populations, are displayed in Figure 92.4. The Gender column indicates the population corresponding to the statistics in that row. The sample size (N), mean, standard deviation, standard error, and minimum and maximum values are displayed.

Confidence limits for means and standard deviations are shown in Figure 92.5.

Figure 92.5 Simple Statistics

Gender	Method	Mean	95% CL Mean		Std Dev
f		76.8571	74.5036	79.2107	2.5448
m		82.7143	79.8036	85.6249	3.1472
Diff (1-2)	Pooled	-5.8571	-9.1902	-2.5241	2.8619
Diff (1-2)	Satterthwaite	-5.8571	-9.2064	-2.5078	

Gender	Method	95% CL Std Dev		95% UMPU CL Std Dev	
f		1.6399	5.6039	1.5634	5.2219
m		2.0280	6.9303	1.9335	6.4579
Diff (1-2)	Pooled	2.0522	4.7242	2.0019	4.5727
Diff (1-2)	Satterthwaite				

For the mean differences, both pooled (assuming equal variances for males and females) and Satterthwaite (assuming unequal variances) 95% intervals are shown. The confidence limits for the standard deviations are of the equal-tailed variety.

The test statistics, associated degrees of freedom, and p-values are displayed in Figure 92.6.

Figure 92.6 *t* Tests

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	12	-3.83	0.0024
Satterthwaite	Unequal	11.496	-3.83	0.0026
Cochran	Unequal	6	-3.83	0.0087

The Method column denotes which t test is being used for that row, and the Variances column indicates what assumption about variances is being made. The pooled test assumes that the two populations have equal variances and uses degrees of freedom $n_1 + n_2 - 2$, where n_1 and n_2 are the sample sizes for the two populations. The remaining two tests do not assume that the populations have equal variances. The Satterthwaite test uses the Satterthwaite approximation for degrees of freedom, while the Cochran test uses the Cochran and Cox approximation for the p-value. All three tests result in highly significant p-values, supporting the conclusion of a significant difference between males' and females' golf scores.

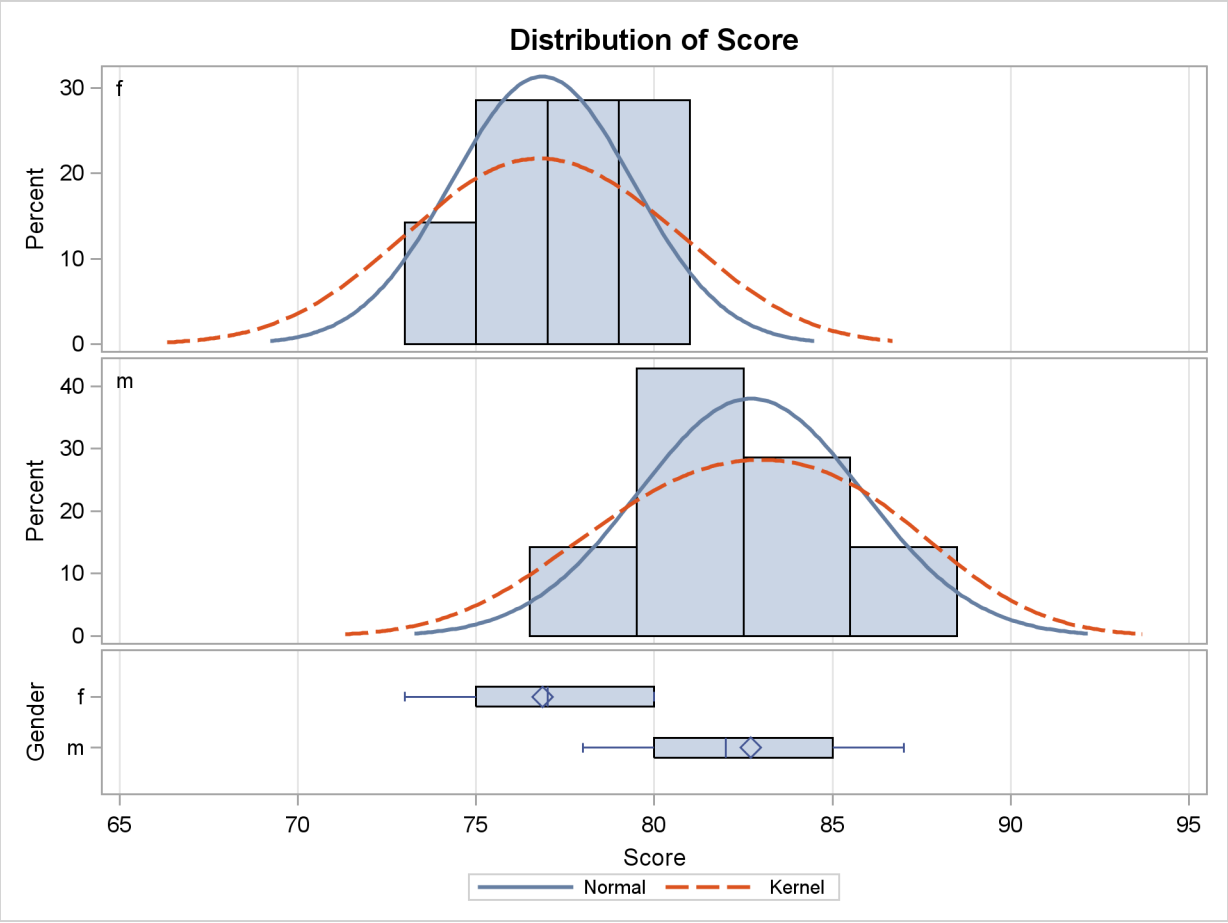
The "Equality of Variances" test in Figure 92.7 reveals insufficient evidence of unequal variances (the Folded F statistic $F' = 1.53$, with $p = 0.6189$).

Figure 92.7 Tests of Equality of Variances

Equality of Variances				
Method	Num DF	Den DF	F Value	Pr > F
Folded F	6	6	1.53	0.6189

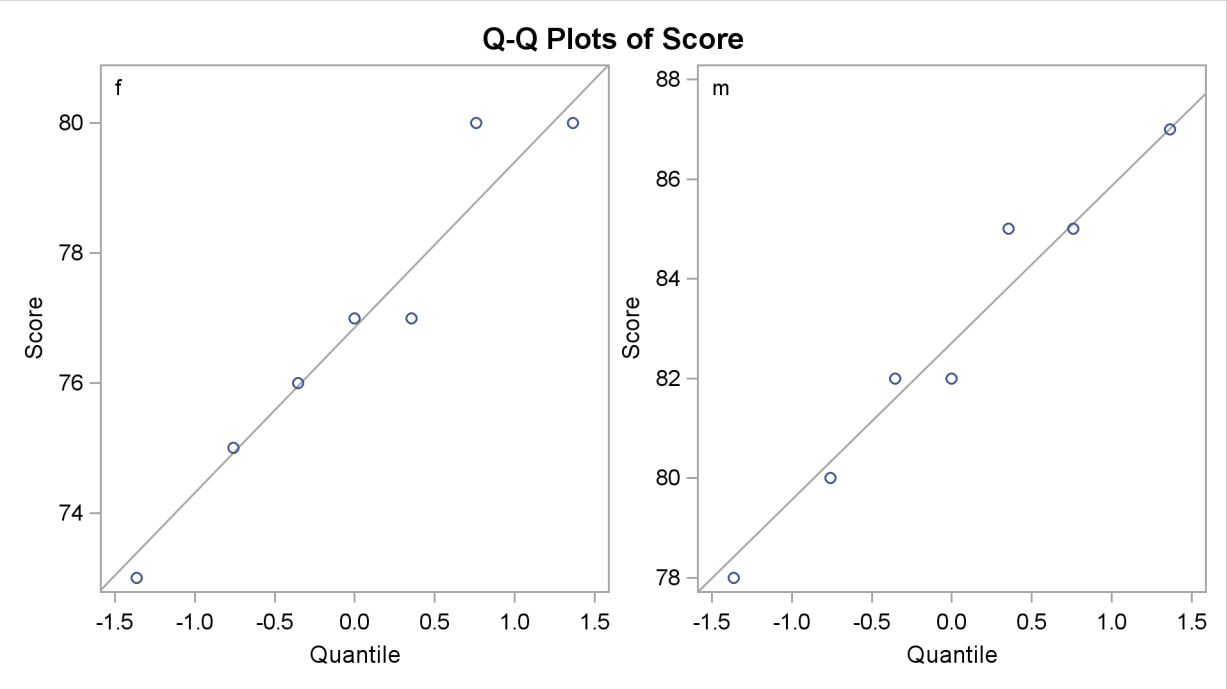
The summary panel in Figure 92.8 shows comparative histograms, normal and kernel densities, and box plots, comparing the distribution of golf scores between genders.

Figure 92.8 Summary Panel



The Q-Q plots in Output 92.9 assess the normality assumption for each gender.

Figure 92.9 Q-Q Plot



The plots for both males and females show no obvious deviations from normality. You can check the assumption of normality more rigorously by using PROC UNIVARIATE with the NORMAL option; if the assumption of normality is not reasonable, you should analyze the data with the nonparametric Wilcoxon rank sum test by using PROC NPAR1WAY.

Syntax: TTEST Procedure

The following statements are available in PROC TTEST:

PROC TTEST < *options* > ;
  CLASS *variable* ;
  PAIRED *variables* ;
  BY *variables* ;
  VAR *variables* < / *options* > ;
  FREQ *variable* ;
  WEIGHT *variable* ;

No statement can be used more than once. There is no restriction on the order of the statements after the PROC TTEST statement.

PROC TTEST Statement

PROC TTEST < *options* > **;**

The PROC TTEST statement invokes the procedure. Table 92.2 summarizes the options in the PROC TTEST statement by function. The options are then described fully in alphabetical order.

Table 92.2 PROC TTEST Statement Options

Option	Description
Basic Options	
DATA=	specifies input data set
ORDER=	determines sort order of CLASS variable or CROSSOVER treatment variables
Analysis Options	
ALPHA=	specifies 1− confidence level
DIST=	specifies distributional assumption (normal or lognormal)
H0=	specifies null value
SIDES=	specifies number of sides and direction
TEST=	specifies test criterion (difference or ratio)
TOST	requests equivalence test and specifies bounds
Displayed Output	
CI=	requests confidence interval for standard deviation or CV
COCHRAN	requests Cochran t test
PLOTS	produces ODS statistical graphics
Output Ordering	
BYVAR	groups results by PAIRED or VAR variables
NOBYVAR	groups results by tables

The following options can appear in the PROC TTEST statement.

ALPHA=*p*

specifies that confidence intervals (except test-based mean confidence intervals when the TOST option is used) are to be $100(1-p)\%$ confidence intervals, where $0 < p < 1$. When the TOST option is used, the test-based mean confidence intervals are $100(1-2p)\%$ confidence intervals. By default, PROC TTEST uses ALPHA=0.05. If p is 0 or less, or 1 or more, an error message is printed.

BYVAR

groups the results by the PAIRED or VAR variables. The BYVAR option is enabled by default. Note that this represents a change from previous releases for how the results are grouped

with respect to variables and tables. Prior to SAS 9.2, multiple variables were included in each table, similar to the new NOBYVAR option.

CI=EQUAL | UMPU | NONE

CL=EQUAL | UMPU | NONE

specifies whether a confidence interval is displayed for σ and, if so, what kind. The CI=EQUAL option specifies an equal-tailed confidence interval, and it is the default. The CI=UMPU option specifies an interval based on the uniformly most powerful unbiased test of $H_0: \sigma = \sigma_0$. The CI=NONE option requests that no confidence interval be displayed for σ. The values EQUAL and UMPU together request that both types of confidence intervals be displayed. If the value NONE is specified with one or both of the values EQUAL and UMPU, NONE takes precedence. For more information, see the section "Two-Independent-Sample Design" on page 7416.

COCHRAN

requests the Cochran and Cox (1950) approximation of the probability level for the unequal variances situation. For more information, see the section "Two-Independent-Sample Design" on page 7416.

DATA=*SAS-data-set*

names the SAS data set for the procedure to use. By default, PROC TTEST uses the most recently created SAS data set. The input data set can contain summary statistics of the observations instead of the observations themselves. The number, mean, and standard deviation of the observations are required for each BY group (one sample and paired differences) or for each class within each BY group (two samples). For more information about the DATA= option, see the section "Input Data Set of Statistics" on page 7411.

DIST=LOGNORMAL | NORMAL

specifies the underlying distribution assumed for the data. The default is NORMAL, unless TEST=RATIO is specified, in which case the default is LOGNORMAL.

H0=*m*

requests tests against a null value of m, unless the TOST option is used, in which case m is merely used to derive the lower and upper equivalence bounds. For the crossover design, the value m applies for both treatment and period tests. By default, PROC TTEST uses H0=0 when TEST=DIFF (or DIST=NORMAL for a one-sample design) and H0=1 when TEST=RATIO (or DIST=LOGNORMAL for a one-sample design).

NOBYVAR

includes all PAIRED or VAR variables together in each output table. If the NOBYVAR option is not specified, then the BYVAR option is enabled, grouping the results by the PAIRED and VAR variables.

ORDER=DATA | FORMATTED | FREQ | INTERNAL | MIXED

specifies the sorting order for the levels of the classification variable (specified in the CLASS statement) and treatment variables (specified in the CROSSOVER option in the VAR statement). The default is ORDER=MIXED, which corresponds to the ordering in releases previous to SAS 9.2.

The following table shows how PROC TTEST interprets values of the ORDER= option.

Value of ORDER=	Levels Sorted By
DATA	order of appearance in the input data set
FORMATTED	external formatted value, except for numeric variables with no explicit format, which are sorted by their unformatted (internal) value
FREQ	descending frequency count; levels with the most observations come first in the order. In the event of a tie, ORDER=MIXED is used.
INTERNAL	unformatted value
MIXED	same as ORDER=FORMATTED if the unformatted variable is character-valued; same as ORDER=INTERNAL otherwise (unformatted variable is numeric-valued)

For FORMATTED and INTERNAL, the sort order is machine dependent. For more information about sorting order, see the chapter on the SORT procedure in the *Base SAS Procedures Guide* and the discussion of BY-group processing in *SAS Language Reference: Concepts*.

PLOTS **<** **(***global-plot-options***)** **>** **<** **=** *plot-request* **<** *(options)* **>** **>**

PLOTS **<** **(***global-plot-options***)** **>** **<** **=** **(***plot-request* **<** *(options)* **>** **<** **...** *plot-request* **<** *(options)* **>** **>)** **>**

controls the plots produced through ODS Graphics. When you specify only one plot request, you can omit the parentheses around the plot request. Here are some examples:

```
plots=none
plots=(histogram boxplot interval qq profiles agreement)
plots(unpack)=summary
plots(showh0)=interval(type=pergroup)
plots=(summary(unpack) interval(type=period))
```

You must enable ODS Graphics before requesting plots, for example, like this:

```
ods graphics on;

proc ttest plots=all;
   var oxygen;
run;

ods graphics off;
```

For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." If you have enabled ODS Graphics but do not specify the PLOTS option, then PROC TTEST produces a default set of plots. (**NOTE:** The graphical results are unavailable if your input data set contains summary statistics rather than observation values.)

For a one-sample design, the default plots are the following:

- summary plot (histogram with overlaid normal and kernel densities, box plot, and confidence interval band)
- Q-Q plot

For a two-independent-sample design, the default plots are the following:

- summary plot (comparative histograms with overlaid densities and box plots)
- Q-Q plot

For a paired design, the default plots are the following:

- summary plot (histogram, densities, box plot, and confidence interval) of the difference or ratio
- Q-Q plot of the difference or ratio
- profiles plot
- agreement plot

For a crossover design, the default plots are the following:

- comparative histograms with overlaid densities by treatment and period
- comparative box plots by treatment and period
- Q-Q plots by treatment and period
- profiles over treatment plot
- agreement of treatments plot

For more detailed descriptions of plots, see the section "Interpreting Graphs" on page 7428.

The global plot options include the following:

ONLY
suppresses the default plots. Only plots specifically requested are displayed.

SHOWH0
SHOWNULL
shows the null value (as specified by the H0= option in the PROC TTEST statement) in all relevant plots. For one-sample and paired designs, the null value can appear in SUMMARY, BOX, and INTERVAL. For two-independent-sample and crossover designs, the null value can appear only in INTERVAL.

UNPACKPANEL
UNPACK
suppresses paneling. By default, multiple plots can appear in some output panels. Specify UNPACKPANEL to get each plot in a separate panel. You can specify PLOTS(UNPACKPANEL) to unpack the default plots. You can also specify UNPACK as a suboption with SUMMARY.

The plot requests include the following:

ALL
produces all appropriate plots. You can specify other options with ALL; for example, to request all plots and specify that intervals should be for the period difference in a crossover design, specify PLOTS=(ALL INTERVAL(TYPE=PERIOD)).

AGREEMENT

AGREEMENTPLOT

requests an agreement plot. This plot is produced by default for paired and crossover designs, the only designs for which the AGREEMENT option is valid.

For paired designs, the second response in each pair is plotted against the first response. See the section "Agreement Plots for Paired Designs" on page 7428 for further details.

For crossover designs, the AGREEMENT plot request has the following options:

TYPE=PERIOD

plots the response in the second period against the response in the first period. See the section "Period Agreement Plots for Crossover Designs" on page 7428 for further details.

TYPE=TREATMENT

plots the response associated with the second treatment against the response associated with the first treatment. This is the default TYPE= option for crossover designs. See the section "Treatment Agreement Plots for Crossover Designs" on page 7429 for further details.

BOX

BOXPLOT

requests a box plot or comparative box plots. This plot is produced by default for crossover designs. For other designs, a box plot appears as part of the SUMMARY plot by default.

For one-sample and paired designs, a confidence interval for the mean is shown as a band in the background, along with the equivalence bounds if the TOST option is used in the PROC TTEST statement.

For a two-independent-sample design, comparative box plots (one for each class) are shown. For a crossover design, comparative box plots for all four combinations of the two treatments and two periods are shown.

See the section "Box Plots" on page 7429 for further details.

HISTOGRAM

HIST

HISTDENS

requests a histogram or comparative histograms with overlaid normal and kernel densities. This plot is produced by default for crossover designs. For other designs, it appears as part of the SUMMARY plot by default.

For one-sample and paired designs, the histogram and densities are based on the test criterion (which is the mean difference or ratio for a paired design). For a two-independent-sample design, comparative histograms (one for each class) are shown. For a crossover design, histograms for all four combinations of the two treatments and two periods are shown.

See the section "Histograms" on page 7429 for further details.

INTERVAL

INTERVALPLOT

requests plots of confidence interval for means.

For a two-independent-sample design, the INTERVAL plot request has the following options:

TYPE=PERGROUP

shows two separate two-sided confidence intervals, one for each class. This option cannot be used along with the SHOWH0 global plot option.

TYPE=TEST

shows pooled and Satterthwaite confidence intervals. This is the default TYPE= option for two-independent-sample designs.

For a crossover design, The INTERVAL plot request has the following options:

TYPE=PERGROUP

shows four separate two-sided intervals, one for each treatment-by-period combination. This option cannot be used along with the SHOWH0 global plot option.

TYPE=PERIOD

shows pooled and Satterthwaite confidence intervals for the period difference or ratio. This option is invalid if the IGNOREPERIOD option is used in the VAR statement.

TYPE=TREATMENT

shows pooled and Satterthwaite confidence intervals for the treatment difference or ratio. This is the default TYPE= option for crossover designs.

See the section "Confidence Intervals" on page 7429 for further details.

NONE

suppresses all plots.

PROFILES

PROFILESPLOT

requests a profiles plot. This plot is produced by default for paired and crossover designs, the only designs for which the PROFILES option is valid.

For paired designs, a line is drawn for each observation from left to right connecting the first response to the second response. See the section "Profiles for Paired Designs" on page 7430 for further details.

For crossover designs, the PROFILES plot request has the following options:

TYPE=PERIOD

shows response profiles over period, connecting the first period on the left to the second period on the right for each subject. See the section "Profiles over Period for Crossover Designs" on page 7430 for further details.

TYPE=TREATMENT
shows response profiles over treatment values, connecting the first treatment on the left to the second treatment on the right for each observation. This is the default TYPE= option for crossover designs. See the section "Profiles over Treatment for Crossover Designs" on page 7430 for further details.

QQ
QQPLOT
requests a normal quantile-quantile (Q-Q) plot. This plot is produced by default for all designs.

For two-sample designs, separate plots are shown for each class in a single panel. For crossover design, separate plots are shown for each treatment-by-period combination in a single panel.

See the section "Q-Q Plots" on page 7430 for further details.

SUMMARY
SUMMARYPLOT
requests HISTOGRAM and BOX plots together in a single panel, sharing common X axes. This plot is produced by default for one-sample, paired, and two-independent-sample designs, the only designs for which the SUMMARY option is valid. See the documentation for the BOX and HISTOGRAM plot requests for details. The SUMMARY plot request has the following option:

UNPACK
plots histograms with overlaid densities in one panel and box plots (along with confidence interval bands, if one-sample or paired design) in another. Note that specifying PLOTS(ONLY)=SUMMARY(UNPACK) is exactly the same as specifying PLOTS(ONLY)=(BOX HISTOGRAM).

SIDES=2 | L | U
SIDED=2 | L | U
SIDE=2 | L | U
specifies the number of sides (or tails) and direction of the statistical tests and test-based confidence intervals. The values are interpreted as follows:

SIDES=2 (the default) specifies two-sided tests and confidence intervals for means.

SIDES=L specifies lower one-sided tests, in which the alternative hypothesis indicates a mean less than the null value, and lower one-sided confidence intervals between minus infinity and the upper confidence limit.

SIDES=U specifies upper one-sided tests, in which the alternative hypothesis indicates a mean greater than the null value, and upper one-sided confidence intervals between the lower confidence limit and infinity.

TEST=DIFF | RATIO
specifies the test criterion. Use TEST=DIFF to test the difference of means and TEST=RATIO to test the ratio of means. The default is DIFF, unless DIST=LOGNORMAL

is specified, in which case the default is RATIO. This option is ignored for one-sample designs.

TOST (< *lower* **, >** *upper* **)**
requests Schuirman's TOST equivalence test. The *upper* equivalence bound must be specified. If TEST=DIFF, then the default value for the *lower* equivalence bound is $2m-$*upper*, where m is the value of the H0= option. If TEST=RATIO, then the default value for *lower* is $m/$*upper*.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC TTEST to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the TTEST procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variable* **;**

A CLASS statement giving the name of the classification (or grouping) variable must accompany the PROC TTEST statement in the two-independent-sample case. It should be omitted for the one-sample, paired, and AB/BA crossover designs. If it is used without the VAR statement, all numeric variables in the input data set (except those appearing in the CLASS, BY, FREQ, or WEIGHT statement) are included in the analysis.

The classification variable must have two, and only two, levels. PROC TTEST divides the observations into the two groups for the t test by using the levels of this variable. You can use either a numeric or a character variable in the CLASS statement.

Class levels are determined from the formatted values of the CLASS variable. Thus, you can use formats to define group levels. See the discussions of the FORMAT procedure, the FORMAT statement, formats, and informats in *SAS Language Reference: Dictionary*.

FREQ Statement

FREQ *variable* **;**

The *variable* in the FREQ statement identifies a variable that contains the frequency of occurrence of each observation. PROC TTEST treats each observation as if it appears n times, where n is the value of the FREQ variable for the observation. If the value is not an integer, only the integer portion is used. If the frequency value is less than 1 or is missing, the observation is not used in the analysis. When the FREQ statement is not specified, each observation is assigned a frequency of 1. The FREQ statement cannot be used if the DATA= data set contains statistics instead of the original observations.

PAIRED Statement

PAIRED *PairLists* **;**

The *PairLists* in the PAIRED statement identifies the variables to be compared in paired comparisons. You can use one or more *PairLists*. Variables or lists of variables are separated by an asterisk (*) or a colon (:). The asterisk requests comparisons between each variable on the left with each variable on the right. The colon requests comparisons between the first variable on the left and the first on the right, the second on the left and the second on the right, and so forth. The number of variables on the left must equal the number on the right when the colon is used. The differences are calculated by taking the variable on the left minus the variable on the right for both the asterisk and colon. A pair formed by a variable with itself is ignored. Use the PAIRED statement only for paired comparisons. The CLASS and VAR statements cannot be used with the PAIRED statement.

Examples of the use of the asterisk and the colon are shown in Table 92.3.

Table 92.3 PAIRED Statement in the TTEST Procedure

These PAIRED statements...	yield these comparisons
`PAIRED A*B;`	A-B
`PAIRED A*B C*D;`	A-B and C-D
`PAIRED (A B)*(C D);`	A-C, A-D, B-C, and B-D
`PAIRED (A B)*(C B);`	A-C, A-B, and B-C
`PAIRED (A1-A2)*(B1-B2);`	A1-B1, A1-B2, A2-B1, and A2-B2
`PAIRED (A1-A2):(B1-B2);`	A1-B1 and A2-B2

VAR Statement

VAR *variables* < / *options* > **;**

The VAR statement names the variables to be used in the analyses. One-sample comparisons are conducted when the VAR statement is used without the CROSSOVER option or CLASS statement. Two-independent-sample comparisons are conducted when the VAR statement is used with a CLASS statement.

An AB/BA crossover analysis is conducted when the CROSSOVER option is used in the VAR statement. In this case, you must specify an even number of variables. Each set of two variables represents the responses in the first and second periods of the AB/BA crossover design. For example, if you use the CROSSOVER option and specify **`VAR x1 x2 x3 x4`**, then you will get two analyses. One analysis will have x1 as the period 1 response and x2 as the period 2 response. The other analysis will have x3 as the period 1 response and x4 as the period 2 response.

The VAR statement cannot be used with the PAIRED statement. If the VAR statement is omitted, all numeric variables in the input data set (except a numeric variable appearing in the BY, CLASS, FREQ, or WEIGHT statement) are included in the analysis.

You can specify the following options after a slash (/):

CROSSOVER= (*variable1 variable2* **)**
specifies the variables representing the treatment applied in each of the two periods in an AB/BA crossover design. The treatment variables must have two, and only two, levels. For any given observation, the levels for the two variables must be different, due to the restrictions of the AB/BA crossover design. You can use either numeric or character variables.

Treatment levels are determined from the formatted values of the variables. Thus, you can use formats to define the treatment levels. See the discussions of the FORMAT procedure, the FORMAT statement, formats, and informats in *SAS Language Reference: Dictionary*.

IGNOREPERIOD
indicates that the period effect shall be ignored—that is, assumed to be equal to 0 (if TEST=DIFF) or 1 (if TEST=RATIO). This assumption increases the degrees of freedom for the test of the treatment difference by one and is usually more powerful, but it risks incorrect results if there is actually a period effect.

WEIGHT Statement

WEIGHT *variable* **;**

The WEIGHT statement weights each observation in the input data set by the value of the WEIGHT variable. The values of the WEIGHT variable can be nonintegral, and they are not truncated. Observations with negative, zero, or missing values for the WEIGHT variable are not used in the analyses. Each observation is assigned a weight of 1 when the WEIGHT statement is not used. The WEIGHT statement cannot be used with an input data set of summary statistics.

Details: TTEST Procedure

Input Data Set of Statistics

PROC TTEST accepts data containing either observation values or summary statistics. Observation values are supported for all analyses, whereas summary statistics are supported only for a subset of analyses. If the analysis involves the paired design, the AB/BA crossover design, or the lognormal distributional assumption (DIST=LOGNORMAL), then observation values must be used. The graphical results are unavailable if your input data set contains summary statistics rather than raw observed values.

PROC TTEST assumes that the DATA= data set contains statistics if it contains a character variable with name _TYPE_ or _STAT_. The TTEST procedure expects this character variable to contain the names of statistics. If both _TYPE_ and _STAT_ variables exist and are of type character, PROC TTEST expects _TYPE_ to contain the names of statistics including 'N', 'MEAN', and 'STD' for each BY group (or for each class within each BY group for two-sample *t* tests). If no 'N', 'MEAN', or 'STD' statistics exist, an error message is printed.

FREQ, WEIGHT, and PAIRED statements cannot be used with input data sets of statistics. BY, CLASS, and VAR statements are the same regardless of data set type. For paired comparisons, see the _DIF_ values for the _TYPE_=T observations in output produced by the OUTSTATS= option in the PROC COMPARE statement (see the *Base SAS Procedures Guide*).

Missing Values

An observation is omitted from the calculations if it has a missing value for either the CLASS variable, a CROSSOVER variable, a PAIRED variable, the variable to be tested (in a one-sample or two-independent-sample design), or either of the two response variables (in a crossover design). If more than one variable or pair of variables is listed in the VAR statement, a missing value in one variable or pair does not eliminate the observation from the analysis of other nonmissing variables or variable pairs.

Computational Methods

This section describes the computational formulas for the estimates, confidence limits, and tests for each analysis in the TTEST procedure. The first subsection defines some common notation. The second subsection discusses the distinction between arithmetic and geometric means. The third subsection explains the concept of the coefficient of variation. The following four subsections address the four supported designs (one-sample, paired, two-independent-sample, and AB/BA crossover). The content in each of those subsections is divided into separate discussions according to differ-

ent values of the DIST= and TEST= options in the PROC TTEST statement. The last subsection describes TOST equivalence analyses.

Common Notation

Table 92.4 displays notation for some of the commonly used symbols.

Table 92.4 Common Notation

Symbol	Description
μ	population value of (arithmetic) mean
μ_0	null value of test (value of H0= option in PROC TTEST statement)
σ^2	population variance
σ	population value of standard deviation
γ	population value of geometric mean
CV	population value of coefficient of variation (ratio of population standard deviation and population arithmetic mean)
α	value of ALPHA= option in PROC TTEST statement
$t_{p,\nu}$	pth percentile of t distribution with ν degrees of freedom (d.f.)
F_{p,ν_1,ν_2}	pth percentile of F distribution with ν_1 numerator d.f. and ν_2 denominator d.f.
$\chi^2_{p,\nu}$	pth percentile of chi-square distribution with ν d.f.

Arithmetic and Geometric Means

The *arithmetic mean* (more commonly called simply the *mean*) of the distribution of a random variable X is its expected value, $E(X)$. The arithmetic mean is the natural parameter of interest for a normal distribution because the distribution of the difference of normal random variables has a known normal distribution, and the arithmetic mean of a normal difference is equal to the difference of the individual arithmetic means. (No such convenient property holds for geometric means with normal data, with either differences or ratios.)

The usual estimate of an arithmetic mean is the sum of the values divided by the number of values:

$$\text{arithmetic mean} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

The *geometric mean* of the distribution of a random variable X is $\exp(E(\log(X))$, the exponentiation of the mean of the natural logarithm. The geometric mean is the natural parameter of interest for a lognormal distribution because the distribution of a ratio of lognormal random variables has a known lognormal distribution, and the geometric mean of a lognormal ratio is equal to the ratio of the individual geometric means. (No such convenient property holds for arithmetic means with lognormal data, with either differences or ratios.)

The usual estimate of a geometric mean is the product of the values raised to the power $1/n$, where n is the number of values:

$$\text{geometric mean} = \left(\prod_{i=1}^{n} y_i \right)^{\frac{1}{n}}$$

Coefficient of Variation

The *coefficient of variation* (abbreviated "CV") of the distribution of a random variable X is the ratio of the standard deviation to the (arithmetic) mean, or $\sqrt{\text{Var}(X)}/E(X)$. Conceptually, it is a measure of the variability of X expressed in units corresponding to the mean of X.

For lognormal data, the CV is the natural measure of variability (rather than the standard deviation) because the CV is invariant to multiplication of a lognormal variable by a constant. For a two-independent-sample design, the assumption of equal CVs on a lognormal scale is analogous to the assumption of equal variances on the normal scale. When the CVs of two independent samples of lognormal data are assumed equal, the pooled estimate of variability is used.

One-Sample Design

Define the following notation:

$$\begin{aligned}
n^\star &= \text{number of observations in data set} \\
y_i &= \text{value of } i\text{th observation}, \quad i \in \{1, \ldots, n^\star\} \\
f_i &= \text{frequency of } i\text{th observation}, \quad i \in \{1, \ldots, n^\star\} \\
w_i &= \text{weight of } i\text{th observation}, \quad i \in \{1, \ldots, n^\star\} \\
n &= \text{sample size} = \sum_{i}^{n^\star} f_i
\end{aligned}$$

Normal Data (DIST=NORMAL)

The mean estimate $\bar{y}$, standard deviation estimate s, and standard error SE are computed as follows:

$$\begin{aligned}
\bar{y} &= \frac{\sum_i^{n^\star} f_i w_i y_i}{\sum_i^{n^\star} f_i w_i} \\
s &= \left(\frac{\sum_i^{n^\star} f_i w_i (y_i - \bar{y})^2}{n-1} \right)^{\frac{1}{2}} \\
\text{SE} &= \frac{s}{\sum_i^{n^\star} f_i w_i}
\end{aligned}$$

The $100(1-\alpha)\%$ confidence interval for the mean μ is

$$\begin{aligned}
&\left(\bar{y} - t_{1-\frac{\alpha}{2},n-1}\text{SE} \;,\; \bar{y} + t_{1-\frac{\alpha}{2},n-1}\text{SE}\right), && \text{SIDES=2} \\
&\left(-\infty \;,\; \bar{y} + t_{1-\alpha,n-1}\text{SE}\right), && \text{SIDES=L} \\
&\left(\bar{y} - t_{1-\alpha,n-1}\text{SE} \;,\; \infty\right), && \text{SIDES=U}
\end{aligned}$$

The t value for the test is computed as

$$t = \frac{\bar{y} - \mu_0}{\text{SE}}$$

The p-value of the test is computed as

$$p\text{-value} = \begin{cases} P\left(t^2 > F_{1-\alpha,1,n-1}\right) & , \quad \text{2-sided} \\ P\left(t < t_{\alpha,n-1}\right) & , \quad \text{lower 1-sided} \\ P\left(t > t_{1-\alpha,n-1}\right) & , \quad \text{upper 1-sided} \end{cases}$$

The equal-tailed confidence interval for the standard deviation (CI=EQUAL) is based on the acceptance region of the test of $H_0: \sigma = \sigma_0$ that places an equal amount of area ($\frac{\alpha}{2}$) in each tail of the chi-square distribution:

$$\left\{ \chi^2_{\frac{\alpha}{2},n-1} \leq \frac{(n-1)s^2}{\sigma_0^2} \leq \chi^2_{\frac{1-\alpha}{2},n-1} \right\}$$

The acceptance region can be algebraically manipulated to give the following $100(1-\alpha)\%$ confidence interval for σ^2:

$$\left(\frac{(n-1)s^2}{\chi^2_{1-\frac{\alpha}{2},n-1}} \quad , \quad \frac{(n-1)s^2}{\chi^2_{\frac{\alpha}{2},n-1}} \right)$$

Taking the square root of each side yields the $100(1-\alpha)\%$ CI=EQUAL confidence interval for σ:

$$\left(\left(\frac{(n-1)s^2}{\chi^2_{1-\frac{\alpha}{2},n-1}} \right)^{\frac{1}{2}} \quad , \quad \left(\frac{(n-1)s^2}{\chi^2_{\frac{\alpha}{2},n-1}} \right)^{\frac{1}{2}} \right)$$

The other confidence interval for the standard deviation (CI=UMPU) is derived from the uniformly most powerful unbiased test of $H_0: \sigma = \sigma_0$ (Lehmann 1986). This test has acceptance region

$$\left\{ c_1 \leq \frac{(n-1)s^2}{\sigma_0^2} \leq c_2 \right\}$$

where the critical values c_1 and c_2 satisfy

$$\int_{c_1}^{c_2} f_{n-1}(y)dy = 1 - \alpha$$

and

$$\int_{c_1}^{c_2} y f_{n-1}(y)dy = (n-1)(1-\alpha)$$

where $f_\nu(y)$ is the p.d.f. of the chi-square distribution with ν degrees of freedom. This acceptance region can be algebraically manipulated to arrive at

$$P\left\{ \frac{(n-1)s^2}{c_2} \leq \sigma^2 \leq \frac{(n-1)s^2}{c_1} \right\} = 1 - \alpha$$

where c_1 and c_2 solve the preceding two integrals. To find the area in each tail of the chi-square distribution to which these two critical values correspond, solve $c_1 = \chi^2_{1-\alpha_2,n-1}$ and $c_2 = \chi^2_{\alpha_1,n-1}$ for α_1 and α_2; the resulting α_1 and α_2 sum to α. Hence, a $100(1-\alpha)\%$ confidence interval for σ^2 is given by

$$\left(\frac{(n-1)s^2}{\chi^2_{1-\alpha_2,n-1}} \ , \ \frac{(n-1)s^2}{\chi^2_{\alpha_1,n-1}} \right)$$

Taking the square root of each side yields the $100(1-\alpha)\%$ CI=UMPU confidence interval for σ:

$$\left(\left(\frac{(n-1)s^2}{\chi^2_{1-\alpha_2,n-1}} \right)^{\frac{1}{2}} \ , \ \left(\frac{(n-1)s^2}{\chi^2_{\alpha_1,n-1}} \right)^{\frac{1}{2}} \right)$$

Lognormal Data (DIST=LOGNORMAL)

The DIST=LOGNORMAL analysis is handled by log-transforming the data and null value, performing a DIST=NORMAL analysis, and then transforming the results back to the original scale. This simple technique is based on the properties of the lognormal distribution as discussed in Johnson, Kotz, and Balakrishnan (1994, Chapter 14).

Taking the natural logarithms of the observation values and the null value, define

$$z_i = \log(y_i) \ , \ \ i \in \{1, \ldots, n^\star\}$$
$$\gamma_0 = \log(\mu_0)$$

First a DIST=NORMAL analysis is performed on z_i in place of y_i. The geometric mean estimate $\hat{\gamma}$ and CV estimate $\widehat{CV}$ of the original lognormal data are computed as follows:

$$\hat{\gamma} = \exp(\bar{y})$$
$$\widehat{CV} = \left(\exp(s^2) - 1\right)^{\frac{1}{2}}$$

The t value and p-value remain the same. The confidence limits for the geometric mean and CV on the original lognormal scale are computed from the confidence limits for the arithmetic mean and standard deviation in the DIST=NORMAL analysis on the log-transformed data, in the same way that $\hat{\gamma}$ is derived from $\bar{y}$ and $\widehat{CV}$ is derived from s.

Paired Design

Define the following notation:

$$n^\star = \text{number of observations in data set}$$
$$y_{1i} = \text{value of } i\text{th observation for first PAIRED variable, } \ i \in \{1, \ldots, n^\star\}$$
$$y_{2i} = \text{value of } i\text{th observation for second PAIRED variable, } \ i \in \{1, \ldots, n^\star\}$$
$$f_i = \text{frequency of } i\text{th observation, } \ i \in \{1, \ldots, n^\star\}$$
$$w_i = \text{weight of } i\text{th observation, } \ i \in \{1, \ldots, n^\star\}$$
$$n = \text{sample size} = \sum_i^{n^\star} f_i$$

Normal Difference (DIST=NORMAL TEST=DIFF)

The analysis is the same as the analysis for the one-sample design in the section "Normal Data (DIST=NORMAL)" on page 7413 based on the differences

$$d_i = y_{1i} - y_{2i} \quad , \quad i \in \{1, \ldots, n^\star\}$$

Lognormal Ratio (DIST=LOGNORMAL TEST=RATIO)

The analysis is the same as the analysis for the one-sample design in the section "Lognormal Data (DIST=LOGNORMAL)" on page 7415 based on the ratios

$$r_i = y_{1i}/y_{2i} \quad , \quad i \in \{1, \ldots, n^\star\}$$

Normal Ratio (DIST=NORMAL TEST=RATIO)

The hypothesis $H_0: \mu_1/\mu_2 = \mu_0$, where μ_1 and μ_2 are the means of the first and second PAIRED variables, respectively, can be rewritten as $H_0: \mu_1 - \mu_0\mu_2 = 0$. The t value and p-value are computed in the same way as in the one-sample design in the section "Normal Data (DIST=NORMAL)" on page 7413 based on the transformed values

$$z_i = y_{1i} - \mu_0 y_{2i} \quad , \quad i \in \{1, \ldots, n^\star\}$$

Estimates and confidence limits are not computed for this situation.

Two-Independent-Sample Design

Define the following notation:

$n_1^\star$ = number of observations at first class level
$n_2^\star$ = number of observations at second class level
y_{1i} = value of ith observation at first class level, $i \in \{1, \ldots, n_1^\star\}$
y_{2i} = value of ith observation at second class level, $i \in \{1, \ldots, n_2^\star\}$
f_{1i} = frequency of ith observation at first class level, $i \in \{1, \ldots, n_1^\star\}$
f_{2i} = frequency of ith observation at second class level, $i \in \{1, \ldots, n_2^\star\}$
w_{1i} = weight of ith observation at first class level, $i \in \{1, \ldots, n_1^\star\}$
w_{2i} = weight of ith observation at second class level, $i \in \{1, \ldots, n_2^\star\}$

$$n_1 = \text{sample size for first class level} = \sum_i^{n_1^\star} f_{1i}$$

$$n_2 = \text{sample size for second class level} = \sum_i^{n_2^\star} f_{2i}$$

Normal Difference (DIST=NORMAL TEST=DIFF)

Observations at the first class level are assumed to be distributed as $\mathrm{N}(\mu_1, \sigma_1^2)$, and observations at the second class level are assumed to be distributed as $\mathrm{N}(\mu_2, \sigma_2^2)$, where μ_1, μ_2, σ_1, and σ_2 are unknown.

The within-class-level mean estimates ($\bar{y}_1$ and $\bar{y}_2$), standard deviation estimates (s_1 and s_2), standard errors (SE_1 and SE_2), and confidence limits for means and standard deviations are computed in the same way as for the one-sample design in the section "Normal Data (DIST=NORMAL)" on page 7413.

The mean difference $\mu_1 - \mu_2 = \mu_d$ is estimated by

$$\bar{y}_d = \bar{y}_1 - \bar{y}_2$$

Under the assumption of equal variances ($\sigma_1^2 = \sigma_2^2$), the pooled estimate of the common standard deviation is

$$s_p = \left(\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2} \right)^{\frac{1}{2}}$$

The pooled standard error (the estimated standard deviation of $\bar{y}_d$ assuming equal variances) is

$$\mathrm{SE}_p = s_p \left(\frac{1}{\sum_{i=1}^{n_1^\star} f_{1i} w_{1i}} + \frac{1}{\sum_{i=1}^{n_2^\star} f_{2i} w_{2i}} \right)^{\frac{1}{2}}$$

The pooled $100(1-\alpha)\%$ confidence interval for the mean difference μ_d is

$$\begin{aligned} \left(\bar{y}_d - t_{1-\frac{\alpha}{2}, n_1+n_2-2} \mathrm{SE}_p \ , \ \bar{y}_d + t_{1-\frac{\alpha}{2}, n_1+n_2-2} \mathrm{SE}_p \right), & \quad \text{SIDES=2} \\ \left(-\infty \ , \ \bar{y}_d + t_{1-\alpha, n_1+n_2-2} \mathrm{SE}_p \right), & \quad \text{SIDES=L} \\ \left(\bar{y}_d - t_{1-\alpha, n_1+n_2-2} \mathrm{SE}_p \ , \ \infty \right), & \quad \text{SIDES=U} \end{aligned}$$

The t value for the pooled test is computed as

$$t_p = \frac{\bar{y}_d - \mu_0}{\mathrm{SE}_p}$$

The p-value of the test is computed as

$$p\text{-value} = \begin{cases} P\left(t_p^2 > F_{1-\alpha,1,n_1+n_2-2}\right) \ , & \text{2-sided} \\ P\left(t_p < t_{\alpha, n_1+n_2-2}\right) \ , & \text{lower 1-sided} \\ P\left(t_p > t_{1-\alpha, n_1+n_2-2}\right) \ , & \text{upper 1-sided} \end{cases}$$

Under the assumption of unequal variances (the Behrens-Fisher problem), the unpooled standard error is computed as

$$\mathrm{SE}_u = \left(\frac{s_1^2}{\sum_{i=1}^{n_1^\star} f_{1i} w_{1i}} + \frac{s_2^2}{\sum_{i=1}^{n_2^\star} f_{2i} w_{2i}} \right)^{\frac{1}{2}}$$

Satterthwaite's (1946) approximation for the degrees of freedom, extended to accommodate weights, is computed as

$$\mathrm{df}_u = \frac{\mathrm{SE}_u^4}{\frac{s_1^4}{(n_1-1)\left(\sum_{i=1}^{n_1^\star} f_{1i} w_{1i}\right)^2} + \frac{s_2^4}{(n_2-1)\left(\sum_{i=1}^{n_2^\star} f_{2i} w_{2i}\right)^2}}$$

The unpooled Satterthwaite $100(1-\alpha)\%$ confidence interval for the mean difference μ_d is

$$\begin{aligned}\left(\bar{y}_d - t_{1-\frac{\alpha}{2},\mathrm{df}_u}\mathrm{SE}_u \;,\; \bar{y}_d + t_{1-\frac{\alpha}{2},\mathrm{df}_u}\mathrm{SE}_u\right), &\quad \text{SIDES=2}\\ \left(-\infty \;,\; \bar{y}_d + t_{1-\alpha,\mathrm{df}_u}\mathrm{SE}_u\right), &\quad \text{SIDES=L}\\ \left(\bar{y}_d - t_{1-\alpha,\mathrm{df}_u}\mathrm{SE}_u \;,\; \infty\right), &\quad \text{SIDES=U}\end{aligned}$$

The t value for the unpooled Satterthwaite test is computed as

$$t_u = \frac{\bar{y}_d - \mu_0}{\mathrm{SE}_u}$$

The p-value of the unpooled Satterthwaite test is computed as

$$p\text{-value} = \begin{cases} P\left(t_u^2 > F_{1-\alpha,1,\mathrm{df}_u}\right) \;, & \text{2-sided}\\ P\left(t_u < t_{\alpha,\mathrm{df}_u}\right) \;, & \text{lower 1-sided}\\ P\left(t_u > t_{1-\alpha,\mathrm{df}_u}\right) \;, & \text{upper 1-sided}\end{cases}$$

When the COCHRAN option is specified in the PROC TTEST statement, the Cochran and Cox (1950) approximation of the p-value of the t_u statistic is the value of p such that

$$t_u = \frac{\left(\frac{s_1^2}{\sum_{i=1}^{n_1^\star} f_{1i}w_{1i}}\right)t_1 + \left(\frac{s_2^2}{\sum_{i=1}^{n_2^\star} f_{2i}w_{2i}}\right)t_2}{\left(\frac{s_1^2}{\sum_{i=1}^{n_1^\star} f_{1i}w_{1i}}\right) + \left(\frac{s_2^2}{\sum_{i=1}^{n_2^\star} f_{2i}w_{2i}}\right)}$$

where t_1 and t_2 are the critical values of the t distribution corresponding to a significance level of p and sample sizes of n_1 and n_2, respectively. The number of degrees of freedom is undefined when $n_1 \neq n_2$. In general, the Cochran and Cox test tends to be conservative (Lee and Gurland 1975).

The $100(1-\alpha)\%$ CI=EQUAL and CI=UMPU confidence intervals for the common population standard deviation σ assuming equal variances are computed as discussed in the section "Normal Data (DIST=NORMAL)" on page 7413 for the one-sample design, except replacing s^2 by s_p^2 and $(n-1)$ by (n_1+n_2-1).

The folded form of the F statistic, F', tests the hypothesis that the variances are equal (Steel and Torrie 1980), where

$$F' = \frac{\max(s_1^2, s_2^2)}{\min(s_1^2, s_2^2)}$$

A test of F' is a two-tailed F test because you do not specify which variance you expect to be larger. The p-value gives the probability of a greater F value under the null hypothesis that $\sigma_1^2 = \sigma_2^2$. Note that this test is not very robust to violations of the assumption that the data are normally distributed, and thus it is not recommended without confidence in the normality assumption.

Lognormal Ratio (DIST=LOGNORMAL TEST=RATIO)

The DIST=LOGNORMAL analysis is handled by log-transforming the data and null value, performing a DIST=NORMAL analysis, and then transforming the results back to the original scale.

See the section "Normal Data (DIST=NORMAL)" on page 7413 for the one-sample design for details on how the DIST=NORMAL computations for means and standard deviations are transformed into the DIST=LOGNORMAL results for geometric means and CVs. As mentioned in the section "Coefficient of Variation" on page 7413, the assumption of equal CVs on the lognormal scale is analogous to the assumption of equal variances on the normal scale.

Normal Ratio (DIST=NORMAL TEST=RATIO)

The distributional assumptions, equality of variances test, and within-class-level mean estimates ($\bar{y}_1$ and $\bar{y}_2$), standard deviation estimates (s_1 and s_2), standard errors (SE_1 and SE_2), and confidence limits for means and standard deviations are the same as in the section "Normal Difference (DIST=NORMAL TEST=DIFF)" on page 7416 for the two-independent-sample design.

The mean ratio $\mu_1/\mu_2 = \mu_r$ is estimated by

$$\hat{\mu}_r = \bar{y}_1/\bar{y}_2$$

No estimates or confidence intervals for the ratio of standard deviations are computed.

Under the assumption of equal variances ($\sigma_1^2 = \sigma_2^2$), the pooled confidence interval for the mean ratio is the Fieller (1954) confidence interval, extended to accommodate weights. Let

$$a_p = \frac{s_p^2 t^2_{1-\frac{\alpha}{2},n_1+n_2-2}}{\sum_{i=1}^{n_2^\star} f_{2i} w_{2i}} - \bar{y}_2^2$$
$$b_p = \bar{y}_1 \bar{y}_2$$
$$c_p = \frac{s_p^2 t^2_{1-\frac{\alpha}{2},n_1+n_2-2}}{\sum_{i=1}^{n_1^\star} f_{1i} w_{1i}} - \bar{y}_1^2$$

where s_p is the pooled standard deviation defined in the section "Normal Difference (DIST=NORMAL TEST=DIFF)" on page 7416 for the two-independent-sample design. If $a_p \geq 0$ (which occurs when $\bar{y}_2$ is too close to zero), then the pooled two-sided $100(1-\alpha)\%$ Fieller confidence interval for μ_r does not exist. If $a < 0$, then the interval is

$$\left(-\frac{b_p}{a_p} + \frac{\left(b_p^2 - a_p c_p\right)^{\frac{1}{2}}}{a_p} \; , -\frac{b_p}{a_p} - \frac{\left(b_p^2 - a_p c_p\right)^{\frac{1}{2}}}{a_p} \right)$$

For the one-sided intervals, let

$$a_p^\star = \frac{s_p^2 t^2_{1-\alpha,n_1+n_2-2}}{\sum_{i=1}^{n_2^\star} f_{2i} w_{2i}} - \bar{y}_2^2$$
$$c_p^\star = \frac{s_p^2 t^2_{1-\alpha,n_1+n_2-2}}{\sum_{i=1}^{n_1^\star} f_{1i} w_{1i}} - \bar{y}_1^2$$

which differ from a_p and c_p only in the use of α in place of $\alpha/2$. If $a_p^\star \geq 0$, then the pooled one-sided $100(1-\alpha)\%$ Fieller confidence intervals for μ_r do not exist. If $a_p^\star < 0$, then the intervals are

$$\left(-\infty \ , -\frac{b_p}{a_p^\star} - \frac{\left(b_p^2 - a_p^\star c_p^\star\right)^{\frac{1}{2}}}{a_p^\star} \right), \quad \text{SIDES=L}$$

$$\left(-\frac{b_p}{a_p^\star} + \frac{\left(b_p^2 - a_p^\star c_p^\star\right)^{\frac{1}{2}}}{a_p^\star} \ , \ \infty \right), \quad \text{SIDES=U}$$

The pooled t test assuming equal variances is the Sasabuchi (1988a, 1988b) test. The hypothesis $H_0\colon \mu_r = \mu_0$ is rewritten as $H_0\colon \mu_1 - \mu_0\mu_2 = 0$, and the pooled t test in the section "Normal Difference (DIST=NORMAL TEST=DIFF)" on page 7416 for the two-independent-sample design is conducted on the original y_{1i} values ($i \in \{1, \ldots, n_1^\star\}$) and transformed values of y_{2i}

$$y_{2i}^\star = \mu_0 y_{2i} \ , \ \ i \in \{1, \ldots, n_2^\star\}$$

with a null difference of 0. The t value for the Sasabuchi pooled test is computed as

$$t_p = \frac{\bar{y}_1 - \mu_0 \bar{y}_2}{s_p \left(\frac{1}{\sum_{i=1}^{n_1^\star} f_{1i} w_{1i}} + \frac{\mu_0^2}{\sum_{i=1}^{n_2^\star} f_{2i} w_{2i}} \right)^{\frac{1}{2}}}$$

The p-value of the test is computed as

$$p\text{-value} = \begin{cases} P\left(t_p^2 > F_{1-\alpha,1,n_1+n_2-2}\right) & , \quad \text{2-sided} \\ P\left(t_p < t_{\alpha,n_1+n_2-2}\right) & , \quad \text{lower 1-sided} \\ P\left(t_p > t_{1-\alpha,n_1+n_2-2}\right) & , \quad \text{upper 1-sided} \end{cases}$$

Under the assumption of unequal variances, the unpooled Satterthwaite-based confidence interval for the mean ratio μ_r is computed according to the method in Dilba, Schaarschmidt, and Hothorn (2006), extended to accommodate weights. The degrees of freedom are computed as

$$\mathrm{df}_u = \frac{\left(\frac{s_1^2}{\sum_{i=1}^{n_1^\star} f_{1i} w_{1i}} + \frac{\hat{\mu}_r^2 s_2^2}{\sum_{i=1}^{n_2^\star} f_{2i} w_{2i}} \right)^2}{\frac{s_1^4}{(n_1-1)\left(\sum_{i=1}^{n_1^\star} f_{1i} w_{1i}\right)^2} + \frac{\hat{\mu}_r^4 s_2^4}{(n_2-1)\left(\sum_{i=1}^{n_2^\star} f_{2i} w_{2i}\right)^2}}$$

Note that the estimate $\hat{\mu}_r = \bar{y}_1/\bar{y}_2$ is used in df_u. Let

$$a_u = \frac{s_2^2 t_{1-\frac{\alpha}{2},\mathrm{df}_u}^2}{\sum_{i=1}^{n_2^\star} f_{2i} w_{2i}} - \bar{y}_2^2$$

$$b_u = \bar{y}_1 \bar{y}_2$$

$$c_u = \frac{s_1^2 t_{1-\frac{\alpha}{2},\mathrm{df}_u}^2}{\sum_{i=1}^{n_1^\star} f_{1i} w_{1i}} - \bar{y}_1^2$$

where s_1 and s_2 are the within-class-level standard deviations defined in the section "Normal Difference (DIST=NORMAL TEST=DIFF)" on page 7416 for the two-independent-sample design. If $a_u \geq 0$ (which occurs when $\bar{y}_2$ is too close to zero), then the unpooled Satterthwaite-based two-sided $100(1-\alpha)\%$ confidence interval for μ_r does not exist. If $a_u < 0$, then the interval is

$$\left(-\frac{b_u}{a_u} + \frac{\left(b_u^2 - a_u c_u\right)^{\frac{1}{2}}}{a_u} \;\; , -\frac{b_u}{a_u} - \frac{\left(b_u^2 - a_u c_u\right)^{\frac{1}{2}}}{a_u} \right)$$

The t test assuming unequal variances is the test derived in Tamhane and Logan (2004). The hypothesis $H_0\colon \mu_r = \mu_0$ is rewritten as $H_0\colon \mu_1 - \mu_0\mu_2 = 0$, and the Satterthwaite t test in the section "Normal Difference (DIST=NORMAL TEST=DIFF)" on page 7416 for the two-independent-sample design is conducted on the original y_{1i} values ($i \in \{1, \ldots, n_1^\star\}$) and transformed values of y_{2i}

$$y_{2i}^\star = \mu_0 y_{2i} \quad , \quad i \in \{1, \ldots, n_2^\star\}$$

with a null difference of 0. The degrees of freedom used in the unpooled t test differs from the df_u used in the unpooled confidence interval. The mean ratio μ_0 under the null hypothesis is used in place of the estimate $\hat{\mu}_r$:

$$\mathrm{df}_u^\star = \frac{\left(\frac{s_1^2}{\sum_{i=1}^{n_1^\star} f_{1i} w_{1i}} + \frac{\mu_0^2 s_2^2}{\sum_{i=1}^{n_2^\star} f_{2i} w_{2i}} \right)^2}{\frac{s_1^4}{(n_1-1)\left(\sum_{i=1}^{n_1^\star} f_{1i} w_{1i}\right)^2} + \frac{\mu_0^4 s_2^4}{(n_2-1)\left(\sum_{i=1}^{n_2^\star} f_{2i} w_{2i}\right)^2}}$$

The t value for the Satterthwaite-based unpooled test is computed as

$$t_u = \frac{\bar{y}_1 - \mu_0 \bar{y}_2}{\left(\frac{s_1^2}{\sum_{i=1}^{n_1^\star} f_{1i} w_{1i}} + \frac{\mu_0^2 s_2^2}{\sum_{i=1}^{n_2^\star} f_{2i} w_{2i}} \right)^{\frac{1}{2}}}$$

The p-value of the test is computed as

$$p\text{-value} = \begin{cases} P\left(t_u^2 > F_{1-\alpha,1,\mathrm{df}_u^\star}\right) & , \quad \text{2-sided} \\ P\left(t_u < t_{\alpha,\mathrm{df}_u^\star}\right) & , \quad \text{lower 1-sided} \\ P\left(t_u > t_{1-\alpha,\mathrm{df}_u^\star}\right) & , \quad \text{upper 1-sided} \end{cases}$$

AB/BA Crossover Design

Let "A" and "B" denote the two treatment values. Define the following notation:

$n_1^\star$ = number of observations with treatment sequence AB
$n_2^\star$ = number of observations with treatment sequence BA
y_{11i} = response value of ith observation in sequence AB during period 1, $i \in \{1, \ldots, n_1^\star\}$
y_{12i} = response value of ith observation in sequence AB during period 2, $i \in \{1, \ldots, n_1^\star\}$
y_{21i} = response value of ith observation in sequence BA during period 1, $i \in \{1, \ldots, n_2^\star\}$
y_{22i} = response value of ith observation in sequence BA during period 2, $i \in \{1, \ldots, n_2^\star\}$

So $\{y_{11i}, \ldots, y_{11n_1^\star}\}$ and $\{y_{22i}, \ldots, y_{22n_2^\star}\}$ are all observed at treatment level A, and $\{y_{12i}, \ldots, y_{12n_2^\star}\}$ and $\{y_{21i}, \ldots, y_{21n_1^\star}\}$ are all observed at treatment level B.

Define the *period difference* for an observation as the difference between period 1 and period 2 response values:

$$\mathrm{pd}_{kji} = y_{k1i} - y_{k2i}$$

for $k \in \{1, 2\}$ and $i \in \{1, \ldots, n_k^\star\}$. Similarly, the *period ratio* is the ratio between period 1 and period 2 response values:

$$\mathrm{pr}_{kji} = y_{k1i} / y_{k2i}$$

The *crossover difference* for an observation is the difference between treatment A and treatment B response values:

$$\mathrm{cd}_{kji} = \begin{cases} y_{k1i} - y_{k2i} & , \quad k = 1 \\ y_{k2i} - y_{k1i} & , \quad k = 2 \end{cases}$$

Similarly, the *crossover ratio* is the ratio between treatment A and treatment B response values:

$$\mathrm{cr}_{kji} = \begin{cases} y_{k1i} / y_{k2i} & , \quad k = 1 \\ y_{k2i} / y_{k1i} & , \quad k = 2 \end{cases}$$

In the absence of the IGNOREPERIOD option in the PROC TTEST statement, the data are split into two groups according to treatment sequence and analyzed as a two-independent-sample design. If DIST=NORMAL, then the analysis of the treatment effect is based on the half period differences $\{\mathrm{pd}_{kji}/2\}$, and the analysis for the period effect is based on the half crossover differences $\{\mathrm{cd}_{kji}/2\}$. The computations for the normal difference analysis are the same as in the section "Normal Difference (DIST=NORMAL TEST=DIFF)" on page 7416 for the two-independent-sample design. The normal ratio analysis without the IGNOREPERIOD option is not supported for the AB/BA crossover design. If DIST=LOGNORMAL, then the analysis of the treatment effect is based on the square root of the period ratios $\{\sqrt{\mathrm{pr}_{kji}}\}$, and the analysis for the period effect is based on the square root of the crossover ratios $\{\sqrt{\mathrm{cr}_{kji}}\}$. The computations are the same as in the section "Lognormal Ratio (DIST=LOGNORMAL TEST=RATIO)" on page 7418 for the two-independent-sample design.

If the IGNOREPERIOD option is specified, then the treatment effect is analyzed as a paired analysis on the (treatment A, treatment B) response value pairs, regardless of treatment sequence. So the set of pairs is taken to be the concatenation of $\{(y_{111}, y_{121}), \ldots, (y_{11n_1^\star}, y_{12n_1^\star})\}$ and $\{(y_{221}, y_{211}), \ldots, (y_{22n_2^\star}, y_{22n_2^\star})\}$. The computations are the same as in the section "Paired Design" on page 7415.

See Senn (2002, Chapter 3) for a more detailed discussion of the AB/BA crossover design.

TOST Equivalence Test

The hypotheses for an equivalence test are

$$H_0{:}\mu < \theta_L \quad \text{or} \quad \mu > \theta_U$$
$$H_1{:}\theta_L \le \mu \le \theta_U$$

where θ_L and θ_U are the lower and upper bounds specified in the TOST option in the PROC TTEST statement, and μ is the analysis criterion (mean, mean ratio, or mean difference, depending on the analysis). Following the two one-sided tests (TOST) procedure of Schuirmann (1987), the equivalence test is conducted by performing two separate tests:

$$H_{a0}{:}\mu < \theta_L$$
$$H_{a1}{:}\mu \geq \theta_L$$

and

$$H_{b0}{:}\mu > \theta_U$$
$$H_{b1}{:}\mu \leq \theta_U$$

The overall p-value is the larger of the two p-values of those tests.

Rejection of H_0 in favor of H_1 at significance level α occurs if and only if the $100(1-2\alpha)\%$ confidence interval for μ is contained completely within (θ_L, θ_U). So, the $100(1-2\alpha)\%$ confidence interval for μ is displayed in addition to the usual $100(1-\alpha)\%$ interval.

See Phillips (1990), Diletti, Hauschke, and Steinijans (1991), and Hauschke et al. (1999) for further discussion of equivalence testing for the designs supported in the TTEST procedure.

Displayed Output

For an AB/BA crossover design, the "CrossoverVarInfo" table shows the variables specified for the response and treatment values in each period of the design.

The summary statistics in the "Statistics" table and confidence limits in the "ConfLimits" table are displayed for certain variables and/or transformations or subgroups of the variables in the analysis, depending on the design. For a one-sample design, summary statistics are displayed for all variables in the analysis. For a paired design, statistics are displayed for the difference if you specify the TEST=DIFF option in the PROC TTEST statement, or for the ratio if you specify TEST=RATIO. For a two-independent-sample design, the statistics for each of the two groups and for the difference (if TEST=DIFF) or ratio (if TEST=RATIO) are displayed. For an AB/BA crossover design, statistics are displayed for each of the four cells in the design (all four combinations of the two periods and two treatments). If the IGNOREPERIOD option is absent, then if TEST=DIFF is specified, statistics are displayed for the treatment difference within each sequence and overall, and also for the period difference. If TEST=RATIO, statistics are displayed for the treatment ratio within each sequence and overall, and also for the period ratio. If the IGNOREPERIOD option is specified in the VAR statement, then statistics are displayed for the overall treatment difference if TEST=DIFF or for the overall treatment ratio if TEST=RATIO.

The "Statistics" table displays the following summary statistics:

- the name of the variable(s), displayed if the NOBYVAR option is used in the PROC TTEST statement
- the name of the classification variable (if two-independent-sample design) or treatment and period (if AB/BA crossover design)

- N, the number of nonmissing values
- the (arithmetic) Mean, displayed if the DIST=NORMAL option is specified in the PROC TTEST statement
- the Geometric Mean, displayed if the DIST=LOGNORMAL option is specified in the PROC TTEST statement
- Std Dev, the standard deviation, displayed if the DIST=NORMAL option is specified in the PROC TTEST statement
- the Coefficient of Variation, displayed if the DIST=LOGNORMAL option is specified in the PROC TTEST statement
- Std Err, the standard error of the mean, displayed if the DIST=NORMAL option is specified in the PROC TTEST statement
- the Minimum value
- the Maximum value

The "ConfLimits" table displays the following:

- the name of the variable(s), displayed if the NOBYVAR option is used in the PROC TTEST statement
- the name of the classification variable (if two-independent-sample design) or treatment and period (if AB/BA crossover design)
- the (arithmetic) Mean, displayed if the DIST=NORMAL option is specified in the PROC TTEST statement
- the Geometric Mean, displayed if the DIST=LOGNORMAL option is specified in the PROC TTEST statement
- $100(1-\alpha)\%$ CL Mean, the lower and upper confidence limits for the mean. Separate pooled and Satterthwaite confidence limits are shown for the difference or ratio transformations in two-independent-sample designs and AB/BA crossover designs without the IGNOREPERIOD option.
- Std Dev, the standard deviation, displayed if the DIST=NORMAL option is specified in the PROC TTEST statement
- the Coefficient of Variation, displayed if the DIST=LOGNORMAL option is specified in the PROC TTEST statement
- $100(1-\alpha)\%$ CL Std Dev, the equal-tailed confidence limits for the standard deviation, displayed if the DIST=NORMAL and CI=EQUAL options are specified in the PROC TTEST statement
- $100(1-\alpha)\%$ UMPU CL Std Dev, the UMPU confidence limits for the standard deviation, displayed if the DIST=NORMAL and CI=UMPU options are specified in the PROC TTEST statement

- $100(1-\alpha)\%$ CL CV, the equal-tailed confidence limits for the coefficient of variation, displayed if the DIST=LOGNORMAL and CI=EQUAL options are specified in the PROC TTEST statement
- $100(1-\alpha)\%$ UMPU CL CV, the UMPU confidence limits for the coefficient of variation, displayed if the DIST=LOGNORMAL and CI=UMPU options are specified in the PROC TTEST statement

The confidence limits in the "EquivLimits" table and test results in the "TTests" and "EquivTests" tables are displayed only for the test criteria—that is, the variables or transformations being tested. For a one-sample design, results are displayed for all variables in the analysis. For a paired design, results are displayed for the difference if you specify the TEST=DIFF option in the PROC TTEST statement, or for the ratio if you specify TEST=RATIO. For a two-independent-sample design, the results for the difference (if TEST=DIFF) or ratio (if TEST=RATIO) are displayed. For an AB/BA crossover design, results are displayed for the treatment difference (if TEST=DIFF) or ratio (if TEST=RATIO). If the IGNOREPERIOD option is absent, then results are also displayed for the period difference (if TEST=DIFF) or ratio (if TEST=RATIO).

The "EquivLimits" table, produced only if the TOST option is specified in the PROC TTEST statement, displays the following:

- the name of the variable(s), displayed if the NOBYVAR option is used in the PROC TTEST statement
- the (arithmetic) Mean, displayed if the DIST=NORMAL option is specified in the PROC TTEST statement
- the Geometric Mean, displayed if the DIST=LOGNORMAL option is specified in the PROC TTEST statement
- Lower Bound, the lower equivalence bound for the mean specified in the TOST option in the PROC TTEST statement
- $100(1-2\alpha)\%$ CL Mean, the lower and upper confidence limits for the mean relevant to the equivalence test. Separate pooled and Satterthwaite confidence limits are shown for two-independent-sample designs and AB/BA crossover designs without the IGNOREPERIOD option.
- Upper Bound, the upper equivalence bound for the mean specified in the TOST option in the PROC TTEST statement
- Assessment, the result of the equivalence test at the significance level specified by the ALPHA= option in the PROC TTEST statement, either "Equivalent" or "Not equivalent"

The "TTests" table is produced only if the TOST option is *not* specified in the PROC TTEST statement. Separate results for pooled and Satterthwaite tests (and also the Cochran and Cox test, if the COCHRAN option is specified in the PROC TTEST statement) are displayed for two-independent-sample designs and AB/BA crossover designs without the IGNOREPERIOD option. The table includes the following results:

- the name of the variable(s), displayed if the NOBYVAR option is used in the PROC TTEST statement
- t Value, the t statistic for comparing the mean to the null value as specified by the H0= option in the PROC TTEST statement
- DF, the degrees of freedom
- the p-value, the probability of obtaining a t statistic at least as extreme as the observed t value under the null hypothesis

The "EquivTests" table is produced only if the TOST option is specified in the PROC TTEST statement. Separate results for pooled and Satterthwaite tests are displayed for two-independent-sample designs and AB/BA crossover designs without the IGNOREPERIOD option. Each test consists of two separate one-sided tests. The overall p-value is the larger p-value from these two tests. The table includes the following results:

- the name of the variable(s), displayed if the NOBYVAR option is used in the PROC TTEST statement
- Null, the lower equivalence bound for the Upper test or the upper equivalence bound for the Lower test, as specified by the TOST option in the PROC TTEST statement
- t Value, the t statistic for comparing the mean to the Null value
- DF, the degrees of freedom
- the p-value, the probability of obtaining a t statistic at least as extreme as the observed t value under the null hypothesis

The "Equality" table gives the results of the test of equality of variances. It is displayed for two-independent-sample designs and AB/BA crossover designs without the IGNOREPERIOD option. The table includes the following results:

- the name of the variable(s), displayed if the NOBYVAR option is used in the PROC TTEST statement
- Num DF and Den DF, the numerator and denominator degrees of freedom
- F Value, the F' (folded) statistic
- Pr > F, the probability of a greater F' value. This is the two-tailed p-value.

ODS Table Names

PROC TTEST assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 92.5. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 92.5 ODS Tables Produced by PROC TTEST

ODS Table Name	Description	Syntax
ConfLimits	$100(1-\alpha)\%$ confidence limits for means, standard deviations, and/or coefficients of variation	by default
Equality	Tests for equality of variance	CLASS statement or VAR / CROSSOVER
EquivLimits	$100(1-2\alpha)\%$ confidence limits for means	PROC TTEST TOST
EquivTests	Equivalence t tests	PROC TTEST TOST
Statistics	Univariate summary statistics	by default
TTests	t tests	by default

ODS Graphics

ODS Graph Names

To request graphics with PROC TTEST, you must first enable ODS Graphics by specifying the **ODS GRAPHICS ON** statement. See Chapter 21, "Statistical Graphics Using ODS," for more information. You can reference every graph produced through ODS Graphics with a name. The names of the graphs that PROC TTEST generates are listed in Table 92.6, along with the required statements and options.

Table 92.6 ODS Graphics Produced by PROC TTEST

ODS Graph Name	Plot Description	Option
AgreementOfPeriods	Plot of period 2 against period 1 response values for an AB/BA crossover design	VAR / CROSSOVER PLOTS=AGREEMENT(TYPE=PERIOD)
AgreementOfTreatments	Plot of second treatment against first treatment response values for an AB/BA crossover design	VAR / CROSSOVER PLOTS=AGREEMENT
AgreementPlot	Plot of second response against first response for a paired design	PAIRED statement PLOTS=AGREEMENT
BoxPlot	Box plots, also with confidence band for one-sample or paired design	PLOTS=BOX PLOTS=SUMMARY(UNPACK)
Histogram	Histograms with overlaid kernel densities, and also normal densities if DIST=NORMAL	PLOTS=HISTOGRAM PLOTS=SUMMARY(UNPACK)
Interval	Confidence intervals for means	PLOTS=INTERVAL

Table 92.6 *continued*

ODS Graph Name	Plot Description	Option
ProfilesOverPrd	Plot of response profiles over periods 1 and 2 for an AB/BA crossover design	VAR / CROSSOVER PLOTS=PROFILES(TYPE=PERIOD)
ProfilesOverTrt	Plot of response profiles over first and second treatments for an AB/BA crossover design	VAR / CROSSOVER PLOTS=PROFILES
ProfilesPlot	Plot of response profiles over first and second response values for a paired design	PAIRED statement PLOTS=PROFILES
QQPlot	Normal quantile-quantile plots	PLOTS=QQ
SummaryPanel	Histograms with overlaid kernel densities (and also normal densities if DIST=NORMAL) and box plots (and also with confidence band for one-sample or paired design)	PLOTS=SUMMARY

Interpreting Graphs

Agreement Plots for Paired Designs

For paired designs, the second response of each pair is plotted against the first response, with the mean shown as a large bold symbol. If the WEIGHT statement is used, then the mean is the weighted mean. A diagonal line with slope=0 and y-intercept=1 is overlaid. The location of the points with respect to the diagonal line reveals the strength and direction of the difference or ratio. The tighter the clustering along the same direction as the line, the stronger the positive correlation of the two measurements for each subject. Clustering along a direction perpendicular to the line indicates negative correlation.

Period Agreement Plots for Crossover Designs

The response in the second period is plotted against the response in the first period, with plot symbols distinguishing the two treatment sequences and the two sequence means shown larger in bold. If the WEIGHT statement is used, then the means are weighted means. A diagonal line with slope=0 and y-intercept=1 is overlaid.

In the absence of a strong period effect, the points from each sequence will appear as mirror images about the diagonal line, farther apart with stronger treatment effects. Deviations from symmetry about the diagonal line indicate a period effect. The spread of points within each treatment sequence is an indicator of between-subject variability. The tighter the clustering along the same direction as the line (within each treatment sequence), the stronger the positive correlation of the

two measurements for each subject. Clustering along a direction perpendicular to the line indicates negative correlation.

The period agreement plot is usually less informative than the treatment agreement plot. The exception is when the period effect is stronger than the treatment effect.

Treatment Agreement Plots for Crossover Designs

The response associated with the second treatment is plotted against the response associated with the first treatment, with plot symbols distinguishing the two treatment sequences and the two sequence means shown larger in bold. If the WEIGHT statement is used, then the means are weighted means. A diagonal line with slope=0 and y-intercept=1 is overlaid.

The location of the points with respect to the diagonal line reveals the strength and direction of the treatment effect. Substantial location differences between the two sequences indicates a strong period effect. The spread of points within each treatment sequence is an indicator of between-subject variability. The tighter the clustering along the same direction as the line (within each treatment sequence), the stronger the positive correlation of the two measurements for each subject. Clustering along a direction perpendicular to the line indicates negative correlation.

Box Plots

The box is drawn from the 25th percentile (lower quartile) to the 75th percentile (upper quartile). The vertical line inside the box shows the location of the median. If DIST=NORMAL, then a diamond symbol shows the location of the mean. The whiskers extend to the minimum and maximum observations, and circles beyond the whiskers identify outliers.

For one-sample and paired designs, a confidence interval for the mean is shown as a band in the background. If the analysis is an equivalence analysis (with the TOST option in the PROC TTEST statement), then the interval is a $100(1-2\alpha)\%$ confidence interval shown along with the equivalence bounds. The inclusion of this interval completely within the bounds is indicative of a significant p-value. If the analysis is not an equivalence analysis, then the confidence level is $100(1-\alpha)\%$. If the SHOWH0 global plot option is used, then the null value for the test is shown. If the WEIGHT statement is used, then weights are incorporated in the confidence intervals.

Histograms

The WEIGHT statement is ignored in the computation of the normal and kernel densities.

Confidence Intervals

If the analysis is an equivalence analysis (with the TOST option in the PROC TTEST statement), then unless the TYPE=PERGROUP option is used, the interval is a $100(1-2\alpha)\%$ mean confidence interval shown along with the equivalence bounds. The inclusion of this interval completely within the bounds is indicative of a significant p-value.

If the analysis is not an equivalence analysis, or if the TYPE=PERGROUP option is used, then the confidence level is $100(1-\alpha)\%$. If the SHOWH0 global plot option is used, then the null value for the test is shown.

If the SIDES=L or SIDES=U option is used in the PROC TTEST statement, then the unbounded side of the one-sided interval is represented with an arrowhead. Note that the actual location of the arrowhead is irrelevant.

If the WEIGHT statement is used, then weights are incorporated in the confidence intervals.

Profiles for Paired Designs

For paired designs, a line is drawn for each observation from left to right connecting the first response to the second response. The mean first response and mean second response are connected with a bold line. If the WEIGHT statement is used, then the means are weighted means. The more extreme the slope, the stronger the effect. A wide spread of profiles indicates high between-subject variability. Consistent positive slopes indicate strong positive correlation. Widely varying slopes indicate lack of correlation, while consistent negative slopes indicate strong negative correlation.

Profiles over Period for Crossover Designs

For each observation, the response for the first period is connected to the response for second period, regardless of the treatment applied in each period. The means for each treatment sequence are shown in bold. If the WEIGHT statement is used, then the means are weighted means.

In the absence of a strong period effect, the profiles for each sequence will appear as mirror images about an imaginary horizontal line in the center. Deviations from symmetry about this imaginary horizontal line indicate a period effect. A wide spread of profiles within sequence indicates high between-subject variability.

The TYPE=PERIOD plot is usually less informative than the TYPE=TREATMENT plot. The exception is when the period effect is stronger than the treatment effect.

Profiles over Treatment for Crossover Designs

For each observation, the response for the first treatment is connected to the response for the second treatment, regardless of the periods in which they occur. The means for each treatment sequence are shown in bold. If the WEIGHT statement is used, then the means are weighted means.

In general, the more extreme the slope, the stronger the treatment effect. Slope differences between the two treatment sequences measure the period effect. A wide spread of profiles within sequence indicates high between-subject variability.

Q-Q Plots

Q-Q plots are useful for diagnosing violations of the normality and homoscedasticity assumptions. If the data in a Q-Q plot come from a normal distribution, the points will cluster tightly around the

reference line. You can use the UNIVARIATE procedure with the NORMAL option to numerically check the normality assumption.

Examples: TTEST Procedure

Example 92.1: Using Summary Statistics to Compare Group Means

This example, taken from Huntsberger and Billingsley (1989), compares two grazing methods using 32 steers. Half of the steers are allowed to graze continuously while the other half are subjected to controlled grazing time. The researchers want to know if these two grazing methods affect weight gain differently. The data are read by the following DATA step:

```
data graze;
   length GrazeType $ 10;
   input GrazeType $ WtGain @@;
   datalines;
controlled  45   controlled  62
controlled  96   controlled 128
controlled 120   controlled  99
controlled  28   controlled  50
controlled 109   controlled 115
controlled  39   controlled  96
controlled  87   controlled 100
controlled  76   controlled  80
continuous  94   continuous  12
continuous  26   continuous  89
continuous  88   continuous  96
continuous  85   continuous 130
continuous  75   continuous  54
continuous 112   continuous  69
continuous 104   continuous  95
continuous  53   continuous  21
;
run;
```

The variable GrazeType denotes the grazing method: “controlled” is controlled grazing and “continuous” is continuous grazing. The dollar sign ($) following GrazeType makes it a character variable, and the trailing at signs (@@) tell the procedure that there is more than one observation per line.

If you have summary data—that is, just means and standard deviations, as computed by PROC MEANS—then you can still use PROC TTEST to perform a simple *t* test analysis. This example demonstrates this mode of input for PROC TTEST. Note, however, that graphics are unavailable when summary statistics are used as input.

The MEANS procedure is invoked to create a data set of summary statistics with the following statements:

```
proc sort;
   by GrazeType;
proc means data=graze noprint;
   var WtGain;
   by GrazeType;
   output out=newgraze;
run;
```

The NOPRINT option eliminates all printed output from the MEANS procedure. The VAR statement tells PROC MEANS to compute summary statistics for the WtGain variable, and the BY statement requests a separate set of summary statistics for each level of GrazeType. The OUTPUT OUT= statement tells PROC MEANS to put the summary statistics into a data set called newgraze so that it can be used in subsequent procedures. This new data set is displayed in Output 92.1.1 by using PROC PRINT as follows:

```
proc print data=newgraze;
run;
```

The _STAT_ variable contains the names of the statistics, and the GrazeType variable indicates which group the statistic is from.

Output 92.1.1 Output Data Set of Summary Statistics

Obs	GrazeType	_TYPE_	_FREQ_	_STAT_	WtGain
1	continuous	0	16	N	16.000
2	continuous	0	16	MIN	12.000
3	continuous	0	16	MAX	130.000
4	continuous	0	16	MEAN	75.188
5	continuous	0	16	STD	33.812
6	controlled	0	16	N	16.000
7	controlled	0	16	MIN	28.000
8	controlled	0	16	MAX	128.000
9	controlled	0	16	MEAN	83.125
10	controlled	0	16	STD	30.535

The following statements invoke PROC TTEST with the newgraze data set, as denoted by the DATA= option:

```
proc ttest data=newgraze;
   class GrazeType;
   var WtGain;
run;
```

The CLASS statement contains the variable that distinguishes between the groups being compared, in this case GrazeType. The summary statistics and confidence intervals are displayed first, as shown in Output 92.1.2.

Output 92.1.2 Summary Statistics and Confidence Limits

The TTEST Procedure

Variable: WtGain

GrazeType	N	Mean	Std Dev	Std Err	Minimum	Maximum
continuous	16	75.1875	33.8117	8.4529	12.0000	130.0
controlled	16	83.1250	30.5350	7.6337	28.0000	128.0
Diff (1-2)		-7.9375	32.2150	11.3897		

GrazeType	Method	Mean	95% CL Mean		Std Dev
continuous		75.1875	57.1705	93.2045	33.8117
controlled		83.1250	66.8541	99.3959	30.5350
Diff (1-2)	Pooled	-7.9375	-31.1984	15.3234	32.2150
Diff (1-2)	Satterthwaite	-7.9375	-31.2085	15.3335	

GrazeType	Method	95% CL Std Dev	
continuous		.	.
controlled		.	.
Diff (1-2)	Pooled	25.7434	43.0609
Diff (1-2)	Satterthwaite		

In Output 92.1.2, The GrazeType column specifies the group for which the statistics are computed. For each class, the sample size, mean, standard deviation and standard error, and maximum and minimum values are displayed. The confidence bounds for the mean are also displayed; however, since summary statistics are used as input, the confidence bounds for the standard deviation of the groups are not calculated.

Output 92.1.3 shows the results of tests for equal group means and equal variances.

Output 92.1.3 *t* Tests

Method	Variances	DF	t Value	Pr > |t|
Pooled	Equal	30	-0.70	0.4912
Satterthwaite	Unequal	29.694	-0.70	0.4913

Equality of Variances

Method	Num DF	Den DF	F Value	Pr > F
Folded F	15	15	1.23	0.6981

A group test statistic for the equality of means is reported for both equal and unequal variances. Both tests indicate a lack of evidence for a significant difference between grazing methods ($t = -0.70$ and $p = 0.4912$ for the pooled test, $t = -0.70$ and $p = 0.4913$ for the Satterthwaite test). The equality of variances test does not indicate a significant difference in the two variances

($F' = 1.23, p = 0.6981$). Note that this test assumes that the observations in both data sets are normally distributed; this assumption can be checked in PROC UNIVARIATE by using the NORMAL option with the raw data.

Although the ability to use summary statistics as input is useful if you lack access to the original data, some of the output that would otherwise be produced in an analysis on the original data is unavailable. There are also limitations on the designs and distributional assumptions that can be used with summary statistics as input. For more information, see the section "Input Data Set of Statistics" on page 7411.

Example 92.2: One-Sample Comparison with the FREQ Statement

This example examines children's reading skills. The data consist of Degree of Reading Power (DRP) test scores from 44 third-grade children and are taken from Moore (1995, p. 337). Their scores are given in the following DATA step:

```
data read;
   input score count @@;
   datalines;
40 2   47 2   52 2   26 1   19 2
25 2   35 4   39 1   26 1   48 1
14 2   22 1   42 1   34 2   33 2
18 1   15 1   29 1   41 2   44 1
51 1   43 1   27 2   46 2   28 1
49 1   31 1   28 1   54 1   45 1
;
run;
```

The following statements invoke the TTEST procedure to test if the mean test score is equal to 30.

```
ods graphics on;

proc ttest data=read h0=30;
   var score;
   freq count;
run;

ods graphics off;
```

The ODS GRAPHICS statement requests graphical output. The count variable contains the frequency of occurrence of each test score; this is specified in the FREQ statement. The output, shown in Output 92.2.1, contains the results.

Output 92.2.1 TTEST Results

The TTEST Procedure

Variable: score

N	Mean	Std Dev	Std Err	Minimum	Maximum
44	34.8636	11.2303	1.6930	14.0000	54.0000

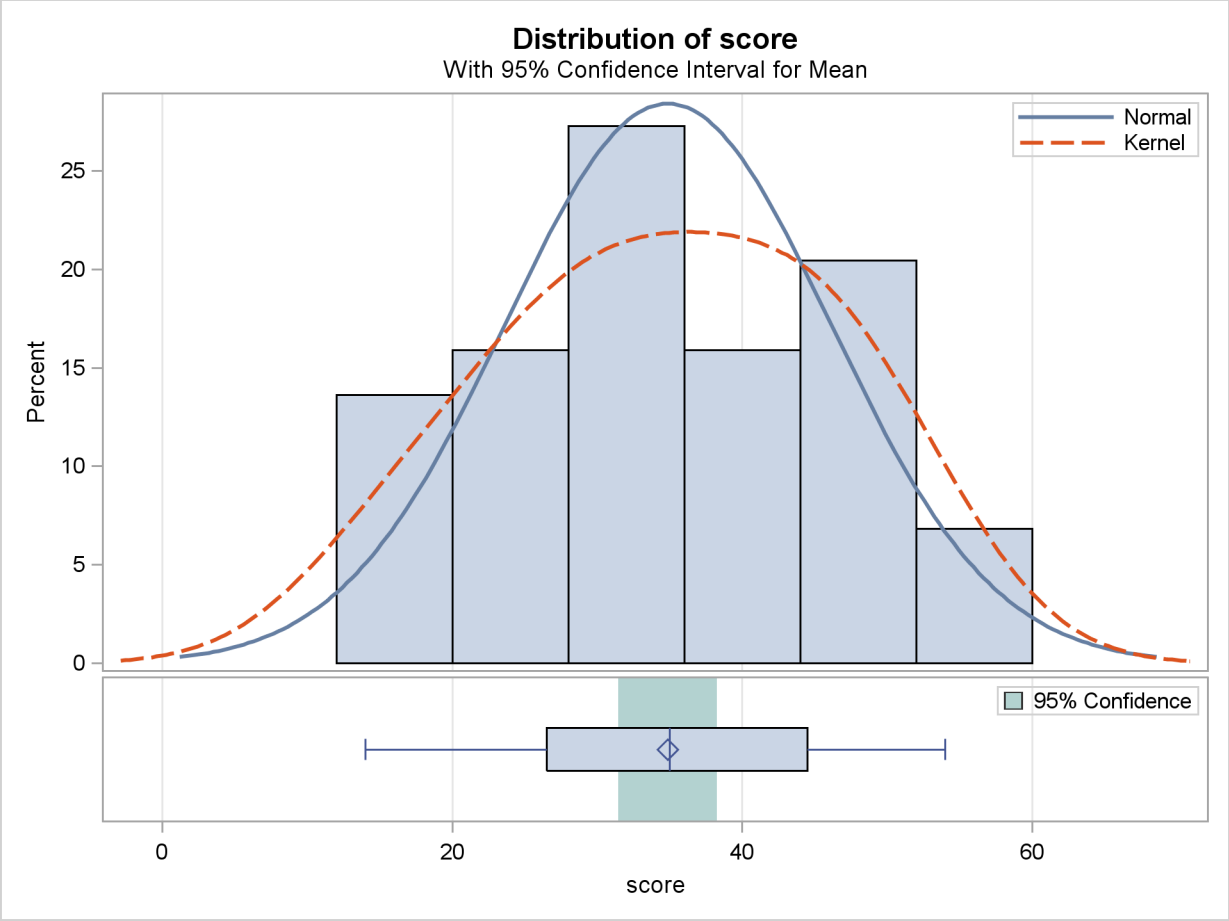
Mean	95% CL Mean		Std Dev	95% CL Std Dev	
34.8636	31.4493	38.2780	11.2303	9.2788	14.2291

DF	t Value	Pr > |t|
43	2.87	0.0063

The SAS log states that 30 observations and two variables have been read. However, the sample size given in the TTEST output is N=44. This is due to specifying the count variable in the FREQ statement. The test is significant ($t = 2.87$, $p = 0.0063$) at the 5% level, so you can conclude that the mean test score is different from 30.

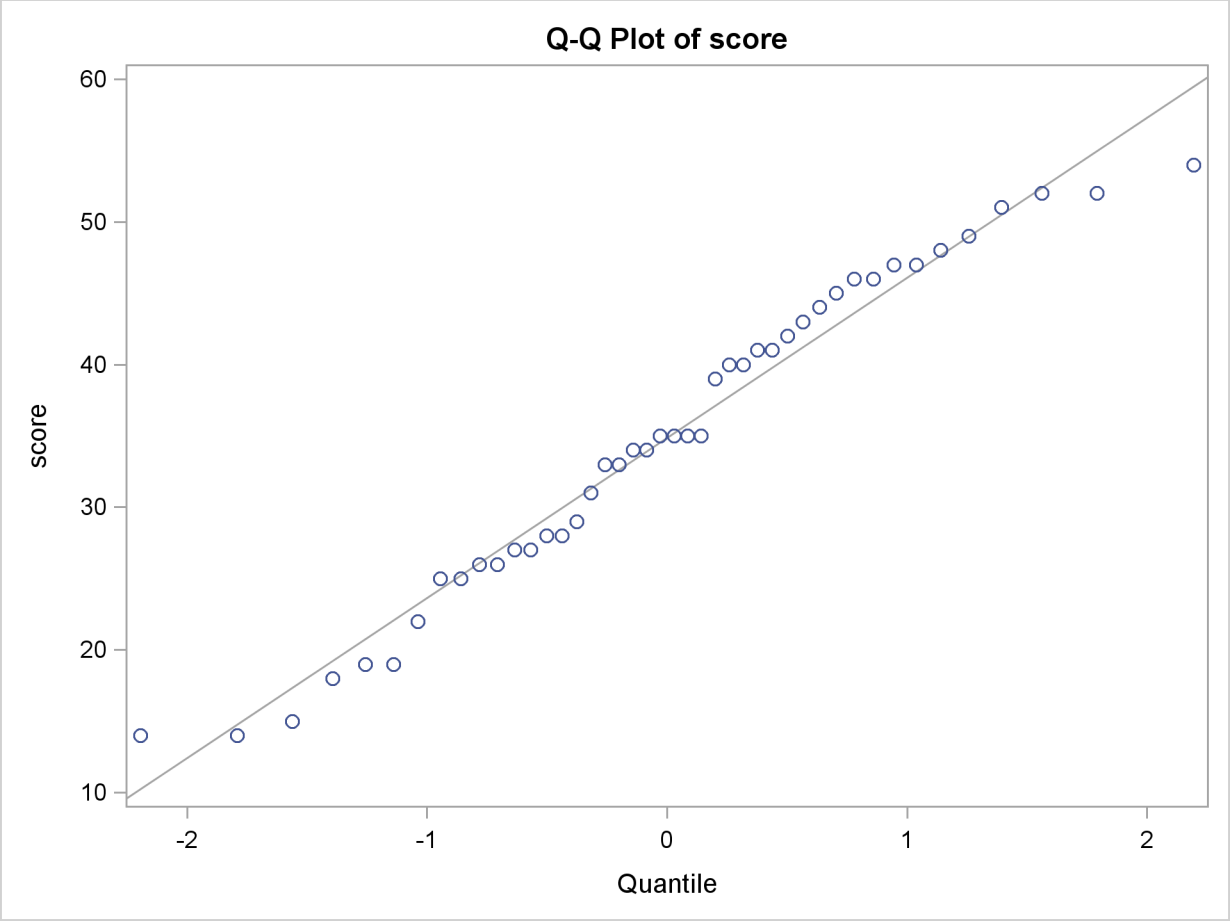
The summary panel in Output 92.2.2 shows a histogram with overlaid normal and kernel densities, a box plot, and the 95% confidence interval for the mean.

Output 92.2.2 Summary Panel



The Q-Q plot in Output 92.2.3 assesses the normality assumption.

Output 92.2.3 Q-Q Plot



The tight clustering of the points around the diagonal line is consistent with the normality assumption. You could use the UNIVARIATE procedure with the NORMAL option to numerically check the normality assumption.

Example 92.3: Paired Comparisons

When it is not feasible to assume that two groups of data are independent, and a natural pairing of the data exists, it is advantageous to use an analysis that takes the correlation into account. Using this correlation results in higher power to detect existing differences between the means. The differences between paired observations are assumed to be normally distributed. Some examples of this natural pairing are as follows:

- pre- and post-test scores for a student receiving tutoring
- fuel efficiency readings of two fuel types observed on the same automobile
- sunburn scores for two sunblock lotions, one applied to the individual's right arm, one to the left arm
- political attitude scores of husbands and wives

In this example, taken from the *SUGI Supplemental Library User's Guide, Version 5 Edition*, a stimulus is being examined to determine its effect on systolic blood pressure. Twelve men participate in the study. Each man's systolic blood pressure is measured both before and after the stimulus is applied. The following statements input the data:

```
data pressure;
      input SBPbefore SBPafter @@;
      datalines;
120 128   124 131   130 131   118 127
140 132   128 125   140 141   135 137
126 118   130 132   126 129   127 135
;
run;
```

The variables SBPbefore and SBPafter denote the systolic blood pressure before and after the stimulus, respectively.

The statements to perform the test follow:

```
ods graphics on;

proc ttest;
   paired SBPbefore*SBPafter;
run;

ods graphics off;
```

The ODS GRAPHICS statement requests graphical output. The PAIRED statement is used to test whether the mean change in systolic blood pressure is significantly different from zero. The tabular output is displayed in Output 92.3.1.

Output 92.3.1 TTEST Results

The TTEST Procedure

Difference: SBPbefore - SBPafter

N	Mean	Std Dev	Std Err	Minimum	Maximum
12	-1.8333	5.8284	1.6825	-9.0000	8.0000

Mean	95% CL Mean		Std Dev	95% CL Std Dev	
-1.8333	-5.5365	1.8698	5.8284	4.1288	9.8958

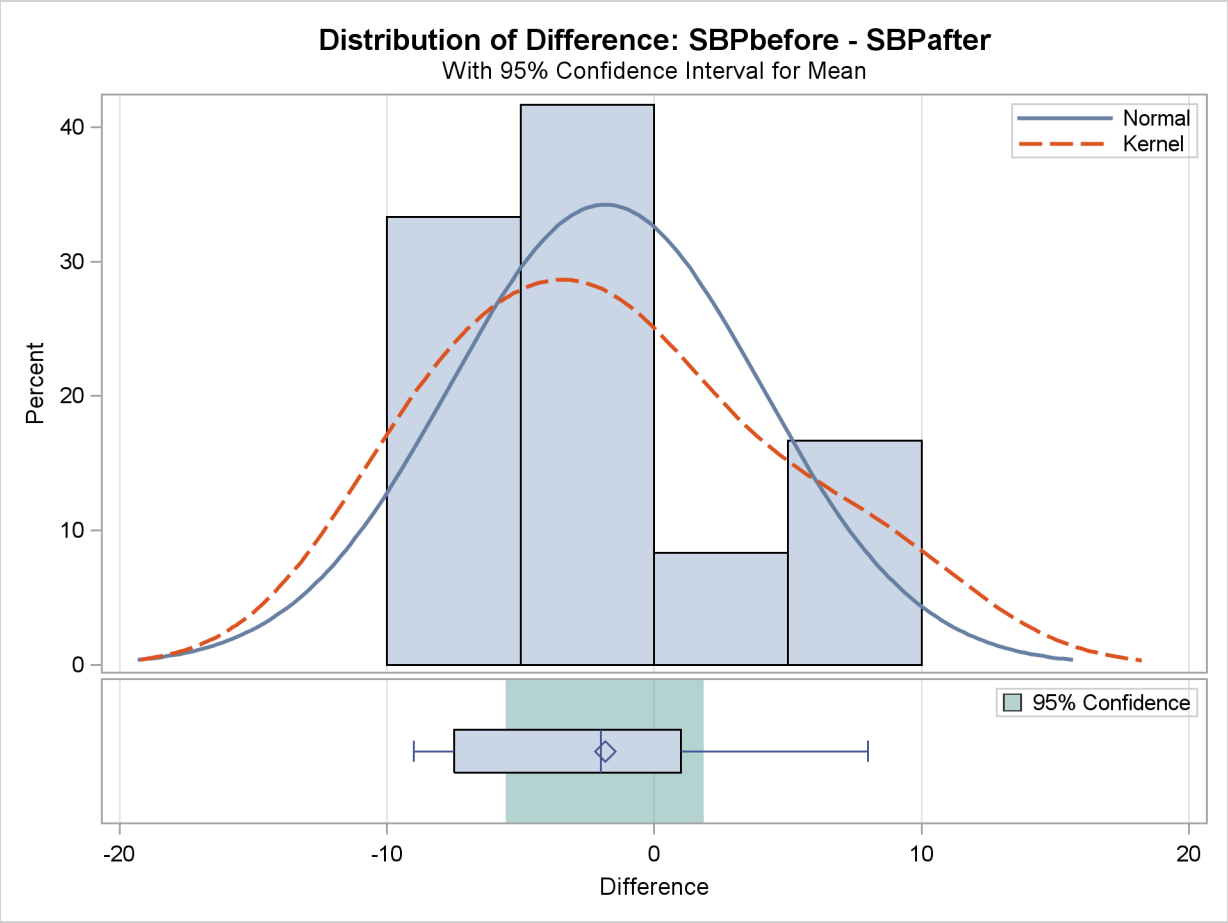
DF	t Value	Pr > |t|
11	-1.09	0.2992

The variables SBPbefore and SBPafter are the paired variables with a sample size of 12. The summary statistics of the difference are displayed (mean, standard deviation, and standard error) along with their confidence limits. The minimum and maximum differences are also displayed. The *t* test

is not significant ($t = -1.09$, $p = 0.2992$), indicating that the stimuli did not significantly affect systolic blood pressure.

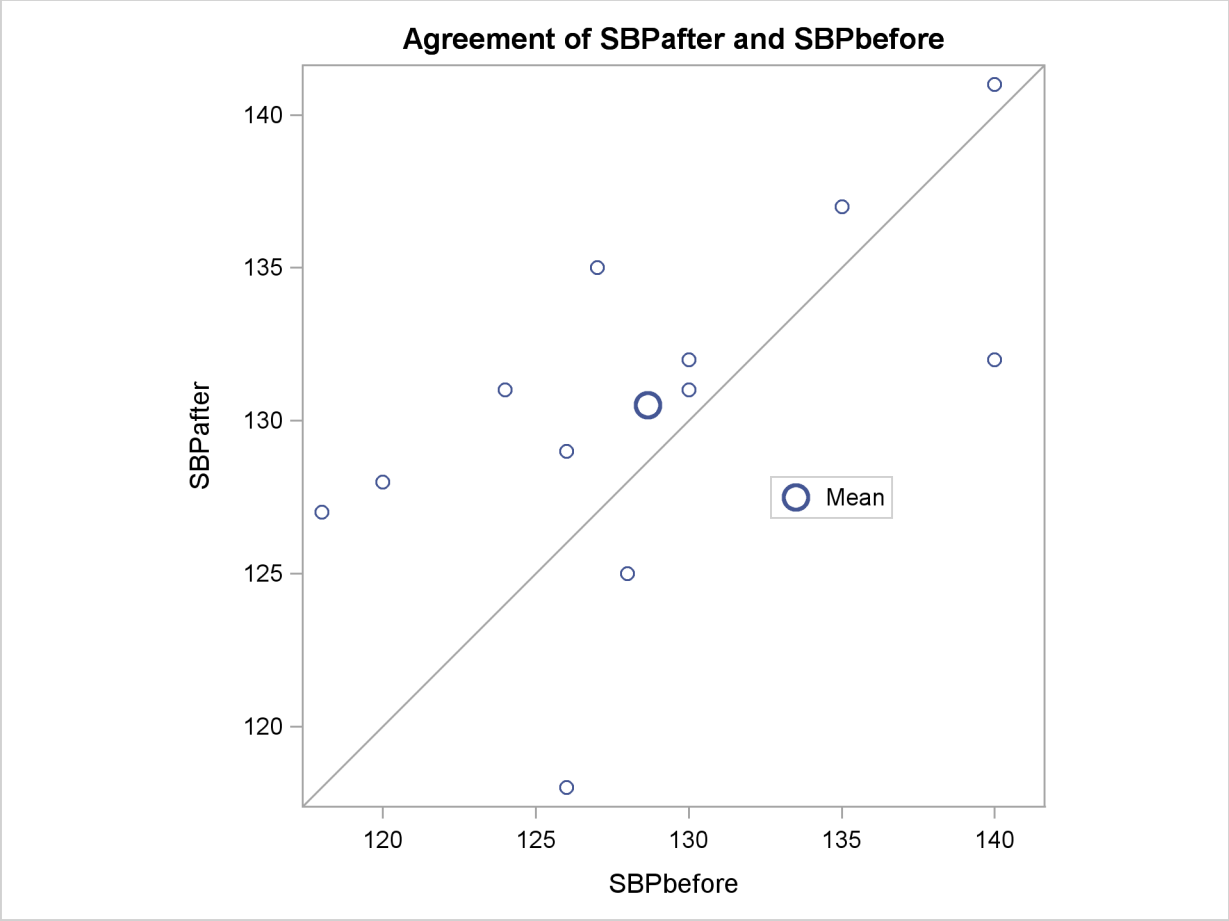
The summary panel in Output 92.3.2 shows a histogram, normal and kernel densities, box plot, and $100(1-\alpha)\% = 95\%$ confidence interval of the SBPbefore − SBPafter difference.

Output 92.3.2 Summary Panel



The agreement plot in Output 92.3.3 reveals that only three men have higher blood pressure before the stimulus than after.

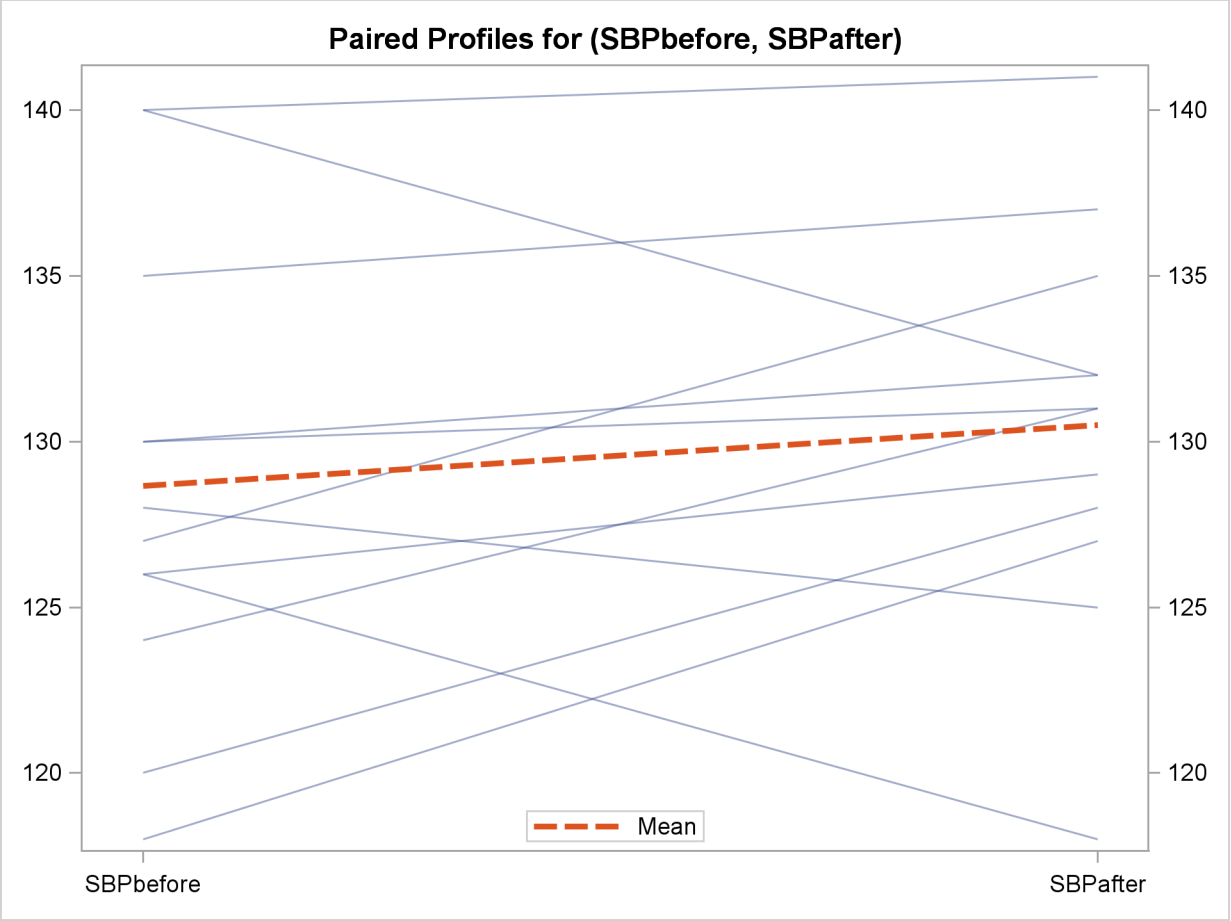
Output 92.3.3 Agreement of Treatments



But the differences for these men are relatively large, keeping the mean difference only slightly negative.

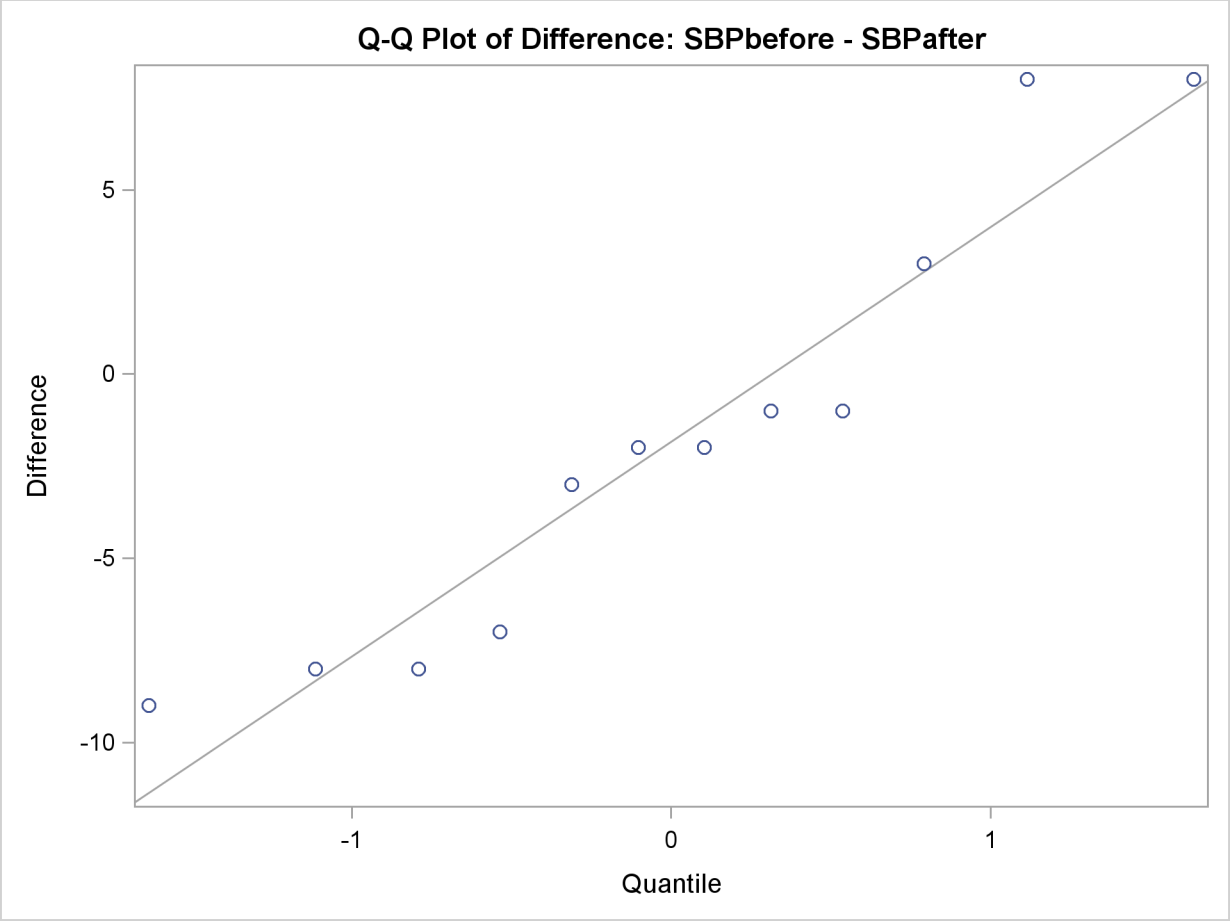
The profiles plot in Output 92.3.4 is a different view of the same information contained in Output 92.3.3, plotting the blood pressure from before to after the stimulus.

Output 92.3.4 Profiles over Treatments



The Q-Q plot in Output 92.3.5 assesses the normality assumption.

Output 92.3.5 Q-Q Plot



The Q-Q plot shows no obvious deviations from normality. You can check the assumption of normality more rigorously by using PROC UNIVARIATE with the NORMAL option.

Example 92.4: AB/BA Crossover Design

Senn (2002, Chapter 3) discusses a study comparing the effectiveness of two bronchodilators, formoterol ("for") and salbutamol ("sal"), in the treatment of childhood asthma. A total of 13 children are recruited for an AB/BA crossover design. A random sample of 7 of the children are assigned to the treatment sequence for/sal, receiving a dose of formoterol upon an intial visit ("period 1") and then a dose of salbutamol upon a later visit ("period 2"). The other 6 children are assigned to the sequence sal/for, receiving the treatments in the reverse order but otherwise in a similar manner. Periods 1 and 2 are sufficiently spaced so that no carryover effects are suspected. After a child inhales a dose of a bronchodilator, peak expiratory flow (PEF) is measured. Higher PEF indicates greater effectiveness. The data are assumed to be approximately normally distributed.

The data set is generated with the following statements:

```
data asthma;
   input Drug1 $ Drug2 $ PEF1 PEF2 @@;
   datalines;
for sal 310 270   for sal 310 260  for sal 370 300
for sal 410 390   for sal 250 210  for sal 380 350
for sal 330 365
sal for 370 385   sal for 310 400  sal for 380 410
sal for 290 320   sal for 260 340  sal for 90  220
;
run;
```

You can display the data by using the following statements, which produce Output 92.4.1:

```
proc print data=asthma;
run;
```

Output 92.4.1 Asthma Study Data

Obs	Drug1	Drug2	PEF1	PEF2
1	for	sal	310	270
2	for	sal	310	260
3	for	sal	370	300
4	for	sal	410	390
5	for	sal	250	210
6	for	sal	380	350
7	for	sal	330	365
8	sal	for	370	385
9	sal	for	310	400
10	sal	for	380	410
11	sal	for	290	320
12	sal	for	260	340
13	sal	for	90	220

The variables PEF1 and PEF2 represent the responses for the first and second periods, respectively. The variables Drug1 and Drug2 represent the treatment in each period.

You can analyze this crossover design by using the CROSSOVER option after a slash (/) in the VAR statement:

```
ods graphics on;

proc ttest data=asthma plots=interval;
   var PEF1 PEF2 / crossover= (Drug1 Drug2);
run;

ods graphics off;
```

The ODS GRAPHICS statement requests graphical output. With the default PROC TTEST options TEST=DIFF and DIST=NORMAL and the lack of the IGNOREPERIOD option in the VAR statement, both the treatment difference and the period difference are assessed. The PROC TTEST default options H0=0, SIDES=2, and ALPHA=0.05 specify a two-sided analysis with 95% confidence limits comparing treatment and period differences to a default difference of zero. The

default CI=EQUAL option in the PROC TTEST statement requests equal-tailed confidence intervals for standard deviations. The PLOTS=INTERVAL option produces TYPE=TREATMENT confidence intervals, in addition to the default plots AGREEMENT(TYPE=TREATMENT), BOX, HISTOGRAM, PROFILES(TYPE=TREATMENT), and QQ.

Output 92.4.2 summarizes the response and treatment variables for each period.

Output 92.4.2 Crossover Variable Information

The TTEST Procedure

Response Variables: PEF1, PEF2

Crossover Variable Information

Period	Response	Treatment
1	PEF1	Drug1
2	PEF2	Drug2

Output 92.4.3 displays basic summary statistics (sample size, mean, standard deviation, standard error, minimum, and maximum) for each of the four cells in the design, the treatment difference within each treatment sequence, the overall treatment difference, and the overall period difference.

Output 92.4.3 Statistics

Sequence	Treatment	Period	N	Mean	Std Dev	Std Err
1	for	1	7	337.1	53.7631	20.3206
2	for	2	6	345.8	70.8814	28.9372
2	sal	1	6	283.3	105.4	43.0245
1	sal	2	7	306.4	64.7247	24.4636
1	Diff (1-2)		7	30.7143	32.9682	12.4608
2	Diff (1-2)		6	62.5000	44.6934	18.2460
Both	Diff (1-2)			46.6071	19.3702	10.7766
Both		Diff (1-2)		-15.8929	19.3702	10.7766

Sequence	Treatment	Period	Minimum	Maximum
1	for	1	250.0	410.0
2	for	2	220.0	410.0
2	sal	1	90.0000	380.0
1	sal	2	210.0	390.0
1	Diff (1-2)		-35.0000	70.0000
2	Diff (1-2)		15.0000	130.0
Both	Diff (1-2)			
Both		Diff (1-2)		

The treatment difference "Diff (1-2)" corresponds to the "for" treatment minus the "sal" treatment, because "for" appears before "sal" in the output, according to the ORDER=MIXED default PROC TTEST option. Its mean estimate is 46.6071, favoring formoterol over salbutamol. The standard deviation (Std Dev) reported for a "difference" is actually the pooled standard deviation across both

treatment sequence (for/sal and sal/for), assuming equal variances. The standard error (Std Err) is the standard deviation of the mean estimate.

The top half of the table in Output 92.4.4 shows 95% two-sided confidence limits for the means for the same criteria addressed in the table in Output 92.4.3.

Output 92.4.4 Confidence Limits

Sequence	Treatment	Period	Method	Mean	95% CL Mean	
1	for	1		337.1	287.4	386.9
2	for	2		345.8	271.4	420.2
2	sal	1		283.3	172.7	393.9
1	sal	2		306.4	246.6	366.3
1	Diff (1-2)			30.7143	0.2238	61.2048
2	Diff (1-2)			62.5000	15.5972	109.4
Both	Diff (1-2)		Pooled	46.6071	22.8881	70.3262
Both	Diff (1-2)		Satterthwaite	46.6071	21.6585	71.5558
Both		Diff (1-2)	Pooled	-15.8929	-39.6119	7.8262
Both		Diff (1-2)	Satterthwaite	-15.8929	-40.8415	9.0558

Sequence	Treatment	Period	Method	Std Dev	95% CL Std Dev	
1	for	1		53.7631	34.6446	118.4
2	for	2		70.8814	44.2447	173.8
2	sal	1		105.4	65.7841	258.5
1	sal	2		64.7247	41.7082	142.5
1	Diff (1-2)			32.9682	21.2445	72.5982
2	Diff (1-2)			44.6934	27.8980	109.6
Both	Diff (1-2)		Pooled	19.3702	13.7217	32.8882
Both	Diff (1-2)		Satterthwaite			
Both		Diff (1-2)	Pooled	19.3702	13.7217	32.8882
Both		Diff (1-2)	Satterthwaite			

For the mean differences, both pooled (assuming equal variances for both treatment sequences) and Satterthwaite (assuming unequal variances) intervals are shown. For example, the pooled confidence limits for the overall treatment mean difference (for − sal) assuming equal variances are 22.8881 and 70.3262.

The bottom half of Output 92.4.4 shows 95% equal-tailed confidence limits for the standard deviations within each cell and for the treatment difference within each sequence. It also shows confidence limits for the pooled common standard deviation assuming equal variances. Note that the pooled standard deviation of 19.3702 and associated confidence limits 13.7217 and 32.8882 apply to both difference tests (treatment and period), since each of those tests involves the same pooled standard deviation.

Output 92.4.5 shows the results of t tests of treatment and period differences.

Output 92.4.5 *t* Tests

Treatment	Period	Method	Variances	DF	t Value	Pr > |t|
Diff (1-2)		Pooled	Equal	11	4.32	0.0012
Diff (1-2)		Satterthwaite	Unequal	9.1017	4.22	0.0022
	Diff (1-2)	Pooled	Equal	11	-1.47	0.1683
	Diff (1-2)	Satterthwaite	Unequal	9.1017	-1.44	0.1838

Both pooled and Satterhthwaite versions of the test of treatment difference are highly significant ($p = 0.0012$ and $p = 0.0022$), and both versions of the test of period difference are insignificant ($p = 0.1683$ and $p = 0.1838$).

The folded F test of equal variances in each treatment sequence is shown in Output 92.4.6.

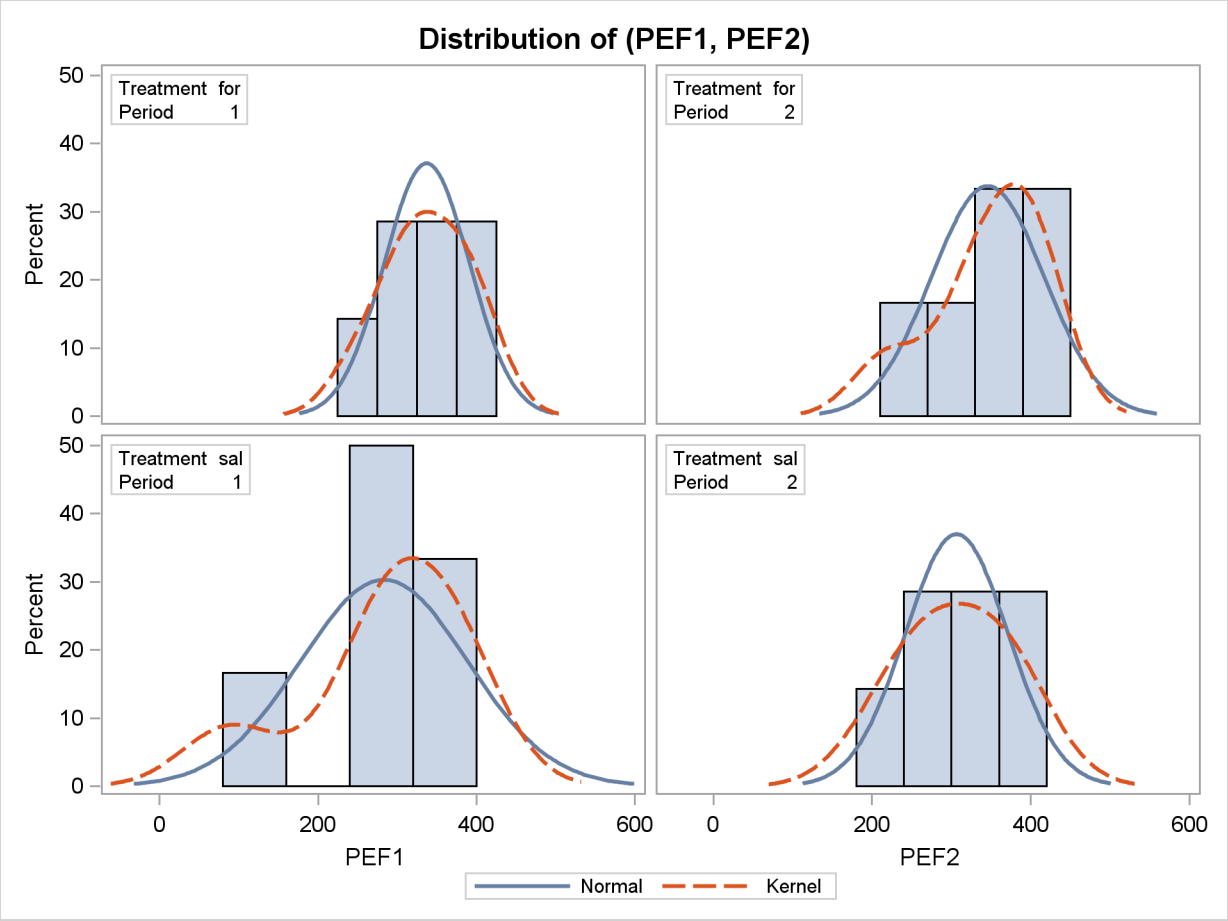
Output 92.4.6 Equality of Variances Test

Equality of Variances				
Method	Num DF	Den DF	F Value	Pr > F
Folded F	5	6	1.84	0.4797

The insignificant result ($p = 0.48$) implies a lack of evidence for unequal variances. However, it does not demonstrate equal variances, and it is not very robust to deviations from normality.

Output 92.4.7 shows the distribution of the response variables PEF1 and PEF2 within each of the four cells (combinations of two treatments and two periods) of the AB/BA crossover design, in terms of histograms and normal and kernel density estimates.

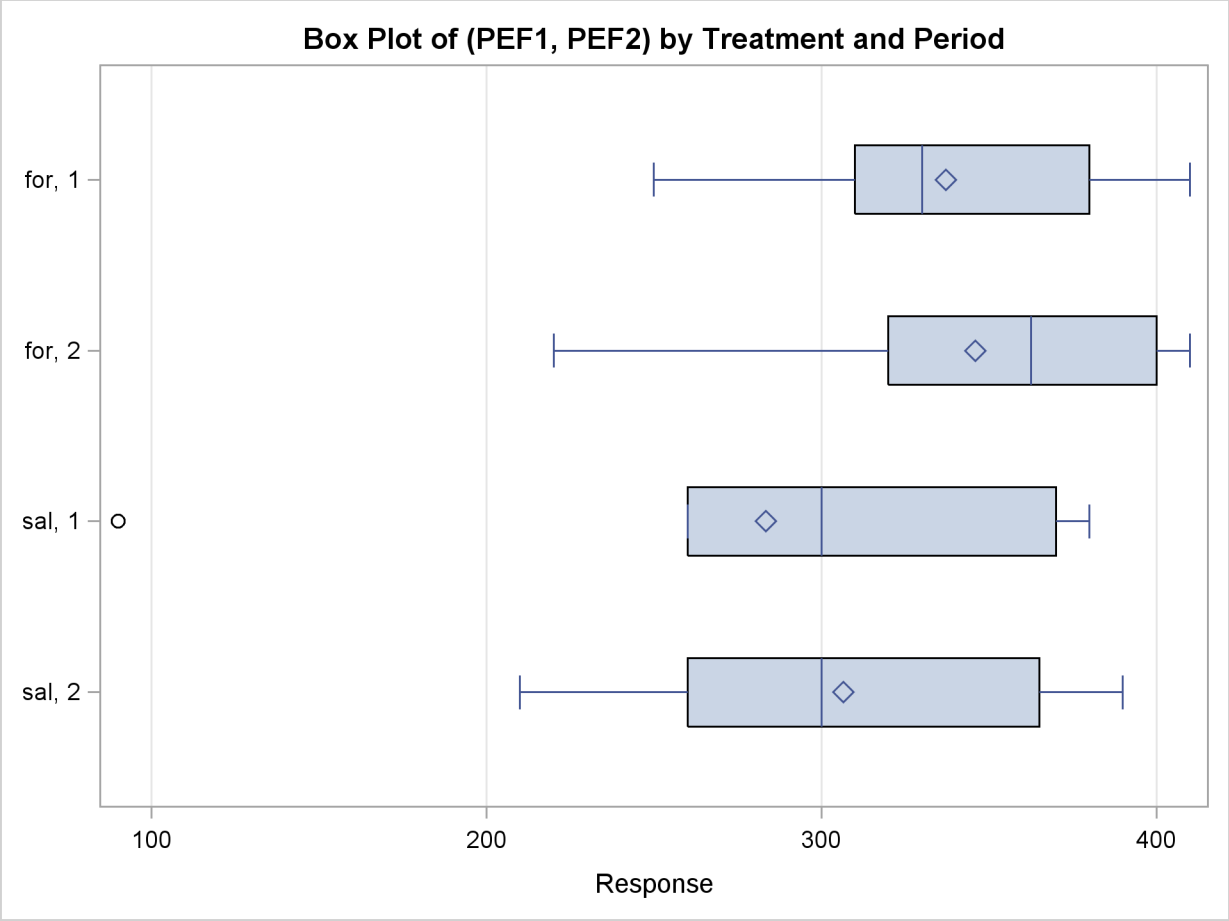
Output 92.4.7 Comparative Histograms



The distributions for the first treatment sequence (for/sal) appear to be somewhat symmetric, and the distributions for the sal/for sequence appear to be skewed to the left.

Output 92.4.8 shows a similar distributional summary but in terms of box plots.

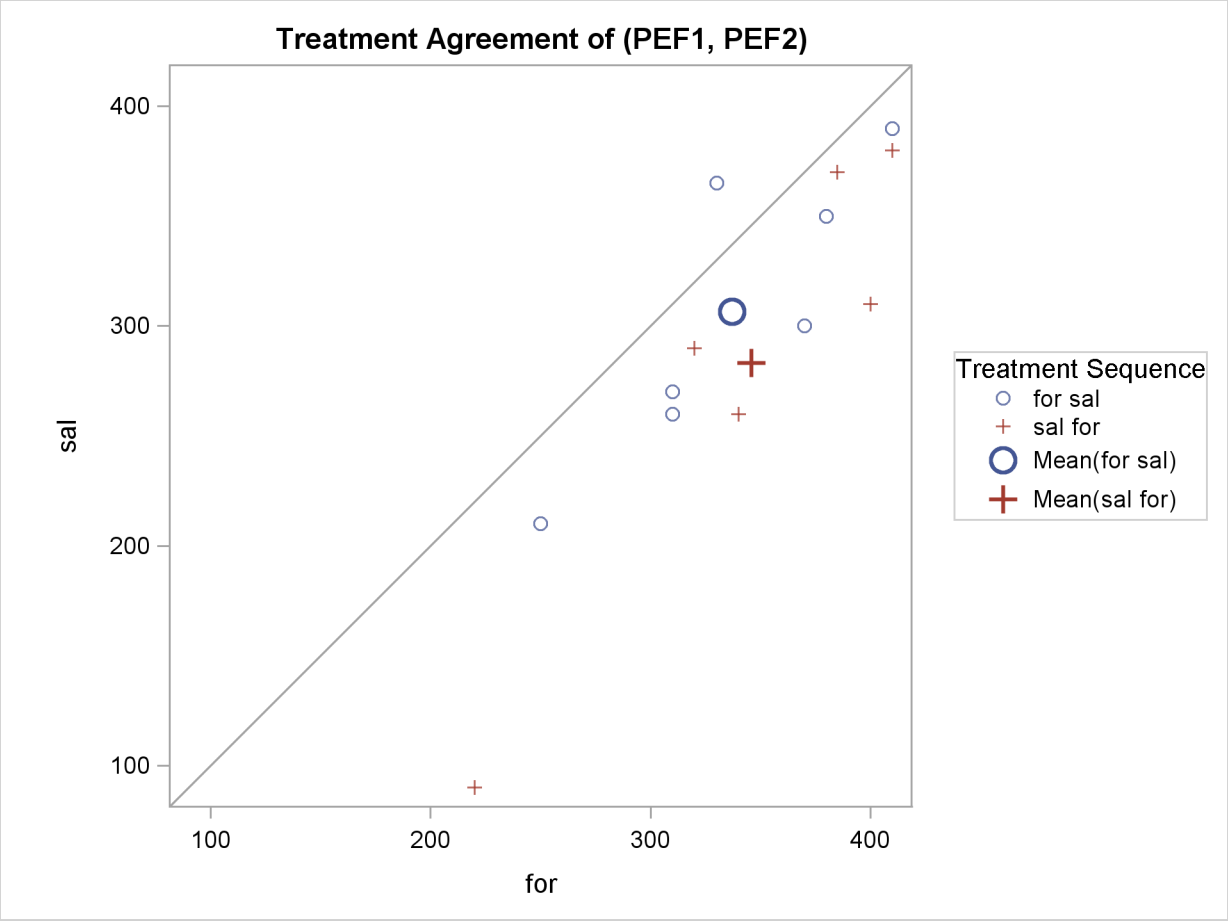
Output 92.4.8 Comparative Box Plots



The relative locations of means and medians in each box plot corroborate the fact that the distributions for the sal/for sequence are skewed to the left. The distributions for the for/sal sequence appear to be skewed slightly to the right. The box plot for the salbutamol treatment in the first period shows an outlier (the circle on the far left side of the plot).

The treatment agreement plot in Output 92.4.9 reveals that only a single observation has a higher peak expiratory flow for salbutamol.

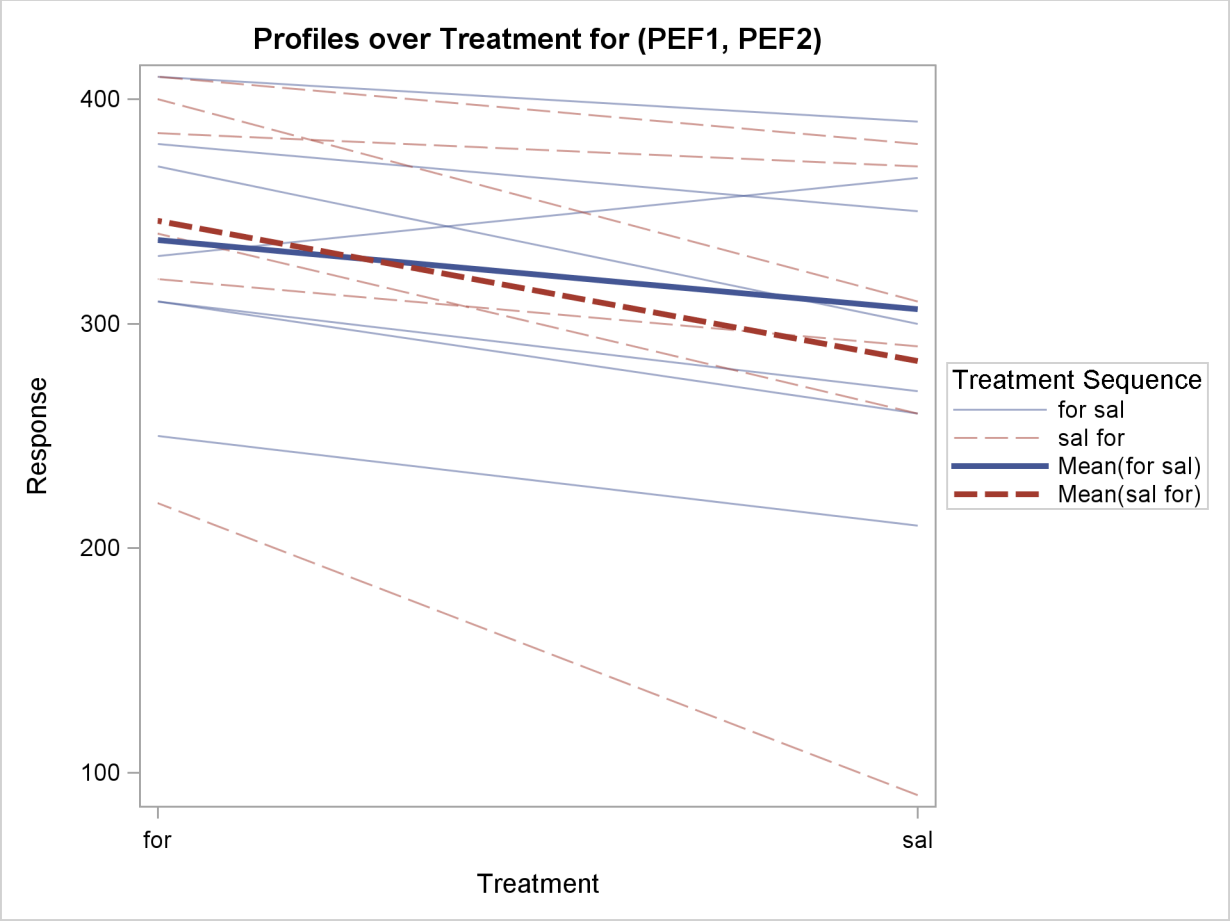
Output 92.4.9 Agreement of Treatments Plot



The mean for the sal/for treatment sequence is farther from the diagonal equivalence line, revealing that the treatment difference is more pronounced for the 6 observations in the sal/for sequence than for the 7 observations in the for/sal sequence. This fact is also seen numerically in Output 92.4.3 and Output 92.4.4, which show within-sequence treatment differences of 30.7 for for/sal and 62.5 for sal/for.

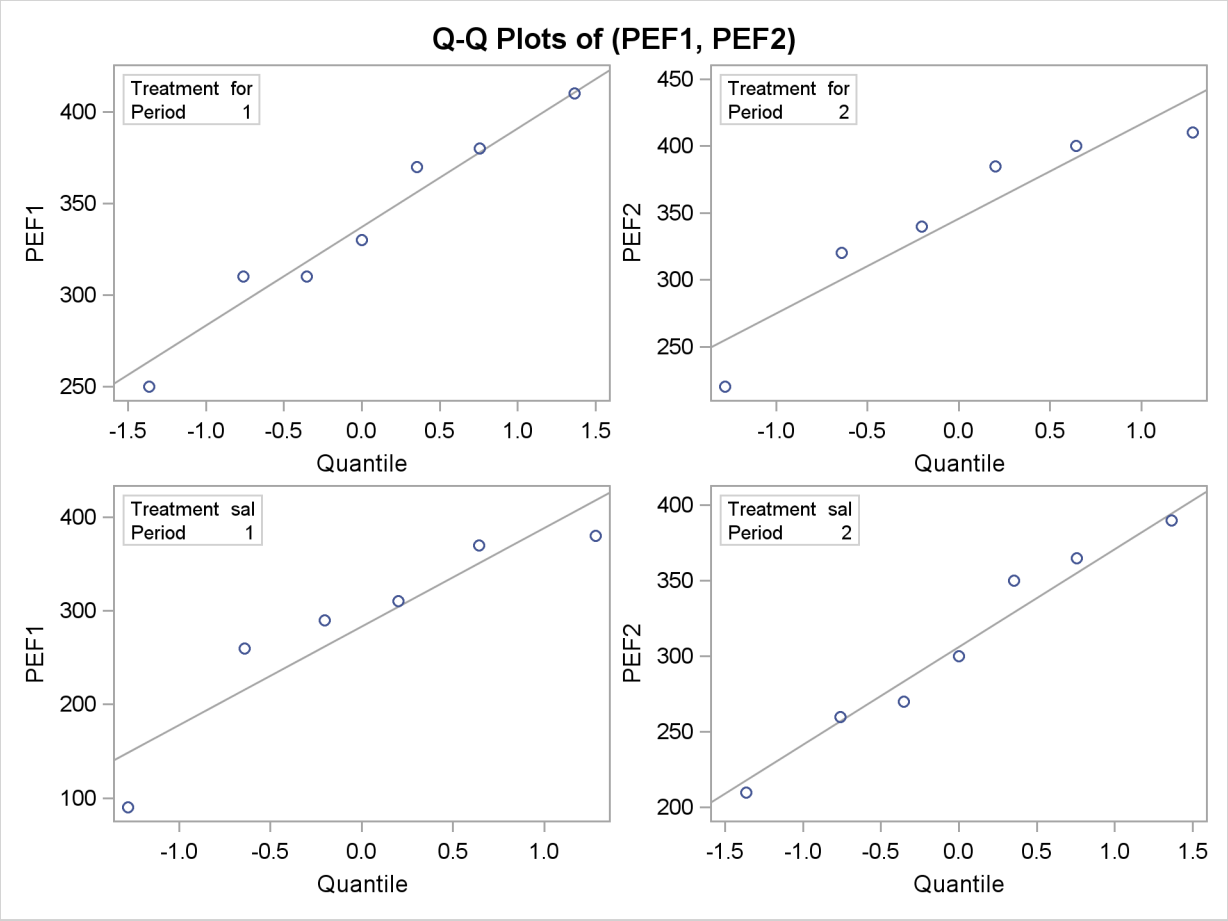
The profiles over treatment plot in Output 92.4.10 is a different view of the same information contained in Output 92.4.9, plotting the profiles from formoterol to salbutamol treatments. The lone observation for which the peak expiratory flow is higher for salbutamol appears as the only line with negative slope.

Output 92.4.10 Profiles over Treatment



The Q-Q plots in Output 92.4.11 assess normality assumption within each of the four cells of the design.

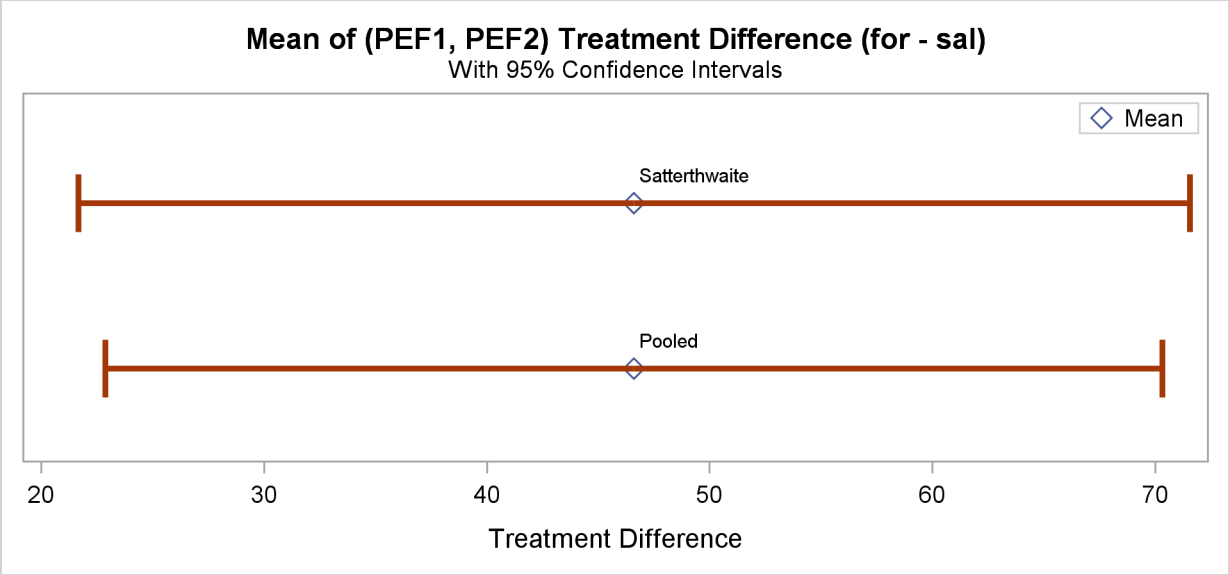
Output 92.4.11 Q-Q Plots



The two Q-Q plots for the sal/for sequence (lower left and upper right) suggest some possible normality violations in the tails, but the sample size is too small to make any strong conclusions. You could use the UNIVARIATE procedure with the NORMAL option to numerically check the normality assumptions.

Finally, Output 92.4.12 shows both pooled and Satterthwaite two-sided 95% confidence intervals for the treatment difference.

Output 92.4.12 Confidence Intervals for Treatment Difference



The pooled interval is slightly smaller than the Satterthwaite interval. (This is not always the case.)

Example 92.5: Equivalence Testing with Lognormal Data

Wellek (2003, p. 212) discusses an average bioequivalence study comparing the AUC (area under serum-concentration curve) measurements for two different drugs, denoted "Test" and "Reference," over a period of 20 hours. This example looks at a portion of Wellek's data, conducting an equivalence analysis with a paired design that uses AUC values on the original scale (assumed to be lognormally distributed). Each subject in the study received the Test drug upon one visit and then the Reference drug upon a later visit, sufficiently spaced so that no carryover effects would occur.

The goal is to test whether the geometric mean AUC ratio between Test and Reference is between 0.8 and 1.25, corresponding to the traditional FDA (80%, 125%) equivalence criterion. See the section "Arithmetic and Geometric Means" on page 7412 for a discussion of the use of geometric means for lognormal data.

The following SAS statements generate the data set:

```
data auc;
input TestAUC RefAUC @@;
datalines;
103.4 90.11  59.92 77.71  68.17 77.71  94.54 97.51
69.48 58.21  72.17 101.3  74.37 79.84  84.44 96.06
96.74 89.30  94.26 97.22  48.52 61.62  95.68 85.80
;
run;
```

You can display the data by using the following statements, which produce Output 92.5.1:

```
proc print data=auc;
run;
```

Output 92.5.1 AUC Data for Test and Reference Drugs

Obs	Test AUC	RefAUC
1	103.40	90.11
2	59.92	77.71
3	68.17	77.71
4	94.54	97.51
5	69.48	58.21
6	72.17	101.30
7	74.37	79.84
8	84.44	96.06
9	96.74	89.30
10	94.26	97.22
11	48.52	61.62
12	95.68	85.80

The TestAUC and RefAUC variables represent the AUC measurements for each subject under the Test and Reference drugs, respectively. Use the following SAS statements to perform the equivalence analysis:

```
ods graphics on;

proc ttest data=auc dist=lognormal tost(0.8, 1.25);
   paired TestAUC*RefAUC;
run;

ods graphics off;
```

The ODS GRAPHICS statement requests graphical output. The DIST=LOGNORMAL option specifies the lognormal distributional assumption and requests an analysis in terms of geometric mean and coefficient of variation. The TOST option specifies the equivalence bounds 0.8 and 1.25.

Output 92.5.2 shows basic summary statistics for the ratio of TestAUC to RefAUC.

Output 92.5.2 Summary Statistics

The TTEST Procedure

Ratio: TestAUC / RefAUC

N	Geometric Mean	Coefficient of Variation	Minimum	Maximum
12	0.9412	0.1676	0.7124	1.1936

The geometric mean ratio of 0.9412 is the sample mean of the log-transformed data exponentiated

to bring it back to the original scale. So the plasma concentration over the 20-hour period is slightly lower for the Test drug than for the Reference drug. The CV of 0.1676 is the ratio of the standard deviation to the (arithmetic) mean.

Output 92.5.3 shows the $100(1-\alpha)\% = 95\%$ confidence limits for the geometric mean ratio (0.8467 and 1.0462) and CV (0.1183 and 0.2884).

Output 92.5.3 Confidence Limits

Geometric Mean	95% CL Mean		Coefficient of Variation	95% CL CV	
0.9412	0.8467	1.0462	0.1676	0.1183	0.2884

Output 92.5.4 shows the $100(1-2\alpha)\% = 90\%$ confidence limits for the geometric mean ratio, 0.8634 and 1.0260.

Output 92.5.4 Equivalence Limits

Geometric Mean	Lower Bound		90% CL Mean		Upper Bound	Assessment
0.9412	0.8	<	0.8634	1.0260 <	1.25	Equivalent

The assessment of "Equivalent" reflects the fact that these limits are contained within the equivalence bounds 0.8 and 1.25. This result occurs if and only if the p-value of the test is less than the α value specified in the ALPHA= option in the PROC TTEST statement, and it is the reason that $100(1-2\alpha)\%$ confidence limits are shown in addition to the usual $100(1-\alpha)\%$ limits.

Output 92.5.5 shows the p-values for the two one-sided tests against the upper and lower equivalence bounds.

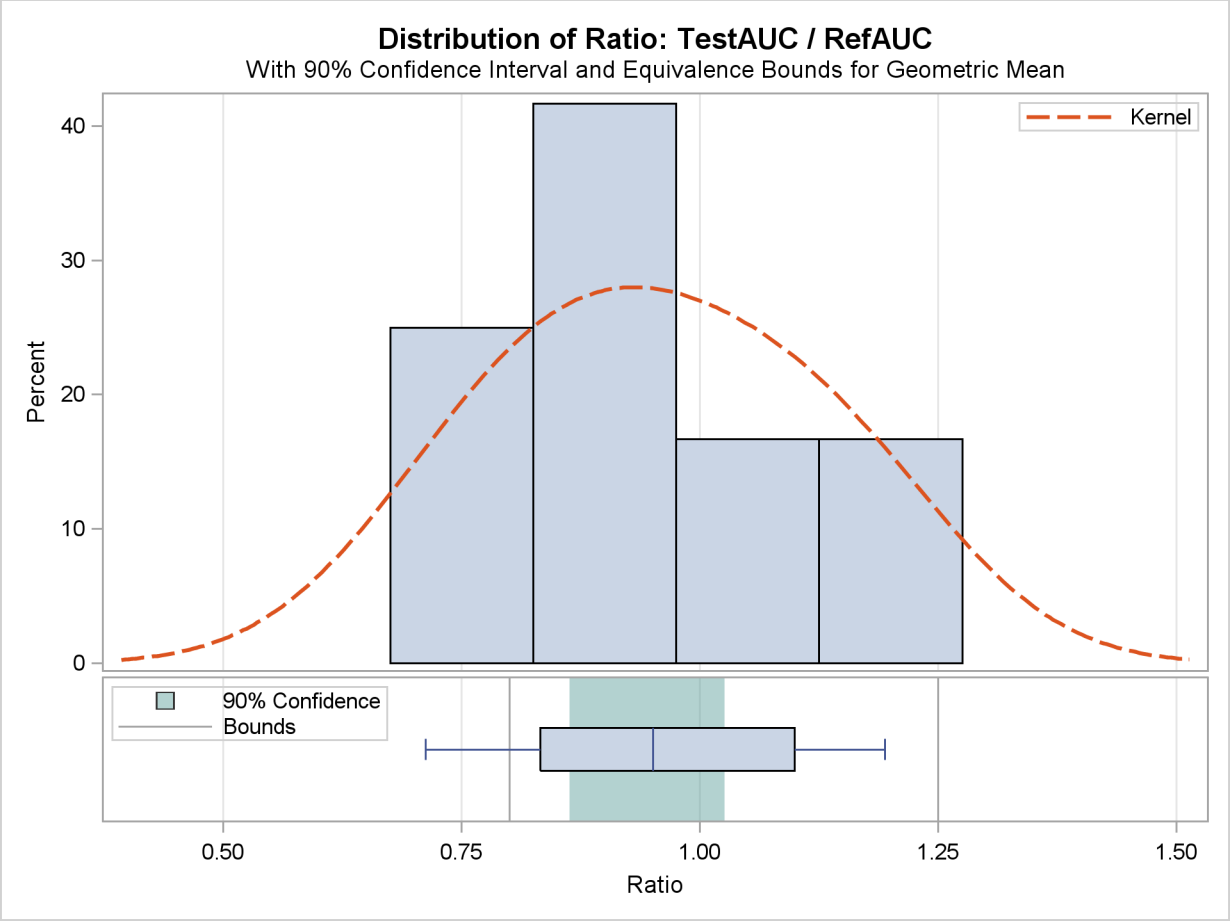
Output 92.5.5 TOST Equivalence Test

Test	Null	DF	t Value	P-Value
Upper	0.8	11	3.38	0.0031
Lower	1.25	11	-5.90	<.0001
Overall				0.0031

The overall p-value of 0.0031, the larger of the two one-sided p-values, indicates significant evidence of equivalence between the Test and Reference drugs.

The summary panel in Output 92.5.6 shows a histogram, kernel density, box plot, and $100(1-2\alpha)\% = 90\%$ confidence interval of the Test-to-Reference ratio of AUC, along with the equivalence bounds.

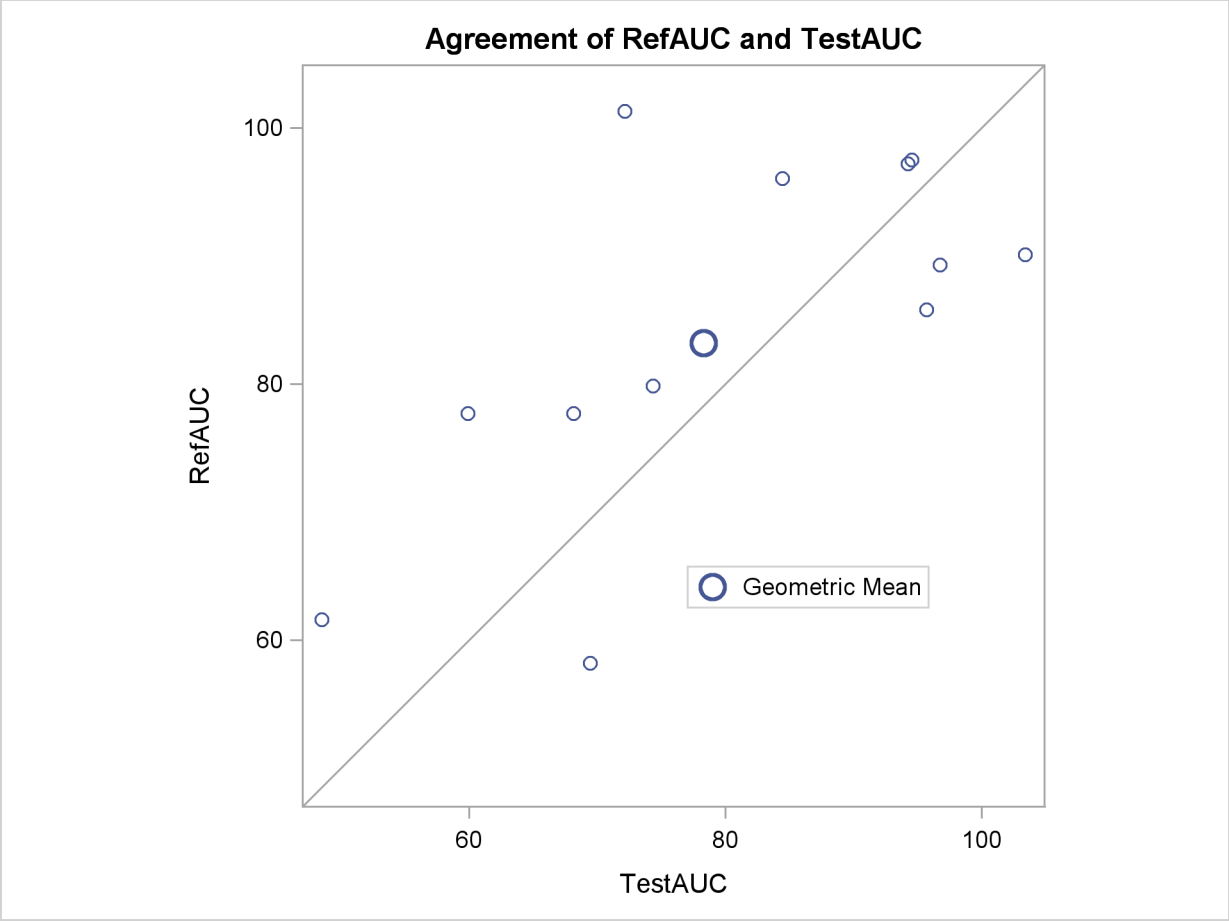
Output 92.5.6 Summary Panel



The confidence interval is closer to the lower equivalence bound than the upper bound and contained entirely within the bounds.

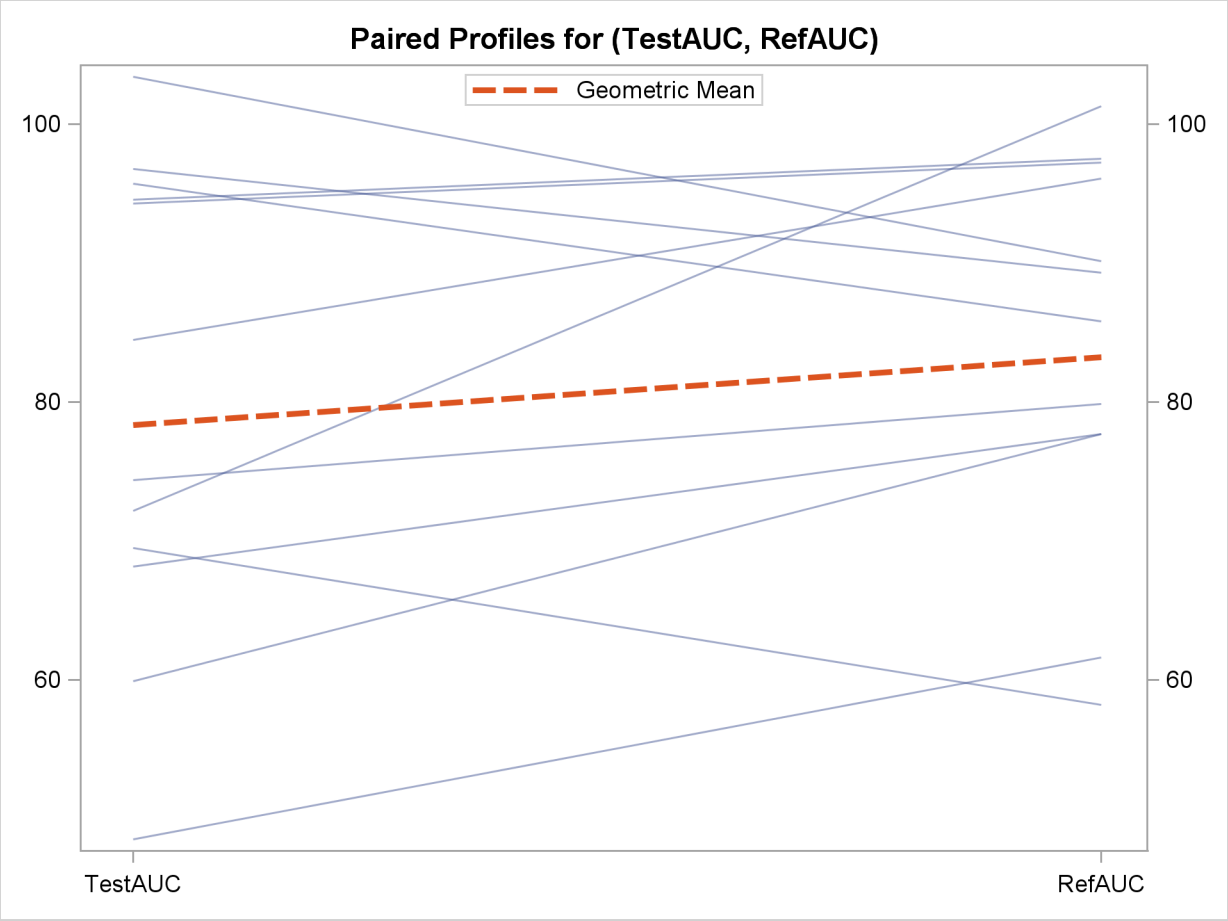
The agreement plot in Output 92.5.7 reveals that the only four subjects with higher AUC for the Test drug are at the far lower or far upper end of the AUC distribution. This might merit further investigation.

Output 92.5.7 Agreement Plot



The profiles plot in Output 92.5.8 is a different view of the same information contained in Output 92.5.7, plotting the AUC from Test to Reference drug.

Output 92.5.8 Profiles Plot



References

Best, D. I. and Rayner, C. W. (1987), "Welch's Approximate Solution for the Behren's-Fisher Problem," *Technometrics*, 29, 205–210.

Chow, S. and Liu, J. (2000), *Design and Analysis of Bioavailability and Bioequivalence Studies*, Second Edition, New York: Marcel Dekker.

Cochran, W. G. and Cox, G. M. (1950), *Experimental Designs*, New York: John Wiley & Sons.

Dilba, G., Schaarschmidt, F., and Hothorn, L. A. (2006), *mratios: A Package for Inference about Ratios of Normal Means,* UseR! Conference.

Diletti, D., Hauschke, D., and Steinijans, V. W. (1991), "Sample Size Determination for Bioequivalence Assessment by Means of Confidence Intervals," *International Journal of Clinical Pharmacology, Therapy and Toxicology*, 29, 1–8.

Fieller, E. C. (1954), "Some Problems in Interval Estimation," *Journal of the Royal Statistical*

Society Series B, 16, 175–185.

Hauschke, D., Kieser, M., Diletti, E., and Burke, M. (1999), "Sample Size Determination for Proving Equivalence Based on the Ratio of Two Means for Normally Distributed Data," *Statistics in Medicine*, 18, 93–105.

Huntsberger, David V. and Billingsley, Patrick P. (1989), *Elements of Statistical Inference*, Dubuque, IA: Wm. C. Brown.

Johnson, N. L. Kotz, S. and Balakrishnan, N. (1994), *Continuous Univariate Distributions, Volume 1*, Second Edition, New York: John Wiley & Sons.

Jones, B. and Kenward, M. G. (2003), *Design and Analysis of Cross-Over Trials*, Second Edition, Washington, DC: Chapman & Hall/CRC.

Lee, A. F. S. and Gurland, J. (1975), "Size and Power of Tests for Equality of Means of Two Normal Populations with Unequal Variances," *Journal of the American Statistical Association*, 70, 933–941.

Lehmann, E. L. (1986), *Testing Statistical Hypostheses*, New York: John Wiley & Sons.

Moore, David S. (1995), *The Basic Practice of Statistics*, New York: W. H. Freeman.

Phillips, K. F. (1990), "Power of the Two One-Sided Tests Procedure in Bioequivalence," *Journal of Pharmacokinetics and Biopharmaceutics*, 18, 137–144.

Posten, H. O., Yeh, Y. Y., and Owen, D. B. (1982), "Robustness of the Two-Sample t Test under Violations of the Homogeneity of Variance Assumption," *Communications in Statistics*, 11, 109–126.

Ramsey, P. H. (1980), "Exact Type I Error Rates for Robustness of Student's t Test with Unequal Variances," *Journal of Educational Statistics*, 5, 337–349.

Robinson, G. K. (1976), "Properties of Student's t and of the Behrens-Fisher Solution to the Two Mean Problem," *Annals of Statistics*, 4, 963–971.

SAS Institute Inc. (1986), *SUGI Supplemental Library User's Guide, Version 5 Edition*, Cary, NC: SAS Institute Inc.

Sasabuchi, S. (1988a), "A Multivariate Test with Composite Hypotheses Determined by Linear Inequalities When the Covariance Matrix Has an Unknown Scale Factor," *Memoirs of the Faculty of Science, Kyushu University, Series A*, 42, 9–19.

Sasabuchi, S. (1988b), "A Multivariate Test with Composite Hypotheses When the Covariance Matrix Is Completely Unknown," *Memoirs of the Faculty of Science, Kyushu University, Series A*, 42, 37–46.

Satterthwaite, F. W. (1946), "An Approximate Distribution of Estimates of Variance Components," *Biometrics Bulletin*, 2, 110–114.

Scheffe, H. (1970), "Practical Solutions of the Behrens-Fisher Problem," *Journal of the American Statistical Association*, 65, 1501–1508.

Schuirmann, D. J. (1987), "A Comparison of the Two One-Sided Tests Procedure and the Power Approach for Assessing the Equivalence of Average Bioavailability," *Journal of Pharmacokinetics and Biopharmaceutics*, 15, 657–680.

Senn, S. (2002), *Cross-over Trials in Clinical Research*, Second Edition, New York: John Wiley & Sons.

Steel, R. G. D. and Torrie, J. H. (1980), *Principles and Procedures of Statistics*, Second Edition, New York: McGraw-Hill.

Tamhane, A. C. and Logan, B. R. (2004), "Finding the Maximum Safe Dose Level for Heteroscedastic Data," *Journal of Biopharmaceutical Statistics*, 14, 843–856.

Wang, Y. Y. (1971), "Probabilities of the Type I Error of the Welch Tests for the Behren's-Fisher Problem," *Journal of the American Statistical Association*, 66, 605–608.

Wellek, S. (2003), *Testing Statistical Hypotheses of Equivalence*, Boca Raton, FL: Chapman & Hall/CRC Press LLC.

Yuen, K. K. (1974), "The Two-Sample Trimmed t for Unequal Population Variances," *Biometrika*, 61, 165–170.

Chapter 93

The VARCLUS Procedure

Contents

Overview: VARCLUS Procedure **7461**
Getting Started: VARCLUS Procedure **7463**
Syntax: VARCLUS Procedure **7468**
PROC VARCLUS Statement 7468
BY Statement 7474
FREQ Statement 7474
PARTIAL Statement 7475
SEED Statement 7475
VAR Statement 7475
WEIGHT Statement 7475
Details: VARCLUS Procedure **7476**
Missing Values 7476
Using the VARCLUS procedure 7476
Output Data Sets 7477
Computational Resources 7479
Interpreting VARCLUS Procedure Output 7479
Displayed Output 7480
ODS Table Names 7481
Example: VARCLUS Procedure **7483**
Example 93.1: Correlations among Physical Variables 7483
References **7492**

Overview: VARCLUS Procedure

The VARCLUS procedure divides a set of numeric variables into disjoint or hierarchical clusters. Associated with each cluster is a linear combination of the variables in the cluster. This linear combination can be either the first principal component (the default) or the centroid component (if you specify the CENTROID option). The first principal component is a weighted average of the variables that explains as much variance as possible. See Chapter 69, "The PRINCOMP Procedure," for further details. Centroid components are unweighted averages of either the standardized variables (the default) or the raw variables (if you specify the COVARIANCE option). The VARCLUS

procedure tries to maximize the variance that is explained by the cluster components, summed over all the clusters.

The cluster components are oblique, not orthogonal, even when the cluster components are first principal components. In an ordinary principal component analysis, all components are computed from the same variables, and the first principal component is orthogonal to the second principal component and to every other principal component. In the VARCLUS procedure, each cluster component is computed from a different set of variables than all the other cluster components. The first principal component of one cluster might be correlated with the first principal component of another cluster. Hence, the VARCLUS algorithm is a type of oblique component analysis.

As in principal component analysis, either the correlation or the covariance matrix can be analyzed. If correlations are used, all variables are treated as equally important. If covariances are used, variables with larger variances have more importance in the analysis.

The VARCLUS procedure creates an output data set that can be used with the SCORE procedure to compute component scores for each cluster. A second output data set can be used by the TREE procedure to draw a tree diagram of hierarchical clusters.

The VARCLUS procedure can be used as a variable-reduction method. A large set of variables can often be replaced by the set of cluster components with little loss of information. A given number of cluster components does not generally explain as much variance as the same number of principal components on the full set of variables, but the cluster components are usually easier to interpret than the principal components, even if the latter are rotated.

For example, an educational test might contain 50 items. The VARCLUS procedure can be used to divide the items into, say, five clusters. Each cluster can then be treated as a subtest, with the subtest scores given by the cluster components. If the cluster components are centroid components of the covariance matrix, each subtest score is simply the sum of the item scores for that cluster.

The VARCLUS algorithm is both divisive and iterative. By default, the VARCLUS procedure begins with all variables in a single cluster. It then repeats the following steps:

1. A cluster is chosen for splitting. Depending on the options specified, the selected cluster has either the smallest percentage of variation explained by its cluster component (using the PROPORTION= option) or the largest eigenvalue associated with the second principal component (using the MAXEIGEN= option).

2. The chosen cluster is split into two clusters by finding the first two principal components, performing an orthoblique rotation (raw quartimax rotation on the eigenvectors; Harris and Kaiser 1964), and assigning each variable to the rotated component with which it has the higher squared correlation.

3. Variables are iteratively reassigned to clusters to try to maximize the variance accounted for by the cluster components. You can require the reassignment algorithms to maintain a hierarchical structure for the clusters.

The procedure stops splitting when either of the following conditions holds:

- The number of clusters is greater than or equal to the maximum number of clusters as specified by the MAXCLUSTERS= option is reached.
- Every cluster satisfies the stopping criteria specified by the PROPORTION= (percentage of variation explained) and/or the MAXEIGEN= (second eigenvalue) options.

By default, VARCLUS stops splitting when every cluster has only one eigenvalue greater than one, thus satisfying the most popular criterion for determining the sufficiency of a single underlying dimension.

The iterative reassignment of variables to clusters proceeds in two phases. The first is a nearest component sorting (NCS) phase, similar in principle to the nearest centroid sorting algorithms described by Anderberg (1973). In each iteration, the cluster components are computed, and each variable is assigned to the component with which it has the highest squared correlation. The second phase involves a search algorithm in which each variable is tested to see if assigning it to a different cluster increases the amount of variance explained. If a variable is reassigned during the search phase, the components of the two clusters involved are recomputed before the next variable is tested. The NCS phase is much faster than the search phase but is more likely to be trapped by a local optimum.

If principal components are used, the NCS phase is an alternating least-squares method and converges rapidly. The search phase can be very time-consuming for a large number of variables. But if the default initialization method is used, the search phase is rarely able to substantially improve the results of the NCS phase, so the search takes few iterations. If random initialization is used, the NCS phase might be trapped by a local optimum from which the search phase can escape.

If centroid components are used, the NCS phase is not an alternating least-squares method and might not increase the amount of variance explained; therefore it is limited, by default, to one iteration.

You can have VARCLUS do the clustering hierarchically by restricting the reassignment of variables such that the clusters maintain a tree structure. In this case, when a cluster is split, a variable in one of the two resulting clusters can be reassigned to the other cluster resulting from the split but not to a cluster that is not part of the original cluster (the one that is split).

Getting Started: VARCLUS Procedure

This example demonstrates how you can cluster variables using the VARCLUS procedure.

The following data are job ratings of police officers. The officers were rated by their supervisors on 13 job skills on a scale from 1 to 9. There is also an overall rating that is not used in this analysis. The following DATA step creates the SAS data set JobRat:

```
data JobRat;
   input
      (Communication_Skills
       Problem_Solving
       Learning_Ability
       Judgement_under_Pressure
       Observational_Skills
       Willingness_to_Confront_Problems
       Interest_in_People
       Interpersonal_Sensitivity
       Desire_for_Self_Improvement
       Appearance
       Dependability
       Physical_Ability
       Integrity
       Overall_Rating)
      (1.);
datalines;
26838853879867
74758876857667
56757863775875
67869777988997

... more lines ...

99997899799799
99899899899899
76656399567486
;
```

The following statements cluster the variables:

```
proc varclus data=JobRat maxclusters=3;
   var Communication_Skills--Integrity;
run;
```

The DATA= option specifies the SAS data set JobRat as input.

The MAXCLUSTERS=3 option specifies that no more than three clusters be computed. By default, PROC VARCLUS splits and optimizes clusters until all clusters have a second eigenvalue less than one. In this example, the default setting would produce only two clusters, but going to three clusters produces a more interesting result.

The VAR statement lists the numeric variables (Communication_Skills–Integrity) to be used in the analysis. The overall rating is omitted from the list of variables.

Although the VARCLUS procedure displays output for one cluster, two clusters, and three clusters, the following figures display only the final analysis for three clusters.

For each cluster, Figure 93.1 displays the number of variables in the cluster, the cluster variation, the total explained variation, and the proportion of the total variance explained by the variables in the cluster. The variance explained by the variables in a cluster is similar to the variance explained

by a factor in common factor analysis, but it includes contributions only from the variables in the cluster rather than from all variables.

The line labeled "Total variation explained" in Figure 93.1 gives the sum of the explained variation over all clusters. The final "Proportion" represents the total explained variation divided by the sum of cluster variation. This value, 0.6715, indicates that about 67% of the total variation in the data can be accounted for by the three cluster components.

Figure 93.1 Cluster Summary for Three Clusters from the VARCLUS Procedure

Oblique Principal Component Cluster Analysis

Cluster Summary for 3 Clusters

Cluster	Members	Cluster Variation	Variation Explained	Proportion Explained	Second Eigenvalue
1	6	6	3.771349	0.6286	0.7093
2	5	5	3.575933	0.7152	0.5035
3	2	2	1.382005	0.6910	0.6180

Total variation explained = 8.729286 Proportion = 0.6715

Figure 93.2 shows how the variables are clustered. Figure 93.2 also displays the R-square value of each variable with its own cluster and the R-square value with its nearest cluster. The R-square value for a variable with the nearest cluster should be low if the clusters are well separated. The last column displays the ratio of $(1 - R^2_{own})/(1 - R^2_{nearest})$ for each variable. Small values of this ratio indicate good clustering.

Figure 93.2 R-Square Values from the VARCLUS Procedure

3 Clusters		R-squared with		
Cluster	Variable	Own Cluster	Next Closest	1-R**2 Ratio
Cluster 1	Communication_Skills	0.6403	0.3599	0.5620
	Problem_Solving	0.5412	0.2895	0.6458
	Learning_Ability	0.6561	0.1692	0.4139
	Observational_Skills	0.6889	0.2584	0.4194
	Willingness_to_Confront_Problems	0.6480	0.3402	0.5335
	Desire_for_Self_Improvement	0.5968	0.3473	0.6177
Cluster 2	Judgement_under_Pressure	0.6263	0.3719	0.5950
	Interest_in_People	0.8122	0.1885	0.2314
	Interpersonal_Sensitivity	0.7566	0.1387	0.2826
	Dependability	0.6163	0.4419	0.6875
	Integrity	0.7645	0.2724	0.3237
Cluster 3	Appearance	0.6910	0.3047	0.4444
	Physical_Ability	0.6910	0.1871	0.3801

Figure 93.3 displays the standardized scoring coefficients that are used to compute the first principal component of each cluster. Since each variable is assigned to one and only one cluster, each row of the scoring coefficients contains only one nonzero value.

Figure 93.3 Standardized Scoring Coefficients from the VARCLUS Procedure

Standardized Scoring Coefficients

Cluster	1	2	3
Communication_Skills	0.212170	0.000000	0.000000
Problem_Solving	0.195058	0.000000	0.000000
Learning_Ability	0.214781	0.000000	0.000000
Judgement_under_Pressure	0.000000	0.221313	0.000000
Observational_Skills	0.220086	0.000000	0.000000
Willingness_to_Confront_Problems	0.213452	0.000000	0.000000
Interest_in_People	0.000000	0.252025	0.000000
Interpersonal_Sensitivity	0.000000	0.243245	0.000000
Desire_for_Self_Improvement	0.204848	0.000000	0.000000
Appearance	0.000000	0.000000	0.601493
Dependability	0.000000	0.219544	0.000000
Physical_Ability	0.000000	0.000000	0.601493
Integrity	0.000000	0.244507	0.000000

Figure 93.4 displays the cluster structure and the intercluster correlations. The structure table displays the correlation of each variable with each cluster component. The table of intercorrelations contains the correlations between the cluster components.

Figure 93.4 Cluster Correlations and Intercorrelations from the VARCLUS Procedure

Cluster Structure

Cluster	1	2	3
Communication_Skills	0.800169	0.599909	0.427341
Problem_Solving	0.735630	0.538017	0.425463
Learning_Ability	0.810014	0.411316	0.376333
Judgement_under_Pressure	0.609876	0.791401	0.345399
Observational_Skills	0.830021	0.407807	0.508305
Willingness_to_Confront_Problems	0.805002	0.362927	0.583265
Interest_in_People	0.434138	0.901225	0.387770
Interpersonal_Sensitivity	0.372371	0.869826	0.287658
Desire_for_Self_Improvement	0.772554	0.589334	0.494842
Appearance	0.552003	0.393759	0.831266
Dependability	0.664778	0.785073	0.574460
Physical_Ability	0.432590	0.416070	0.831266
Integrity	0.521876	0.874342	0.477885

Figure 93.4 *continued*

Inter-Cluster Correlations

Cluster	1	2	3
1	1.00000	0.60851	0.59223
2	0.60851	1.00000	0.48711
3	0.59223	0.48711	1.00000

The VARCLUS procedure next displays the summary table of statistics for the cluster history (Figure 93.5). The first three columns give the number of clusters, the total variation explained by clusters, and the proportion of variation explained by clusters, respectively.

As displayed in the first row of Figure 93.5, the variation explained by the first principal component of all the variables is 6.547402, and the proportion of variation explained is 0.5036.

When the number of clusters is two, the total variation explained is 7.967753, and the proportion of variation explained by the two clusters is 0.6129. The larger second eigenvalue of the clusters is 0.937902, so by default, the VARCLUS procedure would stop splitting clusters at this point. But because the MAXCLUSTERS=3 option was specified in this example, the VARCLUS procedure continues to the three-cluster solution.

When the number of clusters increases to three, the total variation explained is 8.729286, and the proportion of variation explained by the two clusters is 0.6715. The largest second eigenvalue of the clusters is 0.709323. The statistical improvement from increasing the number of clusters from two to three seems modest, but the interpretabilty of the three clusters argues for the three-cluster solution.

Figure 93.5 also displays the minimum proportion of variance explained by a cluster, the minimum R square for a variable, and the maximum $(1 - R^2)$ ratio for a variable. The last quantity is the maximum ratio of the value $1 - R^2$ for a variable's own cluster to the value $1 - R^2$ for its nearest cluster.

Figure 93.5 Final Cluster Summary Table from the VARCLUS Procedure

Number of Clusters	Total Variation Explained by Clusters	Proportion of Variation Explained by Clusters	Minimum Proportion Explained by a Cluster	Maximum Second Eigenvalue in a Cluster	Minimum R-squared for a Variable	Maximum 1-R**2 Ratio for a Variable
1	6.547402	0.5036	0.5036	1.772715	0.2995	
2	7.967753	0.6129	0.5475	0.937902	0.3123	0.8026
3	8.729286	0.6715	0.6286	0.709323	0.5412	0.6875

Syntax: VARCLUS Procedure

The following statements are available in PROC VARCLUS:

PROC VARCLUS < *options* > **;**
VAR *variables* **;**
SEED *variables* **;**
PARTIAL *variables* **;**
WEIGHT *variables* **;**
FREQ *variables* **;**
BY *variables* **;**

Usually you need only the VAR statement in addition to the PROC VARCLUS statement. The following sections give detailed syntax information about each of the statements, beginning with the PROC VARCLUS statement. The remaining statements are listed in alphabetical order.

PROC VARCLUS Statement

PROC VARCLUS < *options* > **;**

The PROC VARCLUS statement starts the VARCLUS procedure. By default, VARCLUS clusters the numeric variables in the most recently created SAS data set, starting with one cluster and splitting clusters until all clusters have at most one eigenvalue greater than one.

VARCLUS chooses a cluster to split based on two options: MAXEIGEN= and PROPORTION=.

1. If you specify *either* or *both* of these two options, then *only* the specified options affect the choice of the cluster to split.
2. If you specify *neither* of these options, the criterion for choice of cluster to split depends on the CENTROID option:
   a) If you specify CENTROID, VARCLUS splits the cluster with the smallest percentage of variation explained by its cluster component, as if you had specified the PROPORTION= option.
   b) If you do not specify CENTROID, VARCLUS splits the cluster with the largest eigenvalue associated with the second principal component, as if you had specified the MAXEIGEN= option.

The final number of clusters is controlled by three options: MAXCLUSTERS=, MAXEIGEN=, and PROPORTION=.

1. If you specify *any* of these three options, then *only* the options you specify affect the final number of clusters.
2. If you specify *none* of these options, VARCLUS continues to split clusters until the default splitting criterion is satisfied. The default splitting criterion depends on the CENTROID option:

a) If you specify CENTROID, the default splitting criterion is PROPORTION=0.75.

b) If you do not specify CENTROID, splitting is based on the MAXEIGEN= criterion, with a default depending on the COVARIANCE option:

   i. For analyzing a correlation matrix (no COVARIANCE option), the default value for MAXEIGEN= is one.

   ii. For analyzing a covariance matrix (using the COVARIANCE option), the default value for MAXEIGEN= is the average variance of the variables being clustered.

VARCLUS continues to split clusters until any of the following conditions holds:

- The number of cluster equals the value specified for MAXCLUSTERS=.
- No cluster qualifies for splitting according to the MAXEIGEN= or PROPORTION= criterion.
- A cluster was chosen for splitting, but after iteratively reassigning variables to clusters, one of the cluster has no members.

Table 93.1 summarizes the options available in the PROC VARCLUS statement.

Table 93.1 Options Available in the PROC VARCLUS Statement

Option	Description
Data Sets	
DATA=	specifies the input SAS data set
OUTSTAT=	specifies the output SAS data set containing statistics
OUTTREE=	specifies the output SAS data set for use with PROC TREE
Input Data Processing	
COVARIANCE	uses the covariance matrix instead of the correlation matrix
NOINT	omits intercept
VARDEF=	specifies the divisor for variances
Number of Clusters	
MAXCLUSTERS=	specifies the maximum number of clusters
MINCLUSTERS=	specifies the minimum number of clusters
MAXEIGEN=	specifies the maximum second eigenvalue in a cluster
PROPORTION=	specifies the minimum proportion of variance explained by a cluster component
Clustering Methods	
CENTROID	uses centroid components instead of principal components
HIERARCHY	clusters hierarchically
INITIAL=	specifies the initialization method
MAXITER=	specifies the maximum iterations during the alternating least-squares phase
MAXSEARCH=	specifies the maximum iterations during the search phase
MULTIPLEGROUP	performs a multiple group component analysis
RANDOM=	specifies the random number seed
Control Displayed Output	
CORR	displays the correlation matrix
NOPRINT	suppresses displayed output

Table 93.1 *continued*

Option	Description
SHORT	suppresses display of large matrices
SIMPLE	displays means and standard deviations
SUMMARY	suppresses all default displayed output except the final summary table
TRACE	displays the cluster to which each variable is assigned during the iterations

The following list gives details on these options. The list is in alphabetical order.

CENTROID

uses centroid components rather than principal components. You should specify centroid components if you want the cluster components to be unweighted averages of the standardized variables (the default) or the unstandardized variables (if you specify the COVARIANCE option). It is possible to obtain locally optimal clusterings in which a variable is not assigned to the cluster component with which it has the highest squared correlation. You cannot specify both the CENTROID and MAXEIGEN= options.

CORR

C

displays the correlation matrix.

COVARIANCE

COV

analyzes the covariance matrix instead of the correlation matrix. The COVARIANCE option causes variables with a large variance to have more effect on the cluster components than variables with a small variance.

DATA=*SAS-data-set*

specifies the input data set to be analyzed. The data set can be an ordinary SAS data set or TYPE=CORR, UCORR, COV, UCOV, FACTOR, or SSCP. If you do not specify the DATA= option, the most recently created SAS data set is used. See Appendix A, "Special SAS Data Sets," for more information about types of SAS data sets.

HIERARCHY

HI

requires the clusters at different levels to maintain a hierarchical structure. To draw a tree diagram, use the OUTTREE= option and the TREE procedure.

INITIAL=GROUP

INITIAL=INPUT

INITIAL=RANDOM

INITIAL=SEED

specifies the method for initializing the clusters. If the INITIAL= option is omitted and the MINCLUSTERS= option is greater than 1, the initial cluster components are obtained by extracting the required number of principal components and performing an orthoblique rotation (raw quartimax rotation on the eigenvectors; Harris and Kaiser 1964). The following list describes the values for the INITIAL= option:

GROUP obtains the cluster membership of each variable from an observation in the DATA= data set where the _TYPE_ variable has a value of "GROUP". In this observation, the variables to be clustered must each have an integer value ranging from one to the number of clusters. You can use this option only if the DATA= data set is a TYPE=CORR, UCORR, COV, UCOV, or FACTOR data set. You can use a data set created either by a previous run of PROC VARCLUS or in a DATA step.

INPUT obtains scoring coefficients for the cluster components from observations in the DATA= data set where the _TYPE_ variable has a value of "SCORE". You can use this option only if the DATA= data set is a TYPE=CORR, UCORR, COV, UCOV, or FACTOR data set, You can use scoring coefficients from the FACTOR procedure or a previous run of the VARCLUS procedure, or you can enter other coefficients in a DATA step.

RANDOM assigns variables randomly to clusters.

SEED initializes each cluster component to be one of the variables named in the SEED statement. Each variable listed in the SEED statement becomes the sole member of a cluster, and the other variables are initially unassigned. If you do not specify the SEED statement, the first MINCLUSTERS= variables in the VAR statement are used as seeds.

MAXCLUSTERS=*n*

MAXC=*n*

specifies the largest number of clusters desired. The default value is the number of variables. VARCLUS stops splitting clusters after the number of clusters reaches the value of the MAXCLUSTERS= option, regardless of what other splitting options are specified.

MAXEIGEN=*n*

specifies that when choosing a cluster to split, VARCLUS should choose the cluster with the largest second eigenvalue, provided that its second eigenvalue is greater than the MAXEIGEN= value. The MAXEIGEN= option cannot be used with the CENTROID or MULTIPLEGROUP options.

If you do not specify MAXEIGEN=, the default behavior depends on other options as follows:

- If you specify PROPORTION=, CENTROID, or MULTIPLEGROUP, cluster splitting does not depend on the second eigenvalue.
- Otherwise, if you specify MAXCLUSTERS=, the default value for MAXEIGEN= is zero.
- Otherwise, the default value for MAXEIGEN= is either 1.0 if the correlation matrix is analyzed, or the average variance if the COVARIANCE option is specified.

If you specify both MAXEIGEN= and MAXCLUSTERS=, the number of clusters will never exceed the value of the MAXCLUSTERS= option.

If you specify both MAXEIGEN= and PROPORTION=, VARCLUS first looks for a cluster to split based on the MAXEIGEN= criterion. If no cluster meets that criterion, VARCLUS then looks for a cluster to split based on the PROPORTION= criterion.

MAXITER=*n*

specifies the maximum number of iterations during the NCS phase. The default value is 1 if you specify the CENTROID option; the default is 10 otherwise.

MAXSEARCH=*n*

specifies the maximum number of iterations during the search phase. The default is 1000 divide by the number of variables.

MINCLUSTERS=*n*

MINC=*n*

specifies the smallest number of clusters desired. The default value is 2 for INITIAL=RANDOM or INITIAL=SEED; otherwise, VARCLUS begins with one cluster and tries to split it in accordance with the PROPORTION= and/or MAXEIGEN= options.

MULTIPLEGROUP

MG

performs a multiple group component analysis (Harman 1976). You specify which variables belong to which clusters. No clusters are split, and no variables are reassigned to a different cluster. The input data set must be TYPE=CORR, UCORR, COV, UCOV, FACTOR, or SSCP and must contain an observation with _TYPE_="GROUP" defining the variable groups. Specifying the MULTIPLEGROUP option is equivalent to specifying all of the following options: INITIAL=GROUP, MINC=1, MAXITER=0, MAXSEARCH=0, PROPORTION=0, and MAXEIGEN=large number.

NOINT

requests that no intercept be used; covariances or correlations are not corrected for the mean. If you specify the NOINT option, the OUTSTAT= data set is TYPE=UCORR.

NOPRINT

suppresses displayed output. Note that this option temporarily disables the Output Delivery System (ODS). For more information, see Chapter 20, "Using the Output Delivery System."

OUTSTAT=*SAS-data-set*

creates an output data set to contain statistics including means, standard deviations, correlations, cluster scoring coefficients, and the cluster structure. If you want to create a permanent SAS data set, you must specify a two-level name. The OUTSTAT= data set is TYPE=UCORR if the NOINT option is specified. For more information about permanent SAS data sets, see "SAS Files" and "DATA Step Concepts" in *SAS Language Reference: Concepts*. For information about types of SAS data sets, see Appendix A, "Special SAS Data Sets."

OUTTREE=*SAS-data-set*

creates an output data set to contain information on the tree structure that can be used by the TREE procedure to display a tree diagram. The OUTTREE= option implies the HIERARCHY option. See Example 93.1 for use of the OUTTREE= option. If you want to create a permanent SAS data set, you must specify a two-level name. For more information on permanent SAS data sets, see "SAS Files" and "DATA Step Concepts" in *SAS Language Reference: Concepts*.

PROPORTION=*n*

PERCENT=*n*

specifies that when choosing a cluster to split, VARCLUS should choose the cluster with the smallest proportion of variation explained, provided that its proportion of variation explained is less than the PROPORTION= value. Values greater than 1.0 are considered to be percentages, so PROPORTION=0.75 and PERCENT=75 are equivalent.

However, if you specify both MAXEIGEN= and PROPORTION=, VARCLUS first looks for a cluster to split based on the MAXEIGEN= criterion. If no cluster meets that criterion, VARCLUS then looks for a cluster to split based on the PROPORTION= criterion.

If you do not specify PROPORTION=, the default behavior depends on other options as follows:

- If you specify MAXEIGEN=, cluster splitting does not depend on the proportion of variation explained.
- Otherwise, if you specify CENTROID and MAXCLUSTERS=, the default value for PROPORTION= is one.
- Otherwise, if you specify CENTROID, without MAXCLUSTERS=, the default value is PROPORTION=0.75 or PERCENT=75.
- Otherwise, cluster splitting does not depend on the proportion of variation explained.

If you specify both PROPORTION= and MAXCLUSTERS=, the number of clusters will never exceed the value of the MAXCLUSTERS= option.

RANDOM=*n*

specifies a positive integer as a starting value for use with REPLACE=RANDOM. If you do not specify the RANDOM= option, the time of day is used to initialize the pseudo-random number sequence.

SHORT

suppresses display of the cluster structure, scoring coefficient, and intercluster correlation matrices.

SIMPLE

S

displays means and standard deviations.

SUMMARY

suppresses all default displayed output except the final summary table.

TRACE

displays the cluster to which each variable is assigned during the iterations.

VARDEF=DF

VARDEF=N

VARDEF=WDF

VARDEF=WEIGHT | WGT

specifies the divisor to be used in the calculation of variances and covariances. The default value is VARDEF=DF. The values and associated divisors are displayed in the following table.

Value	Divisor	Formula
DF	degrees of freedom	$n - i$
N	number of observations	n
WDF	sum of weights minus one	$(\sum_j w_j) - 1$
WEIGHT | WGT	sum of weights	$\sum_j w_j$

In the preceding table, $i = 0$ if the NOINT option is specified, and $i = 1$ otherwise.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC VARCLUS to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for PROC VARCLUS. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure.

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

FREQ Statement

FREQ *variable* **;**

If a variable in your data set represents the frequency of occurrence for the other values in the observation, include the variable's name in a FREQ statement. The procedure then treats the data set

as if each observation appears *n* times, where *n* is the value of the FREQ variable for the observation. If the value of the FREQ variable is less than 1, the observation is not used in the analysis. Only the integer portion of the value is used. The total number of observations is considered equal to the sum of the FREQ variable.

PARTIAL Statement

PARTIAL *variable* **;**

If you want to base the clustering on partial correlations, list the variables to be partialed out in the PARTIAL statement.

SEED Statement

SEED *variables* **;**

The SEED statement specifies variables to be used as seeds to initialize the clusters. It is not necessary to use INITIAL=SEED if the SEED statement is present, but if any other INITIAL= option is specified, the SEED statement is ignored.

VAR Statement

VAR *variables* **;**

The VAR statement specifies the variables to be clustered. If you do not specify the VAR statement and do not specify TYPE=SSCP, all numeric variables not listed in other statements (except the SEED statement) are processed. The default VAR variable list does not include the variable INTERCEPT if the DATA= data set is TYPE=SSCP. If the variable INTERCEPT is explicitly specified in the VAR statement with a TYPE=SSCP data set, the NOINT option is enabled.

WEIGHT Statement

WEIGHT *variables* **;**

If you want to specify relative weights for each observation in the input data set, place the weights in a variable in the data set and specify the name in a WEIGHT statement. This is often done when the variance associated with each observation is different and the values of the weight variable are proportional to the reciprocals of the variances. The WEIGHT variable can take nonintegral values. An observation is used in the analysis only if the value of the WEIGHT variable is greater than zero.

Details: VARCLUS Procedure

Missing Values

Observations containing missing values are omitted from the analysis.

Using the VARCLUS procedure

Default options for the VARCLUS procedure often provide satisfactory results. If you want to change the final number of clusters, use one or more of the MAXCLUSTERS=, MAXEIGEN=, or PROPORTION= options. The MAXEIGEN= and PROPORTION= options usually produce similar results but occasionally cause different clusters to be selected for splitting. The MAXEIGEN= option tends to choose clusters with a large number of variables, while the PROPORTION= option is more likely to select a cluster with a small number of variables.

Execution time

The VARCLUS procedure usually requires more computer time than principal factor analysis, but it can be faster than some of the iterative factoring methods. If you have more than 30 variables, you might want to reduce execution time by one or more of the following methods:

- Specify the MINCLUSTERS= and MAXCLUSTERS= options if you know how many clusters you want.
- Specify the HIERARCHY option.
- Specify the SEED statement if you have some prior knowledge of what clusters to expect.

If computer time is not a limiting factor, you might want to try one of the following methods to obtain a better solution:

- If the clustering algorithm has not converged, specify larger values for MAXITER= and MAXSEARCH=.
- Try several factoring and rotation methods with PROC FACTOR to use as input to PROC VARCLUS.
- Run PROC VARCLUS several times, specifying INITIAL=RANDOM.

Output Data Sets

OUTSTAT= Data Set

The OUTSTAT= data set is TYPE=CORR, and it can be used as input to the SCORE procedure or a subsequent run of PROC VARCLUS. The OUSTAT= data set contains the following variables:

- BY variables
- _NCL_, a numeric variable giving the number of clusters
- _TYPE_, a character variable indicating the type of statistic the observation contains
- _NAME_, a character variable containing a variable name or a cluster name, which is of the form CLUS*n*, where *n* is the number of the cluster
- the variables that are clustered

The values of the _TYPE_ variable are listed in the following table.

Table 93.2 _TYPE_

TYPE	**Contents**
MEAN	means
STD	standard deviations
USTD	uncorrected standard deviations, produced when the NOINT option is specified
N	number of observations
CORR	correlations
UCORR	uncorrected correlation matrix, produced when the NOINT option is specified
MEMBERS	number of members in each cluster
VAREXP	variance explained by each cluster
PROPOR	proportion of variance explained by each cluster
GROUP	number of the cluster to which each variable belongs
RSQUARED	squared multiple correlation of each variable with its cluster component
SCORE	standardized scoring coefficients
USCORE	scoring coefficients to be applied without subtracting the mean from the raw variables, produced when the NOINT option is specified
STRUCTUR	cluster structure
CCORR	correlations between cluster components

The observations with _TYPE_="MEAN", "STD", "N", and "CORR" have missing values for the _NCL_ variable. All other values of the _TYPE_ variable are repeated for each cluster solution, with different solutions distinguished by the value of the _NCL_ variable. If you want to specify the OUTSTAT= data set with the SCORE procedure, you can use a DATA step to select observations with the _NCL_ variable missing or equal to the desired number of clusters as follows:

```
data Coef2;
   set Coef;
   if _ncl_ = . or _ncl_ = 3;
   drop _ncl_;
run;

proc score data=NewScore score=Coef2; run;
```

PROC SCORE standardizes the new data by subtracting the original variable means that are stored in the _TYPE_='MEAN' observations, and dividing by the original variable standard deviations from the _TYPE_='STD' observations. Then PROC SCORE multiplies the standardized variables by the coefficients from the _TYPE_='SCORE' observations to get the cluster scores.

OUTTREE= Data Set

The OUTTREE= data set contains one observation for each variable clustered plus one observation for each cluster of two or more variables—that is, one observation for each node of the cluster tree. The total number of output observations is between n and $2n-1$, where n is the number of variables clustered.

The OUTTREE= data set contains the following variables:

- BY variables, if any
- _NAME_, a character variable giving the name of the node. If the node is a cluster, the name is CLUSn, where n is the number of the cluster. If the node is a single variable, the variable name is used.
- _PARENT_, a character variable giving the value of _NAME_ of the parent of the node. If the node is the root of the tree, _PARENT_ is blank.
- _LABEL_, a character variable giving the label of the node. If the node is a cluster, the label is CLUSn, where n is the number of the cluster. If the node is a single variable, the variable label is used.
- _NCL_, the number of clusters
- _VAREXP_, the total variance explained by the clusters at the current level of the tree
- _PROPOR_, the total proportion of variance explained by the clusters at the current level of the tree
- _MINPRO_, the minimum proportion of variance explained by a cluster component
- _MAXEIG_, the maximum second eigenvalue of a cluster

Computational Resources

Let

n = number of observations
v = number of variables
c = number of clusters

It is assumed that, at each stage of clustering, the clusters all contain the same number of variables.

Time

The time required for the VARCLUS procedure to analyze a given data set varies greatly depending on the number of clusters requested, the number of iterations in both the alternating least-squares and search phases, and whether centroid or principal components are used.

The time required to compute the correlation matrix is roughly proportional to nv^2.

Default cluster initialization requires time roughly proportional to v^3. Any other method of initialization requires time roughly proportional to cv^2.

In the alternating least-squares phase, each iteration requires time roughly proportional to cv^2 if centroid components are used or

$$\left(c + 5\frac{v}{c^2}\right) v^2$$

if principal components are used.

In the search phase, each iteration requires time roughly proportional to v^3/c if centroid components are used or v^4/c^2 if principal components are used. The HIERARCHY option speeds up each iteration after the first split by as much as $c/2$.

Memory

The amount of memory, in bytes, needed by the VARCLUS procedure is approximately

$$v^2 + 2vc + 20v + 15c$$

Interpreting VARCLUS Procedure Output

Because the VARCLUS algorithm is a type of oblique component analysis, its output is similar to the output from the FACTOR procedure for oblique rotations. The scoring coefficients have the same meaning in both PROC VARCLUS and PROC FACTOR; they are coefficients applied

to the standardized variables to compute component scores. The cluster structure is analogous to the factor structure containing the correlations between each variable and each cluster component. A cluster pattern is not displayed because it would be the same as the cluster structure, except that zeros would appear in the same places in which zeros appear in the scoring coefficients. The intercluster correlations are analogous to interfactor correlations; they are the correlations among cluster components.

The VARCLUS procedure also displays a cluster summary and a cluster listing. The cluster summary gives the number of variables in each cluster and the variation explained by the cluster component. The latter is similar to the variation explained by a factor but includes contributions from only the variables in that cluster rather than from all variables, as in PROC FACTOR. The proportion of variance explained is obtained by dividing the variance explained by the total variance of variables in the cluster. If the cluster contains two or more variables and the CENTROID option is not used, the second largest eigenvalue of the cluster is also displayed.

The cluster listing gives the variables in each cluster. Two squared correlations are calculated for each cluster. The column labeled "Own Cluster" gives the squared correlation of the variable with its own cluster component. This value should be higher than the squared correlation with any other cluster unless an iteration limit has been exceeded or the CENTROID option has been used. The larger the squared correlation is, the better. The column labeled "Next Closest" contains the next-highest squared correlation of the variable with a cluster component. This value is low if the clusters are well separated. The column labeled "1–R**2 Ratio" gives the ratio of one minus the "Own Cluster" R square to one minus the "Next Closest" R square. A small "1–R**2 Ratio" indicates a good clustering.

Displayed Output

The following items are displayed for each cluster solution unless the NOPRINT or SUMMARY option is specified. The CLUSTER SUMMARY table includes the following columns:

- the Cluster number
- Members, the number of members in the cluster
- Cluster Variation of the variables in the cluster
- Variation Explained by the cluster component. This statistic is based only on the variables in the cluster rather than on all variables.
- Proportion Explained, the result of dividing the variation explained by the cluster variation
- Second Eigenvalue, the second largest eigenvalue of the cluster. This is displayed if the cluster contains more than one variable and the CENTROID option is not specified

The VARCLUS procedure also displays the following:

- Total variation explained, the sum across clusters of the variation explained by each cluster

- Proportion, the total explained variation divided by the total variation of all the variables

The cluster listing includes the following columns:

- Variable, the variables in each cluster
- R square with Own Cluster, the squared correlation of the variable with its own cluster component; and R square with Next Closest, the next highest squared correlation of the variable with a cluster component. Own Cluster values should be higher than the R square with any other cluster unless an iteration limit is exceeded or you specify the CENTROID option. Next Closest should be a low value if the clusters are well separated.
- 1−R**2 Ratio, the ratio of one minus the value in the Own Cluster column to one minus the value in the Next Closest column. The occurrence of low ratios indicates well-separated clusters.

If the SHORT option is not specified, the VARCLUS procedure also displays the following tables:

- Standardized Scoring Coefficients, standardized regression coefficients for predicting cluster components from variables
- Cluster Structure, the correlations between each variable and each cluster component
- Inter-Cluster Correlations, the correlations between the cluster components

If the analysis includes partitions for two or more numbers of clusters, a final summary table is displayed. Each row of the table corresponds to one partition. The columns include the following:

- Number of Clusters
- Total Variation Explained by Clusters
- Proportion of Variation Explained by Clusters
- Minimum Proportion (of variation) Explained by a Cluster
- Maximum Second Eigenvalue in a Cluster
- Minimum R square for a Variable
- Maximum 1−R**2 Ratio for a Variable

ODS Table Names

The VARCLUS procedure assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These ODS table names are listed in Table 93.3. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 93.3 ODS Tables Produced by the VARCLUS procedure

ODS Table Name	Description	Option
ClusterQuality	Cluster quality	default
ClusterStructure	Cluster structure	default
ClusterSummary	Cluster summary	default
ConvergenceStatus	Convergence status	default
Corr	Correlations between variables	CORR
DataOptSummary	Data and options summary table	default
InterClusterCorr	Correlations between cluster components	default
IterHistory	Iteration history	TRACE
RSquare	R-squares between variables and clusters	default
SimpleStatistics	Means and standard deviations	SIMPLE
StdScoreCoef	Standardized scoring coefficients	default

Example: VARCLUS Procedure

Example 93.1: Correlations among Physical Variables

The data in this example are correlations among eight physical variables as given by Harman (1976). The first PROC VARCLUS run clusters on the basis of principal components. The second run clusters on the basis of centroid components. The third analysis is hierarchical, and the TREE procedure is used to display a tree diagram. The following statements create the data set and perform the analysis:

```
data phys8(type=corr);
   title 'Eight Physical Measurements on 305 School Girls';
   title2 'Harman: Modern Factor Analysis, 3rd Ed, p22';
   label ArmSpan='Arm Span'              Forearm='Length of Forearm'
         LowerLeg='Length of Lower Leg' BitDiam='Bitrochanteric Diameter'
         Girth='Chest Girth'            Width='Chest Width';
   input _Name_ $ 1-8
         (Height ArmSpan Forearm LowerLeg Weight BitDiam
          Girth Width)(7.);
   _Type_='corr';
   datalines;
Height     1.0    .846   .805   .859   .473   .398   .301   .382
ArmSpan    .846   1.0    .881   .826   .376   .326   .277   .415
Forearm    .805   .881   1.0    .801   .380   .319   .237   .345
LowerLeg   .859   .826   .801   1.0    .436   .329   .327   .365
Weight     .473   .376   .380   .436   1.0    .762   .730   .629
BitDiam    .398   .326   .319   .329   .762   1.0    .583   .577
Girth      .301   .277   .237   .327   .730   .583   1.0    .539
Width      .382   .415   .345   .365   .629   .577   .539   1.0
;

proc varclus data=phys8;
run;
```

The PROC VARCLUS statement invokes the procedure. By default, the VARCLUS procedure clusters using principal components.

As displayed in Output 93.1.1, when there is only one cluster, the cluster component (by default, the first principal component) explains 58.41% of the total variation of the eight variables.

The cluster is split because the second eigenvalue is greater than 1 (the default value of the MAXEIGEN option).

The two resulting cluster components explain 80.33% of the variation in the original variables. The cluster summary table shows that the variables Height, ArmSpan, Forearm, and LowerLeg have been assigned to the first cluster, and that the variables Weight, BitDiam, Girth, and Width have been assigned to the second cluster.

The standardized scoring coefficients in Output 93.1.1 show that each cluster component has similar scores for each of its associated variables. This suggests that the principal cluster component solution should be similar to the centroid cluster component solution, which follows in the next PROC VARCLUS run.

The cluster structure table displays high correlations between the variables and their own cluster component. The correlations between the variables and the opposite cluster component are all moderate.

The intercluster correlation table shows that the two cluster components have a moderate correlation of 0.44513.

Output 93.1.1 Principal Component Clusters

```
          Eight Physical Measurements on 305 School Girls
            Harman: Modern Factor Analysis, 3rd Ed, p22

            Oblique Principal Component Cluster Analysis

        Observations          10000    Proportion          0
        Variables                 8    Maxeigen            1


  Clustering algorithm converged.

                     Cluster Summary for 1 Cluster

                       Cluster    Variation    Proportion        Second
  Cluster    Members    Variation    Explained     Explained    Eigenvalue
  ------------------------------------------------------------------------
        1          8            8      4.67288        0.5841        1.7710

          Total variation explained = 4.67288 Proportion = 0.5841

Cluster 1 will be split because it has the largest second eigenvalue, 1.770983,
which is greater than the MAXEIGEN=1 value.


  Clustering algorithm converged.

                     Cluster Summary for 2 Clusters

                       Cluster    Variation    Proportion        Second
  Cluster    Members    Variation    Explained     Explained    Eigenvalue
  ------------------------------------------------------------------------
        1          4            4      3.509218        0.8773        0.2361
        2          4            4      2.917284        0.7293        0.4764

          Total variation explained = 6.426502 Proportion = 0.8033
```

Output 93.1.1 *continued*

2 Clusters	Variable	R-squared with Own Cluster	Next Closest	1-R**2 Ratio	Variable Label
Cluster 1	ArmSpan	0.9002	0.1658	0.1196	Arm Span
	Forearm	0.8661	0.1413	0.1560	Length of Forearm
	LowerLeg	0.8652	0.1829	0.1650	Length of Lower Leg
	Height	0.8777	0.2088	0.1545	
Cluster 2	BitDiam	0.7386	0.1341	0.3019	Bitrochanteric Diameter
	Girth	0.6981	0.0929	0.3328	Chest Girth
	Width	0.6329	0.1619	0.4380	Chest Width
	Weight	0.8477	0.1974	0.1898	

Standardized Scoring Coefficients

Cluster		1	2
ArmSpan	Arm Span	0.270377	0.000000
Forearm	Length of Forearm	0.265194	0.000000
LowerLeg	Length of Lower Leg	0.265057	0.000000
BitDiam	Bitrochanteric Diameter	0.000000	0.294591
Girth	Chest Girth	0.000000	0.286407
Width	Chest Width	0.000000	0.272710
Height		0.266977	0.000000
Weight		0.000000	0.315597

Cluster Structure

Cluster		1	2
ArmSpan	Arm Span	0.948813	0.407210
Forearm	Length of Forearm	0.930624	0.375865
LowerLeg	Length of Lower Leg	0.930142	0.427715
BitDiam	Bitrochanteric Diameter	0.366201	0.859404
Girth	Chest Girth	0.304779	0.835529
Width	Chest Width	0.402430	0.795572
Height		0.936881	0.456908
Weight		0.444281	0.920686

Inter-Cluster Correlations

Cluster	1	2
1	1.00000	0.44513
2	0.44513	1.00000

Output 93.1.1 *continued*

```
No cluster meets the criterion for splitting.


                Total     Proportion        Minimum        Maximum     Minimum
  Number     Variation   of Variation     Proportion         Second   R-squared
      of     Explained      Explained      Explained     Eigenvalue       for a
Clusters   by Clusters    by Clusters   by a Cluster   in a Cluster    Variable
-------------------------------------------------------------------------------
       1      4.672880         0.5841         0.5841       1.770983      0.3810
       2      6.426502         0.8033         0.7293       0.476418      0.6329

                                      Maximum
                           Number   1-R**2 Ratio
                               of          for a
                         Clusters       Variable
                         ----------------------
                                1
                                2         0.4380
```

In the following statements, the CENTROID option in the PROC VARCLUS statement specifies that cluster centroids be used as the basis for clustering:

```
proc varclus data=phys8 centroid;
run;
```

The first cluster component, which, in the centroid method, is an unweighted sum of the standardized variables, explains 57.89% of the variation in the data. This value is near the maximum possible variance explained, 58.41%, which is attained by the first principal component shown previously in Output 93.1.1.

The default behavior in the centroid method is to split any cluster with less than 75% of the total cluster variance explained by the centroid component. Since the centroid component for the one-cluster solution explains only 57.89% of the variation as shown in Output 93.1.2, the variables are split into two clusters. The resulting clusters are the same two clusters created by the principal component method. Recall that this outcome was suggested by the similar standardized scoring coefficients in the principal cluster component solution.

In the two-cluster solution, the centroid component of the second cluster explains only 72.75% of the total variation of the cluster. Since this percentage is less than 75%, the second cluster is split.

In the R-square table for two clusters, the Width variable has a weaker relation to its cluster than any other variable. In the three-cluster solution this variable is in a cluster of its own.

Each cluster component is an unweighted average of the cluster's standardized variables. Thus, the coefficients for each of the cluster's associated variables are identical in the centroid cluster component solution.

The centroid method stops at the three-cluster solution. The three centroid components account for 86.15% of the variability in the eight variables, and all cluster components account for at least 79.44% of the total variation in the corresponding cluster. Additionally, the smallest squared corre-

lation between the variables and their own cluster component is 0.7482.

Note that if the PROPORTION= option were set to a value between 0.5789 (the proportion of variance explained in the one-cluster solution) and 0.7275 (the minimum proportion of variance explained in the two-cluster solution), the VARCLUS procedure would stop at the two-cluster solution, and the centroid solution would find the same clusters as the principal components solution, although the cluster components would be slightly different.

Output 93.1.2 Centroid Component Clusters

```
          Eight Physical Measurements on 305 School Girls
            Harman: Modern Factor Analysis, 3rd Ed, p22

            Oblique Centroid Component Cluster Analysis

        Observations          10000    Proportion       0.75
        Variables                 8    Maxeigen            0

  Clustering algorithm converged.

                      Cluster Summary for 1 Cluster

                              Cluster    Variation    Proportion
        Cluster    Members    Variation    Explained     Explained
        ------------------------------------------------------------
              1          8            8        4.631        0.5789

          Total variation explained = 4.631 Proportion = 0.5789

Cluster 1 will be split because it has the smallest proportion of variation
explained, 0.578875, which is less than the PROPORTION=0.75 value.


  Clustering algorithm converged.

                     Cluster Summary for 2 Clusters

                              Cluster    Variation    Proportion
        Cluster    Members    Variation    Explained     Explained
        ------------------------------------------------------------
              1          4            4        3.509        0.8773
              2          4            4         2.91        0.7275

          Total variation explained = 6.419 Proportion = 0.8024
```

Output 93.1.2 *continued*

2 Clusters		R-squared with Own Cluster	Next Closest	1-R**2 Ratio	Variable Label
Cluster 1	ArmSpan	0.8994	0.1669	0.1208	Arm Span
	Forearm	0.8663	0.1410	0.1557	Length of Forearm
	LowerLeg	0.8658	0.1824	0.1641	Length of Lower Leg
	Height	0.8778	0.2075	0.1543	
Cluster 2	BitDiam	0.7335	0.1341	0.3078	Bitrochanteric Diameter
	Girth	0.6988	0.0929	0.3321	Chest Girth
	Width	0.6473	0.1618	0.4207	Chest Width
	Weight	0.8368	0.1975	0.2033	

Standardized Scoring Coefficients

Cluster		1	2
ArmSpan	Arm Span	0.266918	0.000000
Forearm	Length of Forearm	0.266918	0.000000
LowerLeg	Length of Lower Leg	0.266918	0.000000
BitDiam	Bitrochanteric Diameter	0.000000	0.293105
Girth	Chest Girth	0.000000	0.293105
Width	Chest Width	0.000000	0.293105
Height		0.266918	0.000000
Weight		0.000000	0.293105

Cluster Structure

Cluster		1	2
ArmSpan	Arm Span	0.948361	0.408589
Forearm	Length of Forearm	0.930744	0.375468
LowerLeg	Length of Lower Leg	0.930477	0.427054
BitDiam	Bitrochanteric Diameter	0.366212	0.856453
Girth	Chest Girth	0.304821	0.835936
Width	Chest Width	0.402246	0.804574
Height		0.936883	0.455485
Weight		0.444419	0.914781

Inter-Cluster Correlations

Cluster	1	2
1	1.00000	0.44484
2	0.44484	1.00000

Cluster 2 will be split because it has the smallest proportion of variation explained, 0.7275, which is less than the PROPORTION=0.75 value.

Clustering algorithm converged.

Output 93.1.2 *continued*

Cluster Summary for 3 Clusters

Cluster	Members	Cluster Variation	Variation Explained	Proportion Explained
1	4	4	3.509	0.8773
2	3	3	2.383333	0.7944
3	1	1	1	1.0000

Total variation explained = 6.892333 Proportion = 0.8615

3 Clusters		R-squared with Own Cluster	Next Closest	1-R**2 Ratio	Variable Label
Cluster 1	ArmSpan	0.8994	0.1722	0.1215	Arm Span
	Forearm	0.8663	0.1225	0.1524	Length of Forearm
	LowerLeg	0.8658	0.1668	0.1611	Length of Lower Leg
	Height	0.8778	0.1921	0.1513	
Cluster 2	BitDiam	0.7691	0.3329	0.3461	Bitrochanteric Diameter
	Girth	0.7482	0.2905	0.3548	Chest Girth
	Weight	0.8685	0.3956	0.2175	
Cluster 3	Width	1.0000	0.4259	0.0000	Chest Width

Standardized Scoring Coefficients

Cluster		1	2	3
ArmSpan	Arm Span	0.26692	0.00000	0.00000
Forearm	Length of Forearm	0.26692	0.00000	0.00000
LowerLeg	Length of Lower Leg	0.26692	0.00000	0.00000
BitDiam	Bitrochanteric Diameter	0.00000	0.37398	0.00000
Girth	Chest Girth	0.00000	0.37398	0.00000
Width	Chest Width	0.00000	0.00000	1.00000
Height		0.26692	0.00000	0.00000
Weight		0.00000	0.37398	0.00000

Cluster Structure

Cluster		1	2	3
ArmSpan	Arm Span	0.94836	0.36613	0.41500
Forearm	Length of Forearm	0.93074	0.35004	0.34500
LowerLeg	Length of Lower Leg	0.93048	0.40838	0.36500
BitDiam	Bitrochanteric Diameter	0.36621	0.87698	0.57700
Girth	Chest Girth	0.30482	0.86501	0.53900
Width	Chest Width	0.40225	0.65259	1.00000
Height		0.93688	0.43830	0.38200
Weight		0.44442	0.93196	0.62900

Output 93.1.2 *continued*

Inter-Cluster Correlations

Cluster	1	2	3
1	1.00000	0.41716	0.40225
2	0.41716	1.00000	0.65259
3	0.40225	0.65259	1.00000

No cluster meets the criterion for splitting.

Number of Clusters	Total Variation Explained by Clusters	Proportion of Variation Explained by Clusters	Minimum Proportion Explained by a Cluster	Minimum R-squared for a Variable	Maximum 1-R**2 Ratio for a Variable
1	4.631000	0.5789	0.5789	0.4306	
2	6.419000	0.8024	0.7275	0.6473	0.4207
3	6.892333	0.8615	0.7944	0.7482	0.3548

In the following statements, the MAXC= option computes all clustering solutions, from one to eight clusters. The SUMMARY option suppresses all output except the final cluster quality table, and the OUTTREE= option saves the results of the analysis to an output data set and forces the clusters to be hierarchical. The TREE procedure is invoked to produce a graphical display of the clusters as follows:

```
proc varclus data=phys8 maxc=8 summary outtree=tree;
run;

title  h=10pct 'Eight Physical Measurements on 305 School Girls';
title2 h=5pct 'Harman: Modern Factor Analysis, 3rd Ed, p22';
goptions htext=4pct ftext="Albany AMT";
axis1 order=(0.5 to 1 by 0.1);
axis2 label=none;
proc tree horizontal haxis=axis1 vaxis=axis2;
   height _propor_;
   id _label_;
run;
```

The results from PROC VARCLUS are shown in Output 93.1.3.

Output 93.1.3 Hierarchical Clusters and the SUMMARY Option

Eight Physical Measurements on 305 School Girls
Harman: Modern Factor Analysis, 3rd Ed, p22

Oblique Principal Component Cluster Analysis

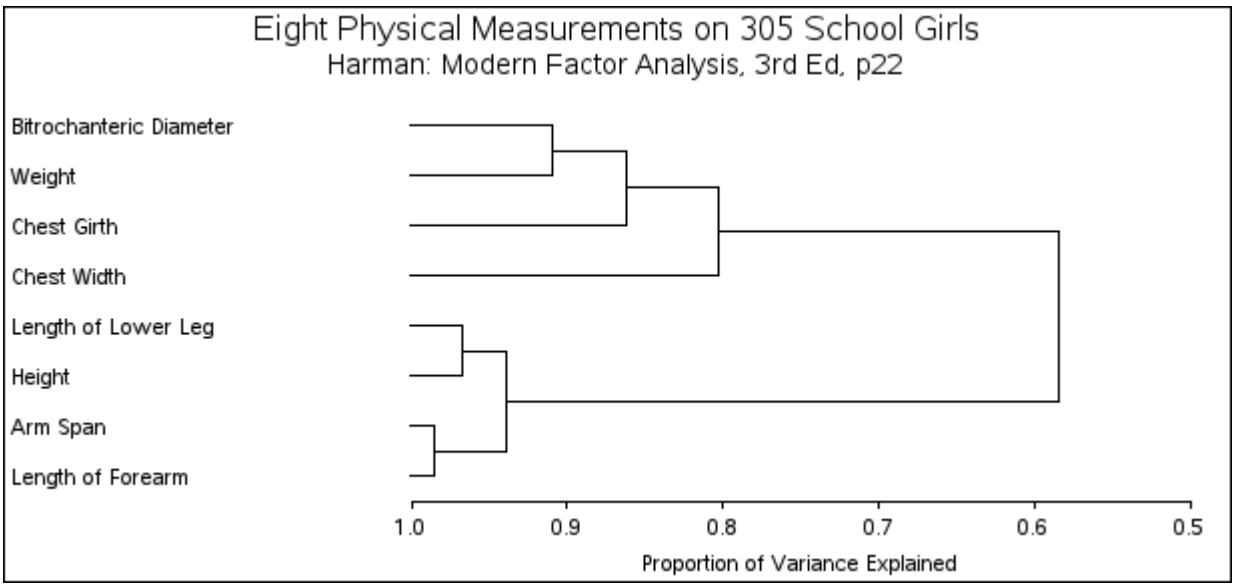
Observations	10000	Proportion	1
Variables	8	Maxeigen	0

Clustering algorithm converged.

Number of Clusters	Total Variation Explained by Clusters	Proportion of Variation Explained by Clusters	Minimum Proportion Explained by a Cluster	Maximum Second Eigenvalue in a Cluster	Minimum R-squared for a Variable	Maximum 1-R**2 Ratio for a Variable
1	4.672880	0.5841	0.5841	1.770983	0.3810	
2	6.426502	0.8033	0.7293	0.476418	0.6329	0.4380
3	6.895347	0.8619	0.7954	0.418369	0.7421	0.3634
4	7.271218	0.9089	0.8773	0.238000	0.8652	0.2548
5	7.509218	0.9387	0.8773	0.236135	0.8652	0.1665
6	7.740000	0.9675	0.9295	0.141000	0.9295	0.2560
7	7.881000	0.9851	0.9405	0.119000	0.9405	0.2093
8	8.000000	1.0000	1.0000	0.000000	1.0000	0.0000

The principal component method first separates the variables into the same two clusters that were created in the first PROC VARCLUS run. Note that, in creating the third cluster, the principal component method identifies the variable Width. This is the same variable that is put into its own cluster in the preceding centroid method example. The tree diagram in Output 93.1.4 displays the cluster hierarchy.

Output 93.1.4 Tree Diagram from PROC TREE



It appears from the diagram that there are two, or possibly three, clusters present. However, the MAXC=8 option forces the VARCLUS procedure to split the clusters until each variable is in its own cluster.

References

Anderberg, M. R. (1973), *Cluster Analysis for Applications*, New York: Academic Press Inc.

Hand, D. J., Daly, F., Lunn, A. D., McConway, K. J., and Ostrowski E. (1994), *A Handbook of Small Data Sets*, London: Chapman & Hall, 297–298.

Harman, H. H. (1976), *Modern Factor Analysis*, Third Edition, Chicago: University of Chicago Press.

Harris, C. W., and Kaiser, H. F. (1964), "Oblique Factor Analytic Solutions by Orthogonal Transformation," *Psychometrika,* 32, 363–379.

Chapter 94

The VARCOMP Procedure

Contents

Overview: VARCOMP Procedure **7493**
Getting Started: VARCOMP Procedure **7494**
Analyzing the Cure Rate of Rubber 7494
Syntax: VARCOMP Procedure **7497**
PROC VARCOMP Statement 7498
BY Statement 7499
CLASS Statement 7499
MODEL Statement 7499
Details: VARCOMP Procedure **7500**
Missing Values 7500
Fixed and Random Effects 7500
Negative Variance Component Estimates 7501
Computational Methods 7502
Gauge Repeatability and Reproducibility Analysis 7503
Confidence Limits 7505
Displayed Output 7506
ODS Table Names 7507
Relationship to PROC MIXED 7508
Examples: VARCOMP Procedure **7508**
Example 94.1: Using the Four General Estimation Methods 7508
Example 94.2: Using the GRR Method 7512
References **7515**

Overview: VARCOMP Procedure

The VARCOMP procedure handles general linear models that have random effects. Random effects are classification effects with levels that are assumed to be randomly selected from an infinite population of possible levels. PROC VARCOMP estimates the contribution of each of the random effects to the variance of the dependent variable.

A single MODEL statement specifies the dependent variables and the effects: main effects, interactions, and nested effects. The effects must be composed of classification variables; no continuous variables are allowed on the right side of the equal sign.

You can specify certain effects as fixed (nonrandom) by putting them first in the MODEL statement and indicating the number of fixed effects with the FIXED= option. An intercept is always fitted and assumed fixed. Except for the effects specified as fixed, all other effects are assumed to be random. Their contribution to the model can be thought of as an observation from a distribution that is normally and independently distributed.

The dependent variables are grouped based on the similarity of their missing values. Each group of dependent variables is then analyzed separately. The columns of the design matrix $\mathbf{X}$ are formed in the same order in which the effects are specified in the MODEL statement. A singular parameterization involving just 0–1 dummy variables is used, as in the GLM procedure.

You can specify four general methods of estimation in the PROC VARCOMP statement by using the METHOD= option. They are TYPE1 (based on computation of Type I sum of squares for each effect), MIVQUE0, maximum likelihood (METHOD=ML), and restricted maximum likelihood (METHOD=REML). A fifth method, METHOD=GRR, provides a specialized analysis for gauge repeatability and reproducibility (R&R) studies. See the section "Gauge Repeatability and Reproducibility Analysis" on page 7503 for further details. Note that this method, along with the CL option in the MODEL statement for confidence limits, applies only to certain designs, namely balanced one-way or two-way designs. The other four general methods apply to any random-effects model and design.

Other procedures, such as PROC GLM, PROC MIXED, and PROC GLIMMIX, fit similar random effects models. The VARCOMP procedure is usually more computationally efficient for certain special designs and models. See the section "Relationship to PROC MIXED" on page 7508 for a more precise comparison with the MIXED procedure in particular.

The GAUGE application in SAS/QC software provides a graphical interface for computing many of the same statistics as METHOD=GRR in PROC VARCOMP.

Getting Started: VARCOMP Procedure

Analyzing the Cure Rate of Rubber

This example, using data from Hicks (1973), concerns an experiment to determine the sources of variability in cure rates of rubber. The goal of the experiment was to find out if the different laboratories contributed more to the variance of cure rates than did the different batches of raw materials. This information would be useful in trying to control the cure rate of the final product because it would provide insight into the sources of the variability in cure rates. The rubber used was cured at three temperatures, which were taken to be fixed. Three laboratories were chosen at random, and three different batches of raw material were tested at each combination of temperature and laboratory. The following statements read the data into the SAS data set Cure.

```
data Cure;
   input Lab Temp Batch $ Cure @@;
   datalines;
1 145 A 18.6   1 145 A 17.0   1 145 A 18.7   1 145 A 18.7
1 145 B 14.5   1 145 B 15.8   1 145 B 16.5   1 145 B 17.6
1 145 C 21.1   1 145 C 20.8   1 145 C 21.8   1 145 C 21.0
1 155 A  9.5   1 155 A  9.4   1 155 A  9.5   1 155 A 10.0
1 155 B  7.8   1 155 B  8.3   1 155 B  8.9   1 155 B  9.1
1 155 C 11.2   1 155 C 10.0   1 155 C 11.5   1 155 C 11.1
1 165 A  5.4   1 165 A  5.3   1 165 A  5.7   1 165 A  5.3
1 165 B  5.2   1 165 B  4.9   1 165 B  4.3   1 165 B  5.2
1 165 C  6.3   1 165 C  6.4   1 165 C  5.8   1 165 C  5.6
2 145 A 20.0   2 145 A 20.1   2 145 A 19.4   2 145 A 20.0
2 145 B 18.4   2 145 B 18.1   2 145 B 16.5   2 145 B 16.7
2 145 C 22.5   2 145 C 22.7   2 145 C 21.5   2 145 C 21.3
2 155 A 11.4   2 155 A 11.5   2 155 A 11.4   2 155 A 11.5
2 155 B 10.8   2 155 B 11.1   2 155 B  9.5   2 155 B  9.7
2 155 C 13.3   2 155 C 14.0   2 155 C 12.0   2 155 C 11.5
2 165 A  6.8   2 165 A  6.9   2 165 A  6.0   2 165 A  5.7
2 165 B  6.0   2 165 B  6.1   2 165 B  5.0   2 165 B  5.2
2 165 C  7.7   2 165 C  8.0   2 165 C  6.6   2 165 C  6.3
3 145 A 19.7   3 145 A 18.3   3 145 A 16.8   3 145 A 17.1
3 145 B 16.3   3 145 B 16.7   3 145 B 14.4   3 145 B 15.2
3 145 C 22.7   3 145 C 21.9   3 145 C 19.3   3 145 C 19.3
3 155 A  9.3   3 155 A 10.2   3 155 A  9.8   3 155 A  9.5
3 155 B  9.1   3 155 B  9.2   3 155 B  8.0   3 155 B  9.0
3 155 C 11.3   3 155 C 11.0   3 155 C 10.9   3 155 C 11.4
3 165 A  6.7   3 165 A  6.0   3 165 A  5.0   3 165 A  4.8
3 165 B  5.7   3 165 B  5.5   3 165 B  4.6   3 165 B  5.4
3 165 C  6.6   3 165 C  6.5   3 165 C  5.9   3 165 C  5.8
;
```

The variables Lab, Temp, and Batch contain levels of laboratory, temperature, and batch, respectively. The Cure variable contains the response values.

The following SAS statements perform a restricted maximum likelihood variance component analysis.

```
title 'Analyzing the Cure Rate of Rubber';
proc varcomp method=reml data=cure;
   class temp lab batch;
   model cure=temp|lab batch(lab temp) / fixed=1;
run;
```

The FIXED=1 option indicates that the first factor, Temp, is fixed. The effect specification Temp|Lab is equivalent to putting the three terms Temp, Lab, and Temp*Lab in the model. Batch(Lab Temp) is equivalent to putting Batch(Temp*Lab) in the MODEL statement. The results of this analysis are displayed in Figure 94.1 through Figure 94.4.

Figure 94.1 Class Level Information

Analyzing the Cure Rate of Rubber

Variance Components Estimation Procedure

Class Level Information

Class	Levels	Values
Temp	3	145 155 165
Lab	3	1 2 3
Batch	3	A B C

Number of Observations Read 108
Number of Observations Used 108

Dependent Variable: Cure

Figure 94.1 provides information about the variables used in the analysis and the number of observations and specifies the dependent variable.

Figure 94.2 Iteration History

REML Iterations

Iteration	Objective	Var(Lab)	Var(Temp*Lab)	Var(Batch(Temp*Lab))	Var(Error)
0	13.4500060254	0.5094464340	0	2.4004888633	0.5787185225
1	13.0898262160	0.3194348317	0	2.0869636935	0.6016005334
2	13.0893125570	0.3176048001	0	2.0738906134	0.6026217204
3	13.0893125555	0.3176017115	0	2.0738685461	0.6026234568

Convergence criteria met.

The "REML Iterations" table in Figure 94.2 displays the iteration history, which includes the value of the objective function associated with REML and the values of the variance components at each iteration.

Figure 94.3 REML Estimates

REML Estimates

Variance Component	Estimate
Var(Lab)	0.31760
Var(Temp*Lab)	0
Var(Batch(Temp*Lab))	2.07387
Var(Error)	0.60262

Figure 94.3 displays the REML estimates of the variance components.

Figure 94.4 Covariance Matrix for REML Estimates

Asymptotic Covariance Matrix of Estimates

	Var(Lab)	Var(Temp*Lab)
Var(Lab)	0.32452	0
Var(Temp*Lab)	0	0
Var(Batch(Temp*Lab))	-0.04998	0
Var(Error)	1.026E-12	0

Asymptotic Covariance Matrix of Estimates

	Var(Batch(Temp*Lab))	Var(Error)
Var(Lab)	-0.04998	1.026E-12
Var(Temp*Lab)	0	0
Var(Batch(Temp*Lab))	0.45042	-0.0022417
Var(Error)	-0.0022417	0.0089668

The "Asymptotic Covariance Matrix of Estimates" table in Figure 94.4 displays the asymptotic covariance matrix of the REML estimates.

The results of the analysis show that the variance attributable to Batch(Temp*Lab) (with a variance component of 2.0739) is considerably larger than the variance attributable to Lab (0.3176). Therefore, attempts to reduce the variability of cure rates should concentrate on improving the homogeneity of the batches of raw material used rather than standardizing the practices or equipment within the laboratories. Also, note that since the Batch(Temp*Lab) variance is considerably larger than the experimental error (Var(Error)=0.6026), the Batch(Temp*Lab) variability plays an important part in the overall variability of the cure rates.

Syntax: VARCOMP Procedure

The following statements are available in PROC VARCOMP:

```
PROC VARCOMP < options > ;
   CLASS variables ;
   MODEL dependent = < effects > < / options > ;
   BY variables ;
```

Only one MODEL statement is allowed. The BY, CLASS, and MODEL statements are described after the PROC VARCOMP statement.

PROC VARCOMP Statement

PROC VARCOMP < *options* > **;**

This statement invokes the VARCOMP procedure. You can specify the following options in the PROC VARCOMP statement.

DATA=*SAS-data-set*
specifies the input SAS data set to use. If this option is omitted, the most recently created SAS data set is used.

EPSILON=*number*
specifies the convergence value of the objective function for METHOD=ML or METHOD=REML. By default, EPSILON=1E−8.

MAXITER=*number*
specifies the maximum number of iterations for METHOD=ML or METHOD=REML. By default, MAXITER=50.

METHOD=TYPE1 | MIVQUE0 | ML | REML | GRR < *(options)* >
specifies which of the five methods (TYPE1, MIVQUE0, ML, REML, or GRR) you want to use. By default, METHOD=MIVQUE0. METHOD=GRR provides a specialized analysis only for certain designs, whereas the other four methods apply to any random-effects model and design. You can specify the following options in parentheses after METHOD=GRR.

SPECLIMITS=*(LSL,USL,*< *k*>*)*

SL=*(LSL,USL,*< *k*>*)*
specifies the specification limits for the first random factor, which is regarded as the product being tested in the gauge R&R study. The lower limit *(LSL)* must be smaller than the upper limit *(USL)*. The value *k* is optional. The default value is 6, which corresponds to the number of standard deviations between the "natural" tolerance limits containing the middle 99.73% of a normal process. *SPECLIMITS=(LSL,USL,k)* requests the estimates of the parameters PTR(LSL,USL,k) and Cp(LSL,USL,k) to be displayed.

RATIO
specifies that certain additional ratios of variance components should also be computed and displayed, such as proportion of total variance due to the process. These ratios are listed in Table 94.4.

For more information see the section "Computational Methods" on page 7502.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC VARCOMP to obtain separate analyzes on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement option NOTSORTED or DESCENDING in the BY statement for the VARCOMP procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

CLASS Statement

CLASS *variables* **;**

The CLASS statement specifies the classification variables to be used in the analysis. All effects in the MODEL statement must be composed of effects that appear in the CLASS statement. Classification variables can be either numeric or character; if they are character, only the first 16 characters are used.

Numeric classification variables are not restricted to integers since a variable's format determines the levels. For more information, see the discussion of the FORMAT statement in *SAS Language Reference: Dictionary*.

MODEL Statement

MODEL *dependent* = < *effects* > < / *options* > **;**

The MODEL statement gives the dependent variables and independent effects. If you specify more than one dependent variable, a separate analysis is performed for each one. The independent effects are limited to main effects, interactions, and nested effects; no continuous effects are allowed. All

independent effects must be composed of effects that appear in the CLASS statement. Effects are specified in the VARCOMP procedure in the same way as described for the ANOVA procedure. Only one MODEL statement is allowed.

The following options are available in the MODEL statement.

FIXED=*n*

specifies that the first *n* effects in the MODEL statement are fixed effects. The remaining effects are assumed to be random. By default, PROC VARCOMP assumes that all effects are random in the model. Keep in mind that if you use bar notation and, for example, specify Y=A|B / FIXED=2, then A*B is considered a random effect.

CL

specifies that confidence limits for all of the parameters of interest should be computed and displayed. Note that this option applies only to balanced one-way or two-way designs for METHOD=TYPE1 or GRR.

ALPHA=*α*

specifies the level of significance α for $(1-\alpha)100\%$ two-sided confidence limits. The value of α must be between 0 and 1. By default, α is equal to 0.05.

Details: VARCOMP Procedure

Missing Values

If an observation has a missing value for any variable used in the independent effects, then the analyses of all dependent variables omit this observation. An observation is deleted from the analysis of a given dependent variable if the observation's value for that dependent variable is missing. Note that a missing value in one dependent variable does not eliminate an observation from the analysis of the other dependent variables.

During processing, PROC VARCOMP groups the dependent variables on their missing values across observations so that sums of squares and crossproducts can be computed in the most efficient manner.

Fixed and Random Effects

Central to the idea of variance components models is the idea of fixed and random effects. Each effect in a variance components model must be classified as either a fixed or a random effect. Fixed effects arise when the levels of an effect constitute the entire population in which you are interested. For example, if a plant scientist is comparing the yields of three varieties of soybeans, then Variety would be a fixed effect, providing that the scientist was concerned about making inferences

about only these three varieties of soybeans. Similarly, if an industrial experiment focused on the effectiveness of two brands of a machine, Machine would be a fixed effect only if the experimenter's interest did not go beyond the two machine brands.

On the other hand, an effect is classified as a random effect when you want to make inferences about an entire population, and the levels in your experiment represent only a sample from that population. Psychologists comparing test results between different groups of subjects would consider Subject as a random effect. Depending on the psychologists' particular interest, the Group effect might be either fixed or random. For example, if the groups are based on the sex of the subject, then Sex would be a fixed effect. But if the psychologists are interested in the variability in test scores due to different teachers, then they might choose a random sample of teachers as being representative of the total population of teachers, and Teacher would be a random effect. Note that, in the soybean example presented earlier, if the scientists are interested in making inferences about the entire population of soybean varieties and randomly choose three varieties for testing, then Variety would be a random effect.

If all the effects in a model (except for the intercept) are considered random effects, then the model is called a *random-effects model*; likewise, a model with only fixed effects is called a *fixed-effects model*. The more common case, where some factors are fixed and others are random, is called a *mixed model*. In PROC VARCOMP, by default, effects are assumed to be random. You specify which effects are fixed by using the FIXED= option in the MODEL statement. In general, if an interaction or nested effect contains any effect that is random, then the interaction or nested effect should be considered a random effect as well.

In the linear model, each level of a fixed effect contributes a fixed amount to the expected value of the dependent variable. What makes a random effect different is that each level of a random effect contributes an amount that is viewed as a sample from a population of normally distributed variables, each with mean 0, and an unknown variance, much like the usual random error term that is a part of all linear models. The estimate of the variance associated with the random effect is known as the *variance component* because it measures the part of the overall variance contributed by that effect. Thus, PROC VARCOMP estimates the variance of the random variables that are associated with the random effects in your model, and the variance components tell you how much each of the random factors contributes to the overall variability in the dependent variable.

Negative Variance Component Estimates

The variance components estimated by PROC VARCOMP should theoretically be nonnegative because they are assumed to represent the variance of a random variable. Nevertheless, when you are using METHOD=MIVQUE0, TYPE1, or GRR, some estimates of variance components might become negative. (Due to the nature of the algorithms used for METHOD=ML and METHOD=REML, negative estimates are constrained to zero.) These negative estimates might arise for a variety of reasons:

- The variability in your data might be large enough to produce a negative estimate, even though the true value of the variance component is positive.

- Your data might contain outliers. Refer to Hocking (1983) for a graphical technique for detecting outliers in variance components models by using the SAS System.
- A different model for interpreting your data might be appropriate. Under some statistical models for variance components analysis, negative estimates are an indication that observations in your data are negatively correlated. Refer to Hocking (1984) for further information about these models.

Assuming you are satisfied that the model that PROC VARCOMP is using is appropriate for your data, it is common practice to treat negative variance components as if they are zero.

Computational Methods

Four methods of estimation can be specified in the PROC VARCOMP statement by using the METHOD= option. They are described in the following sections.

The Type I Method

This method (METHOD=TYPE1) computes the Type I sum of squares for each effect, equates each mean square involving only random effects to its expected value, and solves the resulting system of equations (Gaylor, Lucas, and Anderson 1970). The $\mathbf{X}'\mathbf{X} \mid \mathbf{X}'\mathbf{Y}$ matrix is computed and adjusted in segments whenever memory is not sufficient to hold the entire matrix.

The MIVQUE0 Method

Based on the technique suggested by Hartley, Rao, and LaMotte (1978), the MIVQUE0 method (METHOD=MIVQUE0) produces unbiased estimates that are invariant with respect to the fixed effects of the model and that are locally best quadratic unbiased estimates given that the true ratio of each component to the residual error component is zero. The technique is similar to TYPE1 except that the random effects are adjusted only for the fixed effects. This affords a considerable timing advantage over the TYPE1 method; thus, MIVQUE0 is the default method used in PROC VARCOMP. The $\mathbf{X}'\mathbf{X}|\mathbf{X}'\mathbf{Y}$ matrix is computed and adjusted in segments whenever memory is not sufficient to hold the entire matrix. Each element (i, j) of the form

$$\mathrm{SSQ}(\mathbf{X}_i'\mathbf{M}\mathbf{X}_j)$$

is computed, where

$$\mathbf{M} = \mathbf{I} - \mathbf{X}_0(\mathbf{X}_0'\mathbf{X}_0)^{-}\mathbf{X}_0'$$

and where $\mathbf{X}_0$ is part of the design matrix for the fixed effects, $\mathbf{X}_i$ is part of the design matrix for one of the random effects, and SSQ is an operator that takes the sum of squares of the elements. For more information refer to Rao (1971, 1972) and Goodnight (1978).

The Maximum Likelihood Method

The maximum likelihood method (METHOD=ML) computes maximum likelihood estimates of the variance components; refer to Searle, Casella, and McCulloch (1992). The computing algorithm makes use of the W-transformation developed by Hemmerle and Hartley (1973) and Goodnight and Hemmerle (1979). The procedure uses a Newton-Raphson algorithm, iterating until the log-likelihood objective function converges.

The objective function for METHOD=ML is $\ln(|\mathbf{V}|) + \mathbf{r}'\mathbf{V}^{-1}\mathbf{r}$, where

$$\mathbf{V} = \sigma_0^2\mathbf{I} + \sum_{i=1}^{n_r} \sigma_i^2 \mathbf{X}_i \mathbf{X}_i'$$

and where σ_0^2 is the residual variance, n_r is the number of random effects in the model, σ_i^2 represents the variance components, $\mathbf{X}_i$ is part of the design matrix for one of the random effects, and

$$\mathbf{r} = \mathbf{y} - \mathbf{X}_0(\mathbf{X}_0'\mathbf{V}^{-1}\mathbf{X}_0)^{-}\mathbf{X}_0'\mathbf{V}^{-1}\mathbf{y}$$

is the vector of residuals.

The Restricted Maximum Likelihood Method

The restricted maximum likelihood method (METHOD=REML) is similar to the maximum likelihood method, but it first separates the likelihood into two parts: one that contains the fixed effects and one that does not (Patterson and Thompson 1971). The procedure uses a Newton-Raphson algorithm, iterating until convergence is reached for the log-likelihood objective function of the portion of the likelihood that does not contain the fixed effects. Using notation from earlier methods, the objective function for METHOD=REML is $\ln(|\mathbf{V}|) + \mathbf{r}'\mathbf{V}^{-1}\mathbf{r} + \ln(|\mathbf{X}_0'\mathbf{V}^{-1}\mathbf{X}_0|)$. Refer to Searle, Casella, and McCulloch (1992) for additional details.

The GRR Method

Based on the technique suggested by Burdick, Borror, and Montgomery (2005), the GRR method (METHOD=GRR) produces minimum variance unbiased estimators.

Gauge Repeatability and Reproducibility Analysis

In a typical gauge R&R experiment, each operator (O_j) makes multiple observations on each of several similar parts (P_i) from a monitored process. The statistical model used to describe the response variable is the balanced two-factor crossed random model with interaction

$$y_{ijk} = \mu_y + P_i + O_j + (PO)_{ij} + E_{ijk}$$

where $i = 1, \ldots, p$, $j = 1, \ldots, o$, $k = 1, \ldots, r$, μ_y is an unknown constant, and $P_i, O_j, (PO)_{ij}, E_{ijk}$ are jointly independent normal random variables with means of zero and variances $\mathrm{Var}(P)$, $\mathrm{Var}(O)$, $\mathrm{Var}(PO)$, and $\mathrm{Var}(E)$, respectively. The corresponding SAS statements are as follows:

```
proc varcomp method=grr;
   class P O;
   model y = P|O;
run;
```

The first random effect in the MODEL statement is assumed to be the "Part" effect and the second is "Operator."

The ANOVA table for the preceding model is shown in Table 94.1.

Table 94.1 GRR Analysis of Variance

Source	DF	Mean Square	Expected Mean Square
Parts(P)	$p-1$	S_P^2	$\mathrm{Var}(E)+r\mathrm{Var}(PO)+or\mathrm{Var}(P)$
Operators(O)	$o-1$	S_O^2	$\mathrm{Var}(E)+r\mathrm{Var}(PO)+pr\mathrm{Var}(O)$
P×O	$(p-1)(o-1)$	S_{PO}^2	$\mathrm{Var}(E)+r\mathrm{Var}(PO)$
Error(E)	$po(r-1)$	S_E^2	$\mathrm{Var}(E)$

The gauge R&R parameters of interest are given in Table 94.2 in terms of $\mathrm{Var}(P)$, $\mathrm{Var}(O)$, $\mathrm{Var}(PO)$, and $\mathrm{Var}(E)$.

Table 94.2 Gauge R&R Parameters

Parameter	Formula
Variance of the monitored process	$\gamma_P = \mathrm{Var}(P)$
Variance of the measurement system	$\gamma_M = \mathrm{Var}(O)+\mathrm{Var}(PO)+\mathrm{Var}(E)$
Total variance of the response variable	$\gamma_y = \mathrm{Var}(y) = \gamma_P+\gamma_M$
Ratio of process variance to measurement variance	$\gamma_R = \gamma_P/\gamma_M$
Proportion of total variance due to the process	$\rho_P = \gamma_P/\gamma_y = \frac{\gamma_R}{1+\gamma_R}$
Proportion of total variance due to the measurement	$\rho_M = \gamma_M/\gamma_y = 1-\rho_P$
Signal-to-noise ratio	$\mathrm{SNR} = \sqrt{2\times\gamma_R}$
Discrimination ratio	$\mathrm{DR} = 1+2\gamma_R$

For a one-way model, $\gamma_M = \mathrm{Var}(E)$, and for a two-way model with no interaction, $\gamma_M = \mathrm{Var}(O)+\mathrm{Var}(E)$.

If you use the SPECLIMITS option to give specification limits, the two parameters in Table 94.3 will also be estimated and displayed.

Table 94.3 Gauge R&R Parameters Related to Specification Limits

Parameter	Formula
Precision-to-tolerance ratio	$\mathrm{PTR(LSL, USL, k)} = k\sqrt{\gamma_M}/(\mathrm{USL}-\mathrm{LSL})$
Process capability ratio	$\mathrm{Cp(LSL, USL, k)} = (\mathrm{USL}-\mathrm{LSL})/(k\sqrt{\gamma_P})$

Here, USL and LSL are the specification limits, and the value k corresponds to the number of standard deviations between the "natural" tolerance limits of a normal process.

If you use the RATIO option, the ratios in Table 94.4 will also be estimated and displayed.

Table 94.4 Gauge R&R Ratios

Ratio	Formula
Ratio of process variance to total variance	$\text{Var}(P)/\gamma_y$
Ratio of operator variance to total variance	$\text{Var}(O)/\gamma_y$
Ratio of process by operator variance to total variance	$\text{Var}(PO)/\gamma_y$
Ratio of process variance to residual variance	$\text{Var}(P)/\text{Var}(E)$
Ratio of operator variance to residual variance	$\text{Var}(O)/\text{Var}(E)$
Ratio of process by operator variance to residual variance	$\text{Var}(PO)/\text{Var}(E)$

Confidence Limits

When you are analyzing a balanced one-way or two-way design, then if you specify the CL option with METHOD=TYPE1 or GRR, the VARCOMP procedure will compute confidence limits by using modified large-sample (MLS) methods, as discussed in Burdick, Borror, and Montgomery (2005).

The formulas for these confidence limits are very complicated. Those for a balanced two-way crossed random model with interaction are given in Table 94.5. Refer to Burdick, Borror, and Montgomery (2005) for the formulas for one-way or balanced two-way with no interaction models.

Note that confidence limits for parameters such as variances and their ratios might not contain the corresponding point estimates, since negative confidence bounds are increased to zero.

Table 94.5 $100(1-\alpha)\%$ MLS Confidence Limits

Parameter	Lower Bound	Upper Bound
γ_P	$\hat{\gamma}_P - \sqrt{V_{LP}}/(or)$	$\hat{\gamma}_P + \sqrt{V_{UP}}/(or)$
γ_M	$\hat{\gamma}_M - \sqrt{V_{LM}}/(pr)$	$\hat{\gamma}_M + \sqrt{V_{UM}}/(pr)$
γ_y	$\hat{\gamma}_y - \sqrt{V_{LT}}/(por)$	$\hat{\gamma}_y + \sqrt{V_{UT}}/(por)$
γ_R	L_R	U_R
ρ_P	$L_R/(1+L_R)$	$U_R/(1+U_R)$
ρ_M	$1/(1+U_R)$	$1/(1+L_R)$

The terms in Table 94.5 are defined as follows:

$$
\begin{aligned}
V_{LP} &= G_1^2 S_P^4 + H_3^2 S_{PO}^4 + G_{13} S_P^2 S_{PO}^2 \\
V_{UP} &= H_1^2 S_P^4 + G_3^2 S_{PO}^4 + H_{13} S_P^2 S_{PO}^2 \\
V_{LM} &= G_2^2 S_O^4 + G_3^2 (p-1)^2 S_{PO}^4 + G_4^2 p^2 (r-1)^2 S_E^4 \\
V_{UM} &= H_2^2 S_O^4 + H_3^2 (p-1)^2 S_{PO}^4 + H_4^2 p^2 (r-1)^2 S_E^4 \\
V_{LT} &= G_1^2 p^2 S_P^4 + G_2^2 o^2 S_O^4 + G_3^2 (po-p-o)^2 S_{PO}^4 + G_4^2 (po)^2 (r-1)^2 S_E^4 \\
V_{UT} &= H_1^2 p^2 S_P^4 + H_2^2 o^2 S_O^4 + H_3^2 (po-p-o)^2 S_{PO}^4 + H_4^2 (po)^2 (r-1)^2 S_E^4 \\
L_R &= \frac{p(1-G_1)(S_P^2 - F_1 S_{PO}^2)}{po(r-1)S_E^2 + o(1-G_1)F_3 S_O^2 + o(p-1)S_{PO}^2} \\
U_R &= \frac{p(1+H_1)(S_P^2 - F_2 S_{PO}^2)}{po(r-1)S_E^2 + o(1+H_1)F_4 S_O^2 + o(p-1)S_{PO}^2} \\
G_1 &= 1 - F_{\alpha/2:\infty,p-1} \\
G_2 &= 1 - F_{\alpha/2:\infty,o-1} \\
G_3 &= 1 - F_{\alpha/2:\infty,(p-1)(o-1)} \\
G_4 &= 1 - F_{\alpha/2:\infty,po(r-1)} \\
H_1 &= F_{1-\alpha/2:\infty,p-1} - 1 \\
H_2 &= F_{1-\alpha/2:\infty,o-1} - 1 \\
H_3 &= F_{1-\alpha/2:\infty,(p-1)(o-1)} - 1 \\
H_4 &= F_{1-\alpha/2:\infty,po(r-1)} - 1 \\
F_1 &= F_{1-\alpha/2:p-1,(p-1)(o-1)} \\
F_2 &= F_{\alpha/2:p-1,(p-1)(o-1)} \\
F_3 &= F_{1-\alpha/2:p-1,o-1} \\
F_4 &= F_{\alpha/2:p-1,o-1} \\
G_{13} &= \frac{(F_1-1)^2 - G_1^2 F_1^2 - H_3^2}{F_1} \\
H_{13} &= \frac{(1-F_2)^2 - H_1^2 F_2^2 - G_3^2}{F_2}
\end{aligned}
$$

The symbol $F_{\alpha:df1,df2}$ represents the percentile of an F distribution with *df1* and *df2* degrees of freedom and area α to the left.

Displayed Output

PROC VARCOMP displays the following items:

- Class Level Information for verifying the levels in your data
- Number of observations read from the data set and number of observations used in the analysis

- for METHOD=TYPE1, an analysis-of-variance table with Source, DF, Type I Sum of Squares, Type I Mean Square, and Expected Mean Square, and a table of Type I variance component estimates
- for METHOD=MIVQUE0, the SSQ Matrix containing sums of squares of partitions of the $\mathbf{X}'\mathbf{X}$ crossproducts matrix adjusted for the fixed effects
- for METHOD=ML and METHOD=REML, the iteration history, including the objective function, a table of variance component estimates, and the estimated Asymptotic Covariance Matrix of the variance components
- for METHOD=GRR, an analysis-of-variance table with Source, DF, GRR Sum of Squares, GRR Mean Square, and Expected Mean Square, and a table of GRR parameter estimates. If the CL option is specified, confidence limits for each parameter estimate will also be displayed.

ODS Table Names

PROC VARCOMP assigns a name to each table it creates. You can use these names to reference the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in Table 94.6. For more information about ODS, see Chapter 20, "Using the Output Delivery System."

Table 94.6 ODS Tables Produced by PROC VARCOMP

ODS Table Name	Description	Statement
ANOVA	Type 1 analysis of variance	METHOD=TYPE1 or GRR
AsyCov	Asymptotic covariance matrix of estimates	METHOD=ML or REML
ClassLevels	Class level information	default
ConvergenceStatus	Convergence status	METHOD=ML or REML
DepVar	Dependent variable	METHOD=TYPE1, REML, ML, or GRR
DependentInfo	Dependent variable info (multiple variables)	
Estimates	Variance component estimates	default
IterHistory	Iteration history	METHOD=ML or REML
NObs	Number of observations	default
SSCP	Sum of squares matrix	METHOD=MIVQUE0

In situations where multiple dependent variables are analyzed that differ in their missing value pattern, separate names for ANOVA*n*, AsyCov*n*, Estimates*n*, IterHistory*n*, and SSCP*n* tables are no longer required. The results are combined into a single output data set. For METHOD=TYPE1, ML, or REML, the variable Dependent in the output data set identifies the dependent variable. For METHOD=MIVQUE0, a variable is added to the output data set for each dependent variable.

Relationship to PROC MIXED

The MIXED procedure effectively performs the same analyzes as PROC VARCOMP and many others, including Type I, Type II, and Type III tests of fixed effects, confidence limits, customized contrasts, and least squares means. Furthermore, continuous variables are permitted as both fixed and random effects in PROC MIXED, and numerous other covariance structures besides variance components are available. The VARCOMP procedure is more computationally efficient for some special designs and models.

To translate PROC VARCOMP code into PROC MIXED code, move all random effects to the RANDOM statement in PROC MIXED. For example, the syntax for the example in the section "Getting Started: VARCOMP Procedure" on page 7494 is as follows:

```
proc mixed;
   class Temp Lab Batch;
   model Cure = Temp;
   random Lab Temp*Lab Batch(Lab Temp);
run;
```

REML is the default estimation method in PROC MIXED, and you can specify other methods by using the METHOD= option.

Examples: VARCOMP Procedure

Example 94.1: Using the Four General Estimation Methods

In this example, a and b are classification variables and y is the dependent variable. a is declared fixed, and b and a*b are random. Note that this design is unbalanced because the cell sizes are not all the same. PROC VARCOMP is invoked four times, once for each of the general estimation methods. The data are from Hemmerle and Hartley (1973). The following statements produce Output 94.1.1.

```
data a;
   input a b y @@;
   datalines;
1 1 237   1 1 254   1 1 246   1 2 178   1 2 179
2 1 208   2 1 178   2 1 187   2 2 146   2 2 145   2 2 141
3 1 186   3 1 183   3 2 142   3 2 125   3 2 136
;

proc varcomp method=type1 data=a;
   class a b;
   model y=a|b / fixed=1;
run;
```

Output 94.1.1 VARCOMP Procedure: Method=TYPE1

Variance Components Estimation Procedure

Class Level Information

Class	Levels	Values
a	3	1 2 3
b	2	1 2

Number of Observations Read 16
Number of Observations Used 16

Dependent Variable: y

Type 1 Analysis of Variance

Source	DF	Sum of Squares	Mean Square	Expected Mean Square
a	2	11736	5868.218750	Var(Error) + 2.725 Var(a*b) + 0.1 Var(b) + Q(a)
b	1	11448	11448	Var(Error) + 2.6308 Var(a*b) + 7.8 Var(b)
a*b	2	299.041026	149.520513	Var(Error) + 2.5846 Var(a*b)
Error	10	786.333333	78.633333	Var(Error)
Corrected Total	15	24270		

Type 1 Estimates

Variance Component	Estimate
Var(b)	1448.4
Var(a*b)	27.42659
Var(Error)	78.63333

The "Class Level Information" table in Output 94.1.1 displays the levels of each variable specified in the CLASS statement. You can check this table to make sure the data are input correctly.

The Type I analysis of variance in Output 94.1.1 consists of a sequential partition of the total sum of squares. The mean square is the sum of squares divided by the degrees of freedom, and the expected mean square is the expected value of the mean square under the mixed model. The "Q" notation in the expected mean squares refers to a quadratic form in parameters of the parenthesized effect.

The Type I estimates of the variance components in Output 94.1.1 result from solving the linear system of equations established by equating the observed mean squares to their expected values.

The following statements are the same as before, except that the estimation method is MIVQUE0 instead of the default TYPE1. They produce Output 94.1.2.

```
proc varcomp method=mivque0 data=a;
   class a b;
   model y=a|b / fixed=1;
run;
```

Output 94.1.2 VARCOMP Procedure: Method=MIVQUE0

```
                  Variance Components Estimation Procedure

                              MIVQUE(0) SSQ Matrix

Source                    b                  a*b                Error                    y

b                  60.84000            20.52000             7.80000             89295.4
a*b                20.52000            20.52000             7.80000             30181.3
Error               7.80000             7.80000            13.00000             12533.5

                              MIVQUE(0) Estimates

                         Variance Component                  y

                         Var(b)                         1466.1
                         Var(a*b)                     -35.49170
                         Var(Error)                   105.73660
```

The MIVQUE0 estimates in Output 94.1.2 result from solving the equations established by the MIVQUE0 SSQ matrix. Note that the estimate of the variance component for the interaction effect, Var(a*b), is negative for this example.

The following statements use METHOD=ML to invoke maximum likelihood estimation. They produce Output 94.1.3.

```
proc varcomp method=ml data=a;
   class a b;
   model y=a|b / fixed=1;
run;
```

Output 94.1.3 VARCOMP Procedure: Method=ML

```
                  Variance Components Estimation Procedure

                          Maximum Likelihood Iterations

Iteration        Objective           Var(b)         Var(a*b)       Var(Error)

        0    78.3850371200       1031.49070                0    74.3909717935
        1    78.2637043807   732.3606453635                0    77.4011688154
        2    78.2635471161   723.6867470850                0    77.5301774839
        3    78.2635471152   723.6658365289                0    77.5304926877

       Convergence criteria met.
```

Output 94.1.3 *continued*

Maximum Likelihood Estimates

Variance Component	Estimate
Var(b)	723.66584
Var(a*b)	0
Var(Error)	77.53049

Asymptotic Covariance Matrix of Estimates

	Var(b)	Var(a*b)	Var(Error)
Var(b)	537826.1	0	-107.33905
Var(a*b)	0	0	0
Var(Error)	-107.33905	0	858.71104

The "Maximum Likelihood Iterations" table in Output 94.1.3 shows that the Newton-Raphson algorithm used by PROC VARCOMP requires three iterations to converge.

The ML estimate of Var(a*b) is zero for this example, and the other two estimates are smaller than their Type I and MIVQUE0 counterparts.

One benefit of using likelihood-based methods is that an approximate covariance matrix is available from the matrix of second derivatives evaluated at the ML solution. This covariance matrix is valid asymptotically and can be unreliable in small samples.

Here the variance component estimates for B and the Error are negatively correlated, and the elements for Var(a*b) are set to zero because the estimate equals zero. Also, the very large variance for Var(b) indicates a lot of uncertainty about the estimate for Var(b), and one contributing explanation is that B has only two levels in this data set.

Finally, the following statements use the restricted maximum likelihood (REML) for estimation. They produce Output 94.1.4.

```
proc varcomp method=reml data=a;
   class a b;
   model y=a|b / fixed=1;
run;
```

Output 94.1.4 VARCOMP Procedure: Method=REML

Variance Components Estimation Procedure

REML Iterations

Iteration	Objective	Var(b)	Var(a*b)	Var(Error)
0	63.4134144942	1269.52701	0	91.5581191305
1	63.0446869787	1601.84199	32.7632417174	76.9355562461
2	63.0311530508	1468.82932	27.2258186561	78.7548276319
3	63.0311265148	1464.33646	26.9564053003	78.8431476502
4	63.0311265127	1464.36727	26.9588525177	78.8423898761

Convergence criteria met.

REML Estimates

Variance Component	Estimate
Var(b)	1464.4
Var(a*b)	26.95885
Var(Error)	78.84239

Asymptotic Covariance Matrix of Estimates

	Var(b)	Var(a*b)	Var(Error)
Var(b)	4401703.8	1.29359	-273.39651
Var(a*b)	1.29359	3559.1	-502.85157
Var(Error)	-273.39651	-502.85157	1249.7

The "REML Iterations" table in Output 94.1.4 shows that the REML optimization requires four iterations to converge.

The REML estimates in Output 94.1.4 are all larger than the corresponding ML estimates (adjusting for potential downward bias) and are fairly similar to the Type I estimates.

The "Asymptotic Covariance Matrix of Estimates" table in Output 94.1.4 shows that the Error variance component estimate is negatively correlated with the other two variance component estimates, and the estimated variances are all larger than their ML counterparts.

Example 94.2: Using the GRR Method

In this example from Houf and Burman (1988), the response variable is the thermal performance of a module measured in Celsius degrees per watt. Each of three operators measures 10 parts three times. It is assumed that parts and operators are selected at random from larger populations. The following statements produce Output 94.2.1.

```
data Houf;
   input a b y @@;
   datalines;
1  1 37    1  1 38    1 1 37
1  2 41    1  2 41    1 2 40
1  3 41    1  3 42    1 3 41
2  1 42    2  1 41    2 1 43
2  2 42    2  2 42    2 2 42
2  3 43    2  3 42    2 3 43
3  1 30    3  1 31    3 1 31
3  2 31    3  2 31    3 2 31
3  3 29    3  3 30    3 3 28
4  1 42    4  1 43    4 1 42
4  2 43    4  2 43    4 2 43
4  3 42    4  3 42    4 3 42
5  1 28    5  1 30    5 1 29
5  2 29    5  2 30    5 2 29
5  3 31    5  3 29    5 3 29
6  1 42    6  1 42    6 1 43
6  2 45    6  2 45    6 2 45
6  3 44    6  3 46    6 3 45
7  1 25    7  1 26    7 1 27
7  2 28    7  2 28    7 2 30
7  3 29    7  3 27    7 3 27
8  1 40    8  1 40    8 1 40
8  2 43    8  2 42    8 2 42
8  3 43    8  3 43    8 3 41
9  1 25    9  1 25    9 1 25
9  2 27    9  2 29    9 2 28
9  3 26    9  3 26    9 3 26
10 1 35   10  1 34   10 1 34
10 2 35   10  2 35   10 2 34
10 3 35   10  3 34   10 3 35
;

proc varcomp data=Houf method=grr (speclimits=(18,58) ratio);
   class a b;
   model y=a|b/cl;
run;
```

You specify METHOD=GRR in this example to drive the VARCOMP procedure to produce a gauge repeatability and reproducibility analysis. With the option speclimits=(18 58), the parameters PTR and Cp are estimated and displayed. With the option ratio, certain additional ratios of variance components are also estimated and displayed.

Output 94.2.1 Class Level Information Using Method=GRR

```
Variance Components Estimation Procedure

Class Level Information

Class          Levels    Values

a                  10    1 2 3 4 5 6 7 8 9 10

b                   3    1 2 3

Number of Observations Read          90
Number of Observations Used          90

Dependent Variable:    y
```

The "Class Level Information" table in Output 94.2.1 displays the levels of each variable specified in the CLASS statement.

Output 94.2.2 Analysis of Variance Using Method=GRR

```
GRR Analysis of Variance

                                    Sum of
Source                  DF         Squares     Mean Square

a                        9     3935.955556      437.328395
b                        2       39.266667       19.633333
a*b                     18       48.511111        2.695062
Error                   60       30.666667        0.511111
Corrected Total         89     4054.400000

GRR Analysis of Variance

Source              Expected Mean Square

a                   Var(Error) + 3 Var(a*b) + 9 Var(a)
b                   Var(Error) + 3 Var(a*b) + 30 Var(b)
a*b                 Var(Error) + 3 Var(a*b)
Error               Var(Error)
Corrected Total
```

The GRR analysis of variance in Output 94.2.2 is the same as for the Type I analysis when the design is balanced.

Finally, the estimates of the GRR parameters of interest and their confidence limits are displayed in Output 94.2.3.

Output 94.2.3 Parameter Estimates Using Method=GRR

GRR Estimates

Parameter	Estimate	95% Confidence Limits	
Var(a)	48.29259	22.69452	161.63918
Var(b)	0.56461	0.07296	25.75077
Var(a*b)	0.72798	0.33273	1.79272
Var(Error)	0.51111	0.36816	0.75754
Gamma Y	50.09630	24.48844	166.22217
Gamma P	48.29259	22.69452	161.63918
Gamma M	1.80370	1.20623	27.01724
Gamma R	26.77413	1.69168	105.60895
SNR	7.31767	1.83939	14.53334
PTR(18,58,6)	0.20145	0.16474	0.77967
Cp(18,58,6)	0.95933	0.52437	1.39942
DR	54.54825	4.38336	212.21791
Rho P	0.96400	0.62848	0.99062
Rho M	0.03600	0.0093801	0.37152
Var(a)/Gamma Y	0.96400	0.62848	0.99062
Var(b)/Gamma Y	0.01127	0.0008700	0.34151
Var(a*b)/Gamma Y	0.01453	0.0027083	0.04744
Var(a)/Var(Error)	94.48551	40.19199	327.32469
Var(b)/Var(Error)	1.10467	0.13662	50.37744
Var(a*b)/Var(Error)	1.42432	0.55232	3.74691

You can draw the following inferences from the results of the analysis. Most of the variation is due to differences between parts because of the relative larger value of Gamma R. The measurement system is nearly inadequate because the PTR exceeds 20%. However, the measurement system is of value in monitoring the process since the SNR is greater than five. Refer to Burdick, Borror, and Montgomery (2003) for more information about interpreting gauge R&R studies.

References

Burdick, R. K., Borror, C. M., and Montgomery, D. C. (2003), "A Review of Methods for Measurement Systems Capability Analysis," *Journal of Quality Technology*, 35, 342–354.

Burdick, R. K., Borror, C. M., and Montgomery, D. C. (2005), *Design and Analysis of Gauge R&R Studies: Making Decisions with Confidence Intervals in Random and Mixed ANOVA Models*, Philadelphia, PA and Alexandria, VA: SIAM and ASA.

Gaylor, D. W., Lucas, H. L., and Anderson, R. L. (1970), "Calculation of Expected Mean Squares by the Abbreviated Doolittle and Square Root Methods," *Biometrics*, 26, 641–655.

Goodnight, J. (1978), *Computing MIVQUE0 Estimates of Variance Components*, Technical report, SAS Institute Inc, Cary, NC, SAS Technical Report R-105 Edition.

Goodnight, J. H. and Hemmerle, W. J. (1979), "A Simplified Algorithm for the W-Transformation in Variance Component Estimation," *Technometrics*, 21, 265–268.

Hartley, H. O., Rao, J. N. K., and LaMotte, L. (1978), "A Simple Synthesis-Based Method of Variance Component Estimation," *Biometrics*, 34, 233–244.

Hemmerle, W. J. and Hartley, H. O. (1973), "Computing Maximum Likelihood Estimates for the Mixed AOV Model Using the W-Transformation," *Technometrics*, 15, 819–831.

Hicks, C. R. (1973), *Fundamental Concepts in the Design of Experiments*, New York: Holt, Rinehart and Winston.

Hocking, R. R. (1983), "A Diagnostic Tool for Mixed Models with Applications to Negative Estimates of Variance Components," in *Proceedings of the Eighth Annual SAS Users Group International Conference*, Cary, NC: SAS Institute Inc.

Hocking, R. R. (1984), *Analysis of Linear Models*, Monterey, CA: Brooks/Cole.

Houf, R. E. and Burman, D. B. (1988), "Statistical Analysis of Power Module Thermal Test Equipment Performance," *IEEE Transactions on Components Hybrids, and Manufacturing Technology*, 11, 516–520.

Patterson, H. D. and Thompson, R. (1971), "Recovery of Inter-Block Information When Block Sizes Are Unequal," *Biometrika*, 58, 545–554.

Rao, C. R. (1971), "Minimum Variance Quadratic Unbiased Estimation of Variance Components," *Journal of Multivariate Analysis*, 1, 445–456.

Rao, C. R. (1972), "Estimation of Variance and Covariance Components in Linear Models," *Journal of the American Statistical Association*, 67, 112–115.

Searle, S. R., Casella, G., and McCulloch, C. E. (1992), *Variance Components*, New York: John Wiley & Sons.

Chapter 95

The VARIOGRAM Procedure

Contents

Overview: VARIOGRAM Procedure **7518**
Introduction to Spatial Prediction 7518
Getting Started: VARIOGRAM Procedure **7519**
Preliminary Spatial Data Analysis 7520
Empirical Semivariogram Computation 7525
Syntax: VARIOGRAM Procedure **7527**
PROC VARIOGRAM Statement 7529
BY Statement 7533
COMPUTE Statement 7533
COORDINATES Statement 7538
DIRECTIONS Statement 7539
VAR Statement 7539
Details: VARIOGRAM Procedure **7540**
Theoretical Semivariogram Models 7540
Theoretical and Computational Details of the Semivariogram 7544
Stationarity 7546
Ergodicity 7547
Anisotropy 7547
Pair Formation 7548
Angle Classification 7549
Distance Classification 7551
Bandwidth Restriction 7552
Computation of the Distribution Distance Classes 7553
Semivariance Computation 7558
Empirical Semivariograms and Surface Trends 7559
Autocorrelation Statistics (Experimental) 7560
Autocorrelation Weights 7560
Autocorrelation Statistics Types 7562
Interpretation 7564
Computational Resources 7565
Output Data Sets 7565
Displayed Output 7568
ODS Table Names 7569
ODS Graphics 7569

Examples: VARIOGRAM Procedure **7570**
Example 95.1: Theoretical Semivariogram Model Fitting 7570
Example 95.2: An Anisotropic Case Study with Surface Trend in the Data . 7574
Analysis with Surface Trend Removal 7578
Example 95.3: Analysis without Surface Trend Removal 7587
Example 95.4: Covariogram and Semivariogram 7596
Example 95.5: A Box Plot of the Square Root Difference Cloud 7600
References **7604**

Overview: VARIOGRAM Procedure

The VARIOGRAM procedure computes empirical measures of spatial continuity for two-dimensional spatial data. These measures are a function of the distances between the sample data pairs. When the data are free of nonrandom (or systematic) surface trends, the estimated continuity measures are the empirical semivariance and covariance. These measures can be used in subsequent analysis to perform spatial prediction. The procedure plots empirical semivariograms, which can also be saved to an output data set to enable parameter estimation for theoretical semivariogram or covariance models. Both isotropic and anisotropic measures are available.

In addition, PROC VARIOGRAM provides the Moran I and Geary c spatial autocorrelation statistics. The procedure also produces the OUTPAIR= and OUTDISTANCE= data sets that contain information about the semivariogram analysis.

The VARIOGRAM procedure now uses ODS Graphics to create graphs as part of its output. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For more information about the graphics available in PROC VARIOGRAM, see the section "ODS Graphics" on page 7569.

Introduction to Spatial Prediction

Many activities in science and technology involve measurements of one or more quantities at given spatial locations, with the goal of predicting the measured quantities at unsampled locations. Application areas include reservoir prediction in mining and petroleum exploration, as well as modeling in a broad spectrum of fields (for example, environmental health, environmental pollution, natural resources and energy, hydrology, risk analysis). Often, the unsampled locations are on a regular grid, and the predictions are used to produce surface plots or contour maps.

The preceding tasks fall within the scope of *spatial prediction*, which, in general, is any prediction method that incorporates spatial dependence. The study of these tasks involves naturally occurring uncertainties that cannot be ignored. Stochastic analysis frameworks and methods are used to account for these uncertainties. Hence, the terms *stochastic spatial prediction* and *stochastic modeling* are also used to characterize this type of analysis.

A popular method of spatial prediction is *ordinary kriging*, which produces both predicted values and associated standard errors. Ordinary kriging requires the complete specification (the form and parameter values) of the spatial dependence characterizing the spatial process. For this purpose, models for the spatial dependence are expressed in terms of the distance between any two locations in the spatial domain of interest. These models take the form of a covariance or semivariance function.

Spatial prediction, then, involves two steps. First, you model the covariance or semivariance of the spatial process. These measures are typically not known in advance. This step involves computing an empirical estimate, as well as determining both the mathematical form and the values of any parameters for a theoretical form of the dependence model. Second, you use this dependence model to solve the kriging system at a specified set of spatial points, resulting in predicted values and associated standard errors.

SAS/STAT software has two procedures corresponding to these steps for spatial prediction of two-dimensional data. The VARIOGRAM procedure is used in the first step (that is, calculating and modeling the dependence model), and the KRIGE2D procedure performs the kriging operations to produce the final predictions.

This introduction concludes with a note on terminology. You might commonly encounter the terms *estimation* and *prediction* used interchangeably by experts in different fields; this could be a source of confusion. A precise statistical vernacular uses the term *estimation* to refer to inferences about the value of fixed but unknown parameters, whereas *prediction* concerns inferences about the value of random variables—see, for example, Cressie (1993, p. 106). In light of these definitions, kriging methods are clearly predictive techniques, since they are concerned with making inferences about the value of a spatial random field at observed or unobserved locations. The SAS/STAT suite of procedures for spatial analysis and prediction (VARIOGRAM, KRIGE2D, and SIM2D) follows the statistical vernacular in the use of the terms *estimation* and *prediction*.

Getting Started: VARIOGRAM Procedure

PROC VARIOGRAM uses your data to compute the empirical semivariogram. This computation refers to the steps you take to derive the empirical semivariance from the data, and then to produce the corresponding semivariogram plot.

You can proceed further with the semivariogram analysis if the data are free of systematic trends. In that case, you can use the empirical outcome to determine a theoretical semivariogram model by using automated or visual methods. The model characterizes the type of theoretical semivariance function you will use to describe spatial dependence in your data set.

Some of the following graphical displays are requested by using the ODS GRAPHICS statement. For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." For specific information about the graphics available in the VARIOGRAM procedure, see the section "ODS Graphics" on page 7569.

Preliminary Spatial Data Analysis

The following data set simulates measurements of coal seam thickness (in feet) taken over an approximately square area. The coordinates are offsets from a point in the southwest corner of the measurement area, with the north and east distances in units of thousands of feet.

```
data thick;
   input East North Thick @@;
   label Thick='Coal Seam Thickness';
   datalines;
 0.7  59.6  34.1   2.1  82.7  42.2   4.7  75.1  39.5
 4.8  52.8  34.3   5.9  67.1  37.0   6.0  35.7  35.9
 6.4  33.7  36.4   7.0  46.7  34.6   8.2  40.1  35.4
13.3   0.6  44.7  13.3  68.2  37.8  13.4  31.3  37.8
17.8   6.9  43.9  20.1  66.3  37.7  22.7  87.6  42.8
23.0  93.9  43.6  24.3  73.0  39.3  24.8  15.1  42.3
24.8  26.3  39.7  26.4  58.0  36.9  26.9  65.0  37.8
27.7  83.3  41.8  27.9  90.8  43.3  29.1  47.9  36.7
29.5  89.4  43.0  30.1   6.1  43.6  30.8  12.1  42.8
32.7  40.2  37.5  34.8   8.1  43.3  35.3  32.0  38.8
37.0  70.3  39.2  38.2  77.9  40.7  38.9  23.3  40.5
39.4  82.5  41.4  43.0   4.7  43.3  43.7   7.6  43.1
46.4  84.1  41.5  46.7  10.6  42.6  49.9  22.1  40.7
51.0  88.8  42.0  52.8  68.9  39.3  52.9  32.7  39.2
55.5  92.9  42.2  56.0   1.6  42.7  60.6  75.2  40.1
62.1  26.6  40.1  63.0  12.7  41.8  69.0  75.6  40.1
70.5  83.7  40.9  70.9  11.0  41.7  71.5  29.5  39.8
78.1  45.5  38.7  78.2   9.1  41.7  78.4  20.0  40.8
80.5  55.9  38.7  81.1  51.0  38.6  83.8   7.9  41.6
84.5  11.0  41.5  85.2  67.3  39.4  85.5  73.0  39.8
86.7  70.4  39.6  87.2  55.7  38.8  88.1   0.0  41.6
88.4  12.1  41.3  88.4  99.6  41.2  88.8  82.9  40.5
88.9   6.2  41.5  90.6   7.0  41.5  90.7  49.6  38.9
91.5  55.4  39.0  92.9  46.8  39.1  93.4  70.9  39.7
55.8  50.5  38.1  96.2  84.3  40.3  98.2  58.2  39.5
;
```

It is instructive to see the locations of the measured points in the area where you want to perform spatial prediction. It is desirable to have the sampling locations scattered evenly throughout the prediction area. If the locations are not scattered evenly, the prediction error might be unacceptably large where measurements are sparse.

You can run PROC VARIOGRAM in this preliminary analysis to determine potential problems. In the following statements, the NOVARIOGRAM option in the COMPUTE statement specifies that only the descriptive summaries and a plot of the raw data should be produced.

```
ods graphics on;

proc variogram data=thick plots=pairs(thr=30);
   compute novariogram nhc=20;
   coordinates xc=East yc=North;
   var Thick;
run;
```

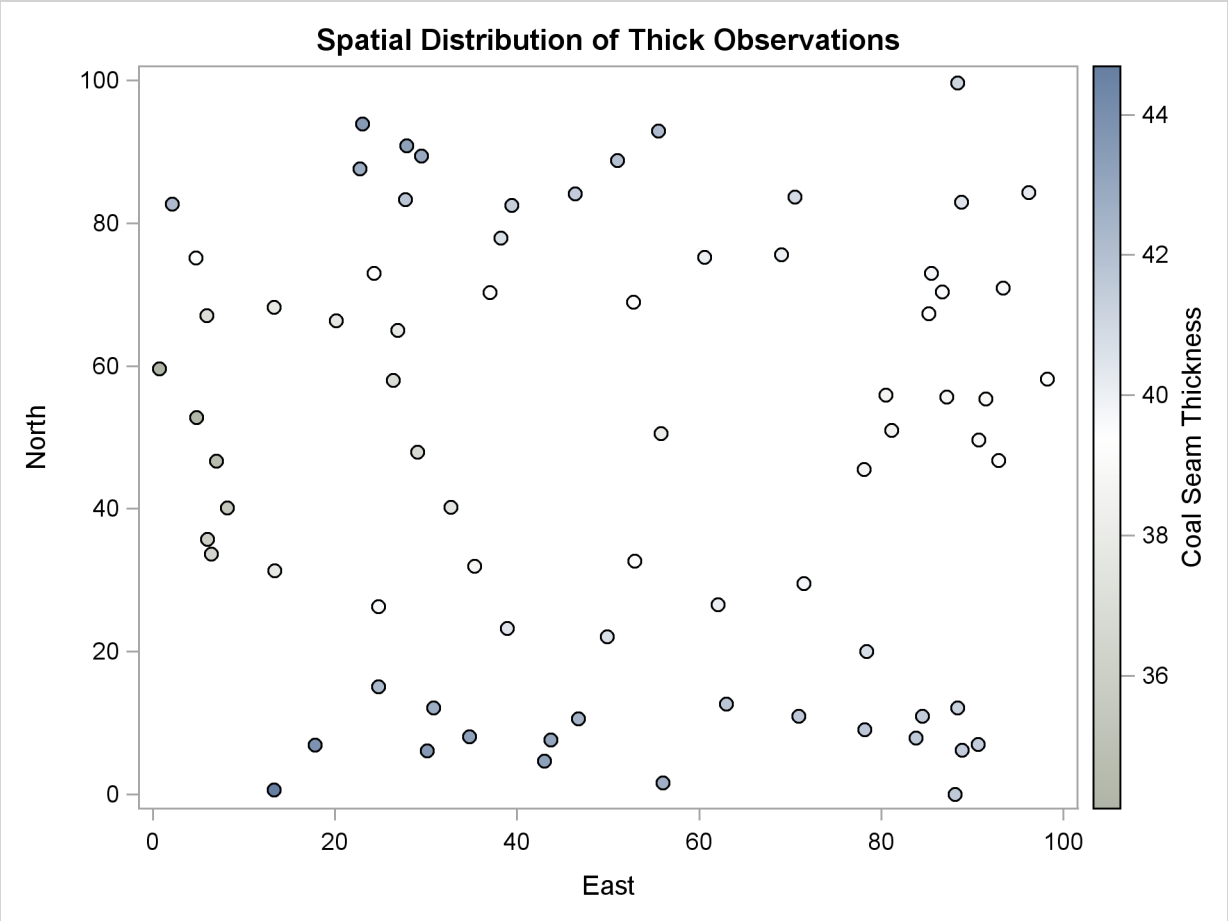
PROC VARIOGRAM produces the table in Figure 95.1 that shows the number of Thick observations read and used. This table provides you with useful information in case you have missing values in the input data.

Figure 95.1 Number of Observations for the thick Data Set

```
                    The VARIOGRAM Procedure
                  Dependent Variable: Thick

          Number of Observations Read          75
          Number of Observations Used          75
```

Then, the scatter plot of the observed data is produced as shown in Figure 95.2. According to the figure, while the locations are not ideally spread around the prediction area, there are not any extended areas lacking measurements. The same graph also provides the values of the measured variable by using colored markers.

Figure 95.2 Scatter Plot of the Observations Spatial Distribution



The following is a crucial step. Any obvious surface trend must be removed before you compute the empirical semivariogram and proceed to estimate a model of spatial dependence (the theoretical semivariogram model). You can observe in Figure 95.2 the small-scale variation typical of spatial data, but a first inspection indicates no obvious major systematic trend.

Assuming, therefore, that the data are free of surface trends, you can work with the original thickness rather than residuals obtained from a trend removal process. The following analysis also assumes that the spatial characterization is independent of the direction of the line connecting any two equidistant pairs of data; this is a property known as isotropy. See "Example 95.2: An Anisotropic Case Study with Surface Trend in the Data" on page 7574 for a more detailed approach to trend analysis and the issue of anisotropy.

Following the previous exploratory analysis, you then need to classify each data pair as a member of a distance interval (lag). PROC VARIOGRAM performs this grouping with two required options for semivariogram computation: the LAGDISTANCE= and MAXLAGS= options. These options are based on your assessment of how to group the data pairs within distance classes.

The meaning of the required LAGDISTANCE= option is as follows. Classify all pairs of points into intervals according to their pairwise distance. The width of each distance interval is the LAGDISTANCE= value. The meaning of the required MAXLAGS= option is simply the number of intervals you consider. The problem is that given only the scatter plot of the measurement locations, it is not clear what values to give to the LAGDISTANCE= and MAXLAGS= options.

Ideally, you would like a sufficient number of distance classes that capture the extent to which your data are correlated, and that each contain a minimum of data pairs to increase the accuracy in your computations. A rule of thumb used in semivariogram computations is that you should have at least 30 pairs per lag class. This is an empirical arbitrary threshold; see the section "Choosing the Size of Classes" on page 7555 for further details.

In the preliminary analysis, you use the option NHC= in the COMPUTE statement to help you experiment with these numbers and choose values for the LAGDISTANCE= and MAXLAGS= options. Here, in particular, you requested NHC=20 to preview a classification that uses 20 distance classes across your spatial domain. A zero lag class is always considered; therefore the output shows the number of distance classes to be one more than the number you specified.

Based on your selection of the NHC= option, the NOVARIOGRAM option produces a pairwise distances table from your observations shown in Figure 95.3, and the corresponding histogram in Figure 95.4. For illustration purposes, you also specified in the code a threshold of minimum data pairs per distance class in the PAIRS option as THR=30. As a result, a reference line appears in the histogram so that you can visually identify any lag classes with pairs that fall below your specified threshold.

Figure 95.3 Pairwise Distance Intervals Table

Pairwise Distance Intervals

Lag Class	Bounds		Number of Pairs	Percentage of Pairs
0	0.00	3.48	7	0.25%
1	3.48	10.45	81	2.92%
2	10.45	17.42	138	4.97%
3	17.42	24.39	167	6.02%
4	24.39	31.36	204	7.35%
5	31.36	38.33	210	7.57%
6	38.33	45.30	213	7.68%
7	45.30	52.27	253	9.12%
8	52.27	59.24	237	8.54%
9	59.24	66.20	280	10.09%
10	66.20	73.17	252	9.08%
11	73.17	80.14	230	8.29%
12	80.14	87.11	217	7.82%
13	87.11	94.08	154	5.55%
14	94.08	101.05	71	2.56%
15	101.05	108.02	41	1.48%
16	108.02	114.99	14	0.50%
17	114.99	121.96	5	0.18%
18	121.96	128.93	1	0.04%
19	128.93	135.89	0	0.00%
20	135.89	142.86	0	0.00%

The NOVARIOGRAM option also produces a table with useful facts about the pairs and the distances between the most remote data in selected directions, shown in Figure 95.5. In particular, the lag distance value is calculated based on your selection of the NHC= option. The last three table entries report the overall maximum distance among your data pairs, as well as the maximum distances in the main axes directions (that is, the vertical or N–S axis, and the horizontal or E–W axis). This information is also provided in the inset of Figure 95.4. When you specify a threshold in the PAIRS suboption of the PLOTS= option, as in this example, the threshold also appears in the table. Then, the line that follows indicates the highest lag class with the following property: Each one of the distance classes that lie farther away from this lag features a pairs population below the specified threshold.

With the preceding information you can determine appropriate values for the LAGDISTANCE= and MAXLAGS= options in the COMPUTE statement. In particular, the classification that uses 20 distance classes is satisfactory, and you can choose LAGDISTANCE=7 after following the suggestion in Figure 95.5.

Figure 95.4 Distribution of Pairwise Distances

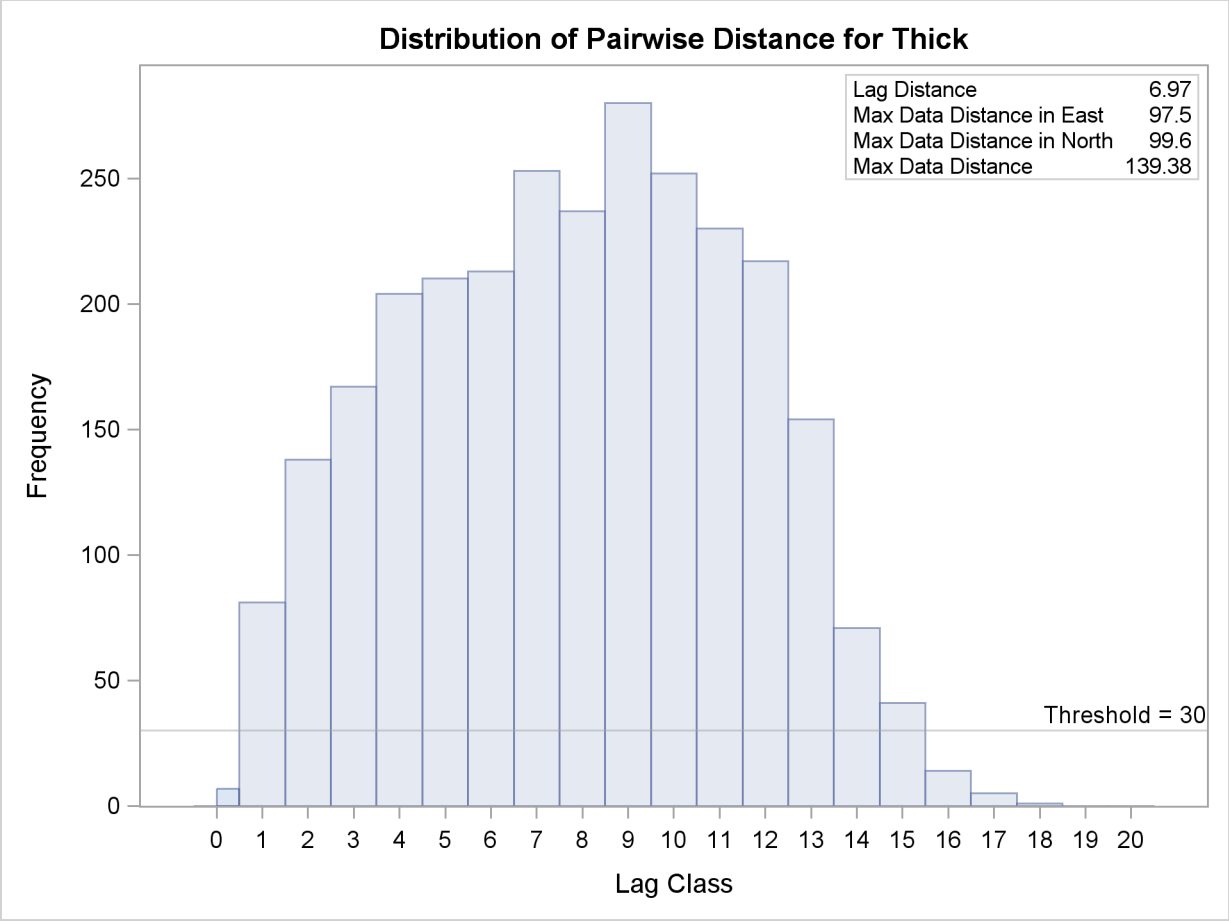


Figure 95.5 Pairs Information Table

Pairs Information

Number of Lags	21
Lag Distance	6.97
Minimum Pairs Threshold	30
Highest Lag With Pairs > Threshold	15
Maximum Data Distance in East	97.50
Maximum Data Distance in North	99.60
Maximum Data Distance	139.38

The MAXLAGS= option needs to be specified based on the spatial extent to which your data are correlated. Unless you know this size, in the present omnidirectional case you can assume the correlation extent to be roughly equal to half the overall maximum distance between data points.

The table in Figure 95.5 suggests that this number corresponds to 139,380 feet, which is most likely on or close to a diagonal direction (that is, the northeast–southwest or northwest–southeast direction). Hence, you can expect the correlation extent in this scale to be around $139.4/2 = 69,700$ feet. Consequently, this is the distance up to which you should consider lag classes for the empirical semivariogram computations. Given your lag size selection, Figure 95.3 indicates that this distance corresponds to about 10 lags; hence you can set MAXLAGS=10.

Overall, for a specific NHC= choice of class count, you can expect your choice of MAXLAGS= to be approximately half the number of the lag classes (see the section "Spatial Extent of the Empirical Semivariogram" on page 7556 for more details).

Once you have starting values for the LAGDISTANCE= and MAXLAGS= options, you can run the VARIOGRAM procedure multiple times to inspect and compare the results you get by specifying different values for these options.

Empirical Semivariogram Computation

Using the values of LAGDISTANCE=7 and MAXLAGS=10 computed previously, rerun PROC VARIOGRAM without the NOVARIOGRAM option in order to compute the empirical semivariogram. You specify the CL option in the COMPUTE statement to calculate the 95% confidence limits for the classical semivariance. The section "COMPUTE Statement" on page 7533 describes how to use the ALPHA= option to specify a different confidence level.

Also, you can request a robust version of the semivariance with the ROBUST option in the COMPUTE statement. PROC VARIOGRAM produces a plot showing both the classical and the robust empirical semivariograms. See the details of the PLOT option to specify different instances of plots of the empirical semivariogram. In addition, ask for the autocorrelation Moran's I and Geary's c statistics under the assumption of randomization using binary weights. The following statements implement all of the preceding requests:

```
proc variogram data=thick outv=outv;
   compute lagd=7 maxlag=10 cl robust
           autocorr(assum=random);
   coordinates xc=East yc=North;
   var Thick;
run;

ods graphics off;
```

Figure 95.6 displays the PROC VARIOGRAM output empirical semivariogram table for the preceding code.

Figure 95.6 Output Table for the Empirical Semivariogram Analysis

The VARIOGRAM Procedure
Dependent Variable: Thick

Empirical Semivariogram

Lag Class	Pair Count	Average Distance	Semivariance Robust	Semivariance Classical	Standard Error	95% Confidence Limits	
0	7	2.64	0.0284	0.0336	0.0179	0.000	0.0687
1	82	7.29	0.2098	0.3937	0.0615	0.273	0.5142
2	138	14.16	1.0079	1.1794	0.1420	0.901	1.4577
3	169	21.08	3.0183	2.7988	0.3045	2.202	3.3956
4	205	27.93	4.8107	4.6024	0.4546	3.711	5.4934
5	213	35.17	5.9904	5.9278	0.5744	4.802	7.0536
6	214	42.20	8.1040	7.5181	0.7268	6.094	8.9426
7	250	48.78	7.5326	7.2210	0.6459	5.955	8.4869
8	247	56.16	8.0662	7.1952	0.6475	5.926	8.4642
9	281	62.89	8.2792	6.8445	0.5774	5.713	7.9763
10	250	69.93	8.1440	6.3577	0.5686	5.243	7.4722

Figure 95.7 shows the output from the requested autocorrelation analysis. This includes the observed (computed) Moran's I and Geary's c coefficients, the expected value and standard deviation for each coefficient, the corresponding Z score, and the p-value in the Pr $>|$ Z $|$ column. The low p-values suggest strong autocorrelation for both statistics types. Note that a two-sided p-value is reported, which is the probability that the observed coefficient lies farther away from $|$ Z $|$ on either side of the coefficient's expected value—that is, lower than $-Z$ or higher than Z. The sign of Z for both Moran's I and Geary's c coefficients indicates positive autocorrelation in the Thick data values; see the section "Interpretation" on page 7564 for more details.

Figure 95.7 Output Table for the Autocorrelation Statistics

Autocorrelation Statistics

Assumption	Coefficient	Observed	Expected	Std Dev	Z	Pr > |Z|
Randomization	Moran's I	1.0833	-0.0135	0.147	7.47	<.0001
Randomization	Geary's c	0.0185	1.0000	0.228	-4.30	<.0001

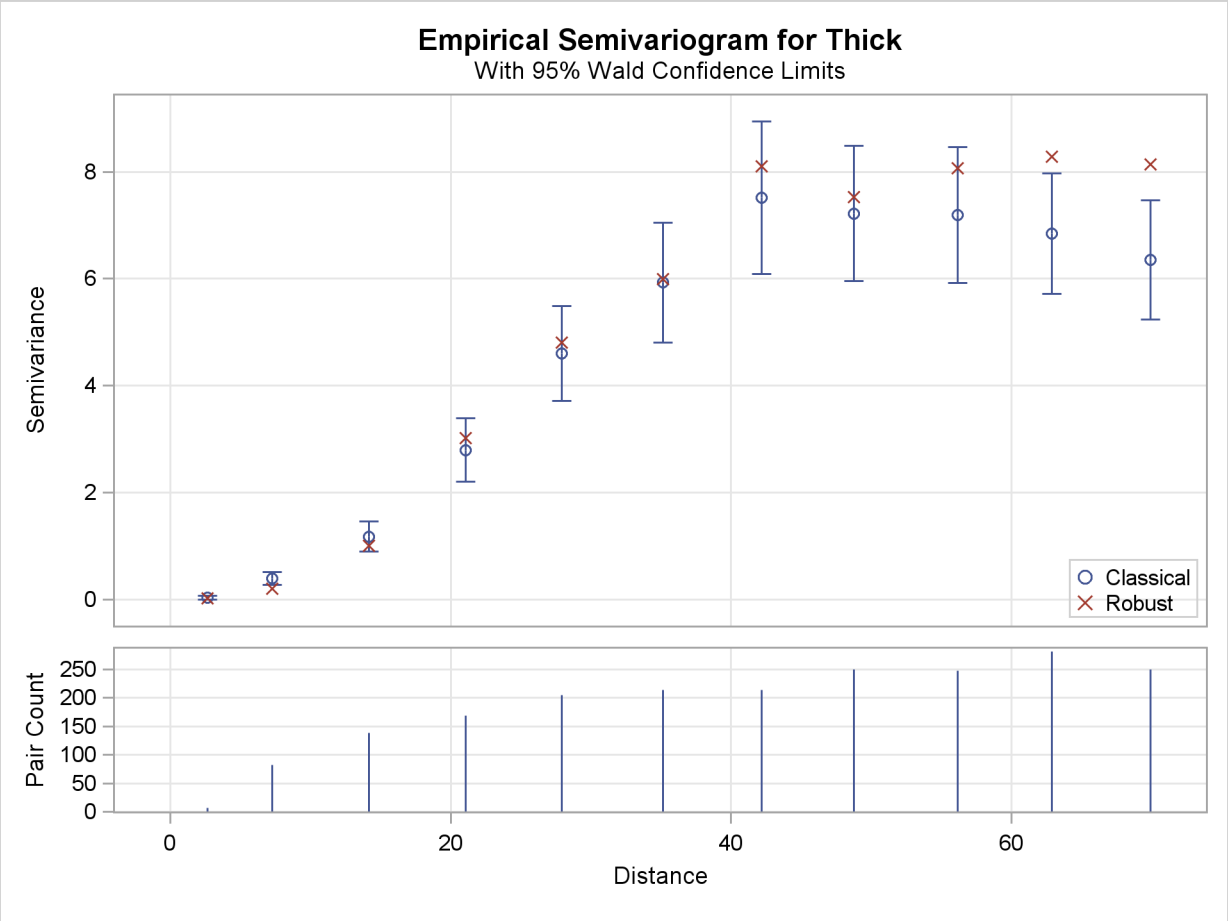
Figure 95.8 shows both the classical and robust empirical semivariograms. In addition, the plot features the approximate 95% confidence limits for the classical semivariance. The figure exhibits a typical behavior of the computed semivariance uncertainty, where in general the variance increases with distance from the origin at Distance=0.

The needle plot in the lower part of the Figure 95.8 provides the number of pairs that were used in the computation of the empirical semivariance for each lag class shown. Note that in general this is a pairwise distribution that is different from the distribution depicted in Figure 95.4. First, the number of pairs shown in the needle plot depends on the particular criteria you specify in the

COMPUTE statement of PROC VARIOGRAM. Second, the distances shown for each lag on the Distance axis are not the midpoints of the lag classes as in the pairwise distances plot, but rather the average distance from the origin Distance=0 of all pairs in a given lag class.

"Example 95.1: Theoretical Semivariogram Model Fitting" on page 7570 continues from this point to show how to choose a theoretical model for the thickness spatial dependence.

Figure 95.8 Classical and Robust Empirical Semivariograms for Coal Seam Thickness Data



Syntax: VARIOGRAM Procedure

The following statements are available in PROC VARIOGRAM:

PROC VARIOGRAM *options* ;
  BY *variables* ;
  COMPUTE *computation-options* ;
  COORDINATES *coordinate-variables* ;
  DIRECTIONS *directions-list* ;
  VAR *analysis-variables-list* ;

The COMPUTE and COORDINATES statements are required.

Table 95.1 outlines the options available in PROC VARIOGRAM classified by function.

Table 95.1 Options Available in the VARIOGRAM Procedure

Task	Statement	Option
Data Set Options		
Specify input data set	PROC VARIOGRAM	DATA=
Suppress normal display of results	PROC VARIOGRAM	NOPRINT
Write autocorrelation weights information	PROC VARIOGRAM	OUTACWEIGHTS=
Write distance histogram information	PROC VARIOGRAM	OUTDISTANCE=
Write pairwise point information	PROC VARIOGRAM	OUTPAIR=
Write spatial continuity measures	PROC VARIOGRAM	OUTVAR=
Specify plot display and options	PROC VARIOGRAM	PLOTS
Declaring the Role of Variables		
Specify variables to define analysis subgroups	BY	
Specify the analysis variables	VAR	
Specify the *x*, *y* coordinates in the DATA= data set	COORDINATES	XCOORD= YCOORD=
Controlling Continuity Measure Computations		
Specify the confidence level	COMPUTE	ALPHA=
Specify the angle tolerances for angle classes	COMPUTE	ANGLETOL=
Compute autocorrelation statistics	COMPUTE	AUTOCORRELATION
Specify the bandwidths for angle classes	COMPUTE	BANDWIDTH=
Compute semivariance estimate variance	COMPUTE	CL
Specify the minimum distance that indicates any two distinct points are not collocated	COMPUTE	DEPSILON=
Specify the basic lag distance	COMPUTE	LAGDISTANCE=
Specify the tolerance around the lag distance	COMPUTE	LAGTOLERANCE=
Specify the maximum number of lags in computations	COMPUTE	MAXLAGS=
Specify the number of angle classes	COMPUTE	NDIRECTIONS=
Suppress computation of all continuity measures	COMPUTE	NOVARIOGRAM
Compute robust semivariance	COMPUTE	ROBUST
Controlling Distance Histogram Data Set		
Specify the distance histogram data set	PROC VARIOGRAM	OUTDISTANCE=
Specify the number of histogram classes	COMPUTE	NHCLASSES=
Controlling Pairwise Information Data Set		
Specify the pairwise data set	PROC VARIOGRAM	OUTPAIR=
Specify the maximum distance for the pairwise data set	COMPUTE	OUTPDISTANCE=

PROC VARIOGRAM Statement

PROC VARIOGRAM *options* **;**

You can specify the following options in the PROC VARIOGRAM statement.

DATA=*SAS-data-set*
specifies a SAS data set containing the x and y coordinate variables and the VAR statement variables.

NOPRINT
suppresses the normal display of results. The NOPRINT option is useful when you want only to create one or more output data sets with the procedure. Note that this option temporarily disables the Output Delivery System (ODS); see the section "ODS Graphics" on page 7569 for more information.

OUTACWEIGHTS=*SAS-data-set*
OUTACW=*SAS-data-set*
OUTA=*SAS-data-set*
specifies a SAS data set in which to store the autocorrelation weights information for each pair of points in the DATA= data set. This option should be used with caution when the DATA= data set is large. If n denotes the number of observations in the DATA= data set, the OUTACWEIGHTS= data set contains $[n(n-1)]/2$ observations.

See the section "OUTACWEIGHTS=*SAS-data-set*" on page 7565 for details.

OUTDISTANCE=*SAS-data-set*
OUTDIST=*SAS-data-set*
OUTD=*SAS-data-set*
specifies a SAS data set in which to store summary distance information. This data set contains a count of all pairs of data points within a given distance interval. The number of distance intervals is controlled by the NHCLASSES= option in the COMPUTE statement. The OUTDISTANCE= data set is useful for plotting modified histograms of the count data for determining appropriate lag distances.

See the section "OUTDIST=*SAS-data-set*" on page 7566 for details.

OUTPAIR=*SAS-data-set*
OUTP=*SAS-data-set*
specifies a SAS data set in which to store distance and angle information for each pair of points in the DATA= data set. This option should be used with caution when the DATA= data set is large. If n denotes the number of observations in the DATA= data set, the OUTPAIR= data set contains $[n(n-1)]/2$ observations unless you restrict it with the OUTPDISTANCE= option in the COMPUTE statement. The OUTPDISTANCE= option in the COMPUTE statement excludes pairs of points when the distance between the pairs exceeds the OUTPDISTANCE= value.

See the section "OUTPAIR=*SAS-data-set*" on page 7566 for details.

OUTVAR=*SAS-data-set*

OUTVR=*SAS-data-set*

specifies a SAS data set in which to store the continuity measures.

See the section "OUTVAR=*SAS-data-set*" on page 7567 for details.

PLOTS <(*global-plot-options***) > < = *plot-request* < *(options)* > >**

PLOTS <(*global-plot-options***) > < = (***plot-request* < *(options)* > **< ...** *plot-request* < *(options)* > **>) >**

controls the plots produced through ODS Graphics. When you specify only one plot request, you can omit the parentheses around the plot request. Here are some examples:

```
plots=none
plots=observ
plots=(observ semivar)
plots(unpack)=semivar
plots=(semivar(cla unpack) semivar semivar(rob))
```

You must enable ODS Graphics before requesting plots, for example, like this:

```
ods graphics on;

proc variogram data=thick;
   compute novariogram;
   coordinates xc=East yc=North;
   var Thick;
run;

ods graphics off;
```

For general information about ODS Graphics, see Chapter 21, "Statistical Graphics Using ODS." If you have enabled ODS Graphics but do not specify the PLOTS= option or have specified PLOTS=ALL, then PROC VARIOGRAM produces a default set of plots, which might be different for different COMPUTE statement options, as discussed in the following.

- If you specify NOVARIOGRAM in the COMPUTE statement, the VARIOGRAM procedure produces a scatter plot of your observations spatial distribution, as well as the histogram of the pairwise distances of your data. For an example of the observations plot, see Figure 95.2. For an example of the pairwise distances plot, see Figure 95.4.
- If you do not specify NOVARIOGRAM in the COMPUTE statement, the VARIOGRAM procedure computes the empirical semivariogram for the specified LAGDISTANCE= and MAXLAGS= options. The observations plot appears by default in this case, too. The VARIOGRAM procedure also produces a plot of the classical empirical semivariogram. If you further specify ROBUST in the COMPUTE statement, then the VARIOGRAM procedure instead produces a plot of both the classical and robust empirical semivariograms, in addition to the observations plot. For an example of the empirical semivariograms plot, see Output 95.3.4.

The *global-plot-options* include the following:

ONLY

suppresses the default plots. Only plots specifically requested are displayed.

UNPACKPANEL

UNPACK

suppresses paneling. By default, multiple plots can appear in some output panels. Specify UNPACKPANEL to get each plot in a separate panel. You can specify PLOTS(UNPACKPANEL) to unpack the default plots. You can also specify UNPACKPANEL as a suboption with the SEMIVAR option.

The following individual *plot-requests* and plot *options* are available:

ALL

produces all appropriate plots. You can specify other options with ALL. For example, to request all default plots and an additional classical empirical semivariogram, specify PLOTS=(ALL SEMIVAR(CLA)).

EQUATE

specifies that all appropriate plots be produced in a way that the axes coordinates have equal size units.

NONE

suppresses all plots.

OBSERVATIONS < (*observations-plot-options*) >

OBSERV < (*observations-plot-options*) >

OBS < (*observations-plot-options*) >

produces the observed data plot. Only one observations plot will be created if you specify the OBSERVATIONS option more than once within a PLOTS option.

The OBSERVATIONS option has the following suboptions:

GRADIENT

specifies that observations be displayed as circles colored by the observed measurement.

OUTLINE

specifies that observations be displayed as circles with a border but with a completely transparent fill.

OUTLINEGRADIENT

is the same as OBSERVATIONS(GRADIENT) except that a border is shown around each observation.

SHOWMISSING

specifies that observations with missing values be displayed in addition to the observations with nonmissing values. By default, missing values locations are not shown on the plot. If you specify multiple instances of the OBSERVATIONS option, and you specify the SHOWMISSING suboption in any of those, then the resulting observations plot will display the observations with missing values.

For the GRADIENT, OUTLINE, and OUTLINEGRADIENT suboptions: The OUTLINEGRADIENT is the default suboption if you do not specify any of those three. If you specify multiple instances of the OBSERVATIONS option or multiple suboptions for OBSERVATIONS, then the resulting observations plot will honor the last specified GRADIENT, OUTLINE, or OUTLINEGRADIENT suboption.

PAIRS < (*pairs-plot-options*) >
specifies that the pairwise distances histogram be produced. By default, the horizontal axis displays the lag class number. The vertical axis shows the frequency (count) of pairs in the lag classes. Notice that the zero lag class width is half the width of the other classes.

The PAIRS option has the following suboptions:

MIDPOINT
MID
specifies that the plot created with the PAIRS option display the lag class midpoint value on the horizontal axis, rather than the default lag class number. The midpoint value is the actual distance of a lag class center from the assumed origin point at distance zero (see also the illustration in Figure 95.14).

NOINSET
NOI
specifies that the plot created with the PAIRS option be produced without the default inset that provides additional information about the pairs distribution.

THRESHOLD=*minimum pairs*
THR=*minimum pairs*
specifies that a reference line appear in the plot created with the PAIRS option to indicate the *minimum pairs* frequency of data pairs. You can use this line as an exploratory tool when you want to select lag classes that contain at least THRESHOLD point pairs. The option helps you to identify visually any portion of the PAIRS distribution that lies below the specified THRESHOLD value.

Only one pairwise distances histogram will be created if you specify the PAIRS option within a PLOTS option. If you specify multiple instances of the PAIRS option, the resulting plot will have the following features:

- If the MIDPOINT or NOINSET suboption has been specified in any of the instances, it will be activated in the resulting plot.
- If you have specified the THRESHOLD= suboption more than once, then the THRESHOLD= value specified last will prevail.

SEMIVARIOGRAM < (*semivar-plot-options*) >
SEMIVAR < (*semivar-plot-options*) >
specifies that the empirical semivariogram plot be produced. You can specify the SEMIVAR option multiple times in the same PLOTS option to request instances of plots with the following *semivar-plot-options*:

ALL | CLASSICAL | ROBUST

ALL | CLA | ROB

specifies a single type of empirical semivariogram (classical or robust) to plot, or specifies that all the available types be included in the same plot. The default is ALL.

UNPACKPANEL

UNPACK

specifies that paneled semivariogram plots be displayed separately. By default, plots appear in a panel, when appropriate.

BY Statement

BY *variables* **;**

You can specify a BY statement with PROC VARIOGRAM to obtain separate analyses on observations in groups defined by the BY variables. When a BY statement appears, the procedure expects the input data set to be sorted in order of the BY variables. The *variables* are one or more variables in the input data set.

If your input data set is not sorted in ascending order, use one of the following alternatives:

- Sort the data by using the SORT procedure with a similar BY statement.
- Specify the BY statement options NOTSORTED or DESCENDING in the BY statement for the VARIOGRAM procedure. The NOTSORTED option does not mean that the data are unsorted but rather that the data are arranged in groups (according to values of the BY variables) and that these groups are not necessarily in alphabetical or increasing numeric order.
- Create an index on the BY variables by using the DATASETS procedure (in Base SAS software).

For more information about the BY statement, see *SAS Language Reference: Concepts*. For more information about the DATASETS procedure, see the *Base SAS Procedures Guide*.

COMPUTE Statement

COMPUTE *computation-options* **;**

The COMPUTE statement provides a number of options that control the computation of the semivariance, the robust semivariance, and the covariance.

ALPHA=*number*

specifies a parameter to obtain the confidence level for constructing confidence limits in the classical empirical semivariance estimation. The value of *number* must be between 0 and 1, and the confidence level is $1-number$. The default is ALPHA=0.05, which corresponds to the default confidence level of 95%. If the CL option is not specified, ALPHA= is ignored.

ANGLETOLERANCE=*angle tolerance*

ANGLETOL=*angle tolerance*

ATOL=*angle tolerance*

specifies the tolerance, in degrees, around the angles determined by the NDIRECTIONS= specification. The default is $180^\circ/(2n_d)$, where n_d is the NDIRECTIONS= specification. If you do not specify the NDIRECTIONS= option or the DIRECTIONS statement, ANGLETOLERANCE= is ignored.

See the section "Theoretical and Computational Details of the Semivariogram" on page 7544 for further information.

Experimental

AUTOCORRELATION < (*autocorrelation-options***) >**

AUTOCORR < (*autocorrelation-options***) >**

AUTOC < (*autocorrelation-options***) >**

specifies that autocorrelation statistics be calculated. You can further specify the following *autocorrelation-options* in parentheses following the experimental AUTOCORRELATION option.

ASSUMPTION < = *assumption-options* >

ASSUM < = *assumption-options* >

specifies the type of autocorrelation assumption to use. The *assumption-options* can be one of the following:

NORMALITY | NORMAL | NOR

specifies use of the normality assumption.

RANDOMIZATION | RANDOM | RAN

specifies use of the randomization assumption.

The default is ASSUMPTION=NORMALITY.

STATISTICS < = (*stats-options*) >

STATS < = (*stats-options*) >

specifies the autocorrelation statistics in detail. The *stats-options* can be one or more of the following:

ALL

applies all available types of autoregression statistics.

GEARY | GEA

specifies use of the Geary c statistics.

MORAN | MOR

specifies use of the Moran I statistics.

The default is STATISTICS=ALL.

WEIGHTS < = *weights-options* >

WEI < = *weights-options* >

specifies the scheme used for the computation of the autocorrelation weights. You can choose one of the following *weights-options*:

BINARY < (*binary-option*) >

specifies that binary weights be used. You also have the following *binary-option*:

ROWAVERAGING | ROWAVG | ROW

specifies that asymmetric autocorrelation weights be assigned to data pairs. For each observation, if there are nonzero weights, the ROWAVG option standardizes those weights so that they sum to 1. No row averaging is performed by default.

DISTANCE < (*distance-options*) >

specifies that autocorrelation weights be assigned based on the point pair distances. You also have the following *distance-options*:

NORMALIZE | NORMAL | NOR

specifies that normalized pair distances be used in the distance-based weights expression. The distances are normalized with respect to the maximum pairwise distance h_b, as it is defined in the section "Computation of the Distribution Distance Classes" on page 7553. By default, nonnormalized values are used in the computations.

POWER | POW

specifies the power to which the pair distance is raised in the distance-based weights expression. POWER is a nonnegative number, and its default value is POWER=1.

ROWAVERAGING | ROWAVG | ROW

specifies that asymmetric autocorrelation weights be assigned to data pairs. For each observation, if there are nonzero weights, the ROWAVG option standardizes those weights so that they sum to 1. No row averaging is performed by default.

SCALE | SCA

specifies the scaling factor in the distance-based weights expression. SCALE is a nonnegative number, and its default value is SCALE=1.

The default is WEIGHTS=BINARY. See the section "Autocorrelation Statistics (Experimental)" on page 7560 for further details about the autocorrelation weights.

When you specify the AUTOCORRELATION option with no *autocorrelation-options*, PROC VARIOGRAM computes by default both the Moran I and Geary c statistics with p-values computed under the normality assumption with binary weights.

If you specify more than one ASSUMPTION in the *autocorrelation-options*, all but the last specified ASSUMPTION will be ignored. The same holds if you specify more than one POWER= or SCALE= parameter in the WEIGHT=DISTANCE *distance-options*.

If you specify the WEIGHT=BINARY option in the AUTOCORRELATION option and the NOVARIOGRAM option at the same time, then you must also specify the LAGDISTANCE= option in the COMPUTE statement. See the section "Autocorrelation Weights" on page 7560 for more information.

BANDWIDTH=*bandwidth distance*

BANDW=*bandwidth distance*

specifies the bandwidth, or perpendicular distance cutoff for determining the angle class for a given pair of points. The distance classes define a series of cylindrically shaped areas, while the angle classes radially cut these cylindrically shaped areas. For a given angle class $(\theta_1 - \delta\theta_1, \theta_1 + \delta\theta_1)$, as you proceed out radially, the area encompassed by this angle class becomes larger. The BANDWIDTH= option restricts this area by excluding all points with a perpendicular distance from the line $\theta = \theta_1$ that is greater than the BANDWIDTH= value. See Figure 95.15 for a visual representation of the bandwidth.

If you do not specify the BANDWIDTH= option, no restriction occurs. If you do not specify the NDIRECTIONS= option or the DIRECTIONS statement, BANDWIDTH= is ignored.

CL

requests confidence limits for the classical semivariance estimate. The confidence limits' lower bound is always nonnegative, adhering to the behavior of the theoretical semivariance. You can control the confidence level with the ALPHA= option.

DEPSILON=*distance value*

DEPS=*distance value*

specifies the distance value for declaring that two distinct points are zero distance apart. Such pairs, if they occur, cause numeric problems. If you specify DEPSILON=$\Delta\varepsilon$, then pairs of points P_1 and P_2 for which the distance between them $\mid P_1P_2 \mid < \Delta\varepsilon$ are excluded from the continuity measure calculations. The default value of the DEPSILON= option is 100 times the machine precision; this product is approximately 1E–10 on most computers.

LAGDISTANCE=*distance unit*

LAGDIST=*distance unit*

LAGD=*distance unit*

specifies the basic distance unit defining the lags. For example, a specification of LAGDISTANCE=x results in lag distance classes that are multiples of x. For a given pair of points P_1 and P_2, the distance between them, denoted $\mid P_1P_2 \mid$, is calculated. If $\mid P_1P_2 \mid = x$, then this pair is in the first lag class. If $\mid P_1P_2 \mid = 2x$, then this pair is in the second lag class, and so on.

For irregularly spaced data, the pairwise distances are unlikely to fall exactly on multiples of the LAGDISTANCE= value. A distance tolerance of δx is used to accommodate a spread of distances around multiples of x (the LAGTOLERANCE= option specifies the distance tolerance). For example, if $\mid P_1P_2 \mid$ is within $x \pm \delta x$, you would place this pair in the first lag class; if $\mid P_1P_2 \mid$ is within $2x \pm \delta x$, you would place this pair in the second lag class, and so on.

You can experiment and determine the candidate values for the LAGDISTANCE= option by plotting the pairwise distance histogram for different numbers of histogram classes, using the NHC= option.

A LAGDISTANCE= value is required for the semivariance and the autocorrelation computations. You need not specify LAGDISTANCE= only when you specify the NOVARIOGRAM option without the AUTOCORRELATION option.

See the section "Theoretical and Computational Details of the Semivariogram" on page 7544 for more information.

LAGTOLERANCE=*tolerance number*

LAGTOL=*tolerance number*

LAGT=*tolerance number*

specifies the tolerance around the LAGDISTANCE= value for grouping distance pairs into lag classes. See the description of the LAGDISTANCE= option for information about the use of the LAGTOLERANCE= option, and the section "Theoretical and Computational Details of the Semivariogram" on page 7544 for more details.

If you do not specify the LAGTOLERANCE= option, a default value of $\frac{1}{2}$ times the LAGDISTANCE= value is used.

MAXLAGS=*number of lags*

MAXLAG=*number of lags*

MAXL=*number of lags*

specifies the maximum number of lag classes used in constructing the continuity measures. This option excludes any pair of points P_1 and P_2 for which the distance between them, $| P_1 P_2 |$, exceeds the MAXLAGS= value times the LAGDISTANCE= value.

You can determine candidate values for the MAXLAGS= option by plotting or displaying the OUTDISTANCE= data set.

A MAXLAGS= value is required unless you specify the NOVARIOGRAM option.

NDIRECTIONS=*number of directions*

NDIR=*number of directions*

ND=*number of directions*

specifies the number of angle classes to use in computing the continuity measures. This option is useful when there is potential anisotropy in the spatial continuity measures. Anisotropy is a field property where the characterization of spatial continuity depends on the data pair orientation (or angle between the N–S direction and the axis defined by the data pair). Isotropy is the absence of this effect; that is, the description of spatial continuity depends only on the distance between the points, not the angle.

The angle classes formed from the NDIRECTIONS= option start from N–S and proceed clockwise. For example, NDIRECTIONS=3 produces three angle classes. In terms of compass points, these classes are centered at 0° (or its reciprocal, 180°), 60° (or its reciprocal, 240°), and 120° (or its reciprocal, 300°). For irregularly spaced data, the angles between pairs are unlikely to fall exactly in these directions, so an angle tolerance of $\delta\theta$ is used (the ANGLETOLERANCE= option specifies the angle tolerance). If NDIRECTIONS=n_d, the base angle is $\theta = 180°/n_d$, and the angle classes are

$$(k\theta - \delta\theta, k\theta + \delta\theta) \quad k = 0, \ldots, n_d - 1$$

If you do not specify the NDIRECTIONS= option, no angles are formed. This is the omnidirectional case where the spatial continuity measures are assumed to be isotropic.

The NDIRECTIONS= option is useful for exploring possible anisotropy. The DIRECTIONS statement, described in the section "DIRECTIONS Statement" on page 7539, provides greater control over the angle classes.

See the section "Theoretical and Computational Details of the Semivariogram" on page 7544 for more information.

NHCLASSES=*number of histogram classes*

NHCLASS=*number of histogram classes*

NHC=*number of histogram classes*

specifies the number of distance classes to consider in the spatial domain in the exploratory stage of the empirical semivariogram computation. The actual number of classes is one more than the NHCLASSES= value, since a special lag zero class is also computed. The NHC= option is used to produce the distance intervals table, the histogram of pairwise distances, and the OUTDISTANCE= data set. See the OUTDISTANCE= option as well as the section "OUTDIST=*SAS-data-set*" on page 7566 and the section "Theoretical and Computational Details of the Semivariogram" on page 7544 for more information.

The default value of the NHCLASSES= option is 10.

NOVARIOGRAM

prevents the computation of the continuity measures. This option is useful for preliminary analysis, or when you require only the OUTDISTANCE= or OUTPAIR= data sets.

OUTPDISTANCE=*distance limit*

OUTPDIST=*distance limit*

OUTPD=*distance limit*

specifies the cutoff distance for writing observations to the OUTPAIR= data set. If you specify OUTPDISTANCE=d_{max}, the distance $\mid P_1 P_2 \mid$ between each pair of points P_1 and P_2 is checked against d_{max}. If $\mid P_1 P_2 \mid > d_{max}$, the observation for this pair is not written to the OUTPAIR= data set. If you do not specify the OUTPDISTANCE= option, all distinct pairs are written. This option is ignored if you do not specify an OUTPAIR= data set.

ROBUST

requests that a robust version of the semivariance be calculated in addition to the classical semivariance.

COORDINATES Statement

COORDINATES *coordinate-variables* **;**

The following two options give the names of the variables in the DATA= data set containing the values of the x and y coordinates of the data.

Only one COORDINATES statement is allowed, and it is applied to all the analysis variables. In other words, it is assumed that all the VAR variables have the same x and y coordinates.

XCOORD=*(variable-name)*

XC=*(variable-name)*

X=*(variable-name)*

gives the name of the variable containing the x coordinate of the data in the DATA= data set.

YCOORD=*(variable-name)*

YC=*(variable-name)*

Y=*(variable-name)*

gives the name of the variable containing the y coordinate of the data in the DATA= data set.

DIRECTIONS Statement

DIRECTIONS *directions-list* **;**

You use the DIRECTIONS statement to define angle classes. You can specify angle classes as a list of angles, separated by commas, with optional angle tolerances and bandwidths within parentheses following the angle. You must specify at least one angle.

If you do not specify the optional angle tolerance, the default value of 45° is used. If you do not specify the optional bandwidth, no bandwidth is checked. If you specify a bandwidth, you must also specify an angle tolerance.

For example, suppose you want to compute three separate semivariograms at angles $\theta_1 = 0°, \theta_2 = 60°$, and $\theta_3 = 120°$, with corresponding angle tolerances $\delta\theta_1 = 22.5°, \delta\theta_2 = 12.5°$, and $\delta\theta_3 = 22.5°$, with bandwidths 50 and 40 distance units on the first two angle classes and no bandwidth check on the last angle class.

The appropriate DIRECTIONS statement is as follows:

```
directions 0.0(22.5,50), 60.0(12.5,40),120(22.5);
```

VAR Statement

VAR *analysis-variables-list* **;**

Use the VAR statement to specify the analysis variables. You can specify only numeric variables. If you do not specify a VAR statement, all numeric variables in the DATA= data set that are not in the COORDINATES statement are used.

Details: VARIOGRAM Procedure

Theoretical Semivariogram Models

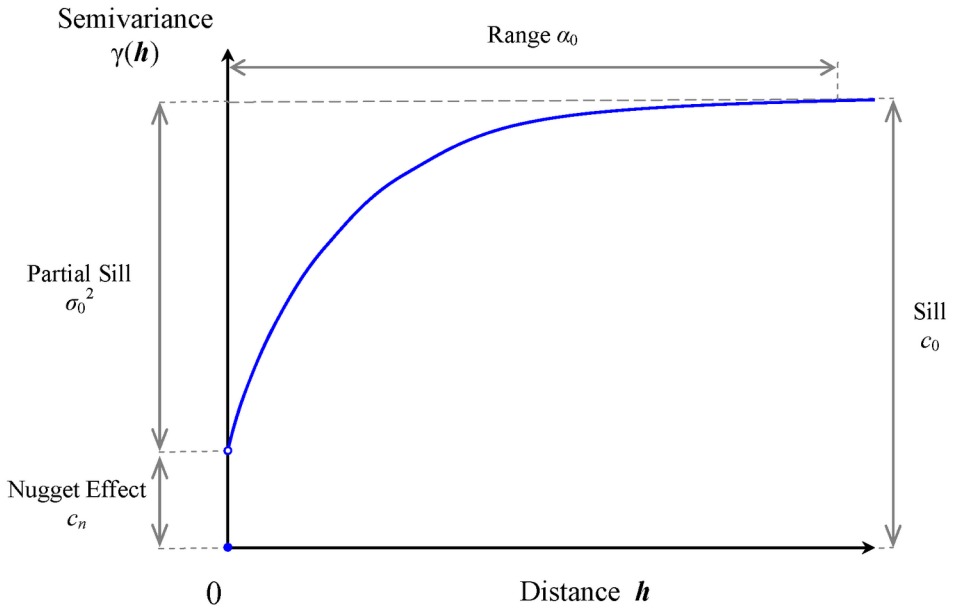
The VARIOGRAM procedure computes the empirical (also known as *sample* or *experimental*) semivariogram from a set of point measurements. Semivariograms are used in the first steps of spatial prediction as tools that provide insight into the spatial continuity and structure of a random process. Naturally occurring randomness is accounted for by describing a process in terms of the *spatial random field* (SRF) concept (Christakos 1992). An SRF is a collection of random variables throughout your spatial domain of prediction. For some of them you already have measurements, and your data set constitutes part of a single realization of this SRF. Based on your sample, spatial prediction aims to provide you with values of the SRF at locations where no measurements are available.

Prediction of the SRF values at unsampled locations by techniques such as ordinary kriging requires the use of a theoretical semivariogram or covariance model. Due to the randomness involved in stochastic processes, the theoretical semivariance cannot be computed. Instead, it is possible that the empirical semivariance can provide an estimate of the theoretical semivariance, which can then be used to characterize the spatial structure of the process.

It is critical to note that the empirical semivariance provides an estimate of its theoretical counterpart only when the SRF satisfies stationarity conditions. These conditions imply that the SRF has a constant (or zero) expected value. Consequently, your data need to be sampled from a trend-free random field and need to have a constant mean, as assumed in "Example 95.1: Theoretical Semivariogram Model Fitting" on page 7570. Equivalently, your data could be residuals of an initial sample that has had a surface trend removed, as portrayed in "Example 95.2: An Anisotropic Case Study with Surface Trend in the Data" on page 7574. For a closer look at stationarity, see the section "Stationarity" on page 7546. For details about different stationarity types and conditions see, for example, Chilès and Delfiner (1999, Section 1.1.4).

When you obtain a valid empirical estimate of the theoretical semivariance, it is then necessary to choose a type of theoretical semivariogram model based on that estimate. Commonly used theoretical semivariogram shapes rise monotonically as a function of distance. The shape is typically characterized in terms of particular parameters; these are the *range* a_0, the *sill* (or *scale*) c_0, and the *nugget effect* c_n. Figure 95.9 displays a theoretical semivariogram of a spherical semivariance model, and points out the semivariogram characteristics.

Figure 95.9 A Theoretical Semivariogram of Spherical Type and Its Characteristics



Specifically, the sill is the semivariogram upper bound. The range a_0 denotes the distance at which the semivariogram reaches the sill. When the semivariogram increases asymptotically toward its sill value, as occurs in the exponential and Gaussian semivariogram models, the term *effective* (or *practical*) range is also used. The effective range r_ϵ is defined as the distance at which the semivariance value achieves 95% of the sill. In particular, for these models the relationship between the range and effective range is $r_\epsilon = 3a_0$ (exponential model) and $r_\epsilon = \sqrt{3}a_0$ (Gaussian model).

The nugget effect c_n represents a discontinuity of the semivariogram that can be present at the origin. It is typically attributed to microscale effects or measurement errors. The semivariance is always 0 at distance $\boldsymbol{h} = 0$; hence, the nugget effect demonstrates itself as a jump in the semivariance as soon as $\boldsymbol{h} > 0$ (note in Figure 95.9 the discontinuity of the function at $\boldsymbol{h} = 0$ in the presence of a nugget effect).

The sill c_0 comprises the nugget effect, if present, and the *partial sill* ${\sigma_0}^2$; that is, $c_0 = c_n + {\sigma_0}^2$. If the SRF $Z(\boldsymbol{s})$ is second-order stationary (see the section "Stationarity" on page 7546), the estimate of the sill is an estimate of the constant variance $\mathrm{Var}[Z(\boldsymbol{s})]$ of the field. Nonstationary processes have variances that depend on the location $\boldsymbol{s}$. Their semivariance increases with distance, hence their semivariograms do not have a sill.

Not every function is a suitable candidate for a theoretical semivariogram model. The semivariance function $\gamma_z(\boldsymbol{h})$, as defined in the following section, is a so-called *conditionally negative-definite* function that satisfies (Cressie 1993, p. 60)

$$\sum_{i=1}^{m}\sum_{j=i}^{m} q_i q_j \gamma_z(\boldsymbol{s}_i - \boldsymbol{s}_j) \leq 0$$

for any number m of locations $\boldsymbol{s}_i$, $\boldsymbol{s}_j$ in $\mathcal{R}^2$ with $\boldsymbol{h} = \boldsymbol{s}_i - \boldsymbol{s}_j$, and any real numbers q_i such that $\sum_{i=1}^{m} q_i = 0$. Permissible, commonly used theoretical semivariogram models include the ones shown in Table 95.2.

Table 95.2 Some Permissible Theoretical Semivariogram Models ($a_0 > 0$)

Model Type	Semivariance
Exponential	$\gamma_z(\boldsymbol{h}) = \begin{cases} 0 & , \text{if } \mid \boldsymbol{h} \mid = 0 \\ c_n + {\sigma_0}^2 \left[1 - \exp\left(-\frac{\mid \boldsymbol{h} \mid}{a_0}\right)\right] & , \text{if } 0 < \mid \boldsymbol{h} \mid \end{cases}$
Gaussian	$\gamma_z(\boldsymbol{h}) = \begin{cases} 0 & , \text{if } \mid \boldsymbol{h} \mid = 0 \\ c_n + {\sigma_0}^2 \left[1 - \exp\left(-\frac{\mid \boldsymbol{h} \mid^2}{a_0^2}\right)\right] & , \text{if } 0 < \mid \boldsymbol{h} \mid \end{cases}$
Power	$\gamma_z(\boldsymbol{h}) = \begin{cases} 0 & , \text{if } \mid \boldsymbol{h} \mid = 0 \\ c_n + {\sigma_0}^2 \boldsymbol{h}^{a_0} & , \text{if } 0 < \mid \boldsymbol{h} \mid \end{cases}$
Spherical	$\gamma_z(\boldsymbol{h}) = \begin{cases} 0 & , \text{if } \mid \boldsymbol{h} \mid = 0 \\ c_n + {\sigma_0}^2 \left[\frac{3}{2}\frac{\mid \boldsymbol{h} \mid}{a_0} - \frac{1}{2}\left(\frac{\mid \boldsymbol{h} \mid}{a_0}\right)^3\right] & , \text{if } 0 < \mid \boldsymbol{h} \mid \leq a_0 \\ c_0, & , \text{if } a_0 < \mid \boldsymbol{h} \mid \end{cases}$

You can review these models in further detail in the section "Theoretical Semivariogram Models" on page 2947 in the KRIGE2D procedure documentation.

The theoretical semivariogram models are used to describe the spatial structure of random processes. Based on their shape and characteristics, the semivariograms of these models can provide a plethora of information (Christakos 1992, Section 7.3):

- Examination of the semivariogram variation in different directions provides information about the isotropy of the random process (see also the discussion about isotropy in the following section).
- The semivariogram range determines the zone of influence extending from any given location. Values at surrounding locations within this zone are correlated with the value at the specific location by means of the particular semivariogram.
- The semivariogram behavior at large distances indicates the degree of stationarity of the process. In particular, an asymptotic behavior suggests a stationary process, whereas either a linear increase and slow convergence to the sill or a fast increase is an indicator of nonstationarity.
- The semivariogram behavior close to the origin indicates the degree of regularity of the process variation. Specifically, a parabolic behavior at the origin implies a very regular spatial variation, whereas a linear behavior characterizes a nonsmooth process. The presence of a nugget effect is additional evidence of irregularity in the process.
- The semivariogram behavior within the range provides description of potential periodicities or anomalies in the spatial process.

A brief note on terminology: In some fields (for example, geostatistics) the term homogeneity is sometimes used instead of stationarity in spatial analysis, whereas in statistics homogeneity is defined differently (Banerjee, Carlin, and Gelfand 2004, Section 2.1.3). In particular, the alternative terminology characterizes as homogeneous the stationary SRF in $\mathcal{R}^n, n > 1$, whereas it retains the term stationary for such SRF in $\mathcal{R}^1$ (SRF in $\mathcal{R}^1$ are also known as *random processes*). Often, studies in a single dimension refer to temporal processes; hence, you might see time-stationary random processes called "temporally stationary" or simply stationary, and stationary SRF in $\mathcal{R}^n, n > 1$, characterized as "spatially homogeneous" or simply homogeneous. This distinction made by the alternative nomenclature is more evident in spatiotemporal random fields (S/TRF), where the different terms clarify whether stationarity applies in the spatial or the temporal part of the S/TRF.

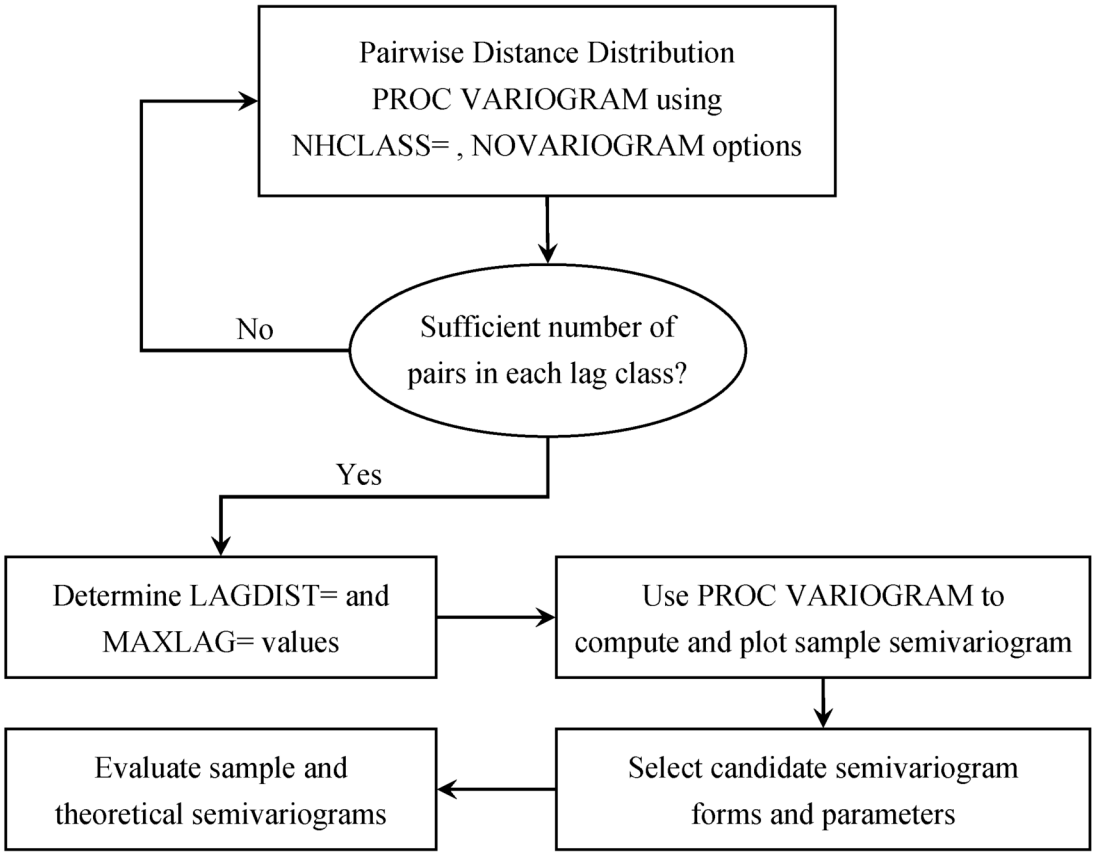
Typically, you choose a theoretical semivariogram model to fit the empirical semivariance in an automated manner. For this task you can use methods such as least squares, maximum likelihood, and robust methods (Cressie 1993, Section 2.6). "Example 95.1: Theoretical Semivariogram Model Fitting" on page 7570 illustrates the fitting process by using ordinary and weighted least squares methods. A different approach is manual fitting, where a theoretical semivariogram model is chosen based on visual inspection of the empirical semivariogram; see, for example, Hohn (1988, p. 25).

In some cases, you might see that using a combination of theoretical models results in a more accurate fit onto the empirical semivariance than using a single model. This is known as model nesting. Nested models, anisotropic models, and the nugget effect increase the scope of theoretical models available. All these concepts are discussed in the section "Theoretical Semivariogram Models" on page 2947 in the KRIGE2D procedure documentation.

Overall, Goovaerts (1997, Section 4.2.4) suggests that fitting a theoretical model should aim to capture the major spatial features. An accurate fit is desirable, but overfitting does not offer advantages, because you might find yourself trying to model possibly spurious details of the empirical semivariogram.

Note the general flow of investigation. The empirical semivariogram is computed after a suitable choice is made for the LAGDISTANCE= and MAXLAGS= options. For computations in more than one directions you can further use the NDIR= option or the DIRECTIONS statement. Potential theoretical models (which can also incorporate nesting, anisotropy, and the nugget effect) are then plotted against the empirical semivariogram and evaluated. The flow of this analytical process is illustrated in Figure 95.10. After a suitable theoretical model is determined, it is used in PROC KRIGE2D for the prediction stage. The prediction analysis is presented in detail in the section "Details of Ordinary Kriging" on page 2962 in the KRIGE2D procedure documentation.

Figure 95.10 Flowchart for Variogram Selection



Theoretical and Computational Details of the Semivariogram

Let $\{Z(\boldsymbol{s}), \boldsymbol{s} \in D \subset \mathcal{R}^2\}$ be a spatial random field (SRF) with n measured values $z_i = Z(\boldsymbol{s}_i)$ at respective locations $\boldsymbol{s}_i$, $i = 1, \ldots, n$. You use the VARIOGRAM procedure because you want to gain insight into the spatial continuity and structure of $Z(\boldsymbol{s})$. A good measure of the spatial continuity of $Z(\boldsymbol{s})$ is defined by means of the variance of the difference $Z(\boldsymbol{s}_i) - Z(\boldsymbol{s}_j)$, where $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ are locations in D. Specifically, if you consider $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ to be spatial increments such that $\boldsymbol{h} = \boldsymbol{s}_j - \boldsymbol{s}_i$, then the variance function based on the increments $\boldsymbol{h}$ is independent of the actual locations $\boldsymbol{s}_i, \boldsymbol{s}_j$. Most commonly, the continuity measure used in practice is one half of this variance, better known as the *semivariance* function:

$$\gamma_z(\boldsymbol{h}) = \frac{1}{2}\mathrm{Var}[Z(\boldsymbol{s}+\boldsymbol{h}) - Z(\boldsymbol{s})]$$

or, equivalently,

$$\gamma_z(\boldsymbol{h}) = \frac{1}{2}\left(\mathrm{E}\{[Z(\boldsymbol{s}+\boldsymbol{h}) - Z(\boldsymbol{s})]^2\} - \{\mathrm{E}[Z(\boldsymbol{s}+\boldsymbol{h})] - \mathrm{E}[Z(\boldsymbol{s})]\}^2\right)$$

The plot of semivariance as a function of $\boldsymbol{h}$ is the *semivariogram*. In extension to its meaning, you might commonly see the term *semivariogram* used instead of the term *semivariance*, as well.

Assume that the SRF $Z(\boldsymbol{s})$ is free of nonrandom (or systematic) surface trends. Then, the expected value $\mathrm{E}[Z(\boldsymbol{s})]$ of $Z(\boldsymbol{s})$ will be a constant for all $\boldsymbol{s} \in \mathcal{R}^2$, and the semivariance expression is simplified to the following:

$$\gamma_z(\boldsymbol{h}) = \frac{1}{2}\mathrm{E}\{[Z(\boldsymbol{s}+\boldsymbol{h}) - Z(\boldsymbol{s})]^2\}$$

Given the preceding assumption, you can compute an estimate $\hat{\gamma}_z(\boldsymbol{h})$ of the semivariance $\gamma_z(\boldsymbol{h})$ from a finite set of points in a practical way by using the formula

$$\hat{\gamma}_z(\boldsymbol{h}) = \frac{1}{2 \mid N(\boldsymbol{h}) \mid} \sum_{N(\boldsymbol{h})} [Z(\boldsymbol{s}_i) - Z(\boldsymbol{s}_j)]^2$$

where the sets $N(\boldsymbol{h})$ contain all the neighboring pairs at distance $\boldsymbol{h}$:

$$N(\boldsymbol{h}) = \{i, j : \boldsymbol{s}_i - \boldsymbol{s}_j = \boldsymbol{h}\}$$

and $\mid N(\boldsymbol{h}) \mid$ is the number of such pairs (i, j).

The expression for $\hat{\gamma}_z(\boldsymbol{h})$ is called the *empirical semivariance* (Matheron 1963). This is the quantity that PROC VARIOGRAM computes, and its corresponding plot is the *empirical semivariogram*. According to Cressie (1993, p. 96), the estimate $\hat{\gamma}_z(\boldsymbol{h})$ has approximate variance

$$\mathrm{Var}[\hat{\gamma}_z(\boldsymbol{h})] \simeq \frac{2[\gamma_z(\boldsymbol{h})]^2}{N(\boldsymbol{h})}$$

The empirical semivariance $\hat{\gamma}_z(\boldsymbol{h})$ is also referred to as *classical*. This name is used so that it can be distinguished from the *robust semivariance* estimate $\bar{\gamma}_z(\boldsymbol{h})$ and the corresponding *robust semivariogram*. The robust semivariance was introduced by Cressie and Hawkins (1980) and is described by Cressie (1993, p. 75) as:

$$\bar{\gamma}_z(\boldsymbol{h}) = \frac{\Psi^4(\boldsymbol{h})}{2[0.457 + 0.494/N(\boldsymbol{h})]}$$

In the preceding expression the parameter $\Psi(\boldsymbol{h})$ is defined as:

$$\Psi(\boldsymbol{h}) = \frac{1}{N(\boldsymbol{h})} \sum_{P_i P_j \in N(\boldsymbol{h})} [Z(\boldsymbol{s}_i) - Z(\boldsymbol{s}_j)]^{\frac{1}{2}}$$

Note that if your data include a surface trend, then the empirical semivariance $\hat{\gamma}_z(\boldsymbol{h})$ is not an estimate of the theoretical semivariance function $\gamma_z(\boldsymbol{h})$. Instead, rather than the spatial increments variance, it represents a different quantity known as *pseudo-semivariance*, and its corresponding plot is a *pseudo-semivariogram*. In principle, pseudo-semivariograms do not provide measures of the spatial continuity. They can thus lead to misinterpretations of the $Z(\boldsymbol{s})$ spatial structure, and are consequently unsuitable for the purpose of spatial prediction. For further information, see the detailed discussion in the section "Empirical Semivariograms and Surface Trends" on page 7559. Under certain conditions you might be able to gain some insight about the spatial continuity with a pseudo-semivariogram. This case is presented in "Example 95.3: Analysis without Surface Trend Removal" on page 7587.

Stationarity

In the combined presence of the previous two assumptions—that is, when $\mathrm{E}[Z(\boldsymbol{s})]$ is constant and spatial increments are used to define $\gamma_z(\boldsymbol{h})$—the SRF $Z(\boldsymbol{s})$ is characterized as *intrinsically stationary* (Cressie 1993, p. 40).

The expected value $\mathrm{E}[Z(\boldsymbol{s})]$ is the first statistical moment of the SRF $Z(\boldsymbol{s})$. The second statistical moment of the SRF $Z(\boldsymbol{s})$ is the *covariance* function between two points $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ in $Z(\boldsymbol{s})$, and it is defined as

$$C_z(\boldsymbol{s}_i, \boldsymbol{s}_j) = \mathrm{E}\left([Z(\boldsymbol{s}_i) - \mathrm{E}[Z(\boldsymbol{s}_i)]]\left[Z(\boldsymbol{s}_j) - \mathrm{E}[Z(\boldsymbol{s}_j)]\right]\right)$$

Note that when $\boldsymbol{s}_i = \boldsymbol{s}_j = \boldsymbol{s}$, the covariance expression provides the variance at $\boldsymbol{s}$.

The assumption of a constant $\mathrm{E}[Z(\boldsymbol{s})] = m$ means that the expected value is invariant with respect to translations of the spatial location $\boldsymbol{s}$. The covariance is considered invariant to such translations when it depends only on the distance $\boldsymbol{h} = \boldsymbol{s}_i - \boldsymbol{s}_j$ between any two points $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$. If both of these conditions are true, then the preceding expression becomes

$$C_z(\boldsymbol{s}_i, \boldsymbol{s}_j) = C_z(\boldsymbol{s}_i - \boldsymbol{s}_j) = C_z(\boldsymbol{h}) = \mathrm{E}\left([Z(\boldsymbol{s}) - m][Z(\boldsymbol{s} + \boldsymbol{h}) - m]\right)$$

When both $\mathrm{E}[Z(\boldsymbol{s})]$ and $C(\boldsymbol{s}_i, \boldsymbol{s}_j)$ are invariant to spatial translations, the SRF $Z(\boldsymbol{s})$ is characterized as *second-order stationary* (Cressie 1993, p. 53).

In a second-order stationary SRF the quantity $C(\boldsymbol{h})$ is the same for any two points that are separated by distance $\boldsymbol{h}$. Based on the preceding formula, for $\boldsymbol{h} = 0$ you can see that the variance is constant throughout a second-order stationary SRF. Hence, second-order stationarity is a stricter condition than intrinsic stationarity.

Under the assumption of second-order stationarity, the semivariance definition at the beginning of this section leads to the conclusion that

$$\gamma_z(\boldsymbol{h}) = C(\mathbf{0}) - C(\boldsymbol{h})$$

which relates the theoretical semivariance and covariance. Note that the empirical estimates of these quantities are not related in exactly the same way, as indicated in Schabenberger and Gotway (2005, Section 4.2.1).

Ergodicity

In addition to the constant $\mathrm{E}[Z(\boldsymbol{s})]$ and the assumption of intrinsic stationarity, *ergodicity* is a necessary third hypothesis to estimate the empirical semivariance. Assume that for the SRF $Z(\boldsymbol{s})$ you have measurements z_i whose sample mean is estimated by $\bar{Z}$. The hypothesis of ergodicity dictates that $\bar{Z} = \mathrm{E}[Z(\boldsymbol{s})]$.

In general, an SRF $Z(\boldsymbol{s})$ is characterized as ergodic if the statistical moments of its realizations coincide with the corresponding ones of the SRF. In spatial analysis we are often interested in the first two statistical moments, and consequently a more relaxed ergodicity assumption is made only for them. See Christakos (1992, Section 2.12) for the use of the ergodicity hypothesis in SRF, and Cressie (1993, p. 57) for a more detailed discussion of ergodicity.

The semivariogram analysis makes implicit use of the ergodicity hypothesis. The VARIOGRAM procedure works with the residual centered values $V(\boldsymbol{s}_i) = v_i = z_i - \bar{Z}, i = 1, \ldots, n$, where it is assumed that the sample mean $\bar{Z}$ is the constant expected value $\mathrm{E}[Z(\boldsymbol{s})]$ of $Z(\boldsymbol{s})$. This is equivalent to using the original values, since $V(\boldsymbol{s}_i) - V(\boldsymbol{s}_j) = Z(\boldsymbol{s}_i) - Z(\boldsymbol{s}_j)$, which shows the property of the semivariance to filter out the mean. See the section "Semivariance Computation" on page 7558 for the exact expressions PROC VARIOGRAM uses to compute the empirical classical $\hat{\gamma}_z(\boldsymbol{h})$ and robust $\bar{\gamma}_z(\boldsymbol{h})$ semivariances.

Anisotropy

Semivariance is defined on the basis of the spatial increment vector $\boldsymbol{h}$. If the variance characteristics of $Z(\boldsymbol{s})$ are independent of the spatial direction, then $Z(\boldsymbol{s})$ is called *isotropic*; if not, then $Z(\boldsymbol{s})$ is called *anisotropic*. In the case of isotropy, the semivariogram depends only on the length h of $\boldsymbol{h}$ and $\gamma_z(\boldsymbol{h}) = \gamma_z(h)$. Anisotropy is characterized as *geometric*, when the range a_0 of the semivariogram varies in different directions, and *zonal*, when the semivariogram sill c_0 depends on the spatial direction. Either type or both types of anisotropy can be present.

In the more general case, an SRF can be anisotropic. For an accurate characterization of the spatial structure it is necessary to perform individual analyses in multiple directions. Goovaerts (1997, p. 98) suggests an initial investigation in at least one direction more than the working spatial dimensions—for example, at least three different directions in $\mathcal{R}^2$. Olea (2006) supports exploring as many directions as possible when the data set allows.

You might not know in advance whether you have anisotropy or not. If the semivariogram characteristics do not change in different directions, then you assume the SRF is isotropic. If your directional analysis reveals anisotropic behavior in particular directions, then you proceed to focus your analysis on these directions. For example, in an anisotropic SRF in $\mathcal{R}^2$ you should expect to find two distinct directions where you observe the *major axis* and the *minor axis* of anisotropy. The major axis direction is the one in which the semivariogram has maximum range, and hence has the strongest spatial continuity. Conversely, in the minor axis direction the SRF has minimum range and the weakest spatial continuity. See "Example 95.2: An Anisotropic Case Study with Surface Trend in the Data" on page 7574 for a detailed demonstration of a case with anisotropy when you are using PROC VARIOGRAM.

You can find some additional information about anisotropy analysis in the section "Anisotropic Models" on page 2955 in the KRIGE2D procedure documentation.

Pair Formation

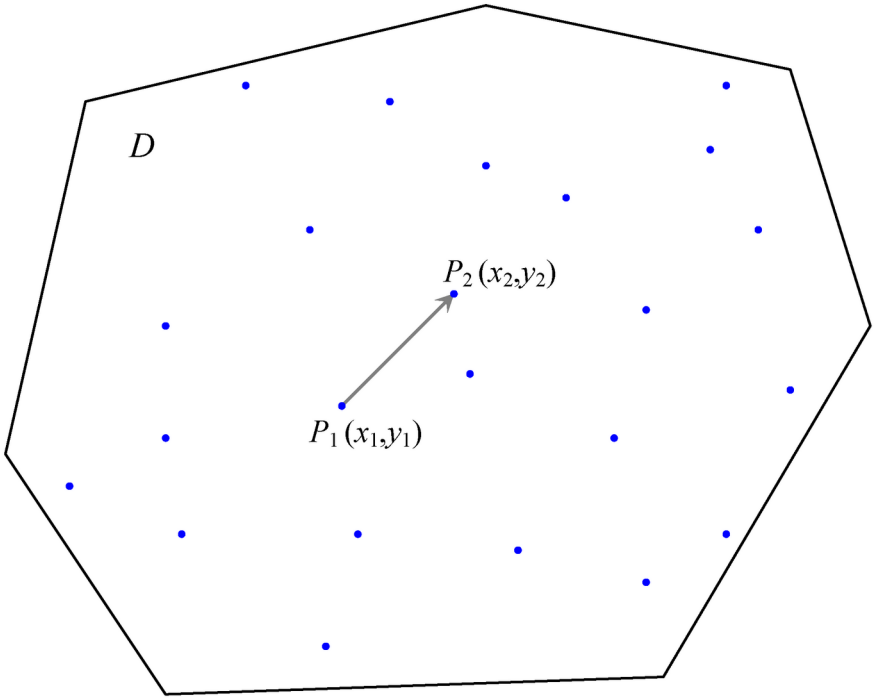
The basic starting point in computing the empirical semivariance is the enumeration of pairs of points for the spatial data. Figure 95.11 shows the spatial domain D and the set of n measurements z_i, $i = 1, \ldots, n$, that have been sampled at the indicated locations in D. Two data points P_1 and P_2, with coordinates $\boldsymbol{s}_1 = (x_1, y_1)$ and $\boldsymbol{s}_2 = (x_2, y_2)$, respectively, are selected for illustration.

A vector, or directed line segment, is drawn between these points. If the length $\mid P_i P_j \mid = \mid \boldsymbol{s}_2 - \boldsymbol{s}_1 \mid = (x_2 - x_1)^2 + (y_2 - y_1)^2$ of this vector is smaller than the specified DEPSILON= value, then the pair is excluded from the continuity measure calculations as the two points P_1 and P_2 are considered to be at zero distance apart (or *collocated*). Spatial collocation might appear due to different scales in sampling, observations made at the same spatial location at different time instances, and errors in the data sets. PROC VARIOGRAM excludes such pairs from the pairwise distance and semivariance computations because they can cause numeric problems in spatial analysis.

If this pair is not discarded on the basis of collocation, it is then classified—first by orientation of the directed line segment $\boldsymbol{s}_2 - \boldsymbol{s}_1$, and then by its length $\mid P_i P_j \mid$. For example, it is unlikely for actual data that the distance $\mid P_i P_j \mid$ between any pair of data points P_i and P_j located at $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, respectively, would exactly satisfy $\mid P_i P_j \mid = \mid \boldsymbol{h} \mid = h$ in the preceding computation of $\hat{\gamma}_z(\boldsymbol{h})$. A similar argument can be made for the orientation of the segment $\boldsymbol{s}_2 - \boldsymbol{s}_1$. Consequently, the pair $P_1 P_2$ is placed into an angle and distance class.

The following subsections give more details about the nature of these classifications. You will also find extensive discussions about the size and the number of classes to consider for the computation of the empirical semivariogram.

Figure 95.11 Selection of Points P_1 and P_2 in Spatial Domain D



Angle Classification

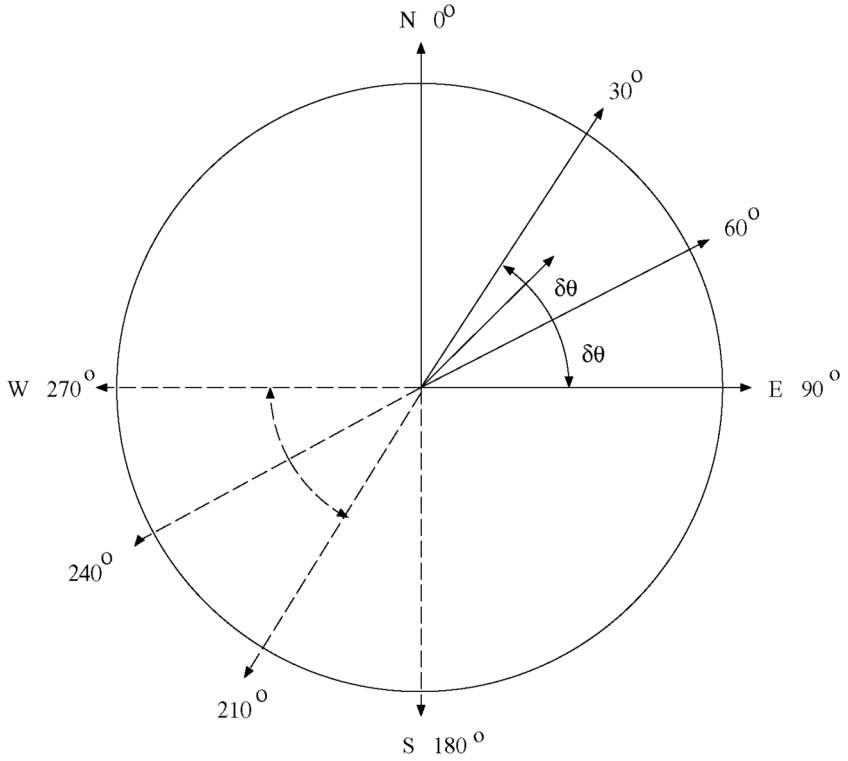
Suppose you specify NDIR=3 in the COMPUTE statement in PROC VARIOGRAM. This results in three angle classes defined by midpoint angles between 0° and 180°: $0° \pm \delta\theta$, $60° \pm \delta\theta$, and $120° \pm \delta\theta$, where $\delta\theta$ is the angle tolerance. If you do not specify an angle tolerance by using the ATOL= option in the COMPUTE statement, the following default value is used:

$$\delta\theta = \frac{180°}{2 \times \text{NDIR}}$$

For example, if NDIR=3, the default angle tolerance is $\delta\theta = 30°$. When the directed line segment P_1P_2 in Figure 95.11 is superimposed on the coordinate system showing the angle classes, its angle is approximately 45°, measured clockwise from north. In particular, it falls within $[60° - \delta\theta, 60° + \delta\theta) = [30°, 90°)$, the second angle class (Figure 95.12).

Note that if the designated points P_1 and P_2 are labeled in the opposite order, the orientation is in the opposite direction—that is, approximately 225° instead of approximately 45°. This does not affect angle class selection; the angle classes $[60° - \delta\theta, 60° + \delta\theta)$ and $[240° - \delta\theta, 240° + \delta\theta)$ are the same.

Figure 95.12 Selected Pair $P_1 P_2$ Falls within the Second Angle Class



If you specify an angle tolerance less than the default, such as ATOL=15°, some point pairs might be excluded. For example, the selected point pair $P_1 P_2$ in Figure 95.12, while closest to the 60° axis, might lie outside $[60 - \delta\theta, 60 + \delta\theta) = [45°, 75°)$. In this case, the point pair $P_1 P_2$ would be excluded from the semivariance computation. This setting can be desirable if you want to reduce interference between neighboring angles. An angle tolerance that is too small might result in too few point pairs in some distance classes for the empirical semivariance estimation (see also the discussion in the section "Choosing the Size of Classes" on page 7555).

On the other hand, you can specify an angle tolerance *greater* than the default. This can result in a point pair being counted in more than one angle classes. This has a smoothing effect on the variogram and is useful when there is a small amount of data or the available data are sparsely located. However, in cases of anisotropy the smoothing effect might have the side effect of amplifying weaker anisotropy in some direction and weakening stronger anisotropy in another (Deutsch and Journel 1992, p. 59).

Changes in the values of the BANDW= option have a similar effect. See the section "Bandwidth Restriction" on page 7552 for an explanation of how BANDW= functions.

An alternative way to specify angle classes and angle tolerances is with the DIRECTIONS statement. The DIRECTIONS statement is useful when angle classes are not equally spaced. When you use the DIRECTIONS statement, you should also specify the angle tolerance, too. The default value of the angle tolerance is 45° when a DIRECTIONS statement is used instead of the NDIRECTIONS= option in the COMPUTE statement. This might not be appropriate for a particular set of angle classes. See the section "DIRECTIONS Statement" on page 7539 for more details.

Distance Classification

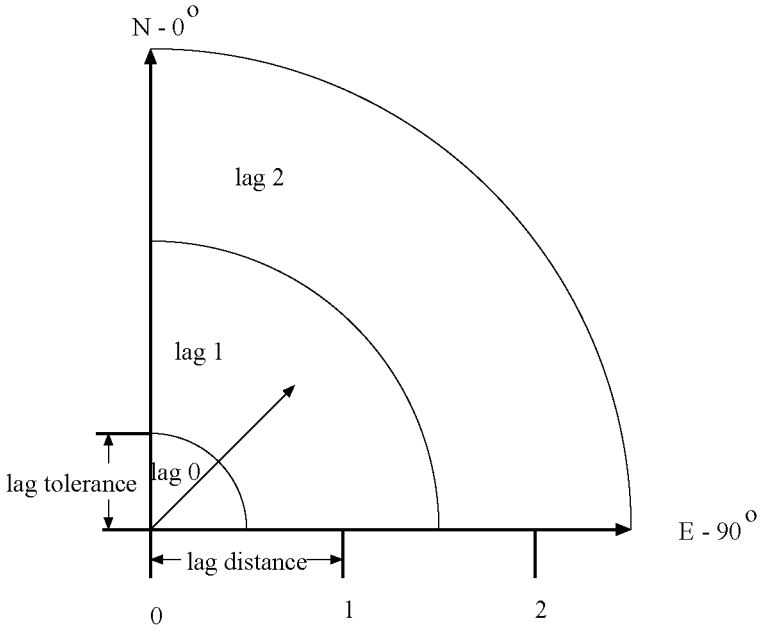
The distance class for a point pair $P_1 P_2$ is determined as follows. The directed line segment $P_1 P_2$ is superimposed on the coordinate system showing the distance or lag classes. These classes are determined by the LAGDISTANCE= option in the COMPUTE statement. Denoting the length of the line segment by $\mid P_1 P_2 \mid$ and the LAGDISTANCE= value by Δ, the lag class L is determined by

$$L(P_1 P_2) = \left\lfloor \frac{\mid P_1 P_2 \mid +0.5}{\Delta} \right\rfloor$$

where $\lfloor x \rfloor$ denotes the largest integer $\leq x$.

When the directed line segment $P_1 P_2$ is superimposed on the coordinate system showing the distance classes, it is seen to fall in the first lag class; see Figure 95.13 for an illustration for $\Delta = 1$.

Figure 95.13 Selected Pair $P_1 P_2$ Falls within the First Lag Class

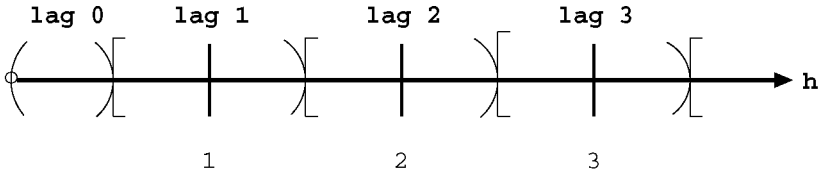


Because pairwise distances are positive, lag class zero is smaller than lag classes $1, \cdots,$ MAXLAG-1. For example, if you specify LAGDISTANCE=1 and MAXLAG=10, and you do not specify a LAGTOL= value in the COMPUTE statement in PROC VARIOGRAM, the 10 lag classes generated by the preceding equation are

$$[0, 0.5), [0.5, 1.5), [1.5, 2.5), \cdots, [8.5, 9.5)$$

This is because the default lag tolerance is half the LAGDISTANCE= value, resulting in no gaps between the distance class intervals. This is shown in Figure 95.14.

Figure 95.14 Lag Distance Axis Showing Lag Classes



On the other hand, if you do specify a distance tolerance with the LAGTOL= option in the COMPUTE statement, a further check is performed to see if the point pair falls within this tolerance of the nearest lag. In the preceding example, if you specify LAGDISTANCE=1 and MAXLAG=10 (as before) and also specify LAGTOL=0.25, the intervals become

$$[0, 0.25), [0.75, 1.25), [1.75, 2.25), \cdots, [8.75, 9.25)$$

Note that this specification results in gaps in the lag classes; a point pair $P_1 P_2$ might fall in an interval such as

$$\mid P_1 P_2 \mid \in [1.25, 1.75)$$

and hence be excluded from the semivariance calculation. The maximum LAGTOL= value allowed is half the LAGDISTANCE= value; no overlap of the distance classes is allowed.

In the section "Computation of the Distribution Distance Classes" on page 7553 there is a more extensive discussion of practical aspects in the specification of the LAGDISTANCE= and MAXLAGS= options.

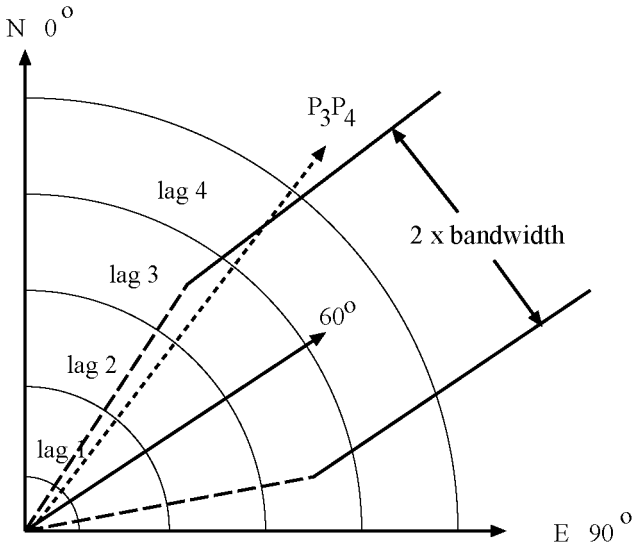
Bandwidth Restriction

Because the areal segments generated from the angle and distance classes increase in area as the lag distance increases, it is sometimes desirable to restrict this area (Deutsch and Journel 1992, p. 45). If you specify the BANDW= option in the COMPUTE statement, the lateral, or perpendicular, distance from the axis defining the angle classes is fixed.

For example, suppose two points P_3, P_4 are picked from the domain in Figure 95.11 and are superimposed on the grid defining distance and angle classes, as shown in Figure 95.15.

The endpoint of vector $P_3 P_4$ falls within the angle class around 60° and the 5th lag class; however, it falls outside the restricted area defined by the bandwidth. Hence, it is excluded from the semivariance calculation.

Figure 95.15 Selected Pair $P_3 P_4$ Falls Outside Bandwidth Limit



Finally, a pair $P_i P_j$ that falls in a lag class larger than the value of the MAXLAGS= option is excluded from the semivariance calculation.

The BANDW= option complements the angle and lag tolerances in determining how point pairs are included in distance classes. Clearly, the number of pairs within each angle/distance class is strongly affected by the angle and lag tolerances and whether BANDW= has been specified. See also the section "Angle Classification" on page 7549 for more details about the effects these rules can have, since BANDW= operates in a manner similar to the ATOL= option.

Computation of the Distribution Distance Classes

This section deals with theoretical considerations and practical aspects when you specify the LAGDISTANCE= and MAXLAGS= options. In principle, these values depend on the amount and spatial distribution of your experimental data.

The value of the LAGDISTANCE= option regulates how many pairs of data are contained within each distance class. In effect, this information defines the pairwise distance distribution (see the following subsection). Your choice of MAXLAGS= specifies how many of these lags you want to include in the empirical semivariogram computation. Adjusting the values of these parameters is a crucial part of your analysis. Based on your observations sample, they determine whether you have sufficient points for a descriptive empirical semivariogram, and they can affect the accuracy of the estimated semivariance, too.

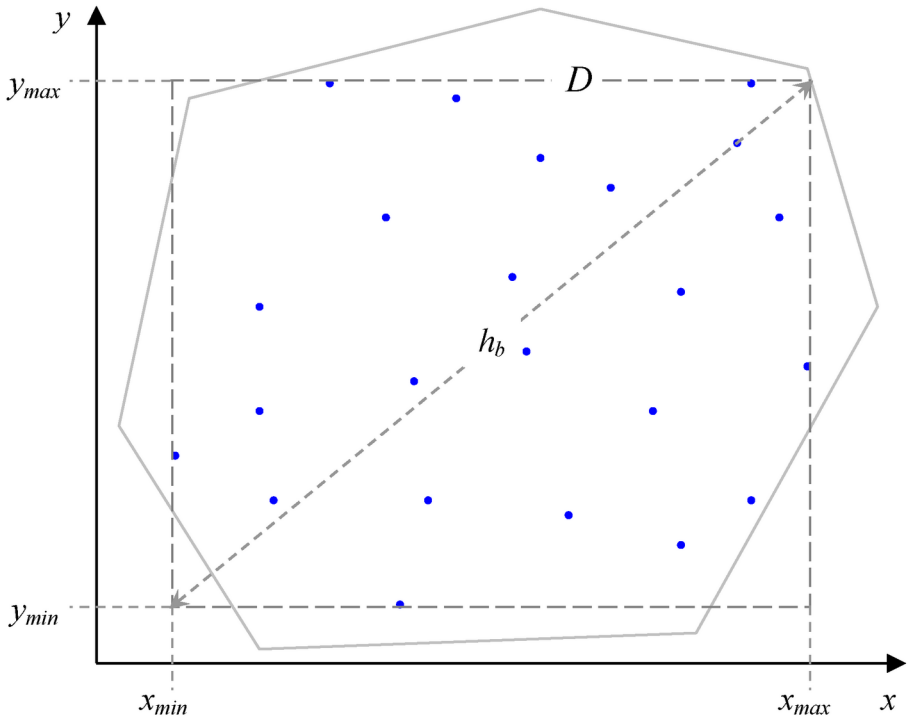
The simplest way of determining the distribution of pairwise distances is to determine the maximum distance h_{max} between any pair of points in your data, and then to divide this distance by some number N of intervals to produce distance classes of length $\delta = h_{max}/N$. The distance $\mid P_1 P_2 \mid$ between each pair of points P_1, P_2 is computed, and the pair $P_1 P_2$ is counted in the kth distance class if $\mid P_1 P_2 \mid \in [(k-1)\delta, k\delta)$ for $k = 1, \cdots, N$.

The actual computation is a slight variation of this. A bound, rather than the actual maximum distance, is computed. This bound is the length of the diagonal of a bounding rectangle for the data points. This bounding rectangle is found by using the maximum and minimum x and y coordinates, $x_{max}, x_{min}, y_{max}, y_{min}$, and forming the rectangle determined by the following points:

(x_{min}, y_{max})	(x_{max}, y_{max})
(x_{min}, y_{min})	(x_{max}, y_{min})

See Figure 95.16 for an illustration of the bounding rectangle applied to the data of the domain D in Figure 95.11. PROC VARIOGRAM provides you with the sizes of $x_{max} - x_{min}$, $y_{max} - y_{min}$, and h_b. For example, in Figure 95.4 in the preliminary analysis, the specified parameters named "Max Data Distance in East," "Max Data Distance in North," and "Max Data Distance" correspond to the lengths $x_{max} - x_{min}$, $y_{max} - y_{min}$, and h_b, respectively.

Figure 95.16 Bounding Rectangle to Determine Maximum Pairwise Distance in Domain D



The pairwise distance bound, denoted by h_b, is given by

$$h_b = \sqrt{(x_{max} - x_{min})^2 + (y_{max} - y_{min})^2}$$

Using h_b, the interval $(0, h_b]$ is divided into $N + 1$ subintervals, where N is the value of the NHCLASSES= option specified in the COMPUTE statement, or $N = 10$ (default) if the NHCLASSES= option is not specified. The basic distance unit is $h_0 = \frac{h_b}{N}$; the distance intervals are centered on $h_0, 2h_0, \cdots, Nh_0$, with a distance tolerance of $\pm\frac{h_0}{2}$. The extra subinterval is $(0, h_0/2)$ and corresponds to the 0th lag. It is half the length of the remaining subintervals, and it often contains the smallest number of pairs. Figure 95.14 shows an example where the lag classes correspond to $h_0 = 1$. This method of partitioning the interval $(0, h_b]$ is used in the empirical semivariogram computation.

Choosing the Size of Classes

When you start with a data sample, the VARIOGRAM procedure computes all the distinct point pairs in the sample. The OUTPAIR= output data set, described in the section "OUTPAIR=*SAS-data-set*" on page 7566, contains information about these pairs. The point pairs are then categorized in classes. The size of each class depends on the common distance that separates consecutive classes. In PROC VARIOGRAM you need to provide this distance value with the LAGDISTANCE= option. Practically, you can define the distance between classes to be about the size of the average sampling distance (Olea 2006).

Under a more scrutinized approach, before you specify a value for the LAGDISTANCE= option, it is helpful to be aware of two issues. First, you should have an estimate of how many classes of data pairs you will need. Each class contributes one point to the empirical semivariogram. Therefore, you need enough classes for an adequate number of points, so that your empirical semivariogram can suggest a suitable theoretical model shape for the description of the spatial continuity. Second, you should keep in mind that a larger number of data pairs in a class can contribute to a more accurate estimate of the corresponding semivariogram point.

The first consideration is a more general issue, and both this and the following subsection address it in detail. Based on the second consideration, the class size problem translates into having a sufficient number of data pairs in each class to produce an accurate semivariance estimate. However, only empirical rules of thumb exist to guide you with this choice. Examples of minimum pairs empirical rules include the suggestion by Journel and Huijbregts (1978, p. 194) to use at least 30 point pairs for each lag class. Also, in a different approach, Chilès and Delfiner (1999, p. 38) increase this number to 50 point pairs.

Obviously, smaller data samples will provide fewer data pairs in the sample. According to Olea (2006), it is difficult to properly estimate a semivariogram with fewer than 50 measurements. The preceding minimum pairs practical rules are useful in cases where small samples are involved. When you work with a relatively small sample, the key is to specify the value of LAGDISTANCE= such that you can strike a balance between the number of the classes you can form and their pairs count. In the coal seam thickness example of the section "Preliminary Spatial Data Analysis" on page 7520, it is not possible to create a desirable large number of classes and maintain an adequate size for each one. On the other hand, there is no practical need to invoke these rules in the case of the much larger sample of ozone concentrations in "Example 95.2: An Anisotropic Case Study with Surface Trend in the Data" on page 7574.

The spatial distribution of the sample might also affect the grouping of pairs into classes. For example, data that are sampled in clusters might prove difficult to classify according to the preceding practical rules. One strategy to address this problem is to accept fewer than 30 pairs for the underpopulated distance classes. Then, at the stage when you determine what theoretical semivariogram model to use, either disregard the corresponding empirical semivariogram points or use them and accept the increased uncertainty.

The VARIOGRAM procedure can help you decide on a suitable class size before you proceed with the empirical semivariogram computation. First, provide a number for the class count by specifying the NHCLASSES= value. Run the procedure with the option NOVARIOGRAM in the COMPUTE statement and examine the distribution data pairs. Use different values of NHCLASSES= to investigate how this will affect the data pairs distribution in each distance class. The pairwise distance intervals table (for example, Figure 95.3) shows the number of pairs in each distance class in the "Number of Pairs" column, and you can use the preceding rule of thumb to adjust the NHCLASSES= value accordingly.

PROC VARIOGRAM displays a rounded value of the distance between the lag bounds as the "Lag Distance" parameter in the pairs information table (see Figure 95.5) or the pairwise distances histogram (see Figure 95.4), which you can use for the LAGDISTANCE= specification. However, this is only one tool. For the semivariogram computation you can specify your own LAGDISTANCE= value based on your experience. Smaller LAGDISTANCE= values result in fewer data pairs in the classes. In that sense, you might find smaller values useful when you work with large samples so that you obtain more semivariogram points. Also, if the LAGDISTANCE= value is too large, you might end up "wasting" too many point pairs in fewer classes at the expense of computing fewer semivariogram points and no significant accuracy gains in the estimation.

As explained earlier, depending on the sample size and its spatial distribution you might have classes with fewer points than what the practical rules advise. Most commonly, the deficient distance classes are the limiting ones close to the origin $\boldsymbol{h} = 0$, as well as the most remote ones at large $\boldsymbol{h}$. The classes near the origin correspond to lags 0 and 1. They are crucial, because the empirical semivariogram in small distances $\boldsymbol{h}$ characterizes the process smoothness and can be further used to detect the presence of a nugget effect. However, as discussed in the section "Distance Classification" on page 7551, lag 0 is half the size of the rest of the classes by definition, so it can be expected to violate the rule of thumb for the number of pairs in a class.

The classes located at higher and extreme distances within a spatial domain are often not accounted for in the empirical semivariogram. The fewer pairs that can be formed in these distances do not allow for an accurate assessment of the spatial correlation, as is explained in the following section.

Spatial Extent of the Empirical Semivariogram

Given your choice for the LAGDISTANCE= value in your spatial domain, the following paragraphs provide guidelines on how many classes to consider when you compute the empirical semivariogram.

Obviously, you want to include no more classes beyond the limit where the pairs count falls below the minimum pairs empirical rule threshold, as discussed in the preceding subsection. PROC VARIOGRAM provides you with a visual way to inspect this upper limit, if you decide to make use of the minimum pairs empirical rule. In particular, specify your threshold choice for the minimum pairs per class by using the THRESHOLD= parameter for the PLOT=PAIRS option.

Then, the procedure produces in the pairwise distances histogram a reference line at the specified THRESHOLD=, which leaves below the line all lags whose pairs count is lower than the threshold value; see, for example, Figure 95.4. The last lag class whose pair population is above the THRESHOLD= is reported in the pairs information table as "Highest Lag With Pairs > Threshold." Note that this value is not a recommendation for the MAXLAGS= option, but rather is an upper limit for your choice. Detailed information about the pairs count in each class is displayed in the corresponding pairwise distance intervals table, as Figure 95.3 demonstrates.

The preceding suggests that you have an upper limit indication, but you still need some criterion to decide how many lags to include in the semivariogram estimation. The criterion is the extent of spatial dependence in your domain.

Spatial dependence can exist beyond your domain limits. However, you do not have data past your domain scale to define a range for larger-scale spatial dependencies. As you look for pairs of data that are gradually farther apart, the number of pairs naturally decreases with distance. The pairs at the more distant classes might be so few that they are likely to be independent with respect to the spatial dependence scale that you can detect. These will only contribute added noise if you include the largest distances in your empirical semivariogram plot. Note that in the same sense, you cannot explore in detail spatial dependencies in scales smaller than an average minimum distance between your data. The nugget effect is then used to represent microscale correlations whose effect is evident in your working scale.

You specify the spatial dependence extent with commonly used measures such as the *correlation range* (or *correlation length*) ϵ and the *correlation radius* h_c. Both are defined in a similar manner. The correlation range ϵ is the distance at which the covariance is 5% of its value at $\boldsymbol{h} = 0$, and shows that beyond ϵ the covariance is considered to be negligible. The correlation radius h_c is the distance at which the covariance is about half the variance at $\boldsymbol{h} = 0$, and indicates the distance over which significant correlations prevail (Christakos 1992, p. 76). The physical meanings of these measures are similar to that of the semivariogram range. Also, the effective range r_ϵ used in asymptotically increasing semivariance models has essentially the same definition as the correlation range ϵ (see the section "Theoretical Semivariogram Models" on page 7540).

A rough estimate of the correlation extent measures might be available from previous studies of a similar site, or from prior information about related measurements. In such an event, you typically want to consider a maximum pairwise distance that does not exceed the length of two or three correlation radii, or one and a half correlation ranges. You can then specify the MAXLAGS= value on the basis of the lags that fit in that distance.

When there are no estimates of correlation extent measures, you can use first use a crude measure to get started with your analysis: You can typically expect MAXLAGS= to be about half of the lag classes shown in the pairwise distances histogram.

Then, if necessary, you can refine your MAXLAGS= choice by using the following maximum lags rule of thumb: Journel and Huijbregts (1978, p. 194) advise considering lags up to about half of the extreme distance between data in the direction of interest. The VARIOGRAM procedure assists you in this task by providing the overall extreme data distance h_b, as well as the extreme data distances in the vertical and horizontal axes directions. For example, h_b is reported in the pairs information table as "Maximum Data Distance" (see Figure 95.5), and in the pairwise distances histogram as "Max Data Distance" (see Figure 95.4).

Overall, you do not want to deviate significantly from the maximum lags rule of thumb. As was stated earlier, a MAXLAGS= value that takes you well beyond the half-extreme distance between data in a given direction might give you limited accuracy in the empirical semivariance estimates at higher distances. At the other end, a value of MAXLAGS= that is too small might lead you to omit important information about the spatial structure that potentially lies within the range of distances you skipped.

Semivariance Computation

With the classification of a point pair $P_i P_j$ into an angle/distance class, as shown earlier in this section, the semivariance computation proceeds as follows.

Denote all pairs $P_i P_j$ belonging to angle class $[\theta_k - \delta\theta_k, \theta_k + \delta\theta_k)$ and distance class $L = L(P_i P_j)$ as $N(\theta_k, L)$. For example, based on Figure 95.12 and Figure 95.13, $P_1 P_2$ belongs to $N(60°, 1)$.

Let $\mid N(\theta_k, L) \mid$ denote the *number* of such pairs. The component of the standard (or method of moments) semivariance corresponding to angle/distance class $N(\theta_k, L)$ is given by

$$\hat{\gamma}(h_k) = \frac{1}{2 \mid N(\theta_k, L) \mid} \sum_{P_i P_j \in N(\theta_k, L)} [V(\boldsymbol{s}_i) - V(\boldsymbol{s}_j)]^2$$

where h_k is the average distance in class $N(\theta_k, L)$; that is,

$$h_k = \frac{1}{\mid N(\theta_k, L) \mid} \sum_{P_i P_j \in N(\theta_k, L)} \mid P_i P_j \mid$$

The robust version of the semivariance is given by

$$\bar{\gamma}(h_k) = \frac{\Psi^4(h_k)}{2[0.457 + 0.494/N(\theta_k, L)]}$$

where

$$\Psi(h_k) = \frac{1}{N(\theta_k, L)} \sum_{P_i P_j \in N(\theta_k, L)} [V(\boldsymbol{s}_i) - V(\boldsymbol{s}_j)]^{\frac{1}{2}}$$

This robust version of the semivariance is computed when you specify the ROBUST option in the COMPUTE statement in PROC VARIOGRAM.

PROC VARIOGRAM computes and writes to the OUTVAR= data set the quantities $h_k, \theta_k, L, N(\theta_k, L), \hat{\gamma}(h)$, and $\bar{\gamma}(h)$.

Empirical Semivariograms and Surface Trends

It was stressed in the beginning of the section "Theoretical and Computational Details of the Semivariogram" on page 7544 that if your data are not free of nonrandom surface trends, then the empirical semivariance $\hat{\gamma}_z(\boldsymbol{h})$ you obtain from PROC VARIOGRAM represents a pseudo-semivariance rather than an estimate of the theoretical semivariance $\gamma_z(\boldsymbol{h})$.

In practice, two major difficulties appear. First, you might have no knowledge of underlying surface trends in your SRF $Z(\boldsymbol{s})$. It can be possible to have this information when you deal with a repetitive phenomenon (Chilès and Delfiner 1999, p. 123), or if you work within a subdomain of a broader region with known characteristics; often, though, this is not the case. Second, even if you suspect the existence of an underlying nonrandom trend, its precise nature might be unknown (Cressie 1993, p. 114, 162).

Based on the last remark, the criteria to define the exact form of a surface trend can be subjective. However, statistical methods can identify the presence and remove an estimate of such a trend. Different trend forms can be estimated in your SRF depending on the trend estimation model that you choose. This choice can lead to different degrees of smoothing in the residual random fluctuations. It might also have an effect on the residuals spatial structure characterization, as trend removals with different models are essentially different operations acting upon the values of your original observations. Note the comment by Chilès and Delfiner (1999, Section 2.7.3) that there are as many semivariograms of residuals as there are ways of estimating the trend. The same source also examines the introduction of bias in the semivariance of the residuals as a side effect of trend removal processes. This bias is small when you examine distances close to the origin $\boldsymbol{h} = 0$, and can increase with distance.

Keeping in mind the preceding remarks, an approach you can take is to use one of the many predictive modeling tools in SAS/STAT software to estimate the unknown trend. Then you use PROC VARIOGRAM to analyze the residuals after you remove the trend. If the resulting model does not require too many degrees of freedom (such as if you use a low-order polynomial), then this approach might be sufficient. The section "Analysis with Surface Trend Removal" on page 7578 demonstrates how to use PROC GLM (see Chapter 39, "The GLM Procedure") for that purpose.

Apart from the standard semivariogram analysis, you can attempt to fit a theoretical semivariogram model to your empirical semivariogram if (a) either the analysis itself or your knowledge of the SRF does not clearly suggest the presence of any surface trend, or (b) the analysis can indicate a potentially trend-free direction, along which your data will have a constant mean.

For example, you might observe overall similar values in your data. This can be an indication that your data are free of nonrandom trends, or that a very mild trend is present. The case falls under the preceding option (a). A very mild trend still allows a good determination of the semivariance at short distances according to Chilès and Delfiner (1999, p. 125), and this can be sufficient for your spatial prediction goal. An analysis of this type is assumed in the section "Preliminary Spatial Data Analysis" on page 7520.

If you observe similar values locally across a particular direction, this an instance of option (b). Olea (2006) suggests recognizing a trend-free direction as being perpendicular to the axis of the maximum dip in the values of $Z(\boldsymbol{s})$. If you suspect that there is at least one such direction in your data, then run PROC VARIOGRAM for a series of directions in the angular vicinity. The trend-free direction, if it exists, will coincide with the one whose pseudo-semivariogram exhibits minimal

increase with distance; see "Example 95.3: Analysis without Surface Trend Removal" on page 7587 for a demonstration of this approach. However, you cannot test $Z(\boldsymbol{s})$ for anisotropy in this case, because you can investigate the semivariogram only in the single trend-free direction (Olea 1999, p. 76). Chilès and Delfiner (1999, Section 2.7.4) suggest fitting a theoretical model in a trend-free direction only if the hypothesis of an isotropic semivariogram appears reasonable in your analysis.

As a result, you need to be very cautious when you choose to perform semivariogram analysis on data you have not previously examined for surface trends. In this event, both of the options (a) and (b) that were reviewed in the preceding paragraphs rely mostly on empirical and subjective criteria. As noted in this section, a degree of subjectivity exists in the selection of the surface trend itself. This fact suggests that a significant part of the semivariogram analysis is based on meta-statistical decisions, as well as on your understanding of your data and the physical considerations that govern your study. In any case, as shown in the section "Theoretical and Computational Details of the Semivariogram" on page 7544, your semivariogram analysis relies fundamentally on the use of trend-free data.

Autocorrelation Statistics (Experimental)

Spatial autocorrelation measures offer you additional insight into the interdependence of spatial data. These measures quantify the correlation of an SRF $Z(\boldsymbol{s})$ with itself at different locations, and they can be very useful whether you have information at exact locations (point-referenced data) or measurements characterizing an area type such as counties, census tracts, zip codes, etc. (areal data).

As in the semivariogram computation, a key issue for the autocorrelation statistics is that you work with a set z_i of measurements, $i = 1, \ldots, n$, that are free of nonrandom surface trends and have a constant mean.

Autocorrelation Weights

In general, the choice of a weighting scheme is subjective. You can obtain different results by using different schemes, options and parameters. PROC VARIOGRAM offers you considerable flexibility in choosing weights that are appropriate for prior considerations such as different hypotheses about neighboring areas, definition of the neighborhood structure, and accounting for natural barriers or other spatial characteristics; see the discussion in Cliff and Ord (1981, p. 17). As stressed for all types of spatial analysis, it is important to have good knowledge of your data. In the autocorrelation statistics, this knowledge will help you avoid spurious correlations when you choose the weights.

The starting point is to assign individual weights to each one of the n data values z_i, $i = 1, \ldots, n$, with respect to the rest. An $n \times n$ matrix of weights is thus defined, such that for any two locations $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, the weight w_{ij} denotes the effect of the value z_i at location $\boldsymbol{s}_i$ on the value z_j at location $\boldsymbol{s}_j$. Depending on the nature of your study, the weights w_{ij} need not be symmetric; that is, it can be that $w_{ij} \neq w_{ji}$.

Binary and Nonbinary Weights

The weights w_{ij} can be either binary or nonbinary values. Binary values of 1 or 0 are assigned if the SRF $Z(\boldsymbol{s}_i)$ at one location $\boldsymbol{s}_i$ is deemed to be connected or not, respectively, to its value $Z(\boldsymbol{s}_j)$ at another location $\boldsymbol{s}_j$. Nonbinary values can be used in the presence of more refined measures of connectivity between any two data points P_i and P_j. PROC VARIOGRAM offers a choice between a binary and a distance-based nonbinary weighting scheme.

In the binary weighting scheme the weight $w_{ij} = 1$, if the data pair at $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ is closer than the user-defined distance LAGDISTANCE=, and $w_{ij} = 0$, if $i = j$ or in any other case. For that reason, in the COMPUTE statement, if you specify the WEIGHTS=BINARY suboption of the AUTOCORRELATION option when the NOVARIOGRAM option is also specified, then you must also specify the LAGDISTANCE= option.

The nonbinary weighting scheme is based on the pair distances and is invoked with the WEIGHTS=DISTANCE suboption of the AUTOCORRELATION option. PROC VARIOGRAM uses a variation of the Pareto form functional to set the weights. Namely, the autocorrelation weight for every point pair P_i and P_j located at $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, respectively, is defined as

$$w_{ij} = s\frac{1}{1+\mid \boldsymbol{h} \mid^p}$$

where $\boldsymbol{h} = \boldsymbol{s}_i - \boldsymbol{s}_j$, and $p \geq 0, s \geq 0$ are user-defined parameters for the adjustment of the weights.

In particular, the power parameter p is specified in the POWER= option of the DISTANCE suboption within the AUTOCORRELATION option. The default value for this parameter is $p = 1$. Also, the scaling parameter s is specified by the SCALE= option in the DISTANCE suboption of the AUTOCORRELATION option. The default value for the scaling parameter is $s = 1$. You can use the p and s parameters to adjust the actual values of the weights according to your needs. Note that variations in the scaling parameter s do not affect the computed values of the Moran's I and Geary's c autocorrelation coefficients that are introduced in the section "Autocorrelation Statistics Types" on page 7562.

Nonbinary Weights with Normalized Distances

PROC VARIOGRAM offers additional flexibility in the DISTANCE weighting scheme through an option to use normalized pair distances. You can invoke this feature by specifying the NORMALIZE option in the DISTANCE suboption of the AUTOCORRELATION option. In this case, the distances used in the definition of the weights are normalized by the maximum pairwise distance h_b (see the section "Computation of the Distribution Distance Classes" on page 7553 and Figure 95.16); the weights are then defined as $w_{ij} = s/[1 + (\mid \boldsymbol{h} \mid / h_b)^p]$.

Note that h_b most likely has a different value for different data sets. Hence, it is suggested that you avoid using the weights you obtain from the preceding equation and a data set for comparisons with the weights you derive from different data sets.

Symmetric and Asymmetric Weights

The weighting schemes presented in the preceding paragraphs are symmetric; that is, $w_{ij} = w_{ji}$ for every data pair at locations $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$. However, you can also define asymmetric weights w'_{ij} such that

$$\sum_{j \in J} w'_{ij} = 1$$

for $i = 1, 2, \cdots, n$, where $w'_{ij} = w_{ij} / \sum_{j \in J} w_{ij}$, $i = 1, 2, \cdots, n$. In the distance-based scheme, J is the set of all locations that form point pairs with the point at $\boldsymbol{s}_i$. In the binary scheme, J is the set of the locations that are connected to $\boldsymbol{s}_i$ based on your selection of the LAGDISTANCE= option; see Cliff and Ord (1981, p. 18). The weights w'_{ij} are *row-averaged* (or *standardized* by the count of their connected neighbors). You can apply row averaging in weights when you specify the ROWAVG option within either the BINARY or DISTANCE suboptions in the AUTOCORRELATION option.

Autocorrelation Statistics Types

One measure of spatial autocorrelation provided by PROC VARIOGRAM is Moran's I statistic, which was introduced by Moran (1950) and is defined as

$$I = \frac{n}{(n-1)S^2 W} \sum_i \sum_j w_{ij} v_i v_j$$

where $S^2 = (n-1)^{-1} \sum_i v_i^2$, and $W = \sum_i \sum_{j \neq i} w_{ij}$.

Another measure of spatial autocorrelation in PROC VARIOGRAM is Geary's c statistic (Geary 1954), defined as

$$c = \frac{1}{2S^2 W} \sum_i \sum_j w_{ij} (z_i - z_j)^2$$

Note that Moran's I coefficient makes use of the centered variable, whereas the Geary's c expression uses the noncentered values in the summation.

Inference on these two statistic types comes from approximate tests based on the asymptotic distribution of I and c, which both tend to a normal distribution as n increases. To this end, PROC VARIOGRAM calculates the means and variances of I and c. The outcome depends on the assumption made regarding the distribution $Z(\boldsymbol{s})$. In particular, you can choose to investigate any of the statistics under the *normality* (also known as *Gaussianity*) or the *randomization* assumption. Cliff and Ord (1981) provided the equations for the means and variances of the I and c distributions, as described in the following.

The normality assumption asserts that the random field $Z(\boldsymbol{s})$ follows a normal distribution of constant mean ($\bar{Z}$) and variance, from which the z_i values are drawn. In this case, the I statistics yield

$$\mathrm{E}_g[I] = -\frac{1}{n-1}$$

and

$$\mathrm{E}_g[I^2] = \frac{1}{(n+1)(n-1)W^2}(n^2 S_1 - nS_2 + 3W^2)$$

where $S_1 = 0.5\sum_i \sum_{j\neq i}(w_{ij} + w_{ji})^2$ and $S_2 = \sum_i(\sum_j w_{ij} + \sum_j w_{ji})^2$. The corresponding moments for the c statistics are

$$\mathrm{E}_g[c] = 1$$

and

$$\mathrm{Var}_g[c] = \frac{(2S_1 + S_2)(n-1) - 4W^2}{2(n+1)W^2}$$

According to the randomization assumption, the I and c observations are considered in relation to all the different values that I and c could take, respectively, if the n z_i values were repeatedly randomly permuted around the domain D. The moments for the I statistics are now

$$\mathrm{E}_r[I] = -\frac{1}{n-1}$$

and

$$\mathrm{E}_r[I^2] = \frac{A_1 + A_2}{(n-1)(n-2)(n-3)W^2}$$

where $A_1 = n[(n^2 - 3n + 3)S_1 - nS_2 + 3W^2]$, $A_2 = -b_2[n(n-1)S_1 - 2nS_2 + 6W^2]$. The factor $b_2 = m_4/(m_2{}^2)$ is the coefficient of kurtosis that uses the sample moments $m_k = \frac{1}{n}\sum_i v_i^k$ for $k = 2, 4$. Finally, the c statistics under the randomization assumption are given by

$$\mathrm{E}_r[c] = 1$$

and

$$\mathrm{Var}_r[c] = \frac{B_1 + B_2 + B_3}{n(n-2)(n-3)W^2}$$

with $B_1 = (n-1)S_1[n^2 - 3n + 3 - (n-1)b_2]$, $B_2 = -\frac{1}{4}(n-1)S_2[n^2 + 3n - 6 - (n^2 - n + 2)b_2]$, and $B_3 = W^2[n^2 - 3 - b_2(n-1)^2]$.

Note that when you specify LAGDISTANCE= to be larger than the maximum data distance in your domain, the binary weighting scheme used by the VARIOGRAM procedure leads to all weights $w_{ij} = 1, i \neq j$. In this extreme case the preceding definitions can show that the variances of the I and c statistics become zero under either the normality or the randomization assumption.

A similar effect might occur when you have collocated observations (see the section "Pair Formation" on page 7548). The Moran's I and Geary's c statistics allow for the inclusion of such pairs in the computations. Hence, contrary to the semivariance analysis, PROC VARIOGRAM does not exclude pairs of collocated data from the autocorrelation statistics.

Interpretation

For Moran's I coefficient, $I > \mathrm{E}[I]$ indicates positive autocorrelation. Positive autocorrelation suggests that neighboring values $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$ tend to have similar feature values z_i and z_j, respectively. When $I < \mathrm{E}[I]$, this is a sign of negative autocorrelation, or dissimilar values at neighboring locations. A measure of strength of the autocorrelation is the size of the absolute difference $| I - \mathrm{E}[I] |$.

Geary's c coefficient interpretation is analogous to that of Moran's I. The only difference is that $c > \mathrm{E}[c]$ indicates negative autocorrelation and dissimilarity, whereas $c < \mathrm{E}[c]$ signifies positive autocorrelation and similarity of values.

The VARIOGRAM procedure uses the mathematical definitions in the preceding section to provide the observed and expected values, and the standard deviation of the autocorrelation coefficients in the autocorrelation statistics table. The Z scores for each type of statistics are computed as follows:

$$Z_I = \frac{I - \mathrm{E}[I]}{\sqrt{\mathrm{Var}[I]}}$$

for Moran's I coefficient, and

$$Z_c = \frac{c - \mathrm{E}[c]}{\sqrt{\mathrm{Var}[c]}}$$

for Geary's c coefficient. PROC VARIOGRAM also reports the two-sided p-value for each coefficient under the null hypothesis that the sample values are not autocorrelated. Smaller p-values correspond to stronger autocorrelation for both the I and c statistics. However, the p-value does not tell you whether the autocorrelation is positive or negative. Based on the preceding remarks, you have positive autocorrelation when $Z_I > 0$ or $Z_c < 0$, and you have negative autocorrelation when $Z_I < 0$ or $Z_c > 0$.

Computational Resources

The fundamental computation of the VARIOGRAM procedure is binning: for each pair of observations in the input data set, a distance class and an angle class are determined and recorded. Let N_d denote the number of distance classes, N_a denote the number of angle classes, and N_v denote the number of VAR variables. The memory requirements for these operations are proportional to $N_d \times N_a \times N_v$. This is typically small.

The CPU time required for the computations is proportional to the number of pairs of observations, or to $N^2 \times N_v$, where N is the number of observations in the input data set.

Output Data Sets

The VARIOGRAM procedure produces four data sets: the OUTACWEIGHTS=*SAS-data-set*, the OUTDIST=*SAS-data-set*, the OUTPAIR=*SAS-data-set*, and the OUTVAR=*SAS-data-set*. These data sets are described in the following sections.

OUTACWEIGHTS=*SAS-data-set*

The OUTACWEIGHTS= data set contains one observation for each pair of points P_1, P_2 in the original data set, where P_1 is different from P_2, with information about the data distance and autocorrelation weight of each point pair.

Note that the OUTACWEIGHTS= data set can be very large, even for a moderately sized DATA= data set. For example, if the DATA= data set has NOBS=500, then the OUTACWEIGHTS= data set has NOBS(NOBS $-$ 1)/2=124,750 observations.

When you perform autocorrelation computations, the OUTACWEIGHTS= data set is a practical way to save the autocorrelation weights for further use.

The OUTACWEIGHTS= data set contains the following variables:

- ACWGHT12, the autocorrelation weight for the pair P_1, P_2
- ACWGHT21, the autocorrelation weight for the pair P_2, P_1
- DISTANCE, the distance between the data in the pair

- V1, the variable value for the first point in the pair
- V2, the variable value for the second point in the pair
- VARNAME, the variable name for the current VAR= variable
- X1, the x coordinate of the first point in the pair
- X2, the x coordinate of the second point in the pair
- Y1, the y coordinate of the first point in the pair
- Y2, the y coordinate of the second point in the pair

When the autocorrelation weights are symmetric, the pair P_1, P_2 has the same weight as the pair P_2, P_1. For this reason, in the case of symmetric weights the OUTACWEIGHTS= data set contains only the autocorrelation weights ACWGHT12.

OUTDIST=*SAS-data-set*

The OUTDIST= data set contains counts for a modified histogram showing the distribution of pairwise distances. This data set provides you with information related to the choice of values for the LAGDISTANCE= option in the COMPUTE statement.

To request an OUTDIST= data set, specify the OUTDIST= data set in the PROC VARIOGRAM statement and the NOVARIOGRAM option in the COMPUTE statement. The NOVARIOGRAM option prevents any semivariogram or covariance computation from being performed.

The following variables are written to the OUTDIST= data set:

- COUNT, the number of pairs falling into this lag class
- LAG, the lag class value
- LB, the lower bound of the lag class interval
- UB, the upper bound of the lag class interval
- PER, the percent of all pairs falling in this lag class
- VARNAME, the name of the current VAR= variable

OUTPAIR=*SAS-data-set*

The OUTPAIR= data set contains one observation for each distinct pair of points P_1, P_2 in the original data set, unless you specify the OUTPDISTANCE= option in the COMPUTE statement.

If you specify OUTPDISTANCE=D_{max} in the COMPUTE statement, all pairs P_1, P_2 in the original data set that satisfy the relation $| P_1 P_2 | \leq D_{max}$ are written to the OUTPAIR= data set.

Note that the OUTPAIR= data set can be very large even for a moderately sized DATA= data set.

For example, if the DATA= data set has NOBS=500, then the OUTPAIR= data set has NOBS(NOBS − 1)/2=124,750 if no OUTPDISTANCE= restriction is given in the COMPUTE statement.

The OUTPAIR= data set contains information about the distance and orientation of each point pair, and you can use it for specialized continuity measure calculations.

The OUTPAIR= data set contains the following variables:

- AC, the angle class value
- COS, the cosine of the angle between pairs
- DC, the distance (lag) class
- DISTANCE, the distance between the data in pairs
- V1, the variable value for the first point in the pair
- V2, the variable value for the second point in the pair
- VARNAME, the variable name for the current VAR= variable
- X1, the x coordinate of the first point in the pair
- X2, the x coordinate of the second point in the pair
- Y1, the y coordinate of the first point in the pair
- Y2, the y coordinate of the second point in the pair

OUTVAR=*SAS-data-set*

The OUTVAR= data set contains the standard and robust versions of the sample semivariance, the covariance, and other information in each lag class.

The OUTVAR= data set contains the following variables:

- ANGLE, the angle class value (clockwise from N to S)
- ATOL, the angle tolerance for the lag/angle class
- AVERAGE, the average variable value for the lag/angle class
- BANDW, the bandwidth for the lag/angle class
- COUNT, the number of pairs in the lag/angle class
- COVAR, the covariance value for the lag/angle class
- DISTANCE, the average lag distance for the lag/angle class

- LAG, the lag class value (in LAGDISTANCE= units)
- RVARIO, the sample robust semivariance value for the lag/angle class
- VARIOG, the sample semivariance value for the lag/angle class
- VARNAME, the name of the current VAR= variable

The robust semivariance estimate, RVARIO, is not included in the data set if the user has not specified the option ROBUST in the DIRECTIONS statement.

The bandwidth variable, BANDW, is not included in the data set if no bandwidth specification is given in the COMPUTE statement or in a DIRECTIONS statement.

The OUTVAR= data set contains a line where the LAG variable is -1. The AVERAGE variable in this line displays the sample mean value $\bar{Z}$ of the SRF $Z(\boldsymbol{s})$, and the COVAR variable shows the sample variance $\mathrm{Var}[Z(\boldsymbol{s})]$.

Displayed Output

In addition to the output data sets, the VARIOGRAM procedure produces several output objects. Most of those are produced depending on whether you specify either NOVARIOGRAM or LAGDISTANCE= and MAXLAGS= in the COMPUTE statement. The VARIOGRAM procedure output objects are the following:

- a default "Number of Observations" table that displays the number of observations read from the input data set and the number of observations used in the analysis
- a default map showing the spatial distribution of the observations of the current variable in the VAR statement. The observations are displayed by default with circled markers whose color indicates the VAR value at the corresponding location.
- a table with basic information about the lags and the extreme distance between data pairs, when NOVARIOGRAM is specified
- a table that describes the distribution of data pairs in distance intervals, when NOVARIOGRAM is specified
- a histogram plot of the pairwise distance distribution, when NOVARIOGRAM is specified). The plot also displays a reference line at a user-specified pairs frequency threshold when you specify the THRESHOLD= parameter in the PLOT=PAIRS option. The option PLOT=PAIRS(NOINSET) forces the informational inset that appears in the plot to hide.
- empirical semivariogram details, when NOVARIOGRAM is not specified and LAGDISTANCE= and MAXLAGS= are specified. This table also includes the semivariance estimate variance and confidence limits when CL is specified, and estimates of the robust semivariance when ROBUST is specified.

- plots of the appropriate empirical semivariograms, when NOVARIOGRAM is not specified and LAGDISTANCE= and MAXLAGS= are specified. If you perform the analysis in more than one direction simultaneously, the output is a panel comprising the empirical semivariogram plots for the specified angles. If the semivariograms are nonpaneled, then each plot includes in the lower part a needle plot of the contributing pairs distribution.
- a table that provides autocorrelation statistics, when the options AUTOCORRELATION and LAGDISTANCE= are specified

ODS Table Names

Each table created by PROC VARIOGRAM has a name associated with it, and you must use this name to reference the table when using ODS Graphics. These names are listed in Table 95.4.

Table 95.4 ODS Tables Produced by PROC VARIOGRAM

ODS Table Name	Description	Statement	Option
AutoCorrStats	Autocorrelation statistics information	COMPUTE	AUTOCORR
DistanceIntervals	Pairwise distances matrix	COMPUTE	NOVARIOGRAM
NObs	Number of observations read and used	PROC	default output
PairsInformation	General information about the pairs distribution in classes and data maximum distances in selected directions	COMPUTE	NOVARIOGRAM
SemivariogramTable	Empirical semivariance classes, parameters, and estimates	COMPUTE	LAGD=, MAXLAGS=

ODS Graphics

This section describes the use of the Output Delivery System (ODS) for creating graphics with the VARIOGRAM procedure.

To request these graphs, you must specify the ODS GRAPHICS statement. For additional control of the graphics that are displayed, see the PLOTS= option in the section "PROC VARIOGRAM Statement" on page 7529. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

ODS Graph Names

PROC VARIOGRAM assigns a name to each graph it creates by using ODS Graphics. You can use these names to reference the graphs when using ODS Graphics. The names are listed in Table 95.5.

Table 95.5 ODS Graphics Produced by PROC VARIOGRAM

ODS Graph Name	Plot Description	Statement	Option
ObservationsPlot	Scatter plot of observed data and colored markers indicating observed values	PROC	PLOTS=OBSERV
PairDistPlot	Histogram of the pairwise distance distribution	PROC	PLOTS=PAIRS
Semivariogram	Plots of empirical classical and robust (optional) semivariograms	PROC	PLOTS=SEMIVAR
SemivariogramPanel	Panel of empirical classical and robust (optional) semivariogram plots	PROC	PLOTS=SEMIVAR

To request these graphs, you must specify the ODS GRAPHICS statement in addition to the statements indicated in Table 95.5. For more information about the ODS GRAPHICS statement, see Chapter 21, "Statistical Graphics Using ODS."

Examples: VARIOGRAM Procedure

Example 95.1: Theoretical Semivariogram Model Fitting

This example continues the introduction study presented in the section "Getting Started: VARIOGRAM Procedure" on page 7519 by fitting a theoretical semivariogram model to the estimated classical sample semivariogram in Figure 95.8. You will use PROC NLIN to perform the model fit and to compare two different approaches: the ordinary least squares (OLS) and the weighted least squares (WLS) fitting methods.

A review of the coal seam thickness empirical semivariogram in Figure 95.8 shows first a slow, then rapid rise from the origin, suggesting a Gaussian-type form:

$$\gamma_z(h) = c_0\left[1 - \exp\left(-\frac{h^2}{a_0^2}\right)\right]$$

as shown in the section "Theoretical Semivariogram Models" on page 7540.

By experimentation, you find that a sill of $c_0 = 7.5$ and a range of $a_0 = 30,000$ feet (effective range $r_\epsilon = \sqrt{3}a_0 \approx 52,000$ feet) provide a reasonable fit of the preceding semivariogram model. You can use these values as initial guesses in the least squares fitting process. For the fitting you need the output data set that is produced by the parameters used in the section "Theoretical Semivariogram Models" on page 7540. The following statements skip the autocorrelation statistics and the creation of plots, and produce the required output data for the semivariogram table. Note that for the fitting

process you need information about the semivariance variance, which you obtain by specifying the CL option in the COMPUTE statement.

```
title 'Theoretical Semivariogram Model Fitting Example';
ods graphics on;

proc variogram data=thick outv=outv plots=none;
   compute lagd=7 maxlag=10 robust cl;
   coordinates xc=East yc=North;
   ods output SemivariogramTable=sv;
   var Thick;
run;
```

Since MAXLAG=10, you computed the empirical semivariogram at 11 points (see also Figure 95.8). You would like to obtain a smooth theoretical semivariogram plot, so you need to estimate the theoretical model at more points on the horizontal (distance) axis. The following statements create a sequence of such distance points from 0 to 70,000 feet and space them 500 feet apart:

```
data pv;
   do Distance = 0 to 70 by 0.5;
      Semivariance = .;
      output;
   end;
run;

data sv; set sv pv; by Distance;
run;
```

PROC NLIN performs its own analysis based on the Gaussian model you provided as input. By invoking the NLIN procedure twice, as shown in the following statements, you obtain the estimates for the theoretical model parameters for the OLS and WLS fitting methods. Notice in the WLS case that the weights are defined as the inverse of the computed semivariance variance.

```
proc nlin data=sv;
   parms Range=30
         Sill=7.5;
   model Semivariance =
         Sill*(1-exp(-Distance*Distance/(Range*Range)));
   output out=OLS p=OLS;
run;

proc nlin data=sv;
   parms Range=30
         Sill=7.5;
   model Semivariance =
         Sill*(1-exp(-Distance*Distance/(Range*Range)));
   _weight_ = 1/SemivarianceStdErr/SemivarianceStdErr;
   output out=WLS p=WLS;
run;
```

Output 95.1.1 shows part of the NLIN procedure output that displays the model parameters statistics for the OLS methods.

Output 95.1.1 Ordinary Least Squares Fitting Parameter Estimates

Theoretical Semivariogram Model Fitting Example

The NLIN Procedure

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits	
Range	27.1706	1.9211	22.8247	31.5165
Sill	7.1628	0.2781	6.5337	7.7919

The corresponding PROC NLIN output for the WLS method is displayed in the following Output 95.1.2.

Output 95.1.2 Weighted Least Squares Fitting Parameter Estimates

Theoretical Semivariogram Model Fitting Example

The NLIN Procedure

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits	
Range	30.6239	1.7382	26.6917	34.5560
Sill	7.2881	0.4082	6.3646	8.2116

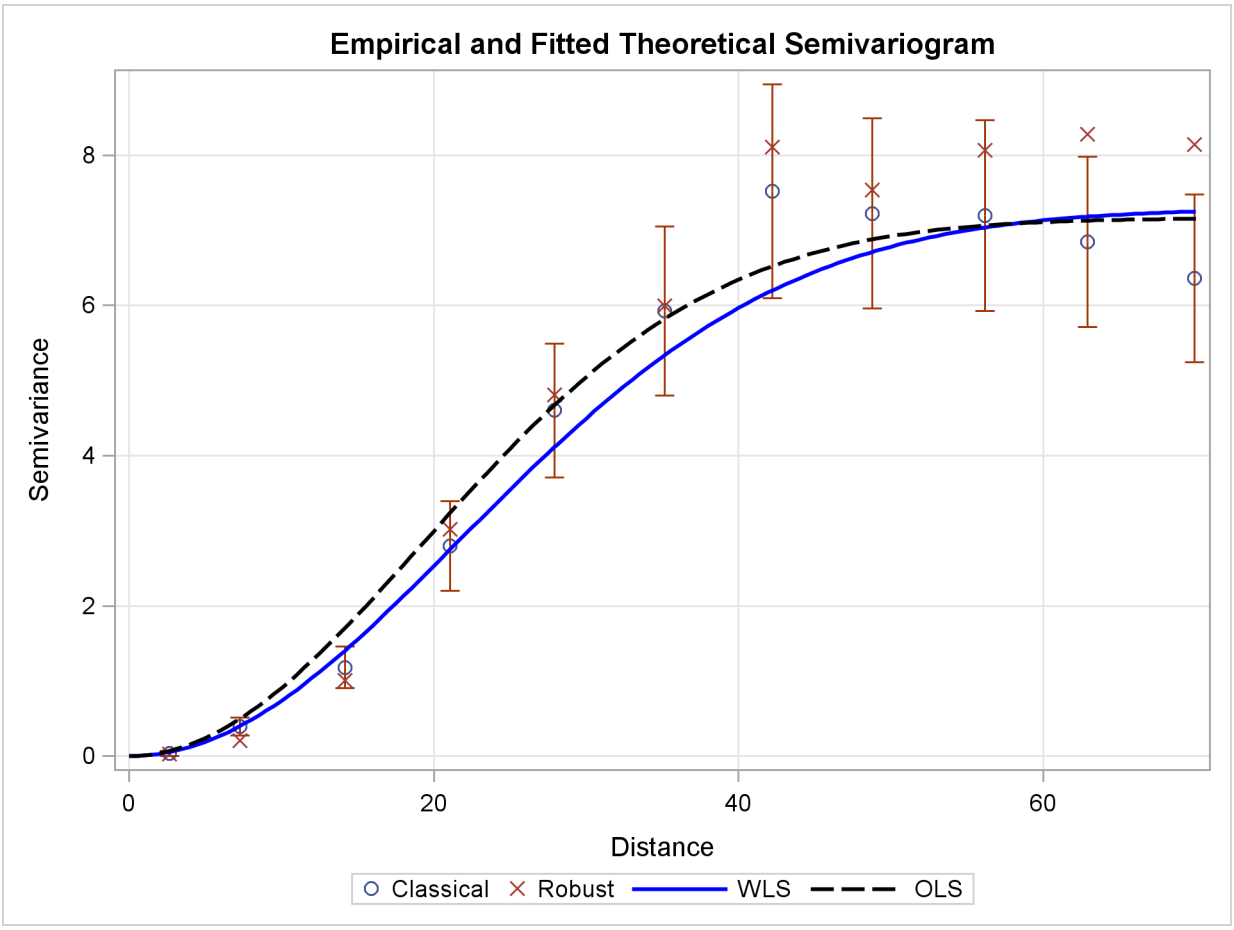
Finally, you visualize the outcome of the fitting analysis. You start with a DATA step to arrange the WLS and OLS data in the same data set. Then you use the SGPLOT procedure to produce a plot showing the empirical and fitted theoretical semivariograms. This sequence is exhibited in the following statements:

```
data pv;
   merge WLS OLS;
run;

proc sgplot data=pv;
   title "Empirical and Fitted Theoretical Semivariogram";
   xaxis label = "Distance" grid;
   yaxis label = "Semivariance" grid;
   scatter y=Semivariance x=Distance /
           markerattrs = GraphData1(symbol=circle)
           name = 'SemiVarClassical'
           yerrorupper = UpperCLSemivariance
           yerrorlower = LowerCLSemivariance;
   scatter y=RobustSemivariance x=Distance /
           markerattrs = GraphData2(symbol=X)
           name = 'SemiVarRobust';
```

```
   series x=Distance y=WLS /
          lineattrs = (thickness=2px color=blue)
          name = 'SemivarPredWgh';
   series x=Distance y=OLS /
          lineattrs = (thickness=2px color=black
                       pattern=MediumDash)
          name = 'SemivarPredUnw';
   discretelegend 'SemiVarClassical' 'SemiVarRobust'
                  'SemivarPredWgh' 'SemivarPredUnw';
run;

ods graphics off;
```

Output 95.1.3 demonstrates the difference between the ordinary and weighted least squares fitting results: WLS achieves a more accurate fit closer to the empirical points with the smaller variances, because the weights are expressed as the inverse of these variances. In the presence of a sizable population per distance class, you expect the points with lower variance to be situated close to the origin, as the semivariance variance expression suggests in the section "Theoretical and Computational Details of the Semivariogram" on page 7544. Hence, you expect in general the WLS method to perform with increased accuracy at short distances because it acknowledges the smaller variance at small $\boldsymbol{h}$. In contrast, the OLS approach performs a least squares overall best fit as it assumes constant variance.

Output 95.1.3 Fitted Theoretical and Empirical Semivariogram for Coal Seam Thickness Data



The WLS method is preferred over the OLS method because it is important to obtain accurate estimates of the spatial continuity closer to the origin $\boldsymbol{h} = 0$. Another advantage of WLS over OLS is that OLS falsely assumes that the differences in the optimization process are normally distributed and independent. However, WLS has the disadvantage that the weights depend on the fitting parameters.

Other fitting methods include maximum likelihood approaches that rely crucially on the normality assumption for the data distribution, and the generalized least squares method, which offers better accuracy but is computationally more demanding. You can find extensive discussions of these issues in Cressie (1993, Section 2.3), Jian, Olea, and Yu (1996), Stein (1988), and Schabenberger and Gotway (2005).

Example 95.2: An Anisotropic Case Study with Surface Trend in the Data

This example shows how to examine data for nonrandom surface trends and anisotropy. You will use simulated data where the variable is atmospheric ozone (O_3) concentrations measured in Dobson units (DU). The coordinates are offsets from a point in the southwest corner of the measurement area, with the east and north distances in units of kilometers (km). You will be working with 300 measurements in a square area of 100 km $\times$ 100 km.

The following statements read the data set.

```
title 'Semivariogram in Anisotropic Case With Trend Removal Example';
data ozoneSet;
   input East North Ozone @@;
   datalines;
   34.9 68.2 286  39.2 12.5 270  44.4 37.7 275  90.5 27.0 282
   91.1 40.8 285  98.6 61.6 294  61.8 26.7 281  64.0 11.5 274
   22.4 26.5 274  89.3 18.3 279  32.3 28.3 274  31.1 53.1 279
   43.0 17.5 272  79.3 42.3 283  99.9 57.9 291   1.8 24.1 273
   81.7 73.5 294  22.9 32.0 273  64.9 67.5 292  76.5 56.3 285
   78.7 11.7 276  61.8 99.3 307  49.1 86.6 299  40.0 35.8 273
   69.3  3.8 278  23.4  9.3 270  66.3 94.3 304  71.3  6.5 275
    9.7 54.4 280  85.2 81.7 300  30.3 60.9 284  94.6 94.3 309
   10.6 10.3 271  73.0 43.0 280   4.9 50.7 280  19.0 79.4 289
    2.4 73.1 287  77.7 25.2 278   8.4 27.1 276  93.5 19.7 279
    0.2 34.5 275  50.4 91.3 302  55.7 26.2 279  50.3  2.3 274
   16.3 84.4 293  19.0  6.9 272  57.1 92.3 303  61.0  0.4 275
   10.7 18.7 271  15.2 43.5 277  67.0 87.4 301  79.0 54.0 285
   36.0 53.3 279  58.3 52.1 282  56.6 79.7 294  40.4 32.4 275
   48.9 64.1 286  54.0 54.9 281  27.5 48.5 279  36.4 30.3 275
   10.5 31.0 273  87.0 39.4 283  47.9 37.5 274  64.7 63.4 288
    0.5 90.8 294  22.8 22.4 275  31.1 78.8 291  93.6 49.8 290
    2.5 39.3 273  83.6 25.6 282  49.8 24.1 278  73.1 91.8 305
   30.5 90.6 297  26.0 61.2 284  58.4 66.2 289  30.5  4.3 273
   38.3 85.6 298  89.2 96.6 309  53.4  6.3 275  27.3 12.8 271
   43.4 56.5 281  99.5 86.9 305  85.8 22.8 281  83.0 10.9 278
   24.8 16.7 271  51.1 18.8 275  59.0 54.3 283  35.5 91.4 298
   18.1 56.0 279  78.0 36.4 277  56.8  6.9 275  21.1 44.5 277
```

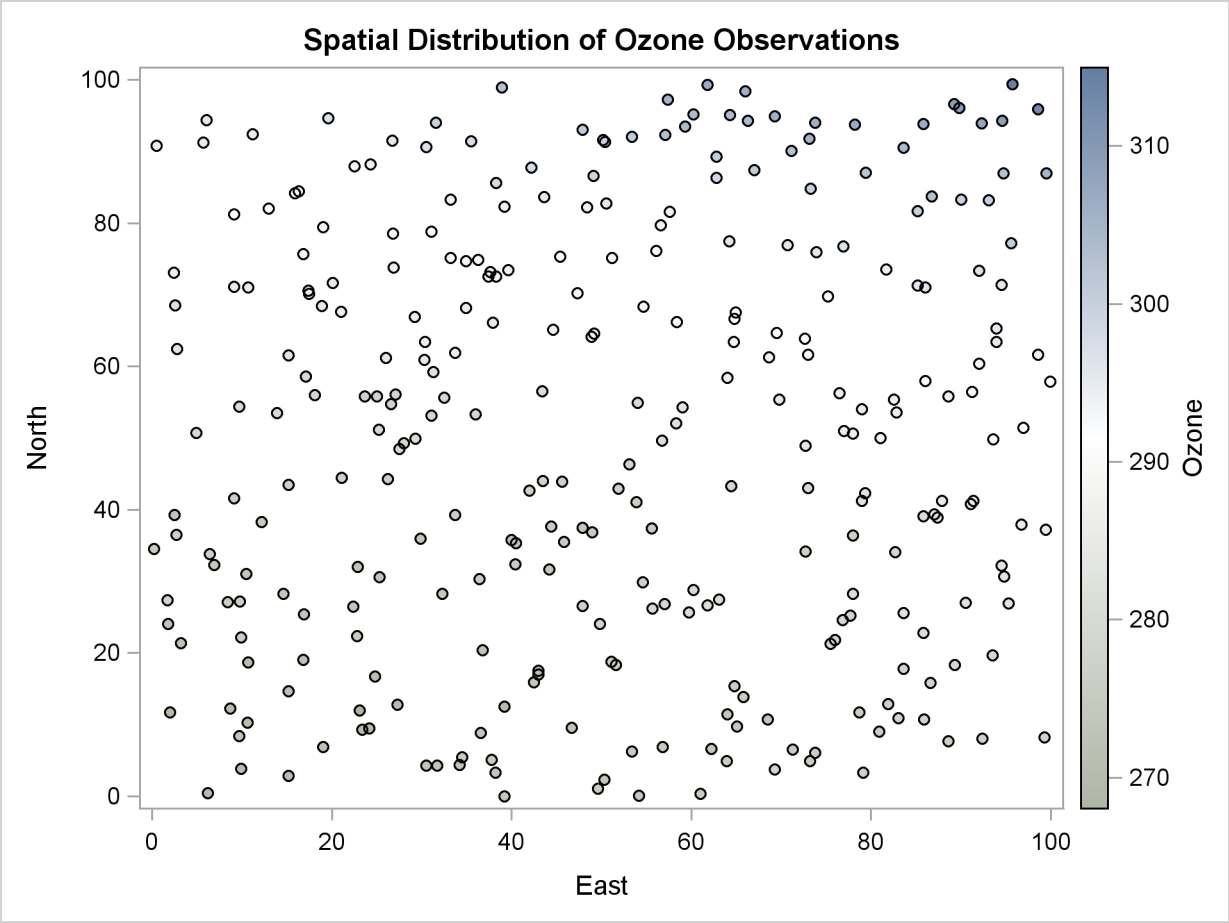
```
   73.9 75.9 296  54.2  0.1 274  33.2 75.1 290  38.2  3.3 274
   15.2 14.7 272  15.9 84.2 292  60.2 95.2 304   9.8 27.2 276
   91.2 56.4 289  94.7 86.9 303  56.7 49.6 281  24.2  9.5 270
   43.0 17.0 272  85.9 10.7 278  53.9 41.1 276  30.4 63.4 286
   62.8 86.3 299  76.8 24.6 279  31.6 94.0 300  26.9 73.8 287
   18.9 68.4 284  99.4 37.2 285  79.1  3.3 277  34.9 74.7 289
    6.4 33.8 277  48.4 82.2 294  86.0 58.0 289  92.0 60.4 293
   50.2 91.6 300  12.2 38.3 275  72.7 48.9 283  82.7 34.1 279
   77.0 51.0 286  86.6 15.8 278  42.0 42.7 277  99.3  8.2 278
   17.4 70.6 286  11.2 92.4 295  60.2 28.8 280  92.0 73.3 297
   25.3 30.6 273  36.6  8.9 274  34.2  4.4 273  26.6 54.7 278
    1.7 27.4 278  49.6  1.1 275  62.8 89.3 301  28.0 49.3 279
   51.2 75.1 293  59.3 93.5 304  83.6 90.5 304  79.4 87.0 302
   78.0 28.3 281  16.8 19.1 272   9.1 81.2 292  23.7 55.8 277
   75.5 21.3 279  64.4 43.3 279  38.9 98.9 303  22.5 87.9 293
   96.7 37.9 285  92.3 93.9 308  16.9 25.4 273  15.2 61.5 283
   73.8 94.0 306  57.4 97.2 305  73.2  4.9 276  39.2 82.3 294
   95.7 99.4 315  66.0 98.4 306  95.3 26.9 283  45.4 75.3 291
   64.8 15.4 276  69.8 55.4 284  36.3 74.9 290   9.9 22.2 276
   65.8 13.9 276  13.0 82.0 293  95.6 77.2 301  32.5 55.6 279
   45.8 35.5 275  62.2  6.6 274  25.2 51.2 279  92.4  8.1 277
   40.5 35.3 273   9.9  3.9 271  43.5 44.0 278  68.6 61.3 287
   64.2 77.5 296  57.6 81.6 294  69.5 64.7 291  64.3 95.1 304
    2.8 62.4 283  33.2 83.3 294  10.7 71.0 285  24.3 88.2 294
   94.5 32.2 283  21.0 67.6 286  20.1 71.6 286  85.2 71.3 296
   94.8 30.7 283  53.4 92.0 301  81.0 50.0 287  54.6 29.9 277
   71.1 90.1 303  15.2  2.9 271  83.6 17.8 278  76.0 21.8 279
   55.6 37.4 275  86.7 83.7 303  43.6 83.6 295  44.2 31.7 274
   90.0 83.3 300   6.2  0.5 270  42.2 87.7 298  31.7  4.3 273
   91.4 41.2 285  78.0 50.6 286  27.1 56.1 278  72.6 63.9 291
   29.3 49.9 281  49.0 36.9 275  13.9 53.5 280  93.1 83.2 300
   73.0 61.6 289  63.1 27.5 280  38.3 72.5 287  72.7 34.2 277
    6.9 32.3 274  17.1 58.6 280  19.6 94.6 297   2.7 36.5 276
   34.5  5.5 275  98.6 95.9 313   9.1 71.1 285  88.6 55.8 287
   26.8 78.5 289  64.8 66.6 292  59.7 25.7 280  47.3 70.2 288
    6.1 94.4 296  50.5 82.7 296   9.1 41.6 276  86.0 71.0 296
   75.2 69.8 293  73.3 84.8 300  42.5 15.9 274  56.1 76.1 292
   87.9 41.2 285  65.1  9.8 274  79.0 41.2 282  44.6 65.1 287
   54.7 68.3 289  57.0 26.8 279   8.7 12.3 270  33.7 61.9 286
   25.0 55.8 278  69.3 94.9 306  49.2 64.6 287  78.2 93.7 307
   47.9 26.6 277  96.9 51.4 292  39.6 73.4 287  37.9 66.1 285
   94.5 71.4 296  51.6 18.3 276  37.6 73.2 287  68.5 10.7 274
   46.7  9.6 273  87.4 38.9 282  45.6 43.9 277  70.7 76.9 296
   82.8 53.6 287  82.5 55.4 286  37.8  5.1 275  89.8 96.1 309
   63.9  4.9 276   2.0 11.7 270  31.3 59.2 282  93.9 65.3 296
   47.9 93.0 301  29.9 36.0 274  14.6 28.3 274  17.5 70.1 286
    2.6 68.5 282  23.1 12.0 268  36.8 20.4 273  80.9  9.0 276
   39.2  0.0 274  26.2 44.3 276  81.9 12.9 277   3.2 21.4 272
   76.9 76.7 297  88.6  7.7 277   9.7  8.4 273  26.7 91.5 296
   73.8  6.1 276  33.7 39.3 276  64.0 58.4 286   5.7 91.2 295
   85.8 93.8 307  85.8 39.1 281  93.9 63.4 295  53.1 46.3 278
   51.9 42.9 277  16.8 75.7 288  29.2 66.9 285  37.4 72.5 287
   ;
run;
```

The initial step is to explore the data set by inspecting the data spatial distribution. Run PROC VARIOGRAM, specifying the NOVARIOGRAM option in the COMPUTE statement as follows:

```
ods graphics on;

proc variogram data=ozoneSet;
   compute novariogram nhc=35;
   coord xc=East yc=North;
   var Ozone;
run;
```

The result is a scatter plot of the observed data shown in Output 95.2.1. The scatter plot suggests an almost uniform spread of the measurements throughout the prediction area. No direct inference can be made about the existence of a surface trend in the data. However, the apparent stratification of ozone values in the northeast–southwest direction might indicate a nonrandom trend.

Output 95.2.1 Ozone Observation Data Scatter Plot



You will need to define the size and count of the data classes by specifying suitable values for the LAGDISTANCE= and MAXLAGS= options, respectively. Compared to the smaller sample of thickness data used in "Example 95.1: Theoretical Semivariogram Model Fitting" on page 7570, the larger size of the ozoneSet data results in more densely populated distance classes for the same

value of the NHC= option. Once you experiment with a variety of values for the NHC= option, you can adjust LAGDISTANCE= to have a relatively small number. Then you can account for a large value of MAXLAGS= so that you obtain many sample semivariogram points within your data correlation range. Specifying these values requires some exploration, for which you might need to return to this point from a later stage in your semivariogram analysis. For illustration purposes you now specify NHC=35.

Your choice of NHC=35 yields the pairwise distance intervals table in Output 95.2.2 and the corresponding histogram in Output 95.2.3.

Output 95.2.2 Pairwise Distance Intervals Table

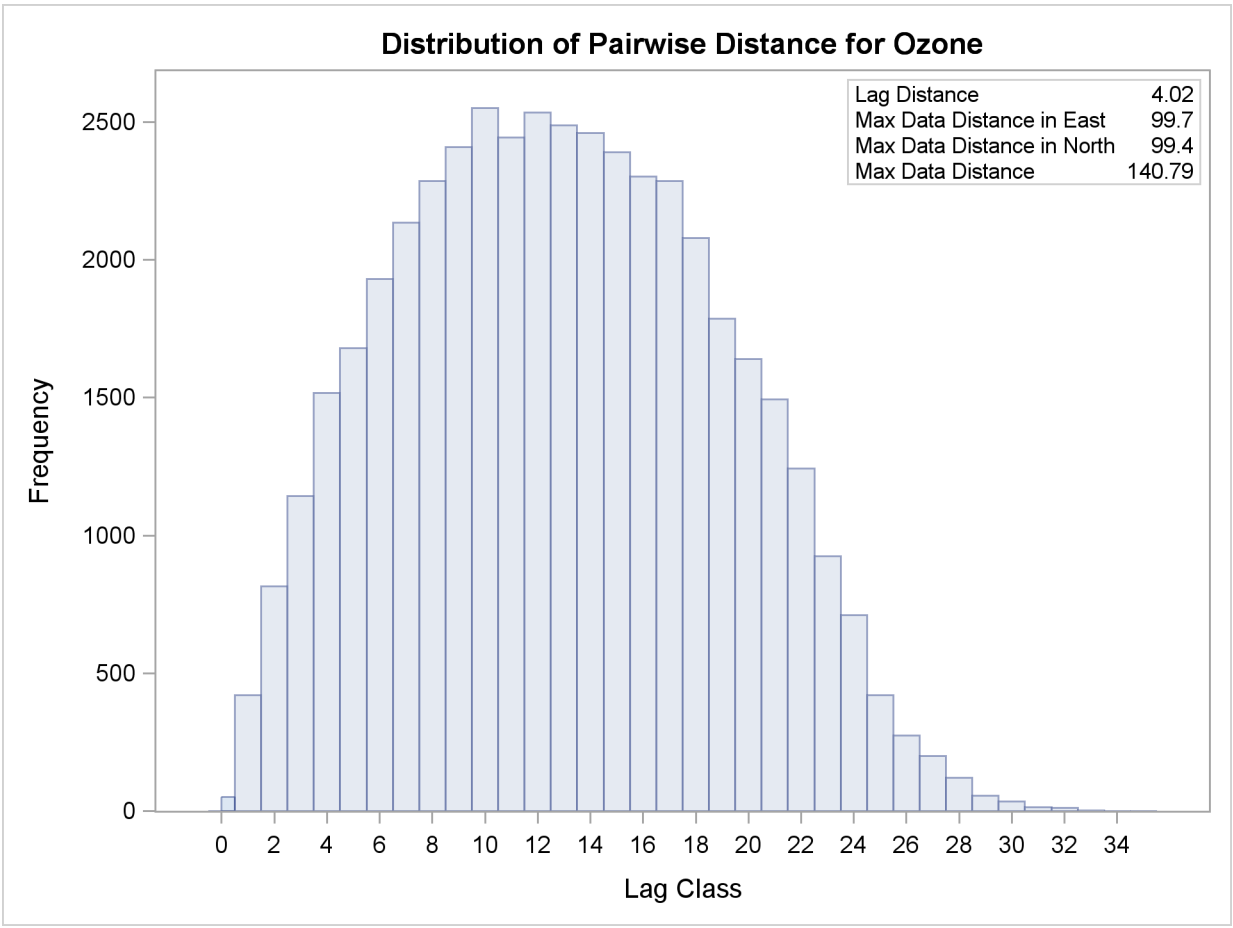
Pairwise Distance Intervals

Lag Class	Bounds		Number of Pairs	Percentage of Pairs
0	0.00	2.01	52	0.12%
1	2.01	6.03	420	0.94%
2	6.03	10.06	815	1.82%
3	10.06	14.08	1143	2.55%
4	14.08	18.10	1518	3.38%
5	18.10	22.12	1680	3.75%
6	22.12	26.15	1931	4.31%
7	26.15	30.17	2135	4.76%
8	30.17	34.19	2285	5.09%
9	34.19	38.21	2408	5.37%
10	38.21	42.24	2551	5.69%
11	42.24	46.26	2444	5.45%
12	46.26	50.28	2535	5.65%
13	50.28	54.30	2487	5.55%
14	54.30	58.33	2460	5.48%
15	58.33	62.35	2391	5.33%
16	62.35	66.37	2302	5.13%
17	66.37	70.39	2285	5.09%
18	70.39	74.41	2079	4.64%
19	74.41	78.44	1786	3.98%
20	78.44	82.46	1640	3.66%
21	82.46	86.48	1493	3.33%
22	86.48	90.50	1243	2.77%
23	90.50	94.53	925	2.06%
24	94.53	98.55	710	1.58%
25	98.55	102.57	421	0.94%
26	102.57	106.59	274	0.61%
27	106.59	110.62	200	0.45%
28	110.62	114.64	120	0.27%
29	114.64	118.66	55	0.12%
30	118.66	122.68	35	0.08%
31	122.68	126.71	14	0.03%
32	126.71	130.73	11	0.02%
33	130.73	134.75	2	0.00%
34	134.75	138.77	0	0.00%
35	138.77	142.80	0	0.00%

Notice the overall high pair count in the majority of classes in Output 95.2.2. You can see that even for higher values of NHC= the classes are still sufficiently populated for your semivariogram analysis according to the rule of thumb stated in the section "Choosing the Size of Classes" on page 7555. Based on the displayed information in Output 95.2.3, you specify LAGDISTANCE=4 km. You can further experiment with smaller lag sizes to obtain more points in your sample semivariogram.

You will return to the MAXLAGS= specification later. The important step now is to investigate the presence of trends in the measurement. The following section makes a suggestion about how to remove surface trends from your data, and then continues the semivariogram analysis with the detrended data.

Output 95.2.3 Distribution of Pairwise Distances for Ozone Observation Data



Analysis with Surface Trend Removal

You can use a SAS/STAT predictive modeling procedure to extract surface trends from your original data. If your goal is spatial prediction, you can continue processing the detrended data for the prediction tasks, and at the end you can reinstate the trend at the prediction locations to report your analysis results.

In general, the exact form of the trend is unknown, as discussed in the section "Empirical Semivariograms and Surface Trends" on page 7559. In this case, the spatial distribution of the measurements shown in Figure 95.2.1 suggests that you can use a quadratic model to describe the surface trend like the one that follows:

$$T(\text{East}, \text{North}) = f_0 + f_1\,[\text{East}] + f_2\,[\text{East}]^2 + f_3\,[\text{North}] + f_4\,[\text{North}]^2$$

The following statements show how to invoke the GLM procedure for your ozone data, and how to extract the preceding trend from them:

```
proc glm data=ozoneSet;
   model ozone = East East*East North North*North;
   output out=gmout predicted=pred residual=ResidualOzone;
run;
```

Among other output, PROC GLM produces estimates for the parameters $f_0, \ldots, f_4$ in the preceding trend model. Output 95.2.4 shows the table with the parameter estimates. In this table, the coefficient f_0 corresponds to the intercept estimate, and the rest of the coefficients correspond to their matching variables; for example, the estimate in the line of "East*East" refers to f_2 in the preceding model. For more information about the syntax and the PROC GLM output, see Chapter 39, "The GLM Procedure."

Output 95.2.4 Parameter Estimates for the Surface Trend Model

Semivariogram in Anisotropic Case With Trend Removal Example

The GLM Procedure

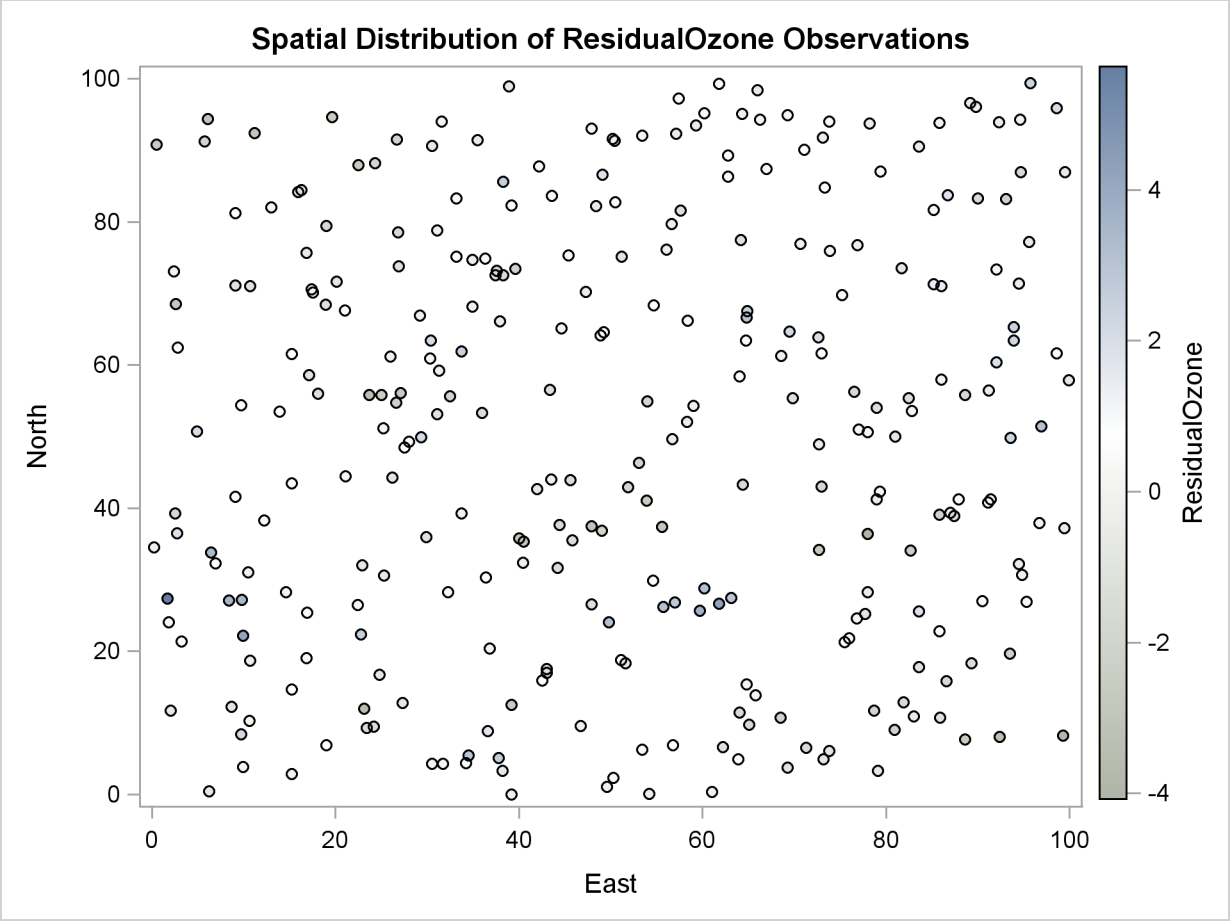
Dependent Variable: Ozone

Parameter	Estimate	Standard Error	t Value	Pr > |t|
Intercept	270.6798273	0.40595731	666.77	<.0001
East	0.0065148	0.01360281	0.48	0.6323
East*East	0.0010726	0.00012987	8.26	<.0001
North	-0.0369159	0.01297491	-2.85	0.0047
North*North	0.0035587	0.00012659	28.11	<.0001

The detrending process leaves you with the GMOUT data set, which contains the ResidualOzone data residuals. You can run PROC VARIOGRAM again as follows, with the NOVARIOGRAM option to inspect the detrended residuals. Note that you ask only for the OBSERVATIONS plot, shown in Output 95.2.5.

```
proc variogram data=gmout plots(only)=observ;
  compute novariogram nhc=35;
  coord xc=East yc=North;
  var ResidualOzone;
run;
```

Output 95.2.5 Ozone Residual Observation Data Scatter Plot



Before you proceed with the empirical semivariogram computation and model fitting, examine your data for anisotropy. This investigation is necessary to portray the spatial structure of your SRF accurately. If anisotropy exists, it will manifest itself as different ranges and/or sills for the empirical semivariograms in different directions.

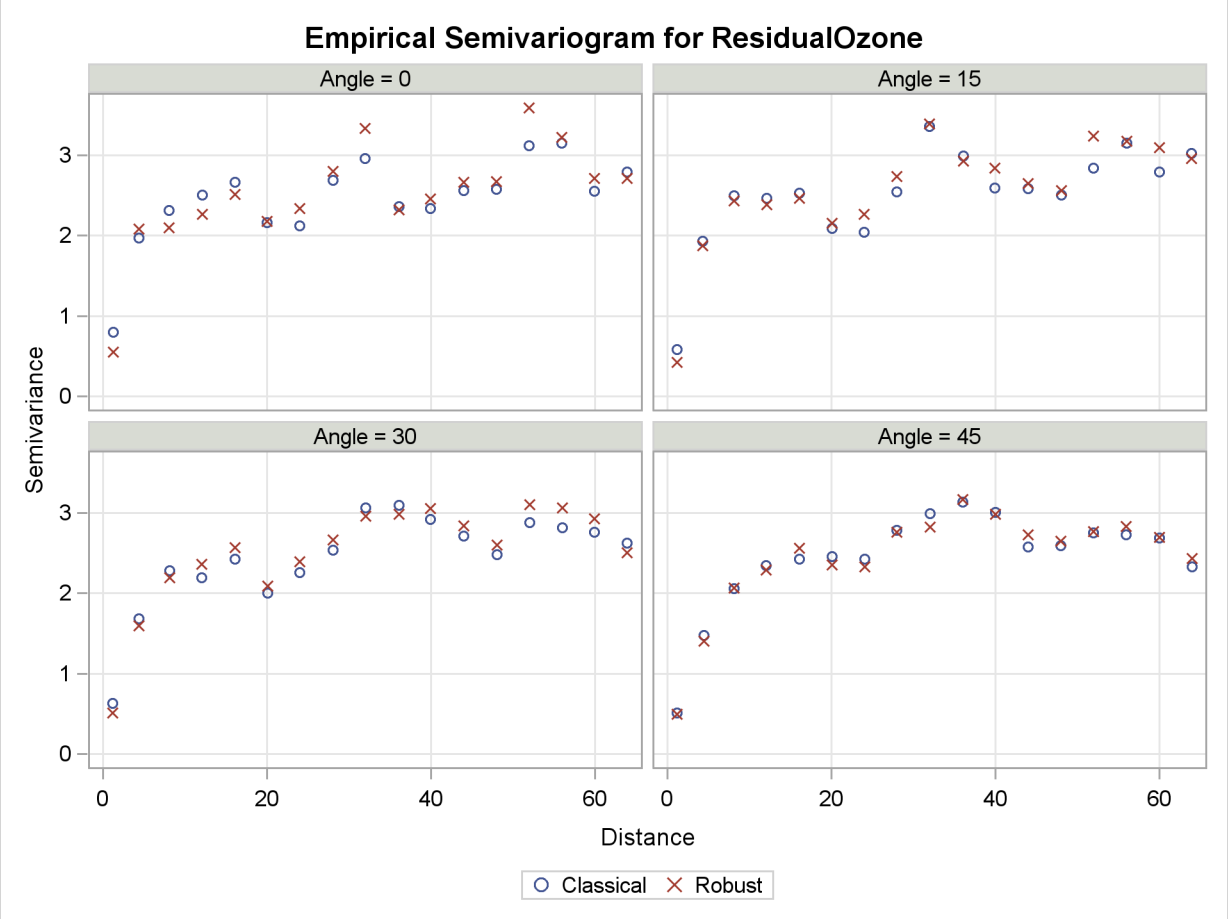
You want detail in your analysis, so you ask for the empirical semivariance in 12 directions by specifying NDIRECTIONS=12. Based on the NDIRECTIONS= option, empirical semivariograms are produced in increments of the base angle $\theta = 180°/12 = 15°$.

You also choose ANGLETOLERANCE=22.5 and BANDWIDTH=20. A different choice of values will produce different empirical semivariograms, because these options can regulate the number of pairs that are included in a class. You do not want these parameters to have values that are too small, so that you can allow for an adequate number of point pairs per class. On the other hand, the higher the values of these parameters, the more data pairs coming from closely neighboring directions will be included in each lag. In this case, there is a risk of losing information along the particular direction. This can be a side effect because you will incorporate data pairs from a broader spectrum of angles, and thus potentially amplify weaker anisotropy or weaken stronger anisotropy, as noted in the section "Angle Classification" on page 7549. You can experiment with different ANGLETOLERANCE= and BANDWIDTH= values to reach this balance with your data, if necessary.

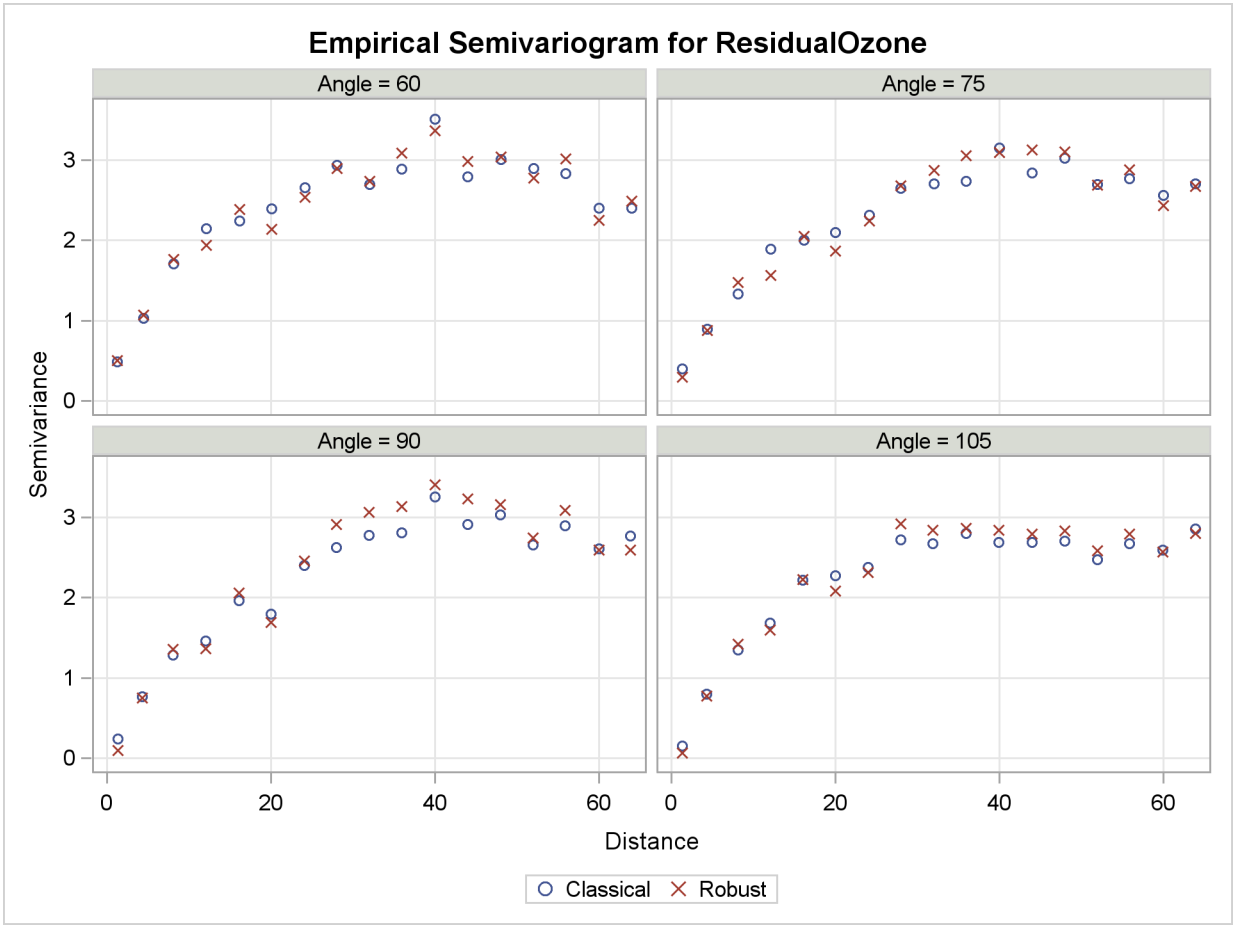
With the following statements you ask to display only the SEMIVAR plots in the specified number of directions. Multiple empirical semivariograms are placed by default in panels, as Output 95.2.6 shows. If you want an individual plot for each angle, then you need to further specify the plot option SEMIVAR(UNPACK).

```
proc variogram data=gmout plot(only)=semivar;
   compute lagd=4 maxlag=16 ndir=12 atol=22.5 bandw=20 robust;
   coord xc=East yc=North;
   var ResidualOzone;
run;
```

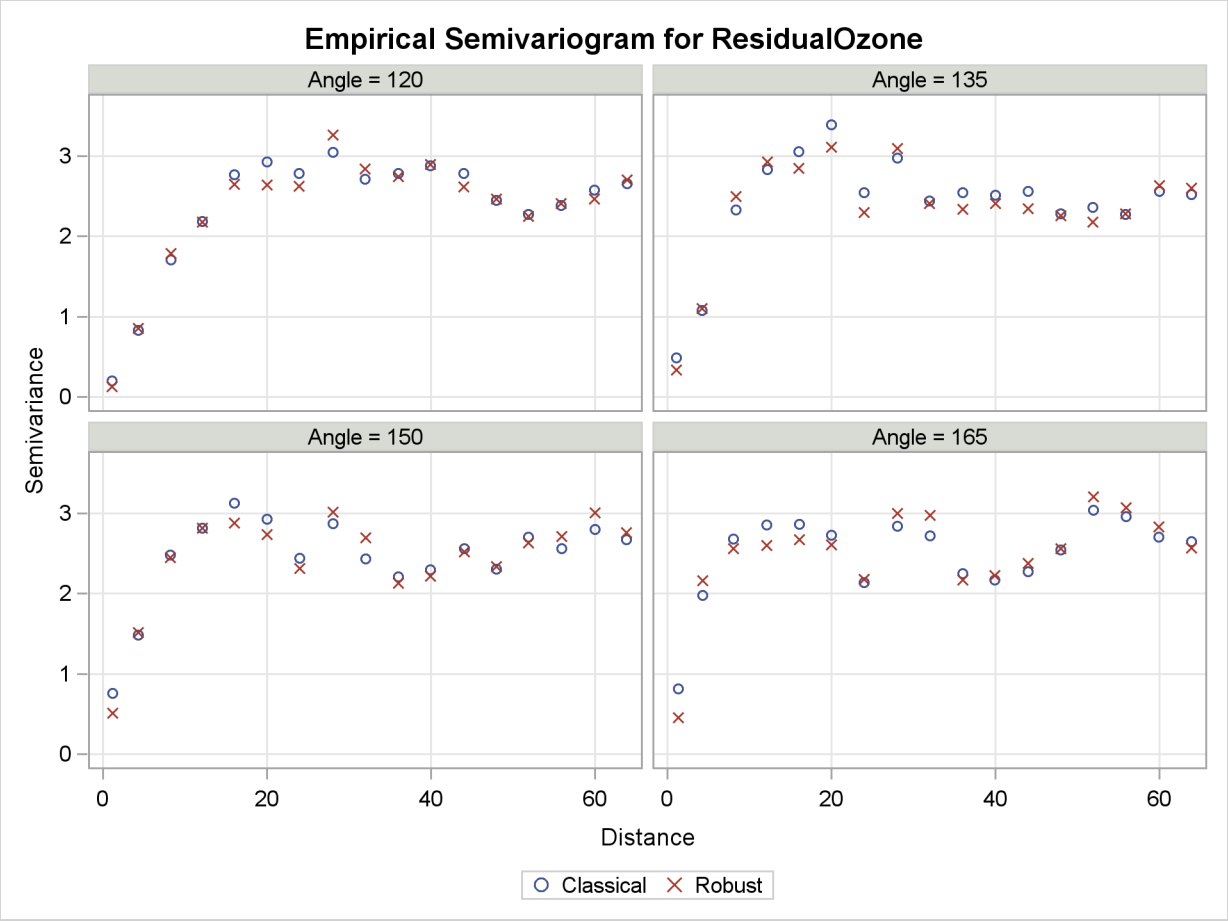
Output 95.2.6 Ozone Empirical Semivariograms with $0° \leq \theta < 180°$ and $\delta\theta = 15°$



Output 95.2.6 *continued*



Output 95.2.6 *continued*



Note that in some of the directions, such as for $\theta = 0°$, the directional plots tend to exhibit a somewhat noisy structure. This behavior can be due to the pairs distribution across the particular direction. Specifically, based on the LAGDISTANCE= choice there might be insufficient pairs present in a class. Also, depending on the ANGLETOLERANCE= and BANDWIDTH= values, too many pairs might be considered from neighboring angles that potentially follow a modified structure. These are factors that can increase the variability in the semivariance estimate. A different explanation might lie in the existence of outliers in the data set; this aspect is further explored in "Example 95.5: A Box Plot of the Square Root Difference Cloud" on page 7600.

This behavior is relatively mild here and should not obstruct your goal to study anisotropy in your data. You can also perform individual computations in any direction. By doing so, you can fine-tune the computation parameters and attempt to obtain smoother estimates of the sample semivariance.

Further in this study, the directional plots in Output 95.2.6 suggest that during shifting from $\theta = 0°$ to $\theta = 90°$, the empirical semivariogram range increases. Beyond the angle $\theta = 90°$, the range starts decreasing again until the whole circle is traversed at 180° and small range values are encountered around the N–S direction at $\theta = 0°$. Note that the sill seems to remain overall the same. This analysis suggests that there is anisotropy in the ozone concentrations, with the major axis oriented at about $\theta = 90°$ and the minor axis situated perpendicular to the major axis at $\theta = 0°$.

The multidirectional analysis requires that for a given LAGDISTANCE= you also specify a MAXLAGS= value. Since the ozone correlation range might be unknown (as assumed here), you can apply the rule of thumb that suggests use of the half-extreme data distance in the direction of interest, as explained in the section "Spatial Extent of the Empirical Semivariogram" on page 7556. Following the information displayed in Output 95.2.3, for different directions this distance varies between $99.4/2 = 49.7$ and $140.8/2 = 70.4$ km. In turn, the pairwise distances table in Output 95.2.2 indicates that within this range of distances you can specify MAXLAG= to be between 12 and 17 lags. In this example you specify MAXLAG=16.

At this point you are ready to continue with fitting theoretical semivariogram models to the empirical semivariogram in the selected directions of $\theta = 0°$ and $\theta = 90°$. This process follows the exact steps shown in "Example 95.1: Theoretical Semivariogram Model Fitting" on page 7570. You apply weighted least squares fitting. In the following, PROC NLIN is used for the simultaneous theoretical semivariogram fitting in both directions of interest. By trying out different models, you see that an exponential one is suitable for your empirical data:

$$\gamma_z(h) = c_0 \left[1 - \exp\left(-\frac{h}{a_0} \right) \right]$$

where $\gamma_z(0) = 0$ and $a_0 > 0$. First, you run PROC VARIOGRAM to create the necessary input information for PROC NLIN. The NLIN procedure needs information about the semivariance variance, which you obtain when you specify the CL option in the COMPUTE statement. You use the following statements:

```
proc variogram data=gmout plot=none;
   compute lagd=4 maxlag=16 robust cl;
   directions 0(22.5,10) 90(22.5,10);
   coord xc=East yc=North;
   ods output SemivariogramTable=svg;
   var ResidualOzone;
run;
```

You also request a vector of points throughout the Distance axis where PROC NLIN estimates the theoretical model values for the two selected directions. Note that essentially you need such a vector for each one of these directions. Then, the output of PROC VARIOGRAM is combined with the added Distance points into the PROC NLIN input data set, as shown in the following statements:

```
data pv;
   do Angle = 0 to 90 by 90;
      do Distance = 0 to 64 by 0.5;
         Semivariance = .;
         output;
      end;
   end;
run;

data svg; set svg pv; by Angle Distance;
run;
```

PROC NLIN requires initial values for the fitting parameters. For the $\theta = 0°$ direction, Output 95.2.6 suggests that a range $a_{00,init} = 4$ km (effective range $r_{\epsilon 0} = 12$ km) and a partial sill ${\sigma_0}^2{}_{0,init} = 2.5$ can be used. Output 95.2.6 indicates that in the direction of $\theta = 90°$, a range of $a_{090,init} = 16$ km (effective range $r_{\epsilon 9}0 = 48$ km) and a partial sill ${\sigma_0}^2{}_{90,init} = 3$ can be used. Both cases indicate that there is no significant nugget effect present. The nugget effect is initialized to the value $c_{ninit} = 0$. Note that you use one nugget effect value for all directions; that is, it is considered isotropic. The following statements implement these considerations, and PROC SGPLOT is eventually called to create the sample and theoretical semivariogram plots:

```
proc nlin data=svg;
   parms Nugget=0 Range0=4   PartSill0=2.5
                  Range90=16 PartSill90=3;
   if (Angle=0) then
      y = Nugget + PartSill0*(1-exp(-Distance/Range0));
   else if (Angle=90) then
      y = Nugget + PartSill90*(1-exp(-Distance/Range90));
   model Semivariance = y;
   _weight_ = 1/SemivarianceStdErr/SemivarianceStdErr;
   output out=WLS p=WLSprediction;
run;

proc sgplot data=WLS;
   title "Empirical and Fitted Theoretical Semivariogram";
   xaxis label = "Distance" grid;
   yaxis label = "Semivariance" grid;
   scatter y=Semivariance x=Distance /
           name='SemiVarClassical'
           group=Angle;
   series x=Distance y=WLSprediction /
          lineattrs=(thickness=2px color=black)
          name='SemivarPredWgh'
          group=Angle;
   discretelegend 'SemiVarClassical' 'SemivarPredWgh';
run;

ods graphics off;
```

Output 95.2.7 shows the PROC NLIN estimates of the fitting parameters. From this table, you can easily compute the estimates for the sills in $\theta = 0°$ and $\theta = 90°$, which are $c_{00} = 2.5339$ and $c_{090} = 2.8703$, respectively. The fitted and empirical semivariograms are displayed in Output 95.2.8.

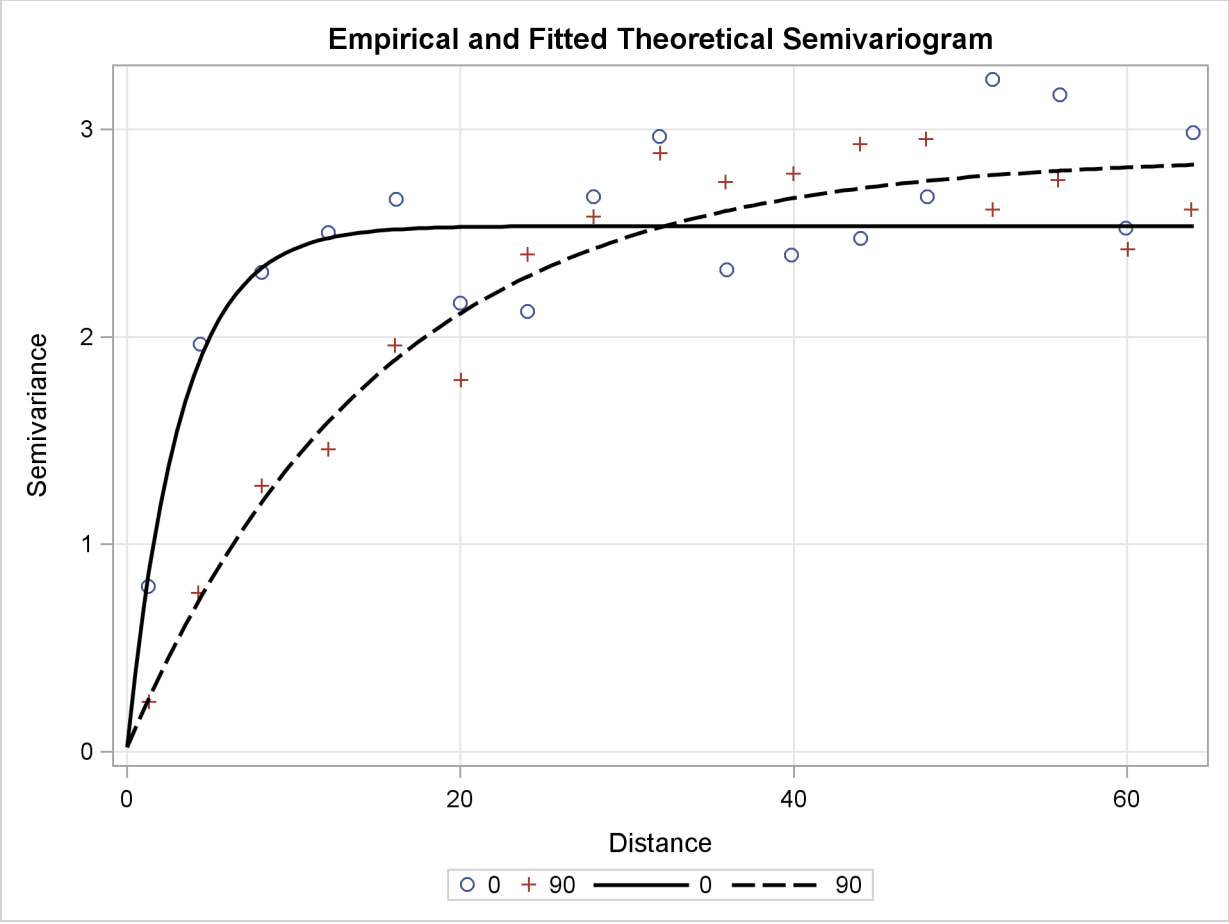
Output 95.2.7 Weighted Least Squares Fitting Parameter Estimates for the Selected Directions $\theta = 0°$ and $\theta = 90°$

Semivariogram in Anisotropic Case With Trend Removal Example

The NLIN Procedure

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits	
Nugget	0.0214	0.1517	-0.2888	0.3316
Range0	3.2073	1.1413	0.8730	5.5415
PartSill0	2.5125	0.1677	2.1696	2.8554
Range90	15.0962	2.7113	9.5510	20.6414
PartSill90	2.8489	0.1740	2.4930	3.2048

Output 95.2.8 Fitted Theoretical and Empirical Semivariogram for the Ozone Data in the $\theta = 0°$ and $\theta = 90°$ Directions



Conclusively, your semivariogram analysis on the detrended ozone data suggests that the ozone SRF exhibits anisotropy in the perpendicular directions of N–S ($\theta = 0°$) and E–W ($\theta = 90°$). The sills in the two directions of anisotropy are similar in size, whereas the range in the major axis is about 4.5 times larger than the minor axis range.

Example 95.3: Analysis without Surface Trend Removal

This example uses PROC VARIOGRAM without removing potential surface trends in a data set, in order to investigate a distinguished spatial direction in the data. In doing so, this example also serves as a guide to examine under which circumstances you might be able to bypass the effect of a trend on a semivariogram. Typically, though, for theoretical semivariogram estimations you follow the analysis presented in "Example 95.2: An Anisotropic Case Study with Surface Trend in the Data" on page 7574.

As explained in the section "Details: VARIOGRAM Procedure" on page 7540, when you compute the empirical semivariance for data that contain underlying surface trends, the outcome is the pseudo-semivariance. Pseudo-semivariograms are not estimates of the theoretical semivariogram; hence, they do not provide any information about the spatial continuity of your SRF.

However, in the section "Empirical Semivariograms and Surface Trends" on page 7559 it is mentioned that you might still be able to perform a semivariogram analysis with potentially non-trend-free data, if you suspect that your measurements might be trend-free across one or more specific directions. The example demonstrates this approach.

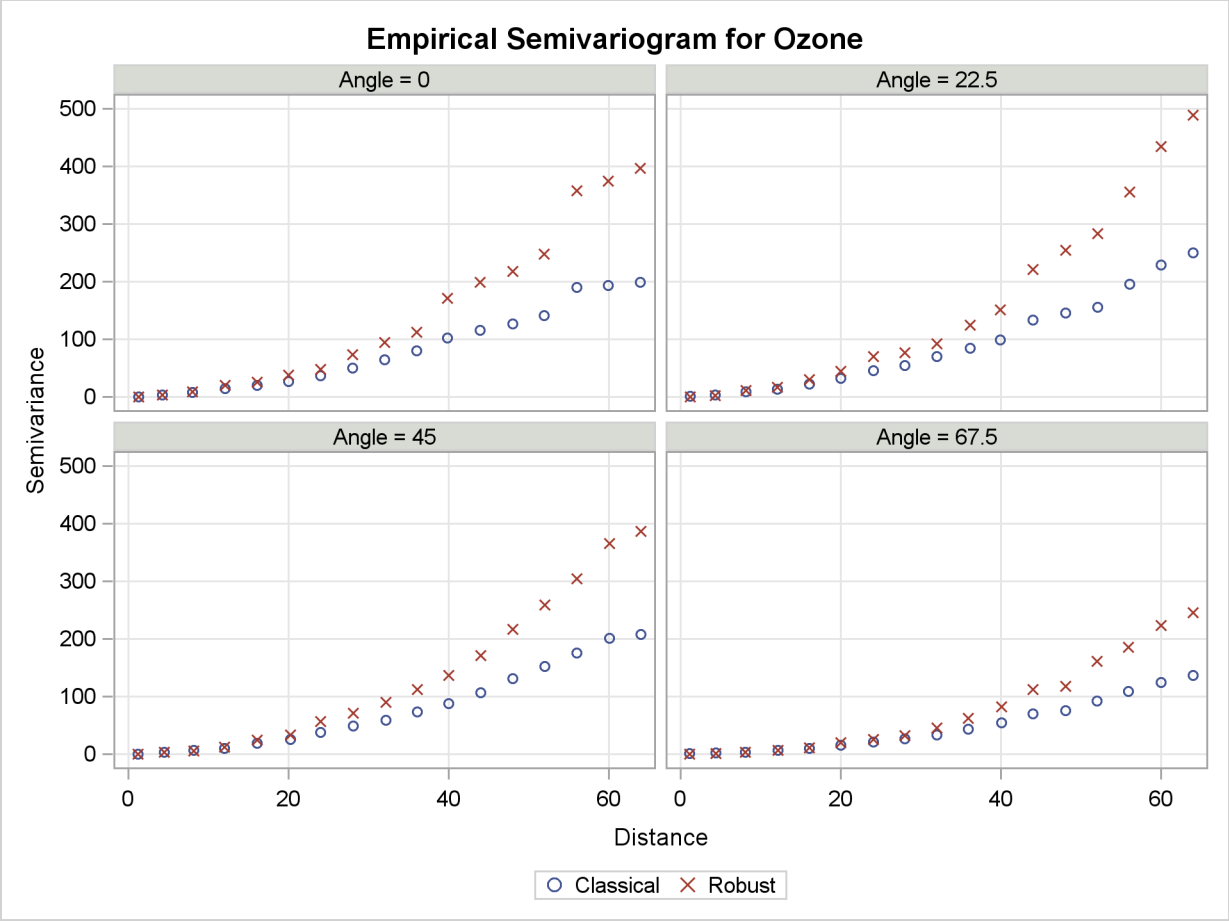
Reconsider the ozone data presented at the beginning of "Example 95.2: An Anisotropic Case Study with Surface Trend in the Data" on page 7574. The spatial distribution of the data is shown in Figure 95.2.1, and the pairwise distance distribution for NHC=35 is illustrated in Figure 95.2.3. This exploratory analysis suggested a LAGDISTANCE=4 km, and Figure 95.2.2 indicated that for this LAGDISTANCE= you can consider a value of MAXLAGS=16.

Recall from the section "Empirical Semivariograms and Surface Trends" on page 7559 that you need to investigate the empirical semivariogram of the data in a few different directions, in order to identify a trend-free direction. If such a direction exists, then you can proceed with this special type of analysis. The following statements employ NDIRECTIONS=8 to examine eight directions:

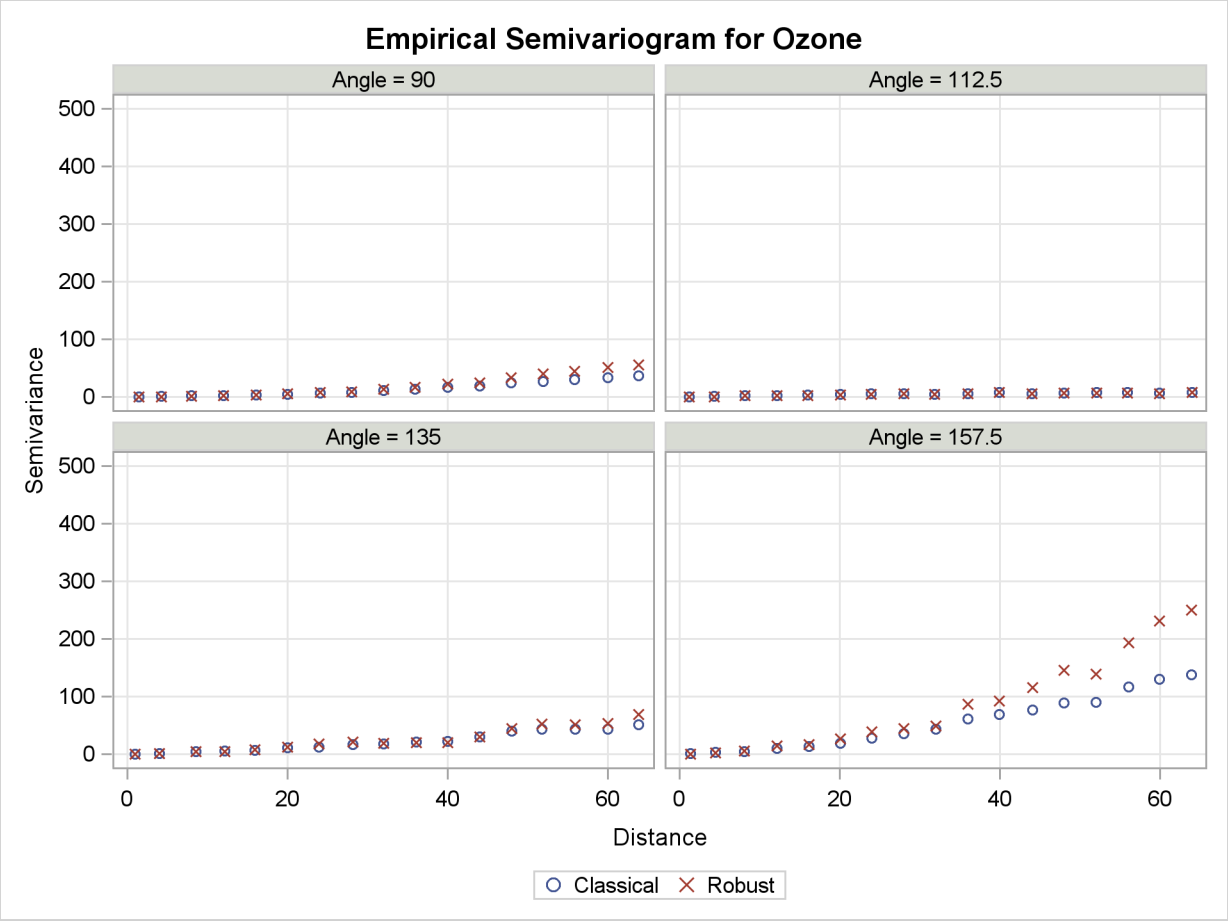
```
title 'Semivariogram Without Trend Removal Example';
ods graphics on;

proc variogram data=ozoneSet plot(only)=semivar;
   compute lagd=4 maxlag=16 ndirections=8 robust;
   coord xc=East yc=North;
   var Ozone;
run;
```

By default, the range of 180° is divided into eight equally distanced angles: $\theta = 0°$, $\theta = 22.5°$, $\theta = 45°$, $\theta = 67.5°$, $\theta = 90°$, $\theta = 112.5°$, $\theta = 135°$, and $\theta = 157.5°$. The resulting empirical semivariograms for these angles are shown in Output 95.3.1.

Output 95.3.1 Ozone Empirical Semivariograms with $0° \leq \theta < 180°$ and $\delta\theta = 22.5°$



Output 95.3.1 *continued*



The figures in Output 95.3.1 suggest that in all directions there is an overall continuing increase with distance of the semivariance. As explained in the section "Theoretical Semivariogram Models" on page 7540, this can be an indication of systematic trends in the data. However, in the direction of $\theta = 112.5°$ there is a clear indication that the increase rate, if any, is smaller than the corresponding rates across the rest of the directions. You then want to search whether there exists a trend-free direction in the neighborhood of this angle.

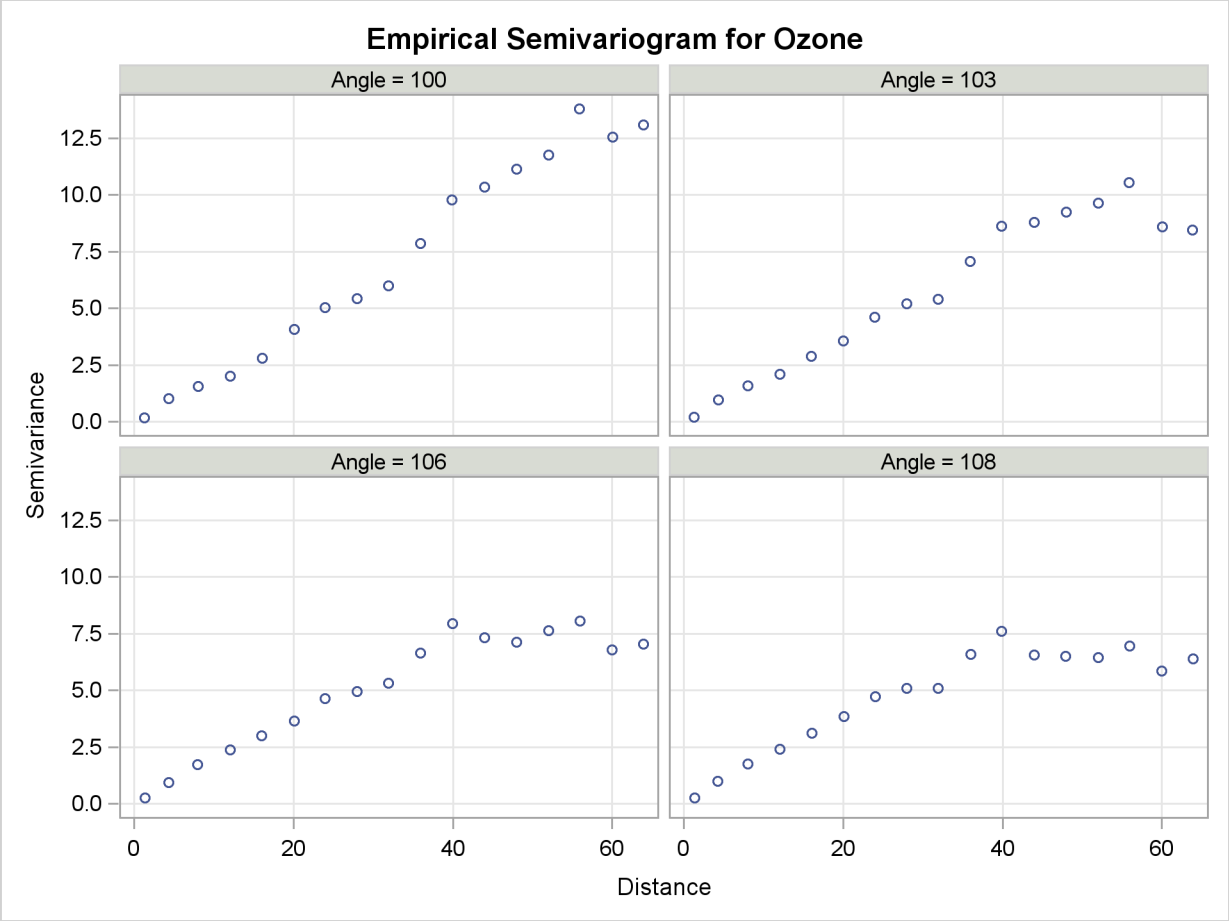
Run PROC VARIOGRAM again, specifying several directions within an interval of angles where you want to close in and suspect the existence of a trend-free direction. In the following step you use an ANGLETOL=15°, which is smaller than the default value of 22.5°, as well as a BANDWIDTH=10 km. The smaller values help with minimization of the interference with neighboring directions, as discussed in the section "Angle Classification" on page 7549.

The aforementioned considerations are addressed in the following statements:

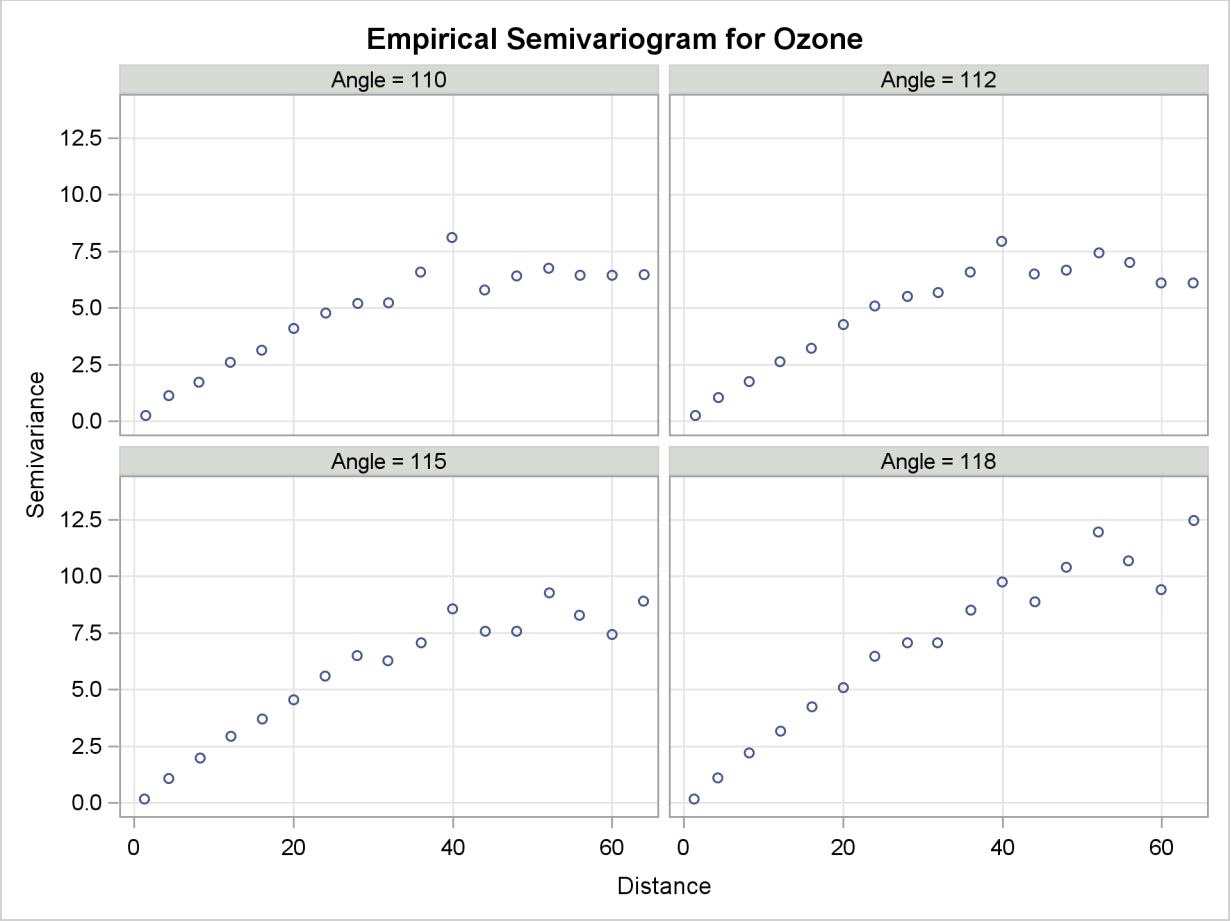
```
proc variogram data=ozoneSet plot(only)=semivar(cla);
   compute lagd=4 maxlag=16 robust;
   directions 100(15,10) 103(15,10)
              106(15,10) 108(15,10)
              110(15,10) 112(15,10)
              115(15,10) 118(15,10);
   coord xc=East yc=North;
   var Ozone;
run;
```

Your analysis has brought you to examine a narrow strip of angles within $\theta = 100°$ and $\theta = 118°$. The pseudo-semivariograms in Output 95.3.2 and Output 95.3.3 indicate that at the boundaries of this strip, the angles display increasing semivariance with distance. On the other hand, within this interval there are directions across which the semivariance is tentatively reaching a sill, and these are potential candidates to be trend-free directions.

Output 95.3.2 Ozone Empirical Semivariograms in 100°, 103°, 106°, and 108°



Output 95.3.3 Ozone Empirical Semivariograms in 110°, 112°, 115°, and 118°



You could further investigate this angle spectrum in more detail. For example, you can monitor additional angles in between, or use a smaller LAGDISTANCE= and increased MAXLAGS= values to single out the most qualified candidate. For the purpose of this example, you can consider the direction $\theta = 108°$ to very likely be the trend-free one you are looking for.

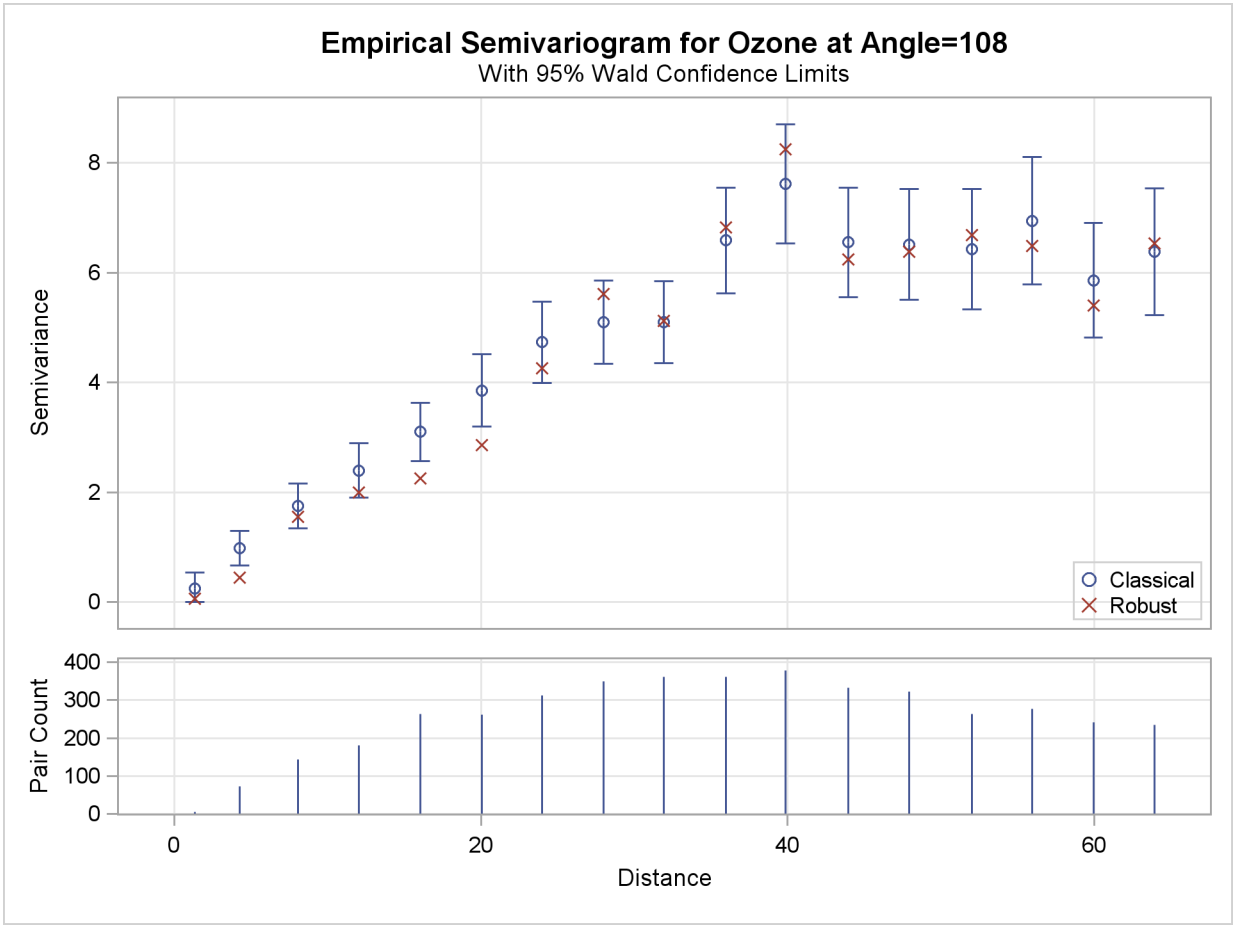
From a physical standpoint, it is also mentioned in the section "Empirical Semivariograms and Surface Trends" on page 7559 that you should expect the trend-free direction, if it exists, to be perpendicular to the direction of the maximum dip in the values of the ozone field. If you cross-examine the ozone data distribution in Output 95.2.1, the figure suggests that this direction exists and is slightly tilted clockwise with respect to the E–W axis. This direction emerges from the mild stratification of the ozone values in your data distribution. The ozone concentrations across it are similar when compared to surrounding directions, and as such, it has been identified as a trend-free direction.

Your next step is to obtain the empirical semivariogram in the suspected trend-free direction of $\theta = 108°$, and to perform a theoretical model fit as shown in "Example 95.1: Theoretical Semivariogram Model Fitting" on page 7570 and "Example 95.2: An Anisotropic Case Study with Surface Trend in the Data" on page 7574.

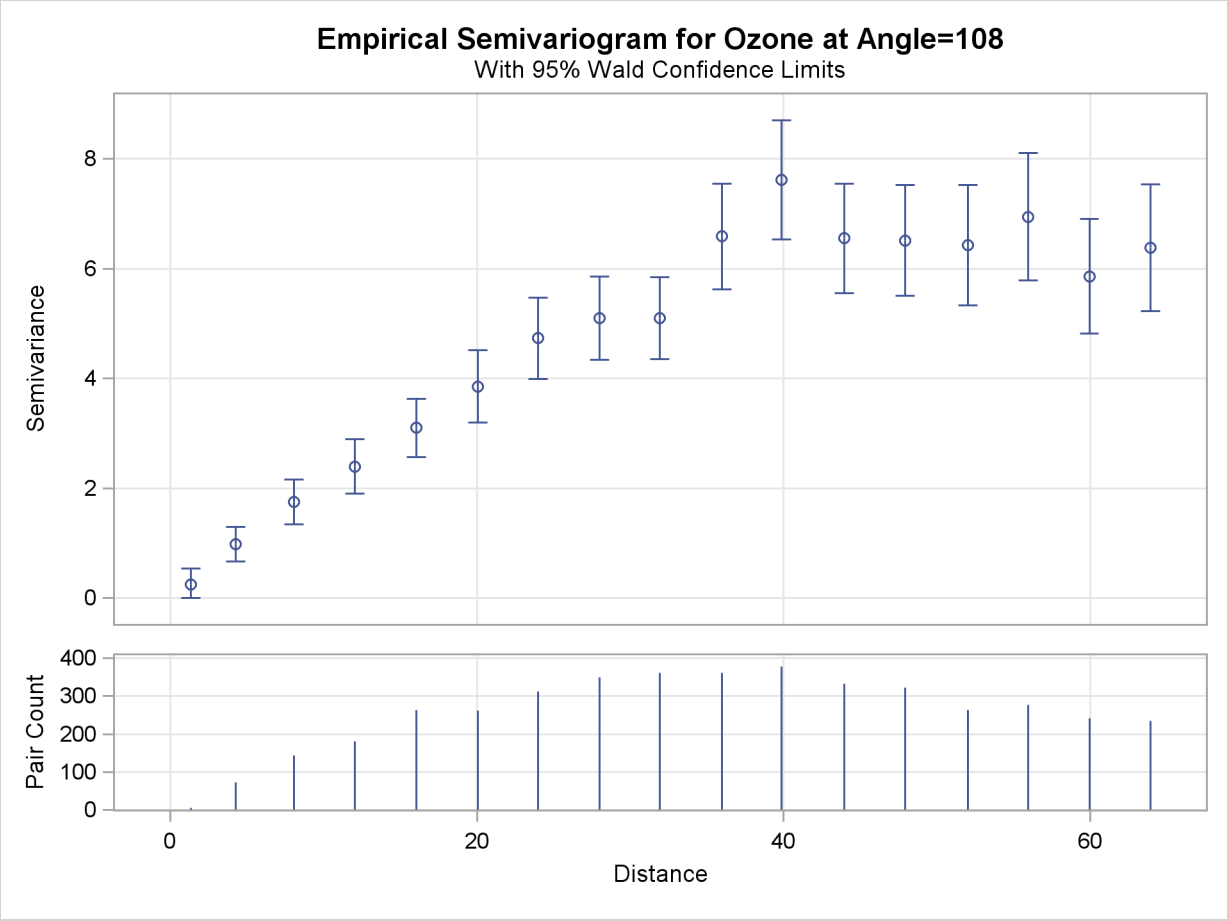
First, assume that you want to inspect the classical and robust empirical semivariograms in the selected direction, as well as a separate plot of the classical one and its confidence limits. Use the PLOT option as shown in the following statements to produce the two empirical semivariograms for your selected angle. The resulting plots are displayed in Output 95.3.4 and Output 95.3.5.

```
proc variogram data=ozoneSet
               plot(only)=(semivar semivar(cla));
   compute lagd=4 maxlag=16 robust cl;
   directions 108(15,10);
   coord xc=East yc=North;
   ods output SemivariogramTable=sv;
   var ozone;
run;
```

Output 95.3.4 Ozone Classical and Robust Empirical Semivariograms in $\theta = 108°$



Output 95.3.5 Ozone Classical Empirical Semivariogram in $\theta = 108°$



Then, you create a vector of points on the Distance axis to provide enough theoretical model estimation locations for a smooth plot. Also, you combine this data set with the output of PROC VARIOGRAM to create the input data set for PROC NLIN. The statements you use are as follows:

```
data pv;
   do Distance = 0 to 64 by 0.5;
      Semivariance = .;
      output;
   end;
run;

data sv; set sv pv; by Distance;
run;
```

Finally, as in the previous examples, you employ PROC NLIN to fit a theoretical model to the empirical semivariogram shown in Output 95.3.5. Note that you have not omitted specifying the CL option earlier in the COMPUTE statement in PROC VARIOGRAM, so that you can provide PROC NLIN with semivariance variance information. The semivariance exhibits a slow, almost linear rise at short distances and seems to be reaching the sill fast, rather than asymptotically.

You can accommodate this behavior by using the following spherical model:

$$\gamma_z(h) = \begin{cases} c_n + {\sigma_0}^2 \left[\frac{3}{2}\frac{h}{a_0} - \frac{1}{2}(\frac{h}{a_0})^3 \right], & \text{for } 0 < h \leq a_0 \\ c_0, & \text{for } a_0 < h \end{cases}$$

where $\gamma_z(0) = 0$ and $a_0 > 0$. PROC NLIN requires initial values for the model parameters. A reasonable initial choice for the sill value is about 7, and for the range $a_{0init} = 40$ km. Though there does not seem to be a nugget effect, you use an initial value $c_{ninit} = 0$ and you initialize the partial sill by using ${\sigma_0}^2{}_{init} = 7$. The following statements implement these considerations. Consequently, PROC SGPLOT is invoked to create the empirical and theoretical semivariogram plots.

```
proc nlin data=sv;
   parms Nugget=0 Range=40 PartSill=7;
   if (Distance<Range)
      then y = Nugget + PartSill*(1.5*(Distance/Range) -
                                  0.5*(Distance/Range)**3);
      else y = Nugget + PartSill;
   model Semivariance = y;
   _weight_ = 1/SemivarianceStdErr/SemivarianceStdErr;
   output out=WLS p=WLSprediction;
run;

proc sgplot data=WLS;
   title "Empirical and Fitted Theoretical Semivariogram";
   xaxis label = "Distance" grid;
   yaxis label = "Semivariance" grid;
   scatter y=Semivariance x=Distance /
           markerattrs = GraphData1(symbol=circle)
           name='SemiVarClassical';
   series x=Distance y=WLSprediction /
          lineattrs = (thickness=2px color=black)
          name = 'SemivarPredWgh';
   discretelegend 'SemiVarClassical' 'SemivarPredWgh';
run;

ods graphics off;
```

PROC NLIN fits the requested model, for which the range, partial sill, and nugget effect estimates are shown in Figure 95.3.6. Clearly, there is an almost negligible nugget effect based on the weighted least squares PROC NLIN analysis. From the fitted values the estimate for the theoretical semivariogram sill is $c_0 = 6.55601$. Figure 95.3.7 displays the fitted and empirical semivariograms in the selected direction $\theta = 108°$.

Output 95.3.6 Weighted Least Squares Fitting Parameter Estimates for $\theta = 108°$

Semivariogram Without Trend Removal Example

The NLIN Procedure

Parameter	Estimate	Approx Std Error	Approximate 95% Confidence Limits	
Nugget	0.00991	0.1111	−0.2284	0.2482
Range	47.0906	2.7872	41.1126	53.0685
PartSill	6.5461	0.2208	6.0725	7.0197

Output 95.3.7 Fitted Theoretical and Empirical Semivariogram for Ozone Data in the Direction $\theta = 108°$



A comparative look at the empirical and fitted semivariograms in Output 95.3.7 and Output 95.2.8 suggests that the analysis of the trend-free ResidualOzone produces a different outcome from that of the original Ozone values. In fact, a more suitable comparison can be made between the semivariograms in the assumed trend-free direction $\theta = 108°$ of the current scenario, and the one shown in Output 95.2.6 in the nearly identical direction $\theta = 105°$. It might seem unreasonable that these two semivariograms are produced both in the same ozone study and in a narrow band of directions free

of apparent surface trends, yet they bear no resemblance. However, the lack of similarity in these plots stems from operating on two different data sets where the outcome depends on the actual data values.

More specifically, in the eyes of semivariogram analysis the trend-free ozone set and the original ozone measurements are treated as different quantities. The process of detrending the original Ozone values is a transformation of these values into the trend-free ones of ResidualOzone. Any existing spatial correlation in the original data is not directly memorized into the transformed data, but is rather affected by the transformation features. In principle, the emerging data set has its own characteristics, as demonstrated in this example.

A final remark concerns the issue of isotropy. Based on the details presented in the section "Empirical Semivariograms and Surface Trends" on page 7559, your knowledge of the spatial structure of the ozoneSet data set is limited to the selected trend-free direction you indicated in the present example. You can generalize this outcome for all spatial directions only if you consider the hypothesis of isotropy in the ozone field to be reasonable. Note that you cannot infer the assumption of anisotropy in the present example based on the analysis in the section "Analysis with Surface Trend Removal" on page 7578. Again, the reason is that you currently use the observed Ozone values, whereas the ResidualOzone data in the previous example emerged from a transformation of the current data. Hence, you have essentially two data sets that do not necessarily share the same properties.

Example 95.4: Covariogram and Semivariogram

The covariance that was reviewed in the section "Stationarity" on page 7546 is an alternative measure of spatial continuity that can be used instead of the semivariance. In a similar manner to the empirical semivariance that was presented in the section "Theoretical and Computational Details of the Semivariogram" on page 7544, you can also compute the empirical covariance. The covariograms are plots of this quantity and can be used to fit permissible theoretical covariance models, in correspondence to the semivariogram analysis presented in the section "Theoretical Semivariogram Models" on page 7540. This example displays a comparative view of the empirical covariogram and semivariogram, and examines some additional aspects of these two measures.

You consider 500 simulations of an SRF $Z(\boldsymbol{s})$ in a square domain of 100×100 (10^6 km^2). The following DATA step defines the data locations:

```
title 'Covariogram and Semivariogram Example';
data dataCoord;
   retain seed 837591;
   do i=1 to 100;
      East = round(100*ranuni(seed),0.1);
      North = round(100*ranuni(seed),0.1);
      output;
   end;
run;
```

For the simulations you use PROC SIM2D, which produces Gaussian simulations of SRFs with user-specified covariance structure—see Chapter 79, "The SIM2D Procedure." The Gaussian SRF

implies full knowledge of the SRF expected value $\mathrm{E}[Z(\boldsymbol{s})]$ and variance $\mathrm{Var}[Z(\boldsymbol{s})]$ at every location $\boldsymbol{s}$. The following statements simulate an isotropic, second-order stationary SRF with constant expected value and variance throughout the simulation domain:

```
ods graphics on;

proc sim2d outsim=dataSims;
   simulate numreal=500 seed=79750
            nugget=2 scale=6 range=10 form=exp;
   mean 30;
   grid gdata=dataCoord xc=East yc=North;
run;
```

Here, the SIMULATE statement accommodates the simulation parameters. The NUMREAL= option specifies that you want to perform 500 simulations, and the SEED= option specifies the seed for the simulation random number generator. You use the MEAN statement to specify the expected value $\mathrm{E}[Z(\boldsymbol{s})] = 30$ units of Z. You also specify two variance components. The first is the nugget effect, and you use the NUGGET= option to set it to $c_n = 2$. The second is the partial sill ${\sigma_0}^2 = 6$ that you specify with the SCALE= option. The two variance components make up the total SRF variance $\mathrm{Var}[Z(\boldsymbol{s})] = c_n + {\sigma_0}^2 = 8$. You assume an exponential covariance structure to describe the field spatial continuity, where ${\sigma_0}^2$ is the sill value, and its range $a_0 = 10$ km (effective range $a_\epsilon = 3a_0 = 30$ km) is specified by the RANGE= option. The option FORM= specifies the covariance structure type.

The empirical semivariance and covariance are computed by the VARIOGRAM procedure, and are available either in the ODS output semivariogram table (as variables Semivariance and Covariance, respectively) or in the OUTVAR= data set. In the following statements you obtain these variables by using the OUTVAR= data set of the VARIOGRAM procedure:

```
proc variogram data=dataSims outv=outv noprint;
   compute lagd=3 maxlag=18;
   coord xc=gxc yc=gyc;
   by _ITER_;
   var svalue;
run;
```

For each distance lag you take the average of the empirical measures over the number of simulations. PROC SORT is used to prepare the input data for PROC MEANS, which produces these averages and stores them in the dataAvgs data set. This sequence is performed with the following statements:

```
proc sort data=outv;
   by lag;
run;

proc means data=outv n mean noprint;
   var Distance variog covar;
   by lag;
   output out=dataAvgs mean(variog)=Semivariance
                       mean(covar)=Covariance
                       mean(Distance)=Distance;
run;
```

The SGPLOT procedure is used to create the plot of the average empirical semivariogram and covariogram, as in the following statements:

```
proc sgplot data=dataAvgs;
   title "Empirical Semivariogram and Covariogram";
   xaxis label = "Distance" grid;
   yaxis  label = "Semivariance" min=-0.5 max=9 grid;
   y2axis label = "Covariance"   min=-0.5 max=9;
   scatter y=Semivariance x=Distance /
           markerattrs = GraphData1
           name='Semivar'
           legendlabel='Semivariance';
   scatter y=Covariance x=Distance /
           y2axis
           markerattrs = GraphData2
           name='Covar'
           legendlabel='Covariance';
   discretelegend 'Semivar' 'Covar';
run;

ods graphics off;
```

The plot of the average empirical semivariance and covariance of the preceding analysis is shown in Output 95.4.1. Note that the high number of simulations led to averages of empirical continuity measures that accurately approximate the simulated SRF characteristics. Specifically, the empirical semivariogram and covariogram both exhibit clearly exponential behavior. The semivariogram sill is approximately at the specified variance $\mathrm{Var}[Z(\boldsymbol{s})] = 8$ of the SRF.

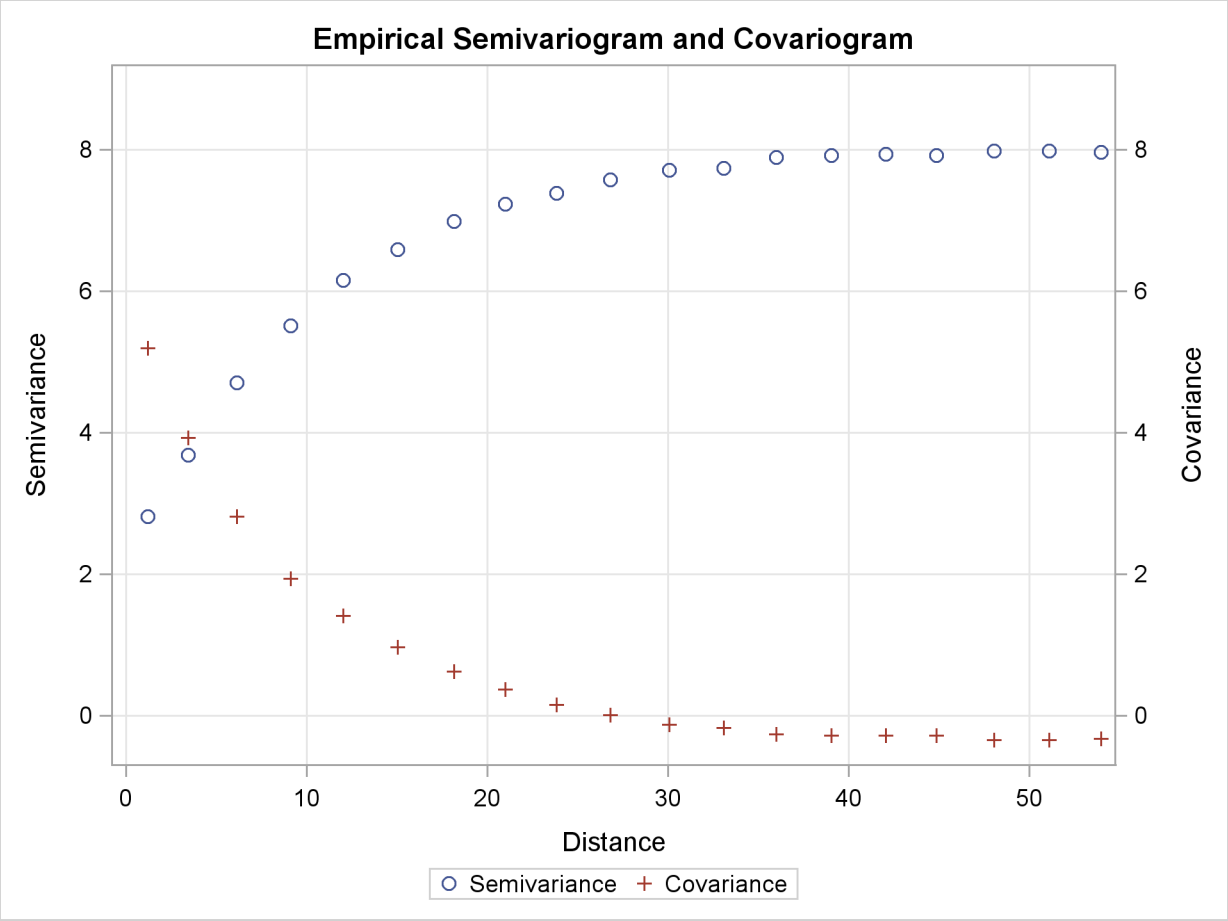
The simulated SRF is second-order stationary, so you expect at each lag the sum of the empirical semivariance and covariance to approximate the field variance $\mathrm{Var}[Z(\boldsymbol{s})]$, as explained in the section "Stationarity" on page 7546. This behavior is evident in Output 95.4.1.

We conclude with a discussion of basic reasons why the empirical semivariogram analysis is commonly preferred to the empirical covariance analysis. A first reason comes from the assumptions that are necessary to compute each of these two measures. The condition of intrinsic stationarity that is required in order to define the empirical semivariogram is less restrictive than the condition of second-order stationarity that is required in order to consider the covariance function as a parameter of the process.

Also, an empirical semivariogram can indicate whether a nugget effect is present in your data sample, whereas the empirical covariogram itself might not reveal this information. This point is illustrated in Output 95.4.1, where you expect to see that $C(\mathbf{0}) = \mathrm{Var}[Z(\boldsymbol{s})]$, but the empirical covariogram cannot have a point at exactly $\boldsymbol{h} = \mathbf{0}$. There is a practical way to investigate for a nugget effect when you use empirical covariograms. Recall that the OUTVAR= data set provides you with the sample variance (shown in the COVAR column for LAG=-1), as the following statement shows:

```
/* Obtain the sample variance from the data set ----------------*/

proc print data=dataAvgs (obs=1);
```

Output 95.4.1 Average Empirical Semivariogram and Covariogram from 500 Simulations



Output 95.4.2 is a partial output of the dataAvgs data set, which contains averages of the OUTVAR= data set, and shows the computed average $C(\mathbf{0})$ in the Covariance column. The combination of the empirical covariogram and the $C(\mathbf{0})$ value can help you fit a theoretical covariance model that will include any nugget effect, if present. See also the discussion in Schabenberger and Gotway (2005, Section 4.2.2) about the Matérn definition of the covariance function that is related to this issue. In particular, this definition provides for an additional variance component in the covariance expression at $\boldsymbol{h} = \mathbf{0}$ to account for the corresponding nugget effect in the semivariogram.

Output 95.4.2 Partial Outcome of the dataAvgs Data Set

Empirical Semivariogram and Covariogram

Obs	LAG	_TYPE_	_FREQ_	Semivariance	Covariance	Distance
1	-1	0	500	.	7.74832	.

In addition to the preceding points, if the SRF is nonstationary, the empirical semivariogram indicates that the SRF variance will increase with distance $\boldsymbol{h}$, as Output 95.3.1 shows in "Example 95.3: Analysis without Surface Trend Removal" on page 7587. In that case it makes no sense to compute the empirical covariogram. Specifically, the covariogram could provide you with an estimate of the sample variance, which is not sufficient to indicate that the SRF might not be stationary (see also Chilès and Delfiner 1999, p. 31).

Finally, the definitions of the empirical semivariance and covariance in the section "Theoretical and Computational Details of the Semivariogram" on page 7544 clearly show that the sample mean $\bar{Z}$ and the SRF expected value $\mathrm{E}[Z(\boldsymbol{s})]$ are not important for the computation of the semivariance, but either one is necessary for the covariance. Hence, the semivariance expression filters the mean, which is especially useful when it is unknown. On the other hand, if $\mathrm{E}[Z(\boldsymbol{s})]$ is unknown and the empirical covariance is computed based on the sample mean $\bar{Z}$, this can induce additional bias in the covariance computation.

Example 95.5: A Box Plot of the Square Root Difference Cloud

The Gaussian form chosen for the semivariogram in the section "Getting Started: VARIOGRAM Procedure" on page 7519 is based on consideration of the plots of the sample semivariogram. For the coal thickness data, the Gaussian form appears to be a reasonable choice.

However, it can often happen that a plot of the sample variogram shows so much scatter that no particular form is evident. The cause of this scatter can be one or more outliers in the pairwise differences of the measured quantities.

A method of identifying potential outliers is discussed in Cressie (1993, Section 2.2.2). This example illustrates how to use the OUTPAIR= data set from PROC VARIOGRAM to produce a square root difference cloud, which is useful in detecting outliers.

For the SRF $Z(\boldsymbol{s}), \boldsymbol{s} \in \mathcal{R}^2$, the square root difference cloud for a particular direction $\boldsymbol{e}$ is given by

$$| Z(s_i + he) - Z(s_i) |^{\frac{1}{2}}$$

for a given lag distance h. In the actual computation, all pairs $P_1 P_2$ of points P_1, P_2 within a distance tolerance around h and an angle tolerance around the direction $\boldsymbol{e}$ are used. This generates a number of point pairs for each lag class h. The spread of these values gives an indication of outliers.

Following the example in the section "Getting Started: VARIOGRAM Procedure" on page 7519, this example uses a basic LAGDISTANCE=7, with a distance tolerance of 3.5, and a direction of N–S, with an angle tolerance ATOL=30°.

First, use PROC VARIOGRAM to produce an OUTPAIR= data set. Then use a DATA step to subset this data by choosing pairs within 30° of N–S. In addition, compute lag class and square root difference variables, as the following statements show:

```
title 'Square Root Difference Cloud Example';
data thick;
   input East North Thick @@;
   label Thick='Coal Seam Thickness';
   datalines;
    0.7  59.6  34.1   2.1  82.7  42.2   4.7  75.1  39.5
    4.8  52.8  34.3   5.9  67.1  37.0   6.0  35.7  35.9
    6.4  33.7  36.4   7.0  46.7  34.6   8.2  40.1  35.4
   13.3   0.6  44.7  13.3  68.2  37.8  13.4  31.3  37.8
   17.8   6.9  43.9  20.1  66.3  37.7  22.7  87.6  42.8
   23.0  93.9  43.6  24.3  73.0  39.3  24.8  15.1  42.3
   24.8  26.3  39.7  26.4  58.0  36.9  26.9  65.0  37.8
   27.7  83.3  41.8  27.9  90.8  43.3  29.1  47.9  36.7
   29.5  89.4  43.0  30.1   6.1  43.6  30.8  12.1  42.8
   32.7  40.2  37.5  34.8   8.1  43.3  35.3  32.0  38.8
   37.0  70.3  39.2  38.2  77.9  40.7  38.9  23.3  40.5
   39.4  82.5  41.4  43.0   4.7  43.3  43.7   7.6  43.1
   46.4  84.1  41.5  46.7  10.6  42.6  49.9  22.1  40.7
   51.0  88.8  42.0  52.8  68.9  39.3  52.9  32.7  39.2
   55.5  92.9  42.2  56.0   1.6  42.7  60.6  75.2  40.1
   62.1  26.6  40.1  63.0  12.7  41.8  69.0  75.6  40.1
   70.5  83.7  40.9  70.9  11.0  41.7  71.5  29.5  39.8
   78.1  45.5  38.7  78.2   9.1  41.7  78.4  20.0  40.8
   80.5  55.9  38.7  81.1  51.0  38.6  83.8   7.9  41.6
   84.5  11.0  41.5  85.2  67.3  39.4  85.5  73.0  39.8
   86.7  70.4  39.6  87.2  55.7  38.8  88.1   0.0  41.6
   88.4  12.1  41.3  88.4  99.6  41.2  88.8  82.9  40.5
   88.9   6.2  41.5  90.6   7.0  41.5  90.7  49.6  38.9
   91.5  55.4  39.0  92.9  46.8  39.1  93.4  70.9  39.7
   55.8  50.5  38.1  96.2  84.3  40.3  98.2  58.2  39.5
   ;

ods graphics on;

proc variogram data=thick outp=outp noprint;
   compute novariogram;
   coordinates xc=East yc=North;
   var Thick;
run;

data sqroot;
   set outp;
   /*- Include only points +/- 30 degrees of N-S -------*/
   where abs(cos) < 0.5;
   /*- Unit lag of 7, distance tolerance of 3.5 --------*/
   lag_class=int(distance/7 + 0.5000001);
   sqr_diff=sqrt(abs(v1-v2));
run;

proc sort data=sqroot;
   by lag_class;
run;
```

Next, summarize the results by using the MEANS procedure. The statements follow, and the output is shown in Output 95.5.1.

```
proc means data=sqroot noprint n mean std;
   var sqr_diff;
   by lag_class;
   output out=msqrt n=n mean=mean std=std;
run;
title2 'Summary of Results';

proc print data=msqrt;
   id lag_class;
   var n mean std;
run;
```

Output 95.5.1 Summary of Results

Square Root Difference Cloud Example
Summary of Results

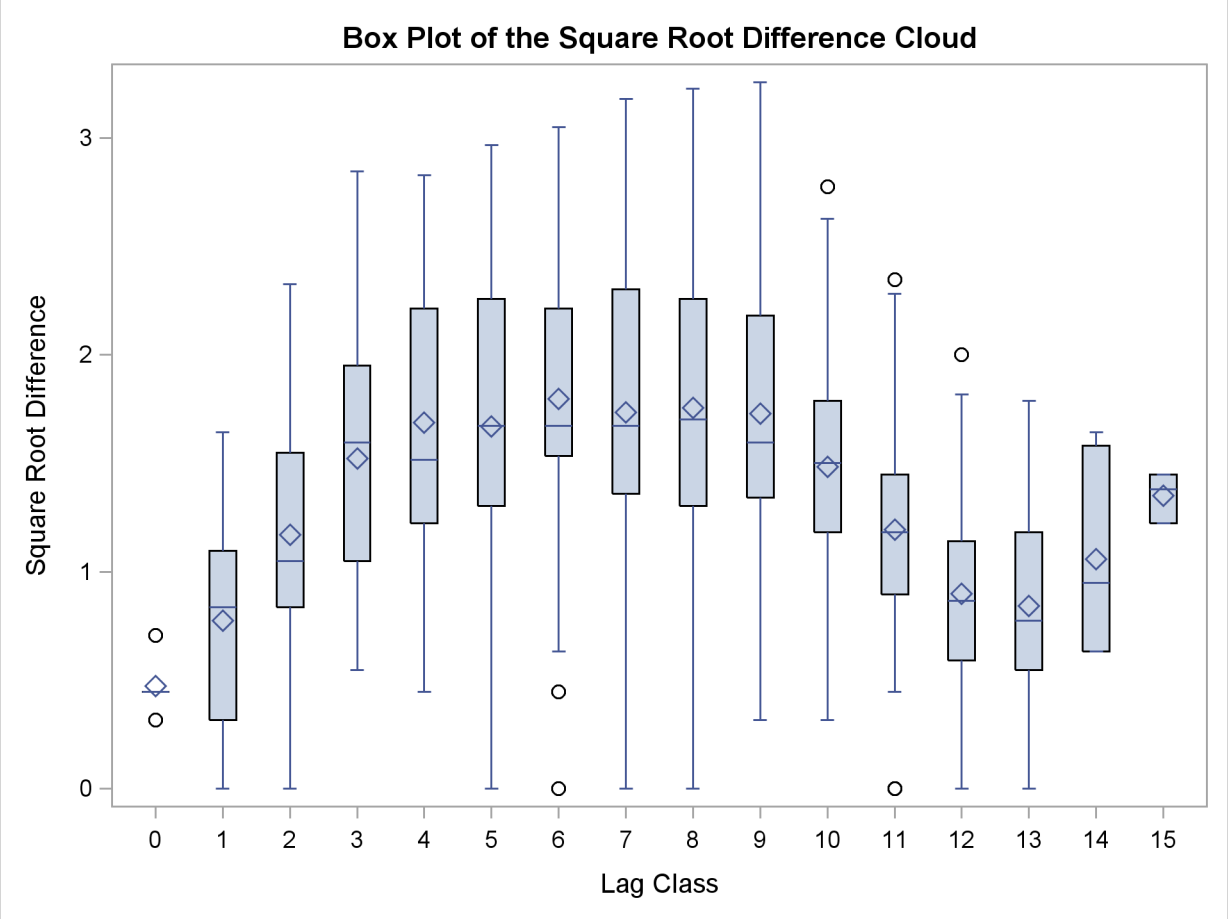
lag_class	n	mean	std
0	5	0.47300	0.14263
1	31	0.77338	0.41467
2	51	1.17052	0.47800
3	58	1.52287	0.51454
4	65	1.68625	0.58465
5	65	1.66963	0.68582
6	80	1.79693	0.62929
7	88	1.73334	0.73191
8	83	1.75528	0.68767
9	108	1.72901	0.58274
10	80	1.48268	0.48695
11	84	1.19242	0.47037
12	68	0.89765	0.42510
13	38	0.84223	0.44249
14	7	1.05653	0.42548
15	3	1.35076	0.11472

Finally, present the results in a box plot by using the SGPLOT procedure. The box plot facilitates the detection of outliers. The statements are as follows:

```
proc sgplot data=sqroot;
   xaxis label = "Lag Class";
   yaxis label = "Square Root Difference";
   title "Box Plot of the Square Root Difference Cloud";
   vbox sqr_diff /category=lag_class;
run;

ods graphics off;
```

Output 95.5.2 suggests that there do not appear to be any outliers adversely affecting the empirical semivariogram in the N–S direction for the coal seam thickness data. The conclusion from Output 95.5.2 is consistent with our previous semivariogram analysis of the same data set in the section "Getting Started: VARIOGRAM Procedure" on page 7519. The effect of the isolated outliers in lag classes 6 and 10–12 in Output 95.5.2 is demonstrated as the divergence between the classical and robust empirical semivariance estimates in the higher distances in Output 95.8. The difference in these estimates comes from the definition of the robust semivariance estimator $\bar{\gamma}_z(\boldsymbol{h})$ (see the section "Theoretical and Computational Details of the Semivariogram" on page 7544), which imposes a smoothing effect on the outlier influence.

Output 95.5.2 Box Plot of the Square Root Difference Cloud



References

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2004), *Hierarchical Modeling and Analysis for Spatial Data*, Boca Raton, FL: Chapman & Hall/CRC.

Chilès, J. P. and Delfiner, P. (1999), *Geostatistics-Modeling Spatial Uncertainty*, New York: John Wiley & Sons.

Christakos, G. (1992), *Random Field Models in Earth Sciences*, New York: Academic Press.

Cliff, A. D. and Ord, J. K. (1981), *Spatial Processes: Models and Applications*, London: Pion Ltd.

Cressie, N. and Hawkins, D. M. (1980), "Robust Estimation of the Variogram: I," *Mathematical Geology*, 12(2), 115–125.

Cressie, N. A. C. (1993), *Statistics for Spatial Data*, New York: John Wiley & Sons.

Deutsch, C. V. and Journel, A. G. (1992), *GSLIB: Geostatistical Software Library and User's Guide*, New York: Oxford University Press.

Geary, R. C. (1954), "The Contiguity Ratio and Statistical Mapping," *The Incorporated Statistician*, 5, 115–145.

Goovaerts, P. (1997), *Geostatistics for Natural Resources Evaluation*, New York: Oxford University Press.

Hohn, M. (1988), *Geostatistics and Petroleum Geology*, New York: Van Nostrand Reinhold.

Jian, X., Olea, R. A., and Yu, Y.-S. (1996), "Semivariogram Modeling by Weighted Least Squares," *Computers & Geosciences*, 22(4), 387–397.

Journel, A. G. and Huijbregts, C. J. (1978), *Mining Geostatistics*, New York: Academic Press.

Matheron, G. (1963), "Principles of Geostatistics," *Economic Geology*, 58, 1246–1266.

Moran, P. A. P. (1950), "Notes on Continuous Stochastic Phenomena," *Biometrica*, 37, 17–23.

Olea, R. A. (1999), *Geostatistics for Engineers and Earth Scientists*, Boston: Kluwer Academic.

Olea, R. A. (2006), "A Six-Step Practical Approach to Semivariogram Modeling," *Stochastic Environmental Research and Risk Assessment*, 20(5), 307–318.

Schabenberger, O. and Gotway, C. A. (2005), *Statistical Methods for Spatial Data Analysis*, Boca Raton, FL: Chapman & Hall/CRC.

Stein, M. L. (1988), "Asymptotically Efficient Prediction of a Random Field with a Misspecified Covariance Function," *Annals of Statistics*, 16, 55–63.

Appendix A

Special SAS Data Sets

Contents

Introduction to Special SAS Data Sets **7605**
Special SAS Data Sets **7609**
TYPE=ACE Data Sets 7609
TYPE=BOXPLOT Data Sets 7609
TYPE=CHARTSUM Data Sets 7609
TYPE=CORR Data Sets 7609
TYPE=COV Data Sets 7613
TYPE=CSSCP Data Sets 7613
TYPE=DISTANCE Data Sets 7614
TYPE=EST Data Sets 7614
TYPE=FACTOR Data Sets 7615
TYPE=LINEAR Data Sets 7616
TYPE=LOGISMOD Data Sets 7616
TYPE=MIXED Data Sets 7616
TYPE=QUAD Data Sets 7616
TYPE=RAM Data Sets 7617
TYPE=SSCP Data Sets 7617
TYPE=TREE Data Sets 7618
TYPE=UCORR Data Sets 7618
TYPE=UCOV Data Sets 7619
TYPE=WEIGHT Data Sets 7619
Definitional Formulas **7619**

Introduction to Special SAS Data Sets

All SAS/STAT procedures create SAS data sets. Any table generated by a procedure can be saved to a data set by using the Output Delivery System (ODS), and many procedures also have syntax that enables you to save other statistics to data sets. Some of these data sets are organized according to certain conventions so that they can be read by a SAS/STAT procedure for further analysis. Such specially organized data sets are recognized by the TYPE= data set attribute.

The CORR procedure (see the *Base SAS Procedures Guide: Statistical Procedures*), for example, can create a data set with the attribute TYPE=CORR containing a correlation matrix. This TYPE=CORR data set can be read by the REG or FACTOR procedure, among others. If the original data set is large, using a special SAS data set in this way can save computer time by avoiding the recomputation of the correlation matrix in subsequent analyses.

PROC REG, for example, can create a TYPE=EST data set containing estimated regression coefficients. If you need to make predictions for new observations, you can use the SCORE procedure to read both the TYPE=EST data set and a data set containing the new observations. PROC SCORE can then compute predicted values or residuals without repeating the entire regression analysis. See Chapter 76, "The SCORE Procedure," for an example.

A special SAS data set might contain different kinds of statistics. A special variable called _TYPE_ is used to distinguish the various statistics. For example, in a TYPE=CORR data set, an observation in which _TYPE_='MEAN' contains the means of the variables in the analysis, and an observation in which _TYPE_='STD' contains the standard deviations. Correlations appear in observations with _TYPE_='CORR'. Another special variable, _NAME_, is needed to identify the row of the correlation matrix. Thus, the correlation between variables X and Y is given by the value of the variable X in the observation for which _TYPE_='CORR' and _NAME_='Y', or by the value of the variable Y in the observation for which _TYPE_='CORR' and _NAME_='X'.

The special data sets created by SAS/STAT procedures can generally be used directly by other procedures without modification. However, if you create an output data set with PROC CORR and use the NOCORR option to omit the correlation matrix from the OUT= data set, you need to set the TYPE= option either in parentheses following the OUT= data set name in the PROC CORR statement or in parentheses following the DATA= option in any other procedure that recognizes the special TYPE= attribute. In either case, the TYPE= option should be set to COV, CSSCP, or SSCP according to what type of matrix is stored in the data set and what data set types are accepted as input by the other procedures you plan to use. If you do not follow these steps and you use the TYPE=CORR data set with no correlation matrix as input to another procedure, the procedure might issue an error message indicating that the correlation matrix is missing from the data set.

You can create special SAS data sets directly in a DATA step by specifying the TYPE= option in parentheses after the data set name in the DATA statement. See "Example A.2: Creating a TYPE=CORR Data Set in a DATA Step" on page 7612 for an example. If you use a DATA step with a SET statement to modify a special SAS data set, you must specify the TYPE= option in the DATA statement. The TYPE= attribute of the data set in the SET statement is *not* automatically copied to the data set being created. You can determine the TYPE= attribute of a data set by using the CONTENTS procedure (see "Example A.1: A TYPE=CORR Data Set Produced by PROC CORR" on page 7611 and the *Base SAS Procedures Guide* for details).

Table A.1 summarizes the TYPE= data sets that can be used as input to SAS/STAT procedures. Table A.2 summarizes the TYPE= data sets that are created by SAS/STAT procedures and the statements each procedure uses to create its special output data sets. Most procedures accept ordinary SAS data sets and create ordinary output SAS data sets with no TYPE= specification in addition to the special data sets shown in the tables. When you specify a data set with a type that the procedure does not recognize, the procedure prints an error message and stops executing.

Table A.1 SAS/STAT Procedures That Accept Special Input Data Sets Types

Procedure	Special TYPE= Data Sets Accepted
ACECLUS	ACE, CORR, COV, SSCP, UCORR, UCOV
BOXPLOT	BOXPLOT, CHARTSUM
CALIS	CORR, COV, FACTOR, RAM, SSCP, UCORR, UCOV, WEIGHT
CANDISC	CORR, COV, SSCP, CSSCP
CATMOD	EST
CLUSTER	DISTANCE
DISCRIM	CORR, COV, SSCP, CSSCP, LINEAR, QUAD, MIXED
FACTOR	ACE, CORR, COV, FACTOR, SSCP, UCORR, UCOV
LIFEREG	EST
LOGISTIC	EST LOGISMOD
MI	EST, COV, CORR
MIANALYZE	EST, COV, CORR
MODECLUS	DISTANCE
PHREG	EST
PRINCOMP	ACE, CORR, COV, EST, FACTOR, SSCP, UCORR, UCOV
PROBIT	EST
QUANTREG	EST
REG	CORR, COV, SSCP, UCORR, UCOV
ROBUSTREG	EST
SCORE	SCORE= data set can be of any type
SIMNORM	CORR, COV
SURVEYLOGISTIC	EST
STEPDISC	CORR, COV, SSCP, CSSCP
TREE	TREE
VARCLUS	CORR, COV, FACTOR, SSCP, UCORR, UCOV

Table A.2 SAS/STAT Procedures That Create Special Output Data Set Types

Procedure	TYPE=	Statement and Option Required
ACECLUS	ACE	PROC ACECLUS OUTSTAT=
BOXPLOT	BOXPLOT	PLOT / OUTBOX=
	CHARTSUM	PLOT / OUTHISTORY=
CALIS	CORR	PROC CALIS OUTSTAT=
	COV	PROC CALIS COV OUTSTAT=
	EST	PROC CALIS OUTEST=
	UCORR	PROC CALIS NOINT OUTSTAT=
	UCOV	PROC CALIS NOINT COV OUTSTAT=
	RAM	PROC CALIS OUTRAM=
	WEIGHT	PROC CALIS OUTWGT=
CANCORR	CORR	PROC CANCORR OUTSTAT=
	UCORR	PROC CANCORR NOINT OUTSTAT=
CANDISC	CORR	PROC CANDISC OUTSTAT=

Table A.2 *continued*

Procedure	TYPE=	Statement and Option Required
CATMOD	EST	RESPONSE / OUTEST=
CLUSTER	TREE	PROC CLUSTER OUTTREE=
DISCRIM	LINEAR	PROC DISCRIM POOL=YES OUTSTAT=
	QUAD	PROC DISCRIM POOL=NO OUTSTAT=
	MIXED	PROC DISCRIM POOL=TEST OUTSTAT=
	CORR	PROC DISCRIM METHOD=NPAR OUTSTAT=
DISTANCE	DISTANCE	PROC DISTANCE METHOD=*distance-method* OUT=
	SIMILAR	PROC DISTANCE METHOD=*similarity-method* OUT=
FACTOR	FACTOR	PROC FACTOR OUTSTAT=
LIFEREG	EST	PROC LIFEREG OUTEST=
LOGISTIC	EST	PROC LOGISTIC OUTEST=
	LOGISMOD	PROC LOGISTIC OUTMODEL=
MI	COV	EM OUTEM=
	COV	EM OUTITER=
	COV	MCMC OUTITER=
	EST	MCMC OUTEST=
NLIN	EST	PROC NLIN OUTEST=
ORTHOREG	EST	PROC ORTHOREG OUTEST=
PHREG	EST	PROC PHREG OUTEST=
PRINCOMP	CORR	PROC PRINCOMP OUTSTAT=
	COV	PROC PRINCOMP COV OUTSTAT=
	UCORR	PROC PRINCOMP NOINT OUTSTAT=
	UCOV	PROC PRINCOMP NOINT COV OUTSTAT=
PROBIT	EST	PROC PROBIT OUTEST=
QUANTREG	EST	PROC QUANTREG OUTEST=
REG	EST	PROC REG OUTEST=
	SSCP	PROC REG OUTSSCP=
ROBUSTREG	EST	PROC ROBUSTREG OUTEST=
VARCLUS	CORR	PROC VARCLUS OUTSTAT=
	UCORR	PROC VARCLUS NOINT OUTSTAT=
	TREE	PROC VARCLUS OUTTREE=

Special SAS Data Sets

TYPE=ACE Data Sets

A TYPE=ACE data set is created by the ACECLUS procedure, and it contains the approximate within-cluster covariance estimate, as well as eigenvalues and eigenvectors from a canonical analysis, among other statistics. It can be used as input to the ACECLUS procedure to initialize another execution of PROC ACECLUS. It can also be used to compute canonical variable scores with PROC SCORE and as input to PROC FACTOR, specifying METHOD=SCORE, to rotate the canonical variables. See Chapter 22, "The ACECLUS Procedure," for details.

TYPE=BOXPLOT Data Sets

A TYPE=BOXPLOT data set is created by and used by the BOXPLOT procedure. The data set contains the group summary statistics and outlier values required for constructing a schematic box plot. Each observation in a TYPE=BOXPLOT data set records the value of a single feature of one group's box-and-whiskers plot, such as its mean. Consequently, a TYPE=BOXPLOT data set contains multiple observations per group. These must appear consecutively in the data set. The _TYPE_ variable identifies the feature whose value is recorded in a given observation. _TYPE_ values of 'N', 'MIN', 'Q1', 'MEDIAN', 'MEAN', 'Q3', and 'MAX 'are required for each group. See Chapter 24, "The BOXPLOT Procedure," for details.

TYPE=CHARTSUM Data Sets

A TYPE=CHARTSUM data set is created by and used by the BOXPLOT procedure. The data set contains group summary statistics associated with box-and-whiskers plots. See Chapter 24, "The BOXPLOT Procedure," for details.

TYPE=CORR Data Sets

A TYPE=CORR data set usually contains a correlation matrix and possibly other statistics including means, standard deviations, and the number of observations in the original SAS data set from which the correlation matrix was computed. Using PROC CORR with an output data set option (OUTP=, OUTS=, OUTK=, OUTH=, or OUT=) produces a TYPE=CORR data set. (For a complete description of the CORR procedure, see the *Base SAS Procedures Guide: Statistical Procedures*.) The CALIS, CANCORR, CANDISC, DISCRIM, PRINCOMP, and VARCLUS procedures

can also create a TYPE=CORR data set with additional statistics. A TYPE=CORR data set containing a correlation matrix can be used as input for the ACECLUS, CALIS, CANCORR, CANDISC, DISCRIM, FACTOR, PRINCOMP, REG, SCORE, STEPDISC, and VARCLUS procedures. The variables in a TYPE=CORR data set are as follows:

- the BY variable or variables, if a BY statement is used with the procedure
- _TYPE_, a character variable of length eight with values identifying the type of statistic in each observation, such as 'MEAN', 'STD', 'N', and 'CORR'
- _NAME_, a character variable with values identifying the variable with which a given row of the correlation matrix is associated
- other variables that were analyzed by the CORR procedure or other procedures

The usual values of the _TYPE_ variable are as follows:

TYPE	Contents
MEAN	mean of each variable analyzed
STD	standard deviation of each variable
N	number of observations used in the analysis. PROC CORR records the number of nonmissing values for each variable unless the NOMISS option is used. If the NOMISS option is specified, or if the CALIS, CANCORR, CANDISC, PRINCOMP, or VARCLUS procedure is used to create the data set, observations with one or more missing values are omitted from the analysis, so this value is the same for each variable and provides the number of observations with no missing values. If a FREQ statement is used with the procedure that creates the data set, the number of observations is the sum of the relevant values of the variable in the FREQ statement. Procedures that read a TYPE=CORR data set use the smallest value in the observation with _TYPE_='N' as the number of observations in the analysis.
SUMWGT	sum of the observation weights if a WEIGHT statement is used with the procedure that creates the data set. The values are determined analogously to those of the _TYPE_='N' observation.
CORR	correlations with the variable named by the _NAME_ variable

There might be additional observations in a TYPE=CORR data set depending on the particular procedure and options used.

If you create a TYPE=CORR data set yourself, the data set need not contain the observations with _TYPE_='MEAN', 'STD', 'N', or 'SUMWGT', unless you intend to use one of the discriminant procedures. Procedures assume that all of the means are 0.0 and that the standard deviations are 1.0 if this information is not in the TYPE=CORR data set. If _TYPE_='N' does not appear, most procedures assume that the number of observations is 10,000; significance tests and other statistics that depend on the number of observations are, of course, meaningless. In the CALIS and CANCORR procedures, you can use the EDF= option instead of including a _TYPE_='N' observation.

A correlation matrix is symmetric; that is, the correlation between X and Y is the same as the correlation between Y and X. The CALIS, CANCORR, CANDISC, CORR, DISCRIM, PRINCOMP, and VARCLUS procedures output the entire correlation matrix. If you create the data set yourself, you need to include only one of the two occurrences of the correlation between two variables; the other can be given a missing value.

If you create a TYPE=CORR data set yourself, the _TYPE_ and _NAME_ variables are not necessary except for use with the discriminant procedures and PROC SCORE. If there is no _TYPE_ variable, then all observations are assumed to contain correlations. If there is no _NAME_ variable, the first observation is assumed to correspond to the first variable in the analysis, the second observation to the second variable, and so on. However, if you omit the _NAME_ variable, you will not be able to analyze arbitrary subsets of the variables or list the variables in a VAR or MODEL statement in a different order.

Example A.1: A TYPE=CORR Data Set Produced by PROC CORR

See Figure A.1 for an example of a TYPE=CORR data set produced by the following SAS statements. Figure A.2 displays partial output from PROC CONTENTS, which indicates that the "Data Set Type" is 'CORR'.

```
title 'Five Socioeconomic Variables';
title2 'Harman (1976), Modern Factor Analysis, Third Edition';

data SocEcon;
   input Pop School Employ Services House;
   datalines;
5700     12.8      2500      270       25000
1000     10.9      600       10        10000
3400     8.8       1000      10        9000
3800     13.6      1700      140       25000
4000     12.8      1600      140       25000
8200     8.3       2600      60        12000
1200     11.4      400       10        16000
9100     11.5      3300      60        14000
9900     12.5      3400      180       18000
9600     13.7      3600      390       25000
9600     9.6       3300      80        12000
9400     11.4      4000      100       13000
;

proc corr noprint out=corrcorr;
run;

proc print data=corrcorr;
run;

proc contents data=corrcorr;
run;
```

Figure A.1 A TYPE=CORR Data Set Produced by PROC CORR

Five Socioeconomic Variables
Harman (1976), Modern Factor Analysis, Third Edition

Obs	_TYPE_	_NAME_	Pop	School	Employ	Services	House
1	MEAN		6241.67	11.4417	2333.33	120.833	17000.00
2	STD		3439.99	1.7865	1241.21	114.928	6367.53
3	N		12.00	12.0000	12.00	12.000	12.00
4	CORR	Pop	1.00	0.0098	0.97	0.439	0.02
5	CORR	School	0.01	1.0000	0.15	0.691	0.86
6	CORR	Employ	0.97	0.1543	1.00	0.515	0.12
7	CORR	Services	0.44	0.6914	0.51	1.000	0.78
8	CORR	House	0.02	0.8631	0.12	0.778	1.00

Figure A.2 Contents of a TYPE=CORR Data Set

Five Socioeconomic Variables
Harman (1976), Modern Factor Analysis, Third Edition

The CONTENTS Procedure

Data Set Name	WORK.CORRCORR	Observations	8
Member Type	DATA	Variables	7
Engine	V9	Indexes	0
Created	Thu, Nov 08, 2007 11:01:29 AM	Observation Length	56
Last Modified	Thu, Nov 08, 2007 11:01:29 AM	Deleted Observations	0
Protection		Compressed	NO
Data Set Type	CORR	Sorted	NO
Label	Pearson Correlation Matrix		
Data Representation	WINDOWS_32		
Encoding	wlatin1 Western (Windows)		

Example A.2: Creating a TYPE=CORR Data Set in a DATA Step

This example creates a TYPE=CORR data set by reading a correlation matrix in a DATA step. Figure A.3 shows the resulting data set.

```
title 'Five Socioeconomic Variables';

data datacorr(type=corr);
   infile cards missover;
   _type_='corr';
   input _Name_ $ Pop School Employ Services House;
   datalines;
Pop         1.00000
School      0.00975   1.00000
Employ      0.97245   0.15428   1.00000
Services    0.43887   0.69141   0.51472   1.00000
House       0.02241   0.86307   0.12193   0.77765   1.00000
;
```

```
proc print data=datacorr;
run;
```

Figure A.3 A TYPE=CORR Data Set Created by a DATA Step

Five Socioeconomic Variables

Obs	_type_	_Name_	Pop	School	Employ	Services	House
1	corr	Pop	1.00000	.	.	.	.
2	corr	School	0.00975	1.00000	.	.	.
3	corr	Employ	0.97245	0.15428	1.00000	.	.
4	corr	Services	0.43887	0.69141	0.51472	1.00000	.
5	corr	House	0.02241	0.86307	0.12193	0.77765	1

TYPE=COV Data Sets

A TYPE=COV data set is similar to a TYPE=CORR data set except that it has _TYPE_='COV' observations containing covariances instead of or in addition to _TYPE_='CORR' observations containing correlations. The CALIS and PRINCOMP procedures create a TYPE=COV data set if the COV option is used. You can also create a TYPE=COV data set by using PROC CORR with the COV and NOCORR options and specifying the data set option TYPE=COV in parentheses following the name of the output data set. You can use only the OUTP= or OUT= option to create a TYPE=COV data set with PROC CORR. Another way to create a TYPE=COV data set is to read a covariance matrix in a data set, in the same manner as shown in "Example A.2: Creating a TYPE=CORR Data Set in a DATA Step" on page 7612 for a TYPE=CORR data set. TYPE=COV data sets are used by the same procedures that use TYPE=CORR data sets.

TYPE=CSSCP Data Sets

A TYPE=CSSCP data set contains a corrected sum of squares and crossproducts (CSSCP) matrix. TYPE=CSSCP data sets are created by using the CORR procedure with the CSSCP option and specifying the data set option TYPE=CSSCP in parentheses following the name of the OUTP= or OUT= data set. You can also create TYPE=CSSCP data sets in a DATA step; in this case, TYPE=CSSCP must be specified as a data set option. The variables in a TYPE=CSSCP data set are the same as those found in a TYPE=SSCP data set, except that there is not a variable called Intercept or a row with _NAME_='Intercept'. TYPE=CSSCP data sets are read by only the CANDISC, DISCRIM, and STEPDISC procedures. Formulas useful for illustrating differences between corrected and uncorrected matrices in some special SAS data sets are shown in the section "Definitional Formulas" on page 7619.

TYPE=DISTANCE Data Sets

PROC DISTANCE creates a TYPE=DISTANCE or TYPE=SIMILAR data set, depending on the METHOD= option. TYPE=DISTANCE can be used as an input data set to PROC MODECLUS or PROC CLUSTER, but TYPE=SIMILAR cannot be used as an input to any procedures. The proximity measures are stored as a lower triangular matrix or a square matrix in the OUT= data set (depending on the SHAPE= option). See Chapter 32, "The DISTANCE Procedure," for details. You can also create a TYPE=DISTANCE data set in a DATA step by reading or computing a lower triangular or symmetric matrix of dissimilarity values, such as a chart of mileage between cities. The number of observations must be equal to the number of variables used in the analysis. This type of data set is used as input by the CLUSTER and MODECLUS procedures. PROC CLUSTER ignores the upper triangular portion of a TYPE=DISTANCE data set and assumes that all main diagonal values are zero, even if they are missing. PROC MODECLUS uses the entire distance matrix and does not require the matrix to be symmetric. See Chapter 29, "The CLUSTER Procedure," and Chapter 57, "The MODECLUS Procedure," for examples and details.

TYPE=EST Data Sets

A TYPE=EST data set contains parameter estimates. The CALIS, CATMOD, LIFEREG, LOGISTIC, NLIN, ORTHOREG, PHREG, PROBIT, and REG procedures create TYPE=EST data sets when the OUTEST= option is specified. A TYPE=EST data set produced by PROC LIFEREG, PROC ORTHOREG, or PROC REG can be used with PROC SCORE to compute residuals or predicted values. The variables in a TYPE=EST data set include the following:

- the BY variables, if a BY statement is used
- _TYPE_, a character variable of length eight, that indicates the type of estimate. The values depend on which procedure created the data set. Usually a value of 'PARM' or 'PARMS' indicates estimated regression coefficients, and a value of 'COV' or 'COVB' indicates estimated covariances of the parameter estimates. Some procedures, such as PROC NLIN, have other values of _TYPE_ for special purposes.
- _NAME_, a character variable that contains the values of the names of the rows of the covariance matrix when the procedure outputs the covariance matrix of the parameter estimates
- variables that contain the parameter estimates, usually the same variables that appear in the VAR statement or in any MODEL statement. See Chapter 25, "The CALIS Procedure," Chapter 28, "The CATMOD Procedure," and Chapter 60, "The NLIN Procedure," for details on the variable names used in output data sets created by those procedures.

Other variables can be included depending on the particular procedure and options used.

Example A.3: A TYPE=EST Data Set Produced by PROC REG

Figure A.4 shows the TYPE=EST data set produced by the following statements:

```
proc reg data=SocEcon outest=regest covout;
   full:   model house=pop school employ services / noprint;
   empser: model house=employ services / noprint;
run; quit;

proc print data=regest;
run;
```

Figure A.4 A TYPE=EST Data Set Produced by PROC REG

Five Socioeconomic Variables

Obs	_MODEL_	_TYPE_	_NAME_	_DEPVAR_	_RMSE_	Intercept
1	full	PARMS		House	3122.03	-8074.21
2	full	COV	Intercept	House	3122.03	109408014.44
3	full	COV	Pop	House	3122.03	-9157.04
4	full	COV	School	House	3122.03	-9784744.54
5	full	COV	Employ	House	3122.03	20612.49
6	full	COV	Services	House	3122.03	102764.89
7	empser	PARMS		House	3789.96	15021.71
8	empser	COV	Intercept	House	3789.96	5824096.19
9	empser	COV	Employ	House	3789.96	-1915.99
10	empser	COV	Services	House	3789.96	-1294.94

Obs	Pop	School	Employ	Services	House
1	0.65	2140.10	-2.92	27.81	-1
2	-9157.04	-9784744.54	20612.49	102764.89	.
3	2.32	852.86	-6.20	-5.20	.
4	852.86	907886.36	-2042.24	-9608.59	.
5	-6.20	-2042.24	17.44	6.50	.
6	-5.20	-9608.59	6.50	202.56	.
7	.	.	-1.94	53.88	-1
8	.	.	-1915.99	-1294.94	.
9	.	.	1.15	-6.41	.
10	.	.	-6.41	134.49	.

TYPE=FACTOR Data Sets

A TYPE=FACTOR data set is created by PROC FACTOR when the OUTSTAT= option is specified. The CALIS, CANCORR, FACTOR, PRINCOMP, SCORE, and VARCLUS procedures can use TYPE=FACTOR data sets as input. The variables are the same as in a TYPE=CORR data set. The statistics include means, standard deviations, sample size, correlations, eigenvalues, eigenvectors, factor patterns, residual correlations, scoring coefficients, and others depending on the options specified. See Chapter 33, "The FACTOR Procedure," for details. When the NOINT option is used

with the OUTSTAT= option in PROC FACTOR, the value of the _TYPE_ variable is set to 'USCORE' instead of 'SCORE' to indicate that the scoring coefficients have not been corrected for the mean. If this data set is used with PROC SCORE, the value of the _TYPE_ variable tells PROC SCORE whether or not to subtract the mean from the scoring coefficients.

TYPE=LINEAR Data Sets

A TYPE=LINEAR data set contains the coefficients of a linear function of the variables in observations with _TYPE_='LINEAR'. PROC DISCRIM stores linear discriminant function coefficients in a TYPE=LINEAR data set when you specify METHOD=NORMAL (the default method), POOL=YES, and an OUTSTAT= data set; the data set can be used in a subsequent invocation of PROC DISCRIM to classify additional observations. Many other statistics can be included depending on the options used. See Chapter 31, "The DISCRIM Procedure," for details.

TYPE=LOGISMOD Data Sets

A TYPE=LOGISMOD data set contains information about a logistic regression model fit by PROC LOGISTIC. PROC LOGISTIC both creates and reads TYPE=LOGISMOD data sets. See Chapter 51, "The LOGISTIC Procedure," for details.

TYPE=MIXED Data Sets

A TYPE=MIXED data set contains coefficients of either a linear or a quadratic function, or both if there are BY groups. PROC DISCRIM produces a TYPE=MIXED data set when you specify METHOD=NORMAL (the default method), POOL=TEST, and an OUTSTAT= data set. See Chapter 31, "The DISCRIM Procedure," for details.

TYPE=QUAD Data Sets

A TYPE=QUAD data set contains the coefficients of a quadratic function of the variables in observations with _TYPE_='QUAD'. PROC DISCRIM stores quadratic discriminant function coefficients in a TYPE=QUAD data set when you specify METHOD=NORMAL (the default method), POOL=NO, and an OUTSTAT= data set; the data set can be used in a subsequent invocation of PROC DISCRIM to classify additional observations. Many other statistics can be included depending on the options used. See Chapter 31, "The DISCRIM Procedure," for details.

TYPE=RAM Data Sets

PROC CALIS creates and accepts as input a TYPE=RAM data set. This data set contains the model specification and the computed parameter estimates. A TYPE=RAM data set is intended to be reused as an input data set to specify good initial values in subsequent analyses by PROC CALIS. See Chapter 25, "The CALIS Procedure," for details.

TYPE=SSCP Data Sets

A TYPE=SSCP data set contains an uncorrected sum of squares and crossproducts (SSCP) matrix. TYPE=SSCP data sets are produced by PROC REG when the OUTSSCP= option is specified in the PROC REG statement. You can also create a TYPE=SSCP data set by using PROC CORR with the SSCP option and specifying the data set option TYPE=SSCP in parentheses following the name of the OUTP= or OUT= data set. You can also create TYPE=SSCP data sets in a DATA step; in this case, TYPE=SSCP must be specified as a data set option.

The variables in a TYPE=SSCP data set include those found in a TYPE=CORR data set. In addition, there is a variable called Intercept that contains crossproducts for the intercept (sums of the variables). The SSCP matrix is stored in observations with _TYPE_='SSCP', including a row with _NAME_='Intercept'. PROC REG also outputs an observation with _TYPE_='N'. PROC CORR includes observations with _TYPE_='MEAN' and _TYPE_='STD' as well. TYPE=SSCP data sets are used by the same procedures that use TYPE=CORR data sets.

Example A.4: A TYPE=SSCP Data Set Produced by PROC REG

The following statements create a TYPE=SSCP data set from the SocEcon input data set created in "Example A.1: A TYPE=CORR Data Set Produced by PROC CORR" on page 7611:

```
proc reg data=SocEcon outsscp=regsscp;
   model house=pop school employ services / noprint;
run; quit;

proc print data=regsscp;
run;
```

The data set is created by PROC REG and is displayed in Figure A.5.

Figure A.5 A TYPE=SSCP Data Set Produced by PROC REG

Five Socioeconomic Variables

Obs	_TYPE_	_NAME_	Intercept	Pop	School	Employ	Services	House
1	SSCP	Intercept	12.0	74900	137.30	28000	1450	204000
2	SSCP	Pop	74900.0	597670000	857640.00	220440000	10959000	1278700000
3	SSCP	School	137.3	857640	1606.05	324130	18152	2442100
4	SSCP	Employ	28000.0	220440000	324130.00	82280000	4191000	486600000
5	SSCP	Services	1450.0	10959000	18152.00	4191000	320500	30910000
6	SSCP	House	204000.0	1278700000	2442100.00	486600000	30910000	3914000000
7	N		12.0	12	12.00	12	12	12

TYPE=TREE Data Sets

Some clustering procedures produce TYPE=TREE data sets. For example, in PROC CLUSTER, a TYPE=TREE data set contains one observation for each observation in the input data set, plus one observation for each cluster of two or more observations (that is, one observation for each node of the cluster tree). The total number of output observations is usually $2n - 1$, where n is the number of input observations. The density methods might produce fewer output observations when the number of clusters cannot be reduced to one.

In PROC VARCLUS, the OUTTREE= data set contains one observation for each variable clustered plus one observation for each cluster of two or more variables—that is, one observation for each node of the cluster tree. The total number of output observations is between n and $2n - 1$, where n is the number of variables clustered. See Chapter 29, "The CLUSTER Procedure," and Chapter 93, "The VARCLUS Procedure," for details.

TYPE=UCORR Data Sets

A TYPE=UCORR data set is almost identical to a TYPE=CORR data set, except that the correlations are uncorrected for the mean. The corresponding value of the _TYPE_ variable is 'UCORR' instead of 'CORR'. Uncorrected standard deviations are in observations with _TYPE_='USTD'. A TYPE=UCORR data set can be used as input for every SAS/STAT procedure that uses a TYPE=CORR data set, except for the CANDISC, DISCRIM, and STEPDISC procedures. TYPE=UCORR data sets can be created by the CALIS, CANCORR, PRINCOMP, and VARCLUS procedures.

TYPE=UCOV Data Sets

A TYPE=UCOV data set is similar to a TYPE=COV data set, except that the covariances are uncorrected for the mean. Also, the corresponding value of the _TYPE_ variable is 'UCOV' instead of 'COV'. A TYPE=UCOV data set can be used as input for every SAS/STAT procedure that uses a TYPE=COV data set, except for the CANDISC, DISCRIM, and STEPDISC procedures. TYPE=UCOV data sets can be created by the CALIS and PRINCOMP procedures.

TYPE=WEIGHT Data Sets

The CALIS procedure creates and accepts as input a TYPE=WEIGHT data set. This data set contains the weight matrix used in generalized, weighted, or diagonally weighted least squares estimation. See Chapter 25, "The CALIS Procedure," for details.

Definitional Formulas

This section contrasts corrected and uncorrected SSCP, COV, and CORR matrices by showing how these matrices can be computed. In the following formulas, assume that the data consist of two variables, X and Y, with n observations.

$$\text{SSCP} = \begin{bmatrix} n & \sum X & \sum Y \\ \sum X & \sum X^2 & \sum XY \\ \sum Y & \sum XY & \sum Y^2 \end{bmatrix}$$

$$\text{CSSCP} = \begin{bmatrix} \sum(X-\bar{X})^2 & \sum(X-\bar{X})(Y-\bar{Y}) \\ \sum(X-\bar{X})(Y-\bar{Y}) & \sum(Y-\bar{Y})^2 \end{bmatrix}$$

$$\text{COV} = \frac{\text{CSSCP}}{n-1} = \frac{1}{n-1}\begin{bmatrix} \sum(X-\bar{X})^2 & \sum(X-\bar{X})(Y-\bar{Y}) \\ \sum(X-\bar{X})(Y-\bar{Y}) & \sum(Y-\bar{Y})^2 \end{bmatrix}$$

$$\text{UCOV} = \frac{1}{n}\begin{bmatrix} \sum X^2 & \sum XY \\ \sum XY & \sum Y^2 \end{bmatrix}$$

$$\text{CORR} = \begin{bmatrix} 1 & \dfrac{\sum(X-\bar{X})(Y-\bar{Y})}{\sqrt{\sum(X-\bar{X})^2\sum(Y-\bar{Y})^2}} \\ \dfrac{\sum(X-\bar{X})(Y-\bar{Y})}{\sqrt{\sum(X-\bar{X})^2\sum(Y-\bar{Y})^2}} & 1 \end{bmatrix}$$

$$\text{UCORR} = \begin{bmatrix} 1 & \dfrac{\sum XY}{\sqrt{\sum X^2\sum Y^2}} \\ \dfrac{\sum XY}{\sqrt{\sum X^2\sum Y^2}} & 1 \end{bmatrix}$$

Subject Index

p-value computation
    MODECLUS procedure, 4116
g2 inverse
    NLIN procedure, 4302
g4 inverse
    NLIN procedure, 4302
2D geometric anisotropic structure
    MIXED procedure, 3959

AB/BA crossover
    TTEST procedure, 7410
AB/BA crossover design
    TTEST procedure, 7442
AB/BA crossover designs
    power and sample size (POWER), 4995
absolute fit indices, 6761, 6762
absolute level of measurement, definition
    DISTANCE procedure, 1485
absorption of effects
    ANOVA procedure, 703
    GLM procedure, 2452, 2507
accelerated failure time model
    NLMIXED procedure, 4427
accelerated failure time models
    LIFEREG procedure, 2992
acceptance repeated confidence intervals
    SEQTEST procedure, 6023
ACECLUS procedure
    analyzing data in groups, 666, 678, 683
    between-cluster SSCP matrix, 660
    clustering methods, 676
    compared with other procedures, 663
    computational resources, 681
    controlling iterations, 662
    decomposition of the SSCP matrix, 660
    eigenvalues and eigenvectors, 670, 671, 677, 680, 682, 683
    initial estimates, 662, 675
    memory requirements, 681
    missing values, 679
    output data sets, 680
    output table names, 683
    time requirements, 681
    within-cluster SSCP matrix, 660
active constraints, 6761, 6762
active set methods
    NLMIXED procedure, 4395
actual power
    GLMPOWER procedure, 2661, 2663, 2669
    POWER procedure, 4825, 4922, 4924
actuarial estimates, *see* life-table estimates
adaptive algorithms
    adaptive rejection Metropolis sampling (ARMS), 155
    adaptive rejection sampling (ARS), 155
    Introduction to Bayesian Analysis, 155
    Markov chain Monte Carlo, 155
adaptive FDR adjustment
    MULTTEST procedure, 4189
adaptive Gaussian quadrature
    GLIMMIX procedure, 2103
    NLMIXED procedure, 4382
adaptive Hochberg adjustment
    MULTTEST procedure, 4188
adaptive Holm adjustment
    MULTTEST procedure, 4189
adaptive methods
    MULTTEST procedure, 4215
additive models
    TRANSREG procedure, 7160
ADF method
    TCALIS procedure, 6889, 6891
adjacent-category logits, *see also* response functions
    specifying in CATMOD procedure, 1131
    using (CATMOD), 1146
adjacent-level contrasts, 717
ADJRSQ
    SURVEYREG procedure, 6555
adjusted degrees of freedom
    MI procedure, 3795
    MIANALYZE procedure, 3856
adjusted means
    See least squares means, 2458
adjusted odds ratio, 1774
adjusted *p*-value
    MULTTEST procedure, 4182, 4211
adjusted R^2 selection (REG), 5525
adjusted R-square
    SURVEYREG procedure, 6571
adjusted residuals
    GENMOD procedure, 1984
adjusted treatment means
    LATTICE procedure, 2985
advantages and disadvantages of Bayesian analysis

Introduction to Bayesian Analysis, 149
affine step
QUANTREG procedure, 5388
agglomerative hierarchical clustering analysis, 1230
aggregates of residuals, 2047, 2054
AGK estimate
STDIZE procedure, 6231
agreement, measures of, 1765
Akaike information criterion (GENMOD), 1971
Akaike's information criterion
example (MIXED), 4017, 4031, 4060
GLIMMIX procedure, 2099
HPMIXED procedure, 2811, 2838
LOGISTIC procedure, 3329
MIXED procedure, 3907, 3976, 3997
PHREG procedure, 4610
SURVEYLOGISTIC procedure, 6420
Akaike's information criterion (finite sample corrected version)
GLIMMIX procedure, 2099
MIXED procedure, 3907, 3997
Akaike's information criterion (finite sample corrected version)
HPMIXED procedure, 2811, 2838
aliasing
GENMOD procedure, 1904
aliasing structure
GLM procedure, 2475
aliasing structure (GLM), 2610
allocation
SURVEYSELECT procedure, 6640, 6654
alpha, 5090
alpha factor analysis, 1534, 1552
alpha level, 415
ANOVA procedure, 711
contrast estimates (PHREG), 4561
FREQ procedure, 1693, 1701
Gelman-Rubin diagnostics (PHREG), 4548
GLIMMIX procedure, 2130, 2136, 2145, 2157, 2164, 2180, 2187
GLM procedure, 2460, 2469, 2475, 2479
GLMPOWER procedure, 2658
halfwidth tests (PHREG), 4548
hazard ratio estimates (LIFETEST), 4566
hazard ratio estimates (PHREG), 4562
HPMIXED procedure, 2819, 2821, 2822, 2830
LIFETEST procedure, 3113
MIXED procedure, 3905, 3921, 3925, 3929, 3950
NLIN procedure, 4289
NPAR1WAY procedure, 4457, 4464
PHREG procedure, 4536, 4544
posterior intervals (PHREG), 4555
POWER procedure, 4916
REG procedure, 5454
stationarity tests (PHREG), 4547
SURVEYFREQ procedure, 6315
SURVEYLOGISTIC procedure, 6383, 6393, 6398, 6404
SURVEYMEANS procedure, 6474
SURVEYREG procedure, 6543, 6557
TRANSREG procedure, 7160
TTEST procedure, 7401
ALR algorithm
GENMOD procedure, 2039
alternate forms, 5078
alternating least squares
MDS procedure, 3712
Alternating Logistic Regressions (ALR)
GENMOD procedure, 2039
alternative hypothesis, 415, 4916
alternative reference
SEQDESIGN procedure, 5873
analyses, available, 5049
analysis of covariance
MODEL statements (GLM), 2489
examples (GLM), 2571
power and sample size (GLMPOWER), 2672
analysis of covariation (NESTED), 4259
analysis of means
comparing LS-means (GLM), 2520
analysis of variance, *see also* ANOVA procedure, *see also* TTEST procedure
MODEL statements (GLM), 2489
categorical data, 1092
CATMOD procedure, 1094
corrected total sum of squares (Introduction to Modeling), 64
geometry (Introduction to Modeling), 63
Introduction to ANOVA Procedures, 115
mixed models (GLM), 2595
model (Introduction to Modeling), 34
multivariate (ANOVA), 706
multivariate (CANDISC), 1065
multivariate (GLM), 2464, 2531, 2582
nested design, 4253
one-way layout, example, 693
one-way tests (NPAR1WAY), 4469
one-way, variance-weighted, 714, 2473
power and sample size (GLMPOWER), 2643, 2666, 2672
power and sample size (POWER), 4859, 4863, 4957, 4981
quadratic response surfaces, 5736

repeated measures (CATMOD), 1150
repeated measures (GLM), 2532, 2590, 2598
sum of squares (Introduction to Modeling), 34
SURVEYREG procedure, 6570
three-way design (GLM), 2578
unbalanced (GLM), 2435, 2512, 2565
uncorrected total sum of squares (Introduction to Modeling), 64
within-subject factors, repeated measurements, 719
analysis statements
POWER procedure, 4826
ANALYSIS style
ODS Styles, 510
analyst's model
MI procedure, 3796
analyzing data in groups, 4374
ACECLUS procedure, 666, 678
CANCORR procedure, 1043
FACTOR procedure, 1564
FASTCLUS procedure, 1640
MDS procedure, 3721
MODECLUS procedure, 4095, 4108, 4113
PRINCOMP procedure, 5156
SCORE procedure, 5767
STDIZE procedure, 6225
TREE procedure, 7367
VARCLUS procedure, 7474
Andersen-Gill model
PHREG procedure, 4524, 4590, 4606
angle
classes (VARIOGRAM), 7536–7539, 7549, 7550
tolerance (VARIOGRAM), 7534, 7539, 7549, 7550
anisotropic
models (KRIGE2D), 2955–2957, 2959, 2962
nugget effect (KRIGE2D), 2962
anisotropic power covariance structure
GLIMMIX procedure, 2202
MIXED procedure, 3960
anisotropic spatial power structure
GLIMMIX procedure, 2202
MIXED procedure, 3960
anisotropy
factor (KRIGE2D), 2956
geometric (KRIGE2D), 2956
geometric (VARIOGRAM), 7547
major axis (VARIOGRAM), 7548, 7583
minor axis (VARIOGRAM), 7548, 7583
VARIOGRAM procedure, 7537, 7547, 7560, 7574
zonal (KRIGE2D), 2956
zonal (VARIOGRAM), 7547
annotate
global data set (REG), 5454
local data set (REG), 5493
annotating
CDF plots, 5272
ipp plots, 5282
lpred plots, 5290
pplot plots, 3028
predicted probability plots, 5304
ANOM adjustment
GLIMMIX procedure, 2144
anom plot
GLIMMIX procedure, 2288
ANOVA
codings (TRANSREG), 7230
SURVEYREG procedure, 6555, 6570
TRANSREG procedure, 7289
ANOVA (row mean scores) statistic, 1773
ANOVA procedure
absorption of effects, 703
alpha level, 711
balanced data, 692
Bartlett's test, 712
block diagonal matrices, 692
Brown and Forsythe's test, 712
canonical analysis, 707
characteristic roots and vectors, 706
compared to other procedures, 2434
complete block design, 697
computational methods, 725
confidence intervals, 711
contrasts, 717
dependent variable, 692
disk space, 702
effect specification, 720
factor name, 716
homogeneity of variance tests, 712
hypothesis tests, 719
independent variable, 692
interactive use, 723
interactivity and missing values, 724
introductory example, 692
level values, 716, 717
Levene's test for homogeneity of variance, 713
means, 709
memory requirements, 703, 725
missing values, 702, 724
model specification, 720
multiple comparison procedures, 710
multiple comparisons, 711–714
multivariate analysis of variance, 702, 706

O'Brien's test, 713
ODS graph names, 729
ODS table names, 727
orthonormalizing transformation matrix, 708
output data sets, 703, 724
pooling, automatic, 723
repeated measures, 715
sphericity tests, 718
SSCP matrix for multivariate tests, 706
transformations, 716, 717
transformations for MANOVA, 706
unbalanced data, caution, 692
Welch's ANOVA, 714
WHERE statement, 723
ANOVA table
GLMSELECT procedure, 2745
TRANSREG procedure, 7166, 7264
Ansari-Bradley scores
NPAR1WAY procedure, 4471
ANTE(1) structure
GLIMMIX procedure, 2193
MIXED procedure, 3959
ante-dependence structure
GLIMMIX procedure, 2193
MIXED procedure, 3959
apparent error rate, 1407
applicable tests
SEQTEST procedure, 6026
approximate
covariance, options (CALIS), 895
standard errors (CALIS), 854, 867, 928, 965
standard errors (TCALIS), 6729, 6744, 6894, 6937
approximate Bayesian bootstrap
MI procedure, 3776
approximate covariance estimation
clustering, 660
AR(1) structure
GLIMMIX procedure, 2194
MIXED procedure, 3959
arbitrary missing pattern
MI procedure, 3773
arcsine-square root transformation
confidence intervals (LIFETEST), 3114, 3137, 3139
arrays
MCMC procedure, 3508
monitor values of (MCMC), 3639
NLMIXED procedure, 4372
ASN plot
SEQDESIGN procedure, 5881
SEQTEST procedure, 6032
assessing MCMC convergence
autocorrelation, 169
effective sample sizes (ESS), 169
Gelman and Rubin diagnostics, 161
Geweke diagnostics, 163
Heidelberger and Welch diagnostics, 165
Introduction to Bayesian Analysis, 156
Markov chain Monte Carlo, 156
Raftery and Lewis diagnostics, 166
visual inspection, 156
association tests
LIFETEST procedure, 3101, 3108, 3168
association, measures of
FREQ procedure, 1736
asterisk (*) operator
TRANSREG procedure, 7139
asymmetric
data (MDS), 3720
asymmetric binary variable
DISTANCE procedure, 1485
asymmetric lambda, 1736, 1743
asymmetric two-sided design
SEQDESIGN procedure, 5878
asymptotic covariance
CALIS procedure, 852, 925
GLIMMIX procedure, 2095, 2099
MIXED procedure, 3905
TCALIS procedure, 6726, 6891
asymptotic variances
CALIS procedure, 926
asymptotically distribution free estimation
CALIS procedure, 859, 925
TCALIS procedure, 6734, 6891
at sign (@) operator
ANOVA procedure, 722
CATMOD procedure, 1143
GLM procedure, 2490
MIXED procedure, 3983, 4055
TRANSREG procedure, 7139
autocorrelation
Geary *c* coefficient (VARIOGRAM), 7518, 7534, 7562
Moran *I* coefficient (VARIOGRAM), 7518, 7534, 7562
REG procedure, 5581
VARIOGRAM procedure, 7518, 7560
autocorrelation function plot
MI procedure, 3790
autocorrelation weights
row-averaged (VARIOGRAM), 7562
standardized (VARIOGRAM), 7562
VARIOGRAM procedure, 7561
autocorrelations
Bayesian analysis (PHREG), 4639
automatic variables
GLIMMIX procedure, 2140, 2209

autoregressive moving-average structure
GLIMMIX procedure, 2194
MIXED procedure, 3959
autoregressive structure
example (MIXED), 4025
GLIMMIX procedure, 2194
MIXED procedure, 3959
average linkage
CLUSTER procedure, 1239, 1250
average relative increase in variance
MIANALYZE procedure, 3858
average sample number
SEQDESIGN procedure, 5873
average sample numbers plot
SEQDESIGN procedure, 5802
average variance of means
LATTICE procedure, 2984
axis customization
ODS Graphics, 637
azimuth
KRIGE2D procedure, 2957

B-spline
spline basis (Shared Concepts), 389
B-spline basis
GLIMMIX procedure, 389
GLMSELECT procedure, 389
TRANSREG procedure, 7142, 7264
backward elimination
GLMSELECT procedure, 2722
LOGISTIC procedure, 3301, 3329
PHREG procedure, 4570, 4617
REG procedure, 5435, 5524
badness of fit
MDS procedure, 3715, 3717, 3718, 3725, 3726
balanced data
ANOVA procedure, 692
example, complete block, 2556
balanced design, 4254
balanced repeated replication
Introduction to Survey Procedures, 267
SURVEYLOGISTIC procedure, 6429
SURVEYMEANS procedure, 6506
SURVEYREG procedure, 6565
variance estimation (SURVEYFREQ), 6334
variance estimation (SURVEYLOGISTIC), 6429
variance estimation (SURVEYMEANS), 6506
variance estimation (SURVEYREG), 6565
balanced square lattice
LATTICE procedure, 2980
banded Toeplitz structure
GLIMMIX procedure, 2203
MIXED procedure, 3959
bandwidth
optimal (DISCRIM), 1406
selection (KDE), 2903
VARIOGRAM procedure, 7536, 7539, 7552
bar (|) operator
ANOVA procedure, 721
CATMOD procedure, 1143
GENMOD procedure, 1974
GLM procedure, 2489
MIXED procedure, 3982, 3983, 4055
TRANSREG procedure, 7139
bar (|) operator
Shared Concepts, 369
bar (|) operator
POWER procedure, 4920
Bartlett's test
ANOVA procedure, 712
GLM procedure, 2471, 2526
Base SAS software, 23
Bayes estimation
NLMIXED procedure, 4382
Bayes' theorem
DISCRIM procedure, 1402
Introduction to Bayesian Analysis, 142
LOGISTIC procedure, 3298, 3342
MI procedure, 3780
Bayesian analysis
MIXED procedure, 3945
Bayesian analysis linear regression
GENMOD procedure, 1906
Bayesian confidence intervals
splines, 7075
Bayesian credible intervals
definition of, 149
equal-tail intervals, 149, 171
highest posterior density (HPD) intervals, 149, 171
Introduction to Bayesian Analysis, 149
Bayesian hypothesis testing
Introduction to Bayesian Analysis, 148
Bayesian inference
MI procedure, 3780
Bayesian information criterion
(GENMOD), 1971
Bayesian interval estimation
Introduction to Bayesian Analysis, 149
Bayesian models
Introduction to Modeling, 40
Bayesian probability
Introduction to Bayesian Analysis, 142
Behrens-Fisher problem
MCMC procedure, 3488

TTEST procedure, 7417
Bernoulli distribution
definition of (MCMC), 3531
GLIMMIX procedure, 2168
MCMC procedure, 3512, 3531
NLMIXED procedure, 4376
best subset selection
LOGISTIC procedure, 3292, 3301, 3329
PHREG procedure, 4570, 4618, 4652
beta distribution
definition of (MCMC), 3530
GLIMMIX procedure, 2168
MCMC procedure, 3512, 3530
between-cluster SSCP matrix
ACECLUS procedure, 660
between-imputation covariance matrix
MIANALYZE procedure, 3858
between-imputation variance
MI procedure, 3794
MIANALYZE procedure, 3856
between-subject factors
repeated measures, 2481, 2535
Bhapkar's test, 1208
bias
GLIMMIX procedure, 2232
bifactor model
TCALIS procedure, 7005, 7012
bimodality coefficient
CLUSTER procedure, 1246, 1258
bin-sort algorithm, 1634
binary data
Introduction to Regression, 89
binary distribution
definition of (MCMC), 3531
GLIMMIX procedure, 2168
MCMC procedure, 3512, 3531
NLMIXED procedure, 4376
Binary Lance and Williams nonmetric coefficient
DISTANCE procedure, 1513
binning
KDE procedure, 2899
binomial distribution
definition of (MCMC), 3531
GENMOD procedure, 1966
GLIMMIX procedure, 2168
MCMC procedure, 3512, 3531
NLMIXED procedure, 4376
binomial proportion confidence interval
power and sample size (POWER), 4949–4952
binomial proportion confidence interval precision
power and sample size (POWER), 4852
binomial proportion test, 1745
examples, 1809
power and sample size (POWER), 4844, 4850, 4934, 4935, 4937, 4986
bioequivalence, *see* equivalence tests, *see* equivalence tests
biological assay data, 5258, 5315
biplot
PRINQUAL procedure, 5238
biquartimax method, 1534, 1560, 1561, 6754, 6755
biquartimin method, 1534, 1561, 6755
bivariate density estimation
DISCRIM procedure, 1441
bivariate histogram
KDE procedure, 2906
biweight kernel (DISCRIM), 1404
block diagonal matrices
ANOVA procedure, 692
blocking
MCMC procedure, 3523
BLUE
MIXED procedure, 3977
BLUP
GLIMMIX procedure, 2192, 2209, 2210, 2278
MIXED procedure, 3977
Bonferroni *t* test, 711, 2470, 2517
Bonferroni adjustment
GLIMMIX procedure, 2144
GLM procedure, 2459
LIFETEST procedure, 3127
MIXED procedure, 3924
MULTTEST procedure, 4189, 4212
bootstrap
MI procedure, 3760
bootstrap adjustment
MULTTEST procedure, 4185, 4189, 4213, 4229
bootstrap FDR adjustment
MULTTEST procedure, 4190
boundary adjustment method
SEQTEST procedure, 6005
boundary adjustments
SEQTEST procedure, 6017, 6019, 6020
boundary constraints, 4373
GLIMMIX procedure, 2182, 2185
HPMIXED procedure, 2827, 2829
MIXED procedure, 3944, 3945, 4011
boundary for Whitehead one-sided design
SEQDESIGN procedure, 5848
boundary information
SEQDESIGN procedure, 5876
boundary key
SEQDESIGN procedure, 5853
SEQTEST procedure, 6006

boundary plot
SEQDESIGN procedure, 5802, 5881
boundary scale
SEQTEST procedure, 6006
boundary scales
SEQDESIGN procedure, 5832
boundary variables
SEQDESIGN procedure, 5834
bounds
NLMIXED procedure, 4373
Bowker's test of symmetry, 1765, 1766
box plot
ODS Graphics, 601, 645
reading group summary statistics, 754
saving summary statistics with outliers, 756
box plot, defined, 750
box plots
GLIMMIX procedure, 2282
NPAR1WAY procedure, 4450, 4459, 4493
reading group summary statistics, 795
saving group summary statistics, 791, 792
box plots, clipping boxes, 776, 777
examples, 807, 808
box plots, labeling
angles for, 782
points, 771
Box's epsilon, 2536
box-and-whiskers plots
schematic, 818
side-by-side, 750
skeletal, 817
statistics represented, 753, 790
styles of, 796
Box-Cox example
TRANSREG procedure, 7316
Box-Cox parameter
TRANSREG procedure, 7149
Box-Cox transformation
estimate $\lambda = 0$, 3588
MCMC procedure, 3583
Box-Cox transformations
TRANSREG procedure, 7181
BOXPLOT procedure
continuous group variables, 798
missing values, 798
ODS graph names, 811
percentile computation, 797
branch-and-bound algorithm
LOGISTIC procedure, 3329
PHREG procedure, 4618, 4652
Bray and Curtis coefficient
DISTANCE procedure, 1513
Breslow method
likelihood (PHREG), 4572, 4587
surival estimates (PHREG), 4545
survival estimates (PHREG), 4575
Breslow test, *see* Wilcoxon test for homogeneity
Breslow-Day test, 1776
Brewer's method
SURVEYSELECT procedure, 6652, 6667
Brown and Forsythe's test
ANOVA procedure, 712
GLM procedure, 2471, 2527
Brown-Mood test
NPAR1WAY procedure, 4470
Broyden-Fletcher-Goldfarb-Shanno update, 4371
BRR
SURVEYLOGISTIC procedure, 6429
SURVEYMEANS procedure, 6506, 6528
SURVEYREG procedure, 6565
BRR variance estimation
Introduction to Survey Procedures, 267
SURVEYFREQ procedure, 6334
SURVEYLOGISTIC procedure, 6429
SURVEYMEANS procedure, 6506
SURVEYREG procedure, 6565
Building the SSCP Matrix
GLMSELECT procedure, 2737
burn-in for MCMC
Introduction to Bayesian Analysis, 155
Markov chain Monte Carlo, 155
Burt table
CORRESP procedure, 1327

calibration data set
DISCRIM procedure, 1384, 1411
CALIS procedure
approximate covariance, 895
approximate standard errors, 854, 867, 928, 965
asymptotic covariance, 852, 925
asymptotic variances, 926
asymptotically distribution free estimation, 859, 925
chi-square indices, 932
chi-square, adjusted, 934
chi-square, displaying, 963, 964
chi-square, reweighted, 934
coefficient of determination, 863
compared to MIXED procedure, 3895
computation method, Hessian matrix, 855
computation method, Lagrange multipliers, 855
computation method, standard errors, 855
computational problems, 958, 960
computational problems, identification, 949, 959, 1005, 1022
constrained estimation, 845

constraints, 845, 871, 880, 886, 955, 956
constraints, programming statements, 845, 903, 905, 955
COSAN model, 832, 873
degrees of freedom, 854, 860, 861, 956
determination index, 936
displayed output options, 896
disturbances, prefix, 881
EQS program, 835
estimation criteria, 925
estimation methods, 829, 859, 923, 924, 926
exogenous variables, 943
factor analysis model, 834, 877, 879
factor analysis model, COSAN statement, 874
factor analysis model, LINEQS statement, 882
factor analysis model, RAM statement, 901
factor loadings, 920
FACTOR procedure, 847, 853, 877, 959
factor rotation, 879
factor scores, 845, 920, 922, 937, 966
fit function, 924
goodness of fit, 924, 928, 931, 963
gradient, 913, 945
Hessian matrix, 895, 896, 913, 927, 945, 946
initial values, 830, 865, 881, 884, 900, 941
input data set, 909
kappa, 939
kurtosis, 828, 853, 857, 938, 940
latent variable scores, 922, 937
latent variables, 828, 881, 908
likelihood ratio test, 932, 954
LINEQS model, 833, 881
LISREL model, 833
manifest variables, 828
matrix inversion, 927
matrix names, default, 879
matrix properties, COSAN model, 873, 874
MODEL procedure, 959
modification indices, 854, 860, 861, 928, 953, 954, 966
ODS graph names, 970
optimization, 830, 856, 857, 867–870, 896, 945, 947, 952
optimization history, 948
optimization, initial values, 941, 946
optimization, memory problems, 946
optimization, termination criteria, 886, 890, 892–895
output data sets, 913
output table names, 967
predicted model matrix, 922, 924, 943, 960, 963
prefix name, 882, 884, 900, 906
PRINCOMP procedure, 847
programming statements, 905
programming statements, differentiation, 854
RAM model, 832, 899
reciprocal causation, 863
REG procedure, 959
residuals, 929, 6901
residuals, prefix, 881
SCORE procedure, 862, 864, 922
significance level, 851
simplicity functions, 878
singularity criterion, 867
singularity criterion, covariance matrix, 852, 860, 870
skewness, 939
squared multiple correlation, 936, 965
step length, 856
structural equation, 832, 863, 907, 937
SYSLIN procedure, 959
SYSNLIN procedure, 959
t value, 928, 965
test indices, constraints, 860
variable selection, 942
Wald test, probability limit, 867
CANALS method
TRANSREG procedure, 7162
Canberra metric coefficient
DISTANCE procedure, 1510
CANCORR procedure
analyzing data in groups, 1043
canonical coefficients, 1030
canonical redundancy analysis, 1030, 1041
computational resources, 1048
correction for means, 1040
correlation, 1039
eigenvalues, 1045
eigenvalues and eigenvectors, 1033, 1049
examples, 1031, 1053
formulas, 1045
input data set, 1039
missing values, 1044
OUT= data sets, 1046
output data sets, 1040, 1046
output table names, 1051
OUTSTAT= data sets, 1040, 1046
partial correlation, 1040, 1041, 1043
principal components, relation to, 1045
regression coefficients, 1039
semipartial correlation, 1041
singularity checking, 1041

squared multiple correlation, 1041
squared partial correlation, 1041
squared semipartial correlation, 1041
statistical methods used, 1030
statistics computed, 1030
suppressing output, 1040
weighted product-moment correlation coefficients, 1044
candidates for addition or removal
GLMSELECT procedure, 2744
CANDISC procedure
computational details, 1074
computational resources, 1078
input data set, 1075
introductory example, 1063
Mahalanobis distance, 1084
MANOVA, 1065
memory requirements, 1078
missing values, 1074
multivariate analysis of variance, 1065
ODS table names, 1081
output data sets, 1070, 1071, 1076
time requirements, 1079
canonical analysis
ANOVA procedure, 707
GLM procedure, 2465
repeated measurements, 718
response surfaces, 5737
RSREG procedure, 5737
canonical coefficients, 1062
canonical component, 1062
canonical correlation
CANCORR procedure, 1029
definition, 1030
hypothesis tests, 1030
TRANSREG procedure, 7171, 7180
canonical discriminant analysis, 1061, 1384
canonical factor solution, 1539
canonical joint distribution
SEQDESIGN procedure, 5872
canonical redundancy analysis
CANCORR procedure, 1030, 1041
canonical variables, 1061
ANOVA procedure, 707
TRANSREG procedure, 7171
canonical weights, 1030, 1062
cascaded density estimates
MODECLUS procedure, 4112
case deletion diagnostics
GENMOD procedure, 1999
case weight
PHREG procedure, 4579
case-control studies
odds ratio, 1760, 1774
PHREG procedure, 4524, 4572, 4664
casewise deletion
PRINQUAL procedure, 5206
catalog
traditional graphics (LIFETEST), 3115
categorical data analysis, *see* CATMOD procedure
categorical variable, 183
SURVEYMEANS procedure, 6495
categorical variables, *see* classification variables
CATMOD procedure
analysis of variance, 1094
at sign (@) operator, 1143
AVERAGED models, 1158
bar (|) operator, 1143
cautions, 1147, 1163
cell count data, 1139
classification variables, 1142
compared to other procedures, 1094, 1146, 1147, 2495
computational method, 1167–1170
continuous variables, 1142
continuous variables, caution, 1147
contrast examples, 1195
contrasts, comparing with GLM, 1112
convergence criterion, 1121
design matrix, 1125, 1126
design matrix, REPEATED statement, 1159
effect specification, 1142
effective sample sizes, 1163
estimation methods, 1096
_F_ specification, 1118, 1140
hypothesis tests, 1165
input data sets, 1093, 1138
interactive use, 1097, 1107
introductory example, 1098
iterative proportional fitting, 1121
linear models, 1093
log-linear models, 1094, 1148, 1193, 1195, 1901
logistic analysis, 1095, 1146, 1208
logistic regression, 1094, 1146, 1188
maximum likelihood estimation, 1096
maximum likelihood estimation formulas, 1171
memory requirements, 1173
missing values, 1138
MODEL statement, examples, 1118
ordering of parameters, 1157
ordering of populations, 1141
ordering of responses, 1141
ordinal model, 1147
output data sets, 1132, 1144, 1145
parameterization, 1124

parameterization, comparing with GLM, 1112
positional requirements for statements, 1107
quasi-independence model, 1195
regression, 1095
repeated measures, 1094, 1129, 1150, 1202, 1206, 1208, 1212
repeated measures, MODEL statements, 1153
REPEATED statement, examples, 1151
response functions, 1114, 1118, 1130, 1132, 1134, 1135, 1137, 1140, 1178, 1184, 1219
RESPONSE keyword, 1114, 1117, 1118, 1120, 1129, 1142, 1148, 1150, 1158, 1160, 1165, 1174
RESPONSE= option, 1115, 1130
restrictions on parameters, 1137
sample survey analysis, 1095
sampling zeros and log-linear analyses, 1148
sensitivity, 1215
singular covariance matrix, 1163
specificity, 1215
time requirements, 1173
types of analysis, 1093, 1142
underlying model, 1096
weighted least squares, 1096, 1124
zeros, structural and sampling, 1164, 1195, 1200
Cauchy distribution
definition of (MCMC), 3531
MCMC procedure, 3512, 3531
CDF, *see* cumulative distribution function
CDF plots
annotating, 5272
axes, color, 5272
font, specifying, 5273
options summarized by function, 5269, 5288
reference lines, options, 5273–5276
threshold lines, options, 5275
cdfplot
PROBIT procedure, 5268
ceiling sample size
GLMPOWER procedure, 2663
POWER procedure, 4825, 4924
cell count data, 1724
CATMOD procedure, 1139
example (FREQ), 1799
cell of a contingency table, 184
cell-means coding
TRANSREG procedure, 7155, 7181, 7231
censored
data (LIFEREG), 2992
data, example (NLMIXED), 4427
LIFETEST procedure, 3151
observations (PHREG), 4666
survival times (PHREG), 4523, 4664, 4666
values (PHREG), 4528, 4644
censored values summary
PHREG procedure, 4633, 4637
censoring, 2992
LIFEREG procedure, 3020
MCMC procedure, 3544, 3664
variable (PHREG), 4528, 4564, 4575, 4666
censoring symbol
traditional graphics (LIFETEST), 3113
center-point coding
TRANSREG procedure, 7153, 7235, 7242
centering
GLIMMIX procedure, 2172
TRANSREG procedure, 7157
centering step
QUANTREG procedure, 5388
centroid component, 7463
definition, 7461
centroid method
CLUSTER procedure, 1239, 1251
chaining, reducing when clustering, 1247
character OPSCORE variables
PRINQUAL procedure, 5233
TRANSREG procedure, 7253
characteristic roots and vectors
ANOVA procedure, 706
GLM procedure, 2464
Chebychev distance coefficient
DISTANCE procedure, 1509
chi-square
adjusted (CALIS), 934
adjusted (TCALIS), 6904
displaying (CALIS), 963, 964
indices (CALIS), 932
reweighted (CALIS), 934
chi-square distribution
definition of (MCMC), 3532
MCMC procedure, 3512, 3532
chi-square mixture
GLIMMIX procedure, 2132
chi-square test
GLIMMIX procedure, 2126, 2157, 2164, 2167
HPMIXED procedure, 2817
MIXED procedure, 3919, 3929
chi-square tests
examples (FREQ), 1805, 1812, 1815
FREQ procedure, 1731, 1732
power and sample size (POWER), 4844, 4850, 4884, 4889, 4935, 4937, 4967, 4986

Rao-Scott (SURVEYFREQ), 6346
Wald (SURVEYFREQ), 6352
Wald log-linear (SURVEYFREQ), 6354
chi-squared coefficient
DISTANCE procedure, 1510, 1511
choice experiments
TRANSREG procedure, 7297
Cholesky
covariance structure (GLIMMIX), 2195
method (GLIMMIX), 2096
parameterization (NLMIXED), 4414
root (GLIMMIX), 2195, 2197
choosing optimization algorithm
Shared Concepts, 405
Chromy's method
SURVEYSELECT procedure, 6647, 6651
Cicchetti-Allison weights, 1768
Cityblock distance coefficient
DISTANCE procedure, 1509
class, *see* lag (VARIOGRAM), *see* lag distance (VARIOGRAM), *see* angle classes (VARIOGRAM)
class level
GLIMMIX procedure, 2108, 2274
HPMIXED procedure, 2813
MIXED procedure, 3909, 3995
class level coding
GLMSELECT procedure, 2744
class level information
GLMSELECT procedure, 2744
CLASS statement
Shared Concepts, 366
classification criterion
DISCRIM procedure, 1384
error rate estimation (DISCRIM), 1407
classification effect
Introduction to Modeling, 33
classification table
LOGISTIC procedure, 3298, 3340, 3341, 3419
classification variable
SURVEYMEANS procedure, 6484, 6495, 6500
classification variables, 183
ANOVA procedure, 692, 720
CATMOD procedure, 1142
GENMOD procedure, 1974
GLIMMIX procedure, 2122
GLM procedure, 2488
GLMPOWER procedure, 2652, 2653
HPMIXED procedure, 2815
MIXED procedure, 3916
Shared Concepts, 366
sort order of levels (GENMOD), 1922
SURVEYREG procedure, 6549
TRANSREG procedure, 7142, 7155
VARCOMP procedure, 7499
clinical trial
SEQDESIGN procedure, 5784, 5817
SEQTEST procedure, 5984
cluster
centers, 1623, 1637
definition (MODECLUS), 4117
deletion, 1635
elliptical, 659
final, 1623
initial, 1622, 1623
mean, 1637
median, 1634, 1637
midrange, 1637
minimum distance separating, 1623
plotting (MODECLUS), 4117
seeds, 1622
cluster analysis
disjoint, 1621
large data sets, 1621
robust, 1622, 1637
tree diagrams, 7355, 7356
cluster analysis (STDIZE)
standardizing, 6236
CLUSTER procedure, *see also* TREE procedure
algorithms, 1259
average linkage, 1230
centroid method, 1230
clustering methods, 1230, 1250
complete linkage, 1230
computational resources, 1260
density linkage, 1230, 1239
Euclidean distances, 1230
F statistics, 1246, 1258
FASTCLUS procedure, compared, 1230
flexible-beta method, 1230, 1240, 1241, 1255
hierarchical clusters, 1230
input data sets, 1241
interval scale, 1262
*k*th-nearest-neighbor method, 1230
maximum likelihood, 1230, 1239
McQuitty's similarity analysis , 1230
median method, 1230
memory requirements, 1260
missing values, 1261
non-Euclidean distances, 1230
ODS Graph names, 1269
output data sets, 1244, 1263
output table names, 1268
pseudo F and t statistics, 1246
ratio scale, 1262

single linkage, 1230
size, shape, and correlation, 1262
test statistics, 1241, 1246, 1247
ties, 1261
time requirements, 1260
TREE procedure, compared, 1230
two-stage density linkage, 1230
types of data sets, 1230
using macros for many analyses, 1290
Ward's minimum-variance method, 1230
Wong's hybrid method, 1230
cluster sampling
Introduction to Survey Procedures, 265
SURVEYSELECT procedure, 6645
clustering, *see also* cluster sampling, 1229, *see also* CLUSTER procedure, *see also* cluster sampling
approximate covariance estimation, 660
average linkage, 1239, 1250
centroid method, 1239, 1251
complete linkage method, 1239, 1251
density linkage methods, 1239, 1240, 1242, 1243, 1246, 1252, 1254, 1256
disjoint clusters of variables, 7461
Gower's method, 1240, 1255
hierarchical clusters of variables, 7461
maximum-likelihood method, 1244, 1254, 1255
McQuitty's similarity analysis, 1240, 1255
median method, 1240, 1255
methods affected by frequencies, 1248
outliers in, 1230, 1247
penalty coefficient, 1244
single linkage, 1240, 1255, 1256
smoothing parameters, 1253
standardizing variables, 1246
SURVEYFREQ procedure, 6310, 6321
SURVEYLOGISTIC procedure, 6391
SURVEYMEANS procedure, 6485
SURVEYREG procedure, 6550
transforming variables, 1230
two-stage density linkage, 1240
variables, 7461
Ward's method, 1240, 1257
weighted average linkage, 1240, 1255
clustering and computing distance matrix
Correlation coefficients, example, 1523
Jaccard coefficients, example, 1515
clustering and scaling
DISTANCE procedure, example, 1488
MODECLUS procedure, 4094, 4095, 4113
STDIZE procedure, example, 6236
clustering criterion
FASTCLUS procedure, 1622, 1636, 1637
clustering methods
ACECLUS procedure, 676
FASTCLUS procedure, 1622, 1624
MODECLUS procedure, 4095, 4113
CMF, *see* cumulative mean function
PHREG procedure, 4542
Cochran and Cox *t* approximation
TTEST procedure, 7402, 7418
Cochran's Q test, 1765, 1770
Cochran's Q test, 1824
Cochran-Armitage test for trend, 1762, 1818
continuity correction (MULTTEST), 4204
MULTTEST procedure, 4201, 4203, 4225
permutation distribution (MULTTEST), 4204
two-tailed test (MULTTEST), 4206
Cochran-Mantel-Haenszel statistics (FREQ), 1695, 1770, *see* chi-square tests
ANOVA (row mean scores) statistic, 1773
correlation statistic, 1772
examples, 1817
general association statistic, 1773
coefficient
alpha (FACTOR), 1580
of determination (CALIS), 863
of determination (TCALIS), 6740
of relationship (INBREED), 2869
coefficient of determination
definition (Introduction to Modeling), 64
definition (Introduction to Regression), 98, 110
coefficient of variation, 5101
SURVEYFREQ procedure, 6341
SURVEYMEANS procedure, 6500
coefficient prior
PHREG procedure, 4638
coefficients, redundancy
TRANSREG procedure, 7177
cohort studies, 1817
relative risk, 1761, 1775
collection effect
GLIMMIX procedure, 378
GLMSELECT procedure, 378
collinearity
REG procedure, 5555
collocation
VARIOGRAM procedure, 7548, 7564
combinations
generating with PLAN procedure, 4755
combined boundary plot
SEQDESIGN procedure, 5802, 5881
combining inferences
MI procedure, 3794
MIANALYZE procedure, 3855

common factor
defined for factor analysis, 1535
common factor analysis
common factor rotation, 1537
communality, 1536
compared with principal component analysis, 1535
Harris component analysis, 1536
image component analysis, 1536
interpreting, 1536
salience of loadings, 1537
uniqueness, 1536
comparing
dependent samples (Introduction to Nonparametric Analysis), 294, 296
distributions (Introduction to Nonparametric Analysis), 292
groups (GLM), 2511
independent samples (Introduction to Nonparametric Analysis), 293, 295
means (TTEST), 7392, 7431
variances (TTEST), 7418, 7431
comparing splines
GLIMMIX procedure, 2406
comparing trends
NLIN procedure, 4331
comparisonwise error rate (GLM), 2517
complementarity
QUANTREG procedure, 5387
complementary log-log model
SURVEYLOGISTIC procedure, 6425
complete block design
example (ANOVA), 697
example (GLM), 2556
complete linkage
CLUSTER procedure, 1239, 1251
complete separation
LOGISTIC procedure, 3327
SURVEYLOGISTIC procedure, 6419
completely randomized design
examples, 730
components
PLS procedure, 4766
compound symmetry structure
example (MIXED), 3970, 4026, 4031
GLIMMIX procedure, 2196
MIXED procedure, 3959
computational details
GRR method (VARCOMP), 7503
Hessian matrix (CALIS), 855
KDE procedure, 2897
Lagrange multipliers (CALIS), 855
LIFEREG procedure, 3037
maximum likelihood method (VARCOMP), 7503
MIVQUE0 method (VARCOMP), 7502
MIXED procedure, 4010
restricted maximum likelihood method (VARCOMP), 7503
SIM2D procedure, 6179
standard errors (CALIS), 855
SURVEYREG procedure, 6563
Type I method (VARCOMP), 7502
VARCOMP procedure, 7502, 7508
computational problems
CALIS procedure, 958
convergence (CALIS), 958
convergence (FASTCLUS), 1635
convergence (MIXED), 4011
convergence (TCALIS), 6932
identification (CALIS), 949, 959, 1005, 1022
identification (TCALIS), 6927, 6933
negative eigenvalues (CALIS), 960
negative eigenvalues (TCALIS), 6934
negative R square (CALIS), 960
negative R-square (TCALIS), 6934
NLMIXED procedure, 4399
overflow (CALIS), 958
overflow (TCALIS), 6932
singular predicted covariance model (TCALIS), 6933
singular predicted model (CALIS), 960
TCALIS procedure, 6932
time (CALIS), 960
time (TCALIS), 6934
computational resources
ACECLUS procedure, 681
CANCORR procedure, 1048
CLUSTER procedure, 1260
FACTOR procedure, 1578
FASTCLUS procedure, 1647
LIFEREG procedure, 3052
MODECLUS procedure, 4121
MULTTEST procedure, 4220
NLMIXED procedure, 4404
PRINCOMP procedure, 5161
QUANTREG procedure, 5399
ROBUSTREG procedure, 5690
SURVEYREG procedure, 6573
VARCLUS procedure, 7479
computed variables
GLIMMIX procedure, 2140
concordant observations, 1736
conditional and unconditional simulation
SIM2D procedure, 6156
conditional data

MDS procedure, 3713
conditional distributions of multivariate normal random variables
SIM2D procedure, 6178
conditional logistic regression
LOGISTIC procedure, 3314, 3355
PHREG procedure, 4524, 4666
conditional power
SEQTEST procedure, 6009, 6011, 6021, 6028, 6033
conditional power plot
SEQTEST procedure, 6033
conditional residuals
MIXED procedure, 3987
confidence bands
LIFETEST procedure, 3114, 3137
Confidence intervals
LIFEREG procedure, 3044
confidence intervals, 416, 5049
confidence coefficient (GENMOD), 1947
fitted values of the mean (GENMOD), 1951, 1982
individual observation (RSREG), 5732, 5733
means (ANOVA), 711
means (RSREG), 5732, 5733
means, power and sample size (POWER), 4852, 4859, 4871, 4879, 4891, 4900, 4956, 4965, 4974, 5009
model confidence interval (NLIN), 4307
pairwise differences (ANOVA), 711
parameter confidence interval (NLIN), 4306
profile likelihood (GENMOD), 1950, 1980
profile likelihood (LOGISTIC), 3299, 3333
TTEST procedure, 7402
Wald (GENMOD), 1954, 1981
Wald (LOGISTIC), 3302, 3335
Wald (SURVEYLOGISTIC), 6434
confidence intervals (LIFETEST)
log transformation, 3139
confidence intervals, FACTOR procedure, 1571
confidence level
SEQTEST procedure, 6007
SURVEYMEANS procedure, 6474
SURVEYREG procedure, 6543
VARIOGRAM procedure, 7534
confidence limits
adjusted (GLIMMIX), 2135, 2144, 2156, 2357
adjusted, simulated (GLIMMIX), 2144
and isotronic contrasts (GLIMMIX), 2357
and step-down (GLIMMIX), 2138, 2154, 2159
asymptotic (FREQ), 1737
covariance parameters (GLIMMIX), 2130
estimate, lower (GLIMMIX), 2139
estimate, upper (GLIMMIX), 2140
estimated likelihood (GLIMMIX), 2131
estimates (GLIMMIX), 2136
exact (FREQ), 1691
exponentiated (GLIMMIX), 2158
fixed effects (GLIMMIX), 2164
HPMIXED procedure, 2819, 2821, 2822, 2826
in mean plot (GLIMMIX), 2114, 2151
inversely linked (GLIMMIX), 2147, 2279
least squares mean estimate (GLIMMIX), 2156
least squares mean estimate, lower (GLIMMIX), 2159
least squares mean estimate, upper (GLIMMIX), 2161
least squares means (GLIMMIX), 2092, 2144, 2146
least squares means (HPMIXED), 2821
least squares means estimates (GLIMMIX), 2157
LIFETEST procedure, 3136, 3148, 3149
likelihood-based, details (GLIMMIX), 2240
LOGISTIC procedure, 3339
MIXED procedure, 3906
odds ratios (GLIMMIX), 2109, 2118, 2173, 2257
profile likelihood (GLIMMIX), 2131
random-effects solution (GLIMMIX), 2187
SEQTEST procedure, 6007, 6024
solution for random effects (HPMIXED), 2830
SURVEYFREQ procedure, 6339
SURVEYLOGISTIC procedure, 6438
SURVEYMEANS procedure, 6499, 6501
SURVEYREG procedure, 6555
TRANSREG procedure, 7160, 7171, 7172, 7174
truncation (GLIMMIX), 2279
VARCOMP procedure, 7505
VARIOGRAM procedure, 7536
vs. prediction limits (GLIMMIX), 2278
Wald (GLIMMIX), 2132
confidence limits, FACTOR procedure, 1571
configuration
MDS procedure, 3704
conjoint analysis
TRANSREG procedure, 7168, 7180, 7329, 7333
conjugate
descent (GLIMMIX), 402
descent (NLMIXED), 4372

gradient (GLIMMIX), 401
gradient (NLMIXED), 4371
gradient algorithm (CALIS), 856, 857, 867, 869, 945
gradient algorithm (TCALIS), 6731, 6736, 6737, 6923
conjugate gradient method
Shared Concepts, 408
connectedness method, *see* single linkage
Conover scores
NPAR1WAY procedure, 4472
constant transformations
avoiding (PRINQUAL), 5232
constant transformations, avoiding
TRANSREG procedure, 7253
constant variables
PRINQUAL procedure, 5233
TRANSREG procedure, 7164, 7253
constants specification
MCMC procedure, 3509
constrained estimation
CALIS procedure, 845
constraints
boundary (CALIS), 845, 871, 955
boundary (GLIMMIX), 2182, 2185
boundary (HPMIXED), 2827, 2829
boundary (MIXED), 3944, 3945
boundary (TCALIS), 6746, 6884
linear (CALIS), 845, 880, 956
linear (TCALIS), 6766, 6886
modification indices (CALIS), 860, 861
modification indices (TCALIS), 6734, 6735
nonlinear (CALIS), 845, 886
nonlinear (TCALIS), 6794
ordered (CALIS), 956
ordered (TCALIS), 6885
program statements (TCALIS), 6814
programming statements (CALIS), 845, 903, 905, 955
programming statements (TCALIS), 6882
test indices (CALIS), 860
test indices (TCALIS), 6734
constructed effects
GLIMMIX procedure, 2134, 2412
GLMSELECT procedure, 2706
containment method
GLIMMIX procedure, 2166
MIXED procedure, 3930, 3931
contingency coefficient, 1731, 1736
contingency tables, 184, 1676, 1697
CATMOD procedure, 1096
SURVEYFREQ procedure, 6312
continuity-adjusted chi-square, 1731, 1733
continuous variables, 720, 2488
GENMOD procedure, 1974
continuous-by-class effects
MIXED procedure, 3984
model parameterization (GLM), 2493
Shared Concepts, 372
specifying (GLM), 2488
continuous-nesting-class effects
MIXED procedure, 3984
model parameterization (GLM), 2493
Shared Concepts, 371
specifying (GLM), 2489
contrast, 5120
contrast specification
HPMIXED procedure, 2816
contrast-specification
GLIMMIX procedure, 2123, 2134
contrasts, 4374
comparing CATMOD and GLM, 1112
GENMOD procedure, 1943
GLIMMIX procedure, 2123
GLM procedure, 2454
HPMIXED procedure, 2816
MIXED procedure, 3917, 3920
power and sample size (GLMPOWER), 2647, 2652, 2665, 2666, 2672
power and sample size (POWER), 4859, 4860, 4863, 4957, 4981
repeated measurements (ANOVA), 717
repeated measures (GLM), 2483
specifying (CATMOD), 1110
SURVEYREG procedure, 6551, 6573
control
comparing treatments to (GLM), 2515, 2519
control charts, 25
control plot
GLIMMIX procedure, 2288
control sorting
SURVEYSELECT procedure, 6622, 6639, 6644, 6664
converge in EM algorithm
MI procedure, 3755
convergence
MCMC procedure, 3567
convergence criterion
ACECLUS procedure, 675
CATMOD procedure, 1121
GENMOD procedure, 1947, 1959
GLIMMIX procedure, 393, 395, 396, 403, 2095, 2105, 2111, 2269, 2276, 2303, 2349
MDS procedure, 3713, 3715, 3716
MIXED procedure, 2269, 3905, 3906, 3996, 4012
profile likelihood (LOGISTIC), 3299

convergence diagnostics, *see* assessing MCMC convergence
convergence in EM algorithm
MI procedure, 3760
convergence in MCMC
MI procedure, 3788, 3799
convergence problems
MIXED procedure, 4011
NLMIXED procedure, 4400
convergence status
GLIMMIX procedure, 2276
HPMIXED procedure, 2838
MIXED procedure, 3996
NLMIXED procedure, 4407
convolution
distribution (MULTTEST), 4205
KDE procedure, 2900
Cook's *D* influence statistic, 2478
RSREG procedure, 5732
Cook's D
MIXED procedure, 3991
Cook's D for covariance parameters
MIXED procedure, 3991
CORR procedure, 24
Introduction to Nonparametric Analysis, 296
correction for means
CANCORR procedure, 1040
correlated data
GEE (GENMOD), 1895, 1986
correlated proportions, *see* McNemar's test
correlation, 5049
CANCORR procedure, 1039
estimates (MIXED), 3951, 3953, 3958, 4027
length (VARIOGRAM), 7557
matrix (GENMOD), 1948, 1970
matrix (REG), 5454
matrix, estimated (CATMOD), 1120
principal components, 5160, 5162
radius (VARIOGRAM), 7557
range (KRIGE2D), 2932
range (VARIOGRAM), 7557
correlation coefficients
power and sample size (POWER), 4840, 4932, 4933
correlation dissimilarity coefficient
DISTANCE procedure, 1509
correlation matrix
Bayesian analysis (PHREG), 4639
PHREG procedure, 4639
correlation similarity coefficient
DISTANCE procedure, 1509
correlation statistic, 1772
correlations of least squares means
HPMIXED procedure, 2822
CORRESP procedure, 1320
adjusted inertias, 1355
algorithm, 1351
analyse des correspondances, 1320
appropriate scoring, 1320
Best variables, 1357
binary design matrix, 1336
Burt table, 1327, 1337
coding, 1343
COLUMN= option, use, 1352
computational resources, 1350
correspondence analysis, 1320
doubling, 1343
dual scaling, 1320
fuzzy coding, 1343, 1344
geometry of distance between points, 1354, 1364, 1369
homogeneity analysis, 1320
inertia, definition, 1320
input tables and variables, 1323, 1334
matrix decompositions, 1330, 1353
matrix formulas for statistics, 1356
memory requirements, 1350
missing values, 1327, 1343, 1346
multiple correspondence analysis (MCA), 1327, 1354, 1374
optimal scaling, 1320
optimal scoring, 1320
OUTC= data set, 1348
OUTF= data set, 1349
output data sets, 1348
output table names, 1361
partial contributions to inertia table, 1356
PROFILE= option, use, 1352
quantification method, 1320
reciprocal averaging, 1320
ROW= option, use, 1352
scalogram analysis, 1320
supplementary rows and columns, 1331, 1356
syntax, abbreviations, 1323
TABLES statement, use, 1323, 1332, 1334
time requirements, 1351
VAR statement, use, 1323, 1332, 1342
correspondence analysis
CORRESP procedure, 1320
COSAN model
CALIS procedure, 832, 873
specification, 840
structural model example (CALIS), 842
cosine coefficient
DISTANCE procedure, 1510
counting process
PHREG procedure, 4588

covariance
LATTICE procedure, 2985
matrix, definition (Introduction to Modeling), 58
of random variables (Introduction to Modeling), 57
parameter estimates (MIXED), 3906, 3907
parameter estimates, ratio (MIXED), 3915
parameters (GLIMMIX), 2084, 2085, 2089, 2095
parameters (MIXED), 3892
parameters, confidence interval (GLIMMIX), 2127
parameters, testing (GLIMMIX), 2127
principal components, 5160, 5162
SURVEYFREQ procedure, 6330
VARIOGRAM procedure, 7518, 7546, 7596
covariance coefficients, *see* INBREED procedure
covariance matrix
Bayesian analysis (PHREG), 4639
for parameter estimates (CATMOD), 1120
for response functions (CATMOD), 1120
GENMOD procedure, 1948, 1970
NLMIXED procedure, 4402, 4407
PHREG procedure, 4537, 4566, 4595
REG procedure, 5454
singular (CATMOD), 1163
symmetric and positive definite (SIM2D), 6177
covariance parameter estimates
GLIMMIX procedure, 2102, 2111, 2277
HPMIXED procedure, 2838
MIXED procedure, 3997
covariance similarity coefficient
DISTANCE procedure, 1509
covariance structure
anisotropic power (GLIMMIX), 2202
anisotropic power (MIXED), 3966
ante-dependence (GLIMMIX), 2193
ante-dependence (MIXED), 3962
autoregressive (GLIMMIX), 2084, 2194
autoregressive (MIXED), 3963
autoregressive moving-average (GLIMMIX), 2194
autoregressive moving-average (MIXED), 3963
banded (GLIMMIX), 2202
banded (MIXED), 3966
Cholesky type (GLIMMIX), 2195
compound symmetry (GLIMMIX), 2196
compound symmetry (MIXED), 3963
equi-correlation (GLIMMIX), 2196
equi-correlation (MIXED), 3963
examples (GLIMMIX), 2204
examples (MIXED), 3960
exponential (GLIMMIX), 2201
exponential anisotropic (MIXED), 3964
factor-analytic (GLIMMIX), 2197
factor-analytic (MIXED), 3963
G-side (GLIMMIX), 2083, 2127, 2193
gaussian (GLIMMIX), 2201
general (GLIMMIX), 2081
general linear (GLIMMIX), 2197
general linear (MIXED), 3964
heterogeneous autoregressive (GLIMMIX), 2194
heterogeneous autoregressive (MIXED), 3963
heterogeneous compound symmetry (GLIMMIX), 2197
heterogeneous compound symmetry (MIXED), 3963
heterogeneous Toeplitz (GLIMMIX), 2203
heterogeneous Toeplitz (MIXED), 3966
Huynh-Feldt (GLIMMIX), 2197
Huynh-Feldt (MIXED), 3964
Kronecker (MIXED), 3967
Matérn (GLIMMIX), 2201
Matérn (MIXED), 3965
misspecified (GLIMMIX), 2081
MIXED procedure, 3894, 3959
parameter reordering (GLIMMIX), 2182
penalized B-spline (GLIMMIX), 2189, 2191, 2198
positive (semi-)definite, 2195
power (GLIMMIX), 2202
power (MIXED), 3966
R-side (GLIMMIX), 2083, 2127, 2182, 2186, 2192, 2193
R-side with profiled scale (GLIMMIX), 2182
radial smooth (GLIMMIX), 2189, 2199
simple (GLIMMIX), 2201
simple (MIXED), 3964
spatial (GLIMMIX), 2188, 2201
spatial geometric anisotropic (MIXED), 3965
spherical (GLIMMIX), 2202
Toeplitz (GLIMMIX), 2202
Toeplitz (MIXED), 3966
unstructured (GLIMMIX), 2203
unstructured (MIXED), 3966
unstructured, correlation (GLIMMIX), 2203
unstructured, correlation (MIXED), 3966
variance components (GLIMMIX), 2204
variance components (MIXED), 3967
with second derivatives (GLIMMIX), 2167
working, independence (GLIMMIX), 2098

covariance structure analysis model, *see* COSAN model
covariance structures
examples (MIXED), 4019
covariances of least squares means
HPMIXED procedure, 2822
covariates, 5126
GLMPOWER procedure, 2653, 2658–2660, 2666, 2672
MIXED procedure, 3982
model parameterization (GLM), 2491
multiple correlation, 5126
proportional reduction in variation, 5126
covarimin method, 1534, 1561, 6755
covariogram
VARIOGRAM procedure, 7596
coverage displays
FACTOR procedure, 1572
CovRatio
MIXED procedure, 3992
CovRatio for covariance parameters
MIXED procedure, 3992
COVRATIO statistic, 5560
CovTrace
MIXED procedure, 3992
CovTrace for covariance parameters
MIXED procedure, 3992
Cox models
MCMC procedure, 3647
Cox regression analysis
PHREG procedure, 4530
semiparametric model (PHREG), 4523
Cramer's V statistic, 1731, 1736
Cramer-von Mises test
NPAR1WAY procedure, 4476
Crawford-Ferguson family, 1534
Crawford-Ferguson method, 1560, 1561, 6755, 6756
Crime Rates Data, example
PRINCOMP procedure, 5145
cross validated density estimates
MODECLUS procedure, 4111
cross validation
DISCRIM procedure, 1408
GLMSELECT procedure, 2741
PLS procedure, 4775, 4788, 4789
cross validation details
GLMSELECT procedure, 2747
crossed effects
design matrix (CATMOD), 1154
GENMOD procedure, 1974
GLIMMIX procedure, 2261
MIXED procedure, 3983
model parameterization (GLM), 2492
Shared Concepts, 370
specifying (ANOVA), 720, 721
specifying (CATMOD), 1142
specifying (GLM), 2488
crossover design
TTEST procedure, 7410, 7442
crossover designs
analyzing with GLIMMIX procedure, 4759
generating with PLAN procedure, 4759
power and sample size (POWER), 4995
crossproducts matrix
REG procedure, 5584
crosstabulation tables, 1676, 1697
SURVEYFREQ procedure, 6312, 6358
cubic clustering criterion, 1243, 1247
CLUSTER procedure, 1241
cumulative distribution function, 3043, 5258
LIFETEST procedure, 3100
cumulative logit model
SURVEYLOGISTIC procedure, 6424
cumulative logits, *see also* response functions
examples, (CATMOD), 1147
specifying in CATMOD procedure, 1131
using (CATMOD), 1146
cumulative martingale residuals
PHREG procedure, 4540, 4619, 4643
cumulative mean function, *see* mean function
cumulative residuals, 2047, 2054
custom scoring coefficients, example
SCORE procedure, 5780
customized odds ratio
SURVEYLOGISTIC procedure, 6409
customizing graphs, 5079
Czekanowski/Sorensen similarity coefficient
DISTANCE procedure, 1513

Data set option
TYPE=ACE, 7609
TYPE=BOXPLOT, 7609
TYPE=CHARTSUM, 7609
TYPE=CORR, 7609
TYPE=COV, 7613
TYPE=CSSCP, 7613
TYPE=DISTANCE, 7614
TYPE=EST, 7614
TYPE=LINEAR, 7616
TYPE=LOGISMOD, 7616
TYPE=MIXED, 7616
TYPE=QUAD, 7616
TYPE=RAM, 7617
TYPE=SSCP, 7617
TYPE=TREE, 7618
TYPE=UCORR, 7618
TYPE=UCOV, 7619

TYPE=WEIGHT, 7619
DATA step, 23
Davidon-Fletcher-Powell update, 402, 4372
decomposition of the SSCP matrix
ACECLUS procedure, 660
default estimation technique
GLIMMIX procedure, 2273
default output
GLIMMIX procedure, 2273
MIXED procedure, 3995
DEFAULT style
ODS Styles, 509
definition of
effective sample sizes (ESS), 169
definition of (MCMC)
posterior predictive distribution, 3560
degrees of freedom
between-within method (GLIMMIX), 2166
between-within method (MIXED), 3907, 3931
CALIS procedure, 854, 860, 861, 956
chi-square mixture (GLIMMIX), 2132
containment method (GLIMMIX), 2166
containment method (MIXED), 3930, 3931
FACTOR procedure, 1563
GLIMMIX procedure, 2125, 2126, 2135, 2136, 2146, 2157, 2165, 2166, 2241
HPMIXED procedure, 2817, 2819, 2822, 2823
infinite (GLIMMIX), 2136, 2146, 2157, 2167
infinite (HPMIXED), 2821, 2823
Kenward-Roger method (GLIMMIX), 2167, 3933
method (GLIMMIX), 2166
method (MIXED), 3930
MI procedure, 3794
MIANALYZE procedure, 3856, 3858
MIXED procedure, 3919, 3921, 3926, 3930
models with classification variables (GLM), 2495
NLMIXED procedure, 4359
residual method (GLIMMIX), 2167
residual method (HPMIXED), 2823
residual method (MIXED), 3931
Satterthwaite method (GLIMMIX), 2167
Satterthwaite method (MIXED), 3931
SURVEYFREQ procedure, 6340
SURVEYMEANS procedure, 6498
SURVEYREG procedure, 6568
TCALIS procedure, 6729, 6735
TRANSREG procedure, 7265
delete variables (REG), 5468
deleting observations
REG procedure, 5569
dendritic method, *see* single linkage
dendrogram, 7355
density estimation
DISCRIM procedure, 1425, 1441
MODECLUS procedure, 4109
density function, *see* probability density function
density linkage
CLUSTER procedure, 1239, 1240, 1242, 1243, 1246, 1252, 1254, 1256
dependent effect, definition, 720
dependent FDR adjustment
MULTTEST procedure, 4189
derivatives
NLIN procedure, 4289
derived parameters
SEQDESIGN procedure, 5851
description
traditional graphics (LIFETEST), 3115
descriptive statistics, *see also* UNIVARIATE procedure
LOGISTIC procedure, 3275
mixed model (HPMIXED), 2838
PHREG procedure, 4539
design effects
SURVEYFREQ procedure, 6341
SURVEYREG procedure, 6571
design information
SEQDESIGN procedure, 5877
SEQTEST procedure, 6028
design matrix
formulas (CATMOD), 1170
generation in CATMOD procedure, 1153
GENMOD procedure, 1975
GLMMOD procedure, 2621, 2629, 2631
TRANSREG procedure, 7173
design of experiments, *see* experimental design
design points, TPSPLINE procedure, 7060, 7079
design-adjusted chi-square tests
SURVEYFREQ procedure, 6346
destination, closing
ODS Graphics, 546
destinations
ODS, 434, 439, 461
ODS Graphics, 542
determination coefficients (TCALIS)
dependent variables, 6750
determination index
CALIS procedure, 936
TCALIS procedure, 6912
deviance
definition (GENMOD), 1899
GENMOD procedure, 1947
LOGISTIC procedure, 3292, 3300, 3343

PROBIT procedure, 5299, 5314
scaled (GENMOD), 1970
deviance information criterion, 2005, 3055
Introduction to Bayesian Analysis, 172
deviance information criterion (DIC)
definition of, 172
deviance residuals
GENMOD procedure, 1983, 1984
LOGISTIC procedure, 3350
PHREG procedure, 4574, 4611, 4685
deviations-from-means coding
TRANSREG procedure, 7153, 7181, 7235, 7242, 7293
DFBETA statistics
PHREG procedure, 4573, 4614
DFBETAS statistic (REG), 5561
DFBETAS statistics
LOGISTIC procedure, 3350
DFFITS
MIXED procedure, 3991
DFFITS statistic
GLM procedure, 2478
REG procedure, 5560
dgeneral distribution
MCMC procedure, 3513, 3541
diagnostic plots
GLIMMIX procedure, 2282
diagnostic statistics
REG procedure, 5557, 5558
diagnostics
GENMOD procedure, 1948, 1999
diagnostics panel
ODS Graphics, 633
diameter method, *see* complete linkage
DIC, *see* deviance information criterion, 2005, 3055
Dice coefficient
DISTANCE procedure, 1513
difference between means
confidence intervals, 711
difference test
TTEST procedure, 7407
diffogram
GLIMMIX procedure, 2288
dimension coefficients
MDS procedure, 3704, 3705, 3712, 3717, 3718, 3724, 3725
dimension information
GLIMMIX procedure, 2275
GLMSELECT procedure, 2744
MIXED procedure, 3995
dimensions
HPMIXED procedure, 2813
MIXED procedure, 3909
direct effect
TCALIS procedure, 6751
direct effects
design matrix (CATMOD), 1156
specifying (CATMOD), 1143
direct product structure
MIXED procedure, 3959
discordant observations, 1736
discrete logistic model
likelihood (PHREG), 4588
PHREG procedure, 4524, 4572, 4666
discrete variables, *see* classification variables, *see* classification variables
DISCRIM procedure
background, 1401
Bayes' theorem, 1402
bivariate density estimation, 1441
calibration data set, 1384, 1411
classification criterion, 1384
computational resources, 1418
cross validation, 1408
density estimate, 1402, 1405, 1406, 1425, 1441
error rate estimation, 1407, 1410
input data sets, 1412, 1414
introductory example, 1385
kernel density estimates, 1434, 1452
memory requirements, 1418
missing values, 1401
nonparametric methods, 1403
ODS table names, 1423
optimal bandwidth, selection, 1406
output data sets, 1414, 1415
parametric methods, 1402
%PLOTDEN macro, 1427
%PLOTPROB macro, 1427
posterior probability, 1403, 1405, 1425, 1441
posterior probability error rate, 1408, 1410
quasi inverse, 1408
resubstitution, 1407
squared distance, 1402
test set classification, 1407
time requirements, 1419
training data set, 1384
univariate density estimation, 1425
discriminant analysis, 1384
canonical, 1061, 1384
error rate estimation, 1385
misclassification probabilities, 1385
nonparametric methods, 1403
parametric methods, 1402
stepwise selection, 6267
discriminant function method

MI procedure, 3777
discriminant functions, 1062
disjoint clustering, 1621, 1622, 1624
dispersion parameter
estimation (GENMOD), 1898, 1971, 1978, 1979
GENMOD procedure, 1972
GLIMMIX procedure, 2215
LOGISTIC procedure, 3343
PROBIT procedure, 5314
weights (GENMOD), 1962
displayed output
GLMSELECT procedure, 2743
PHREG procedure, 4632
dissimilarity data
MDS procedure, 3704, 3713, 3720
distance, *see* lag (VARIOGRAM), *see* lag tolerance (VARIOGRAM)
between clusters (FASTCLUS), 1642
classification (VARIOGRAM), 7551
data (FASTCLUS), 1622
data (MDS), 3704
Euclidean (FASTCLUS), 1623
distance data
MDS procedure, 3713, 3720
DISTANCE data sets
CLUSTER procedure, 1241
distance measures available in DISTANCE procedure, *see* See proximity measures
DISTANCE procedure
absent-absent match, asymmetric binary variable, 1485
absent-absent match, example, 1516
absolute level of measurement, definition, 1485
affine transformation, 1484
asymmetric binary variable, 1485
available levels of measurement, 1500
available options for the option list, 1500
Binary Lance and Williams nonmetric coefficient, 1513
Bray and Curtis coefficient, 1513
Canberra metric coefficient, 1510
Chebychev distance coefficient, 1509
chi-squared coefficient, 1510, 1511
Cityblock distance coefficient, 1509
computing distances with weights, 1504
correlation dissimilarity coefficient, 1509
correlation similarity coefficient, 1509
cosine coefficient, 1510
covariance similarity coefficient, 1509
Czekanowski/Sorensen similarity coefficient, 1513
Dice coefficient, 1513
dot Product coefficient, 1510
Euclidean distance coefficient, 1508
examples, 1486, 1515, 1523
extension of binary variable, 1485
formatted values, 1514
formulas for proximity measures, 1507
frequencies, 1506
functional summary, 1492
fuzz factor, 1493
generalized Euclidean distance coefficient, 1509
Gower's dissimilarity coefficient, 1508
Gower's similarity coefficient, 1508
Hamann coefficient, 1512
Hamming distance coefficient, 1511
identity transformation, 1485
initial estimates for A-estimates, 1493
interval level of measurement, 1484
Jaccard dissimilarity coefficient, 1513
Jaccard similarity coefficient, 1513
Kulcynski 1 coefficient, 1514
Lance-Williams nonmetric coefficient, 1510
levels of measurement, 1484
linear transformation, 1484
log-interval level of measurement, 1484
many-to-one transformation, 1484
Minkowski L_p distance coefficient, 1509
missing values, 1497, 1499, 1503, 1514
monotone increasing transformation, 1484
nominal level of measurement, 1484
nominal variable, 1485
normalization, 1498, 1499
one-to-one transformation, 1484
ordinal level of measurement, 1484
output data sets, 1498, 1515
Overlap dissimilarity coefficient, 1510
Overlap similarity coefficient, 1510
phi-squared coefficient, 1511
power distance coefficient, 1509
power transformation, 1484
ratio level of measurement, 1484
Roger and Tanimoto coefficient, 1512
Russell and Rao similarity coefficient, 1513
scaling variables, 1485
shape distance coefficient, 1508
similarity Ratio coefficient, 1509
similarity ratio coefficient, 1510
simple Matching coefficient, 1511
simple matching dissimilarity coefficient, 1511
size distance coefficient, 1508
Sokal and Sneath 1 coefficient, 1512
Sokal and Sneath 3 coefficient, 1512

squared correlation dissimilarity coefficient, 1509
squared correlation similarity coefficient, 1509
squared Euclidean distance coefficient, 1508
squared simple matching dissimilarity coefficient, 1511
standardization methods, 1501
standardization suppression, 1498
standardization with frequencies, 1506
standardization with weights, 1506
standardization, default methods, 1486, 1502
standardization, example, 1488
standardization, mandatory, 1486
strictly increasing transformation, 1484
summary of options, 1492
symmetric binary variable, 1485
transforming ordinal variables to interval, 1485, 1498
weights, 1504, 1506
distribution tests, 5049
distributions
Gompertz, 5258
logistic, 5258
normal, 5258
dlogden distribution
MCMC procedure, 3513
DOCUMENT destination
ODS Graphics, 606
document path
ODS Graphics, 608
DOCUMENT procedure
document path, 608
ODS Graphics, 606
Documents window
ODS Graphics, 607
dollar-unit sampling
SURVEYSELECT procedure, 6670
domain analysis
SURVEYFREQ procedure, 6323, 6366
SURVEYLOGISTIC procedure, 6394, 6432
SURVEYMEANS procedure, 6493
SURVEYREG procedure, 6573, 6603
domain statistics
SURVEYMEANS procedure, 6502
donor stratum
SURVEYLOGISTIC procedure, 6431
SURVEYMEANS procedure, 6508
SURVEYREG procedure, 6567
DOT product (SCORE), 5759
dot Product coefficient
DISTANCE procedure, 1510
double arcsine test
MULTTEST procedure, 4206
double dogleg
algorithm (TCALIS), 6730, 6736, 6738, 6923
method (GLIMMIX), 401
method (NLMIXED), 4371
double exponential distribution
definition of (MCMC), 3536
MCMC procedure, 3514, 3536
double-dogleg
algorithm (CALIS), 856, 868, 869, 945
double-dogleg method
Shared Concepts, 408
doubly iterative algorithm
GLIMMIX procedure, 2271
drift parameter
SEQDESIGN procedure, 5822, 5825, 5826, 5852, 5873
dual scaling
CORRESP procedure, 1320
dummy variable creation
TRANSREG procedure, 7126, 7141, 7142, 7155, 7173, 7174, 7230–7233, 7235, 7237, 7239, 7240, 7242, 7244–7248, 7293, 7296–7298
Duncan's multiple range test, 711, 2470, 2522
Duncan-Waller test, 714, 2473, 2523
error seriousness ratio, 713, 2472
multiple comparison (ANOVA), 733
Dunnett's adjustment
GLIMMIX procedure, 2144
GLM procedure, 2459
LIFETEST procedure, 3127
MIXED procedure, 3924
Dunnett's test, 2470, 2519, 2520
one-tailed lower, 712, 2470
one-tailed upper, 712
two-tailed, 711
DWLS method
TCALIS procedure, 6892

EBE
GLIMMIX procedure, 2193, 2278
EBLUP
GLIMMIX procedure, 2192
MIXED procedure, 3940
EDF plots
NPAR1WAY procedure, 4459, 4498
EDF tests
NPAR1WAY procedure, 4474
effect
definition, 720, 1142, 2488
name length (GLIMMIX), 2108
name length (MIXED), 3909

specification (ANOVA), 720
specification (CATMOD), 1142
specification (GENMOD), 1974
specification (GLM), 2488
effect coding
TRANSREG procedure, 7127, 7153, 7235, 7242, 7293
EFFECT parameterization
SURVEYLOGISTIC procedure, 6412
effect parameterization
Shared Concepts, 373
effect size, 418, 419
power and sample size (POWER), 4882
effect sizes
GLM procedure, 2502
MODEL statement (GLM), 2475
EFFECT statement
collection effect (Shared Concepts), 378
multimember effect (Shared Concepts), 378
polynomial effect (Shared Concepts), 380
spline effect (Shared Concepts), 384
Syntax (Shared Concepts), 377
effect testing
SURVEYREG procedure, 6570
effective number of parameters, 2005, 3055
effective sample size
LIFETEST procedure, 3134
effective sample sizes
Bayesian analysis (PHREG) procedure, 4640
effective sample sizes (ESS)
definition of, 169
Introduction to Bayesian Analysis, 169
effects
TCALIS procedure, 6751
effects (TCALIS), 6751
Efron method
likelihood (PHREG), 4572, 4588
eigenvalues and eigenvectors
ACECLUS procedure, 670, 671, 677, 680, 682, 683
CANCORR procedure, 1033, 1049
PRINCOMP procedure, 5144, 5160, 5162
RSREG procedure, 5742
Einot and Gabriel's multiple range test
ANOVA procedure, 713
examples (GLM), 2560
GLM procedure, 2472, 2523
Ekblom-Newton algorithm
FASTCLUS procedure, 1637
elementary linkage analysis, *see* single linkage
EM algorithm
MI procedure, 3769, 3799
EM-REML
HPMIXED procedure, 2834
empirical Bayes estimate
NLMIXED procedure, 4344, 4352, 4359, 4377, 4379
empirical Bayes estimates
GLIMMIX procedure, 2104, 2193, 2278
empirical Bayes estimation
GLIMMIX procedure, 2104
NLMIXED procedure, 4382
empirical best linear unbiased prediction
MIXED procedure, 3940
empirical distribution function
plots (NPAR1WAY), 4459, 4498
tests (Introduction to Nonparametric Analysis), 292
tests (NPAR1WAY), 4474
empirical estimator
GLIMMIX procedure, 2096, 2243, 2246
MIXED procedure, 3907
empirical power, *see* simulation
Epanechnikov kernel (DISCRIM), 1404
EQS program
CALIS procedure, 835
TCALIS procedure, 6704
equal-precision bands
LIFETEST procedure, 3114, 3139, 3175
equal-tail intervals
credible intervals (PHREG), 4556, 4639
definition of, 149
Introduction to Bayesian Analysis, 149, 171
equality
of means (TTEST), 7392, 7431
of variances (TTEST), 7418, 7431
equamax method, 1534, 1560, 1561, 6754, 6755
equivalence test
TTEST procedure, 7408, 7452
equivalence tests, 416, 5049
power and sample size (POWER), 4851, 4852, 4858, 4871, 4879, 4891, 4900, 4955, 4964, 4973, 4974, 4995
ergodicity
VARIOGRAM procedure, 7547
error rate estimation
DISCRIM procedure, 1407, 1410
discriminant analysis, 1385
error seriousness ratio
Waller-Duncan test, 713, 2472
error spending
SEQTEST procedure, 6011
error spending function method
SEQDESIGN procedure, 5876
error spending information
SEQDESIGN procedure, 5877
SEQTEST procedure, 6028

error spending method
SEQDESIGN procedure, 5791, 5829, 5848, 5849, 5876
error spending plot
SEQDESIGN procedure, 5803, 5882
SEQTEST procedure, 6033
error sum of squares clustering method, *see* Ward's method
estimability
definition (Introduction to Modeling), 66
definition (Introduction to Regression), 102
GLIMMIX procedure, 2124
GLM procedure, 2455
HPMIXED procedure, 2816, 2818, 2820, 2823
MIXED procedure, 3917
estimability checking
GENMOD procedure, 1942
LOGISTIC procedure, 3283
PHREG procedure, 4561
SURVEYLOGISTIC procedure, 6394
estimable function
definition (Introduction to Modeling), 66
definition (Introduction to Regression), 102
Introduction to ANOVA Procedures, 116, 120
estimable functions
checking (GLM), 2455
displaying (GLM), 2475
example (GLM), 2496
general form of, 2497
GLM procedure, 2456, 2457, 2463, 2477, 2496–2499, 2501, 2511
MIXED procedure, 3939
printing (GLM), 2475
SURVEYREG procedure, 6553
estimated population marginal means, *see* least squares means
estimates
GLIMMIX procedure, 2134
HPMIXED procedure, 2818
multiple comparison adjustment (GLIMMIX), 2135
estimating equations
Introduction to Modeling, 31
estimation
dispersion parameter (GENMOD), 1898
KRIGE2D procedure, 2931
maximum likelihood (GENMOD), 1968
mixed model (MIXED), 3974
regression parameters (GENMOD), 1898
VARIOGRAM procedure, 7519
estimation criteria
CALIS procedure, 925
TCALIS procedure, 6892
estimation method
baseline estimation (PHREG), 4545
estimation methods
CALIS procedure, 829, 859, 923, 924, 926
GLIMMIX procedure, 2102
HPMIXED procedure, 2812
MIXED procedure, 3908
TCALIS procedure, 6888, 6892, 6893
VARCOMP procedure, 7498
Euclidean distance coefficient
DISTANCE procedure, 1508
Euclidean distances, 1242, 1244, 1403, 1623
clustering, 1230
MDS procedure, 3705, 3712, 3724
Euclidean length
STDIZE procedure, 6231
event symbol
traditional graphics (LIFETEST), 3115
event times
PHREG procedure, 4523, 4528
event values summary
PHREG procedure, 4633, 4637
events/trials format for response
GENMOD procedure, 1946, 1966
exact logistic regression
LOGISTIC procedure, 3285, 3359
exact method
likelihood (PHREG), 4572, 4587
exact tests
computational algorithms (FREQ), 1780
computational algorithms (NPAR1WAY), 4477
computational resources (FREQ), 1781
computational resources (NPAR1WAY), 4478
confidence limits, 1691
examples (NPAR1WAY), 4499, 4502
FREQ procedure, 1779, 1819
MONTE Carlo estimation (NPAR1WAY), 4479
network algorithm (FREQ), 1780
NPAR1WAY procedure, 4476
p-value definitions (NPAR1WAY), 4478
p-value, definitions, 1781
permutation test (MULTTEST), 4204
examples, GLIMMIX
k-d tree information, 2252
LINP, 2209–2211
LOGL, 2402
_MU_, 2210, 2211
VARIANCE, 2211, 2335
adding computed variables to output data set, 2141

analysis of means, ANOM, 2297
analysis of summary data, 2418
anom plot, 2149
anom plots, 2297
binary data, 2339
binary data, GLMM, 2314
binary data, pseudo-likelihood, 2314
binary data, sort order, 2110
binomial data, 2087
binomial data, GLM, 2301
binomial data, GLMM, 2306
binomial data, overdispersed, 2330
binomial data, spatial covariance, 2310
bivariate data; Poisson, binary, 2341
blotch incidence data, 2330
box plots, 2286
bucket size in *k-d* tree, 2252
central *t* distribution, 2170
Cholesky covariance structure, 2359
collection effect, 2263
computed variables, 2322, 2347
constructed random effect, 2413
containment hierarchy, 2104, 2249
contrast, among covariance parameters, 2129, 2130, 2368
contrast, differences of splines, 2267
contrast, nonpositional syntax, 2124, 2264, 2409
contrast, positional syntax, 2123, 2264
contrast, with groups, 2126
contrast, with spline effects, 2266
control plot, 2150, 2295
covariance structure, 2204
covariates in LS-mean construction, 2145
COVTEST statement, 2133
COVTEST with keywords, 2129
COVTEST with no restrictions, 2237
COVTEST with specified values, 2129
cow weight data, 2345
diallel experiment, 2413
diffogram, 2150, 2292, 2293, 2318
diffplot, 2150
empirical Bayes estimates, 2379
ENSO data, 2372
epileptic seizure data, 2385
equivalent models, TYPE=VC, 2247
equivalent models, with and without subject, 2247
estimate, multi-row, 2137
estimate, with groups, 2137
estimate, with varied divisors, 2137
ferrite cores data, 2357
FIRSTORDER option for Kenward-Roger method, 2362
Fisher's iris data, 2364
foot shape data, 2378
FREQ statement, 2379
G-side spatial covariance, 2188
GEE-type model, 2098, 2234, 2388
generalized logit, 2125, 2136
generalized logit with random effects, 2268
generalized Poisson distribution, 2402
getting started, 2086
GLM mode, 2234
GLMM mode, 2234
graphics, anom plots, 2297
graphics, box plots, 2286
graphics, control plot, 2295
graphics, custom template, 2326
graphics, diffogram, 2292, 2293, 2318
graphics, mean plots, 2288
graphics, Pearson residual panel, 2334
graphics, predicted profiles, 2351
graphics, residual panel, 2282, 2283
graphics, studentized residual panel, 2325
group option in contrast, 2126
group option in estimate, 2137
group-specific smoothing, 2352
grouped analysis, 2379
groups in RANDOM statement, 2128, 2365
herniorrhaphy data, 2338
Hessian fly data, 2300
holding covariance parameters fixed, 2181
homogeneity of covariance parameters, 2128, 2129, 2365
identity model, 2418
infinite degrees of freedom, 2386
inverse linking, 2092, 2159
isotonic contrast, 2357
joint model (DIST=BYOBS), 2169
joint model, independent, 2341
joint model, marginal correlation, 2343
joint model, shared random effect, 2342
Kenward-Roger method, 2359
knot construction, *k-d* tree, 2190, 2252, 2347
knot construction, equal, 2190
knot construction, optimization, 2191
Laplace approximation, 2226, 2227, 2231, 2395
LDATA= option, 2198
least squares mean estimate, 2155, 2357
least squares mean estimate, multi-row, 2158
least squares mean estimate, with varied divisors, 2158
least squares means, 2092
least squares means, AT option, 2145
least squares means, covariate, 2145

least squares means, differences against control, 2147
least squares means, slice, 2152
least squares means, slice differences, 2152
linear combination of LS-means, 2155
linear covariance structure, 2198, 2418
logistic model with random effects, binomial data, 2306
logistic model, binomial data, 2301
logistic regression with random intercepts, 2086
logistic regression, binary data, 2257, 2339
logistic regression, binomial data, 2271
marginal variance matrix, 2206
mean plot, sliced interaction, 2151
mean plot, three-way, 2151
mean plots, 2288
MIVQUE0 estimates, 2184
multicenter clinical trial, 2390
multimember effect, 379, 2263, 2413
multinomial data, 2125, 2136, 2268, 2379, 2395
multiple local minima, 2376
multiple plot requests, 2116
multiplicity adjustment, 2357, 2392, 2395, 2411
multivariate distributions, 2169
multivariate normal model, 2365
nesting v. crossing, 2249
NLIN procedure, 2416
NLOPTIONS statement, 2347
NOFIT option, 2252
NOITER option for covariance parameters, 2183
nonlinear regression, 2416
NOPROFILE option, 2373
odds ratio, 2173, 2258
odds ratio, all pairwise differences, 2174, 2259
odds ratio, with interactions, 2173, 2259
odds ratio, with reference value, 2173, 2259
odds ratio, with specified units, 2174, 2259
ordinal data, 2379, 2395
OUTDESIGN option, 2407, 2413
output statistics, 2177, 2209, 2322, 2347, 2407
overdispersion, 2084, 2098, 2186, 2234, 2330
parallel shifted smooths, 2354
Pearson residual panel, 2334
penalized B-spline, 2199
Poisson model with offset, 2322, 2386
Poisson model with random effects, 2400
Poisson regression, 2340
Pothoff-Roy repeated measures data, 2358
proportional odds model with random effect, 2379, 2395
quadrature approximation, 2229, 2379, 2400
quasi-likelihood, 2335
R-side covariance structure, 2359, 2388
R-side covariance, binomial data, 2310
radial smooth, with parallel shifts, 2354
radial smoothing, 2252, 2347, 2373
radial smoothing, group-specific, 2352
remote monitoring, 403
REPEATED in MIXED vs RANDOM in GLIMMIX, 2269
residual panel, 2282, 2283
row-wise adjustment of LS-mean differences, 2143
salamander data, 2313
Satterthwaite method, 2143
saturated model, 2418
Scottish lipcancer data, 2321
SGPANEL procedure, 2351
SGPLOT procedure, 2375, 2376, 2383, 2406, 2408
SGRENDER procedure, 2326
simple differences, 2152, 2318
simple differences with control, 2153
simulated *p*-values, 2392, 2395, 2411
slice *F* test, 2152
slice differences, 2152, 2318
slice differences with control, 2153
space-filling design, 2191
spatial covariance, binomial data, 2310
specifying lower bounds, 2182
specifying values for degrees of freedom, 2165
spline differences, 2411
spline effect, 377, 2263, 2407
splines in interactions, 2411
standardized mortlity rate, 2322
starting values, 2376
starting values and BY groups, 2185
starting values from data set, 2185
step-down *p*-values, 2392, 2395, 2411
studentized maximum modulus, 2143
studentized residual panel, 2325
subject processing, 2104, 2247
subject processing, containment, 2249
subject processing, crossed effects, 2249
subject processing, nested effects, 2249
subject-processing, asymptotics, 2226, 2227
syntax, differences to MIXED, 2269, 2270
test for independence, 2382
test for Poisson distribution, 2402

testing covariance parameters, 2128, 2365, 2368, 2373
theophylline data, 2415
TYPE=CS and TYPE=VC equivalence, 2196
user-defined log-likelihood function, 2402
user-defined variance function, 2231
user-specified link function, 2210, 2211
user-specified variance function, 2211, 2335
working independence, 2098, 2234
examples, GLMSELECT
multimember effect, 379
examples, HPMIXED
animal breeding data, 2808
getting started, 2808
many fixed and random effects, 2808
NOITER option for covariance parameters, 2828
starting values and BY groups, 2829
starting values from data set, 2828
examples, MCMC
array subscripts, 3509
arrays, 3508
arrays, store data set variables, 3599
BEGINCNST/ENDCNST statements, 3599
Behrens-Fisher problem, 3488
blocking, 3523
Box-Cox transformation, 3583
censoring, 3545, 3664
change point models, 3630
cloglog transformation, 3551
constrained analysis, 3666
Cox models, 3647
Cox models, time dependent covariates, 3657
Cox models, time independent covariates, 3649
deviance information criterion, 3645
discrete priors, 3588
error finding using the PUT statement, 3568
estimate functionals, 3596, 3639
estimate posterior probabilities, 3491
exponential models, survival analysis, 3635
FCMP procedure, 3521, 3675, 3678
Gelman-Rubin diagnostics, 3693
generalized linear models, 3592
GENMOD procedure, BAYES statement, 3601, 3604
getting started, 3479
GLIMMIX procedure, 3618
graphics, box plots, 3643
graphics, custom template, 3558, 3563, 3622, 3627, 3655, 3662, 3681
graphics, fit plots, 3634
graphics, kernel density comparisons, 3581, 3583, 3623, 3630
graphics, multiple chains, 3697
graphics, posterior predictive checks, 3565
graphics, PSRF plots, 3700
graphics, scatter plots, 3631, 3688, 3692, 3693
graphics, survival curves, 3644
hierarchical centering, 3615
IF-ELSE statement, 3489
implement a conjugate Gibbs sampler, 3521
implement a new sampling algorithm, 3672
improve mixing, 3605, 3683
improving mixing, 3615
initial values, 3529
interval censoring, 3664
Jeffreys' prior, 3598
JOINTMODEL option, 3556, 3652, 3660
LAG functions, 3650
linear regression, 3480
log transformation, 3551
logistic regression, diffuse prior, 3592
logistic regression, Jeffreys' prior, 3598
logistic regression, random-effects, 3614
logistic regression, sampling via Gibbs, 3672
logit transformation, 3551
matrix functions, 3599, 3670, 3679
MISSING= option, 3659
mixed-effects models, 3492, 3614
mixing, 3605, 3683
mixture of normal densities, 3581
model comparison, 3645
modelling dependent data, 3556
MONITOR= option, arrays, 3639
multivariate priors, 3670
NLMIXED procedure, 3626
nonlinear Poisson regression, 3605
PHREG procedure, BAYES statement, 3654, 3661
Poisson regression, 3602
Poisson regression, nonlinear, 3605, 3623
Poisson regression, random-effects, 3623
posterior predictive distribution, 3561
probit transformation, 3551
proportional hazard models, 3647
random-effects models, 3614
regenerate diagnostics plots, 3557
SGPLOT procedure, 3580, 3582, 3631, 3632, 3642, 3644, 3687, 3691, 3696, 3698
SGRENDER procedure, 3559, 3564, 3623, 3627, 3655, 3662, 3681
specifying a new distribution, 3541

store data set variables in arrays, 3599
survival analysis, 3634
survival analysis, exponential models, 3635
survival analysis, Weibull model, 3639
TEMPLATE procedure, 3558, 3563, 3622, 3627, 3655, 3662, 3681
truncated distributions, 3545, 3670
UDS statement, 3520, 3672
use macros to construct loglikelihood, 3657
user-defined samplers, 3520, 3672
Weibull model, survival analysis, 3639
examples, MIXED
ASYCOV matrix, 4023
asymptotic covariance of covariance parameters, 4023
autoregressive structure, R-side, 4025
box plots, 4076
box plots, paneling, 3914
broad inference space, 3918, 3920
compound symmetry, G-side setup, 3971, 4029
compound symmetry, R-side setup, 3971, 4026
constrained anisotropic model, 3964
covariates in LS-mean construction, 3925
COVTEST option, 4020, 4032
deletion estimates, 4061
doubly repeated measure, 3967
estimate, with subject, 3921
fat absorption data, 4061
fixed-effect solutions, 4047
full-rank parameterization, 4024
GDATA= option in RANDOM statement, 4040
geometrically anisotropic model, 3965
getting started, 3896
GLM procedure, split-plot design, 4017
graphics, box plots, 4076
graphics, influence diagnostics, 4061, 4072
graphics, residual panel, 4004
graphics, studentized residual panel, 4004
GROUP= effect in RANDOM statement, 4029
height data, 3896
holding covariance parameters fixed, 3944, 3964, 3965
IML procedure, reading ASYCOV, 4023
inference space, broad, 3918, 3920
inference space, intermediate, 3920
inference space, narrow, 3918, 3920
inference spaces, 4018
influence analysis, iterative, 4061, 4072
influence analysis, non-iterative, 4071
influence analysis, set deletion, 4071, 4072
influence analysis, tuples, 3937
intermediate inference space, 3920
known covariance parameters, 3943
known G and R matrix, 4040
Kronecker covariance structure, 3967
L-components, 3939, 4079, 4082
least squares mean, slice, 3928
least squares means, AT option, 3925
least squares means, covariate, 3925
least squares means, differences against control, 3926
line-source sprinkler data, 4055
local power-of-mean model, 3956
maximum likelihood estimation, 4020
mixed model equations, 4032, 4040
mixed model equations, solution, 4032, 4040
multiple plot requests, 3915
multiple traits data, 4039
multivariate analysis, 3967
narrow inference space, 3918, 3920
nested error structure, 4051
nested random effects, 3899
NOITER option, 3943, 4040
oven data (Hemmerle and Hartley, 1973), 4032
parameter grid search, 4032
pharmaceutical stability data, 4047
polynomial model, 4082
POM data set, 3956
POM fitting, iterated, 3957
Pothoff and Roy growth measurements, 4019, 4071
random coefficient model, 4025, 4047, 4075
random-effect solutions, 4047
residual panel, 4004
row-wise multiplicity adjustment, 3924
Satterthwaite method, 3924
set deletion, 4072
SGRENDER procedure, 4038
slice *F* test, 3928
spatial power structure, 4060
specifying lower bounds, 3944
specifying values for degrees of freedom, 3930
split-plot design, 3972, 4015
split-plot design, data, 4014, 4079
split-plot design, equivalent model, 4019
starting values, 4032
studentized maximum modulus, 3924
studentized residual panel, 4004
subject and no-subject formulation, 3971
subject contrasts, 3921
subject v. no-subject formulation, 4019

subject-specific R matrices, 3957
subject-specific V matrices, 3953
Toeplitxz structure, 4055
tuples, influence analysis, 3937
two-way analysis of variance, 3896
unstructured covariance, G-side, 3953
unstructured covariance, R-side, 4020, 4071
varying covariance parameters, 4029
examples, NLIN
array, constant, 4311
array, variable, 4311
biweight, 4319
boundary specifications, 4283
cancer remission data, 4322
conditional model expression, 4314
constant array, 4311
contrasts among parameters, 4337
convergence status, 4312
derivative code, 4293
derivatives, output, 4289
derivatives, zero, 4299
discontinuity, 4300
divergence, 4300
dose-response data, 4325
enzyme data, 4273
expected value parameterization, 4329
GENMOD procedure, 4323
getting started, 4272
GLIMMIX procedure, 4323
Hougaard's skewness measure, 4273, 4326, 4327, 4329
iteratively reweighted least squares, 4319
join point, 4314
join point, estimated, 4317
LD50 parameterization, 4327
local minimum, 4300
log-logistic model, 4326
LOGISTIC procedure, 4323
machine-level code, 4294
maximum likelihood estimation, 4322
model code, 4293
Newton-Raphson algorithm, 4322
non-identifiable parameter, 4299
one-compartment model, 4335
output, derivatives, 4289
output, predicted, 4329
output, predicted values, 4314
output, prediction limits, 4329
parameter differences, 4337
pharmacokinetic model, 4335
plateau model, 4314
plot, observed and predicted, 4316, 4329
predicted values, 4314
probit model, 4322
programming statements, 4292
programming statements, efficiency, 4297
PUT statement, 4314
reparameterization, 4317, 4327, 4329, 4337
reparameterizing constraint, 4284
robust regression, 4319
ROBUSTREG procedure, 4320
segmented model, 4314
SGPLOT procedure, 4316, 4325, 4326, 4329, 4332, 4340
SIGSQ= option, 4322
starting values, data set, 4291
starting values, grid, 4290
starting values, multiple, 4273
sum-of-squares reduction test, 4337, 4339
switching function, 4326
theophylline data, 4331
U.S. population growth data, 4319
variable array, 4311
varying parameters by groups, 4336
weight variable, 4319
weighted least-squares, 4298
examples, NLMIXED
binomial data, 4416
binomial-normal model, 4351
boundary specification, 4373
censored data, 4427
Cholesky parameterization, 4414
ESTIMATE statement, 4416
failure time model, 4427
frailty, 4427
gamma distribution, 4385
general distribution, 4420, 4427, 4432
getting started, 4346
GLIMMIX procedure, 4385, 4386
graphics, predicted profiles, 4436
group-specific random effects, 4416
growth curve, 4346
intraclass correlation coefficient, 4420
logistic-normal model, 4351
logsig parameterization, 4424
mutli-center clinical trial, 4351
negative binomial distribution, 4386
one-compartment model, 4412
orange tree data, 4346
pharmacokinetic model, 4412
Poisson-normal example, 4424
PREDICT statement, 4416
predicted profiles, graphics, 4436
probit-normal model, binomial, 4415
probit-normal model, ordinal, 4419
random frailty, 4432
SGPLOT procedure, 4436
single random effect, 4347, 4351, 4378

starting values from data set, 4400
theophylline data, 4409
three random effects, 4378
two random effects, 4378, 4412
user-specified log likelihood, 4420, 4427, 4432
excluded observations
PRINQUAL procedure, 5210, 5234
TRANSREG procedure, 7255
exclusion list
ODS, 439
exemplary data set
power and sample size (GLMPOWER), 2642, 2643, 2650, 2653, 2660, 2661, 2667
exogenous variables
CALIS procedure, 943
path diagram (CALIS), 944
expansion locus
theory (GLIMMIX), 2224
expected Fisher information
SEQDESIGN procedure, 5818, 5820
expected mean sample size
SEQTEST procedure, 6011
expected mean squares
computing, types (GLM), 2542
random effects, 2540
expected sample size
SEQDESIGN procedure, 5878
SEQTEST procedure, 6029
expected trend
MULTTEST procedure, 4206
expected value
definition (Introduction to Modeling), 57
of vector (Introduction to Modeling), 58
experimental design, 25, 2613, *see also* PLAN procedure
aliasing structure (GLM), 2610
experimentwise error rate (GLM), 2517
explicit intercept
TRANSREG procedure, 7254
exploratory data analysis, 25
VARIOGRAM procedure, 7520
exponential chi-square distribution
definition of (MCMC), 3532
MCMC procedure, 3513, 3532
exponential covariance structure
GLIMMIX procedure, 2201
MIXED procedure, 3960
exponential distribution
definition of (MCMC), 3534
GENMOD procedure, 2032
GLIMMIX procedure, 2168
MCMC procedure, 3513, 3534
exponential exponential distribution
definition of (MCMC), 3532
MCMC procedure, 3513, 3532
exponential family
Introduction to Modeling, 38
Introduction to Regression, 88
exponential gamma distribution
definition of (MCMC), 3533
MCMC procedure, 3513, 3533
exponential inverse chi-square distribution
definition of (MCMC), 3533
MCMC procedure, 3513, 3533
exponential inverse-gamma distribution
definition of (MCMC), 3533
MCMC procedure, 3513, 3533
exponential scaled inverse chi-square distribution
definition of (MCMC), 3534
MCMC procedure, 3513, 3534
exponential semivariogram model
KRIGE2D procedure, 2951
external studentization
MIXED procedure, 3987
external unfolding
MDS procedure, 3704
extreme value distribution
PROBIT procedure, 5311

F statistics
CLUSTER procedure, 1246, 1258
GENMOD procedure, 1981
factor
defined for factor analysis, 1535
factor analysis
compared to component analysis, 1534, 1535
factor analysis model
CALIS procedure, 834, 877, 879
COSAN statement (CALIS), 874
identification (CALIS), 877
identification (TCALIS), 6752
LINEQS statement (CALIS), 882
MATRIX statement (CALIS), 885
path diagram (CALIS), 902, 903
RAM statement (CALIS), 901
specification (CALIS), 840
structural model example (CALIS), 844
factor analytic structures
MIXED procedure, 3959
factor loadings
CALIS procedure, 920
TCALIS procedure, 6832
FACTOR model
simple factor model example (TCALIS), 6710

factor parsimax method, 1534, 1560, 1561, 6754, 6755
FACTOR procedure
  CALIS procedure, 847, 853, 877
  computational resources, 1578
  coverage displays, 1572
  degrees of freedom, 1563
  Heywood cases, 1576
  number of factors extracted, 1553
  ODS graph names, 1585
  OUT= data sets, 1559
  output data sets, 1540, 1559
  simplicity functions, 1537, 1559, 1572
  TCALIS procedure, 6753
  time requirements, 1574
  variances, 1563
Factor rotation
  with FACTOR procedure, 1541
factor rotation methods, 1534
factor scores
  CALIS procedure, 845, 920, 922, 937, 966
  displaying (CALIS), 966
  displaying (TCALIS), 6938
  TCALIS procedure, 6832, 6835, 6938
factor scoring coefficients
  FACTOR procedure, 5760
  SCORE procedure, 5760, 5770
factor structure, 1541
factor-analytic structure
  GLIMMIX procedure, 2197
factors
  PLAN procedure, 4736, 4737, 4745
  PLS procedure, 4766
failure time
  LIFEREG procedure, 2992
false discovery rate, 4211
  adjustment (MULTTEST), 4216
false negative, false positive rate
  LOGISTIC procedure, 3298, 3341, 3420
familywise error rate, 4211
  adjustment (MULTTEST), 4212
fast Fourier transform
  KDE procedure, 2902
  MULTTEST procedure, 4205
FASTCLUS procedure
  algorithm for updating cluster seeds, 1637
  bin-sort algorithm, 1634
  cluster deletion, 1635
  clustering criterion, 1622, 1636, 1637
  clustering methods, 1622, 1624
  compared to other procedures, 1647
  computational problems, convergence, 1635
  computational resources, 1647
  controlling iterations, 1638
  convergence criterion, 1634
  distance, 1622, 1623, 1642
  DRIFT option, 1623
  Ekblom-Newton algorithm, 1637
  homotopy parameter, 1635
  imputation of missing values, 1636
  incompatibilities, 1642
  iteratively reweighted least squares, 1636
  L_p clustering, 1622, 1636
  MEAN= data sets, 1638
  memory requirements, 1647
  Merle-Spath algorithm, 1637
  missing values, 1623, 1636, 1638, 1642
  Newton algorithm, 1637
  OUT= data sets, 1643
  outliers, 1622
  output data sets, 1638, 1643
  output table names, 1652
  OUTSTAT= data set, 1638, 1645
  random number generator, 1639
  scale estimates, 1635, 1637, 1642, 1644
  seed replacement, 1623, 1639
  weighted cluster means, 1639
Fay coefficient
  SURVEYLOGISTIC procedure, 6386, 6430
  SURVEYMEANS procedure, 6480, 6507
  SURVEYREG procedure, 6546, 6566
Fay's BRR method
  variance estimation (SURVEYFREQ), 6335
  variance estimation (SURVEYLOGISTIC), 6430
  variance estimation (SURVEYMEANS), 6507
  variance estimation (SURVEYREG), 6566
FDR, *see* false discovery rate
features, 5049
fiducial limits, 5265, 5313
finite differences
  theory (GLIMMIX), 2230
finite differencing
  NLMIXED procedure, 4361, 4392
finite population correction
  Introduction to Survey Procedures, 266
  SURVEYFREQ procedure, 6303
  SURVEYLOGISTIC procedure, 6384, 6422
  SURVEYMEANS procedure, 6476, 6477, 6492
  SURVEYREG procedure, 6544, 6545, 6561
first canonical variable, 1062
first-order algorithm
  Shared Concepts, 405
first-order method
  NLMIXED procedure, 4383
first-stage sampling units

Introduction to Survey Procedures, 266
Firth's penalized likelihood
LOGISTIC procedure, 3326
Fisher combination
adjustment (MULTTEST), 4215
Fisher exact test
MULTTEST procedure, 4199, 4201, 4208, 4236
Fisher information
SEQDESIGN procedure, 5818
Fisher information matrix
example (MIXED), 4032
MIXED procedure, 3997
Fisher scoring algorithm
LOGISTIC procedure, 3301, 3302, 3324
Fisher scoring method
GENMOD procedure, 1969
SURVEYLOGISTIC procedure, 6401, 6417
Fisher's exact test
FREQ procedure, 1731, 1734, 1735
Introduction to Nonparametric Analysis, 294
power and sample size (POWER), 4884, 4890, 4969
Fisher's LSD test, 714, 2473
Fisher's scoring method
GENMOD procedure, 1953
GLIMMIX procedure, 2095, 2119
MIXED procedure, 3905, 3915, 4012
Fisher's z test for correlation
power and sample size (POWER), 4840, 4843, 4932, 5002
fit diagnostics
examples (REG), 5592
fit statistics
GLIMMIX procedure, 2276
GLMSELECT procedure, 2727, 2746
PHREG procedure, 4639
fitted covariance matrix
TCALIS procedure, 6888
fitted mean vector
TCALIS procedure, 6888
fitting information
GLIMMIX procedure, 2276
fixed efectsl
HPMIXED procedure, 2822
fixed effects
GLIMMIX procedure, 2083
MIXED procedure, 3894
sum-to-zero assumptions, 2542
VARCOMP procedure, 7494, 7500
fixed-effects model
VARCOMP procedure, 7501
fixed-effects parameters
MIXED procedure, 3892, 3969
fixed-radius kernels
MODECLUS procedure, 4110
Fleiss-Cohen weights, 1768
Fleming-Harrington G_ρ test for homogeneity
LIFETEST procedure, 3100, 3130
flexible-beta method
CLUSTER procedure, 1230, 1240, 1241, 1255
floating point errors
MCMC procedure, 3565
NLMIXED procedure, 4399
folded form F statistic
TTEST procedure, 7418
fonts, modifying
ODS Graphics, 570
formatted values
DISTANCE procedure, 1514
formulas
CANCORR procedure, 1045
forward selection
GLMSELECT procedure, 2719
LOGISTIC procedure, 3301, 3328
PHREG procedure, 4570, 4617
REG procedure, 5435, 5523
Forward-Dolittle transformation, 2498
fraction of missing information
MI procedure, 3795
MIANALYZE procedure, 3856
fractional frequencies
PHREG procedure, 4562
STDIZE procedure, 6227
fractional sample size
GLMPOWER procedure, 2663
POWER procedure, 4825, 4924
frailty
NLMIXED procedure, 4427
random, example (NLMIXED), 4427
Freeman-Halton test, 1735
Freeman-Tukey test
MULTTEST procedure, 4201, 4206, 4229
FREQ procedure
alpha level, 1693, 1701
binomial proportion, 1745, 1809
Bowker's test of symmetry, 1765
Breslow-Day test, 1776
cell count data, 1724, 1799
chi-square tests, 1731, 1732, 1805, 1812
Cochran's Q test, 1765
Cochran-Mantel-Haenszel statistics, 1817
computation time, limiting, 1693
computational methods, 1780
computational resources, 1781, 1783
contingency coefficient, 1731
contingency table, 1812

continuity-adjusted chi-square, 1731, 1733
correlation statistic, 1772
Cramer's *V* statistic, 1731
default tables, 1697, 1698
displayed output, 1787
exact *p*-values, 1781
EXACT statement, used with TABLES, 1692
exact tests, 1691, 1779, 1819
Fisher's exact test, 1731
Friedman's chi-square statistic, 1822
gamma statistic, 1736
general association statistic, 1773
grouping variables, 1725
input data sets, 1688, 1724
Introduction to Nonparametric Analysis, 294, 295, 297
kappa coefficient, 1768, 1769
Kendall's tau-*b* statistic, 1736
lambda asymmetric, 1736
lambda symmetric, 1736
likelihood-ratio chi-square test, 1731
Mantel-Haenszel chi-square test, 1731
McNemar's test, 1765
measures of association, 1736
missing values, 1726
Monte Carlo estimation, 1691, 1693, 1782
multiway tables, 1789, 1791, 1792
network algorithm, 1780
odds ratio, 1760, 1774
ODS graph names, 1798
ODS table names, 1795
one-way frequency tables, 1731, 1732, 1787, 1788, 1805
order of variables, 1690
output data sets, 1694, 1784–1786, 1799, 1815
output variable prefixes, 1786
overall kappa coefficient, 1765
Pearson chi-square test, 1731, 1732
Pearson correlation coefficient, 1736
phi coefficient, 1731
polychoric correlation coefficient, 1736
relative risk, 1761, 1774, 1775
row mean scores statistic, 1773
scores, 1730
SCORES=RANK (Introduction to Nonparametric Analysis), 294
simple kappa coefficient, 1765
Somers' *D* statistics, 1736
Spearman rank correlation coefficient, 1736
statistical computations, 1730
stratified table, 1817
Stuart's tau-*c* statistic, 1736
two-way frequency tables, 1731, 1732, 1812
uncertainty coefficients, 1736
weighted kappa coefficient, 1765
FREQ statement
and RMSSTD statement (CLUSTER), 1248, 1249
frequency tables, 1676, 1697
generating (CATMOD), 1121, 1124
input to CATMOD procedure, 1139
one-way (FREQ), 1731, 1732, 1787, 1788, 1805
SURVEYFREQ procedure, 6312
two-way (FREQ), 1731, 1732, 1812
frequency variable
GLIMMIX procedure, 2140
LOGISTIC procedure, 3287
PRINQUAL procedure, 5211
programming statements (PHREG), 4575
SURVEYLOGISTIC procedure, 6395
TRANSREG procedure, 7138
value (PHREG), 4562
frequentist probability
Introduction to Bayesian Analysis, 142
Friedman's chi-square statistic, 1822
Friedman's test
Introduction to Nonparametric Analysis, 296
full sibs mating
INBREED procedure, 2873, 2874
full-rank coding
TRANSREG procedure, 7153
function
estimable, definition (Introduction to Modeling), 66
estimable, definition (Introduction to Regression), 102
furthest neighbor clustering, *see* complete linkage
futility index
SEQTEST procedure, 6022
fuzzy coding
CORRESP procedure, 1344
FWE, *see* familywise error rate

G matrix
GLIMMIX procedure, 2186, 2188
HPMIXED procedure, 2829, 2832
MIXED procedure, 3894, 3949, 3950, 3969, 3970, 4052
G-G epsilon, 2536
G-side random effect
GLIMMIX procedure, 2083
Gabriel's multiple-comparison procedure
ANOVA procedure, 712
GLM procedure, 2471, 2519
GAM procedure

comparing PROC GAM with PROC LOESS, 1877
Estimates from PROC GAM, 1860
generalized additive model with binary data, 1862
graphics, 1840
ODS graph names, 1862
ODS table names, 1861
Poisson regression analysis of component reliability, 1871
response level ordering, 1843
response variable options, 1843
gamma distribution, 2992, 3022, 3039
definition of (MCMC), 3534
GENMOD procedure, 1965
GLIMMIX procedure, 2168
MCMC procedure, 3513, 3534
NLMIXED procedure, 4376
gamma error spending function
SEQDESIGN procedure, 5806
gamma error spending method
SEQDESIGN procedure, 5850
gamma statistic, 1736, 1738
gauge R&R
VARCOMP procedure, 7498, 7503
Gaussian assumption
SIM2D procedure, 6156
gaussian covariance structure
GLIMMIX procedure, 2201
MIXED procedure, 3960
Gaussian distribution
definition of (MCMC), 3538
MCMC procedure, 3514, 3538
NLMIXED procedure, 4376
gaussian distribution
GLIMMIX procedure, 2168
Gaussian random field
SIM2D procedure, 6155
Gaussian semivariogram model
KRIGE2D procedure, 2950
GCV function, TPSPLINE procedure, 7057, 7059, 7076, 7084
nonhomogeneous variance, 7076
Geary *c* coefficient, *see* autocorrelation
GEE, *see* Generalized Estimating Equations, *see also* generalized estimating equations
Gehan test, *see* Wilcoxon test for homogeneity
power and sample size (POWER), 4901, 4910, 4975
Gelman-Rubin diagnostics
Bayesian analysis (PHREG), 4639
MCMC procedure, 3693
general association statistic, 1773
general distribution
MCMC procedure, 3514, 3541
NLMIXED procedure, 4376
General Effects
Shared Concepts, 372
general linear covariance structure
GLIMMIX procedure, 2197
MIXED procedure, 3959
general linear model
Introduction to Regression, 77
general linear models, 5049, 5111
contrast, 5120
covariates, 5126
defining factors, 5112
model selection, 5113, 5124
generalized Crawford-Ferguson family, 1534
generalized Crawford-Ferguson method, 1560, 1561, 6755, 6756
generalized cross validation function (GCV), 7057, 7084
generalized cyclic incomplete block design
generating with PLAN procedure, 4754
generalized estimating equations
compound symmetry (GLIMMIX), 2388
Introduction to ANOVA Procedures, 118
Introduction to Regression, 76, 89, 90
working independence (GLIMMIX), 2098, 2234
Generalized Estimating Equations (GEE), 1918, 1958, 1986, 2036, 2042
generalized Euclidean distance coefficient
DISTANCE procedure, 1509
generalized inverse, 3977
MIXED procedure, 3919
NLIN procedure, 4302
NLMIXED procedure, 4362
generalized least squares, *see* weighted least squares estimation
generalized linear mixed model, *see also* GLIMMIX procedure
Introduction to Regression, 77, 89
generalized linear mixed model (GLIMMIX)
least squares means, 2155
theory, 2218
generalized linear model, *see also* GLIMMIX procedure
GENMOD procedure, 1896, 1897
Introduction to ANOVA Procedures, 118
Introduction to Modeling, 37, 68, 70, 71
Introduction to Regression, 76, 88, 89, 93
theory (GENMOD), 1964
generalized linear model (GLIMMIX)
theory, 2212
generalized linear models
MCMC procedure, 3592

generalized logit
example (GLIMMIX), 2125, 2136
generalized logit model
SURVEYLOGISTIC procedure, 6426
generalized logits, *see also* response functions
examples (CATMOD), 1146
formulas (CATMOD), 1169
specifying in CATMOD procedure, 1131
using (CATMOD), 1146
generalized Poisson distribution
GLIMMIX procedure, 2398
generalized two-sided test
SEQDESIGN procedure, 5826
generation (INBREED)
nonoverlapping, 2857, 2861, 2862
number, 2866
overlapping, 2857, 2860
variable, 2866
GENMOD procedure
adjusted residuals, 1984
AIC, 1971
Akaike information criterion, 1971
aliasing, 1904
Bayesian analysis linear regression, 1906
Bayesian information criterion, 1971
BIC, 1971
binomial distribution, 1966
built-in link function, 1898
built-in probability distribution, 1898
case deletion diagnostics, 1999
classification variables, 1974
confidence intervals, 1947
confidence limits, 1945
continuous variables, 1974
contrasts, 1943
convergence criterion, 1947, 1959
correlated data, 1895, 1986
correlation matrix, 1948, 1970
correlations, least-squares means, 1945
covariance matrix, 1948, 1970
covariances, least-squares means, 1946
crossed effects, 1974
design matrix, 1975
deviance, 1947
deviance definition, 1899
deviance residuals, 1984
diagnostics, 1948, 1999
dispersion parameter, 1972
dispersion parameter estimation, 1898, 1978, 1979
dispersion parameter weights, 1962
effect specification, 1974
estimability, 1945
estimability checking, 1942
events/trials format for response, 1946, 1966
expected information matrix, 1969
exponential distribution, 2032
F statistics, 1981
Fisher scoring method, 1969
Fisher's scoring method, 1953
gamma distribution, 1965
GEE, 1895, 1918, 1958, 1986, 2036, 2039, 2042
Generalized Estimating Equations (GEE), 1895
generalized linear model, 1896, 1897
geometric distribution, 1965
goodness of fit, 1970
gradient, 1969
Hessian matrix, 1969
information matrix, 1953
initial values, 1949, 1959
intercept, 1899, 1902, 1950
inverse Gaussian distribution, 1965
L matrices, 1945
Lagrange multiplier statistics, 1981
life data, 2029
likelihood residuals, 1984
linear model, 1896
linear predictor, 1894, 1896, 1902, 1975, 2011
link function, 1894, 1896, 1967
log-likelihood functions, 1967
log-linear models, 1901
logistic regression, 2024
main effects, 1974
maximum likelihood estimation, 1968
MEAN automatic variable, 1957
Model checking, 2047
model checking, 2054
multinomial distribution, 1966
multinomial models, 1984
negative binomial distribution, 1965
nested effects, 1974
Newton-Raphson algorithm, 1968
normal distribution, 1965
observed information matrix, 1969
offset, 1952, 2011
offset variable, 1901
ordinal data, 2032
output data sets, 2007
output ODS Graphics table names, 2022
output table names, 2019
overdispersion, 1973
Pearson residuals, 1984
Pearson's chi-square, 1947, 1970, 1971
Poisson distribution, 1966
Poisson regression, 1900

polynomial effects, 1974
profile likelihood confidence intervals, 1950, 1980
programming statements, 1957
QIC, 1993
quasi-likelihood, 1973
quasi-likelihood functions, 1994
quasi-likelihood information criterion, 1993
raw residuals, 1983
regression parameters estimation, 1898
regressor effects, 1974
repeated measures, 1895, 1986
residuals, 1952, 1983, 1984
RESP automatic variable, 1957
scale parameter, 1966
scaled deviance, 1970
score statistics, 1981
singular contrast matrix, 1942
subpopulation, 1947
suppressing output, 1926
Type 1 analysis, 1899, 1978
Type 3 analysis, 1899, 1979
user-defined link function, 1944
variance function, 1898
Wald confidence intervals, 1954, 1981
working correlation matrix, 1959–1961, 1987
XBETA automatic variable, 1957
zero-inflated Poisson distribution, 1966
zero-inflated Poisson models, 1985
Gentleman-Givens computational method, 4505
geometric anisotropy
KRIGE2D procedure, 2956, 2957
geometric distribution
definition of (MCMC), 3534
GENMOD procedure, 1965
GLIMMIX procedure, 2168
MCMC procedure, 3513, 3534
Geweke diagnostics
Bayesian analysis (PHREG), 4640
Gibbs sampler
Introduction to Bayesian Analysis, 151, 154
Markov chain Monte Carlo, 151, 154
GLIMMIX procedure
adaptive Gaussian quadrature, 2103
Akaike's information criterion, 2099
Akaike's information criterion (finite sample corrected version), 2099
alpha level, 2130, 2136, 2145, 2157, 2164, 2180, 2187
anisotropic power covariance structure, 2202
anisotropic spatial power structure, 2202
ANOM adjustment, 2144
anom plot, 2288
ANTE(1) structure, 2193
ante-dependence structure, 2193
AR(1) structure, 2194
asymptotic covariance, 2095, 2099
automatic variables, 2140, 2209
autoregressive moving-average structure, 2194
autoregressive structure, 2194
B-spline basis, 389
banded Toeplitz structure, 2203
Bernoulli distribution, 2168
beta distribution, 2168
between-within method, 2166
bias of estimates, 2232
binary distribution, 2168
binomial distribution, 2168
BLUP, 2192, 2209, 2210, 2278
Bonferroni adjustment, 2144
boundary constraints, 2182, 2185
box plots, 2282
BYLEVEL processing of LSMEANS, 2146, 2149, 2157
centering, 2172
chi-square mixture, 2132
chi-square test, 2126, 2157, 2164, 2167
Cholesky covariance structure, 2195
Cholesky method, 2096
Cholesky root, 2195, 2197
class level, 2108, 2274
classification variables, 2122
collection effect, 378
comparing splines, 2406
comparison with the MIXED procedure, 2268
compound symmetry structure, 2196
computed variables, 2140
confidence interval, 2138, 2159, 2187
confidence limits, 2136, 2146, 2157, 2164, 2187
confidence limits, covariance parameters, 2130
constrained covariance parameters, 2133
constructed effects, 2134, 2412
containment method, 2166
continuous effects, 2193
contrast-specification, 2123, 2134
contrasts, 2123
control plot, 2288
convergence criterion, 393, 395, 396, 403, 2095, 2105, 2111, 2269, 2276, 2303, 2349
convergence status, 2276
correlations of least squares means, 2146

correlations of least squares means contrasts, 2157
covariance parameter estimates, 2102, 2277
covariance parameters, 2084
covariance structure, 2193, 2204
covariances of least squares means, 2146
covariances of least squares means contrasts, 2157
covariate values for LSMEANS, 2145, 2157
crossed effects, 2261
crossover designs, 4759
default estimation technique, 2273
default output, 2273
default variance function, 2208
degrees of freedom, 2125, 2126, 2132, 2135, 2136, 2142, 2143, 2146, 2156, 2157, 2165, 2166, 2175, 2241, 2262
diagnostic plots, 2282
diffogram, 2150, 2288, 2292, 2293, 2318
dimension information, 2275
dispersion parameter, 2215
doubly iterative algorithm, 2271
Dunnett's adjustment, 2144
EBE, 2193, 2278
EBLUP, 2192
effect name length, 2108
empirical Bayes estimates, 2104, 2193, 2278
empirical Bayes estimation, 2104
empirical estimator, 2246
estimability, 2124, 2126, 2138, 2141, 2175, 2262
estimated-likelihood interval, 2131
estimates, 2134
estimation methods, 2102
estimation modes, 2233
examples, *see also* examples, GLIMMIX, 2300
expansion locus, 2224
exponential covariance structure, 2201
exponential distribution, 2168
factor-analytic structure, 2197
finite differences, 2230
Fisher's scoring method, 2095, 2119
fit statistics, 2276
fitting information, 2276
fixed effects, 2083
fixed-effects parameters, 2175
functional convergence criteria, 394
G matrix, 2186, 2188
G-side random effect, 2083
gamma distribution, 2168
gaussian covariance structure, 2201
gaussian distribution, 2168
general linear covariance structure, 2197
generalized linear mixed model theory, 2218
generalized linear model theory, 2212
generalized Poisson distribution, 2398
geometric distribution, 2168
GLM mode, 2098, 2215, 2233, 2234
GLMM mode, 2098, 2234
grid search, 2181
group effect, 2188
Hannan-Quinn information criterion, 2099
Hessian matrix, 2095, 2099
Hessian scaling, 396
heterogeneous AR(1) structure, 2194
heterogeneous autoregressive structure, 2194
heterogeneous compound symmetry structure, 2197
heterogeneous Toeplitz structure, 2203
Hsu's adjustment, 2144
Huynh-Feldt covariance structure, 2197
infinite degrees of freedom, 2126, 2136, 2146, 2157, 2167
information criteria, 2099
initial values, 2181
input data sets, 2096
integral approximation, 2218
interaction effects, 2261
intercept, 2261
intercept random effect, 2186
introductory example, 2086
inverse gaussian distribution, 2168
iteration details, 2101
iteration history, 2275
iterations, 2275
Kackar-Harville-Jeske adjusted estimator, 2246
Kenward-Roger method, 2167
knot selection, 2251
KR adjusted estimator, 2246
L matrices, 2123, 2141
lag functionality, 411, 2208
Laplace approximation, 2102, 2224
least squares means, 2141, 2147, 2153, 2821
likelihood ratio test, 2127, 2234
line-search methods, 398
line-search precision, 399
linear covariance structure, 2197
linearization, 2218, 2220
link function, 2083, 2171
log-normal distribution, 2168
Matérn covariance structure, 2201
maximum likelihood, 2102, 2273
missing level combinations, 2262
MIVQUE0 estimation, 2184, 2275

mixed model smoothing, 2189, 2191, 2198, 2199, 2250
model information, 2274
multimember effect, 378, 2412
multimember example, 2412
multinomial distribution, 2168
multiple comparisons of estimates, 2135
multiple comparisons of least squares means, 2143, 2144, 2147, 2153, 2156
multiplicity adjustment, 2135, 2138, 2143, 2144, 2154, 2156, 2159
negative binomial distribution, 2168
Nelson's adjustment, 2144
nested effects, 2261
Newton-Raphson algorithm, 402
Newton-Raphson algorithm with ridging, 402
non-full-rank parameterization, 2262
non-positional syntax, 2124, 2264, 2406
normal distribution, 2168
notation, 2083
number of observations, 2274
numerical integration, 2227
odds estimation, 2256
odds ratio estimation, 2256
odds ratios, 2109
ODS graph names, 2299
ODS Graphics, 2112, 2282
ODS table names, 2280
offset, 2174, 2209, 2322, 2323, 2386
optimization, 2175
optimization information, 2275
optimization technique, 401
ordering of effects, 2110
output statistics, 2278
overdispersion, 2398
P-spline, 2198
parameterization, 2261
penalized B-spline, 2198
Poisson distribution, 2168
Poisson mixture, 2398
polynomial effect, 380
population average, 2224
positive definiteness, 2195
power covariance structure, 2202
profile-likelihood interval, 2131
profiling residual variance, 2109, 2119
programming statements, 2207
pseudo-likelihood, 2102, 2273
quadrature approximation, 2102, 2227
quasi-likelihood, 2273
R-side random effect, 2083, 2192
radial smoother structure, 2199
radial smoothing, 2189, 2199, 2250
random effects, 2083, 2186
random-effects parameter, 2192
reference category, 2267
remote monitoring, 401, 403
residual effect, 2186
residual likelihood, 2102
residual maximum likelihood, 2273
residual method, 2167
residual plots, 2282
response level ordering, 2163, 2267
response profile, 2267, 2274
response variable options, 2163
restricted maximum likelihood, 2273
sandwich estimator, 2246
Satterthwaite method, 2167, 2241
scale parameter, 2083–2085, 2100, 2109, 2119, 2131, 2181, 2182, 2185, 2208, 2212, 2215–2217, 2221, 2222, 2224, 2228, 2239, 2270, 2275, 2277, 2315, 2331, 2335, 2348, 2352, 2381, 2399
Schwarz's Bayesian information criterion, 2099
scoring, 2095
Sidak's adjustment, 2144
simple covariance matrix, 2201
simple effects, 2152
simple effects differences, 2152
simulation-based adjustment, 2144
singly iterative algorithm, 2271
spatial covariance structure, 2201
spatial exponential structure, 2201
spatial gaussian structure, 2201
spatial Matérn structure, 2201
spatial power structure, 2202
spatial spherical structure, 2202
spherical covariance structure, 2202
spline bases, 387
spline comparisons, 2406
spline effect, 384
spline smoothing, 2198, 2199
standard error adjustment, 2096, 2167
statistical graphics, 2282, 2299
subject effect, 2193
subject processing, 2247
subject-specific, 2224
t distribution, 2168
table names, 2280
test-specification for covariance parameters, 2127
testing covariance parameters, 2127, 2234
tests of fixed effects, 2278
thin plate spline (approx.), 2250
Toeplitz structure, 2202
TPF basis, 388

truncated power function basis, 388
Tukey's adjustment, 2144
Type I testing, 2171
Type II testing, 2171
Type III testing, 2171
unstructured covariance, 2203
unstructured covariance matrix, 2197
user-defined link function, 2208
V matrix, 2205
Wald test, 2278
Wald tests of covariance parameters, 2134
weighting, 2206
GLM, *see also* GLIMMIX procedure
GLM parameterization
Shared Concepts, 374
SURVEYLOGISTIC procedure, 6412
GLM procedure
absorption of effects, 2452, 2507
aliasing structure, 2475, 2610
alpha level, 2460, 2469, 2475, 2479
Bartlett's test, 2471, 2526
Bonferroni adjustment, 2459
Brown and Forsythe's test, 2471, 2527
canonical analysis, 2465
characteristic roots and vectors, 2464
compared to other procedures, 2434, 2480, 2540, 2597, 2613, 3895, 4254, 4505, 4506, 5720
comparing groups, 2511
computational method, 2547
computational resources, 2544
contrasts, 2454, 2483
covariate values for least squares means, 2460
disk space, 2447
Dunnett's adjustment, 2459
effect sizes, 2502
effect specification, 2488
error effect, 2464
estimability, 2455–2457, 2463, 2477, 2497, 2511
estimable functions, 2496
ESTIMATE specification, 2509
homogeneity of variance tests, 2471, 2526
Hsu's adjustment, 2459
hypothesis tests, 2485, 2496
interactive use, 2490
interactivity and BY statement, 2453
interactivity and missing values, 2491, 2544
introductory example, 2435
least squares means (LS-means), 2458
Levene's test for homogeneity of variance, 2471, 2527
means, 2468
means versus least squares means, 2512
memory requirements, reduction of, 2452
missing values, 2446, 2464, 2530, 2543
model specification, 2488
multiple comparisons, least squares means, 2459, 2462, 2514, 2516
multiple comparisons, means, 2470–2473, 2514, 2516
multiple comparisons, procedures, 2468
multivariate analysis of variance, 2446, 2464, 2531
Nelson's adjustment, 2459
nonstandard weights for least squares means, 2461
O'Brien's test, 2471
observed margins for least squares means, 2461
ODS graph names, 2555
ODS table names, 2551
output data sets, 2477, 2548, 2549
parameterization, 2491
positional requirements for statements, 2444
predicted population margins, 2458
Q effects, 2541
random effects, 2480, 2540
regression, quadratic, 2438
relation to GLMMOD procedure, 2621
repeated measures, 2481, 2532
Sidak's adjustment, 2459
simple effects, 2463
simulation-based adjustment, 2459
singularity checking, 2455, 2457, 2463, 2476
sphericity tests, 2484, 2536
SSCP matrix for multivariate tests, 2464
statistical assumptions, 2487
summary of features, 2433
tests, hypothesis, 2454
transformations for MANOVA, 2464
transformations for repeated measures, 2483
Tukey's adjustment, 2459
types of least squares means comparisons, 2462
unbalanced analysis of variance, 2435, 2512, 2565
unbalanced design, 2435, 2512, 2540, 2565, 2595
weighted analysis, 2486
weighted means, 2527
Welch's ANOVA, 2473
WHERE statement, 2491
GLMM, *see also* GLIMMIX procedure
GLMMOD alternative
TRANSREG procedure, 7173, 7293

GLMMOD procedure
design matrix, 2621, 2629, 2631
input data sets, 2626
introductory example, 2622
missing values, 2630, 2631
ODS table names, 2631
output data sets, 2627, 2630, 2631
relation to GLM procedure, 2621
screening experiments, 2637
GLMPOWER procedure
actual power, 2661, 2663, 2669
alpha level, 2658
analysis of variance, 2643, 2666, 2672
ceiling sample size, 2663
compared to other power and sample size tools, 414
compared to other procedures, 2642, 4818
computational methods, 2664
contrasts, 2647, 2652, 2665, 2666, 2672
covariates, class and continuous, 2653, 2658–2660, 2666, 2672
displayed output, 2662
exemplary data set, 2642, 2643, 2650, 2653, 2660, 2661, 2667
fractional sample size, 2663
introductory example, 2643
nominal power, 2661, 2663, 2669
number-lists, 2660
ODS table names, 2663
plots, 2642, 2649, 2651, 2654
positional requirements for statements, 2649
sample size adjustment, 2661
summary of analyses, 414
summary of statements, 2650
value lists, 2660
GLMSELECT procedure
ANOVA table, 2745
B-spline basis, 389
backward elimination, 2722
Building the SSCP Matrix, 2737
candidates for addition or removal, 2744
class level coding, 2744
class level information, 2744
collection effect, 378
constructed effects, 2706
cross validation, 2741
cross validation details, 2747
dimension information, 2744
displayed output, 2743
fit statistics, 2727, 2746
forward selection, 2719
hierarchy, 2708
lasso selection, 2725
least angle regression, 2725
macro variables, 2734
model hierarchy, 2708
model information, 2743
model selection, 2719
model selection issues, 2726
multimember effect, 378
number of observations, 2743
ODS graph names, 2755
ODS graphics, 2749
output table names, 2748
Parallel BY Group Computation, 2738
parameter estimates, 2747
performance settings, 2743
polynomial effect, 380
score information, 2748
selected effects, 2745
selection summary, 2744
spline bases, 387
spline effect, 384
stepwise selection, 2723
stop details, 2745
stop reason, 2745
test data, 2739
timing breakdown, 2748
TPF basis, 388
truncated power function basis, 388
validation data, 2739
GLMSelect procedure
introductory example, 2686
global annotate
traditional graphics (LIFETEST), 3113
global influence
LD statistic (PHREG), 4574, 4614
LMAX statistic (PHREG), 4574, 4615
global kriging
KRIGE2D procedure, 2931
global null hypothesis
PHREG procedure, 4528, 4597, 4634
score test (PHREG), 4569, 4646
GLS method
TCALIS procedure, 6888
Gompertz distribution, 5258
goodness of fit
GENMOD procedure, 1970
Gower's dissimilarity coefficient
DISTANCE procedure, 1508
Gower's method, *see also* median method
CLUSTER procedure, 1240, 1255
Gower's similarity coefficient
DISTANCE procedure, 1508
gradient
CALIS procedure, 913, 945
GENMOD procedure, 1969
LOGISTIC procedure, 3331

MIXED procedure, 3906, 3996
SURVEYLOGISTIC procedure, 6433
TCALIS procedure, 6823, 6923
Graeco-Latin square
generating with PLAN procedure, 4742
graph
customizing, 5058, 5079, 5137
graph label
ODS Graphics, 538
graph modification
ODS Graphics, 525
graph name
ODS Graphics, 538, 607
graph resolution
ODS Graphics, 525, 547
graph size
ODS Graphics, 525, 547
graph template language
ODS Graphics, 577
graph templates
ODS Graphics, 577
graph templates, customizing
ODS Graphics, 587
graph templates, definition
ODS Graphics, 612
graph templates, displaying
ODS Graphics, 583
graph templates, editing
ODS Graphics, 585
graph templates, locating
ODS Graphics, 582
graph templates, reverting to default
ODS Graphics, 588
graph templates, saving
ODS Graphics, 587, 613
graph titles, modifying
ODS Graphics, 613
graphics, *see* plots
keywords (REG), 5490
options (REG), 5491
saving output (MI), 3759
traditional plots (REG), 5489
TTEST procedure, 7427
graphics catalog, specifying
LIFEREG procedure, 3006
PROBIT procedure, 5265
graphics image file
file type, 542
ODS Graphics, 541–543
PostScript, 546, 604
graphics image file, saving
ODS Graphics, 544
graphics image file, type
ODS Graphics, 542
graphs, *see* plots
Greenhouse-Geisser epsilon, 2536
grid lines
ODS Graphics, 619
grid search
example (MIXED), 4032
HPMIXED procedure, 2826
group average clustering, *see* average linkage
group comparisons
NLIN procedure, 4331
group effect
GLIMMIX procedure, 2188
group sequential design, 5784
SEQDESIGN procedure, 5827
group sequential trial
SEQDESIGN procedure, 5784, 5984
grouped-name-lists
POWER procedure, 4919
grouped-number-lists
POWER procedure, 4919
growth curve analysis
example (CATMOD), 1208
example (MIXED), 3970
GSK models, 1097
GT2 multiple-comparison method, 714, 2473, 2519

H-F epsilon, 2536
Hadamard matrix
BRR variance estimation (SURVEYFREQ), 6336
SURVEYLOGISTIC procedure, 6386, 6432
SURVEYMEANS procedure, 6480, 6509
SURVEYREG procedure, 6546, 6568
half-fraction design, analysis, 2608
half-width, confidence intervals, 416, 4916
Hall-Wellner bands
LIFETEST procedure, 3114, 3138, 3175
Hamann coefficient
DISTANCE procedure, 1512
Hamming distance coefficient
DISTANCE procedure, 1511
handling error messages
MCMC procedure, 3568
Hannan-Quinn information criterion
GLIMMIX procedure, 2099
HPMIXED procedure, 2811
MIXED procedure, 3907
Hanurav-Vijayan method
SURVEYSELECT procedure, 6648
Harris component analysis, 1534, 1536, 1552
Harris-Kaiser method, 1534, 1561
hat matrix, 5560
LOGISTIC procedure, 3349

Haybittle-Peto method
SEQDESIGN procedure, 5791, 5806, 5829, 5844, 5875
hazard function
baseline (PHREG), 4523, 4524
cumulative (PHREG), 4616
definition (PHREG), 4579
discrete (PHREG), 4572, 4579
LIFETEST procedure, 3100, 3182
PHREG procedure, 4523
rate (PHREG), 4667
ratio (PHREG), 4528, 4530
SEQDESIGN procedure, 5867
hazard ratios
Bayesian analysis (PHREG), 4628
confidence intervals (PHREG), 4570, 4634
estimates (PHREG), 4634, 4667
PHREG procedure, 4528, 4640
profile-likelihood confidence limits (PHREG), 4569, 4599
Wald's confidence limits (PHREG), 4569, 4599
hazards ratios
confidence intervals (PHREG), 4598
Heidelberger-Welch diagnostics
Bayesian (PHREG) procedure, 4640
Hertzsprung-Russell Plot, example
MODECLUS procedure, 4170
Hessian matrix
CALIS procedure, 855, 895, 896, 913, 927, 945, 946
GENMOD procedure, 1969
GLIMMIX procedure, 2095, 2099
LOGISTIC procedure, 3301, 3331
MIXED procedure, 3905, 3906, 3915, 3944, 3996, 3997, 4011, 4012, 4023, 4032
NLMIXED procedure, 4364
SURVEYLOGISTIC procedure, 6401, 6433
TCALIS procedure, 6823, 6897, 6923, 6924
Hessian scaling
GLIMMIX procedure, 396
NLMIXED procedure, 4394
heterogeneity
example (MIXED), 4029
HPMIXED procedure, 2830
MIXED procedure, 3951, 3955
heterogeneous
AR(1) structure (MIXED), 3959
compound-symmetry structure (MIXED), 3959
covariance structures (MIXED), 3968
Toeplitz structure (MIXED), 3959
heterogeneous AR(1) structure
GLIMMIX procedure, 2194
heterogeneous autoregressive structure
GLIMMIX procedure, 2194
heterogeneous compound symmetry structure
GLIMMIX procedure, 2197
heterogeneous Toeplitz structure
GLIMMIX procedure, 2203
heteroscedasticity
Introduction to Modeling, 69
testing (REG), 5575
Heywood cases
FACTOR procedure, 1576
hierarchical centering
MCMC procedure, 3615
hierarchical clustering, 1239, 1254, 7463
hierarchical design
generating with PLAN procedure, 4750
hierarchical factor model
TCALIS procedure, 7005, 7012
hierarchical model
example (MIXED), 4047
hierarchy
GLMSELECT procedure, 2708
LOGISTIC procedure, 3294
PHREG procedure, 4567
higher-order factor model
TCALIS procedure, 7005, 7006
highest posterior density (HPD) intervals
definition of, 149
Introduction to Bayesian Analysis, 149, 171
histogram
ODS Graphics, 604
Hochberg
adjustment (MULTTEST), 4215
Hochberg's GT2 multiple-comparison method, 714, 2473, 2519
Hodges-Lehmann estimation
NPAR1WAY procedure, 4472
Hommel
adjustment (MULTTEST), 4215
homogeneity analysis
CORRESP procedure, 1320
homogeneity of variance tests, 712, 2471, 2526
Bartlett's test (ANOVA), 712
Bartlett's test (GLM), 2471, 2526
Brown and Forsythe's test (ANOVA), 712
Brown and Forsythe's test (GLM), 2471, 2527
DISCRIM procedure, 1395
examples, 2605
Levene's test (ANOVA), 713
Levene's test (GLM), 2471, 2527
O'Brien's test (ANOVA), 713
O'Brien's test (GLM), 2471
Welch's ANOVA, 2527

homogeneity tests
LIFETEST procedure, 3100, 3107, 3142, 3167
homoscedasticity
Introduction to Modeling, 61
homotopy parameter
FASTCLUS procedure, 1635
honestly significant difference test, 714, 2473, 2518, 2520
Hosmer-Lemeshow test
LOGISTIC procedure, 3296, 3344
test statistic (LOGISTIC), 3345
Hotelling-Lawley trace, 706, 2464, 2536
Hotelling-Lawley-McKeon statistic
MIXED procedure, 3955
Hotelling-Lawley-Pillai-Samson statistic
MIXED procedure, 3955
how to use
alternate forms, 5078
Preferences, 5072
Results page, 5080
Howe's solution, 1539
HPD intervals
credible intervals (PHREG), 4556, 4639
HPMIXED procedure
Akaike's information criterion, 2811, 2838
Akaike's information criterion (finite sample corrected version) , 2811, 2838
alpha level, 2819, 2821, 2822, 2830
average information, 2806, 2834
basic features, 2804
BLUPs, 2830
boundary constraints, 2827, 2829
chi-square test, 2817, 2823
class level, 2813
classification variables, 2815
comparing HPMIXED and MIXED, 2843
confidence interval, 2830
confidence limits, 2819, 2821, 2822, 2826, 2830
conjugate gradient algorithm, 2806
continuous effects, 2830, 2831
contrast specification, 2816
contrasts, 2816
convergence status, 2838
correlations of least squares means, 2822
covariance parameter estimates, 2838
covariances of least squares means, 2822
degrees of freedom, 2817, 2819, 2821–2823
dimensions, 2813
effect name length, 2813
EM-REML, 2834
estimability, 2816, 2818, 2820, 2821, 2823
estimates, 2818
estimation methods, 2812
expected information, 2834
first and second derivatives, 2833
fitting information, 2838
fixed efectsl, 2822
fixed-effects parameters, 2823
G matrix, 2829, 2832
grid search, 2826
Hannan-Quinn information criterion, 2811
heterogeneity, 2830
hypothesis tests, 2837
infinite degrees of freedom, 2819, 2821, 2823
information criteria, 2811
initial values, 2826
input data sets, 2811
intercept effect, 2823, 2830
introductory example, 2808
iteration details, 2812
iterations, 2838
L matrices, 2816, 2821
likelihood computation, 2833
microarray data, 2850
mixed model, 2822
mixed model equations, 2813
model assumptions, 2832
model information, 2813
number of observations, 2813
ODS table names, 2839
ordering of effects, 2814
parameter constraints, 2827
profiling residual variance, 2813
R matrix, 2832
random effects, 2829
random-effects parameter, 2830
residual likelihood, 2812
residual method, 2823
residual variance tolerance, 2814
restricted maximum likelihood, 2812
rounding error, 2836
Schwarz's Bayesian information criterion, 2811, 2838
singularity, 2814
sparse matrix storage, 2836
sparse matrix techniques, 2806, 2835
starting values, 2834
subject effect, 2831
table names, 2839
type III tests, 2831
variance ratios, 2827
weighting, 2832
HSD test, 714, 2473, 2518, 2520
Hsu's adjustment
GLIMMIX procedure, 2144

GLM procedure, 2459
MIXED procedure, 3924
HTML destination
ODS, 448
ODS Graphics, 541, 545
HTML links
ODS, 477, 482
Huynh-Feldt
epsilon (GLM), 2536
structure (GLM), 2536
structure (MIXED), 3959
stucture (GLIMMIX), 2197
HYBRID option
and FREQ statement (CLUSTER), 1248
and other options (CLUSTER), 1246
PROC CLUSTER statement, 1252
hypergeometric
distribution (MULTTEST), 4209
variance (MULTTEST), 4204
hypothesis test
mixed model (HPMIXED), 2831
hypothesis testing
Introduction to Modeling, 66
hypothesis tests
comparing adjusted means (GLM), 2463
contrasts (CATMOD), 1110
contrasts (GLM), 2454
contrasts, examples (GLM), 2559, 2579, 2588
custom tests (ANOVA), 719
customized (GLM), 2485
exact (FREQ), 1691
for intercept (ANOVA), 715
for intercept (GLM), 2476
GLM procedure, 2496
incorrect hypothesis (CATMOD), 1165
lack of fit (RSREG), 5736
MANOVA (GLM), 2531
mixed model (MIXED), 3978, 3998
multivariate (REG), 5577
nested design (NESTED), 4259
parametric, comparing means (TTEST), 7392, 7431
parametric, comparing variances (TTEST), 7418, 7431
random effects (GLM), 2481, 2540
REG procedure, 5480, 5506
repeated measures (GLM), 2535
TRANSREG procedure, 7264
Type I sum of squares (GLM), 2498
Type II sum of squares (GLM), 2499
Type III sum of squares (GLM), 2501
Type IV sum of squares (GLM), 2501
ID variables
TRANSREG procedure, 7138
ideal point model
TRANSREG procedure, 7180
ideal point models
TRANSREG procedure, 7345
identification variables, 4375
identity transformation
PRINQUAL procedure, 5216
TRANSREG procedure, 7147
ill-conditioned data
ORTHOREG procedure, 4505
image component analysis, 1534, 1536, 1552
implicit intercept
TRANSREG procedure, 7254
imputation methods
MI procedure, 3773
imputation model
MI procedure, 3798
imputation of missing values
FASTCLUS procedure, 1636
imputer's model
MI procedure, 3796
INBREED procedure
coancestry, computing, 2870
coefficient of relationship, computing, 2869
covariance coefficients, 2857, 2860, 2862, 2864, 2867, 2869
covariance coefficients matrix, output, 2865
first parent, 2866, 2867
full sibs mating, 2873, 2874
generation number, 2866
generation variable, 2866
generation, nonoverlapping, 2857, 2861, 2862
generation, overlapping, 2857, 2860
inbreeding coefficients, 2858, 2860, 2864, 2867, 2869
inbreeding coefficients matrix, output, 2865
individuals, outputting coefficients, 2864
individuals, specifying, 2862, 2866, 2867
initial covariance value, 2867
initial covariance value, assigning, 2865
initial covariance value, specifying, 2860
kinship coefficient, 2869
last generation's coefficients, output, 2864, 2865
mating, offspring and parent, 2873
mating, self, 2872
matings, output, 2867
monoecious population analysis, example, 2877
offspring, 2864, 2865, 2872
ordering observations, 2858

OUTCOV= data set, 2865, 2875
output table names, 2877
panels, 2875, 2882
pedigree analysis, 2857, 2858
pedigree analysis, example, 2880, 2882
population, monoecious, 2877
population, multiparous, 2864, 2865, 2868
population, nonoverlapping, 2866
population, overlapping, 2859, 2860, 2871
progeny, 2868, 2870, 2872, 2873, 2880
second parent, 2866, 2867
selective matings, output, 2867
specifying gender, 2862
theoretical correlation, 2869
unknown or missing parents, 2874
variables, unaddressed, 2867, 2868
incomplete block design
generating with PLAN procedure, 4752, 4754
incomplete principal components
REG procedure, 5456, 5477
incremental fit indices, 6761
indepdendent
random variables (Introduction to Modeling), 59
independence model chi-square, 6761
independence model chi-square degrees of freedom , 6761
independence sampler
Introduction to Bayesian Analysis, 153
Markov chain Monte Carlo, 153
independent variable
defined (ANOVA), 692
Introduction to Regression, 79
index counter
ODS Graphics, 543
indirect effect
TCALIS procedure, 6751
individual difference models
MDS procedure, 3704
INDSCAL model
MDS procedure, 3704, 3712
inertia, definition
CORRESP procedure, 1320
INEST= data sets
LIFEREG procedure, 3050
QUANTREG procedure, 5398
ROBUSTREG procedure, 5689
infeasibility
QUANTREG procedure, 5387
inference
design-based (Introduction to Modeling), 29
mixed model (MIXED), 3978
model-based (Introduction to Modeling), 29
space, mixed model (MIXED), 3917, 3918, 3920, 4018
infinite degrees of freedom
GLIMMIX procedure, 2126, 2136, 2146, 2157, 2167
HPMIXED procedure, 2819, 2821
infinite likelihood
MIXED procedure, 3954, 4011, 4012
infinite parameter estimates
LOGISTIC procedure, 3297, 3327
SURVEYLOGISTIC procedure, 6400, 6418
influence diagnostics
examples (REG), 5592
MIXED procedure, 3988
influence diagnostics, details
MIXED procedure, 3986
influence plots
MIXED procedure, 4006
influence statistics
REG procedure, 5559
information criteria
GLIMMIX procedure, 2099
HPMIXED procedure, 2811
MIXED procedure, 3907
information matrix, 5310
expected (GENMOD), 1969
LIFEREG procedure, 2992, 2993, 3037
observed (GENMOD), 1969
initial covariance value
assigning (INBREED), 2865
INBREED procedure, 2867
specifying (INBREED), 2860
initial estimates
ACECLUS procedure, 675
LIFEREG procedure, 3037
initial seed
SURVEYSELECT procedure, 6637
initial seeds
FASTCLUS procedure, 1622, 1623, 1639
initial values
CALIS procedure, 830, 853, 865, 881, 884, 900, 941
GENMOD procedure, 1949, 1959
GLIMMIX procedure, 2181
HPMIXED procedure, 2826
LOGISTIC procedure, 3365
MCMC procedure, 3479, 3500, 3515, 3527–3529
MDS procedure, 3715–3720, 3728
MIXED procedure, 3943
NLIN procedure, 4289
PHREG procedure, 4631, 4639
SURVEYLOGISTIC procedure, 6421
TCALIS procedure, 6728, 6742, 6922

initialization
random (PRINQUAL), 5233
TRANSREG procedure, 7252
input data set
MI procedure, 3752, 3759, 3791
input data sets
MIANALYZE procedure, 3851
input fixed-sample D
SEQDESIGN procedure, 5810
input fixed-sample N
SEQDESIGN procedure, 5810
input number of events for fixed-sample design
SEQDESIGN procedure, 5856
input sample size for fixed-sample design
SEQDESIGN procedure, 5856
inset
LIFEREG procedure, 3018
PROBIT procedure, 5277
insets
background color, 762, 765
background color of header, 762, 765
drop shadow color, 763
frame color, 763, 766
header text color, 763, 766
header text, specifying, 763, 766
positioning, details, 800–803, 805
positioning, options, 763, 766
suppressing frame, 763, 766
text color, 763, 766
instantaneous failure rate
PHREG procedure, 4579
integral approximation
theory (GLIMMIX), 2218
integral approximations
NLMIXED procedure, 4367, 4382
intensity model, *see* Andersen-Gill model
interaction effects
GLIMMIX procedure, 2261
MIXED procedure, 3983
model parameterization (GLM), 2492
Shared Concepts, 370
specifying (ANOVA), 720, 721
specifying (CATMOD), 1142
specifying (GLM), 2488
TRANSREG procedure, 7139, 7180, 7181
interactions, quantitative
TRANSREG procedure, 7181
intercept
GENMOD procedure, 1899, 1902, 1950
GLIMMIX procedure, 2261
hypothesis tests for (ANOVA), 715
hypothesis tests for (GLM), 2476
MIXED procedure, 3982
model parameterization (GLM), 2491
no intercept (TRANSREG), 7163
Shared Concepts, 369
internal studentization
MIXED procedure, 3987
interpretation
factor rotation, 1536
interpreting factors, elements to consider, 1537
interpreting output
VARCLUS procedure, 7479
interval determination
LIFETEST procedure, 3136
interval estimates
PHREG procedure, 4639
interval level of measurement
DISTANCE procedure, 1484
interval variable, 183
interval width
life-table method (LIFETEST), 3124
intervals
life-table estimates (LIFETEST), 3115
intraclass correlation coefficient
MIXED procedure, 4027
NLMIXED procedure, 4420
Introduction to ANOVA
SAS/STAT procedures, 115
Introduction to ANOVA Procedures
analysis of covariance, 117
analysis of ranks, 116, 124
analysis of variance, 117
balanced design, 122
Bayesian analysis, 118
categorical response, 118
classification effect, 116, 118, 119
constructing designs, 124
controlled experiment, 119
covariance structure modeling, 117
definition, 115
design matrix, 116
empirical Bayes predictions, 117
estimable function, 116, 120
exact test, 121
expected mean squares, 120, 121
experimental data, 116, 119
F-test based on sum of squares, 120
fixed effect, 121
general analysis of variance model, 116
general ANOVA procedures, 118
generalized estimating equations, 118
generalized linear mixed model, 118
generalized linear model, 118
group comparisons, 122
hat matrix, 119
hypothesis sum of squares, 120
lattice design, 117

least squares, 119
linear model, 116
mean squares, 120
mean squares, expected, 120, 121
method of moments, 117
model sum of squares, 119
multiple comparisons, 122
multivariate analysis of variance, 117
nested model, 117
nonlinear transformation, 117
nonparametric analysis, 116, 118, 124
observational data, 116, 119
p-value, 121
projection, 116
random effect, 121
repeated measures, 117
residual sum of squares, 120
Satterthwaite approximation, 121
spline transformation, 117
sum of squares decomposition, 116, 118
Type I sum of squares, 116
Type III sum of squares, 116
variance components, 117
Introduction to Bayesian Analysis, 141
adaptive algorithms, 155
advantages and disadvantages of Bayesian analysis, 149
assessing MCMC convergence, 156
Bayes' theorem, 142
Bayesian credible intervals, 149
Bayesian hypothesis testing, 148
Bayesian interval estimation, 149
Bayesian probability, 142
burn-in for MCMC, 155
deviance information criterion, 172
effective sample sizes (ESS), 169
equal-tail intervals, 149, 171
frequentist probability, 142
Gibbs sampler, 151, 154
highest posterior density (HPD) intervals, 149, 171
independence sampler, 153
Jeffreys' prior, 146
likelihood function, 143
likelihood principle, 150
marginal distribution, 143
Markov chain Monte Carlo, 151
Metropolis algorithm, 151
Metropolis-Hastings algorithm, 151
Monte Carlo standard error (MCSE), 148, 170
normalizing constant, 143
posterior distribution, 142
posterior summary statistics, 170
prior distribution, 142, 144
spectral density estimate at zero frequency, 164
thinning of MCMC, 155
Introduction to Mixed Modeling
assumptions, 129, 130
clustered data, 132
compound symmetry, 130
conditional distribution, 130, 132, 137
correlated error model, 127, 130
covariance matrix, 129
covariance parameters, 129
covariance structure, 132, 134
diagnostics, 128, 135
distribution, conditional, 130, 132, 137
distribution, marginal, 131, 132
fixed effect, 127
G matrix, 129, 131, 135
G-side random effect, 129–131
gauge R & R, 135
GEE, 131
generalized estimating equations, 131
generalized linear mixed model, 128, 130, 137
GENMOD v. GLIMMIX, 138
GLIMMIX v. GENMOD, 138
GLM v. MIXED, 135
GLMM, 130
groups, 134
heterocatanomic data, 137
hierarchical data, 132
HPMIXED v. MIXED, 136
lattice design, 128, 135
level-1 units, 133
level-2 units, 133
likelihood, residual, 130, 135
likelihood, restricted, 128, 130, 135
linear mixed model, 128, 129, 132, 135
link function, 130, 137
logit link, 131
marginal distribution, 131, 132
marginal model, 131
mean structure, 132
method of moments, 130, 135
mixed model smoothing, 137
mixed model, definition, 127
MIXED v. GLM, 135
MIXED v. HPMIXED, 136
monographs, 128
multiplicity adjustment, 137
nested model, 128, 135
nonlinear mixed model, 129, 132
parameter estimation, 130, 131
procedures, 128

R matrix, 129, 131, 132, 135
R-side random effect, 129, 132
random effect, 127
random effect, G-side, 129–131
random effect, R-side, 129, 132
residual likelihood, 130, 135
restricted likelihood, 128, 130, 135
smoothing, 137
sparse techniques, 128
splines, 137
subjects, 133
subjects, compared to groups, 134
variance components, 129
Introduction to Modeling
additive error, 31
analysis of variance, 34, 63, 64
augmented crossproduct matrix, 72
Bayesian models, 40
Cholesky decomposition, 55, 71
Cholesky residual, 71
classification effect, 33
coefficient of determination, 64
column space, 66
covariance, 57
covariance matrix, 58
crossproduct matrix, 72
curvilinear models, 32
deletion residual, 71
dependent variable, 31
deviance residual, 71
diagonal matrix, 50, 69
effect genesis, 36
estimable, 66
estimating equations, 31
expectation operator, 31
expected value, 57
expected value of vector, 58
exponential family, 38
externally studentized residual, 70
fitted residual, 69
fixed effect, 35
fixed-effects model, 35
g1-inverse, 52
g2-inverse, 52, 68, 72
generalized inverse, 52, 66, 72
generalized least squares, 44, 68
generalized linear model, 37, 68, 70, 71
hat matrix, 64, 69
heterocatanomic data, 35
heterogeneous multivariate data, 35
heteroscedasticity, 69
homocatanomic data, 35
homogeneous multivariate data, 35
homoscedasticity, 61
hypothesis testing, 66
idempotent matrix, 64
independent random variables, 59
independent variable, 31
inner product of vectors, 50
internally studentized residual, 70
inverse of matrix, 50
inverse of partitioned matrix, 51
inverse of patterned sum of matrices, 51
inverse, generalized, 52, 66, 72
iteratively reweighted least squares, 44
latent variable models, 38
LDU decomposition, 55
least squares, 43
leave-one-out residual, 71
levelization, 33
leverage, 69, 71
likelihood, 45
likelihood ratio test, 67
linear hypothesis, 66
linear inference, 66, 68
linear model theory, 61
linear regression, 33
link function, 37
LU decomposition, 55
matrix addition, 50
matrix decomposition, Cholesky, 55, 71
matrix decomposition, LDU, 55
matrix decomposition, LU, 55
matrix decomposition, singular-value, 56
matrix decomposition, spectral, 55
matrix decompositions, 55
matrix differentiation, 53
matrix dot product, 50
matrix inverse, 50
matrix inverse, g1, 52
matrix inverse, g2, 52, 68, 72
matrix inverse, Moore-Penrose, 52, 56, 57
matrix inverse, partitioned, 51
matrix inverse, patterned sum, 51
matrix inverse, reflexive, 52, 68, 72
matrix multiplication, 50
matrix order, 49
matrix partition, 72
matrix subtraction, 50
matrix transposition, 50
matrix, column space, 66
matrix, diagonal, 50, 69
matrix, idempotent, 64
matrix, projection, 64
matrix, rank deficient, 66
matrix, square, 50
matrix, sweeping, 71
mean function, 31

mean squared error, 60
model fitting, 29
model-based v. design-based, 29
Moore-Penrose inverse, 52, 56, 57
multivariate model, 34
nonlinear model, 31, 68
outcome variable, 31
parameter, 28
Pearson-type residual, 70
power, 49
PRESS statistic, 71
projected residual, 69
projection matrix, 64
pseudo-likelihood, 45
quadratic forms, 60
quasi-likelihood, 45
R-square, 64
random effect, 35
random-effects model, 35
rank deficient matrix, 66
raw residual, 69
reduction principle, testing, 66
reflexive inverse, 52, 68, 72
residual analysis, 69
residual, Cholesky, 71
residual, deletion, 71
residual, deviance, 71
residual, externally studentized, 70
residual, fitted, 69
residual, internally studentized, 70
residual, leave-one-out, 71
residual, Pearson-type, 70
residual, PRESS, 71
residual, projected, 69
residual, raw, 69
residual, scaled, 70
residual, standardized, 70
residual, studentized, 70
response variable, 31
sample size, 49
scaled residual, 70
singular-value decomposition, 56
spectral decomposition, 55
square matrix, 50
standardized residual, 70
statistical model, 28
stochastic model, 28
studentized residual, 70
sum of squares reduction test, 67, 68
sweep, elementary operations, 72
sweep, log determinant, 72
sweep, operator, 71
sweep, pivots, 72
testable hypothesis, 66, 68
testing hypotheses, 66
uncorrelated random variables, 59
univariate model, 34
variance, 57
variance matrix, 58
variance-covariance matrix, 59
weighted least squares, 44
Introduction to Regression
adj. R-square selection, 87
adjusted R-square, 98
assumptions, 87, 95
backward elimination, 86
Bayesian analysis, 89, 90
binary data, 89
breakdown value, 94
canonical correlation, 78, 94
coefficient of determination, 98, 110
collinearity, 109
collinearity diagnostics, 85
conditional logistic, 90
confidence interval, 101
conjoint analysis, 78, 94
contingency table, 76
controlled experiment, 108
correlation matrix, 97
covariance matrix, 97
Cox model, 77
Cp selection, 87
cross validation, 78
diagnostics, 77, 83
dichotomous response, 78, 89
errors-in-variable, 110
estimable, 102
estimate of precision, 97
exact conditional logistic, 90
exponential family, 88
extreme value regression, 90
failure-time data, 77
forecasting, 101
forward selection, 86
function approximation, 92
GEE, 76, 89, 90
general linear model, 77
generalized additive model, 76, 93
generalized estimating equations, 76, 89, 90
generalized least squares, 78
generalized linear mixed model, 77, 89
generalized linear model, 76, 88, 89, 93
generalized logit, 89, 90
Gentleman-Givens algorithm, 77, 90
gompit regression, 90
heterogeneous conditional distribution, 91
homoscedasticity, 87
Hotelling-Lawley trace, 104

Huber M estimation, 78, 94
hypothesis testing, 102
ideal point preference mapping, 78, 94
ill-conditioned data, 90
independent variable, 79
influence diagnostics, 85
interactive procedures, 87, 94
intercept, 98
inverse link function, 88
lack of fit, 85, 88
least trimmed squares, 94
levelization, 87
leverage, 83, 100
linear mixed model, 77
linear regression, 78
link function, 88
local regression, 77, 92, 93
LOESS, 77, 93
logistic regression, 76, 78, 89
LTS estimation, 94
M estimation, 78, 94
max R-square selection, 86
min R-square selection, 87
MM estimation, 94
model selection, 86
model selection, adj. R-square, 87
model selection, backward, 86
model selection, Cp, 87
model selection, forward, 86
model selection, max R-square, 86
model selection, min R-square, 87
model selection, R-square, 87
model selection, stepwise, 86
multivariate tests, 102, 104
nonlinear least squares, 77, 92
nonlinear mixed model, 77
nonlinear model, 77, 91
nonparametric, 76, 77, 92
normal equations, 96
observational study, 108
odds ratio, 89
orthogonal regressors, 109
outcome variable, 79
outlier detection, 78
partial least squares, 78, 88
penalized least squares, 93
Pillai's trace, 104
Poisson regression, 76
polychotomous response, 78, 89
polynomial model, 77
predicted value, 100
prediction interval, 101
predictor variable, 79
principal component regression, 78
probit regression, 78, 90
proportional hazard, 77
proportional odds model, 90
quantal response, 90
quantile regression, 78, 91
R-square, 98, 110
R-square selection, 87
R-square, adjusted, 98
raw residual, 83, 100
reduced rank regression, 78
redundancy analysis, 78, 94
regressor variable, 79
residual, 100
residual plot, 82
residual variance, 97
residual, raw, 100
residual, studentized, 100, 101
response surface regression, 78, 88
response variable, 79
ridge regression, 88
robust regression, 78, 93
Roy's maximum root, 104
S estimation, 94
semiparametric model, 93
spline basis function, 93
spline transformation, 78, 94
standard error of prediction, 100
standard error, estimated, 97
statistical graphics, 85
stepwise selection, 86
stratification, 90
studentized residual, 83, 100, 101
success probability, 89
survey data, 78, 90
survival analysis, 77
survival data, 77
time-series diagnostics, 85
transformation, 78, 94
Type I sum of squares, 97, 102
Type II sum of squares, 97, 102
variable selection, 86
variance inflation, 98
Wilk's Lambda, 103
Introduction to Survey Procedures
BRR variance estimation, 267
cluster sampling, 265
jackknife variance estimation, 267
multistage sampling, 266
population, 265
population totals, 266
primary sampling units (PSUs), 266
sample, 265
sample design, 264
sampling rates, 266

sampling units, 265
sampling weights, 266
stratified sampling, 265
survey data analysis, 259
survey sampling, 259
SURVEYFREQ procedure, 259, 263
SURVEYLOGISTIC procedure, 259, 264
SURVEYMEANS procedure, 259, 262, 269
SURVEYREG procedure, 259, 263, 269
SURVEYSELECT procedure, 259, 262, 268
Taylor series variance estimation, 267
variance estimation, 267
inverse chi-square distribution
definition of (MCMC), 3535
MCMC procedure, 3514, 3535
inverse confidence limits
PROBIT procedure, 5265, 5315
inverse Gaussian distribution
definition of (MCMC), 3541
GENMOD procedure, 1965
MCMC procedure, 3515, 3541
inverse gaussian distribution
GLIMMIX procedure, 2168
inverse-gamma distribution
definition of (MCMC), 3536
MCMC procedure, 3514, 3536
IPC analysis
REG procedure, 5456, 5477, 5583
ipp plots
annotating, 5282
axes, color, 5282
font, specifying, 5283
options summarized by function, 5280
reference lines, options, 5283–5286
threshold lines, options, 5285
ippplot
PROBIT procedure, 5279
isotropy, *see* anisotropy (VARIOGRAM)
VARIOGRAM procedure, 7522, 7537, 7547
iterated factor analysis, 1534
iteration details
GLIMMIX procedure, 2101
HPMIXED procedure, 2812
NLMIXED procedure, 4364
iteration history
GLIMMIX procedure, 2275
MIXED procedure, 3996
NLMIXED procedure, 4364, 4405
iterations
history (GLIMMIX), 2275
history (HPMIXED), 2838
history (MIXED), 3996
history (PHREG), 4568, 4633
PRINQUAL procedure, 5227
restarting (PRINQUAL), 5209, 5233
iterations, restarting
TRANSREG procedure, 7252
iterative proportional fitting
estimation (CATMOD), 1121
formulas (CATMOD), 1172

Jaccard dissimilarity coefficient
DISTANCE procedure, 1513
Jaccard similarity coefficient
DISTANCE procedure, 1513
jackknife
SURVEYLOGISTIC procedure, 6431
SURVEYMEANS procedure, 6506, 6508, 6528
SURVEYREG procedure, 6567
jackknife coefficients
SURVEYFREQ procedure, 6337
SURVEYLOGISTIC procedure, 6431, 6441
SURVEYMEANS procedure, 6508, 6510
SURVEYREG procedure, 6567, 6575
jackknife variance estimation
Introduction to Survey Procedures, 267
SURVEYFREQ procedure, 6337
SURVEYLOGISTIC procedure, 6431
SURVEYMEANS procedure, 6508
SURVEYREG procedure, 6567
Jacobian
NLIN procedure, 4279
Jeffreys' prior
definition of, 146
Introduction to Bayesian Analysis, 146
joint selection probabilities
SURVEYSELECT procedure, 6627
Jonckheere-Terpstra test, 1763
JOURNAL style
ODS Styles, 604
JOURNAL2 style
ODS Styles, 512

k-means clustering, 1622
k-sample tests, *see* homogeneity tests
k-th-nearest neighbor, *see also* density linkage, *see also* single linkage
k-th-nearest neighbor
estimation (CLUSTER), 1242, 1246
k-th-nearest-neighbor
estimation (CLUSTER), 1252
K= option
and other options (CLUSTER), 1242, 1246
Kaplan-Meier estimates, *see* product-limit estimates
kappa coefficient, 1766, 1767
tests, 1769

weights, 1768
Karush-Kuhn-Tucker (KKT) conditions
QUANTREG procedure, 5387
KDE procedure
bandwidth selection, 2903
binning, 2899
bivariate histogram, 2906
computational details, 2897
convolution, 2900
examples, 2907
fast Fourier transform, 2902
Introduction to Nonparametric Analysis, 297
ODS graph names, 2906
options, 2889
output table names, 2904
Kendall's tau-*b* statistic, 1736, 1738
Kenward-Roger method
GLIMMIX procedure, 2167
MIXED procedure, 3933
kernel density estimates
DISCRIM procedure, 1403, 1404, 1434, 1452
KDE procedure, 2885
keyword-lists
POWER procedure, 4919
Klotz scores
NPAR1WAY procedure, 4472
knot selection
GLIMMIX procedure, 2251
knots
PRINQUAL procedure, 5218, 5219
TRANSREG procedure, 7150, 7151, 7192, 7262
knots, exterior
TRANSREG procedure, 7208
Kolmogorov-Smirnov test
NPAR1WAY procedure, 4475
KRIGE2D procedure
anisotropic models, 2955–2957, 2959, 2962
azimuth, 2957
best linear unbiased prediction (BLUP), 2966
correlation range, 2932
discontinuity, 2954
effective range, 2932, 2950, 2951
estimation, 2931
examples, 2970, 2973
exponential semivariogram model, 2951
Gaussian semivariogram model, 2950
geometric anisotropy, 2956, 2957
global kriging, 2931
input data set, 2937
kriging with trend, 2964
local kriging, 2930, 2931
modeling, 2930
nested models, 2952–2954
nugget effect, 2946, 2954
ODS graph names, 2969
ODS Graphics, 2937
ODS table names, 2969
ordinary kriging, 2930, 2962–2966
OUTEST= data sets, 2967
OUTNBHD= data set, 2967, 2968
output data sets, 2937, 2967, 2968
power semivariogram model, 2952
practical range, 2932, 2950, 2951
prediction, 2930, 2931
sill, 2949
spatial continuity, 2930
spatial covariance, 2932
spatial data, 2962
spatial prediction, 2930
spatial random fields, 2963
spherical semivariogram model, 2949
standard errors, 2930
stochastic analysis, 2930
uncertainty, 2930
zonal anisotropy, 2959
KRIGE2D procedure, tables
Kriging Information, 2968, 2969
Model Information, 2969
Number of Observations, 2968
kriging
ordinary kriging (KRIGE2D), 2930, 2964
ordinary kriging (VARIOGRAM), 7519
with trend (KRIGE2D), 2964
Kronecker product structure
MIXED procedure, 3959
Kruskal-Wallis test
Introduction to Nonparametric Analysis, 296
NPAR1WAY procedure, 4470
Kuiper test
NPAR1WAY procedure, 4476
Kulcynski 1 coefficient
DISTANCE procedure, 1514
kurtosis
CALIS procedure, 828, 853, 857, 938, 940
displayed in CLUSTER procedure, 1246
TCALIS procedure, 6681, 6727, 6731, 6919, 6921

L matrices
GENMOD procedure, 1945
GLIMMIX procedure, 2123, 2141
HPMIXED procedure, 2816, 2821
mixed model (GLIMMIX), 2123, 2141
mixed model (MIXED), 3917, 3922, 3978
MIXED procedure, 3917, 3922, 3978

lack of fit
examples (REG), 5642
lack of fit tests, 5265, 5313
lack-of-fit
testing (REG), 5576
lack-of-fit tests
RSREG procedure, 5736
lag
classification (VARIOGRAM), 7522, 7551
count (VARIOGRAM), 7522, 7553
distance (VARIOGRAM), 7522, 7536, 7553, 7556
number of point pairs in (VARIOGRAM), 7555
pairwise distance (VARIOGRAM), 7522
tolerance (VARIOGRAM), 7536, 7552
VARIOGRAM procedure, 7522
lag functionality
GLIMMIX procedure, 411, 2208
NLMIXED procedure, 4380
Lagrange multiplier
covariance matrix, 4308
NLIN procedure, 4284
NLMIXED procedure, 4395
statistics (GENMOD), 1981
test statistics (LIFEREG), 3038
test, modification indices (CALIS), 860, 954
test, modification indices (TCALIS), 6734, 6918
lambda asymmetric, 1736, 1743
lambda symmetric, 1736, 1744
Lance-Williams flexible-beta method, *see* flexible-beta method
Lance-Williams nonmetric coefficient
DISTANCE procedure, 1510
Laplace approximation
GLIMMIX procedure, 2102
theory (GLIMMIX), 2224
Laplace distribution
definition of (MCMC), 3536
MCMC procedure, 3514, 3536
lasso selection
GLMSELECT procedure, 2725
latent variable models
Introduction to Modeling, 38
latent variable scores
CALIS procedure, 922, 937
latent variables
CALIS procedure, 828, 881, 908
PLS procedure, 4766
TCALIS procedure, 6681
latent vectors
PLS procedure, 4766
LaTeX destination
ODS Graphics, 541, 546, 604
Latin square design
ANOVA procedure, 739
generating with PLAN procedure, 4753
lattice design
balanced square lattice (LATTICE), 2979, 2980
efficiency (LATTICE), 2981, 2985, 2988
partially balanced square lattice (LATTICE), 2979, 2985
rectangular lattice (LATTICE), 2979
LATTICE procedure
adjusted treatment means, 2985
ANOVA table, 2984
Block variable, 2980, 2982, 2983
compared to MIXED procedure, 3895
covariance, 2985
Group variable, 2980, 2982, 2983
lattice design efficiency, 2981, 2985
least significant differences, 2985
missing values, 2984
ODS table names, 2985
Rep variable, 2980, 2982, 2983
response variable, 2982
Treatmnt variable, 2980, 2982, 2983
variance of means, 2984
LD statistic
PHREG procedure, 4574, 4614
leader algorithm, 1622
least angle regression
GLMSELECT procedure, 2725
least significant differences
LATTICE procedure, 2985
least squares
correlation matrix (Introduction to Regression), 97
covariance matrix (Introduction to Regression), 97
definition (Introduction to Modeling), 43
estimator (Introduction to Regression), 96
generalized (Introduction to Modeling), 44, 68
Introduction to ANOVA Procedures, 119
iteratively reweighted (Introduction to Modeling), 44
nonlinear (Introduction to Regression), 77, 92
normal equations (Introduction to Regression), 96
ordinary (Introduction to Regression), 88
partial (Introduction to Regression), 78, 88
penalized (Introduction to Regression), 93
weighted (Introduction to Modeling), 44
least squares means

Bonferroni adjustment (GLIMMIX), 2144
Bonferroni adjustment (GLM), 2459
Bonferroni adjustment (MIXED), 3924
BYLEVEL processing (GLIMMIX), 2146, 2149, 2157
BYLEVEL processing (MIXED), 3926
coefficient adjustment, 2530
compared to means (GLM), 2512
comparison types (GLIMMIX), 2147, 2153
comparison types (GLM), 2462
comparison types (MIXED), 3926
construction of, 2528
covariate values (GLIMMIX), 2145, 2157
covariate values (GLM), 2460
covariate values (MIXED), 3925
Dunnett's adjustment (GLIMMIX), 2144
Dunnett's adjustment (GLM), 2459
Dunnett's adjustment (MIXED), 3924
examples (GLM), 2567, 2580
examples (MIXED), 4032, 4056
generalized linear mixed model (GLIMMIX), 2141, 2155
GLIMMIX procedure, 2821
GLM procedure, 2458
Hsu's adjustment (GLIMMIX), 2144
Hsu's adjustment (GLM), 2459
Hsu's adjustment (MIXED), 3924
mixed model (MIXED), 3922
multiple comparison adjustment (GLIMMIX), 2143, 2144, 2156
multiple comparison adjustment (MIXED), 3923, 3924
multiple comparisons adjustment (GLM), 2459, 2462
Nelson's adjustment (GLIMMIX), 2144
Nelson's adjustment (GLM), 2459
nonstandard weights (GLM), 2461
nonstandard weights (MIXED), 3927
observed margins (GLIMMIX), 2148, 2159
observed margins (GLM), 2461
observed margins (MIXED), 3927
Scheffe's adjustment (GLIMMIX), 2144
Sidak's adjustment (GLIMMIX), 2144
Sidak's adjustment (GLM), 2459
Sidak's adjustment (MIXED), 3924
simple effects (GLIMMIX), 2152
simple effects (GLM), 2463, 2525
simple effects (MIXED), 3928
simple effects differences (GLIMMIX), 2152
simulation-based adjustment (GLIMMIX), 2144
simulation-based adjustment (GLM), 2459
simulation-based adjustment (MIXED), 3924
Tukey's adjustment (GLIMMIX), 2144
Tukey's adjustment (GLM), 2459
Tukey's adjustment (MIXED), 3924
least-significant-difference test, 714, 2473
least-squares estimation
LIFEREG procedure, 3037
Lee-Wei-Amato model
PHREG procedure, 4605, 4697
left truncation time
PHREG procedure, 4567, 4589
less-than-full-rank model
TRANSREG procedure, 7155, 7237
level of measurement
MDS procedure, 3705, 3716
levelization
Introduction to Regression, 87
Shared Concepts, 366
levels of measurement
DISTANCE procedure, 1484
levels, of classification variable, 2488
Levenberg-Marquardt algorithm
CALIS procedure, 856, 868, 945
TCALIS procedure, 6730, 6736, 6923
Levene's test for homogeneity of variance
ANOVA procedure, 713
GLM procedure, 2471, 2527, 2605
leverage, 2478
LOGISTIC procedure, 3349
MIXED Procedure, 3990
TRANSREG procedure, 7174
life data
GENMOD procedure, 2029
life-table estimates
LIFETEST procedure, 3100, 3151, 3179
lifereg analysis
insets, 3018
LIFEREG procedure, 2992
accelerated failure time models, 2992
censoring, 3020
computational details, 3037
computational resources, 3052
Confidence intervals, 3044
failure time, 2992
INEST= data sets, 3050
information matrix, 2992, 2993, 3037
initial estimates, 3037
inset, 3018
Lagrange multiplier test statistics, 3038
least-squares estimation, 3037
log-likelihood function, 2993, 3037
log-likelihood ratio tests, 2993
main effects, 3036

maximum likelihood estimates, 2992
missing values, 3036
Newton-Raphson algorithm, 2992
OUTEST= data sets, 3051
output data sets, 3056
output ODS Graphics table names, 3062
output table names, 3060
predicted values, 3043
supported distributions, 3039
survival function, 2993, 3039
Tobit model, 2994, 3069
XDATA= data sets, 3051
LIFETEST procedure
alpha level, 3113
association tests, 3101, 3108, 3145, 3158, 3168
Bonferroni adjustment, 3127
censored, 3151
computational formulas, 3131
confidence bands, 3114, 3137
confidence limits, 3136, 3148, 3149
cumulative distribution function, 3100
Dunnett's adjustment, 3127
effective sample size, 3134
equal-precision bands, 3139, 3175
estimation method, 3116
Fleming-Harrington G_ρ test for homogeneity, 3100, 3130
Hall-Wellner bands, 3138, 3175
hazard function, 3100, 3182
homogeneity tests, 3100, 3107, 3142, 3167
input data set, 3114
interval determination, 3136
life-table estimates, 3100, 3116, 3134, 3151, 3176, 3179
likelihood ratio test for homogeneity, 3100, 3142
line printer plots, 3111, 3116
log-rank test for association, 3101, 3145
log-rank test for homogeneity, 3100, 3130, 3142
maximum time in plots, 3116
median residual time, 3151
missing stratum values, 3117, 3125, 3129
missing values, 3131
modified Peto-Peto test for homogeneity, 3100, 3130
multiple-comparison adjustments, 3101
Nelson-Aalen estimates, 3117
ODS Graph names, 3156
ODS table names, 3155
output data sets, 3148
OUTSURV= data set, 3117
OUTTEST= data set, 3117
partial listing, 3124
Peto-Peto test for homogeneity, 3100, 3130
probability density function, 3100, 3183
product-limit curves, 3105
product-limit estimates, 3100, 3103, 3131, 3150, 3158
Scheffe's adjustment, 3127
Sidak's adjustment, 3127
simulated adjustment, 3127
smoothed hazard, 3101
stratified tests, 3100, 3101, 3108, 3110, 3129, 3143, 3153
studentized maximum modulus adjustment, 3127
survival distribution function, 3100, 3131
Tarone-Ware test for homogeneity, 3100, 3130
traditional graphics, 3111
transformations for confidence intervals, 3114
trend tests, 3100, 3129, 3145, 3153
Tukey's adjustment, 3127
Wilcoxon test for association, 3101, 3145
Wilcoxon test for homogeneity, 3100, 3130, 3142
LIFETEST procedure)
product-limit estimates, 3116
likelihood
function (Introduction to Modeling), 45
Introduction to Modeling, 45
likelihood displacement
PHREG procedure, 4574, 4614
likelihood distance
MIXED procedure, 3993
likelihood function, 5310
Introduction to Bayesian Analysis, 143
likelihood function specification
MCMC procedure, 3512
likelihood functions
SURVEYLOGISTIC procedure, 6423
likelihood principle
Introduction to Bayesian Analysis, 150
likelihood ratio chi-square test, 5311, 5313
Rao-Scott (SURVEYFREQ), 6350
likelihood ratio test, 4017
Bartlett's modification, 1395
CALIS procedure, 932, 954
example (MIXED), 4031
GLIMMIX procedure, 2127, 2234
Introduction to Modeling, 67
mixed model (MIXED), 3978, 3979
MIXED procedure, 3998
PHREG procedure, 4598, 4634
TCALIS procedure, 6904, 6919

likelihood ratio test for homogeneity
LIFETEST procedure, 3100
likelihood residuals
GENMOD procedure, 1984
likelihood-ratio chi-square test, 1731
power and sample size (POWER), 4884, 4890, 4968
likelihood-ratio test
chi-square (FREQ), 1733
line printer plots
LIFETEST procedure, 3111, 3116
REG procedure, 5497, 5534
line printer plots (LIFETEST), 3111
characters used, 3115
no censor plot, 3117
line search
PHREG procedure, 4569
line-search methods
GLIMMIX procedure, 398
NLMIXED procedure, 4364, 4365, 4397
linear covariance structure
GLIMMIX procedure, 2197
MIXED procedure, 3959
linear discriminant function, 1384
linear equations model, *see* LINEQS model
linear hypotheses
PHREG procedure, 4524, 4577, 4600
linear hypothesis
consistency (Introduction to Modeling), 66
definition (Introduction to Modeling), 66
Introduction to Modeling, 66
linear inference principle (Introduction to Modeling), 66, 68
reduction principle (Introduction to Modeling), 66
testable (Introduction to Modeling), 66, 68
testing (Introduction to Modeling), 66
testing, linear inference (Introduction to Modeling), 66, 68
testing, reduction principle (Introduction to Modeling), 66
linear mixed model
Introduction to Regression, 77
linear model
GENMOD procedure, 1896, 1897
linear model theory
Introduction to Modeling, 61
linear models
CATMOD procedure, 1093
compared with log-linear models, 1097
linear predictor
GENMOD procedure, 1894, 1896, 1902, 1975, 2011
PHREG procedure, 4544, 4573, 4574, 4685, 4686
linear rank tests, *see* association tests
linear regression
Introduction to Modeling, 33
TRANSREG procedure, 7179
linear structural relationship model, *see* LISREL model
linear transformation
baseline confidence intervals (PHREG), 4545
confidence intervals (LIFETEST), 3114, 3137
PRINQUAL procedure, 5215
TRANSREG procedure, 7145, 7260
linearization
theory (GLIMMIX), 2218, 2220
linearization method
SURVEYLOGISTIC procedure, 6427
SURVEYREG procedure, 6564
LINEQS model
CALIS procedure, 833, 881
specification, 840
structural model example (CALIS), 837, 841
structural model example (TCALIS), 6708
TCALIS procedure, 6766
link function
built-in (GENMOD), 1898, 1949
cumulative (Introduction to Regression), 89
GENMOD procedure, 1894, 1896, 1967
GLIMMIX procedure, 2083, 2171
Introduction to Mixed Modeling, 130, 137
Introduction to Modeling, 37
Introduction to Regression, 88
inverse (Introduction to Mixed Modeling), 130
inverse (Introduction to Regression), 88
LOGISTIC procedure, 3257, 3296, 3322, 3332
logit (Introduction to Mixed Modeling), 131
user-defined (GENMOD), 1944
user-defined (GLIMMIX), 2208
link functions
SURVEYLOGISTIC procedure, 6375, 6399, 6415
links, HTML
ODS, 477, 482
LISMOD
structural model example (TCALIS), 6709
LISMOD model
TCALIS procedure, 6772
LISREL model
CALIS procedure, 833
structural model example (CALIS), 838

LISTING destination
ODS Graphics, 545
LISTING style
ODS Styles, 518
LM tests
TCALIS procedure, 6776
LMAX statistic
PHREG procedure, 4614
LMSELECT procedure
ODS Graphics, 2698, 5456
local and remote configurations, 5085
local influence
DFBETA statistics (PHREG), 4573, 4614
score residuals (PHREG), 4574, 4612
weighted score residuals (PHREG), 4614
local kriging
KRIGE2D procedure, 2931
LOESS
Introduction to Regression, 77, 93
LOESS procedure
approximate degrees of freedom, 3224
automatic smoothing parameter selection, 3221
data scaling, 3216
degrees of freedom, 3220
direct fitting method, 3217
introductory example, 3189
iterative reweighting, 3219
kd trees and blending, 3218
local polynomials, 3219
local weighting, 3218
lookup degrees of freedom, 3220
missing values, 3215
ODS graph names, 3226
ODS Graphics, 3202, 3226
output data sets, 3215
output table names, 3225
scoring data sets, 3225
smoothing matrix, 3220
statistical graphics, 3226
statistical inference, 3220
log likelihood
output data sets (LOGISTIC), 3272
log odds
LOGISTIC procedure, 3335
SURVEYLOGISTIC procedure, 6435
log odds-ratio statistic
SEQDESIGN procedure, 5864
log relative risk statistic
SEQDESIGN procedure, 5865
log transformation
baseline confidence intervals (PHREG), 4544
confidence intervals (LIFETEST), 3114, 3137, 3139
log-hazard
PHREG procedure, 4591
log-interval level of measurement
DISTANCE procedure, 1484
log-likelihood
functions (GENMOD), 1967
log-likelihood function
LIFEREG procedure, 2993, 3037
PROBIT procedure, 5310
log-likelihood ratio tests
LIFEREG procedure, 2993
log-linear models
CATMOD procedure, 1094, 1148, 1901
compared with linear models, 1097
design matrix (CATMOD), 1160
examples (CATMOD), 1193, 1195
GENMOD procedure, 1901
multiple populations (CATMOD), 1149
one population (CATMOD), 1148
log-linear variance model
MIXED procedure, 3956
log-log transformation
baseline confidence intervals (PHREG), 4544
confidence intervals (LIFETEST), 3114, 3137, 3139
log-logistic model
NLIN procedure, 4324
log-normal distribution
GLIMMIX procedure, 2168
log-rank test
PHREG procedure, 4529
SEQDESIGN procedure, 5866
log-rank test for association
LIFETEST procedure, 3101
log-rank test for homogeneity
LIFETEST procedure, 3100, 3130, 3142
power and sample size (POWER), 4901, 4909, 4975, 5007
log-rank test for two survival distributions
SEQDESIGN procedure, 5814
logden distribution
MCMC procedure, 3514
logistic
diagnostics (Introduction to Regression), 77
regression (Introduction to Regression), 76, 78, 89
regression, diagnostics (Introduction to Regression), 77
regression, ordinal (Introduction to Regression), 78

regression, survey data (Introduction to Regression), 78, 90
logistic analysis
CATMOD procedure, 1095, 1146
caution (CATMOD), 1147
examples (CATMOD), 1208
ordinal data, 1095
logistic distribution, 2992, 3022, 3039, 5258
definition of (MCMC), 3537
MCMC procedure, 3514, 3537
PROBIT procedure, 5311
LOGISTIC procedure
Akaike's information criterion, 3329
Bayes' theorem, 3298
best subset selection, 3292
branch-and-bound algorithm, 3329
classification table, 3298, 3340, 3341, 3419
conditional logistic regression, 3314, 3355
confidence intervals, 3299, 3302, 3333, 3335
confidence limits, 3339
convergence criterion, 3291
customized odds ratio, 3316
descriptive statistics, 3275
deviance, 3292, 3300, 3343
DFBETAS diagnostic, 3350
dispersion parameter, 3343
displayed output, 3371
estimability checking, 3283
exact logistic regression, 3285, 3359
existence of MLEs, 3327
Firth's penalized likelihood, 3326
Fisher scoring algorithm, 3301, 3302, 3324
frequency variable, 3287
goodness of fit, 3292, 3300
gradient, 3331
hat matrix, 3349
Hessian matrix, 3301, 3331
hierarchy, 3294
Hosmer-Lemeshow test, 3296, 3344, 3345
infinite parameter estimates, 3297
initial values, 3365
introductory example, 3260
leverage, 3349
link function, 3257, 3296, 3322, 3332
log odds, 3335
maximum likelihood algorithms, 3324
missing values, 3318
model fitting criteria, 3329
model hierarchy, 3258, 3294
model selection, 3290, 3301, 3328
multiple classifications, 3299
Newton-Raphson algorithm, 3301, 3302, 3324, 3326
odds ratio confidence limits, 3292, 3299
odds ratio estimation, 3335
odds ratios with interactions, 3303
ODS graph names, 3379
ODS table names, 3377
output data sets, 3272, 3364, 3366–3368
overdispersion, 3300, 3342, 3343
Pearson's chi-square, 3292, 3300, 3343
predicted probabilities, 3339
prior event probability, 3298, 3342, 3419
profile-likelihood convergence criterion, 3299
rank correlation, 3338
regression diagnostics, 3349
residuals, 3350
response level ordering, 3268, 3288, 3318
ROC curve, 3298, 3310, 3346, 3368
ROC curve, comparing, 3311, 3347
Schwarz criterion, 3329
score statistics, 3331
scoring data sets, 3312, 3352
selection methods, 3290, 3301, 3328
singular contrast matrix, 3283
subpopulation, 3292, 3300, 3343
testing linear hypotheses, 3315, 3348
Williams' method, 3343
logistic regression, *see also* LOGISTIC procedure, 5049, 5258, *see also* SURVEYLOGISTIC procedure
CATMOD procedure, 1094, 1146
examples (CATMOD), 1188
GENMOD procedure, 1897, 2024
Introduction to Regression, 76, 89
power and sample size (POWER), 4827, 4928, 5037
logistic regression method
MI procedure, 3779
logit transformation
confidence intervals (LIFETEST), 3114, 3137, 3139
logits, *see also* cumulative logits, *see also* generalized logits, *see* adjacent-category logits
loglogistic distribution, 2992, 3022, 3039
lognormal data
power and sample size (POWER), 4854, 4858, 4873, 4878, 4879, 4893, 4899, 4900, 4954, 4955, 4963, 4964, 4972, 4974, 4999
lognormal distribution, 2992, 3022, 3039
definition of (MCMC), 3537
MCMC procedure, 3514, 3537
TTEST procedure, 7402, 7452
long run times

MCMC procedure, 3566
NLMIXED procedure, 4399
Longley data set, 4506
lower one-sided *t* test
TTEST procedure, 7407
L_p clustering
FASTCLUS procedure, 1622
L_p clustering
FASTCLUS procedure, 1636
lpred plots
annotating, 5290
axes, color, 5290
font, specifying, 5291
reference lines, options, 5291–5294
threshold lines, options, 5294
lpredplot
PROBIT procedure, 5287
LR ordering
SEQTEST procedure, 6026
LR statistics
MI procedure, 3788
LS-means, *see* least squares means
lsd (least significant differences)
LATTICE procedure, 2985
LSD test, 2473

MAC method
PRINQUAL procedure, 5221, 5229
macro variables
GLMSELECT procedure, 2734
macros
TRANSREG procedure, 7174
Mahalanobis distance, 1070, 1392, 1403
CANDISC procedure, 1084
main effects
design matrix (CATMOD), 1154
GENMOD procedure, 1974
LIFEREG procedure, 3036
MIXED procedure, 3982
model parameterization (GLM), 2491
Shared Concepts, 369
specifying (ANOVA), 720, 721
specifying (CATMOD), 1142
specifying (GLM), 2488
TRANSREG procedure, 7139, 7180, 7181
Mallows' C_p selection
REG procedure, 5526
manifest variables
CALIS procedure, 828
TCALIS procedure, 6681
Mann-Whitney-Wilcoxon test, *see* Wilcoxon-Mann-Whitney (rank-sum) test
NPAR1WAY procedure, 4470
MANOVA, *see* multivariate analysis of variance, *see* multivariate analysis of variance
CANDISC procedure, 1065
Mantel-Haenszel chi-square test, 1731, 1734
Mantel-Haenszel test
log-rank test (PHREG), 4529
MAR
MI procedure, 3771, 3798
marginal distribution
definition of, 143
Introduction to Bayesian Analysis, 143
MCMC procedure, 3561
marginal probabilities, *see also* response functions
specifying in CATMOD procedure, 1131
marginal residuals
MIXED procedure, 3987
Markov chain Monte Carlo
adaptive algorithms, 155
assessing MCMC convergence, 156
burn-in for MCMC, 155
Gibbs sampler, 151, 154
independence sampler, 153
Introduction to Bayesian Analysis, 151
Metropolis algorithm, 151, 152
Metropolis-Hastings algorithm, 151, 152
posterior summary statistics, 170
thinning of MCMC, 155
martingale residuals
PHREG procedure, 4574, 4685
Matérn covariance structure
GLIMMIX procedure, 2201
MIXED procedure, 3959
matched comparisons, *see* paired comparisons
mating
offspring and parent (INBREED), 2873
self (INBREED), 2872
matrix
addition (Introduction to Modeling), 50
Choleksy decomposition (Introduction to Modeling), 55, 71
column space (Introduction to Modeling), 66
correlation (Introduction to Regression), 97
covariance (Introduction to Mixed Modeling), 129
covariance (Introduction to Regression), 97
crossproduct (Introduction to Modeling), 72
crossproduct, augmented (Introduction to Modeling), 72
decomposition, Cholesky (Introduction to Modeling), 55, 71
decomposition, LDU (Introduction to Modeling), 55

decomposition, LU (Introduction to Modeling), 55
decomposition, singular-value (Introduction to Modeling), 56
decomposition, spectral (Introduction to Modeling), 55
decompositions (CORRESP), 1330, 1353
decompositions (Introduction to Modeling), 55
design (Introduction to ANOVA Procedures), 116
determinant, by sweeping (Introduction to Modeling), 72
diagonal (Introduction to Modeling), 50, 69
diagonal (Introduction to Regression), 96
differentiation (Introduction to Modeling), 53
dot product (Introduction to Modeling), 50
factor, defined for factor analysis (FACTOR), 1535
g1-inverse (Introduction to Modeling), 52
g2-inverse (Introduction to Modeling), 52, 68, 72
generalized inverse (Introduction to Modeling), 52, 66, 72
hat (Introduction to ANOVA Procedures), 119
hat (Introduction to Modeling), 64, 69
idempotent (Introduction to Modeling), 64
inner product (Introduction to Modeling), 50
inverse (Introduction to Modeling), 50
inverse, g1 (Introduction to Modeling), 52
inverse, g2 (Introduction to Modeling), 52, 68, 72
inverse, generalized (Introduction to Modeling), 52, 66, 72
inverse, Moore-Penrose (Introduction to Modeling), 52, 56, 57
inverse, partitioned (Introduction to Modeling), 51
inverse, patterned (Introduction to Modeling), 51
inverse, reflexive (Introduction to Modeling), 52, 68, 72
inversion (CALIS), 927
inversion (TCALIS), 6897
LDU decomposition (Introduction to Modeling), 55
leverage (Introduction to Modeling), 69
LU decomposition (Introduction to Modeling), 55
Moore-Penrose inverse (Introduction to Modeling), 52, 56, 57
multiplication (Introduction to Modeling), 50
multiplication (SCORE), 5759
names, default (CALIS), 879
notation, theory (MIXED), 3968
order (Introduction to Modeling), 49
partition (Introduction to Modeling), 72
projection (Introduction to ANOVA Procedures), 116
projection (Introduction to Modeling), 64
properties, COSAN model (CALIS), 873
rank deficient (Introduction to Modeling), 66
reflexive inverse (Introduction to Modeling), 52, 68, 72
singular-value decomposition (Introduction to Modeling), 56
spectral decomposition (Introduction to Modeling), 55
square (Introduction to Modeling), 50
subtraction (Introduction to Modeling), 50
sweep (Introduction to Modeling), 71
transposition (Introduction to Modeling), 50
matrix properties
COSAN model (CALIS), 874
MATRIX statement (CALIS)
factor analysis model, 885
Maximum a posteriori
MCMC procedure, 3527
maximum average correlation method
PRINQUAL procedure, 5221, 5229
maximum information
SEQDESIGN procedure, 5822, 5826, 5855, 5873
maximum likelihood
algorithms (LOGISTIC), 3324
algorithms (SURVEYLOGISTIC), 6417
estimates (LIFEREG), 2992
estimates (LOGISTIC), 3327
estimates (SURVEYLOGISTIC), 6418
estimation (CATMOD), 1096, 1121, 1171
estimation (GENMOD), 1968
GLIMMIX procedure, 2102, 2273
hierarchical clustering (CLUSTER), 1239, 1244, 1254, 1255
NLMIXED procedure, 4345
VARCOMP procedure, 7498
maximum likelihood estimate
SEQDESIGN procedure, 5818
maximum likelihood estimate scale
SEQDESIGN procedure, 5832
maximum likelihood estimates
PHREG procedure, 4638
maximum likelihood estimation
mixed model (MIXED), 3975

maximum likelihood factor analysis, 1534, 1552
with FACTOR procedure, 1539, 1540
maximum method, *see* complete linkage
maximum redundancy analysis
TRANSREG procedure, 7162
maximum time
confidence bands (LIFETEST), 3113
maximum time in plots
LIFETEST procedure, 3116
maximum total variance method
PRINQUAL procedure, 5220
MBN adjusted sandwich estimators
GLIMMIX procedure, 2244
MCAR
MI procedure, 3771
MCF, *see* mean function
MCMC method
MI procedure, 3780
MCMC monotone-data imputation
MI procedure, 3799
MCMC procedure, 3478
arrays, 3508
Behrens-Fisher problem, 3488
Bernoulli distribution, 3512, 3531
beta distribution, 3512, 3530
binary distribution, 3512, 3531
binomial distribution, 3512, 3531
blocking, 3523
Box-Cox transformation, 3583
Cauchy distribution, 3512, 3531
censoring, 3544, 3664
chi-square distribution, 3512, 3532
compared with other SAS procedures, 3479
computational resources, 3570
constants specification, 3509
convergence, 3567
Cox models, 3647
deviance information criterion, 3645
dgeneral distribution, 3513, 3541
dlogden distribution, 3513
double exponential distribution, 3514, 3536
examples, *see also* examples, MCMC, 3578
exponential chi-square distribution, 3513, 3532
exponential distribution, 3513, 3534
exponential exponential distribution, 3513, 3532
exponential gamma distribution, 3513, 3533
exponential inverse chi-square distribution, 3513, 3533
exponential inverse-gamma distribution, 3513, 3533
exponential scaled inverse chi-square distribution, 3513, 3534
floating point errors, 3565
gamma distribution, 3513, 3534
Gaussian distribution, 3514, 3538
Gelman-Rubin diagnostics, 3693
general distribution, 3514, 3541
generalized linear models, 3592
geometric distribution, 3513, 3534
handling error messages, 3568
hierarchical centering, 3615
hyperprior distribution, 3511, 3516
initial values, 3479, 3500, 3515, 3527–3529
inverse chi-square distribution, 3514, 3535
inverse Gaussian distribution, 3515, 3541
inverse-gamma distribution, 3514, 3536
Laplace distribution, 3514, 3536
likelihood function specification, 3512
logden distribution, 3514
logistic distribution, 3514, 3537
lognormal distribution, 3514, 3537
long run times, 3566
marginal distribution, 3561
Maximum a posteriori, 3527
mixed-effects models, 3614
mixing, 3605, 3683
model specification, 3512
modeling dependent data, 3648
negative binomial distribution, 3514, 3538
nonlinear Poisson regression, 3605
normal distribution, 3514, 3538
options, 3497
options summary, 3496
output ODS Graphics table names, 3577
output table names, 3575
overflows, 3565
parameters specification, 3515
pareto distribution, 3514, 3539
Poisson distribution, 3514, 3539
posterior predictive distribution, 3560
posterior samples data set, 3502
precision of solution, 3568
prior distribution, 3511, 3516
prior predictive distribution, 3560
programming statements, 3516
proposal distribution, 3525
random-effects models, 3614
run times, 3566, 3570
scaled inverse chi-square distribution, 3515, 3539
specifying a new distribution, 3541
standard distributions, 3530
survival analysis, 3634
syntax summary, 3495
t distribution, 3515, 3539
truncated distributions, 3544

tuning, 3525
UDS statement, 3518
uniform distribution, 3515, 3540
user defined sampler statement, 3518
user-defined distribution, 3514
user-defined samplers, 3520, 3672
using the IF-ELSE logical control, 3583
Wald distribution, 3515, 3541
Weibull distribution, 3515, 3541
McNemar's test, 1765
Introduction to Nonparametric Analysis, 295
power and sample size (POWER), 4864, 4869, 4871, 4960
McQuitty's similarity analysis
CLUSTER procedure, 1240
MDFFITS
MIXED procedure, 3991
MDFFITS for covariance parameters
MIXED procedure, 3992
MDPREF analysis
PRINQUAL procedure, 5238
MDS procedure
alternating least squares, 3712
analyzing data in groups, 3721
asymmetric data, 3720
badness of fit, 3715, 3717, 3718, 3725, 3726
conditional data, 3713
configuration, 3704, 3717, 3718, 3725
convergence criterion, 3713, 3715, 3716
coordinates, 3717, 3718, 3724, 3725
data weights, 3723
dimension coefficients, 3704, 3705, 3712, 3717, 3718, 3724, 3725
dissimilarity data, 3704, 3713, 3720
distance data, 3704, 3713, 3720
Euclidean distances, 3705, 3712, 3724
external unfolding, 3704
individual difference models, 3704
INDSCAL model, 3704, 3712
initial values, 3715–3720, 3728
measurement level, 3705, 3716
metric multidimensional scaling, 3704
missing values, 3728
multidimensional scaling, 3704
nonmetric multidimensional scaling, 3704
normalization of the estimates, 3728
ODS Graph names, 3731
optimal transformations, 3704, 3705, 3716
output table names, 3731
partitions, 3713, 3724
plot of configuration, 3740
plot of dimension coefficients, 3740
plot of linear fit, 3740
proximity data, 3704, 3713, 3720
residuals, 3718, 3723, 3724, 3727, 3740
similarity data, 3704, 3713, 3720
stress formula, 3715, 3725
subject weights, 3704, 3712
three-way multidimensional scaling, 3704
ties, 3721
transformations, 3704, 3705, 3716, 3717, 3720, 3723–3725
transformed data, 3727
transformed distances, 3727
unfolding, 3704
weighted Euclidean distance, 3705, 3712, 3724
weighted Euclidean model, 3704, 3712
weighted least squares, 3723
mean function
linear (Introduction to Modeling), 31
nonlinear (Introduction to Modeling), 31
PHREG procedure, 4542, 4545, 4594, 4595, 4606, 4608
mean per element
SURVEYMEANS procedure, 6495
mean separation tests, *see* multiple-comparison procedures
mean squared error
Introduction to Modeling, 60
mean survival time
time limit (LIFETEST), 3123
MEAN= data sets
FASTCLUS procedure, 1638
means
ANOVA procedure, 709
compared to least squares means (GLM), 2512
displayed in CLUSTER procedure, 1246
GLM procedure, 2468
power and sample size (POWER), 4852, 4871, 4891, 4898, 4969
SURVEYMEANS procedure, 6495
weighted (GLM), 2527
MEANS procedure, 24
means, difference between
independent samples, 7392, 7431
paired observations, 7392
means, ratio of
independent samples, 7392
paired samples, 7392
measurement level
MDS procedure, 3705, 3716
measures of agreement, 1765
measures of spatial continuity
VARIOGRAM procedure, 7518, 7544, 7596
median
cluster, 1634, 1637

method (CLUSTER), 1240, 1255
median residual time
LIFETEST procedure, 3151
median scores
NPAR1WAY procedure, 4470
median unbiased estimate
SEQTEST procedure, 6024
Medical Expenditure Panel Survey (MEPS)
SURVEYLOGISTIC procedure, 6455
Mehta and Patel
network algorithm (NPAR1WAY), 4477
Mehta and Patel, network algorithm, 1780
memory requirements
ACECLUS procedure, 681
CLUSTER procedure, 1260
FACTOR procedure, 1578
FASTCLUS procedure, 1647
MIXED procedure, 4013
reduction of (ANOVA), 703
reduction of (GLM), 2452
VARCLUS procedure, 7479
memory usage
SIM2D procedure, 6180
Merle-Spath algorithm
FASTCLUS procedure, 1637
method information
SEQDESIGN procedure, 5877
METHOD= specification
PROC CLUSTER statement, 1239
methods of estimation
VARCOMP procedure, 7494, 7508
metric conjoint analysis
TRANSREG procedure, 7333
metric multidimensional scaling
MDS procedure, 3704
Metropolis algorithm
Introduction to Bayesian Analysis, 151
Markov chain Monte Carlo, 151, 152
Metropolis-Hastings algorithm
Introduction to Bayesian Analysis, 151
Markov chain Monte Carlo, 151, 152
MGV method
PRINQUAL procedure, 5221
MI procedure
adjusted degrees of freedom, 3795
analyst's model, 3796
approximate Bayesian bootstrap, 3776
arbitrary missing pattern, 3773
autocorrelation function plot, 3790
Bayes' theorem, 3780
Bayesian inference, 3780
between-imputation variance, 3794
bootstrap, 3760
combining inferences, 3794
converge in EM algorithm, 3755
convergence in EM algorithm, 3760
convergence in MCMC, 3788, 3799
degrees of freedom, 3794
discriminant function method, 3777
EM algorithm, 3769, 3799
fraction of missing information, 3795
imputation methods, 3773
imputation model, 3798
imputer's model, 3796
input data set, 3752, 3759, 3791
introductory example, 3746
logistic regression method, 3779
LR statistics, 3788
MAR, 3771, 3798
MCAR, 3771
MCMC method, 3780
MCMC monotone-data imputation, 3799
missing at random, 3771, 3798
monotone missing pattern, 3745, 3772
multiple imputation efficiency, 3795
multivariate normality assumption, 3798
number of imputations, 3798
ODS graph names, 3800
ODS table names, 3800
output data sets, 3753, 3760, 3792
output parameter estimates, 3760
parameter simulation, 3797
predictive mean matching method, 3775
producing monotone missingness, 3785
propensity score method, 3776, 3799
random number generators, 3753
regression method, 3774, 3798
relative efficiency, 3795
relative increase in variance, 3795
saving graphics output, 3759
singularity, 3754
Summary of Issues in Multiple Imputation, 3798
suppressing output, 3753
syntax, 3750
total variance, 3794
trace plot, 3789
transformation, 3767
within-imputation variance, 3794
worst linear function of parameters, 3789
MI procedure, EM statement
output data sets, 3755
MIANALYZE procedure
adjusted degrees of freedom, 3856
average relative increase in variance, 3858
between-imputation covariance matrix, 3858
between-imputation variance, 3856
combining inferences, 3855

degrees of freedom, 3856, 3858
fraction of missing information, 3856
input data sets, 3851
introductory example, 3841
multiple imputation efficiency, 3857
multivariate inferences, 3857
ODS table names, 3861
relative efficiency, 3857
relative increase in variance, 3856
syntax, 3844
testing linear hypotheses, 3849, 3859
total covariance matrix, 3858
total variance, 3856
within-imputation covariance matrix, 3857
within-imputation variance, 3856
minimum error spending
SEQTEST procedure, 6007, 6019
minimum generalized variance method
PRINQUAL procedure, 5221
minimum time
confidence bands (LIFETEST), 3113
Minkowski L_p distance coefficient
DISTANCE procedure, 1509
Minkowski metric
STDIZE procedure, 6231
misclassification probabilities
discriminant analysis, 1385
missing at random
MI procedure, 3771, 3798
missing level combinations
GLIMMIX procedure, 2262
MIXED procedure, 3986
missing stratum values
LIFETEST procedure, 3117, 3125, 3129
missing values
ACECLUS procedure, 679
and interactivity (GLM), 2491
CANCORR procedure, 1044
character (PRINQUAL), 5215
CLUSTER procedure, 1261
DISTANCE procedure, 1497, 1499, 1503, 1514
FASTCLUS procedure, 1623, 1636, 1638, 1642
LIFEREG procedure, 3036
LIFETEST procedure, 3131
LOGISTIC procedure, 3318
MDS procedure, 3728
MODECLUS procedure, 4121
MULTTEST procedure, 4219
NPAR1WAY procedure, 4466
PHREG procedure, 4564, 4576, 4669
PRINCOMP procedure, 5158
PRINQUAL procedure, 5206, 5227, 5234
PROBIT procedure, 5309
SCORE procedure, 5769
STDIZE procedure, 6223, 6224
strata variables (PHREG), 4577
SURVEYFREQ procedure, 6323
SURVEYLOGISTIC procedure, 6383, 6410
SURVEYMEANS procedure, 6474, 6491, 6526
SURVEYREG procedure, 6543, 6560
SURVEYSELECT procedure, 6644
TRANSREG procedure, 7164, 7249, 7250, 7255
TREE procedure, 7369
VARCOMP procedure, 7500
MIVQUE0 estimation
GLIMMIX procedure, 2184, 2275
mixed model
assumptions (Introduction to Mixed Modeling), 129, 130
clustered data (Introduction to Mixed Modeling), 132
compound symmetry (Introduction to Mixed Modeling), 130
conditional distribution (Introduction to Mixed Modeling), 130, 132, 137
covariance matrix (Introduction to Mixed Modeling), 129
covariance parameters (Introduction to Mixed Modeling), 129
covariance structure (Introduction to Mixed Modeling), 132, 134
definition (Introduction to Mixed Modeling), 127
diagnostics (Introduction to Mixed Modeling), 128, 135
distribution, conditional (Introduction to Mixed Modeling), 130, 132, 137
distribution, marginal (Introduction to Mixed Modeling), 131, 132
fixed effect (Introduction to Mixed Modeling), 127
G matrix (Introduction to Mixed Modeling), 129, 131, 135
G-side random effect (Introduction to Mixed Modeling), 129–131
gauge R & R study (Introduction to Mixed Modeling), 135
GEE (Introduction to Mixed Modeling), 131
generalized estimating equations (Introduction to Mixed Modeling), 131
generalized linear (Introduction to ANOVA Procedures), 118
generalized linear (Introduction to Mixed Modeling), 128, 130, 137

GENMOD and GLIMMIX compared (Introduction to Mixed Modeling), 138
GLIMMIX and GENMOD compared (Introduction to Mixed Modeling), 138
GLM and MIXED compared (Introduction to Mixed Modeling), 135
GLMM (Introduction to Mixed Modeling), 130
groups (Introduction to Mixed Modeling), 134
hierarchical data (Introduction to Mixed Modeling), 132
HPMIXED and MIXED compared (Introduction to Mixed Modeling), 136
HPMIXED procedure, 2822
lattice design (Introduction to Mixed Modeling), 128, 135
level-1 units (Introduction to Mixed Modeling), 133
level-2 units (Introduction to Mixed Modeling), 133
likelihood, residual (Introduction to Mixed Modeling), 130, 135
likelihood, restricted (Introduction to Mixed Modeling), 128, 130, 135
linear (Introduction to ANOVA Procedures), 117
linear (Introduction to Mixed Modeling), 128, 129, 132, 135
link function (Introduction to Mixed Modeling), 130, 137
logit link (Introduction to Mixed Modeling), 131
marginal distribution (Introduction to Mixed Modeling), 131, 132
marginal model (Introduction to Mixed Modeling), 131
mean structure (Introduction to Mixed Modeling), 132
method of moments (Introduction to Mixed Modeling), 130, 135
MIXED and GLM compared (Introduction to Mixed Modeling), 135
MIXED and HPMIXED compared (Introduction to Mixed Modeling), 136
monographs (Introduction to Mixed Modeling), 128
multiplicity adjustment (Introduction to Mixed Modeling), 137
nested (Introduction to Mixed Modeling), 128, 135
nonlinear (Introduction to Mixed Modeling), 129, 132
parameter estimation (Introduction to Mixed Modeling), 130, 131
procedures in SAS/STAT (Introduction to Mixed Modeling), 128
R matrix (Introduction to Mixed Modeling), 129, 131, 132, 135
R-side random effect (Introduction to Mixed Modeling), 129, 132
random effect (Introduction to Mixed Modeling), 127
random effect, G-side (Introduction to Mixed Modeling), 129–131
random effect, R-side (Introduction to Mixed Modeling), 129, 132
residual likelihood (Introduction to Mixed Modeling), 130, 135
restricted likelihood (Introduction to Mixed Modeling), 128, 130, 135
smoothing (Introduction to Mixed Modeling), 137
sparse techniques (Introduction to Mixed Modeling), 128
splines (Introduction to Mixed Modeling), 137
subjects (Introduction to Mixed Modeling), 133
subjects, compared to groups (Introduction to Mixed Modeling), 134
unbalanced (GLM), 2595
VARCOMP procedure, 7501
variance component (Introduction to Mixed Modeling), 129
mixed model (GLIMMIX)
parameterization, 2261
mixed model (HPMIXED)
descriptive statistics, 2838
hypothesis test, 2831
objective function, 2838
mixed model (MIXED), *see also* MIXED procedure
estimation, 3974
formulation, 3969
hypothesis tests, 3978, 3998
inference, 3978
inference space, 3917, 3918, 3920, 4018
least squares means, 3922
likelihood ratio test, 3978, 3979
linear model, 3892
maximum likelihood estimation, 3975
notation, 3894
objective function, 3996
parameterization, 3981
predicted values, 3922
restricted maximum likelihood, 4016

theory, 3968
Wald test, 3978, 4023
mixed model equations
example (MIXED), 4032
HPMIXED procedure, 2813
MIXED procedure, 3909, 3976
mixed model smoothing
GLIMMIX procedure, 2189, 2191, 2198, 2199, 2250
Introduction to Mixed Modeling, 137
MIXED Procedure
leverage, 3990
PRESS Residual, 3990
PRESS Statistic, 3990
MIXED procedure, *see also* mixed model
2D geometric anisotropic structure, 3959
Akaike's information criterion, 3907, 3976, 3997
Akaike's information criterion (finite sample corrected version), 3907, 3997
alpha level, 3905, 3921, 3925, 3929, 3950
anisotropic power covariance structure, 3960
anisotropic spatial power structure, 3960
ANTE(1) structure, 3959
ante-dependence structure, 3959
AR(1) structure, 3959
ARIMA procedure, compared, 3895
ARMA structure, 3959
assumptions, 3892
asymptotic covariance, 3905
AUTOREG procedure, compared, 3895
autoregressive moving-average structure, 3959
autoregressive structure, 3959, 4025
banded Toeplitz structure, 3959
basic features, 3893
Bayesian analysis, 3945
between-within method, 3907, 3931
BLUE, 3977
BLUP, 3977, 4046
Bonferroni adjustment, 3924
boundary constraints, 3944, 3945, 4011
BYLEVEL processing of LSMEANS, 3926
CALIS procedure, compared, 3895
chi-square test, 3919, 3929
Cholesky root, 3941, 3988, 4010
class level, 3909, 3995
classification variables, 3916
comparison with the GLIMMIX procedure, 2268
compound symmetry structure, 3959, 3970, 4026, 4031
computational details, 4010
computational order, 4010
conditional residuals, 3987
confidence interval, 3921, 3950
confidence limits, 3906, 3921, 3926, 3929, 3950
containment method, 3930, 3931
continuous effects, 3951, 3952, 3955, 3958
continuous-by-class effects, 3984
continuous-nesting-class effects, 3984
contrasted SAS procedures, 2434, 2540, 2597, 3895, 4254
contrasts, 3917, 3920
convergence criterion, 2269, 3905, 3906, 3996, 4012
convergence problems, 4011
convergence status, 3996
Cook's D, 3991
Cook's D for covariance parameters, 3991
correlation estimates, 3951, 3953, 3958, 4027
correlations of least squares means, 3926
covariance parameter estimates, 3906, 3907, 3997
covariance parameter estimates, ratio, 3915
covariance parameters, 3892
covariance structure, 3894, 3959, 3960, 4019
covariances of least squares means, 3926
covariate values for LSMEANS, 3925
covariates, 3982
CovRatio, 3992
CovRatio for covariance parameters, 3992
CovTrace, 3992
CovTrace for covariance parameters, 3992
CPU requirements, 4013
crossed effects, 3983
default output, 3995
degrees of freedom, 3918–3922, 3926, 3930, 3942, 3979, 3986, 3998, 4011, 4046
DFFITS, 3991
dimension information, 3995
dimensions, 3908, 3909
direct product structure, 3959
Dunnett's adjustment, 3924
EBLUPs, 3952, 3977, 4038, 4053
effect name length, 3909
empirical best linear unbiased prediction, 3940
empirical estimator, 3907
estimability, 3917, 3919–3922, 3927, 3942, 3978, 3986
estimable functions, 3939
estimation methods, 3908
exponential covariance structure, 3960
factor analytic structures, 3959

Fisher information matrix, 3997, 4032
Fisher's scoring method, 3905, 3915, 4012
fitting information, 3997, 3998
fixed effects, 3894
fixed-effects parameters, 3892, 3942, 3969
fixed-effects variance matrix, 3942
function evaluations, 3908
G matrix, 3894, 3949, 3950, 3969, 3970, 4052
gaussian covariance structure, 3960
general linear covariance structure, 3959
generalized inverse, 3919, 3977
GLIMMIX procedure, compared, 3896
gradient, 3906, 3996
grid search, 3943, 4032
growth curve analysis, 3970
Hannan-Quinn information criterion, 3907
Hessian matrix, 3905, 3906, 3915, 3944, 3996, 3997, 4011, 4012, 4023, 4032
heterogeneity, 3951, 3955, 4029
heterogeneous AR(1) structure, 3959
heterogeneous compound-symmetry structure, 3959
heterogeneous covariance structures, 3968
heterogeneous Toeplitz structure, 3959
hierarchical model, 4047
Hotelling-Lawley-McKeon statistic, 3955
Hotelling-Lawley-Pillai-Sampson statistic, 3955
Hsu's adjustment, 3924
Huynh-Feldt structure, 3959
infinite likelihood, 3954, 4011, 4012
influence diagnostics, 3938, 3988
influence plots, 4006
information criteria, 3907
initial values, 3943
input data sets, 3907
interaction effects, 3983
intercept, 3982
intercept effect, 3940, 3949
intraclass correlation coefficient, 4027
introductory example, 3896
iteration history, 3996
iterations, 3908, 3996
Kenward-Roger method, 3933
Kronecker product structure, 3959
LATTICE procedure, compared, 3895
least squares means, 3926, 4032, 4056
leave-one-out-estimates, 4006
likelihood distance, 3993
likelihood ratio test, 3998
linear covariance structure, 3959
log-linear variance model, 3956
main effects, 3982
marginal residuals, 3987
Matérn covariance structure, 3959
matrix notation, 3968
MDFFITS, 3991
MDFFITS for covariance parameters, 3992
memory requirements, 4013
missing level combinations, 3986
mixed linear model, 3892
mixed model, 3969
mixed model equations, 3909, 3976, 4032
mixed model theory, 3968
model information, 3909, 3995
model selection, 3976
multilevel model, 4047
multiple comparisons of least squares means, 3923, 3924, 3926
multiple tables, 4001
multiplicity adjustment, 3923
multivariate tests, 3955
nested effects, 3983
nested error structure, 4051
NESTED procedure, compared, 3895
Newton-Raphson algorithm, 3975
non-full-rank parameterization, 3895, 3956, 3986
nonstandard weights for LSMEANS, 3927
nugget effect, 3956
number of observations, 3995
oblique projector, 3990
observed margins for LSMEANS, 3927
ODS graph names, 4008
ODS Graphics, 3911, 4004
ODS table names, 3999
ordering of effects, 3910, 3985
over-parameterization, 3982
parameter constraints, 3944, 4011
parameterization, 3981
Pearson residual, 3941
pharmaceutical stability, example, 4047
plotting the likelihood, 4038
polynomial effects, 3982
power-of-the-mean model, 3956
predicted means, 3941
predicted value confidence intervals, 3929
predicted values, 3940, 4032
prior density, 3946
profiling residual variance, 3910, 3944, 3956, 3975, 4010
R matrix, 3894, 3954, 3957, 3969, 3970
random coefficients, 4025, 4047
random effects, 3894, 3949
random-effects parameters, 3893, 3952, 3969
regression effects, 3982

rejection sampling, 3947
repeated measures, 3893, 3954, 4019
residual diagnostics, details, 3986
residual method, 3931
residual plots, 4004
residual variance tolerance, 3941
restricted maximum likelihood (REML), 3893
ridging, 3915, 3975
sandwich estimator, 3907
Satterthwaite method, 3931
scaled residual, 3942, 3988
Schwarz's Bayesian information criterion, 3907, 3976, 3997
scoring, 3905, 3915, 4012
Sidak's adjustment, 3924
simple effects, 3928
simulation-based adjustment, 3924
singularities, 4012
spatial anisotropic exponential structure, 3959
spatial covariance structure, 3960, 3968, 4011
split-plot design, 3972, 4014
standard linear model, 3894
statement positions, 3902
studentized residual, 3941, 3991
subject effect, 3918, 3952, 3958, 4013, 4019
summary of commands, 3903
sweep operator, 3991, 4010
table names, 3999
test components, 3939
Toeplitz structure, 3959, 4056
TSCSREG procedure, compared, 3895
Tukey's adjustment, 3924
Type 1 estimation, 3908
Type 1 testing, 3934
Type 2 estimation, 3908
Type 2 testing, 3934
Type 3 estimation, 3908
Type 3 testing, 3934, 3998
unstructured correlations, 3959
unstructured covariance matrix, 3959
unstructured R matrix, 3958
V matrix, 3953
VARCOMP procedure, example, 4032
variance components, 3893, 3959
variance ratios, 3944, 3952
Wald test, 3997, 3998
weighted LSMEANS, 3927
weighting, 3968
zero design columns, 3933
zero variance component estimates, 4011

mixed-effects models
MCMC procedure, 3614

mixing
convergence (MCMC), 3683
improving (MCMC), 3567, 3605, 3683
MCMC procedure, 3605, 3683

mixture
chi-square (GLIMMIX), 2127, 2132
chi-square, weights (GLIMMIX), 2134
Poisson (GLIMMIX), 2398

ML factor analysis
and computer time, 1539
and confidence intervals, 1537, 1539, 1571
and multivariate normal distribution, 1539
and standard errors, 1539

ML method
TCALIS procedure, 6889

MLE
SEQDESIGN procedure, 5818

MLE ordering
SEQTEST procedure, 6026

modal clusters
density estimation (CLUSTER), 1243

modal region, definition, 4117

MODECLUS procedure
p-value computation, 4116
analyzing data in groups, 4095, 4108, 4113
cascaded density estimates, 4112
clustering methods, 4095, 4113
clusters, definition, 4117
clusters, plotting, 4117
compared with other procedures, 4095
cross validated density estimates, 4111
density estimation, 4109
example using GPLOT procedure, 4170
example using SGPLOT procedure, 4162
example using TRACE option, 4174
example using TRANSPOSE procedure, 4151
fixed-radius kernels, 4110
functional summary, 4101
Hertzsprung-Russell Plot, example, 4170
JOIN option, discussion, 4119
modal region, 4117
neighborhood distribution function (NDF), definition, 4117
nonparametric clustering methods, 4094
output data sets, 4121
plotting samples from univariate distributions, 4128
population clusters, risks of estimating, 4116
saddle test, definition, 4118
scaling variables, 4094
significance tests, 4162
standardizing, 4094

summary of options, 4101
variable-radius kernels, 4110
model
fit criteria (PHREG), 4610
fit summary (REG), 5557
fitting criteria (LOGISTIC), 3329
hierarchy (GLMSELECT), 2708
hierarchy (LOGISTIC), 3258, 3294
hierarchy (PHREG), 4567
information (GLIMMIX), 2274
information (MIXED), 3909
parameterization (GLM), 2491
specification (ANOVA), 720
specification (GLM), 2488
specification (NLMIXED, 4376
model assessment, 2047, 2054
PHREG procedure, 4540, 4618, 4700
model building
examples (REG), 5592
model checking, 2047, 2054
model information
GLMSELECT procedure, 2743
HPMIXED procedure, 2813
MIXED procedure, 3995
PHREG procedure, 4632, 4633, 4637
model parameters
SURVEYLOGISTIC procedure, 6423
model selection, 5113, 5124
entry (PHREG), 4571
examples (REG), 5608
GLMSELECT procedure, 2719
LOGISTIC procedure, 3290, 3301, 3328
MIXED procedure, 3976
PHREG procedure, 4524, 4565, 4570, 4617
REG procedure, 5435, 5523, 5526, 5527
removal (PHREG), 4571
model selection issues
GLMSELECT procedure, 2726
model specification
MCMC procedure, 3512
modeling
KRIGE2D procedure, 2930
VARIOGRAM procedure, 7518
modeling language (TCALIS), 6714
modification indices
CALIS procedure, 854, 928, 953
constraints (CALIS), 860, 861
constraints (TCALIS), 6734, 6735
displaying (CALIS), 966
displaying (TCALIS), 6938
Lagrange multiplier test (CALIS), 860, 954
Lagrange multiplier test (TCALIS), 6734, 6918
TCALIS procedure, 6729, 6895, 6917
Wald test (CALIS), 860, 954
Wald test (TCALIS), 6734, 6918
modified Peto-Peto test for homogeneity
LIFETEST procedure, 3100, 3130
modified ridit scores, 1731
monoecious population analysis
example (INBREED), 2877
monotone likelihood
PHREG procedure, 4567, 4595, 4662
monotone missing pattern
MI procedure, 3745, 3772
monotone transformations
TRANSREG procedure, 7179
monotonic
transformation (PRINQUAL), 5215, 5216
transformation, B-spline (PRINQUAL), 5215
transformation, B-spline (TRANSREG), 7261
monotonic B-spline transformation
TRANSREG procedure, 7146
monotonic transformation, ties not preserved
TRANSREG procedure, 7146
monotonic transformation, ties preserved
TRANSREG procedure, 7146, 7260
Monte Carlo estimation
exact tests (NPAR1WAY), 4479
FREQ procedure, 1691, 1693, 1782
Monte Carlo standard error (MCSE)
Introduction to Bayesian Analysis, 148, 170
Mood scores
NPAR1WAY procedure, 4472
Moore-Penrose inverse
NLIN procedure, 4302
MORALS method
TRANSREG procedure, 7162
Moran *I* coefficient, *see* autocorrelation
mortality test
MULTTEST procedure, 4207, 4233
MSE
SURVEYREG procedure, 6571
MSTRUCT model
simple covariance structures model example (TCALIS), 6712
TCALIS procedure, 6792
MTV method
PRINQUAL procedure, 5220
multicollinearity
REG procedure, 5555
multidimensional preference analysis
PRINQUAL procedure, 5238
multidimensional scaling
MDS procedure, 3704
metric (MDS), 3704

nonmetric (MDS), 3704
three-way (MDS), 3704
multilevel model
example (MIXED), 4047
multilevel response, 5313
multimember effect
GLIMMIX procedure, 378, 2412
GLMSELECT procedure, 378
multimember example
GLIMMIX procedure, 2412
multinomial
distribution (GENMOD), 1966
models (GENMOD), 1984
multinomial distribution
GLIMMIX procedure, 2168
multiple classifications
cutpoints (LOGISTIC), 3299
multiple comparison adjustment (GLIMMIX)
estimates, 2135
least squares means, 2143, 2144, 2156
multiple comparison adjustment (MIXED)
least squares means, 3923, 3924
multiple comparison procedures
GLM procedure, 2468
multiple-stage tests, 2522
pairwise (GLM), 2515
recommendations, 2524
with a control (GLM), 2519
with the average (GLM), 2520
multiple comparisons of estimates
GLIMMIX procedure, 2135
multiple comparisons of least squares means, *see also* multiple-comparison procedures
GLIMMIX procedure, 2143, 2144, 2147, 2153, 2156
GLM procedure, 2459, 2462, 2516
interpretation, 2524
MIXED procedure, 3923, 3924, 3926
multiple comparisons of means, *see also* multiple-comparison procedures
ANOVA procedure, 710
Bonferroni *t* test, 711, 2470
Duncan's multiple range test, 711, 2470
Dunnett's test, 711, 712, 2470
error mean square, 712, 2471
examples, 2556
Fisher's LSD test, 714, 2473
Gabriel's procedure, 712, 2471
GLM procedure, 2514, 2516
GT2 method, 714, 2473
interpretation, 2524
Ryan-Einot-Gabriel-Welsch test, 713, 2472
Scheffé's procedure, 714, 2472
Sidak's adjustment, 714, 2473
SMM, 714, 2473
Student-Newman-Keuls test, 714, 2473
Tukey's studentized range test, 714, 2473
Waller-Duncan method, 713
Waller-Duncan test, 714, 2473
multiple correspondence analysis (MCA)
CORRESP procedure, 1327, 1354, 1374
multiple imputation efficiency
MI procedure, 3795
MIANALYZE procedure, 3857
multiple imputations analysis, 3744, 3840
multiple R-square
SURVEYREG procedure, 6570
multiple redundancy coefficients
TRANSREG procedure, 7177
multiple regression, 5049
TRANSREG procedure, 7179
multiple tables
MIXED procedure, 4001
multiple-comparison adjustments
LIFETEST procedure, 3101
multiple-comparison procedures, 2514, *see also* multiple comparisons of least squares means, *see also* multiple comparisons of means
pairwise (GLM), 2514
with a control (GLM), 2515
multiple-stage tests, *see* multiple comparison procedures, 2522
multiplicative hazards model, *see* Andersen-Gill model
multiplicity adjustment
Bonferroni (GLIMMIX), 2144, 2156
Bonferroni (LIFETEST), 3127
Dunnett (GLIMMIX), 2144
Dunnett (LIFETEST), 3127
estimates (GLIMMIX), 2135
GLIMMIX procedure, 2135
Hsu (GLIMMIX), 2144
least squares means (GLIMMIX), 2144
least squares means estimates (GLIMMIX), 2156
MIXED procedure, 3923
Nelson (GLIMMIX), 2144
row-wise (GLIMMIX), 2135, 2143
row-wise (MIXED), 3923
Scheffe (GLIMMIX), 2144, 2156
Scheffe (LIFETEST), 3127
Sidak (GLIMMIX), 2144, 2156
Sidak (LIFETEST), 3127
Simulate (GLIMMIX), 2156
simulated (LIFETEST), 3127
simulation-based (GLIMMIX), 2144

step-down *p*-values (GLIMMIX), 2138, 2154, 2159
studentized maximum modulus (LIFETEST), 3127
T (GLIMMIX), 2156
Tukey (GLIMMIX), 2144
Tukey (LIFETEST), 3127
multistage sampling
Introduction to Survey Procedures, 266
SURVEYSELECT procedure, 6614
multivariate analysis of variance, 702, 706
CANDISC procedure, 1065
examples (GLM), 2582
GLM procedure, 2446, 2464, 2531
hypothesis tests (GLM), 2531
partial correlations, 2531
multivariate data
heterocatanomic (Introduction to Mixed Modeling), 137
heterocatanomic (Introduction to Modeling), 35
heterogeneous (Introduction to Modeling), 35
homocatanomic (Introduction to Modeling), 35
homogeneous (Introduction to Modeling), 35
multivariate general linear hypothesis, 2531
multivariate inferences
MIANALYZE procedure, 3857
multivariate multiple regression
TRANSREG procedure, 7179
multivariate normality assumption
MI procedure, 3798
multivariate tests
MIXED procedure, 3955
REG procedure, 5577
repeated measures, 2535, 2536
multiway tables
SURVEYFREQ procedure, 6312, 6358
MULTTEST procedure
adaptive FDR adjustment, 4189
adaptive Hochberg adjustment, 4188
adaptive Holm adjustment, 4189
adaptive methods, 4215
adjusted *p*-value, 4182, 4211
Bonferroni adjustment, 4189, 4212
bootstrap adjustment, 4185, 4189, 4213
bootstrap FDR adjustment, 4190
Cochran-Armitage test, 4201, 4203, 4206, 4225
computational resources, 4220
convolution distribution, 4205
dependent FDR adjustment, 4189
displayed output, 4222
double arcsine test, 4206
expected trend, 4206
false discovery rate, 4211
false discovery rate adjustment, 4216
familywise error rate, 4211
familywise error rate adjustment, 4212
fast Fourier transform, 4205
Fisher combination adjustment, 4215
Fisher exact test, 4199, 4201, 4208
Freeman-Tukey test, 4201, 4206, 4229
Hochberg adjustment, 4215
Hommel adjustment, 4215
introductory example, 4184
linear trend test, 4204
missing values, 4219
ODS graph names, 4224
ODS table names, 4223
output data sets, 4220
p-value adjustments, 4182, 4211
permutation adjustment, 4194, 4213, 4236
permutation FDR adjustment, 4190
Peto test, 4201, 4207, 4233
positive false discovery rate, 4211
positive FDR adjustment, 4194, 4218
resampled data sets, 4222
Sidak's adjustment, 4197, 4213
statistical tests, 4203
step-down methods, 4214
strata weights, 4206
t test, 4201, 4209, 4229
Murthy's method
SURVEYSELECT procedure, 6653

name-lists
POWER procedure, 4919
narratives, 5063
Power and Sample Size application, 425
natural response rate, 5258, 5266
nearest centroid sorting, 1622, 1623
nearest neighbor method, *see also* single linkage
DISCRIM procedure, 1403, 1405
negative binomial distribution
definition of (MCMC), 3538
GENMOD procedure, 1965
GLIMMIX procedure, 2168
MCMC procedure, 3514, 3538
NLMIXED procedure, 4376
negative variance components
VARCOMP procedure, 7501
neighborhood distribution function (NDF), definition
MODECLUS procedure, 4117
Nelder-Mead simplex

method (GLIMMIX), 401
method (NLMIXED), 4371
Nelder-Mead simplex method
Shared Concepts, 409
Nelson's adjustment
GLIMMIX procedure, 2144
GLM procedure, 2459
Nelson-Aalen estimates
LIFETEST procedure, 3117
nested design, 4253
error terms, 4260
generating with PLAN procedure, 4750
hypothesis tests (NESTED), 4259
nested effects
design matrix (CATMOD), 1155
GENMOD procedure, 1974
GLIMMIX procedure, 2261
MIXED procedure, 3983
model parameterization (GLM), 2492
Shared Concepts, 371
specifying (ANOVA), 721, 722
specifying (CATMOD), 1142
specifying (GLM), 2488
nested error structure
MIXED procedure, 4051
nested models
KRIGE2D procedure, 2952–2954
VARIOGRAM procedure, 7543
NESTED procedure
analysis of covariation, 4259
compared to other procedures, 2434, 3895, 4254
computational method, 4260
input data sets, 4257
introductory example, 4255
missing values, 4259
ODS table names, 4263
random effects, 4259
unbalanced design, 4259
nested v. crossed effects
Shared Concepts, 371
nested-by-value effects
specifying (CATMOD), 1143
network algorithm, 1780
exact tests (NPAR1WAY), 4477
Newman-Keuls' multiple range test, 714, 2473, 2522
Newton algorithm
FASTCLUS procedure, 1637
Newton-Raphson algorithm
CALIS procedure, 856, 857, 868, 945
GENMOD procedure, 1968
GLIMMIX procedure, 402
iteration (PHREG), 4537
LIFEREG procedure, 2992
LOGISTIC procedure, 3301, 3302, 3324, 3326
MIXED procedure, 3975
NLMIXED procedure, 4371
PHREG procedure, 4595
PROBIT procedure, 5310
SURVEYLOGISTIC procedure, 6401, 6418
TCALIS procedure, 6731, 6736, 6923
Newton-Raphson algorithm with ridging
GLIMMIX procedure, 402
NLMIXED procedure, 4371
Newton-Raphson method
Shared Concepts, 406
Newton-Raphson with ridging
Shared Concepts, 407
Neyman allocation
SURVEYSELECT procedure, 6641, 6656
NLIN procedure
g2 inverse, 4302
g4 inverse, 4302
alpha level, 4289
analytic derivatives, 4285, 4293
automatic derivatives, 4293
close-to-linear, 4295
comparing trends, 4331
confidence interval, 4287, 4288, 4306, 4307
convergence, 4299
convergence criterion, 4279
covariance matrix, 4307
cross-referencing variables, 4283
debugging execution, 4280
derivatives, 4285, 4289, 4293
G4 inverse, 4280
Gauss iterative method, 4281
Gauss-Newton iterative method, 4303, 4304
Gauss-Newton method, 4302
generalized inverse, 4302
gradient method, 4302, 4305
group comparisons, 4331
Hessian, 4308
Hougaard's measure, 4280, 4295
incompatibilities, 4310, 4311
initial values, 4289
iteratively reweighted least squares example, 4318
Jacobian, 4279
Lagrange multiplier, 4284
Lagrange multipliers, covariance matrix, 4308
log-logistic model, 4324
Marquardt iterative method, 4281, 4302, 4305
maximum iterations, 4281

maximum subiterations, 4281
mean square error specification, 4282
missing values, 4297
model confidence interval, 4307
model.variable syntax, 4286
Moore-Penrose inverse, 4302
Newton iterative method, 4281, 4302–4304
object convergence measure, 4275
options summary (PROC statement), 4278
output table names, 4311
parameter confidence interval, 4306
PPC convergence measure, 4275
predicted values, output, 4287
prediction interval, 4287, 4288
R convergence measure, 4275
residual values, output, 4287
retaining variables, 4285, 4292
RPC convergence measure, 4275
segmented model example, 4313
singularity criterion, 4282
skewness, 4276, 4280, 4295
SMETHOD=GOLDEN step size search, 4306
special variables, 4297
specifying bounds, 4283
standard error, 4288
starting values, 4289
steepest descent method, 4302, 4305
step size search, 4306
switching model, 4324
trend comparisons, 4331
troubleshooting, 4299
tuning display of iteration computation, 4282
weighted regression, 4298
NLMIXED procedure
accelerated failure time model, 4427
active set methods, 4395
adaptive Gaussian quadrature, 4382
additional estimates, 4375, 4407
alpha level, 4358
arrays, 4372
assumptions, 4381
Bernoulli distribution, 4376
binary distribution, 4376
binomial distribution, 4376
bounds, 4373
Cholesky parameterization, 4414
compared with other SAS procedures and macros, 4345
computational problems, 4399
computational resources, 4404
contrasts, 4374
convergence criteria, 4357, 4363, 4388
convergence problems, 4400
convergence status, 4407
covariance matrix, 4358, 4402, 4407
cross-referencing variables, 4372
degrees of freedom, 4359
empirical Bayes estimate, 4344, 4352, 4359, 4377, 4379
empirical Bayes estimation, 4382
empirical Bayes options, 4359
examples, *see also* examples, NLMIXED, 4409
finite differencing, 4361, 4392
first-order method, 4383
fit statistics, 4407
floating point errors, 4399
frailty, 4427
frailty model example, 4427
functional convergence criteria, 4361
gamma distribution, 4376
Gaussian distribution, 4376
general distribution, 4376
generalized inverse, 4362
growth curve example, 4346
Hessian matrix, 4364
Hessian scaling, 4363, 4394
integral approximations, 4367, 4382
intraclass correlation coefficient, 4420
iteration details, 4364
iteration history, 4364, 4405
lag functionality, 4380
Lagrange multiplier, 4395
line-search methods, 4364, 4365, 4397
log-likelihood function, 4384
logistic-normal example, 4351
long run times, 4399
maximum likelihood, 4345
negative binomial distribution, 4376
Newton-Raphson algorithm with ridging, 4371
normal distribution, 4376, 4378
notation, 4381
ODS table names, 4408
optimization techniques, 4370, 4387
options summary, 4355
overflows, 4399
parameter estimates, 4407
parameter rescaling, 4399
parameter specification, 4376
pharmakokinetics example, 4409
Poisson distribution, 4376
Poisson-normal example, 4424
precision, 4402
prediction, 4377, 4403
probit-normal-binomial example, 4415

probit-normal-ordinal example, 4419
programming statements, 4379
projected gradient, 4395
projected Hessian, 4395
quadrature options, 4369
random effects, 4378
references, 4437
replicate subjects, 4379
sandwich estimator, 4360
singularity tolerances, 4370
sorting of input data set, 4359, 4378
stationary point, 4401
step length options, 4397
syntax summary, 4354
termination criteria, 4357, 4388
update methods, 4371
NLOPTIONS statement
Syntax (Shared Concepts), 391
nominal level of measurement
DISTANCE procedure, 1484
nominal power
GLMPOWER procedure, 2661, 2663, 2669
POWER procedure, 4825, 4922, 4924
nominal variable
DISTANCE procedure, 1485
nominal variables, 183, *see also* classification variables, *see also* classification variables
non-full-rank models
REG procedure, 5553
non-full-rank parameterization
GLIMMIX procedure, 2262
MIXED procedure, 3895, 3956, 3986
non-positional syntax
GLIMMIX procedure, 2124, 2264, 2406
noncentral distributions, 417
noncentrality parameter, 417
noninferiority tests, 416
power and sample size (POWER), 4851, 4999
noninferiority trial
SEQDESIGN procedure, 5823
nonlinear
mixed models (NLMIXED), 4344
nonlinear fit functions
TRANSREG procedure, 7209
nonlinear fit transformations
TRANSREG procedure, 7144
nonlinear model
Introduction to Modeling, 31
Introduction to Regression, 77, 91
nonlinear Poisson regression
MCMC procedure, 3605
nonlinear transformations
TRANSREG procedure, 7179
nonmetric conjoint analysis
TRANSREG procedure, 7329
nonmetric multidimensional scaling
MDS procedure, 3704
nonoptimal transformations
PRINQUAL procedure, 5214
TRANSREG procedure, 7143
nonparametric clustering methods
MODECLUS procedure, 4094
nonparametric discriminant analysis, 1403
nonparametric measures of association
Introduction to Nonparametric Analysis, 296
nonparametric regression
TPSPLINE procedure, 7057
nonparametric tests
Introduction to Nonparametric Analysis, 291
NPAR1WAY procedure, 4444
power and sample size (POWER), 4911, 4915
nonrandom trend, *see* surface trend
nonsuperiority tests, 416
NOPRINT option
ODS, 447
normal distribution, 2992, 3022, 3039, 5258
definition of (MCMC), 3538
GENMOD procedure, 1965
GLIMMIX procedure, 2168
MCMC procedure, 3514, 3538
NLMIXED procedure, 4376, 4378
PROBIT procedure, 5311
TTEST procedure, 7402
normal kernel (DISCRIM), 1404
normal scores
NPAR1WAY procedure, 4471
normality
testing for (Introduction to Nonparametric Analysis), 292
normality assumption
SEQDESIGN procedure, 5872
VARIOGRAM procedure, 7534, 7562
normalization of the estimates
MDS procedure, 3728
normalizing constant
definition of, 143
Introduction to Bayesian Analysis, 143
NOSQUARE option
algorithms used (CLUSTER), 1259
notation
GLIMMIX procedure, 2083
NPAR1WAY procedure
alpha level, 4457, 4464
Ansari-Bradley scores, 4471
box plots, 4450, 4459, 4493

Brown-Mood test, 4470
compared to other procedures, 2434
computational resources, 4478
Conover scores, 4472
Cramer-von Mises test, 4476
displayed output, 4486
EDF plots, 4459, 4498
EDF tests, 4474
exact *p*-values, 4478
exact tests, 4476
Hodges-Lehmann estimation, 4472
Introduction to Nonparametric Analysis, 291–293, 295
introductory example, 4445
Klotz scores, 4472
Kolmogorov-Smirnov test, 4475
Kruskal-Wallis test, 4470
Kuiper test, 4476
Mann-Whitney-Wilcoxon test, 4470
median plots, 4459, 4497
median scores, 4470
missing values, 4466
Monte Carlo estimation, 4479
Mood scores, 4472
network algorithm, 4477
normal scores, 4471
ODS Graphics, 4459, 4492
ODS table names, 4490
one-way ANOVA tests, 4469
output data sets, 4465, 4480
permutation test, 4461, 4470
Pitman's test, 4461, 4470
rank tests, 4467
Savage scores, 4471
scores, 4470
Siegel-Tukey scores, 4471
statistical computations, 4467
tied values, 4467
Van der Waerden scores, 4471
Wilcoxon scores, 4470
nugget effect
KRIGE2D procedure, 2946, 2954
MIXED procedure, 3956
VARIOGRAM procedure, 7543
null hypothesis, 415, 4916
number of imputations
MI procedure, 3798
number of intervals
life-table estimates (LIFETEST), 3117
number of observations
GLIMMIX procedure, 2274
GLMSELECT procedure, 2743
HPMIXED procedure, 2813
MIXED procedure, 3995
PHREG procedure, 4632, 4637
number of replicates
SURVEYLOGISTIC procedure, 6387, 6429–6431
SURVEYMEANS procedure, 6482, 6506–6508
SURVEYREG procedure, 6548, 6565–6567
number of stages
SEQDESIGN procedure, 5872
SEQTEST procedure, 6008
number of subjects at risk
PHREG procedure, 4573
number-lists
GLMPOWER procedure, 2660
POWER procedure, 4919
numerical integration
theory (GLIMMIX), 2227

O'Brien's test for homogeneity of variance
ANOVA procedure, 713
GLM procedure, 2471
O'Brien-Fleming method
SEQDESIGN procedure, 5807, 5827, 5842, 5874
O'Brien-Fleming-type error spending function
SEQDESIGN procedure, 5806
O'Brien-Fleming-type error spending method
SEQDESIGN procedure, 5849
objective function
mixed model (HPMIXED), 2838
mixed model (MIXED), 3996
objects
ODS, 434
oblimin method, 1534, 1561, 6756
oblique component analysis, 7462
oblique projector
MIXED procedure, 3990
oblique transformation, 1537, 1541
observed Fisher information
SEQDESIGN procedure, 5818, 5820
odds estimation
GLIMMIX procedure, 2256
odds ratio
adjusted, 1774
Breslow-Day test, 1776
case-control studies, 1760, 1774
confidence limits (LOGISTIC), 3292, 3299
customized (LOGISTIC), 3316
estimation (LOGISTIC), 3335
Introduction to Regression, 89
logit estimate, 1774
Mantel-Haenszel estimate, 1774
power and sample size (POWER), 4884, 4889, 4966, 4967

SURVEYFREQ procedure, 6344
SURVEYLOGISTIC procedure, 6434
with interactions (LOGISTIC), 3303
odds ratio estimation
GLIMMIX procedure, 2256
SURVEYLOGISTIC procedure, 6434
ODS
correlation matrix, 486
covariance matrix, 486
data set concatenation, 461
default behavior, 433
destinations, 434, 439, 461
exclusion list, 439
HTML destination, 448
HTML links, 477, 482
interactive procedures, 439
links, HTML, 477, 482
NOPRINT option, 447
objects, 434
ODS Graphics, 499
output data set creation, 455, 457, 460
output exclusion, 454
output formats, 433
output objects, 434
output selection, 451
output, suppressing, 447
path names, 437
paths, 435
Results window, 440
RUN-group processing, 439, 461
SASUSER.Templat, 445
selection list, 439, 453
Statistical Graphics Using ODS, 499
style templates, 441
table names, 435, 457
table templates, 441
TEMPLATE procedure, 441, 464, 479
templates, 434, 441
templates, displaying contents, 441
templates, modifying, 444, 464, 472, 479
trace output, 437
ODS destination
ODS Graphics, 542
ODS destination FILE= option
ODS Graphics, 536
ODS destination statement
ODS Graphics, 523, 532, 535
ODS examples
GLMMOD procedure, 2637
ORTHOREG procedure, 4516
PLS procedure, 4810
ODS Graph names
CLUSTER procedure, 1269
LIFETEST procedure, 3156
MDS procedure, 3731
PHREG procedure, 4643
PRINCOMP procedure, 5163
TRANSREG procedure, 7301
ODS graph names
ANOVA procedure, 729
CALIS procedure, 970
FACTOR procedure, 1585
FREQ procedure, 1798
GAM procedure, 1862
GLIMMIX procedure, 2299
GLM procedure, 2555
GLMSELECT procedure, 2755
KDE procedure, 2906
KRIGE2D procedure, 2969
LOESS procedure, 3226
LOGISTIC procedure, 3379
MI procedure, 3800
MIXED procedure, 4008
MULTTEST procedure, 4224
PLS procedure, 4793
PRINCOMP procedure, 5163
PRINQUAL procedure, 5237
PROBIT procedure, 5328
QUANTREG procedure, 5405
REG procedure, 5591
ROBUSTREG procedure, 5696
RSREG procedure, 5747
SIM2D procedure, 6182
TCALIS procedure, 6952
TTEST procedure, 7427
VARIOGRAM procedure, 7569
ODS GRAPHICS
examples (REG), 5592
ODS Graphics, 499
accessing individual graphs, 524
axis customization, 637
axis labels, modifying, 613
box plot, 601, 645
contour plot, 546
destination, closing, 546
destinations, 542
diagnostics panel, 633
DOCUMENT destination, 606
document path, 608
DOCUMENT procedure, 606
Documents window, 607
editing templates, 612
excluding graphs, 540
file name, base, 543
fonts, modifying, 570
getting started, 501
GLIMMIX procedure, 2112, 2282
graph label, 538

graph modification, 525
graph name, 538, 607
graph resolution, 525, 547
graph size, 525, 547
graph template language, 577
graph templates, 577
graph templates, customizing, 587
graph templates, definition, 612
graph templates, displaying, 583
graph templates, editing, 585
graph templates, locating, 582
graph templates, reverting to default, 588
graph templates, saving, 587, 613
graph titles, modifying, 613
graphics image file, 541–543
graphics image file, saving, 544
graphics image file, type, 542
grid lines, 619
histogram, 604
HTML destination, 541, 545
index counter, 543
KRIGE2D procedure, 2937
LaTeX destination, 541, 546, 604
lines, 567
LISTING destination, 545
LMSELECT procedure, 2698, 5456
LOESS procedure, 3202, 3226
markers, 567
MIXED procedure, 3911, 4004
multiple destinations, 544, 546
NPAR1WAY procedure, 4459, 4492
ODS destination, 542
ODS destination FILE= option, 536
ODS destination statement, 523, 532, 535
ODS Graphics Editor, 548
ODS GRAPHICS statement, 529
PDF destination, 546
PostScript, 546
presentations, 602
primer, 519
procedures, 528
reference lines, 637
referring to graphs, 538
replaying output, 607
Results Window, 537
RTF destination, 602
SASHELP.Tmplmst, 587
SASUSER.Templat, 587
scatter plot, 578
selecting graphs, 540, 607
SGPANEL procedure, 593
SGPLOT procedure, 589
SGRENDER procedure, 596
SGSCATTER procedure, 591
SIM2D procedure, 6166
statistical graphics procedures, 589
style, 520, 554
style elements, 558
style modification, 619
style modification, ModStyle macro, 564
style, box plot, 644
style, customizing, 573
style, default, 575
surface plot, 546
survival plot, 504, 622
template library, default, 553
template modification, 610
template primary statement, 653
template statement order, 653
text, adding to plots, 647
tooltips, 601
trace output, 610
traditional graphics, 528
TTEST procedure, 7427
VARIOGRAM procedure, 7530
viewing graphs, 537
ODS graphics
GLMSELECT procedure, 2749
ODS Graphics Editor
ODS Graphics, 548
ODS GRAPHICS statement
ODS Graphics, 529
ODS path, 575, 587
ODS Statistical Graphics, *see* ODS Graphics
ODS Styles
ANALYSIS style, 510
DEFAULT style, 509
JOURNAL style, 604
JOURNAL2 style, 512
LISTING style, 518
RTF style, 515
STATISTICAL style, 514
ODS table names
KRIGE2D procedure, 2969
SIM2D procedure, 6181
SURVEYLOGISTIC procedure, 6447
SURVEYREG procedure, 6581
VARIOGRAM procedure, 7569
offset
GENMOD procedure, 1952, 2011
GLIMMIX procedure, 2174, 2209, 2322, 2323, 2386
offset variable
GENMOD procedure, 1901
PHREG procedure, 4569
offspring
INBREED procedure, 2864, 2865, 2872
one-sample *t* test, 5052

one-sample *t*-test
power and sample size (POWER), 4818, 4852, 4857, 4858, 4953–4955
one-sample *t* test
TTEST procedure, 7392
one-sample test for binomial proportion
SEQDESIGN procedure, 5812
one-sample test for mean
SEQDESIGN procedure, 5811
one-sample tests
SEQDESIGN procedure, 5858
one-sided *t* test
TTEST procedure, 7407
one-sided repeated confidence intervals
SEQTEST procedure, 6023
one-sided test
SEQDESIGN procedure, 5821
one-way ANOVA, 5049, 5111
power and sample size (POWER), 4859, 4863, 4957, 4981
one-way ANOVA tests
NPAR1WAY procedure, 4469
online documentation, 23
operations research, 25
optimal
scoring (PRINQUAL), 5215
transformations (MDS), 3704, 3705, 3716
transformations (PRINQUAL), 5215
optimal allocation
SURVEYSELECT procedure, 6641, 6655
optimal scaling
TRANSREG procedure, 7259
optimal scoring
TRANSREG procedure, 7146, 7260
optimal transformations
TRANSREG procedure, 7145
optimization
CALIS procedure, 830, 896, 945
conjugate gradient (CALIS), 856, 857, 867, 869, 945
conjugate gradient (TCALIS), 6731, 6736, 6737, 6923
double dogleg (TCALIS), 6730, 6736, 6738, 6923
double-dogleg (CALIS), 856, 868, 869, 945
GLIMMIX procedure, 2175
history (CALIS), 948
history (TCALIS), 6926
initial values (CALIS), 941, 946
initial values (TCALIS), 6922, 6924
Levenberg-Marquardt (CALIS), 856, 868, 945
Levenberg-Marquardt (TCALIS), 6730, 6736, 6923
line search (CALIS), 857, 952
line search (TCALIS), 6731, 6930
memory problems (CALIS), 946
memory problems (TCALIS), 6924
Newton-Raphson (CALIS), 856, 857, 868, 945
Newton-Raphson (TCALIS), 6731, 6736, 6923
nonlinear constraints (CALIS), 947
nonlinear constraints (TCALIS), 6925
quasi-Newton (CALIS), 856, 857, 868, 870, 945, 947
quasi-Newton (TCALIS), 6731, 6737, 6738, 6923, 6925
step length (CALIS), 952
step length (TCALIS), 6930
TCALIS procedure, 6684, 6923
techniques (NLMIXED), 4370, 4387
termination criteria (CALIS), 886, 890, 892–895
trust region (TCALIS), 6737, 6923
trust-region (CALIS), 856, 868, 945
trust-region (TCALIS), 6730
update method (CALIS), 869
update method (TCALIS), 6737
optimization information
GLIMMIX procedure, 2275
optimization statements (TCALIS), 6718
optimization technique
GLIMMIX procedure, 401
options summary
LSMEANS statement, (GLIMMIX), 2142
LSMEANS statement, (MIXED), 3922
MODEL statement (GLIMMIX), 2161
MODEL statement (LOESS), 3208
MODEL statement (MIXED), 3928
NLOPTIONS statement (GLIMMIX), 392
NLOPTIONS statement (HPMIXED), 392
NLOPTIONS statement (TCALIS), 392
PROC GLIMMIX statement, 2094
PROC MIXED statement, 3904
RANDOM statement (GLIMMIX), 2186
RANDOM statement (MIXED), 3949
REPEATED statement (MIXED), 3954
options summary (PROC statement)
NLIN procedure, 4278
order statistics, *see* RANK procedure
ordering
of class levels (Shared Concepts), 367
ordering observations
INBREED procedure, 2858
ordinal level of measurement
DISTANCE procedure, 1484
ordinal model

CATMOD procedure, 1147
GENMOD procedure, 2032
ORDINAL parameterization
SURVEYLOGISTIC procedure, 6413
ordinal parameterization
Shared Concepts, 374
ordinal variable, 183
ordinal variables
transformed to interval (RANKSCORE=), 1498
ordinary kriging
KRIGE2D procedure, 2962–2966
ORTHEFFECT parameterization
SURVEYLOGISTIC procedure, 6414
ortheffect parameterization
Shared Concepts, 375
orthoblique rotation, 7462
orthogonal coding
TRANSREG procedure, 7154
orthogonal polynomial contrasts, 717
orthogonal transformation, 1537, 1541
orthomax method, 1534, 1560, 6755
orthonormalizing transformation matrix
ANOVA procedure, 708
GLM procedure, 2466
ORTHORDINAL parameterization
SURVEYLOGISTIC procedure, 6414
orthordinal parameterization
Shared Concepts, 376
ORTHOREG procedure
compared to other procedures, 4505, 4506
input data sets, 4509
introductory example, 4506
missing values, 4512
ODS table names, 4514
output data sets, 4510, 4512
orthoterm parameterization
Shared Concepts, 376
ORTHOTHERM parameterization
SURVEYLOGISTIC procedure, 6414
ORTHPOLY parameterization
SURVEYLOGISTIC procedure, 6414
orthpoly parameterization
Shared Concepts, 376
ORTHREF parameterization
SURVEYLOGISTIC procedure, 6414
orthref parameterization
Shared Concepts, 376
OUT= data sets
ACECLUS procedure, 680
CANCORR procedure, 1046
FACTOR procedure, 1559, 1568
FASTCLUS procedure, 1643
PRINCOMP procedure, 5158
SCORE procedure, 5769
TREE procedure, 7369
OUTEST= data sets
KRIGE2D procedure, 2967
LIFEREG procedure, 3051
QUANTREG procedure, 5399
ROBUSTREG procedure, 5690
outliers
FASTCLUS procedure, 1622
MODECLUS procedure, 4111
OUTNBHD= data set
KRIGE2D procedure, 2967, 2968
output data set
SCORE procedure, 5766, 5769
output data set creation
ODS, 455, 457, 460
output data sets
ACECLUS procedure, 680
CALIS procedure, 913
CANCORR procedure, 1040, 1046
CLUSTER procedure, 1244
FACTOR procedure, 1540, 1559, 1568, 1569
FASTCLUS procedure, 1638, 1643
GENMOD procedure, 2007
KRIGE2D procedure, 2937, 2967, 2968
LIFEREG procedure, 3056
LIFETEST procedure, 3148
LOGISTIC procedure, 3364, 3366–3368
MI procedure, 3753, 3760, 3792
MI procedure, EM statement, 3755
MODECLUS procedure, 4121
MULTTEST procedure, 4220, 4222
OUTCOV= data set (INBREED), 2865, 2875
PHREG procedure, 4630–4632
PRINQUAL procedure, 5229
SIM2D procedure, 6166, 6180
SURVEYLOGISTIC procedure, 6439
SURVEYMEANS procedure, 6509
SURVEYREG procedure, 6574
TCALIS procedure, 6823
TREE procedure, 7369
VARCLUS procedure, 7472, 7477
VARIOGRAM procedure, 7529, 7555, 7565, 7566
output exclusion
ODS, 454
output jackknife coefficient
SURVEYLOGISTIC procedure, 6441
SURVEYMEANS procedure, 6510
SURVEYREG procedure, 6575
output objects
ODS, 434

output ODS Graphics table names
GENMOD procedure, 2022
LIFEREG procedure, 3062
MCMC procedure, 3577
output parameter estimates
MI procedure, 3760
output replicate weights
SURVEYLOGISTIC procedure, 6440
SURVEYMEANS procedure, 6509
SURVEYREG procedure, 6575
output selection
ODS, 451
output statistics
GLIMMIX procedure, 2278
output table names
ACECLUS procedure, 683
CALIS procedure, 967
CANCORR procedure, 1051
CLUSTER procedure, 1268
FASTCLUS procedure, 1652
GENMOD procedure, 2019
INBREED procedure, 2877
KDE procedure, 2904
LIFEREG procedure, 3060
MCMC procedure, 3575
MDS procedure, 3731
MODECLUS procedure, 4127
PRINCOMP procedure, 5162
PRINQUAL procedure, 5237
PROBIT procedure, 5324
QUANTREG procedure, 5400
ROBUSTREG procedure, 5691
SURVEYLOGISTIC procedure, 6447
SURVEYMEANS procedure, 6517
SURVEYREG procedure, 6581
TCALIS procedure, 6939
TREE procedure, 7370
VARCLUS procedure, 7481
output, suppressing
ODS, 447
OUTQ= data set, 4369
OUTSIM= data set
SIM2D procedure, 6180
OUTSTAT= data sets
CANCORR procedure, 1040, 1046
FACTOR procedure, 1569
over-parameterization
MIXED procedure, 3982
overall kappa coefficient, 1765, 1769
overdispersion
GENMOD procedure, 1973
GLIMMIX procedure, 2398
LOGISTIC procedure, 3300, 3342, 3343
PROBIT procedure, 5298
overflows
MCMC procedure, 3565
NLMIXED procedure, 4399
Overlap dissimilarity coefficient
DISTANCE procedure, 1510
overlap of data points
LOGISTIC procedure, 3327
SURVEYLOGISTIC procedure, 6419
Overlap similarity coefficient
DISTANCE procedure, 1510
overlapping β boundaries
SEQDESIGN procedure, 5850
SEQTEST procedure, 6020

P-P plots
REG procedure, 5583
P-spline
GLIMMIX procedure, 2198
p-value
SEQTEST procedure, 6024
p-value scale
SEQDESIGN procedure, 5832
p-value adjustments
adaptive FDR (MULTTEST), 4189
adaptive Hochberg (MULTTEST), 4188
adaptive Holm (MULTTEST), 4189
Bonferroni (MULTTEST), 4189, 4212
bootstrap (MULTTEST), 4185, 4189, 4213, 4229
bootstrap FDR (MULTTEST), 4190
dependent FDR (MULTTEST), 4189
false discovery rate (MULTTEST), 4216
familywise error rate (MULTTEST), 4212
Fisher combination (MULTTEST), 4215
Hochberg (MULTTEST), 4215
Hommel (MULTTEST), 4215
MULTTEST procedure, 4182, 4211
permutation (MULTTEST), 4194, 4213, 4236
permutation FDR (MULTTEST), 4190
positive FDR (MULTTEST), 4194, 4218
Sidak (MULTTEST), 4197, 4213, 4233
painting line-printer plots
REG procedure, 5543
paired comparisons, 7437
TTEST procedure, 7392
paired proportions, *see* McNemar's test
paired *t* test, 7409
power and sample size (POWER), 4871, 4878, 4962, 4963
paired-difference *t* test, *see* paired *t* test
paired-difference *t* test
TTEST procedure, 7392
paired-ratio *t* test

TTEST procedure, 7392
pairwise comparisons
GLM procedure, 2514, 2515
pairwise distance, *see* lag classification (VARIOGRAM)
distribution (VARIOGRAM), 7553
VARIOGRAM procedure, 7522
panels
INBREED procedure, 2875, 2882
plot (VARIOGRAM), 7530, 7581, 7587, 7590
Parallel BY Group Computation
GLMSELECT procedure, 2738
parameter
definition (Introduction to Modeling), 28
parameter constraints
HPMIXED procedure, 2827
MIXED procedure, 3944, 4011
parameter estimates
covariance matrix (CATMOD), 1120
example (REG), 5527
GENMOD procedure, 2016
GLMSELECT procedure, 2747
LIFEREG procedure, 3058
NLMIXED procedure, 4407
PHREG procedure, 4530, 4537, 4631, 4633, 4634
REG procedure, 5585
SEQTEST procedure, 6029
parameter information
PHREG procedure, 4638
parameter rescaling
NLMIXED procedure, 4399
parameter simulation
MI procedure, 3797
parameter specification
NLMIXED procedure, 4376
parameterization
CATMOD procedure, 1124
effect (Shared Concepts), 373
GLIMMIX procedure, 2261
GLM (Shared Concepts), 374
mixed model (GLIMMIX), 2261
mixed model (MIXED), 3981
MIXED procedure, 3981
of models (GLM), 2491
ordinal (Shared Concepts), 374
ortheffect (Shared Concepts), 375
orthordinal (Shared Concepts), 376
orthoterm (Shared Concepts), 376
orthpoly (Shared Concepts), 376
orthref (Shared Concepts), 376
polynomial (Shared Concepts), 374
reference (Shared Concepts), 375
Shared Concepts, 368
SURVEYLOGISTIC procedure, 6412
thermometer (Shared Concepts), 374
parameters specification
MCMC procedure, 3515
parametric discriminant analysis, 1402
parametric functions (TCALIS)
tests, 6814, 6817
Pareto charts, 25
pareto distribution
definition of (MCMC), 3539
MCMC procedure, 3514, 3539
parsimax method, 1534, 1561, 1562, 6755, 6756
parsimonious fit indices, 6761
part-worth utilities
TRANSREG procedure, 7329
partial canonical correlation, 1031
partial correlation
CANCORR procedure, 1040, 1041, 1043
principal components, 5162
partial correlations
multivariate analysis of variance, 2531
power and sample size (POWER), 4840, 4843, 4932, 4933, 5002
partial least squares, 4765, 4766, 4784
partial likelihood
PHREG procedure, 4523, 4524, 4587, 4588, 4590
partial listing
product-limit estimate (LIFETEST), 3124
partial regression leverage plots
REG procedure, 5567
partially balanced square lattice
LATTICE procedure, 2979
partitions
MDS procedure, 3713, 3724
passive observations
PRINQUAL procedure, 5234
TRANSREG procedure, 7255
path diagram (CALIS)
exogenous variables, 944
factor analysis model, 902, 903
structural model example, 835, 903
path diagram (TCALIS)
structural model example, 6705
PATH model
structural model example (TCALIS), 6706
TCALIS procedure, 6799
path names
ODS, 437
paths
ODS, 435
PDF, *see* probability density function
PDF destination

ODS Graphics, 546
Pearson chi-square test, 1731, 1732, 5265, 5313
power and sample size (POWER), 4884, 4889, 4967
Pearson correlation coefficient, 1736, 1740
Pearson correlation statistics
power and sample size (POWER), 4840, 4932, 4933, 5002
Pearson residual
MIXED procedure, 3941
Pearson residuals
GENMOD procedure, 1983, 1984
LOGISTIC procedure, 3350
Pearson's chi-square
GENMOD procedure, 1947, 1970, 1971
LOGISTIC procedure, 3292, 3300, 3343
PROBIT procedure, 5296, 5299, 5314
pedigree analysis
example (INBREED), 2880, 2882
INBREED procedure, 2857, 2858
penalized B-spline
GLIMMIX procedure, 2198
penalized B-spline example
TRANSREG procedure, 7323
penalized B-spline lambda
TRANSREG procedure, 7152
penalized B-spline t-options
TRANSREG procedure, 7152
penalized B-splines
TRANSREG procedure, 7219
penalized least squares, TPSPLINE procedure, 7057–7059, 7079
percentile
SURVEYMEANS procedure, 6504
percentiles
SURVEYMEANS procedure, 6504
weighted, 6233
performance settings
GLMSELECT procedure, 2743
permutation
generating with PLAN procedure, 4755
p-value adjustments (MULTTEST), 4194, 4213, 4236
permutation FDR adjustment
MULTTEST procedure, 4190
permutation test
NPAR1WAY procedure, 4461, 4470
Peto test
MULTTEST procedure, 4201, 4207, 4233
Peto-Peto test for homogeneity
LIFETEST procedure, 3100, 3130
Peto-Peto-Prentice, *see* Peto-Peto test for homogeneity
pFDR, *see* positive false discovery rate
pharmaceutical stability
example (MIXED), 4047
pharmakokinetics example
NLMIXED procedure, 4409
phenogram, 7355
phi coefficient, 1731, 1735
phi-squared coefficient
DISTANCE procedure, 1511
PHREG procedure
Akaike's information criterion, 4610
alpha level, 4544, 4547, 4548, 4555, 4562, 4566
Andersen-Gill model, 4524, 4590, 4606
autocorrelations, 4639
baseline hazard function, 4524
BASELINE statistics, 4541
baseline statistics, 4543, 4544
branch-and-bound algorithm, 4618, 4652
Breslow likelihood, 4572
case weight, 4579
case-control studies, 4524, 4572, 4664
censored values summary, 4633, 4637
coefficient prior, 4638
conditional logistic regression, 4524, 4666
continuous time scale, 4524, 4572, 4668
correlation matrix, 4639
counting process, 4588
covariance matrix, 4537, 4566, 4595, 4639
Cox regression analysis, 4523, 4530
cumulative martingale residuals, 4540, 4619, 4643
DATA step statements, 4530, 4575, 4671
descriptive statistics, 4539
DFBETA statistics, 4573
discrete logistic model, 4524, 4572, 4666
disk space, 4537
displayed output, 4632
effective sample sizes, 4640
Efron likelihood, 4572
equal-tail intervals, 4556, 4639
estimability checking, 4561
event times, 4523, 4528
event values summary, 4633, 4637
exact likelihood, 4572
fit statistics, 4639
fractional frequencies, 4562
Gelman-Rubin diagnostics, 4639
Geweke diagnostics, 4640
global influence, 4574, 4614
global null hypothesis, 4528, 4597, 4634
hazard function, 4523, 4579
hazard ratio confidence intervals, 4566, 4569
hazard ratios, 4528, 4590, 4598, 4640

hazards ratio confidence interval, 4570
Heidelberger-Welch Diagnostics, 4640
hierarchy, 4567
HPD intervals, 4556, 4639
initial values, 4631, 4639
interval estimates, 4639
iteration history, 4568, 4633
Lee-Wei-Amato model, 4605, 4697
left truncation time, 4567, 4589
likelihood displacement, 4574, 4614
likelihood ratio test, 4598, 4634
line search, 4569
linear hypotheses, 4524, 4577, 4600
linear predictor, 4544, 4573, 4574, 4685, 4686
local influence, 4573, 4614
log-hazard, 4591
log-rank test, 4529
Mantel-Haenszel test, 4529
maximum likelihood estimates, 4638
mean function, 4542, 4545, 4594, 4595, 4606, 4608
missing values, 4564, 4576, 4669
missing values as strata, 4577
model assessment, 4540, 4618, 4700
model fit statistics, 4610
model hierarchy, 4567
model information, 4632, 4633, 4637
model selection, 4524, 4565, 4570, 4571, 4617
monotone likelihood, 4567, 4595, 4662
Newton-Raphson algorithm, 4595
number of observations, 4632, 4637
number of subjects at risk, 4573
ODS Graph names, 4643
ODS table names, 4640
offset variable, 4569
output data sets, 4630–4632
OUTPUT statistics, 4573, 4574
parameter estimates, 4530, 4537, 4631, 4633, 4634
parameter information, 4638
partial likelihood, 4523, 4524, 4587, 4588, 4590
piecewise constant baseline hazard model, 4525, 4549, 4621
Prentice-Williams-Peterson model, 4608
programming statements, 4530, 4537, 4575, 4576, 4671
proportional hazards model, 4523, 4530, 4572
Raftery and Lewis diagnostics, 4640
rate function, 4593, 4606
rate/mean model, 4593, 4606
recurrent events, 4524, 4542, 4545, 4593
residual chi-square, 4571
residuals, 4574, 4611–4614, 4685
response variable, 4528, 4666
ridging, 4569
risk set, 4530, 4587, 4588, 4671
risk weights, 4569
Robust Score test, 4598
Robust Wald test, 4598
Schwarz criterion, 4610
score test, 4569, 4571, 4598, 4634, 4646, 4648
selection methods, 4524, 4565, 4570, 4617
singular contrast matrix, 4561
singularity criterion, 4571
standard error, 4573, 4574, 4634
standard error ratio, 4634
standardized score process, 4620, 4643
step halving, 4595
strata variables, 4577
stratified analysis, 4524, 4577
summary statistics, 4639
survival distribution function, 4579
survival times, 4523, 4664, 4666
survivor function, 4523, 4543, 4574, 4579, 4615, 4683, 4684
ties, 4524, 4529, 4572, 4588, 4632, 4637
time intervals, 4638
time-dependent covariates, 4524, 4530, 4537, 4541, 4573, 4575
type 3 test, 4634
Wald test, 4577, 4598, 4600, 4634, 4667
Wei-Lin-Weissfeld model, 4602
piecewise constant baseline hazard model
PHREG procedure, 4525, 4549, 4621
piecewise exponential model, *see* piecewise constant baseline hazard model
piecewise polynomial splines
TRANSREG procedure, 7143, 7264
Pillai's trace, 706, 2464, 2536
Pitman's test
NPAR1WAY procedure, 4461, 4470
PLAN procedure
combinations, 4755
compared to other procedures, 4732
crossover designs, 4759
factor, selecting levels for, 4736, 4737
generalized cyclic incomplete block design, 4754
hierarchical design, 4750
incomplete block design, 4752, 4754
input data sets, 4736, 4739, 4740
introductory example, 4733
Latin square design, 4753

nested design, 4750
ODS table names, 4749
output data sets, 4736, 4739, 4740, 4743, 4744
permutations, 4755
random number generators, 4736
randomizing designs, 4744, 4748
specifying factor structures, 4745
split-plot design, 4749
treatments, specifying, 4742
using interactively, 4743
%PLOTDEN macro
DISCRIM procedure, 1427
%PLOTPROB macro
DISCRIM procedure, 1427
plots
keywords (REG), 5490
likelihood (MIXED), 4038
line printer (REG), 5497, 5534
of configuration (MDS), 3740
of dimension coefficients (MDS), 3740
of linear fit (MDS), 3740
options (REG), 5491, 5493
power and sample size, 421, 423
power and sample size (GLMPOWER), 2642, 2649, 2651, 2654
power and sample size (POWER), 4817, 4826, 4827, 4880, 5012
traditional (REG), 5489, 5547
plotting samples from univariate distributions
MODECLUS procedure, 4128
PLS procedure
algorithms, 4776
centering, 4790
compared to other procedures, 4766
components, 4766
computation method, 4776
cross validation, 4766, 4788, 4789
cross validation method, 4775
examples, 4795
factors, 4766
factors, selecting the number of, 4769
introductory example, 4767
latent variables, 4766
latent vectors, 4766
missing values, 4777
ODS graph names, 4793
ODS table names, 4792
outlier detection, 4803
output data sets, 4783
output keywords, 4783
partial least squares regression, 4765, 4766, 4784
predicting new observations, 4772
principal components regression, 4766, 4785
reduced rank regression, 4766, 4785
scaling, 4790
SIMPLS method, 4785
test set validation, 4789, 4805
Pocock method
SEQDESIGN procedure, 5807, 5827, 5842, 5874
Pocock-type error spending function
SEQDESIGN procedure, 5806
Pocock-type error spending method
SEQDESIGN procedure, 5849
point estimation
Introduction to Bayesian Analysis, 147
point models
TRANSREG procedure, 7255
point pairs
VARIOGRAM procedure, 7522, 7545, 7548
Poisson distribution
definition of (MCMC), 3539
GENMOD procedure, 1966
GLIMMIX procedure, 2168
MCMC procedure, 3514, 3539
NLMIXED procedure, 4376
Poisson mixture
GLIMMIX procedure, 2398
Poisson regression
GENMOD procedure, 1897, 1900
Introduction to Regression, 76
Poisson-normal example
NLMIXED procedure, 4424
POLY parameterization
SURVEYLOGISTIC procedure, 6413
polychoric correlation coefficient, 1711, 1736, 1742
polynomial effect
GLIMMIX procedure, 380
GLMSELECT procedure, 380
polynomial effects
GENMOD procedure, 1974
MIXED procedure, 3982
model parameterization (GLM), 2491
Shared Concepts, 369
specifying (GLM), 2488
polynomial model
GLMMOD procedure, 2621
Introduction to Regression, 77
POLYNOMIAL parameterization
SURVEYLOGISTIC procedure, 6413
polynomial parameterization
Shared Concepts, 374
polynomial regression
REG procedure, 5440

polynomial-spline basis
TRANSREG procedure, 7143, 7264
pooled stratum
SURVEYREG procedure, 6572
pooled within-cluster covariance matrix
definition, 660
population
Introduction to Survey Procedures, 265
profile (CATMOD), 1099
SURVEYSELECT procedure, 6614
population (INBREED)
monoecious, 2877
multiparous, 2864, 2865, 2868
nonoverlapping, 2866
overlapping, 2859, 2860, 2871
population clusters
risks of estimating (MODECLUS), 4116
population profile, 184
positive definiteness
GLIMMIX procedure, 2195
positive false discovery rate, 4211
positive FDR adjustment
MULTTEST procedure, 4194, 4218
posterior distribution
definition of, 142
improper, 144, 146
Introduction to Bayesian Analysis, 142
posterior predictive distribution
definition of (MCMC), 3560
MCMC procedure, 3560
posterior probability
DISCRIM procedure, 1441
error rate estimation (DISCRIM), 1410
posterior summary statistics
correlation, 171
covariance, 171
equal-tail intervals, 171
highest posterior density (HPD) intervals, 171
Introduction to Bayesian Analysis, 170
mean, 170
Monte Carlo standard error (MCSE), 170
percentiles, 171
standard deviation, 170
standard error of the mean estimate, 170
PostScript
graphics image file, 546, 604
ODS Graphics, 546
power, 413
Introduction to Modeling, 49
overview of power concepts, 413
overview of power concepts (POWER), 4916
overview of SAS tools, 414
See GLMPOWER procedure, 2641
See POWER procedure, 4816
SEQDESIGN procedure, 5878
SEQTEST procedure, 6011, 6029
simulation, 417, 429
solving for, 5088
Power and Sample Size application
compared to other power and sample size tools, 414, 415
narratives, 425
power by sample size graph, 5063
customizing, 5058
power covariance structure
GLIMMIX procedure, 2202
power curves, *see* plots
power curves plot
SEQDESIGN procedure, 5803
power distance coefficient
DISTANCE procedure, 1509
power error spending function
SEQDESIGN procedure, 5806
power error spending method
SEQDESIGN procedure, 5850
power family method
SEQDESIGN procedure, 5807, 5828, 5843, 5875
power plot
SEQDESIGN procedure, 5882
SEQTEST procedure, 6033
POWER procedure
AB/BA crossover designs, 4995
actual alpha, 4924
actual power, 4825, 4922, 4924
actual prob(width), 4924
analysis of variance, 4859, 4863, 4957, 4981
analysis statements, 4826
bar (|) operator, 4920
binomial proportion confidence interval, 4949–4952
binomial proportion confidence interval precision, 4852
binomial proportion tests, 4844, 4850, 4934, 4935, 4937, 4986
ceiling sample size, 4825, 4924
compared to other power and sample size tools, 414
compared to other procedures, 2643, 4818
computational methods, 4926
computational resources, 4926
confidence intervals for means, 4852, 4859, 4871, 4879, 4891, 4900, 4956, 4965, 4974, 5009
contrasts, analysis of variance, 4859, 4860, 4863, 4957, 4981

correlated proportions, 4864, 4869, 4871, 4960
correlation, 4840, 4932, 4933, 5002
crossover designs, 4995
displayed output, 4924
effect size, 4882
equivalence tests, 4851, 4852, 4858, 4871, 4879, 4891, 4900, 4955, 4964, 4973, 4974, 4995
Fisher's exact test, 4884, 4890, 4969
Fisher's *z* test for correlation, 4840, 4843, 4932, 5002
fractional sample size, 4825, 4924
Gehan test, 4901, 4910, 4975
grouped-name-lists, 4919
grouped-number-lists, 4919
introductory example, 4818
keyword-lists, 4919
likelihood-ratio chi-square test, 4884, 4890, 4968
log-rank test for comparing survival curves, 4901, 4909, 4975, 5007
logistic regression, 4827, 4928, 5037
lognormal data, 4854, 4858, 4873, 4878, 4879, 4893, 4899, 4900, 4954, 4955, 4963, 4964, 4972, 4974, 4999
McNemar's test, 4864, 4869, 4871, 4960
name-lists, 4919
nominal power, 4825, 4922, 4924
noninferiority tests, 4851, 4999
notation for formulas, 4926
number-lists, 4919
odds ratio, 4884, 4889, 4966, 4967
ODS table names, 4925
one-sample *t* test, 4818, 4852, 4857, 4858, 4953, 4954
one-way ANOVA, 4859, 4863, 4957, 4981
overview of power concepts, 4916
paired proportions, 4864, 4869, 4871, 4960
paired *t* test, 4871, 4878, 4962, 4963
partial correlation, 4840, 4843, 4932, 4933, 5002
Pearson chi-square test, 4884, 4889, 4967
Pearson correlation, 4840, 4843, 4932, 4933, 5002
plots, 4817, 4826, 4827, 4880, 5012
regression, 4835, 4839, 4930, 5002
relative risk, 4884, 4889, 4966, 4967
sample size adjustment, 4922
summary of analyses, 414, 4916
summary of statements, 4826
superiority tests, 4852
survival analysis, 4901, 4909, 4975
t test for correlation, 4840, 4843, 4933
t tests, 4852, 4857, 4858, 4871, 4878, 4891, 4898, 4953, 4954, 4962, 4969, 4972, 5012
Tarone-Ware test, 4901, 4910, 4975
two-sample *t* test, 4821, 4891, 4898, 4899, 4969, 4971, 4972, 5012
value lists, 4919
Wilcoxon-Mann-Whitney (rank-sum) test, 4911, 4915, 4978
Wilcoxon-Mann-Whitney test, 5039
z test, 4844, 4850, 4935, 4937
power semivariogram model
KRIGE2D procedure, 2952
power-of-the-mean model
MIXED procedure, 3956
%POWTABLE macro, 424
compared to other power and sample size tools, 414
PPC convergence measure, 4308
pplot plots
annotating, 3028
axes, color, 3028
font, specifying, 3029
reference lines, options, 3028–3030, 3032–3036
PPS sampling
SURVEYSELECT procedure, 6614, 6645
PPS sampling, with replacement
SURVEYSELECT procedure, 6650
PPS sampling, without replacement
SURVEYSELECT procedure, 6648
PPS sequential sampling
SURVEYSELECT procedure, 6651
PPS systematic sampling
SURVEYSELECT procedure, 6650
precision
NLMIXED procedure, 4402
precision of solution
MCMC procedure, 3568
precision, confidence intervals, 416, 4916
predicted covariance matrix
displaying (TCALIS), 6936
TCALIS procedure, 6836
predicted covariance model matrix
singular (TCALIS), 6933
predicted mean vector
displaying (TCALIS), 6936
predicted means
MIXED procedure, 3941
predicted model matrix
CALIS procedure, 922, 924, 943
displaying (CALIS), 963
singular (CALIS), 960
predicted population margins

GLM procedure, 2458
predicted probabilities
LOGISTIC procedure, 3339
SURVEYLOGISTIC procedure, 6438
predicted probability plots
annotating, 5304
axes, color, 5304
font, specifying, 5305
options summarized by function, 5301
reference lines, options, 5304–5306, 5308
threshold lines, options, 5307
predicted residual sum of squares
RSREG procedure, 5733
predicted value confidence intervals
MIXED procedure, 3929
predicted values
example (MIXED), 4032
LIFEREG procedure, 3043
mixed model (MIXED), 3922
MIXED procedure, 3940
NLIN procedure, 4287
REG procedure, 5526, 5531
response functions (CATMOD), 1124
prediction
example (REG), 5608
KRIGE2D procedure, 2930, 2931
NLMIXED procedure, 4377, 4403
VARIOGRAM procedure, 7519
predictive mean matching method
MI procedure, 3775
predictive power
SEQTEST procedure, 6009, 6022, 6030
predpplot
PROBIT procedure, 5300
preference mapping
TRANSREG procedure, 7180, 7345
preference models
TRANSREG procedure, 7172
Preferences window, 5072
preferences, setting, 5072
prefix name
LINEQS statement (CALIS), 882
LINEQS statement (TCALIS), 6768
MATRIX statement (CALIS), 884
MATRIX statement (TCALIS), 6787
RAM statement (CALIS), 900
STD statement (CALIS), 906
STD statement (TCALIS), 6815
preliminary clusters
definition (CLUSTER), 1252
using in CLUSTER procedure, 1242
preliminary data analysis, *see* exploratory data analysis
Prentice-Williams-Peterson model
PHREG procedure, 4608
PRESS residual
MIXED Procedure, 3990
PRESS statistic, 2478
MIXED Procedure, 3990
RSREG procedure, 5733
prevalence test
MULTTEST procedure, 4207, 4233
Primal-Dual with Predictor-Corrector algorithm
QUANTREG procedure, 5388
primary sampling units (PSUs)
Introduction to Survey Procedures, 266
SURVEYFREQ procedure, 6310
SURVEYLOGISTIC procedure, 6423
SURVEYMEANS procedure, 6493
SURVEYREG procedure, 6562
principal component analysis, 1534
compared with common factor analysis, 1535
PRINQUAL procedure, 5228
with FACTOR procedure, 1537
principal components, *see also* PRINCOMP procedure
definition, 5143
interpreting eigenvalues, 5147
partialing out variables, 5157
properties of, 5144, 5145
regression (PLS), 4766, 4785
rotating, 5161
using weights, 5158
principal factor analysis
with FACTOR procedure, 1538
PRINCOMP procedure
CALIS procedure, 847
computational resources, 5161
correction for means, 5152
Crime Rates Data, example, 5145
DATA= data set, 5159
eigenvalues and eigenvectors, 5144, 5160, 5162
examples, 5163, 5164
input data set, 5152
ODS Graph names, 5163
ODS graph names, 5163
output data sets, 5152, 5158–5160
output table names, 5162
OUTSTAT= data set, 5159
replace missing values, example, 5168
SCORE procedure, 5161
suppressing output, 5152
weights, 5158
PRINQUAL procedure
biplot, 5238
casewise deletion, 5206

character OPSCORE variables, 5233
constant transformations, avoiding, 5232
constant variables, 5233
excluded observations, 5210, 5234
frequency variable, 5211
identity transformation, 5216
iterations, 5209, 5227, 5233
knots, 5218, 5219
linear transformation, 5215
MAC method, 5221, 5229
maximum average correlation method, 5221, 5229
maximum total variance method, 5220
MDPREF analysis, 5238
MGV method, 5221
minimum generalized variance method, 5221
missing character values, 5215
missing values, 5206, 5227, 5234
monotonic B-spline transformation, 5215
monotonic transformation, 5215, 5216
MTV method, 5220
multidimensional preference analysis, 5238
nonoptimal transformations, 5214
ODS graph names, 5237
optimal scoring, 5215
optimal transformations, 5215
output data sets, 5229
output table names, 5237
passive observations, 5234
principal component analysis, 5228
random initializations, 5233
reflecting the transformation, 5219
renaming variables, 5219
reusing variables, 5219
smoothing spline transformation, 5216
spline t-options, 5218
spline transformation, 5215
standardization, 5232
transformation options, 5216
variable names, 5231
weight variable, 5220
printing, 5065
prior density
MIXED procedure, 3946
prior distribution
conjugate, 146
data-set-dependent (MCMC), 3614
definition of, 142
diffuse, 144
distribution specification (MCMC), 3511, 3516
flat, 144
hyperprior specification (MCMC), 3511, 3516
improper, 145
informative, 146
Introduction to Bayesian Analysis, 142, 144
Jeffreys' prior, 146
noninformative, 144, 147
objective, 144
predictive distribution (MCMC), 3560, 3561
subjective, 144
user-defined (MCMC), 3514, 3541
vague, 144
prior event probability
LOGISTIC procedure, 3298, 3342, 3419
probability density function
LIFETEST procedure, 3100, 3183
probability distribution
built-in (GENMOD), 1898, 1948
exponential family (GENMOD), 1964
user-defined (GENMOD), 1942
probability distributions
GLIMMIX procedure, 2168
probability sampling
Introduction to Survey Procedures, 259
SURVEYSELECT procedure, 6614
probit analysis
insets, 5278
probit equation, 5258, 5313
probit model
SURVEYLOGISTIC procedure, 6425
PROBIT procedure
Abbot's formula, 5309
binary response data, 5258, 5259, 5313
cdfplot, 5268
deviance, 5299, 5314
deviance statistic, 5313
dispersion parameter, 5314
extreme value distribution , 5311
goodness-of-fit, 5296, 5299
goodness-of-fit tests, 5265, 5296, 5313
inset, 5277
inverse confidence limits, 5315
ippplot, 5279
log-likelihood function, 5310
logistic distribution, 5311
lpredplot, 5287
maximum-likelihood estimates, 5258
missing values, 5309
models, 5313
multilevel response data, 5258, 5259, 5313
natural response rate, 5259
Newton-Raphson algorithm, 5310
normal distribution, 5311
ODS graph names, 5328

output table names, 5324
overdispersion, 5298
Pearson chi-square, 5313
Pearson's chi-square, 5296, 5299, 5314
predpplot, 5300
subpopulation, 5296, 5299, 5315
threshold response rate, 5259
tolerance distribution, 5315
probit-normal-binomial example
NLMIXED procedure, 4415
probit-normal-ordinal example
NLMIXED procedure, 4419
PROC GLIMMIX procedure
residual variance tolerance, 2120
Procrustes method, 1534
Procrustes rotation, 1562
producing monotone missingness
MI procedure, 3785
product-limit curves
LIFETEST procedure, 3105
product-limit estimates
LIFETEST procedure, 3100, 3103, 3131, 3150
profile likelihood confidence intervals
GENMOD procedure, 1980
profile, population and response, 184
CATMOD procedure, 1099
profiling residual variance
HPMIXED procedure, 2813
MIXED procedure, 4010
progeny
INBREED procedure, 2868, 2870, 2872, 2873, 2880
programming statements
constraints (CALIS), 845, 903, 905, 955
constraints (TCALIS), 6814, 6882
differentiation (CALIS), 854
GENMOD procedure, 1957
GLIMMIX procedure, 2207
MCMC procedure, 3516
NLMIXED procedure, 4379
PHREG procedure, 4530, 4537, 4575, 4576
Shared Concepts, 410
projected gradient
NLMIXED procedure, 4395
projected Hessian
NLMIXED procedure, 4395
promax method, 1534, 1562
propensity score method
MI procedure, 3776, 3799
proportion estimation
SURVEYMEANS procedure, 6500
proportional allocation
SURVEYSELECT procedure, 6641, 6654, 6673
proportional hazard
Introduction to Regression, 77
proportional hazards model
assumption (PHREG), 4530
distribution (LIFEREG), 3039
PHREG procedure, 4523, 4572
proportional odds model
SURVEYLOGISTIC procedure, 6424
proportional rates/means model, *see* rate/mean model
proportions, 5049
proposal distribution
MCMC procedure, 3525
prospective power, 414, 2642, 4817
prospective study, 1817
proximity data
MDS procedure, 3704, 3713, 3720
proximity measures
available methods for computing (DISTANCE), 1494
formulas(DISTANCE), 1507
pseudo *F* and *t* statistics
CLUSTER procedure, 1246
pseudo-likelihood
GLIMMIX procedure, 2102, 2273
pseudo-semivariance
VARIOGRAM procedure, 7546
pseudo-semivariogram
VARIOGRAM procedure, 7546, 7587
PSS, 5048
available analyses, 5049
available platforms, 5085
features, 5049
installation, 5085
local and remote configurations, 5085
Preferences window, 5072
Results page, 5080
software requirements, 5085

Q-Q plots
REG procedure, 5583
quadratic discriminant function, 1384
quadratic forms
Introduction to Modeling, 60
quadratic forms for fixed effects
displaying (GLM), 2481
quadratic regression, 2438
quadrature approximation
GLIMMIX procedure, 2102
theory (GLIMMIX), 2227
quadrature options
NLMIXED procedure, 4369

qualitative variables, 183, *see* classification variables, *see* classification variables
REG procedure, 5633
quantal response data, 5258
quantification method
CORRESP procedure, 1320
quantile
SURVEYMEANS procedure, 6504
quantile computation
STDIZE procedure, 6212, 6232
quantiles
SURVEYMEANS procedure, 6504
QUANTREG procedure, 5358
affine step, 5388
centering step, 5388
complementarity, 5387
computational resources, 5399
INEST= data sets, 5398
infeasibility, 5387
Karush-Kuhn-Tucker (KKT) conditions, 5387
ODS graph names, 5405
OUTEST= data sets, 5399
output table names, 5400
Primal-Dual with Predictor-Corrector algorithm, 5388
QUANTTREG procedure
syntax, 5372
quartimax method, 1534, 1561, 1562, 6755, 6756
quartimin method, 1534, 1562, 6756
quasi inverse, 1408
quasi-complete separation
LOGISTIC procedure, 3327
SURVEYLOGISTIC procedure, 6419
quasi-independence model, 1195
quasi-likelihood
functions (GENMOD), 1994
GENMOD procedure, 1973
GLIMMIX procedure, 2273
quasi-likelihood information criterion (GENMOD), 1993
quasi-Newton, 4371
quasi-Newton algorithm
CALIS procedure, 856, 857, 868, 870, 945, 947
TCALIS procedure, 6731, 6737, 6738, 6923, 6925
quasi-Newton method
Shared Concepts, 407

R convergence measure, 4308
R matrix
HPMIXED procedure, 2832
MIXED procedure, 3894, 3954, 3957, 3969, 3970
R-notation, 2498
R-side random effect
GLIMMIX procedure, 2083
R-square
definition (Introduction to Modeling), 64
definition (Introduction to Regression), 98, 110
R-square statistic
CLUSTER procedure, 1246
LOGISTIC procedure, 3300, 3330
SURVEYLOGISTIC procedure, 6401, 6421
R^2 improvement
REG procedure, 5524, 5525
R^2 selection
REG procedure, 5525
R= option
and other options (CLUSTER), 1242, 1246
radial smoother structure
GLIMMIX procedure, 2199
radial smoothing
GLIMMIX procedure, 2189, 2199, 2250
radius of sphere of support, 1246
Raftery and Lewis diagnostics
Bayesian analysis (PHREG) procedure, 4640
RAM model
CALIS procedure, 832, 899
specification, 840
structural model example (CALIS), 837, 842
structural model example (TCALIS), 6707
TCALIS procedure, 6805
random coefficients
example (MIXED), 4025, 4047
random effects
expected mean squares, 2540
GLIMMIX procedure, 2083, 2186
GLM procedure, 2480, 2540
HPMIXED procedure, 2829
MIXED procedure, 3894, 3949
NESTED procedure, 4259
NLMIXED procedure, 4378
VARCOMP procedure, 7493, 7500
random effects model, *see also* nested design
VARCOMP procedure, 7501
random initializations
TRANSREG procedure, 7252
random number generators
MI procedure, 3753
PLAN procedure, 4736
random sampling
SURVEYSELECT procedure, 6614
random-effects models

MCMC procedure, 3614
random-effects parameters
MIXED procedure, 3893, 3969
randomization assumption
VARIOGRAM procedure, 7534, 7562
randomization of designs
using PLAN procedure, 4748
randomized complete block design
example, 2556
range
correlation (KRIGE2D), 2932
effective (KRIGE2D), 2932, 2950, 2951
effective (VARIOGRAM), 7541
practical (KRIGE2D), 2932, 2950, 2951
practical (VARIOGRAM), 7541
VARIOGRAM procedure, 7540
range ϵ
KRIGE2D procedure, 2950
rank correlation
LOGISTIC procedure, 3338
SURVEYLOGISTIC procedure, 6437
rank order typal analysis, *see* complete linkage
RANK procedure, 24
order statistics, 24
rank scores, 1731
Introduction to Nonparametric Analysis, 294, 297
NPAR1WAY procedure, 4470
rank tests
NPAR1WAY procedure, 4467
Rank-sum test, *see* Wilcoxon-Mann-Whitney (rank-sum) test
Rao-Scott chi-square test
SURVEYFREQ procedure, 6346
Rao-Scott likelihood ratio test
SURVEYFREQ procedure, 6350
rate function
PHREG procedure, 4593, 4606
rate/mean model
PHREG procedure, 4593, 4606
ratio analysis
SURVEYMEANS procedure, 6486, 6497
ratio level of measurement
DISTANCE procedure, 1484
ratio test
TTEST procedure, 7407
ratios
SURVEYMEANS procedure, 6486, 6497
raw residuals
GENMOD procedure, 1983
receiver operating characteristic, *see* ROC curve
reciprocal averaging
CORRESP procedure, 1320
reciprocal causation
CALIS procedure, 863
TCALIS procedure, 6729, 6915
rectangular lattice
LATTICE procedure, 2979
rectangular table
SURVEYMEANS procedure, 6476, 6510
recurrent events
PHREG procedure, 4524, 4542, 4545, 4593
reduced rank regression, 4766
PLS procedure, 4785
reduction notation, 2498
redundancy analysis
CANCORR procedure, 1031
TRANSREG procedure, 7162, 7177, 7180, 7256
REF parameterization
SURVEYLOGISTIC procedure, 6413
reference category
GLIMMIX procedure, 2267
reference improvement
SEQDESIGN procedure, 5826
reference level
TRANSREG procedure, 7155
reference lines
ODS Graphics, 637
REFERENCE parameterization
SURVEYLOGISTIC procedure, 6413
reference parameterization
Shared Concepts, 375
reference structure, 1541
reference-cell coding
TRANSREG procedure, 7155, 7181, 7233, 7240
references, 1531
refitting models
REG procedure, 5570
reflecting the transformation
PRINQUAL procedure, 5219
TRANSREG procedure, 7158
REG procedure
adding variables, 5467
adjusted R^2 selection, 5525
alpha level, 5454
annotations, 5454, 5493
ANOVA table, 5584
autocorrelation, 5581
backward elimination, 5435, 5524
collinearity, 5555
compared to other procedures, 2434, 4505
computational methods, 5584
correlation matrix, 5454
covariance matrix, 5454
crossproducts matrix, 5584
delete variables, 5468

deleting observations, 5569
diagnostic statistics, 5557, 5558
dictionary of options, 5493
fit diagnostics, 5592
forward selection, 5435, 5523
graphics keywords and options, 5490, 5491
graphics plots, traditional, 5489
heteroscedasticity, testing, 5575
hypothesis tests, 5480, 5506
incomplete principal components, 5456, 5477
influence diagnostics, 5592
influence statistics, 5559
input data sets, 5508
interactive analysis, 5450, 5519
introductory example, 5436
IPC analysis, 5456, 5477, 5583
lack of fit, 5642
lack-of-fit, testing, 5576
line printer plots, 5497, 5534
Mallows' C_p selection, 5526
missing values, 5507
model building, 5592
model fit summary statistics, 5557
model selection, 5435, 5523, 5526, 5527, 5608
multicollinearity, 5555
multivariate tests, 5577
new regressors, 5508
non-full-rank models, 5553
ODS graph names, 5591
ODS GRAPHICS, 5592
ODS table names, 5587
output data sets, 5512, 5518
P-P plots, 5583
painting line-printer plots, 5543
parameter estimates, 5527, 5585
partial regression leverage plots, 5567
plot keywords and options, 5490, 5491, 5493
plots, traditional, 5489
polynomial regression, 5440
predicted values, 5526, 5531, 5608
Q-Q plots, 5583
qualitative variables, 5633
R^2 improvement, 5524, 5525
R^2 selection, 5525
refitting models, 5570
residual values, 5531
restoring weights, 5571
reweighting observations, 5569
ridge regression, 5466, 5477, 5496, 5583, 5638
singularities, 5584
stepwise selection, 5435, 5524
summary statistics, 5557
sweep algorithm, 5584
time series data, 5581
traditional graphics, 5547
variance inflation factors (VIF), 5456
regression
MODEL statements (GLM), 2489
adj. R-square selection (Introduction to Regression), 87
adjusted R-square (Introduction to Regression), 98
analysis (REG), 5434
assumptions (Introduction to Regression), 87, 95
backward elimination (Introduction to Regression), 86
Bayesian analysis (Introduction to Regression), 89, 90
breakdown value (Introduction to Regression), 94
canonical correlation (Introduction to Regression), 78, 94
CATMOD procedure, 1095
collinearity (Introduction to Regression), 109
collinearity diagnostics (Introduction to Regression), 85
conditional logistic (Introduction to Regression), 90
confidence interval (Introduction to Regression), 101
conjoint analysis (Introduction to Regression), 78, 94
contingency table (Introduction to Regression), 76
controlled experiment (Introduction to Regression), 108
Cook's D (Introduction to Regression), 83
correlation matrix (Introduction to Regression), 97
covariance matrix (Introduction to Regression), 97
Cox model (Introduction to Regression), 77
Cp selection (Introduction to Regression), 87
diagnostics (Introduction to Regression), 77, 83
diagnostics, collinearity (Introduction to Regression), 85
diagnostics, influence (Introduction to Regression), 85
diagnostics, logistic (Introduction to Regression), 77

errors-in-variable (Introduction to Regression), 110
estimate of precision (Introduction to Regression), 97
exact conditional logistic (Introduction to Regression), 90
examples (GLM), 2562
failure-time data (Introduction to Regression), 77
forecasting (Introduction to Regression), 101
forward selection (Introduction to Regression), 86
function approximation (Introduction to Regression), 92
GEE (Introduction to Regression), 76, 89, 90
general linear model (Introduction to Regression), 77
generalized additive model (Introduction to Regression), 76, 93
generalized estimating equations (Introduction to Regression), 76, 89, 90
generalized least squares (Introduction to Regression), 78
generalized linear mixed model (Introduction to Regression), 77, 89
generalized linear model (Introduction to Regression), 76, 88, 89, 93
generalized logit (Introduction to Regression), 89, 90
Gentleman-Givens algorithm (Introduction to Regression), 77, 90
gompit (Introduction to Regression), 90
heterogeneous conditional distribution (Introduction to Regression), 91
homoscedasticity (Introduction to Regression), 87
Hotelling Lawley trace (Introduction to Regression), 104
ideal point preference mapping (Introduction to Regression), 78, 94
ill-conditioned data, 4505
ill-conditioned data (Introduction to Regression), 90
influence diagnostics (Introduction to Regression), 85
intercept (Introduction to Regression), 98
lack-of-fit (Introduction to Regression), 85, 88
least trimmed squares (Introduction to Regression), 94
leverage (Introduction to Regression), 83, 100
linear (Introduction to Regression), 78
linear mixed model (Introduction to Regression), 77
linear, survey data (Introduction to Regression), 78
local (Introduction to Regression), 77, 92, 93
logistic (Introduction to Regression), 76, 78
logistic, conditional (Introduction to Regression), 90
logistic, exact conditional (Introduction to Regression), 90
LTS estimation (Introduction to Regression), 94
M estimation (Introduction to Regression), 78, 94
max R-square selection (Introduction to Regression), 86
min R-square selection (Introduction to Regression), 87
MN estimation (Introduction to Regression), 94
model selection, adj. R-square (Introduction to Regression), 87
model selection, backward (Introduction to Regression), 86
model selection, Cp (Introduction to Regression), 87
model selection, forward (Introduction to Regression), 86
model selection, max R-square (Introduction to Regression), 86
model selection, min R-square (Introduction to Regression), 87
model selection, R-square (Introduction to Regression), 87
model selection, stepwise (Introduction to Regression), 86
multivariate tests (Introduction to Regression), 102, 104
nonlinear (Introduction to Regression), 77, 91
nonlinear least squares (Introduction to Regression), 77, 92
nonlinear mixed model (Introduction to Regression), 77
nonparametric, 7057
nonparametric (Introduction to Regression), 76, 92
normal equations (Introduction to Regression), 96
observational study (Introduction to Regression), 108
orthogonal regressors (Introduction to Regression), 109
ORTHOREG procedure, 4505

partial least squares (Introduction to Regression), 78
partial least squares (PROC PLS), 4766, 4784
Pillai's trace (Introduction to Regression), 104
Poisson (Introduction to Regression), 76
polynomial (Introduction to Regression), 77
power and sample size (POWER), 4835, 4839, 4930, 5002
precision, estimate (Introduction to Regression), 97
predicted value (Introduction to Regression), 100
prediction interval (Introduction to Regression), 101
principal components (Introduction to Regression), 78
principal components (PROC PLS), 4766, 4785
probit (Introduction to Regression), 78, 90
proportional hazard (Introduction to Regression), 77
proportional odds model (Introduction to Regression), 90
quadratic (GLM), 2438
quantal (Introduction to Regression), 90
quantile (Introduction to Regression), 78, 91
R-square (Introduction to Regression), 98, 110
R-square selection (Introduction to Regression), 87
R-square, adjusted (Introduction to Regression), 98
raw residual (Introduction to Regression), 100
reduced rank (PROC PLS), 4766, 4785
redundancy analysis (Introduction to Regression), 78, 94
regressor variable (Introduction to Regression), 79
residual (Introduction to Regression), 100
residual plot (Introduction to Regression), 82
residual variance (Introduction to Regression), 97
residual, raw (Introduction to Regression), 100
residual, studentized (Introduction to Regression), 100
response surface (Introduction to Regression), 78, 88
ridge (Introduction to Regression), 88
robust (Introduction to Regression), 78, 93
Roy's maximum root (Introduction to Regression), 104
S estimation (Introduction to Regression), 94
semiparametric model (Introduction to Regression), 93
semiparametric models, 7058, 7060
smoothing splines, 7057
spline (Introduction to Regression), 93
spline transformation (Introduction to Regression), 78, 94
spline, basis function(Introduction to Regression), 93
standard error of prediction (Introduction to Regression), 100
standard error, estimated (Introduction to Regression), 97
stepwise selection (Introduction to Regression), 86
stratification (Introduction to Regression), 90
studentized residual (Introduction to Regression), 100
sum of squares, Type I (Introduction to Regression), 97, 102
sum of squares, Type II (Introduction to Regression), 97, 102
surface (Introduction to Regression), 77
survey data (Introduction to Regression), 78, 90
survival data (Introduction to Regression), 77
testing hypotheses (Introduction to Regression), 102
transformation (Introduction to Regression), 78, 94
Type I sum of squares (Introduction to Regression), 97, 102
Type II sum of squares (Introduction to Regression), 97, 102
variance inflation (Introduction to Regression), 98
Wilk's Lambda (Introduction to Regression), 103
regression coefficients
CANCORR procedure, 1039
SURVEYREG procedure, 6563
using with SCORE procedure, 5760
regression diagnostics
LOGISTIC procedure, 3349
regression effects
MIXED procedure, 3982
model parameterization (GLM), 2491
Shared Concepts, 369

specifying (GLM), 2488
regression estimator
SURVEYREG procedure, 6595
regression estimators
SURVEYREG procedure, 6588
regression functions, separate
TRANSREG procedure, 7213
regression method
MI procedure, 3774, 3798
regression parameter estimates, example
SCORE procedure, 5775
regression parameters
SURVEYLOGISTIC procedure, 6423
regression table
TRANSREG procedure, 7166
regression tests
SEQDESIGN procedure, 5868
regressor effects
GENMOD procedure, 1974
regressor variable
Introduction to Regression, 79
rejection repeated confidence intervals
SEQTEST procedure, 6023
rejection sampling
MIXED procedure, 3947
relative cumulative error spending
SEQDESIGN procedure, 5806
relative efficiency
MI procedure, 3795
MIANALYZE procedure, 3857
relative increase in variance
MI procedure, 3795
MIANALYZE procedure, 3856
relative risk, 1774
cohort studies, 1761
logit estimate, 1775
Mantel-Haenszel estimate, 1775
power and sample size (POWER), 4884, 4889, 4966, 4967
relative risks
SURVEYFREQ procedure, 6345
REML, *see* restricted maximum likelihood
Remote monitoring
Shared concepts, 403
remote monitoring
GLIMMIX procedure, 401, 403
renaming and reusing variables
PRINQUAL procedure, 5219
renaming parameters
TCALIS procedure, 6812
repeated confidence intervals
SEQTEST procedure, 6010, 6012, 6022, 6030, 6034
repeated confidence intervals plot
SEQTEST procedure, 6034
repeated measures
ANOVA procedure, 715
CATMOD procedure, 1094, 1129, 1150
contrasts (GLM), 2483
data organization (GLM), 2533
doubly multivariate design, 2598
examples (CATMOD), 1202, 1206, 1208, 1212
examples (GLM), 2485, 2590
GEE (GENMOD), 1895, 1986
GLM procedure, 2481, 2532
hypothesis tests (GLM), 2535, 2537
MIXED procedure, 3893, 3954, 4019
more than one factor (ANOVA), 715, 719
more than one factor (GLM), 2537
multiple populations (CATMOD, 1152
one population (CATMOD), 1151
RESPONSE statement (CATMOD), 1150
specifying factors (CATMOD), 1129
transformations, 2537–2539
repeated significance test
SEQDESIGN procedure, 5827
replicate subjects
NLMIXED procedure, 4379
replicate weights
SURVEYFREQ procedure, 6310
SURVEYLOGISTIC procedure, 6406, 6427
SURVEYMEANS procedure, 6488
SURVEYREG procedure, 6558, 6564
replicated sampling
SURVEYSELECT procedure, 6634, 6664
replication, *see* replicated sampling
replication methods
SURVEYLOGISTIC procedure, 6385, 6427
SURVEYMEANS procedure, 6479, 6506, 6528
SURVEYREG procedure, 6545, 6564, 6606
replication-based variance estimation
SURVEYFREQ procedure, 6327
resampled data sets
MULTTEST procedure, 4222
residual
Cook's D (Introduction to Regression), 83
raw (Introduction to Regression), 83
studentized (Introduction to Regression), 83
residual chi-square
PHREG procedure, 4571
residual likelihood
GLIMMIX procedure, 2102
HPMIXED procedure, 2812
residual maximum likelihood, *see also* restricted maximum likelihood
residual maximum likelihood (REML)

MIXED procedure, 3975, 4016
residual plots
GLIMMIX procedure, 2282
MIXED procedure, 4004
residual variance tolerance
HPMIXED procedure, 2814
residual-based sandwich estimators
GLIMMIX procedure, 2243
residuals
and partial correlation (PRINCOMP), 5159
CALIS procedure, 929, 6901
Cholesky (Introduction to Modeling), 71
deletion (Introduction to Modeling), 71
deviance (Introduction to Modeling), 71
deviance (PHREG), 4574, 4611, 4685
externally studentized (Introduction to Modeling), 70
fitted (Introduction to Modeling), 69
GENMOD procedure, 1952, 1983, 1984
internally studentized (Introduction to Modeling), 70
leave-one-out (Introduction to Modeling), 71
LOGISTIC procedure, 3350
martingale (PHREG), 4574, 4685
MDS procedure, 3718, 3723, 3724, 3727, 3740
NLIN procedure, 4287
partial correlation (PRINCOMP), 5157
Pearson-type (Introduction to Modeling), 70
prefix (CALIS), 881
PRESS (Introduction to Modeling), 71
projected, (Introduction to Modeling), 69
raw (Introduction to Modeling), 69
raw (Introduction to Regression), 100
REG procedure, 5531
scaled (Introduction to Modeling), 70
Schoenfeld (PHREG), 4574, 4612, 4613
score (PHREG), 4574, 4612
standardized (Introduction to Modeling), 70
studentized (Introduction to Modeling), 70
studentized (Introduction to Regression), 100, 101
studentized, external (Introduction to Modeling), 70
studentized, internal (Introduction to Modeling), 70
weighted Schoenfeld (PHREG), 4574, 4613
weighted score (PHREG), 4614
residuals, details
MIXED procedure, 3986
response functions (CATMOD)
covariance matrix, 1120
formulas, 1168
identifying with FACTORS statement, 1114
predicted values, 1124
related to design matrix, 1154, 1156
variance formulas, 1168, 1169
response level ordering
GAM procedure, 1843
GLIMMIX procedure, 2163, 2267
LOGISTIC procedure, 3268, 3288, 3318
SURVEYLOGISTIC procedure, 6396, 6411
response profile, 184
CATMOD procedure, 1099
GLIMMIX procedure, 2267, 2274
response surfaces, 5719
canonical analysis, interpreting, 5737
covariates, 5741
experiments, 5735
plotting, 5738
ridge analysis, 5737
response variable, 720, 2488
PHREG procedure, 4528, 4575, 4666
sort order of levels (GENMOD), 1925
response variable options
GAM procedure, 1843
GLIMMIX procedure, 2163
restoring weights
REG procedure, 5571
restricted maximum likelihood
GLIMMIX procedure, 2273
HPMIXED procedure, 2812
MIXED procedure, 3893, 3975, 4016
VARCOMP procedure, 7498
restrictions
of parameters (CATMOD), 1137
resubstitution
DISCRIM procedure, 1407
Results page, 5080
Results Window
ODS Graphics, 537
Results window
ODS, 440
reticular action model, *see* RAM model
retrospective power, 414, 2642, 4817
reverse response level ordering
GAM procedure, 1843
GLIMMIX procedure, 2163
LOGISTIC procedure, 3318
SURVEYLOGISTIC procedure, 6396, 6411
reweighting observations
REG procedure, 5569
ridge analysis
RSREG procedure, 5737
ridge regression
REG procedure, 5466, 5477, 5496, 5583, 5638

ridging
MIXED procedure, 3915, 3975
PHREG procedure, 4569
ridit scores, 1731
risk difference
SURVEYFREQ procedure, 6342
risk set
PHREG procedure, 4530, 4587, 4588, 4671
risk weights
PHREG procedure, 4569
risks
SURVEYFREQ procedure, 6342
risks and risk differences, 1752
RMSSTD statement
and FREQ statement (CLUSTER), 1248, 1249
robust
cluster analysis, 1622, 1637
estimators (STDIZE), 6230
Robust Score test
PHREG procedure, 4598
Robust Wald test
PHREG procedure, 4598
ROBUSTREG procedure, 5648
computational resources, 5690
INEST= data sets, 5689
ODS graph names, 5696
OUTEST= data sets, 5690
output table names, 5691
ROC curve
comparing (LOGISTIC), 3311, 3347
LOGISTIC procedure, 3298, 3310, 3346, 3368
Roger and Tanimoto coefficient
DISTANCE procedure, 1512
root MSE
SURVEYREG procedure, 6571
rotating principal components, 5161
row mean scores statistic, 1773
Roy's greatest root, 706, 2464, 2536
RPC convergence measure, 4308
RSREG procedure
canonical analysis, 5737
coding variables, 5738, 5744
compared to other procedures, 2434, 5720
computational methods, 5742
confidence intervals, 5732, 5733
Cook's *D* influence statistic, 5732
covariates, 5721
eigenvalues, 5742
eigenvectors, 5742
factor variables, 5721
input data sets, 5726, 5731
introductory example, 5721
missing values, 5738
ODS graph names, 5747
ODS table names, 5746
output data sets, 5727, 5734, 5743
PRESS statistic, 5733
response variables, 5721
ridge analysis, 5737
RTF destination
ODS Graphics, 602
RTF style
ODS Styles, 515
run times
MCMC procedure, 3566, 3570
Russell and Rao similarity coefficient
DISTANCE procedure, 1513
Ryan's multiple range test, 713, 2472, 2523
examples, 2560

S convergence measure, 4308
saddle test, definition
MODECLUS procedure, 4118
salience of loadings, FACTOR procedure, 1537, 1571
Sampford's method
SURVEYSELECT procedure, 6653
sample
Introduction to Survey Procedures, 265
SURVEYSELECT procedure, 6614
sample design
Introduction to Survey Procedures, 264
SURVEYFREQ procedure, 6320
SURVEYSELECT procedure, 6614
sample selection
Introduction to Survey Procedures, 259, 268
SURVEYSELECT procedure, 6614
sample selection methods
SURVEYSELECT procedure, 6645
sample size, 413
CATMOD procedure, 1163
Introduction to Modeling, 49
overview of power concepts, 413
overview of power concepts (POWER), 4916
overview of SAS tools, 414
per group, 5108, 5128
See GLMPOWER procedure, 2641
See POWER procedure, 4816
SEQDESIGN procedure, 5874, 5879
solving for, 5108, 5112, 5128
SURVEYSELECT procedure, 6636
total, 5108
weights, 5109, 5121
sample size adjustment
GLMPOWER procedure, 2661

POWER procedure, 4922
sample size allocation
SURVEYSELECT procedure, 6640, 6654
sample size computation
SEQDESIGN procedure, 5854, 5858, 5860, 5868
SEQTEST procedure, 6026
sample size summary
SEQDESIGN procedure, 5878
sample size weights, 5109, 5121
sample space ordering
SEQTEST procedure, 6008
sample space orderings
SEQTEST procedure, 6025
sample survey analysis, ordinal data, 1095
sampling, *see also* survey sampling
Introduction to Survey Procedures, 259
SURVEYSELECT procedure, 6614
sampling fractions
Introduction to Survey Procedures, 266
sampling frame
Introduction to Survey Procedures, 265
SURVEYSELECT procedure, 6614, 6627
sampling rate
SURVEYSELECT procedure, 6634
sampling rates
Introduction to Survey Procedures, 266
SURVEYFREQ procedure, 6303, 6322
SURVEYLOGISTIC procedure, 6384, 6422
SURVEYMEANS procedure, 6476, 6492
SURVEYREG procedure, 6544, 6561
sampling units
Introduction to Survey Procedures, 265
SURVEYSELECT procedure, 6616, 6645
sampling weights
Introduction to Survey Procedures, 266
SURVEYFREQ procedure, 6320, 6321
SURVEYLOGISTIC procedure, 6406, 6409
SURVEYMEANS procedure, 6488, 6490
SURVEYREG procedure, 6558, 6560
SURVEYSELECT procedure, 6617
sampling zeros
and log-linear analyses (CATMOD), 1148
and structural zeros (CATMOD), 1164
sandwich estimator, *see also* empirical estimator
GLIMMIX procedure, 2096, 2243, 2246
MIXED procedure, 3907
NLMIXED procedure, 4360
SAS code, 5082
SAS connection
defining, 5070
selecting, 5068
SAS data set
DATA step, 23
summarizing, 24
SAS log, 5082
SAS Registry, 537
SAS Registry Editor, 575
SAS/ETS software, 24
SAS/GRAPH software, 24
SAS/IML software, 24
SAS/INSIGHT software, 25
SAS/OR software, 25
SAS/QC software, 25
SASHELP.Tmplmst
template library, 587
SASUSER.Templat
template library, 445, 587
Satterthwaite method
GLIMMIX procedure, 2167, 2241
MIXED procedure, 3931
Satterthwaite *t* test
power and sample size (POWER), 4891, 4899, 4971
Satterthwaite's approximation
testing random effects, 2542
TTEST procedure, 7417
Savage scores
NPAR1WAY procedure, 4471
sawtooth power function, 4986
scale estimates
FASTCLUS procedure, 1635, 1637, 1642, 1644
scale parameter
GENMOD procedure, 1966
GLIMMIX compared to GENMOD, 2216
GLIMMIX procedure, 2083–2085, 2100, 2109, 2119, 2131, 2181, 2182, 2185, 2208, 2212, 2215, 2217, 2221, 2222, 2224, 2228, 2239, 2270, 2275, 2277, 2315, 2331, 2335, 2348, 2352, 2381, 2399
scaled inverse chi-square distribution
definition of (MCMC), 3539
MCMC procedure, 3515, 3539
scaled residual
MIXED procedure, 3942, 3988
scaling variables
DISTANCE procedure, 1485
MODECLUS procedure, 4094
STDIZE procedure, 6236
scalogram analysis
CORRESP procedure, 1320
scatter plot
ODS Graphics, 578
scenarios, 5058
Scheffé's multiple-comparison procedure, 2518
Scheffe's adjustment

LIFETEST procedure, 3127
Scheffé's multiple-comparison procedure, 714
Scheffe's multiple-comparison procedure
Scheffé's multiple-comparison procedure, 2472
Schoenfeld residuals
PHREG procedure, 4574, 4612, 4613
Schwarz criterion
LOGISTIC procedure, 3329
PHREG procedure, 4610
SURVEYLOGISTIC procedure, 6420
Schwarz's Bayesian information criterion
example (MIXED), 4017, 4031, 4060
GLIMMIX procedure, 2099
HPMIXED procedure, 2811, 2838
MIXED procedure, 3907, 3976, 3997
score function
SEQDESIGN procedure, 5818, 5819
score information
GLMSELECT procedure, 2748
SCORE procedure
CALIS procedure, 862, 864, 922
computational resources, 5770
examples, 5762, 5770
input data set, 5766
OUT= data sets, 5769
output data set, 5766, 5769
PRINCOMP procedure, 5161
regression parameter estimates from REG procedure, 5769
scoring coefficients, 5760
TCALIS procedure, 6739, 6741, 6835
score residuals
PHREG procedure, 4574, 4612
score statistic
SEQDESIGN procedure, 5819
score statistic scale
SEQDESIGN procedure, 5832
score statistics
GENMOD procedure, 1981
LOGISTIC procedure, 3331
SURVEYLOGISTIC procedure, 6433
score test
PHREG procedure, 4569, 4571, 4598, 4634, 4646, 4648
score variables
interpretation (SCORE), 5769
scores
NPAR1WAY procedure, 4470
scoring
GLIMMIX procedure, 2095
MIXED procedure, 3905, 3915, 4012
scoring coefficients (SCORE), 5759
scree plot, 1579
screening design, analysis, 2608
screening experiments
GLMMOD procedure, 2637
SDF, *see* survival distribution function
second-order algorithm
Shared Concepts, 405
second-order factor model
TCALIS procedure, 7006
seed
initial (SURVEYSELECT), 6637
selected effects
GLMSELECT procedure, 2745
selection list
ODS, 439, 453
selection methods, *see* model selection, *see* model selection, *see* model selection
selection summary
GLMSELECT procedure, 2744
semiparametric model
PHREG procedure, 4523
semipartial correlation
CANCORR procedure, 1041
formula (CLUSTER), 1258
semivariance, *see also* semivariogram
classical (VARIOGRAM), 7525, 7545
computation (VARIOGRAM), 7558
empirical (VARIOGRAM), 7545
robust (VARIOGRAM), 7525, 7538, 7545
theoretical models, 7542
VARIOGRAM procedure, 7518, 7544, 7596
semivariogram
analysis (VARIOGRAM), 7519
and covariogram (VARIOGRAM), 7596
computation (VARIOGRAM), 7519
empirical (VARIOGRAM), 7519, 7545
parameters (VARIOGRAM), 7540
robust (VARIOGRAM), 7545
theoretical models, 7519, 7522
theoretical models (VARIOGRAM), 7540
theoretical models fitting, 7570, 7584, 7593
VARIOGRAM procedure, 7518, 7545
sensitivity
CATMOD procedure, 1215
sensitivity analysis
power and sample size, 419
separate regression functions
TRANSREG procedure, 7213
SEQDESIGN procedure
alternative reference, 5873
ASN plot, 5881
asymmetric two-sided design, 5878
average sample number, 5873
average sample numbers plot, 5802

boundary for Whitehead one-sided design, 5848
boundary information, 5876
boundary key, 5853
boundary plot, 5802, 5881
boundary scales, 5832
boundary variables, 5834
canonical joint distribution, 5872
clinical trial, 5784, 5817
combined boundary plot, 5802, 5881
derived parameters, 5851
design information, 5877
drift parameter, 5822, 5825, 5826, 5852, 5873
error spending function method, 5876
error spending information, 5877
error spending method, 5791, 5829, 5848, 5849, 5876
error spending plot, 5803, 5882
expected Fisher information, 5818, 5820
expected sample size, 5878
Fisher information, 5818
gamma error spending function, 5806
gamma error spending method, 5850
generalized two-sided test, 5826
group sequential design, 5827
group sequential trial, 5784, 5984
Haybittle-Peto method, 5791, 5806, 5829, 5844, 5875
hazard function, 5867
input fixed-sample D, 5810
input fixed-sample N, 5810
input number of events for fixed-sample design, 5856
input sample size for fixed-sample design, 5856
introductory example, 5791
log odds-ratio statistic, 5864
log relative risk statistic, 5865
log-rank test, 5866
log-rank test for two survival distributions, 5814
maximum information, 5822, 5826, 5855, 5873
maximum likelihood estimate, 5818
maximum likelihood estimate scale, 5832
method information, 5877
MLE, 5818
noninferiority trial, 5823
normality assumption, 5872
number of stages, 5872
O'Brien-Fleming method, 5807, 5827, 5842, 5874
O'Brien-Fleming-type error spending function, 5806
O'Brien-Fleming-type error spending method, 5849
observed Fisher information, 5818, 5820
ODS graphics names, 5882
ODS table names, 5880
one-sample test for binomial proportion, 5812
one-sample test for mean, 5811
one-sample tests, 5858
one-sided test, 5821
overlapping β boundaries, 5850
p-value scale, 5832
Pocock method, 5807, 5827, 5842, 5874
Pocock-type error spending function, 5806
Pocock-type error spending method, 5849
power, 5878
power curves plot, 5803
power error spending function, 5806
power error spending method, 5850
power family method, 5807, 5828, 5843, 5875
power plot, 5882
reference improvement, 5826
regression tests, 5868
relative cumulative error spending, 5806
repeated significance test, 5827
sample size, 5874, 5879
sample size computation, 5854, 5858, 5860, 5868
sample size summary, 5878
score function, 5818, 5819
score statistic, 5819
score statistic scale, 5832
specified parameters, 5851
standardized Z scale, 5832
statistical assumptions, 5829
stopping probabilities, 5879
superiority trial, 5823
survival function, 5867
symmetric two-sided test, 5824
syntax, 5799
test for a binomial proportion, 5859
test for a normal mean, 5858
test for a parameter, 5869–5871
test for difference between two normal means, 5860
test for logistic regression parameter, 5816
test for proportional hazards regression parameter, 5816
test for regression parameter, 5815
test for two binomial proportions, 5862, 5864, 5865

test for two survival distributions, 5866
two-sample test for binomial proportions, 5813
two-sample test for mean difference, 5812
two-sample tests, 5860
two-sided asymmetric design, 5878
two-sided test, 5824
Type I error, 5804, 5836
Type I error probability, 5827, 5837, 5838
Type II error, 5804, 5836
Type II error probability, 5827, 5837, 5838
unified family method, 5791, 5807, 5828, 5839, 5875
unified family method shape parameters, 5840
unified family triangular method, 5807, 5843, 5875
Whitehead method, 5791, 5829, 5845, 5875
Whitehead one-sided asymmetric design, 5846
Whitehead one-sided symmetric design, 5845
Whitehead two-sided design, 5846
Whitehead's double-triangular design, 5806
Whitehead's triangular design, 5806
Whitehead's triangular method, 5828
SEQTEST procedure
acceptance repeated confidence intervals, 6023
applicable tests, 6026
ASN plot, 6032
boundary adjustment method, 6005
boundary adjustments, 6017, 6019, 6020
boundary key, 6006
boundary scale, 6006
clinical trial, 5984
conditional power, 6009, 6011, 6021, 6028, 6033
conditional power plot, 6033
confidence level, 6007
confidence limits, 6007, 6024
design information, 6028
error spending, 6011
error spending information, 6028
error spending plot, 6033
expected mean sample size, 6011
expected sample size, 6029
futility index, 6022
introductory example, 5988
LR ordering, 6026
median unbiased estimate, 6024
minimum error spending, 6007, 6019
MLE ordering, 6026
number of stages, 6008
ODS graphics names, 6034
ODS table names, 6032
one-sided repeated confidence intervals, 6023
overlapping β boundaries, 6020
p-value, 6024
parameter estimates, 6029
power, 6011, 6029
power plot, 6033
predictive power, 6009, 6022, 6030
rejection repeated confidence intervals, 6023
repeated confidence intervals, 6010, 6012, 6022, 6030, 6034
repeated confidence intervals plot, 6034
sample size computation, 6026
sample space ordering, 6008
sample space orderings, 6025
sequential test plot, 6034
stagewise ordering, 6025
stochastic curtailment, 6021
stopping probabilities, 6030
syntax, 6003
test information, 6031
two-sided repeated confidence intervals, 6023
sequential random sampling
SURVEYSELECT procedure, 6647, 6664
sequential test plot
SEQTEST procedure, 6034
serpentine sorting
SURVEYSELECT procedure, 6644
SGPANEL procedure
ODS Graphics, 593
SGPLOT procedure
ODS Graphics, 589
SGRENDER procedure
ODS Graphics, 596
SGSCATTER procedure
ODS Graphics, 591
shape distance coefficient
DISTANCE procedure, 1508
Shared Concepts
bar (|) operator, 369
choosing optimization algorithm, 405
CLASS statement, 366
classification variables, 366
collection effect (EFFECT statement), 378
conjugate gradient method, 408
continuous-by-class effects, 372
continuous-nesting-class effects, 371
crossed effects, 370
double-dogleg method, 408
effect parameterization, 373
EFFECT statement, 377

first-order algorithm, 405
General Effects, 372
GLM parameterization, 374
interaction effects, 370
intercept, 369
levelization, 366
main effects, 369
missing values, class variables, 367
multimember effect (EFFECT statement), 378
Nelder-Mead simplex method, 409
nested effects, 371
nested v. crossed effects, 371
Newton-Raphson method, 406
Newton-Raphson with ridging, 407
NLOPTIONS statement, 391
ORDER= option, 367
ordering of class levels, 367
ordinal parameterization, 374
ortheffect parameterization, 375
orthordinal parameterization, 376
orthoterm parameterization, 376
orthpoly parameterization, 376
orthref parameterization, 376
parameterization, 368
polynomial effect (EFFECT statement), 380
polynomial effects, 369
polynomial parameterization, 374
programming statements, 410
quasi-Newton method, 407
reference parameterization, 375
regression effects, 369
second-order algorithm, 405
simplex method, 409
singular parameterization, 370
sort order of class levels, 367
spline bases, 387
spline basis, B-spline, 389
spline basis, truncated power function, 388
spline effect (EFFECT statement), 384
splines, 387
thermometer parameterization, 374
trust region method, 406
Shared concepts
Remote monitoring, 403
Shewhart control charts, 25
Sidak's *t* test, 2473, 2518
Sidak's adjustment
GLIMMIX procedure, 2144
GLM procedure, 2459
LIFETEST procedure, 3127
MIXED procedure, 3924
MULTTEST procedure, 4197, 4213, 4233
Sidak's inequality, 714
Siegel-Tukey scores
NPAR1WAY procedure, 4471
sign test
Introduction to Nonparametric Analysis, 295
significance level, *see* alpha level, 5090
CALIS procedure, 851
entry (PHREG), 4571
removal (PHREG), 4571, 4650
TCALIS procedure, 6726
significance tests
MODECLUS procedure, 4115, 4162
sill
KRIGE2D procedure, 2949
VARIOGRAM procedure, 7540
SIM2D procedure
Cholesky root, 6177
computational details, 6179
conditional and unconditional simulation, 6156
conditional distributions of multivariate normal random variables, 6178
conditional simulation, 6155, 6178
examples, 6157, 6182, 6187
Gaussian assumption, 6156
Gaussian random field, 6155
LU decomposition, 6176
memory usage, 6180
ODS graph names, 6182
ODS Graphics, 6166
ODS table names, 6181
output data sets, 6166, 6180
OUTSIM= data set, 6180
quadratic form, 6178
simulation of spatial random fields, 6176–6179
unconditional simulation, 6155, 6178
SIM2D procedure, tables
Model Information, 6181
Number of Observations, 6181
Simulation Information, 6181
similarity data
MDS procedure, 3704, 3713, 3720
similarity Ratio coefficient
DISTANCE procedure, 1509
similarity ratio coefficient
DISTANCE procedure, 1510
SIMNORMAL procedure
conditional simulation, 6207
Gaussian random variables, 6193
introductory example, 6194
LU decomposition method, 6206
normal random variables, 6193
simulation, 6193
unconditional simulation, 6206

simple cluster-seeking algorithm, 1624
simple covariance matrix
GLIMMIX procedure, 2201
simple covariance structures model example (TCALIS), 6712
simple effects
GLIMMIX procedure, 2152
GLM procedure, 2463, 2525
MIXED procedure, 3928
simple effects differences
GLIMMIX procedure, 2152
simple factor model example (TCALIS), 6710
simple kappa coefficient, 1765, 1766
simple Matching coefficient
DISTANCE procedure, 1511
simple matching dissimilarity coefficient
DISTANCE procedure, 1511
simple random cluster sampling
SURVEYREG procedure, 6585
simple random sampling
SURVEYMEANS procedure, 6467
SURVEYREG procedure, 6535, 6582
SURVEYSELECT procedure, 6616, 6646
simplex method
Shared Concepts, 409
simplicity functions
CALIS procedure, 878
FACTOR procedure, 1537, 1559, 1572
TCALIS procedure, 6754
SIMPLS method
PLS procedure, 4785
simulated adjustment
LIFETEST procedure, 3127
simulation
conditional (SIM2D), 6155, 6178
power, 417, 429
unconditional (SIM2D), 6155, 6178
simulation of spatial random fields
SIM2D procedure, 6176–6179
simulation-based adjustment
GLIMMIX procedure, 2144
GLM procedure, 2459
MIXED procedure, 3924
single linkage
CLUSTER procedure, 1240, 1255
singly iterative algorithm
GLIMMIX procedure, 2271
singular parameterization
Shared Concepts, 370
singularities
MIXED procedure, 4012
REG procedure, 5584
singularity
HPMIXED procedure, 2814
MI procedure, 3754
singularity checking
CANCORR procedure, 1041
GLM procedure, 2455, 2457, 2463, 2476
singularity criterion
CALIS procedure, 867
contrast matrix (GENMOD), 1942
contrast matrix (LOGISTIC), 3283
contrast matrix (PHREG), 4561
covariance matrix (CALIS), 852, 860, 870
covariance matrix (TCALIS), 6726, 6734, 6744
information matrix (GENMOD), 1953
NLIN procedure, 4282
PHREG procedure, 4571
TCALIS procedure, 6743
TRANSREG procedure, 7166
singularity level
SURVEYREG procedure, 6552, 6554, 6556
singularity tolerances
NLMIXED procedure, 4370
size distance coefficient
DISTANCE procedure, 1508
size measure
PPS sampling (SURVEYSELECT), 6639
skewness
CALIS procedure, 939
displayed in CLUSTER procedure, 1246
NLIN procedure, 4280
TCALIS procedure, 6919
SMM multiple-comparison method, 714, 2473, 2519
smoothed hazard
LIFETEST procedure, 3101
smoothing parameter
cluster analysis, 1253
MODECLUS procedure, 4103, 4110
optimal (DISCRIM), 1406
smoothing parameter, default
MODECLUS procedure, 4111
smoothing spline transformation
PRINQUAL procedure, 5216
TRANSREG procedure, 7147, 7222
software requirements, 5085
Sokal and Sneath 1 coefficient
DISTANCE procedure, 1512
Sokal and Sneath 3 coefficient
DISTANCE procedure, 1512
Somers' *D* statistics, 1736, 1739
sort order
of class levels (Shared Concepts), 367
spacing
STDIZE procedure, 6231
sparse matrix techniques

HPMIXED procedure, 2835
spatial anisotropic exponential structure
MIXED procedure, 3959
spatial continuity
KRIGE2D procedure, 2930
VARIOGRAM procedure, 7518, 7519, 7544, 7546, 7587, 7596
spatial covariance structure
examples (MIXED), 3960
GLIMMIX procedure, 2201
MIXED procedure, 3960, 3968, 4011
spatial dependence, *see* spatial continuity, *see* spatial continuity, *see* spatial continuity
spatial exponential structure
GLIMMIX procedure, 2201
spatial gaussian structure
GLIMMIX procedure, 2201
spatial Matérn structure
GLIMMIX procedure, 2201
spatial power structure
GLIMMIX procedure, 2202
spatial prediction
KRIGE2D procedure, 2930
VARIOGRAM procedure, 7518, 7546
spatial random field
autocorrelation (VARIOGRAM), 7560
ergodicity (VARIOGRAM), 7547
intrinsically stationary (VARIOGRAM), 7546
isotropic (VARIOGRAM), 7547, 7560
realization (VARIOGRAM), 7540
spatial continuity (VARIOGRAM), 7544, 7587
VARIOGRAM procedure, 7540
spatial spherical structure
GLIMMIX procedure, 2202
spatial structure, *see* spatial continuity, *see* spatial continuity, *see* spatial continuity
Spearman rank correlation coefficient, 1736, 1741
specificity
CATMOD procedure, 1215
specified parameters
SEQDESIGN procedure, 5851
specifying a new distribution
MCMC procedure, 3541
spectral density estimate at zero frequency
Introduction to Bayesian Analysis, 164
spherical covariance structure
GLIMMIX procedure, 2202
spherical semivariogram model
KRIGE2D procedure, 2949
sphericity tests, 718, 2484, 2593
Spline bases
Shared Concepts, 387
spline bases
GLIMMIX procedure, 387
GLMSELECT procedure, 387
spline comparisons
GLIMMIX procedure, 2406
spline effect
GLIMMIX procedure, 384
GLMSELECT procedure, 384
spline smoothing
GLIMMIX procedure, 2198, 2199
spline t-options
PRINQUAL procedure, 5218
TRANSREG procedure, 7150
spline transformation
PRINQUAL procedure, 5215
TRANSREG procedure, 7146, 7261
Splines
Shared Concepts, 387
splines
Bayesian confidence intervals, 7075
goodness of fit, 7076
regression model, 7057, 7075, 7079
thin-plate smoothing, 7057
TPSPLINE procedure, 7057
TRANSREG procedure, 7142, 7143, 7192, 7264, 7278, 7307
split-plot design
ANOVA procedure, 720, 737, 739, 741
generating with PLAN procedure, 4749
MIXED procedure, 3972, 4014
square root difference cloud
VARIOGRAM procedure, 7600
squared correlation dissimilarity coefficient
DISTANCE procedure, 1509
squared correlation similarity coefficient
DISTANCE procedure, 1509
squared Euclidean distance coefficient
DISTANCE procedure, 1508
squared multiple correlation
CALIS procedure, 936, 965
CANCORR procedure, 1041
TCALIS procedure, 6912, 6937
squared partial correlation
CANCORR procedure, 1041
squared semipartial correlation
CANCORR procedure, 1041
formula (CLUSTER), 1258
squared simple matching dissimilarity coefficient
DISTANCE procedure, 1511
SRF, *see* spatial random field
SSCP matrix
displaying, for multivariate tests, 708
for multivariate tests, 706

for multivariate tests (GLM), 2464, 2466
stability coefficient
TCALIS procedure, 6729, 6915
stacking table
SURVEYMEANS procedure, 6476, 6510
stagewise ordering
SEQTEST procedure, 6025
standard deviation
CLUSTER procedure, 1246
standard deviations
SURVEYMEANS procedure, 6501
standard distributions
MCMC procedure, 3530
standard error
baseline estimation (PHREG), 4616
GENMOD procedure, 2016
LIFEREG procedure, 3058
PHREG procedure, 4543, 4573, 4574, 4634
standard error ratio
PHREG procedure, 4634
standard errors
KRIGE2D procedure, 2930
SURVEYMEANS procedure, 6496
VARIOGRAM procedure, 7519
standard linear model
MIXED procedure, 3894
STANDARD procedure, 24
standardized values, 24
standardization
comparisons between DISTANCE and STDIZE procedures, 1485
standardization suppression
DISTANCE procedure, 1498
standardized score process
PHREG procedure, 4620, 4643
standardized Z scale
SEQDESIGN procedure, 5832
standardizing
cluster analysis (STDIZE), 6236
CLUSTER procedure, 1246
MODECLUS procedure, 4094
raw data (SCORE), 5761
TRANSREG procedure, 7158
values (STANDARD), 24
values (STDIZE), 6211
star (*) operator
TRANSREG procedure, 7139
starting values
NLIN procedure, 4289
stationarity
intrinsic (VARIOGRAM), 7546
second-order (VARIOGRAM), 7546, 7598
VARIOGRAM procedure, 7540
stationary point
NLMIXED procedure, 4401
statistic-keywords
SURVEYMEANS procedure, 6477
statistical
assumptions (GLM), 2487
quality control, 25
tests (MULTTEST), 4203
statistical assumptions
SEQDESIGN procedure, 5829
statistical computations
SURVEYMEANS procedure, 6493
statistical graphics
GLIMMIX procedure, 2282, 2299
LOESS procedure, 3226
TTEST procedure, 7427
Statistical Graphics Using ODS, *see* ODS Graphics
statistical model
definition (Introduction to Modeling), 28
STATISTICAL style
ODS Styles, 514
STD option (MODECLUS), 4094
STD= option (DISTANCE), 1501
STDIZE procedure
AGK estimate, 6231
analyzing data in groups, 6225
Andrew's wave estimate, 6231
breakdown point and efficiency, 6230
comparisons of quantile computation, PCTLMTD option, 6232
computational methods, PCTLDEF option, 6232
Euclidean length, 6231
examples, 6212, 6236
final output value, 6212
formulas for statistics, 6230
fractional frequencies, 6227
fuzz factor, 6222
Huber's estimate, 6231
initial estimates for A estimates, 6222
input data set (METHOD=IN()), 6229
methods resistant to clustering, 6230
methods resistant to outliers, 6216, 6230
Minkowski metric, 6231
missing values, 6223, 6224, 6233
normalization, 6223, 6225
one-pass quantile computations, 6232
OUT= data set, 6222, 6234
output data sets, 6223, 6224, 6234
output table names, 6235
OUTSTAT= data set, 6234
quantile computation, 6212, 6232
robust estimators, 6230
spacing, 6231

standardization methods, 6211, 6229
standardization with weights, 6227
Tukey's biweight estimate, 6218, 6231
tuning constant, 6218, 6229
unstandardization, 6225
weights, 6227
step halving
PHREG procedure, 4595
step length
CALIS procedure, 856
TCALIS procedure, 6730
step length options
NLMIXED procedure, 4397
step-down methods
MULTTEST procedure, 4214, 4233
STEPDISC procedure
average squared canonical correlation, 6283
computational resources, 6281
input data sets, 6280
introductory example, 6269
memory requirements, 6281
methods, 6267
missing values, 6280
ODS table names, 6284
Pillai's trace, 6283
stepwise selection, 6269
time requirements, 6281
tolerance, 6283
Wilks' lambda, 6283
stepdown methods
GLM procedure, 2522
stepwise discriminant analysis, 6267
stepwise selection
GLMSELECT procedure, 2723
LOGISTIC procedure, 3301, 3329, 3381
PHREG procedure, 4570, 4618, 4644
REG procedure, 5435, 5524
STEPDISC procedure, 6269
stochastic analysis
KRIGE2D procedure, 2930
VARIOGRAM procedure, 7518
stochastic curtailment
SEQTEST procedure, 6021
stochastic model
definition (Introduction to Modeling), 28
stochastic modeling, *see* modeling, *see* modeling
stochastic spatial prediction, *see* spatial prediction, *see* spatial prediction
stop details
GLMSELECT procedure, 2745
stop reason
GLMSELECT procedure, 2745
stopping probabilities
SEQDESIGN procedure, 5879
SEQTEST procedure, 6030
stored data algorithm, 1259
stored distance algorithms, 1259
strata variables
PHREG procedure, 4577
programming statements (PHREG), 4575
strata weights
MULTTEST procedure, 4206
stratification, *see also* stratified sampling, *see also* stratified sampling
SURVEYFREQ procedure, 6312, 6321
SURVEYLOGISTIC procedure, 6407
SURVEYMEANS procedure, 6489
SURVEYREG procedure, 6559
stratified analysis
FREQ procedure, 1676, 1697
PHREG procedure, 4524, 4577
stratified cluster sample
SURVEYMEANS procedure, 6518
stratified sampling
Introduction to Survey Procedures, 265
SURVEYMEANS procedure, 6469
SURVEYREG procedure, 6538, 6589
SURVEYSELECT procedure, 6618, 6640
stratified table
example, 1817
stratified tests
LIFETEST procedure, 3100, 3101, 3108, 3110, 3129, 3143, 3153
stratum collapse
SURVEYREG procedure, 6572, 6598
stress formula
MDS procedure, 3715, 3725
strip-split-plot design
ANOVA procedure, 741
structural equation (CALIS)
definition, 863, 907, 937
dependent variables, 907
models, 832
structural equation (TCALIS)
definition, 6741
dependent variables, 6817
structural model example (CALIS), 835
COSAN model, 842
factor analysis model, 844
LINEQS model, 837, 841
LISREL model, 838
path diagram, 835, 903
RAM model, 837, 842
structural model example (TCALIS), 6704
LINEQS model, 6708
LISMOD, 6709
path diagram, 6705
PATH model, 6706

RAM model, 6707
Stuart's tau-*c* statistic, 1736, 1739
Student's multiple range test, 714, 2473, 2522
Studentized maximum modulus
pairwise comparisons, 714, 2473, 2519
studentized maximum modulus adjustment
LIFETEST procedure, 3127
studentized residual, 2478, 2479, 5560
external, 3991
internal, 3991
MIXED procedure, 3941, 3991
study planning, 418
style
ODS Graphics, 520, 554
style elements
ODS Graphics, 558
style modification
ODS Graphics, 619
style modification, ModStyle macro
ODS Graphics, 564
style templates
ODS, 441
style, box plot
ODS Graphics, 644
style, customizing
ODS Graphics, 573
style, default
ODS Graphics, 575
subdomain analysis, *see also* domain analysis, *see also* domain analysis, *see also* domain analysis, *see also* domain analysis
subgroup analysis, *see also* domain analysis, *see also* domain analysis, *see also* domain analysis, *see also* domain analysis
subject effect
GLIMMIX procedure, 2193
HPMIXED procedure, 2831
MIXED procedure, 3918, 3952, 3958, 4013, 4019
subject processing
GLIMMIX procedure, 2247
subject weights
MDS procedure, 3704, 3712
subpopulation
GENMOD procedure, 1947
LOGISTIC procedure, 3300
PROBIT procedure, 5296, 5299, 5315
subpopulation analysis, *see also* domain analysis, *see also* domain analysis, *see also* domain analysis, *see also* domain analysis
subsidiary group specification statements (TCALIS), 6716
subsidiary model specification statements (TCALIS), 6718
sum of squares
corrected total (Introduction to Modeling), 64
decomposition (Introduction to ANOVA Procedures), 116, 118
F-test (Introduction to ANOVA Procedures), 120
for linear hypothesis (Introduction to ANOVA Procedures), 120
model (Introduction to ANOVA Procedures), 119
residual (Introduction to ANOVA Procedures), 120
Type I (Introduction to ANOVA Procedures, 116
Type III (Introduction to ANOVA Procedures, 116
uncorrected total (Introduction to Modeling), 64
sum of squares reduction test
Introduction to Modeling, 67, 68
sum-to-zero assumptions, 2542
summary of commands
MIXED procedure, 3903
summary statistics
PHREG procedure, 4639
REG procedure, 5557
summary table, 5061
sums of squares
GLM procedure, 2477
Type II (GLM), 2477
Type II (TRANSREG), 7166
superiority tests
power and sample size (POWER), 4852
superiority trial
SEQDESIGN procedure, 5823
suppressing output
CANCORR procedure, 1040
GENMOD procedure, 1926
MI procedure, 3753
surface trend
VARIOGRAM procedure, 7518, 7522, 7546, 7559, 7574, 7578
survey data analysis
Introduction to Survey Procedures, 259, 269
SURVEYFREQ procedure, 6294
survey design
Introduction to Survey Procedures, 264
survey sampling, *see also* SURVEYREG procedure
cluster sampling (Introduction to Survey Procedures), 265

data analysis (SURVEYFREQ), 6294
descriptive statistics, 6466
Introduction to Survey Procedures, 259
multistage sampling (Introduction to Survey Procedures), 266
population (Introduction to Survey Procedures), 265
primary sampling units (PSUs) (Introduction to Survey Procedures), 266
regression analysis, 6534
sample design (Introduction to Survey Procedures), 264
sample selection (SURVEYSELECT), 6614
sampling frame (Introduction to Survey Procedures), 265
sampling units (Introduction to Survey Procedures), 265
sampling weights (Introduction to Survey Procedures), 266
stratified sampling (Introduction to Survey Procedures), 265
SURVEYSELECT procedure, 6614
variance estimation (Introduction to Survey Procedures), 267
survey weights, *see* sampling weights, *see* sampling weights
SURVEYFREQ procedure, 6294
alpha level, 6315
BRR variance estimation, 6334
clustering, 6310, 6321
coefficient of variation, 6341
column proportions, 6332
confidence limits, 6339
covariance, 6330
crosstabulation tables, 6312, 6358
degrees of freedom, 6340
design effects, 6341
design-adjusted chi-square tests, 6346
displayed output, 6356
domain analysis, 6323, 6366
expected frequencies, 6342
Fay's BRR variance estimation, 6335
finite population correction, 6303
frequency tables, 6312
Hadamard matrix (BRR variance estimation), 6336
Introduction to Survey Procedures, 259, 263
introductory example, 6294
jackknife coefficients, 6337
jackknife variance estimation, 6337
missing values, 6323
multiway tables, 6358
odds ratio, 6344
ODS table names, 6362
one-way frequency tables, 6357
order of levels, 6303
output data sets, 6355, 6368
population totals, 6304, 6322
primary sampling units (PSUs), 6310
proportions, 6331
Rao-Scott chi-square test, 6346
Rao-Scott likelihood ratio test, 6350
relative risks, 6345
replicate weights, 6310
risk difference, 6342
risks, 6342
row proportions, 6332
sample design, 6320
sampling rates, 6303, 6322
sampling weights, 6320, 6321
statistical computations, 6326
stratification, 6312, 6321
Taylor series variance estimation, 6327
totals, 6329
variance estimation, 6326
Wald chi-square test, 6352
Wald log-linear chi-square test, 6354
weighting, 6320, 6321
SURVEYLOGISTIC procedure
Akaike's information criterion, 6420
alpha level, 6383, 6393, 6398, 6404
analysis of maximum likelihood estimates table, 6445
association of predicted probabilities and observed responses table, 6446
balanced repeated replication, 6429
BRR, 6429
BRR variance estimation, 6429
class level information table, 6443
clustering, 6391
complementary log-log model, 6425
confidence intervals, 6434
confidence limits, 6438
convergence criterion, 6398, 6399
cumulative logit model, 6424
customized odds ratio, 6409
data summary table, 6443
displayed output, 6441
domain analysis, 6394, 6432
domain variable, 6394
donor stratum, 6431
EFFECT parameterization, 6412
estimability checking, 6394
estimated covariance matrix table, 6446
existence of MLEs, 6418
Fay coefficient, 6386, 6430
Fay's BRR variance estimation, 6430
finite population correction, 6384, 6422

first-stage sampling rate, 6384
Fisher scoring method, 6401, 6417
GLM parameterization, 6412
gradient, 6433
Hadamard matrix, 6386, 6432, 6447
Hessian matrix, 6401, 6433
infinite parameter estimates, 6400
initial values, 6421
Introduction to Survey Procedures, 259, 264
jackknife, 6431
jackknife coefficients, 6431, 6441
jackknife variance estimation, 6431
likelihood functions, 6423
linear hypothesis results table, 6447
linearization method, 6427
link functions, 6375, 6399, 6415
log odds, 6435
maximum likelihood algorithms, 6417
maximum likelihood iteration history table, 6444
Medical Expenditure Panel Survey (MEPS), 6455
missing values, 6383, 6410
model fit statistics table, 6444
model fitting criteria, 6420
model information table, 6441
model parameters, 6423
Newton-Raphson algorithm, 6401, 6418
number of replicates, 6387, 6429–6431
odds ratio, 6434
odds ratio confidence limits, 6398
odds ratio estimates table, 6446
odds ratio estimation, 6434
ORDINAL parameterization, 6413
ORTHEFFECT parameterization, 6414
ORTHORDINAL parameterization, 6414
ORTHOTHERM parameterization, 6414
ORTHPOLY parameterization, 6414
ORTHREF parameterization, 6414
output data sets, 6439
output jackknife coefficient, 6441
output replicate weights, 6440
output table names, 6447
parameterization, 6412
POLY parameterization, 6413
POLYNOMIAL parameterization, 6413
population totals, 6384, 6422
predicted probabilities, 6438
primary sampling units (PSUs), 6423
probit model, 6425
proportional odds model, 6424
rank correlation, 6437
REF parameterization, 6413
REFERENCE parameterization, 6413
regression parameters, 6423
replicate weights, 6406, 6427
replication methods, 6385, 6427
response profile table, 6443
reverse response level ordering, 6396, 6411
sampling rates, 6384, 6422
sampling weights, 6406, 6409
Schwarz criterion, 6420
score statistics, 6433
score test table, 6444
stratification, 6407
stratum information table, 6444
Taylor series variance estimation, 6388, 6427
testing linear hypotheses, 6408, 6434
type 3 analysis of effects table, 6445
variance estimation, 6427
variance estimation table, 6442
Wald confidence interval for odds ratios table, 6446
Wald confidence interval for parameters table, 6446
weighting, 6406, 6409
SURVEYMEANS procedure, 6466
alpha level, 6474
balanced repeated replication, 6506
BRR, 6506, 6528
BRR variance estimation, 6506
categorical variable, 6484, 6495, 6500
class level information table, 6513
classification variable, 6495
clustering, 6485
coefficient of variation, 6500
confidence level, 6474
confidence limits, 6499, 6501
data and sample design summary table, 6512
degrees of freedom, 6498
denominator variable, 6486
domain analysis, 6493
domain analysis table, 6516
domain means, 6502
domain statistics, 6502
domain totals, 6503
domain variable, 6485
donor stratum, 6508
estimated frequencies, 6500
estimated totals, 6500
Fay coefficient, 6480, 6507
Fay's BRR variance estimation, 6507
finite population correction, 6476, 6477, 6492
first-stage sampling rate, 6476
Hadamard matrix, 6480, 6509, 6517

Introduction to Survey Procedures, 259, 262, 269
jackknife, 6506, 6508, 6528
jackknife coefficients, 6508, 6510
jackknife variance estimation, 6508
mean per element, 6495
means, 6495
missing values, 6474, 6491, 6526
number of replicates, 6482, 6506–6508
numerator variable, 6486
ODS table names, 6517
output data sets, 6472, 6509
output jackknife coefficient, 6510
output replicate weights, 6509
output table names, 6517
percentile, 6504
percentiles, 6504
population totals, 6477, 6492
primary sampling units (PSUs), 6493
proportion estimation, 6500
quantile, 6504
quantiles, 6504
quantiles table, 6515
ratio analysis, 6486, 6497
ratio analysis table, 6516
ratios, 6486, 6497
rectangular table, 6476, 6510
replicate weights, 6488
replication methods, 6479, 6506, 6528
sampling rates, 6476, 6492
sampling weights, 6488, 6490
simple random sampling, 6467
stacking table, 6476, 6510
standard deviations of totals, 6501
standard errors, 6496
standard errors of means, 6496
standard errors of ratios, 6497
statistic-keywords, 6477
statistical computations, 6493
statistics table, 6514
stratification, 6489
stratified cluster sample, 6518
stratified sampling, 6469
stratum information table, 6513
t test, 6498
Taylor series variance estimation, 6483, 6496, 6498, 6501
valid observation, 6512
variance estimation, 6493
variance estimation table, 6513
variances of means, 6496
variances of totals, 6501
weighting, 6488, 6490
SURVEYREG procedure, 6534
ADJRSQ, 6555
adjusted R-square, 6571
alpha level, 6543, 6557
analysis of contrasts table, 6580
analysis of estimable functions table, 6580
analysis of variance, 6570
ANOVA, 6555, 6570
ANOVA table, 6579
balanced repeated replication, 6565
BRR, 6565
BRR variance estimation, 6565
classification level table, 6578
classification variables, 6549
clustering, 6550
coefficients of contrast table, 6580
coefficients of estimate table, 6580
computational details, 6563
computational resources, 6573
confidence level, 6543
confidence limits, 6555
contrasts, 6551, 6573
covariance of estimated regression coefficients table, 6580
data summary table, 6576
degrees of freedom, 6568
design effects, 6571
design summary table, 6576
domain analysis, 6573, 6603
domain summary table, 6577
domain variable, 6552
donor stratum, 6567
effect testing, 6570
estimable functions, 6553
Fay coefficient, 6546, 6566
Fay's BRR variance estimation, 6566
finite population correction, 6544, 6545, 6561
first-stage sampling rate, 6544
fit statistics table, 6577
Hadamard matrix, 6546, 6568, 6581
Introduction to Survey Procedures, 259, 263, 269
inverse matrix of X'X, 6579
jackknife, 6567
jackknife coefficients, 6567, 6575
jackknife variance estimation, 6567
linearization method, 6564
missing values, 6543, 6560
MSE, 6571
multiple R-square, 6570
number of replicates, 6548, 6565–6567
output data sets, 6541, 6574
output jackknife coefficient, 6575
output replicate weights, 6575

output table names, 6581
pooled stratum, 6572
population totals, 6545, 6561
primary sampling units (PSUs), 6562
regression coefficients, 6563
regression coefficients table, 6579
regression estimator, 6595
regression estimators, 6588
replicate weights, 6558, 6564
replication methods, 6545, 6564, 6606
root MSE, 6571
sampling rates, 6544, 6561
sampling weights, 6558, 6560
simple random cluster sampling, 6585
simple random sampling, 6535, 6582
singularity level, 6552, 6554, 6556
stratification, 6559
stratified sampling, 6538, 6589
stratum collapse, 6572, 6598
stratum information table, 6578
subpopulation analysis, 6603
Taylor series variance estimation, 6549, 6564
testing effect, 6570
tests of model effects table, 6579
variance estimation, 6564
variance estimation table, 6577
Wald test, 6570, 6573
weighting, 6558, 6560
X'X matrix, 6578
SURVEYSELECT procedure, 6614
Brewer's method, 6652, 6667
certainty size measure, 6625
certainty size proportion, 6626
Chromy's method, 6647, 6651
control sorting, 6622, 6639, 6644, 6664
displayed output, 6661
dollar-unit sampling, 6670
Hanurav-Vijayan method, 6648
initial seed, 6637
Introduction to Survey Procedures, 259, 262, 268
introductory example, 6615
joint selection probabilities, 6627
maximum size measure, 6628
minimum size measure, 6631
missing values, 6644
Murthy's method, 6653
nested sorting, 6644
Neyman allocation, 6641, 6656
ODS table names, 6663
optimal allocation, 6641, 6655
output data sets, 6657
PPS sampling, with replacement, 6650
PPS sampling, without replacement, 6648
PPS sequential sampling, 6651
PPS systematic sampling, 6650
proportional allocation, 6641, 6654, 6673
replicated sampling, 6634, 6664
Sampford's method, 6653
sample selection methods, 6645
sample size, 6636
sample size allocation, 6640, 6654
sampling rate, 6634
secondary input data set, 6656
sequential random sampling, 6647, 6664
serpentine sorting, 6644
simple random sampling, 6616, 6646
size measure, 6639
stratified sampling, 6618, 6640
systematic random sampling, 6647
unrestricted random sampling, 6646
with-replacement sampling, 6645
without-replacement sampling, 6645
survival analysis, 5049, 5127
hazard ratio, 5131
hazards, 5131
Introduction to Regression, 77
MCMC procedure, 3634
median survival times, 5131, 5141
power and sample size (POWER), 4901, 4909, 4975
rank tests, 5128
survival curves, 5131
survival data
Introduction to Regression, 77
survival distribution function
LIFETEST procedure, 3100, 3131, 3149
PHREG procedure, 4579
survival function
LIFEREG procedure, 2993, 3039
SEQDESIGN procedure, 5867
survival models, parametric, 2992
survival plot
ODS Graphics, 622
survival times
PHREG procedure, 4523, 4664, 4666
survivor function, *see* survival distribution function
definition (PHREG), 4579
estimate (PHREG), 4684
estimates (LOGISTIC), 3465
estimates (PHREG), 4543, 4574, 4615, 4683
PHREG procedure, 4523, 4574, 4579
sweep algorithm
REG procedure, 5584
Sweep operator

and generalized inverse (Introduction to Modeling), 71
and log determinant (Introduction to Modeling), 72
elementary operations (Introduction to Modeling), 72
Gauss-Jordan elimination (Introduction to Modeling), 71
pivots (Introduction to Modeling), 72
row operations (Introduction to Modeling), 71
switching model
NLIN procedure, 4324
symmetric and positive definite (SIM2D)
covariance matrix, 6177
symmetric binary variable
DISTANCE procedure, 1485
symmetric two-sided test
SEQDESIGN procedure, 5824
syntax
QUANTTREG procedure, 5372
systematic random sampling
SURVEYSELECT procedure, 6647
systematic trend, *see* surface trend

t distribution
definition of (MCMC), 3539
GLIMMIX procedure, 2168
MCMC procedure, 3515, 3539
t statistic
for equality of means, 7411
t test, 5049, 5086
equal variances, 5086
mean ratio, 5101
MULTTEST procedure, 4201, 4209, 4229
one-sample, 5052
power and sample size (POWER), 4852, 4857, 4871, 4878, 4891, 4898, 4953, 4954, 4962, 4969
Satterthwaite method, 5098
SURVEYMEANS procedure, 6498
two-sample , 5086
unequal variances, 5098
t test for correlation
power and sample size (POWER), 4840, 4843, 4933
t value
CALIS procedure, 928
displaying (CALIS), 965
displaying (TCALIS), 6937
TCALIS procedure, 6895
t-square statistic
CLUSTER procedure, 1246, 1258
table names
GLIMMIX procedure, 2280
HPMIXED procedure, 2839
MIXED procedure, 3999
ODS, 435, 457
table scores, 1731
table templates
ODS, 441
tables
contingency (SURVEYFREQ), 6312
crosstabulation (SURVEYFREQ), 6312, 6358
frequency and crosstabulation (FREQ), 1676, 1697
multiway, 1789, 1791, 1792
multiway (SURVEYFREQ), 6312
one-way frequency, 1787, 1788
one-way frequency (SURVEYFREQ), 6312, 6357
one-way, tests, 1731, 1732
two-way, tests, 1731, 1732
tables (KRIGE2D procedure)
Kriging Information, 2968, 2969
Model Information, 2969
Number of Observations, 2968
tables (SIM2D procedure)
Model Information, 6181
Number of Observations, 6181
Simulation Information, 6181
tables (VARIOGRAM procedure)
Autocorrelation Statistics, 7526, 7568, 7569
Empirical Semivariogram, 7525, 7568, 7569
Number of Observations, 7568
Pairs Information, 7523, 7556, 7557, 7568, 7569
Pairwise Distance Intervals, 7522, 7556, 7557, 7568, 7569, 7577, 7583
PROC VARIOGRAM statements, 7528
TABLES statement, use
CORRESP procedure, 1323
TABULATE procedure, 24
Tarone's adjustment
Breslow-Day test, 1776
Tarone-Ware test for homogeneity
LIFETEST procedure, 3100, 3130
power and sample size (POWER), 4901, 4910, 4975
Taylor series variance estimation
Introduction to Survey Procedures, 267
SURVEYFREQ procedure, 6327
SURVEYLOGISTIC procedure, 6388, 6427
SURVEYMEANS procedure, 6483, 6496, 6498, 6501
SURVEYREG procedure, 6549, 6564
TCALIS procedure

ADF method, 6889, 6891
approximate standard errors, 6729, 6744, 6894, 6937
asymptotic covariance, 6726, 6891
asymptotically distribution free estimation, 6734, 6891
bifactor model, 7005, 7012
chi-square, adjusted, 6904
coefficient of determination, 6740
computational problems, 6932–6934
computational problems, identification, 6927, 6933
constraints, 6746, 6766, 6794, 6884–6886
constraints, program statements, 6814
constraints, programming statements, 6882
degrees of freedom, 6729, 6735
determination coefficients, 6750
determination index, 6912
direct effect, 6751
discrepancy function, 6888
DWLS method, 6892
effects, 6751
EQS Model, 6839
EQS program, 6704
estimation criteria, 6892
estimation methods, 6888, 6892, 6893
FACTOR, 6863
factor analysis model, 6752
factor loadings, 6832
FACTOR Model, 6863
FACTOR procedure, 6753, 6933
factor rotation, 6754
factor scores, 6832, 6835, 6938
fit function, 6888
fitted covariance matrix, 6888
fitted mean vector, 6888
GLS method, 6888
gradient, 6823, 6894, 6923
hessian, 6894
Hessian matrix, 6823, 6897, 6923, 6924
hierarchical factor model, 7005, 7012
higher-order factor model, 7005, 7006
indirect effect, 6751
information matrix, 6894
initial values, 6742, 6922
input data set, 6819
kappa, 6920
kurtosis, 6681, 6727, 6731, 6919, 6921
latent variables, 6681
likelihood ratio test, 6904, 6919
LINEQS, 6839
LINEQS model, 6766
LISMOD, 6847
LISMOD model, 6772
LISREL, 6847
LM tests, 6776
manifest variables, 6681
matrix inversion, 6897
ML method, 6889
MODEL procedure, 6933
modeling languages, 6745, 6811
modification indices, 6729, 6734, 6735, 6895, 6917, 6918, 6938
MSTRUCT, 6877
MSTRUCT Model, 6877
MSTRUCT model, 6792
naming parameters, 6880
naming variables, 6880
ODS graph names, 6952
optimization, 6684, 6730, 6731, 6736–6738, 6923, 6925, 6930
optimization history, 6926
optimization statements, 6718
optimization, initial values, 6922, 6924
optimization, memory problems, 6924
output data sets, 6823
output table names, 6939
parameter names, 6880
PATH, 6870
PATH model, 6799
predicted covariance matrix, 6836, 6936
predicted covariance model matrix, 6933
predicted mean vector, 6936
prefix name, 6768, 6787, 6815
RAM, 6855
RAM model, 6805
reciprocal causation, 6729, 6915
REG procedure, 6933
renaming parameters, 6812
SCORE procedure, 6739, 6741, 6835
second-order factor model, 7006
significance level, 6726
simplicity functions, 6754
singularity criterion, 6743
singularity criterion, covariance matrix, 6726, 6734, 6744
skewness, 6919
squared multiple correlation, 6912, 6937
stability coefficient, 6729, 6915
step length, 6730
structural equation, 6741, 6817
subsidiary group specification statements, 6716
subsidiary model specification statements, 6718
SYSLIN procedure, 6933
SYSNLIN procedure, 6933
t value, 6895, 6937

test indices, constraints, 6734
testing parametric functions, 6814, 6817
total effect, 6751
ULS method, 6888
variable names, 6880
variable selection, 6887
Wald test, probability limit, 6743
weight matrix input, 6731
WLS method, 6889
Template Browser window, 612
template library
SASHELP.Tmplmst, 587
SASUSER.Templat, 445, 587
user-defined, 587
template library, default
ODS Graphics, 553
template modification
ODS Graphics, 610
template primary statement
ODS Graphics, 653
TEMPLATE procedure
ODS, 441, 464, 479
template statement order
ODS Graphics, 653
templates
ODS, 434, 441
Templates window, 557, 583
templates, displaying contents
ODS, 441
templates, modifying
ODS, 444, 464, 472, 479
termination criteria (CALIS), *see* optimization
test components
MIXED procedure, 3939
test data
GLMSELECT procedure, 2739
test for a binomial proportion
SEQDESIGN procedure, 5859
test for a normal mean
SEQDESIGN procedure, 5858
test for a parameter
SEQDESIGN procedure, 5869–5871
test for difference between two normal means
SEQDESIGN procedure, 5860
test for logistic regression parameter
SEQDESIGN procedure, 5816
test for proportional hazards regression parameter
SEQDESIGN procedure, 5816
test for regression parameter
SEQDESIGN procedure, 5815
test for two binomial proportions
SEQDESIGN procedure, 5862, 5864, 5865
test for two survival distributions
SEQDESIGN procedure, 5866
test indices
constraints (CALIS), 860
constraints (TCALIS), 6734
test information
SEQTEST procedure, 6031
test of circularity, 6727
test of compound symmetry, 6727
test of equal covariance matrices, 6727
test of equal variances and equal covariances, 6727
test of sphericity, 6727
test of type H pattern, 6727
test set classification
DISCRIM procedure, 1407
test set validation
PLS procedure, 4789
test-specification for covariance parameters
GLIMMIX procedure, 2127
testable hypothesis
Introduction to Modeling, 66, 68
testing covariance parameters
GLIMMIX procedure, 2127, 2234
testing effect
SURVEYREG procedure, 6570
testing hypotheses
Introduction to Modeling, 66
testing linear hypotheses
LOGISTIC procedure, 3315, 3348
MIANALYZE procedure, 3849, 3859
SURVEYLOGISTIC procedure, 6408, 6434
tests of fixed effects
GLIMMIX procedure, 2278
tests, hypothesis
examples (GLM), 2559
GLM procedure, 2454
tetrachoric correlation coefficient, 1711, 1742
text, adding to plots
ODS Graphics, 647
theophylline data
examples, GLIMMIX, 2415
examples, NLIN, 4331
examples, NLMIXED, 4409
theoretical correlation
INBREED procedure, 2869
theoretical semivariogram models, *see* semivariogram (VARIOGRAM)
thermometer parameterization
Shared Concepts, 374
thin plate spline (approx.)
GLIMMIX procedure, 2250
thin-plate smoothing splines, 7057
bootstrap, 7094, 7098
large data sets, 7089
theoretical foundation, 7073

thinning of MCMC
Introduction to Bayesian Analysis, 155
Markov chain Monte Carlo, 155
three-way multidimensional scaling
MDS procedure, 3704
threshold response rate, 5258
ties
checking for in CLUSTER procedure, 1244
MDS procedure, 3721
PHREG procedure, 4524, 4529, 4572, 4588, 4632, 4637
time intervals
PHREG procedure, 4638
time requirements
ACECLUS procedure, 681
CLUSTER procedure, 1260
FACTOR procedure, 1574, 1578
VARCLUS procedure, 7476, 7479
time series data
REG procedure, 5581
time-dependent covariates
PHREG procedure, 4524, 4530, 4537, 4541, 4573, 4575
timing breakdown
GLMSELECT procedure, 2748
Tobit model
LIFEREG procedure, 2994, 3069
Toeplitz structure
example (MIXED), 4056
GLIMMIX procedure, 2202
MIXED procedure, 3959
tolerance, *see* angle tolerance (VARIOGRAM), *see* lag tolerance (VARIOGRAM)
angle (VARIOGRAM), 7537
tooltips
ODS Graphics, 601
TOST equivalence test
TTEST procedure, 7408, 7452
total covariance matrix
MIANALYZE procedure, 3858
total effect
TCALIS procedure, 6751
total variance
MI procedure, 3794
MIANALYZE procedure, 3856
TPF basis
GLIMMIX procedure, 388
GLMSELECT procedure, 388
TPSPLINE procedure
bootstrap, 7094, 7098
computational formulas, 7073
large data sets, 7089
nonhomogeneous variance, 7076
ODS table names, 7076
partial spline model, 7077
smoothing parameter, 7076
smoothing penalty, 7087
trace output
ODS, 437
ODS Graphics, 610
trace plot
MI procedure, 3789
trace W method, *see* Ward's method
traditional graphics
LIFETEST procedure, 3111
REG procedure, 5547
traditional graphics (LIFETEST)
description, 3115
local annotate, 3116
training data set
DISCRIM procedure, 1384
transformation
MI procedure, 3767
transformation matrix
orthonormalizing, 708, 2466
transformation options
PRINQUAL procedure, 5216
TRANSREG procedure, 7147
transformation standardization
TRANSREG procedure, 7158
transformations
affine(DISTANCE), 1484
ANOVA procedure, 717
cluster analysis, 1230
for multivariate ANOVA, 706, 2464
identity(DISTANCE), 1485
linear(DISTANCE), 1484
many-to-one(DISTANCE), 1484
MDS procedure, 3704, 3705, 3716, 3717, 3720, 3723–3725
monotone increasing(DISTANCE), 1484
oblique, 1537, 1541
one-to-one(DISTANCE), 1484
orthogonal, 1537, 1541
power(DISTANCE), 1484
repeated measures, 2537–2539
strictly increasing(DISTANCE), 1484
transformations for confidence intervals
LIFETEST procedure, 3114
transformations for repeated measures
GLM procedure, 2483
transformed data
MDS procedure, 3727
transformed distances
MDS procedure, 3727
transforming ordinal variables to interval
DISTANCE procedure, 1485
TRANSREG procedure

TYPE, 7272
additive models, 7160
algorithms, 7162
alpha level, 7160
ANOVA, 7289
ANOVA codings, 7230
ANOVA table, 7166, 7264
ANOVA table in OUTTEST= data set, 7275
asterisk (*) operator, 7139
at sign (@) operator, 7139
B-spline basis, 7142, 7264
bar (|) operator, 7139
Box-Cox alpha, 7156
Box-Cox convenient lambda, 7156
Box-Cox convenient lambda list, 7156
Box-Cox example, 7316
Box-Cox geometric mean, 7156
Box-Cox lambda, 7156
Box-Cox parameter, 7149
Box-Cox transformations, 7181
CANALS method, 7162
canonical correlation, 7171, 7180
canonical variables, 7171
casewise deletion, 7164
cell-means coding, 7155, 7181, 7231
center-point coding, 7153, 7235, 7242
centering, 7157, 7298
character OPSCORE variables, 7253
choice experiments, 7297
CLASS variables, prefix, 7161
classification variables, 7142, 7155
coefficients, redundancy, 7177
confidence limits, 7160, 7171, 7172, 7174
confidence limits, individual, 7171
confidence limits, mean, 7171
confidence limits, prefix, 7171, 7172, 7174
conjoint analysis, 7168, 7180, 7329, 7333
constant transformations, avoiding, 7253
constant variables, 7164, 7253
degrees of freedom, 7265
dependent variable list, 7174
dependent variable name, 7173
design matrix, 7173
details of model, 7161
deviations-from-means coding, 7153, 7181, 7235, 7242, 7293
dummy variable creation, 7126, 7141, 7142, 7155, 7173, 7174, 7230–7233, 7235, 7237, 7239, 7240, 7242, 7244–7248, 7293, 7296–7298
duplicate variable names, 7275
effect coding, 7127, 7153, 7235, 7242, 7293
excluded observations, 7255
excluding nonscore observations, 7167
expansions, 7141
explicit intercept, 7254
frequency variable, 7138
full-rank coding, 7153
GLMMOD alternative, 7173, 7293
history, iteration, 7162
hypothesis tests, 7166, 7264
ID variables, 7138
ideal point model, 7180
ideal point models, 7345
identity transformation, 7147
implicit intercept, 7254
independent variable list, 7174
individual model fitting, 7162
initialization, 7161, 7252
interaction effects, 7139, 7180, 7181
interactions, quantitative, 7181
intercept, 7254
intercept, none, 7163
iteration histories, displaying, 7162
iterations, 7251
iterations, maximum number of, 7162
iterations, restarting, 7165, 7252
knots, 7150, 7151, 7192, 7262
knots, after expansion, 7157
knots, exterior, 7208
less-than-full-rank model, 7155, 7180, 7237
leverage, 7174
linear regression, 7179
linear transformation, 7145, 7260
macros, 7174
main effects, 7139, 7180, 7181
maximum redundancy analysis, 7162
METHOD=MORALS rolled output data set, 7272
METHOD=MORALS variable names, 7275
metric conjoint analysis, 7333
missing value restoration option, 7177
missing values, 7162, 7164, 7249, 7250, 7255
monotone transformations, 7179
monotonic B-spline transformation, 7146, 7261
monotonic transformation, ties not preserved, 7146, 7260
monotonic transformation, ties preserved, 7146, 7260
MORALS dependent variable name, 7173
MORALS method, 7162
multiple redundancy coefficients, 7177
multiple regression, 7179
multivariate multiple regression, 7179
names of variables, 7157
nonlinear fit functions, 7209

nonlinear fit transformations, 7144
nonlinear regression functions, 7179
nonlinear transformations, 7179
nonmetric conjoint analysis, 7329
nonoptimal transformations, 7143
ODS Graph names, 7301
optimal scaling, 7259
optimal scoring, 7146, 7260
optimal transformations, 7145
order of CLASS levels, 7154, 7165
orthogonal coding, 7154
OUT= data set, 7168, 7272
output table names, 7299
output, limiting, 7166
OUTTEST= data set, 7133
part-worth utilities, 7329
passive observations, 7255
penalized B-spline example, 7323
penalized B-spline lambda, 7152
penalized B-spline t-options, 7152
penalized B-splines, 7219
piecewise polynomial splines, 7143, 7264
point models, 7255
polynomial-spline basis, 7143, 7264
predicted values, 7177
preference mapping, 7180, 7345
preference models, 7172
prefix, canonical variables, 7171
prefix, redundancy variables, 7178
prefix, residuals, 7177
random initializations, 7252
redundancy analysis, 7162, 7177, 7178, 7180, 7256
redundancy analysis, standardization, 7178
reference level, 7155, 7165, 7178
reference-cell coding, 7155, 7181, 7233, 7240
reflecting the transformation, 7158
regression functions, separate, 7213
regression table, 7166
regression table in OUTTEST= data set, 7275
reiteration, 7165, 7252
renaming and reusing variables, 7157
residuals, 7178
residuals, prefix, 7177
separate regression functions, 7213
short output, 7166
singularity criterion, 7166
smoothing spline transformation, 7147, 7222
spline t-options, 7150
spline transformation, 7146, 7261
splines, 7142, 7143, 7192, 7264, 7278, 7307
  standardization, redundancy variables, 7178
  standardization, transformation, 7158, 7167
  standardizing, 7158
  star (*) operator, 7139
  transformation options, 7147
  transformation standardization, 7158, 7167
  Type II sums of squares, 7166
  types of observations, 7167
  utilities, 7168, 7329, 7333
  utilities in OUTTEST= data set, 7275
  variable list macros, 7174
  variable names, 7274
  vector preference models, 7172
  weight variable, 7179
  z scores, 7158
treatments in a design
  specifying in PLAN procedure, 4742
tree diagram
  binary tree, 7356
  branch, 7356
  children, 7356
  definitions, 7356
  leaves, 7356
  node, 7356
  parent, 7356
  root, 7356
tree diagrams
  cluster analysis, 7356
TREE procedure, 7355
  missing values, 7369
  OUT= data sets, 7369
  output data sets, 7369
  output table names, 7370
trend, *see* surface trend (VARIOGRAM)
trend comparisons
  NLIN procedure, 4331
trend test, 1762, 1818
trend tests
  LIFETEST procedure, 3100, 3129, 3145, 3153
TRIM= option
  and other options (CLUSTER), 1242, 1246
triweight kernel (DISCRIM), 1405
truncated distributions
  MCMC procedure, 3544
Truncated power function
  spline basis (Shared Concepts), 388
truncated power function basis
  GLIMMIX procedure, 388
  GLMSELECT procedure, 388
trust region (TR), 4371
trust region method
  Shared Concepts, 406
trust-region algorithm

CALIS procedure, 856, 868, 945
TCALIS procedure, 6730, 6737, 6923
TTEST procedure
AB/BA crossover, 7410
AB/BA crossover design, 7442
alpha level, 7401
Behrens-Fisher problem, 7417
Cochran and Cox *t* approximation, 7402, 7418
compared to other procedures, 2435
computational methods, 7411
confidence intervals, 7402
crossover design, 7410, 7442
difference test, 7407
equivalence test, 7408, 7452
folded form *F* statistic, 7418
graphics, 7427
input data set, 7411
introductory example, 7393
lognormal distribution, 7402, 7452
lower one-sided *t* test, 7407
missing values, 7411
normal distribution, 7402
ODS graph names, 7427
ODS Graphics, 7427
ODS table names, 7426
one-sided *t* test, 7407
paired *t* test, 7409
paired comparisons, 7392, 7437
ratio test, 7407
Satterthwaite's approximation, 7417
statistical graphics, 7427
TOST equivalence test, 7408, 7452
two-sided *t* test, 7407
uniformly most powerful unbiased test, 7414
upper one-sided *t* test, 7407
Tucker and Lewis's Reliability Coefficient, 1580
Tukey's adjustment
GLIMMIX procedure, 2144
GLM procedure, 2459
LIFETEST procedure, 3127
MIXED procedure, 3924
Tukey's studentized range test, 714, 2473, 2518, 2520
Tukey-Kramer test, 714, 2473, 2518, 2520
tuning
MCMC procedure, 3525
two-sample *t* test, 5086, 7431
two-sample *t*-test
power and sample size (POWER), 4821, 4891, 4898, 4899, 4969, 4971, 4972, 5012
two-sample *t* test
TTEST procedure, 7392
two-sample test for binomial proportions
SEQDESIGN procedure, 5813
two-sample test for mean difference
SEQDESIGN procedure, 5812
two-sample tests
SEQDESIGN procedure, 5860
two-sided *t* test
TTEST procedure, 7407
two-sided asymmetric design
SEQDESIGN procedure, 5878
two-sided repeated confidence intervals
SEQTEST procedure, 6023
two-sided test
SEQDESIGN procedure, 5824
two-stage density linkage
CLUSTER procedure, 1240, 1256
Type 1 analysis
GENMOD procedure, 1899, 1978
Type 1 error rate
repeated multiple comparisons, 2516
Type 1 estimation
MIXED procedure, 3908
Type 1 testing
MIXED procedure, 3934
Type 2 estimation
MIXED procedure, 3908
Type 2 testing
MIXED procedure, 3934
Type 3 analysis
GENMOD procedure, 1899, 1979
Type 3 estimation
MIXED procedure, 3908
type 3 test
PHREG procedure, 4634
Type 3 testing
MIXED procedure, 3934, 3998
Type H covariance structure, 2536
Type I error, 413, 415, 4916
SEQDESIGN procedure, 5804, 5836
Type I error probability
SEQDESIGN procedure, 5827, 5837, 5838
Type I sum of squares
computing in GLM, 2542
displaying (GLM), 2477
estimable functions for, 2475
estimable functions for (GLM), 2498
examples, 2567
Type I testing
GLIMMIX procedure, 2171
Type II error, 413, 415, 4916
SEQDESIGN procedure, 5804, 5836
Type II error probability
SEQDESIGN procedure, 5827, 5837, 5838
Type II sum of squares

computing in GLM, 2542
displaying (GLM), 2477
estimable functions for, 2475
estimable functions for (GLM), 2499
examples, 2567
Type II sums of squares
TRANSREG procedure, 7166
Type II testing
GLIMMIX procedure, 2171
Type III sum of squares
displaying (GLM), 2477
estimable functions for, 2475
estimable functions for (GLM), 2501
examples, 2567
Type III testing
GLIMMIX procedure, 2171
type III tests
HPMIXED procedure, 2831
Type IV sum of squares
computing in GLM, 2542
displaying (GLM), 2477
estimable functions for, 2475
estimable functions for (GLM), 2501
examples, 2567
TYPE= data sets
FACTOR procedure, 1569
TYPE=ACE
Data set option, 7609
TYPE=BOXPLOT
Data set option, 7609
TYPE=CHARTSUM
Data set option, 7609
TYPE=CORR
Data set option, 7609
TYPE=COV
Data set option, 7613
TYPE=CSSCP
Data set option, 7613
TYPE=DISTANCE
Data set option, 7614
TYPE=EST
Data set option, 7614
TYPE=LINEAR
Data set option, 7616
TYPE=LOGISMOD
Data set option, 7616
TYPE=MIXED
Data set option, 7616
TYPE=QUAD
Data set option, 7616
TYPE=RAM
Data set option, 7617
TYPE=SSCP
Data set option, 7617
TYPE=TREE
Data set option, 7618
TYPE=UCORR
Data set option, 7618
TYPE=UCOV
Data set option, 7619
TYPE=WEIGHT
Data set option, 7619

UDS statement
MCMC procedure, 3518
ULS method
TCALIS procedure, 6888
ultra-Heywood cases, FACTOR procedure, 1576
ultrametric, definition, 1259
unbalanced data
caution (ANOVA), 692
unbalanced design
GLM procedure, 2435, 2512, 2540, 2565, 2595
NESTED procedure, 4259
uncertainty
KRIGE2D procedure, 2930
power and sample size, 419
VARIOGRAM procedure, 7518
uncertainty coefficients, 1736, 1744, 1745
uncorrelated
random variables (Introduction to Modeling), 59
unfolding
MDS procedure, 3704
unified family method
SEQDESIGN procedure, 5791, 5807, 5828, 5839, 5875
unified family method shape parameters
SEQDESIGN procedure, 5840
unified family triangular method
SEQDESIGN procedure, 5807, 5843, 5875
uniform distribution
definition of (MCMC), 3540
MCMC procedure, 3515, 3540
uniform kernel (DISCRIM), 1404
uniform-kernel estimation
CLUSTER procedure, 1246, 1252
uniformly most powerful unbiased test
TTEST procedure, 7414
unique factor
defined for factor analysis, 1535
univariate distributions, example
MODECLUS procedure, 4128
UNIVARIATE procedure, 24
Introduction to Nonparametric Analysis, 292, 295
univariate tests

repeated measures, 2535, 2536
unknown or missing parents
INBREED procedure, 2874
unrestricted random sampling
SURVEYSELECT procedure, 6646
unsquared Euclidean distances, 1242, 1244
unstructured correlations
MIXED procedure, 3959
unstructured covariance
GLIMMIX procedure, 2203
unstructured covariance matrix
GLIMMIX procedure, 2197
MIXED procedure, 3959
unweighted least squares factor analysis, 1534
unweighted pair-group clustering, *see* average linkage, *see* centroid method
update methods
NLMIXED procedure, 4371
UPGMA, *see* average linkage
UPGMC, *see* centroid method
upper one-sided *t* test
TTEST procedure, 7407
user defined sampler statement
MCMC procedure, 3518
user-defined distribution
MCMC procedure, 3514
user-defined samplers
MCMC procedure, 3520, 3672
using alternate forms, 5078
using the IF-ELSE logical control
MCMC procedure, 3583
utilities
TRANSREG procedure, 7329, 7333

V matrix
GLIMMIX procedure, 2205
MIXED procedure, 3953
validation data
GLMSELECT procedure, 2739
value lists
GLMPOWER procedure, 2660
POWER procedure, 4919
Van der Waerden scores
NPAR1WAY procedure, 4471
VAR statement, use
CORRESP procedure, 1323
VARCLUS procedure, *see also* TREE procedure
alternating least squares, 7463
analyzing data in groups, 7474
centroid component, 7470
cluster components, 7462
cluster splitting, 7462, 7463, 7468, 7471, 7473
cluster, definition, 7461
computational resources, 7479
controlling number of clusters, 7472
eigenvalues, 7462, 7463, 7471
how to choose options, 7476
initializing clusters, 7470
interpreting output, 7479
iterative reassignment, 7462, 7463
MAXCLUSTERS= option, using, 7476
MAXEIGEN= option, using, 7476
memory requirements, 7479
missing values, 7476
multiple group component analysis, 7472
nearest component sorting phase, 7463
number of clusters, 7462, 7463, 7468, 7471–7473
orthoblique rotation, 7462, 7470
output data sets, 7472, 7477
output table names, 7481
OUTSTAT= data set, 7472, 7477
OUTTREE= data set, 7478
PROPORTION= option, using, 7476
search phase, 7463
splitting criteria, 7462, 7463, 7468, 7471, 7473
stopping criteria, 7468
time requirements, 7476, 7479
TYPE=CORR data set, 7477
VARCOMP procedure
classification variables, 7499
compared to MIXED procedure, 3895
compared to other procedures, 2435
computational details, 7502
confidence level, 7500
Confidence limits, 7505
confidence limits, 7500
dependent variables, 7494, 7499
estimation methods, 7498
example (MIXED), 4032
fixed effects, 7494, 7500
fixed-effects model, 7501
gauge R&R, 7498, 7503
input data sets, 7498
introductory example, 7494
maximum likelihood, 7498
methods of estimation, 7494, 7508
missing values, 7500
mixed model, 7501
negative variance components, 7501
ODS table names, 7507
random effects, 7493, 7500
random-effects model, 7501
relationship to PROC MIXED, 7508
repeatability and reproducibility, 7494
restricted maximum likelihood, 7498

variability, 7494
variance component, 7501
variability
VARCOMP procedure, 7494
variable (PHREG)
censoring, 4528
variable importance for projection, 4801
variable list macros
TRANSREG procedure, 7174
variable selection
CALIS procedure, 942
discriminant analysis, 6267
TCALIS procedure, 6887
variable-radius kernels
MODECLUS procedure, 4110
variable-reduction method, 7462
variables, *see also* classification variables
frequency (PRINQUAL), 5211
renaming (PRINQUAL), 5219
reusing (PRINQUAL), 5219
weight (PRINQUAL), 5220
variables, unaddressed
INBREED procedure, 2867, 2868
variance
matrix, definition (Introduction to Modeling), 58
of random variable (Introduction to Modeling), 57
variance component
VARCOMP procedure, 7501
variance components, 4254
MIXED procedure, 3893, 3959
variance estimation
BRR (Introduction to Survey Procedures), 267
BRR (SURVEYFREQ), 6334
BRR (SURVEYLOGISTIC), 6429
BRR (SURVEYMEANS), 6506
BRR (SURVEYREG), 6565
Introduction to Survey Procedures, 267
jackknife (Introduction to Survey Procedures), 267
jackknife (SURVEYFREQ), 6337
jackknife (SURVEYLOGISTIC), 6431
jackknife (SURVEYMEANS), 6508
jackknife (SURVEYREG), 6567
SURVEYFREQ procedure, 6326
SURVEYLOGISTIC procedure, 6427
SURVEYMEANS procedure, 6493
SURVEYREG procedure, 6564
Taylor series (Introduction to Survey Procedures), 267
Taylor series (SURVEYFREQ), 6327
Taylor series (SURVEYLOGISTIC), 6388, 6427
Taylor series (SURVEYMEANS), 6483, 6496, 6498, 6501
Taylor series (SURVEYREG), 6549, 6564
variance function
GENMOD procedure, 1898
GLIMMIX procedure, 2208
user-defined (GLIMMIX), 2208
variance inflation factors (VIF)
REG procedure, 5456
variance of means
LATTICE procedure, 2984
variance ratios
HPMIXED procedure, 2827
MIXED procedure, 3944, 3952
variance-covariance matrix
definition (Introduction to Modeling), 59
variances
FACTOR procedure, 1563
ratio of, 7431
ratio of (TTEST), 7418
variances of totals
SURVEYMEANS procedure, 6501
variances, test for equal, 1395
varimax method, 1534, 1561, 1562, 6755, 6756
VARIOGRAM procedure
angle classes, 7536–7539, 7549, 7550
angle tolerance, 7534, 7537, 7539, 7549, 7550
anisotropy, 7537, 7574
autocorrelation, 7518, 7560
autocorrelation weights, 7561
bandwidth, 7536, 7539, 7552
collocation, 7548, 7564
confidence level, 7534
confidence limits, 7536
correlation measures, 7557
covariance, 7518, 7546, 7596
cutoff distance, 7538
DATA= data set, 7529
distance classes, 7553, 7555, 7556
distance classification, 7551
distance interval, 7522
ergodicity, 7547
estimation, 7519
examples, 7520, 7570, 7574, 7587, 7596, 7600
exploratory data analysis, 7520
Geary *c* coefficient, 7518, 7534, 7562
input data set, 7529
isotropy, 7537
lag, 7522, 7553, 7555
lag distance, 7536, 7553

lag tolerance, 7536
measures of spatial continuity, 7518, 7544, 7596
modeling, 7518
Moran *I* coefficient, 7518, 7534, 7562
nested models, 7543
normality assumption, 7534, 7562
nugget effect, 7543
ODS graph names, 7569
ODS Graphics, 7530
ODS table names, 7569
ordinary kriging, 7519
OUTACWEIGHTS= data set, 7529
OUTDIST= data set, 7529, 7555
OUTPAIR= data set, 7529
output data sets, 7529, 7530, 7565, 7566
OUTVAR= data set, 7530
pairwise distance, 7522, 7536
panel plots, 7530
point pairs, 7522, 7545, 7548
prediction, 7519
pseudo-semivariance, 7546
pseudo-semivariogram, 7546, 7587
randomization assumption, 7534, 7562
semivariance, 7518, 7544, 7596
semivariance computation, 7558
semivariance, classical, 7525, 7545
semivariance, empirical, 7545
semivariance, robust, 7525, 7538, 7545
semivariogram, 7518, 7545
semivariogram analysis, 7519
semivariogram and covariogram, 7596
semivariogram computation, 7519
semivariogram effective range, 7541
semivariogram parameters, 7540
semivariogram range, 7540
semivariogram sill, 7540
semivariogram, empirical, 7519, 7545
semivariogram, robust, 7545
spatial continuity, 7518, 7519, 7544, 7546, 7587, 7596
spatial prediction, 7518, 7546
spatial random field, 7540, 7544, 7546, 7547, 7560, 7587
square root difference cloud, 7600
standard errors, 7519
stationarity, 7540, 7546, 7598
stochastic analysis, 7518
surface trend, 7518, 7522, 7546, 7559, 7574, 7578
theoretical semivariogram models, 7519, 7522, 7540, 7542, 7570, 7584, 7593
uncertainty, 7518
VARIOGRAM procedure, tables
Autocorrelation Statistics, 7526, 7568, 7569
Empirical Semivariogram, 7525, 7568, 7569
Number of Observations, 7568
Pairs Information, 7523, 7556, 7557, 7568, 7569
Pairwise Distance Intervals, 7522, 7556, 7557, 7568, 7569, 7577, 7583
PROC VARIOGRAM statements, 7528
vector preference models
TRANSREG procedure, 7172
VIF, *see* variance inflation factors
VIP, 4801

Wald chi-square test
SURVEYFREQ procedure, 6352
Wald distribution
definition of (MCMC), 3541
MCMC procedure, 3515, 3541
Wald log-linear chi-square test
SURVEYFREQ procedure, 6354
Wald test
GLIMMIX procedure, 2278
mixed model (MIXED), 3978, 4023
MIXED procedure, 3997, 3998
modification indices (CALIS), 860, 954
modification indices (TCALIS), 6734, 6918
PHREG procedure, 4577, 4598, 4600, 4634, 4667
probability limit (CALIS), 867
probability limit (TCALIS), 6743
PROBIT procedure, 5311
SURVEYREG procedure, 6570, 6573
Wald tests of covariance parameters
GLIMMIX procedure, 2134
Waller-Duncan test, 714, 2473, 2523
error seriousness ratio, 713, 2472
examples, 2560
multiple comparison (ANOVA), 733
Wampler data set, 4517
Ward's minimum-variance method
CLUSTER procedure, 1240, 1257
Wei-Lin-Weissfeld model
PHREG procedure, 4602
Weibull distribution, 2992, 3022, 3039
definition of (MCMC), 3541
MCMC procedure, 3515, 3541
weight matrix input
TCALIS procedure, 6731
weight variable
PRINQUAL procedure, 5220
weighted average linkage
CLUSTER procedure, 1240, 1255
weighted Euclidean distance
MDS procedure, 3705, 3712, 3724

weighted Euclidean model
MDS procedure, 3704, 3712
weighted kappa coefficient, 1765, 1767
weighted least squares
CATMOD procedure, 1096, 1124
formulas (CATMOD), 1170
MDS procedure, 3723
normal equations (GLM), 2487
weighted means
GLM procedure, 2527
weighted pair-group methods, *see* McQuitty's similarity analysis, *see* median method
weighted percentiles, 6233
weighted product-moment correlation coefficients
CANCORR procedure, 1044
weighted Schoenfeld residuals
PHREG procedure, 4574, 4613
weighted score residuals
PHREG procedure, 4614
weighted-group method, *see* centroid method
weighting, *see also* sampling weights
GLIMMIX procedure, 2206
HPMIXED procedure, 2832
Introduction to Survey Procedures, 266
MIXED procedure, 3968
SURVEYFREQ procedure, 6320, 6321
SURVEYLOGISTIC procedure, 6406, 6409
SURVEYMEANS procedure, 6488, 6490
SURVEYREG procedure, 6558, 6560
weighting variables
FACTOR procedure, 1576
Welch *t* test
power and sample size (POWER), 4891, 4899, 4971
Welch's ANOVA, 714, 2473
homogeneity of variance tests, 2527
using homogeneity of variance tests, 2605
Welsch's multiple range test, 713, 2472, 2523
examples, 2560
WHERE statement
GLM procedure, 2491
Whitehead method
SEQDESIGN procedure, 5791, 5829, 5845, 5875
Whitehead one-sided asymmetric design
SEQDESIGN procedure, 5846
Whitehead one-sided symmetric design
SEQDESIGN procedure, 5845
Whitehead two-sided design
SEQDESIGN procedure, 5846
Whitehead's double-triangular design
SEQDESIGN procedure, 5806
Whitehead's triangular design
SEQDESIGN procedure, 5806
Whitehead's triangular method
SEQDESIGN procedure, 5828
width, confidence intervals, 416, 4916
Wilcoxon rank-sum test, *see* Wilcoxon-Mann-Whitney (rank-sum) test
Wilcoxon scores
NPAR1WAY procedure, 4470
Wilcoxon signed rank test
Introduction to Nonparametric Analysis, 295
Wilcoxon test for association
LIFETEST procedure, 3101
Wilcoxon test for homogeneity
LIFETEST procedure, 3100, 3130, 3142
Wilcoxon-Mann-Whitney (rank-sum) test
power and sample size (POWER), 4911, 4978
Wilcoxon-Mann-Whitney test
power and sample size (POWER), 4915, 5039
Wilks' lambda, 706, 2464, 2536
Williams' method
overdispersion (LOGISTIC), 3343
with-replacement sampling
SURVEYSELECT procedure, 6645
within-cluster SSCP matrix
ACECLUS procedure, 660
within-imputation covariance matrix
MIANALYZE procedure, 3857
within-imputation variance
MI procedure, 3794
MIANALYZE procedure, 3856
within-subject factors
repeated measures, 2481, 2535
without-replacement sampling
SURVEYSELECT procedure, 6645
WLS method
TCALIS procedure, 6889
Wong's hybrid method
CLUSTER procedure, 1242, 1252
working correlation matrix
GENMOD procedure, 1959–1961, 1987
worst linear function of parameters
MI procedure, 3789
WPGMA, *see* McQuitty's similarity analysis
WPGMC, *see* median method

XDATA= data sets
LIFEREG procedure, 3051

Yule's Q statistic, 1738

z scores
TRANSREG procedure, 7158

z test
power and sample size (POWER), 4844, 4850, 4935, 4937
zero variance component estimates
MIXED procedure, 4011
zero-inflated Poisson
distribution (GENMOD), 1966
models (GENMOD), 1985
zeros, structural and random
FREQ procedure, 1770
zeros, structural and sampling
CATMOD procedure, 1164
examples (CATMOD), 1195, 1200
zonal anisotropy
KRIGE2D procedure, 2959

Syntax Index

AB option
  EXACT statement (NPAR1WAY), 4463
  OUTPUT statement (NPAR1WAY), 4465
  PROC NPAR1WAY statement, 4457
ABSCONV option
  NLOPTIONS statement (GLIMMIX), 393
  NLOPTIONS statement (HPMIXED), 393
  NLOPTIONS statement (TCALIS), 393
ABSCONV= option
  NLOPTIONS statement (CALIS), 890
  PROC NLMIXED statement, 4357
ABSENT= option
  PROC DISTANCE statement, 1493
  VAR statement, 1503
ABSFCONV option
  MODEL statement (LOGISTIC), 3291
  MODEL statement (SURVEYLOGISTIC), 6398
  NLOPTIONS statement (GLIMMIX), 393
  NLOPTIONS statement (HPMIXED), 393
  NLOPTIONS statement (TCALIS), 393
ABSFCONV= option
  NLOPTIONS statement (CALIS), 892
  PROC NLMIXED statement, 4357
ABSFTOL= option
  NLOPTIONS statement (CALIS), 892
ABSGCONV option
  NLOPTIONS statement (GLIMMIX), 393
  NLOPTIONS statement (HPMIXED), 393
  NLOPTIONS statement (TCALIS), 393
ABSGCONV= option
  NLOPTIONS statement (CALIS), 892, 895
  PROC NLMIXED statement, 4358
ABSGTOL option
  NLOPTIONS statement (GLIMMIX), 393
  NLOPTIONS statement (HPMIXED), 393
  NLOPTIONS statement (TCALIS), 393
ABSGTOL= option
  NLOPTIONS statement (CALIS), 892, 895
ABSOLUTE option
  PROC ACECLUS statement, 675
  PROC MIXED statement, 3905, 3996
ABSORB statement
  ANOVA procedure, 703
  GLM procedure, 2452
ABSPCONV option
  PROC GLIMMIX statement, 2095
ABSTOL option
  NLOPTIONS statement (GLIMMIX), 393
  NLOPTIONS statement (HPMIXED), 393
  NLOPTIONS statement (TCALIS), 393
ABSXCONV option
  NLOPTIONS statement (GLIMMIX), 393
  NLOPTIONS statement (HPMIXED), 393
  NLOPTIONS statement (TCALIS), 393
ABSXCONV= option
  NLOPTIONS statement (CALIS), 892
  PROC NLMIXED statement, 4358
ABSXTOL option
  NLOPTIONS statement (GLIMMIX), 393
  NLOPTIONS statement (HPMIXED), 393
  NLOPTIONS statement (TCALIS), 393
ABSXTOL= option
  NLOPTIONS statement (CALIS), 892
ACCEPTTOL= option
  PROC MCMC statement, 3497
ACCRUALTIME= option
  TWOSAMPLESURVIVAL statement (POWER), 4902
ACECLUS procedure
  syntax, 673
ACECLUS procedure, BY statement, 678
ACECLUS procedure, FREQ statement, 679
ACECLUS procedure, PROC ACECLUS statement, 674
  ABSOLUTE option, 675
  CONVERGE= option, 675
  DATA= option, 675
  INITIAL= option, 675
  MAXITER= option, 675
  METHOD= option, 676
  METRIC= option, 676
  MPAIRS= option, 676
  N= option, 676
  NOPRINT option, 676
  OUT= option, 676
  OUTSTAT= option, 677
  P= option, 677
  PERCENT= option, 677
  PP option, 677
  PREFIX= option, 677
  PROPORTION= option, 677
  QQ option, 677
  SHORT option, 677
  SINGULAR= option, 677
  T= option, 678

THRESHOLD= option, 678
ACECLUS procedure, VAR statement, 679
ACECLUS procedure, WEIGHT statement, 679
ACF option
MCMC statement (MI), 3761
ACFPLOT option
MCMC statement (MI), 3757
ACOV option
MODEL statement (REG), 5472
ACOVMETHOD= option
MODEL statement (REG), 5472
ACTUAL option
MODEL statement (RSREG), 5731
ADAPTIVEFDR option
PROC MULTTEST statement, 4189, 4218
ADAPTIVEHOCHBERG option
PROC MULTTEST statement, 4188
ADAPTIVEHOLM option
PROC MULTTEST statement, 4189
ADD statement, REG procedure, 5467
ADD= option
PROC DISTANCE statement, 1493
PROC STDIZE statement, 6222
ADDCELL= option
MODEL statement (CATMOD), 1120
ADDITIVE option
MODEL statement (TRANSREG), 7160
PROC GAM statement, 1840
ADJACENTPAIRS option
ROCCONTRAST statement (LOGISTIC), 3311
ADJBOUND= option
MODEL statement (SURVEYLOGISTIC), 6402
ADJDFE= option
ESTIMATE statement (GLIMMIX), 2135
LSMEANS statement (GLIMMIX), 2143
LSMEANS statement (MIXED), 3923
LSMESTIMATE statement (GLIMMIX), 2156
ADJRSQ
STATS= option (GLMSELECT), 2713
ADJRSQ option
MODEL statement (REG), 5472
MODEL statement (SURVEYREG), 6555
ADJUST= option
ESTIMATE statement (GLIMMIX), 2135
LSMEANS statement (GLIMMIX), 2144
LSMEANS statement (GLM), 2458
LSMEANS statement (MIXED), 3924
LSMESTIMATE statement (GLIMMIX), 2156
STRATA statement (LIFETEST), 3127
ADPREFIX= option
OUTPUT statement (TRANSREG), 7170
AFTER option
MODEL statement (TRANSREG), 7157
AGGREGATE= option
MODEL statement (GENMOD), 1947
MODEL statement (LOGISTIC), 3292
MODEL statement (PROBIT), 5296
AGREE option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1695
TABLES statement (FREQ), 1700
TEST statement (FREQ), 1723
AIC
STATS= option (GLMSELECT), 2713
AIC option
MODEL statement (REG), 5472
MODEL statement (TRANSREG), 7152
PLOT statement (REG), 5493
AICC
STATS= option (GLMSELECT), 2713
AICC option
MODEL statement (TRANSREG), 7152
AIPREFIX option
OUTPUT statement (TRANSREG), 7170
AJCHI option
OUTPUT statement (FREQ), 1695
ALG= option
PRIOR statement (MIXED), 3947
ALGORITHM option
PROC QUANTREG statement, 5372
ALGORITHM= option
PROC PLS statement, 4776
ALIASING option
MODEL statement (GLM), 2475, 2610
ALL
DETAILS=STEPS option (GLMSELECT), 2708
ALL effect
MANOVA statement (ANOVA), 706
MANOVA statement, H= option (GLM), 2464
ALL option
MODEL statement (LOESS), 3208
MODEL statement (REG), 5472
NLOPTIONS statement (CALIS), 896
OUTPUT statement (FREQ), 1695
PROC CALIS statement, 863
PROC CANCORR statement, 1039
PROC CANDISC statement, 1070
PROC CORRESP statement, 1324
PROC DISCRIM statement, 1391
PROC FACTOR statement, 1550
PROC MODECLUS statement, 4103
PROC REG statement, 5454

PROC STEPDISC statement, 6275
PROC TCALIS statement, 6740
TABLES statement (FREQ), 1701
all option
PROC GAM statement, 1839
ALLLABEL= option
BOXPLOT procedure, 771
ALLNEWPARMS option
REFMODEL statement, 6812
ALLOBS option
PAINT statement (REG), 5486
REWEIGHT statement (REG), 5504
ALLOC= option
STRATA statement (SURVEYSELECT), 6641
ALLSTATS option
OUTPUT statement (GLIMMIX), 2179
OUTPUT statement (HPMIXED), 2826
ALOGIT function
RESPONSE statement (CATMOD), 1131
ALPHA= option
BASELINE statement (PHREG), 4544
COMPUTE statement (VARIOGRAM), 7534
CONTRAST statement (CATMOD), 1111
CONTRAST statement (LOGISTIC), 3283
CONTRAST statement (PHREG), 4561
CONTRAST statement (SURVEYLOGISTIC), 6393
DESIGN statement (SEQDESIGN), 5804
ESTIMATE statement (GENMOD), 1943
ESTIMATE statement (GLIMMIX), 2136
ESTIMATE statement (HPMIXED), 2819
ESTIMATE statement (MIXED), 3921
ESTIMATE statement (NLMIXED), 4375
EXACT statement (FREQ), 1693
EXACT statement (LOGISTIC), 3285
EXACT statement (NPAR1WAY), 4464
HAZARDRATIO statement (PHREG), 4562
LOGISTIC statement (POWER), 4828
LSMEANS statement (GENMOD), 1945
LSMEANS statement (GLIMMIX), 2145
LSMEANS statement (GLM), 2460
LSMEANS statement (HPMIXED), 2821
LSMEANS statement (MIXED), 3925
LSMESTIMATE statement (GLIMMIX), 2157
MEANS statement (ANOVA), 711
MEANS statement (GLM), 2469
MODEL statement (CATMOD), 1120
MODEL statement (GAM), 1844
MODEL statement (GENMOD), 1947
MODEL statement (GLM), 2475
MODEL statement (HPMIXED), 2822
MODEL statement (LIFEREG), 3022
MODEL statement (LOESS), 3208
MODEL statement (LOGISTIC), 3292
MODEL statement (PHREG), 4566
MODEL statement (REG), 5472
MODEL statement (ROBUSTREG), 5669
MODEL statement (SURVEYLOGISTIC), 6398
MODEL statement (TPSPLINE), 7070
MODEL statement (TRANSREG), 7156, 7160
MODEL statement(VARCOMP), 7500
MULTREG statement (POWER), 4836
ONECORR statement (POWER), 4841
ONESAMPLEFREQ statement (POWER), 4845
ONESAMPLEMEANS statement (POWER), 4854
ONEWAYANOVA statement (POWER), 4860
OUTPUT statement (GLIMMIX), 2180
OUTPUT statement (GLM), 2479
OUTPUT statement (HPMIXED), 2826
OUTPUT statement (LOGISTIC), 3306
OUTPUT statement (NLIN), 4289
OUTPUT statement (SURVEYLOGISTIC), 6404
OUTPUT statement (SURVEYREG), 6557
PAIREDFREQ statement (POWER), 4865
PAIREDMEANS statement (POWER), 4872
POWER statement (GLMPOWER), 2658
PREDICT statement (NLMIXED), 4377
PROC FACTOR statement, 1550
PROC GLM statement, 2446
PROC LIFETEST statement, 3113
PROC LOGISTIC statement, 3268
PROC MI statement, 3752
PROC MIANALYZE statement, 3845
PROC MIXED statement, 3905
PROC NLIN statement, 4279
PROC NLMIXED statement, 4358
PROC NPAR1WAY statement, 4457
PROC PHREG statement, 4536
PROC QUANTREG (QUANTREG), 5373
PROC REG statement, 5454
PROC SURVEYLOGISTIC statement, 6383
PROC SURVEYMEANS statement, 6474
PROC SURVEYREG statement, 6543
PROC TTEST statement, 7401
RANDOM statement (GLIMMIX), 2187
RANDOM statement (HPMIXED), 2830
RANDOM statement (MIXED), 3950
RANDOM statement (NLMIXED), 4379

SCORE statement (LOGISTIC), 3312
TABLES statement (FREQ), 1701
TABLES statement (SURVEYFREQ), 6315
TWOSAMPLEFREQ statement (POWER), 4885
TWOSAMPLEMEANS statement (POWER), 4892
TWOSAMPLESURVIVAL statement (POWER), 4902
TWOSAMPLEWILCOXON statement (POWER), 4912
ALPHAECV= option
FITINDEX statement, 6760
PROC CALIS statement, 851
PROC TCALIS statement, 6726
ALPHAINIT= option
REPEATED statement (GENMOD), 1959
ALPHAP= option
MODEL statement (MIXED), 3929
ALPHAQT= option
PROC LIFETEST statement, 3113
ALPHARMS= option
PROC CALIS statement, 852
PROC TCALIS statement, 6726
ALPHARMSEA= option
FITINDEX statement, 6760
ALT= option
DESIGN statement (SEQDESIGN), 5804
ALTERNATE= option
PROC MDS statement, 3712
ALTREF= option
PROC SEQDESIGN statement, 5800
AM option
PROC MODECLUS statement, 4103
ANGLE= option
MODEL statement (KRIGE2D), 2945
SIMULATE statement (SIM2D), 6172
ANGLETOLERANCE= option
COMPUTE statement (VARIOGRAM), 7534
ANNOTATE= option
PLOT statement (BOXPLOT), 771
PLOT statement (REG), 5493
PROC BOXPLOT statement, 759
PROC LIFETEST statement, 3113
PROC REG statement, 5454
ANODEV= option
MODEL statement (GAM), 1844
ANOVA
DETAILS=STEPS option (GLMSELECT), 2708
ANOVA option
MODEL statement (SURVEYREG), 6555
OUTPUT statement (NPAR1WAY), 4465
PROC CANDISC statement, 1070
PROC DISCRIM statement, 1391
PROC NPAR1WAY statement, 4457
ANOVA procedure
syntax, 701
ANOVA procedure, ABSORB statement, 703
ANOVA procedure, BY statement, 704
ANOVA procedure, CLASS statement, 705
TRUNCATE option, 705
ANOVA procedure, FREQ statement, 705
ANOVA procedure, MANOVA statement, 706
ALL effect, 706
CANONICAL option, 707
E= option, 706
H= option, 706
INTERCEPT effect, 706
M= option, 706
MNAMES= option, 707
MSTAT= option, 707
ORTH option, 708
PREFIX= option, 707
PRINTE option, 708
PRINTH option, 708
SUMMARY option, 708
ANOVA procedure, MEANS statement, 709
ALPHA= option, 711
BON option, 711
CLDIFF option, 711
CLM option, 711
DUNCAN option, 711
DUNNETT option, 711
DUNNETTL option, 712
DUNNETTU option, 712
E= option, 712
GABRIEL option, 712
HOVTEST option, 712
KRATIO= option, 713
LINES option, 713
LSD option, 713, 714
NOSORT option, 713
REGWQ option, 713
SCHEFFE option, 714
SIDAK option, 714
SMM option, 714
SNK option, 714
TUKEY option, 714
WALLER option, 714
WELCH option, 714
ANOVA procedure, MODEL statement, 715
INTERCEPT option, 715
NOUNI option, 715
ANOVA procedure, PROC ANOVA statement, 702
DATA= option, 702

MANOVA option, 702
MULTIPASS option, 702
NAMELEN= option, 702
NOPRINT option, 702
ORDER= option, 702
OUTSTAT= option, 703
PLOTS= option, 703
ANOVA procedure, REPEATED statement, 715
CANONICAL option, 718
CONTRAST keyword, 717
E= effects, 720
factor specification, 716
H= effects, 720
HELMERT keyword, 717
IDENTITY keyword, 717
MEAN keyword, 717
MSTAT= option, 718
NOM option, 718
NOU option, 718
POLYNOMIAL keyword, 717
PRINTE option, 718
PRINTH option, 718
PRINTM option, 718
PRINTRV option, 719
PROFILE keyword, 717
SUMMARY option, 719
ANOVA procedure, TEST statement, 719
ANOVAF option
PROC MIXED statement, 3905
ANTIALIAS= option
ODS GRAPHICS statement, 529
ANTIALIASMAX= option
ODS GRAPHICS statement, 529
AOV option
PROC NESTED statement, 4257
APPROXIMATIONS option
OUTPUT statement (TRANSREG), 7170
PROC PRINQUAL statement, 5204
APREFIX= option
PROC PRINQUAL statement, 5204
ARRAY statement
MCMC procedure, 3508
NLMIXED procedure, 4373
ARSIN transformation
MODEL statement (TRANSREG), 7143
TRANSFORM statement (PRINQUAL), 5214
ASE
STATS= option (GLMSELECT), 2713
ASINGULAR= option
NLOPTIONS statement (CALIS), 890
NLOPTIONS statement (GLIMMIX), 394
NLOPTIONS statement (HPMIXED), 394
NLOPTIONS statement (TCALIS), 394
PROC CALIS statement, 852
PROC NLMIXED statement, 4358
PROC TCALIS statement, 6726
ASSESS statement
GENMOD procedure, 1926
PHREG procedure, 4540
ASYCORR option
PROC GLIMMIX statement, 2095
PROC MIXED statement, 3905
ASYCOV option
PROC GLIMMIX statement, 2095
PROC MIXED statement, 3905, 4032
ASYCOV= option
PROC CALIS statement, 852
PROC TCALIS statement, 6726
AT MEANS option
LSMEANS statement (GLIMMIX), 2145
LSMEANS statement (MIXED), 3925
LSMESTIMATE statement (GLIMMIX), 2157
AT option
LSMEANS statement (GLIMMIX), 2145, 2146
LSMEANS statement (GLM), 2460, 2529
LSMEANS statement (MIXED), 3925, 3926
LSMESTIMATE statement (GLIMMIX), 2157
ODDSRATIO statement (LOGISTIC), 3304
AT= option
HAZARDRATIO statement (PHREG), 4562
AUGMENT option
PROC CALIS statement, 852
AUTOCORLAG= option
PROC MCMC statement, 3497
AUTOCORRELATION option
VARIOGRAM procedure, COMPUTE statement, 7534
AUTOCORRELATION STATISTICS= option
VARIOGRAM procedure, COMPUTE statement, 7534
AVERAGE option
PROC INBREED statement, 2864
TEST statement (PHREG), 4578
AVERAGED option
MODEL statement (CATMOD), 1120

B option
MODEL statement (REG), 5472
PROC CANCORR statement, 1039
BANDMAX= option, *see* BANDMAXTIME= option
BANDMAXTIME= option
PROC LIFETEST statement, 3113

BANDMIN= option, *see* BANDMINTIME= option
BANDMINTIME= option
PROC LIFETEST statement, 3113
BANDWIDTH= option
COMPUTE statement (VARIOGRAM), 7536
BASELINE statement
PHREG procedure, 4541
BASEMODEL statement
TCALIS procedure, 6745
BASEMODEL statement, TCALIS procedure, 6745
BASIS option
EFFECT statement, spline (GLIMMIX), 384
EFFECT statement, spline (GLMSELECT), 384
BAYES statement
GENMOD procedure, 1928
LIFEREG procedure, 3007
PHREG procedure, 4545
BCORR option
PROC CANDISC statement, 1070
PROC DISCRIM statement, 1391
PROC STEPDISC statement, 6275
BCOV option
PROC CANDISC statement, 1070
PROC DISCRIM statement, 1391
PROC MIANALYZE statement, 3845
PROC STEPDISC statement, 6275
TEST statement (MIANALYZE), 3850
BDATA= option
PRIOR statement (MIXED), 3947
BDCHI option
OUTPUT statement (FREQ), 1695
BDT option
TABLES statement (FREQ), 1701
BEGINCNST statement
MCMC procedure, 3509
BEGINNODATA statement
MCMC procedure, 3511
BEGINPRIOR statement
MCMC procedure, 3511
BENZECRI option
PROC CORRESP statement, 1325
BEST= option
MODEL statement (LOGISTIC), 3292
MODEL statement (PHREG), 4566
MODEL statement (REG), 5472
PARMS statement (NLMIXED), 4377
PROC NLIN statement, 4279
BETA= option
DESIGN statement (SEQDESIGN), 5804
PROC CLUSTER statement, 1241
BETAOVERLAP= option
DESIGN statement, 5804
PROC SEQTEST statement, 6004
BIASKUR option
PROC CALIS statement, 853
PROC TCALIS statement, 6727
BIASTEST option
PROC ROBUSTREG statement, 5666
BIC
STATS= option (GLMSELECT), 2713
BIC option
MODEL statement (REG), 5473
PLOT statement (REG), 5493
BINARY option
PROC CORRESP statement, 1325
BINOMIAL option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1695
TABLES statement (FREQ), 1701
TEST statement (MULTTEST), 4201
BINOMIALC option
TABLES statement (FREQ), 1705
BINS= option
PROC FASTCLUS statement, 1634
BINWIDTH= option
MODEL statement (LOGISTIC), 3292
BIVAR statement
KDE procedure, 2890
BIVSTATS option
BIVAR statement, 2891
BLOCKLABELPOS= option
PLOT statement (BOXPLOT), 772
BLOCKLABTYPE= option
PLOT statement (BOXPLOT), 772
BLOCKPOS= option
PLOT statement (BOXPLOT), 772
BLOCKREP option
PLOT statement (BOXPLOT), 772
BLOCKVAR= option
PLOT statement (BOXPLOT), 772
BODY= option
ODS HTML statement, 532
BON option
MEANS statement (ANOVA), 711
MEANS statement (GLM), 2470
BONFERRONI option
PROC MULTTEST statement, 4189, 4212
BOOTSTRAP option
MCMC statement (MI), 3760
PROC MULTTEST statement, 4184, 4189, 4213, 4229
BORDER= option
ODS GRAPHICS statement, 530

BOUNDARY option
PROC MODECLUS statement, 4103
BOUNDARY= option
PROC SEQTEST statement, 6005
BOUNDARYADJ= option
PROC SEQTEST statement, 6005
BOUNDARYKEY= option
DESIGN statement, 5805
PROC SEQTEST statement, 6006
BOUNDARYSCALE= option
PROC SEQDESIGN statement, 5800
PROC SEQTEST statement, 6006
BOUNDS statement
CALIS procedure, 871
NLIN procedure, 4283
NLMIXED procedure, 4373
TCALIS procedure, 6746
BOX= option
PROC BOXPLOT statement, 759
BOXCONNECT= option
PLOT statement (BOXPLOT), 772
BOXCOX transformation
MODEL statement (TRANSREG), 7144
TRANSFORM statement (MI), 3767
BOXES= option
PLOT statement (BOXPLOT), 773
BOXFILL= option
PLOT statement (BOXPLOT), 773
BOXPLOT procedure
HISTORY= option, 754
syntax, 759
BOXPLOT procedure, BY statement, 760
BOXPLOT procedure, ID statement, 760
BOXPLOT procedure, INSET statement, 761
CFILL= option, 762
CFILLH= option, 762
CFRAME= option, 763
CHEADER= option, 763
CSHADOW= option, 763
CTEXT= option, 763
DATA option, 763
FONT= option, 763
FORMAT= option, 763
HEADER= option, 763
HEIGHT= option, 763
NOFRAME option, 763
POSITION= option, 763, 800–802
REFPOINT= option, 764
BOXPLOT procedure, INSETGROUP statement, 764
CFILL= option, 765
CFILLH= option, 765
CFRAME= option, 766
CHEADER= option, 766
CTEXT= option, 766
FONT= option, 766
FORMAT= option, 766
HEADER= option, 766
HEIGHT= option, 766
NOFRAME option, 766
POSITION= option, 766
BOXPLOT procedure, PLOT statement, 767
ALLLABEL= option, 771
ANNOTATE= option, 771
BLOCKLABELPOS= option, 772
BLOCKLABTYPE= option, 772
BLOCKPOS= option, 772
BLOCKREP option, 772
BLOCKVAR= option, 772
BOX= data set, 793
BOXCONNECT= option, 772
BOXES= option, 773
BOXFILL= option, 773
BOXSTYLE= option, 773, 817
BOXWIDTH= option, 774
BOXWIDTHSCALE= option, 774, 822
BWSLEGEND option, 774
CAXIS= option, 775
CBLOCKLAB= option, 775
CBLOCKVAR= option, 775
CBOXES= option, 775
CBOXFILL= option, 775
CCLIP= option, 776
CCONNECT= option, 776
CCOVERLAY= option, 776
CFRAME= option, 776
CGRID= option, 776
CHREF= option, 776
CLABEL= option, 776
CLIPFACTOR= option, 776, 808
CLIPLEGEND= option, 777
CLIPLEGPOS= option, 777
CLIPSUBCHAR= option, 777
CLIPSYMBOL= option, 777
CLIPSYMBOLHT= option, 777
CONTINUOUS option, 777
COVERLAY= option, 777
COVERLAYCLIP= option, 778
CTEXT= option, 778
CVREF= option, 778
DATA= data set, 793
DESCRIPTION= option, 778
ENDGRID option, 778
FONT= option, 778
FRONTREF option, 778
GRID= option, 778
HAXIS= option, 778
HEIGHT= option, 779

HISTORY= data set, 794, 795
HMINOR= option, 779
HOFFSET= option, 779
HORIZONTAL option, 779
HREF= option, 780
HREFLABELS= option, 780
HREFLABPOS= option, 780
HTML= option, 781
IDCOLOR= option, 781
IDCTEXT= option, 781
IDFONT= option, 781
IDHEIGHT= option, 781
IDSYMBOL= option, 781
INTERVAL= option, 781
LABELANGLE= option, 782
LBOXES= option, 782
LENDGRID= option, 783
LGRID= option, 783
LHREF= option, 783
LOVERLAY= option, 783
LVREF= option, 783
MAXPANELS= option, 783
MISSBREAK option, 783
NAME= option, 783
NLEGEND option, 784
NOBYREF option, 784
NOCHART option, 784
NOFRAME option, 784
NOHLABEL option, 784
NOOVERLAYLEGEND option, 784
NOSERIFS option, 784
NOTCHES option, 784, 821
NOTICKREP option, 785
NOVANGLE option, 785
NPANELPOS= option, 785
ODS graphics, 824
OUTBOX= data set, 791
OUTBOX= option, 756, 785
OUTHISTORY= data set, 792
OUTHISTORY= option, 786
OVERLAY= option, 786
OVERLAYCLIPSYM= option, 786
OVERLAYCLIPSYMHT= option, 786
OVERLAYHTML= option, 786
OVERLAYID= option, 786
OVERLAYLEGLAB= option, 786
OVERLAYSYM= option, 786
OVERLAYSYMHT= option, 787
PAGENUM= option, 787
PAGENUMPOS= option, 787
PCTLDEF= option, 787
REPEAT option, 787
SKIPHLABELS= option, 788
SYMBOLLEGEND= option, 788
SYMBOLORDER= option, 788
TOTPANELS= option, 788
TURNHLABELS option, 788
VAXIS= option, 788
VFORMAT= option, 789
VMINOR= option, 789
VOFFSET= option, 789
VREF= option, 789
VREFLABELS= option, 789
VREFLABPOS= option, 789
VZERO option, 790
WAXIS= option, 790
WGRID= option, 790
WOVERLAY= option, 790
BOXPLOT procedure, plot statement
OUTHIGHHTML= option, 786
OUTLOWHTML= option, 786
BOXPLOT procedure, plot statements
INTSTART= option, 782
BOXPLOT procedure, PROC BOXPLOT statement, 759
ANNOTATE= option, 759
BOX= option, 759
DATA= option, 759
GOUT= option, 759
BOXSTYLE= option
PLOT statement (BOXPLOT), 773
BOXWIDTH= option
PLOT statement (BOXPLOT), 774
BOXWIDTHSCALE= option
PLOT statement (BOXPLOT), 774
BSCALE= option
PROC SEQDESIGN statement, 5800
PROC SEQTEST statement, 6006
BSPLINE transformation
MODEL statement (TRANSREG), 7142
BSSCP option
PROC CANDISC statement, 1070
PROC DISCRIM statement, 1391
PROC STEPDISC statement, 6275
BUCKET= option
MODEL statement (LOESS), 3209
BUCKET= suboption
RANDOM statement (GLIMMIX), 2189
BUILDSSCP= option
PERFORMANCE statement (GLMSELECT), 2717
BWM= option
BIVAR statement, 2891
UNIVAR statement, 2893
BWSLEGEND option
PLOT statement (BOXPLOT), 774
BY statement
ACECLUS procedure, 678

ANOVA procedure, 704
BOXPLOT procedure, 760
CALIS procedure, 872
CANCORR procedure, 1043
CANDISC procedure, 1072
CATMOD procedure, 1109
CORRESP procedure, 1331
DISCRIM procedure, 1397
FACTOR procedure, 1564
FREQ procedure, 1690
GAM procedure, 1840
GENMOD procedure, 1937
GLIMMIX procedure, 2121
GLM procedure, 2452
GLMMOD procedure, 2628
GLMPOWER procedure, 2651
GLMSELECT procedure, 2702
HPMIXED procedure, 2814
INBREED procedure, 2865
KDE procedure, 2896
KRIGE2D procedure, 2940
LATTICE procedure, 2983
LIFEREG procedure, 3017
LIFETEST procedure, 3124
LOESS procedure, 3207
LOGISTIC procedure, 3279
MCMC procedure, 3511
MDS procedure, 3721
MI procedure, 3754
MIANALYZE procedure, 3847
MIXED procedure, 3916
MODECLUS procedure, 4108
MULTTEST procedure, 4197
NESTED procedure, 4257
NLIN procedure, 4284
NLMIXED procedure, 4374
NPAR1WAY procedure, 4462
ORTHOREG procedure, 4511
PHREG procedure, 4556
PLS procedure, 4781
PRINCOMP procedure, 5156
PRINQUAL procedure, 5211
QUANTREG procedure, 5377
REG procedure, 5467
ROBUSTREG procedure, 5667
RSREG procedure, 5730
SIM2D procedure, 6169
SIMNORMAL procedure, 6204
STEPDISC procedure, 6278
SURVEYFREQ procedure, 6309
SURVEYLOGISTIC procedure, 6388
SURVEYMEANS procedure, 6483
SURVEYREG procedure, 6549
TCALIS procedure, 6746
TPSPLINE procedure, 7068
TRANSREG procedure, 7137
TREE procedure, 7367
TTEST procedure, 7408
VARCLUS procedure, 7474
VARCOMP procedure, 7499
VARIOGRAM procedure, 7533
BYCAT option
CONTRAST statement (GLIMMIX), 2125
ESTIMATE statement (GLIMMIX), 2136
BYCATEGORY option
CONTRAST statement (GLIMMIX), 2125
ESTIMATE statement (GLIMMIX), 2136
BYDATA option
PARMS statement (NLMIXED), 4377
BYLEVEL option
LSMEANS statement (GLIMMIX), 2146
LSMEANS statement (GLM), 2460, 2530
LSMEANS statement (MIXED), 3926, 3927
LSMESTIMATE statement (GLIMMIX), 2157
BYOUT option
MODEL statement (RSREG), 5731
BYVAR option
PROC TTEST statement, 7401

C option
PROC CANCORR statement, 1039
C= option
OUTPUT statement (LOGISTIC), 3306
TRANSFORM statement (MI), 3768
CA option
TEST statement (MULTTEST), 4201, 4203, 4225
CALIS procedure, 846
syntax, 846
CALIS procedure, BOUNDS statement, 871
CALIS procedure, BY statement, 872
CALIS procedure, COSAN statement, 873
CALIS procedure, COV statement, 874
CALIS procedure, FACTOR statement, 877
COMPONENT option, 877
HEYWOOD option, 878
N= option, 878
NORM option, 878
RCONVERGE= option, 878
RITER= option, 878
ROTATE= option, 879
CALIS procedure, FREQ statement, 880
CALIS procedure, LINCON statement, 880
CALIS procedure, LINEQS statement, 881
CALIS procedure, MATRIX statement, 883
CALIS procedure, NLINCON statement, 886
CALIS procedure, NLOPTIONS statement, 886

ABSCONV= option, 890
ABSFCONV= option, 892
ABSFTOL= option, 892
ABSGCONV= option, 892, 895
ABSGTOL= option, 892, 895
ABSXCONV= option, 892
ABSXTOL= option, 892
ALL option, 896
ASINGULAR= option, 890
CFACTOR= option, 895
COVSING= option, 890
DAMPSTEP= option, 896
FCONV2= option, 892, 895
FCONV= option, 889, 892
FDIGITS= option, 893
FSIZE= option, 893
FTOL2= option, 892, 895
FTOL= option, 889, 892
G4= option, 889
GCONV2= option, 893
GCONV= option, 889, 893, 895
GTOL2= option, 893
GTOL= option, 889, 893, 895
HESCAL= option, 896
INSTEP= option, 889
LCDEACT= option, 897
LCEPSILON= option, 897
LCSINGULAR= option, 897
LINESEARCH= option, 889
LSPRECISION= option, 889
MAXFUNC= option, 889, 891
MAXITER= option, 889, 891
MAXTIME= option, 891
MINITER= option, 891
MSINGULAR= option, 890
NOEIGNUM option, 898
NOHLF option, 895
OMETHOD= option, 889
PALL option, 896
PCRPJAC option, 896
PHESSIAN option, 896
PHISTORY option, 896
PINIT option, 896
PJTJ option, 896
PNLCJAC option, 896
PRINT option, 896
RADIUS= option, 889
RESTART= option, 898
SALPHA= option, 889
SINGULAR= option, 890
TECHNIQUE= option, 889
UPDATE= option, 889
VERSION= option, 898
VSINGULAR= option, 890
XCONV= option, 894
XSIZE= option, 894
XTOL= option, 894
CALIS procedure, PARAMETERS statement, 898
CALIS procedure, PARTIAL statement, 899
CALIS procedure, PROC CALIS statement, 848
ALL option, 863
ALPHAECV= option, 851
ALPHARMS= option, 852
ASINGULAR= option, 852
ASYCOV= option, 852
AUGMENT option, 852
BIASKUR option, 853
CORR option, 863
COVARIANCE option, 853
COVSING= option, 853
DATA= option, 853
DEMPHAS= option, 853
DFE= option, 854
DFR= option, 866
DFREDUCE= option, 854
EDF= option, 854
ESTDATA= option, 856
FCONV= option, 854
FDCODE option, 854
FTOL= option, 854
G4= option, 854
GCONV= option, 855
GTOL= option, 855
HESSALG= option, 855
INEST= option, 856
INRAM= option, 856
INSTEP= option, 856
INVAR= option, 856
INWGT= option, 856
KURTOSIS option, 857
LINESEARCH= option, 857
LSPRECISION= option, 857
MAXFUNC= option, 858
MAXITER= option, 858
METHOD= option, 859
MODIFICATION option, 860
MSINGULAR= option, 860
NOADJDF option, 860
NOBS= option, 860
NODIAG option, 860
NOINT option, 861
NOMOD option, 861
NOPRINT option, 861
NOSTDERR option, 861
OM= option, 867
OMETHOD= option, 867
OUTEST= option, 861

OUTJAC option, 862
OUTRAM= option, 862
OUTSTAT= option, 862
OUTVAR= option, 861
OUTWGT= option, 862
PALL option, 863
PCORR option, 863
PCOVES option, 863
PDETERM option, 863
PESTIM option, 864
PINITIAL option, 864
PJACPAT option, 864
PLATCOV option, 864
PLOTS= option, 864
PREDET option, 865
PRIMAT option, 865
PRINT option, 865
PRIVEC option, 865
PSHORT option, 865
PSUMMARY option, 865
PWEIGHT option, 865
RADIUS= option, 865
RANDOM= option, 865
RDF= option, 866
RESIDUAL= option, 866
RIDGE= option, 866
SALPHA= option, 866
SHORT option, 865
SIMPLE option, 866
SINGULAR= option, 867
SLMW= option, 867
SMETHOD= option, 857
SPRECISION= option, 857, 867
START= option, 867
STDERR option, 867
SUMMARY option, 865
TECHNIQUE= option, 867
TOTEFF option, 869
UCORR option, 869
UCOV option, 869
UPDATE= option, 869
VARDEF= option, 870
VSINGULAR= option, 870
WPENALTY= option, 870
WRIDGE= option, 871
CALIS procedure, Programming statements
ABORT statement, 904
CALL statement, 904
DELETE statement, 904
DO statement, 904
GOTO statement, 904
IF statement, 904
LINK statement, 904
PUT statement, 904
RETURN statement, 904
SELECT statement, 904
STOP statement, 904
SUBSTR statement, 904
WHEN statement, 904
CALIS procedure, RAM statement, 899
CALIS procedure, STD statement, 906
CALIS procedure, STRUCTEQ statement, 907
CALIS procedure, VAR statement, 907
CALIS procedure, VARNAMES statement, 908
CALIS procedure, WEIGHT statement, 909
CAN option
PROC DISCRIM statement, 1391
CANCORR procedure
syntax, 1037
CANCORR procedure, BY statement, 1043
CANCORR procedure, FREQ statement, 1043
CANCORR procedure, PARTIAL statement, 1043
CANCORR procedure, PROC CANCORR statement, 1037
ALL option, 1039
B option, 1039
C option, 1039
CLB option, 1039
CORR option, 1039
CORRB option, 1039
DATA= option, 1039
EDF= option, 1039
INT option, 1039
MSTAT= option, 1039
NCAN= option, 1040
NOINT option, 1040
NOPRINT option, 1040
OUT= option, 1040
OUTSTAT= option, 1040
PARPREFIX= option, 1041
PCORR option, 1040
PPREFIX= option, 1041
PROBT option, 1040
RDF= option, 1041
RED option, 1041
REDUNDANCY option, 1041
S option, 1041
SEB option, 1041
SHORT option, 1041
SIMPLE option, 1041
SING= option, 1041
SINGULAR= option, 1041
SMC option, 1041
SPCORR option, 1041
SQPCORR option, 1041
SQSPCORR option, 1041
STB option, 1042

T option, 1042
VDEP option, 1042
VN= option, 1042
VNAME= option, 1042
VP= option, 1042
VPREFIX= option, 1042
VREG option, 1042
WDEP option, 1042
WN= option, 1042
WNAME= option, 1042
WP= option, 1042
WPREFIX= option, 1042
WREG option, 1042
CANCORR procedure, VAR statement, 1044
CANCORR procedure, WEIGHT statement, 1044
CANCORR procedure, WITH statement, 1044
CANDISC procedure
syntax, 1068
CANDISC procedure, BY statement, 1072
CANDISC procedure, CLASS statement, 1072
CANDISC procedure, FREQ statement, 1073
CANDISC procedure, PROC CANDISC statement, 1069
ALL option, 1070
ANOVA option, 1070
BCORR option, 1070
BCOV option, 1070
BSSCP option, 1070
DATA= option, 1070
DISTANCE option, 1070
MAHALANOBIS option, 1070
NCAN= option, 1070
NOPRINT option, 1070
OUT= option, 1070
OUTSTAT= option, 1071
PCORR option, 1071
PCOV option, 1071
PREFIX= option, 1071
PSSCP option, 1071
SHORT option, 1071
SIMPLE option, 1071
SINGULAR= option, 1071
STDMEAN option, 1071
TCORR option, 1071
TCOV option, 1072
TSSCP option, 1072
WCORR option, 1072
WCOV option, 1072
WSSCP option, 1072
CANDISC procedure, VAR statement, 1073
CANDISC procedure, WEIGHT statement, 1073
CANONICAL option
MANOVA statement (ANOVA), 707
MANOVA statement (GLM), 2465
OUTPUT statement (TRANSREG), 7171
PROC DISCRIM statement, 1391
REPEATED statement (ANOVA), 718
REPEATED statement (GLM), 2484
CANPREFIX= option
PROC DISCRIM statement, 1391
CANPRINT option
MTEST statement (REG), 5481
CASCADE= option
PROC MODECLUS statement, 4103
CATMOD, 1092
CATMOD procedure
syntax, 1107
CATMOD procedure, BY statement, 1109
CATMOD procedure, CONTRAST statement, 1110
ALPHA= option, 1111
ESTIMATE= option, 1111
CATMOD procedure, DIRECT statement, 1113
CATMOD procedure, FACTORS statement, 1114
PROFILE= option, 1115
RESPONSE= option, 1115
TITLE= option, 1116
CATMOD procedure, LOGLIN statement, 1117
TITLE= option, 1117
CATMOD procedure, MODEL statement, 1118
ADDCELL= option, 1120
ALPHA= option, 1120
AVERAGED option, 1120
CLPARM option, 1120
CORRB option, 1120
COV option, 1120
COVB option, 1120
DESIGN option, 1120
EPSILON= option, 1121
FREQ option, 1121
GLS option, 1124
ITPRINT option, 1121
MAXITER= option, 1121
MISSING= option, 1123
ML option, 1121
NODESIGN option, 1123
NOINT option, 1123
NOITER option, 1123
NOPARM option, 1123
NOPREDVAR option, 1123
NOPRINT option, 1123
NOPROFILE option, 1123
NORESPONSE option, 1124
ONEWAY option, 1124
PARAM= option, 1124
PRED= option, 1124
PREDICT option, 1124

PROB option, 1124
PROFILE option, 1124
RESPONSE keyword, 1114, 1117, 1118, 1120, 1129, 1142, 1148, 1150, 1158, 1160, 1165, 1174
TITLE= option, 1124
WLS option, 1124
XPX option, 1124
ZERO= option, 1125
CATMOD procedure, POPULATION statement, 1126
CATMOD procedure, PROC CATMOD statement, 1108
DATA=option, 1108
NAMELEN= option, 1108
NOPRINT option, 1108
ORDER= option, 1109
CATMOD procedure, REPEATED statement, 1129
PROFILE= option, 1129
RESPONSE= option, 1130
TITLE= option, 1130
CATMOD procedure, RESPONSE statement, 1130
ALOGIT function, 1131
CLOGIT function, 1131
JOINT function, 1131
LOGIT function, 1131
MARGINAL function, 1131
MEAN function, 1131
OUT= option, 1132
OUTEST= option, 1132
READ function, 1132
TITLE= option, 1132
CATMOD procedure, RESTRICT statement, 1137
CATMOD procedure, WEIGHT statement, 1138
CAXIS= option
PLOT statement (BOXPLOT), 775
PLOT statement (REG), 5493
CBAR= option
OUTPUT statement (LOGISTIC), 3306
CBLOCKLAB= option
PLOT statement (BOXPLOT), 775
CBLOCKVAR= option
PLOT statement (BOXPLOT), 775
CBOXES= option
PLOT statement (BOXPLOT), 775
CBOXFILL= option
PLOT statement (BOXPLOT), 775
CCC option
OUTPUT statement (TRANSREG), 7171
PROC CLUSTER statement, 1241
CCLIP= option
PLOT statement (BOXPLOT), 776
CCONF= option
MCMC statement (MI), 3757
CCONNECT= option
MCMC statement (MI), 3763
PLOT statement (BOXPLOT), 776
CCONVERGE= option
MODEL statement (TRANSREG), 7160
PROC PRINQUAL statement, 5204
CCOVERLAY= option
PLOT statement (BOXPLOT), 776
CDF keyword
OUTPUT statement (LIFEREG), 3026
CDFPLOT statement, *see* PROBIT procedure, CDFPLOT statement, *see* PROBIT procedure, CDFPLOT statement
options summarized by function, 5269
PROBIT procedure, 5268
CDPREFIX= option
OUTPUT statement (TRANSREG), 7171
CEC option
OUTPUT statement (TRANSREG), 7171
CELLCHI2 option
PROC CORRESP statement, 1325
TABLES statement (FREQ), 1705
CENSCALE option
PROC PLS statement, 4775
CENSORED keyword
OUTPUT statement (LIFEREG), 3026
CENSOREDSYMBOL= option
PROC LIFETEST statement, 3113
CENTER option
MODEL statement (TRANSREG), 7157
PROC MULTTEST statement, 4189
CENTER= option
RIDGE statement (RSREG), 5734
CENTROID option
PROC VARCLUS statement, 7470
CERTSIZE= option
PROC SURVEYSELECT statement, 6625
CERTSIZE=P= option
PROC SURVEYSELECT statement, 6626
CFACTOR= option
NLOPTIONS statement (CALIS), 895
PROC NLMIXED statement, 4358
CFRAME= option
MCMC statement (MI), 3757, 3763
PLOT statement (BOXPLOT), 776
PLOT statement (REG), 5493
PROC TREE statement, 7362
CGRID= option
BOXPLOT procedure, 776
CHAIN= option
MCMC statement (MI), 3759

CHANGE= option
PROC PRINQUAL statement, 5204
CHECKDEPENDENCY= option
STRATA statement (LOGISTIC), 3315
CHICORR option
PROC TCALIS statement, 6727
CHICORRECT option
PROC TCALIS statement, 6727
CHICORRECT= option
FITINDEX statement, 6760
CHIF option
PROC ROBUSTREG statement, 5665, 5666
CHISQ option
CONTRAST statement (GLIMMIX), 2126
CONTRAST statement (HPMIXED), 2817
CONTRAST statement (MIXED), 3919
EXACT statement (FREQ), 1692, 1812
LSMESTIMATE statement (GLIMMIX), 2157
MODEL statement (GLIMMIX), 2164
MODEL statement (MIXED), 3929
OUTPUT statement (FREQ), 1695
TABLES statement (FREQ), 1705, 1732, 1812
TABLES statement (SURVEYFREQ), 6315
TEST statement (HPMIXED), 2831
CHISQ1 option
TABLES statement (SURVEYFREQ), 6315
CHOCKING= option
PLOT statement (REG), 5493
CHOL option
PROC GLIMMIX statement, 2096
CHOLESKY option
PROC GLIMMIX statement, 2096
CHOOSE= option
MODEL statement (GLMSELECT), 2710
CHREF= option
PLOT statement (BOXPLOT), 776
PLOT statement (REG), 5493
CI option
PROC QUANTREG statement, 5373
CI= option
ONESAMPLEFREQ statement (POWER), 4846
ONESAMPLEMEANS statement (POWER), 4854
PAIREDMEANS statement (POWER), 4873
PROC TTEST statement, 7402
TWOSAMPLEMEANS statement (POWER), 4893
CIALPHA= option
PROC SEQTEST statement, 6007
CICONV= option
MODEL statement (GENMOD), 1947
CILPREFIX= option
OUTPUT statement (TRANSREG), 7171
CIPREFIX= option
OUTPUT statement (TRANSREG), 7171
CITYPE= option
PROC SEQTEST statement, 6007
CIUPREFIX= option
OUTPUT statement (TRANSREG), 7171
CK= option
PROC MODECLUS statement, 4104
CL option
COMPUTE statement (VARIOGRAM), 7536
COVTEST statement (GLIMMIX), 2130
ESTIMATE statement (GLIMMIX), 2136
ESTIMATE statement (HPMIXED), 2819
ESTIMATE statement (MIXED), 3921
LSMEANS statement (GENMOD), 1945
LSMEANS statement (GLIMMIX), 2146
LSMEANS statement (GLM), 2460
LSMEANS statement (HPMIXED), 2821
LSMEANS statement (MIXED), 3926
LSMESTIMATE statement (GLIMMIX), 2157
MODEL statement (GENMOD), 1947
MODEL statement (GLIMMIX), 2164
MODEL statement (HPMIXED), 2822
MODEL statement (LOGISTIC), 3302
MODEL statement (MIXED), 3929
MODEL statement (TRANSREG), 7160
MODEL statement(VARCOMP), 7500
RANDOM statement (GLIMMIX), 2187
RANDOM statement (HPMIXED), 2830
RANDOM statement (MIXED), 3950
TABLES statement (FREQ), 1705
TABLES statement (SURVEYFREQ), 6316
CL= option
HAZARDRATIO statement (PHREG), 4563
ODDSRATIO statement (LOGISTIC), 3304
PROC MIXED statement, 3906
CLABEL= option
BOXPLOT procedure, 776
CLASS statement
ANOVA procedure, 705
CANDISC procedure, 1072
DISCRIM procedure, 1398
GAM procedure, 1840
GENMOD procedure, 1937
GLIMMIX procedure, 2122
GLM procedure, 2453
GLMMOD procedure, 2628
GLMPOWER procedure, 2652
GLMSELECT procedure, 2702

HPMIXED procedure, 2815, 2837
INBREED procedure, 2866
LIFEREG procedure, 3017
LOGISTIC procedure, 3280
MI procedure, 3754
MIANALYZE procedure, 3848
MIXED procedure, 3916, 3995
MULTTEST procedure, 4198
NESTED procedure, 4258
NPAR1WAY procedure, 4462
ORTHOREG procedure, 4511
PHREG procedure, 4557
PLS procedure, 4782
QUANTREG procedure, 5377
ROBUSTREG procedure, 5668
STEPDISC procedure, 6278
SURVEYLOGISTIC procedure, 6389
SURVEYMEANS procedure, 6484
SURVEYREG procedure, 6549
TTEST procedure, 7408
VARCOMP procedure, 7499
CLASS transformation
MODEL statement (TRANSREG), 7142
CLASSICAL option
COVTEST statement (GLIMMIX), 2132
CLASSVAR= option
PROC MIANALYZE statement, 3847
CLB option
MODEL statement (REG), 5473
PROC CANCORR statement, 1039
CLDIFF option
MEANS statement (ANOVA), 711
MEANS statement (GLM), 2470
CLEAR option
PLOT statement (REG), 5497
CLI option
MODEL statement (GLM), 2475
MODEL statement (REG), 5473
OUTPUT statement (TRANSREG), 7171
CLINE= option
PLOT statement (REG), 5493
CLIPFACTOR= option
BOXPLOT procedure, 776, 808
CLIPLEGEND= option
BOXPLOT procedure, 777
CLIPLEGPOS= option
BOXPLOT procedure, 777
CLIPSUBCHAR= option
BOXPLOT procedure, 777
CLIPSYMBOL= option
BOXPLOT procedure, 777
CLIPSYMBOLHT= option
BOXPLOT procedure, 777
CLL= option
MODEL statement (TRANSREG), 7156
CLM option
MEANS statement (ANOVA), 711
MEANS statement (GLM), 2470
MODEL statement (GLM), 2475
MODEL statement (LOESS), 3209
MODEL statement (REG), 5473
OUTPUT statement (TRANSREG), 7171
PROC GAM statement, 1840
SCORE statement (LOESS), 3214
SCORE statement (LOGISTIC), 3312
CLODDS option
MODEL statement (SURVEYLOGISTIC), 6398
CLODDS= option
MODEL statement (LOGISTIC), 3292
CLOGIT function
RESPONSE statement (CATMOD), 1131
CLOSEFIT option
PROC TCALIS statement, 6728
CLOSEFIT= option
FITINDEX statement, 6760
CLPARM option
MODEL statement (CATMOD), 1120
MODEL statement (GLM), 2475
MODEL statement (SURVEYLOGISTIC), 6399
MODEL statement (SURVEYREG), 6555
CLPARM= option
MODEL statement (LOGISTIC), 3293
CLTYPE= option
BASELINE statement (PHREG), 4544
EXACT statement (LOGISTIC), 3286
CLUSTER procedure
syntax, 1239
CLUSTER procedure, BY statement, 1247
CLUSTER procedure, COPY statement, 1247
CLUSTER procedure, FREQ statement, 1248
CLUSTER procedure, ID statement, 1248
CLUSTER procedure, PROC CLUSTER statement, 1239
BETA= option, 1241
CCC option, 1241
DATA= option, 1241
DIM= option, 1242
HYBRID option, 1242
K= option, 1242
MODE= option, 1243
NOEIGEN option, 1243
NOID option, 1243
NONORM option, 1243
NOPRINT option, 1243
NOSQUARE option, 1244
NOTIE option, 1244

OUTTREE= option, 1244
PENALTY= option, 1244
PLOTS option, 1244
PRINT= option, 1246
PSEUDO= option, 1246
R= option, 1246
RMSSTD option, 1246
RSQUARE option, 1246
SIMPLE option, 1246
STANDARD option, 1246
TRIM= option, 1246
CLUSTER procedure, RMSSTD statement, 1249
CLUSTER procedure, VAR statement, 1249
CLUSTER statement
SURVEYFREQ procedure, 6310
SURVEYLOGISTIC procedure, 6391
SURVEYMEANS procedure, 6485
SURVEYREG procedure, 6550
CLUSTER= option
PROC FASTCLUS statement, 1634
PROC MODECLUS statement, 4104
CLUSTERLABEL= option
PROC FASTCLUS statement, 1634
CLWT option
TABLES statement (SURVEYFREQ), 6316
CMALLOWS= option
PLOT statement (REG), 5494
CMH option
OUTPUT statement (FREQ), 1695
TABLES statement (FREQ), 1706
CMH1 option
OUTPUT statement (FREQ), 1695
TABLES statement (FREQ), 1706
CMH2 option
OUTPUT statement (FREQ), 1695
TABLES statement (FREQ), 1706
CMHCOR option
OUTPUT statement (FREQ), 1695
CMHGA option
OUTPUT statement (FREQ), 1695
CMHRMS option
OUTPUT statement (FREQ), 1695
CMLPREFIX= option
OUTPUT statement (TRANSREG), 7172
CMUPREFIX= option
OUTPUT statement (TRANSREG), 7172
CNEEDLES= option
MCMC statement (MI), 3758
COCHQ option
OUTPUT statement (FREQ), 1695
COCHRAN option
PROC TTEST statement, 7402
CODING= option
MODEL statement (GENMOD), 1947
COEF= option
PROC MDS statement, 3712
COEFFICIENTS option
OUTPUT statement (TRANSREG), 7172
COEFFPRIOR= option
BAYES statement(PHREG), 4546
COL option
TABLES statement (SURVEYFREQ), 6316
COLLECT option
PLOT statement (REG), 5497
COLLIN option
MODEL statement (REG), 5473
COLLINOINT option
MODEL statement (REG), 5473
COLUMN= option
PROC CORRESP statement, 1325
COMMONAXES option
PROC GAM statement, 1840
COMOR option
EXACT statement (FREQ), 1692
COMPONENT option
FACTOR statement (CALIS), 877
FACTOR statement (TCALIS), 6753
components option
PROC GAM statement, 1839
COMPRESS option
PROC FREQ statement, 1688
COMPUTE statement
VARIOGRAM procedure, 7533
CONDITION statement
CONDITION statement (SIMNORMAL), 6205
CONDITION= option
PROC MDS statement, 3713
CONDPOWER option
PROC SEQTEST statement, 6009
CONF option
PLOT statement (REG), 5494
CONFBAND= option
PROC LIFETEST statement, 3114
CONFTYPE= option
PROC LIFETEST statement, 3114
CONOVER option
EXACT statement (NPAR1WAY), 4463
OUTPUT statement (NPAR1WAY), 4465
PROC NPAR1WAY statement, 4457
CONTAIN option
MODEL statement (MIXED), 3930, 3931
CONTENTS= option
ODS HTML statement, 532
TABLES statement (FREQ), 1706
CONTGY option
OUTPUT statement (FREQ), 1695
CONTINUITY= option

TEST statement (MULTTEST), 4202
CONTINUOUS option
PLOT statement (BOXPLOT), 777
CONTRAST keyword
REPEATED statement (ANOVA), 717
CONTRAST option
REPEATED statement (GLM), 2483, 2538
CONTRAST statement
CATMOD procedure, 1110
GENMOD procedure, 1940
GLIMMIX procedure, 2123
GLM procedure, 2454
GLMPOWER procedure, 2652
HPMIXED procedure, 2816
LOGISTIC procedure, 3282
MIXED procedure, 3917
MULTTEST procedure, 4199
NLMIXED procedure, 4374
PHREG procedure, 4559
SURVEYLOGISTIC procedure, 6391
SURVEYREG procedure, 6551
CONTRAST= option
ONEWAYANOVA statement (POWER), 4860
CONTROL keyword
OUTPUT statement (LIFEREG), 3026
CONTROL statement
NLIN procedure, 4285
SURVEYSELECT procedure, 6639
CONVENIENT option
MODEL statement (TRANSREG), 7156
CONVERGE option
EM statement (MI), 3755
MODEL statement (TRANSREG), 7160
CONVERGE= option
MCMC statement (MI), 3760
MODEL statement (GENMOD), 1947
MODEL statement (LIFEREG), 3022
PROC ACECLUS statement, 675
PROC FACTOR statement, 1550
PROC FASTCLUS statement, 1634
PROC MDS statement, 3713
PROC NLIN statement, 4279
PROC PRINQUAL statement, 5204
REPEATED statement (GENMOD), 1959
TABLES statement (FREQ), 1707
CONVERGENCE option
PROC ROBUSTREG statement, 5663, 5667
CONVERGEOBJ= option
PROC NLIN statement, 4279
CONVERGEPARM= option
PROC NLIN statement, 4280
CONVF option
PROC MIXED statement, 3906, 3996
CONVG option
PROC MIXED statement, 3906, 3996
CONVG= option
MODEL statement (LIFEREG), 3022
CONVH option
PROC MIXED statement, 3906, 3996
CONVH= option
MODEL statement (GENMOD), 1948
COOKD keyword
OUTPUT statement (GLM), 2478
COORDINATES statement
KRIGE2D procedure, 2941
SIM2D procedure, 6169
VARIOGRAM procedure, 7538
COORDINATES= option
OUTPUT statement (TRANSREG), 7172
COPY statement
DISTANCE procedure, 1505
TREE procedure, 7367
CORE option
PROC MODECLUS statement, 4104
CORR option
LSMEANS statement (GENMOD), 1945
LSMEANS statement (GLIMMIX), 2146
LSMEANS statement (HPMIXED), 2822
LSMEANS statement (MIXED), 3926
LSMESTIMATE statement (GLIMMIX), 2157
PROC CALIS statement, 863
PROC CANCORR statement, 1039
PROC FACTOR statement, 1551
PROC NLMIXED statement, 4359
PROC REG statement, 5454
PROC TCALIS statement, 6740
PROC VARCLUS statement, 7470
CORR= option
ONECORR statement (POWER), 4841
PAIREDFREQ statement (POWER), 4865
PAIREDMEANS statement (POWER), 4873
REPEATED statement (GENMOD), 1961
CORRB option
MODEL statement (CATMOD), 1120
MODEL statement (GENMOD), 1948
MODEL statement (GLIMMIX), 2164
MODEL statement (LIFEREG), 3022
MODEL statement (LOGISTIC), 3293
MODEL statement (MIXED), 3930
MODEL statement (PHREG), 4566
MODEL statement (QUANTREG), 5379
MODEL statement (REG), 5473
MODEL statement (ROBUSTREG), 5669
MODEL statement (SURVEYLOGISTIC), 6399

PROC CANCORR statement, 1039
REPEATED statement (GENMOD), 1959
CORRECT=NO option
PROC NPAR1WAY statement, 4457
CORRELATION option
PROC TCALIS statement, 6728
CORRELATIONS option
PROC PRINQUAL statement, 5205
CORRESP procedure
syntax, 1323
CORRESP procedure, BY statement, 1331
CORRESP procedure, ID statement, 1331
CORRESP procedure, PROC CORRESP statement, 1323
ALL option, 1324
BENZECRI option, 1325
BINARY option, 1325
CELLCHI2 option, 1325
COLUMN= option, 1325
CP option, 1325
CROSS= option, 1325
DATA= option, 1326
DEVIATION option, 1326
DIMENS= option, 1326
EXPECTED option, 1326
FREQOUT option, 1326
GREENACRE option, 1327
MCA option, 1327
MCA= option, 1354
MININERTIA= option, 1327
MISSING option, 1327
NOCOLUMN= option, 1327
NOPRINT option, 1327
NOROW= option, 1328
NVARS= option, 1328
OBSERVED option, 1328
OUTC= option, 1328
OUTF= option, 1328
PLOTS= option, 1328
PRINT= option, 1329
PROFILE= option, 1329
ROW= option, 1330
RP option, 1330
SHORT option, 1330
SINGULAR= option, 1330
SOURCE option, 1330
UNADJUSTED option, 1330
CORRESP procedure, SUPPLEMENTARY statement, 1331
CORRESP procedure, TABLES statement, 1332
CORRESP procedure, VAR statement, 1332
CORRESP procedure, WEIGHT statement, 1333
CORRW option
REPEATED statement (GENMOD), 1959
CORRXY= option
POWER statement (GLMPOWER), 2658
COSAN statement, CALIS procedure, 873
COST= option
STRATA statement (SURVEYSELECT), 6642
COV option
LSMEANS statement (GENMOD), 1946
LSMEANS statement (GLIMMIX), 2146
LSMEANS statement (GLM), 2461
LSMEANS statement (HPMIXED), 2822
LSMEANS statement (MIXED), 3926
LSMESTIMATE statement (GLIMMIX), 2157
MCMC statement (MI), 3757, 3762
MODEL statement (CATMOD), 1120
PROC LATTICE statement, 2982
PROC NLMIXED statement, 4359
PROC PRINCOMP statement, 5151
ROCCONTRAST statement (LOGISTIC), 3311
COV statement, CALIS procedure, 874
COV statement, TCALIS procedure, 6747
COVAR option
PROC INBREED statement, 2864
COVAR= option
MODEL statement (RSREG), 5732
COVARIANCE option
PROC ROBUSTREG statement, 5663, 5665
PROC CALIS statement, 853
PROC FACTOR statement, 1551
PROC LATTICE statement, 2982
PROC PRINCOMP statement, 5151
PROC PRINQUAL statement, 5205
PROC ROBUSTREG statement, 5666
PROC TCALIS statement, 6728
PROC VARCLUS statement, 7470
COVARIATES= option
BASELINE statement (PHREG), 4541
LOGISTIC statement (POWER), 4829
COVB option
MODEL statement (CATMOD), 1120
MODEL statement (GENMOD), 1948
MODEL statement (GLIMMIX), 2164
MODEL statement (LIFEREG), 3022
MODEL statement (LOGISTIC), 3293
MODEL statement (MIXED), 3930
MODEL statement (PHREG), 4566
MODEL statement (QUANTREG), 5379
MODEL statement (REG), 5473
MODEL statement (ROBUSTREG), 5669
MODEL statement (SURVEYLOGISTIC), 6399
MODEL statement (SURVEYREG), 6555

REPEATED statement (GENMOD), 1959
COVB= option
PROC MIANALYZE statement, 3846
COVBI option
MODEL statement (GLIMMIX), 2165
MODEL statement (MIXED), 3930
COVER= option
PROC FACTOR statement, 1551
COVERLAY= option
PLOT statement (BOXPLOT), 777
COVERLAYCLIP= option
PLOT statement (BOXPLOT), 778
COVM option
PROC PHREG statement, 4536
COVODDSRATIOS= option
LOGISTIC statement (POWER), 4829
COVOUT option
PROC LIFEREG statement, 3006
PROC LOGISTIC statement, 3268
PROC PHREG statement, 4536
PROC PROBIT statement, 5264
PROC REG statement, 5454
PROC ROBUSTREG statement, 5660
COVRATIO keyword
OUTPUT statement (GLM), 2478
COVREGCOEFFS= option
LOGISTIC statement (POWER), 4829
COVS option, *see* COVSANDWICH option
COVSANDWICH option
PROC PHREG statement, 4536
COVSING= option
NLOPTIONS statement (CALIS), 890
PROC CALIS statement, 853
PROC NLMIXED statement, 4359
PROC TCALIS statement, 6728
COVTEST option
PROC MIXED statement, 3907, 3997
COVTEST statement
GLIMMIX procedure, 2127
CP
STATS= option (GLMSELECT), 2714
CP option
MODEL statement (REG), 5474
PLOT statement (REG), 5494
PROC CORRESP statement, 1325
CPC option
OUTPUT statement (TRANSREG), 7172
CPREFIX= option
CLASS statement (GLMSELECT), 2703
CLASS statement (LOGISTIC), 3280
CLASS statement (PHREG), 4557
CLASS statement (SURVEYLOGISTIC), 6389
MODEL statement (TRANSREG), 7153, 7161
CPUCOUNT option
PERFORMANCE statement (QUANTREG), 5382
PERFORMANCE statement (ROBUSTREG), 5671
CPUCOUNT= option
PERFORMANCE statement (GLMSELECT), 2716
CQC option
OUTPUT statement (TRANSREG), 7172
CR= option
PROC MODECLUS statement, 4104
CRAMV option
OUTPUT statement (FREQ), 1695
CREF= option
MCMC statement (MI), 3758
CRITMIN= option
PROC MDS statement, 3717
CROSS option
PROC MODECLUS statement, 4104
CROSS= option
PROC CORRESP statement, 1325
CROSSLIST option
PROC DISCRIM statement, 1391
PROC MODECLUS statement, 4104
TABLES statement (FREQ), 1707
CROSSLISTERR option
PROC DISCRIM statement, 1391
CROSSOVER= option
PROC TTEST statement, 7410
CROSSVALIDATE option
PROC DISCRIM statement, 1392
CRPANEL option
ASSESS statement (PHREG), 4540
CSTEP option
PROC ROBUSTREG statement, 5664
CSYMBOL= option
MCMC statement (MI), 3758, 3763
CTABLE option
MODEL statement (LOGISTIC), 3293
CTEXT= option
PLOT statement (BOXPLOT), 778
PLOT statement (REG), 5494
CUMCOL option
TABLES statement (FREQ), 1707
CUMULATIVE option
SCORE statement (LOGISTIC), 3312
CURVE= option
TWOSAMPLESURVIVAL statement (POWER), 4902
CUTOFF option
MODEL statement (QUANTREG), 5379

MODEL statement (ROBUSTREG), 5669
CUTOFF= option
PROC MDS statement, 3713
CV option
MODEL statement (TRANSREG), 7152
TABLES statement (SURVEYFREQ), 6316
CV= option
ONESAMPLEMEANS statement (POWER), 4854
PAIREDMEANS statement (POWER), 4873
PROC PLS statement, 4775
TWOSAMPLEMEANS statement (POWER), 4893
CVALS= option
OUTPUT statement (PLAN), 4740
CVDETAILS option
MODEL statement (GLMSELECT), 2707
CVMETHOD option
MODEL statement (GLMSELECT), 2707
CVREF= option
PLOT statement (BOXPLOT), 778
PLOT statement (REG), 5494
CVTEST= option
PROC PLS statement, 4776
CVWT option
TABLES statement (SURVEYFREQ), 6316

D option
MODEL statement (RSREG), 5732
PROC NPAR1WAY statement, 4458
DAMPSTEP option, 4398
NLOPTIONS statement (GLIMMIX), 394
PROC NLMIXED statement, 4359
DAMPSTEP= option
NLOPTIONS statement (CALIS), 896
DAPPROXIMATIONS option
OUTPUT statement (TRANSREG), 7173
Data set option
TYPE=ACE, 7609
TYPE=BOXPLOT, 7609
TYPE=CHARTSUM, 7609
TYPE=CORR, 7609
TYPE=COV, 7613
TYPE=CSSCP, 7613
TYPE=DISTANCE, 7614
TYPE=EST, 7614
TYPE=LINEAR, 7616
TYPE=LOGISMOD, 7616
TYPE=MIXED, 7616
TYPE=QUAD, 7616
TYPE=RAM, 7617
TYPE=SSCP, 7617
TYPE=TREE, 7618
TYPE=UCORR, 7618
TYPE=UCOV, 7619
TYPE=WEIGHT, 7619
DATA= option
OUTPUT statement (GLIMMIX), 2177
PARMS statement (NLMIXED), 4377
PRIOR statement (MIXED), 3946
PROC ACECLUS statement, 675
PROC ANOVA statement, 702
PROC BOXPLOT statement, 759
PROC CALIS statement, 853
PROC CANCORR statement, 1039
PROC CANDISC statement, 1070
PROC CATMOD statement, 1108
PROC CLUSTER statement, 1241
PROC CORRESP statement, 1326
PROC DISCRIM statement, 1392
PROC DISTANCE statement, 1493
PROC FACTOR statement, 1551
PROC FASTCLUS statement, 1635
PROC FREQ statement, 1688
PROC GAM statement, 1838
PROC GENMOD statement, 1921
PROC GLIMMIX statement, 2096
PROC GLM statement, 2446
PROC GLMMOD statement, 2626
PROC GLMPOWER statement, 2650
PROC GLMSELECT statement, 2695
PROC HPMIXED statement, 2811
PROC INBREED statement, 2864
PROC KDE statement, 2889
PROC KRIGE2D statement, 2937
PROC LATTICE statement, 2982
PROC LIFEREG statement, 3006
PROC LIFETEST statement, 3114
PROC LOESS statement, 3202
PROC LOGISTIC statement, 3268
PROC MCMC statement, 3500
PROC MDS statement, 3713
PROC MI statement, 3752
PROC MIANALYZE statement, 3846
PROC MIXED statement, 3907
PROC MODECLUS statement, 4104
PROC MULTTEST statement, 4189
PROC NESTED statement, 4257
PROC NLIN statement, 4280
PROC NLMIXED statement, 4359
PROC NPAR1WAY statement, 4458
PROC ORTHOREG statement, 4509
PROC PHREG statement, 4537
PROC PLS statement, 4776
PROC PRINCOMP statement, 5152
PROC PRINQUAL statement, 5205
PROC PROBIT statement, 5264

PROC QUANTREG statement, 5374
PROC REG statement, 5455
PROC ROBUSTREG statement, 5660
PROC RSREG statement, 5726
PROC SCORE statement, 5766
PROC SEQTEST statement, 6007
PROC SIM2D statement, 6165
PROC SIMNORMAL statement, 6203
PROC STDIZE statement, 6222
PROC STEPDISC statement, 6275
PROC SURVEYFREQ statement, 6302
PROC SURVEYLOGISTIC statement, 6383
PROC SURVEYMEANS statement, 6474
PROC SURVEYREG statement, 6543
PROC SURVEYSELECT statement, 6627
PROC TCALIS statement, 6728
PROC TPSPLINE statement, 7068
PROC TRANSREG statement, 7133
PROC TREE statement, 7363
PROC TTEST statement, 7402
PROC VARCLUS statement, 7470
PROC VARCOMP statement, 7498
PROC VARIOGRAM statement, 7529
SCORE statement (GAM), 1847
SCORE statement (LOGISTIC), 3312
SCORE statement (TPSPLINE), 7072
DATABOUNDARY option
EFFECT statement, spline (GLIMMIX), 385
EFFECT statement, spline (GLMSELECT), 385
DDF= option
MODEL statement (GLIMMIX), 2165
MODEL statement (HPMIXED), 2823
MODEL statement (MIXED), 3930
DDFM= option
MODEL statement (GLIMMIX), 2166
MODEL statement (HPMIXED), 2823
MODEL statement (MIXED), 3930
TEST statement (MULTTEST), 4202
DECIMALS= option
PROC MDS statement, 3714
default option
LMTESTS statement, 6776
DEFAULT= option
UNITS statement (LOGISTIC), 3317
UNITS statement (SURVEYLOGISTIC), 6409
DEFAULTNBINS= option
LOGISTIC statement (POWER), 4829
DEFAULTUNIT= option
LOGISTIC statement (POWER), 4829
DEFF option
MODEL statement (SURVEYREG), 6555
TABLES statement (SURVEYFREQ), 6316
DEFFBOUND= option
MODEL statement (SURVEYLOGISTIC), 6402
DEGREE option
EFFECT statement, polynomial (GLIMMIX), 380
EFFECT statement, polynomial (GLMSELECT), 380
EFFECT statement, spline (GLIMMIX), 385
EFFECT statement, spline (GLMSELECT), 385
DEGREE= option
MODEL statement (LOESS), 3209
MODEL statement (TRANSREG), 7150
TRANSFORM statement (PRINQUAL), 5218
DELETE statement, REG procedure, 5468
DELETE= option
PROC FASTCLUS statement, 1635
DELIMITER option
CLASS statement (GLMSELECT), 2702
DEMPHAS= option
PROC CALIS statement, 853
PROC TCALIS statement, 6728
DENSITY= option
PROC MODECLUS statement, 4104
DEPENDENT option
POWER statement (GLMPOWER), 2658
DEPENDENT= option
OUTPUT statement (TRANSREG), 7173
DEPENDENTFDR option
PROC MULTTEST statement, 4189, 4217
DEPONLY option
MEANS statement (GLM), 2470
DEPSILON= option
COMPUTE statement (VARIOGRAM), 7536
DER option
OUTPUT statement (GLIMMIX), 2180
OUTPUT statement (NLIN), 4289
PREDICT statement (NLMIXED), 4378
DER statement
NLIN procedure, 4285
DERIVATIVES option
OUTPUT statement (GLIMMIX), 2180
DESCENDING option
CLASS statement (GAM), 1841
CLASS statement (GENMOD), 1938
CLASS statement (GLMSELECT), 2703
CLASS statement (LOGISTIC), 3280
CLASS statement (PHREG), 4557

CLASS statement (SURVEYLOGISTIC), 6389
MODEL statement, 1843, 2163, 3288, 6396
PROC GAM statement, 1838
PROC LOGISTIC statement, 3268
PROC TREE statement, 7363
DESCRIPTION= option
PLOT statement (BOXPLOT), 778
PLOT statement (GLMPOWER), 2656
PLOT statement (POWER), 4884
PLOT statement (REG), 5494
PROC LIFETEST statement, 3115
PROC TREE statement, 7363
DESIGN option
MODEL statement (CATMOD), 1120
DESIGN statement
SEQDESIGN procedure, 5803
DESIGN= option
OUTPUT statement (TRANSREG), 7173
DETAIL option
MODEL statement (TRANSREG), 7161
DETAILS option
EFFECT statement, multimember (GLIMMIX), 379
EFFECT statement, multimember (GLMSELECT), 379
EFFECT statement, polynomial (GLIMMIX), 381
EFFECT statement, polynomial (GLMSELECT), 381
EFFECT statement, spline (GLIMMIX), 385
EFFECT statement, spline (GLMSELECT), 385
MODEL statement (GLMSELECT), 2708
MODEL statement (LOESS), 3209
MODEL statement (LOGISTIC), 3293
MODEL statement (PHREG), 4566
MODEL statement (REG), 5474
MTEST statement (REG), 5481
PERFORMANCE statement (GLMSELECT), 2717
PERFORMANCE statement (QUANTREG), 5383
PERFORMANCE statement (ROBUSTREG), 5671
PROC PLS statement, 4776
DETERM statement, TCALIS procedure, 6750
DEVIANCE statement, GENMOD procedure, 1942, 1958
DEVIATION option
PROC CORRESP statement, 1326
TABLES statement (FREQ), 1708
DEVIATIONS option
MODEL statement (TRANSREG), 7153
DF= option
CONTRAST statement (GLIMMIX), 2126
CONTRAST statement (HPMIXED), 2817
CONTRAST statement (MIXED), 3919
CONTRAST statement (NLMIXED), 4375
COVTEST statement (GLIMMIX), 2132
ESTIMATE statement (GLIMMIX), 2136
ESTIMATE statement (HPMIXED), 2819
ESTIMATE statement (MIXED), 3921
ESTIMATE statement (NLMIXED), 4375
LSMEANS statement (GLIMMIX), 2146
LSMEANS statement (HPMIXED), 2822
LSMEANS statement (MIXED), 3926
LSMESTIMATE statement (GLIMMIX), 2157
MODEL statement (GLIMMIX), 2165
MODEL statement (SURVEYREG), 6555
MODEL statement (TPSPLINE), 7070
PREDICT statement (NLMIXED), 4378
PROC NLMIXED statement, 4359
RANDOM statement (NLMIXED), 4379
REPWEIGHTS statement (SURVEYFREQ), 6311
REPWEIGHTS statement (SURVEYLOGISTIC), 6406
REPWEIGHTS statement (SURVEYMEANS), 6488
REPWEIGHTS statement (SURVEYREG), 6558
TABLES statement (SURVEYFREQ), 6316
DFADJ option
DOMAIN statement (SURVEYMEANS), 6486
VARMETHOD=BRR (PROC SURVEYFREQ statement), 6305
VARMETHOD=BRR (PROC SURVEYMEANS statement), 6480
VARMETHOD=JK (PROC SURVEYFREQ statement), 6308
VARMETHOD=JK (PROC SURVEYMEANS statement), 6482
DFBETAS= option
OUTPUT statement (LOGISTIC), 3306
DFBW option
PROC MIXED statement, 3907
DFE= option
PROC CALIS statement, 854
PROC TCALIS statement, 6729
DFFITS keyword
OUTPUT statement (GLM), 2478
DFMETHOD= option
MODEL statement (LOESS), 3209
DFMETHOD=APPROX(Cutoff=) option

MODEL statement (LOESS), 3210
DFMETHOD=APPROX(Quantile=) option
MODEL statement (LOESS), 3210
DFR= option
PROC CALIS statement, 866
PROC TCALIS statement, 6742
DFREDUCE= option
FITINDEX statement, 6760
PROC CALIS statement, 854
PROC TCALIS statement, 6729
DIAG= option
PROC MCMC statement, 3498
DIAGNOSTICS option
MODEL statement (GENMOD), 1948
MODEL statement (QUANTREG), 5379
MODEL statement (ROBUSTREG), 5669
DIAGNOSTICS= option
BAYES statement(PHREG), 4547
PROC MCMC statement, 3498
DIAHES option
PROC NLMIXED statement, 4359
DIC option
PROC MCMC statement, 3500
DIFCHISQ= option
OUTPUT statement (LOGISTIC), 3306
DIFDEV= option
OUTPUT statement (LOGISTIC), 3307
DIFF option
LSMEANS statement (GENMOD), 1946
LSMEANS statement (GLIMMIX), 2147
LSMEANS statement (MIXED), 3926
DIFF= option
HAZARDRATIO statement (PHREG), 4563
ODDSRATIO statement (LOGISTIC), 3304
STRATA statement (LIFETEST), 3128
DIM= option
PROC CLUSTER statement, 1242
DIMENS= option
PROC CORRESP statement, 1326
DIMENSION= option
PROC MDS statement, 3714
PROC MODECLUS statement, 4104
DIRECT option
MODEL statement (LOESS), 3210
DIRECT statement, CATMOD procedure, 1113
DIRECTIONS statement
VARIOGRAM procedure, 7539
DISCPROPDIFF= option
PAIREDFREQ statement (POWER), 4865
DISCPROPORTIONS= option
PAIREDFREQ statement (POWER), 4865
DISCPROPRATIO= option
PAIREDFREQ statement (POWER), 4865
DISCRETE= option
PROC MCMC statement, 3497
DISCRIM option
MONOTONE statement (MI), 3765
DISCRIM procedure
syntax, 1389
DISCRIM procedure, BY statement, 1397
DISCRIM procedure, CLASS statement, 1398
DISCRIM procedure, FREQ statement, 1398
DISCRIM procedure, ID statement, 1398
DISCRIM procedure, PRIORS statement, 1399
DISCRIM procedure, PROC DISCRIM statement, 1389
ALL option, 1391
ANOVA option, 1391
BCORR option, 1391
BCOV option, 1391
BSSCP option, 1391
CAN option, 1391
CANONICAL option, 1391
CANPREFIX= option, 1391
CROSSLIST option, 1391
CROSSLISTERR option, 1391
CROSSVALIDATE option, 1392
DATA= option, 1392
DISTANCE option, 1392
K= option, 1392
KERNEL= option, 1392
KPROP= option, 1392
LIST option, 1393
LISTERR option, 1393
MAHALANOBIS option, 1392
MANOVA option, 1393
METHOD= option, 1393
METRIC= option, 1393
NCAN= option, 1393
NOCLASSIFY option, 1393
NOPRINT option, 1393
OUT= option, 1394
OUTCROSS= option, 1394
OUTD= option, 1394
OUTSTAT= option, 1394
PCORR option, 1394
PCOV option, 1394
POOL= option, 1394
POSTERR option, 1395
PSSCP option, 1395
R= option, 1395
SHORT option, 1395
SIMPLE option, 1395
SINGULAR= option, 1395
SLPOOL= option, 1396
STDMEAN option, 1396
TCORR option, 1396
TCOV option, 1396

TESTDATA= option, 1396
TESTLIST option, 1396
TESTLISTERR option, 1396
TESTOUT= option, 1396
TESTOUTD= option, 1396
THRESHOLD= option, 1397
TSSCP option, 1397
WCORR option, 1397
WCOV option, 1397
WSSCP option, 1397
DISCRIM procedure, TESTCLASS statement, 1400
DISCRIM procedure, TESTFREQ statement, 1400
DISCRIM procedure, TESTID statement, 1400
DISCRIM procedure, VAR statement, 1400
DISCRIM procedure, WEIGHT statement, 1401
DISPLAYINIT option
MCMC statement (MI), 3759
DISSIMILAR option
PROC TREE statement, 7363
DIST = option
MODEL statement (GAM), 1845
DIST= option
MODEL statement (GENMOD), 1948
MODEL statement (GLIMMIX), 2168
ONECORR statement (POWER), 4841
ONESAMPLEMEANS statement (POWER), 4854
PAIREDFREQ statement (POWER), 4865
PAIREDMEANS statement (POWER), 4873
PROC TTEST statement, 7402
TWOSAMPLEMEANS statement (POWER), 4893
DISTANCE option
PROC CANDISC statement, 1070
PROC DISCRIM statement, 1392
PROC FASTCLUS statement, 1635
DISTANCE procedure, BY statement, 1505
DISTANCE procedure, COPY statement, 1505
DISTANCE procedure, FREQ statement, 1506
DISTANCE procedure, ID statement, 1505
DISTANCE procedure, PROC DISTANCE statement, 1491
ABSENT= option, 1493
ADD= option, 1493
DATA= option, 1493
FUZZ= option, 1493
INITIAL= option, 1493
METHOD= option, 1494
MULT= option, 1497
NOMISS, 1497
NORM option, 1498
NOSTD, 1498
OUT= option, 1498
OUTSDZ= option, 1498
PREFIX= option, 1498
RANKSCORE= option, 1498
REPLACE, 1499
REPONLY, 1499
SHAPE= option, 1499
SNORM option, 1499
STDONLY option, 1499
UNDEF= option, 1499
VARDEF= option, 1499
DISTANCE procedure, VAR statement
ABSENT= option, 1503
MISSING= option, 1503
ORDER= option, 1504
WEIGHTS= option, 1504
DISTANCE procedure, WGT statement, 1506
DISTANCE= option
MODEL statement (TPSPLINE), 7070
DISTRIBUTION= option
MODEL statement (GLIMMIX), 2168
MODEL statement (LIFEREG), 3023
DIVISOR= option
ESTIMATE statement (GLIMMIX), 2136
ESTIMATE statement (GLM), 2457
ESTIMATE statement (HPMIXED), 2819
ESTIMATE statement (MIXED), 3921
ESTIMATE statement (SURVEYREG), 6554
LSMESTIMATE statement (GLIMMIX), 2158
DK= option
PROC MODECLUS statement, 4105
DOCK= option
PROC MODECLUS statement, 4105
PROC TREE statement, 7363
DOCUMENT procedure
LIST statement, 608
REPLAY statement, 607
DOMAIN statement
SURVEYLOGISTIC procedure, 6394
SURVEYMEANS procedure, 6485
SURVEYREG procedure, 6552
DR= option
PROC MODECLUS statement, 4105
DREPLACE option
OUTPUT statement (TRANSREG), 7173
DRIFT option
PROC FASTCLUS statement, 1635
DROP= option
MODEL statement (GLMSELECT), 2711
DROPSQUARE= option
MODEL statement (LOESS), 3210

DSCALE
MODEL statement (GENMOD), 1953
DUMMY option
MODEL statement (TRANSREG), 7161
PROC PRINQUAL statement, 5205
DUNCAN option
MEANS statement (ANOVA), 711
MEANS statement (GLM), 2470
DUNNETT option
MEANS statement (ANOVA), 711
MEANS statement (GLM), 2470
DUNNETTL option
MEANS statement (ANOVA), 712
MEANS statement (GLM), 2470
DUNNETTU option
MEANS statement (ANOVA), 712
MEANS statement (GLM), 2470
DW option
MODEL statement (REG), 5474
DWPROB option
MODEL statement (REG), 5474

E option
CONTRAST statement (GENMOD), 1942
CONTRAST statement (GLIMMIX), 2126
CONTRAST statement (GLM), 2454
CONTRAST statement (HPMIXED), 2818
CONTRAST statement (LOGISTIC), 3283
CONTRAST statement (MIXED), 3919
CONTRAST statement (PHREG), 4561
CONTRAST statement (SURVEYLOGISTIC), 6393
CONTRAST statement (SURVEYREG), 6551
ESTIMATE statement (GENMOD), 1943
ESTIMATE statement (GLIMMIX), 2137
ESTIMATE statement (GLM), 2457
ESTIMATE statement (HPMIXED), 2820
ESTIMATE statement (MIXED), 3921
ESTIMATE statement (SURVEYREG), 6554
HAZARDRATIO statement (PHREG), 4563
LSMEANS statement (GENMOD), 1946
LSMEANS statement (GLIMMIX), 2147
LSMEANS statement (GLM), 2461
LSMEANS statement (HPMIXED), 2822
LSMEANS statement (MIXED), 3927
LSMESTIMATE statement (GLIMMIX), 2158
MODEL statement (GLIMMIX), 2170
MODEL statement (GLM), 2475
MODEL statement (MIXED), 3933
ROCCONTRAST statement (LOGISTIC), 3311
TEST statement (HPMIXED), 2831
TEST statement (PHREG), 4578
E1 option
MODEL statement (GLIMMIX), 2171
MODEL statement (GLM), 2475
MODEL statement (MIXED), 3933
E2 option
MODEL statement (GLIMMIX), 2171
MODEL statement (GLM), 2475
MODEL statement (MIXED), 3933
E3 option
MODEL statement (GLIMMIX), 2171
MODEL statement (GLM), 2475
MODEL statement (MIXED), 3933
TEST statement (HPMIXED), 2831
E4 option
MODEL statement (GLM), 2475
E= effects
REPEATED statement (ANOVA), 720
E= option
CONTRAST statement (GLM), 2455
MANOVA statement (ANOVA), 706
MANOVA statement (GLM), 2464
MEANS statement (ANOVA), 712
MEANS statement (GLM), 2471
REPEATED statement (GLM), 2486
EARLY option
PROC MODECLUS statement, 4105
EBOPT option
PROC NLMIXED statement, 4359
EBSSFRAC option
PROC NLMIXED statement, 4359
EBSSTOL option
PROC NLMIXED statement, 4359
EBSTEPS option
PROC NLMIXED statement, 4360
EBSUBSTEPS option
PROC NLMIXED statement, 4360
EBTOL option
PROC NLMIXED statement, 4360
EBZSTART option
PROC NLMIXED statement, 4360
ECORR option
PROC NLMIXED statement, 4360
ECORRB option
REPEATED statement (GENMOD), 1959
ECOV option
PROC NLMIXED statement, 4360
ECOVB option
REPEATED statement (GENMOD), 1959
EDER option
PROC NLMIXED statement, 4360
EDF option
MODEL statement (REG), 5474

OUTPUT statement (NPAR1WAY), 4465
PLOT statement (REG), 5494
PROC NPAR1WAY statement, 4458
PROC REG statement, 5455
EDF= option
PROC CALIS statement, 854
PROC CANCORR statement, 1039
PROC MIANALYZE statement, 3846
PROC TCALIS statement, 6729
EFF option
PROC ROBUSTREG statement, 5665, 5667
EFFECT statement
collection effect, 378
GLIMMIX procedure, 377, 2134
GLMSELECT procedure, 377, 2706
multimember effect, 378
polynomial effect, 380
QUANTREG procedure, 5378
spline effect, 384
EFFECT= modifier
INFLUENCE option, MODEL statement (MIXED), 3935
EFFECTS option
MODEL statement (TRANSREG), 7153
EFFECTSIZE option
MODEL statement (GLM), 2475
EFFECTVAR= option
PROC MIANALYZE statement, 3846
EFFPART statement, TCALIS procedure, 6751
EIGENVECTORS option
PROC FACTOR statement, 1551
ELSM option
LSMESTIMATE statement (GLIMMIX), 2158
EM statement
MI procedure, 3755
EMPIRICAL option
MIXED, 3907
PROC NLMIXED statement, 4360
EMPIRICAL= option
PROC GLIMMIX statement, 2096
ENDCNST statement
MCMC procedure, 3509
ENDGRID option
PLOT statement (BOXPLOT), 778
ENDNODATA statement
MCMC procedure, 3511
ENDPRIOR statement
MCMC procedure, 3511
ENTRYTIME= option
MODEL statement (PHREG), 4567
EPOINT transformation
MODEL statement (TRANSREG), 7142
EPSILON = option
MODEL statement (GAM), 1845
EPSILON= option
MODEL statement (CATMOD), 1121
PROC MDS statement, 3714
PROC MULTTEST statement, 4190
PROC PLS statement, METHOD=PLS option, 4777
PROC PLS statement, MISSING=EM option, 4777
PROC VARCOMP statement, 7498
EPSSCORE = option
MODEL statement (GAM), 1845
EQCONS= option
PARMS statement (MIXED), 3944
EQKAP option
OUTPUT statement (FREQ), 1695
EQOR option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1695
EQUIVBOUNDS= option
ONESAMPLEFREQ statement (POWER), 4846
EQWKP option
OUTPUT statement (FREQ), 1695
ERR= option
MODEL statement (GENMOD), 1948
ERROR= option
MODEL statement (GLIMMIX), 2168
ERRSPEND option
PROC SEQDESIGN statement, 5801
PROC SEQTEST statement, 6009
ERRSPENDADJ= option
PROC SEQTEST statement, 6005
ERRSPENDMIN= option
PROC SEQTEST statement, 6007
ESTDATA= option
PROC CALIS statement, 856
PROC TCALIS statement, 6730
ESTIMATE option
EXACT statement (LOGISTIC), 3285
ROCCONTRAST statement (LOGISTIC), 3311
ESTIMATE statement
GENMOD procedure, 1943
GLIMMIX procedure, 2134
GLM procedure, 2456, 2509
HPMIXED procedure, 2818
MIXED procedure, 3920
NLMIXED procedure, 4375
SURVEYREG procedure, 6553
ESTIMATE= option
CONTRAST statement (CATMOD), 1111
CONTRAST statement (LOGISTIC), 3283
CONTRAST statement (PHREG), 4561

CONTRAST statement (SURVEYLOGISTIC), 6394
ESTIMATES modifier
INFLUENCE option, MODEL statement (MIXED), 3935
ESTIMATES option
COVTEST statement (GLIMMIX), 2133
ETYPE option
LSMEANS statement (GLM), 2461
ETYPE= option
CONTRAST statement (GLM), 2455
MANOVA statement (GLM), 2465
MEANS statement (GLM), 2471
TEST statement (GLM), 2486
EVENLY option
MODEL statement (TRANSREG), 7150
TRANSFORM statement (PRINQUAL), 5218
EVENT= option
MODEL statement, 3288
EVENTSYMBOL= option
PROC LIFETEST statement, 3115
EXACT option
OUTPUT statement (FREQ), 1695
EXACT statement
FREQ procedure, 1691
LOGISTIC procedure, 3285
NPAR1WAY procedure, 4462
EXACTONLY option
PROC LOGISTIC statement, 3268
EXACTOPTIONS option
PROC LOGISTIC statement, 3269
EXKNOTS= option
MODEL statement (TRANSREG), 7151
EXP option
ESTIMATE statement (GENMOD), 1943
ESTIMATE statement (GLIMMIX), 2137
LSMESTIMATE statement (GLIMMIX), 2158
EXP transformation
MODEL statement (TRANSREG), 7144
TRANSFORM statement (MI), 3767
TRANSFORM statement (PRINQUAL), 5214
EXPECTED option
MODEL statement (GENMOD), 1948
PROC CORRESP statement, 1326
TABLES statement (FREQ), 1708
TABLES statement (SURVEYFREQ), 6317
EXPEST option
MODEL statement (LOGISTIC), 3293
MODEL statement (SURVEYLOGISTIC), 6399
EXPHESSIAN option
PROC GLIMMIX statement, 2099

_F_ specification
MODEL statement (CATMOD), 1118, 1140
FACTOR procedure, 1548
syntax, 1548
FACTOR procedure, BY statement, 1564
FACTOR procedure, FREQ statement, 1564
FACTOR procedure, PARTIAL statement, 1565
FACTOR procedure, PRIORS statement, 1565
FACTOR procedure, PROC FACTOR statement, 1548
ALL option, 1550
ALPHA= option, 1550
CONVERGE= option, 1550
CORR option, 1551
COVARIANCE option, 1551
COVER= option, 1551
DATA= option, 1551
EIGENVECTORS option, 1551
FLAG= option, 1551
FUZZ= option, 1551
GAMMA= option, 1552
HEYWOOD option, 1552
HKPOWER= option, 1552
MAXITER= option, 1552
METHOD= option, 1552
MINEIGEN= option, 1553
MSA option, 1553
NFACTORS= option, 1554
NOBS= option, 1554
NOCORR option, 1554
NOINT option, 1554
NOPRINT option, 1554
NOPROMAXNORM option, 1554
NORM= option, 1554
NPLOTS= option, 1555
OUT= option, 1555
OUTSTAT= option, 1555
PARPREFIX= option, 1555
PLOT option, 1555
PLOTREF option, 1555
PLOTS= option, 1555
POWER= option, 1557
PREFIX= option, 1558
PREPLOT option, 1558
PREROTATE= option, 1558
PRINT option, 1558
PRIORS= option, 1558
PROPORTION= option, 1559
RANDOM= option, 1559
RCONVERGE= option, 1559
REORDER option, 1560
RESIDUALS option, 1560

RITER= option, 1560
ROTATE= option, 1560
ROUND option, 1562
SCORE option, 1562
SCREE option, 1562
SE option, 1562
SIMPLE option, 1563
SINGULAR= option, 1563
TARGET= option, 1563
TAU= option, 1563
ULTRAHEYWOOD option, 1563
VARDEF= option, 1563
WEIGHT option, 1563
FACTOR procedure, VAR statement, 1565
FACTOR procedure, WEIGHT statement, 1566
factor specification
REPEATED statement (GLM), 2482
FACTOR statement, CALIS procedure, 877
FACTOR statement, TCALIS procedure, 6752
factor-value-settings option
OUTPUT statement (PLAN), 4740
FACTORS statement
CATMOD procedure, 1114
PLAN procedure, 4737
FAST option
MODEL statement (LOGISTIC), 3294
FASTCLUS procedure
MAXCLUSTERS= option, 1623
RADIUS= option, 1623
syntax, 1632
FASTCLUS procedure, BY statement, 1640
FASTCLUS procedure, FREQ statement, 1640
FASTCLUS procedure, ID statement, 1641
FASTCLUS procedure, PROC FASTCLUS statement, 1632
BINS= option, 1634
CLUSTER= option, 1634
CLUSTERLABEL= option, 1634
CONVERGE= option, 1634
DATA= option, 1635
DELETE= option, 1635
DISTANCE option, 1635
DRIFT option, 1635
HC= option, 1635
HP= option, 1635
IMPUTE option, 1636
INSTAT= option, 1636
IRLS option, 1636
L= option, 1636
LEAST= option, 1636
LIST option, 1637
MAXCLUSTERS= option, 1632
MAXITER= option, 1638
MEAN= option, 1638
NOMISS option, 1638
NOPRINT option, 1638
OUT= option, 1638
OUTITER option, 1638
OUTS= option, 1638
OUTSEED= option, 1638
OUTSTAT= option, 1638
RADIUS= option, 1632
RANDOM= option, 1639
REPLACE= option, 1639
SEED= option, 1639
SHORT option, 1639
STRICT= option, 1639
SUMMARY option, 1639
VARDEF= option, 1639
FASTCLUS procedure, VAR statement, 1641
FASTCLUS procedure, WEIGHT statement, 1641
FAY= option
VARMETHOD=BRR (PROC SURVEYFREQ statement), 6306
VARMETHOD=BRR (PROC SURVEYLOGISTIC statement), 6386
VARMETHOD=BRR (PROC SURVEYMEANS statement), 6480
VARMETHOD=BRR (PROC SURVEYREG statement), 6546
FCONV option
NLOPTIONS statement (GLIMMIX), 394
NLOPTIONS statement (HPMIXED), 394
NLOPTIONS statement (TCALIS), 394
FCONV2 option
NLOPTIONS statement (GLIMMIX), 395
NLOPTIONS statement (HPMIXED), 395
NLOPTIONS statement (TCALIS), 395
FCONV2= option
NLOPTIONS statement (CALIS), 892, 895
PROC NLMIXED statement, 4361
FCONV= option
MODEL statement (LOGISTIC), 3294
MODEL statement (PHREG), 4567
MODEL statement (SURVEYLOGISTIC), 6399
NLOPTIONS statement (CALIS), 889, 892
PROC CALIS statement, 854
PROC NLMIXED statement, 4361
PROC TCALIS statement, 6729
FD= option
PROC NLMIXED statement, 4361
FDCODE option
PROC CALIS statement, 854
FDHESSIAN= option
PROC NLMIXED statement, 4362
FDIGITS= option, 4393

NLOPTIONS statement (CALIS), 893
PROC GLIMMIX statement, 2099
PROC NLMIXED statement, 4362
FDR option
PROC MULTTEST statement, 4190, 4216
FDRBOOT option
PROC MULTTEST statement, 4190, 4217
FDRPERM option
PROC MULTTEST statement, 4190, 4217
FIADJUST option
PROC ROBUSTREG statement, 5664
FILE= option
ODS destination statement, 536
ODS PDF statement, 546
FILLCHAR= option
PROC TREE statement, 7363
FIRTH option
MODEL statement (LOGISTIC), 3294
MODEL statement (PHREG), 4567
FISHER option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1695
TABLES statement (FREQ), 1708
TEST statement (MULTTEST), 4199, 4201, 4208, 4236
FISHER_C option
PROC MULTTEST statement, 4190, 4215
FIT= option
PROC MDS statement, 3714
FITSTAT option
SCORE statement (LOGISTIC), 3313
FITSTATISTICS
DETAILS=STEPS option (GLMSELECT), 2708
FIXED= option
MODEL statement (VARCOMP), 7500
FLAG= option
PROC FACTOR statement, 1551
FLAT option
PRIOR statement (MIXED), 3947
FLOW option
PROC NLIN statement, 4280
PROC NLMIXED statement, 4362
FOLLOWUPTIME= option
TWOSAMPLESURVIVAL statement (POWER), 4903
FONT= option
PLOT statement (BOXPLOT), 778
FORM= option
MODEL statement (KRIGE2D), 2945
SIMULATE statement (SIM2D), 6172
FORMAT= option
TABLES statement (FREQ), 1708
FORMCHAR= option
PROC FREQ statement, 1689
PROC LIFETEST statement, 3115
FORMULA= option
PROC MDS statement, 3715
FRAME= option
ODS HTML statement, 532
FREQ option
MODEL statement (CATMOD), 1121
FREQ procedure
syntax, 1687
FREQ procedure, BY statement, 1690
FREQ procedure, EXACT statement, 1691
AGREE option, 1692
ALPHA= option, 1693
BINOMIAL option, 1692
CHISQ option, 1692, 1812
COMOR option, 1692
EQOR option, 1692
FISHER option, 1692
JT option, 1692
KAPPA option, 1692
LRCHI option, 1692
MAXTIME= option, 1693
MC option, 1693
MCNEM option, 1692
MEASURES option, 1692
MHCHI option, 1692
N= option, 1693
OR option, 1692, 1812
PCHI option, 1692
PCORR option, 1692
POINT option, 1693
RISKDIFF option, 1692
RISKDIFF1 option, 1692
RISKDIFF2 option, 1692
SCORR option, 1692
SEED= option, 1694
TREND option, 1692, 1819
WTKAP option, 1692
FREQ procedure, OUTPUT statement, 1694
AGREE option, 1695
AJCHI option, 1695
ALL option, 1695
BDCHI option, 1695
BINOMIAL option, 1695
CHISQ option, 1695
CMH option, 1695
CMH1 option, 1695
CMH2 option, 1695
CMHCOR option, 1695
CMHGA option, 1695
CMHRMS option, 1695
COCHQ option, 1695
CONTGY option, 1695

CRAMV option, 1695
EQKAP option, 1695
EQOR option, 1695
EQWKP option, 1695
EXACT option, 1695
FISHER option, 1695
GAMMA option, 1695
JT option, 1696
KAPPA option, 1696
KENTB option, 1696
LAMCR option, 1696
LAMDAS option, 1696
LAMRC option, 1696
LGOR option, 1696
LGRRC1 option, 1696
LGRRC2 option, 1696
LRCHI option, 1696, 1815
MCNEM option, 1696
MEASURES option, 1696
MHCHI option, 1696
MHOR option, 1696
MHRRC1 option, 1696
MHRRC2 option, 1696
N option, 1696
NMISS option, 1696
OR option, 1696
OUT= option, 1694
PCHI option, 1696, 1815
PCORR option, 1696
PHI option, 1696
PLCORR option, 1696
RDIF1 option, 1696
RDIF2 option, 1696
RELRISK option, 1696
RISKDIFF option, 1696
RISKDIFF1 option, 1696
RISKDIFF2 option, 1696
RRC1 option, 1696
RRC2 option, 1696
RSK1 option, 1696
RSK11 option, 1696
RSK12 option, 1696
RSK2 option, 1696
RSK21 option, 1696
RSK22 option, 1697
SCORR option, 1697
SMDCR option, 1697
SMDRC option, 1697
STUTC option, 1697
TREND option, 1697
TSYMM option, 1697
U option, 1697
UCR option, 1697
URC option, 1697
WTKAP option, 1697
FREQ procedure, PROC FREQ statement, 1688
COMPRESS option, 1688
DATA= option, 1688
FORMCHAR= option, 1689
NLEVELS option, 1689
NOPRINT option, 1689
ORDER= option, 1690
PAGE option, 1690
FREQ procedure, TABLES statement, 1697
AGREE option, 1700
ALL option, 1701
ALPHA= option, 1701
BDT option, 1701
BINOMIAL option, 1701
BINOMIALC option, 1705
CELLCHI2 option, 1705
CHISQ option, 1705, 1732, 1812
CL option, 1705
CMH option, 1706
CMH1 option, 1706
CMH2 option, 1706
CONTENTS= option, 1706
CONVERGE= option, 1707
CROSSLIST option, 1707
CUMCOL option, 1707
DEVIATION option, 1708
EXPECTED option, 1708
FISHER option, 1708
FORMAT= option, 1708
JT option, 1708
LIST option, 1709
MAXITER= option, 1709
MEASURES option, 1709
MISSING option, 1709
MISSPRINT option, 1710
NOCOL option, 1710
NOCUM option, 1710
NOFREQ option, 1710
NOPERCENT option, 1710
NOPRINT option, 1710
NOROW option, 1710
NOSPARSE option, 1710
NOWARN option, 1711
OUT= option, 1711
OUTCUM option, 1711
OUTEXPECT option, 1711, 1800
OUTPCT option, 1711
PLCORR option, 1711
PLOTS= option, 1712
PRINTKWT option, 1717
RELRISK option, 1717, 1812
RISKDIFF option, 1717
RISKDIFFC option, 1720

SCORES= option, 1720, 1822
SCOROUT option, 1720
SPARSE option, 1721, 1800
TESTF= option, 1721, 1732
TESTP= option, 1721, 1732, 1806
TOTPCT option, 1721
TREND option, 1722, 1819
FREQ procedure, TEST statement, 1722
AGREE option, 1723
GAMMA option, 1723
KAPPA option, 1723
KENTB option, 1723
MEASURES option, 1723
PCORR option, 1723
SCORR option, 1723
SMDCR option, 1723, 1819
SMDRC option, 1723
STUTC option, 1723
WTKAP option, 1723
FREQ procedure, WEIGHT statement, 1723
ZEROS option, 1724
FREQ statement
ANOVA procedure, 705
CALIS procedure, 880
CANDISC procedure, 1073
DISCRIM procedure, 1398
DISTANCE procedure, 1506
FACTOR procedure, 1564
GAM procedure, 1841
GENMOD procedure, 1944
GLIMMIX procedure, 2140
GLM procedure, 2457
GLMMOD procedure, 2629
GLMSELECT procedure, 2706
KDE procedure, 2896
LIFETEST procedure, 3125
LOGISTIC procedure, 3287
MI procedure, 3756
MODECLUS procedure, 4109
MULTTEST procedure, 4200
NPAR1WAY procedure, 4465
PHREG procedure, 4561
PRINCOMP procedure, 5157
PRINQUAL procedure, 5211
REG procedure, 5468
STDIZE procedure, 6226
STEPDISC procedure, 6279
SURVEYLOGISTIC procedure, 6395
TCALIS procedure, 6763
TPSPLINE procedure, 7069
TRANSREG procedure, 7138
TREE procedure, 7368
TTEST procedure, 7409
VARCLUS procedure, 7474
FREQOUT option
PROC CORRESP statement, 1326
FRONTREF option
PLOT statement (BOXPLOT), 778
FSIZE option
NLOPTIONS statement (GLIMMIX), 395
NLOPTIONS statement (HPMIXED), 395
NLOPTIONS statement (TCALIS), 395
FSIZE= option
NLOPTIONS statement (CALIS), 893
PROC NLMIXED statement, 4362
FT option
TEST statement (MULTTEST), 4201, 4206, 4229
FTEST option
LSMESTIMATE statement (GLIMMIX), 2158
FTOL option
NLOPTIONS statement (GLIMMIX), 394
NLOPTIONS statement (HPMIXED), 394
NLOPTIONS statement (TCALIS), 394
FTOL2 option
NLOPTIONS statement (GLIMMIX), 395
NLOPTIONS statement (HPMIXED), 395
NLOPTIONS statement (TCALIS), 395
FTOL2= option
NLOPTIONS statement (CALIS), 892, 895
FTOL= option
NLOPTIONS statement (CALIS), 889, 892
PROC CALIS statement, 854
PROC TCALIS statement, 6729
FULLX option
MODEL statement (MIXED), 3925, 3933
FUZZ= option
PROC DISTANCE statement, 1493
PROC FACTOR statement, 1551
PROC STDIZE statement, 6222
FVALUE
STATS= option (GLMSELECT), 2714
FWDLINK statement, GENMOD procedure, 1944, 1958
FWLS= option
PROC ROBUSTREG statement, 5661

G option
RANDOM statement (GLIMMIX), 2188
RANDOM statement (MIXED), 3950
G4 option
PROC NLIN statement, 4280
G4= option
NLOPTIONS statement (CALIS), 889
PROC CALIS statement, 854
PROC NLMIXED statement, 4363
PROC TCALIS statement, 6729

GABRIEL option
MEANS statement (ANOVA), 712
MEANS statement (GLM), 2471
GAM procedure, 1838
syntax, 1838
GAM procedure, BY statement, 1840
GAM procedure, CLASS statement, 1840
DESCENDING option, 1841
ORDER= option, 1841
TRUNCATE option, 1841
GAM procedure, FREQ statement, 1841
GAM procedure, MODEL statement, 1842
ALPHA= option, 1844
ANODEV= option, 1844
DESCENDING option, 1843
DIST= option, 1845
EPSILON= option, 1845
EPSSCORE= option, 1845
ITPRINT option, 1845
MAXITER= option, 1845
MAXITSCORE= option, 1845
METHOD= option, 1845
ORDER= option, 1844
GAM procedure, OUTPUT statement, 1846
OUT= option, 1846
GAM procedure, PROC GAM statement, 1838
ADDITIVE option, 1840
all option, 1839
CLM option, 1840
COMMONAXES option, 1840
component option, 1839
DATA= option, 1838
DESCENDING option, 1838
none option, 1839
ORDER option, 1838
plots(unpack) option, 1839
PLOTS= option, 1839
UNPACKPANELS option, 1840
GAM procedure, SCORE statement, 1847
DATA= option, 1847
OUT= option, 1847
GAMMA option
OUTPUT statement (FREQ), 1695
TEST statement (FREQ), 1723
GAMMA= option
FACTOR statement, 6753
PROC FACTOR statement, 1552
GC option
RANDOM statement (GLIMMIX), 2188
RANDOM statement (MIXED), 3950
GCI option
RANDOM statement (GLIMMIX), 2188
RANDOM statement (MIXED), 3950
GCONV option
NLOPTIONS statement (GLIMMIX), 395
NLOPTIONS statement (HPMIXED), 395
NLOPTIONS statement (TCALIS), 395
GCONV2 option
NLOPTIONS statement (GLIMMIX), 396
NLOPTIONS statement (HPMIXED), 396
NLOPTIONS statement (TCALIS), 396
GCONV2= option
NLOPTIONS statement (CALIS), 893
GCONV= option
MODEL statement (LOGISTIC), 3294
MODEL statement (PHREG), 4567
MODEL statement (SURVEYLOGISTIC), 6399
NLOPTIONS statement (CALIS), 889, 893, 895
PROC CALIS statement, 855
PROC NLMIXED statement, 4363
PROC TCALIS statement, 6730
GCONVERGE= option
PROC MDS statement, 3715
GCOORD= option
RANDOM statement (GLIMMIX), 2188
GCORR option
RANDOM statement (GLIMMIX), 2188
RANDOM statement (MIXED), 3951
GCV option
MODEL statement (TRANSREG), 7152
GDATA= option
RANDOM statement (MIXED), 3951
GENDER statement, INBREED procedure, 2866
GENMOD procedure
syntax, 1921
GENMOD procedure, ASSESS statement, 1926
GENMOD PROCEDURE, BAYES statement, 1928
GENMOD procedure, BAYES statement
STATISTICS= option, 1936
THINNING= option, 1937
GENMOD procedure, BY statement, 1937
GENMOD procedure, CLASS statement, 1937
DESCENDING option, 1938
MISSING option, 1938
ORDER= option, 1938
PARAM= option, 1939
REF= option, 1939
TRUNCATE option, 1939
GENMOD procedure, CONTRAST statement, 1940
E option, 1942
SINGULAR= option, 1942
WALD option, 1942
GENMOD procedure, DEVIANCE statement, 1942, 1958

GENMOD procedure, ESTIMATE statement
ALPHA= option, 1943
E option, 1943
EXP option, 1943
SINGULAR= option, 1944
GENMOD procedure, FREQ statement, 1943, 1944
GENMOD procedure, FWDLINK statement, 1944, 1958
GENMOD procedure, INVLINK statement, 1944, 1958
GENMOD procedure, LSMEANS statement, 1945
ALPHA= option, 1945
CL option, 1945
CORR option, 1945
COV option, 1946
DIFF option, 1946
E option, 1946
SINGULAR= option, 1946
GENMOD procedure, MODEL statement, 1946
AGGREGATE= option, 1947
ALPHA= option, 1947
CICONV= option, 1947
CL option, 1947
CODING= option, 1947
CONVERGE= option, 1947
CONVH= option, 1948
CORRB option, 1948
COVB option, 1948
DIAGNOSTICS option, 1948
DIST= option, 1948
ERR= option, 1948
EXPECTED option, 1948
INFLUENCE option, 1948
INITIAL= option, 1949
INTERCEPT= option, 1949
ITPRINT option, 1949
LINK= option, 1949
LRCI option, 1950
MAXIT= option, 1950
NOINT option, 1950
NOSCALE option, 1950
OBSTATS option, 1950
OFFSET= option, 1952
PRED option, 1952
PREDICTED option, 1952
RESIDUALS option, 1952
SCALE= option, 1953
SCORING= option, 1953
SINGULAR= option, 1953
TYPE1 option, 1953
TYPE3 option, 1953
WALD option, 1954
WALDCI option, 1954
XVARS option, 1954
GENMOD procedure, OUTPUT statement, 1954
keyword= option, 1955
OUT= option, 1955
GENMOD procedure, PROC GENMOD statement, 1921
DATA= option, 1921
NAMELEN= option, 1922
ORDER= option, 1922
PLOTS= option, 1922
RORDER= option, 1925
GENMOD procedure, REPEATED statement, 1918, 1958
ALPHAINIT= option, 1959
CONVERGE= option, 1959
CORR= option, 1961
CORRB option, 1959
CORRW option, 1959
COVB option, 1959
ECORRB option, 1959
ECOVB option, 1959
INITIAL= option, 1959
INTERCEPT= option, 1959
LOGOR= option, 1960
MAXITER= option, 1960
MCORRB option, 1960
MCOVB option, 1960
MODELSE option, 1960
RUPDATE= option, 1960
SORTED option, 1961
SUBCLUSTER= option, 1961
SUBJECT= option, 1958
TYPE= option, 1961
V6CORR option, 1961
WITHIN= option, 1961
WITHINSUBJECT= option, 1961
YPAIR= option, 1962
ZDATA= option, 1962
ZROW= option, 1962
GENMOD procedure, SCWGT statement, 1962
GENMOD procedure, VARIANCE statement, 1962
GENMOD procedure, WEIGHT statement, 1962
GENMOD procedure, ZEROMODEL statement, 1963
LINK= option, 1963
GEOMETRICMEAN option
MODEL statement (TRANSREG), 7156
GI option
RANDOM statement (GLIMMIX), 2188
RANDOM statement (MIXED), 3951
GLIMMIX procedure, 2093
BY statement, 2121

CLASS statement, 2122
CONTRAST statement, 2123
COVTEST statement, 2127
EFFECT statement, 2134
ESTIMATE statement, 2134
FREQ statement, 2140
ID statement, 2140
LSMEANS statement, 2141
LSMESTIMATE statement, 2155
MODEL statement, 2161
NLOPTIONS statement, 391, 2175
OUTPUT statement, 2176
PARMS statement, 2181
PROC GLIMMIX statement, 2094
Programming statements, 2207
RANDOM statement, 2186
syntax, 2093
WEIGHT statement, 2206
GLIMMIX procedure, BY statement, 2121
GLIMMIX procedure, CLASS statement, 2122
TRUNCATE option, 2122
GLIMMIX procedure, CONTRAST statement, 2123
BYCAT option, 2125
BYCATEGORY option, 2125
CHISQ option, 2126
DF= option, 2126
E option, 2126
GROUP option, 2126
SINGULAR= option, 2126
SUBJECT option, 2126
GLIMMIX procedure, COVTEST statement, 2127
CL option, 2130
CLASSICAL option, 2132
ESTIMATES option, 2133
MAXITER= option, 2133
PARMS option, 2133
RESTART option, 2133
TOLERANCE= option, 2134
WALD option, 2134
WGHT= option, 2134
GLIMMIX procedure, DF= statement
CLASSICAL option, 2132
GLIMMIX procedure, EFFECT statement, 2134
BASIS option (spline), 384
collection effect, 378
DATABOUNDARY option (spline), 385
DEGREE option (polynomial), 380
DEGREE option (spline), 385
DETAILS option (multimember), 379
DETAILS option (polynomial), 381
DETAILS option (spline), 385
KNOTMAX option (spline), 385
KNOTMETHOD option (spline), 385
KNOTMIN option (spline), 386
LABELSTYLE option (polynomial), 381
MDEGREE option (polynomial), 381
multimember effect, 378
NOEFFECT option (multimember), 379
NOSEPARATE option (polynomial), 382
polynomial effect, 380
SEPARATE option (spline), 386
spline effect, 384
STANDARDIZE option (polynomial), 382
GLIMMIX procedure, ESTIMATE statement, 2134
ADJDFE= option, 2135
ADJUST= option, 2135
ALPHA= option, 2136
BYCAT option, 2136
BYCATEGORY option, 2136
CL option, 2136
DF= option, 2136
DIVISOR= option, 2136
E option, 2137
EXP option, 2137
GROUP option, 2137
ILINK option, 2137
LOWERTAILED option, 2138
SINGULAR= option, 2138
STEPDOWN option, 2138
SUBJECT option, 2139
UPPERTAILED option, 2140
GLIMMIX procedure, FREQ statement, 2140
GLIMMIX procedure, ID statement, 2140
GLIMMIX procedure, LSMEANS statement, 2141
ADJUST= option, 2144
ALPHA= option, 2145
AT MEANS option, 2145
AT option, 2145, 2146
BYLEVEL option, 2146
CL option, 2146
CORR option, 2146
COV option, 2146
DF= option, 2146
DIFF option, 2147
E option, 2147
ILINK option, 2147
LINES option, 2148
OBSMARGINS option, 2148
ODDS option, 2148
ODDSRATIO option, 2148
OM option, 2148
PDIFF option, 2147, 2149
PLOT option, 2149
PLOTS option, 2149

SIMPLEDIFF= option, 2152
SIMPLEDIFFTYPE option, 2153
SINGULAR= option, 2152
SLICE= option, 2152
SLICEDIFF= option, 2152
SLICEDIFFTYPE option, 2153
STEPDOWN option, 2154
GLIMMIX procedure, LSMESTIMATE statement, 2155
ADJUST= option, 2156
ALPHA= option, 2157
AT MEANS option, 2157
AT option, 2157
BYLEVEL option, 2157
CHISQ option, 2157
CL option, 2157
CORR option, 2157
COV option, 2157
DF= option, 2157
DIVISOR= option, 2158
E option, 2158
ELSM option, 2158
EXP option, 2158
FTEST option, 2158
ILINK option, 2158
LOWERTAILED option, 2159
OBSMARGINS option, 2159
OM option, 2159
SINGULAR= option, 2159
STEPDOWN option, 2159
UPPERTAILED option, 2161
GLIMMIX procedure, MODEL statement, 2161
CHISQ option, 2164
CL option, 2164
CORRB option, 2164
COVB option, 2164
COVBI option, 2165
DDF= option, 2165
DDFM= option, 2166
DESCENDING option, 2163
DF= option, 2165
DIST= option, 2168
DISTRIBUTION= option, 2168
E option, 2170
E1 option, 2171
E2 option, 2171
E3 option, 2171
ERROR= option, 2168
HTYPE= option, 2171
INTERCEPT option, 2171
LINK= option, 2171
LWEIGHT= option, 2172
NOCENTER option, 2172
NOINT option, 2173, 2261
ODDSRATIO option, 2173
OFFSET= option, 2174
ORDER= option, 2163
REFLINP= option, 2175
SOLUTION option, 2175, 2262
STDCOEF option, 2175
ZETA= option, 2175
GLIMMIX procedure, NLOPTIONS statement
ABSCONV option, 393
ABSFCONV option, 393
ABSGCONV option, 393
ABSGTOL option, 393
ABSTOL option, 393
ABSXCONV option, 393
ABSXTOL option, 393
ASINGULAR= option, 394
DAMPSTEP option, 394
FCONV option, 394
FCONV2 option, 395
FSIZE option, 395
FTOL option, 394
FTOL2 option, 395
GCONV option, 395
GCONV2 option, 396
GTOL option, 395
GTOL2 option, 396
HESCAL option, 396
HS option, 396
INHESS option, 396
INHESSIAN option, 396
INSTEP option, 397
LCDEACT= option, 397
LCEPSILON= option, 398
LCSINGULAR= option, 398
LINESEARCH option, 398
LIS option, 398
LSP option, 399
LSPRECISION option, 399
MAXFU option, 399
MAXFUNC option, 399
MAXIT option, 399
MAXITER option, 399
MAXSTEP option, 400
MAXTIME option, 400
MINIT option, 400
MINITER option, 400
MSINGULAR= option, 400
REST option, 401
RESTART option, 401
SINGULAR= option, 401
SOCKET option, 401
TECH option, 401
TECHNIQUE option, 401
UPD option, 402

UPDATE option, 402
VSINGULAR= option, 403
XCONV option, 403
XSIZE option, 403
XTOL option, 403
GLIMMIX procedure, OUTPUT statement, 2176
ALLSTATS option, 2179
ALPHA= option, 2180
DATA= option, 2177
DER option, 2180
DERIVATIVES option, 2180
keyword= option, 2177
NOMISS option, 2180
NOUNIQUE option, 2180
NOVAR option, 2180
OBSCAT option, 2180
OUT= option, 2177
SYMBOLS option, 2180
GLIMMIX procedure, PARMS statement, 2181
HOLD= option, 2181
LOWERB= option, 2182
NOBOUND option, 2183
NOITER option, 2183
PARMSDATA= option, 2184
PDATA= option, 2184
UPPERB= option, 2185
GLIMMIX procedure, PROC GLIMMIX statement, 2094
ABSPCONV option, 2095
ASYCORR option, 2095
ASYCOV option, 2095
CHOL option, 2096
CHOLESKY option, 2096
DATA= option, 2096
EMPIRICAL= option, 2096
EXPHESSIAN option, 2099
FDIGITS= option, 2099
GRADIENT option, 2099
HESSIAN option, 2099
IC= option, 2099
INFOCRIT= option, 2099
INITGLM option, 2101
INITITER option, 2101
ITDETAILS option, 2101
LIST option, 2101
MAXLMMUPDATE option, 2101
MAXOPT option, 2101
METHOD= option, 2102
NAMELEN= option, 2108
NOBOUND option, 2108
NOBSDETAIL option, 2108
NOCLPRINT option, 2108
NOFIT option, 2108
NOINITGLM option, 2109
NOITPRINT option, 2109
NOPROFILE option, 2109
NOREML option, 2109
ODDSRATIO option, 2109
ORDER= option, 2110
OUTDESIGN option, 2111
PCONV option, 2111
PLOT option, 2112
PLOTS option, 2112
PROFILE option, 2119
SCOREMOD option, 2119
SCORING= option, 2119
SINGCHOL= option, 2120
SINGULAR= option, 2120
STARTGLM option, 2120
SUBGRADIENT option, 2120
GLIMMIX procedure, Programming statements, 2207
ABORT statement, 2207
CALL statement, 2207
DELETE statement, 2207
DO statement, 2207
GOTO statement, 2207
IF statement, 2207
LINK statement, 2207
PUT statement, 2207
RETURN statement, 2207
SELECT statement, 2207
STOP statement, 2207
SUBSTR statement, 2207
WHEN statement, 2207
GLIMMIX procedure, RANDOM statement, 2186
ALPHA= option, 2187
CL option, 2187
G option, 2188
GC option, 2188
GCI option, 2188
GCOORD= option, 2188
GCORR option, 2188
GI option, 2188
GROUP= option, 2188
KNOTINFO option, 2189
KNOTMAX= option, 2189
KNOTMETHOD= option, 2189
KNOTMIN= option, 2191
LDATA= option, 2191
NOFULLZ option, 2192
RESIDUAL option, 2192
RSIDE option, 2192
SOLUTION option, 2192
SUBJECT= option, 2193
TYPE= option, 2193
V option, 2205

VC option, 2206
VCI option, 2206
VCORR option, 2206
VI option, 2206
GLIMMIX procedure, WEIGHT statement, 2206
GLM procedure
syntax, 2444
GLM procedure, ABSORB statement, 2452
GLM procedure, BY statement, 2452
GLM procedure, CLASS statement, 2453
TRUNCATE option, 2454
GLM procedure, CONTRAST statement, 2454
E option, 2454
E= option, 2455
ETYPE= option, 2455
INTERCEPT effect, 2454, 2456
SINGULAR= option, 2455
GLM procedure, ESTIMATE statement, 2456
DIVISOR= option, 2457
E option, 2457
SINGULAR= option, 2457
GLM procedure, FREQ statement, 2457
GLM procedure, ID statement, 2458
GLM procedure, LSMEANS statement, 2458
ADJUST= option, 2458
ALPHA= option, 2460
AT option, 2460, 2529
BYLEVEL option, 2530
BYLEVEL options, 2460
CL option, 2460
COV option, 2461
E option, 2461
ETYPE option, 2461
LINES option, 2461
NOPRINT option, 2461
OBSMARGINS option, 2461, 2530
OM option, 2461, 2530
OUT= option, 2462
PDIFF option, 2462
SINGULAR option, 2463
SLICE= option, 2463
STDERR option, 2463
TDIFF option, 2463
GLM procedure, MANOVA statement, 2464
ALL effect (H= option), 2464
CANONICAL option, 2465
E= option, 2464
ETYPE= option, 2465
H= option, 2464
HTYPE= option, 2465
INTERCEPT effect (H= option), 2464
M= option, 2464
MNAMES= option, 2465
MSTAT= option, 2466
ORTH option, 2466
PREFIX= option, 2465
PRINTE option, 2466
PRINTH option, 2466
SUMMARY option, 2466
GLM procedure, MEANS statement, 2468, 2527
ALPHA= option, 2469
BON option, 2470
CLDIFF option, 2470
CLM option, 2470
DEPONLY option, 2470
DUNCAN option, 2470
DUNNETT option, 2470
DUNNETTL option, 2470
DUNNETTU option, 2470
E= option, 2471
ETYPE= option, 2471
GABRIEL option, 2471
GT2 option, 2471
HOVTEST option, 2471, 2526
HTYPE= option, 2472
KRATIO= option, 2472
LINES option, 2472
LSD option, 2472
NOSORT option, 2472
REGWQ option, 2472
SCHEFFE option, 2472
SIDAK option, 2473
SMM option, 2473
SNK option, 2473
T option, 2473
TUKEY option, 2473
WALLER option, 2473
WELCH option, 2473
GLM procedure, MODEL statement, 2474
ALIASING option, 2475, 2610
ALPHA= option, 2475
CLI option, 2475
CLM option, 2475
CLPARM option, 2475
E option, 2475
E1 option, 2475
E2 option, 2475
E3 option, 2475
E4 option, 2475
EFFECTSIZE option, 2475
I option, 2476
INTERCEPT option, 2476
INVERSE option, 2476
NOINT option, 2476
NOUNI option, 2476
P option, 2476
SINGULAR= option, 2476
SOLUTION option, 2477

SS1 option, 2477
SS2 option, 2477
SS3 option, 2477
SS4 option, 2477
TOLERANCE option, 2477
XPX option, 2477
ZETA= option, 2477
GLM procedure, OUTPUT statement, 2477
ALPHA= option, 2479
COOKD keyword, 2478
COVRATIO keyword, 2478
DFFITS keyword, 2478
H keyword, 2478
keyword= option, 2478
LCL keyword, 2478
LCLM keyword, 2478
OUT= option, 2479
PREDICTED keyword, 2478
PRESS keyword, 2478
RESIDUAL keyword, 2478
RSTUDENT keyword, 2478
STDI keyword, 2479
STDP keyword, 2479
STDR keyword, 2479
STUDENT keyword, 2479
UCL keyword, 2479
UCLM keyword, 2479
GLM procedure, PROC GLM statement, 2446
ALPHA= option, 2446
DATA= option, 2446
MANOVA option, 2446
MULTIPASS option, 2447
NAMELEN= option, 2447
NOPRINT option, 2447
ORDER= option, 2447, 2511
OUTSTAT= option, 2448
PLOTS= option, 2448
GLM procedure, RANDOM statement, 2480
Q option, 2481, 2541
TEST option, 2481
GLM procedure, REPEATED statement, 2481
CANONICAL option, 2484
CONTRAST option, 2483, 2538
E= option, 2486
factor specification, 2482
H= option, 2486
HELMERT option, 2483, 2539
HTYPE= option, 2484
IDENTITY option, 2483, 2599
MEAN option, 2483, 2484, 2539
MSTAT= option, 2484
NOM option, 2484
NOU option, 2484
POLYNOMIAL option, 2483, 2538, 2591
PRINTE option, 2484, 2536
PRINTH option, 2484
PRINTM option, 2484
PRINTRV option, 2485
PROFILE option, 2483, 2539
SUMMARY option, 2485
GLM procedure, TEST statement, 2485
ETYPE= option, 2486
HTYPE= option, 2486
GLM procedure, WEIGHT statement, 2486
GLMMOD procedure
syntax, 2626
GLMMOD procedure, BY statement, 2628
GLMMOD procedure, CLASS statement, 2628
TRUNCATE option, 2628
GLMMOD procedure, FREQ statement, 2629
GLMMOD procedure, MODEL statement, 2629
NOINT option, 2629
GLMMOD procedure, PROC GLMMOD statement, 2626
DATA= option, 2626
NAMELEN= option, 2626
NOPRINT option, 2626
ORDER= option, 2626
OUTDESIGN= option, 2627, 2631
OUTPARM= option, 2627, 2630
PREFIX= option, 2627
ZEROBASED option, 2627
GLMMOD procedure, WEIGHT statement, 2629
GLMPOWER procedure
syntax, 2649
GLMPOWER procedure, BY statement, 2651
GLMPOWER procedure, CLASS statement, 2652
GLMPOWER procedure, CONTRAST statement, 2652
SINGULAR= option, 2653
GLMPOWER procedure, MODEL statement, 2653
GLMPOWER procedure, PLOT statement, 2654
DESCRIPTION= option, 2656
INTERPOL= option, 2654
KEY= option, 2654
MARKERS= option, 2655
MAX= option, 2655
MIN= option, 2655
NAME= option, 2657
NPOINTS= option, 2655
STEP= option, 2655
VARY option, 2655
X= option, 2656
XOPTS= option, 2656
Y= option, 2656
YOPTS= option, 2656

GLMPOWER procedure, POWER statement, 2657
ALPHA= option, 2658
CORRXY= option, 2658
DEPENDENT option, 2658
NCOVARIATES= option, 2658
NFRACTIONAL option, 2658
NTOTAL= option, 2659
OUTPUTORDER= option, 2659
POWER= option, 2659
PROPVARREDUCTION= option, 2659
STDDEV= option, 2659
GLMPOWER procedure, PROC GLMPOWER statement, 2650
DATA= option, 2650
ORDER= option, 2651
PLOTONLY= option, 2651
GLMPOWER procedure, WEIGHT statement, 2660
GLMSELECT procedure, 2694
syntax, 2694
GLMSELECT procedure, BY statement, 2702
GLMSELECT procedure, CLASS statement, 2702
CPREFIX= option, 2703
DELIMITER option, 2702
DESCENDING option, 2703
LPREFIX= option, 2703
MISSING option, 2703
ORDER= option, 2703
PARAM= option, 2704
REF= option, 2704
SHOWCODING option, 2702
SPLIT option, 2705
GLMSELECT procedure, DETAILS=STEPS(ALL) option
ALL, 2708
GLMSELECT procedure, DETAILS=STEPS(ANOVA) option
ANOVA, 2708
GLMSELECT procedure, DETAILS=STEPS(FITSTATISTICS) option
FITSTATISTICS, 2708
GLMSELECT procedure, DETAILS=STEPS(PARAMETERESTIMATES) option
PARAMETERESTIMATES, 2708
GLMSELECT procedure, EFFECT statement, 2706
BASIS option (spline), 384
collection effect, 378
DATABOUNDARY option (spline), 385
DEGREE option (polynomial), 380
DEGREE option (spline), 385
DETAILS option (multimember), 379
DETAILS option (polynomial), 381
DETAILS option (spline), 385
KNOTMAX option (spline), 385
KNOTMETHOD option (spline), 385
KNOTMIN option (spline), 386
LABELSTYLE option (polynomial), 381
MDEGREE option (polynomial), 381
multimember effect, 378
NOEFFECT option (multimember), 379
NOSEPARATE option (polynomial), 382
polynomial effect, 380
SEPARATE option (spline), 386
spline effect, 384
SPLIT option (spline), 387
STANDARDIZE option (polynomial), 382
GLMSELECT procedure, FREQ statement, 2706
GLMSELECT procedure, MODEL statement, 2706
CHOOSE= option, 2710
CVDETAILS option, 2707
CVMETHOD option, 2707
DETAILS option, 2708
DROP= option, 2711
HIERARCHY= option, 2708
INCLUDE= option, 2711
LSCOEFFS option, 2711
MAXSTEP option, 2712
NOINT option, 2709
ORDERSELECT option, 2709
SELECT= option, 2712
SELECTION= option, 2709
SHOWPVALUES option, 2714
SLENTRY= option, 2712
SLSTAY= option, 2712
STATS option, 2713
STB option, 2714
STOP= option, 2712
GLMSELECT procedure, OUTPUT statement, 2714
keyword option, 2715
OUT= option, 2715
PREDICTED keyword, 2715
RESIDUAL keyword, 2715
GLMSELECT procedure, PARTITION statement, 2715
FRACTION option, 2716
ROLEVAR= option, 2716
GLMSELECT procedure, PERFORMANCE statement, 2716
BUILDSSCP= option, 2717
CPUCOUNT= option, 2716
DETAILS option, 2717

NOTHREADS option, 2717
THREADS option, 2717
GLMSELECT procedure, PROC GLMSELECT statement, 2694
DATA= option, 2695
MAXMACRO= option, 2695
NAMELEN= option, 2696
NOPRINT option, 2696
OUTDESIGN= option, 2696
PLOT option, 2698
PLOTS option, 2698
SEED= option, 2701
TESTDATA= option, 2701
VALDATA= option, 2701
GLMSELECT procedure, SCORE statement, 2717
keyword option, 2718
OUT= option, 2718
PREDICTED keyword, 2718
RESIDUAL keyword, 2718
GLMSELECT procedure, STATS= option
ADJRSQ, 2713
AIC, 2713
AICC, 2713
ASE, 2713
BIC, 2713
CP, 2714
FVALUE, 2714
PRESS, 2714
RSQUARE, 2714
SBC, 2714
SL, 2714
GLMSELECT procedure, WEIGHT statement, 2718
GLS option
MODEL statement (CATMOD), 1124
GMSEP option
MODEL statement (REG), 5474
PLOT statement (REG), 5494
GOUT= option
MCMC statement (MI), 3759
PROC BOXPLOT statement, 759
PROC LIFEREG statement, 3006
PROC LIFETEST statement, 3115
PROC PROBIT statement, 5265
PROC REG statement, 5455
PROC TREE statement, 7363
GPATH= option
ODS HTML statement, 545
GRADIENT option
PROC GLIMMIX statement, 2099
GREENACRE option
PROC CORRESP statement, 1327
GRID statement
KRIGE2D procedure, 2942
SIM2D procedure, 6170
GRID= option
PLOT statement (BOXPLOT), 778
PRIOR statement (MIXED), 3947
GRIDDATA= option
GRID statement (KRIGE2D), 2942
GRID statement (SIM2D), 6171
GRIDL= option
BIVAR statement, 2891
UNIVAR statement, 2894
GRIDT= option
PRIOR statement (MIXED), 3947
GRIDU= option
BIVAR statement, 2891
UNIVAR statement, 2894
GROUP option
CONTRAST statement (GLIMMIX), 2126
CONTRAST statement (HPMIXED), 2818
CONTRAST statement (MIXED), 3919
ESTIMATE statement (GLIMMIX), 2137
ESTIMATE statement (HPMIXED), 2820
ESTIMATE statement (MIXED), 3921
GROUP statement
TCALIS procedure, 6764
GROUP= option
BASELINE statement (PHREG), 4545
MODEL statement, 6790
RANDOM statement (GLIMMIX), 2188
RANDOM statement (HPMIXED), 2830
RANDOM statement (MIXED), 3951
REPEATED statement (MIXED), 3955
STRATA statement (LIFETEST), 3129
GROUPLOSS= option
TWOSAMPLESURVIVAL statement (POWER), 4903
GROUPLOSSEXPHAZARDS= option
TWOSAMPLESURVIVAL statement (POWER), 4904
GROUPMEANS= option
ONEWAYANOVA statement (POWER), 4860
TWOSAMPLEMEANS statement (POWER), 4893
GROUPMEDLOSSTIMES= option
TWOSAMPLESURVIVAL statement (POWER), 4904
GROUPMEDSURVTIMES= option
TWOSAMPLESURVIVAL statement (POWER), 4904
GROUPNAMES= option
MODEL statement (REG), 5474
GROUPNS= option

ONEWAYANOVA statement (POWER), 4860
TWOSAMPLEFREQ statement (POWER), 4886
TWOSAMPLEMEANS statement (POWER), 4893
TWOSAMPLESURVIVAL statement (POWER), 4904
TWOSAMPLEWILCOXON statement (POWER), 4912
GROUPPROPORTIONS= option
TWOSAMPLEFREQ statement (POWER), 4886
GROUPS= option
MODEL statement, 6790
GROUPSTDDEVS= option
TWOSAMPLEMEANS statement (POWER), 4893
GROUPSURVEXPHAZARDS= option
TWOSAMPLESURVIVAL statement (POWER), 4904
GROUPSURVIVAL= option
TWOSAMPLESURVIVAL statement (POWER), 4905
GROUPWEIGHTS= option
ONEWAYANOVA statement (POWER), 4860
TWOSAMPLEFREQ statement (POWER), 4886
TWOSAMPLEMEANS statement (POWER), 4894
TWOSAMPLESURVIVAL statement (POWER), 4905
TWOSAMPLEWILCOXON statement (POWER), 4912
GT2 option
MEANS statement (GLM), 2471
GTOL option
NLOPTIONS statement (GLIMMIX), 395
NLOPTIONS statement (HPMIXED), 395
NLOPTIONS statement (TCALIS), 395
GTOL2 option
NLOPTIONS statement (GLIMMIX), 396
NLOPTIONS statement (HPMIXED), 396
NLOPTIONS statement (TCALIS), 396
GTOL2= option
NLOPTIONS statement (CALIS), 893
GTOL= option
NLOPTIONS statement (CALIS), 889, 893, 895
PROC CALIS statement, 855
PROC TCALIS statement, 6730

H keyword
OUTPUT statement (GLM), 2478
H option
PROC ROBUSTREG statement, 5665
H0= option
PROC TTEST statement, 7402
H= effects
REPEATED statement (ANOVA), 720
H= option
MANOVA statement (ANOVA), 706
MANOVA statement (GLM), 2464
OUTPUT statement (LOGISTIC), 3307
OUTPUT statement (NLIN), 4287
REPEATED statement (GLM), 2486
VARMETHOD=BRR (PROC SURVEYLOGISTIC statement), 6386
VARMETHOD=BRR (PROC SURVEYMEANS statement), 6480
VARMETHOD=BRR (PROC SURVEYREG statement), 6546
HADAMARD= option
VARMETHOD=BRR (PROC SURVEYFREQ statement), 6306
VARMETHOD=BRR (PROC SURVEYLOGISTIC statement), 6386
VARMETHOD=BRR (PROC SURVEYMEANS statement), 6480
VARMETHOD=BRR (PROC SURVEYREG statement), 6546
HALFWIDTH= option
ONESAMPLEFREQ statement (POWER), 4846
ONESAMPLEMEANS statement (POWER), 4854
PAIREDMEANS statement (POWER), 4873
TWOSAMPLEMEANS statement (POWER), 4894
HAXIS= option
PLOT statement (BOXPLOT), 778
PLOT statement (REG), 5494
PROC TREE statement, 7363
HAZARDRATIO statement
PHREG procedure, 4562
HAZARDRATIO= option
TWOSAMPLESURVIVAL statement (POWER), 4905
HC= option
PROC FASTCLUS statement, 1635
HCC option
MODEL statement (REG), 5475
HCCMETHOD= option
MODEL statement (REG), 5475
HEIGHT statement
TREE procedure, 7368

HEIGHT= option
ODS GRAPHICS statement, 530
PLOT statement (BOXPLOT), 779
PROC TREE statement, 7363
HELMERT keyword
REPEATED statement (ANOVA), 717
HELMERT option
REPEATED statement (GLM), 2483, 2539
HESCAL option
NLOPTIONS statement (GLIMMIX), 396
NLOPTIONS statement (HPMIXED), 396
NLOPTIONS statement (TCALIS), 396
HESCAL= option
NLOPTIONS statement (CALIS), 896
PROC NLMIXED statement, 4363
HESS option
PROC NLMIXED statement, 4364
HESSALG= option
PROC CALIS statement, 855
HESSIAN option
PROC GLIMMIX statement, 2099
HEYWOOD option
FACTOR statement (CALIS), 878
FACTOR statement (TCALIS), 6753
PROC FACTOR statement, 1552
HIERARCHY option
PROC VARCLUS statement, 7470
HIERARCHY= option
MODEL statement (GLMSELECT), 2708
MODEL statement (LOGISTIC), 3294
MODEL statement (PHREG), 4567
HISTORY option
MODEL statement (TRANSREG), 7162
HKPOWER= option
PROC FACTOR statement, 1552
HL option
EXACT statement (NPAR1WAY), 4463
OUTPUT statement (NPAR1WAY), 4465
PROC NPAR1WAY statement, 4458
HLM option
REPEATED statement (MIXED), 3955
HLPS option
REPEATED statement (MIXED), 3955
HM option
PROC MODECLUS statement, 4105
HMINOR= option
PLOT statement (BOXPLOT), 779
HOC option
PROC MULTTEST statement, 4190, 4215
HOFFSET= option
PLOT statement (BOXPLOT), 779
HOLD= option
PARMS statement (GLIMMIX), 2181
PARMS statement (HPMIXED), 2827
PARMS statement (MIXED), 3944
HOLM option
PROC MULTTEST statement, 4190, 4197
HOM option
PROC MULTTEST statement, 4190
HOMMEL option
PROC MULTTEST statement, 4215
HORDISPLAY= option
PROC TREE statement, 7364
HORIZONTAL option
PLOT statement (BOXPLOT), 779
PROC TREE statement, 7364
HOUGAARD option
PROC NLIN statement, 4280
HOVTEST option
MEANS statement (ANOVA), 712
MEANS statement (GLM), 2471, 2526
HP= option
PROC FASTCLUS statement, 1635
HPAGES= option
PROC TREE statement, 7364
HPLOTS= option
PLOT statement (REG), 5498
HPMIXED procedure
BY statement, 2814
CLASS statement, 2815
CONTRAST statement, 2816
ESTIMATE statement, 2818
ID statement, 2821
LSMEANS statement, 2821
MODEL statement, 2822
NLOPTIONS statement, 391, 2824
OUTPUT statement, 2824
PARMS statement, 2826
PROC HPMIXED statement, 2811
RANDOM statement, 2829
TEST statement, 2831
WEIGHT statement, 2832
HPMIXED procedure, BY statement, 2814
HPMIXED procedure, CLASS statement, 2815, 2837
TRUNCATE option, 2815
HPMIXED procedure, CONTRAST statement, 2816
CHISQ option, 2817
DF= option, 2817
E option, 2818
GROUP option, 2818
SINGULAR= option, 2818
SUBJECT= option, 2818
HPMIXED procedure, ESTIMATE statement, 2818
ALPHA= option, 2819
CL option, 2819

DF= option, 2819
DIVISOR= option, 2819
E option, 2820
GROUP option, 2820
SINGULAR= option, 2820
SUBJECT= option, 2820
HPMIXED procedure, ID statement, 2821
HPMIXED procedure, LSMEANS statement, 2821
ALPHA= option, 2821
CL option, 2821
CORR option, 2822
COV option, 2822
DF= option, 2822
E option, 2822
SINGULAR= option, 2822
HPMIXED procedure, MODEL statement, 2822
ALPHA= option, 2822
CL option, 2822
DDF= option, 2823
DDFM= option, 2823
NOINT option, 2823
SOLUTION option, 2823
ZETA= option, 2823
HPMIXED procedure, NLOPTIONS statement, 2824
ABSCONV option, 393
ABSFCONV option, 393
ABSGCONV option, 393
ABSGTOL option, 393
ABSTOL option, 393
ABSXCONV option, 393
ABSXTOL option, 393
ASINGULAR= option, 394
FCONV option, 394
FCONV2 option, 395
FSIZE option, 395
FTOL option, 394
FTOL2 option, 395
GCONV option, 395
GCONV2 option, 396
GTOL option, 395
GTOL2 option, 396
HESCAL option, 396
HS option, 396
INHESSIAN option, 396
LCDEACT= option, 397
LCEPSILON= option, 398
LCSINGULAR= option, 398
LINESEARCH option, 398
LIS option, 398
LSP option, 399
LSPRECISION option, 399
MAXFU option, 399
MAXFUNC option, 399
MAXIT option, 399
MAXITER option, 399
MAXSTEP option, 400
MAXTIME option, 400
MINIT option, 400
MINITER option, 400
MSINGULAR= option, 400
REST option, 401
RESTART option, 401
SINGULAR= option, 401
SOCKET option, 401
TECH option, 401
TECHNIQUE option, 401
UPD option, 402
XSIZE option, 403
XTOL option, 403
HPMIXED procedure, OUTPUT statement, 2824
ALLSTATS option, 2826
ALPHA= option, 2826
LCL= option, 2824
NOMISS option, 2826
NOUNIQUE option, 2826
NOVAR option, 2826
OUT= option, 2824
PEARSON= option, 2824
PREDICTED= option, 2824
RESIDUAL= option, 2824
STDERR= option, 2824
STUDENT= option, 2824
UCL= option, 2824
VARIANCE= option, 2824
HPMIXED procedure, PARMS statement, 2826
HOLD= option, 2827
LOWERB= option, 2827
NOITER option, 2827
PARMSDATA= option, 2828
PDATA= option, 2828
UPPERB= option, 2829
HPMIXED procedure, PROC HPMIXED statement, 2811
DATA= option, 2811
IC= option, 2811
INFOCRIT= option, 2811
ITDETAILS option, 2812
MAXCLPRINT= option, 2812
METHOD= option, 2812
MMEQ option, 2813
NAMELEN= option, 2813
NOCLPRINT option, 2813
NOFIT option, 2813
NOINFO option, 2813
NOITPRINT option, 2813
NOPRINT option, 2813

NOPROFILE option, 2813
ORDER= option, 2814
SINGCHOL= option, 2814
SINGRES= option, 2814
SINGULAR= option, 2814
HPMIXED procedure, RANDOM statement, 2829
ALPHA= option, 2830
CL option, 2830
GROUP= option, 2830
NOFULLZ option, 2830
SOLUTION option, 2830
SUBJECT= option, 2831
TYPE= option, 2831
HPMIXED procedure, TEST statement, 2831
CHISQ option, 2831
E option, 2831
E3 option, 2831
HTYPE= option, 2831
HPMIXED procedure, WEIGHT statement, 2832
HPMIXEDprocedure, NLOPTIONS statement
INSTEP option, 397
HPROB= option
PROC PROBIT statement, 5265
HREF= option
PLOT statement (BOXPLOT), 780
PLOT statement (REG), 5494
HREFLABELS= option
PLOT statement (BOXPLOT), 780
HREFLABPOS= option
PLOT statement (BOXPLOT), 780
HS option
NLOPTIONS statement (GLIMMIX), 396
NLOPTIONS statement (HPMIXED), 396
NLOPTIONS statement (TCALIS), 396
HSYMBOL= option
MCMC statement (MI), 3758, 3763
HTML= option
PLOT statement (BOXPLOT), 781
HTYPE= option
MANOVA statement (GLM), 2465
MEANS statement (GLM), 2472
MODEL statement (GLIMMIX), 2171
MODEL statement (MIXED), 3934
REPEATED statement (GLM), 2484
TEST statement (GLM), 2486
TEST statement (HPMIXED), 2831
HYBRID option
PROC CLUSTER statement, 1242
HYPERPRIOR statement
MCMC procedure, 3516

I option
MODEL statement (GLM), 2476
MODEL statement (REG), 5475
IAPPROXIMATIONS option
OUTPUT statement (TRANSREG), 7173
IC option
PROC MIXED statement, 3907
IC= option
PROC GLIMMIX statement, 2099
PROC HPMIXED statement, 2811
ID statement
BOXPLOT procedure, 760
CORRESP procedure, 1331
DISCRIM procedure, 1398
DISTANCE procedure, 1505
GLIMMIX procedure, 2140
GLM procedure, 2458
HPMIXED procedure, 2821
LIFETEST procedure, 3125
LOESS procedure, 3207
MDS procedure, 3721
MIXED procedure, 3922
MODECLUS procedure, 4109
NLIN procedure, 4285
NLMIXED procedure, 4375
PHREG procedure, 4564
PLS procedure, 4782
PRINCOMP procedure, 5157
PRINQUAL procedure, 5212
QUANTREG procedure, 5378
REG procedure, 5468
ROBUSTREG procedure, 5668
RSREG procedure, 5730
SURVEYSELECT procedure, 6639
TPSPLINE procedure, 7069
TRANSREG procedure, 7138
TREE procedure, 7368
ID= option
ODS PDF statement, 546
IDCOLOR= option
PLOT statement (BOXPLOT), 781
IDCTEXT= option
PLOT statement (BOXPLOT), 781
IDENTITY keyword
REPEATED statement (ANOVA), 717
IDENTITY option
REPEATED statement (GLM), 2483, 2599
IDENTITY transformation
MODEL statement (TRANSREG), 7147
TRANSFORM statement (PRINQUAL), 5216
IDFONT= option
PLOT statement (BOXPLOT), 781
IDHEIGHT= option
PLOT statement (BOXPLOT), 781
IDSYMBOL= option

PLOT statement (BOXPLOT), 781
IFACTOR= option
PRIOR statement (MIXED), 3947
IGNOREPERIOD option
PROC TTEST statement, 7410
ILINK option
ESTIMATE statement (GLIMMIX), 2137
LSMEANS statement (GLIMMIX), 2147
LSMESTIMATE statement (GLIMMIX), 2158
IMAGE_DPI= option
ODS destination statement, 532
IMAGEFMT= option
ODS GRAPHICS statement, 530
IMAGEMAP= option
ODS GRAPHICS statement, 530
IMAGENAME= option
ODS GRAPHICS statement, 530
IMPUTE option
PROC FASTCLUS statement, 1636
IMPUTE= option
MCMC statement (MI), 3759
IN option
PLOT statement (REG), 5495
INAV= option
PROC MDS statement, 3715
INBREED procedure
syntax, 2863
INBREED procedure, BY statement, 2865
INBREED procedure, CLASS statement, 2866
INBREED procedure, GENDER statement, 2866
INBREED procedure, MATINGS statement, 2867
INBREED procedure, PROC INBREED statement, 2864
AVERAGE option, 2864
COVAR option, 2864
DATA= option, 2864
IND option, 2864
INDL option, 2864
INIT= option, 2865
MATRIX option, 2865
MATRIXL option, 2865
NOPRINT option, 2865
OUTCOV= option, 2865
INBREED procedure, VAR statement, 2867
INC= option
PROC TREE statement, 7364
INCLUDE= option
MODEL statement (GLMSELECT), 2711
MODEL statement (LOGISTIC), 3295
MODEL statement (PHREG), 4568
MODEL statement (REG), 5475
PROC STEPDISC statement, 6275
IND option
PROC INBREED statement, 2864
INDIVIDUAL option
MODEL statement (TRANSREG), 7162
INDL option
PROC INBREED statement, 2864
INEST= option
MCMC statement (MI), 3759
PROC CALIS statement, 856
PROC LIFEREG statement, 3006
PROC LOGISTIC statement, 3270
PROC PHREG statement, 4537
PROC PROBIT statement, 5265
PROC QUANTREG statement, 5374
PROC ROBUSTREG statement, 5661
PROC SURVEYLOGISTIC statement, 6383
PROC TCALIS statement, 6730
INF= option
PROC MCMC statement, 3500
INFLUENCE option
MODEL statement (GENMOD), 1948
MODEL statement (LOGISTIC), 3295
MODEL statement (MIXED), 3934
MODEL statement (REG), 5475
INFO option
PROC MIXED statement, 3908
STRATA statement (LOGISTIC), 3315
INFO= option
DESIGN statement (SEQDESIGN), 5805
INFOCRIT= option
PROC GLIMMIX statement, 2099
PROC HPMIXED statement, 2811
INHESS option
NLOPTIONS statement (GLIMMIX), 396
NLOPTIONS statement (HPMIXED), 396
NLOPTIONS statement (TCALIS), 396
INHESSIAN option
NLOPTIONS statement (GLIMMIX), 396
NLOPTIONS statement (HPMIXED), 396
NLOPTIONS statement (TCLAIS), 396
PROC NLMIXED statement, 4364
INIT= option
PROC INBREED statement, 2865
PROC MCMC statement, 3500
INITEST option
PROC ROBUSTREG statement, 5667
INITGLM option
PROC GLIMMIX statement, 2101
INITH option
PROC ROBUSTREG statement, 5667
INITIAL option
EM statement (MI), 3755
INITIAL= option
BAYES statement(PHREG), 4549

MCMC statement (MI), 3759
MODEL statement (GENMOD), 1949
MODEL statement (LIFEREG), 3024
PROC ACECLUS statement, 675
PROC DISTANCE statement, 1493
PROC MDS statement, 3716
PROC STDIZE statement, 6222
PROC VARCLUS statement, 7470
REPEATED statement (GENMOD), 1959
INITITER option
PROC GLIMMIX statement, 2101
INITITER= option
PROC PRINQUAL statement, 5205
INMODEL= option
PROC LOGISTIC statement, 3271
PROC TCALIS statement, 6730
INPVALUES= option
PROC MULTTEST statement, 4191
INRAM= option
PROC CALIS statement, 856
PROC TCALIS statement, 6730
INSET statement
BOXPLOT procedure, 761
LIFEREG procedure, 3018
PROBIT procedure, 5277
INSETGROUP statement
BOXPLOT procedure, 764
INSTAT= option
PROC FASTCLUS statement, 1636
INSTEP option
NLOPTIONS statement (GLIMMIX), 397
NLOPTIONS statement (HPMIXED), 397
NLOPTIONS statement (TCALIS), 397
INSTEP= option, 4398
NLOPTIONS statement (CALIS), 889
PROC CALIS statement, 856
PROC NLMIXED statement, 4364
PROC TCALIS statement, 6730
INT option
PROC CANCORR statement, 1039
INTERCEPT effect
CONTRAST statement (GLM), 2454, 2456
MANOVA statement (ANOVA), 706
MANOVA statement, H= option (GLM), 2464
INTERCEPT option
MODEL statement (ANOVA), 715
MODEL statement (GLIMMIX), 2171
MODEL statement (GLM), 2476
MODEL statement (MIXED), 3939
MODEL statement (PLS), 4783
INTERCEPT= option
LOGISTIC statement (POWER), 4830
MODEL statement (GENMOD), 1949
MODEL statement (LIFEREG), 3024
REPEATED statement (GENMOD), 1959
INTERP= option
MODEL statement (LOESS), 3210
INTERPOL= option
PLOT statement (GLMPOWER), 2654
PLOT statement (POWER), 4880
INTERVAL= option
PLOT statement (BOXPLOT), 781
INTERVALS= option
PROC LIFETEST statement, 3115
INTSTART= option
BOXPLOT procedure, 782
INVAR statement, MDS procedure, 3722
INVAR= option
PROC CALIS statement, 856
PROC TCALIS statement, 6730
INVERSE option
MODEL statement (GLM), 2476
MODEL statement (SURVEYREG), 6555
INVERSECL option
PROC PROBIT statement, 5265
INVLINK statement, GENMOD procedure, 1944, 1958
INWGT= option
PROC CALIS statement, 856
PROC TCALIS statement, 6731
INWGTINV option
PROC TCALIS statement, 6731
IPLOTS option
MODEL statement (LOGISTIC), 3296
IPPPLOT statement
options summarized by function, 5280
PROBIT procedure, 5279
IREPLACE option
OUTPUT statement (TRANSREG), 7174
IRLS option
PROC FASTCLUS statement, 1636
ITDETAILS option
PROC GLIMMIX statement, 2101
PROC HPMIXED statement, 2812
PROC MIXED statement, 3908
PROC NLMIXED statement, 4364
ITER= modifier
INFLUENCE option, MODEL statement (MIXED), 3935
ITER= option
PROC MDS statement, 3716
ITERATIONS= option
MODEL statement (LOESS), 3210
ITPRINT option
EM statement (MI), 3755
MCMC statement (MI), 3760
MODEL statement, 5379, 5669

MODEL statement (CATMOD), 1121
MODEL statement (GAM), 1845
MODEL statement (GENMOD), 1949
MODEL statement (LIFEREG), 3024
MODEL statement (LOGISTIC), 3296
MODEL statement (PHREG), 4568
MODEL statement (SURVEYLOGISTIC), 6399
PROC ROBUSTREG statement, 5661

JEFFREYS option
PRIOR statement (MIXED), 3947
JKCOEFS= option
REPWEIGHTS statement (SURVEYFREQ), 6311
REPWEIGHTS statement (SURVEYLOGISTIC), 6406
REPWEIGHTS statement (SURVEYMEANS), 6488
REPWEIGHTS statement (SURVEYREG), 6558
JOIN= option
PROC MODECLUS statement, 4105
JOINCHAR= option
PROC TREE statement, 7364
JOINT function
RESPONSE statement (CATMOD), 1131
JOINT option
EXACT statement (LOGISTIC), 3286
JOINTMODEL option
PROC MCMC statement, 3501
JOINTONLY option
EXACT statement (LOGISTIC), 3286
JP option
MODEL statement (REG), 5475
PLOT statement (REG), 5495
JT option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1696
TABLES statement (FREQ), 1708
JTPROBS option
PROC SURVEYSELECT statement, 6627

K0 option
PROC ROBUSTREG statement, 5666, 5667
K= option
PROC CLUSTER statement, 1242
PROC DISCRIM statement, 1392
PROC MODECLUS statement, 4105
KAPPA option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1696
TEST statement (FREQ), 1723
KAPPA= option
PROC QUANTREG statement, 5373
KDE, 2885
KDE procedure, 2885
syntax, 2889
KDE procedure, BIVAR statement, 2890
BIVSTATS option, 2891
BWM= option, 2891
GRIDL= option, 2891
GRIDU= option, 2891
LEVELS= option, 2891
NGRID= option, 2891
NOPRINT option, 2891
OUT= option, 2892
PLOTS= option, 2892, 2905
UNISTATS option, 2893
KDE procedure, BY statement, 2896
KDE procedure, FREQ statement, 2896
KDE procedure, PROC KDE statement, 2889
DATA= option, 2889
KDE procedure, UNIVAR statement, 2893
BWM= option, 2893
GRIDL= option, 2894
GRIDU= option, 2894
METHOD= option, 2894
NGRID= option, 2894
NOPRINT option, 2894
OUT= option, 2894
PLOTS= option, 2894, 2905
UNISTATS option, 2895
KDE procedure, WEIGHT statement, 2897
KEEP= modifier
INFLUENCE option, MODEL statement (MIXED), 3936
KEEPLEN option
PROC STDIZE statement, 6223
KENTB option
OUTPUT statement (FREQ), 1696
TEST statement (FREQ), 1723
KERNEL= option
PROC DISCRIM statement, 1392
KEY= option
PLOT statement (GLMPOWER), 2654
PLOT statement (POWER), 4880
keyword option
OUTPUT statement (GLMSELECT), 2715
SCORE statement (GLMSELECT), 2718
keyword= option
BASELINE statement (PHREG), 4541
OUTPUT statement (GENMOD), 1955
OUTPUT statement (GLIMMIX), 2177
OUTPUT statement (GLM), 2478
OUTPUT statement (LIFEREG), 3026
OUTPUT statement (PHREG), 4573
OUTPUT statement (QUANTREG), 5381

OUTPUT statement (REG), 5483
OUTPUT statement (ROBUSTREG), 5670
OUTPUT statement (SURVEYREG), 6557
KLOTZ option
EXACT statement (NPAR1WAY), 4463
OUTPUT statement (NPAR1WAY), 4465
PROC NPAR1WAY statement, 4458
KNOTINFO option
RANDOM statement (GLIMMIX), 2189
KNOTMAX option
EFFECT statement, spline (GLIMMIX), 385
EFFECT statement, spline (GLMSELECT), 385
KNOTMAX= option
RANDOM statement (GLIMMIX), 2189
KNOTMETHOD option
EFFECT statement, spline (GLIMMIX), 385
EFFECT statement, spline (GLMSELECT), 385
KNOTMETHOD= option
RANDOM statement (GLIMMIX), 2189
KNOTMIN option
EFFECT statement, spline (GLIMMIX), 386
EFFECT statement, spline (GLMSELECT), 386
KNOTMIN= option
RANDOM statement (GLIMMIX), 2191
KNOTS= option
MODEL statement (TRANSREG), 7151
TRANSFORM statement (PRINQUAL), 5219
KNOTTYPE= suboption
RANDOM statement (GLIMMIX), 2190
KPROP= option
PROC DISCRIM statement, 1392
KRATIO= option
MEANS statement (ANOVA), 713
MEANS statement (GLM), 2472
KRIGE2D procedure, 2930
syntax, 2936
KRIGE2D procedure, BY statement, 2940
KRIGE2D procedure, COORDINATES statement, 2941
XCCORD= option, 2941
YCCORD= option, 2941
KRIGE2D procedure, GRID statement, 2942
GRIDDATA= option, 2942
X= option, 2942
XCOORD= option, 2942
Y= option, 2942
YCOORD= option, 2942
KRIGE2D procedure, MODEL statement, 2944
ANGLE= option, 2945
FORM= option, 2945
MDATA= option, 2945
NUGGET= option, 2946
RANGE= option, 2946
RATIO= option, 2946
SCALE= option, 2947
SINGULAR= option, 2947
KRIGE2D procedure, PREDICT statement, 2943
MAXPOINTS= option, 2943
MINPOINTS= option, 2943
NODECREMENT option, 2943
NOINCREMENT option, 2943
NUMPOINTS= option, 2944
RADIUS= option, 2944
VAR= option, 2944
KRIGE2D procedure, PROC KRIGE2D statement, 2937
DATA= option, 2937
NOPRINT option, 2937
OUTEST= option, 2937
OUTNBHD= option, 2937
PLOTS option, 2937
PLOTS(ONLY) option, 2938
PLOTS=ALL option, 2938
PLOTS=EQUATE option, 2938
PLOTS=NONE option, 2938
PLOTS=OBSERVATIONS option, 2938
PLOTS=PREDICTION option, 2939
SINGULARMSG= option, 2940
KS option
EXACT statement (NPAR1WAY), 4463
KURTOSIS option
PROC CALIS statement, 857
PROC TCALIS statement, 6731

L95 option
MODEL statement (RSREG), 5732
L95= option
OUTPUT statement (NLIN), 4287
L95M option
MODEL statement (RSREG), 5732
L95M= option
OUTPUT statement (NLIN), 4287
L= option
PROC FASTCLUS statement, 1636
LABEL= option
GROUP statement, 6765
MODEL statement, 6790
LABELANGLE= option
BOXPLOT procedure, 782
LABELMAX= option
ODS GRAPHICS statement, 530

LABELSTYLE option
EFFECT statement, polynomial (GLIMMIX), 381
EFFECT statement, polynomial (GLMSELECT), 381
LACKFIT option
MODEL statement (LOGISTIC), 3296
MODEL statement (REG), 5475
MODEL statement (RSREG), 5732
PROC PROBIT statement, 5265
LAGDISTANCE= option
COMPUTE statement (VARIOGRAM), 7536
LAGTOLERANCE= option
COMPUTE statement (VARIOGRAM), 7537
LAMBDA0= option
MODEL statement (TPSPLINE), 7070
LAMBDA= option
MODEL statement (TPSPLINE), 7071
MODEL statement (TRANSREG), 7152, 7156
TRANSFORM statement (MI), 3768
LAMCR option
OUTPUT statement (FREQ), 1696
LAMDAS option
OUTPUT statement (FREQ), 1696
LAMRC option
OUTPUT statement (FREQ), 1696
LANNOTATE= option
PROC LIFETEST statement, 3116
LATTICE procedure, 2982
syntax, 2982
LATTICE procedure, BY statement, 2983
LATTICE procedure, PROC LATTICE statement, 2982
COV option, 2982
COVARIANCE option, 2982
DATA= option, 2982
LATTICE procedure, VAR statement, 2983
LBOXES= option
PLOT statement (BOXPLOT), 782
LCDEACT= option
NLOPTIONS statement (CALIS), 897
NLOPTIONS statement (GLIMMIX), 397
NLOPTIONS statement (HPMIXED), 397
NLOPTIONS statement (TCALIS), 397
PROC NLMIXED statement, 4365
LCEPSILON= option
NLOPTIONS statement (CALIS), 897
NLOPTIONS statement (GLIMMIX), 398
NLOPTIONS statement (HPMIXED), 398
NLOPTIONS statement (TCALIS), 398
PROC NLMIXED statement, 4365
LCL keyword
OUTPUT statement (GLM), 2478
LCL= option
OUTPUT statement (HPMIXED), 2824
OUTPUT statement (NLIN), 4287
LCLM keyword
OUTPUT statement (GLM), 2478
OUTPUT statement (SURVEYREG), 6557
LCLM= option
OUTPUT statement (NLIN), 4287
LCOMPONENTS option
MODEL statement (MIXED), 3939
LCONF= option
MCMC statement (MI), 3758
LCONNECT= option
MCMC statement (MI), 3763
LCSINGULAR= option
NLOPTIONS statement (CALIS), 897
NLOPTIONS statement (GLIMMIX), 398
NLOPTIONS statement (HPMIXED), 398
NLOPTIONS statement (TCALIS), 398
PROC NLMIXED statement, 4365
LDATA= option
RANDOM statement (GLIMMIX), 2191
RANDOM statement (MIXED), 3951
REPEATED statement (MIXED), 3956
LEAFCHAR= option
PROC TREE statement, 7364
LEAST= option
PROC FASTCLUS statement, 1636
LEGEND= option
PLOT statement (REG), 5495
LENDGRID= option
PLOT statement (BOXPLOT), 783
LEVEL= option
PROC MDS statement, 3716
PROC TREE statement, 7365
LEVELS= option
BIVAR statement, 2891
LEVERAGE keyword
OUTPUT statement (QUANTREG), 5381
OUTPUT statement (ROBUSTREG), 5670
LEVERAGE option
MODEL statement, 5379, 5669
LEVERAGE= option
OUTPUT statement (TRANSREG), 7174
LGOR option
OUTPUT statement (FREQ), 1696
LGRID= option
PLOT statement (BOXPLOT), 783
LGRRC1 option
OUTPUT statement (FREQ), 1696
LGRRC2 option
OUTPUT statement (FREQ), 1696

LHREF= option
PLOT statement (BOXPLOT), 783
PLOT statement (REG), 5495
LIFEREG procedure
syntax, 3005
LIFEREG PROCEDURE, BAYES statement, 3007
LIFEREG procedure, BAYES statement
STATISTICS= option, 3015
THINNING= option, 3015
LIFEREG procedure, BY statement, 3017
LIFEREG procedure, CLASS statement, 3017
LIFEREG procedure, INSET statement, 3018
keywords, 3018
LIFEREG procedure, MODEL statement, 3020
ALPHA= option, 3022
CONVERGE= option, 3022
CONVG= option, 3022
CORRB option, 3022
COVB option, 3022
DISTRIBUTION= option, 3023
INITIAL= option, 3024
INTERCEPT= option, 3024
ITPRINT option, 3024
MAXITER= option, 3024
NOINT option, 3024
NOLOG option, 3024
NOSCALE option, 3025
NOSHAPE1 option, 3025
OFFSET= option, 3025
SCALE= option, 3025
SHAPE1= option, 3025
SINGULAR= option, 3025
LIFEREG procedure, OUTPUT statement, 3025
CDF keyword, 3026
CENSORED keyword, 3026
CONTROL keyword, 3026
keyword= option, 3026
OUT= option, 3026
PREDICTED keyword, 3026
QUANTILES keyword, 3026
STD_ERR keyword, 3027
XBETA keyword, 3027
LIFEREG procedure, PPLOT statement
ANNOTATE= option, 3028
CAXIS= option, 3028
CCENSOR option, 3028
CENBIN, 3028
CENCOLOR option, 3028
CENSYMBOL option, 3028
CFIT= option, 3028
CFRAME= option, 3028
CGRID= option, 3028
CHREF= option, 3028
CTEXT= option, 3028
CVREF= option, 3029
DESCRIPTION= option, 3029
FONT= option, 3029
HCL, 3029, 3033
HEIGHT= option, 3029
HLOWER= option, 3029, 3033
HOFFSET= option, 3029
HREF= option, 3029, 3033
HREFLABELS= option, 3029, 3034
HREFLABPOS= option, 3030
HUPPER= option, 3029, 3033
INBORDER option, 3030
INTERTILE option, 3030
ITPRINTEM option, 3030, 3034
JITTER option, 3030
LFIT option, 3030
LGRID option, 3030
LHREF= option, 3030
LVREF= option, 3030
MAXITEM= option, 3030, 3034
NAME= option, 3030
NOCENPLOT option, 3031, 3034
NOCONF option, 3031, 3034
NODATA option, 3031, 3034
NOFIT option, 3031, 3034
NOFRAME option, 3031, 3034
NOGRID option, 3031, 3034
NOHLABEL option, 3031
NOHTICK option, 3031
NOPOLISH option, 3031, 3034
NOVLABEL option, 3031
NOVTICK option, 3031
NPINTERVALS option, 3031, 3034
PCTLIST option, 3031, 3034
PLOWER= option, 3031, 3034
PPOS option, 3032, 3035
PPOUT option, 3032, 3035
PRINTPROBS option, 3031, 3035
PROBLIST option, 3032, 3035
PUPPER= option, 3032, 3035
ROTATE option, 3032, 3035
SQUARE option, 3032, 3035
TOLLIKE option, 3032, 3035
TOLPROB option, 3032, 3035
VAXISLABEL= option, 3032
VREF= option, 3032, 3035
VREFLABELS= option, 3033, 3036
VREFLABPOS= option, 3033
WAXIS= option, 3033
WFIT= option, 3033
WGRID= option, 3033
WREFL= option, 3033

LIFEREG procedure, PROBPLOT statement, 3027
LIFEREG procedure, PROC LIFEREG statement, 3006
  COVOUT option, 3006
  DATA= option, 3006
  GOUT= option, 3006
  INEST= option, 3006
  NAMELEN= option, 3006
  NOPRINT option, 3006
  ORDER= option, 3006
  OUTEST= option, 3007
  XDATA= option, 3007
LIFEREG procedure, WEIGHT statement, 3036
LIFETEST procedure, 3099
  BY statement, 3124
  FREQ statement, 3125
  ID statement, 3125
  PROC LIFETEST statement, 3111
  STRATA statement, 3125
  syntax, 3110
  TEST statement, 3130
  TIME statement, 3131
LIFETEST procedure, BY statement, 3124
LIFETEST procedure, FREQ statement, 3125
LIFETEST procedure, ID statement, 3125
LIFETEST procedure, PROC LIFETEST statement, 3111
  ALPHA= option, 3113
  ALPHAQT= option, 3113
  ANNOTATE= option, 3113
  BANDMAXTIME= option, 3113
  BANDMINTIME= option, 3113
  CENSOREDSYMBOL= option, 3113
  CONFBAND= option, 3114
  CONFTYPE= option, 3114
  DATA= option, 3114
  DESCRIPTION= option, 3115
  EVENTSYMBOL= option, 3115
  FORMCHAR= option, 3115
  GOUT= option, 3115
  INTERVALS= option, 3115
  LANNOTATE= option, 3116
  LINEPRINTER option, 3116
  MAXTIME= option, 3116
  METHOD= option, 3116
  MISSING option, 3117
  NELSON option, 3117
  NINTERVAL= option, 3117
  NOCENSPLOT option, 3117
  NOPRINT option, 3117
  NOTABLE option, 3117
  OUTSURV= option, 3117
  OUTTEST= option, 3117
  PLOTS= option, 3118, 3122
  REDUCEOUT option, 3123
  SINGULAR= option, 3123
  STDERR option, 3123
  TIMELIM= option, 3123
  TIMELIST= option, 3124
  WIDTH= option, 3124
LIFETEST procedure, STRATA statement, 3125
  ADJUST= option, 3127
  DIFF= option, 3128
  GROUP= option, 3129
  MISSING option, 3129
  NODETAIL option, 3129
  NOTEST option, 3129
  TEST= option, 3129
  TREND option, 3129
LIFETEST procedure, TEST statement, 3130
LIFETEST procedure, TIME statement, 3131
LILPREFIX= option
  OUTPUT statement (TRANSREG), 7174
LINCON statement, CALIS procedure, 880
LINCON statement, TCALIS procedure, 6766
LINEAR transformation
  MODEL statement (TRANSREG), 7145
  TRANSFORM statement (PRINQUAL), 5215
LINEPRINTER option
  PROC LIFETEST statement, 3116
  PROC REG statement, 5455
  PROC TREE statement, 7365
LINEQS statement, CALIS procedure, 881
LINEQS statement, TCALIS procedure, 6766
LINES option
  LSMEANS statement (GLIMMIX), 2148
  LSMEANS statement (GLM), 2461
  MEANS statement (ANOVA), 713
  MEANS statement (GLM), 2472
LINES= option
  PROC TREE statement, 7365
LINESEARCH option
  NLOPTIONS statement (GLIMMIX), 398
  NLOPTIONS statement (HPMIXED), 398
  NLOPTIONS statement (TCALIS), 398
LINESEARCH= option, 4397
  NLOPTIONS statement (CALIS), 889
  PROC CALIS statement, 857
  PROC NLMIXED statement, 4365
  PROC TCALIS statement, 6731
LINK= option
  MODEL statement (GENMOD), 1949
  MODEL statement (GLIMMIX), 2171
  MODEL statement (LOGISTIC), 3296
  MODEL statement (SURVEYLOGISTIC), 6399

ROC statement (LOGISTIC), 3310
ZEROMODEL statement (GENMOD), 1963
LIS option
NLOPTIONS statement (GLIMMIX), 398
NLOPTIONS statement (HPMIXED), 398
NLOPTIONS statement (TCALIS), 398
LISMOD statement, TCALIS procedure, 6772
LIST option
PROC DISCRIM statement, 1393
PROC FASTCLUS statement, 1637
PROC GLIMMIX statement, 2101
PROC MCMC statement, 3501
PROC MODECLUS statement, 4105
PROC NLIN statement, 4281
PROC NLMIXED statement, 4366
PROC TREE statement, 7365
STRATA statement (SURVEYFREQ), 6312
STRATA statement (SURVEYLOGISTIC), 6407
STRATA statement (SURVEYMEANS), 6490
STRATA statement (SURVEYREG), 6560
TABLES statement (FREQ), 1709
LISTALL option
PROC NLIN statement, 4281
LISTCODE option
PROC MCMC statement, 3501
PROC NLIN statement, 4281
PROC NLMIXED statement, 4366
LISTDEP option
PROC NLIN statement, 4281
PROC NLMIXED statement, 4366
LISTDER option
PROC NLIN statement, 4281
PROC NLMIXED statement, 4366
LISTERR option
PROC DISCRIM statement, 1393
LIUPREFIX= option
OUTPUT statement (TRANSREG), 7174
LLINE= option
PLOT statement (REG), 5495
LMLPREFIX= option
OUTPUT statement (TRANSREG), 7174
lmmat option
LMTESTS statement, 6777
LMTESTS statement, TCALIS procedure, 6776
LOCAL option
PROC MODECLUS statement, 4105
LOCAL= option
REPEATED statement (MIXED), 3956
LOCALW option
REPEATED statement (MIXED), 3957
LOESS procedure, BY statement, 3207
LOESS procedure, ID statement, 3207
LOESS procedure, MODEL statement, 3208
ALL option, 3208
ALPHA= option, 3208
BUCKET= option, 3209
CLM= option, 3209
DEGREE= option, 3209
DETAILS option, 3209
DFMETHOD= option, 3209
DFMETHOD=APPROX(Cutoff=) option, 3210
DFMETHOD=APPROX(Quantile=) option, 3210
DIRECT option, 3210
DROPSQUARE= option, 3210
INTERP= option, 3210
ITERATIONS= option, 3210
RESIDUAL option, 3210
SCALE= option, 3210
SCALEDINDEP option, 3211
SELECT= option, 3211
SMOOTH= option, 3213
STD option, 3213
T option, 3213
TRACEL option, 3213
LOESS procedure, PROC LOESS statement, 3202
DATA= option, 3202
PLOT option, 3202
PLOTS option, 3202
LOESS procedure, SCORE statement, 3213
CLM option, 3214
PRINT option, 3214
SCALEDINDEP option, 3214
STEPS option, 3214
LOESS procedure, WEIGHT statement, 3214
LOG option
MCMC statement (MI), 3758, 3763
PROC PROBIT statement, 5265
LOG transformation
MODEL statement (TRANSREG), 7144
TRANSFORM statement (MI), 3767
TRANSFORM statement (PRINQUAL), 5214
LOG10 option
PROC PROBIT statement, 5266
LOGDETH option
PARMS statement (MIXED), 3944
LOGISTIC option
MONOTONE statement (MI), 3765
LOGISTIC procedure, 3266
syntax, 3266
LOGISTIC procedure, BY statement, 3279
LOGISTIC procedure, CLASS statement, 3280

CPREFIX= option, 3280
DESCENDING option, 3280
LPREFIX= option, 3280
MISSING option, 3280
ORDER= option, 3280
PARAM= option, 3281
REF= option, 3281
TRUNCATE option, 3281
LOGISTIC procedure, CONTRAST statement, 3282
ALPHA= option, 3283
E option, 3283
ESTIMATE= option, 3283
SINGULAR= option, 3283
LOGISTIC procedure, EXACT statement, 3285
ALPHA= option, 3285
CLTYPE= option, 3286
ESTIMATE option, 3285
JOINT option, 3286
JOINTONLY option, 3286
MIDPFACTOR= option, 3286
ONESIDED option, 3286
OUTDIST= option, 3286
LOGISTIC procedure, FREQ statement, 3287
LOGISTIC procedure, MODEL statement, 3288
ABSFCONV option, 3291
AGGREGATE= option, 3292
ALPHA= option, 3292
BEST= option, 3292
BINWIDTH= option, 3292
CL option, 3302
CLODDS= option, 3292
CLPARM= option, 3293
CORRB option, 3293
COVB option, 3293
CTABLE option, 3293
DESCENDING option, 3288
DETAILS option, 3293
EVENT= option, 3288
EXPEST option, 3293
FAST option, 3294
FCONV= option, 3294
FIRTH option, 3294
GCONV= option, 3294
HIERARCHY= option, 3294
INCLUDE= option, 3295
INFLUENCE option, 3295
IPLOTS option, 3296
ITPRINT option, 3296
LACKFIT option, 3296
LINK= option, 3296
MAXFUNCTION= option, 3297
MAXITER= option, 3297
MAXSTEP= option, 3297
NOCHECK option, 3297
NODESIGNPRINT= option, 3297
NODUMMYPRINT= option, 3297
NOFIT option, 3298
NOINT option, 3298
NOLOGSCALE option, 3298
OFFSET= option, 3298
ORDER= option, 3289
OUTROC= option, 3298
PARMLABEL option, 3298
PEVENT= option, 3298
PLCL option, 3299
PLCONV= option, 3299
PLRL option, 3299
PPROB= option, 3299
REFERENCE= option, 3289
RIDGING= option, 3299
RISKLIMITS option, 3299
ROCEPS= option, 3299
RSQUARE option, 3300
SCALE= option, 3300
SELECTION= option, 3301
SEQUENTIAL option, 3301
SINGULAR= option, 3301
SLENTRY= option, 3301
SLSTAY= option, 3301
START= option, 3301
STB option, 3302
STOP= option, 3302
STOPRES option, 3302
TECHNIQUE= option, 3302
WALDCL option, 3302
WALDRL option, 3299
XCONV= option, 3303
LOGISTIC procedure, ODDSRATIO statement, 3303
AT option, 3304
CL= option, 3304
DIFF= option, 3304
PLCONV= option, 3304
PLMAXITER= option, 3304
PLSINGULAR= option, 3304
LOGISTIC procedure, OUTPUT statement, 3305
ALPHA= option, 3306
C= option, 3306
CBAR= option, 3306
DFBETAS= option, 3306
DIFCHISQ= option, 3306
DIFDEV= option, 3307
H= option, 3307
LOWER= option, 3307
OUT= option, 3307
PREDICTED= option, 3307
PREDPROBS= option, 3307

RESCHI= option, 3308
RESDEV= option, 3308
STDXBETA = option, 3308
UPPER= option, 3308
XBETA= option, 3308
LOGISTIC procedure, PROC LOGISTIC statement, 3267
ALPHA= option, 3268
COVOUT option, 3268
DATA= option, 3268
DESCENDING option, 3268
EXACTONLY option, 3268
EXACTOPTIONS option, 3269
INEST= option, 3270
INMODEL= option, 3271
MULTIPASS option, 3271
NAMELEN= option, 3271
NOCOV option, 3271
NOPRINT option, 3271
ORDER= option, 3271
OUTDESIGN= option, 3271
OUTDESIGNONLY option, 3272
OUTEST= option, 3272
OUTMODEL= option, 3272
PLOTS option, 3272
ROCOPTIONS option, 3274
SIMPLE option, 3275
TRUNCATE option, 3275
LOGISTIC procedure, ROC statement, 3310
LINK= option, 3310
NOOFFSET option, 3310
LOGISTIC procedure, ROCCONTRAST statement, 3311
ADJACENTPAIRS option, 3311
COV option, 3311
E option, 3311
ESTIMATE option, 3311
REFERENCE option, 3311
LOGISTIC procedure, SCORE statement, 3312
ALPHA= option, 3312
CLM option, 3312
CUMULATIVE option, 3312
DATA= option, 3312
FITSTAT option, 3313
OUT= option, 3313
OUTROC= option, 3313
PRIOR= option, 3313
PRIOREVENT= option, 3313
ROCEPS= option, 3313
LOGISTIC procedure, STRATA statement, 3314
CHECKDEPENDENCY= option, 3315
INFO option, 3315
MISSING option, 3314
NOSUMMARY option, 3315
LOGISTIC procedure, TEST statement, 3315
PRINT option, 3316
LOGISTIC procedure, UNITS statement, 3316
DEFAULT= option, 3317
LOGISTIC procedure, WEIGHT statement, 3317
NORMALIZE option, 3317
LOGISTIC statement
POWER procedure, 4827
LOGIT function
RESPONSE statement (CATMOD), 1131
LOGIT transformation
MODEL statement (TRANSREG), 7144
TRANSFORM statement (MI), 3767
TRANSFORM statement (PRINQUAL), 5214
LOGLIN statement
CATMOD procedure, 1117
LOGNLAMBDA0= option
MODEL statement (TPSPLINE), 7071
LOGNLAMBDA= option
MODEL statement (TPSPLINE), 7071
LOGNOTE option
PROC MIXED statement, 3908
PROC NLMIXED statement, 4366
LOGNOTE= option
PRIOR statement (MIXED), 3947
LOGOR= option
REPEATED statement (GENMOD), 1960
LOGRBOUND= option
PRIOR statement (MIXED), 3947
LOWER= option
ONESAMPLEFREQ statement (POWER), 4846
ONESAMPLEMEANS statement (POWER), 4854
OUTPUT statement (LOGISTIC), 3307
OUTPUT statement (SURVEYLOGISTIC), 6403
PAIREDMEANS statement (POWER), 4873
TWOSAMPLEMEANS statement (POWER), 4894
LOWERB= option
PARMS statement (GLIMMIX), 2182
PARMS statement (HPMIXED), 2827
PARMS statement (MIXED), 3944
LOWERTAILED option
ESTIMATE statement (GLIMMIX), 2138
ESTIMATE statement (MIXED), 3921
LSMESTIMATE statement (GLIMMIX), 2159
TEST statement (MULTTEST), 4202
LPREDPLOT statement
options summarized by function, 5288

PROBIT procedure, 5287
LPREFIX= option
CLASS statement (GLMSELECT), 2703
CLASS statement (LOGISTIC), 3280
CLASS statement (PHREG), 4557
CLASS statement (SURVEYLOGISTIC), 6389
MODEL statement (TRANSREG), 7153, 7162
LRCHI option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1696, 1815
LRCHISQ option
TABLES statement (SURVEYFREQ), 6317
LRCHISQ1 option
TABLES statement (SURVEYFREQ), 6317
LRCI option
MODEL statement (GENMOD), 1950
LREF= option
MCMC statement (MI), 3758
LSCOEFFS option
MODEL statement (GLMSELECT), 2711
LSD option
MEANS statement (ANOVA), 713, 714
MEANS statement (GLM), 2472
LSMEANS statement
GLIMMIX procedure, 2141
GLM procedure, 2458
HPMIXED procedure, 2821
MIXED procedure, 3922
LSMESTIMATE statement
GLIMMIX procedure, 2155
LSP option
NLOPTIONS statement (GLIMMIX), 399
NLOPTIONS statement (HPMIXED), 399
NLOPTIONS statement (TCALIS), 399
LSPRECISION option
NLOPTIONS statement (GLIMMIX), 399
NLOPTIONS statement (HPMIXED), 399
NLOPTIONS statement (TCALIS), 399
LSPRECISION= option
NLOPTIONS statement (CALIS), 889
PROC CALIS statement, 857
PROC NLMIXED statement, 4366
PROC TCALIS statement, 6732
LVREF= option
PLOT statement (BOXPLOT), 783
PLOT statement (REG), 5495
LWEIGHT= option
MODEL statement (GLIMMIX), 2172

M= option
MANOVA statement (ANOVA), 706
MANOVA statement (GLM), 2464
MODEL statement (TPSPLINE), 7071
MACRO option
OUTPUT statement (TRANSREG), 7174
MAHADIST keyword
OUTPUT statement (QUANTREG), 5381
MAHALANOBIS option
PROC CANDISC statement, 1070
PROC DISCRIM statement, 1392
main model specification statements, TCALIS procedure, 6717
MANOVA option
PROC ANOVA statement, 702
PROC DISCRIM statement, 1393
PROC GLM statement, 2446
MANOVA statement
ANOVA procedure, 706
GLM procedure, 2464
MARGIN= option
ONESAMPLEFREQ statement (POWER), 4847
MARGINAL function
RESPONSE statement (CATMOD), 1131
MARKERS= option
PLOT statement (GLMPOWER), 2655
PLOT statement (POWER), 4881
MATINGS statement, INBREED procedure, 2867
MATRIX option
PROC INBREED statement, 2865
MATRIX statement
CALIS procedure, 883
MDS procedure, 3722
TCALIS procedure, 6784
MATRIXL option
PROC INBREED statement, 2865
MAX= option
PLOT statement (GLMPOWER), 2655
PLOT statement (POWER), 4881
PREDICT statement (KRIGE2D), 2943
MAXCLPRINT= option
PROC HPMIXED statement, 2812
MAXCLUSTERS= option
PROC FASTCLUS statement, 1632
PROC MODECLUS statement, 4106
PROC VARCLUS statement, 7471
MAXEIGEN= option
PROC VARCLUS statement, 7471
MAXFU option
NLOPTIONS statement (GLIMMIX), 399
NLOPTIONS statement (HPMIXED), 399
NLOPTIONS statement (TCALIS), 399
MAXFUNC option
NLOPTIONS statement (GLIMMIX), 399
NLOPTIONS statement (HPMIXED), 399

NLOPTIONS statement (TCALIS), 399
MAXFUNC= option
NLOPTIONS statement (CALIS), 889, 891
PROC CALIS statement, 858
PROC MIXED statement, 3908
PROC NLMIXED statement, 4367
PROC TCALIS statement, 6732
MAXFUNCTION= option
MODEL statement (LOGISTIC), 3297
MAXHEIGHT= option
PROC TREE statement, 7365
MAXIMUM option
RIDGE statement (RSREG), 5734
MAXIMUM= option
PROC MI statement, 3752
MAXINFO= option
PROC SEQDESIGN statement, 5801
MAXIT option
NLOPTIONS statement (GLIMMIX), 399
NLOPTIONS statement (HPMIXED), 399
NLOPTIONS statement (TCALIS), 399
MAXIT= option
MODEL statement (GENMOD), 1950
PROC QUANTREG statement, 5373
MAXITER = option
MODEL statement (GAM), 1845
MAXITER option
NLOPTIONS statement (GLIMMIX), 399
NLOPTIONS statement (HPMIXED), 399
NLOPTIONS statement (TCALIS), 399
MAXITER= option
COVTEST statement (GLIMMIX), 2133
EM statement (MI), 3755
MCMC statement (MI), 3760
MODEL statement (CATMOD), 1121
MODEL statement (LIFEREG), 3024
MODEL statement (LOGISTIC), 3297
MODEL statement (PHREG), 4568
MODEL statement (SURVEYLOGISTIC), 6400
MODEL statement (TRANSREG), 7162
NLOPTIONS statement (CALIS), 889, 891
PROC ACECLUS statement, 675
PROC CALIS statement, 858
PROC FACTOR statement, 1552
PROC FASTCLUS statement, 1638
PROC MDS statement, 3716
PROC MIXED statement, 3908
PROC NLIN statement, 4281
PROC NLMIXED statement, 4367
PROC PLS statement, METHOD=PLS option, 4777
PROC PLS statement, MISSING=EM option, 4777
PROC PRINQUAL statement, 5205
PROC ROBUSTREG statement (ROBUSTREG), 5663, 5666, 5667
PROC TCALIS statement, 6733
PROC VARCLUS statement, 7472
PROC VARCOMP statement, 7498
REPEATED statement (GENMOD), 1960
TABLES statement (FREQ), 1709
MAXITSCORE = option
MODEL statement (GAM), 1845
MAXLAGS= option
COMPUTE statement (VARIOGRAM), 7537
MAXLEGENDAREA= option
ODS GRAPHICS statement, 530
MAXLMMUPDATE option
PROC GLIMMIX statement, 2101
MAXMACRO= option
PROC GLMSELECT statement, 2695
PROC STEPDISC statement, 6275
MAXOPT option
PROC GLIMMIX statement, 2101
MAXPANELS= option
PLOT statement (BOXPLOT), 783
maxrank option
LMTESTS statement, 6776
MAXSEARCH= option
PROC VARCLUS statement, 7472
MAXSIZE= option
PROC SURVEYSELECT statement, 6628
MAXSTEP option
MODEL statement (GLMSELECT), 2712
MODEL statement (REG), 5476
NLOPTIONS statement (GLIMMIX), 400
NLOPTIONS statement (HPMIXED), 400
NLOPTIONS statement (TCALIS), 400
MAXSTEP= option
MODEL statement (LOGISTIC), 3297
MODEL statement (PHREG), 4568
PROC NLMIXED statement, 4367
PROC STEPDISC statement, 6276
MAXSUBIT= option
PROC NLIN statement, 4281
MAXTIME option
NLOPTIONS statement (GLIMMIX), 400
NLOPTIONS statement (HPMIXED), 400
NLOPTIONS statement (TCALIS), 400
MAXTIME= option
EXACT statement (FREQ), 1693
EXACT statement (NPAR1WAY), 4464
NLOPTIONS statement (CALIS), 891
PROC LIFETEST statement, 3116
PROC NLMIXED statement, 4367
MAXTUNE= option

PROC MCMC statement, 3501
MC option
EXACT statement (FREQ), 1693
EXACT statement (NPAR1WAY), 4464
MCA option
PROC CORRESP statement, 1327
MCA= option, PROC CORRESP statement, 1354
MCMC procedure, 3495
ARRAY statement, 3508
BEGINCNST statement, 3509
BEGINNODATA statement, 3511
BEGINPRIOR statement, 3511
BY statement, 3511
ENDCNST statement, 3509
ENDNODATA statement, 3511
ENDPRIOR statement, 3511
HYPERPRIOR statement, 3516
MODEL statement, 3512
PARMS statement, 3515
PRIOR statement, 3516
syntax, 3495
MCMC procedure, ARRAY statement, 3508
MCMC procedure, BEGINCNST statement, 3509
MCMC procedure, BEGINNODATA statement, 3511
MCMC procedure, BEGINPRIOR statement, 3511
MCMC procedure, BY statement, 3511
MCMC procedure, ENDCNST statement, 3509
MCMC procedure, ENDNODATA statement, 3511
MCMC procedure, ENDPRIOR statement, 3511
MCMC procedure, HYPERPRIOR statement, 3516
MCMC procedure, MODEL statement, 3512
MCMC procedure, PARMS statement, 3515
MCMC procedure, PRIOR statement, 3516
MCMC procedure, PROC MCMC statement
ACCEPTTOL= option, 3497
AUTOCORLAG= option, 3497
DATA= option, 3500
DIAG= option, 3498
DIAGNOSTICS= option, 3498
DIC option, 3500
DISCRETE= option, 3497
INF= option, 3500
INIT= option, 3500
JOINTMODEL option, 3501
LIST option, 3501
LISTCODE option, 3501
MAXTUNE= option, 3501
MINTUNE= option, 3502
MISSING= option, 3502
MONITOR= option, 3502
NBI= option, 3502
NMC= option, 3502
NTU= option, 3502
OUTPOST=option, 3502
PLOTS= option, 3503
PROPCOV= option, 3505
PROPDIST= option, 3506
SCALE option, 3506
SEED option, 3506
SIMREPORT= option, 3506
SINGDEN= option, 3506
STATISTICS= option, 3507
STATS= option, 3507
TARGACCEPT= option, 3508
TARGACCEPTI= option, 3508
THIN= option, 3508
TRACE option, 3508
TUNEWT= option, 3508
MCMC procedure, Programming statements
ABORT statement, 3517
CALL statement, 3517
DELETE statement, 3517
DO statement, 3517
GOTO statement, 3517
IF statement, 3517
LINK statement, 3517
PUT statement, 3517
RETURN statement, 3517
SELECT statement, 3517
STOP statement, 3517
SUBSTR statement, 3517
WHEN statement, 3517
MCMC statement
MI procedure, 3756
MCNEM option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1696
MCONVERGE= option
PROC MDS statement, 3716
MCORRB option
REPEATED statement (GENMOD), 1960
MCOVB option
REPEATED statement (GENMOD), 1960
MDATA= option
MODEL statement (KRIGE2D), 2945
SIMULATE statement (SIM2D), 6172
MDEGREE option
EFFECT statement, polynomial (GLIMMIX), 381
EFFECT statement, polynomial (GLMSELECT), 381
MDPREF= option

PROC PRINQUAL statement, 5205
MDS procedure
syntax, 3709
MDS procedure, BY statement, 3721
MDS procedure, ID statement, 3721
MDS procedure, INVAR statement, 3722
MDS procedure, MATRIX statement, 3722
MDS procedure, PROC MDS statement, 3710
ALTERNATE= option, 3712
COEF= option, 3712
CONDITION= option, 3713
CONVERGE= option, 3713
CRITMIN= option, 3717
CUTOFF= option, 3713
DATA= option, 3713
DECIMALS= option, 3714
DIMENSION= option, 3714
EPSILON= option, 3714
FIT= option, 3714
FORMULA= option, 3715
GCONVERGE= option, 3715
INAV= option, 3715
INITIAL= option, 3716
ITER= option, 3716
LEVEL= option, 3716
MAXITER= option, 3716
MCONVERGE= option, 3716
MINCRIT= option, 3717
NEGATIVE option, 3717
NONORM option, 3717
NOPHIST option, 3717
NOPRINT option, 3717
NOULB option, 3717
OCOEF option, 3717
OCONFIG option, 3717
OCRIT option, 3717
OTRANS option, 3717
OUT= option, 3718
OUTFIT= option, 3718
OUTITER option, 3717
OUTRES= option, 3718
OVER= option, 3718
PCOEF option, 3718
PCONFIG option, 3718
PDATA option, 3718
PFINAL option, 3718
PFIT option, 3718
PFITROW option, 3718
PINAVDATA option, 3718
PINEIGVAL option, 3718
PINEIGVEC option, 3719
PININ option, 3719
PINIT option, 3719
PITER option, 3719
PTRANS option, 3720
RANDOM= option, 3720
RIDGE= option, 3720
SHAPE= option, 3720
SIMILAR= option, 3720
SINGULAR= option, 3720
UNTIE option, 3721
MDS procedure, VAR statement, 3722
MDS procedure, WEIGHT statement, 3723
MEAN function
RESPONSE statement (CATMOD), 1131
MEAN keyword
REPEATED statement (ANOVA), 717
MEAN option
MCMC statement (MI), 3757, 3763
REPEATED statement (GLM), 2483, 2484, 2539
TEST statement (MULTTEST), 4201, 4209, 4229
MEAN statement
SIM2D procedure, 6175
MEAN statement, TCALIS procedure, 6788
MEAN= option
ONESAMPLEMEANS statement (POWER), 4854
PROC FASTCLUS statement, 1638
MEANDIFF= option
PAIREDMEANS statement (POWER), 4873
TWOSAMPLEMEANS statement (POWER), 4894
MEANRATIO= option
PAIREDMEANS statement (POWER), 4874
TWOSAMPLEMEANS statement (POWER), 4894
MEANS option
OUTPUT statement (TRANSREG), 7176
MEANS statement
ANOVA procedure, 709
GLM procedure, 2468
MEANSTR option
PROC TCALIS statement, 6733
MEASURES option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1696
TABLES statement (FREQ), 1709
TEST statement (FREQ), 1723
MEC option
OUTPUT statement (TRANSREG), 7176
MEDIAN option
EXACT statement (NPAR1WAY), 4463
OUTPUT statement (NPAR1WAY), 4465
PROC NPAR1WAY statement, 4458

METHOD= < (options)>
PROC ROBUSTREG statement, 5663
METHOD= option
BASELINE statement (PHREG), 4545
DESIGN statement (SEQDESIGN), 5805
MODEL statement (GAM), 1845
MODEL statement (TRANSREG), 7162
ONESAMPLEFREQ statement (POWER), 4847
OUTPUT statement (PHREG), 4575
PAIREDFREQ statement (POWER), 4866
PROC ACECLUS statement, 676
PROC CALIS statement, 859
PROC DISCRIM statement, 1393
PROC DISTANCE statement, 1494
PROC FACTOR statement, 1552
PROC GLIMMIX statement, 2102
PROC HPMIXED statement, 2812
PROC LIFETEST statement, 3116
PROC MIXED statement, 3908, 4020
PROC MODECLUS statement, 4106
PROC NLIN statement, 4281
PROC NLMIXED statement, 4368
PROC PLS statement, 4776
PROC PRINQUAL statement, 5206
PROC STDIZE statement, 6223
PROC STEPDISC statement, 6276
PROC SURVEYSELECT statement, 6629
PROC TCALIS statement, 6733
PROC VARCOMP statement, 7498
UNIVAR statement, 2894
METRIC= option
PROC ACECLUS statement, 676
PROC DISCRIM statement, 1393
MHCHI option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1696
MHOR option
OUTPUT statement (FREQ), 1696
MHRRC1 option
OUTPUT statement (FREQ), 1696
MHRRC2 option
OUTPUT statement (FREQ), 1696
MI procedure, BY statement, 3754
MI procedure, CLASS statement, 3754
MI procedure, EM statement, 3755
CONVERGE option, 3755
INITIAL= option, 3755
ITPRINT option, 3755
MAXITER= option, 3755
OUT= option, 3755
OUTEM= option, 3756
OUTITER= option, 3756
XCONV option, 3755
MI procedure, FREQ statement, 3756
MI procedure, MCMC statement, 3756
ACF option, 3761
ACFPLOT option, 3757
BOOTSTRAP option, 3760
CCONF= option, 3757
CCONNECT= option, 3763
CFRAME= option, 3757, 3763
CHAIN= option, 3759
CNEEDLES= option, 3758
CONVERGE= option, 3760
COV option, 3757, 3762
CREF= option, 3758
CSYMBOL= option, 3758, 3763
DISPLAYINIT option, 3759
GOUT= option, 3759
HSYMBOL= option, 3758, 3763
IMPUTE= option, 3759
INEST= option, 3759
INITIAL= option, 3759
ITPRINT option, 3760
LCONF= option, 3758
LCONNECT= option, 3763
LOG option, 3758, 3763
LREF= option, 3758
MAXITER= option, 3760
MEAN option, 3757, 3763
NAME= option, 3758, 3763
NBITER= option, 3760
NITER= option, 3760
NLAG= option, 3758
OUTEST= option, 3760
OUTITER= option, 3760
PRIOR= option, 3762
START= option, 3762
SYMBOL= option, 3758, 3763
TIMEPLOT option, 3762
TITLE= option, 3758, 3763
TRACE option, 3761
WCONF= option, 3758
WCONNECT= option, 3763
WLF option, 3757, 3763, 3764
WNEEDLES= option, 3758
WREF= option, 3758
XCONV= option, 3760
MI procedure, MONOTONE statement, 3764
DISCRIM option, 3765
LOGISTIC option, 3765
PROPENSITY option, 3766
REG option, 3766
REGPMM option, 3766
REGPREDMEANMATCH option, 3766
REGRESSION option, 3766
MI procedure, PROC MI statement, 3751

ALPHA= option, 3752
DATA= option, 3752
MAXIMUM= option, 3752
MINIMUM= option, 3752
MINMAXITER= option, 3752
MU0= option, 3752
NIMPUTE= option, 3753
NOPRINT option, 3753
OUT= option, 3753
ROUND= option, 3753
SEED option, 3753
SIMPLE, 3754
SINGULAR option, 3754
THETA0= option, 3752
MI procedure, TRANSFORM statement, 3767
BOXCOX transformation, 3767
C= option, 3768
EXP transformation, 3767
LAMBDA= option, 3768
LOG transformation, 3767
LOGIT transformation, 3767
POWER transformation, 3767
MI procedure, VAR statement, 3768
MIANALYZE procedure, BY statement, 3847
MIANALYZE procedure, CLASS statement, 3848
MIANALYZE procedure, MODELEFFECTS statement, 3848
MIANALYZE procedure, PROC MIANALYZE statement, 3845
ALPHA= option, 3845
BCOV option, 3845
CLASSVAR= option, 3847
COVB= option, 3846
DATA= option, 3846
EDF= option, 3846
EFFECTVAR= option, 3846
MU0= option, 3847
MULT option, 3846
PARMINFO= option, 3847
PARMS= option, 3847
TCOV option, 3847
THETA0= option, 3847
WCOV option, 3847
XPXI= option, 3847
MIANALYZE procedure, STDERR statement, 3849
MIANALYZE procedure, TEST statement, 3849
BCOV option, 3850
MULT option, 3850
TCOV option, 3850
WCOV option, 3851
MIDPFACTOR= option
EXACT statement (LOGISTIC), 3286
MIN= option
PLOT statement (GLMPOWER), 2655
PLOT statement (POWER), 4881
MINC= option
PROC VARCLUS statement, 7472
MINCLUSTERS= option
PROC VARCLUS statement, 7472
MINCRIT= option
PROC MDS statement, 3717
MINEIGEN= option
PROC FACTOR statement, 1553
MINHEIGHT= option
PROC TREE statement, 7365
MINIMUM option
RIDGE statement (RSREG), 5734
MINIMUM= option
PROC MI statement, 3752
MININERTIA= option
PROC CORRESP statement, 1327
MINIT option
NLOPTIONS statement (GLIMMIX), 400
NLOPTIONS statement (HPMIXED), 400
NLOPTIONS statement (TCALIS), 400
MINITER option
NLOPTIONS statement (GLIMMIX), 400
NLOPTIONS statement (HPMIXED), 400
NLOPTIONS statement (TCALIS), 400
MINITER= option
NLOPTIONS statement (CALIS), 891
PROC NLMIXED statement, 4368
MINMAXITER= option
PROC MI statement, 3752
MINPOINTS= option
PREDICT statement (KRIGE2D), 2943
MINSIZE= option
PROC SURVEYSELECT statement, 6631
MINTUNE= option
PROC MCMC statement, 3502
MISSBREAK option
PLOT statement (BOXPLOT), 783
MISSING option
CLASS statement (GENMOD), 1938
CLASS statement (GLMSELECT), 2703
CLASS statement (LOGISTIC), 3280
CLASS statement (PHREG), 4557
PROC CORRESP statement, 1327
PROC LIFETEST statement, 3117
PROC NPAR1WAY statement, 4458
PROC SURVEYFREQ statement, 6302
PROC SURVEYLOGISTIC statement, 6383
PROC SURVEYMEANS statement, 6474
PROC SURVEYREG statement, 6543
STRATA statement (LIFETEST), 3129
STRATA statement (LOGISTIC), 3314

STRATA statement (PHREG), 4577
TABLES statement (FREQ), 1709
MISSING= option
MODEL statement (CATMOD), 1123
PROC MCMC statement, 3502
PROC PLS statement, 4777
PROC STDIZE statement, 6223
VAR statement, 1503
MISSPRINT option
TABLES statement (FREQ), 1710
MIXED procedure, 3902
INFLUENCE option, 3934
syntax, 3902
MIXED procedure, BY statement, 3916
MIXED procedure, CLASS statement, 3916, 3995
TRUNCATE option, 3917
MIXED procedure, CONTRAST statement, 3917
CHISQ option, 3919
DF= option, 3919
E option, 3919
GROUP option, 3919
SINGULAR= option, 3920
SUBJECT option, 3920
MIXED procedure, ESTIMATE statement, 3920
ALPHA= option, 3921
CL option, 3921
DF= option, 3921
DIVISOR= option, 3921
E option, 3921
GROUP option, 3921
LOWERTAILED option, 3921
SINGULAR= option, 3921
SUBJECT option, 3921
UPPERTAILED option, 3922
MIXED procedure, ID statement, 3922
MIXED procedure, LSMEANS statement, 3922, 4032
ADJUST= option, 3924
ALPHA= option, 3925
AT MEANS option, 3925
AT option, 3925, 3926
BYLEVEL option, 3926, 3927
CL option, 3926
CORR option, 3926
COV option, 3926
DF= option, 3926
DIFF option, 3926
E option, 3927
OBSMARGINS option, 3927
PDIFF option, 3926, 3927
SINGULAR= option, 3928
SLICE= option, 3928
MIXED procedure, MODEL statement, 3928
ALPHAP= option, 3929
CHISQ option, 3929
CL option, 3929
CONTAIN option, 3930, 3931
CORRB option, 3930
COVB option, 3930
COVBI option, 3930
DDF= option, 3930
DDFM= option, 3930
E option, 3933
E1 option, 3933
E2 option, 3933
E3 option, 3933
FULLX option, 3925, 3933
HTYPE= option, 3934
INFLUENCE option, 3934
INTERCEPT option, 3939
LCOMPONENTS option, 3939
NOCONTAIN option, 3940
NOINT option, 3940, 3982
NOTEST option, 3940
ORDER= option, 3986
OUTP= option, 4032
OUTPRED= option, 3940
OUTPREDM= option, 3941
RESIDUAL option, 3941, 3988
SINGCHOL= option, 3941
SINGRES= option, 3941
SINGULAR= option, 3942
SOLUTION option, 3942, 3986
VCIRY option, 3942, 3988
XPVIX option, 3942
XPVIXI option, 3942
ZETA= option, 3942
MIXED procedure, MODEL statement, INFLUENCE option
EFFECT=, 3935
ESTIMATES, 3935
ITER=, 3935
KEEP=, 3936
SELECT=, 3936
SIZE=, 3937
MIXED procedure, PARMS statement, 3943, 4032
EQCONS= option, 3944
HOLD= option, 3944
LOGDETH option, 3944
LOWERB= option, 3944
NOBOUND option, 3944
NOITER option, 3944
NOPROFILE option, 3944
OLS option, 3945
PARMSDATA= option, 3945
PDATA= option, 3945

RATIOS option, 3945
UPPERB= option, 3945
MIXED procedure, PRIOR statement, 3945
ALG= option, 3947
BDATA= option, 3947
DATA= option, 3946
FLAT option, 3947
GRID= option, 3947
GRIDT= option, 3947
IFACTOR= option, 3947
JEFFREYS option, 3947
LOGNOTE= option, 3947
LOGRBOUND= option, 3947
NSAMPLE= option, 3948
NSEARCH= option, 3948
OUT= option, 3948
OUTG= option, 3948
OUTGT= option, 3948
PSEARCH option, 3948
PTRANS option, 3948
SEED= option, 3948
SFACTOR= option, 3948
TDATA= option, 3949
TRANS= option, 3949
UPDATE= option, 3949
MIXED procedure, PROC MIXED statement, 3904
ABSOLUTE option, 3905, 3996
ALPHA= option, 3905
ANOVAF option, 3905
ASYCORR option, 3905
ASYCOV option, 3905, 4032
CL= option, 3906
CONVF option, 3906, 3996
CONVG option, 3906, 3996
CONVH option, 3906, 3996
COVTEST option, 3907, 3997
DATA= option, 3907
DFBW option, 3907
IC option, 3907
INFO option, 3908
ITDETAILS option, 3908
LOGNOTE option, 3908
MAXFUNC= option, 3908
MAXITER= option, 3908
METHOD= option, 3908, 4020
MMEQ option, 3909, 4032
MMEQSOL option, 3909, 4032
NAMELEN= option, 3909
NOBOUND option, 3909
NOCLPRINT option, 3909
NOINFO option, 3909
NOITPRINT option, 3909
NOPROFILE option, 3910, 3975
ORD option, 3910
ORDER= option, 3910, 3982
PLOT option, 3911
PLOTS option, 3911
RATIO option, 3915, 3997
RIDGE= option, 3915
SCORING= option, 3915
SIGITER option, 3915
UPDATE option, 3915
MIXED procedure, RANDOM statement, 3895, 3949, 4014
ALPHA= option, 3950
CL option, 3950
G option, 3950
GC option, 3950
GCI option, 3950
GCORR option, 3951
GDATA= option, 3951
GI option, 3951
GROUP= option, 3951
LDATA= option, 3951
NOFULLZ option, 3952
RATIOS option, 3952
SOLUTION option, 3952
SUBJECT= option, 3918, 3952
TYPE= option, 3952
V option, 3953
VC option, 3953
VCI option, 3953
VCORR option, 3953
VI option, 3953
MIXED procedure, REPEATED statement, 3895, 3954, 4019
GROUP= option, 3955
HLM option, 3955
HLPS option, 3955
LDATA= option, 3956
LOCAL= option, 3956
LOCALW option, 3957
NONLOCALW option, 3957
R option, 3957
RC option, 3958
RCI option, 3958
RCORR option, 3958
RI option, 3958
SSCP option, 3958
SUBJECT= option, 3958
TYPE= option, 3959
MIXED procedure, WEIGHT statement, 3968
ML option
MODEL statement (CATMOD), 1121
MMEQ option
PROC HPMIXED statement, 2813
PROC MIXED statement, 3909, 4032

MMEQSOL option
PROC MIXED statement, 3909, 4032
MNAMES= option
MANOVA statement (ANOVA), 707
MANOVA statement (GLM), 2465
MODE= option
PROC CLUSTER statement, 1243
PROC MODECLUS statement, 4106
MODECLUS procedure
syntax, 4101
MODECLUS procedure, BY statement, 4108
MODECLUS procedure, FREQ statement, 4109
MODECLUS procedure, ID statement, 4109
MODECLUS procedure, PROC MODECLUS statement, 4101
ALL option, 4103
AM option, 4103
BOUNDARY option, 4103
CASCADE= option, 4103
CK= option, 4104
CLUSTER= option, 4104
CORE option, 4104
CR= option, 4104
CROSS option, 4104
CROSSLIST option, 4104
DATA= option, 4104
DENSITY= option, 4104
DIMENSION= option, 4104
DK= option, 4105
DOCK= option, 4105
DR= option, 4105
EARLY option, 4105
HM option, 4105
JOIN= option, 4105
K= option, 4105
LIST option, 4105
LOCAL option, 4105
MAXCLUSTERS= option, 4106
METHOD= option, 4106
MODE= option, 4106
NEIGHBOR option, 4106
NOPRINT option, 4106
NOSUMMARY option, 4106
OUT= option, 4107
OUTCLUS= option, 4107
OUTLENGTH= option, 4107
OUTSUM= option, 4107
POWER= option, 4107
R= option, 4107
SHORT option, 4107
SIMPLE option, 4108
STANDARD option, 4108
SUM option, 4108
TEST option, 4108
THRESHOLD= option, 4108
TRACE option, 4108
MODECLUS procedure, VAR statement, 4109
model analysis statements, TCALIS procedure, 6718
MODEL statement
ANOVA procedure, 715
CATMOD procedure, 1118
GAM procedure, 1842
GENMOD procedure, 1946
GLIMMIX procedure, 2161
GLM procedure, 2474
GLMMOD procedure, 2629
GLMPOWER procedure, 2653
GLMSELECT procedure, 2706
HPMIXED procedure, 2822
KRIGE2D procedure, 2944
LIFEREG procedure, 3020
LOESS procedure, 3208
LOGISTIC procedure, 3288
MCMC procedure, 3512
MIXED procedure, 3928
NLIN procedure, 4286
NLMIXED procedure, 4376
ORTHOREG procedure, 4512
PHREG procedure, 4564
PLS procedure, 4782
QUANTREG procedure, 5378
REG procedure, 5469
ROBUSTREG procedure, 5668
RSREG procedure, 5731
SURVEYLOGISTIC procedure, 6395
SURVEYREG procedure, 6555
TCALIS procedure, 6789
TPSPLINE procedure, 7069
TRANSREG procedure, 7138
VARCOMP procedure, 7499
MODEL= option
MULTREG statement (POWER), 4836
ONECORR statement (POWER), 4841
SAMPLESIZE statement (SEQDESIGN), 5809
MODELEFFECTS statement
MIANALYZE procedure, 3848
MODELFONT option
PLOT statement (REG), 5495
MODELHT option
PLOT statement (REG), 5495
MODELLAB option
PLOT statement (REG), 5495
MODELSE option
REPEATED statement (GENMOD), 1960
MODIFICATION option
PROC CALIS statement, 860

PROC TCALIS statement, 6734
Modifiers of INFLUENCE option
MODEL statement (MIXED), 3934
MONITOR= option
PROC MCMC statement, 3502
MONOTONE statement
MI procedure, 3764
MONOTONE transformation
MODEL statement (TRANSREG), 7146
TRANSFORM statement (PRINQUAL), 5215
MONOTONE= option
MODEL statement (TRANSREG), 7163
PROC PRINQUAL statement, 5206
MOOD option
EXACT statement (NPAR1WAY), 4463
OUTPUT statement (NPAR1WAY), 4465
PROC NPAR1WAY statement, 4458
MPAIRS= option
PROC ACECLUS statement, 676
MPC option
OUTPUT statement (TRANSREG), 7176
MQC option
OUTPUT statement (TRANSREG), 7176
MRC option
OUTPUT statement (TRANSREG), 7177
MREDUNDANCY option
OUTPUT statement (TRANSREG), 7177
MSA option
PROC FACTOR statement, 1553
MSE option
MODEL statement (REG), 5476
PLOT statement (REG), 5495
MSINGULAR= option
NLOPTIONS statement (CALIS), 890
NLOPTIONS statement (GLIMMIX), 400
NLOPTIONS statement (HPMIXED), 400
NLOPTIONS statement (TCALIS), 400
PROC CALIS statement, 860
PROC NLMIXED statement, 4368
PROC TCALIS statement, 6734
MSPLINE transformation
MODEL statement (TRANSREG), 7146
TRANSFORM statement (PRINQUAL), 5215
MSTAT= option
MANOVA statement (ANOVA), 707
MANOVA statement (GLM), 2466
MTEST statement (REG), 5482
REPEATED statement (ANOVA), 718
REPEATED statement (GLM), 2484
MSTRUCT statement, TCALIS procedure, 6792
MTEST statement
REG procedure, 5480
MU0= option
PROC MI statement, 3752
PROC MIANALYZE statement, 3847
MULT option
PROC MIANALYZE statement, 3846
TEST statement (MIANALYZE), 3850
MULT= option
PROC DISTANCE statement, 1497
PROC STDIZE statement, 6223
MULTIPASS option
PROC ANOVA statement, 702
PROC GLM statement, 2447
PROC LOGISTIC statement, 3271
PROC PHREG statement, 4537
MULTIPLEGROUP option
PROC VARCLUS statement, 7472
MULTREG statement
POWER procedure, 4835
MULTTEST procedure, 4187
syntax, 4187
MULTTEST procedure, BY statement, 4197
MULTTEST procedure, CLASS statement, 4198
TRUNCATE option, 4198
MULTTEST procedure, CONTRAST statement, 4199
MULTTEST procedure, FREQ statement, 4200
MULTTEST procedure, PROC MULTTEST statement, 4187
ADAPTIVEFDR option, 4189, 4218
ADAPTIVEHOCHBERG option, 4188
ADAPTIVEHOLM option, 4189
BONFERRONI option, 4189, 4212
BOOTSTRAP option, 4184, 4189, 4213, 4229
CENTER option, 4189
DATA= option, 4189
DEPENDENTFDR option, 4189, 4217
EPSILON= option, 4190
FDR option, 4190, 4216
FDRBOOT option, 4190, 4217
FDRPERM option, 4190, 4217
FISHER_C option, 4190, 4215
HOC option, 4190, 4215
HOLM option, 4190, 4197
HOM option, 4190
HOMMEL option, 4215
INPVALUES= option, 4191
NOCENTER option, 4191
NOPRINT option, 4191
NOPVALUE option, 4191
NOTABLES option, 4191
NOZEROS option, 4191
NSAMPLE= option, 4191
NTRUENULL= option, 4191

ORDER= option, 4193, 4236
OUT= option, 4194, 4220
OUTPERM= option, 4194, 4221, 4225
OUTSAMP= option, 4194, 4222, 4229
PDATA= option, 4194
PERMUTATION option, 4194, 4213, 4225, 4236
PFDR option, 4194, 4218
PLOTS= option, 4195
PTRUENULL= option, 4196
RANUNI option, 4196
SEED= option, 4196
SIDAK option, 4197, 4213, 4233
STEPBON option, 4197
STEPBOOT option, 4197
STEPPERM option, 4197
STEPSID option, 4197, 4233
MULTTEST procedure, STRATA statement, 4200
WEIGHT= option, 4200, 4206
MULTTEST procedure, TEST statement, 4201
BINOMIAL option, 4201
CA option, 4201, 4203, 4225
CONTINUITY= option, 4202
DDFM= option, 4202
FISHER option, 4199, 4201, 4208, 4236
FT option, 4201, 4206, 4229
LOWERTAILED option, 4202
MEAN option, 4201, 4209, 4229
PERMUTATION= option, 4202, 4204, 4225
PETO option, 4201, 4207, 4233
TIME= option, 4202
UPPERTAILED option, 4202

N option
OUTPUT statement (FREQ), 1696
N= option
EXACT statement (FREQ), 1693
EXACT statement (NPAR1WAY), 4464
FACTOR statement (CALIS), 878
FACTOR statement (TCALIS), 6754
PROC ACECLUS statement, 676
PROC PRINCOMP statement, 5152
PROC PRINQUAL statement, 5206
PROC SURVEYLOGISTIC statement, 6384
PROC SURVEYMEANS statement, 6477
PROC SURVEYREG statement, 6545
NAME statement
TREE procedure, 7368
NAME= option
GROUP statement, 6765
MCMC statement (MI), 3758, 3763
MODEL statement, 6790
MODEL statement (TRANSREG), 7157
PLOT statement (BOXPLOT), 783
PLOT statement (GLMPOWER), 2657
PLOT statement (POWER), 4884
PLOT statement (REG), 5495
PROC TREE statement, 7365
TRANSFORM statement (PRINQUAL), 5219
NAMELEN= option
PROC ANOVA statement, 702
PROC CATMOD statement, 1108
PROC GENMOD statement, 1922
PROC GLIMMIX statement, 2108
PROC GLM statement, 2447
PROC GLMMOD statement, 2626
PROC GLMSELECT statement, 2696
PROC HPMIXED statement, 2813
PROC LIFEREG statement, 3006
PROC LOGISTIC statement, 3271
PROC MIXED statement, 3909
PROC PHREG statement, 4537
PROC PROBIT statement, 5266
PROC QUANTREG statement, 5374
PROC ROBUSTREG statement, 5661
PROC SURVEYLOGISTIC statement, 6383
NARROW option
PROC SIM2D statement, 6166
NBEST option
PROC ROBUSTREG statement, 5665
NBI= option
BAYES statement(PHREG), 4549
PROC MCMC statement, 3502
NBINS= option
LOGISTIC statement (POWER), 4830
TWOSAMPLEWILCOXON statement (POWER), 4912
NBITER= option
MCMC statement (MI), 3760
NCAN= option
MODEL statement (TRANSREG), 7163
PROC CANCORR statement, 1040
PROC CANDISC statement, 1070
PROC DISCRIM statement, 1393
NCLUSTERS= option
PROC TREE statement, 7365
NCOVARIATES= option
POWER statement (GLMPOWER), 2658
NDIRECTIONS= option
COMPUTE statement (VARIOGRAM), 7537
NEAREST suboption
RANDOM statement (GLIMMIX), 2190
NEGATIVE option
PROC MDS statement, 3717
NEIGHBOR option

PROC MODECLUS statement, 4106
NELSON option
PROC LIFETEST statement, 3117
NESTED procedure
syntax, 4256
NESTED procedure, BY statement, 4257
NESTED procedure, CLASS statement, 4258
TRUNCATE option, 4258
NESTED procedure, PROC NESTED statement, 4257
AOV option, 4257
DATA= option, 4257
NESTED procedure, VAR statement, 4258
NFAC= option
PROC PLS statement, 4777
NFACTORS= option
PROC FACTOR statement, 1554
NFRACTIONAL option
LOGISTIC statement (POWER), 4830
MULTREG statement (POWER), 4837
ONECORR statement (POWER), 4842
ONESAMPLEMEANS statement (POWER), 4854
ONEWAYANOVA statement (POWER), 4861
PAIREDFREQ statement (POWER), 4866
PAIREDMEANS statement (POWER), 4874
POWER statement (GLMPOWER), 2658
TWOSAMPLEFREQ statement (POWER), 4886
TWOSAMPLEMEANS statement (POWER), 4894
TWOSAMPLESURVIVAL statement (POWER), 4905
NFRACTIONAL= option
ONESAMPLEFREQ statement (POWER), 4847
TWOSAMPLEWILCOXON statement (POWER), 4912
NFULLPREDICTORS= option
MULTREG statement (POWER), 4837
NGRID= option
BIVAR statement, 2891
UNIVAR statement, 2894
NHCLASSES= option
COMPUTE statement (VARIOGRAM), 7538
NIMPUTE= option
PROC MI statement, 3753
NINTERVAL= option
PROC LIFETEST statement, 3117
NITER= option
MCMC statement (MI), 3760
PROC PLS statement, 4775
NKNOTS= option
MODEL statement (TRANSREG), 7151
TRANSFORM statement (PRINQUAL), 5219
NLAG= option
MCMC statement (MI), 3758
NLEGEND option
PLOT statement (BOXPLOT), 784
NLEVELS option
PROC FREQ statement, 1689
NLIN procedure
syntax, 4277
NLIN procedure, BOUNDS statement, 4283
NLIN procedure, BY statement, 4284
NLIN procedure, CONTROL statement, 4285
NLIN procedure, DER statement, 4285
NLIN procedure, ID statement, 4285
NLIN procedure, MODEL statement, 4286
NLIN procedure, OUTPUT statement, 4286
ALPHA= option, 4289
DER option, 4289
H= option, 4287
L95= option, 4287
L95M= option, 4287
LCL= option, 4287
LCLM= option, 4287
OUT= option, 4286
PARMS= option, 4287
PREDICTED= option, 4287
RESIDUAL= option, 4287
SSE= option, 4288
STDI= option, 4288
STDP= option, 4288
STDR= option, 4288
STUDENT= option, 4288
U95= option, 4288
U95M= option, 4288
UCL= option, 4288
UCLM= option, 4288
WEIGHT= option, 4288
NLIN procedure, PARAMETERS statement, 4289
NLIN procedure, PROC NLIN statement, 4278
ALPHA= option, 4279
BEST= option, 4279
CONVERGE= option, 4279
CONVERGEOBJ= option, 4279
CONVERGEPARM= option, 4280
DATA= option, 4280
FLOW option, 4280
G4 option, 4280
HOUGAARD option, 4280
LIST option, 4281

LISTALL option, 4281
LISTCODE option, 4281
LISTDEP option, 4281
LISTDER option, 4281
MAXITER= option, 4281
MAXSUBIT= option, 4281
METHOD= option, 4281
NOHALVE option, 4282
NOITPRINT option, 4282
NOPRINT option, 4282
OUTEST= option, 4282
PRINT option, 4282
RHO= option, 4282
SAVE option, 4282
SIGSQ= option, 4282
SINGULAR= option, 4282
SMETHOD= option, 4283
TAU= option, 4283
TOTALSS option, 4283
TRACE option, 4283
XREF option, 4283
NLIN procedure, program statements, 4292
NLIN procedure, programming statements, 4292
NLIN procedure, RETAIN statement, 4292
NLINCON statement, CALIS procedure, 886
NLINCON statement, TCALIS procedure, 6794
NLMIXED procedure, 4354
syntax, 4354
NLMIXED procedure, ARRAY statement, 4373
NLMIXED procedure, BOUNDS statement, 4373
NLMIXED procedure, BY statement, 4374
NLMIXED procedure, CONTRAST statement, 4374
DF= option, 4375
NLMIXED procedure, ESTIMATE statement, 4375
ALPHA= option, 4375
DF= option, 4375
NLMIXED procedure, ID statement, 4375
NLMIXED procedure, MODEL statement, 4376
NLMIXED procedure, NLOPTIONS statement
VSINGULAR= option, 403
NLMIXED procedure, PARMS statement, 4376
BEST= option, 4377
BYDATA option, 4377
DATA= option, 4377
NLMIXED procedure, PREDICT statement, 4377
ALPHA= option, 4377
DER option, 4378
DF= option, 4378
NLMIXED procedure, PROC NLMIXED statement
ABSCONV= option, 4357
ABSFCONV= option, 4357
ABSGCONV= option, 4358
ABSXCONV= option, 4358
ALPHA= option, 4358
ASINGULAR= option, 4358
CFACTOR= option, 4358
CORR option, 4359
COV option, 4359
COVSING= option, 4359
DAMPSTEP option, 4359
DATA= option, 4359
DF= option, 4359
DIAHES option, 4359
EBOPT option, 4359
EBSSFRAC option, 4359
EBSSTOL option, 4359
EBSTEPS option, 4360
EBSUBSTEPS option, 4360
EBTOL option, 4360
EBZSTART option, 4360
ECORR option, 4360
ECOV option, 4360
EDER option, 4360
EMPIRICAL option, 4360
FCONV2= option, 4361
FCONV= option, 4361
FD= option, 4361
FDHESSIAN= option, 4362
FDIGITS= option, 4362
FLOW option, 4362
FSIZE= option, 4362
G4= option, 4363
GCONV= option, 4363
HESCAL= option, 4363
HESS option, 4364
INHESSIAN option, 4364
INSTEP= option, 4364
ITDETAILS option, 4364
LCDEACT= option, 4365
LCEPSILON= option, 4365
LCSINGULAR= option, 4365
LINESEARCH= option, 4365
LIST option, 4366
LISTCODE option, 4366
LISTDEP option, 4366
LISTDER option, 4366
LOGNOTE option, 4366
LSPRECISION= option, 4366
MAXFUNC= option, 4367
MAXITER= option, 4367
MAXSTEP= option, 4367
MAXTIME= option, 4367
METHOD= option, 4368

MINITER= option, 4368
MSINGULAR= option, 4368
NOAD option, 4368
NOADSCALE option, 4368
OPTCHECK option, 4369
OUTQ= option, 4369
QFAC option, 4369
QMAX option, 4369
QPOINTS option, 4369
QSCALEFAC option, 4369
QTOL option, 4369
RESTART option, 4369
SEED option, 4370
SINGCHOL= option, 4370
SINGHESS= option, 4370
SINGSWEEP= option, 4370
SINGVAR option, 4370
START option, 4370
SUBGRADIENT option, 4370
TECHNIQUE= option, 4370, 4371
TRACE option, 4371
UPDATE= option, 4371, 4372
VSINGULAR= option, 4372
XCONV= option, 4372
XREF option, 4372
XSIZE= option, 4372
NLMIXED procedure, RANDOM statement, 4378
ALPHA= option, 4379
DF= option, 4379
OUT= option, 4379
NLMIXED procedure, REPLICATE statement, 4379
NLOPTIONS statement
GLIMMIX procedure, 391, 2175
HPMIXED procedure, 391, 2824
TCALIS procedure, 391
NLOPTIONS statement, CALIS procedure, 886
NLOPTIONS statement, TCALIS procedure, 6795
NMARKERS= option
PROC STDIZE statement, 6223
NMAX= option
PROC SURVEYSELECT statement, 6632
NMC= option
BAYES statement(PHREG), 4549
PROC MCMC statement, 3502
NMIN= option
PROC SURVEYSELECT statement, 6632
NMISS option
OUTPUT statement (FREQ), 1696
NOAD option
PROC NLMIXED statement, 4368
NOADJDF option
FITINDEX statement, 6760
PROC CALIS statement, 860
PROC TCALIS statement, 6735
NOADSCALE option
PROC NLMIXED statement, 4368
NOANOVA option
MODEL statement (RSREG), 5732
NOBOUND option
PARMS statement (GLIMMIX), 2183
PARMS statement (MIXED), 3944
PROC GLIMMIX statement, 2108
PROC MIXED statement, 3909
NOBS= option
PROC CALIS statement, 860
PROC FACTOR statement, 1554
PROC TCALIS statement, 6735
NOBSDETAIL option
PROC GLIMMIX statement, 2108
NOBYREF option
PLOT statement (BOXPLOT), 784
NOBYVAR option
PROC TTEST statement, 7402
NOCELLPERCENT option
TABLES statement (SURVEYFREQ), 6317
NOCENSPLOT option
PROC LIFETEST statement, 3117
NOCENTER option
MODEL statement (GLIMMIX), 2172
PROC MULTTEST statement, 4191
PROC PLS statement, 4777
NOCHART option
BOXPLOT procedure, 784
NOCHECK option
MODEL statement (LOGISTIC), 3297
MODEL statement (SURVEYLOGISTIC), 6400
PROC PRINQUAL statement, 5206
NOCLASSIFY option
PROC DISCRIM statement, 1393
NOCLPRINT option
PROC GLIMMIX statement, 2108
PROC HPMIXED statement, 2813
PROC MIXED statement, 3909
NOCODE option
MODEL statement (RSREG), 5732
NOCOL option
TABLES statement (FREQ), 1710
NOCOLLAPSE option
STRATA statement (SURVEYREG), 6560
NOCOLLECT option
PLOT statement (REG), 5498
NOCOLUMN= option
PROC CORRESP statement, 1327
NOCONTAIN option

MODEL statement (MIXED), 3940
NOCORR option
PROC FACTOR statement, 1554
NOCOV option
PROC LOGISTIC statement, 3271
NOCUM option
TABLES statement (FREQ), 1710
NOCVSTDIZE option
PROC PLS statement, 4777
NODECREMENT option
PREDICT statement (KRIGE2D), 2943
nodefault option
LMTESTS statement, 6776
NODESIGN option
MODEL statement (CATMOD), 1123
NODESIGNPRINT= option, *see* NODUMMYPRINT option)
MODEL statement (LOGISTIC), 3297
MODEL statement (SURVEYLOGISTIC), 6400
NODETAIL option
STRATA statement (LIFETEST), 3129
NODIAG option
MODEL statement (QUANTREG), 5379
PROC CALIS statement, 860
NODUMMYPRINT= option
MODEL statement (LOGISTIC), 3297
MODEL statement (PHREG), 4569
MODEL statement (SURVEYLOGISTIC), 6400
NOEFFECT option
EFFECT statement, multimember (GLIMMIX), 379
EFFECT statement, multimember (GLMSELECT), 379
NOEIGEN option
PROC CLUSTER statement, 1243
NOEIGNUM option
NLOPTIONS statement (CALIS), 898
NOFILL option
CONTRAST statement (SURVEYREG), 6551
ESTIMATE statement (SURVEYREG), 6554
NOFIT option
MODEL statement (LOGISTIC), 3298
MODEL statement (PHREG), 4569
PROC GLIMMIX statement, 2108
PROC HPMIXED statement, 2813
NOFRAME option
PLOT statement (BOXPLOT), 784
NOFREQ option
TABLES statement (FREQ), 1710
TABLES statement (SURVEYFREQ), 6317
NOFULLZ option
RANDOM statement (GLIMMIX), 2192
RANDOM statement (HPMIXED), 2830
RANDOM statement (MIXED), 3952
NOGOODFIT option
MODEL statement (ROBUSTREG), 5669
NOHALVE option
PROC NLIN statement, 4282
NOHLABEL option
PLOT statement (BOXPLOT), 784
NOHLF option
NLOPTIONS statement (CALIS), 895
NOID option
PROC CLUSTER statement, 1243
NOINCREMENT option
PREDICT statement (KRIGE2D), 2943
NOINDEXTYPE option
FITINDEX statement, 6760
PROC TCALIS statement, 6735
NOINFO option
PROC HPMIXED statement, 2813
PROC MIXED statement, 3909
NOINITGLM option
PROC GLIMMIX statement, 2109
NOINT option
MODEL statement (CATMOD), 1123
MODEL statement (GENMOD), 1950
MODEL statement (GLIMMIX), 2173, 2261
MODEL statement (GLM), 2476
MODEL statement (GLMMOD), 2629
MODEL statement (GLMSELECT), 2709
MODEL statement (HPMIXED), 2823
MODEL statement (LIFEREG), 3024
MODEL statement (LOGISTIC), 3298
MODEL statement (MIXED), 3940, 3982
MODEL statement (ORTHOREG), 4512
MODEL statement (QUANTREG), 5379
MODEL statement (REG), 5476
MODEL statement (ROBUSTREG), 5669
MODEL statement (SURVEYLOGISTIC), 6400
MODEL statement (SURVEYREG), 6556
MODEL statement (TRANSREG), 7163
MULTREG statement (POWER), 4837
PROC CALIS statement, 861
PROC CANCORR statement, 1040
PROC FACTOR statement, 1554
PROC PRINCOMP statement, 5152
PROC VARCLUS statement, 7472
NOITER option
MODEL statement (CATMOD), 1123
PARMS statement (GLIMMIX), 2183
PARMS statement (HPMIXED), 2827

PARMS statement (MIXED), 3944
NOITPRINT option
PROC GLIMMIX statement, 2109
PROC HPMIXED statement, 2813
PROC MIXED statement, 3909
PROC NLIN statement, 4282
NOLINE option
PLOT statement (REG), 5495
NOLIST option
PAINT statement (REG), 5486
REWEIGHT statement (REG), 5504
NOLOG option
MODEL statement (LIFEREG), 3024
NOLOGSCALE option
MODEL statement (LOGISTIC), 3298
NOM option
REPEATED statement (ANOVA), 718
REPEATED statement (GLM), 2484
NOMCAR option
PROC SURVEYFREQ statement, 6302
PROC SURVEYLOGISTIC statement, 6383
PROC SURVEYMEANS statement, 6474
PROC SURVEYREG statement, 6543
NOMISS option
MODEL statement (TRANSREG), 7164
OUTPUT statement (GLIMMIX), 2180
OUTPUT statement (HPMIXED), 2826
PROC DISTANCE statement, 1497
PROC FASTCLUS statement, 1638
PROC PRINQUAL statement, 5206
PROC STDIZE statement, 6223
NOMOD option
PROC CALIS statement, 861
PROC TCALIS statement, 6735
NOMODEL option
PLOT statement (REG), 5496
none option
PROC GAM statement, 1839
NONLOCALW option
REPEATED statement (MIXED), 3957
NONORM option
PROC CLUSTER statement, 1243
PROC MDS statement, 3717
NONSYMCL option
PROC SURVEYMEANS statement, 6475
NOOFFSET option
ROC statement (LOGISTIC), 3310
NOOPTIMAL option
MODEL statement (RSREG), 5732
NOORDERSPEC option
PROC TCALIS statement, 6735
NOOVERLAYLEGEND option
PLOT statement (BOXPLOT), 784
NOPARM option
MODEL statement (CATMOD), 1123
NOPARMNAME option
PROC TCALIS statement, 6735
NOPERCENT option
TABLES statement (FREQ), 1710
TABLES statement (SURVEYFREQ), 6317
NOPHIST option
PROC MDS statement, 3717
NOPREDVAR option
MODEL statement (CATMOD), 1123
NOPRINT
BIVAR statement, 2891
UNIVAR statement, 2894
NOPRINT option
FACTOR statement (PLAN), 4737
LSMEANS statement (GLM), 2461
MODEL statement (CATMOD), 1123
MODEL statement (REG), 5476
MODEL statement (RSREG), 5732
MODEL statement (TRANSREG), 7164
PROC ACECLUS statement, 676
PROC ANOVA statement, 702
PROC CALIS statement, 861
PROC CANCORR statement, 1040
PROC CANDISC statement, 1070
PROC CATMOD statement, 1108
PROC CLUSTER statement, 1243
PROC CORRESP statement, 1327
PROC DISCRIM statement, 1393
PROC FACTOR statement, 1554
PROC FASTCLUS statement, 1638
PROC FREQ statement, 1689
PROC GLM statement, 2447
PROC GLMMOD statement, 2626
PROC GLMSELECT statement, 2696
PROC HPMIXED statement, 2813
PROC INBREED statement, 2865
PROC KRIGE2D statement, 2937
PROC LIFEREG statement, 3006
PROC LIFETEST statement, 3117
PROC LOGISTIC statement, 3271
PROC MDS statement, 3717
PROC MI statement, 3753
PROC MODECLUS statement, 4106
PROC MULTTEST statement, 4191
PROC NLIN statement, 4282
PROC NPAR1WAY statement, 4458
PROC ORTHOREG statement, 4510
PROC PHREG statement, 4537
PROC PLS statement, 4778
PROC PRINCOMP statement, 5152
PROC PRINQUAL statement, 5207
PROC PROBIT statement, 5266
PROC REG statement, 5455

PROC RSREG statement, 5726
PROC SIM2D statement, 6166
PROC SURVEYSELECT statement, 6632
PROC TCALIS statement, 6735
PROC TREE statement, 7366
PROC VARCLUS statement, 7472
PROC VARIOGRAM statement, 7529
RIDGE statement (RSREG), 5734
TABLES statement (FREQ), 1710
TABLES statement (SURVEYFREQ), 6318
NOPROFILE option
MODEL statement (CATMOD), 1123
PARMS statement (MIXED), 3944
PROC GLIMMIX statement, 2109
PROC HPMIXED statement, 2813
PROC MIXED statement, 3910, 3975
NOPROMAXNORM option
PROC FACTOR statement, 1554
NOPVALUE option
PROC MULTTEST statement, 4191
norank option
LMTESTS statement, 6776
NOREML option
PROC GLIMMIX statement, 2109
NORESPONSE option
MODEL statement (CATMOD), 1124
NORESTOREMISSING option
OUTPUT statement (TRANSREG), 7177
NORM option
FACTOR statement (CALIS), 878
FACTOR statement (TCALIS), 6754
PROC DISTANCE statement, 1498
PROC STDIZE statement, 6223
NORM= option
PROC FACTOR statement, 1554
NORMAL option
EXACT statement (NPAR1WAY), 4463
OUTPUT statement (NPAR1WAY), 4465
PROC NPAR1WAY statement, 4461
NORMALIZE option
WEIGHT statement (LOGISTIC), 3317
WEIGHT statement (PHREG), 4579
NOROW option
TABLES statement (FREQ), 1710
NOROW= option
PROC CORRESP statement, 1328
NOSAMPLE option
STRATA statement (SURVEYSELECT), 6643
NOSCALE option
MODEL statement (GENMOD), 1950
MODEL statement (LIFEREG), 3025
PROC PLS statement, 4778, 4781
NOSCORES option
OUTPUT statement (TRANSREG), 7177
NOSEPARATE option
EFFECT statement, polynomial (GLIMMIX), 382
EFFECT statement, polynomial (GLMSELECT), 382
NOSERIFS option
PLOT statement (BOXPLOT), 784
NOSHAPE1 option
MODEL statement (LIFEREG), 3025
NOSORT option
MEANS statement (ANOVA), 713
MEANS statement (GLM), 2472
PROC SURVEYLOGISTIC statement, 6384
NOSPARSE option
PROC SURVEYMEANS statement, 6475
TABLES statement (FREQ), 1710
TABLES statement (SURVEYFREQ), 6318
NOSQUARE option
PROC CLUSTER statement, 1242, 1244
NOSTAND option
PROC TCALIS statement, 6735
NOSTAT option
PLOT statement (REG), 5496
NOSTD option
PROC DISTANCE statement, 1498
PROC SCORE statement, 5766
TABLES statement (SURVEYFREQ), 6318
NOSTDERR option
PROC CALIS statement, 861
PROC TCALIS statement, 6736
NOSUMMARY option
MODEL statement (QUANTREG), 5380
PROC MODECLUS statement, 4106
PROC PHREG statement, 4537
PROC SURVEYFREQ statement, 6303
STRATA statement (LOGISTIC), 3315
NOTABLE option
PROC LIFETEST statement, 3117
NOTABLES option
PROC MULTTEST statement, 4191
NOTCHES option
PLOT statement (BOXPLOT), 784
NOTEST option
MODEL statement (MIXED), 3940
STRATA statement (LIFETEST), 3129
NOTHREADS option
PERFORMANCE statement (GLMSELECT), 2717
PERFORMANCE statement (QUANTREG), 5383
PERFORMANCE statement (ROBUSTREG), 5671
NOTICKREP option

PLOT statement (BOXPLOT), 785
NOTIE option
PROC CLUSTER statement, 1244
NOTOTAL option
TABLES statement (SURVEYFREQ), 6318
NOTRUNCATE option
FREQ statement (PHREG), 4562
FREQ statement (STDIZE), 6227
NOU option
REPEATED statement (ANOVA), 718
REPEATED statement (GLM), 2484
NOULB option
PROC MDS statement, 3717
NOUNI option
MODEL statement (ANOVA), 715
MODEL statement (GLM), 2476
NOUNIQUE option
OUTPUT statement (GLIMMIX), 2180
OUTPUT statement (HPMIXED), 2826
NOVANGLE option
PLOT statement (BOXPLOT), 785
NOVAR option
OUTPUT statement (GLIMMIX), 2180
OUTPUT statement (HPMIXED), 2826
NOVARIOGRAM option
COMPUTE statement (VARIOGRAM), 7538
NOWARN option
TABLES statement (FREQ), 1711
NOWT option
TABLES statement (SURVEYFREQ), 6318
NOZEROCONSTANT option
OUTPUT statement (TRANSREG), 7164
NOZEROS option
PROC MULTTEST statement, 4191
NP option
PLOT statement (REG), 5496
NPAIRS= option
PAIREDFREQ statement (POWER), 4866
PAIREDMEANS statement (POWER), 4874
NPANELPOS= option
PLOT statement (BOXPLOT), 785
NPAR1WAY procedure
syntax, 4456
NPAR1WAY procedure, BY statement, 4462
NPAR1WAY procedure, CLASS statement, 4462
NPAR1WAY procedure, EXACT statement, 4462
AB option, 4463
ALPHA= option, 4464
CONOVER option, 4463
HL option, 4463
KLOTZ option, 4463
KS option, 4463
MAXTIME= option, 4464
MC option, 4464
MEDIAN option, 4463
MOOD option, 4463
N= option, 4464
NORMAL option, 4463
POINT option, 4464
SAVAGE option, 4463
SCORES=DATA option, 4463
SEED= option, 4464
ST option, 4463
VW option, 4463
WILCOXON option, 4463
NPAR1WAY procedure, FREQ statement, 4465
NPAR1WAY procedure, OUTPUT statement, 4465
AB option, 4465
ANOVA option, 4465
CONOVER option, 4465
EDF option, 4465
HL option, 4465
KLOTZ option, 4465
MEDIAN option, 4465
MOOD option, 4465
NORMAL option, 4465
OUT= option, 4465
output statistics, 4465
SAVAGE option, 4465
SCORES=DATA option, 4465
ST option, 4465
VW option, 4465
WILCOXON option, 4465
NPAR1WAY procedure, PROC NPAR1WAY statement, 4456
AB option, 4457
ALPHA= option, 4457
ANOVA option, 4457
CONOVER option, 4457
CORRECT=NO option, 4457
D option, 4458
DATA= option, 4458
EDF option, 4458
HL option, 4458
KLOTZ option, 4458
MEDIAN option, 4458
MISSING option, 4458
MOOD option, 4458
NOPRINT option, 4458
NORMAL option, 4461
PLOTS= option, 4459
SAVAGE option, 4461
SCORES=DATA option, 4461
ST option, 4461
VW option, 4461

WILCOXON option, 4461
NPAR1WAY procedure, VAR statement, 4466
NPARTIALVARS= option
ONECORR statement (POWER), 4842
NPATHS= option
ASSESS statement (PHREG), 4540
NPERGROUP= option
ONEWAYANOVA statement (POWER), 4861
TWOSAMPLEFREQ statement (POWER), 4886
TWOSAMPLEMEANS statement (POWER), 4895
TWOSAMPLESURVIVAL statement (POWER), 4905
TWOSAMPLEWILCOXON statement (POWER), 4912
NPLOTS= option
PROC FACTOR statement, 1555
NPOINTS= option
PLOT statement (GLMPOWER), 2655
PLOT statement (POWER), 4881
NREDUCEDPREDICTORS= option
MULTREG statement (POWER), 4837
NREP option
PROC ROBUSTREG statement, 5665, 5666
NSAMPLE= option
PRIOR statement (MIXED), 3948
PROC MULTTEST statement, 4191
NSEARCH= option
PRIOR statement (MIXED), 3948
NSR option
MODEL statement (TRANSREG), 7164
NSTAGES= option
DESIGN statement (SEQDESIGN), 5808
PROC SEQTEST statement, 6008
NSUBINTERVAL= option
TWOSAMPLESURVIVAL statement (POWER), 4905
NTEST= option
PROC PLS statement, 4775
NTESTPREDICTORS= option
MULTREG statement (POWER), 4837
NTICK= option
PROC TREE statement, 7365
NTOTAL= option
LOGISTIC statement (POWER), 4830
MULTREG statement (POWER), 4837
ONECORR statement (POWER), 4842
ONESAMPLEFREQ statement (POWER), 4847
ONESAMPLEMEANS statement (POWER), 4855
ONEWAYANOVA statement (POWER), 4861
POWER statement (GLMPOWER), 2659
TWOSAMPLEFREQ statement (POWER), 4886
TWOSAMPLEMEANS statement (POWER), 4895
TWOSAMPLESURVIVAL statement (POWER), 4906
TWOSAMPLEWILCOXON statement (POWER), 4912
NTRUENULL= option
PROC MULTTEST statement, 4191
NTU= option
PROC MCMC statement, 3502
NUGGET= option
MODEL statement (KRIGE2D), 2946
SIMULATE statement (SIM2D), 6174
NULLCONTRAST= option
ONEWAYANOVA statement (POWER), 4861
NULLCORR= option
ONECORR statement (POWER), 4842
NULLDIFF= option
PAIREDMEANS statement (POWER), 4874
TWOSAMPLEMEANS statement (POWER), 4895
NULLDISCPROPRATIO= option
PAIREDFREQ statement (POWER), 4866
NULLMEAN= option
ONESAMPLEMEANS statement (POWER), 4855
NULLODDSRATIO= option
TWOSAMPLEFREQ statement (POWER), 4886
NULLPROPORTION= option
ONESAMPLEFREQ statement (POWER), 4847
NULLPROPORTIONDIFF= option
TWOSAMPLEFREQ statement (POWER), 4887
NULLRATIO= option
PAIREDMEANS statement (POWER), 4874
TWOSAMPLEMEANS statement (POWER), 4895
NULLRELATIVERISK= option
TWOSAMPLEFREQ statement (POWER), 4887
NUMPOINTS= option
PREDICT statement (KRIGE2D), 2944
NUMREAL= option
PROC SIMNORMAL statement, 6204

SIMULATE statement (SIM2D), 6171
NVALS= option
OUTPUT statement (PLAN), 4740
NVARS= option
PROC CORRESP statement, 1328

OBSCAT option
OUTPUT statement (GLIMMIX), 2180
OBSERVED option
PROC CORRESP statement, 1328
OBSMARGINS option
LSMEANS statement (GLIMMIX), 2148
LSMEANS statement (GLM), 2461, 2530
LSMEANS statement (MIXED), 3927
LSMESTIMATE statement (GLIMMIX), 2159
OBSTATS option
MODEL statement (GENMOD), 1950
OCOEF option
PROC MDS statement, 3717
OCONFIG option
PROC MDS statement, 3717
OCRIT option
PROC MDS statement, 3717
ODDS option
LSMEANS statement (GLIMMIX), 2148
ODDSRATIO option
LSMEANS statement (GLIMMIX), 2148
MODEL statement (GLIMMIX), 2173
PROC GLIMMIX statement, 2109
ODDSRATIO statement
LOGISTIC procedure, 3303
ODDSRATIO= option
PAIREDFREQ statement (POWER), 4866
TWOSAMPLEFREQ statement (POWER), 4887
ODS destination statement
FILE= option, 536
IMAGE_DPI= option, 532
STYLE= option, 532
ODS DOCUMENT statement, 607
ODS EXCLUDE statement, 439, 540
PERSIST option, 455
ODS Graphics
PLOTS= option, 533
ODS GRAPHICS statement
ANTIALIAS= option, 529
ANTIALIASMAX= option, 529
BORDER= option, 530
HEIGHT= option, 530
IMAGEFMT= option, 530
IMAGEMAP= option, 530
IMAGENAME= option, 530
LABELMAX= option, 530
MAXLEGENDAREA= option, 530
RESET= option, 530
SCALE= option, 531
TIPMAX= option, 531
WIDTH= option, 531
ODS HTML statement, 449
BODY= option, 532
CONTENTS= option, 532
FRAME= option, 532
GPATH= option, 545
NEWFILE= option, 484
PATH= option, 545
URL= suboption, 545
ODS LATEX statement
STYLE= option, 604
ODS OUTPUT statement, 455
data set options, 459
PERSIST option, 461
ODS PATH statement, 445
RESET option, 588
SHOW option, 587
ODS PDF statement
FILE= option, 546
ID= option, 546
ODS RTF statement, 603
ODS SELECT statement, 438, 540
ODS SHOW statement, 452
ODS TRACE statement, 436, 538
LABEL option, 539
LISTING option, 437, 539
OFFLIST= option
FITINDEX statement, 6760
OFFSET= option
MODEL statement (GENMOD), 1952
MODEL statement (GLIMMIX), 2174
MODEL statement (LIFEREG), 3025
MODEL statement (LOGISTIC), 3298
MODEL statement (PHREG), 4569
MODEL statement (SURVEYLOGISTIC), 6400
OLS option
PARMS statement (MIXED), 3945
OM option
LSMEANS statement (GLIMMIX), 2148
LSMEANS statement (GLM), 2461, 2530
LSMESTIMATE statement (GLIMMIX), 2159
OM= option
PROC CALIS statement, 867
PROC TCALIS statement, 6736
OMETHOD= option
NLOPTIONS statement (CALIS), 889
PROC CALIS statement, 867
PROC TCALIS statement, 6736

ONECORR statement
POWER procedure, 4840
ONESAMPLEFREQ statement
POWER procedure, 4844
ONESAMPLEMEANS statement
POWER procedure, 4852
ONESIDED option
EXACT statement (LOGISTIC), 3286
ONEWAY option
MODEL statement (CATMOD), 1124
ONEWAYANOVA statement
POWER procedure, 4859
ONLIST= option
FITINDEX statement, 6761
OPSCORE transformation
MODEL statement (TRANSREG), 7146
TRANSFORM statement (PRINQUAL), 5215
OPTC option
PROC PROBIT statement, 5264, 5266
OPTCHECK option
PROC NLMIXED statement, 4369
optimization statements, TCALIS procedure, 6718
OPTION statement
QUANTREG procedure, 5379
options
CDFPLOT statement (PROBIT), 5269
IPPPLOT statement (PROBIT), 5280
LPREDPLOT statement (PROBIT), 5287
PREDPPLOT statement (PROBIT), 5301
OR option
EXACT statement (FREQ), 1692, 1812
OUTPUT statement (FREQ), 1696
TABLES statement (SURVEYFREQ), 6318
ORD option
PROC MIXED statement, 3910
ORDER= option
CLASS statement (GAM), 1841
CLASS statement (GENMOD), 1938
CLASS statement (GLMSELECT), 2703
CLASS statement (LOGISTIC), 3280
CLASS statement (PHREG), 4557
CLASS statement (SURVEYLOGISTIC), 6389
CLASS statement (SURVEYREG), 6544
MODEL statement, 1844, 2163, 3289, 6396
MODEL statement (MIXED), 3986
MODEL statement (TRANSREG), 7154, 7165
OUTPUT statement (PHREG), 4575
PROC ANOVA statement, 702
PROC CATMOD statement, 1109
PROC FREQ statement, 1690
PROC GAM statement, 1838
PROC GENMOD statement, 1922
PROC GLIMMIX statement, 2110
PROC GLM statement, 2447, 2511
PROC GLMMOD statement, 2626
PROC GLMPOWER statement, 2651
PROC HPMIXED statement, 2814
PROC LIFEREG statement, 3006
PROC LOGISTIC statement, 3271
PROC MIXED statement, 3910, 3982
PROC MULTTEST statement, 4193, 4236
PROC ORTHOREG statement, 4510
PROC PROBIT statement, 5266
PROC QUANTREG statement, 5374
PROC ROBUSTREG statement, 5661
PROC SEQTEST statement, 6008
PROC SURVEYFREQ statement, 6303
PROC SURVEYMEANS statement, 6475
PROC TTEST statement, 7402
VAR statement, 1504
ORDERALL option
PROC TCALIS statement, 6738
ORDERED option
OUTPUT statement (PLAN), 4741
PROC PLAN statement, 4736
ORDERGROUPS option
PROC TCALIS statement, 6738
ORDERMODELS option
PROC TCALIS statement, 6738
ORDERSELECT option
MODEL statement (GLMSELECT), 2709
ORDERSPEC option
PROC TCALIS statement, 6738
ORIGINAL option
MODEL statement (TRANSREG), 7149
TRANSFORM statement (PRINQUAL), 5217
ORTH option
MANOVA statement (ANOVA), 708
MANOVA statement (GLM), 2466
ORTHOGONAL option
MODEL statement (TRANSREG), 7154
ORTHOREG procedure
syntax, 4509
ORTHOREG procedure, BY statement, 4511
ORTHOREG procedure, CLASS statement, 4511
TRUNCATE option, 4511
ORTHOREG procedure, MODEL statement, 4512
NOINT option, 4512
ORTHOREG procedure, PROC ORTHOREG statement, 4509
DATA= option, 4509
NOPRINT option, 4510

ORDER= option, 4510
OUTEST= option, 4510
SINGULAR= option, 4510
ORTHOREG procedure, WEIGHT statement, 4512
OTRANS option
PROC MDS statement, 3717
OUT= option
BASELINE statement (PHREG), 4541
BIVAR statement, 2892
EM statement (MI), 3755
LSMEANS statement (GLM), 2462
OUTPUT statement (FREQ), 1694
OUTPUT statement (GAM), 1846
OUTPUT statement (GENMOD), 1955
OUTPUT statement (GLIMMIX), 2177
OUTPUT statement (GLM), 2479
OUTPUT statement (GLMSELECT), 2715
OUTPUT statement (HPMIXED), 2824
OUTPUT statement (LIFEREG), 3026
OUTPUT statement (LOGISTIC), 3307
OUTPUT statement (NLIN), 4286
OUTPUT statement (NPAR1WAY), 4465
OUTPUT statement (PHREG), 4573
OUTPUT statement (PLAN), 4740
OUTPUT statement (QUANTREG), 5381
OUTPUT statement (REG), 5482
OUTPUT statement (ROBUSTREG), 5670
OUTPUT statement (SURVEYLOGISTIC), 6403
OUTPUT statement (SURVEYREG), 6556
OUTPUT statement (TPSPLINE), 7072
OUTPUT statement (TRANSREG), 7168
PRIOR statement (MIXED), 3948
PROC ACECLUS statement, 676
PROC CANCORR statement, 1040
PROC CANDISC statement, 1070
PROC DISCRIM statement, 1394
PROC DISTANCE statement, 1498
PROC FACTOR statement, 1555
PROC FASTCLUS statement, 1638
PROC MDS statement, 3718
PROC MI statement, 3753
PROC MODECLUS statement, 4107
PROC MULTTEST statement, 4194, 4220
PROC PRINCOMP statement, 5152
PROC PRINQUAL statement, 5207
PROC RSREG statement, 5727
PROC SCORE statement, 5766
PROC SIMNORMAL statement, 6204
PROC STDIZE statement, 6223
PROC SURVEYSELECT statement, 6632
PROC TREE statement, 7366
RANDOM statement (NLMIXED), 4379
RESPONSE statement (CATMOD), 1132
SCORE statement (GAM), 1847
SCORE statement (GLMSELECT), 2718
SCORE statement (LOGISTIC), 3313
SCORE statement (TPSPLINE), 7072
TABLES statement (FREQ), 1711
UNIVAR statement, 2894
OUTACWEIGHTS= option
PROC VARIOGRAM statement, 7529
OUTALL option
PROC SURVEYSELECT statement, 6633
OUTBOX= option
BOXPLOT procedure, 785
OUTC= option
PROC CORRESP statement, 1328
OUTCLUS= option
PROC MODECLUS statement, 4107
OUTCOND option
PROC SIMNORMAL statement, 6204
OUTCOV= option
PROC INBREED statement, 2865
OUTCROSS= option
PROC DISCRIM statement, 1394
OUTCUM option
TABLES statement (FREQ), 1711
OUTD= option
PROC DISCRIM statement, 1394
OUTDESIGN option
PROC GLIMMIX statement, 2111
OUTDESIGN= option
PROC GLMMOD statement, 2627, 2631
PROC GLMSELECT statement, 2696
PROC LOGISTIC statement, 3271
OUTDESIGNONLY option
PROC LOGISTIC statement, 3272
OUTDIST= option
EXACT statement (LOGISTIC), 3286
OUTDISTANCE= option
PROC VARIOGRAM statement, 7529
OUTEM= option
EM statement (MI), 3756
OUTEST= option
MCMC statement (MI), 3760
PROC CALIS statement, 861
PROC KRIGE2D statement, 2937
PROC LIFEREG statement, 3007
PROC LOGISTIC statement, 3272
PROC NLIN statement, 4282
PROC ORTHOREG statement, 4510
PROC PHREG statement, 4537
PROC PROBIT statement, 5266
PROC QUANTREG statement, 5375
PROC REG statement, 5455
PROC ROBUSTREG statement, 5661

PROC TCALIS statement, 6738
RESPONSE statement (CATMOD), 1132
OUTEXPECT option
TABLES statement (FREQ), 1711, 1800
OUTF= option
PROC CORRESP statement, 1328
OUTFILE statement
TCALIS procedure, 6796
OUTFILES statement
TCALIS procedure, 6796
OUTFIT option
PROC TCALIS statement, 6739
OUTFIT= option
FITINDEX statement, 6763
PROC MDS statement, 3718
OUTG= option
PRIOR statement (MIXED), 3948
OUTGT= option
PRIOR statement (MIXED), 3948
OUTHIGHHTML= option
BOXPLOT procedure, 786
OUTHISTORY= option
BOXPLOT procedure, 786
OUTHITS option
PROC SURVEYSELECT statement, 6633
OUTITER option
PROC FASTCLUS statement, 1638
PROC MDS statement, 3717
OUTITER= option
EM statement (MI), 3756
MCMC statement (MI), 3760
OUTJAC option
PROC CALIS statement, 862
OUTJKCOEFS= option
VARMETHOD=JK (PROC SURVEYFREQ statement), 6308
VARMETHOD=JK (PROC SURVEYLOGISTIC statement), 6388
VARMETHOD=JK (PROC SURVEYMEANS statement), 6483
VARMETHOD=JK (PROC SURVEYREG statement), 6548
OUTLENGTH= option
PROC MODECLUS statement, 4107
OUTLIER keyword
OUTPUT statement (QUANTREG), 5382
OUTPUT statement (ROBUSTREG), 5670
OUTLOWHTML= option
BOXPLOT procedure, 786
OUTMODEL= option
PROC LOGISTIC statement, 3272
OUTNBHD= option
PROC KRIGE2D statement, 2937
OUTP= option
MODEL statement (MIXED), 4032
OUTPAIR= option
PROC VARIOGRAM statement, 7529
OUTPARM= option
PROC GLMMOD statement, 2627, 2630
OUTPCT option
TABLES statement (FREQ), 1711
OUTPDISTANCE= option
COMPUTE statement (VARIOGRAM), 7538
OUTPERM= option
PROC MULTTEST statement, 4194, 4221, 4225
OUTPOST= option
BAYES statement(PHREG), 4549
PROC MCMC statement, 3502
OUTPRED= option
MODEL statement (MIXED), 3940
OUTPREDM= option
MODEL statement (MIXED), 3941
output data sets
VARIOGRAM procedure, 7518
OUTPUT statement
FREQ procedure, 1694
GAM procedure, 1846
GENMOD procedure, 1954
GLIMMIX procedure, 2176
GLM procedure, 2477
GLMSELECT procedure, 2714
HPMIXED procedure, 2824
LIFEREG procedure, 3025
LOGISTIC procedure, 3305
NLIN procedure, 4286
NPAR1WAY procedure, 4465
PHREG procedure, 4573
PLAN procedure, 4739
PLS procedure, 4783
QUANTREG procedure, 5381
REG procedure, 5482
ROBUSTREG procedure, 5670
SURVEYLOGISTIC procedure, 6402
SURVEYREG procedure, 6556
TPSPLINE procedure, 7072
TRANSREG procedure, 7168
OUTPUTORDER= option
LOGISTIC statement (POWER), 4831
MULTREG statement (POWER), 4838
ONECORR statement (POWER), 4842
ONESAMPLEFREQ statement (POWER), 4848
ONESAMPLEMEANS statement (POWER), 4855
ONEWAYANOVA statement (POWER), 4861

PAIREDFREQ statement (POWER), 4866
PAIREDMEANS statement (POWER), 4874
POWER statement (GLMPOWER), 2659
TWOSAMPLEFREQ statement (POWER), 4887
TWOSAMPLEMEANS statement (POWER), 4895
TWOSAMPLESURVIVAL statement (POWER), 4906
TWOSAMPLEWILCOXON statement (POWER), 4913
OUTQ= option
PROC NLMIXED statement, 4369
OUTR= option
RIDGE statement (RSREG), 5734
OUTRAM= option
PROC CALIS statement, 862
PROC TCALIS statement, 6739
OUTRES= option
PROC MDS statement, 3718
OUTROC= option
MODEL statement (LOGISTIC), 3298
SCORE statement (LOGISTIC), 3313
OUTS= option
PROC FASTCLUS statement, 1638
OUTSAMP= option
PROC MULTTEST statement, 4194, 4222, 4229
OUTSDZ= option
PROC DISTANCE statement, 1498
OUTSEB option
MODEL statement (REG), 5476
PROC REG statement, 5455
OUTSEED option
PROC SIMNORMAL statement, 6204
PROC SURVEYSELECT statement, 6633
OUTSEED= option
PROC FASTCLUS statement, 1638
OUTSIM= option
PROC SIM2D statement, 6166
OUTSIZE option
PROC SURVEYSELECT statement, 6633
OUTSORT= option
PROC SURVEYSELECT statement, 6634
OUTSSCP= option
PROC REG statement, 5455
OUTSTAT= option
PROC ACECLUS statement, 677
PROC ANOVA statement, 703
PROC CALIS statement, 862
PROC CANCORR statement, 1040
PROC CANDISC statement, 1071
PROC DISCRIM statement, 1394
PROC FACTOR statement, 1555
PROC FASTCLUS statement, 1638
PROC GLM statement, 2448
PROC PRINCOMP statement, 5152
PROC STDIZE statement, 6224
PROC TCALIS statement, 6739
PROC VARCLUS statement, 7472
OUTSTB option
MODEL statement (REG), 5476
PROC REG statement, 5456
OUTSUM= option
PROC MODECLUS statement, 4107
OUTSURV= option
PROC LIFETEST statement, 3117
OUTTEST= option
PROC LIFETEST statement, 3117
PROC TRANSREG statement, 7133
OUTTREE= option
PROC CLUSTER statement, 1244
PROC VARCLUS statement, 7472
OUTVAR= option
PROC CALIS statement, 861
PROC TCALIS statement, 6738
PROC VARIOGRAM statement, 7530
OUTVIF option
MODEL statement (REG), 5476
PROC REG statement, 5456
OUTWEIGHTS= option
VARMETHOD=BRR (PROC SURVEYFREQ statement), 6307
VARMETHOD=BRR (PROC SURVEYLOGISTIC statement), 6387
VARMETHOD=BRR (PROC SURVEYMEANS statement), 6481
VARMETHOD=BRR (PROC SURVEYREG statement), 6547
VARMETHOD=JK (PROC SURVEYFREQ statement), 6308
VARMETHOD=JK (PROC SURVEYLOGISTIC statement), 6388
VARMETHOD=JK (PROC SURVEYMEANS statement), 6483
VARMETHOD=JK (PROC SURVEYREG statement), 6548
OUTWGT= option
PROC CALIS statement, 862
PROC TCALIS statement, 6739
OVER= option
PROC MDS statement, 3718
OVERLAP= option
DESIGN statement, 5804
PROC SEQTEST statement, 6004
OVERLAY option
PLOT statement (REG), 5496, 5498

OVERLAY= option
PLOT statement (BOXPLOT), 786
OVERLAYCLIPSYM= option
BOXPLOT procedure, 786
OVERLAYCLIPSYMHT= option
BOXPLOT procedure, 786
OVERLAYHTML= option
PLOT statement (BOXPLOT), 786
OVERLAYID= option
BOXPLOT procedure, 786
OVERLAYLEGLAB= option
PLOT statement (BOXPLOT), 786
OVERLAYSYM= option
PLOT statement (BOXPLOT), 786
OVERLAYSYMHT= option
PLOT statement (BOXPLOT), 787

P option
MODEL statement (GLM), 2476
MODEL statement (REG), 5476
P= option
PROC ACECLUS statement, 677
PAGE option
PROC FREQ statement, 1690
PROC SURVEYFREQ statement, 6303
PAGENUM= option
PLOT statement (BOXPLOT), 787
PAGENUMPOS= option
PLOT statement (BOXPLOT), 787
PAGES= option
PROC TREE statement, 7366
PAINT statement
REG procedure, 5484
PAIRED statement
TTEST procedure, 7409
PAIREDCVS= option
PAIREDMEANS statement (POWER), 4875
PAIREDFREQ statement
POWER procedure, 4864
PAIREDMEANS statement
POWER procedure, 4871
PAIREDMEANS= option
PAIREDMEANS statement (POWER), 4875
PAIREDPROPORTIONS= option
PAIREDFREQ statement (POWER), 4867
PAIREDSTDDEVS= option
PAIREDMEANS statement (POWER), 4875
PALL option
NLOPTIONS statement (CALIS), 896
PROC CALIS statement, 863
PROC TCALIS statement, 6740

PARAM= option
CLASS statement (GENMOD), 1939
CLASS statement (GLMSELECT), 2704
CLASS statement (LOGISTIC), 3281
CLASS statement (PHREG), 4558
CLASS statement (SURVEYLOGISTIC), 6390
MODEL statement (CATMOD), 1124
PARAMETER= option
MODEL statement (TRANSREG), 7149
TRANSFORM statement (PRINQUAL), 5217
PARAMETERESTIMATES
DETAILS=STEPS option (GLMSELECT), 2708
PARAMETERS statement
CALIS procedure, 898
NLIN procedure, 4289
TCALIS procedure, 6798
PARENT statement
TREE procedure, 7369
PARM_PREFIX option
REFMODEL statement, 6812
PARM_SUFFIX option
REFMODEL statement, 6812
PARMINFO= option
PROC MIANALYZE statement, 3847
PARMLABEL option
MODEL statement (LOGISTIC), 3298
MODEL statement (SURVEYLOGISTIC), 6400
MODEL statement (SURVEYREG), 6556
PARMNAME option
PROC TCALIS statement, 6740
PARMS option
COVTEST statement (GLIMMIX), 2133
PARMS statement
GLIMMIX procedure, 2181
HPMIXED procedure, 2826
MCMC procedure, 3515
MIXED procedure, 3943, 4032
NLMIXED procedure, 4376
PARMS= option
OUTPUT statement (NLIN), 4287
PROC MIANALYZE statement, 3847
PROC SEQTEST statement, 6009
PARMSDATA= option
PARMS statement (GLIMMIX), 2184
PARMS statement (HPMIXED), 2828
PARMS statement (MIXED), 3945
PARPREFIX= option
PROC CANCORR statement, 1041
PROC FACTOR statement, 1555
PROC PRINCOMP statement, 5155

PARTIAL option
MODEL statement (REG), 5476
PARTIAL statement
CALIS procedure, 899
FACTOR procedure, 1565
PRINCOMP procedure, 5157
TCALIS procedure, 6798
VARCLUS procedure, 7475
PARTIALCORR= option
MULTREG statement (POWER), 4838
PARTIALDATA option
MODEL statement (REG), 5477
PARTIALR2 option
MODEL statement (REG), 5477
PARTITION statement
GLMSELECT procedure, 2715
PATH statement, TCALIS procedure, 6799
PATH= option
ODS HTML statement, 545
PBO option
MODEL statement (TRANSREG), 7165
PBSPLINE transformation
MODEL statement (TRANSREG), 7145
PC option
MODEL statement (REG), 5477
PLOT statement (REG), 5496
PCHI option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1696, 1815
PCOEF option
PROC MDS statement, 3718
PCOMIT= option
MODEL statement (REG), 5477
PROC REG statement, 5456
PCONFIG option
PROC MDS statement, 3718
PCONV option
PROC GLIMMIX statement, 2111
PCORR option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1696
PROC CALIS statement, 863
PROC CANCORR statement, 1040
PROC CANDISC statement, 1071
PROC DISCRIM statement, 1394
PROC STEPDISC statement, 6276
PROC TCALIS statement, 6740
TEST statement (FREQ), 1723
PCORR1 option
MODEL statement (REG), 5477
PCORR2 option
MODEL statement (REG), 5477
PCOV option
PROC CANDISC statement, 1071
PROC DISCRIM statement, 1394
PROC STEPDISC statement, 6276
PCOV statement, TCALIS procedure, 6802
PCOVES option
PROC CALIS statement, 863
PROC TCALIS statement, 6740
PCRPJAC option
NLOPTIONS statement (CALIS), 896
PCTLDEF= option
PLOT statement (BOXPLOT), 787
PROC STDIZE statement, 6224
PCTLMTD option
PROC STDIZE statement, 6224
PCTLPTS option
PROC STDIZE statement, 6224
PDATA option
PROC MDS statement, 3718
PDATA= option
PARMS statement (GLIMMIX), 2184
PARMS statement (HPMIXED), 2828
PARMS statement (MIXED), 3945
PROC MULTTEST statement, 4194
PDETERM option
PROC CALIS statement, 863
PROC TCALIS statement, 6740
PDIFF option
LSMEANS statement (GLIMMIX), 2147, 2149
LSMEANS statement (GLM), 2462
LSMEANS statement (MIXED), 3926, 3927
PEARSON= option
OUTPUT statement (HPMIXED), 2824
PENALTY= option
PROC CLUSTER statement, 1244
PERCENT= option
PROC ACECLUS statement, 677
PROC VARCLUS statement, 7473
PERCENTILE= option
PROC SURVEYMEANS statement, 6475
PERFORMANCE statement
GLMSELECT procedure, 2716
QUANTREG procedure, 5382
ROBUSTREG procedure, 5671
PERMUTATION option
PROC MULTTEST statement, 4194, 4213, 4225, 4236
PERMUTATION= option
TEST statement (MULTTEST), 4202, 4204, 4225
PESTIM option
PROC CALIS statement, 864
PROC TCALIS statement, 6741
PETO option

TEST statement (MULTTEST), 4201, 4207, 4233
PEVENT= option
MODEL statement (LOGISTIC), 3298
PFDR option
PROC MULTTEST statement, 4194, 4218
PFINAL option
PROC MDS statement, 3718
PFIT option
PROC MDS statement, 3718
PFITROW option
PROC MDS statement, 3718
PH option, *see* PROPORTIONALHAZARDS option
PHESSIAN option
NLOPTIONS statement (CALIS), 896
PHI option
OUTPUT statement (FREQ), 1696
PHISTORY option
NLOPTIONS statement (CALIS), 896
PHREG procedure
syntax, 4535
PHREG procedure, ASSESS statement, 4540
CRPANEL option, 4540
NPATHS= option, 4540
PROPORTIONALHAZARDS option, 4540
RESAMPLE= option, 4540
SEED= option, 4540
VAR= option, 4540
PHREG procedure, BASELINE statement, 4541
ALPHA= option, 4544
CLTYPE= option, 4544
COVARIATES= option, 4541
GROUP= option, 4545
keyword= option, 4541
METHOD= option, 4545
OUT= option, 4541
ROWID= option, 4545
TIMELIST= option, 4541
PHREG procedure, BAYES statement, 4545
COEFFPRIOR= option, 4546
DIAGNOSTIC= option, 4547
INITIAL= option, 4549
NBI= option, 4549
NMC= option, 4549
OUTPOST= option, 4549
PIECEWISE= option, 4549
PLOTS= option, 4552
SEED= option, 4555
STATISTICS= option, 4555
THINNING= option, 4556
PHREG procedure, BY statement, 4556
PHREG procedure, CLASS statement, 4557
CPREFIX= option, 4557
DESCENDING option, 4557
LPREFIX= option, 4557
MISSING option, 4557
ORDER= option, 4557
PARAM= option, 4558
REF= option, 4558
PHREG procedure, CONTRAST statement, 4559
ALPHA= option, 4561
E option, 4561
ESTIMATE= option, 4561
SINGULAR= option, 4561
PHREG procedure, FREQ statement, 4561
NOTRUNCATE option, 4562
PHREG procedure, HAZARDRATIO statement, 4562
ALPHA= option, 4562
AT= option, 4562
CL= option, 4563
DIFF= option, 4563
E option, 4563
PLCONV= option, 4563
PLMAXIT= option, 4563
PLSINGULAR= option, 4563
UNITS= option, 4563
PHREG procedure, ID statement, 4564
PHREG procedure, MODEL statement, 4564
ALPHA= option, 4566
BEST= option, 4566
CORRB option, 4566
COVB option, 4566
DETAILS option, 4566
ENTRYTIME= option, 4567
FCONV= option, 4567
FIRTH option, 4567
GCONV= option, 4567
HIERARCHY= option, 4567
INCLUDE= option, 4568
ITPRINT option, 4568
MAXITER= option, 4568
MAXSTEP= option, 4568
NODESIGNPRINT= option, 4569
NODUMMYPRINT= option, 4569
NOFIT option, 4569
OFFSET= option, 4569
PLCONV= option, 4569
RIDGEINIT= option, 4569
RIDGING= option, 4569
RISKLIMITS= option, 4569
SELECTION= option, 4570
SEQUENTIAL option, 4570
SINGULAR= option, 4571
SLENTRY= option, 4571
SLSTAY= option, 4571
START= option, 4571

STOP= option, 4571
STOPRES option, 4571
TIES= option, 4572
PHREG procedure, OUTPUT statement, 4573
keyword= option, 4573
METHOD= option, 4575
ORDER= option, 4575
OUT= option, 4573
PHREG procedure, PROC PHREG statement, 4536
ALPHA= option, 4536
COVM option, 4536
COVOUT option, 4536
COVSANDWICH option, 4536
DATA= option, 4537
INEST= option, 4537
MULTIPASS option, 4537
NAMELEN= option, 4537
NOPRINT option, 4537
NOSUMMARY option, 4537
OUTEST= option, 4537
PLOTS= option, 4537
SIMPLE option, 4539
PHREG procedure, STRATA statement, 4577
MISSING option, 4577
PHREG procedure, TEST statement, 4577
AVERAGE option, 4578
E option, 4578
PRINT option, 4578
PHREG procedure, WEIGHT statement, 4578
NORMALIZE option, 4579
PIECEWISE= option
BAYES statement(PHREG), 4549
PINAVDATA option
PROC MDS statement, 3718
PINEIGVAL option
PROC MDS statement, 3718
PINEIGVEC option
PROC MDS statement, 3719
PININ option
PROC MDS statement, 3719
PINIT option
NLOPTIONS statement (CALIS), 896
PROC MDS statement, 3719
PINITIAL option
PROC CALIS statement, 864
PROC TCALIS statement, 6741
PITER option
PROC MDS statement, 3719
PJACPAT option
PROC CALIS statement, 864
PJTJ option
NLOPTIONS statement (CALIS), 896
PLAN procedure
factor-value-setting specification, 4740, 4741
syntax, 4736
PLAN procedure, FACTOR statement
NOPRINT option, 4737
PLAN procedure, FACTORS statement, 4737
PLAN procedure, OUTPUT statement, 4739
CVALS= option, 4740
factor-value-settings option, 4740
NVALS= option, 4740
ORDERED option, 4741
OUT= option, 4740
RANDOM option, 4741
PLAN procedure, PROC PLAN statement, 4736
ORDERED option, 4736
SEED option, 4736
PLAN procedure, TREATMENTS statement, 4742
PLATCOV option
PROC CALIS statement, 864
PROC TCALIS statement, 6741
PLCL option
MODEL statement (LOGISTIC), 3299
PLCONV= option
HAZARDRATIO statement (PHREG), 4563
MODEL statement (LOGISTIC), 3299
MODEL statement (PHREG), 4569
ODDSRATIO statement (LOGISTIC), 3304
PLCORR option
OUTPUT statement (FREQ), 1696
TABLES statement (FREQ), 1711
PLMAXIT= option
HAZARDRATIO statement (PHREG), 4563
PLMAXITER= option
ODDSRATIO statement (LOGISTIC), 3304
PLOT option
LSMEANS statement (GLIMMIX), 2149
PROC FACTOR statement, 1555
PROC GLIMMIX statement, 2112
PROC GLMSELECT statement, 2698
PROC LOESS statement, 3202
PROC MIXED statement, 3911
PROC REG statement, 5456
PLOT statement
BOXPLOT procedure, 767
GLMPOWER procedure, 2654
POWER procedure, 4880
REG procedure, 5487
PLOT= option
PROC PROBIT statement, 5267
PROC ROBUSTREG statement, 5661
PLOTONLY= option
PROC GLMPOWER statement, 2651
PROC POWER statement, 4827

PLOTREF option
PROC FACTOR statement, 1555
PLOTS option
KRIGE2D procedure, PROC KRIGE2D statement, 2937
LSMEANS statement (GLIMMIX), 2149
PROC CLUSTER statement, 1244
PROC GLIMMIX statement, 2112
PROC GLMSELECT statement, 2698
PROC LOESS statement, 3202
PROC LOGISTIC statement, 3272
PROC MIXED statement, 3911
PROC REG statement, 5456
PROC SEQDESIGN statement, 5802
PROC SEQTEST statement, 6011
PROC TTEST statement, 7403
SIM2D procedure, PROC SIM2D statement, 6166
VARIOGRAM procedure, PROC VARIOGRAM statement, 7530
PLOTS(ONLY) option
KRIGE2D procedure, PROC KRIGE2D statement, 2938
SIM2D procedure, PROC SIM2D statement, 6166
VARIOGRAM procedure, PROC VARIOGRAM statement, 7530
PLOTS(UNPACKPANEL) option
VARIOGRAM procedure, PROC VARIOGRAM statement, 7530
PLOTS= option
BAYES statement(PHREG), 4552
BIVAR statement, 2892, 2905
ODS Graphics, 533
PROC ANOVA statement, 703
PROC CALIS statement, 864
PROC CORRESP statement, 1328
PROC FACTOR statement, 1555
PROC GAM statement, 1839
PROC GENMOD statement, 1922
PROC GLM statement, 2448
PROC LIFETEST statement, 3118, 3122
PROC MCMC statement, 3503
PROC MULTTEST statement, 4195
PROC NPAR1WAY statement, 4459
PROC PHREG statement, 4537
PROC PLS statement, 4778
PROC PRINCOMP statement, 5153
PROC PRINQUAL statement, 5207
PROC RSREG statement, 5727
PROC TCALIS statement, 6741
PROC TRANSREG statement, 7133
TABLES statement (FREQ), 1712
UNIVAR statement, 2894, 2905
PLOTS=ALL option
KRIGE2D procedure, PROC KRIGE2D statement, 2938
SIM2D procedure, PROC SIM2D statement, 6167
VARIOGRAM procedure, PROC VARIOGRAM statement, 7531
PLOTS=EQUATE option
KRIGE2D procedure, PROC KRIGE2D statement, 2938
SIM2D procedure, PROC SIM2D statement, 6167
VARIOGRAM procedure, PROC VARIOGRAM statement, 7531
PLOTS=MEANS option
SIM2D procedure, PROC SIM2D statement, 6168
PLOTS=NONE option
KRIGE2D procedure, PROC KRIGE2D statement, 2938
SIM2D procedure, PROC SIM2D statement, 6167
VARIOGRAM procedure, PROC VARIOGRAM statement, 7531
PLOTS=OBSERVATIONS option
KRIGE2D procedure, PROC KRIGE2D statement, 2938
SIM2D procedure, PROC SIM2D statement, 6167
VARIOGRAM procedure, PROC VARIOGRAM statement, 7531
PLOTS=PAIRS option
VARIOGRAM procedure, PROC VARIOGRAM statement, 7532
PLOTS=PREDICTION option
KRIGE2D procedure, PROC KRIGE2D statement, 2939
PLOTS=SEMIVARIOGRAM option
VARIOGRAM procedure, PROC VARIOGRAM statement, 7532
PLRL option
MODEL statement (LOGISTIC), 3299
PLS procedure
syntax, 4775
PLS procedure, BY statement, 4781
PLS procedure, CLASS statement, 4782
TRUNCATE option, 4782
PLS procedure, ID statement, 4782
PLS procedure, MODEL statement, 4782
INTERCEPT option, 4783
SOLUTION option, 4783
PLS procedure, OUTPUT statement, 4783
PLS procedure, PROC PLS statement, 4775
ALGORITHM= option, 4776

CENSCALE option, 4775
CV= option, 4775
CVTEST= option, 4776
DATA= option, 4776
DETAILS option, 4776
METHOD= option, 4776
MISSING= option, 4777
NFAC= option, 4777
NITER= option, 4775
NOCENTER option, 4777
NOCVSTDIZE option, 4777
NOPRINT option, 4778
NOSCALE option, 4778, 4781
NTEST= option, 4775
PLOTS= option, 4778
PVAL= option, 4776
SEED= option, 4775, 4776
STAT= option, 4776
VARSCALE option, 4781
PLS procedure, PROC PLS statement, METHOD=PLS option
EPSILON= option, 4777
MAXITER= option, 4777
PLS procedure, PROC PLS statement, MISSING=EM option
EPSILON= option, 4777
MAXITER= option, 4777
PLSINGULAR= option
HAZARDRATIO statement (PHREG), 4563
ODDSRATIO statement (LOGISTIC), 3304
PNLCJAC option
NLOPTIONS statement (CALIS), 896
POINT option
EXACT statement (FREQ), 1693
EXACT statement (NPAR1WAY), 4464
POINT transformation
MODEL statement (TRANSREG), 7143
POLYNOMIAL keyword
REPEATED statement (ANOVA), 717
POLYNOMIAL option
REPEATED statement (GLM), 2483, 2538, 2591
POOL= option
PROC DISCRIM statement, 1394
POPULATION statement, CATMOD procedure, 1126
POS= option
PROC TREE statement, 7366
POSTERR option
PROC DISCRIM statement, 1395
POWER procedure
syntax, 4826
POWER procedure, LOGISTIC statement, 4827
ALPHA= option, 4828
COVARIATES= option, 4829
COVODDSRATIOS= option, 4829
COVREGCOEFFS= option, 4829
DEFAULTNBINS= option, 4829
DEFAULTUNIT= option, 4829
INTERCEPT= option, 4830
NBINS= option, 4830
NFRACTIONAL option, 4830
NTOTAL= option, 4830
OUTPUTORDER= option, 4831
POWER= option, 4831
RESPONSEPROB= option, 4831
TEST= option, 4831
TESTODDSRATIO= option, 4832
TESTPREDICTOR= option, 4832
TESTREGCOEFF= option, 4832
UNITS= option, 4832
VARDIST= option, 4832
POWER procedure, MULTREG statement, 4835
ALPHA= option, 4836
MODEL= option, 4836
NFRACTIONAL option, 4837
NFULLPREDICTORS= option, 4837
NOINT option, 4837
NREDUCEDPREDICTORS= option, 4837
NTESTPREDICTORS= option, 4837
NTOTAL= option, 4837
OUTPUTORDER= option, 4838
PARTIALCORR= option, 4838
POWER= option, 4838
RSQUAREDIFF= option, 4838
RSQUAREFULL= option, 4839
RSQUAREREDUCED= option, 4839
TEST= option, 4839
POWER procedure, ONECORR statement, 4840
ALPHA= option, 4841
CORR= option, 4841
DIST= option, 4841
MODEL= option, 4841
NFRACTIONAL option, 4842
NPARTIALVARS= option, 4842
NTOTAL= option, 4842
NULLCORR= option, 4842
OUTPUTORDER= option, 4842
POWER= option, 4843
SIDES= option, 4843
TEST= option, 4843
POWER procedure, ONESAMPLEFREQ statement, 4844
ALPHA= option, 4845
CI= option, 4846
EQUIVBOUNDS= option, 4846
HALFWIDTH= option, 4846
LOWER= option, 4846

MARGIN= option, 4847
METHOD= option, 4847
NFRACTIONAL= option, 4847
NTOTAL= option, 4847
NULLPROPORTION= option, 4847
OUTPUTORDER= option, 4848
POWER= option, 4848
PROBWIDTH= option, 4848
PROPORTION= option, 4848
SIDES= option, 4848
TEST= option, 4849
UPPER= option, 4849
VAREST= option, 4849
POWER procedure, ONESAMPLEMEANS statement, 4852
ALPHA= option, 4854
CI= option, 4854
CV= option, 4854
DIST= option, 4854
HALFWIDTH= option, 4854
LOWER= option, 4854
MEAN= option, 4854
NFRACTIONAL option, 4854
NTOTAL= option, 4855
NULLMEAN= option, 4855
OUTPUTORDER= option, 4855
POWER= option, 4856
PROBTYPE= option, 4856
PROBWIDTH= option, 4856
SIDES= option, 4856
STDDEV= option, 4857
TEST= option, 4857
UPPER= option, 4857
POWER procedure, ONEWAYANOVA statement, 4859
ALPHA= option, 4860
CONTRAST= option, 4860
GROUPMEANS= option, 4860
GROUPNS= option, 4860
GROUPWEIGHTS= option, 4860
NFRACTIONAL option, 4861
NPERGROUP= option, 4861
NTOTAL= option, 4861
NULLCONTRAST= option, 4861
OUTPUTORDER= option, 4861
POWER= option, 4862
SIDES= option, 4862
STDDEV= option, 4862
TEST= option, 4862
POWER procedure, PAIREDFREQ statement, 4864
ALPHA= option, 4865
CORR= option, 4865
DISCPROPDIFF= option, 4865
DISCPROPORTIONS= option, 4865
DISCPROPRATIO= option, 4865
DIST= option, 4865
METHOD= option, 4866
NFRACTIONAL option, 4866
NPAIRS= option, 4866
NULLDISCPROPRATIO= option, 4866
ODDSRATIO= option, 4866
OUTPUTORDER= option, 4866
PAIREDPROPORTIONS= option, 4867
POWER= option, 4867
PROPORTIONDIFF= option, 4867
REFPROPORTION= option, 4867
RELATIVERISK= option, 4868
SIDES= option, 4868
TEST= option, 4868
TOTALPROPDISC= option, 4868
POWER procedure, PAIREDMEANS statement, 4871
ALPHA= option, 4872
CI= option, 4873
CORR= option, 4873
CV= option, 4873
DIST= option, 4873
HALFWIDTH= option, 4873
LOWER= option, 4873
MEANDIFF= option, 4873
MEANRATIO= option, 4874
NFRACTIONAL option, 4874
NPAIRS= option, 4874
NULLDIFF= option, 4874
NULLRATIO= option, 4874
OUTPUTORDER= option, 4874
PAIREDCVS= option, 4875
PAIREDMEANS= option, 4875
PAIREDSTDDEVS= option, 4875
POWER= option, 4875
PROBTYPE= option, 4876
PROBWIDTH= option, 4876
SIDES= option, 4876
STDDEV= option, 4876
TEST= option, 4877
UPPER= option, 4877
POWER procedure, PLOT statement, 4880
DESCRIPTION= option, 4884
INTERPOL= option, 4880
KEY= option, 4880
MARKERS= option, 4881
MAX= option, 4881
MIN= option, 4881
NAME= option, 4884
NPOINTS= option, 4881
STEP= option, 4881
VARY option, 4882

X= option, 4882
XOPTS= option, 4883
Y= option, 4883
YOPTS= option, 4883
POWER procedure, PROC POWER statement, 4827
PLOTONLY= option, 4827
POWER procedure, TWOSAMPLEFREQ statement, 4884
ALPHA= option, 4885
GROUPNS= option, 4886
GROUPPROPORTIONS= option, 4886
GROUPWEIGHTS= option, 4886
NFRACTIONAL option, 4886
NPERGROUP= option, 4886
NTOTAL= option, 4886
NULLODDSRATIO= option, 4886
NULLPROPORTIONDIFF= option, 4887
NULLRELATIVERISK= option, 4887
ODDSRATIO= option, 4887
OUTPUTORDER= option, 4887
POWER= option, 4888
PROPORTIONDIFF= option, 4888
REFPROPORTION= option, 4888
RELATIVERISK= option, 4888
SIDES= option, 4888
TEST= option, 4888
POWER procedure, TWOSAMPLEMEANS statement, 4891
ALPHA= option, 4892
CI= option, 4893
CV= option, 4893
DIST= option, 4893
GROUPMEANS= option, 4893
GROUPNS= option, 4893
GROUPSTDDEVS= option, 4893
GROUPWEIGHTS= option, 4894
HALFWIDTH= option, 4894
LOWER= option, 4894
MEANDIFF= option, 4894
MEANRATIO= option, 4894
NFRACTIONAL option, 4894
NPERGROUP= option, 4895
NTOTAL= option, 4895
NULLDIFF= option, 4895
NULLRATIO= option, 4895
OUTPUTORDER= option, 4895
POWER= option, 4896
PROBTYPE= option, 4896
PROBWIDTH= option, 4896
SIDES= option, 4896
STDDEV= option, 4897
TEST= option, 4897
UPPER= option, 4897
POWER procedure, TWOSAMPLESURVIVAL statement, 4901
ACCRUALTIME= option, 4902
ALPHA= option, 4902
CURVE= option, 4902
FOLLOWUPTIME= option, 4903
GROUPLOSS= option, 4903
GROUPLOSSEXPHAZARDS= option, 4904
GROUPMEDLOSSTIMES= option, 4904
GROUPMEDSURVTIMES= option, 4904
GROUPNS= option, 4904
GROUPSURVEXPHAZARDS= option, 4904
GROUPSURVIVAL= option, 4905
GROUPWEIGHTS= option, 4905
HAZARDRATIO= option, 4905
NFRACTIONAL option, 4905
NPERGROUP= option, 4905
NSUBINTERVAL= option, 4905
NTOTAL= option, 4906
OUTPUTORDER= option, 4906
POWER= option, 4906
REFSURVEXPHAZARD= option, 4907
REFSURVIVAL= option, 4907
SIDES= option, 4907
TEST= option, 4907
TOTALTIME= option, 4907
POWER procedure, TWOSAMPLEWILCOXON statement, 4911
ALPHA= option, 4912
GROUPNS= option, 4912
GROUPWEIGHTS= option, 4912
NBINS= option, 4912
NFRACTIONAL= option, 4912
NPERGROUP= option, 4912
NTOTAL= option, 4912
OUTPUTORDER= option, 4913
POWER= option, 4913
SIDES= option, 4913
TEST= option, 4913
VARDIST= option, 4914
VARIABLES= option, 4914
POWER statement
GLMPOWER procedure, 2657
POWER transformation
MODEL statement (TRANSREG), 7144
TRANSFORM statement (MI), 3767
TRANSFORM statement (PRINQUAL), 5214
POWER= option
LOGISTIC statement (POWER), 4831
MULTREG statement (POWER), 4838
ONECORR statement (POWER), 4843

ONESAMPLEFREQ statement (POWER), 4848
ONESAMPLEMEANS statement (POWER), 4856
ONEWAYANOVA statement (POWER), 4862
PAIREDFREQ statement (POWER), 4867
PAIREDMEANS statement (POWER), 4875
POWER statement (GLMPOWER), 2659
PROC FACTOR statement, 1557
PROC MODECLUS statement, 4107
TWOSAMPLEFREQ statement (POWER), 4888
TWOSAMPLEMEANS statement (POWER), 4896
TWOSAMPLESURVIVAL statement (POWER), 4906
TWOSAMPLEWILCOXON statement (POWER), 4913
PP option
PROC ACECLUS statement, 677
PP= option
PROC QUANTREG statement, 5377
PPREFIX option
OUTPUT statement (TRANSREG), 7177
PPROB= option
MODEL statement (LOGISTIC), 3299
PR2ENTRY= option
PROC STEPDISC statement, 6276
PR2STAY= option
PROC STEPDISC statement, 6276
PRED option
MODEL statement (GENMOD), 1952
PLOT statement (REG), 5496
PRED= option
MODEL statement (CATMOD), 1124
PREDET option
PROC CALIS statement, 865
PREDICT option
MODEL statement (CATMOD), 1124
MODEL statement (RSREG), 5733
PROC SCORE statement, 5766
PREDICT statement
NLMIXED procedure, 4377
PREDICT statement (KRIGE2D), 2943
PREDICTED keyword
OUTPUT statement (GLM), 2478
OUTPUT statement (GLMSELECT), 2715
OUTPUT statement (LIFEREG), 3026
OUTPUT statement (QUANTREG), 5382
OUTPUT statement (ROBUSTREG), 5670
OUTPUT statement (SURVEYREG), 6557
SCORE statement (GLMSELECT), 2718
PREDICTED option
MODEL statement (GENMOD), 1952
OUTPUT statement (TRANSREG), 7177
PREDICTED= option
OUTPUT statement (HPMIXED), 2824
OUTPUT statement (LOGISTIC), 3307
OUTPUT statement (NLIN), 4287
OUTPUT statement (SURVEYLOGISTIC), 6403
PREDPOWER option
PROC SEQTEST statement, 6009
PREDPPLOT statement
options summarized by function, 5301
PROBIT procedure, 5300
PREDPROBS= option
OUTPUT statement (LOGISTIC), 3307
OUTPUT statement (SURVEYLOGISTIC), 6403
PREFIX= option
MANOVA statement (ANOVA), 707
MANOVA statement (GLM), 2465
PROC ACECLUS statement, 677
PROC CANDISC statement, 1071
PROC DISTANCE statement, 1498
PROC FACTOR statement, 1558
PROC GLMMOD statement, 2627
PROC PRINCOMP statement, 5155
PROC PRINQUAL statement, 5208
PREPLOT option
PROC FACTOR statement, 1558
PREROTATE= option
PROC FACTOR statement, 1558
PRESS
STATS= option (GLMSELECT), 2714
PRESS keyword
OUTPUT statement (GLM), 2478
PRESS option
MODEL statement (REG), 5477
MODEL statement (RSREG), 5733
PROC REG statement, 5466
PRIMAT option
PROC CALIS statement, 865
PROC TCALIS statement, 6742
PRINCOMP procedure
syntax, 5150
PRINCOMP procedure, BY statement, 5156
PRINCOMP procedure, FREQ statement, 5157
PRINCOMP procedure, ID statement, 5157
PRINCOMP procedure, PARTIAL statement, 5157
PRINCOMP procedure, PROC PRINCOMP statement, 5151
COV option, 5151
COVARIANCE option, 5151

DATA= option, 5152
N= option, 5152
NOINT option, 5152
NOPRINT option, 5152
OUT= option, 5152
OUTSTAT= option, 5152
PARPREFIX= option, 5155
PLOTS= option, 5153
PPREFIX= option, 5155
PREFIX= option, 5155
SING= option, 5156
SINGULAR= option, 5156
STANDARD option, 5156
STD option, 5156
VARDEF= option, 5156
PRINCOMP procedure, VAR statement, 5158
PRINCOMP procedure, WEIGHT statement, 5158
PRINQUAL procedure
syntax, 5202
PRINQUAL procedure, BY statement, 5211
PRINQUAL procedure, FREQ statement, 5211
PRINQUAL procedure, ID statement, 5212
PRINQUAL procedure, PROC PRINQUAL statement, 5203
APPROXIMATIONS option, 5204
APREFIX= option, 5204
CCONVERGE= option, 5204
CHANGE= option, 5204
CONVERGE= option, 5204
CORRELATIONS option, 5205
COVARIANCE option, 5205
DATA= option, 5205
DUMMY option, 5205
INITITER= option, 5205
MAXITER= option, 5205
MDPREF= option, 5205
METHOD= option, 5206
MONOTONE= option, 5206
N= option, 5206
NOCHECK option, 5206
NOMISS option, 5206
NOPRINT option, 5207
OUT= option, 5207
PLOTS= option, 5207
PREFIX= option, 5208
REFRESH= option, 5209
REITERATE option, 5209
REPLACE option, 5209
SCORES option, 5209
SINGULAR= option, 5209
STANDARD option, 5209
TPREFIX= option, 5210
TSTANDARD= option, 5210
TYPE= option, 5210
UNTIE= option, 5211
PRINQUAL procedure, TRANSFORM statement
ARSIN transformation, 5214
DEGREE= option, 5218
EVENLY option, 5218
EXP transformation, 5214
IDENTITY transformation, 5216
KNOTS= option, 5219
LINEAR transformation, 5215
LOG transformation, 5214
LOGIT transformation, 5214
MONOTONE transformation, 5215
MSPLINE transformation, 5215
NAME= option, 5219
NKNOTS= option, 5219
OPSCORE transformation, 5215
ORIGINAL option, 5217
PARAMETER= option, 5217
POWER transformation, 5214
RANK transformation, 5214
REFLECT option, 5219
SM= option, 5218
SPLINE transformation, 5215
SSPLINE transformation, 5216
TSTANDARD= option, 5220
UNTIE transformation, 5216
PRINQUAL procedure, WEIGHT statement, 5220
PRINT option
MTEST statement (REG), 5482
NLOPTIONS statement (CALIS), 896
PROC CALIS statement, 865
PROC FACTOR statement, 1558
PROC NLIN statement, 4282
PROC TCALIS statement, 6742
SCORE statement (LOESS), 3214
TEST statement (LOGISTIC), 3316
TEST statement (PHREG), 4578
TEST statement (REG), 5506
TEST statement (SURVEYLOGISTIC), 6408
PRINT statement, REG procedure, 5499
PRINT= option
PROC CLUSTER statement, 1246
PROC CORRESP statement, 1329
PRINTE option
MANOVA statement (ANOVA), 708
MANOVA statement (GLM), 2466
REPEATED statement (ANOVA), 718
REPEATED statement (GLM), 2484, 2536
PRINTH option
MANOVA statement (ANOVA), 708
MANOVA statement (GLM), 2466

REPEATED statement (ANOVA), 718
REPEATED statement (GLM), 2484
VARMETHOD=BRR (PROC SURVEYFREQ statement), 6307
VARMETHOD=BRR (PROC SURVEYLOGISTIC statement), 6387
VARMETHOD=BRR (PROC SURVEYMEANS statement), 6481
VARMETHOD=BRR (PROC SURVEYREG statement), 6547
PRINTKWT option
TABLES statement (FREQ), 1717
PRINTM option
REPEATED statement (ANOVA), 718
REPEATED statement (GLM), 2484
PRINTRV option
REPEATED statement (ANOVA), 719
REPEATED statement (GLM), 2485
PRIOR statement
MCMC procedure, 3516
MIXED procedure, 3945
PRIOR= option
MCMC statement (MI), 3762
SCORE statement (LOGISTIC), 3313
PRIOREVENT= option
SCORE statement (LOGISTIC), 3313
PRIORS statement
DISCRIM procedure, 1399
FACTOR procedure, 1565
PRIORS= option
PROC FACTOR statement, 1558
PRIVEC option
PROC CALIS statement, 865
PROB option
MODEL statement (CATMOD), 1124
PROBIT, 5258
PROBIT procedure, 5258
syntax, 5263
PROBIT procedure, BY statement, 5268
PROBIT procedure, CDFPLOT statement, 5268
ANNOTATE= option, 5272
CAXIS= option, 5272
CFIT= option, 5272
CFRAME= option, 5272
CGRID= option, 5272
CHREF= option, 5273
CLABBOX= option, 5272
CTEXT= option, 5273
CVREF= option, 5273
DESCRIPTION= option, 5273
FONT= option, 5273
HAXIS= option, 5273
HEIGHT= option, 5273
HLOWER= option, 5273
HOFFSET= option, 5273
HREF= option, 5274
HREFLABELS= option, 5274
HREFLABPOS= option, 5274
HUPPER= option, 5274
INBORDER option, 5274
LEVEL option, 5274
LFIT option, 5274
LGRID option, 5274
LHREF= option, 5274
LVREF= option, 5275
NAME= option, 5275
NOFIT option, 5275
NOFRAME option, 5275
NOGRID option, 5275
NOHLABEL option, 5275
NOHTICK option, 5275
NOTHRESH option, 5275
NOVLABEL option, 5275
NOVTICK option, 5275
options, 5269
THRESHLABPOS= option, 5275
VAR= option, 5269
VAXIS= option, 5275
VAXISLABEL= option, 5276
VLOWER= option, 5276
VREF= option, 5276
VREFLABELS= option, 5276
VREFLABPOS= option, 5276
VUPPER= option, 5276
WAXIS= option, 5276
WFIT= option, 5276
WGRID= option, 5276
WREFL= option, 5276
PROBIT procedure, CLASS statement, 5277
PROBIT procedure, INSET statement, 5277, 5278
keywords, 5278
PROBIT procedure, IPPPLOT statement, 5279
ANNOTATE= option, 5282
CAXIS= option, 5282
CFIT= option, 5283
CFRAME= option, 5283
CGRID= option, 5283
CHREF= option, 5283
CTEXT= option, 5283
CVREF= option, 5283
DESCRIPTION= option, 5283
FONT= option, 5283
HAXIS= option, 5283
HEIGHT= option, 5283
HLOWER= option, 5284
HOFFSET= option, 5284
HREF= option, 5284

HREFLABELS= option, 5284
HREFLABPOS= option, 5284
HUPPER= option, 5284
INBORDER option, 5284
LFIT option, 5284
LGRID option, 5284
LHREF= option, 5285
LVREF= option, 5285
NAME= option, 5285
NOCONF option, 5285
NODATA option, 5285
NOFIT option, 5285
NOFRAME option, 5285
NOGRID option, 5285
NOHLABEL option, 5285
NOHTICK option, 5285
NOTHRESH option, 5285
NOVLABEL option, 5285
NOVTICK option, 5285
options, 5280
THRESHLABPOS= option, 5285
VAR= option, 5279
VAXIS= option, 5286
VAXISLABEL= option, 5286
VLOWER= option, 5286
VREF= option, 5286
VREFLABELS= option, 5286
VREFLABPOS= option, 5286
VUPPER= option, 5286
WAXIS= option, 5286
WFIT= option, 5287
WGRID= option, 5287
WREFL= option, 5287
PROBIT procedure, LPREDPLOT statement, 5287
ANNOTATE= option, 5290
CAXIS= option, 5290
CFIT= option, 5290
CFRAME= option, 5291
CGRID= option, 5291
CHREF= option, 5291
CTEXT= option, 5291
CVREF= option, 5291
DESCRIPTION= option, 5291
FONT= option, 5291
HAXIS= option, 5291
HEIGHT= option, 5291
HLOWER= option, 5292
HOFFSET= option, 5292
HREF= option, 5292
HREFLABELS= option, 5292
HREFLABPOS= option, 5292
HUPPER= option, 5292
INBORDER option, 5292
LEVEL option, 5292
LFIT option, 5293
LGRID option, 5293
LHREF= option, 5293
LVREF= option, 5293
NAME= option, 5293
NOCONF option, 5293
NODATA option, 5293
NOFIT option, 5293
NOFRAME option, 5293
NOGRID option, 5293
NOHLABEL option, 5293
NOHTICK option, 5293
NOTHRESH option, 5293
NOVLABEL option, 5293
NOVTICK option, 5294
options, 5287
THRESHLABPOS= option, 5294
VAR= option, 5287
VAXIS= option, 5294
VAXISLABEL= option, 5294
VLOWER= option, 5294
VREF= option, 5294
VREFLABELS= option, 5294
VREFLABPOS= option, 5294
VUPPER= option, 5295
WAXIS= option, 5295
WFIT= option, 5295
WGRID= option, 5295
WREFL= option, 5295
PROBIT procedure, MODEL statement, 5295
AGGREGATE= option, 5296
ALPHA= option, 5296
CONVERGE option, 5296
CORRB option, 5296
COVB option, 5296
DISTRIBUTION= option, 5297
HPROB= option, 5297
INITIAL option, 5297
INTERCEPT= option, 5297
INVERSECL option, 5297
ITPRINT option, 5298
MAXITER= option, 5298
NOINT option, 5298
SCALE= option, 5298
SINGULAR= option, 5299
PROBIT procedure, OUTPUT statement, 5299
PROBIT procedure, PREDPLOT statement
LEVEL option, 5306
PROBIT procedure, PREDPPLOT statement, 5300
ANNOTATE= option, 5304
CAXIS= option, 5304
CFIT= option, 5304

CFRAME= option, 5304
CGRID= option, 5304
CHREF= option, 5304
CTEXT= option, 5305
CVREF= option, 5305
DESCRIPTION= option, 5305
FONT= option, 5305
HAXIS= option, 5305
HEIGHT= option, 5305
HLOWER= option, 5305
HOFFSET= option, 5305
HREF= option, 5305
HREFLABELS= option, 5306
HREFLABPOS= option, 5306
HUPPER= option, 5305
INBORDER option, 5306
LFIT option, 5306
LGRID option, 5306
LHREF= option, 5306
LVREF= option, 5306
NAME= option, 5307
NOCONF option, 5307
NODATA option, 5307
NOFIT option, 5307
NOFRAME option, 5307
NOGRID option, 5307
NOHLABEL option, 5307
NOHTICK option, 5307
NOTHRESH option, 5307
NOVLABEL option, 5307
NOVTICK option, 5307
options, 5301
THRESHLABPOS= option, 5307
VAR= option, 5300
VAXIS= option, 5307
VAXISLABEL= option, 5308
VLOWER= option, 5308
VREF= option, 5308
VREFLABELS= option, 5308
VREFLABPOS= option, 5308
VUPPER= option, 5308
WAXIS= option, 5308
WFIT= option, 5308
WGRID= option, 5308
WREFL= option, 5308
PROBIT procedure, PROC PROBIT statement, 5264
COVOUT option, 5264
DATA= option, 5264
GOUT= option, 5265
HPROB= option, 5265
INEST= option, 5265
INVERSECL option, 5265
LACKFIT option, 5265
LOG option, 5265
LOG10 option, 5266
NAMELEN= option, 5266
NOPRINT option, 5266
OPTC option, 5264, 5266
ORDER=option, 5266
OUTEST= option, 5266
PLOT= option, 5267
XDATA= option, 5268
PROBIT procedure, WEIGHT statement, 5309
PROBPLOT statement
LIFEREG procedure, 3027
PROBT option
PROC CANCORR statement, 1040
PROBTYPE= option
ONESAMPLEMEANS statement (POWER), 4856
PAIREDMEANS statement (POWER), 4876
TWOSAMPLEMEANS statement (POWER), 4896
PROBWIDTH= option
ONESAMPLEFREQ statement (POWER), 4848
ONESAMPLEMEANS statement (POWER), 4856
PAIREDMEANS statement (POWER), 4876
TWOSAMPLEMEANS statement (POWER), 4896
PROC ACECLUS statement, *see* ACECLUS procedure
PROC ANOVA statement, *see* ANOVA procedure
PROC BOXPLOT statement, *see* BOXPLOT procedure
PROC CALIS statement, *see* CALIS procedure
PROC CANCORR statement, *see* CANCORR procedure
PROC CANDISC statement, *see* CANDISC procedure
PROC CATMOD statement, *see* CATMOD procedure
PROC CLUSTER statement, *see* CLUSTER procedure
PROC CORRESP statement, *see* CORRESP procedure
PROC DISCRIM statement, *see* DISCRIM procedure
PROC DISTANCE statement, *see* DISTANCE procedure
PROC FACTOR statement, *see* FACTOR procedure

PROC FASTCLUS statement, *see* FASTCLUS procedure
PROC FREQ statement, *see* FREQ procedure
PROC GAM statement, *see* GAM procedure
PROC GENMOD statement, *see* GENMOD procedure
PROC GLIMMIX procedure, PROC GLIMMIX statement
  SINGRES= option, 2120
PROC GLIMMIX statement, *see* GLIMMIX procedure
  GLIMMIX procedure, 2094
PROC GLM statement, *see* GLM procedure
PROC GLMMOD statement, *see* GLMMOD procedure
PROC GLMPOWER statement, *see* GLMPOWER procedure
PROC GLMSELECT statement, *see* GLMSELECT procedure
PROC HPMIXED statement, *see* HPMIXED procedure
  HPMIXED procedure, 2811
PROC INBREED statement, *see* INBREED procedure
PROC KDE statement, *see* KDE procedure
PROC KRIGE2D statement, *see* KRIGE2D procedure
PROC LATTICE statement, *see* LATTICE procedure
PROC LIFEREG statement, *see* LIFEREG procedure
PROC LIFETEST statement, *see* LIFETEST procedure
  LIFETEST procedure, 3111
PROC LOESS statement, *see* LOESS procedure
PROC LOGISTIC statement, *see* LOGISTIC procedure
PROC MDS statement, *see* MDS procedure
PROC MI statement, *see* MI procedure
PROC MIANALYZE statement, *see* MIANALYZE procedure
PROC MIXED statement, *see* MIXED procedure
PROC MODECLUS statement, *see* MODECLUS procedure
PROC MULTTEST statement, *see* MULTTEST procedure
PROC NESTED statement, *see* NESTED procedure
PROC NLIN statement, *see* NLIN procedure
PROC NPAR1WAY statement, *see* NPAR1WAY procedure
PROC ORTHOREG statement, *see* ORTHOREG procedure
PROC PHREG statement, *see* PHREG procedure
PROC PLAN statement, *see* PLAN procedure
PROC PLS statement, *see* PLS procedure
PROC POWER statement, *see* POWER procedure
PROC PRINCOMP statement, *see* PRINCOMP procedure
PROC PRINQUAL statement, *see* PRINQUAL procedure
PROC PROBIT statement, *see* PROBIT procedure
PROC QUANTREG statement, *see* QUANTREG procedure
PROC REG statement, *see* REG procedure
PROC ROBUSTREG statement, *see* ROBUSTREG procedure
PROC RSREG statement, *see* RSREG procedure
PROC SCORE statement, *see* SCORE procedure
PROC SEQDESIGN statement, *see* SEQDESIGN procedure
PROC SEQTEST statement, *see* SEQTEST procedure
PROC SIM2D statement, *see* SIM2D procedure
PROC statement
  SIMNORMAL procedure, 6202
PROC STDIZE statement, *see* STDIZE procedure
PROC STEPDISC statement, *see* STEPDISC procedure
PROC SURVEYFREQ statement, 6302, *see* SURVEYFREQ procedure
PROC SURVEYLOGISTIC statement, *see* SURVEYLOGISTIC procedure
PROC SURVEYMEANS statement, *see* SURVEYMEANS procedure
PROC SURVEYREG statement, *see* SURVEYREG procedure
PROC SURVEYSELECT statement, 6623, *see* SURVEYSELECT procedure
PROC TCALIS statement, *see* TCALIS procedure
PROC TPSPLINE statement, *see* TPSPLINE procedure
PROC TRANSREG statement, *see* TRANSREG procedure
PROC TREE statement, *see* TREE procedure
PROC TTEST
  See TTEST procedure, 7401
PROC VARCLUS statement, *see* VARCLUS procedure
PROC VARCOMP statement, *see* VARCOMP procedure
  VARCOMP procedure, 7498
PROC VARIOGRAM statement, *see* VARIOGRAM procedure

PROFILE keyword
REPEATED statement (ANOVA), 717
PROFILE option
MODEL statement (CATMOD), 1124
PROC GLIMMIX statement, 2119
REPEATED statement (GLM), 2483, 2539
PROFILE= option
FACTORS statement (CATMOD), 1115
PROC CORRESP statement, 1329
REPEATED statement (CATMOD), 1129
Programming statements
GLIMMIX procedure, 2207
PROPCOV=method
PROC MCMC statement, 3505
PROPDIST= option
PROC MCMC statement, 3506
PROPENSITY option
MONOTONE statement (MI), 3766
PROPORTION= option
ONESAMPLEFREQ statement (POWER), 4848
PROC ACECLUS statement, 677
PROC FACTOR statement, 1559
PROC VARCLUS statement, 7473
PROPORTIONALHAZARDS option
ASSESS statement (PHREG), 4540
PROPORTIONDIFF= option
PAIREDFREQ statement (POWER), 4867
TWOSAMPLEFREQ statement (POWER), 4888
PROPVARREDUCTION= option
POWER statement (GLMPOWER), 2659
PSCALE
MODEL statement (GENMOD), 1953
PSEARCH option
PRIOR statement (MIXED), 3948
PSEUDO= option
PROC CLUSTER statement, 1246
PSHORT option
PROC CALIS statement, 865
PROC TCALIS statement, 6742
PSPLINE transformation
MODEL statement (TRANSREG), 7143
PSS option
PROC SEQDESIGN statement, 5801
PROC SEQTEST statement, 6009
PSSCP option
PROC CANDISC statement, 1071
PROC DISCRIM statement, 1395
PROC STEPDISC statement, 6276
PSTAT option
PROC STDIZE statement, 6224
PSUMMARY option
PROC CALIS statement, 865
PROC TCALIS statement, 6742
PTRANS option
PRIOR statement (MIXED), 3948
PROC MDS statement, 3720
PTRUENULL= option
PROC MULTTEST statement, 4196
PVAL= option
PROC PLS statement, 4776
PVAR statement, TCALIS procedure, 6803
PWEIGHT option
PROC CALIS statement, 865
PROC TCALIS statement, 6742

Q option
RANDOM statement (GLM), 2481, 2541
QFAC option
PROC NLMIXED statement, 4369
QMAX option
PROC NLMIXED statement, 4369
QPOINT transformation
MODEL statement (TRANSREG), 7143
QPOINTS option
PROC NLMIXED statement, 4369
QQ option
PROC ACECLUS statement, 677
QSCALEFAC option
PROC NLMIXED statement, 4369
QTOL option
PROC NLMIXED statement, 4369
QUANTILE= option
PROC SURVEYMEANS statement, 6475
QUANTILES keyword
OUTPUT statement (LIFEREG), 3026
OUTPUT statement (QUANTREG), 5382
QUANTILES option
MODEL statement (QUANTREG), 5380
QUANTREG procedure, BY statement, 5377
QUANTREG procedure, CLASS statement, 5377
QUANTREG procedure, EFFECT statement, 5378
QUANTREG procedure, ID statement, 5378
QUANTREG procedure, MODEL statement, 5378
CORRB option, 5379
COVB option, 5379
CUTOFF option, 5379
DIAGNOSTICS option, 5379
ITPRINT option, 5379
LEVERAGE option, 5379
NODIAG option, 5379
NOINT option, 5379
NOSUMMARY option, 5380
PLOT= plot option, 5380
QUANTILES option, 5380

SCALE option, 5380
SINGULAR= option, 5381
QUANTREG procedure, OPTION2 statement, 5379
QUANTREG procedure, OUTPUT statement, 5381
keyword= option, 5381
LEVERAGE keyword, 5381
MAHADIST keyword, 5381
OUT= option, 5381
OUTLIER keyword, 5382
PREDICTED keyword, 5382
QUANTILES keyword, 5382
RESIDUAL keyword, 5382
ROBDIST keyword, 5382
SPLINE keyword, 5382
SRESIDUAL keyword, 5382
STD_ERR keyword, 5382
QUANTREG procedure, PERFORMANCE statement, 5382
CPUCOUNT option, 5382
DETAILS option, 5383
NOTHREADS option, 5383
THREADS option, 5383
QUANTREG procedure, PROC QUANTREG statement, 5372
ALGORITHM option, 5372
ALPHA= option, 5373
CI option, 5373
DATA= option, 5374
INEST= option, 5374
KAPPA= option, 5373
MAXIT= option, 5373
NAMELEN= option, 5374
ORDER= option, 5374
OUTEST= option, 5375
PP option, 5377
RRATIO= option, 5373
TOLERANCE= option, 5373
QUANTREG procedure, PROC statement
PLOT= plot option, 5375
QUANTREG procedure, TEST statement, 5383
QUANTREG procedure, WEIGHT statement, 5383
QUANTREG procedure,MODEL statement
SEED option, 5380

R option
MODEL statement (REG), 5477
REPEATED statement (MIXED), 3957
R= option
PROC CLUSTER statement, 1246
PROC DISCRIM statement, 1395
PROC MODECLUS statement, 4107
PROC SURVEYLOGISTIC statement, 6384
PROC SURVEYMEANS statement, 6476
PROC SURVEYREG statement, 6544
RADIUS= option
NLOPTIONS statement (CALIS), 889
PREDICT statement (KRIGE2D), 2944
PROC CALIS statement, 865
PROC FASTCLUS statement, 1632
PROC TCALIS statement, 6742
RIDGE statement (RSREG), 5734
RAM statement, CALIS procedure, 899
RAM statement, TCALIS procedure, 6805
RANDOM option
GLMSELECT procedure, PARTITION statement, 2716
OUTPUT statement (PLAN), 4741
RANDOM statement
GLIMMIX procedure, 2186
GLM procedure, 2480
HPMIXED procedure, 2829
MIXED procedure, 3949
NLMIXED procedure, 4378
RANDOM statement (GLIMMIX)
BUCKET= suboption, 2189
KNOTTYPE= suboption, 2190
NEAREST suboption, 2190
TREEINFO suboption, 2190
RANDOM= option
PROC CALIS statement, 865
PROC FACTOR statement, 1559
PROC FASTCLUS statement, 1639
PROC MDS statement, 3720
PROC TCALIS statement, 6742
PROC VARCLUS statement, 7473
RANGE option
MODEL statement (TRANSREG), 7152
RANGE= option
MODEL statement (KRIGE2D), 2946
SIMULATE statement (SIM2D), 6174
RANK transformation
MODEL statement (TRANSREG), 7144
TRANSFORM statement (PRINQUAL), 5214
RANKSCORE= option
PROC DISTANCE statement, 1498
RANUNI option
PROC MULTTEST statement, 4196
RATE= option
PROC SURVEYFREQ statement, 6303
PROC SURVEYLOGISTIC statement, 6384
PROC SURVEYMEANS statement, 6476
PROC SURVEYREG statement, 6544
RATIO option
PROC MIXED statement, 3915, 3997

PROC VARCOMP statement, 7498
RATIO statement
SURVEYMEANS procedure, 6486
RATIO= option
MODEL statement (KRIGE2D), 2946
SIMULATE statement (SIM2D), 6174
RATIOS option
PARMS statement (MIXED), 3945
RANDOM statement (MIXED), 3952
RC option
REPEATED statement (MIXED), 3958
RCI option
PROC SEQTEST statement, 6010
REPEATED statement (MIXED), 3958
RCONVERGE= option
FACTOR statement (CALIS), 878
FACTOR statement (TCALIS), 6754
PROC FACTOR statement, 1559
RCORR option
REPEATED statement (MIXED), 3958
RDF= option
PROC CALIS statement, 866
PROC CANCORR statement, 1041
PROC TCALIS statement, 6742
RDIF1 option
OUTPUT statement (FREQ), 1696
RDIF2 option
OUTPUT statement (FREQ), 1696
RDPREFIX= option
OUTPUT statement (TRANSREG), 7177
READ function
RESPONSE statement (CATMOD), 1132
RED option
PROC CANCORR statement, 1041
REDUCEOUT option
PROC LIFETEST statement, 3123
REDUNDANCY option
PROC CANCORR statement, 1041
REDUNDANCY= option
OUTPUT statement (TRANSREG), 7178
REF= option
CLASS statement (GENMOD), 1939
CLASS statement (GLMSELECT), 2704
CLASS statement (LOGISTIC), 3281
CLASS statement (PHREG), 4558
CLASS statement (SURVEYLOGISTIC), 6391
REFERENCE option
ROCCONTRAST statement (LOGISTIC), 3311
REFERENCE= option
MODEL statement, 3289
MODEL statement (TRANSREG), 7165
OUTPUT statement (TRANSREG), 7178
REFINE option
PROC ROBUSTREG statement, 5666
REFIT statement, REG procedure, 5500
REFLECT option
MODEL statement (TRANSREG), 7158
TRANSFORM statement, 5219
REFLINP= option
MODEL statement (GLIMMIX), 2175
REFMODEL statement
TCALIS procedure, 6811
REFMODEL statement, TCALIS procedure, 6811
REFPROPORTION= option
PAIREDFREQ statement (POWER), 4867
TWOSAMPLEFREQ statement (POWER), 4888
REFRESH= option
PROC PRINQUAL statement, 5209
REFSURVEXPHAZARD= option
TWOSAMPLESURVIVAL statement (POWER), 4907
REFSURVIVAL= option
TWOSAMPLESURVIVAL statement (POWER), 4907
REG option
MONOTONE statement (MI), 3766
REG procedure
syntax, 5451
REG procedure, ADD statement, 5467
REG procedure, BY statement, 5467
REG procedure, DELETE statement, 5468
REG procedure, FREQ statement, 5468
REG procedure, ID statement, 5468
REG procedure, MODEL statement, 5469
ACOV option, 5472
ACOVMETHOD= option, 5472
ADJRSQ option, 5472
AIC option, 5472
ALL option, 5472
ALPHA= option, 5472
B option, 5472
BEST= option, 5472
BIC option, 5473
CLB option, 5473
CLI option, 5473
CLM option, 5473
COLLIN option, 5473
COLLINOINT option, 5473
CORRB option, 5473
COVB option, 5473
CP option, 5474
DETAILS option, 5474
DW option, 5474
DWPROB option, 5474

EDF option, 5474
GMSEP option, 5474
GROUPNAMES= option, 5474
HCC option, 5475
HCCMETHOD= option, 5475
I option, 5475
INCLUDE= option, 5475
INFLUENCE option, 5475
JP option, 5475
LACKFIT option, 5475
MAXSTEP option, 5476
MSE option, 5476
NOINT option, 5476
NOPRINT option, 5476
OUTSEB option, 5476
OUTSTB option, 5476
OUTVIF option, 5476
P option, 5476
PARTIAL option, 5476
PARTIALDATA option, 5477
PARTIALR2 option, 5477
PC option, 5477
PCOMIT= option, 5477
PCORR1 option, 5477
PCORR2 option, 5477
PRESS option, 5477
R option, 5477
RIDGE= option, 5477
RMSE option, 5478
RSQUARE option, 5478
SBC option, 5478
SCORR1 option, 5478
SCORR2 option, 5478
SELECTION= option, 5435, 5478
SEQB option, 5479
SIGMA= option, 5479
SINGULAR= option, 5479
SLENTRY= option, 5479
SLSTAY= option, 5479
SP option, 5479
SPEC option, 5479
SS1 option, 5479
SS2 option, 5479
SSE option, 5479
START= option, 5480
STB option, 5480
STOP= option, 5480
TOL option, 5480
VIF option, 5480
WHITE option, 5480
XPX option, 5480
REG procedure, MTEST statement, 5480
CANPRINT option, 5481
DETAILS option, 5481
MSTAT= option, 5482
PRINT option, 5482
REG procedure, OUTPUT statement, 5482
keyword= option, 5483
OUT= option, 5482
REG procedure, PAINT statement, 5484
ALLOBS option, 5486
NOLIST option, 5486
RESET option, 5486
STATUS option, 5487
SYMBOL= option, 5486
UNDO option, 5487
REG procedure, PLOT statement, 5487
AIC option, 5493
ANNOTATE= option, 5493
BIC option, 5493
CAXIS= option, 5493
CFRAME= option, 5493
CHOCKING= option, 5493
CHREF= option, 5493
CLEAR option, 5497
CLINE= option, 5493
CMALLOWS= option, 5494
COLLECT option, 5497
CONF option, 5494
CP option, 5494
CTEXT= option, 5494
CVREF= option, 5494
DESCRIPTION= option, 5494
EDF option, 5494
GMSEP option, 5494
HAXIS= option, 5494
HPLOTS= option, 5498
HREF= option, 5494
IN option, 5495
JP option, 5495
LEGEND= option, 5495
LHREF= option, 5495
LLINE= option, 5495
LVREF= option, 5495
MODELFONT option, 5495
MODELHT option, 5495
MODELLAB option, 5495
MSE option, 5495
NAME= option, 5495
NOCOLLECT option, 5498
NOLINE option, 5495
NOMODEL option, 5496
NOSTAT option, 5496
NP option, 5496
OVERLAY option, 5496, 5498
PC option, 5496
PRED option, 5496
RIDGEPLOT option, 5496

SBC option, 5496
SP option, 5496
SSE option, 5496
STATFONT option, 5496
STATHT option, 5497
summary of options, 5490, 5491
SYMBOL= option, 5498
USEALL option, 5497
VAXIS= option, 5497
VPLOTS= option, 5499
VREF= option, 5497
REG procedure, PRINT statement, 5499
REG procedure, PROC REG statement, 5453
ALL option, 5454
ALPHA= option, 5454
ANNOTATE= option, 5454
CORR option, 5454
COVOUT option, 5454
DATA= option, 5455
EDF option, 5455
GOUT= option, 5455
LINEPRINTER option, 5455
NOPRINT option, 5455
OUTEST= option, 5455
OUTSEB option, 5455
OUTSSCP= option, 5455
OUTSTB option, 5456
OUTVIF option, 5456
PCOMIT= option, 5456
PLOT option, 5456
PLOTS option, 5456
PRESS option, 5466
RIDGE= option, 5466
RSQUARE option, 5466
SIMPLE option, 5466
SINGULAR= option, 5466
TABLEOUT option, 5467
USSCP option, 5467
REG procedure, REFIT statement, 5500
REG procedure, RESTRICT statement, 5500
REG procedure, REWEIGHT statement, 5502
ALLOBS option, 5504
NOLIST option, 5504
RESET option, 5504
STATUS option, 5505
UNDO option, 5505
WEIGHT= option, 5505
REG procedure, TEST statement, 5506
PRINT option, 5506
REG procedure, VAR statement, 5507
REG procedure, WEIGHT statement, 5507
REGPMM option
MONOTONE statement (MI), 3766
REGPREDMEANMATCH option
MONOTONE statement (MI), 3766
REGRESSION option
MONOTONE statement (MI), 3766
REGWQ option
MEANS statement (ANOVA), 713
MEANS statement (GLM), 2472
REITERATE option
MODEL statement (TRANSREG), 7165
PROC PRINQUAL statement, 5209
RELATIVERISK= option
PAIREDFREQ statement (POWER), 4868
TWOSAMPLEFREQ statement (POWER), 4888
RELRISK option
OUTPUT statement (FREQ), 1696
TABLES statement (FREQ), 1717, 1812
RENAMEPARM statement, TCALIS procedure, 6812
REORDER option
PROC FACTOR statement, 1560
REPEAT option
PLOT statement (BOXPLOT), 787
REPEATED statement
ANOVA procedure, 715
CATMOD procedure, 1129
GENMOD procedure, 1918, 1958
GLM procedure, 2481
MIXED procedure, 3954, 4019
REPLACE option
OUTPUT statement (TRANSREG), 7178
PROC DISTANCE statement, 1499
PROC PRINQUAL statement, 5209
PROC STDIZE statement, 6224
REPLACE= option
PROC FASTCLUS statement, 1639
REPLICATE statement
NLMIXED procedure, 4379
REPONLY option
PROC DISTANCE statement, 1499
PROC STDIZE statement, 6224
REPS= option
PROC SURVEYSELECT statement, 6634
VARMETHOD=BRR (PROC SURVEYFREQ statement), 6307
VARMETHOD=BRR (PROC SURVEYLOGISTIC statement), 6387
VARMETHOD=BRR (PROC SURVEYMEANS statement), 6482
VARMETHOD=BRR (PROC SURVEYREG statement), 6548
REPWEIGHTS statement
SURVEYFREQ procedure, 6310
SURVEYLOGISTIC procedure, 6406
SURVEYMEANS procedure, 6488

SURVEYREG procedure, 6558
RESAMPLE= option
ASSESS statement (PHREG), 4540
RESCHI= option
OUTPUT statement (LOGISTIC), 3308
RESDEV= option
OUTPUT statement (LOGISTIC), 3308
RESET option
ODS PATH statement, 588
PAINT statement (REG), 5486
REWEIGHT statement (REG), 5504
RESET= option
ODS GRAPHICS statement, 530
RESIDUAL keyword
OUTPUT statement (GLM), 2478
OUTPUT statement (GLMSELECT), 2715
OUTPUT statement (QUANTREG), 5382
OUTPUT statement (ROBUSTREG), 5670
OUTPUT statement (SURVEYREG), 6557
SCORE statement (GLMSELECT), 2718
RESIDUAL option
MIXED procedure, MODEL statement, 3988
MODEL statement (LOESS), 3210
MODEL statement (MIXED), 3941
MODEL statement (RSREG), 5733
PROC SCORE statement, 5766
RANDOM statement (GLIMMIX), 2192
RESIDUAL= option
OUTPUT statement (HPMIXED), 2824
OUTPUT statement (NLIN), 4287
PROC CALIS statement, 866
PROC TCALIS statement, 6742
RESIDUALS option
MODEL statement (GENMOD), 1952
OUTPUT statement (TRANSREG), 7178
PROC FACTOR statement, 1560
response functions (CATMOD), 1130, 1132, 1134, 1135, 1137, 1140, 1178, 1184, 1219
RESPONSE keyword
MODEL statement (CATMOD), 1114, 1117, 1118, 1120, 1129, 1142, 1148, 1150, 1158, 1160, 1165, 1174
RESPONSE statement
CATMOD procedure, 1130
RESPONSE= option
FACTORS statement (CATMOD), 1115
RESPONSE= option
REPEATED statement (CATMOD), 1130
RESPONSEPROB= option
LOGISTIC statement (POWER), 4831
REST option
NLOPTIONS statement (GLIMMIX), 401
NLOPTIONS statement (HPMIXED), 401
NLOPTIONS statement (TCALIS), 401
RESTART option
COVTEST statement (GLIMMIX), 2133
NLOPTIONS statement (GLIMMIX), 401
NLOPTIONS statement (HPMIXED), 401
NLOPTIONS statement (TCALIS), 401
PROC NLMIXED statement, 4369
RESTART= option
NLOPTIONS statement (CALIS), 898
RESTRICT statement
CATMOD procedure, 1137
REG procedure, 5500
RETAIN statement
NLIN procedure, 4292
REWEIGHT statement, REG procedure, 5502
RHO= option
PROC NLIN statement, 4282
RI option
REPEATED statement (MIXED), 3958
RIDGE statement
RSREG procedure, 5733
RIDGE= option
MODEL statement (REG), 5477
PROC CALIS statement, 866
PROC MDS statement, 3720
PROC MIXED statement, 3915
PROC REG statement, 5466
PROC TCALIS statement, 6743
RIDGEINIT= option
MODEL statement (PHREG), 4569
RIDGEPLOT option
PLOT statement (REG), 5496
RIDGING= option
MODEL statement (LOGISTIC), 3299
MODEL statement (PHREG), 4569
MODEL statement (SURVEYLOGISTIC), 6401
RISK option
TABLES statement (SURVEYFREQ), 6318
RISK1 option
TABLES statement (SURVEYFREQ), 6318
RISK2 option
TABLES statement (SURVEYFREQ), 6319
RISKDIFF option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1696
TABLES statement (FREQ), 1717
RISKDIFF1 option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1696
RISKDIFF2 option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1696

RISKDIFFC option
TABLES statement (FREQ), 1720
RISKLIMITS option
MODEL statement (LOGISTIC), 3299
RISKLIMITS= option
MODEL statement (PHREG), 4569
RITER= option
FACTOR statement (CALIS), 878
FACTOR statement (TCALIS), 6754
PROC FACTOR statement, 1560
RMSE option
MODEL statement (REG), 5478
RMSSTD option
PROC CLUSTER statement, 1246
ROBDIST keyword
OUTPUT statement (QUANTREG), 5382
ROBUST option
COMPUTE statement (VARIOGRAM), 7538
ROBUSTREG procedure
syntax, 5660
ROBUSTREG procedure, BY statement, 5667
ROBUSTREG procedure, CLASS statement, 5668
ROBUSTREG procedure, ID statement, 5668
ROBUSTREG procedure, MODEL statement, 5668
ALPHA= option, 5669
CORRB option, 5669
COVB option, 5669
CUTOFF option, 5669
DIAGNOSTICS option, 5669
ITPRINT option, 5669
LEVERAGE option, 5669
NOGOODFIT option, 5669
NOINT option, 5669
SINGULAR= option, 5669
ROBUSTREG procedure, OUTPUT statement, 5670
keyword= option, 5670
LEVEARAGE keyword, 5670
OUT= option, 5670
OUTLIER keyword, 5670
PREDICTED keyword, 5670
RESIDUAL keyword, 5670
SRESIDUAL keyword, 5670
STD_ERR keyword, 5670
XBETA keyword, 5670
ROBUSTREG procedure, PERFORMANCE statement, 5671
CPUCOUNT option, 5671
DETAILS option, 5671
NOTHREADS option, 5671
THREADS option, 5671
ROBUSTREG procedure, PROC ROBUSTREG statement, 5660
BIASTEST option, 5666
CHIF option, 5665, 5666
CONVERGENCE option, 5663, 5667
COVARIANCE option, 5663, 5665
COVOUT option, 5660
CSTEP option, 5664
DATA= option, 5660
EFF option, 5665, 5667
FWLS= option, 5661
H option, 5665
IADJUST option, 5664
INEST= option, 5661
INITEST option, 5667
INITH option, 5667
ITPRINT option, 5661
K0 option, 5666, 5667
MAXITER= option, 5663, 5666, 5667
NAMELEN= option, 5661
NBEST option, 5665
NREP option, 5665, 5666
ORDER= option, 5661
OUTEST= option, 5661
PLOT= option, 5661
REFINE option, 5666
SCALE option, 5664
SUBANALYSIS option, 5665
SUBGROUPSIZE option, 5665
SUBSETSIZE option, 5666
TOLERANCE option, 5666
WEIGHTFUNCTION option, 5664
ROBUSTREG procedure, TEST statement, 5671
ROBUSTREG procedure, WEIGHT statement, 5672
ROBUSTREG procedure,PROC ROBUSTREG statement
COVARIANCE option, 5666
SEED option, 5663
ROC statement
LOGISTIC procedure, 3310
ROCCONTRAST statement
LOGISTIC procedure, 3311
ROCEPS= option
MODEL statement (LOGISTIC), 3299
SCORE statement (LOGISTIC), 3313
ROCOPTIONS option
PROC LOGISTIC statement, 3274
ROLEVAR= option
GLMSELECT procedure, PARTITION statement, 2716
ROOT= option
PROC TREE statement, 7366
RORDER= option

PROC GENMOD statement, 1925
ROTATE= option
FACTOR statement (CALIS), 879
FACTOR statement (TCALIS), 6754
PROC FACTOR statement, 1560
ROUND option
PROC FACTOR statement, 1562
ROUND= option
PROC MI statement, 3753
ROW option
TABLES statement (SURVEYFREQ), 6319
ROW= option
PROC CORRESP statement, 1330
ROWID= option
BASELINE statement (PHREG), 4545
RP option
PROC CORRESP statement, 1330
RPREFIX= option
OUTPUT statement (TRANSREG), 7178
PROC CANCORR statement, 1041
PROC PRINCOMP statement, 5155
RRATIO= option
PROC QUANTREG statement, 5373
RRC1 option
OUTPUT statement (FREQ), 1696
RRC2 option
OUTPUT statement (FREQ), 1696
RSIDE option
RANDOM statement (GLIMMIX), 2192
RSK1 option
OUTPUT statement (FREQ), 1696
RSK11 option
OUTPUT statement (FREQ), 1696
RSK12 option
OUTPUT statement (FREQ), 1696
RSK2 option
OUTPUT statement (FREQ), 1696
RSK21 option
OUTPUT statement (FREQ), 1696
RSK22 option
OUTPUT statement (FREQ), 1697
RSQUARE
STATS= option (GLMSELECT), 2714
RSQUARE option
MODEL statement (LOGISTIC), 3300
MODEL statement (REG), 5478
MODEL statement (SURVEYLOGISTIC), 6401
MODEL statement (TRANSREG), 7166
PROC CLUSTER statement, 1246
PROC REG statement, 5466
RSQUAREDIFF= option
MULTREG statement (POWER), 4838
RSQUAREFULL= option
MULTREG statement (POWER), 4839
RSQUAREREDUCED= option
MULTREG statement (POWER), 4839
RSREG procedure
syntax, 5726
RSREG procedure, BY statement, 5730
RSREG procedure, ID statement, 5730
RSREG procedure, MODEL statement, 5731
ACTUAL option, 5731
BYOUT option, 5731
COVAR= option, 5732
D option, 5732
L95 option, 5732
L95M option, 5732
LACKFIT option, 5732
NOANOVA option, 5732
NOCODE option, 5732
NOOPTIMAL option, 5732
NOPRINT option, 5732
PREDICT option, 5733
PRESS option, 5733
RESIDUAL option, 5733
U95 option, 5733
U95M option, 5733
RSREG procedure, PROC RSREG statement, 5726
DATA= option, 5726
NOPRINT option, 5726
OUT= option, 5727
PLOTS= option, 5727
RSREG procedure, RIDGE statement, 5733
CENTER= option, 5734
MAXIMUM option, 5734
MINIMUM option, 5734
NOPRINT option, 5734
OUTR= option, 5734
RADIUS= option, 5734
RSREG procedure, WEIGHT statement, 5735
RSTUDENT keyword
OUTPUT statement (GLM), 2478
RUPDATE= option
REPEATED statement (GENMOD), 1960

S option
PROC CANCORR statement, 1041
SALPHA= option
NLOPTIONS statement (CALIS), 889
PROC CALIS statement, 866
PROC TCALIS statement, 6743
SAMPLESIZE statement
SEQDESIGN procedure, 5809
SAMPRATE= option
PROC SURVEYSELECT statement, 6634
SAMPSIZE= option

PROC SURVEYSELECT statement, 6636
SAVAGE option
EXACT statement (NPAR1WAY), 4463
OUTPUT statement (NPAR1WAY), 4465
PROC NPAR1WAY statement, 4461
SAVE option
PROC NLIN statement, 4282
SBC
STATS= option (GLMSELECT), 2714
SBC option
MODEL statement (REG), 5478
MODEL statement (TRANSREG), 7153
PLOT statement (REG), 5496
SCALE option
PROC ROBUSTREG statement, 5664
MODEL statement (QUANTREG), 5380
PROC MCMC statement, 3506
SCALE= option
MODEL statement (GENMOD), 1953
MODEL statement (KRIGE2D), 2947
MODEL statement (LIFEREG), 3025
MODEL statement (LOESS), 3210
MODEL statement (LOGISTIC), 3300
MODEL statement (PROBIT), 5298
ODS GRAPHICS statement, 531
SIMULATE statement (SIM2D), 6174
SCALEDINDEP option
MODEL statement (LOESS), 3211
Score statement (LOESS), 3214
SCHEFFE option
MEANS statement (ANOVA), 714
MEANS statement (GLM), 2472
SCORE option
PROC FACTOR statement, 1562
SCORE procedure
syntax, 5766
SCORE procedure, BY statement, 5767
SCORE procedure, ID statement, 5768
SCORE procedure, PROC SCORE statement, 5766
DATA= option, 5766
NOSTD option, 5766
OUT= option, 5766
PREDICT option, 5766
RESIDUAL option, 5766
SCORE= option, 5767
TYPE= option, 5767
SCORE procedure, VAR statement, 5768
SCORE statement
GLMSELECT procedure, 2717
LOESS procedure, 3213
LOGISTIC procedure, 3312
SCORE statement, GAM procedure, 1847
SCORE statement, TPSPLINE procedure, 7071
SCORE= option
PROC SCORE statement, 5767
SCOREMOD option
PROC GLIMMIX statement, 2119
SCORES option
PROC PRINQUAL statement, 5209
SCORES= option
TABLES statement (FREQ), 1720, 1822
SCORES=DATA option
EXACT statement (NPAR1WAY), 4463
OUTPUT statement (NPAR1WAY), 4465
PROC NPAR1WAY statement, 4461
SCORING= option
MODEL statement (GENMOD), 1953
PROC GLIMMIX statement, 2119
PROC MIXED statement, 3915
SCOROUT option
TABLES statement (FREQ), 1720
SCORR option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1697
TEST statement (FREQ), 1723
SCORR1 option
MODEL statement (REG), 5478
SCORR2 option
MODEL statement (REG), 5478
SCREE option
PROC FACTOR statement, 1562
SCWGT statement
GENMOD procedure, 1962
SE option
PROC FACTOR statement, 1562
SEB option
PROC CANCORR statement, 1041
SEED option
MODEL statement (QUANTREG), 5380
PROC MCMC statement, 3506
PROC MI statement, 3753
PROC NLMIXED statement, 4370
PROC PLAN statement, 4736
PROC ROBUSTREG statement (ROBUSTREG), 5663
SEED statement
VARCLUS procedure, 7475
SEED= option
ASSESS statement (PHREG), 4540
BAYES statement(PHREG), 4555
EXACT statement (FREQ), 1694
EXACT statement (NPAR1WAY), 4464
PRIOR statement (MIXED), 3948
PROC FASTCLUS statement, 1639
PROC GLMSELECT statement, 2701
PROC MULTTEST statement, 4196
PROC PLS statement, 4775, 4776

PROC SIMNORMAL statement, 6203
PROC SURVEYSELECT statement, 6637
SIMULATE statement (SIM2D), 6174
SEEDBY option
PROC SIMNORMAL statement, 6203
SELECT= modifier
INFLUENCE option, MODEL statement (MIXED), 3936
SELECT= option
MODEL statement (GLMSELECT), 2712
MODEL statement (LOESS), 3211
SELECTALL option
PROC SURVEYSELECT statement, 6638
SELECTION= option
MODEL statement (GLMSELECT), 2709
MODEL statement (LOGISTIC), 3301
MODEL statement (PHREG), 4570
MODEL statement (REG), 5478
REG procedure, MODEL statement, 5435
SEPARATE option
EFFECT statement, spline (GLIMMIX), 386
EFFECT statement, spline (GLMSELECT), 386
SEPARATORS= option
MODEL statement (TRANSREG), 7154, 7166
SEQB option
MODEL statement (REG), 5479
SEQDESIGN procedure, DESIGN statement, 5803
ALPHA= option, 5804
ALT= option, 5804
BETA= option, 5804
BETAOVERLAP= option, 5804
BOUNDARYKEY= option, 5805
INFO= option, 5805
METHOD= option, 5805
NSTAGES= option, 5808
OVERLAP= option, 5804
STOP= option, 5808
SEQDESIGN procedure, PROC SEQDESIGN statement, 5799
ALTREF= option, 5800
BOUNDARYSCALE= option, 5800
BSCALE= option, 5800
ERRSPEND option, 5801
MAXINFO= option, 5801
PLOTS option, 5802
PSS option, 5801
STOPPROB option, 5801
SEQDESIGN procedure, SAMPLESIZE statement, 5809
MODEL= option, 5809
SEQTEST procedure, PROC SEQTEST statement, 6003
BETAOVERLAP= option, 6004
BOUNDARY= option, 6005
BOUNDARYADJ= option, 6005
BOUNDARYKEY= option, 6006
BOUNDARYSCALE= option, 6006
CIALPHA= option, 6007
CITYPE= option, 6007
CONDPOWER option, 6009
DATA= option, 6007
ERRSPEND option, 6009
ERRSPENDADJ= option, 6005
ERRSPENDMIN= option, 6007
NSTAGES= option, 6008
ORDER= option, 6008
OVERLAP= option, 6004
PARMS= option, 6009
PLOTS option, 6011
PREDPOWER option, 6009
PSS option, 6009
RCI option, 6010
STOPPROB option, 6010
SEQUENTIAL option
MODEL statement (LOGISTIC), 3301
MODEL statement (PHREG), 4570
SFACTOR= option
PRIOR statement (MIXED), 3948
SHAPE1= option
MODEL statement (LIFEREG), 3025
SHAPE= option
PROC DISTANCE statement, 1499
PROC MDS statement, 3720
SHORT option
MODEL statement (TRANSREG), 7166
PROC ACECLUS statement, 677
PROC CALIS statement, 865
PROC CANCORR statement, 1041
PROC CANDISC statement, 1071
PROC CORRESP statement, 1330
PROC DISCRIM statement, 1395
PROC FASTCLUS statement, 1639
PROC MODECLUS statement, 4107
PROC STEPDISC statement, 6277
PROC TCALIS statement, 6742
PROC VARCLUS statement, 7473
SHOWCODING option
CLASS statement (GLMSELECT), 2702
SHOWPVALUES option
MODEL statement (GLMSELECT), 2714
SIDAK option
MEANS statement (ANOVA), 714
MEANS statement (GLM), 2473

PROC MULTTEST statement, 4197, 4213, 4233
SIDES= option
ONECORR statement (POWER), 4843
ONESAMPLEFREQ statement (POWER), 4848
ONESAMPLEMEANS statement (POWER), 4856
ONEWAYANOVA statement (POWER), 4862
PAIREDFREQ statement (POWER), 4868
PAIREDMEANS statement (POWER), 4876
PROC TTEST statement, 7407
TWOSAMPLEFREQ statement (POWER), 4888
TWOSAMPLEMEANS statement (POWER), 4896
TWOSAMPLESURVIVAL statement (POWER), 4907
TWOSAMPLEWILCOXON statement (POWER), 4913
SIGITER option
PROC MIXED statement, 3915
SIGMA= option
MODEL statement (REG), 5479
SIGSQ= option
PROC NLIN statement, 4282
SIM2D procedure, 6155
syntax, 6164
SIM2D procedure, BY statement, 6169
SIM2D procedure, COORDINATES statement, 6169
XCOORD= option, 6170
YCOORD= option, 6170
SIM2D procedure, GRID statement, 6170
GRIDDATA= option, 6171
X= option, 6170
XCCORD= option, 6171
Y= option, 6170
YCOORD= option, 6171
SIM2D procedure, MEAN statement, 6175
SIM2D procedure, PROC SIM2D statement, 6165
DATA= option, 6165
NARROW option, 6166
NOPRINT option, 6166
OUTSIM= option, 6166
PLOTS option, 6166
PLOTS(ONLY) option, 6166
PLOTS=ALL option, 6167
PLOTS=EQUATE option, 6167
PLOTS=MEANS option, 6168
PLOTS=NONE option, 6167
PLOTS=OBSERVATIONS option, 6167
SIM2D procedure, SIMULATE statement, 6171
ANGLE= option, 6172
FORM= option, 6172
MDATA= option, 6172
NUGGET= option, 6174
NUMREAL= option, 6171
RANGE= option, 6174
RATIO= option, 6174
SCALE= option, 6174
SEED= option, 6174
SINGULAR= option, 6174
VAR= option, 6171
SIMILAR option
PROC TREE statement, 7366
SIMILAR= option
PROC MDS statement, 3720
SIMNORMAL procedure
syntax, 6202
SIMNORMAL procedure, BY statement, 6204
SIMPLE option
PROC CALIS statement, 866
PROC CANCORR statement, 1041
PROC CANDISC statement, 1071
PROC CLUSTER statement, 1246
PROC DISCRIM statement, 1395
PROC FACTOR statement, 1563
PROC LOGISTIC statement, 3275
PROC MI statement, 3754
PROC MODECLUS statement, 4108
PROC PHREG statement, 4539
PROC REG statement, 5466
PROC STEPDISC statement, 6277
PROC TCALIS statement, 6743
PROC VARCLUS statement, 7473
SIMPLEDIFFTYPE option
LSMEANS statement (GLIMMIX), 2153
SIMPLEEDIFF= option
LSMEANS statement (GLIMMIX), 2152
SIMREPORT= option
PROC MCMC statement, 3506
SIMTEST statement, TCALIS procedure, 6814
SIMULATE statement (SIM2D), 6171
SING= option
PROC CANCORR statement, 1041
PROC PRINCOMP statement, 5156
SINGCHOL= option
MODEL statement (MIXED), 3941
PROC GLIMMIX statement, 2120
PROC HPMIXED statement, 2814
PROC NLMIXED statement, 4370
SINGDEN= option
PROC MCMC statement, 3506
SINGHESS= option

PROC NLMIXED statement, 4370
SINGRES= option
MODEL statement (MIXED), 3941
PROC GLIMMIX statement (GLIMMIX), 2120
PROC HPMIXED statement, 2814
SINGSWEEP= option
PROC NLMIXED statement, 4370
SINGULAR option
CONTRAST statement (GLM), 2455
LSMEANS statement (GLM), 2463
PROC MI statement, 3754
SINGULAR1= option
PROC SIMNORMAL statement, 6204
SINGULAR2= option
PROC SIMNORMAL statement, 6204
SINGULAR= option
CONTRAST statement (GENMOD), 1942
CONTRAST statement (GLIMMIX), 2126
CONTRAST statement (GLMPOWER), 2653
CONTRAST statement (HPMIXED), 2818
CONTRAST statement (LOGISTIC), 3283
CONTRAST statement (MIXED), 3920
CONTRAST statement (PHREG), 4561
CONTRAST statement (SURVEYLOGISTIC), 6394
CONTRAST statement (SURVEYREG), 6552
ESTIMATE statement (GENMOD), 1944
ESTIMATE statement (GLIMMIX), 2138
ESTIMATE statement (GLM), 2457
ESTIMATE statement (HPMIXED), 2820
ESTIMATE statement (MIXED), 3921
ESTIMATE statement (SURVEYREG), 6554
LSMEANS statement (GENMOD), 1946
LSMEANS statement (GLIMMIX), 2152
LSMEANS statement (HPMIXED), 2822
LSMEANS statement (MIXED), 3928
LSMESTIMATE statement (GLIMMIX), 2159
MODEL statement (GENMOD), 1953
MODEL statement (GLM), 2476
MODEL statement (KRIGE2D), 2947
MODEL statement (LIFEREG), 3025
MODEL statement (LOGISTIC), 3301
MODEL statement (MIXED), 3942
MODEL statement (PHREG), 4571
MODEL statement (QUANTREG), 5381
MODEL statement (REG), 5479
MODEL statement (ROBUSTREG), 5669
MODEL statement (SURVEYLOGISTIC), 6401
MODEL statement (SURVEYREG), 6556
MODEL statement (TRANSREG), 7166
NLOPTIONS statement (CALIS), 890
NLOPTIONS statement (GLIMMIX), 401
NLOPTIONS statement (HPMIXED), 401
NLOPTIONS statement (TCALIS), 401
PROC ACECLUS statement, 677
PROC CALIS statement, 867
PROC CANCORR statement, 1041
PROC CANDISC statement, 1071
PROC CORRESP statement, 1330
PROC DISCRIM statement, 1395
PROC FACTOR statement, 1563
PROC GLIMMIX statement, 2120
PROC LIFETEST statement, 3123
PROC MDS statement, 3720
PROC NLIN statement, 4282
PROC ORTHOREG statement, 4510
PROC PRINCOMP statement, 5156
PROC PRINQUAL statement, 5209
PROC REG statement, 5466
PROC SINGCHOL statement, 2814
PROC STEPDISC statement, 6277
PROC TCALIS statement, 6743
SIMULATE statement (SIM2D), 6174
SINGULARMSG= option
PROC KRIGE2D statement, 2940
SINGVAR option
PROC NLMIXED statement, 4370
SIZE statement
SURVEYSELECT procedure, 6639
SIZE= modifier
INFLUENCE option, MODEL statement (MIXED), 3937
SKIPHLABELS= option
PLOT statement (BOXPLOT), 788
SL
STATS= option (GLMSELECT), 2714
SLENTRY= option
MODEL statement (GLMSELECT), 2712
MODEL statement (LOGISTIC), 3301
MODEL statement (PHREG), 4571
MODEL statement (REG), 5479
PROC STEPDISC statement, 6277
SLICE= option
LSMEANS statement (GLIMMIX), 2152
LSMEANS statement (GLM), 2463
LSMEANS statement (MIXED), 3928
SLICEDIFF= option
LSMEANS statement (GLIMMIX), 2152
SLICEDIFFTYPE option
LSMEANS statement (GLIMMIX), 2153
SLMW= option
PROC CALIS statement, 867

PROC TCALIS statement, 6743
SLPOOL= option
PROC DISCRIM statement, 1396
SLSTAY= option
MODEL statement (GLMSELECT), 2712
MODEL statement (LOGISTIC), 3301
MODEL statement (PHREG), 4571
MODEL statement (REG), 5479
PROC STEPDISC statement, 6277
SM= option
MODEL statement (TRANSREG), 7150
TRANSFORM statement, 5218
SMC option
PROC CANCORR statement, 1041
SMDCR option
OUTPUT statement (FREQ), 1697
TEST statement (FREQ), 1723, 1819
SMDRC option
OUTPUT statement (FREQ), 1697
TEST statement (FREQ), 1723
SMETHOD= option
PROC CALIS statement, 857
PROC NLIN statement, 4283
PROC TCALIS statement, 6731
SMM option
MEANS statement (ANOVA), 714
MEANS statement (GLM), 2473
SMOOTH transformation
MODEL statement (TRANSREG), 7145
SMOOTH= option
MODEL statement (LOESS), 3213
SNK option
MEANS statement (ANOVA), 714
MEANS statement (GLM), 2473
SNORM option
PROC DISTANCE statement, 1499
PROC STDIZE statement, 6225
SOCKET option
NLOPTIONS statement (GLIMMIX), 401
NLOPTIONS statement (HPMIXED), 401
NLOPTIONS statement (TCALIS), 401
SOLUTION option
MODEL statement (GLIMMIX), 2175, 2262
MODEL statement (GLM), 2477
MODEL statement (HPMIXED), 2823
MODEL statement (MIXED), 3942, 3986
MODEL statement (PLS), 4783
MODEL statement (SURVEYREG), 6556
RANDOM statement (GLIMMIX), 2192
RANDOM statement (HPMIXED), 2830
RANDOM statement (MIXED), 3952
SOLVE option
MODEL statement (TRANSREG), 7161
SORT option
PROC TREE statement, 7366
SORT= option
PROC SURVEYSELECT statement, 6638
SORTED option
REPEATED statement (GENMOD), 1961
SOURCE option
PROC CORRESP statement, 1330
SOURCE statement
TEMPLATE procedure, 557
SP option
MODEL statement (REG), 5479
PLOT statement (REG), 5496
SPACES= option
PROC TREE statement, 7366
SPARSE option
TABLES statement (FREQ), 1721, 1800
SPCORR option
PROC CANCORR statement, 1041
SPEC option
MODEL statement (REG), 5479
SPECLIMITS= option
PROC VARCOMP statement, 7498
SPLINE keyword
OUTPUT statement (QUANTREG), 5382
SPLINE transformation
MODEL statement (TRANSREG), 7146
TRANSFORM statement (PRINQUAL), 5215
SPLIT option
CLASS statement (GLMSELECT), 2705
EFFECT statement, spline (GLMSELECT), 387
SPRECISION= option
PROC CALIS statement, 857, 867
PROC TCALIS statement, 6732, 6743
SQPCORR option
PROC CANCORR statement, 1041
SQSPCORR option
PROC CANCORR statement, 1041
SRESIDUAL keyword
OUTPUT statement (QUANTREG), 5382
OUTPUT statement (ROBUSTREG), 5670
SS1 option
MODEL statement (GLM), 2477
MODEL statement (REG), 5479
SS2 option
MODEL statement (GLM), 2477
MODEL statement (REG), 5479
MODEL statement (TRANSREG), 7166
SS3 option
MODEL statement (GLM), 2477
SS4 option
MODEL statement (GLM), 2477

SSCP option
REPEATED statement (MIXED), 3958
SSE option
MODEL statement (REG), 5479
PLOT statement (REG), 5496
SSE= option
OUTPUT statement (NLIN), 4288
SSPLINE transformation
MODEL statement (TRANSREG), 7147
TRANSFORM statement (PRINQUAL), 5216
ST option
EXACT statement (NPAR1WAY), 4463
OUTPUT statement (NPAR1WAY), 4465
PROC NPAR1WAY statement, 4461
STACKING option
PROC SURVEYMEANS statement, 6476
STANDARD option
PROC CLUSTER statement, 1246
PROC MODECLUS statement, 4108
PROC PRINCOMP statement, 5156
PROC PRINQUAL statement, 5209
STANDARDIZE option
EFFECT statement, polynomial (GLIMMIX), 382
EFFECT statement, polynomial (GLMSELECT), 382
STANDORTH option
MODEL statement (TRANSREG), 7154
START option
PROC NLMIXED statement, 4370
START= option
MCMC statement (MI), 3762
MODEL statement (LOGISTIC), 3301
MODEL statement (PHREG), 4571
MODEL statement (REG), 5480
PROC CALIS statement, 867
PROC STEPDISC statement, 6277
PROC TCALIS statement, 6743
STARTGLM option
PROC GLIMMIX statement, 2120
STAT= option
PROC PLS statement, 4776
STATFONT option
PLOT statement (REG), 5496
STATHT option
PLOT statement (REG), 5497
STATISTICS= option
BAYES statement(GENMOD), 1936
BAYES statement(PHREG), 3015, 4555
PROC MCMC statement, 3507
STATS option
MODEL statement (GLMSELECT), 2713
PROC SURVEYSELECT statement, 6638
STATS= option
PROC MCMC statement, 3507
STATUS option
PAINT statement (REG), 5487
REWEIGHT statement (REG), 5505
STB option
MODEL statement (GLMSELECT), 2714
MODEL statement (LOGISTIC), 3302
MODEL statement (REG), 5480
MODEL statement (SURVEYLOGISTIC), 6401
PROC CANCORR statement, 1042
STD keyword
OUTPUT statement (SURVEYREG), 6557
STD option
MODEL statement (LOESS), 3213
PROC PRINCOMP statement, 5156
STD statement, CALIS procedure, 906
STD statement, TCALIS procedure, 6815
STD_ERR keyword
OUTPUT statement (LIFEREG), 3027
OUTPUT statement (QUANTREG), 5382
OUTPUT statement (ROBUSTREG), 5670
STDCOEF option
MODEL statement (GLIMMIX), 2175
STDDEV= option
ONESAMPLEMEANS statement (POWER), 4857
ONEWAYANOVA statement (POWER), 4862
PAIREDMEANS statement (POWER), 4876
POWER statement (GLMPOWER), 2659
TWOSAMPLEMEANS statement (POWER), 4897
STDERR option
LSMEANS statement (GLM), 2463
PROC CALIS statement, 867
PROC LIFETEST statement, 3123
PROC TCALIS statement, 6744
STDERR statement
MIANALYZE procedure, 3849
STDERR= option
OUTPUT statement (HPMIXED), 2824
STDI keyword
OUTPUT statement (GLM), 2479
STDI= option
OUTPUT statement (NLIN), 4288
STDIZE procedure
syntax, 6220
STDIZE procedure, BY statement, 6225
STDIZE procedure, FREQ statement, 6226
NOTRUNCATE option, 6227
STDIZE procedure, LOCATION statement, 6227

STDIZE procedure, PROC STDIZE statement, 6221
ADD= option, 6222
DATA= option, 6222
FUZZ= option, 6222
INITIAL= option, 6222
KEEPLEN, 6223
METHOD= option, 6223
MISSING= option, 6223
MULT= option, 6223
NMARKERS= option, 6223
NOMISS option, 6223
NORM option, 6223
OUT= option, 6223
OUTSTAT= option, 6224
PCTLDEF= option, 6224
PCTLMTD option, 6224
PCTLPTS option, 6224
PSTAT option, 6224
REPLACE option, 6224
REPONLY option, 6224
SNORM option, 6225
UNSTD option, 6225
VARDEF option, 6225
STDIZE procedure, SCALE statement, 6227
STDIZE procedure, VAR statement, 6227
STDIZE procedure, WGT statement, 6227
STDMEAN option
PROC CANDISC statement, 1071
PROC DISCRIM statement, 1396
PROC STEPDISC statement, 6277
STDONLY option
PROC DISTANCE statement, 1499
STDP keyword
OUTPUT statement (GLM), 2479
OUTPUT statement (SURVEYREG), 6557
STDP= option
OUTPUT statement (NLIN), 4288
STDR keyword
OUTPUT statement (GLM), 2479
STDR= option
OUTPUT statement (NLIN), 4288
STDXBETA= option
OUTPUT statement (LOGISTIC), 3308
OUTPUT statement (SURVEYLOGISTIC), 6404
STEP= option
PLOT statement (GLMPOWER), 2655
PLOT statement (POWER), 4881
STEPBON option
PROC MULTTEST statement, 4197
STEPBOOT option
PROC MULTTEST statement, 4197
STEPDISC procedure
syntax, 6273
STEPDISC procedure, BY statement, 6278
STEPDISC procedure, CLASS statement, 6278
STEPDISC procedure, FREQ statement, 6279
STEPDISC procedure, PROC STEPDISC statement, 6274
ALL option, 6275
BCORR option, 6275
BCOV option, 6275
BSSCP option, 6275
DATA= option, 6275
INCLUDE= option, 6275
MAXMACRO= option, 6275
MAXSTEP= option, 6276
METHOD= option, 6276
PCORR option, 6276
PCOV option, 6276
PR2ENTRY= option, 6276
PR2STAY= option, 6276
PSSCP option, 6276
SHORT option, 6277
SIMPLE option, 6277
SINGULAR= option, 6277
SLENTRY= option, 6277
SLSTAY= option, 6277
START= option, 6277
STDMEAN option, 6277
STOP= option, 6277
TCORR option, 6277
TCOV option, 6277
TSSCP option, 6277
WCORR option, 6278
WCOV option, 6278
WSSCP option, 6278
STEPDISC procedure, VAR statement, 6279
STEPDISC procedure, WEIGHT statement, 6279
STEPDOWN option
ESTIMATE statement (GLIMMIX), 2138
LSMEANS statement (GLIMMIX), 2154
LSMESTIMATE statement (GLIMMIX), 2159
STEPPERM option
PROC MULTTEST statement, 4197
STEPS option
SCORE statement (LOESS), 3214
STEPSID option
PROC MULTTEST statement, 4197, 4233
STOP= option
DESIGN statement (SEQDESIGN), 5808
MODEL statement (GLMSELECT), 2712
MODEL statement (LOGISTIC), 3302
MODEL statement (PHREG), 4571
MODEL statement (REG), 5480
PROC STEPDISC statement, 6277

STOPPROB option
PROC SEQDESIGN statement, 5801
PROC SEQTEST statement, 6010
STOPRES option
MODEL statement (LOGISTIC), 3302
MODEL statement (PHREG), 4571
STRATA statement
LIFETEST procedure, 3125
LOGISTIC procedure, 3314
MULTTEST procedure, 4200
PHREG procedure, 4577
SURVEYFREQ procedure, 6312
SURVEYLOGISTIC procedure, 6407
SURVEYMEANS procedure, 6489
SURVEYREG procedure, 6559
SURVEYSELECT procedure, 6640
STRICT= option
PROC FASTCLUS statement, 1639
STRUCTEQ statement, CALIS procedure, 907
STRUCTEQ statement, TCALIS procedure, 6817
STUDENT keyword
OUTPUT statement (GLM), 2479
STUDENT= option
OUTPUT statement (HPMIXED), 2824
OUTPUT statement (NLIN), 4288
STUTC option
OUTPUT statement (FREQ), 1697
TEST statement (FREQ), 1723
STYLE= option
ODS destination statement, 532
ODS LATEX statement, 604
SUBANALYSIS option
PROC ROBUSTREG statement, 5665
SUBCLUSTER= option
REPEATED statement (GENMOD), 1961
SUBGRADIENT option
PROC GLIMMIX statement, 2120
PROC NLMIXED statement, 4370
SUBGROUP statement
SURVEYLOGISTIC procedure, 6394
SURVEYMEANS procedure, 6485
SURVEYREG procedure, 6552
SUBGROUPSIZE option
PROC ROBUSTREG statement, 5665
SUBJECT option
CONTRAST statement (GLIMMIX), 2126
CONTRAST statement (MIXED), 3920
ESTIMATE statement (GLIMMIX), 2139
ESTIMATE statement (MIXED), 3921
SUBJECT= option
CONTRAST statement (HPMIXED), 2818
ESTIMATE statement (HPMIXED), 2820
RANDOM statement (GLIMMIX), 2193
RANDOM statement (HPMIXED), 2831
RANDOM statement (MIXED), 3918, 3952
REPEATED statement (GENMOD), 1958
REPEATED statement (MIXED), 3958
SUBSETSIZE option
PROC ROBUSTREG statement, 5666
subsidiary group specification statements, TCALIS procedure, 6716
subsidiary model specification statements, TCALIS procedure, 6718
SUM option
PROC MODECLUS statement, 4108
SUMMARY option
MANOVA statement (ANOVA), 708
MANOVA statement (GLM), 2466
PROC CALIS statement, 865
PROC FASTCLUS statement, 1639
PROC TCALIS statement, 6742
PROC VARCLUS statement, 7473
REPEATED statement (ANOVA), 719
REPEATED statement (GLM), 2485
SUPPLEMENTARY statement
CORRESP procedure, 1331
SURVEYFREQ procedure
syntax, 6301
SURVEYFREQ procedure, BY statement, 6309
SURVEYFREQ procedure, CLUSTER statement, 6310
SURVEYFREQ procedure, PROC SURVEYFREQ statement, 6302
DATA= option, 6302
DFADJ option (VARMETHOD=BRR), 6305
DFADJ option (VARMETHOD=JK), 6308
FAY= option (VARMETHOD=BRR), 6306
HADAMARD= option (VARMETHOD=BRR), 6306
MISSING option, 6302
NOMCAR option, 6302
NOSUMMARY option, 6303
ORDER= option, 6303
OUTJKCOEFS= option (VARMETHOD=JK), 6308
OUTWEIGHTS= option (VARMETHOD=BRR), 6307
OUTWEIGHTS= option (VARMETHOD=JK), 6308
PAGE option, 6303
PRINTH option (VARMETHOD=BRR), 6307
RATE= option, 6303
REPS= option (VARMETHOD=BRR), 6307
TOTAL= option, 6304
VARHEADER= option, 6304

VARMETHOD= option, 6304
SURVEYFREQ procedure, REPWEIGHTS statement, 6310
DF= option, 6311
JKCOEFS= option, 6311
SURVEYFREQ procedure, STRATA statement, 6312
LIST option, 6312
SURVEYFREQ procedure, TABLES statement, 6312
ALPHA= option, 6315
CHISQ option, 6315
CHISQ1 option, 6315
CL option, 6316
CLWT option, 6316
COL option, 6316
CV option, 6316
CVWT option, 6316
DEFF option, 6316
DF= option, 6316
EXPECTED option, 6317
LRCHISQ option, 6317
LRCHISQ1 option, 6317
NOCELLPERCENT option, 6317
NOFREQ option, 6317
NOPERCENT option, 6317
NOPRINT option, 6318
NOSPARSE option, 6318
NOSTD option, 6318
NOTOTAL option, 6318
NOWT option, 6318
OR option, 6318
RISK option, 6318
RISK1 option, 6318
RISK2 option, 6319
ROW option, 6319
TESTP= option, 6319
VAR option, 6319
VARWT option, 6319
WCHISQ option, 6320
WLLCHISQ option, 6320
WTFREQ option, 6320
SURVEYFREQ procedure, WEIGHT statement, 6320
SURVEYLOGISTIC procedure, 6382
syntax, 6382
SURVEYLOGISTIC procedure, BY statement, 6388
SURVEYLOGISTIC procedure, CLASS statement, 6389
CPREFIX= option, 6389
DESCENDING option, 6389
LPREFIX= option, 6389
ORDER= option, 6389
PARAM= option, 6390, 6412
REF= option, 6391
SURVEYLOGISTIC procedure, CLUSTER statement, 6391
SURVEYLOGISTIC procedure, CONTRAST statement, 6391
ALPHA= option, 6393
E option, 6393
ESTIMATE= option, 6394
SINGULAR= option, 6394
SURVEYLOGISTIC procedure, DOMAIN statement, 6394
SURVEYLOGISTIC procedure, FREQ statement, 6395
SURVEYLOGISTIC procedure, MODEL statement, 6395
ABSFCONV option, 6398
ADJBOUND= option, 6402
ALPHA= option, 6398
CLODDS option, 6398
CLPARM option, 6399
CORRB option, 6399
COVB option, 6399
DEFFBOUND= option, 6402
DESCENDING option, 6396
EXPEST option, 6399
FCONV= option, 6399
GCONV= option, 6399
ITPRINT option, 6399
LINK= option, 6399
MAXITER= option, 6400
NOCHECK option, 6400
NODESIGNPRINT= option, 6400
NODUMMYPRINT= option, 6400
NOINT option, 6400
OFFSET= option, 6400
ORDER= option, 6396
PARMLABEL option, 6400
RIDGING= option, 6401
RSQUARE option, 6401
SINGULAR= option, 6401
STB option, 6401
TECHNIQUE= option, 6401
VADJUST= option, 6401
XCONV= option, 6402
SURVEYLOGISTIC procedure, OUTPUT statement, 6402
ALPHA= option, 6404
LOWER= option, 6403
OUT= option, 6403
PREDICTED= option, 6403
PREDPROBS= option, 6403
STDXBETA = option, 6404
UPPER= option, 6404

XBETA= option, 6404
SURVEYLOGISTIC procedure, PROC SURVEYLOGISTIC statement, 6383
ALPHA= option, 6383
DATA= option, 6383
FAY= option (VARMETHOD=BRR), 6386
H= option (VARMETHOD=BRR), 6386
HADAMARD= option (VARMETHOD=BRR), 6386
INEST= option, 6383
MISSING option, 6383
N= option, 6384
NAMELEN= option, 6383
NOMCAR option, 6383
NOSORT option, 6384
OUTJKCOEFS= option (VARMETHOD=JK), 6388
OUTWEIGHTS= option (VARMETHOD=BRR), 6387
OUTWEIGHTS= option (VARMETHOD=JK), 6388
PRINTH option (VARMETHOD=BRR), 6387
R= option, 6384
RATE= option, 6384
REPS= option (VARMETHOD=BRR), 6387
TOTAL= option, 6384
VARMETHOD= option, 6385
SURVEYLOGISTIC procedure, REPWEIGHTS statement, 6406
DF= option, 6406
JKCOEFS= option, 6406
SURVEYLOGISTIC procedure, STRATA statement, 6407
LIST option, 6407
SURVEYLOGISTIC procedure, TEST statement, 6408
PRINT option, 6408
SURVEYLOGISTIC procedure, UNITS statement, 6409
DEFAULT= option, 6409
SURVEYLOGISTIC procedure, WEIGHT statement, 6409
SURVEYMEANS procedure
syntax, 6473
SURVEYMEANS procedure, BY statement, 6483
SURVEYMEANS procedure, CLASS statement, 6484
SURVEYMEANS procedure, CLUSTER statement, 6485
SURVEYMEANS procedure, DOMAIN statement, 6485
DFADJ option, 6486
SURVEYMEANS procedure, PROC SURVEYMEANS statement, 6474
ALPHA= option, 6474
DATA= option, 6474
DFADJ option (VARMETHOD=BRR), 6480
DFADJ option (VARMETHOD=JK), 6482
FAY= option (VARMETHOD=BRR), 6480
H= option (VARMETHOD=BRR), 6480
HADAMARD= option (VARMETHOD=BRR), 6480
MISSING option, 6474
N= option, 6477
NOMCAR option, 6474
NONSYMCL option, 6475
NOSPARSE option, 6475
ORDER= option, 6475
OUTJKCOEFS= option (VARMETHOD=JK), 6483
OUTWEIGHTS= option (VARMETHOD=BRR), 6481
OUTWEIGHTS= option (VARMETHOD=JK), 6483
PERCENTILE= option, 6475
PRINTH option (VARMETHOD=BRR), 6481
QUANTILE= option, 6475
R= option, 6476
RATE= option, 6476
REPS= option (VARMETHOD=BRR), 6482
STACKING option, 6476
TOTAL= option, 6477
VARMETHOD= option, 6479
SURVEYMEANS procedure, RATIO statement, 6486
SURVEYMEANS procedure, REPWEIGHTS statement, 6488
DF= option, 6488
JKCOEFS= option, 6488
SURVEYMEANS procedure, STRATA statement, 6489
LIST option, 6490
SURVEYMEANS procedure, VAR statement, 6490
SURVEYMEANS procedure, WEIGHT statement, 6490
SURVEYREG procedure
syntax, 6542
SURVEYREG procedure, BY statement, 6549
SURVEYREG procedure, CLASS statement, 6549
ORDER= option, 6544

SURVEYREG procedure, CLUSTER statement, 6550
SURVEYREG procedure, CONTRAST statement, 6551
  E option, 6551
  NOFILL option, 6551
  SINGULAR= option, 6552
SURVEYREG procedure, DOMAIN statement, 6552
SURVEYREG procedure, ESTIMATE statement, 6553
  DIVISOR= option, 6554
  E option, 6554
  NOFILL option, 6554
  SINGULAR= option, 6554
SURVEYREG procedure, MODEL statement, 6555
  ADJRSQ option, 6555
  ANOVA option, 6555
  CLPARM option, 6555
  COVB option, 6555
  DEFF option, 6555
  INVERSE option, 6555
  NOINT option, 6556
  PARMLABEL option, 6556
  SINGULAR= option, 6556
  SOLUTION option, 6556
  VADJUST= option, 6556
  XPX option, 6556
SURVEYREG procedure, MODEL statement (SURVEYREG)
  DF= option, 6555
SURVEYREG procedure, OUTPUT statement, 6556
  ALPHA= option, 6557
  keyword= option, 6557
  LCLM keyword, 6557
  OUT= option, 6556
  PREDICTED keyword, 6557
  RESIDUAL keyword, 6557
  STD keyword, 6557
  STDP keyword, 6557
  UCLM keyword, 6557
SURVEYREG procedure, PROC SURVEYREG statement, 6543
  ALPHA= option, 6543
  DATA= option, 6543
  FAY= option (VARMETHOD=BRR), 6546
  H= option (VARMETHOD=BRR), 6546
  HADAMARD= option (VARMETHOD=BRR), 6546
  MISSING option, 6543
  N= option, 6545
  NOMCAR option, 6543
  OUTJKCOEFS= option (VARMETHOD=JK), 6548
  OUTWEIGHTS= option (VARMETHOD=BRR), 6547
  OUTWEIGHTS= option (VARMETHOD=JK), 6548
  PRINTH option (VARMETHOD=BRR), 6547
  R= option, 6544
  RATE= option, 6544
  REPS= option (VARMETHOD=BRR), 6548
  TOTAL= option, 6545
  TRUNCATE option, 6545
  VARMETHOD= option, 6545
SURVEYREG procedure, REPWEIGHTS statement, 6558
  DF= option, 6558
  JKCOEFS= option, 6558
SURVEYREG procedure, STRATA statement, 6559
  LIST option, 6560
  NOCOLLAPSE option, 6560
SURVEYREG procedure, WEIGHT statement, 6560
SURVEYSELECT procedure
  syntax, 6623
SURVEYSELECT procedure, CONTROL statement, 6639
SURVEYSELECT procedure, ID statement, 6639
SURVEYSELECT procedure, PROC SURVEYSELECT statement, 6623
  CERTSIZE= option, 6625
  CERTSIZE=P= option, 6626
  DATA= option, 6627
  JTPROBS option, 6627
  MAXSIZE= option, 6628
  METHOD= option, 6629
  MINSIZE= option, 6631
  NMAX= option, 6632
  NMIN= option, 6632
  NOPRINT option, 6632
  OUT= option, 6632
  OUTALL option, 6633
  OUTHITS option, 6633
  OUTSEED option, 6633
  OUTSIZE option, 6633
  OUTSORT= option, 6634
  REPS= option, 6634
  SAMPRATE= option, 6634
  SAMPSIZE= option, 6636
  SEED= option, 6637
  SELECTALL option, 6638
  SORT= option, 6638

STATS option, 6638
SURVEYSELECT procedure, SIZE statement, 6639
SURVEYSELECT procedure, STRATA statement, 6640
ALLOC= option, 6641
COST= option, 6642
NOSAMPLE option, 6643
VAR= option, 6643
SYMBOL= option
MCMC statement (MI), 3758, 3763
PAINT statement (REG), 5486
PLOT statement (REG), 5498
SYMBOLLEGEND= option
PLOT statement (BOXPLOT), 788
SYMBOLORDER= option
PLOT statement (BOXPLOT), 788
SYMBOLS option
OUTPUT statement (GLIMMIX), 2180
syntax
GLIMMIX procedure, 2093
LIFETEST procedure, 3110

T option
MEANS statement (GLM), 2473
MODEL statement (LOESS), 3213
PROC CANCORR statement, 1042
T= option
PROC ACECLUS statement, 678
TABLEOUT option
PROC REG statement, 5467
TABLES statement
CORRESP procedure, 1332
FREQ procedure, 1697
SURVEYFREQ procedure, 6312
TARGACCEPT= option
PROC MCMC statement, 3508
TARGACCEPTI= option
PROC MCMC statement, 3508
TARGET= option
PROC FACTOR statement, 1563
TAU= option
FACTOR statement, 6756
PROC FACTOR statement, 1563
PROC NLIN statement, 4283
TCALIS procedure, 6715
NLOPTIONS statement, 391
syntax, 6715
TCALIS procedure, BASEMODEL statement, 6745
TCALIS procedure, BOUNDS statement, 6746
TCALIS procedure, BY statement, 6746
TCALIS procedure, COV statement, 6747
TCALIS procedure, DETERM statement, 6750
TCALIS procedure, EFFPART statement, 6751
TCALIS procedure, FACTOR statement, 6752
COMPONENT option, 6753
GAMMA= option, 6753
HEYWOOD option, 6753
N= option, 6754
NORM option, 6754
RCONVERGE= option, 6754
RITER= option, 6754
ROTATE= option, 6754
TCALIS procedure, FITINDEX statement
ALPHAECV= option, 6760
ALPHARMSEA= option, 6760
CHICORRECT= option, 6760
CLOSEFIT= option, 6760
DFREDUCE= option, 6760
NOADJDF option, 6760
NOINDEXTYPE option, 6760
OFFLIST= option, 6760
ONLIST= option, 6761
OUTFIT= option, 6763
TCALIS procedure, FREQ statement, 6763
TCALIS procedure, GROUP statement, 6764
LABEL= option, 6765
NAME= option, 6765
TCALIS procedure, LINCON statement, 6766
TCALIS procedure, LINEQS statement, 6766
TCALIS procedure, LISMOD statement, 6772
TCALIS procedure, LMTESTS statement, 6776
default option, 6776
lmmat option, 6777
maxrank option, 6776
nodefault option, 6776
norank option, 6776
TCALIS procedure, main model specification statements, 6717
TCALIS procedure, MATRIX statement, 6784
TCALIS procedure, MEAN statement, 6788
TCALIS procedure, model analysis statements, 6718
TCALIS procedure, MODEL statement, 6789
GROUP= option, 6790
GROUPS= option, 6790
LABEL= option, 6790
NAME= option, 6790
TCALIS procedure, MSTRUCT statement, 6792
TCALIS procedure, NLINCON statement, 6794
TCALIS procedure, NLOPTIONS statement, 6795
ABSCONV option, 393
ABSFCONV option, 393
ABSGCONV option, 393
ABSGTOL option, 393
ABSTOL option, 393

ABSXCONV option, 393
ABSXTOL option, 393
ASINGULAR= option, 394
FCONV option, 394
FCONV2 option, 395
FSIZE option, 395
FTOL option, 394
FTOL2 option, 395
GCONV option, 395
GCONV2 option, 396
GTOL option, 395
GTOL2 option, 396
HESCAL option, 396
HS option, 396
INHESSIAN option, 396
LCDEACT= option, 397
LCEPSILON= option, 398
LCSINGULAR= option, 398
LINESEARCH option, 398
LIS option, 398
LSP option, 399
LSPRECISION option, 399
MAXFU option, 399
MAXFUNC option, 399
MAXIT option, 399
MAXITER option, 399
MAXSTEP option, 400
MAXTIME option, 400
MINIT option, 400
MINITER option, 400
MSINGULAR= option, 400
REST option, 401
RESTART option, 401
SINGULAR= option, 401
SOCKET option, 401
TECH option, 401
TECHNIQUE option, 401
UPD option, 402
VSINGULAR= option, 403
XSIZE option, 403
XTOL option, 403
TCALIS procedure, optimization statements, 6718
TCALIS procedure, OUTFILE statement, 6796
TCALIS procedure, OUTFILES statement, 6796
TCALIS procedure, PARAMETERS statement, 6798
TCALIS procedure, PARTIAL statement, 6798
TCALIS procedure, PATH statement, 6799
TCALIS procedure, PCOV statement, 6802
TCALIS procedure, PROC TCALIS statement, 6721
ALL option, 6740
ALPHAECV= option, 6726
ALPHARMS= option, 6726
ASINGULAR= option, 6726
ASYCOV= option, 6726
BIASKUR option, 6727
CHICORR= option, 6727
CHICORRECT= option, 6727
CLOSEFIT option, 6728
CORR option, 6740
CORRELATION option, 6728
COVARIANCE option, 6728
COVSING= option, 6728
DATA= option, 6728
DEMPHAS= option, 6728
DFE= option, 6729
DFR= option, 6742
DFREDUCE= option, 6729
EDF= option, 6729
ESTDATA= option, 6730
FCONV= option, 6729
FTOL= option, 6729
G4= option, 6729
GCONV= option, 6730
GTOL= option, 6730
INEST= option, 6730
INMODEL= option, 6730
INRAM= option, 6730
INSTEP= option, 6730
INVAR= option, 6730
INWGT= option, 6731
INWGTINV option, 6731
KURTOSIS option, 6731
LINESEARCH= option, 6731
LSPRECISION= option, 6732
MAXFUNC= option, 6732
MAXITER= option, 6733
MEANSTR option, 6733
METHOD= option, 6733
MODIFICATION option, 6734
MSINGULAR= option, 6734
NOADJDF option, 6735
NOBS= option, 6735
NOINDEXTYPE option, 6735
NOMOD option, 6735
NOORDERSPEC option, 6735
NOPARMNAME option, 6735
NOPRINT option, 6735
NOSTDERR option, 6735, 6736
OM= option, 6736
OMETHOD= option, 6736
ORDERALL option, 6738
ORDERGROUPS option, 6738
ORDERMODELS option, 6738
ORDERSPEC option, 6738
OUTEST= option, 6738

OUTFIT option, 6739
OUTRAM= option, 6739
OUTSTAT= option, 6739
OUTVAR= option, 6738
OUTWGT= option, 6739
PALL option, 6740
PARMNAME option, 6740
PCORR option, 6740
PCOVES option, 6740
PDETERM option, 6740
PESTIM option, 6741
PINITIAL option, 6741
PLATCOV option, 6741
PLOTS= option, 6741
PRIMAT option, 6742
PRINT option, 6742
PSHORT option, 6742
PSUMMARY option, 6742
PWEIGHT option, 6742
RADIUS= option, 6742
RANDOM= option, 6742
RDF= option, 6742
RESIDUAL= option, 6742
RIDGE= option, 6743
SALPHA= option, 6743
SHORT option, 6742
SIMPLE option, 6743
SINGULAR= option, 6743
SLMW= option, 6743
SMETHOD= option, 6731
SPRECISION= option, 6732, 6743
START= option, 6743
STDERR option, 6744
SUMMARY option, 6742
TECHNIQUE= option, 6736
TOTEFF option, 6729
UPDATE= option, 6737
VARDEF= option, 6744
VSINGULAR= option, 6744
WPENALTY= option, 6744
WRIDGE= option, 6745
TCALIS procedure, PVAR statement, 6803
TCALIS procedure, RAM statement, 6805
TCALIS procedure, REFMODEL statement, 6811
ALLNEWPARMS option, 6812
PARM_PREFIX option, 6812
PARM_SUFFIX option, 6812
TCALIS procedure, RENAMEPARM statement, 6812
TCALIS procedure, SIMTESTS statement, 6814
TCALIS procedure, STD statement, 6815
TCALIS procedure, STRUCTEQ statement, 6817
TCALIS procedure, subsidiary group specification statements, 6716
TCALIS procedure, subsidiary model specification statements, 6718
TCALIS procedure, TESTFUNC statement, 6817
TCALIS procedure, VAR statement, 6818
TCALIS procedure, WEIGHT statement, 6818
TCALIS procedure,FACTOR statement
TAU= option, 6756
TCALISprocedure, NLOPTIONS statement
INSTEP option, 397
TCORR option
PROC CANDISC statement, 1071
PROC DISCRIM statement, 1396
PROC STEPDISC statement, 6277
TCOV option
PROC CANDISC statement, 1072
PROC DISCRIM statement, 1396
PROC MIANALYZE statement, 3847
PROC STEPDISC statement, 6277
TEST statement (MIANALYZE), 3850
TDATA= option
PRIOR statement (MIXED), 3949
TDIFF option
LSMEANS statement (GLM), 2463
TDPREFIX= option
OUTPUT statement (TRANSREG), 7178
TECH option
NLOPTIONS statement (GLIMMIX), 401
NLOPTIONS statement (HPMIXED), 401
NLOPTIONS statement (TCALIS), 401
TECHNIQUE option
NLOPTIONS statement (GLIMMIX), 401
NLOPTIONS statement (HPMIXED), 401
NLOPTIONS statement (TCALIS), 401
TECHNIQUE= option
MODEL statement (LOGISTIC), 3302
MODEL statement (SURVEYLOGISTIC), 6401
NLOPTIONS statement (CALIS), 889
PROC CALIS statement, 867
PROC NLMIXED statement, 4370
PROC TCALIS statement, 6736
TEMPLATE procedure
SOURCE statement, 557
TEST option
MODEL statement (TRANSREG), 7166
PROC MODECLUS statement, 4108
RANDOM statement (GLM), 2481
TEST statement
ANOVA procedure, 719
FREQ procedure, 1722
GLM procedure, 2485
HPMIXED procedure, 2831

LIFETEST procedure, 3130
LOGISTIC procedure, 3315
MIANALYZE procedure, 3849
MULTTEST procedure, 4201
PHREG procedure, 4577
QUANTREG procedure, 5383
REG procedure, 5506
ROBUSTREG procedure, 5671
SURVEYLOGISTIC procedure, 6408
TEST= option
LOGISTIC statement (POWER), 4831
MULTREG statement (POWER), 4839
ONECORR statement (POWER), 4843
ONESAMPLEFREQ statement (POWER), 4849
ONESAMPLEMEANS statement (POWER), 4857
ONEWAYANOVA statement (POWER), 4862
PAIREDFREQ statement (POWER), 4868
PAIREDMEANS statement (POWER), 4877
PROC TTEST statement, 7407
STRATA statement (LIFETEST), 3129
TWOSAMPLEFREQ statement (POWER), 4888
TWOSAMPLEMEANS statement (POWER), 4897
TWOSAMPLESURVIVAL statement (POWER), 4907
TWOSAMPLEWILCOXON statement (POWER), 4913
TESTCLASS statement, DISCRIM procedure, 1400
TESTDATA= option
PROC DISCRIM statement, 1396
PROC GLMSELECT statement, 2701
TESTF= option
TABLES statement (FREQ), 1721, 1732
TESTFREQ statement, DISCRIM procedure, 1400
TESTFUNC statement, TCALIS procedure, 6817
TESTID statement, DISCRIM procedure, 1400
TESTLIST option
PROC DISCRIM statement, 1396
TESTLISTERR option
PROC DISCRIM statement, 1396
TESTODDSRATIO= option
LOGISTIC statement (POWER), 4832
TESTOUT= option
PROC DISCRIM statement, 1396
TESTOUTD= option
PROC DISCRIM statement, 1396
TESTP= option
TABLES statement (FREQ), 1721, 1732, 1806
TABLES statement (SURVEYFREQ), 6319
TESTPREDICTOR= option
LOGISTIC statement (POWER), 4832
TESTREGCOEFF= option
LOGISTIC statement (POWER), 4832
THETA0= option
PROC MI statement, 3752
PROC MIANALYZE statement, 3847
THIN= option
PROC MCMC statement, 3508
THINNING= option
BAYES statement(GENMOD), 1937
BAYES statement(LIFEREG), 3015
BAYES statement(PHREG), 4556
THREADS option
PERFORMANCE statement (GLMSELECT), 2717
PERFORMANCE statement (QUANTREG), 5383
PERFORMANCE statement (ROBUSTREG), 5671
THRESHOLD= option
PROC ACECLUS statement, 678
PROC DISCRIM statement, 1397
PROC MODECLUS statement, 4108
TICKPOS= option
PROC TREE statement, 7366
TIES= option
MODEL statement (PHREG), 4572
TIME statement
LIFETEST procedure, 3131
TIME= option
TEST statement (MULTTEST), 4202
TIMELIM= option
PROC LIFETEST statement, 3123
TIMELIST= option
BASELINE statement (PHREG), 4541
PROC LIFETEST statement, 3124
TIMEPLOT option
MCMC statement (MI), 3762
TIPMAX= option
ODS GRAPHICS statement, 531
TIPREFIX option
OUTPUT statement (TRANSREG), 7178
TITLE= option
FACTORS statement (CATMOD), 1116
LOGLIN statement (CATMOD), 1117
MCMC statement (MI), 3758, 3763
MODEL statement (CATMOD), 1124
REPEATED statement (CATMOD), 1130
RESPONSE statement (CATMOD), 1132
TOL option

MODEL statement (REG), 5480
TOLERANCE option
MODEL statement (GLM), 2477
PROC ROBUSTREG statement, 5666
TOLERANCE= option
COVTEST statement (GLIMMIX), 2134
PROC QUANTREG statement, 5373
TOST option
PROC TTEST statement, 7408
TOTAL= option
PROC SURVEYFREQ statement, 6304
PROC SURVEYLOGISTIC statement, 6384
PROC SURVEYMEANS statement, 6477
PROC SURVEYREG statement, 6545
TOTALPROPDISC= option
PAIREDFREQ statement (POWER), 4868
TOTALTIME= option
TWOSAMPLESURVIVAL statement (POWER), 4907
TOTEFF option
PROC CALIS statement, 869
PROC TCALIS statement, 6729
TOTPANELS= option
PLOT statement (BOXPLOT), 788
TOTPCT option
TABLES statement (FREQ), 1721
TPREFIX= option
PROC PRINQUAL statement, 5210
TPSPLINE procedure
syntax, 7068
TPSPLINE procedure, BY statement, 7068
TPSPLINE procedure, FREQ statement, 7069
TPSPLINE procedure, ID statement, 7069
TPSPLINE procedure, MODEL statement, 7069
ALPHA= option, 7070
DF= option, 7070
DISTANCE= option, 7070
LAMBDA0= option, 7070
LAMBDA= option, 7071
LOGNLAMBDA0= option, 7071
LOGNLAMBDA= option, 7071
M= option, 7071
TPSPLINE procedure, OUTPUT statement, 7072
OUT= option, 7072
TPSPLINE procedure, PROC TPSPLINE statement, 7068
DATA= option, 7068
TPSPLINE procedure, SCORE statement, 7071
DATA= option, 7072
OUT= option, 7072
TRACE option
MCMC statement (MI), 3761
PROC MCMC statement, 3508
PROC MODECLUS statement, 4108
PROC NLIN statement, 4283
PROC NLMIXED statement, 4371
PROC VARCLUS statement, 7473
TRACEL option
MODEL statement (LOESS), 3213
TRANS= option
PRIOR statement (MIXED), 3949
TRANSFORM statement
MI procedure, 3767
TRANSFORM statement (PRINQUAL)
ARSIN transformation, 5214
DEGREE= option, 5218
EVENLY option, 5218
EXP transformation, 5214
IDENTITY transformation, 5216
KNOTS= option, 5219
LINEAR transformation, 5215
LOG transformation, 5214
LOGIT transformation, 5214
MONOTONE transformation, 5215
MSPLINE transformation, 5215
NAME= option, 5219
NKNOTS= option, 5219
OPSCORE transformation, 5215
ORIGINAL option, 5217
PARAMETER= option, 5217
POWER transformation, 5214
RANK transformation, 5214
SPLINE transformation, 5215
SSPLINE transformation, 5216
TSTANDARD= option, 5220
UNTIE transformation, 5216
TRANSREG, 7106
TRANSREG procedure
syntax, 7129
TRANSREG procedure, BY statement, 7137
TRANSREG procedure, FREQ statement, 7138
TRANSREG procedure, ID statement, 7138
TRANSREG procedure, MODEL statement, 7138
ADDITIVE option, 7160
AFTER option, 7157
AIC option, 7152
AICC option, 7152
ALPHA= option, 7156, 7160
ARSIN transformation, 7143
Box-Cox transformation, 7144
BSPLINE transformation, 7142
CCONVERGE= option, 7160
CENTER option, 7157
CL option, 7160
CLASS transformation, 7142
CLL= option, 7156
CONVENIENT option, 7156

CONVERGE option, 7160
CPREFIX= option, 7153, 7161
CV option, 7152
DEGREE= option, 7150
DETAIL option, 7161
DEVIATIONS option, 7153
DUMMY option, 7161
EFFECTS option, 7153
EPOINT transformation, 7142
EVENLY option, 7150
EXKNOTS= option, 7151
EXP transformation, 7144
GCV option, 7152
GEOMETRICMEAN option, 7156
HISTORY option, 7162
IDENTITY transformation, 7147
INDIVIDUAL option, 7162
KNOTS= option, 7151
LAMBDA= option, 7152, 7156
LINEAR transformation, 7145
LOG transformation, 7144
LOGIT transformation, 7144
LPREFIX= option, 7153, 7162
MAXITER= option, 7162
METHOD= option, 7162
MONOTONE transformation, 7146
MONOTONE= option, 7163
MSPLINE transformation, 7146
NAME= option, 7157
NCAN= option, 7163
NKNOTS= option, 7151
NOINT option, 7163
NOMISS option, 7164
NOPRINT option, 7164
NSR option, 7164
OPSCORE transformation, 7146
ORDER= option, 7154, 7165
ORIGINAL option, 7149
ORTHOGONAL option, 7154
PARAMETER= option, 7149
PBOXCOXTABLE option, 7165
PBSPLINE transformation, 7145
POINT transformation, 7143
POWER transformation, 7144
PSPLINE transformation, 7143
QPOINT transformation, 7143
RANGE option, 7152
RANK transformation, 7144
REFERENCE= option, 7165
REFLECT option, 7158
REITERATE option, 7165
RSQUARE option, 7166
SBC option, 7153
SEPARATORS= option, 7154, 7166
SHORT option, 7166
SINGULAR= option, 7166
SM= option, 7150
SMOOTH transformation, 7145
SOLVE option, 7161
SPLINE transformation, 7146
SS2 option, 7166
SSPLINE transformation, 7147
STANDORTH option, 7154
TEST option, 7166
TSTANDARD= option, 7158, 7167
TSUFFIX= option, 7166
TYPE= option, 7167
UNTIE transformation, 7146
UNTIE= option, 7168
UTILITIES option, 7168
Z option, 7158
ZERO= option, 7155
TRANSREG procedure, OUTPUT statement, 7168
ADPREFIX= option, 7170
AIPREFIX option, 7170
APPROXIMATIONS option, 7170
CANONICAL option, 7171
CCC option, 7171
CDPREFIX= option, 7171
CEC option, 7171
CILPREFIX= option, 7171
CIPREFIX= option, 7171
CIUPREFIX= option, 7171
CLI option, 7171
CLM option, 7171
CMLPREFIX= option, 7172
CMUPREFIX= option, 7172
COEFFICIENTS option, 7172
COORDINATES= option, 7172
CPC option, 7172
CQC option, 7172
DAPPROXIMATIONS option, 7173
DEPENDENT= option, 7173
DESIGN= option, 7173
DREPLACE option, 7173
IAPPROXIMATIONS option, 7173
IREPLACE option, 7174
LEVERAGE= option, 7174
LILPREFIX= option, 7174
LIUPREFIX= option, 7174
LMLPREFIX= option, 7174
LMUPREFIX= option, 7174
MACRO option, 7174
MEANS option, 7176
MEC option, 7176
MPC option, 7176
MQC option, 7176

MRC option, 7177
MREDUNDANCY option, 7177
NORESTOREMISSING option, 7177
NOSCORES option, 7177
NOZEROCONSTANT option, 7164
OUT= option, 7168
PPREFIX option, 7177
PREDICTED option, 7177
RDPREFIX= option, 7177
REDUNDANCY= option, 7178
REFERENCE= option, 7178
REPLACE option, 7178
RESIDUALS option, 7178
RPREFIX= option, 7178
TDPREFIX= option, 7178
TIPREFIX option, 7178
TRANSREG procedure, PROC TRANSREG statement, 7130
DATA= option, 7133
OUTTEST= option, 7133
PLOTS= option, 7133
TRANSREG procedure, WEIGHT statement, 7179
TREATMENTS statement
PLAN procedure, 4742
TREE procedure
syntax, 7361
TREE procedure, BY statement, 7367
TREE procedure, COPY statement, 7367
TREE procedure, FREQ statement, 7368
TREE procedure, HEIGHT statement, 7368
TREE procedure, ID statement, 7368
TREE procedure, NAME statement, 7368
TREE procedure, PARENT statement, 7369
TREE procedure, PROC TREE statement, 7361
CFRAME= option, 7362
DATA= option, 7363
DESCENDING option, 7363
DESCRIPTION= option, 7363
DISSIMILAR option, 7363
DOCK= option, 7363
FILLCHAR= option, 7363
GOUT= option, 7363
HAXIS= option, 7363
HEIGHT= option, 7363
HORDISPLAY= option, 7364
HORIZONTAL option, 7364
HPAGES= option, 7364
INC= option, 7364
JOINCHAR= option, 7364
LEAFCHAR= option, 7364
LEVEL= option, 7365
LINEPRINTER option, 7365
LINES= option, 7365
LIST option, 7365
MAXHEIGHT= option, 7365
MINHEIGHT= option, 7365
NAME= option, 7365
NCLUSTERS= option, 7365
NOPRINT option, 7366
NTICK= option, 7365
OUT= option, 7366
PAGES= option, 7366
POS= option, 7366
ROOT= option, 7366
SIMILAR option, 7366
SORT option, 7366
SPACES= option, 7366
TICKPOS= option, 7366
TREECHAR= option, 7367
VAXIS= option, 7367
VPAGES= option, 7367
TREECHAR= option
PROC TREE statement, 7367
TREEINFO suboption
RANDOM statement (GLIMMIX), 2190
TREND option
EXACT statement (FREQ), 1692, 1819
OUTPUT statement (FREQ), 1697
STRATA statement (LIFETEST), 3129
TABLES statement (FREQ), 1722, 1819
TRIM= option
and other options, 1242
PROC CLUSTER statement, 1242, 1246
TRUNCATE option
CLASS statement (ANOVA), 705
CLASS statement (GAM), 1841
CLASS statement (GENMOD), 1939
CLASS statement (GLIMMIX), 2122
CLASS statement (GLM), 2454
CLASS statement (GLMMOD), 2628
CLASS statement (HPMIXED), 2815
CLASS statement (LOGISTIC), 3281
CLASS statement (MULTTEST), 4198
CLASS statement (NESTED), 4258
CLASS statement (ORTHOREG), 4511
CLASS statement (PLS), 4782
PROC LOGISTIC statement, 3275
PROC SURVEYREG statement, 6545
TSSCP option
PROC CANDISC statement, 1072
PROC DISCRIM statement, 1397
PROC STEPDISC statement, 6277
TSTANDARD= option
MODEL statement (TRANSREG), 7158, 7167
PROC PRINQUAL statement, 5210

TRANSFORM statement (PRINQUAL), 5220
TSUFFIX= option
MODEL statement (TRANSREG), 7166
TSYMM option
OUTPUT statement (FREQ), 1697
TTEST procedure
syntax, 7400
TTEST procedure, BY statement, 7408
TTEST procedure, CLASS statement, 7408
TTEST procedure, FREQ statement, 7409
TTEST procedure, PAIRED statement, 7409
TTEST procedure, PROC TTEST statement, 7401
ALPHA= option, 7401
BYVAR option, 7401
CI= option, 7402
COCHRAN option, 7402
CROSSOVER= option, 7410
DATA= option, 7402
DIST= option, 7402
H0= option, 7402
IGNOREPERIOD option, 7410
NOBYVAR option, 7402
ORDER= option, 7402
PLOTS option, 7403
SIDES= option, 7407
TEST= option, 7407
TOST option, 7408
TTEST procedure, VAR statement, 7410
TTEST procedure, WEIGHT statement, 7410
TTOTALSS option
PROC NLIN statement, 4283
TUKEY option
MEANS statement (ANOVA), 714
MEANS statement (GLM), 2473
TUNEWT= option
PROC MCMC statement, 3508
TURNHLABELS option
PLOT statement (BOXPLOT), 788
TWOSAMPLEFREQ statement
POWER procedure, 4884
TWOSAMPLEMEANS statement
POWER procedure, 4891
TWOSAMPLESURVIVAL statement
POWER procedure, 4901
TWOSAMPLEWILCOXON statement
POWER procedure, 4911
TYPE1 option
MODEL statement (GENMOD), 1953
TYPE3 option
MODEL statement (GENMOD), 1953
TYPE= option
MODEL statement (TRANSREG), 7167
PROC PRINQUAL statement, 5210
PROC SCORE statement, 5767
RANDOM statement (GLIMMIX), 2193
RANDOM statement (HPMIXED), 2831
RANDOM statement (MIXED), 3952
REPEATED statement (GENMOD), 1961
REPEATED statement (MIXED), 3959
TYPE=ACE
Data set option, 7609
TYPE=BOXPLOT
Data set option, 7609
TYPE=CHARTSUM
Data set option, 7609
TYPE=CORR
Data set option, 7609
TYPE=COV
Data set option, 7613
TYPE=CSSCP
Data set option, 7613
TYPE=DISTANCE
Data set option, 7614
TYPE=EST
Data set option, 7614
TYPE=LINEAR
Data set option, 7616
TYPE=LOGISMOD
Data set option, 7616
TYPE=MIXED
Data set option, 7616
TYPE=QUAD
Data set option, 7616
TYPE=RAM
Data set option, 7617
TYPE=SSCP
Data set option, 7617
TYPE=TREE
Data set option, 7618
TYPE=UCORR
Data set option, 7618
TYPE=UCOV
Data set option, 7619
TYPE=WEIGHT
Data set option, 7619

U option
OUTPUT statement (FREQ), 1697
U95 option
MODEL statement (RSREG), 5733
U95= option
OUTPUT statement (NLIN), 4288
U95M option
MODEL statement (RSREG), 5733
U95M= option
OUTPUT statement (NLIN), 4288

UCL keyword
OUTPUT statement (GLM), 2479
UCL= option
OUTPUT statement (HPMIXED), 2824
OUTPUT statement (NLIN), 4288
UCLM keyword
OUTPUT statement (GLM), 2479
OUTPUT statement (SURVEYREG), 6557
UCLM= option
OUTPUT statement (NLIN), 4288
UCORR option
PROC CALIS statement, 869
UCOV option
PROC CALIS statement, 869
UCR option
OUTPUT statement (FREQ), 1697
ULTRAHEYWOOD option
PROC FACTOR statement, 1563
UNADJUSTED option
PROC CORRESP statement, 1330
UNDEF= option
PROC DISTANCE statement, 1499
UNDO option
PAINT statement (REG), 5487
REWEIGHT statement (REG), 5505
UNISTATS option
BIVAR statement, 2893
UNIVAR statement, 2895
UNITS statement, LOGISTIC procedure, 3316
UNITS statement, SURVEYLOGISTIC procedure, 6409
UNITS= option
HAZARDRATIO statement (PHREG), 4563
LOGISTIC statement (POWER), 4832
UNIVAR statement
KDE procedure, 2893
unpack option
PROC GAM statement, 1839
UNPACKPANELS option
PROC GAM statement, 1840
UNSTD option
PROC STDIZE statement, 6225
UNTIE option
PROC MDS statement, 3721
UNTIE transformation
MODEL statement (TRANSREG), 7146
TRANSFORM statement (PRINQUAL), 5216
UNTIE= option
MODEL statement (TRANSREG), 7168
PROC PRINQUAL statement, 5211
UPD option
NLOPTIONS statement (GLIMMIX), 402
NLOPTIONS statement (HPMIXED), 402
NLOPTIONS statement (TCALIS), 402
UPDATE option
NLOPTIONS statement (GLIMMIX), 402
NLOPTIONS statement (HPMIXED), 402
NLOPTIONS statement (TCALIS), 402
PROC MIXED statement, 3915
UPDATE= option
NLOPTIONS statement (CALIS), 889
PRIOR statement (MIXED), 3949
PROC CALIS statement, 869
PROC NLMIXED statement, 4371
PROC TCALIS statement, 6737
UPPER= option
ONESAMPLEFREQ statement (POWER), 4849
ONESAMPLEMEANS statement (POWER), 4857
OUTPUT statement (LOGISTIC), 3308
OUTPUT statement (SURVEYLOGISTIC), 6404
PAIREDMEANS statement (POWER), 4877
TWOSAMPLEMEANS statement (POWER), 4897
UPPERB= option
PARMS statement (GLIMMIX), 2185
PARMS statement (HPMIXED), 2829
PARMS statement (MIXED), 3945
UPPERTAILED option
ESTIMATE statement (GLIMMIX), 2140
ESTIMATE statement (MIXED), 3922
LSMESTIMATE statement (GLIMMIX), 2161
TEST statement (MULTTEST), 4202
URC option
OUTPUT statement (FREQ), 1697
URL= suboption
ODS HTML statement, 545
USEALL option
PLOT statement (REG), 5497
USSCP option
PROC REG statement, 5467
UTILITIES option
MODEL statement (TRANSREG), 7168

V option
RANDOM statement (GLIMMIX), 2205
RANDOM statement (MIXED), 3953
V6CORR option
REPEATED statement (GENMOD), 1961
VADJUST= option
MODEL statement (SURVEYLOGISTIC), 6401
MODEL statement (SURVEYREG), 6556

VALDATA= option
PROC GLMSELECT statement, 2701
VAR option
TABLES statement (SURVEYFREQ), 6319
VAR statement
CALIS procedure, 907
CANDISC procedure, 1073
CORRESP procedure, 1332
DISCRIM procedure, 1400
FACTOR procedure, 1565
INBREED procedure, 2867
LATTICE procedure, 2983
MDS procedure, 3722
MI procedure, 3768
NESTED procedure, 4258
NPAR1WAY procedure, 4466
PRINCOMP procedure, 5158
REG procedure, 5507
STDIZE procedure, 6227
STEPDISC procedure, 6279
SURVEYMEANS procedure, 6490
TCALIS procedure, 6818
TTEST procedure, 7410
VAR statement (SIMNORMAL), 6205
VARCLUS procedure, 7475
VARIOGRAM procedure, 7539
VAR= option
ASSESS statement (PHREG), 4540
CDFPLOT statement (PROBIT), 5269
IPPPLOT statement (PROBIT), 5279
LPREDPLOT statement (PROBIT), 5287
PREDICT statement (KRIGE2D), 2944
PREDPPLOT statement (PROBIT), 5300
SIMULATE statement (SIM2D), 6171
STRATA statement (SURVEYSELECT), 6643
VARCLUS procedure
syntax, 7468
VARCLUS procedure, BY statement, 7474
VARCLUS procedure, FREQ statement, 7474
VARCLUS procedure, PARTIAL statement, 7475
VARCLUS procedure, PROC VARCLUS statement, 7468
CENTROID option, 7470
CORR option, 7470
COVARIANCE option, 7470
DATA= option, 7470
HIERARCHY option, 7470
INITIAL= option, 7470
MAXCLUSTERS= option, 7471
MAXEIGEN= option, 7471
MAXITER= option, 7472
MAXSEARCH= option, 7472
MINC= option, 7472
MINCLUSTERS= option, 7472
MULTIPLEGROUP option, 7472
NOINT option, 7472
NOPRINT option, 7472
OUTSTAT= option, 7472
OUTTREE= option, 7472
PERCENT= option, 7473
PROPORTION= option, 7473
RANDOM= option, 7473
SHORT option, 7473
SIMPLE option, 7473
SUMMARY option, 7473
TRACE option, 7473
VARDEF= option, 7474
VARCLUS procedure, SEED statement, 7475
VARCLUS procedure, VAR statement, 7475
VARCLUS procedure, WEIGHT statement, 7475
VARCOMP procedure, 7497
BY statement, 7499
CLASS statement, 7499
MODEL statement, 7499
PROC VARCOMP statement, 7498
SYNTAX, 7497
VARCOMP procedure, BY statement, 7499
VARCOMP procedure, CLASS statement, 7499
VARCOMP procedure, MODEL statement, 7499
ALPHA= option, 7500
CL option, 7500
FIXED= option, 7500
VARCOMP procedure, PROC VARCOMP statement, 7498
DATA= option, 7498
EPSILON= option, 7498
MAXITER= option, 7498
METHOD= option, 7498
RATIO option, 7498
SPECLIMITS= option, 7498
VARDEF option
PROC STDIZE statement, 6225
VARDEF= option
PROC CALIS statement, 870
PROC DISTANCE statement, 1499
PROC FACTOR statement, 1563
PROC FASTCLUS statement, 1639
PROC PRINCOMP statement, 5156
PROC TCALIS statement, 6744
PROC VARCLUS statement, 7474
VARDIST= option
LOGISTIC statement (POWER), 4832
TWOSAMPLEWILCOXON statement (POWER), 4914
VAREST= option
ONESAMPLEFREQ statement (POWER), 4849

VARHEADER= option
PROC SURVEYFREQ statement, 6304
VARIABLES= option
TWOSAMPLEWILCOXON statement (POWER), 4914
VARIANCE statement, GENMOD procedure, 1962
VARIANCE= option
OUTPUT statement (HPMIXED), 2824
VARIOGRAM procedure, 7518
output data sets, 7518
syntax, 7527
VARIOGRAM procedure, BY statement, 7533
VARIOGRAM procedure, COMPUTE statement, 7533
ALPHA= option, 7534
ANGLETOLERANCE= option, 7534
AUTOCORRELATION option, 7534
AUTOCORRELATION STATISTICS= option, 7534
BANDWIDTH= option, 7536
CL option, 7536
DEPSILON= option, 7536
LAGDISTANCE= option, 7536
LAGTOLERANCE= option, 7537
MAXLAGS= option, 7537
NDIRECTIONS= option, 7537
NHCLASSES= option, 7538
NOVARIOGRAM option, 7538
OUTPDISTANCE= option, 7538
ROBUST option, 7538
VARIOGRAM procedure, COORDINATES statement, 7538
XCOORD= option, 7539
YCOORD= option, 7539
VARIOGRAM procedure, DIRECTIONS statement, 7539
VARIOGRAM procedure, PROC VARIOGRAM statement, 7529
DATA= option, 7529
NOPRINT option, 7529
OUTACWEIGHTS= option, 7529
OUTDISTANCE= option, 7529
OUTPAIR= option, 7529
OUTVAR= option, 7530
PLOTS option, 7530
PLOTS(ONLY) option, 7530
PLOTS(UNPACKPANEL) option, 7530
PLOTS=ALL option, 7531
PLOTS=EQUATE option, 7531
PLOTS=NONE option, 7531
PLOTS=OBSERVATIONS option, 7531
PLOTS=PAIRS option, 7532
PLOTS=SEMIVARIOGRAM option, 7532
VARIOGRAM procedure, VAR statement, 7539
VARMETHOD= option
PROC SURVEYFREQ statement, 6304
PROC SURVEYLOGISTIC statement, 6385
PROC SURVEYMEANS statement, 6479
PROC SURVEYREG statement, 6545
VARNAMES statement, CALIS procedure, 908
VARSCALE option
PROC PLS statement, 4781
VARWT option
TABLES statement (SURVEYFREQ), 6319
VARY option
PLOT statement (GLMPOWER), 2655
PLOT statement (POWER), 4882
VAXIS= option
PLOT statement (BOXPLOT), 788
PLOT statement (REG), 5497
PROC TREE statement, 7367
VC option
RANDOM statement (GLIMMIX), 2206
RANDOM statement (MIXED), 3953
VCI option
RANDOM statement (GLIMMIX), 2206
RANDOM statement (MIXED), 3953
VCIRY option
MIXED procedure, MODEL statement, 3988
MODEL statement (MIXED), 3942
VCORR option
RANDOM statement (GLIMMIX), 2206
RANDOM statement (MIXED), 3953
VDEP option
PROC CANCORR statement, 1042
VERSION= option
NLOPTIONS statement (CALIS), 898
VFORMAT= option
BOXPLOT procedure, 789
VI option
RANDOM statement (GLIMMIX), 2206
RANDOM statement (MIXED), 3953
VIF option
MODEL statement (REG), 5480
VMINOR= option
PLOT statement (BOXPLOT), 789
VN= option
PROC CANCORR statement, 1042
VNAME= option
PROC CANCORR statement, 1042
VOFFSET= option
PLOT statement (BOXPLOT), 789
VP= option
PROC CANCORR statement, 1042
VPAGES= option
PROC TREE statement, 7367

VPLOTS= option
PLOT statement (REG), 5499
VPREFIX= option
PROC CANCORR statement, 1042
VREF= option
PLOT statement (BOXPLOT), 789
PLOT statement (REG), 5497
VREFLABELS= option
PLOT statement (BOXPLOT), 789
VREFLABPOS= option
PLOT statement (BOXPLOT), 789
VREG option
PROC CANCORR statement, 1042
VSINGULAR= option
NLOPTIONS statement (CALIS), 890
NLOPTIONS statement (GLIMMIX), 403
NLOPTIONS statement (HPMIXED), 403
NLOPTIONS statement (TCALIS), 403
PROC CALIS statement, 870
PROC NLMIXED statement, 4372
PROC TCALIS statement, 6744
VW option
EXACT statement (NPAR1WAY), 4463
OUTPUT statement (NPAR1WAY), 4465
PROC NPAR1WAY statement, 4461
VZERO option
PLOT statement (BOXPLOT), 790

WALD option
CONTRAST statement (GENMOD), 1942
COVTEST statement (GLIMMIX), 2134
MODEL statement (GENMOD), 1954
WALDCI option
MODEL statement (GENMOD), 1954
WALDCL option
MODEL statement (LOGISTIC), 3302
WALDRL option
MODEL statement (LOGISTIC), 3299
WALLER option
MEANS statement (ANOVA), 714
MEANS statement (GLM), 2473
WAXIS= option
PLOT statement (BOXPLOT), 790
WCHISQ option
TABLES statement (SURVEYFREQ), 6320
WCONF= option
MCMC statement (MI), 3758
WCONNECT= option
MCMC statement (MI), 3763
WCORR option
PROC CANDISC statement, 1072
PROC DISCRIM statement, 1397
PROC STEPDISC statement, 6278
WCOV option
PROC CANDISC statement, 1072
PROC DISCRIM statement, 1397
PROC MIANALYZE statement, 3847
PROC STEPDISC statement, 6278
TEST statement (MIANALYZE), 3851
WDEP option
PROC CANCORR statement, 1042
WEIGHT option
PROC FACTOR statement, 1563
WEIGHT statement
CALIS procedure, 909
CANDISC procedure, 1073
CATMOD procedure, 1138
CORRESP procedure, 1333
DISCRIM procedure, 1401
FACTOR procedure, 1566
FREQ procedure, 1723
GENMOD procedure, 1962
GLIMMIX procedure, 2206
GLM procedure, 2486
GLMMOD procedure, 2629
GLMPOWER procedure, 2660
GLMSELECT procedure, 2718
HPMIXED procedure, 2832
KDE procedure, 2897
LIFEREG procedure, 3036
LOESS procedure, 3214
LOGISTIC procedure, 3317
MDS procedure, 3723
MIXED procedure, 3968
ORTHOREG procedure, 4512
PHREG procedure, 4578
PRINCOMP procedure, 5158
PRINQUAL procedure, 5220
QUANTREG procedure, 5383
REG procedure, 5507
ROBUSTREG procedure, 5672
RSREG procedure, 5735
STEPDISC procedure, 6279
SURVEYFREQ procedure, 6320
SURVEYLOGISTIC procedure, 6409
SURVEYMEANS procedure, 6490
SURVEYREG procedure, 6560
TCALIS procedure, 6818
TRANSREG procedure, 7179
TTEST procedure, 7410
VARCLUS procedure, 7475
WEIGHT= option
OUTPUT statement (NLIN), 4288
REWEIGHT statement (REG), 5505
STRATA statement (MULTTEST), 4200, 4206
WEIGHTFUNCTIONT option
PROC ROBUSTREG statement, 5664

WEIGHTS= option
VAR statement, 1504
WELCH option
MEANS statement (ANOVA), 714
MEANS statement (GLM), 2473
WGHT= option
COVTEST statement (GLIMMIX), 2134
WGRID= option
PLOT statement (BOXPLOT), 790
WGT statement
DISTANCE procedure, 1506
STDIZE procedure, 6227
WHERE statement
ANOVA procedure, 723
GLM procedure, 2491
WHITE option
MODEL statement (REG), 5480
WIDTH= option
ODS GRAPHICS statement, 531
PROC LIFETEST statement, 3124
WILCOXON option
EXACT statement (NPAR1WAY), 4463
OUTPUT statement (NPAR1WAY), 4465
PROC NPAR1WAY statement, 4461
WITHIN= option
REPEATED statement (GENMOD), 1961
WITHINSUBJECT= option
REPEATED statement (GENMOD), 1961
WLF option
MCMC statement (MI), 3757, 3763, 3764
WLLCHISQ option
TABLES statement (SURVEYFREQ), 6320
WLS option
MODEL statement (CATMOD), 1124
WN= option
PROC CANCORR statement, 1042
WNAME= option
PROC CANCORR statement, 1042
WNEEDLES= option
MCMC statement (MI), 3758
WOVERLAY= option
PLOT statement (BOXPLOT), 790
WP= option
PROC CANCORR statement, 1042
WPENALTY= option
PROC CALIS statement, 870
PROC TCALIS statement, 6744
WPREFIX= option
PROC CANCORR statement, 1042
WREF= option
MCMC statement (MI), 3758
WREG option
PROC CANCORR statement, 1042
WRIDGE= option
PROC CALIS statement, 871
PROC TCALIS statement, 6745
WSSCP option
PROC CANDISC statement, 1072
PROC DISCRIM statement, 1397
PROC STEPDISC statement, 6278
WTFREQ option
TABLES statement (SURVEYFREQ), 6320
WTKAP option
EXACT statement (FREQ), 1692
OUTPUT statement (FREQ), 1697
TEST statement (FREQ), 1723

X= option
GRID statement (KRIGE2D), 2942
GRID statement (SIM2D), 6170
PLOT statement (GLMPOWER), 2656
PLOT statement (POWER), 4882
XBETA keyword
OUTPUT statement (LIFEREG), 3027
OUTPUT statement (ROBUSTREG), 5670
XBETA= option
OUTPUT statement (LOGISTIC), 3308
OUTPUT statement (SURVEYLOGISTIC), 6404
XCONV option
EM statement (MI), 3755
NLOPTIONS statement (GLIMMIX), 403
NLOPTIONS statement (HPMIXED), 403
NLOPTIONS statement (TCALIS), 403
XCONV= option
MCMC statement (MI), 3760
MODEL statement (LOGISTIC), 3303
MODEL statement (SURVEYLOGISTIC), 6402
NLOPTIONS statement (CALIS), 894
PROC NLMIXED statement, 4372
XCOORD= option
COORDINATES statement (KRIGE2D), 2941
COORDINATES statement (SIM2D), 6170
GRID statement (KRIGE2D), 2942
GRID statement (SIM2D), 6171
XCOORD=option
COORDINATES statement (VARIOGRAM), 7539
XDATA= option
PROC LIFEREG statement, 3007
PROC PROBIT statement, 5268
XOPTS= option
PLOT statement (GLMPOWER), 2656
PLOT statement (POWER), 4883
XPVIX option
MODEL statement (MIXED), 3942

XPVIXI option
MODEL statement (MIXED), 3942
XPX option
MODEL statement (CATMOD), 1124
MODEL statement (GLM), 2477
MODEL statement (REG), 5480
MODEL statement (SURVEYREG), 6556
XPXI= option
PROC MIANALYZE statement, 3847
XREF option
PROC NLIN statement, 4283
PROC NLMIXED statement, 4372
XSIZE option
NLOPTIONS statement (GLIMMIX), 403
NLOPTIONS statement (HPMIXED), 403
NLOPTIONS statement (TCALIS), 403
XSIZE= option
NLOPTIONS statement (CALIS), 894
PROC NLMIXED statement, 4372
XTOL option
NLOPTIONS statement (GLIMMIX), 403
NLOPTIONS statement (HPMIXED), 403
NLOPTIONS statement (TCALIS), 403
XTOL= option
NLOPTIONS statement (CALIS), 894
XVARS option
MODEL statement (GENMOD), 1954

Y= option
GRID statement (KRIGE2D), 2942
GRID statement (SIM2D), 6170
PLOT statement (GLMPOWER), 2656
PLOT statement (POWER), 4883
YCOORD= option
COORDINATES statement (KRIGE2D), 2941
COORDINATES statement (SIM2D), 6170
GRID statement (KRIGE2D), 2942
GRID statement (SIM2D), 6171
YCOORD=option
COORDINATES statement (VARIOGRAM), 7539
YOPTS= option
PLOT statement (GLMPOWER), 2656
PLOT statement (POWER), 4883
YPAIR= option
REPEATED statement (GENMOD), 1962

Z option
MODEL statement (TRANSREG), 7158
ZDATA= option
REPEATED statement (GENMOD), 1962
ZERO= option
MODEL statement (CATMOD), 1125
MODEL statement (TRANSREG), 7155
ZEROBASED option
PROC GLMMOD statement, 2627
ZEROMODEL statement
GENMOD procedure, 1963
ZEROS option
WEIGHT statement (FREQ), 1724
ZETA= option
MODEL statement (GLIMMIX), 2175
MODEL statement (GLM), 2477
MODEL statement (HPMIXED), 2823
MODEL statement (MIXED), 3942
ZROW= option
REPEATED statement (GENMOD), 1962

Your Turn

We welcome your feedback.

- If you have comments about this book, please send them to **`yourturn@sas.com`**. Include the full title and page numbers (if applicable).
- If you have comments about the software, please send them to **`suggest@sas.com`**.

SAS® Publishing Delivers!

Whether you are new to the work force or an experienced professional, you need to distinguish yourself in this rapidly changing and competitive job market. SAS® Publishing provides you with a wide range of resources to help you set yourself apart. Visit us online at support.sas.com/bookstore.

SAS® Press

Need to learn the basics? Struggling with a programming problem? You'll find the expert answers that you need in example-rich books from SAS Press. Written by experienced SAS professionals from around the world, SAS Press books deliver real-world insights on a broad range of topics for all skill levels.

support.sas.com/saspress

SAS® Documentation

To successfully implement applications using SAS software, companies in every industry and on every continent all turn to the one source for accurate, timely, and reliable information: SAS documentation. We currently produce the following types of reference documentation to improve your work experience:

- Online help that is built into the software.
- Tutorials that are integrated into the product.
- Reference documentation delivered in HTML and PDF – **free** on the Web.
- Hard-copy books.

support.sas.com/publishing

SAS® Publishing News

Subscribe to SAS Publishing News to receive up-to-date information about all new SAS titles, author podcasts, and new Web site features via e-mail. Complete instructions on how to subscribe, as well as access to past issues, are available at our Web site.

support.sas.com/spn



sas® THE POWER TO KNOW®

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration. Other brand and product names are trademarks of their respective companies. © 2009 SAS Institute Inc. All rights reserved. 518177_1US.0109